Développement et validation de méthodes de comblement de lacunes de mesures (gapfilling) sur les traits de plantes

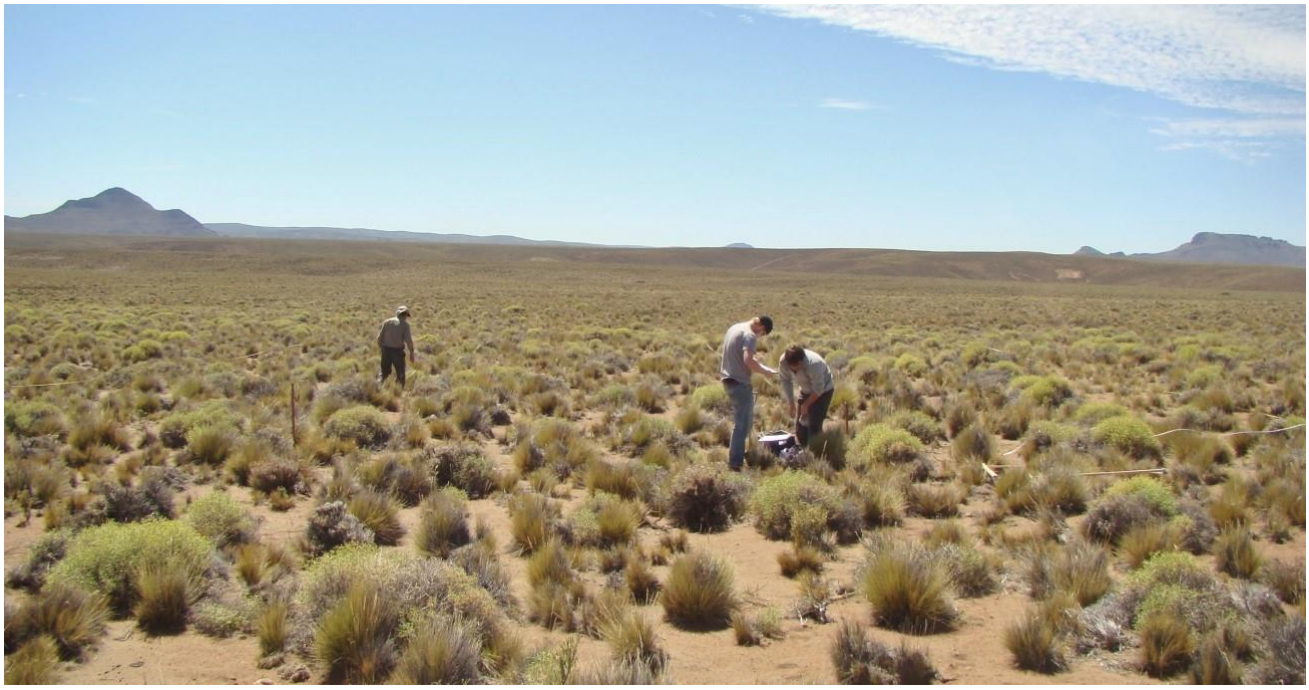
Julien Bénédit

BENEDIT Julien

Année universitaire 2021-2022
DUT STID / 2A

STAGE DE FIN D'ETUDE / DUT STID

Développement et validation de méthodes de comblement de lacunes de mesures (gapfilling) sur les traits de plantes.



Enseignant référent :
CISSE Papa Ousmane

Tuteurs organisme d'accueil :
MARTIN Raphaël et GROSS Nicolas

Remerciements

Je tiens à remercier madame Catherine PICON-COCHARD, la directrice de l'Unité Mixte de Recherche sur l'Ecosystème Prairial (UMR UREP) de l'INRAE de Clermont-Ferrand pour m'avoir accueilli au sein de son unité.

Merci à mes tuteurs ; Raphaël MARTIN, pour sa disponibilité, son accompagnement constructif et son aide si précieuse et Nicolas GROSS de m'avoir accordé toute sa confiance et pour son partage de connaissances.

À Laurence BENEDIT, un sincère merci pour son chaleureux accueil qui m'a permis une intégration rapide au sein de l'équipe.

À tous les autres stagiaires avec lesquels j'ai partagé de véritables moments de complicité, un grand merci pour leur bienveillance et leurs échanges de grande qualité.

Plus généralement, je remercie tout le personnel de l'UREP de l'INRAE de Clermont-Ferrand-Crouël pour m'avoir accueilli avec gentillesse et humanité.

Merci à mon enseignant référent, Papa Ousmane CISSE pour son aimable soutien.

Et, enfin, je tiens à remercier vivement tous les professeurs de l'IUT d'Aurillac pour les connaissances transmises lors de mes 2 années de formation.

Résumé et mots clés

Dans le cadre de ma deuxième année du DUT STID, Statistique et Informatique Décisionnelle d'Aurillac, j'ai effectué mon stage de fin d'études à l'INRAE (Institut National de Recherche pour l'Agriculture, l'Alimentation et l'Environnement) de Clermont-Ferrand. J'ai travaillé au sein de l'Unité mixte de Recherche sur l'Ecosystème Prairial (UREP). Ce stage s'inscrivait dans un projet européen s'intéressant aux liens entre la biodiversité des zones arides, le fonctionnement des écosystèmes et leurs réponses aux changements climatiques ainsi qu'au pâturage. L'objectif de ce stage était de trouver des méthodes statistiques permettant de déterminer les valeurs manquantes (gapfill) de traits de plantes et leurs incertitudes.

Tous ces traitements et analyses de données ont été réalisés grâce au logiciel de statistique R.

Mots clés

Base de données, dryland, écologie, fichier, gapifiller, graphique, moyenne, pourcentage d'erreur, projet Biodesert, quantile, script R, trait fonctionnel.

Table des matières

I - Identité de la structure d'accueil et environnement de travail 8
I.1 – Origines et évolution de l'INRAE 8
I.2 – L'Unité de Recherche sur l'Ecosystème Prairial 8
I.3 – La vie du laboratoire 10
II - Mon activité au sein de l'entreprise. 11
II.1 – Les missions confiées 11
II.2 – Mise en place d'une stratégie de réponse à la problématique 15
II.3 - Réalisation du script R 19
II.4 - Analyse de données 22
Conclusion 31

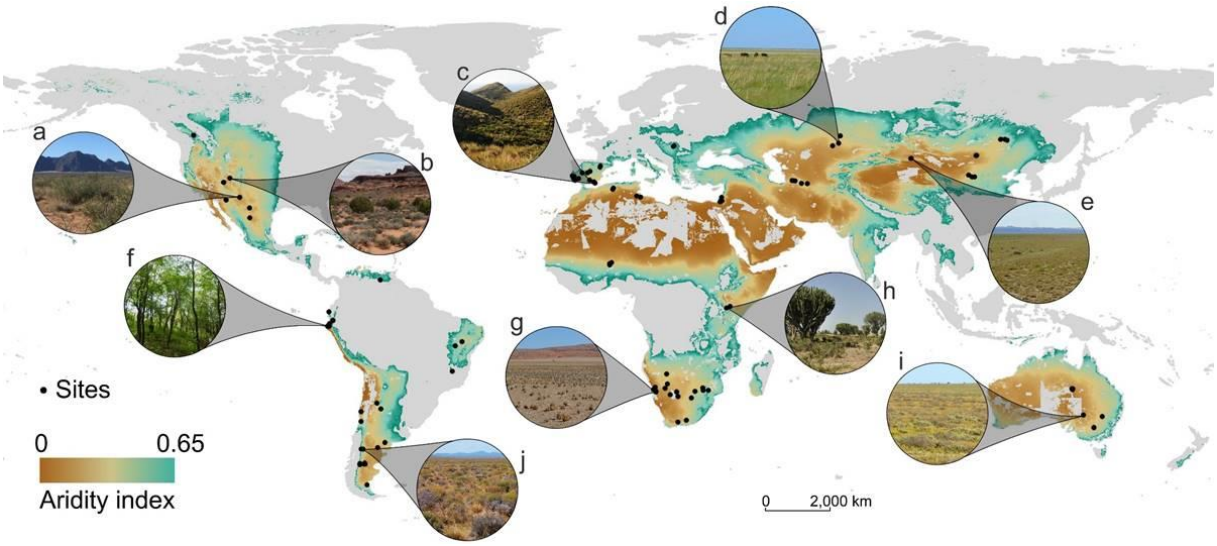
Introduction

Les zones arides et sèches

Les zones arides et sèches (dryland en anglais) recouvrent 41.3% de la surface terrestre. Ces zones sont caractérisées par un indice d'aridité inférieur à 0.65 (indicateur numérique impliquant la pluviométrie annuelle et l'évapotranspiration potentielle des plantes à un endroit donné), impliquant une productivité des plantes est limitée par la ressource en eau. Ces zones habitées par l'homme sont présentes majoritairement dans les pays du sud. Ces zones sont cependant sensibles à la désertification qui est un phénomène naturel et qui désigne la dégradation progressive des sols dans les zones arides. Cette désertification, encouragée par le changement climatique et l'activité humaine, peut engendrer la détérioration de la végétation, l'érosion des sols et finalement la migration de la population. Préserver ces écosystèmes représente donc un enjeu majeur pour ces populations afin d'éviter de voir leurs ressources quotidiennes, issues de la culture et de l'élevage du bétail ainsi menacées.

Le projet européen Biodesert

Avec comme sujet d'étude les zones arides de la planète et la désertification, le projet européen Biodesert (https://biodesert.maestrelab.com/) s'intéresse aux liens entre la biodiversité des zones arides, le fonctionnement des écosystèmes et leurs réponses aux changements climatiques ainsi qu'au pâturage. C'est un projet collaboratif financé par le Conseil Européen de la Recherche (ERC) qui a réuni plus de 60 équipes de recherche travaillant dans 26 pays afin d'étudier 98 sites expérimentaux. L'hypothèse centrale du projet est de soutenir que des écosystèmes abritant une biodiversité plus importante sont plus résistants aux effets des changements globaux tel que le changement climatique et le surpâturage.



La vaste répartition des sites d'études a permis de réaliser des observations sur un grand nombre d'écosystèmes différents, avec un échantillonnage sur un gradient d'aridité variant selon la situation géographique du site expérimental étudié. En effet, bien que ce soit tous des écosystèmes arides, l'indice d'aridité n'est pas le même que ce soit entre les steppes de Patagonie, les zones subdésertiques ou encore les forêts sèches, par exemple.

De plus, chaque site est composé de 3 à 4 parcelles pour lesquelles une contrainte de pâturage a été instaurée et expérimentée. Avec différentes intensités, allant de peu, moyennement ou beaucoup pâturé (et certaine fois pas du tout pour les sites ayant une quatrième parcelle) afin d'évaluer l'effet de celui-ci. C'est donc, au total, 336 parcelles dans lesquelles des mesures des espèces végétales présentes localement ont été réalisées afin de quantifier la biodiversité locale et ses liens avec le fonctionnement de ces écosystèmes.

