Partiel PSB : cours Programmation R avec Mr.LAUDE

Thuy AUFRERE

31/01/2021

1. Introduction

Les 5 travaux R sélectionnés sont :

- **ggplot2** de Claude REN
- **dplyr** de Soukaina EL GHALDY et Jiayue LIU
- **quandmod** d'Antoine SERREAU
- **lubridate** de Gaspard PALAY
- **mnist** de Kanlanfeyi KABIROU et Jordy HOUNSINOU

Pourquoi j'ai sélectionné ces 5 travaux :

J'ai sélectionné ces 5 travaux pour plusieurs raisons. La première concerne l'originalité de certains packages. La deuxième sur l'utilité de certains packages que je serais sûrement amenée à utiliser plus tard. Enfin la troisième raison concerne la qualité du R markdown permettant de reproduire facilement les exemples. Ne pouvant choisir que 5, il a fallu faire un choix mais je tiens à souligner qu''il y a aussi d'autres travaux de qualité.

Pour chaque travail, je commenterais selon le plan suivant :

- les informations sur le travail (nom du package, nom de l'auteur et le lien vers Github)
- une rapide synthèse reflétant ma compréhension du travail
- une explication de certains codes R : je prendrais au moins deux codes par travaux pour les expliquer
- une évaluation du travail selon mes critères qui sont : i) sur la forme (structure, clarté, fautes d'orthographe) et ii) sur le fond (explication, exemple, données). Je n'attribuerais pas de notes pour les travaux. En effet, étant débutante sur R, je ne me sens pas légitime d'attribuer une note. Cette évaluation sera suivie d'une petite conclusion concernant l'appréciation du travail.

1. Travail 1

Nom du package R : **ggplot2**

Auteur du travail : Claude REN

Lien du travail sur Github :
https://github.com/Cldren/REN_PSBx/tree/main/Packages_presentation/ggplot2

1.1. Synthèse du travail

Ce tutorat porte sur le package ggplot2 dans R. Il a pour but de montrer la démarche à suivre ainsi que les représentations que l'on peut réaliser avec ggplot2. Ce tutorat se base sur un jeu de données sur les voitures ayant comme information le poids, la vitesse etc des voitures.

1.2. Explication de certains codes

Premier code

```
# Initialization of the plot
g <- ggplot2(cars2, aes(x=Weight, y=Displacement))
# Adding points
g <- g + geom_point(col="blue", size=3)
# Changing x and y axis
g <- g + coord_cartesian(ylim=c(100, 500), xlim=c(2500, 5000))
# Adding linear model
g <- g + geom_smooth(method="lm")
# Adding title and legends
g <- g + labs(title="Displacement by weight", subtitle="From cars2 dataset",
              y="Displacement", x="Weight", caption="cars2")
# Customize axis
g <- g + scale_x_continuous(breaks=seq(2500, 5000, 250))
# Plotting
plot(g)
```

Ce code extrait du tutorat de ggplot 2 de Claude permet de faire un plot avec un strict minimum de paramètres. Tout d'abord on initialise le plot avec les données du jeu de donnés "cars2" et on définit un esthétisme avec *aes*. Ensuite,le code définit le type de points avec l'attribution d'une couleur et d'une taille. Puis le code modifie les paramètres de l'axe des abscisses et des coordonnées. Ensuite le code informe du modèle utilisé, ici c'est le modèle linéaire *lm*. Après le code permet d'ajouter un titre, une légende au plot avec une échelle aux axes.

==> Ce code est très intéressant pour un débutant qui ne connait pas R. Le code est simple mais contient beaucoup d'informations à apprendre.

Deuxième code

```
g <- ggplot(cars2, aes(x=Weight, y=Displacement))
g <- g + geom_point()
```

```
g <- g + geom_smooth(method="loess", se=F)
theme_set(theme_bw())
# Facet wrap with free scales
# manufacturer in rows and class in columns
g1 <- g + facet_grid( ~ Origin)
g2 <- g + facet_wrap( ~ Origin, scales = "free") +
  labs(title="Displacement by weight", caption = "Source: mpg",
       subtitle="Ggplot2 - Faceting - Multiple plots in one figure with free
scales")
ggarrange(g1, g2,
          labels = c("A", "B"),
          ncol = 1, nrow = 2)
```

Tout d'abord, on initialise le plot avec les données du jeu de donnés "cars2" et on définit un esthétisme avec *aes*. Puis il précise le type de points du plot avec la méthode utilisée, ici *loess* est le modèle non linéaire avec *se* l'intervalle de confiance. Ensuite, le code définit les données à mettre dans le plot (ici c'est les manufacteurs). Par la suite, le code définit une autre fonction avec les données, le titre, l'image, les sous-titres. Enfin le code finit avec le choix des 2 fonctions g1 et g2 intitulées respectivement dans le graphe A et B.

