RESUME DES EXPOSES DE R

Groupe 4: Alex - Durel- Moustapha

2023-04-07

Table of contents

1 Let's Go **5**

2 Introduction **6**

3 Janitor **7**
3.1 Description du package 7
3.2 Quelques fonctions de nettoyage 8
3.3 Quelques fonction de tableaux 8
3.4 Quelques limites du package 9

4 GtSummary **10**
4.1 Présentation du package 10
4.2 Différences avec d'autres packages 10
4.3 Quelques fonctions essentielles 11

5 Rmarkdown **12**
5.1 Introduction 12
5.2 Préliminaire 12
5.3 Création d'un document 12
5.4 Présentation du Markdown 13

6 Quarto **15**
6.1 INTRODUCTION 15
6.2 Insertion des titres 15
6.3 Titre 2 15
6.3.1 Titre 3 15
6.4 Style du texte 16
6.5 Mise en couleur 16
6.6 Bordure de texte 16
6.7 Listes 17
6.8 Insérer une image 17
6.9 Insertion des tableaux 19
6.10 Insertion des diagrammes 19
6.11 Insérer un graphique 24

6.12 Insérer un saut de page 24
6.13 Insérer un lien 25
6.14 Insérer une équation 25
6.15 Insérer des symboles ou emoji 25
6.16 Insérer des liens hypertextes 25
6.17 Insérer des notes de bas de pages 26
6.18 Insérer des notes 26
6.19 Insérer des références bibliothèque 26

7 R vers Excel **28**
7.1 Introduction 28
7.2 Présentation du package R2Excel 28
7.3 Principales fonctions 28
7.4 Conclusion 29

8 Text Mining sur R **30**
8.1 Introduction 30
8.2 Les étapes du text mining 30
8.3 Quelques fonctions à connaitre 31

9 Calcul Parallèle **32**
9.1 Introduction 32
9.2 Principes généraux 32
9.3 Etapes du calcul en parallèle 33
9.4 Calcul en parallèle avec R 33
9.5 Quelques fonctions 34

10 Système d'équation sur R **35**
10.1 Présentation des packages et fonctions 35
10.1.1 Package rootSolve 35
10.1.2 Fonction multiroot() 35
10.2 Le package nleqslv 35
10.2.1 Fonction nleqslv() 36
10.3 Le package pracma 36
10.4 Les méthodes indirectes 36

11 Python dans R: le package Reticulate **38**
11.1 Introduction 38
11.2 Comment marche Reticulate ? 38

12 Cartographie avec R **40**
12.1 Introduction 40

12.2 Principaux packages 40
12.3 Les types de données 41

13 Rshiny **43**
13.1 Introduction 43
13.2 Présentation de Shiny 43
13.3 Les différentes parties d'une application Shiny . . 44

14 Conclusion **45**

References **48**

1 Let's Go

2 Introduction

Dans le cadre de notre module de R de 20h, cours qui nous a été dispensé par M. Aboubacar HEMA, Research Analyst à IFPRI, nous avons été à faire des exposés en vue d'enrichir le cours. Ces exposés portent sur les thèmes suivants:

- Package Janitor
- Package GtSummary
- Rmarkdown
- RQuarto
- R vers Excel
- Text Mining avec R
- Système d'équation
- Calcul parallèle
- Python sur R (package Reticulate)
- Cartographie avec R
- RShiny

Dans notre présent livre, nous vous présenterons brièvement les différents points abordés lors de ces présentations.

3 Janitor

3.1 Description du package

Le package Janitor est un ensemble de fonctions qui permettent d'améliorer la manipulation et le formatage des données dans les data.frames. Certaines des principales fonctions du package Janitor sont:

1. Formater les noms de colonnes : Il permet de formater correctement les noms des colonnes dans un data.frame pour les rendre plus lisibles et cohérents.
2. Comptages rapides des combinaisons de variables : Il fournit des fonctionnalités pour effectuer rapidement des comptages des combinaisons de variables dans un data.frame, ce qui permet d'obtenir les fréquences de chaque combinaison.
3. Recherche des enregistrements en double : Il permet de détecter les enregistrements en double dans un data.frame, c'est-à-dire les lignes qui ont les mêmes valeurs pour toutes les variables.
4. Formatage des résultats de tabulation : Le package Janitor fournit également des fonctions pour formater joliment les résultats de la tabulation, ce qui facilite leur lecture et leur interprétation.

En résumé, le package Janitor facilite le nettoyage, la mise en forme et l'exploration des données dans R. Il offre des fonctionnalités similaires à celles de SPSS et Microsoft Excel, et il est conçu pour être convivial et facile à utiliser, en suivant les principes du "pipe" (%>%) pour faciliter les opérations en chaîne.

Il joue essentiellment deux fonctions: **le nettoyage des données** et **la visualisation des données en tableaux.**

3.2 Quelques fonctions de nettoyage

Plusieurs fonctions permettent de rendre propre la base. En voici quelques unes:

- *clean_names()*: elle permet de nettoyer les noms des variables. En ce sens, quand il y a des espaces entre les noms de colonnes il les remplace par des _, quand deux colonnes ont même nom il remplace le deuxème par même nom_2. Il gère aussi la casse en mettant tous les noms sous le même format.
- *Compare_df_cols()* : compare les colonnes sur quelques bases, ls classes des éléments des colonnes
- *Make_clean_names()* : permet de nettoyer les noms de variables par des séries de transformation
- *Remove_constant()* : Permet de supprimer une colonne qui a que des valeurs constantes
- *Get_dupes()* : Permet de repérer des lignes qui se répètent avec leur identifiant

3.3 Quelques fonction de tableaux

- *adorn_totals()*; ajoute une colonne de totaux à un dataframe.
- *adorn_percentage()*: transforme un dataframe d'effectifs en pourcentage

3.4 Quelques limites du package

Le package `Janitor` n'est pas un package tout fait pour les tableaux ou un package qui peut tout seul réaliser ses taches. Il a besoin d'autres fonctions pour ses taches. Aussi, il n'offre pas pour ses tableaux, la possibilités de combiner plusieurs variables.

4 GtSummary

4.1 Présentation du package

Le package gtsummary est un package spécialisé dans la confection de tableaux bien stylés. Le plus souvent, lorsqu'on sort un tableau sur Rmarkdown, son format n'est pas vraiment joli. Pour répondre à ce besoin, le package ci a été développé. Il permet de réaliser des tableaux avec la combinaison de plusieurs variables, des tableaux de résumés statistiques ou encore la regression. Comme tout package, avant d'être utilisé, gtsummary a besoin d'être installé avec la commande usuelle que l'on connait (install.package("gtsummary"). Il est également important de télécharger la version de développement depuis github (remotes::install_github("ddsjoberg/gtsummary")).

4.2 Différences avec d'autres packages

Il y a 3 packages essentiels desquels diffère `gtsummary`.

