**Institute
and Faculty
of Actuaries**

E PERITIA RATIO

# R   Contents

# Contents

# Get up and running with

## ⓘ Who is this for?

This is a guide to help students, members and users with the set up and running of R and RStudio software.

## ⓘ What is R?

R is a freely available, open source, statistical programming language that has become increasingly popular in a variety of disciplines. Used in academia and industry, it provides a diverse and powerful toolset for the visualisation, processing, analysis and reporting of quantitative data.

## ⓘ What is RStudio?

RStudio is a freely available, open source, integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

## ⓘ Why two pieces of software?

R is the programming language and execution environment necessary for anyone that wants to run R code.

RStudio is a popular way to write and run R code due to its ease of use and collection of handy convenience tools.

## ⊙ Do I need to install both?

---

Students are advised to prepare for the CS1B – Actuarial Statistics and CS2B – Risk Modelling and Survival Analysis exams with both R and RStudio.

It is necessary to install R first, followed by RStudio which will automatically detect R and prepare to use it.

# How to install R

## Downloading and installing R on your computer

The most common way of using R is by downloading the R program from its website and installing and running it just like any other program.

R can be downloaded from the official website of 'The R Project for Statistical Computing', the organisation that maintains and updates R. Their website is known as the 'Comprehensive R Archive Network', or CRAN for short. The URL is http://www.r-project.org/.

## How do I download R?

**Step 1**  Go to → https://www.r-project.org/

**Step 2**  At the main page, click on the 'download R' link which appears in the 'Getting Started' section at the top of the page.

**Step 3**  Choose a CRAN mirror.

> ⊙  A mirror is essentially a copy of the available downloads at different locations.

> ⊙  We recommend choosing the https://cloud.r-project.org/ mirror which appears under the location '0-Cloud'. This is a cloud-distributed mirror which will automatically perform the download from a suitable server.

> ⊙  You then select the operating system you will be downloading R to: Linux, (Mac) OS X, or Windows. For specific information for each major operating system see the relevant section below.

# R website homepage

This software is downloaded worldwide. As we have already mentioned, a 'mirror site' provides the same files for your approximate location, which makes your download quicker.

## Windows

**Step 1**     To automatically be taken to the download page for  the RStudio cloud mirror go to ↓
https://cloud.r-project.org/bin/windows/base/

**Step 2**     At this point you will be presented with a **Download R for Windows** option with the most recent version number.

**Step 3**     Clicking this link will download an executable **(.exe)** file which can be used to install the R software.

**Step 4**     Double click on the downloaded executable and work  through the installation prompts.

## (Mac) OS X

Step 1     Got to https://www.rstudio.com/products/rstudio/download/.

Step 2     Click on the Download button under the RStudio Desktop FREE option.

Step 3      Under Installers for Supported Platforms choose RStudio - Mac OS X 10.6+ (64-bit).

## Linux

R is supported on many distributions of Linux with detailed installation guides maintained by the CRAN R project authors.
For  a direct link to the instructions on installation of R with your chosen Linux distribution go to ↓
 https://cloud.r-project.org/bin/linux/

# Install RStudio

## Windows

**Step 1**   Go to ↓
https://www.rstudio.com/products/rstudio/download/.

**Step 2**   Click on the 'Download' button under the RStudio
Desktop FREE option.

**Step 3**   Under Installers for Supported Platforms choose
RStudio – Windows Vista/7/8/10.

## Linux

RStudio is available on many distributions of Linux.

**Step 1**   Go to ↓
https://www.rstudio.com/products/rstudio/download/.

**Step 2**   Click on the 'Download' button under the RStudio
Desktop FREE option.

**Step 3**   Under Installers for Supported Platforms choose the
installer that matches your chosen Linux distribution.

## (Mac) OS X

Step 1    Go to https://cloud.r-project.org/bin/macosx/.

Step 2    Choose the latest version of the '.pkg' file suitable for your version of MAC OSX.

Step 3    Install the downloaded file, keeping all default settings.

Step 4    If you have OS X 10.9 (Mavericks) or higher, download and install XQuartz from
http://xquartz.macosforge.org/. This step should not be necessary for OS X versions
below this.

# Installation at work

Installation of the R and RStudio software may require certain
administrator privileges on your machine. If you are unable to
install the software directly yourself, your IT team should be able
to help you if you provide them with a copy of this guide.

# Basic Use - RStudio

RStudio IDE  is a user-friendly way to write, store, manage and
run all your R code. This section will get you started on running
your first piece of code in RStudio.

