Dow Jones Index Case Study

Saket Mishra [jjw317] | Siva Manoj Gannamani [tht046] | Diego Aldo Pettorossi [zho125] | Jonnattan De Leon [gcd097] | Shreya Budhathoki [atg696]

Executive Summary

This case study project is dependent on the prediction of the stock prices for the Dow jones Index case study. The predictive modeling techniques that are being used for the case study are the three models of linear regression, support vector regression and decision trees. The findings from this case study shows that support vector regression model proved to be the best model to forecast the stock returns prediction. As such, the linear regression model yielded better results for the entire index however, decision tree model yielded better results for individual stocks.

The better model was concluded based on the MSE result values where SVR model yielded the better results for the entire index as well as for individual stocks with the MSE value of 8.05. As per the CAPM results, to determine the risks of investment in different stocks, the beta value from the model helped to determine that the riskiest stock to invest in would be in in Boeing, Caterpillar Inc., and Disney with beta values of 1.62, 1.49 and 1.48 respectively. As such, the case study also showed the results that the safest stock investment would be in the stocks Craft, Procter & Gamble and Merck & Co. respectively with the beta values of 0.18, 0.48 and 0.54 respectively.

Table of Contents

EXECUTIVE SUMMARY **1**

THE PROBLEM **2**

LITERATURE REVIEW **3**

METHODOLOGY **4**

LINEAR REGRESSION 4
SUPPORT VECTOR REGRESSION 5
DECISION TREES 6

DATA **6**

EXPLORING THE DATASET 8

LAG PLOTS **12**

FINDINGS **18**

CAPITAL ASSET PRICING MODEL (CAPM) **20**

CONCLUSION AND RECOMMENDATION **22**

APPENDIX **23**

REFERENCES 23
PLOTS 24
CODE FOR THE MODEL 28

The Problem

Stock market prediction is an upmost analytical field where financial advisors have been clashing for a while. When people look to invest, their first choice is investment in the stock market. However, to keep afloat the critical point in delving into investment is understanding the market and the companies to invest in. This case study is focused on the Dow Jones Index for stock returns. The Dow Jones Indexes includes the different sectors of the industry of the US economy that is heavily price-weighted like companies from Microsoft, Disney, Intel, Boeing, and Pfizer.

The main objective of this case study is to build different models to predict the price of the stocks, while evaluating the risks of those stocks. The prediction on the stock price is determined by correlating it to the highest rate of return of the stocks the following week. The models to be used are both parametric and non-parametric methodologies, which includes models such as linear regression, Support Vector Machines (SVM) and Decision Trees. The data for the case study includes 30 stocks that we are to fit in the various models individually to determine which model would outperform the other with the technique of RMSE (Root Mean Squares Error). Next, the technique of CAPM (Capital Asset Pricing Model) is to be used to calculate the risks for the different Dow Jones Index's stocks.

Literature Review

The Dow Jones Index case study is a project that many have worked with for the forecasting of the prices. As such, this research paper used techniques such as probabilistic price interval strategy to forecast the stock price. The other technique used was direct weight optimization method that makes use of linear estimators and convex optimization. The Dow Jones Index was used as a case study for the paper in forecasting the stock prices for up to 5 days and validating the results with persistence and neural network-based predictors as baseline approaches. Also, the validation of the predicted price intervals was done with quantile regression. The paper indicates that the techniques are valuable that could be implemented on the existing techniques for the forecasting of the stock price (Alfonso, Carnerero, Ramirez and T. Alamo, 2021).

This second research paper for Dow Jones Index case study is a project that is focused on forecasting of the prices. As such, this research paper used the prediction model LSTM-AdaBoost (Long Short-Term Memory and Adaptive Boosting) on the index's characteristics. The Dow Jones Index was used as a case study using the AdaBoost algorithm with LSTM classifiers that were weak. The model was iterative to update the weights for each of the version to yield a robust classifier. The results from the case study shows that the model obtained results of 43.76% increase in the R_square stating that the model gave significantly higher results than those of the traditional classification model (Ying, Shou, and Liu, 2021).

Methodology

The dataset that is being used for this project is divided into training and testing dataset based on the quarter variable. The training dataset is taken from quarter 1 (i.e., data from January through March) and the testing dataset is taken from quarter 2 (i.e., data from April through June). The target variable for this case study objective is the percent_change_next_weeks_price as this variable yields the result for percent of change in the price of the stock returns the following week.

The first process involved in the case study was to clean the data and complete the data preprocessing step. The provided dataset was balanced, so moving to the preprocessing step, the data was cleaned by removing the dollar prefix ('$') from the variable price. A few of the variables were also converted to factors and the variable date was also changed to a proper format. Next step involved checking for the missing values in the dataset where we found a few missing observations. This was dealt with by replacing the missing values with the mean of the variable.

The next step with the case study involved building the models for the prediction. The models used for the study were linear regression, Support Vector Regression (SVR) and Decision Trees.

Linear Regression

Linear Regression model is defined as a linear approach that assumes two input variables to have a linear relationship with a single output variable. This model is generically used for predictive analysis and uses the equation:

$$\boldsymbol{Y = a + bX}$$

where,

- ◊ *Y is the dependent variable*
- ◊ *X is the independent variable*
- ◊ *A is the y-intercept and b is the slope of the line.*

The linear regression model works by finding the line of best fit between the data points to predict the output values for those observations that are not present in the data points. This approach is used to predict values of a variable based on the value of another variable. The advantages and disadvantages to linear regression model are:

Advantages:

- ◊ Simple implementation with coefficients of output that are easy to interpret.
- ◊ Susceptible to over-fitting which, can be avoided using dimension reduction and cross-validation techniques.
- ◊ Best approach to be used when there is a linear relationship between the dependent and independent variable in comparison to other algorithms complexity.

Disadvantages:

- ◊ Outlier data points hugely impacts the effects on regression with linear boundaries i.e., sensitive model to outliers
- ◊ Susceptible to underfitting when the stated hypothesis cannot fit with the data providing low accuracy.
- ◊ Assumption that the data is independent.

Support Vector Regression

Support Vector Regression is a SVM model that works with the flexibility that it provides in defining the acceptable error in the model. The main objective behind using the support vector regression model is to minimize the coefficients and tolerate errors through acceptable error margin.

The SVR (Support Vector Regression) model is generally used in prediction of discrete values following the same principle as the SVM (Support Vector Machine) model. The advantages and disadvantages to support vector regression model are:

Advantages:

- ◊ The model is robust to outlier data points.
- ◊ The model has better capability of generalization and higher accuracy of prediction.
- ◊ The model has simple implementation, and the decision model can be updated easily.

Disadvantages:

- ◊ The model is not suitable for prediction with large dataset.

- ◊ The model would underperform if the number of features exceeds the training data samples number.
- ◊ The model underperforms when the dataset has more target classes that overlaps.

Decision Trees

Decision tree model is a non-parametric method with tree-like models of decisions that is generally used for classification and regression. The main objective of the decision tree model is to forecast the value of the target variable by learning the simple rules of decision that were inferred from the features of the data.

The decision tree model holds a tree like structure where the internal node represents a test and each branch represents the outcome of the test, while each leaf holds the class label. The advantages and disadvantages to decision tree model are:

Advantages:

- ◊ The model yields clear visualization and is simple and easy to understand.
- ◊ The model can handle both categorical and continuous variables and does not require feature scaling.
- ◊ The model handles missing values and is robust to outliers and requires less training period.

Disadvantages:

- ◊ The model has issues with overfitting and high variance.
- ◊ The model could be unstable with adding a new data point and is affected by noise.
- ◊ The model is not suitable with large datasets.

Data

The dataset used for this case study was taken from the UCI Machine Learning Repository and the link is provided below:

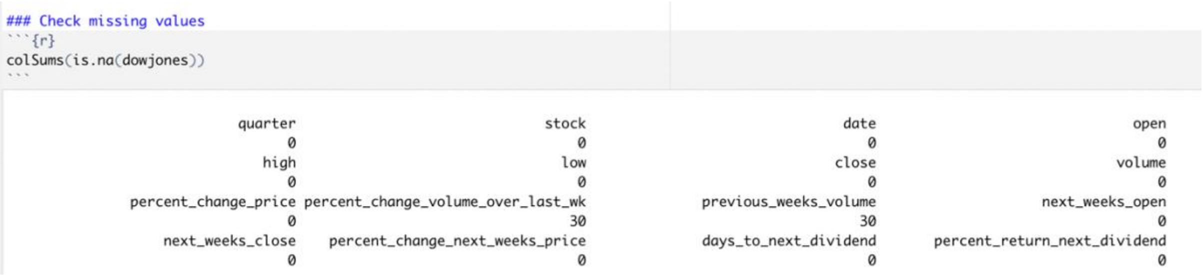
https://archive.ics.uci.edu/ml/datasets/Dow+Jones+Index

The link above holds the dataset used for this case study where the training and testing dataset were split based on the variables quarterly. The value 1 for quarterly was taken as the training dataset and the value 2 was taken as the test dataset for the case study. The characteristic of the data is that it is a time-series data with 16 variables.

The variables available in the dataset are explained below along with what they determine:

- Quarter: the yearly quarter (1 = Jan – Mar; 2 = Apr – Jun)
- Stock: the stock symbol
- Date: the last business day of the work (typically a Friday of the week)
- Open: the price of the stock at the beginning of the week.
- High: the highest price of the stock during the week
- Low: the lowest price of the stock at the end of the week
- Close: the price of the stock at the end of the week.
- Volume: the number of shares of stock that traded hands in the week
- Percent_change_price: the percentage change in price throughout the week
- Percent_change_volume_over_last_week: the percentage change in the number of shares of stock that traded hands for this week compared to the previous week.
- Previous_weeks_volume: the number of shares of stock that traded hands in the previous week
- Next_weeks_open: the opening price of the stock in the following week
- Next_weeks_close: the closing price of the stock in the following week
- Percent_change_next_weeks_price: the percentage change in price of the stock in the following week
- Days_to_next_dividend: the number of days until the next dividend
- Percent_return_next_dividend: the percentage of return on the next dividend

The dataset held data that was balanced and so for data preprocessing and cleaning we are checking the missing observations in the dataset. The missing observation was dealt with for both variables by replacing it with the mean of the variable. The preprocessing section of the data also included in formatting the date to a proper format. The next variable was price, where the prefix of the dollar sign (‘$’) was removed from the data. Some of the variables were converted to its factors to work with it easier for the case study.



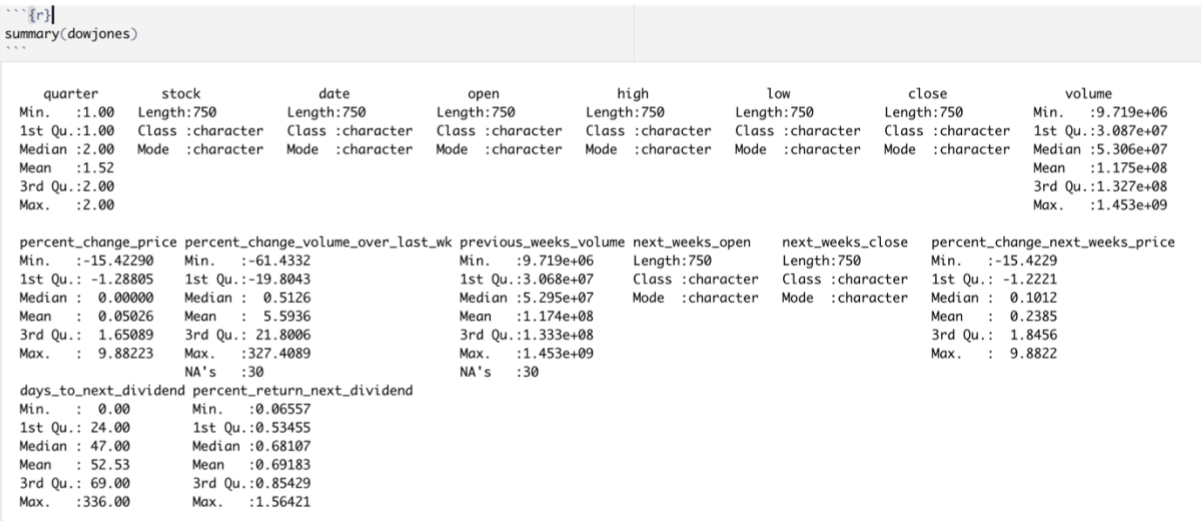
~~~
### Check missing values
```{r}
colSums(is.na(dowjones))
```

                        quarter                            stock                             date                             open 
                              0                                0                                0                                0 
                           high                              low                            close                           volume 
                              0                                0                                0                                0 
           percent_change_price percent_change_volume_over_last_wk            previous_weeks_volume                  next_weeks_open 
                              0                               30                               30                                0 
               next_weeks_close    percent_change_next_weeks_price             days_to_next_dividend     percent_return_next_dividend 
                              0                                0                                0                                0 
~~~

Fig: Checking for missing values in the dataset

The results from the exploratory data analysis to understand the data in the dataset and its values are as shown in the section below.