Ces mesures ont pour objectif de déterminer la diversité des traits fonctionnels des plantes de façon intra et interspécifique (i.e. entre les individus de même espèce ou entre individus d'espèces différentes). Un trait fonctionnel est une caractéristique morphologique, physiologique ou phénologique qui est l'expression de l'évolution de la plante dans son écosystème. Étudier ces traits et leur structuration spatiale permet donc de cerner le fonctionnement de la biodiversité. L'intérêt est que la diversité fonctionnelle permet de prédire non seulement comment les plantes répondent à l'aridité ou au pâturage (quelle sont leur stratégies fonctionnelles) mais aussi comment en retour elles vont modifier le fonctionnement de l'écosystème.

Toutes les informations relevées ont été renseignées dans une base de données. Ainsi chaque plante observée (un individu) est renseignée sous la forme d'une ligne. Les colonnes, quant à elles, représentent différents facteurs et variables observés (e.g. l'espèce de la plante, ses traits fonctionnels, l'indice d'aridité, l'intensité de pâturage, la localisation et le numéro de la parcelle…).

Méthodes de comblement de lacunes de mesures

Lors de ce type de projet de recherche à grande échelle, il n'est pas rare que certains prélèvements ou analyses n'aient pas été effectués, du fait d'un oubli, d'un ennui technique sur le terrain ou encore d'un manque de budget. Ainsi, la base de données finale contient des données manquantes, que l'on qualifiera de trous. L'objectif de mon stage est le développement et la validation de méthode de comblement de lacunes de mesures (gapfilling) sur les traits de plantes. En effet, il peut être intéressant, dans certains cas, d'avoir une matrice de données complète. Après avoir rappelé l'importance de combler ces données manquantes, nous cherchons ensuite à déterminer la valeur de ces données et à estimer le pourcentage d'erreurs s'y affairant, en utilisant un outil statistique basique comme la moyenne. Il faudra ensuite déterminer l'efficacité de cette méthode en illustrant les résultats avec des graphiques.

Annonce du plan

Nous allons commencer ce rapport de stage par une présentation de la structure qui m'accueille, l'INRAE, et plus précisément l'UREP située à Clermont-Ferrand sur le site de Crouël, en évoquant son objet d'étude principal, l'écologie et de la biodiversité.

Nous commencerons par nous intéresser à la création et à la structuration du jeu de données. Ensuite nous allons définir plus précisément la mission de mon stage en expliquant quelle stratégie de réponse à la problématique a été mise en place, en schématisant le raisonnement imaginé qui sera à l'origine du script R. Nous dresserons, finalement, une analyse des résultats effectués sur les données dont nous disposons.

I - Identité de la structure d'accueil et environnement de travail

I.1 – Origines et évolution de l'INRAE

INRAE, Institut National de Recherche pour l'Agriculture, l'Alimentation et l'Environnement est un établissement public à caractère scientifique et technologique (EPST) français. Cet organisme de recherche publique est placé sous tutelle du Ministère de l'Enseignement Supérieur, de la Recherche et de l'Innovation (MESRI) et de celui chargé de l'Agriculture et de l'Alimentation (MAA). Cet institut est le résultat de la fusion, en 2020, de l'INRA, l'Institut National de la Recherche Agronomique et de l'Institut national de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture (IRSTEA).

Aujourd'hui ses recherches concernent trois domaines fortement imbriqués : l'alimentation, l'agriculture et l'environnement avec l'ambition de développer une agriculture à la fois compétitive, respectueuse de l'environnement, des territoires et des ressources naturelles, et mieux adaptée aux besoins nutritionnels de l'homme ainsi qu'aux nouvelles utilisations des produits agricoles.

Les principales vocations de l'INRAE sont de produire, publier et diffuser les connaissances scientifiques résultant de ses travaux de recherche et d'expertise et de mobiliser ces connaissances au service de l'innovation, de l'expertise et de l'appui aux politiques publiques. Cet institut recrute à la fois des chercheurs, des ingénieurs d'étude, des ingénieurs de recherche et des techniciens, auxquels s'ajoutent les stagiaires, doctorants et post-doctorants qui sont les bienvenus pour apporter du dynamisme à la recherche.

Le dépôt de brevets et la publication de papiers de recherche est au cœur de métier de l'Institut.

L'INRAE est également fortement engagé dans l'accompagnement des parcours professionnels individuels et des collectifs de travail. Chaque centre est d'ailleurs doté d'un responsable formation et d'une équipe dédiée.

I.2 – L'Unité de Recherche sur l'Ecosystème Prairial

Définition et missions de l'UREP

L'Unité mixte de Recherche sur l'Ecosystème Prairial (UREP) appartient au département ECOlogie et bioDIVersité des milieux forestiers, prairiaux et aquatiques (ECODIV) et est présente sur le site de Crouël INRAE Clermont Auvergne Rhône Alpes.

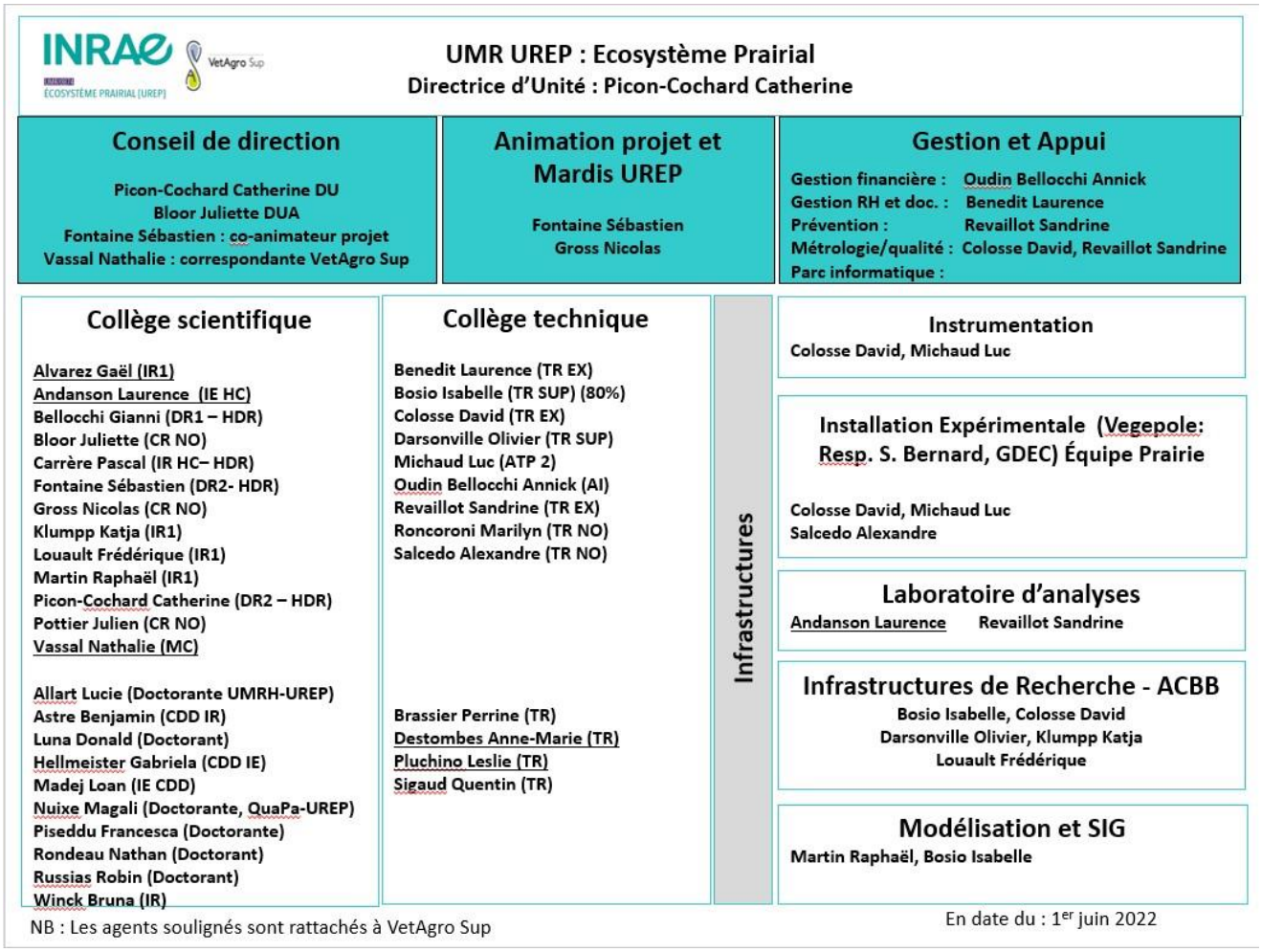
Elle possède une expertise internationale dans le domaine de l'écologie prairiale et plus particulièrement sur l'impact du changement climatique, les bilans de gaz à effet de serre, la séquestration de carbone et azote, les interactions plantes-sol (micro-organismes) et herbe-animal ou encore les effets des pratiques de gestion sur la dynamique prairiale.

L'UREP fait des campagnes de mesures et de prélèvements sur de parcelles pour théoriser l'intérêt de la biodiversité dans la résistance des espèces aux changements globaux.

Afin de développer des approches prédictives, l'unité s'appuie sur la modélisation.

Chiffres et organigramme de l'UREP

L'UREP compte 22 agents permanents, dont 12 chercheurs et ingénieurs, 1 maître de conférences, 1 assistant ingénieur, 7 techniciens, 1 adjoint technicien et 15 agents non titulaires. Les agents non titulaires ne prennent pas en compte les stagiaires qui sont au moins une dizaine.



Organigramme de l'UREP de juin 2022.

I.3 – La vie du laboratoire

Les outils informatiques

J'ai eu à ma disposition, lors de ma période de stage, un ordinateur portable sur lequel était installée une série de logiciels, dont RStudio qui m'a servi à développer mes scripts en R, ainsi que Skype Entreprise afin de participer aux réunions en distanciel. J'ai, de plus, eu accès à la suite office. L'ensemble de ces outils sont liés à ma session LDAP INRA (identifiant unique créé à mon arrivée et pour toute la durée du stage).

L'UREP dispose d'un site Intranet, qui est une interface à la disposition de l'ensemble des agents permettant d'accéder facilement, via des raccourcis, aux ressources essentielles. Parmi les plus importants et ceux que j'ai principalement utilisés, il existe l'outil #TEMPS, qui est le site de gestion des pointages et des congés.

Dès mon arrivé il m'a été confié un badge à mon nom que je dois présenter, lors de mes entrées et sorties de l'unité, à une borne ; ainsi mes heures sont retranscrites sur le site #TEMPS. De plus, je peux consulter mes heures effectuées chaque jour.

Ce badge m'a servi également pour effectuer le paiement de mes repas, puisqu'une cantine est disponible sur le site et est affiliée à l'interface SO HAPPY, permettant d'alimenter le compte.

De plus, j'ai travaillé sur une forge informatique propre à l'UREP, qui est un environnement de travail collaboratif sur lequel chaque participant du projet peut apporter des modifications à celui-ci rendant le partage et le développement plus facile. Cette forge est une forge institutionnelle INRAE déployée via l'outil GitLab.