## Hello World

**Step 1**      Start an RStudio session.

**Step 2**      Select File → New File → Rscript. By writing all our
code in an Rscript in the text editor, rather than
directly into the console, we can save it, edit it and
reuse it later.
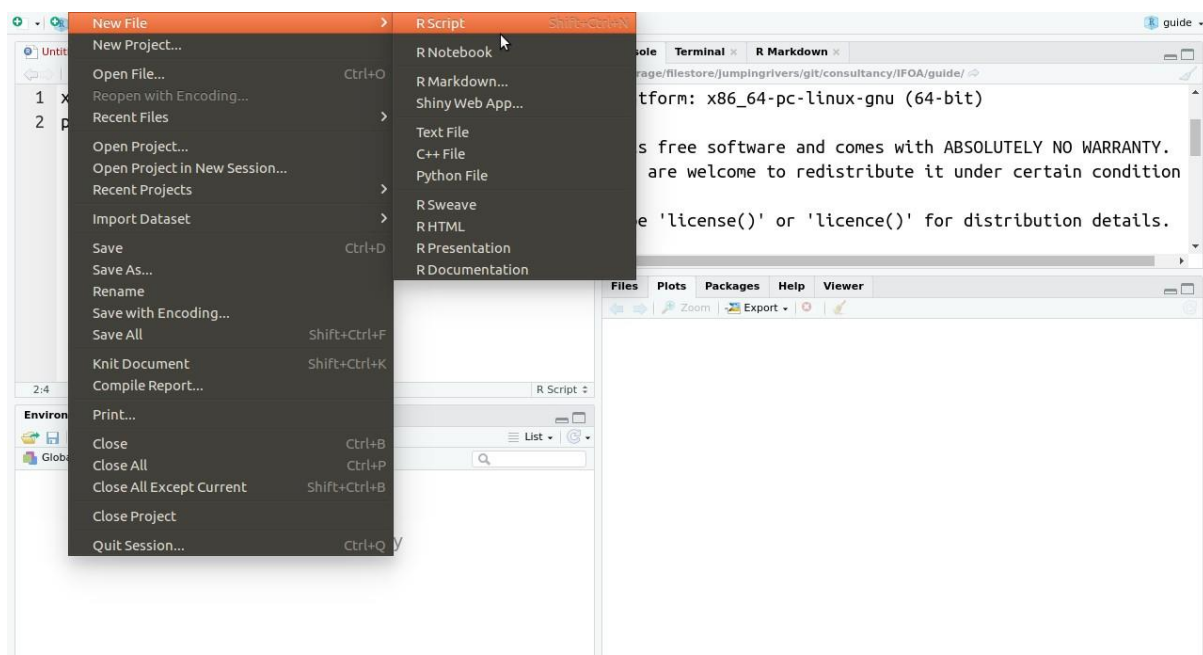
**Step 3**     Type:

```
x = "Hello World"
print(x)
```

       into the Rscript in the text editor.

**Step 4**     The text editor itself doesn't run any code, that is the job of the console. To run code in the editor, highlight the code you would like to run and click **'run'** at the top right of the script pane. (Alternatively, press **'Ctrl + enter'** on the keyboard.) If no code is highlighted, the console will run the line of code where the cursor currently is.
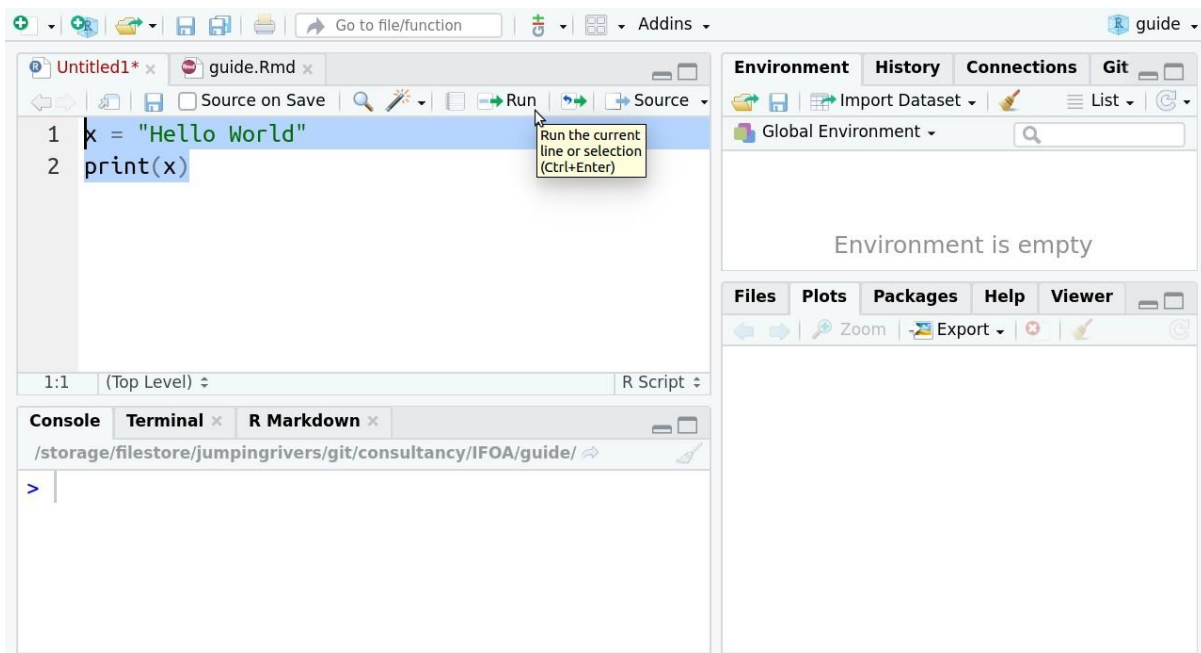


You have successfully created your first variable and used your first function to print out the value of that variable.

