PARTIEL mathématique et programmation R

WILLIAM ROBACHE

30/01/2021

Les sujets sélectionnés

Sujets de mathématique:

- Régression Logistique et application avec R
- Algèbre de groupe en caractéristique 1 et distances invariantes sur un groupe fini
- K- means++
- Arbres de décision
- Regression sur une variable fonctionnelle : Estimation, tests de structures.

Sujets de R:

- Hadoop et Spark
- GGPLOT2 (Maxime & Siva)
- Forecast package et séries temporelles
- Forecast Leaflet
- GDATA PACKAGE

Mes 2 Sujets:
- Shiny
- Cryptographie

5 Critères d'évaluation

- Compréhensibilité
- Logique d'écriture
- Complexité
- Longueur
- Explications

1er sujet : Régression Logistique et application avec R

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

Synthèse de travail

La régression logistique selon Gaspard permet d'analyser une variable binaire en fonction d'une variable quantitative. Elle est utilisée dans plusieurs domaine, notamment la médecine, ou les sciences humaines et sociales. Gaspard analyse dans un premier temps le modèle mathématique de la régreession logistique, il réalise ensuite un test de régression en machine learning sur R grâce à un jeu de données du Titanic. Dans le fichier PDF, nous pouvons retrouver des équations matématiques réaliser sous Overleaf et des algorithmes R réaliser grâce à Rstudio.

Ce qu'il faut retenir

La regression logistique est un prolongement de la regression linéaire. Elle a pour hypothèse fondamentale selon le théorème de Bayes :

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

Ce théorème à 2 avantages: - Champ d'application plus large - Traitement des coefficients plus intéressante

Mais la régression logique peut s'écrire également grâce au modèle LOGIT:

$$ln[\frac{\pi(X)}{1-\pi(X)}] = a_0 + a_1 X_1 + ... + a_j X_j$$

Le rapport de chance de la fonction logistique, ODDS, s'écrit :

$$\frac{\pi(X)}{1-\pi(X)} = \frac{P(+/X)}{P(-/X)}$$

Au début de la section "Regresssion logistique avec R" Gaspard explique comment installer le Dataset, le package de machine learning pour utiliser la régression logistique et d'autres outils indispensable pour l'utilisation de la régression.

Après le néttoyage et le paramétrage du modèle:

```
#fitControl <- trainControl(method ="cv", number=10, savePredictions=TRUE)


#lr_model <- train(factor(Survived)~.,
#                  data=train,
#                  method='glm',
#                  family=binomial(),
#                  trControl=fitControl)

#summary(lr_model)
```

Ce modèle de prédiction s'est trompé sur 12 passagers en prévoyant qu'ils survivraient, alors qu'ils sont décédés. Il a prévue que 59 pesonnes survivraient alors qu'il en manque 27. Il obtient donc un score de 82% qui est améliorable grâce à du fitter engineering.

Notation

- Compréhensibilité 3/4
- Logique d'écriture 4/4
- Complexité 2/4
- Longueur 4/4
- Explication 3/4

Note: 16/20

Conclusion

Gaspard a effectué un travail sur la régression logistique très correct, il est clair et simple dans ses explications. Gaspard aurait cependant pû entrer un peu plus dans le détail des explications mathématique. L'utilisation d'un algortithme de prédiction en R est un vrai plus dans sa présentation.

2ème sujet : Algèbre de groupe en caractéristique 1 et distances invariantes sur un groupe fini

Auteur: Marion DANYACH
Lien github : https://github.com/OlfaLmt/PSBX/blob/main/Th%C3%A8ses/Algebre%20de%20groupe%20en%20caracteristique%201%20et%20distances%20invariantes%20sur%20un%20groupe%20fini.pdf

Synthèse de travail

La présentation de Marion à pour sujet un papier de recherche de Dominique Castella et Stephane Gaubert à propos de l'algèbre de groupe en caractéristique 1 et distances invariantes sur un groupe fini. Tout au long de sa présentation, marion explique comment comprendre et utiliser le papier de recherche. Elle commence par expliquer l'algébre tropicale en introduction. En 2ème partie, elle explique les insights du papier, en 3ème partie elle fait un zoom sur quelques formulations mathématiques, et enfin elle donne son avis sur le papier.

Ce qu'il faut retenir

L'algèbre tropicale inventé par Imre Simon (Brésilien) re-définit l'addition et la multiplication ainsi que toutes les opérations associées.

