synthèse

Groupe 8 : KABASSINA Ange & AYANOU Samuel & DIAKITE Moussa

2023-07-05

Table of Contents

# 1 Exposé 1: Janitor

Il ressort de l’exposé du groupe 1 que Janitor est un package conçu par des statisticiens pour des statisticiens notamment pour faciliter de traitement des données d’enquête. Ainsi, Janitor propose deux (2) grandes familles de fonctions

## 1.1 Les fonction de nétoyage des données

Cette première famille fournit des fonctions qui permettent de faciliter l’apurement des données. C’est le cas par exemple des fonctions : • Clear\_names () : qui supprime les caractères spéciaux (come %, @, $) du nom des variables et réorganise la casse ;

* Remove\_empty () : qui supprime les variables avec 100 % de valeurs manquantes ;
* remove\_constant () : qui supprime les colonnes avec des valeurs constantes c’est à dire identiques pour tous les individus. Il faut noter que cela est très utile puisque ce genre de colonnes n’intéresse pas le statisticien car ce dernier travaille sur des variables donc des colonnes avec des valeurs qui devraient varier suivant les individus
* get\_dupes () : qui renvoie les doublons c’est à dire l’ensemble des lignes qui se répètent dans la table de données.

## 1.2 Les fonction de tableaux

Cette famille de fonction quant à elle fournit un ensemble de fonctions pour faire des opérations sur les tableaux (croisement, …). Nous pouvons citer entre autres fonctions tabyl (), adorn\_totals, adorn\_percentages (), adorn\_pct\_formatting, adorn\_ns (), adorn\_title () … Cependant il faut noter que le but de Janitor c’est plus l’analyse de données que la tabulation d’autant plus qu’on ne peut tabuler qu’au plus deux variables à la fois, d’où la nécessité pour obtenir de jolis tableaux de l’utiliser conjointement avec d’autres packages comme “gt” ou “gtsummary” que nous verrons tout de suite après.

# 2 Exposé 2 : Gtsummary

Comme mentionné plus haut, gtsummary est un package qui intervient dans la tabulation, mais il faut noter qu’au début qui a été créer pour faire de l’évaluation d’impact. L’évaluation d’impact est une évaluation qui essaie d’établir un lien causal entre un programme ou intervention et un ensemble de résultats. Elle fournit des informations sur les effets induits par une intervention et a pour but d’analyser l’impact d’un projet, d’un programme ou d’une politique sur le groupe cible et de mesurer l’ampleur de cet impact d’où le format un peu spécial de gtsummary. En effet, on peut fournir en entrer plusieurs variables dont une dichotomique et il sortira des statistiques suivant les modalités de la variable dichotomique tout en précisant la différence des moyennes entre les deux groupes de même que la significativité de cette différence (p-value). Ce qui est un clin d’œil à l’évaluation d’impact où on a deux groupes l’un ayant bénéficié de l’intervention et l’autre non et l’on cherche à analyser l’écart (donc la significativité de la différence) de performance pour certains indicateurs entre ces deux groupes.

Par ailleurs il faut noter que le package permet également de sortir les statistiques suivant les modalités d’une variable catégorielle que cette dernière soit dichotomique ou non. Cependant la différence et la p-value dont nous venons de parler ne seraient plus pertinente (pas interprétables). D’ailleurs lorsque la variable n’est pas dichotomique gtsummary ne donne plus ces informations.

Une autre force de gtsummary est qu’il permet contrairement à Janitor de tabuler plus de deux variables en même temps. On peut également sélectionner des modalités pour ne tabuler que ces dernières, combiner plusieurs tableaux ou encore mettre en forme les tableaux sortants etc…

Concrètement, ce travail est fait grâces à des fonctions telles que : • La fonction tbl\_summary qui donne un resumé des variables multicotomiques ; - la fonction tbl\_cross permet de faire de tableau croisé ;

* La fonction tbl\_continuous et tbl\_custom\_summary () permet de fournir la statistique relative à une variable continue ;
* Les régressions se font grâce à la fonction tbl\_regression () ;
* tbl\_unvregression réalise plusieurs régressions univarié ;
* tbl\_stack() permet de coller deux ou plusieurs tableaux ;
* tbl\_merge pour fusionner des tableaux

Ces beaux tableaux après avoir été créés, peuvent être exporter en plusieurs formats : Word, PDF… Mais ne serait-il pas plus intéressant de les intégré directement dans un document écrit sur R ? Eh bien cela est possible grâce à R-markdown et Quarto.