Il existe encore bien d'autres outils que je n'ai pas eu l'occasion d'utiliser, comme pour la réservation de véhicule ou de salles mais aussi l'accès à des archives ou des publications…

Activités collectives

Les mardis de l'UREP sont des rendez-vous hebdomadaires lors desquels les agents, ainsi que les stagiaires, sont invités à présenter leurs travaux/projets devant l'unité. J'ai moi-même eu l'occasion d'y participer, en tant que spectateur mais également en tant que présentateur.

J'ai eu la chance de participer à une Assemblé Générale de l'unité qui est l'occasion de faire une mise au point trimestrielle de la vie de l'unité. C'est une réunion lors de laquelle des sujets importants sont abordés, comme la présentation des comptes de l'unité ainsi que les dépenses réalisées sur différentes périodes. Les membres de la réunion sont libres et encouragés de faire des réclamations par des prises de paroles spontanées. Des bilans sur les projets réalisés et en cours sont effectués. Par exemple, le nombre de publications des articles dans des revues scientifiques à comité de lecture (ACL) est comparé aux chiffres des années précédentes.

J'ai dû suivre une formation à la sécurité, puisque l'UREP est une unité contenant un laboratoire avec des produits chimiques. Cela exige donc de connaitre et d'appliquer les règles de sécurité à suivre en sein de celui-ci. Il existe de plus des modules numériques, avec les consignes et les conseils à appliquer.

II - Mon activité au sein de l'entreprise.

II.1 – Les missions confiées

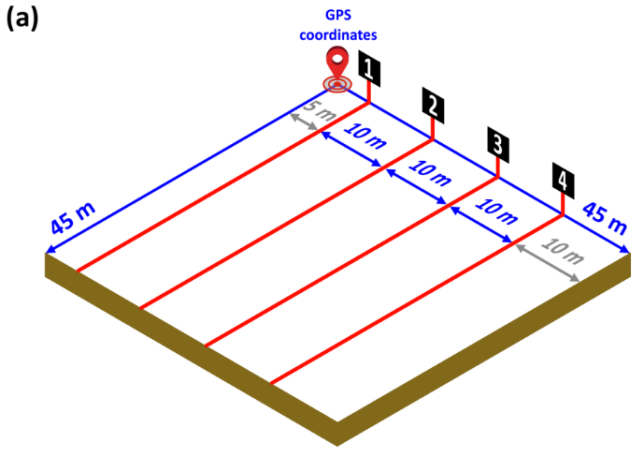
Le Projet Biodesert

Dans un premier temps il m'a fallu me familiariser avec ma mission et notamment avec le contexte dans lequel elle s'inscrit. Mon stage se situe, en effet, dans le domaine informatique mais s'applique, également, dans le cadre d'un projet de recherche complexe à l'origine de la création d'une vaste base de données. De ce fait, il était primordial de comprendre comment le protocole d'échantillonnage de ce projet a été réalisé et de commencer à analyser les données sur lesquelles j'allais travailler avant de les manipuler.

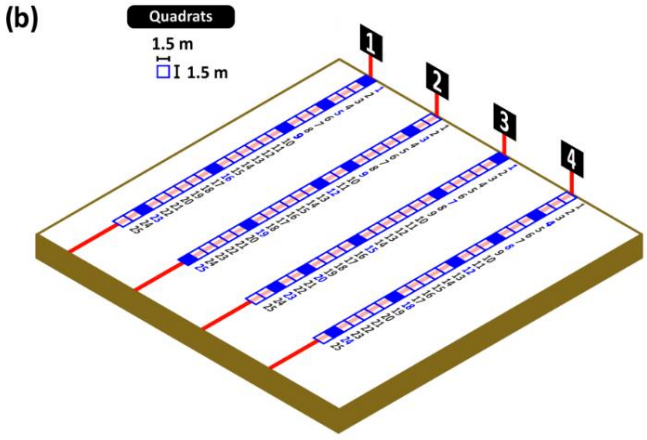
Plan d'échantillonnage

Les mesures sur le terrain se sont déroulées entre 2016 et 2018. Pour un site d'étude il existe trois à quatre parcelles (avec un gradient local d'intensité de pâturage différent). Dans une parcelle il y a quatre transects, eux-mêmes composés de vingt-cinq quadrats, c'est donc au total cent quadrats d'une dimension de 1,5 mètre carré, présents par parcelle. Pour chaque parcelle, vingt quadrats ont été sélectionnés aléatoirement afin de réaliser une mesure de toutes les espèces végétales présentes localement dans ceux-ci.

Pour chaque espèces présentes l'individu le mieux développé a été sélectionné et mesuré. On a mesuré sa taille (plant Height, H, cm), sa largeur (Lateral Spread, LS cm²). On a ensuite prélevé sur chaque individu des feuilles, exposées au soleil, matures et sans trace d'herbivorie ou de maladie. Sur ces feuilles plusieurs mesures ont été réalisées : la longueur des feuilles (Leaf Length, LL cm), la surface (Leaf Area, LA cm²) ainsi qu'au laboratoire des mesures de teneur en matière sèche (Leaf Dry Matter Content, LDMC, $g.g^{-1}$) et de surface spécifique foliaire (Specific Leaf Area, SLA, $cm^2.g^{-1}$), deux traits liés à la capacité de croissance des plantes et à leur résistance aux herbivores et à l'aridité.



Quatre transects par parcelle.



Vingt-cinq quadrats par transect.

De la donnée brute à la donnée élaborée

Avant d'arriver dans l'unité, un travail en amont avait déjà était réalisé par mes tuteurs. En effet, à l'origine les données issues du projet Biodesert étaient fournies de façon éparse dans différents dossiers sous formes de nombreux fichiers au format 'xlsx'. Afin de faciliter leur utilisation, ils ont décidé de rédiger différents scripts R afin de les regrouper.

C'est un processus de traitement qui a engendré la création de nouveaux fichiers, cette fois-ci au format 'RDS'. C'est donc un travail assez conséquent qui a été effectué et que j'ai dû m'approprier, afin de prendre connaissance de tout ce qui avait déjà était fait et pour, par la suite, pouvoir travailler sur une base solide que je maitrise. L'opération de traitement de fichiers peut être résumée par le schéma ci-dessous :

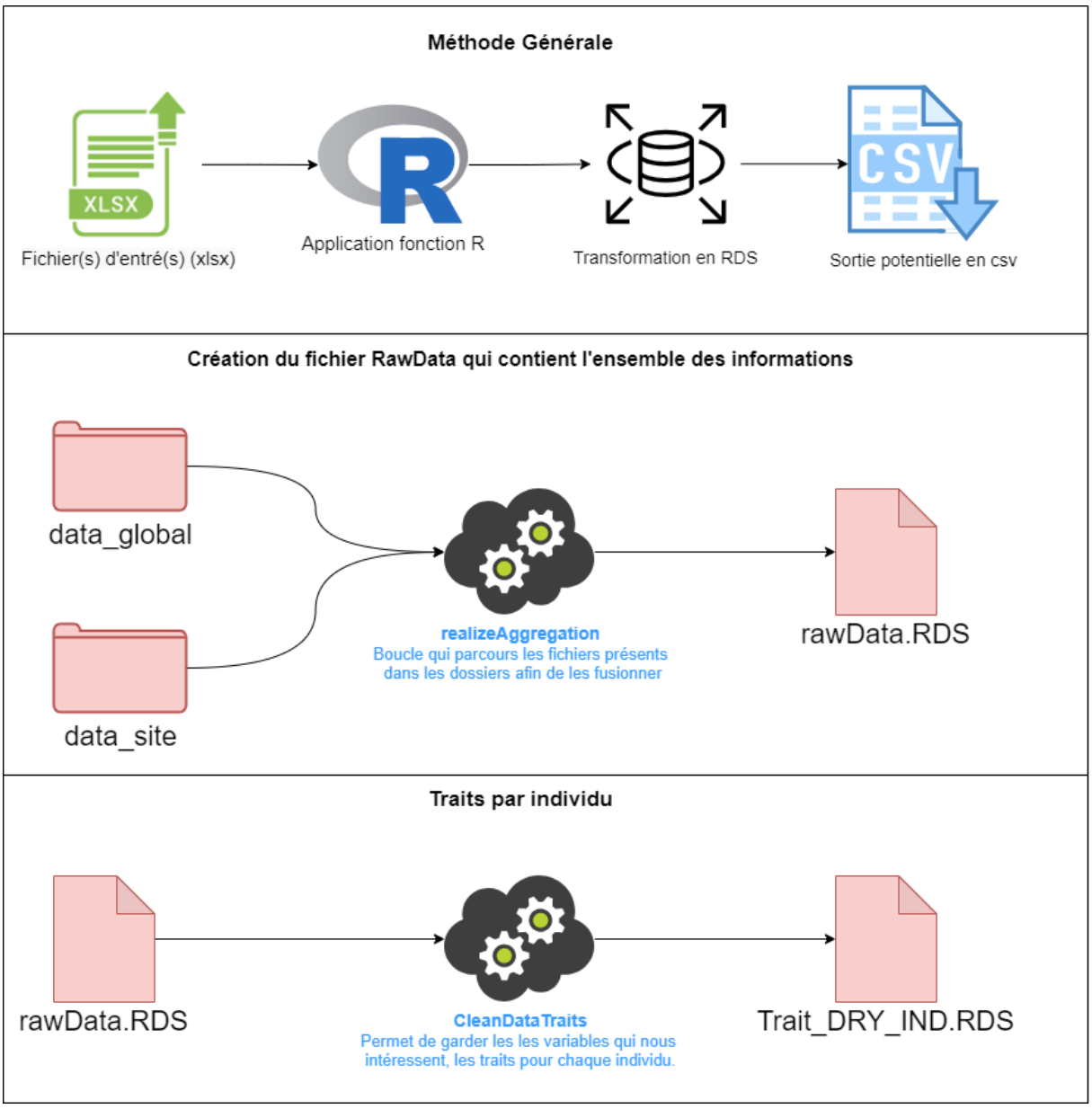


Schéma récapitulatif du traitement de données.

Analyse du jeu de données

Parmi les nouveaux fichiers produits, celui qui m'intéresse est *'Trait_DRY_IND.RDS'*. Il contient des informations sur 21 374 individus, chacun sous forme d'une ligne. Il existe 122 variables descriptives, plus ou moins importantes, en colonne, qui décrivent l'individu. Celles qui nous intéressent sont principalement les traits fonctionnels mais aussi l'indice d'aridité et l'intensité de pâturage qui sont toutes des variables quantitatives. Il y a aussi d'autres paramètres tels que le nom de l'espèce auquel appartient l'individu plante échantillonné, sa localisation géographique (pays, site, plot, transect, quadrat), qui eux correspondent à des variables qualitatives.

