GH 811 R Learning Module 1

Andrew Stokes

September 6, 2018

Welcome

Welcome to GH 811 and to your first R training module! The philosophy of GH 811 is that when it comes to programming, we learn best by doing. Just like learning a new language, the lecture format simply does not cut it. Ultimately, what really makes a difference is immmersion in the language and culture, where you have no choice but to try to communicate from day one.

Consistent with this philosophy, the R training modules in this course are intended as active learning exercises for you to complete prior to each week's class session. Since you will be going through these modules before class, we can use valuable class time for interactive learning which challenges you to push yourself beyond what you imagined possible. These interactive learning sesesions will typically represent a third of the overall class session, running from about 5-6 PM and the key to successfully completing them will be putting in the time before class to learn the competencies discussed in each learning module. So without further delay let's get started on the first learning module.

Learning Module 1

The objectives of learning module 1 are as follows:

- Why R?
- Downloading R/R Studio
- R Studio Interface
- Keyboard shortcuts
- Introductory commands
- Logical operators
- The working directory
- Reading in data
- Data structures in R

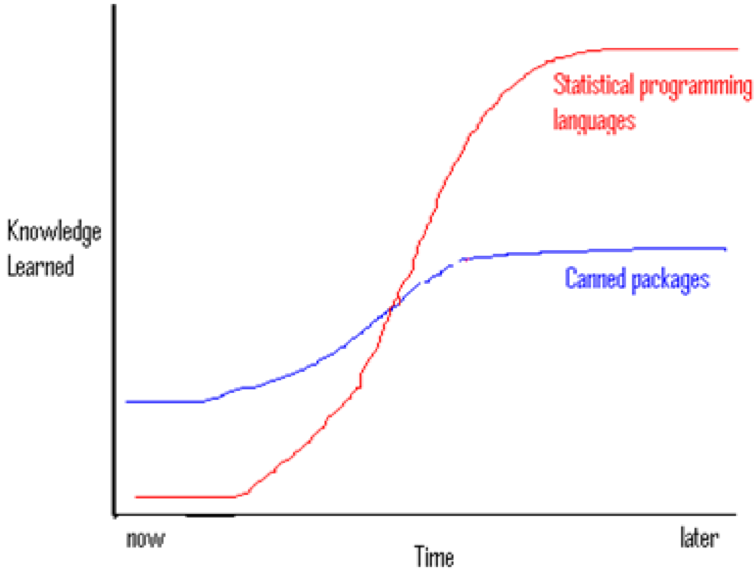
Introduction to R

- R is a programming language for statistical data manipulation and analysis
- R is an alternative to programs like SAS and Stata

Why R is growing in popularity

- Open-source, which means the source code is available for free on the internet
- Free (vs. SAS and Stata, which have expensive licensing fees)
- Large and thriving user community
- Many user-contributed packages that extend functionality and increase relevance to latest methods
- Flexible, useful and compelling graphics

Statistical programming languages vs. canned packages[1]



Downloading R and RStudio

- Base R is the default platform for programming in R and can be downloaded from the Comprehensive R Archive Network (CRAN) at https://cran.r-project.org/.
- R Studio (https://www.rstudio.com/) is a platform for programming in R. It manages version control, includes autocomplete features, saves plot history, manipulates graphs interactively and saves search command history.
- To run R Studio, you have to install CRAN R first.

