



UNIVERSITÉ DE NANTES
FACULTÉ DES SCIENCES
ET DES TECHNIQUES



UNIVERSITÉ DE NANTES
FACULTÉ DES SCIENCES
ET DES TECHNIQUES

UNIVERSITÉ DE NANTES
UFR Mathématiques
Master 2 Mathématiques appliquées, Ingénierie Statistique

RAPPORT DE STAGE

Effectué à l'Agence nationale de sécurité sanitaire de l'alimentation, de l'environnement et du travail (ANSES)

**Développement d'un outil d'aide à la décision
pour le choix des tests statistiques appropriés aux études utilisées
en évaluation de risque réglementaire
volet environnement/écotoxicologie**



Boutrin Hervé

Maître de Stage : Dr. Emilie Perrat (Evaluatrice Scientifique et technique ; **Tutrice**) &
Dr. Patrice Carpentier (Chef de projet veille et développement scientifiques ; Co-tuteur)
Tuteur académique : Dr. Aymeric Stamm (Ingénieurs de Recherche)

Année universitaire : 2023 / 2024
Stage effectué du 3 Avril 2024 au 30 Septembre 2024

Résumé

Ce rapport présente le développement d'un outil d'aide à la décision en ligne pour le choix des tests statistiques appropriés aux études utilisées en évaluation de risque réglementaire en écotoxicologie. L'objectif est de simplifier et centraliser les informations nécessaires à l'application des tests statistiques, actualisées avec les méthodes récentes, et permettant d'obtenir un indicateur de fiabilité des tests.

Pour cela une analyse détaillée des données d'écotoxicologie comportant des réplicats soumis à plusieurs concentrations de produits chimiques sera effectué. Et nous utilisons une gamme de méthodes statistiques, y compris l'ajustement de modèles dose-réponse, des tests de signification, et des techniques avancées telles que les méthodes bayésiennes, bootstrap et permutation , pour évaluer les effets toxiques et estimer les paramètres clés comme l' EC_{50} et l' EC_{10} .

Ce projet s'inscrit dans un contexte de réglementation stricte et de mise à jour des pratiques statistiques conformément aux directives de l'OCDE et de la réglementation européenne.

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au bon déroulement de mon stage et m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord, j'adresse mes remerciements au Dr Vanessa Mazerolles, cheffe de l'unité Évaluation, Écotoxicologie, Environnement des Intrants du Végétal (U3EiV), pour m'avoir permis d'effectuer mon stage dans son unité, ainsi qu'aux Drs Émilie Perrat et Patrice Carpentier pour m'avoir confié ce projet passionnant, intéressant et enrichissant.

Je remercie également Mme Perrat pour sa supervision, sa confiance, sa patience et le partage de ses connaissances biologiques.

Enfin je remercie toute l'équipe de l'unité U3EiV pour leur accueil, leur esprit d'équipe, leur bienveillance et le partage de leurs connaissances. Enfin, je tiens à remercier toutes les personnes qui m'ont conseillée et relu lors de la rédaction de ce rapport de stage, en particulier les Drs Émilie Perrat et Patrice Carpentier.

Sommaire

1 Table des illustrations	6
2 Table des abréviations	7
3 Glossaire	8
4 Introduction	9
5 Méthodologie	11
5.1 Description des données des essais d'écotoxicité	11
5.1.1 Répliquats et méthode de randomisation	12
5.1.2 Exposition des organismes	13
5.2 Préparation des données pour l'analyse	13
5.2.1 Inspection Visuelle et Nettoyage des Données	13
5.2.2 Vérification de l'Effet Solvant	14
5.2.3 Contrôle des Hypothèses et Transformations des Données	14
5.3 Application de Modèles Statistiques Appropriés (illustration cf figure 5.1)	15
5.4 Formes Mathématiques des Modèles Statistiques Utilisés dans l'application .	16
5.4.1 Modèle de Weibull	16
5.4.2 Modèle log-logistique	16
5.4.3 Modèle logit	16
5.4.4 Modèle probit	16
5.4.5 Modèle de Hill	17
5.4.6 Modèle de Gumbel	17
5.4.7 Modèle de Poisson	17
5.4.8 Modèle binomial négatif	17
5.4.9 Modèle exponentiel (Slob 2002 [3])	17
5.5 Sélection de modèle	18
5.5.1 Vérification des Modèles	24
5.6 Détermination des valeurs de toxicité	25
5.7 Analyse de la significativité des différences observées dans les ECx par la méthode de permutation [23]	26
5.8 Intervalles de confiance des ECx par Bootstrap	26
5.9 Limites et contraintes liées aux données	27
6 Résultats et discussion	28
6.1 Codage et Transformation des Données :	28
6.2 Préparation des données pour l'analyse	29
6.2.1 Visualisation	29
6.2.2 Gestion des outliers	30
6.2.3 Agrégation des Données :	32
6.3 Analyse des données	34
6.3.1 Vérification de la normalité et de l'homoscédasticité des données : . . .	34
6.3.2 Analyse descriptive	37
6.4 Détermination des LOEC/D/L et NOEC/D/L	38
7 Calcul de Puissance	39

8 Modélisation Dose-Réponse	41
8.1 Comparaison des Modèles	43
9 Estimation des ECx ou LCx	47
9.0.1 Test de Permutation et Intervalles de Confiance Bootstrap	48
10 Conclusion & Perspectives	50
A Annexe A : Inventaire des tests	51
B Annexe :Application interface	52
C Annexe :Application architecture	54
D Distributions de Probabilité	201

1 Table des illustrations

Figures

4.1	ANSES	9
5.1	Illustration NOEC, LOEC, EC _x sur courbe dose réponse [21]	28
6.1	Diagramme Essai 201	30
6.2	Boxplot Essai 201	31
6.3	Densité cellulaire à 72h en fonction des concentrations : concaténation	33
6.4	QQplot par concentration	35
6.5	normale <i>vs</i> non normale	36
6.6	scatterplots homoscédasticité <i>vs</i> hétéroscléasticité	36
8.1	Trace log-logistique Pente (b)	45
8.2	Trace log-logistique (EC_{50})	45
8.3	Trace log-normal Moyenne logarithmique	45
8.4	Trace log-normal Variance logarithmique	45
8.5	Trace Weibull Paramètre de forme	45
8.6	Trace Weibull Paramètre d'échelle	45
8.7	ACF pour le modèle log-logistique Pente (b)	46
8.8	ACF pour le modèle log-logistique (EC_{50})	46
8.9	ACF pour le modèle log-normal Moyenne logarithmique	46
8.10	ACF pour le modèle log-normal Variance logarithmique	46
8.11	ACF pour le modèle Weibull Paramètre de forme	46
8.12	ACF pour le modèle Weibull Paramètre d'échelle	46
C.1	Architecture générale de l'application	54
C.2	Diagramme essais standardisés	55

Tableaux

1	Résumé des modèles couramment utilisés avec leurs paramètres possibles (1 à 5)	18
2	Outliers détectés par la méthode de Tukey	31
3	Résultats du test de Shapiro-Wilk pour chaque concentration	34
4	Vérification Homoscédasticité	34
5	Statistiques pour chaque concentration de substance testée (Test 201 OCDE)	37
6	Résultats des tests ANOVA pour déterminer l'effet de chaque concentration sur la croissance des algues (Test 201 OCDE)	38
7	Résultats du test de Kruskal-Wallis pour les différentes concentrations (Test Non-Paramétrique)	39
8	Estimation des paramètres pour les modèles log-logistique, log-normal et Weibull.	42
9	Valeurs de l'AIC	43
10	Facteurs de Bayes	43
11	Taux d'acceptation pour chaque modèle.	44
12	Estimations des doses effectives à 10%, 20%, et 50% pour le modèle Weibull.	47
13	Intervalles de confiance bootstrap à 95% pour les doses effectives estimées.	48
14	Forces et Limites des Tests Statistiques	51
15	Critères de Choix des Tests Statistiques	51
16	Tests Paramétriques: Forces, Limites et Critères de Choix	200
17	Tests Non-Paramétriques: Forces, Limites et Critères de Choix	201

2 Table des abréviations

Abréviation	Signification
EC	Effective Concentration (Concentration Efficace)
ED	Effective Dose
LC	Lethal Concentration
LD	Lethal Dose
LOEC	Lowest Observed Effect Concentration
LOEL	Lowest Observed Effect Level
NOEC	No Observed Effect Concentration
NOEL	No Observed Effect Level
OCDE	Organisation de Coopération et de Développement Économiques

3 Glossaire

Biocide: Ensemble de produits destinés à détruire, repousser ou rendre inoffensifs les organismes nuisibles, à en prévenir l'action ou à les combattre, par une action chimique ou biologique.

Biomasse: La biomasse est l'ensemble de la matière organique d'origine végétale ou animale présente dans un espace fini, un biotope par exemple, à un instant t

ECx (Effective Concentration x%) (Concentration Efficace): La concentration d'une substance/produit qui provoque des effets sublétaux pour x% des organismes testés.

Effet subletal: Les effets sublétaux peuvent affecter, entre autres, la durée de vie, le développement, la croissance de la population, la fertilité et le comportement.

Essai dose-reponse Employé pour estimer la concentration à laquelle un pourcentage donné d'organismes montre une réponse, comme l' EC_{50} . Cette méthode utilise plusieurs concentrations pour établir une courbe dose-réponse, fournissant une vue plus complète de la toxicité de la substance testée.

Essai limite: Utilisé pour déterminer si une substance est toxique à une concentration donnée. Cette approche se concentre sur une ou quelques concentrations spécifiques, souvent à des niveaux élevés, pour évaluer la présence ou l'absence d'effet toxique significatif.

Intrants du végétal : substances et produits phytopharmaceutiques, matières fertilisantes et supports de culture, macro-organismes non indigènes utiles aux végétaux.

ISO : Organisation internationale de normalisation.

LCx/LDx (Lethal Concentration/Dose x%): Concentration ou dose d'une substance/produit qui est mortelle pour x% de la population étudiée.

LOEC (Lowest Observed Effect Concentration) ou LOEL (Lowest Observed Effect Level): La concentration ou dose testée la plus faible d'un contaminant où un effet significatif est observé sur les organismes testés.

Minimum Detectable Difference (MDD) : Une mesure utilisée pour déterminer la plus petite différence entre des groupes de traitement qui peut être détectée de manière statistiquement significative dans une étude expérimentale. Le MDD dépend de la variance des données, de la taille de l'échantillon et du niveau de signification statistique choisi. En écotoxicologie, le MDD permet de déterminer la sensibilité de l'essai pour détecter les effets des substances chimiques sur les organismes testés.

NOEC (No Observed Effect Concentration) ou NOEL (No Observed Effect Level): La concentration ou dose testée la plus élevée d'une substance/produit où aucun effet significatif n'est observé sur les organismes testés.

OCDE (Organisation de Coopération et de Développement Économiques): Organisation internationale qui publie des documents guides standardisés pour la mise en application de nombreux essais de laboratoire, y compris les tests d'écotoxicité.

Organismes non-cibles : organismes qui ne sont pas visés par un traitement ou une substance chimique, mais qui peuvent en être affectés.

4 Introduction

L'Agence nationale de sécurité sanitaire de l'alimentation, de l'environnement et du travail (Anses) assure des missions de veille, d'expertise, de recherche et de référence sur un large champ couvrant la santé humaine, la santé et le bien-être animal, la santé végétale et la santé de l'environnement. Elle offre une lecture transversale des questions sanitaires et appréhende ainsi, de manière globale, les expositions auxquelles l'Homme peut être soumis à travers ses modes de vie et de consommation ou les caractéristiques de son environnement, y compris professionnel. L'Anses informe les autorités compétentes, répond à leurs demandes d'expertise. L'Agence exerce ses missions en étroite relation avec ses homologues européens. Afin d'assurer ses diverses missions, l'agence est découpée en plusieurs pôles et directions :



Figure 4.1: ANSES

J'ai effectué mon stage dans le pôle **Produits réglementés**, et plus précisément au sein de la **DEPR** (Direction de l'Évaluation des Produits Réglementés (DEPR) dont l'**unité d'Évaluation Écotoxicologie et Environnement des Intrants du Végétal (U3EiV)** fait partie. Cette structure permet de remplir ses missions au travers de ses multiples directions. La DEPR évalue, selon la réglementation en vigueur, les risques liés à l'utilisation des substances actives, des produits phytopharmaceutiques et **biocides**, ainsi que les matières fertilisantes, supports de cultures, substances actives et produits biocides. Cette évaluation est réalisée, avant décision d'autorisations de mise sur le marché. En ce qui concerne l'évaluation des intrants du végétal (substances et produits phytopharmaceutiques, matières fertilisantes et supports de culture, macro-organismes non indigènes utiles aux végétaux), la DEPR met en œuvre une expertise scientifique interne et peut faire appel à ses comités d'experts spécialisés dédiés. Pour répondre à des saisines des ministères, elle peut faire appel, en complément des experts internes, à des, aux groupes de travail constitués auprès des experts de l'Anses et peut travailler en liaison avec les autres directions de l'Agence. Mon stage s'est déroulé au sein de l'U3EiV. Cette unité est chargée de l'évaluation scientifique des parties devenir et comportement dans l'environnement et écotoxicologie des dossiers d'autorisation des produits phytopharmaceutiques, matières fertilisantes, supports de cultures et des substances actives. À ce titre, elle évalue les parties correspondantes des dossiers, assure la cohérence des évaluations réalisées dans le respect des réglementations européennes et nationales. Cette unité est dirigée par Vanessa Mazerolles (Cheffe d'unité (CU)), assistée de ses adjoints Émilie Farama (CU adjointe) et Pierre-François Chatton (adjoint au CU). Mon stage au sein de cette unité m'a offert l'opportunité d'observer et de comprendre les processus d'évaluation. Il m'a également permis d'acquérir une expérience concrète en travaillant aux côtés de professionnels expérimentés, ce qui m'a aidé à mieux appréhender les défis et les exigences liés à un domaine très différent du mien : l'évaluation réglementaire en écotoxicologie. Cette expérience a permis d'apporter ma contribution pour améliorer la méthodologie d'évaluation statistique des évaluateurs de risque en développant un outil d'aide à la décision pour le choix des tests statistiques les plus appropriés à employer en fonction des tests **d'écotoxicité** déjà réalisés pour une substance

ou un produit. L'écotoxicologie est une science qui étudie les effets des substances chimiques/produits sur les organismes vivants dans l'environnement (ref : Green et al., 2006 [3]). Cette science adopte une approche hiérarchique, où les processus observés à un certain niveau (moléculaire) sont expliqués par des mécanismes à un niveau inférieur (par exemple, l'organisme) et peuvent avoir des conséquences à un niveau supérieur (population ou écosystème). Cette structure complexe permet une évaluation plus intégrée et holistique (c'est-à-dire globale) des effets des polluants. Afin de répondre aux requis réglementaires européens concernant l'évaluation des risques des substances actives et produits phytopharmaceutiques (règlement (CE) n°1107/2009 ref : eu2009 [1]), de nombreux tests en laboratoire et en conditions plus réalistes sont nécessaires pour caractériser le comportement de l'intrant dans l'environnement ainsi que ses effets sur les organismes non-cibles. La majorité de ces essais d'écotoxicité sont **standardisés** et suivent des documents guides lignes directrices pour chaque groupe **d'organismes non-cibles**.

- Oiseaux via les documents OCDE 223[40] ou l'OCDE 205[30] par exemple
- Poissons via les documents OCDE 210[32] par exemple
- Invertébrés d'eau douce (e.g. daphnie, chironomes) OCDE 211[33] ou 218[37] par exemple
- Pollinisateurs via les documents OCDE 245[43] par exemple
- Arthropodes, ESCORT II [49] par exemple
- Organismes du sol via les documents OCDE 222[39] par exemple
- Activité microbienne (e.g. nitrification du sol) via les documents OCDE 216[36] par exemple
- Plantes via les documents OCDE 208[46] par exemple

Ces lignes directrices d'essais sont notamment celles définies via les normes OCDE/ISO [2]. Ces tests permettent d'obtenir des valeurs de toxicité pour chaque groupe d'organismes non-cibles. Pour cela, les organismes sont exposés à différentes concentrations, doses ou niveaux d'exposition afin d'établir des courbes doses-réponses établissant concrètement le lien entre concentration/dose en substance/polluant et effets observés (inhibition de croissance, réduction du nombre de jeunes...) selon des protocoles expérimentaux clairement établis. Toutefois, les lignes directrices ne précisent pas nécessairement les tests statistiques qui seraient les plus appropriés pour déterminer ces valeurs de toxicité au vu du protocole expérimental et du type de résultats. Le document guide OCDE 54 (2006)[2], présente de manière très générale des propositions adaptées à l'écotoxicologie en fonction du type d'essai. Depuis la parution de ce document, de nouvelles méthodes statistiques ont été proposées afin d'améliorer l'analyse des données des essais réalisés en écotoxicologie. De fait, les discussions entre évaluateurs de risque incluent les meilleures méthodes statistiques à utiliser pour analyser les données de toxicité issues de ces essais standardisés. Ce stage vise à de développer un guide des bonnes pratiques d'analyses statistiques pour les tests écotoxicologiques, sous forme d'une application web via R Shiny. Cette application a pour but de guider les évaluateurs de risque dans le choix des tests statistiques les plus appropriés, en tenant compte des caractéristiques spécifiques des données obtenues lors des tests en laboratoire. Elle intégrera les recommandations des documents guides et les avancées récentes en méthodes statistiques, afin d'améliorer la précision et la reproductibilité des évaluations de risque. L'outil d'appui statistique est développé en anglais et permettra à l'évaluateur

de risque de :

- De disposer d'un outil d'aide à la décision afin de mieux justifier le choix du test statistique le plus pertinent pour l'essai concerné en se basant sur les contraintes et le protocole expérimental défini dans les normes ;
- De réaliser le/les test(s) statistique(s) adéquat(s) tout en mettant en évidence la validité des conditions d'application du test employé et, si possible, sa/leur puissance et sa/leur robustesse ;
- De fournir, sous forme d'un rapport succinct, un résumé des résultats associés les plus pertinents à l'issue de l'utilisation de l'application en ligne.

De plus, ce projet permet également de :

- Simplifier et centraliser les informations sur les tests statistiques disponibles.
- Mettre à jour les méthodes statistiques employées par les évaluateurs de risque avec des techniques récentes.
- Développer un indicateur de fiabilité des tests utilisés (puissance post-hoc).

5 Méthodologie

Ce travail s'appuie sur le **document guide OCDE 54 (2006)**[2], qui définit les protocoles pour la préparation et le traitement des données obtenues dans les études d'écotoxicité à la suite de leur mise en application "biologique" dans les laboratoires d'essai. La première étape du développement de l'outil a consisté en une analyse exhaustive de la littérature existante, en particulier les documents guides pour les essais d'écotoxicité, de l'**OCDE 54 (2006)** et des ouvrages scientifiques récents sur les méthodes statistiques appliquées en écototoxicologie (ref : Ggreen et al., 2018),[3]. Cette revue de la littérature a permis d'identifier les méthodes statistiques couramment utilisées, leurs avantages et leurs limitations, ainsi que les recommandations actuelles pour l'analyse des données d'écotoxicité. En parallèle de l'analyse de la littérature scientifique, j'ai consacré une partie importante de mon temps à l'apprentissage de RShiny grâce à des ressources disponibles en ligne. Cette maîtrise de RShiny a été essentielle pour la visualisation interactive et l'analyse des données écotoxicologiques afin de mieux comprendre les résultats expérimentaux et d'améliorer l'interface utilisateur pour l'outil d'aide à la décision développé.

5.1 Description des données des essais d'écotoxicité

Les tests d'écotoxicité consistent à exposer des organismes non cibles à un traitement en utilisant soit une concentration unique (**essai limite**), soit plusieurs concentrations différentes suivant un gradient géométrique (**essai dose-réponse**), permettant ainsi d'évaluer les effets potentiels du traitement sur ces organismes. Le **design expérimental** en écototoxicologie inclut souvent la **randomisation des traitements** appliqués aux sujets d'étude pour éviter les biais dans la collecte des données. La **réplication**, ou répétition des expériences, est essentielle pour assurer la fiabilité et la validité des résultats obtenus. Les types de données en lien avec ce design expérimental peuvent être classés selon leur nature et la manière dont elles seront analysées. Cette classification aide à déterminer les tests statistiques les plus appropriés et à interpréter les résultats de manière précise.

1. Données Quantales (*i.e. quantitatives Binaires*):

Les données binaires sont souvent utilisées pour enregistrer la présence ou l'absence d'un effet, comme la mortalité ou la survie des organismes testés. Par exemple, dans un test de toxicité, 0 pourrait indiquer la survie et 1 la mortalité.

2. Données Continues:

Ces données incluent des mesures qui varient sur un continuum (*i.e. en fonction de la concentration de la substance toxique, appliquée selon un gradient continu*), telles que la concentration de substances chimiques dans l'eau ou la **biomasse** des organismes après exposition. Lors de mon stage, ce type de données a été utilisé pour établir des modèles **dose-réponse**.

3. Données Discrètes:

Ces données occupent une position intermédiaire entre les données quantales et continues dans les études d'écotoxicité; Elles apparaissent généralement sous forme de dénominbrements.

4. Et définies comme des données Discrètes de comptages :

- **Nominales:** Les données nominales, comme le type de tissu affecté par un toxique, ne comportent **aucun ordre implicite**. Elles servent à catégoriser les échantillons sans échelle de mesure.
- **Ordinales:** Ces données impliquent un **ordre**, comme le degré de toxicité observé (faible, moyen, fort), qui peut être utilisé pour des analyses de tendance.
- **Intervalle:** Moins fréquentes, les données d'intervalle comme la température mesurée en degrés Celsius, où le zéro est arbitraire, permettent de réaliser des analyses plus complexes en conservant la signification des différences entre les valeurs.

Les données traitées comprennent diverses mesures, telles que **la survie, la croissance, la reproduction**, et d'autres indicateurs biologiques pertinents pour évaluer les effets écotoxicologiques. Chaque mesure est enregistrée pour chaque réplicat et chaque concentration nominale, ce qui génère un ensemble de données complexes à exploiter.

5.1.1 Réplicats et méthode de randomisation

Les réplicats sont analysés pour vérifier leur cohérence et sont essentiels pour estimer la variabilité naturelle, fournissant ainsi une base solide aux tests statistiques. Toutefois, toute variabilité excessive ne pourra être prise en compte que lors de l'interprétation des résultats, car les études sont déjà réalisées. La robustesse et la répétabilité des résultats expérimentaux en écotoxicologie reposent sur l'utilisation de réplicats et la mise en œuvre de méthodes de randomisation rigoureuses. En écotoxicologie, les réplicats consistent en des répétitions indépendantes d'une même condition expérimentale, capturant ainsi la variabilité naturelle des organismes et des conditions environnementales. En multipliant les réplicats, il est possible de calculer des mesures de variabilité, telles que l'écart-type ou l'intervalle de confiance, augmentant la précision des estimations des valeurs de toxicité. L'utilisation de réplicats dans différents groupes de traitement soumis à des concentrations variées de substances, souvent qualifiées de toxiques en fonction de leurs effets, garantit que les données obtenues sont à la fois représentatives et fiables. Ces réplicats sont essentiels pour évaluer l'effet

dose-réponse et minimiser la variabilité expérimentale, assurant ainsi des conclusions basées sur des réponses biologiques véritables. Les données écotoxicologiques peuvent être biaisées de diverses manières, et les réplicats jouent un rôle clé dans l’atténuation de ces biais. Par exemple, le biais de sélection, qui survient lorsque les échantillons ne représentent pas la population d’intérêt, peut être réduit en augmentant la diversité des échantillons grâce aux réplicats. Le biais de mesure, lié à l’imprécision des instruments, peut être identifié et corrigé avec des réplicats, améliorant ainsi la précision des données. De même, les biais de confondance, de traitement et d’observateur peuvent être minimisés par une conception expérimentale rigoureuse et l’utilisation de réplicats. La randomisation, quant à elle, est cruciale pour minimiser les biais expérimentaux. En répartissant aléatoirement les traitements et les réplicats, on s’assure que les variations observées sont dues aux traitements appliqués et non à des facteurs confondants, renforçant ainsi la validité interne des résultats.

5.1.2 Exposition des organismes

Les organismes sont exposés à différents types de concentrations (C), doses (D) ou niveaux d’exposition (L). Pour simplifier, nous nous référerons uniquement aux concentrations (C) dans la suite de ce document, bien que les principes s’appliquent également aux doses (D) et niveaux d’exposition (L). Les expériences incluent différentes concentrations des substances testées, disposées selon un gradient géométrique, allant d’un contrôle (sans substance) à plusieurs niveaux de concentration. Le ou les contrôles sont toujours comparés à une ou plusieurs concentrations de la substance testée, et un contrôle solvant est souvent inclus si un solvant est utilisé.

L’organisation des données inclut typiquement les éléments suivants :

- **Contrôle :**

Groupe témoin sans exposition à la substance testée, servant de référence pour comparer les effets des différentes concentrations.

- **Contrôle solvant :**

Groupe témoin exposé uniquement au solvant utilisé pour servir de vecteur à la substance testée, pour évaluer l’effet potentiel du solvant lui-même.

- **Concentrations d’essai :**

Différents niveaux de la substance testée, généralement disposés en gradient géométrique pour couvrir une large gamme de concentrations.

Ces concentrations permettent de déterminer la relation dose-réponse, c’est-à-dire comment les effets varient en fonction de la concentration de la substance.

5.2 Préparation des données pour l’analyse

Conformément aux protocoles détaillés dans le document **OCDE 54**, la préparation des données pour l’analyse statistique dans les études d’écotoxicité est une étape cruciale. Cette phase inclut plusieurs étapes spécifiques qui garantissent l’intégrité et la validité des analyses suivantes, pour chaque type de données, que ce soit pour des essais limite ou dose-réponse.

5.2.1 Inspection Visuelle et Nettoyage des Données

La première étape consiste à réaliser une inspection visuelle des données afin d’identifier les valeurs aberrantes (appelées « outliers ») et d’autres anomalies potentielles. Cette inspection permet de détecter des incohérences, telles que des réplicats qui divergent de manière

significative des autres, ce qui pourrait indiquer des erreurs expérimentales ou des biais systématiques. Cette phase de vérification est essentielle pour éviter que ces anomalies ne biaissent les résultats finaux.

5.2.2 Vérification de l'Effet Solvant

Les solvants ne sont pas systématiquement utilisés dans tous les essais dose-réponse ou les essais limite. Leur usage dépend de la solubilité de la substance testée. Un test statistique non paramétrique, tel que le test de Wilcoxon-Mann-Whitney, est appliqué pour comparer ces deux groupes. Ce test est choisi en raison de sa robustesse aux distributions non normales, souvent observées dans les données écotoxicologiques. Si le test montre qu'il n'y a pas de différence statistiquement significative entre les deux groupes ($p > 0.05$), les données du groupe contrôle solvant et du groupe contrôle peuvent être regroupées (concaténées) en un seul groupe contrôle (qui sera noté "0"). En revanche, si une différence significative est détectée, une comparaison des moyennes est réalisée pour déterminer quel groupe présente la réponse la plus faible :

- Si la moyenne de la réponse dans le groupe contrôle solvant est inférieure à celle du groupe contrôle pour des paramètres comme la reproduction, cela suggère un effet毒ique du solvant. Dans ce cas, seules les données du groupe contrôle solvant sont utilisées pour les analyses ultérieures.
- À l'inverse, si la moyenne de la réponse dans le groupe contrôle est inférieure à celle du groupe contrôle solvant pour des paramètres comme la mortalité, cela indique un effet du contrôle. Seules les données du groupe contrôle sont alors retenues pour les analyses.

5.2.3 Contrôle des Hypothèses et Transformations des Données

Avant d'appliquer des méthodes statistiques spécifiques, il est nécessaire de vérifier les hypothèses sous-jacentes des analyses envisagées, telles que la normalité des résidus et l'homogénéité des variances. Si ces hypothèses ne sont pas respectées, des transformations de données pourraient être nécessaires. Par exemple, pour les données continues qui montrent une variance hétérogène, une transformation logarithmique pourrait être appliquée pour stabiliser la variance[3]. Les principales hypothèses à respecter incluent :

1. **Indépendance des observations:** Les observations doivent être indépendantes les unes des autres pour éviter les biais dans l'estimation des paramètres. Cette hypothèse est cruciale pour assurer la fiabilité des conclusions tirées du modèle.
2. **Spécification correcte du modèle:** Le modèle choisi doit correspondre à la relation réelle entre les variables. Une mauvaise spécification peut entraîner des biais importants et des erreurs dans l'interprétation des résultats.
3. **Convergence de l'algorithme d'estimation:** Lors de l'estimation des paramètres des modèles non linéaires, il est essentiel que l'algorithme converge vers une solution optimale globale optimale, et non vers un minimum local. La convergence assure que les estimations des paramètres sont fiables et reproductibles.
4. **Homoscédasticité:** Les résidus du modèle doivent présenter une variance constante (homoscédasticité) à travers toutes les valeurs des variables indépendantes. Si cette condition n'est pas respectée, les estimations des paramètres pourraient être inefficaces, et les tests statistiques pourraient conduire à des conclusions incorrectes.

5. **Normalité des erreurs:** Bien que non indispensable pour toutes les méthodes d'estimation, la normalité des résidus est souvent souhaitée, surtout si l'on souhaite réaliser des tests de significativité ou construire des intervalles de confiance.
6. **Multicolinéarité:** Dans les modèles multi-paramétriques, il est important de s'assurer qu'il n'y a pas de multicolinéarité excessive entre les variables explicatives, car cela pourrait rendre les estimations des paramètres instables.
7. **Sensibilité aux valeurs aberrantes:** Les modèles non linéaires peuvent être sensibles aux valeurs aberrantes. Il est important d'identifier et de traiter ces valeurs avant l'analyse.
8. **Identifiabilité des paramètres:** Tous les paramètres doivent être identifiables, c'est-à-dire qu'ils doivent pouvoir être estimés de manière unique à partir des données disponibles.

Ces hypothèses doivent être systématiquement vérifiées pour garantir que les modèles non linéaires produisent des résultats fiables et interprétables. Pour plus de détails sur ces aspects techniques, on peut se référer à l'ouvrage de Bates et Watts (1988) [53], qui fournit une analyse approfondie de ces considérations dans le contexte des modèles non linéaires.

5.3 Application de Modèles Statistiques Appropriés (illustration cf figure 5.1)

La modélisation statistique en écotoxicologie représente une pierre angulaire de l'analyse des données, permettant de quantifier et de prédire les effets toxiques en fonction des concentrations de substances chimiques. L'application de modèles adaptés[54] est essentielle pour capturer avec précision la relation dose-réponse et garantir des conclusions robustes. Selon les caractéristiques spécifiques des données et les objectifs de l'étude, divers modèles peuvent être employés, chacun offrant des avantages uniques. Les **modèles dose-réponse non linéaires** sont souvent privilégiés pour leur capacité à décrire fidèlement la complexité des interactions biologiques. Ces modèles, tels que les modèles de Hill, Weibull, ou log-logistique, permettent de représenter la relation entre la concentration d'un toxique et l'effet observé de manière plus précise que les modèles linéaires. Le choix du modèle dépend de plusieurs critères, dont la qualité de l'ajustement aux données, la simplicité du modèle, et la nature des données qu'elles soient continues, discrètes ou binaires. Lorsque les données de dénombrement présentent une grande variabilité, elles peuvent être traitées comme des réponses continues après une transformation adéquate, telle qu'une transformation en racine carrée. Cette approche permet de mieux répondre aux exigences des tests statistiques en minimisant l'impact des écarts-types disproportionnés. Dans les cas où les données de dénombrement suivent une distribution de Poisson, l'utilisation d'un **modèle linéaire mixte généralisé (GLMM)**[3] se révèle particulièrement efficace, offrant une approche robuste et flexible pour des données présentant une grande variabilité. Les méthodes statistiques doivent être soigneusement choisies en fonction du type de données à analyser. Pour les données binaires, les tests de proportions sont souvent utilisés, tandis que les données continues ou ordinaires nécessitent des modèles plus sophistiqués pour capter la nature complexe des interactions biologiques. Par exemple, le **modèle probit** est couramment employé pour estimer les effets toxiques à faible concentration, alors que le **modèle à seuil** (comme le modèle en bâton de hockey) est idéal pour décrire des situations où l'effet toxique n'apparaît qu'après avoir atteint un certain seuil de concentration. L'application rigoureuse des modèles statistiques ne se limite pas à leur sélection ; il est nécessaire de vérifier les hypothèses sous-jacentes

pour chaque modèle utilisé. Par exemple, le **modèle de Weibull** nécessite une vérification de la distribution asymétrique des résidus, tandis que le **modèle de Hill** doit correspondre à une relation sigmoïde avec une phase de montée rapide suivie d'une saturation. De même, le **modèle log-logistique** est vérifié pour son aptitude à décrire des courbes dose-réponse asymétriques. Enfin, bien que les modèles non linéaires offrent une flexibilité considérable, ils nécessitent une validation attentive pour garantir la fiabilité des résultats. En combinant une sélection judicieuse des modèles avec une vérification rigoureuse des hypothèses, il est possible de garantir que les conclusions écotoxicologiques reposent sur une base statistique solide, offrant ainsi des prédictions précises et utiles pour la gestion des risques.

5.4 Formes Mathématiques des Modèles Statistiques Utilisés dans l'application

Dans cette sous-section, nous présentons les formes mathématiques des modèles statistiques les plus couramment utilisés en écotoxicologie : Weibull, log-logistique, logit, probit, Hill, Gumbel, Poisson, binomial négatif, et exponentiel (Slob).

5.4.1 Modèle de Weibull

Le modèle de Weibull, souvent utilisé pour modéliser des données de survie, est donné par :

$$Y = a \cdot \left(1 - e^{-\left(\frac{C}{b}\right)^c}\right) + \delta \quad (1)$$

où Y est la réponse, C est la concentration.

5.4.2 Modèle log-logistique

Utilisé pour des données continues et quantales avec une relation dose-réponse sigmoïde :

$$Y = \frac{a}{1 + \left(\frac{C}{EC_{50}}\right)^b} + \delta \quad (2)$$

où a est la réponse maximale, C est la concentration.

5.4.3 Modèle logit

Adapté pour des relations dose-réponse sigmoïdes, souvent pour des données quantales ou continues :

$$Y = a + \frac{b - a}{1 + e^{-k(C-EC_{50})}} + \delta \quad (3)$$

où C est la concentration.

5.4.4 Modèle probit

Le modèle probit, utilisé principalement pour les données quantales, est exprimé par :

$$Y = a + b \cdot \Phi^{-1} \left(\frac{C}{EC_{50}} \right) + \delta \quad (4)$$

où Φ^{-1} est la fonction quantile de la distribution normale

5.4.5 Modèle de Hill

Adapté pour des réponses continues avec une courbe sigmoïde, souvent utilisé en biologie :

$$Y = \frac{E_{\max} \cdot C^n}{EC_{50}^n + C^n} + \delta \quad (5)$$

où C est la concentration et n est le coefficient de Hill.

5.4.6 Modèle de Gumbel

Souvent utilisé pour modéliser les données de survie, ce modèle est donné par :

$$Y = a \cdot \left(1 - e^{-e^{\frac{C-b}{c}}} \right) \quad (6)$$

5.4.7 Modèle de Poisson

Utilisé pour des données de comptage, lorsque la variance est égale à la moyenne :

$$P(Y = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (7)$$

où λ est le taux de comptage.

5.4.8 Modèle binomial négatif

Utilisé pour des données de comptage avec surdispersion (variance > moyenne) :

$$P(Y = k) = \frac{\Gamma(k + \theta)}{\Gamma(\theta)k!} \left(\frac{\theta}{\theta + \mu} \right)^\theta \left(\frac{\mu}{\theta + \mu} \right)^k \quad (8)$$

où μ est la moyenne, et θ est le paramètre de dispersion.

5.4.9 Modèle exponentiel (Slob 2002 [3])

Utilisé pour modéliser des réponses exponentielles :

$$Y = a \exp \left(-\frac{C}{b} \right) \quad (9)$$

où C est la concentration.

Modèle	Données	Forme	Paramètres
Weibull 1,2,3,4	Continues	Asymétrique Sigmoïde si $\alpha > 1$	a (amplitude), b (échelle), c (forme), δ (décalage)
Hill 2,3,4,5	Continues	Sigmoïde	E_{\max} (effet max), EC_{50} , n (pente), δ (décalage)
Log-logistique 2,3,4	Continues	Sigmoïde	a (amplitude), EC_{50} ,
	Quantales		b (pente), δ (décalage)
Probit 2,3	Quantales	Sigmoïde	a (intercept), b (pente), δ (décalage)
Logit 2,3,4	Quantales	Sigmoïde	a (intercept), b (pente), k (taux de croissance), δ (décalage)
	Continues		
Gumbel	Quantales	Asymétrique	a (amplitude), b (position), c (pente)
Poisson	Comptage	Exponentiel	λ (taux de comptage)
binomial négatif	Comptage	Exponentiel	μ (moyenne), θ (dispersion)
exponentiel 1,2,3	Continues	Exponentiel	a (amplitude), b (échelle), c (décroissance)
	Quantales		

Table 1: Résumé des modèles couramment utilisés avec leurs paramètres possibles (1 à 5)

5.5 Sélection de modèle

Dans le domaine de la sélection de modèles, le **Critère d'Information d'Akaike (AIC)**[8] et le **Critère d'Information Bayésien (BIC)**[7] figurent parmi les outils les plus fréquemment utilisés. Ces critères offrent des approches quantitatives pour comparer différents modèles en intégrant une pénalisation de la complexité pour éviter le **surajustement**. En complément de ces méthodes, le facteur de Bayes se présente également comme une alternative puissante, permettant d'évaluer la probabilité relative des modèles tout en prenant en compte les distributions a priori.

1. Critère d'Information d'Akaike (AIC):

L'AIC est fondé sur l'idée de minimiser la divergence de Kullback-Leibler, qui mesure la perte d'information entre le modèle ajusté et la "vérité" des données. Le critère AIC est défini comme :

$$AIC = -2 \log(L) + 2k$$

Où :

- $\log(L)$ est le logarithme de la vraisemblance du modèle.
- k est le nombre de paramètres estimés dans le modèle.

L'objectif est de minimiser l'AIC. Un AIC plus faible indique un modèle qui offre un bon ajustement aux données avec un nombre raisonnable de paramètres. Cependant, l'AIC peut parfois favoriser des modèles plus complexes, car il pénalise faiblement le nombre de paramètres.

2. Critère d'Information Bayésien (BIC) :

Le BIC, également connu sous le nom de critère de Schwarz, introduit une pénalisation plus forte pour la complexité du modèle, en particulier lorsque la taille de l'échantillon est grande. Le BIC est défini comme suit :

$$BIC = -2 \log(L) + k \log(n)$$

Où :

- n est la taille de l'échantillon.
- $\log(L)$ est le logarithme de la vraisemblance du modèle.
- k est le nombre de paramètres estimés.

Comme pour l'AIC, un BIC plus faible est préférable. Cependant, en raison de la pénalisation plus forte liée à la taille de l'échantillon, le BIC tend à favoriser des modèles plus simples, particulièrement lorsque l'échantillon est grand. L'AIC privilégie un bon ajustement avec une pénalisation légère de la complexité. Il est souvent utilisé lorsque l'objectif est de faire des prédictions et que l'on préfère un modèle qui capture plus de nuances des données, même au risque de surajuster. Tandis que, le **BIC** : Privilégie des modèles plus parcimonieux avec une pénalisation plus sévère pour la complexité. Il est préféré lorsque l'on cherche à identifier le modèle "vrai" parmi un ensemble de candidats, en particulier dans un contexte bayésien. En pratique, lors de la vérification et de la sélection des modèles, il est recommandé d'examiner à la fois l'AIC et le BIC pour guider la décision. Si les deux critères suggèrent le même modèle, cela renforce la confiance dans le choix du modèle. Cependant, si les critères AIC et BIC favorisent des modèles différents, le choix dépendra de la priorité donnée à l'ajustement par rapport à la simplicité. Dans les situations où le nombre total d'observations n est limité, une correction de l'AIC, notée **AICc**, est utilisée pour ajuster la pénalisation en fonction de la taille de l'échantillon :

$$AICc = -2 \log(L) + 2k \frac{n}{n - k - 1}$$

Cette correction est particulièrement importante lorsque le ratio entre le nombre de paramètres et la taille de l'échantillon est élevé, garantissant ainsi une évaluation plus précise du modèle. Lorsque les valeurs d'AICc de deux modèles sont proches, il peut être utile de calculer la probabilité qu'un modèle soit meilleur qu'un autre. Cette probabilité est donnée par :

$$\text{Probabilité} = \frac{e^{-D/2}}{1 + e^{-D/2}}$$

où D est la différence entre les valeurs AICc des deux modèles. Cette formule permet d'évaluer la supériorité relative d'un modèle par rapport à un autre, renforçant ainsi

le processus décisionnel. Le **principe de parcimonie**, également connu sous le nom de Rasoir d'Occam, recommande de préférer le modèle le plus simple parmi ceux offrant un ajustement adéquat. Le BIC est particulièrement adapté à ce principe, en pénalisant fortement les modèles trop complexes, ce qui encourage la sélection d'un modèle qui explique les données avec le moins de paramètres possible.

3. Approche Bayésiennes [11] :

Cette approche bayésienne, essentiels pour l'analyse des données écotoxicologiques, offre un cadre robuste pour la comparaison des modèles en permettant d'incorporer des informations a priori et d'obtenir des probabilités a posteriori interprétables. Cette méthode se distingue notamment par l'utilisation des **facteurs de Bayes** (Bayes Factors, BF), qui permettent de comparer directement les probabilités a posteriori de différents modèles en tenant compte de la vraisemblance des données sous chacun d'eux.

Pour appliquer la méthode de sélection de modèles bayésienne, il est essentiel de commencer par définir clairement les modèles en compétition. Le modèle nul, souvent considéré comme modèle de référence, est désigné par M_0 , tandis que M_1 représente le modèle alternatif que l'on souhaite comparer au modèle nul. Chacun de ces modèles est associé à un ensemble de paramètres θ_i , pour lequel on doit spécifier une distribution a priori $\pi_i(\theta)$. Cette distribution intègre les connaissances préalables sur les paramètres, par exemple, si des études antérieures suggèrent que la moyenne d'une distribution est centrée autour d'une certaine valeur, cette information sera incluse dans $\pi(\theta)$. Une fois les modèles définis, il faut spécifier les distributions a priori pour les paramètres sous chaque modèle. Pour le modèle M_0 , la distribution a priori $\pi_0(\theta)$ est choisie en fonction des hypothèses formulées pour M_0 . De même, pour le modèle M_1 , une autre distribution a priori $\pi_1(\theta)$ est déterminée. La fonction de vraisemblance $f^{(n)}(\text{données} \mid \theta)$ est ensuite utilisée pour représenter la probabilité d'observer les données pour un ensemble de paramètres donné. Enfin, le facteur de Bayes $B_{0/1}$ est calculé pour comparer la probabilité marginale des données sous le modèle M_0 par rapport à celle sous le modèle M_1 .

$$B_{0/1} = \frac{\text{Vraisemblance marginale de } M_0}{\text{Vraisemblance marginale de } M_1}$$

Ou en notation plus détaillée :

$$B_{0/1} = \frac{\int_{\Theta_0} \pi_0(\theta_0) f^{(n)}(\text{données} \mid \theta_0) d\theta_0}{\int_{\Theta_1} \pi_1(\theta_1) f^{(n)}(\text{données} \mid \theta_1) d\theta_1}$$

Ce facteur est déterminant pour interpréter les résultats : lorsque $B_{0/1}$ est supérieur à 1, les données fournissent un soutien plus fort en faveur du modèle M_0 , suggérant sa préférence. En revanche, si $B_{0/1}$ est inférieur à 1, cela indique que le modèle M_1 est préféré, les données le soutenant davantage. Le facteur de Bayes $B_{0/1}$ pour comparer deux modèles non linéaires $Y = f(X, \theta) + \epsilon$, M_0 et M_1 est donné par :

$$B_{0/1} = \frac{\int_{\Theta_0} f_0(Y \mid X_0, \theta_0) \pi_0(\theta_0) d\theta_0}{\int_{\Theta_1} f_1(Y \mid X_1, \theta_1) \pi_1(\theta_1) d\theta_1}$$

Où:

- $f_0(Y | X_0, \theta_0)$: Est la fonction de vraisemblance du modèle M_0 , c'est-à-dire la probabilité des données Y sachant X_0 et les paramètres θ_0 .
- $\pi_0(\theta_0)$: Est la distribution a priori des paramètres θ_0 pour le modèle M_0 .
- Θ_0 : Est l'espace des paramètres pour M_0 .
- $f_1(Y | X_1, \theta_1)$ et $\pi_1(\theta_1)$: Sont les équivalents pour le modèle M_1 .
- $d\theta_0$ et $d\theta_1$: Indiquent l'intégration sur tous les paramètres possibles pour chaque modèle.

Par exemple, un $B_{0/1} = 0,5$ signifie que les données sont deux fois plus probables sous M_1 que sous M_0 , et donc on pourrait préférer le modèle sous la forme :

$$Y = f(X_1, \theta_1) + \epsilon$$

Une remarque importante est que $B_{1/0}$ est simplement l'inverse de $B_{0/1}$, soit :

$$B_{1/0} = \frac{1}{B_{0/1}}$$

Après avoir calculé les facteurs de Bayes, les probabilités a posteriori pour les modèles peuvent être obtenues. Si l'on veut estimer la probabilité qu'un modèle soit correct après avoir observé les données, on utilise la formule suivante :

$$P(M_1 | \text{données}) = \frac{B_{1/0} \cdot P(M_1)}{P(M_0) + B_{1/0} \cdot P(M_1)}$$

Par exemple, si la probabilité a priori $P(M_1)$ est de 0,7 (indiquant une croyance forte que M_1 est correct avant d'observer les données) et que le facteur de Bayes calculé $B_{1/0}$ est de 5, cela renforcerait considérablement la probabilité a posteriori en faveur de M_1 . Il est possible de comparer plusieurs modèles en calculant les facteurs de Bayes entre chaque paire de modèles, grâce à la transitivité du facteur de Bayes.

Par exemple, pour comparer trois modèles M_0 , M_1 , et M_2 , on peut calculer :

$$B_{0/1} = \frac{M_0(x)}{M_1(x)} \quad \text{et} \quad B_{1/2} = \frac{M_1(x)}{M_2(x)}$$

Puis, on combiner les facteurs pour comparer M_0 à M_2 :

$$B_{0/2} = B_{0/1} \times B_{1/2} = \frac{M_0(x)}{M_2(x)}$$

Ce qui nous donnera :

- Si $B_{0/1} > 1$, cela signifie que les données soutiennent davantage M_0 par rapport à M_1 .
- Si $B_{1/2} > 1$, cela signifie que les données soutiennent davantage M_2 par rapport à M_1 .

- On aura donc dans ce cas $B_{0/2} > 1$, c'est à dire que les données soutiennent davantage M_2 .

Des valeurs élevées de $B_{0/1}$ ou $B_{1/0}$ indiquent une preuve plus forte en faveur de l'un ou l'autre modèle.

A priori : [14][15][16][17][18]

Le choix des a priori est un élément central de l'inférence bayésienne et doit être fait avec soin. Il s'agit d'un compromis entre l'incorporation des connaissances préalables et la flexibilité pour laisser les données guider l'analyse. Dans le contexte de l'écotoxicologie, où les décisions sont souvent prises avec des données limitées et incertaines, l'utilisation judicieuse des a priori permet d'améliorer la précision des estimations et la robustesse des conclusions.

1. A priori informatifs

Un a priori informatif est une distribution qui incorpore des connaissances préexistantes ou des croyances fortes sur les paramètres. Ces connaissances peuvent provenir d'études antérieures, d'expertise dans le domaine, ou de résultats d'expériences précédentes. Par exemple, si plusieurs études ont déjà montré que la probabilité de mortalité sous un certain traitement est faible, un a priori informatif pourrait être une distribution bêta très concentrée autour de faibles valeurs. L'avantage de cet a priori est qu'il permet d'intégrer de manière explicite ces connaissances dans l'analyse, ce qui peut conduire à des conclusions plus robustes, surtout lorsque les nouvelles données sont limitées.

2. A priori non informatifs

Les a priori non informatifs, également appelés a priori neutres ou a priori plats, sont des distributions qui n'incorporent aucune information spécifique sur les paramètres. Ils sont utilisés pour minimiser l'influence de l'a priori sur les résultats, permettant aux données de jouer un rôle dominant. Un exemple classique est l'utilisation d'une distribution uniforme ou d'une distribution bêta(1,1). Ce type d'a priori est utile lorsqu'il y a peu ou pas de connaissances préalables sur les paramètres ou lorsque l'on souhaite que les données soient le principal moteur de l'inférence.

3. A priori partiellement informatifs

Les a priori partiellement informatifs sont un compromis entre les a priori informatifs et non informatifs. Ils intègrent une certaine quantité d'information préalable tout en laissant de la place aux données pour influencer l'inférence. Par exemple, dans une situation où l'on a des connaissances approximatives sur les paramètres mais où l'on souhaite rester ouvert à des résultats inattendus, un a priori partiellement informatif peut être une distribution bêta avec des paramètres modérément concentrés. Cela permet de capturer les connaissances préexistantes tout en restant suffisamment flexible pour être mis à jour de manière significative par les nouvelles données.

4. A priori conjugués

Un a priori conjugué est un a priori qui, lorsqu'il est combiné avec une certaine fonction de vraisemblance, produit une distribution a posteriori de la même famille que l'a priori. Par exemple, une distribution bêta est conjuguée à la distribution binomiale, ce qui signifie que si l'a priori est une distribution bêta et que la vraisemblance est binomiale, la distribution a posteriori sera également une distribution bêta. L'utilisation d'a priori conjugués simplifie souvent les calculs et permet de dériver facilement les distributions a posteriori. Cette propriété est particulièrement utile lorsque l'on doit

mettre à jour les croyances de manière itérative avec l'arrivée de nouvelles données.

5. A priori empiriques

Les a priori empiriques sont dérivés directement des données. Au lieu d'être spécifiés indépendamment des données, ils sont construits en utilisant des statistiques résumées ou des estimations provenant des données elles-mêmes. Par exemple, dans un contexte où l'on a une grande quantité de données historiques, on pourrait utiliser la moyenne et la variance des données pour construire une distribution a priori qui reflète ces caractéristiques. Les a priori empiriques permettent d'adapter les analyses aux données spécifiques tout en conservant une base bayésienne.

Une fois que le modèle statistique et le type d'a priori ont été spécifiés, l'étape suivante consiste à examiner les données collectées et à les utiliser pour mettre à jour les croyances via la distribution a posteriori. Cette mise à jour prend en compte à la fois les données et les a priori, et permet de produire des inférences qui sont enrichies par les connaissances antérieures tout en étant guidées par les nouvelles observations.

Facteur de Bayes utilisant Metropolis-Hastings (MH)

Le facteur de Bayes est utilisé pour comparer deux modèles statistiques M_0 et M_1 en termes de probabilité que chaque modèle soit le générateur des données observées. Le facteur de Bayes est défini comme le rapport des vraisemblances marginales des deux modèles :

$$B_{0/1} = \frac{P(D | M_0)}{P(D | M_1)}$$

où $P(D | M_0)$ est la vraisemblance marginale des données D sous le modèle M_0 , calculée comme :

$$P(D | M_0) = \int_{\theta_0} P(D | \theta_0, M_0)P(\theta_0 | M_0)d\theta_0$$

L'intégrale ci-dessus est souvent très complexe analytiquement, ce qui justifie l'utilisation de méthodes numériques telles que l'algorithme de Metropolis-Hastings. L'algorithme de Metropolis-Hastings (MH) est utilisé pour générer une chaîne de Markov dont les états sont distribués selon la densité postérieure $P(\theta | D, M)$. Une fois la chaîne construite, on peut utiliser les échantillons pour estimer la vraisemblance marginale.

Pour un modèle donné M , la densité postérieure non normalisée est donnée par :

$$P(\theta | D, M) \propto P(D | \theta, M)P(\theta | M)$$

En utilisant Metropolis-Hastings, on génère une chaîne $\theta_1, \theta_2, \dots, \theta_n$ qui suit cette distribution postérieure.

La vraisemblance marginale peut ensuite être estimée en prenant la moyenne des densités postérieures non normalisées évaluées aux échantillons générés par MH :

$$P(D | M) \approx \frac{1}{N} \sum_{i=1}^N \frac{P(D | \theta_i, M)P(\theta_i | M)}{q(\theta_i)}$$

où $q(\theta_i)$ est la distribution de proposition utilisée dans MH.

Le facteur de Bayes entre deux modèles M_0 et M_1 est alors estimé comme le ratio des vraisemblances marginales :

$$B_{0/1} \approx \frac{\frac{1}{N_0} \sum_{i=1}^{N_0} \frac{P(D|\theta_{0,i}, M_0)P(\theta_{0,i}|M_0)}{q(\theta_{0,i})}}{\frac{1}{N_1} \sum_{i=1}^{N_1} \frac{P(D|\theta_{1,i}, M_1)P(\theta_{1,i}|M_1)}{q(\theta_{1,i})}}$$

Cela donne une mesure de la force de l'évidence en faveur de l'un des modèles par rapport à l'autre. Une approche simplifiée pour le calcul du facteur de Bayes, basée sur la LogVraisemblance des modèles, a été réalisée afin d'approximer la vraisemblance marginale lorsque les prioris sont plats ou uniformes. Cette méthode revient à comparer les vraisemblances exponentielles des deux modèles en utilisant la relation suivante :

$$B_{0/1} \approx \exp(\ln L(\hat{\theta}_0) - \ln L(\hat{\theta}_1))$$

où $\ln L(\hat{\theta}_0)$ et $\ln L(\hat{\theta}_1)$ représentent les log-Vraisemblances des modèles comparés.

Ce choix simplifie le calcul des vraisemblances marginales, car les intégrations sur les prioris se réduisent à des constantes, qui s'annulent lors du calcul des facteurs de Bayes. Ainsi, l'évaluation relative des modèles devient plus directe et moins constitutionnellement intensive.

Par ailleurs, l'Autocorrelation Function (ACF) étant une mesure statistique qui décrit la corrélation entre les valeurs d'une série temporelle à différents temps décalés, elle a été utilisée pour évaluer l'autocorrélation entre les échantillons successifs de la chaîne. Où, l'ACF pour un décalage (lag) k est définie comme :

$$\text{ACF}(k) = \frac{\text{Cov}(X_t, X_{t+k})}{\text{Var}(X_t)}$$

où X_t est la valeur de la chaîne à l'itération t , $\text{Cov}(X_t, X_{t+k})$ est la covariance entre les valeurs à deux temps décalés par k , et $\text{Var}(X_t)$ est la variance de la chaîne. L'ACF a donc été utilisée pour diagnostiquer la qualité de l'échantillonnage effectué par l'algorithme de Metropolis-Hastings:

- **Faible autocorrélation:** Une faible autocorrélation indique que les échantillons successifs sont peu dépendants, ce qui est souhaitable pour obtenir une bonne exploration de l'espace des paramètres.
- **Autocorrélation élevée:** Si l'ACF montre une forte autocorrélation pour des décalages k faibles, cela signifie que la chaîne met plus de temps à explorer l'espace des paramètres, ce qui peut nécessiter un échantillonnage plus long ou un ajustement de la proposition dans l'algorithme.

Un diagnostic courant est de tracer l'ACF pour différentes valeurs de k afin de vérifier que la chaîne converge rapidement vers zéro. Si ce n'est pas le cas, cela peut indiquer que les échantillons sont trop corrélés, et des modifications de la stratégie d'échantillonnage peuvent être nécessaires.

5.5.1 Vérification des Modèles

Après l'application des modèles statistiques, il est nécessaire de vérifier que les modèles sélectionnés sont correctement ajustés aux données. Cette vérification peut inclure des tests de résidus et des vérifications des hypothèses du modèle. Les histogrammes des résidus, les

graphes de normalité, et les tests de conformité des hypothèses sous-jacentes sont autant d'outils pour s'assurer que le modèle choisi est non seulement statistiquement justifié, mais aussi biologiquement pertinent. Si un modèle ne s'ajuste pas correctement, il peut être nécessaire de le réajuster ou d'explorer des alternatives.

En suivant ces étapes, les données préparées sont prêtes pour une analyse statistique robuste, ce qui permet d'obtenir des résultats fiables et scientifiquement valides pour l'évaluation de la toxicité. Cette rigueur dans la préparation des données est essentielle pour garantir que les conclusions tirées des études soient valides et applicables dans un contexte réglementaire.

5.6 Détermination des valeurs de toxicité

Les valeurs de toxicité, telles que la concentration à laquelle un certain pourcentage d'organismes montre une réponse (par exemple, EC_{50}), sont cruciales pour évaluer les effets des substances chimiques sur les organismes vivants. Ces valeurs peuvent être déterminées à partir de différents types d'essais d'écotoxicité, notamment les essais limite et les essais dose-réponse. **Les essais limite** permettent de déterminer si une substance a un effet observable sur les organismes à une concentration donnée. Si aucun effet n'est observé à cette concentration, la substance est souvent considérée comme non toxique à cette concentration. Par contre, **les essais dose-réponse** fournissent des informations plus détaillées, en permettant de tracer une courbe dose-réponse à partir de laquelle on peut déterminer des valeurs clés telles que la concentration efficace médiane (EC_{50}), la concentration létale médiane (LC_{50}), ainsi que d'autres paramètres critiques.

Outre l' EC_{50} et la LC_{50} , d'autres valeurs de toxicité sont fréquemment utilisées pour une évaluation plus fine :

- **EC_{10} et EC_{20} cf figure 5.1:** Ces valeurs représentent les concentrations à lesquelles respectivement 10% et 20% des individus d'une population exposée montrent une réponse spécifique. Ces valeurs sont particulièrement utiles dans les évaluations de toxicité chronique, où des **effets sublétaux** à faibles concentrations sont étudiés. L' EC_{10} , en particulier, est souvent utilisé pour déterminer la PNEC (Predicted No Effect Concentration), en raison de son rôle dans l'évaluation des effets à faible niveau.
- **NOEC (No Observed Effect Concentration):** C'est la plus haute concentration à laquelle aucun effet nocif n'est observé. Cette valeur est cruciale pour identifier des seuils de toxicité chronique.
- **LOEC (Lowest Observed Effect Concentration):** Il s'agit de la plus faible concentration à laquelle un effet nocif est observé. Elle est souvent utilisée en complément du NOEC pour caractériser les effets sublétaux ou chroniques.
- **PNEC (Predicted No Effect Concentration):** C'est la concentration en dessous de laquelle on s'attend à ce qu'il n'y ait aucun effet adverse sur l'environnement. Cette valeur est utilisée pour évaluer les risques potentiels des substances chimiques pour les écosystèmes aquatiques, terrestres, et autres.

Ces valeurs sont essentielles dans la démarche d'évaluation des risques écotoxicologiques. Par exemple, l' EC_{10} et l' EC_{20} sont fréquemment utilisés pour caractériser les effets à long terme de faibles expositions, tandis que la PNEC est souvent comparée aux concentrations environnementales prédictives (PEC) pour évaluer le risque posé par une substance dans un environnement donné. Les essais et les valeurs obtenues sont utilisés pour établir des **valeurs de référence écotoxicologiques** qui permettent de caractériser le risque environnemental des substances chimiques. Dans ce rapport, les ECx ont été déterminées à partir des modèles

de régression non linéaire employés. Ces modèles permettent d'estimer avec précision la concentration à laquelle un certain pourcentage de la population répond à la substance testée, ce qui est essentiel pour évaluer les risques écotoxicologiques associés au contaminant étudié.

5.7 Analyse de la significativité des différences observées dans les ECx par la méthode de permutation [23]

La théorie de la permutation a été intégrée dans cette méthodologie, renforçant ainsi l'approche non paramétrique employée.

C'est une méthode statistique **non paramétrique** permettant de tester des hypothèses en comparant la statistique observée à une distribution obtenue par **permutation aléatoire** des données sous l'hypothèse nulle. Concrètement, cette méthode consiste à définir une hypothèse nulle (par exemple, H_0 : "les deux groupes sont indépendants") et à permuter aléatoirement les étiquettes des groupes dans les données. Pour chaque permutation, une statistique de test T^* est calculée, comme le **coefficent de corrélation de Pearson** ou le **tau de Kendall**, afin de construire une distribution empirique \hat{F}_{T^*} sous H_0 .

La p-valeur est ensuite obtenue en comparant la statistique de test observée T_{obs} à cette distribution :

p-valeur = $P(T^* \geq T_{\text{obs}})$, ce qui permet de décider si l'hypothèse nulle doit être rejetée ou non. Les tests par permutation offrent une flexibilité importante, car ils ne nécessitent pas d'hypothèses strictes sur la distribution des données, ce qui les rend particulièrement utiles lorsque les hypothèses des tests paramétriques, telles que **la normalité** ou **homoscédasticité**, ne sont pas vérifiées. Cette approche permet également un contrôle rigoureux du **taux d'erreur de type I**, ce qui est essentiel dans le cas de petits échantillons ou lorsque la distribution des données est incertaine.

5.8 Intervalles de confiance des ECx par Bootstrap

L'approche par **bootstrap**[10] est une méthode statistique particulièrement puissante en écotoxicologie, permettant d'estimer la variabilité des paramètres tels que les ECx (par exemple, EC_{10} , EC_{20} , EC_{50}) en **rééchantillonnant** plusieurs fois les données observées. Cette technique de rééchantillonnage est cruciale lorsque les **hypothèses de normalité des résidus ou homoscédasticité sont remises en question**, ou lorsque les **distributions des erreurs sont inconnues ou non normales**, ce qui est souvent le cas dans les modèles non linéaires. Le bootstrap permet ainsi d'estimer la distribution empirique des paramètres sans faire d'hypothèses strictes sur la distribution des erreurs, rendant cette méthode particulièrement robuste face aux valeurs aberrantes et aux distributions de données non standards. En rééchantillonnant **avec remise** à partir de l'échantillon initial, on peut générer un grand nombre d'échantillons bootstrapés, chacun permettant d'ajuster un modèle dose-réponse pour estimer les ECx correspondants. Ces estimations forment ensuite une distribution empirique des ECx, qui peut être utilisée pour calculer des intervalles de confiance, souvent définis par les percentiles de cette distribution bootstrap.

La forme générale d'un intervalle de confiance bootstrap pour un estimateur \hat{EC}_x est donc donnée par :

$$[\hat{EC}_{x_{\alpha/2}}^*, \hat{EC}_{x_{1-\alpha/2}}^*],$$

où $\hat{EC}_{x_{\alpha/2}}^*$ et $\hat{EC}_{x_{1-\alpha/2}}^*$ sont les percentiles $\alpha/2$ et $1 - \alpha/2$ de la distribution bootstrap des \hat{EC}_x^* . De plus, le bootstrap permet également d'estimer le biais et la variance de l'estimateur, fournissant ainsi des informations précieuses sur la précision et la stabilité des estimations obtenues. Un aspect important du bootstrap est l'estimation par **injection**, où l'idée est de remplacer la loi inconnue de la population par sa fonction de répartition empirique, notée \hat{F}_n , qui est construite à partir des données observées. En pratique, les échantillons bootstrap sont générés en tirant avec remise selon \hat{F}_n , ce qui assure que les nouvelles variables X_i^* , conditionnées à l'échantillon initial, suivent une loi proche de la loi réelle. Cette approche permet de simuler la distribution de l'estimateur sans recourir à des hypothèses paramétriques rigides, offrant ainsi une méthode flexible pour évaluer la fiabilité des estimations dans des contextes complexes. En écotoxicologie, cette technique est souvent utilisée pour estimer les propriétés des ECx, tels que le biais, la variance, et les intervalles de confiance, en utilisant des échantillons rééchantillonnés pour simuler la distribution de l'estimateur. Ainsi, le bootstrap constitue un outil essentiel pour réaliser des inférences statistiques robustes là où les méthodes traditionnelles pourraient être inappropriées ou insuffisantes.

Une autre méthode non paramétrique, tout aussi puissante mais appliquée dans un contexte différent, est l'approche par permutation. Alors que le bootstrap se concentre sur l'estimation des intervalles de confiance et des biais en s'appuyant sur des rééchantillonnages avec remplacement, l'approche par permutation est particulièrement judicieuse pour tester des hypothèses en réarrangeant aléatoirement les données, offrant ainsi une solution robuste et flexible pour évaluer la significativité des résultats sans supposer de distribution spécifique

5.9 Limites et contraintes liées aux données

L'exploitation des données écotoxicologiques comporte plusieurs limites et contraintes. Ces limitations incluent la variabilité biologique, les incertitudes expérimentales et les défis liés à l'extrapolation des résultats de laboratoire aux conditions environnementales réelles. De plus, certaines données peuvent être incomplètes ou présenter des biais, ce qui nécessite des adaptations méthodologiques pour assurer la fiabilité des conclusions tirées. Certains documents guides de tests réglementaires (Essai 201[27] de toxicité aiguë pour les algues, l'Essai 202 citeTest Guideline No 202 de toxicité aiguë pour les *Daphnia magna*, Essai 210[32] de toxicité prolongée pour les poissons et l'Essai 211[33] de reproduction chez les *Daphnia magna*) exigent la détermination d'une NOEC pour une ou plusieurs réponses, il est donc essentiel de comprendre les modèles statistiques disponibles pour cet objectif. Cela implique une compréhension des conceptions expérimentales qui soutiennent ces tests, ainsi que des propriétés telles que la **puissance, les sensibilités et les limitations** qui déterminent leur utilité.

Il existe plusieurs tests appropriés pour l'analyse des réponses continues, quantales, discrètes et ordinaires provenant d'études écotoxicologiques. Pour la détermination des NOEC, ils sont divisés en deux catégories : les tests paramétriques et non paramétriques. Tous ces tests reposent sur des modèles ayant des critères spécifiques, et l'utilisation de données qui ne les respectent pas peut conduire à des conclusions erronées. Il est donc important d'explorer des outils formels et informels pour déterminer si un ensemble de données répond aux exigences de normalité et d'homogénéité de variance, et s'il est compatible avec une relation dose-réponse monotone (*i.e* une relation où la réponse change systématiquement dans une seule direction à mesure que la concentration augmente ou diminue). Par ailleurs, il est important de rappeler que l'unité de base d'analyse est le **réplicat**, pas l'animal ou la

plante individuelle, et que les organismes au sein de chaque réplicat sont interdépendants.

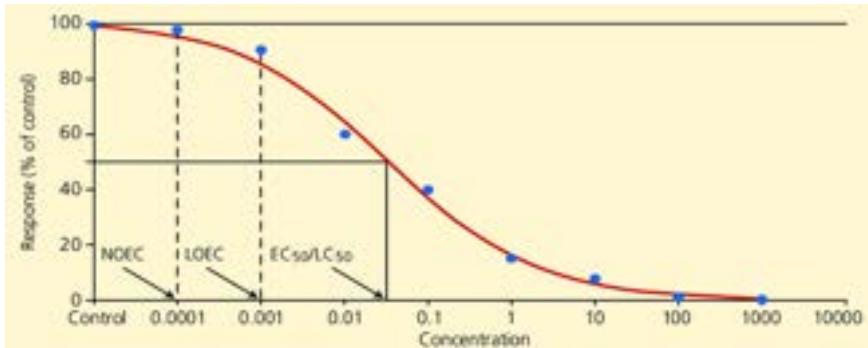


Figure 5.1: Illustration NOEC, LOEC, ECx sur courbe dose réponse [21]

6 Résultats et discussion

Cette étude s'appuie sur l'essai 201 de l'OCDE, qui établit des lignes directrices pour l'évaluation de la toxicité aiguë des substances chimiques sur les algues et les cyanobactéries aquatiques. L'essai 201 est une méthode standardisée utilisée pour déterminer l'impact de différentes concentrations de substances chimiques sur la croissance et la santé des populations algales sur une période de 72 heures. L'objectif principal de cet essai est de déterminer la concentration efficace médiane (EC_{50}), c'est-à-dire la concentration à laquelle la substance testée inhibe la croissance des algues de 50%.

6.1 Codage et Transformation des Données :

Pour harmoniser les données avec l'application Rshiny, les données nettoyées doivent souvent être transformées ou codées. Cela implique de mettre les données en forme pour les adapter aux exigences des analyses statistiques prévues par type de test d'écotoxicité. Par exemple, les catégories de concentration ou de traitement peuvent être codifiées en valeurs numériques pour faciliter l'application des tests statistiques.

La transformation des données en **format long** est essentielle pour simplifier l'analyse des données écotoxicologiques, facilitant ainsi l'analyse statistique, la visualisation, et la gestion des réplicats expérimentaux. Ce format améliore la compatibilité interne avec l'application, permet l'ajout de nouvelles variables et simplifie la comparaison entre différents groupes ou traitements. **Exemple de format long pour le Test OCDE 201 :**

```
Replicat,concentration,response
Replicat1,Control,61.2
Replicat1,Control_Solvent,63.3
Replicat1,0.1,58.2
Replicat1,0.15,48.6
...
```

Dans ce format long, chaque ligne correspond à une observation unique pour un réplicat donné, avec les colonnes représentant le réplicat, la concentration testée, et la réponse mesurée.

Pour la **Ligne directrice OCDE 201** qui concerne l'inhibition de la croissance des algues et des cyanobactéries. Les données doivent être structurées pour permettre une comparaison

claire entre les groupes témoins et les groupes traités, en codifiant par exemple les concentrations de l'agent testée en valeurs numériques.

Et le format d'entrée attendu par l'application est :

```
Replicat,Control,Control_Solvent,0.1,0.15,0.23,0.34,0.51,0.76,1.14
```

```
Replicat1,61.2,63.3,58.2,48.6,48.5,44.1,25.5,2.5,1.2
```

```
Replicat2,63.1,62.3,60.8,47.5,47.9,46.4,24.3,2.3,1.3
```

```
...
```

Pour la **Ligne directrice OCDE 202** qui concerne la toxicité aiguë sur Daphnia, les données de mortalité sont souvent présentées sous forme de pourcentages. Ces données doivent être organisées de manière à pouvoir être facilement analysées pour déterminer la concentration létale médiane (CL_{50}).

Et le format d'entrée attendu par l'application est :

```
Replicat,time,concentration,total,immobilized
```

```
1,24,0,20,0
```

```
2,24,5,20,0
```

```
...
```

Lorsqu'on introduit un test qui n'est pas répertorié dans les essais standards de l'application Rshiny, il est nécessaire de mettre en forme le jeu de données sous le format de données compatible, comme le format CSV ou TXT. L'application fournit un exemple du format attendu pour guider l'utilisateur dans la mise en forme des données. Il est important de suivre ces formats pour assurer la compatibilité et la précision des analyses.

Pour un nouveau test, il faut structurer les données brutes en CSV ou txt comme suit :

```
Replicat,concentration,response
```

```
Replicat1,0.1,58.2
```

```
Replicat1,0.15,48.6
```

```
...
```

6.2 Préparation des données pour l'analyse

La préparation des données est une étape cruciale pour assurer la validité des analyses écotoxicologiques. Conformément aux directives de l'OCDE 54, les données collectées ont été préparées minutieusement avant toute analyse statistique (cf méthodologie). Pour illustrer ce processus, nous allons prendre l'exemple d'une étude sur l'effet toxique d'un pesticide sur une espèce d'algues d'eau douce (Essai 201).

6.2.1 Visualisation

Dans l'analyse des données **dose-réponse**, une première étape cruciale consiste à inspecter visuellement les données. Cette étape ne s'applique pas directement aux **essais limite**, qui visent uniquement à déterminer la toxicité à une concentration spécifique. Pour les données **continues**, il est utile de tracer les réponses individuelles ainsi que les moyennes de groupe en fonction de la concentration. Pour les données **quantales**, les fréquences observées de réponse sont tracées en fonction de la dose.

Ces graphiques permettent d'évaluer si les données montrent une relation dose-réponse cohérente et peuvent révéler des valeurs aberrantes, c'est-à-dire des points de données qui s'écartent significativement des attentes ou du schéma général observé.

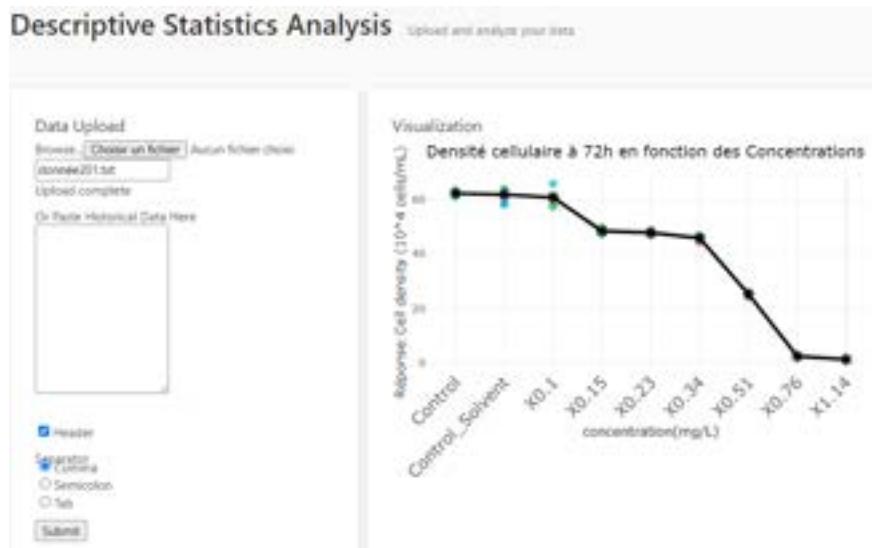


Figure 6.1: Diagramme Essai 201

La Figure 6.1 présente, à travers l'application développée, un diagramme de dispersion illustrant les données de densité cellulaire dans une expérience d'inhibition de la croissance chez les algues (Essai OCDE 201). Les réponses moyennes aux différents traitements sont représentées par des étoiles et reliées par des segments de ligne, ce qui permet de visualiser les tendances entre les traitements et les réponses. Ce type de graphique permet de comparer visuellement la dispersion des données au sein des traitements et de rechercher des preuves de variance inégale. Les moyennes reliées facilitent l'évaluation de la cohérence d'une réponse monotone en fonction de la concentration, ou, à l'inverse, révèlent des réponses non monotones, indiquant des variations plus complexes en fonction de la dose. Par exemple, dans la figure présentée, bien que l'on observe de légers pics aux concentrations de 0,1 et 0,34 mg/L, la tendance générale reste interprétable comme une réponse monotone à la dose, car les fluctuations sont mineures par rapport à la tendance globale. Cependant, si la réponse n'était pas monotone, ce type de visualisation permettrait de détecter ces irrégularités. Cette visualisation initiale permet d'identifier préalablement les tendances et les anomalies potentielles, telles que des valeurs aberrantes qui pourraient biaiser l'analyse ultérieure. La finalité de cette inspection est d'informer une analyse descriptive, qui synthétise les données avant de passer aux phases d'analyse et de traitement des données. Le diagramme de dispersion peut également être utile pour identifier ces valeurs, en montrant des déviations notables par rapport aux tendances attendues, qu'elles soient monotones ou non.

La Figure 6.2 illustre la dispersion des données de l'essai dose-réponse n°201 à l'aide de boxplots pour chaque concentration. Ces boxplots permettent une analyse visuelle de la distribution des résultats, en mettant en évidence la dispersion des données autour des médianes et en identifiant les valeurs aberrantes. Les points situés en dehors des moustaches des boxplots indiquent des outliers potentiels, et cette visualisation offre également la possibilité de détecter un éventuel effet solvant.

6.2.2 Gestion des outliers

À la lumière des observations précédentes, les données ont été vérifiées afin de garantir leur adéquation aux analyses statistiques prévues. Les données initiales ont effectivement révélé la présence d'outliers (voir tableau 1), notamment pour certaines concentrations spécifiques. Mais aucune valeur aberrante identifiée n'a été exclue de l'analyse à ce stade, car aucune

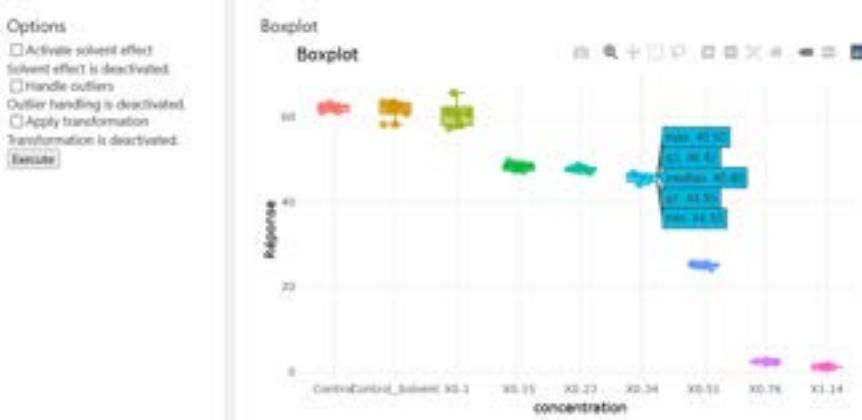


Figure 6.2: Boxplot Essai 201

Concentrations	Outliers
0	58.2
0.1	65.6
0.51	24.3
1.14	1.3

Table 2: Outliers détectés par la méthode de Tukey

erreur expérimentale détectée ne le justifiait. Ce processus vise à maintenir l'intégrité statistique des données tout en assurant que les résultats reflètent fidèlement les effets réels des concentrations testées. Après la visualisation, les données sont donc traitées pour garantir leur adéquation aux analyses statistiques prévues. Le nettoyage des données est une étape cruciale pour assurer leur qualité et fiabilité, en particulier en ce qui concerne la gestion des valeurs aberrantes et des incohérences. Les valeurs aberrantes, définies comme des observations qui deviennent notablement des autres dans des conditions similaires (c'est-à-dire qu'elles semblent se situer en dehors de la dispersion habituelle des données au sein d'un groupe de traitement) ([4]), peuvent biaiser les résultats en augmentant la probabilité de faux positifs et faux négatifs ([5]). Ces valeurs sont particulièrement indésirables dans les groupes de concentration car elles peuvent altérer les résultats de manière significative. Pour confirmer statistiquement une valeur aberrante, des tests tels que le test de Tukey et le test de Grubbs peuvent être appliqués. Cependant, pour leur suppression, une justification biologique est nécessaire [24] [25] [26]. Deux cas se présentent alors :

- **Si la valeur aberrante résulte d'une déviation expérimentale naturelle ou inexplicable**, et que la taille de l'échantillon est petite, il est recommandé de conserver la donnée conformément aux recommandations de l'OCDE 54.
- **Si la valeur aberrante peut être attribuée à une erreur expérimentale identifiable**, elle peut être exclue de l'analyse pour éviter de fausser les résultats, en accord avec l'OCDE 54.

Les valeurs aberrantes dans les données **continues** peuvent biaiser les estimations des effets et augmenter la variance résiduelle, ce qui réduit la puissance des tests statistiques et la précision des paramètres estimés. Par conséquent, si des raisons peuvent être trouvées pour

expliquer les valeurs aberrantes, il est préférable de les supprimer de l'analyse.

Dans les données **quantales**, elles affectent généralement un groupe de traitement entier, ce qui peut indiquer des erreurs systémiques dans la mise en place de l'expérience. Elles peuvent fausser les résultats en entraînant des estimations des effets soit trop faibles, soit trop élevées, augmentant ainsi la variance résiduelle ou le bruit. Cela réduit la puissance des tests statistiques et la précision des paramètres estimés. Les valeurs aberrantes affectent donc davantage l'analyse des données quantales que celle des données continues, car dans les données quantales, les valeurs aberrantes concernent souvent un groupe de traitement entier, puisque la réponse déviante individuelle ne peut être détectée sur une base binaire (oui/non).