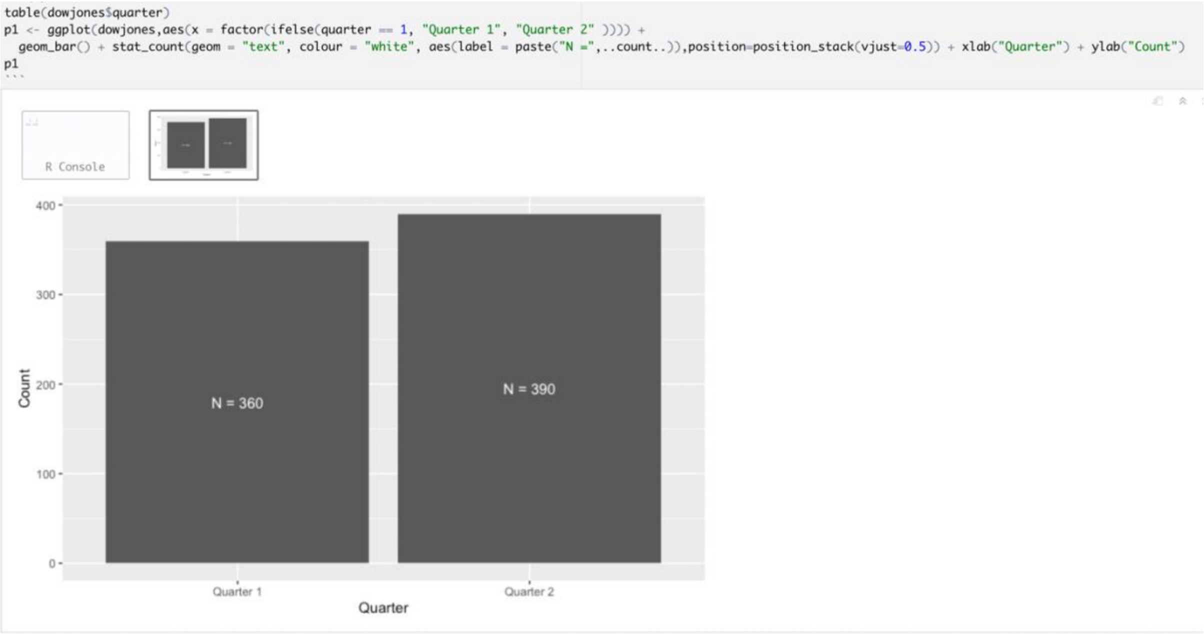
Exploring the dataset

In this section, we explored the dataset to get information about the dataset at hand and understand how the data is distributed as well as how it is related with the target variable.



~~~
```{r}
summary(dowjones)
```

    quarter          stock               date               open               high               low                close              volume         
 Min.   :1.00   Length:750         Length:750         Length:750         Length:750         Length:750         Length:750         Min.   :9.719e+06  
 1st Qu.:1.00   Class :character   Class :character   Class :character   Class :character   Class :character   Class :character   1st Qu.:3.087e+07  
 Median :2.00   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Median :5.306e+07  
 Mean   :1.52                                                                                                                     Mean   :1.175e+08  
 3rd Qu.:2.00                                                                                                                     3rd Qu.:1.327e+08  
 Max.   :2.00                                                                                                                     Max.   :1.453e+09  

 percent_change_price percent_change_volume_over_last_wk previous_weeks_volume next_weeks_open    next_weeks_close   percent_change_next_weeks_price
 Min.   :-15.42290    Min.   :-61.4332                   Min.   :9.719e+06     Length:750         Length:750         Min.   :-15.4229               
 1st Qu.: -1.28805    1st Qu.:-19.8043                   1st Qu.:3.068e+07     Class :character   Class :character   1st Qu.: -1.2221               
 Median :  0.00000    Median :  0.5126                   Median :5.295e+07     Mode  :character   Mode  :character   Median :  0.1012               
 Mean   :  0.05026    Mean   :  5.5936                   Mean   :1.174e+08                                           Mean   :  0.2385               
 3rd Qu.:  1.65089    3rd Qu.: 21.8006                   3rd Qu.:1.333e+08                                           3rd Qu.:  1.8456               
 Max.   :  9.88223    Max.   :327.4089                   Max.   :1.453e+09                                           Max.   :  9.8822               
                      NA's   :30                         NA's   :30                                                                                 
 days_to_next_dividend percent_return_next_dividend
 Min.   :  0.00         Min.   :0.06557             
 1st Qu.: 24.00         1st Qu.:0.53455             
 Median : 47.00         Median :0.68107             
 Mean   : 52.53         Mean   :0.69183             
 3rd Qu.: 69.00         3rd Qu.:0.85429             
 Max.   :336.00         Max.   :1.56421             
~~~

Fig: Summary of the dataset



```
table(dowjones$quarter)
p1 <- ggplot(dowjones,aes(x = factor(ifelse(quarter == 1, "Quarter 1", "Quarter 2" )))) +
  geom_bar() + stat_count(geom = "text", colour = "white", aes(label = paste("N =",..count..)),position=position_stack(vjust=0.5)) + xlab("Quarter") + ylab("Count")
p1
```
```

*Fig: Bar plot analysis for the quarter variable*

*Fig: Checking to see if there is any duplicate data in the dataset*

Some exploratory visual analysis to find insights with the data is shown below:

◊ From the graph below, we can see that stock 'T (AT&T)' has the highest dividend percentage.
```

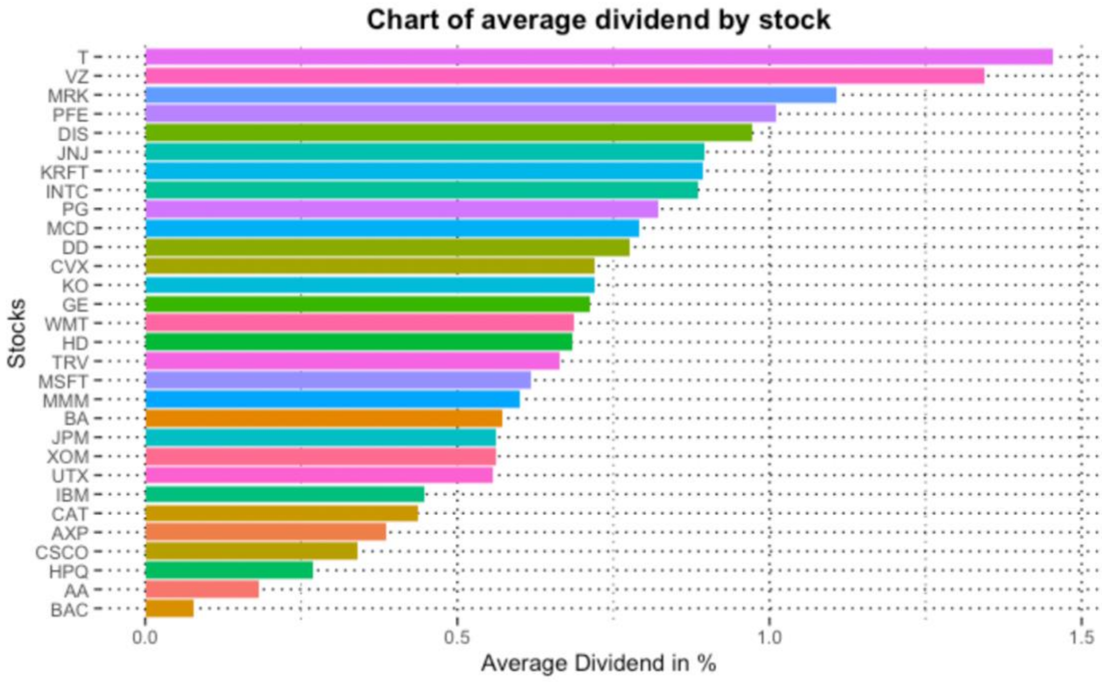


Fig: stocks with highest average dividend

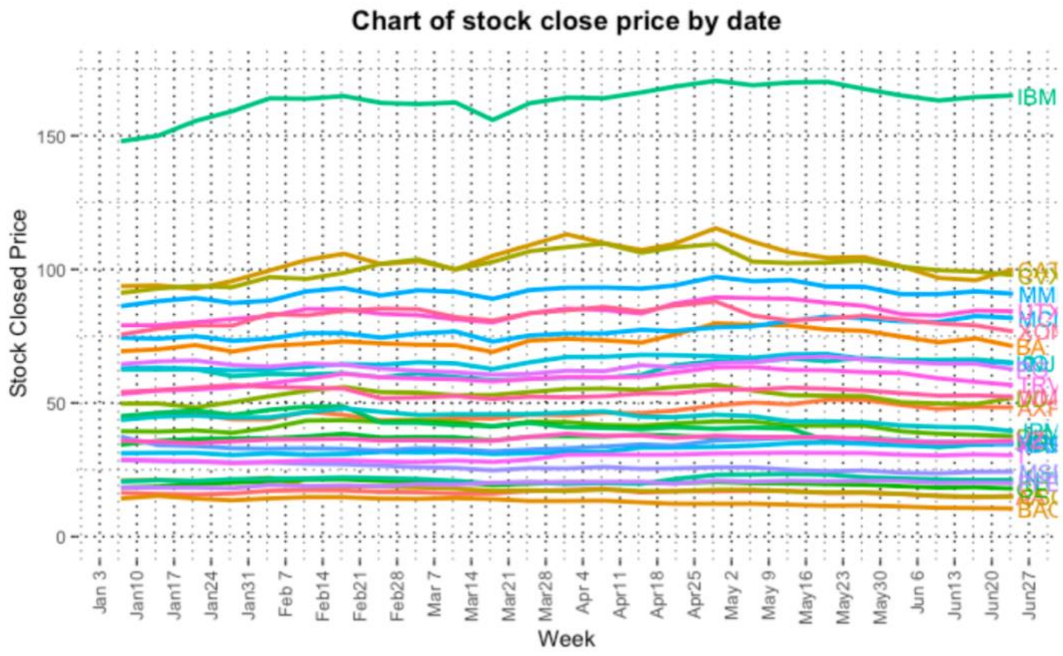
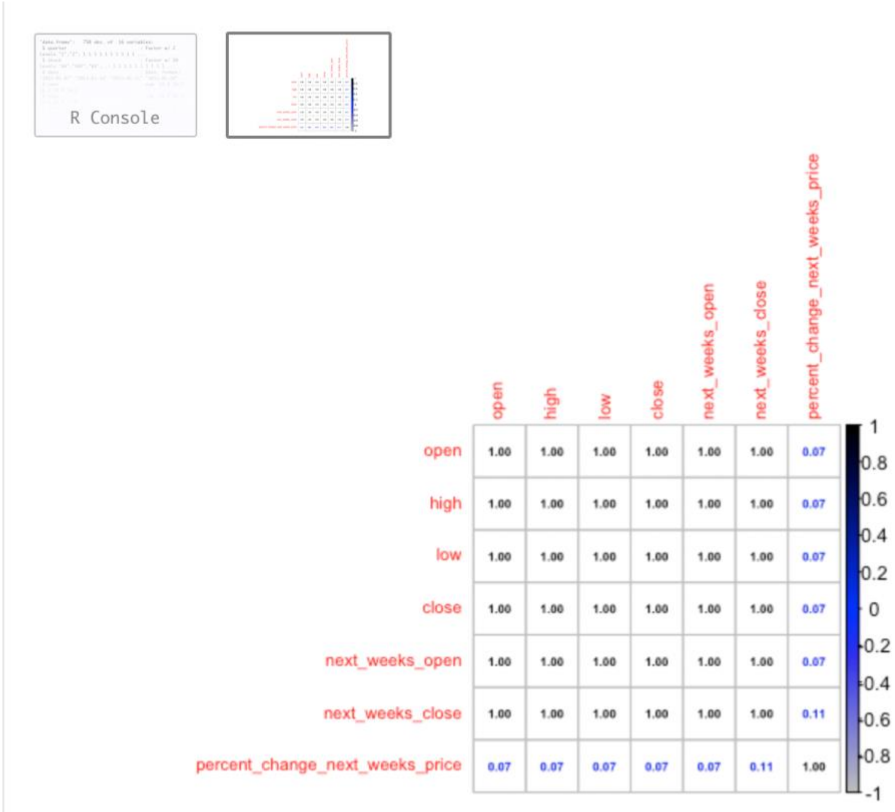
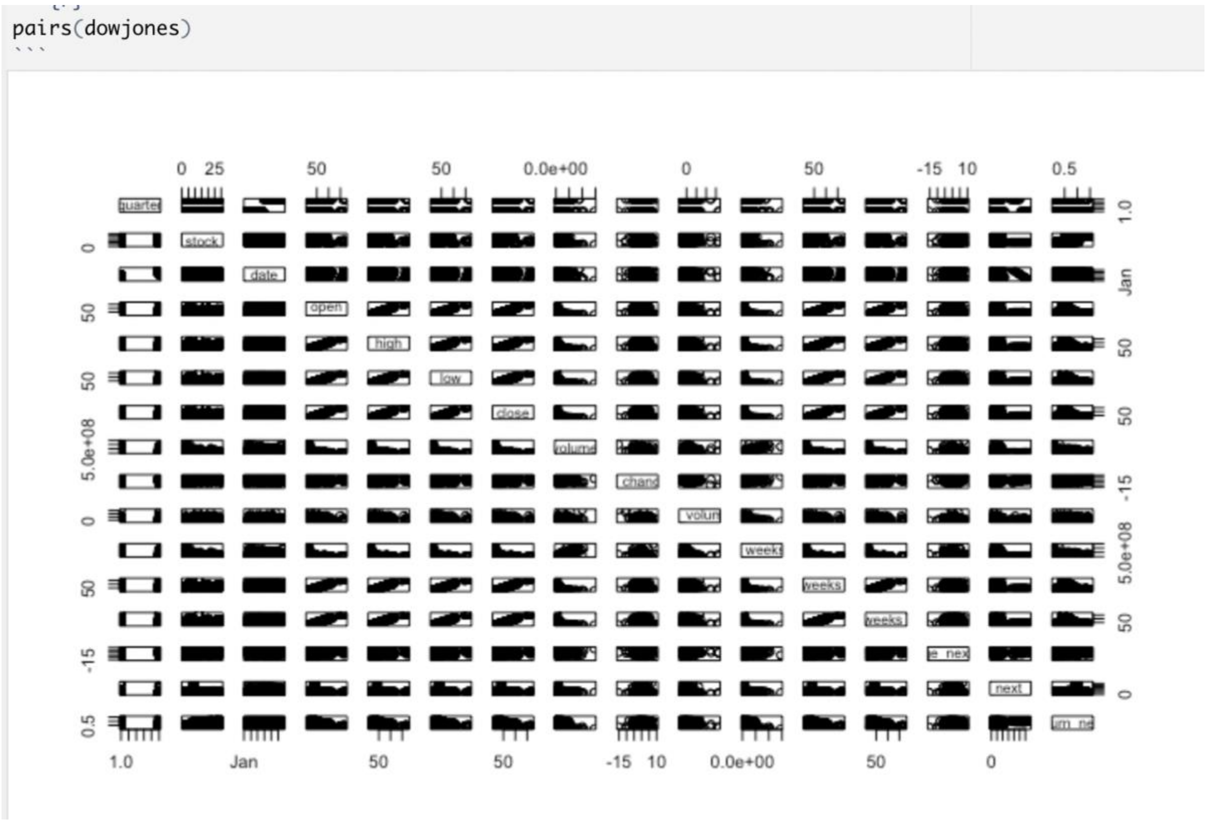


Fig: stock price over time



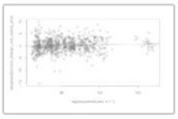
	open	high	low	close	next_weeks_open	next_weeks_close	percent_change_next_weeks_price
open	1.00	1.00	1.00	1.00	1.00	1.00	0.07
high	1.00	1.00	1.00	1.00	1.00	1.00	0.07
low	1.00	1.00	1.00	1.00	1.00	1.00	0.07
close	1.00	1.00	1.00	1.00	1.00	1.00	0.07
next_weeks_open	1.00	1.00	1.00	1.00	1.00	1.00	0.07
next_weeks_close	1.00	1.00	1.00	1.00	1.00	1.00	0.11
percent_change_next_weeks_price	0.07	0.07	0.07	0.07	0.07	0.11	1.00

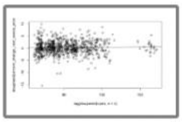
Fig: Correlational plot



```
pairs(dowjones)
```
```

*Fig: Pairwise plot*
```