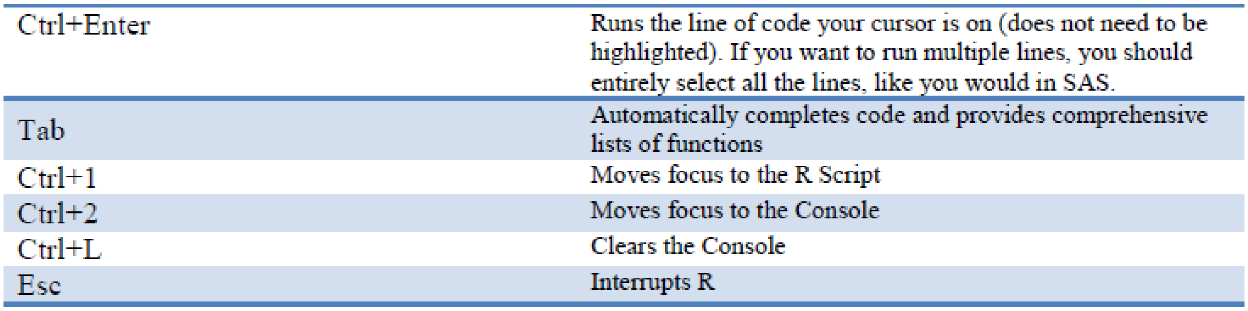
R Studio Interface

The R Studio Interface has four parts:

- Top left: R script window, where you will write your R program.
- Bottom left: this is the console and is where code is implemented and the output appears. Note that you can write code directly in the console, but this should be done sparingly
- Top right: this is the Workspace. The Workspace includes every object used in your current R Studio session. Objects include data sets, vectors, regression results, etc.
- Bottom right: This quadrant includes several tabs, which include any plots you generate in R, access to help files and packages you have installed and activated.

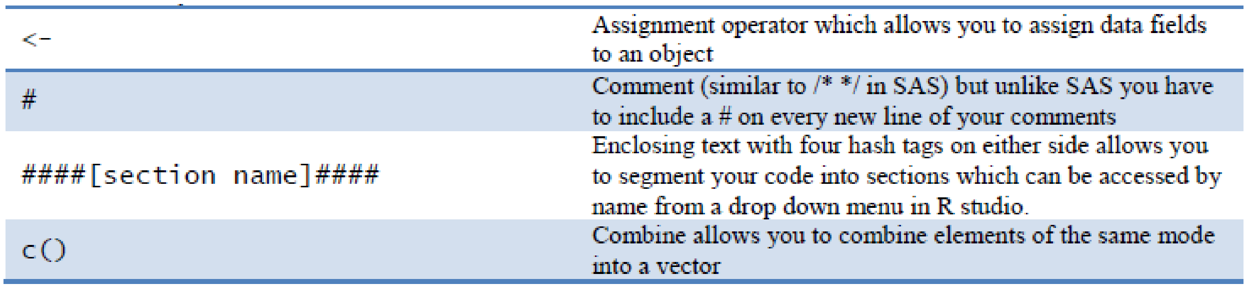
[1] Source: Gary King

Helpful Keyboard Shortcuts in RStudio



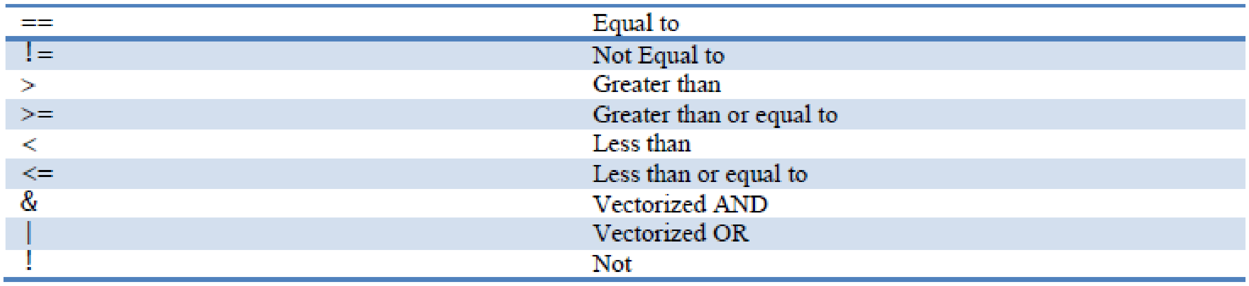
Ctrl+Enter	Runs the line of code your cursor is on (does not need to be highlighted). If you want to run multiple lines, you should entirely select all the lines, like you would in SAS.
Tab	Automatically completes code and provides comprehensive lists of functions
Ctrl+1	Moves focus to the R Script
Ctrl+2	Moves focus to the Console
Ctrl+L	Clears the Console
Esc	Interrupts R