- `Janitor` que l'on a déjà vu et qui est spécialisé dans ce que l'on peut appeler le preprocessing. Il sert à nettoyer notre base de données et certaines variables en vue de les rendre prête à être utilisées.
- Stargazer est principalement utilisé pour créer des tableaux de regressions et de statistiques descriptives à partir des modèles alors que gtsummary se concentre sur la réalisation de résumés statistiques complets incluant des mesures descriptives, des tests statistiques et des graphiques.

4.3 Quelques fonctions essentielles

- *tbl_summary()*: elle permet de faire une table de résumé pour toute la base. Cela consiste à sortir pour chaue variable les résumés en terme de fréquence d'apparition de chaque modalité (pour des variables catégorielles) ou encore médiane, moyenne et autres pour des variables continues. Par ailleurs, suivant les types de variables dont on veut sortir le résumé, des fonctions sont adaptées. Par exemple respectivement `binary %>% tbl_summary()` , `multicoto %>% tbl_summary()` et `continu %>% tbl_summary()` pour des variables **binaires**, **multichotomiques** et **continues.**
- *tbl_cross()*: pour des tableaux croisés
- *tbl_regress()*: pour sortir les tableaux issus d'une regression.
- *tbl_stack():* permet de coller deux (ou plus) tableaux l'un au-dessus de l'autre
- *tbl_merge()*: les placera côte-à-côte, en s'assurant qu'une même variable sera bien affichée sur la même ligne.
- *tbl_spit()*: permet de découper un tableau qui est trop long en précisant les variables suivant lesquelles on découpe.

Il est également possible de spécifier le format d'affichage du tableau en utilisant des thèmes avec la fonction *theme_gtsummary*()*. Nous avons par exemple la fonction *theme_gtsummary_journal()* qui adopte le format standard de certaines grandes revues scientifiques. Nous pouvons aussi nous même définir un thème à appliquer à nos tableaux sous forme de fonction. La possibilité nous est donné de modifier les styles d'affichages (en gras, italique, textes présents dans les tableaux.

Voilà un peu ce qui met fin à ce chapitre. Il faudra retenir que gtsummary est un package de sortie de tableaux.

5 Rmarkdown

5.1 Introduction

Rmarkdown est un format de fichier qui permet de créer des documents avec R. En écrivant un rapport dont nous sortons les tableaux, graphiques ou autres sur R, nous n'avons plus besoin d'aller sur Word après pour coller ses graphiques et continuer nos analyses. La perte de temps issu de ce va-et-vient entre Word et R est désormais résolu grâce à Rmarkdown qui nous offre la possibilité d'écrire directement notre rapport ou document et le sortir après sous format HTML, Word ou PDF. Cependant il nous permet également de créer des présentations sous format PDF (Beamer) ou encore PowerPoint.

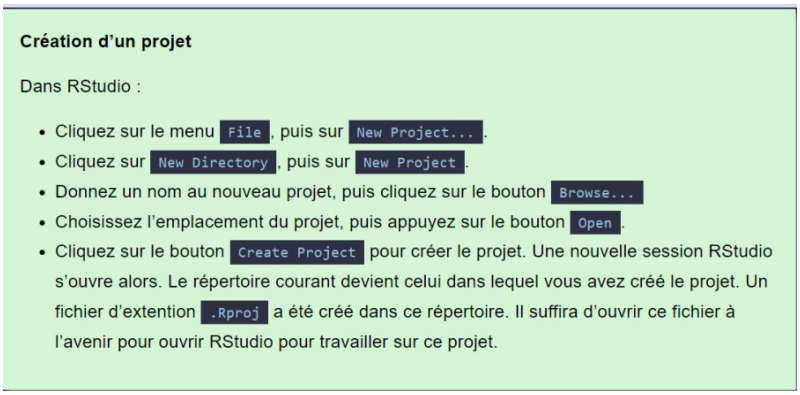
5.2 Préliminaire

Avant toute chose il faudra installer le package `Rmarkdown`. Cependant, on peut utiliser un créer un document Rmarkdown même sans avoir installé ce package.

5.3 Création d'un document

On peut soit créer un projet ou encore un fichier. S'agissant d'un projet, voici les étapes:

Quant au fichier Markdown il faudra cliquer l'onglet `File` ensuite `New file` et enfin `Rmarkdown`. Après selon qu"on veuille un document ou une présentation, on choisit `Document` ou `Presentation`. Pour finir, il faudra préciser le format de sortie, le titre du document et le(s) auteur(s).



Création d'un projet

Dans RStudio :

- Cliquez sur le menu `File`, puis sur `New Project...`.
- Cliquez sur `New Directory`, puis sur `New Project`.
- Donnez un nom au nouveau projet, puis cliquez sur le bouton `Browse...`
- Choisissez l'emplacement du projet, puis appuyez sur le bouton `Open`.
- Cliquez sur le bouton `Create Project` pour créer le projet. Une nouvelle session RStudio s'ouvre alors. Le répertoire courant devient celui dans lequel vous avez créé le projet. Un fichier d'extention `.Rproj` a été créé dans ce répertoire. Il suffira d'ouvrir ce fichier à l'avenir pour ouvrir RStudio pour travailler sur ce projet.

5.4 Présentation du Markdown

Le document Markdown comprend 3 parties que sont l'en-tête (YAML), le texte et les blocs de codes tous deux formant le corps du document. Dans l'en-tete sont précisés des formats additionnels pour le document (Table de matières, références s'il y en a, des packages Latex à inclure si possible).

Concernant le corps du devoir, quelques astcuces sont en mentionnées:

1. Pour insérer un nouveau titre, nous utilisons "#", entre l'élément que nous voulons en titre, par exemple: "#" Titre 1 "##" Titre 2 "###" Titre 3
2. Pour forcer le retour à la ligne, il faut mettre "deux espaces vides à la fin d'une ligne;
3. Pour écrire en gras il faut mettre deux étoiles (ou deus tirets de 8) entre le mot souhaité;
4. Pour mettre un texte en italique, on le met entre deux tirets de huit (2 étoiles);
5. Pour mettre en gras et en italique,on le met enre 3 étoiles (ou 3 tirets de 8);
6. Pour barer un texte, précédé le de 2 tildes et refermez la chaine avec 2 tildes;
7. Pour mettre un mot en exposant,encadez le du symbole "^";

8. Pour l'obtention d'un tiret long (respectuvement moyen) ,alignez 3 tirets de 6 successifs (respectivement 2 tirets de 6 successifs)

Il est aussi possible d'insérer des images, tableaux et autres. Cependant l'usage de tout cela nécessite la connaissance des codes de chaque élément à mettre. Pour simplifier alors tout cela et rendre la création de document sur R facile à tous et surtout ceux qui n'aiment pas coder, Quarto a été developpé avec une interface semblable à celle de Word. C'est ce que nous verrons au chapitre suivant.

