Exercise Set 1

T. Evgeniou

The purpose of this exercise is to become familiar with:

1. Basic statistics functions in R;
2. Simple matrix operations;
3. Simple data manipulations;
4. The idea of functions as well as some useful customized functions provided.

While doing this exercise we will also see how to generate replicable and customizable reports. For this purpose the exercise uses the R Markdown capabilities (see [Markdown Cheat Sheet](https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf) or a [basic introduction to R Markdown](http://rmarkdown.rstudio.com/authoring_basics.html)). These capabilities allow us to create dynamic reports. For example today's date is 2016-06-07 (you need to see the .Rmd to understand that this is *not* a static typed-in date but it changes every time you compile the .Rmd - if the date changed of course).

Before starting, make sure you have pulled the [exercise files](https://github.com/InseadDataAnalytics/INSEADAnalytics/tree/master/Exercises/Exerciseset1) on your github repository (if you pull the course github repository you also get the exercise set files automatically). Moreover, make sure you are in the directory of this exercise. Directory paths may be complicated, and sometimes a frustrating source of problems, so it is recommended that you use these R commands to find out your current working directory and, if needed, set it where you have the main files for the specific exercise/project (there are other ways, but for now just be aware of this path issue). For example, assuming we are now in the "MYDIRECTORY/INSEADAnalytics" directory, we can do these:

getwd()  
setwd("Exercises/Exerciseset1/")  
list.files()

**Note:** you can always use the help command in Rstudio to find out about any R function (e.g. type help(list.files) to learn what the R function list.files does).

Let's now see the exercise.

**IMPORTANT:** You should answer all questions by simply adding your code/answers in this document through editing the file ExerciseSet1.Rmd and then clicking on the "Knit HTML" button in RStudio. Once done, please post your .Rmd and html files in your github repository.

### Exercise Data

We download daily prices (open, high, low, close, and adjusted close) and volume data of publicly traded companies and markets from the web (e.g. Yahoo! or Google, etc). This is done by sourcing the file data.R as well as some helper functions in herpersSet1.R which also installs a number of R libraries (hence the first time you run this code you will see a lot of red color text indicating the *download* and *installation* process):

source("helpersSet1.R")  
source("dataSet1.R")

[1] "ticker SPY ..." [1] "ticker AAPL ..." [1] "ticker GS ..." [1] "ticker TSLA ..." [1] "ticker FB ..." [1] "ticker MSFT ..."

For more information on downloading finance data from the internet as well as on finance related R tools see these starting points (there is a lot more of course available):

* [Some finance data loading tools](http://www.r-bloggers.com/r-code-yahoo-finance-data-loading/)
* [Connecting directly to Bloomberg](http://www.r-bloggers.com/rblpapi-connecting-r-to-bloomberg/)
* [Some time series plot tools](http://www.r-bloggers.com/plotting-time-series-in-r-using-yahoo-finance-data/)
* [Various finance code links](https://cran.r-project.org/web/views/Finance.html)
* [More links](http://blog.revolutionanalytics.com/2013/12/quantitative-finance-applications-in-r.html)
* [Even more links](http://www.r-bloggers.com/financial-data-accessible-from-r-part-iv/)
* Of course endless available code (e.g. like this one that seems to [get companies' earnings calendars](https://github.com/gsee/qmao/blob/master/R/getCalendar.R))

#### Optional Question

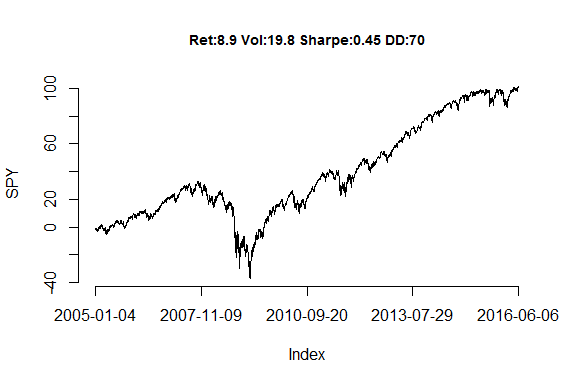
1. Can you find some interesting finance related R package or github repository? **Your Answers here:**

