Les 12 travaux d'Asterix

Corentin Bretonniere

Le dossier suivant consiste à une évaluation de 12 travaux de recherche sur R et en maths effectués par la promo 2020/2022 MSc Data Management. Le système de notation est le suivant :

- La mise en forme | 3 |
- La fluidité et l'intelligibilité du contenu | 3 |
- La pertinence du travail | 5 |
- La mise en perspective (exemples) | 4 |
- La compréhension du sujet | 5 |

L'attribution de notes et le jugement du travail de mes camarades me met dans un situation inconfortable car je ne suis pas légitime en tant qu'étudiant et camarade à les évaluer. Merci de ne prendre que peu en compte mes remarques pour chaque travail dans votre notation finale.

```
library("knitr")
```

1/ Régression Logistique et application avec R" - MATHS

Ce travail à été effectué par Gaspard Palay, il est disponible sur son Github : "GaspardPalay"

https://github.com/GaspardPalay/PSBX/blob/main/Regression%20Logistique/Regression_Logistique%20_Maths.pdf

Synthèse du travail :

Dans ce travail Gaspard explique le principe de l'algorithme de regression logistique, cet algorithme ne fais pas de classification, uniquement de la régression.

Après une présentation mathématique succinte il fais une application sur R avec un jeu de donnée du Titanic. Par la suite il développe un modèle de machine learning après un nettoyage des données afin de comparer l'apprentissage suite à la régression logistique avec la réalité. On remarque que le modèle est fiable à 82% avec une erreur concernant 12 passagers sur leur survie et de 27 passagers sur leurs morts.

Synthèse du développement mathématique :

Pour ce qui est des concepts mathématiques, voici ceux présentés par Gaspard :

La regression logistique est un prolongement de la regression linéaire.
Avec un modèle de regression linéaire classique, on aura le modèle mathématique suivant :
$= \alpha X + \beta$

L'espérance sera donc calculée avec la foction suivante :
$(Y) = \alpha X + \beta$

La fonction Y, dans un modèle de regression logistique étant distribuée de manière binaire, on considère une fonction généralisée de **lien** :
$(E(Y)) = \alpha X + \beta$

La fonction de lien, pour une regression logistique est exprimée comme telle :
$(p) = log(\frac{p}{1-p})$

Théorème de Bayes
Probabilités conditionnelles – On se place dans le cadre binaire $Y \in \{+, -\}$

Estimer la probabilité conditionnelle P(Y/X) $\begin{cases} P(Y = y_k) = \frac{P(Y=y_k) \times P(X/Y=y_k)}{P(X)} \\ \qquad\qquad = \frac{P(Y=y_k) \times P(X/Y=y_k)}{\sum_{l=1}^{k} P(Y=y_l \times P(X/Y=y_l)} \end{cases}$

Hypothèse fondamentale de la régression logistique

$$ln\left[\frac{P(X/Y=+)}{P(X/Y=-)}\right] = b_0 + b_1 + ... + b_j X_j$$

Le théroème de Bayes est un théorème mathématique de probalibilité reposant sur la connaissance d'un événement pour prédire la probabilité de l'événement suivant. Ce théorème couvre les distribution suivantes :

- Loi Gamma, Beta, Poisson
- Loi Exponentielle
- lLoi Normale
- Loi Discrète
- Mélange de variable binaire

La fonction de repartition logistique est bornée entre 0 et 1, tend vers - infini en 0 et + infini en 1 elle est égale à 0 en 0,5 et a une forme de "S".

Evaluation du travail :

J'ai apprécié le travail de Gaspard, la mise en forme est très correcte, son contenu est fluide et intélligible bien que la partie mathématique soi assez peu développée. Le travail est très pertinent et la mise en perspective avec l'exemple sur R du jeu de donnée du Titanic est excellente.

D'après mon système de notation Gaspard aurait eu 17/20.