Lag plots





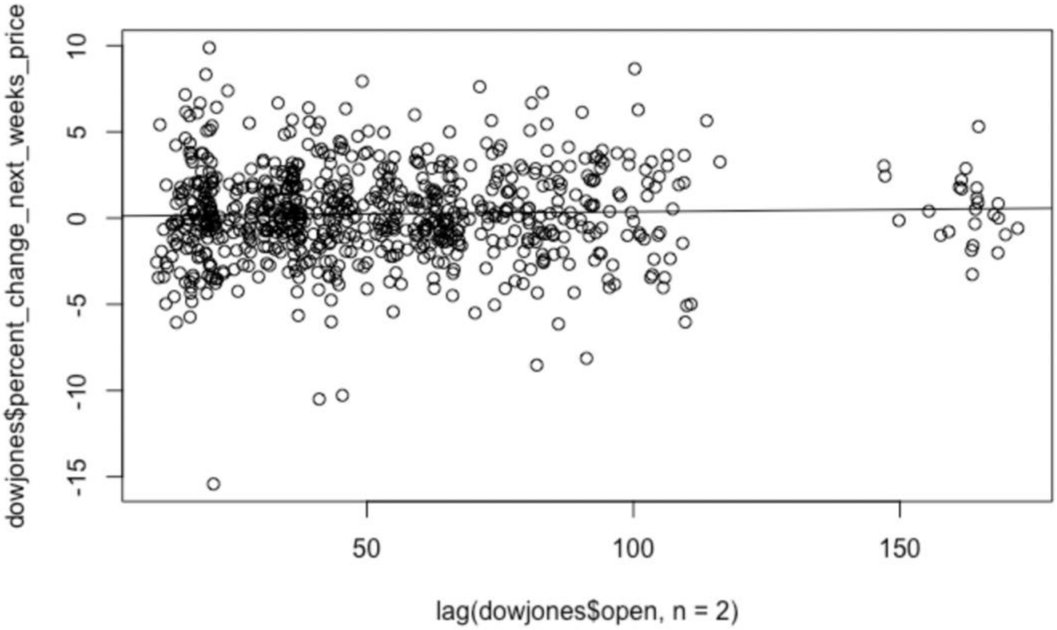
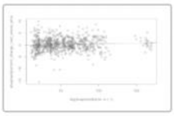
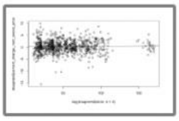


Fig: Lag plot between target variables and variable open





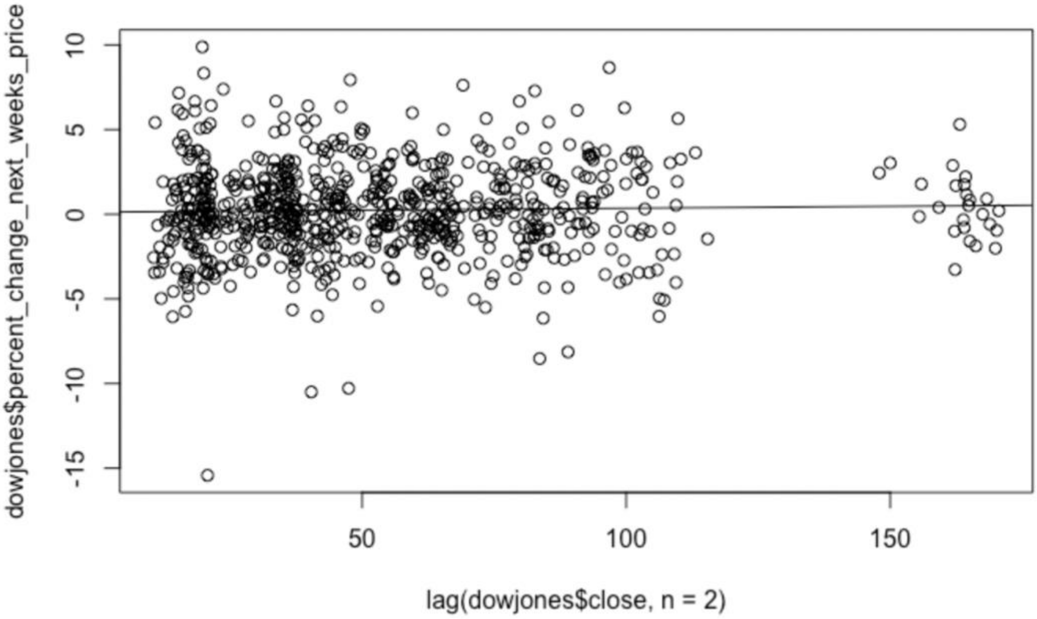
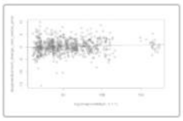
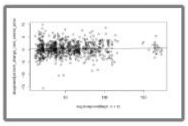


Fig: Lag plot between target variables and variable close





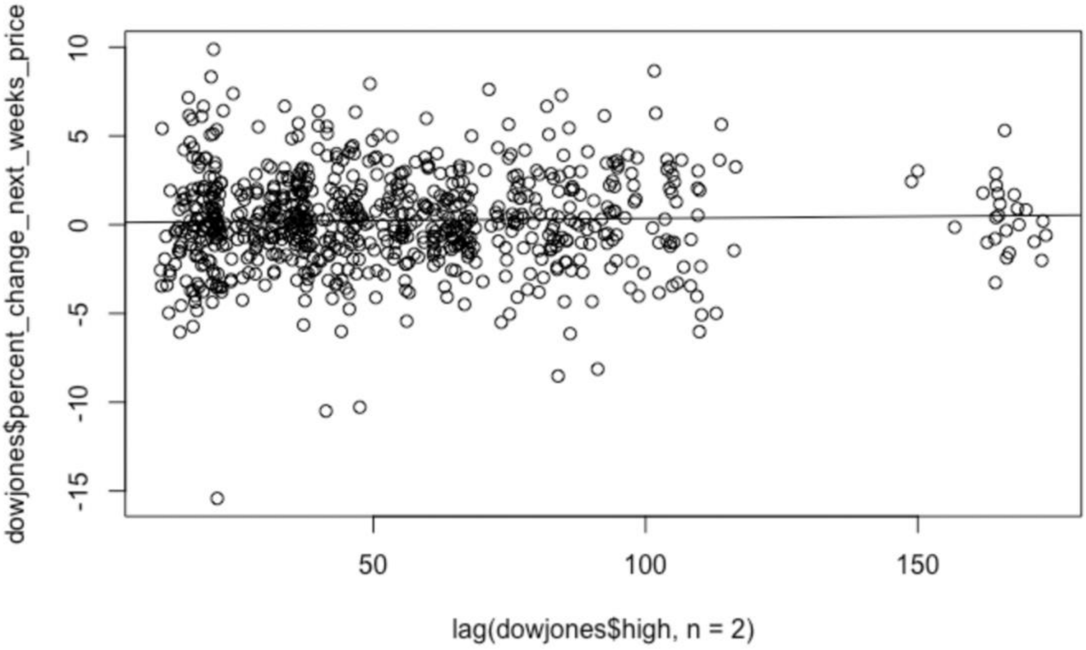
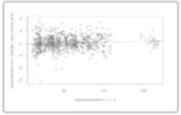
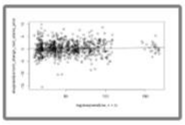


Fig: Lag plot between target variables and variable high





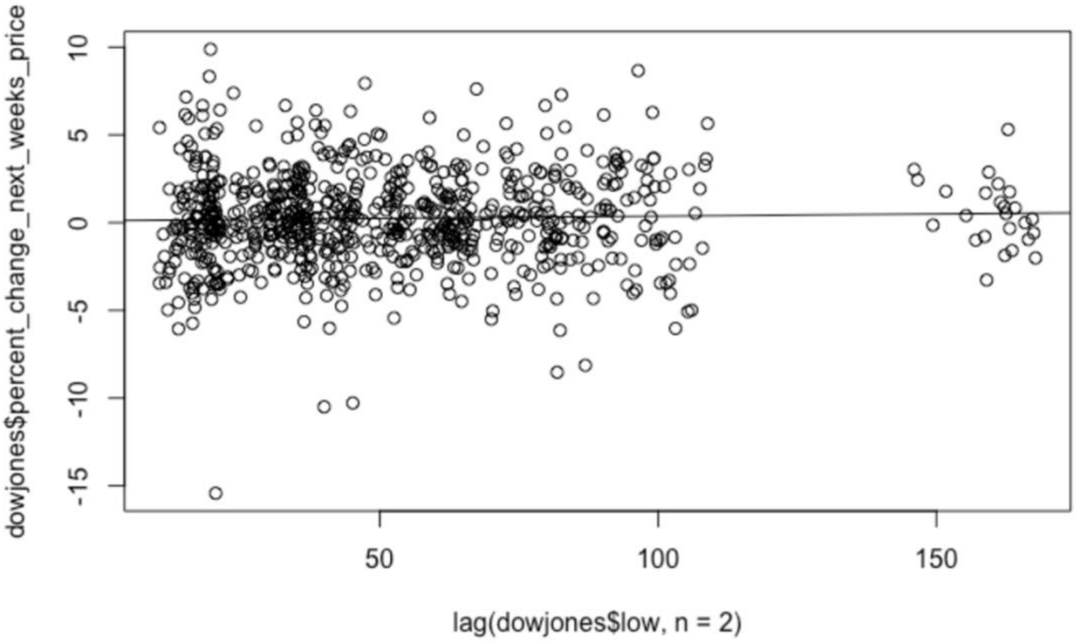
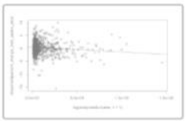
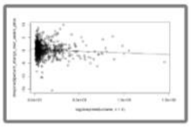


Fig: Lag plot between target variables and variable low





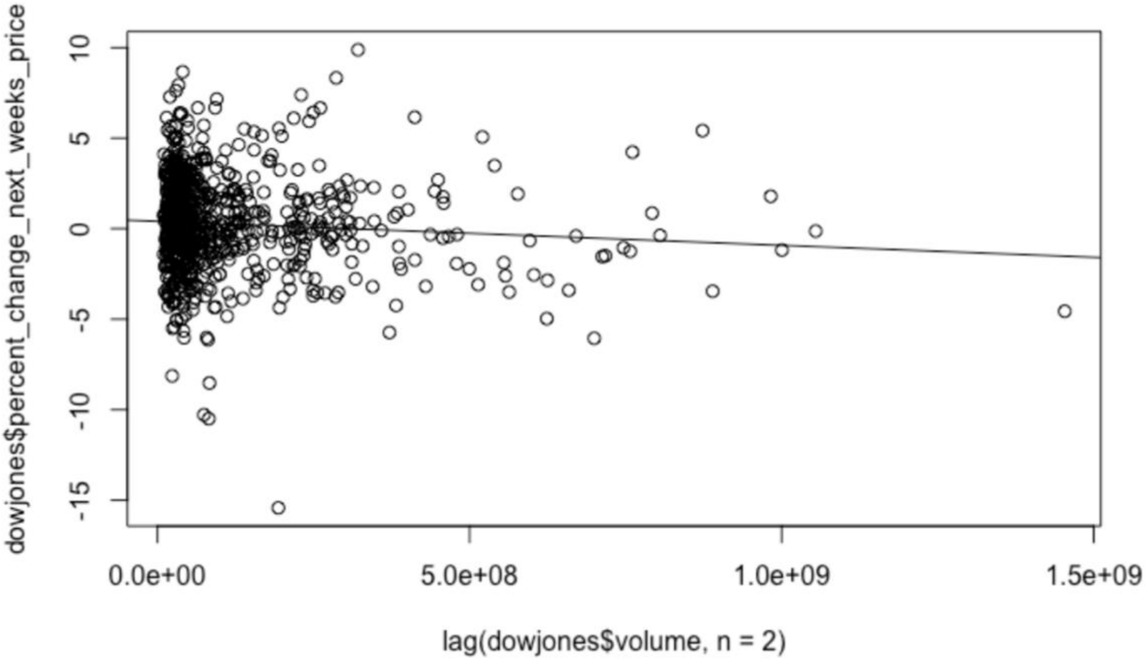
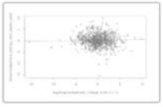
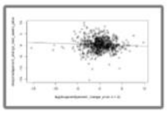


Fig: Lag plot between target variables and variable volume





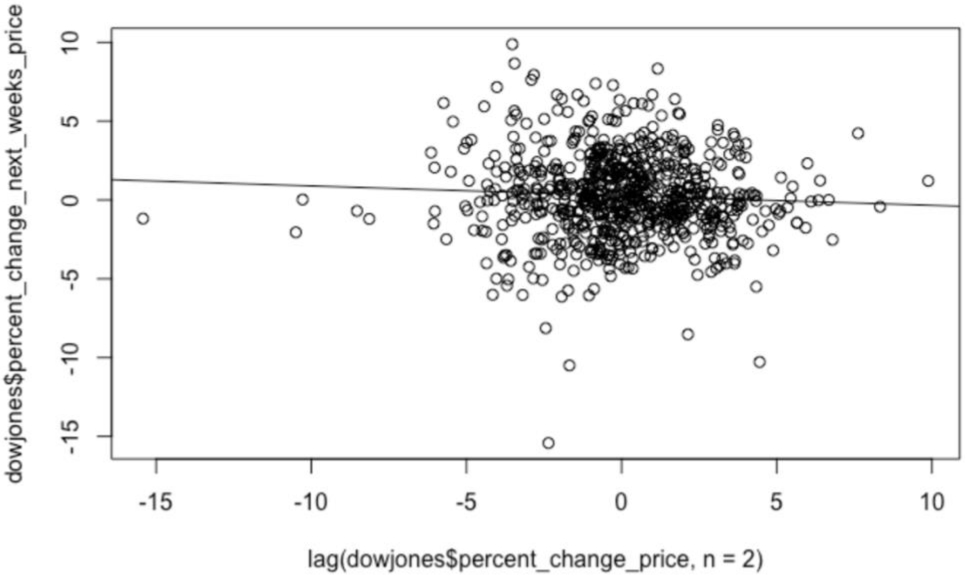


Fig: Lag plot between target variables and variable price

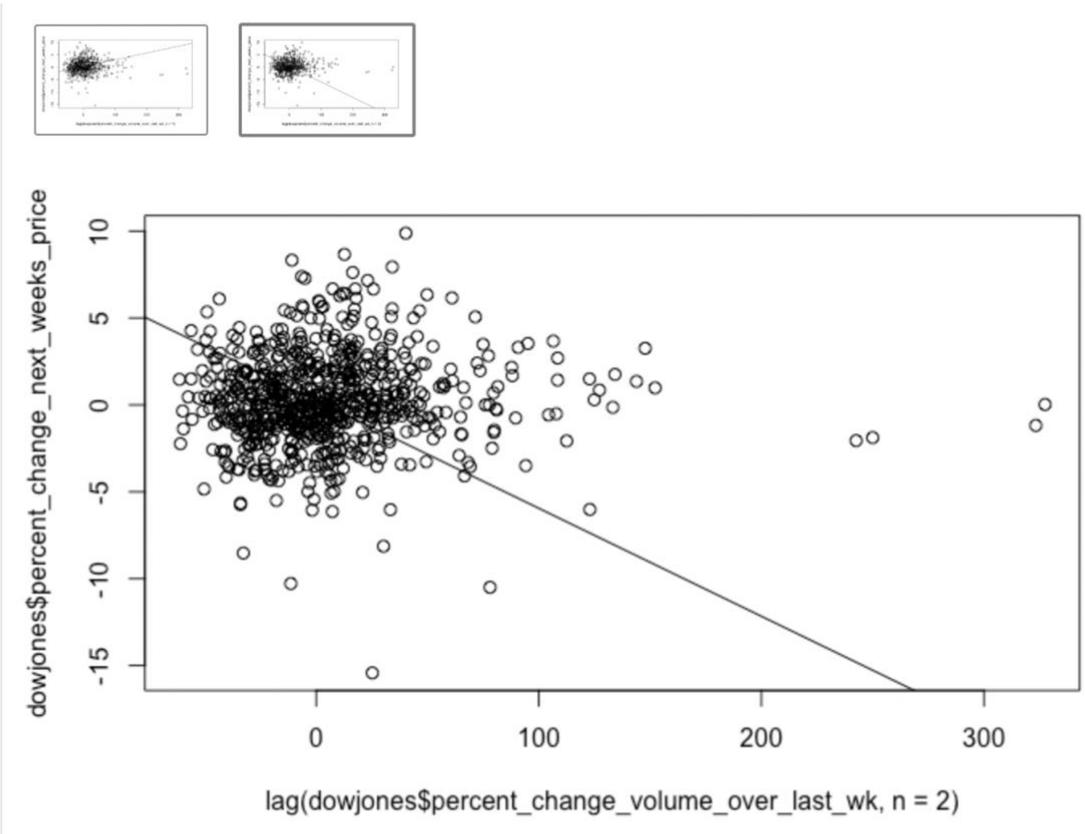


Fig: Lag plot between target variables and variable percent_change_volume_over_last_wk

