12. Would you invest in the same way if this was your account and your money? What would be different?
13. Did you invest in exchanges other than the US?
14. If so, in which exchanges, besides the US, did you buy securities?
15. Are you an international student?
16. Are you a natural born US citizen?
17. If not, what is your nationality?
18. On a scale of one to ten, with one being the least and ten the most, rate how this simulation aided in your knowledge of investments.
19. On a scale of one to ten, with one being the least and ten the most, rate how this simulation aided in your knowledge of derivatives.
20. Case studies were assigned based on activity in group portfolios. On a scale of one to ten, with one being the least and ten the most, rate how the case studies aided in the simulation.
21. If this were an individual exercise, with each person investing their own portfolio, would the cost of the simulation be a problem?
22. Is there anything you would change about the simulation?

**Exhibit 3. Peer Review given to graduate students on StockTrak stock market simulation game.**

List all members in your group (include yourself). Complete a peer review sheet for each member other than yourself.

Member to be reviewed __________________________________________________________

1. On a scale of one to ten, rate the member's contribution to the exercise.
2. What percentage of the security selections were the idea of this member?
3. Did this member obstruct the simulation in any way? Explain.
4. Did the member do any of the trading in the account?
5. Did the member "take over" the simulation, ignoring the input of other members?
6. If the overall grade for your group was an "A," what grade would you award the member

# Empirical Finance in R: An Introduction

*Omid Sabbaghi*

*This article provides an introduction to the R statistical programming language with teaching applications in finance. R is a statistical software environment for data analysis and graphics. Specific applications presented in this study include the calculation of descriptive statistics for financial returns of 2 stocks and the S&P 500 Index. The R language is useful in today's finance classrooms because of its open-source coding environment, ease of learning, and high number of financial data analysis packages contributed by users worldwide.*

## INTRODUCTION

Empirical finance has attracted much attention in recent years. Modeling financial data and the empirical implementation of financial econometric models provide unique insights in regards to the behavior of securities and financial instruments trading in today's global markets. Research in finance continues to evolve rapidly and new results appear regularly. Importantly, financial time series is an empirical discipline and forms the foundation for making inference. The importance of econometric methods in finance has increased in recent years because of the increase in systematic data collection. That is, our financial markets provide a data-rich environment for the finance student. Thus, modeling financial returns requires knowledge of a programming language that is easy to use.

To date, there has been an absence of research focusing on alternative financial technology applications, thus providing the motivation for the present study. In today's finance classrooms, fewer than 1% of AACSB-accredited business schools offer a dedicated course in financial technology applications (Payne and Tanner 2011). Financial technology courses in business schools' finance curriculum are becoming increasingly important since increases in the availability of financial data and technology have increased the speed at which trades and investment decisions take place. Importantly, business graduates exhibiting a higher competency in technology applications will have a significant comparative advantage in career placement and future success (Truitt forthcoming).

The R software environment emerges as a front runner in teaching finance applications. Specifically, R is a language and environment for statistical computing and graphics that is based on the "S" language developed at Bell Labs. Using R in today's finance classrooms is effective since finance students can implement the commands on their laptops at the same time the instructor is demonstrating the commands in the classroom or finance lab. By having students implement the R commands in real-time, they gain the ability to analyze financial data via learning by doing, a valuable teaching technique. In other words, the R environment allows for experiential learning in financial data analysis.

It is estimated that nearly 250,000 people work with R on a regular basis, and is considered the second language for individuals recently emerging from graduate school (Vance 2009). Additionally, its popularity stems from the ease of using it, its flexibility in coding estimation techniques, and additionally, its unique graphical representation of data. Zeileis and Koenker (2008) argue that the R language provides a rich toolbox for matrix-based computations and that it is attractive for teaching because of its open-source license. In addition, the latter study acknowledges additional benefits of using R, including: (1) the ease by which development efforts can be undertaken; and (2) standards and templates for documentation, version control, and consistency checking. Gaining proficiency with R is useful for finance-related data applications as well as for quantitative methods in related business disciplines, such as accounting, decision sciences, management, and marketing: R is valuable for business undergraduates and MBA students in any field concentration and is an important tool for their future careers.

This study introduces the statistical analysis of financial data in the R computing environment. Using financial returns for 2 companies and the S&P 500 index, we demonstrate how to load the financial returns data in R and calculate sample descriptive statistics, such as sample average returns, medians, and standard deviations.

