Institute
and Faculty
of Actuaries

Get up and running with R

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 select 'download R'

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 - 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.

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/

(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).

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.

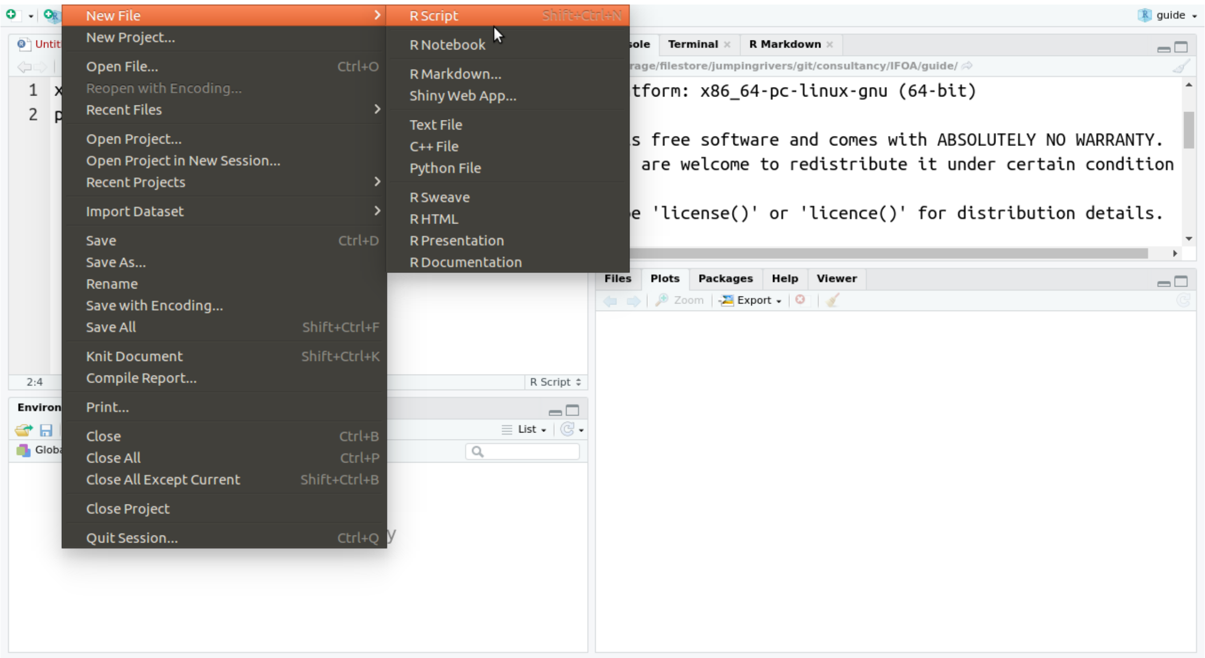
Basic Use

RStudio 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.



