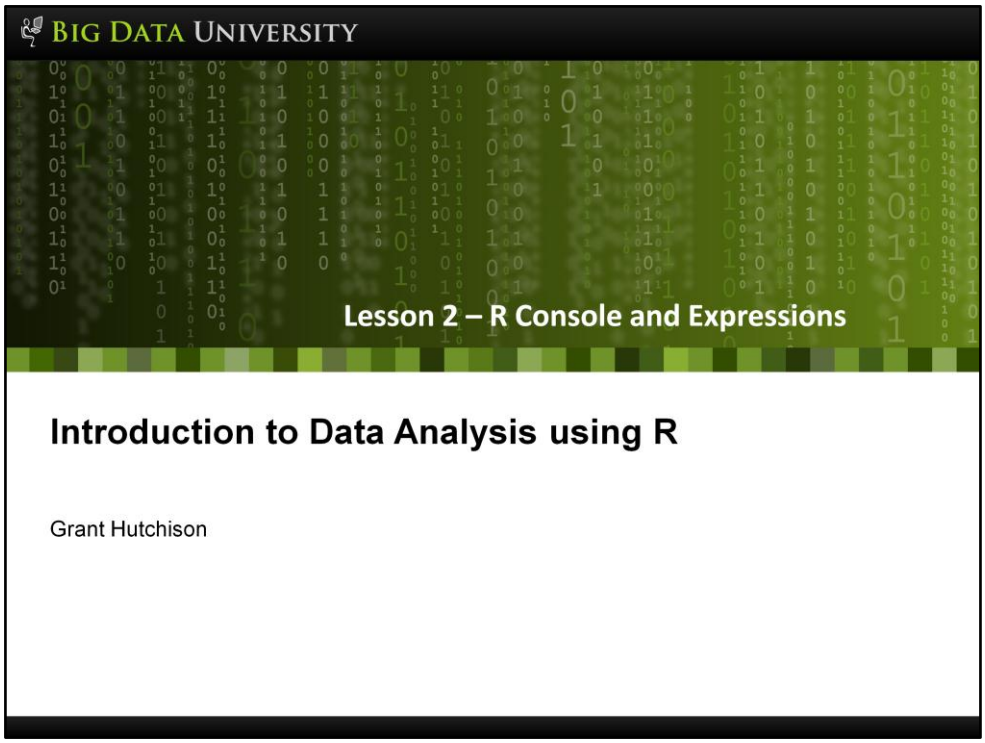
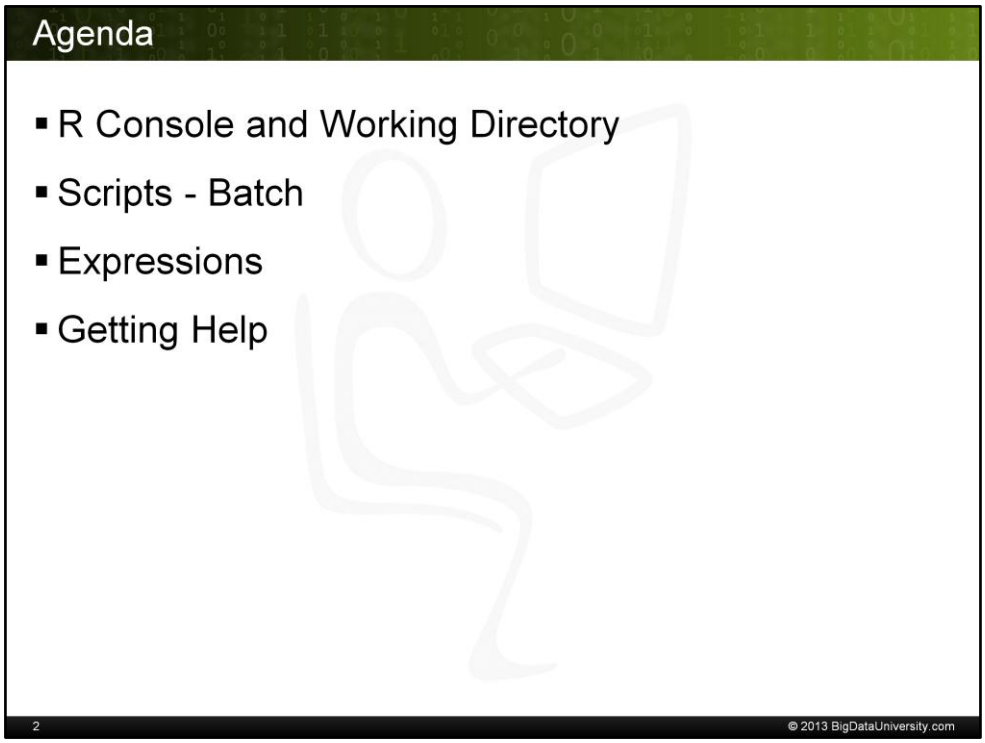


Welcome to the this lesson on how to use the R Console and R Expressions.