# 3 Exposé 3 : R-markdown

Nous avons sûrement presque tous déjà entendu parler de Word pour la rédaction de document ou de PowerPoint pour faire des présentations. Eh bien il est tout à fait possible de travailler entièrement sur R depuis l’analyse des données jusqu’à la rédaction d’un document final ou d’une présentation grâce à R-markdown. De plus l’un des avantages de R-markdown c’est que les documents qu’il produits sont reproductibles c’est à dire que si tu remets le code source à un collaborateur, il peut exécuter et obtenir le même document ou changer la base de données par exemple et exécuter pour obtenir un document du même format (seulement les résultats c’est à dire les chiffres, graphiques etc. seront différent).

Pour créer un nouveau fichier R-markdown il faut aller dans la menue file ensuite new file - R-markdown et spécifier le nom de l’auteur, la date et le format de la sortie. Une fois créé vous pourrez remarquer que le fichier a 3 grande partie :

* Le yaml qui marque le début du document avec les informations sur la sortie
* La partie après le yaml où on écrit du texte
* Les chunks où on peut écrire du code r, python, bash, etc…

Ainsi on peut avoir en fonction des sorties un document ou une présentation

## 3.1 R-mardown document

La présente synthèse a été écrite de bout en bout sur R-markdown, vous avez donc la possibilité d’écrire vos documents et de les exporter sous format Word, PDF ou encore html. Notons tout de même que le format Word est à privilégier dans le cadre professionnel pour permettre aux collaborateurs d’y apporter leurs modifications. Aussi, pour exporter sous format PDF, il vous faut installer latex ou miktex sur votre machine.

Par ailleurs, il est possible d’adapter la sortie d’un document Word avec un modèle Word prédéfini grâce au paramètre “reference.doc”

## 3.2 R-markdown présentation

Il est également possible de faire créer des slides et donc des présentations sur R qu’on pourra exporter sous format beamer (PDF), html ou PowerPoint. La particularité ici c’est que pour passer d’une slide à l’autre, il faut mettre “##” séparé de haut en bas par une ligne vide.

Au-delà de tout ceci, il est important de rappeler qu’on peut écrire du code latex dans la partie réservée au texte sur R-markdown. Si vous êtes déjà fasciné par R-markdown, attendez de voir Quarto.

# 4 Exposé 4 : Quarto

En quelques mots, Quarto c’est R-markdown en mieux ! Avec Quarto, on a la possibilité de créer des documents interactifs et reproductibles avec une interface plus présentable et semblable à celle de Word (possibilité de cliquer). Pour créer un nouveau document Quarto, on va juste dans le menu file ensuite sur New file puis Quarto document. Comme avec R-markdown, on pourra choisir le format de la sortie (Word, PDF). Il faut noter que Quarto nous donne également la possibilité de faire des présentations animées comme sur PowerPoint. Une autre possibilité intéressante de Quarto c’est qu’il nous permet de créer des livres où l’on peut ranger tous ces travaux bien expliqués afin de les déployer sur nos comptes GitHub ou encore les joindre à nos CV.

# 5 Exposé 5 : R vers Excel

Le saviez-vous ? Depuis R, on peut ouvrir un classeur Excel (ou en créer), ouvrir ou créer une nouvelle feuille et y mettre des informations. Eh oui c’est bien possible. Pour ceux qui travaillent encore majoritairement sur office, il est possible après avoir fait des analyses sur R de les exporter sous forme de tableau Excel et d’y faire les manipulations que l’on veut (graphiques). On pourrait par exemple exporter un tableau fait grâce à gtsummary et cela grâce au package “r2excel” développé par kassambara mais pas encore disponible sur CRAN. Pour l’installer il faut donc utiliser le package “devtools” et installer java pour l’utiliser.

# 6 Exposé 6 : Text mining

Dans les enquêtes statistiques, il arrive qu’on insère au niveau du questionnaire, des questions ouvertes pour capter plus d’informations. Mais comment gérer les réponses à ces questions ouvertes ?

* Les packages tidytext, dplyr et tm nous permettent de nettoyer notre base de données textuelle : supprimer les mots indésirables tels que les caractères spéciaux (en créant un dictionnaire de mots à supprimer), les espaces, convertir en minuscule
* Le package wordcloud, lui permet de faire le nuage des mots avec les mots les plus fréquents.