Dans les données de **comptage**, les valeurs aberrantes peuvent également avoir un impact significatif, en particulier lorsqu'elles sont présentes en faible nombre. Ces données sont souvent analysées à l'aide de modèles de régression pour données de comptage, tels que les modèles de Poisson ou binomiaux négatifs. Une valeur aberrante dans un jeu de données de comptage peut entraîner des estimations biaisées des paramètres du modèle, ce qui pourrait nécessiter des techniques de correction ou de robustesse, comme l'utilisation de modèles qui tiennent compte de la surdispersion ou des poids ajustés pour réduire l'influence des valeurs extrêmes.

Par ailleurs, les observations influentes (c'est-à-dire une valeur inhabituellement grande ou petite d'une variable explicative), qui peuvent avoir une grande influence sur une analyse, doivent également être prises en compte.

6.2.3 Agrégation des Données :

L'étape suivante a consisté à vérifier l'effet d'un solvant lorsqu'il a été nécessaire d'en employer un pour le test d'écotoxicité. Des tests statistiques ont donc été appliqués pour comparer le groupe témoin sans solvant (contrôle) avec le groupe témoin avec solvant (contrôle + solvant). Pour illustrer cette analyse, nous avons utilisé le test de Wilcoxon (non paramétrique) afin de comparer les médianes de deux groupes distincts. Les résultats de ce test indiquent que l'effet du solvant est statistiquement non significatif. De manière similaire, le test t de Student (paramétrique) a été employé pour comparer les moyennes des mêmes groupes. Les résultats obtenus montrent également qu'il n'y a pas de différence statistiquement significative entre le groupe témoin et le groupe témoin avec solvant. Ces analyses confirment ainsi les observations faites lors de la visualisation des données, à savoir que le solvant utilisé dans cette étude n'a pas d'effet sur la densité cellulaire des algues. Par conséquent, les colonnes "contrôle" et "contrôle + solvant" ont été fusionnées en une seule colonne intitulée "0", qui sera utilisée pour l'interprétation des résultats finaux. Cette approche vise à renforcer la robustesse et la cohérence de l'analyse, tout en évitant la complexité inutile liée à la gestion séparée des données avec et sans solvant.

Pour vérifier en pratique l'existence de cet effet solvant (*i.e* que les deux contrôles sont différents), il est nécessaire de comparer les données issues du groupe témoin à celles du groupe témoin solvant. Cette comparaison peut être réalisée visuellement, afin d'identifier rapidement des différences potentielles dans la distribution des données entre les deux groupes, souvent à l'aide de boxplots, ainsi que statistiquement à l'aide de tests appropriés.

Lorsque les données suivent une distribution normale et que les variances des deux groupes sont égales (homogénéité des variances), un test t de Student (pour échantillons indépendants) est utilisé pour comparer les moyennes des deux groupes. La statistique de test, notée t , est calculée à partir des différences entre les moyennes des groupes, la variance de

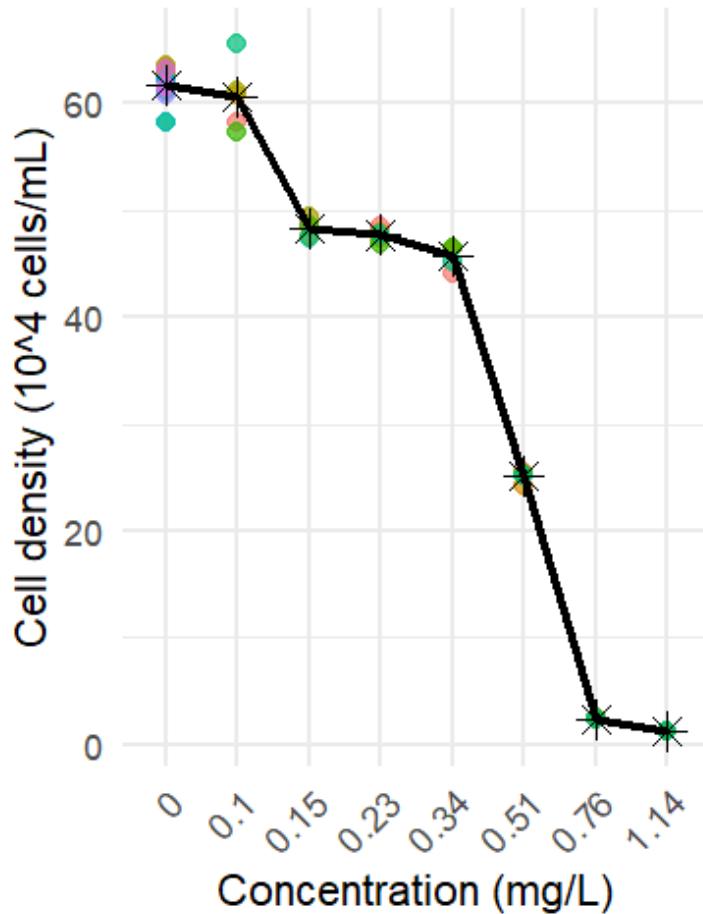


Figure 6.3: Densité cellulaire à 72h en fonction des concentrations : concaténation

chaque groupe, et la taille de chaque échantillon. L'hypothèse nulle (H_0) de ce test est que les moyennes des groupes contrôle et contrôle-solvant sont égales.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

où \bar{X}_i est la moyenne du groupe i , s_i^2 est la variance, et n_i est la taille de l'échantillon.

Et si les données ne suivent pas une distribution normale, un test non paramétrique, tel que le test de Wilcoxon (pour deux échantillons indépendants), est utilisé. Ce test repose sur les rangs des observations et compare les distributions des deux groupes sans faire d'hypothèses sur la forme de la distribution. L'hypothèse nulle (H_0) de ce test est que les distributions des deux groupes sont identiques. Le Test de Mann-Whitney U, est une autre alternative non paramétrique au test t de Student. Il est utilisé lorsque les données ne suivent pas une distribution normale et teste si les deux groupes proviennent de la même distribution. La statistique de test U est définie par :

$$U = \min(U_1, U_2)$$

où

$$U_1 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1$$

et

$$U_2 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - R_2$$

avec n_1 et n_2 représentant les tailles des deux échantillons, et R_1 et R_2 les sommes des rangs pour les groupes respectifs.

Pour chacun de ces tests, si la valeur p est supérieure à 0,05, l'hypothèse nulle (H_0) ne sera pas rejetée, ce qui indique qu'il n'y a pas de différence significative entre le contrôle et le contrôle-solvant. Dans ce cas, les données des deux contrôles peuvent être regroupées, en les concaténant sous une même colonne '0'. Si la valeur p est inférieure à 0,05, l'hypothèse nulle (H_0) est rejetée, ce qui indique une différence significative, et il est alors nécessaire de déterminer quel groupe contrôle utiliser pour l'analyse ultérieure.

6.3 Analyse des données

6.3.1 Vérification de la normalité et de l'homoscédasticité des données :

Concentration	p-value
0	0.008540683
0.1	0.5391251
0.15	0.5995957
0.23	0.940293
0.34	0.08510793
0.51	0.05367589
0.76	0.006470001
1.14	0.0001309782

Table 3: Résultats du test de Shapiro-Wilk pour chaque concentration

Test	Statistique	p-value
Bartlett	K-squared = 56.609, df = 7	7.148e-10
Levene	F value = 2.6038, df = 7, 42	0.02514

Table 4: Vérification Homoscédasticité

Les hypothèses statistiques des tests appliqués ont été rigoureusement vérifiées. Comme indiqué à l'étape précédente, aucune différence n'a été observée entre les deux groupes de contrôle. Par conséquent, les données concaténées ont été utilisées pour tester les hypothèses d'homogénéité des variances et de normalité à l'aide des tests de Shapiro-Wilk et de Levene. Les résultats ont révélé des violations de ces hypothèses pour certaines concentrations (voir les tableaux 2 et 3, ainsi que la figure 6.4), ce qui justifie l'utilisation ultérieure de méthodes d'inférence statistique non paramétriques.

Les tests de normalité effectués sur les différentes concentrations ont révélé que, bien que l'effet solvant ait été confirmé comme négligeable, les données ne suivent pas une distribution

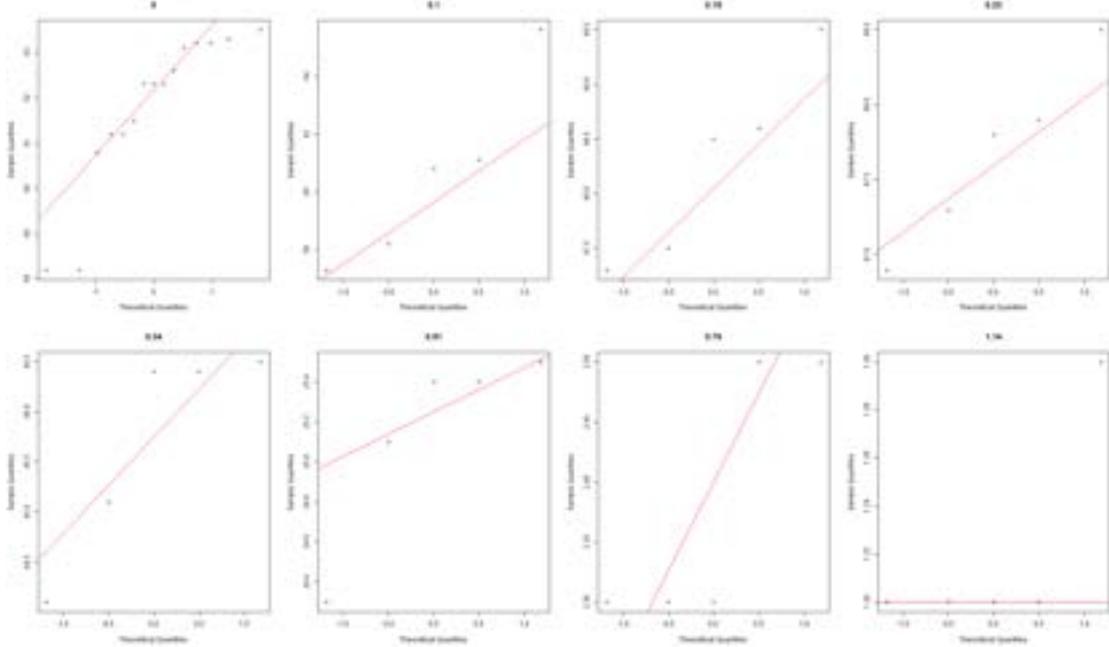


Figure 6.4: QQplot par concentration

normale pour certaines concentrations (cf figure 6.4). Par conséquent, il sera nécessaire d'appliquer des méthodes statistiques non paramétriques dans les analyses ultérieures

En effet, tous les tests et procédures statistiques reposent sur des hypothèses concernant la nature des données à analyser. Parmi celles-ci, deux hypothèses clés sont généralement considérées comme très importantes pour les données continues[3]:

- La distribution des données au sein et entre les groupes de concentration.
Il est important de déterminer si les données de chaque groupe de concentration sont cohérentes avec une distribution normale.
- Et, le cas échéant, si les données des différents groupes de concentration respectent l'hypothèse d'homoscédasticité (entre les concentrations), une hypothèse qui peut être examinée dès la visualisation des boxplots (cf étape de visualisation).

En revanche, pour les données quantales ou de comptage, l'accent est mis sur leur conformité avec une distribution binomiale ou de Poisson et sur la détection éventuelle de surdispersion (c'est-à-dire une variance supérieure à celle attendue). Cependant, étant donné que les tests d'écotoxicité se font souvent sur de petits échantillons, il est préférable d'évaluer la normalité des résidus à l'échelle de l'ensemble de l'étude. Le test de Shapiro-Wilk est couramment utilisé à cette fin, et une transformation logarithmique des données peut être recommandée pour les petits échantillons.[2] L'homoscédasticité est vérifiée à l'aide des tests de Bartlett et de Levene, où une p-value supérieure à 0,05 indique que les variances sont homogènes. Des outils visuels, tels que les histogrammes, les Q-Q plots (cf. figure 6.5) et les scatterplots (nuages de points) (cf. figure 6.6), sont également utilisés pour évaluer la distribution des données continues. Ces outils permettent d'examiner la normalité et l'homoscédasticité, aidant ainsi à déterminer si la dispersion est symétrique et homogène, et à mieux comprendre la distribution des variances:

Pour pallier l'absence de normalité et d'homoscédasticité, la transformation temporaire des données sur une échelle logarithmique (*i.e* $y'_i = \log(y_i)$) est une approche couramment

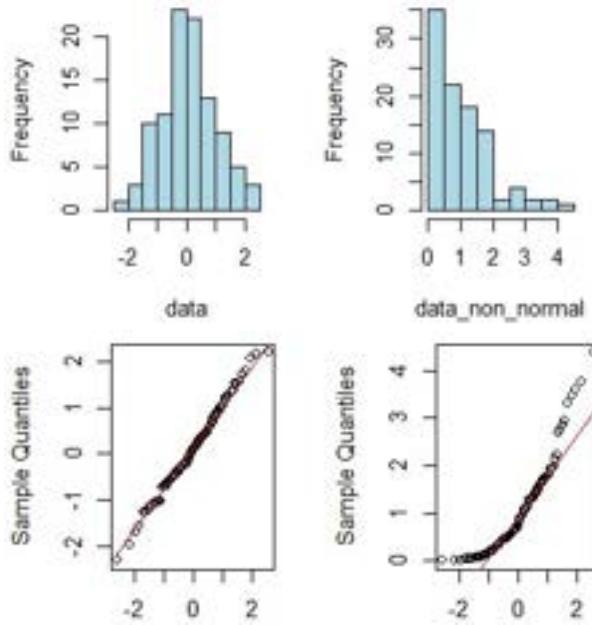


Figure 6.5: normale *vs* non normale

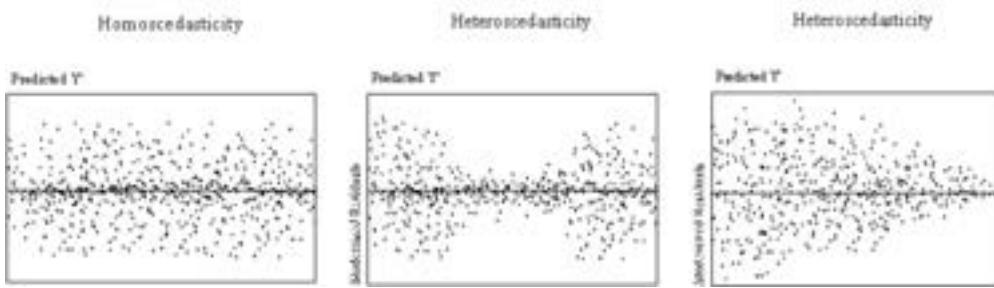


Figure 6.6: scatterplots homoscédasticité vs hétéroscédasticité

utilisée. Cette transformation est particulièrement pertinente lorsque les données présentent une asymétrie marquée ou lorsque la variance des données augmente avec la moyenne (hétéroscédasticité). Car, elle permet d'atténuer ces problèmes en stabilisant la variance et en rapprochant la distribution des données d'une distribution normale[3].

En utilisant par exemple, une transformation Box-Cox définie par l'équation suivante pour les données $y > 0$:

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{si } \lambda \neq 0 \\ \log(y) & \text{si } \lambda = 0 \end{cases}$$

Cette transformation permet d'optimiser la normalisation des données en ajustant le paramètre λ , ce qui peut être particulièrement utile lorsque les écarts par rapport à la normalité et à l'homoscédasticité sont complexes

Pour les **proportions** (données exprimées sous forme de pourcentage ou de fraction d'un total), la distribution sous-jacente est généralement binomiale. Une transformation racine carrée arsinus ($\text{arcsin}(\sqrt{p})$) est souvent utilisée pour stabiliser la variance, en particulier lorsque les proportions sont très proches de 0 ou de 1, ce qui peut entraîner une variance non constante[3]. Cela permet de réduire la corrélation entre la moyenne et la variance, rendant les données plus adaptées à l'analyse statistique. Pour les données de comptage (comme par exemple le nombre d'événements ou d'occurrences dans un échantillon), qui suivent souvent une distribution de Poisson, une transformation par la racine carrée peut

être utilisée pour éliminer la relation entre la variance et la moyenne, améliorant ainsi la normalité des résidus et stabilisant la variance. De plus, lorsqu'une surdispersion (variance supérieure à la moyenne) est présente (peut être due à l'hétérogénéité des réponses ou à la présence d'un grand nombre de zéros dans les données), les modèles standard de Poisson peuvent ne pas convenir. Dans ce contexte, des modèles binomiaux négatifs, qui introduisent un paramètre supplémentaire pour modéliser la variance excédentaire, ou les modèles à effets mixtes, qui prennent en compte les variations aléatoires non observées, sont souvent utilisés pour gérer cette surdispersion[3]. Pour les données quantales, la vérification de la normalité et de l'homoscédasticité n'est généralement pas pertinente en raison de la nature binaire de ces données. En effet, les données quantales suivent une distribution binomiale plutôt qu'une distribution normale. Ainsi, plutôt que de chercher à transformer les données quantales pour vérifier la normalité ou l'homoscédasticité, il est plus pertinent d'utiliser directement des modèles et des méthodes statistiques conçus spécifiquement pour les données binaires. Cela permettra de tirer directement des conclusions fiables sur les probabilités d'occurrence des événements d'intérêt. Cependant, l'utilisation de transformations n'est pas sans risque et rend souvent les résultats plus difficiles à interpréter car elles modifient l'échelle d'origine des données, ce qui complique la compréhension des effets biologiques et peut rendre les comparaisons avec d'autres études plus difficiles. De plus, la complexité ajoutée par ces transformations peut conduire à des erreurs d'interprétation si elles ne sont pas bien comprises ou appliquées correctement.[3] Car, les moyennes et intervalles de confiance doivent être retranscrits à l'échelle originale pour faciliter l'interprétation. De plus, la pertinence de rester sous la forme transformée doit être évaluée. Lorsque les hypothèses de normalité et d'homoscédasticité ne sont pas respectées, des méthodes statistiques alternatives peuvent être envisagées. Parmi celles-ci, les **tests non-paramétriques** sont particulièrement utiles, car ils ne reposent pas sur des hypothèses strictes concernant la distribution des données. Contrairement aux tests paramétriques, ces tests n'exigent pas que les données suivent une distribution spécifique, ce qui les rend plus robustes en présence de déviations par rapport aux hypothèses classiques[3]. Ils sont également adaptés lorsque les échantillons sont petits comme c'est majoritairement le cas pour les essais d'écotoxicité[3].

6.3.2 Analyse descriptive

Concentration	Count	Means	Median	STD _ DEV	STD _ ERR
0	15	61.7933	62.3	1.6875	0.4357
0.1	5	60.6000	60.8	3.2381	1.4481
0.15	5	48.2800	48.5	0.8955	0.4005
0.23	5	47.6800	47.8	0.6099	0.2728
0.34	5	45.7000	46.4	1.0654	0.4765
0.51	5	25.1400	25.4	0.4930	0.2205
0.76	5	2.3800	2.3	0.1095	0.0490
1.14	5	1.2200	1.2	0.0447	0.0200

Table 5: Statistiques pour chaque concentration de substance testée (Test 201 OCDE)

Dans cet exemple d'étude, les algues ont été exposées à plusieurs concentrations du pes-

ticide, avec des réplicats pour chaque concentration. Pour chaque concentration, on peut comparer les réplicats pour vérifier la cohérence des résultats. Par exemple, si les réplicats montrent des taux de croissance similaires, cela indique une bonne répétabilité. Dans ce cas, la moyenne des taux de croissance, l'écart-type et la médiane sont fournis pour chaque concentration (cf table 4). Les résultats montrent une diminution progressive des moyennes et des médianes des taux de croissance des algues avec l'augmentation des concentrations du produit testé, ce qui est un indicateur clé d'une inhibition dose-dépendante. À partir de 0,51 mg/L, une chute drastique de la densité cellulaire indique que cette concentration atteint un seuil critique où l'effet devient significatif. Cette tendance suggère que cela pourrait avoir des implications écologiques importantes, étant donné le rôle fondamental des algues dans l'écosystème aquatique.

On constate aussi, une variabilité notable, notamment à la concentration de 0,1 mg/L, où un écart-type plus élevé a été observé. Cette variabilité suggère une réponse plus hétérogène des algues à cette concentration, potentiellement due à des observations influentes ou aberrantes (qui peuvent tirer la moyenne vers le haut ou vers le bas de manière disproportionnée, affectant l'interprétation globale). Toutefois, la suppression des outliers n'a pas été justifiée biologiquement, et ces valeurs ont été maintenues dans l'analyse. Les résultats confirment donc une inhibition marquée de la croissance algale avec l'augmentation de la concentration du produit, malgré une certaine variabilité dans les réponses, particulièrement à 0,1 mg/L.

6.4 Détermination des LOEC/D/L et NOEC/D/L

Les LOEC, NOEC, LOED (Lowest Observed Effect Dose), et NOED (No Observed Effect Dose) sont des valeurs de toxicité en écotoxicologie. Elles permettent de déterminer les concentrations à partir desquelles un effet est observé sur les organismes testés. Dans l'exemple traité, les données représentent les réponses en 10^4 cells/mL d'algues après exposition à plusieurs concentrations d'une substance chimique. Les moyennes et écarts-types pour chaque condition sont donnés dans la table 4. Des tests ANOVA ont été réalisés pour comparer chaque concentration testée par rapport au témoin (*i.e* à "0").

Concentration(mg/L)	F-value	p-value
0.1	0.728	0.418
0.15	165.59	1.26e-06
0.23	194.81	6.73e-07
0.34	220.46	4.17e-07
0.51	1311.48	3.70e-10
0.76	3582.96	6.74e-12
1.14	3730.99	5.74e-12

Table 6: Résultats des tests ANOVA pour déterminer l'effet de chaque concentration sur la croissance des algues (Test 201 OCDE)

Au vu des résultats présentés dans le tableau 6, la concentration **NOEC** est 0.1mg/L et la concentration **LOEC** est 0.15mg/L, ce qui signifie qu'à partir de 0.15mg/L, la substance commence à avoir un effet statistiquement significatif sur l'organisme étudié par rapport

au témoin, parce que la p-value est inférieure à 0.05, indiquant que la différence observée n'est probablement pas due au hasard. Les tests paramétriques peuvent être plus sensibles, mais lorsqu'ils sont appliqués sur des données non conformes aux hypothèses, ils peuvent induire en erreur. Les tests non-paramétriques, bien qu'ils soient moins puissants, sont plus robustes face aux violations d'hypothèses. Dans cette étude, bien que l'application de l'ANOVA ne soit pas pleinement justifiée par la distribution des données, ce test a été utilisé en raison des recommandations des tests OCDE, qui supposent généralement une distribution normale des données. Malgré cela, les résultats obtenus ont été cohérents, soulignant que les données biologiques peuvent parfois produire des résultats fiables même dans des conditions non idéales. Étant donné que les données ne respectaient pas les hypothèses de normalité et d'homogénéité des variances, des tests non-paramétriques ont été appliqués pour analyser les différences entre les groupes de concentrations testées. Le **test de Kruskal-Wallis** a été utilisé comme alternative non-paramétrique à l'ANOVA pour comparer les groupes. Ce test est particulièrement adapté aux données qui ne suivent pas une distribution normale. Les résultats du test de Kruskal-Wallis pour chaque concentration sont les suivants :

Concentration(mg/L)	K-Wallis Stat	p-value
0.1	1.098	0.295
0.15	6.818	0.009
0.23	6.818	0.009
0.34	6.860	0.009
0.51	6.860	0.009
0.76	7.031	0.008
1.14	7.258	0.007

Table 7: Résultats du test de Kruskal-Wallis pour les différentes concentrations (Test Non-Paramétrique)

Les résultats de l'analyse non-paramétrique ont conduit à des valeurs de NOEC et LOEC identiques à celles obtenues avec l'ANOVA. La **NOEC** est donc déterminée à **0.1 mg/L**. Cette concentration n'a pas montré de différence significative par rapport au contrôle ($p > 0.05$). La **LOEC** est déterminée à **0.15 mg/L**, qui est la concentration la plus faible à présenter un effet statistiquement significatif par rapport au contrôle ($p < 0.05$). Ces résultats fournissent une première conclusion essentielle pour les évaluateurs de risques : il est crucial de déterminer la valeur seuil à partir de laquelle l'effet devient significatif pour l'organisme testé. Cette détermination est fondamentale pour établir des seuils de sécurité environnementale et pour orienter les décisions réglementaires concernant l'utilisation des substances chimiques.

7 Calcul de Puissance

La puissance d'un test statistique est un élément fondamental pour évaluer la capacité d'une étude à détecter un effet réel, en particulier dans le contexte de la détermination d'une NOEC (concentration sans effet observé) ou d'autres résultats statistiquement significatifs.

La puissance se définit comme la probabilité de rejeter l'hypothèse nulle (H_0) lorsqu'elle est fausse, minimisant ainsi le risque d'erreurs de type II (faux négatifs). Une étude bien conçue doit avoir une puissance suffisante, généralement fixée à 80% en écotoxicologie, pour garantir que les différences biologiquement pertinentes ne passent pas inaperçues. L'objectif principal du calcul de puissance est d'évaluer la **capacité d'un test à détecter un effet significatif**, ce qui permet de s'assurer que l'étude est suffisamment sensible pour identifier des différences entre les groupes testés.

Ce calcul peut être effectué à deux moments clés :

- **Puissance a priori** : Utilisée pour déterminer la taille d'échantillon nécessaire avant l'expérience, afin de garantir une puissance adéquate. Cette approche est essentielle pour planifier des études qui soient suffisamment robustes dès le départ.
- **Puissance post hoc** : Évaluée après l'obtention des résultats pour estimer la sensibilité des tests réalisés. En évaluation du risque, notamment en écotoxicologie, la puissance post hoc est souvent la seule option disponible, car les tests sont généralement effectués après coup et non planifiés a priori. Cette contrainte impose des défis significatifs pour l'interprétation des résultats, car il est plus difficile de garantir rétrospectivement que l'étude avait une puissance suffisante pour détecter des effets écologiquement pertinents.

La puissance d'un test d'hypothèse dépend de plusieurs paramètres, dont la taille de l'effet, la taille de l'échantillon, et le niveau de signification (α). Par exemple, pour comparer la moyenne d'un groupe témoin avec celle d'un groupe exposé, l'hypothèse nulle (H_0) stipule que les moyennes sont égales ($\mu_0 = \mu_k$). La puissance du test est alors la probabilité de détecter une différence réelle ($H_a : \mu_0 > \mu_k$) si elle existe. Le calcul de la puissance peut être réalisé en utilisant les fonctions de distribution associées aux tests statistiques utilisés, tels que le test de Dunnett ou le test t de Student. Par exemple, pour détecter une diminution de 15% par rapport au contrôle, la puissance peut être visualisée via des courbes de puissance en fonction de la taille de l'échantillon et de la taille de l'effet attendu. Le calcul de puissance pour un test ANOVA se base sur la **taille de l'effet** (f), le nombre de groupes (k), la taille de l'échantillon par groupe (n), et le niveau de signification (α). La taille de l'effet est définie par Cohen comme :

$$f = \sqrt{\frac{\eta^2}{1 - \eta^2}}$$

où η^2 est la proportion de la variance expliquée par les groupes. La puissance est alors déterminée en utilisant la non-centralité λ , donnée par :

$$\lambda = n \times k \times f^2$$

La statistique de test suit une distribution F avec $k - 1$ et $N - k$ degrés de liberté, où N est le nombre total d'observations. La puissance est calculée comme :

$$\text{Puissance} = P(F > F_{\text{crit}}; \lambda, df_1, df_2)$$

où F_{crit} est la valeur critique de la distribution F pour le niveau de signification α , $df_1 = k - 1$, et $df_2 = N - k$.

Pour les données en écotoxicologie, cette analyse de puissance est particulièrement importante, car elle permet de s'assurer que les tests statistiques utilisés sont suffisamment sensibles pour détecter des effets écotoxicologiques significatifs, comme une diminution de la

survie, de la croissance, ou de la reproduction des organismes exposés à des substances chimiques. Si la puissance est faible, il est possible que même des effets biologiquement importants ne soient pas détectés, ce qui pourrait conduire à des conclusions erronées quant à la sécurité d'un produit chimique. Par exemple, en analysant la puissance pour un test ANOVA avec un effet attendu faible (f), si la puissance calculée est inférieure à 0,8, il serait recommandé d'augmenter la taille de l'échantillon ou de revoir le design expérimental. Cela pourrait signifier qu'il y a un risque élevé de ne pas détecter un effet significatif, ce qui est particulièrement critique en écotoxicologie, où la protection des écosystèmes repose sur des tests sensibles et fiables.

Une étude bien conçue doit donc avoir une puissance suffisante et l'objectif principal du calcul de puissance est de s'assurer que l'étude est suffisamment sensible pour identifier des différences entre les groupes testés. Cependant, lorsqu'une relation dose-réponse claire émerge, comme dans notre exemple de l'essai 201, une approche plus approfondie devient nécessaire. En effet, la simple identification des concentrations sans effet observable (NOEC) ou avec effet observable le plus bas (LOEC) peut ne pas suffire à caractériser pleinement l'impact d'un contaminant. Dans ces cas, la modélisation **dose-réponse** est indispensable pour quantifier plus précisément l'effet d'un contaminant en fonction de sa concentration.

8 Modélisation Dose-Réponse

L'analyse des effets d'un contaminant ou d'une substance chimique sur un organisme ou un groupe d'organismes est couramment réalisée *via* des essais dose-réponse. Ces essais permettent de déterminer des valeurs critiques telles que l'EC_x (Concentration Efficace pour x% de la population) ou la LC_x (Concentration Létale pour x% de la population), des paramètres essentiels pour évaluer le risque environnemental d'un polluant (cf méthodologie paragraphe 5.6).

Dans le cadre de ce stage, trois modèles de régression non linéaire ont été utilisés : le modèle log-logistique, le modèle log-normal, et le modèle Weibull. Ces modèles sont couramment employés en écotoxicologie pour décrire la relation dose-réponse et permettent de calculer les EC_x ou LC_x avec précision (E et L dépend du type de test cela doit etre défini dans ma méthodologie, mais pour simplifier je ne parlereai que de EC_x.Peux tu donc me faire une phrase ou deux bien ecrite à mettre dans ma methodo?).

Les paramètres estimés de ces modèles sont présentés dans le tableau 8. D'après les résultats, on voit que :

1. Paramètre b (Pente de la courbe dose-réponse):

Ce paramètre détermine la pente de la courbe dose-réponse. Une valeur plus élevée de b indique une transition plus abrupte entre les faibles et les fortes réponses à la dose, suggérant que les effets de la substance augmentent rapidement sur une courte plage de concentrations. Dans le modèle log-logistique, $b = 3.9007$ ($p = 0.0021$), ce qui indique une réponse rapide et significative à la dose. Le modèle Weibull montre une pente encore plus forte ($b = 4.9421$, $p = 0.00001$), renforçant l'idée que les effets sont fortement liés à une petite variation de la concentration.

2. Paramètre c (Effet minimum observé ou limite inférieure):

Ce paramètre représente l'effet minimal observé (l'asymptote inférieure) lorsque la concentration est faible ou nulle. Dans certains cas, c peut être interprété comme la réponse de base ou de fond. Les valeurs de c sont négatives dans tous les modèles, ce

qui peut indiquer des difficultés d'ajustement pour ces paramètres, ou une base d'effet inférieure à zéro, ce qui n'est pas biologiquement plausible dans tous les contextes. Les p-values non significatives suggèrent que c n'est pas un paramètre robuste dans ces modèles.

3. Paramètre d (Effet maximum observé ou limite supérieure) :

Ce paramètre décrit l'effet maximal observé (l'asymptote supérieure) à haute concentration. C'est une estimation de l'effet lorsque la concentration atteint un niveau où presque toute la population est affectée. Les trois modèles montrent des valeurs de d très significatives (toutes avec $p < 2.2e-16$), ce qui indique un ajustement robuste pour la réponse maximale à haute dose. Par exemple, $d = 60.2790$ pour le modèle log-logistique montre que la réponse maximale attendue dans cette condition est bien capturée.

4. Paramètre e (EC_{50} ou concentration inflexion de la courbe) :

Ce paramètre correspond à la concentration médiane efficace, EC_{50} , c'est-à-dire la concentration à laquelle 50% de l'effet maximal est observé. Il représente le point d'inflexion de la courbe dose-réponse. Le modèle log-logistique donne une estimation de $e = 6.8654$ ($p < 0.0001$), ce qui suggère qu'une concentration relativement basse est nécessaire pour atteindre 50% de l'effet maximal. En revanche, le modèle log-normal présente un $e = 15.5224$ ($p = 0.4005$), beaucoup moins significatif, indiquant que ce modèle pourrait mal ajuster les données dans la zone critique de l' EC_{50} , c'est-à-dire dans la région où la réponse de la population change le plus rapidement. Cette faible significativité pourrait être due à une mauvaise adéquation du modèle log-normal aux données expérimentales, particulièrement dans les zones de transition importantes.

Paramètre	Log-Logistique	Log-Normale	Weibull
b	3.9007 ($p = 0.0021$)	-1.2779 ($p = 0.0472$)	4.9421 ($p = 0.00001$)
c	-36.9821 ($p = 0.2377$)	-262.0235 ($p = 0.6049$)	-2.6619 ($p = 0.4808$)
d	60.2790 ($p < 2.2e-16$)	61.1410 ($p < 2e-16$)	59.1626 ($p < 2.2e-16$)
e	6.8654 ($p < 0.0001$)	15.5224 ($p = 0.4005$)	6.1774 ($p < 2.2e-16$)

Table 8: Estimation des paramètres pour les modèles log-logistique, log-normal et Weibull.

Le choix du modèle adéquat est important pour estimer avec précision les ECx, car il influence directement la qualité et la fiabilité des résultats obtenus. Sur l'exemple des données obtenues avec l'essai 201[?], le modèle log-normal, bien que souvent utilisé, a montré des limitations dans ce contexte particulier. La faible significativité de certains paramètres critiques, comme le paramètre e , suggère que ce modèle pourrait ne pas capturer adéquatement la relation dose-réponse pour les données étudiées. Cette conclusion est renforcée par la comparaison systématique des modèles, où le modèle Weibull a offert des estimations plus robustes et mieux adaptées. Les limitations du modèle log-normal peuvent ainsi être attribuées à son incapacité à ajuster correctement les variations rapides de réponse autour de l' EC_{50} , ce qui est essentiel pour une modélisation précise des effets à des concentrations spécifiques.

C'est pourquoi la comparaison systématique des modèles est une étape indispensable pour garantir la robustesse des conclusions tirées de l'étude.

8.1 Comparaison des Modèles

Afin de déterminer quel modèle s'ajuste le mieux aux données, nous avons utilisé deux critères de sélection : le critère d'information d'Akaike (AIC) et le facteur de Bayes ($B_{1/0}$) (cf méthodologie). Ces critères permettent de comparer les modèles en termes de qualité d'ajustement tout en tenant compte de la complexité du modèle. Notamment, nous n'avons pas utilisé le critère d'information bayésien (BIC) dans cette analyse, pour capturer avec précision la complexité des données expérimentales, sans sur-pénaliser les modèles qui offrent un meilleur ajustement, même s'ils sont légèrement plus complexes. En effet, le BIC (trop strict) a tendance à pénaliser excessivement des modèles qui, bien qu'un peu plus complexes, capturent mieux les nuances des données.

Modèle	AIC
Log-logistique	307.1229
Log-normal	310.4866
Weibull	301.9307

Table 9: Valeurs de l'AIC

Comparaison des Modèles	$(B_{1/0})$	$(B_{1/0})$ avec MH
Log-logistique vs Log-normal	0.01545374	0.01545374
Log-logistique vs Weibull	0.0006039546	0.0006039546
Log-normal vs Weibull	0.03908144	0.03908144

Table 10: Facteurs de Bayes

Les résultats montrent que le modèle Weibull possède la valeur d'AIC la plus faible (301.9307), indiquant qu'il s'ajuste le mieux aux données tout en restant relativement simple. Le modèle log-logistique suit avec un AIC de 307.1229, tandis que le modèle log-normal présente l'AIC le plus élevé (310.4866), ce qui en fait le modèle le moins performant dans cette comparaison. Les facteurs de Bayes (avec ou sans MH) corroborent les conclusions tirées de l'AIC. Le modèle log-logistique est préféré au modèle log-normal, mais il est moins performant que le modèle Weibull. De plus, la comparaison entre le modèle log-normal et le modèle Weibull montre une nette préférence pour le modèle Weibull. L'analyse bayésienne a utilisé l'algorithme de Metropolis-Hastings (MH) pour échantillonner les distributions postérieures des paramètres, essentiel pour calculer les facteurs de Bayes avec a priori explicite. Deux approches ont été appliquées :

1. **Méthode avec MH (a priori explicite):** Cette méthode utilise explicitement les distributions a priori pour échantillonner les postérieures. Ici, le MH a bien fonctionné pour les approximations, mais avec des défis de convergence pour certains paramètres, comme le montre la variabilité dans les trace plots et les taux d'acceptation. Par exemple, dans le modèle Weibull, un taux d'acceptation de 0.2248 a été observé, suggérant une exploration parfois limitée de l'espace des paramètres.
2. **Méthode sans MH (a priori implicite):** Cette méthode repose sur des a priori implicites, où la distribution postérieure est estimée sans un échantillonnage ex-

plicite via MH. Ici, les résultats ont montré une cohérence avec les méthodes explicites, renforçant la robustesse des conclusions. Cependant, cette approche peut sous-estimer l'incertitude, particulièrement dans des espaces de paramètres complexes, ce qui souligne l'importance d'une évaluation soigneuse des diagnostics de convergence.

Modèle	Taux d'acceptation
Modèle log-logistique	0.5251
Modèle log-normal	0.7885
Modèle Weibull	0.2248

Table 11: Taux d'acceptation pour chaque modèle.

Les graphiques ACF montrent que certains paramètres, notamment la pente du modèle log-logistique et la variance logarithmique du modèle log-normal, présentent une meilleure indépendance entre les échantillons successifs.

En revanche, les paramètres de concentration effective (EC_{50}), moyenne logarithmique, et ceux du modèle Weibull montrent une forte autocorrélation, suggérant un mélange moins efficace et une convergence potentiellement plus lente. Il pourrait être bénéfique de prolonger les chaînes ou d'ajuster les paramètres d'échantillonnage pour ces derniers. Les résultats des différentes approches de modélisation sont remarquablement cohérents. L'approche initiale basée sur l'AIC et la log-vraisemblance a identifié le modèle de Weibull comme le meilleur ajustement pour les données, suivi du modèle log-logistique. Cette conclusion a été renforcée par l'analyse bayésienne utilisant des chaînes MCMC, qui a également favorisé le modèle de Weibull. Cependant, il est important de noter que, bien que les résultats des chaînes MCMC confirment ceux de l'approche initiale, des diagnostics ont montré des lacunes dans la convergence et l'efficacité de l'échantillonnage. Cela suggère qu'une optimisation supplémentaire des chaînes MCMC pourrait encore affiner ces conclusions.

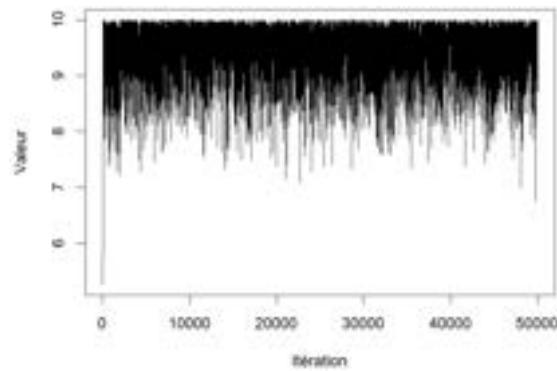


Figure 8.1: Trace log-logistique Pente (b)

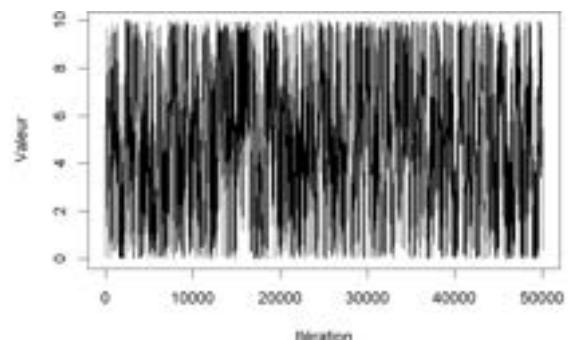


Figure 8.2: Trace log-logistique (EC_{50})

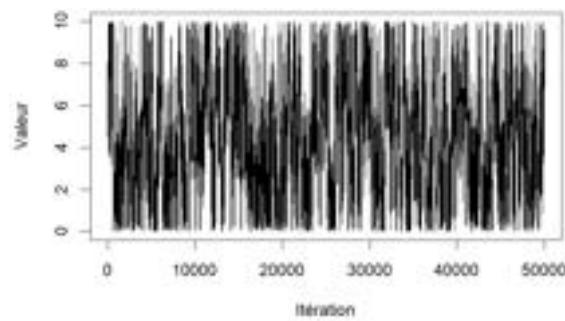


Figure 8.3: Trace log-normal Moyenne logarithmique

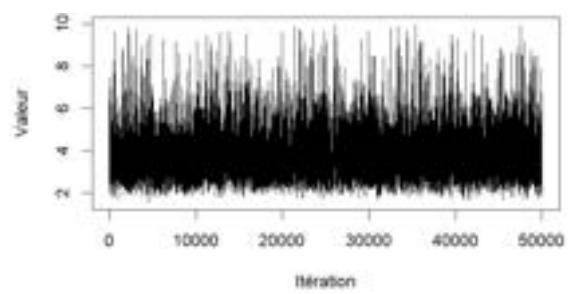


Figure 8.4: Trace log-normal Variance logarithmique

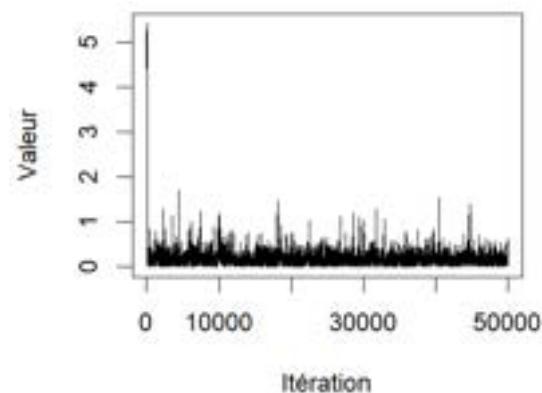


Figure 8.5: Trace Weibull Paramètre de forme

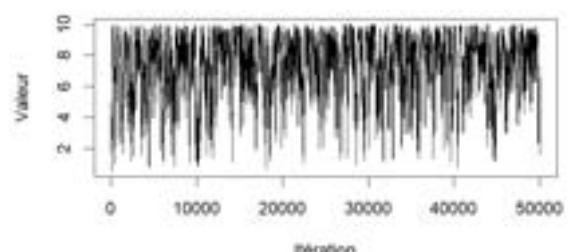


Figure 8.6: Trace Weibull Paramètre d'échelle

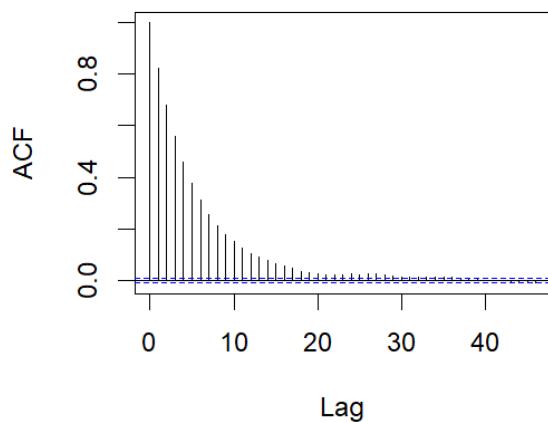


Figure 8.7: ACF pour le modèle logistique Pente (b)

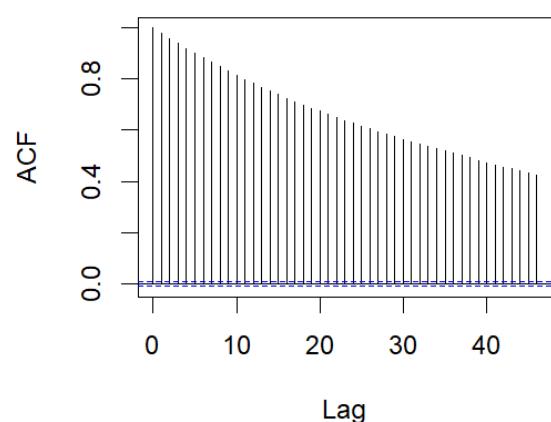


Figure 8.8: ACF pour le modèle logistique (EC_{50})

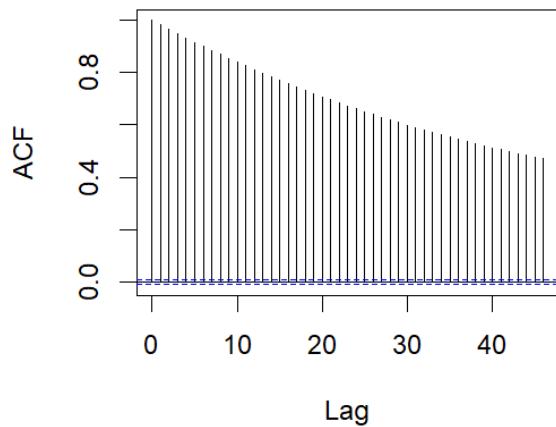


Figure 8.9: ACF pour le modèle log-normal Moyenne logarithmique

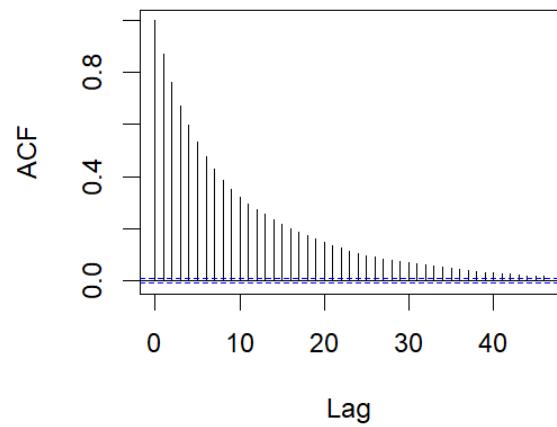


Figure 8.10: ACF pour le modèle log-normal Variance logarithmique

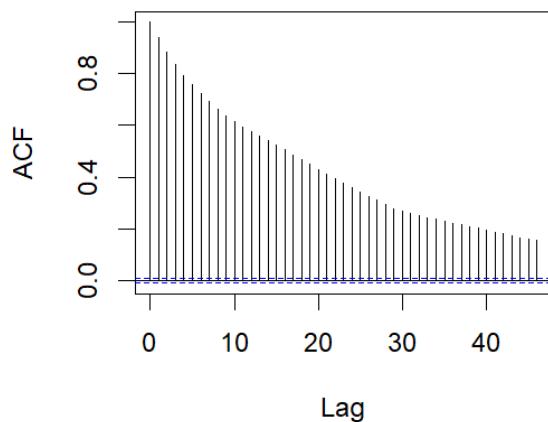


Figure 8.11: ACF pour le modèle Weibull Paramètre de forme

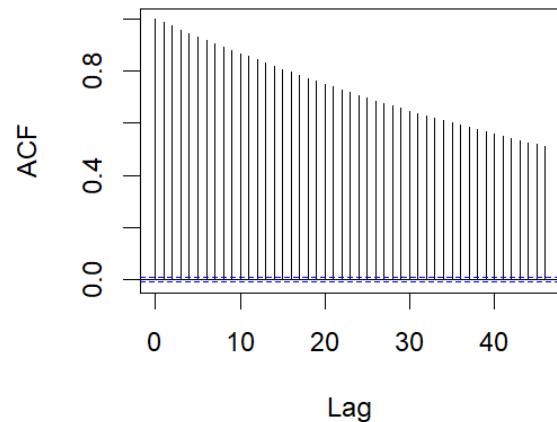


Figure 8.12: ACF pour le modèle Weibull Paramètre d'échelle

9 Estimation des ECx ou LCx

Par exemple, les valeurs de l' EC_{10} , l' EC_{20} , et l' EC_{50} , indiquent les concentrations où 10%, 20%, et 50% de l'effet maximal sont atteints. Ces mesures sont essentielles pour l'évaluation des risques écotoxicologiques, car elles permettent de quantifier la toxicité d'une substance de manière précise et fiable.

Dose Effective	Estimation	Erreur Standard
ERC10 (Weibull)	3.9179	0.3020
ERC20 (Weibull)	4.5603	0.2189
ERC50 (Weibull)	5.7359	0.1224

Table 12: Estimations des doses effectives à 10%, 20%, et 50% pour le modèle Weibull.

Pour le modèle Weibull (cf méthodologie) l' EC_{50} est la concentration à laquelle la réponse est à mi-chemin entre c et d , c'est-à-dire où $f(EC_{50}) = \frac{c+d}{2}$.

Donc pour $f(x) = \frac{c+d}{2}$, on a :

$$\frac{c+d}{2} = d - (d-c) \cdot \exp\left(-\left(\frac{EC_{50}}{b}\right)^a\right)$$

En isolant $\frac{EC_{50}}{b}$, on obtient :

$$\exp\left(-\left(\frac{EC_{50}}{b}\right)^a\right) = \frac{1}{2}$$

Ce qui conduit à :

$$\frac{EC_{50}}{b} = (\log(2))^{\frac{1}{a}} \quad \text{et donc} \quad EC_{50} = b \cdot (\log(2))^{\frac{1}{a}}$$

L'ECx correspond à la concentration x pour laquelle $f(ECx) = c + (d - c) \cdot \frac{x}{100}$.
Donc, pour EC_{10} et EC_{20} on a : $f(EC10) = c + 0.1 \cdot (d - c)$ et $f(EC20) = c + 0.2 \cdot (d - c)$
On peut réécrire ceci en isolant ECx :

$$f(ECx) = d - (d-c) \cdot \exp\left(-\left(\frac{ECx}{b}\right)^a\right)$$

Cela conduit à l'équation suivante :

$$c + \left(\frac{x}{100}\right) \cdot (d - c) = d - (d-c) \cdot \exp\left(-\left(\frac{ECx}{b}\right)^a\right)$$

Isolons ECx :

$$\exp\left(-\left(\frac{ECx}{b}\right)^a\right) = 1 - \frac{x}{100}$$

Cela peut être résolu pour ECx :

$$ECx = b \cdot \left(-\log \left(1 - \frac{x}{100} \right) \right)^{\frac{1}{a}}$$

Pour vérifier les calculs des ECx (EC_{10} , EC_{20} , EC_{50}) en utilisant les paramètres fournis dans la table 8 pour le modèle Weibull.

$d = 59.1626$ (approximé à partir des réponses maximales au contrôle).

$c = -2.6619$ (approximé à partir des réponses minimales aux hautes concentrations).

$b = 4.9421$ mg/L (approximé à partir de l' EC_{50} obtenu avec drc)

$e = a = 6.1774$ (approximé à partir des résultats drc).

Donc, on a :

1.

$$EC_{50} = 4.9421 \cdot (\log(2))^{\frac{1}{6.1774}} = 4.9421 \cdot (0.6931)^{\frac{1}{6.1774}} \approx 4.9421 \cdot 0.9126 \approx \mathbf{4.508} \text{ mg/L}$$

2.

$$EC_{10} = 4.9421 \cdot \left(-\log \left(1 - \frac{10}{100} \right) \right)^{\frac{1}{6.1774}} = 4.9421 \cdot (-\log(0.9))^{\frac{1}{6.1774}}$$

$$EC_{10} = 4.9421 \cdot (0.1054)^{\frac{1}{6.1774}} \approx 4.9421 \cdot 0.7057 \approx \mathbf{3.487} \text{ mg/L}$$

3.

$$EC_{20} = 4.9421 \cdot \left(-\log \left(1 - \frac{20}{100} \right) \right)^{\frac{1}{6.1774}} = 4.9421 \cdot (-\log(0.8))^{\frac{1}{6.1774}}$$

$$EC_{20} = 4.9421 \cdot (0.2231)^{\frac{1}{6.1774}} \approx 4.9421 \cdot 0.8046 \approx \mathbf{3.975} \text{ mg/L}$$

Comparées aux valeurs obtenues avec le package "drc", il est notable que les résultats peuvent varier légèrement en fonction de l'ajustement spécifique des modèles et des méthodes utilisées pour l'estimation des paramètres. Les intervalles de confiance (IC) fournissent une mesure de la précision des estimations des ECx, en donnant une indication sur la variabilité autour de ces valeurs. Afin de mieux quantifier cette variabilité et d'obtenir des IC plus robustes, nous appliquons la méthode du bootstrap

9.0.1 Test de Permutation et Intervalles de Confiance Bootstrap

Dose Effective	Intervalle de Confiance à 95% (Percentile)
ERC10 (Weibull)	[3.192, 6.511]
ERC20 (Weibull)	[4.089, 9.127]
ERC50 (Weibull)	[5.609, 15.403]

Table 13: Intervalles de confiance bootstrap à 95% pour les doses effectives estimées.

Le test de permutation a été utilisé pour évaluer la significativité des résultats obtenus précédemment. Étant une méthode non paramétrique, il permet de vérifier de manière robuste les estimations des concentrations efficaces (EC_{10} , EC_{20} , et EC_{50}) issues du modèle

dose-réponse Weibull à 4 paramètres, sans crainte de biais lié aux hypothèses de distribution. Selon les estimations réelles effectuées, les valeurs d'ECx étaient de 3.91788 mg/L pour l' EC_{10} , 4.56033 mg/L pour l' EC_{20} , et 5.73586 mg/L pour l' EC_{50} . Le test a donc procédé en permutant aléatoirement les données pour générer plusieurs ensembles de données fictives, où les relations entre les concentrations et les réponses biologiques étaient perturbées. Pour chaque ensemble permué, les estimations des ECx ont été recalculées, créant ainsi une distribution nulle des valeurs d'ECx sous l'hypothèse nulle (H_0), selon laquelle la relation observée initialement est due au hasard. En comparant les estimations réelles à ces distributions permutées, les p-values obtenues pour l' EC_{10} , l' EC_{20} , et l' EC_{50} étaient toutes de 0.989, ce qui est très élevé. Cela suggère qu'il y a une probabilité de 98.9% que les différences observées entre les estimations réelles et celles issues des permutations soient dues au hasard, et non à un effet réel de la concentration sur la réponse biologique. Les intervalles de confiance Bootstrapé obtenus (cf tableau 13) sont cruciaux pour évaluer la fiabilité des estimations d'ECx. Ils montrent l'étendue possible des valeurs d'ECx, offrant une plage dans laquelle la vraie valeur d'ECx est susceptible de se situer à 95% de confiance. Les résultats révèlent que l'incertitude des estimations augmente avec le niveau de l'effet (de l' EC_{10} à l' EC_{50}). L' EC_{10} a un intervalle relativement étroit (3.192 à 6.511 mg/L), indiquant une estimation plus précise de cette concentration. Cela suggère que l' EC_{10} est une mesure relativement stable et fiable, avec une moindre sensibilité aux variations dans les données expérimentales.

L' EC_{20} présente un intervalle plus large (4.089 à 9.127 mg/L), ce qui montre une augmentation de l'incertitude. L'estimation est moins précise que celle de l' EC_{10} , reflétant une plus grande variabilité dans cette région de la courbe dose-réponse. Et enfin, l' EC_{50} a l'intervalle de confiance le plus large (5.609 à 15.403 mg/L), soulignant une incertitude encore plus grande. Ce résultat est attendu puisque, dans les modèles dose-réponse, la précision diminue généralement à mesure que l'on s'éloigne de l' EC_{50} en raison de la pente de la courbe qui devient moins raide. Les intervalles de confiance montrent que, malgré les incertitudes croissantes aux doses plus élevées, les estimations d'ECx sont robustes. Le modèle Weibull à 4 paramètres s'avère être une méthode efficace pour capturer la relation dose-réponse, et l'ajout des IC bootstrap renforce la confiance dans ces estimations. Cependant, il est important de noter que l'approche par permutation et l'approche par bootstrap peuvent être considérées comme **complémentaires** plutôt qu'exclusives l'une de l'autre. Le bootstrap, permet entre autre, de construire des intervalles de confiances sans nécessiter l'hypothèse d'échangeabilité requise par les tests de permutation. Le choix entre les deux méthodes dépend donc du contexte spécifique de l'analyse et des caractéristiques des données.

En pratique, si les données présentent une structure où l'hypothèse d'échangeabilité est raisonnable, les tests par permutation peuvent fournir des résultats **robustes** pour tester l'indépendance ou la comparaison de distributions entre groupes. Dans les cas où cette hypothèse est difficile à justifier, ou lorsque l'objectif est d'estimer des intervalles de confiance pour des paramètres de modèle, le bootstrap peut être préférable. Ainsi, il peut être judicieux de comparer les résultats obtenus par ces deux approches pour une même analyse, afin d'évaluer leur concordance ou de comprendre les divergences éventuelles. En écotoxicologie, cette complémentarité permet d'aborder l'incertitude des estimations sous différents angles, en renforçant la confiance dans les conclusions tirées, ou en identifiant des aspects des données qui nécessitent une attention particulière.

En somme, les intervalles de confiance Bootstrapés et l'analyse par modèle Weibull offrent une vision claire et robuste des estimations d'ECx, malgré une incertitude croissante avec l'augmentation du niveau d'effet. Cependant, une conclusion importante pour les évaluations

teurs de risque réside dans la détermination de la **valeur seuil à x% d'effet significatif pour l'espèce d'algue testé**, qui constitue une finalité cruciale de cette application. Cette valeur seuil, obtenue en combinant les approches par permutation et bootstrap, permet de fournir une évaluation complète et fiable des risques écotoxicologiques, en tenant compte des incertitudes associées à chaque niveau d'effet.

10 Conclusion & Perspectives

Le développement de l'application a permis de réaliser avec succès le processus d'évaluation en offrant aux évaluateurs un outil pour visualiser graphiquement des endpoints valides et fiables. Grâce à cette application, les évaluateurs pourront suivre un processus guidé, pour identifier les tests statistiques les plus appropriés. Les endpoints sont vérifiés sous les angles de la fiabilité et de la robustesse, ce qui constitue une avancée significative dans l'automatisation et la fiabilisation de l'évaluation. Un exemple concret de l'utilisation de cette application est l'analyse de l'essai 201 de l'OCDE, conçu pour évaluer les effets toxiques d'une substance sur des organismes aquatiques. Cet essai a été méticuleusement analysé au travers de plusieurs étapes statistiques de l'application. Les données, après dépistage des outliers qui n'ont pas été éliminés, ont été utilisées pour estimer les concentrations effectives EC_x à différents niveaux (EC₁₀, EC₂₀, EC₅₀) en utilisant un modèle Weibull, sélectionné par le critère d'information d'Akaike (AIC) et par le facteur Bayes. L'application RShiny que j'ai développée au cours de ce stage est encore à une phase initiale, mais elle a déjà permis de coder plusieurs essais d'écotoxicité. Jusqu'à présent, six essais d'écotoxicité (fonctionnels) ont été entièrement codés, couvrant une gamme variée de besoins identifiés par l'unité U3EiV. De plus, l'application offre des options flexibles pour analyser différents jeux de données, indépendamment du type d'essai standardisé employé. Durant le dernier mois de mon stage, mon objectif est de compléter le codage d'une partie des 15 essais d'écotoxicité qui m'ont été suggérés par l'équipe, afin de répondre de manière plus exhaustive aux exigences spécifiques de l'unité. Je prévois également d'implémenter une fonctionnalité permettant de générer automatiquement des rapports au format PDF ou Word. Cette fonctionnalité offrira aux évaluateurs la possibilité d'extraire un résumé succinct des analyses statistiques effectuées via l'application RShiny. Il est toutefois important de noter que, compte tenu de l'ampleur du projet, le développement de l'application devra se poursuivre après la fin de mon stage pour aboutir à un produit finalisé, utilisable de manière routinière par les évaluateurs de risques. L'application présente un fort potentiel d'utilisation, non seulement dans le domaine de l'écotoxicologie, mais également dans d'autres disciplines, telles que la toxicologie. Elle pourrait également servir de support lors de discussions au niveau des instances européennes et internationales, notamment pour la réévaluation des documents guides statistiques ou pour le développement d'un outil harmonisé à l'échelle européenne. En résumé, le travail accompli durant ce stage représente une première étape essentielle dans un projet de grande envergure. Ce projet pose les bases d'un outil polyvalent, qui pourra être enrichi et utilisé par d'autres, dans un cadre élargi. Mon implication dans ce projet a non seulement permis d'avancer significativement dans le développement de l'application, mais a également ouvert la voie à de nombreuses perspectives de développement et d'amélioration futures.

A Annexe A : Inventaire des tests

Table 14: Forces et Limites des Tests Statistiques

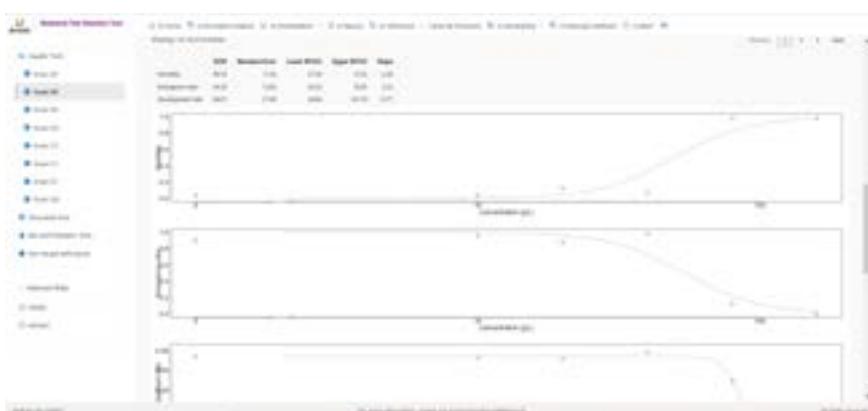
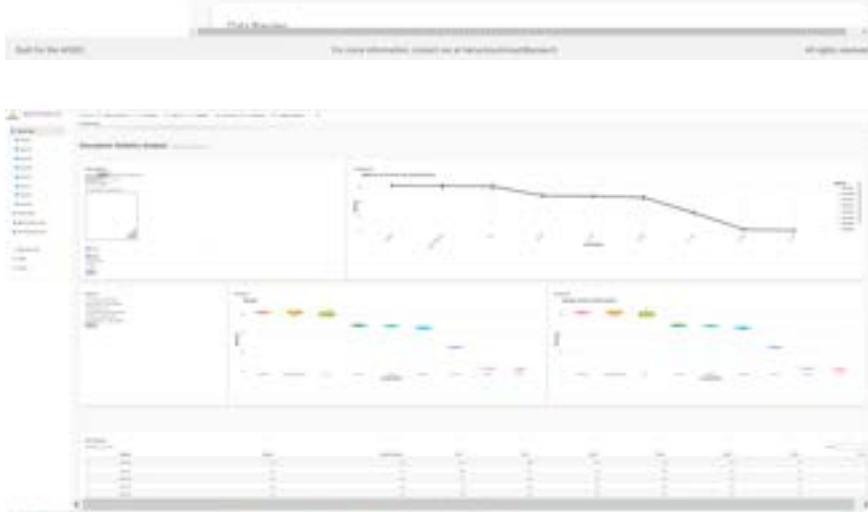
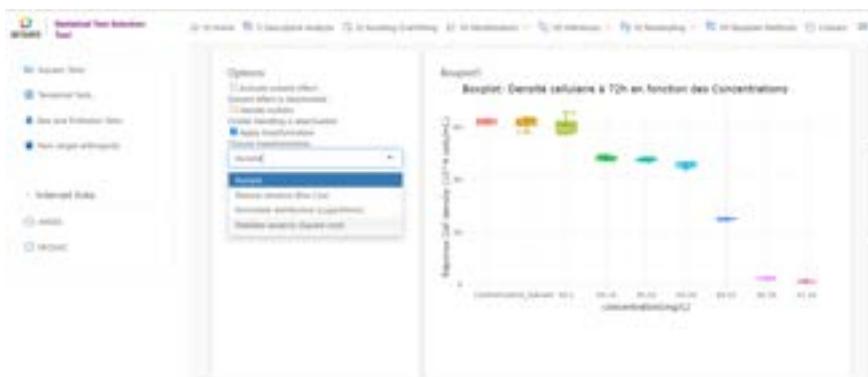
Test	Forces	Limites
T-test	Simple à comprendre et à exécuter; puissant pour comparer les moyennes de deux groupes	Sensible aux écarts de normalité, trompeur avec outliers ou variances inégales
ANOVA	Compare les moyennes de trois groupes ou plus; plus robuste que multiples t-tests	Ne précise pas quels groupes diffèrent; sensible à la non-normalité et hétérogénéité des variances
Régression Linéaire	Relation quantitative entre variables; peut inclure multiples variables indépendantes	Nécessite linéarité, indépendance, normalité des résidus, homoscédasticité; sensible aux valeurs aberrantes et multicollinearité
Test du Chi-carré	Flexible, utilisable pour comparaisons catégorielles; requiert seulement des données de fréquence	Cases du tableau doivent avoir suffisamment d'observations; ne fournit pas l'ampleur de l'association
Mann-Whitney & Kruskal-Wallis	Convient aux distributions non normales; utilisables avec petits échantillons	ne fournissent pas de mesure de l'ampleur de l'effet Moins puissants que les équivalents paramétriques;

Table 15: Critères de Choix des Tests Statistiques

Test	Critères de Choix
T-test	Données quantitatives continues pour deux groupes; nécessite une distribution normale et des variances similaires
ANOVA	Données continues pour trois groupes ou plus; nécessite une distribution normale et des variances égales parmi les groupes
Régression Linéaire	Pour prédire une variable continue à partir d'une ou plusieurs variables indépendantes; relations entre variables doivent être linéaires
Test du Chi-carré	Comparer distributions de variables catégorielles ou tester l'indépendance entre deux variables catégorielles
Mann-Whitney & Kruskal-Wallis	Données ne respectant pas la normalité et/ou l'homogénéité des variances; Mann-Whitney pour deux groupes, Kruskal-Wallis pour trois ou plus

B Annexe :Application interface





C Annexe :Application architecture

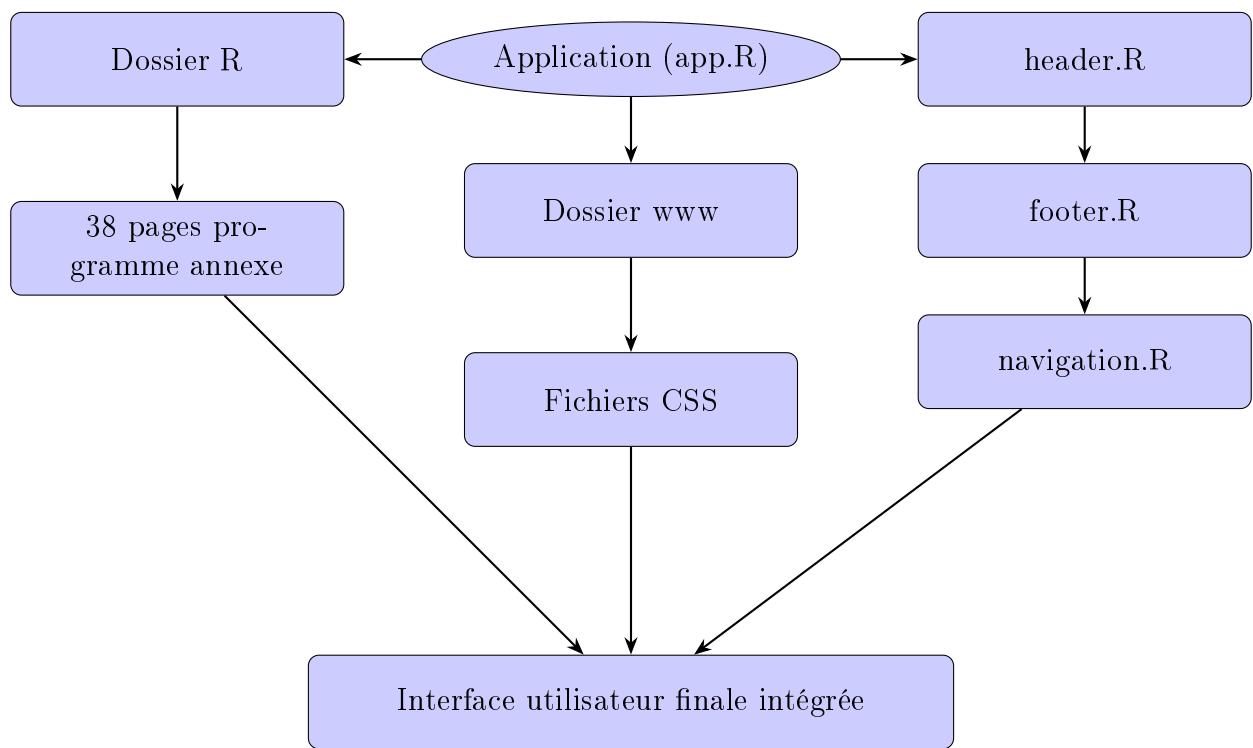


Figure C.1: Architecture générale de l'application

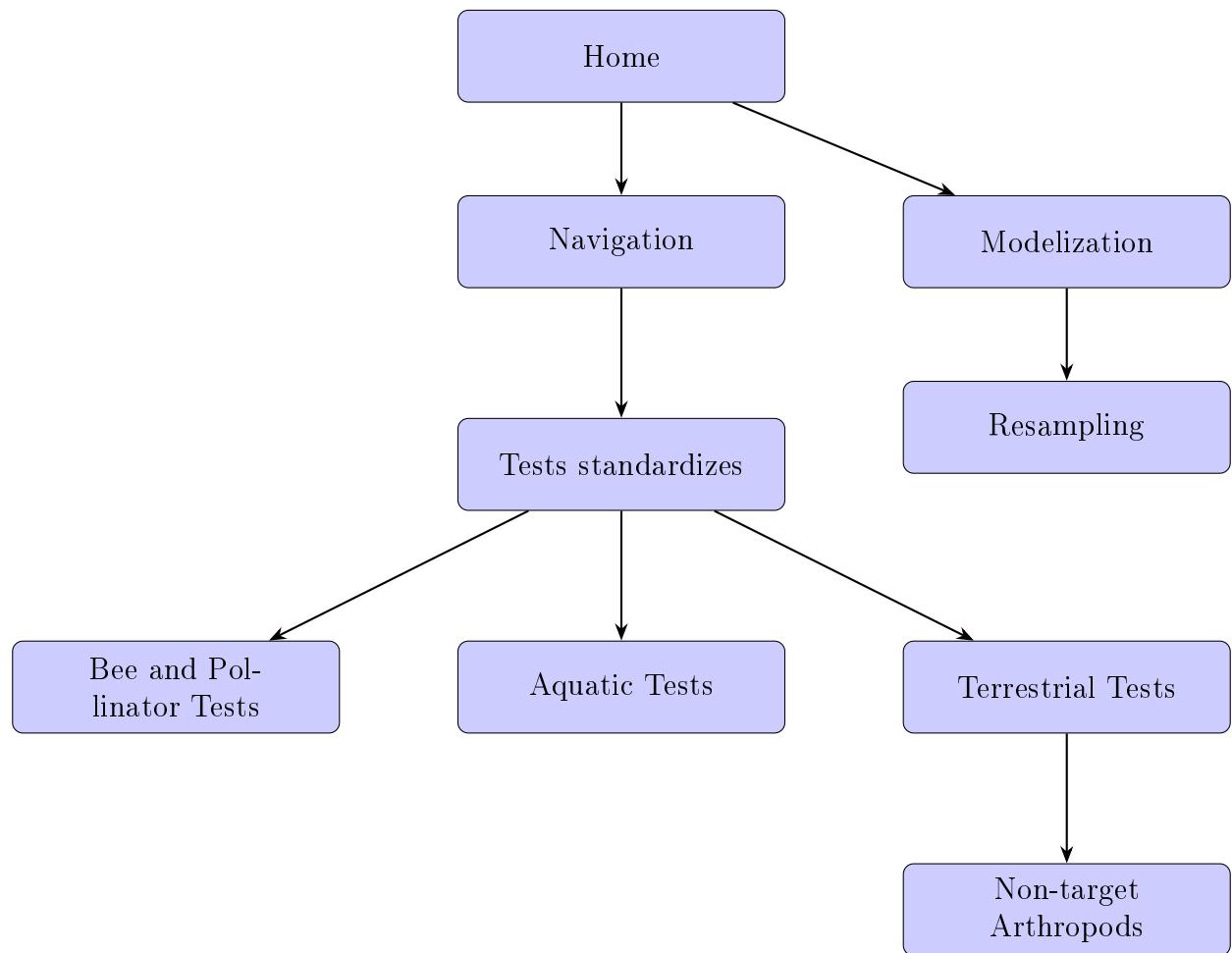


Figure C.2: Diagramme essais standardisés

Annexe : Script application

```

1 library(shiny)
2 library(shiny.fluent)
3 library(shiny.router)
4 library(sass)
5 library(leaflet)
6 library(ggplot2)
7 library(plotly)
8 library(glue)
9 library(dplyr)
10 library(tidyr)
11 library(DT)
12 library(DescTools)
13 library(Publish)
14 library(car)
15 library(multcomp)
16 library(ggpubr)
17 library(reshape2)
18 library(tidyverse)
19 library(drc)
20 library(here)
  
```

```

21 library(rmarkdown)
22 library(shinyjs)
23 library(shinyCSSloaders)
24 library(MASS) # Pour la transformation Box-Cox
#Pour bayesien
25 library(BayesFactor)
26 library(boot)
27 library(coin)
28 library(plotrix)
29 library(stats)
30 library(bayesmeta)
31 library(pwr)
32 ## Fonctions #####
33 SummaryVisu <- function(df, resp, trt) {
34   df$trt <- as.factor(df[[trt]])
35   df$resp <- df[[resp]]
36   df <- df[complete.cases(df[, c("trt", "resp")]), ]
37   concentration <- levels(df$trt)
38   concentration <- as.vector(concentration)
39   Means <- tapply(df$resp, df$trt, mean, na.rm = TRUE)
40   Means <- round(as.vector(Means), 4)
41   Sample_Size <- tapply(df$resp, df$trt, length)
42   Sample_Size <- as.vector(Sample_Size)
43   Median <- tapply(df$resp, df$trt, median)
44   Median <- round(as.vector(Median), 4)
45   Standard_Deviation <- tapply(df$resp, df$trt, sd)
46   Standard_Deviation <- round(as.vector(Standard_Deviation), 4)
47   Standard_Error <- round(Standard_Deviation / sqrt(Sample_Size), 4)
48   out <- data.frame(concentration, Sample_Size, Means, Median,
49     Standard_Deviation, Standard_Error)
50   return(out)
51 }
52 # Fonction Dunnett
53 analyze_dunnett <- function(data) {
54   data$response <- as.numeric(data$response)
55   data$treatment <- as.factor(data$treatment)
56
57   aov1 <- aov(response ~ treatment, data = data)
58   summary_aov <- summary(aov1)
59   significant <- ifelse(summary_aov[[1]][["Pr(>F)"]][[1]] <= 0.05,
60     "ANOVA results: Difference is statistically significant.",
61     "ANOVA results: Difference is not statistically significant.")
62
63   sw <- shapiro.test(residuals(aov1))
64   normality <- ifelse(sw$p.value <= 0.05, "Normality is rejected",
65     "Normality is accepted")
66
67   b <- bartlett.test(response ~ treatment, data = data)
68   homoscedasticity <- ifelse(b$p.value <= 0.05, "Homoscedasticity is rejected",
69     "Homoscedasticity is accepted")

```

```

68
69  Res_dunnett <- DunnettTest(data$response, data$treatment, control
70    = "0", conf.level = 0.95)
71  summary_dunnett <- summary(glht(aov1, linfct = mcp(treatment = "Dunnett")))
72
73  return(list(summary_aov = summary_aov,
74    significant = significant,
75    normality = normality,
76    homoscedasticity = homoscedasticity,
77    Res_dunnett = Res_dunnett,
78    summary_dunnett = summary_dunnett))
79
80 # Fonction ANOVA
81 analyze_anova <- function(data) {
82   data$response <- as.numeric(data$response)
83   data$concentration <- as.factor(data$concentration)
84
85   aov1 <- aov(response ~ concentration, data = data)
86   summary_aov <- summary(aov1)
87   significant <- ifelse(summary_aov[[1]][["Pr(>F)"]][[1]] <= 0.05,
88     "ANOVA results: Difference is statistically significant.",
89     "ANOVA results: Difference is not statistically significant.")
90
91 # Fonction Shapiro/bartlett
92 sw <- shapiro.test(residuals(aov1))
93 normality <- ifelse(sw$p.value <= 0.05, "Normality is rejected", "Normality is accepted")
94
95 b <- bartlett.test(response ~ concentration, data = data)
96 homoscedasticity <- ifelse(b$p.value <= 0.05, "Homoscedasticity is rejected", "Homoscedasticity is accepted")
97
98 return(list(summary_aov = summary_aov,
99    significant = significant,
100   normality = normality,
101   homoscedasticity = homoscedasticity))
102
103 # Fonction Bootstrap
104 #analyze_bootstrap <- function(data, num_bootstrap = 1000) {
105 #  boot_means <- replicate(num_bootstrap, {
106 #    sample_data <- sample(data$response, replace = TRUE)
107 #    mean(sample_data)
108 #  })
109 #  ci <- quantile(boot_means, probs = c(0.025, 0.975))
110 #  mean_val <- mean(boot_means)
111
112

```

```

113 #   bootstrap_data_frame <- data.frame(Bootstrap = 1:num_bootstrap,
114 #                                         Mean = boot_means)
115 #
116 #   p <- ggplot(bootstrap_data_frame, aes(x = Mean)) +
117 #         geom_histogram(bins = 30, fill = "blue", alpha = 0.6) +
118 #         geom_vline(aes(xintercept = mean_val), color = "red", linetype
119 #                    = "dashed") +
120 #         labs(title = "Distribution des moyennes bootstrap",
121 #               x = "Moyenne bootstrap", y = "Fréquence") +
122 #         theme_minimal()
123 #
124 #   return(list(mean = mean_val, ci = ci, plot = p))
125 #}
126 calculate_ec <- function(response, concentration) {
127   tryCatch({
128     #logistic model
129     fit <- nls(response ~ 1 / (1 + exp(-b * (log(concentration) -
130       log(EC_{50})))), start = list(b = 1, EC50 = median(concentration)),
131       control = nls.control(warnOnly = TRUE))
132     #coefficients
133     b <- coef(fit)["b"]
134     EC50 <- coef(fit)["EC50"]
135
136     #EC values
137     EC10 <- exp((log(10) - b * log(EC50)) / b)
138     EC20 <- exp((log(20) - b * log(EC50)) / b)
139
140     return(c(EC10 = EC10, EC20 = EC20, EC50 = EC50))
141   }, error = function(e) {
142     return(c(EC10 = NA, EC20 = NA, EC50 = NA))
143   })
144 }
145 analyze_bootstrap <- function(data, num_bootstrap = 1000) {
146   data$response <- as.numeric(as.character(data$response))
147   data$concentration <- as.numeric(as.character(data$concentration))
148   data <- data[!is.na(data$response) & !is.na(data$concentration), ]
149
150   if (nrow(data) == 0) {
151     stop("No valid numeric data found in the response or
152       concentration columns.")
153   }
154
155   boot_ecs <- replicate(num_bootstrap, {
156     sample_indices <- sample(seq_len(nrow(data)), replace = TRUE)
157     sample_data <- data[sample_indices, ]
158     calculate_ec(sample_data$response, sample_data$concentration)
159   })
160
161   boot_ecs <- t(boot_ecs)
162   colnames(boot_ecs) <- c("EC10", "EC20", "EC50")