==> Ce code est aussi très intéressant pour apprendre à faire des comparaisons entre les paramètres. Cela permet de comprendre rapidement la tendance des données.

1.3. Evaluation du travail

Sur le fond: Dès le début de la lecture,le lecteur comprend rapidement ce qu'il va trouver dans le tutorat. Celui-ci est très compréhensible car accompagné d'un jeu de données à disposition du lecteur. Par ailleurs, chaque exemple et notion sont accompagnés d'un commentaire, indiquant la maitrise du sujet par l'auteur. Chaque code est accompagné d'une explication de la syntaxe et d'un visuel. Enfin, l'auteur donne son avis et donne des arguments pour et contre (exemple: page 5 et 6). Ce travail d'autocritique est très positif.

Sur la forme: Le tutorat est bien structuré, clair, aéré (séparation claire entre les paragraphes, les codes, les graphes). De plus, les codes sont faciles à comprendre car ils sont tous accompagnés par un # avec des commentaires. Cela permet, notamment aux lecteurs qui ne connaissent pas bien R (comme moi) de comprendre la signification du code. Enfin, l'auteur a montré une grande honnêté intellectuelle en citant toutes ces sources et en donnant tous les liens.

Ce tutorat a été très agréableà à lire car bien construit, expliqué et fluide. Je tiens à féliciter le fait qu'il a été fait en anglais, permettant à tout le monde de le lire. Si je devais faire une petite critique, je dirais qu'on reste un peu sur sa faim et qu'on aurait peut-être aimé une petite comparaison avec d'autres types de représentations graphiques pour aider les lecteurs à choisir quels packages utilisés lorsqu'on désire représenter des données visuellement.

2. Package 2

Nom du package R : **dplyr**

Auteur du travail : Soukaina EL GHALDY et Jiayue LIU

Lien du travail sur Github :

https://github.com/soukainaElGhaldy/PSB-X/tree/main/R_packages/dplyr_package

https://github.com/liu-jiayue/psbx

2.1. Synthèse du travail

Ce tutorat porte sur le package dplyr dans R. Il a pour but de montrer comment manipuler et explorer les données avec dplyr. il détaille avec précisions toutes les fonctions offcielles de dplyr.

2.2. Explication de certains codes

Premier code

```
filter(data, age >= 70 & age <= 80)

OU

filter(data, age >= 70, age <= 80)
```

Ce code montre 2 manières de faire un filtre sur les données.

==> Ce code est très intéressant pour les personnes qui font de l'analyse de données (comme moi en apprentissage), il est important de savoir nettoyer des données ou sortir facilement les données qui nous intéressent pour proposer des analyses pertinantes.

Deuxième code

```
data <- mutate(data,
                age_enf = esp/2,
                esp_enf = 72 - age_enf)
select(data, nom, age, esp, age_enf, esp_enf)
```

Ce code montre comment créer une nouvelle colonne avec les variables de l'âge et l'espérance de vie des enfants. Puis le code sélectionne quelques colonnes du dataframe.

==> Ce code est intéressant pour apprendre à créer des colonnes avec de nouvelles variables.

Troisième code

```
data1 %>%
  summarise(esp_moy = mean(esp, na.rm=TRUE),
            esp_enf_moy = mean(esp_enf, na.rm=TRUE))
```

Ce code montre comment on fait pour avoir les agrégats sur les données, ici c'est la moyenne de l'espérance de vie des personnes et des enfants. Il aurait été intéressant de savoir quels étaient les autres aggrégats que l'on peut résumer (ex: l'écart-type, la variance ? etc).

==> Ce code est intéressant pour avoir la vue rapidement sur les aggrégats.

