```
1  ##############################################################################
2  #
3  # Strategies de recherche en sciences sociales
4  #
5  # Introduction au logiciel R - Partie I
6  #
7  # Professeur: Dominique Joye
8  # Assistants: Francesco Laganà, Julien Chevillard, Julie Falcon
9  #
10 # Index
11 # Introduction informatique:
12 #
13 # Introduction a R
14 # Operation elementaires avec R
15 # Management des fichiers avec R
16 #
17 #
18 #
19
20 # Ou telecharger R?
21 #
22 # Pour Windows et Mac: http://stat.ethz.ch/CRAN/
23 #
24 # Pour Linux (Ubuntu): dans le terminal tapez: sudo apt-get install r-base
25 #
26 # PRECISATION: R64 bit soit R32bit?
27 #
28 # Vous pouvez installer le R64 sur des ordinateurs dont le processeur
29 # supporte le 64bit.
30 # Autrement vous devriez installer le R32 bit.
31 # Pour le travail qu'on fera dans le seminaire on utilisera R32
32 #
33 # Quand on ecris une synthaxe la symbole "#" represente le "comment": R
34 # interpret tout ce qui suit comme n'etant pas une commande.
35
36 # Seance organisee a l'aide de syntaxes afin que vous puissiez
37 # tout faire ou reproduire vous-memes.
38 # Les syntaxes repondent en effet au principe de transparence d'une
39 # demarche scientifique : tout est evaluable et reproductible
40
41 # Travailler avec des synatxes vous permettra par ailleurs de reproduire
42 # facilement ce que vous avez fait, ou de modifier vos analyses rapidement
43 #
44 # R comme outil de la production de science est public et son code
45 # est libre.
46 # Chacun peut lire et modifier le code pour les analyses.
47 # Il existe differents bulletins:
48 #
49 # http://journal.r-project.org/
50 #
51 # http://www.jstatsoft.org/
52 #
53 # Info sur R
54 #
55 # http://www.r-project.org/
56 #
57 # http://www.statmethods.net/
58 #
59 # Une importante caracteristique de R est que chacun peut contribuer au
60 # projet avec
61 # des extensions, qui permettent d'effectuer differents types d'analyse.
```

```
#
#
# Installer des extensions:
#
# install.packages("nom_de_l'extension", dep=T)

install.packages("rgrs", dep=T)

# Charger les extensions
# (Il faut faire ca avant toute analyse).
# Faire reference a cette page web:
# http://www.statmethods.net/interface/packages.html
#
# commande (exemple): library(extension)

library(foreign)
library(rgrs)
library(Hmisc)

#
# R est une collection d'extensions qui permettent de faire de
# l'analyse statistique.
# Le premier pas consiste a assigner a une variable une certaine valeur:
# Cela est permis par le signe d'assignation "<-"

# Dans le cas d'un scalaire:

x <- 4

# ou dans le cas d'un vecteur:

y.vector <- c(2,6,8,10,20)

# Position d'elements dans un vecteur
# Les paranteses carrees donnent l'element de position 3 dans le vecteur

y.vector[3]



################################################################################
# R comme calculateur
################################################################################

# Vous pouvez utiliser R comme calculateur

4+5

# or

x <- 4

y <- 5

x+y

y.vec2 <- y.vector^2

logyvec <- log(y.vector)

# Combien d'element il y a dans notre vecteur?
```

```
123
124 length(y.vector)
125
126
127 # R peut travailler aussi avec des matrices:
128
129 m1<-matrix(1:12,nrow=3)
130
131 m1
132
133 ##############################################################################
134 # Object et typologies de donnees
135 ##############################################################################
136
137 #  Differement des autres logiciels statistique R peut manager different
138 # types de donnes
139 #  La chose la plus importante est de considerer que tout dans R est
140 # considere comme un objet
141
142 # IMPORTANT: Il existe des caracteres reserves qu'on ne peut pas utiliser
143 # dans R:
144
145 # 1) JAMAIS nommer une variable en commencant par un chiffre
146 # Exemple:
147
148 0a <- 56
149
150 # Mais le contraire (utiliser un chiffre qui ne figure pas en premiere
151 # position dans le nom de la variable ou de l'objet) est possible:
152
153 a0 <- 56
154
155 a0
156
157 # 2) En outre il y a des caracteres reserves:
158
159 # FALSE TRUE Inf NA NaN NULL
160 # break else for function if in next repeat while
161
162 # 3) Il n'est pas conseille de nommer des objets avec le nom d'une fonction:
163 # Exemples: Eviter de nommer vos objet: 1) c (parce que dans le language R
164 # cela signifie "combine")
165 #                                       2) avec des underscores ("_")
166 #                                       3) q (parce que dans le language R
167 #                                            cela signifie "quit")
168
169
170 # Types d'objets dans R:
171
172 # character
173
174 str <- c("Dominique","Julie","Julien","Francesco")
175
176 str
177
178 is.character(str)
179
180 # numeric: les elements appartiennent aux nombres Reels
181 # Ils representent une quantite sur un espace continus: Ex. Revenu
182
183 sqz <- seq(1.575, 5.125, by=0.05)
```

```
184
185 sqz
186
187 is.numeric(sqz)
188
189 # integer
190 # Il representent une quantite sur un espace discret (nombres entiers): Ex.