```

```

161  ec10_ci <- quantile(boot_ecs[, "EC10"], probs = c(0.025, 0.975),
162    na.rm = TRUE)
163  ec20_ci <- quantile(boot_ecs[, "EC20"], probs = c(0.025, 0.975),
164    na.rm = TRUE)
165  ec50_ci <- quantile(boot_ecs[, "EC50"], probs = c(0.025, 0.975),
166    na.rm = TRUE)
167
168
169  bootstrap_data_frame <- data.frame(Bootstrap = 1:num_bootstrap,
170    EC10 = boot_ecs[, "EC10"],
171    EC20 = boot_ecs[, "EC20"], EC50
172      = boot_ecs[, "EC50"])
173
174  p <- ggplot(bootstrap_data_frame, aes(x = EC50)) +
175    geom_histogram(bins = 30, fill = "blue", alpha = 0.6) +
176    geom_vline(aes(xintercept = mean_ec50), color = "red", linetype
177      = "dashed") +
178    labs(title = "Distribution des valeurs EC50 bootstrap",
179      x = "EC50 bootstrap", y = "Fréquence") +
180    theme_minimal()
181
182  return(list(mean_ec10 = mean_ec10, ci_ec10 = ec10_ci,
183    mean_ec20 = mean_ec20, ci_ec20 = ec20_ci,
184    mean_ec50 = mean_ec50, ci_ec50 = ec50_ci, plot = p))
185}
186
187
188# Fonction permutation
189analyze_permutation <- function(data_long, num_permutations = 1000)
190  {
191    model <- aov(response ~ concentration, data = data_long)
192    obs_stat <- summary(model)[[1]]$`Pr(>F)`[1]
193
194    if (!is.numeric(obs_stat)) {
195      stop("La statistique observée n'est pas numérique.") }
196
197    perm_stats <- numeric(num_permutations)
198
199    for (i in 1:num_permutations) {
200      perm_data <- data_long %>%
201        mutate(response = sample(response, replace = FALSE))#response
202        et non Density
203      perm_model <- aov(response ~ concentration, data = perm_data)
204      perm_stat <- summary(perm_model)[[1]]$`Pr(>F)`[1]
205
206      if (is.numeric(perm_stat)) {
207        perm_stats[i] = perm_stat
208      } else {
209        perm_stats[i] = NA

```



```

248     chain[i + 1, ] <- current
249   }
250 }
251
252 acc_rate <- acc_rate / nchain
253 return(list(chain = chain, acc_rate = acc_rate))
254 }
```

Interface utilisateur : UI

```

1 #Fichier CSS:
2 sass(sass_file("style.scss"), output = "www/style.css")
3 ui <- fluentPage(
4   tags$head(
5     tags$link(href = "style.css", rel = "stylesheet", type = "text/
6       css")),
7   div(
8     class = "grid-container",
9     div(class = "header", header()),
10    div(class = "sidenav", tagList(navigation())))
11  ,
12  div(
13    class = "main",
14    uiOutput("progressIndicator"),
15    router_ui(
16      route("/", HomePage()),
17      route("analysis", analysisPage()),
18      route("permutation", permutationPage()),
19      route("bootstrap", bootstrapPage()),
20      route("bayesian", bayesianPage()),
21      route("Select", BayoyoPage()),
22      route("calcPuiss", PuissancePage()),
23      route("Essai201", Essai201Page()),
24      route("Essai202", Essai202Page()),
25      route("Essai203", Essai203Page()),
26      route("Essai205", Essai205Page()),
27      route("Essai206", Essai206Page()),
28      route("Essai208", Essai208Page()),
29      route("Essai210", Essai210Page()),
30      route("Essai211", Essai211Page()),
31      route("Essai216", Essai216Page()),
32      route("Essai218", Essai218Page()),
33      route("Essai213214", Essai213214Page()),
34      route("Essai221", Essai221Page()),
35      route("Essai222", Essai222Page()),
36      route("Essai223", Essai223Page()),
37      route("Essai225", Essai225Page()),
38      route("Essai226", Essai226Page()),
39      route("Essai227", Essai227Page()),
40      route("Essai232", Essai232Page()),
41      route("Essai239", Essai239Page()),
42      route("Essai245", Essai245Page()),
43      route("Essai246247", Essai246247Page()),
```

```

43         route("ESCORT", ESCORTPage())
44
45     )
46   ),
47   div(class = "footer", footer())
48 )
49 )

```

SERVER:

```

1 server <- function(input, output, session) {
2   router_server(root_page = "/")
3   global_data <- reactiveVal()
4   progress <- reactiveVal(0)
5   output$progressIndicator <- renderUI({
6     ProgressIndicator(label = "Processing data...", percentComplete
7       = progress(1))
8   })
9   ##  visualisation data/stats desc  ###
10  observeEvent(input$submit, {
11    req(input$submit)
12    df <- reactive({
13      isolate({
14        if (!is.null(input$file_analysis)) {
15          file_data <- read.table(input$file_analysis$datapath,
16            header = input$header_analysis, sep = input$sep_
17            analysis)
18          global_data(file_data)
19          return(file_data)
20        } else if (input$textdata != "") {
21          file_data <- read_delim(text(input$textdata), delim =
22            input$sep_analysis)
23          global_data(file_data)
24          return(file_data)
25        } else {
26          return(NULL)
27        }
28      })
29    })
30
31    output$uploaded_data <- renderDT({
32      df_data <- df()
33      if (!is.null(df_data)) {
34        datatable(df_data)
35      } else {
36        NULL
37      }
38    })
39  })
40  databb <- reactive({
41    req(input$submit)
42    isolate({
43      if (!is.null(input$file_analysis)) {

```

```

40     file_data <- readLines(input$file_analysis$datapath, header
41         = input$header_analysis, sep = input$sep_analysis)
42     global_data(file_data)
43     return(file_data)
44 } else if (input$textdata != "") {
45     file_data <- read_delim(text(input$textdata), delim = input$sep_analysis)
46     global_data(file_data)
47     return(file_data)
48 } else {
49     return(NULL)
50 }
51 })
52 # Visualisation des données
53 observeEvent(global_data(), {
54     df <- global_data()
55     output$plot_visualization <- renderPlotly({
56         req(df)
57         df_long <- df %>%
58             pivot_longer(cols = -Replicat, names_to = "concentration",
59                         values_to = "response") %>%
60             drop_na(response) %>%
61             mutate(concentration = factor(concentration)) %>%
62             mutate(response = as.numeric(response))
63         DataMean <- SummaryVisu(df_long, 'response', 'concentration')
64         visualization <- #ggplot(df_long, aes(x = as.numeric(
65             concentration), y = response)) +
66             #geom_point(size = 2, alpha = 0.7) +
67             #scale_x_log10() +
68             #labs(
69                 # x = paste0("concentration (", input$conc_unit, ")"),
70                 # y = "response"
71                 #) +
72                 #theme_minimal(base_size = 15) +
73                 #theme(axis.text = element_text(size = 14),
74                 #       axis.title = element_text(size = 16))
75         ggplot(df_long, aes(x = concentration, y = response, color =
76                 Replicat)) +
77             geom_point(size = 2, alpha = 0.7) +
78             labs(title = "Réplicats en fonction des concentrations",
79                 x = "concentration",
80                 y = "Réponse") +
81             theme_minimal() +
82             theme(axis.text.x = element_text(angle = 45, hjust = 1))
83
84         visualization <- visualization + geom_point(data = DataMean,
85             aes(x = concentration, y = Means), color = "black", size =
86             3, shape = 8) +
87             geom_line(data = DataMean, aes(x = concentration, y = Means,
88                 group = 1), color = "black", linewidth = 1)
89
90
91
92
93

```

```

84     ggplotly(visualization)
85   })
86 }
87 output$desc_stats <- renderDT({
88   df <- global_data()
89   if (!is.null(df)) {
90     df_long <- df %>%
91       pivot_longer(cols = -Replicat, names_to = "concentration",
92                     values_to = "response") %>%
93       drop_na(response) %>%
94       mutate(concentration = factor(concentration)) %>%
95       mutate(response = as.numeric(response))
96     DataMean <- SummaryVisu(df_long, 'response', 'concentration')
97     datatable(DataMean)
98   } else {
99     NULL
100   }
101 })
102 observeEvent(global_data(), {#ou input$file_analysis à la place de
103   global_data()
104   df <- global_data()
105
106   output$plot_box <- renderPlotly({
107     req(df)
108     df_long <- df %>%
109       pivot_longer(cols = -Replicat, names_to = "concentration",
110                     values_to = "response") %>%
111       drop_na(response) %>%
112       mutate(concentration = factor(concentration)) %>%
113       mutate(response = as.numeric(response))
114
115     boxP <- ggplot(df_long, aes(x = concentration, y = response,
116                               colour = concentration, fill = concentration)) +
117       geom_boxplot(alpha = 0.5, outlier.shape = NA) +
118       geom_jitter(position = position_jitter(0.2), size = 2, alpha
119                   = 0.7) +
120       stat_summary(fun = mean, geom = "point", shape = 8, size =
121                   3, color = "black", show.legend = FALSE) +
122       labs(title = "Boxplot",
123             x = "concentration",
124             y = "Réponse",
125             caption = "*moyenne") +
126       theme_minimal() +
127       theme(legend.position = "none", plot.caption = element_text(
128         hjust = 0, face = "italic"))
129
130     ggplotly(boxP)
131   })
132 }
133
134 #### Outliers/Transformation/effet solvant #####
135 observe({
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
927
928
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1256
1257
1258
1258
1259
1260
1261
1262
1263
1264
1265
1266
1266
1267
1268
1268
1269
1270
1271
1272
1273
1274
1275
1275
1276
1277
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1306
1307
1308
1308
1309
1310
1311
1312
1313
1314
1315
1315
1316
1317
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1325
1326
1327
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1335
1336
1337
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1345
1346
1347
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1355
1356
1357
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1365
1366
1367
1367
1368
1369
1369
1370
1371
1372
1373
1374
1374
1375
1376
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1384
1385
1386
1386
1387
1388
1388
1389
1390
1391
1392
1393
1393
1394
1395
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1404
1405
1406
1406
1407
1408
1408
1409
1410
1411
1412
1413
1413
1414
1415
1415
1416
1417
1417
1418
1419
1419
1420
1421
1422
1423
1424
1424
1425
1426
1426
1427
1428
1428
1429
1430
1431
1432
1433
1433
1434
1435
1435
1436
1437
1437
1438
1439
1439
1440
1441
1442
1443
1444
1444
1445
1446
1446
1447
1448
1448
1449
1450
1451
1452
1453
1453
1454
1455
1455
1456
1457
1457
1458
1459
1459
1460
1461
1462
1463
1464
1464
1465
1466
1466
1467
1468
1468
1469
1470
1471
1472
1473
1473
1474
1475
1475
1476
1477
1477
1478
1479
1479
1480
1481
1482
1483
1484
1484
1485
1486
1486
1487
1488
1488
1489
1490
1491
1492
1493
1493
1494
1495
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1504
1505
1506
1506
1507
1508
1508
1509
1510
1511
1512
1513
1513
1514
1515
1515
1516
1517
1517
1518
1519
1519
1520
1521
1522
1523
1524
1524
1525
1526
1526
1527
1528
1528
1529
1530
1531
1532
1533
1534
1534
1535
1536
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1544
1545
1546
1546
1547
1548
1548
1549
1550
1551
1552
1553
1553
1554
1555
1555
1556
1557
1557
1558
1559
1559
1560
1561
1562
1563
1564
1564
1565
1566
1566
1567
1568
1568
1569
1570
1571
1572
1573
1573
1574
1575
1575
1576
1577
1577
1578
1579
1579
1580
1581
1582
1583
1584
1584
1585
1586
1586
1587
1588
1588
1589
1590
1591
1592
1593
1593
1594
1595
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1604
1605
1606
1606
1607
1608
1608
1609
1610
1611
1612
1613
1613
1614
1615
1615
1616
1617
1617
1618
1619
1619
1620
1621
1622
1623
1624
1624
1625
1626
1626
1627
1628
1628
1629
1630
1631
1632
1633
1633
1634
1635
1635
1636
1637
1637
1638
1639
1639
1640
1641
1642
1643
1644
1644
1645
1646
1646
1647
1648
1648
1649
1650
1651
1652
1653
1653
1654
1655
1655
1656
1657
1657
1658
1659
1659
1660
1661
1662
1663
1664
1664
1665
1666
1666
1667
1668
1668
1669
1670
1671
1672
1673
1673
1674
1675
1675
1676
1677
1677
1678
1679
1679
1680
1681
1682
1683
1684
1684
1685
1686
1686
1687
1688
1688
1689
1690
1691
1692
1693
1693
1694
1695
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1704
1705
1706
1706
1707
1708
1708
1709
1710
1711
1712
1713
1713
1714
1715
1715
1716
1717
1717
1718
1719
1719
1720
1721
1722
1723
1724
1724
1725
1726
1726
1727
1728
1728
1729
1730
1731
1732
1733
1733
1734
1735
1735
1736
1737
1737
1738
1739
1739
1740
1741
1742
1743
1744
1744
1745
1746
1746
1747
1748
1748
1749
1750
1751
1752
1753
1753
1754
1755
1755
1756
1757
1757
1758
1759
1759
1760
1761
1762
1763
1764
1764
1765
1766
1766
1767
1768
1768
1769
1770
1771
1772
1773
1773
1774
1775
1775
1776
1777
1777
1778
1779
1779
1780
1781
1782
1783
1784
1784
1785
1786
1786
1787
1788
1788
1789
1790
1791
1792
1793
1793
1794
1795
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1804
1805
1806
1806
1807
1808
1808
1809
1810
1811
1812
1813
1813
1814
1815
1815
1816
1817
1817
1818
1819
1819
1820
1821
1822
1823
1824
1824
1825
1826
1826
1827
1828
1828
1829
1830
1831
1832
1833
1833
1834
1835
1835
1836
1837
1837
1838
1839
1839
1840
1841
1842
1843
1844
1844
1845
1846
1846
1847
1848
1848
1849
1850
1851
1852
1853
1853
1854
1855
1855
1856
1857
1857
1858
1859
1859
1860
1861
1862
1863
1864
1864
1865
1866
1866
1867
1868
1868
1869
1870
1871
1872
1873
1873
1874
1875
1875
1876
1877
1877
1878
1879
1879
1880
1881
1882
1883
1884
1884
1885
1886
1886
1887
1888
1888
1889
1890
1891
1892
1893
1893
1894
1895
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1904
1905
1906
1906
1907
1908
1908
1909
1910
1911
1912
1913
1913
1914
1915
1915
1916
1917
1917
1918
1919
1919
1920
1921
1922
1923
1924
1924
1925
1926
1926
1927
1928
1928
1929
1930
1931
1932
1933
1933
1934
1935
1935
1936
1937
1937
1938
1939
1939
1940
1941
1942
1943
1944
1944
1945
1946
1946
1947
1948
1948
1949
1950
1951
1952
1953
1953
1954
1955
1955
1956
1957
1957
1958
1959
1959
1960
1961
1962
1963
1964
1964
1965
1966
1966
1967
1968
1968
1969
1970
1971
1972
1973
1973
1974
1975
1975
1976
1977
1977
1978
1979
1979
1980
1981
1982
1983
1984
1984
1985
1986
1986
1987
1988
1988
1989
1990
1991
1992
1993
1993
1994
1995
1995
1996
1997
1997
1998
1999
1999
2000
2001
2002
2003
2004
2004
2005
2006
2006
2007
2008
2008
2009
2010
2011
2012
2013
2013
2014
2015
2015
2016
2017
2017
2018
2019
2019
2020
2021
2022
2023
2024
2024
2025
2026
2026
2027
2028
2028
2029
2030
2031
2032
2033
2033
2034
2035
2035
2036
2037
2037
2038
2039
2039
2040
2041
2042
2043
2044
2044
2045
2046
2046
2047
2048
2048
2049
2050
2051
2052
2053
2053
2054
2055
2055
2056
2057
2057
2058
2059
2059
2060
2061
2062
2063
2064
2064
2065
2066
2066
2067
2068
2068
2069
2070
2071

```

```

129  if (!is.null(input$apply_transformation) && input$apply_
130    transformation) {
131    output$apply_transformation_menu <- renderUI({
132      selectInput("transformation_type", "Choose transformation:",
133        choices = c("Aucune" = "Aucune",
134                     "Reduce variance (Box-Cox)" = "boxcox",
135                     "Normalize distribution (Logarithmic)" = "log",
136                     "Stabilize variance (Square root)" = "sqrt"),
137        selected = "Aucune")
138    })
139  } else {
140    output$apply_transformation_menu <- renderUI({
141      NULL
142    })
143  }
144
145  transformed_data <- reactive({
146    req(global_data())
147    df2 <- global_data()
148
149    # Gestion des outliers
150    if (!is.null(input$manage_outliers) && input$manage_outliers) {
151      numeric_cols <- sapply(df2, is.numeric)
152      df2_numeric <- df2[, numeric_cols]
153      Q1 <- apply(df2_numeric, 2, quantile, 0.25, na.rm = TRUE)
154      Q3 <- apply(df2_numeric, 2, quantile, 0.75, na.rm = TRUE)
155      IQR <- Q3 - Q1
156      outlier_filter <- apply(df2_numeric, 1, function(row) {
157        any(row < Q1 - 1.5 * IQR | row > Q3 + 1.5 * IQR)
158      })
159
160      # Affichage des indices des lignes identifiées comme outliers
161      outliers_indices <- which(outlier_filter)
162      print(outliers_indices)
163
164      df2_outliers <- df2[outliers_indices, ] # Sélection des
165      # outliers
166      df2 <- df2[!outlier_filter, ] # Filtrer les outliers
167    }
168
169    # Application des transformations
170    if (!is.null(input$apply_transformation) && input$apply_
171      transformation) {
172      if (input$transformation_type == "boxcox")
173        {df2 <- as.data.frame(lapply(df2, function(col) {
174          if (is.numeric(col) && all(col > 0, na.rm = TRUE)) { # Pour
175            verifier si les données sont toutes positives
176            boxcox_result <- MASS::boxcox(col ~ 1, lambda = seq(-5,

```

```

174         5, 0.1))
175     if (length(boxcox_result$y) == length(col)) {
176         return(boxcox_result$y) # envoi la transformation
177         uniquement si les longueurs correspondent
178     } else {
179         warning("La transformation de Box-Cox a échoué pour
180         une colonne.")
181         return(col)
182     }
183 }
184
185 else if (input$transformation_type == "log") {
186     df2 <- as.data.frame(lapply(df2, function(col) {
187         if (is.numeric(col)) {
188             return(log(col + 1))
189         } else {
190             return(col)
191         }
192     }))
193 } else if (input$transformation_type == "sqrt") {
194     df2 <- as.data.frame(lapply(df2, function(col) {
195         if (is.numeric(col)) {
196             return(sqrt(col))
197         } else {
198             return(col)
199         }
200     }))
201 }
202
203 # Affichage des outliers dans un tableau (si disponible)
204 if (exists("df2_outliers")) {
205     output$outliers_table <- renderTable({
206         df2_outliers
207     })
208 }
209
210 df2
211 }
212 output$plot_box2 <- renderPlotly({
213     req(transformed_data())
214
215     df_long2 <- transformed_data() %>%
216         pivot_longer(cols = -Replicat, names_to = "concentration",
217             values_to = "response") %>%
218         drop_na(response) %>%
219         mutate(concentration = factor(concentration)) %>%
220         mutate(response = as.numeric(response))
221
222     boxP2 <- ggplot(df_long2, aes(x = concentration, y = response,

```

```

222     colour = concentration, fill = concentration)) +
223     geom_boxplot(alpha = 0.5, outlier.shape = NA) +
224     geom_jitter(position = position_jitter(0.2), size = 2, alpha =
225     0.7) +
226     stat_summary(fun = mean, geom = "point", shape = 8, size = 3,
227     color = "black", show.legend = FALSE) +
228     labs(title = "Boxplot apres transformation",
229     x = "concentration",
230     y = "Response",
231     caption = "*:mean") +
232     theme_minimal() +
233     theme(legend.position = "none", plot.caption = element_text(
234     hjust = 0, face = "italic"))

235 ggplotly(boxP2)
236 }

237 output$dataTF <- renderTable(
238   req(transformed_data())
239   as.data.frame(transformed_data())
240 }

241 output$effect_solvantValue <- renderText({
242   if (!is.null(input$effect_solvant) && input$effect_solvant) {
243     "Solvent effect is activated."
244   } else {
245     "Solvent effect is deactivated."
246   }
247 })

248 output$manage_outliersValue <- renderText({
249   if (!is.null(input$manage_outliers) && input$manage_outliers) {
250     "Outlier handling is activated."
251   } else {
252     "Outlier handling is deactivated."
253   }
254 })

255 output$apply_transformationValue <- renderText({
256   if (!is.null(input$apply_transformation) && input$apply_
257     transformation) {
258     "Transformation is applied."
259   } else {
260     "Transformation is deactivated."
261   }
262 })

263

264

265 # Application de l'effet solvant
266 # Fonction réactive pour appliquer l'effet solvant
267 effetSov_data <- reactive({

```

```

269     req(input$effect_solvant)
270     req(global_data())
271
272     dfSolv <- global_data()
273
274     # Vérification colonnes
275     if ("contrôle" %in% names(dfSolv) && "contrôle_solvant" %in%
276         names(dfSolv)) {
277
278         # Test de Student
279         test_result <- t.test(dfSolv$contrôle, dfSolv$contrôle_solvant
280             )
281
282         # résultat du test
283         output$test_result <- renderPrint({
284             test_result
285         })
286
287         # Si test non significatif, concaténer les colonnes
288         if (test_result$p.value > 0.05)
289
290         {
291             # dfSolv$concentration[df$concentration %in% c("control", "
292             # control solvent")] <- "0"
293             dfSolv <- dfSolv %>%
294                 mutate(concentration = ifelse(concentration %in% c("
295                 "contrôle", "contrôle_solvant"), "0", concentration))
296             cat("Effet_solvant non significatif, colonnes combinées en "
297                 '0'.\n')
298         } else {
299             cat("Effet_solvant significatif, ne pas combiner les "
300                 colonnes.\n")
301         }
302
303         # dfSolv$'0' <- c(dfSolv$contrôle, dfSolv$contrôle_solvant)
304         # Supprimer les anciennes colonnes
305         dfSolv <- dfSolv %>% select(-contrôle, -contrôle_solvant)
306
307         # Mise à jour de la table de données
308         output$dataTable <- renderDT({
309             datatable(dfSolv)
310         })
311
312     } else {
313         output$test_result <- renderText("Effet_solvant désactivé.")
314         output$dataTable <- renderDT({
315             datatable(dfSolv)
316         })
317     }
318
319     return(dfSolv)
320 }

```

```

315
316 # Mise à jour du boxplot en fonction des données modifiées
317 observeEvent(effetSov_data(), {
318   df <- effetSov_data()
319
320   output$plot_box <- renderPlotly({
321     req(df)
322
323     df_long <- df %>%
324       pivot_longer(cols = -Replicat, names_to = "concentration",
325                     values_to = "response") %>%
326       drop_na(response) %>%
327       mutate(concentration = factor(concentration)) %>%
328       mutate(response = as.numeric(response))
329
330     boxP <- ggplot(df_long, aes(x = concentration, y = response,
331                       colour = concentration, fill = concentration)) +
332       geom_boxplot(alpha = 0.5, outlier.shape = NA) +
333       geom_jitter(position = position_jitter(0.2), size = 2, alpha
334                     = 0.7) +
335       stat_summary(fun = mean, geom = "point", shape = 8, size =
336                     3, color = "black", show.legend = FALSE) +
337       labs(title = "Boxplot",
338             x = "Concentration",
339             y = "Réponse",
340             caption = "*:moyenne") +
341       theme_minimal() +
342       theme(legend.position = "none", plot.caption = element_text(
343                     hjust = 0, face = "italic"))
344
345     ggplotly(boxP)
346   })
347 }
348
349
350
351 # Modelisation
352 ##  ##Methodes Bayesiennes#
353 #####Selection modele(cf method rapport)
354 observeEvent(input$run_analysis, {
355   req(input$file)
356
357   # Chargement des données si on utilise pas les données global
358   # du début(anayle independante)
359   data <- read.csv(input$file$datapath, header = FALSE)
360   colnames(data) <- c("Replicat", "Control", "Control_Solvent",
361                     "0.1", "0.15", "0.23", "0.34", "0.51", "0.76", "1.14")
362   data <- data[,-c(2,3)] # Suppression des colonnes Control et
363   Control_Solvent
364   data <- melt(data, id.vars = "Replicat", variable.name =
365                 "concentration", value.name = "response")
366   data$concentration <- as.numeric(as.character(data$concentration))

```

```

357
358     # Préparation des données pour l'analyse
359     data <- data[!is.na(data$response), ]
360     data_aggregated <- aggregate(response ~ concentration, data =
361         data, FUN = mean)
362
363     # Fonctions de vraisemblance
364     logLikelihood_loglogistic <- function(params, concentration,
365         response) {
366         alpha <- params[1]
367         beta <- params[2]
368         predicted <- 1 / (1 + (concentration / beta) ^ (-alpha))
369         ll <- sum(dlogis(response, location = predicted, scale =
370             alpha, log = TRUE))
371         return(ll)
372     }
373
374     logLikelihood_lognormal <- function(params, concentration,
375         response) {
376         meanlog <- params[1]
377         sdlog <- params[2]
378         predicted <- plnorm(concentration, meanlog, sdlog)
379         ll <- sum(dlnorm(response, meanlog = predicted, sdlog =
380             sdlog, log = TRUE))
381         return(ll)
382     }
383
384     logLikelihood_weibull <- function(params, concentration,
385         response) {
386         shape <- params[1]
387         scale <- params[2]
388         predicted <- pweibull(concentration, shape = shape, scale =
389             scale)
390         ll <- sum(dweibull(response, shape = predicted, scale =
391             scale, log = TRUE))
392         return(ll)
393     }
394     prior_uniform <- function(params) {
395         if (all(params > 0 & params < 10)) {
396             return(1)
397         } else {
398             return(0)
399         }
400     }
401
402     posterior_unnormalized <- function(params, logLikelihoodFunc,
403         priorFunc, concentration, response) {
404         likelihood <- exp(logLikelihoodFunc(params, concentration,
405             response))
406         prior <- priorFunc(params)
407
408         if (is.na(likelihood) || is.na(prior) || likelihood == 0 ||
409             prior == 0) {

```

```

398         return(0)
399     }
400     return(likelihood * prior)
401 }
402 metropolis_hastings <- function(logLikelihoodFunc, priorFunc,
403   concentration, response, start, iterations, proposal_sd) {
404   current_params <- start
405   chain <- matrix(NA, nrow = iterations, ncol = length(start))
406   acceptance <- 0
407   for (i in 1:iterations) {
408     proposed_params <- current_params + rnorm(length(start),
409       mean = 0, sd = proposal_sd)
410     current_posterior <- posterior_unnormalized(current_params
411       , logLikelihoodFunc, priorFunc, concentration, response
412       )
413     proposed_posterior <- posterior_unnormalized(proposed_
414       params, logLikelihoodFunc, priorFunc, concentration,
415       response)
416     acceptance_ratio <- proposed_posterior / current_posterior
417     if (!is.na(acceptance_ratio) && acceptance_ratio > 0 &&
418       runif(1) < acceptance_ratio) {
419       current_params <- proposed_params
420       acceptance <- acceptance + 1
421     }
422     chain[i, ] <- current_params
423   }
424   acceptance_rate <- acceptance / iterations
425   list(chain = chain, acceptance_rate = acceptance_rate)
426 }
427 # Metropolis-Hastings
428 mh_results_loglogistic <- metropolis_hastings(
429   logLikelihoodFunc = logLikelihood_loglogistic,
430   priorFunc = prior_uniform,
431   concentration = data_aggregated$concentration,
432   response = data_aggregated$response,
433   start = c(5, 5),
434   iterations = input$iterations,
435   proposal_sd = c(0.7, 1.0)
436 )
437
438 mh_results_lognormal <- metropolis_hastings(
439   logLikelihoodFunc = logLikelihood_lognormal,
440   priorFunc = prior_uniform,
441   concentration = data_aggregated$concentration,
442   response = data_aggregated$response,
443   start = c(5, 5),
444   iterations = input$iterations,
445   proposal_sd = c(0.7, 1.0)
446 )
447
448 mh_results_weibull <- metropolis_hastings(
449   logLikelihoodFunc = logLikelihood_weibull,

```

```

443     priorFunc = prior_uniform,
444     concentration = data_aggregated$concentration,
445     response = data_aggregated$response,
446     start = c(5, 5),
447     iterations = input$iterations,
448     proposal_sd = c(0.7, 1.0)
449   )
450
451   burn_in <- input$burn_in
452
453   post_burn_in_loglogistic <- mh_results_loglogistic$chain[-(1:
454     burn_in), ]
455   post_burn_in_lognormal <- mh_results_lognormal$chain[-(1:burn_
456     in), ]
457   post_burn_in_weibull <- mh_results_weibull$chain[-(1:burn_in),
458     ]
459
460   marginal_likelihood_loglogistic <- mean(apply(post_burn_in_
461     loglogistic, 1, function(params) {
462       posterior_unnormalized(params, logLikelihood_loglogistic,
463         prior_uniform, data_aggregated$concentration, data_
464         aggregated$response)
465     }))
466
467   marginal_likelihood_lognormal <- mean(apply(post_burn_in_
468     lognormal, 1, function(params) {
469       posterior_unnormalized(params, logLikelihood_lognormal,
470         prior_uniform, data_aggregated$concentration, data_
471         aggregated$response)
472     }))
473
474   marginal_likelihood_weibull <- mean(apply(post_burn_in_weibull
475     , 1, function(params) {
476       posterior_unnormalized(params, logLikelihood_weibull, prior_
477         uniform, data_aggregated$concentration, data_aggregated$_
478         response)
479     }))
480
481   BF_loglogistic_lognormal <- marginal_likelihood_loglogistic /
482     marginal_likelihood_lognormal
483   BF_loglogistic_weibull <- marginal_likelihood_loglogistic /
484     marginal_likelihood_weibull
485   BF_lognormal_weibull <- marginal_likelihood_lognormal /
486     marginal_likelihood_weibull
487
488   output$bayes_factors <- renderPrint({
489     cat("BayesFactors:\n")
490     cat("Loglogistic vs Lognormal:", BF_loglogistic_lognormal, "
491       \n")
492     cat("Loglogistic vs Weibull:", BF_loglogistic_weibull, "\n")
493     cat("Lognormal vs Weibull:", BF_lognormal_weibull, "\n")
494   })

```



```

516     df <- global_data()
517     output$column_select <- renderUI({
518       colnames <- names(df)
519       tagList(
520         selectInput("column1", "Select Column 1:", choices =
521           colnames),
522         selectInput("column2", "Select Column 2:", choices =
523           colnames)
524       )
525     })
526
527     values <- reactiveValues()
528
529     observe({
530       req(input$column1, input$column2)
531       col1 <- input$column1
532       col2 <- input$column2
533       if (!is.numeric(df[[col1]]) || !is.numeric(df[[col2]])) {
534         return(NULL)
535       }
536
537       if (input$design == "single") {
538         t_val <- t.test(df[[col1]], df[[col2]], paired = TRUE)$
539         statistic
540         n1 <- nrow(df)
541         result <- ttest.tstat(t = t_val, n1 = n1)
542       } else {
543         t_val <- t.test(df[[col1]], df[[col2]], paired = FALSE)$
544         statistic
545         n1 <- length(df[[col1]])
546         n2 <- length(df[[col2]])
547         result <- ttest.tstat(t = t_val, n1 = n1, n2 = n2)
548       }
549       values$BF10 <- exp(result[['bf']])
550       values$BF01 <- 1 / exp(result[['bf']])
551       values$pOdds <- input$pH0 / (1 - input$pH0)
552     })
553
554     output$bfText <- renderUI({
555       if (is.null(values$BF10)) {
556         shiny:::flushReact()
557         return()
558       }
559
560       if (values$BF01 > 1) {
561         p(withMathJax(sprintf("The Bayes factor for the null is
562           \\\(\text{B}_{0/1}\) = %.2f", values$BF01)))
563       } else if (values$BF10 > 1) {
564         p(withMathJax(sprintf("The Bayes factor for the alternative
565           is \\\(\text{B}_{1/0}\) = %.2f", values$BF10)))
566       }
567     })

```

```

562
563   output$plotOutput <- renderPlot({
564     if (is.null(values$BF10)) return(NULL)
565
566     bf_data <- data.frame(
567       Hypothesis = c("H0", "H1"),
568       BayesFactor = c(values$BF01, values$BF10)
569     )
570
571     ggplot(bf_data, aes(x = Hypothesis, y = BayesFactor, fill =
572       Hypothesis)) +
573       geom_bar(stat = "identity") +
574       theme_minimal() +
575       labs(title = "Bayes Factor Comparison", x = "Hypothesis", y
576         = "Bayes Factor") +
577       scale_fill_manual(values = c("H0" = "skyblue", "H1" = "orange"))
578   })
579
580   output$description <- renderUI({
581     if (is.null(values$BF10)) {
582       shiny:::flushReact()
583       return()
584     }
585
586     if (values$BF01 > 1) {
587       p(withMathJax(sprintf("The Bayes factor for the null is
588         \\\(\text{B}\_{0/1}\) = %.2f", values$BF01)))
589     } else if (values$BF10 > 1) {
590       p(withMathJax(sprintf("The Bayes factor for the alternative
591         is \\\(\text{B}\_{1/0}\) = %.2f", values$BF10)))
592     }
593
594     if (values$BF10 > 1) {
595       p(withMathJax(sprintf("This means that the observed data are
596         approximately %.2f times more likely under \\\(\mathcal{H}\_{1}\)
597         than under \\\(\mathcal{H}\_{0}\)", values$BF10)))
598     } else {
599       p(withMathJax(sprintf("This means that the observed data are
600         approximately %.2f times more likely under \\\(\mathcal{H}\_{0}\)
601         than under \\\(\mathcal{H}\_{1}\)", values$BF01)))
602     }
603
604   output$postNull <- renderUI({
605     p(withMathJax(sprintf("The posterior probability for \\\(
606       \mathcal{H}\_{0}\) is %.4f", (values$BF01 * values$pOdds) / (1
607       + (values$BF01 * values$pOdds)))))
608   })
609
610   output$postAlt <- renderUI({
611     p(withMathJax(sprintf("The posterior probability for \\\(

```



```

693     }
694     chain <- chain[-(1:burnin), ]
695     EC10 <- (log(0.1) - chain[, "beta0"]) / chain[, "beta1"]
696     EC20 <- (log(0.2) - chain[, "beta0"]) / chain[, "beta1"]
697     EC50 <- (log(0.5) - chain[, "beta0"]) / chain[, "beta1"]
698
699     ec10_mean <- mean(EC10)
700     ec10_sd <- sd(EC10)
701     ec20_mean <- mean(EC20)
702     ec20_sd <- sd(EC20)
703     ec50_mean <- mean(EC50)
704     ec50_sd <- sd(EC50)
705
706     interpretation <- paste(
707         "L'analyse a été réalisée avec succès.<br><br>",
708         "Voici un résumé des résultats:<br>",
709         "- Taux d'acceptation des propositions:<span style='color: red;'>", round(analysissB()$acc_rate, 3), "</span><br>",
710         "- Estimation de l'EC10:<span style='color:red;'>", round(
711             ec10_mean, 3), "</span> (écart-type:<span style='color: red;'>", round(ec10_sd, 3), "</span>)<br>",
712         "- Estimation de l'EC20:<span style='color:red;'>", round(
713             ec20_mean, 3), "</span> (écart-type:<span style='color: red;'>", round(ec20_sd, 3), "</span>)<br>",
714         "- Estimation de l'EC50:<span style='color:red;'>", round(
715             ec50_mean, 3), "</span> (écart-type:<span style='color: red;'>", round(ec50_sd, 3), "</span>)<br><br>",
716         "L'estimation des EC10, EC20 et EC50 (concentrations efficaces pour 10%, 20% et 50% des sujets) signifie que ces doses sont celles auxquelles on s'attend à ce que 10%, 20% et 50% des animaux exposés montrent une réponse spécifique.<br>",
717         "Ces estimations sont accompagnées d'une incertitude représentée par l'écart-type.<br>",
718         "Plus l'écart-type est faible, plus l'estimation est précise ."
719     )
720     HTML(interpretation)
721 }
722
723 # Test d'hypotheses sur ECx et calcul de FB
724 observeEvent(input$test_hypothesis, {
725     req(values$EC10, values$EC20, values$EC50)
726
727     # Define the hypotheses
728     hyp_EC10 <- input$hyp_ec10
729     hyp_EC20 <- input$hyp_ec20
730     hyp_EC50 <- input$hyp_ec50
731
732     # proportion d'ech pour hypotheses
733     bf_EC10 <- mean(values$EC10 > hyp_EC10)

```

```

732 bf_EC20 <- mean(values$EC20 > hyp_EC20)
733 bf_EC50 <- mean(values$EC50 > hyp_EC50)
734
735 # résultats (dans valeurs réactives)
736 values$BF_EC10 <- bf_EC10
737 values$BF_EC20 <- bf_EC20
738 values$BF_EC50 <- bf_EC50
739 }
740
741 # Facteur de Bayes (FB)
742 output$bayes_factors <- renderUI({
743   req(values$BF_EC10, values$BF_EC20, values$BF_EC50)
744   HTML(paste(
745     "Facteur de Bayes pour EC10 > ", input$hyp_ec10, " : <span style='color:red;'>", round(values$BF_EC10, 3), "</span><br>",
746     "Facteur de Bayes pour EC20 > ", input$hyp_ec20, " : <span style='color:red;'>", round(values$BF_EC20, 3), "</span><br>",
747     "Facteur de Bayes pour EC50 > ", input$hyp_ec50, " : <span style='color:red;'>", round(values$BF_EC50, 3), "</span>"
748   )))
749 })
750
751 output$interpretation <- renderUI({
752   req(values$BF_EC10, values$BF_EC20, values$BF_EC50)
753
754   interpretation <- paste(
755     "Interprétation des résultats : <br>",
756     "- Un facteur de Bayes proche de 1 suggère que les données ne favorisent pas particulièrement une hypothèse par rapport à l'autre. <br>",
757     "- Un facteur de Bayes > 1 indique que les données sont plus probables sous l'hypothèse alternative (par exemple, ECx > hypothèse spécifiée). <br>",
758     "- Un facteur de Bayes < 1 indique que les données sont plus probables sous l'hypothèse nulle."
759   )
760   HTML(interpretation)
761 })
762
763 ##### # Resampling# #####
764 # Permutation.R(actionButton("run_permutation")+fonction permut
765
766 observeEvent(input$run_permutation, {
767   df <- global_data()
768   if (ncol(df) < 2) {
769     output$perm_result <- renderUI({
770       p("The data must have at least two columns for the permutation test.")
771     })
772     return()
773   }
774   df <- df[, 1]
775   df$perm <- sample(df$perm)
776   df$perm <- as.factor(df$perm)
777   output$perm_result <- renderTable(df)
778 })
779
780 
```

```

773  }
774
775  df_long <- df %>%
776    pivot_longer(cols = -Replicat, names_to = "concentration",
777      values_to = "response") %>%
778    drop_na(response) %>%
779    mutate(concentration = factor(concentration))%>%
780    mutate(response <- as.numeric(response))
781
782  #df_long <- df %>%
783  #pivot_longer(cols = -Replicat, names_to = "concentration",
784  #  values_to = "response") %>%
785  #drop_na(response) %>%
786  #mutate(concentration = suppressWarnings(as.numeric(as.
787  #  character(concentration))),
788  #        response = suppressWarnings(as.numeric(response)))
789
790  analysis_results <- analyze_permutation(df_long)
791  output$results <- renderDT({
792    analysis_results$result
793  })
794
795
796  output$permutation_data <- renderDT({
797    analysis_results$permutations
798  })
799
800
801 # Bootstrap.R(actionButton("run_bootstrap")+fonction boots
802 observeEvent(input$run_bootstrap, {
803   df <- global_data()
804   if (ncol(df) < 1) {
805     output$bootstrap_result <- renderUI({
806       p("The data must have at least one column for the bootstrap
807       test.")
808     })
809     return()
810   }
811
812   df_long <- df %>%
813     pivot_longer(cols = -Replicat, names_to = "concentration",
814       values_to = "response") %>%
815     drop_na(response) %>%
816     mutate(concentration = factor(concentration))%>%
817     mutate(response <- as.numeric(response))
818
819   tryCatch({
820     analysis_results <- analyze_bootstrap(df_long)
821     output$bootstrap_result <- renderPrint({

```

```

820     paste("MeanEC10:", analysis_results$mean_ec10, "\n95%CIEC10:", analysis_results$ci_ec10[1], "-", analysis_results$ci_ec10[2],
821         "\nMeanEC20:", analysis_results$mean_ec20, "\n95%CIEC20:", analysis_results$ci_ec20[1], "-", analysis_results$ci_ec20[2],
822         "\nMeanEC50:", analysis_results$mean_ec50, "\n95%CIEC50:", analysis_results$ci_ec50[1], "-", analysis_results$ci_ec50[2])
823     })
824
825     output$bootstrap_plot <- renderPlot({
826         print(analysis_results$plot)
827     })
828     }, error = function(e) {
829         output$bootstrap_result <- renderPrint({
830             paste("Error:", e$message)
831         })
832     })
833 }
834
835 ##### # NOEC / LOEC #
836 # Dunnett #####
837
838 observeEvent(input$run_dunnett, {
839     df <- global_data()
840     if (ncol(df) < 2) {
841         output$dunnett_result <- renderUI({
842             p("The data must have at least two columns for the Dunnnett test.")
843         })
844         return()
845     }
846
847     analysis_results <- analyze_dunnett(df)
848     output$dunnett_result <- renderPrint({
849         paste("ANOVASummary:", analysis_results$summary_aov,
850             "\nSignificance:", analysis_results$significant,
851             "\nNormality:", analysis_results$normality,
852             "\nHomoscedasticity:", analysis_results$homoscedasticity
853             ,
854             "\nDunnnett Test:", analysis_results$summary_dunnett)
855     })
856
857     output$dunnett_plot <- renderPlot({
858         plot(analysis_results$Res_dunnett)
859     })
860
861
862 ##### ANOVA #####
863

```

```

864 observeEvent(input$run_anova, {
865   df <- global_data()
866   if (ncol(df) < 2) {
867     output$anova_result <- renderUI({
868       p("The data must have at least two columns for the ANOVA test.")
869     })
870     return()
871   }
872
873   analysis_results <- analyze_anova(df)
874   output$anova_result <- renderPrint({
875     paste("ANOVA Summary:", analysis_results$summary_aov,
876           "\nSignificance:", analysis_results$significant,
877           "\nNormality:", analysis_results$normality,
878           "\nHomoscedasticity:", analysis_results$homoscedasticity
879           )
880   })
881
882
883 #####      #Puissance#    ###
884 observeEvent(global_data(), {
885   df <- global_data()
886   output$column_select2 <- renderUI({
887     colnames <- names(df)
888     tagList(
889       updateSelectInput(session, "groupColumn_t", choices =
890         colnames),
891       updateSelectInput(session, "valueColumn_t", choices =
892         colnames),
893       updateSelectInput(session, "groupColumn_a", choices =
894         colnames),
895       updateSelectInput(session, "valueColumn_a", choices =
896         colnames),
897       updateSelectInput(session, "doseColumn_l", choices =
898         colnames),
899       updateSelectInput(session, "responseColumn_l", choices =
900         colnames)
901     )
902   })
903
904
905 # Calcul de la puissance et visualisation
906 output$powerPlot <- renderPlot({
907   req(global_data())
908
909   if (input$method == "t-test") {
910     req(input$groupColumn_t, input$valueColumn_t)
911     group_data <- df[, input$groupColumn_t]
912     value_data <- df[, input$valueColumn_t]
913     t_test <- t.test(value_data ~ group_data)
914     effect_size <- t_test$estimate / sd(value_data)
915   }
916 }

```

```

908     n <- length(value_data)
909     power_result <- pwr.t.test(n = n/2, d = effect_size, sig.
910       level = input$alphaLevel_t, power = NULL)
911     plot(power_result)
912   } else if (input$method == "ANOVA") {
913     req(input$groupColumn_a, input$valueColumn_a)
914     group_data <- df[, input$groupColumn_a]
915     value_data <- df[, input$valueColumn_a]
916     aov_model <- aov(value_data ~ group_data)
917     summary_aov <- summary(aov_model)
918     f_value <- summary_aov[[1]][["F_value"]][1]
919     n <- length(value_data) / length(unique(group_data))
920     power_result <- pwr.anova.test(k = length(unique(group_data))
921       ), n = n, f = f_value, sig.level = input$alphaLevel_a,
922       power = NULL)
923     plot(power_result)
924   } else if (input$method == "Régression Log-Logistique") {
925     req(input$doseColumn_l, input$responseColumn_l)
926     dose_data <- df[, input$doseColumn_l]
927     response_data <- df[, input$responseColumn_l]
928     ll_model <- drm(response_data ~ dose_data, fct = LL.4())
929     plot(ll_model, type = "all")
930   }
931 }
932
933 output$powerResult <- renderPrint({
934   req(global_data())
935
936   if (input$method == "t-test") {
937     req(input$groupColumn_t, input$valueColumn_t)
938     group_data <- df[, input$groupColumn_t]
939     value_data <- df[, input$valueColumn_t]
940     t_test <- t.test(value_data ~ group_data)
941     effect_size <- t_test$estimate / sd(value_data)
942     n <- length(value_data)
943     power_result <- pwr.t.test(n = n/2, d = effect_size, sig.
944       level = input$alphaLevel_t, power = NULL)
945     print(power_result)
946   } else if (input$method == "ANOVA") {
947     req(input$groupColumn_a, input$valueColumn_a)
948     group_data <- df[, input$groupColumn_a]
949     value_data <- df[, input$valueColumn_a]
950     aov_model <- aov(value_data ~ group_data)
951     summary_aov <- summary(aov_model)
952     f_value <- summary_aov[[1]][["F_value"]][1]
953     n <- length(value_data) / length(unique(group_data))
954     power_result <- pwr.anova.test(k = length(unique(group_data))
955       ), n = n, f = f_value, sig.level = input$alphaLevel_a,
956       power = NULL)
957     print(power_result)
958   } else if (input$method == "Régression Log-Logistique") {
959     req(input$doseColumn_l, input$responseColumn_l)

```

```

954     dose_data <- df[, input$doseColumn_1]
955     response_data <- df[, input$responseColumn_1]
956     ll_model <- drm(response_data ~ dose_data, fct = LL.4())
957     summary(ll_model)
958   }
959 }
960 }
961
962 #####      # Essaies standardisés # #####
963 #201#
964
965 intest_type201 <- reactive({ 'Essai201' })
966
967 validateFile <- function(filename) {
968   extFile <- tools::file_ext(filename)
969   validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
970     sont acceptés."))
971 }
972
973 filedatal201 <- eventReactive(input$buttonRunStep201, {
974   if(intest_type201() == 'Essai201') {
975     req(input$datafile_ESSAI201)
976     validateFile(input$datafile_ESSAI201)
977     ff <- input$datafile_ESSAI201
978     read.csv(file = ff$datapath, header = TRUE)
979   }
980 })
981
982 output$rawdata201 <- DT::renderDataTable({
983   filedatal201()
984 })
985
986 fitmodel201 <- eventReactive(input$buttonRunStep201, {
987   data <- filedatal201()
988   if(intest_type201() == 'Essai201') {
989     if(input$model_ESSAI201 == 'll2') {
990       fit <- drm(response ~ as.numeric(concentration), data = data
991                   , fct = LL.2())
992     } else if(input$model_ESSAI201 == 'll3') {
993       fit <- drm(response ~ as.numeric(concentration), data = data
994                   , fct = LL.3())
995     } else if(input$model_ESSAI201 == 'll4') {
996       fit <- drm(response ~ as.numeric(concentration), data = data
997                   , fct = LL.4())
998     }
999     return(fit)
1000   }
1001 })
1002 ECx201 <- eventReactive(input$buttonRunStep201, {
1003   fit <- fitmodel201()

```

```

1002  if(intest_type201() == 'Essai201') {
1003    XX <- as.numeric(input$ecx_ESSAI201)
1004    ec_values <- ED(fit, c(XX, 10, 50), interval = "delta",
1005                      display = FALSE)
1006
1007    drc_df <- data.frame(
1008      EC = c(XX, 10, 50),
1009      Estimate = ec_values[,1],
1010      'Standard Error' = ec_values[,2],
1011      'Lower 95% CI' = ec_values[,3],
1012      'Upper 95% CI' = ec_values[,4]
1013    )
1014
1015    drc_df$Pente <- coefficients(fit)[[1]]
1016
1017    return(drc_df)
1018  }
1019 }
1020
1021 output$drc_result201 <- renderTable({
1022   ECx201()
1023 }, rownames = TRUE)
1024
1025 output$drc_plot201 <- renderPlot({
1026   fit <- fitmodel201()
1027   data <- filedatal201()
1028   ec_values <- ECx201()
1029
1030   TestResult201 <- eventReactive(input$buttonRunStep201, {
1031     data <- filedatal201()
1032     if(intest_type201() == 'Essai201') {
1033       data$concentration <- as.factor(data$concentration)
1034       fit <- lm(response ~ concentration, data = data)
1035       Res <- summary(glht(fit, linfct = mcp(concentration = "Dunnett"), alternative = "less"))
1036       return(Res)
1037     }
1038   })
1039
1040   output$test_result201 <- renderPrint({TestResult201()})
1041
1042   print(ec_values)
1043
1044   if(intest_type201() == 'Essai201') {
1045     plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit201, ")"),
1046           ylab = "Réponse")
1047
1048     ec10 <- ec_values[ec_values$EC == 10,]
1049     ec50 <- ec_values[ec_values$EC == 50,]
1050
1051     print(ec10)

```

```

1050     print(ec50)
1051
1052     if (nrow(ec10) > 0) {
1053         segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)#
1054         y1 = ec10$"Upper 95% CI", col = "blue")
1055         segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)#
1056         y1 = ec10$"Lower 95% CI", col = "blue")
1057         abline(v = ec10$Estimate, col = "red", lty = 2)
1058     }
1059     if (nrow(ec50) > 0) {
1060         segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)#
1061         y1 = ec50$"Upper 95% CI", col = "blue")
1062         segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)#
1063         y1 = ec50$"Lower 95% CI", col = "blue")
1064         abline(v = ec50$Estimate, col = "green", lty = 2)
1065     }
1066
1067     test_results <- TestResult201()
1068     if (!is.null(test_results)) {
1069         noec_values <- test_results$test$p.value > 0.05
1070         loec_values <- test_results$test$p.value <= 0.05
1071         print(noec_values) # Vérifiez les valeurs NOEC
1072         print(loec_values) # Vérifiez les valeurs LOEC
1073         if (any(noec_values, na.rm = TRUE)) {
1074             NOEC <- min(test_results$test$p.value[noec_values], na.rm
1075                         = TRUE)
1076             abline(v = NOEC, col = "purple", lty = 2)
1077         }
1078         if (any(loec_values, na.rm = TRUE)) {
1079             LOEC <- max(test_results$test$p.value[loec_values], na.rm
1080                         = TRUE)
1081             abline(v = LOEC, col = "orange", lty = 2)
1082         }
1083     }
1084 }
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095

```

```

1096  output$rawdata202 <- DT::renderDataTable({
1097    filedata202()
1098  })
1099
1100 fitmodel202 <- eventReactive(input$buttonRunStep202, {
1101   data <- filedata202()
1102   if(intest_type202() == 'Essai202') {
1103     fit <- switch(input$model_ESSAI202,
1104                   '112' = {
1105                     fit1 <- drm(immobilized / total ~
1106                       concentration, data = filedata202() %>%
1107                         dplyr::filter(time == "24"), fct = LL.2(),
1108                         type = "binomial")
1109                     fit2 <- drm(immobilized / total ~
1110                       concentration, data = filedata202() %>%
1111                         dplyr::filter(time == "48"), fct = LL.2(),
1112                         type = "binomial")
1113                     list(fit1 = fit1, fit2 = fit2)
1114                   },
1115                   '113' = {
1116                     fit1 <- drm(immobilized / total ~
1117                       concentration, data = filedata202() %>%
1118                         dplyr::filter(time == "24"), fct = LL.3(),
1119                         type = "binomial")
1120                     fit2 <- drm(immobilized / total ~
1121                       concentration, data = filedata202() %>%
1122                         dplyr::filter(time == "48"), fct = LL.3(),
1123                         type = "binomial")
1124                     list(fit1 = fit1, fit2 = fit2)
1125                   },
1126                   '114' = {
1127                     fit1 <- drm(immobilized / total ~
1128                       concentration, data = filedata202() %>%
1129                         dplyr::filter(time == "24"), fct = LL.4(),
1130                         type = "binomial")
1131                     fit2 <- drm(immobilized / total ~
1132                       concentration, data = filedata202() %>%
1133                         dplyr::filter(time == "48"), fct = LL.4(),
1134                         type = "binomial")
1135                     list(fit1 = fit1, fit2 = fit2)
1136                   }
1137     )
1138     return(fit)
1139   }
1140 }
1141 })
1142
1143 ECx202 <- eventReactive(input$buttonRunStep202, {
1144   fit <- fitmodel202()
1145   if (intest_type202() == 'Essai202') {
1146     XX <- as.numeric(input$ecx_ESSAI202)
1147
1148     fit1 <- fit$fit1

```

```

1130     fit2 <- fit$fit2
1131
1132     drc_df1 <- data.frame(ED(fit1, c(XX), interval = "delta",
1133                               display = FALSE), "Slope" = coefficients(fit1)[[1]])
1134     drc_df2 <- data.frame(ED(fit2, c(XX), interval = "delta",
1135                               display = FALSE), "Slope" = coefficients(fit2)[[1]])
1136
1137     colnames(drc_df1) <- c(paste0('EC', XX), 'Standard_Error', ,
1138                               'Lower_95%CI', 'Upper_95%CI', 'Slope')
1139     colnames(drc_df2) <- c(paste0('EC', XX), 'Standard_Error', ,
1140                               'Lower_95%CI', 'Upper_95%CI', 'Slope')
1141
1142     drc_df <- rbind(drc_df1, drc_df2)
1143     rownames(drc_df) <- c('24\u2075h', '48\u2075h')
1144
1145     return(drc_df)
1146   }
1147 }
1148
1149 output$drc_result202 <- renderTable({
1150   ECx202()
1151 }, rownames = TRUE)
1152
1153 output$drc_plot202 <- renderPlot({
1154   fit <- fitmodel202()
1155   data <- filedata202()
1156   ec_values <- ECx202()
1157
1158   #Pas de test
1159   print(ec_values)
1160
1161   if(intest_type202() == 'Essai202') {
1162     fit <- fitmodel202()
1163     fit1 <- fit$fit1
1164     fit2 <- fit$fit2
1165
1166     par(mar=c(5,5,2,2))
1167     plot(fit1, log="x", broken=TRUE, xlab=paste0("concentration",
1168           , input$conc_unit202, ")), ylab="Mortality",
1169           ylim=c(0,1), lty="dotted", cex=2, cex.axis=2, cex.lab=2)
1170     par(new=TRUE)
1171     plot(fit2, log="x", broken=TRUE, xlab="", ylab="", main="",
1172           ylim=c(0,1), col="#D55E00", cex=2, cex.axis=2, cex.lab=2)
1173     legend("topleft", inset=0.05, legend=c("24\u2075h", "48\u2075h"), col=c(
1174       "black", "#D55E00"), lty=c("dotted", "solid"), cex=2)
1175   }
1176 }
1177
1178 # 203
1179
1180 intest_type203 <- reactive({ 'Essai203' })

```

```

1176
1177 validateFile <- function(filename) {
1178   extFile <- tools::file_ext(filename)
1179   validate(need(extFile %in% c("csv", "txt"), "Seuls les fichiers "
1180             "CSV ou TXT sont acceptés."))
1181 }
1182
1183 filedata203 <- eventReactive(input$buttonRunStep203, {
1184   if (intest_type203() == 'Essai203') {
1185     req(input$datafile_ESSAI203)
1186     validateFile(input$datafile_ESSAI203$name)
1187     ff <- input$datafile_ESSAI203
1188     read.csv(file = ff$datapath, header = TRUE) %>%
1189       mutate(time = as.character(time)) # Ensure 'time' is
1190             treated as a character
1191   }
1192 })
1193
1194 output$rawdata203 <- DT::renderDataTable({
1195   filedata203()
1196 })
1197
1198 fitmodel203 <- eventReactive(input$buttonRunStep203, {
1199   data <- filedata203()
1200   if (intest_type203() == 'Essai203') {
1201     fit_function <- switch(input$model_ESSAI203,
1202                             112 = LL.2(),
1203                             113 = LL.3u(),
1204                             114 = LL.4())
1205     fit_list <- lapply(c("24", "48", "72", "96"), function(t) {
1206       drm(dead/total ~ as.numeric(concentration), data = data %>%
1207             filter(time == t), fct = fit_function, type = "binomial")
1208     })
1209     names(fit_list) <- c("fit1", "fit2", "fit3", "fit4")
1210     return(fit_list)
1211   }
1212 })
1213
1214 ECx203 <- eventReactive(input$buttonRunStep203, {
1215   fit_list <- fitmodel203()
1216   XX <- as.numeric(input$ecx_ESSAI203)
1217
1218   ec_df_list <- lapply(fit_list, function(fit) {
1219     data.frame(ED(fit, c(XX), interval = "delta", display = FALSE)
1220               , "Slope" = coefficients(fit)[[1]])
1221   })
1222
1223   colnames_list <- c(paste0('EC', XX), 'Standard_Error', 'Lower_'
1224                         '95%CI', 'Upper_95%CI', 'Slope')
1225   ec_df_list <- lapply(ec_df_list, setNames, colnames_list)
1226
1227   drc_df <- do.call(rbind, ec_df_list)

```

```

1223     rownames(drc_df) <- c('24h', '48h', '72h', '96h')
1224
1225     return(drc_df)
1226   }
1227
1228   output$drc_result203 <- renderTable({
1229     ECx203()
1230   }, rownames = TRUE)
1231
1232   TestResult203 <- eventReactive(input$buttonRunStep203, {
1233     data <- filedata203() %>%
1234       group_by(concentration, time) %>%
1235       summarize(total = sum(total), dead = sum(dead), .groups = 'drop')
1236
1237     total_24 <- data %>% filter(time == "24" & concentration == "0") %>% pull(total)
1238     total_48 <- data %>% filter(time == "48" & concentration == "0") %>% pull(total)
1239     total_72 <- data %>% filter(time == "72" & concentration == "0") %>% pull(total)
1240     total_96 <- data %>% filter(time == "96" & concentration == "0") %>% pull(total)
1241     dead_24 <- data %>% filter(time == "24" & concentration == "0") %>% pull(dead)
1242     dead_48 <- data %>% filter(time == "48" & concentration == "0") %>% pull(dead)
1243     dead_72 <- data %>% filter(time == "72" & concentration == "0") %>% pull(dead)
1244     dead_96 <- data %>% filter(time == "96" & concentration == "0") %>% pull(dead)
1245
1246     data_24 <- data %>% mutate(total_ctrl = total_24, dead_ctrl = dead_24) %>%
1247       filter(concentration != "0" & time == "24")
1248     data_48 <- data %>% mutate(total_ctrl = total_48, dead_ctrl = dead_48) %>%
1249       filter(concentration != "0" & time == "48")
1250     data_72 <- data %>% mutate(total_ctrl = total_72, dead_ctrl = dead_72) %>%
1251       filter(concentration != "0" & time == "72")
1252     data_96 <- data %>% mutate(total_ctrl = total_96, dead_ctrl = dead_96) %>%
1253       filter(concentration != "0" & time == "96")
1254
1255     fisher <- function(a, b, c, d) {
1256       dt <- matrix(c(a, b, c, d), ncol = 2)
1257       c(pvalue = fisher.test(dt)$p.value)
1258     }
1259
1260     Res1 <- data_24 %>%
1261       rowwise() %>%

```

```

1262     mutate(pvalue = fisher(dead, total - dead, dead_ctrl, total_
1263     ctrl - dead_ctrl)) %>% ungroup() %>%
1264     mutate(p_adjusted = p.adjust(pvalue, "holm")) %>%
1265     mutate(Asterisk = ifelse(p_adjusted < 0.05, ifelse(p_adjusted
1266     > 0.01, "*", "**"), ""))
1267 Res2 <- data_48 %>%
1268     rowwise() %>%
1269     mutate(pvalue = fisher(dead, total - dead, dead_ctrl, total_
1270     ctrl - dead_ctrl)) %>% ungroup() %>%
1271     mutate(p_adjusted = p.adjust(pvalue, "holm")) %>%
1272     mutate(Asterisk = ifelse(p_adjusted < 0.05, ifelse(p_adjusted
1273     > 0.01, "*", "**"), ""))
1274 Res3 <- data_72 %>%
1275     rowwise() %>%
1276     mutate(pvalue = fisher(dead, total - dead, dead_ctrl, total_
1277     ctrl - dead_ctrl)) %>% ungroup() %>%
1278     mutate(p_adjusted = p.adjust(pvalue, "holm")) %>%
1279     mutate(Asterisk = ifelse(p_adjusted < 0.05, ifelse(p_adjusted
1280     > 0.01, "*", "**"), ""))
1281 list("24_h" = knitr::kable(Res1), "48_h" = knitr::kable(Res2), "72_h" = knitr::kable(Res3), "96_h" = knitr::kable(Res4))
1282 }
1283
1284 output$test_result203 <- renderPrint({
1285   TestResult203()
1286 })
1287
1288 output$drc_plot203 <- renderPlot({
1289   fit_list <- fitmodel203()
1290
1291   par(mar = c(5, 5, 2, 2))
1292   colors <- c("black", "#0072B2", "tomato", "black")
1293   linetypes <- c("dotted", "solid", "dotted", "solid")
1294
1295   plot(fit_list[[1]], log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit203, ")), ylab = "Mortality",
1296   ylim = c(0, 1), lty = linetypes[1], cex = 2, cex.axis = 2,
1297   cex.lab = 2, col = colors[1])
1298
1299 for (i in 2:4) {
1300   par(new = TRUE)
1301   plot(fit_list[[i]], log = "x", broken = TRUE, xlab = "", ylab = "", main = "",
```

```

1301         ylim = c(0, 1), lty = linetypes[i], col = colors[i], cex
1302             = 2, cex.axis = 2, cex.lab = 2, ann = FALSE)
1303     }
1304
1304     legend("topleft", inset = 0.05, legend = c("24\u00b9h", "48\u00b9h", "72\u00b9h
1305             ", "96\u00b9h"), col = colors, lty = linetypes, cex = 2)
1306 }
1307 #205
1308
1309 intest_type205 <- reactive({ 'Essai205' })
1310
1311 validateFile <- function(filename) {
1312     extFile <- tools::file_ext(filename)
1313     validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
1314             sont acceptés."))
1315 }
1316
1316 filedatal205 <- eventReactive(input$buttonRunStep205, {
1317     if(intest_type205() == 'Essai205') {
1318         req(input$datafile_ESSAI205)
1319         validateFile(input$datafile_ESSAI205)
1320         ff <- input$datafile_ESSAI205
1321         read.csv(file = ff$datapath, header = TRUE)
1322     }
1323 })
1324
1325 output$rawdata205 <- DT::renderDataTable({
1326     filedatal205()
1327 })
1328
1329 fitmodel205 <- eventReactive(input$buttonRunStep205, {
1330     data <- filedatal205()
1331     if(intest_type205() == 'Essai205') {
1332         if(input$model_ESSAI205 == '112') {
1333             fit <- drm(response ~ as.numeric(dose), data = data, fct =
1334                 LL.2())
1335         } else if(input$model_ESSAI205 == '113') {
1336             fit <- drm(response ~ as.numeric(dose), data = data, fct =
1337                 LL.3())
1338         } else if(input$model_ESSAI205 == '114') {
1339             fit <- drm(response ~ as.numeric(dose), data = data, fct =
1340                 LL.4())
1341         }
1342         return(fit)
1343     }
1344 })
1343 ECx205 <- eventReactive(input$buttonRunStep205, {
1344     fit <- fitmodel205()
1345     if(intest_type205() == 'Essai205') {
1346         XX <- as.numeric(input$ecx_ESSAI205)

```



```

1396     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1397     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1398     abline(v = ec50$Estimate, col = "green", lty = 2)
1399   }
1400
1401   test_results <- TestResult205()
1402   if (!is.null(test_results)) {
1403     noec_values <- test_results$test$p.value > 0.05
1404     loec_values <- test_results$test$p.value <= 0.05
1405     if (any(noec_values, na.rm = TRUE)) {
1406       NOEC <- min(test_results$test$p.value[noec_values], na.rm
1407                     = TRUE)
1408       abline(v = NOEC, col = "purple", lty = 2)
1409     }
1410     if (any(loec_values, na.rm = TRUE)) {
1411       LOEC <- max(test_results$test$p.value[loec_values], na.rm
1412                     = TRUE)
1413       abline(v = LOEC, col = "orange", lty = 2)
1414     }
1415   }
1416 }
1417 #206
1418
1419 intest_type206 <- reactive({ 'Essai206' })
1420
1421 validateFile <- function(filename) {
1422   extFile <- tools::file_ext(filename)
1423   validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
1424   sont acceptés."))
1425 }
1426
1427 filedatal206 <- eventReactive(input$buttonRunStep206, {
1428   if(intest_type206() == 'Essai206') {
1429     req(input$datafile_ESSAI206)
1430     validateFile(input$datafile_ESSAI206)
1431     ff <- input$datafile_ESSAI206
1432     read.csv(file = ff$datapath, header = TRUE)
1433   }
1434 })
1435
1436 output$rawdata206 <- DT::renderDataTable({
1437   filedatal206()
1438 })
1439
1440 fitmodel206 <- eventReactive(input$buttonRunStep206, {
1441   data <- filedatal206()
1442   if(intest_type206() == 'Essai206') {
1443     if(input$model_ESSAI206 == '112') {
1444       fit <- drm(response ~ as.numeric(dose), data = data, fct =
1445                 LL.2())

```

```

1444     } else if(input$model_ESSAI206 == '113') {
1445     fit <- drm(response ~ as.numeric(dose), data = data, fct =
1446                 LL.3())
1447     } else if(input$model_ESSAI206 == '114') {
1448     fit <- drm(response ~ as.numeric(dose), data = data, fct =
1449                 LL.4())
1450   }
1451   return(fit)
1452 }
1453 ECx206 <- eventReactive(input$buttonRunStep206, {
1454   fit <- fitmodel206()
1455   if(intest_type206() == 'Essai206') {
1456     XX <- as.numeric(input$ecx_ESSAI206)
1457     ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
1458                       FALSE)
1459
1460     drc_df <- data.frame(
1461       EC = c(XX, 50),
1462       Estimate = ec_values[,1],
1463       'Standard Error' = ec_values[,2],
1464       'Lower 95% CI' = ec_values[,3],
1465       'Upper 95% CI' = ec_values[,4]
1466     )
1467
1468     drc_df$Pente <- coefficients(fit)[[1]]
1469
1470     return(drc_df)
1471   }
1472 }
1473 output$drc_result206 <- renderTable({
1474   ECx206()
1475 }, rownames = TRUE)
1476
1477 output$drc_plot <- renderPlot({
1478   fit <- fitmodel206()
1479   data <- filedatal()
1480   ec_values <- ECx206()
1481
1482   TestResult206 <- eventReactive(input$buttonRunStep206, {
1483     data <- filedatal()
1484     if(intest_type206() == 'Essai206') {
1485       data$dose <- as.factor(data$dose)
1486       fit <- lm(response ~ dose, data = data)
1487       Res <- summary(glht(fit, linfct = mcp(dose = "Dunnett"),
1488                           alternative = "less"))
1489       return(Res)
1490     }
1491   })

```

```

1492     output$test_result <- renderPrint({TestResult206()})
1493
1494     if(intest_type206() == 'Essai206') {
1495       plot(fit, log = "x", broken = TRUE, xlab = paste0("Dose",",
1496           input$conc_unit206, ")"), ylab = "Réponse")
1497
1498     ec10 <- ec_values[ec_values$EC == 10,]
1499     ec50 <- ec_values[ec_values$EC == 50,]
1500
1501     if (nrow(ec10) > 0) {
1502       segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
1503       segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
1504       abline(v = ec10$Estimate, col = "red", lty = 2)
1505     }
1506     if (nrow(ec50) > 0) {
1507       segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1508       segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1509       abline(v = ec50$Estimate, col = "green", lty = 2)
1510     }
1511
1512     test_results <- TestResult206()
1513     if (!is.null(test_results)) {
1514       noec_values <- test_results$test$p.value > 0.05
1515       loec_values <- test_results$test$p.value <= 0.05
1516       if (any(noec_values, na.rm = TRUE)) {
1517         NOEC <- min(test_results$test$p.value[noec_values], na.rm
1518             = TRUE)
1519         abline(v = NOEC, col = "purple", lty = 2)
1520       }
1521       if (any(loec_values, na.rm = TRUE)) {
1522         LOEC <- max(test_results$test$p.value[loec_values], na.rm
1523             = TRUE)
1524         abline(v = LOEC, col = "orange", lty = 2)
1525       }
1526     }
1527   }
1528
1529   #208
1530
1531   intest_type208 <- reactive({ 'Essai208' })
1532
1533   validateFile <- function(filename) {
1534     extFile <- tools::file_ext(filename)
1535     validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
1536         sont acceptés."))
1537   }
1538
1539   filedatal208 <- eventReactive(input$buttonRunStep208, {
1540     if(intest_type208() == 'Essai208') {
1541       req(input$datafile_ESSAI208)
1542       validateFile(input$datafile_ESSAI208)

```

```

1540     ff <- input$datafile_ESSAI208
1541     read.csv(file = ff$datapath, header = TRUE)
1542   }
1543 }
1544
1545 output$rawdata208 <- DT::renderDataTable({
1546   filedata208()
1547 })
1548
1549 fitmodel208 <- eventReactive(input$buttonRunStep208, {
1550   data <- filedata208()
1551   if(intest_type208() == 'Essai208') {
1552     if(input$model_ESSAI208 == '112') {
1553       fit <- drm(response ~ as.numeric(concentration), data = data
1554         , fct = LL.2())
1555     } else if(input$model_ESSAI208 == '113') {
1556       fit <- drm(response ~ as.numeric(concentration), data = data
1557         , fct = LL.3())
1558     } else if(input$model_ESSAI208 == '114') {
1559       fit <- drm(response ~ as.numeric(concentration), data = data
1560         , fct = LL.4())
1561     }
1562     return(fit)
1563   }
1564 })
1565
1566 ECx208 <- eventReactive(input$buttonRunStep208, {
1567   fit <- fitmodel208()
1568   if(intest_type208() == 'Essai208') {
1569     XX <- as.numeric(input$ecx_ESSAI208)
1570     ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
1571       FALSE)
1572
1573     drc_df <- data.frame(
1574       EC = c(XX, 50),
1575       Estimate = ec_values[,1],
1576       'Standard Error' = ec_values[,2],
1577       'Lower 95% CI' = ec_values[,3],
1578       'Upper 95% CI' = ec_values[,4]
1579     )
1580
1581     drc_df$Pente <- coefficients(fit)[[1]]
1582
1583     return(drc_df)
1584   }
1585 })
1586
1587 output$drc_result208 <- renderTable({
1588   ECx208()
1589   , rownames = TRUE)
1590
1591 output$drc_plot <- renderPlot({

```

```

1588 fit <- fitmodel208()
1589 data <- filedatal208()
1590 ec_values <- ECx208()
1591
1592 TestResult208 <- eventReactive(input$buttonRunStep208, {
1593   data <- filedatal208()
1594   if(intest_type208() == 'Essai208') {
1595     data$concentration <- as.factor(data$concentration)
1596     fit <- lm(response ~ concentration, data = data)
1597     Res <- anova(fit)
1598     return(Res)
1599   }
1600 })
1601
1602 output$test_result <- renderPrint({TestResult208()})
1603
1604 if(intest_type208() == 'Essai208') {
1605   plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit208, ")), ylab = "Réponse")
1606
1607   ec10 <- ec_values[ec_values$EC == 10,]
1608   ec50 <- ec_values[ec_values$EC == 50,]
1609
1610   if (nrow(ec10) > 0) {
1611     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
1612     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
1613     abline(v = ec10$Estimate, col = "red", lty = 2)
1614   }
1615   if (nrow(ec50) > 0) {
1616     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1617     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1618     abline(v = ec50$Estimate, col = "green", lty = 2)
1619   }
1620
1621   test_results <- TestResult208()
1622   if (!is.null(test_results)) {
1623     noec_values <- test_results$test$p.value > 0.05
1624     loec_values <- test_results$test$p.value <= 0.05
1625     if (any(noec_values, na.rm = TRUE)) {
1626       NOEC <- min(test_results$test$p.value[noec_values], na.rm = TRUE)
1627       abline(v = NOEC, col = "purple", lty = 2)
1628     }
1629     if (any(loec_values, na.rm = TRUE)) {
1630       LOEC <- max(test_results$test$p.value[loec_values], na.rm = TRUE)
1631       abline(v = LOEC, col = "orange", lty = 2)
1632     }
1633   }
1634 }
1635