2.3. Evaluation du travail

Sur le fond : Dès le début de la lecture,le lecteur comprend rapidement ce qu'il va trouver dans le tutorat. De plus, il est assez agréable d'avoir eu une explication sur le choix des auteurs de travailler sur ce package. Par ailleurs, plusieurs bonnes explications ont été faites tout au long du tutorat telles ques :

- l'explication très claire sur l'installation du package, facilement reproductible par tous
- l'explication de dplyr faisant parti de tidy

Il a été sympathique d'avoir eu une explication sur l'origine du package. Enfin, les auteurs ont défini chaque terme, permettant à tous les niveaux dont les débutants de bien comprendre chaque notion.

Sur la forme : Le tutorat est bien structuré, clair, aéré (séparation claire entre les paragraphes, les codes, les graphes). De plus, les codes sont faciles à comprendre. En effet, quand c'est nécessaire, les codes sont accompagnés par un # avec des commentaires pour une meilleure compréhension.

Enfin, les auteurs ont bien cité toutes les sources utilisées pour réaliser ce tutorat.

Ce tutorat a été bien agréable à lire. Si je devais faire une petite critique, il aurait été intéressant d'avoir une rapide description des autres packages regroupés dans *tidyverse* pour donner envie et inciter le lecteur à aller plus loin. Par ailleurs, le mélange du français et de l'anglais (exemple page 12) n'est pas à mon sens cohérent pour certains lecteurs, soit c'est tout en anglais soit tout en français.

3. Package 3

Nom du package R : **quantmod**

Auteur du travail : Antoine SERREAU

Lien du travail sur Github :
https://github.com/aserreau/PSB1/tree/main/Travaux%20Packages

3.1. Synthèse du travail

Ce tutorat porte sur le package quantmod dans R. Il a pour but de montrer la démarche à suivre pour faire de l'analyse financière des actions comme un vrai trader. Ce package permet de savoir comment analyser le cours des actions en bourse. Le tutorat se focalise sur les parties : i) charger les données ii) cartographier la donnée iii) revue des indicateurs clés sur le cours des actions.

3.2. Explication de certains codes

Premier code

```
chartSeries(AAPL,
            name = "Evolution du cours d'Apple",  #ajout d'un titre
            subset="2020-06::2020-11",  #choix du mois de mars à novembre 202
0
            type = "candlesticks",
            theme = "white",  #choix du theme en blanc
            TA = c(addRSI(), addBBands())) #Expl : ajout d'indicateurs
```

Ce code montre comment on crée un graphe montrant l'évolution des cours d'Apple entre juin et novembre 2020. Le conde permet de préciser le type de graphe (ici les barres *candlesicks*) avec le fond en blanc. Le code indique comment on sélectionne les indicateurs que l'on veut mettre dans le graphe, ici on prend addRSE et addBBands.

==> Ce code est intéressant car il montre qu'il est assez simple de faire une courbe d'évolution des cours des actions d'Apple. Il aurait intéressant de savoir si ce sont les mêmes paramètres qu'on utilise pour créer d'autres types de graphe d'évolution (exemple le chiffre d'affaires selon les magasins d'une marque.)

Deuxième code

```
getSymbols("AAPL",
           from='2020-01-01', to='2020-12-31')
sma <-SMA(Cl(AAPL)) #intégration dans une variable de SMA, syntaxe par défaut
tail(sma,n=5) #lecture des 5 dernières données Simple Moving Average (SMA)
```

Ce code montre comment faire pour obtenir les données en tableau et non forcément graphiquement. Le code indique la période voulue des données puis il crée une nouvelle fonction où sont integrées les données, ici *sma*. Enfin le code stipule l'intervalité des données, ici 5.

==> Ce code est intéressant car selon les personnes qui lisent les rapports, certains veulent de la datavisualisations qui permettent de comprendre rapidement les données pour prendre des décisions. D'autres au contraire sont plus dans le détail et souhaitent avoir toutes les données. Donc les avoir sous format tableau peut être pertinant.

3.3. Evaluation du travail

Sur le fond: Dès le début de la lecture,le lecteur comprend rapidement ce qu'il va trouver dans le tutorat. Chaque partie est très bien expliquée. L'auteur donne à chaque fois des éléments d'explications sur ce que le lecteur peut faire. Chaque explication est accompagnée d'exemples. Si le lecteur suit bien le tutorat, il peut reproduire facilement les exemples.

La conclusion a été bien écrite car l'auteur donne tous les éléments nécessaires pour donner envie au lecteur d'aller plus loin dans la découverte du package quantmod.