2/ Arbre de Décision - MATHS

Ce travail à été effectué par Rindra Lutz et Nicolas Allix, il est disponible sur le Github : "rindra-lutz"

https://github.com/rindra-lutz/psb1/blob/Travaux-Math%C3%A9matiques/Arbres-de-D%C3%A9cision.pdf

Synthèse du travail :

Ce travail developpe le principe général des arbres de régressions, il y a une différentiation à faire entre les arbres de regression et les arbres de classification.

Les arbres de regressions permettent de prédire une réponse quantitative (le prix d'une maison par exemple) tandis que les arbres de classifications permettent de prédire une réponse qualitative (la maladie ou le virus dont quelqu'un est atteint en fonction de ses symptomes).

Les arbres de décisions ont pour très gros avantage d'être simple à representer, à chaque noeud un partionnement est fait en fonction d'une règle (aussi appelé "seuil de coupure").

Synthèse du développement mathématique :

Mathématique, les arbres de décisions reposent sur des concepts abordables, la pureté et le coût des différents noeuds :'

Pureté

On considère un noead pur si tous les individus associés à une des valeurs appartiennent effectivement à cette classe.
La pureté d'un noeud se mesure avec l'indice de Gini, plus la valeur de l'indice est proche de 0, plus le noeud est pur.

$$G_i = 1 - \sum_{k=1}^{n} P_i, k^2$$

Coût du noeud

Le coût du noeud va permettre de mesurer la pertinence du choix de la variable de décision.
Ce coût est calculé via la formule suivante :

$$J(k) = (\frac{m_{gauche}}{m}) G_{gauche+} (\frac{m_{droite}}{m}) G_{droite}$$

L'indice de Gini sert à mesurer la pureté d'un noeud, un noeud est pur si tous les individus associés à une valeur appartiennent effectivement à cette classe. Par exemple avec la base de donnée Iris, si le seuil de coupure est : Petal lengh > 5, il risque d'y avoir quelques virginica qui iront dans la classe des versicolor, et inversement.

C'est pour ce genre de soucis que le cout du noeud est utile. En effet le calcul à effectuer pour chaque noeud va mesurer la pertinence du seuil de coupure, plus le resultat est proche de 0, plus le noeud est pur.

Evaluation du travail :

La mise en forme du travail de Rindra est correcte, le contenu est fluide et intelligible, le travail est assez pertinent bien que je déplore la non présence d'un exemple sur R ou d'une mise en perspective. Le sujet est compréhsensible même s'il ne développe pas les interets du calcul de la pureté et du coût du noeud.

Avec mon système de notation Rindra et Nicolas auraientt eu 11/20.

3/ Validation croisée - MATHS

Ce travail à été effectué par Nicolas Allix et Rindra Lutz, il est disponible sur le Github "Nicolas-all" :

https://github.com/Nicolas-all/PSB1/blob/main/Validation-Crois%C3%A9e.pdf

Synthèse du travail :

La validation croisé a pour principe de valider l'apprentissage d'un echantillon pour mesurer la fiabilité d'un échantillon. Il est donc nécessaire de définir une population d'apprentissage, une population de test et une population de validation.

On essaye plusieurs modèles sur la population d'apprentissage, on identifie ensuite le modèle le plus robuste sur la population de validation et pour finir on test sur les données de l'échantillon test.

Il existe 3 principes de méthodes de validation croisée : - LOOCV - LKOCV - k-fold

Cependant ils ne sont pas développés dans ce travail.

Synthèse du développement mathématique :

Il n'y a pas d'équations mathématique, ni exemple, ni de mise en perceptive, ce travail est un amat de paragraphes "définition" sans contexte ni développement.

Evaluation du travail :

D'après mon système de notation Rindra et Nicolas auraient eu 6/20 sur ce travail. Cette note est assez sévère, mais la présence d'un développement des 3 méthodes de validation croisée cité manque vraiment à ce travail. Même si ces concepts sont surement compliqué il n'était pas demandé de les maitriser mais au moins de les aborder et donner les outils necessaire à leurs compréhension. Le fait qu'il n'y ai pas d'exemple non plus rend la compréhension des validations croisée beaucoup plus difficile. L'absence pure et simple de concepts mathématiques, d'équations et d'exemples est la raison de cette faible note.