Les mots pris isolement peuvent nous induire en erreur quant à la vrai pensée des individus d’où l’importance d’étudier les associations. Ces associations nous permettrons de créer de nouvelles variables dichotomiques qui prendraient la valeur “1” si l’individu en à parler et 0 sinon : voilà le but ultime du texte mining. En d’autres termes, le text mining permet de recoder les modalités “autres à préciser”.

# 7 Exposé 7 : Calcul parallèle

Il arrive qu’on soit amené à travailler sur d’énormes bases de données que notre machine ne peut même pas stocker. Que faire alors ? On fait du calcul parallèle tout simplement ! Le principe c’est:

* Importer cette base volumineuse grâce aux packages “fs” et “fst” ;
* Diviser la base en plusieurs sous bases et y faire nos analyses (map) : le microprocesseur est divisé en mini processeurs virtuels (cœurs) auxquels sont affectés les sous-bases et les tâches à y effectuer ;
* Agréger les résultats en un résultat global (reduce). C’est le principe du “map reduce” et aussi le fonctionnement du Big data (on parle de Big data lorsque le nombre d’observations excèdent le nombre de lignes d’Excel) et il permet de minimiser le temps de calcul. Sur R les packages qui permettent de faire du calcul parallèle sont “parallel” et “doParallel”. Par contre il ne sert à rien d’utiliser le calcul parallèle sur une petite base de données vu qu’il n’a pas été conçu pour cela.

# 8 Exposé 8 : Résolution d’équations différentielles

Dans le cadre de nos projets professionnels, nous sommes souvent amenés à résoudre des systèmes d’équations différentielles qui pourraient prendre des semaines des mois voire des années si on voulait les résoudre manuellement. Ce qui a encouragé l’implémentation de méthodes de résolutions numériques sur des logiciels mathématiques spécialisés comme Matlab ou Scilab mais aussi sur des logiciels et langages statistiques comme stata ou R. On distingue en général deux grande familles de méthodes

## 8.1 Méthodes directes

Sur R le package qui permet la résolution par des méthodes directes est “rootsolve” qui fournit des fonctions comme “multiroot” qui permet une résolution suivant la méthode de Newton. On peut également utiliser Le package nleqslv qui permet de résoudre numériquement des systèmes d’équations non linéaires multivariées en utilisant les méthodes de Newton et de Broyden. Le package pracma joue le même rôle.

## 8.2 Méthodes indirectes

Les méthodes indirectes (ou du statisticien) : Ici le principe c’est de transformer le système d’équation en fonction objective qu’il faut minimiser ou maximiser grâce aux fonctions optim () dans le package stats ou optimx () dans le package optimx

# 9 Exposé 9 : Python dans R : Reticulate

En tant que Data scientists, nous avons idéalement une bonne compréhension de R et de Python. Et les deux ont des forces uniques que nous pouvons utiliser à notre avantage alors pourquoi choisir ? Solution : Reticulate (= reticulé ou reticulée qui veux dire “En forme de réseau”) Reticulate est un package R qui permet de mettre en relation Python et R en fournissant un ensemble assez complet d’outils pour assurer l’interopérabilité entre Python et R. Il permet entre autres de :

* Appeler de Python à partir de R de diverses manières, R Markdown, référencement de scripts Python, importation de modules Python et utilisation interactive de Python dans une session R ;
* Convertir les objets R en Python et vice versa (par exemple, entre des data frame R et Pandas, ou entre des matrices R et des tableaux NumPy) ;

# 10 Exposé 10 : Cartographie avec R

La visualisation des données en statistique ne s’arrête pas seulement aux graphiques mais va jusqu’à la représentation des cartes. Et quand on parle de cartographie, on distingue 2 types de données :

* Les vecteurs qui regroupent les points, lignes et polygones permettant de représenter respectivement (sans exhaustivité) des localités, des routes (ou fleuves) et des contours administratifs (pays, régions…) ;
* Les rasters ou images satellites qui sont des matrices divisées en petits carrés (appelés résolution).

