Daniel Hong

daniel.hong@spsmail.cuny.edu

DATA 698 Captsone

Stock Market Forecast Methods

Table of Contents

**Introduction1**

**Background1**

**Research3**

Figure 1. SPY Prices5

Figure 2. S&P 500 Regressors5

Figure 3. Distribution6

Figure 4. Density7

Figure 5. Boxplots7

Figure 6. Correlation Plot8

Figure 7. Regression Output…………………………………………………………………………………………………………………8

Figure 8. Relative Importance………………………………………………………………………………………………………………9

**Individual Stocks9**

Figure 10. FAANG Stocks Returns.………………………………………………………………………………………………………10

Figure 11. FB Log Returns……….....………………………………………………………………………………………………………10

Figure 12. ADF……………….……….....………………………………………………………………………………………………………11

Figure 13. ACF……………….……….....………………………………………………………………………………………………………11

Figure 14. PACF…………….……….....……………………………………………………………………………………………………….12

Figure 15. Actual vs. Forecasted..……………………………………………………………………………………………………….12

**Efficient or Inefficient Markets13**

Figure 15. Important Indicators13

Figure 16. Coefficient and Selection14

**Additional Work14**

**References16**

**Appendix (Code)17**

**Introduction**

The goal of this Capstone project is to apply research and signal processing techniques to develop a model for the prices of certain equities traded in the S&P 500 index and the overall index with the intent of producing short-term forecasts. Because of the short-term nature, ARIMA is one of the forecasting models employed to predict stock prices to confirm the hypothesis mean reversion and momentum strategies can be used to outperform a buy and hold strategy over time. To clearly outline the hypothesis, asset prices (in this case stocks) are mean reverting, the theory that prices eventually move back to their average value. The final model utilizes previous stock prices and exchange index values (S&P 500) to provide a tool for making informed stock picking decisions. Additional models will look at linear regression and machine learning techniques. This model is tested using different iterations and will be completed over the course of an academic term with three distinct phases of stock price model development. The purpose of this project is to create models to predict future stock prices and compare strategies, linear regression, ARIMA, Neural Networks, Support Vector Machines and Multivariate Adaptive Regression Splines. Each of these strategies have been selected because of the strong following and an interest in bringing a systematic stock selection model to those with an interest.

The development of this model can help those interested in generating stock portfolios not only through diversification of industries but other factors such as relative strength and timing. Work in this area has helped individual investors immensely and further research can continue to pay dividends, especially if it can be passed to those with little knowledge of the workings of stock markets. This tool can allow those interested in earning a profit without having to perform research of their own, although this may not be a great idea as some degree of investigation should be done separately. The notion came from classes undertook in Finance where analysts have created systematic approaches like Joel Greenblatt’s Magic Formula Investing that has seen immense success over the years. Many people, cynical or not, still believe we live in an age where only big investment banks or a hedge funds “in the know” make money trading stocks. If successful, the implementation of these models would allow individuals to have a better chance of outperforming the overall market, something many professionals cannot achieve.

Over the course of the first period, linear regression was performed on the overall market represented by the SPY, an ETF tracking the S&P 500 index. Variables most relevant to the index price movement are investigated to see what the drivers are, however additional analysis needs to be performed to see what factors drive short term price gains in underlying stocks. Various analysis techniques will be used to implement a mathematical model of the daily closing prices of stocks. The base model only looks at previous values of the given stock’s price. Different combinations of analysis methods will be experimented with and those which provide the most accurate forecasts will be chosen for future research. For the second phase of the project, efforts will be focused on the development of a ARIMA model for individual stocks. The working theory is that individual stocks (bottom-up investing) provide greater value than just buying the index (top-down investing). Combing these two methods could bolster alpha or excessive returns. In either scenario, the least forecast error (possibly measured by RMSE) will be chosen for future analysis. It may be outside of the scope of this project but integrating company research, professional opinions, news sentiment may help make more informed decisions.

**Background**

The concept of the modern stock market can be traced back to 17th century Europe where wealthy businessmen would purchase goods for merchants’ travels in return for better deals on returning goods or for a share of the profits. This concept can be equated to modern day dividends and was thought to be mutually beneficial assuming the merchant could be trusted to remain an ongoing concern. In both modern day and in a historical context, profits should increase as the parties acquire more trading resources and the counterparts gain increasingly more money. In modern times, this basic concept exists on a much larger scale as there are far more public and private companies. The stock market represents publicly traded companies that share the wealth of success as well as the defeats and will be the focus of this research. Companies today operate globally and require in depth analysis to understand if investments will come to fruition as complexity has grown immensely. Stock market investment today provides a much higher level of risk, however with this risk should come a greater magnitude of reward with the possible exposure to losses. Improvements in technology have made the stock market a way for nearly anyone to participate with capital and information. It has become a widely used mechanism for investments of retirement savings and other capital. The increasing popularity of trading in stocks has been facilitated by the accessibility of information and other trading resources, e.g. trading platforms, methods, analytic tools. Modern financial concepts allow for investments in either direction, positive or negative changes, for stock values. A stock can be either bought (long) with the hopes that its value

will rise or sold (short) based on the assumption that it will decrease in value. If an investor could correctly predict the future value of stock prices, profits would be realized regardless of the outcome. Efficient market infrastructure, matching buyers and sellers, allows for liquidity and massive amounts of data collection which makes the stock market a very good candidate for modeling. Various research applies knowledge from signal analysis with the goal of interpreting the price of a stock as a discrete signal with a real value for both time and magnitude. Market participants will see prices fluctuate throughout a trading day 09:30 to 16:00 (US equities) but the analysis methods used in this project focuses on the closing prices of a day.

**Research**

Autocorrelation is one of the most important aspects of short-term stock signal modeling as it finds the applications to the model due to correlation to the most recent data. We account for a trailing 12-month period of daily closing stock prices when looking for the autocorrelation value and it is found by shifting the data by an increasing number of samples and finding the correlation between the original and shifted time series. The process requires data to be continuously shifted (lagged) until the correlation between the original and shifted series is equal to zero. The number of lags is equal to the autocorrelation period and is used to determine how many samples to use for model creation and signal analysis.

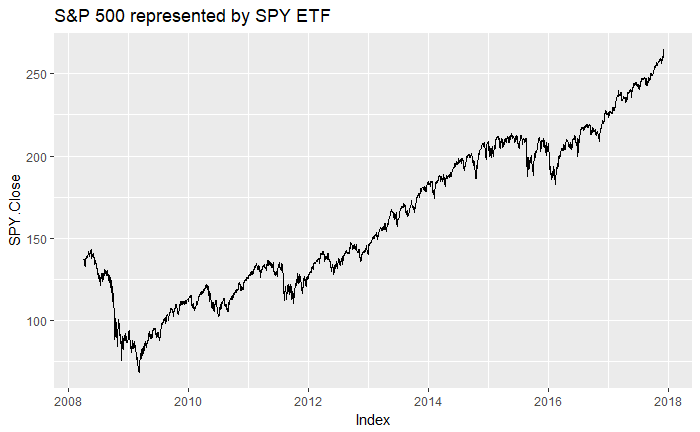
Least-Squares Linear Regression is another common technique that predicts a best fit line for the set of points and then calculates the average square distance from that line to each point. This is to make sure that the line properly represents the entire data series and not just parts of it. The process is repeated until the line with the lowest average square distance from the data is found to find the line with the least error. For this project, the Least-Squares Linear Regression is used to determine the most basic trend of a stock signal, which often represents the direction towards which a stock is headed and correlates to the overall trend of success of the company.