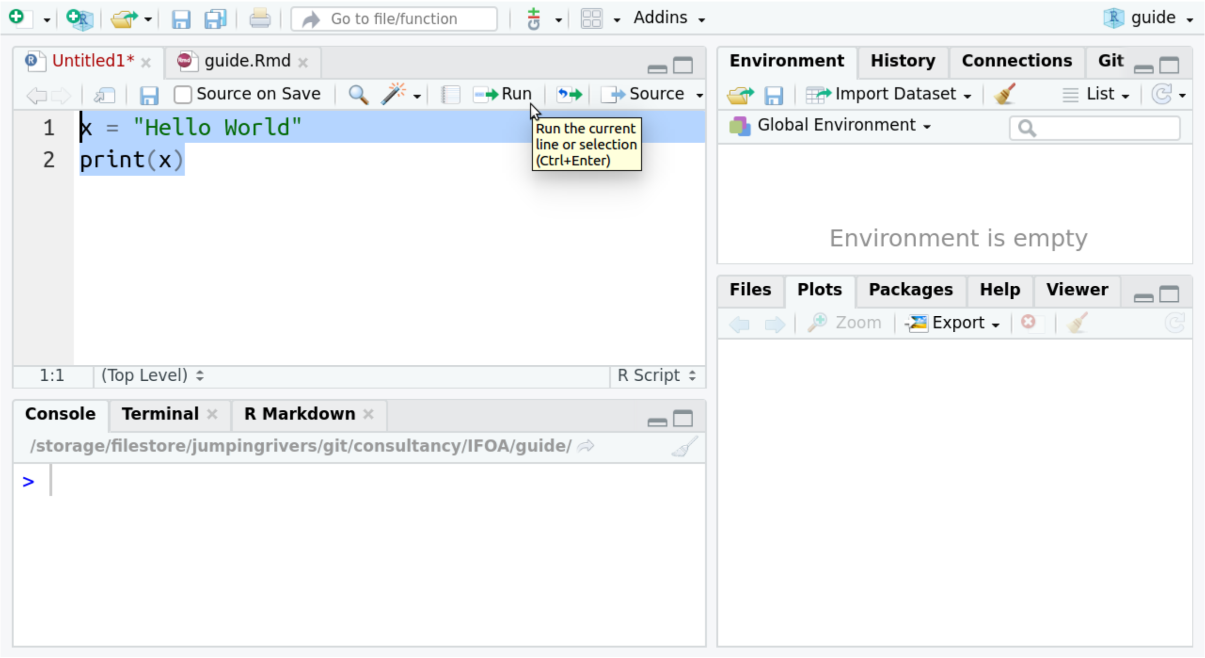
```
tform: x86_64-pc-linux-gnu (64-bit)

s free software and comes with ABSOLUTELY NO WARRANTY.
 are welcome to redistribute it under certain condition

e 'license()' or 'licence()' for distribution details.
```

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.



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

Run the current line or selection (Ctrl+Enter)

/storage/filestore/jumpingrivers/git/consultancy/IFOA/guide/

Environment is empty

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.

Example installation

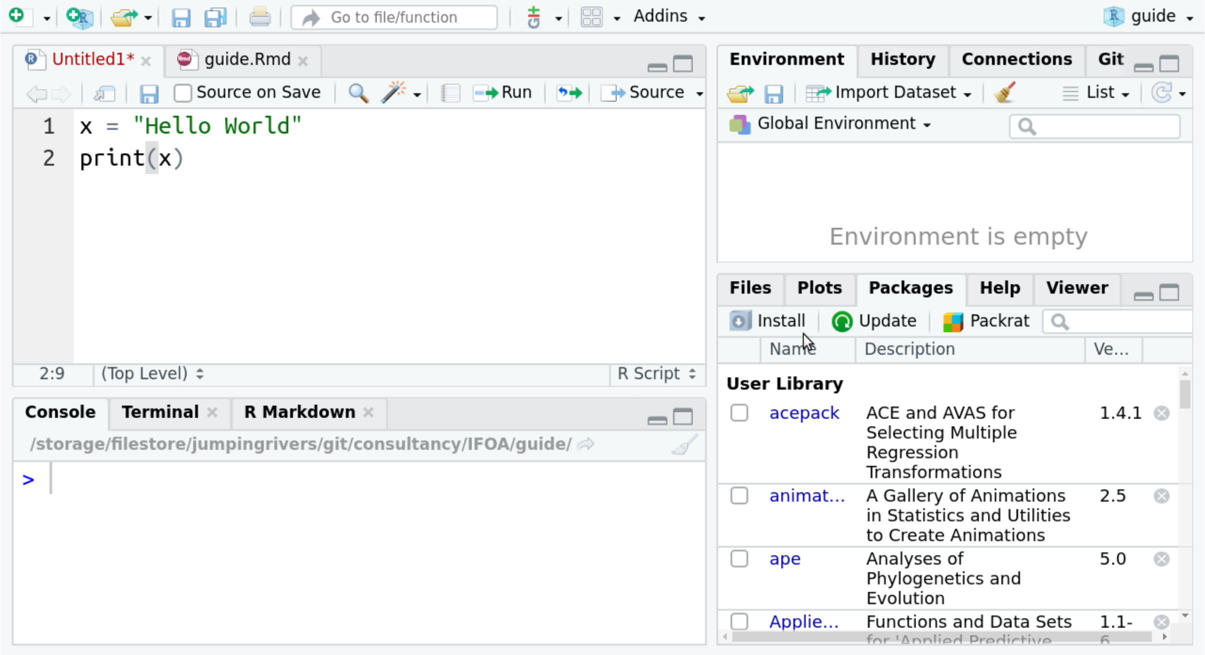
As an example, we will run through the installation of a popular R package for graphics, ‘**ggplot2**’.

