Lab 03

Jaime Montana

16/9/2021

Reminder of last session

- Working directory
- How to open a data set
- Libraries (using GUI and command line)
  - Installing packages `install.packages("PACKAGE_NAME")`
  - loading libraries `library(PACKAGE_NAME)`
- Open CSV using `read_csv()` [from **readr** package] and `fread()` [from **data.table** package]
- Explore the data
- Data manipulation (changing variables name, creating new variables, . . .)
- `t-test` in R
- First plot in R

Libraries to use in todays lab.

Exercise: Use the `library()` command to load the following libraries in your session. - `data.table` - `stargazer` - `ggplot2`

Tip: Recall to install the package so the library is available to be used by Rstudio.

Libraries to use in todays lab.

Exercise: Use the `library()` command to load the following libraries in your session. - `data.table` - `stargazer` - `ggplot2`

Tip: Recall to install the package so the library is available to be used by Rstudio.

```r
#install.packages("data.table")
#install.packages("stargazer")
#install.packages("ggplot2")
library(data.table)
library(stargazer)
library(ggplot2)
```

How to open a dataset (csv) **RECALL**

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.

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

Problems when importing `CSV`, `TXT` data to R session?

1. Encoding and value separators. é. Size (Big data). How is big data big? (Number of observations, number of variables)
2. Speed / efficiency

How to open a CSV (II)

What if the data is too big?

1. First import a few lines (i.e. 100), to see the content of the data set.
2. If you have a codebook available check the available information and make a list of the variables names that you need for your analysis.
3. Import the desired variables.

```
sample_data <- fread("ceosal2.csv", nrows = 100)
View(sample_data)
```

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(dt.ceo.salaries)
```

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

Remove an undesired object in your environment

To remove an object (or a list of objects) from your environment, use the function ‘`rm()`

```
rm(sample_data)
```

If you wish to remove everything, use `rm(list = ls())`. **USE IT WITH CAUTION!!!**

Import only the desired variables `data.table`

If we want to import only the variables: `salary`, `age`, `profits`. We store the name of the variables in a vector. And then we use the vector in the option "select" in the `fread()` function.

```
selected_variables <- c("salary",
                        "age",
                        "profits")

selected_ceo_data <- fread("ceosal2.csv",
                           select = selected_variables)
```

Visual inspection of your data

It is always a good idea to plot the relevant variables in your data, to have an idea on what is the relationship between the variables.

There are many type of graphs. The most common:

- Scatter plot
- Histogram
- Density plot
- Bar plots
- ...

A brief introduction to ggplot2

Today lab is just an introduction to Data Visualization in R. An introductory online guide can be found here. You can find also this e-book a valuable resource.

How `ggplot()` works?

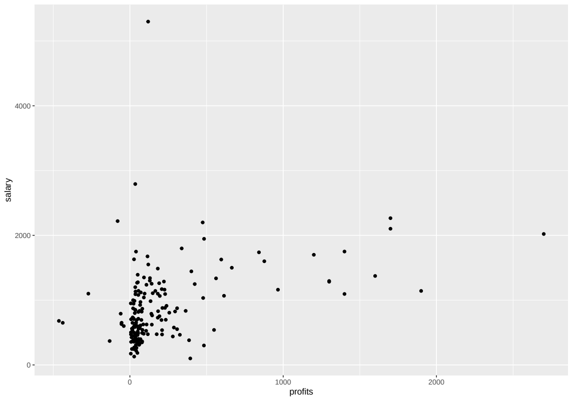
To better understand the notion of *layer* and *mapping*.

1. First a layer with the data: `ggplot(data=dt.ceo.salaries)` creates an empty graph.
2. Then we could add a layer of points: we use the function `geom_point()`
3. We define the **mapping** establishing the aesthetic: `aes(x = VAR1, y = VAR2)`