Ces données cartographiques ainsi que les indicateurs qu’on souhaiterait représenter sont stockés dans 4 fichiers de même nom indispensable pour faire de la cartographie sur R :

* Le fichier .prg qui renseigne sur la position (des éléments) sur la terre ;
* Le fichier .dbf qui est la base de données regroupant les indicateurs à représenter ;
* Le fichier .shp qui contient les vecteurs (lignes, points, polygones) ;
* Le fichier .shx, le pointeur qui donnera l’ordre de représentation. Il dira par exemple qu’elle ligne suit l’autre quand on veut dessiner un polygone. Par ailleurs, il faut noter que les images satellites ont pour extension .tif.

Dans R pour faire de la cartographie, on utilise plusieurs packages à savoir : cartography”, (réaliser des cartes) “classInt”( discrétisation de variables quantitatives )“ggspatial”, (syntaxe complémentaires à la ggplot) “GISTools”, (outils pour faire de la carto )“leaflet”, (interractivité avec JavaScript) “maptools”, (manipulation de données )“spatial”, “OpenStreetMap”, ( OSM) “osrm”, ( openstreetmap avec R ) “popcircle”, ( représentation style bubble plot )“raster”, ( manipulation de données raster) “RColorBrewer”, (palette de couleurs pour carto) “rgdal”, (import de données spatiales) “rgeos”, (manipulation de données spatiales) “sf”, (nouvelle classe d’objets spatials ) “sp”, (ancienne classe d’objets spatiales ) “tidyverse” etc. Et les fichiers de cartographies ont deux formats “sp” et “sf” avec sf qui se présente comme un dataframe ce qui rend sa manipulation un peu plus aisée car on peut lui appliquer les fonctions des packages dplyr ou tidyverse…

En poursuivant notre voyage dans le fabuleux monde de la cartographie sur R, il est très important de rappeler qu’avant toute représentation, il faut définir le système de référencement de coordonnées géographique le “CRS” dans lequel on se trouve. Le CRS ou “Coordinate Reference System” est un système de coordonnées utilisé pour définir et représenter les positions géographiques sur une surface plane ou courbe, telle que la Terre. Ainsi en fonction des CRS la forme de la terre peut varier (ronde, elliptique, plane) d’où l’importance de gérer ce paramètre surtout lorsque l’on voudrait superposer deux ou plusieurs cartes.

Par ailleurs, on peut transformer de simples data.frame contenant des variables de coordonnées géographiques en l’un des format sf ou sp grâce aux packages du même nom et exporter le résultat. Comme nous l’avions dit, il faut impérativement 4 types de fichiers pour faire la cartographie et pas d’inquiétude R les génèrera automatiquement lorsque vous exportez le résultat sous format .shp (il créera automatiquement les autres)

Eh bien maintenant il ne reste plus qu’à exploiter ces packages cités plus haut pour créer tout type de cartes ; y ajouter des flèches de flux ; calculer la distance entre deux points et plus encore…

# 11 Exposé 11 : R shiny

Shiny est un package R développé par Studio qui permet de créer des applications web interactives. C’est une plateforme qui facilite la création d’interfaces utilisateur dynamiques pour analyser des données, afficher des graphiques, exécuter des modèles statistiques, et bien plus encore, le tout directement à partir de R. Sur une interface Shiny, on peut retrouver par exemple le fluidPage qui délimite l’environnement, c’est comme le carton ou la maison dans laquelle sont ranger les éléments de l’interface comme le titlePanel qui affiche le titre de la page shiny ; le textInput une zone de saisie;… Parler de shiny revient à parler du UI (interface de l’utilisateur) et du SERVER. Pour créer une belle interface shiny, il nécessite plusieurs fonctions. Au niveau du UI il y a par exemple selectInput (), sliderInput () verbatimTextOuput, textInput, plotOuput, tableOutput ; et dans le serveur renderText (), renderPlot(), renderTable() , observeEvent(), reactive() , renderUI. Par ailleurs, il faut noter qu’il est impossible d’utiliser le server sans avoir programmer l’UI et que c’est dans le server que seront spécifié les propriétés des éléments de l’UI et les tâches à effectuer lorsqu’un utilisateur interagira avec ces derniers.

Après avoir créé l’UI et le Serveur, on peut lancer l’application en utilisant la fonction shinyApp qui combine les définitions de l’interface utilisateur et du code serveur pour créer l’application Shiny. Ensuite, pour exécuter l’application, on utilise la fonction runApp qui lance un serveur local et ouvre l’application dans le navigateur par défaut.