Data input, Return Calculation, and basic R graphics

jsn-jin @ UCLA

Contents

Part 1: Download Financial Data from Yahoo Finance

To get data from Yahoo Finance, you can follow the following step:

1. Go to http://finance.yahoo.com/
2. In the Quote Lookup box type the symbol of the stock for which you want data. For example, the symbol for Microsoft is MSFT. If you do not know the ticker symbol, you can type the full name of the company in the Quote Lookup box. Then you will see its ticker symbol from the list below the box.
3. The recent quote for your stock as well as a wealth of other information will be presented on a new page. To get historical prices, click the "Historical Data" link on the upper middle panel of the page.
4. A new page will open and you will be able to specify the date ranges for the data to be downloaded and the frequency of the data (daily, weekly or monthly).
5. After setting the date and frequency information, click the Apply button.
6. At the bottom of the "Apply" button click on "Download Data". This will bring up a Save As dialogue box. Specify a name for the file and save to disk. Each file will contain 6 columns. For monthly data, the column labeled Open gives the opening price at the beginning of the month, the column labeled High gives the highest price during the month, the column labeled Low gives the lowest price during the month, the column labeled Close gives the closing price during the month and the column labeled Volume gives the total monthly volume. The open and the close data have been adjusted for dividends and stock splits. Use the (adjusted) close data for your analysis.

Exercise: Download the histrical weekly price data of Apple Inc. (AAPL) and save the corresponding data set to your laptop. The time period is 01/01/2015-12/31/2018, the frequency of the price is weekly. We will use this data set for today's discussion.

Part 2: Basic Data Analysis

###Load the Data Set

```
df <- read.csv(file = "aapl.csv", head = TRUE, sep=",", stringsAsFactors = FALSE)
head(df)
```

```
##         Date   Open   High    Low  Close Adj.Close    Volume
## 1 2014-12-29 113.64 113.92 107.35 109.33 100.45430 124489500
## 2 2015-01-05 108.29 113.25 104.63 112.01 102.91672 283252500
## 3 2015-01-12 112.60 112.80 105.20 105.99  97.38545 304226600
## 4 2015-01-19 107.84 113.75 106.50 112.98 103.80799 198737000
## 5 2015-01-26 113.74 120.00 109.03 117.16 107.64864 465842700
```

```
## 6 2015-02-02 118.05 120.51 116.08 118.93 109.27495 270757300
```

```
# Better Choice: tidyverse
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyvers
```

```
## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  3.0.0     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------------- tidyverse_con
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
df2 <- read_csv(file = "aapl.csv")
```

```
## Parsed with column specification:
## cols(
##   Date = col_date(format = ""),
##   Open = col_double(),
##   High = col_double(),
##   Low = col_double(),
##   Close = col_double(),
##   `Adj Close` = col_double(),
##   Volume = col_double()
## )
```

###Data Manipulation

```
options(digits = 4)
Date <- df$Date
Open <- df$Open
AdjClose <- df$Adj.Close
Low <- df$Low
High <- df$High
Volume <- df$Volume

class(df)
```

```
## [1] "data.frame"
```

```
head(df)
```

```
##         Date  Open  High   Low Close Adj.Close    Volume
## 1 2014-12-29 113.6 113.9 107.3 109.3    100.45 124489500
## 2 2015-01-05 108.3 113.2 104.6 112.0    102.92 283252500
## 3 2015-01-12 112.6 112.8 105.2 106.0     97.39 304226600
## 4 2015-01-19 107.8 113.8 106.5 113.0    103.81 198737000
## 5 2015-01-26 113.7 120.0 109.0 117.2    107.65 465842700
## 6 2015-02-02 118.1 120.5 116.1 118.9    109.27 270757300
```

```
tail(df)
```

```
##           Date  Open  High   Low Close Adj.Close    Volume
## 204 2018-11-19 190.0 190.7 172.1 172.3     169.7 164498700
## 205 2018-11-26 174.2 182.8 170.3 178.6     175.9 213749900
## 206 2018-12-03 184.5 184.9 168.3 168.5     166.0 167526800
```

```
## 207 2018-12-10 165.0 172.6 163.3 165.5     163.0 217537700
## 208 2018-12-17 165.4 168.4 149.6 150.7     148.5 287694300
## 209 2018-12-24 148.1 158.5 146.6 156.2     153.9 191160200
```

```
# class(Date)
# class(Open)
# class(AdjClose)
# class(Low)
# class(High)
# class(Volume)
```

Next, we will focus on the variable "Close". Let's first calculate some summary statistics.

