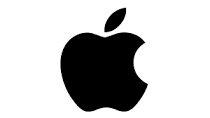
**Exploratory Data Analysis of Stock Prices**





Econ\_stocks.R

2022-10-21

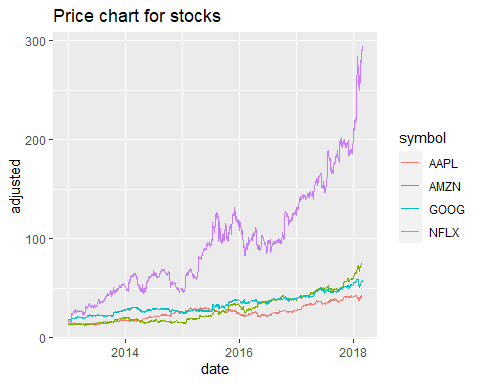
library(tidyquant)  
library(timetk)  
library(ggplot2)  
library(dbplyr)  
library(tidyverse)  
  
tickers <- c("FB", "AMZN", "AAPL", "NFLX", "GOOG")   
  
# import the stock-price data  
  
stocks <- tq\_get(tickers,  
 from = "2000-01-01",  
 to = "2018-03-01",  
 get = "stock.prices")

stocks<-c("GE", "LOW","CI") %>%

tq\_get(get = "stock.prices", from = "2000-01-01")%>%

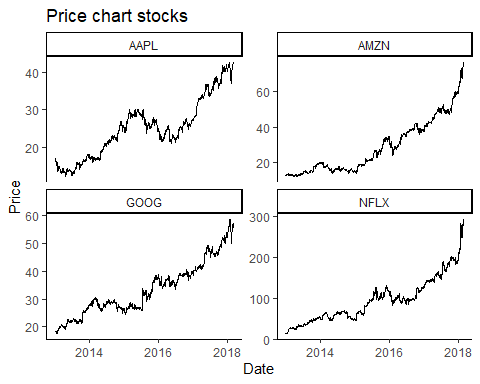
select(symbol, date, adjusted)

stocks %>%  
 ggplot(aes(x = date, y = adjusted, color = symbol)) +  
 geom\_line() +  
 ggtitle("Price chart for stocks")



stocks %>%  
 ggplot(aes(x = date, y = adjusted)) +  
 geom\_line() +  
 facet\_wrap(~symbol, scales = "free\_y") + # facet\_wrap is used to make diff frames  
 theme\_classic() + # using a new theme  
 labs(x = "Date", y = "Price") +  
 ggtitle("Price chart stocks")

These companies are measured on various scales due to their vast price discrepancies (AMZN is over $1950 while FB is under $165). We can get around this issue by charting stocks on separate y axes. Netflix is seen to have the highest adjusted price.



Two-line charts are created to visualize the adjusted stock prices over time. The first chart shows all stocks on a shared y-axis, while the second chart uses separate y-axes for each stock due to significant price discrepancies between them. It is observed that Netflix (NFLX) has the highest adjusted price among the selected companies.

**Calculating Returns**

The script calculates both daily and monthly returns for the stocks using the tidyquant package. Daily returns are computed for each stock and visualized with line charts, while monthly returns are visualized with bar charts.

Additionally, cumulative returns are calculated since 2013 for all stocks and displayed in a line chart. These cumulative returns analysis helps in understanding the overall performance of the selected stocks.

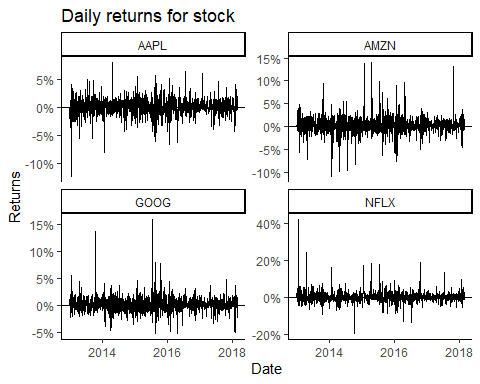
#Calculate the daily returns for the various stocks  
  
stock\_daily\_returns <- stocks %>%  
 group\_by(symbol) %>% # We are grouping the stocks by the stock symbol  
 tq\_transmute(select = adjusted,  
 mutate\_fun = periodReturn,  
 period = 'daily',  
 col\_rename = 'returns')  
  