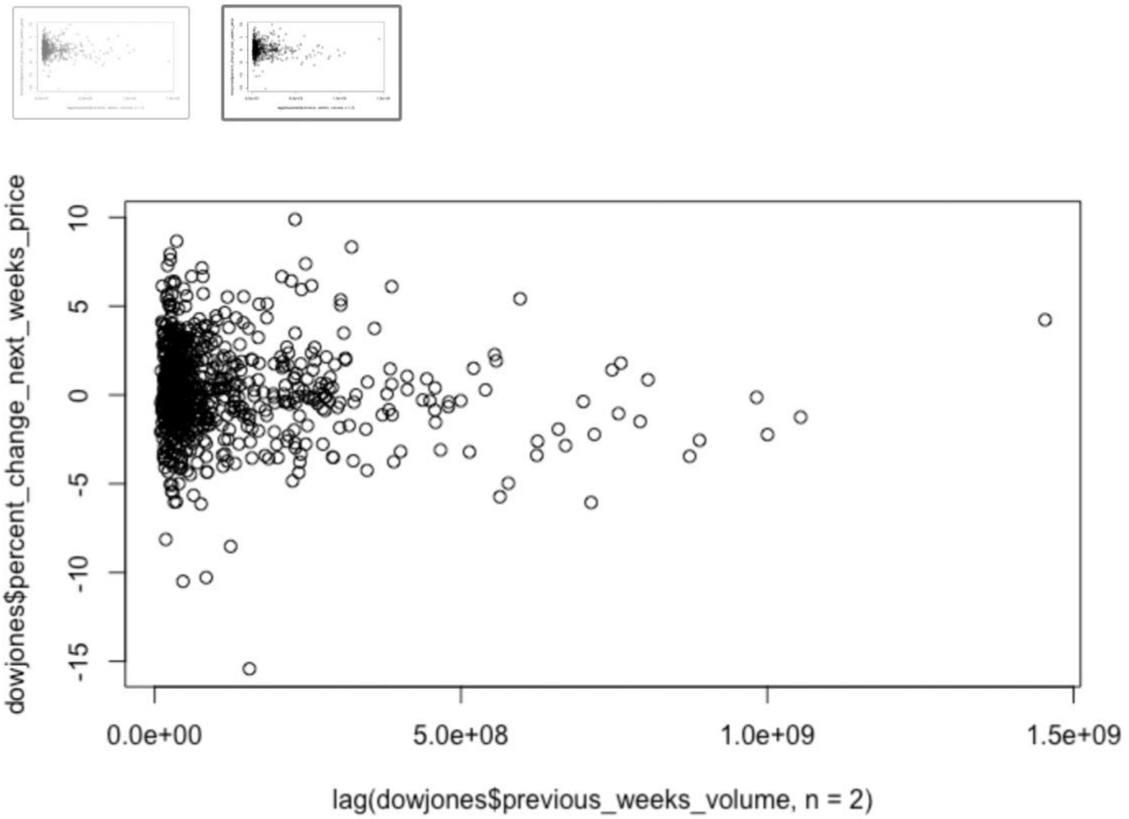


Fig: Lag plot between target variables and variable previous_weeks_volume

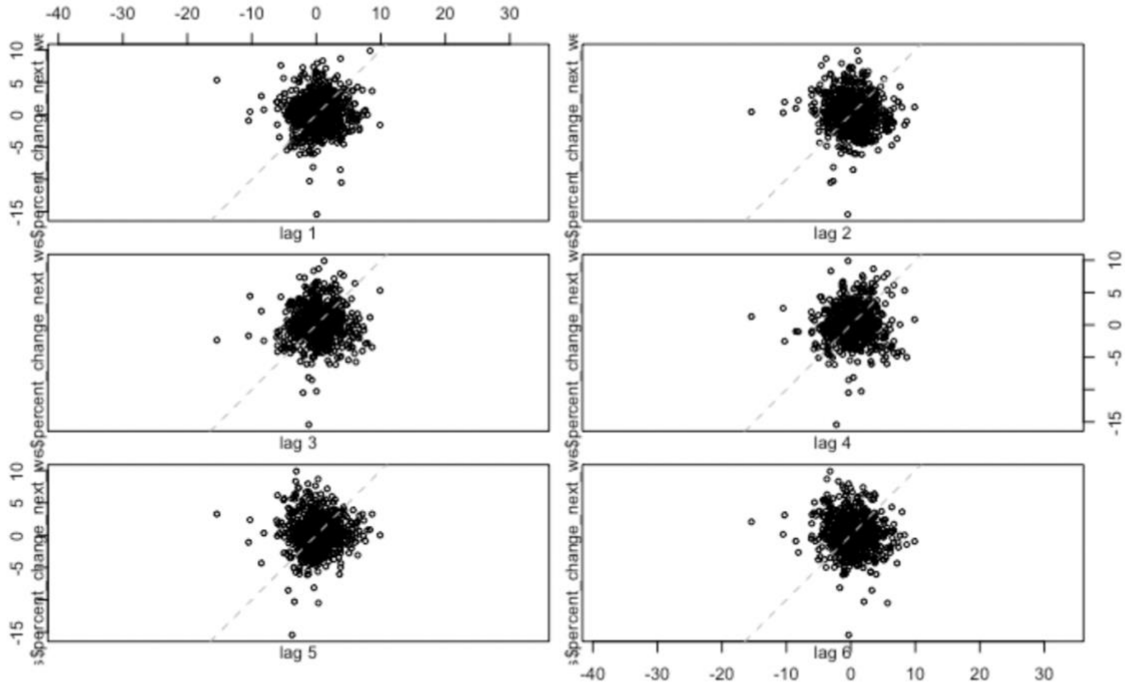


Fig: Lag plot

Findings

The findings for this case study suggests that the Lag 1 data is closer to the absolute line. As such, lag 2 data is spread more entirely. The insights also showed the observation in slight decrease in volume with time. As such, the data is more concentrated near the absolute line in lag 1 as compared to lag 2.

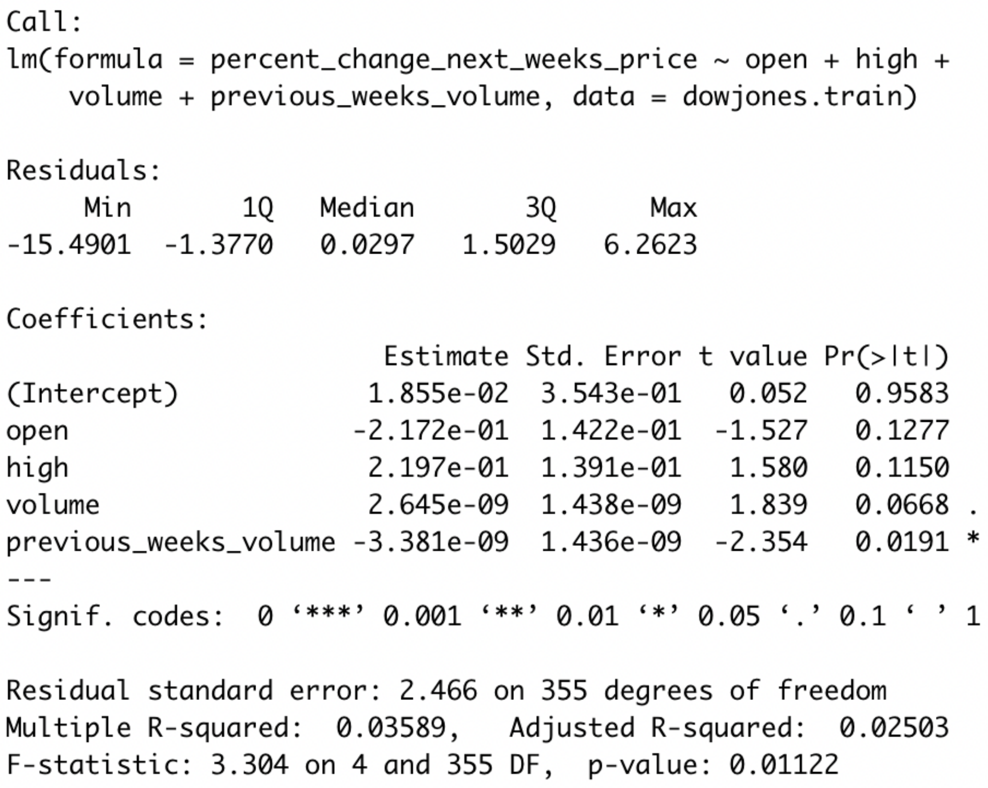
The major findings showed the model linear regression and SVR are better than decision tree when taking the entirety of the stocks while decision tree model yielded very high variance due to the size of the sample.

The MSE value for each of the model are:

- ◊ Linear Regression: 8.01
- ◊ SVM model: 8.05
- ◊ Decision Tree: 11.32
- ◊ Decision Tree (pruned): 9.12

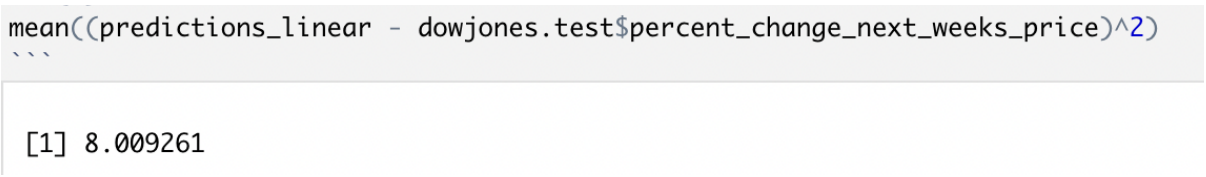
As per the results above, we can conclude that SVR model yielded the best results, while linear regression model had very high variance due to the sample size for the stock individually. As such, decision tree performed well for individual stocks. We can conclude that SVR model yielded better results for the stocks in entirety as well as for the stocks individually.

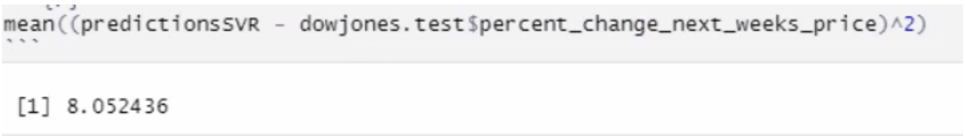
The screenshots of the results yielded are as shown below:



```
Call:
lm(formula = percent_change_next_weeks_price ~ open + high + 
    volume + previous_weeks_volume, data = dowjones.train)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.4901  -1.3770   0.0297   1.5029   6.2623 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)  
(Intercept)            1.855e-02  3.543e-01   0.052   0.9583  
open                  -2.172e-01  1.422e-01  -1.527   0.1277  
high                   2.197e-01  1.391e-01   1.580   0.1150  
volume                 2.645e-09  1.438e-09   1.839   0.0668 .
previous_weeks_volume -3.381e-09  1.436e-09  -2.354   0.0191 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.466 on 355 degrees of freedom
Multiple R-squared:  0.03589,	Adjusted R-squared:  0.02503 
F-statistic: 3.304 on 4 and 355 DF,  p-value: 0.01122
```