4/ Regression linéaire simple et multiple - MATHS

Ce travail à été effectué par Nina Zoumanigui, il est disponible sur son Github "Nina809" :

https://github.com/Nina809/PSBX/blob/main/Regression.Rmd

Synthèse du travail :

Une régression linéaire multiple à exactement le même principe qu'une régression linéaire simple, la seule différence est qu'une regression linéaire multiple a plusieurs variables explicatives indépendantes. Il est important de déterminer une hypothèse nulle : l'hypothèse nulle est qu'il n'y à pas de corrélation entre la ou les variables explicatives et la variable expliquée. L'hypothèse alternative par conséquent est l'existence d'une corrélation, la variable expliquée est en effet influencée par la ou les variables explicatives indépendantes.

Synthèse du développement mathématique :

Le document est publié seulement en rmd et Nina ne donne pas la version PDF. Cependant il ne fournit pas le jeu de donnée.. il est donc impossible pour le lecteur de visualiser les resultats, on doit se contenter des lignes de codes. Cependant le fichier est bien ordonné.

Nina défini d'abord les variables puis assigne à "data" sont jeu de donnée et plot le resultat.

```
data=read.csv("C:\\Users\\ninaz\\OneDrive\\Bureau\\R\\mtcars.csv",sep = ";", header= T)
attach(data)
head(data)
```

```
plot(mpg~wt,pch=20)
fit= lm(mpg~wt,data=data)
fit
abline(fit,col="red",lwd=2)
```

Evaluation du travail :

La lecture est facile bien que le fichier soi en rmd. Evoquer la regression logistique aurai pu être interressante car c'est la suite "logique". Les points abordés sont bien expliqués.

D'après mon système de notation, Nina aurait eu 14/20.

5/ Algèbre Tropical - MATHS

Ce travail à été effectué par Marion Danyach, il est disponible sur son Github "MarionD436" :

https://github.com/MarionD436/MATHS

Synthèse du travail :

L'algèbre tropical à pour but de redéfinir l'addition, la multiplication et toutes les opérations associées. Avec quelques définitions, l'autrice développe certains concepts assez compliqués qui pour être honnête me dépassent.

Concrétement l'Algèbre tropical est utile pour la modélisation, l'analyse, l'évaluation de performance pour des classes bien reprtoriées de système à évènements discrets deterministes ou stochastique. La notion d'idempotence est très importante en Algèbre topical, l'idempotence en mathématique et en informatique signifie qu'une opération a le même effet qu'on l'applique une ou plusieurs fois.

Par exemple la valeur absolue est idempotence : abs(abs(abs(x))) = abs(x)

Synthèse du développement mathématique :

Si e et f sont idempotent, alors e*f = e + f, intuitivement on comprend que si 2 éléments sont alors ils sont idempotent entre eux.

Les concepts lié à l'Algebre tropical sont compliqué et un niveau élevé en maths est nécessaire pour être en capacité de les comprendres. Cependant je perçois l'intuition derrière les formules.

Les idempotents peuvent eux même être décomposés de façon unique en somme d'idempotents minimaux.

Evaluation du travail :

C'était un sujet très compliqué que Marion à su vulgariser. Même si je n'en ai saisi seulement les principes vaguement c'est un travail interressant, pour bien le comprendre ça necessite des connaissances poussées en mathématique et en Algèbre.

D'après mon système de notation, Marion aurait eu 12/20.

6/ Prédiction avec Random Forest - R

Ce travail à été effectué par Thomas Masse, il est disponible sur son Github "Thomas-MAS"

https://raw.githubusercontent.com/Thomas-MAS/PSB1/main/randomForest/predictionPUBG.pdf

Synthèse du travail :

Pour appliquer Random Forest sur R, l'auteur à choisi d'utiliser un jeu de donnée disponible sur kaggle à l'adresse : https://www.kaggle.com/c/pubg-finish-placement-prediction , c'est un jeu de donnée regroupant plus de 65 000 parties séparées en jeu d'entrainement et en jeu de test, l'objectif est simple : prédire les chances de victoire de chaque joueurs.

