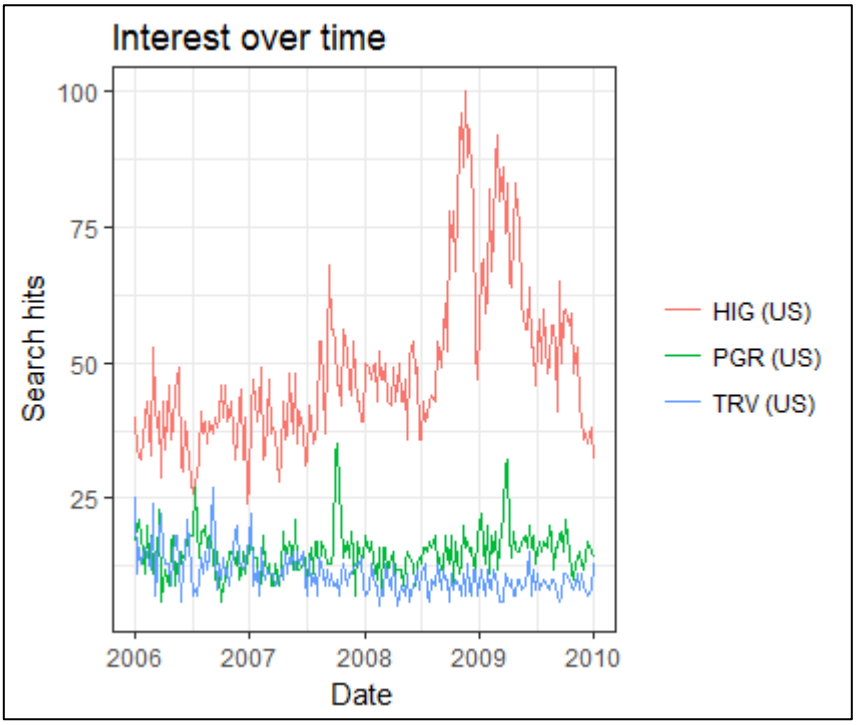


GOOGLING INSURANCE RESULTS

Analyzing the predictive ability of Google trend data on commercial insurance stock performance

ABSTRACT

In this report we analyze the ability of Google search term volume to predict, at various degrees of accuracy, the performance of three, publically traded American commercial insurance stocks. This analysis pulls Google and stock performance data into R using R APIs, combines and cleans that data, and uses multiple statistical analysis methods, including linear regression, logistic regression, regression tree, QDA, LDA, and KNN to support our conclusions. Our resulting prediction models have high accuracy but do not adequately prove our hypothesis. A majority of Google search term data is rendered insignificant in our models and has only minor correlations to our selected stock returns.

Ryne Kitzrow, McCree Lake, Stephen Sloan, Gang Zhu

FE 582, Spring 2017

Table of Contents

Introduction

Trends shape our modern life in numerous ways, ranging from the products that we buy to the entertainment that we choose. Through this project, we explore if trends from Google searches can be used to predict the performance of selected stocks over a period of time.

For the purposes of this analysis, we have narrowed our study to how seven Google trend keywords predict each of three commercial insurance stocks over an eleven-year time period, from January 1st 2006 (the earliest point at which Google trend data was collected) through December 31st 2016. The Google data will then be included in a larger prediction model to predict both the exact weekly returns and weekly movement of those stocks. We will measure the ability of the google trend data to predict the current week's returns, the following week's returns, and the returns two weeks in the future.

Through our methodology, any stock could be measured against particular keywords that we think could be of significance or insignificance to a particular stock. Based on our industry knowledge and expertise, we've chosen insurance companies in order to create our template and analyze our initial results. For our initial analysis, we'll be looking at the stock returns for The Hartford (HIG), Travelers (TRV), and Progressive (PGR). Through Google Trends, we chose keywords that are relevant to the finance world as well as keywords that matched our stock picks to create an initial baseline. These keywords were chosen anecdotally rather than scientifically. We have chosen the negative phrases, "Sell Stocks" and "Bear Market," the positive phrases, "Buy Stocks" and "Bull Market," as well as the neutral ticker abbreviations for HIG, TRV, and PGR. Our objective is to determine if these positive, negative, or neutral keywords/phrases, can be used to predict the weekly returns of our selected insurance stocks.

Our analysis will require substantial data cleaning and consolidation to combine all Google and stock return data into a single data set. Dates and data types will need to be aligned and new columns created for returns one and two weeks in the future as well as for stock return directions. Our analysis will consist of both non classification methods and classification methods. For non classification methods, we will attempt to minimize the mean squared error. For classification methods, we will attempt to maximize the prediction success rate.

Data Sources and Tools

Tools

The primary tools used for this analysis are R version 3.4.0 and R studio version 0.99.903. R and R studio were used to pull all initial data sets, perform data cleaning, and run the primary analyses.

The secondary tool used for this analysis is Microsoft Excel. Excel was used to save modified data sets and reload them back into our analysis independent of the Google trend and insurance stock APIs.

Google Trend Data

Google trend data was imported into R using the "gtrendsR" package version 1.3.5 and the "gtrends" function. "gtrendsR" is an API that pulls Google trend data (keyword hit scores) from Google into R, based on certain parameters such as google keywords, geography, keyword category, google product line, start date, and end date (Package 'gtrendsR', 2016).

The keyword hit data used in this analysis is a normalized search hit value created by dividing the number of search hits obtained for the specified geography by the maximum number of hits obtained over the specified period (How Trends Data is Adjusted, 2017).

Our analysis looks at normalized search hits for the keywords and phrases: "Buy Stocks," "Sell Stocks," "Bull Market," "Bear Market," "HIG," "TRV," and "PGR". These weekly normalized search hits are for the United States and encompass eleven years of weekly data from 01-01-2006 to 12-31-2016 (Google).

Stock Performance Data

Stock performance data was imported into R using the "stockPortfolio" package version 1.2 and the "getReturns" function. The "stockPortfolio" package is an API that pulls stock performance data from Yahoo Finance into R, based on ticker, frequency, start dates, and end dates (Package 'stockPortfolio', 2015).

Or analysis pulls the weekly stock performance data for the Hartford Insurance Group (HIG), Traveler's Insurance (TRV), and Progressive Insurance (PGR) from 01-01-2006 to 12-31-2016. The weekly performance is provided in terms of a % return for that week.

Methodology

Google Trend Data Selection

We decided to limit our Google trend pull to hits from keywords and phrases that might influence the insurance stocks we were analyzing. Our final list includes two phrases generally associated with good stock performance ("Buy Stocks" and "Bull Market"), two phrases generally associated with poor stock performance ("Sell Stocks" and "Bear Market"), and, for each of the companies analyzed, the ticker abbreviations (HIG, PGR, and TRV).

The hypothesis behind the positive market indicator phrases is that users may search for terms such as "Buy Stocks" or "Bull Market" prior to purchasing stocks. Increased searches may then correlate with an increase in stock price.

Similarly, the hypothesis behind negative market indicator phrases is that users may search for the terms "Sell Stocks" or "Bear Market" before selling stocks. In that scenario, those good search terms may foretell a decrease in stock performance.

Separately, we hypothesized that the searching of a stock ticker abbreviation may be indicative of its upcoming performance, whether that be positive or negative. For example, users may search for the stock abbreviation TRV prior to purchasing the stock (to possibly confirm their desire) or prior to selling the stock (to possibly review the latest stock price and/or news about the company).

Stock Selection

During our proposal phase, we looked into analyzing a range of stocks and stock indexes. Ultimately, we decided to maximize our analysis by performing a greater range of analytical techniques on a select few stocks. Stocks chosen were HIG, TRV, and PGR.

HIG, TRV, and PGR were chosen because they are all tickers for commercial insurance companies that were publicly traded during the entire 10 year period of our analysis. The commercial insurance industry is significant because we did not want to add industry variance into our analysis (variance that some industry stock returns may respond differently to Google trend data). We focused our analysis on commercial insurance stocks so that our analysis would fit into the class's financial services framework. The 10 years of publicly traded data is important as well. Since all three stocks have the same data points, we will be able to make comparisons between stock performance and gtrend data for the entire 10 year time frame.

Monday Adjustments

We first created a function that leveraged R's data/time functionality to find the next Monday following a date that was entered. This is accomplished by converting the data to a numeric form recognized by R and making the time adjustments to calculate the next day. We then execute the function against the data containing the dates of the gTrends search results which start the week on Sunday. This converts them to a standard Monday start to align with stock data for clearer comparisons.

Use of 1 and 2 Week Lag Returns

In our raw analysis, each given week has a Google trend hit value (the normalized number of hits a keyword had), plus a stock return value for each stock analyzed. We were concerned that looking at

Google Trend hits and stock returns for the same week might not reveal all correlations, particularly if Google searches generated 1 to 2 weeks prior had an impact on the present day stock price.

To compensate for that concern, we measured the impact of hits of returns for the given week, plus the impact of hits on returns one week later, and the impact of hits on returns two weeks later. Dates and hits are static in our data set. Returns have values of return, return_lag1, and return_lag2.

