

# LAB 01

ENVX1002

## Table of contents

Getting started: Excel and R . . . . .	1
Before you begin . . . . .	1
Settling in . . . . .	1
AnswerGardens . . . . .	2
Exercises . . . . .	2
1. Equations in MS Word . . . . .	2
2. Introduction to MS Excel . . . . .	3
3. Introduction to R coding . . . . .	8
4. Setting up a project in R and coding in R Markdown . . . . .	14
Summing up . . . . .	18

## Getting started: Excel and R

---

### Learning Outcomes

At the end of this practical students should be able to:

- use Microsoft Word for writing equations
- use Excel and R to calculate simple summary statistics
- Understand the link between R and R Markdown
- Produce your own knitted Markdown document

## Before you begin

---

Make sure you have access to:

- Microsoft Word and Excel
- R and RStudio
- The dataset for today: Lead\_content.csv (download from lab page)

## Settling in

---

In the last two years we have seen big changes, especially in the student experience, making it harder to build connections, even limiting that sense of being part of the University. Everybody deserves to feel a sense of belonging, especially in such an important and exciting stage of their lives.

At the beginning of the next few weeks we will be doing some short activities before getting into the Stats to help you foster a sense of belonging, learn more about your peers, and help better prepare you for your studies. This week we will start with a simple introduction, but before we do this, we would like to acknowledge those who were here before us:

*We would like to acknowledge and pay respect to the traditional owners of the land on which we meet; the Gadigal people of the Eora Nation. It is upon their ancestral lands that the University of Sydney is built. As we share our own knowledge, teaching, learning and research practices within this university may we also pay respect to the knowledge embedded forever within the Aboriginal Custodianship of Country.*

To learn more about why we do Acknowledgement of Country, and the difference to Welcome to Country, see the following page: [Welcome and Acknowledgement](#).

## AnswerGardens

We are all from diverse backgrounds and have followed different paths to get to where we are today. To help you get to know your peers, your demonstrator will lead a class discussion, posting a number of questions on AnswerGarden, where you can then anonymously post your answer to the questions. Links will be provided once your demonstrator has set up the question.

After about 20 minutes of discussion, we can get started on the Stats! Welcome to ENVX1002!

## Exercises

### 1. Equations in MS Word

Specialised software is needed to write out equations when writing documents and reports. One option is Microsoft Equation Editor which comes as part of Microsoft Word.

In a word document click your cursor on the Equation icon outlined in red in the screenshot below.

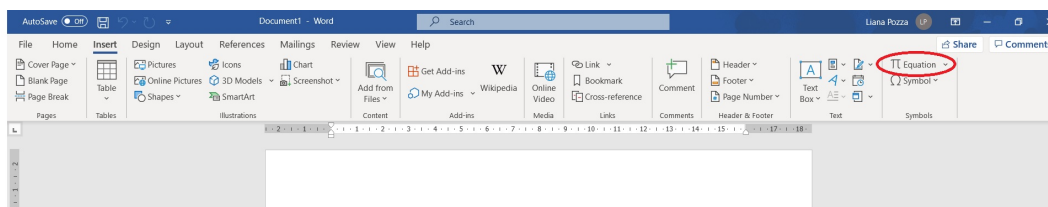


Figure 1: Screenshot of Excel Equation tool highlighted

This will open up a menu (see screenshot below) for writing equations which is quite intuitive for most forms of equations.

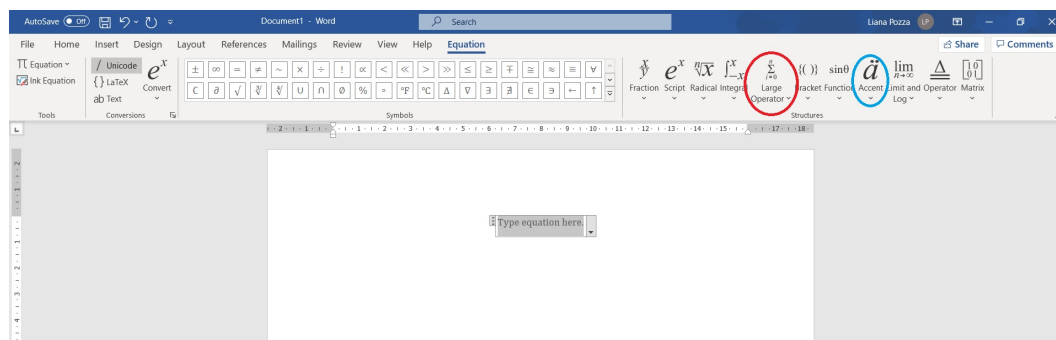


Figure 2: Screenshot of Excel Equation menu with Large Operator & Accent highlighted

The screenshot has outlined some less intuitive parts of equation you will need, the red outline is for equations requiring sigma notation ( $\sigma$ ) such as the population variance, and outline in blue is for equations with accents such y-bar ( $\bar{y}$ ) for the sample mean and y-tilde ( $\tilde{y}$ ) for the sample median.

Use MS Word to type the following equations

1. Mean:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

2. Variance:

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

3. The equation for a straight line

## 2. Introduction to MS Excel

Excel has limited statistical capabilities but is quite useful for storing and manipulating small datasets. Due to the global dominance of MS Windows it is also the most commonly used format for storing and distributing data within workplaces.

**Excel worksheets and cells** Excel files come in series of worksheets where data is stored in cells. The columns are given letters and the rows are given numbers, enabling a particular cell to be referenced by a combination a letter and number. In the screenshot below the number 2 in the orange cell could be referenced by **B3**. In a blank worksheet type **2** in the **B3** cell.

In cell **C3** type **=B3**

The equals sign tells Excel you are calculating something or referring to a cell. You should now have 2 in cell C3.