# R packages

Most of the available functionality in R is stored in packages. R takes care of package management, so you will install new R packages directly from the R program.  Some packages will be available by default in the System Library, such as:

- stats
- utils
- graphics

To enable these packages simply tick the check box and the package will be loaded to your session as shown below.
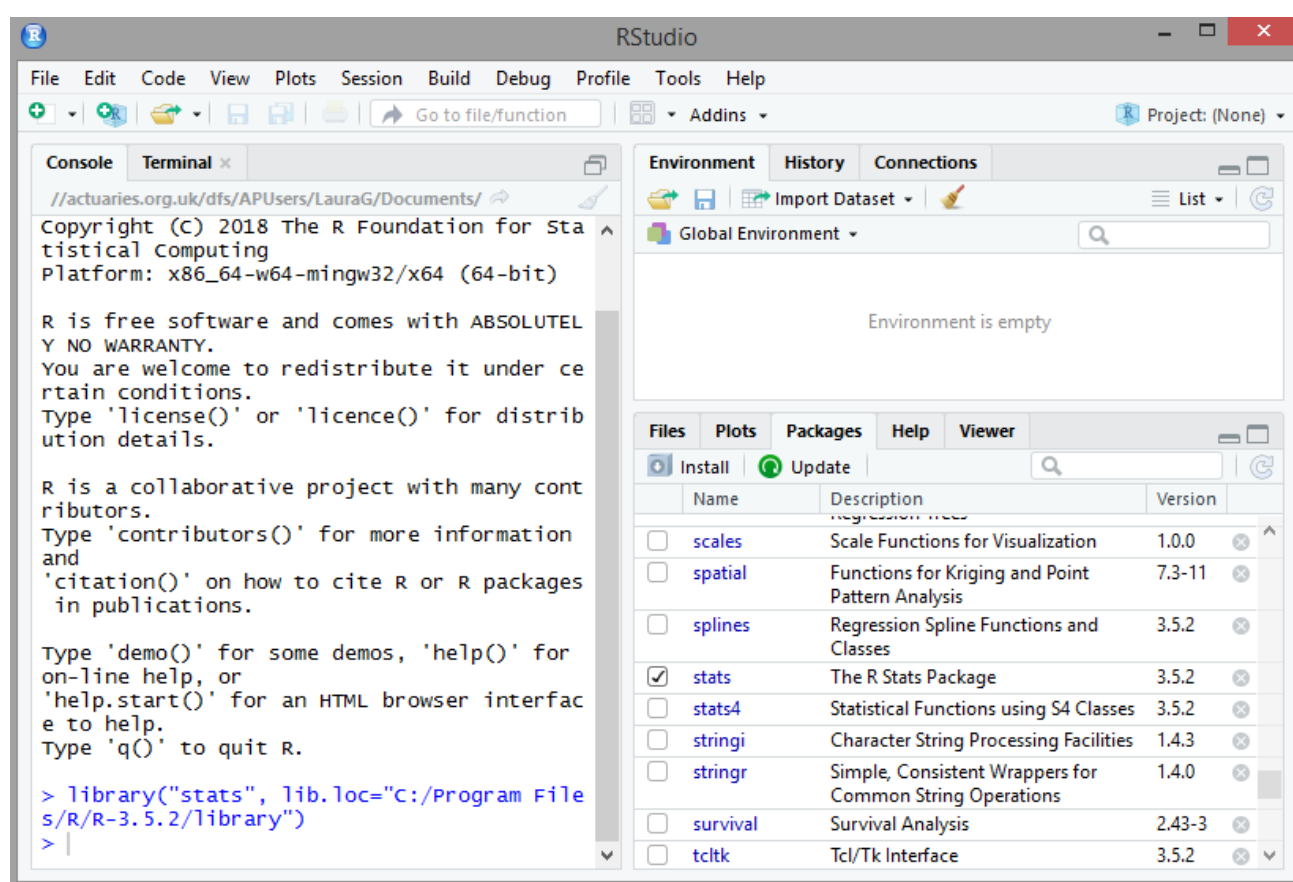


Some packages will not be listed in the System Library and will need to be installed.

## Example installation

As an example, we will run through the installation of a popular package for graphics, '**ggplot2**'.

Note: package names in RStudio are case sensitive, care should be taken to install correct packages.

There are two methods for installing R packages in RStudio.