Variables

27 variables were used in this analysis

Non Keyword or Stock Specific

1. Date (weekly date in YYYY-MM-DD format)
2. geo.x (Marked as US to identify that the Google Trend data is US specific)

Keyword Specific

1. [keyword]_hit (buy, sell, bull, bear, hig, prg, trv showing the weekly normalized hit data for each of those key words or phrases in the US)

Stock Specific

1. [ticker]_return (hig, trv, pgr weekly stock return)
2. [ticker]_r_Lag1 (hig, trv, pgr stock return data from +1 week after the specific hit date)
3. [ticker]_r_Lag2 (hig, trv, pgr stock return data from +2 weeks after the specific hit date)
4. [ticker]_return_d (for hig, trv, pgr tickers, the direction, “Up” or “Down” of the return for that week)
5. [ticker]_r_Lag1.d (hig, trv, pgr tickers, , the direction, “Up” or “Down” of the return for the lag 1 week)
6. [ticker]_r_Lag1.d (hig, trv, pgr tickers, , the direction, “Up” or “Down” of the return for the lag 2 week)

Analytical Techniques

Our goal in this analysis was to test our hypothesis with a range of analytical approaches. This was important because we did not have a prior understanding of how relationships in this data set would be distributed. We found it necessary to plan analysis using both non classification regression and classification techniques. Our application was performed in R using the techniques and processes defined in standard statistical learning in R textbooks (James, Witten, Hastie, & Tibshirani, 2014).

Cross Validation

For all applicable techniques we applied cross validation to prevent overfitting our results. Our data set was divided into two chunks, one for training data and the other for testing data. Predictions were formed against the training data set, and that model was then used to test against our testing data set.

Best Subset Selection

For linear and polynomial analysis, we used best subset selection to determine which selection of dependent variables to use in our measurements. In R, best subset analysis can be achieved using the “regsubsets” function. The output identifies which model order should be used to minimize CP,

minimize BIC, and maximize ADJ. While those three indicators do not always align, we choose the model order represented in 2 of the 3 whenever possible. For example, if a CP was minimized with a model order of 3, BIC minimized in a model order of 4, and BIC maximized in a model order of 3, we would use a model order of 3 for our subsequent analysis.

Linear Regression

Linear regressions were performed using the "lm" function in R. Dependent variables were selected using best subset selection. Predictions were made from the training set and those predictions were tested with the testing/validation set. Mean Squared Error (MSE) was then calculated based off those final results. The objective of linear regression is to minimize the MSE.

Polynomial

Polynomial regressions were performed using the "lm" function in R, along with the "poly" function. Dependent variables were selected using best subset selection. Predictions were made from the training set and those predictions were tested with the testing/validation set. Mean Squared Error (MSE) was then calculated based off those final results for degrees of freedom (df) 1 to 10. The objective of polynomial regression is to choose the df that minimizes the MSE.

Regression Trees (with and without Pruning)

Regression trees were performed using the "tree" function in R. Predictions were made from the training set and those predictions were tested with the testing/validation set. Mean Squared Error (MSE) was then calculated based off those final results.

Additionally, our trees were cross validated and pruned using the "cv.tree" and "prune.tree" functions in R. Cross validation and pruning limits the number of nodes without substantially increasing MSE.

Logistical Regression

Logistical regression was performed using the "glm" function in R. Only significant dependent variables were used in analysis. Predictions were made from the training set and those predictions were tested with the testing/validation set. Success rates were then determined by comparing the predicted class ("Up" indicating that stock would have positive returns or "Down" indicating that the stock would have negative returns) to the actual classes.

Linear Discriminant Analysis (LDA)

Linear discriminant analysis was performed using the "lda" function in R. Only significant dependent variables were used in analysis. Predictions were made from the training set and those predictions were tested with the testing/validation set. Success rates were then determined by comparing the predicted class ("Up" indicating that stock would have positive returns or "Down" indicating that the stock would have negative returns) to the actual classes.

Quadratic Discriminant Analysis (QDA)

Quadratic discriminant analysis was performed using the "qda" function in R. Only significant dependent variables were used in analysis. Predictions were made from the training set and those predictions were tested with the testing/validation set. Success rates were then determined by comparing the predicted class ("Up" indicating that stock would have positive returns or "Down" indicating that the stock would have negative returns) to the actual classes.

K Nearest Neighbor

K nearest neighbors analysis was performed using the “knn” function in R. K values of 1 to 50 were tested. Only significant dependent variables were used in analysis. Predictions were made from the training set and those predictions were tested with the testing/validation set. Success rates were then determined by comparing the predicted class (“Up” indicating that stock would have positive returns or “Down” indicating that the stock would have negative returns) to the actual classes.

Data Clean Up

Google Trend Data

Extracting data from Google Trends

Our first task was to extract Google Trend data from Google using the grends API. Because extractions greater than 4 years are returned as monthly data, we made three extractions of 4, 4, and 3 years to pull weekly data.

Six queries were made in total. The first three were for phrases and the second three were for ticker abbreviations.

```
27  #Set a query
28  market.query = c("Buy Stocks","Sell Stocks","Bull Market","Bear Market")
29  company.query = c("HIG","PGR","TRV")
30
31  #Extract Data for queries
32  trend.market.US.1 = gtrends(market.query, geo="US", time = '2006-01-01 2010-01-01')
33  trend.market.US.2 = gtrends(market.query, geo="US", time = '2010-01-02 2014-01-01')
34  trend.market.US.3 = gtrends(market.query, geo="US", time = '2014-01-02 2016-12-31')
35  trend.company.US.1 = gtrends(company.query, geo="US", time = '2006-01-01 2010-01-01')
36  trend.company.US.2 = gtrends(company.query, geo="US", time = '2010-01-02 2014-01-01')
37  trend.company.US.3 = gtrends(company.query, geo="US", time = '2014-01-02 2016-12-31')
38
```

Identifying and extracting interest of time tables from gtrend results

Several tables are returned from a gtrend query. We were only interested in looking at the hits over time information so we isolated only those results into a new data frame.

```
48  #Separate out trend data
49  market.US.1 = trend.market.US.1$interest_over_time
50  market.US.2 = trend.market.US.2$interest_over_time
51  market.US.3 = trend.market.US.3$interest_over_time
52  company.US.1 = trend.company.US.1$interest_over_time
53  company.US.2 = trend.company.US.2$interest_over_time
54  company.US.3 = trend.company.US.3$interest_over_time
```

Binding time periods into a single data set

Our next step was to rbind the three time periods together into a single data set. Two data sets remained at this point, one for phrase queries, and one for ticker queries.

```
56  #Use rbind to bind all three date ranges together into 1 data set
57  market.US = rbind(market.US.1,market.US.2,market.US.3)
58  company.US = rbind(company.US.1,company.US.2,company.US.3)
```

Turning query phrases and words into columns

Our goal was to analyze all query phrases and tickers by date. Each query needed to be summarized and converted into columns.

```
#Separate out data by keyword
Buy = market.US[market.US$keyword == market.query[1],]
Sell = market.US[market.US$keyword == market.query[2],]
Bull = market.US[market.US$keyword == market.query[3],]
Bear = market.US[market.US$keyword == market.query[4],]

HIG = company.US[company.US$keyword == company.query[1],]
PGR = company.US[company.US$keyword == company.query[2],]
TRV = company.US[company.US$keyword == company.query[3],]

#Rename hits columns in each new file
colnames(Buy)[2] = "buy_hit"
colnames(Sell)[2] = "sell_hit"
colnames(Bull)[2] = "bull_hit"
colnames(Bear)[2] = "bear_hit"
colnames(HIG)[2] = "hig_hit"
colnames(PGR)[2] = "pgr_hit"
colnames(TRV)[2] = "trv_hit"
```

Merging company and market data sets together and reducing columns

The company and market data sets were then combined together. We eliminated all undesired columns for the consolidated data set.

```
#merg all gtrends query data sets together
market.company = Reduce(function(x, y) merge(x, y, by = "date"), list(Buy,Sell,Bull,Bear,HIG,PGR,TRV))
dim(market.company)
names(market.company)
#select only the hit, date and geo columns
market.company = market.company[,c(1,2,4,7,12,17,22,27,32)]
dim(market.company)
names(market.company)
range(market.company$date)
date.min = min(market.company$date)
date.max = max(market.company$date)
```

Monday Clean Ups

Due to holidays, our stock data and gtrend weekly data did not always have the same week ending date. Although this only accounted for a handful of weeks every year, we decided to adjust our week ending dates to ensure that no data points were lost by mismatching week ending dates. All week end dates were moved to the next Monday. This meant that, even if a stock week ended on a Thursday or Friday due to a holiday, the week end date would always be the date of the next Monday.

```
#Update dates in searches to be Monday instead of Sunday
FindMonday <- function(x) 7 * ceiling(as.numeric(x - 1)/7) + as.Date(-3)
market.company$NextMonday = FindMonday(as.Date(market.company$date))
```

Stock Data

Extract stock data for the tickers and dates queried in gtrends

We then pulled stock data from Yahoo finance using the getReturns API. We pulled only the stocks and dates from the gtrends queries.

```
#Pull weekly stock results
#Setup stock data frame
stock.data = data.frame(performance=NA,symbol=NA,week=NA)

for(i in company.query){

  #Get stock data
  returns <- getReturns(i, freq="week", start = date.min, end = date.max)
  stock.data.add = data.frame(returns$R)
  stock.data.add$symbol = i
  stock.data.add$date = rownames(stock.data.add)
  colnames(stock.data.add) = c('performance','symbol','week')
  stock.data = rbind(stock.data,stock.data.add)
  stock.data = na.omit(stock.data)
}
```

