Statistical programming with R Exercises

Emmanuel Kemel (HEC Paris, CNRS)

Outline

1. Manipulating vectors
2. Manipulating data frames
3. Plots
4. Functions and loops

Outline

1. Manipulating vectors

2. Manipulating data frames

3. Plots

4. Functions and loops

Exercise 1: running calculations

- Generate $n = 100$ draws from a standard normal distribution (with function *rnorm*) and store them in a variable x
- Compute the mean, median and standard deviation of x
- Compute the 95%CI of the mean of x using the formula: $[\bar{x} - 1.96\frac{\bar{\sigma_x}}{\sqrt{n}}, \bar{x} + 1.96\frac{\bar{\sigma_x}}{\sqrt{n}}]$
- Store this result in a variable *ci* that contains a vector of length 2 with each value corresponding to a bound of the confidence interval.
- Run your code several times, till you observe that the confidence interval does not contain the population mean.

Exercise 1: correction

```r
x=rnorm(100)
mean(x)
sd(x)
n=length(x)
ci=c(mean(x)-1.96*sd(x)/sqrt(n),mean(x)+1.96*sd(x)/sqrt(n))
# or equivalently
ci=mean(x)+1.96*sd(x)/sqrt(n)*c(-1,1)
```

Exercise 2: manipulating NAs

- Run the following code

```
y=sample(c(1:10,NA),100, replace=T)
```

- Compute the mean of y
- Compute the mean of y when
  - NA are omitted
  - NAs are placed by 0s
  - NAs are replaced by the mean of y

Exercise 2: correction

```
y=sample(c(1:10,NA),100, replace=T)
mean(y)
mean(y,na.rm=T) # or equivalently: mean(y[!is.na(y)])
# replacing NAs by 0
yNA0=y
yNA0[is.na(yNA0)]=0
mean(yNA0)
# replacing NAs by the mean of y without NAs
yNAM=y
yNAM[is.na(yNAM)]=mean(y,na.rm=T)
mean(yNAM)
```

Exercise 3: subsetting vectors

- Run the following code to simulate the gender, type and scores of Master students

```
gender=sample(c("male","female"),100, replace=T)
```

```
type=sample(c("MS","GE"),100, replace=T)
```

```
score=rnorm(100, 14,3)
```

- Compute the mean score of males and the mean score of females
- Is the difference statistically significant (use function *t.test*)
- Compute the mean score of GE males and compare it to the mean score of MS females
- Create a variable "grade" that converts scores into letters
  A: score≥ 18, B: $18 >$score≥ 16, C: $16 >$score≥ 14, D: $14 >$score≥ 12, E: $12 >$score≥ 10, F: $10 >$score
- What is the distribution of letters among students?

Exercise 3: correction

```
mean(score[gender=="male"])
mean(score[gender=="female"])
t.test(score[gender=="male"],score[gender=="female"])
mean(score[gender=="male" & type=="GE"])
mean(score[gender=="female" & type=="MS"])
```

Exercise 3: correction (continued)

```
grade=rep("A",length(score))
grade[score<18]="B"
grade[score<16]="C"
grade[score<14]="D"
grade[score<12]="E"
grade[score<10]="F"
table(grade)
```

Outline

1. Manipulating vectors
2. Manipulating data frames
3. Plots
4. Functions and loops

Exercise 4: loading a data set

- Load the data set cps08.csv
- Look at the dimensions of the data
- Look at the names of the variables (columns) in the data set
- Run a summary of the data

Exercise 4: correction

```r
setwd("~/Dropbox/MFE_Econometrics/data")
data=read.table("cps08.csv",header=T, sep=";",dec=",")
dim(data)
names(data)
summary(data)
```

Exercise 5: calculations on a data set

- compute the mean “average hourly earning” (ahe)
  - for males,
  - males without a bachelor

Exercise 5: correction

```r
mean(data[data$female==0,"ahe"])
# or equivalently
mean(data[data$female==0,]$ahe)

mean(data[data$female==0 & data$bachelor==0,"ahe"])
# or equivalently
mean(data[data$female==0 & data$bachelor==0,]$ahe)
```

Exercise 6: calculations on a data set

- compute the mean “average hourly earning” (ahe) conditionaly on
    - gender (variable “female”)
    - education (variable “bachelor”)

  with the function aggregate

Exercise 6: correction

```r
aggregate(ahe~female+bachelor,data,mean)
```

Outline

1. Manipulating vectors

2. Manipulating data frames

3. Plots

4. Functions and loops

Exercise 7: scatter plots

- load the data set “profsalary.txt”
- scatter plot the relationship between salary and experience
- add a title to this plot
- save the plot in a pdf file on your desktop

Exercise 7: correction

```
setwd("~/Dropbox/MFE_Econometrics/data")
data=read.table("profsalary.txt",header=T, sep="")
dim(data)
plot(Salary~Experience,data)
plot(Salary~Experience,data,
main="Evolution of salary with Experience",
xlab="Years of experience")

###
setwd("~/Desktop")
pdf("myplot.pdf")
plot(Salary~Experience,data,
main="Evolution of salary with experience",
xlab="Years of experience")
dev.off()
```

Outline

1. Manipulating vectors

2. Manipulating data frames

3. Plots

4. Functions and loops

Exercise 8: functions

- create a function *ci* that takes a vector *x* in argument and returns a vector containing the bounds of the 95CI of its mean
- create a function *fun* that takes a vector *x* in argument and returns a list containing
  - the mean of x under the name "mean"
  - the bounds of the 95CI of its mean, under the name "ci"

Exercise 8: correction

```r
ci=function(x){
n=length(x)
result=mean(x)+1.96*sd(x)/sqrt(n)*c(-1,1)
return(result)
}
########
fun=function(x){
n=length(x)
result=list(mean=mean(x), ci=mean(x)+1.96*sd(x)/sqrt(n)*c(-1,1))
return(result)
}
```

Exercise 9: loops

- Load the data set cps08.csv
- use a “for loop” to compute the mean “average hourly salary” for each value of age in the data set
- compare the result to the result of aggregate

Exercise 9: correction

```
setwd("~/Dropbox/MFE_Econometrics/data")
data=read.table("cps08.csv",header=T, sep=";",dec=",")
########
for (i in unique(data$age)){print(mean(data[data$age==i,"ahe"]))}

# same storing the results in a matrix
results=c()
for (i in unique(data$age)){
results=rbind(results, c(i,mean(data[data$age==i,"ahe"])))
}
results

######
aggregate(ahe~age,data,mean)
```