ECON 147 Homework 1

Answer Keys

Part I: Return calculations

Consider the following (actual) monthly adjusted closing price data for Starbucks stock over the period December 2011 through December 2012

End of Month Price Data for Starbucks Stock	
December, 2011	\$44.89
January, 2012	\$46.76
February, 2012	\$47.55
March, 2012	\$54.73
April, 2012	\$56.17
May, 2012	\$53.91
June, 2012	\$52.37
July, 2012	\$44.47
August, 2012	\$48.91
September, 2012	\$50
October, 2012	\$45.26
November, 2012	\$51.36
December, 2012	\$53.1

1. Using the data in the table, what is the simple monthly return between the end of December, 2011 and the end of January 2012? If you invested \$10,000 in Starbucks at the end of December 2011, how much would the investment be worth at the end of January 2012?

   This is a one month investment. The simple return between December and January is

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{\$46.76 - \$44.89}{\$44.89} = 0.0417 \text{ or } 4.17\%.$$

   The future value of \$10,000 *is then*

$$FV = \$10,000 \times (1 + R_t) = \$10,000 \times (1 + 0.0417) = \$10,417.$$

2. Using the data in the table, what is the continuously compounded monthly return between December, 2011 and January 2012? Convert this continuously compounded return to a simple return (you should get the same answer as in part a).

   The continuously compounded return is defined as

   $$r_t = \ln(1 + R_t) = \ln(P_t/P_{t-1})$$

   Using $R_t = 0.0417$ *gives*

   $$r_t = \ln(1 + 0.0417) = 0.04085.$$

   Notice that

   $$R_t = e^{r_t} - 1 = e^{0.04085} - 1 = 0.0417.$$

3. Assuming that the simple monthly return you computed in part (1) is the same for 12 months, what is the annual return with monthly compounding?

   The annual return assuming $R_t = 0.0417$ *every month for a year is*

   $$R_t(12) = R_A = (1 + R_t)^{12} - 1 = (1 + 0.0417)^{12} - 1 = 0.6327 \text{ or } 63.27\%.$$

4. Assuming that the continuously compounded monthly return you computed in part (2) is the same for 12 months, what is the continuously compounded annual return?

   The annual continuously compounded return assuming $r_t = 0.0417$ *every month for a year is*

   $$r_t(12) = r_A = 12 \times r_t = 12 \times (0.04085) = 0.4902 \text{ or } 49.02\%.$$

5. Using the data in the table, compute the actual simple annual return between December 2011 and December 2012. If you invested \$10,000 in Starbucks at the end of December 2011, how much would the investment be worth at the end of December 2012? Compare with your result in part (3).

   The annual return is just the percentage change in price:

   $$R_t(12) = R_A = \frac{P_t - P_{t-12}}{P_{t-12}} = \frac{\$53.1 - \$44.89}{\$44.89} = 0.1829 \text{ or } 18.29\%.$$

This is a lot smaller than the annual return computed in part 1. The future value of \$10,000 *is then*

$$FV = \$10,000 \times (1 + R_A) = \$10,000 \times (1 + 0.1829) = \$11,829.$$

6. Using the data in the table, compute the actual annual continuously compounded return between December 2011 and December 2012. Compare with your result in part (4). Convert this continuously compounded return to a simple return (you should get the same answer as in part 5).

   The annual continuously compounded return is defined as

   $$r_A = r_t(12) = \ln(1 + R_t(12)) = \ln(P_t/P_{t-12})$$

   Using $R_t(12) = R_A = 0.1829$ *gives*

   $$r_A = \ln(1 + 0.1829) = 0.1680 \text{ or } 16.8\%$$

   The implied simple return is

   $$R_A = e^{r_A} - 1 = e^{0.1680} - 1 = 0.1829.$$

Part II. R Exercises

Go to `http://finance.yahoo.com` and download monthly data on Starbucks (ticker symbol sbux) over the period March, 1998 to March, 2013. See Part III for how to get data.

Read the data into Excel and make sure to reorder the data so that time runs forward. Delete all columns except those containing the dates and the adjusted closing prices. Save the file as a .csv (comma separated value) file and call it `sbuxPrices.csv`. This is important because base R does not have functions for importing data from an Excel spreadsheet (see the RODBC and xlsReadWrite packages for functions to read and write directly to Excel files).

Start R and open the file *Econ147_HW1_Code.r*. Execute the commands in this file line by line. Copy and paste your output into a Word (or whatever word processor you use) document to show that you have done this assignment.

1. Import the data in the file `sbuxPrices.csv` using the R function `read.csv()` into the data.frame object `sbux.df`. Follow the commands in *Econ147_HW1_Code.r* to manipulate the data.

