1- Introduction to the R language

Alex Sanchez, Miriam Mota, Ricardo Gonzalo and Santiago Perez-Hoyos

Statistics and Bioinformatics Unit. Vall d'Hebron Institut de Recerca

Readme

- License: Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License http://creativecommons.org/licenses/by-nc-sa/4.0/
- You are free to:
  - **Share** : copy and redistribute the material
  - **Adapt** : rebuild and transform the material
- Under the following conditions:
  - **Attribution** : You must give appropriate credit, provide a link to the license, and indicate if changes were made.
  - **NonCommercial** : You may not use this work for commercial purposes.
  - **Share Alike** : If you remix, transform, or build upon this work, you must distribute your contributions under the same license to this one.

Section 1

Introduction to R

Outline

- A first contact with R & Rstudio.
  - How does one work with R
- A primer of data import
  - Reading data into R
- A primer of communication
  - R Notebooks and RMarkdown

What is R?

- R is a *language and environment* for statistical computing and graphics.
- R provides a wide variety of statistical and graphical techniques, and is highly extensible.
- It compiles and runs on a wide variety of UNIX platforms and similar systems Windows and MacOS.

R PRO's (why you are here!)

- The system is
  - free (as in *free beer*)
  - It's platform independent
  - It is constantly improving (2 new versions/year)
- It is a statistical tool
  - Implements almost every statistical method that exists
  - Great graphics (Examples)
  - Simple reporting tools
  - Also state-of-the-art in Bioinformatics through the Bioconductor Project.
- Programming language
  - Easy to automate repetitive tasks (Example_1.1)
  - Possibility to create user friendly web interfaces with a moderate effort. (Examples)