Summarize stock ticker data and convert into columns

Similar to conversion done for the gtrends data, we then summarized the stock information by data and converted the ticker information into columns. Each row became a date, and each column stood for the returns for a specific ticker.

```
#Separate out stocks
colnames(stock.data)[3] = "NextMonday"
stock.data$NextMonday = as.Date(stock.data$NextMonday)
stock.data.HIG = stock.data[stock.data$symbol == company.query[1],]
stock.data.PGR = stock.data[stock.data$symbol == company.query[2],]
stock.data.TRV = stock.data[stock.data$symbol == company.query[3],]

#Define distinct performance by stock
colnames(stock.data.HIG)[1] = "hig_return"
colnames(stock.data.PGR)[1] = "pgr_return"
colnames(stock.data.TRV)[1] = "trv_return"
```

Merge gtrend and stock data

Finally, the gtrend and Yahoo finance files were merged and the columns cleaned up.

```
#Consolidate data (not yet modified for multipe days off. rows drop from 209 to 189)
  #1 Day off
master.data = Reduce(function(x, y) merge(x, y, by = "NextMonday"),
                     list(market.company,stock.data.HIG,stock.data.PGR,stock.data.TRV))
head(master.data)
dim(master.data)
names(master.data)

#Clean Up columns
clean.data = master.data[,c(1,3,4,5,6,7,8,9,10,11,13,15)]
colnames(clean.data)[1] = "date"
clean.data = clean.data[order(clean.data$date),]
```

Lag and Direction columns added

Lag columns of 1 week (Lag1) and 2 week (Lag2) were added. Lag1 columns for a row represent the return +1 week later. Lag2 columns for a row represent the return +2 weeks later.

```
#Create return lag columns to show returns 1 and 2 weeks after hits
#In the same row, hits for week 1, returns for weeks 2, and weeks 3.
#520 rows
#1 week lag for hig_return
clean.data$hig_r_Lag1 = NA
clean.data$hig_r_Lag1[1:519]= clean.data$hig_return[2:520]
```

Direction columns were also created to identify the direction (Up or Down) of all raw return data plus each lag return.

```
#direction for 1 week lag fields
direction_Lag1 = function(y){
  z = rep(NA, length(y))
  for(i in 1:length(y)){
    z[i] = if(y[i]>0) {"Up"} else {"Down"}
  }
  return(z)
}

clean.data$hig_r_Lag1.d = as.factor(direction_Lag1(clean.data$hig_r_Lag1))
clean.data$trv_r_Lag1.d = as.factor(direction_Lag1(clean.data$trv_r_Lag1))
clean.data$pgr_r_Lag1.d = as.factor(direction_Lag1(clean.data$pgr_r_Lag1))
```

Results

Descriptive Results on Prediction Accuracy and Success

Our analysis was initially performed using each analysis technique (as identified in the methodology section) for all stocks and keywords, using the stock return, stock return lag1, and stock return lag 2 values as dependent variables.

For both classification and non classification methods, variables with insignificant coefficients were removed. In cases where all variables were deemed insignificant, no MSE or classification success rate are given.

For HIG

MSE

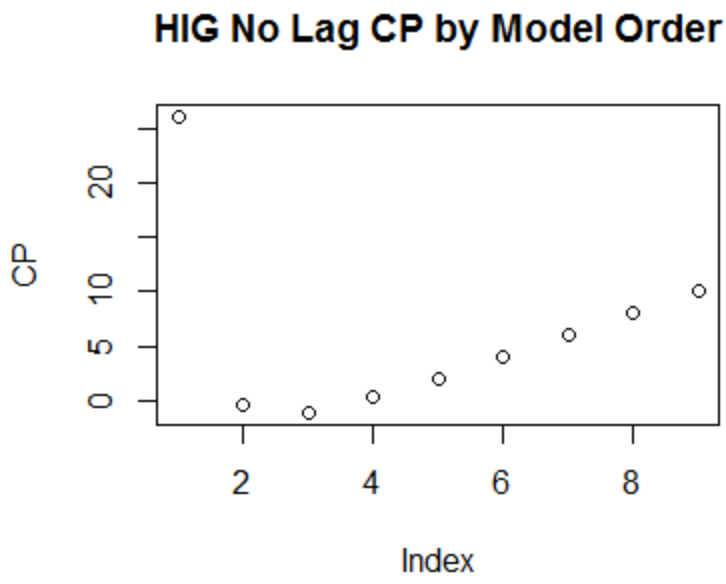
	Hig Linear	Hig Poly	Hig Tree	Hig Prune
No Lag	0.00109	0.00102	0.00228	0.00228
Lag1	0.00628	NA	0.00206	0.00206
Lag2	0.00382	0.00281	0.00480	0.00431

The only regression with no significant coefficients was polynomial regression for Lag1 returns. For No Lag returns, polynomial regression had the lowest MSE. For Lag1 returns, regression tree analysis (pruned and unpruned) had the lowest MSE. For Lag2 returns, polynomial regression had the lowest MSE.

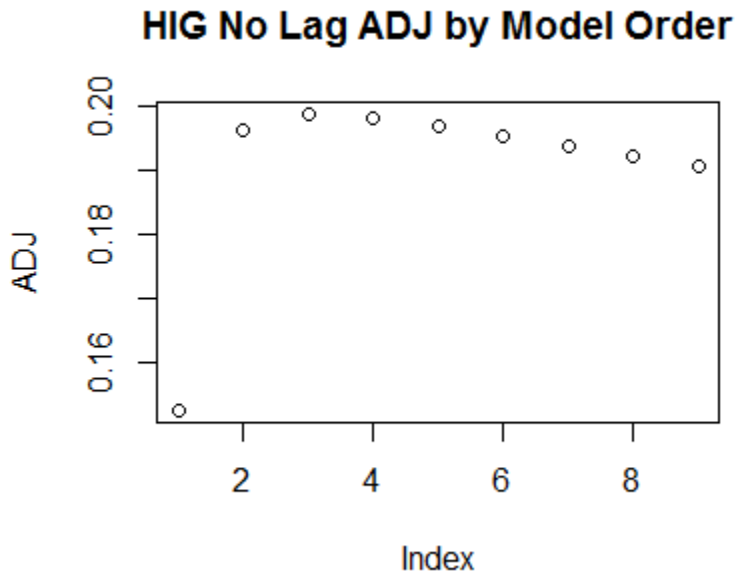
No Lag Best Subset Selection Model Order

	1	2	3	4	5	6	7	8	9
(Intercept)	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
buy_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
sell_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
bull_hit	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
bear_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
hig_hit	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_hit	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_hit	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_return	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_return	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

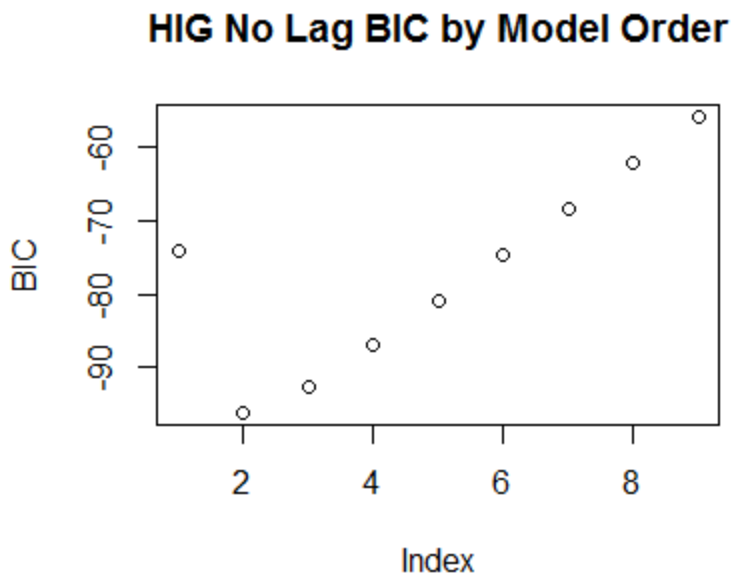
We selected the 3 model order because it minimizes CP and maximizes ADJ. Although it does not minimize BIC, it is the second lowest BIC.



