R crash course: A quick introduction to R

Alex Sanchez-Pla
Departamento de Genetica, Microbiologa y Estadistica (UB)
Unidad de Estadistica y Bioinformatica (VHIR)
Version 2021-11-17

Introduction

A Crash Course in R. Outline

- Why R
  - R basics
  - How does one work with R and Rstudio
- Getting Started
  - A primer of data import
  - Variables and data types
  - Functions, Packages and more stuff
- Working with data
  - *Selecting, Filtering and ordering datasets*
  - *A primer of statistics and plots*
  - R Notebooks and RMarkdown

Motivation

- We (you) all work with data, most of the time and often we need to do "things" with those data.
    - I have three lists of genes and I would like to see which genes they have in common (or which ones appear only in one list).
    - We have received the data from that lab but I only want to work with a subset of the samples.
    - Is it possible to repeat that plot changing the line colors, the font size etc?
    - I have some scripts tu re-run an analysis but I don't know how to start
- These, and many other things can be done with a basic knowledge of R.

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 can be used fro simple tasks to highly complex reproducible projects.
- It compiles and runs on a wide variety of UNIX platforms and similar systems Windows and MacOS.

R PRO's (why you are here!)

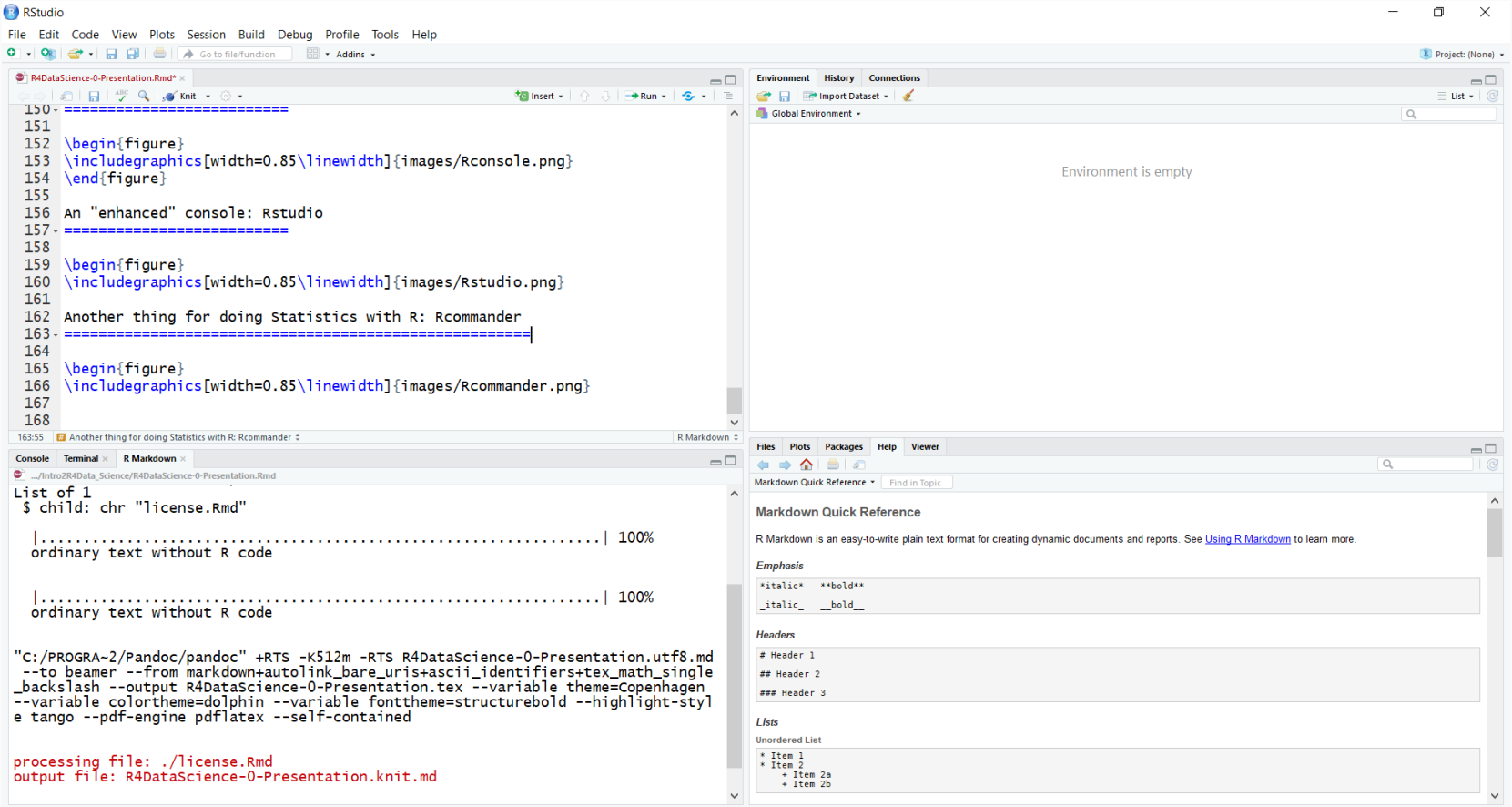
- The system is
  - free (as in *free beer*)
  - It's platform independent
  - It is constantly improving (2 new versions/year)
- It is a statistical tool
  - Implements almost every statistical method that exists
  - Great graphics (Examples)
  - Simple reporting tools
  - Also state-of-the-art in Bioinformatics 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