Pour exemple :

$$a \oplus b = max(a, b)$$

$$a \odot b = a + b$$

Deux concepts sont importants à retenir: L'algèbre min-plus définie avec le minimum pour addition et l'addition pour la multipliation. L'algèbre max-plus, définie avec le maximum pour addition et l'addition pour la multiplication.

Définition l'idempotence: C'est le fait qu'une opération aura le même effet si elle est appliquée une ou plusieurs fois. Pour exemple:$abs(abs(X)) = abs(X)$

On a donc un élément x d'un magma $(M, \bullet)$ idempotent si $1, 2 : x \bullet x = x$ Si tous les éléments de M sont idempotents pour $\bullet$, alors $\bullet$ est dite idempotente.

L'algèbre tropicale est utilisé dans la modélisation, l'analyse, l'évaluation de performance et la synthèse de lois de commande (ateliers flexibles, réseaux de tranport, résaux de communications etc) Par exemple, Un script SQL est écrit de façon idempotent pour ne pas modifier la base de données.

Si e et f sont deux idempotents centraux, $e + f$ est idempotent si et seulement si

$$ef = e + f$$

Un idempotent e, tel que $N(e) \neq 1k$ est irréductible s'il est central et vérifie la condition $e = fg$ où f et g sont deux idempotent centraux.

Notation

- Compréhensibilité 4/4
- Logique d'écriture 3/4
- Complexité 1/4
- Longueur 3/4
- Explication 3/4

Note: 14/20

Conclusion

La présentation de Marion explique le papier de Dominique Castella et Stephane Gaubert. Sa présentation est simple et compréhensible, cependant, on aurait aimé un peu plus d'explication et une utilisation concrète de l'algèbre tropical avec un exemple détaillé.

3ème sujet : K- means++

Auteur: Lucas Billaud
Lien github : https://github.com/Lucasblld/PSBX/blob/main/maths/k-means.pdf

Synthèse de travail

Le contenu de la présentation a pour sujet principal une analyse d'un papier de recherche de David Arthur et Sergei Vassilvitskii à propos de K- means++. Il commence par donner quelques définitions autour de k-means, il explique ensuite les problème de k-means et comment l'optimiser grâce à k-means++ et enfin il parle des résultats. Pour expliquer des concepts mathématiques, il utilise des graphiques de représentation et des formules LateX. L'objectif est de comprendre comment k-means ++ permet une amélioration de k-means en expliquant certaines démonstrations mathématique faites par David Arthur et Sergei Vassilvitskii.

Ce qu'il faut retenir

La méthode k-means est une technique de machine learning non supervisée permettant de créer des clusters. K-means utilise la distance euclidienne entre chaque points. Ce qui lui permet d'avoir une rapidité d'implémentaton dans la pratique. k-means utilise une complexité quadratique $O(n2)$ basé sur la minimisation des distances

Cependant, cet algorithme rencontre un problème de placement initial des centroids car ils sont placés de façon aléatoire. L'objectif est donc d'augmenter la précision et la rapidité de l'algorithme en pondérant le choix de positionnement des centroids grâce au théorème suivant:

Theorem

$If\ C\ is\ constructed\ with\ k-means++,\ then\ the\ corresponding\ potential\ function\ \phi\ satisies,\ E[\phi] \leq 8(lnk+2)\phi OPT$

La première partie de la démonstration du théorème permet de montrer que les clusters sont plus précis avec k-means++ grâce notamment à l'équation suivante :

$$E[\phi(A)] = 2\sum ||a - c(A)||\text{š}$$

Ce qui correspond à un cluster optimal.

La 2ème partie de la démonstration permet de prouver que K-means++ est de complexité $O(log)$

Dans la dernière partie de la présentation, un tableau montre les résultats de K-means++, qui a une meilleur performance par rapport à k-means

Notation

- Compréhensibilité 4/4
- Logique d'écriture 3/4
- Complexité 3/4
- Longueur 4/4
- Explication 4/4

Note: 18/20

Conclusion

Pour conclure, Lucas a réalisé une présentation de k-means++ très détaillée et facile à comprendre. Chaque notion qu'il aborde est très bien expliquée. Un exemple détaillé aurait pu être rajouté.