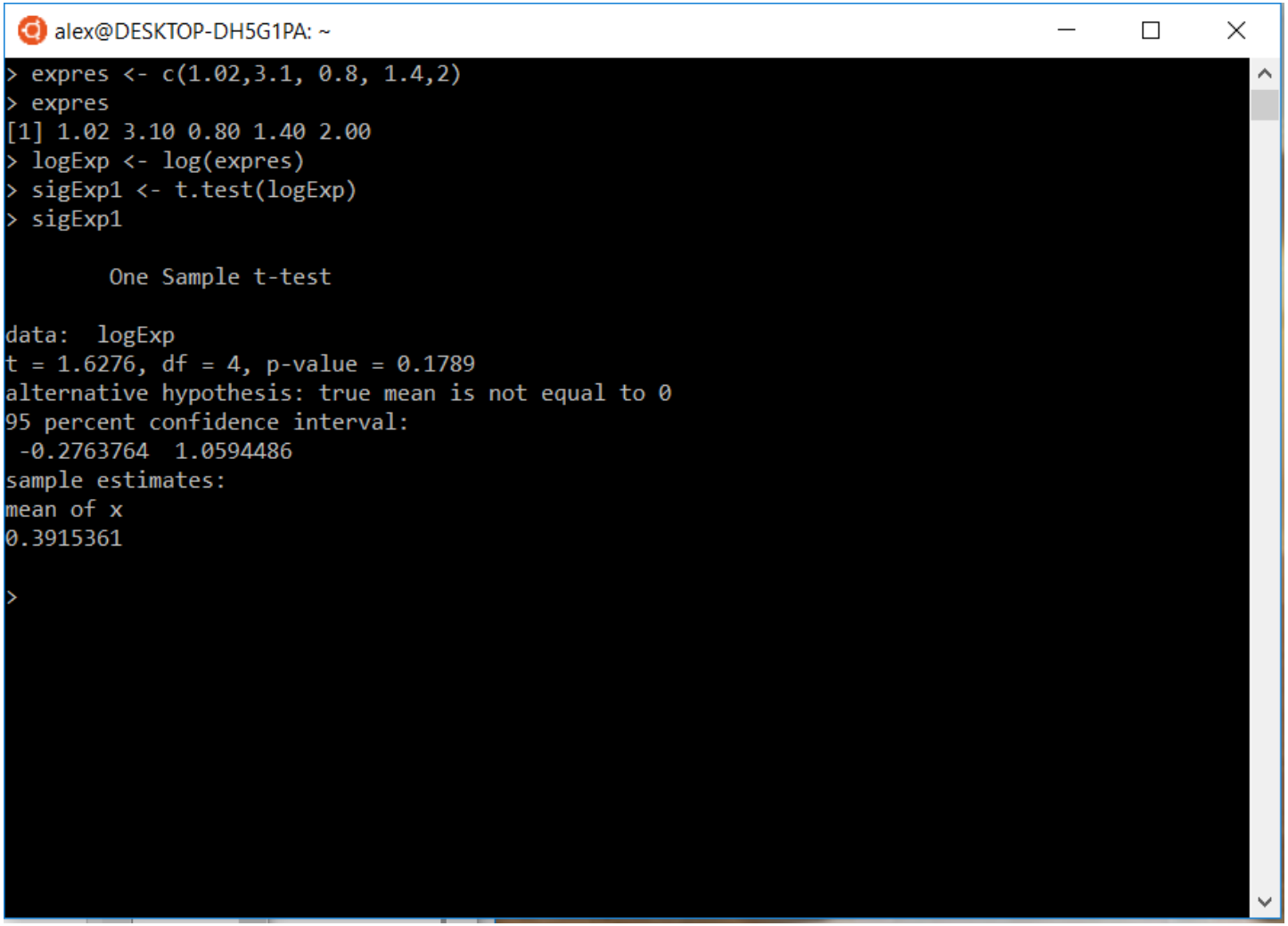
R CON's

- R is mainly used issuing commands from a console
  - less user friendly than almost any other statistical tool you may know.
- Constantly having new versions may affect our projects
- Not necessarily the best language nor suitable for every existing task

