1- Introduction to the R language

Alex Sanchez, Miriam Mota and Santiago Perez-Hoyos

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

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 report
  - R Notebooks and RMarkdown

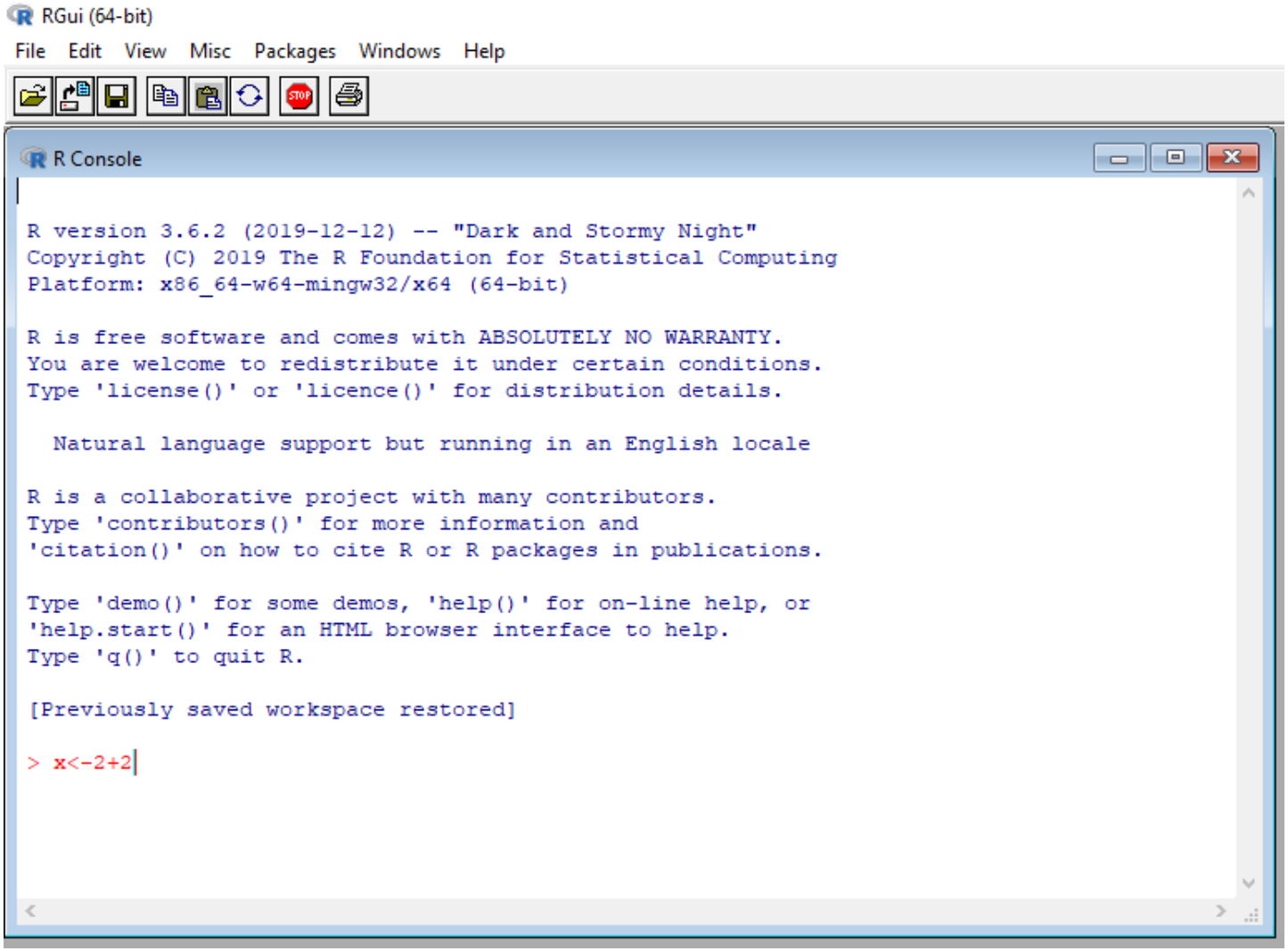
Key Terms

- ***R*** is a *language and environment* for statistical computing and graphics.
- ***R Studio*** Graphical User Interface for easier use of R
- ***Objects*** Everything you store in R (datasets, variables, lists, results and graphs ouputs) that can be referenced and reused
- ***Functions*** Pre-builded lines of code that execute actions after inputing some parameters.
- ***Packages or library*** Shareable bundle of code and documentation that contains pre-defined functions. R contains base packages and for some analysis you must install and call specific ones.
- ***Scripts*** Document file that hold your commands that can be run later.
- ***Rmarkdown*** Special type of Scrip that can mixed text and comments with R commands that can be compiled in a final pdf,

