Programmation dans R
Ch. 1. Bases de R
M2 CEE

Pr. Philippe Polomé, Université Lumière Lyon 2

2018 – 2019

GATE Lyon / St-Etienne

UNIVERSITÉ DE LYON

Sommaire

SWIRL

- Vous avez fait Course 1 : R programming Lessons 1-9 + 14
  - On refait Lesson 1 pour voir si ça marche pour tout le monde
  - pour sortir d'une lesson : esc
  - répondez "no" aux propositions de "register"
  - À la suite de ...
    - il faut presser ↩
    - Parfois, il s'ensuit pas mal de texte – lisez tout
- Suivre les commandes sur PR2017.R
  - Avec les diapos

SWIRL R programming

- Passez maintenant à R-Studio
  - Swirl R programming 1-9 + 14

1 : Basic Building Blocks	2 : Workspace and Files
3 : Sequences of Numbers	4 : Vectors
5 : Missing Values	6 : Subsetting Vectors
7 : Matrices and Data Frames	8 : Logic
9 : Functions	10 : lapply and sapply
11 : vapply and tapply	12 : Looking at Data
13 : Simulation	14 : Dates and Times
15 : Base Graphics	

Quelques commandes hors SWIRL

- On va voir quelques compléments de bases, hors SWIRL
  - Suivre les commandes sur PR2017.R
- Fonctions math communes : log, exp, sign, sqrt, abs, min, max
  - log(exp(sin(pi/4)^2)*exp(cos(pi/4)^2))
- Vecteurs spéciaux
  - ones <- rep(1, 10)
  - even <- seq(from = 2, to = 20, by =2)
  - trend <- 1981 :2005
- diag(4) matrice identité de taille 4

Opérations sur matrices

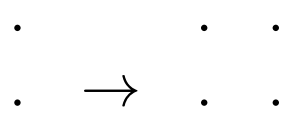
- A<-matrix(1 :6, nrow = 2)
  - A regardez à quoi ça ressemble et comment R donne la position des éléments
- t(A) = transposée de A (*pas A'*)
- dim(A) = dimensions de A (L puis C)
- nrow(A) ; ncol(A) nbr L ; C
- A[i,j] extrait l'élément (i,j)
- A[,j] extrait C j (toutes les L) en un vecteur
  - A[i,] même chose pour L i
- A1<-A[1 :2, c(1, 3)] A1 a 2 L qui contiennent les 1er et 3eme éléments de chaque L de A
  - autre façon A[,-2]

Opérations sur matrices

- det(A1) déterminant
- eigen(A1) eigenvalues
- chol(A1) décomposition de Cholesky
- solve(A1) inverse
- A %*% B produit matriciel
  - A*A produit élément-par-élément
  - kronecker(A, B) produit de Kronecker $\otimes$
- crossprod(A, B) est un calcul efficient de A'B
- diag(A1) extrait la diag

Opérations sur matrices

- cbind(1, A1) "combine" une C de 1 et A1



- rbind(A1, diag(4, 2)) "empile" A1 et une matrice diag de taille 2 avec des 4 sur la diag

Sommaire

Dataframe

- "Frame" = cadre "contexte"
  - Dans R, un "Dataframe" est une matrice de données
    - un ensemble de vecteurs de même longueur
    - mis l'un à côté de l'autre horizontalement
- Chaque vecteur = 1 C = 1 variable
  - possiblement de nature $\neq$
    - quantitative, numérique mais qualitative, caractères, dates...
  - qui contient de plus des méta-informations
    - p.e. le type de variable ou le nom des catégories
- Chaque L = 1 observations
- Rem.
  - Une "array" est, dans R, un objet plus général car elle peut avoir + de 2 dimensions
  - Un "tibble" est une version plus récente d'un dataframe