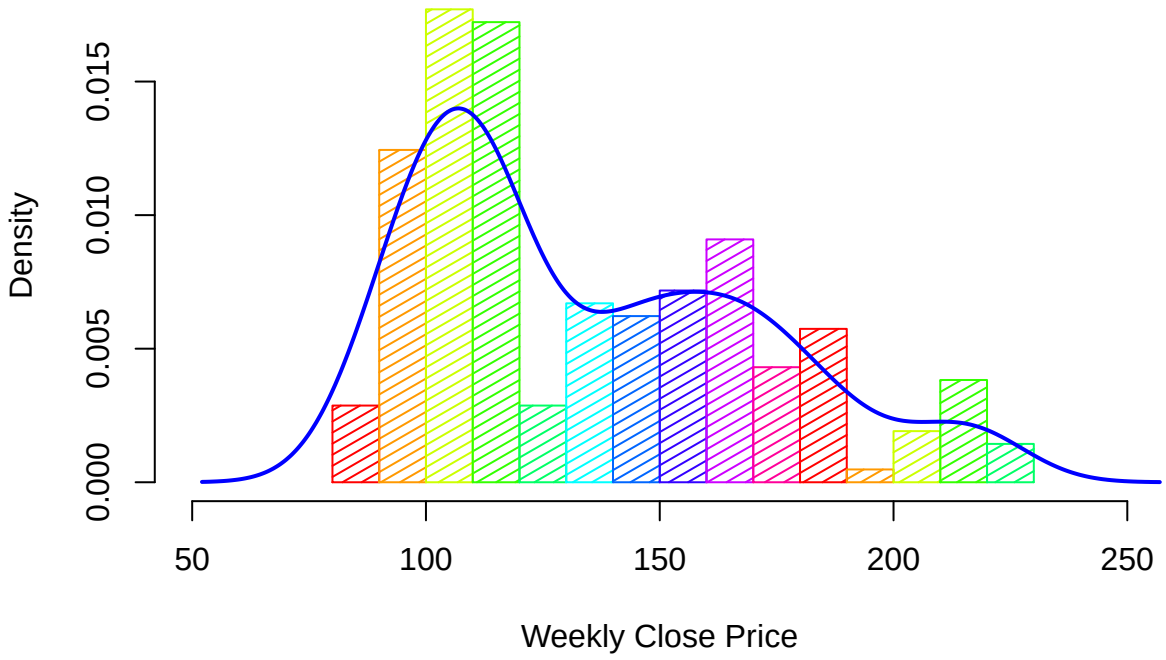
```
summary(AdjClose)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    85.6   105.1   119.5   134.7   163.5   223.5
```