```
ggplot(data = dt.ceo.salaries) +
  geom_point(mapping = aes(x = profits,
                           y = salary))
```

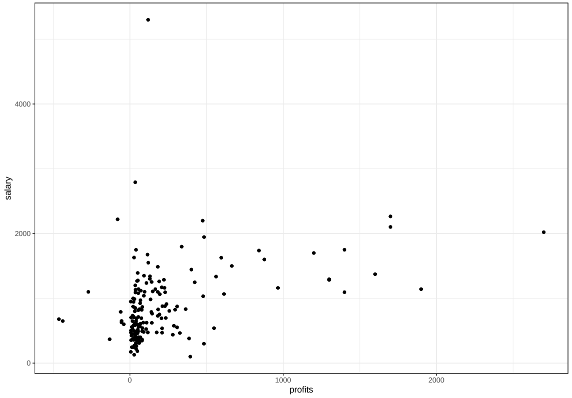
Brief intro to ggplot2 (II)

```
ggplot(data = dt.ceo.salaries) +
  geom_point(mapping = aes(x = profits,
                           y = salary))
```



Adding styles

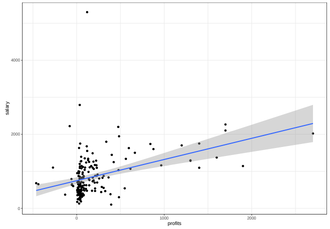
```
ggplot(data = dt.ceo.salaries) +
  geom_point(mapping = aes(x = profits,
                           y = salary)) +
  theme_bw()
```



Adding more layers

If the mapping is common to all layers, you can set it up in the plot definition, and it will inherit to next layers.

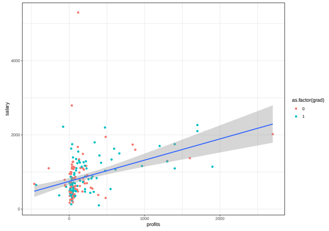
```
ggplot(data = dt.ceo.salaries,
       aes(x = profits,
           y = salary)) +
  geom_point() +
  theme_bw() +
  geom_smooth(method = "lm")
```



Adding more layers

You can add additional mappings.

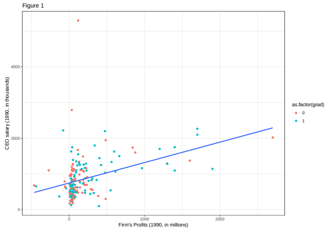
```
ggplot(data = dt.ceo.salaries,
       aes(x = profits,
           y = salary)) +
  geom_point(mapping = aes(colour = as.factor(grad))) +
  theme_bw() +
  geom_smooth(method = "lm")
```



Adding more layers

You can also modify labels and other graphical aspects.

```
ggplot(data = dt.ceo.salaries,
       aes(x = profits,
           y = salary)) +
  geom_point(mapping = aes(colour = as.factor(grad))) +
  theme_bw() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = 'Figure 1',
       x = "Firm's Profits (1990, in millions)",
       y = 'CEO salary (1990, in thousands)')
```

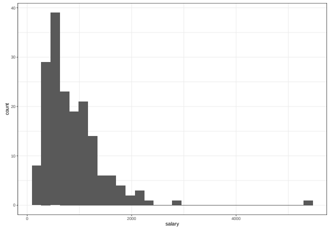


Other type of graphs

What about graphs that take in consideration one variable (i.e. if we are interested in the distribution).

Same concept. Define data, aesthetics and add a layer to study its distribution: `geom_histogram`

```
ggplot(data = dt.ceo.salaries,
       aes(x = salary)) +
  geom_histogram() +
  theme_bw()
```



Beyond visual inspection: Correlation

$$Cor(X,Y) = \rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

```
dt.ceo.salaries[, cor(profits, salary)]
```

```
## [1] 0.3939276
```

Beyond visual inspection: linear regression

To estimate a linear regression we use the function `lm()`. The function requires as input, two objects:

1. The data: a data.table or data frame type of objects.
2. A formula, that express the model (the relationship) to be estimated.

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

- In the left hand side y_i is the dependent variable. - The right hand side of the model contains the intercept , the independent variables (x_i), and the error term (ϵ_i).

The values β_k are k values to be estimated.

Beyond visual inspection: linear regression (II)

We need then to determine the model. For example, our proposed model studies the relationship between CEO salary and firm profits.

$$\mathit{salary} = \beta_0 + \beta_1 \mathit{profits} + \mathit{error}$$

Where:

- the dependent variable is the CEO salary
- the independent variable is the firm profits

1. We first define the model:

```
model0 <- as.formula(salary ~ profits)
```

Beyond visual inspection: linear regression (III)

2. We estimate the model in our data. As best practice we store in an object the result of our estimation.

