A Peek into Quantitative Finance Applications in R

Demi Liu

10/29/2017

Background

Mathematical finance, also known as quantitative finance, is a field of applied mathematics, concerned with financial markets. As its popularity continues to grow in recent time, more and more programming languages are applied in the field of Quantitative Finance to forecast future returns, build financial models, and estimate prices. R's ability to facilitate rapid data analysis and visualization, its great number of available functions and algorithms and the ease with which it can interface to new data sources and other computing environments has made it a flexible tool that evolves and adapts at a pace that matches developments in the financial industry.

Motivation

Hi there! I am an Economics and Statistics major with an interest in Quantitative Finance. However, though I am very interested in this field, I have little exposure to it, and I have no idea how the process of analyzing data, building models, and extracting predictions work. It always sounds super fancy to me, and this post is an awesome opportunity for me to utilize what I have learned in class and get some hand-on experiences with Quantitative Finance. In R, many functions and packages are useful for financial analysis. The rgl package to do interactive 3-D graphs and the nortest package to test normality were mentioned above. There is also an online list of packages useful for empirical finance, see Eddelbuettel (2009) if you are interested to learn more.

Introduction

This post will be discussing multiple functions and steps needed to be taken when analyzing financial data using r. Some of the functions we have seen them in the lecture, and I would like to extend to talk about father applications of them. Some of them might be unfamiliar for you, and I hope that by reading this post you can get some insights into them. Some procedures and functions used in this post are:

- Extracting the Data Using *which( )*
- Plot the Data Using *plot( )*
- Calculate Simple Returns
- Graphically Presnt Simple Returns
- Analyze Data Trends Based on the Model We Built

At the end of this post, I will be creating a very simple financial model: the Simple Returns of the Starbucks stock. And I will dive deeper into some data frame manipulation techniques to clean, arrange, extract, and summarize data. In the end, you should expect to see a data visualization of the changes in monthly returns and get useful information and build further assumptions based on that.

Hopefully, after reading this post you would learn more about how to set up a financial forcast model in R.

A Few Things to Keep in Mind

Main Goal

The main goal of this post is to show how to calculate, analyze and plot simple returns in R.

Data and Models Used

- The data that will be used in the post is the monthly stock returns of Starbucks (ticker: SBUX).
- The model I will be building is based on the Simple Compouded Returns.

Examples of Procedures

1. Load the monthly Starbucks return data

```
# Assign the URL to the CSV file
data_url <- "http://assets.datacamp.com/course/compfin/sbuxPrices.csv"

# Load the data frame using read.csv
sbux_df <-read.csv(file = data_url, header = TRUE, stringsAsFactors = FALSE)
```

2. Get a feel for the data

- The `str()` function compactly displays the structure of an R object. It is arguably one of the most useful R functions.
- The `head()` and `tail()` functions shows you the first and the last part of an R object, respectively.
- The `class()` function shows you the class of an R object.

```
# Check the structure of sbux_df
str(sbux_df)
```

```
## 'data.frame':    181 obs. of  2 variables:
##  $ Date     : chr  "3/31/1993" "4/1/1993" "5/3/1993" "6/1/1993" ...
##  $ Adj.Close: num  1.13 1.15 1.43 1.46 1.41 1.44 1.63 1.59 1.32 1.32 ...
```

```
# Check the first and last part of sbux_df
head(sbux_df)
```

```
##        Date Adj.Close
## 1 3/31/1993      1.13
## 2  4/1/1993      1.15
## 3  5/3/1993      1.43
## 4  6/1/1993      1.46
## 5  7/1/1993      1.41
## 6  8/2/1993      1.44
```

```
tail(sbux_df)
```

```
##          Date Adj.Close
## 176 10/1/2007     25.37
## 177 11/1/2007     22.24
## 178 12/3/2007     19.46
## 179  1/2/2008     17.98
## 180  2/1/2008     17.10
## 181  3/3/2008     16.64
```

```
# Get the class of the Date column of sbux_df
class(sbux_df$Date)
```

```
## [1] "character"
```

3. Extract the price data

```
closing_prices <- sbux_df[, "Adj.Close", drop = FALSE]
```

4. Find indices associated within a peroid of time

It will often be useful to select stock data between certain dates.