6 Quarto

6.1 INTRODUCTION

Notre objectif dans cette présentation est d'explorer les différentes façons de présenter un document sur Quarto. Un bon environnement qui est très pratique et simple à utiliser contrairement à Rmarkdown qui nous fait écrire des codes et perdre du temps. La chose la plus intéressante sur ce environnement est qu'il offre une interface visuel et source (pour les codes). Nous allons ainsi voir comment produire un bon document avec Quarto, en d'autres termes que doit-on savoir pour produire un bon document Quarto ? La répone dans la suite

6.2 Insertion des titres

Pour insérer un titre dans Quarto il faut juste cliquer sur l'onglet `Normal` et on choisit le niveau de titre que l'on souhaite. On a Header 1 à Header 6 correspondant à Titre de niveau 1 à Titre de niveau 6.

6.3 Titre 2

6.3.1 Titre 3

6.3.1.1 Titre 4

6.3.1.1.1 Titre 5

6.3.1.1.1.1 Titre 6

Par défaut le texte est sous le format `Normal`

6.4 Style du texte

On peut préciser le style du texte que l'on est entrain d'écrire. Par exemple, si nous voulons que cela soit en gras, en italique, en majuscule, minuscule, sous forme d'indice ou encore d'exposant. Pour ce faire il suffit d'aller dans l'onglet `Format` de notre barre des taches.

Pour les textes en majuscule ou minuscule, barré il faut aller à `Format Text`.

6.5 Mise en couleur

Pour les mises en couleur, il faudra une maitrîse du langage CSS. Par exemple supposons que nous voulons metrre le texte suivant: `Je suis content` en bleu, je sélectionne d'abord mon texte , ensuite je vais dans `Format Div` et dans l'attribut `CSS Style` je précise mon code pour la couleur. Cependant il peut arriver que je veuilles que seul mon texte `content` soit en bleu. Là je vais la même chose mais au lieu d'insérer une div, j'insère un span.

Comme illustration de ce que je dis :

Je suis content

Je suis content

6.6 Bordure de texte

Je peux aussi vouloir encadre mon texte. Il faut dans ce cas aussi, faire appel à des compétences en CSS. Et procéder comme au niveau de la mise en couleur selon que veut que ce soit tout le texte ou juste une pârtie.

Je suis content

Je suis content

Pour la position du texte (centrée, justifié, à droite ou à gauche la même chose est utilisée avec du code CSS

Centré

Aligné à gauche

Aligné à droite

6.7 Listes

Pour créer une liste, on peut la faire soit de façon ordonnée ou non. Pour ce faire, on part dans l'onglet `Format  Bullets & Numbering` et on choisit le type selon qu'on veuille oronnée ou non ordonnée.

Par exemple

- Alex
- Durel
- Mohamed

Liste non ordonnée

1. Alex
2. Durel
3. Mohamed

Liste ordonnée

6.8 Insérer une image

Pour insérer une image, nous allons dans le menu `Insérer` et on choisit `Image/Figure`. On ala possibilité de choisir sa taille, sa largeur et aussi son titre.

Maintenant il peut arriver qu'on veuille insérer deux image l'une à coté de l'autre et non l'une après l'autre. Pour cela, nous devons inérer d'abord une div et préciser dans son attribut other que notre div contient 2 colonnes par `layout-ncol= 2`. En voici un exemple

Figure 6.1: Une image

Supposons qu'on veuille mettre 3 images dont deux en haut et une en bas. Il suffit juste de préciser dans Other ceci: `layout="[[1,1], [1]]"`

Supposons je veux mettre 4 images 2 en lignes et 2 en colonnes.

Maintenant si on veut deux images dont l'une est grand et l'autre petite en bas. On précise ceci dans Other `layout="[25,-2,10]" layout-valign="bottom"`

6.9 Insertion des tableaux

Pour insérer un tableau il suffit d'aller dans le menu `Table` et de cliquer `Insert Table`. Ensuite préciser les dimensions et le titre de ce tableau.

Table 6.1: Mon premier tableau

Col1	Col2	Col3

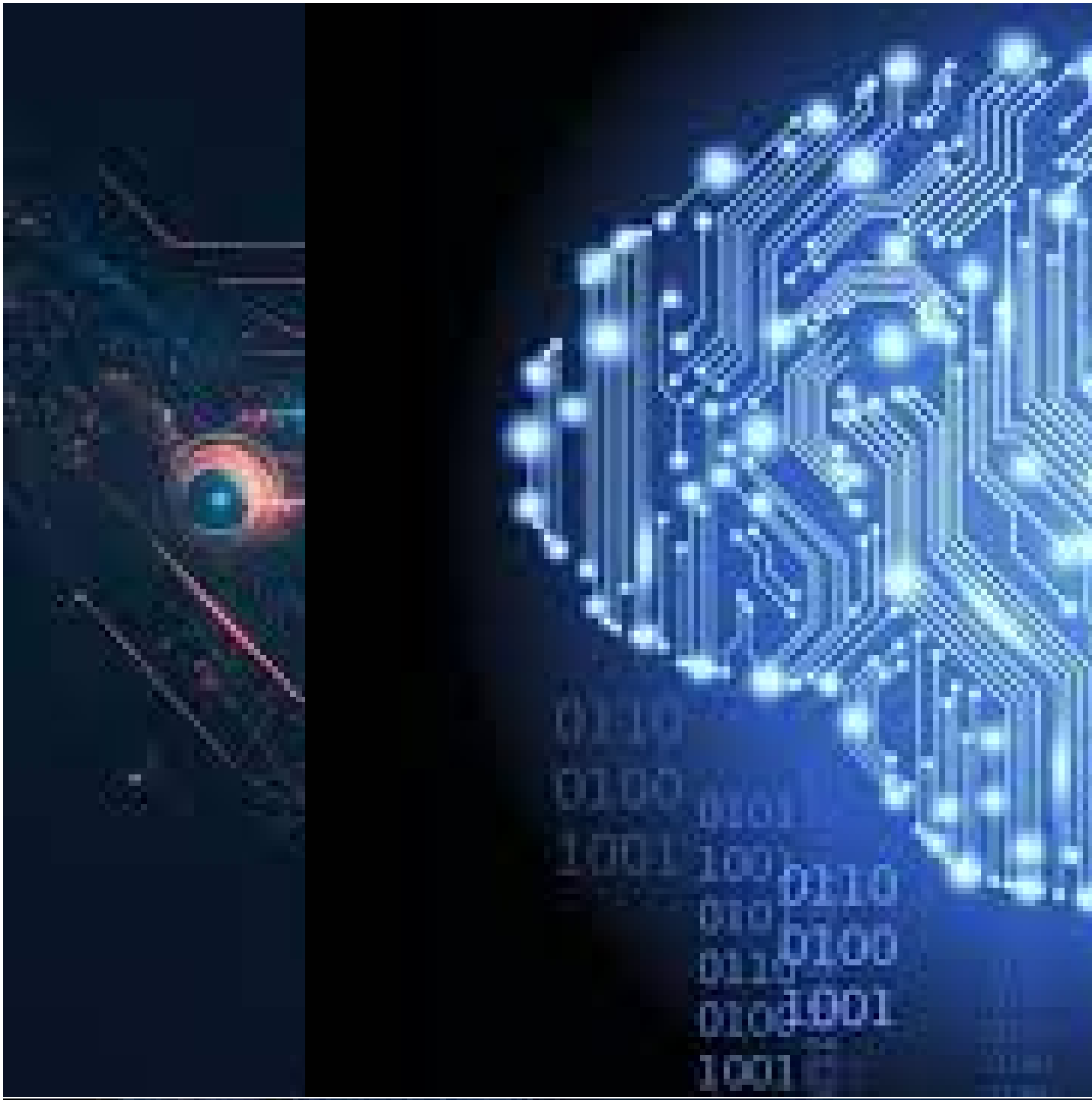
On peut comme les imges mettre aussi plusieurs tableau selon ce qu'on veut. Voici ici deux tableau plaé sur la même ligne.