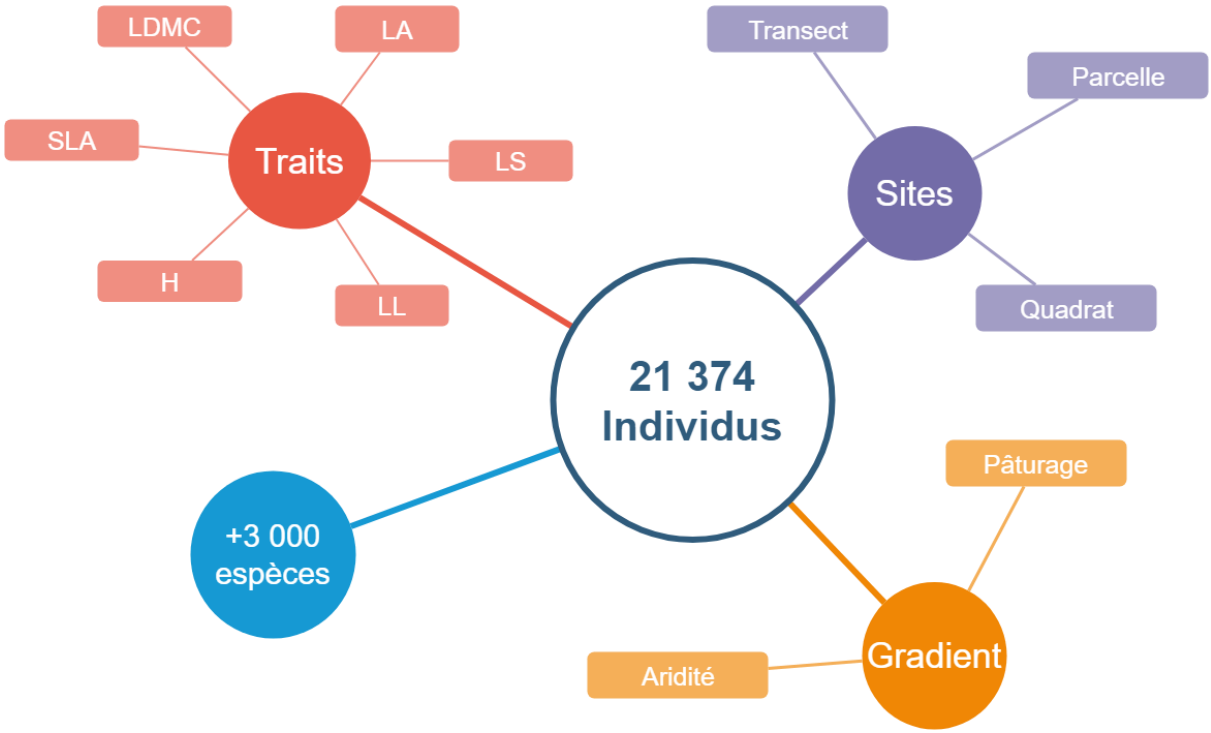


Diagramme décrivant notre jeu de données.

Puisqu'on souhaite gapfiller les traits fonctionnels de notre jeu de données, il est important de repérer ces derniers dans celui-ci et de les définir. Ils sont au nombre de 6 et sont tous des variables quantitatives présents en colonne dans le jeu de données.

Mesure sur le terrain à même la plante :

- H : *Height*, est la hauteur de la plante. Elle est, par nature, très variable, aussi bien entre deux espèces différentes (e.g. arbre vs. graminée) qu'au sein de la même espèce (un individu juvénile vs. un individu à sa taille maximal).
- LS : *Lateral Spread*, un indicateur de la surface de la plante. I est déterminé en mesurant le diamètre dans deux directions orthogonales depuis le centre de la plante. C'est une multiplication entre les deux diamètres, le résultat est une valeur assez élevée puisque c'est un produit.
- LL : *Leaf Lenght*, est simplement la mesure de la feuille la plus longue pour chaque individu.

Mesure en laboratoire sur une ou plusieurs feuilles sélectionnées :

- LA : *Leaf Area*, est la surface foliaire, c'est l'espace occupé par la feuille.
- LDMC : *Leaf Dry Matter Content*, que nous pouvons traduire par la teneur en matière sèche des feuilles. Elle se calcule par un ratio entre la masse sèche et la masse fraiche de la feuille.
- SLA : *Specific Leaf Area*, ou, en français, surface foliaire spécifique (SFS) est quantifiée par le ratio entre la surface foliaire et la masse sèche de la feuille.

Informations complémentaires

Maintenant que nous savons comment est structuré notre jeu de données, nous aimerions en savoir davantage sur les valeurs que peuvent adopter chacun des traits fonctionnels.

Depuis RStudio nous pouvons utiliser la commande 'summary()' dans un premier temps pour obtenir des informations intéressantes comme la moyenne et la médiane. Summary communique également une information sur les NA's (Not Available) qui sont les valeurs manquantes pour chaque trait.

De plus, il existe un package R servant spécifiquement à la visualisation des données manquantes. Celui-ci nous permet d'obtenir des informations supplémentaires plus rapidement sans avoir à faire le calcul du pourcentage ou de la différence.

```
23  install.packages("naniar") ; library(naniar)
24  #Installation et chargement du package naniar.
25
26  pct_miss() #Pourcentage de NA.
27  n_miss() #Nombre de NA.
28  n_complete() #Nombre !NA.
```

Exemple d'utilisation du package 'naniar'.

Présentation des traits fonctionnels et résumé de leurs informations :

	H	LS	LL	SLA	LDMC	LA
Moyenne	47.28	13 262	8.806	125.309	0.377	4.407
Médiane	21.00	300	5.000	110.984	0.359	0.905
Variance	8570.63	8.69^{e}+9	114.62	5839.13	0.025	499.31
Nombre NA	584	613	1303	4359	6478	3210
NA en %	2.732	2.867	6.096	20.393	30.307	15.018
Obs. sans NA	20 790	20 761	20 071	17 015	14 896	18 164

Tableau avec les informations importantes de chaque trait fonctionnel.

Nous observons qu'il y a plus de données manquantes pour les mesures réalisés en laboratoire.

II.2 – Mise en place d'une stratégie de réponse à la problématique

L'intérêt de gapfiller

Maintenant que nous en savons plus sur les variables sur lesquelles nous allons travailler et notamment la quantité de valeurs manquantes pour chaque trait fonctionnel, il est important de rappeler pourquoi nous souhaitons les combler. Il existe 3 raisons principales de vouloir avoir un jeu de données complet :

- Valorisation de l'information existante : en effet, si nous décidons d'ignorer ces valeurs manquantes, cela revient à réduire notre jeu de données en supprimant les lignes incomplètes puisqu'elles seront inutilisables lors de futures analyses statistiques (on utilise uniquement les lignes complètes). Ce sont donc de nombreux individus retirés pour quelques traits manquants.
- Connaitre l'erreur de mesure : gapfiller nous permet quantifier l'erreur dû à la variabilité intraspécifique et de déterminer la différence entre les traits.
- Connaitre l'origine de l'erreur afin de la minimiser : un intérêt important est de mieux caractériser les sources de variabilités et donc d'erreurs.

Combler par la moyenne

Il existe une multitude de méthodes afin de combler ces valeurs manquantes. Une première approche est de combler les NA par le calcul de la moyenne du trait des individus de la même espèce. Cette méthode est basique mais a l'avantage de pouvoir facilement être facilement mise en œuvre et implémentable sous RStudio. Cependant nous ne savons pas si celle-ci est efficace et si c'est le cas à partir de combien d'individus de référence devient-elle applicable ?

Individu	Espèce	Trait
1	A	10
2	A	20
3	A	50
4	A	NA

$$Trait_4 = \frac{10 + 20 + 50}{3} \cong 26{,}66$$

Simulation avec la moyenne pour quantifier l'erreur

Comme présenté précédemment, nous souhaitons être capable de déterminer l'efficacité de la méthode employée en estimant un pourcentage d'erreurs en utilisant celle-ci. Pour évaluer les méthodes, nous réalisons une simulation de gapfilling sur des valeurs connues afin d'observer et de comparer les résultats obtenus. Nous appliquerons donc, dans un premier temps, ce processus en gapfillant par la moyenne.

Processus de gapfiller par la méthode de la moyenne

Avant de se lancer dans la conception d'un script R, nous pouvons d'abord schématiser le raisonnement à suivre pour mieux déterminer les principales étapes qu'il faudra par la suite réaliser.

- Étape 1 : Suppression des individus ayant une valeur non-renseignée :

Individu	Espèce	Trait
1	A	10
2	A	20
3	A	50
4	A	NA

Individu	Espèce	Trait
1	A	10
2	A	20
3	A	50

-Étape 2 : Calcul de l'erreur pour un individu de référence : $E = 100 * \frac{mean(Référent(s)) - Observé}{Observé}$

Individu	Espèce	Trait
1	A	10
2	A	20
3	A	50

Individu	Espèce	Trait
1	A	NA
2	A	20
3	A	50

Individu	Espèce	Trait
1	A	10
2	A	NA
3	A	50

Individu	Espèce	Trait
1	A	10
2	A	20
3	A	NA

$$E_1 = 100 * \frac{|mean(50) - 10|}{10} \qquad E_1 = 100 * \frac{|mean(10) - 20|}{20} \qquad E_1 = 100 * \frac{|mean(10) - 50|}{50}$$

$$E_k = 100 * \frac{|mean(50) - 10|}{10} \qquad E_k = 100 * \frac{|mean(50) - 20|}{20} \qquad E_k = 100 * \frac{|mean(10) - 50|}{50}$$

$$E_N = 100 * \frac{|mean(20) - 10|}{10} \qquad E_N = 100 * \frac{|mean(10) - 20|}{20} \qquad E_N = 100 * \frac{|mean(20) - 50|}{50}$$

$$\bar{E} = \frac{\big((400 + 400 + 100) + (50 + 150 + 50) + (80 + 80 + 150)\big)}{9} \cong \mathbf{162\%}$$

Calcul du pourcentage d'erreur

Nous parcourons notre jeu de données, nous sélectionnons le premier individu, qui est l'individu observé. Nous supprimons artificiellement sa valeur afin de la combler par la suite. Nous regardons tous les individus de la même espèce que ce dernier, dans notre cas de figure c'est l'espèce A, parmi l'ensemble des individus de cette espèce, nous effectuons un tirage aléatoire sans remise pour tirer notre individu référent. Ce tirage est effectué cent fois afin d'obtenir une meilleure précision. Nous comparons l'individu référent, tiré aléatoirement, à celui observé. L'objectif étant de quantifier la différence entre les deux valeurs grâce au calcul. Puis nous répétons ce processus pour le prochain individu et ainsi de suite.

-Étape 3 : Même chose mais cette fois-ci pour deux individus de référence.