Models	CP	ADJ	BIC
3.0000000	-1.1120548	0.1985755	-92.4678929



Models	CP	ADJ	BIC
3.0000000	-1.1120548	0.1985755	-92.4678929

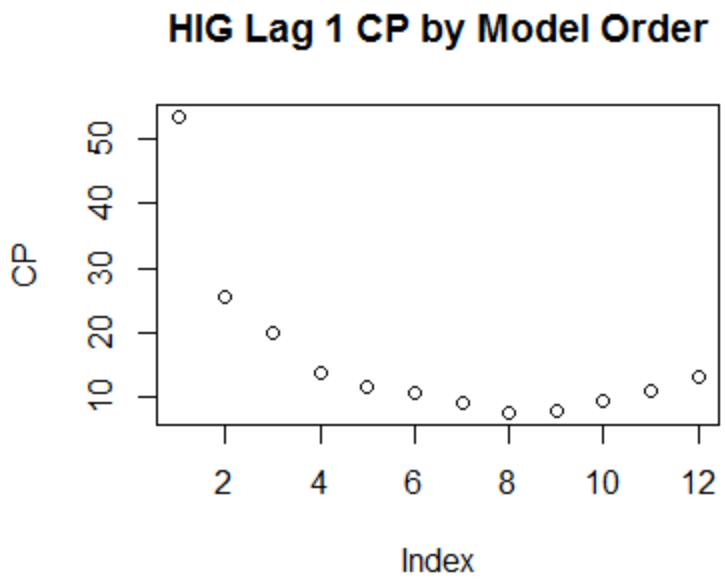


Models	CP	ADJ	BIC
2.0000000	-0.4842787	0.1959963	-96.0479107

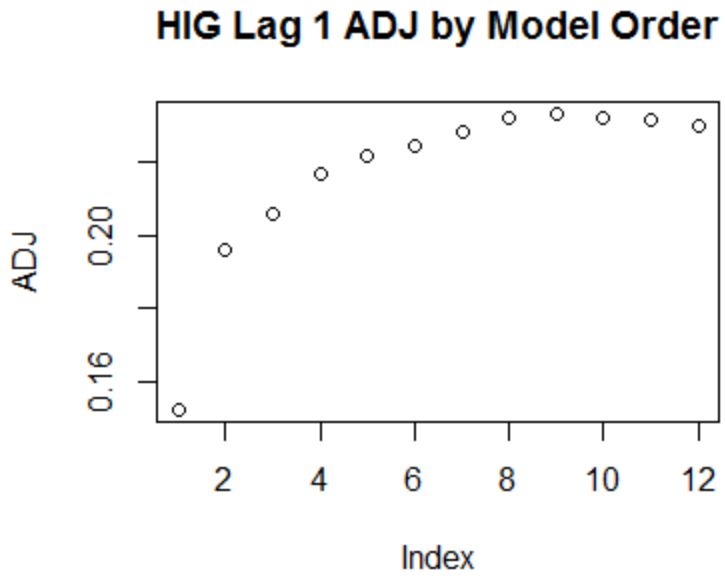
Lag 1 Best Subset Selection Model Order

(Intercept)	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
buy_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
sell_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
bull_hit	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
bear_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
hig_hit	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
trv_return	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_return	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_return	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_r_Lag1	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_r_Lag1	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

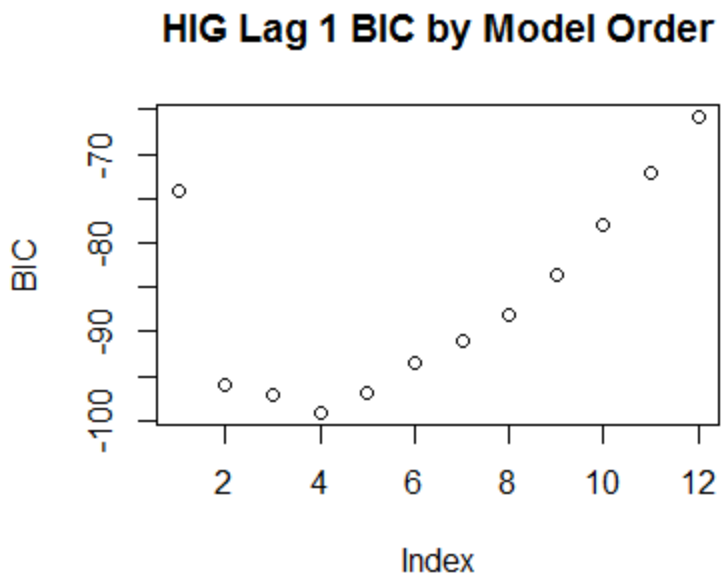
We selected the 8 model order. The 8 model order minimizes CP and comes close to the max ADJ. Because the minimum BIC is from the 4th model order, we chose the middle model order that meets our measurement criteria.



Models	CP	ADJ	BIC
8.0000000	8.0003496	0.1921019	-62.1319889



Models	CP	ADJ	BIC
9.0000000	10.0000000	0.1905089	-55.8843026

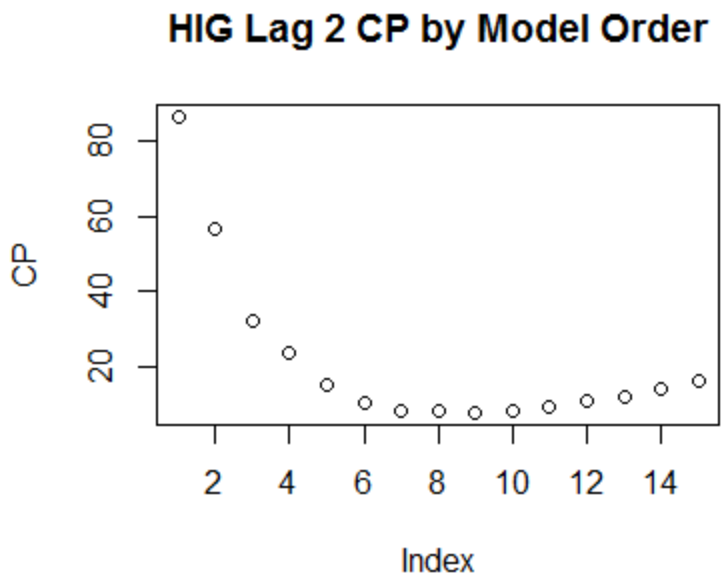


Models	CP	ADJ	BIC
4.0000000	0.2834324	0.1979660	-86.8355749

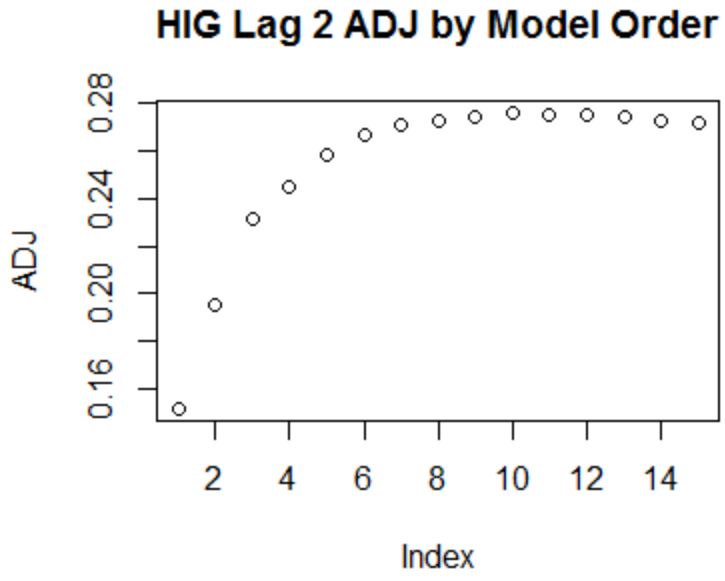
Lag 2 Best Subset Selection Model Order

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(Intercept)	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
buy_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
sell_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
bull_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
bear_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
hig_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_return	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
pgr_return	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_return	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_r_Lag1	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_r_Lag1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_r_Lag1	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_r_Lag2	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_r_Lag2	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

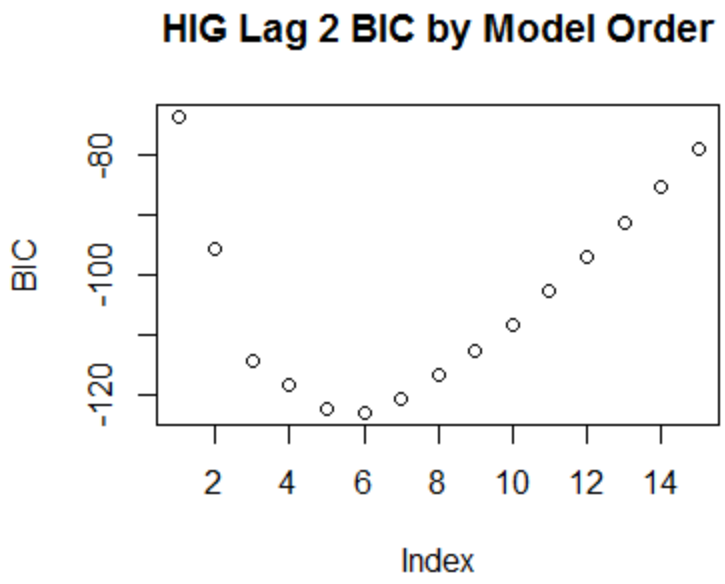
We selected the 9 model order. Similar to Lag 1, the 9 model order minimizes CP and comes close to maximizing ADJ at 10. Since BIC minimizes at the lower 6 model order, we chose the middle model order of 9.



Models	CP	ADJ	BIC
9.0000000	7.8484631	0.2745177	-112.5315239



Models	CP	ADJ	BIC
10.0000000	7.9558319	0.2758091	-108.2253093



Models	CP	ADJ	BIC
6.0000000	10.1898757	0.2668689	-122.8032666

Significant Linear Variables by Return Lag

No Lag: trv_return + pgr_return

Lag 1: hig_hit + pgr_hit + pgr_return + hig_return + pgr_r_Lag1 + trv_r_Lag1

Lag 2: hig_hit + hig_return + pgr_r_Lag1 + hig_r_Lag1 + pgr_r_Lag2

Significant Polynomial Variables and Degrees of Freedom by Return Lag

No Lag: trv_return + pgr_return (4 degrees of freedom)

Lag 1: N/A (No degrees of freedom are significant)

Lag 2: hig_hit + hig_return + pgr_r_Lag1 + hig_r_Lag1 + pgr_r_Lag2 (3 degrees of freedom)

Unpruned Tree Variables and Nodes by Return Lag

No Lag: 7 nodes

Lag 1: 7 nodes

Lag 2: 7 nodes

Pruned Tree Variables and Nodes by Return Lag

No Lag: 7 nodes

Lag 1: 7 nodes

Lag 2: 4 nodes

Prediction Success Rate

	HIG GLM	HIG LDA	HIG QDA	HIG KNN
No Lag	44.02	67.95	75.29	75.68
Lag1	42.08	68.34	69.11	72.97
Lag2	40.54	50.19	69.50	73.75

All prediction classification methods for No Lag, Lag1, and Lag2 returns had significant coefficients. KNN produced the highest success rate for No Lag, Lag1, and Lag2 returns.

