

Exercise 6.

Input / Output

Create the script exercise6.R (in R Studio: File ->) and save it to the “Rintro/day3” directory: you will save all the commands of exercise 6 in that script.

Remember you can comment the code using #.

Exercise 6a. I/O on vectors

1. Make sure you are in **Rintro/day3**; if not, go there!

setwd(); getwd()

2. Copy directory **i_o_files (that is in /nfs/users/bi/sbonnin)** in **Rintro/day3**.

Use **file.copy()**: the “recursive” argument should be set to TRUE for directories.

3. Scan elements from **ex6a_input.txt** (that is in the **i_o_files** directory), and save in object **z**. How many elements are in **z**?

scan()

2. Sort elements of **z** and save into object **zsorted**.

sort()

3. Write **zsorted** into file **ex6a_output.txt**.

write()

4. Check the output; write the file again setting the argument **ncolumns** to 1 and see how it differs.

Exercise 6b - I/O on data frame: play with the arguments!

1. field separator

read **ex6b_IO_commas_noheader.txt** in the object **fs**.

```
fs <- read.table("i_o_files/ex6b_IO_commas_noheader.txt")
```

What are the dimensions of **fs**?

Fields/columns are separated by commas: change the default value of the **sep** argument and read in the file again.

What are now the dimension of **fs**?

2. field separator + header

read **ex6b_IO_commas_header.txt** in the object **fs_c**.

```
fs_c <- read.table("i_o_files/ex6b_IO_commas_header.txt")
```

What are the dimensions of **fs_c**? Change the default field separator.

The first row is supposed to be the header (column names): change the default value of the header parameter and read in the file again.

What are now the dimensions of **fs_c**?

3. skipping lines

read **ex6b_IO_skip.txt** in the object **sk**.
`sk <- read.table("i_o_files/ex6b_IO_skip.txt")`

R is complaining about line 2.... check “manually” the file (in the R Studio file browser for example).

The **skip** argument allows you to ignore lines before reading in a file. Introduce this argument and read the file again.

What are the dimensions of **sk**? Change the default field separator.

What are now the dimensions of **sk**?

4. comment lines

read **ex6b_IO_comment.txt** in the object **cl**.
`cl <- read.table("i_o_files/ex6b_IO_comment.txt")`

R is complaining (again!): check manually the file and try to find out what is wrong.

Adjust the **comment.char** argument and read the file again.
Adjust also the **header** and **sep** arguments to read in the file correctly.

What are now the dimensions of **cl**?

4. final

read **ex6b_IO_final.txt** in object **fin**.
`fin <- read.table("i_o_files/ex6b_IO_final.txt")`

Adjust the appropriate parameters to obtain the data frame **fin** of dimensions 167 x 4.

Exercise 6c – I/O on data frame

1. Read in file **ex6c_input.txt** (that is in **i_o_files**) in object **ex6**. The file has a header!
2. Check the structure of **ex6** with the **str** function.
3. Now read in the same file but, this time, set the argument **as.is** to TRUE. Check again the structure: what has changed?
4. What are the column names of **ex6**?
5. Change the name of the first column of **ex6** to "Country"
6. How many countries are in the Eurozone, according to this table?
table()
7. Reassign "**TRUE**" with "**yes**" and "**FALSE**" with "**no**" in the Eurozone column.

8. How many country names contain the letter “c”?
Remember the **grep** function. Check the help page.

9. How many people live, according to that table:
* in the European union (whole table)?
* in the Eurozone?

10. Write **ex6** into file “**ex6c_output.txt**”.

write.table()

* Try with the default arguments.

* Add **row.names** set o FALSE.

* Add **quote** set to FALSE.

* Add **sep** set to “\t” or to “,”

Check the output at each step.

Exercise 7

Packages

Create the script exercise7.R (in R Studio: File ->) and save it to the “Rintro/day3” directory: you will save all the commands of exercise 7 in that script.

Remember you can comment the code using #.

1. Install and load packages **ggplot2** and **WriteXLS**.

You can do it either from R Studio or from the terminal with **install.packages** function.

2. ggplot2 loads automatically the **diamonds** dataset in the working environment: you can treat it as an object after ggplot2 is loaded.

What are the dimensions of **diamonds**? What are its column names?

You can read the help page of the **diamonds** dataset to understand what it contains.

Note: diamonds is a data frame: is.data.frame(diamonds) returns TRUE.

3. Select the columns **carat**, **cut**, **color** and price of **diamonds** and store in the object **diams1**.

4. Install and load package **dplyr**.

5. Randomly sample **200 lines of diams1**: save in **diams** object. use the **sample_n** function of dplyr. Check the help page.

6. Export diams into 2 files:

a. diamonds200.txt – **write.table()**

b. diamonds200.xls – **WriteXLS()**

Note: read about and play with the different options of both functions and check the output files.