

Programmation dans

M2 CEE

Pr. Philippe Polomé, Université Lumière Lyon 2

2016 – 2017



Outline

Ch1. Bases de R

SWIRL

Gestion de données

R graphique

Fonctionnalités d'édition de document



SWIRL

- ▶ Faites Course 1 : R programming
 - ▶ Lessons 1-9 + 14
 - ▶ 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 AE2016.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 AE2016.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]`
- ▶ `det(A1)` déterminant



Opérations sur matrices

- ▶ `eigen(A1)` eigenvalues
- ▶ `chol(A1)` décomposition de Cholesky
- ▶ `solve(A1)` inverse
- ▶ `A %*% B` produit matriciel
 - ▶ `A*A` produit élément-par-élément
 - ▶ `kroncker(A, B)` produit de Kronecker \otimes
- ▶ `crossprod(A, B)` est un calcul efficient de $A'B$
- ▶ `diag(A1)` extrait la diag
- ▶ `cbind(1, A1)` “combine” une C de 1 et A1 $\begin{matrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix} \rightarrow \begin{matrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{matrix}$
- ▶ `rbind(A1, diag(4, 2))` “empile” A1 et une matrice diag de taille

$\begin{matrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{matrix}$
 2 avec des 4 sur la diag \uparrow
 $\begin{matrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{matrix}$



Outline

Ch1. Bases de R

SWIRL

Gestion de données

R graphique

Fonctionnalités d'édition de document



Dataframe

- ▶ “Frame” = cadre
 - ▶ Dans R, un “Dataframe” est une matrice de données
 - ▶ qui contient de plus des méta-informations
 - ▶ p.e. le type de variable ou le nom des catégories
 - ▶ et peut avoir des cols de natures différentes
- ▶ Un dataframe est un ensemble de vecteurs
 - ▶ de même longueur, mais posiblement de nature \neq
 - ▶ mis l’un à côté de l’autre horizontalement
 - ▶ Donc un rectangle, dans lequel
 - ▶ Les colonnes sont les variables
 - ▶ Les lignes sont les observations
 - ▶ Rem. Une “array” est, dans R, un objet plus général car elle peut avoir + de 2 dimensions



Création

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



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 variable
- ▶ 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

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

- ▶ Le format du fichier texte est
- ▶ 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
 - ▶ D'abord voir s'il est possible d'exporter de ce système en txt ou csv
 - ▶ perte des méta-données
 - ▶ Google
 - ▶ 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 journaux) 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 mtm
 - ▶ Difficulté car $\hat{\sigma}^2$ est perçu comme une matrice 1×1 non conformable avec $(X'X)^{-1}$
 - ▶ Pls solutions p.e. \otimes produit Kronecker
 - ▶ Résultats : 85 et -15 environ ?
- ▶ `detach(Journals)`
 - ▶ “détacher” pour éviter des confusions de nom



Outline

Ch1. Bases de R

SWIRL

Gestion de données

R graphique

Fonctionnalités d'édition de document



Plot

- ▶ D'abord SWIRL
 - ▶ course R-programming, lesson 15 Base graphics
- ▶ Quelques éléments graphiques supplémentaires
 - ▶ Sur base du package **plot**
 - ▶ Packages **lattice** **ggplot2** plus sophistiqués
 - ▶ http://varianceexplained.org/RData/code/code_lesson2/#segment1
 - ▶ R est considéré comme ayant de très bons graphiques
 - ▶ Mais pas très aisés
- ▶ La commande `plot ()`
 - ▶ `plot()` est la commande par défaut pour représenter graphiquement beaucoup d'objet :
 - ▶ dataframes, séries temp, modèles linéaires ajustés
 - ▶ C'est aussi une vieille commande, assez rustique



“Scatterplots” - graphiques de dispersion ou XY

- ▶ Probablement les + communs en stat
- ▶ `attach(Journals)`
 - ▶ `plot(log(subs), log(citeprice))`
 - ▶ `rug(log(subs))`
 - ▶ `rug(log(citeprice), side=2)`
- ▶ `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
 - ▶ "Export" envoie 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 normal 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)`



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, on y reviendra
- ▶ `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



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 un Table de Contingence
 - ▶ `xtabs()` avec une interface formule :
 - ▶ p.e. `xtabs(~ gender + occupation, data = CPS1985)` `data=` optionel
 - ▶ `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” (1^o & 3^o quartiles) et montre la médiane
 - ▶ Hors du box, 2 lignes indiquent les obs. les + petites & + grandes
 - ▶ à moins de $1.5 \times$ 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 : Box plot

- ▶ **Un QQ plot appareille les quantiles de 2 distributions**
 - ▶ Se rappeler que les quantiles sont des quantités
 - ▶ p.e. le 1^{er} 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



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
- ▶ Trouver un autre jeu de données sur lequel vous adapterez l'analyse exploratoire ci-dessus



Outline

Ch1. Bases de R

SWIRL

Gestion de données

R graphique

Fonctionnalités d'édition de document



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. \LaTeX permet 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