Création

- Un Dataframe peut être créé de pls façons
  - clavier, lecture fichier R, importation
- Création clavier (cfr Swirl programming lesson 7)
  - alternative 1
    - mydata <- data.frame(one = 1 :10, two = 11 :20, three = 21 :30)
  - alternative 2
    - mydata <- as.data.frame(matrix(1 :30, ncol=3)) and names(mydata) <- c("one", "two", "three")
  - Clairement, R n'est pas le meilleur logiciel pour entrer des données manuellement

attach

- Lorsqu'un dataframe est "attached"
  - en utilisant la commande attach,
  - alors les noms des variables dans le dataframe peuvent être utilisés dans des commandes
- Par exemple
  - mean(two) produit un message d'erreur
  - attach(mydata) et puis mean(two) produit la moyenne de la variable "two"
- detach(mydata) permet de détacher le Dataframe
  - p.e. pour éviter des confusions sur les noms de variables
- Pour attacher pour une seule opération
  - with(mydata, mean(two))

Sélection de sous-ensemble (subset)

- On accède à un sous-ensemble du Dataframe par [ou $
  - $ extrait une seule varaible
- Pour travailler avec un sous-ensemble
  - on utilise [or subset
- mydata.sub<-subset(mydata, two<=16, select = -two)
  - prend toutes les observations des variables one & three
    - pour lesquelles les observations correspondantes de la two sont ≤ 16

Exporter (write) un dataframe

- write.table(mydata, file="mydata.txt", col.names=TRUE)
  - crée un fichier texte mydata.txt dans le répertoire de travail
  - Les méta-informations ne passent pas
  - Le format du fichier texte est

"one"	"two"	"three"	
"1"	1	11	21
"2"	2	12	22
...			

  - Rem. on dirait que les intitulés de colonnes sont décalées à G
    - Selon le logiciel qui servira à ouvrir, il faut parfois insérer un espace

Importer (read) un dataframe

- À parti d'un fichier texte (.txt ou .csv)
  - newdata <- read.table("mydata.txt", header=TRUE)
    - lit un fichier texte dont la première ligne porte le nom des variables
    - qui est placé dans un "data.frame" appelé newdata
  - Si les titres de col. ne sont pas décalées à G
    - comme si ça venait d'un tableur
    - R comprendra la col. avec les numéros de ligne comme une autre variable
- read.table accepte des options
  - sur le séparateur de col. (, ;)
  - sur le séparateur décimal (. ,)
  - peut lire du .csv
  - voir ?read.table

Importer un dataframe

- scan est utilisé pour des données qui ne sont pas en matrice
  - ?scan pour les détails
- Pour importer d'un autre système
  - Le + facile exporter de ce système en txt ou csv
    - perte des méta-données
    - rarement possible (pas accès à l'autre système)
  - R-Studio propose pls formats
    - Ne marche pas svt car ces logiciels changent souvent de format
  - Google p.e. "R import Stata 17 data"
  - Voir www.statmethods.net/input/importingdata.html
    - pour quelques formats

Un peu de pratique : "Journals"

- Exemple, les données JOURNALS :
- souscriptions à des journaux d'économie pour quelques bibliothèques aux USA en 2000
  - 180 observations (les journals) sur 10 variables, entre autres :
    - subs (# of library subscriptions),
    - price (subscription price),
    - citations (total number of cites per journal)
- On étudie la relation entre le nbr de souscriptions et le prix par citation
  - La citation reflétant l'importance du journal
    - Donc l'intérêt pour une bibliothèque

Un peu de pratique : "Journals"

- data("Journals", package = "AER")
  - Journals$citeprice <- Journals$price/Journals$citations
  - " / " est une division élément par élément pour des vecteurs
- attach(Journals)
  - "attacher" pour utiliser dans les formules
- Calculez l'estimateur MCO de la régression de log(subs) sur log(price/citations)
  - et ses t-stats
  - au moyen des formules de mtx
    - Difficulté car $\hat{\sigma^2}$ est perçu comme une matrice 1×1 non conformable avec $\left(X'X\right)^{-1}$
    - Pls solutions p.e. $\otimes$ produit Kronecker
    - Résultats : 85 et -15 environ ?
- detach(Journals)
  - "détacher" pour éviter des confusions de nom