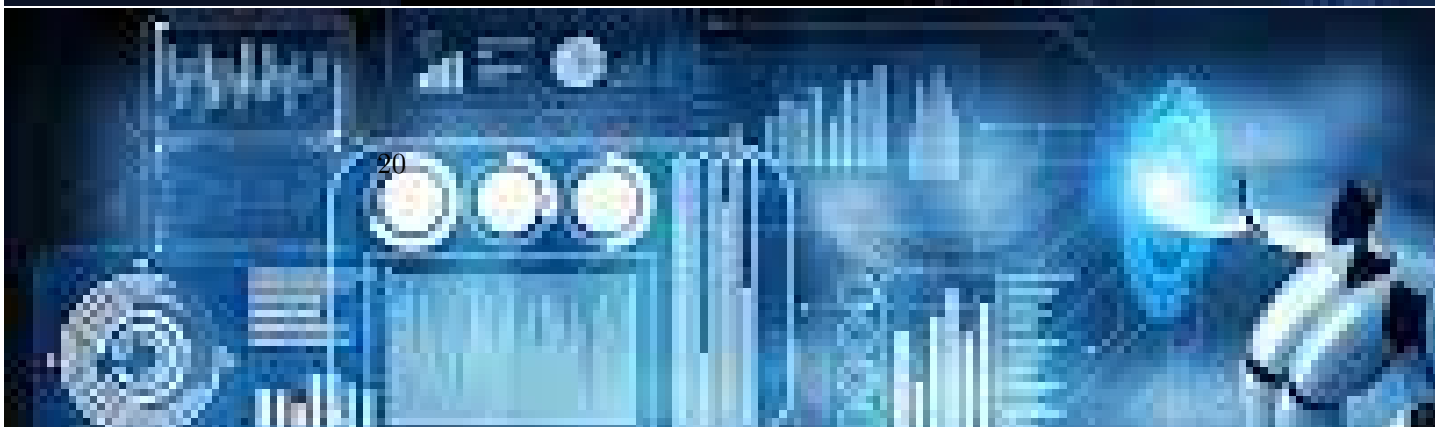
Supposons mainteant que l'on veuille commenter le tableau mais que le commentaire se trouve à gauche du tableau. On place alors dans notre précédente div un texte à la place du deuxième tableau

6.10 Insertion des diagrammes

On peut être amené dans notre présentation à insérer un diagramme. Pour cela, nous devons avoir des compétences en Graphviz ou encore Mermaid.

Un exemple ici


0110
0100
1001
0110
0100
1001




0110
0100
1001
AI

Table 6.2: Tableau 1

Col1	Col2	Col3

Table 6.3: Tableau 2

Col1	Col2	Col3

Le score final du match a été 2-1 en faveur des ISEP3/AS3

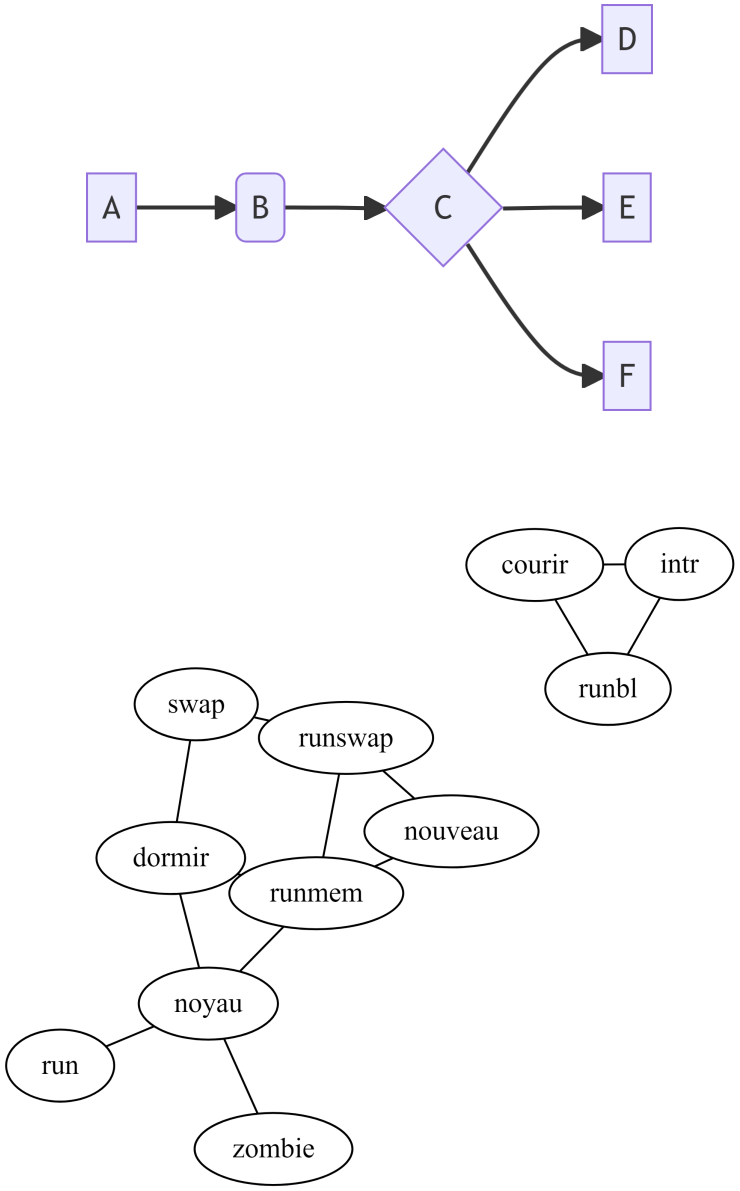
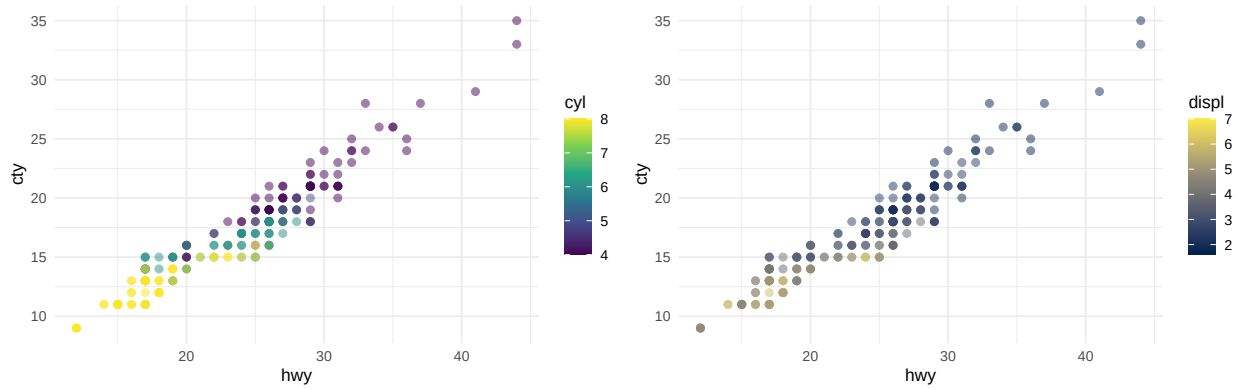