```

```

1636
1637 # 210
1638
1639 intest_type210 <- reactive({ 'Essai210' })
1640
1641 validateFile <- function(filename) {
1642   extFile <- tools::file_ext(filename)
1643   validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
1644   sont acceptés."))
1645 }
1646
1647 filedatal210 <- eventReactive(input$buttonRunStep210, {
1648   if(intest_type210() == 'Essai210') {
1649     req(input$datafile_ESSAI210)
1650     validateFile(input$datafile_ESSAI210)
1651     ff <- input$datafile_ESSAI210
1652     read.csv(file = ff$datapath, header = TRUE)
1653   }
1654 })
1655
1656 output$rawdata210 <- DT::renderDataTable({
1657   filedatal210()
1658 })
1659
1660 fitmodel210 <- eventReactive(input$buttonRunStep210, {
1661   data <- filedatal210()
1662   if(intest_type210() == 'Essai210') {
1663     if(input$model_ESSAI210 == '112') {
1664       fit <- drm(response ~ as.numeric(concentration), data = data
1665       , fct = LL.2())
1666     } else if(input$model_ESSAI210 == '113') {
1667       fit <- drm(response ~ as.numeric(concentration), data = data
1668       , fct = LL.3())
1669     } else if(input$model_ESSAI210 == '114') {
1670       fit <- drm(response ~ as.numeric(concentration), data = data
1671       , fct = LL.4())
1672     }
1673     return(fit)
1674   }
1675 })
1676
1677 ECx210 <- eventReactive(input$buttonRunStep210, {
1678   fit <- fitmodel210()
1679   if(intest_type210() == 'Essai210') {
1680     XX <- as.numeric(input$ecx_ESSAI210)
1681     ec_values <- ED(fit, c(XX, 10, 50), interval = "delta",
1682     display = FALSE)
1683
1684     drc_df <- data.frame(
1685       EC = c(XX, 10, 50),
1686       Estimate = ec_values[,1],
1687       'Standard Error' = ec_values[,2],
1688       'Lower CI' = ec_values[,3],
1689       'Upper CI' = ec_values[,4]
1690     )
1691   }
1692 })
1693
1694 output$ECx210 <- DT::renderDataTable({
1695   ECx210()
1696 })
1697
1698 output$rawdata210 <- DT::renderDataTable({
1699   rawdata210()
1700 })
1701
1702 output$fitmodel210 <- DT::renderDataTable({
1703   fitmodel210()
1704 })
1705
1706 output$ECx210 <- DT::renderDataTable({
1707   ECx210()
1708 })
1709
1710 output$rawdata210 <- DT::renderDataTable({
1711   rawdata210()
1712 })
1713
1714 output$fitmodel210 <- DT::renderDataTable({
1715   fitmodel210()
1716 })
1717
1718 output$ECx210 <- DT::renderDataTable({
1719   ECx210()
1720 })
1721
1722 output$rawdata210 <- DT::renderDataTable({
1723   rawdata210()
1724 })
1725
1726 output$fitmodel210 <- DT::renderDataTable({
1727   fitmodel210()
1728 })
1729
1730 output$ECx210 <- DT::renderDataTable({
1731   ECx210()
1732 })
1733
1734 output$rawdata210 <- DT::renderDataTable({
1735   rawdata210()
1736 })
1737
1738 output$fitmodel210 <- DT::renderDataTable({
1739   fitmodel210()
1740 })
1741
1742 output$ECx210 <- DT::renderDataTable({
1743   ECx210()
1744 })
1745
1746 output$rawdata210 <- DT::renderDataTable({
1747   rawdata210()
1748 })
1749
1750 output$fitmodel210 <- DT::renderDataTable({
1751   fitmodel210()
1752 })
1753
1754 output$ECx210 <- DT::renderDataTable({
1755   ECx210()
1756 })
1757
1758 output$rawdata210 <- DT::renderDataTable({
1759   rawdata210()
1760 })
1761
1762 output$fitmodel210 <- DT::renderDataTable({
1763   fitmodel210()
1764 })
1765
1766 output$ECx210 <- DT::renderDataTable({
1767   ECx210()
1768 })
1769
1770 output$rawdata210 <- DT::renderDataTable({
1771   rawdata210()
1772 })
1773
1774 output$fitmodel210 <- DT::renderDataTable({
1775   fitmodel210()
1776 })
1777
1778 output$ECx210 <- DT::renderDataTable({
1779   ECx210()
1780 })
1781
1782 output$rawdata210 <- DT::renderDataTable({
1783   rawdata210()
1784 })
1785
1786 output$fitmodel210 <- DT::renderDataTable({
1787   fitmodel210()
1788 })
1789
1790 output$ECx210 <- DT::renderDataTable({
1791   ECx210()
1792 })
1793
1794 output$rawdata210 <- DT::renderDataTable({
1795   rawdata210()
1796 })
1797
1798 output$fitmodel210 <- DT::renderDataTable({
1799   fitmodel210()
1800 })
1801
1802 output$ECx210 <- DT::renderDataTable({
1803   ECx210()
1804 })
1805
1806 output$rawdata210 <- DT::renderDataTable({
1807   rawdata210()
1808 })
1809
1810 output$fitmodel210 <- DT::renderDataTable({
1811   fitmodel210()
1812 })
1813
1814 output$ECx210 <- DT::renderDataTable({
1815   ECx210()
1816 })
1817
1818 output$rawdata210 <- DT::renderDataTable({
1819   rawdata210()
1820 })
1821
1822 output$fitmodel210 <- DT::renderDataTable({
1823   fitmodel210()
1824 })
1825
1826 output$ECx210 <- DT::renderDataTable({
1827   ECx210()
1828 })
1829
1830 output$rawdata210 <- DT::renderDataTable({
1831   rawdata210()
1832 })
1833
1834 output$fitmodel210 <- DT::renderDataTable({
1835   fitmodel210()
1836 })
1837
1838 output$ECx210 <- DT::renderDataTable({
1839   ECx210()
1840 })
1841
1842 output$rawdata210 <- DT::renderDataTable({
1843   rawdata210()
1844 })
1845
1846 output$fitmodel210 <- DT::renderDataTable({
1847   fitmodel210()
1848 })
1849
1850 output$ECx210 <- DT::renderDataTable({
1851   ECx210()
1852 })
1853
1854 output$rawdata210 <- DT::renderDataTable({
1855   rawdata210()
1856 })
1857
1858 output$fitmodel210 <- DT::renderDataTable({
1859   fitmodel210()
1860 })
1861
1862 output$ECx210 <- DT::renderDataTable({
1863   ECx210()
1864 })
1865
1866 output$rawdata210 <- DT::renderDataTable({
1867   rawdata210()
1868 })
1869
1870 output$fitmodel210 <- DT::renderDataTable({
1871   fitmodel210()
1872 })
1873
1874 output$ECx210 <- DT::renderDataTable({
1875   ECx210()
1876 })
1877
1878 output$rawdata210 <- DT::renderDataTable({
1879   rawdata210()
1880 })
1881
1882 output$fitmodel210 <- DT::renderDataTable({
1883   fitmodel210()
1884 })
1885
1886 output$ECx210 <- DT::renderDataTable({
1887   ECx210()
1888 })
1889
1890 output$rawdata210 <- DT::renderDataTable({
1891   rawdata210()
1892 })
1893
1894 output$fitmodel210 <- DT::renderDataTable({
1895   fitmodel210()
1896 })
1897
1898 output$ECx210 <- DT::renderDataTable({
1899   ECx210()
1900 })
1901
1902 output$rawdata210 <- DT::renderDataTable({
1903   rawdata210()
1904 })
1905
1906 output$fitmodel210 <- DT::renderDataTable({
1907   fitmodel210()
1908 })
1909
1910 output$ECx210 <- DT::renderDataTable({
1911   ECx210()
1912 })
1913
1914 output$rawdata210 <- DT::renderDataTable({
1915   rawdata210()
1916 })
1917
1918 output$fitmodel210 <- DT::renderDataTable({
1919   fitmodel210()
1920 })
1921
1922 output$ECx210 <- DT::renderDataTable({
1923   ECx210()
1924 })
1925
1926 output$rawdata210 <- DT::renderDataTable({
1927   rawdata210()
1928 })
1929
1930 output$fitmodel210 <- DT::renderDataTable({
1931   fitmodel210()
1932 })
1933
1934 output$ECx210 <- DT::renderDataTable({
1935   ECx210()
1936 })
1937
1938 output$rawdata210 <- DT::renderDataTable({
1939   rawdata210()
1940 })
1941
1942 output$fitmodel210 <- DT::renderDataTable({
1943   fitmodel210()
1944 })
1945
1946 output$ECx210 <- DT::renderDataTable({
1947   ECx210()
1948 })
1949
1950 output$rawdata210 <- DT::renderDataTable({
1951   rawdata210()
1952 })
1953
1954 output$fitmodel210 <- DT::renderDataTable({
1955   fitmodel210()
1956 })
1957
1958 output$ECx210 <- DT::renderDataTable({
1959   ECx210()
1960 })
1961
1962 output$rawdata210 <- DT::renderDataTable({
1963   rawdata210()
1964 })
1965
1966 output$fitmodel210 <- DT::renderDataTable({
1967   fitmodel210()
1968 })
1969
1970 output$ECx210 <- DT::renderDataTable({
1971   ECx210()
1972 })
1973
1974 output$rawdata210 <- DT::renderDataTable({
1975   rawdata210()
1976 })
1977
1978 output$fitmodel210 <- DT::renderDataTable({
1979   fitmodel210()
1980 })
1981
1982 output$ECx210 <- DT::renderDataTable({
1983   ECx210()
1984 })
1985
1986 output$rawdata210 <- DT::renderDataTable({
1987   rawdata210()
1988 })
1989
1990 output$fitmodel210 <- DT::renderDataTable({
1991   fitmodel210()
1992 })
1993
1994 output$ECx210 <- DT::renderDataTable({
1995   ECx210()
1996 })
1997
1998 output$rawdata210 <- DT::renderDataTable({
1999   rawdata210()
2000 })
2001
2002 output$fitmodel210 <- DT::renderDataTable({
2003   fitmodel210()
2004 })
2005
2006 output$ECx210 <- DT::renderDataTable({
2007   ECx210()
2008 })
2009
2010 output$rawdata210 <- DT::renderDataTable({
2011   rawdata210()
2012 })
2013
2014 output$fitmodel210 <- DT::renderDataTable({
2015   fitmodel210()
2016 })
2017
2018 output$ECx210 <- DT::renderDataTable({
2019   ECx210()
2020 })
2021
2022 output$rawdata210 <- DT::renderDataTable({
2023   rawdata210()
2024 })
2025
2026 output$fitmodel210 <- DT::renderDataTable({
2027   fitmodel210()
2028 })
2029
2030 output$ECx210 <- DT::renderDataTable({
2031   ECx210()
2032 })
2033
2034 output$rawdata210 <- DT::renderDataTable({
2035   rawdata210()
2036 })
2037
2038 output$fitmodel210 <- DT::renderDataTable({
2039   fitmodel210()
2040 })
2041
2042 output$ECx210 <- DT::renderDataTable({
2043   ECx210()
2044 })
2045
2046 output$rawdata210 <- DT::renderDataTable({
2047   rawdata210()
2048 })
2049
2050 output$fitmodel210 <- DT::renderDataTable({
2051   fitmodel210()
2052 })
2053
2054 output$ECx210 <- DT::renderDataTable({
2055   ECx210()
2056 })
2057
2058 output$rawdata210 <- DT::renderDataTable({
2059   rawdata210()
2060 })
2061
2062 output$fitmodel210 <- DT::renderDataTable({
2063   fitmodel210()
2064 })
2065
2066 output$ECx210 <- DT::renderDataTable({
2067   ECx210()
2068 })
2069
2070 output$rawdata210 <- DT::renderDataTable({
2071   rawdata210()
2072 })
2073
2074 output$fitmodel210 <- DT::renderDataTable({
2075   fitmodel210()
2076 })
2077
2078 output$ECx210 <- DT::renderDataTable({
2079   ECx210()
2080 })
2081
2082 output$rawdata210 <- DT::renderDataTable({
2083   rawdata210()
2084 })
2085
2086 output$fitmodel210 <- DT::renderDataTable({
2087   fitmodel210()
2088 })
2089
2090 output$ECx210 <- DT::renderDataTable({
2091   ECx210()
2092 })
2093
2094 output$rawdata210 <- DT::renderDataTable({
2095   rawdata210()
2096 })
2097
2098 output$fitmodel210 <- DT::renderDataTable({
2099   fitmodel210()
2100 })
2101
2102 output$ECx210 <- DT::renderDataTable({
2103   ECx210()
2104 })
2105
2106 output$rawdata210 <- DT::renderDataTable({
2107   rawdata210()
2108 })
2109
2110 output$fitmodel210 <- DT::renderDataTable({
2111   fitmodel210()
2112 })
2113
2114 output$ECx210 <- DT::renderDataTable({
2115   ECx210()
2116 })
2117
2118 output$rawdata210 <- DT::renderDataTable({
2119   rawdata210()
2120 })
2121
2122 output$fitmodel210 <- DT::renderDataTable({
2123   fitmodel210()
2124 })
2125
2126 output$ECx210 <- DT::renderDataTable({
2127   ECx210()
2128 })
2129
2130 output$rawdata210 <- DT::renderDataTable({
2131   rawdata210()
2132 })
2133
2134 output$fitmodel210 <- DT::renderDataTable({
2135   fitmodel210()
2136 })
2137
2138 output$ECx210 <- DT::renderDataTable({
2139   ECx210()
2140 })
2141
2142 output$rawdata210 <- DT::renderDataTable({
2143   rawdata210()
2144 })
2145
2146 output$fitmodel210 <- DT::renderDataTable({
2147   fitmodel210()
2148 })
2149
2150 output$ECx210 <- DT::renderDataTable({
2151   ECx210()
2152 })
2153
2154 output$rawdata210 <- DT::renderDataTable({
2155   rawdata210()
2156 })
2157
2158 output$fitmodel210 <- DT::renderDataTable({
2159   fitmodel210()
2160 })
2161
2162 output$ECx210 <- DT::renderDataTable({
2163   ECx210()
2164 })
2165
2166 output$rawdata210 <- DT::renderDataTable({
2167   rawdata210()
2168 })
2169
2170 output$fitmodel210 <- DT::renderDataTable({
2171   fitmodel210()
2172 })
2173
2174 output$ECx210 <- DT::renderDataTable({
2175   ECx210()
2176 })
2177
2178 output$rawdata210 <- DT::renderDataTable({
2179   rawdata210()
2180 })
2181
2182 output$fitmodel210 <- DT::renderDataTable({
2183   fitmodel210()
2184 })
2185
2186 output$ECx210 <- DT::renderDataTable({
2187   ECx210()
2188 })
2189
2190 output$rawdata210 <- DT::renderDataTable({
2191   rawdata210()
2192 })
2193
2194 output$fitmodel210 <- DT::renderDataTable({
2195   fitmodel210()
2196 })
2197
2198 output$ECx210 <- DT::renderDataTable({
2199   ECx210()
2200 })
2201
2202 output$rawdata210 <- DT::renderDataTable({
2203   rawdata210()
2204 })
2205
2206 output$fitmodel210 <- DT::renderDataTable({
2207   fitmodel210()
2208 })
2209
2210 output$ECx210 <- DT::renderDataTable({
2211   ECx210()
2212 })
2213
2214 output$rawdata210 <- DT::renderDataTable({
2215   rawdata210()
2216 })
2217
2218 output$fitmodel210 <- DT::renderDataTable({
2219   fitmodel210()
2220 })
2221
2222 output$ECx210 <- DT::renderDataTable({
2223   ECx210()
2224 })
2225
2226 output$rawdata210 <- DT::renderDataTable({
2227   rawdata210()
2228 })
2229
2230 output$fitmodel210 <- DT::renderDataTable({
2231   fitmodel210()
2232 })
2233
2234 output$ECx210 <- DT::renderDataTable({
2235   ECx210()
2236 })
2237
2238 output$rawdata210 <- DT::renderDataTable({
2239   rawdata210()
2240 })
2241
2242 output$fitmodel210 <- DT::renderDataTable({
2243   fitmodel210()
2244 })
2245
2246 output$ECx210 <- DT::renderDataTable({
2247   ECx210()
2248 })
2249
2250 output$rawdata210 <- DT::renderDataTable({
2251   rawdata210()
2252 })
2253
2254 output$fitmodel210 <- DT::renderDataTable({
2255   fitmodel210()
2256 })
2257
2258 output$ECx210 <- DT::renderDataTable({
2259   ECx210()
2260 })
2261
2262 output$rawdata210 <- DT::renderDataTable({
2263   rawdata210()
2264 })
2265
2266 output$fitmodel210 <- DT::renderDataTable({
2267   fitmodel210()
2268 })
2269
2270 output$ECx210 <- DT::renderDataTable({
2271   ECx210()
2272 })
2273
2274 output$rawdata210 <- DT::renderDataTable({
2275   rawdata210()
2276 })
2277
2278 output$fitmodel210 <- DT::renderDataTable({
2279   fitmodel210()
2280 })
2281
2282 output$ECx210 <- DT::renderDataTable({
2283   ECx210()
2284 })
2285
2286 output$rawdata210 <- DT::renderDataTable({
2287   rawdata210()
2288 })
2289
2290 output$fitmodel210 <- DT::renderDataTable({
2291   fitmodel210()
2292 })
2293
2294 output$ECx210 <- DT::renderDataTable({
2295   ECx210()
2296 })
2297
2298 output$rawdata210 <- DT::renderDataTable({
2299   rawdata210()
2300 })
2301
2302 output$fitmodel210 <- DT::renderDataTable({
2303   fitmodel210()
2304 })
2305
2306 output$ECx210 <- DT::renderDataTable({
2307   ECx210()
2308 })
2309
2310 output$rawdata210 <- DT::renderDataTable({
2311   rawdata210()
2312 })
2313
2314 output$fitmodel210 <- DT::renderDataTable({
2315   fitmodel210()
2316 })
2317
2318 output$ECx210 <- DT::renderDataTable({
2319   ECx210()
2320 })
2321
2322 output$rawdata210 <- DT::renderDataTable({
2323   rawdata210()
2324 })
2325
2326 output$fitmodel210 <- DT::renderDataTable({
2327   fitmodel210()
2328 })
2329
2330 output$ECx210 <- DT::renderDataTable({
2331   ECx210()
2332 })
2333
2334 output$rawdata210 <- DT::renderDataTable({
2335   rawdata210()
2336 })
2337
2338 output$fitmodel210 <- DT::renderDataTable({
2339   fitmodel210()
2340 })
2341
2342 output$ECx210 <- DT::renderDataTable({
2343   ECx210()
2344 })
2345
2346 output$rawdata210 <- DT::renderDataTable({
2347   rawdata210()
2348 })
2349
2350 output$fitmodel210 <- DT::renderDataTable({
2351   fitmodel210()
2352 })
2353
2354 output$ECx210 <- DT::renderDataTable({
2355   ECx210()
2356 })
2357
2358 output$rawdata210 <- DT::renderDataTable({
2359   rawdata210()
2360 })
2361
2362 output$fitmodel210 <- DT::renderDataTable({
2363   fitmodel210()
2364 })
2365
2366 output$ECx210 <- DT::renderDataTable({
2367   ECx210()
2368 })
2369
2370 output$rawdata210 <- DT::renderDataTable({
2371   rawdata210()
2372 })
2373
2374 output$fitmodel210 <- DT::renderDataTable({
2375   fitmodel210()
2376 })
2377
2378 output$ECx210 <- DT::renderDataTable({
2379   ECx210()
2380 })
2381
2382 output$rawdata210 <- DT::renderDataTable({
2383   rawdata210()
2384 })
2385
2386 output$fitmodel210 <- DT::renderDataTable({
2387   fitmodel210()
2388 })
2389
2390 output$ECx210 <- DT::renderDataTable({
2391   ECx210()
2392 })
2393
2394 output$rawdata210 <- DT::renderDataTable({
2395   rawdata210()
2396 })
2397
2398 output$fitmodel210 <- DT::renderDataTable({
2399   fitmodel210()
2400 })
2401
2402 output$ECx210 <- DT::renderDataTable({
2403   ECx210()
2404 })
2405
2406 output$rawdata210 <- DT::renderDataTable({
2407   rawdata210()
2408 })
2409
2410 output$fitmodel210 <- DT::renderDataTable({
2411   fitmodel210()
2412 })
2413
2414 output$ECx210 <- DT::renderDataTable({
2415   ECx210()
2416 })
2417
2418 output$rawdata210 <- DT::renderDataTable({
2419   rawdata210()
2420 })
2421
2422 output$fitmodel210 <- DT::renderDataTable({
2423   fitmodel210()
2424 })
2425
2426 output$ECx210 <- DT::renderDataTable({
2427   ECx210()
2428 })
2429
2430 output$rawdata210 <- DT::renderDataTable({
2431   rawdata210()
2432 })
2433
2434 output$fitmodel210 <- DT::renderDataTable({
2435   fitmodel210()
2436 })
2437
2438 output$ECx210 <- DT::renderDataTable({
2439   ECx210()
2440 })
2441
2442 output$rawdata210 <- DT::renderDataTable({
2443   rawdata210()
2444 })
2445
2446 output$fitmodel210 <- DT::renderDataTable({
2447   fitmodel210()
2448 })
2449
2450 output$ECx210 <- DT::renderDataTable({
2451   ECx210()
2452 })
2453
2454 output$rawdata210 <- DT::renderDataTable({
2455   rawdata210()
2456 })
2457
2458 output$fitmodel210 <- DT::renderDataTable({
2459   fitmodel210()
2460 })
2461
2462 output$ECx210 <- DT::renderDataTable({
2463   ECx210()
2464 })
2465
2466 output$rawdata210 <- DT::renderDataTable({
2467   rawdata210()
2468 })
2469
2470 output$fitmodel210 <- DT::renderDataTable({
2471   fitmodel210()
2472 })
2473
2474 output$ECx210 <- DT::renderDataTable({
2475   ECx210()
2476 })
2477
2478 output$rawdata210 <- DT::renderDataTable({
2479   rawdata210()
2480 })
2481
2482 output$fitmodel210 <- DT::renderDataTable({
2483   fitmodel210()
2484 })
2485
2486 output$ECx210 <- DT::renderDataTable({
2487   ECx210()
2488 })
2489
2490 output$rawdata210 <- DT::renderDataTable({
2491   rawdata210()
2492 })
2493
2494 output$fitmodel210 <- DT::renderDataTable({
2495   fitmodel210()
2496 })
2497
2498 output$ECx210 <- DT::renderDataTable({
2499   ECx210()
2500 })
2501
2502 output$rawdata210 <- DT::renderDataTable({
2503   rawdata210()
2504 })
2505
2506 output$fitmodel210 <- DT::renderDataTable({
2507   fitmodel210()
2508 })
2509
2510 output$ECx210 <- DT::renderDataTable({
2511   ECx210()
2512 })
2513
2514 output$rawdata210 <- DT::renderDataTable({
2515   rawdata210()
2516 })
2517
2518 output$fitmodel210 <- DT::renderDataTable({
2519   fitmodel210()
2520 })
2521
2522 output$ECx210 <- DT::renderDataTable({
2523   ECx210()
2524 })
2525
2526 output$rawdata210 <- DT::renderDataTable({
2527   rawdata210()
2528 })
2529
2530 output$fitmodel210 <- DT::renderDataTable({
2531   fitmodel210()
2532 })
2533
2534 output$ECx210 <- DT::renderDataTable({
2535   ECx210()
2536 })
2537
2538 output$rawdata210 <- DT::renderDataTable({
2539   rawdata210()
2540 })
2541
2542 output$fitmodel210 <- DT::renderDataTable({
2543   fitmodel210()
2544 })
2545
2546 output$ECx210 <- DT::renderDataTable({
2547   ECx210()
2548 })
2549
2550 output$rawdata210 <- DT::renderDataTable({
2551   rawdata210()
2552 })
2553
2554 output$fitmodel210 <- DT::renderDataTable({
2555   fitmodel210()
2556 })
2557
2558 output$ECx210 <- DT::renderDataTable({
2559   ECx210()
2560 })
2561
2562 output$rawdata210 <- DT::renderDataTable({
2563   rawdata210()
2564 })
2565
2566 output$fitmodel210 <- DT::renderDataTable({
2567   fitmodel210()
2568 })
2569
2570 output$ECx210 <- DT::renderDataTable({
2571   ECx210()
2572 })
2573
2574 output$rawdata210 <- DT::renderDataTable({
2575   rawdata210()
2576 })
2577
2578 output$fitmodel210 <- DT::renderDataTable({
2579   fitmodel210()
2580 })
2581
2582 output$ECx210 <- DT::renderDataTable({
2583   ECx210()
2584 })
2585
2586 output$rawdata210 <- DT::renderDataTable({
2587   rawdata210()
2588 })
2589
2590 output$fitmodel210 <- DT::renderDataTable({
2591   fitmodel210()
2592 })
2593
2594 output$ECx210 <- DT::renderDataTable({
2595   ECx210()
2596 })
2597
2598 output$rawdata210 <- DT::renderDataTable({
2599   rawdata210()
2600 })
2601
2602 output$fitmodel210 <- DT::renderDataTable({
2603   fitmodel210()
2604 })
2605
2606 output$ECx210 <- DT::renderDataTable({
2607   ECx210()
2608 })
2609
2610 output$rawdata210 <- DT::renderDataTable({
2611   rawdata210()
2612 })
2613
2614 output$fitmodel210 <- DT::renderDataTable({
2615   fitmodel210()
2616 })
2617
2618 output$ECx210 <- DT::renderDataTable({
2619   ECx210()
2620 })
2621
2622 output$rawdata210 <- DT::renderDataTable({
2623   rawdata210()
2624 })
2625
2626 output$fitmodel210 <- DT::renderDataTable({
2627   fitmodel210()
2628 })
2629
2630 output$ECx210 <- DT::renderDataTable({
2631   ECx210()
2632 })
2633
2634 output$rawdata210 <- DT::renderDataTable({
2635   rawdata210()
2636 })
2637
2638 output$fitmodel210 <- DT::renderDataTable({
2639   fitmodel210()
2640 })
2641
2642 output$ECx210 <- DT::renderDataTable({
2643   ECx210()
2644 })
2645
2646 output$rawdata210 <- DT::renderDataTable({
2647   rawdata210()
2648 })
2649
2650 output$fitmodel210 <- DT::renderDataTable({
2651   fitmodel210()
2652 })
2653
2654 output$ECx210 <- DT::renderDataTable({
2655   ECx210()
2656 })
2657
2658 output$rawdata210 <- DT::renderDataTable({
2659   rawdata210()
2660 })
2661
2662 output$fitmodel210 <- DT::renderDataTable({
2663   fitmodel210()
2664 })
2665
2666 output$ECx210 <- DT::renderDataTable({
2667   ECx210()
2668 })
2669
2670 output$rawdata210 <- DT::renderDataTable({
2671   rawdata210()
2672 })
2673
2674 output$fitmodel210 <- DT::renderDataTable({
2675   fitmodel210()
2676 })
2677
2678 output$ECx210 <- DT::renderDataTable({
2679   ECx210()
2680 })
2681
2682 output$rawdata210 <- DT::renderDataTable({
2683   rawdata210()
2684 })
2685
2686 output$fitmodel210 <- DT::renderDataTable({
2687   fitmodel210()
2688 })
2689
2690 output$ECx210 <- DT::renderDataTable({
2691   ECx210()
2692 })
2693
2694 output$rawdata210 <- DT::renderDataTable({
2695   rawdata210()
2696 })
2697
2698 output$fitmodel210 <- DT::renderDataTable({
2699   fitmodel210()
2700 })
2701
2702 output$ECx210 <- DT::renderDataTable({
2703   ECx210()
2704 })
2705
2706 output$rawdata210 <- DT::renderDataTable({
2707   rawdata210()
2708 })
2709
2710 output$fitmodel210 <- DT::renderDataTable({
2711   fitmodel210()
2712 })
2713
2714 output$ECx210 <- DT::renderDataTable({
2715   ECx210()
2716 })
2717
2718 output$rawdata210 <- DT::renderDataTable({
2719   rawdata210()
2720 })
2721
2722 output$fitmodel210 <- DT::renderDataTable({
2723   fitmodel210()
2724 })
2725
2726 output$ECx210 <- DT::renderDataTable({
2727   ECx210()
2728 })
2729
2730 output$rawdata210 <- DT::renderDataTable({
2731   rawdata210()
2732 })
2733
2734 output$fitmodel210 <- DT::renderDataTable({
2735   fitmodel210()
2736 })
2737
2738 output$ECx210 <- DT::renderDataTable({
2739   ECx210()
2740 })
2741
2742 output$rawdata210 <- DT::renderDataTable({
2743   rawdata210()
2744 })
2745
2746 output$fitmodel210 <- DT::renderDataTable({
2747   fitmodel210()
2748 })
2749
2750 output$ECx210 <- DT::renderDataTable({
2751   ECx210()
2752 })
2753
2754 output$rawdata210 <- DT::renderDataTable({
2755   rawdata210()
2756 })
2757
2758 output$fitmodel210 <- DT::renderDataTable({
2759   fitmodel210()
2760 })
2761
2762 output$ECx210 <- DT::renderDataTable({
2763   ECx210()
2764 })
2765
2766 output$rawdata210 <- DT::renderDataTable({
2767   rawdata210()
2768 })
2769
2770 output$fitmodel210 <- DT::renderDataTable({
2771   fitmodel210()
2772 })
2773
2774 output$ECx210 <- DT::renderDataTable({
2775   ECx210()
2776 })
2777
2778 output$rawdata210 <- DT::renderDataTable({
2779   rawdata210()
2780 })
2781
2782 output$fitmodel210 <- DT::renderDataTable({
2783   fitmodel210()
2784 })
2785
2786 output$ECx210 <- DT::renderDataTable({
2787   ECx210()
2788 })
2789
2790 output$rawdata210 <- DT::renderDataTable({
2791   rawdata210()
2792 })
2793
2794 output$fitmodel210 <- DT::renderDataTable({
2795   fitmodel210()
2796 })
2797
2798 output$ECx210 <- DT::renderDataTable({
2799   ECx210()
2800 })
2801
2802 output$rawdata210 <- DT::renderDataTable({
2803   rawdata210()
2804 })
2805
2806 output$fitmodel210 <- DT::renderDataTable({
2807   fitmodel210()
2808 })
2809
2810 output$ECx210 <- DT::renderDataTable({
2811   ECx210()
2812 })
2813
2814 output$rawdata210 <- DT::renderDataTable({
2815   rawdata210()
2816 })
2817
2818 output$fitmodel210 <- DT::renderDataTable({
2819   fitmodel210()
2820 })
2821
2822 output$ECx210 <- DT::renderDataTable({
2823   ECx210()
2824 })
2825
2826 output$rawdata210 <- DT::renderDataTable({
2827   rawdata210()
2828 })
2829
2830 output$fitmodel210 <- DT::renderDataTable({
2831   fitmodel210()
2832 })
2833
2834 output$ECx210 <- DT::renderDataTable({
2835   ECx210()
2836 })
2837
2838 output$rawdata210 <- DT::renderDataTable({
2839   rawdata210()
2840 })
2841
2842 output$fitmodel210 <- DT::renderDataTable({
2843   fitmodel210()
2844 })
2845
2846 output$ECx210 <- DT::renderDataTable({
2847   ECx210()
2848
```

```

1683     'Lower 95% CI' = ec_values[,3],
1684     'Upper 95% CI' = ec_values[,4]
1685   )
1686
1687   drc_df$Pente <- coefficients(fit)[[1]]
1688
1689   return(drc_df)
1690 }
1691 )
1692
1693 output$drc_result210 <- renderTable({
1694   ECx210()
1695 }, rownames = TRUE)
1696
1697 output$drc_plot <- renderPlot({
1698   fit <- fitmodel210()
1699   data <- filedatal210()
1700   ec_values <- ECx210()
1701
1702   TestResult210 <- eventReactive(input$buttonRunStep210, {
1703     data <- filedatal210()
1704     if(intest_type210() == 'Essai210') {
1705       data$concentration <- as.factor(data$concentration)
1706       fit <- lm(response ~ concentration, data = data)
1707       Res <- summary(glht(fit, linfct = mcp(concentration = "Dunnett"), alternative = "less"))
1708       return(Res)
1709     }
1710   })
1711
1712   output$test_result <- renderPrint({TestResult210()})
1713
1714   if(intest_type210() == 'Essai210') {
1715     plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit210, ")), ylab = "Réponse")
1716
1717     ec10 <- ec_values[ec_values$EC == 10,]
1718     ec50 <- ec_values[ec_values$EC == 50,]
1719
1720     if (nrow(ec10) > 0) {
1721       segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
1722       segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
1723       abline(v = ec10$Estimate, col = "red", lty = 2)
1724     }
1725     if (nrow(ec50) > 0) {
1726       segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1727       segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1728       abline(v = ec50$Estimate, col = "green", lty = 2)
1729     }
1730
1731     test_results <- TestResult210()

```

```

1732     if (!is.null(test_results)) {
1733       noec_values <- test_results$test$p.value > 0.05
1734       loec_values <- test_results$test$p.value <= 0.05
1735       if (any(noec_values, na.rm = TRUE)) {
1736         NOEC <- min(test_results$test$p.value[noec_values], na.rm
1737                       = TRUE)
1738         abline(v = NOEC, col = "purple", lty = 2)
1739       }
1740       if (any(loec_values, na.rm = TRUE)) {
1741         LOEC <- max(test_results$test$p.value[loec_values], na.rm
1742                       = TRUE)
1743         abline(v = LOEC, col = "orange", lty = 2)
1744       }
1745     }
1746   )
1747 #211
1748
1749 intest_type211 <- reactive({ 'Essai211' })
1750
1751 validateFile <- function(filename) {
1752   extFile <- tools::file_ext(filename)
1753   validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
1754   sont acceptés."))
1755 }
1756
1757 filedata211 <- eventReactive(input$buttonRunStep211, {
1758   if(intest_type211() == 'Essai211') {
1759     req(input$datafile_ESSAI211)
1760     validateFile(input$datafile_ESSAI211)
1761     ff <- input$datafile_ESSAI211
1762     read.csv(file = ff$datapath, header = TRUE)
1763   }
1764 })
1765
1766 output$rawdata211 <- DT::renderDataTable({
1767   filedata211()
1768 })
1769
1770 fitmodel211 <- eventReactive(input$buttonRunStep211, {
1771   data <- filedata211()
1772   if(intest_type211() == 'Essai211') {
1773     if(input$model_ESSAI211 == 'lm') {
1774       fit <- lm(response ~ as.numeric(concentration), data = data)
1775     } else if(input$model_ESSAI211 == 'nls') {
1776       fit <- nls(response ~ fct(concentration), data = data)
1777     } else if(input$model_ESSAI211 == 'lme') {
1778       fit <- lme(response ~ as.numeric(concentration), random = ~
1779                   1 | group, data = data)
1780     }
1781   }
1782   return(fit)

```

```

1780     }
1781   })
1782
1783   ECx211 <- eventReactive(input$buttonRunStep211, {
1784     fit <- fitmodel211()
1785     if(intest_type211() == 'Essai211') {
1786       XX <- as.numeric(input$ecx_ESSAI211)
1787       ec_values <- ED(fit, c(XX, 10, 50), interval = "delta",
1788         display = FALSE)
1789
1790       drc_df <- data.frame(
1791         EC = c(XX, 10, 50),
1792         Estimate = ec_values[,1],
1793         'Standard Error' = ec_values[,2],
1794         'Lower 95% CI' = ec_values[,3],
1795         'Upper 95% CI' = ec_values[,4]
1796       )
1797
1798       drc_df$Pente <- coefficients(fit)[[1]]
1799
1800       return(drc_df)
1801     }
1802   })
1803
1804   output$drc_result211 <- renderTable({
1805     ECx211()
1806   }, rownames = TRUE)
1807
1808   output$drc_plot <- renderPlot({
1809     fit <- fitmodel211()
1810     data <- filedatal211()
1811     ec_values <- ECx211()
1812
1813     TestResult211 <- eventReactive(input$buttonRunStep211, {
1814       data <- filedatal211()
1815       if(intest_type211() == 'Essai211') {
1816         data$concentration <- as.factor(data$concentration)
1817         fit <- lm(response ~ concentration, data = data)
1818         Res <- anova(fit)
1819         return(Res)
1820       }
1821     })
1822
1823     output$test_result <- renderPrint({TestResult211()})
1824
1825     if(intest_type211() == 'Essai211') {
1826       plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration",
1827         input$conc_unit211, ")"), ylab = "Réponse")
1828       ec10 <- ec_values[ec_values$EC == 10,]

```

```

1829     ec50 <- ec_values[ec_values$EC == 50,]
1830
1831     if (nrow(ec10) > 0) {
1832         segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
1833         segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
1834         abline(v = ec10$Estimate, col = "red", lty = 2)
1835     }
1836     if (nrow(ec50) > 0) {
1837         segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1838         segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1839         abline(v = ec50$Estimate, col = "green", lty = 2)
1840     }
1841
1842     test_results <- TestResult211()
1843     if (!is.null(test_results)) {
1844         noec_values <- test_results$test$p.value > 0.05
1845         loec_values <- test_results$test$p.value <= 0.05
1846         if (any(noec_values, na.rm = TRUE)) {
1847             NOEC <- min(test_results$test$p.value[noec_values], na.rm
1848                           = TRUE)
1849             abline(v = NOEC, col = "purple", lty = 2)
1850         }
1851         if (any(loec_values, na.rm = TRUE)) {
1852             LOEC <- max(test_results$test$p.value[loec_values], na.rm
1853                           = TRUE)
1854             abline(v = LOEC, col = "orange", lty = 2)
1855         }
1856     }
1857
1858 #213_214
1859
1860 intest_type213214 <- reactive({ 'Essai213214' })
1861
1862 validateFile <- function(filename) {
1863     extFile <- tools::file_ext(filename)
1864     validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
1865     sont acceptés."))
1866 }
1867
1868 filedatal213214 <- eventReactive(input$buttonRunStep213214, {
1869     if(intest_type213214() == 'Essai213214') {
1870         req(input$datafile_ESSAI213214)
1871         validateFile(input$datafile_ESSAI213214)
1872         ff <- input$datafile_ESSAI213214
1873         read.csv(file = ff$datapath, header = TRUE)
1874     }
1875 }
1876
1877 output$rawdata213214 <- DT::renderDataTable({
1878     filedatal213214()

```

```

1878  })
1879
1880  fitmodel213214 <- eventReactive(input$buttonRunStep213214, {
1881    data <- filedata213214()
1882    if(intest_type213214() == 'Essai213214') {
1883      if(input$model_ESSAI213214 == 'probit') {
1884        fit <- drm(response ~ as.numeric(dose), data = data, fct =
1885          PROBIT())
1886      } else if(input$model_ESSAI213214 == 'logit') {
1887        fit <- drm(response ~ as.numeric(dose), data = data, fct =
1888          LOGIT())
1889      }
1890      return(fit)
1891    }
1892  })
1893
1894  ECx213214 <- eventReactive(input$buttonRunStep213214, {
1895    fit <- fitmodel213214()
1896    if(intest_type213214() == 'Essai213214') {
1897      XX <- as.numeric(input$ecx_ESSAI213214)
1898      ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
1899        FALSE)
1900
1901      drc_df <- data.frame(
1902        EC = c(XX, 50),
1903        Estimate = ec_values[,1],
1904        'Standard Error' = ec_values[,2],
1905        'Lower 95% CI' = ec_values[,3],
1906        'Upper 95% CI' = ec_values[,4]
1907      )
1908
1909      drc_df$Pente <- coefficients(fit)[[1]]
1910
1911      return(drc_df)
1912    }
1913  })
1914
1915  output$drc_result213214 <- renderTable({
1916    ECx213214()
1917  }, rownames = TRUE)
1918
1919  output$drc_plot <- renderPlot({
1920    fit <- fitmodel213214()
1921    data <- filedata213214()
1922    ec_values <- ECx213214()
1923
1924    TestResult213214 <- eventReactive(input$buttonRunStep213214, {
1925      data <- filedata213214()
1926      if(intest_type213214() == 'Essai213214') {
1927        data$dose <- as.factor(data$dose)
1928        fit <- lm(response ~ dose, data = data)
1929      }
1930    })
1931  })

```

```

1927     Res <- summary(glht(fit, linfct = mcp(dose = "Dunnett"),
1928                     alternative = "less"))
1929     return(Res)
1930   }
1931 }
1932 output$test_result <- renderPrint({TestResult214()})
1933
1934 if(intest_type213214() == 'Essai213214') {
1935   plot(fit, log = "x", broken = TRUE, xlab = paste0("Dose",",
1936   input$conc_unit213214, ")"), ylab = "Réponse")
1937
1938   ec10 <- ec_values[ec_values$EC == 10,]
1939   ec50 <- ec_values[ec_values$EC == 50,]
1940
1941   if (nrow(ec10) > 0) {
1942     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
1943     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
1944     abline(v = ec10$Estimate, col = "red", lty = 2)
1945   }
1946   if (nrow(ec50) > 0) {
1947     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1948     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
1949     abline(v = ec50$Estimate, col = "green", lty = 2)
1950   }
1951
1952   test_results <- TestResult214()
1953   if (!is.null(test_results)) {
1954     noec_values <- test_results$test$p.value > 0.05
1955     loec_values <- test_results$test$p.value <= 0.05
1956     if (any(noec_values, na.rm = TRUE)) {
1957       NOEC <- min(test_results$test$p.value[noec_values], na.rm
1958                     = TRUE)
1959       abline(v = NOEC, col = "purple", lty = 2)
1960     }
1961     if (any(loec_values, na.rm = TRUE)) {
1962       LOEC <- max(test_results$test$p.value[loec_values], na.rm
1963                     = TRUE)
1964       abline(v = LOEC, col = "orange", lty = 2)
1965     }
1966   }
1967 }
1968 #216
1969
1970 intest_type216 <- reactive({ 'Essai216' })
1971
1972 validateFile <- function(filename) {
1973   extFile <- tools::file_ext(filename)
1974   validate(need(extFile == c("csv", "txt")), "Seuls les fichiers CSV"

```

```

1975         "sont acceptés."))
1976
1977     filedata216 <- eventReactive(input$buttonRunStep216, {
1978       if(intest_type216() == 'Essai216') {
1979         req(input$datafile_ESSAI216)
1980         validateFile(input$datafile_ESSAI216)
1981         ff <- input$datafile_ESSAI216
1982         read.csv(file = ff$datapath, header = TRUE)
1983     }
1984   })
1985
1986   output$rawdata216 <- DT::renderDataTable({
1987     filedata216()
1988   })
1989
1990   fitmodel216 <- eventReactive(input$buttonRunStep216, {
1991     data <- filedata216()
1992     if(intest_type216() == 'Essai216') {
1993       if(input$model_ESSAI216 == 'lm') {
1994         fit <- lm(response ~ as.numeric(concentration), data = data)
1995       } else if(input$model_ESSAI216 == 'nls') {
1996         fit <- nls(response ~ fct(concentration), data = data)
1997       } else if(input$model_ESSAI216 == 'lme') {
1998         fit <- lme(response ~ as.numeric(concentration), random = ~
1999                     1 | group, data = data)
2000       }
2001       return(fit)
2002     }
2003   })
2004
2005   ECx216 <- eventReactive(input$buttonRunStep216, {
2006     fit <- fitmodel216()
2007     if(intest_type216() == 'Essai216') {
2008       XX <- as.numeric(input$ecx_ESSAI216)
2009       ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
2010                     FALSE)
2011
2012       drc_df <- data.frame(
2013         EC = c(XX, 50),
2014         Estimate = ec_values[,1],
2015         'Standard Error' = ec_values[,2],
2016         'Lower 95% CI' = ec_values[,3],
2017         'Upper 95% CI' = ec_values[,4]
2018     )
2019
2020       drc_df$Pente <- coefficients(fit)[[1]]
2021
2022       return(drc_df)
2023     }
2024   })

```

```

2024  output$drc_result216 <- renderTable({
2025    ECx216()
2026  }, rownames = TRUE)
2027
2028  output$drc_plot <- renderPlot({
2029    fit <- fitmodel216()
2030    data <- filedatal216()
2031    ec_values <- ECx216()
2032
2033    TestResult216 <- eventReactive(input$buttonRunStep216, {
2034      data <- filedatal216()
2035      if(intest_type216() == 'Essai216') {
2036        data$concentration <- as.factor(data$concentration)
2037        fit <- lm(response ~ concentration, data = data)
2038        Res <- anova(fit)
2039        return(Res)
2040      }
2041    })
2042
2043    output$test_result <- renderPrint({TestResult216()})
2044
2045    if(intest_type216() == 'Essai216') {
2046      plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit216, ")), ylab = "Réponse")
2047
2048      ec10 <- ec_values[ec_values$EC == 10,]
2049      ec50 <- ec_values[ec_values$EC == 50,]
2050
2051      if (nrow(ec10) > 0) {
2052        segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2053        segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2054        abline(v = ec10$Estimate, col = "red", lty = 2)
2055      }
2056      if (nrow(ec50) > 0) {
2057        segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2058        segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2059        abline(v = ec50$Estimate, col = "green", lty = 2)
2060    }
2061
2062    test_results <- TestResult216()
2063    if (!is.null(test_results)) {
2064      noec_values <- test_results$test$p.value > 0.05
2065      loec_values <- test_results$test$p.value <= 0.05
2066      if (any(noec_values, na.rm = TRUE)) {
2067        NOEC <- min(test_results$test$p.value[noec_values], na.rm = TRUE)
2068        abline(v = NOEC, col = "purple", lty = 2)
2069      }
2070      if (any(loec_values, na.rm = TRUE)) {
2071        LOEC <- max(test_results$test$p.value[loec_values], na.rm = TRUE)
2072      }
2073    }
2074  }
2075
```

```

2072         abline(v = LOEC, col = "orange", lty = 2)
2073     }
2074   }
2075 }
2076 }
2077
2078
2079 #218
2080
2081 intest_type218 <- reactive({ 'Essai218' })
2082
2083 validateFile <- function(filename) {
2084   extFile <- tools::file_ext(filename)
2085   validate(need(extFile %in% c("csv", "txt"), "Seuls les fichiers CSV ou TXT sont acceptés."))
2086 }
2087
2088 filedata218 <- eventReactive(input$buttonRunStep218, {
2089   if (intest_type218() == 'Essai218') {
2090     req(input$datafile_ESSAI218)
2091     validateFile(input$datafile_ESSAI218$name)
2092     ff <- input$datafile_ESSAI218
2093     read.csv(file = ff$datapath, header = TRUE)
2094   }
2095 })
2096
2097 output$rawdata218 <- DT::renderDataTable({
2098   filedata218()
2099 })
2100
2101 fitmodel218 <- eventReactive(input$buttonRunStep218, {
2102   data <- filedata218()
2103   if (intest_type218() == 'Essai218') {
2104     fit_function_mortality <- switch(input$model_E218_mortality,
2105                                         ll2 = LL.2(),
2106                                         ll3 = LL.3(),
2107                                         ll4 = LL.4())
2108     fit_function_emergence <- switch(input$model_E218_emergence,
2109                                         ll2 = LL.2(),
2110                                         ll3 = LL.3(),
2111                                         ll4 = LL.4())
2112     fit_function_development <- switch(input$model_E218_development,
2113                                         ll3 = LL.3(),
2114                                         ll4 = LL.4())
2115
2116     fit1 <- drm(dead/total ~ concentration, data = data, fct = fit_function_mortality, type = "binomial")
2117     fit2 <- drm(emerged/total ~ concentration, data = data, fct = fit_function_emergence, type = "binomial")
2118     fit3 <- drm(development ~ concentration, data = data, fct = fit_function_development, type = "continuous")

```

```

2119         fit <- list(fit1 = fit1, fit2 = fit2, fit3 = fit3)
2120         return(fit)
2121     }
2122 }
2123 )
2124
2125 ECx218 <- eventReactive(input$buttonRunStep218, {
2126     fit <- fitmodel218()
2127     fit1 <- fit$fit1
2128     fit2 <- fit$fit2
2129     fit3 <- fit$fit3
2130     if (intest_type218() == 'Essai218') {
2131         XX <- as.numeric(input$ecx_ESSAI218)
2132
2133         drc_df1 <- data.frame(ED(fit1, c(XX), interval = "delta",
2134             display = FALSE), "Slope" = coefficients(fit1)[[1]])
2135         drc_df2 <- data.frame(ED(fit2, c(XX), interval = "delta",
2136             display = FALSE), "Slope" = coefficients(fit2)[[1]])
2137         drc_df3 <- data.frame(ED(fit3, c(XX), interval = "delta",
2138             display = FALSE), "Slope" = coefficients(fit3)[[1]])
2139
2140         colnames_list <- c(paste0('EC', XX), 'Standard_Error', 'Lower_95%CI',
2141             'Upper_95%CI', 'Slope')
2142
2143         colnames(drc_df1) <- colnames_list
2144         colnames(drc_df2) <- colnames_list
2145         colnames(drc_df3) <- colnames_list
2146
2147         drc_df <- rbind(drc_df1, drc_df2, drc_df3)
2148         rownames(drc_df) <- c('Mortality', 'Emergence_ratio',
2149             'Development_rate')
2150
2151         return(drc_df)
2152     }
2153 }
2154
2155 output$drc_result218 <- renderTable({
2156     ECx218()
2157 }, rownames = TRUE)
2158
2159 output$drc_plot218 <- renderPlot({
2160     fit <- fitmodel218()
2161     fit1 <- fit$fit1
2162     fit2 <- fit$fit2
2163     fit3 <- fit$fit3
2164     data <- filedatal218()
2165     ec_values <- ECx218()
2166
2167     TestResult218 <- eventReactive(input$buttonRunStep218, {
2168         data <- filedatal218()
2169         if (intest_type218() == 'Essai218') {
2170             data$concentration <- as.factor(data$concentration)
2171         }
2172     })
2173 }
2174 )
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
2999

```

```

2166     Res_variance <- bartlett.test(development ~ concentration,
2167     data = data)
2168     if (input$test_method_E218_mortality == "CA") {
2169       data_CA <- data %>% group_by(concentration) %>% summarize(
2170         dead = sum(dead), total = sum(total))
2171       Res1 <- data.frame(concentration = data_CA$concentration,
2172         total = data_CA$total, dead = data_CA$dead)
2173       LENGTH <- nrow(data_CA)
2174       for (i in 3:LENGTH) {
2175         Res1[i, 4] <- prop.trend.test(data_CA[1:i, ]$dead, data_
2176           CA[1:i, ]$total)$statistic
2177         Res1[i, 5] <- prop.trend.test(data_CA[1:i, ]$dead, data_
2178           CA[1:i, ]$total)$p.value
2179       }
2180       colnames(Res1) <- c("concentration", "total", "dead", "Chi
2181         _squared", "pvalue")
2182       Res1 <- Res1 %>%
2183         mutate(Asterisk = ifelse(pvalue < 0.05, ifelse(pvalue >
2184           0.01, "*", "**"), ""))
2185     } else if (input$test_method_E218_mortality == "Fisher") {
2186       data_F <- filedatal218() %>% group_by(concentration) %>%
2187         summarize(total = sum(total), dead = sum(dead)) %>%
2188         ungroup()
2189       total <- data_F %>% dplyr::filter(concentration == "0")
2190       %>% dplyr::select(total) %>% as.numeric()
2191       dead_ctrl <- data_F %>% dplyr::filter(concentration == "0")
2192       %>% dplyr::select(dead) %>% as.numeric()
2193       data2 <- data_F %>% mutate(total_ctrl = total, dead_ctrl =
2194         dead_ctrl) %>% dplyr::filter(concentration != "0")
2195       fisher <- function(a, b, c, d) {
2196         dt <- matrix(c(a, b, c, d), ncol = 2)
2197         c(pvalue = fisher.test(dt)$p.value)
2198       }
2199       Res1 <- data2 %>%
2200         rowwise() %>%
2201         mutate(pvalue = fisher(dead, total - dead, dead_ctrl,
2202           total_ctrl - dead_ctrl)) %>% ungroup() %>%
2203         mutate(p_adjusted = p.adjust(pvalue, "holm")) %>%
2204         mutate(Asterisk = ifelse(p_adjusted < 0.05, ifelse(p_
2205           adjusted > 0.01, "*", "**"), ""))
2206     }
2207     if (input$test_method_E218_emergence == "CA") {
2208       data_CA <- data %>% group_by(concentration) %>% summarize(
2209         emerged = sum(emerged), total = sum(total))
2210       Res2 <- data.frame(concentration = data_CA$concentration,
2211         total = data_CA$total, emerged = data_CA$emerged)
2212       LENGTH <- nrow(data_CA)
2213       for (i in 3:LENGTH) {
2214         Res2[i, 4] <- prop.trend.test(data_CA[1:i, ]$emerged,
2215           data_CA[1:i, ]$total)$statistic
2216         Res2[i, 5] <- prop.trend.test(data_CA[1:i, ]$emerged,
2217           data_CA[1:i, ]$total)$p.value

```

```

2201     }
2202     colnames(Res2) <- c("concentration", "total", "emerged", "Chi_squared", "pvalue")
2203     Res2 <- Res2 %>%
2204       mutate(Asterisk = ifelse(pvalue < 0.05, ifelse(pvalue >
2205         0.01, "*", "**"), ""))
2206   } else if (input$test_method_E218_emergence == "Fisher") {
2207     data_F_emer <- filedatal218() %>% group_by(concentration) %>%
2208       summarize(total = sum(total), emerged = sum(emerged)) %>% ungroup()
2209     total <- data_F_emer %>% dplyr::filter(concentration == "0") %>% dplyr::select(total) %>% as.numeric()
2210     emerged_ctrl <- data_F_emer %>% dplyr::filter(concentration == "0") %>% dplyr::select(emerged) %>% as.numeric()
2211     data2_emer <- data_F_emer %>% mutate(total_ctrl = total, EMER_ctrl = emerged_ctrl) %>% dplyr::filter(concentration != "0")
2212     fisher <- function(a, b, c, d) {
2213       dt <- matrix(c(a, b, c, d), ncol = 2)
2214       c(pvalue = fisher.test(dt)$p.value)
2215     }
2216     Res2 <- data2_emer %>% rowwise() %>% mutate(pvalue = fisher(emerged, total - emerged, EMER_ctrl, total_ctrl - EMER_ctrl)) %>% ungroup() %>% mutate(p_adjusted = p.adjust(pvalue, "holm")) %>% mutate(Asterisk = ifelse(p_adjusted < 0.05, ifelse(p_adjusted > 0.01, "*", "**"), ""))
2217   }
2218   if (input$test_method_E218_development == "Dunnett") {
2219     data_raw <- filedatal218()
2220     data_raw$concentration <- as.factor(data_raw$concentration)
2221     fit <- aov( development ~ concentration, data = data_raw)
2222     Res3 <- summary (glht (fit, linfct=mcp (concentration="Dunnett"), alternative="less"))
2223     list("Bartlett's test for development rate (DR)" = Res_variance, "Dunnett's test for DR" = Res3)
2224   } else if (input$test_method_E218_development == "Steel") {
2225     data_raw=fildeatal218()
2226     data_raw$concentration <- as.factor(data_raw$concentration)
2227     Res3 <- steel.test(development ~ concentration, data = data_raw, control = "0", alternative="less") %>% mutate(Asterisk = ifelse(p.value<0.05, ifelse(p.value >0.01, "*", "**"), ""))
2228   }
2229   list("Mortality" = Res1, "Emergence_ratio" = Res2, "Bartlett's test for development rate (DR)" = Res_

```

```

                variance, "Development_rate" = Res3 )
2235 }
2236 }
2237 )
2238
2239 output$test_result218 <- renderPrint({TestResult218()})
2240
2241 if(intest_type218() == 'Essai218') {par(mar=c(5,5,2,2), mfrow=c
2242 (3,1),mgp=c(2.5, 0.7, 0))
2243 plot(fit1, log="x", broken=TRUE, xlab=paste0("concentration",
2244 , input$conc_unit218, ")"), ylab="Mortality",
2245 ylim=c(0,1),lty="dotted",cex=2,cex.axis =2, cex.lab=2)
2246 plot(fit2, log="x", broken=TRUE, xlab=paste0("concentration",
2247 , input$conc_unit218, ")"), ylab="Emergence_ratio",
2248 ylim=c(0,1),lty="dotted",cex=2,cex.axis =2, cex.lab=2)
2249 plot(fit3, log="x", broken=TRUE, xlab=paste0("concentration",
2250 , input$conc_unit218, ")"), ylab="Development_rate",
2251 lty="dotted",cex=2,cex.axis =2, cex.lab=2)}
2252
2253 }
2254
2255 # 221
2256
2257
2258 intest_type221 <- reactive({ 'Essai221' })
2259
2260 validateFile <- function(filename) {
2261   extFile <- tools::file_ext(filename)
2262   validate(need(extFile == c("csv","txt"), "Seuls les fichiers CSV
2263   sont acceptés."))
2264 }
2265
2266
2267 filedata221 <- eventReactive(input$buttonRunStep221, {
2268   if(intest_type221() == 'Essai221') {
2269     req(input$datafile_ESSAI221)
2270     validateFile(input$datafile_ESSAI221)
2271     ff <- input$datafile_ESSAI221
2272     read.csv(file = ff$datapath, header = TRUE)
2273   }
2274 })
2275
2276
2277 output$rawdata221 <- DT::renderDataTable({
2278   filedata221()
2279 })
2280
2281
2282 fitmodel221 <- eventReactive(input$buttonRunStep221, {
2283   data <- filedata221()
2284   if(intest_type221() == 'Essai221') {
2285     if(input$model_ESSAI221 == '112') {
2286       fit <- drm(response ~ as.numeric(concentration), data = data
2287 , fct = LL.2())
2288     } else if(input$model_ESSAI221 == '113') {
2289       fit <- drm(response ~ as.numeric(concentration), data = data
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3388
3389
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3398
3399
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3438
3439
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3448
3449
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3478
3479
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3488
3489
3489
3490
3491
3492
3493
3494
3495
3496
3497
3497
3498
3498
3499
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3538
3539
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3548
3549
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3578
3579
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3588
3589
3589
3590
3591
3592
3593
3594
3595
3596
3597
3597
3598
3598
3599
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3638
3639
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3648
3649
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3678
3679
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3688
3689
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3698
3699
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3738
3739
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3748
3749
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3778
3779
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3788
3789
3789
3790
3791
3792
3793
3794
3795
3796
3797
3797
3798
3798
3799
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3838
3839
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3848
3849
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3878
3879
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3888
3889
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3898
3899
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3938
3939
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3948
3949
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3978
3979
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3988
3989
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3998
3999
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4038
4039
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4048
4049
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4078
4079
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4088
4089
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4098
4099
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4138
4139
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4148
4149
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4178
4179
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4188
4189
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
419
```

```

            , fct = LL.3())
2280 } else if(input$model_ESSAI221 == '114') {
2281   fit <- drm(response ~ as.numeric(concentration), data = data
2282   , fct = LL.4())
2283 }
2284 return(fit)
2285 }
2286 }
2287 ECx221 <- eventReactive(input$buttonRunStep221, {
2288   fit <- fitmodel221()
2289   if(intest_type221() == 'Essai221') {
2290     XX <- as.numeric(input$ecx_ESSAI221)
2291     ec_values <- ED(fit, c(XX, 10, 50), interval = "delta",
2292     display = FALSE)
2293
2294   drc_df <- data.frame(
2295     EC = c(XX, 10, 50),
2296     Estimate = ec_values[,1],
2297     'Standard Error' = ec_values[,2],
2298     'Lower 95% CI' = ec_values[,3],
2299     'Upper 95% CI' = ec_values[,4]
2300   )
2301
2302   drc_df$Pente <- coefficients(fit)[[1]]
2303
2304   return(drc_df)
2305 }
2306 }
2307
2308 output$drc_result221 <- renderTable({
2309   ECx221()
2310 }, rownames = TRUE)
2311
2312 output$drc_plot <- renderPlot({
2313   fit <- fitmodel221()
2314   data <- filedatal221()
2315   ec_values <- ECx221()
2316
2317   TestResult221 <- eventReactive(input$buttonRunStep221, {
2318     data <- filedatal221()
2319     if(intest_type221() == 'Essai221') {
2320       data$concentration <- as.factor(data$concentration)
2321       fit <- lm(response ~ concentration, data = data)
2322       Res <- summary(glht(fit, linfct = mcp(concentration = "
2323         Dunnett"), alternative = "less"))
2324       return(Res)
2325     }
2326   })
2327   output$test_result221 <- renderPrint({TestResult221()})

```

```

2328
2329  if(intest_type221() == 'Essai221') {
2330    plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit221, ")"), ylab = "Réponse")
2331
2332  ec10 <- ec_values[ec_values$EC == 10,]
2333  ec50 <- ec_values[ec_values$EC == 50,]
2334
2335  if (nrow(ec10) > 0) {
2336    segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2337    segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2338    abline(v = ec10$Estimate, col = "red", lty = 2)
2339  }
2340  if (nrow(ec50) > 0) {
2341    segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2342    segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2343    abline(v = ec50$Estimate, col = "green", lty = 2)
2344  }
2345
2346  test_results <- TestResult221()
2347  if (!is.null(test_results)) {
2348    noec_values <- test_results$test$p.value > 0.05
2349    loec_values <- test_results$test$p.value <= 0.05
2350    if (any(noec_values, na.rm = TRUE)) {
2351      NOEC <- min(test_results$test$p.value[noec_values], na.rm = TRUE)
2352      abline(v = NOEC, col = "purple", lty = 2)
2353    }
2354    if (any(loec_values, na.rm = TRUE)) {
2355      LOEC <- max(test_results$test$p.value[loec_values], na.rm = TRUE)
2356      abline(v = LOEC, col = "orange", lty = 2)
2357    }
2358  }
2359 }
2360 }
2361
2362 # 222
2363
2364 intest_type222 <- reactive({ 'Essai222' })
2365
2366 validateFile <- function(filename) {
2367   extFile <- tools::file_ext(filename)
2368   validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV sont acceptés."))
2369 }
2370
2371
2372 filedata222 <- eventReactive(input$buttonRunStep222, {
2373   if(intest_type222() == 'Essai222') {
2374     req(input$datafile_ESSAI222)

```

```

2375     validateFile(input$datafile_ESSAI222)
2376     ff <- input$datafile_ESSAI222
2377     read.csv(file = ff$datapath, header = TRUE)
2378   }
2379 }
2380
2381 output$rawdata222 <- DT::renderDataTable({
2382   filedata222()
2383 })
2384
2385 fitmodel222 <- eventReactive(input$buttonRunStep222, {
2386   data <- filedata222()
2387   if(intest_type222() == 'Essai222') {
2388     if(input$model_ESSAI222 == 'lm') {
2389       fit <- lm(response ~ as.numeric(concentration), data = data)
2390     } else if(input$model_ESSAI222 == 'nls') {
2391       fit <- nls(response ~ fct(concentration), data = data)
2392     } else if(input$model_ESSAI222 == 'lme') {
2393       fit <- lme(response ~ as.numeric(concentration), random = ~
2394         1 | group, data = data)
2395     }
2396     return(fit)
2397   }
2398 }
2399 ECx222 <- eventReactive(input$buttonRunStep222, {
2400   fit <- fitmodel222()
2401   if(intest_type222() == 'Essai222') {
2402     XX <- as.numeric(input$ecx_ESSAI222)
2403     ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
2404       FALSE)
2405
2406     drc_df <- data.frame(
2407       EC = c(XX, 50),
2408       Estimate = ec_values[,1],
2409       'Standard Error' = ec_values[,2],
2410       'Lower 95% CI' = ec_values[,3],
2411       'Upper 95% CI' = ec_values[,4]
2412     )
2413
2414     drc_df$Pente <- coefficients(fit)[[1]]
2415
2416     return(drc_df)
2417   }
2418 }
2419
2420 output$drc_result222 <- renderTable({
2421   ECx222()
2422   }, rownames = TRUE)
2423
2424 output$drc_plot <- renderPlot({

```

```

2425 fit <- fitmodel222()
2426 data <- filedatal222()
2427 ec_values <- ECx222()
2428
2429 TestResult222 <- eventReactive(input$buttonRunStep222, {
2430   data <- filedatal222()
2431   if(intest_type222() == 'Essai222') {
2432     data$concentration <- as.factor(data$concentration)
2433     fit <- lm(response ~ concentration, data = data)
2434     Res <- anova(fit)
2435     return(Res)
2436   }
2437 })
2438
2439 output$test_result <- renderPrint({TestResult222()})
2440
2441 if(intest_type222() == 'Essai222') {
2442   plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit222, ")), ylab = "Réponse")
2443
2444   ec10 <- ec_values[ec_values$EC == 10,]
2445   ec50 <- ec_values[ec_values$EC == 50,]
2446
2447   if (nrow(ec10) > 0) {
2448     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2449     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2450     abline(v = ec10$Estimate, col = "red", lty = 2)
2451   }
2452   if (nrow(ec50) > 0) {
2453     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2454     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2455     abline(v = ec50$Estimate, col = "green", lty = 2)
2456   }
2457
2458   test_results <- TestResult222()
2459   if (!is.null(test_results)) {
2460     noec_values <- test_results$test$p.value > 0.05
2461     loec_values <- test_results$test$p.value <= 0.05
2462     if (any(noec_values, na.rm = TRUE)) {
2463       NOEC <- min(test_results$test$p.value[noec_values], na.rm = TRUE)
2464       abline(v = NOEC, col = "purple", lty = 2)
2465     }
2466     if (any(loec_values, na.rm = TRUE)) {
2467       LOEC <- max(test_results$test$p.value[loec_values], na.rm = TRUE)
2468       abline(v = LOEC, col = "orange", lty = 2)
2469     }
2470   }
2471 }
2472 })

```

```

2473
2474
2475 #223
2476 intest_type223 <- reactive({ 'Essai223' })
2477
2478 validateFile <- function(filename) {
2479   extFile <- tools::file_ext(filename)
2480   validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
2481   sont acceptés."))
2482 }
2483
2484 filedata223 <- eventReactive(input$buttonRunStep223, {
2485   if(intest_type223() == 'Essai223') {
2486     req(input$datafile_ESSAI223)
2487     validateFile(input$datafile_ESSAI223)
2488     ff <- input$datafile_ESSAI223
2489     read.csv(file = ff$datapath, header = TRUE)
2490   }
2491 }
2492
2493 output$rawdata223 <- DT::renderDataTable({
2494   filedata223()
2495 }
2496
2497 fitmodel223 <- eventReactive(input$buttonRunStep223, {
2498   data <- filedata223()
2499   if(intest_type223() == 'Essai223') {
2500     if(input$model_ESSAI223 == 'probit') {
2501       fit <- drm(response ~ as.numeric(dose), data = data, fct =
2502             PROBIT())
2503     } else if(input$model_ESSAI223 == 'logit') {
2504       fit <- drm(response ~ as.numeric(dose), data = data, fct =
2505             LOGIT())
2506     }
2507     return(fit)
2508   }
2509 }
2510
2511 ECx223 <- eventReactive(input$buttonRunStep223, {
2512   fit <- fitmodel223()
2513   if(intest_type223() == 'Essai223') {
2514     XX <- as.numeric(input$ecx_ESSAI223)
2515     ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
2516       FALSE)
2517
2518     drc_df <- data.frame(
2519       EC = c(XX, 50),
2520       Estimate = ec_values[,1],
2521       'Standard Error' = ec_values[,2],
2522       'Lower 95% CI' = ec_values[,3],
2523       'Upper 95% CI' = ec_values[,4]
2524     )
2525   }
2526 }
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3279
3280
3281
3282
3283
3284
3285
3286
3287
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3379
3380
3381
3382
3383
3384
3385
3386
3387
3387
3388
3389
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3398
3399
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3588
3589
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3598
3599
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3688
3689
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3698
3699
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3778
3779
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3788
3789
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3798
3799
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3878
3879
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3888
3889
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3898
3899
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3998
3999
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4078
4079
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4098
4099
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4278
4279
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4378
4379
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4398
4399
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4459
4460
4461
4462
4463
4464
4465
4466

```

```

2521     drc_df$Pente <- coefficients(fit)[[1]]
2522
2523     return(drc_df)
2524   }
2525 }
2526 )
2527
2528
2529 output$drc_result223 <- renderTable({
2530   ECx223()
2531 }, rownames = TRUE)
2532
2533 output$drc_plot <- renderPlot({
2534   fit <- fitmodel223()
2535   data <- filedatal223()
2536   ec_values <- ECx223()
2537
2538   TestResult223 <- eventReactive(input$buttonRunStep223, {
2539     data <- filedatal223()
2540     if(intest_type223() == 'Essai223') {
2541       data$dose <- as.factor(data$dose)
2542       fit <- lm(response ~ dose, data = data)
2543       Res <- summary(glht(fit, linfct = mcp(dose = "Dunnett"),
2544                         alternative = "less"))
2545       return(Res)
2546     }
2547   })
2548
2549   output$test_result <- renderPrint({TestResult223()})
2550
2551   if(intest_type223() == 'Essai223') {
2552     plot(fit, log = "x", broken = TRUE, xlab = paste0("Dose", " ",
2553       input$conc_unit223, ")), ylab = "Réponse")
2554
2555     ec10 <- ec_values[ec_values$EC == 10,]
2556     ec50 <- ec_values[ec_values$EC == 50,]
2557
2558     if (nrow(ec10) > 0) {
2559       segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2560       segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2561       abline(v = ec10$Estimate, col = "red", lty = 2)
2562     }
2563     if (nrow(ec50) > 0) {
2564       segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2565       segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2566       abline(v = ec50$Estimate, col = "green", lty = 2)
2567     }
2568
2569     test_results <- TestResult223()
2570     if (!is.null(test_results)) {
2571       noec_values <- test_results$test$p.value > 0.05
2572       loec_values <- test_results$test$p.value <= 0.05

```

```

2571     if (any(noec_values, na.rm = TRUE)) {
2572       NOEC <- min(test_results$test$p.value[noec_values], na.rm
2573                     = TRUE)
2574       abline(v = NOEC, col = "purple", lty = 2)
2575     }
2576     if (any(loec_values, na.rm = TRUE)) {
2577       LOEC <- max(test_results$test$p.value[loec_values], na.rm
2578                     = TRUE)
2579       abline(v = LOEC, col = "orange", lty = 2)
2580     }
2581   }
2582 }
2583
2584
2585 # 225
2586
2587 intest_type225 <- reactive({ 'Essai225' })
2588
2589 validateFile <- function(filename) {
2590   extFile <- tools::file_ext(filename)
2591   validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
2592   sont acceptés."))
2593 }
2594
2595 filedatal225 <- eventReactive(input$buttonRunStep225, {
2596   if(intest_type225() == 'Essai225') {
2597     req(input$datafile_ESSAI225)
2598     validateFile(input$datafile_ESSAI225)
2599     ff <- input$datafile_ESSAI225
2600     read.csv(file = ff$datapath, header = TRUE)
2601   }
2602 })
2603
2604 output$rawdata225 <- DT::renderDataTable({
2605   filedatal225()
2606 })
2607
2608 fitmodel225 <- eventReactive(input$buttonRunStep225, {
2609   data <- filedatal225()
2610   if(intest_type225() == 'Essai225') {
2611     if(input$model_ESSAI225 == 'logit') {
2612       fit <- drm(response ~ as.numeric(concentration), data = data
2613                   , fct = LOGIT())
2614     }
2615     return(fit)
2616   }
2617 }
2618 )
2619
2620 ECx225 <- eventReactive(input$buttonRunStep225, {
2621   fit <- fitmodel225()

```

```

2619     if(intest_type225() == 'Essai225') {
2620       XX <- as.numeric(input$ecx_ESSAI225)
2621       ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
2622         FALSE)
2623
2624       drc_df <- data.frame(
2625         EC = c(XX, 50),
2626         Estimate = ec_values[,1],
2627         'Standard Error' = ec_values[,2],
2628         'Lower 95% CI' = ec_values[,3],
2629         'Upper 95% CI' = ec_values[,4]
2630       )
2631
2632       drc_df$Pente <- coefficients(fit)[[1]]
2633
2634       return(drc_df)
2635     }
2636   )
2637
2638   output$drc_result225 <- renderTable({
2639     ECx225()
2640   }, rownames = TRUE)
2641
2642   output$drc_plot <- renderPlot({
2643     fit <- fitmodel225()
2644     data <- filedatal225()
2645     ec_values <- ECx225()
2646
2647     TestResult225 <- eventReactive(input$buttonRunStep225, {
2648       data <- filedatal225()
2649       if(intest_type225() == 'Essai225') {
2650         data$concentration <- as.factor(data$concentration)
2651         fit <- lm(response ~ concentration, data = data)
2652         Res <- summary(glht(fit, linfct = mcp(concentration = "Dunnett"), alternative = "less"))
2653         return(Res)
2654       }
2655     })
2656
2657     output$test_result <- renderPrint({TestResult225()})
2658
2659     if(intest_type225() == 'Essai225') {
2660       plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit225, ")), ylab = "Réponse")
2661
2662       ec10 <- ec_values[ec_values$EC == 10,]
2663       ec50 <- ec_values[ec_values$EC == 50,]
2664
2665       if (nrow(ec10) > 0) {
2666         segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)

```

```

2667     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2668     abline(v = ec10$Estimate, col = "red", lty = 2)
2669   }
2670   if (nrow(ec50) > 0) {
2671     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2672     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2673     abline(v = ec50$Estimate, col = "green", lty = 2)
2674   }
2675
2676   test_results <- TestResult225()
2677   if (!is.null(test_results)) {
2678     noec_values <- test_results$test$p.value > 0.05
2679     loec_values <- test_results$test$p.value <= 0.05
2680     if (any(noec_values, na.rm = TRUE)) {
2681       NOEC <- min(test_results$test$p.value[noec_values], na.rm
2682                     = TRUE)
2683       abline(v = NOEC, col = "purple", lty = 2)
2684     }
2685     if (any(loec_values, na.rm = TRUE)) {
2686       LOEC <- max(test_results$test$p.value[loec_values], na.rm
2687                     = TRUE)
2688       abline(v = LOEC, col = "orange", lty = 2)
2689     }
2690   }
2691
2692
2693 # 226
2694
2695 intest_type226 <- reactive({ 'Essai226' })
2696
2697 validateFile <- function(filename) {
2698   extFile <- tools::file_ext(filename)
2699   validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
2700   sont acceptés."))
2701 }
2702
2703 filedata226 <- eventReactive(input$buttonRunStep226, {
2704   if(intest_type226() == 'Essai226') {
2705     req(input$datafile_ESSAI226)
2706     validateFile(input$datafile_ESSAI226)
2707     ff <- input$datafile_ESSAI226
2708     read.csv(file = ff$datapath, header = TRUE)
2709   }
2710 }
2711
2712 output$rawdata226 <- DT::renderDataTable({
2713   filedata226()
2714 })
2715
2716 fitmodel226 <- eventReactive(input$buttonRunStep226, {

```

```

2716     data <- filedatal226()
2717     if(intest_type226() == 'Essai226') {
2718       if(input$model_ESSAI226 == 'lm') {
2719         fit <- lm(response ~ as.numeric(concentration), data = data)
2720       } else if(input$model_ESSAI226 == 'nls') {
2721         fit <- nls(response ~ fct(concentration), data = data)
2722       } else if(input$model_ESSAI226 == 'lme') {
2723         fit <- lme(response ~ as.numeric(concentration), random = ~
2724           1 | group, data = data)
2725       }
2726       return(fit)
2727     }
2728   })
2729
2730   ECx226 <- eventReactive(input$buttonRunStep226, {
2731     fit <- fitmodel226()
2732     if(intest_type226() == 'Essai226') {
2733       XX <- as.numeric(input$ecx_ESSAI226)
2734       ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
2735         FALSE)
2736
2737       drc_df <- data.frame(
2738         EC = c(XX, 50),
2739         Estimate = ec_values[,1],
2740         'Standard Error' = ec_values[,2],
2741         'Lower 95% CI' = ec_values[,3],
2742         'Upper 95% CI' = ec_values[,4]
2743       )
2744
2745       drc_df$Pente <- coefficients(fit)[[1]]
2746
2747       return(drc_df)
2748     }
2749   })
2750
2751   output$drc_result226 <- renderTable({
2752     ECx226()
2753   }, rownames = TRUE)
2754
2755   output$drc_plot <- renderPlot({
2756     fit <- fitmodel226()
2757     data <- filedatal226()
2758     ec_values <- ECx226()
2759
2760     TestResult226 <- eventReactive(input$buttonRunStep226, {
2761       data <- filedatal226()
2762       if(intest_type226() == 'Essai226') {
2763         data$concentration <- as.factor(data$concentration)
2764         fit <- lm(response ~ concentration, data = data)
2765         Res <- anova(fit)
2766         return(Res)
2767       }
2768     })
2769   })

```

```

2766   })
2767
2768   output$test_result <- renderPrint({TestResult226()})
2769
2770   if(intest_type226() == 'Essai226') {
2771     plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit226, ")"), ylab = "Réponse")
2772
2773   ec10 <- ec_values[ec_values$EC == 10,]
2774   ec50 <- ec_values[ec_values$EC == 50,]
2775
2776   if (nrow(ec10) > 0) {
2777     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2778     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2779     abline(v = ec10$Estimate, col = "red", lty = 2)
2780   }
2781   if (nrow(ec50) > 0) {
2782     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2783     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2784     abline(v = ec50$Estimate, col = "green", lty = 2)
2785   }
2786
2787   test_results <- TestResult226()
2788   if (!is.null(test_results)) {
2789     noec_values <- test_results$test$p.value > 0.05
2790     loec_values <- test_results$test$p.value <= 0.05
2791     if (any(noec_values, na.rm = TRUE)) {
2792       NOEC <- min(test_results$test$p.value[noec_values], na.rm = TRUE)
2793       abline(v = NOEC, col = "purple", lty = 2)
2794     }
2795     if (any(loec_values, na.rm = TRUE)) {
2796       LOEC <- max(test_results$test$p.value[loec_values], na.rm = TRUE)
2797       abline(v = LOEC, col = "orange", lty = 2)
2798     }
2799   }
2800 }
2801
2802
2803
2804 # 227
2805
2806 intest_type227 <- reactive({ 'Essai227' })
2807
2808 validateFile <- function(filename) {
2809   extFile <- tools::file_ext(filename)
2810   validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV sont acceptés."))
2811 }
2812

```

```

2813  filedatal227 <- eventReactive(input$buttonRunStep227, {
2814    if(intest_type227() == 'Essai227') {
2815      req(input$datafile_ESSAI227)
2816      validateFile(input$datafile_ESSAI227)
2817      ff <- input$datafile_ESSAI227
2818      read.csv(file = ff$datapath, header = TRUE)
2819    }
2820  })
2821
2822  output$rawdata227 <- DT::renderDataTable({
2823    filedatal227()
2824  })
2825
2826  fitmodel227 <- eventReactive(input$buttonRunStep227, {
2827    data <- filedatal227()
2828    if(intest_type227() == 'Essai227') {
2829      if(input$model_ESSAI227 == '112') {
2830        fit <- drm(response ~ as.numeric(concentration), data = data
2831        , fct = LL.2())
2832      } else if(input$model_ESSAI227 == '113') {
2833        fit <- drm(response ~ as.numeric(concentration), data = data
2834        , fct = LL.3())
2835      } else if(input$model_ESSAI227 == '114') {
2836        fit <- drm(response ~ as.numeric(concentration), data = data
2837        , fct = LL.4())
2838      }
2839      return(fit)
2840    }
2841  })
2842
2843  ECx227 <- eventReactive(input$buttonRunStep227, {
2844    fit <- fitmodel227()
2845    if(intest_type227() == 'Essai227') {
2846      XX <- as.numeric(input$ecx_ESSAI227)
2847      ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
2848        FALSE)
2849
2850      drc_df <- data.frame(
2851        EC = c(XX, 50),
2852        Estimate = ec_values[,1],
2853        'Standard Error' = ec_values[,2],
2854        'Lower 95% CI' = ec_values[,3],
2855        'Upper 95% CI' = ec_values[,4]
2856      )
2857
2858      drc_df$Pente <- coefficients(fit)[[1]]
2859
2860      return(drc_df)
2861    }
2862  })
2863
2864  output$drc_result227 <- renderTable({

```

```

2861   ECx227()
2862 }, rownames = TRUE)
2863
2864 output$drc_plot <- renderPlot({
2865   fit <- fitmodel227()
2866   data <- filedatal227()
2867   ec_values <- ECx227()
2868
2869 TestResult227 <- eventReactive(input$buttonRunStep227, {
2870   data <- filedatal227()
2871   if(intest_type227() == 'Essai227') {
2872     data$concentration <- as.factor(data$concentration)
2873     fit <- lm(response ~ concentration, data = data)
2874     Res <- anova(fit)
2875     return(Res)
2876   }
2877 })
2878
2879 output$test_result <- renderPrint({TestResult227()})
2880
2881 if(intest_type227() == 'Essai227') {
2882   plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit227, ")"), ylab = "Réponse")
2883
2884   ec10 <- ec_values[ec_values$EC == 10,]
2885   ec50 <- ec_values[ec_values$EC == 50,]
2886
2887   if (nrow(ec10) > 0) {
2888     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2889     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2890     abline(v = ec10$Estimate, col = "red", lty = 2)
2891   }
2892   if (nrow(ec50) > 0) {
2893     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2894     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
2895     abline(v = ec50$Estimate, col = "green", lty = 2)
2896   }
2897
2898   test_results <- TestResult227()
2899   if (!is.null(test_results)) {
2900     noec_values <- test_results$test$p.value > 0.05
2901     loec_values <- test_results$test$p.value <= 0.05
2902     if (any(noec_values, na.rm = TRUE)) {
2903       NOEC <- min(test_results$test$p.value[noec_values], na.rm = TRUE)
2904       abline(v = NOEC, col = "purple", lty = 2)
2905     }
2906     if (any(loec_values, na.rm = TRUE)) {
2907       LOEC <- max(test_results$test$p.value[loec_values], na.rm = TRUE)
2908       abline(v = LOEC, col = "orange", lty = 2)

```



```

2958     Estimate = ec_values[,1],
2959     'Standard Error' = ec_values[,2],
2960     'Lower 95% CI' = ec_values[,3],
2961     'Upper 95% CI' = ec_values[,4]
2962   )
2963
2964   drc_df$Pente <- coefficients(fit)[[1]]
2965
2966   return(drc_df)
2967 }
2968 )
2969
2970 output$drc_result232 <- renderTable({
2971   ECx232()
2972 }, rownames = TRUE)
2973
2974 output$drc_plot <- renderPlot({
2975   fit <- fitmodel232()
2976   data <- filedatal232()
2977   ec_values <- ECx232()
2978
2979   TestResult232 <- eventReactive(input$buttonRunStep232, {
2980     data <- filedatal232()
2981     if(intest_type232() == 'Essai232') {
2982       data$concentration <- as.factor(data$concentration)
2983       fit <- lm(response ~ concentration, data = data)
2984       Res <- anova(fit)
2985       return(Res)
2986     }
2987   })
2988
2989   output$test_result <- renderPrint({TestResult232()})
2990
2991   if(intest_type232() == 'Essai232') {
2992     plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit232, ")), ylab = "Réponse")
2993
2994     ec10 <- ec_values[ec_values$EC == 10,]
2995     ec50 <- ec_values[ec_values$EC == 50,]
2996
2997     if (nrow(ec10) > 0) {
2998       segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
2999       segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3000       abline(v = ec10$Estimate, col = "red", lty = 2)
3001     }
3002     if (nrow(ec50) > 0) {
3003       segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3004       segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3005       abline(v = ec50$Estimate, col = "green", lty = 2)
3006     }
3007   }

```

```

3008     test_results <- TestResult232()
3009     if (!is.null(test_results)) {
3010       noec_values <- test_results$test$p.value > 0.05
3011       loec_values <- test_results$test$p.value <= 0.05
3012       if (any(noec_values, na.rm = TRUE)) {
3013         NOEC <- min(test_results$test$p.value[noec_values], na.rm
3014                       = TRUE)
3015         abline(v = NOEC, col = "purple", lty = 2)
3016       }
3017       if (any(loec_values, na.rm = TRUE)) {
3018         LOEC <- max(test_results$test$p.value[loec_values], na.rm
3019                       = TRUE)
3020         abline(v = LOEC, col = "orange", lty = 2)
3021       }
3022     }
3023   }
3024
3025
3026 #239
3027
3028 intest_type239 <- reactive({ 'Essai239' })
3029
3030 validateFile <- function(filename) {
3031   extFile <- tools::file_ext(filename)
3032   validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
3033   sont acceptés."))
3034 }
3035
3036 filedata239 <- eventReactive(input$buttonRunStep239, {
3037   if(intest_type239() == 'Essai239') {
3038     req(input$datafile_ESSAI239)
3039     validateFile(input$datafile_ESSAI239)
3040     ff <- input$datafile_ESSAI239
3041     read.csv(file = ff$datapath, header = TRUE)
3042   }
3043 }
3044
3045 output$rawdata239 <- DT::renderDataTable({
3046   filedata239()
3047 }
3048
3049 fitmodel239 <- eventReactive(input$buttonRunStep239, {
3050   data <- filedata239()
3051   if(intest_type239() == 'Essai239') {
3052     if(input$model_ESSAI239 == 'probit') {
3053       fit <- drm(response ~ as.numeric(dose), data = data, fct =
3054                   PROBIT())
3055     } else if(input$model_ESSAI239 == 'logit') {
3056       fit <- drm(response ~ as.numeric(dose), data = data, fct =
3057                   LOGIT())

```

```

3055     }
3056     return(fit)
3057   }
3058 }
3059
3060 ECx239 <- eventReactive(input$buttonRunStep239, {
3061   fit <- fitmodel239()
3062   if(intest_type239() == 'Essai239') {
3063     XX <- as.numeric(input$ecx_ESSAI239)
3064     ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
3065                   FALSE)
3066
3067     drc_df <- data.frame(
3068       EC = c(XX, 50),
3069       Estimate = ec_values[,1],
3070       'Standard Error' = ec_values[,2],
3071       'Lower 95% CI' = ec_values[,3],
3072       'Upper 95% CI' = ec_values[,4]
3073     )
3074
3075     drc_df$Pente <- coefficients(fit)[[1]]
3076
3077     return(drc_df)
3078   }
3079 }
3080
3081 output$drc_result239 <- renderTable({
3082   ECx239()
3083 }, rownames = TRUE)
3084
3085 output$drc_plot <- renderPlot({
3086   fit <- fitmodel239()
3087   data <- filedatal239()
3088   ec_values <- ECx239()
3089
3090   TestResult239 <- eventReactive(input$buttonRunStep239, {
3091     data <- filedatal239()
3092     if(intest_type239() == 'Essai239') {
3093       data$dose <- as.factor(data$dose)
3094       fit <- lm(response ~ dose, data = data)
3095       Res <- summary(glht(fit, linfct = mcp(dose = "Dunnett"),
3096                           alternative = "less"))
3097       return(Res)
3098     }
3099   })
3100
3101   output$test_result <- renderPrint({TestResult239()})
3102
3103   if(intest_type239() == 'Essai239') {
3104     plot(fit, log = "x", broken = TRUE, xlab = paste0("Dose", " ",
3105           input$conc_unit239, " ")), ylab = "Réponse")

```

```

3104
3105     ec10 <- ec_values[ec_values$EC == 10,]
3106     ec50 <- ec_values[ec_values$EC == 50,]
3107
3108     if (nrow(ec10) > 0) {
3109         segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3110         segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3111         abline(v = ec10$Estimate, col = "red", lty = 2)
3112     }
3113     if (nrow(ec50) > 0) {
3114         segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3115         segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3116         abline(v = ec50$Estimate, col = "green", lty = 2)
3117     }
3118
3119     test_results <- TestResult239()
3120     if (!is.null(test_results)) {
3121         noec_values <- test_results$test$p.value > 0.05
3122         loec_values <- test_results$test$p.value <= 0.05
3123         if (any(noec_values, na.rm = TRUE)) {
3124             NOEC <- min(test_results$test$p.value[noec_values], na.rm
3125                         = TRUE)
3126             abline(v = NOEC, col = "purple", lty = 2)
3127         }
3128         if (any(loec_values, na.rm = TRUE)) {
3129             LOEC <- max(test_results$test$p.value[loec_values], na.rm
3130                         = TRUE)
3131             abline(v = LOEC, col = "orange", lty = 2)
3132         }
3133     }
3134
3135
3136
3137 # 245
3138
3139     intest_type245 <- reactive({ 'Essai245' })
3140
3141     validateFile <- function(filename) {
3142         extFile <- tools::file_ext(filename)
3143         validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
3144                         sont acceptés."))
3145     }
3146
3147     filedatal245 <- eventReactive(input$buttonRunStep245, {
3148         if(intest_type245() == 'Essai245') {
3149             req(input$datafile_ESSAI245)
3150             validateFile(input$datafile_ESSAI245)
3151             ff <- input$datafile_ESSAI245
3152             read.csv(file = ff$datapath, header = TRUE)
3153         }

```

```

3153   })
3154
3155   output$rawdata245 <- DT::renderDataTable({
3156     filedata245()
3157   })
3158
3159   fitmodel245 <- eventReactive(input$buttonRunStep245, {
3160     data <- filedata245()
3161     if(intest_type245() == 'Essai245') {
3162       fit <- coxph(Surv(time, response) ~ dose, data = data)
3163       return(fit)
3164     }
3165   })
3166
3167   ECx245 <- eventReactive(input$buttonRunStep245, {
3168     fit <- fitmodel245()
3169     if(intest_type245() == 'Essai245') {
3170       XX <- as.numeric(input$ecx_ESSAI245)
3171       ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
3172         FALSE)
3173
3174       drc_df <- data.frame(
3175         EC = c(XX, 50),
3176         Estimate = ec_values[,1],
3177         'Standard Error' = ec_values[,2],
3178         'Lower 95% CI' = ec_values[,3],
3179         'Upper 95% CI' = ec_values[,4]
3180       )
3181
3182       drc_df$Pente <- coefficients(fit)[[1]]
3183
3184       return(drc_df)
3185     }
3186   })
3187
3188   output$drc_result245 <- renderTable({
3189     ECx245()
3190   }, rownames = TRUE)
3191
3192   output$drc_plot <- renderPlot({
3193     fit <- fitmodel245()
3194     data <- filedata245()
3195     ec_values <- ECx245()
3196
3197     TestResult245 <- eventReactive(input$buttonRunStep245, {
3198       data <- filedata245()
3199       if(intest_type245() == 'Essai245') {
3200         data$dose <- as.factor(data$dose)
3201         fit <- lm(response ~ dose, data = data)
3202         Res <- summary(glht(fit, linfct = mcp(dose = "Dunnett"),
3203           alternative = "less"))

```

```

3203     return(Res)
3204   }
3205 }
3206
3207 output$test_result <- renderPrint({TestResult245()})
3208
3209 if(intest_type245() == 'Essai245') {
3210   plot(fit, log = "x", broken = TRUE, xlab = paste0("Dose", " ",
3211     input$conc_unit245, ")), ylab = "Réponse")
3212
3213   ec10 <- ec_values[ec_values$EC == 10,]
3214   ec50 <- ec_values[ec_values$EC == 50,]
3215
3216   if (nrow(ec10) > 0) {
3217     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3218     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3219     abline(v = ec10$Estimate, col = "red", lty = 2)
3220   }
3221   if (nrow(ec50) > 0) {
3222     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3223     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3224     abline(v = ec50$Estimate, col = "green", lty = 2)
3225   }
3226
3227   test_results <- TestResult245()
3228   if (!is.null(test_results)) {
3229     noec_values <- test_results$test$p.value > 0.05
3230     loec_values <- test_results$test$p.value <= 0.05
3231     if (any(noec_values, na.rm = TRUE)) {
3232       NOEC <- min(test_results$test$p.value[noec_values], na.rm
3233                     = TRUE)
3234       abline(v = NOEC, col = "purple", lty = 2)
3235     }
3236     if (any(loec_values, na.rm = TRUE)) {
3237       LOEC <- max(test_results$test$p.value[loec_values], na.rm
3238                     = TRUE)
3239       abline(v = LOEC, col = "orange", lty = 2)
3240     }
3241   }
3242
3243
3244 # 246_247
3245
3246 intest_type246247 <- reactive({ 'Essai246247' })
3247
3248 validateFile <- function(filename) {
3249   extFile <- tools::file_ext(filename)
3250   validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
3251   sont acceptés."))

