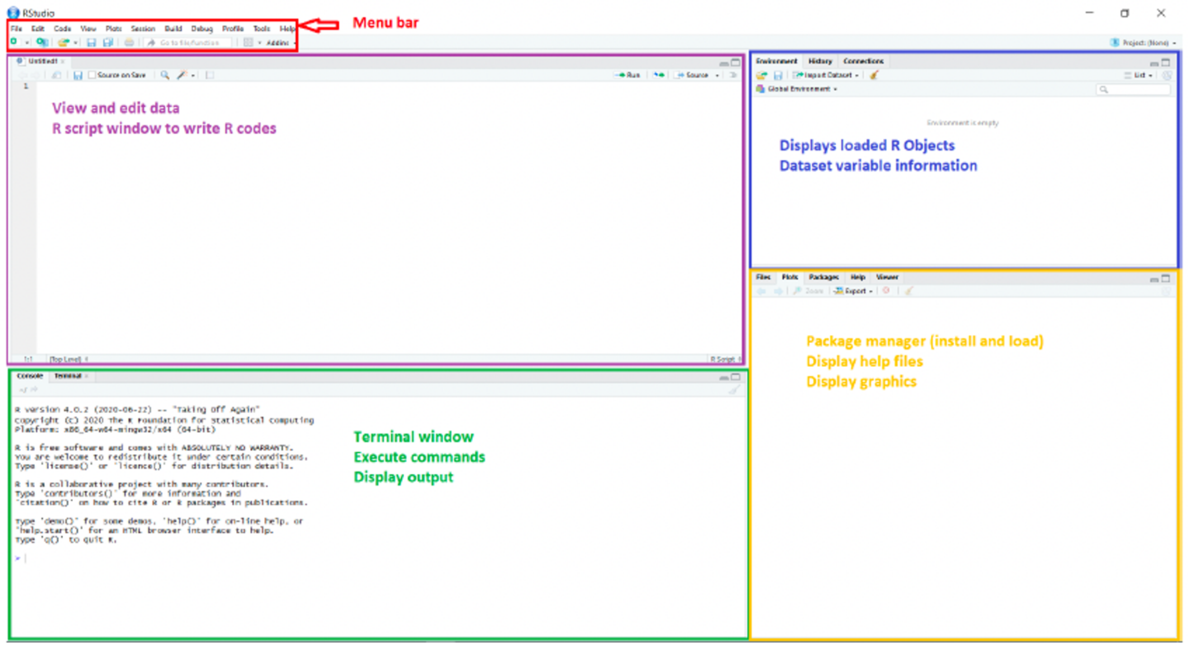
# BS2280 - Econometrics
# R Workshop 0 - Your first Step

## Exploring RStudio and Searching

RStudio has four main windows, that often have more than just one purpose. Figure 1 provides a brief description of each window. We will use all of them during our module, but the most important ones will be the console and editor pane.