Sur la forme: Le tutorat est bien structuré, clair, aéré (séparation claire entre les paragraphes, les codes, les graphes). De plus, les codes sont faciles à comprendre. En effet, quand c'est nécessaire, les codes sont accompagnés par un # avec des commentaires pour une meilleure compréhension.

L'auteur a donné un réel exemple concret facilitant la compréhension.

Ce tutorat a été bien agréable à lire. Je tiens à souligner l'originalité du choix de ce package, qui est (je pense) connu des personnes en finance mais ne l'est pas des autres profils.

Si je devais faire une petite critique, il y a quelques fautes d'orthographes. De plus, il aurait été peut-être pertinent de commenter les graphes pour aider les lecteurs n'ont familarisé avec des notions d'actions et de bourse et souhaitant "jouer"" avec R et quandmod.

4. Package 4

Nom du package R : **lubridate**

Auteur du travail : Gaspard PALAY

Lien du travail sur Github :
https://github.com/GaspardPalay/PSBX/tree/main/TutorielLubridate

4.1. Synthèse du travail

Ce tutorat porte sur le package lubridate dans R. Il a pour but de montrer comment gérer les données temporelles telles que les dates et les heures. ce tutorat permet de mettre au propre les données temporelles avec lubridate afin de pouvoir réaliser des analyses.

4.2. Explication de certains codes

Premier code

```
t <- ymd_hms("2020.11.09_17.56.32")
date(t)
hour(t)
minute(t)
second(t)
```

Ce code montre comment on identifie une date, des heures, des minutes et des secondes. D'abord le code crée une variable qui prend comme valeur un ensemble d'éléments. Puis le package *lubridate* permet d'extraires les données de date, de l'heure, de minutes et de secondes.

Deuxième code

```
t1 <- dmy("12/09/2020")
t2 <- dmy("30/01/2016")
diff <- t1-t2
as.duration(diff)
as.period(diff)
```

Ce code montre comment on calcule une durée entre 2 dates. Le code définit une variable t1 et une autre t2 puis il soustrait t1 avec t1. La fonction as.duration calcule le durée entre 2 dates et la fonction as.period renvoie l'intervalle entre 2 dates.

Troisième code

```
t1+months(9) # t1 + 9 mois
t1+ddays(287) # t1 + exactement 287 jours
ddays(287)/dweeks(1) # combien de semaines (exactement) pour 287 jours?
t2-dweeks(7) # t2 - 7 semaines
```

Ce code montre comment on ajoute des mois, des jours et semaines à une date. La première ligne ajoute 9 mois à la date t1, puis 287 jours à t1. Ensuite le code indique combien de

semaines comportent 287 jours. Enfin la dernière ligne soustrait la variable date t2 de 7 semaines.

==> Ces 3 codes sont très intéressants. En effet, en analyse de données, les dates sont extrêment importans. Avoir des données dates propres et comme on le souhaite permet de mieux analyser des données.

4.3. Evaluation du travail

Sur le fond: Dès le début de la lecture,le lecteur comprend rapidement ce qu'il va trouver dans le tutorat. Avant chaque code, il y a des explications qui sont par ailleurs simples mais très claires. L'auteur prend bien en compte tous les éléments pouvant prêter problèmes quand on analyse des données temporelles (exemple: les années bissextiles).

Sur la forme: Le tutorat est bien structuré, clair, aéré (séparation claire entre les paragraphes, les codes, les graphes). De plus, les codes sont faciles à comprendre. En effet, quand c'est nécessaire, les codes sont accompagnés par un # avec des commentaires pour une meilleure compréhension. Cependant j'aurais bien aimé voir le résultat dans le pdf de certains codes.

Exemple page 2:

```
t <- ymd_hms("2020.11.09_17.56.32")
date(t)
hour(t)
minute(t)
second(t)
```

Ce tutorat a été bien agréable et rapide à lire. Je le trouve très utile car les données de temps sont extrêmement importantes en analyse de données. Je sais que j'aurais besoin de ce tutorat pour "nettoyer" les données de dates avant de faire n'importe quelle analyse. Si je devais faire une petite critique, il aurait été intéressant d'avoir une conclusion à la fin. De plus, j'aurais bien aimé savoir si l'auteur a inventé lui-même les exemples ou s'il a utilisé une source particulière. Enfin, attention à 2 et 3 fautes de frappe.