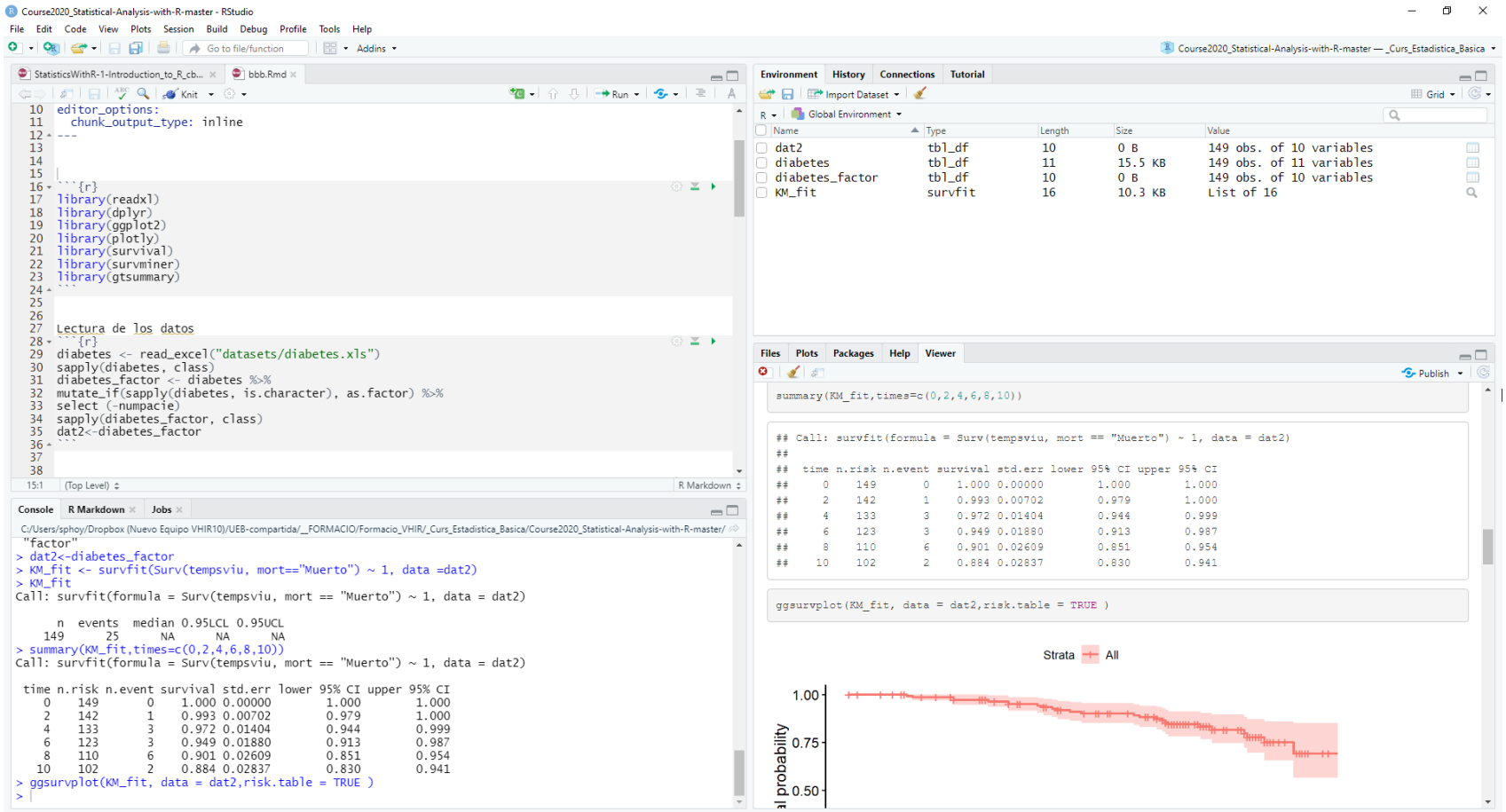
How is R used

- 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 with menus, such as **Rcommander** or **Jamovi** allowing to concetrate on Statistics, not in commands.