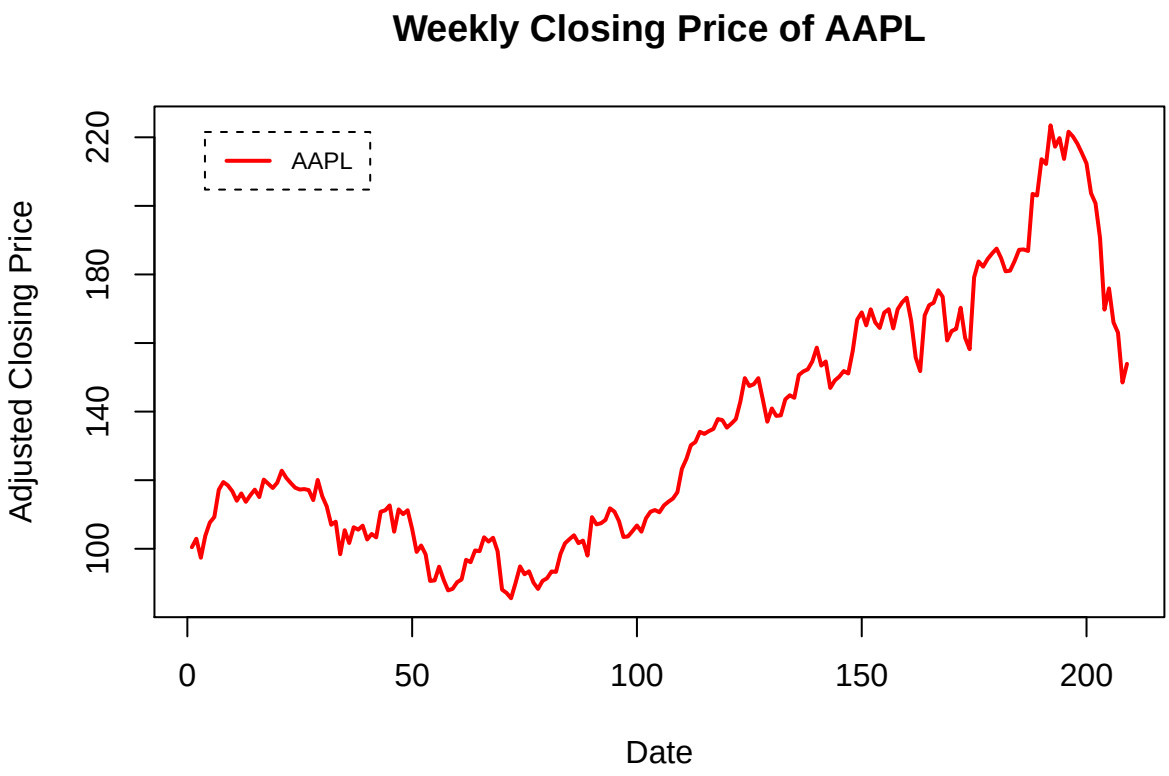
```
length(AdjClose) # Number of rows.
```

```
## [1] 209
```