Il y a 2 set de data, un set d'entrainements avec 29 colonnes telles que la distance marchée, les armes récupérées, le nombre d'enemis tués etc.. et un data set de test avec les mêmes colonnes sauf 1 qu'il faudra donc prédire : "winPlacePerc".

Dans un premier temps l'auteur effectue un travail de fourmis en analysant chacune des colonnes et nettoie les data d'éventuelles valeurs absurdes. Il à réussi à repérer 2 possibles tricheurs avec un nombre de kill bien trop élevé en connaissance des autres facteurs.

L'auteur retravaille ensuite ses colonnes en supprimant celles ne servant pas aux joueurs solo, en regroupant celles des objets consommables et en rajoutant une colonne ratio des kill par head shot comparé aux kill totaux.

L'étape suivante consiste à entrainer le modèle avec les 10 000 première lignes pour avoir un temps de calcul raisonnable. Il remarque quelles sont les variables les plus impactante sur la chance de gagner une partie, il se trouve que c'est la distance parcouru (ce qui n'est pas très étonnant car un joueur qui arrive à beaucoup se déplacer est forcément très expériementé).

L'algorithme peut etre optimisé en jouant avec "mtry" et "ntree" qui sont respectivement le nombre de variable utilisé à chaque séparation et le nombre d'arbre que l'algorithme construit. Après avoir tracé le MSE (mean squared tree), le nombre d'arbre optimal sembre être 200, le mtry optimal est au alentour de 7.

Pour ensuite passer à la prédiction, il faut effectuer les mêmes retouches de colonnes sur le data set test que sur le data set d'entrainement.

Une fois cela effectué, il suffit de lancer la prédiction en affichant les 10 joueurs avec un winPlacePrec les plus élevé. C'est le joueurs 2f70df5da78353 qui à le plus de chance de gagner avec un winPlacePerc de 0,88, c'est à dire que sur 100 parties avec les 10 000 joueurs utilisés, il en gagnera 0,88 ce qui est plutôt pas mal ! Ce joueur à de bonnes statistiques avec distance parcouru de 4810, 4 armes ramassées, et une kill place de 20.

Synthèse du code R :

Le travail de Thomas faisant au total 47 pages, il y a beaucoup de lignes de codes, toutes très bien utilisées et pertinente, je vous conseille sincèrement d'aller voir son travail, cependant je vais quand même en expliquer une partie.

```
trainsetPUBG$winPlacePerc <- as.numeric(trainsetPUBG$winPlacePerc)
trainsetPUBG$damageDealt <- as.numeric(trainsetPUBG$damageDealt)
trainsetPUBG$longestKill <- as.numeric(trainsetPUBG$longestKill)
trainsetPUBG$rideDistance <- as.numeric(trainsetPUBG$rideDistance)
trainsetPUBG$swimDistance <- as.numeric(trainsetPUBG$swimDistance)
trainsetPUBG$walkDistance <- as.numeric(trainsetPUBG$walkDistance)
```

Après avoir importé ces jeux de data, l'auteur s'est rendu compte que certaines colonnes comportant des chiffres étaient en format "character", il les convertis donc avec la fonction "as.numeric".

Avec le code suivant, pour chaque colonne il observe la distribution des valeurs, c'est grace à ces repartitions que l'on peu remarquer de possibles valeurs étranges, et c'est le cas pour le joueur f83f0bfaafb7d8 avec notamment une valeur DBNOS de 53 qui parait trop élevé.

```
distributionDBNOs <- count(trainsetPUBG, vars=DBNOs)
plot(distributionDBNOs, type="b", col="red", main="Distribution DBNOs", xlab="DBNOs", ylab=
"Nombre")
```

Pour vérifier le comportement de ce joueur et potentiellement l'exclure de l'étude, il étudie plus en profondeur les valeurs extrèmes d'autres colonnes avec le code suivant, pour voir si il retrouve ce joueur.

```
wA = trainsetPUBG[trainsetPUBG$weaponsAcquired > 150, ]
print(wA)
```

Voici présenté succintement en language R quelques unes des premières étapes du travail de Thomas.