Individu	Espèce	Trait
1	A	10
2	A	20
3	A	50

Individu	Espèce	Trait
1	A	NA
2	A	20
3	A	50

Individu	Espèce	Trait
1	A	10
2	A	NA
3	A	50

Individu	Espèce	Trait
1	A	10
2	A	20
3	A	NA

$$E_1 = 100 * \frac{|\text{mean}(50;20) - 10|}{10} \qquad E_1 = 100 * \frac{|\text{mean}(50;10) - 20|}{20} \qquad E_1 = 100 * \frac{|\text{mean}(10;20) - 50|}{50}$$

$$E_k = 100 * \frac{|\text{mean}(50;20) - 10|}{10} \qquad E_k = 100 * \frac{|\text{mean}(50;10) - 20|}{20} \qquad E_k = 100 * \frac{|\text{mean}(10;20) - 50|}{50}$$

$$E_N = 100 * \frac{|\text{mean}(50;20) - 10|}{10} \qquad E_N = 100 * \frac{|\text{mean}(50;10) - 20|}{20} \qquad E_N = 100 * \frac{|\text{mean}(10;20) - 50|}{50}$$

$$\bar{E} = \frac{(250 + 250 + 250) + (50 + 50 + 50) + (70 + 70 + 70)}{9} \cong \mathbf{123}\%$$

Un individu référent supplémentaire

Le processus reste identique, nous parcourons progressivement le jeu de données afin de choisir l'individu observé, seulement cette fois-ci nous allons tirer aléatoirement deux individus référents de la même espèce. En effet, c'est seulement le calcul de la moyenne des individus référents qui change, pour lequel cette fois-ci, ce sont 2 individus référents qui ont été tirés aléatoirement dans notre jeu de données. Nous procédons de la même manière, seulement nous calculons progressivement pour un nombre d'individus de référence allant de 1 à 40. Puisque logiquement, le pourcentage d'erreur devrait baisser avec une plus grande sélection. C'est en effet le cas dans notre exemple : $\overline{E_1} > \overline{E_2}$. De plus, cela nous permet de savoir à partir de quand celui-ci se stabilise.

II.3 - Réalisation du script R

Fonction principale pour calculer le pourcentage d'erreurs de la méthode moyenne.

On souhaite désormais implémenter sous R l'algorithme présenté précédemment. Pour cela nous allons développer une série de fonction afin de factoriser le code :

```r
NomFonction <-function(paramètre1,paramètre2...){

  #entre les accolades on rédige le contenu de notre fonction.
  #cela peut être des commandes R sous forme de calcul ou autre.


  return(Résultat) #On indique la variable de résultat qu'on souhaite renvoyer.
}

#une fois la fonction créée et compilée  on peut l'utiliser:
NomFonction(paramètre1,paramètre2)
```

Exemple d'architecture globale d'une fonction R.

Nous envisageons donc la création d'une fonction, qui, depuis notre jeu de données et selon n'importe quel trait fonctionnel souhaité, calcule, de façon automatique le pourcentage d'erreurs en suivant chacune des étapes imaginées.

La première partie de notre fonction, l'initialisation, est constituée principalement de plusieurs manipulations sur le jeu de données :

```r
funErrorRate <- function(pathDataFile, nbRandMax, varToGapfill, nbIndMax){

  dfTraitInd <- readRDS(pathDataFile)
  #Chargement de notre fichier avec les données.

  indexVarToGapfill <- which(colnames(dfTraitInd) == varToGapfill)
  #Nous renseignons l'indice du nom du trait qu'on souhaite calculer.

  dfTraitInd <- dfTraitInd[which(!is.na(dfTraitInd[,indexVarToGapfill])),]
  #Nous retirons les lignes ayant des valeurs manquantes dans la variable a gapfiller.

  dfTraitInd$completeSpeciesName <- paste0(as.character(dfTraitInd$Genus_Final),
    "_",as.character(dfTraitInd$Species_Final))
  #Ajout de la colonne 'complteteSpeciesName' au jeu de données, addition de deux colonnes qualitatives.

  tableOccurenceSpecies <- table(dfTraitInd$completeSpeciesName)
  #Table avec le nombre d'individu par espèce.

  indexCompleteSpeciesName <- which(colnames(dfTraitInd)=="completeSpeciesName")
  #Nous recuperons l'indice de la colonne 'complteteSpeciesName'.

  dfTraitInd$NbOccurenceSpecies <- apply(X = dfTraitInd, MARGIN = c(1),
    FUN = function(x) as.numeric(tableOccurenceSpecies[as.character(x[indexCompleteSpeciesName])]))
  #Nous ajoutons une nouvelle colonne avec le nombre de fois que l'espèce apparait pour chaque individu.

  dfTraitInd <- dfTraitInd[which(dfTraitInd$NbOccurenceSpecies>1),]
  #Nous supprimons les individus avec une seule observation.

  dfError <- data.frame(matrix(ncol= nbIndMax, nrow = nrow(dfTraitInd)*nbRandMax))
  #Création d'un jeu de données vide de taille définie pour accueilir nos futurs résultats.
```

Première partie de notre fonction, « Initialisation ».

Cette première étape consiste à l'importation du jeu de données, puis à la préparation de celui-ci en supprimant les lignes contenant des NA pour le trait souhaité et d'autres manipulations.

Nous pouvons désormais rédiger la seconde partie de notre fonction, qui est cette fois-ci réservée au calcul du pourcentage d'erreurs que nous souhaitons estimer. Cette partie n'est pas forcément plus longue mais introduit le concept de boucle.

```
52  for(indexIndiv in 1:nrow(dfTraitInd)){ #Première boucle qui parcourt le jeu de données pour chaque individu.
53
54    indexSameSpecies <- which(dfTraitInd$completeSpeciesName[indexIndiv] == dfTraitInd$completeSpeciesName)
55    #Nous déterminons la place des individus appartenant à la même espèce.
56
57    indexSameSpecies <- indexSameSpecies[- which(indexSameSpecies==indexIndiv)]
58    #Nous retirons l'individu concerné.
59
60    for(nbIndivRef in which(1:(dfTraitInd$NbOccurenceSpecies[indexIndiv]-1) <= nbIndMax)){
61    #Seconde boucle en fonction du nombre d'individu maximum - 1 par espèce.
62
63      tabError <- numeric(length = nbRandMax)
64      #Création d'une table pour accueillir  les erreurs de taille nbRandMax.
65
66      for(randomIndex in 1:nbRandMax){ #Boucle en fonction du nombre d'individu référent.
67
68        tabIndexRandom <- sample(size = nbIndivRef, replace = FALSE, x = indexSameSpecies)
69        #Tirage aléatoire sans remise, des individus réfs parmi tous ceux dispo.
70
71        gapfilledVariable <- mean(x = dfTraitInd[tabIndexRandom,indexVarToGapfill])
72        #Moyenne des individus référents de notre tirage.
73
74        tabError[randomIndex] <- 100 * abs(gapfilledVariable - dfTraitInd[indexIndiv,indexVarToGapfill])/
75          dfTraitInd[indexIndiv,indexVarToGapfill]
76        #Nous ajoutons dans notre table le résultat des calculs -> E = 100 * abs(meanRef - obs)/obs
77      }
78      dfError[((indexIndiv-1)*nbRandMax+1):(indexIndiv*nbRandMax), nbIndivRef] <- tabError
79      #Imputation de chaque valeur de tabError dans notre jeu de données vide créé auparavant.
80    }
81  }
82  return(dfError) #La fonction renvoi le jeu de données complété par les pourcentages d'erreur.
83 }
```

Deuxième partie de notre fonction, calcul des pourcentages d'erreur.

Les différentes boucles permettent de répéter une opération de façon automatique un certain de nombre de fois spécifié. Elles parcourent le jeu de données pour sélectionner successivement un individu pour lequel le pourcentage d'erreur est calculé avec un nombre d'individus référent croissant.

Enfin, il ne nous reste plus qu'à exécuter notre fonction sans oublier de spécifier les paramètres, à savoir :

```
85  dfResError <- funErrorRate(pathDataFile = traitsPerIndividualPathRDS, nbRandMax = 100,
86                             varToGapfill = "LDMC", nbIndMax = 40)
```

Exécution de la fonction.

- pathDataFile : sert à indiquer l'emplacement de notre jeu de données.
- nbRandMax : nombre de répétition du tirage aléatoire des individus référents.
- nbIndMax : le nombre d'individus référents maximum tiré aléatoirement.
- varToGapfill : le nom du trait fonctionnel que nous souhaitons étudier.

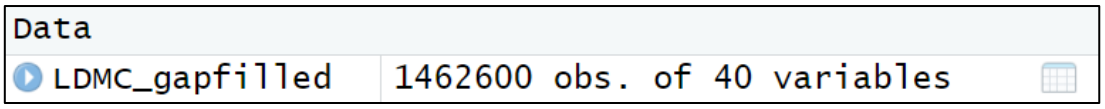
Fonction complémentaire

Nous avons désormais une fonction opérationnelle, qui renvoie le résultat souhaité attendu. Cependant, celle-ci est très longue à exécuter et il faut la relancer pour chacun des traits. Nous rédigeons alors une nouvelle fonction complémentaire afin d'optimiser et d'automatiser le processus, permettant ainsi d'avoir les résultats enregistrés dans des fichiers RDS.

Seconde fonction automatisée afin d'enregistrer les résultats.

```
computeErrors <- function(pathDataFile, nbRandMax, varToGapfill, nbIndMax, returnList = TRUE, saveAsRDS = FALSE){

  listErrors <- list()
  #Création d'une liste vide pour acueillir nos résultats.

  for (varToFill in varToGapfill){ #Boucle qui parcourt notre liste de traits.

    dfTemp <- funErrorRate(pathDataFile, nbRandMax, varToFill, nbIndMax)
    #Execution de notre fonction principale pour le trait concerné.

    if(returnList == TRUE){
      listErrors[[varToFill]] <- dfTemp
      #Attribution du résultat dans la liste.
    }
    if(saveAsRDS == TRUE){
      saveRDS(object = dfTemp, file = paste0(rdsFolderPath,"/",varToFill,".RDS"))
      #Sauvergarde du jeu de données en format RDS.
    }
  }
  if(returnList == TRUE){
    return(listErrors)
  }
}
computeErrors(pathDataFile = traitsPerIndividualPathRDS, nbRandMax = 100,
              varToGapfill = c("H", "LS", "LL", "SLA", "LDMC", "LA", "volume", "LT"),
              nbIndMax = 40, saveAsRDS = TRUE)
```

Ce sont les mêmes paramètres qui sont utilisés puisque notre fonction principale est exécutée dans celle-ci. Cependant cette fois-ci, nous pouvons spécifier l'intégralité de nos traits fonctionnels grâce à la boucle qui les sélectionne un par un. De plus, nous indiquons '*saveAsRDS = TRUE'* pour obtenir une sortie au format RDS de nos fichiers. Après son exécution, nous obtenons donc six nouveaux jeux de données, qui sont composés des pourcentages d'erreurs associés à chacun d'eux après le calcul de la simulation de gapfilling.