dplyr

- package dplyr - library(dplyr)
  - Partie d'un ensemble de packages "tidyverse" visant à simplifier la gestion de données dans R
- Pour illustrer, on va utiliser le jeu de données fligths
  - install.packages("nycflights13") puis charger avec library(nycflights13)
  - dim(flights) : quelles dimensions ?
  - flights pour voir à quoi il ressemble
- Filtrer des lignes avec filter( )
  - 1º arg : le jeu de données, 2º argument et les suivants : condition pour sélectionner les L si l'arg est true
  - Exercice : sélectionner dans flights les vols du mois de janvier uniquement
    - Puis du 1º janvier uniquement

dplyr

- Trier des lignes avec arrange( )
  - Fonctionne comme filter, mais ordonne selon le 2º arg, puis le 3º (s'il est présent) ...
    - Exercice : ordonner flights selon l'année, le mois et le jour
  - Utiliser desc() pour ordonner en décroissant
    - Exercice : ordonner flights par ordre décroissant de retard à l'arrivée arr_delay
- Sélectionner des col avec select()
  - Souvent, on ne travaille qu'avec quelques colonnes
- Ajouter des col avec mutate()
- Il y a qqs autres fonctions
  - sample_n() et sample_frac() pour prendre un échantillon aléatoire de L
  - Résumer des valeurs avec summarise()
  - Moins utiles

Sommaire

Plot

- D'abord SWIRL
  - course R-programming, lesson 15 Base graphics
- Quelques éléments graphiques supplémentaires
  - Sur base du package **plot**
  - Autres packages
    - **lattice** plus sophistiqué
    - on fera **ggplot2** un peu + loin
    - http ://varianceexplained.org/RData/code/code_lesson2/#segment1
  - R est considéré comme ayant de très bons graphiques
- La commande plot ( )
  - plot( ) est la commande par défaut pour représenter graphiquement beaucoup d'objets :
    - dataframes, séries temp, modèles linéaires ajustés
    - C'est aussi une vielle commande, assez rustique

"Scatterplots" - graphiques de dispersion ou XY

- Probablement les + communs en économétrie
- attach(Journals)
  - plot(log(subs), log(citeprice))
- detach(Journals)
- plot(log(subs)~log(citeprice), data=Journals)
  - alternative pour éviter d'avoir à attacher le dataframe

R Paramètres Graphiques

- Les résultats d'un plot peuvent être modifiés de nombreuses façons
  - P.e. argument type contrôle si le plot génère des points (type = p), lignes (type = l), les 2 (type = b), des pas (type = s) ou d'autres
- Pls douzaines de paramètres modifiables sont disponibles
  - Voir ?par
  - Soit en les fixant avec par( ) après une commande plot
  - Soit en les fournissant à dans la fonction plot( ) p.e. plot(log(subs)~log(citeprice), data=Journals, pch=20, col="blue", ylim=c(0,8), xlim=c(-7,4), main="Library Subscriptions")
- Prochaine diapo : liste de par

R Paramètres Graphiques

Argument	Description
axes	should axes be drawn ?
bg	background color
cex	size of a point or symbol
col	color
las	orientation of axis label
lty, lwd	line type and line width
main, subs	main title and subtitle
mar	size of margins
mfcol, mfrow	array defining layout for several graphs on one plot
pch	plotting symbol
type	types
xlab, ylab	axis labels
xlim, ylim	axis ranges
xlog, ylog, log	logarithmic scales

R Paramètres Graphiques

- Ajouter des couches à un plot : lines( ), points( ), text( ), legend( )
  - Ajouter 1 droite abline(a, b)
    - a est intercept, b pente
- 2 plots l'un sur l'autre
  - x <- rnorm(50)
  - x2 <- rnorm(50, -1)
  - plot(ecdf(x), xlim = range(c(x, x2)))
    - ecdf empirical cumulative density function
  - plot(ecdf(x2), add = TRUE, lty = "dashed")
- Barplots, pie charts, boxplots, QQ plots & histograms
  - barplot( ), pie( ), boxplot( ), qqplot( ), hist( )
  - On y reviendra