### Part I: Statistics of S&P Daily Returns

We have 2875 days of data, starting from 2005-01-04 until 2016-06-06. Here are some basic statistics about the S&P returns:

1. The cumulative returns of the S&P index during this period is 101.5 108.4%.
2. The average daily returns of the S&P index during this period is 0.0350.028%;
3. The standard deviation of the daily returns of the S&P index during this period is 1.2481.249%;

Here are returns of the S&P in this period (note the use of the helper function pnl\_plot - defined in file helpersSet1.R):



#### Questions

1. Notice that the code also downloads the returns of Apple during the same period. Can you explain where this is done in the code (including the .R files used)?
2. What are the cumulative, average daily returns, and the standard deviation of the daily returns of Apple in the same period?
3. *(Extra points)* What if we want to also see the returns of another company, say Yahoo!, in the same period? Can you get that data and report the statistics for Yahoo!'s stock, too?

**Your Answers here:** 1. The data is from the file 'DataSet1.R' 2. Cum. return: 574.8 'r round(100*sum(StockReturns[,2]),1)' Av daily return: 0.148% 'r round(100*mean(StockReturns[,2]),3)' St. dev. of daily return:2.425% 'r round(100\*sd(StockReturns[,2]),3)'

3.add my tickers c("SPY","AAPL","YHOO","GOOG") Cumulative returns: 275.1, Average daily returns: 0.071, Standard deviation: 3.064

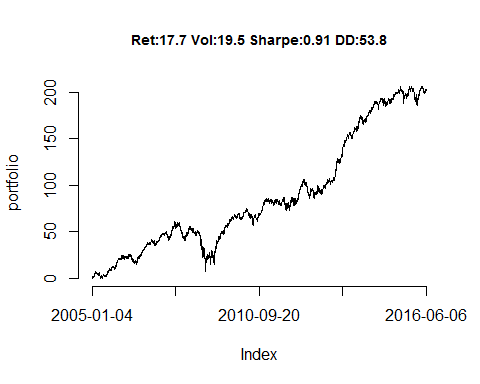
### Part II: Simple Matrix Manipulations

For this part of the exercise we will do some basic manipulations of the data. First note that the data are in a so-called matrix format. If you run these commands in RStudio (use help to find out what they do) you will see how matrices work:

class(StockReturns)  
dim(StockReturns)  
nrow(StockReturns)  
ncol(StockReturns)  
StockReturns[1:4,]  
head(StockReturns,5)  
tail(StockReturns,5)

We will now use an R function for matrices that is extremely useful for analyzing data. It is called *apply*. Check it out using help in R.

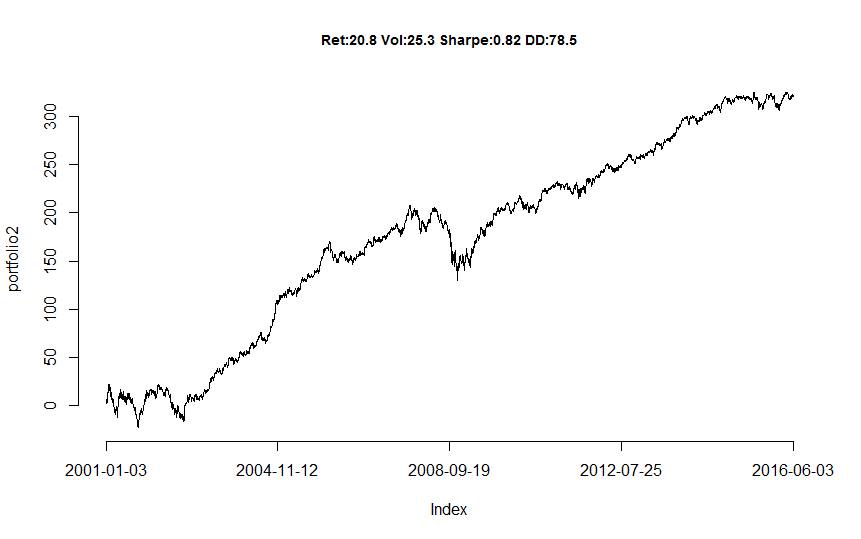
For example, we can now quickly estimate the average returns of S&P and Apple (of course this can be done manually, too, but what if we had 500 stocks - e.g. a matrix with 500 columns?) and plot the returns of that 50-50 on S&P and Apple portfolio:



We can also transpose the matrix of returns to create a new "horizontal" matrix. Let's call this matrix (variable name) transposedData. We can do so using this command: transposedData = t(StockReturns).

#### Questions

1. What R commands can you use to get the number of rows and number of columns of the new matrix called transposedData?
2. Based on the help for the R function *apply* (help(apply)), can you create again the portfolio of S&P and Apple and plot the returns in a new figure below?

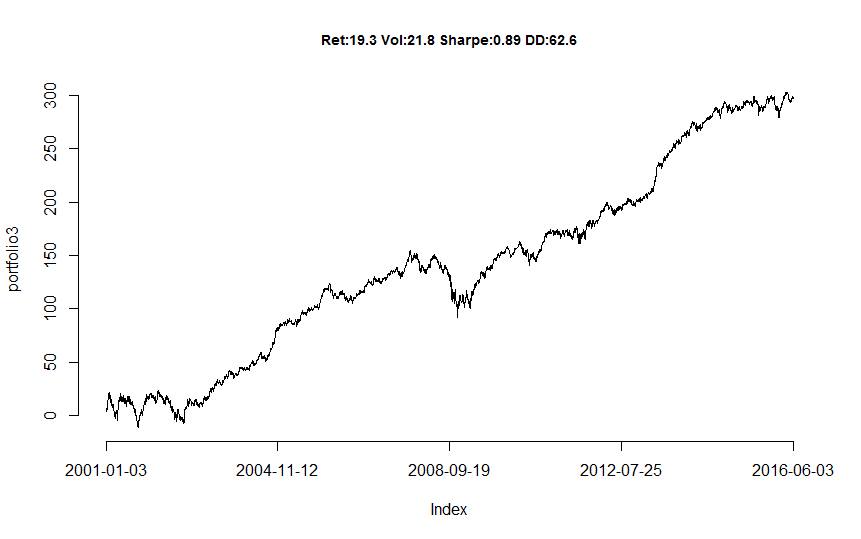
**Your Answers here:** 1. ncol(transposedData) [1] 3878 nrow(transposedData) [1] 2 2. Portfolio (of SPY, AAPL, YHOO, GOOG): 

### Part III: Reproducibility and Customization

This is an important step and will get you to think about the overall process once again.

#### Questions

1. We want to re-do all this analysis with data since 2001-01-01: what change do we need to make in the code (hint: all you need to change is one line - exactly 1 number! - in data.R file), and how can you get the new exercise set with the data since 2001-01-01?
2. *(Extra Exercise)* Can you get the returns of a few companies and plot the returns of an equal weighted portfolio with those companies during some period you select? St

**Your Answers here:** 1) The source code is set-up for data since 2001-01-01 (line 9, startdate: startDate = "2001-01-01"). New data would require to run the entire dataSet1.R again (mark all lines, click "Run") Portfolio (of SPY, AAPL, YHOO, GOOG, TSLA, MSFT): 

### Part IV: Read/Write .CSV files

Finally, one can read and write data in .CSV files. For example, we can save the first 20 days of data for S&P and Apple in a file using the command:

write.csv(StockReturns[1:20,c("SPY","AAPL")], file = "twentydays.csv", row.names = TRUE, col.names = TRUE)

Do not get surpsised if you see the csv file in your directories suddenly! You can then read the data from the csv file using the read.csv command. For example, this will load the data from the csv file and save it in a new variable that now is called "myData":

myData <- read.csv(file = "twentydays.csv", header = TRUE, sep=";")

Try it!