There are two methods for installing R packages in RStudio.

GUI method

Step 1 Click on the ‘Packages’ tab.

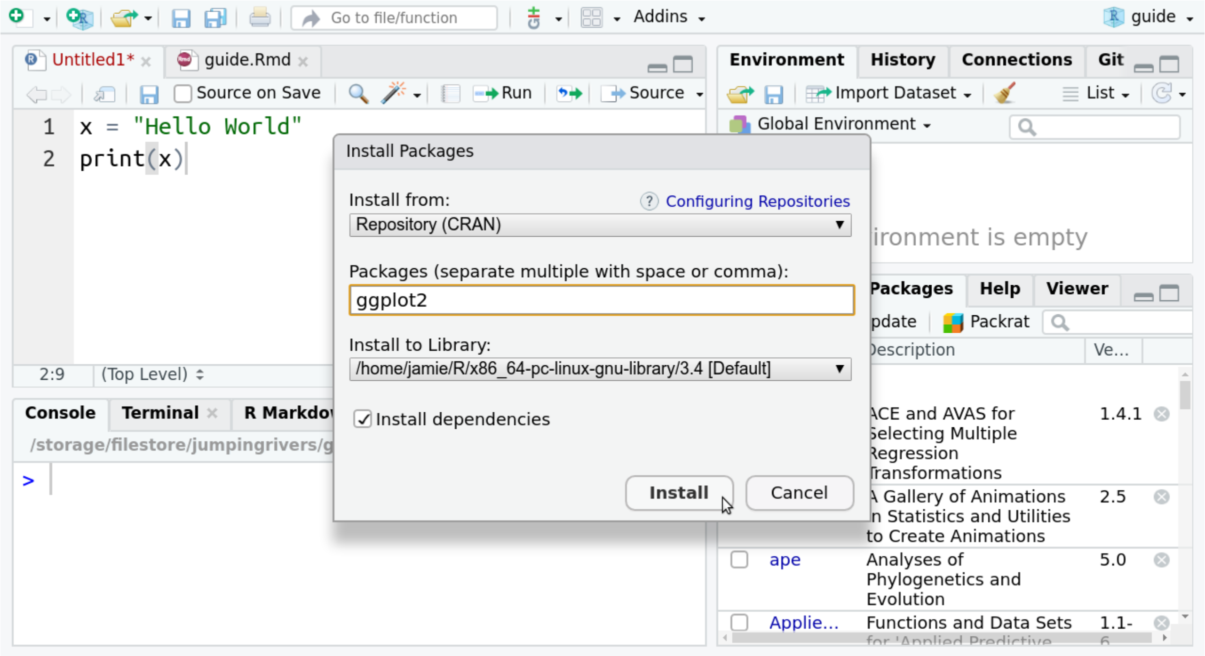
Step 2 Click ‘Install’.



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

	Name	Description	Ve...
	User Library		
□	acepack	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
□	animat...	A Gallery of Animations in Statistics and Utilities to Create Animations	2.5
□	ape	Analyses of Phylogenetics and Evolution	5.0
□	Applie...	Functions and Data Sets for 'Applied Predictive	1.1-6

Step 3 In the resulting popup window, type the name of the package `ggplot2` under ‘Packages’.