Hopefully you have R installed at this point and you can practice using R.

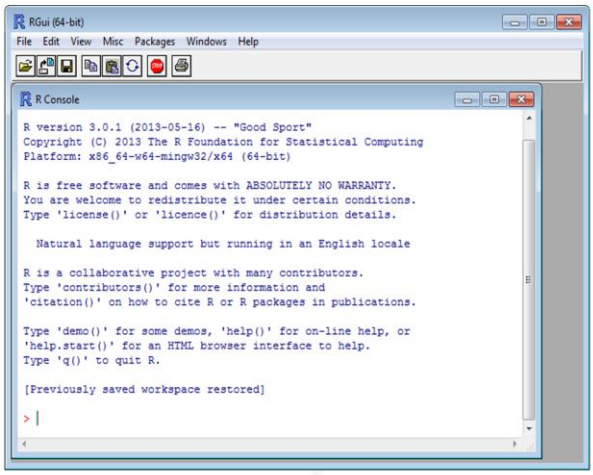
Let's get started.



R scripts can be executed within the R Console or outside of the R Console as batch operations.

We will examine some simple R Expressions and learn how to obtain help along the way.

R - Console



R Console prompt is the '>' symbol

- If the expression is complete the it is executed
- Otherwise, the prompt changes to the '+" symbol

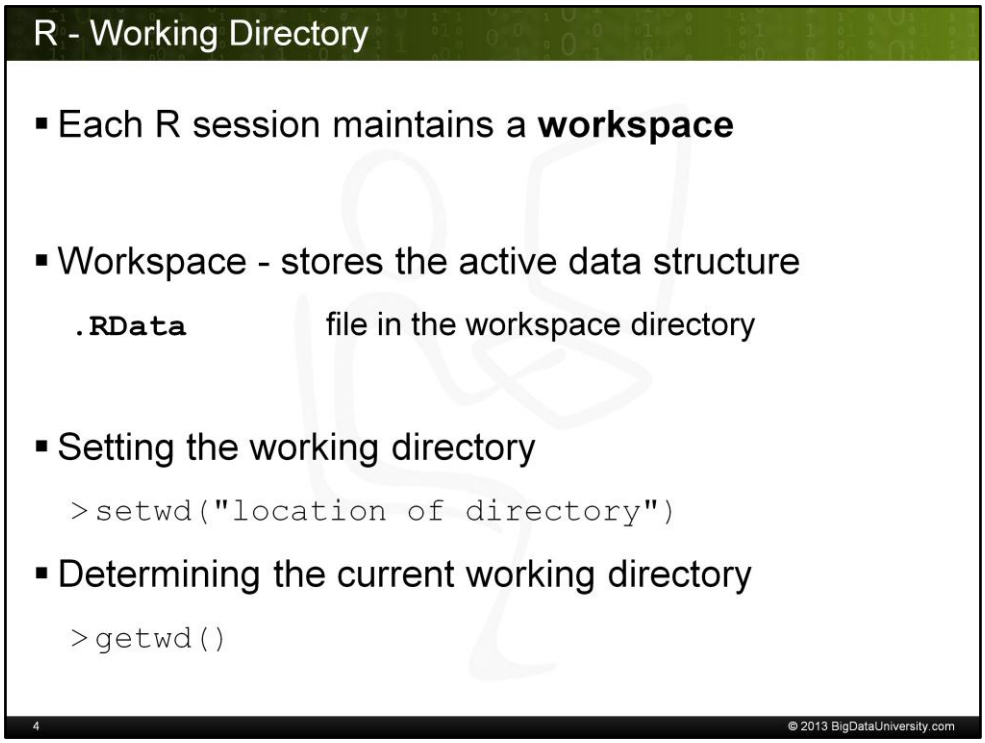
The starting point for every new R user is the R Console.

The R Console can be used to issue interactive R functions or expressions.

The greater than (>) symbol is the default prompt for R.

When an R expression is complete, within the R Console, it can be executed.