## GUI method

**Step 1**     Click on the 'Packages' tab.

**Step 2**     Click 'Install'.
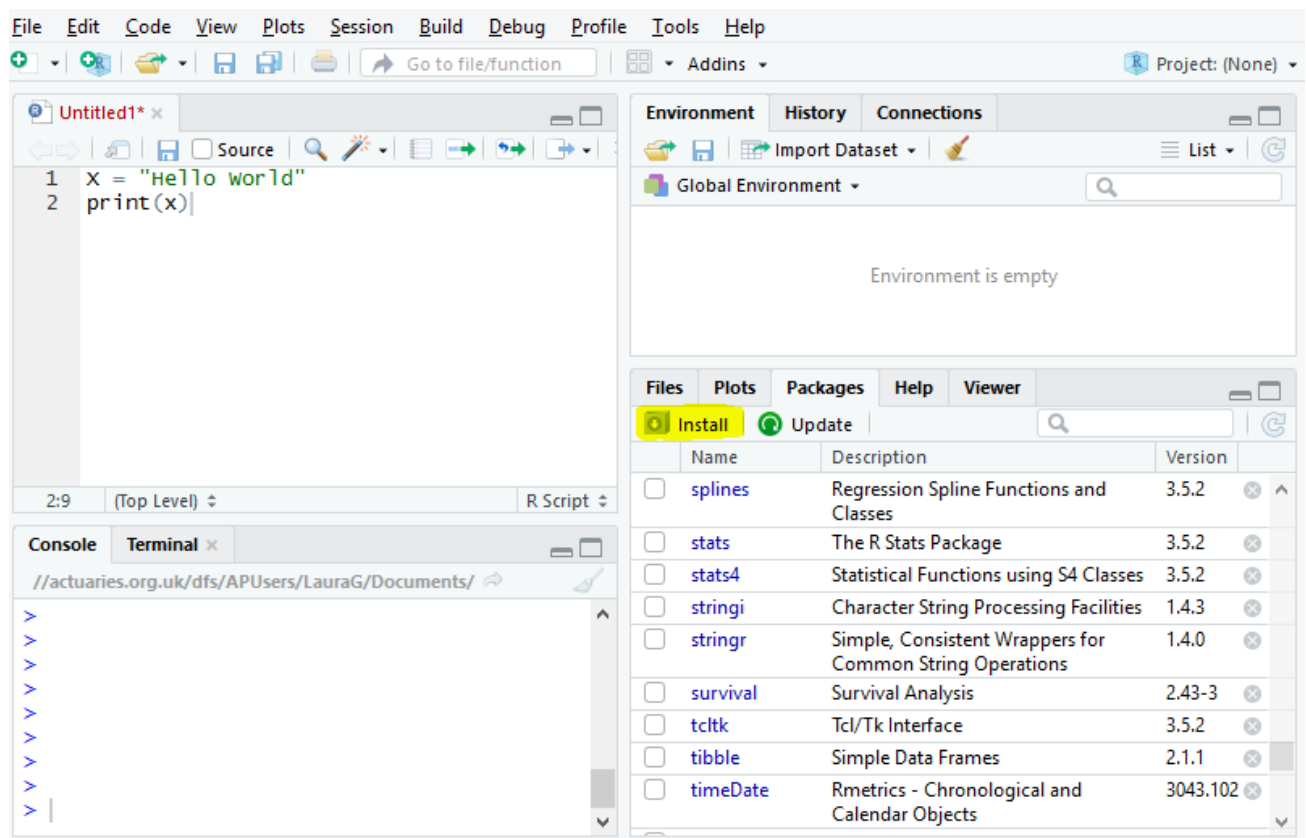


**Step 3**     In the resulting popup window, type the name of the
package `ggplot2` under 'Packages'.
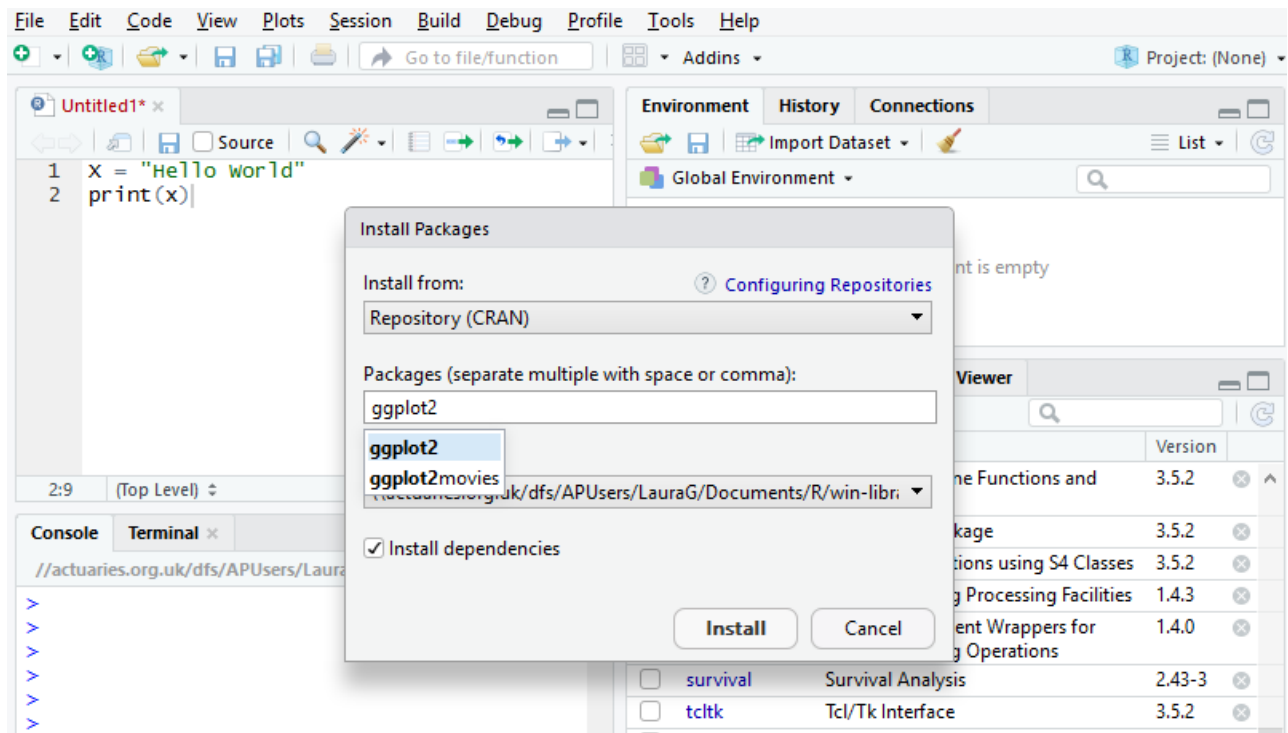
Your package install status will be shown in the console window:



## Using code to install packages

It is also possible to install packages in the console.

Go to the console window, as shown above

and write the following code and press  enter.

```
install.packages("ggplot2")
```

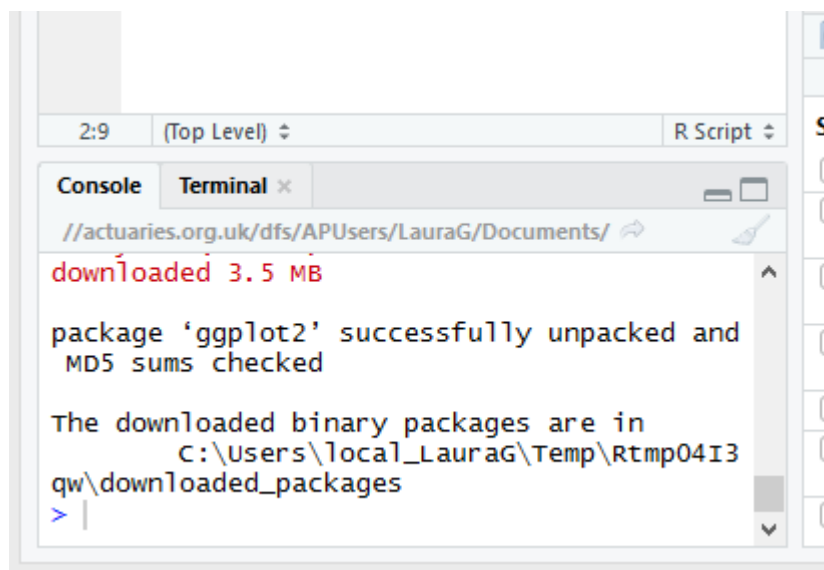Note that the name of the package must be in quotation marks, e.g
`"ggplot2"`.
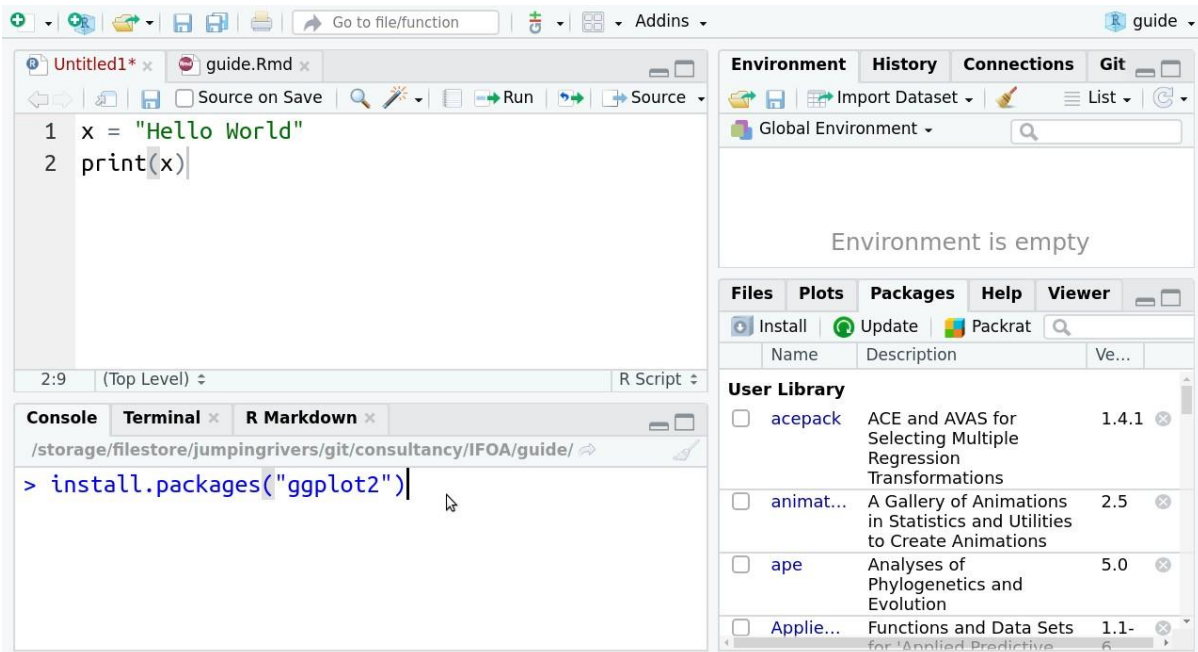


## Using a package

The installation process will add that package into your
computer, but these packages are not automatically loaded into
the session on start up because many people have hundreds of
packages, and it would be unmanageable and slow to load them
all at once.