Exporter des graphiques

- Pour utiliser les graphiques R dans d'autres logiciels
  - il faut les exporter
- Le + simple : bouton "Export" Plots window
- R peut aussi envoyer le graphe sur un "device"
  - Au fond : une extension de fichier pdf ou jpg
    - Tous les devices fonctionnent pareil dans R
    - Voir ?devices

1. le device est ouvert par une fonction qui porte son nom, p.e. pdf( )
2. Ensuite, le plot est exécuté
3. Finalement, le device est fermé dev.off( )

- Exemple
  - pdf("myfile.pdf", height=5, width=6)
  - plot(1 :20, pch=1 :20, col=1 :20, cex=2)
  - dev.off()
  - Cherchez myfile.pdf sur votre machine

Formules math dans un Plot

- R peut passer une formule dans un plot via LaTeX
  - Voir ?plotmath
- Exemple
  - plot de la densité normale std avec sa définition math
  - curve(dnorm, from=-5, to=5, col="slategray", lwd=3, main="Density of the Standard Normal Distribution")
  - text(-5, 0.3, expression(f(x) == frac(1, sigma ~~ sqrt(2*pi)) ~~ e^{-frac((x - mu)^2, 2*sigma^2)}), adj=0)
    - Malheureusement, il faut connaitre LaTeX
    - & les paramètres ne sont pas faciles

Histogrammes & boxplots

- data("CPS1985")
  - jeu de données sur le salaire & ses déterminants
- summary(CPS1985)
  - révèle que certaines variables sont catégoriques
    - Catégoriques : appelées factors par R
- Factors = vecteurs de "noms" appelés catégorie
  - parfois avec metadata p.e. noms des catégories
  - g <- rep(0 :1, c(2,4))
  - g <- factor(g, levels=0 :1, labels=c("male", "female"))
    - Nomme cat. (0,1) de g en "Male"(=0) & "Female"
    - donc g est [1] male male female female female female

6 Factors dans CPS1985

- Factors sont comptés (fréquences)
  - car la moyenne ne signifie rien
- levels(CPS1985$occupation)[c(2,6)] <- c("techn", "mgmt")
  - abrège 2 noms d'occupation
- attach(CPS1985)
  - permet d'accéder aux variables de CPS1985 par leurs noms
- On va voir des graphes selon la nature des données
  - Numérique ou cat.
  - 1 seule variable ou 2 en relation

Une variable numérique : histogramme & densité

- hist(wage, freq=FALSE)
  - option freq=FALSE
    - fréquences relatives, sinon absolues (comptage)
  - option binwidth=x
    - permet de choisir la longueur des bases
- hist(log(wage), freq=FALSE)
- lines(density(log(wage)), col=4)
  - La fonction density calcule un histogramme lissé (voir cours np)
- Remarque
  - La distribution du log est moins asymétrique que celle des données brutes
  - Les données en log sont svt + proches d'une normale

Une catégorique

- La moyenne & la variance ne signifient rien
  - mais bien les fréquences
- summary(occupation) : fréquences absolues (comptes)
- tab <- table(occupation) : stocke ces fréq. ds une table
- prop.table(tab) calcule les proportions (fréq. relatives)
- Barplots & pie visualisent souvent bien les données cat.
  - barplot(tab)
  - pie(tab)

2 catégoriques

- Habituellement présentées ds une Table de Contingence
  - xtabs( ) avec une interface formule :
    - p.e. xtabs(~ gender + occupation, data = CPS1985)
    - data= optionel car on a attaché
  - table(gender, occupation) mêmes résultats
- Plot de ça est un "spine plot"
  - plot(gender ~ occupation)
  - plot(gender, occupation) regardez les différences

2 numériques

- Coefficient de corrélation r est typique
  - Variables positives & asymétriques : Spearman's ρ
    - corrélation des *ranks*, au lieu des valeurs est souvent préféré car r n'est pas robuste à l'asymétrie
- cor(log(wage), education)
- cor(log(wage), education, method="spearman")
  - Résultats semblables pour ces données
- plot(log(wage)~education)
  - Le scatterplot montre peu de corrélation