Autoregressive Integrated Moving Average (ARIMA) models have been explored for time series prediction across many industries. Box and Jenkins in 1970 introduced the ARIMA model and it has proven to be prominent and efficiently capable in generating short-term forecasts. The model coefficients are constrained in a manner that cause forecasts to converge to the original time series mean as time progresses; long-term forecasts likely won’t be accurate. Auto Regression (AR) – In auto-regression the values of a given time series data are regressed on their own lagged values, which is indicated by the “p” value in the model. Differencing (I-for Integrated) – This involves differencing the time series data to remove the trend and convert a non-stationary time series to a stationary one. This is indicated by the “d” value in the model. If d = 1, it looks at the difference between two-time series entries, if d = 2 it looks at the differences of the differences obtained at d =1, and so forth. Moving Average (MA) – The moving average nature of the model is represented by the “q” value which is the number of lagged values of the error term. Testing for stationarity – We test for stationarity using the Augmented Dickey-Fuller unit root test. The p-value resulting from the ADF test must be less than 0.05 for a time series to be stationary. If the p-value is greater than 0.05 or 5%, you conclude that the time series has a unit root which means that it is a non-stationary process. Differencing – To convert a non-stationary process to a stationary process, we apply the differencing method. Differencing a time series means finding the differences between consecutive values of a time series data. The differenced values form a new time series dataset which can be tested to uncover new correlations or other interesting statistical properties. We can apply the differencing method consecutively more than once, giving rise to the “first differences”, “second order differences”, etc. We apply the appropriate differencing order (d) to make a time series stationary before we can proceed to the next step. Identification of p and q In this step, we identify the appropriate order of Autoregressive (AR) and Moving average (MA) processes by using the Autocorrelation function (ACF) and Partial Autocorrelation function (PACF). For AR models, the ACF will dampen exponentially and the PACF will be used to identify the order (p) of the AR model. If we have one significant spike at lag 1 on the PACF, then we have an AR model of the order 1, i.e. AR(1). If we have significant spikes at lag 1, 2, and 3 on the PACF, then we have an AR model of the order 3, i.e. AR(3). Identifying the q order of MA model For MA models, the PACF will dampen exponentially and the ACF plot will be used to identify the order of the MA process. If we have one significant spike at lag 1 on the ACF, then we have an MA model of the order 1, i.e. MA(1). If we have significant spikes at lag 1, 2, and 3 on the ACF, then we have an MA model of the order 3, i.e. MA(3). Estimation and Forecasting - Once we have determined the parameters (p,d,q) we estimate the accuracy of the ARIMA model on a training data set and then use the fitted model to forecast the values of the test data set using a forecasting function. In the end, we cross check whether our forecasted values are in line with the actual values.

Technical Analysis indicators are introduced not for forecasting as standalone indicators but more as a representation of price movement. Several indicators are considered to measure volatility, volume, momentum or mean reversion. Volatility of stock prices, measured by standard deviation, can be used to determine the magnitude of the price movement where stocks with high volatility have a higher risk, while stocks with low volatility are less likely to deviate from their mean in the short term, hence considered lower risk. A high volatility stock can potentially offer larger returns with the possibility of larger losses as investors expect compensation for taking on larger risks. On the contrary lower volatility stocks may not have a high expected return earn, but the possibility of losses is considerably lower. There have been attempts to create different metrics to normalize volatility and one particularly interesting approach divides the standard deviation of a stock by its mean to more accurately compare across stocks.

Volume has historically been an important measure when analyzing stock prices as price moves with stronger volume has been considered more reliable than higher prices with declining volume. With the introduction of dark pools and non-exchange transactions, volume has become somewhat of a less important metric, however the measurement of shares traded per day still indicates the popularity of a stock. Some less popular, lower volume stocks could offer potentially larger gains because of the risk/reward tradeoff as some of the more popular names may offer less expected returns. This doesn’t appear to be the case in recent times as the popular FAANG (Facebook, Apple, Amazon, Netflix, Google) have outperformed most other smaller stocks.

Figure 1.



SPY is an ETF tracking the S&P 500 Index and has a value approximately 1/10. Figure 1 shows the upward trend between 04-01-2008 and 11-30-2017. The prevailing theory suggests that it is very difficult to make money picking individual stocks and a passive strategy, i.e. buying the index is a much safer strategy. A cursory glance may support that theory; however, it is difficult to interpret the fluctuations in daily prices. Furthermore, the first iteration attempts a linear regression to identify independent variables that drive the index and compare which variables potentially drive individual stock performance. We will explore a shorter-term window in the next iteration using ARIMA on individual stocks but for the regression, quarterly data points back to Q1 2008 were gathered for the analysis.

When identifying independent variables that drive SP500 performance, it is better to start with many variables and perform a Principal Components Analysis (PCA) to select the most relevant. For this exercise, ten independent variables for the U.S. economy were selected to explain SP500 growth: GDP, Unemployment, Durable Goods, Capacity Utilization, Building Permits, CPI, PPI, Consumer Confidence, M2 and Spread (10yr minus 2yr Treasury yield).

Figure 2

|  | **vars** | **n** | **mean** | **sd** | **median** | **trimmed** | **mad** | **min** | **max** | **range** | **skew** | **kurtosis** | **se** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GDP | 1 | 38 | 1.54 | 2.61 | 2.00 | 1.87 | 1.78 | -8.20 | 5.20 | 13.40 | -1.75 | 3.89 | 0.42 |
| Unemployment | 2 | 38 | -0.33 | 6.27 | -1.93 | -1.24 | 3.67 | -8.96 | 19.67 | 28.63 | 1.73 | 2.85 | 1.02 |
| DurableGoods | 3 | 38 | 0.17 | 5.73 | 0.34 | 0.79 | 3.71 | -19.33 | 11.18 | 30.51 | -1.41 | 3.02 | 0.93 |
| CapacityUtilization | 4 | 38 | -0.12 | 1.83 | 0.13 | 0.03 | 0.77 | -6.30 | 3.15 | 9.45 | -1.24 | 2.49 | 0.30 |
| BuildingPermits | 5 | 38 | 1.39 | 12.15 | 1.26 | 2.12 | 8.31 | -32.46 | 26.32 | 58.78 | -0.68 | 1.03 | 1.97 |
| CPI | 6 | 38 | 0.38 | 0.64 | 0.46 | 0.43 | 0.40 | -2.29 | 1.54 | 3.83 | -1.83 | 5.94 | 0.10 |
| PPI | 7 | 38 | 0.28 | 1.68 | 0.64 | 0.45 | 1.02 | -7.12 | 2.61 | 9.72 | -2.13 | 7.37 | 0.27 |
| ConsumerConfidence | 8 | 38 | 0.09 | 0.54 | 0.16 | 0.10 | 0.51 | -1.15 | 1.25 | 2.40 | -0.08 | -0.15 | 0.09 |
| M2 | 9 | 38 | 1.55 | 0.84 | 1.48 | 1.47 | 0.66 | 0.23 | 3.85 | 3.62 | 0.98 | 0.65 | 0.14 |
| Spread | 10 | 38 | -0.07 | 20.01 | -4.15 | -0.90 | 17.05 | -38.83 | 50.60 | 89.43 | 0.51 | -0.37 | 3.25 |
| SP500 | 11 | 38 | 2.04 | 8.11 | 3.28 | 2.63 | 4.96 | -22.56 | 15.22 | 37.78 | -0.90 | 0.82 | 1.32 |

Visualizing each one of the factors helps navigate through many variables, still in this case it is helpful to understand the distribution of each one of the variables. (Figure 3) There appears to be some skew in each but the negative values make sense since we are viewing the percent change for each of the variables. Additionally, density plots (Figure 4) can help better understand these data and look for abnormalities. According to the density plots, it appears GDP, Durable Goods, Capacity Utilization, CPI and PPI are most relatable to SP500 growth, we still need to investigate further. Figure 5 shows that there are several outliers but since these are actual observations we will not impute values for the analysis. Figure 6 shows the SP500 has some positive correlation to all the variables except for unemployment and M2 Money Supply. The negative correlation to Unemployment makes sense but the relationship to M2 seems less intuitive because as the economy expands and markets perform well, one would expect the M2 to expand as well.