If the expression is not complete, the R Console prompt will change to the plus (+) symbol.



Each R session has a working directory.

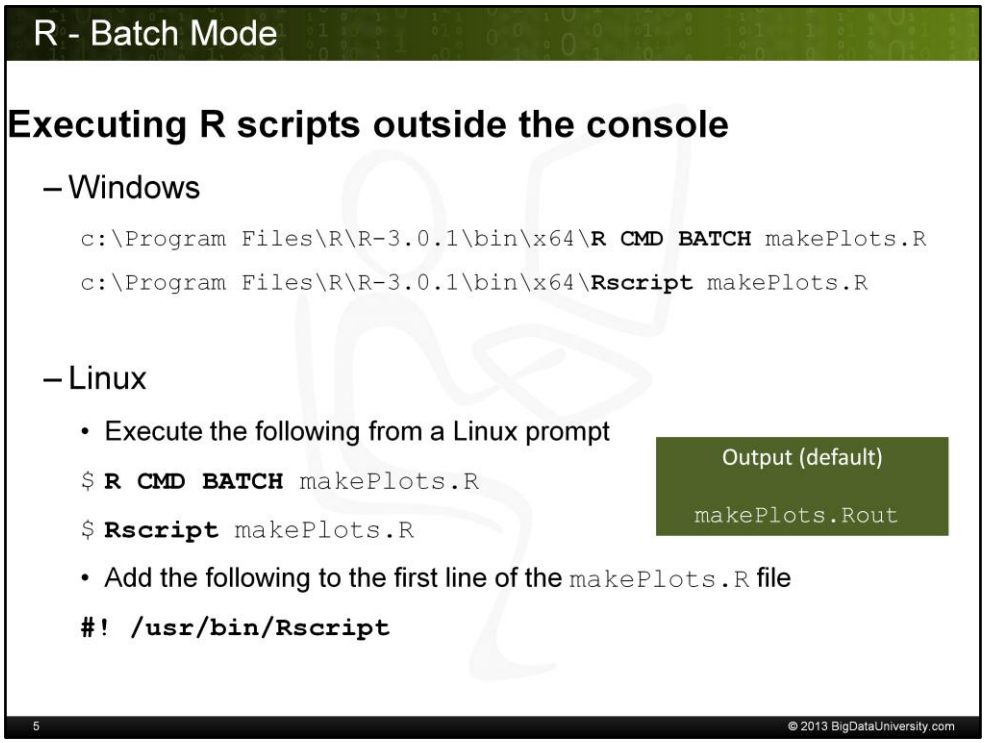
A working directory contains a workspace file, named **.Rdata**, which contains a list of R objects. R will also keep file with a history of previously issues R commands.

The R working directory can be changed using the **setwd()** function.

To determine the current working directory simply issue the **getwd()** function.

When you complete a session of using the R Console you will be prompted to consider saving your workspace.

You may want to save your workspace so you can return to your analysis at a later time, but it is always a good practice to create R scripts from your R Console sessions so you do not accidently overwrite your workspace file.



Executing R scripts outside of the R Console can help automate your data analysis activity.

Let's consider a scenario where you have created an R script that performs data analysis and generates a set of data plots as PDF files.

It would be desirable to automate this task and avoid using the R Console altogether.