```

```

3251  }
3252
3253  filedata246247 <- eventReactive(input$buttonRunStep246247, {
3254    if(intest_type246247() == 'Essai246247') {
3255      req(input$datafile_ESSAI246247)
3256      validateFile(input$datafile_ESSAI246247)
3257      ff <- input$datafile_ESSAI246247
3258      read.csv(file = ff$datapath, header = TRUE)
3259    }
3260  })
3261
3262  output$rawdata246247 <- DT::renderDataTable({
3263    filedata246247()
3264  })
3265
3266  fitmodel246247 <- eventReactive(input$buttonRunStep246247, {
3267    data <- filedata246247()
3268    if(intest_type246247() == 'Essai246247') {
3269      if(input$model_ESSAI246247 == 'probit') {
3270        fit <- drm(response ~ as.numeric(dose), data = data, fct =
3271          PROBIT())
3272      } else if(input$model_ESSAI246247 == 'logit') {
3273        fit <- drm(response ~ as.numeric(dose), data = data, fct =
3274          LOGIT())
3275      }
3276      return(fit)
3277    }
3278  })
3279
3280  ECx246247 <- eventReactive(input$buttonRunStep246247, {
3281    fit <- fitmodel246247()
3282    if(intest_type246247() == 'Essai246247') {
3283      XX <- as.numeric(input$ecx_ESSAI246247)
3284      ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
3285        FALSE)
3286
3287      drc_df <- data.frame(
3288        EC = c(XX, 50),
3289        Estimate = ec_values[,1],
3290        'Standard Error' = ec_values[,2],
3291        'Lower 95% CI' = ec_values[,3],
3292        'Upper 95% CI' = ec_values[,4]
3293      )
3294
3295      drc_df$Pente <- coefficients(fit)[[1]]
3296
3297      return(drc_df)
3298    }
3299  })
3300
3301  output$drc_result246247 <- renderTable({

```

```

3300   ECx246247()
3301 }, rownames = TRUE)
3302
3303 output$drc_plot <- renderPlot({
3304   fit <- fitmodel246247()
3305   data <- filedatal246247()
3306   ec_values <- ECx246247()
3307
3308 TestResult246247 <- eventReactive(input$buttonRunStep246247, {
3309   data <- filedatal246247()
3310   if(intest_type246247() == 'Essai246247') {
3311     data$dose <- as.factor(data$dose)
3312     fit <- lm(response ~ dose, data = data)
3313     Res <- summary(glht(fit, linfct = mcp(dose = "Dunnett"),
3314     alternative = "less"))
3315     return(Res)
3316   }
3317 }
3318
3319 output$test_result <- renderPrint({TestResult246247()})
3320
3321 if(intest_type246247() == 'Essai246247') {
3322   plot(fit, log = "x", broken = TRUE, xlab = paste0("Dose", " ",
3323     input$conc_unit246247, ")), ylab = "Réponse")
3324
3325   ec10 <- ec_values[ec_values$EC == 10,]
3326   ec50 <- ec_values[ec_values$EC == 50,]
3327
3328   if (nrow(ec10) > 0) {
3329     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3330     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3331     abline(v = ec10$Estimate, col = "red", lty = 2)
3332   }
3333   if (nrow(ec50) > 0) {
3334     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3335     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3336     abline(v = ec50$Estimate, col = "green", lty = 2)
3337   }
3338
3339   test_results <- TestResult246247()
3340   if (!is.null(test_results)) {
3341     noec_values <- test_results$test$p.value > 0.05
3342     loec_values <- test_results$test$p.value <= 0.05
3343     if (any(noec_values, na.rm = TRUE)) {
3344       NOEC <- min(test_results$test$p.value[noec_values], na.rm
3345       = TRUE)
3346       abline(v = NOEC, col = "purple", lty = 2)
3347     }
3348     if (any(loec_values, na.rm = TRUE)) {
3349       LOEC <- max(test_results$test$p.value[loec_values], na.rm
3350       = TRUE)
3351       abline(v = LOEC, col = "orange", lty = 2)
3352   }
3353 }
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4289
4290
4291
4292
4293
4294
4295
4296
4297
4297
4298
4299
4299
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4399
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4489
4490
4491
4492
4493
4494
4495
4496
4497
4497
4498
4499
4499
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4539
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4549
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4589
4589
4590
4591
4592
4593
4594
4595
4596
4597
4597
4598
4599
4599
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4629
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4639
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4649
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4659
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4669
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4679
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4689
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4699
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4729
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4739
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4749
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4759
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4769
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4779
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4789
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4799
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4819
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4829
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4839
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4849
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4859
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4869
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4879
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4889
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4899
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4919
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4929
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4939
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4949
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4959
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4969
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4979
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4989
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
4999
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5009
5010
5011
5012
5013
5014
5015
5016
5017
5018
5019
5019
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5029
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5039
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5049
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5059
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5069
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5079
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5089
5089
5090
5091
5092
5093
5094
5095
5096
5097
5098
5099
5099
5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119
5119
5119
5120
5121
5122
5123
5124
5125
5126
5127
5128
5129
5129
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5139
5139
5140
5141
5142
5143
5144
5145
5146
5147
5148
5149
5149
5149
5150
5151
5152
5153
5154
5155
5156
5157
5158
5159
5159
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5169
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5179
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5189
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5199
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208
5209
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5219
5219
5220
5221
5222
5223
5224
5
```



```

3396     Estimate = ec_values[,1],
3397     'Standard Error' = ec_values[,2],
3398     'Lower 95% CI' = ec_values[,3],
3399     'Upper 95% CI' = ec_values[,4]
3400   )
3401
3402   drc_df$Pente <- coefficients(fit)[[1]]
3403
3404   return(drc_df)
3405 }
3406 )
3407
3408 output$drc_result246 <- renderTable({
3409   ECx246()
3410 }, rownames = TRUE)
3411
3412 output$drc_plot <- renderPlot({
3413   fit <- fitmodel246()
3414   data <- filedatal246()
3415   ec_values <- ECx246()
3416
3417   TestResult246 <- eventReactive(input$buttonRunStep246, {
3418     data <- filedatal246()
3419     if(intest_type246() == 'Essai246') {
3420       data$concentration <- as.factor(data$concentration)
3421       fit <- lm(response ~ concentration, data = data)
3422       Res <- summary(glht(fit, linfct = mcp(concentration = "Dunnett"), alternative = "less"))
3423       return(Res)
3424     }
3425   })
3426
3427   output$test_result <- renderPrint({TestResult246()})
3428
3429   if(intest_type246() == 'Essai246') {
3430     plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_unit246, ")), ylab = "Réponse")
3431
3432     ec10 <- ec_values[ec_values$EC == 10,]
3433     ec50 <- ec_values[ec_values$EC == 50,]
3434
3435     if (nrow(ec10) > 0) {
3436       segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3437       segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3438       abline(v = ec10$Estimate, col = "red", lty = 2)
3439     }
3440     if (nrow(ec50) > 0) {
3441       segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3442       segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3443       abline(v = ec50$Estimate, col = "green", lty = 2)
3444     }

```

```

3445
3446     test_results <- TestResult246()
3447     if (!is.null(test_results)) {
3448         noec_values <- test_results$test$p.value > 0.05
3449         loec_values <- test_results$test$p.value <= 0.05
3450         if (any(noec_values, na.rm = TRUE)) {
3451             NOEC <- min(test_results$test$p.value[noec_values], na.rm
3452                         = TRUE)
3453             abline(v = NOEC, col = "purple", lty = 2)
3454         }
3455         if (any(loec_values, na.rm = TRUE)) {
3456             LOEC <- max(test_results$test$p.value[loec_values], na.rm
3457                         = TRUE)
3458             abline(v = LOEC, col = "orange", lty = 2)
3459         }
3460     }
3461
3462
3463
3464 # 247
3465
3466     intest_type247 <- reactive({ 'Essai247' })
3467
3468     validateFile <- function(filename) {
3469         extFile <- tools::file_ext(filename)
3470         validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
3471                         sont acceptés."))
3472     }
3473
3474     filedatal247 <- eventReactive(input$buttonRunStep247, {
3475         if(intest_type247() == 'Essai247') {
3476             req(input$datafile_ESSAI247)
3477             validateFile(input$datafile_ESSAI247)
3478             ff <- input$datafile_ESSAI247
3479             read.csv(file = ff$datapath, header = TRUE)
3480         }
3481     })
3482
3483     output$rawdata247 <- DT::renderDataTable({
3484         filedatal247()
3485     })
3486
3487     fitmodel247 <- eventReactive(input$buttonRunStep247, {
3488         data <- filedatal247()
3489         if(intest_type247() == 'Essai247') {
3490             if(input$model_ESSAI247 == 'probit') {
3491                 fit <- drm(response ~ as.numeric(concentration), data = data
3492                         , fct = PROBIT())
3493             } else if(input$model_ESSAI247 == 'logit') {
3494                 fit <- drm(response ~ as.numeric(concentration), data = data
3495                         , fct = LOGIT())
3496             }
3497         }
3498     })

```

```

            , fct = LOGIT())
3493     }
3494     return(fit)
3495   }
3496 }
3497
ECx247 <- eventReactive(input$buttonRunStep247, {
3498   fit <- fitmodel247()
3499   if(intest_type247() == 'Essai247') {
3500     XX <- as.numeric(input$ecx_ESSAI247)
3501     ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
3502       FALSE)
3503
3504     drc_df <- data.frame(
3505       EC = c(XX, 50),
3506       Estimate = ec_values[,1],
3507       'Standard Error' = ec_values[,2],
3508       'Lower 95% CI' = ec_values[,3],
3509       'Upper 95% CI' = ec_values[,4]
3510     )
3511
3512     drc_df$Pente <- coefficients(fit)[[1]]
3513
3514     return(drc_df)
3515   }
3516 })
3517
3518 output$drc_result247 <- renderTable({
3519   ECx247()
3520 }, rownames = TRUE)
3521
3522 output$drc_plot <- renderPlot({
3523   fit <- fitmodel247()
3524   data <- filedatal247()
3525   ec_values <- ECx247()
3526
3527   TestResult247 <- eventReactive(input$buttonRunStep247, {
3528     data <- filedatal247()
3529     if(intest_type247() == 'Essai247') {
3530       data$concentration <- as.factor(data$concentration)
3531       fit <- lm(response ~ concentration, data = data)
3532       Res <- summary(glht(fit, linfct = mcp(concentration =
3533         "Dunnett"), alternative = "less"))
3534       return(Res)
3535     }
3536   })
3537
3538   output$test_result <- renderPrint({TestResult247()})
3539
3540   if(intest_type247() == 'Essai247') {
3541     plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration",
3542       input$conc_unit247, ")), ylab = "Réponse"

```

```

        "))

3541   ec10 <- ec_values[ec_values$EC == 10,]
3542   ec50 <- ec_values[ec_values$EC == 50,]

3544

3545   if (nrow(ec10) > 0) {
3546     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3547     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3548     abline(v = ec10$Estimate, col = "red", lty = 2)
3549   }
3550   if (nrow(ec50) > 0) {
3551     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3552     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3553     abline(v = ec50$Estimate, col = "green", lty = 2)
3554   }

3555

3556   test_results <- TestResult247()
3557   if (!is.null(test_results)) {
3558     noec_values <- test_results$test$p.value > 0.05
3559     loec_values <- test_results$test$p.value <= 0.05
3560     if (any(noec_values, na.rm = TRUE)) {
3561       NOEC <- min(test_results$test$p.value[noec_values], na.rm
3562                     = TRUE)
3563       abline(v = NOEC, col = "purple", lty = 2)
3564     }
3565     if (any(loec_values, na.rm = TRUE)) {
3566       LOEC <- max(test_results$test$p.value[loec_values], na.rm
3567                     = TRUE)
3568       abline(v = LOEC, col = "orange", lty = 2)
3569     }
3570   }
3571 }

3572 #ESCORT

3573

3574   intest_typeESCORT <- reactive({ 'ESCORT' })

3575

3576   validateFile <- function(filename) {
3577     extFile <- tools::file_ext(filename)
3578     validate(need(extFile == c("csv", "txt"), "Seuls les fichiers CSV
3579       sont acceptés."))
3580   }

3581   filedataESCORT <- eventReactive(input$buttonRunStepESCORT, {
3582     if(intest_typeESCORT() == 'ESCORT') {
3583       req(input$datafile_ESCORT)
3584       validateFile(input$datafile_ESCORT)
3585       ff <- input$datafile_ESCORT
3586       read.csv(file = ff$datapath, header = TRUE)
3587     }
3588   })

```

```

3589
3590   output$rawdataESCORT <- DT::renderDataTable({
3591     filedataESCORT()
3592   })
3593
3594   fitmodelESCORT <- eventReactive(input$buttonRunStepESCORT, {
3595     data <- filedataESCORT()
3596     if(intest_typeESCORT() == 'ESCORT') {
3597       if(input$model_ESCORT == 'll2') {
3598         fit <- drm(response ~ as.numeric(concentration), data = data
3599                   , fct = LL.2())
3600     } else if(input$model_ESCORT == 'll3') {
3601       fit <- drm(response ~ as.numeric(concentration), data = data
3602                   , fct = LL.3())
3603     } else if(input$model_ESCORT == 'll4') {
3604       fit <- drm(response ~ as.numeric(concentration), data = data
3605                   , fct = LL.4())
3606     } else if(input$model_ESCORT == 'lm') {
3607       fit <- lm(response ~ as.numeric(concentration), data = data)
3608     } else if(input$model_ESCORT == 'nls') {
3609       fit <- nls(response ~ fct(concentration), data = data)
3610     } else if(input$model_ESCORT == 'lme') {
3611       fit <- lme(response ~ as.numeric(concentration), random = ~
3612                   1 | group, data = data)
3613     }
3614     return(fit)
3615   }
3616 }
3617
3618 ECxESCORT <- eventReactive(input$buttonRunStepESCORT, {
3619   fit <- fitmodelESCORT()
3620   if(intest_typeESCORT() == 'ESCORT') {
3621     XX <- as.numeric(input$ecx_ESCORT)
3622     ec_values <- ED(fit, c(XX, 50), interval = "delta", display =
3623                   FALSE)
3624
3625     drc_df <- data.frame(
3626       EC = c(XX, 50),
3627       Estimate = ec_values[,1],
3628       'Standard Error' = ec_values[,2],
3629       'Lower 95% CI' = ec_values[,3],
3630       'Upper 95% CI' = ec_values[,4]
3631     )
3632
3633     drc_df$Pente <- coefficients(fit)[[1]]
3634
3635     return(drc_df)
3636   }
3637 }
3638
3639 output$drc_resultESCORT <- renderTable({
3640   ECxESCORT()

```

```

3636 }, rownames = TRUE)
3637
3638 output$drc_plot <- renderPlot({
3639   fit <- fitmodeLESCORT()
3640   data <- filedataESCORt()
3641   ec_values <- ECxESCORt()
3642
3643 TestResultESCORt <- eventReactive(input$buttonRunStepESCORt, {
3644   data <- filedataESCORt()
3645   if(intest_typeESCORt() == 'ESCORt') {
3646     data$concentration <- as.factor(data$concentration)
3647     fit <- lm(response ~ concentration, data = data)
3648     Res <- anova(fit)
3649     return(Res)
3650   }
3651 })
3652
3653 output$test_result <- renderPrint({TestResultESCORt()})
3654
3655 if(intest_typeESCORt() == 'ESCORt') {
3656   plot(fit, log = "x", broken = TRUE, xlab = paste0("concentration", input$conc_uniteESCORt, ")), ylab = "Réponse")
3657
3658   ec10 <- ec_values[ec_values$EC == 10,]
3659   ec50 <- ec_values[ec_values$EC == 50,]
3660
3661   if (nrow(ec10) > 0) {
3662     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3663     segments(x0 = ec10$Estimate, y0 = 0, x1 = ec10$Estimate)
3664     abline(v = ec10$Estimate, col = "red", lty = 2)
3665   }
3666   if (nrow(ec50) > 0) {
3667     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3668     segments(x0 = ec50$Estimate, y0 = 0, x1 = ec50$Estimate)
3669     abline(v = ec50$Estimate, col = "green", lty = 2)
3670   }
3671
3672 test_results <- TestResultESCORt()
3673 if (!is.null(test_results)) {
3674   noec_values <- test_results$test$p.value > 0.05
3675   loec_values <- test_results$test$p.value <= 0.05
3676   if (any(noec_values, na.rm = TRUE)) {
3677     NOEC <- min(test_results$test$p.value[noec_values], na.rm = TRUE)
3678     abline(v = NOEC, col = "purple", lty = 2)
3679   }
3680   if (any(loec_values, na.rm = TRUE)) {
3681     LOEC <- max(test_results$test$p.value[loec_values], na.rm = TRUE)
3682     abline(v = LOEC, col = "orange", lty = 2)
3683   }

```

```

3684     }
3685   }
3686 })
3687
3688 ## Rapport ##
3689 output$downloadReport <- downloadHandler(
3690   filename = function() {
3691     paste('my-report', sep = '.', switch(
3692       input$format, PDF = 'pdf', HTML = 'html', Word = 'docx',
3693     ))
3694   },
3695   content = function(file) {
3696     src <- normalizePath('report.Rmd')
3697     owd <- setwd(tempdir())
3698     on.exit(setwd(owd))
3699     file.copy(src, 'report.Rmd', overwrite = TRUE)
3700
3701     library(rmarkdown)
3702     out <- render('report.Rmd', switch(
3703       input$format,
3704       PDF = pdf_document(), HTML = html_document(), Word = word_
3705       document()
3706     ))
3707     file.rename(out, file)
3708   }
3709 )
3710
3711
3712
3713 output$contents <- renderDT({
3714   global_data()
3715 })
3716
3717 output$contents2 <- renderDT({
3718   global_data()
3719 })
3720
3721 output$contents3 <- renderDT({
3722   global_data()
3723 })
3724
3725 output$contents4 <- renderDT({
3726   global_data()
3727 })
3728 }
3729
3730 shinyApp(ui, server)

```

Analysis page:

```

1 library(dplyr)
2 library(ggplot2)

```

```

3  library(glue)
4  library(leaflet)
5  library(plotly)
6  library(sass)
7  library(shiny)
8  library(shiny.fluent)
9  library(shiny.router)
10
11 filters <- function() {
12   Stack(
13     tokens = list(childrenGap = 10),
14     fileInput("file_analysis", "", 
15               multiple = FALSE,
16               accept = c("text/csv", "text/comma-separated-values",
17                         "text/plain", ".txt/.csv")),
18     textAreaInput("textdata", "Or Paste Historical Data Here", "", 
19                   rows = 10),
20     tags$hr(),
21     checkboxInput("header_analysis", "Header", TRUE),
22     radioButtons("sep_analysis", "Separator",
23                  choices = c(Comma = ",",
24                             Semicolon = ";",
25                             Tab = "\t"),
26                  selected = ","),
27     actionButton("submit", "Submit")
28   )
29 }
30
31
32
33
34
35 solvant <- function(id) {
36   div(
37     checkboxInput(id, label = "Activate solvent effect", value = 
38                   FALSE),
39     textOutput(paste0(id, "Value"))
40   )
41 }
42
43
44
45 outlier <- function(id) {
46   div(
47     checkboxInput(id, label = "Handle outliers", value = FALSE),
48     textOutput(paste0(id, "Value"))
49   )
50 }
51
52
53
54
55 transformation <- function(id) {
56   div(
57     checkboxInput(id, label = "Apply transformation", value = FALSE),
58     uiOutput(paste0(id, "_menu")),
59     textOutput(paste0(id, "Value"))
60   )
61 }
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149

```

```

50
51 #      Boxplot      #####
52 ff <- sidebarLayout(
53   sidebarPanel(
54     solvant("effect_solvant"),
55     outlier("manage_outliers"),
56     transformation("apply_transformation"),
57     actionButton("go", "Execute")
58   ),
59   mainPanel(
60     verbatimTextOutput("result"),
61     plotOutput("plot")
62   )
63 )
64
65 analysisPage <- function() {
66   tagList(
67     makePage(
68       "Descriptive_Statistics_Analysis",
69       "Upload_and_analyze_your_data",
70       div(
71         HorizontalStack(
72           makeCard("Data_Upload", filters(), size = 4, style = "
73             max-height: 520px"),
74           makeCard("Visualization", plotlyOutput("plot_"
75             visualization"), size = 8, style = "max-height: 520px"
76             )
77         ),
78         Stack(
79           tokens = list(childrenGap = 10), horizontal = TRUE,
80           makeCard("Options", ff, size = 4, style = "max-height: 520px"),
81           makeCard("Boxplot1", plotlyOutput("plot_box"), size = 9,
82             style = "max-height: 520px"),
83           makeCard("Boxplot2", plotlyOutput("plot_box2"), size =
84             9, style = "max-height: 520px")
85         ),
86         makeCard("Data_Preview", DTOutput("uploaded_data"), style =
87           "margin-top: 100px;"),
88         makeCard("Data_Preview_after_transformation", tableOutput(
90           "dataTF"), style = "margin-top: 100px;"),
91         makeCard("Outliers", tableOutput("outliers_table"), style =
92           "margin-top: 100px;"),
93         makeCard("Descriptive_Statistics", DTOutput("desc_stats"),
94           style = "margin-top: 100px;")
95       )
96     )
97   )
98 }

```

Page ANOVA:

```

1 anovaPage <- function() {

```

```
2 tagList(
3     makePage(
4         "ANOVA\u2014Test",
5         "ANOVA\u2014test\u2014results",
6         div(
7             HorizontalStack(
8                 makeCard("File\u2014input",
9                     div(
10                         actionButton("run_anova", "Run\u2014ANOVA\u2014Test")
11                     ),
12                     size = 4, style = "max-height:320px"
13                 ),
14                 makeCard("Data\u2014Preview", DTOutput("contents4"), size = 8,
15                     style = "max-height:320px")
16             ),
17             div(
18                 makeCard("Results", verbatimTextOutput("anova_result"),
19                     style = "margin-top:20px; ")
20             )
21         )
22     )
23 }
```

Page Tests Bayesien

```

1 library(shiny)
2 library(plotrix)
3 library(ggplot2)
4 library(coda)
5 library(shinyjs)
6
7 bayesianPage <- function() {
8   navbarPage(
9     "",
10
11   # Onglet d'introduction / théorie
12   tabPanel("",

13     fluidPage(
14       h2(),
15       p(),
16       h3("Théorie"),
17       h4("Tests Bayésiens"),
18       p("Les tests bayésiens permettent d'incorporer des
19         informations a priori dans l'analyse des données.
20         Le facteur de Bayes (B) compare la plausibilité
21         des données sous deux hypothèses : l'hypothèse
22         nulle (\mathcal{H}_0) et l'hypothèse
23         alternative (\mathcal{H}_1)."),
24       p("Le facteur de Bayes est calculé comme suit :"),
25       p("$$B_{1/0} = \frac{P(\text{Données} | \mathcal{H}_1)}{P(\text{Données} | \mathcal{H}_0)}$$"),
26       p("Les probabilités a posteriori des hypothèses
27         sont alors :"),
28       p("$$P(\mathcal{H}_1 | \text{Données}) = \frac{B_{1/0} \cdot P(\mathcal{H}_1)}{1 + B_{1/0} \cdot P(\mathcal{H}_1)}$$"),
29       p("$$P(\mathcal{H}_0 | \text{Données}) = \frac{1}{1 + B_{1/0} \cdot P(\mathcal{H}_1)}$$"),
30     )
31   )
32 }

```

```

22           nulles et alternatives peuvent etre calculees en
23           utilisant le facteur de Bayes et une probabilite a
24           priori de l'hypothese nulle.")
25     )),
26
27 # Onglet d'analyse bayesienne standard
28 tabPanel("", 
29   pageWithSidebar(
30     headerPanel("Test effet solvant"),
31     sidebarPanel(
32       uiOutput("column_select"),
33       radioButtons(inputId = "design",
34                     label = "Design:",
35                     choiceNames = c("Single/paired sample(
36                                     s)", "Independent samples"),
37                     choiceValues = c("single", "independent"),
38                     selected = "single"),
39       radioButtons(inputId = "direction",
40                     label = "Predicted direction:",
41                     choiceNames = c("None", "Positive effect",
42                                     "Negative effect"),
43                     choiceValues = c("none", "positive", "negative"),
44                     selected = "none"),
45       numericInput(inputId = "t",
46                     label = "t-statistic:",
47                     value = 0,
48                     min = -1e5,
49                     max = 1e5,
50                     step=0.1),
51       numericInput(inputId = "pHO",
52                     label = "Prior probability of null:",
53                     value=0.5,
54                     min=0,
55                     max=1,
56                     step=0.1),
57       numericInput('nchain', 'Nombre d\'iterations',
58                     value=10000),
59       numericInput('burnin', 'Période de burn-in', value=500),
60       numericInput('sd_beta0', 'Ecart-type pour beta0',
61                     value=1),
62       numericInput('sd_beta1', 'Ecart-type pour beta1',
63                     value=2),
64       actionButton('run_EstiBayesian', 'Lancer l\'analyse
65     '),
66       br(),
67       br(),
68       p(""))
69     ),
70     mainPanel(

```



```

97         div(helpText('Interprétation des résultats :'),
98             style="font-size:140%"),
99         div(uiOutput("interpretation"), align = "left",
100            style="font-size:130%;")
101        )
102    )
}

```

Page selection de modèle (BAYESIEN)

```

1 BayoyoPage <- function() {
2   tagList(
3     makePage(
4       "Analyse de données bayésiennes",
5       "Téléchargez et analysez vos données avec des méthodes bayésiennes",
6       div(
7         HorizontalStack(
8           makeCard(
9             "Téléchargement des données",
10            fileInput("file", "Choisissez un fichier CSV ou TXT sans
11              nom de colonnes",
12            accept = c("text/csv", "text/comma-separated-
13              values", "text/plain", ".csv", ".txt")),
14            size = 4,
15            style = "max-height:520px"
16          ),
17          makeCard(
18            "Options d'analyse",
19            div(
20              numericInput("iterations", "Nombre d'itérations (Metropolis-Hastings)", value = 50000),
21              numericInput("burn_in", "Burn-in", value = 1000),
22              actionButton("run_analysis", "Lancer l'analyse")
23            ),
24            size = 4,
25            style = "max-height:520px"
26          )
27        ),
28        Stack(
29          tokens = list(childrenGap = 10), horizontal = TRUE,
30          makeCard("Log-Logistic Distribution",
31            plotOutput("plot_loglogistic_1"),
32            #plotOutput("plot_loglogistic_2"),
33            size = 9, style = "max-height:520px"),
34          makeCard("Log-Normal Distribution",
35            plotOutput("plot_lognormal_1"),
36            #plotOutput("plot_lognormal_2"),
37            size = 9, style = "max-height:520px"),
38          makeCard("Weibull Distribution",
39            plotOutput("plot_weibull_1"),
40            #plotOutput("plot_weibull_2"),
41          )
42        )
43      )
44    )
45  )
46}

```

```

39             size = 9, style = "max-height: 520px")
40         ),
41         makeCard("Facteurs_bayésiens", verbatimTextOutput("bayes_
42             factors"), style = "margin-top: 100px;"),
43         makeCard("Taux_d'acceptation", verbatimTextOutput("_
44             acceptance_rates"), style = "margin-top: 100px;")
45     )
46 }

```

Page BOOTSTRAP

```

1 bootstrapPage <- function() {
2   tagList(
3     makePage(
4       "Bootstrap_Test",
5       "Bootstrap_test_results",
6       div(
7         HorizontalStack(
8           makeCard("File_input",
9             div(
10              actionButton("run_bootstrap", "Run_Bootstrap_
11                  Test")
12            ),
13            size = 4, style = "max-height: 320px"
14          ),
15          makeCard("Data_Preview", DTOutput("contents2"), size = 8,
16            style = "max-height: 320px")
17        ),
18        div(
19          makeCard("Results", verbatimTextOutput("bootstrap_result")
20            , style = "margin-top: 20px;"),
21          makeCard("Bootstrap_Statistics", plotOutput("bootstrap_
22              plot"), style = "margin-top: 20px;")
23        )
24      )
25    )
26  )
27 }

```

Page DUNNETT:

```

1 dunnettPage <- function() {
2   tagList(
3     makePage(
4       "Dunnett_Test",
5       "Dunnett_test_results",
6       div(
7         HorizontalStack(
8           makeCard("File_input",
9             div(
10              actionButton("run_dunnett", "Run_Dunnett_Test")
11            )
12          )
13        )
14      )
15    )
16  )
17 }

```

```

11         ),
12         size = 4, style = "max-height:320px"
13     ),
14     makeCard("DataPreview", DTOutput("contents3"), size = 8,
15         style = "max-height:320px")
16 ),
17 div(
18     makeCard("Results", verbatimTextOutput("dunnett_result"),
19         style = "margin-top:20px"),
20     makeCard("DunnettPlot", plotOutput("dunnett_plot"), style
21         = "margin-top:20px")
22 )
23 )

```

```

1 footer <- function() {
2     Stack(
3         horizontal = TRUE,
4         horizontalAlign = "space-between",
5         tokens = list(childrenGap = 20),
6         Text(variant = "medium", "Version personnelle pour Université",
7             block = TRUE),
8         Text(variant = "medium", nowrap = FALSE, "For more information, "
9             contact_me_at_herve.boutrin@etu.univ-nantes.fr"),
10        Text(variant = "medium", nowrap = FALSE, "All rights reserved."))
11    )
12 }

```

Menu header:

```

1 library(shiny)
2 library(shiny.fluent)
3 library(shiny.router)
4
5 # Éléments du menu secondaire
6 farItems <- list(
7     list(
8         key = "tile",
9         text = "Grid view",
10        ariaLabel = "Grid view",
11        iconOnly = TRUE,
12        iconProps = list(iconName = "Tiles")
13    )
14 )
15
16
17 # Barre de commandes
18 command_bar <- CommandBar(
19     items = list(
20         list(
21             key = "home",

```

```

22     text = "0) □ Home",
23     iconProps = list(iconName = "Home"),
24     href = "#!/"
25   ),
26   list(
27     key = "analysis",
28     text = "I) □ Descriptive □ Analysis",
29     iconProps = list(iconName = "ExploreData"),
30     href = "#!/analysis"
31   ),
32   list(
33     key = "modeling",
34     text = "III) □ Modelization",
35     iconProps = list(iconName = "StackedLineChart"),
36     subMenuProps = list(
37       items = list(
38         list(
39           text = "Continuous □ Data □ Models",
40           key = "continuousDataModels",
41           subMenuProps = list(
42             items = list(
43               list(text = "Probit □ Model", key = "probitModel",
44                 href = "#!/probitModel"), #ok
45               list(text = "Logit □ Model", key = "logitModel", href
46                 = "#!/logitModel"), #ok
47               list(text = "Weibull □ Model", key = "weibullModel",
48                 href = "#!/weibullModel",
49                 subMenuProps = list(
50                   items = list(
51                     list(text = "", key = "", href = "#!/"),
52                     list(text = "", key = "", href = "#!/"))
53                 )
54               )), #ok
55               list(text = "Gompertz □ Model", key = "gompertzModel",
56                 href = "#!/gompertzModel"), #ok
57               list(text = "Hill □ Model", key = "hillModel", href =
58                 "#!/hillModel"), #ok
59               list(text = "Logistic □ Model", key = "logisticModel",
60                 href = "#!/logisticModel"), #ok
61               list(text = "Log-Logistic □ Model", key = "
62                 logLogisticModel", href = "#!/logLogisticModel"),
63                 #ok
64               list(text = "Exponential □ Model", key = "
65                 exponentialModel", href = "#!/exponentialModel"),
66                 #ok
67               list(text = "Beta-Binomial □ Model", key = "
68                 betaBinomialModel", href = "#!/betaBinomialModel"
69                 ),
70               list(text = "Bruce-Versteeg □ Model", key = "Bruce-
71                 VersteegModel", href = "#!/Bruce-VersteegModel"),
72                 #ok
73               list(text = "quadratique □ Model", key = "
74                 quadratiqueModel", href = "#!/quadratiqueModel")
75             )
76           )
77         )
78       )
79     )
80   )
81 
```

```

                quadratiqueModel", href = "#!/quadratiqueModel"),
# ok
60      list(text = "Seuil\u2022Model", key = "SeuilModel", href
61        = "#!/SeuilModel",
62        subMenuProps = list(
63          items = list(
64            list(text = "", key = "", href = "#!/"),
65            list(text = "", key = "", href = "#!/"),
66            list(text = "", key = "", href = "#!/"),
67            list(text = "", key = "", href = "#!/")
68          )
69        ),# ok
70      list(text = "Horm\u00e9tiques\u2022Model", key = "Horm\u00e9
71        tiquesmodel", href = "#!/Horm\u00e9tiquesmodel",
72        subMenuProps = list(
73          items = list(
74            list(text = "", key = "", href = "#!/"),
75            list(text = "", key = "", href = "#!/")
76          )
77        ),# ok
78      list(text = "ANCOVA\u2022Model", key = "ANCOVAmodel",
79        href = "#!/ANCOVAmodel")
80    )
81  ),
82  list(
83    text = "Quantal\u2022Data\u2022Models",
84    key = "quantalDataModels",
85    subMenuProps = list(
86      items = list(
87        list(text = "Binary\u2022Logistic\u2022Regression", key = "
88          binaryLogisticRegression", href = "#!/"
89          binaryLogisticRegression),
90        list(text = "Probit\u2022Regression", key = "
91          probitRegression", href = "#!/probitRegression"),
92        list(text = "Weibull\u2022Model", key = "WeibullModel",
93          href = "#!/WeibullModel"),
94        list(text = "Gumbel\u2022Model", key = "GumbelModel",
95          href = "#!/GumbelModel")
96      )
97    )
98  ),
99  list(
100    text = "Count\u2022Data\u2022Models",
101    key = "countDataModels",
102    subMenuProps = list(
103      items = list(
104        list(text = "Poisson\u2022Regression", key = "
105          poissonRegression", href = "#!/poissonRegression"
106          ),
107        list(text = "Negative\u2022Binomial\u2022Regression", key = "
108          negativeBinomialRegression", href = "#!/"
109          )
110      )
111    )
112  )
113  )
114  )
115  )
116  )
117  )
118  )
119  )
120  )
121  )
122  )
123  )
124  )
125  )
126  )
127  )
128  )
129  )
130  )
131  )
132  )
133  )
134  )
135  )
136  )
137  )
138  )
139  )
140  )
141  )
142  )
143  )
144  )
145  )
146  )
147  )
148  )
149  )
150  )
151  )
152  )
153  )
154  )
155  )
156  )
157  )
158  )
159  )
160  )
161  )
162  )
163  )
164  )
165  )
166  )
167  )
168  )
169  )
170  )
171  )
172  )
173  )
174  )
175  )
176  )
177  )
178  )
179  )
180  )
181  )
182  )
183  )
184  )
185  )
186  )
187  )
188  )
189  )
190  )
191  )
192  )
193  )
194  )
195  )
196  )
197  )
198  )
199  )
200  )
201  )
202  )
203  )
204  )
205  )
206  )
207  )
208  )
209  )
210  )
211  )
212  )
213  )
214  )
215  )
216  )
217  )
218  )
219  )
220  )
221  )
222  )
223  )
224  )
225  )
226  )
227  )
228  )
229  )
230  )
231  )
232  )
233  )
234  )
235  )
236  )
237  )
238  )
239  )
240  )
241  )
242  )
243  )
244  )
245  )
246  )
247  )
248  )
249  )
250  )
251  )
252  )
253  )
254  )
255  )
256  )
257  )
258  )
259  )
260  )
261  )
262  )
263  )
264  )
265  )
266  )
267  )
268  )
269  )
270  )
271  )
272  )
273  )
274  )
275  )
276  )
277  )
278  )
279  )
280  )
281  )
282  )
283  )
284  )
285  )
286  )
287  )
288  )
289  )
290  )
291  )
292  )
293  )
294  )
295  )
296  )
297  )
298  )
299  )
300  )
301  )
302  )
303  )
304  )
305  )
306  )
307  )
308  )
309  )
310  )
311  )
312  )
313  )
314  )
315  )
316  )
317  )
318  )
319  )
320  )
321  )
322  )
323  )
324  )
325  )
326  )
327  )
328  )
329  )
330  )
331  )
332  )
333  )
334  )
335  )
336  )
337  )
338  )
339  )
340  )
341  )
342  )
343  )
344  )
345  )
346  )
347  )
348  )
349  )
350  )
351  )
352  )
353  )
354  )
355  )
356  )
357  )
358  )
359  )
360  )
361  )
362  )
363  )
364  )
365  )
366  )
367  )
368  )
369  )
370  )
371  )
372  )
373  )
374  )
375  )
376  )
377  )
378  )
379  )
380  )
381  )
382  )
383  )
384  )
385  )
386  )
387  )
388  )
389  )
390  )
391  )
392  )
393  )
394  )
395  )
396  )
397  )
398  )
399  )
400  )
401  )
402  )
403  )
404  )
405  )
406  )
407  )
408  )
409  )
410  )
411  )
412  )
413  )
414  )
415  )
416  )
417  )
418  )
419  )
420  )
421  )
422  )
423  )
424  )
425  )
426  )
427  )
428  )
429  )
430  )
431  )
432  )
433  )
434  )
435  )
436  )
437  )
438  )
439  )
440  )
441  )
442  )
443  )
444  )
445  )
446  )
447  )
448  )
449  )
450  )
451  )
452  )
453  )
454  )
455  )
456  )
457  )
458  )
459  )
460  )
461  )
462  )
463  )
464  )
465  )
466  )
467  )
468  )
469  )
470  )
471  )
472  )
473  )
474  )
475  )
476  )
477  )
478  )
479  )
480  )
481  )
482  )
483  )
484  )
485  )
486  )
487  )
488  )
489  )
490  )
491  )
492  )
493  )
494  )
495  )
496  )
497  )
498  )
499  )
500  )
501  )
502  )
503  )
504  )
505  )
506  )
507  )
508  )
509  )
510  )
511  )
512  )
513  )
514  )
515  )
516  )
517  )
518  )
519  )
520  )
521  )
522  )
523  )
524  )
525  )
526  )
527  )
528  )
529  )
530  )
531  )
532  )
533  )
534  )
535  )
536  )
537  )
538  )
539  )
540  )
541  )
542  )
543  )
544  )
545  )
546  )
547  )
548  )
549  )
550  )
551  )
552  )
553  )
554  )
555  )
556  )
557  )
558  )
559  )
560  )
561  )
562  )
563  )
564  )
565  )
566  )
567  )
568  )
569  )
570  )
571  )
572  )
573  )
574  )
575  )
576  )
577  )
578  )
579  )
580  )
581  )
582  )
583  )
584  )
585  )
586  )
587  )
588  )
589  )
589  )
590  )
591  )
592  )
593  )
594  )
595  )
596  )
597  )
598  )
599  )
600  )
601  )
602  )
603  )
604  )
605  )
606  )
607  )
608  )
609  )
610  )
611  )
612  )
613  )
614  )
615  )
616  )
617  )
618  )
619  )
620  )
621  )
622  )
623  )
624  )
625  )
626  )
627  )
628  )
629  )
630  )
631  )
632  )
633  )
634  )
635  )
636  )
637  )
638  )
639  )
640  )
641  )
642  )
643  )
644  )
645  )
646  )
647  )
648  )
649  )
650  )
651  )
652  )
653  )
654  )
655  )
656  )
657  )
658  )
659  )
660  )
661  )
662  )
663  )
664  )
665  )
666  )
667  )
668  )
669  )
669  )
670  )
671  )
672  )
673  )
674  )
675  )
676  )
677  )
678  )
679  )
680  )
681  )
682  )
683  )
684  )
685  )
686  )
687  )
688  )
689  )
689  )
690  )
691  )
692  )
693  )
694  )
695  )
696  )
697  )
698  )
699  )
699  )
700  )
701  )
702  )
703  )
704  )
705  )
706  )
707  )
708  )
709  )
709  )
710  )
711  )
712  )
713  )
714  )
715  )
716  )
717  )
718  )
719  )
719  )
720  )
721  )
722  )
723  )
724  )
725  )
726  )
727  )
728  )
729  )
729  )
730  )
731  )
732  )
733  )
734  )
735  )
736  )
737  )
738  )
739  )
739  )
740  )
741  )
742  )
743  )
744  )
745  )
746  )
747  )
748  )
749  )
749  )
750  )
751  )
752  )
753  )
754  )
755  )
756  )
757  )
758  )
759  )
759  )
760  )
761  )
762  )
763  )
764  )
765  )
766  )
767  )
768  )
769  )
769  )
770  )
771  )
772  )
773  )
774  )
775  )
776  )
777  )
778  )
779  )
779  )
780  )
781  )
782  )
783  )
784  )
785  )
786  )
787  )
788  )
789  )
789  )
790  )
791  )
792  )
793  )
794  )
795  )
796  )
797  )
798  )
799  )
799  )
800  )
801  )
802  )
803  )
804  )
805  )
806  )
807  )
808  )
809  )
809  )
810  )
811  )
812  )
813  )
814  )
815  )
816  )
817  )
818  )
819  )
819  )
820  )
821  )
822  )
823  )
824  )
825  )
826  )
827  )
828  )
829  )
829  )
830  )
831  )
832  )
833  )
834  )
835  )
836  )
837  )
838  )
839  )
839  )
840  )
841  )
842  )
843  )
844  )
845  )
846  )
847  )
848  )
849  )
849  )
850  )
851  )
852  )
853  )
854  )
855  )
856  )
857  )
858  )
859  )
859  )
860  )
861  )
862  )
863  )
864  )
865  )
866  )
867  )
868  )
869  )
869  )
870  )
871  )
872  )
873  )
874  )
875  )
876  )
877  )
878  )
879  )
879  )
880  )
881  )
882  )
883  )
884  )
885  )
886  )
887  )
888  )
889  )
889  )
890  )
891  )
892  )
893  )
894  )
895  )
896  )
897  )
898  )
899  )
899  )
900  )
901  )
902  )
903  )
904  )
905  )
906  )
907  )
908  )
909  )
909  )
910  )
911  )
912  )
913  )
914  )
915  )
916  )
917  )
918  )
919  )
919  )
920  )
921  )
922  )
923  )
924  )
925  )
926  )
927  )
928  )
929  )
929  )
930  )
931  )
932  )
933  )
934  )
935  )
936  )
937  )
938  )
939  )
939  )
940  )
941  )
942  )
943  )
944  )
945  )
946  )
947  )
948  )
949  )
949  )
950  )
951  )
952  )
953  )
954  )
955  )
956  )
957  )
958  )
959  )
959  )
960  )
961  )
962  )
963  )
964  )
965  )
966  )
967  )
968  )
969  )
969  )
970  )
971  )
972  )
973  )
974  )
975  )
976  )
977  )
978  )
979  )
979  )
980  )
981  )
982  )
983  )
984  )
985  )
986  )
987  )
988  )
989  )
989  )
990  )
991  )
992  )
993  )
994  )
995  )
996  )
997  )
998  )
999  )
999  )
1000  )
1001  )
1002  )
1003  )
1004  )
1005  )
1006  )
1007  )
1008  )
1009  )
1009  )
1010  )
1011  )
1012  )
1013  )
1014  )
1015  )
1016  )
1017  )
1018  )
1019  )
1019  )
1020  )
1021  )
1022  )
1023  )
1024  )
1025  )
1026  )
1027  )
1028  )
1029  )
1029  )
1030  )
1031  )
1032  )
1033  )
1034  )
1035  )
1036  )
1037  )
1038  )
1039  )
1039  )
1040  )
1041  )
1042  )
1043  )
1044  )
1045  )
1046  )
1047  )
1048  )
1049  )
1049  )
1050  )
1051  )
1052  )
1053  )
1054  )
1055  )
1056  )
1057  )
1058  )
1059  )
1059  )
1060  )
1061  )
1062  )
1063  )
1064  )
1065  )
1066  )
1067  )
1068  )
1069  )
1069  )
1070  )
1071  )
1072  )
1073  )
1074  )
1075  )
1076  )
1077  )
1078  )
1078  )
1079  )
1080  )
1081  )
1082  )
1083  )
1084  )
1085  )
1086  )
1087  )
1088  )
1088  )
1089  )
1090  )
1091  )
1092  )
1093  )
1094  )
1095  )
1096  )
1097  )
1098  )
1098  )
1099  )
1100  )
1101  )
1102  )
1103  )
1104  )
1105  )
1106  )
1107  )
1108  )
1109  )
1109  )
1110  )
1111  )
1112  )
1113  )
1114  )
1115  )
1116  )
1117  )
1118  )
1119  )
1119  )
1120  )
1121  )
1122  )
1123  )
1124  )
1125  )
1126  )
1127  )
1128  )
1129  )
1129  )
1130  )
1131  )
1132  )
1133  )
1134  )
1135  )
1136  )
1137  )
1138  )
1139  )
1139  )
1140  )
1141  )
1142  )
1143  )
1144  )
1145  )
1146  )
1147  )
1148  )
1149  )
1149  )
1150  )
1151  )
1152  )
1153  )
1154  )
1155  )
1156  )
1157  )
1158  )
1159  )
1159  )
1160  )
1161  )
1162  )
1163  )
1164  )
1165  )
1166  )
1167  )
1168  )
1169  )
1169  )
1170  )
1171  )
1172  )
1173  )
1174  )
1175  )
1176  )
1177  )
1178  )
1178  )
1179  )
1180  )
1181  )
1182  )
1183  )
1184  )
1185  )
1186  )
1187  )
1188  )
1188  )
1189  )
1190  )
1191  )
1192  )
1193  )
1194  )
1195  )
1196  )
1197  )
1198  )
1198  )
1199  )
1200  )
1201  )
1202  )
1203  )
1204  )
1205  )
1206  )
1207  )
1208  )
1209  )
1209  )
1210  )
1211  )
1212  )
1213  )
1214  )
1215  )
1216  )
1217  )
1218  )
1219  )
1219  )
1220  )
1221  )
1222  )
1223  )
1224  )
1225  )
1226  )
1227  )
1228  )
1229  )
1229  )
1230  )
1231  )
1232  )
1233  )
1234  )
1235  )
1236  )
1237  )
1238  )
1239  )
1239  )
1240  )
1241  )
1242  )
1243  )
1244  )
1245  )
1246  )
1247  )
1248  )
1249  )
1249  )
1250  )
1251  )
1252  )
1253  )
1254  )
1255  )
1256  )
1257  )
1258  )
1259  )
1259  )
1260  )
1261  )
1262  )
1263  )
1264  )
1265  )
1266  )
1267  )
1268  )
1269  )
1269  )
1270  )
1271  )
1272  )
1273  )
1274  )
1275  )
1276  )
1277  )
1278  )
1278  )
1279  )
1280  )
1281  )
1282  )
1283  )
1284  )
1285  )
1286  )
1287  )
1288  )
1288  )
1289  )
1290  )
1291  )
1292  )
1293  )
1294  )
1295  )
1296  )
1297  )
1298  )
1298  )
1299  )
1300  )
1301  )
1302  )
1303  )
1304  )
1305  )
1306  )
1307  )
1308  )
1309  )
1309  )
1310  )
1311  )
1312  )
1313  )
1314  )
1315  )
1316  )
1317  )
1318  )
1319  )
1319  )
1320  )
1321  )
1322  )
1323  )
1324  )
1325  )
1326  )
1327  )
1328  )
1329  )
1329  )
1330  )
1331  )
1332  )
1333  )
1334  )
1335  )
1336  )
1337  )
1338  )
1339  )
1339  )
1340  )
1341  )
1342  )
1343  )
1344  )
1345  )
1346  )
1347  )
1348  )
1349  )
1349  )
1350  )
1351  )
1352  )
1353  )
1354  )
1355  )
1356  )
1357  )
1358  )
1359  )
1359  )
1360  )
1361  )
1362  )
1363  )
1364  )
1365  )
1366  )
1367  )
1368  )
1369  )
1369  )
1370  )
1371  )
1372  )
1373  )
1374  )
1375  )
1376  )
1377  )
1378  )
1378  )
1379  )
1380  )
1381  )
1382  )
1383  )
1384  )
1385  )
1386  )
1387  )
1388  )
1388  )
1389  )
1390  )
1391  )
1392  )
1393  )
1394  )
1395  )
1396  )
1397  )
1398  )
1398  )
1399  )
1400  )
1401  )
1402  )
1403  )
1404  )
1405  )
1406  )
1407  )
1408  )
1409  )
1409  )
1410  )
1411  )
1412  )
1413  )
1414  )
1415  )
1416  )
1417  )
1418  )
1419  )
1419  )
1420  )
1421  )
1422  )
1423  )
1424  )
1425  )
1426  )
1427  )
1428  )
1429  )
1429  )
1430  )
1431  )
1432  )
1433  )
1434  )
1435  )
1436  )
1437  )
1438  )
1439  )
1439  )
1440  )
1441  )
1442  )
1443  )
1444  )
1445  )
1446  )
1447  )
1448  )
1449  )
1449  )
1450  )
1451  )
1452  )
1453  )
1454  )
1455  )
1456  )
1457  )
1458  )
1459  )
1459  )
1460  )
1461  )
1462  )
1463  )
1464  )
1465  )
1466  )
1467  )
1468  )
1469  )
1469  )
1470  )
1471  )
1472  )
1473  )
1474  )
1475  )
1476  )
1477  )
1478  )
1478  )
1479  )
1480  )
1481  )
1482  )
1483  )
1484  )
1485  )
1486  )
1487  )
1488  )
1488  )
1489  )
1490  )
1491  )
1492  )
1493  )
1494  )
1495  )
1496  )
1497  )
1498  )
1498  )
1499  )
1500  )
1501  )
1502  )
1503  )
1504  )
1505  )
1506  )
1507  )
1508  )
1509  )
1509  )
1510  )
1511  )
1512  )
1513  )
1514  )
1515  )
1516  )
1517  )
1518  )
1519  )
1519  )
1520  )
1521  )
1522  )
1523  )
1524  )
1525  )
1526  )
1527  )
1528  )
1529  )
1529  )
1530  )
1531  )
1532  )
1533  )
1534  )
1535  )
1536  )
1537  )
1538  )
1539  )
1539  )
1540  )
1541  )
1542  )
1543  )
1544  )
1545  )
1546  )
1547  )
1548  )
1549  )
1549  )
1550  )
1551  )
1552  )
1553  )
1554  )
1555  )
1556  )
1557  )
1558  )
1559  )
1559  )
1560  )
1561  )
1562  )
1563  )
1564  )
1565  )
1566  )
1567  )
1568  )
1569  )
1569  )
1570  )
1571  )
1572  )
1573  )
1574  )
1575  )
1576  )
1577  )
1578  )
1578  )
1579  )
1580  )
1581  )
1582  )
1583  )
1584  )
1585  )
1586  )
1587  )
1588  )
1588  )
1589  )
1590  )
1591  )
1592  )
1593  )
1594  )
1595  )
1596  )
1597  )
1598  )
1598  )
1599  )
1600  )
1601  )
1602  )
1603  )
1604  )
1605  )
1606  )
1607  )
1608  )
1609  )
1609  )
1610  )
1611  )
1612  )
1613  )
1614  )
1615  )
1616  )
1617  )
1618  )
1619  )
1619  )
1620  )
1621  )
1622  )
1623  )
1624  )
1625  )
1626  )
1627  )
1628  )
1629  )
1629  )
1630  )
1631  )
1632  )
1633  )
1634  )
1635  )
1636  )
1637  )
1638  )
1639  )
1639  )
1640  )
1641  )
1642  )
1643  )
1644  )
1645  )
1646  )
1647  )
1648  )
1649  )
1649  )
1650  )
1651  )
1652  )
1653  )
1654  )
1655  )
1656  )
1657  )
1658  )
1659  )
1659  )
1660  )
1661  )
1662  )
1663  )
1664  )
1665  )
1666  )
1667  )
1668  )
1669  )
1669  )
1670  )
1671  )
1672  )
1673  )
1674  )
1675  )
1676  )
1677  )
1678  )
1678  )
1679  )
1680  )
1681  )
1682  )
1683  )
1684  )
1685  )
1686  )
1687  )
1688  )
1688  )
1689  )
1690  )
1691  )
1692  )
1693  )
1694  )
1695  )
1696  )
1697  )
1698  )
1698  )
1699  )
1700  )
1701  )
1702  )
1703  )
1704  )
1705  )
1706  )
1707  )
1708  )
1709  )
1709  )
1710  )
1711  )
1712  )
1713  )
1714  )
1715  )
1716  )
1717  )
1718  )
1719  )
1719  )
1720  )
1721  )
1722  )
1723  )
1724  )
1725  )
1726  )
1727  )
1728  )
1729  )
1729  )
1730  )
1731  )
1732  )
1733  )
1734  )
1735  )
1736  )
1737  )
1738  )
1739  )
1739  )
1740  )
1741  )
1742  )
1743  )
1744  )
1745  )
1746  )
1747  )
1748  )
1749  )
1749  )
1750  )
1751  )
1752  )
1753  )
1754  )
1755  )
1756  )
1757  )
1758  )
1759  )
1759  )
1760  )
1761  )
1762  )
1763  )
1764  )
1765  )
1766  )
1767  )
1768  )
1769  )
1769  )
1770  )
1771  )
1772  )
1773  )
1774  )
1775  )
1776  )
1777  )
1778  )
1778  )
1779  )
1780  )
1781  )
1782  )
1783  )
1784  )
1785  )
1786  )
1787  )
1788  )
1788  )
1789  )
1790  )
1791  )
1792  )
1793  )
1794  )
1795  )
1796  )
1797  )
1798  )
1798  )
1799  )
1800  )
1801  )
1802  )
1803  )
1804  )
1805  )
1806  )
1807  )
1808  )
1809  )
1809  )
1810  )
1811  )
1812  )
1813  )
1814  )
1815  )
1816  )
1817  )
1818  )
1819  )
1819  )
1820  )
1821  )
1822  )
1823  )
1824  )
1825  )
1826  )
1827  )
1828  )
1829  )
1829  )
1830  )
1831  )
1832  )
1833  )
1834  )
1835  )
1836  )
1837  )
1838  )
1839  )
1839  )
1840  )
1841  )
1842  )
1843  )
1844  )
1845  )
1846  )
1847  )
1848  )
1849  )
1849  )
1850  )
1851  )
1852  )
1853  )
1854  )
1855  )
1856  )
1857  )
1858  )
1859  )
1859  )
1860  )
1861  )
1862  )
1863  )
1864  )
1865  )
1866  )
1867  )
1868  )
1869  )
1869  )
1870  )
1871  )
1872  )
1873  )
1874  )
1875  )
1876  )
1877  )
1878  )
1878  )
1879  )
1880  )
1881  )
1882  )
1883  )
1884  )
1885  )
1886  )
1887  )
1888  )
1888  )
1889  )

```



```

176         kolmogorov", href = "#!/kolmogorov"),
177     list(text = "Anderson-Darling\u2022test", key = "
178         andersonDarling", href = "#!/andersonDarling"),
179     list(text = "Shapiro-Wilk\u2022test", key = "shapiroWilk"
180         , href = "#!/shapiroWilk"),
181     list(text = "Lilliefors\u2022test", key = "lilliefors",
182         href = "#!/lilliefors"),
183     list(text = "Cramer-von\u2022Mises\u2022criterion", key = "
184         cramerVonMises", href = "#!/cramerVonMises"),
185     list(text = "Mood's\u2022median\u2022test", key = "moodMedian"
186         , href = "#!/moodMedian"),
187     list(text = "Savage\u2022score\u2022test", key = "savage",
188         href = "#!/savage"),
189     list(text = "Kolmogorov-Smirnov\u2022goodness-of-fit\u2022test"
190         , key = "ksGoodness", href = "#!/ksGoodness"),
191     list(text = "Mann-Kendall\u2022trend\u2022test", key = "
192         mannKendall", href = "#!/mannKendall"),
193     list(text = "Sen's\u2022slope\u2022estimate", key = "senSlope"
194         , href = "#!/senSlope"),
195     list(text = "Pettitt's\u2022test", key = "pettitt", href
196         = "#!/pettitt"),
197     list(text = "Seasonal\u2022Kendall\u2022test", key = "
198         seasonalKendall", href = "#!/seasonalKendall"),
199     list(text = "Durbin-Watson\u2022test", key = "
200         durbinWatson", href = "#!/durbinWatson"),
201     list(text = "Van\u2022der\u2022Waerden\u2022test", key = "
202         vanDerWaerden", href = "#!/vanDerWaerden"),
203     list(text = "Brown-Forsythe\u2022test", key = "
204         brownForsythe", href = "#!/brownForsythe"),
205     list(text = "Bartlett's\u2022test", key = "bartlett",
206         href = "#!/bartlett"),
207     list(text = "Levene's\u2022test", key = "levene", href =
208         "#!/levene"),
209     list(text = "Durbin\u2022test", key = "durbin", href = "#"
210         !/durbin")
211         )
212     )
213 ),
214 list(
215     text = "Quantal\u2022Data\u2022Tests",
216     key = "quantalDataTests",
217     subMenuProps = list(
218         items = list(
219             list(text = "Chi-squared\u2022test\u2022for\u2022independence", key
220                 = "chiSquared", href = "#!/chiSquared"),
221             list(text = "Fisher's\u2022exact\u2022test", key = "fisher",
222                 href = "#!/fisher"),
223             list(text = "McNemar's\u2022test", key = "mcnemar", href
224                 = "#!/mcnemar"),
225             list(text = "Cochran's\u2022Q\u2022test", key = "cochran",
226                 href = "#!/cochran"),
227             list(text = "Jonckheere-Terpstra\u2022test", key = "
228                 "
229             )
230         )
231     )
232 )
233 )
234 )
235 )
236 )
237 )
238 )
239 )
240 )
241 )
242 )
243 )
244 )
245 )
246 )
247 )
248 )
249 )
250 )
251 )
252 )
253 )
254 )
255 )
256 )
257 )
258 )
259 )
260 )
261 )
262 )
263 )
264 )
265 )
266 )
267 )
268 )
269 )
270 )
271 )
272 )
273 )
274 )
275 )
276 )
277 )
278 )
279 )
280 )
281 )
282 )
283 )
284 )
285 )
286 )
287 )
288 )
289 )
290 )
291 )
292 )
293 )
294 )
295 )
296 )
297 )
298 )
299 )
300 )
301 )
302 )
303 )
304 )
305 )
306 )
307 )
308 )
309 )
310 )
311 )
312 )
313 )
314 )
315 )
316 )
317 )
318 )
319 )
320 )
321 )
322 )
323 )
324 )
325 )
326 )
327 )
328 )
329 )
330 )
331 )
332 )
333 )
334 )
335 )
336 )
337 )
338 )
339 )
340 )
341 )
342 )
343 )
344 )
345 )
346 )
347 )
348 )
349 )
350 )
351 )
352 )
353 )
354 )
355 )
356 )
357 )
358 )
359 )
360 )
361 )
362 )
363 )
364 )
365 )
366 )
367 )
368 )
369 )
370 )
371 )
372 )
373 )
374 )
375 )
376 )
377 )
378 )
379 )
380 )
381 )
382 )
383 )
384 )
385 )
386 )
387 )
388 )
389 )
390 )
391 )
392 )
393 )
394 )
395 )
396 )
397 )
398 )
399 )
400 )
401 )
402 )
403 )
404 )
405 )
406 )
407 )
408 )
409 )
410 )
411 )
412 )
413 )
414 )
415 )
416 )
417 )
418 )
419 )
420 )
421 )
422 )
423 )
424 )
425 )
426 )
427 )
428 )
429 )
430 )
431 )
432 )
433 )
434 )
435 )
436 )
437 )
438 )
439 )
440 )
441 )
442 )
443 )
444 )
445 )
446 )
447 )
448 )
449 )
450 )
451 )
452 )
453 )
454 )
455 )
456 )
457 )
458 )
459 )
460 )
461 )
462 )
463 )
464 )
465 )
466 )
467 )
468 )
469 )
470 )
471 )
472 )
473 )
474 )
475 )
476 )
477 )
478 )
479 )
480 )
481 )
482 )
483 )
484 )
485 )
486 )
487 )
488 )
489 )
490 )
491 )
492 )
493 )
494 )
495 )
496 )
497 )
498 )
499 )
500 )
501 )
502 )
503 )
504 )
505 )
506 )
507 )
508 )
509 )
510 )
511 )
512 )
513 )
514 )
515 )
516 )
517 )
518 )
519 )
520 )
521 )
522 )
523 )
524 )
525 )
526 )
527 )
528 )
529 )
530 )
531 )
532 )
533 )
534 )
535 )
536 )
537 )
538 )
539 )
540 )
541 )
542 )
543 )
544 )
545 )
546 )
547 )
548 )
549 )
550 )
551 )
552 )
553 )
554 )
555 )
556 )
557 )
558 )
559 )
560 )
561 )
562 )
563 )
564 )
565 )
566 )
567 )
568 )
569 )
570 )
571 )
572 )
573 )
574 )
575 )
576 )
577 )
578 )
579 )
580 )
581 )
582 )
583 )
584 )
585 )
586 )
587 )
588 )
589 )
589 )
590 )
591 )
592 )
593 )
594 )
595 )
596 )
597 )
598 )
599 )
599 )
600 )
601 )
602 )
603 )
604 )
605 )
606 )
607 )
608 )
609 )
609 )
610 )
611 )
612 )
613 )
614 )
615 )
616 )
617 )
618 )
619 )
619 )
620 )
621 )
622 )
623 )
624 )
625 )
626 )
627 )
628 )
629 )
629 )
630 )
631 )
632 )
633 )
634 )
635 )
636 )
637 )
638 )
639 )
639 )
640 )
641 )
642 )
643 )
644 )
645 )
646 )
647 )
648 )
649 )
649 )
650 )
651 )
652 )
653 )
654 )
655 )
656 )
657 )
658 )
659 )
659 )
660 )
661 )
662 )
663 )
664 )
665 )
666 )
667 )
668 )
669 )
669 )
670 )
671 )
672 )
673 )
674 )
675 )
676 )
677 )
678 )
679 )
679 )
680 )
681 )
682 )
683 )
684 )
685 )
686 )
687 )
688 )
689 )
689 )
690 )
691 )
692 )
693 )
694 )
695 )
696 )
697 )
698 )
699 )
699 )
700 )
701 )
702 )
703 )
704 )
705 )
706 )
707 )
708 )
709 )
709 )
710 )
711 )
712 )
713 )
714 )
715 )
716 )
717 )
718 )
719 )
719 )
720 )
721 )
722 )
723 )
724 )
725 )
726 )
727 )
728 )
729 )
729 )
730 )
731 )
732 )
733 )
734 )
735 )
736 )
737 )
738 )
739 )
739 )
740 )
741 )
742 )
743 )
744 )
745 )
746 )
747 )
748 )
749 )
749 )
750 )
751 )
752 )
753 )
754 )
755 )
756 )
757 )
758 )
759 )
759 )
760 )
761 )
762 )
763 )
764 )
765 )
766 )
767 )
768 )
769 )
769 )
770 )
771 )
772 )
773 )
774 )
775 )
776 )
777 )
778 )
779 )
779 )
780 )
781 )
782 )
783 )
784 )
785 )
786 )
787 )
788 )
789 )
789 )
790 )
791 )
792 )
793 )
794 )
795 )
796 )
797 )
798 )
799 )
799 )
800 )
801 )
802 )
803 )
804 )
805 )
806 )
807 )
808 )
809 )
809 )
810 )
811 )
812 )
813 )
814 )
815 )
816 )
817 )
818 )
819 )
819 )
820 )
821 )
822 )
823 )
824 )
825 )
826 )
827 )
828 )
829 )
829 )
830 )
831 )
832 )
833 )
834 )
835 )
836 )
837 )
838 )
839 )
839 )
840 )
841 )
842 )
843 )
844 )
845 )
846 )
847 )
848 )
849 )
849 )
850 )
851 )
852 )
853 )
854 )
855 )
856 )
857 )
858 )
859 )
859 )
860 )
861 )
862 )
863 )
864 )
865 )
866 )
867 )
868 )
869 )
869 )
870 )
871 )
872 )
873 )
874 )
875 )
876 )
877 )
878 )
879 )
879 )
880 )
881 )
882 )
883 )
884 )
885 )
886 )
887 )
888 )
889 )
889 )
890 )
891 )
892 )
893 )
894 )
895 )
896 )
897 )
898 )
899 )
899 )
900 )
901 )
902 )
903 )
904 )
905 )
906 )
907 )
908 )
909 )
909 )
910 )
911 )
912 )
913 )
914 )
915 )
916 )
917 )
918 )
919 )
919 )
920 )
921 )
922 )
923 )
924 )
925 )
926 )
927 )
928 )
929 )
929 )
930 )
931 )
932 )
933 )
934 )
935 )
936 )
937 )
938 )
939 )
939 )
940 )
941 )
942 )
943 )
944 )
945 )
946 )
947 )
948 )
949 )
949 )
950 )
951 )
952 )
953 )
954 )
955 )
956 )
957 )
958 )
959 )
959 )
960 )
961 )
962 )
963 )
964 )
965 )
966 )
967 )
968 )
969 )
969 )
970 )
971 )
972 )
973 )
974 )
975 )
976 )
977 )
978 )
979 )
979 )
980 )
981 )
982 )
983 )
984 )
985 )
986 )
987 )
988 )
989 )
989 )
990 )
991 )
992 )
993 )
994 )
995 )
996 )
997 )
998 )
999 )
999 )
1000 )
1001 )
1002 )
1003 )
1004 )
1005 )
1006 )
1007 )
1008 )
1009 )
1009 )
1010 )
1011 )
1012 )
1013 )
1014 )
1015 )
1016 )
1017 )
1018 )
1019 )
1019 )
1020 )
1021 )
1022 )
1023 )
1024 )
1025 )
1026 )
1027 )
1028 )
1029 )
1029 )
1030 )
1031 )
1032 )
1033 )
1034 )
1035 )
1036 )
1037 )
1038 )
1039 )
1039 )
1040 )
1041 )
1042 )
1043 )
1044 )
1045 )
1046 )
1047 )
1048 )
1049 )
1049 )
1050 )
1051 )
1052 )
1053 )
1054 )
1055 )
1056 )
1057 )
1058 )
1059 )
1059 )
1060 )
1061 )
1062 )
1063 )
1064 )
1065 )
1066 )
1067 )
1068 )
1069 )
1069 )
1070 )
1071 )
1072 )
1073 )
1074 )
1075 )
1076 )
1077 )
1078 )
1078 )
1079 )
1080 )
1081 )
1082 )
1083 )
1084 )
1085 )
1086 )
1087 )
1088 )
1088 )
1089 )
1090 )
1091 )
1092 )
1093 )
1094 )
1095 )
1096 )
1097 )
1098 )
1098 )
1099 )
1099 )
1100 )
1101 )
1102 )
1103 )
1104 )
1105 )
1106 )
1107 )
1108 )
1109 )
1109 )
1110 )
1111 )
1112 )
1113 )
1114 )
1115 )
1116 )
1117 )
1118 )
1119 )
1119 )
1120 )
1121 )
1122 )
1123 )
1124 )
1125 )
1126 )
1127 )
1128 )
1129 )
1129 )
1130 )
1131 )
1132 )
1133 )
1134 )
1135 )
1136 )
1137 )
1138 )
1139 )
1139 )
1140 )
1141 )
1142 )
1143 )
1144 )
1145 )
1146 )
1147 )
1148 )
1149 )
1149 )
1150 )
1151 )
1152 )
1153 )
1154 )
1155 )
1156 )
1157 )
1158 )
1159 )
1159 )
1160 )
1161 )
1162 )
1163 )
1164 )
1165 )
1166 )
1167 )
1168 )
1169 )
1169 )
1170 )
1171 )
1172 )
1173 )
1174 )
1175 )
1176 )
1177 )
1178 )
1178 )
1179 )
1180 )
1181 )
1182 )
1183 )
1184 )
1185 )
1186 )
1187 )
1188 )
1189 )
1189 )
1190 )
1191 )
1192 )
1193 )
1194 )
1195 )
1196 )
1197 )
1198 )
1199 )
1199 )
1200 )
1201 )
1202 )
1203 )
1204 )
1205 )
1206 )
1207 )
1208 )
1209 )
1209 )
1210 )
1211 )
1212 )
1213 )
1214 )
1215 )
1216 )
1217 )
1218 )
1219 )
1219 )
1220 )
1221 )
1222 )
1223 )
1224 )
1225 )
1226 )
1227 )
1228 )
1229 )
1229 )
1230 )
1231 )
1232 )
1233 )
1234 )
1235 )
1236 )
1237 )
1238 )
1239 )
1239 )
1240 )
1241 )
1242 )
1243 )
1244 )
1245 )
1246 )
1247 )
1248 )
1249 )
1249 )
1250 )
1251 )
1252 )
1253 )
1254 )
1255 )
1256 )
1257 )
1258 )
1259 )
1259 )
1260 )
1261 )
1262 )
1263 )
1264 )
1265 )
1266 )
1267 )
1268 )
1269 )
1269 )
1270 )
1271 )
1272 )
1273 )
1274 )
1275 )
1276 )
1277 )
1278 )
1278 )
1279 )
1280 )
1281 )
1282 )
1283 )
1284 )
1285 )
1286 )
1287 )
1288 )
1288 )
1289 )
1290 )
1291 )
1292 )
1293 )
1294 )
1295 )
1296 )
1297 )
1298 )
1298 )
1299 )
1299 )
1300 )
1301 )
1302 )
1303 )
1304 )
1305 )
1306 )
1307 )
1308 )
1309 )
1309 )
1310 )
1311 )
1312 )
1313 )
1314 )
1315 )
1316 )
1317 )
1318 )
1319 )
1319 )
1320 )
1321 )
1322 )
1323 )
1324 )
1325 )
1326 )
1327 )
1328 )
1329 )
1329 )
1330 )
1331 )
1332 )
1333 )
1334 )
1335 )
1336 )
1337 )
1338 )
1339 )
1339 )
1340 )
1341 )
1342 )
1343 )
1344 )
1345 )
1346 )
1347 )
1348 )
1349 )
1349 )
1350 )
1351 )
1352 )
1353 )
1354 )
1355 )
1356 )
1357 )
1358 )
1359 )
1359 )
1360 )
1361 )
1362 )
1363 )
1364 )
1365 )
1366 )
1367 )
1368 )
1369 )
1369 )
1370 )
1371 )
1372 )
1373 )
1374 )
1375 )
1376 )
1377 )
1378 )
1378 )
1379 )
1380 )
1381 )
1382 )
1383 )
1384 )
1385 )
1386 )
1387 )
1388 )
1388 )
1389 )
1389 )
1390 )
1391 )
1392 )
1393 )
1394 )
1395 )
1396 )
1397 )
1398 )
1398 )
1399 )
1399 )
1400 )
1401 )
1402 )
1403 )
1404 )
1405 )
1406 )
1407 )
1408 )
1409 )
1409 )
1410 )
1411 )
1412 )
1413 )
1414 )
1415 )
1416 )
1417 )
1418 )
1419 )
1419 )
1420 )
1421 )
1422 )
1423 )
1424 )
1425 )
1426 )
1427 )
1428 )
1429 )
1429 )
1430 )
1431 )
1432 )
1433 )
1434 )
1435 )
1436 )
1437 )
1438 )
1439 )
1439 )
1440 )
1441 )
1442 )
1443 )
1444 )
1445 )
1446 )
1447 )
1448 )
1449 )
1449 )
1450 )
1451 )
1452 )
1453 )
1454 )
1455 )
1456 )
1457 )
1458 )
1459 )
1459 )
1460 )
1461 )
1462 )
1463 )
1464 )
1465 )
1466 )
1467 )
1468 )
1469 )
1469 )
1470 )
1471 )
1472 )
1473 )
1474 )
1475 )
1476 )
1477 )
1478 )
1478 )
1479 )
1479 )
1480 )
1481 )
1482 )
1483 )
1484 )
1485 )
1486 )
1487 )
1488 )
1488 )
1489 )
1489 )
1490 )
1491 )
1492 )
1493 )
1494 )
1495 )
1496 )
1497 )
1497 )
1498 )
1498 )
1499 )
1499 )
1500 )
1501 )
1502 )
1503 )
1504 )
1505 )
1506 )
1507 )
1508 )
1509 )
1509 )
1510 )
1511 )
1512 )
1513 )
1514 )
1515 )
1516 )
1517 )
1518 )
1519 )
1519 )
1520 )
1521 )
1522 )
1523 )
1524 )
1525 )
1526 )
1527 )
1528 )
1529 )
1529 )
1530 )
1531 )
1532 )
1533 )
1534 )
1535 )
1536 )
1537 )
1538 )
1539 )
1539 )
1540 )
1541 )
1542 )
1543 )
1544 )
1545 )
1546 )
1547 )
1548 )
1549 )
1549 )
1550 )
1551 )
1552 )
1553 )
1554 )
1555 )
1556 )
1557 )
1558 )
1559 )
1559 )
1560 )
1561 )
1562 )
1563 )
1564 )
1565 )
1566 )
1567 )
1568 )
1569 )
1569 )
1570 )
1571 )
1572 )
1573 )
1574 )
1575 )
1576 )
1577 )
1578 )
1578 )
1579 )
1579 )
1580 )
1581 )
1582 )
1583 )
1584 )
1585 )
1586 )
1587 )
1588 )
1588 )
1589 )
1589 )
1590 )
1591 )
1592 )
1593 )
1594 )
1595 )
1596 )
1597 )
1597 )
1598 )
1598 )
1599 )
1599 )
1600 )
1601 )
1602 )
1603 )
1604 )
1605 )
1606 )
1607 )
1608 )
1609 )
1609 )
1610 )
1611 )
1612 )
1613 )
1614 )
1615 )
1616 )
1617 )
1618 )
1619 )
1619 )
1620 )
1621 )
1622 )
1623 )
1624 )
1625 )
1626 )
1627 )
1628 )
1629 )
1629 )
1630 )
1631 )
1632 )
1633 )
1634 )
1635 )
1636 )
1637 )
1638 )
1639 )
1639 )
1640 )
1641 )
1642 )
1643 )
1644 )
1645 )
1646 )
1647 )
1648 )
1649 )
1649 )
1650 )
1651 )
1652 )
1653 )
1654 )
1655 )
1656 )
1657 )
1658 )
1659 )
1659 )
1660 )
1661 )
1662 )
1663 )
1664 )
1665 )
1666 )
1667 )
1668 )
1669 )
1669 )
1670 )
1671 )
1672 )
1673 )
1674 )
1675 )
1676 )
1677 )
1678 )
1678 )
1679 )
1679 )
1680 )
1681 )
1682 )
1683 )
1684 )
1685 )
1686 )
1687 )
1688 )
1688 )
1689 )
1689 )
1690 )
1691 )
1692 )
1693 )
1694 )
1695 )
1696 )
1697 )
1697 )
1698 )
1698 )
1699 )
1699 )
1700 )
1701 )
1702 )
1703 )
1704 )
1705 )
1706 )
1707 )
1708 )
1709 )
1709 )
1710 )
1711 )
1712 )
1713 )
1714 )
1715 )
1716 )
1717 )
1718 )
1719 )
1719 )
1720 )
1721 )
1722 )
1723 )
1724 )
1725 )
1726 )
1727 )
1728 )
1729 )
1729 )
1730 )
1731 )
1732 )
1733 )
1734 )
1735 )
1736 )
1737 )
1738 )
1739 )
1739 )
1740 )
1741 )
1742 )
1743 )
1744 )
1745 )
1746 )
1747 )
1748 )
1749 )
1749 )
1750 )
1751 )
1752 )
1753 )
1754 )
1755 )
1756 )
1757 )
1758 )
1759 )
1759 )
1760 )
1761 )
1762 )
1763 )
1764 )
1765 )
1766 )
1767 )
1768 )
1769 )
1769 )
1770 )
1771 )
1772 )
1773 )
1774 )
1775 )
1776 )
1777 )
1778 )
1778 )
1779 )
1779 )
1780 )
1781 )
1782 )
1783 )
1784 )
1785 )
1786 )
1787 )
1788 )
1788 )
1789 )
1789 )
1790 )
1791 )
1792 )
1793 )
1794 )
1795 )
1796 )
1797 )
1797 )
1798 )
1798 )
1799 )
1799 )
1800 )
1801 )
1802 )
1803 )
1804 )
1805 )
1806 )
1807 )
1808 )
1809 )
1809 )
1810 )
1811 )
1812 )
1813 )
1814 )
1815 )
1816 )
1817 )
1818 )
1819 )
1819 )
1820 )
1821 )
1822 )
1823 )
1824 )
1825 )
1826 )
1827 )
1828 )
1829 )
1829 )
1830 )
1831 )
1832 )
1833 )
1834 )
1835 )
1836 )
1837 )
1838 )
1839 )
1839 )
1840 )
1841 )
1842 )
1843 )
1844 )
1845 )
1846 )
1847 )
1848 )
1849 )
1849 )
1850 )
1851 )
1852 )
1853 )
1854 )
1855 )
1856 )
1857 )
1858 )
1859 )
1859 )
1860 )
1861 )
1862 )
1863 )
1864 )
1865 )
1866 )
1867 )
1868 )
1869 )
1869 )
1870 )
1871 )
1872 )
1873 )
1874 )
1875 )
1876 )
1877 )
1878 )
1878 )
1879 )
1879 )
1880 )
1881 )
1882 )
1883 )
1884 )
1885 )
1886 )
1887 )
1888 )
1888 )
1889 )
1889 )
1890 )
1891 )
1892 )
1893 )
1894 )
1895 )
1896 )
1897 )
1897 )
1898 )
1898 )
1899 )
1899 )
1900 )
1901 )
1902 )
1903 )
1904 )
1905 )
1906 )
1907 )
1908 )
1909 )
1909 )
1910 )
1911 )
1912 )
1913 )
1914 )
1915 )
1916 )
1917 )
1918 )
1919 )
1919 )
1920 )
1921 )
1922 )
1923 )
1924 )
1925 )
1926 )
1927 )
1928 )
1929 )
1929 )
1930 )
1931 )
1932 )
1933 )
1934 )
1935 )
1936 )
1937 )
1938 )
1939 )
1939 )
1940 )
1941 )
1942 )
1943 )
1944 )
1945 )
1946 )
1947 )
1948 )
1949 )
1949 )
1950 )
1951 )
1952 )
1953 )
1954 )
1955 )
1956 )
1957 )
1958 )
1959 )
1959 )
1960 )
1961 )
1962 )
1963 )
1964 )
1965 )
1966 )
1967 )
1968 )
1969 )
1969 )
1970 )
1971 )
1972 )
1973 )
1974 )
1975 )
1976 )
1977 )
1978 )
1978 )
1979 )
1979 )
1980 )
1981 )
1982 )
1983 )
1984 )
1985 )
1986 )
1987 )
1988 )
1988 )
1989 )
1989 )
1990 )
1991 )
1992 )
1993 )
1994 )
1995 )
1996 )
1997 )
1998 )
1999 )
1999 )
2000 )
2001 )
2002 )
2003 )
2004 )
2005 )
2006 )
2007 )
2008 )
2009 )
2009 )
2010 )
2011 )
2012 )
2013 )
2014 )
2015 )
2016 )
2017 )
2018 )
2019 )
2019 )
2020 )
2021 )
2022 )
2023 )
2024 )
2025 )
2026 )
2027 )
2028 )
2029 )
2029 )
2030 )
2031 )
2032 )
2033 )
2034 )
2035 )
2036 )
2037 )
2038 )
2039 )
2039 )
2040 )
2041 )
2042 )
2043 )
2044 )
2045 )
2046 )
2047 )
2048 )
2049 )
2049 )
2050 )
2051 )
2052 )
2053 )
2054 )
2055 )
2056 )
2057 )
2058 )
2059 )
2059 )
2060 )
2061 )
2062 )
2063 )
2064 )
2065 )
2066 )
2067 )
2068 )
2069 )
2069 )
2070 )
2071 )
2072 )
2073 )
2074 )
2075 )
2076 )
2077 )
2078 )
2078 )
2079 )
2079 )
2080 )
2081 )
2082 )
2083 )
2084 )
2085 )
2086 )
2087 )
2088 )
2088 )
2089 )
2089 )
2090 )
2091 )
2092 )
2093 )
2094 )
2095 )
2096 )
2097 )
2097 )
2098 )
2098 )
2099 )
2099 )
2100 )
2101 )
2102 )
2103 )
2104 )
2105 )
2106 )
2107 )
2108 )
2109 )
2109 )
2110 )
2111 )
2112 )
2113 )
2114 )
2115 )
2116 )
2117 )
2118 )
2119 )
2119 )
2120 )
2121 )
2122 )
2123 )
2124 )
2125 )
2126 )
2127 )
2128 )
2129 )
2129 )
2130 )
2131 )
2132 )
2133 )
2134 )
2135 )
2136 )
2137 )
2138 )
2139 )
2139 )
2140 )
2141 )
2142 )
2143 )
2144 )
2145 )
2146 )
2147 )
2148 )
2149 )
2149 )
2150 )
2151 )
2152 )
2153 )
2154 )
2155 )
2156 )
2157 )
2158 )
2159 )
2159 )
2160 )
2161 )
2162 )
2163 )
2164 )
2165 )
2166 )
2167 )
2168 )
2169 )
2169 )
2170 )
2171 )
2172 )
2173 )
2174 )
2175 )
2176 )
2177 )
2178 )
2178 )
2179 )
2179 )
2180 )
2181 )
2182 )
2183 )
2184 )
2185 )
2186 )
2187 )
2188 )
2188 )
2189 )
2189 )
2190 )
2191 )
2192 )
2193 )
2194 )
2195 )
2196 )
2197 )
2198 )
2199 )
2199 )
2200 )
2201 )
2202 )
2203 )
2204 )
2205 )
2206 )
2207 )
2208 )
2209 )
2209 )
2210 )
2211 )
2212 )
2213 )
2214 )
2215 )
2216 )
2217 )
2218 )
2219 )
2219 )
2220 )