The `which()` function returns the indices for which a condition is TRUE, and it will be used to find some prices.

```r
# Find indices associated with the dates 3/1/1994 and 3/1/1995
index_1 <- which(sbux_df$Date == "3/1/1994")
index_2 <- which(sbux_df$Date == "3/1/1995")

# Extract prices between 3/1/1994 and 3/1/1995
some_prices <- sbux_df[index_1:index_2, "Adj.Close"]
```

5. Subset directly on dates

When analyzing stock returns, it would be helpful if we could select data from a specific day.

When you create a data frame that has the dates of the stock price as row names, you can select the price on a specific day much more easily.

```r
# Create a new data frame that contains the price data with the dates as the row names
sbux_prices_df <- sbux_df[, "Adj.Close", drop = FALSE]
rownames(sbux_prices_df) <- sbux_df$Date
head(sbux_prices_df)
```

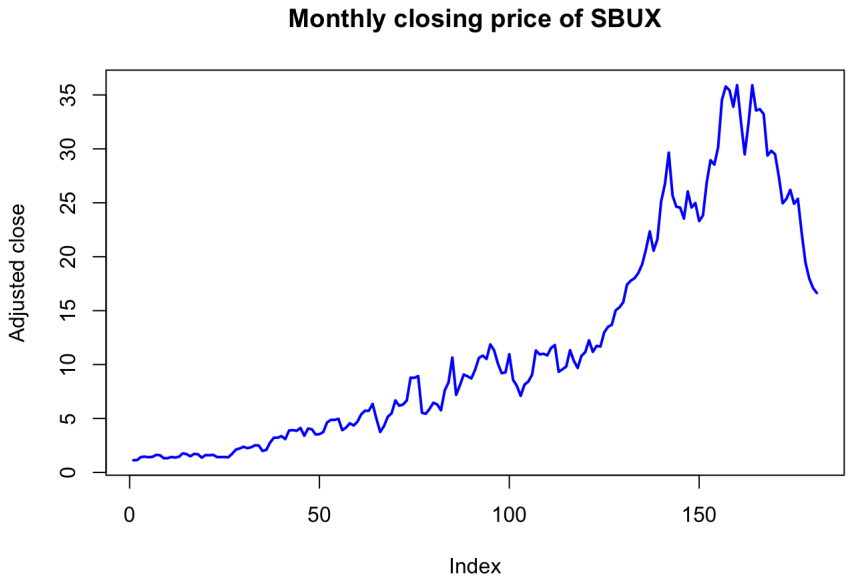
```
##           Adj.Close
## 3/31/1993      1.13
## 4/1/1993       1.15
## 5/3/1993       1.43
## 6/1/1993       1.46
## 7/1/1993       1.41
## 8/2/1993       1.44
```

```r
# With Dates as rownames, you can subset directly on the dates.
# Find indices associated with the dates 3/1/1994 and 3/1/1995.
price_1 <- sbux_prices_df["3/1/1994", ]
price_2 <- sbux_prices_df["3/1/1995", ]
```

6. Plot the price data

We will plot the Starbucks closing prices as a function of time using the `plot()` function

```r
# Now add all relevant arguments to the plot function below to get a nicer plot
plot(sbux_df$Adj.Close, type = "l", col = "blue",
     lwd = 2, ylab = "Adjusted close",
     main = "Monthly closing price of SBUX")
```



7. Calculate simple returns

The simple return is calculated by the formula:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