Figure 3

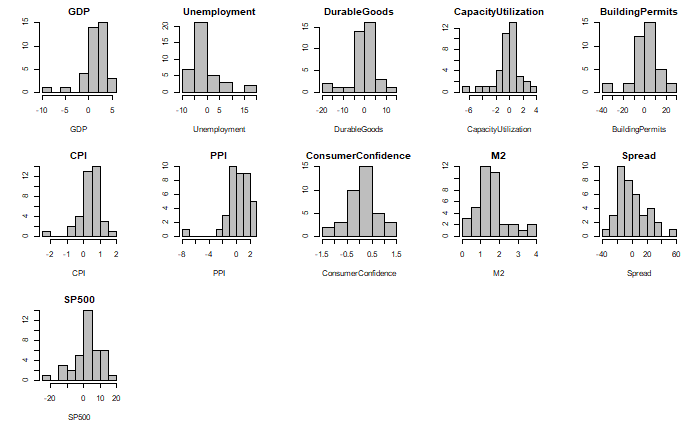


Figure 4

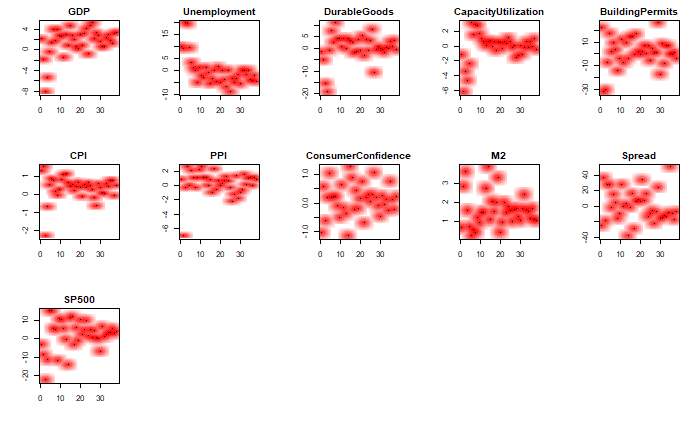


Figure 5

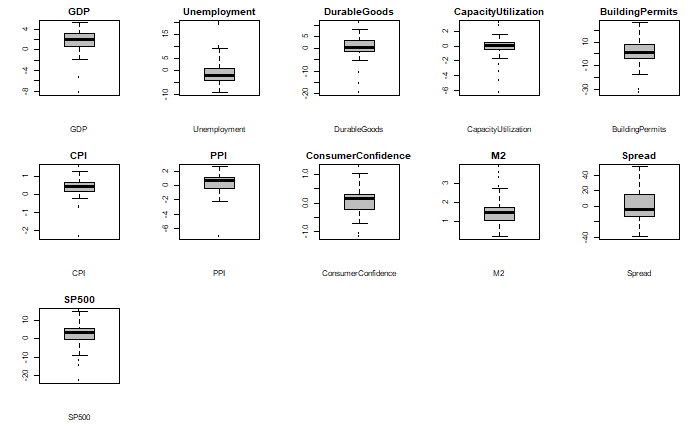
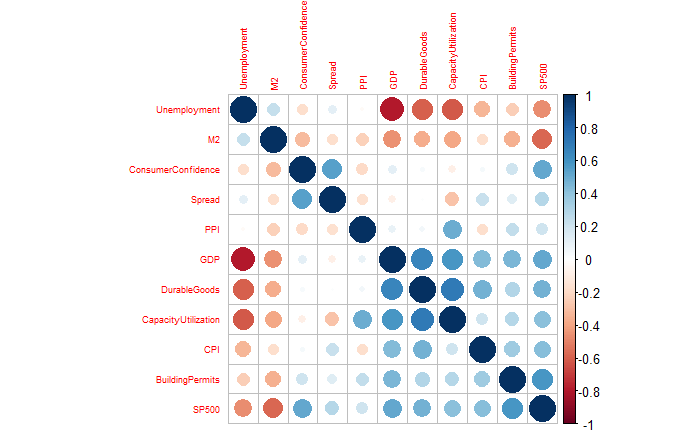
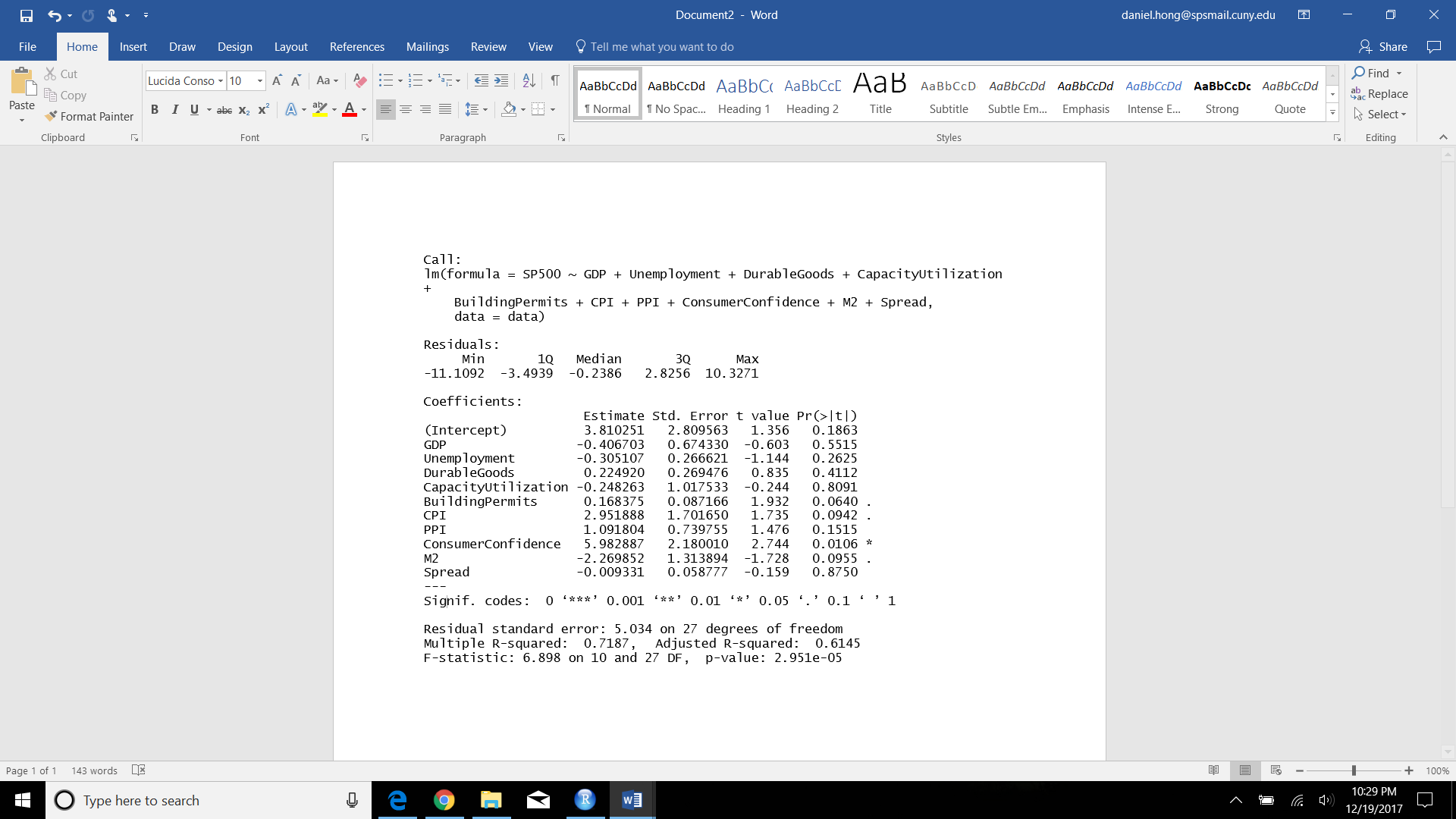


