Getting StaRted

ECO 6416

Last Updated: 2022-08-19

Contents

Typically, I will show all the packages needed to compile this document here. For this document, I am going to layer them within the document

1 What is R?

R is a programming language designed for statistical analysis and data visualization. R can also be used as a general-purpose programming language, complete with a menu of data types, tools for conditional logic, repetition, functions and methods.

2 Uses of R

There are many things you can do with R, but for our class, we should only focus on how data is viewed, how to use functions, utilize packages, and ways find help.

2.1 Basic Operations

R can be used as a basic calculator.

```
5 + 8
```

```
## [1] 13
```

We can also store our operations into variables. We can reference those variables later.

```
unlucky <- 5 + 8
lucky <- unlucky - 6
```

2.2 Using Functions and Packages

Using a function is simple. These are some built-in functions

```
abs(-3)
```

```
## [1] 3
```

```
sqrt(lucky)
```

```
## [1] 2.645751
```

2.3 Installing and Using Packages

Sometimes, you need to use functions outside of the built-in ones. For an example, let's install the package swirl:

```
install.packages("swirl")
```

This command pulls and stores the package into your library. If you want to actually use it, you must pull it from your library:

```
library(swirl)
```

```
##
## | Hi! I see that you have some variables saved in your workspace. To keep
## | things running smoothly, I recommend you clean up before starting swirl.
##
## | Type ls() to see a list of the variables in your workspace. Then, type
## | rm(list=ls()) to clear your workspace.
##
## | Type swirl() when you are ready to begin.
```

`swirl` is an important package for beginners because it will show you how to do the basics step by step.

2.4 Help

Also if you need help with any commands, you can put a question mark before a command to understand how to use it.

```
?merge
```

You can also search by keywords: `??matrix`, `??"identity matrix"`

2.5 Handling Data

For this example we are going to use one of the embedded datasets within R.

```
mtcars
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
```

```
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

Looking at this output, you can see this dataframe has 32 rows, and 11 columns. You could also use the `dim(mtcars)` function We can do some basic operations on this dataframe.

2.5.1 Column Selection

If you are interested in looking at the horsepower data for each one of these vehicles, you can simply do it two different ways:

```
mtcars$hp
```

```
##  [1] 110 110  93 110 175 105 245  62  95 123 123 180 180 180 205 215 230  66  52
## [20]  65  97 150 150 245 175  66  91 113 264 175 335 109
```

```
mtcars[,4]
```

```
##  [1] 110 110  93 110 175 105 245  62  95 123 123 180 180 180 205 215 230  66  52
## [20]  65  97 150 150 245 175  66  91 113 264 175 335 109
```

2.5.2 Selecting Rows

You can simply select which row you wish by doing so:

```
mtcars[1,]
```

```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4  21   6  160 110  3.9 2.62 16.46  0  1    4    4
```

```
mtcars["Mazda RX4",]
```

```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4  21   6  160 110  3.9 2.62 16.46  0  1    4    4
```

2.5.3 Selecting Row and Column

You can bring both together.

```
mtcars$hp[1]
```

```
## [1] 110
```

```
mtcars[1,4]
```

```
## [1] 110
```

2.6 Creating New Column

You will often times need to transform one column into another. Suppose we want the square root of horsepower

```
sqrt(mtcars$hp)
```

```
##  [1] 10.488088 10.488088  9.643651 10.488088 13.228757 10.246951 15.652476
##  [8]  7.874008  9.746794 11.090537 11.090537 13.416408 13.416408 13.416408
## [15] 14.317821 14.662878 15.165751  8.124038  7.211103  8.062258  9.848858
## [22] 12.247449 12.247449 15.652476 13.228757  8.124038  9.539392 10.630146
## [29] 16.248077 13.228757 18.303005 10.440307
```

If we want to add this column to the overall dataset, we will need to assign it.

```
mtcars$sqrtHP <- sqrt(mtcars$hp)
```

3 Getting Your Bearings

A crucial part to using any software is that you are in the proper location. I typically think of folders within a computer as houses and the files are in each house. If you are in the wrong house, you cannot get the items in the other house.

```
getwd()
```

```
## [1] "C:/Users/jo585802/OneDrive - University of Central Florida/Documents/GitHub/ECO6416/01-GettingS
```

One benefit of the project environment is that it guarantees that you are in a specific spot. If you need to change locations, you can easily do that as well using `setwd()`.

An important thing to note is \ vs /. Many languages use \ to ignore the next action. This is called an escape character.

4 Importing Data

Depending upon the filetype, you may need special packages in order to import the data. For this example, since it is an excel file, we will need to import a new package.

```
library(readxl)

Wealth1percent <- read_excel("../Data/Wealth1percent.xlsx",
                             col_types = c("date", "numeric","numeric"))
```