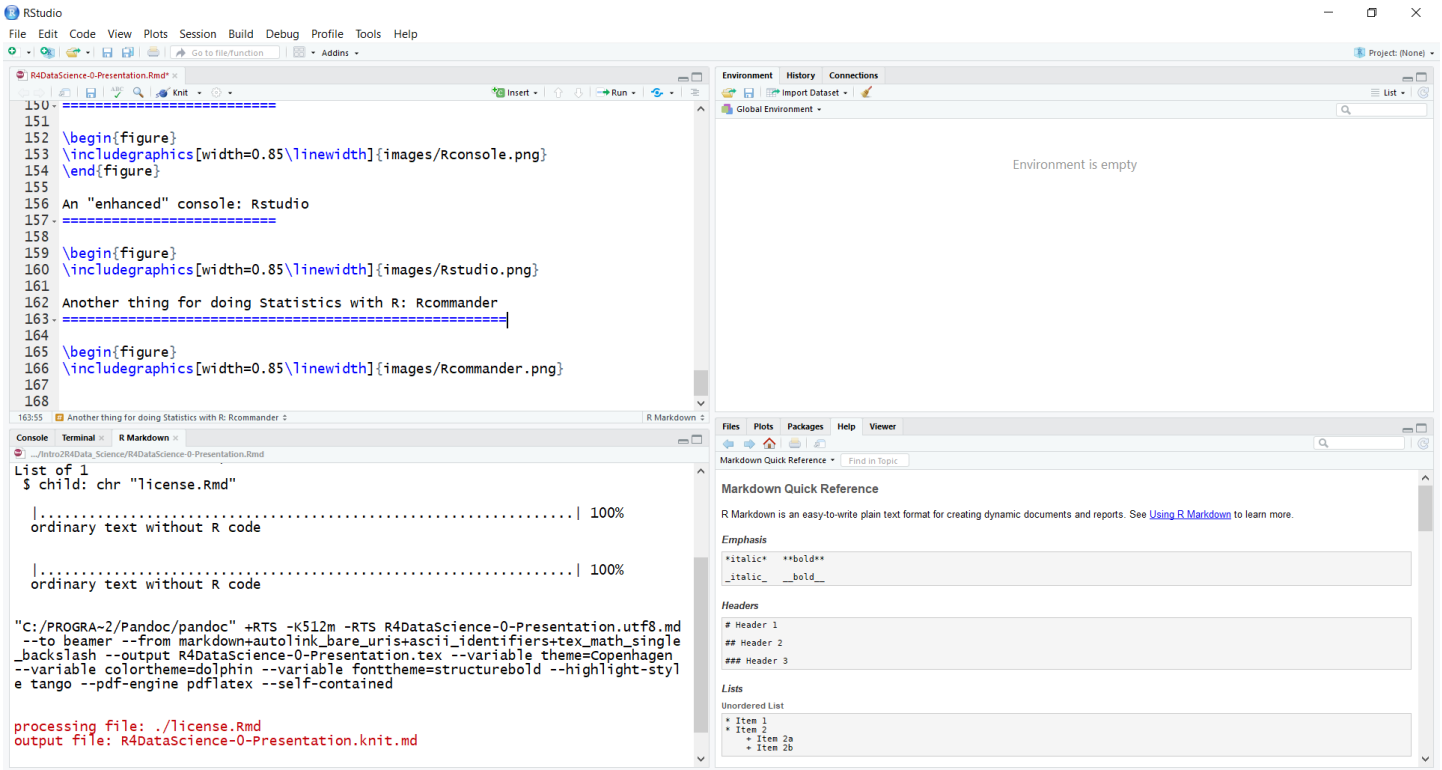
How is R used

- Traditionally R was used from an Operating System console ("Terminal")
- This is an intimidating approach for many users
- A variety of options exist to decrease the learning curve.
  - Use a supportive development environment such as **Rstudio**
  - Use an interface to Statistical tools, such as **Rcommander** or ::DeduceR** allowing to concetrate an Statistics, not in commands.