191 # Household size
192
193 sqint <- seq(1,8, by=1)
194
195 sqint
196
197 sqint <- as.integer(sqint)
198
199 # logical
200 # Il s'agit d'operateurs booleens : ils peuvent assumer les valeurs VRAI ou
201 # FAUX
202
203 log <- c(TRUE, FALSE, TRUE, TRUE, TRUE, FALSE)
204
205 # complex, nombres complexe
206 # Ils appartiennent pas aux nombres reels.
207
208 cmp <- complex(real=1:10, imaginary=-1:9)
209
210 ##### Differents objets peuvent etre regroupes dans une liste:
211
212 elle <- list(STR=str, SQZ = sqz, SQINT= sqint, BOOL=log, CMP=cmp)
213 elle
214
215 ##############################################################################
216 # Managing your analyses:
217 #
218 # Setting your working directory / Assigner votre dossier/emplacement de
219 # travail
220 #
221 # IMPORTANT
222 # Sur ce site vous pourrez trouver beaucoup de documentation:
223 # http://www.statmethods.net/interface/workspace.html
224
225 setwd("/Users/yourlogin/Desktop")
226
227 # knowing your directory:
228
229 getwd()
230
231 # Which files are stored in your working directory? / quels sont
232 # les fichiers figurant dans votre dossier de travail ?
233
234 dir()
235
236 # Il est parfois utile de sauvegarder le contenu de vos analyses
237 # Pour ce faire, il faut creer un fichier log de vos analyses
238 # On fait ca dans R avec la commande sink()
239
240
241 # output directed to output.txt in c:\projects directory.
242 # output overwrites existing file. no output to terminal.
243 # Attention, la commande ci-dessous va effacer le document appele
244 # "Seance6.txt"
```

```
# veillez donc a changer le nom du fichier dans la commande si vous
# souhaitez conserver le fichier original
# sink("Seance6.txt")
# output directed to myfile.txt in cwd. output is appended
# to existing file. output also send to terminal.
# vous pouvez ajouter comme ci-dessous "append"  la commande initiale
# Dans ce cas, le fichier original ne sera pas efface, mais les
# modifications et nouvelles commandes/analyses que vous avez effectuees y
# seront ajoutees

# sink("Seance6.txt", append=TRUE, split=TRUE)

# Knowing the past history of your command:

history(max.show=100)

# R est case sensitive: majuscules et minuscules font la difference:

x <- 46

x

X

#############################################################################
#
# DATA FRAMES:
#
#############################################################################

# Le dataframe est l'objet de R que nous interesse le plus:
# Il est comme une matrice mais, a la difference d'une matrice
# Il accepte aussi des  valeurs non numeriques.
# Par consequent chaque matrice peut etre un dataframe mais
# un dataframe n'est pas forcement une matrice.

# En R on convertit une matrice ou un objet en dataframe avec
# la commande: data.frame

mdat <- data.frame(m1)

is.data.frame(m1)

is.data.frame(mdat)


# Il y a differents moyens de lire les dataframes dans R:
# 1) Manuellement avec la fonction scan()

x<- scan()
1 2 4 5 6 7 9 10 11 23 9 32 4 5 34 23

x

# 2) Lire un fichier deja organise comme data frame:


# Depuis un format proprietaire:
# La livrairie foreign permet de lire des fichier crees avec d'autres
# logiciels
# comme SPSS, SAS, Stata
```

```
# Pour cela vous devriez utiliser l'Extension foreign
#

# Nous allons travailler avec un format de type text file / txt, les donnees
sont separees
# par une virgule (*.csv pour comma separated value :fichier ou les donnees
sont separees par des virgules ).
# Pour l'analyser, vous devez pouvoir lire ce type de fichier dans R
# Pour cela, nous utiliserons la fonction read.table()
#

ess08csv <- read.table("ess08_Methode_31032011.csv", sep=",", header=T)