Significant GLM Variables by Return Lag (used for all classification methods)

Lag: trv_return + pgr_return

Lag1: pgr_return + hig_return + trv_r_Lag1 + pgr_r_Lag1

Lag 2: pgr_r_Lag1 + hig_r_Lag1 + trv_r_Lag2 + pgr_r_Lag2

For TRV

MSE

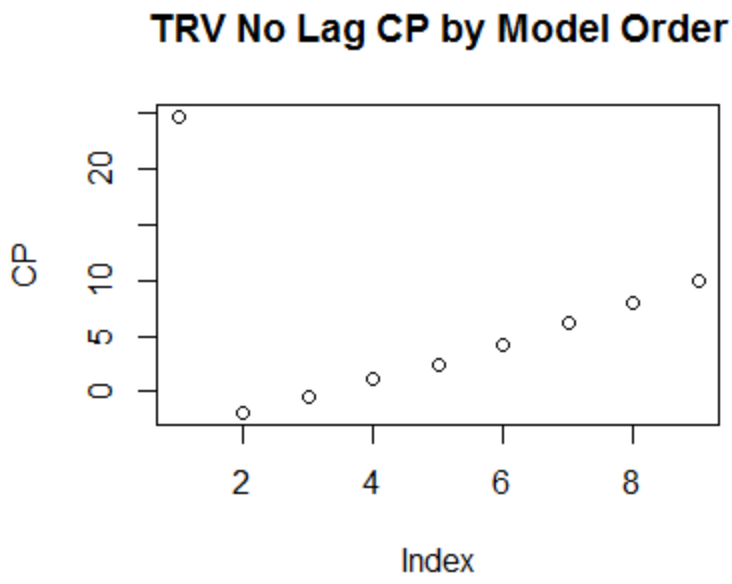
	TRV Linear	TRV Poly	TRV Tree	TRV Prune
No Lag	0.00034	0.00033	0.00047	0.00042
Lag1	0.00076	0.00043	0.00053	0.00053
Lag2	0.00096	0.00059	0.00038	0.00038

For TRV, all non classification prediction methods had significant coefficients. For No Lag and Lag1 returns, polynomial regression had the lowest MSE. For Lag2 returns, regression tree produced the lowest MSE.

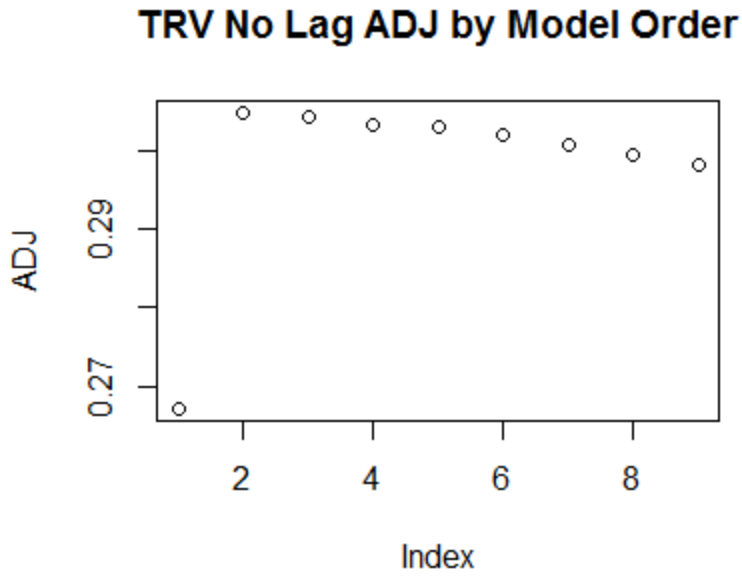
No Lag Best Subset Selection Model Order

	1	2	3	4	5	6	7	8	9
(Intercept)	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
buy_hit	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
sell_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
bull_hit	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
bear_hit	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
pgr_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
trv_hit	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
hig_return	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_return	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

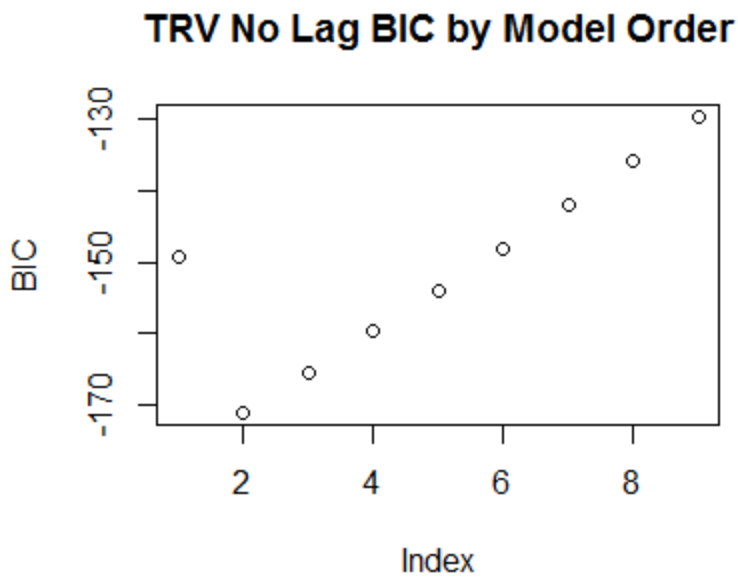
We selected the 2 model order because it minimizes CP and BIC and maximizes ADJ.



Models	CP	ADJ	BIC
2.0000000	56.8144704	0.1951461	-95.5014961



Models	CP	ADJ	BIC
2.0000000	56.8144704	0.1951461	-95.5014961

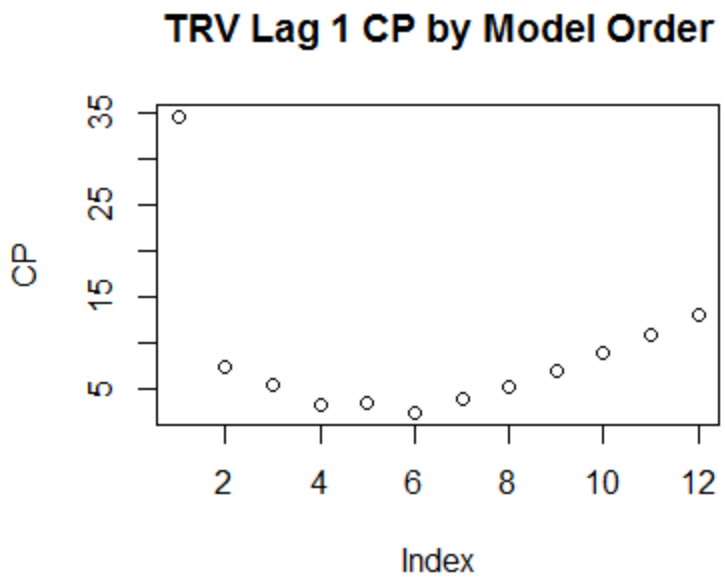


Models	CP	ADJ	BIC
2.0000000	56.8144704	0.1951461	-95.5014961

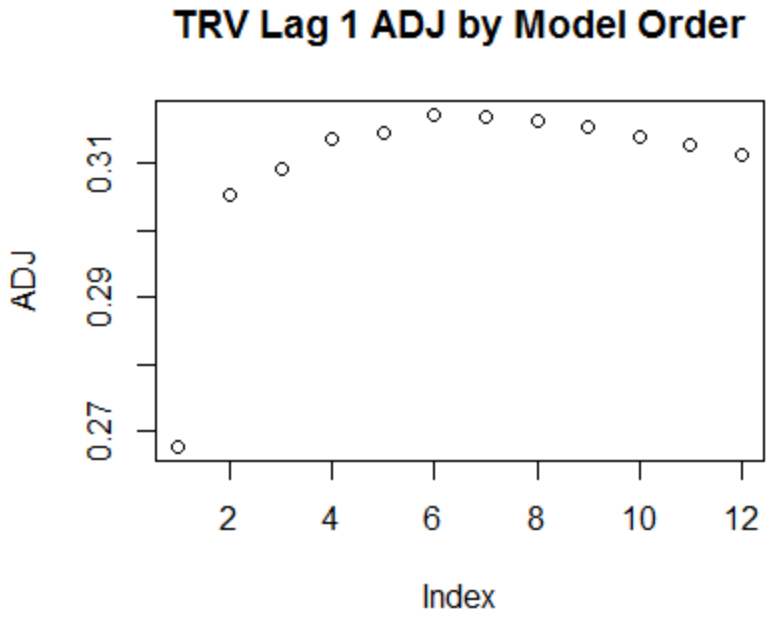
Lag 1 Best Subset Selection Model Order

	1	2	3	4	5	6	7	8	9	10	11	12
(Intercept)	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
buy_hit	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
sell_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
bull_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
bear_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
pgr_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
trv_hit	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_return	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
pgr_return	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_return	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_r_Lag1	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_r_Lag1	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