## THE R ENVIRONMENT

The R statistical environment was developed in the 1990s at the University of Auckland in New Zealand. R is open-source software, meaning that it is free for anyone to use and modify. Its capabilities are extended via user-submitted packages which are collections of R functions, data, and compiled code in well-defined format. These packages massively extend the functionality of R. The base R software may be downloaded freely at R's homepage which is located at http://www.r-project.org/. The contributed user-submitted packages may be downloaded freely from the Comprehensive R Archive Network (CRAN) via the Internet at http://cran.r-project.org/web/packages. CRAN refers to the network of FTP and Web servers around the world. There are currently more than 70 mirror sites that comprise CRAN.

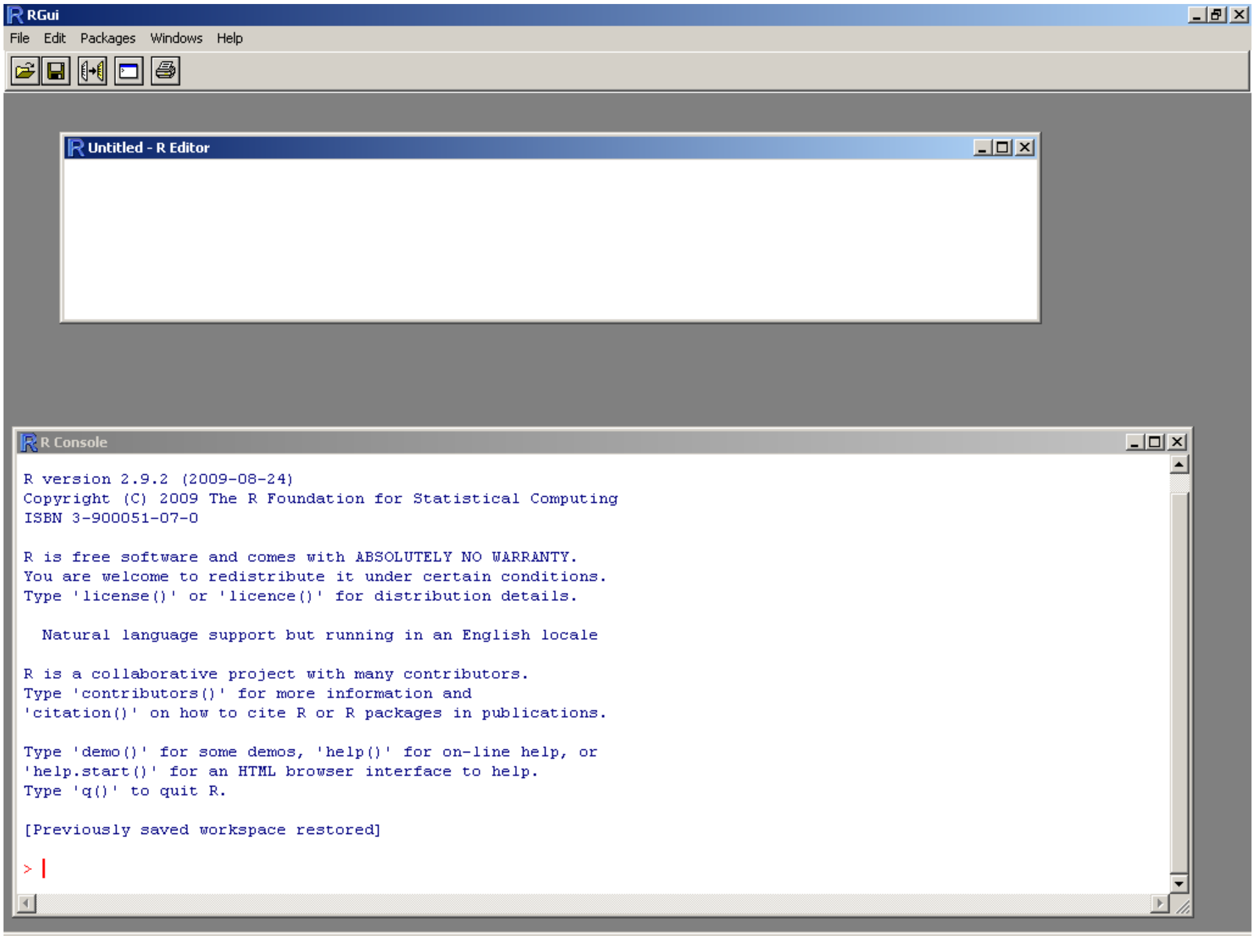
The R environment is attractive for a variety of reasons. R is free and open-source software. Additionally, R exhibits state-of-the-art data analysis and provides a platform for programming new methods. In terms of operating system, R runs on Windows, Linux, as well as Mac OS X. There is an enormous user base and allows for reproducible research.