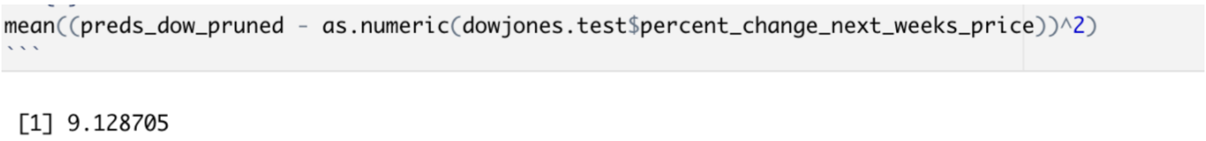
Fig: Linear Regression model prediction

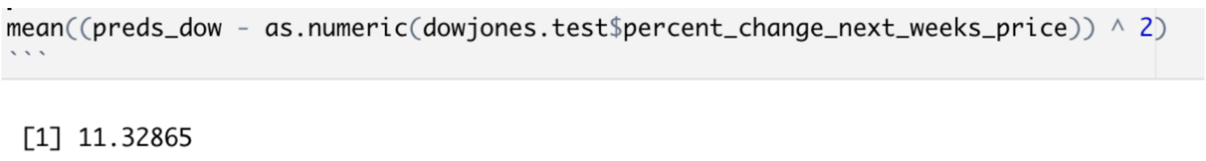
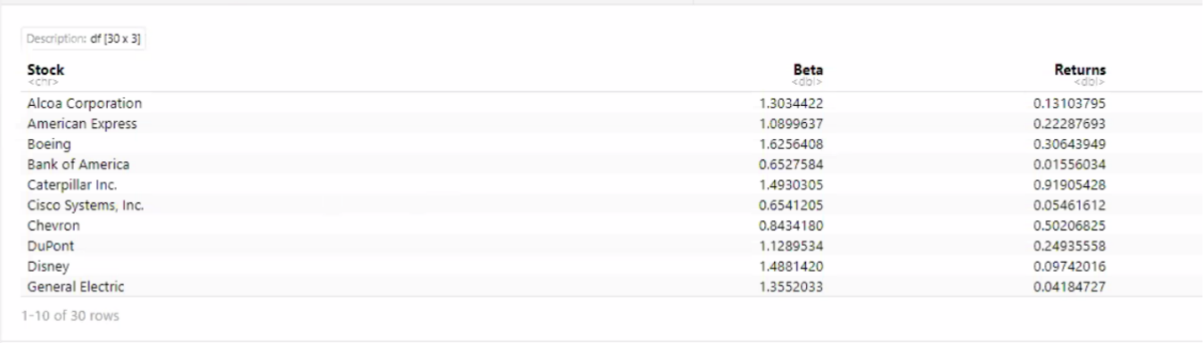
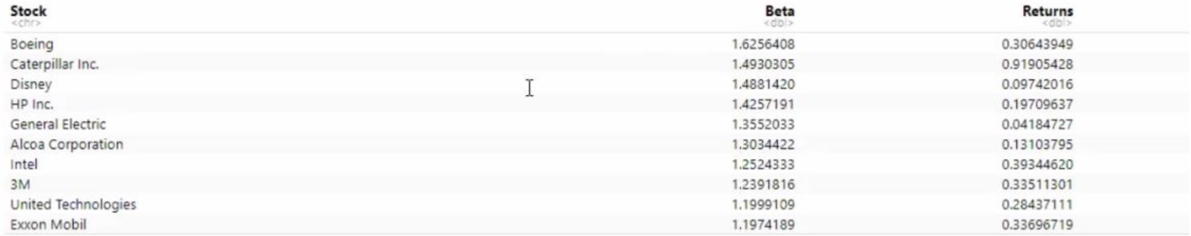
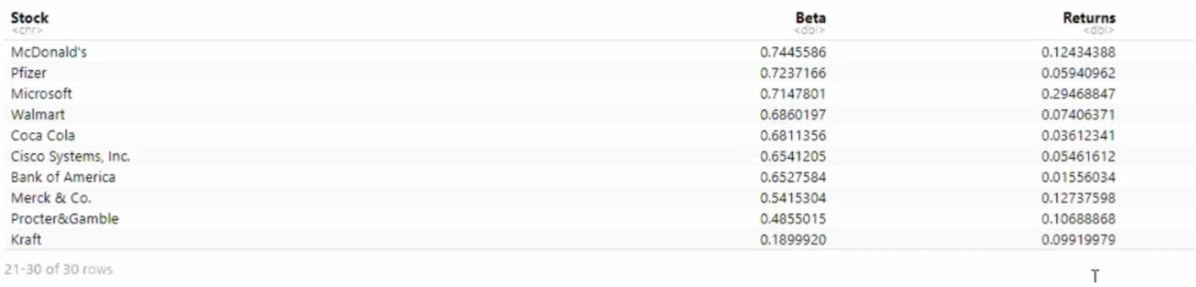
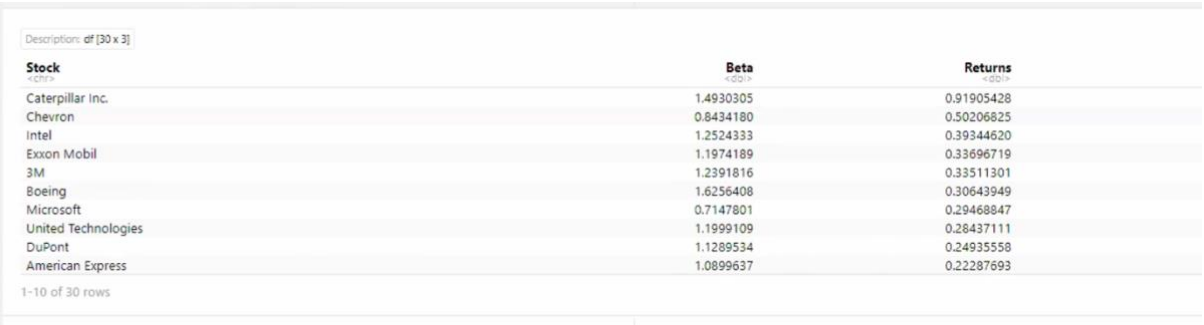
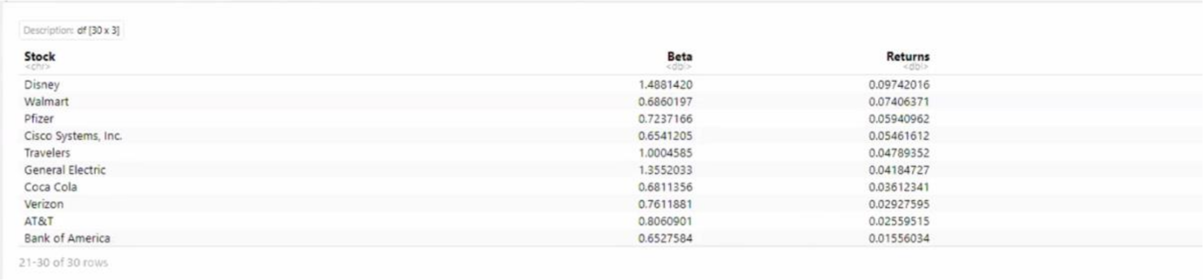
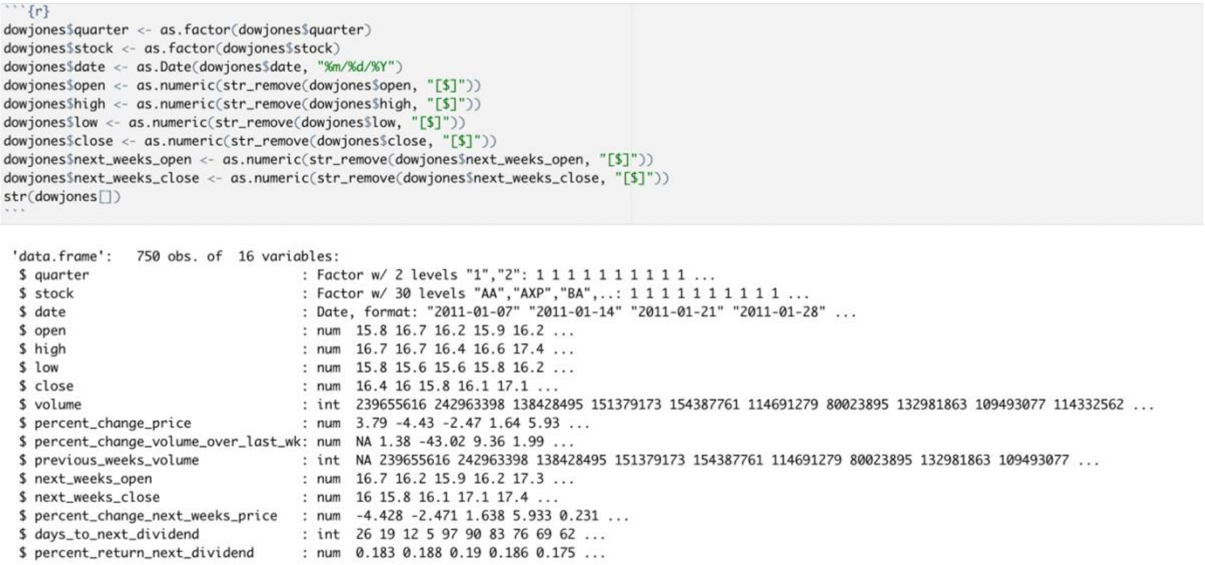
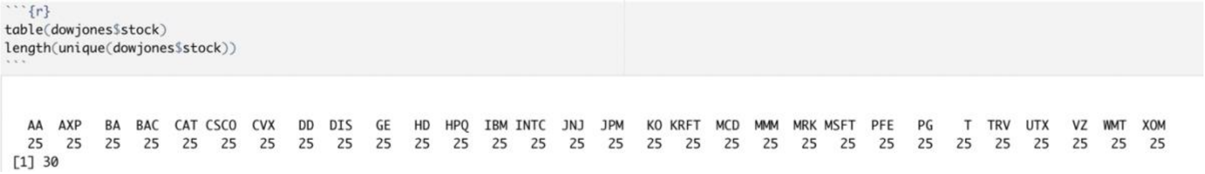
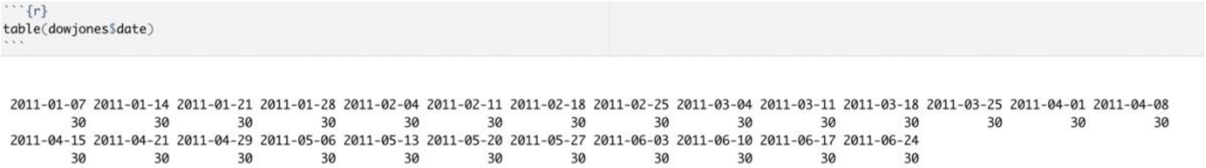
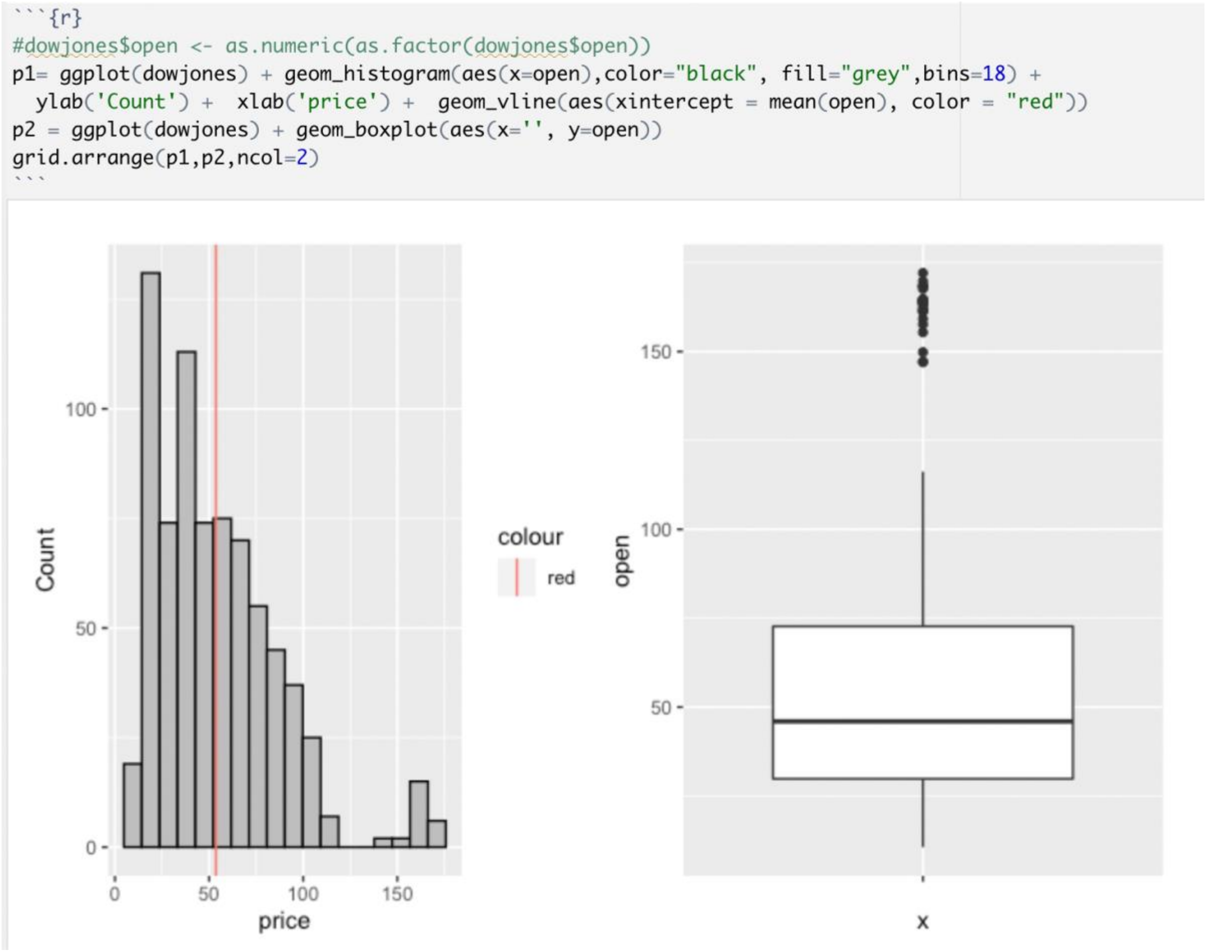
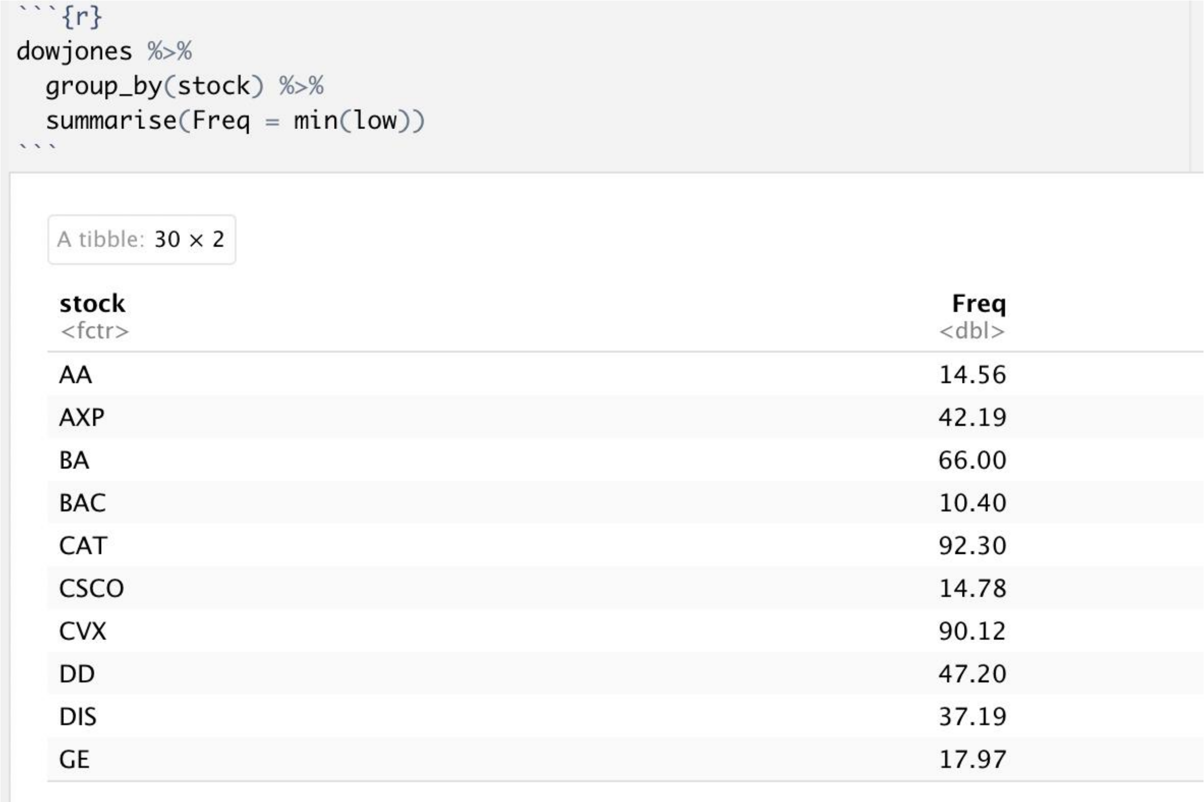
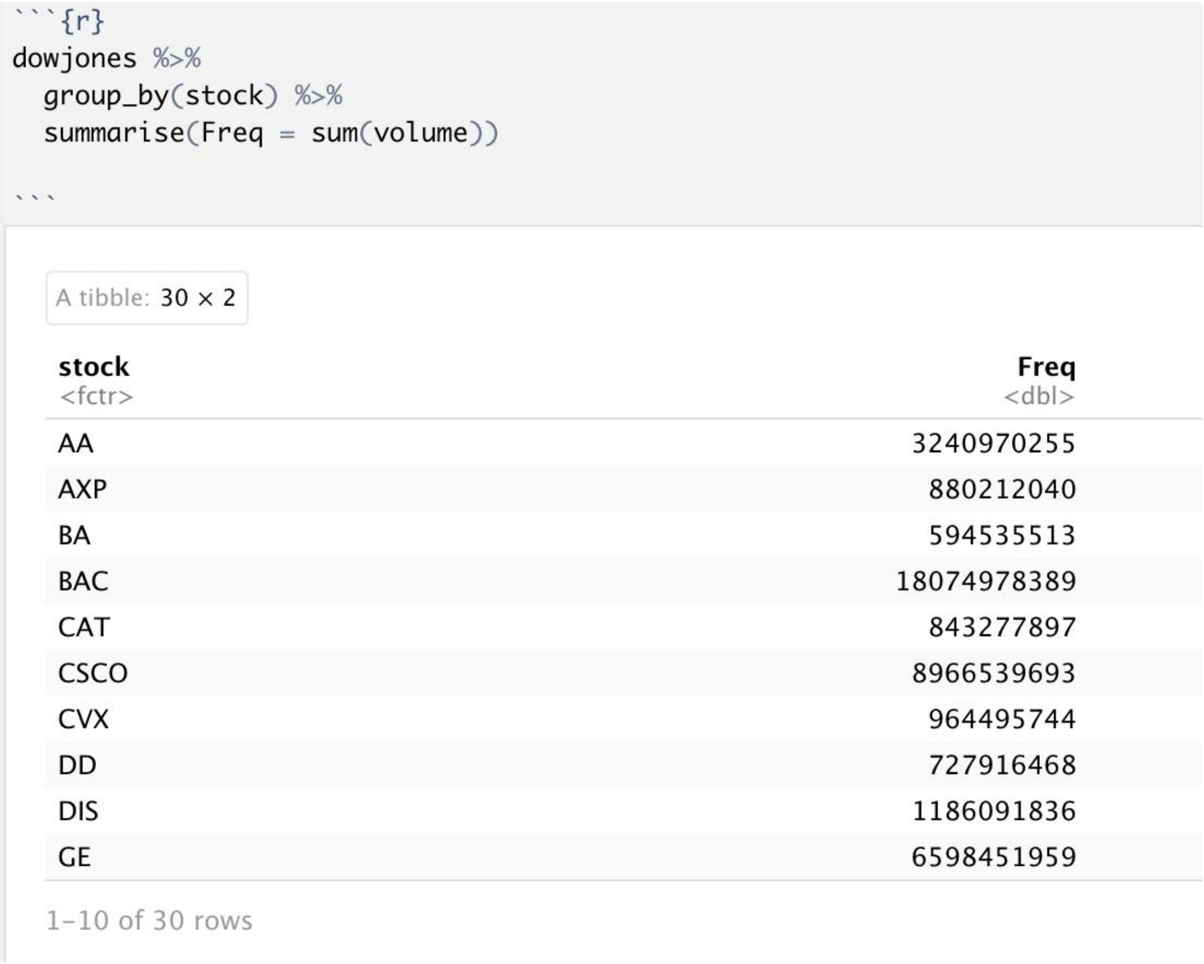
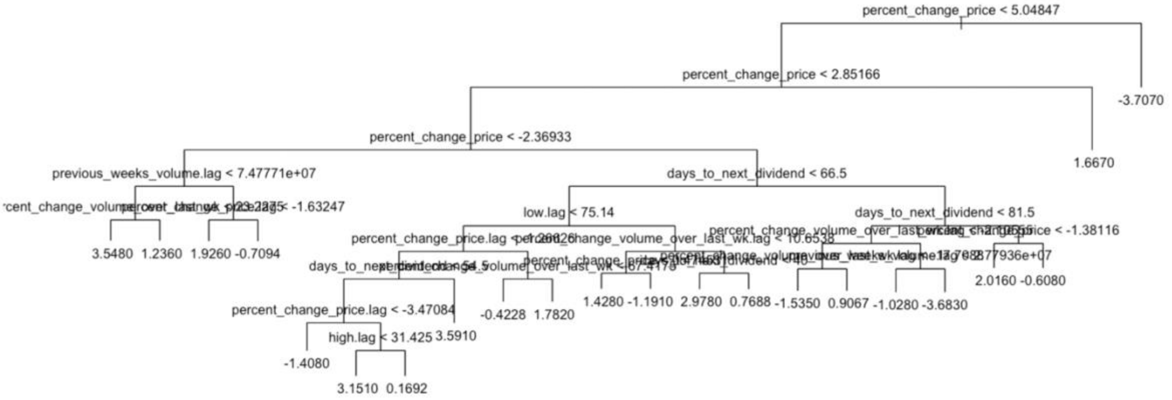
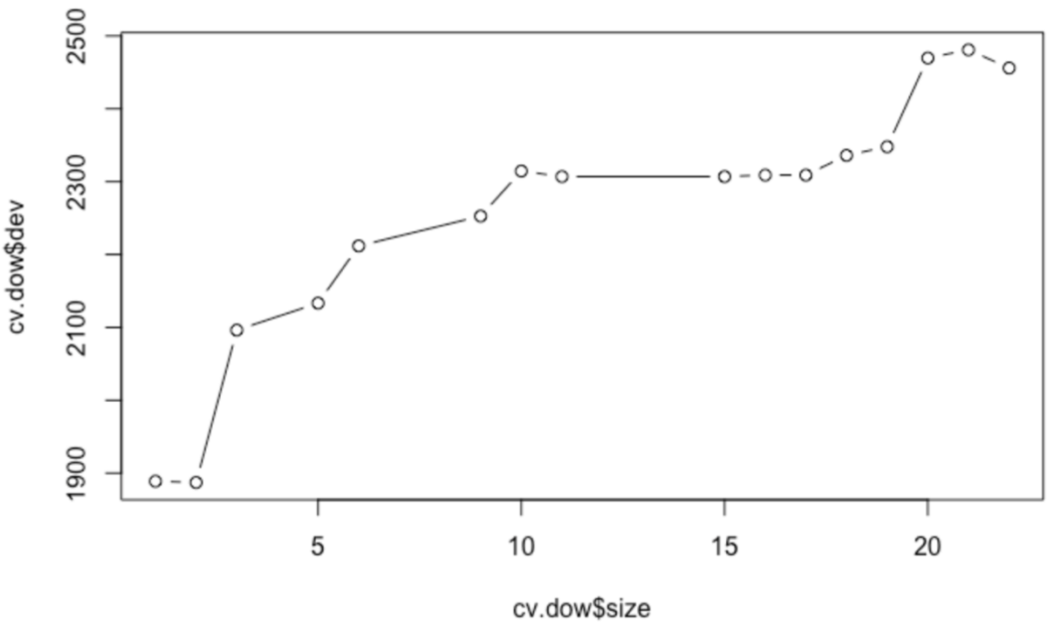
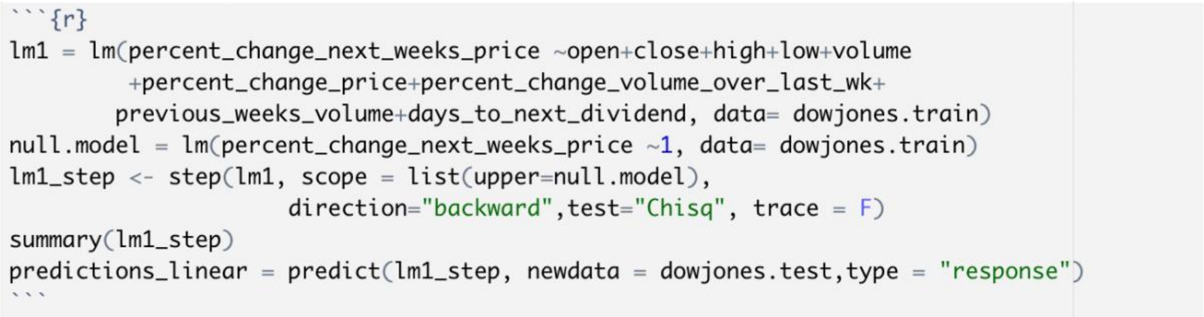
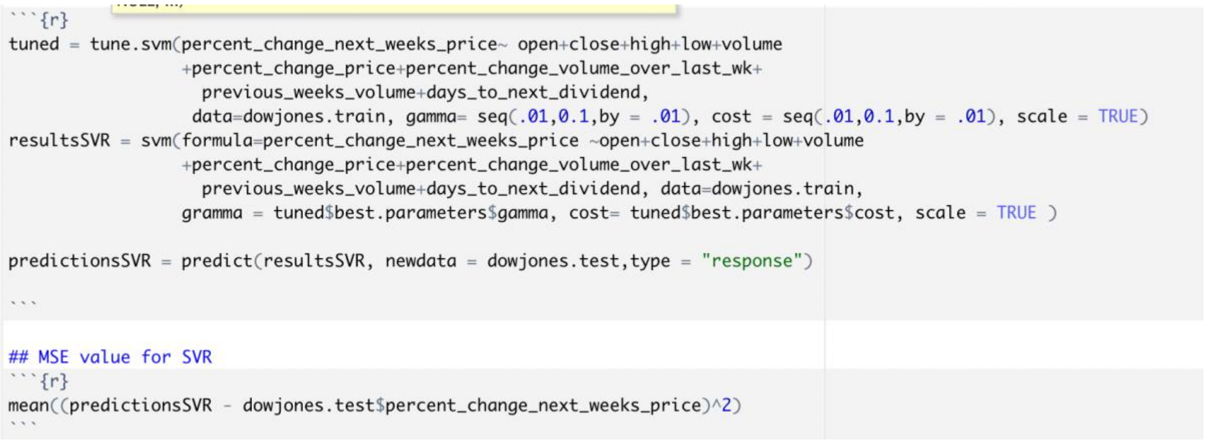


~~~~
mean((predictions_linear - dowjones.test$percent_change_next_weeks_price)^2)
```