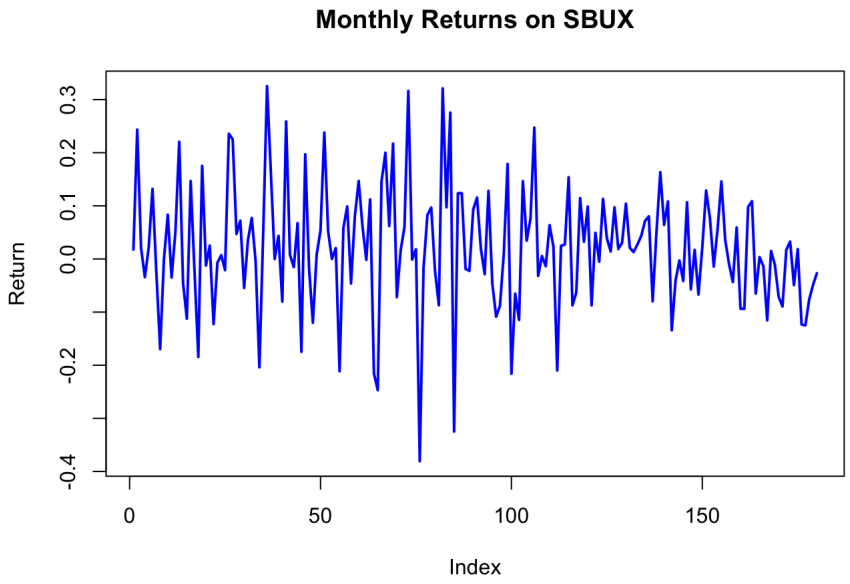
where R_t is the simple returns, P_t denotes the stock price at the end of month t, and P_{t-1} reprents the stock price at the end of month t - 1.

```r
wsbux_prices_df <- sbux_df[, "Adj.Close", drop = FALSE]

# Denote n the number of time periods:
n <- nrow(sbux_prices_df)
sbux_ret <- (sbux_prices_df[2:n, 1] - sbux_prices_df[1:(n - 1), 1]) / sbux_prices_df[1:(n - 1), 1]

# Notice that sbux_ret is not a data frame object
class(sbux_ret)
```

```
## [1] "numeric"
```

8. Graphically present the simple compounded returns

```
# Plot the returns
plot(sbux_ret, type = "l", col = "blue", lwd = 2, ylab = "Return", main = "Monthly Returns on SBUX")
```



Conclusion

This post is a very simple example of how r could manipulate and analyze data to draw financial conclusions. The Monthly Returns of SBUX we just got could be put into many future uses such as predicting the future returns, being compared with the continuously compounded returns to make investment decisions, or getting an overview of how one company is doing by looking at its stock prices fluctuations. And as mentioned in the post, many functions and data analyzing skills such as functions plot(), which(), data frame manipulation techniques, and connecting different data sets are applied. As some of them are mentioned in classes, I hope you learn something new in this post.

Take-home Message

1. Some basic r functions to analyze financial data are plot(), data frame related functions…
2. Before we jump into building the financial models, it is important for us to take a look at the data and extract useful information from it. For example, in the case of Starbucks stocks, we could use which(), drop(), head(), tail() to get the data in a certain period of time.
3. The plot() function is a super useful tool in the data visualization part of financial analysis. And the result of it could be applied into many further research.

References

- https://en.wikipedia.org/wiki/Rate_of_return#Comparisons_between_various_rates_of_return
- https://www.r-bloggers.com/r-and-finance/
- http://www.investopedia.com/exam-guide/cfa-level-1/quantitative-methods/discrete-continuous-compounding.asp
- *Mastering R for Quantitative Finance* by Edina Berlinger and Ferenc Illes
- http://assets.datacamp.com/course/compfin/sbuxPrices.csv
- https://stackoverflow.com/questions/1295955/what-is-the-most-useful-r-trick
- https://www.datacamp.com/home
- http://rmarkdown.rstudio.com/authoring_basics.html

Final Words

Thanks for bearing with me and taking time to read until the end! It is the first time I use outside data to do analysis myself, and I did learn a lot from this experience. I hope you also learned something after reading my post, and feel free to reach out to me if you have any further questions! XDDD