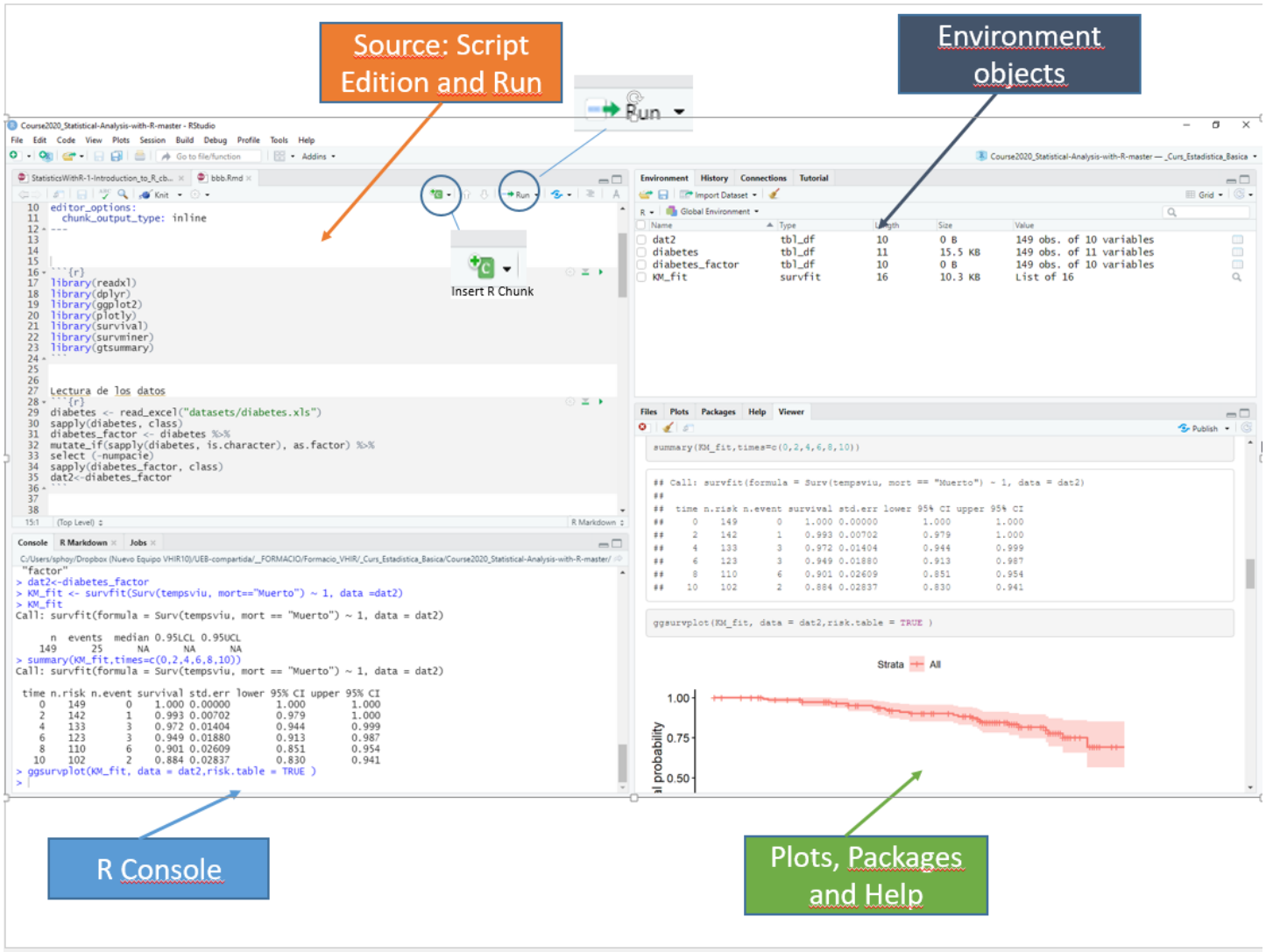
A raw R console



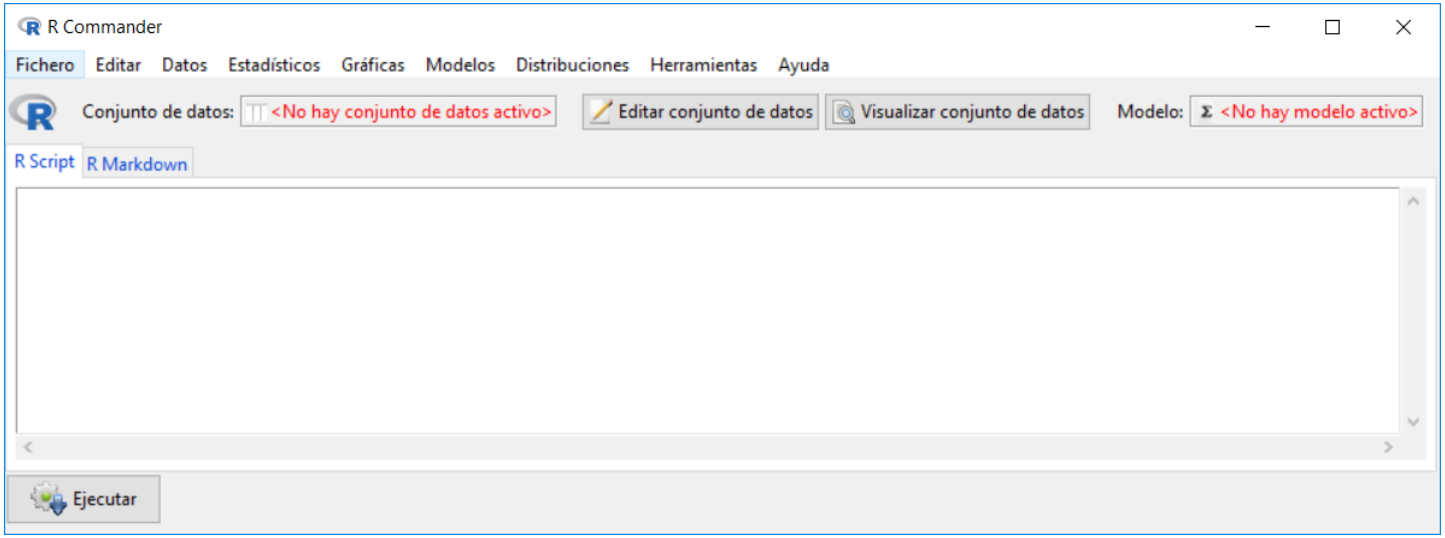
An "enhanced" console: Rstudio



An "enhanced" console: Rstudio



Something that is not a console: Rcommander



Exercise

- Open R-Studio
- Indentify Panes in R
- Calculate 2+2 in the console

Open new Script

- Run 2+2 Command line in Script

Section 2

Using R

Commands, Objects and Functions

Examples

- Vector

```
x1<-c(3,4,6,9,12)
x2<-c(3,4,6,9,20)
```

- Data Frames

```
dades<-data.frame(x1,x2)
```

- Results of execution of Functions

```
summary(x1,dat=dades)
```

Exercise

- Create a Script with the commands of the previous slide and see the results

```
x1<-c(3,4,6,9,12)  # Create vector
x1  # Show vector
x2<-c(3,4,6,9,20)  # Create vector
x2 # Show vector
dades<-data.frame(x1,x2)  # Create database
dades # Show database
summary(x1,dat=dades) # Summary measures from database
```