To load a package for use in an R session we use:

`library(ggplot2)`

We can now use all the functions that are in the `ggplot2`
package.

Typically, we install once and go back to **'library'** whenever we
need that package.

# Data in R

Data in R are stored under different names. Some example data sets already exist in a default installation of R. To view a print out of the data we can just write the name. One such example data set is the *iris* data.

*iris*

The first few rows of the printed output are shown below.

```
##      Sepal.Length  Sepal.Width  Petal.Length  Petal.Width   Species
## 1        5.1           3.5          1.4           0.2        setosa
## 2        4.9           3.0          1.4           0.2        setosa
## 3        4.7           3.2          1.3           0.2        setosa
## 4        4.6           3.1          1.5           0.2        setosa
## 5        5.0           3.6          1.4           0.2        setosa
## 6        5.4           3.9          1.7           0.4        setosa
```

R will print the values into the console. If you prefer, a spreadsheet style view can be obtained using the `View()` function. Try:

`View(iris)`

This view is not editable, it serves as a preview of how the data will be presented.

## Getting data from files

There are numerous ways to load data from a file into RStudio. Here we will examine loading data into an RStudio session from three common file formats (**.csv, .xlsx** and **.Rdata**).

It is recommended to save all data files to a specific folder on your computer.

The simplest way to load data is to first set the working directory in RStudio, to do this click: "Session" > "Set Working Directory" > "Choose Directory…"
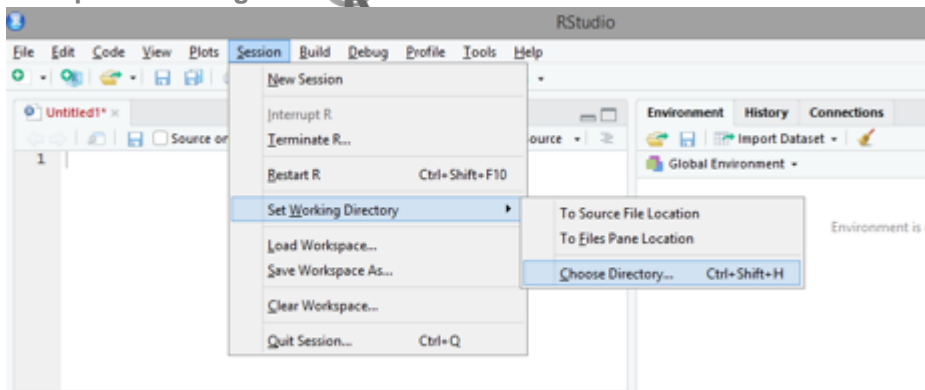
Navigate to the folder you have saved your data files and click "Open".

You will now see your data files in the "Files" tab as shown below:
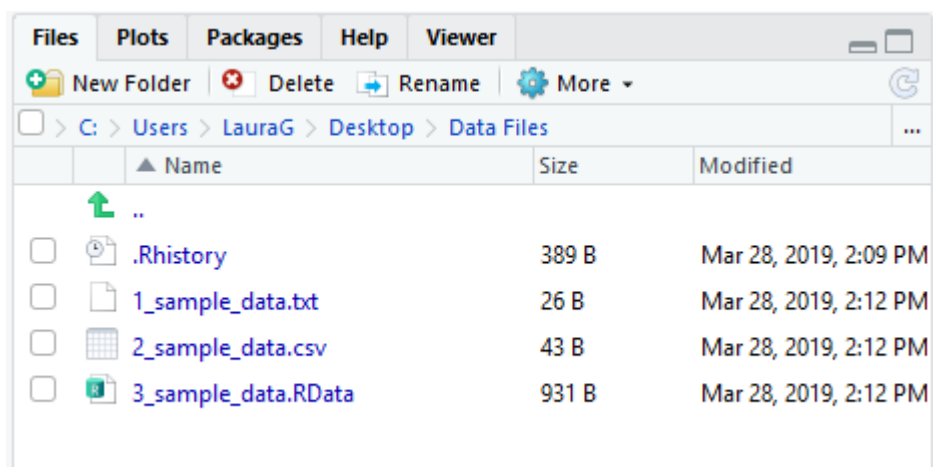


The alternative way to set the working directory is to input the following code:

```
setwd("input exact file location")
```

> Note: RStudio uses *forward slashes*.

> Correct file location: C:/Users/My Documents/data.csv

> Incorrect file location: C:\Users\My Documents\data.csv

To import a .txt or .csv file click on "Import Dataset…" within the "Enviroment" tab and select "From Text (base).

**Environment**  **History**  **Connections**

Import Dataset ▾

Global

From Text (base)...

From Text (readr)...