- Different ways to use R, but the best trade-off simplicity-efficiency is provided by Rstudio



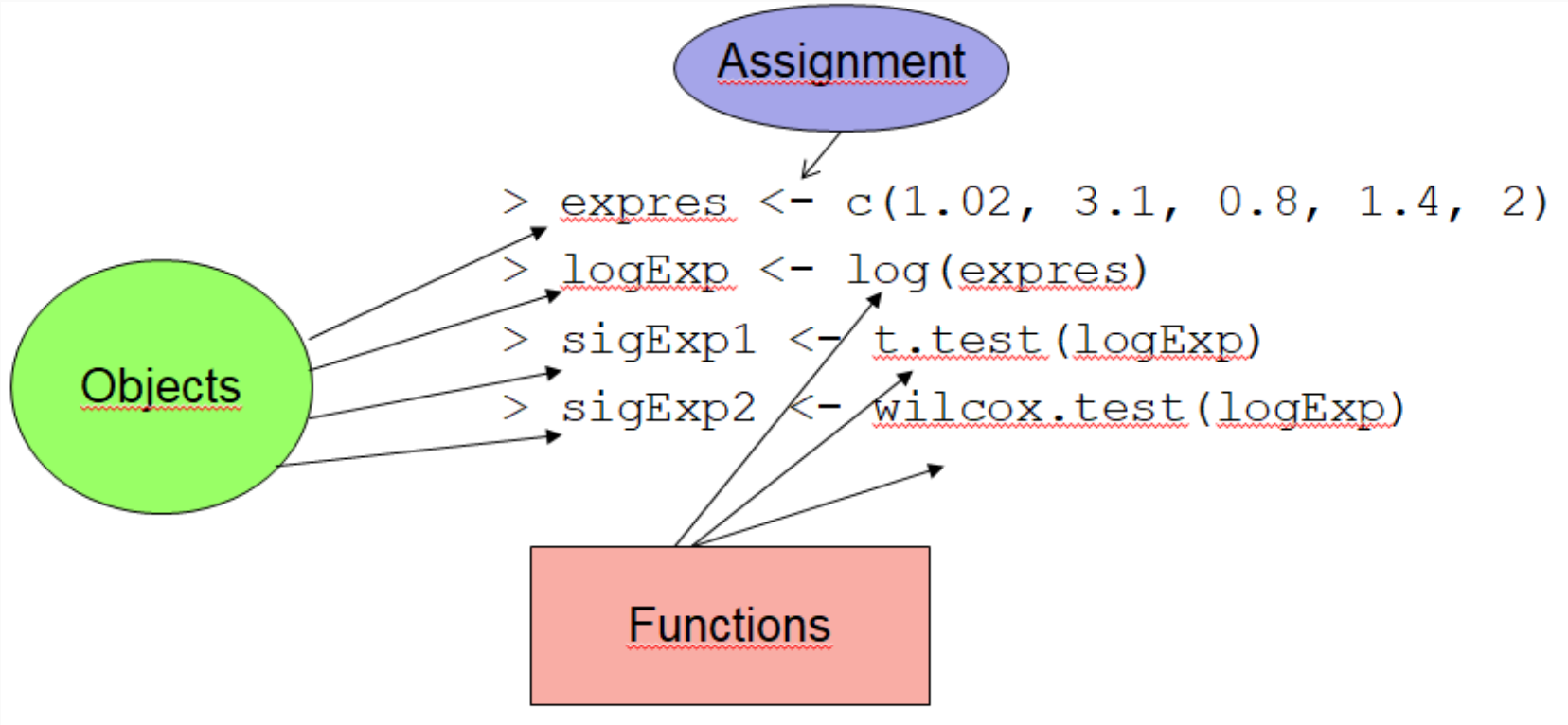
Exercise

- Get to know R. Visit the R-project page and see what can be found there.
- If you haven't done it before, download and install R and Rstudio in your computer
- Open R studio. Look at the panels and figgure out what can we do at each window.

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

- Variables can be contained in distinct *structures*
  - vectors
  - matrices
  - data.frames
  - lists
  - tibble