Evaluation du travail :

C'est un travail très complet, très bien expliqué avec des commentaires étapes par étapes. L'algorithme random forest est un sujet intérressant et la mise en perspective avec les données d'un jeu vidéo pour prédire des chances de victoire est très pertinent,ça permet une parfaite compréhension.

C'est un excellent travail, avec mon système de notation Thomas aurait eu 20/20.

7/ DPLYR - R

Ce travail à été effectué par Grégoire Fontaine, il est disponible sur son Github "gfontainepsb" :

https://github.com/gfontainepsb/Cours-R/blob/main/dplyr.pdf

Synthèse du travail :

L'auteur de ce travail décrit l'utilité du package dplyr dans une rapide introduction, dplyr est un package comprenant des fontion aidant à la manipulation courante de données.

Il comprend 5 fonctions qui sont : - mutate() pour ajouter des nouvelles variables - select() pour selectionner des variables en fonction de leurs noms - filter() séléctionner des éléments en fonction de leurs valeurs - summarise() pour réduire plusieurs valeurs à un seul résumé - arrange() qui change l'ordre des lignes

Synthèse du code R :

L'auteur donne un exemple pour chacune des fonction présentées si dessous à l'aide une base de donnée de star wars.

Par exemple, dans l'exemple ci dessous il utilise la commande filter() pour selectionner tous les éléments avant pour valeurs "Droid" dans la colonne "species" :

```
starwars %>%
  filter(species == "Droid")
```

```
## # A tibble: 6 x 14
##   name  height  mass hair_color skin_color eye_color birth_year sex   gender
##   <chr>  <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
## 1 C-3PO    167    75 <NA>       gold       yellow           112 none  mascu~
## 2 R2-D2     96    32 <NA>       white, bl~ red               33 none  mascu~
## 3 R5-D4     97    32 <NA>       white, red red               NA none  mascu~
## 4 IG-88    200   140 none       metal      red               15 none  mascu~
## 5 R4-P~     96    NA none       silver, r~ red, blue         NA none  femin~
## 6 BB8       NA    NA none       none       black             NA none  mascu~
## # ... with 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

```
#> # A tibble: 6 x 14
#>   name  height  mass hair_color skin_color eye_color birth_year sex   gender
#>   <chr>  <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
#> 1 C-3PO    167    75 <NA>       gold       yellow           112 none  mascu...
#> 2 R2-D2     96    32 <NA>       white, bl... red               33 none  mascu...
#> 3 R5-D4     97    32 <NA>       white, red red               NA none  mascu...
#> 4 IG-88    200   140 none       metal      red               15 none  mascu...
#> 5 R4-P...     96    NA none       silver, r... red, blue         NA none  femin...
#> # ... with 1 more row, and 5 more variables: homeworld <chr>, species <chr>,
#> #   films <list>, vehicles <list>, starships <list>
```

Ci dessous, voici comment utiliser la fonction "select", l'auteur décide de sélectionner toutes les colonnes donnant une valeur correspondant à une couleur (donc les cheveux, les yeux et la couleur de peau) :