From Excel...                    ment is empty

From SPSS...

From SAS...

From Stata...
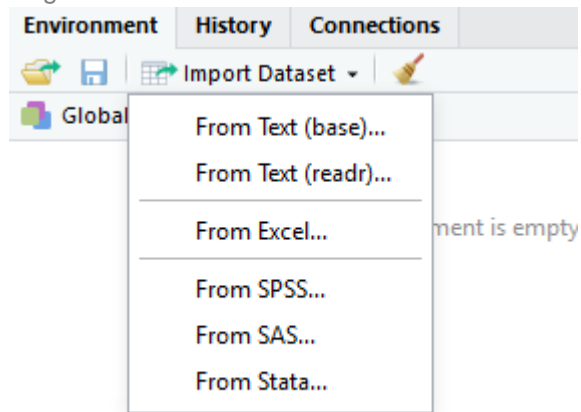
To import an Excel file select "From Excel..."

To import a .Rdata file click on the file and in the pop up window confirm the load by clicking "Yes".

**Confirm Load RData**   ?   ✕

Do you want to load the R data file "C:/Users/LauraG/Desktop/Data Files/
3_sample_data.RData" into the global environment?

Yes          No

☐  📄  1_sample_data.txt
☐  ▦  2_sample_data.csv
☑  Ⓡ  3_sample_data.RData

ıte it under certain conditions.
()' for distribution details
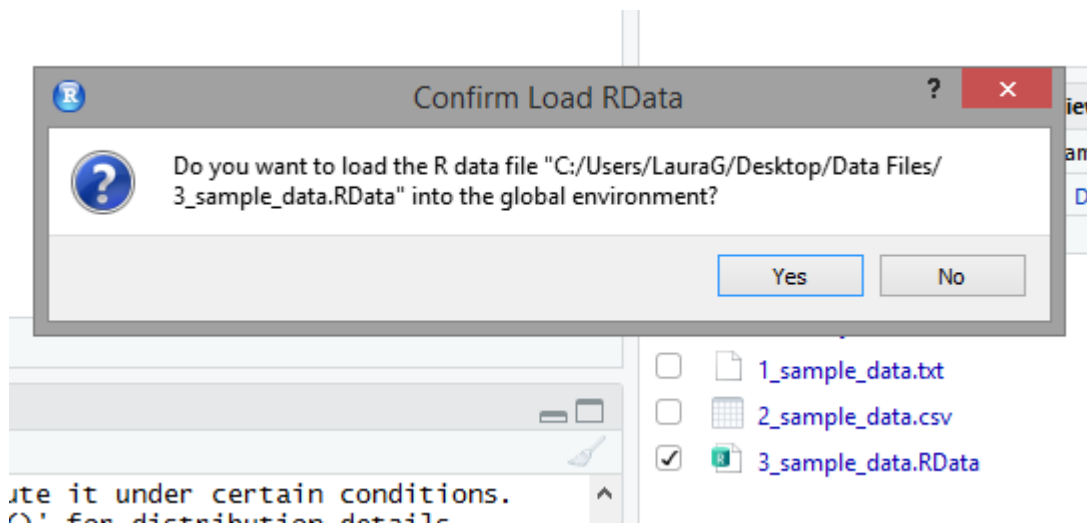
# Basic Use – R Console

Similar to RStudio there are a number of pre-installed packages available in R Console.  To select a pre-loaded package got to "Packages" > "Load Package" and select the package from the resulting pop up window.

To install additional packages, go to "Packages" > "Install Packages" and select your required package from the resulting pop up window.