# Explication de la commande:
#       1) ess08csv represente le nouvel objet ou vos donnees
# sont enregistrees
#       2) entre parentheses, vous avez le fichier .csv ou .dat que vous
# voulez lire ("ess08_Methode_31032011.csv")
#       3) l'option "header" dit a R que la premiere ligne contient les noms
# des variables
#       4) l'option "sep" dit a R que les colonnes (donc les variables) sont
# separees par une virgule.

d <- ess08csv


# Pour savoir quels objets sont presents dans votre workspace il faut
# utiliser la fonction ls()

ls()

############################################################
#
# Trois facons de sauver votre output
# Sauver le workspace

save(d, file="ess08_Methode19Apr2011.rda")

# Sauver l'image de votre travail:
save.image(file = "Methode_Objet31032011.RData")

# Sauver vos données
write.table(d, file="ess08_Methode_31032011.csv", sep=",")

#
# Obtenir de l'aide:
# la commande "help(fonction)" nous permet de demander a R de l'aide sur une
fonction:

help(read.table)

# Ou de facon equivalente:

?read.table
```

```
##############################################################################
#                                                                            #
# Strategies de recherche en sciences sociales                               #
#                                                                            #
# Introduction au logiciel R - Partie II                                     #
#                                                                            #
# Professeur: Dominique Joye                                                 #
# Assistants: Francesco Laganà, Julien Chevillard, Julie Falcon              #
#                                                                            #
# 1) Exploration de la base des données                                      #
# 2) Tableaux des frequences                                                 #
# 3) Manipulation des données: Selectionner des observations                 #
# et des variables et Types de recodage                                      #
# 4) Analyzing relations between two variables:                              #
# 4.1) Variables nominales;                                                  #
# 4.2) Variable interval  ~ variable nominale                                #
# 4.3) Variable interval  ~ variable interval                                #
# 4.4) Variable interval ~ variable nominale + variable interval             #
# 5) Appendix :                                                              #
# 5.1) Elementary database management                                        #
# 5.2) Missing data recoding                                                 #
# 5.3) Selectionner variables par indexation                                 #
##############################################################################


# Loading libraries into R:
library(rgrs)
library(Hmisc)
library(car)
library(psych)

##############################################################################
# 1) Exploration de la base des données
##############################################################################

# Loading data into R
load("ess08_Methode7Apr2011.Rda")

# Exploration du databse:

# Which are the variables you have?

names(d)

# On accede aux descriptives de la variable avec

describe(d$txearn)


# What kind of variables you have? factors? Numeric?
# La fonction str() nous permet de lire savoir le nom des variables et
# leur type: facteur, numerique, integer, etc...

str(d, list.len=200)


# Dans chaque objet il y a different elements. Exemple: dans un data frame il
# y a differents variables, comme le gendre, le revenu, l'occupation etc...
# Pour se referer ŧ une variable presente dans le dataframe d'interet on
# utilise
# le signe "Dollar" ($)
```

```
d$gndr

# si vous n'indiquez pas le dataframe auquel la variable appartient vous
# obtientrez un message d'erreur

gndr

################################################################################
# 2) Tableaux des frequences
#
# Variables facteurs (Nominales ou Orindales)
#
################################################################################

table(d$gndr)

table(d$txearn)

# Pourcentages: la fonction prop.table:

prop.table(table(d$gndr))

# Inspection visuelle des frequences
# Pour des variable nominales on peut utiliser le graphic a bars
# ou ancore le dotplot.

# Valeurs absolus
taprt <- table(d$prtclbch)
dotchart(taprt)

# Pourcentage
dotchart(prop.table(table(d$prtclbch))*100, xlim=c(1,100))

plot(table(d$mar.reg))

barplot(prop.table(table(d$mar.reg))*100, ylim=c(0,100))

# Or la meme chose:

tamar <- prop.table(table(d$mar.reg))*100
barplot(tamar, ylim=c(0,100))

# Numeric variables
min(d$gvjbevn.num, na.rm=T) # La plus petite valeur de la variable
max(d$gvjbevn.num, na.rm=T) # La plus grande valeur de la variable
mean(d$gvjbevn.num, na.rm=T) # La moyenne de la variable
sd(d$gvjbevn.num, na.rm=T) # L'ecart type
var(d$gvjbevn.num, na.rm=T) # La variance
summary(d$gvjbevn.num) # Minimum;  1st Quartile;  Mediane, Moyenne, 3rd
Quartile, Maximum, Valeurs manquantes

# Inspection visuelle des variables interval:

hist(d$gvjbevn.num)
abline(v=mean(d$gvjbevn.num, na.rm=T), col="red", lwd=1)
abline(v=median(d$gvjbevn.num, na.rm=T), col="red", lwd=1)

# Approfondissement: Pour verifier si la distribution de la variable est une
# distribution normale

```

```
# hist(d$gvjbevn.num, probability=T)
# lines(density(d$gvjbevn.num, na.rm=T), lwd=2, col="red")


##############################################################################
# 3) Manipulation des données: Selectionner des observations                 #
#       et des variables et Types de recodage                                #
##############################################################################

# Renommer une variable

d$txe <- d$txearn

# Dans R il y a deux moyens de selectioner observations et variables:
# 1) indexation directe (Barnier "R pour sociologues", p. 43-51)
# [[[[VOIR APPENDIX]]]].

# 2) Fonction subset:
# Le premier argument indique le dataframe duquel il faut "subsetter";
# Le deuxième argument indique la variable et la modalité par laquelle
# il est possible de selectioner les obseravtions (gndr== "Male")
# le troisième argument indique les variables qu'on veut selectionner
# (ne pas indiquer si on veut selectioner toutes les variables)

# Assignation des conditions:
# == egale à
# != different de
# > strictement superieur à
# < strinctement inferieur à
# >= superieur ou egal à
# <= inferieur ou egal à