Exercise

```
x1<-c(3,4,6,9,12)  # Create vector
x1  # Show vector
```

```
## [1]  3  4  6  9 12
```

```
x2<-c(3,4,6,9,20)  # Create vector
x2 # Show vector
```

```
## [1]  3  4  6  9 20
```

```
dades<-data.frame(x1,x2)  # Create database
dades # Show database
```

```
##   x1 x2
## 1  3  3
## 2  4  4
## 3  6  6
## 4  9  9
## 5 12 20
```

```
summary(x1,dat=dades) # Summary measures from database
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     3.0     4.0     6.0     6.8     9.0    12.0
```

Section 3

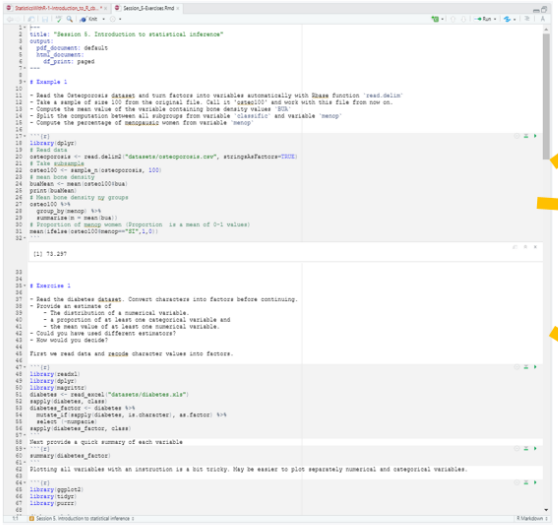
Dynamic output with Rmarkdown

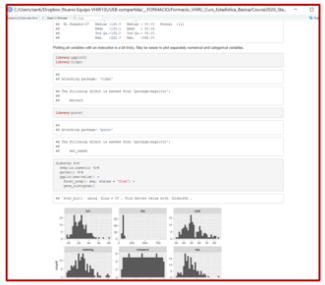
Reproducible research with Rmarkdown

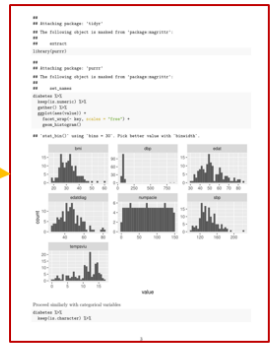
- R and Rstudio are strongly involved in promoting reproducibility and reproducible research.
- This is implemented in **Rmarkdown**
- A Rmarkdown combines
  - Natural language text, e.g. describing what we are doing in our own words.
  - R code with the instructions needed to do the data management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

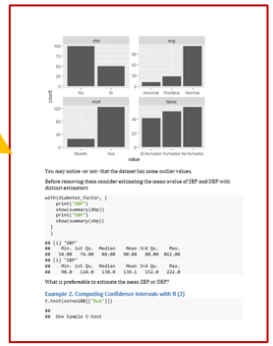
Reports in Rmarkdown

Rmardown Script









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"

Rmardown Script

StatisticsWithR-1-Introduction_to_R_cb... | Untitled1

Knit | Run

```
---
title: "Demo Rmarkdown"
author: "UEB"
date: "27/5/2021"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
```

YAML

R Chunk

Markdown

R Chunk

Markdown

R Chunk

Markdown

Exercise

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

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 `Import 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

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.