AGENCE NATIONALE DE LA STATISQUE ET DE LA DEMOGRAPHIE

ECOLE NATIONALE DE LA STATISTIQUE ET DE L’ANALYSE ECONOMIQUE



PROJET STATISTIQUE SOUS R : RESUME

MEMBRES DU GROUPE

AMADOU Moussa

TANGUOU KUETE Ivana

GNING Ibrahim

SOUS LA SUPERVISION DE :

HEMA ABOUBAKAR

ANALYST REASERCH

TABLE DES MATIERES

Table des matières

[2 Package Janitor : 5](#_Toc139461982)

[2.1 Fonctions principales 5](#_Toc139461983)

[2.2 Fonctions à l’intérieur du package 5](#_Toc139461984)

[2.3 Limites : 5](#_Toc139461985)

[3 package gtsummary 5](#_Toc139461986)

[3.1 A-Fonctions principales 5](#_Toc139461987)

[3.2 B- Fonctions à l’intérieur du package 6](#_Toc139461988)

[3.3 Limites 6](#_Toc139461989)

[4 Rmarkdown 6](#_Toc139461990)

[4.1 A- Fonctions principales 6](#_Toc139461991)

[4.2 Limites : 7](#_Toc139461992)

[5 QUARTO 7](#_Toc139461993)

[5.1 A- Fonctions principales 7](#_Toc139461994)

[5.2 Limite : 8](#_Toc139461995)

[6 R vers Excel : R2excel 8](#_Toc139461996)

[6.1 Quelques fonctions dans le package 8](#_Toc139461997)

[7 SYSTEME D’EQUATION LINEAIRE 8](#_Toc139461998)

[7.1 a- méthode directe 8](#_Toc139461999)

[7.2 b- méthode indirecte 9](#_Toc139462000)

[8 python dans R 9](#_Toc139462001)

[8.1 Pourquoi R dans R ? 9](#_Toc139462002)

[8.2 Recommandations : 10](#_Toc139462003)

[9 CALCUL\_PARALLELE 10](#_Toc139462004)

[9.1 1-Comment importer les données de grande taille ? 10](#_Toc139462005)

[9.2 2-Packages nécessaires pour le calcul parallèle 10](#_Toc139462006)

[10 Cartographie sur R 11](#_Toc139462007)

[11 R SHINING 11](#_Toc139462008)

Introduction

Afin de permettre aux élèves de parfaire les notions essentielles sur logiciel Rstudio, nous avons eu l’honneur non seulement de faire une présentation sur une thématique mais également de faire le résumé des présentations des autres groupes. Pour ce faire nous avons fait le résumé suivant l’ordre de présentation des différents groupes.

# Package Janitor :

## Fonctions principales

1- Apurement et le nettoyage des données  
2- Effectuer des opérations sur des tableaux

## Fonctions à l’intérieur du package

Fonctions de nettoyage  
a- la fonction clean\_names() permet de labéliser les variables  
b- la fonction compare\_df\_cols() compare deux dataframe  
c- la fonction make\_clean\_names() permet de labéliser également  
d- la fonction remove\_empty() permet le nettoyage de valeurs manquantes  
c- la fonction remove\_constant() permet le nettoyage des constantes  
d- la fonction get\_dupes() permet le nettoyage des doublons  
Les tableaux  
tabyl () permet de faire des tableaux de

## Limites :

Après présentation du package le groupe a mentionné quelques limites du package tels que :  
• il faut appel à d’autres packages pour pouvoir faire des tableaux, sans quoi ça ne marchera pas  
• suivre l’ordre dans l’utilisation des paramètres adorn dans la fonction tabyl  
• il ne peut pas combiner trois tableaux

# package gtsummary

## A-Fonctions principales

1- permet de faire des tableaux statistiques des tri à plat  
2- surtout réputé pour les évaluations d’impact

## B- Fonctions à l’intérieur du package

a- tbl\_summary() pour créer un tableau de synthèse descriptif pour une variable ou un groupe de variables  
b- tbl\_uv() pour créer un tableau de synthèse croisé pour deux variables  
c- tbl\_regression() pour créer un tableau de synthèse pour une régression linéaire  
d- tbl\_merge() pour fusionner plusieurs tableaux de synthèse en un seul. Il faut noter qu’à l’intérieur de chaque fonction il y a des arguments qu’on peut spécifier en fonction de nos besoins.