```

```

206         jonckheere", href = "#!/jonckheere"),
207         list(text = "Sign\u2022test", key = "sign", href = "#!/sign"),
208         list(text = "Wald-Wolfowitz\u2022runs\u2022test", key = "waldWolfowitz",
209               href = "#!/waldWolfowitz"),
210         list(text = "Median\u2022test", key = "median", href = "#!/median"),
211         list(text = "Page's\u2022L\u2022trend\u2022test", key = "pageL",
212               href = "#!/pageL")
213     )
214   )
215   list(
216     text = "Count\u2022Data\u2022Tests",
217     key = "countDataTests",
218     subMenuProps = list(
219       items = list(
220         list(text = "Poisson\u2022Regression", key = "poissonRegression",
221               href = "#!/poissonRegression"),
222         list(text = "Negative\u2022Binomial\u2022Regression", key = "negativeBinomialRegression",
223               href = "#!/negativeBinomialRegression"),
224         list(text = "Zero\u2022Inflated\u2022Poisson\u2022(ZIP)\u2022Model", key = "zipModel",
225               href = "#!/zipModel"),
226         list(text = "Zero\u2022Inflated\u2022Negative\u2022Binomial\u2022(ZINB)\u2022Model", key = "zinbModel",
227               href = "#!/zinbModel")
228       ,
229       list(text = "Hurdle\u2022Models", key = "hurdleModels",
230             href = "#!/hurdleModels"),
231       list(text = "Quasi\u2022Poisson\u2022Regression", key = "quasiPoissonRegression",
232             href = "#!/quasiPoissonRegression"),
233       list(text = "Autoregressive\u2022Poisson\u2022Models", key = "autoregressivePoissonModels",
234             href = "#!/autoregressivePoissonModels"),
235       list(text = "Conway\u2022Maxwell\u2022Poisson\u2022Model", key = "conwayMaxwellPoissonModel",
236             href = "#!/conwayMaxwellPoissonModel"),
237       list(text = "Integer\u2022Valued\u2022ARIMA\u2022Models", key = "integerValuedARIMAModels",
238             href = "#!/integerValuedARIMAModels"),
239       list(text = "Latent\u2022Class\u2022Poisson\u2022Model", key = "latentClassPoissonModel",
240             href = "#!/latentClassPoissonModel"),
241       list(text = "Multilevel\u2022Poisson\u2022Regression", key = "multilevelPoissonRegression",
242             href = "#!/multilevelPoissonRegression"),
243       list(text = "Dirichlet\u2022Multinomial\u2022Model", key = "dirichletMultinomialModel",
244             href = "#!/dirichletMultinomialModel"),
245       list(text = "Overdispersed\u2022Poisson\u2022Model", key = "overdispersedPoissonModel",
246             href = "#!/overdispersedPoissonModel")
247     )
248   )
249 
```

```

231             overdispersedPoissonModel", href = "#!/overdispersedPoissonModel"),
232             list(text = "Hierarchical Poisson Regression", key =
233                   "hierarchicalPoissonRegression", href = "#!/hierarchicalPoissonRegression")
234         )
235     )
236   )
237   )
238   )
239   )
240 ),
241 list(
242   key = "calcPuiss",
243   text = "Calcul de Puissance",
244   iconProps = list(iconName = ""),
245   href = "#!/calcPuiss"),
246 list(
247   key = "Data analysis : Resampling",
248   text = "V) Resampling",
249   iconProps = list(iconName = "DatabaseSync"),
250   subMenuProps = list(
251     items = list(list(text = "Bootstrap", key = "bootstrap",
252                     href = "#!/bootstrap"),
253                     list(text = "Permutation Test", key =
254                           "permutation", href = "#!/permutation")
255                 )
256   ),
257   list(
258     key = "Bayesian Test",
259     text = "VI) Bayesian Methods",
260     iconProps = list(iconName = "ExploreData"),
261     href = "#!/bayesian"
262   ),
263   list(
264     key = "emailMessage",
265     text = "Contact",
266     iconProps = list(iconName = "Website"),
267     href = "mailto:me@example.com"
268   ),
269   list(
270     key = "download",
271     text = "Download Report",
272     iconProps = list(iconName = "Download")#ou CommandBarItem(
273       Download Report,"Download",onClick = JS("() => alert('
274       Download Report clicked')))
275   ),
276   farItems = farItems,

```

```

275     style = list(width = "100%")
276   )
277
278 # En-tete
279 header <- function() {
280   tagList(
281     img(src = "Boubouim.png", class = "logo"),
282     div(Text(variant = "medium", "Statistical Test Selection Tool",
283              style = list(color = "purple", fontWeight = "bold")), class =
284              "title"),
285     command_bar
286   )
287 }

```

Home page:

```

1  homePage <- function() {
2    tagList(
3      makePage(
4        "Decision Support Tool for Statistical Test Selection in
5          Environmental Risk Assessment",
6        "Choose the Right Statistical Method for Your Study",
7        div(
8          makeCard(
9            "Welcome to the Statistical Test Selection Tool!",
10           Stack(
11             Text("This tool helps you choose the appropriate
12               statistical test for environmental/ecotoxicology
13               studies."),
14             Text("Use the menu on the left to explore statistical
15               methods .")
16           )
17         ),
18         makeCard(
19           "Bootstrap Method",
20           Stack(
21             Text("The bootstrap method is a resampling technique
22               used to estimate statistics on a population by
23               sampling a dataset with replacement. It helps in
24               assessing the variability and confidence intervals of
25               the estimated parameters.")
26         )
27       ),
28       makeCard(
29         "Permutation Tests",
30         Stack(
31           Text("Permutation tests, also known as randomization
32             tests, involve randomly rearranging the data points
33             and recalculating the test statistic for each
34             rearrangement. This method is useful for hypothesis
35             testing when the distribution of the test statistic
36             under the null hypothesis is unknown .")
37         )
38     )
39   )
40 }

```

```

25     ),
26     makeCard(
27       "Bayesian Methods",
28       Stack(
29         Text("Bayesian methods incorporate prior knowledge or
30           beliefs into the statistical analysis. They update
31           the probability of a hypothesis as more evidence or
32           data becomes available. Bayesian inference is
33           particularly useful in complex models and when prior
34           information is available."))
35     )

```

fonction de mise en forme:

```

1  makePage <- function(title, subtitle, contents) {
2    tagList(
3      div(
4        class = "page-title",
5        span(
6          title,
7          class = "ms-fontSize-32 ms-fontWeight-semibold",
8          style = "color: #323130"
9        ),
10       span(
11         subtitle,
12         class = "ms-fontSize-14 ms-fontWeight-regular",
13         style = "color: #605E5C; margin: 14px;"
14       )
15     ),
16     contents
17   )
18 }

19
20
21 makeCard <- function(title, content, size = 12, style = "") {
22   div(
23     class = glue("card ms-depth-8 ms-sm-{size} ms-xl-{size}"),
24     style = style,
25     Stack(
26       tokens = list(childrenGap = 5),
27       Text(variant = "large", title, block = TRUE),
28       content
29     )
30   )
31 }
32
33
34 HorizontalStack <- function(...) {

```

```

35   Stack(
36     horizontal = TRUE,
37     tokens = list(childrenGap = 10),
38     ...
39   )
40 }

```

Menu navigation:

```

1 library(shiny)
2 library(shiny.fluent)
3 library(shiny.router)
4
5 navigation <- function() {
6   Nav(
7     groups = list(
8       list(
9         links = list(
10          list(
11            name = "AquaticTests",
12            key = "aquaticTests",
13            icon = "HistoricalWeather",
14            links = list(
15              list(
16                name = "Essai201",
17                url = "#!/Essai201",
18                key = "Essai201",
19                icon = "AdminELogoInverse32"
20              ),
21              list(
22                name = "Essai218",
23                url = "#!/Essai218",
24                key = "Essai218",
25                icon = "AdminELogoInverse32"
26              ),
27              list(
28                name = "Essai202",
29                url = "#!/Essai202",
30                key = "Essai202",
31                icon = "AdminELogoInverse32"
32              ),
33              list(
34                name = "Essai203",
35                url = "#!/Essai203",
36                key = "Essai203",
37                icon = "AdminELogoInverse32"
38              ),
39              list(
40                name = "Essai210",
41                url = "#!/Essai210",
42                key = "Essai210",
43                icon = "AdminELogoInverse32"
44              ),

```

```

45     list(
46         name = "Essai_211",
47         url = "#!/Essai211",
48         key = "Essai211",
49         icon = "AdminELogoInverse32"
50     ),
51     list(
52         name = "Essai_221",
53         url = "#!/Essai221",
54         key = "Essai221",
55         icon = "AdminELogoInverse32"
56     ),
57     list(
58         name = "Essai_239",
59         url = "#!/Essai239",
60         key = "Essai239",
61         icon = "AdminELogoInverse32"
62     )
63 )
64 ),
65 list(
66     name = "Terrestrial_Tests",
67     key = "terrestrialTests",
68     icon = "Globe",
69     links = list(
70         list(
71             name = "Essai_205",
72             url = "#!/Essai205",
73             key = "Essai205",
74             icon = "AdminELogoInverse32"
75         ),
76         list(
77             name = "Essai_206",
78             url = "#!/Essai206",
79             key = "Essai206",
80             icon = "AdminELogoInverse32"
81         ),
82         list(
83             name = "Essai_208",
84             url = "#!/Essai208",
85             key = "Essai208",
86             icon = "Flower"
87         ),
88         list(
89             name = "Essai_216",
90             url = "#!/Essai216",
91             key = "Essai216",
92             icon = "AdminELogoInverse32"
93         ),
94         list(
95             name = "Essai_222",
96             url = "#!/Essai222",

```

```

97         key = "Essai222",
98         icon = "AdminELogoInverse32"
99     ),
100    list(
101        name = "Essai_223",
102        url = "#!/Essai223",
103        key = "Essai223",
104        icon = "AdminELogoInverse32"
105    ),
106    list(
107        name = "Essai_226",
108        url = "#!/Essai226",
109        key = "Essai226",
110        icon = "AdminELogoInverse32"
111    ),
112    list(
113        name = "Essai_227",
114        url = "#!/Essai227",
115        key = "Essai227",
116        icon = "Flower"
117    ),
118    list(
119        name = "Essai_232",
120        url = "#!/Essai232",
121        key = "Essai232",
122        icon = "AdminELogoInverse32"
123    )
124 )
125 ),
126 list(
127     name = "Bee_and_Pollinator_Tests",
128     key = "beePollinatorTests",
129     icon = "LadybugSolid",
130     links = list(
131         list(
132             name = "Essai_213",
133             url = "#!/Essai213214",
134             key = "Essai213",
135             icon = "AdminELogoInverse32"
136         ),
137         list(
138             name = "Essai_214",
139             url = "#!/Essai213214",
140             key = "Essai214",
141             icon = "AdminELogoInverse32"
142         ),
143         list(
144             name = "Essai_245",
145             url = "#!/Essai245",
146             key = "Essai245",
147             icon = "AdminELogoInverse32"
148         ),

```

```

149         list(
150             name = "Essai_246",
151             url = "#!/Essai246247",
152             key = "Essai246",
153             icon = "AdminELogoInverse32"
154         ),
155         list(
156             name = "Essai_247",
157             url = "#!/Essai246247",
158             key = "Essai247",
159             icon = "AdminELogoInverse32"
160         )
161     )
162 ),
163 list(
164     name = "Non-target_arthropods",
165     key = "NTarthrop",
166     icon = "BugSolid",
167     links = list(
168         list(
169             name = "ESCORT",
170             url = "#!/ESCORT",
171             key = "ESCORT",
172             icon = "Bug"
173         )
174     )
175 )
176 ),
177 list(
178     name = "Internet_links",
179     links = list(
180         list(
181             name = "",
182             url = "",
183             key = "",
184             icon = ""
185         ),
186         list(
187             name = "",
188             url = "",
189             key = "",
190             icon = ""
191         )
192     )
193 )
194 )
195 ),
196 initialSelectedKey = "home",
197 styles = list(
198     root = list(
199         height = "100%",
200         boxSizing = "border-box",

```

```

201         border = "1px solid #eee",
202         overflowY = "auto"
203     )
204 )
205 }
206 }
```

Menu permutation

```

1 permutationPage <- function() {
2   tagList(
3     makePage(
4       "PermutationTest",
5       "PermutationTest_results",
6       div(
7         HorizontalStack(
8           makeCard("Fileinput",
9             div(
10              actionButton("run_permutation", "Run"
11                  PermutationTest")
12            ),
13            size = 4, style = "max-height: 320px"
14          ),
15          makeCard("DataPreview", DTOutput("contents"), size = 8,
16            style = "max-height: 320px")
17        ),
18        div(
19          makeCard("Results", DTOutput("results"), style = "margin-
20            top: 20px;"),
21          makeCard("PermutationStatistics", DTOutput("permutation_
22            data"), style = "margin-top: 20px;"),
23          makeCard("PermutationPlot", plotOutput("permutation_plot"
24            ), style = "margin-top: 20px;"))
25        )
26      )
27    )
28  )
29 }
```

Menu puissance

```

1 PuissancePage <- function() {
2   navbarPage(
3     "Calcul de Puissance",
4
5     # Onglet d'introduction/théorie
6     tabPanel("Introduction",
7       fluidPage(
8         h2("Introduction au Calcul de la Puissance"
9             Statistique"),
10        p("Cette section fournit une introduction à la
11          puissance statistique, ses concepts et son
12          importance dans les analyses statistiques."),
13      )
14    )
15  )
16 }
```

```

10
11 h3("Théorie"),
12 p("La puissance statistique est la probabilité de détecter un effet, si cet effet existe réellement. Un test statistique avec une puissance élevée a plus de chances de rejeter une hypothèse nulle fausse."),
13 p("Les paramètres influençant la puissance incluent le niveau de signification ( $\alpha$ ), la taille de l'effet, et la taille de l'échantillon."),
14 p("Il est crucial de calculer la puissance avant de mener une étude pour s'assurer que l'étude est suffisamment sensible pour détecter l'effet d'intérêt.")
15
16
17 # Onglet de calcul de la puissance
18 tabPanel("Calcul de Puissance",
19   pageWithSidebar(
20     headerPanel("Calcul de la Puissance Statistique"),
21     sidebarPanel(
22       # Sélection de la méthode de calcul de puissance
23       selectInput("method", "Choisir la méthode de calcul de puissance:",
24                  choices = c("t-test", "ANOVA", "Régression Log-Logistique")),
25
26       # Inputs pour t-test
27       conditionalPanel(
28         condition = "input.method=='t-test'",
29         h4("Paramètres pour t-test"),
30         selectInput("groupColumn_t", "Choisir la colonne de groupe", choices = NULL),
31         selectInput("valueColumn_t", "Choisir la colonne des valeurs", choices = NULL),
32         sliderInput("alphaLevel_t", "Niveau de signification(alpha):", min = 0.01, max = 0.1, value = 0.05)
33     ),
34
35       # Inputs pour ANOVA
36       conditionalPanel(
37         condition = "input.method=='ANOVA'",
38         h4("Paramètres pour ANOVA"),
39         selectInput("groupColumn_a", "Choisir la colonne de groupe", choices = NULL),
40         selectInput("valueColumn_a", "Choisir la colonne des valeurs", choices = NULL),
41         sliderInput("alphaLevel_a", "Niveau de signification(alpha):", min = 0.01, max = 0.1, value = 0.05)
42     )
43   )
44 )
45
46
47 
```

```

42     ),
43
44     # Inputs pour Régression Log-Logistique
45     conditionalPanel(
46       condition = "input.method=='Régression Log-
47       Logistique'",
48       h4("Paramètres pour Régression Log-Logistique"),
49       selectInput("doseColumn_1", "Choisir la colonne
50       des doses", choices = NULL),
51       selectInput("responseColumn_1", "Choisir la
52       colonne des réponses", choices = NULL),
53       sliderInput("alphaLevel_1", "Niveau de
54       signification (alpha):", min = 0.01, max =
55       0.1, value = 0.05)
56     )
57   ),
58   mainPanel(
59     h4("Graphique de Puissance"),
60     plotOutput("powerPlot"),
61     h4("Résultat de la Puissance"),
62     verbatimTextOutput("powerResult"),
63     uiOutput("column_select2")
64   )
65 )
66 )
67 }

```

Page essai standardisé ESCORT

```

1 library(shiny)
2 library(DT)
3 library(nlme)
4 library(drc)
5 library(shinyCSSloaders)
6
7 ESCORTPage <- function() {
8   tagList(
9     fluidPage(
10       titlePanel("ESCORT : Arthropodes non cibles"),
11       fluidRow(
12         column(4,
13          textInput(inputId = "chemical", "Nom du produit
14             chimique testé", value = "Produit chimique"),
15           selectInput("conc_unit", "Unité de Concentration",
16             choices = c("g/L", "mg/L", "microg/L", "ng/L")),
17           fileInput('datafile_ESCORT', 'Sélectionner un fichier
18             ', accept = c('.csv')),
19           radioButtons('model_ESCORT', 'Sélectionner le modèle
20             de régression',
21             choices = c('log-logistic 2 paramètres'
22             = '112',
23             'log-logistic 3 paramètres'
24             = '113'))
25         )
26       )
27     )
28   )
29 }

```

```

19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
}

```

Page essai standardisé 201

```
1 library(shiny)
2 library(DT)
3 library(drc)
4 library(multcomp)
5 library(shinyCSSloaders)
6
7 Essai201Page <- function() {
8   tagList(
9     makePage(
10       "Freshwater\u00a0Alga\u00a0and\u00a0Cyanobacteria,\u00a0Growth\u00a0Inhibition\u00a0Test",
11       "(Essai\u00a0201)",
12       fluidPage(
13         column(4,
14          textInput(inputId = "chemical", "Nom\u00a0du\u00a0produit\u00a0
15             chimique\u00a0test\u00e9", value = "Produit\u00a0chimique"),
16           selectInput("conc_unit201", "Unit\u00e9\u00a0de\u00a0Concentration",
17             choices = c("g/L", "mg/L", "microg/L", "ng/L")),
18           fileInput('datafile_ESSAI201', ''),
19             accept = c('.csv')),
20           tags$br(),
21           selectInput('model_ESSAI201', 'S\u00e9lectionner\u00a0le\u00a0mod\u00e8le
22             \u00a0de\u00a0r\u00e9gression',
23             choices = c('log-logistic\u00a02\u00a0param\u00e8tres'
24               = '112',
25               'log-logistic\u00a03\u00a0param\u00e8tres'
26               = '113',
27               'log-logistic\u00a04\u00a0param\u00e8tres'
28               = '114'),
29             selected = '112'),
30           numericInput(inputId = "ecx_ESSAI201", "D\u00e9terminer\u00a0la
31             \u00a0concentration\u00a0efficace\u00a0X%", value = 50, min = 0,
32             max = 100),
33           selectInput('test_method_ESSAI201', 'S\u00e9lectionner\u00a0la\u00a0
34             m\u00e9thode\u00a0de\u00a0test\u00a0d\u2019hypoth\u00e8se',
35             choices = c('Test\u00a0de\u00a0Dunnett' = 'Dunnett',
36             'Dunnett'), selected = 'Dunnett'),
37           actionButton("buttonRunStep201", "Ex\u00e9cuter", icon = icon("check"),
38             width = "100px"),
39             ),
40             column(8,
41               mainPanel(
42                 width = 12,
43                 DT::dataTableOutput('rawdata201'),
44                 br(),
45                 tableOutput('drc_result201'),
46                 withSpinner(plotOutput('drc_plot201', width = "100%",
47                   height = "850px", type = 4, color = "#9e0093'),
48                   br(),
49                   withSpinner(verbatimTextOutput("test_result201")),
50                   br()
51                 )
52               )
53             )
54           )
55         )
56       )
57     )
58   )
59 }
```

```

39     ),
40     tabPanel(
41       "Télécharger le rapport",
42       fixedRow(
43         column(
44           8,
45           br(),
46           HTML("<font face=verdana size=5 color=#fff><b>Télé",
47                 "charger le rapport d'analyse</b></font>"),
48           br(), br(), br(),
49           radioButtons('format201', 'Document format', c('PDF',
50                 'HTML', 'Word'), inline = TRUE),
51           downloadButton("DownloadReport201", "Télécharger le",
52                           rapport", icon = icon("fas fa-download"))#, style =
53                           'background-color:#ffff; color:#000000; border-
54                           color:#ffff;')
55         )
56     )
}

```

Page essai standardisé 202

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(multcomp)
5 library(shinyCSSloaders)
6
7 Essai202Page <- function() {
8   tagList(
9     makePage(
10       "Daphnia sp. Acute Immobilisation Test",
11       "(Essai 202)",
12       fluidRow(
13         column(4,
14          textInput(inputId = "chemical", "Nom du produit",
15                     chimique testé", value = "Produit chimique"),
16           selectInput("conc_unit202", "Unité de Concentration",
17                     choices = c("g/L", "mg/L", "microg/L", "ng/L")),
18           fileInput('datafile_ESSAI202', '',
19                     accept = c('.csv')),
20           tags$br(),
21           selectInput('model_ESSAI202', 'Sélectionner le modèle',
22                     de régression',
23                     choices = c('log-logistic 2 paramètres' =
24                     '112',
25                     'log-logistic 3 paramètres' =
26                     '113',
27                     'log-logistic 4 paramètres' =
28                     '114'))
29     )
30   )
31 }

```

```

22                               = '114'),
23                               selected = '112'),
24               numericInput(inputId = "ecx_ESSAI202", "Déterminer la
25                               concentration efficace X%", value = 50, min = 0,
26                               max = 100),
27               selectInput('test_method_ESSAI202', 'Sélectionner la
28                               méthode de test d\'hypothèse',
29                               choices = c('Test de Dunnett', 'Dunnett
30                               '),
31                               selected = 'Dunnett'),
32               actionButton("buttonRunStep202", "Exécuter", icon = icon
33                               ("check"), width = "100px")
34               ),
35               column(8,
36               mainPanel(
37                               width = 12,
38                               DT::dataTableOutput('rawdata202'),
39                               br(),
40                               tableOutput('drc_result202'),
41                               withSpinner(plotOutput('drc_plot202', width =
42                               "100%", height = "850px"), type = 4, color = '#9e0093'),
43                               br()
44                               )
45               ),
46               tabPanel(
47                               "Télécharger le rapport",
48                               fixedRow(
49                                   column(
50                                       8,
51                                       br(),
52                                       HTML("<font face=verdana size=5 color=#9e0093><b>Télé
53                               charger le rapport d'analyse</b></font>"),
54                                       br(), br(), br(),
55                                       radioButtons("format", "Sélectionner le format du
56                               rapport", c('Word'), inline = TRUE),
57                                       downloadButton("DownloadReport", "Télécharger le
58                               rapport", icon = icon("fas fa-download"), style = ,
59                               background-color:#9e0093; color:#000000; border-
60                               color:#9d9d9d;")
61                               )
62               )
63           )
64       )
65   )
66 }

```

Page essai standardisé 203

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(multcomp)
5 library(shinyCSSloaders)
6

```

```

7 | Essai203Page <- function() {
8 |   tagList(
9 |     makePage(
10 |       "Fish, Acute Toxicity Test",
11 |       "(Essai 203)",
12 |       fluidRow(
13 |         column(4,
14 |          textInput(inputId = "chemical", "Nom du produit
15 |                         chimique testé", value = "Produit chimique"),
16 |           selectInput("conc_unit203", "Unité de Concentration",
17 |                         choices = c("g/L", "mg/L", "microg/L", "ng/L")),
18 |           fileInput('datafile_ESSAI203', ''),
19 |           accept = c('.csv')
20 |         ),
21 |         tags$br(),
22 |         selectInput('model_ESSAI203', 'Sélectionner le modèle
23 |                         de régression',
24 |                         choices = c('log-logistic 2 paramètres'
25 |                         = '112',
26 |                         'log-logistic 3 paramètres'
27 |                         = '113',
28 |                         'log-logistic 4 paramètres'
29 |                         = '114'),
30 |                         selected = '112'),
31 |         numericInput(inputId = "ecx_ESSAI203", "Déterminer la
32 |                         concentration efficace X%", value = 50, min = 0,
33 |                         max = 100),
34 |         selectInput('test_method_ESSAI203', 'Sélectionner la
35 |                         méthode de test d\'hypothèse',
36 |                         choices = c('Test de Dunnett' = 'Dunnett',
37 |                         'Dunnett' = 'Dunnett'),
38 |         ),
39 |         column(8,
40 |           mainPanel(
41 |             width = 12,
42 |             DT::dataTableOutput('rawdata203'),
43 |             br(),
44 |             tableOutput('drc_result203'),
45 |             withSpinner(plotOutput('drc_plot203', width =

```

```

46         HTML("<font face=verdana size=5 color=#9e0093><b>Télé
47             charger le rapport d'analyse</b></font>"),
48         br(), br(), br(),
49         radioButtons("format", "Sélectionner le format du
50             rapport", c('Word'), inline = TRUE),
51         downloadButton("DownloadReport", "Télécharger le
52             rapport", icon = icon("fas fa-download"), style = ,
53             background-color:#9e0093; color:#000000; border-
54             color:#9d9d9d;')
55     )
  )
)
)
)
}

```

Page essai standardisé 205

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(multcomp)
5 library(shinyCSSloaders)
6
7 Essai205Page <- function() {
8     tagList(
9         fluidPage(
10            titlePanel("Avian Dietary Toxicity Test"),
11            fluidRow(
12                column(4,
13                   textInput(inputId = "chemical", "Nom du produit
14                        chimique testé", value = "Produit chimique"),
15                    selectInput("conc_unit", "Unité de Dose", choices = c
16                        ("g/kg", "mg/kg", "microg/kg", "ng/kg")),
17                    fileInput('datafile_ESSAI205', 'Sélectionner un
18                        fichier', accept = c('.csv')),
19                    radioButtons('model_ESSAI205', 'Sélectionner le modè
20                        le de régression',
21                        choices = c('log-logistic 2 paramètres'
22                            = '112',
23                                'log-logistic 3 paramètres'
24                                = '113',
25                                'log-logistic 4 paramètres'
26                                = '114'),
27                        selected = '112'),
28                    numericInput(inputId = "ecx_ESSAI205", "Déterminer la
29                        dose efficace X%", value = 50, min = 0, max =
30                        100),
31                    radioButtons('test_method_ESSAI205', 'Sélectionner la
32                        méthode de test d\'hypothèse',
33                        choices=c('Test de Dunnett', 'Dunnett
34                            '),
35                        selected='Dunnett'),
36                    actionButton("buttonRunStep1", "Exécuter", icon=
37

```

```

25   icon("check"), width = "100px")
26   ),
27   column(8,
28   mainPanel(
29   DT::dataTableOutput('rawdata205'),
30   br(),
31   tableOutput('drc_result205'),
32   withSpinner(plotOutput('drc_plot205', width = "100%", height = "850px"), type = 4, color = '#9e0093'),
33   br(),
34   withSpinner(verbatimTextOutput("test_result205")),
35   br()
36   )
37   ),
38   tabPanel(
39   "Télécharger le rapport",
40   fixedRow(
41   column(
42   8,
43   br(),
44   HTML("<font face=verdana size=5 color=#9e0093><b>Télé
45   charger le rapport d'analyse</b></font>"),
46   br(), br(), br(),
47   radioButtons("format", "Sélectionner le format du
48   rapport", c('Word'), inline = TRUE),
49   downloadButton("DownloadReport", "Télécharger le
50   rapport", icon = icon("fas fa-download"), style = 'background-color:#9e0093; color:#000000; border-color:#9d9d9d; ')
51   )
52   )
53 })
}

```

```

14         selectInput("conc_unit", "Unité de Dose", choices = c
15             ("g/kg", "mg/kg", "microg/kg", "ng/kg")),
16         fileInput('datafile_ESSAI206', 'Sélectionner un
17             fichier', accept = c('.csv')),
18         radioButtons('model_ESSAI206', 'Sélectionner le modèle
19             de régression',
20             choices = c('log-logistique 2 paramètres'
21             = '112',
22             'log-logistique 3 paramètres'
23             = '113',
24             'log-logistique 4 paramètres'
25             = '114'),
26             selected = '112'),
27         numericInput(inputId = "ecx_ESSAI206", "Déterminer la
28             dose efficace X%", value = 50, min = 0, max =
29             100),
30         radioButtons('test_method_ESSAI206', 'Sélectionner la
31             méthode de test d\'hypothèse',
32             choices = c('Test de Dunnett' = 'Dunnett
33             '),
34             selected = 'Dunnett'),
35         actionButton("buttonRunStep1", "Exécuter", icon =
36             icon("check"), width = "100px")
37         ),
38         column(8,
39             mainPanel(
40                 width = 12,
41                 DT::dataTableOutput('rawdata206'),
42                 br(),
43                 tableOutput('drc_result206'),
44                 withSpinner(plotOutput('drc_plot206', width =
45                     "100%", height = "850px"), type = 4, color = '#9e0093'),
46                     br(),
47                     withSpinner(verbatimTextOutput("test_result206")),
48                     br()
49             )
50         ),
51         tabPanel(
52             "Télécharger le rapport",
53             fixedRow(
54                 column(
55                     8,
56                     br(),
57                     HTML("<font face=verdana size=5 color=#9e0093><b>Télé
58                         charger le rapport d'analyse</b></font>"),
59                     br(), br(), br(),
60                     radioButtons("format", "Sélectionner le format du
61                         rapport", c('Word'), inline = TRUE),
62                     downloadButton("DownloadReport", "Télécharger le
63                         rapport", icon = icon("fas fa-download"), style =
64                         'background-color:#9e0093; color:#000000; border-
65                         color:#9d9d9d;')
66             )
67         )
68     )
69 )
70 )
71 )
72 )
73 )
74 )
75 )
76 )
77 )
78 )
79 )
80 )
81 )
82 )
83 )
84 )
85 )
86 )
87 )
88 )
89 )
90 )
91 )
92 )
93 )
94 )
95 )
96 )
97 )
98 )
99 )
100 )
101 )
102 )
103 )
104 )
105 )
106 )
107 )
108 )
109 )
110 )
111 )
112 )
113 )
114 )
115 )
116 )
117 )
118 )
119 )
120 )
121 )
122 )
123 )
124 )
125 )
126 )
127 )
128 )
129 )
130 )
131 )
132 )
133 )
134 )
135 )
136 )
137 )
138 )
139 )
140 )
141 )
142 )
143 )
144 )
145 )
146 )
147 )
148 )
149 )
150 )
151 )
152 )
153 )
154 )
155 )
156 )
157 )
158 )
159 )
160 )
161 )
162 )
163 )
164 )
165 )
166 )
167 )
168 )
169 )
170 )
171 )
172 )
173 )
174 )
175 )
176 )
177 )
178 )
179 )
180 )
181 )
182 )
183 )
184 )
185 )
186 )
187 )
188 )
189 )
190 )
191 )
192 )
193 )
194 )
195 )
196 )
197 )
198 )
199 )
200 )
201 )
202 )
203 )
204 )
205 )
206 )
207 )
208 )
209 )
210 )
211 )
212 )
213 )
214 )
215 )
216 )
217 )
218 )
219 )
220 )
221 )
222 )
223 )
224 )
225 )
226 )
227 )
228 )
229 )
230 )
231 )
232 )
233 )
234 )
235 )
236 )
237 )
238 )
239 )
240 )
241 )
242 )
243 )
244 )
245 )
246 )
247 )
248 )
249 )
250 )
251 )
252 )
253 )
254 )
255 )
256 )
257 )
258 )
259 )
260 )
261 )
262 )
263 )
264 )
265 )
266 )
267 )
268 )
269 )
270 )
271 )
272 )
273 )
274 )
275 )
276 )
277 )
278 )
279 )
280 )
281 )
282 )
283 )
284 )
285 )
286 )
287 )
288 )
289 )
290 )
291 )
292 )
293 )
294 )
295 )
296 )
297 )
298 )
299 )
300 )
301 )
302 )
303 )
304 )
305 )
306 )
307 )
308 )
309 )
310 )
311 )
312 )
313 )
314 )
315 )
316 )
317 )
318 )
319 )
320 )
321 )
322 )
323 )
324 )
325 )
326 )
327 )
328 )
329 )
330 )
331 )
332 )
333 )
334 )
335 )
336 )
337 )
338 )
339 )
340 )
341 )
342 )
343 )
344 )
345 )
346 )
347 )
348 )
349 )
350 )
351 )
352 )
353 )
354 )
355 )
356 )
357 )
358 )
359 )
360 )
361 )
362 )
363 )
364 )
365 )
366 )
367 )
368 )
369 )
370 )
371 )
372 )
373 )
374 )
375 )
376 )
377 )
378 )
379 )
380 )
381 )
382 )
383 )
384 )
385 )
386 )
387 )
388 )
389 )
390 )
391 )
392 )
393 )
394 )
395 )
396 )
397 )
398 )
399 )
400 )
401 )
402 )
403 )
404 )
405 )
406 )
407 )
408 )
409 )
410 )
411 )
412 )
413 )
414 )
415 )
416 )
417 )
418 )
419 )
420 )
421 )
422 )
423 )
424 )
425 )
426 )
427 )
428 )
429 )
430 )
431 )
432 )
433 )
434 )
435 )
436 )
437 )
438 )
439 )
440 )
441 )
442 )
443 )
444 )
445 )
446 )
447 )
448 )
449 )
450 )
451 )
452 )
453 )
454 )
455 )
456 )
457 )
458 )
459 )
460 )
461 )
462 )
463 )
464 )
465 )
466 )
467 )
468 )
469 )
470 )
471 )
472 )
473 )
474 )
475 )
476 )
477 )
478 )
479 )
480 )
481 )
482 )
483 )
484 )
485 )
486 )
487 )
488 )
489 )
490 )
491 )
492 )
493 )
494 )
495 )
496 )
497 )
498 )
499 )
500 )
501 )
502 )
503 )
504 )
505 )
506 )
507 )
508 )
509 )
510 )
511 )
512 )
513 )
514 )
515 )
516 )
517 )
518 )
519 )
520 )
521 )
522 )
523 )
524 )
525 )
526 )
527 )
528 )
529 )
530 )
531 )
532 )
533 )
534 )
535 )
536 )
537 )
538 )
539 )
540 )
541 )
542 )
543 )
544 )
545 )
546 )
547 )
548 )
549 )
550 )
551 )
552 )
553 )
554 )
555 )
556 )
557 )
558 )
559 )
560 )
561 )
562 )
563 )
564 )
565 )
566 )
567 )
568 )
569 )
570 )
571 )
572 )
573 )
574 )
575 )
576 )
577 )
578 )
579 )
580 )
581 )
582 )
583 )
584 )
585 )
586 )
587 )
588 )
589 )
590 )
591 )
592 )
593 )
594 )
595 )
596 )
597 )
598 )
599 )
599 )
600 )
601 )
602 )
603 )
604 )
605 )
606 )
607 )
608 )
609 )
610 )
611 )
612 )
613 )
614 )
615 )
616 )
617 )
618 )
619 )
620 )
621 )
622 )
623 )
624 )
625 )
626 )
627 )
628 )
629 )
630 )
631 )
632 )
633 )
634 )
635 )
636 )
637 )
638 )
639 )
640 )
641 )
642 )
643 )
644 )
645 )
646 )
647 )
648 )
649 )
649 )
650 )
651 )
652 )
653 )
654 )
655 )
656 )
657 )
658 )
659 )
659 )
660 )
661 )
662 )
663 )
664 )
665 )
666 )
667 )
668 )
669 )
669 )
670 )
671 )
672 )
673 )
674 )
675 )
676 )
677 )
678 )
679 )
679 )
680 )
681 )
682 )
683 )
684 )
685 )
686 )
687 )
688 )
689 )
689 )
690 )
691 )
692 )
693 )
694 )
695 )
696 )
697 )
698 )
699 )
699 )
700 )
701 )
702 )
703 )
704 )
705 )
706 )
707 )
708 )
709 )
709 )
710 )
711 )
712 )
713 )
714 )
715 )
716 )
717 )
718 )
719 )
719 )
720 )
721 )
722 )
723 )
724 )
725 )
726 )
727 )
728 )
729 )
729 )
730 )
731 )
732 )
733 )
734 )
735 )
736 )
737 )
738 )
739 )
739 )
740 )
741 )
742 )
743 )
744 )
745 )
746 )
747 )
748 )
749 )
749 )
750 )
751 )
752 )
753 )
754 )
755 )
756 )
757 )
758 )
759 )
759 )
760 )
761 )
762 )
763 )
764 )
765 )
766 )
767 )
768 )
769 )
769 )
770 )
771 )
772 )
773 )
774 )
775 )
776 )
777 )
778 )
779 )
779 )
780 )
781 )
782 )
783 )
784 )
785 )
786 )
787 )
788 )
789 )
789 )
790 )
791 )
792 )
793 )
794 )
795 )
796 )
797 )
798 )
799 )
799 )
800 )
801 )
802 )
803 )
804 )
805 )
806 )
807 )
808 )
809 )
809 )
810 )
811 )
812 )
813 )
814 )
815 )
816 )
817 )
818 )
819 )
819 )
820 )
821 )
822 )
823 )
824 )
825 )
826 )
827 )
828 )
829 )
829 )
830 )
831 )
832 )
833 )
834 )
835 )
836 )
837 )
838 )
839 )
839 )
840 )
841 )
842 )
843 )
844 )
845 )
846 )
847 )
848 )
849 )
849 )
850 )
851 )
852 )
853 )
854 )
855 )
856 )
857 )
858 )
859 )
859 )
860 )
861 )
862 )
863 )
864 )
865 )
866 )
867 )
868 )
869 )
869 )
870 )
871 )
872 )
873 )
874 )
875 )
876 )
877 )
878 )
878 )
879 )
880 )
881 )
882 )
883 )
884 )
885 )
886 )
887 )
888 )
889 )
889 )
890 )
891 )
892 )
893 )
894 )
895 )
896 )
897 )
898 )
899 )
899 )
900 )
901 )
902 )
903 )
904 )
905 )
906 )
907 )
908 )
909 )
909 )
910 )
911 )
912 )
913 )
914 )
915 )
916 )
917 )
918 )
919 )
919 )
920 )
921 )
922 )
923 )
924 )
925 )
926 )
927 )
928 )
929 )
929 )
930 )
931 )
932 )
933 )
934 )
935 )
936 )
937 )
938 )
939 )
939 )
940 )
941 )
942 )
943 )
944 )
945 )
946 )
947 )
948 )
949 )
949 )
950 )
951 )
952 )
953 )
954 )
955 )
956 )
957 )
958 )
959 )
959 )
960 )
961 )
962 )
963 )
964 )
965 )
966 )
967 )
968 )
969 )
969 )
970 )
971 )
972 )
973 )
974 )
975 )
976 )
977 )
978 )
978 )
979 )
980 )
981 )
982 )
983 )
984 )
985 )
986 )
987 )
988 )
989 )
989 )
990 )
991 )
992 )
993 )
994 )
995 )
996 )
997 )
998 )
999 )
999 )
1000 )
1001 )
1002 )
1003 )
1004 )
1005 )
1006 )
1007 )
1008 )
1009 )
1009 )
1010 )
1011 )
1012 )
1013 )
1014 )
1015 )
1016 )
1017 )
1018 )
1019 )
1019 )
1020 )
1021 )
1022 )
1023 )
1024 )
1025 )
1026 )
1027 )
1028 )
1029 )
1029 )
1030 )
1031 )
1032 )
1033 )
1034 )
1035 )
1036 )
1037 )
1038 )
1039 )
1039 )
1040 )
1041 )
1042 )
1043 )
1044 )
1045 )
1046 )
1047 )
1048 )
1049 )
1049 )
1050 )
1051 )
1052 )
1053 )
1054 )
1055 )
1056 )
1057 )
1058 )
1059 )
1059 )
1060 )
1061 )
1062 )
1063 )
1064 )
1065 )
1066 )
1067 )
1068 )
1069 )
1069 )
1070 )
1071 )
1072 )
1073 )
1074 )
1075 )
1076 )
1077 )
1078 )
1078 )
1079 )
1080 )
1081 )
1082 )
1083 )
1084 )
1085 )
1086 )
1087 )
1088 )
1088 )
1089 )
1090 )
1091 )
1092 )
1093 )
1094 )
1095 )
1096 )
1097 )
1098 )
1098 )
1099 )
1099 )
1100 )
1101 )
1102 )
1103 )
1104 )
1105 )
1106 )
1107 )
1108 )
1109 )
1109 )
1110 )
1111 )
1112 )
1113 )
1114 )
1115 )
1116 )
1117 )
1118 )
1119 )
1119 )
1120 )
1121 )
1122 )
1123 )
1124 )
1125 )
1126 )
1127 )
1128 )
1129 )
1129 )
1130 )
1131 )
1132 )
1133 )
1134 )
1135 )
1136 )
1137 )
1138 )
1138 )
1139 )
1140 )
1141 )
1142 )
1143 )
1144 )
1145 )
1146 )
1147 )
1148 )
1148 )
1149 )
1150 )
1151 )
1152 )
1153 )
1154 )
1155 )
1156 )
1157 )
1158 )
1159 )
1159 )
1160 )
1161 )
1162 )
1163 )
1164 )
1165 )
1166 )
1167 )
1168 )
1168 )
1169 )
1170 )
1171 )
1172 )
1173 )
1174 )
1175 )
1175 )
1176 )
1177 )
1178 )
1179 )
1179 )
1180 )
1181 )
1182 )
1183 )
1184 )
1185 )
1186 )
1187 )
1188 )
1188 )
1189 )
1190 )
1191 )
1192 )
1193 )
1194 )
1195 )
1196 )
1197 )
1198 )
1198 )
1199 )
1199 )
1200 )
1201 )
1202 )
1203 )
1204 )
1205 )
1206 )
1207 )
1208 )
1209 )
1209 )
1210 )
1211 )
1212 )
1213 )
1214 )
1215 )
1216 )
1217 )
1218 )
1218 )
1219 )
1220 )
1221 )
1222 )
1223 )
1224 )
1225 )
1226 )
1227 )
1228 )
1228 )
1229 )
1230 )
1231 )
1232 )
1233 )
1234 )
1235 )
1236 )
1237 )
1238 )
1238 )
1239 )
1240 )
1241 )
1242 )
1243 )
1244 )
1245 )
1246 )
1247 )
1248 )
1248 )
1249 )
1250 )
1251 )
1252 )
1253 )
1254 )
1255 )
1256 )
1257 )
1258 )
1259 )
1259 )
1260 )
1261 )
1262 )
1263 )
1264 )
1265 )
1266 )
1267 )
1268 )
1268 )
1269 )
1270 )
1271 )
1272 )
1273 )
1274 )
1275 )
1276 )
1277 )
1278 )
1278 )
1279 )
1280 )
1281 )
1282 )
1283 )
1284 )
1285 )
1286 )
1287 )
1288 )
1288 )
1289 )
1290 )
1291 )
1292 )
1293 )
1294 )
1295 )
1296 )
1297 )
1298 )
1298 )
1299 )
1299 )
1300 )
1301 )
1302 )
1303 )
1304 )
1305 )
1306 )
1307 )
1308 )
1309 )
1309 )
1310 )
1311 )
1312 )
1313 )
1314 )
1315 )
1316 )
1317 )
1318 )
1318 )
1319 )
1320 )
1321 )
1322 )
1323 )
1324 )
1325 )
1326 )
1327 )
1328 )
1328 )
1329 )
1330 )
1331 )
1332 )
1333 )
1334 )
1335 )
1336 )
1337 )
1338 )
1338 )
1339 )
1340 )
1341 )
1342 )
1343 )
1344 )
1345 )
1346 )
1347 )
1348 )
1348 )
1349 )
1350 )
1351 )
1352 )
1353 )
1354 )
1355 )
1356 )
1357 )
1358 )
1359 )
1359 )
1360 )
1361 )
1362 )
1363 )
1364 )
1365 )
1366 )
1367 )
1368 )
1368 )
1369 )
1370 )
1371 )
1372 )
1373 )
1374 )
1375 )
1376 )
1377 )
1378 )
1378 )
1379 )
1380 )
1381 )
1382 )
1383 )
1384 )
1385 )
1386 )
1387 )
1388 )
1388 )
1389 )
1390 )
1391 )
1392 )
1393 )
1394 )
1395 )
1396 )
1397 )
1398 )
1398 )
1399 )
1399 )
1400 )
1401 )
1402 )
1403 )
1404 )
1405 )
1406 )
1407 )
1408 )
1409 )
1409 )
1410 )
1411 )
1412 )
1413 )
1414 )
1415 )
1416 )
1417 )
1418 )
1418 )
1419 )
1420 )
1421 )
1422 )
1423 )
1424 )
1425 )
1426 )
1427 )
1428 )
1428 )
1429 )
1430 )
1431 )
1432 )
1433 )
1434 )
1435 )
1436 )
1437 )
1438 )
1438 )
1439 )
1440 )
1441 )
1442 )
1443 )
1444 )
1445 )
1446 )
1447 )
1448 )
1448 )
1449 )
1450 )
1451 )
1452 )
1453 )
1454 )
1455 )
1456 )
1457 )
1458 )
1459 )
1459 )
1460 )
1461 )
1462 )
1463 )
1464 )
1465 )
1466 )
1467 )
1468 )
1468 )
1469 )
1470 )
1471 )
1472 )
1473 )
1474 )
1475 )
1476 )
1477 )
1478 )
1478 )
1479 )
1480 )
1481 )
1482 )
1483 )
1484 )
1485 )
1486 )
1487 )
1488 )
1488 )
1489 )
1490 )
1491 )
1492 )
1493 )
1494 )
1495 )
1496 )
1497 )
1498 )
1498 )
1499 )
1499 )
1500 )
1501 )
1502 )
1503 )
1504 )
1505 )
1506 )
1507 )
1508 )
1509 )
1509 )
1510 )
1511 )
1512 )
1513 )
1514 )
1515 )
1516 )
1517 )
1518 )
1518 )
1519 )
1520 )
1521 )
1522 )
1523 )
1524 )
1525 )
1526 )
1527 )
1528 )
1528 )
1529 )
1530 )
1531 )
1532 )
1533 )
1534 )
1535 )
1536 )
1537 )
1538 )
1538 )
1539 )
1540 )
1541 )
1542 )
1543 )
1544 )
1545 )
1546 )
1547 )
1548 )
1548 )
1549 )
1550 )
1551 )
1552 )
1553 )
1554 )
1555 )
1556 )
1557 )
1558 )
1559 )
1559 )
1560 )
1561 )
1562 )
1563 )
1564 )
1565 )
1566 )
1567 )
1568 )
1568 )
1569 )
1570 )
1571 )
1572 )
1573 )
1574 )
1575 )
1576 )
1577 )
1578 )
1578 )
1579 )
1580 )
1581 )
1582 )
1583 )
1584 )
1585 )
1586 )
1587 )
1588 )
1588 )
1589 )
1590 )
1591 )
1592 )
1593 )
1594 )
1595 )
1596 )
1597 )
1598 )
1598 )
1599 )
1599 )
1600 )
1601 )
1602 )
1603 )
1604 )
1605 )
1606 )
1607 )
1608 )
1609 )
1609 )
1610 )
1611 )
1612 )
1613 )
1614 )
1615 )
1616 )
1617 )
1618 )
1618 )
1619 )
1620 )
1621 )
1622 )
1623 )
1624 )
1625 )
1626 )
1627 )
1628 )
1628 )
1629 )
1630 )
1631 )
1632 )
1633 )
1634 )
1635 )
1636 )
1637 )
1638 )
1638 )
1639 )
1640 )
1641 )
1642 )
1643 )
1644 )
1645 )
1646 )
1647 )
1648 )
1648 )
1649 )
1650 )
1651 )
1652 )
1653 )
1654 )
1655 )
1656 )
1657 )
1658 )
1659 )
1659 )
1660 )
1661 )
1662 )
1663 )
1664 )
1665 )
1666 )
1667 )
1668 )
1668 )
1669 )
1670 )
1671 )
1672 )
1673 )
1674 )
1675 )
1676 )
1677 )
1678 )
1678 )
1679 )
1680 )
1681 )
1682 )
1683 )
1684 )
1685 )
1686 )
1687 )
1688 )
1688 )
1689 )
1690 )
1691 )
1692 )
1693 )
1694 )
1695 )
1696 )
1697 )
1698 )
1698 )
1699 )
1699 )
1700 )
1701 )
1702 )
1703 )
1704 )
1705 )
1706 )
1707 )
1708 )
1709 )
1709 )
1710 )
1711 )
1712 )
1713 )
1714 )
1715 )
1716 )
1717 )
1718 )
1718 )
1719 )
1720 )
1721 )
1722 )
1723 )
1724 )
1725 )
1726 )
1727 )
1728 )
1728 )
1729 )
1730 )
1731 )
1732 )
1733 )
1734 )
1735 )
1736 )
1737 )
1738 )
1738 )
1739 )
1740 )
1741 )
1742 )
1743 )
1744 )
1745 )
1746 )
1747 )
1748 )
1748 )
1749 )
1750 )
1751 )
1752 )
1753 )
1754 )
1755 )
1756 )
1757 )
1758 )
1759 )
1759 )
1760 )
1761 )
1762 )
1763 )
1764 )
1765 )
1766 )
1767 )
1768 )
1768 )
1769 )
1770 )
1771 )
1772 )
1773 )
1774 )
1775 )
1776 )
1777 )
1778 )
1778 )
1779 )
1779 )
1780 )
1781 )
1782 )
1783 )
1784 )
1785 )
1786 )
1787 )
1788 )
1788 )
1789 )
1789 )
1790 )
1791 )
1792 )
1793 )
1794 )
1795 )
1796 )
1797 )
1798 )
1798 )
1799 )
1799 )
1800 )
1801 )
1802 )
1803 )
1804 )
1805 )
1806 )
1807 )
1808 )
1809 )
1809 )
1810 )
1811 )
1812 )
1813 )
1814 )
1815 )
1816 )
1817 )
1818 )
1818 )
1819 )
1819 )
1820 )
1821 )
1822 )
1823 )
1824 )
1825 )
1826 )
1827 )
1828 )
1828 )
1829 )
1829 )
1830 )
1831 )
1832 )
1833 )
1834 )
1835 )
1836 )
1837 )
1838 )
1838 )
1839 )
1839 )
1840 )
1841 )
1842 )
1843 )
1844 )
1845 )
1846 )
1847 )
1848 )
1848 )
1849 )
1849 )
1850 )
1851 )
1852 )
1853 )
1854 )
1855 )
1856 )
1857 )
1858 )
1859 )
1859 )
1860 )
1861 )
1862 )
1863 )
1864 )
1865 )
1866 )
1867 )
1868 )
1868 )
1869 )
1869 )
1870 )
1871 )
1872 )
1873 )
1874 )
1875 )
1876 )
1877 )
1878 )
1878 )
1879 )
1879 )
1880 )
1881 )
1882 )
1883 )
1884 )
1885 )
1886 )
1887 )
1888 )
1888 )
1889 )
1889 )
1890 )
1891 )
1892 )
1893 )
1894 )
1895 )
1896 )
1897 )
1898 )
1898 )
1899 )
1899 )
1900 )
1901 )
1902 )
1903 )
1904 )
1905 )
1906 )
1907 )
1908 )
1909 )
1909 )
1910 )
1911 )
1912 )
1913 )
1914 )
1915 )
1916 )
1917 )
1918 )
1918 )
1919 )
1919 )
1920 )
1921 )
1922 )
1923 )
1924 )
1925 )
1926 )
1927 )
1928 )
1928 )
1929 )
1929 )
1930 )
1931 )
1932 )
1933 )
1934 )
1935 )
1936 )
1937 )
1938 )
1938 )
1939 )
1939 )
1940 )
1941 )
1942 )
1943 )
1944 )
1945 )
1946 )
1947 )
1948 )
1948 )
1949 )
1949 )
1950 )
1951 )
1952 )
1953 )
1954 )
1955 )
1956 )
1957 )
1958 )
1959 )
1959 )
1960 )
1961 )
1962 )
1963 )
1964 )
1965 )
1966 )
1967 )
1968 )
1968 )
1969 )
1969 )
1970 )
1971 )
1972 )
1973 )
1974 )
1975 )
1976 )
1977 )
1978 )
1978 )
1979 )
1979 )
1980 )
1981 )
1982 )
1983 )
1984 )
1985 )
1986 )
1987 )
1988 )
1988 )
1989 )
1989 )
1990 )
1991 )
1992 )
1993 )
1994 )
1995 )
1996 )
1997 )
1998 )
1998 )
1999 )
1999 )
2000 )
2001 )
2002 )
2003 )
2004 )
2005 )
2006 )
2007 )
2008 )
2009 )
2009 )
2010 )
2011 )
2012 )
2013 )
2014 )
2015 )
2016 )
2017 )
2018 )
2018 )
2019 )
2019 )
2020 )
2021 )
2022 )
2023 )
2024 )
2025 )
2026 )
2027 )
2028 )
2029 )
2029 )
2030 )
2031 )
2032 )
2033 )
2034 )
2035 )
2036 )
2037 )
2038 )
2038 )
2039 )
2039 )
2040 )
2041 )
2042 )
2043 )
2044 )
2045 )
2046 )
2047 )
2048 )
2048 )
2049 )
2049 )
2050 )
2051 )
2052 )
2053 )
2054 )
2055 )
2056 )
2057 )
2058 )
2059 )
2059 )
2060 )
2061 )
2062 )
2063 )
2064 )
2065 )
2066 )
2067 )
2068 )
2068 )
2069 )
2069 )
2070 )
2071 )
2072 )
2073 )
2074 )
2075 )
2076 )
2077 )
2078 )
2078 )
2079 )
2079 )
2080 )
2081 )
2082 )
2083 )
2084 )
2085 )
2086 )
2087 )
2088 )
2088 )
2089 )
2089 )
2090 )
2091 )
2092 )
2093 )
2094 )
2095 )
2096 )
2097 )
2098 )
2098 )
2099 )
2099 )
2100 )
2101 )
2102 )
2103 )
2104 )
2105 )
2106 )
2107 )
2108 )
2109 )
2109 )
2110 )
2111 )
2112 )
2113 )
2114 )
2115 )
2116 )
2117 )
2118 )
2119 )
2119 )
2120 )
2121 )
2122 )
2123 )
2124 )
2125 )
2126 )
2127 )
2128 )
2129 )
2129 )
2130 )
2131 )
2132 )
2133 )
2134 )
2135 )
2136 )
2137 )
2138 )
2138 )
2139 )
2139 )
2140 )
2141 )
2142 )
2143 )
2144 )
2145 )
2146 )
2147 )
2148 )
2148 )
2149 )
2149 )
2150 )
2151 )
2152 )
2153 )
2154 )
2155 )
2156 )
2157 )
2158 )
2159 )
2159 )
2160 )
2161 )
2162 )
2163 )
2164 )
2165 )
2166 )
2167 )
2168 )
2168 )
2169 )
2169 )
2170 )
2171 )
2172 )
217
```

```

49         )
50     )
51   ))
52 }
53 }
```

Page essai standardisé 208

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(shinyCSSloaders)
5
6 Essai208Page <- function() {
7   tagList(
8     fluidPage(
9       titlePanel("Terrestrial Plant Test : Seedling Emergence and
10      Seedling Growth Test"),
11     fluidRow(
12       column(4,
13        textInput(inputId = "chemical", "Nom du produit
14          chimique testé", value = "Produit chimique"),
15         selectInput("conc_unit", "Unité de Concentration",
16           choices = c("g/L", "mg/L", "microg/L", "ng/L")),
17         fileInput('datafile_ESSAI208', 'Sélectionner un
18          fichier', accept = c('.csv')),
19         radioButtons('model_ESSAI208', 'Sélectionner le modèle
20          de régression',
21           choices = c('log-logistic 2 paramètres'
22             = '112',
23             'log-logistic 3 paramètres'
24             = '113',
25             'log-logistic 4 paramètres'
26             = '114'),
27           selected = '112'),
28         numericInput(inputId = "ecx_ESSAI208", "Déterminer la
29          concentration efficace X%", value = 50, min = 0,
30          max = 100),
31         radioButtons('test_method_ESSAI208', 'Sélectionner la
32          méthode de test d\'hypothèse',
33           choices = c('ANOVA' = 'anova'), selected
34           = 'anova'),
35         actionButton("buttonRunStep1", "Exécuter", icon =
36           icon("check"), width = "100px")
37       ),
38       column(8,
39         mainPanel(
40           width = 12,
41           DT::dataTableOutput('rawdata'),
42           br(),
43           tableOutput('drc_result'),
44           withSpinner(plotOutput('drc_plot', width = "100%",
45             height = "850px"), type = 4, color = '#9e0093'),
46           )
47         )
48       )
49     )
50   ))
51 }
52 }
```

```

32             br(),
33             withSpinner(verbatimTextOutput("test_result")),
34             br()
35         )
36     ),
37     tabPanel(
38         "Télécharger le rapport",
39         fixedRow(
40             column(
41                 8,
42                 br(),
43                 HTML("<font face=verdana size=5 color=#9e0093><b>Télé"
44                     "charger le rapport d'analyse</b></font>"),
45                 br(), br(), br(),
46                 radioButtons("format", "Sélectionner le format du"
47                     "rapport", c('Word'), inline = TRUE),
48                 downloadButton("DownloadReport", "Télécharger le"
49                     "rapport", icon = icon("fas fa-download"), style = ,
50                     background-color:#9e0093; color:#000000; border-
51                     color:#9d9d9d; )
52             )
53         )
54     )
55 )
56 )
57 )
58 )
59 }

```

Page essai standardisé 210

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(shinyCSSloaders)
5
6 Essai210Page <- function() {
7   tagList(
8     makePage(
9       "Fish, Early-life Stage Toxicity Test",
10      "(Essai 210)",
11      fluidRow(
12        column(4,
13         textInput(inputId = "chemical", "Nom du produit"
14            "chimique testé", value = "Produit chimique"),
15          selectInput("conc_unit210", "Unité de Concentration",
16            choices = c("g/L", "mg/L", "microg/L", "ng/L")),
17          fileInput('datafile_ESSAI210', '', accept = c('.csv')
18            ),
19          tags$br(),
20          selectInput('model_ESSAI210', 'Sélectionner le modèle'
21            "de régression",
22            choices = c('log-logistic 2 paramètres'
23            = '112',
24            'log-logistic 3 paramètres'
25            )
26        )
27      )
28    )
29  )
30 )
31 )
32 )
33 )
34 )
35 )
36 )
37 )
38 )
39 )
40 )
41 )
42 )
43 )
44 )
45 )
46 )
47 )
48 )
49 )
50 )
51 )
52 )
53 )
54 )
55 )
56 )
57 )
58 )
59 )
60 )
61 )
62 )
63 )
64 )
65 )
66 )
67 )
68 )
69 )
70 )
71 )
72 )
73 )
74 )
75 )
76 )
77 )
78 )
79 )
80 )
81 )
82 )
83 )
84 )
85 )
86 )
87 )
88 )
89 )
90 )
91 )
92 )
93 )
94 )
95 )
96 )
97 )
98 )
99 )
100 )
101 )
102 )
103 )
104 )
105 )
106 )
107 )
108 )
109 )
110 )
111 )
112 )
113 )
114 )
115 )
116 )
117 )
118 )
119 )
120 )
121 )
122 )
123 )
124 )
125 )
126 )
127 )
128 )
129 )
130 )
131 )
132 )
133 )
134 )
135 )
136 )
137 )
138 )
139 )
140 )
141 )
142 )
143 )
144 )
145 )
146 )
147 )
148 )
149 )
150 )
151 )
152 )
153 )
154 )
155 )
156 )
157 )
158 )
159 )
160 )
161 )
162 )
163 )
164 )
165 )
166 )
167 )
168 )
169 )
170 )
171 )
172 )
173 )
174 )
175 )
176 )
177 )
178 )
179 )
180 )
181 )
182 )
183 )
184 )
185 )
186 )
187 )
188 )
189 )
190 )
191 )
192 )
193 )
194 )
195 )
196 )
197 )
198 )
199 )
200 )
201 )
202 )
203 )
204 )
205 )
206 )
207 )
208 )
209 )
210 )
211 )
212 )
213 )
214 )
215 )
216 )
217 )
218 )
219 )
220 )
221 )
222 )
223 )
224 )
225 )
226 )
227 )
228 )
229 )
230 )
231 )
232 )
233 )
234 )
235 )
236 )
237 )
238 )
239 )
240 )
241 )
242 )
243 )
244 )
245 )
246 )
247 )
248 )
249 )
250 )
251 )
252 )
253 )
254 )
255 )
256 )
257 )
258 )
259 )
260 )
261 )
262 )
263 )
264 )
265 )
266 )
267 )
268 )
269 )
270 )
271 )
272 )
273 )
274 )
275 )
276 )
277 )
278 )
279 )
280 )
281 )
282 )
283 )
284 )
285 )
286 )
287 )
288 )
289 )
290 )
291 )
292 )
293 )
294 )
295 )
296 )
297 )
298 )
299 )
300 )
301 )
302 )
303 )
304 )
305 )
306 )
307 )
308 )
309 )
310 )
311 )
312 )
313 )
314 )
315 )
316 )
317 )
318 )
319 )
320 )
321 )
322 )
323 )
324 )
325 )
326 )
327 )
328 )
329 )
330 )
331 )
332 )
333 )
334 )
335 )
336 )
337 )
338 )
339 )
340 )
341 )
342 )
343 )
344 )
345 )
346 )
347 )
348 )
349 )
350 )
351 )
352 )
353 )
354 )
355 )
356 )
357 )
358 )
359 )
360 )
361 )
362 )
363 )
364 )
365 )
366 )
367 )
368 )
369 )
370 )
371 )
372 )
373 )
374 )
375 )
376 )
377 )
378 )
379 )
380 )
381 )
382 )
383 )
384 )
385 )
386 )
387 )
388 )
389 )
390 )
391 )
392 )
393 )
394 )
395 )
396 )
397 )
398 )
399 )
400 )
401 )
402 )
403 )
404 )
405 )
406 )
407 )
408 )
409 )
410 )
411 )
412 )
413 )
414 )
415 )
416 )
417 )
418 )
419 )
420 )
421 )
422 )
423 )
424 )
425 )
426 )
427 )
428 )
429 )
430 )
431 )
432 )
433 )
434 )
435 )
436 )
437 )
438 )
439 )
440 )
441 )
442 )
443 )
444 )
445 )
446 )
447 )
448 )
449 )
450 )
451 )
452 )
453 )
454 )
455 )
456 )
457 )
458 )
459 )
460 )
461 )
462 )
463 )
464 )
465 )
466 )
467 )
468 )
469 )
470 )
471 )
472 )
473 )
474 )
475 )
476 )
477 )
478 )
479 )
480 )
481 )
482 )
483 )
484 )
485 )
486 )
487 )
488 )
489 )
490 )
491 )
492 )
493 )
494 )
495 )
496 )
497 )
498 )
499 )
500 )
501 )
502 )
503 )
504 )
505 )
506 )
507 )
508 )
509 )
510 )
511 )
512 )
513 )
514 )
515 )
516 )
517 )
518 )
519 )
520 )
521 )
522 )
523 )
524 )
525 )
526 )
527 )
528 )
529 )
530 )
531 )
532 )
533 )
534 )
535 )
536 )
537 )
538 )
539 )
540 )
541 )
542 )
543 )
544 )
545 )
546 )
547 )
548 )
549 )
550 )
551 )
552 )
553 )
554 )
555 )
556 )
557 )
558 )
559 )
560 )
561 )
562 )
563 )
564 )
565 )
566 )
567 )
568 )
569 )
570 )
571 )
572 )
573 )
574 )
575 )
576 )
577 )
578 )
579 )
580 )
581 )
582 )
583 )
584 )
585 )
586 )
587 )
588 )
589 )
590 )
591 )
592 )
593 )
594 )
595 )
596 )
597 )
598 )
599 )
600 )
601 )
602 )
603 )
604 )
605 )
606 )
607 )
608 )
609 )
610 )
611 )
612 )
613 )
614 )
615 )
616 )
617 )
618 )
619 )
620 )
621 )
622 )
623 )
624 )
625 )
626 )
627 )
628 )
629 )
630 )
631 )
632 )
633 )
634 )
635 )
636 )
637 )
638 )
639 )
640 )
641 )
642 )
643 )
644 )
645 )
646 )
647 )
648 )
649 )
650 )
651 )
652 )
653 )
654 )
655 )
656 )
657 )
658 )
659 )
660 )
661 )
662 )
663 )
664 )
665 )
666 )
667 )
668 )
669 )
670 )
671 )
672 )
673 )
674 )
675 )
676 )
677 )
678 )
679 )
680 )
681 )
682 )
683 )
684 )
685 )
686 )
687 )
688 )
689 )
690 )
691 )
692 )
693 )
694 )
695 )
696 )
697 )
698 )
699 )
700 )
701 )
702 )
703 )
704 )
705 )
706 )
707 )
708 )
709 )
710 )
711 )
712 )
713 )
714 )
715 )
716 )
717 )
718 )
719 )
720 )
721 )
722 )
723 )
724 )
725 )
726 )
727 )
728 )
729 )
7210 )
7211 )
7212 )
7213 )
7214 )
7215 )
7216 )
7217 )
7218 )
7219 )
7220 )
7221 )
7222 )
7223 )
7224 )
7225 )
7226 )
7227 )
7228 )
7229 )
72210 )
72211 )
72212 )
72213 )
72214 )
72215 )
72216 )
72217 )
72218 )
72219 )
72220 )
72221 )
72222 )
72223 )
72224 )
72225 )
72226 )
72227 )
72228 )
72229 )
722210 )
722211 )
722212 )
722213 )
722214 )
722215 )
722216 )
722217 )
722218 )
722219 )
722220 )
722221 )
722222 )
722223 )
722224 )
722225 )
722226 )
722227 )
722228 )
722229 )
7222210 )
7222211 )
7222212 )
7222213 )
7222214 )
7222215 )
7222216 )
7222217 )
7222218 )
7222219 )
7222220 )
7222221 )
7222222 )
7222223 )
7222224 )
7222225 )
7222226 )
7222227 )
7222228 )
7222229 )
72222210 )
72222211 )
72222212 )
72222213 )
72222214 )
72222215 )
72222216 )
72222217 )
72222218 )
72222219 )
72222220 )
72222221 )
72222222 )
72222223 )
72222224 )
72222225 )
72222226 )
72222227 )
72222228 )
72222229 )
722222210 )
722222211 )
722222212 )
722222213 )
722222214 )
722222215 )
722222216 )
722222217 )
722222218 )
722222219 )
722222220 )
722222221 )
722222222 )
722222223 )
722222224 )
722222225 )
722222226 )
722222227 )
722222228 )
722222229 )
7222222210 )
7222222211 )
7222222212 )
7222222213 )
7222222214 )
7222222215 )
7222222216 )
7222222217 )
7222222218 )
7222222219 )
7222222220 )
7222222221 )
7222222222 )
7222222223 )
7222222224 )
7222222225 )
7222222226 )
7222222227 )
7222222228 )
7222222229 )
72222222210 )
72222222211 )
72222222212 )
72222222213 )
72222222214 )
72222222215 )
72222222216 )
72222222217 )
72222222218 )
72222222219 )
72222222220 )
72222222221 )
72222222222 )
72222222223 )
72222222224 )
72222222225 )
72222222226 )
72222222227 )
72222222228 )
72222222229 )
722222222210 )
722222222211 )
722222222212 )
722222222213 )
722222222214 )
722222222215 )
722222222216 )
722222222217 )
722222222218 )
722222222219 )
722222222220 )
722222222221 )
722222222222 )
722222222223 )
722222222224 )
722222222225 )
722222222226 )
722222222227 )
722222222228 )
722222222229 )
7222222222210 )
7222222222211 )
7222222222212 )
7222222222213 )
7222222222214 )
7222222222215 )
7222222222216 )
7222222222217 )
7222222222218 )
7222222222219 )
7222222222220 )
7222222222221 )
7222222222222 )
7222222222223 )
7222222222224 )
7222222222225 )
7222222222226 )
7222222222227 )
7222222222228 )
7222222222229 )
72222222222210 )
72222222222211 )
72222222222212 )
72222222222213 )
72222222222214 )
72222222222215 )
72222222222216 )
72222222222217 )
72222222222218 )
72222222222219 )
72222222222220 )
72222222222221 )
72222222222222 )
72222222222223 )
72222222222224 )
72222222222225 )
72222222222226 )
72222222222227 )
72222222222228 )
72222222222229 )
722222222222210 )
722222222222211 )
722222222222212 )
722222222222213 )
722222222222214 )
722222222222215 )
722222222222216 )
722222222222217 )
722222222222218 )
722222222222219 )
722222222222220 )
722222222222221 )
722222222222222 )
722222222222223 )
722222222222224 )
722222222222225 )
722222222222226 )
722222222222227 )
722222222222228 )
722222222222229 )
7222222222222210 )
7222222222222211 )
7222222222222212 )
7222222222222213 )
7222222222222214 )
7222222222222215 )
7222222222222216 )
7222222222222217 )
7222222222222218 )
7222222222222219 )
7222222222222220 )
7222222222222221 )
7222222222222222 )
7222222222222223 )
7222222222222224 )
7222222222222225 )
7222222222222226 )
7222222222222227 )
7222222222222228 )
7222222222222229 )
72222222222222210 )
72222222222222211 )
72222222222222212 )
72222222222222213 )
72222222222222214 )
72222222222222215 )
72222222222222216 )
72222222222222217 )
72222222222222218 )
72222222222222219 )
72222222222222220 )
72222222222222221 )
72222222222222222 )
72222222222222223 )
72222222222222224 )
72222222222222225 )
72222222222222226 )
72222222222222227 )
72222222222222228 )
72222222222222229 )
722222222222222210 )
722222222222222211 )
722222222222222212 )
722222222222222213 )
722222222222222214 )
722222222222222215 )
722222222222222216 )
722222222222222217 )
722222222222222218 )
722222222222222219 )
722222222222222220 )
722222222222222221 )
722222222222222222 )
722222222222222223 )
722222222222222224 )
722222222222222225 )
722222222222222226 )
722222222222222227 )
722222222222222228 )
722222222222222229 )
7222222222222222210 )
7222222222222222211 )
7222222222222222212 )
7222222222222222213 )
7222222222222222214 )
7222222222222222215 )
7222222222222222216 )
7222222222222222217 )
7222222222222222218 )
7222222222222222219 )
7222222222222222220 )
7222222222222222221 )
7222222222222222222 )
7222222222222222223 )
7222222222222222224 )
7222222222222222225 )
7222222222222222226 )
7222222222222222227 )
7222222222222222228 )
7222222222222222229 )
72222222222222222210 )
72222222222222222211 )
72222222222222222212 )
72222222222222222213 )
72222222222222222214 )
72222222222222222215 )
72222222222222222216 )
72222222222222222217 )
72222222222222222218 )
72222222222222222219 )
72222222222222222220 )
72222222222222222221 )
72222222222222222222 )
72222222222222222223 )
72222222222222222224 )
72222222222222222225 )
72222222222222222226 )
72222222222222222227 )
72222222222222222228 )
72222222222222222229 )
722222222222222222210 )
722222222222222222211 )
722222222222222222212 )
722222222222222222213 )
722222222222222222214 )
722222222222222222215 )
722222222222222222216 )
722222222222222222217 )
722222222222222222218 )
722222222222222222219 )
722222222222222222220 )
722222222222222222221 )
722222222222222222222 )
722222222222222222223 )
722222222222222222224 )
722222222222222222225 )
722222222222222222226 )
722222222222222222227 )
722222222222222222228 )
722222222222222222229 )
7222222222222222222210 )
7222222222222222222211 )
7222222222222222222212 )
7222222222222222222213 )
7222222222222222222214 )
7222222222222222222215 )
7222222222222222222216 )
7222222222222222222217 )
7222222222222222222218 )
7222222222222222222219 )
7222222222222222222220 )
7222222222222222222221 )
7222222222222222222222 )
7222222222222222222223 )
7222222222222222222224 )
7222222222222222222225 )
7222222222222222222226 )
7222222222222222222227 )
7222222222222222222228 )
7222222222222222222229 )
72222222222222222222210 )
72222222222222222222211 )
72222222222222222222212 )
72222222222222222222213 )
72222222222222222222214 )
72222222222222222222215 )
72222222222222222222216 )
72222222222222222222217 )
72222222222222222222218 )
72222222222222222222219 )
72222222222222222222220 )
72222222222222222222221 )
72222222222222222222222 )
72222222222222222222223 )
72222222222222222222224 )
72222222222222222222225 )
72222222222222222222226 )
72222222222222222222227 )
72222222222222222222228 )
72222222222222222222229 )
722222222222222222222210 )
722222222222222222222211 )
722222222222222222222212 )
722222222222222222222213 )
722222222222222222222214 )
722222222222222222222215 )
722222222222222222222216 )
722222222222222222222217 )
722222222222222222222218 )
722222222222222222222219 )
722222222222222222222220 )
722222222222222222222221 )
722222222222222222222222 )
722222222222222222222223 )
722222222222222222222224 )
722222222222222222222225 )
722222222222222222222226 )
722222222222222222222227 )
722222222222222222222228 )
722222222222222222222229 )
7222222222222222222222210 )
7222222222222222222222211 )
7222222222222222222222212 )
7222222222222222222222213 )
7222222222222222222222214 )
7222222222222222222222215 )
7222222222222222222222216 )
7222222222222222222222217 )
7222222222222222222222218 )
7222222222222222222222219 )
7222222222222222222222220 )
7222222222222222222222221 )
7222222222222222222222222 )
7222222222222222222222223 )
7222222222222222222222224 )
7222222222222222222222225 )
7222222222222222222222226 )
7222222222222222222222227 )
7222222222222222222222228 )
7222222222222222222222229 )
72222222222222222222222210 )
72222222222222222222222211 )
72222222222222222222222212 )
72222222222222222222222213 )
72222222222222222222222214 )
72222222222222222222222215 )
72222222222222222222222216 )
72222222222222222222222217 )
72222222222222222222222218 )
72222222222222222222222219 )
72222222222222222222222220 )
72222222222222222222222221 )
72222222222222222222222222 )
72222222222222222222222223 )
72222222222222222222222224 )
72222222222222222222222225 )
72222222222222222222222226 )
72222222222222222222222227 )
72222222222222222222222228 )
72222222222222222222222229 )
722222222222222222222222210 )
722222222222222222222222211 )
722222222222222222222222212 )
722222222222222222222222213 )
722222222222222222222222214 )
722222222222222222222222215 )
722222222222222222222222216 )
722222222222222222222222217 )
722222222222222222222222218 )
722222222222222222222222219 )
722222222222222222222222220 )
722222222222222222222222221 )
722222222222222222222222222 )
722222222222222222222222223 )
722222222222222222222222224 )
722222222222222222222222225 )
722222
```