4ème sujet : Arbres de décision

Auteur: Rindra LUTZ / Nicolas ALLIX
Lien github : https://github.com/rindra-lutz/psb1/blob/Travaux-Math%C3%A9matiques/Arbres-de-D%C3%A9cision.pdf

Synthèse de travail

Rindra et Nicolas ont réalisé un travail sur les arbres de décision. Ils commencent par expliquer ce qu'est un arbre de décission, d'où il vient et comment le construire. Ils parlent ensuite des avantages de l'arbre de décission, mais aussi des ses limites. Et enfin, ils abordent les arbres de classification et ils font une démonstration par l'exemple. Pour présenter leurs propos, ils utilisent des graphes et des formules Latex. À première vue, le travail sembre clair et aéré avec une longueur acceptable.

Ce qu'il faut retenir

Dans un arbre de décision ils existent 3 types de noeuds :

- Le noeuds racine : l'accès à l'arbre se fait par ce noeud
- Les noeuds internes : les noeuds qui ont des descendants
- Les noeux terminaux : noeuds qui n'ont pas de descendant

Il existe 2 types d'arbres pour la prédiction :

- Les arbres de régressions, qui permettent de prédire une réponse quantitative
- Les arbres de classifications, qui permettent de prédire une réponse qualitative

L'indice de Gini permet de mesurer la pureté d'un noeud. Plus la valeur de l'indice est proche de 0, plus le noeud est pur :

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

Un noeud est pur si tous les individus associés appartiennent à une même classe.

Pour mesurer la pertinence du choix de la variable de décision, il faut calculer le coût du noeud:

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

$G(gauche \ \ et \ \ droite)$ permet de mesurer l'impureté des noeuds descendants et $m(gauche \ \ et \ \ droite)$ est la représentation de la proportion de la population sur les noeuds.

Avantages d'un arbre de décision:
- Mise en oeuvre simple
- Facilité de prise de décision
- Décisions comlexes simplifiées

Inconvénients:
- Instable
- Problème de sur-apprentissage

Notation

- Compréhensibilité 4/4
- Logique d'écriture 3/4
- Complexité 2/4
- Longueur 4/4
- Explication 4/4

Note: 17/20

Conclusion

Le travail de Rindra et Nicolas est très bien expliqué et compréhensible. Les formules sont détaillées ainsi que les représentations graphiques. Un exemple d'arbre de décision permet de mieux comprendre le sujet, c'est un vrai plus. Il manque cependant la bibliographie à la fin du document.

5ème sujet : Regression sur une variable fonctionnelle : Estimation, tests de structures.

Auteur: ALLAKERE HORMO MAXIME
Lien github : https://github.com/mallaker/PSB_X/blob/main/Dossier%20Maths/REGRESSION%20SUR%20UNE%20VARIABLE%20FONCTIONNELLE.pdf

Synthèse de travail

La régression est une méthode d'analyse statistique permettant d'approcher une variable à partir d'autres qui lui sont corrélées. Un test statistique est une procédure de décision entre deux hypothèses. Il s'agit d'une démarche consistant à rejeter ou non une hypothèse statistique dite hypothèse nulle en fonction d'un jeu de données (échantillon)

Dans sa présenation, Maxime introduit d'abord la régression sur une variable fonctionnelle, il parle ensuite des problèmatiques, et de la régression fonctionnelle. Et enfin, il fait un test de structure de la régression fonctionnelle.

Ce qu'il faut retenir

La formule de régression s'écrit : $Y = r(X) + \epsilon$ La variable Y est la valeur réelle La variabl X est la valeur explicative dans un espace semi-métrique $(E,d) de dimension infinie ϵ représente le résidu.

Équation paramétrique

$$Y = aX + b + \epsilon, X(\mu, \gamma^2) \quad et \quad N(0, \sigma^2)$$

Équation non-paramétrique

$$Y = r(X) + \epsilon, r \in C(\mathbb{R} \quad et \quad E(\epsilon|X) = 0$$

Équation paramétrique et non-paramétrique selon 2 points de vue

$$Y = aX + b + \epsilon, E(\epsilon|X) = 0$$

-Non-paramétrique si la loi du couple (X,Y) est supposé appartenir à un espace indexé. -Paramétrique si l'opérateur de régression r est supposé appartenir à un espace indexé par un nombre fini de paramétriques réels.

Dans le document de Maxime il n'y a pas d'autres formules mathématiques.