Figure 6.2: Ceci est un diagramme Graphviz

On peut aussi insérer des formes en utilisant du code CSS.

6.11 Insérer un graphique

Nous pouvons travailler directement avec notre base de données dans Quarto et essayer de sortir des graphiques à partir des variables de la base. Pour cela nous aurons besoin d'insérer un chunk R . Voici un exemple:



(a) Couleur par nombre de cylindres

(b) Couleur parn déplacement d'engin

Figure 6.3: Quelques graphes

6.12 Insérer un saut de page

Ce dernier ne peut focntionner qu'avec les documents Word et PDF. Pour ce fairenil faut insérer un shortcode avec `Insert -shotcode` et préciser `page break`

6.13 Insérer un lien

Pour ce faire, il faudra aller dans `Insérer - Link`. Ensuite mettre le lien et préciser le texte à afficher et sur lequel cliquer pour avoir la page. Voici un exemple

Dashboard Jupyter

6.14 Insérer une équation

Pour insérer une équation dans Quarto, il faut maitriser les codes LateX. Pour ce faire aller dans `Insert - LaTeX Math`. Ensuite suivant qu'on veut mettre ça sur la ligne de notre texte ou encore en bloc seul, il faut choisir InlineMath ou Display Math. $E = mC^2$

$$E = mC^2 \tag{6.1}$$

6.15 Insérer des symboles ou emoji

On peut insérer les symboles dans Quarto en allant à `Insert - Special Character - Insert Unicode` ou encore `Insert - special character - Insert Emoji`

¢

6.16 Insérer des liens hypertextes

On peut insérer des liens hypertextes vers des équations ou encore vers un tableau, une figure ou encore des titres. Pour ce faire il faudra d'abord préciser dans attributs l'ID du titre, texte ou figure. Ensuite vous allez dans `Insert - Cross reference`.

Aller à Equation 6.1

Allez à Figure 6.3

Allez à Section 6.1

6.17 Insérer des notes de bas de pages

Il suffit d'insérer des notes de bas de page juste après le mot concerné. Pour cela cliquez `Insert - Footnote` . Une sorte de nouvelle page apparait en bas avec le numéro du bas de page que vous insérez.

Par exemple. Je m'appelle LABOU [1]Komla Alex.

6.18 Insérer des notes

Si nous voulons mettre en exergue quelque chose dans un document, nous pouvons le faire en insérant un note (callout) en précisant le type que nous voulons (Important, note…).

> **!** Important
>
> Ce texte est très important

> **⚠** Attention
>
> Revoyez votre comportement.

6.19 Insérer des références bibliothèque

Pour insérer une citation tirée d'un document pendant que nous écrivons, nous devons connaitre le DOI de la source. Le DOI est un identifiant que chaque article en ligne a et c'est propre à cet article. Autrement, deux artciles ne peuvent pas avoir le mm DOI.

Pour ce faire une fois que nous avons écrit notre citation, pour la sourcer nous insérons le DOI du fichier dans `Insert - Citation - DOI` et nous ajoutons le DOI du fichier. Automatiquement il insère ce fichier dans notre référence bibligraphique en bas. Ainsi il est important d'insérer un titre référence après le document que nous écrivons.

[1]Nom d'une tribu au Sud du Togo

Par exmple

L'Afrique est un contient très sous développé (Lasida, Minkieba Lompo, and Dubois 2009).

Voici ce qui met fin à notre document. Nous espérons que vous l'avez apprécié.

7 R vers Excel

7.1 Introduction

De nombres fois, nous avons été confronté au problème d'exportation de tableaux réalisés sur R vers Excel. Il fallait copier nous même et ensuite coller et généralement la forme du tableau n'est pas maintenue. Pour pallier ce problème, le package R vers Excel `(R2Excel)` a été développée.

7.2 Présentation du package R2Excel

Le package R2Excel est une bibliothèque R qui permet d'exporter des données et des résultats d'analyses depuis R vers Microsoft Excel. Il fournit des fonctions pratiques pour créer des feuilles de calcul Excel, y écrire des données, formater les cellules, ajouter des graphiques et d'autres éléments visuels.

Son principal objectif est de faciliter le transfert de données et de résultats d'analyses entre R et Excel, en offrant une solution pratique pour les utilisateurs qui souhaitent exploiter les fonctionnalités d'Excel pour présenter, partager ou analyser leurs données.

7.3 Principales fonctions

Voici quelques fonctions utiles et servant à faciliter le passage de R vers Excel

1. `createWorkbook()` : Cette fonction permet de créer un nouveau classeur Excel. Elle renvoie un objet workbook qui représente le classeur.

2. `createSheet()` : Cette fonction permet de créer une nouvelle feuille de calcul Excel à l'intérieur d'un classeur. Elle renvoie un objet sheet qui représente la feuille.

3. `writeData()` : Cette fonction permet d'écrire des données dans une feuille de calcul Excel. Vous pouvez spécifier la plage de cellules où écrire les données et formater les cellules selon vos besoins.

4. `addChart()` : Cette fonction permet d'ajouter un graphique à une feuille de calcul Excel. Vous pouvez spécifier les données à utiliser, le type de graphique (barres, lignes, etc.) et personnaliser les options de présentation.

5. `setCellStyle()` : Cette fonction permet de définir le style des cellules dans une feuille de calcul Excel. Vous pouvez spécifier la police, la couleur de fond, les bordures et d'autres attributs de mise en forme.

6. `saveWorkbook()` : Cette fonction permet de sauvegarder le classeur Excel sur le disque.