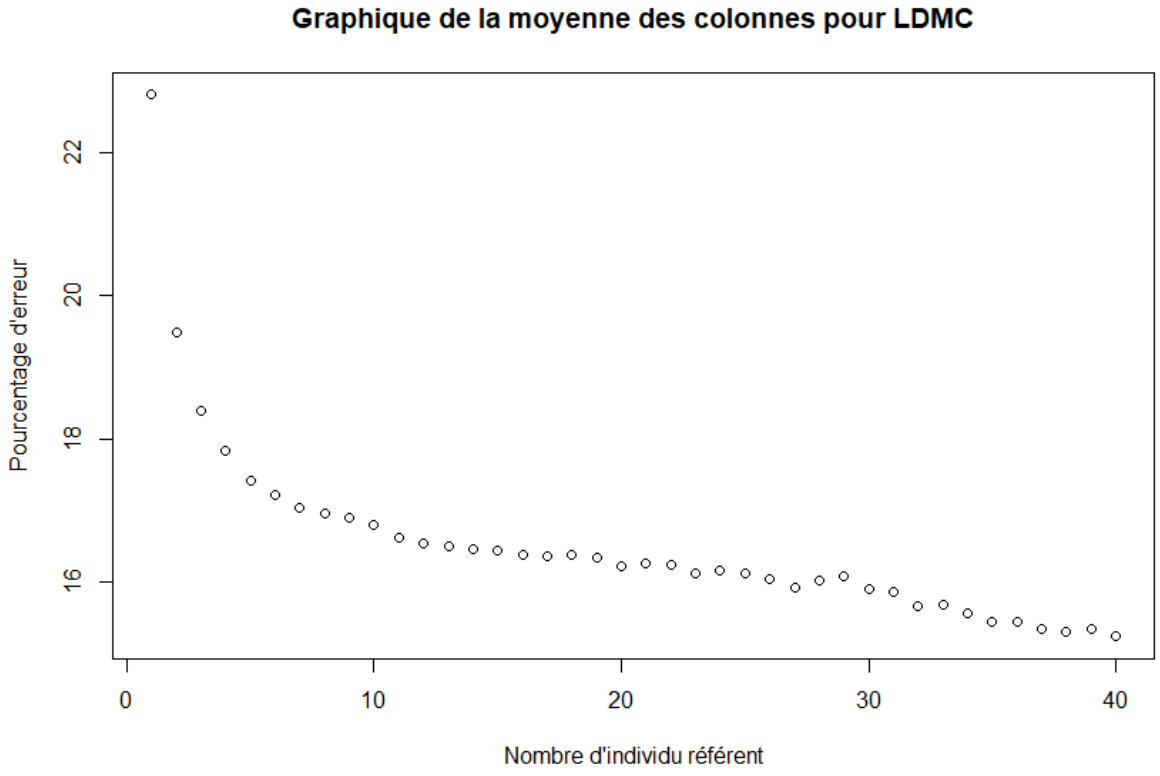
Exemple de la composition d'un fichier de sortie. (LDMC)

Nous retrouvons en colonne le nombre d'individus référent tirés (allant d'un à quarante). En ligne, ce sont les pourcentages d'erreurs calculés pour un individu, calcul effectué cent fois pour chacun d'entre eux (*'nbRandMax'*).

II.4 - Analyse de données

Graphique de la moyenne des colonnes

Maintenant que nous avons un jeu de données avec les différents pourcentages d'erreurs pour chaque trait fonctionnel, il faut les exploiter de façon à les mettre en valeur. Nous souhaitons connaître le pourcentage d'erreurs en fonction du nombre d'individus de référence. Pour ce faire, nous calculons la moyenne de chaque colonne. A l'aide de la commande *'colMeans()'*, nous obtenons les résultats souhaités. Pour avoir une meilleure visualisation de ceux-ci, nous traçons un simple nuage de points grâce à la fonction *'plot()'* :



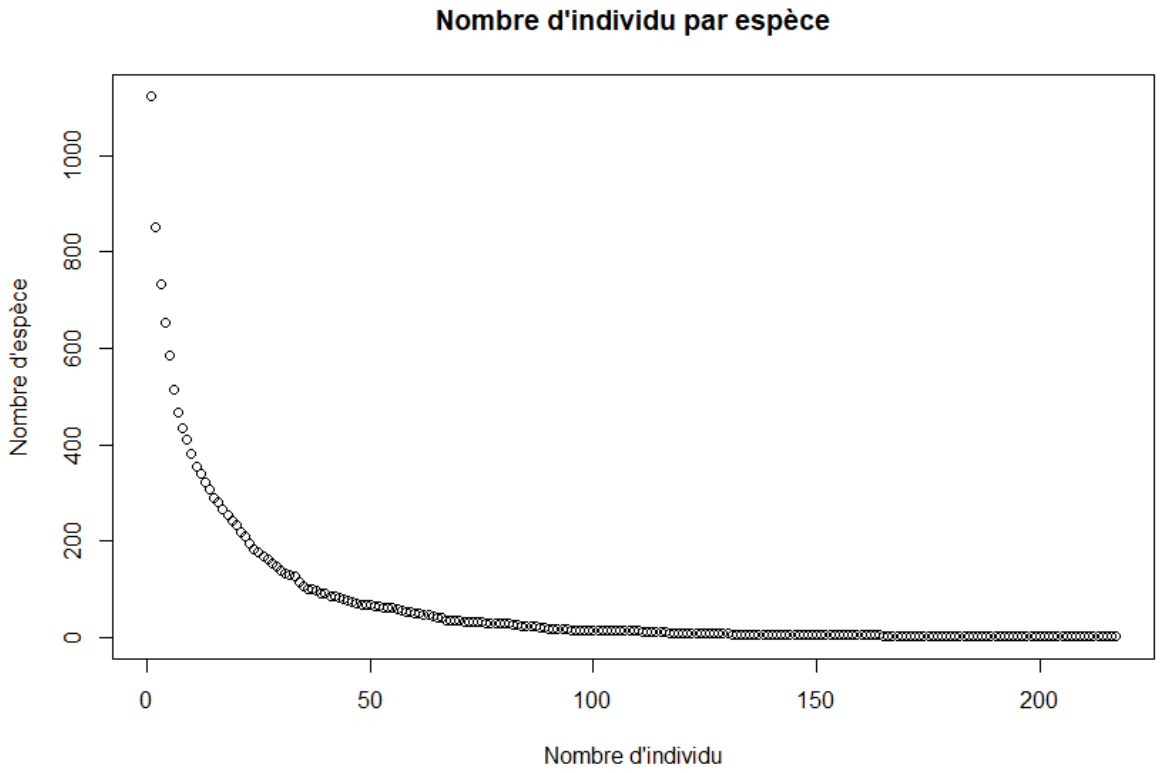
Exemple de graphique pour le trait LDMC.

Interprétations

Avec ce graphique, nous sommes déjà capables de réaliser plusieurs observations. En effet, nous avons notre pourcentage d'erreurs associé au nombre d'individu référents, ainsi nous sommes capables de vérifier la pertinence de notre méthode. De plus, nous savons à partir de combien d'individus référents notre erreur se stabilise. Nous en déduirons alors le nombre d'observations minimum nécessaire sur le terrain. Avec cet exemple, nous pouvons dire qu'à partir de cinq individus observés, nous avons un pourcentage d'erreurs satisfaisant moyen qui est de 17%.

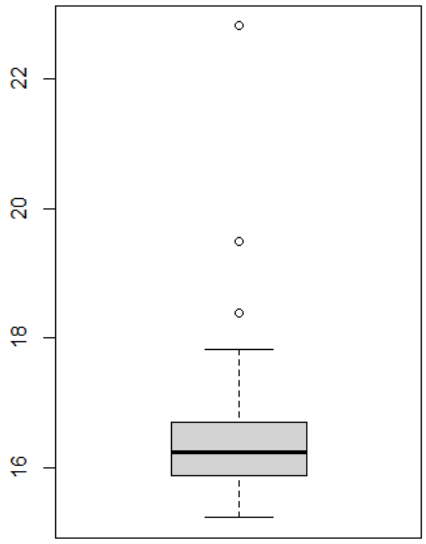
Graphique du nombre d'individus par espèce

Il est important de déterminer combien d'individus référents possède chaque espèce. Après calcul, la moyenne est de 13 individus. Nous pouvons également tracer une courbe pour un résultat plus visuel. Nous observons que la courbe décroit rapidement pour finalement se stabiliser. Cela permet de nous informer qu'au-delà d'un certain seuil c'est seulement une partie des espèces qui est concernée pour un nombre d'individus référents élevé. C'est-à-dire que sur le terrain, les individus d'une même espèce ont été relevés une dizaine de fois au plus, et seulement quelques cas rares, très présents, ont pu être observés un grand nombre de fois. C'est notamment grâce à ce graphique que le paramètre *'nbIndMax'* a été quantifié puisque nous nous rendons compte qu'au-delà de 40 individus de référence, une très faible partie des espèces est concernée.



Boxplot

Nous souhaitons avoir une version plus élaborée de notre graphique précédent en n'affichant plus uniquement la moyenne des erreurs mais aussi leur distribution. L'idéal pour cela serait d'obtenir un intervalle de confiance de notre erreur. Pour ce faire, nous pouvons réaliser un graphique à l'image d'un boxplot (boîte à moustache), cependant une série de 40 boxplot rendrait le graphique illisible.



Nouvelle fonction de visualisation

J'ai alors développé un graphique à l'image d'un boxplot mais plus détaillé. C'est-à-dire un boxplot étendu avec différentes couches de quantiles qui se superposent pour mieux comprendre la dispersion de l'erreur en fonction du nombre d'individus de référence. Ainsi, nous pouvons obtenir un découpage avec la médiane mais aussi avec d'autres quantiles pour obtenir finalement un intervalle de confiance. Pour ce faire, nous utiliserons le package *'ggplot2'* et sa fonction *'geom_area()'* pour obtenir un graphique plus représentatif et élaboré.

```
196 funPlotTrait2 <- function(dfInput){
197
198   #Chargement des packages.
199   library(ggplot2)
200   library(dplyr)
201
202   #Initialisation des variables.
203   valQ <- 1 ; lineQ <- 0 ; col <- "grey" ; cVal <- 100 ; p <- ggplot()
204
205   #Boucle pour changer la valeur du quantile et la teinte du gris.
206   for (i in 1:20){
207     valQ <- valQ - 0.05
208     if(valQ > 0.45){
209       cVal <- cVal - 6
210     }else{
211       cVal <- cVal + 6
212     }
213
214     #Boucle pour calculer le quantile de chaque colonne.
215     for (i in 1:ncol(dfInput)){
216       lineQ[i] <- as.numeric(quantile(dfInput[,i], valQ, na.rm = TRUE))
217     }
```

Première partie de la fonction pour initialiser les valeurs.