```

20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
}

```

Page essai standardisé 211

```

1 library(shiny)
2 library(DT)

```

```

3 | library(drc)
4 | library(multcomp)
5 | library(shinyCSSloaders)
6 |
7 | Essai211Page <- function() {
8 |   tagList(
9 |     makePage(
10 |       "Daphnia magna Reproduction Test",
11 |       "(Essai 211)",
12 |       fluidRow(
13 |         column(4,
14 |          textInput(inputId = "chemical", "Nom du produit
15 |                         chimique testé", value = "Produit chimique"),
16 |           selectInput("conc_unit211", "Unité de Concentration",
17 |                         choices = c("g/L", "mg/L", "microg/L", "ng/L")),
18 |           fileInput('datafile_ESSAI211', ''),
19 |                         accept = c('.csv')),
20 |           tags$br(),
21 |           selectInput('model_ESSAI211', 'Sélectionner le modèle
22 |                         de régression',
23 |                         choices = c('linéaire' = 'lm',
24 |                                     'non-linéaire' = 'nls',
25 |                                     'mixte' = 'lme'),
26 |                         selected = 'lm'),
27 |           numericInput(inputId = "ecx_ESSAI211", "Déterminer la
28 |                         concentration efficace X%", value = 50, min = 0,
29 |                         max = 100),
30 |           selectInput('test_method_ESSAI211', 'Sélectionner la
31 |                         méthode de test d\'hypothèse',
32 |                         choices = c('ANOVA' = 'anova'), selected
33 |                         = 'anova'),
34 |           actionButton("buttonRunStep211", "Exécuter", icon =
35 |                         icon("check"), width = "100px")
36 |           ),
37 |           column(8,
38 |             mainPanel(
39 |               width = 12,
40 |               DT::dataTableOutput('rawdata'),
41 |               br(),
42 |               tableOutput('drc_result'),
43 |               withSpinner(plotOutput('drc_plot', width = "100%", height =
44 |               "850px"), type = 4, color = "#9e0093"),
45 |               br(),
46 |               withSpinner(verbatimTextOutput("test_result")),
47 |               br()
48 |             )
49 |           ),
50 |           tabPanel(
51 |             "Télécharger le rapport",
52 |             fixedRow(
53 |               column(
54 |                 8,
55 |                 )
56 |               )
57 |             )
58 |           )
59 |         )
60 |       )
61 |     )
62 |   )
63 | }

```

```

45
46     br(),
47     HTML( "<font face=verdana size=5 color=#9e0093><b>Télé"
48           charger le rapport d'analyse</b></font>" ),
49     br(), br(), br(),
50     radioButtons("format", "Sélectionner le format du"
51                   rapport", c('Word'), inline = TRUE),
52     downloadButton("DownloadReport", "Télécharger le"
53                   rapport", icon = icon("fas fa-download"), style = "
54                   background-color:#9e0093; color:#000000; border-
55                   color:#9d9d9d; ")
56   )
57 )
58 )
59 )
60 )
61 )
62 )
63 )
64 )
65 }

```

Page essai standardisé 216

```

1 library(shiny)
2 library(DT)
3 library(nlme)
4 library(shinyCSSloaders)

5
6 Essai216Page <- function() {
7   tagList(
8     fluidPage(
9       titlePanel("Soil Microorganisms : Nitrogen Transformation Test",
10      ),
11      fluidRow(
12        column(4,
13         textInput(inputId = "chemical", "Nom du produit chimique testé",
14            value = "Produit chimique"),
15          selectInput("conc_unit", "Unité de Concentration",
16            choices = c("g/kg", "mg/kg", "microg/kg", "ng/kg")),
17          fileInput('datafile_ESSAI216', 'Sélectionner un fichier',
18            accept = c('.csv')),
19          radioButtons('model_ESSAI216', 'Sélectionner le modèle de régression',
20            choices = c('Linéaire' = 'lm', 'Non-linéaire' = 'nls',
21            'Mixte' = 'lme'),
22            selected = 'lm'),
23          numericInput(inputId = "ecx_ESSAI216", "Déterminer la concentration efficace X %",
24            value = 50, min = 0, max = 100),
25          radioButtons('test_method_ESSAI216', 'Sélectionner la méthode de test d'hypothèse',
26            choices = c('ANOVA' = 'anova'), selected = 'anova'),
27          actionButton("buttonRunStep1", "Exécuter", icon = icon("check"),
28            width = "100px")
29        ),
30      )
31    )
32  )
33}

```

Page essai standardisé 218

```

15     fileInput('datafile_ESSAI218', ''),
16     tags$br(),
17     selectInput('model_E218_mortality',
18                 'Select fitting model for mortality',
19                 choices = c('log-logistic 2 parameters'
20                             = '112',
21                             'log-logistic 3 parameters'
22                             = '113',
23                             'log-logistic 4 parameters'
24                             = '114'),
25                 selected = '112'),
26     selectInput('model_E218_emergence',
27                 'Select fitting model for emergence',
28                 choices = c('log-logistic 2 parameters'
29                             = '112',
30                             'log-logistic 3 parameters'
31                             = '113',
32                             'log-logistic 4 parameters'
33                             = '114'),
34                 selected = '112'),
35     selectInput('model_E218_development',
36                 'Select fitting model for development',
37                 choices = c('log-logistic 3 parameters'
38                             = '113',
39                             'log-logistic 4 parameters'
40                             = '114'),
41                 selected = '113'),
42     tags$br(),
43     numericInput(inputId = "ecx_ESSAI218", "Determine"
44                  effect concentration X%", value = 50, min = 0, max
45                  = 100),
46     tags$br(),
47     selectInput('test_method_E218_mortality',
48                 'Select hypothesis testing method for mortality',
49                 choices = c("Cochran-Armitage test" = 'CA',
50                             "Fisher's exact test with BH
51                             correction" = 'Fisher'),
52                 selected = 'CA'),
53     selectInput('test_method_E218_emergence',
54                 'Select hypothesis testing method for emergence ratio',
55                 choices = c("Cochran-Armitage test" = 'CA',
56                             "Fisher's exact test with BH
57                             correction" = 'Fisher'),
58                 selected = 'CA'),
59     selectInput('test_method_E218_development',
60                 'Select hypothesis testing method for

```

```

49                               development_rate',
50                               choices = c("Dunnett's test" = 'Dunnett',
51                                         ,
52                                         "Steel's test" = 'Steel'),
53                                         selected = 'Dunnett'),
54                                         h5("You can see the bartlett's test result for
55                                         homogeneity of variance of development rate, and
56                                         then select testing method."),
57                                         actionButton("buttonRunStep218", "Exécuter", icon =
58                                         icon("check"), width = "100px")
59                                         ),
60                                         column(8,
61                                         mainPanel(
62                                         width = 12,
63                                         DT::dataTableOutput('rawdata218'),
64                                         br(),
65                                         tableOutput('drc_result218'),
66                                         withSpinner(plotOutput('drc_plot218', width = "100%
67                                         ", height = "850px"), type = 4, color = '#9e0093
68                                         '),
69                                         br(),
70                                         withSpinner(verbatimTextOutput("test_result218")),
71                                         br()
72                                         )
73                                         ),
74                                         tabPanel(
75                                         "Télécharger le rapport",
76                                         fixedRow(
77                                         column(
78                                         8,
79                                         br(),
80                                         HTML("<font face=verdana size=5 color=#9e0093><b>Télé
81                                         charger le rapport d'analyse</b></font>"),
82                                         br(), br(), br(),
83                                         radioButtons("format", "Sélectionner le format du
84                                         rapport", c('Word'), inline = TRUE),
85                                         downloadButton("DownloadReport", "Télécharger le
86                                         rapport", icon = icon("fas fa-download"), style = ,
87                                         background-color:#9e0093; color:#000000; border-
88                                         color:#9d9d9d;")
89                                         )
90                                         )
91                                         )
92                                         )
93                                         )
94                                         )
95                                         )
96                                         )
97                                         )
98                                         )
99                                         )
100                                         )
101                                         )
102                                         )
103                                         )
104                                         )
105                                         )
106                                         )
107                                         )
108                                         )
109                                         )
110                                         )
111                                         )
112                                         )
113                                         )
114                                         )
115                                         )
116                                         )
117                                         )
118                                         )
119                                         )
120                                         )
121                                         )
122                                         )
123                                         )
124                                         )
125                                         )
126                                         )
127                                         )
128                                         )
129                                         )
130                                         )
131                                         )
132                                         )
133                                         )
134                                         )
135                                         )
136                                         )
137                                         )
138                                         )
139                                         )
140                                         )
141                                         )
142                                         )
143                                         )
144                                         )
145                                         )
146                                         )
147                                         )
148                                         )
149                                         )
150                                         )
151                                         )
152                                         )
153                                         )
154                                         )
155                                         )
156                                         )
157                                         )
158                                         )
159                                         )
150                                         )
151                                         )
152                                         )
153                                         )
154                                         )
155                                         )
156                                         )
157                                         )
158                                         )
159                                         )
160                                         )
161                                         )
162                                         )
163                                         )
164                                         )
165                                         )
166                                         )
167                                         )
168                                         )
169                                         )
170                                         )
171                                         )
172                                         )
173                                         )
174                                         )
175                                         )
176                                         )
177                                         )
178                                         )
179                                         )
180                                         )
181                                         )
182                                         )
183                                         )
184                                         )
185                                         )
186                                         )
187                                         )
188                                         )
189                                         )
190                                         )
191                                         )
192                                         )
193                                         )
194                                         )
195                                         )
196                                         )
197                                         )
198                                         )
199                                         )
200                                         )
201                                         )
202                                         )
203                                         )
204                                         )
205                                         )
206                                         )
207                                         )
208                                         )
209                                         )
210                                         )
211                                         )
212                                         )
213                                         )
214                                         )
215                                         )
216                                         )
217                                         )
218                                         )
219                                         )
220                                         )
221                                         )
222                                         )
223                                         )
224                                         )
225                                         )
226                                         )
227                                         )
228                                         )
229                                         )
230                                         )
231                                         )
232                                         )
233                                         )
234                                         )
235                                         )
236                                         )
237                                         )
238                                         )
239                                         )
240                                         )
241                                         )
242                                         )
243                                         )
244                                         )
245                                         )
246                                         )
247                                         )
248                                         )
249                                         )
250                                         )
251                                         )
252                                         )
253                                         )
254                                         )
255                                         )
256                                         )
257                                         )
258                                         )
259                                         )
260                                         )
261                                         )
262                                         )
263                                         )
264                                         )
265                                         )
266                                         )
267                                         )
268                                         )
269                                         )
270                                         )
271                                         )
272                                         )
273                                         )
274                                         )
275                                         )
276                                         )
277                                         )
278                                         )
279                                         )
280                                         )
281                                         )
282                                         )
283                                         )
284                                         )
285                                         )
286                                         )
287                                         )
288                                         )
289                                         )
290                                         )
291                                         )
292                                         )
293                                         )
294                                         )
295                                         )
296                                         )
297                                         )
298                                         )
299                                         )
300                                         )
301                                         )
302                                         )
303                                         )
304                                         )
305                                         )
306                                         )
307                                         )
308                                         )
309                                         )
310                                         )
311                                         )
312                                         )
313                                         )
314                                         )
315                                         )
316                                         )
317                                         )
318                                         )
319                                         )
320                                         )
321                                         )
322                                         )
323                                         )
324                                         )
325                                         )
326                                         )
327                                         )
328                                         )
329                                         )
330                                         )
331                                         )
332                                         )
333                                         )
334                                         )
335                                         )
336                                         )
337                                         )
338                                         )
339                                         )
340                                         )
341                                         )
342                                         )
343                                         )
344                                         )
345                                         )
346                                         )
347                                         )
348                                         )
349                                         )
350                                         )
351                                         )
352                                         )
353                                         )
354                                         )
355                                         )
356                                         )
357                                         )
358                                         )
359                                         )
360                                         )
361                                         )
362                                         )
363                                         )
364                                         )
365                                         )
366                                         )
367                                         )
368                                         )
369                                         )
370                                         )
371                                         )
372                                         )
373                                         )
374                                         )
375                                         )
376                                         )
377                                         )
378                                         )
379                                         )
380                                         )
381                                         )
382                                         )
383                                         )
384                                         )
385                                         )
386                                         )
387                                         )
388                                         )
389                                         )
390                                         )
391                                         )
392                                         )
393                                         )
394                                         )
395                                         )
396                                         )
397                                         )
398                                         )
399                                         )
400                                         )
401                                         )
402                                         )
403                                         )
404                                         )
405                                         )
406                                         )
407                                         )
408                                         )
409                                         )
410                                         )
411                                         )
412                                         )
413                                         )
414                                         )
415                                         )
416                                         )
417                                         )
418                                         )
419                                         )
420                                         )
421                                         )
422                                         )
423                                         )
424                                         )
425                                         )
426                                         )
427                                         )
428                                         )
429                                         )
430                                         )
431                                         )
432                                         )
433                                         )
434                                         )
435                                         )
436                                         )
437                                         )
438                                         )
439                                         )
440                                         )
441                                         )
442                                         )
443                                         )
444                                         )
445                                         )
446                                         )
447                                         )
448                                         )
449                                         )
450                                         )
451                                         )
452                                         )
453                                         )
454                                         )
455                                         )
456                                         )
457                                         )
458                                         )
459                                         )
460                                         )
461                                         )
462                                         )
463                                         )
464                                         )
465                                         )
466                                         )
467                                         )
468                                         )
469                                         )
470                                         )
471                                         )
472                                         )
473                                         )
474                                         )
475                                         )
476                                         )
477                                         )
478                                         )
479                                         )
480                                         )
481                                         )
482                                         )
483                                         )
484                                         )
485                                         )
486                                         )
487                                         )
488                                         )
489                                         )
490                                         )
491                                         )
492                                         )
493                                         )
494                                         )
495                                         )
496                                         )
497                                         )
498                                         )
499                                         )
500                                         )
501                                         )
502                                         )
503                                         )
504                                         )
505                                         )
506                                         )
507                                         )
508                                         )
509                                         )
510                                         )
511                                         )
512                                         )
513                                         )
514                                         )
515                                         )
516                                         )
517                                         )
518                                         )
519                                         )
520                                         )
521                                         )
522                                         )
523                                         )
524                                         )
525                                         )
526                                         )
527                                         )
528                                         )
529                                         )
530                                         )
531                                         )
532                                         )
533                                         )
534                                         )
535                                         )
536                                         )
537                                         )
538                                         )
539                                         )
540                                         )
541                                         )
542                                         )
543                                         )
544                                         )
545                                         )
546                                         )
547                                         )
548                                         )
549                                         )
550                                         )
551                                         )
552                                         )
553                                         )
554                                         )
555                                         )
556                                         )
557                                         )
558                                         )
559                                         )
560                                         )
561                                         )
562                                         )
563                                         )
564                                         )
565                                         )
566                                         )
567                                         )
568                                         )
569                                         )
570                                         )
571                                         )
572                                         )
573                                         )
574                                         )
575                                         )
576                                         )
577                                         )
578                                         )
579                                         )
580                                         )
581                                         )
582                                         )
583                                         )
584                                         )
585                                         )
586                                         )
587                                         )
588                                         )
589                                         )
590                                         )
591                                         )
592                                         )
593                                         )
594                                         )
595                                         )
596                                         )
597                                         )
598                                         )
599                                         )
599                                         )
600                                         )
601                                         )
602                                         )
603                                         )
604                                         )
605                                         )
606                                         )
607                                         )
608                                         )
609                                         )
610                                         )
611                                         )
612                                         )
613                                         )
614                                         )
615                                         )
616                                         )
617                                         )
618                                         )
619                                         )
620                                         )
621                                         )
622                                         )
623                                         )
624                                         )
625                                         )
626                                         )
627                                         )
628                                         )
629                                         )
630                                         )
631                                         )
632                                         )
633                                         )
634                                         )
635                                         )
636                                         )
637                                         )
638                                         )
639                                         )
640                                         )
641                                         )
642                                         )
643                                         )
644                                         )
645                                         )
646                                         )
647                                         )
648                                         )
649                                         )
650                                         )
651                                         )
652                                         )
653                                         )
654                                         )
655                                         )
656                                         )
657                                         )
658                                         )
659                                         )
660                                         )
661                                         )
662                                         )
663                                         )
664                                         )
665                                         )
666                                         )
667                                         )
668                                         )
669                                         )
670                                         )
671                                         )
672                                         )
673                                         )
674                                         )
675                                         )
676                                         )
677                                         )
678                                         )
679                                         )
680                                         )
681                                         )
682                                         )
683                                         )
684                                         )
685                                         )
686                                         )
687                                         )
688                                         )
689                                         )
690                                         )
691                                         )
692                                         )
693                                         )
694                                         )
695                                         )
696                                         )
697                                         )
698                                         )
699                                         )
699                                         )
700                                         )
701                                         )
702                                         )
703                                         )
704                                         )
705                                         )
706                                         )
707                                         )
708                                         )
709                                         )
710                                         )
711                                         )
712                                         )
713                                         )
714                                         )
715                                         )
716                                         )
717                                         )
718                                         )
719                                         )
720                                         )
721                                         )
722                                         )
723                                         )
724                                         )
725                                         )
726                                         )
727                                         )
728                                         )
729                                         )
729                                         )
730                                         )
731                                         )
732                                         )
733                                         )
734                                         )
735                                         )
736                                         )
737                                         )
738                                         )
739                                         )
739                                         )
740                                         )
741                                         )
742                                         )
743                                         )
744                                         )
745                                         )
746                                         )
747                                         )
748                                         )
749                                         )
749                                         )
750                                         )
751                                         )
752                                         )
753                                         )
754                                         )
755                                         )
756                                         )
757                                         )
758                                         )
759                                         )
759                                         )
760                                         )
761                                         )
762                                         )
763                                         )
764                                         )
765                                         )
766                                         )
767                                         )
768                                         )
769                                         )
769                                         )
770                                         )
771                                         )
772                                         )
773                                         )
774                                         )
775                                         )
776                                         )
777                                         )
778                                         )
779                                         )
779                                         )
780                                         )
781                                         )
782                                         )
783                                         )
784                                         )
785                                         )
786                                         )
787                                         )
788                                         )
788                                         )
789                                         )
789                                         )
790                                         )
791                                         )
792                                         )
793                                         )
794                                         )
795                                         )
796                                         )
797                                         )
798                                         )
799                                         )
799                                         )
800                                         )
801                                         )
802                                         )
803                                         )
804                                         )
805                                         )
806                                         )
807                                         )
808                                         )
809                                         )
809                                         )
810                                         )
811                                         )
812                                         )
813                                         )
814                                         )
815                                         )
816                                         )
817                                         )
818                                         )
819                                         )
819                                         )
820                                         )
821                                         )
822                                         )
823                                         )
824                                         )
825                                         )
826                                         )
827                                         )
828                                         )
829                                         )
829                                         )
830                                         )
831                                         )
832                                         )
833                                         )
834                                         )
835                                         )
836                                         )
837                                         )
838                                         )
839                                         )
839                                         )
840                                         )
841                                         )
842                                         )
843                                         )
844                                         )
845                                         )
846                                         )
847                                         )
848                                         )
849                                         )
849                                         )
850                                         )
851                                         )
852                                         )
853                                         )
854                                         )
855                                         )
856                                         )
857                                         )
858                                         )
859                                         )
859                                         )
860                                         )
861                                         )
862                                         )
863                                         )
864                                         )
865                                         )
866                                         )
867                                         )
868                                         )
869                                         )
869                                         )
870                                         )
871                                         )
872                                         )
873                                         )
874                                         )
875                                         )
876                                         )
877                                         )
878                                         )
879                                         )
879                                         )
880                                         )
881                                         )
882                                         )
883                                         )
884                                         )
885                                         )
886                                         )
887                                         )
888                                         )
888                                         )
889                                         )
889                                         )
890                                         )
891                                         )
892                                         )
893                                         )
894                                         )
895                                         )
896                                         )
897                                         )
898                                         )
898                                         )
899                                         )
899                                         )
900                                         )
901                                         )
902                                         )
903                                         )
904                                         )
905                                         )
906                                         )
907                                         )
908                                         )
909                                         )
909                                         )
910                                         )
911                                         )
912                                         )
913                                         )
914                                         )
915                                         )
916                                         )
917                                         )
918                                         )
919                                         )
919                                         )
920                                         )
921                                         )
922                                         )
923                                         )
924                                         )
925                                         )
926                                         )
927                                         )
928                                         )
929                                         )
929                                         )
930                                         )
931                                         )
932                                         )
933                                         )
934                                         )
935                                         )
936                                         )
937                                         )
938                                         )
939                                         )
939                                         )
940                                         )
941                                         )
942                                         )
943                                         )
944                                         )
945                                         )
946                                         )
947                                         )
948                                         )
949                                         )
949                                         )
950                                         )
951                                         )
952                                         )
953                                         )
954                                         )
955                                         )
956                                         )
957                                         )
958                                         )
959                                         )
959                                         )
960                                         )
961                                         )
962                                         )
963                                         )
964                                         )
965                                         )
966                                         )
967                                         )
968                                         )
969                                         )
969                                         )
970                                         )
971                                         )
972                                         )
973                                         )
974                                         )
975                                         )
976                                         )
977                                         )
978                                         )
979                                         )
979                                         )
980                                         )
981                                         )
982                                         )
983                                         )
984                                         )
985                                         )
986                                         )
987                                         )
988                                         )
988                                         )
989                                         )
989                                         )
990                                         )
991                                         )
992                                         )
993                                         )
994                                         )
995                                         )
996                                         )
997                                         )
998                                         )
999                                         )
999                                         )
1000                                         )
1001                                         )
1002                                         )
1003                                         )
1004                                         )
1005                                         )
1006                                         )
1007                                         )
1008                                         )
1009                                         )
1009                                         )
1010                                         )
1011                                         )
1012                                         )
1013                                         )
1014                                         )
1015                                         )
1016                                         )
1017                                         )
1018                                         )
1019                                         )
1019                                         )
1020                                         )
1021                                         )
1022                                         )
1023                                         )
1024                                         )
1025                                         )
1026                                         )
1027                                         )
1028                                         )
1029                                         )
1029                                         )
1030                                         )
1031                                         )
1032                                         )
1033                                         )
1034                                         )
1035                                         )
1036                                         )
1037                                         )
1038                                         )
1039                                         )
1039                                         )
1040                                         )
1041                                         )
1042                                         )
1043                                         )
1044                                         )
1045                                         )
1046                                         )
1047                                         )
1048                                         )
1049                                         )
1049                                         )
1050                                         )
1051                                         )
1052                                         )
1053                                         )
1054                                         )
1055                                         )
1056                                         )
1057                                         )
1058                                         )
1059                                         )
1059                                         )
1060                                         )
1061                                         )
1062                                         )
1063                                         )
1064                                         )
1065                                         )
1066                                         )
1067                                         )
1068                                         )
1069                                         )
1069                                         )
1070                                         )
1071                                         )
1072                                         )
1073                                         )
1074                                         )
1075                                         )
1076                                         )
1077                                         )
1078                                         )
1079                                         )
1079                                         )
1080                                         )
1081                                         )
1082                                         )
1083                                         )
1084                                         )
1085                                         )
1086                                         )
1087                                         )
1088                                         )
1088                                         )
1089                                         )
1089                                         )
1090                                         )
1091                                         )
1092                                         )
1093                                         )
1094                                         )
1095                                         )
1095                                         )
1096                                         )
1097                                         )
1098                                         )
1099                                         )
1099                                         )
1100                                         )
1101                                         )
1102                                         )
1103                                         )
1104                                         )
1105                                         )
1106                                         )
1107                                         )
1108                                         )
1109                                         )
1109                                         )
1110                                         )
1111                                         )
1112                                         )
1113                                         )
1114                                         )
1115                                         )
1116                                         )
1117                                         )
1118                                         )
1119                                         )
1119                                         )
1120                                         )
1121                                         )
1122                                         )
1123                                         )
1124                                         )
1125                                         )
1126                                         )
1127                                         )
1128                                         )
1129                                         )
1129                                         )
1130                                         )
1131                                         )
1132                                         )
1133                                         )
1134                                         )
1135                                         )
1136                                         )
1137                                         )
1138                                         )
1138                                         )
1139                                         )
1139                                         )
1140                                         )
1141                                         )
1142                                         )
1143                                         )
1144                                         )
1145                                         )
1146                                         )
1147                                         )
1148                                         )
1148                                         )
1149                                         )
1149                                         )
1150                                         )
1151                                         )
1152                                         )
1153                                         )
1154                                         )
1155                                         )
1156                                         )
1157                                         )
1158                                         )
1158                                         )
1159                                         )
1159                                         )
1160                                         )
1161                                         )
1162                                         )
1163                                         )
1164                                         )
1165                                         )
1166                                         )
1167                                         )
1168                                         )
1169                                         )
1169                                         )
1170                                         )
1171                                         )
1172                                         )
1173                                         )
1174                                         )
1175                                         )
1176                                         )
1177                                         )
1178                                         )
1179                                         )
1179                                         )
1180                                         )
1181                                         )
1182                                         )
1183                                         )
1184                                         )
1185                                         )
1186                                         )
1187                                         )
1188                                         )
1188                                         )
1189                                         )
1189                                         )
1190                                         )
1191                                         )
1192                                         )
1193                                         )
1194                                         )
1195                                         )
1195                                         )
1196                                         )
1197                                         )
1198                                         )
1199                                         )
1199                                         )
1200                                         )
1201                                         )
1202                                         )
1203                                         )
1204                                         )
1205                                         )
1206                                         )
1207                                         )
1208                                         )
1209                                         )
1209                                         )
1210                                         )
1211                                         )
1212                                         )
1213                                         )
1214                                         )
1215                                         )
1216                                         )
1217                                         )
1218                                         )
1218                                         )
1219                                         )
1219                                         )
1220                                         )
1221                                         )
1222                                         )
1223                                         )
1224                                         )
1225                                         )
1226                                         )
1227                                         )
1228                                         )
1229                                         )
1229                                         )
1230                                         )
1231                                         )
1232                                         )
1233                                         )
1234                                         )
1235                                         )
1236                                         )
1237                                         )
1238                                         )
1238                                         )
1239                                         )
1239                                         )
1240                                         )
1241                                         )
1242                                         )
1243                                         )
1244                                         )
1245                                         )
1246                                         )
1247                                         )
1248                                         )
1248                                         )
1249                                         )
1249                                         )
1250                                         )
1251                                         )
1252                                         )
1253                                         )
1254                                         )
1255                                         )
1256                                         )
1257                                         )
1258                                         )
1258                                         )
1259                                         )
1259                                         )
1260                                         )
1261                                         )
1262                                         )
1263                                         )
1264                                         )
1265                                         )
1266                                         )
1267                                         )
1268                                         )
1268                                         )
1269                                         )
1269                                         )
1270                                         )
1271                                         )
1272                                         )
1273                                         )
1274                                         )
1275                                         )
1276                                         )
1277                                         )
1278                                         )
1278                                         )
1279                                         )
1279                                         )
1280                                         )
1281                                         )
1282                                         )
1283                                         )
1284                                         )
1285                                         )
1286                                         )
1287                                         )
1288                                         )
1288                                         )
1289                                         )
1289                                         )
1290                                         )
1291                                         )
1292                                         )
1293                                         )
1294                                         )
1295                                         )
1295                                         )
1296                                         )
1297                                         )
1298                                         )
1299                                         )
1299                                         )
1300                                         )
1301                                         )
1302                                         )
1303                                         )
1304                                         )
1305                                         )
1306                                         )
1307                                         )
1308                                         )
1309                                         )
1309                                         )
1310                                         )
1311                                         )
1312                                         )
1313                                         )
1314                                         )
1315                                         )
1316                                         )
1317                                         )
1318                                         )
1318                                         )
1319                                         )
1319                                         )
1320                                         )
1321                                         )
1322                                         )
1323                                         )
1324                                         )
1325                                         )
1326                                         )
1327                                         )
1328                                         )
1328                                         )
1329                                         )
1329                                         )
1330                                         )
1331                                         )
1332                                         )
1333                                         )
1334                                         )
1335                                         )
1336                                         )
1337                                         )
1338                                         )
1338                                         )
1339                                         )
1339                                         )
1340                                         )
1341                                         )
1342                                         )
1343                                         )
1344                                         )
1345                                         )
1346                                         )
1347                                         )
1348                                         )
1348                                         )
1349                                         )
1349                                         )
1350                                         )
1351                                         )
1352                                         )
1353                                         )
1354                                         )
1355                                         )
1356                                         )
1357                                         )
1358                                         )
1358                                         )
1359                                         )
1359                                         )
1360                                         )
1361                                         )
1362                                         )
1363                                         )
1364                                         )
1365                                         )
1366                                         )
1367                                         )
1368                                         )
1368                                         )
1369                                         )
1369                                         )
1370                                         )
1371                                         )
1372                                         )
1373                                         )
1374                                         )
1375                                         )
1376                                         )
1377                                         )
1378                                         )
1378                                         )
1379                                         )
1379                                         )
1380                                         )
1381                                         )
1382                                         )
1383                                         )
1384                                         )
1385                                         )
1386                                         )
1387                                         )
1388                                         )
1388                                         )
1389                                         )
1389                                         )
1390                                         )
1391                                         )
1392                                         )
1393                                         )
1394                                         )
1395                                         )
1396                                         )
1397                                         )
1398                                         )
1398                                         )
1399                                         )
1399                                         )
1400                                         )
1401                                         )
1402                                         )
1403                                         )
1404                                         )
1405                                         )
1406                                         )
1407                                         )
1408                                         )
1409                                         )
1409                                         )
1410                                         )
1411                                         )
1412                                         )
1413                                         )
1414                                         )
1415                                         )
1416                                         )
1417                                         )
1418                                         )
1418                                         )
1419                                         )
1419                                         )
1420                                         )
1421                                         )
1422                                         )
1423                                         )
1424                                         )
1425                                         )
1426                                         )
1427                                         )
1428                                         )
1428                                         )
1429                                         )
1429                                         )
1430                                         )
1431                                         )
1432                                         )
1433                                         )
1434                                         )
1435                                         )
1436                                         )
1437                                         )
1438                                         )
1438                                         )
1439                                         )
1439                                         )
1440                                         )
1441                                         )
1442                                         )
1443                                         )
1444                                         )
1445                                         )
1446                                         )
1447                                         )
1448                                         )
1448                                         )
1449                                         )
1449                                         )
1450                                         )
1451                                         )
1452                                         )
1453                                         )
1454                                         )
1455                                         )
1456                                         )
1457                                         )
1458                                         )
1458                                         )
1459                                         )
1459                                         )
1460                                         )
1461                                         )
1462                                         )
1463                                         )
1464                                         )
1465                                         )
1466                                         )
1467                                         )
1468                                         )
1468                                         )
1469                                         )
1469                                         )
1470                                         )
1471                                         )
1472                                         )
1473                                         )
1474                                         )
1475                                         )
1476                                         )
1477                                         )
1478                                         )
1478                                         )
1479                                         )
1479                                         )
1480                                         )
1481                                         )
1482                                         )
1483                                         )
1484                                         )
1485                                         )
1486                                         )
1487                                         )
1488                                         )
1488                                         )
1489                                         )
1489                                         )
1490                                         )
1491                                         )
1492                                         )
1493                                         )
1494                                         )
1495                                         )
1496                                         )
1497                                         )
1498                                         )
1498                                         )
1499                                         )
1499                                         )
1500                                         )
1501                                         )
1502                                         )
1503                                         )
1504                                         )
1505                                         )
1506                                         )
1507                                         )
1508                                         )
1509                                         )
1509                                         )
1510                                         )
1511                                         )
1512                                         )
1513                                         )
1514                                         )
1515                                         )
1516                                         )
1517                                         )
1518                                         )
1518                                         )
1519                                         )
1519                                         )
1520                                         )
1521                                         )
1522                                         )
1523                                         )
1524                                         )
1525                                         )
1526                                         )
1527                                         )
1528                                         )
1528                                         )
1529                                         )
1529                                         )
1530                                         )
1531                                         )
1532                                         )
1533                                         )
1534                                         )
1535                                         )
1536                                         )
1537                                         )
1538                                         )
1538                                         )
1539                                         )
1539                                         )
1540                                         )
1541                                         )
1542                                         )
1543                                         )
1544                                         )
1545                                         )
1546                                         )
1547                                         )
1548                                         )
1548                                         )
1549                                         )
1549                                         )
1550                                         )
1551                                         )
1552                                         )
1553                                         )
1554                                         )
1555                                         )
1556                                         )
1557                                         )
1558                                         )
1558                                         )
1559                                         )
1559                                         )
1560                                         )
1561                                         )
1562                                         )
1563                                         )
1564                                         )
1565                                         )
1566                                         )
1567                                         )
1568                                         )
1568                                         )
1569                                         )
1569                                         )
1570                                         )
1571                                         )
1572                                         )
1573                                         )
1574                                         )
1575                                         )
1576                                         )
1577                                         )
1578                                         )
1578                                         )
1579                                         )
1579                                         )
1580                                         )
1581                                         )
1582                                         )
1583                                         )
1584                                         )
1585                                         )
1586                                         )
1587                                         )
1588                                         )
1588                                         )
1589                                         )
1589                                         )
1590                                         )
1591                                         )
1592                                         )
1593                                         )
1594                                         )
1595                                         )
1596                                         )
1597                                         )
1598                                         )
1598                                         )
1599                                         )
1599                                         )
1600                                         )
1601                                         )
1602                                         )
1603                                         )
1604                                         )
1605                                         )
1606                                         )
1607                                         )
1608                                         )
1609                                         )
1609                                         )
1610                                         )
1611                                         )
1612                                         )
1613                                         )
1614                                         )
1615                                         )
1616                                         )
1617                                         )
1618                                         )
1618                                         )
1619                                         )
1619                                         )
1620                                         )
1621                                         )
1622                                         )
1623                                         )
1624                                         )
1625                                         )
1626                                         )
1627                                         )
1628                                         )
1628                                         )
1629                                         )
1629                                         )
1630                                         )
1631                                         )
1632                                         )
1633                                         )
1634                                         )
1635                                         )
1636                                         )
1637                                         )
1638                                         )
1638                                         )
1639                                         )
1639                                         )
1640                                         )
1641                                         )
1642                                         )
1643                                         )
1644                                         )
1645                                         )
1646                                         )
1647                                         )
1648                                         )
1648                                         )
1649                                         )
1649                                         )
1650                                         )
1651                                         )
1652                                         )
1653                                         )
1654                                         )
1655                                         )
1656                                         )
1657                                         )
1658                                         )
1658                                         )
1659                                         )
1659                                         )
1660                                         )
1661                                         )
1662                                         )
1663                                         )
1664                                         )
1665                                         )
1666                                         )
1667                                         )
1668                                         )
1668                                         )
1669                                         )
1669                                         )
1670                                         )
1671                                         )
1672                                         )
1673                                         )
1674                                         )
1675                                         )
1676                                         )
1677                                         )
1678                                         )
1678                                         )
1679                                         )
1679                                         )
1680                                         )
1681                                         )
1682                                         )
1683                                         )
1684                                         )
1685                                         )
1686                                         )
1687                                         )
1688                                         )
1688                                         )
1689                                         )
1689                                         )
1690                                         )
1691                                         )
1692                                         )
1693                                         )
1694                                         )
1695                                         )
1696                                         )
1697                                         )
1698                                         )
1698                                         )
1699                                         )
1699                                         )
1700                                         )
1701                                         )
1702                                         )
1703                                         )
1704                                         )
1705                                         )
1706                                         )
1707                                         )
1708                                         )
1709                                         )
1709                                         )
1710                                         )
1711                                         )
1712                                         )
1713                                         )
1714                                         )
1715                                         )
1716                                         )
1717                                         )
1718                                         )
1718                                         )
1719                                         )
1719                                         )
1720                                         )
1721                                         )
1722                                         )
1723                                         )
1724                                         )
1725                                         )
1726                                         )
1727                                         )
1728                                         )
1728                                         )
1729                                         )
1729                                         )
1730                                         )
1731                                         )
1732                                         )
1733                                         )
1734                                         )
1735                                         )
1736                                         )
1737                                         )
1738                                         )
1738                                         )
1739                                         )
1739                                         )
1740                                         )
1741                                         )
1742                                         )
1743                                         )
1744                                         )
1745                                         )
1746                                         )
1747                                         )
1748                                         )
1748                                         )
1749                                         )
1749                                         )
1750                                         )
1751                                         )
1752                                         )
1753                                         )
1754                                         )
1755                                         )
1756                                         )
1757                                         )
1758                                         )
1758                                         )
1759                                         )
1759                                         )
1760                                         )
1761                                         )
1762                                         )
1763                                         )
1764                                         )
1765                                         )
1766                                         )
1767                                         )
1768                                         )
1768                                         )
1769                                         )
1769                                         )
1770                                         )
1771                                         )
1772                                         )
1773                                         )
1774                                         )
1775                                         )
1776                                         )
1777                                         )
1778                                         )
1778                                         )
1779                                         )
1779                                         )
1780                                         )
1781                                         )
1782                                         )
1783                                         )
1784                                         )
1785                                         )
1786                                         )
1787                                         )
1788                                         )
1788                                         )
1789                                         )
1789                                         )
1790                                         )
1791                                         )
1792                                         )
1793                                         )
1794                                         )
1795                                         )
1796                                         )
1797                                         )
1798                                         )
1798                                         )
1799                                         )
1799                                         )
1800                                         )
1801                                         )
1802                                         )
1803                                         )
1804                                         )
1805                                         )
1806                                         )
1807                                         )
1808                                         )
1809                                         )
1809                                         )
1810                                         )
1811                                         )
1812                                         )
1813                                         )
1814                                         )
1815                                         )
1816                                         )
1817                                         )
1818                                         )
1818                                         )
1819                                         )
1819                                         )
1820                                         )
1821                                         )
1822                                         )
1823                                         )
1824                                         )
1825                                         )
1826                                         )
1827                                         )
1828                                         )
1828                                         )
1829                                         )
1829                                         )
1830                                         )
1831                                         )
1832                                         )
1833                                         )
1834                                         )
1835                                         )
1836                                         )
1837                                         )
1838                                         )
1838                                         )
1839                                         )
1839                                         )
184
```

```

5 | library(shinyCSSloaders)
6 |
7 | Essai221Page <- function() {
8 |   tagList(
9 |     makePage(
10 |       "Lemna sp. Growth Inhibition Test",
11 |       "(Essai 221)",
12 |       fluidRow(
13 |         column(4,
14 |          textInput(inputId = "chemical", "Nom du produit
15 |                         chimique testé", value = "Produit chimique"),
16 |           selectInput("conc_unit221", "Unité de Concentration",
17 |                         choices = c("g/L", "mg/L", "microg/L", "ng/L")),
18 |           fileInput('datafile_ESSAI221', '', accept = c('.csv'))
19 |         ),
20 |           tags$br(),
21 |           selectInput('model_ESSAI221', 'Sélectionner le modèle
22 |                         de régression',
23 |                         choices = c('log-logistic 2 paramètres'
24 |                                     = '112',
25 |                                     'log-logistic 3 paramètres'
26 |                                     = '113',
27 |                                     'log-logistic 4 paramètres'
28 |                                     = '114'),
29 |           selected = '112'),
30 |           numericInput(inputId = "ecx_ESSAI221", "Déterminer la
31 |                         concentration efficace X%", value = 50, min = 0,
32 |                         max = 100),
33 |           selectInput('test_method_ESSAI221', 'Sélectionner la
34 |                         méthode de test d\'hypothèse',
35 |                         choices = c('Test de Dunnett' = 'Dunnett
36 |                                     '),
37 |           selected = 'Dunnett'),
38 |           actionButton("buttonRunStep221", "Exécuter", icon = icon("check"),
39 |                         width = "100px")
40 |         ),
41 |         column(8,
42 |           mainPanel(
43 |             width = 12,
44 |             DT::dataTableOutput('rawdata'),
45 |             br(),
46 |             tableOutput('drc_result'),
47 |             withSpinner(plotOutput('drc_plot', width = "100%",
48 |                                     height = "850px"), type = 4, color = '#9e0093'),
49 |             br(),
50 |             withSpinner(verbatimTextOutput("test_result")),
51 |             br()
52 |           )
53 |         ),
54 |         tabPanel(
55 |           "Télécharger le rapport",
56 |           fixedRow(
57 |             column(

```

```

44     8,
45     br(),
46     HTML("<font face=verdana size=5 color=#9e0093><b>Télé
47     charger le rapport d'analyse</b></font>"),
48     br(), br(), br(),
49     radioButtons("format", "Sélectionner le format du
50     rapport", c('Word'), inline = TRUE),
51     downloadButton("DownloadReport", "Télécharger le
52     rapport", icon = icon("fas fa-download"), style = ,
53     background-color:#9e0093; color:#000000; border-
54     color:#9d9d9d; )
55   )
  )
)
)
)
)
}

```

Page essai standardisé 222

```

1 library(shiny)
2 library(DT)
3 library(nlme)
4 library(shinyCSSloaders)
5
6 Essai222Page <- function() {
7   tagList(
8     fluidPage(
9       titlePanel("Earthworm Reproduction Test (Eisenia fetida/
10      Eisenia andrei)"),
11       fluidRow(
12         column(4,
13          textInput(inputId = "chemical", "Nom du produit
14             chimique testé", value = "Produit chimique"),
15           selectInput("conc_unit", "Unité de Concentration",
16             choices = c("g/kg", "mg/kg", "microg/kg", "ng/kg")
17             ),
18           fileInput('datafile_ESSAI222', 'Sélectionner un
19             fichier', accept = c('.csv')),
20           radioButtons('model_ESSAI222', 'Sélectionner le modèle
21             de régression',
22             choices = c('Linéaire' = 'lm', 'Non-liné
23               aire' = 'nls', 'Mixte' = 'lme'),
24             selected = 'lm'),
25           numericInput(inputId = "ecx_ESSAI222", "Déterminer la
26             concentration efficace X%", value = 50, min = 0,
27             max = 100),
28           radioButtons('test_method_ESSAI222', 'Sélectionner la
29             méthode de test \ 'hypothèse',
30             choices = c('ANOVA' = 'anova'), selected
31             = 'anova'),
32           actionButton("buttonRunStep1", "Exécuter", icon =
33             icon("check"), width = "100px")
  )
  )
)
)
)
}

```

```

22  ),
23  column(8,
24  mainPanel(
25  width=12,
26  DT::dataTableOutput('rawdata'),
27  br(),
28  tableOutput('drc_result'),
29  withSpinner(plotOutput('drc_plot', width="100%", height="850px"), type=4, color="#9e0093'),
30  br(),
31  withSpinner(verbatimTextOutput("test_result")),
32  br()
33  )
34  ),
35  tabPanel(
36  "Télécharger le rapport",
37  fixedRow(
38  column(
39  8,
40  br(),
41  HTML("<font face=verdana size=5 color=#9e0093><b>Télé
42  charger le rapport d'analyse</b></font>"),
43  br(), br(), br(),
44  radioButtons("format", "Sélectionner le format du
45  rapport", c('Word'), inline = TRUE),
46  downloadButton("DownloadReport", "Télécharger le
47  rapport", icon = icon("fas fa-download"), style = '
48  background-color:#9e0093; color:#000000; border-
49  color:#9d9d9d;')
50  )
51  )
52  )
53  )
54  )
55  )
56  )
57  )
58  )
59  )
50 }

```

Page essai standardisé 223

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(multcomp)
5 library(shinyCSSloaders)
6
7 Essai223Page <- function() {
8   tagList(
9     fluidPage(
10       titlePanel("Avian Acute Oral Toxicity Test"),
11       fluidRow(
12         column(4,
13            textInput(inputId = "chemical", "Nom du produit
14             chimique testé", value = "Produit chimique"),
15             selectInput("conc_unit", "Unité de Dose", choices = c
16             )
17           )
18         )
19       )
20     )
21   )
22 }

```

```

15         ("g/kg", "mg/kg", "microg/kg", "ng/kg"))),
16     fileInput('datafile_ESSAI223', 'Sélectionner un
17         fichier', accept = c('.csv')),
18     radioButtons('model_ESSAI223', 'Sélectionner le modèle
19         de régression',
20             choices = c('Probit' = 'probit', 'Logit'
21             = 'logit'),
22             selected = 'probit'),
23     numericInput(inputId = "ecx_ESSAI223", "Déterminer la
24         dose efficace X%", value = 50, min = 0, max =
25         100),
26     radioButtons('test_method_ESSAI223', 'Sélectionner la
27         méthode de test d\'hypothèse',
28         choices = c('Test de Dunnett' = 'Dunnett
29             '),
30             selected = 'Dunnett'),
31     actionButton("buttonRunStep1", "Exécuter", icon = u
32         icon("check"), width = "100px")
33     ),
34     column(8,
35         mainPanel(
36             width = 12,
37             DT::dataTableOutput('rawdata'),
38             br(),
39             tableOutput('drc_result'),
40             withSpinner(plotOutput('drc_plot', width = "100%", u
41                 height = "850px"), type = 4, color = '#9e0093'),
42                 br(),
43                 withSpinner(verbatimTextOutput("test_result")),
44                 br()
45             )
46         ),
47         tabPanel(
48             "Télécharger le rapport",
49             fixedRow(
50                 column(
51                     8,
52                     br(),
53                     HTML("<font face=verdana size=5 color=#9e0093><b>Télé
54                         charger le rapport d'analyse</b></font>"),
55                     br(), br(), br(),
56                     radioButtons("format", "Sélectionner le format du
57                         rapport", c('Word'), inline = TRUE),
58                     downloadButton("DownloadReport", "Télécharger le
59                         rapport", icon = icon("fas fa-download"), style = ,
60                         background-color:#9e0093; color:#000000; border-
61                         color:#9d9d9d; )
62                 )
63             )
64         )
65     )
66 )
67 )
68 ))
69 }

```

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(multcomp)
5 library(shinyCSSloaders)
6
7 Essai225Page <- function() {
8   tagList(
9     fluidPage(
10       titlePanel("Essai 225"),
11       fluidRow(
12         column(4,
13          textInput(inputId = "chemical", "Nom du produit chimique testé", value = "Produit chimique"),
14           selectInput("conc_unit", "Unité de Concentration", choices = c("g/kg", "mg/kg", "microg/kg", "ng/kg")),
15           fileInput('datafile_ESSAI225', 'Sélectionner un fichier', accept = c('.csv')),
16           radioButtons('model_ESSAI225', 'Sélectionner le modèle de régression',
17                         choices = c('Logistique' = 'logit'),
18                         selected = 'logit'),
19           numericInput(inputId = "ecx_ESSAI225", "Déterminer la concentration efficace X%", value = 50, min = 0, max = 100),
20           radioButtons('test_method_ESSAI225', 'Sélectionner la méthode de test d\'hypothèse',
21                         choices = c('Test de Dunnett' = 'Dunnett'),
22                         selected = 'Dunnett'),
23           actionButton("buttonRunStep1", "Exécuter", icon = icon("check"), width = "100px")
24         ),
25         column(8,
26           mainPanel(
27             DT::dataTableOutput('rawdata'),
28             br(),
29             tableOutput('drc_result'),
30             withSpinner(plotOutput('drc_plot', width = "100%", height = "850px"), type = 4, color = "#9e0093"),
31             br(),
32             withSpinner(verbatimTextOutput("test_result")),
33             br()
34           )
35         ),
36         tabPanel(
37           "Télécharger le rapport",
38           fixedRow(
39             column(
40               8,
41               8,
42               8,
43               8,
44               8,
45               8,
46               8,
47               8,
48               8,
49               8,
50               8,
51               8,
52               8,
53               8,
54               8,
55               8,
56               8,
57               8,
58               8,
59               8,
60               8,
61               8,
62               8,
63               8,
64               8,
65               8,
66               8,
67               8,
68               8,
69               8,
70               8,
71               8,
72               8,
73               8,
74               8,
75               8,
76               8,
77               8,
78               8,
79               8,
80               8,
81               8,
82               8,
83               8,
84               8,
85               8,
86               8,
87               8,
88               8,
89               8,
90               8,
91               8,
92               8,
93               8,
94               8,
95               8,
96               8,
97               8,
98               8,
99               8,
100              8,
101              8,
102              8,
103              8,
104              8,
105              8,
106              8,
107              8,
108              8,
109              8,
110              8,
111              8,
112              8,
113              8,
114              8,
115              8,
116              8,
117              8,
118              8,
119              8,
120              8,
121              8,
122              8,
123              8,
124              8,
125              8,
126              8,
127              8,
128              8,
129              8,
130              8,
131              8,
132              8,
133              8,
134              8,
135              8,
136              8,
137              8,
138              8,
139              8,
140              8,
141              8,
142              8,
143              8,
144              8,
145              8,
146              8,
147              8,
148              8,
149              8,
150              8,
151              8,
152              8,
153              8,
154              8,
155              8,
156              8,
157              8,
158              8,
159              8,
160              8,
161              8,
162              8,
163              8,
164              8,
165              8,
166              8,
167              8,
168              8,
169              8,
170              8,
171              8,
172              8,
173              8,
174              8,
175              8,
176              8,
177              8,
178              8,
179              8,
180              8,
181              8,
182              8,
183              8,
184              8,
185              8,
186              8,
187              8,
188              8,
189              8,
190              8,
191              8,
192              8,
193              8,
194              8,
195              8,
196              8,
197              8,
198              8,
199              8,
200              8,
201              8,
202              8,
203              8,
204              8,
205              8,
206              8,
207              8,
208              8,
209              8,
210              8,
211              8,
212              8,
213              8,
214              8,
215              8,
216              8,
217              8,
218              8,
219              8,
220              8,
221              8,
222              8,
223              8,
224              8,
225              8,
226              8,
227              8,
228              8,
229              8,
230              8,
231              8,
232              8,
233              8,
234              8,
235              8,
236              8,
237              8,
238              8,
239              8,
240              8,
241              8,
242              8,
243              8,
244              8,
245              8,
246              8,
247              8,
248              8,
249              8,
250              8,
251              8,
252              8,
253              8,
254              8,
255              8,
256              8,
257              8,
258              8,
259              8,
260              8,
261              8,
262              8,
263              8,
264              8,
265              8,
266              8,
267              8,
268              8,
269              8,
270              8,
271              8,
272              8,
273              8,
274              8,
275              8,
276              8,
277              8,
278              8,
279              8,
280              8,
281              8,
282              8,
283              8,
284              8,
285              8,
286              8,
287              8,
288              8,
289              8,
290              8,
291              8,
292              8,
293              8,
294              8,
295              8,
296              8,
297              8,
298              8,
299              8,
300              8,
301              8,
302              8,
303              8,
304              8,
305              8,
306              8,
307              8,
308              8,
309              8,
310              8,
311              8,
312              8,
313              8,
314              8,
315              8,
316              8,
317              8,
318              8,
319              8,
320              8,
321              8,
322              8,
323              8,
324              8,
325              8,
326              8,
327              8,
328              8,
329              8,
330              8,
331              8,
332              8,
333              8,
334              8,
335              8,
336              8,
337              8,
338              8,
339              8,
340              8,
341              8,
342              8,
343              8,
344              8,
345              8,
346              8,
347              8,
348              8,
349              8,
350              8,
351              8,
352              8,
353              8,
354              8,
355              8,
356              8,
357              8,
358              8,
359              8,
360              8,
361              8,
362              8,
363              8,
364              8,
365              8,
366              8,
367              8,
368              8,
369              8,
370              8,
371              8,
372              8,
373              8,
374              8,
375              8,
376              8,
377              8,
378              8,
379              8,
380              8,
381              8,
382              8,
383              8,
384              8,
385              8,
386              8,
387              8,
388              8,
389              8,
390              8,
391              8,
392              8,
393              8,
394              8,
395              8,
396              8,
397              8,
398              8,
399              8,
400              8,
401              8,
402              8,
403              8,
404              8,
405              8,
406              8,
407              8,
408              8,
409              8,
410              8,
411              8,
412              8,
413              8,
414              8,
415              8,
416              8,
417              8,
418              8,
419              8,
420              8,
421              8,
422              8,
423              8,
424              8,
425              8,
426              8,
427              8,
428              8,
429              8,
430              8,
431              8,
432              8,
433              8,
434              8,
435              8,
436              8,
437              8,
438              8,
439              8,
440              8,
441              8,
442              8,
443              8,
444              8,
445              8,
446              8,
447              8,
448              8,
449              8,
450              8,
451              8,
452              8,
453              8,
454              8,
455              8,
456              8,
457              8,
458              8,
459              8,
460              8,
461              8,
462              8,
463              8,
464              8,
465              8,
466              8,
467              8,
468              8,
469              8,
470              8,
471              8,
472              8,
473              8,
474              8,
475              8,
476              8,
477              8,
478              8,
479              8,
480              8,
481              8,
482              8,
483              8,
484              8,
485              8,
486              8,
487              8,
488              8,
489              8,
490              8,
491              8,
492              8,
493              8,
494              8,
495              8,
496              8,
497              8,
498              8,
499              8,
500              8,
501              8,
502              8,
503              8,
504              8,
505              8,
506              8,
507              8,
508              8,
509              8,
510              8,
511              8,
512              8,
513              8,
514              8,
515              8,
516              8,
517              8,
518              8,
519              8,
520              8,
521              8,
522              8,
523              8,
524              8,
525              8,
526              8,
527              8,
528              8,
529              8,
530              8,
531              8,
532              8,
533              8,
534              8,
535              8,
536              8,
537              8,
538              8,
539              8,
540              8,
541              8,
542              8,
543              8,
544              8,
545              8,
546              8,
547              8,
548              8,
549              8,
550              8,
551              8,
552              8,
553              8,
554              8,
555              8,
556              8,
557              8,
558              8,
559              8,
560              8,
561              8,
562              8,
563              8,
564              8,
565              8,
566              8,
567              8,
568              8,
569              8,
570              8,
571              8,
572              8,
573              8,
574              8,
575              8,
576              8,
577              8,
578              8,
579              8,
580              8,
581              8,
582              8,
583              8,
584              8,
585              8,
586              8,
587              8,
588              8,
589              8,
590              8,
591              8,
592              8,
593              8,
594              8,
595              8,
596              8,
597              8,
598              8,
599              8,
600              8,
601              8,
602              8,
603              8,
604              8,
605              8,
606              8,
607              8,
608              8,
609              8,
610              8,
611              8,
612              8,
613              8,
614              8,
615              8,
616              8,
617              8,
618              8,
619              8,
620              8,
621              8,
622              8,
623              8,
624              8,
625              8,
626              8,
627              8,
628              8,
629              8,
630              8,
631              8,
632              8,
633              8,
634              8,
635              8,
636              8,
637              8,
638              8,
639              8,
640              8,
641              8,
642              8,
643              8,
644              8,
645              8,
646              8,
647              8,
648              8,
649              8,
650              8,
651              8,
652              8,
653              8,
654              8,
655              8,
656              8,
657              8,
658              8,
659              8,
660              8,
661              8,
662              8,
663              8,
664              8,
665              8,
666              8,
667              8,
668              8,
669              8,
670              8,
671              8,
672              8,
673              8,
674              8,
675              8,
676              8,
677              8,
678              8,
679              8,
680              8,
681              8,
682              8,
683              8,
684              8,
685              8,
686              8,
687              8,
688              8,
689              8,
690              8,
691              8,
692              8,
693              8,
694              8,
695              8,
696              8,
697              8,
698              8,
699              8,
700              8,
701              8,
702              8,
703              8,
704              8,
705              8,
706              8,
707              8,
708              8,
709              8,
710              8,
711              8,
712              8,
713              8,
714              8,
715              8,
716              8,
717              8,
718              8,
719              8,
720              8,
721              8,
722              8,
723              8,
724              8,
725              8,
726              8,
727              8,
728              8,
729              8,
730              8,
731              8,
732              8,
733              8,
734              8,
735              8,
736              8,
737              8,
738              8,
739              8,
740              8,
741              8,
742              8,
743              8,
744              8,
745              8,
746              8,
747              8,
748              8,
749              8,
750              8,
751              8,
752              8,
753              8,
754              8,
755              8,
756              8,
757              8,
758              8,
759              8,
760              8,
761              8,
762              8,
763              8,
764              8,
765              8,
766              8,
767              8,
768              8,
769              8,
770              8,
771              8,
772              8,
773              8,
774              8,
775              8,
776              8,
777              8,
778              8,
779              8,
780              8,
781              8,
782              8,
783              8,
784              8,
785              8,
786              8,
787              8,
788              8,
789              8,
790              8,
791              8,
792              8,
793              8,
794              8,
795              8,
796              8,
797              8,
798              8,
799              8,
800              8,
801              8,
802              8,
803              8,
804              8,
805              8,
806              8,
807              8,
808              8,
809              8,
810              8,
811              8,
812              8,
813              8,
814              8,
815              8,
816              8,
817              8,
818              8,
819              8,
820              8,
821              8,
822              8,
823              8,
824              8,
825              8,
826              8,
827              8,
828              8,
829              8,
830              8,
831              8,
832              8,
833              8,
834              8,
835              8,
836              8,
837              8,
838              8,
839              8,
840              8,
841              8,
842              8,
843              8,
844              8,
845              8,
846              8,
847              8,
848              8,
849              8,
850              8,
851              8,
852              8,
853              8,
854              8,
855              8,
856              8,
857              8,
858              8,
859              8,
860              8,
861              8,
862              8,
863              8,
864              8,
865              8,
866              8,
867              8,
868              8,
869              8,
870              8,
871              8,
872              8,
873              8,
874              8,
875              8,
876              8,
877              8,
878              8,
879              8,
880              8,
881              8,
882              8,
883              8,
884              8,
885              8,
886              8,
887              8,
888              8,
889              8,
890              8,
891              8,
892              8,
893              8,
894              8,
895              8,
896              8,
897              8,
898              8,
899              8,
900              8,
901              8,
902              8,
903              8,
904              8,
905              8,
906              8,
907              8,
908              8,
909              8,
910              8,
911              8,
912              8,
913              8,
914              8,
915              8,
916              8,
917              8,
918              8,
919              8,
920              8,
921              8,
922              8,
923              8,
924              8,
925              8,
926              8,
927              8,
928              8,
929              8,
930              8,
931              8,
932              8,
933              8,
934              8,
935              8,
936              8,
937              8,
938              8,
939              8,
940              8,
941              8,
942              8,
943              8,
944              8,
945              8,
946              8,
947              8,
948              8,
949              8,
950              8,
951              8,
952              8,
953              8,
954              8,
955              8,
956              8,
957              8,
958              8,
959              8,
960              8,
961              8,
962              8,
963              8,
964              8,
965              8,
966              8,
967              8,
968              8,
969              8,
970              8,
971              8,
972              8,
973              8,
974              8,
975              8,
976              8,
977              8,
978              8,
979              8,
980              8,
981              8,
982              8,
983              8,
984              8,
985              8,
986              8,
987              8,
988              8,
989              8,
990              8,
991              8,
992              8,
993              8,
994              8,
995              8,
996              8,
997              8,
998              8,
999              8,
1000             8,
1001             8,
1002             8,
1003             8,
1004             8,
1005             8,
1006             8,
1007             8,
1008             8,
1009             8,
1010             8,
1011             8,
1012             8,
1013             8,
1014             8,
1015             8,
1016             8,
1017             8,
1018             8,
1019             8,
1020             8,
1021             8,
1022             8,
1023             8,
1024             8,
1025             8,
1026             8,
1027             8,
1028             8,
1029             8,
1030             8,
1031             8,
1032             8,
1033             8,
1034             8,
1035             8,
1036             8,
1037             8,
1038             8,
1039             8,
1040             8,
1041             8,
1042             8,
1043             8,
1044             8,
1045             8,
1046             8,
1047             8,
1048             8,
1049             8,
1050             8,
1051             8,
1052             8,
1053             8,
1054             8,
1055             8,
1056             8,
1057             8,
1058             8,
1059             8,
1060             8,
1061             8,
1062             8,
1063             8,
1064             8,
1065             8,
1066             8,
1067             8,
1068             8,
1069             8,
1070             8,
1071             8,
1072             8,
1073             8,
1074             8,
1075             8,
1076             8,
1077             8,
1078             8,
1079             8,
1080             8,
1081             8,
1082             8,
1083             8,
1084             8,
1085             8,
1086             8,
1087             8,
1088             8,
1089             8,
1090             8,
1091             8,
1092             8,
1093             8,
1094             8,
1095             8,
1096             8,
1097             8,
1098             8,
1099             8,
1100             8,
1101             8,
1102             8,
1103             8,
1104             8,
1105             8,
1106             8,
1107             8,
1108             8,
1109             8,
1110             8,
1111             8,
1112             8,
1113             8,
1114             8,
1115             8,
1116             8,
1117             8,
1118             8,
1119             8,
1120             8,
1121             8,
1122             8,
1123             8,
1124             8,
1125             8,
1126             8,
1127             8,
1128             8,
1129             8,
1130             8,
1131             8,
1132             8,
1133             8,
1134             8,
1135             8,
1136             8,
1137             8,
1138             8,
1139             8,
1140             8,
1141             8,
1142             8,
1143             8,
1144             8,
1145             8,
1146             8,
1147             8,
1148             8,
1149             8,
1150             8,
1151             8,
1152             8,
1153             8,
1154             8,
1155             8,
1156             8,
1157             8,
1158             8,
1159             8,
1160             8,
1161             8,
1162             8,
1163             8,
1164             8,
1165             8,
1166             8,
1167             8,
1168             8,
1169             8,
1170             8,
1171             8,
1172             8,
1173             8,
1174             8,
1175             8,
1176             8,
1177             8,
1178             8,
1179             8,
1180             8,
1181             8,
1182             8,
1183             8,
1184             8,
1185             8,
1186             8,
1187             8,
1188             8,
1189             8,
1190             8,
1191             8,
1192             8,
1193             8,
1194             8,
1195             8,
1196             8,
1197             8,
1198             8,
1199             8,
1200             8,
1201             8,
1202             8,
1203             8,
1204             8,
1205             8,
1206             8,
1207             8,
1208             8,
1209             8,
1210             8,
1211             8,
1212             8,
1213             8,
1214             8,
1215             8,
1216             8,
1217             8,
1218             8,
1219             8,
1220             8,
1221             8,
1222             8,
1223             8,
1224             8,
1225             8,
1226             8,
1227             8,
1228             8,
1229             8,
1230             8,
1231             8,
1232             8,
1233             8,
1234             8,
1235             8,
1236             8,
1237             8,
1238             8,
1239             8,
1240             8,
1241             8,
1242             8,
1243             8,
1244             8,
1245             8,
1246             8,
1247             8,
1248             8,
1249             8,
1250             8,
1251             8,
1252             8,
1253             8,
1254             8,
1255             8,
1256             8,
1257             8,
1258             8,
1259             8,
1260             8,
1261             8,
1262             8,
1263             8,
1264             8,
1265             8,
1266             8,
1267             8,
1268             8,
1269             8,
1270             8,
1271             8,
1272             8,
1273             8,
1274             8,
1275             8,
1276             8,
1277             8,
1278             8,
1279             8,
1280             8,
1281             8,
1282             8,
1283             8,
1284             8,
1285             8,
1286             8,
1287             8,
1288             8,
1289             8,
1290             8,
1291             8,
1292             8,
1293             8,
1294             8,
1295             8,
1296             8,
1297             8,
1298             8,
1299             8,
1300             8,
1301             8,
1302             8,
1303             8,
1304             8,
1305             8,
1306             8,
1307             8,
1308             8,
1309             8,
1310             8,
1311             8,
1312             8,
1313             8,
1314             8,
1315             8,
1316             8,
1317             8,
1318             8,
1319             8,
1320             8,
1321             8,
1322             8,
1323             8,
1324             8,
1325             8,
1326             8,
1327             8,
1328             8,
1329             8,
1330             8,
1331             8,
1332             8,
1333             8,
1334             8,
1335             8,
1336             8,
1337             8,
1338             8,
1339             8,
1340             8,
1341             8,
1342             8,
1343             8,
1344             8,
1345             8,
1346             8,
1347             8,
1348             8,
1349             8,
1350             8,
1351             8,
1352             8,
1353             8,
1354             8,
1355             8,
1356             8,
1357             8,
1358             8,
1359             8,
1360             8,
1361             8,
1362             8,
1363             8,
1364             8,
1365             8,
1366             8,
1367             8,
1368             8,
1369             8,
1370             8,
1371             8,
1372             8,
1373             8,
1374             8,
1375             8,
1376             8,
1377             8,
1378             8,
1379             8,
1380             8,
1381             8,
1382             8,
1383             8,
1384             8,
1385             8,
1386             8,
1387             8,
1388             8,
1389             8,
1390             8,
1391             8,
1392             8,
1393             8,
1394             8,
1395             8,
1396             8,
1397             8,
1398             8,
1399             8,
1400             8,
1401             8,
1402             8,
1403             8,
1404             8,
1405             8,
1406             8,
1407             8,
1408             8,
1409             8,
1410             8,
1411             8,
1412             8,
1413             8,
1414             8,
1415             8,
1416             8,
1417             8,
1418             8,
1419             8,
1420             8,
1421             8,
1422             8,
1423             8,
1424             8,
1425             8,
1426             8,
1427             8,
1428             8,
1429             8,
1430             8,
1431             8,
1432             8,
1433             8,
1434             8,
1435             8,
1436             8,
1437             8,
1438             8,
1439             8,
1440             8,
1441             8,
1442             8,
1443             8,
1444             8,
1445             8,
1446             8,
1447             8,
1448             8,
1449             8,
1450             8,
1451             8,
1452             8,
1453             8,
1454             8,
1455             8,
1456             8,
1457             8,
1458             8,
1459             8,
1460             8,
1461             8,
1462             8,
1463             8,
1464             8,
1465             8,
1466             8,
1467             8,
1468             8,
1469             8,
1470             8,
1471             8,
1472             8,
1473             8,
1474             8,
1475             8,
1476             8,
1477             8,
1478             8,
1479             8,
1480             8,
1481             8,
1482             8,
1483             8,
1484             8,
1485             8,
1486             8,
1487             8,
1488             8,
1489             8,
1490             8,
1491             8,
1492             8,
1493             8,
1494             8,
1495             8,
1496             8,
1497             8,
1498             8,
1499             8,
1500             8,
1501             8,
1502             8,
1503             8,
1504             8,
1505             8,
1506             8,
1507             8,
1508             8,
1509             8,
1510             8,
1511             8,
1512             8,
1513             8,
1514             8,
1515             8,
1516             8,
1517             8,
1518             8,
1519             8,
1520             8,
1521             8,
1522             8,
1523             8,
1524             8,
1525             8,
1526             8,
1527             8,
1528             8,
1529             8,
1530             8,
1531             8,
1532             8,
1533             8,
1534             8,
1535             8,
1536             8,
1537             8,
1538             8,
1539             8,
1540             8,
1541             8,
1542             8,
1543             8,
1544             8,
1545             8,
1546             8,
1547             8,
1548             8,
1549             8,
1550             8,
1551             8,
1552             8,
1553             8,
1554             8,
1555             8,
1556             8,
1557             8,
1558             8,
1559             8,
1560             8,
1561             8,
1562             8,
1563             8,
1564             8,
1565             8,
1566             8,
1567             8,
1568             8,
1569             8,
1570             8,
1571             8,
1572             8,
1573             8,
1574             8,
1575             8,
1576             8,
1577             8,
1578             8,
1579             8,
1580             8,
1581             8,
1582             8,
1583             8,
1584             8,
1585             8,
1586             8,
1587             8,
1588             8,
1589             8,
1590             8,
1591             8,
1592             8,
1593             8,
1594             8,
1595             8,
1596             8,
1597             8,
1598             8,
1599             8,
1600             8,
1601             8,
1602             8,
1603             8,
1604             8,
1605             8,
1606             8,
1607             8,
1608             8,
1609             8,
1610             8,
1611             8,
1612             8,
1613             8,
1614             8,
1615             8,
1616             8,
1617             8,
1618             8,
1619             8,
1620             8,
1621             8,
1622             8,
1623             8,
1624             8,
1625             8,
1626             8,
1627             8,
1628             8,
1629             8,
1630             8,
1631             8,
1632             8,
1633             8,
1634             8,
1635             8,
1636             8,
1637             8,
1638             8,
1639             8,
1640             8,
1641             8,
1642             8,
1643             8,
1644             8,
1645             8,
1646             8,
1647             8,
1648             8,
1649             8,

```

```

41
42     br(),
43     HTML("<font face=verdana size=5 color=#9e0093><b>Télé",
44           charger le rapport d'analyse</b></font>"),
45     br(), br(), br(),
46     radioButtons("format", "Sélectionner le format du",
47                   rapport", c('Word'), inline = TRUE),
48     downloadButton("DownloadReport", "Télécharger le",
49                   rapport", icon = icon("fas fa-download"), style = ',
50                   background-color:#9e0093; color:#000000; border-
51                   color:#9d9d9d;')
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
688
689
690
691
692
693
694
695
696
697
698
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
788
789
790
791
792
793
794
795
796
797
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2397
2398
239
```

Page essai standardisé 226

```

1 library(shiny)
2 library(DT)
3 library(nlme)
4 library(shinyCSSloaders)

5
6 Essai226Page <- function() {
7   tagList(
8     fluidPage(
9       titlePanel("Essai 226"),
10      fluidRow(
11        column(4,
12         textInput(inputId = "chemical", "Nom du produit chimique testé", value = "Produit chimique"),
13          selectInput("conc_unit", "Unité de Concentration", choices = c("g/kg", "mg/kg", "microg/kg", "ng/kg")),
14          fileInput('datafile_ESSAI226', 'Sélectionner un fichier', accept = c('.csv')),
15          radioButtons('model_ESSAI226', 'Sélectionner le modèle de régression',
16            choices = c('Linéaire' = 'lm', 'Non-linéaire' = 'nls', 'Mixte' = 'lme'),
17            selected = 'lme'),
18          numericInput(inputId = "ecx_ESSAI226", "Déterminer la concentration efficace X%", value = 50, min = 0, max = 100),
19          radioButtons('test_method_ESSAI226', 'Sélectionner la méthode de test d'hypothèse',
20            choices = c('ANOVA' = 'anova'), selected = 'anova'),
21          actionButton("buttonRunStep1", "Exécuter", icon = icon("check"), width = "100px")
22        ),
23        column(8,

```

```

24     mainPanel(
25       width = 12,
26       DT::dataTableOutput('rawdata'),
27       br(),
28       tableOutput('drc_result'),
29       withSpinner(plotOutput('drc_plot', width = "100%", 
30         height = "850 px"), type = 4, color = '#9e0093'),
31         br(),
32         withSpinner(verbatimTextOutput("test_result")),
33         br()
34       )
35     ),
36     tabPanel(
37       "Télécharger le rapport",
38       fixedRow(
39         column(
40           8,
41           br(),
42           HTML("<font face=verdana size=5 color=#9e0093><b>Télé- 
43             charger le rapport d'analyse</b></font>"),
44           br(), br(), br(),
45           radioButtons("format", "Sélectionner le format du 
46             rapport", c('Word'), inline = TRUE),
47           downloadButton("DownloadReport", "Télécharger le 
48             rapport", icon = icon("fas fa-download"), style = ' 
49             background-color:#9e0093; color:#000000; border- 
50             color:#9d9d9d;')
51         )
52       )
53     )
54   )
55 )
56 )
57 )
58 ))
59 }

```

Page essai standardisé 227

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(shinycssloaders)
5
6 Essai227Page <- function() {
7   tagList(
8     fluidPage(
9       titlePanel("Essai 227"),
10      fluidRow(
11        column(4,
12         textInput(inputId = "chemical", "Nom du produit 
13            chimique testé", value = "Produit chimique"),
14          selectInput("conc_unit", "Unité de Concentration", 
15            choices = c("g/L", "mg/L", "microg/L", "ng/L")),
16          fileInput('datafile_ESSAI227', 'Sélectionner un 
17            fichier', accept = c('.csv'))),
18        )
19      )
20    )
21  )
22 }