# Ex. 1 selectioner des reponse des hommes en ce qui concerne le vote,
# la'acitivté principale
# et l'opinion sur la taxation.

dh <- subset(d, gndr== "Male", select=c("ident", "vote","mnactic","txearn"))

# Ex. 2: selectionner la sous-population des femmes.

dfe <- subset(d, gndr== "Female")

# Ex. 3 On peut aussi selectionner observations par deux ou plusieurs
# conditions, par exemple les homme qui peuvent voter (eligible au vote):

dhvo <- subset(d, gndr== "Male" & vote !="Not eligible to vote",
select=c("ident", "vote","prtclbch","mnactic","txearn"))

table(dhvo$vote)

table(dh$vote)

##############################################################################
# Recoding variables:                                                        #
##############################################################################

# Ex. Recoder Main activity

# Creer une nouvelle variable en regroupant les categories

table(d$mnactic)
```

```
d$mac.r <- as.character(d$mnactic)
d$mac.r[d$mnactic=="Paid work" ] <- "Employment"
d$mac.r[d$mnactic=="Unemployed, looking for job" |
d$mnactic=="Unemployed, not looking for job" ] <- "Unemployment"
d$mac.r[d$mnactic=="Education" |
d$mnactic=="Permanently sick or disabled" |
d$mnactic=="Retired" |
d$mnactic=="Housework, looking after children, others" |
d$mnactic=="Other"] <- "Not active"

d$mac.r <- factor(d$mac.r)

# Check:
table( d$mac.r,d$mnactic)


# Découper une variable interval en classes:
d$age.rec <- cut(d$agea, c(15, 24, 34, 44, 54, 64, 100 ), include.lowest=T,
labels=c("15-24", "25-34", "35-44", "45-54","55-64", "65>"))

table(d$age.rec)


# Croiser variables:
# Variable factor (nominale ou ordinale) + Variable factor (nominale ou
# ordinale)
# Ex. gender et employment status:

# Generer une nouvelle variable où memoriser les resultats:
d$gen.act <- NA

# Generer les consitions:
d$gen.act[d$mac.r=="Employment" & d$gndr=="Female" ] <- "Employed Women"
d$gen.act[d$mac.r=="Employment" & d$gndr=="Male" ] <- "Employed Male"

d$gen.act[d$mac.r=="Unemployment" & d$gndr=="Female" ] <- "Unemployment
Women"
d$gen.act[d$mac.r=="Unemployment" & d$gndr=="Male" ] <- "Unemployment Male"

d$gen.act[d$mac.r=="Not active" & d$gndr=="Female" ] <- "Not active Women"
d$gen.act[d$mac.r=="Not active" & d$gndr=="Male" ] <- "Not active Male"

d$gen.act <- factor(d$gen.act)


# Variable categorielle + variable interval
# Ex. Age et gender:
d$age.gndr <- NA

# Generer les consitions:
d$age.gndr[d$agea<=45 & d$gndr=="Female" ] <- "Women <= 45"
d$age.gndr[d$agea>45 & d$gndr=="Female" ] <- "Women > 45"

d$age.gndr[d$agea<=45 & d$gndr=="Male" ] <- "Male <= 45"
d$age.gndr[d$agea>45 & d$gndr=="Male" ] <- "Male > 45"

table(d$age.gndr)

# Transformation des variables:
# Ex. cstandardiser les variables
d$z.gvjbevn.num <- (d$gvjbevn.num-mean(d$gvjbevn.num,
```

```
na.rm=T))/sd(d$gvjbevn.num, nam.rm=T)

# De cette façon la nouvelle variable est calculé en unités d'ecart type
# (ecart type=1) et centrée sur 0 (mean=0)

mean(d$z.gvjbevn.num, na.rm=T)

sd(d$z.gvjbevn.num, na.rm=T)

# Variable interval + variable interval: Computing scores

# 1. Extraire les variables dont on veut estimer le score:
pro <- d[c("gvjbevn.num", "gvhlthc.num", "gvslvol.num" , "gvslvue.num" ,
"gvcldcr.num" ,"gvpdlwk.num")]

# 2. Calculer la moyenne:
d$state.resp <- rowMeans(pro, na.rm=T)

#############################################################################
# Reliability analysis:                                                     #
# When we want to compute the score for a given set of variables            #
# it is possible to assess the reliability of the scale that                #
# we constructed                                                            #
# The most used test is the Cronach's Alpha.                                #
#                                                                           #
# 1)  Le Cronbach's Alphs peut assumer des valeurs compris parmi 0 et 1 et  #
# nous donne des indications sul le degree de inter-correlation des         #
# differentes items qu'on veut examiner.                                    #
#                                                                            #
# 2) Si les items sont très correlés parmi eux (consistence interne         #
# maximale), il est possible de conclure que chaque item donne une          #
# contribution réelle à la construction de la mesure et que,                #
# dans l'ensemble tous les items se rèferent au meme concept.               #
#                                                                           #
# 3) Si les items ne sont pas très correlés parmi eux, ça signifie que ceux#
# là ne constituent pas une mesure satisfaisante du meme construct.         #
#                                                                           #
# Exemple: si une echelle sur l'anxieté alpha a une valeur de 0.80          #
# cela signifie que l'80% du score obtenu par le sujet est du à l'anxieté   #
# Si la consistence interne de l'echelle est de 0.40, cela signifie que le  #
# 40%  de l'echelle mesure l'anxieté et le restant 60% mesure autre chose.  #
#                                                                           #
# Valeurs conventionels du alpha de cronbach.                               #
# 0.60 inacceptable;                                                        #
# 0.60-0.65 pas desirable;                                                  #
# 0.65-0.70 Acceptable;                                                     #
# 0.70-0.80 Bon;                                                            #
# 0.80 > Optimale                                                           #
#                                                                            #
# Dans R on peut calculer le alpha de Cronbach par la fonction alpha        #
# contenue dans l'extension psych                                           #

alpha(pro)
```

```
################################################################################
# 4. Analyzing relations between two variables:
#
# De l'analyse d'association aux modèles de regression:
#
# Different types d'analyse en fonction des types de variables à analyser:
#
# (1) Variables categorielles
# Tableaux croisées + Test du Chi carré + V de Cramer
#
# (2) Variables variables interval +  variable categorielle:
# T test (deux groupes) ou Analysis de la variance (quand il s'agit de
# plusieurs groupes)
#
# (3) Variable interval + variable interval
# Analysis de la correlation ou regression
#
################################################################################

################################################################################
#
# 4.1) Deux variables categorielles (facteurs)
# Tableaux croisées + Test du Chi carré + V de Cramer
#
#
# Valeurs Absolus:

tab <- table(d$txearn, d$gndr)

tab

# Pourcentages ligne:
prop.table(tab, 1)

# Pourcentages colonne:
prop.table(tab, 2)

# Pourcentages case:
prop.table(tab)

# Quand'il s'agit de calculer les pourcentages pour ligne ou colonne,
# jamais oublier la regle de Galtung: il faut percentualiser en relation
# au valeurs de celle qui est considerée comme variable independente
# dans le modèle de base. (Galtung, 1970)


# Tester l'association parmi deux variables categorielles:

# Comment tester l'hypothese d'association parmi deux
# variables categorielles?

# Association summary(table) -> chi square
# Est-ce que la relation parmi genre et taxation est significative?

# On repond à cette question par le test du chi carré:
# Le test du chi carré teste l'hypothese nulle d'absence de relation
# parmi deux variables.
# Le chi carré il est estimé par la confrontation parmi le tableau
# crée et un tableau hypothetique qui represente la distribution
# conjointe des deux variables en cas d'independence.

```

```
# Les deux commandes donnent des coefficients d'associations parmi deux
# variables (Chi square et V de Cramer);

summary(tab)

# Le Chi square nous dit qu'il y a de l'association parmi deux
# variable et que l'association n'est pas random.
#
# On peut se demander de la force de l'association:
# pour repondre à cette question on utilise le
# V de cramer. Il est calculé comme suit:
# V = SQRT(chi2/(n(k - 1)))

# Où is chi2 est le chi-square and k is the number of rows or columns in the
# table.
# Cramer's V varies between 0 and 1. Close to 0 it shows little association
# between variables. Close to 1, it
# indicates a strong association.
# Where the table is 2 x 2, use Phi.

assocstats(tab)

# Une autre façon d'obtenir les statistiques du chi2 est par la
# commande chisq.test

chisq.test(tab)

# On peut aussi avoir un tableau avec les
# expected frequences en cas d'independence (sous l'hypothese nulle):

chisq.test(tab)$expected

# Et les residuals '(observed - expected)/sqrt(expected)'.:

chisq.test(tab)$residuals

# NOTE about the chi2.
# 1) It is valid only in case of random samples
# 2) Sample size: If a chi square test is conducted on a sample with a
# smaller size,
# then the chi square test will yield # an inaccurate inference. The
# researcher, by using chi square test on small samples,
# might end up committing a Type II  # error.
# 3) Expected Cell Count – Adequate expected cell counts. Some require 5 or
# more, and others require 10 or more.
# A common rule is 5 or more in all cells of a 2-by-2 table, and 5 or more in
# 80% of cells in larger tables,
# but no cells with zero expected count. When this assumption is not met,
# Yates' correction is applied.
# 4) Independence – The observations are always assumed to be independent of
# each other.
# This means chi-square cannot be used to test correlated data (like: matched
# pairs, panel data). In those cases you
# might want to turn to McNemar's test.
#
# In case you have small sample sizes use fisher exact test:

fisher.test(tab)
```

```
##############################################################################
#
# 4.2 Variable interval + variable categorielle:
# ANALYSE DE LA VARIANCE:
#
##############################################################################