R can be used directly at the command line or using its inbuilt script editor. Figure 1 presents the graphical user interface when loading R in Microsoft Windows. The first sub screen that appears is the R Console, labeled A in Figure 1. In the R Console, line commands may be manually entered. Statements consist of functions, assignments, and comments. The output is immediately shown in the Console, thus providing an interactive working session. Thus, the R language is interactive and interpreted as well as functional and object oriented. The assignment operator is ← and everything is accomplished through functions operating on objects. An object within R is anything that can be assigned to a variable name and includes: constants, data structures, functions, and even graphs. Objects are kept in memory. However, defined objects in an R session are usually erased from its working memory upon closing the program. Thus, saving commands in an R script file is important for later use. A new script file, labeled B in Figure 1, is initialized using the File menu within R. Commands may be entered in the script file and saved. The commands in the R script file may be implemented by highlighting the command lines or code in the script file and pressing Ctrl + R. The output is then immediately shown in the Console. The R Editor allows for the subsequent revision of the code contained within the R script file.

## SAMPLE DATA & EMPIRICAL METHODOLOGY

We download historical financial data for Google (GOOG), Ford (F), and the S&P 500 index (^GSPC) from Yahoo! Finance. Specifically, we download price level data at the weekly frequency from January 2013 through December 2013 and compute the continuously compounded returns by taking the first difference in log prices. In the present study, we compute several descriptive statistics for the weekly returns data, namely the sample average, median, standard deviation, minimum, maximum, and the t-ratio for the sample average.

**Figure 1. R Graphical User Interface**




## R AND DESCRIPTIVE STATISTICS

We illustrate the value of using R in the classroom by loading the data in R and subsequently calculating descriptive statistics for the returns of our sample companies and the S&P 500 index. The first step in our empirical analysis consists of loading the original downloaded data and computing the weekly returns data in R. Suppose that our original data, downloaded from Yahoo! Finance, resides in an Excel file. One may load this data in R using several steps. The first step is to save the original Excel file as a Text (Tab-Delimited) (*.txt) file within Excel. Following this step, we load the original data for Ford, Google, and the S&P500 Index in R with the following commands, respectively:

```
FORDdata <-
    read.delim("D:\\data\\FORD.txt",header=
    T)

GOOGLEdata <-
    read.delim("D:\\data\\GOOGLE.txt",heade
    r=T)

SP500data <-
    read.delim("D:\\data\\SP500.txt",header
    =T)
```

In the commands above, we are reading in the tab-delimited text data file from its corresponding location on the computer via the `read.delim` command. The `header=T` argument is allowing R to read the first row of the data as column names. By applying the `read.delim` command, we are defining the original data matrices which contain the calendar date in the first column and associated price level and trading volume data in the remaining columns, along with their respective column names in the first row. As a means of visualizing the data, we extract the first five rows of the original data matrix for Ford stock in the R Console via command,

```
FORDdata[1:5,],
```

and report the results in Table 1:

**Table 1. Output from FORData[1:5,] Command**

| | Date | Open | High | Low | Close | Volume | Adj.Close |
|---|---|---|---|---|---|---|---|
| 1 | 1/2/2013 | 13.23 | 13.70 | 13.00 | 13.57 | 83743100 | 13.11 |
| 2 | 1/7/2013 | 13.52 | 14.07 | 13.20 | 14.00 | 55863600 | 13.53 |
| 3 | 1/14/2013 | 14.04 | 14.30 | 13.86 | 14.11 | 48623700 | 13.64 |
| 4 | 1/22/2013 | 14.06 | 14.19 | 13.64 | 13.68 | 47382000 | 13.22 |
| 5 | 1/28/2013 | 13.49 | 13.82 | 12.67 | 13.02 | 70599600 | 12.67 |

The output above confirms the time-series structure of the data matrix, containing the open, high, low, close, and adjusted closing prices as well as trading volume levels. We proceed to calculate continuously compounded returns for Ford stock by calculating the first difference in log adjusted closing prices. The continuously compounded returns for Ford stock are calculated and defined in R via the following command:

```
FORDreturns <-
    diff(log(FORDdata[,"Adj.Close"]))
```

In the command above, we first apply the `log` command to the column containing the adjusted closing prices and then apply the `diff` command in order to take the first difference of the log prices. Importantly, we identify the column containing the adjusted closing prices by referencing the column name in the original data matrix via the command `FORDdata[,"Adj.Close"]`. Generally, the command `data[i,j]` refers to the `i`th row and `j`th column of the defined `data` object in R. Similarly, we calculate the weekly returns for Google stock and the S&P 500 index, respectively, using the following commands:

```
GOOGLEreturns <-
    diff(log(GOOGLEdata[,"Adj.Close"]))

SP500returns <-
    diff(log(SP500data[,"Adj.Close"]))
```