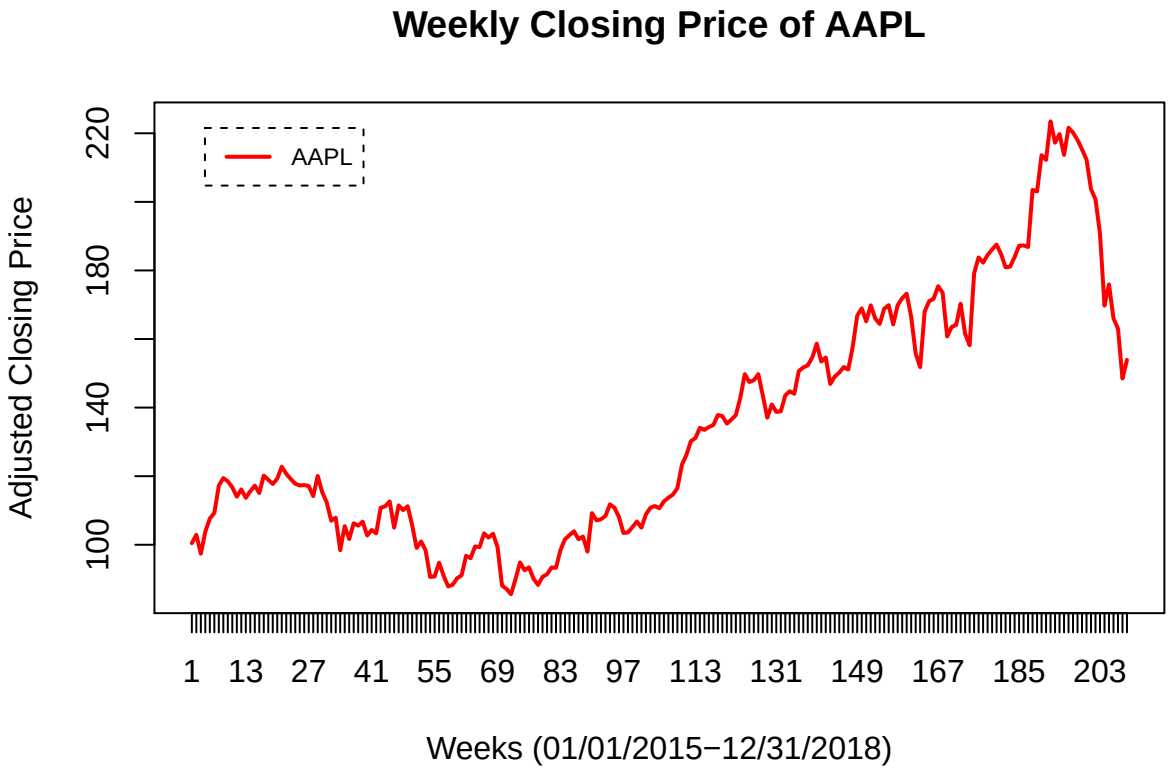
```
hist(AdjClose,
     prob = TRUE,
     xlab = "Weekly Close Price",
     xlim = c(50, 250),
     main = "Histogram of AAPL's Weekly Close Price",
     breaks = 10,
     density = 20,
     angle = 30,
     col = rainbow(10))

lines(density(AdjClose), lwd = 2, col = "blue")
```




```
plot(AdjClose,
     type = "l",
     col = "red",
     lwd = 2,
     xlab = "Date",
     ylab = "Adjusted Closing Price",
     main = "Weekly Closing Price of AAPL")

# now add a legend
legend(x = "topleft",
       legend = "AAPL ",
       lty = 1,
       lwd = 2,
       box.lty = 2,
       col = "red",
       cex = 0.75,
       inset = .05)
```




```
# add week in the x-axis
n <- length(AdjClose)
date_i <- seq(from = 1, to = n, by = 1)
plot(AdjClose,
     type = "l",
     col = "red",
     lwd = 2,
     ylab = "Adjusted Closing Price",
     xlab = "Weeks (01/01/2015-12/31/2018)",
     xaxt = 'n',
     main = "Weekly Closing Price of AAPL")

axis(1, at = date_i)

legend(x = "topleft",
       legend = "AAPL",
       lty = 1,
       lwd = 2,
       box.lty = 2,
       col = "red",
       cex = 0.75,
       inset = 0.05)
```



Part 3: Calculating Returns

###Simple Weekly Returns

```
# simple 1-week returns
n <- nrow(df)
aapl.ret <- (df[2:n,6] - df[1:(n-1),6]) / df[1:(n-1),6]

# example for vectorization
a <- c(1,3,5,7,9)
b <- c(9,7,5,3,1)
a - b
```

```
## [1] -8 -4  0  4  8
```

```
# notice that aapl.ret is not a data.frame object
class(aapl.ret)
```