1 numérique & 1 catégorique

- Souvent, on calcule des moments conditionnels
  - p.e. salaire moyen selon le sexe
  - tapply(log(wage), gender, mean)
    - "Appliquer" mean sur les 2 variables gender & log(wage)
    - Mean peut être remplacé par n'importe quelle fonction valide
- Les Box plots & QQ (quantile-quantile) plots sont souvent utilisés

1 numérique & 1 catégorique : Box plot

- Un box plot est une représentation grossière d'une distribution empirique
  - Le box est limité par des "charnières" (1º & 3º quartiles) et montre la médiane
  - Hors du box, 2 lignes indiquent les obs. les + petites & + grandes
    - à moins de 1.5 × taille du box à partir de la charnière la + proche
  - Toute obs. au-delà est représentées séparément par un point
- boxplot(log(wage)~gender)

1 numérique & 1 catégorique : QQ plot

- Un QQ plot apareille les quantiles de 2 distributions
  - Se rappeler que les quantiles sont des quantités
    - p.e. le 1º quartile du salaire féminin est le salaire t.q 25% des femmes gagnent moins & 75% +
  - Si les 2 distributions sont identiques
    - le QQ plot est la diagonale
  - Sinon, si p.e. les hommes tendent à gagner + que les femmes, alors
    - avec les hommes sur l'axe des x, le QQ plot sera sous la diag.
    - Évoque le graphe du coefficient de Gini
  - `mwage <- subset(CPS1985, gender == "male")$wage`
  - `fwage <- subset(CPS1985, gender == "female")$wage`
  - `qqplot(mwage, fwage)`
  - `abline(0,1)`
- `detach(CPS1985)` pour refermer CPS1985

Sommaire

Intro

Installez & chargez ggplot2
ggplot2 a des données intégrées

- data("diamonds")
- View(diamonds)
  - Carat = poids, cut, color, clarity, prix
    - Toutes des mesures du diamant
  - Autres attributs : help(diamonds)
- Scatterplot : représenter les liens entre ces caractéristiques
  - On peut choisir les axes, mais aussi
    - couleur, taille & forme des points

1º pas

- ggplot(diamonds, aes(x=carat, y=price)) + geom_point()
  - Comment marche cette cmd ?
- 3 parts ds 1 cmd ggplot
  1. data, on écrit "diamonds"
  2. Organisation des attributs
     - aes = “aesthetics”
     - **aesthetics** = toute dimension d’un graphe qu’on peut percevoir visuellement
     - On utilise des parenthèses
  3. On rajoute une “couche” (layer)
     - Ici, on indique qu’on veut un scatter plot avec “geom_point”
     - Bcp de couches ont un nom qui commence par "geom"

Ajouter color

- ggplot(diamonds, aes(x=carat, y=price, color=clarity)) + geom_point()
  - On rajoute un “aesthetics”, ici la couleur
  - Qu’on associe au factor “clarity”
    - Remarquez que ggplot a créé une légende en couleur
    - Essayer avec d’autres factors : regardez dans le fichier de données
- Pour représenter à la fois color & cut
  - on rajoute simplement un aesthetic
  - qui peut être la taille du point : size
    - Faites-le
    - Au lieu de “size”, on peut aussi utiliser “shape”

Couche supplémentaire

- Le scatter plot est un "layer"
  - On peut en rajouter avec le signe +
  - p.e. une courbe lissée pour la tendance générale : geom_smooth
  - En repartant du 1º plot
    - ggplot(diamonds, aes(x=carat, y=price)) + geom_point() + geom_smooth()
  - La zone grise autour de la courbe est un intervalle de confiance
    - On peut l'enlever avec l'option geom_smooth : "se=FALSE"
  - On peut aussi mettre une ligne de régression simple avec l'option method="lm"
    - lm pour "linear model"
    - refaites la cmd avec les 2 options
- Si on utilise en plus l'aesthetic "color"
  - ggplot va estimer une courbe lisse pour chaque couleur
  - Allez-y
  - Si vous enlevez geom_point, vous ne verrez que les courbes

