Contents

Chapter 1

Welcome!

In this tutorial we will be starting from the beginning, showing you how to set up R, then how to plot and manage your data. Finally, we will introduce modelling in R.

Ideally at the end of this tutorial, you will have learned enough to apply your skills to your own research questions and data.

We don't expect you to retain everything in these tutorials - it's unrealistic and it isn't data analysts work - so our main ambition for the end of the tutorial is you know where to look and to be able to understand what you find. This is important as this means **you can learn R yourself** and as your needs grow, you can solve the problems yourself.

We have written the tutorial in a **problem-orientated style**. Each chapter will introduce a problem and then we will use `R` to solve the problem. We have taken this approach as it should hopefully provide you with a context to approach your own problems with.

This tutorial has data sets which can be found here. There files for this tutorial are available for your personal use too. If you reuse sections of this tutorial, please mention use. Similarly, you can find our biblography at the end.

If you have any comments or corrections, please contact us:

Chapter 2

Setting up

Unfortunately, as with any new skill, the first chapter can be the most text heavy. Apologies. We will do our best to;

- Keep it as brief as possible
- clearly structured so you can skip sections as needed
- complete so you know that after this chapter you will be focused on coding.

2.1 About R and RStudio

`R` is a progamming language that is primarily focused on statistics. This means that it's a way of talking to your computer to get it to solve certain problems and `R`'s speciality is statistics and data analysis.

It's not all it can do and it can be a great pathway to transition from point-and-click data analysis to more general programming. The skills you will learn along the way are part of most modern languages, so your time will be well spent!

There are many reasons individuals use`R` but for our needs the main selling points are;

- Integrated platform for all data management, analysis and graphics
- Allows for reproducible research
- Active community of users providing support
- Cost = Free!

2.1.1 Installing R and RStudio

`R` comes bundled with an interface but not many people use it. The vast majority of `R` users actually use **RStudio** on a daily basis to program and so we too.

There are two options at the time of writing; installing `R` and RStudio on your personal computer (local) or using an online version (cloud). Either is fine. The cloud version is, at the time of writing, free but this is likely to change. Also the cloud version can be slow. Basically, if you are uncomfortable or unable to install programs on your computer, then use the cloud version. Otherwise, the local version is the best choice.

2.1.1.1 Local

Two main steps - install R and install RStudio.

1. Install R
    1. Windows - First follow this tutorial https://youtu.be/G7u7cLiyi8o then this https://youtu.be/fFjHbWhfn6s.

2. MacOS - Follow this tutorial https://youtu.be/1PsPfMaLWSk (you can skip the XQuartz part for now).
3. Linux - Search online for the appropriate tutorial for your distribution.

2.1.1.2 Cloud

As of the time of writing, RStudio is providing a free version of their software that you can run in your web-browser. Go to https://rstudio.cloud and sign up as you would most online accounts.

One thing to note is getting data onto the cloud requires an extra step. You'll need to press `Upload` in the bottom right panel and browse your computer for the files. Also, if you want to upload multiple files, you should upload a zip file.

For our tutorial, go to the GitHub Link. Download `Data` as a zip by pressing the button on the top right and then complete the upload process.

2.2 Navigating RStudio

Open RStudio by either opening the program on your computer as you normally would or by going to https://rstudio.cloud, logging in and then pressing new project.

You will be faced with something like this;

![]($C : /Users/FaheemAshraf/Dropbox(SSD)/Faheem_workings/2018-11-22-RCourseEditing/Bookversionfiles/Boo$

Don't panic! We will explain the most important sections.

The first thing to notice is there are three main sections. Each of these sections has "tabs" in the same way your web browser does.

2.2.1 Console

The left most and largest section main tab is called "Console". This is where you type you R code. For example, type `2+2` and then press Enter. You've just run your R command.

2.2.2 Environment

To the top right there is a single important tab; "Environment" which tracks all the things you've made during your current working time with R.

2.2.3 Packages, Help

The bottom right panel has two panels we will use; packages and help. Both will be covered later but briefly packages are where R extensions are listed and Help is where we can access support documents about R in a convenient way.

2.2.4 Source

We are about to introduce our fourth and final panel. Above the "Console" there is a button called `File`. Do the following; `File -> New File -> R Markdown`. We will talk abou R Markdown in the next session but a quick note; this new section is generally called "Source" because this is the source of the R Code you have written which is then sent to your console.

You can see at the bottom, console has been minimised. As you with any other window, you can use the split and maximise button accordingly.

To recap, the main tabs you need to know are "Console", "Environment", "Help", "Source", and "Packages"

2.3 R Markdown

R Markdown documents allow you to write text around your code and output your work into multipile formats, including a HTML, PDF or Word.

The “text mode” is similar to your experience with typing documents in Word or Notepad. Technically it uses “Markdown”, so you have the option for tables and numbered lists in your final output.

Code in R Markdown is implemented in “chunks”. There are two types of chunks, in-line and blocks. To insert a chunk, you need to press `Insert -> R` located at the top right of the “Source” panel.

Once done, enter the code below and press the green play button on the right hand side of the chunk. In this tutorial, code appears in a light grey box and output in a white box.

Code

```
2+2
```

Output

```
## [1] 4
```

Hopefully when you run the code, you see that 2+2 still equals 4!

The code chunks in R Markdown start with “““**{r}**” and ends with “““”.

Do not modify these lines of the chunk otherwise bad things might happen. Modify anything in the middle.

2.4 Comments

Using `#` in a code block stops the anything after the symbol being run. `#` is referred to as “commenting” code because it’s usually used to add commentry to long pieces of code. Try running the code below.

QUESTION: What do you think the code below will output? Why?

```
#I am trying to calculate the following;
2 + 2 # + 2
```

```
## [1] 4
```

```
# + two
```

2.5 Getting the files

For this tutorial you will need to download a collection of files. These can be found at [Link]. One you have downloaded them, you will need to extract them.