Notation

- Compréhensibilité 2/4
- Logique d'écriture 2/4
- Complexité 4/4
- Longueur 4/4
- Explication 2/4

Note: 14/20

Conclusion

Le document de Maxime est assez complexe mais il et possède peu de formules mathématiques ou graphique pour aider à la compréhension. Il utilise un test de structure pour aider à la compréhension. Il manque également la bibliographie.

6ème sujet : HADOOP et SPARK

Auteur: Florine COMLAN, Ramya HOUNTONDJI
Lien github : https://github.com/fcom-stack/PSBX/blob/main/Hadoop_et_spark/HadoodEtSpark.pdf

Synthèse de travail

Ramya et Florine, ont travaillé sur Hadoop et Spark en particulier qui fait parti de son écosystème. En premier temps ils parlent de Hadoop et le BigData, ensuite ils expliquent les composant de base d'un cluster Hadoop. En 3ème partie ils présentent Hadoop et son écosystème, ensuite Hadoop vs Spark puis Comment installer Hadoop avec Docker et enfin ils présentent les packages.

Dans leurs présentations ils utilisent des graphiques et des morceaux de codes R pour expliquer les différents concepts.

Ce qu'il faut retenir

Tout d'abord, ils expliquent comment installer Hadoop avec Docker via un site internet, ils expliquent ensuite comment créer les trois contenaires à partir de l'image téléchargée:

```
#docker pull liliasfaxi/spark-hadoop:hv-2.7.2
```

↪Permet d'installer l'executable qui contient Hadoop, Spark, Kafta et HBase

```
#docker network create --driver=bridge hadoop
```

↪Permet de créer un réseau qui relie les 3 contenaires

```
#docker run -itd --net=hadoop -p 50070:50070 -p 8088:8088 -p 7077:7077 -p 16010:16010 \
#            --name hadoop-master --hostname hadoop-master \
#           liliasfaxi/spark-hadoop:hv-2.7.2
#docker run -itd -p 8040:8042 --net=hadoop \
#        --name hadoop-slave1 --hostname hadoop-slave1 \
#              liliasfaxi/spark-hadoop:hv-2.7.2
#docker run -itd -p 8041:8042 --net=hadoop \
#        --name hadoop-slave2 --hostname hadoop-slave2 \
#              liliasfaxi/spark-hadoop:hv-2.7.2
```

↪Permet de créer et lancer les trois contenaires (les instructions -p permettent de faire un mapping entre les ports de la machine hôte et ceux du contenaire):

Ensuite ils expliquent comment utiliser le contenaire master:

```
#docker exec -it hadoop-master bash
#Permet d'éxécuter le contenaire
```

Ensuite ils expliquent comment ajouter des données dans HDFS:

```
#hadoop fs -mkdir -p input
```

↪Permet de créer un répertoire dans HDFS, appelé input

```
#hadoop fs -put purchases.txt input
```

↪Permet de charger le fichier purchase.txt dans le répertoire input créé.

```
#hadoop fs -ls input
```

↪Permet d'afficher le contenu du répertoire input

```
#hadoop fs -tail input/purchases.txt
```

↪Permet d'afficher les dernières lignes

Ensuite, nous avons un tableau avec les commandes les plus utilisés pour manipuler les fichiers dans HDFS sous forme d'image.

Notation

- Compréhensibilité 3/4
- Logique d'écriture 3/4
- Complexité 3/4
- Longueur 4/4
- Explication 4/4

Note: 17/20

Conclusion

Présentation très clair dans l'ensemble, nous aurions aimé une présentation sous forme d'exemple. Les images ajoutent beaucoup de compréhension aux explications.

7ème sujet : GGPLOT2

Auteur: Maxime & Siva
Lien github :https://github.com/mallaker/PSB_X/blob/main/Package%20ggplot2/Package_ggplot2.pdf

Synthèse de travail

Ils commencent dans l'introduction à expliquer ce qu'est GGPLOT2, quelles sont Ses mécaniques. Ensuite ils présentent comment installer GGPLOT2 puis sur 8 chapitres ils expliquent le fonctionement opérationnel de GGPLOT2:
- Les graphiques conventionnelles
- Tracé par défaut
- Modifications des legendes
- Nuages de points et tracés de gigue
- Diagramme à barres et histogrammes
- Graphiques à secteurs
- Graphiques à bulles
- Graphiques divergents

La présentation est composée de beaucoup d'illustration par image qui correspondent à des lignes de codes ce qui augmente le nombre de page.

