Introduction to Programming

Lecture 7-8: Introduction to R

Clément Mazet-Sonilhac
clement.mazet@sciencespo.fr

Sciences Po Paris

Disclaimer

- Most of the material is drawn from the excellent course prepared by software carpentry (adapted by Hugo Lhuillier for the last year course)
- In particular, most exercises are drawn from it (If you really want to learn something, don't look up the answers)
- Other source of inspiration is the very complete QuantEcon website

What and why ?

R : Let's start !

- Why are we using R ?
  - Better than Stata by ANY metric
  - Free
  - Extremely popular amongst scientists, in particular statistians and economists
  - Exists a large library of external packages

Variables

Create a variable in R

- A variable : a container with a name
- To create a variable called `weight` with value 55, just type :

  `weight <- 55` (or `weight = 55`)
- Can treat the variable like a regular number. Try `weight + 1`
- Can change an variable's value by assigning it a new value. Just type : `weight <- 60`

Variables

Create a variable in R

- R only stores the value, not the calculation used to create a variable ($\neq$ Excel)

  ```
  weightlb <- 2.2 * weightkg
  c(weightkg, weightlb)
  weightkg <- 80
  c(weightkg, weightlb)
  ```

- c is also a function (probably the most used function in R), stands for combine

Variables

Create a variable in R

- Some conventions on the name of variables

  1. start with lower case letters
  2. separate words with underscores
  3. use only lowercase letters, underscores, and numbers

Motivating example

Analyzing data w. `R`

- The data : We are studying inflammation in patients who have been given a new treatment for arthritis, and need to analyze the first dozen data sets. The data sets are stored in comma-separated values (CSV) format. Each row holds the observations for just one patient. Each column holds the inflammation measured in a day, so we have a set of values in successive days.

  1. Go to my Github repo (github.com/CMS27/IP2019) and download `r-novice-inflammation-data`
  2. Goal : load the data, calculate the average value of inflammation per day, plot the results

Motivating example

Analyzing data w. `R`

- Loading data :
  1. Set the directory where the data is stored with `setwd()`
     `setwd("C:/Users/Clement/.../data")`
  2. Import data in `d` with :
     `d = read.csv(file = "inflammation-01.csv", header = FALSE)`
- both setwd() and read.csv() are functions that takes some arguments
  1. the first argument of both functions is a String => put quotes
  2. the second argument of read.csv is what we call a Boolean value (either true or false). Header : whether the first line of the file contains names for the columns of data
  3. d = data frame. more on this later : but basically, like an excel sheet.

Motivating example

Analyzing data w. R

- Manipulating the data :

  1. Display the first lines of the data set with `head` :
     `head(d, n = 3L)`

  2. To take a subset of the data set, provide an index in square bracket : [# row, # column] :

     `d[1,1]` # first row, first column

     `d[c(1, 3, 5), c(10, 20)]` # rows (1, 3 and 5), columns (10 and 20)

     `d[1, 1:5]` # columns from (1 to 5) and row 1

     `d[, 1]` # all columns from row 1

Motivating example

Analyzing data w. `R`

- In our data set, each row is a patient, each column is a day, such that `d[1,1]` is the inflammation measured on patient 1 on day 1
- **Exercise 1** : given that `min(data)`, `max(data)`, `mean(data)` are functions returning the equivalent statistics on data, find :
  1. the minimum inflammation on day 1 across all patients

Motivating example

Analyzing data w. R

- In our data set, each row is a patient, each column is a day, such that `d[1,1]` is the inflammation measured on patient 1 on day 1
- **Exercise 1** : given that `min(data)`, `max(data)`, `mean(data)` are functions returning the equivalent statistics on data, find :
  1. the minimum inflammation on day 1 across all patients
  2. the maximum inflammation experienced by patient 5 (across all days)

Motivating example

Analyzing data w. R

- In our data set, each row is a patient, each column is a day, such that `d[1,1]` is the inflammation measured on patient 1 on day 1
- **Exercise 1** : given that `min(data)`, `max(data)`, `mean(data)` are functions returning the equivalent statistics on data, find :
  1. the minimum inflammation on day 1 across all patients
  2. the maximum inflammation experienced by patient 5 (across all days)
  3. the maximum inflammation on days 4, 8 and 12 across all patients

Motivating example

Analyzing data w. `R`

- In our data set, each row is a patient, each column is a day, such that `d[1,1]` is the inflammation measured on patient 1 on day 1
- **Exercise 1** : given that `min(data)`, `max(data)`, `mean(data)` are functions returning the equivalent statistics on data, find :
  1. the minimum inflammation on day 1 across all patients
  2. the maximum inflammation experienced by patient 5 (across all days)
  3. the maximum inflammation on days 4, 8 and 12 across all patients
  4. the minimum inflammation experienced by patients 3 and 6 from day 1 to 5

Motivating example

Analyzing data w. R

- In our data set, each row is a patient, each column is a day, such that `d[1,1]` is the inflammation measured on patient 1 on day 1

- **Exercise 1** : given that `min(data)`, `max(data)`, `mean(data)` are functions returning the equivalent statistics on data, find :

  1. the minimum inflammation on day 1 across all patients
  2. the maximum inflammation experienced by patient 5 (across all days)
  3. the maximum inflammation on days 4, 8 and 12 across all patients
  4. the minimum inflammation experienced by patients 3 and 6 from day 1 to 5
  5. the mean inflammation experienced by patients 2, 4 and 10 (across all days)

Motivating example

Analyzing data w. `R`

- Faster way to get some sufficient statistics (by columns) : `summary` (ex : `summary(d[, 1:5])`)
- What if we want some info, say the median, for each partient (= row) ? No such things as `rowMedian`
- `apply` : repeat a function on all of the rows (MARGIN = 1) or columns (MARGIN = 2) of a data frame (`apply(d, 1, median)`)
- **Exercise 2** : compute in two different ways the mean for the first 10 patients of our data

Motivating example

Analyzing data w. R

- R plot are very nice :
- Try `plot(apply(d, 2, max), xlab = "day", ylab = "maximum", main = "maximum inflammation by day")`
- and `boxplot(d, main = "Summary")`