## Limites

Bien qu’il permette de faire de bon tableaux, il peut être difficile d’obtenir une table exactement comme souhaité sans avoir à écrire du code supplémentaire.

# Rmarkdown

## A- Fonctions principales

Il permet de produire des documents word, pdf, html. Il offre une multitude de possibilités avec notamment les présentations où il combine différents langage de programmation tels que R lui-même, python, SQL etc. En plus il permet de faire la reproduction d’un document en conservant les mises en forme de base. Contrairement aux packages précédent ici il s’agit plutôt d’un interface composé de deux blocs à savoir le yaml et le body  
• yaml : on peut l’appeler entête puisqu’il permet de spécifier les ingrédients d’entrés c’est à dire :  
– nom du document  
– nom de l’auteur  
– type de document  
– la date de sortie  
– on peut également spécifier d’autres options en fonction de ce que nous voulons avoir comme output  
• Body : c’est dans cette partie qu’on écrire notre texte, dessiner des tableaux. On peut écrire des codes à l’intérieur à travers les chunk selon le langage souhaité  
Bien que RMarkdown soit un outil très puissant pour la création de documents dynamiques, il présente quelques

## Limites :

• la personnalisation de la mise en page peut être difficile pour les utilisateurs qui ne sont pas familiers avec les langages de balisage tels que HTML et CSS.  
• les fonctionnalités de traitement de texte avancées, telles que les tableaux complexes et les graphiques interactifs, peuvent nécessiter des compétences en programmation avancées.

# QUARTO

## A- Fonctions principales

Présentation de l’interface  
Il existe des onglets pour les manipulations comme dans Microsoft.  
• insertion de titres  
• insertion d’image  
• mettre en forme les styles de police  
• mettre des images côte à côte (insert)  
• insertion de diagramme (avoir les compétences en graphiz ou Mermaid)  
• insertion de graphique < • insertion de saut de page {{<- pagebreak >}}  
• insertion de lien ( aller sur link et cliquer sur link )  
• insertion d’équation (code latex ) c’est-à-dire aller à insert latex math (numérotation automatique)  
• insertion de caractères spéciaux ( des caractères emoji)  
• insertion de lien hypertexte (aller à insert)  
• insertion de blocs notes  
• insertion de bibliographie ( aller à insert citation )  
On peut également faire une présentation quarto avec plusieurs fonctionnalités.  
• faire des animations de code avec auto animation = TRUE  
• mise en surbrillance des lignes de codes  
• Faire des animations sur les textes  
• mettre des commentaires en mode présentation  
• faire du magicland (agrandir rétrécir …..)  
• changer le fond background  
• mettre des images en fond d’écran  
• Animer des objets (formes)   
• faire de tabulation (Tabsets)   
• insertion de carte Jupiter widgets, html  
• faire un tableau virtuel (Tableau)   
On peut en outre créer un livre quarto. Pour ce faire, il faut suivre le chemin suivant :   
file->new projet->new directory->quarto book

## Limite :

Bien que le groupe ait présenté quarto, il a souligné que quarto est récent et donc il reste beaucoup à faire pour qu’il soit totalement opérationnel.

# R vers Excel : R2excel

L’auteur du package R2excel est cassabara. Pour l’installer il faut avoir au préalable une distribution java sur sa machine et en plus de cela il faut charger le package xlsx pour pouvoir interagir avec Excel. Pourquoi R vers Excel ? Il permet aux statisticiens d’être opérationnel c’est à dire anticiper sur l’analyse descriptive et donc rédiger rapidement son rapport d’une enquête de collecte de données.