Figure 6



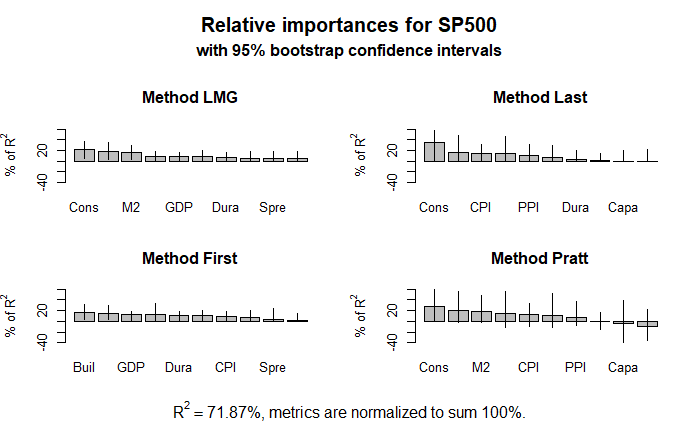
**Multiple Linear Regression**

Figure 7



The model doesn’t appear to have a great fit as only ~72% of the variance can be explained by these variables and the only one with significance is Consumer Confidence. The data are not sufficient to describe the movements in SP500 prices. We can further look at the relative importance of each variable in relation to the SP500 prices:

Figure 8

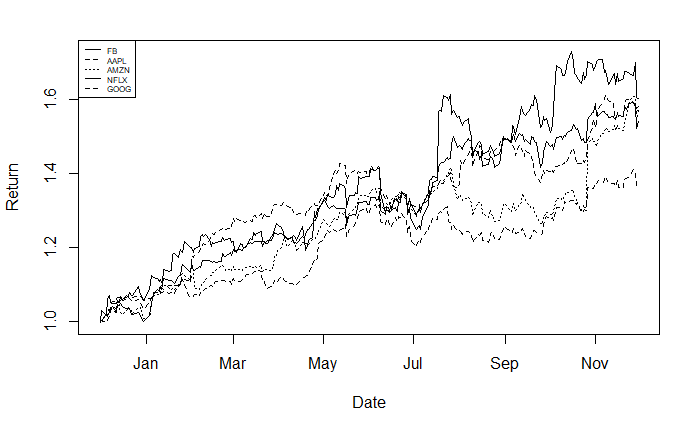


Many of the variables are not significant, p-values above 0.05 and it seems illogical to claim that many closely watched measures of economic performance followed by Wall Street and the Fed has practically no effect on stock prices. It is also strange that some of the coefficients have opposite signs. A possible explanation of these discrepancies might be multicollinearity, which undermines the significance of the individual coefficients. To refine the model, refer to Figure 6 to determine which variables are highly dependent. However, it is important to realize the impossibility of completely removing multicollinearity since all the indicators are related in some way through macroeconomic principles. Although the removal of variables will slightly diminish R2, and hence the predictability of the model, our objective is to find the best combination of the most significant and influential independent indicators in the regression model. Although considerable correlation still exists among certain variables, further removal of variables would prevent a thorough analysis of the influence of these indicators on the stock market.

**Individual Stocks**

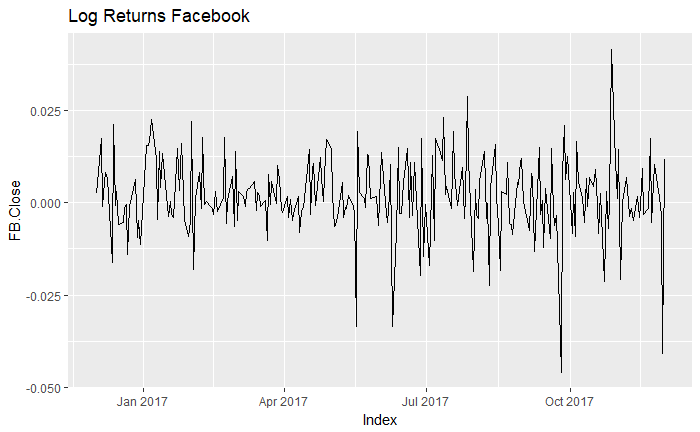
It is important to understand the overall market and which economic indicators could potentially have an impact on these indices, thus individual stocks. It can be argued, however, that the overall market is driven more by economic indicators and individual stocks by other factors so we compensate by applying methods like ARIMA in an attempt to predict individual stock prices. In recent years a lot of attention has been given on stocks that have changed the paradigm in their respective industries. FAANG has become a very popular acronym for Facebook, Apple, Amazon, Netflix and Google. All companies are a part of the S&P 500 index and make up approximately 11% of the market weight of the index, to put things into perspective, the entire tech sector makes up 23% of the index market weight. Despite these staggering numbers, we still believe there are other factors besides the economy that can impact these individual names. The figure below shows the daily return comparison for the FAANG stocks.

Figure 9



We compute the logarithmic returns of the stocks as we want the ARIMA model to forecast the log returns and not the stock price. We will also plot the log return series during the ARIMA forecast. Since these stocks appear to trade very similarly we will examine the best performing (FB) of the group in the November 2017 TTM period.

Figure 10



We call the Augmented Dickey Fuller (ADF) test on the returns series data to check for stationarity. The p-value of 0.01 from the ADF test tells us that the series is stationary. If the series were to be non-stationary, we would have first differenced the returns series to make it stationary.

Figure 11

Augmented Dickey-Fuller Test

data: stock

Dickey-Fuller = -5.7208, Lag order = 6, p-value = 0.01

alternative hypothesis: stationary

Fixed breakpoint will be used to split the returns dataset in two parts (later) and truncate the original returns series up to the breakpoint. Call the ACF and PACF functions on this truncated series.

