

# Practical Session Instructions

*Stefano De Sabbata*

*24 October 2018*

## Libraries

This practical session will make use of the “Tidyverse” libraries. If not already installed, please install Tidyverse before cloning the repository, using the command below or via the *Tools > Install Packages* menu in RStudio.

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

## Markdown

The main tool used to create this reproducible lecture and practical on reproducibility is RMarkdown. That is an R library that allows you to create scripts that mix the Markdown mark-up language and R, to create dynamic documents. RMarkdown script can be compiled, at which point, the Markdown notation is interpreted to create the output files, while the R code is executed and the output incorporated in the document.

The core Markdown notation used in this session is presented below and its interpretation when compiled is further below.

```
# Header 1
## Header 2
### Header 3
#### Header 4
##### Header 5
```

```
**bold**
*italics*
```

```
[This is a link to the University of Leicester](http://le.ac.uk)
```

- Example list
  - Main folder
    - Analysis
    - Data
    - Utils
  - Other bullet point
- And so on
  - and so forth

# Header 1

## Header 2

### Header 3

#### Header 4

#### Header 5

**bold** *italics*

This is a link to the University of Leicester

- Example list
  - Main folder
    - \* Analysis
    - \* Data
    - \* Utils
  - Other bullet point
- And so on
  - and so forth

## R Markdown

R code can be embedded in RMarkdown documents as in the example below. That will result in the code chunk be displayed within the document (as *echo=TRUE* is specified), followed by the output from the execution of the same code.

```
```{r, echo=TRUE}
for (i in 1:4) {
  if (i %% 2 == 0){
    cat("even \n")
  } else {
    cat("odd \n")
  }
}
```
```

```
for (i in 1:4) {
  if (i %% 2 == 0){
    cat("even \n")
  } else {
    cat("odd \n")
  }
}
```

```
## odd
## even
## odd
## even
```

## Part 1

To create an RMarkdown document in RStudio, select *File > New File > R Markdown ....* On the RMarkdown document creation menu, specify a title and your name as author, and select *PDF* as default output format. The new document should contain only the core document information, as in the example below.

```
---
title: "Practical Session Instructions"
author: "Stefano De Sabbata"
date: "24 October 2018"
output: pdf_document
---
```

Copy the following text below the document information and the click on the *Knit* button on the bar above the editor panel (top-left area) in RStudio, on the left side. Check the resulting *pdf* document. Try add some of your own code (e.g., from previous practical sessions) and Markdown text, and compile the document again.

```
# A nice heading
```

```
This is my first [RMarkdown](https://rmarkdown.rstudio.com/) document.
```

```
The code below will print:
```

```
- "even"
  - if the number is even
- "odd"
  - otherwise
```

```
```{r, echo=TRUE}
for (i in 1:4) {
  if (i %% 2 == 0){
    cat("even \n")
  } else {
    cat("odd \n")
  }
}
```
```

## Part 2

### Other necessary software

In order to compile the RMarkdown scripts to *.pdf* documents, RMarkdown requires a LaTeX compiler.

If you are working on the University of Leicester Windows 10 computers, please check that MiKTeX is installed in the *Software Centre*. If not close RStudio if open and install it (this might take a few minutes). Once the installation is completed, re-open RStudio.

MiKTeX should work fine for any other Windows system. On Linux, please install a TeX distribution, e.g. `sudo apt-get install texlive-full` on Ubuntu. On Mac OS, please install a TeX distribution such as MacTeX.

In RStudio, select *Tools > Global Options...* In the *Swave* section, make sure that the following options are selected.

- Weave Rnw files using: *knitr*
- Typeset LaTeX into PDF using: *XeLaTeX*

## Repository

The term “repository” refers to the whole collection of code, data, and other files that compose a project, stored on a version-control system. This project is available on my GitHub ReproducibleResearch repository. Download the zipped repository from BlackBoard or GitHub, or clone the repository using *git* if you are familiar with the tool.

Once downloaded and un-zipped in a folder of your choice, observe the structure of the folder and sub-folders. The *Data* folders contains two scripts that will download some data in the same folder. A back-up version of those data can be found in the *Backup* folder. The *Analysis* folder contains a simple analysis of those data. The *Materials* section contains the lecture and practical session materials, and some additional folders containing the IOSlides templates and the images. The *Utils* folder contains a script to update the git branch connected to the GitHub Pages for this project.

## Load

In RStudio, click on *File > Open Project...* and navigate to the folder where the files have been un-zipped. Select the file *ReproducibleResearch.Rproj* (the extension *.Rproj* might be hidden on Windows, the file type is *R Project*) and click *Open*. RStudio will load the whole project and set the directory where *ReproducibleResearch.Rproj* is located as working directory.

## Build

The *Make.R* script in the main folder can be used to “build” all the scripts in the repository in the correct order. The *Make\_Clean.R* script can be used to delete all the files created by the *Make.R*.

Build the project by running the *Make.R* script. That can be done by loading the file in RStudio and clicking on the *Source* button on bar above the editor panel (top-left area) in RStudio, right side.

Alternatively, you can run the *Make.R* file using the command below from the RStudio console.

```
source('Make.R')
```

## Part 3

Once the project has been built, the *Reproducible\_analysis\_in\_R.pdf* file will be available in the *Analysis* folder. Open the document, read and complete the exercise.

## Close

Once completed the practical, click on the project name *ReproducibleResearch* in the top-left corner in RStudio and then *Close project* to close the project. RStudio might show a prompt asking whether to save the *.RData* file before closing, in which case you can select to close without saving.