## Quelques fonctions dans le package

• xlxs.addheader entête  
• xlsx.addline lignes  
• xlsx.addtable ajout de tableau  
• xlsx.addlinebreak fin du tableau  
• ajouter de paragraphes  
• xlsx.addparagraphe  
pour sauvegarder xlsx:: saveworkbook(wb,filename)  
• labelisation apply\_labels()  
• ouverture du fichier xlsx.openfile(filename)  
Limites : Bien qu’il permet de gagner en temps, il présente un inconvient majeur puisqu’il n’est pas encore disponible sur le site du cran et son installation cause un grand problème.

# SYSTEME D’EQUATION LINEAIRE

Il s’agit ici d’un problème économique comment allouer son revenu pour avoir un niveau de consommation sain. Deux méthodes ont été présentées

## a- méthode directe

1-package nécessaire rootSolve  
fonction interessante dans le package:  
• multiroot pour resoudre des systeme d’équation linéaire multvarié NB: souvent il est interessant d’intégrer un seuil c’est à dire une tolérance d’erreur.  
2-package nleqslv  
La particularité ici on inverse les paramètres d’entré dans la fonction  
NB: la méthode de newton est la plus importante

## b- méthode indirecte

Il faut noter cette méthode a été qualifiée par les exposants comme la méthode du statisticien. Ainsi les packages sont déjà incorporés dans Rstudio et de plus les fonctions portent le même nom que les packages. Le groupe a mentionné :  
• optim  
• stats  
• optimx  
Pour finir un cas pratique a été fait sur une petite base afin de trouver les coefficients optimaux.

# python dans R

## Pourquoi R dans R ?

Nous faisons une formation d’ingénieur et donc on ne sait pas après la formation où on peut être travaillé et avec quels genre d’utilisateurs de langage allons-nous faire face. Il est donc indispensable de de se familiariser avec les différents logiciels.  
Package nécessaire : reticulate  
Comment exécuter un script python dans R ?  
Il y a deux façon de faire à savoir :  
a- Soit directement  
Reticulate :: py\_run\_file(chemin d’accès)  
Pour installer des modules python dans R :  
Reticulate ::Py\_install(module)  
b- installer un environnement virtuel python sur Rstudio pour effectuer des opérations :  
Reticulate :: Py\_install(nom du module , nom de l’environnement )  
Comment importer un module ?  
imd <- import (nom du module)  
Pour accéder aux fonctions contenu dans le module on utilise le symbole $  
Comment appeler des fonctions Python depuis R ?  
Référencer le script Python  
reticulate :: source\_python(“chemin d’accès au script”)  
Utiliser la fonction  
result <- nom\_fonction(arg1, arg2).  
Comment utiliser Python dans R ?  
reticulate:: py\_run\_string(“code”)  
Comment conversion des objets de python vers R et vice versa ?  
convertir un objet R en python  
objet\_py = reticulate::r\_to\_py(objet\_r)  
convertir un objet Python en R  
objet\_r = reticulate::py\_to\_r(objet\_py)

## Recommandations :

 Demande une maîtrise de la syntaxe Python et de la syntaxe R  
 Dépendance aux packages Python: vous devez toujours installer les packages avant de les importer

# CALCUL\_PARALLELE

Le calcul parallèle consiste à faire des opérations sur des données de grande taille qu’on appelle encore le Bigdata. Le principe consiste à affecter des tâches spécifiques aux cœurs de la machine afin de gagner en temps d’exécution de la machine autrement qu’on divise pour mieux régner.

## 1-Comment importer les données de grande taille ?

On utilise le package fs pour importer les données de grande taille. De plus il faut vider la mémoire de la machine avec la commande rm(list=ls()) afin de faire oublier la machine tout ce qu’elle a en mémoire.

## 2-Packages nécessaires pour le calcul parallèle

• package parallèle permet demarrer et d’arrêter le processus  
• package doparallele gère toutes les phases entre le début et la fin du processus  
3- Fonctions essentielles  
• Detectcores permet de détecter le nombre de cœurs  
• Makecluster() permet de créer un cluster de stockets parrallèles  
• stopcluster() permet d’arrêter les cœurs  
• La famille des fonctions Applay permet de faire le traitement

Á travers l’étude de cas, on a mentionné qu’il est inutilité d’appliquer le calcul parrallèle si les données ne sont pas si vastes et dans le cas échéant la méthode du calcul parrallèle prend beaucoup plus de temps que la celle directe.