Menu bar
View and edit data
R script window to write R codes
Displays loaded R Objects
Dataset variable information
Package manager (install and load)
Display help files
Display graphics
Terminal window
Execute commands
Display output

```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

## Preparing your workspace

Before you start working with R, you should undertake the following steps to facilitate your workflow:

**Step 1. Create a folder.**

Before every session, create a folder on your computer to save all your **worksheets, datasets and solutions**. Make sure you remember where that folder is and that you know the path to that folder.

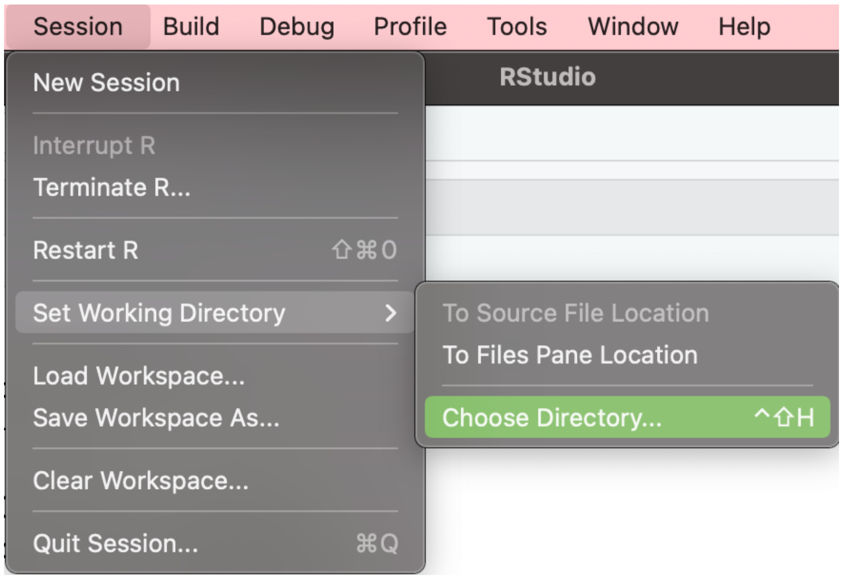
Give your folder any name that makes it easy to remember what it contains, e.g. 'RWorkshop1' in your 'BS2280' folder.

If you use university computer, create a folder on the university network drive (H:), so that you can access your data on any PC on campus.

**Step 2. Open R and set your working directory.**

This step is to tell R to use the folder you created. R will save all files in there and, if you want to open a dataset, R will also look in this folder first.

Menu bar → Click Session → Set Working Directory → Choose Directory → **Select your created folder in step 1**

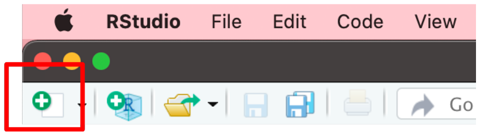


**Example**. Suppose I create a folder called RWorkshop1 under Teaching BS2280 on my desktop, and select folder RWorkshop 1 when set working directly, the console window will show[1]:

```r
setwd("~/Desktop/Teaching/BS2280/RWorkshop1")      # for Mac users
setwd("C:/Teaching/BS2280/RWorkshop1")             # for Windows users
```

**Step 3. Open an R script file before you start with your analysis.**

Click on the little white sheet with the white plus over a green circle to open an R script:



R script file is just a text file, which can help us to save codes.

**Step 4. Type the codes in R script file.**

Type ths codes in R script and you can add comments to your commands through the usage of a hash tag, so that you have commentary about your work and you won't forget what you have done previously. The words after a hash tag will not be executed, for example