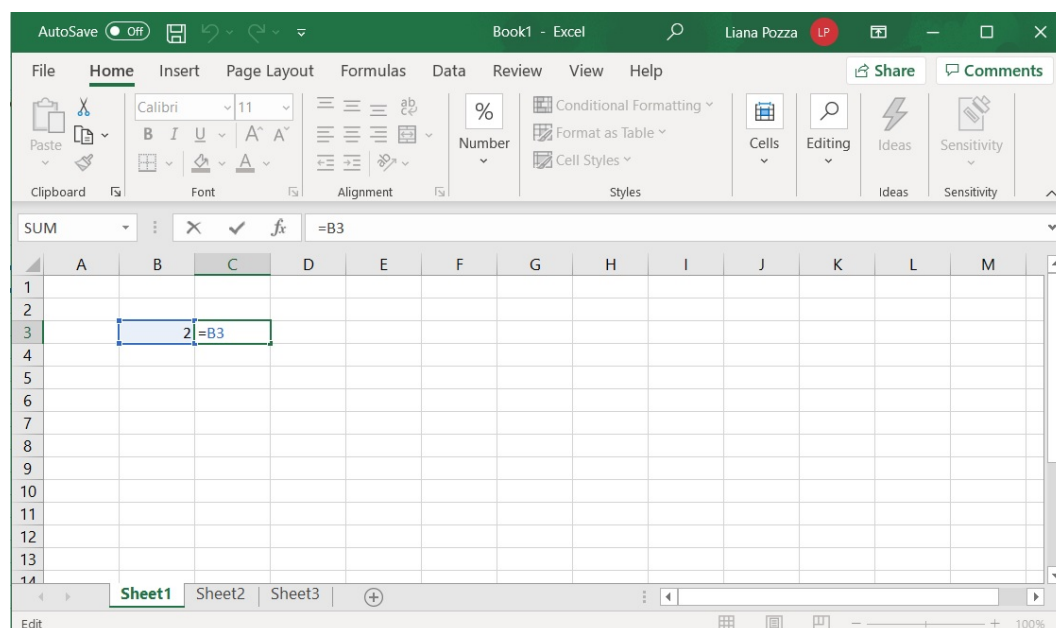


Figure 3: Screenshot of Excel Typing number 2 in cell B3

At the bottom of the Excel page you will see references to each of the worksheets in the file, for example 'Sheet1', 'Sheet2', 'Sheet3'. This enables you to store multiple datasets in the one file. In this unit the datasets for each exercise will be stored in separate worksheets but in the same file.

**Basic arithmetic** When typing equations, make sure you start by typing = . This tells Excel you want to solve the input equation.

The basic arithmetic operators return numeric values:

Key	Operation
+	Addition
-	subtraction
*	Multiplication
/	Division
^	Exponentiation

These can be used in combination with numbers or cell references.

For example, to get a value of **4** in cell **D3** you can type either

=2\*2 (type numbers)

or

=B3\*C3 (reference cells)

It is better to reference cells so that if you change the values the same equations can be applied.

**Basic functions** Some basic functions are:

Function	Operation
SUM	Sums a range of cells
COUNT	Counts a range of cells
LN	Natural Log
EXP	Exponent

These can be used in combination with numbers or cell references.

For example in cell **E3** you can type either

=EXP(4) or =EXP(D3),

Another example is in cell **F3** type

=COUNT(B3:D3)

Note that Excel has an auto-complete function that allows you to select from a list of functions after typing the first letter i.e. =C. Selecting the function gives a brief description of what the function does.

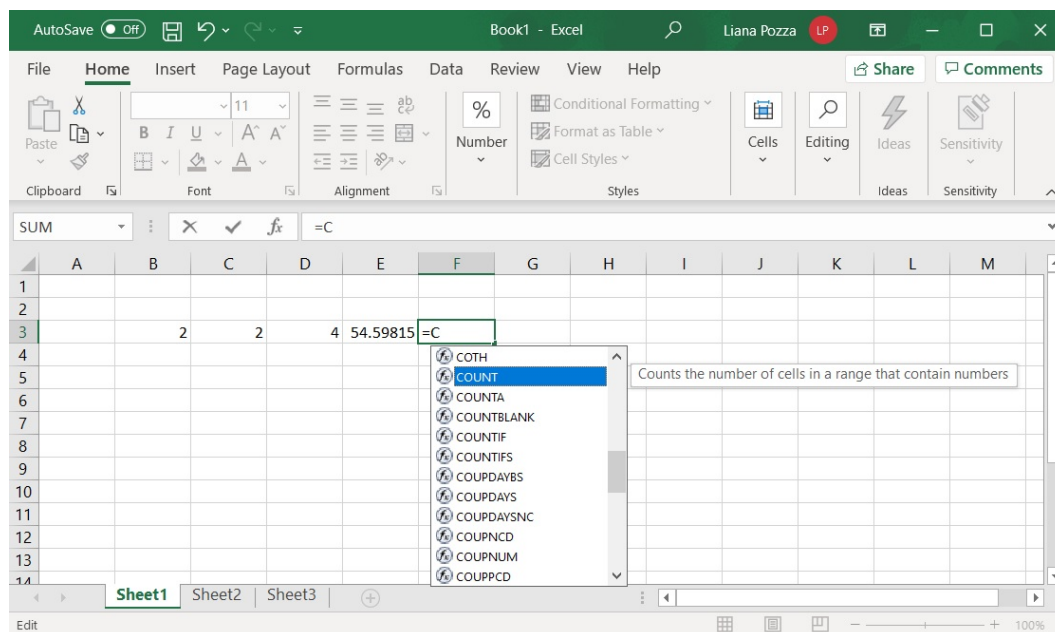


Figure 4: Screenshot of Excel Typing =C to show function description

Once the function has been selected, you can proceed to type the opening bracket and enter in the cell reference, cell range or numeric value. Excel aids you in showing what the required input is as you type the opening bracket (see image below). The square bracket indicates an optional value, in this case if only one cell is selected **=COUNT(E3)**, then the function will return the value **1**.

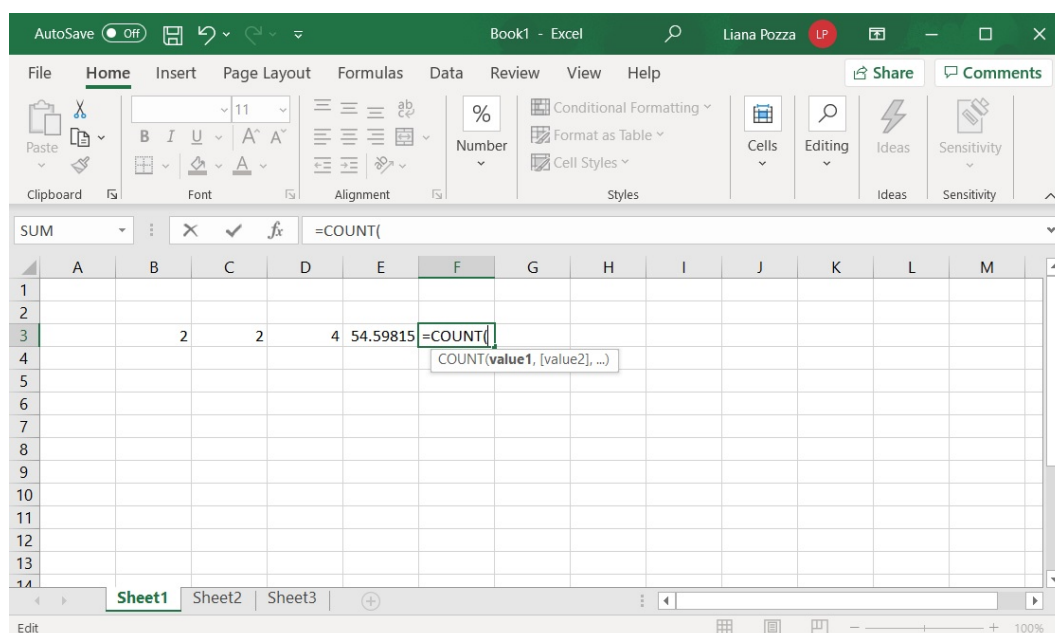


Figure 5: Screenshot of Excel showing required input arguments for COUNT function