```
starwars %>%
  select(name, ends_with("color"))
```

```
## # A tibble: 87 x 4
##    name               hair_color    skin_color  eye_color
##    <chr>              <chr>         <chr>       <chr>
##  1 Luke Skywalker     blond         fair        blue
##  2 C-3PO              <NA>          gold        yellow
##  3 R2-D2              <NA>          white, blue red
##  4 Darth Vader        none          white       yellow
##  5 Leia Organa        brown         light       brown
##  6 Owen Lars          brown, grey   light       blue
##  7 Beru Whitesun lars brown         light       blue
##  8 R5-D4              <NA>          white, red  red
##  9 Biggs Darklighter  black         light       brown
## 10 Obi-Wan Kenobi     auburn, white fair        blue-gray
## # ... with 77 more rows
```

Evaluation du travail :

C'est un travail, certes assez court mais efficace. L'auteur vas droit au but et ses exemples sont parlant et explicite. Des phrases manuscrites expliquant les resultats et quel resultat est recherché n'aurait pas été de trop. Cependant j'ai compris les principes et l'utilisation du package dplyr, n'est ce pas là n'essentiel ?

Selon mon système de notation, Grégoire aurait eu 14/20.

8 Forecast - R

Ce travail à été effectué par Arnaud Forasacco, il est disponible sur son Github "ArnaudFrsc" :

https://github.com/ArnaudFrsc/PSBX/blob/main/Forecast_Pack.pdf

Synthèse du travail :

Le package Forecast sur R est composé de méthodes et d'outils pour affiher et analyser des séries temporelles univariées via des modèles d'espaces et de la modélisation automatique ARIMA.

L'auteur utilise une base de donnée CSV du traffic aérien, à l'aide du package il en fais une série temporelle idéale pour l'analyse et une futur prédiction.

Sans beaucoup d'explications quant à la base de donnée utilisée et l'objectifs des lignes de codes, il est n'est pas facile de comprendre et d'interpreter les différents graphiques tracés.

L'auteur utilise 3 méthodes pour prédire le traffic aérien dans les futures années : - Naive - ETS (Exponential Smoothing Algorithm) - modèle ARIMA

Synthèse du code R :

Après avoir changé les types de données charactère en données numeriques pour afficher les graphiques proprement, l'auteur créer une série temporelle avec le code suivant :

```
Y <- ts(groupe_st[,4],start=c(2000,1), frequency=12)
```

Il représente ensuite graphique sa série temporelle avec le code suivant :

```
autoplot(Y) + ylab("Traffic")
```

Ensuite nous pouvons voir l'application des 3 méthodes de prédiction, en commençant par Naive. La méthode Naive utilise les chiffres réels de la dernière période comme prévision, il est généralement utilisé à postériori à des fins de comparaisons :

```
fit_n <- snaive(Y)
print(summary(fit_n))
```

Pour ce qui est de la méthode ETS, c'est une des plus utilisées en statistique pour prédire une valeur future en fonction des valeurs précédentes, voici le code pour l'utiliser :

```
fit_ets <- ets(Y)
print(summary(fit_ets))
```

Avec la représentation graphique nous pouvons voir le traffic aérien prédit pour la période à venir :

```
fcts_ets <- forecast(fit_ets, h=12)
autoplot(fcts_ets)
```

Avec la même syntaxe que pour les méthodes précédentes, l'auteur présente la fonction ARIMA.

Evaluation du travail :

J'ai bien aimé ce travail, étapes par étapes les codes utilisées sont compréhensible et l'utilité du package transparait correctement. Plus d'explications sur les données utilisées et d'interpretation des résultats n'aurait pas été de trop, cependant c'est un travail plutôt complet qui permet d'appréhender le package forecast et d'en comprendre le principe et l'utilisation.

Selon mon sytème de notation, Arnaud aurait eu 15/20

9/ Introduction à parralel - R

Ce travail à été effectué par Adrien Jupiter, il est disponible sur son Github "akjupiter" :

https://github.com/akjupiter/PSBX/blob/master/Rpackage/parallel.pdf

Synthèse du travail :

Ce package sert à effectuer des calculs en parralèles afin d'optimiser les temps de calculs des programmes R, parfois assez lent. Ce package est installé par defaut sur R. L'auteur précise qu'il est nécessaire d'avoir des notions en informatique théorique pour la compréhension globale de ce package. Compte tenu de ce paramètre, cette présentation ne couvrira pas l'ensemble des fonctionnalitées du package parralel.

L'utilisation de ce package recquière l'accès à plusieurs unitées de calcul, elles peuvent être localisées en CPU (Central Processing Unit) ou en GPU (Graphical Processing Unit). Le calcul parralèle en GPU (donc avec des cartes graphiques) à été démontré plus rapide, c'est nottament pour celà que les mineurs de crypto monnaie utilisent des champs de cartes graphiques.

L'auteur nous montre ensuite la différence des temps de calculs avec l'utilisation des calculs parralèles et sans.

Synthèse du code R :

L'auteur va calculer le temps mis pour calculer la moyenne d'un échantillon de taille r selon une distribution normale avec mean = 5 et sd = 10, 25 fois pour : - r = 10 - r = 1 000 - r = 100 000 - r = 10 000 000