[1] 8.009261
~~~~

Fig: MSE for linear regression model



~~~~
mean((predictionsSVR - dowjones.test$percent_change_next_weeks_price)^2)
```

[1] 8.052436
~~~~

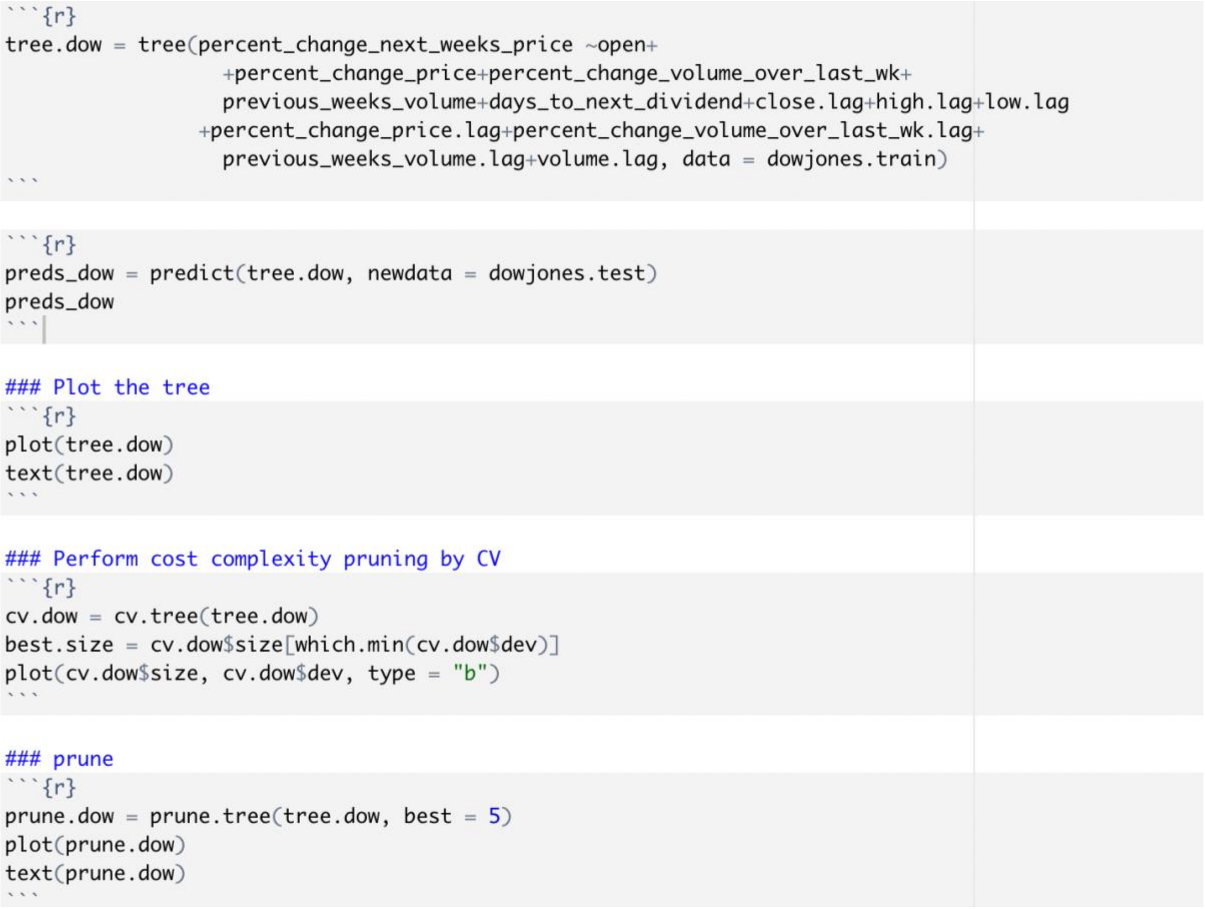
Fig: MSE for SVR model



~~~~
mean((preds_dow_pruned - as.numeric(dowjones.test$percent_change_next_weeks_price))^2)
```

[1] 9.128705
~~~~

Fig: MSE for pruned decision tree model



~~~
mean((preds_dow - as.numeric(dowjones.test$percent_change_next_weeks_price)) ^ 2)
```
~~~

```
 [1] 11.32865
```

Fig: MSE for decision tree model

From the results above, we can see that the linear regression model yielded the mean test error of 8.01.

The decision tree before pruned yielded the mean test error of 11.32 and after pruned, the model yielded the result of 9.12. This shows that the model underperformed in comparison to linear regression model.

The SVR model yielded the mean test error of 8.05.