When reading data saved from Yahoo! into R, use the read.csv() function. This will bring the raw data into a data.frame object. A data.frame object is a rectangular data object whose columns can be of different modes (types) (e.g., one column can be character, one column can be numeric, etc). Here, we read the SBUX data saved in the file sbuxPrices.csv into a data frame using read.csv()

```
> setwd("G:\\Dropbox\\Lecture Notes\\Financial Econometrics\\HW1_Return")
> sbux.df = read.csv(file="sbuxPrices.csv",
+                      header=TRUE, stringsAsFactors=FALSE)
```

The setwd() command sets the working directory (the directory where R reads from and writes to) to the location where the file sbuxPrices.csv lives. In the read.csv() function the optional arguments header=TRUE specifies that the column names are in the first row of the files, and the optional argument stringsAsFactors=FALSE forces character data to be read in a characters instead of being converted to a factor variable. Use the head() function to display the beginning part of the data.frame object

```
> head(sbux.df)
        Date Adj.Close
1 1998-03-31      5.37
2 1998-04-01      5.70
3 1998-05-01      5.69
4 1998-06-01      6.33
5 1998-07-01      4.96
6 1998-08-03      3.74
```

The function tail()for the last observations will give you

```
 Date Adj.Close
176 2012-10-01     45.26
177 2012-11-01     51.36
178 2012-12-03     53.10
179 2013-01-02     55.56
180 2013-02-01     54.51
181 2013-03-01     56.60
```

The data.frame object sbux.df has two columns: Date and Adj.Close. The dates are character data and the adjusted close prices are numeric data.

When working with time series data in a `data.frame` object we can make the dates the row names of the `data.frame` and keep all of the columns in the `data.frame` numeric data. Here is one way to do this

```
> sbuxPrices.df = sbux.df[, "Adj.Close", drop=FALSE]
> rownames(sbuxPrices.df) = sbux.df$Date
> head(sbuxPrices.df)
           Adj.Close
1998-03-31      5.37
1998-04-01      5.70
1998-05-01      5.69
1998-06-01      6.33
1998-07-01      4.96
1998-08-03      3.74
```

Notice the use of drop=FALSE in the first command. This forces the result to be a one column `data.frame` object instead of a numeric vector (this overrides the default behavior of R).

To see the subperiod movement in starbux stock price, for example, during the crisis period (inception to termination, roughly), simply use the subsetting function as

```
> sbux.df[101:132,]
          Date Adj.Close
101 2006-07-03     32.46
102 2006-08-01     29.40
103 2006-09-01     32.29
104 2006-10-02     35.79
105 2006-11-01     33.46
106 2006-12-01     33.58
107 2007-01-03     33.13
108 2007-02-01     29.30
109 2007-03-01     29.74
110 2007-04-02     29.41
111 2007-05-01     27.32
112 2007-06-01     24.88
113 2007-07-02     25.30
114 2007-08-01     26.12
115 2007-09-04     24.84
```

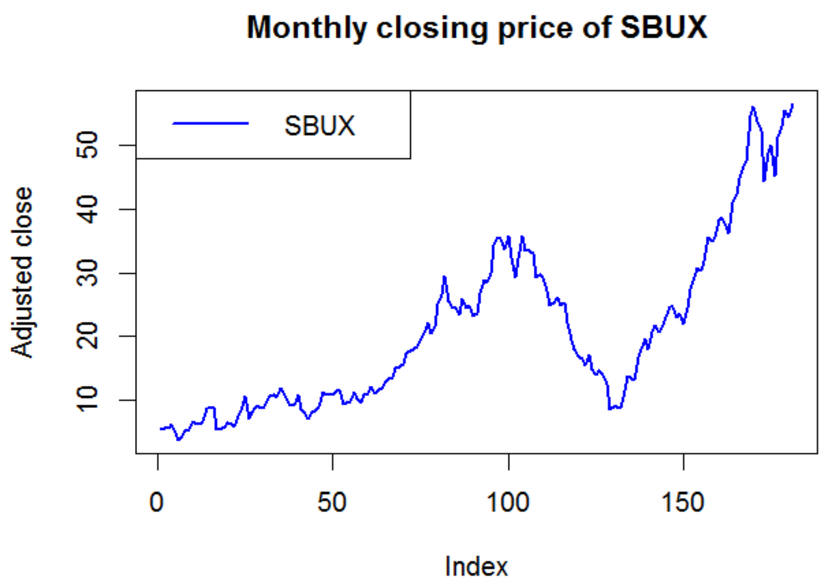
```
116 2007-10-01     25.30
117 2007-11-01     22.18
118 2007-12-03     19.41
119 2008-01-02     17.93
120 2008-02-01     17.05
121 2008-03-03     16.59
122 2008-04-01     15.39
123 2008-05-01     17.25
124 2008-06-02     14.92
125 2008-07-01     13.93
126 2008-08-01     14.75
127 2008-09-02     14.10
128 2008-10-01     12.45
129 2008-11-03      8.47
130 2008-12-01      8.97
131 2009-01-02      8.95
132 2009-02-02      8.68
```

2. Plot the closing price data using the `plot()` function.

Here we make a simple time plot of prices using the generic `plot()` function

```
> plot(sbux.df$Adj.Close, type="l", col="blue",
+      lwd=2, ylab="Adjusted close",
+      main="Monthly closing price of SBUX")
# now add a legend
> legend(x="topleft", legend="SBUX",
+        lty=1, lwd=2, col="blue")
```

The resulting graph is shown in Figure below. Setting `type="l"` specifies a line graph; `col="blue"` makes the line color blue; `lwd=2` makes the line width twice as thick; `ylab=“Adjusted close”` sets the y-axis label; `main=“Monthly closing price of SBUX”` sets the main title. Notice that the dates do not show up on the x-axis in the line plot. We may learn how to fix this in the future.