Figure 12

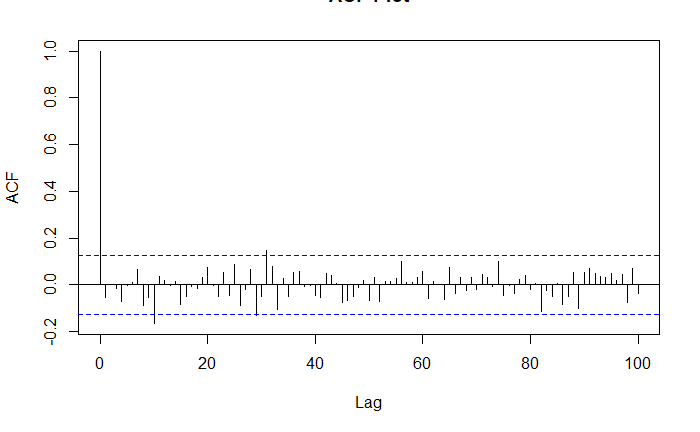
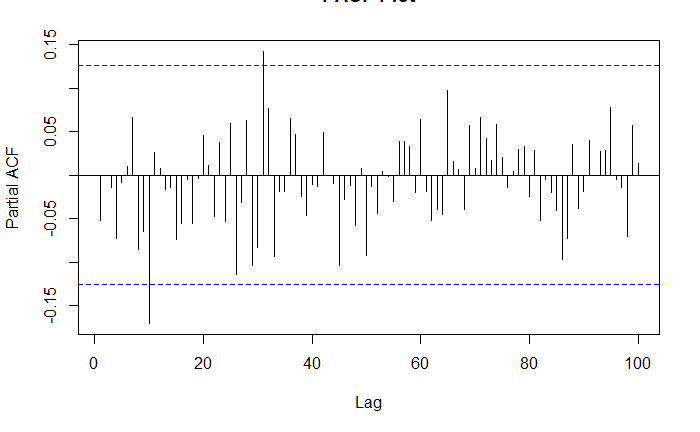
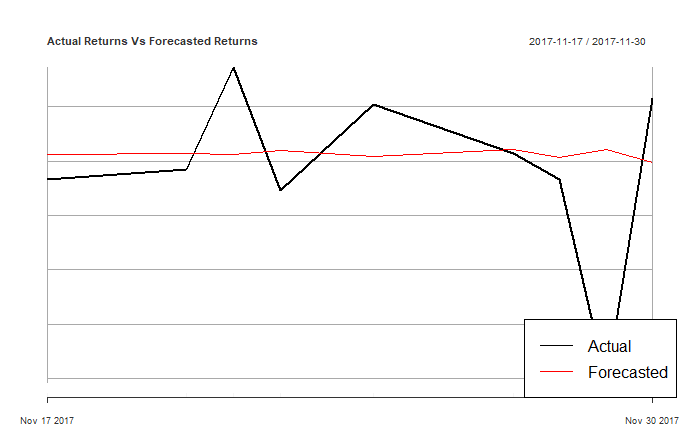


Figure 13



In the Research section we discussed in detail, autocorrelation and different parts of ARIMA. We can observe these plots and arrive at the Autoregressive (AR) order and Moving Average (MA) order. The ACF will dampen exponentially for AR models and the PACF plot will be used to identify the order (p) of the AR model. These plots suggest AR order = 2 and MA order = 2. The ARIMA parameters will be (2,0,2). It doesn’t appear like the forecast was very good:

Figure 14

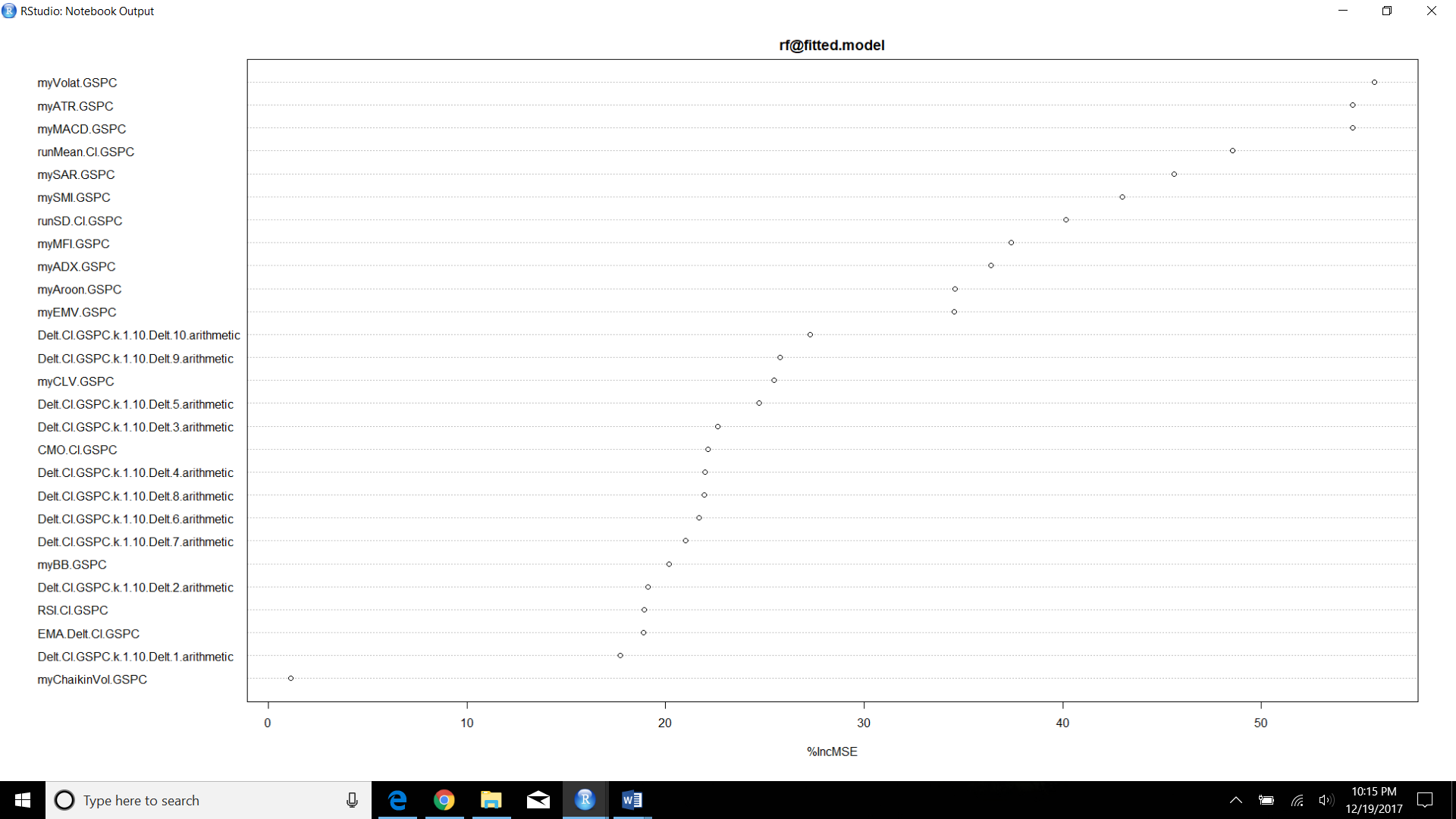


The forecast was only 1/3 accurate, which is bad and could reflect the volatility of these type of stocks. It’s possible that the data collection period (TTM) is not enough data so in the second iteration we will look at the whole history of FB stock data and rerun the analysis. The accuracy improves to ~49% with the longer history but it is still not good enough. For the third iteration, we will look at the SPY, the same ETF we looked at with the linear regression and see there is any improvement. The third iteration gives the best results, ~55% accuracy but that still not much better than a flip of a coin. We investigate an alternative strategy to see if we can get better results.

**Efficient or Inefficient Markets**

After running three ARIMA iterations, it does not appear, assuming the analysis was done correctly, that ARIMA can effectively predict stock prices suggesting that markets are efficient; they adapt so efficiently to price adjustments that there is no systematic way to obtain profits. While there is still a strong following of the efficient markets hypothesis, there is sufficient evidence supporting the theory that there are opportunities facilitated by short term inefficiencies, which over the long term should correct itself as market participants converge. The third approach looks at technical analysis indicators as predictors because they capture dynamics of a price time series (stock prices in this project), despite the debatable effectiveness of technical stock analysis in general. Various indicators were selected because of their unique, descriptive abilities: Average True Range (ATR) which is a volatility measure; Stochastic Momentum Index (SMI) which measures momentum; Welles Wilder’s Directional Movement Index (ADX); Aroon indicator that attempts to identify starting trends; Bollinger Bands which compare volatility over a period of time; the Chaikin Volatility; the Close Location Value (CLV) that relates the session Close to its trading range; the Arms’ Ease of Movement Value (EMV); the Moving Average Convergence Divergence (MACD) oscillator; the Money Flow Index (MFI); the Parabolic Stop-and-Reverse; and the Volatility indicator. These indicators along with a host of others measure price movements and gives the analyst readings on momentum and potential reversion to the mean. While technical analysis uses these indicators to trade, the purpose of this exercise is to use random forests to estimate the importance of these indicators.