Introductory Commands in R



<-	Assignment operator which allows you to assign data fields to an object
#	Comment (similar to /* */ in SAS) but unlike SAS you have to include a # on every new line of your comments
####[section name]####	Enclosing text with four hash tags on either side allows you to segment your code into sections which can be accessed by name from a drop down menu in R studio.
c()	Combine allows you to combine elements of the same mode into a vector

Logical Operators in R



==	Equal to
!=	Not Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
&	Vectorized AND
|	Vectorized OR
!	Not

The Working Directory

The first thing you should do when you begin an R script is to set the working directory. This is the default location on your hard drive that R will look to in order to read and write files.

```r
input <- "C:/Users/acstokes/Desktop/IH 811"
setwd(input)
data<-read.csv("name_of_data_set", header=TRUE)
```

Data Structures in R

R has an object-oriented data type system, with each type of object having its own properties. The major types are:

- scalars
- vectors
- matrices
- lists
- data frames
- factors

Scalars

Scalars are simply an individual number, such as 3.

```
x<-1
x
```

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

```
pi
```

```
## [1] 3.141593
```

Vectors

- A vector is an array of elements with the same mode (e.g.numeric, character, or boolean).
- Scalars are stored as a vector of length 1 in R.
- Columns in a matrix or variables in a data frame are also stored as vectors.

```
pi
```

```
## [1] 3.141593
```

```
is.vector(pi)
```

```
## [1] TRUE
```

```
x<-c(1,2,3,4)
x
```

```
## [1] 1 2 3 4
```

```
ghfac <- c("Beard", "Feeley", "Gill", "Halim", "Onyango")
ghfac
```

```
## [1] "Beard"   "Feeley"  "Gill"    "Halim"   "Onyango"
```

Matrix

- A matrix is a rectangular array of same-length vectors, all of which have the same mode.
- R functions that perform statistical models often require the input to be a matrix.

```
#creation of a two-by-two matrix from a four-element vector
X <- matrix(c(1,2,3,4), nrow=2,ncol=2)
X
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

```r
is.matrix(X)
```

```
## [1] TRUE
```

List

- A list is a collection of other objects, none of which has to be of the same mode or structure (e.g. a list could contain both a character vector and a numeric matrix as separate elements).
- Many R functions that perform statistical models return the output as a list containing various components

Data frames

- In a data frame, each column is a vector that can be of different modes
- Data frames are similar to datasets in SAS or Stata