The next step of process to calculate the risk returns was to implement the CAPM to find the riskiest and safest stocks to invest in which has been shown below.

Capital Asset Pricing Model (CAPM)

The CAPM (Capital Asset Pricing Model) defines the relationship between any systematic risk and the return expectation of the asset (generally stocks). This model is highly used in the financial fields to price risky securities and generate returns for the stocks. One core advantage to CAPM is the objectiveness of the estimation in costs of equity that the model would be able to yield. The basic concept of CAPM can be states as the dealing of the risks and returns on the financial securities while defining them arbitrarily. This would mean that the rate of return that any investor would receive from buying the stocks and holding it for a stated time would be equal to the cash dividends that is receives along with the capital gain.

For the case study, the CAPM model is implemented by calculating the beta of the stock returns. The CAPM beta value represents the relationship between the security risks and returns of the market.



Description: df [30 x 3]

Stock <chr>	Beta <dbl>	Returns <dbl>
Alcoa Corporation	1.3034422	0.13103795
American Express	1.0899637	0.22287693
Boeing	1.6256408	0.30643949
Bank of America	0.6527584	0.01556034
Caterpillar Inc.	1.4930305	0.91905428
Cisco Systems, Inc.	0.6541205	0.05461612
Chevron	0.8434180	0.50206825
DuPont	1.1289534	0.24935558
Disney	1.4881420	0.09742016
General Electric	1.3552033	0.04184727

1-10 of 30 rows

Fig: CAPM result for stocks with the beta value and the prediction returns



Stock <chr>	Beta <dbl>	Returns <dbl>
Boeing	1.6256408	0.30643949
Caterpillar Inc.	1.4930305	0.91905428
Disney	1.4881420	0.09742016
HP Inc.	1.4257191	0.19709637
General Electric	1.3552033	0.04184727
Alcoa Corporation	1.3034422	0.13103795
Intel	1.2524333	0.39344620
3M	1.2391816	0.33511301
United Technologies	1.1999109	0.28437111
Exxon Mobil	1.1974189	0.33696719

Fig: CAPM result for stocks with the highest beta value



Stock <chr>	Beta <dbl>	Returns <dbl>
McDonald's	0.7445586	0.12434388
Pfizer	0.7237166	0.05940962
Microsoft	0.7147801	0.29468847
Walmart	0.6860197	0.07406371
Coca Cola	0.6811356	0.03612341
Cisco Systems, Inc.	0.6541205	0.05461612
Bank of America	0.6527584	0.01556034
Merck & Co.	0.5415304	0.12737598
Procter&Gamble	0.4855015	0.10688868
Kraft	0.1899920	0.09919979

21-30 of 30 rows

Fig: CAPM result for stocks with the lowest beta value



Description: df [30 x 3]

Stock <chr>	Beta <dbl>	Returns <dbl>
Caterpillar Inc.	1.4930305	0.91905428
Chevron	0.8434180	0.50206825
Intel	1.2524333	0.39344620
Exxon Mobil	1.1974189	0.33696719
3M	1.2391816	0.33511301
Boeing	1.6256408	0.30643949
Microsoft	0.7147801	0.29468847
United Technologies	1.1999109	0.28437111
DuPont	1.1289534	0.24935558
American Express	1.0899637	0.22287693

1-10 of 30 rows

Fig: CAPM result for stocks with the highest returns



Description: df [30 x 3]

Stock <chr>	Beta <dbl>	Returns <dbl>
Disney	1.4881420	0.09742016
Walmart	0.6860197	0.07406371
Pfizer	0.7237166	0.05940962
Cisco Systems, Inc.	0.6541205	0.05461612
Travelers	1.0004585	0.04789352
General Electric	1.3552033	0.04184727
Coca Cola	0.6811356	0.03612341
Verizon	0.7611881	0.02927595
AT&T	0.8060901	0.02559515
Bank of America	0.6527584	0.01556034

21-30 of 30 rows

Fig: CAPM result for stocks with the lowest returns

As per the CAPM results, we can see that the riskiest investment in stocks would be in Boeing, Caterpillar Inc., and Disney as per their beta values of 1.62, 1.49 and 1.48 respectively. The model suggests that higher the beta value, higher the volatility and risks for investment and lower the beta value, lower the risks of investment in the said stock. As such, the safest stock for investment are Craft, Procter & Gamble and Merck & Co. respectively with the beta values of 0.18, 0.48 and 0.54.

Conclusion and Recommendation

In conclusion, we can state that the best overall results for the model was provided by the SVR model with better results from individual stocks as well as the overall stock index. As such, the linear regression model performed good for the entire stock index but underperformed with the individual stock predictions. And finally, the decision tree model underperformed with the highest MSE value of 11.2 and for pruned model the MSE value retained to be 9.1, both of which is higher than the other two models stating that decision tree underperformed for the stock index. However, the decision tree model performed the best for the individual stock predictions with better results.

As such, despite that, SVR model came very close to both the models for both the scenarios of individual stock and stock index prediction. As such, SVR can be stated as the better performing model because of its prediction accuracy and the appropriate model for the case study would be SVR with the MSE value of 8.05. As such, the risks of different stocks investment were calculated using the CAPM model which yielded the highest beta value of 1.62 by the company Boeing stating it to be the riskiest and the stock of Craft to be 0.18 which is the lowest and the safest bet for stock prediction returns.

Time series analysis can be applied to study effect of previous n weeks on the next week's return. Time series analysis is commonly used method for prediction of change in stock prices. It can also help to understand trend and seasonality patterns in data. Additional variables can be created, and their effects isolated based on uncommon weeks like 4 business day weeks. The case study could be concluded a little better For overfitting check that SVR and random forest have high Rsquare in training data compared to linear regression but is their MSE lower on test data.

Appendix

References

- ➢ B G. Alfonso, A. D. Carnerero, D. R. Ramirez and T. Alamo, "*Stock Forecasting Using Local Data*," in IEEE Access, vol. 9, pp. 9334-9344, 2021, doi: 10.1109/ACCESS.2020.3047160. Available at: https://ieeexplore.ieee.org/abstract/document/9306750 [Accessed 24 February 2022].
- ➢ R. Ying, Y. Shou, and C. Liu, "*Prediction Model of Dow Jones Index Based on LSTM-Adaboost*," 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), 2021, pp. 808-812, doi: 10.1109/CISCE52179.2021.9445928.
  Available at:
  https://ieeexplore.ieee.org/document/9445928 [Accessed 22 February 2022].

Plots



~~~
```{r}
dowjones$quarter <- as.factor(dowjones$quarter)
dowjones$stock <- as.factor(dowjones$stock)
dowjones$date <- as.Date(dowjones$date, "%m/%d/%Y")
dowjones$open <- as.numeric(str_remove(dowjones$open, "[$]"))
dowjones$high <- as.numeric(str_remove(dowjones$high, "[$]"))
dowjones$low <- as.numeric(str_remove(dowjones$low, "[$]"))
dowjones$close <- as.numeric(str_remove(dowjones$close, "[$]"))
dowjones$next_weeks_open <- as.numeric(str_remove(dowjones$next_weeks_open, "[$]"))
dowjones$next_weeks_close <- as.numeric(str_remove(dowjones$next_weeks_close, "[$]"))
str(dowjones[])
```
~~~

```
'data.frame':   750 obs. of  16 variables:
 $ quarter                           : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ stock                             : Factor w/ 30 levels "AA","AXP","BA",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ date                              : Date, format: "2011-01-07" "2011-01-14" "2011-01-21" "2011-01-28" ...
 $ open                              : num  15.8 16.7 16.2 15.9 16.2 ...
 $ high                              : num  16.7 16.7 16.4 16.6 17.4 ...
 $ low                               : num  15.8 15.6 15.6 15.8 16.2 ...
 $ close                             : num  16.4 16 15.8 16.1 17.1 ...
 $ volume                            : int  239655616 242963398 138428495 151379173 154387761 114691279 80023895 132981863 109493077 114332562 ...
 $ percent_change_price              : num  3.79 -4.43 -2.47 1.64 5.93 ...
 $ percent_change_volume_over_last_wk: num  NA 1.38 -43.02 9.36 1.99 ...
 $ previous_weeks_volume             : int  NA 239655616 242963398 138428495 151379173 154387761 114691279 80023895 132981863 109493077 ...
 $ next_weeks_open                   : num  16.7 16.2 15.9 16.2 17.3 ...
 $ next_weeks_close                  : num  16 15.8 16.1 17.1 17.4 ...
 $ percent_change_next_weeks_price   : num  -4.428 -2.471 1.638 5.933 0.231 ...
 $ days_to_next_dividend             : int  26 19 12 5 97 90 83 76 69 62 ...
 $ percent_return_next_dividend      : num  0.183 0.188 0.19 0.186 0.175 ...
```

Fig: Changing the variable types and prefix.



~~~
```{r}
table(dowjones$stock)
length(unique(dowjones$stock))
```
~~~

```
  AA  AXP   BA  BAC  CAT CSCO  CVX   DD  DIS   GE   HD  HPQ  IBM INTC  JNJ  JPM   KO KRFT  MCD  MMM  MRK MSFT  PFE   PG    T  TRV  UTX   VZ  WMT  XOM 
  25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25   25 
[1] 30
```

Fig: Bar plot analysis for the stock variable



~~~
```{r}
table(dowjones$date)
```
~~~

```
2011-01-07 2011-01-14 2011-01-21 2011-01-28 2011-02-04 2011-02-11 2011-02-18 2011-02-25 2011-03-04 2011-03-11 2011-03-18 2011-03-25 2011-04-01 2011-04-08 
        30         30         30         30         30         30         30         30         30         30         30         30         30         30 
2011-04-15 2011-04-21 2011-04-29 2011-05-06 2011-05-13 2011-05-20 2011-05-27 2011-06-03 2011-06-10 2011-06-17 2011-06-24 
        30         30         30         30         30         30         30         30         30         30         30 
```

Fig: Bar plot analysis for the date variable



```{r}
#dowjones$open <- as.numeric(as.factor(dowjones$open))
p1= ggplot(dowjones) + geom_histogram(aes(x=open),color="black", fill="grey",bins=18) +
  ylab('Count') +  xlab('price') +  geom_vline(aes(xintercept = mean(open), color = "red"))
p2 = ggplot(dowjones) + geom_boxplot(aes(x='', y=open))
grid.arrange(p1,p2,ncol=2)
```

Fig: Bar plot analysis for the open variable



~~~
```{r}
dowjones %>%
  group_by(stock) %>%
  summarise(Freq = min(low))
```
~~~

A tibble: 30 × 2

stock	Freq
<fctr>	<dbl>
AA	14.56
AXP	42.19
BA	66.00
BAC	10.40
CAT	92.30
CSCO	14.78
CVX	90.12
DD	47.20
DIS	37.19
GE	17.97

Fig: Summaries



```{r}
dowjones %>%
  group_by(stock) %>%
  summarise(Freq = sum(volume))

```

A tibble: 30 × 2

stock <fctr>	Freq <dbl>
AA	3240970255
AXP	880212040
BA	594535513
BAC	18074978389
CAT	843277897
CSCO	8966539693
CVX	964495744
DD	727916468
DIS	1186091836
GE	6598451959

1–10 of 30 rows

Fig: Summaries


percent_change_price < 5.04847
percent_change_price < 2.85166
-3.7070
percent_change_price < -2.36933
1.6670
previous_weeks_volume.lag < 7.47771e+07
days_to_next_dividend < 66.5
< -1.63247
low.lag < 75.14
days_to_next_dividend < 81.5
percent_change_price < -1.38116
3.5480
1.2360
1.9260
-0.7094
percent_change_price.lag
percent_change_volume_over_last_wk.lag < 10.6538
days_to_next_dividend < 54.5
2.0160
-0.6080
1.4280
-1.1910
2.9780
0.7688
-1.5350
0.9067
-1.0280
-3.6830
percent_change_price.lag < -3.47084
-0.4228
1.7820
high.lag < 31.425
3.5910
-1.4080
3.1510
0.1692

Fig: Initial Decision Tree



Fig: Pruned decision tree

Code for the model



~~~
```{r}
lm1 = lm(percent_change_next_weeks_price ~open+close+high+low+volume
         +percent_change_price+percent_change_volume_over_last_wk+
        previous_weeks_volume+days_to_next_dividend, data= dowjones.train)
null.model = lm(percent_change_next_weeks_price ~1, data= dowjones.train)
lm1_step <- step(lm1, scope = list(upper=null.model),
                     direction="backward",test="Chisq", trace = F)
summary(lm1_step)
predictions_linear = predict(lm1_step, newdata = dowjones.test,type = "response")
```
~~~

Fig: Code for Linear Regression Model



~~~
```{r}
tuned = tune.svm(percent_change_next_weeks_price~ open+close+high+low+volume
                 +percent_change_price+percent_change_volume_over_last_wk+
                   previous_weeks_volume+days_to_next_dividend,
                  data=dowjones.train, gamma= seq(.01,0.1,by = .01), cost = seq(.01,0.1,by = .01), scale = TRUE)
resultsSVR = svm(formula=percent_change_next_weeks_price ~open+close+high+low+volume
                 +percent_change_price+percent_change_volume_over_last_wk+
                   previous_weeks_volume+days_to_next_dividend, data=dowjones.train,
                 gramma = tuned$best.parameters$gamma, cost= tuned$best.parameters$cost, scale = TRUE )

predictionsSVR = predict(resultsSVR, newdata = dowjones.test,type = "response")

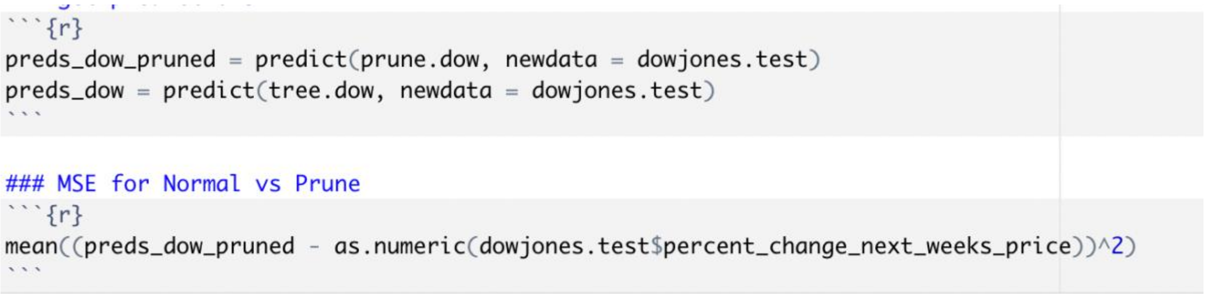
~~~

MSE value for SVR
```{r}
mean((predictionsSVR - dowjones.test$percent_change_next_weeks_price)^2)
```
```

Fig: Code for SVR Model



```
```{r}
tree.dow = tree(percent_change_next_weeks_price ~open+
                  +percent_change_price+percent_change_volume_over_last_wk+
                  previous_weeks_volume+days_to_next_dividend+close.lag+high.lag+low.lag
                +percent_change_price.lag+percent_change_volume_over_last_wk.lag+
                  previous_weeks_volume.lag+volume.lag, data = dowjones.train)
```