# L'analyse de la variance est utilisée pour etudier la relation parmi
# une variable (dependante) interval + une variable (independente)
# categorielle
#
# L'Anova permets de comparer la moyenne d'une variable interval en fonction
# des categories d'une variable nominale ou ordinale
# Tapply estime les moyennes par differents groupes:
# Est-ce qu'il y a des differences parmi employés, non-activs et chomeurs
# dans le soutien des actions de redistribution gouvernamentales
# vers les chomeurs?

tapply(d$gvslvue, d$mac.r, mean, na.rm=T)

# To compare two groups you can use t.test
t.test(d$gvslvue ~ d$gndr)

# NOTE: note that the t.test requires normality in the distribution of
# the two groups and equal variance (faire reference au texte de Barnier pour
# approfondissements.
# Les deux tests fondamentaux sont:
# 1) shapiro-wilk normalty test (fonction: shapiro.test())
# 2) Egalité des variances:(fonction var.test())
# Au cas ou ces deux assumptions ne sont par rencontrées
# il faut utiliser des test non-parametriques:
# wilcox.test()


# To compare the significativity in differences of more than two groups: uses
# lm() function and anova() functions.
# Such functoins allows for an extension of a two-sample t test, testing the
# equality of means of more than two groups.

# Ex. Est-ce que les differences parmi employés, non-activs et chomeurs
# dans le soutien des actions de redistribution gouvernamentales
# vers les chomeurs sont significatives?

mod1 <- lm(d$gvslvue ~ d$mac.r)

summary(mod1)

anova(mod1)

# Inspection visuelle de variables interval + categorielle:
boxplot(d$gvjbevn.num~d$mac.r)

# Dans l'Anova le F represente l'equivalent du t dans le t test.
# Lien mathemiatique: F=t^2

# NOTE:
# Il y a des assumptions qui doivent etre satisfaites pour l'ANOVA:
# Notamment:
# 1) Inependence des valueurs observés.
# 2) Normalité de la distribution de notre variable dependente:
# 3) Homogeneité des variances:
```

```
489
490 # La deuxième assumption peut etre evalué de deux façons:
491 # Le test de normalité di Kolmogorov-Sirnov:
492
493 ks.test(d$gvslvue.num, pnorm, mean(d$gvslvue.num, na.rm=T),
494 sd(d$gvslvue.num,na.rm=T))
495
496 # De l'output on doit lire le p-value. Cela doit etre egale à 0.05 ou plus
497 # grand: l'hypothese nulle dit que la distribution est normale.
498 # le ks.test il est sensible aux valeur dupliqués.
499 # Une alternative est le shapiro.test
500
501 shapiro.test(d$gvslvue.num)
502
503 # Egalement au ks.test on doit regarder le p-value. L'ìhypothese nulle
504 # dit que notre variable dependente estdistribué selon une distribution
505 # normale.
506
507 # En ce qui concerne l'homogeneité des variances
508
509 bartlett.test(d$gvslvue ~ d$mac.r)
510
511 # De l'output on doit lire le p-value. Il doit etre majeure ou egale
512 # à 0.05. L'hypothese nulle soutient que les variances sont homogenes.
513 # Au cas où cette assumption ne soit pas respectée, il faut calculer l'Anova
514 # avec la correction de Welch.
515
516 ##############################################################################
517 #
518 # 4.3) Mesures d'association entre deux variables interval:
519 # Coefficient de CORRELATION et Modele de REGRESSION
520 #
521 ##############################################################################
522
523 # Quand les deux variables sont interval on peut utiliser
524 # analyser la correlation et la regression.
525 # Le coefficient de correlation de Pearson (Pearson's
526 # product-moment correlation)
527 # il est calculé comme le rapport parmi la covariance parmi deux variables (y
528 # and x) et  le product de leurs ecart types:
529 #
530 # rho(x,y) = cov(x,y)/(sigma(x)*sigma(y))
531 #
532 # où cov(x,y) = sum((x-M(x))*(y-M(y))/(n-1)
533
534 cor(d$gvjbevn.num, d$iseiego.num, use="complete.obs")
535
536 # Le coefficient de correlation est egale à 0 en cas d'absence de
537 # relation
538 # Il est egale à 1 en cas de très forte association positive
539 # Il est egale à -1 en cas de très forte association negative
540
541 # On peut tester l'hypothese nulle d'absence de relation parmi les deux
542 # varibles
543
544 cor.test(d$gvjbevn.num, d$iseiego.num, use="complete.obs")
545
546 # ou encore:
547
548 cor(d$gvjbevn.num, d$gvhlthc.num, use="complete.obs")
549
```

```
550 cor.test(d$gvjbevn.num, d$gvhlthc.num, use="complete.obs")
551
552 # NOTE:
553 # Si vous voulez analyser la correlation parmi differentes variables:
554 # Il faut regrouper ces variables dans un data frame comme on
555 # avait fait avant (ligne 256-258).
556
557 cor(pro, use="complete.obs")
558
559 # Si on trouve un coefficient de correlation elevé
560 # on peut se poser la question de la meilleure
561 # ligne qui interpole nos données.
562 # Cela signifie de trouver la fonction qui lie
563 # deux phenomenes à travers la formulation
564 # d'un modèle du type: f(x)= a + b(x)
565
566 # On repond à cette question par l'analyse de la ligne de
567 # regression:
568
569 # Quel est l'impact du Socioeconomic status sur le soutien au gouvernement?