Having defined our returns data within the R computing environment, we proceed to calculate descriptive statistics for our weekly returns data. For illustrative purposes, we calculate the descriptive statistics for Ford's returns data using the following commands in R:

```
mean(FORDreturns)
median(FORDreturns)
sd(FORDreturns)
min(FORDreturns)
max(FORDreturns)
```

The above commands are calculating the sample average, median, standard deviation, minimum, and maximum returns, respectively. A t-test of the null hypothesis that Ford's average returns is equal to zero is conducted with the `t.test` command in R, shown below along with its corresponding output:

```
> t.test(FORDreturns)

        One Sample t-test

data:  FORDreturns
t = 0.6466, df = 51, p-value = 0.5208
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.006279359  0.012245932
sample estimates:
  mean of x
0.002983287
```

We observe from the output above that the t-ratio is not statistically significant and conclude that the Ford weekly returns are statistically indistinguishable from zero. Note that we are able to extract the t-ratio from the sample output by using double brackets via the command: `t.test(FORDreturns)[[1]]`. Using the mentioned commands above, we calculate descriptive statistics for Google and the S&P500 Index and compile our results in Table 2.

Table 2 reports descriptive statistics for the returns of Ford (F), Google (GOOG), and the S&P 500 index (^GSPC). The sample mean, median, standard deviation (SD), minimum, and maximum are reported. For the sample mean return, t-ratios for testing the null hypothesis of zero average returns are presented in parentheses. The sample period is January 2013 through December 2013. Weekly data is obtained from Yahoo! Finance, accessible via finance.yahoo.com.

In Table 2 one observes that average weekly returns are statistically distinguishable from zero for the case of Google stock and the S&P500 index, suggesting an upward trend in returns over the January 2013 – December 2013 time period. In addition, we observe that there is dispersion present in our returns data. Specifically, we find that the annualized return standard deviations range from 9 percent to 24 percent. Minimum weekly returns range from -2.1 percent through -7.3 percent while maximum weekly returns range from 2.9 percent through 14.8 percent across assets during this time period.

**Table 2. Descriptive Statistics**

| Jan 13 - Dec 13 | Ford | Google | S&P500 |
|---|---|---|---|
| Mean | 0.0030 | 0.0080 | 0.0045 |
| t-ratio | (0.65) | (1.89)* | (2.54)** |
| Median | 0.0028 | 0.0083 | 0.0056 |
| SD | 0.0333 | 0.0306 | 0.0127 |
| Minimum | -0.0731 | -0.0491 | -0.0214 |
| Maximum | 0.0839 | 0.1483 | 0.0292 |

***, **, * indicate statistical significance at the 0.01, 0.05, and 0.10 level, respectively.

## CONCLUDING REMARKS

This study investigates the analysis of financial returns data in the R computing environment. Specifically, we provide an overview of the R platform and the importance of its user-friendly programming language in calculating descriptive statistics. The R environment is open source software that is flexible, user-friendly, and allows for a straightforward statistical analysis that is appropriate for the classroom. R is also a very powerful statistical package that students can continue to use in their business careers after graduation as well.

Empirical characteristics of sample financial returns are examined in this study. Specifically, we demonstrate how to calculate sample averages, medians, maximums, minimums, and standard deviations when confronted with returns data. In addition, we show how to conduct a simple t-test when assessing whether returns are statistically indistinguishable from zero.

Several directions lie ahead for teaching finance applications of the R computing environment. Firstly, the R environment can be utilized to conduct asset pricing in the finance classroom. Pricing models such as the Capital Asset Pricing Model (CAPM) and the 3-factor model of Fama and French (1993) can be estimated using historical data. The pricing model estimates can then be utilized to forecast expected returns. Second, the power of the R environment for finance can be extended with user-contributed packages that allow the finance student to predict volatility. In other words, the R computing environment is capable of estimating classical volatility models such as the GARCH model of Bollerslev (1986). We leave these topics for future research.

## REFERENCES

Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31, No. 3, 307-327.

Fama, E.F., and K.R. French (1993). Common Risk Factors in the Returns on Stocks and Bonds. *Journal of Financial Economics*, 33, 3-56.

Payne, J.D., and G. Tanner (2011). Experiential Learning and Finance: A Hands-on-Approach to Financial Modeling. *Journal of Financial Education*, 37, No. 3/4, 82-100.

Truitt, D.L. (forthcoming). The Effect of Training and Development on Employee Attitude as it Related to Training and Work Proficiency. *SAGE Open*.

Vance, A. 2009, January 6. Data Analysts Captivated by R's Power. The New York Times. Retrieved from http://www.nytimes.com

Zeileis, A. and R. Koenker. (2008). Econometrics in R: Past, Present, and Future. *Journal of Statistical Software*, 27, No.1, 1-5.

*Omid Sabbaghi is an Associate Professor of Finance at the University of Detroit Mercy.*