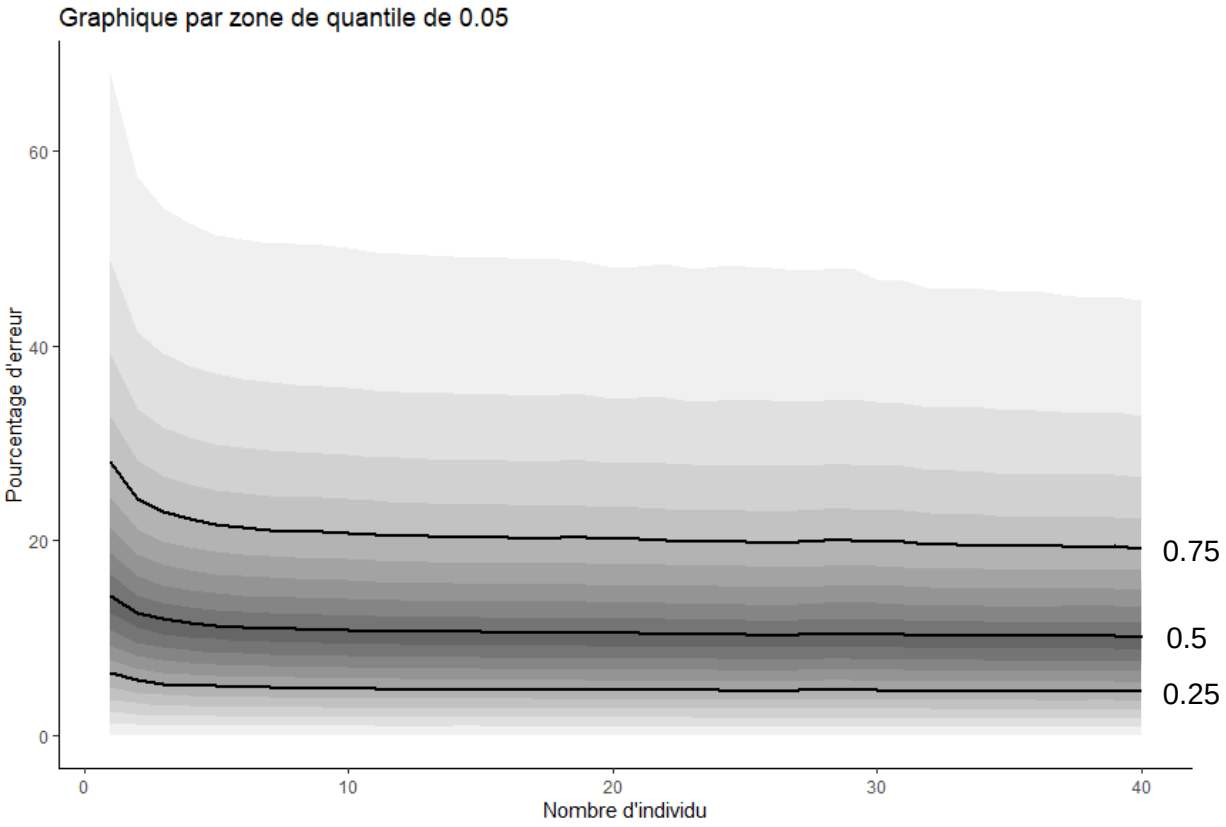
L'idée principale de cette fonction et de cette première partie est de tracer une couche tous les 0.05 quantiles d'une couleur différente en l'occurrence un niveau de gris. Pour réaliser cela, il a fallu créer une boucle qui à chaque tour modifie la valeur du quantile ainsi que la teinte de gris. A l'origine, la valeur du quantile vaut 1, puis la boucle effectue 20 itérations en soustrayant 0.05 à chaque fois pour finalement arriver à 0. Même chose pour la teinte de gris, la boucle modifie son intensité, en devenant de plus en plus foncé jusqu'à la médiane et s'éclaircit une fois celle-ci atteinte.

```
219     colF <- paste0(col,cVal,sep = "") #Couleur finale du gris, nom couleur + valeur couleur.
220     dfResQ <- data.frame(lineQ) #Création du df avec toutes les valeurs de quantile.
221
222     #Pour les quantiles à 25/50/75% on ajoute une courbe foncée.
223     if(between(valQ,0.49, 0.51) | between(valQ,0.74, 0.76) | between(valQ,0.24, 0.26)){
224       p <- p + geom_area(data = dfResQ, aes(x=1:nrow(dfResQ), y=lineQ), fill=colF, colour="black", size=1)
225
226     #Sinon on trace geom_area classique.
227     }else{
228       p <- p + geom_area(data = dfResQ, aes(x=1:nrow(dfResQ), y=lineQ), fill=colF)
229     }
230   }
231   #Détails du graphique.
232   p <- p + theme_classic() + labs(y= "", x = "")
233   return(p)
234 }
235 funPlotTrait2(dfInput)
```

Seconde partie pour tracer le graphique.

Nous pouvons désormais calculer pour chaque colonne de notre jeu de données la valeur des quantiles associés puis la fonction *'geom_area()'* trace une zone pleine en dessous de la courbe de ceux-ci. En commençant par le plus grand quantile (0.95) jusqu'au dernier. Ainsi les zones se superposent pour finalement obtenir différentes couches pour chacun des quantiles.

De plus, pour mettre en évidence la médiane ainsi que le premier et troisième quartile, nous traçons des courbes plus épaisses afin que celles-ci ressortent.



Interprétations

Avec ce nouveau graphique, nous pouvons formuler de nouvelles conclusions. En effet, nous observons que la médiane est plus basse que la moyenne, cela est logique puisque nous remarquons que la borne supérieure tend vers le haut. Cela signifie qu'il existe des valeurs ayant un grand pourcentage d'erreurs et que celles-ci augmentent la moyenne, ce sont les valeurs atypiques qui en sont à l'origine, celles-ci sont, cependant, difficiles à faire apparaitre sur le graphique puisqu'elles sont très grandes.

Fonction complémentaire supplémentaire

Tout comme pour notre fonction principale, j'ai implémenté une nouvelle fonction pour automatiser le processus de création de ces graphiques pour les 6 traits afin que celle-ci produise les deux graphiques principaux et les enregistres dans un dossier au nom de chaque trait.

```r
generateFiles <- function(dfInput){

  #Boucle pour chaque variable de notre liste d'entrée.
  for (varToPlot in dfInput){

    dfName <- paste0(varToPlot,"_2.RDS") #Nom du fichier à importer.
    dfPlot <- readRDS(dfName) #Lecture du fichier.
    dir.create(varToPlot) #Création du dossier de sortie.

    dfColMeans <- colMeans(dfPlot, na.rm = TRUE) #Calcul de la moyenne des colonnes.

    #Enregistrement du premier graphique.
    png(file.path(varToPlot,"meanPlot.png"))
    plot(dfColMeans, xlab = "", ylab = "")
    dev.off()

    p <- funPlotTrait2(dfPlot) #Execution de notre fonction.

    #Enregistrement du second graphique.
    png(file.path(varToPlot,"QPlot.png"))
    plot(p)
    dev.off()
  }
}
generateFiles(dfInput = c("H", "LS", "LL", "SLA", "LDMC", "LA"))
```

Fonction de génération des deux graphiques principaux.

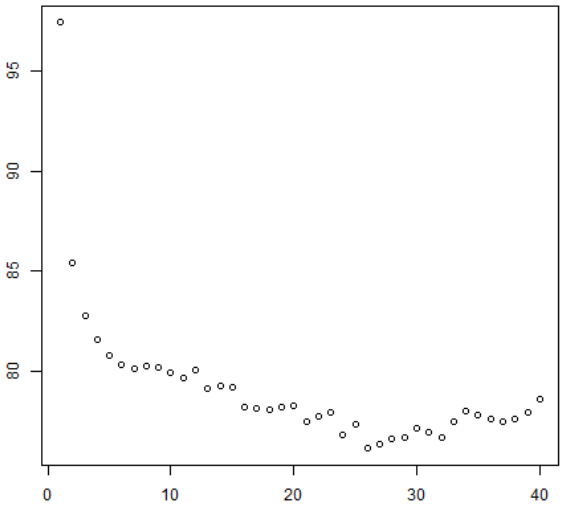
Le fonctionnement de celle-ci est assez simple. Il suffit de charger les jeux de données de nos traits (en les spécifiant en paramètre) puis de tracer nos graphiques dans la fonction en indiquant que nous souhaitons les enregistrer dans le dossier créé plus tôt.

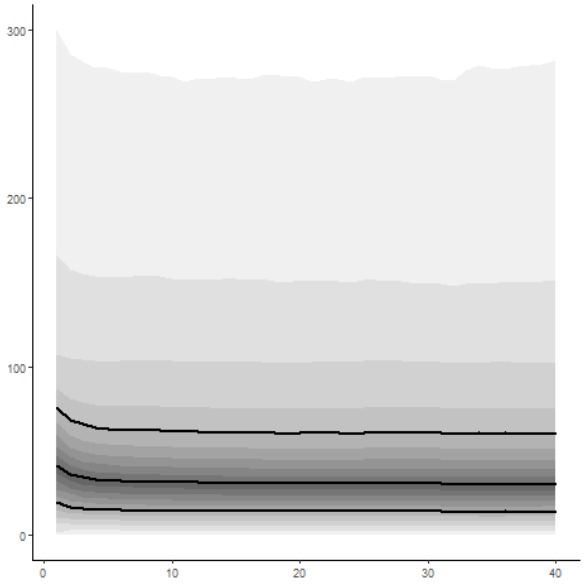
Dossiers de sorties avec les graphiques.

Les résultats graphiques pour chacun des traits

Avec, à chaque fois, le pourcentage d'erreurs en ordonnée et le nombre d'individus référents en abscisse pour les deux graphiques.

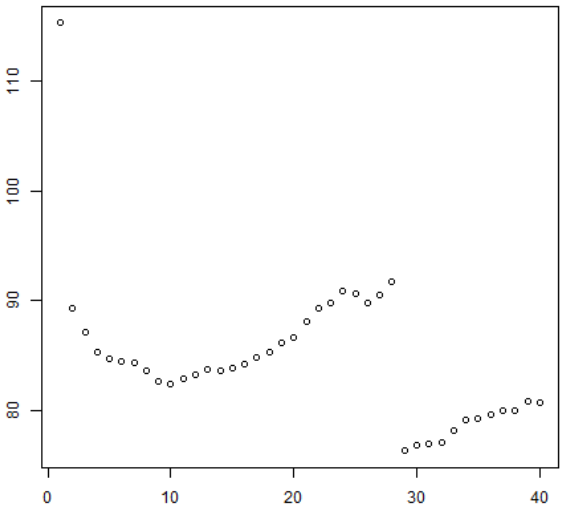
Trait H :

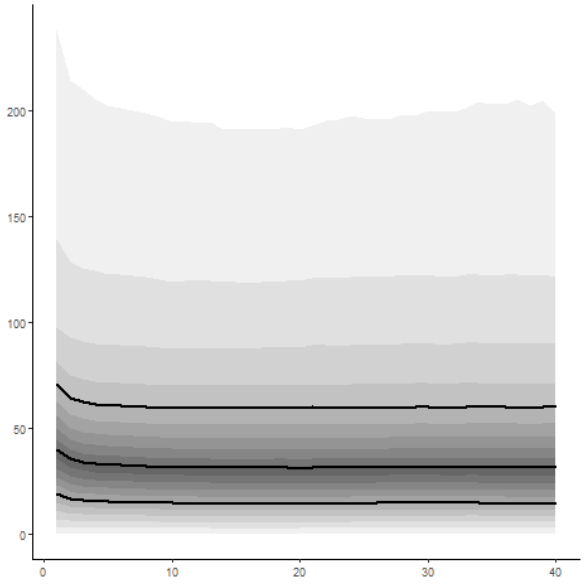




Nous observons que le pourcentage d'erreurs de la moyenne des colonnes débute à 100% puis atteint les 80% pour cinq individus. Nous remarquons que la zone du quantile à 95% tend vers les 300%. Cela signifie qu'il y'a de très grands pourcentages d'erreurs pour ce jeu de données qui font augmenter la moyenne.