```r
5+2          # Summation
```

However, by just typing these commands into the script file nothing is actually happening. If you want R to execute any of the commands in your script file you have to do one of the following:

1. Press the "Run" button, in which case R will run the command in which the mouse curser currently is

[1]This is an example. You may have given your directory a different name. Besides, this is a Windows user example, Mac users will get slightly different results

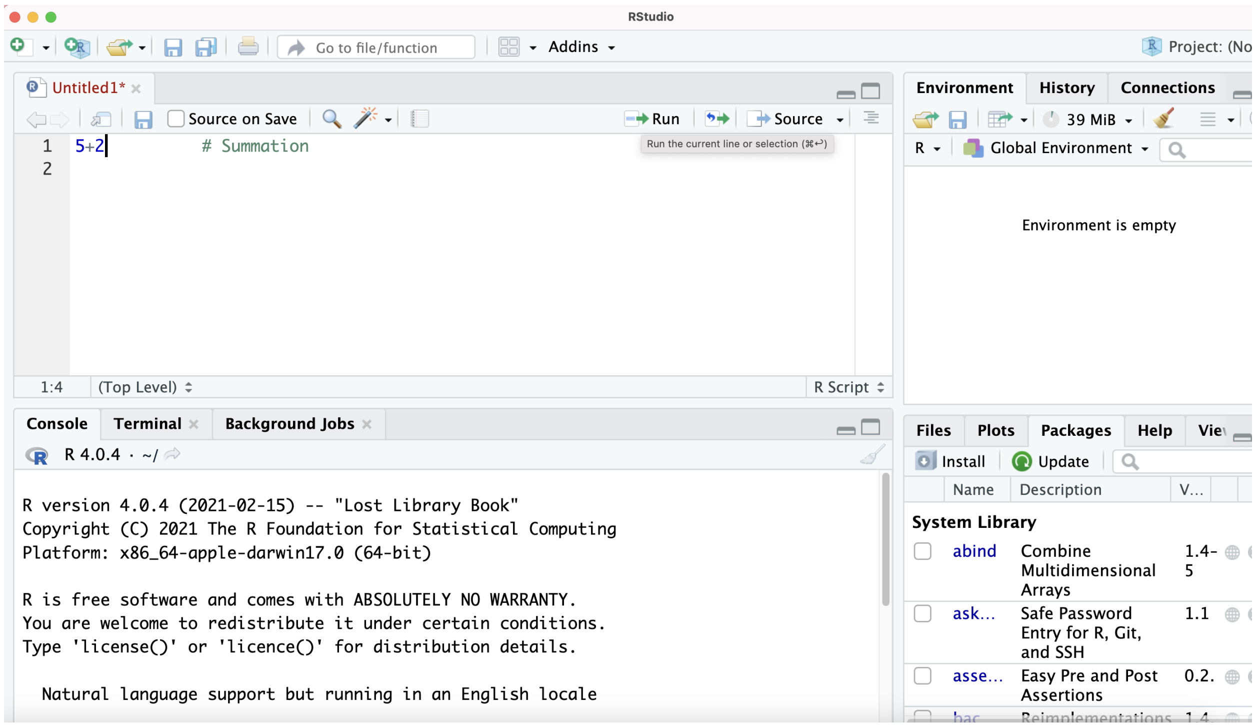
2. Press the "CTRL"+"ENTER" on the keyboard, in which case R will run the command in which the mouse curser currently is

3. Highlight several lines and press "CTRL"+"ENTER" on the keyboard, in which case R will run the commands in the highlighted lines

**Example**

```
5+2          # Summation
```

Type the above command in the R script and put the mouse cursor next to the command



```
5+2          # Summation
```

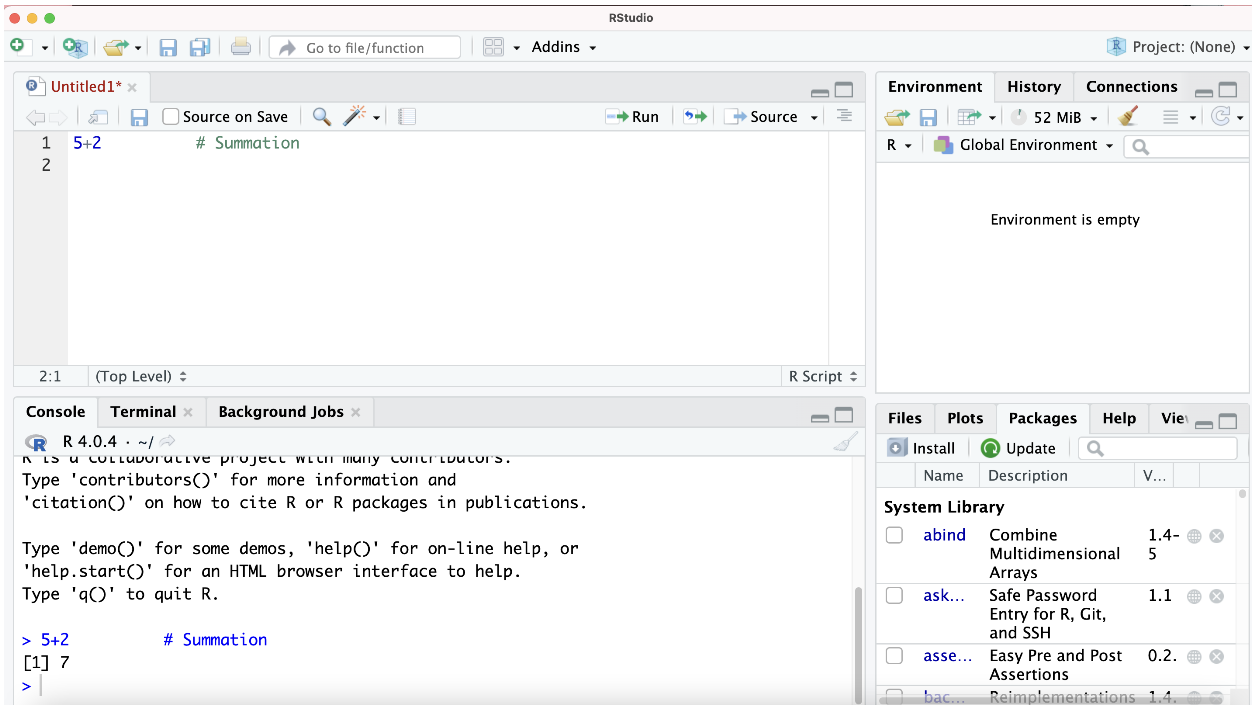
```
R version 4.0.4 (2021-02-15) -- "Lost Library Book"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale
```

| Name | Description | V... |
|---|---|---|
| abind | Combine Multidimensional Arrays | 1.4-5 |
| ask... | Safe Password Entry for R, Git, and SSH | 1.1 |
| asse... | Easy Pre and Post Assertions | 0.2. |
| bac... | Reimplementations | 1.4 |

Click Run and you will see the Console window shows the command that has been executed. Here, R has calculated 5+2 and give you the result 7.