570
571 mod1 <-  lm(d$gvjbevn.num ~ d$iseiego.num)
572 mod1
573
574 # Le coefficient de regression represente le coefficient
575 # angular de la ligne. Il nous dit le rapport
576 # parmi variation de notre x et variation de y.
577 # Quel est la variation de y suite à variations de
578 # notre x.
579 # En termes statistiques il represent la partie de
580 # la covariance parmi x et y qui è explique par la
581 # variance de x.
582 #
583 # beta = cov(x,y)/var(x)
584
585 beta <- cov(d$gvjbevn.num, d$iseiego.num,
586       use="complete.obs")/var(d$iseiego.num, na.rm=T)
587
588 # Visualiser le plot et la ligne de regression
589 plot(jitter(d$iseiego.num, factor=3), jitter(d$gvjbevn.num, factor=2))
590 abline(lm(d$gvjbevn.num ~ d$iseiego.num), col="red")
591
592 # Autre exemple: Correlation parmi Index of Socio-Economic Status de
593 # l'intervieué et celui de son père.
594 plot( jitter(d$iseifa.num, factor=2), jitter(d$iseiego.num, factor=3))
595 abline(lm(d$iseifa.num ~ d$iseiego.num), col="red")
596
597
598 ##############################################################################
599 #
600 # Regression multiple:
601 # Analyses de regression (AnCoVa) :
602 #
603 ##############################################################################
604
605 # Souvent on veut controler si une relation
606 # resiste à des controles ulterieures
607 # avec d'autres variables:
608 #
609 # Pour cette raison on utilise des technique de regression multiple
610 # ou on controle notre relation parmi x et y par l'effet d'autre variables:
```

```
611
612 # Reprenons l'exemple du soutien pour les chomeurs (Job for everyone,
613 # governments' responsibility) et soupposons que la relation depend de:
614
615 # Position sur l'echelle gauche droite
616 # Statut occupationel
617 # Socio-economic status
618 # Variables de control:
619 #     Age (continue); Gender.
620
621 # Avant d'estimer une regression:
622
623 # Selectionner les variables qui font partie de votre modele:
624 dm <- d[c("state.resp","gvjbevn.num", "iseiego.num", "agea.num" ,
625 "lrscale.num", "gndr", "mac.r", "age.rec")]
626
627 # 1) Analyse monovariée de toutes les variables pour detecter outliers...
628 [commande: table(), plot(), summary()]
629
630 # 2) Transformation eventuelle des variables (par example centrer par
631 # rapport)
632 # Centrer l'age par rapport à la moyenne (generalement pour toutes les
633 # variables dont le champ de variation
634 # ne comprend pas le zero – cela sert pour
635 # l'interpretation de l'intercepte.
636
637 dm$Cagea <- dm$agea-mean(dm$agea, na.rm=T)
638 dm$Cisei <- dm$iseiego.num-mean(dm$iseiego.num, na.rm=T)
639
640 # 3) Exploration des relations parmi variables [commande: corr()]
641
642 cor(dm[,1:6], use="complete.obs")
643
644 # 4) Creation des contrasts:
645 # Quand dans la regression il y a des variables quantitatives et qualitatives
646 # il faut s'assurer que ces dernières soient traitées comme factors:
647 #
648
649 is.factor(dm$mac.r)
650 is.factor(dm$gndr)
651
652
653 # Visualiser les contrasts:
654 contrasts(dm$mac.r)
655
656 # On peut changer la baseline des contrasts avec l'option suivante:
657
658 contrasts(dm$mac.r) <- contr.treatment(levels(dm$mac.r), base=1)
659
660 mod1 <-  lm(d$gvjbevn.num ~ d$iseiego.num + agea.num, data=dm)
661
662 mod1 <-  lm(gvjbevn.num ~ Cisei+ Cagea + mac.r+ lrscale.num + gndr, data=dm)
663
664 # Est-ce que l'effet de l'age est lineaire?
665 # On reponds à cette question defragmentant l'age (on utilise age.rec)
666
667 # Creer les contrasts:
668 contrasts(dm$age.rec)
669
670 mod1 <-  lm(gvjbevn.num ~ iseiego.num + age.rec + mac.r+ lrscale.num + gndr,
671 data=dm)
```

```
672
673 summary(mod1)
674
675 # La regression lineaire est seulement un modele de regression qui fait
676 # partie d'une plus grande famille des Generalized Linear Models...
677 # ... mais ça n'est pas objet de notre cours...
678 #
679 # ... FIN
680
681 # A savoir:
682
683 # Trois façons de sauver votre output:
684
685 # Sauver le workspace
686 save(d, file="ess08_Methode19Apr2011.rda")
687
688 # Sauver l'image de votre travail:
689 save.image(file = "Methode_Objet31032011.RData")
690
691 # Sauver vos données
692 write.table(d, file="ess08_Methode_31032011.csv", sep=",")
693
694
695
696 ##############################################################################
697 # APPENDIX
698 # 1) ELEMENTARY DATABASE MANAGEMENT
699 # 2) MISSING DATA RECODING:
700 # 3) SELECTIONNER VARIABLES by indexing
701 ##############################################################################
702
703 ##############################################################################
704 #
705 # 1) ELEMENTARY DATABASE MANAGEMENT
706 #
707
708 # How many cases are there in your database?