**Simple summary statistics** There are functions for calculating summary statistics in Excel. Click on a cell where you want the answer to be entered and then use the menu by **Formulas » Insert Function**. A screenshot for calculating the sum is shown below.

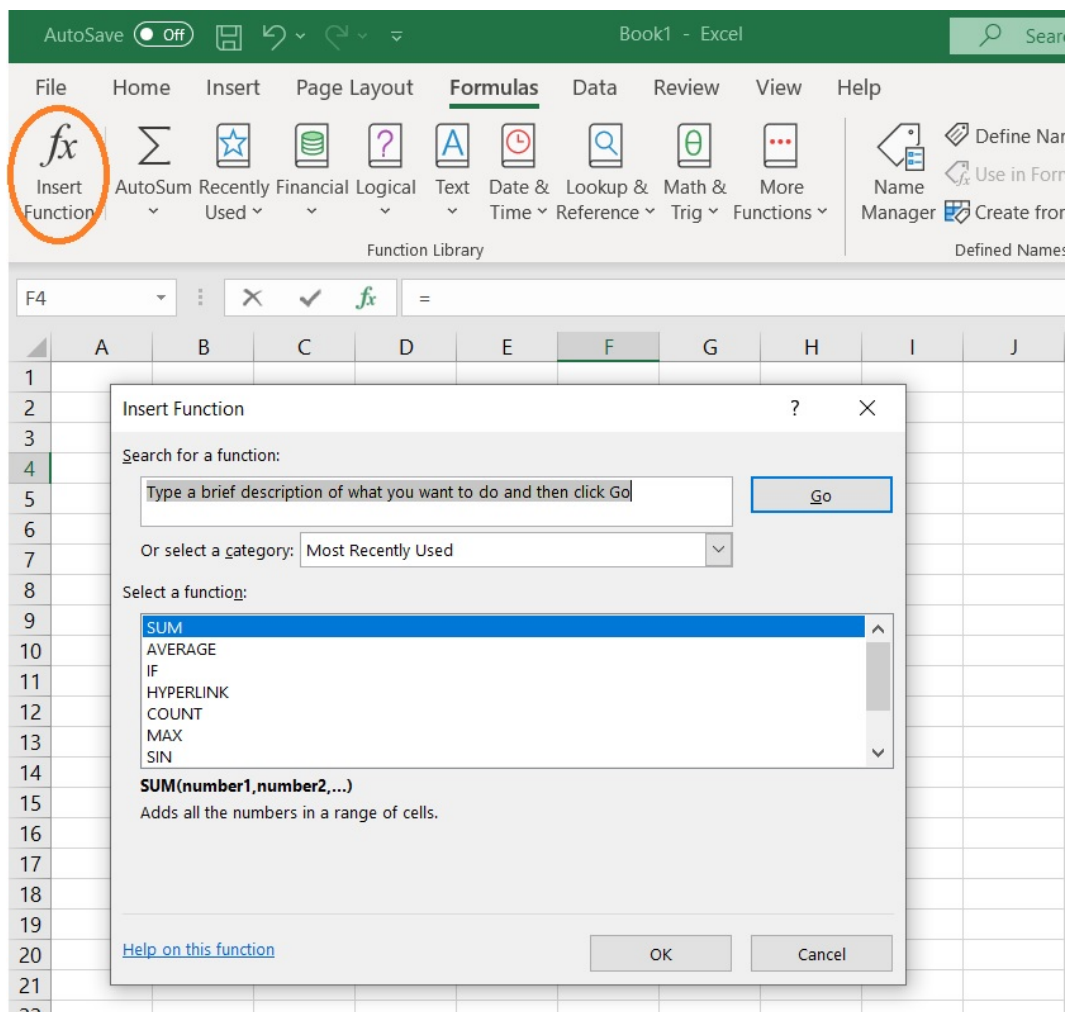


Figure 6: Screenshot of Excel Insert Function

On the next screen you can then select the cells where the observations are located from which then median will be calculated.

After a while you should get to know the name of the functions in Excel and be able to write the arguments in manually. In the screenshot below the function is **MEDIAN** and it refers to cells between (and including) **A2** and **A9**. A row or column of cells can be represented by the starting cell, then colon, then final cell (**A2:A9**).