```r
x1 <- c(1,2,3)
x2 <- c("AB","BC","CD")
D <- data.frame(x1,x2)
D
```

```
##   x1 x2
## 1  1 AB
## 2  2 BC
## 3  3 CD
```

Factors

- Factors are similar to vectors with one difference: factors include a list of its unique values, called levels, sorted alphabetically for character values and in ascending order for numeric values
- This additional information allows R to perform categorical analyses

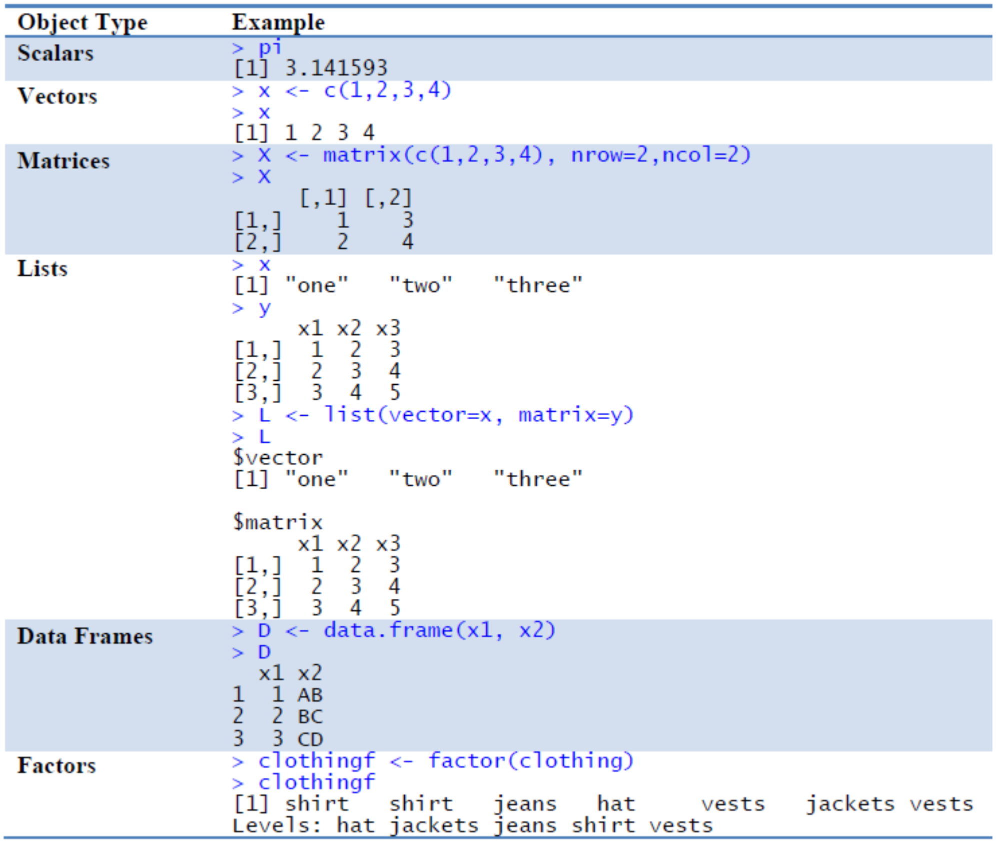
```r
#Creating a factor from a numeric vector
x <- c(1,2,3,3,4,5,5,6)
xf <- factor(x)
x
```

```
## [1] 1 2 3 3 4 5 5 6
```

```r
xf
```

```
## [1] 1 2 3 3 4 5 5 6
## Levels: 1 2 3 4 5 6
```

Examples of R Objects



Object Type	Example

Scalars	`> pi`
	`[1] 3.141593`

Vectors	`> x <- c(1,2,3,4)`
	`> x`
	`[1] 1 2 3 4`

Matrices	`> X <- matrix(c(1,2,3,4), nrow=2,ncol=2)`
	`> X`
	`     [,1] [,2]`
	`[1,]    1    3`
	`[2,]    2    4`

Lists	`> x`
	`[1] "one"   "two"   "three"`
	`> y`
	`     x1 x2 x3`
	`[1,]  1  2  3`
	`[2,]  2  3  4`
	`[3,]  3  4  5`
	`> L <- list(vector=x, matrix=y)`
	`> L`
	`$vector`
	`[1] "one"   "two"   "three"`
	
	`$matrix`
	`     x1 x2 x3`
	`[1,]  1  2  3`
	`[2,]  2  3  4`
	`[3,]  3  4  5`

Data Frames	`> D <- data.frame(x1, x2)`
	`> D`
	`  x1 x2`
	`1  1 AB`
	`2  2 BC`
	`3  3 CD`

Factors	`> clothingf <- factor(clothing)`
	`> clothingf`
	`[1] shirt   shirt   jeans   hat     vests   jackets vests`
	`Levels: hat jackets jeans shirt vests`

For review prior to first session

A series of R programming software tutorials are available here. Please review the following videos prior to class on Tuesday.

- 1.0 (Download, Install and Setup R and R Studio)
- 1.1 (Getting Started with R)
- 1.2 (Creating Vectors, Matrices and Other Intro Topics)
- 1.3 (Import data, copy data from excel to R, both .csv and .txt formats)

Additional resources

See the GH 811 website for additional resources for learning R. There you will find links to free texts, R manuals and other free resources you can use to supplement your training.