#Calculating the monthly returns for multiple stocks

Adding up the profits from various stock investments. In this case, we only need to send an extra parameter. The returns for individual stocks can be computed by using the option group by(symbol).

stock\_monthly\_returns <- stocks %>%  
 group\_by(symbol) %>% # We are grouping the stocks by symbol  
 tq\_transmute(select = adjusted,  
 mutate\_fun = periodReturn,  
 period = 'monthly',  
 col\_rename = 'returns')

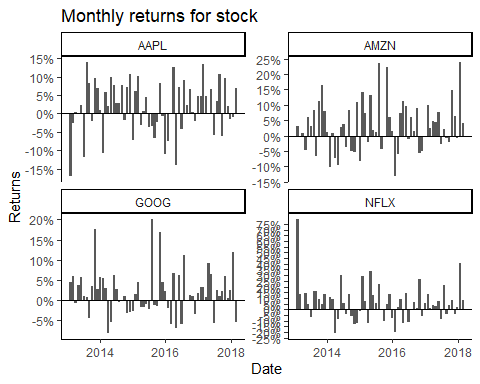
Now that we know how much money we'll make, we can put it on a graph.  
stock\_daily\_returns %>%  
 ggplot(aes(x = date, y = returns)) +  
 geom\_line() +  
 geom\_hline(yintercept = 0) +  
 facet\_wrap(~symbol, scales = "free\_y") +  
 scale\_y\_continuous(labels = scales::percent) +  
 ggtitle("Daily returns for stock") +  
 labs(x = "Date", y = "Returns") +  
 scale\_color\_brewer(palette = "Set2",  
 name = "",  
 guide = FALSE) +  
 theme\_classic()

## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please use  
## `guide = "none"` instead.

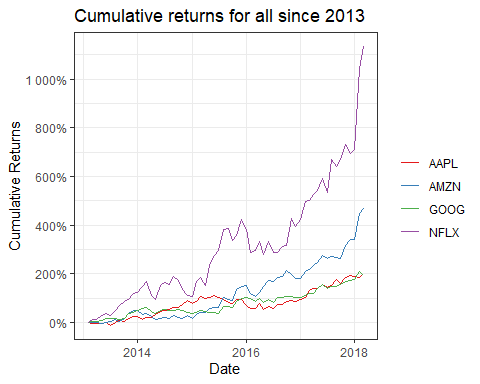


stock\_monthly\_returns %>%  
 ggplot(aes(x = date, y = returns)) +  
 geom\_bar(stat = "identity") +  
 geom\_hline(yintercept = 0) +  
 facet\_wrap(~symbol, scales = "free\_y") +  
 scale\_y\_continuous(labels = scales::percent,  
 breaks = seq(-0.5,0.75,0.05)) +  
 ggtitle("Monthly returns for stock") +  
 labs(x = "Date", y = "Returns") +  
 scale\_fill\_brewer(palette = "Set1", # We will give them different colors instead of black  
 name = "",  
 guide = FALSE) +  
 theme\_classic()

## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please use  
## `guide = "none"` instead.



stock\_monthly\_returns %>%  
 mutate(returns = if\_else(date == "2013-01-31", 0, returns)) %>%  
 group\_by(symbol) %>% # Need to group multiple stocks  
 mutate(cr = cumprod(1 + returns)) %>%  
 mutate(cumulative\_returns = cr - 1) %>%  
 ggplot(aes(x = date, y = cumulative\_returns, color = symbol)) +  
 geom\_line() +  
 labs(x = "Date", y = "Cumulative Returns") +  
 ggtitle("Cumulative returns for all since 2013") +  
 scale\_y\_continuous(breaks = seq(0,20,2),  
 labels = scales::percent) +  
 scale\_color\_brewer(palette = "Set1",  
 name = "") +  
 theme\_bw()