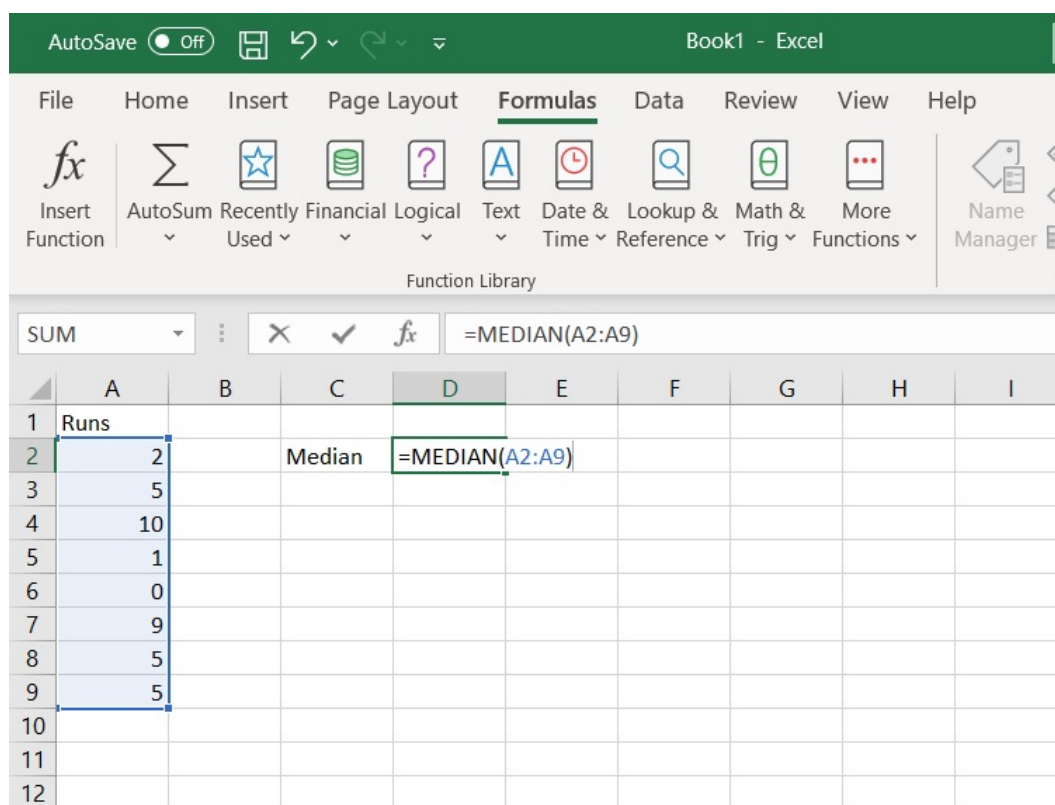


Figure 7: Screenshot of Excel calculating the Median of cells A2 to A9

Some of examples of the functions that can be accessed in Excel are shown below. Note that the .S and .P extensions for variance and standard deviation are from later excel versions.

Statistic	Function
Minimum	=MIN
Maximum	=MAX
Arithmetic mean	=AVERAGE( )
Median	=MEDIAN( )
population variance	=VAR.P()
Sample variance	=VAR( ) or =VAR.S()
Population standard deviation	=STDEV.P()

**Calculating simple summary statistics in Excel** In this exercise you will use the Lead\_content.csv (download from lab page)

This data was collected from a recreational parkland in Sydney and is a measurement of the lead concentration (mg/kg) detected in the soil, measured through chemical analysis (ICP-OES). There are a total of 60 samples collected from around the park. The park was originally a municipal landfill but remediated in 1990, so we expected to find low levels of lead. The guide value set by the Australian Government is 300 mg/kg and this is where further investigation is needed (potential to cause harm).

In excel, calculate the following:

- minimum value
- Maximum value
- mean

- median
- range
- sample variance
- sample standard deviation

From these statistics,

1. Were there any samples higher than the guide value?
2. Were there any samples where no lead was detected?
3. What is the mean value?

### 3. Introduction to R coding

R is a statistical programming language that can be used to store, manipulate, visualise and analyse data. It contains a number of pre-defined analysis techniques but you can also program your own methods.

R is open source which means that you can examine and modify the raw software code if you like. A worldwide collective of scientists, programmers and statisticians are working on improving and extending the capabilities of R.

Of immediate value to you is that it is free so for the rest of your career you can keep using R without burdening future employers with software licence fees. An article on the merits of R can be found by at (<http://monkeysuncle.stanford.edu/?p=367>).

This module will provide an introduction to R. Work through the examples by typing in all code you see to familiarise yourself with the syntax of R and some of the commands available in R.

**Getting a copy of R** R is free and downloadable from (<https://cran.r-project.org/>). This screenshot below shows the site where you can download the installation needed for your operating system.

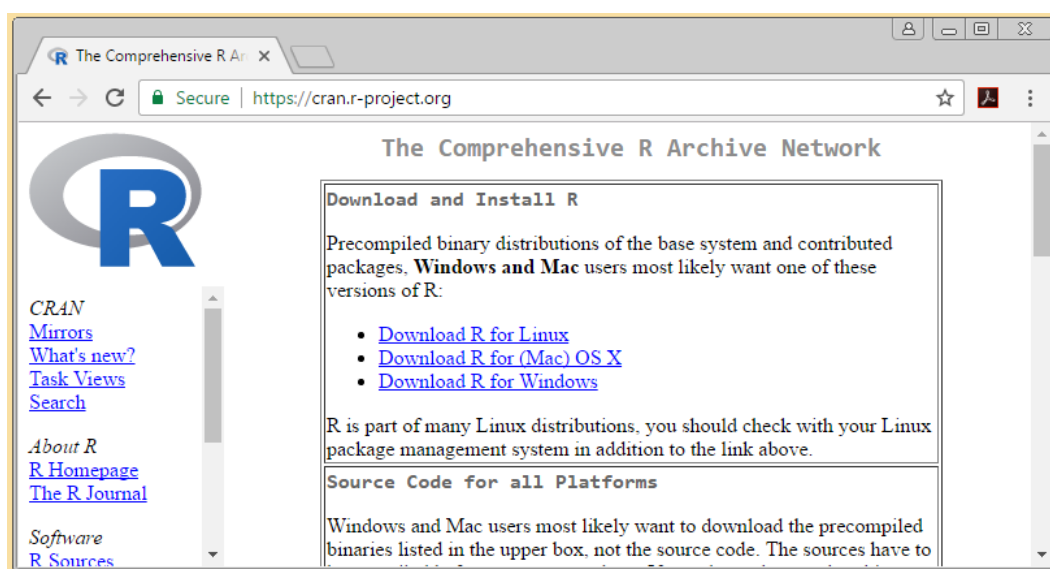


Figure 8: CRAN website



**R basics** To begin open R from the program files menu on your computer. When you open R, a window containing the R console will open. It will look slightly different depending on the operating system you use. The screenshot is from a Mac.