```
system.time(
  resultats$res_non_par <- sapply(r_values, FUN = myfun,
                mean = 5, sd = 10) # options de la fonction myfun
)
```

```
##    user  system elapsed
##  18.276   1.072  19.350
```

En comparaison, voici les lignes de codes necessaire pour utiliser le calcul parralèle et les temps de calculs respectifs :

```
P <- 4  # définir le nombre de coeurs
cl <- makeCluster(P) # réserve 4 coeurs - début du calcul
system.time(
 res_par <- clusterApply(cl, r_values, fun = myfun, # évalue myfun sur r_values
                   mean = 5, sd = 10)  # options de myfun
)
```

```
##    user  system elapsed
##   0.028   0.004   7.838
```

```
stopCluster(cl) # libère 4 coeurs - fin du calcul
```

On remarque que les temps de calculs sont très inférieurs, le package parralel est très efficace !

L'ajout de commentaires à chaque ligne du code par l'auteur rend la compréhension très aisée et c'est une vrai valeur ajoutée, son travail est très didactique.

Evaluation du travail :

Comme dit précédemment, cette présentation par Adrien du package parralel est très didactique, bien expliquée étape par étape, l'illustration avec un exemple permet de bien se rendre compte de gain de temps associé à l'utilisation de parralel.

Selon mon système de notation, Adrien aurait eu 18/20

10/ Network 3D - R

Ce travail à été effectué par Claire Mazzucato, il est disponible sur ton Github "clairemazzucato" :

https://github.com/clairemazzucato/PSBX/blob/main/Packages/NetworkD3/NetworkD3.pdf

Synthèse du travail :

Le package Network3D permet des réseaux sur R. R étant un logiciel de programmation très complet il est pertinent d'y analyser des réseaux puis leurs représentations graphiques, des packages tels que igraph sur R le permettent.

Le package Network3D prend en charge plusieurs type de graphiques de réseau dont : - Force directed network avec simpleNetwork et forceNetwork - Les diagrammes Sankey avec sankeyNetwork Les réseaux Radial avec radialNetwork

Après une contextutalisation pertinente et une explication de l'utilité du package Network3D, l'autrice nosu explique comment installer le package et illustre son utilisation avec la création d'une base de donnée fictive. Les lignes de codes sont aggrémentées de commentaires ce qui rend la compréhension et la lecture fluide.

Synthèse du code R :

Voici le code permettant de créer rapidement une base de donnée fictive et de la plot ensuite :

```
# Chargement du package
library(networkD3)

# Création de données fictives
src <- c("Claire", "Claire", "Claire", "Adrien",
        "Adrien", "Adrien", "Claude", "Claude", "Claude","Siva", "Siva", "Siva","Thuy","Thu
target <- c("Adrien", "Claude", "Siva", "Arnaud",
            "Siva", "Claude", "Claire", "Arnaud", "Siva","Claude","Adrien","Claire","Claire
networkData <- data.frame(src, target)

# Plot
simpleNetwork(networkData)
```

Ensuite avec forceNetwork, une fonction du package Network3D, l'autrice nous présente le code permettant de tracer des réseaux plus compliqués :

```
# Chargement de données
data(MisLinks)
data(MisNodes)

# Plot
forceNetwork(Links = MisLinks, Nodes = MisNodes,
             Source = "source", Target = "target",
             Value = "value", NodeID = "name",
             Group = "group", opacity = 0.8)
```

L'autrice vas par la suite illustrer l'utilisation de sankeyNetwork avec un exemple de base de donnée sur les résultats de vote des 12 régions britanniques pour quitter, ou non, l'union européenne. Encore ici, étape par étapes, l'autrice explique avec des "points d'étapes" les actions nécessaires pour utiliser sankeyNetwork.

Après avoir lié le data set à R, il faut aggréger les données par régions :