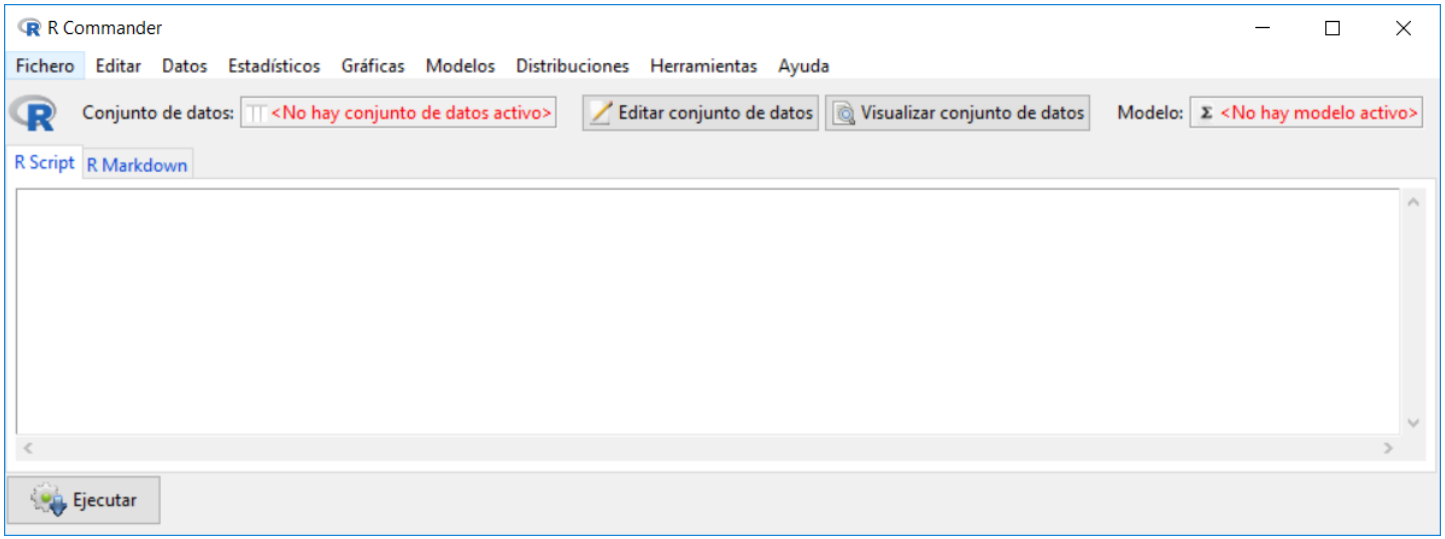
A raw R console in linux



An "enhanced" console: Rstudio



Something that is not a console: Rcommander



Section 2

Dynamic output with Rmarkdown

Reproducible research with Rmarkdown

- R and Rstudio are strongly involved in promoting reproducibility and reproducible research.
- This is implemented in **Rmarkdown**
- A Rmarkdown combines
  - Natural language text, e.g. describing what we are doing in our own words.
  - R code with the instructions needed to do the data management or the analysis.
  - The output of the analysis

Creating Rmarkdown

- A Rmarkdown can be created in Rstudio with
  - `File --> New File --> Rmarkdown`
- The Rmarkdown contains example text and code so it is straightforwoard to adapt it to your analysis.
- To produce an html file with text, code and output:
  - Press the button "Knitr to Html"

Exercise

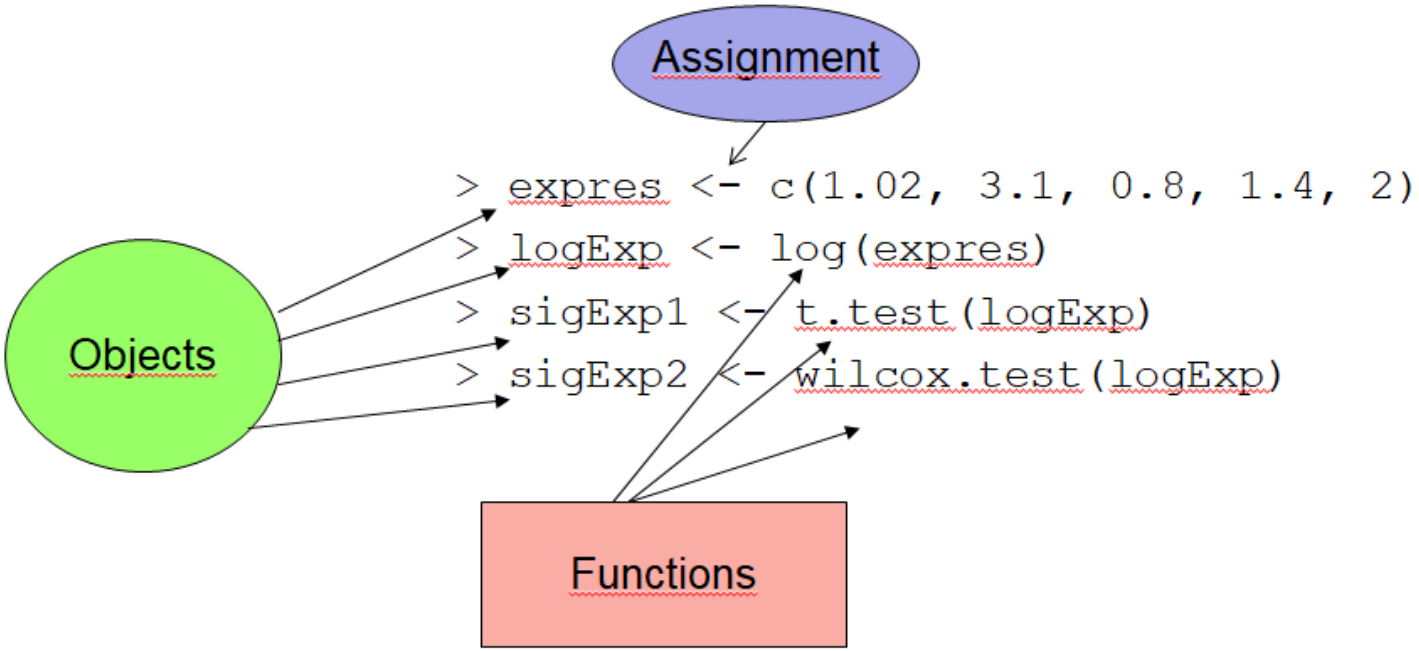
- Create a new Rmarkdown document
  - Include a title and your name
  - Compile document with 'knitr to html'