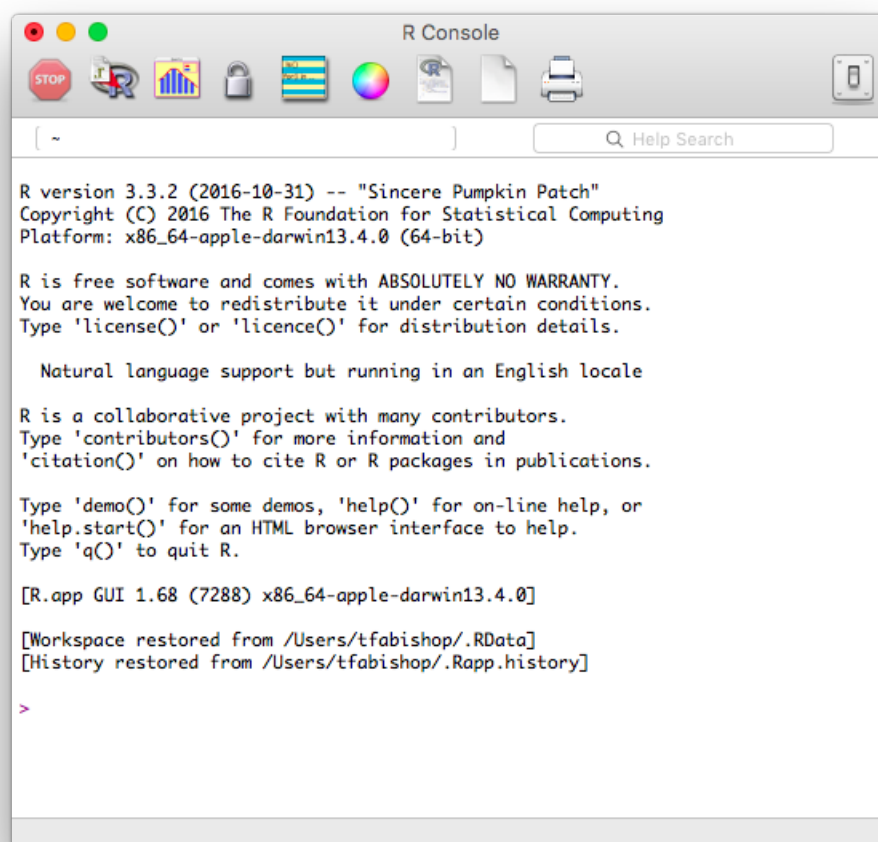


Figure 9: The R Console

At the bottom of the screen is the command prompt `>`. Commands are typed at the command prompt and followed by the **ENTER** key.

From now type all the command you see into R. If you type in an expression, when you hit the **ENTER** key, the expression will be evaluated and the result returned, lets add 5 and 5 together.

```
5+5
```

```
[1] 10
```

R is an object-oriented programming language and the basic unit in R is called an object. Objects can store single numbers, columns of data, modelling output, functions and other kinds of information.

The class of the object determines the way in which commands are executed on an object and the way in which data can be stored by the object. For example, vectors store a single column of data, a matrix can store multiple columns of data and a data frame can store multiple columns of data where the columns may be of different data types (e.g. numeric and logical).

When you type in an expression and hit the **ENTER** key, the expression is evaluated and the result returned:

```
5+5
```

```
[1] 10
```

Alternatively, the value returned by calling a function in R can be saved as a named object using the assignment operator `<-`.

```
myData <- 5+5
```

You can view the contents of an object by typing the object name:

```
myData
```

When you hit the **RETURN** Key, you will see the following output:

```
[1] 10
```

Object names can be made up of letters, numbers and `,` and `_` symbols. A name must start with `.` or a letter. If it starts with `.` the second character must not be a number. R is case sensitive, so calling `mydata` is not the same as `myData` and will generate an error:

```
mydata
```

Error in eval(expr, envir, enclos): object 'mydata' not found

To see a list of all the named objects you've created in R, use the `objects` function:

```
objects()
```

```
[1] "myData"      "pandoc_dir"  "quarto_bin_path"
```

To delete an object, use the `remove` function:

```
remove(myData)
```

If you type and enter an incomplete command, a continuation prompt will appear on the next line: `+`. You can continue typing the command followed by the **ENTER** Key.

```
myData3 <-  
+ 8
```

To cancel a command at the continuation prompt (or during execution of a command), press the **ESC** key. The up and down arrow keys can be used to scroll through previous commands. Comments can be indicated by a hash mark (`#`) - everything on the line following the hash mark will be ignored by R. This can be after R code on a line or on a separate line as shown below.

```
#I am adding 6+6 and saving it to an object called my.Data  
my.Data <- 6+6
```

**Basic arithmetic** The symbols for basic arithmetic operators are shown in the table below.

Operators	Operation
+	addition
-	subtraction
*	multiplication
/	division
^ or **	exponentiation

Parentheses ( ) can be used to specify order of operations.

You can perform basic calculations by typing expressions into the command line.

```
(5*10) ^2
```

```
[1] 2500
```

Better still you can assign results to a named object to be used at a later date.

```
myresult <- 20/10 + 6 - 1
```

For example, We can then halve the value of myresult.

```
myresult <- myresult/2
```

**Text editors** Until now you have copied and pasted commands which you may wish to use again. This is particularly important as you begin to write series of commands to perform a certain task. One option is to save these commands to text files and copy the relevant commands into R as needed. You can save logical groupings of commands into different text files.

A better option is to use text editors – one example is RStudio and unlike notepad it allows syntax highlighting of R commands. When an R session is open, RStudio includes an additional menu and toolbar and it allows the user to interact with R by submitting code in whole or in part. It is available from (<https://www.rstudio.com/products/rstudio-desktop/>).

From now on you should start to use RStudio by copying the commands into a R file and then submitting them to R. By doing this you will have a record of the commands you have used. From now you will be using R through RStudio and not the console. Over time you will develop a library of code to perform analyses and create graphics.

The screenshot below shows RStudio, the top left window shows your code and the bottom left window shows the input and output in R. The top right hand window side shows the objects you have created, for example myData. The bottom left hand corner shows graphics, in this case a histogram, but can also show other useful features such as the help menu.