```
5+2          # Summation
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 5+2          # Summation
[1] 7
>
```

| Name | Description | V... |
|---|---|---|
| abind | Combine Multidimensional Arrays | 1.4-5 |
| ask... | Safe Password Entry for R, Git, and SSH | 1.1 |
| asse... | Easy Pre and Post Assertions | 0.2. |
| bac... | Reimplementations | 1.4. |

With some practice, you will realise that using the keyboard instead of the mouse will be much more time-efficient.

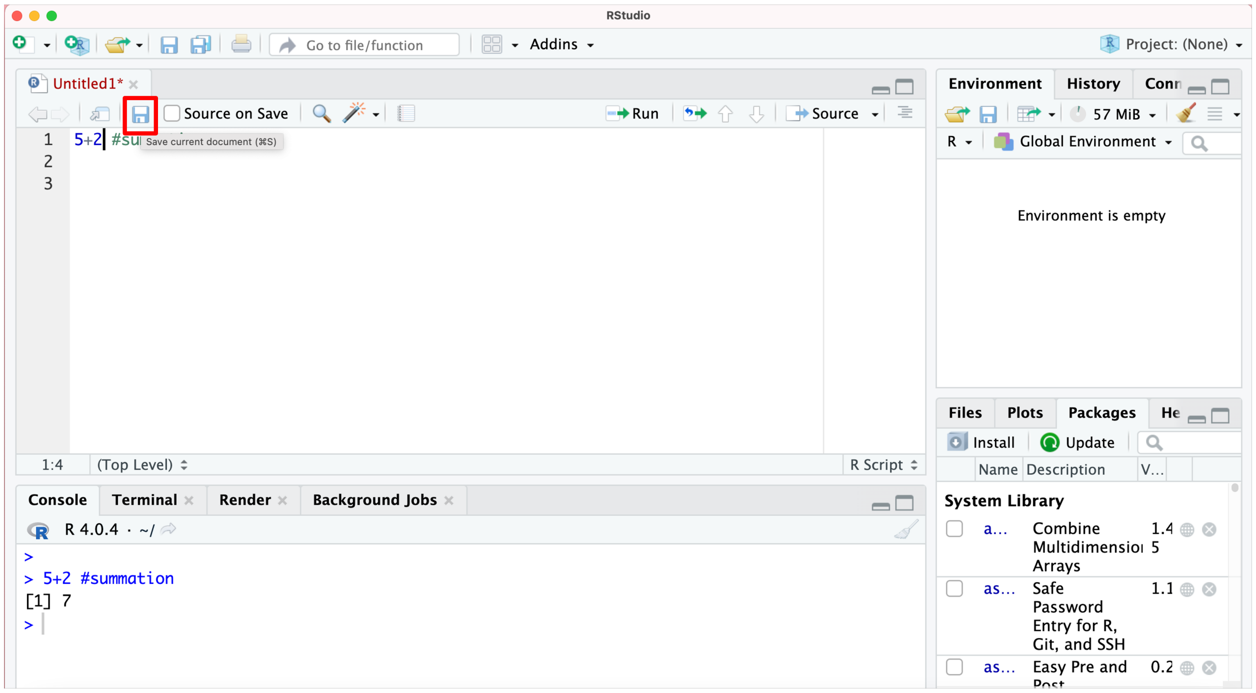
### Step 5. Save your work.

Different ways exist to save your work.

- Save as an R script

Save the file in .R format. The R script will contain all your commands you have used in the current session and allows you to replicate what we have done in the future.

Click the disk icon. In the new window, name the R script file and save it in the folder you created in the step 1 (e.g. RWorkshop1).