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

Install Packages

Install from: Repository (CRAN)

Configuring Repositories

Packages (separate multiple with space or comma): ggplot2

Install to Library: /home/jamie/R/x86_64-pc-linux-gnu-library/3.4 [Default]

☑ Install dependencies

		Description	Ve...
		ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
		A Gallery of Animations in Statistics and Utilities to Create Animations	2.5
□	ape	Analyses of Phylogenetics and Evolution	5.0
□	Applie...	Functions and Data Sets for 'Applied Predictive	1.1-6

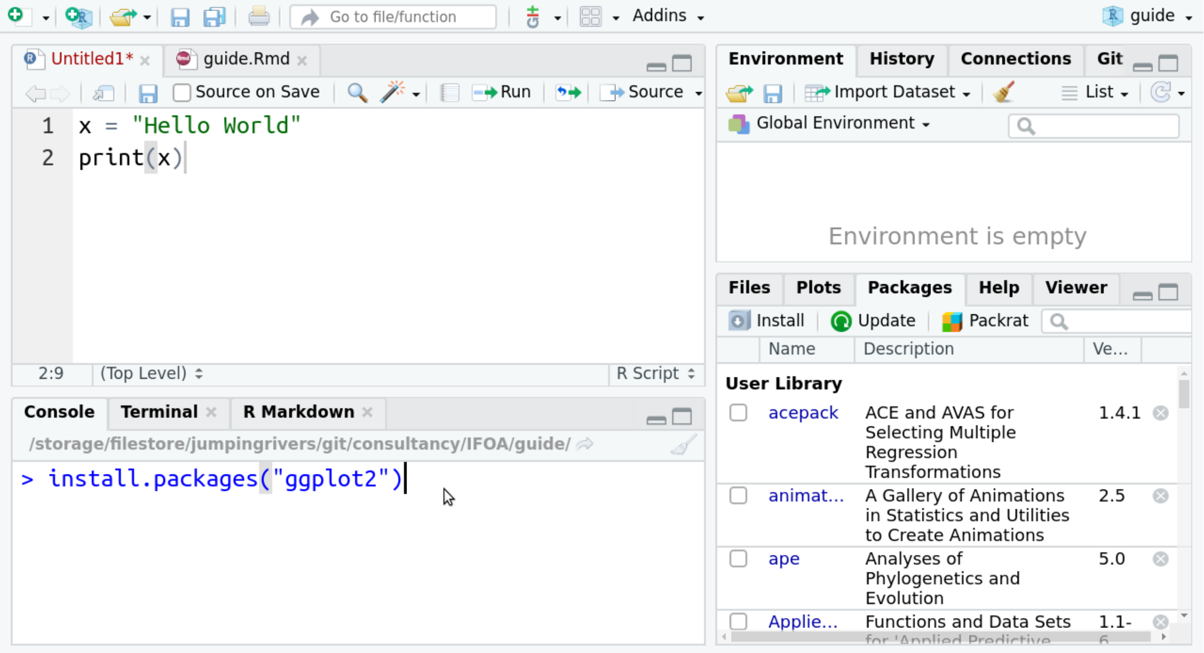
Using code to install packages

It is also possible to install R packages in the console.

Go to the console window 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"`.



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

/storage/filestore/jumpingrivers/git/consultancy/IFOA/guide/

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

Environment is empty

	Name	Description	Ve...
User Library			
□	acepack	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
□	animat...	A Gallery of Animations in Statistics and Utilities to Create Animations	2.5
□	ape	Analyses of Phylogenetics and Evolution	5.0
□	Applie...	Functions and Data Sets for 'Applied Predictive	1.1-6

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

R is a powerful language for manipulating, processing and analysing data and, at first, it may feel intimidating to work with it if you are used to a program like Excel. There is no user-friendly ‘point-and-click’ interface; R operates entirely by you typing up and running ‘scripts’ - based on your knowledge of how the R programming language works. For many people, this is a new kind of way of thinking and interacting with a computer - plus it requires quite a bit of experience coding in R before you can do much with it. As a result, there is often a steep learning curve for people learning R for the first time.