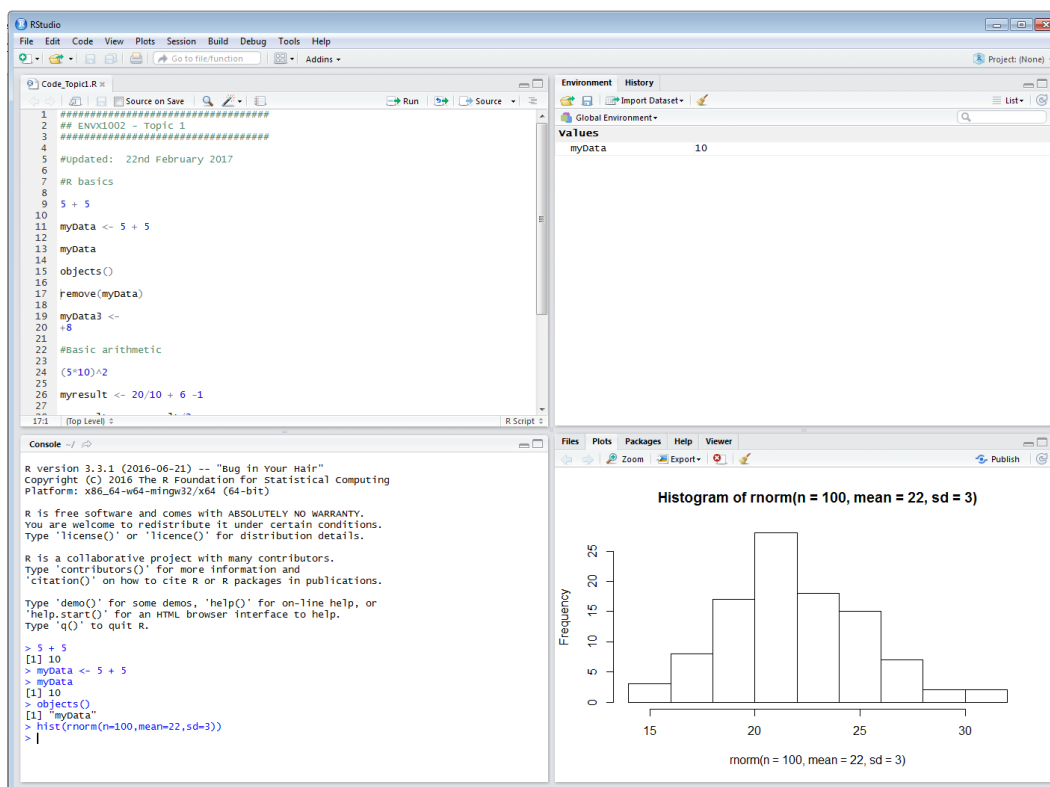


Figure 10: R Studio

**Simple data analysis** Go to the eLearning site and download and open the file **Code\_Topic1.r** and you will see most of the commands you have been typing. Rather than typing commands into R we will now start to use R Studio. You will see the commands you entered until now are all there.

You send code from your open R file which is in the top left hand corner of your R Studio program to the R console which is found in the bottom left hand corner of the R Studio. The output, e.g. the mean, will also appear in the bottom left hand window.

To send a line of code to R from your R file (the **Code\_Topic1.r** file), click anywhere in the line and click on the Run icon or use the short cut **CTRL+ENTER** (short cut for Windows only). Then R studio will move to the next line in your script. The output is shown in the bottom left hand window.

You can also use the **#** symbol to write comments which R will ignore. It is recommended you copy the output into your R file and comment it out so you have a complete record of your work.

**Getting data into R** R can accept data from many different sources; for example directly from scientific instruments or even scraping the internet for data. In this topic we are only considering small datasets so we will enter the data manually via the keyboard.

A vector (or list) of numbers can be manually entered using the assignment operator and the **c** function which essentially means combine, an example is below.

```
myDataset <- c(5,12,52,32,14,6.1)
```

Use the **c** function to enter a soil carbon dataset (48, 56, 90, 78, 86, 71, 42) as an object called carbon. We will then calculate some basic statistics on this dataset.

**Summary statistics** Now we have entered the data in R we want to do something with it, such as calculate summary statistics. R functions behave differently depending on the data type. Some functions will work only on specific data types, other functions will use different methods on different data types.

To find the mean of dataset we use the `mean` function.

```
mean(Carbon)
```

```
[1] 67.28571
```

Other commands related to summary statistics include:

- `median` - median
- `var` - sample variance
- `sd` - sample standard deviation
- `min` - minimum value
- `max` - maximum value
- `length` - number of observations (length of the vector)

Calculate all of the statistics above using R.

```
median(Carbon)
```

```
[1] 71
```

```
var(Carbon)
```

```
[1] 355.5714
```

```
sd(Carbon)
```

```
[1] 18.8566
```

```
min(Carbon)
```

```
[1] 42
```

```
max(Carbon)
```

```
[1] 90
```

```
length(Carbon)
```

```
[1] 7
```

Rather than using all of these individually you can use the `summary` function which gives the minimum, maximum, mean and median values. We will consider the 1st Qu. and 2nd Qu. in the next practical.

## summary(Carbon)

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
42.00	52.00	71.00	67.29	82.00	90.00

Note, that it does not calculate the standard deviation, variance or number of observations.

That is all we will do in R in this practical. Remember to save your code so you can refer and reuse this for your online quiz and other assessments in the future.

### 4. Setting up a project in R and coding in R Markdown

Working smartly with your data and code can save you time and tears. We are going to work with projects as this helps manage your data in a clean and easy way. You can either decide to create a new project for each lab or create one project for the course.

**Project** Your tutors will assist you to get your files organised:

- Set up a course folder called ENVX1002files on your desktop/network drive/USB.
- Set up a project called ENVX1002 in your folder.

The screenshots will help you do this (Figures 1-5).

1. Create a folder called ENVX1002 on your laptop or the class computer. If you are using a class computer you will need to save the file to a usb, upload to cloud or email to yourself at the end of the class.



Figure 11: Folder

2. Open R Studio go to the file dropdown menu and select New Project

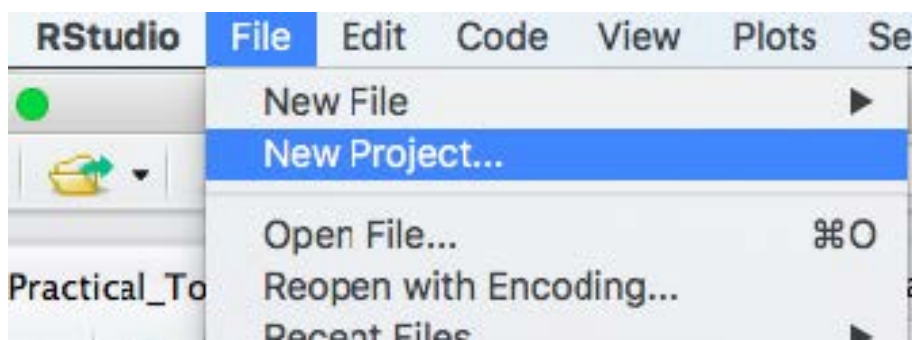


Figure 12: New project

3. Select New directory if you did **not** do step 1 or Existing directory if you did do step 1.

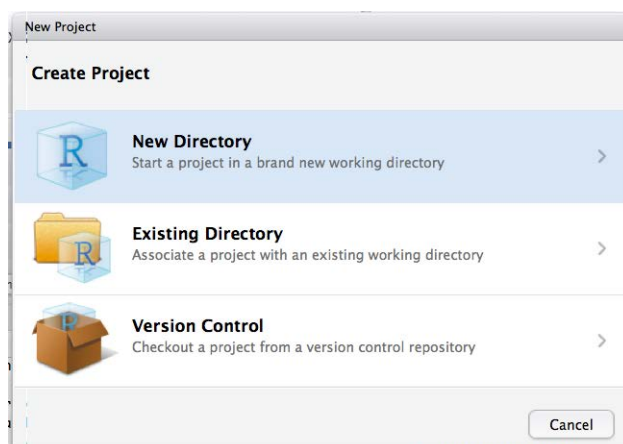


Figure 13: New project

4. Only do this if you did **not** do step 1. To create a new Directory brows to an appropriate folder. Name the Directory something like ENVX1002.