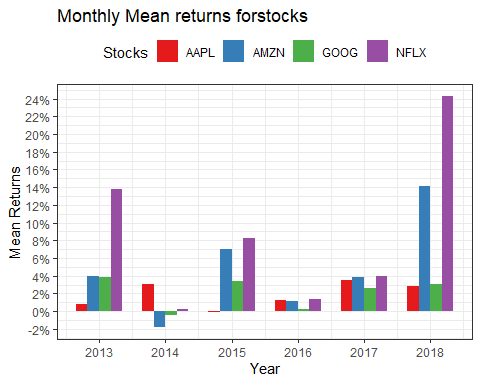


stock\_daily\_returns %>%  
 group\_by(symbol) %>%  
 summarise(mean = mean(returns),  
 sd = sd(returns))

## # A tibble: 4 × 3  
## symbol mean sd  
## <chr> <dbl> <dbl>  
## 1 AAPL 0.000826 0.0151  
## 2 AMZN 0.00153 0.0183  
## 3 GOOG 0.000960 0.0141  
## 4 NFLX 0.00282 0.0300

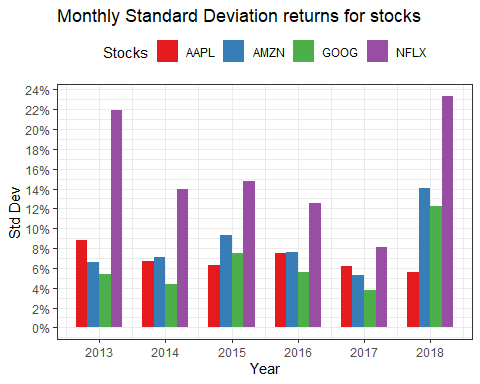
stock\_monthly\_returns %>%  
 mutate(year = year(date)) %>%  
 group\_by(symbol, year) %>%  
 summarise(mean = mean(returns),  
 sd = sd(returns)) %>%  
 ggplot(aes(x = year, y = mean, fill = symbol)) +  
 geom\_bar(stat = "identity", position = "dodge", width = 0.7) +  
 scale\_y\_continuous(breaks = seq(-0.1,0.4,0.02),  
 labels = scales::percent) +  
 scale\_x\_continuous(breaks = seq(2009,2018,1)) +  
 labs(x = "Year", y = "Mean Returns") +  
 theme\_bw() +  
 theme(legend.position = "top") +  
 scale\_fill\_brewer(palette = "Set1",  
 name = "Stocks") +  
 ggtitle("Monthly Mean returns forstocks")

## `summarise()` has grouped output by 'symbol'. You can override using the `.groups`  
## argument.



stock\_monthly\_returns %>%  
 mutate(year = year(date)) %>%  
 group\_by(symbol, year) %>%  
 summarise(mean = mean(returns),  
 sd = sd(returns)) %>%  
 ggplot(aes(x = year, y = sd, fill = symbol)) +  
 geom\_bar(stat = "identity", position = "dodge", width = 0.7) +  
 scale\_y\_continuous(breaks = seq(-0.1,0.4,0.02),  
 labels = scales::percent) +  
 scale\_x\_continuous(breaks = seq(2009,2018,1)) +  
 labs(x = "Year", y = "Std Dev") +  
 theme\_bw() +  
 theme(legend.position = "top") +  
 scale\_fill\_brewer(palette = "Set1",  
 name = "Stocks") +  
 ggtitle("Monthly Standard Deviation returns for stocks")

## `summarise()` has grouped output by 'symbol'. You can override using the `.groups`  
## argument. t.test(stocks$open)



**Hypothesis Testing**

The script conducts one-sample t-tests to test whether the means of various stock price attributes (open, high, low, close, adjusted) are significantly different from a hypothesized value of 50. The results indicate that the p-values for all tests are significantly less than 0.05, leading to the rejection of the null hypothesis. This suggests that the stock prices are significantly different from the hypothesized mean of 50.

t.test(stocks$open, mu = 50)

**One Sample t-test**

t = -6.7448, df = 5195, p-value = 1.699e-11

alternative hypothesis: true mean is not equal to 50

95 percent confidence interval:

45.11451 47.31494

sample estimates:

mean of x

46.21473

t.test(stocks$high, mu = 50)

t = -5.7275, df = 5195, p-value = 1.076e-08

alternative hypothesis: true mean is not equal to 50

95 percent confidence interval:

