Lab 02

Jaime Montana

9/9/2021

Reminder of last session

- Why R?
- Create an `*.rmd` and `*.r` files.
- We learn the basic **Rmarkdown notation**
- R as calculator
- R to store variables
- Functions
- Working directory
- How to open a data set ... (We will review this now!)

Working directory and loading data

Recall: The *working directory* directs to a path in your computer/infrastructure.

```
getwd()
```

```
## [1] "/home/jaime/Dropbox/Catolica - Postdoc/Courses/BRM/
```

```
#setwd("C:/path/to/files/") # change the path to wd

#load("ceosal2.RData")   # If you have saved in wd
#load("C:/path/to/ceosal2.RData") # else
```

Libraries in R

- R is open source
- There are plenty of developed routines, snippets, statistical packages and methods developed by users that are available for other users ($>$ 18133 packages in CRAN). According to a recent udacity blog, the most usefull libraries are:
  - `ggplot2` for data visualization
  - `dplyr` and `data.table` for
  - `caret` for machine learning
  - `readr` for data import
  - `stargazer` for nice table formatting

Libraries in R (II)

There is a simple procedure to use a library:

1. You first need to install the library.
   - Use the GUI of RStudio.
   - - Use the command line of your script * `install.packages("NAME")` **(preferred)**
2. In your script call the library you just installed: `library(NAME)`.

For example, for the `data.table` package.

```r
install.packages("data.table")
library(data.table)
```

Libraries in R (exercise)

Install the following packages and the use the function libraries to activate them in your session.

`data.table, stargazer, ggplot2, ggthemes, doBy`

How to open a dataset (RData)

There is different types of data set that R can handle: It could be an *R dataset*, a text file, a CSV, a SAS file, a STATA file . . . R can handle all of them!

For example to load an environment with Rdata, we can use the function `load("path_to_your_RDATA_file")`

```
load("ceosal2.RData")

ls()
```

```
## [1] "data" "desc" "self"
```

The function `ls()` list all the objects stored in the actual environment.

How to open a dataset (csv)

There are several options to open a csv. We are going to use two ways:

- Using the GUI (from the package `readr`
- Using the function `fread()` from the package `data.table`, in the r-script.

```
data_csv <- read_csv("ceosal2.csv")
```

```
dt.ceo.salaries <- fread("ceosal2.csv")
```

After importing the data. . .

Generally after you import the data you want to **see it**.

- What are the variables?
- Are numeric?
- Are continuous variables?
- Are categories?

To check what are the variables in the data, you can search the variable names using:

```
names(data)
```

```
##  [1] "salary"   "age"      "college"  "grad"     "comten
##  [7] "sales"    "profits"  "mktval"   "lsalary"  "lsales
## [13] "comtensq" "ceotensq" "profmarg"
```

Explore the data (I)

The function `head(your_object)` return the first part of your data. If you specify also `n =`, you can set the set of observations to print.

```
head(data, n = 4)
```

```
##   salary age college grad comten ceoten sales profits mk
## 1   1161  49       1    1      9      2  6200     966  2
## 2    600  43       1    1     10     10   283      48
## 3    379  51       1    1      9      3   169      40
## 4    651  55       1    0     22     22  1100     -54
##     lmktval comtensq ceotensq  profmarg
## 1 10.051908       81        4 15.580646
## 2  7.003066      100      100 16.961130
## 3  7.003066       81        9 23.668638
## 4  6.907755      484      484 -4.909091
```

Explore the data (II)

Alternatively you can see the last observation using the function `tail(your_object)`.

```r
tail(data, n = 4)
```

```
##     salary age college grad comten ceoten sales profits
## 174    185  58       1    0     39      1   766      49
## 175    387  71       1    1     32     13   432      28
## 176   2220  63       1    1     18     18   277     -80
## 177    445  69       1    0     23      0   249      31
##       lsales  lmktval comtensq ceotensq   profmarg
## 174 6.641182 6.327937     1521        1   6.396867
## 175 6.068426 6.167517     1024      169   6.481482
## 176 5.624018 6.291569      324      324 -28.880867
## 177 5.517453 6.719013      529        0  12.449800
```

You can **View** all the data in the window by double clicking the element in the environment or by typing in the console the function `View()`

Rename, and calculate data size

Sometimes you want to change a variables name. Use the function:

```
setnames(dt.name, "OLDNAME", "NEW_NAME")
```

For example:

```r
setnames(dt.ceo.salaries,"lsales", "logsales")
```

To get the number of registries/records in your data use the `nrow(dt.name)`

```r
nrow(dt.ceo.salaries)
```

```
## [1] 177
```

```r
dt.ceo.salaries[, .N]
```

```
## [1] 177
```

To get the number of variables in your data use the `ncol(dt.name)`

Select and subset data

For reference consult the `data.table` chatsheet.