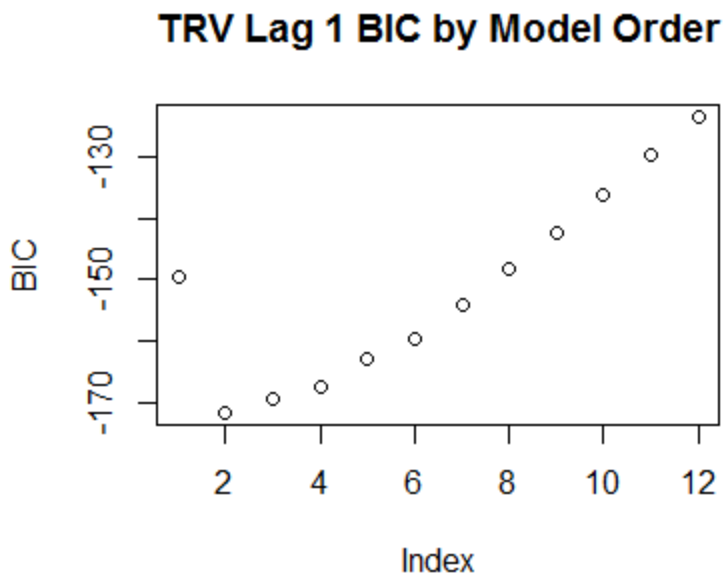
We selected the 6 model order. At 6, CP is minimized and ADJ is maximized.



Models	CP	ADJ	BIC
6.0000000	10.1898757	0.2668689	-122.8032666



Models	CP	ADJ	BIC
6.0000000	10.1898757	0.2668689	-122.8032666

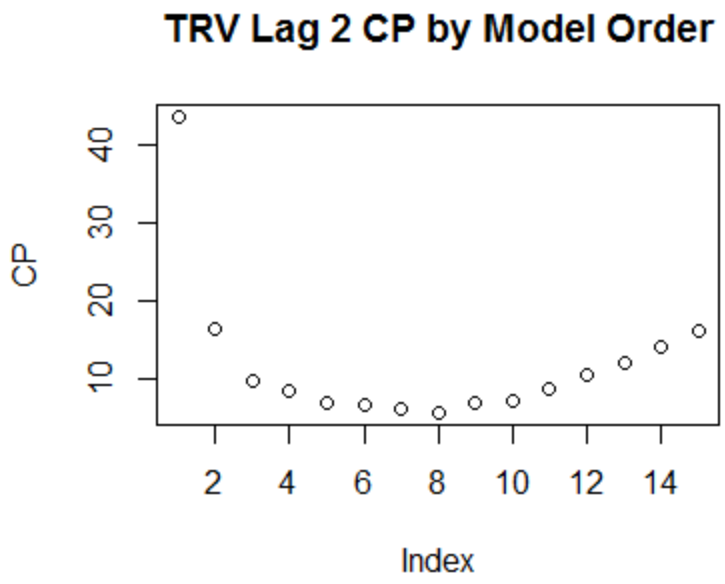


Models	CP	ADJ	BIC
2.0000000	56.8144704	0.1951461	-95.5014961

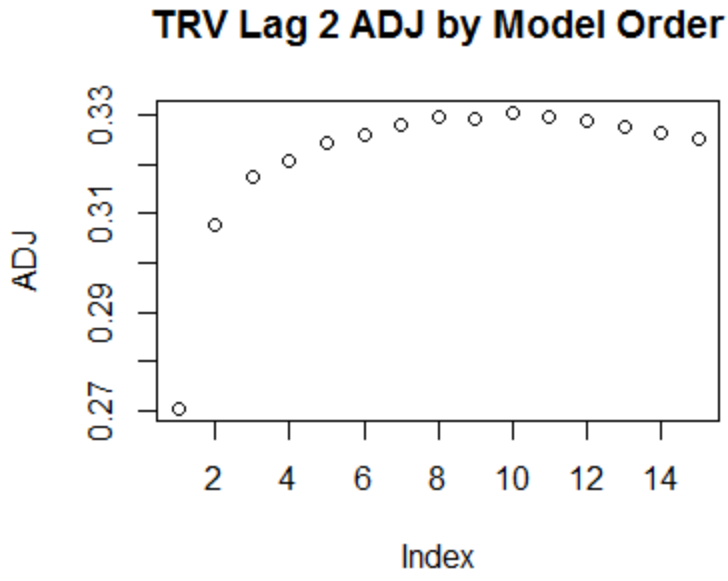
Lag 2 Best Subset Selection Model Order

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(Intercept)	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
buy_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
sell_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
bull_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
bear_hit	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
pgr_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
trv_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
trv_return	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_return	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_return	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_r_Lag1	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_r_Lag1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
hig_r_Lag1	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_r_Lag2	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_r_Lag2	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

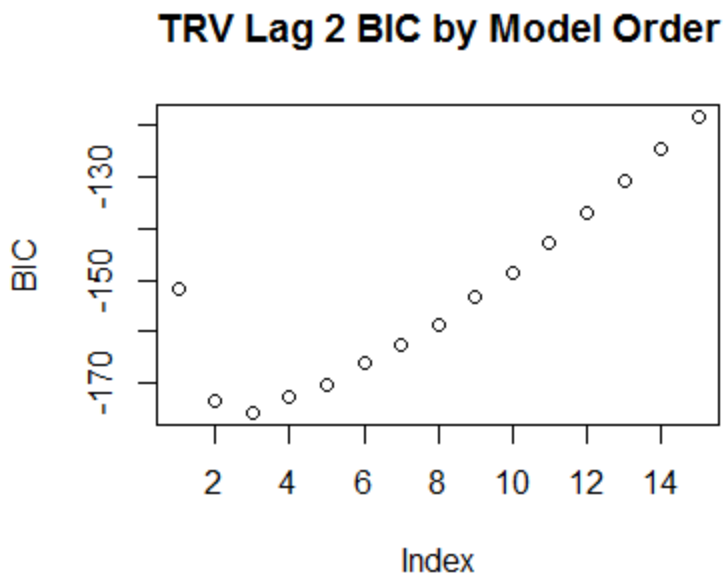
We selected the 6 model order. The 6 model order does not minimize CP (at 8), maximize ADJ (10), or minimize BIC (3), however, 6 is produces favorable results for CP and ADJ while still producing a low BIC.



Models	CP	ADJ	BIC
8.000000	5.592062	0.329626	-158.604309



Models	CP	ADJ	BIC
10.0000000	7.1009776	0.3302989	-148.6668849



Models	CP	ADJ	BIC
3.0000000	9.7503467	0.3175333	-175.5378748

Significant Linear Variables by Return Lag

No Lag: hig_return + pgr_return

Lag 1: pgr_return + hig_return + pgr_r_Lag1 + hig_r_Lag1

Lag 2: trv_return + pgr_r_Lag2

Significant Polynomial Variables and Degrees of Freedom by Return Lag

No Lag: hig_return + pgr_return (9 degrees of freedom)

Lag 1: pgr_return + hig_return + pgr_r_Lag1 + hig_r_Lag1 (7 degrees of freedom)

Lag 2: trv_return + pgr_r_Lag2 (1 degree of freedom)

Unpruned Tree Variables and Nodes by Return Lag

No Lag: 8 nodes

Lag 1: 8 nodes

Lag 2: 9 nodes

Pruned Tree Variables and Nodes by Return Lag

No Lag: 4 nodes

Lag 1: 8 nodes

Lag 2: 9 nodes

Prediction Success Rate

	TRV GLM	TRV LDA	TRV QDA	TRV KNN
No Lag	45.17	75.29	75.68	78.76
Lag1	45.56	76.45	76.83	77.99
Lag2	44.79	50.97	57.92	77.22

For TRV, all classification methods produced significant coefficients. KNN produced the highest prediction success rate for No Lag, Lag1, and Lag2 returns.

Significant GLM Variables by Return Lag (used for all classification methods)

Lag: hig_return + pgr_return

Lag1: hig_r_Lag1 + pgr_r_Lag1

Lag 2: hig_return + hig_r_Lag2 + pgr_r_Lag2

For PGR

MSE

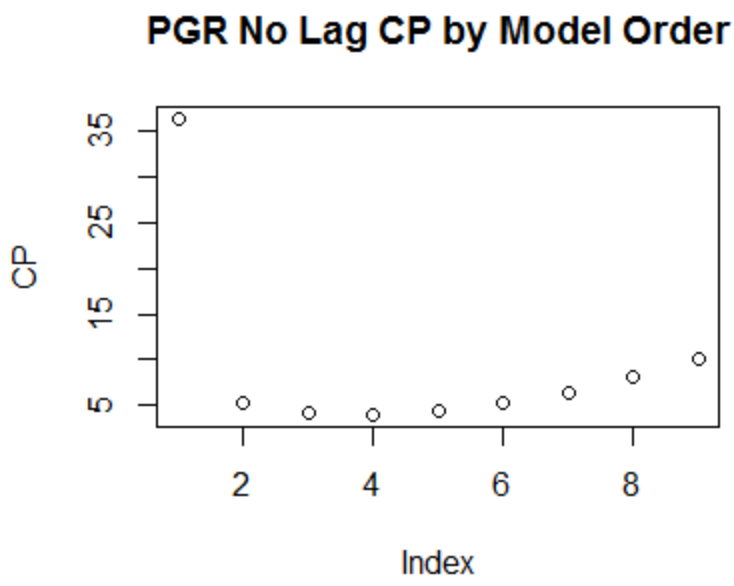
	PGR Linear	PGR Poly	PGR Tree	PGR Prune
No Lag	0.00038	0.00037	0.00116	0.00102
Lag1	0.00093	0.00050	0.00024	0.00024
Lag2	0.00056	0.00122	0.00026	0.00029

For PGR, all non classification methods resulted in significant coefficients. For No Lag returns, pruned regression tree produced the lowest MSE. For Lag1 returns, regression tree (pruned and not pruned) tied for the lowest MSE. For Lag2 returns, unpruned regression tree resulted in the lowest MSE.