GGPLOT2 est un package en R spécialement conçu pour la visualisation de données. Il fonctionne en couche, en commençant par une couche montrant les données brutes collectées puis en ajoutant des couches d'annotations et de résumés statistiques.

Ce qu'il faut retenir

```
#install.packages(ggplot2)
#library(ggplot2)
```

↪Permet d'installer GGPLOT2

```
#airquality$Max <- NA
#airquality$zone <- cut(airquality$Wind, breaks=quantile(airquality$Wind))
#levels(airquality^zone) <- c("Est","Nord","Ouest","Sud)
#table(airquality$zone, airquality$Wind)
```

↪Permet créer un graphique permettant de mettre en relief des données (ici le vent) dans chaque zone créée (Nord, Sud, Est, Ouest). La première ligne permet de créer une nouvelle variable dans un dataframe, et la dernière permet d'avoir une idée sur la quantité de vent dans chaque zone.

```
#Boxplot(wind~zone, data=airquality)
```

↪Permet de visualiser le graphique avec les valeurs aberantes.

```
#H <- ggplot(iris, aes(Sepal.Length, Petal.Length))+ geom_point()
```

↪Permet de créer un nuage de point

```
#J <- ggplot(iris, aes(Sepal.Length, Petal.Length))+geom_point(shape=1)
```

↪Permet d'ajouter des attributs sur les points tracés ↪colour=species permet de rajouter des couleurs dans la syntaxe aes

```
#grid.arrange(H,J)
```

La varation des graphiques cause les graphiques divergents

```
#mtcars$'car name' <- rownames(mtcars)
```

↪Permet de créer une nouvelle colonne "car name"

```
#mtcars$mpg_z<-round((mtcars$mpg - mean(mtcars$mpg))/sd(mtcars$mpg),2)
```

↪Permet de calculer les mpg normalisés (km)

```
#mtcars$mpg_type<-ifelse(mtcars$mpg_z<0,"below","above")
```

↪Permet une classification < et > à 0

```
#mtcars<-mtcars[order(mtcars$mpg_z),]
```

↪Permet de trier

```
#mtcars$'car name'<-factor(mtcars$'car name', levels=mtcars$'car name'
```

↪Permet de convertir les valeurs de la nouvelle colonne en facteur pour le traitement et le tracé

Notation

- Compréhensibilité 4/4
- Logique d'écriture 3/4
- Complexité 3/4
- Longueur 4/4
- Explication 4/4

Note: 18/20

Conclusion

Présentation clair et simple à comprendre, il y a beaucoup d'illustration ce qui permet une meilleur compréhension. Ils utilisent un dataset pour exliquer GGPLOT2. Il manque cependant la bibliographie.

8ème sujet : Forecast package et séries temporelles

Auteur: Arnaud et Nina
Lien github : https://github.com/ArnaudFrsc/PSBX/blob/main/Forecast_Pack.pdf

Synthèse de travail

Le package Forecast permet d'analyser des prévisions de séries temporelles univariées. Dans leur présentaiton, Nina et Arnaud commence par expliquer comment importer le package et les données, ensuite ils expliquent comment créer une série temporelle. Et ils concluent par expliquer comment faire des prédictions.

Leur présentation fait 12 pages, avec plusieurs graphiques d'explications et lignes de codes.

Ce qu'il faut retenir

```
#attach(groupe_st)
#str(groupe_st)
```

↪ Permet d'afficher les données de la dataframe "groupe_st"

```
#names(groupe_st)[1]="Mois"
#names(groupe_st)[2]="CDG"
#names(groupe_st)[3]="ORLY"
```

↪ Permet de modifier l'intitulé des variables

```
#groupe_st$Total=sub(" ","",groupe_st$Total)
#groupe_st$Total<-as.factor(groupe_st$Total)
```

↪ Permet de modifier le type des données en chiffres.

```
#dy<- diff(Y)
#autoplot(DY)+ggtitle("Change in Air traffic") + ylab("Flights")
```

↪ Permet une représentation précise de la data selon la différence.

Les prédictions
naive model C'est une méthode de prédiction dans laquelle les chiffres réels de la dernière période sont utilisés comme prévision.