You can subset data indicating the row numbers you wish to select. For example if you wish to select all the colums from the first to the eight row, the syntax is:

```
dt.ceo.salaries[1:8,]
```

Only the first:

```
dt.ceo.salaries[1,]
```

Based on a condition on column values:

```
dt.ceo.salaries[age <= 45,]
```

Select and subset data (II)

You can also compose a condition with a more complex query. Imagine we would like:

- age less than 45
- are graduate students
- salary is strictly larger than 800

Select and subset data (II)

You can also compose a condition with a more complex query. Imagine we would like:

- age less than 45
- are graduate students
- salary is strictly larger than 800

```
dt.ceo.salaries[age <= 45 & grad == 1 & salary > 800,]
```

```
##    salary age college grad comten ceoten sales profits r
## 1:   1630  39       1    1      8      8   227      27
## 2:    873  41       1    1      2      2   149      21
##     lmktval comtensq ceotensq profmarg
## 1: 6.711740       64       64 11.89427
## 2: 6.340359        4        4 14.09396
```

Transforming variables

To add a new variable to the data.table we use the symbols ‘**:=**’ . We can either create new variables that are transformations of existing variables such as:

```
dt.ceo.salaries[,log_salary := log(salary)]
dt.ceo.salaries[,age_sq     := age^2]
```

To remove a variable use:

```
dt.ceo.salaries[,log_salary := NULL]
```

Descriptive statistics

To print a table with *'basic'* descriptive statistics we use the package `stargazer`. It prints for all **numerical** variables the values of the number of complete observations, average, standard deviation, minimum, maximum and percentile (25 and 75)

```
stargazer(dt.ceo.salaries, type = "text")
```

Descriptive stats (II)

When a variable is not numeric the summary statistics can't be calculated. With `data.table` you can also calculate a conditional statistic:

```
dt.ceo.salaries[grad==1, mean(salary)]
```

```
## [1] 864.2128
```

```
dt.ceo.salaries[, mean(salary), by = grad]
```

```
##    grad       V1
## 1:    1 864.2128
## 2:    0 867.7349
```

Counting with `data.table`

If we need to count observations, we can use the **‘.N’** operator.

- Count the number of observations
- count the number of observations of individuals that are graduated.

```r
dt.ceo.salaries[grad==1, mean(salary)]
```

```
## [1] 864.2128
```

```r
dt.ceo.salaries[, mean(salary), by = grad]
```

```
##    grad       V1
## 1:    1 864.2128
## 2:    0 867.7349
```

Descriptive stats (III)

How many CEOs have/don't have a graduate degree?

```
dt.ceo.salaries[, table(grad)]
```

```
## grad
##  0  1
## 83 94
```

```
# or
table(dt.ceo.salaries[, grad])
```

```
##
##  0  1
## 83 94
```

How many CEOs have/don't have a college degree?

Table on Multiple conditions

```
table(dt.ceo.salaries[, college],dt.ceo.salaries[, grad])
```

```
##
##      0  1
##   0  5  0
##   1 78 94
```

```
dt.ceo.salaries[, list(n_ceo = .N), by = list(college, grad
```

```
##    college grad n_ceo
## 1:       1    1    94
## 2:       1    0    78
## 3:       0    0     5
```

Compute several statistics for a variable

This is very useful for your analysis, to get a grasp on the data.

```r
dt.ceo.salaries[, list(
 mean_salary = mean(salary),
sd_salary = sd(salary),
min_salary = min(salary),
max_salary = max(salary),
median_salary = median(salary))]
```

Or we can do it by group, and get also the results:

```r
dt.ceo.salaries[, list(
 mean_salary = mean(salary),
sd_salary = sd(salary),
min_salary = min(salary),
max_salary = max(salary)), by = list(grad, college)]
```

Well formated on subsampled data

```r
stargazer(dt.ceo.salaries[grad == 1, list(age, salary)],
          type = "text")
```

```
##
## ========================================================
## Statistic N   Mean   St. Dev. Min Pctl(25) Pctl(75)  Max
## --------------------------------------------------------
## age       94 55.457   8.155   38     50       61     86
## salary    94 864.213 501.392  100  481.5   1,167.8  2,26
## --------------------------------------------------------
```

Doing a t-test with R

- It is crucial to form hypothesis on your data.
- In the lectures you learned about Hypothesis Testing.