Faceting : plot multiple

- Une autre façon de communiquer l'info sur un attribut factor
  - Diviser le plot en multiples plots
    - un pour chaque niveau du factor (pas trop de niveaux)
  - On appelle ça "faceting"
    - ggplot : couche "facet_wrap"
  - P.e. faceting sur "clarity" sur des plots (carat, price, cut)
    - ggplot(diamonds, aes(x=carat, y=price, color=cut)) + geom_point() + facet_wrap(~ clarity)
  - On peut rajouter une dimension de faceting avec p.e. facet_wrap(cut~ clarity)
    - Mais ça n'est plus très clair
- On peut rajouter un titre avec + ggtitle("My scatter plot")
  - changer la légende de x + xlab("Weight (carats)")
  - Limiter la longueur des axes, mettre en log...

Histogrammes & Densités

- Histogramme : on ne regarde qu'un seul attribut
  - ggplot(diamonds, aes(x=price)) + geom_histogram()
  - Il suffit de changer + geom_point() en + geom_histogram()
  - R renvoie un signal sur "binwidth" = taille de la base du rectangle
    - Défaut = 30 rectangles
    - On peut changer, p.e. ggplot(diamonds, aes(x=price)) + geom_histogram(binwidth=2000)
- On peut faire à peu près la même chose qu'avec un scatterplot
- Si on écrit geom_density au lieu de geom_histogram
  - on a une densité lissée au lieu d'un histogramme
  - Comment calculer cette densité n'est pas évident

Boxplot & Violons

- Boxplot du prix selon les niveaux de couleur
  - ggplot(diamonds, aes(x=color, y=price)) + geom_boxplot()
  - Il y a bcp d'outliers
    - En rajoutant + scale_y_log10() on les voit moins
- Si la distribution empirique ne suit pas $\pm$ une normale
  - Le box plot est pas terrible
    - p.e. si on a 2 modes
- Une image possiblement + fidèles est le "violin"
  - On remplace geom_boxplot() par geom_violin()
  - Pas super clair si on ne passe pas aux log
  - La largeur en chaque prix est la fréquence de ce prix

Devoir 1

- PARADE
  - Supplément du dimanche de 500 quotidiens US
  - Tous les ans, un échantillon de 120-150 citoyens sélectionnés "aléatoirement"
    - variables : profession, hometown, state & yearly earnings.
  - dataframe Parade2005 contient la version 2005 version
    - avec une dichotomique "celebrity status" fortement sur-échantillonné
- Devoir 1
  - Earning moyen en Californie, discuter
  - Nombre d'échantillonnés en Idaho
    - Qu'en dire sur l'échantillon ?
  - Earning
    - moyen & médian des célébrités, commenter
    - Fréquence & densité
    - Boxplot selon statut célébrité
    - Répéter avec log
    - Essayer d'utiliser ggplot pour améliorer ce boxplot graphiquement
  - Adapter pour un autre jeu de données
  - M'envoyer le fichier .R avant l'examen

Sommaire

SWeave – Knitr – Markdown

- Quelques packages servent à connecter R avec des éditeurs
  - "literate programming"
    1. On écrit le texte y compris formules en latex & commandes R (graphiques, régressions...)
    2. Si les données changent, ou bien le modèle économétrique, tout est ajusté automatiquement
    3. LaTeXpermet de choisir un format approprié : rapport, article, présentation & d'avoir des math bien écrites
- **SWeave** permet d'envoyer tout le script dans LaTeX
- **knitr** fait la même chose, mais combine d'autres packages et résoud quelques problèmes de SWeave
- **Markdown** convertit du texte simple en html (p.e. SPIP)

Markdown

- Markdown dans R-Studio pour convertir un document texte avec des instructions R-Studio en divers formats
  - HTML, PDF, MS Word, LaTeX, diapo...
- Auto-apprentissage
  - http ://rmarkdown.rstudio.com/lesson-1.html
  - selon votre processeur
    - Je ne poursuis pas
- "Compile notebook" (bouton ds éditeur)
  - permet de tester en html, pdf & word
  - http ://rmarkdown.rstudio.com/r_notebook_format.html