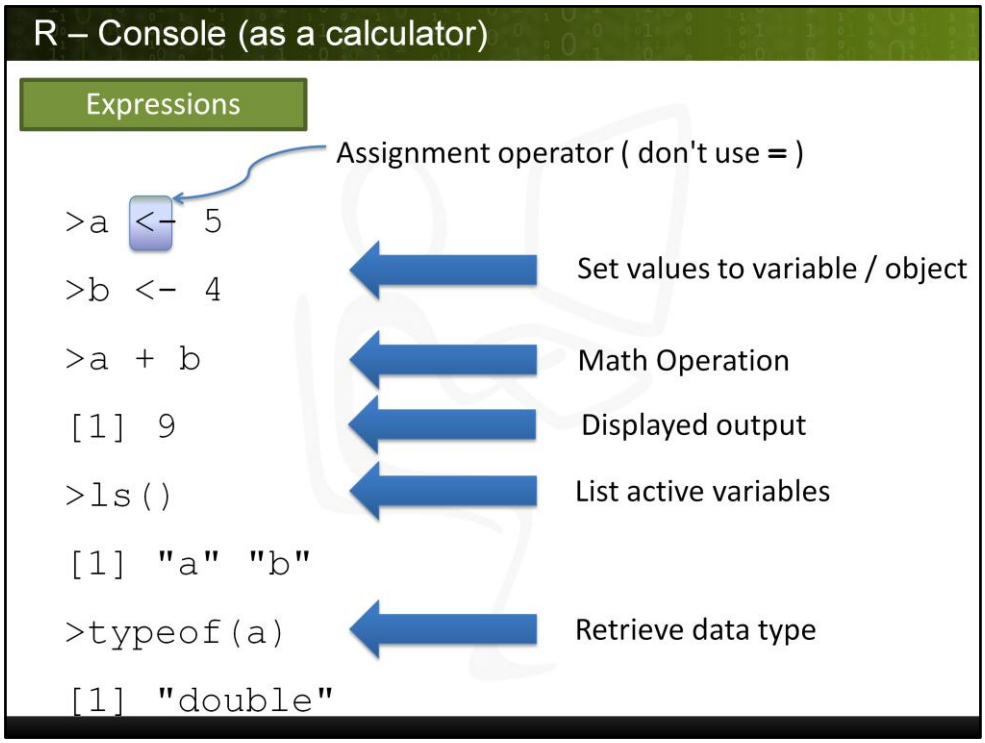
There are a few different methods of executing R scripts in batch mode.

The command R CMD BATCH can be used across all platforms.

There is also an tool called Rscript that can be used across all platforms to run scripts outside of the Console. This tool gives you the ability to obtain input from the user as the script executes, if required.

On Linux, R script files can be changed into an executable script by simply adding a reference to the Rscript executable in the first line of the script file. Be sure to change the file permissions of your script to be executable if you decide to use this approach.

By default, there is an output file created for any information sent to standard output. The name of the output file is the same name as the script with the extension .Rout.



Let's consider using R as a simple calculator.

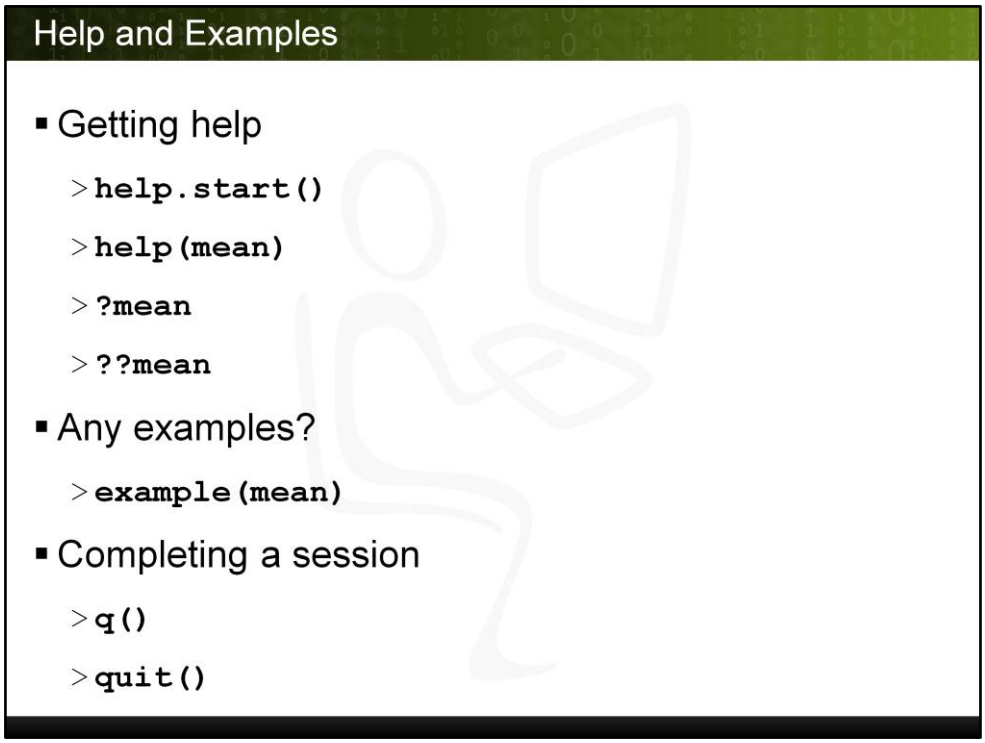
R code consists of a series of expressions. These expressions can be generally classified as: data, computation, or control flow operations. In this first example we are storing the value 5 into an object or variable called 'a'. The data type of object is automatically determined by R.