```
# aggregate by region

results <- refresults %>%
  dplyr::group_by(Region) %>%
  dplyr::summarise(Remain = sum(Remain), Leave = sum(Leave))
```

Puis créer les noeuds et les liens :

```
# création des noeuds

regions <- unique(as.character(results$Region))
nodes <- data.frame(node = c(0:13),
                    name = c(regions, "Leave", "Remain"))

#création des liens

results <- merge(results, nodes, by.x = "Region", by.y = "name")
results <- merge(results, nodes, by.x = "result", by.y = "name")
links <- results[ , c("node.x", "node.y", "vote")]
colnames(links) <- c("source", "target", "value")
```

Et enfin utiliser la fonction sankeyNetwork pour dessiner le réseaux :

```
#draw sankey network

networkD3::sankeyNetwork(Links = links, Nodes = nodes, Source = 'source',
                         Target = 'target', Value = 'value', NodeID = 'name',
                         units = 'votes')
```

Pour ce qui est de la fonction radialNetwork, elle sert à modéliser un data set sous la forme d'un arbre Reingold-Tilford, l'autrice joint un exemple du code nécessaire :

```
Flare <- jsonlite::fromJSON(
  "https://gist.githubusercontent.com/mbostock/4063550/raw/a05a94858375bd0ae023f6950a2b13fac51
  simplifyDataFrame = FALSE
)

hc <- hclust(dist(USArrests), "ave")

radialNetwork(List = Flare, fontSize = 10, opacity = 0.9, margin=0)

radialNetwork(as.radialNetwork(hc))

# and with a different font
radialNetwork(List = Flare, fontSize = 10, opacity = 0.9, margin=0, fontFamily = "sans-serif")

diagonalNetwork(List = Flare, fontSize = 10, opacity = 0.9, margin=0)

diagonalNetwork(as.radialNetwork(hc), height = 700, margin = 50)
```

Evaluation du travail :

Cette présentation de Network3D est bien faite, les fonctions sont illustrées avec des exemples parlant et la construction du devoir est très didactique. Il manque juste la représentation graphique des fonctions et un commentaire d'intérprétation. Sinon c'est un excellent travail.

Avec mon système de notation, Claire aurait en 16/20

11/ Critique de mon travail - Arbres de décision - MATHS

Après avoir consulté et lu de nombreux travaux de recherche en mathématique effectués par mes camarades, j'ai un regard plus critique sur mon travail. Je peux notamment comparer mon travail avec celui de Rindra et Nicolas car nous avons pris le sujet sur les arbres de décisions.

Sans prétention je pense avoir mieux traité le sujet, la valeur ajouté est mon exemple d'application sur R de création d'arbre et l'ouverture sur le random forest que je n'ai malheuresement pas traité.

Ayant à l'époque conscience de l'excellent travail de Thomas sur le Random Forest j'ai trouvé bien plus pertinent que rediriger mon lecteur vers son travail en guise de conclusion/ouverture que de l'expliquer moi aussi, chose que j'aurais surement beaucoup moins bien fait.

Avec du recul, un regard critique, et mon système de notation je pense que mon travail mérite 15/20.

12 / Critique de mon travail - dabr - R

Après avoir consulté et lu de nombreux travaux de mes camarades, j'ai pu avoir un regard plus critique sur la présentation que j'avais faite sur le package dabr.

J'ai remarqué que j'appreciais beaucoup les commentaires et points d'étapes lorsque mes camarades en mettaient, cela rend la compréhension plus limpide et ça fait un travail beaucoup plus didactique.

Je n'ai vraiment pas assez développé, voir pas développé l'utilisation de certaines fonction du package dabr comme la fonction update ou la fonction quote par exemple.

Avec du recul, un retard critique, et mon système de notation je pense que mon travail mérite 13/40.