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

↪Permet de faire une prédictions selon la “naive model”

```
#fcts_n<-forecast(fit_n, h=12)
#autoplot(fcts_n)
```

↪Permet de faire une représenation graphique de la “naive model”

ETS model
C’est une fonction statistique qui permet de prédire une valeur future en fonction des valeurs existantes selon la version AAA de l’algorithme de lissage exponentiel.

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

↪Permet d’utiliser la méthode “ETS model” pour faire une prédiction

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

↪Permet de faire une représentation graphique selon la méthode “ETS model”

Arima model

```
#fit_arima<-auto.arima(Y)
#print(summary(fit_arima))
```

↪Permet d’utiliser la méthode de prédiction Arima

```
#fcts_arima<-forecast(fit_arima, h=12)
#autoplot(fcts_arima)
```

↪Permet de faire la représentation graphique de la méthode Arima.

Notation

- Compréhensibilité 3/4
- Logique d’écriture 3/4
- Complexité 3/4
- Longueur 4/4
- Explication 2/4

Note: 15/20

Conclusion

Le travail de Arnaud et Nina est très complet, cependant nous aurions aimé un peu plus d’explication. Leurs explications sont claires et précises notamment grâce à l’utilisation de beaucoup de graphique qui est un vrai plus pour la compréhensibilité

9ème sujet : Forecast Leaflet

Auteur: Arnaud et Nina
Lien github : https://github.com/ArnaudFrsc/PSBX/blob/main/Leaflet_Pack.pdf

Synthèse de travail

Tout d'abord, leur présentation est en Anglais. Ils commencent par faire une petite présentation de Leaflet, ensuite ils expliquent comment l'installer, et enfin comment l'utiliser. Leaflet est une librairie très populaire pour réaliser des cartes interactives. C'est une librairie open-source Java script utilisée par The New York Times ou The washington Post par exemple.

Dans leur présentation, Nina et Arnaud utilise beaucoup de graphiques / cartes pour expliquer leurs propos.

Ce qu'il faut retenir

```
#m0<-leaflet()%>%
#  addTiles
#m0
```

↪Permet de créer une variable et la map par défaut de Leaflet

```
#data<-[complete.cases(data),]
```

↪ Permet de vérifier qu'il n'y a pas de valeurs manquantes

```
#city[is.na(city)]<-0
```

↪Permet de remplacer les N/A par 0

```
#m1<-leaflet()%>%
#  addTiles()%>%
#  addMarkers(data=data,lng=~long,lat=~lat,popup=~last_name)
```

↪Permet de placer les marqueurs en fonction de leur lattitude et longitude.

```
#m2<-leaflet()%>%
#   addProviderTiles("Esri.Delorme")
```

↪Permet de modifier le style de la map avec le style "Esry.Delorme" qu'il faut télécharger. Et donc de créer des clusters de position.

```
#map<-leaflet()%>%
# addTiles()%>%
# setView(lng=78.0419, lat=27.1750, zoom=15)
# addMarkers(map=map, lng=78.0419, lat=27.1750, popup="Taj Mahal")
```

↪Permet de pointer sur la location du TajMahal et d'y placer un marqueur

Notation

- Compréhensibilité 4/4
- Logique d'écriture 3/4
- Complexité 2/4
- Longueur 3/4
- Explication 3/4

Note: 15/20

Conclusion

Le travail de Nina et Arnaud est clair et simple à comprendre. On aurait aimé cependant un travail un peu plus poussé, c'est un peu court. La présentation est en anglais, ce qui ne nuie pas à la présentation.

10ème sujet : GDATA PACKAGE

Auteur: Akram Bensalem
Lien github :https://github.com/AkramBensalemPSB/PSB/blob/main/package-gdata.pdf

Synthèse de travail

Gdata est un package simple et intuitif qui permet de manipuler des bases de données facilement et faire de la data visualisation. Tout d'abord il commence par expliquer comment installer le package. Ensuite, il explique comment la visualiser basiquement et enfin comment la manipuler. Gdata est téléchargeable depuis CRANou directement sur R-studio.

