Exam R

Arnaud FORASACCO

30/01/2021

Revue et critique des travaux R

Dans ce document, je vais effectuer les revues de 5 documents R présentés par la classe de Msc DM.

Je prendrais comme critères de notation : Présentation, Qualité, Clarté, Cohérence, Contexte

1er Travail : Rstatix fait par Zakaria

https://github.com/chaymae-data/PSBX/blob/main/Packages(KScorrect%2CInfer%2CRstatix)/rstatix.Rmd

Synthèse Le travail sur le package Rstatix est utile afin d'effectuer des tests statistiques afin de tirer des insights de données brutes. Afin de faire ces tests, il faut d'abord traiter les données et les mettre en terme "factor" afin de pouvoir les manipuler. Ensuite, nous pouvons effectuer les tests d'hypothèses ou des comparaisons de moyennes. Ce travail explique bien les différentes fonctions possibles pour les tests statistiques avec différents types de tests et si l'on recherche un thème spécifiquement, le code sera à peu de chose près similaire ce qui permettra un usage facile et compréhensible par tout le monde.

Extrait commenté des parties du code :

```
data("selfesteem", package = "datarium")
head(selfesteem, 6)
```

```
selfesteem <- selfesteem %>%
  gather(key = "time", value = "score", t1, t2, t3) %>%
  convert_as_factor(id, time)
head(selfesteem, 3)
```

La préparation de la data est essentielle afin de ne pas rencontrer d'erreurs plus tard dans le code. Une mauvaise préparation de la data est souvent source de problèmes et peut ne pas permettre de finir une éxécution.

```
bxp <- ggboxplot(selfesteem, x = "time", y = "score", add = "point")
bxp
```

La visualisation est aussi importante car elle permet de voir rapidement et efficacement s'il n'y a pas d'erreur ou d'incohérence dans le jeu de données

```
selfesteem %>%
  group_by(time) %>%
  shapiro_test(score)
```

Ici, un exemple du test Shapiro permettant de savoir si la série de données suit une loi normale et il suffit de changer la partie “shapiro_test” afin de faire un autre test.

Présentation 4/4 Qualité 4/4 Clarté 4/4 Cohérence 4/4 Contexte 3/4

Conclusion appréciation : Pour moi, ce tavail est complet et permet de facilement comprendre comment utiliser le package Rstatix et présente différents exemples d’utilisation comme pour les tests d’hypothèse ou encore les tests de calculs ANOVA.

2ème Travail : Lubridate fait par Gaspard

https://github.com/GaspardPalay/PSBX/blob/main/TutorielLubridate/TutoLubridate.Rmd

Synthèse : Le travail présenté sur le Package Lubridate permet de comprendre comment manipuler des dates dans un jeu de données comme dans des données évoluant sur le temps. Le tutoriel permet de comprendre chaque aspect du package afin de manipuler les odnnées en passant par la transformation de chaine de caractères, de vecteurs en unitées d’heure, des calculs de temps, la création d’intervalle ou encore l’utilisation de l’heure Unix. Il nous présente aussi qu’il est possible d’utilisier l’heure en temps réelle à utiliser dans du code.

Extrait commenté des parties de code :

```
jourJ <- lubridate::dmy("30 may 2020")
class(jourJ)
```

Convertir des données est quelque chose de basique mais essentiel, ce qui nous permettra par la suite de manipuler nos données.

```
ymd("2019/04_11")
ymd_hm("2019.04.11 14h37")
ymd_hms("20190407143752")
hms("14h37min52s")
```

Une heure peut être affichée sous différents aaspects et des fois certains formats seront plus adaptés que d’autres, il faut donc connaître les différents moyens de chager le format.

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

Savoir la différence de temps être un évènement et au autre peut s’avérer utile, ici ce code peut nous aider à connaître l’intervalle de temps.

```
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
```

Les calculs de durées sont utiles et évitent de perdre du temps à faire des calculs et éventuellement évite les erreurs.

Présentation 4/4 Qualité 3/4 Clarté 4/4 Cohérence 4/4 Contexte 4/4

Conclusion appréciation : Cette présentation affiche un clair aperçu des différentes possibilités présentes dans le package Lubridate fait pour la manipulation de données de temps. Les chunk sont utiles et clairs, se qui permet de vite trouver l'information qu'il nous faut.