Figure 15



Although Artificial neural networks (ANN) are often used in financial forecasting because of the ability to deal with highly non-linear problems, it does not appear to do a good job here as the precision/recall is pretty low:

precision recall

s 0.2809917 0.1931818

b 0.3108108 0.2857143

s+b 0.2973978 0.2373887

Support Vector Machine (SVM) looks like it does slightly better than ANN, but it’s unclear if this is a good sign in absolute terms, considering a still low precision:

precision recall

s 0.375 0.017045455

b NaN 0.000000000

s+b 0.375 0.008902077

Multivariate Adaptive Regression Splines (MARS) appears to do as equally bad as ANN with similar precision scores:

precision recall

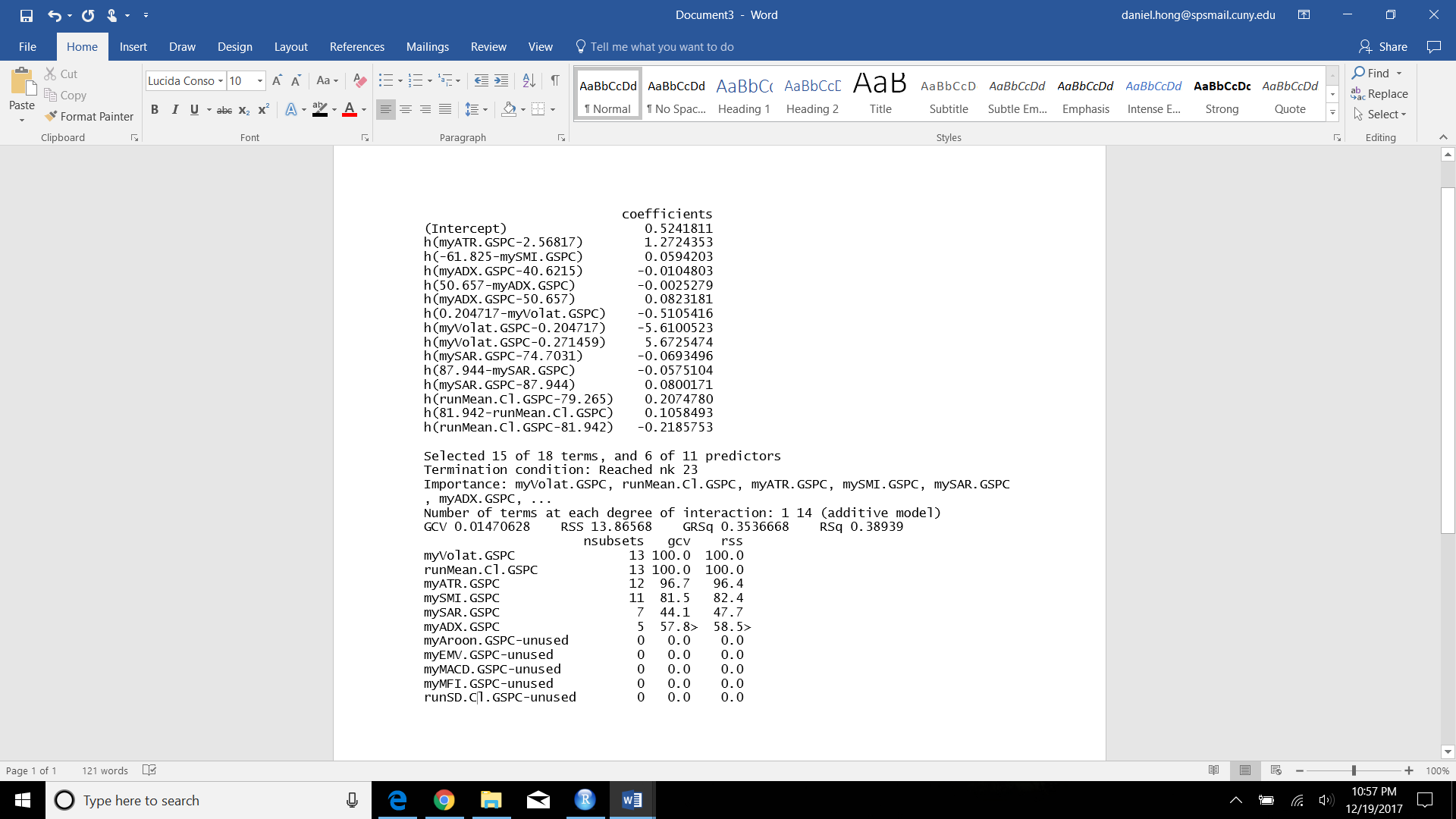
s 0.2894737 0.2500000

b 0.3504274 0.2546584

s+b 0.3159851 0.2522255

None of the models look particularly good but if we had to select one, we would choose SVM based on the 0.375 precision. We investigate the coefficients of the selected indicators and identify 6 predictors below:

Figure 16



**Additional Work**

There are many opportunities for additional research starting with identifying all the indicators that could potentially explain variability in prices. Only a narrow selection of economic and technical analysis indicators were looked at in this research. With hundreds of indicators out there and the advanced computing methods available, many more if not all the indicators should be investigated. Only a handful of stocks and one index were investigated. The S&P 500 is arguably the most widely followed index, but there are so many available, globally, for further analysis. Additionally, many other individual stocks across many industries should also be looked at to confirm or refute the ineffectiveness of some of the models employed. Combining these analyses across a broad range of stocks and indexes, the goal should be to construct a portfolio that can consistently outperform a benchmark.

**References**

Torgo, L., Data Mining With. R

Kuhn, M. and Johnson, K, Applied Predictive Modeling

Miller, C., An Introduction to Stock Market Data Analysis with R (Part 1)

Nielsen, M., Neural Networks and Deep Learning

Dancho, M., Quantitative Stock Analysis Tutorial: Screening the Returns for Every S&P500 Stock in Less than 5 Minutes

Paradkar, M., Forecasting Stock Returns using ARIMA model

Krollner, B. and Vanstone, B “Financial time series forecasting with machine learning techniques: A survey,” in European Symposium on Artiﬁcial Neural Networks: Computational and Machine Learning, Bruges, Belgium

Brownlee, J. A tour of machine learning algorithms.

M. Halls-Moore, “Support vector machines: A guide for beginners,”

S. R. Y. Mayankkumar & B Patel, “Stock prediction using artiﬁcial neural network,” International Journal of Innovative Research in Science, Engineering, and Technology

J. Zhang, “Applying time series analysis builds stock price forecast model,” Modern Applied Science, vol. 3, no. 5

**Appendix**

FAANG Stocks November 30 trailing twelve month (TTM) daily stock prices

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

if (!require("quantmod")) {

install.packages("quantmod")

library(quantmod)

}

start <- as.Date("2016-12-01")

end <- as.Date("2017-11-30")

getSymbols("FB", src = "google", from = start, to = end)

```

```{r}

class(FB)

```

```{r}

autoplot.zoo(FB[, "FB.Close"], main = "Facebook")

```

```{r}

candleChart(FB, up.col = "black", dn.col = "red", theme = "white")

```

```{r}

getSymbols(c("AAPL", "AMZN", "NFLX", "GOOG"), src = "google", from = start, to = end)

```

```{r}

stocks <- as.xts(data.frame(FB = FB[, "FB.Close"],AAPL = AAPL[, "AAPL.Close"], AMZN = AMZN[, "AMZN.Close"],NFLX = NFLX[, "NFLX.Close"], GOOG = GOOG[, "GOOG.Close"]))

head(stocks)

```

```{r}

plot(as.zoo(stocks), screens = 1, lty = 1:3, xlab = "Date", ylab = "Price")

legend("right", c("FB", "AAPL", "AMZN", "NFLX", "GOOG"), lty = 1:3, cex = 0.5)

```