3. Compute monthly simple and continuously compounded returns. Plot these returns separately and on the same graph.

To compute a vector of simple monthly returns use

```
> n = nrow(sbuxPrices.df)
> sbux.ret = (sbuxPrices.df[2:n,1] -
+             sbuxPrices.df[1:(n-1),1])/sbuxPrices.df[1:(n-1),1]
> class(sbux.ret)
[1] "numeric"
# now add dates as names to the vector.
> names(sbux.ret) = rownames(sbuxPrices.df)[2:n]
> head(sbux.ret)
1998-04-01 1998-05-01 1998-06-01 1998-07-01 1998-08-03 1998-09-01
  0.061453  -0.001754   0.112478  -0.216430  -0.245968   0.147059
```

To compute a vector of continuously compounded (cc) returns from the vector of simple returns use

```
> sbux.ccret = log(1 + sbux.ret)
```

Alternatively, to compute the cc returns directly from prices use

```
> sbux.ccret = log(sbuxPrices.df[2:n,1]) - log(sbuxPrices.df[1:(n-1),1])
```

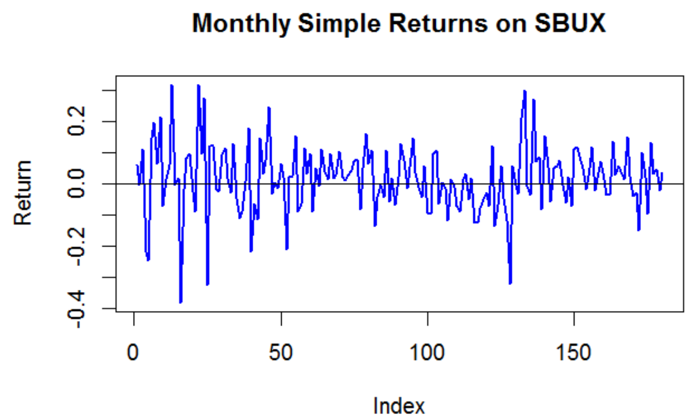
The simple and cc returns are quite similar

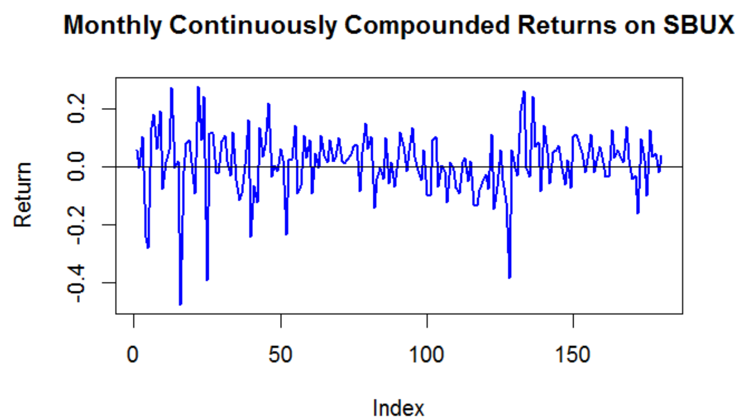
```
> head(cbind(sbux.ret, sbux.ccret))
           sbux.ret sbux.ccret
1998-04-01  0.061453   0.059638
1998-05-01 -0.001754  -0.001756
1998-06-01  0.112478   0.106590
1998-07-01 -0.216430  -0.243894
1998-08-03 -0.245968  -0.282320
1998-09-01  0.147059   0.137201
```

Notice that the cc returns are slightly smaller than the simple returns. To plot the simple and cc returns on separate graphs use

```
# split screen into 2 rows and 1 column
> par(mfrow=c(2,1))
# plot simple returns first
> plot(sbux.ret, type="l", col="blue", lwd=2, ylab="Return",
+      main="Monthly Simple Returns on SBUX")
> abline(h=0)
# next plot the cc returns
> plot(sbux.ccret, type="l", col="blue", lwd=2, ylab="Return",
+      main="Monthly Continuously Compounded Returns on SBUX")
> abline(h=0)
# reset the screen to 1 row and 1 column
> par(mfrow=c(1,1))
```

The resulting graph is shown in Figure below. Notice that the simple and cc returns look almost identical.





To plot the simple and cc returns on the same graph use

```
> par(mfrow=c(1,1))
> # plot the returns on the same graph
> plot(sbux.ret, type="l", col="blue", lwd=2, ylab="Return",
+      main="Monthly Returns on SBUX")
> # add horizontal line at zero
> abline(h=0)
> # add the cc returns
> lines(sbux.ccret, col="red", lwd=2)
> # add a legend
> legend(x="bottomright", legend=c("Simple", "CC"),
```

```
+        lty=1, lwd=2, col=c("blue","red"))
```

The resulting graph is shown in Figure below. The simple returns are in blue and the cc returns are in red. Notice how the cc returns are slightly smaller than the simple returns.