It is possible to explicitly state the data type of a value, but for now we will let R decide. The recommended assignment operator in R is the Less Than sign (<-) followed by a dash. A single equals sign (=) can be used, but it not recommended. The addition operator is a built-in function in R and here we see the result of adding the values stored in objects 'a' and 'b'. The output is displayed in the R Console, but it has not been assigned to a object and therefore it is no longer accessible. The ls(), or List objects, function is useful while using the R Console as it provides a displays a list of all of the currently active objects.

You may have noticed that the output displayed contained the value 1 inside of square brackets [1]. This is how R indicates that the data is the first of possibly many items. There are various other R functions that can be used to describe an R object.

For example, here we are interested in the data type of the object 'a' and we discover that R decided that the datatype for 'a' is a double precision floating point number. As

we will learn later in the course, it is possible to ensure that the values stored in the object 'a' are limited to integers only.



As we become more familiar with R, it is important to know where to go for help.

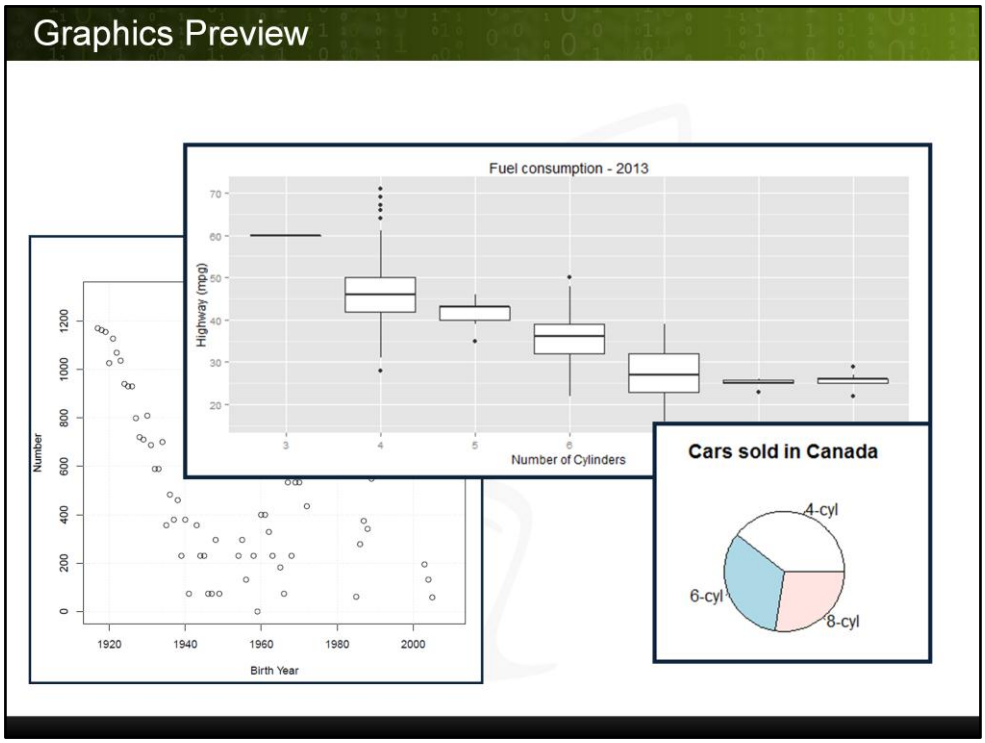
The **help.start()** function will launch a local web server with links to various R reference materials.

If you know the name of an R function, but you can't remember its input arguments or its output, use the **help()** function or use a single question mark to obtain more information.

A set of two question marks (??) followed by a search term can used to perform a search across all of the installed R packages on your system.

Many functions have built-in examples available to help you understand how to use them. The **example()** function can be invoked to access these built-in samples.

When you are ready to terminate a session within the R Console, use the **q()** or **quit()** function.



R has excellent support for creating high quality graphics with minimal effort.

Here are a few simple examples.

First, we have a **scatterplot** showing the trends in naming females born in Ontario, Canada from 1917 through 2010. The name 'Emma' was selected for this plot.

The next example is a **box plot** showing the fuel consumption of cars available in Canada in 2013. The data has been categorized by the number of cylinders the vehicle has.

The final plot is of a simple **pie-chart** showing the distribution of 4, 6, and 8 cylinder cars available in Canada in 2013.

We will learn how to create these and other plots in future lessons.

Thank you for completing this Lesson.

Spend some time getting familiar with the R console, expressions, and R scripts before moving on.