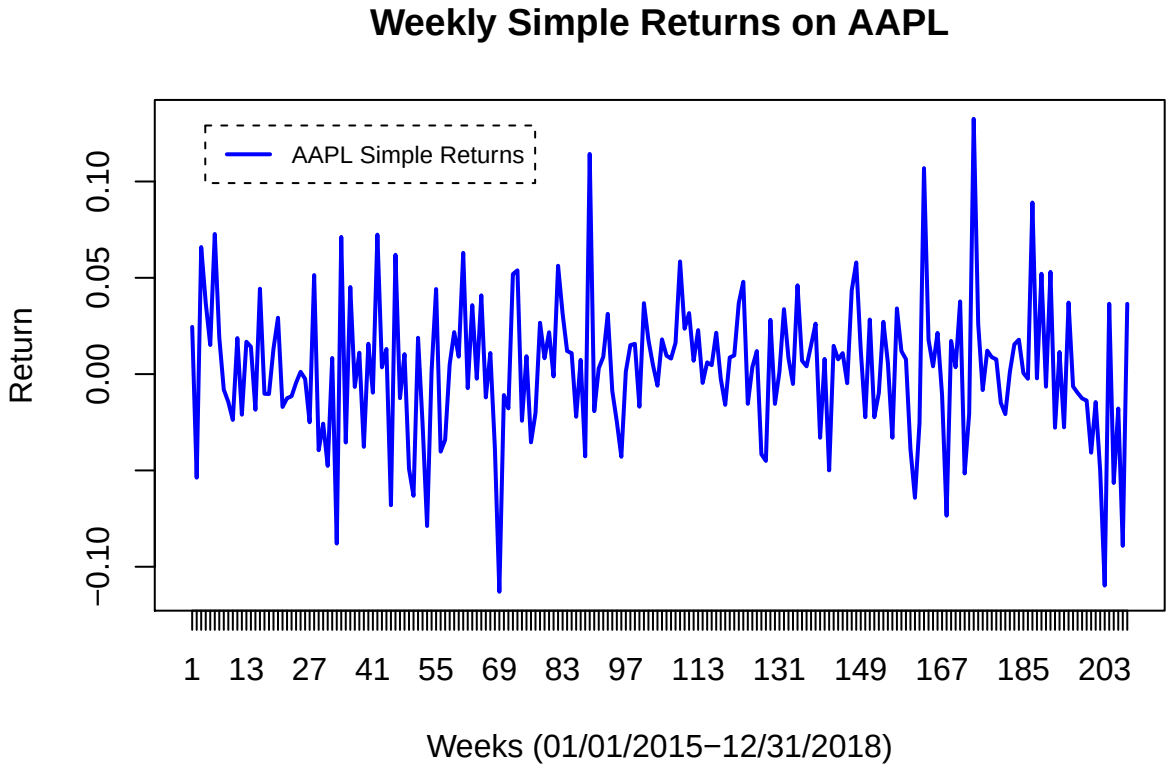
```
## [1] "numeric"
```

```
# now add dates as names to the vector.
names(aapl.ret) <- rownames(df)[2:n]
head(aapl.ret)
```

```
##         2         3         4         5         6         7
##  0.02451 -0.05375  0.06595  0.03700  0.01511  0.07274
```

```
# Note: to ensure that aapl.ret is a data.frame use drop = FALSE when computing returns
aapl.ret.df <- (df[2:n,6, drop = FALSE] - df[1:(n-1), 6, drop = FALSE]) / df[1:(n-1), 6,
                                                                              drop = FALSE]
```

```r
plot(aapl.ret,
     type = "l",
     col = "blue",
     lwd = 2,
     xlab = "Weeks (01/01/2015-12/31/2018)",
     ylab = "Return",
     xaxt = 'n',
     main = "Weekly Simple Returns on AAPL")

date_i = seq(from = 1, to = n-1, by = 1)
axis(1, at=date_i)

legend(x = "topleft",
       legend = "AAPL Simple Returns",
       lty = 1,
       lwd = 2,
       box.lty = 2,
       col = "blue",
       cex = 0.75,
       inset = 0.05)
```



Exercise:

Please repeat the same exercise as above for compound returns and make a plot. You can put both plots into a single graph and compare their difference.

###Compoud Weekly Returns

```r
aapl.ccret = log(1 + aapl.ret)
head(aapl.ccret)
```

```
##        2        3        4        5        6        7
##  0.02422 -0.05524  0.06387  0.03633  0.01499  0.07022
```

```
# alternatively
aapl.ccret = log(df[2:n,6]) - log(df[1:(n-1),6])
names(aapl.ccret) = rownames(df)[2:n]
head(aapl.ccret)
```