Ces fonctions, ainsi que d'autres fournies par le package R2Excel, permettent aux utilisateurs de manipuler facilement les données et les résultats d'analyses dans Excel, en utilisant les fonctionnalités avancées d'Excel pour créer des rapports, des tableaux de bord interactifs, des présentations visuelles, etc.

7.4 Conclusion

En résumé, le package R2Excel facilite l'exportation de données et de résultats d'analyses depuis R vers Excel, en offrant des fonctions pratiques pour créer, écrire et formater des feuilles de calcul Excel, ainsi que pour ajouter des graphiques et d'autres éléments visuels.

8 Text Mining sur R

8.1 Introduction

Etant statisticien, nous connaissons l'importance des enquêtes. Généralement dans les enquêtes, nous avons des questions ouvertes et des questions fermées. Le traitement des questions fermées est chose aisée, nous avons plusieurs méthodes. Par contre avec les questions ouvertes, cela devient un peu délicat. Toutes les techniques de statistiques univariées et bivariées ne servent pas. C'est là qu'intervient le text mining, qui nous permet d'analyser des données textuelles. Sur R, il nécessite quelques packages essentiels à savoir **tidytext, dplyr** et **tm.**

8.2 Les étapes du text mining

Le text mining suit quelques étapes à respecter:

- **Prétraitement du texte**: en vue de rendre nos données exploitables, il est important de les traiter d'abord. Cela passe par la tokenisation (découper chaque texte en mots) ou encore d'autres méthodes allant à la suppression des mots inutiles pour notre analyse.
- **Le cloud**: le nuage de mots, il permet de voir les mots les plus utilisés et les groupes de mots les plus utilisés suivant que l'on a formé des ngrammes (combinaisons de n mots).
- **Le réseau de mots**: permet de voir parmi les mots les plus utilisés, les combinaisons les plus utilisés.

- **Creation d'une variable**: le but du text mining est en quelque sorte cela. L'objectif est d'arriver à créer une variable à partir des mots ou combinaisons de mots les plus utilisés, qui prend 1 si l'individu a utilisé cela et 0 sinon.

8.3 Quelques fonctions à connaitre

Certaines fonctions sont à connaître pour faire du text mining.

- `tm_map()`: Cette fonction permet d'appliquer une transformation à chaque document d'un corpus textuel. Dans vos exemples, différentes transformations sont appliquées à `txte1`.
- `removePunctuation()`: Cette fonction permet de supprimer la ponctuation des documents textuels.
- `removeNumbers()`: Cette fonction permet de supprimer les chiffres des documents textuels.
- `removeWords()`: Cette fonction permet de supprimer des mots spécifiques des documents textuels. Dans votre exemple, les mots vides (stopwords) en anglais sont supprimés à l'aide de la fonction `stopwords("english")`.
- `stripWhitespace()`: Cette fonction permet de supprimer les espaces supplémentaires dans les documents textuels.
- `tm_reduce()`: Cette fonction permet de réduire le corpus en supprimant les termes rares ou très fréquents.

9 Calcul Parallèle

9.1 Introduction

De manière générale la plupart des traitements que nous faisons suivent le principe du calcul séquentiel ou une tâche doit être exécuté avant que les autres ne soient déclenchés et ainsi de suite. Face à des petites bases de données, le temps pour effectuer ces opérations nous semble court. Cependant, avec de grandes bases de données, il est facile de remarquer la lenteur de l'ordinateur à exécuter les tâches que nous lui demandons. Le calcul séquentiel est ainsi limité. Pour pallier ces problèmes et rendre le calcul moins lent, il est impératif d'utiliser le calcul en parallèle dont l'objectif est de découper la grande tâche en des sous tâches indépendantes et ensuite associer le résultat de chaque sous tâche pour sortir le résultat final.

9.2 Principes généraux

1. Briser un calcul informatique en blocs de calcul indépendants;

2. Exécuter simultanément (en parallèle) les blocs de calcul sur plusieurs unités de calcul;

3. Rassembler les résultats et les retourner.

Paralléliser un problème consiste à décomposer ce problème en plusieurs sous problèmes à résoudre simultanément à travers différentes ressources, pour ressortir la solution du problème initial, dans un délai optimal.Ainsi, le principe du calcul parallèle est d'effectuer simultanément une même tâche ou exécuter un même programme de manière parallèle. Cela est aussi possible à travers différentes machines connectées par un réseau où

chacun d'eux reçoit une tache à exécuter. Sur R, L'utilité du calcul en parallèle réside dans le faite qu'il permet d'effectuer plus rapidement et de manière asynchrone l'exécution de programme sur des bases de données volumineuses en exploitant simultanément plusieurs unités de calcul d'un ordinateur appelées cœurs

9.3 Etapes du calcul en parallèle

1. Démarrer m processus "travailleurs" (i.e.cœurs de calcul) et les initialiser;

2. Envoyer les fonctions et données nécessaires pour chaque tache aux travailleurs;

3. Séparer les taches en m opérations d'envergure similaire et les envoyer aux travailleurs;

4. Attendre que tous les travailleurs aient terminer leurs calculs et obtenir leurs résultats;

5. Rassembler les résultats des différents travailleurs;

6. Arrêter les processus travailleurs

9.4 Calcul en parallèle avec R

R dispose de packages spécialisés dans le calcul en parallèle. Il s'agit de:

- Package Parallel: inclus dans la distribution de base de R: Il se base sur l'utilisation de fonctions de la

famille des apply

- Package doParallel et Foreach:
- Le package rmr2 (MapReduce)
- Etc.

9.5 Quelques fonctions

Detectcores(): permet de détecter le nombre cœurs de la machine.

– Makecluster() :

– Stopcluster(): est utilisé pour arrêter et libérer les différents workers.

– La famille des fonctions Applay, adaptées au calcul parallèle sous R permet d'exécuter simultané-

ment les opérations sur les différents blocs.

- parApply() permet d'effectuer des calculs en parallèle sur une matrice ou un tableau

en utilisant un cluster de travailleurs

- parLapply() permet d'appliquer une fonction à chaque élément d'une liste en utilisant un

cluster de travailleurs pour exécuter les calculs en parallèle.

- parSapply() permet d'appliquer une fonction de manière parallèle à des éléments d'une

liste. Elle prend en argument le jeu de donné et la fonction et retourne un vecteur ou une

matrice.

– clusterEvalQ(): Elle est utilisée pour évaluer une expression sur tous les nœuds d'un cluster

parallèle. Elle est utile lorsque vous avez besoin d'exécuter une expression ou de charger des

bibliothèques spécifiques sur chaque nœud du cluster avant d'exécuter des tâches parallèles.

10 Système d'équation sur R

Dans la vie, la plupart des systèmes équations auquels nous faisons sont non linéaires. Dans le cas linéairee, les méthodes du pivot de Gauss et autres ont été développées et sont très efficaces. Ici, dans ce chapitre, nous allons aborder les méthodes dans le cas non lineaires.