# Cartographie sur R

Pour faire cartographie, on a des coordonnées géographiques qu’il faut, transformer en coordonnées cartographique, orienter l’échelle du système de coordonnées et la référence utilisées avec la fonction CRS ; il faut également une maitrise de Fichiers de type spatiales.  
CRS : Il permet de spécifier les référentiels et gère l’affichage des images (spécifie les formes des images)  
library(sp)  
crs <- CRS(“+proj=utm +datum=WGS84 +ellps=WGS84”)

Il y a plusieurs données de type spatiale :  
• Le mod Raster : ce sont des données sous forme image avec des extensions du type .png ou .jpg ; .asc et .db  
• Le mode vecteur : ce sont des données sous format ligne, point ou polygone ; il existe principalement deux formats d’importation de donnée : shapefile et GeoJSON .  
Shapefile : on l’utilise pour importer les données géographiques, sa création engendre la création automatique de 4 fichiers de même nom mais d’extensions différentes à savoir : .shp qui contient des polygones le plus important c’est-à-dire appeler ce fichier dans un programme est suffisant pour appeler tous les autres, mais l’inverse n’est pas vrai ; .dbf contient des attribut ou base de données ; .shx qui permet de suivre les index et enfin de .gbb qui contient les identifiants.  
Gson : Il a l’avantage de contenir l’information (stockée dans un langage tou à fait compréhensible) dans un seul fichier  
. Manipulation des données spaciales  
Pour manipuler les données il est important de passer des données non spatiales aux données spatiale avec la fonction coordinates et pour visualiser on utilise la fonction str.  
Le symbole @ est utilisé pour accéder aux éléments de la base. On peut également ajouter une colonne géométrique longitude et latitude en transformant le fichier sous format sf en utilisant la fonction sf. Cependant il faut noter que l’importation des fichiers dépendant du format. Si c’est au format sp on utilise readOGR , si c’est au format sf on utlise st\_read  
Il existe d’autres fonctions qui permettent de faire la manipulation des données telles que ajouter des cartes sur une meme figure etc.

# R SHINING

R shining est une application de R composée de deux parties, une interface utilisateur (User Interface) et une fonction serveur. De ce fait lors de la création d’un fichier shinny il est nécessaire de choisir entre créer un single file ou un multiple file. Afin ue les codes soit rapide lors de l’exécution, il est nécessaire de de créer deux fichiers de programmation, l’un coté serveur et l’autre pour l’UI. Ainsi pour utiliser les composantes de l’UI au niveau du fichier serveur , il est important de  
Il est intéressant de noter que chaque bloc ou type de variable est indéxé par son nom en d’autre termes tout ce qui a trait avec un frame ou interface commencera par Fluid ; ce qui a trait avec un Items PAR Items ; Ce qui a trait avec une barre déroulante Slide ……..

1-User interface : L’interface utilisateur c’est la partie visible et interactive de l’application. Elle permet aux utilisateurs d’interagir avec l’application, de saisir des données, de visualiser des résultats et d’effectuer des actions spécifiques.  
FluidPage : Crée une page Shiny fluide qui s’adapte à la taille de la fenêtre du navigateur  
TitlePanel : Affiche un titre en haut de la page Shiny.SelectInput / SlideInput :  sélectionner un élément de la liste/ faire varier la barre  
TextInput : Permet de créer un widget de saisie de texte dans une application Shiny.  
tableOutput: permet de générer un tableau depuis le server dans l’interface utilisateur.

2- Fonction serveur : Le server est “le cerveau” de l’application shiny. Il reçoit les interactions de l’utilisateur depuis l’interface utilisateur et génère des sorties dynamiques en fonction de ces interactions.  
Quelques fonctions :  
reactive() :permet de créer une base de données reactive et de mettre des contrôles sur le type d’importation de base ie qui met à jours les donées de la nouvelle base apres chaque importation de base.  
observeEvent() :permet de constater qu’il changement de base  
renderUI(): Elle permet de générer et d’afficher du contenu HTML ou des éléments d’interface utilisateur dynamiquement, en réponse à des événements ou à des calculs.