3ème Travail : Quantmod fait par Antoine

https://github.com/aserreau/PSB1/blob/main/Travaux%20Packages/TutoPackageQuantmodFR.Rmd

Synthèse : Cette présenattion de package est très intéressante pour des personnes qui s'intéressent aux actions en bourse et plein de fonctions y sont détaillées. Nous pouvons utiliser une fonction pour voir l'historique de l'entreprise, le cours d'une action, la visualisation de données y est aussi possible. De nombreux indicateurs techniques y sont détaillés afin d'étudier les mouvements et les courbes d'une action en particulier. Les explications y sont bien détaillées, claires et précises que ce soit pour l'installation du package ou les différents étapes clés d'analyse.

Extrait commenté des parties de code :

```
getSymbols("AAPL")  #charge les données de APPLE dans votre environnement
head(AAPL)  #permet de voir les données
```

Ce passage montrant comment charger les données est une étape essentielle afin d'analyser les données financières et il suffit de remplacer "AAPL" pour trouver les données d'une autre entreprise

```
getSymbols("AAPL",
           from='2018-01-01', to='2018-12-31')  #défini l'intervalle des dates
```

Chercher une date particulière est toujours utile afin de comprendre comment une courbe pourrait évoluer et quel en serait la cause.

```
getSymbols(c("AAPL","GOOG"))
```

Il est interessant de savoir que l'on peut importer 2 jeux de données en même temps afin de comparer les courbes ou les différences de fluctuations.

```
Open <- Op(AAPL)   #Prix d'ouverture
High <- Hi(AAPL)    #Plus haut prix
Low <- Lo(AAPL)  #Plus bas prix
Close<- Cl(AAPL)   #Prix de fermeture
Volume <- Vo(AAPL)   #Volume
AdjClose <- Ad(AAPL) #Fermeture ajustée
```

Il y a ici une multitude d'options faites pour connaître les détails d'une action.

```
chartSeries(AAPL)
```

La cartographie est essentielle pour avoir une vision claire de se qui se passe avec la donnée.

addWPR()

de multiples indicaturs comme celui ci-dessus sont aussi disponibles pour une meilleure analyse.

Présentation 4/4 Qualité 4/4 Clarté 4/4 Cohérence 4/4 Contexte 4/4

Conclusion appréciation : Ce travail est complet et permet de très bien comprendre comment analyser les données financières d'une entreprise pour investir en bourse. Les explications sont claires et compréhensibles par tous.

4ème Travail : ggplot2 fait par Jiayue LIU et Soukaina ELGHALDY

https://github.com/soukainaElGhaldy/PSB-X/blob/main/R_packages/ggplot2_package/ggplot2.Rmd

Synthèse : Ce travail sur la visualisqtion de données est complet et montre diverses façons de modifier dans les moindres détails. Ce travail montre la préparation de données, les différents modèles, la customisation de couleurs, le changement d'apparence, de taille, de densité, l'ajout de labels, l'ajout de lignes de régressions, l'estimation, de densité, l'alignement de graphiques, la création de données et leurs représenation de graphiques. Les explicarions sont claires et les titoriels sont applicavles pour d'autres jeux de données.

Extrait commenté des parties de code :

```
# convertir la colonne cyl envariable de type facteur
mtcars$cyl <- as.factor(mtcars$cyl)
head(mtcars)
```

La préparation des données est essentielle afin d'effectuer une bonne visualisation, ici nous transformons en factuer masi il est aussi possible

```
#Nuage de points simple
ggplot(mtcars, aes(x=wt, y=mpg)) + geom_point()
# Changer la taille et la forme
ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point(size=2, shape=23)
```

Ici nous avons le code afin d'afficher un nuage de points classique avec la personnalisatino des points

```
# Changer la couleur et la forme des points
# Changer le type de trait et la couleur
ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point(shape=18, color="blue")+
  geom_smooth(method=lm, se=FALSE, linetype="dashed",
              color="darkred")
# Changer la couleur de remplissage de l'intervalle de confiance
ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point(shape=18, color="blue")+
  geom_smooth(method=lm,  linetype="dashed",
              color="darkred", fill="blue")
```