10.1 Présentation des packages et fonctions

10.1.1 Package rootSolve

Le package "rootSolve" est une bibliothèque de fonctions disponible dans le logiciel R qui fournit des outils pour la résolution numérique de systèmes d'équations non linéaires.

10.1.2 Fonction multiroot()

Cette fonction est utilisée pour résoudre des systèmes d'équations non linéaires multivariées. Elle prend en entrée un vecteur de fonctions d'équation multivariée, ainsi qu'un vecteur initial de valeurs approchées pour les variables, et retourne une approximation des valeurs des variables qui satisfont le système d'équations. l'algorithme a été implémenté avec la méthode de Newton-Raphson.

10.2 Le package nleqslv

Le package "nleqslv" est une bibliothèque de fonctions disponible dans le logiciel R qui permet de résoudre des systèmes d'équations non linéaires multivariées. Il offre des

fonctionnalités avancées pour trouver les solutions numériques de ces systèmes en utilisant des méthodes itératives basées sur la méthode de Newton et de broyden(méthode par défaut).

10.2.1 Fonction nleqslv()

Cette fonction est le cœur du package. Elle permet de résoudre numériquement des systèmes d'équations non linéaires multivariées en utilisant les méthode de Newton et de Broyden . Elle prend en entrée une fonction d'équation multivariée, un vecteur initial de valeurs approchées pour les variables, et retourne une approximation des valeurs des variables qui satisfont le système d'équations.

10.3 Le package pracma

Le package "pracma" aussi propose des méthodes numériques pour résoudre des équations non linéaires. Par exemple, les fonctions fsolve() et broyden() en utilisant la méthode de Newton ou la méthode de Broyden, respectivement.

10.4 Les méthodes indirectes

On peut aussi résoudre un système d'équations non linéaires à l'aide d'un problème d'optimisation . Cela peut se faire on Transforme le système d'équations en un problème d'optimisation en définissant une fonction objectif à minimiser ou à maximiser. Cette fonction objectif est généralement construite en utilisant une mesure de l'écart entre les valeurs réelles des équations et les valeurs calculées à partir des variables inconnues. On Choisit ensuite une méthode d'optimisation : Il existe plusieurs méthodes d'optimisation disponibles pour résoudre le problème. Certaines des méthodes couramment utilisées incluent **la méthode de Newton**, **la méthode du gradient, la méthode de recherche linéaire, la méthode de Gauss-Newton, etc**. Sous R, propose pour cela les

fonctions comme `optim ()` dans le package stats , `optimx()`
dans le package `optimx`.

11 Python dans R: le package Reticulate

11.1 Introduction

Nous sommes tous au courant de l'éternel débat qu'il y a entre les R-users et les Python-users. Nous même souvent dans nos projets personnels, on sent la facilité de réaliser certaines taches dans R et d'autres sous Python nous faisant ainsi naviguer entre ces deux logiciels. Cependant ce va-et-vient nous fait perdre du temps. Pour ainsi répondre à ce besoin, Python a été intégré à R dans le logiciel Rstudio offrant la possibilité d'éxécuter des codes R et Python simultanément dans un même fichier de code grâce à un package. Ce package nommé **Reticulate** fera l'objet de notre chapitre.

11.2 Comment marche Reticulate ?

Bien que permettant de réconcilier les deux langages, Reticulate n'est pas le seul moyen d'utiliser Python et R. Il existe d'autre options comme **rPython** ou **PythonInR.** Pour pouvoir utiliser ce package il faut naturellement l'installer. Voic les 3 phases essentielles à l'usage de Reticulate

1. Phase 1 : installer le package Reticulate sur R
2. Phase 2 : installer Python : Pour utiliser python dans R il faut s'assurer d'avoir Python installé dans sa machine (versuion >= 2.7)
3. Phase 3 : Configurer l'environnement Python (ie spécifier la version

   de Python à utiliser)

Une fois ces étapes définies, il suffira juste d'insérer notre chunk Python sur Rmarkdown et écrire du code Python comme si on était dans Pycharm ou Spyder.

12 Cartographie avec R

12.1 Introduction

La cartographie est un domaine essentiel qui offre une vue plus facile et compréhensible de certains phénomènes que les graphiques et tableaux ne peuvent bien exprimer.R offre plusieurs packages et fonctionnalités pour la cartographie, ce qui en fait un outil puissant pour l'exploration, l'analyse et la représentation visuelle des données spatiales.

12.2 Principaux packages

- `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,
- `popcircle`, représentation style bubble plot
- `raster`, manipulation de données raster
- `RColorBrewer`, palette de couleurs pour carto
- `rgdal`, import de données spatiales
- `sf`, nouvelle classe d'objets spatials
- `sp`, ancienne classe d'objets spatials
- `tidyverse, ggplot, dplyr`, etc
- `tmaptools` pour la carto

Pour faire une carte, il est important de préciser le type de projection dans lequel on veut représenter ça. Il s'agit du système de coordonnée de référencement (CRS).

12.3 Les types de données

Il existe essentiellement deux types de données exploitables pour réaliser une carte:

- Les vecteurs : Il utilise le concept d'objets géométriques comme les lignes (pour les routes par exemple), les points (pour des lieux par exemple) et les polygones (pour des régions ou quartier par exemple). Il existe sous deux types (le type shapefile et le type GSON).
- Les rasters : il s'agit des images le plus souvent satellites.

Les données spatiales sont téléchargeables sur des sites comme :

- https://www.gadm.org/
- https://geoservices.ign.fr/documentation/diffusion/telechargement-donnees-libres.html

Pour la réalisation de cartes avec des données de type shapefile, 4 fichiers de données sont essentielles et doivent porter les mêmes noms mais avec des extensions différentes. Ces extensions sont .shp, .dbf, .shx et le .prj

1. `.shp` : Le fichier .shp est le fichier principal qui stocke les géométries des entités géographiques. Il peut contenir des points, des lignes ou des polygones représentant des objets tels que des points d'intérêt, des routes, des rivières, des frontières, etc. Ce fichier contient les coordonnées géographiques qui définissent la forme et la position des entités.
2. `.shx` : Le fichier .shx est l'index spatial associé au fichier .shp. Il stocke des informations d'index qui permettent d'accéder rapidement aux entités géographiques dans le fichier .shp. Cela améliore les performances lors de l'affichage ou de l'interrogation des données géographiques.

3. `.dbf` : Le fichier .dbf est une table attributaire qui contient des informations non spatiales sur les entités géographiques. Il peut inclure des attributs tels que des noms, des descriptions, des valeurs numériques, des catégories, etc., associés à chaque entité géographique. Ce fichier permet d'associer des données tabulaires aux géométries stockées dans le fichier .shp.

4. `.prj` : Le fichier .prj est un fichier de projection qui spécifie le système de coordonnées utilisé par les données géographiques dans les fichiers .shp. Il indique comment les coordonnées géographiques sont interprétées spatialement. Ce fichier est essentiel pour garantir que les données géographiques sont correctement projetées et alignées avec d'autres couches de données.