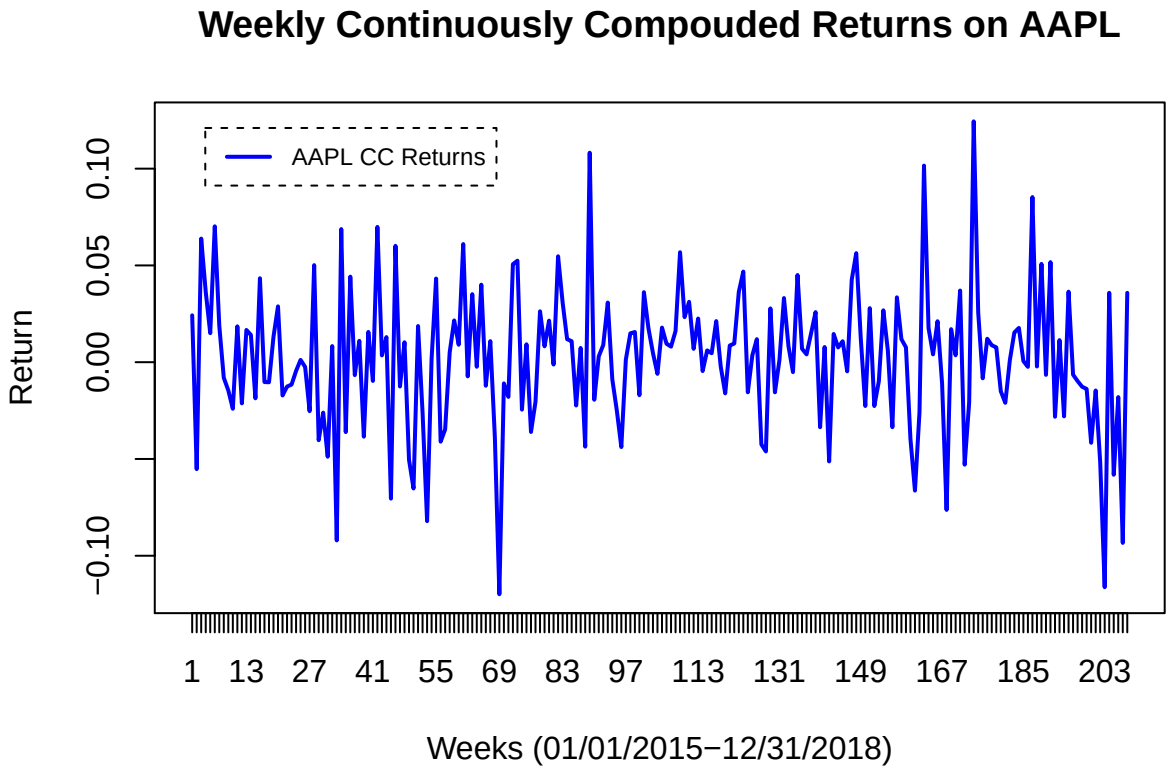
```
##        2        3        4        5        6        7
##  0.02422 -0.05524  0.06387  0.03633  0.01499  0.07022
```

```
head(aapl.ret)
```

```
##        2        3        4        5        6        7
##  0.02451 -0.05375  0.06595  0.03700  0.01511  0.07274
```

```
aapl.ccret.df = log(df[2:n,6]) - log(df[1:(n-1),6])
```

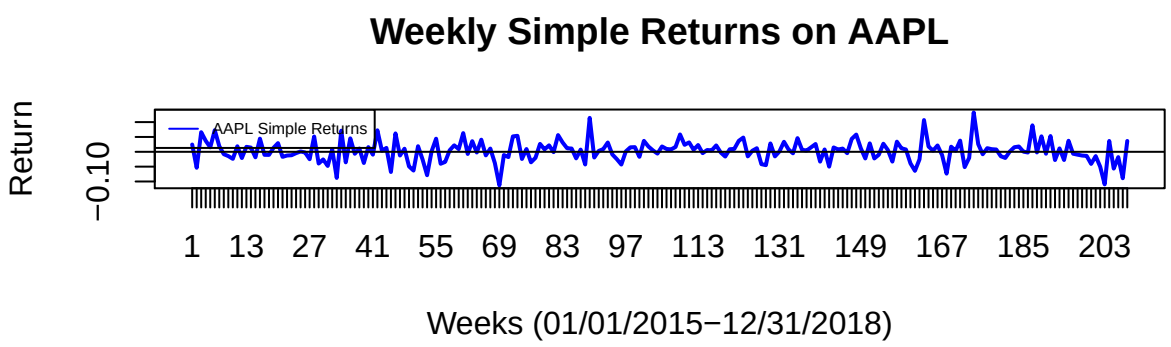
```
plot(aapl.ccret,
     type = "l",
     col = "blue",
     lwd = 2,
     xlab = "Weeks (01/01/2015-12/31/2018)",
     ylab = "Return",
     xaxt = 'n',
     main = "Weekly Continuously Compouded Returns on AAPL")

date_i = seq(from = 1, to = n-1, by = 1)
axis(1, at = date_i)
legend(x = "topleft",
       legend = "AAPL CC Returns",
       lty = 1,
       lwd = 2,
       box.lty = 2,
       cex = 0.75,
       col = "blue",
       inset = 0.05)
```