A t-score is a standardized statistic that can be used to test an hypothetic value for the population mean (μ) given the sample mean $\bar{x}$ and the sample standard deviation (s):

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}}$$

Given the value of the t-score, the p-value is the smallest significance level at which the null hypothesis would be rejected. **We cannot reject the null hypothesis (the population mean is μ) if the p-value is greater than 0.05 for a confidence level of 95%**

Doing a t-test with R (Example)

Question:

Can we say that the mean CEO salary is statistically different from 800?

H_0 :

H_1 :

Doing a t-test with R (Example)

Question:

Can we say that the mean CEO salary is statistically different from 800?

H_0 : The mean salary of CEOs is 800. ($\mu = 800$)

H_1 : The mean salary of CEOs is not 800.

Doing a t-test with R (Example)

```r
dt.ceo.salaries[, t.test(salary, mu = 800)]
```

```
##
##  One Sample t-test
##
## data:  salary
## t = 1.4913, df = 176, p-value = 0.1377
## alternative hypothesis: true mean is not equal to 800
## 95 percent confidence interval:
##  778.7015 953.0274
## sample estimates:
## mean of x
##  865.8644
```

Doing a t-test with R (Example II)

Question:

Is the average salary different for CEOs with a graduate degree and those without?

H_0 :

H_1 :

Doing a t-test with R (Example II)

```r
dt.ceo.salaries[, t.test(salary ~ grad)]
#or
t.test(dt.ceo.salaries[, salary] ~ dt.ceo.salaries[,grad])
```

Doing a t-test with R (Example II)

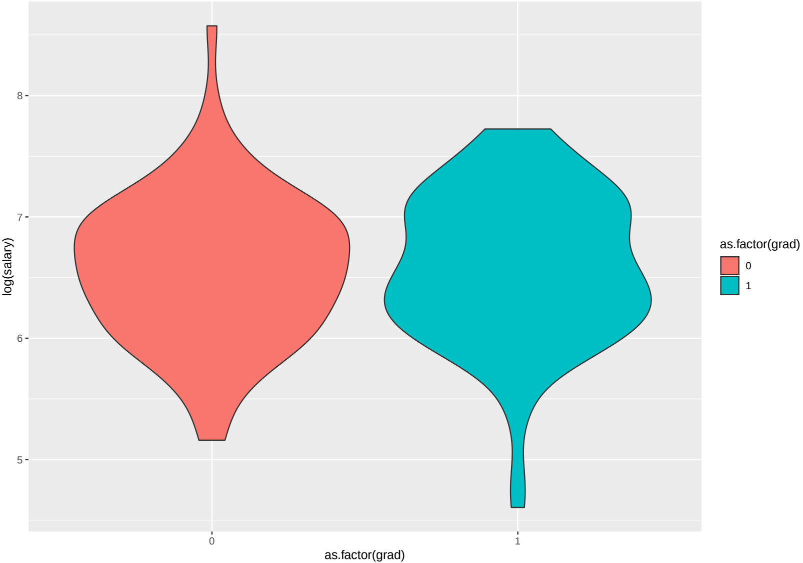
```
dt.ceo.salaries[, t.test(salary ~ grad)]
```

```
##
##  Welch Two Sample t-test
##
## data:  salary by grad
## t = 0.038973, df = 149.94, p-value = 0.969
## alternative hypothesis: true difference in means between
## 95 percent confidence interval:
##  -175.0489  182.0932
## sample estimates:
## mean in group 0 mean in group 1
##        867.7349        864.2128
```

Extra (Visualize)

```
ggplot(data, aes(x=as.factor(grad),
                 y=log(salary),
                 fill=as.factor(grad))) +
  geom_violin()
```

Extra (Visualize)


log(salary)
as.factor(grad)
as.factor(grad)