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 R. Here we will examine loading data into an R session from two common file formats (**.csv** and **.xlsx** Excel files).

CSV files

Csv files are a popular storage format for tabular data as they are easy to read and can be understood by a number of pieces of software on all major operating systems. They are essentially a plain text format of a table of data with commas separating the individual values and new lines separating the rows.

Given a full file location (e.g 'C:/Users/My Documents/data.csv') we can load that data to a variable for use in R with:

```
my_data = read.csv("C:/Users/My Documents/data.csv")
```

Whenever you want to view or use the data from that file you use the variable:

```
my_data
```

Excel files

Reading directly from Excel files requires installation of a package called `readxl`.

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

To make use of this package in our R session:

```
library(readxl)
```

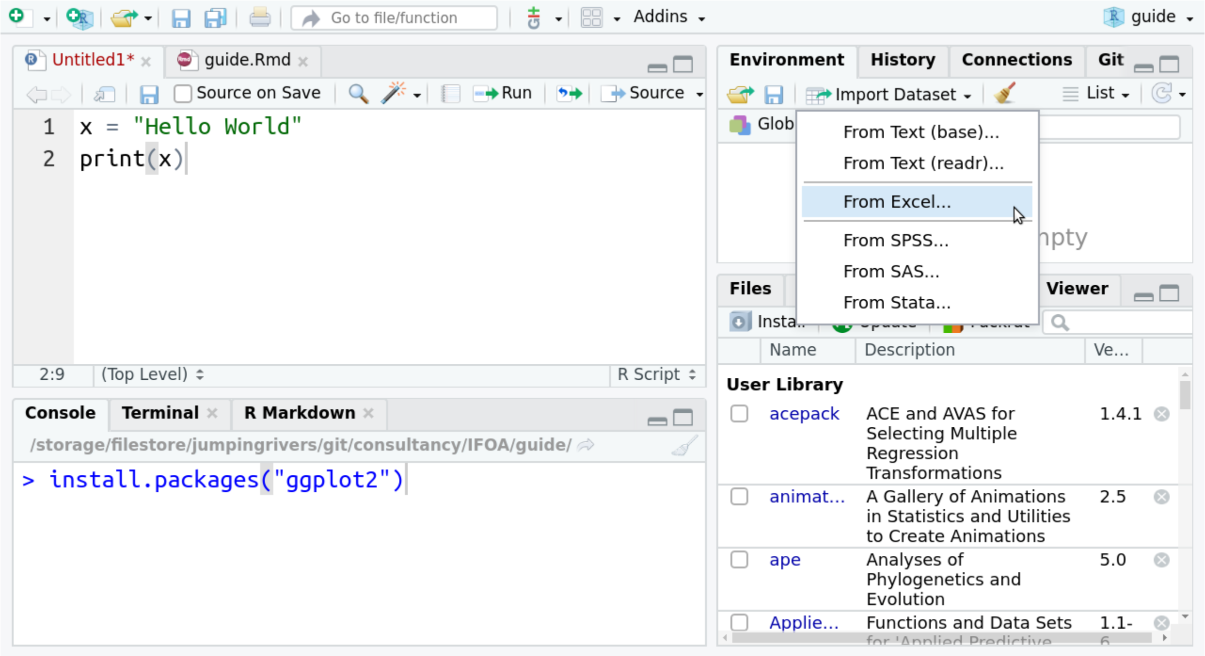
Given a full file location (e.g. "C:/Users/My Documents/data.csv") we can load that data to a variable for use in R with:

```
my_data = read_excel("C:/Users/My Documents/data.csv")
```

There are lots of additional arguments you can change in read_excel if you want to only load specific cells or sheets.

GUI approach

You can also import data in RStudio using the GUI. In the Environment tab click 'Import Dataset'. Choose, for example, 'from Excel' if you want to load data from an Excel file.