```{r}
preds_dow = predict(tree.dow, newdata = dowjones.test)
preds_dow
```

Plot the tree
```{r}
plot(tree.dow)
text(tree.dow)
```

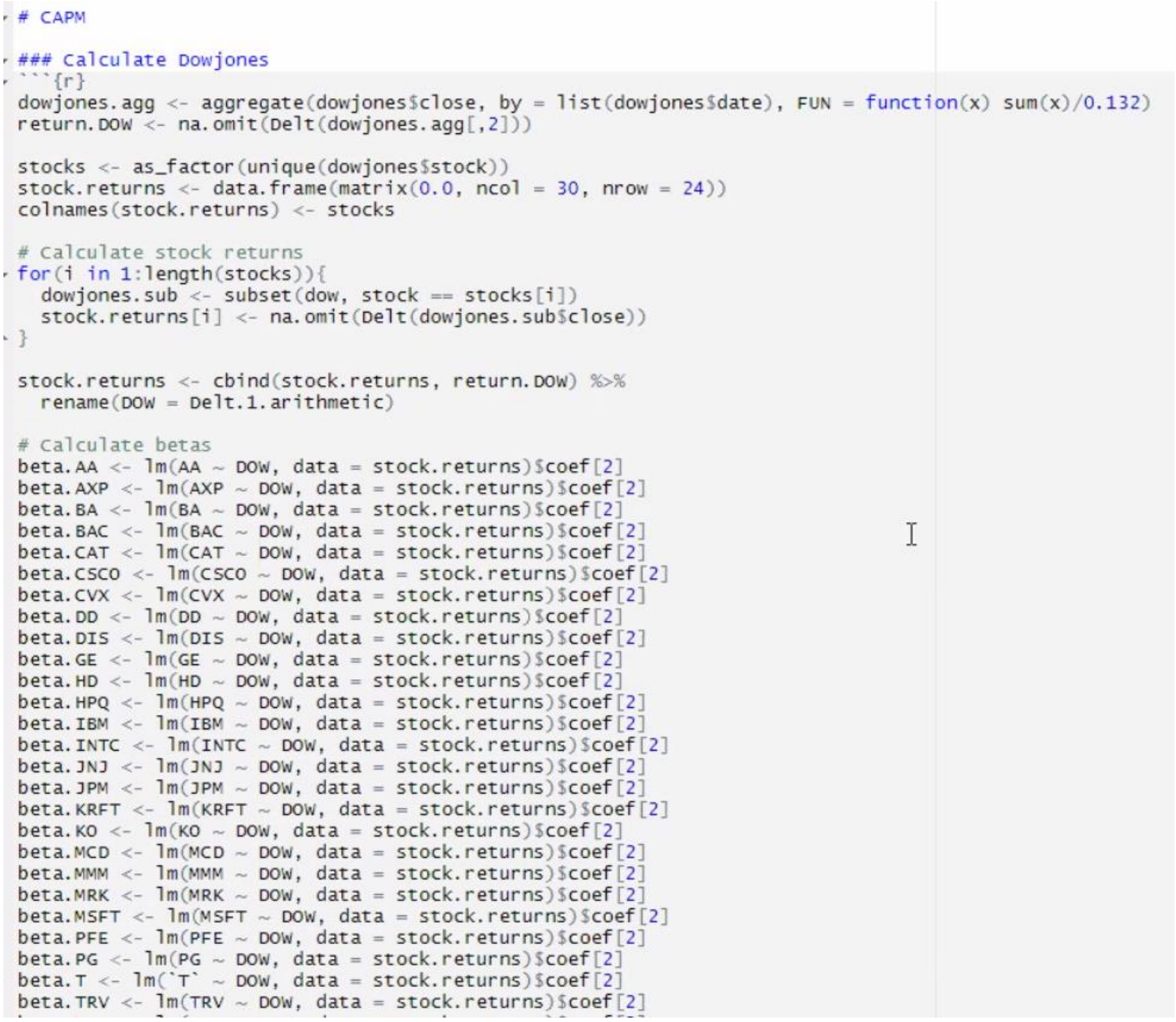
Perform cost complexity pruning by CV
```{r}
cv.dow = cv.tree(tree.dow)
best.size = cv.dow$size[which.min(cv.dow$dev)]
plot(cv.dow$size, cv.dow$dev, type = "b")
```

prune
```{r}
prune.dow = prune.tree(tree.dow, best = 5)
plot(prune.dow)
text(prune.dow)
```
```



```
```{r}
preds_dow_pruned = predict(prune.dow, newdata = dowjones.test)
preds_dow = predict(tree.dow, newdata = dowjones.test)
```

MSE for Normal vs Prune
```{r}
mean((preds_dow_pruned - as.numeric(dowjones.test$percent_change_next_weeks_price))^2)
```
```

```
[1] 9.128705
```
```

*Fig: Code for Decision Tree Model*



~~~
# CAPM

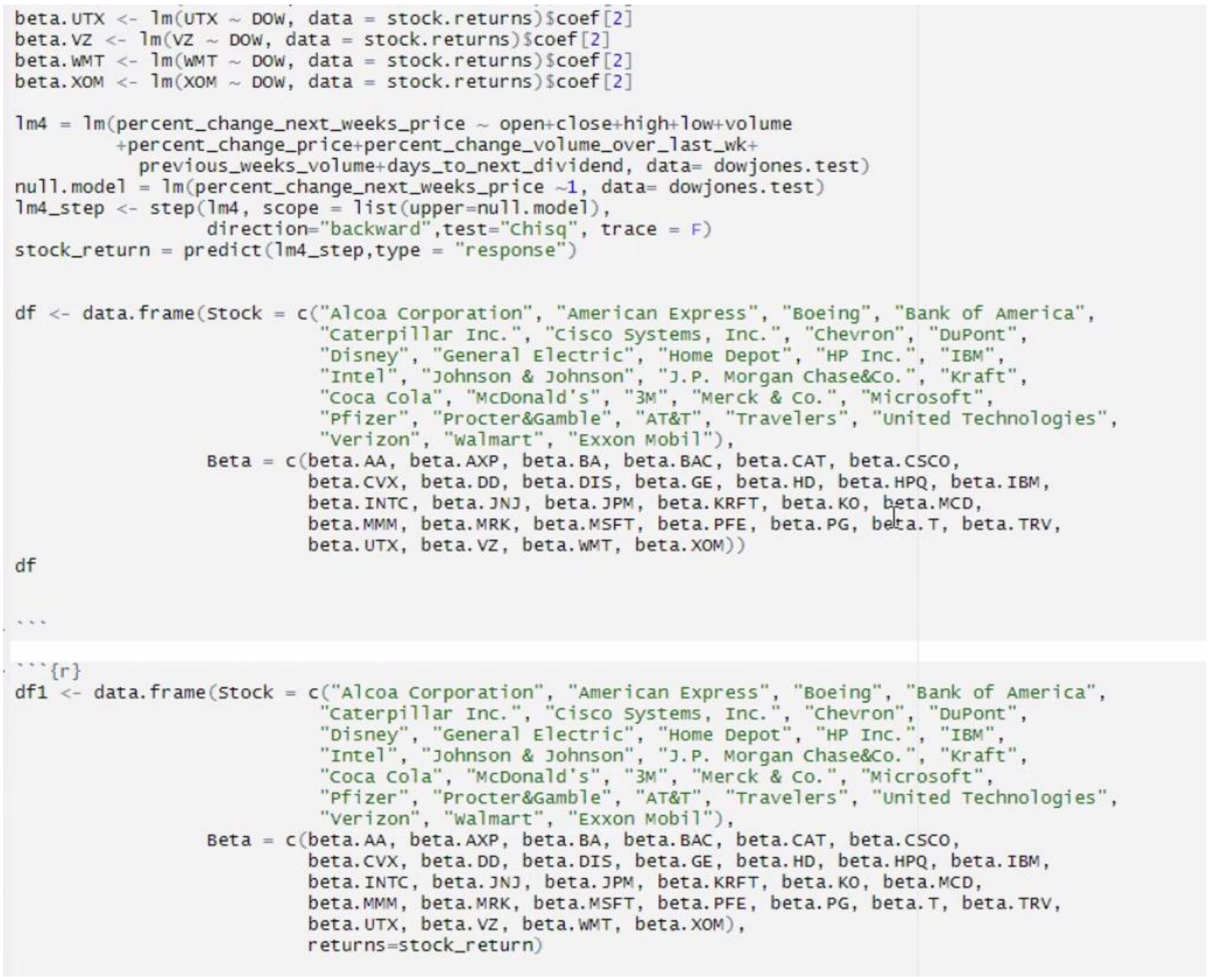
### Calculate Dowjones
```{r}
dowjones.agg <- aggregate(dowjones$close, by = list(dowjones$date), FUN = function(x) sum(x)/0.132)
return.DOW <- na.omit(Delt(dowjones.agg[,2]))

stocks <- as_factor(unique(dowjones$stock))
stock.returns <- data.frame(matrix(0.0, ncol = 30, nrow = 24))
colnames(stock.returns) <- stocks

# Calculate stock returns
for(i in 1:length(stocks)){
  dowjones.sub <- subset(dow, stock == stocks[i])
  stock.returns[i] <- na.omit(Delt(dowjones.sub$close))
}

stock.returns <- cbind(stock.returns, return.DOW) %>%
  rename(DOW = Delt.1.arithmetic)

# Calculate betas
beta.AA <- lm(AA ~ DOW, data = stock.returns)$coef[2]
beta.AXP <- lm(AXP ~ DOW, data = stock.returns)$coef[2]
beta.BA <- lm(BA ~ DOW, data = stock.returns)$coef[2]
beta.BAC <- lm(BAC ~ DOW, data = stock.returns)$coef[2]
beta.CAT <- lm(CAT ~ DOW, data = stock.returns)$coef[2]
beta.CSCO <- lm(CSCO ~ DOW, data = stock.returns)$coef[2]
beta.CVX <- lm(CVX ~ DOW, data = stock.returns)$coef[2]
beta.DD <- lm(DD ~ DOW, data = stock.returns)$coef[2]
beta.DIS <- lm(DIS ~ DOW, data = stock.returns)$coef[2]
beta.GE <- lm(GE ~ DOW, data = stock.returns)$coef[2]
beta.HD <- lm(HD ~ DOW, data = stock.returns)$coef[2]
beta.HPQ <- lm(HPQ ~ DOW, data = stock.returns)$coef[2]
beta.IBM <- lm(IBM ~ DOW, data = stock.returns)$coef[2]
beta.INTC <- lm(INTC ~ DOW, data = stock.returns)$coef[2]
beta.JNJ <- lm(JNJ ~ DOW, data = stock.returns)$coef[2]
beta.JPM <- lm(JPM ~ DOW, data = stock.returns)$coef[2]
beta.KRFT <- lm(KRFT ~ DOW, data = stock.returns)$coef[2]
beta.KO <- lm(KO ~ DOW, data = stock.returns)$coef[2]
beta.MCD <- lm(MCD ~ DOW, data = stock.returns)$coef[2]
beta.MMM <- lm(MMM ~ DOW, data = stock.returns)$coef[2]
beta.MRK <- lm(MRK ~ DOW, data = stock.returns)$coef[2]
beta.MSFT <- lm(MSFT ~ DOW, data = stock.returns)$coef[2]
beta.PFE <- lm(PFE ~ DOW, data = stock.returns)$coef[2]
beta.PG <- lm(PG ~ DOW, data = stock.returns)$coef[2]
beta.T <- lm(`T` ~ DOW, data = stock.returns)$coef[2]
beta.TRV <- lm(TRV ~ DOW, data = stock.returns)$coef[2]
~~~



~~~~
beta.UTX <- lm(UTX ~ DOW, data = stock.returns)$coef[2]
beta.VZ <- lm(VZ ~ DOW, data = stock.returns)$coef[2]
beta.WMT <- lm(WMT ~ DOW, data = stock.returns)$coef[2]
beta.XOM <- lm(XOM ~ DOW, data = stock.returns)$coef[2]

lm4 = lm(percent_change_next_weeks_price ~ open+close+high+low+volume
         +percent_change_price+percent_change_volume_over_last_wk+
           previous_weeks_volume+days_to_next_dividend, data= dowjones.test)
null.model = lm(percent_change_next_weeks_price ~1, data= dowjones.test)
lm4_step <- step(lm4, scope = list(upper=null.model),
                 direction="backward",test="Chisq", trace = F)
stock_return = predict(lm4_step,type = "response")


df <- data.frame(Stock = c("Alcoa Corporation", "American Express", "Boeing", "Bank of America",
                           "Caterpillar Inc.", "Cisco Systems, Inc.", "Chevron", "DuPont",
                           "Disney", "General Electric", "Home Depot", "HP Inc.", "IBM",
                           "Intel", "Johnson & Johnson", "J.P. Morgan Chase&Co.", "Kraft",
                           "Coca Cola", "McDonald's", "3M", "Merck & Co.", "Microsoft",
                           "Pfizer", "Procter&Gamble", "AT&T", "Travelers", "United Technologies",
                           "Verizon", "Walmart", "Exxon Mobil"),
                 Beta = c(beta.AA, beta.AXP, beta.BA, beta.BAC, beta.CAT, beta.CSCO,
                          beta.CVX, beta.DD, beta.DIS, beta.GE, beta.HD, beta.HPQ, beta.IBM,
                          beta.INTC, beta.JNJ, beta.JPM, beta.KRFT, beta.KO, beta.MCD,
                          beta.MMM, beta.MRK, beta.MSFT, beta.PFE, beta.PG, beta.T, beta.TRV,
                          beta.UTX, beta.VZ, beta.WMT, beta.XOM))
df

```

```{r}
df1 <- data.frame(Stock = c("Alcoa Corporation", "American Express", "Boeing", "Bank of America",
                           "Caterpillar Inc.", "Cisco Systems, Inc.", "Chevron", "DuPont",
                           "Disney", "General Electric", "Home Depot", "HP Inc.", "IBM",
                           "Intel", "Johnson & Johnson", "J.P. Morgan Chase&Co.", "Kraft",
                           "Coca Cola", "McDonald's", "3M", "Merck & Co.", "Microsoft",
                           "Pfizer", "Procter&Gamble", "AT&T", "Travelers", "United Technologies",
                           "Verizon", "Walmart", "Exxon Mobil"),
                 Beta = c(beta.AA, beta.AXP, beta.BA, beta.BAC, beta.CAT, beta.CSCO,
                          beta.CVX, beta.DD, beta.DIS, beta.GE, beta.HD, beta.HPQ, beta.IBM,
                          beta.INTC, beta.JNJ, beta.JPM, beta.KRFT, beta.KO, beta.MCD,
                          beta.MMM, beta.MRK, beta.MSFT, beta.PFE, beta.PG, beta.T, beta.TRV,
                          beta.UTX, beta.VZ, beta.WMT, beta.XOM),
                          returns=stock_return)
~~~~

*Fig: Code for CAPM*