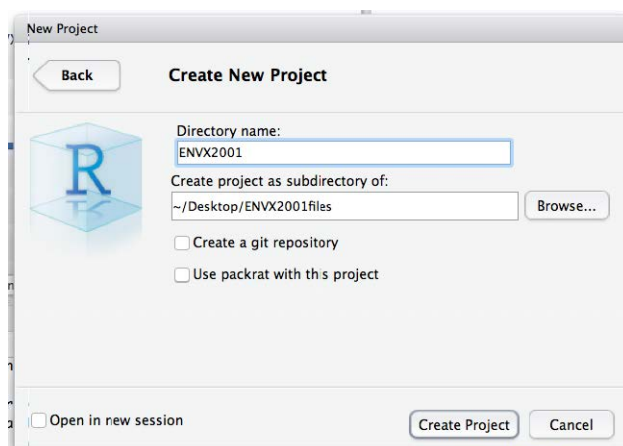


Figure 14: New project name

5. Well done! you have now set up a project.

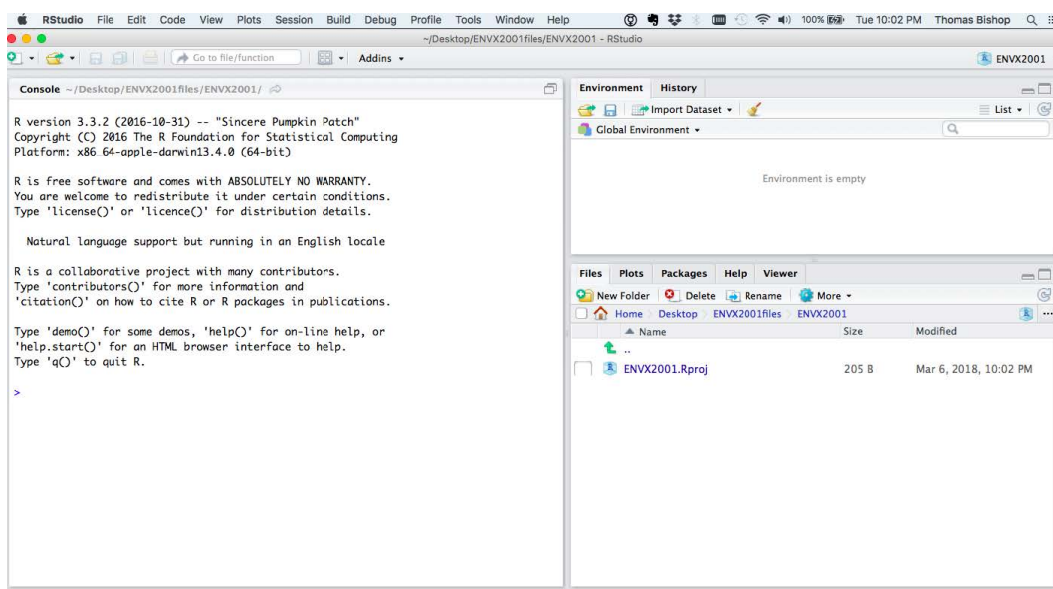


Figure 15: New project name

**Coding in Quarto** Your tutor will assist you to open a Quarto (.qmd) file. You will do this for each Lab and your also for your reports:

- Open a qmd file, and save it as, for example, Prac1.qmd.
- Run the file using knit.
- View Prac1.html in a browser.
- Now experiment with editing the file (both text and code chunks) and then re-run using knit.
- Use this file to store your summary of today's lab work.

The screenshots will help you do this.