5. Package 5

Nom du package R : **mnist**

Auteur du travail : Kanlanfeyi KABIROU et Jordy HOUNSINOU

Lien du travail sur Github :

https://github.com/kabirou7/PSBX

https://github.com/Jordyhsn/PSB_Hounsinou

5.1. Synthèse du travail

Ce tutorat porte sur le package mnist dans R. Il a pour but d'expliquer le fonctionnement de la base de données de mnist qui sert à l'apprentissage dans les technologies de machine learning et deep learning. Le tutoriat s'axe sur l'apprentissage automatique et multiple. Il fait d'abord une présentation de la base de données mnist, puis une explication des algorithmes avec Naive Bayes et Random Forest en lien avec mnist puis des exemples de code R.

5.2. Explication de certains codes

Premier code

```
#Random Forest
rf_MNIST <- randomForest(label ~ ., data = train_mnist, ntree = 10)
pred_MNIST1 <- predict(rf_MNIST, test_mnist)
rf_FASH <- randomForest(label ~ ., data = train_fashion, ntree = 10)
pred_FASH1 <- predict(rf_FASH, test_fashion)


#Naive Bayes
bayes_MNIST <- randomForest(label ~ ., data = train_mnist)
pred_MNIST2 <- predict(bayes_MNIST, test_mnist)
bayes_FASH <- randomForest(label ~ ., data = train_fashion)
pred_FASH2 <- predict(bayes_FASH, test_fashion)
```

Ces 2 codes montrent comment on utilise les fonctions predict avec random forest et naive bayes : le premier argument à prendre en compte est le modèle puis le dataframe utilisé.

Deuxième code

```
valeurs <- matrix(c(cm_nb1$overall["Accuracy"],cm_nb2$overall["Accuracy"],cm_
rf1$overall["Accuracy"],cm_rf2$overall["Accuracy"]), ncol = 2)
colnames(valeurs)<- c("Naive Bayes", "Random Forest")
rownames(valeurs)<- c("MNIST", "Fashion MNIST")
tableau <- as.table(valeurs)
print(tableau)
```

Ce code indique tout d'abord la valeur de la matrice et le nombre de colone puis montre les valeurs en colonnes puis en lignes. Le résultat sera un dataframe avec les valeurs de la matrice.

==> Ces 2 codes semblent intéressant pour faire de la prédiction. Malheuresement, il m'est difficile de me projecter de son utilisation car toutes les explications ne sont pas données et étant une débutante sur R, je n'ai pas bien compris son utilité.

5.3. Evaluation du travail

Sur le fond: Dès le début de la lecture,le lecteur comprend rapidement ce qu'il va trouver dans le tutorat. Certaines parties sont accompagnées de commentaires de la part des auteurs du tutorat. Pour encore aller plus, il aurait été intéressant d'avoir plus d'éléments d'explications sur les résultats pour faciliter la lecture pour tous les niveaux.

Sur la forme: Le tutorat est plutôt bien structuré, clair, aéré (séparation claire entre les paragraphes, les codes, les graphes). Cependant des parties de textes ne sont pas visibles dans le pdf (exemple : page 7) mais nous pouvons retrouver l'information dans le rmd. Ce tutorat a été plutôt agréable à lire. Je tiens à souligner l'originalité du choix de ce package. En effet, la prédiction et l'apprentissage automatique est un domaine en devenir et il est intéressant d'avoir un tutorat dessus avec R. Si je devais faire une petite critique, une conclusion aurait la bienvenue comme par exemple un encourageant à aller explorer le package mnist ou donner des pistes au lecteur pour aller plus loin et plus d'explications sur certains codes.

6. Evaluation de mon travail des packages R par rapport aux 5 travaux sélectionnés

Je pense que sur la forme, le travail que nous avons réalisé avec mon groupe sur les packages R ont la même qualité que les 5 travaux sélectionnés ci-dessus. Nous avons essayé de faire un tutorat structuré, clair, aéré et fluide. Nous avons aussi cité toutes nos sources et essayer d'expliquer au moins chaque notion pour que cela soit compréhensif par tous.

Par contre, à la différence de certains travaux, le tutorat de nos packages a été fait en groupe. Or notre groupe était constitué de 3 personnes dont 3 débutants sur R et un expert en R. Cela a été très intéressant puisque l'expert a pu nous expliquer des notions et les débutants ont été de rendre le tutorat facile à lire à tous les niveaux.

Cependant par rapport à certains travaux, nous aurons pu faire une conclusion dans notre tutorat pour aider le lecteur à aller plus loin dans l'acquisition de ces connaissances.