  - or specific classes that combine multiple structures such as
    - Bioconductor's expressionSet or summarizedExperiment

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
    - *installing* and
    - *loading* the appropriate packages.

Installing and loading packages

- Imagine we want to analyze some data using cox proportional hazards model.
- A colleague has provided us with some code:

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

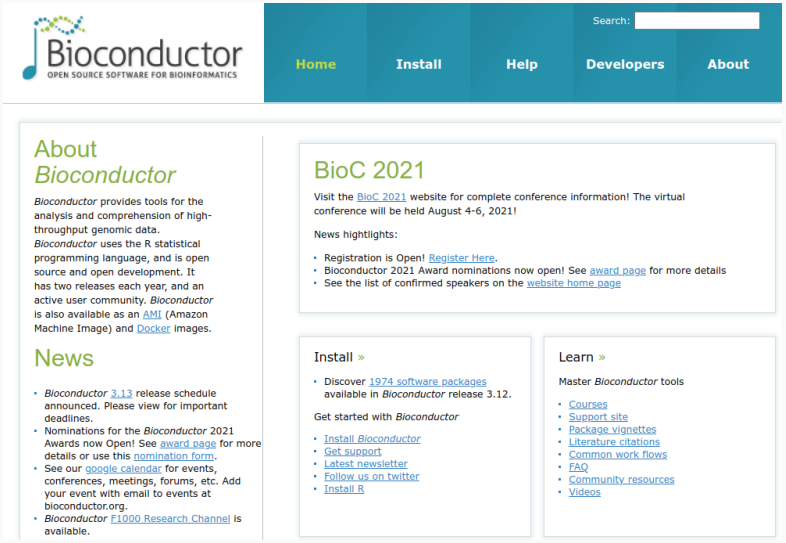
```
Error in coxph(Surv(time, status) ~ sex, data = lung) : could not find function
"coxph"
```