### 1. Navigate to **New File > Quarto Document**

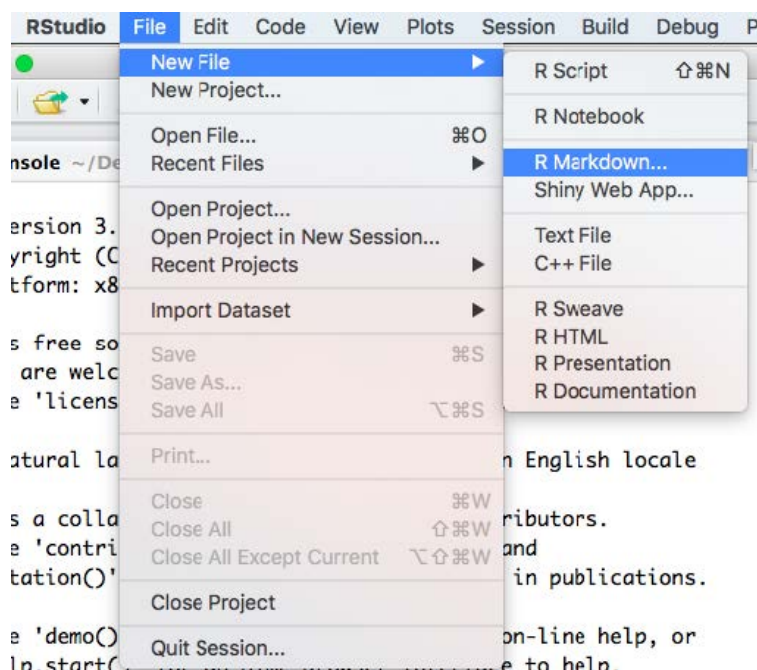


Figure 16: New Rmarkdown



2. Enter in an appropriate name for your file such as ENVX1002\_prac\_1 and enter in your name. Select HTML format.

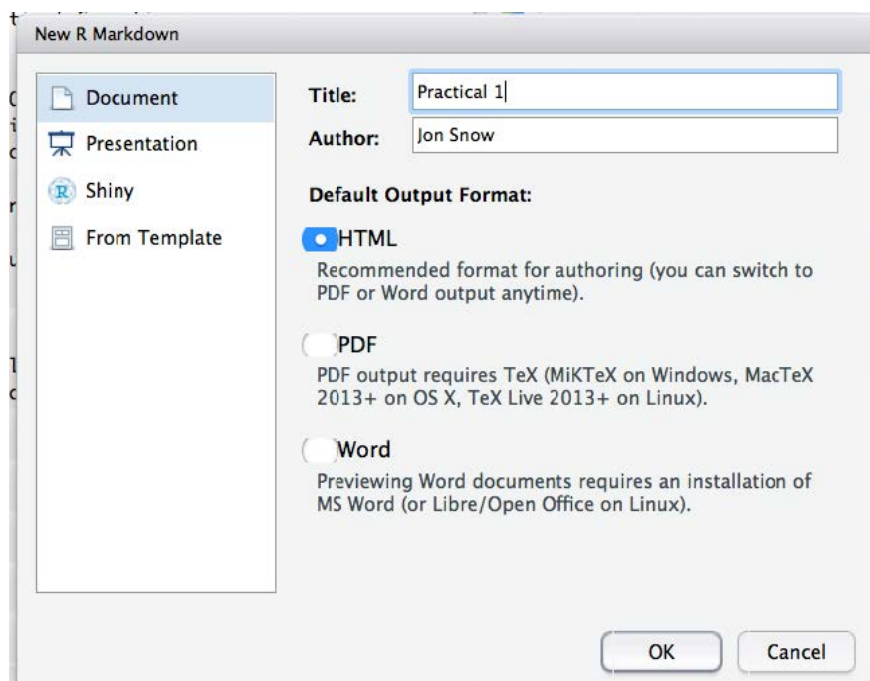


Figure 17: New Rmarkdown

3. Well done! you have created a new R Markdown file. First save your file by navigating to the File menu i.e. **File > Save as**.

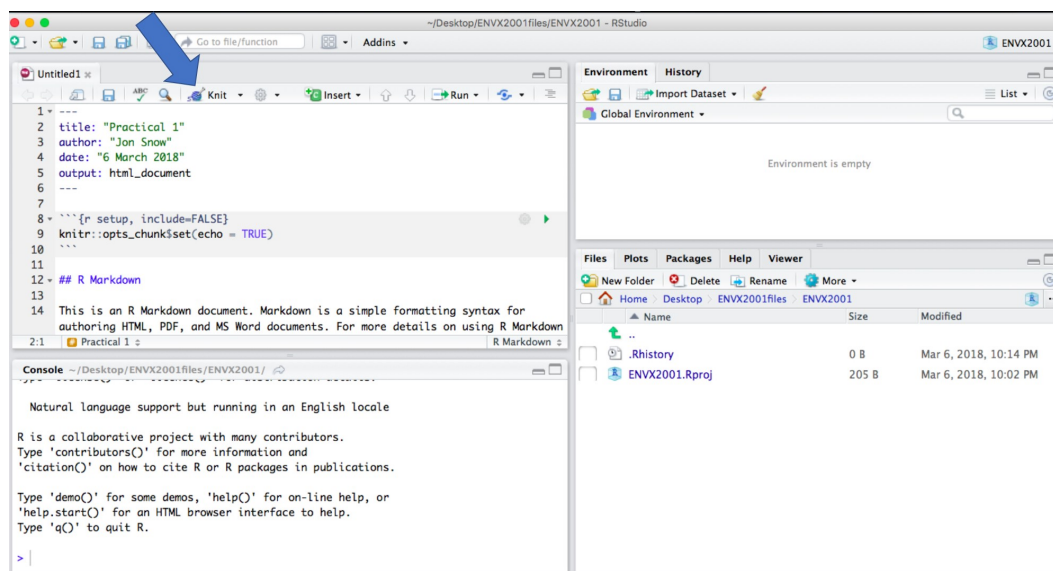


Figure 18: New Rmarkdown

4. Finally you can **Knit** the file by selecting Knit indicated by the blue arrow in the previous figure. A really good exercise is to have the code and generated html code side by side and see what each bit of code has generated.

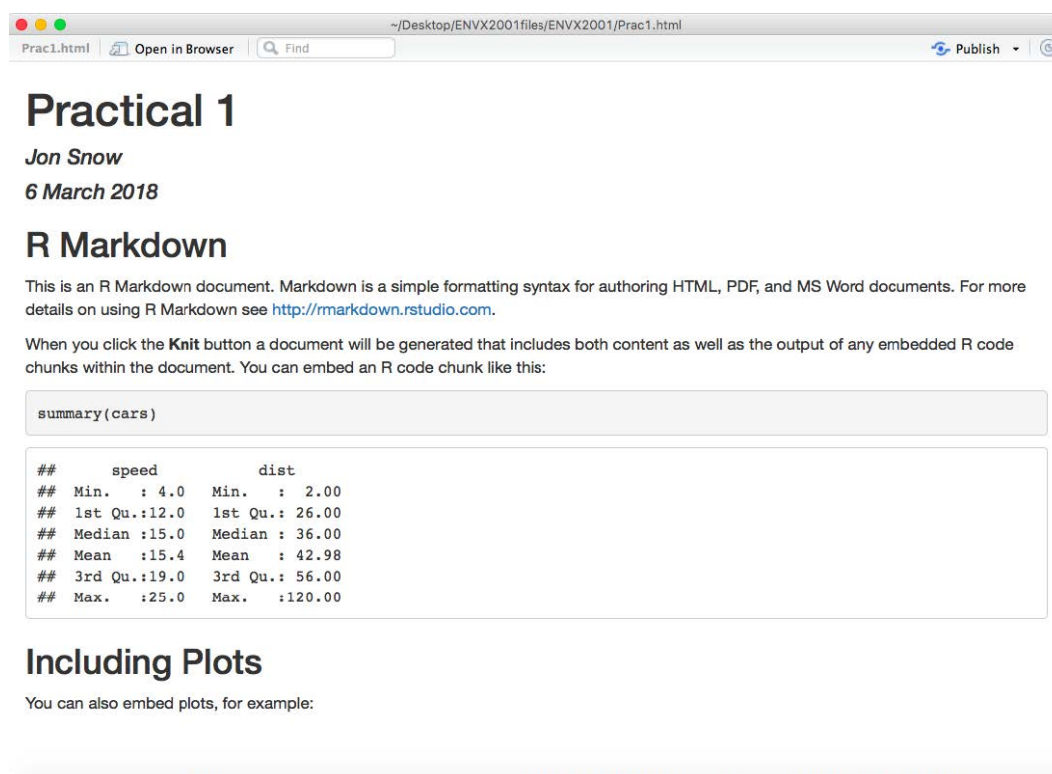


Figure 19: New Rmarkdown

## Summing up

Well done!

You now know how to:

- write equations in a word document
- do basic operations in Excel
- do basic operations in R and set up a project as well as generate an R Markdown File

To do by next week:

- complete anything you have missed from today's lab
- the [O-quiz](#) (Available from March 1st)