```{r}

plot(as.zoo(stocks[, c("FB.Close","AAPL.Close", "AMZN.Close","NFLX.Close", "GOOG.Close")]), screens = 1, lty = 1:2,

xlab = "Date", ylab = "Price")

par(new = TRUE)

plot(as.zoo(stocks[, "GOOG.Close"]), screens = 1, lty = 3, xaxt = "n", yaxt = "n",

xlab = "", ylab = "")

axis(4)

mtext("Price", side = 4, line = 3)

legend("topleft", c("FB (left)","AAPL (left)", "AMZN (left)", "NFLX (left)", "GOOG"), lty = 1:3, cex = 0.5)

```

```{r}

if (!require("magrittr")) {

install.packages("magrittr")

library(magrittr)

}

```

```{r}

stock\_return = apply(stocks, 1, function(x) {x / stocks[1,]}) %>%

t %>% as.xts

head(stock\_return)

```

```{r}

plot(as.zoo(stock\_return), screens = 1, lty = 1:3, xlab = "Date", ylab = "Return")

legend("topleft", c("FB","AAPL", "AMZN","NFLX", "GOOG"), lty = 1:3, cex = 0.5)

```

Facebook ARIMA

```{r}

library(quantmod)

library(tseries)

library(timeSeries)

library(forecast)

library(xts)

start <- as.Date("1993-01-22")

end <- as.Date("2017-11-30")

getSymbols("SPY", src = "google", from = start, to = end)

stock\_prices = SPY[,4]

stock = diff(log(stock\_prices),lag=1)

stock = stock[!is.na(stock)]

# Plot log returns

autoplot.zoo(stock,type='l', main='Log Returns SPY')

# Conduct ADF test on log returns series

print(adf.test(stock))

# Split the dataset in two parts - training and testing

breakpoint = floor(nrow(stock)\*(2.9/3))

# Apply the ACF and PACF functions

par(mfrow = c(1,1))

acf.stock = acf(stock[c(1:breakpoint),], main='ACF Plot', lag.max=100)

pacf.stock = pacf(stock[c(1:breakpoint),], main='PACF Plot', lag.max=100)

# Initialzing an xts object for Actual log returns

Actual\_series = xts(0,as.Date("2017-01-27","%Y-%m-%d"))

# Initialzing a dataframe for the forecasted return series

forecasted\_series = data.frame(Forecasted = numeric())

for (b in breakpoint:(nrow(stock)-1)) {

stock\_train = stock[1:b, ]

stock\_test = stock[(b+1):nrow(stock), ]

# Summary of the ARIMA model using the determined (p,d,q) parameters

fit = arima(stock\_train, order = c(2, 0, 2),include.mean=FALSE)

summary(fit)

# plotting a acf plot of the residuals

acf(fit$residuals,main="Residuals plot")

# Forecasting the log returns

arima.forecast = forecast(fit, h = 1,level=99)

summary(arima.forecast)

# plotting the forecast

par(mfrow=c(1,1))

plot(arima.forecast, main = "ARIMA Forecast")

# Creating a series of forecasted returns for the forecasted period

forecasted\_series = rbind(forecasted\_series,arima.forecast$mean[1])

colnames(forecasted\_series) = c("Forecasted")

# Creating a series of actual returns for the forecasted period

Actual\_return = stock[(b+1),]

Actual\_series = c(Actual\_series,xts(Actual\_return))

rm(Actual\_return)

print(stock\_prices[(b+1),])

print(stock\_prices[(b+2),])

}

# Adjust the length of the Actual return series

Actual\_series = Actual\_series[-1]

# Create a time series object of the forecasted series

forecasted\_series = xts(forecasted\_series,index(Actual\_series))

# Create a plot of the two return series - Actual versus Forecasted

plot(Actual\_series,type='l',main='Actual Returns Vs Forecasted Returns')

lines(forecasted\_series,lwd=1.5,col='red')

legend('bottomright',c("Actual","Forecasted"),lty=c(1,1),lwd=c(1.5,1.5),col=c('black','red'))

# Create a table for the accuracy of the forecast

comparsion = merge(Actual\_series,forecasted\_series)

comparsion$Accuracy = sign(comparsion$Actual\_series)==sign(comparsion$Forecasted)

print(comparsion)

# Compute the accuracy percentage metric

Accuracy\_percentage = sum(comparsion$Accuracy == 1)\*100/length(comparsion$Accuracy)

print(Accuracy\_percentage)

```

SP500

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

# Get quantmod

if (!require("quantmod")) {

install.packages("quantmod")

library(quantmod)

}

library(ggplot2)

library(quantmod)

start <- as.Date("2008-04-01")

end <- as.Date("2017-11-30")

getSymbols("SPY", src = "google", from = start, to = end)

autoplot.zoo(SPY[, "SPY.Close"], main = "S&P 500 represented by SPY ETF")

```

Linear Regression

```{r}

fit <- lm(SP500 ~ GDP+Unemployment+DurableGoods+CapacityUtilization+BuildingPermits+CPI+PPI+ConsumerConfidence

+M2+Spread, data=data)

summary(fit)

coefficients(fit)

confint(fit, level=0.95) # CIs for model parameters

fitted(fit) # predicted values

residuals(fit) # residuals

anova(fit) # anova table

vcov(fit) # covariance matrix for model parameters

influence(fit) # regression diagnostics

layout(matrix(c(1,2,3,4),2,2)) # optional 4 graphs/page

plot(fit)

```

Relative importance of all variables

```{r}

# Calculate Relative Importance for Each Predictor

library(relaimpo)

calc.relimp(fit,type=c("lmg","last","first","pratt"),

rela=TRUE)

# Bootstrap Measures of Relative Importance (1000 samples)

boot <- boot.relimp(fit, b = 1000, type = c("lmg",

"last", "first", "pratt"), rank = TRUE,

diff = TRUE, rela = TRUE)

booteval.relimp(boot) # print result

plot(booteval.relimp(boot,sort=TRUE)) # plot result

```

Review all the variables we are working with in order to better understand the data they are presenting us. We can see the mean, variation and other metrics within the following table for a quick detailed reference.

```{r}

library(corrgram)

library(caret)

library(psych)

library(knitr)

data<-read.table("C:/Users/danielhong/Documents/Data 698/Econ\_Indicator\_Regression.csv", header=TRUE, sep = ",")

head(data)

colSums(is.na(data))

table.desc <- describe(data[,-1])

table.prep <- as.matrix(table.desc)

table.round <- round((table.prep), 2)

kable(table.round)

```

Visualize each one of the factors, its easier to visually navigate through a large number of variables. We are interested to see how data is distributed for each one of the variables. Please refer to above table for more specific information.

```{r}

dataH <- data[2:ncol(data)] #removing factor var

par(mfrow = c(3,5), cex = .5)

for(i in colnames(dataH)){

hist(dataH[,i], xlab = names(data[i]),

main = names(dataH[i]), col="grey", ylab="")

}

```

Denisty Plot can help better understand these data and look for abnormalities

```{r}

par(mfrow = c(3,5), cex = .5)

for (i in colnames(dataH)) {

smoothScatter(dataH[,i], main = names(dataH[i]), ylab = "",

xlab = "", colramp = colorRampPalette(c("white", "red")))

}

```

Boxplots

```{r}

par(mfrow = c(3,5), cex = .5)

for(i in colnames(dataH)){

boxplot(dataH[,i], xlab = names(dataH[i]),

main = names(dataH[i]), col="grey", ylab="")

}

```

Correlation

```{r}

library(corrplot)

correlations <- cor(dataH)

corrplot(correlations, order = "hclust", tl.cex = 0.55)