```

```

15         radioButtons('model_ESSAI227', 'Sélectionner le modèle
16             de régression',
17             choices = c('log-logistic 2 paramètres'
18             = 'l12',
19             'log-logistic 3 paramètres'
20             = 'l13',
21             'log-logistic 4 paramètres'
22             = 'l14'),
23             selected = 'l12'),
24         numericInput(inputId = "ecx_ESSAI227", "Déterminer la
25             concentration efficace X%", value = 50, min = 0,
26             max = 100),
27         radioButtons('test_method_ESSAI227', 'Sélectionner la
28             méthode de test d\'hypothèse',
29             choices = c('ANOVA' = 'anova'), selected
30             = 'anova'),
31         actionButton("buttonRunStep1", "Exécuter", icon = icon("check"),
32             width = "100px"),
33         column(8,
34             mainPanel(
35                 width = 12,
36                 DT::dataTableOutput('rawdata'),
37                 br(),
38                 tableOutput('drc_result'),
39                 withSpinner(plotOutput('drc_plot', width = "100%",
40                     height = "850px"), type = 4, color = "#9e0093),
41                     br(),
42                     withSpinner(verbatimTextOutput("test_result")),
43                     br()
44             )
45         ),
46         tabPanel(
47             "Télécharger le rapport",
48             fixedRow(
49                 column(
50                     8,
51                     br(),
52                     HTML("<font face=verdana size=5 color=#9e0093><b>Télé
53                         charger le rapport d'analyse </b></font>"),
54                     br(), br(), br(),
55                     radioButtons("format", "Sélectionner le format du
56                         rapport", c('Word'), inline = TRUE),
57                     downloadButton("DownloadReport", "Télécharger le
58                         rapport", icon = icon("fas fa-download"), style =
59                         "background-color:#9e0093; color:#000000; border-
60                         color:#9d9d9d;"),
61                 )
62             )
63         )
64     )
65 )
66 )
67 )
68 )
69 )
70 )
71 )
72 )
73 )
74 )
75 )
76 )
77 )
78 )
79 )
80 )
81 )
82 )
83 )
84 )
85 )
86 )
87 )
88 )
89 )
90 )
91 )
92 )
93 )
94 )
95 )
96 )
97 )
98 )
99 )
100 )
101 )
102 )
103 )
104 )
105 )
106 )
107 )
108 )
109 )
110 )
111 )
112 )
113 )
114 )
115 )
116 )
117 )
118 )
119 )
120 )
121 )
122 )
123 )
124 )
125 )
126 )
127 )
128 )
129 )
130 )
131 )
132 )
133 )
134 )
135 )
136 )
137 )
138 )
139 )
140 )
141 )
142 )
143 )
144 )
145 )
146 )
147 )
148 )
149 )
150 )
151 )
152 )
153 )
154 )
155 )
156 )
157 )
158 )
159 )
160 )
161 )
162 )
163 )
164 )
165 )
166 )
167 )
168 )
169 )
170 )
171 )
172 )
173 )
174 )
175 )
176 )
177 )
178 )
179 )
180 )
181 )
182 )
183 )
184 )
185 )
186 )
187 )
188 )
189 )
190 )
191 )
192 )
193 )
194 )
195 )
196 )
197 )
198 )
199 )
200 )
201 )
202 )
203 )
204 )
205 )
206 )
207 )
208 )
209 )
210 )
211 )
212 )
213 )
214 )
215 )
216 )
217 )
218 )
219 )
220 )
221 )
222 )
223 )
224 )
225 )
226 )
227 )
228 )
229 )
230 )
231 )
232 )
233 )
234 )
235 )
236 )
237 )
238 )
239 )
240 )
241 )
242 )
243 )
244 )
245 )
246 )
247 )
248 )
249 )
250 )
251 )
252 )
253 )
254 )
255 )
256 )
257 )
258 )
259 )
260 )
261 )
262 )
263 )
264 )
265 )
266 )
267 )
268 )
269 )
270 )
271 )
272 )
273 )
274 )
275 )
276 )
277 )
278 )
279 )
280 )
281 )
282 )
283 )
284 )
285 )
286 )
287 )
288 )
289 )
290 )
291 )
292 )
293 )
294 )
295 )
296 )
297 )
298 )
299 )
300 )
301 )
302 )
303 )
304 )
305 )
306 )
307 )
308 )
309 )
310 )
311 )
312 )
313 )
314 )
315 )
316 )
317 )
318 )
319 )
320 )
321 )
322 )
323 )
324 )
325 )
326 )
327 )
328 )
329 )
330 )
331 )
332 )
333 )
334 )
335 )
336 )
337 )
338 )
339 )
340 )
341 )
342 )
343 )
344 )
345 )
346 )
347 )
348 )
349 )
350 )
351 )
352 )
353 )
354 )
355 )
356 )
357 )
358 )
359 )
360 )
361 )
362 )
363 )
364 )
365 )
366 )
367 )
368 )
369 )
370 )
371 )
372 )
373 )
374 )
375 )
376 )
377 )
378 )
379 )
380 )
381 )
382 )
383 )
384 )
385 )
386 )
387 )
388 )
389 )
390 )
391 )
392 )
393 )
394 )
395 )
396 )
397 )
398 )
399 )
400 )
401 )
402 )
403 )
404 )
405 )
406 )
407 )
408 )
409 )
410 )
411 )
412 )
413 )
414 )
415 )
416 )
417 )
418 )
419 )
420 )
421 )
422 )
423 )
424 )
425 )
426 )
427 )
428 )
429 )
430 )
431 )
432 )
433 )
434 )
435 )
436 )
437 )
438 )
439 )
440 )
441 )
442 )
443 )
444 )
445 )
446 )
447 )
448 )
449 )
450 )
451 )
452 )
453 )
454 )
455 )
456 )
457 )
458 )
459 )
460 )
461 )
462 )
463 )
464 )
465 )
466 )
467 )
468 )
469 )
470 )
471 )
472 )
473 )
474 )
475 )
476 )
477 )
478 )
479 )
480 )
481 )
482 )
483 )
484 )
485 )
486 )
487 )
488 )
489 )
490 )
491 )
492 )
493 )
494 )
495 )
496 )
497 )
498 )
499 )
500 )
501 )
502 )
503 )
504 )
505 )
506 )
507 )
508 )
509 )
510 )
511 )
512 )
513 )
514 )
515 )
516 )
517 )
518 )
519 )
520 )
521 )
522 )
523 )
524 )
525 )
526 )
527 )
528 )
529 )
530 )
531 )
532 )
533 )
534 )
535 )
536 )
537 )
538 )
539 )
540 )
541 )
542 )
543 )
544 )
545 )
546 )
547 )
548 )
549 )
550 )
551 )
552 )
553 )
554 )
555 )
556 )
557 )
558 )
559 )
560 )
561 )
562 )
563 )
564 )
565 )
566 )
567 )
568 )
569 )
570 )
571 )
572 )
573 )
574 )
575 )
576 )
577 )
578 )
579 )
580 )
581 )
582 )
583 )
584 )
585 )
586 )
587 )
588 )
589 )
590 )
591 )
592 )
593 )
594 )
595 )
596 )
597 )
598 )
599 )
599 )
600 )
601 )
602 )
603 )
604 )
605 )
606 )
607 )
608 )
609 )
610 )
611 )
612 )
613 )
614 )
615 )
616 )
617 )
618 )
619 )
619 )
620 )
621 )
622 )
623 )
624 )
625 )
626 )
627 )
628 )
629 )
629 )
630 )
631 )
632 )
633 )
634 )
635 )
636 )
637 )
638 )
639 )
639 )
640 )
641 )
642 )
643 )
644 )
645 )
646 )
647 )
648 )
649 )
649 )
650 )
651 )
652 )
653 )
654 )
655 )
656 )
657 )
658 )
659 )
659 )
660 )
661 )
662 )
663 )
664 )
665 )
666 )
667 )
668 )
669 )
669 )
670 )
671 )
672 )
673 )
674 )
675 )
676 )
677 )
678 )
679 )
679 )
680 )
681 )
682 )
683 )
684 )
685 )
686 )
687 )
688 )
689 )
689 )
690 )
691 )
692 )
693 )
694 )
695 )
696 )
697 )
698 )
699 )
699 )
700 )
701 )
702 )
703 )
704 )
705 )
706 )
707 )
708 )
709 )
709 )
710 )
711 )
712 )
713 )
714 )
715 )
716 )
717 )
718 )
719 )
719 )
720 )
721 )
722 )
723 )
724 )
725 )
726 )
727 )
728 )
729 )
729 )
730 )
731 )
732 )
733 )
734 )
735 )
736 )
737 )
738 )
739 )
739 )
740 )
741 )
742 )
743 )
744 )
745 )
746 )
747 )
748 )
749 )
749 )
750 )
751 )
752 )
753 )
754 )
755 )
756 )
757 )
758 )
759 )
759 )
760 )
761 )
762 )
763 )
764 )
765 )
766 )
767 )
768 )
769 )
769 )
770 )
771 )
772 )
773 )
774 )
775 )
776 )
777 )
778 )
779 )
779 )
780 )
781 )
782 )
783 )
784 )
785 )
786 )
787 )
788 )
789 )
789 )
790 )
791 )
792 )
793 )
794 )
795 )
796 )
797 )
798 )
799 )
799 )
800 )
801 )
802 )
803 )
804 )
805 )
806 )
807 )
808 )
809 )
809 )
810 )
811 )
812 )
813 )
814 )
815 )
816 )
817 )
818 )
819 )
819 )
820 )
821 )
822 )
823 )
824 )
825 )
826 )
827 )
828 )
829 )
829 )
830 )
831 )
832 )
833 )
834 )
835 )
836 )
837 )
838 )
839 )
839 )
840 )
841 )
842 )
843 )
844 )
845 )
846 )
847 )
848 )
849 )
849 )
850 )
851 )
852 )
853 )
854 )
855 )
856 )
857 )
858 )
859 )
859 )
860 )
861 )
862 )
863 )
864 )
865 )
866 )
867 )
868 )
869 )
869 )
870 )
871 )
872 )
873 )
874 )
875 )
876 )
877 )
878 )
878 )
879 )
880 )
881 )
882 )
883 )
884 )
885 )
886 )
887 )
888 )
889 )
889 )
890 )
891 )
892 )
893 )
894 )
895 )
896 )
897 )
898 )
899 )
899 )
900 )
901 )
902 )
903 )
904 )
905 )
906 )
907 )
908 )
909 )
909 )
910 )
911 )
912 )
913 )
914 )
915 )
916 )
917 )
918 )
919 )
919 )
920 )
921 )
922 )
923 )
924 )
925 )
926 )
927 )
928 )
929 )
929 )
930 )
931 )
932 )
933 )
934 )
935 )
936 )
937 )
938 )
939 )
939 )
940 )
941 )
942 )
943 )
944 )
945 )
946 )
947 )
948 )
949 )
949 )
950 )
951 )
952 )
953 )
954 )
955 )
956 )
957 )
958 )
959 )
959 )
960 )
961 )
962 )
963 )
964 )
965 )
966 )
967 )
968 )
969 )
969 )
970 )
971 )
972 )
973 )
974 )
975 )
976 )
977 )
978 )
978 )
979 )
980 )
981 )
982 )
983 )
984 )
985 )
986 )
987 )
988 )
989 )
989 )
990 )
991 )
992 )
993 )
994 )
995 )
996 )
997 )
998 )
999 )
999 )
1000 )
1001 )
1002 )
1003 )
1004 )
1005 )
1006 )
1007 )
1008 )
1009 )
1009 )
1010 )
1011 )
1012 )
1013 )
1014 )
1015 )
1016 )
1017 )
1018 )
1019 )
1019 )
1020 )
1021 )
1022 )
1023 )
1024 )
1025 )
1026 )
1027 )
1028 )
1029 )
1029 )
1030 )
1031 )
1032 )
1033 )
1034 )
1035 )
1036 )
1037 )
1038 )
1039 )
1039 )
1040 )
1041 )
1042 )
1043 )
1044 )
1045 )
1046 )
1047 )
1048 )
1049 )
1049 )
1050 )
1051 )
1052 )
1053 )
1054 )
1055 )
1056 )
1057 )
1058 )
1059 )
1059 )
1060 )
1061 )
1062 )
1063 )
1064 )
1065 )
1066 )
1067 )
1068 )
1069 )
1069 )
1070 )
1071 )
1072 )
1073 )
1074 )
1075 )
1076 )
1077 )
1078 )
1078 )
1079 )
1080 )
1081 )
1082 )
1083 )
1084 )
1085 )
1086 )
1087 )
1088 )
1088 )
1089 )
1090 )
1091 )
1092 )
1093 )
1094 )
1095 )
1096 )
1097 )
1098 )
1098 )
1099 )
1099 )
1100 )
1101 )
1102 )
1103 )
1104 )
1105 )
1106 )
1107 )
1108 )
1109 )
1109 )
1110 )
1111 )
1112 )
1113 )
1114 )
1115 )
1116 )
1117 )
1118 )
1119 )
1119 )
1120 )
1121 )
1122 )
1123 )
1124 )
1125 )
1126 )
1127 )
1128 )
1129 )
1129 )
1130 )
1131 )
1132 )
1133 )
1134 )
1135 )
1136 )
1137 )
1138 )
1139 )
1139 )
1140 )
1141 )
1142 )
1143 )
1144 )
1145 )
1146 )
1147 )
1148 )
1149 )
1149 )
1150 )
1151 )
1152 )
1153 )
1154 )
1155 )
1156 )
1157 )
1158 )
1159 )
1159 )
1160 )
1161 )
1162 )
1163 )
1164 )
1165 )
1166 )
1167 )
1168 )
1169 )
1169 )
1170 )
1171 )
1172 )
1173 )
1174 )
1175 )
1176 )
1177 )
1178 )
1178 )
1179 )
1180 )
1181 )
1182 )
1183 )
1184 )
1185 )
1186 )
1187 )
1188 )
1188 )
1189 )
1189 )
1190 )
1191 )
1192 )
1193 )
1194 )
1195 )
1196 )
1197 )
1198 )
1198 )
1199 )
1199 )
1200 )
1201 )
1202 )
1203 )
1204 )
1205 )
1206 )
1207 )
1208 )
1209 )
1209 )
1210 )
1211 )
1212 )
1213 )
1214 )
1215 )
1216 )
1217 )
1218 )
1219 )
1219 )
1220 )
1221 )
1222 )
1223 )
1224 )
1225 )
1226 )
1227 )
1228 )
1229 )
1229 )
1230 )
1231 )
1232 )
1233 )
1234 )
1235 )
1236 )
1237 )
1238 )
1239 )
1239 )
1240 )
1241 )
1242 )
1243 )
1244 )
1245 )
1246 )
1247 )
1248 )
1249 )
1249 )
1250 )
1251 )
1252 )
1253 )
1254 )
1255 )
1256 )
1257 )
1258 )
1259 )
1259 )
1260 )
1261 )
1262 )
1263 )
1264 )
1265 )
1266 )
1267 )
1268 )
1269 )
1269 )
1270 )
1271 )
1272 )
1273 )
1274 )
1275 )
1276 )
1277 )
1278 )
1278 )
1279 )
1279 )
1280 )
1281 )
1282 )
1283 )
1284 )
1285 )
1286 )
1287 )
1288 )
1288 )
1289 )
1289 )
1290 )
1291 )
1292 )
1293 )
1294 )
1295 )
1296 )
1297 )
1297 )
1298 )
1298 )
1299 )
1299 )
1300 )
1301 )
1302 )
1303 )
1304 )
1305 )
1306 )
1307 )
1308 )
1309 )
1309 )
1310 )
1311 )
1312 )
1313 )
1314 )
1315 )
1316 )
1317 )
1318 )
1319 )
1319 )
1320 )
1321 )
1322 )
1323 )
1324 )
1325 )
1326 )
1327 )
1328 )
1329 )
1329 )
1330 )
1331 )
1332 )
1333 )
1334 )
1335 )
1336 )
1337 )
1338 )
1339 )
1339 )
1340 )
1341 )
1342 )
1343 )
1344 )
1345 )
1346 )
1347 )
1348 )
1349 )
1349 )
1350 )
1351 )
1352 )
1353 )
1354 )
1355 )
1356 )
1357 )
1358 )
1359 )
1359 )
1360 )
1361 )
1362 )
1363 )
1364 )
1365 )
1366 )
1367 )
1368 )
1369 )
1369 )
1370 )
1371 )
1372 )
1373 )
1374 )
1375 )
1376 )
1377 )
1378 )
1378 )
1379 )
1379 )
1380 )
1381 )
1382 )
1383 )
1384 )
1385 )
1386 )
1387 )
1388 )
1388 )
1389 )
1389 )
1390 )
1391 )
1392 )
1393 )
1394 )
1395 )
1396 )
1397 )
1398 )
1398 )
1399 )
1399 )
1400 )
1401 )
1402 )
1403 )
1404 )
1405 )
1406 )
1407 )
1408 )
1409 )
1409 )
1410 )
1411 )
1412 )
1413 )
1414 )
1415 )
1416 )
1417 )
1418 )
1419 )
1419 )
1420 )
1421 )
1422 )
1423 )
1424 )
1425 )
1426 )
1427 )
1428 )
1429 )
1429 )
1430 )
1431 )
1432 )
1433 )
1434 )
1435 )
1436 )
1437 )
1438 )
1439 )
1439 )
1440 )
1441 )
1442 )
1443 )
1444 )
1445 )
1446 )
1447 )
1448 )
1449 )
1449 )
1450 )
1451 )
1452 )
1453 )
1454 )
1455 )
1456 )
1457 )
1458 )
1459 )
1459 )
1460 )
1461 )
1462 )
1463 )
1464 )
1465 )
1466 )
1467 )
1468 )
1469 )
1469 )
1470 )
1471 )
1472 )
1473 )
1474 )
1475 )
1476 )
1477 )
1478 )
1478 )
1479 )
1479 )
1480 )
1481 )
1482 )
1483 )
1484 )
1485 )
1486 )
1487 )
1488 )
1488 )
1489 )
1489 )
1490 )
1491 )
1492 )
1493 )
1494 )
1495 )
1496 )
1497 )
1498 )
1498 )
1499 )
1499 )
1500 )
1501 )
1502 )
1503 )
1504 )
1505 )
1506 )
1507 )
1508 )
1509 )
1509 )
1510 )
1511 )
1512 )
1513 )
1514 )
1515 )
1516 )
1517 )
1518 )
1519 )
1519 )
1520 )
1521 )
1522 )
1523 )
1524 )
1525 )
1526 )
1527 )
1528 )
1529 )
1529 )
1530 )
1531 )
1532 )
1533 )
1534 )
1535 )
1536 )
1537 )
1538 )
1539 )
1539 )
1540 )
1541 )
1542 )
1543 )
1544 )
1545 )
1546 )
1547 )
1548 )
1549 )
1549 )
1550 )
1551 )
1552 )
1553 )
1554 )
1555 )
1556 )
1557 )
1558 )
1559 )
1559 )
1560 )
1561 )
1562 )
1563 )
1564 )
1565 )
1566 )
1567 )
1568 )
1569 )
1569 )
1570 )
1571 )
1572 )
1573 )
1574 )
1575 )
1576 )
1577 )
1578 )
1578 )
1579 )
1579 )
1580 )
1581 )
1582 )
1583 )
1584 )
1585 )
1586 )
1587 )
1588 )
1588 )
1589 )
1589 )
1590 )
1591 )
1592 )
1593 )
1594 )
1595 )
1596 )
1597 )
1598 )
1598 )
1599 )
1599 )
1600 )
1601 )
1602 )
1603 )
1604 )
1605 )
1606 )
1607 )
1608 )
1609 )
1609 )
1610 )
1611 )
1612 )
1613 )
1614 )
1615 )
1616 )
1617 )
1618 )
1619 )
1619 )
1620 )
1621 )
1622 )
1623 )
1624 )
1625 )
1626 )
1627 )
1628 )
1629 )
1629 )
1630 )
1631 )
1632 )
1633 )
1634 )
1635 )
1636 )
1637 )
1638 )
1639 )
1639 )
1640 )
1641 )
1642 )
1643 )
1644 )
1645 )
1646 )
1647 )
1648 )
1649 )
1649 )
1650 )
1651 )
1652 )
1653 )
1654 )
1655 )
1656 )
1657 )
1658 )
1659 )
1659 )
1660 )
1661 )
1662 )
1663 )
1664 )
1665 )
1666 )
1667 )
1668 )
1669 )
1669 )
1670 )
1671 )
1672 )
1673 )
1674 )
1675 )
1676 )
1677 )
1678 )
1678 )
1679 )
1679 )
1680 )
1681 )
1682 )
1683 )
1684 )
1685 )
1686 )
1687 )
1688 )
1688 )
1689 )
1689 )
1690 )
1691 )
1692 )
1693 )
1694 )
1695 )
1696 )
1697 )
1698 )
1698 )
1699 )
1699 )
1700 )
1701 )
1702 )
1703 )
1704 )
1705 )
1706 )
1707 )
1708 )
1709 )
1709 )
1710 )
1711 )
1712 )
1713 )
1714 )
1715 )
1716 )
1717 )
1718 )
1719 )
1719 )
1720 )
1721 )
1722 )
1723 )
1724 )
1725 )
1726 )
1727 )
1728 )
1729 )
1729 )
1730 )
1731 )
1732 )
1733 )
1734 )
1735 )
1736 )
1737 )
1738 )
1739 )
1739 )
1740 )
1741 )
1742 )
1743 )
1744 )
1745 )
1746 )
1747 )
1748 )
1749 )
1749 )
1750 )
1751 )
1752 )
1753 )
1754 )
1755 )
1756 )
1757 )
1758 )
1759 )
1759 )
1760 )
1761 )
1762 )
1763 )
1764 )
1765 )
1766 )
1767 )
1768 )
1769 )
1769 )
1770 )
1771 )
1772 )
1773 )
1774 )
1775 )
1776 )
1777 )
1778 )
1778 )
1779 )
1779 )
1780 )
1781 )
1782 )
1783 )
1784 )
1785 )
1786 )
1787 )
1788 )
1788 )
1789 )
1789 )
1790 )
1791 )
1792 )
1793 )
1794 )
1795 )
1796 )
1797 )
1798 )
1798 )
1799 )
1799 )
1800 )
1801 )
1802 )
1803 )
1804 )
1805 )
1806 )
1807 )
1808 )
1809 )
1809 )
1810 )
1811 )
1812 )
1813 )
1814 )
1815 )
1816 )
1817 )
1818 )
1819 )
1819 )
1820 )
1821 )
1822 )
1823 )
1824 )
1825 )
1826 )
1827 )
1828 )
1829 )
1829 )
1830 )
1831 )
1832 )
1833 )
1834 )
1835 )
1836 )
1837 )
1838 )
1839 )
1839 )
1840 )
1841 )
1842 )
1843 )
1844 )
1845 )
1846 )
1847 )
1848 )
1849 )
1849 )
1850 )
1851 )
1852 )
1853 )
1854 )
1855 )
1856 )
1857 )
1858 )
1859 )
1859 )
1860 )
1861 )
1862 )
1863 )
1864 )
1865 )
1866 )
1867 )
1868 )
1869 )
1869 )
1870 )
1871 )
1872 )
1873 )
1874 )
1875 )
1876 )
1877 )
1878 )
1878 )
1879 )
1879 )
1880 )
1881 )
1882 )
1883 )
1884 )
1885 )
1886 )
1887 )
1888 )
1888 )
1889 )
1889 )
1890 )
1891 )
1892 )
1893 )
1894 )
1895 )
1896 )
1897 )
1898 )
1898 )
1899 )
1899 )
1900 )
1901 )
1902 )
1903 )
1904 )
1905 )
1906 )
1907 )
1908 )
1909 )
1909 )
1910 )
1911 )
1912 )
1913 )
1914 )
1915 )
1916 )
1917 )
1918 )
1919 )
1919 )
1920 )
1921 )
1922 )
1923 )
1924 )
1925 )
1926 )
1927 )
1928 )
1929 )
1929 )
1930 )
1931 )
1932 )
1933 )
1934 )
1935 )
1936 )
1937 )
1938 )
1939 )
1939 )
1940 )
1941 )
1942 )
1943 )
1944 )
1945 )
1946 )
1947 )
1948 )
1949 )
1949 )
1950 )
1951 )
1952 )
1953 )
1954 )
1955 )
1956 )
1957 )
1958 )
1959 )
1959 )
1960 )
1961 )
1962 )
1963 )
1964 )
1965 )
1966 )
1967 )
1968 )
1969 )
1969 )
1970 )
1971 )
1972 )
1973 )
1974 )
1975 )
1976 )
1977 )
1978 )
1978 )
1979 )
1979 )
1980 )
1981 )
1982 )
1983 )
1984 )
1985 )
1986 )
1987 )
1988 )
1988 )
1989 )
1989 )
1990 )
1991 )
1992 )
1993 )
1994 )
1995 )
1996 )
1997 )
1998 )
1999 )
1999 )
2000 )
2001 )
2002 )
2003 )
2004 )
2005 )
2006 )
2007 )
2008 )
2009 )
2009 )
2010 )
2011 )
2012 )
2013 )
2014 )
2015 )
2016 )
2017 )
2018 )
2019 )
2019 )
2020 )
2021 )
2022 )
2023 )
2024 )
2025 )
2026 )
2027 )
2028 )
2029 )
2029 )
2030 )
2031 )
2032 )
2033 )
2034 )
2035 )
2036 )
2037 )
2038 )
2039 )
2039 )
2040 )
2041 )
2042 )
2043 )
2044 )
2045 )
2046 )
2047 )
2048 )
2049 )
2049 )
2050 )
2051 )
2052 )
2053 )
2054 )
2055 )
2056 )
2057 )
2058 )
2059 )
2059 )
2060 )
2061 )
2062 )
2063 )
2064 )
2065 )
2066 )
2067 )
2068 )
2069 )
2069 )
2070 )
2071 )
2072 )
2073 )
2074 )
2075 )
2076 )
2077 )
2078 )
2078 )
2079 )
2079 )
2080 )
2081 )
2082 )
2083 )
2084 )
2085 )
2086 )
2087 )
2088 )
2088 )
2089 )
2089 )
2090 )
2091 )
2092 )
2093 )
2094 )
2095 )
2096 )
2097 )
2098 )
2098 )
2099 )
2099 )
2100 )
2101 )
2102 )
2103 )
2104 )
2105 )
2106 )
2107 )
2108 )
2109 )
2109 )
2110 )
2111 )
2112 )
2113 )
2114 )
2115 )
2116 )
2117 )
2118 )
2119 )
2119 )
2120 )
2121 )
2122 )
2123 )
2124 )
2125 )
2126 )
2127 )
2128 )
2129 )
2129 )
2130 )
2131 )
2132 )
2133 )
2134 )
2135 )
2136 )
2137 )
2138 )
2139 )
2139 )
2140 )
2141 )
2142 )
2143 )
2144 )
2145 )
2146 )
2147 )
2148 )
2149 )
2149 )
2150 )
2151 )
2152 )
2153 )
2154 )
2155 )
2156 )
2157 )
2158 )
2159 )
2159 )
2160 )
2161 )
2162 )
2163 )
2164 )
2165 )
2166 )
2167 )
2168 )
2169 )
2169 )
2170 )
2171 )
2172 )
2173 )
2174 )
2175 )
2176 )
2177 )
2178 )
2178 )
2179 )
2179 )
2180 )
2181 )
2182 )
2183 )
2184 )
2185 )
2186 )
2187 )
2188 )
2188 )
2189 )
2189 )
2190 )
2191 )
2192 )
2193 )
2194 )
2195 )
2196 )
2197 )
2198 )
2198 )
2199 )
2199 )
2200 )
2201 )
2202 )
2203 )
2204 )
2205 )
2206 )
2207 )
2208 )
2209 )
2209 )
2210 )
2211 )
2212 )
2213 )
2214 )
2215 )
2216 )
2217 )
2218 )
2219 )
2219 )
2220 )
2221 )
2222 )
2223 )
2224 )
2225 )
2226 )
2227 )
2228 )
2229 )
2229 )
2230 )
2231 )
2232 )
2233 )
2234 )
2235 )
2236 )
2237 )
2238 )
2239 )
2239 )
2240 )
2241 )
2242 )
2243 )
2244 )
2245 )
2246 )
2247 )
2248 )
2249 )
2249 )
2250 )
2251 )
2252 )
2253 )
2254 )
2255 )
2256 )
2257 )
2258 )
2259 )
2259 )
2260 )
2261 )
2262 )
2263 )
2264 )
2265 )
2266 )
2267 )
2268 )
2269 )
2269 )
2270 )
2271 )
2272 )
2273 )
2274 )
2275 )
2276 )
2277 )
2278 )
2278 )
2279 )
2279 )
2280 )
2281 )
2282 )
2283 )
2284 )
2285 )
2286 )
2287 )
2288 )
2288 )
2289 )
2289 )
2290 )
2291 )
2292 )
2293 )
2294 )
2295 )
2296 )
2297 )
2298 )
2298 )
2299 )
2299 )
2300 )
2301 )
2302 )
2303 )
2304 )
2305 )
2306 )
2307 )
2308 )
2309 )
2309 )
2310 )
2311 )
2312 )
2313 )
2314 )
2315 )
2316 )
2317 )
2318 )
2319 )
2319 )
2320 )
2321 )
2322 )
2323 )
2324 )
2325 )
2326 )
2327 )
2328 )
2329 )
2329 )
2330 )
2331 )
2332 )
2333 )
2334 )
2335 )
2336 )
2337 )
2338 )
2339 )
2339 )
2340 )
2341 )
2342 )
2343 )
2344 )
2345 )
2346 )
2347 )
2348 )
2349 )
2349 )
2350 )
2351 )
2352 )
2353 )
2354 )
2355 )
2356 )
2357 )
2358 )
2359 )
2359 )
2360 )
2361 )
2362 )
2363 )
2364 )
2365 )
2366 )
2367 )
2368 )
2369 )
2369 )
2370 )
2371 )
2372 )
2373 )
2374 )
2375 )
2376 )
2377 )
2378 )
2378 )
2379 )
2379 )
2380 )
2381 )
2382 )
2383 )
2384 )
2385 )
2386 )
2387 )
2388 )
2388 )
2389 )
2389 )
2390 )
2391 )
2392 )
2393 )
2394 )
2395 )
2396 )
2397 )
2398 )
2398 )
2399 )
2399 )
2400 )
2401 )
24
```

Page essai standardisé 232

```

1 library(shiny)
2 library(DT)
3 library(nlme)
4 library(shinyCSSloaders)
5
6 Essai232Page <- function() {
7   tagList(
8     fluidPage(
9       titlePanel("Collembolan Reproduction Test in Soil"),
10      fluidRow(
11        column(4,
12         textInput(inputId = "chemical", "Nom du produit
13            chimique testé", value = "Produit chimique"),
14          selectInput("conc_unit", "Unité de Concentration",
15            choices = c("g/kg", "mg/kg", "microg/kg", "ng/kg")
16            ),
17          fileInput('datafile_ESSAI232', 'Sélectionner un
18            fichier', accept = c('.csv')),
19          radioButtons('model_ESSAI232', 'Sélectionner le modè
20            le de régression',
21            choices = c('Linéaire' = 'lm', 'Non-liné
22              aire' = 'nls', 'Mixte' = 'lme'),
23            selected = 'lme'),
24          numericInput(inputId = "ecx_ESSAI232", "Déterminer la
25            concentration efficace X%", value = 50, min = 0,
26            max = 100),
27          radioButtons('test_method_ESSAI232', 'Sélectionner la
28            méthode de test d\'hypothèse',
29            choices = c('ANOVA' = 'anova'), selected
30            = 'anova'),
31          actionButton("buttonRunStep1", "Exécuter", icon =
32            icon("check"), width = "100px")
33        ),
34        column(8,
35          mainPanel(
36            width = 12,
37            DT::dataTableOutput('rawdata'),
38            br(),
39            tableOutput('drc_result'),
40            withSpinner(plotOutput('drc_plot', width = "100%",
41              height = "850px"), type = 4, color = "#9e0093"),
42            br(),
43            withSpinner(verbatimTextOutput("test_result")),
44            br()
45          )
46        ),
47        tabPanel(
48          "Télécharger le rapport",
49          fixedRow(

```

```
38     column(
39         8,
40         br(),
41         HTML("<font face=verdana size=5 color=#9e0093><b>Télé-
42             charger le rapport d'analyse</b></font>"),
43         br(), br(), br(),
44         radioButtons("format", "Sélectionner le format du
45             rapport", c('Word'), inline = TRUE),
46         downloadButton("DownloadReport", "Télécharger le
47             rapport", icon = icon("fas fa-download"), style = '
48                 background-color:#9e0093; color:#000000; border-
49                 color:#9d9d9d; ")
50     )
51 }
```

Page essai standardisé 239

```

24     actionButton("buttonRunStep239", "Exécuter", icon = icon("check"), width = "100px")
25   ),
26   column(8,
27     mainPanel(
28       width = 12,
29       DT::dataTableOutput('rawdata'),
30       br(),
31       tableOutput('drc_result'),
32       withSpinner(plotOutput('drc_plot', width = "100%", height = "850px"), type = 4, color = "#9e0093"),
33         br(),
34         withSpinner(verbatimTextOutput("test_result")),
35         br()
36       )
37     ),
38     tabPanel(
39       "Télécharger le rapport",
40       fixedRow(
41         column(
42           8,
43           br(),
44           HTML("<font face=verdana size=5 color=#9e0093><b>Télé
45             charger le rapport d'analyse</b></font>"),
46           br(), br(), br(),
47           radioButtons("format", "Sélectionner le format du
48             rapport", c('Word'), inline = TRUE),
49           downloadButton("DownloadReport", "Télécharger le
50             rapport", icon = icon("fas fa-download"), style = "
51               background-color:#9e0093; color:#000000; border-
52               color:#9d9d9d; ")
53         )
54       )
55     )
56   )
57 )
58 ))
59 }

```

```
library(shiny)
library(DT)
library(drc)
library(multcomp)
library(survival)
library(shinycssloaders)

Essai245Page <- function() {
  tagList(
    fluidPage(
      titlePanel("Honey-Bee (Apis Mellifera L.), Chronic Oral
      Toxicity Test (10-Day Feeding)),
      fluidRow(
```

```

13     column(4,
14        textInput(inputId = "chemical", "Nom du produit chimique testé", value = "Produit chimique"),
15         selectInput("conc_unit", "Unité de Dose", choices = c("g/L", "mg/L", "microg/L", "ng/L")),
16         fileInput('datafile_ESSAI245', 'Sélectionner un fichier', accept = c('.csv')),
17         radioButtons('model_ESSAI245', 'Sélectionner le modèle de régression',
18             choices = c('Cox' = 'coxph'),
19             selected = 'coxph'),
20         numericInput(inputId = "ecx_ESSAI245", "Déterminer la dose efficace X%", value = 50, min = 0, max =
21             100),
22         radioButtons('test_method_ESSAI245', 'Sélectionner la méthode de test d\'hypothèse',
23             choices = c('Test de Dunnett' = 'Dunnett'),
24             selected = 'Dunnett'),
25         actionButton("buttonRunStep1", "Exécuter", icon = icon("check"), width = "100px")
26     ),
27     column(8,
28         mainPanel(
29             width = 12,
30             DT::dataTableOutput('rawdata'),
31             br(),
32             tableOutput('drc_result'),
33             withSpinner(plotOutput('drc_plot', width = "100%", height = "850px"), type = 4, color = '#9e0093'),
34             br(),
35             withSpinner(verbatimTextOutput("test_result")),
36             br()
37         )
38     ),
39     tabPanel(
40         "Télécharger le rapport",
41         fixedRow(
42             column(
43                 8,
44                 br(),
45                 HTML("<font face=verdana size=5 color=#9e0093><b>Télécharger le rapport d'analyse</b></font>"),
46                 br(), br(), br(),
47                 radioButtons("format", "Sélectionner le format du rapport", c('Word'), inline = TRUE),
48                 downloadButton("DownloadReport", "Télécharger le rapport", icon = icon("fas fa-download"), style = 'background-color:#9e0093; color:#000000; border-color:#9d9d9d;')
49             )
50         )
51     )
52 )
53 )
54 )
55 )
56 )
57 )
58 )
59 )
60 )
61 )
62 )
63 )
64 )
65 )
66 )
67 )
68 )
69 )
70 )
71 )
72 )
73 )
74 )
75 )
76 )
77 )
78 )
79 )
80 )
81 )
82 )
83 )
84 )
85 )
86 )
87 )
88 )
89 )
90 )
91 )
92 )
93 )
94 )
95 )
96 )
97 )
98 )
99 )
100 )
101 )
102 )
103 )
104 )
105 )
106 )
107 )
108 )
109 )
110 )
111 )
112 )
113 )
114 )
115 )
116 )
117 )
118 )
119 )
120 )
121 )
122 )
123 )
124 )
125 )
126 )
127 )
128 )
129 )
130 )
131 )
132 )
133 )
134 )
135 )
136 )
137 )
138 )
139 )
140 )
141 )
142 )
143 )
144 )
145 )
146 )
147 )
148 )
149 )
150 )
151 )
152 )
153 )
154 )
155 )
156 )
157 )
158 )
159 )
160 )
161 )
162 )
163 )
164 )
165 )
166 )
167 )
168 )
169 )
170 )
171 )
172 )
173 )
174 )
175 )
176 )
177 )
178 )
179 )
180 )
181 )
182 )
183 )
184 )
185 )
186 )
187 )
188 )
189 )
190 )
191 )
192 )
193 )
194 )
195 )
196 )
197 )
198 )
199 )
200 )
201 )
202 )
203 )
204 )
205 )
206 )
207 )
208 )
209 )
210 )
211 )
212 )
213 )
214 )
215 )
216 )
217 )
218 )
219 )
220 )
221 )
222 )
223 )
224 )
225 )
226 )
227 )
228 )
229 )
230 )
231 )
232 )
233 )
234 )
235 )
236 )
237 )
238 )
239 )
240 )
241 )
242 )
243 )
244 )
245 )
246 )
247 )
248 )
249 )
250 )
251 )
252 )
253 )
254 )
255 )
256 )
257 )
258 )
259 )
260 )
261 )
262 )
263 )
264 )
265 )
266 )
267 )
268 )
269 )
270 )
271 )
272 )
273 )
274 )
275 )
276 )
277 )
278 )
279 )
280 )
281 )
282 )
283 )
284 )
285 )
286 )
287 )
288 )
289 )
290 )
291 )
292 )
293 )
294 )
295 )
296 )
297 )
298 )
299 )
300 )
301 )
302 )
303 )
304 )
305 )
306 )
307 )
308 )
309 )
310 )
311 )
312 )
313 )
314 )
315 )
316 )
317 )
318 )
319 )
320 )
321 )
322 )
323 )
324 )
325 )
326 )
327 )
328 )
329 )
330 )
331 )
332 )
333 )
334 )
335 )
336 )
337 )
338 )
339 )
340 )
341 )
342 )
343 )
344 )
345 )
346 )
347 )
348 )
349 )
350 )
351 )
352 )
353 )
354 )
355 )
356 )
357 )
358 )
359 )
360 )
361 )
362 )
363 )
364 )
365 )
366 )
367 )
368 )
369 )
370 )
371 )
372 )
373 )
374 )
375 )
376 )
377 )
378 )
379 )
380 )
381 )
382 )
383 )
384 )
385 )
386 )
387 )
388 )
389 )
390 )
391 )
392 )
393 )
394 )
395 )
396 )
397 )
398 )
399 )
400 )
401 )
402 )
403 )
404 )
405 )
406 )
407 )
408 )
409 )
410 )
411 )
412 )
413 )
414 )
415 )
416 )
417 )
418 )
419 )
420 )
421 )
422 )
423 )
424 )
425 )
426 )
427 )
428 )
429 )
430 )
431 )
432 )
433 )
434 )
435 )
436 )
437 )
438 )
439 )
440 )
441 )
442 )
443 )
444 )
445 )
446 )
447 )
448 )
449 )
450 )
451 )
452 )
453 )
454 )
455 )
456 )
457 )
458 )
459 )
460 )
461 )
462 )
463 )
464 )
465 )
466 )
467 )
468 )
469 )
470 )
471 )
472 )
473 )
474 )
475 )
476 )
477 )
478 )
479 )
480 )
481 )
482 )
483 )
484 )
485 )
486 )
487 )
488 )
489 )
490 )
491 )
492 )
493 )
494 )
495 )
496 )
497 )
498 )
499 )
500 )
501 )
502 )
503 )
504 )
505 )
506 )
507 )
508 )
509 )
510 )
511 )
512 )
513 )
514 )
515 )
516 )
517 )
518 )
519 )
520 )
521 )
522 )
523 )
524 )
525 )
526 )
527 )
528 )
529 )
530 )
531 )
532 )
533 )
534 )
535 )
536 )
537 )
538 )
539 )
540 )
541 )
542 )
543 )
544 )
545 )
546 )
547 )
548 )
549 )
550 )
551 )
552 )
553 )
554 )
555 )
556 )
557 )
558 )
559 )
560 )
561 )
562 )
563 )
564 )
565 )
566 )
567 )
568 )
569 )
570 )
571 )
572 )
573 )
574 )
575 )
576 )
577 )
578 )
579 )
580 )
581 )
582 )
583 )
584 )
585 )
586 )
587 )
588 )
589 )
589 )
590 )
591 )
592 )
593 )
594 )
595 )
596 )
597 )
598 )
599 )
599 )
600 )
601 )
602 )
603 )
604 )
605 )
606 )
607 )
608 )
609 )
609 )
610 )
611 )
612 )
613 )
614 )
615 )
616 )
617 )
618 )
619 )
619 )
620 )
621 )
622 )
623 )
624 )
625 )
626 )
627 )
628 )
629 )
629 )
630 )
631 )
632 )
633 )
634 )
635 )
636 )
637 )
638 )
639 )
639 )
640 )
641 )
642 )
643 )
644 )
645 )
646 )
647 )
648 )
649 )
649 )
650 )
651 )
652 )
653 )
654 )
655 )
656 )
657 )
658 )
659 )
659 )
660 )
661 )
662 )
663 )
664 )
665 )
666 )
667 )
668 )
669 )
669 )
670 )
671 )
672 )
673 )
674 )
675 )
676 )
677 )
678 )
679 )
679 )
680 )
681 )
682 )
683 )
684 )
685 )
686 )
687 )
688 )
689 )
689 )
690 )
691 )
692 )
693 )
694 )
695 )
696 )
697 )
698 )
699 )
699 )
700 )
701 )
702 )
703 )
704 )
705 )
706 )
707 )
708 )
709 )
709 )
710 )
711 )
712 )
713 )
714 )
715 )
716 )
717 )
718 )
719 )
719 )
720 )
721 )
722 )
723 )
724 )
725 )
726 )
727 )
728 )
729 )
729 )
730 )
731 )
732 )
733 )
734 )
735 )
736 )
737 )
738 )
739 )
739 )
740 )
741 )
742 )
743 )
744 )
745 )
746 )
747 )
748 )
749 )
749 )
750 )
751 )
752 )
753 )
754 )
755 )
756 )
757 )
758 )
759 )
759 )
760 )
761 )
762 )
763 )
764 )
765 )
766 )
767 )
768 )
769 )
769 )
770 )
771 )
772 )
773 )
774 )
775 )
776 )
777 )
778 )
779 )
779 )
780 )
781 )
782 )
783 )
784 )
785 )
786 )
787 )
788 )
789 )
789 )
790 )
791 )
792 )
793 )
794 )
795 )
796 )
797 )
798 )
799 )
799 )
800 )
801 )
802 )
803 )
804 )
805 )
806 )
807 )
808 )
809 )
809 )
810 )
811 )
812 )
813 )
814 )
815 )
816 )
817 )
818 )
819 )
819 )
820 )
821 )
822 )
823 )
824 )
825 )
826 )
827 )
828 )
829 )
829 )
830 )
831 )
832 )
833 )
834 )
835 )
836 )
837 )
838 )
839 )
839 )
840 )
841 )
842 )
843 )
844 )
845 )
846 )
847 )
848 )
849 )
849 )
850 )
851 )
852 )
853 )
854 )
855 )
856 )
857 )
858 )
859 )
859 )
860 )
861 )
862 )
863 )
864 )
865 )
866 )
867 )
868 )
869 )
869 )
870 )
871 )
872 )
873 )
874 )
875 )
876 )
877 )
878 )
878 )
879 )
880 )
881 )
882 )
883 )
884 )
885 )
886 )
887 )
888 )
888 )
889 )
889 )
890 )
891 )
892 )
893 )
894 )
895 )
896 )
897 )
898 )
898 )
899 )
899 )
900 )
901 )
902 )
903 )
904 )
905 )
906 )
907 )
908 )
909 )
909 )
910 )
911 )
912 )
913 )
914 )
915 )
916 )
917 )
918 )
919 )
919 )
920 )
921 )
922 )
923 )
924 )
925 )
926 )
927 )
928 )
929 )
929 )
930 )
931 )
932 )
933 )
934 )
935 )
936 )
937 )
938 )
939 )
939 )
940 )
941 )
942 )
943 )
944 )
945 )
946 )
947 )
948 )
949 )
949 )
950 )
951 )
952 )
953 )
954 )
955 )
956 )
957 )
958 )
959 )
959 )
960 )
961 )
962 )
963 )
964 )
965 )
966 )
967 )
968 )
969 )
969 )
970 )
971 )
972 )
973 )
974 )
975 )
976 )
977 )
978 )
978 )
979 )
979 )
980 )
981 )
982 )
983 )
984 )
985 )
986 )
987 )
988 )
988 )
989 )
989 )
990 )
991 )
992 )
993 )
994 )
995 )
996 )
997 )
998 )
999 )
999 )
1000 )
1001 )
1002 )
1003 )
1004 )
1005 )
1006 )
1007 )
1008 )
1009 )
1009 )
1010 )
1011 )
1012 )
1013 )
1014 )
1015 )
1016 )
1017 )
1018 )
1019 )
1019 )
1020 )
1021 )
1022 )
1023 )
1024 )
1025 )
1026 )
1027 )
1028 )
1029 )
1029 )
1030 )
1031 )
1032 )
1033 )
1034 )
1035 )
1036 )
1037 )
1038 )
1039 )
1039 )
1040 )
1041 )
1042 )
1043 )
1044 )
1045 )
1046 )
1047 )
1048 )
1049 )
1049 )
1050 )
1051 )
1052 )
1053 )
1054 )
1055 )
1056 )
1057 )
1058 )
1059 )
1059 )
1060 )
1061 )
1062 )
1063 )
1064 )
1065 )
1066 )
1067 )
1068 )
1069 )
1069 )
1070 )
1071 )
1072 )
1073 )
1074 )
1075 )
1076 )
1077 )
1078 )
1078 )
1079 )
1079 )
1080 )
1081 )
1082 )
1083 )
1084 )
1085 )
1086 )
1087 )
1088 )
1088 )
1089 )
1089 )
1090 )
1091 )
1092 )
1093 )
1094 )
1095 )
1095 )
1096 )
1096 )
1097 )
1098 )
1099 )
1099 )
1100 )
1101 )
1102 )
1103 )
1104 )
1105 )
1106 )
1107 )
1108 )
1109 )
1109 )
1110 )
1111 )
1112 )
1113 )
1114 )
1115 )
1116 )
1117 )
1118 )
1119 )
1119 )
1120 )
1121 )
1122 )
1123 )
1124 )
1125 )
1126 )
1127 )
1128 )
1129 )
1129 )
1130 )
1131 )
1132 )
1133 )
1134 )
1135 )
1136 )
1137 )
1138 )
1139 )
1139 )
1140 )
1141 )
1142 )
1143 )
1144 )
1145 )
1146 )
1147 )
1148 )
1148 )
1149 )
1149 )
1150 )
1151 )
1152 )
1153 )
1154 )
1155 )
1156 )
1157 )
1158 )
1159 )
1159 )
1160 )
1161 )
1162 )
1163 )
1164 )
1165 )
1166 )
1167 )
1168 )
1169 )
1169 )
1170 )
1171 )
1172 )
1173 )
1174 )
1175 )
1176 )
1177 )
1178 )
1178 )
1179 )
1179 )
1180 )
1181 )
1182 )
1183 )
1184 )
1185 )
1186 )
1187 )
1188 )
1188 )
1189 )
1189 )
1190 )
1191 )
1192 )
1193 )
1194 )
1195 )
1195 )
1196 )
1196 )
1197 )
1198 )
1199 )
1199 )
1200 )
1201 )
1202 )
1203 )
1204 )
1205 )
1206 )
1207 )
1208 )
1209 )
1209 )
1210 )
1211 )
1212 )
1213 )
1214 )
1215 )
1216 )
1217 )
1218 )
1219 )
1219 )
1220 )
1221 )
1222 )
1223 )
1224 )
1225 )
1226 )
1227 )
1228 )
1229 )
1229 )
1230 )
1231 )
1232 )
1233 )
1234 )
1235 )
1236 )
1237 )
1238 )
1238 )
1239 )
1239 )
1240 )
1241 )
1242 )
1243 )
1244 )
1245 )
1246 )
1247 )
1248 )
1248 )
1249 )
1249 )
1250 )
1251 )
1252 )
1253 )
1254 )
1255 )
1256 )
1257 )
1258 )
1259 )
1259 )
1260 )
1261 )
1262 )
1263 )
1264 )
1265 )
1266 )
1267 )
1268 )
1269 )
1269 )
1270 )
1271 )
1272 )
1273 )
1274 )
1275 )
1276 )
1277 )
1278 )
1278 )
1279 )
1279 )
1280 )
1281 )
1282 )
1283 )
1284 )
1285 )
1286 )
1287 )
1288 )
1288 )
1289 )
1289 )
1290 )
1291 )
1292 )
1293 )
1294 )
1295 )
1295 )
1296 )
1296 )
1297 )
1298 )
1299 )
1299 )
1300 )
1301 )
1302 )
1303 )
1304 )
1305 )
1306 )
1307 )
1308 )
1309 )
1309 )
1310 )
1311 )
1312 )
1313 )
1314 )
1315 )
1316 )
1317 )
1318 )
1318 )
1319 )
1319 )
1320 )
1321 )
1322 )
1323 )
1324 )
1325 )
1326 )
1327 )
1328 )
1328 )
1329 )
1329 )
1330 )
1331 )
1332 )
1333 )
1334 )
1335 )
1336 )
1337 )
1338 )
1338 )
1339 )
1339 )
1340 )
1341 )
1342 )
1343 )
1344 )
1345 )
1346 )
1347 )
1348 )
1348 )
1349 )
1349 )
1350 )
1351 )
1352 )
1353 )
1354 )
1355 )
1356 )
1357 )
1358 )
1359 )
1359 )
1360 )
1361 )
1362 )
1363 )
1364 )
1365 )
1366 )
1367 )
1368 )
1368 )
1369 )
1369 )
1370 )
1371 )
1372 )
1373 )
1374 )
1375 )
1376 )
1377 )
1378 )
1378 )
1379 )
1379 )
1380 )
1381 )
1382 )
1383 )
1384 )
1385 )
1386 )
1387 )
1388 )
1388 )
1389 )
1389 )
1390 )
1391 )
1392 )
1393 )
1394 )
1395 )
1395 )
1396 )
1396 )
1397 )
1398 )
1399 )
1399 )
1400 )
1401 )
1402 )
1403 )
1404 )
1405 )
1406 )
1407 )
1408 )
1409 )
1409 )
1410 )
1411 )
1412 )
1413 )
1414 )
1415 )
1416 )
1417 )
1418 )
1418 )
1419 )
1419 )
1420 )
1421 )
1422 )
1423 )
1424 )
1425 )
1426 )
1427 )
1428 )
1428 )
1429 )
1429 )
1430 )
1431 )
1432 )
1433 )
1434 )
1435 )
1436 )
1437 )
1438 )
1438 )
1439 )
1439 )
1440 )
1441 )
1442 )
1443 )
1444 )
1445 )
1446 )
1447 )
1448 )
1448 )
1449 )
1449 )
1450 )
1451 )
1452 )
1453 )
1454 )
1455 )
1456 )
1457 )
1458 )
1459 )
1459 )
1460 )
1461 )
1462 )
1463 )
1464 )
1465 )
1466 )
1467 )
1468 )
1468 )
1469 )
1469 )
1470 )
1471 )
1472 )
1473 )
1474 )
1475 )
1476 )
1477 )
1478 )
1478 )
1479 )
1479 )
1480 )
1481 )
1482 )
1483 )
1484 )
1485 )
1486 )
1487 )
1488 )
1488 )
1489 )
1489 )
1490 )
1491 )
1492 )
1493 )
1494 )
1495 )
1495 )
1496 )
1496 )
1497 )
1498 )
1499 )
1499 )
1500 )
1501 )
1502 )
1503 )
1504 )
1505 )
1506 )
1507 )
1508 )
1509 )
1509 )
1510 )
1511 )
1512 )
1513 )
1514 )
1515 )
1516 )
1517 )
1518 )
1518 )
1519 )
1519 )
1520 )
1521 )
1522 )
1523 )
1524 )
1525 )
1526 )
1527 )
1528 )
1528 )
1529 )
1529 )
1530 )
1531 )
1532 )
1533 )
1534 )
1535 )
1536 )
1537 )
1538 )
1538 )
1539 )
1539 )
1540 )
1541 )
1542 )
1543 )
1544 )
1545 )
1546 )
1547 )
1548 )
1548 )
1549 )
1549 )
1550 )
1551 )
1552 )
1553 )
1554 )
1555 )
1556 )
1557 )
1558 )
1559 )
1559 )
1560 )
1561 )
1562 )
1563 )
1564 )
1565 )
1566 )
1567 )
1568 )
1568 )
1569 )
1569 )
1570 )
1571 )
1572 )
1573 )
1574 )
1575 )
1576 )
1577 )
1578 )
1578 )
1579 )
1579 )
1580 )
1581 )
1582 )
1583 )
1584 )
1585 )
1586 )
1587 )
1588 )
1588 )
1589 )
1589 )
1590 )
1591 )
1592 )
1593 )
1594 )
1595 )
1595 )
1596 )
1596 )
1597 )
1598 )
1599 )
1599 )
1600 )
1601 )
1602 )
1603 )
1604 )
1605 )
1606 )
1607 )
1608 )
1609 )
1609 )
1610 )
1611 )
1612 )
1613 )
1614 )
1615 )
1616 )
1617 )
1618 )
1618 )
1619 )
1619 )
1620 )
1621 )
1622 )
1623 )
1624 )
1625 )
1626 )
1627 )
1628 )
1628 )
1629 )
1629 )
1630 )
1631 )
1632 )
1633 )
1634 )
1635 )
1636 )
1637 )
1638 )
1638 )
1639 )
1639 )
1640 )
1641 )
1642 )
1643 )
1644 )
1645 )
1646 )
1647 )
1648 )
1648 )
1649 )
1649 )
1650 )
1651 )
1652 )
1653 )
1654 )
1655 )
1656 )
1657 )
1658 )
1659 )
1659 )
1660 )
1661 )
1662 )
1663 )
1664 )
1665 )
1666 )
1667 )
1668 )
1668 )
1669 )
1669 )
1670 )
1671 )
1672 )
1673 )
1674 )
1675 )
1676 )
1677 )
1678 )
1678 )
1679 )
1679 )
1680 )
1681 )
1682 )
1683 )
1684 )
1685 )
1686 )
1687 )
1688 )
1688 )
1689 )
1689 )
1690 )
1691 )
1692 )
1693 )
1694 )
1695 )
1695 )
1696 )
1696 )
1697 )
1698 )
1699 )
1699 )
1700 )
1701 )
1702 )
1703 )
1704 )
1705 )
1706 )
1707 )
1708 )
1709 )
1709 )
1710 )
1711 )
1712 )
1713 )
1714 )
1715 )
1716 )
1717 )
1718 )
1718 )
1719 )
1719 )
1720 )
1721 )
1722 )
1723 )
1724 )
1725 )
1726 )
1727 )
1728 )
1728 )
1729 )
1729 )
1730 )
1731 )
1732 )
1733 )
1734 )
1735 )
1736 )
1737 )
1738 )
1738 )
1739 )
1739 )
1740 )
1741 )
1742 )
1743 )
1744 )
1745 )
1746 )
1747 )
1748 )
1748 )
1749 )
1749 )
1750 )
1751 )
1752 )
1753 )
1754 )
1755 )
1756 )
1757 )
1758 )
1759 )
1759 )
1760 )
1761 )
1762 )
1763 )
1764 )
1765 )
1766 )
1767 )
1768 )
1768 )
1769 )
1769 )
1770 )
1771 )
1772 )
1773 )
1774 )
1775 )
1776 )
1777 )
1778 )
1778 )
1779 )
1779 )
1780 )
1781 )
1782 )
1783 )
1784 )
1785 )
1786 )
1787 )
1788 )
1788 )
1789 )
1789 )
1790 )
1791 )
1792 )
1793 )
1794 )
1795 )
1795 )
1796 )
1796 )
1797 )
1798 )
1799 )
1799 )
1800 )
1801 )
1802 )
1803 )
1804 )
1805 )
1806 )
1807 )
1808 )
1809 )
1809 )
1810 )
1811 )
1812 )
1813 )
1814 )
1815 )
1816 )
1817 )
1818 )
1818 )
1819 )
1819 )
1820 )
1821 )
1822 )
1823 )
1824 )
1825 )
1826 )
1827 )
1828 )
1828 )
1829 )
1829 )
1830 )
1831 )
1832 )
1833 )
1834 )
1835 )
1836 )
1837 )
1838 )
1838 )
1839 )
1839 )
1840 )
1841 )
1842 )
1843 )
1844 )
1845 )
1846 )
1847 )
1848 )
1848 )
1849 )
1849 )
1850 )
1851 )
1852 )
1853 )
1854 )
1855 )
1856 )
1857 )
1858 )
1859 )
1859 )
1860 )
1861 )
1862 )
1863 )
1864 )
1865 )
1866 )
1867 )
1868 )
1868 )
1869 )
1869 )
1870 )
1871 )
1872 )
1873 )
1874 )
1875 )
1876 )
1877 )
1878 )
1878 )
1879 )
1879 )
1880 )
1881 )
1882 )
1883 )
1884 )
1885 )
1886 )
1887 )
1888 )
1888 )
1889 )
1889 )
1890 )
1891 )
1892 )
1893 )
1894 )
1895 )
1895 )
1896 )
1896 )
1897 )
1898 )
1899 )
1899 )
1900 )
1901 )
1902 )
1903 )
1904 )
1905 )
1906 )
1907 )
1908 )
1909 )
1909 )
1910 )
1911 )
1912 )
1913 )
1914 )
1915 )
1916 )
1917 )
1918 )
1918 )
1919 )
1919 )
1920 )
1921 )
1922 )
1923 )
1924 )
1925 )
1926 )
1927 )
1928 )
1928 )
1929 )
1929 )
1930 )
1931 )
1932 )
1933 )
1934 )
1935 )
1936 )
1937 )
1938 )
1938 )
1939 )
1939 )
1940 )
1941 )
1942 )
1943 )
1944 )
1945 )
1946 )
1947 )
1948 )
1948 )
1949 )
1949 )
1950 )
1951 )
1952 )
1953 )
1954 )
1955 )
1956 )
1957 )
1958 )
1959 )
1959 )
1960 )
1961 )
1962 )
1963 )
1964 )
1965 )
1966 )
1967 )
1968 )
1968 )
1969 )
1969 )
1970 )
1971 )
1972 )
1973 )
1974 )
1975 )
1976 )
1977 )
1978 )
1978 )
1979 )
1979 )
1980 )
1981 )
1982 )
1983 )
1984 )
1985 )
1986 )
1987 )
1988 )
1988 )
1989 )
1989 )
1990 )
1991 )
1992 )
1993 )
1994 )
1995 )
1995 )
1996 )
1996 )
1997 )
1998 )
1999 )
1999 )
2000 )
2001 )
2002 )
2003 )
2004 )
2005 )
2006 )
2007 )
2008 )
2008 )
2009 )
2009 )
2010 )
2011 )
2012 )
2013 )
2014 )
2015 )
2016 )
2017 )
2018 )
2018 )
2019 )
2019 )
2020 )
2021 )
2022 )
2023 )
2024 )
2025 )
2026 )
2027 )
2028 )
2029 )
2029 )
2030 )
2031 )
2032 )
2033 )
2034 )
2035 )
2036 )
2037 )
2038 )
2038 )
2039 )
2039 )
2040 )
2041 )
2042 )
2043 )
2044 )
2045 )
2046 )
2047 )
2048 )
2048 )
2049 )
2049 )
2050 )
2051 )
2052 )
2053 )
2054 )
2055 )
2056 )
2057 )
2058 )
2059 )
2059 )
2060 )
2061 )
2062 )
2063 )
2064 )
2065 )
2066 )
2067 )
2068 )
2068 )
2069 )
2069 )
2070 )
2071 )
2072 )
2073 )
2074 )
2075 )
2076 )
2077 )
2078 )
2078 )
2079 )
2079 )
2080 )
2081 )
2082 )
2083 )
2084 )
2085 )
2086 )
2087 )
2088 )
2088 )
2089 )
2089 )
2090 )
2091 )
2092 )
2093 )
2094 )
2095 )
2095 )
2096 )
2096 )
2097 )
2098 )
2099 )
2099 )
2100 )
2101 )
2102 )
2103 )
2104 )
2105 )
2106 )
2107 )
2108 )
2109 )
2109 )
2110 )
2111 )
2112 )
2113 )
2114 )
2115 )
2116 )
2117 )
2118 )
2118 )
2119 )
2119 )
2120 )
2121 )
2122 )
2123 )
2124 )
2125 )
2
```

```

50      )
51  ))
52 }

```

Page essai standardisé 246

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(shinyCSSloaders)
5
6 Essai246Page <- function() {
7   tagList(
8     fluidPage(
9       titlePanel("Bumblebee , Acute Contact Toxicity Test"),
10      fluidRow(
11        column(4,
12         textInput(inputId = "chemical", "Nom du produit
13            chimique testé", value = "Produit chimique"),
14          selectInput("conc_unit", "Unité de Concentration",
15            choices = c("g/L", "mg/L", "microg/L", "ng/L")),
16          fileInput('datafile_ESSAI246', 'Sélectionner un
17            fichier', accept = c('.csv')),
18          radioButtons('model_ESSAI246', 'Sélectionner le modè
19            le de régression',
20            choices = c('Probit' = 'probit', 'Logit'
21              = 'logit'),
22            selected = 'probit'),
23          numericInput(inputId = "ecx_ESSAI246", "Déterminer la
24            concentration efficace X%", value = 50, min = 0,
25            max = 100),
26          radioButtons('test_method_ESSAI246', 'Sélectionner la
27            méthode de test \ 'hypothèse',
28            choices = c('Test de Dunnett' = 'Dunnett
29            '),
30            selected = 'Dunnett'),
31          actionButton("buttonRunStep1", "Exécuter", icon =
32            icon("check"),
33            width = "100px"),
34        ),
35        column(8,
36          mainPanel(
37            width = 12,
38            DT::dataTableOutput('rawdata'),
39            br(),
40            tableOutput('drc_result'),
41            withSpinner(plotOutput('drc_plot', width = "100%",
42              height = "850px"),
43              type = 4, color = "#9e0093"),
44              br(),
45              withSpinner(verbatimTextOutput("test_result")),
46              br()
47            )
48          ),
49          tabPanel(
50            "Télécharger le rapport",
51          )
52        )
53      )
54    )
55  )
56 }

```

```

37     fixedRow(
38         column(
39             8,
40             br(),
41             HTML("<font face=verdana size=5 color=#9e0093><b>Télé-
42             charger le rapport d'analyse</b></font>"),
43             br(), br(), br(),
44             radioButtons("format", "Sélectionner le format du
45             rapport", c('Word'), inline = TRUE),
46             downloadButton("DownloadReport", "Télécharger le
47             rapport", icon = icon("fas fa-download"), style = ,
48             background-color:#9e0093; color:#000000; border-
49             color:#9d9d9d; )
50         )
51     )
52 )
53 )
54 )
55 )
56 }

```

Page essai standardisé 247

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(shinyCSSloaders)
5
6 Essai247Page <- function() {
7     tagList(
8         fluidPage(
9             titlePanel("Bumblebee , Acute Oral Toxicity Test"),
10            fluidRow(
11                column(4,
12                   textInput(inputId = "chemical", "Nom du produit
13                    chimique testé", value = "Produit chimique"),
14                    selectInput("conc_unit", "Unité de Concentration",
15                        choices = c("g/L", "mg/L", "microg/L", "ng/L")),
16                    fileInput('datafile_ESSAI247', 'Sélectionner un
17                    fichier', accept = c('.csv')),
18                    radioButtons('model_ESSAI247', 'Sélectionner le modè
19                    le de régression',
20                        choices = c('Probit' = 'probit', 'Logit'
21                            = 'logit'),
22                        selected = 'probit'),
23                    numericInput(inputId = "ecx_ESSAI247", "Déterminer la
24                    concentration efficace X%", value = 50, min = 0,
25                    max = 100),
26                    radioButtons('test_method_ESSAI247', 'Sélectionner la
27                    méthode de test \ 'hypothèse',
28                    choices = c('Test de Dunnett' = 'Dunnett
29                        '),
30                    selected = 'Dunnett'),
31                    actionButton("buttonRunStep1", "Exécuter", icon =
32                        icon("check"), width = "100px")
33                )
34            )
35        )
36    )
37 }

```

```

22  ),
23  column(8,
24  mainPanel(
25  width=12,
26  DT::dataTableOutput('rawdata'),
27  br(),
28  tableOutput('drc_result'),
29  withSpinner(plotOutput('drc_plot', width="100%", height="850px"), type=4, color="#9e0093"),
30  br(),
31  withSpinner(verbatimTextOutput("test_result")),
32  br()
33  )
34  ),
35  tabPanel(
36  "Télécharger le rapport",
37  fixedRow(
38  column(
39  8,
40  br(),
41  HTML("<font face=verdana size=5 color=#9e0093><b>Télé
42  charger le rapport d'analyse</b></font>"),
43  br(), br(), br(),
44  radioButtons("format", "Sélectionner le format du
45  rapport", c('Word'), inline = TRUE),
46  downloadButton("DownloadReport", "Télécharger le
47  rapport", icon = icon("fas fa-download"), style = '
48  background-color:#9e0093; color:#000000; border-
49  color:#9d9d9d;')
50  )
51  )
52  )
53  )
54  )
55  )
56  )
57  )
58  )
59  )
50 }

```

Page essai standardisé 213_214

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(multcomp)
5 library(shinyCSSloaders)
6
7 Essai213214Page <- function() {
8   tagList(
9     fluidPage(
10       titlePanel("Honeybees , Acute Oral/Contact"),
11       fluidRow(
12         column(4,
13            textInput(inputId = "chemical", "Nom du produit
14             chimique testé", value = "Produit chimique"),
15             selectInput("conc_unit", "Unité de Dose", choices = c
16             )
17           )
18         )
19       )
20     )
21   )
22 }

```

```

15         ("g/L", "mg/L", "microg/L", "ng/L"))),
16     fileInput('datafile_ESSAI213214', 'Sélectionner un
17         fichier', accept = c('.csv')),
18     radioButtons('model_ESSAI213214', 'Sélectionner le
19         modèle de régression',
20             choices = c('Probit' = 'probit', 'Logit'
21             = 'logit'),
22             selected = 'probit'),
23     numericInput(inputId = "ecx_ESSAI213214", "Déterminer
24         la dose efficace X%", value = 50, min = 0, max =
25         100),
26     radioButtons('test_method_ESSAI213214', 'Sélectionner
27         la méthode de test d\'hypothèse',
28         choices = c('Test de Dunnett' = 'Dunnett
29             '),
30             selected = 'Dunnett'),
31     actionButton("buttonRunStep1", "Exécuter", icon =
32         icon("check"),
33             width = "100px"),
34         ),
35         ),
36     mainPanel(
37         width = 12,
38         DT::dataTableOutput('rawdata'),
39         br(),
40         tableOutput('drc_result'),
41         withSpinner(plotOutput('drc_plot', width = "100%",
42             height = "850px"),
43             type = 4,
44             color = "#9e0093"),
45             br(),
46             withSpinner(verbatimTextOutput("test_result")),
47             br()
48         )
49     ),
50   )
51 }

```

```

1 library(shiny)
2 library(DT)
3 library(drc)
4 library(multcomp)
5 library(shinyCSSloaders)
6
7 Essai246247Page <- function() {
8   tagList(
9     fluidPage(
10       titlePanel("Bumblebee , Acute Contact / Oral Toxicity"),
11       fluidRow(
12         column(4,
13          textInput(inputId = "chemical", "Nom du produit
14             chimique testé", value = "Produit chimique"),
15           selectInput("conc_unit", "Unité de Dose", choices =
16             c("g/L", "mg/L", "microg/L", "ng/L")),
17           fileInput('datafile_ESSAI246247', 'Sélectionner un
18             fichier', accept = c('.csv')),
19           radioButtons('model_ESSAI246247', 'Sélectionner le
20             modèle de régression',
21             choices = c('Probit' = 'probit', 'Logit'
22               = 'logit'),
23             selected = 'probit'),
24           numericInput(inputId = "ecx_ESSAI246247", "Déterminer
25             la dose efficace X%", value = 50, min = 0, max =
26             100),
27           radioButtons('test_method_ESSAI246247', 'Sélectionner
28             la méthode de test d\'hypothèse',
29             choices = c('Test de Dunnett' = 'Dunnett
30             '),
31             selected = 'Dunnett'),
32           actionButton("buttonRunStep1", "Exécuter", icon =
33             icon("check"), width = "100px")
34         ),
35         column(8,
36           mainPanel(
37             DT::dataTableOutput('rawdata'),
38             br(),
39             tableOutput('drc_result'),
40             withSpinner(plotOutput('drc_plot', width = "100%",
41               height = "850px"), type = 4, color = "#9e0093),
42               br(),
43               withSpinner(verbatimTextOutput("test_result")),
44               br()
45             )
46           ),
47           tabPanel(
48             "Télécharger le rapport",
49             fixedRow(
50               column(
51                 8,
52                 4
53               )
54             )
55           )
56         )
57       )
58     )
59   )
60 }

```

```

41     br(),
42     HTML("<font face=verdana size=5 color=#9e0093><b>Télé
43       charger le rapport d'analyse</b></font>"),
44     br(), br(), br(),
45     radioButtons("format", "Sélectionner le format du
46       rapport", c('Word'), inline = TRUE),
47     downloadButton("DownloadReport", "Télécharger le
48       rapport", icon = icon("fas fa-download"), style = ,
49       background-color:#9e0093; color:#000000; border-
50       color:#9d9d9d;')
51   )
  )
)
)
)
}

```

Annexe A : Inventaire des tests

Test	Forces	Limites	Critères de Choix
Dunnett's	Contrôle le taux d'erreur de Type I;utile avec un groupe de contrôle	Exige normalité et variances équivalentes	Plusieurs traitements comparés à un groupe de contrôle;données normalement distribuées avec homogénéité des variances
Welch's	Ne nécessite pas l'homogénéité des variances	Moins puissant que l'ANOVA avec variances égales	Deux groupes avec variances inégales pour comparaison de moyennes
Brown-Forsythe	Robuste aux écarts de normalité;teste l'égalité des médianes	Moins puissant dans certains cas	Comparaison de moyennes quand l'homogénéité des variances est violée

Table 16: Tests Paramétriques: Forces, Limites et Critères de Choix

Test	Forces	Limites	Critères de Choix
Cochran-Armitage Trend	Déetecte une tendance dans des proportions sur niveaux ordinaux	Conçu uniquement pour données catégorielles ordinaires	Données catégorielles ordinaires avec tendance suspectée
Jonckheere-Terpstra	Teste des alternatives ordonnées sans supposer une distribution spécifique	L'ordre des groupes doit être spécifié à l'avance	Déetecter des tendances ordonnées parmi les médianes de groupes indépendants
Mantel-Haenszel	Adapte pour l'analyse de données stratifiées; contrôle pour les variables de confusion	Limité aux tableaux de contingence 2x2	Test de l'association entre deux variables binaires en contrôlant pour variables de confusion

Table 17: Tests Non-Paramétriques: Forces, Limites et Critères de Choix

Annexe B : Rappels Théoriques

Définitions

Définition du percentile et du quantile

Pour tout nombre x dans l'intervalle $0 \leq x \leq 100$, le x -ième percentile d'un ensemble de données fini est défini comme un nombre qui est supérieur ou égal à au moins $x\%$ des données et est inférieur ou égal à au moins $(100 - x)\%$ des données.

Règle des valeurs aberrantes de Tukey (nommée d'après John W. Tukey, 1977)

Toute réponse supérieure à 1.5 fois l'intervalle interquartile (IQR) au-dessus du troisième quartile ou en dessous du premier quartile est considérée comme une valeur aberrante selon la règle de Tukey. Cette règle est motivée par les propriétés d'une distribution normale.

D Distributions de Probabilité

La fonction de distribution pour une variable aléatoire X est une fonction réelle non négative F définie par :

$$F(x) = P(X \leq x) \quad \text{pour tous les nombres réels } x$$

où $P(E)$ est la probabilité que E se produise.

References

- [1] European Union, *Regulation (CE) n°1107/2009*, Official Journal of the European Union, 2009.
- [2] Organisation for Economic Co-operation and Development, *OECD Guidelines for the Testing of Chemicals*, Document Guide 54,2006.
[https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://one.oecd.org/document/ENV/JM/MONO\(2006\)18/en/pdf&ved=2ahUKEwi04KCy1vyHAxVCRKQEHUpSLTEQFnoECBUQAQ&usg=A0vVaw1IUob7mCm0Ish8Xwl8D85V](https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://one.oecd.org/document/ENV/JM/MONO(2006)18/en/pdf&ved=2ahUKEwi04KCy1vyHAxVCRKQEHUpSLTEQFnoECBUQAQ&usg=A0vVaw1IUob7mCm0Ish8Xwl8D85V).
- [3] John W. Green, Timothy A. Springer, and Henrik Holbech, *Statistical Analysis of Ecotoxicity Studies*, Publisher Name, 2018.
- [4] Moore, D. S., McCabe, G. P., & Craig, B. A., *Introduction to the Practice of Statistics*, W.H. Freeman, 2016.
- [5] Johnson, R. A., & Wichern, D. W., *Applied Multivariate Statistical Analysis (6th ed.)*, Pearson, 2007.
- [6] Bates, D. M., & Watts, D. G. (1988). *Nonlinear Regression Analysis and Its Applications*. Wiley.
- [7] Burnham, Kenneth P., et David R. Anderson. 2004. *Multimodel Inference: Understanding AIC and BIC in Model Selection*. Sociological Methods & Research 33 (2): 261-304. <https://doi.org/10.1177/0049124104268644>.
- [8] Burnham, Kenneth P., David R. Anderson, et Kathryn P. Huyvaert. 2011. « *AIC Model Selection and Multimodel Inference in Behavioral Ecology: Some Background, Observations, and Comparisons* Behavioral Ecology and Sociobiology <https://doi.org/10.1007/s00265-010-1029-6>.
- [9] Philippe, A. *Cours sur le Bootstrap et les Méthodes Bayésiennes*. Université Nantes, Master Ingénierie Statistique, 2023-2024.
- [10] B. Efron. (1979). *Bootstrap Methods: Another Look at the Jackknife*. Annals of Statistics, 7(1), 1-26. <https://doi.org/10.1214/aos/1176344552>
- [11] *Bayesian Data Analysis* Third edition (with errors fixed as of 15 February 2021) by Andrew Gelman, John Carlin, Hal Stern, Donald Rubin, David Dunson, and Aki Vehtari
- [12] *Introducing Monte Carlo Methods with R*. Springer.
- [13] *Bayesian Theory*. Wiley.
- [14] *Choix des a priori en inférence bayésienne*: Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). **Bayesian Data Analysis** (3rd ed.). CRC Press.
- [15] *A priori informatifs*: O'Hagan, A., & Forster, J. (2004). Kendall's Advanced Theory of Statistics, Volume 2B: Bayesian Inference (2nd ed.). Arnold.
- [16] *A priori non informatifs*: Bernardo, J. M., & Smith, A. F. M. (2000). Bayesian Theory. Wiley.

- [17] *A priori partiellement informatifs*: Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1(3), 515-533.
- [18] *A priori conjugués*: Raiffa, H., & Schlaifer, R. (1961). *Applied Statistical Decision Theory*. Wiley.
- [19] *A priori empiriques*: Casella, G., & Berger, R. L. (2002). *Statistical Inference* (2nd ed.). Duxbury.
- [20] *Application à l'écotoxicologie*: Clark, J. S. (2005). Why environmental scientists are becoming Bayesians. *Ecology Letters*, 8(1), 2-14.
- [21] *Hazardous and Radioactive Substances in Danish Marine Waters. Status and Temporal Trends*. January 2009 Publisher: National Environmental Research Institute
- [22] Stamm, A. *Cours sur la Théorie des Tests de Permutation*. Université Nantes, Master Ingénierie Statistique, 2023-2024.
- [23] Good, Phillip I. (2005). *Permutation, Parametric and Bootstrap Tests of Hypotheses*, 3rd ed., Springer. ISBN 0-387-98898-X.
- [24] Barnett, V., & Lewis, T. (1994). *Outliers in Statistical Data* (3rd ed.). John Wiley & Sons.
- [25] Grubbs, F. E. (1969). Procedures for Detecting Outlying Observations in Samples. *Technometrics*, 11(1), 1-21.
- [26] Zuur, A. F., Ieno, E. N., & Elphick, C. S. (2010). A protocol for data exploration to avoid common statistical problems. *Methods in Ecology and Evolution*, 1(1), 3-14.
- [27] OCDE201 : *Algues d'eau douce et cyanobactéries*.
<https://www.oecd-ilibrary.org/fr/test-no-201-alga-growth-inhibition-test-51mqcr2k7s5h.pdf?itemId=%2Fcontent%2Fpublication%2F9789264069923-en&mimeType=pdf>
- [28] OCDE202. *Daphnia sp., Essai d'immobilisation immédiate*.
<https://www.oecd-ilibrary.org/fr/test-no-202-daphnia-sp-acute-immobilisation-test-51mqcr2k7s28.pdf?itemId=%2Fcontent%2Fpublication%2F9789264069947-en&mimeType=pdf>
- [29] OCDE203. *Poisson tox aiguë*.
<https://www.oecd-ilibrary.org/fr/test-no-203-fish-acute-toxicity-test-51mqcr2k7rzt.pdf?itemId=%2Fcontent%2Fpublication%2F9789264069961-en&mimeType=pdf>
- [30] OCDE205. *Oiseaux, Essai de Toxicité Liée au Régime*
<https://www.oecd-ilibrary.org/fr/test-no-205-avian-dietary-toxicity-test-51mqcr2k7rs4.pdf?itemId=%2Fcontent%2Fpublication%2F9789264070004-en&mimeType=pdf>
- [31] OCDE206. *Oiseaux reproduction*.
<https://www.oecd-ilibrary.org/fr/test-no-206-avian-reproduction-test-51mqcr2k7rq8.pdf?itemId=%2Fcontent%2Fpublication%2F9789264070028-en&mimeType=pdf>

- [32] OCDE210. *Poisson, essai de toxicité aux premiers stades de la vie.*
<https://www.oecd-ilibrary.org/fr/test-no-210-fish-early-life-stage-toxicity-test-5k43bdxrtps5.pdf?itemId=%2Fcontent%2Fpublication%2F9789264203785-en&mimeType=pdf>
- [33] OCDE211. *Daphnia magna, essai de reproduction.*
<https://www.oecd-ilibrary.org/fr/test-no-211-daphnia-magna-reproduction-test-5k91hspv1jmr.pdf?itemId=%2Fcontent%2Fpublication%2F9789264185203-en&mimeType=pdf>
- [34] OCDE213. *Abeille domestique, essai de toxicité aiguë par voie orale.*
<https://www.oecd-ilibrary.org/fr/test-no-213-honeybees-acute-oral-toxicity-test-5lmqcr2k7r34.pdf?itemId=%2Fcontent%2Fpublication%2F9789264070165-en&mimeType=pdf>
- [35] OCDE214. *Abeille domestique, essai de toxicité aiguë par contact.*
<https://www.oecd-ilibrary.org/fr/test-no-214-honeybees-acute-contact-toxicity-test-5lmqcr2k7r0n.pdf?itemId=%2Fcontent%2Fpublication%2F9789264070189-en&mimeType=pdf>
- [36] OCDE216. *Micro-organismes du sol essai de transformation de l'azote.*
<https://www.oecd-ilibrary.org/fr/test-no-216-soil-microorganisms-nitrogen-transformation-5lmqcr2k7qtk.pdf?itemId=%2Fcontent%2Fpublication%2F9789264070226-en&mimeType=pdf>
- [37] OCDE218. *Essai de toxicité sur les chironomes dans un système eau-sédiment chargé*
<https://www.oecd-ilibrary.org/fr/test-no-218-sediment-water-chironomid-toxicity-us-5lmng4d8td9w.pdf?itemId=%2Fcontent%2Fpublication%2F9789264070264-en&mimeType=pdf>
- [38] OCDE221. *Lemma sp. Essais d'inhibition de la croissance.*
<https://www.oecd-ilibrary.org/fr/test-no-221-lemma-sp-growth-inhabition-test-519njf5w5bhg.pdf?itemId=%2Fcontent%2Fpublication%2F9789264016194-en&mimeType=pdf>
- [39] OCDE222. *Reproduction chez le ver de terre.*
<https://www.oecd-ilibrary.org/fr/test-no-222-earthworm-reproduction-test-eisenia-5jlsh07trg0r.pdf?itemId=%2Fcontent%2Fpublication%2F9789264264496-en&mimeType=pdf>
- [40] OCDE223. *Oiseaux, toxicité orale aiguë.*
<https://www.oecd-ilibrary.org/fr/test-no-223-avian-acute-oral-toxicity-test-5jlsh05ks9g2.pdf?itemId=%2Fcontent%2Fpublication%2F9789264264519-en&mimeType=pdf>
- [41] OCDE226. *Reproduction d'un acarien prédateur.*
<https://www.oecd-ilibrary.org/fr/test-no-226-predatory-mite-hypoaspis-geolaelaps-a-5jlsh07trg0r.pdf?itemId=%2Fcontent%2Fpublication%2F9789264264519-en&mimeType=pdf>

- 5jlsh04vc8mw.pdf?itemId=%2Fcontent%2Fpublication%2F9789264264557-en&mimeType=pdf
- [42] OCDE232. *Reproduction de collemboles dans le sol.*
<https://www.oecd-ilibrary.org/fr/test-no-232-collembolan-reproduction-test-in-soil-5jlsh03vrzjj.pdf?itemId=%2Fcontent%2Fpublication%2F9789264264601-en&mimeType=pdf>
- [43] OCDE245. *Abeille domestique (Apis mellifera L.), Essai de Toxicité Chronique par Voie Orale (10 jours).*
<https://www.oecd-ilibrary.org/fr/test-no-245-honey-bee-apis-mellifera-l-chronic-or-5jfmclm8xmwg.pdf?itemId=%2Fcontent%2Fpublication%2F9789264284081-en&mimeType=pdf>
- [44] OCDE246. *Bourdon, Essai de Toxicité Aiguë par Contact.*
<https://www.oecd-ilibrary.org/fr/test-no-246-bumblebee-acute-contact-toxicity-test-5jfmchlgbzn.pdf?itemId=%2Fcontent%2Fpublication%2F9789264284104-en&mimeType=pdf>
- [45] OCDE247. *Bourdon, Essai de Toxicité Aiguë par Voie Orale.*
<https://www.oecd-ilibrary.org/fr/test-no-247-bumblebee-acute-oral-toxicity-test-5jfmcldkg1jc.pdf?itemId=%2Fcontent%2Fpublication%2F9789264284128-en&mimeType=pdf>
- [46] OCDE208. *Plantes.*
<https://www.oecd-ilibrary.org/fr/test-no-208-terrestrial-plant-test-seedling-emerg-5lmqcr2k7rjl.pdf?itemId=%2Fcontent%2Fpublication%2F9789264070066-en&mimeType=pdf>
- [47] OCDE227. *Plantes terrestres vigueur végétative.*
<https://www.oecd-ilibrary.org/fr/test-no-227-terrestrial-plant-test-vegetative-vig-519gc7rvtdcf5.pdf?itemId=%2Fcontent%2Fpublication%2F9789264067295-en&mimeType=pdf>
- [48] OCDE239. *Plante aquatique système eau-sédiment.*
<https://www.oecd-ilibrary.org/fr/test-no-239-water-sediment-myriophyllum-spicatum-5jxwv1xxcx1.pdf?itemId=%2Fcontent%2Fpublication%2F9789264224155-en&mimeType=pdf>
- [49] ESCORT. *Escort.*
https://www.pan-europe.info/sites/pan-europe.info/files/public/resources/other/_ESCORT%20%20_non-target%20arthropods.pdf
- [50] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B., *Bayesian Data Analysis* (3rd ed.), CRC Press, 2013.
- [51] O'Hagan, A., & Forster, J., *Kendall's Advanced Theory of Statistics, Volume 2B: Bayesian Inference* (2nd ed.), Arnold, 2004.
- [52] Bernardo, J. M., & Smith, A. F. M., *Bayesian Theory*, Wiley, 2000.

- [53] *Nonlinear Regression Analysis and Its Applications* Douglas M. Bates, Donald G. Watts
First published: 26 August 1988
- [54] *Nonlinear Regression Analysis* Hsin-Hsiung Huang and Qing He [https://arxiv.org/pdf/2402.05342](https://arxiv.org/pdf/2402.05342.pdf)
- [55] Gelman, A., *Prior distributions for variance parameters in hierarchical models*,
Bayesian Analysis, 1(3), 515-533, 2006.
- [56] Raiffa, H., & Schlaifer, R., *Applied Statistical Decision Theory*, Wiley, 1961.
- [57] Casella, G., & Berger, R. L., *Statistical Inference* (2nd ed.), Duxbury, 2002.
- [58] Clark, J. S., *Why environmental scientists are becoming Bayesians*, Ecology Letters, 8(1), 2-14, 2005.