Une customisation est touojours uitle pour distinguer plusieurs informations.

```
# Changer le type de points en fonction des niveaux de cyl
ggplot(mtcars, aes(x=wt, y=mpg, shape=cyl)) +
  geom_point()
# Changer le type et la couleur
ggplot(mtcars, aes(x=wt, y=mpg, shape=cyl, color=cyl)) +
```

```
  geom_point()
# Changer le type, la couleur et la taille
ggplot(mtcars, aes(x=wt, y=mpg, shape=cyl, color=cyl, size=cyl)) +
  geom_point()
```

le changement de points est utile pour montrer des différences entre chaque informations afin qu'elles sont distinguables au premier coup d'oeil.

```
# Ajouter des lignes de régression
ggplot(mtcars, aes(x=wt, y=mpg, color=cyl, shape=cyl)) +
  geom_point() +
  geom_smooth(method=lm)
# Supprimer les intervalles de confiance
# Etendre les droites de régression
ggplot(mtcars, aes(x=wt, y=mpg, color=cyl, shape=cyl)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE, fullrange=TRUE)
```

Ajouter des lignes est utiles pour montrer une certaine fluctuation dans la coube ou encore pour croiser des informations ou pour montrer un objectif à atteindre.

```
library(gridExtra)
grid.arrange(xdensity, blankPlot, scatterPlot, ydensity,
        ncol=2, nrow=2, widths=c(4, 1.4), heights=c(1.4, 4))
```

Regrouper des graphiques peut s'avérer utile pour croiser des informations ou comparer des courbes agfin de trouver des insights.

Présentation 4/4 Qualité 3.5/4 Clarté 4/4 Cohérence 4/4 Contexte 4/4

Conclusion appréciation : Ce travail est complet et met en abysse de nombreuses techniques afin de faire des graphiques compréhensibles par n'importe qui. Il manque peut être quelques exemples d'autres types de graphiques (histogrammes...) mais l'essentiel y est.

5ème Travail : dplyr fait par Soukaina ELGHALDY et Jiayue LIU

https://github.com/soukainaElGhaldy/PSB-X/blob/main/R_packages/dplyr_package/dplyr-tuto.rmd

Synthèse : Ce travail portant sur dplyr, un package sur la manipulation de données présente tout un éventail de possibilités. EN, effet, il est possible de trier et de manipulaer la donnée comme on le souhaite : nous pouvons sélectionner les lignes ou colonnes souhaitées dans un tableau, nous pouvons filtrer selon une condition posée, selectionner les données que l'on souhiate obtenir afin de vite trouver ce que l'on cherche. Il est aussi possible de renommer les intitulés d'un fichier, d'arranger les colonnes afin que ce soit plus clair visuellement, la fonction mutate nous permet aussi d'ajouter de la donnée avec ou sans une variable existante, le groupement de tableaux est aussi possible, et la fonction summarize qui nous permet de faire un rapide calcul sur une condition saisie. Les détails de chaque fonctions sont bien explicites et les exemples de tutoriel sont facile à comprendre et à utiliser directement.

Extrait commenté des parties de code :

```
filter(data,age == 46)
```

Chercher de la donnée peut être compliquée dans de grandes data frame et il est facile de s'y perdre c'est pourquoi la fonction filter

```
select(data,name,age)
select(data,-name,-age)
```

Dans de grands jeux de données, toutes les informations ne sont pas utiles, la fonction “select” nous permet de choisir que certaines colonnes ou ou contraire en enlever d’autres.

```
data1 <- as_tibble(data)
data1 %>% group_by(age)
```

Le groupement de tableaux peut permettre de mieux visualiser les données et cela permet de pouvoir utiliser les données d’un tableau et d’un autre sans recréer de tableau. Effectuer la “requête” group by peut être pratique pour cela.

Présentation 4/4 Qualité 4/4 Clarté 4/4 Cohérence 4/4 Contexte 4/4

Conclusion appréciation : C’est un travail complet et très détaillé, Il est facilement possible d’utiliser ce tutoriel et l’appliquer à des cas concrets. Les explications sont claires et présentent parfaitement l’ul=tilisation de chacune des fonctions.