709 # La commande dim() nous permet de savoir combien de cases il y
710 # a dans notre matrice des donnÈes.
711
712 dim(d)
713
714 # Finally we want to know which are the attributes of our database more
715 # in detail.
716 # The command attributes will give us a list of informations about our
717 # dataset:
718 #     1) Names of variables;
719 #     2) Class of the object;
720 #     3) The names of the row.
721
722 attributes(d)
723
724 # Differement de Spss R peut manager different type d'objets.
725 # Ceux-là il sont storÈ dans le workspace.
726 # Le workspace est comme bureau de travail, la table sur la quelle vous
727 # mettez ce que vous
728 # produisez dans une session de travail avec R.
729 #
730 # Pour savoir quels object sont present dans votre workspace vous utilisez la
731 # fonction ls()
732
```

```
ls()

# On peut definir et acceder aux Variable labels par les fonctions label et
# describe.
# Celle ci donne une liste des variables avec des descriptives:
# la commande se trouve dans la livrairie Hmisc, pour autant elle ne marche
# pas si vous ne lodez sur R cette livrairie.

label(d$txearn)<-"Taxation for higher versus lower earners"

# On accede aux descriptives de la variable avec

describe(d$txearn)


################################################################################
#
# 2) MISSING DATA RECODING:
# Each data contains values that are missing or not applicable.
# Il est donc necessaire de dire à R quels valeurs correspondent à missing.
# D'abitude c'est des labels comme "Don't know", "Refusal", "No answer", ou
# encore 88 ou 99999
# La synthax suivante transforme les missing
# See the file Dataset setting for the complete synthax:

d$mnactic[d$mnactic== 88] <- NA

# Or si vous voulez recoder toutes les valeurs d'un dataframe qui
# sont egales à "Don't know" etc... comme missing, il faut user cette
# code:

d[d=="Don't know" | d == "No answer" | d== "Refusal" ] = NA

# Comme on l'a vu dans la séance 4 il y a different type de variables qui
# R peut manager:
# Variables Qualitatives:
# 1) Echelle nominale:

d$gndr <- factor(d$gndr)
str(d$gndr)

# 2) Echelle ordinale

d$edulvl <- ordered(d$edulvl, levels=c("Not Compl.Prim. Educ",
"Primary or first stage of basic",
"Lower secondary or second stage of basic",
"Upper secondary",
"Post secondary, non-tertiary",
"First stage of tertiary",
"Second stage of tertiary"))

str(d$edulvl)

# 3) Interval:
# Discrete:

str(d$happy.num)

# Continues:

str(d$dweight)
```

```
794
795 # Manipulation sur une variable:
796 # Acceder aux levels d'une variable:
797
798 levels(d$edulvl)
799
800 # De fois les labels des categories d'une variable sont trop longues (surtout
801 quand on fait des graphs).
802 # Par example
803
804 table(d$gndr, d$txearn)
805
806 # On peut donner à la variable des lables plus courts des façon a faciliter
807 le plotting
808 # Procedure:
809 # 1. Creer une nouvelle variable:
810 d$txe <- d$txearn
811 levels(d$txe) <- c("HEHSET","NoneOfThese","SAMT","SSET")
812 table(d$gndr, d$txe)
813
814
815 ##############################################################################
816 #
817 # 3)SELECTIONNER VARIABLES by indexing
818 # Un autre moyen de selectionner variables et observations en R est
819 # par indexation. Dans ce cas on considère le dataframe comme une matrice:
820 # (voir seance 6) de laquelle on slectionne observations et variables.
821
822 # Ex. On commence de la matrice X
823
824 X = matrix(1:100, ncol = 5)
825
826 X
827
828 # De cette matrice on peut extraire lignes, coulonnes ou cases:
829
830 # Ex. Deuxième coulonne:
831 X[,2]
832
833 # Ex. CInquième ligne
834 X[5,]
835
836 # Ex. Element en 5ème ligne et 2ème coulonne
837 X[5,2]
838
839 # Quand on travaill  avec en dataframe on peut faire la meme chose:
840
841 # Ex. 1 selectioner des reponse en ce qui concerne le vote, la'acitivté
842 principale
843 # et l'opinion sur la taxation.
844
845 dh <- d[,c("ident", "vote","mnactic","txearn")]
846
847 # Ex. 2: selectionner la sous-population des femmes.
848
849 dfe <- d[d$gndr=="Female",]
850
851 # Ex. 3 On peut aussi selectionner des souspopulations:
852
853 dhvo <- d[d$gndr== "Male" & d$vote !="Not eligible to vote", c("ident",
854 "vote","prtclbch","mnactic","txearn")]
```

```
855
856  table(dhvo$vote)
857
858  table(dh$vote)
```