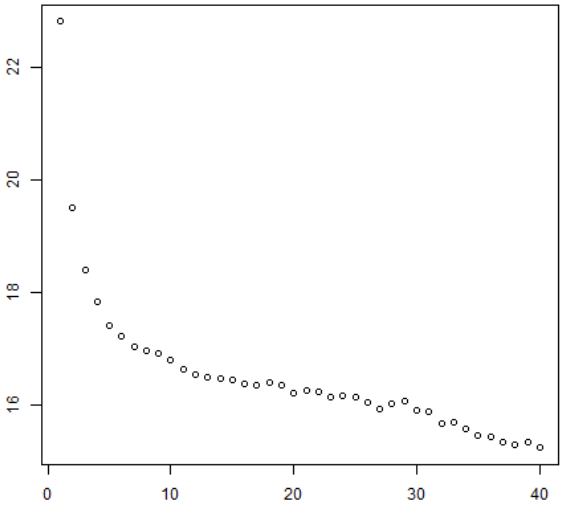
Trait LA :

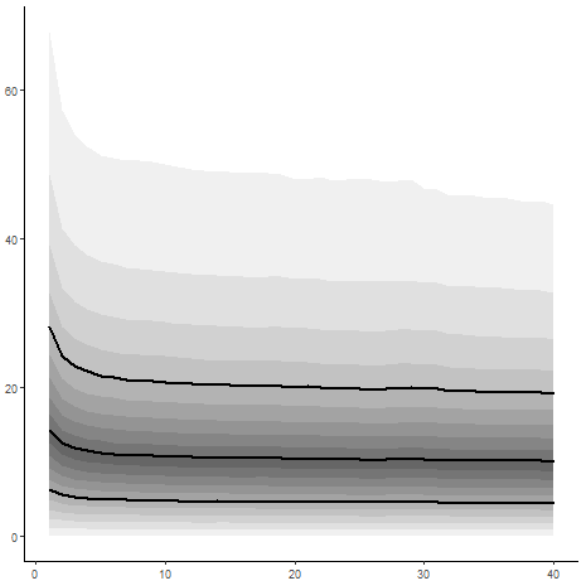




Les graphiques sont relativement similaires à ceux de H. Cependant le pourcentage d'erreur de la moyenne des colonnes est plus élevé alors que borne supérieure ne tend pas vers le haut. Cela signifie que ce jeu de données est plus homogène que le précédent.

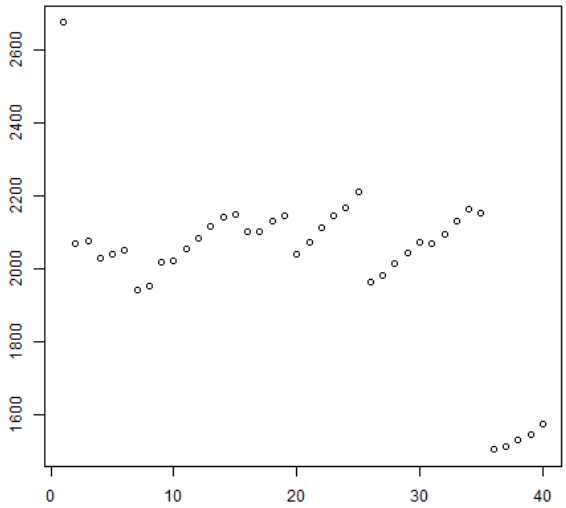
Trait LDMC :

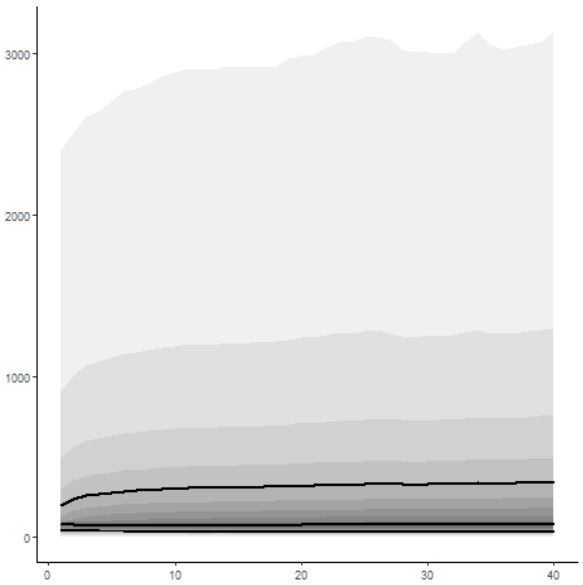




Les pourcentages d'erreurs de la LDMC sont bien moins élevés que les cas de figures précédents. Nous observons qu'à partir de cinq individus référent l'erreur est de 17%. De plus nous avons un intervalle de confiance allant de 0 à 60%. Ce sont des résultats satisfaisants.

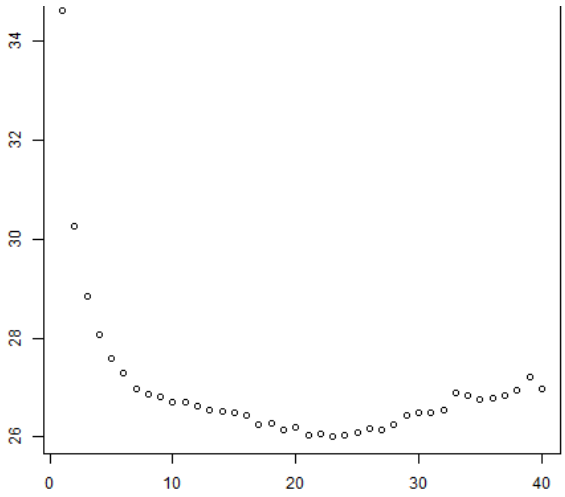
Trait LS :

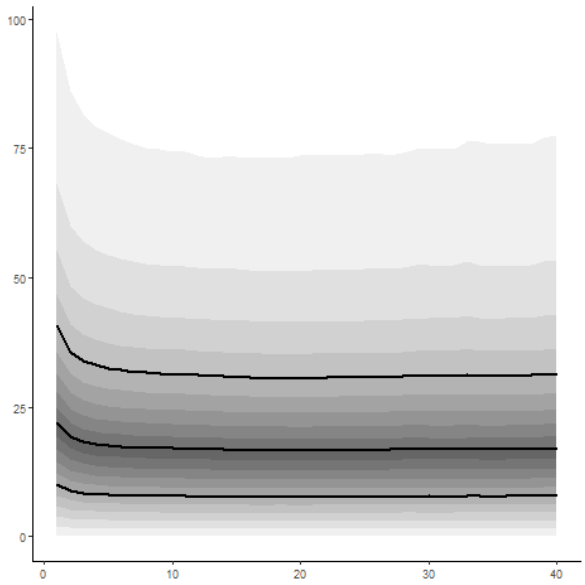




Pour ce trait, nous obtenons un très grand pourcentage d'erreurs, celui-ci est instable et se situe aux alentours des 2 000%. La médiane, elle, est bien plus faible, ce sont les très grandes valeurs qui sont responsables d'une moyenne aussi élevée.

Trait SLA :





Ce trait est très similaire à la LDMC, nous observons cependant un pourcentage d'erreurs plus élevé et un intervalle de confiance plus large (allant jusqu'à 100%). Ces résultats restent corrects.

En ce qui concerne le Trait LL, celui-ci possède des valeurs infinies dont je n'ai pas trouvé l'origines, celles-ci posent problèmes pour tracer les graphiques même en les ayant retirées…

Conclusion méthode moyenne

Ce premier test avec cette méthode nous permet déjà de conclure sur certains points. Nous avons estimé l'efficacité de la méthode de gapfilling par la moyenne à l'échelle individuelle. Celle-ci n'est très pas satisfaisante… En effet, nous avons observé des pourcentages d'erreur souvent élevés avec des intervalles de confiance larges. La méthode n'est pas rigoureuse pour gapfiller ces traits hormis pour la LDMC si nous estimons qu'un pourcentage d'erreur de 20% est acceptable.

Ce que nous aurions voulu déterminer, c'est quelle est la meilleure méthode de gapfilling à employer, en prenant en compte des facteurs comme l'indice d'aridité ou l'intensité de pâturage pour préciser et réduire le pourcentage d'erreur. L'objectif final étant de pouvoir gapfiller de la façon la plus efficace et la plus rigoureuse possible en appliquant la méthode nécessaire à la situation.

Les résultats pourraient être utilisés plus tard dans des documents scientifiques dans le cadre de Biodesert mais également en prenant en compte le futur impact des drylands.

Conclusion

J'ai éprouvé beaucoup de plaisir et de fierté à travailler au sein de l'Institut National de Recherche pour l'Agriculture, l'Alimentation et l'Environnement qui est le leader de la recherche dans le domaine agronomique au niveau européen. Cela m'a permis de découvrir le monde de la recherche au sein d'un institut public.

J'ai pu, ainsi, appliquer mes connaissances théoriques acquises lors de mes deux années de formation à l'IUT sur l'analyse et la visualisation de données ; notamment sur les problématiques liées aux bases de données.

Un aspect central de ce stage est la compréhension des données, des besoins et donc du domaine de l'environnement. C'est un monde que j'ai eu beaucoup de plaisir à découvrir. J'ai également beaucoup appris au niveau du langage de programmation utilisé : R. Les connaissances basiques du langage R que j'avais ne suffisaient pas et j'ai donc dû les renforcer en toute autonomie (notamment à l'aide de recherche sur internet) afin de répondre aux besoins. J'ai ainsi pu diversifier ma palette de langages de programmation.

Ces 10 semaines de stage à l'UREP m'ont permis de mieux découvrir le domaine de la statistique qui est un des deux piliers de ma formation. Cela conforte ma volonté de continuer dans ce domaine. J'ai développé une certaine réflexion et autonomie nécessaires à l'avancée de ce stage et acquis des compétences en programmation et visualisation de données. Ces compétences me seront très utiles dans le futur.

J'ai particulièrement apprécié l'environnement professionnel car j'ai trouvé qu'il y'avait un véritable climat de bienveillance et de coalition entre les corps de métiers. J'ai aussi découvert plusieurs branches de l'agronomie ainsi que les projets sur lesquels les personnes de l'UREP travaillent. Les échanges informels font partie de la vie de l'unité et permettent de contribuer, de manière conviviale, à la symbiose des projets.

Cette expérience de stage m'a profondément nourri aussi bien sur le plan professionnel que personnel et je le dois essentiellement à l'équipe bienveillante et profondément humaine qui m'a accompagné.