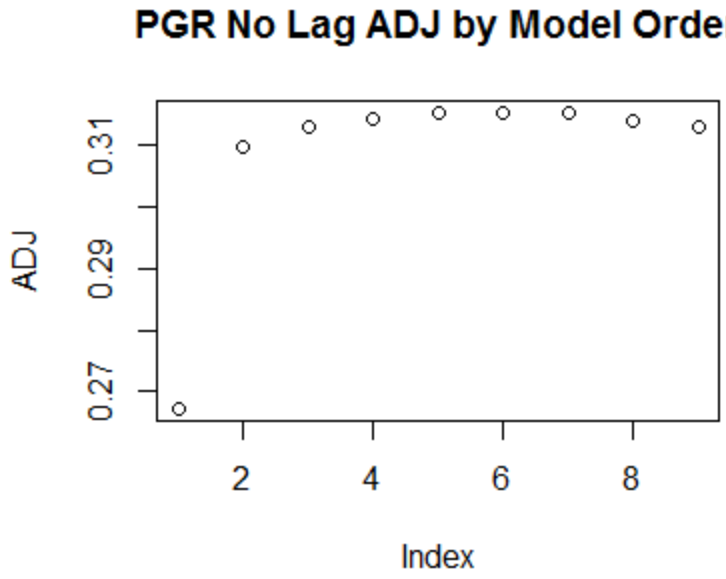
No Lag Best Subset Selection Model Order

	1	2	3	4	5	6	7	8	9
(Intercept)	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
buy_hit	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
sell_hit	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
bull_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
bear_hit	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
hig_hit	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
trv_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
hig_return	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_return	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

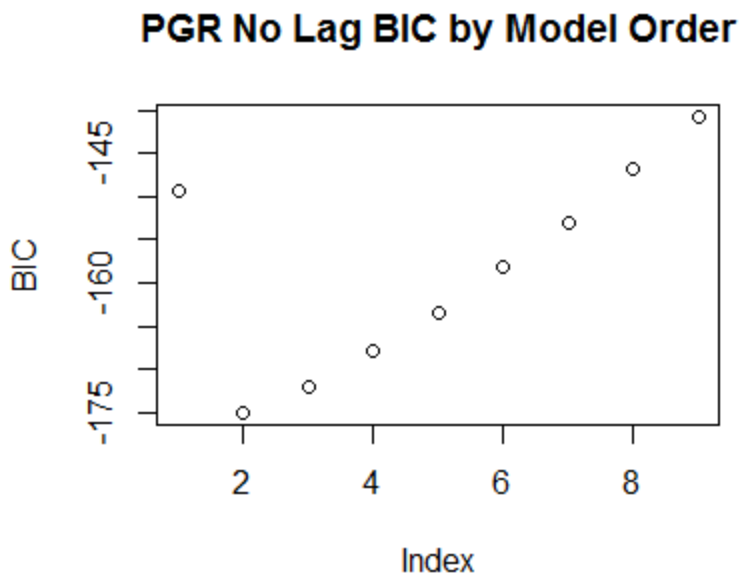
We selected the 3 model order. It does not minimize CP (4), maximize ADJ (7), or minimize BIC (2). However, 3 is still close to the min CP and max ADJ, while also being closer to the min BIC.



Models	CP	ADJ	BIC
4.0000000	4.0322042	0.3143566	-167.8977179



Models	CP	ADJ	BIC
7.0000000	6.3186781	0.3153272	-152.9241983

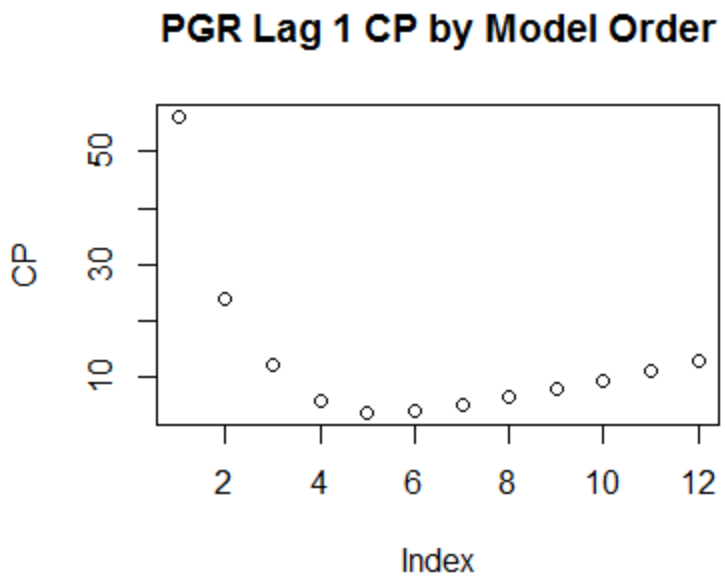


Models	CP	ADJ	BIC
2.0000000	5.3620342	0.3099014	-175.0296567

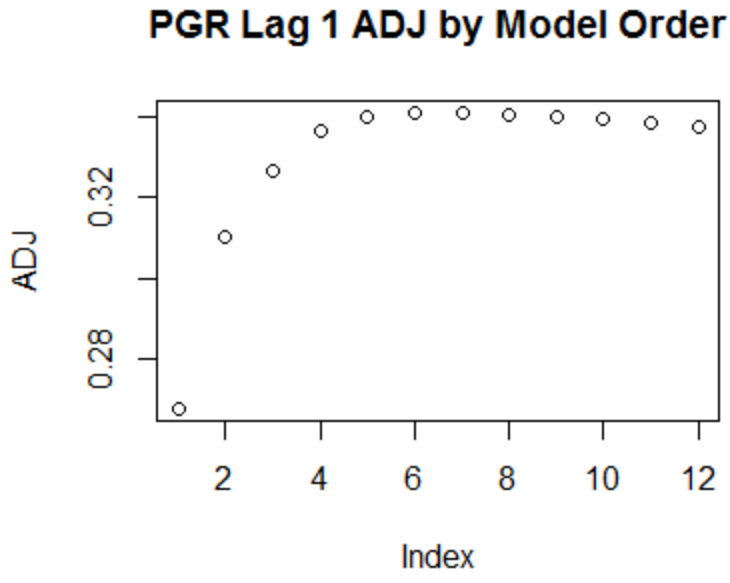
Lag 1 Best Subset Selection Model Order

	1	2	3	4	5	6	7	8	9	10	11	12
(Intercept)	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
buy_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
sell_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
bull_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
bear_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
hig_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_hit	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_return	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_return	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_return	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_r_Lag1	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_r_Lag1	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

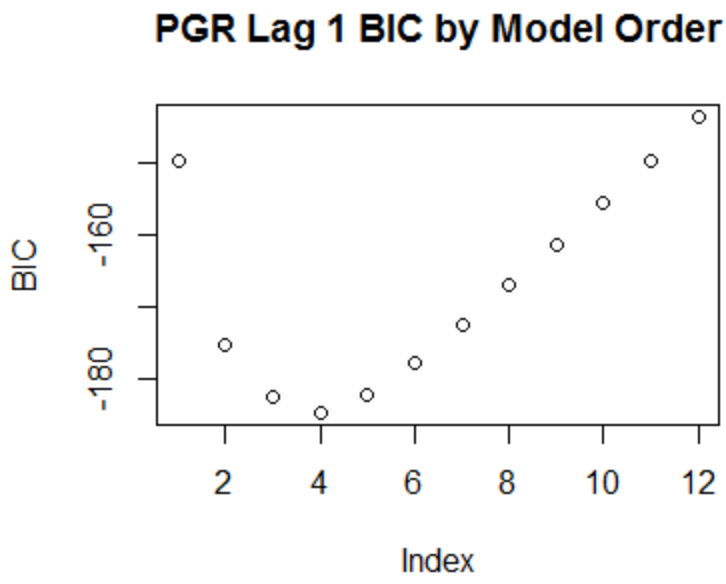
We selected the 5 model order because it is a reasonable middle ground between the minimum CP (5), Max ADJ (6), and min BIC (4).