```

PLSR

```{r}

knitr::opts\_chunk$set(echo = TRUE)

library(caret)

library(mlbench)

library(MASS)

library(AppliedPredictiveModeling)

library(lars)

library(pls)

library(elasticnet)

library(rpart)

library(e1071)

set.seed(1234)

plsFit = plsr(SP500 ~ ., data=data, validation="CV")

pls.pred = predict(plsFit, data[1:5, ], ncomp=1:2)

pls.pred

validationplot(plsFit, val.type="RMSEP")

validationplot(plsFit, val.type="R2")

pls.RMSEP = RMSEP(plsFit, estimate="CV")

plot(pls.RMSEP, main="RMSEP PLS PH", xlab="Components")

min\_comp = which.min(pls.RMSEP$val)

points(min\_comp, min(pls.RMSEP$val), pch=1, col="red", cex=1.5)

min\_comp

plot(plsFit, ncomp=11, asp=1, line=TRUE)

pls.pred2 = predict(plsFit, data, ncomp=11)

summary(pls.pred2)

```

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

library(xts)

data(GSPC, package="DMwR2")

first(GSPC)

last(GSPC)

library(quantmod)

GSPC <- getSymbols("^GSPC",auto.assign=FALSE)

GSPC <- getSymbols("^GSPC",from="1970-01-02",to="2017-11-30",auto.assign=FALSE)

T.ind <- function(quotes, tgt.margin = 0.025, n.days = 10) {

v <- apply(HLC(quotes), 1, mean)

v[1] <- Cl(quotes)[1]

r <- matrix(NA, ncol = n.days, nrow = NROW(quotes))

for (x in 1:n.days) r[, x] <- Next(Delt(v, k = x), x)

x <- apply(r, 1, function(x) sum(x[x > tgt.margin | x < -tgt.margin]))

if (is.xts(quotes)) xts(x, time(quotes)) else x

}

candleChart(last(GSPC,'3 months'),theme='white', TA=NULL)

avgPrice <- function(p) apply(HLC(p),1,mean)

addAvgPrice <- newTA(FUN=avgPrice,col=1,legend='AvgPrice')

addT.ind <- newTA(FUN=T.ind,col='red', legend='tgtRet')

addAvgPrice(on=1)

addT.ind()

```

```{r}

library(TTR)

myATR <- function(x) ATR(HLC(x))[,'atr']

mySMI <- function(x) SMI(HLC(x))[, "SMI"]

myADX <- function(x) ADX(HLC(x))[,'ADX']

myAroon <- function(x) aroon(cbind(Hi(x),Lo(x)))$oscillator

myBB <- function(x) BBands(HLC(x))[, "pctB"]

myChaikinVol <- function(x) Delt(chaikinVolatility(cbind(Hi(x),Lo(x))))[, 1]

myCLV <- function(x) EMA(CLV(HLC(x)))[, 1]

myEMV <- function(x) EMV(cbind(Hi(x),Lo(x)),Vo(x))[,2]

myMACD <- function(x) MACD(Cl(x))[,2]

myMFI <- function(x) MFI(HLC(x), Vo(x))

mySAR <- function(x) SAR(cbind(Hi(x),Cl(x))) [,1]

myVolat <- function(x) volatility(OHLC(x),calc="garman")[,1]

library(randomForest)

data.model <- specifyModel(T.ind(GSPC) ~ Delt(Cl(GSPC),k=1:10) +

myATR(GSPC) + mySMI(GSPC) + myADX(GSPC) + myAroon(GSPC) +

myBB(GSPC) + myChaikinVol(GSPC) + myCLV(GSPC) +

CMO(Cl(GSPC)) + EMA(Delt(Cl(GSPC))) + myEMV(GSPC) +

myVolat(GSPC) + myMACD(GSPC) + myMFI(GSPC) + RSI(Cl(GSPC)) +

mySAR(GSPC) + runMean(Cl(GSPC)) + runSD(Cl(GSPC)))

set.seed(1234)

rf <- buildModel(data.model,method='randomForest',

training.per=c("1995-01-01","2005-12-30"),

ntree=1000, importance=TRUE)

varImpPlot(rf@fitted.model, type = 1)

```

```{r}

imp <- importance(rf@fitted.model, type = 1)

rownames(imp)[which(imp > 30)]

```

```{r}

data.model <- specifyModel(T.ind(GSPC) ~ myATR(GSPC) + mySMI(GSPC) + myADX(GSPC) +

myAroon(GSPC) + myEMV(GSPC) + myVolat(GSPC) +

myMACD(GSPC) + myMFI(GSPC) + mySAR(GSPC) +

runMean(Cl(GSPC)) + runSD(Cl(GSPC)))

```

```{r}

library(DMwR2)

## The regression task

Tdata.train <- as.data.frame(modelData(data.model,

data.window=c('1970-01-02','2005-12-30')))

Tdata.eval <- na.omit(as.data.frame(modelData(data.model,

data.window=c('2006-01-01','2016-01-25'))))

Tform <- as.formula('T.ind.GSPC ~ .')

## The classification task

buy.thr <- 0.1

sell.thr <- -0.1

Tdata.trainC <- cbind(Signal=trading.signals(Tdata.train[["T.ind.GSPC"]],

buy.thr,sell.thr),

Tdata.train[,-1])

Tdata.evalC <- cbind(Signal=trading.signals(Tdata.eval[["T.ind.GSPC"]],

buy.thr,sell.thr),

Tdata.eval[,-1])

TformC <- as.formula("Signal ~ .")

set.seed(1234)

library(nnet)

## The first column is the target variable

norm.data <- data.frame(T.ind.GSPC=Tdata.train[[1]],scale(Tdata.train[,-1]))

nn <- nnet(Tform, norm.data[1:1000, ], size = 5, decay = 0.01,

maxit = 1000, linout = TRUE, trace = FALSE)

preds <- predict(nn, norm.data[1001:2000, ])

sigs.nn <- trading.signals(preds,0.1,-0.1)

true.sigs <- trading.signals(Tdata.train[1001:2000, "T.ind.GSPC"], 0.1, -0.1)

sigs.PR(sigs.nn,true.sigs)

set.seed(1234)

library(nnet)

norm.data <- data.frame(Signal=Tdata.trainC$Signal,scale(Tdata.trainC[,-1]))

nn <- nnet(Signal ~ ., norm.data[1:1000, ], size = 10, decay = 0.01,

maxit = 1000, trace = FALSE)

preds <- predict(nn, norm.data[1001:2000, ], type = "class")

sigs.PR(preds, norm.data[1001:2000, 1])

set.seed(1234)

library(e1071)

sv <- svm(Tform, Tdata.train[1:1000, ], gamma = 0.001, cost = 100)

s.preds <- predict(sv, Tdata.train[1001:2000, ])

sigs.svm <- trading.signals(s.preds, 0.1, -0.1)

true.sigs <- trading.signals(Tdata.train[1001:2000, "T.ind.GSPC"], 0.1, -0.1)

sigs.PR(sigs.svm, true.sigs)

library(kernlab)

ksv <- ksvm(Signal ~ ., Tdata.trainC[1:1000, ], C = 10)

ks.preds <- predict(ksv, Tdata.trainC[1001:2000, ])

sigs.PR(ks.preds, Tdata.trainC[1001:2000, 1])

library(earth)

e <- earth(Tform, Tdata.train[1:1000, ])

e.preds <- predict(e, Tdata.train[1001:2000, ])

sigs.e <- trading.signals(e.preds, 0.1, -0.1)

true.sigs <- trading.signals(Tdata.train[1001:2000, "T.ind.GSPC"], 0.1, -0.1)

sigs.PR(sigs.e, true.sigs)

summary(e)

evimp(e, trim=FALSE)

```