RUG Lesson 2

# Summary

Last week, we got started with R and R Markdown/Notebook documents. This week, we will advance our skills with R, learn how to read in data, format a table, calculate returns and do our first graph.

# Introduction to vectors

R is a matrix oriented language and most of its instructions will work with scalars, vectors and matrices. This capability is a core strength and makes it powerful for working with data naturally represented with these type of structures.

A vector is just multiple numbers or character strings (I don't say "list", because that's a different concept in R we will cover later).

You can create a vector by typing the data in your code as arguments to the function "c", which I think is abbreviation for concatenate. Here are some examples of what you can do with a numeric vector.

x=c(1,3,5,7,9)  
x

## [1] 1 3 5 7 9

length(x) #how many numbers are in x

## [1] 5

x[4] #the fourth element of the vector

## [1] 7

sum(x) #what is the sum of the numbers in x

## [1] 25

mean(x) #what is the average value of the numbers in x

## [1] 5

sd(x) #the standard deviation of the numbers in x

## [1] 3.162278

x+x #pairwise addition of the numbers in x

## [1] 2 6 10 14 18

x\*x #pairwise multiplication of the numbers in x

## [1] 1 9 25 49 81

x%\*%x #vector muliplication of x times x (standard math definition from linear algebra)

## [,1]  
## [1,] 165

outer(x,x,"\*") #generate a multiplication table

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1 3 5 7 9  
## [2,] 3 9 15 21 27  
## [3,] 5 15 25 35 45  
## [4,] 7 21 35 49 63  
## [5,] 9 27 45 63 81

Here are some examples with characters.

a=c("Karl","Eric","Courtney","John","Kerry")  
length(a)

## [1] 5

a[c(4,1)] #the fourth and first (in that order) elements of the vector

## [1] "John" "Karl"

a[-2] #the vector excluding the second element

## [1] "Karl" "Courtney" "John" "Kerry"

grep("K+",a) #which names have a capital "K" in them

## [1] 1 5

grep("r{2}",a) #which name has two r's in it (last two lines use "regular expressions" for pattern matching -- google it if curious -- very powerful text manipulation tool)

## [1] 5

gsub("K","E",a) #replace capital "K" with "E"

## [1] "Earl" "Eric" "Courtney" "John" "Eerry"

tickers=c("IBM","AAPL","MSFT") # some well known tickers  
paste(tickers,"US EQUITY")

## [1] "IBM US EQUITY" "AAPL US EQUITY" "MSFT US EQUITY"

Now, let's work with some real data. I created a file of securities data in a file called "stockdat.csv". If you haven't already, copy this file in to your own directory. You also need to set your working directory by clicking at the top left of R Studio "Session/Set Working Directory/To Source File Location" to make sure you are reading and writing to the correct place.

Here is how you read the file:

stockdat=read.csv("stockdat.csv")  
head(stockdat) #displays the first 6 rows of the data

## X date AAPL.US.EQUITY IBM.US.EQUITY MSFT.US.EQUITY  
## 1 1 2006-12-29 12.1200 97.1500 29.8600  
## 2 2 2007-01-31 12.2471 99.1500 30.8600  
## 3 3 2007-02-28 12.0871 93.2400 28.2700  
## 4 4 2007-03-30 13.2729 94.5643 27.9689  
## 5 5 2007-04-30 14.2571 102.5399 30.0463  
## 6 6 2007-05-31 17.3130 107.3454 30.8994

tail(stockdat[,-1]) #displays the last 6 rows of the data excluding the first column

## date AAPL.US.EQUITY IBM.US.EQUITY MSFT.US.EQUITY  
## 116 2016-07-29 112.9015 195.9099 71.2207  
## 117 2016-08-31 115.5667 195.4952 72.6532  
## 118 2016-09-30 123.1368 195.4583 72.8302  
## 119 2016-10-31 123.6706 189.1091 75.7637  
## 120 2016-11-30 121.0019 201.3276 76.6867  
## 121 2016-12-30 126.8046 206.0065 79.0792

These are monthly stock prices (adjusted for dividends to calculate correct total return) for Apple, Microsoft and IBM. Right now this data is in a data frame (much more about this later). To keep this lesson about vectors, we will extract the price data for the stocks in three separate vectors. (Not explaining now why the following code works, but it does.)

apple=stockdat$AAPL.US.EQUITY  
ibm=stockdat$IBM.US.EQUITY  
microsoft=stockdat$MSFT.US.EQUITY