```r
reg_model0 <- lm(formula = model0,
                 data = dt.ceo.salaries)
```

Beyond visual inspection: linear regression (IV)

3. To print the results we can use the function `summary()`

```
summary(reg_model0)
```

```
##
## Call:
## lm(formula = model0, data = dt.ceo.salaries)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -872.4 -319.7 -119.8  242.0 4484.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 746.9238    45.7979   16.31  < 2e-16 ***
## profits       0.5723     0.1009    5.67 5.81e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 541.6 on 175 degrees of freedom
## Multiple R-squared:  0.1552, Adjusted R-squared:  0.1504
## F-statistic: 32.14 on 1 and 175 DF,  p-value: 5.805e-08
```

$$\mathit{salary} = 746.92 + 0.57 \times \mathit{profits} + \mathit{error}$$

Beyond visual inspection: linear regression (V)

The package `stargazer` allow to have output from the regression in different formats, and well formatted.

```
stargazer(reg_model0, type = "text",  no.space = TRUE)
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                               salary
## -----------------------------------------------
## profits                      0.572***
##                               (0.101)
## Constant                    746.924***
##                              (45.798)
## -----------------------------------------------
## Observations                    177
## R2                             0.155
## Adjusted R2                    0.150
## Residual Std. Error     541.619 (df = 175)
## F Statistic           32.144*** (df = 1; 175)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

The result display the estimation results, and the fit statistics.

Uses of the regression coefficient results

Why we want to estimate the model?

Uses of the regression coefficient results

Why we want to estimate the model?

1. Understand the relationship between the variables, and its significance.
2. In many cases it also helps us to predict, and make “informed guess” or predict based on some information.

Doing predictions based on regression results

Consider our results from the estimation:

$$\textit{salary} = 746.92 + 0.57 \times \textit{profits} + \textit{error}$$

- How we do this in R, for a given profits value (`profits = 100`)?
- How we do this for the observed profit values?

1. Extract the value of the estimated coefficient into an object.
2. Use the object to operate (following our model).

Doing predictions based on regression results (II)

$$salary = 746.92 + 0.57 \times profits + error$$

1. Extract the value of the estimated coefficient into an object.

```r
reg_0.coeffs <- coefficients(reg_model0)
reg_0.coeffs
```

```
## (Intercept)     profits
## 746.9238154   0.5722961
```

Since the result is a vector we can access the k^{th} element of the vector with `vector_name[k]`. In this case for `k`=2

```r
reg_0.coeffs[2]
```

```
##   profits
## 0.5722961
```

Doing predictions based on regression results (II)

2. Use the object to operate (following our model), and the value choosen `profits=100`

$$\textit{salary} = 746.92 + 0.57 \times \textit{profits} + \textit{error}$$

```
reg_0.coeffs[1] + reg_0.coeffs[2]*100
```

```
## (Intercept)
##    804.1534
```

Doing predictions based on regression results (III)

Given that the model is stored in the estimated estimated object, we can use the function `predict()`.

```
predict(reg_model0, data.table(profits = 100))
```

```
##        1
## 804.1534
```

```
predict(reg_model0, data.table(profits = 100),
        interval = 'confidence')
```

```
##        fit      lwr      upr
## 1 804.1534 720.9844 887.3224
```

Predict values for all observations

```r
dt.ceo.salaries[, yhat := predict(reg_model0)]

head(dt.ceo.salaries[, list(salary, profits, yhat)], 5)
```

```
##    salary profits      yhat
## 1:   1161     966 1299.7619
## 2:    600      48  774.3940
## 3:    379      40  769.8157
## 4:    651     -54  716.0198
## 5:    497      28  762.9481
```

Extra: Resource

- If it happens that you have 45 minutes **free** this week:

Listen to the Freakonomics podcast of this week:

Why Does the Richest Country in the World Have So Many Poor Kids?

Among O.E.C.D. nations, the U.S. has one of the highest rates of child poverty. How can that be? To find out, Stephen Dubner speaks with a Republican senator, a Democratic mayor, and a large cast of econo-nerds. Along the way, we hear some surprisingly good news: Washington is finally ready to attack the problem head-on.