45.62210 47.85482

sample estimates:

mean of x

46.73846

t.test(stocks$low, mu = 50)

t = -7.8337, df = 5195, p-value = 5.705e-15

alternative hypothesis: true mean is not equal to 50

95 percent confidence interval:

44.58354 46.75190

sample estimates:

mean of x

45.66772

t.test(stocks$close, mu = 50)

t = -6.7123, df = 5195, p-value = 2.12e-11

alternative hypothesis: true mean is not equal to 50

95 percent confidence interval:

45.12743 47.33027

sample estimates:

mean of x

46.22885

t.test(stocks$adjusted, mu = 50)

t = -7.6546, df = 5195, p-value = 2.297e-14

alternative hypothesis: true mean is not equal to 50

95 percent confidence interval:

44.56174 46.77937

sample estimates:

mean of x

45.67056

**p-value**

The p-value for t-tests above is less than 0.05 which is less than the 5% significance level therefore we reject the null hypothesis.

**T-value**

The significance of a difference is determined by its t-value in relation to the standard deviation of the data in the sample. The likelihood of rejecting the null hypothesis increases as the t-value rises. In our case the t-value ranges between-5.7 and -7.8 which shows that there is significant difference between them.

**95% confidence interval**

Our test has a 95% confidence interval of 45 – 93. Therefore, for hypothesised averages between 45 and 93, the null hypothesis will be rejected at the 5% significance level.

#Calculating the Annova test

anov <- aov(symbol~ volume, data = stock)

summary(anov)

# determining the Co-variance  
stock\_monthly\_returns %>%  
 spread(symbol, value = returns) %>%  
 tk\_xts(silent = TRUE) %>%  
 cov()

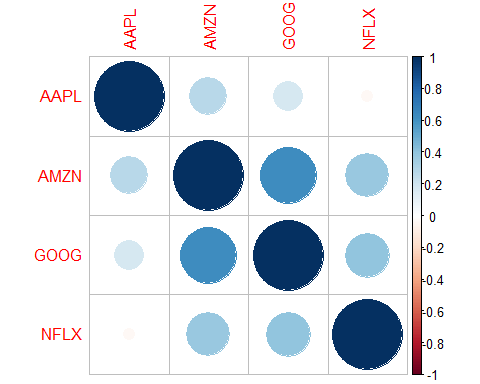
## AAPL AMZN GOOG NFLX  
## AAPL 0.0048655804 0.001548201 0.0007038239 -0.0003317367  
## AMZN 0.0015482008 0.006399439 0.0028486648 0.0047548937  
## GOOG 0.0007038239 0.002848665 0.0032318983 0.0035171155  
## NFLX -0.0003317367 0.004754894 0.0035171155 0.0246420247

# determining the correlation  
  
stock\_monthly\_returns %>%  
 spread(symbol, value = returns) %>%  
 tk\_xts(silent = TRUE) %>%  
 cor()

## AAPL AMZN GOOG NFLX  
## AAPL 1.00000000 0.2774527 0.1774875 -0.03029619  
## AMZN 0.27745273 1.0000000 0.6263853 0.37864451  
## GOOG 0.17748753 0.6263853 1.0000000 0.39411194  
## NFLX -0.03029619 0.3786445 0.3941119 1.00000000

From the correlation results above most of the stocks markets are positively correlated, except for Netflix and AAPL which are negatively correlated.

library(corrplot)  
stock\_monthly\_returns %>%  
 spread(symbol, value = returns) %>%  
 tk\_xts(silent = TRUE) %>%  
 cor() %>%  
 corrplot()



**Covariance and Correlation**

Covariance and correlation matrices are computed to examine the relationships between stock returns. The correlation matrix is visualized using the corrplot package. From the above correlation matrix it’s clear that google and Amazon are more correlated compared to the rest.

It is noted that most stocks are positively correlated, except for Netflix and AAPL, which exhibit a negative correlation.

**Conclusion**

From the complex descriptive statistics, we have carried out, we have observed that the is significance difference between the stocks prices of the various stocks of different entities. In conclusion, the exploratory data analysis reveals significant differences in stock prices and returns among the selected companies. The analysis provides valuable insights into the performance and relationships between these stocks. Further analysis and modeling can be performed to gain more in-depth insights into these financial datasets.