Ce qu'il faut retenir

```
#x<-as.data.frame(df)
```

↪Permet de caster la base de données pour qu'elle soit reconnue en tant que DataFrame.

```
#ll(x, dim=TRUE, sort=FALSE)
```

↪Permet de visualer globalement la dataframe, c'est à dire le poids des colonnes, le nombre de charactère.

```
#first(x,n=4)
```

↪ Permet de visualiser les 4 premières lignes de la dataframe

```
#left(x,n=2)
```

↪Permet de visualer les 2 dernières colonnes.

```
#duplicated2(x$CustomerID, bothWays=FALSE)
```

↪ Permet de renvoyer les doublons.

```
#unique(x$CustomerID, incomparables=FALSE)
```

↪Permet de supprimer les doublons

```
#rename.vars(x,from="CustomerID", to="idclient")
```

↪Permet de changer le nom d'une colonne.

```
#is.what(x)
```

↪Permet de connaître le type de l'objet qu'on manipule.

```
#unknownToNA(x$UnitPrice,unknown="0.00")
```

↪Permet de remplacer les valeurs nulles par N/A dnas la colonne UnitPrice

```
#bindData(x,y,common)
```

↪Permet de faire la jointure entre deux tables (x et y) par une clé colonne (jointure par défaut)

```
#db1<-list(âge=26,taille="grand",fumeur="oui",1,16)
#db2<-list(taille="petit", 3, fumeur="non",4)
#update(db1, db2)
```

↪Permet de mettre à jour certaine donnée dans une liste.

Notation

- Compréhensibilité 3/4
- Logique d'écriture 3/4
- Complexité 3/4
- Longueur 3/4
- Explication 3/4

Note: 15/20

Conclusion

La présentation de Akram est plutôt bien expliquée et claire. Les parties s'enchaîne avec logique et facilité. Il utilise un exemple pour faire sa présentation, mais nous aurions aimé également un cas concret avec un peu plus de données et toutes les possibilités de Gdata. Il manque aussi la bibliographie à la fin de la présentation.

11ème sujet : Shiny

Auteur: Rindra Lutz / William Robache
Lien github :https://github.com/WilliamRbc/PSBX/blob/main/package/Shiny.pdf

Synthèse de travail

Shiny est un package qui permet de partager ses projet R, de créer des applications pour le web. Il a été développé par les équipes de Rstudio. Tout d'abord dans notre présentation nous commencons par exepliquer comment fonctionne Shiny et comment l'installer. Ensuite nous expliquons Shiny via un exemple concret du point de vue UI et serveur. Et enfin, nous donnons quelques informations utiles. Le package est simple d'utilisation et ne demande pas de faire appel à du HTML / CSS.

Ce qu'il faut retenir

```
#install.packages("shiny")
#library(shiny)
```

↪Permet d'installer Shiny. À noter que shiny est pré-installé dans Rstudio.

Interface utilisateur (Permet de stocker les éléments graphiques de l'interface)

```
#  ui <- shinyUI(fluidPage(
#    titlePanel("Aperçu d'un dataset"),
#    sidebarLayout(
#      sidebarPanel(
#        textInput(inputId = "lignes",
#                  label = "Combien de lignes voulez-vous voir ? ",
#                  value = 10),
#        selectInput(inputId = "labs",
#                    label = "Dataset",
#                    choice = c("cars","rock","beaver1", "sleep")
#        )
#      ),
#      mainPanel(
#        tableOutput("dataset")
#      )
#    )))
```

↪Ce code permet de : - définir l'interface utilisateur
- de chosir le titre (Aperçu d'un dataset)
- d'indiquer le layout
- de définir la sidebar de contrôle
- d'insérer un champ et un menu déroulant
- d'indiquer l'élément qui sera dans la fenêtre principale (le dataset sous forme de graph)

Serveur (Endroit où les éléments seront crées et les composantes seront appelées pour être redirigées dans l'interface utilisateur)

```
#server <- shinyServer(function(input, output) {
#  output$dataset <- renderTable({
#    if(input$labs == "cars"){
#      print(head(cars, input$lignes))
#    } else if(input$labs == "rock"){
#      print(head(rock, input$lignes))
#    } else if(input$labs == "sleep"){
#      print(head(sleep, input$lignes))
#    } else {
#     print(head(beaver1, input$lignes))
#    }
#  })
#})
```

↪ Ce code quant-lui permet de : - communiquer avec l'interface utilisateur via l'élément 'output'
- d'imprimer les éléments d'après les données en entrée. Ils sont appellés avec 'input' puis le nm de la composante de UI ('labs' et 'lignes')

```
#shinyApp(ui=ui, server=server)
```