- It has not worked because the needed functions were not available
- 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)
```

Bioconductor

- Packages analyse all kinds of Genomic data (>800)
- Compulsory documentation (*vignettes*) for each package
- 6-month release cycle
- Course Materials
- Example data and workflows
- Common, re-usable framework and functionality
- Available Support
    - Often you will be able to interact with the package maintainers / developers and other power-users of the project software



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:

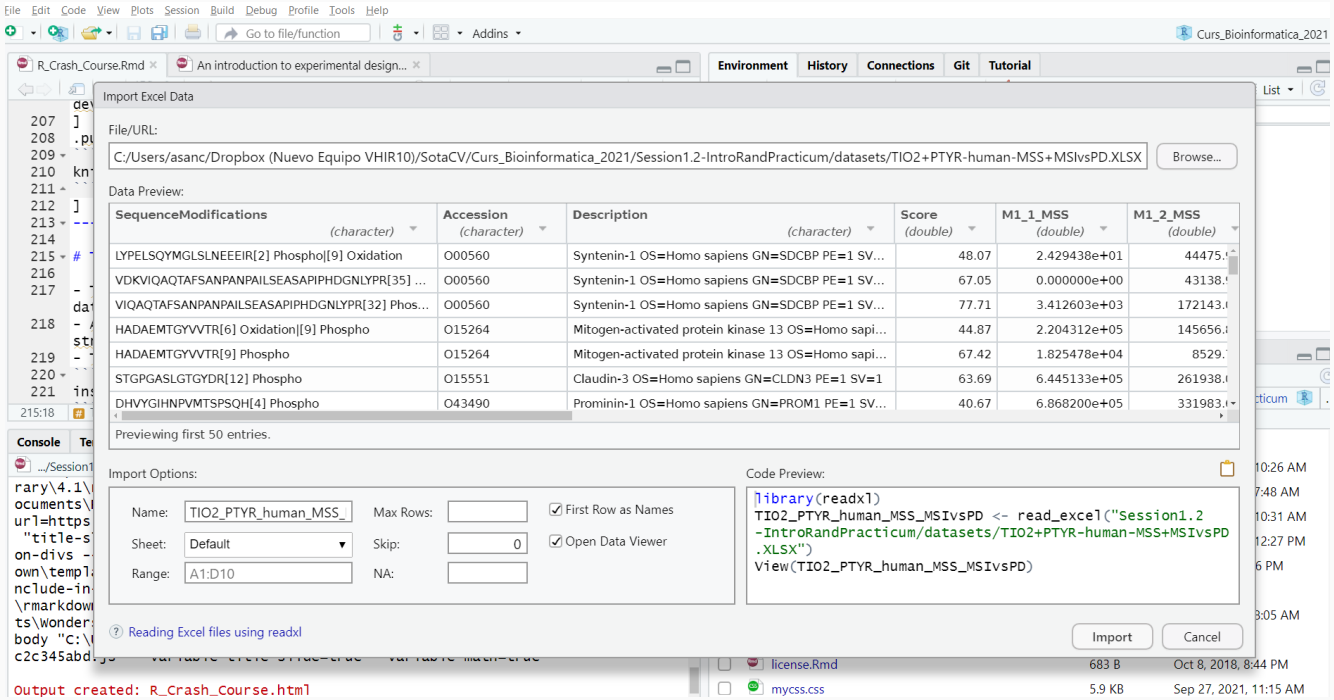
```
install.packages("tidyverse")
```

- https://www.tidyverse.org/

Getting data into R

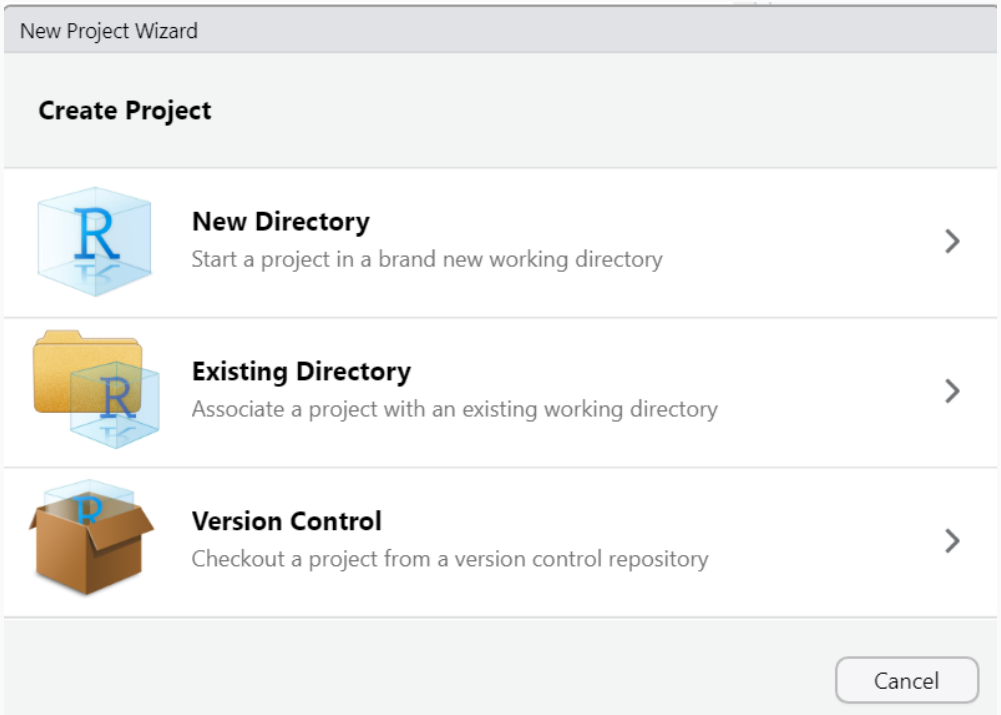
Importing data with Rstudio

- 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` or the `tidyverse`
  - `Base R` functions start with `read.`: `read.table`, `read.csv`
  - `tidyverse` functions start with `read_`: `read_delim`, `read_csv` or `read_excel`



Working with projects

- Files can be read from any location, let it be a physical support or a web site.
- The simplest and best way to control file location and modularity of your analyses is to create an Rstudio project for each new analysis.
  - Easy way to keep together your data, code and results.
  - Increases portability (avoids forgetting a file in an external folder).
  - I opens the door to infinite possibilities when you learn to *clone* github projects.



Reading Excel or csv files

- R allows importing any type of dataset, _either with "base" packages or using additional ones.