13 Rshiny

13.1 Introduction

Nous sommes souvent amenés à faire des statistiques sur des variables d'une base donnée et le plus souvent nous écrivons des codes pour chaque action que nous voulons faire sur notre base de données en l'occurence, le tri à plat d'une variable, croisement des variables ou encore des modèles suivant des variables. Cette tache est vraiment répétitive si nous avons plusieurs variables dont on veut réaliser les memes actions. Ou encore, il se peut qu'on veuille suivre juste la distribution d'une variable par rapport à une autre mais qu'on on souhaite pas faire apparaittre dans notre rapport ou document finale. Pour rémédier à cela, R Shiny nous offre la possibilité de créer une application web interactive qui facilite les analyses de sorte qu'on a nos variables et nos différents modèles déjà programmés et on aura qu'à choisir les variables nous même au lieu d'écrire le code à chaque fois.

13.2 Présentation de Shiny

Shiny est un package R développé par RStudio qui permet de créer des applications web interactives. Avec Shiny, vous pouvez construire des interfaces utilisateur dynamiques sans avoir besoin de connaissances approfondies en HTML, CSS ou JavaScript. Shiny suit un modèle ui-serveur, où le serveur exécute le code R pour générer les sorties dynamiques, tandis que l'interface utilisateur (navigateur web) affiche l'interface et permet aux utilisateurs d'interagir avec l'application. Shiny facilite la création d'applications web pour l'analyse de données, l'affichage de graphiques, l'exécution de modèles statistiques, etc., le tout directement depuis R.

13.3 Les différentes parties d'une application Shiny

Une apllication Shiny comporte essentiellement deux parties :

- `L' ui` (interface utilisateur) : cette partie comprend tout ce qu'on voit sur l'application. C'est elle qui gère le visuel et ce qui nous captive sur l'application. Si l'on devait assimiler une application Shiny à l'homme cette partie est juste l'apparence physique.
- `Le server` (serveur) : c'est elle le cerveau qui gère tous les calculs derrière et nous renvoie les résulats qu'on voit sur l'application.

14 Conclusion

Lors de cet exposé sur divers sujets de R, nous avons pu explorer un large éventail de fonctionnalités et de packages qui enrichissent les capacités d'analyse et de manipulation des données dans R. Voici une conclusion qui résume brièvement chaque exposé :

`Janitor` : Le package Janitor offre des fonctionnalités pratiques pour nettoyer et transformer les données, facilitant ainsi la préparation des données en vue de l'analyse. Il permet de gérer les valeurs manquantes, de renommer les variables de manière cohérente et de réorganiser les données de manière efficace.

`GtSummary` : GtSummary est un package qui simplifie la création de tableaux récapitulatifs et de rapports statistiques à partir de données tabulaires. Il permet de générer rapidement des statistiques descriptives, des résumés de variables et des tableaux croisés, facilitant ainsi la communication des résultats d'analyse de manière claire et concise.

`RMarkdown et RQuarto` : RMarkdown et RQuarto sont des outils puissants pour la création de documents reproductibles combinant du code R, du texte formaté et des éléments visuels tels que des graphiques. Ils permettent de générer facilement des rapports dynamiques et interactifs, intégrant du code exécutable et des visualisations, ce qui facilite la communication des analyses et des résultats.

`R2Excel` : R2Excel est un package qui facilite l'importation et l'exportation de données entre R et Excel. Il permet d'échanger des données entre les deux environnements de manière transparente, offrant ainsi une intégration fluide entre R et Excel pour une utilisation plus efficace des données.

`Text Mining` : L'exploration de données textuelles est facilitée par les techniques de text mining, qui permettent d'extraire des informations significatives à partir de grands volumes de texte. Les packages R dédiés au text mining offrent des fonctionnalités pour le prétraitement des données textuelles, l'analyse de la sentiment, la classification de texte et la génération de mots-clés, permettant ainsi d'explorer et de comprendre les données textuelles de manière approfondie.

`Calcul Parallèle` : Le calcul parallèle dans R permet d'accélérer le traitement des données en répartissant les calculs sur plusieurs cœurs ou machines. Cela permet de réduire considérablement les temps de calcul, en particulier pour les tâches intensives en termes de calcul. Les packages R dédiés au calcul parallèle offrent des fonctionnalités pour exécuter des boucles parallèles, distribuer des calculs sur des clusters de calcul et optimiser les performances.

`Système d'équations` : Les packages R pour la résolution de systèmes d'équations permettent de modéliser et de résoudre des problèmes complexes impliquant des équations simultanées. Ces packages offrent des méthodes numériques et symboliques pour résoudre les systèmes d'équations, ce qui est utile dans de nombreux domaines tels que l'économie, les sciences sociales et l'ingénierie.

`Reticulate` : Le package Reticulate facilite l'intégration de Python dans R, permettant d'utiliser des fonctionnalités et des packages Python directement dans l'environnement R. Cela ouvre de nouvelles possibilités en termes d'analyse de données en combinant les atouts des deux langages, offrant ainsi une plus grande flexibilité et une plus large gamme d'outils et de ressources.

`Cartographie dans R` : Les packages de cartographie dans R offrent des fonctionnalités avancées pour la visualisation et l'analyse des données géospatiales. Ils permettent de créer des cartes interactives, d'effectuer des analyses spatiales et de superposer des données géographiques avec d'autres variables, ce qui facilite la compréhension et l'interprétation des données liées à la géographie.

`RShiny` : RShiny est un package qui permet de créer des applications web interactives à partir de R, sans avoir besoin de connaissances approfondies en programmation web. Il permet de construire des interfaces utilisateur dynamiques pour explorer et visualiser les données, exécuter des modèles statistiques et partager les résultats de manière conviviale, offrant ainsi une expérience interactive et engageante pour les utilisateurs.

En conclusion, les exposés sur Janitor, GtSummary, RMarkdown, RQuarto, R2Excel, Text Mining, Calcul Parallèle, Système d'équations, Reticulate, Cartographie dans R et RShiny ont mis en évidence la richesse des fonctionnalités et des packages disponibles dans R pour l'analyse des données, la manipulation, la visualisation et la communication des résultats. En utilisant ces outils, les utilisateurs de R peuvent améliorer leur productivité, explorer les données de manière approfondie, créer des rapports dynamiques et interactifs, et mettre en œuvre des solutions avancées dans divers domaines de l'analyse de données.

Nous tenons à remercier Monsieur Aboubacar HEMA, pour ces thèmes choisis pour les exposés sans oublier tous les groupes qui se sont mis à fond pour la présentation de ceux-ci.

References

Lasida, Elena, Kevin Minkieba Lompo, and Jean-Luc Dubois. 2009. "La Pauvreté : Une Approche Socio-Économique." *Transversalités* N° 111 (3): 35–47. https://doi.org/10.3917/trans.111.0035.