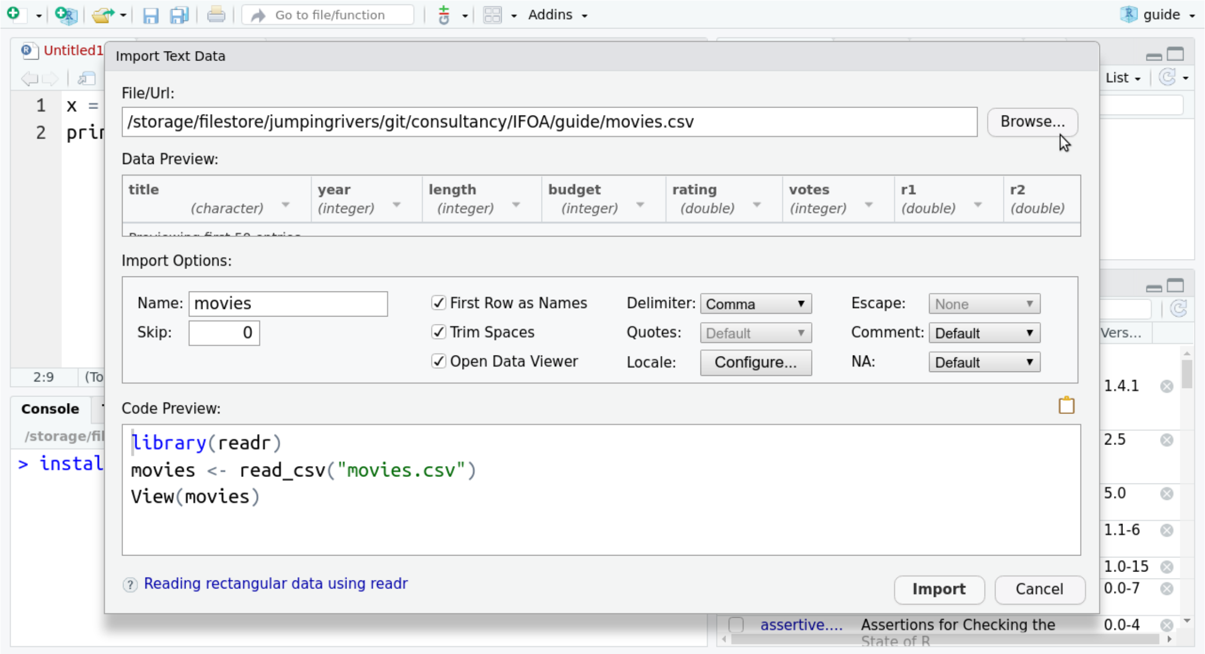


```
x = "Hello World"
print(x)
```

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

	Name	Description	Ve...
	User Library		
□	acepack	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
□	animat...	A Gallery of Animations in Statistics and Utilities to Create Animations	2.5
□	ape	Analyses of Phylogenetics and Evolution	5.0
□	Applie...	Functions and Data Sets for 'Applied Predictive	1.1-6

You will then be presented with a popup that gives a more standard looking file browser to select the data file. On doing so, you will be presented with a preview of the code that will run to load this data and a view of the first few rows of the data to be loaded.



File/Url: /storage/filestore/jumpingrivers/git/consultancy/IFOA/guide/movies.csv

Data Preview:

title	year	length	budget	rating	votes	r1	r2
(character)	(integer)	(integer)	(integer)	(double)	(integer)	(double)	(double)

Import Options:

Name: movies
Skip: 0
☑ First Row as Names
☑ Trim Spaces
☑ Open Data Viewer
Delimiter: Comma
Quotes: Default
Locale: Configure...
Escape: None
Comment: Default
NA: Default

Code Preview:

```
library(readr)
movies <- read_csv("movies.csv")
View(movies)
```

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
```

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://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.

Credit: The R logo is © 2016 The R Foundation, licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.