Section 3

Using R

Commands, Objects and Functions

- Shortly, using R consists of
  - Working with *objects* using *commands* and *functions*



Variables and data types

- Data managed in R . . .
  - is stored as *variables*
- Variables can be of distinct types
  - Numerical
    - numeric (13.7)
    - int (3)
  - Character
    - "R is cute"
  - Factors
    - A,B,C,D
    - WT, Mut

Exercise

- Copy and execute next commands:

```
expres <- c(1.02, 3.1, 0.8, 1.4, 2)
class(expres)
logExp <- log(expres)
sigExp1 <- t.test(logExp)
sigExp2 <- wilcox.test(logExp)
```

R packages

- R can be used for many different types of data processing and analysis from distinct fields, besides statistics such as Ecology, Omics Sciences, Psychology etc.
- All these capabilities are not present from the begining because most of them will never be used by most users.
- Instead, thay can be added when needed by
  (i) installing and
  (ii) loading the appropriate packages.

Installing and loading packages

We want to analyze some data using cox proportional hazards model.

```
res.cox <- coxph(Surv(time, status) ~ sex, data = lung)
```

```
Error in coxph(Surv(time, status) ~ sex, data = lung)
: could not find function "coxph"
```

We need to install and load the package before we can use it.

```
install.packages("survival")
library(survival)
res.cox <- coxph(Surv(time, status) ~ sex, data = lung)
```

The tidyverse

- The tidyverse is an opinionated collection of R packages designed for data science.
- All packages share an underlying design philosophy, grammar, and data structures.
- The complete tidyverse collection can be installed with:

```
install.packages("tidyverse")
```

- https://www.tidyverse.org/

Exercise

- Install the `rio` package from menu.
- Load the `rio` package.

Section 4

Getting data into R

Importing data with Rstudio

- The easiest way to get data into R is to click on the `Ìmport Datasets` button.
- Alternatively R code can be written using functions from `Base R`, the `tidyverse` or `rio` package
  - `Base R` functions start with `read.`: `read.table`, `read.csv`
  - `tidyverse` functions start with `read_`: `read_delim`, `read_csv` or `read_excel`
  - `rio` function is `import`

Reading Excel or csv files

- Files can be read from any location, let it be a physical support or a web site.
- To read files from disk be sure to indicate their location.
- Alternatively the default working directory can be set to the folder where the file is located.
- Assume files `Diabetes.xls` and `Osteoporosis.csv` have been downloaded from url to a sub-folder named `datasets`
- Start setting the default directory to the folder where you have saved the `datasets` folder.
  - `Session --> Set Working directory --> To source file location...`
- Import the `diabetes.xls` and the `osteoporosis.csv` file with the default options

Reading Excel or csv files (continued)

The code generated for reading the files can be reused any time changing the file name if needed.

```
# Read Excel file
library(readxl)
diabetes <- read_excel("../datasets/diabetes.xls")
```

Reading text files

- Text files may require that more information is provided about delimiters, decimal dign, locale (language) or page encoding (UTFB for Mac or Linux vs ISO-8859-1 for Windows).
- All options can be selected from the rstudio importer

```
library(readr)
osteoporosis <- read_delim("../datasets/osteoporosis.csv",
    "\t", escape_double = FALSE, locale = locale(date_name
    decimal_mark = ",", encoding = "ISO-8859-1"))
```

Reading Excel or csv files with rio

```
require(rio)
import("../datasets/diabetes.xls")
import("../datasets/osteoporosis.csv", dec = ",")
```

Interlude: Summarizing data

- Once a dataset is available it is easy to "have a look at it"

```
head(diabetes)
str(diabetes)
dim(diabetes)
summary (diabetes)
```

Section 5

Resources and exercises

Introductory materials

The web is full of all types of materials about R

Below there are a couple of brief introductions:

- A short introduction to R
- Getting started with R

Exercise

- Select a dataset with which you wish to work along the course.
- Read it into R
  - How many variables are there in it
  - What are their types
- Try to summarize it briefly
- Create an Rmarkdown to encapsulate all your steps and share it with somebody.