R Tutorial

Obtaining R

R is available for Linux, MacOS, and Windows. Software can be downloaded from The Comprehensive R Archive Network (CRAN).

Startup

After R is downloaded and installed, simply find and launch R from your Applications folder.

Entering Commands

R is a command line driven program. The user enters commands at the prompt (> by default) and each command is executed one at a time.

The Workspace

The workspace is your current R working environment and includes any user-defined objects (vectors, matrices, data frames, lists, functions). At the end of an R session, the user can save an image of the current workspace that is automatically reloaded the next time R is started.

Graphic User Interfaces

Aside from the built in R console, RStudio is the most popular R code editor, and it interfaces with R for Windows, MacOS, and Linux platforms.

Operators in R

R's binary and logical operators will look very familiar to programmers. Note that binary operators work on vectors and matrices as well as scalars.

Arithmetic Operators include:

Operator	Description
+	addition
-	subtraction
*	multiplication
/	division
^ or **	exponentiation

Logical Operators include:

Operator	Description
>	greater than
>=	greater than or equal to
==	exactly equal to
!=	not equal to

Data Types

R has a wide variety of data types including scalars, vectors (numerical, character, logical), matrices, data frames, and lists.

Creating New Variables

Use the assignment operator **<-** to create new variables.

```
# An example of computing the mean with variables

mydata$sum <- mydata$x1 + mydata$x2
mydata$mean <- (mydata$x1 + mydata$x2)/2
```

Functions

Almost everything in R is done through functions. A function is a piece of code written to carry out a specified task; it may accept arguments or parameters (or not) and it may return one or more values (or not!). In R, a function is defined with the construct:

```
function ( arglist ) {body}
```

The code in between the curly braces is the body of the function. Note that by using built-in functions, the only thing you need to worry about is how to effectively communicate the correct input arguments (arglist) and manage the return value/s (if any).

Importing Data

Importing data into R is fairly simple. R offers options to import many file types, from CSVs to databases.

For example, this is how to import a CSV into R.

```
# first row contains variable names, comma is separator
# assign the variable id to row names
# note the / instead of \ on mswindows systems

mydata <- read.table("c:/mydata.csv", header=TRUE,
  sep=",", row.names="id")
```

Descriptive Statistics

R provides a wide range of functions for obtaining summary statistics. One way to get descriptive statistics is to use the **sapply( )** function with a specified summary statistic.

Below is how to get the mean with the **sapply( )** function:

```
# get means for variables in data frame mydata
# excluding missing values
sapply(mydata, mean, na.rm=TRUE)
```

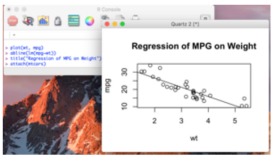
Possible functions used in **sapply** include **mean, sd, var, min, max, median, range** , and **quantile**.

Plotting in R

In R, graphs are typically created interactively. Here is an example:

```
# Creating a Graph
attach(mtcars)
plot(wt, mpg)
abline(lm(mpg~wt))
title("Regression of MPG on Weight")
```

The **plot( )** function opens a graph window and plots weight vs. miles per gallon. The next line of code adds a regression line to this graph. The final line adds a title.



Packages

Packages are collections of R functions, data, and compiled code in a well-defined format. The directory where packages are stored is called the library. R comes with a standard set of packages. Others are available for download and installation. Once installed, they have to be loaded into the session to be used.

```
.libPaths() # get library location
library()   # see all packages installed
search()    # see packages currently loaded
```

Getting Help

Once R is installed, there is a comprehensive built-in help system. At the program's command prompt you can use any of the following:

```
help.start()   # general help
help(foo)      # help about function foo
?foo           # same thing
apropos("foo") # list all functions containing string foo
example(foo)   # show an example of function foo
```

Going Further

If you prefer an online interactive environment to learn R, this free R tutorial by DataCamp is a great way to get started.