- An easy way to learn how to import a datset is to do this using the `import` menu and then to check the resulting R code. after importing file `TIO2+PTYR-human-MSS+MSIvsPD.XLSX` the following code has been created (it appears in the console)

```
library(readxl)
TIO2_PTYR_human_MSS_MSIvsPD <- read_excel("datasets/TIO2+PTYR-human-MSS+MSIvsPD.XLSX")
View(TIO2_PTYR_human_MSS_MSIvsPD)
```

Exercise

- Repeat the import process using the differnt files contained in the `datasets.zip` file
- Can you tell the differences between the files you have imported?
- What is the type of file "Data2HM"

Managing and exploring data

Data Exploration with R

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

```
phosphoProtData← read_excel("Session1.2-IntroRandPracticum/datasets/TIO2+PTYR-human-M
head(phosphoProtData)
str(phosphoProtData)
summary (phosphoProtData)
```

- Do it by yourselves and notice that categorical variables have been read as characters.

Dynamic output with Rmarkdown

Reproducible research with R notebooks

- R and Rstudio are strongly involved in promoting reproducibility and reproducible research.
- This is implemented in **R notebooks**
- A notebook 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 Notebooks

- A notebook can be created in Rstudio with
  - `File ⟶ New File ⟶ R Notebook`
- The notebook 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 "Preview"
  - Or Select "Knitr to Html"

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 R notebook to encapsulate all your steps and share it with somebody.