## Getting data from files

Similar to RStudio you are advised to save data files to a specific folder on your PC.  You can set a working directory in R Console in two ways.
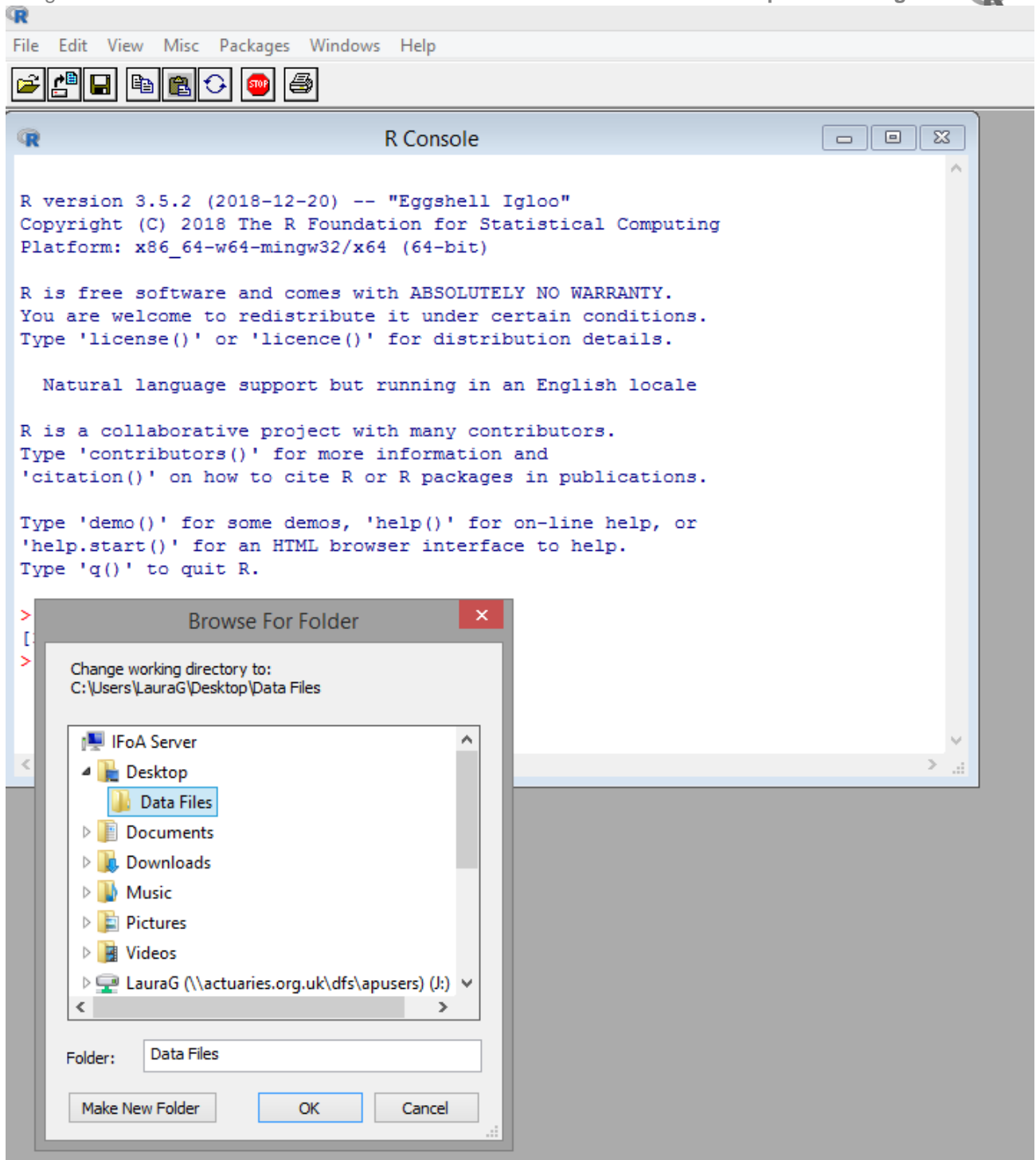
Input the code:
```
setwd("C:/Users/Desktop/Data Files")
```

Note that the code above is using *forward slashes*
To test this has worked by inputting the following:
```
getwd()
```

Alternatively you can navigate to "File" > "Change Dir" and navigate to your specific folder on your PC.

File   Edit   View   Misc   Packages   Windows   Help

R Console ▢ ▣ ✕

```
R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 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.

  Natural language support but running in an English locale

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.
```

Browse For Folder ✕

Change working directory to:
C:\Users\LauraG\Desktop\Data Files

🖥 IFoA Server
▲ 🖳 Desktop
    📁 Data Files
▷ 📄 Documents
▷ 📥 Downloads
▷ 🎵 Music
▷ 🖼 Pictures
▷ 🎞 Videos
▷ 🖥 LauraG (\\actuaries.org.uk\dfs\apusers) (J:)

Folder:   Data Files

Make New Folder        OK        Cancel

Once you have set the working directory you can run the following code in R to load the data:

To load a .txt file run the following code:
```
scan("datafilename.txt")
```

To load a .csv file run the following code:
```
read.csv("datafilename.csv",header=T)
```

To load a .Rdata file run the following code:

```
load("datafilename.Rdata")
```

# Help

R has documentation associated with each function stored in the packages themselves. Every single function that is directly accessible has a help page.

Subsequently, you can view the help documentation of any function in any package you have installed from within R itself. To see a help page you can use the `?` operator in front of a functions name. E.g.,

`?plot` You can also use the help() function e.g. help(plot) to see the help page.

The help pages typically have:

- ⓘ A description of what the function does.

- ⓘ Details on what the function arguments (inputs) represent.

- ⓘ Examples of using that function.

You can copy the example of code from the help page into your text editor and run it for yourself to see what the function does. Alternatively, `example()` function will extract the examples from the help page and run the code for you, stepping through each example. E.g.,

`example(plot)`

# Additional resources

R is a popular programming language and as such there is a wealth of information and code examples on the internet. Often a Google search for what you want to do will return a solution. In addition, there are a number of sites which may prove useful.

**https://rseek.org/** → A customised Google search engine that will only return R related results.

**https://stackoverflow.com/** → an online community for developers to learn and share their programming knowledge.

**https://www.rstudio.com/online-learning/** → Links and suggestions for starting to learn different aspects of R programming.

**https://community.rstudio.com/** → A discussion board for asking questions about R.

**https://www.rstudio.com/resources/cheatsheets/** → pdf cheat-sheets for using RStudio and popular R packages.

**https://www.r-bloggers.com/** → A blog aggregator from contributors with posts and tutorials on interesting things happening in R. As well as some good introductory level tutorials, this is a great place for discovering new packages and learning new things.