#### Questions

1. Once you write and read the data as described above, what happens when you run this command in the console of the RStudio: sum(myData != StockReturns[1:20,])
2. *(Extra exercise)* What do you think will happen if you now run this command, and why:

myData + StockReturns[1:40,])

**Your Answers here:** Provides the sum of cells in myData matrix, which is identical ("!=") to the values in the cells of StockReturns

### Extra Question

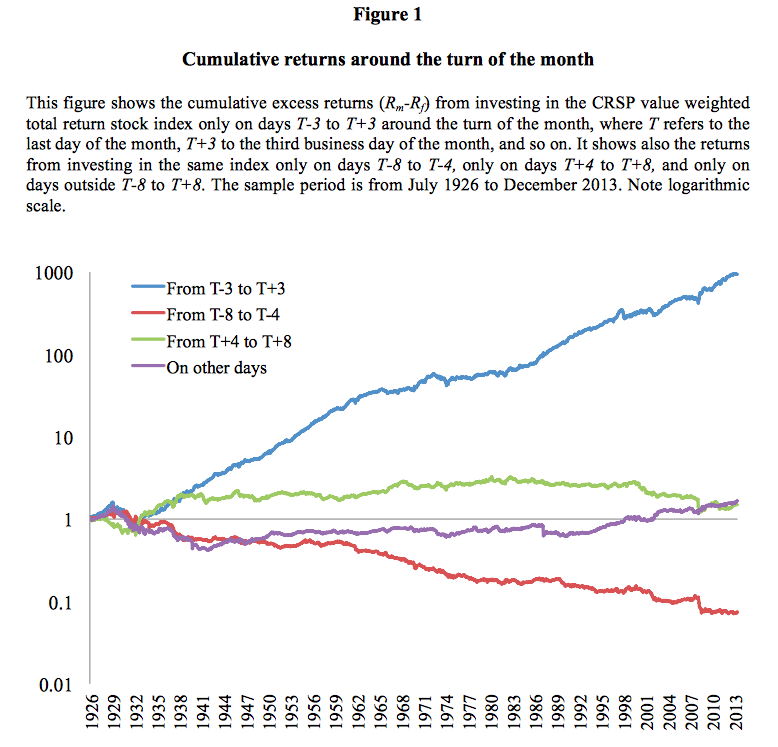
Can you now load another dataset from some CSV file and report some basic statistics about that data?

### Creating Interactive Documents

Finally, just for fun, one can add some interactivity in the report using [Shiny](http://rmarkdown.rstudio.com/authoring_shiny.html).All one needs to do is set the eval flag of the code chunk below (see the .Rmd file) to "TRUE", add the line "runtime: shiny" at the very begining of the .Rmd file, make the markdown output to be "html\_document", and then press "Run Document".

sliderInput("startdate", "Starting Date:", min = 1, max = length(portfolio),   
 value = 1)  
sliderInput("enddate", "End Date:", min = 1, max = length(portfolio),   
 value = length(portfolio))  
  
renderPlot({  
 pnl\_plot(portfolio[input$startdate:input$enddate])  
})

### Endless explorations (optional homework)

This is a [recent research article](http://poseidon01.ssrn.com/delivery.php?ID=851091091009083082092113118102076099034023058067019062072066007100008111081022102123034016097101060099003106125099002090116089026058012038004030005113111105079028059062024121067073126072090091089069014121102110107075029090001011087028011082124103085&EXT=pdf) that won an award in 2016. Can you implement a simple strategy as in Figure 1 of this paper? You may find these R commands useful: names, which, str\_sub,diff,as.vector, length, pmin, pmax, sapply, lapply,Reduce,unique, as.numeric, %in% 

What if you also include information about bonds? (e.g. download the returns of the the ETF with ticker "TLT") Is there any relation between stocks and bonds?

**Have fun**