```
5+2 #su
```

```
>
> 5+2 #summation
[1] 7
>
```

For more complex tasks that require you to reproduce your work, use R scripts. R scripts, which are saved in '.R' format, are like a written recipe of your work which allows you to reproduce your work.[2] If you use R scripts, you do not have to save your workspace image. I recommend to create an R script for each of your workshops.

- Save as a workspace image

If you save your session as a workspace image, all your objects you have created will ba saved in a '.RData' file. You can load this work image file to continue working where you have left off. You can either save it via the menu, or use the command line:

```
save.image("MyRsession.Rdata")
```

Here, I saved the workspace image under the name 'MyRsession.Rdata'. R will also automatically create a '.history' file that is saved alongside the workspace image. You can open it with any text editor, such as Notepad, Emacs, Vim, Pico or Nano. It contains a line-by-line history of all the commands that have been executed in the console window for the associated session.

## Using R as a calculator

Let's start with some basics and use R as a calculator. All the standard mathematical operations can be undertaken with R. Type these codes in your R script file and click Run!

```
5+2          # Summation
```

```
## [1] 7
```

[2]If you have used Stata before, R scripts are equivalent to Stata do-files.

```r
5*2        # Multiplication
```

```
## [1] 10
```

```r
5/2        # Division
```

```
## [1] 2.5
```

```r
5^2        # Power
```

```
## [1] 25
```

```r
sqrt(5)    # Square Root
```

```
## [1] 2.236068
```

## Assigning objects

You can also store any of your results as an R object and use it for further calculations. Type these codes in your R script file and click Run! For example, let's assign the result of a simple mathematical calculation to the object x:

```r
x <- 5*2
x
```

```
## [1] 10
```

```r
X <- 3*2
X
```

```
## [1] 6
```

Note that x and X are different variables, so R is case-sensitive!

We assign a value to an object with '<-' ('smaller than' sign followed by a 'minus'). To print[3] the value of the object, we just type its name into the command line. Let's assign the square of x to a new object y:

```r
y <- x^2
y
```

```
## [1] 100
```

[3]'Print' is used synonymously for 'display', so you don't have to worry that your printer will switch itself on!

## Installing and Loading R Packages

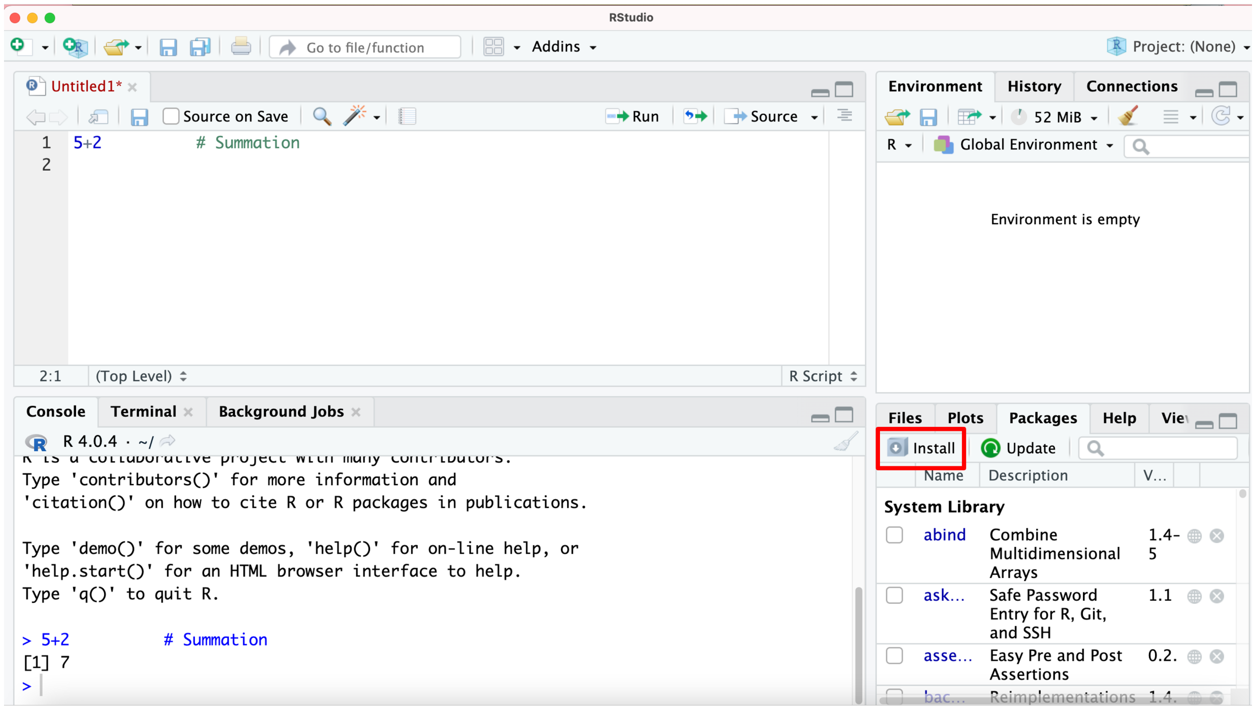
The base installation of R comes already with an impressive number of in-built commands, e.g. the sqrt() command that we have used in the previous exercise. However, for more specialised statistical techniques, we have to install and activate R packages that are developed and maintained by a large number of developers of the R community.

For instance, if we intend to use the "read_excel" function for importing Excel data in R, it's essential to note that "read_excel" is stored in a package called "**readxl**." Therefore, to make use of "read_excel," you must first install and activate the "**readxl**" package. Once activated, you will have access to the "read_excel" function within the "**readxl**" package for data importation.

The easiest way to install packages is through the RStudio package window (that's the window with the yellow border in Figure 1).

Click on the Packages tab and you will see all the packages that are already installed. To activate them, tick the box next to the package name.

If the required package is not there, you can install it via the Install tab. In the newly opened window, just type in the R package name and click 'install'. To update the installed R packages click on the Update tab that is just next to the Install tab.



```
5+2          # Summation
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 5+2          # Summation
[1] 7
>
```

| Name | Description | V... |
|---|---|---|
| abind | Combine Multidimensional Arrays | 1.4-5 |
| ask... | Safe Password Entry for R, Git, and SSH | 1.1 |
| asse... | Easy Pre and Post Assertions | 0.2. |
| bac... | Reimplementations | 1.4. |

You can also use the command line to install, activate and update packages. Assume you want to install the R-package readxl:

```r
install.packages("readxl")
```

After the installation we activate the package with:

```r
library("readxl")
```

**Key point:**

Install package: just do once

Activate package: do it every time you need

This means that you have installed the "**readxl**" package this time. In future instances, there is no need to reinstall it. However, you will need to activate it by typing the command **library("readxl")** every time you want to use the "read_excel" function stored in it.

To update all installed packages, type:

```
update.packages
```

Finally, if you should ever struggle with some of R's commands, a look into R's help-files can be very helpful. To access the help file, you have to type into the console window '?' and then the command name. For example, if you want to know more about the command 'getwd', type the following:

```
?sqrt()
```

The quickest way to close R is to use the command line: q()

Now you are ready for the first data exercise.