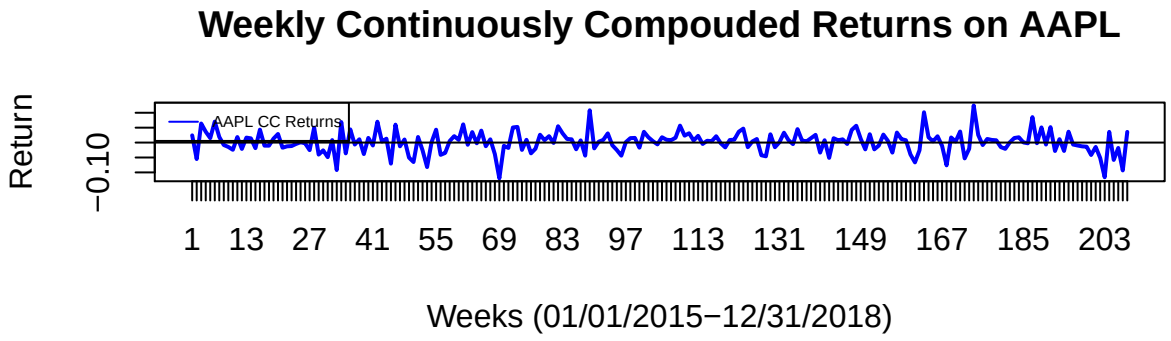


###Put Two Graphs Together

```
par(mfrow = c(2,1))
# par stands for partition

plot(aapl.ret,
     type = "l",
     col = "blue",
     lwd = 2,
     xlab = "Weeks (01/01/2015-12/31/2018)",
     ylab = "Return",
     xaxt = 'n',
     main = "Weekly Simple Returns on AAPL")

date_i = seq(from = 1, to = n-1, by = 1)
axis(1, at = date_i)
legend(x = "topleft",
       legend = "AAPL Simple Returns",
       lty = 1,
       lwd = 1,
       cex = 0.5,
       col = "blue")

abline(h=0)
# next plot the cc returns

plot(aapl.ccret,
     type = "l",
     col = "blue",
     lwd = 2,
```

```
     xlab = "Weeks (01/01/2015-12/31/2018)",
     ylab = "Return",
     xaxt = 'n',
     main = "Weekly Continuously Compouded Returns on AAPL")

date_i = seq(from = 1, to = n-1, by = 1)
axis(1, at = date_i)
legend(x = "topleft",
       legend = "AAPL CC Returns",
       lty = 1,
       lwd = 1,
       cex = 0.5,
       col = "blue")

abline(h=0)
```






```
# reset the screen to 1 row and 1 column
# par(mfrow = c(1,1))
```

```
# plot the returns on the same graph
plot(aapl.ret,
     type = "l",
     col = "brown",
     lwd = 1,
     ylab = "Return",
     xaxt = 'n',
     main = "Weekly Returns on AAPL")

# add horizontal line at zero
abline(h=0)
```

```r
# add the cc returns
lines(aapl.ccret,
      col = "green",
      lwd = 1,
      xaxt = 'n')
axis(1, at = date_i)

help(legend)

# add a legend
legend(x = "topleft",
       legend = c("Simple Return", "CC Return"),
       lty = 1,
       lwd = 2,
       col = c("brown","green"),
       box.lwd = 0.5,
       cex = 0.5,
       inset = 0.05,)
```