↪Permet de voir l'application en local

```
#dashboardPage()
#dashboardHeader()
#dashboardSidebar()
```

↪Ces balises permettent d'imbriquer des éléments (ici : une page, un bandeau d'en tête et une barre latéral)

Notation

- Compréhensibilité 4/4
- Logique d'écriture 3/4
- Complexité 3/4
- Longueur 3/4
- Explication 4/4

Note: 17/20

Conclusion

Notre présentation sur le package Shiny est plutôt claire et bien expliquée. L'exemple concret permet de bien comprendre le package. Cependant, nous aurions aimé rajouter des image de l'interface avec un exemple un peu plus complexe pour voir toutes les possibilités. Il manque également la bibliographie.

12ème sujet : Cryptographie

Auteur: Marko Arsic / William Robache
Lien github :https://github.com/ARSICMrk/ARSIC_PSBx/blob/main/Maths_BD/Cryptographie/Cryptographie.pdf

Synthèse de travail

Notre présentation sur la cryptographie comporte plusieurs sections. Nous commençons par les débuts de la cryptographie, avec les premiers moyen de codage. PLus nous avançons dans la présentation, plus nous évoquons des moyens de cryptage récent et complexe, jusqu'à la cryptographie quantique.

La cryptograhie est apparue au milieu du XXème siècle. Au début, elle était réservée aux militaires. Elle s'est ensuite démocratisée dans les banques, ce qui a révolutionné le secteur. Elle s'est également développée dans plusieurs autres secteur.

Ce qu'il faut retenir

Principe de chiffrement

$$f_D(M) = f_D[F_C(m)] = m$$

Cette formule correspond à l'échange entre un émetteur et un récpteur du message m. Le récepteur commence par annoncer publiquement sa fonction de cryptage f_C et garde en privée sa fonction de décryptage f_D.

Un émetteur peut alors envoyer le message m en envoyant $M = f_C(m)$ Si quelqu'un intercepe le message, il ne pourra donc pas le lire car il n'a pas la fonction de décryptage f_D.

Le système RSA

Le système RSA repose sur une fonction à sens unique avec trappe. Il peut être utilisé comme mécanisme de chiffrement ou signature.

La fonction de cryptage est l'exponentation à la puissance e modulo n
e = clé d'encryptage n = modulo du cryptage

La fonction de décryptage est l'exponentation à la puissance d modulo$n$
d = clé de décryptage

n doit être le produit de deux nombres premiers p et q

Et e et d doivent être choisis de façon à ce que

$$e.d \equiv 1[(p-1)(q-1)]$$

Le système RSA est donc basé sur la difficulté de factoriser le modulo n et de retrouver les premiers p et q qui le composent. Ce système peut être utilisé pour du chiffrement ou de la signature numérique.

Avantage: RSA peut être utilisé seulement si la clé publique provient de l'émetteur.
Inconvénient: Protocole très lent pour échanger en temps réel sur internet.

Notation

- Compréhensibilité 3/4
- Logique d'écriture 4/4
- Complexité 3/4
- Longueur 3/4
- Explication 4/4

Note: 17/20

Conclusion

Notre présentation est assez logique et simple à comprendre. Nous aurions amié un peu plus de formule mathématique et une description par l'exemple aurait pû être incorporé.

Évaluation objectif de mes travaux.

Les présentations que j'ai réalisé avec Rindra (Shiny) et Marko (Cryptographie) sont très différentes. La présentation sur la cryptographie contient beaucoup de texte et des explications des différents moyen de cryptage. Quant-à la présentation de shiny, elle est plus courte mais plus concrète avec un exemple détaillé. Elles ont toutes les deux des défauts et des qualités, il aurait fallu les combiner pour avoir la présentation idéal. Le travail des autres étudiants est très qualitatif dans l'ensemble et mérite d'être salué.
Shiny aurait pû être un peu plus développé, cependant son exemple concret est un vrai plus pour la compréhension.
La cryptographie est assez bien détaillée et expliquée, cependant il manque un travail concret de cryptage d'une donnée. L'utilisation de Latex a bien été respecté. Shiny et la cryptographie correspondent approximativement à la moyenne des autres travaux.

Cet exercice de correction des travaux m'a permis de comprendre tous les travaux que j'avais sélectionné. R et les mathématique sont 2 langages différents mais complémentaires.