Models	CP	ADJ	BIC
5.0000000	3.8999298	0.3400015	-182.3684835



Models	CP	ADJ	BIC
6.0000000	4.2283678	0.3408795	-177.8214037

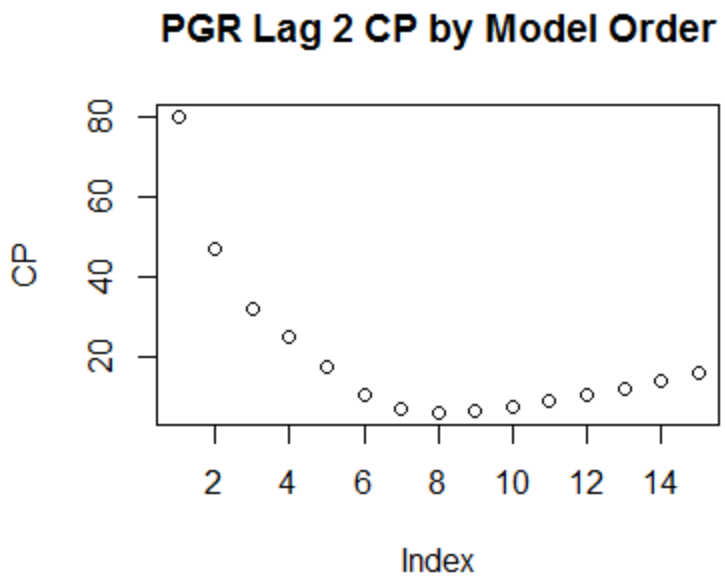


Models	CP	ADJ	BIC
4.0000000	5.7324903	0.3363297	-184.7375518

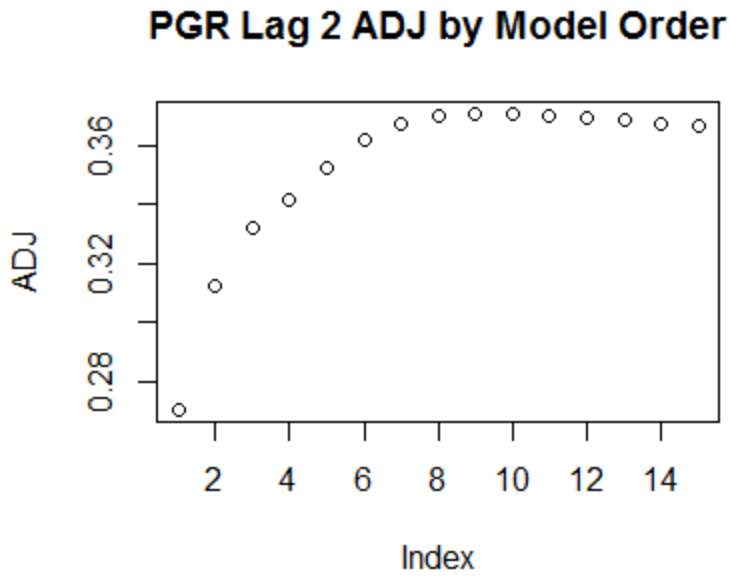
Lag 2 Best Subset Selection Model Order

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(Intercept)	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
buy_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE
sell_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
bull_hit	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
bear_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
hig_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
trv_hit	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_return	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
pgr_return	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_return	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
pgr_r_Lag1	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_r_Lag1	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_r_Lag1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
trv_r_Lag2	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
hig_r_Lag2	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

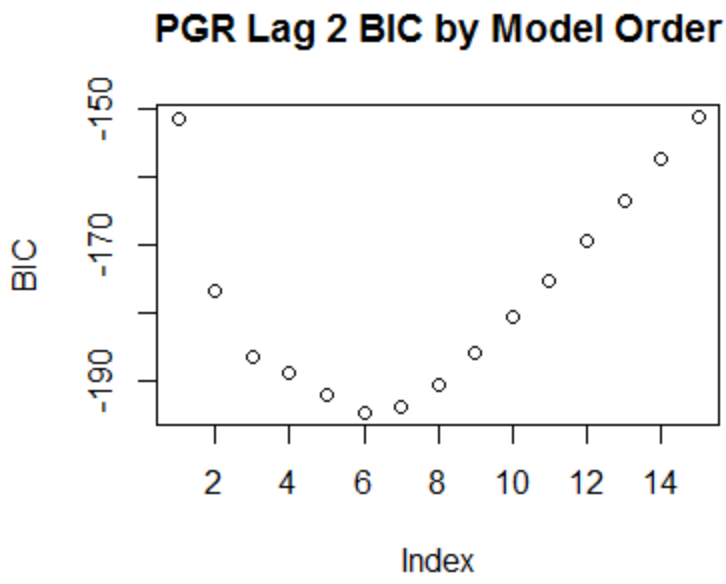
We selected the 8 model order. It is a middle ground between the minimum CP (at 8), maximum ADJ (10), and minimum BIC (6).



Models	CP	ADJ	BIC
8.0000000	6.3263197	0.3698999	-190.6361057



Models	CP	ADJ	BIC
10.0000000	7.6802380	0.3707219	-180.8543423



Models	CP	ADJ	BIC
6.0000000	10.5766470	0.3621238	-194.7594365

Significant Linear Variables by Return Lag

No Lag: hig_return + trv_return

Lag 1: trv_return + pgr_return + trv_r_Lag1 + hig_r_Lag1

Lag 2: sell_hit + bull_hit + pgr_return + pgr_r_Lag1 + trv_r_Lag1 + trv_r_Lag2 + hig_r_Lag2

Significant Polynomial Variables and Degrees of Freedom by Return Lag

No Lag: hig_return + trv_return (10 degrees of freedom)

Lag 1: trv_return + pgr_return + trv_r_Lag1 + hig_r_Lag1 (1 degree of freedom)

Lag 2: sell_hit + bull_hit + pgr_return + pgr_r_Lag1 + trv_r_Lag1 + trv_r_Lag2 + hig_r_Lag2 (3 degrees of freedom)

Unpruned Tree Variables and Nodes by Return Lag

No Lag: 9 nodes

Lag 1: 7 nodes

Lag 2: 8 nodes

Pruned Tree Variables and Nodes by Return Lag

No Lag: 4 nodes

Lag 1: 7 nodes

Lag 2: 4 nodes

Prediction Success Rate

	PGR GLM	PGR LDA	PGR QDA	PGR KNN
No Lag	44.79	66.41	69.11	72.20
Lag1	44.79	67.95	61.00	70.27
Lag2	47.10	49.42	69.50	65.25

For PGR, all classification methods produced significant coefficients. For No Lag and Lag1 returns, KNN produced the highest prediction success rate. For Lag2 returns, QDA produced the highest prediction success rate.

Significant GLM Variables by Return Lag (used for all classification methods)

Lag: hig_return + trv_return

Lag1: pgr_return + hig_r_Lag1 + trv_r_Lag1

Lag 2: bull_hit + pgr_r_Lag1 + hig_r_Lag2 + trv_r_Lag2

Comparative Results on Variables, Stocks, and Google Trend data

Predicted vs. Actual Returns for non classification methods

We analyzed the actuals vs. predicted across our non-classification models included linear regression, polynomial regression, and regression tree.

Predicted vs. Actual Returns for HIG

We found that there were significant residual values for the regression tree predictions in some of the models and then in polynomial models in the 2 week lag model.

For No Lag returns, our predicted values appeared to be closer to the actual values than either the Lag 1 or Lag 2 returns. This is as expected since, when comparing MSE among No Lag, Lag 1, and Lag 2 returns, our linear and polynomial regression methods had the lowest MSE for Non Lag returns. Interestingly, we did notice that around 2012, our tree models dramatically overestimate the actual values. This is true for all three HIG returns. Additionally, none of our models appear to predict dramatic spikes or drops in actual HIG returns. This is true among all HIG returns as well.

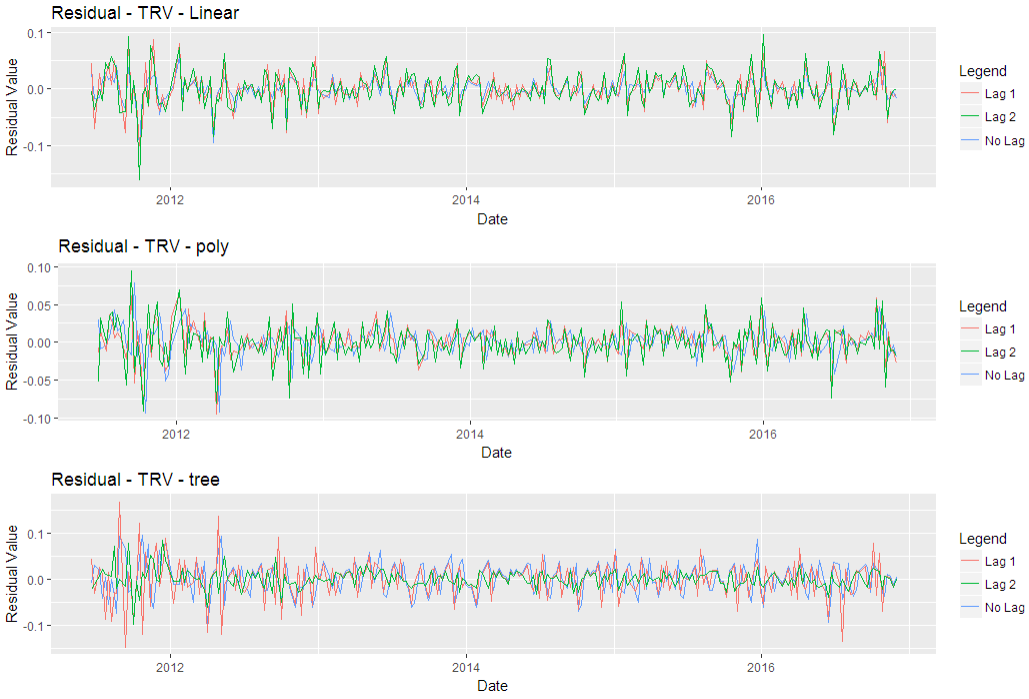
Residual values for HIG

We also compared the residual values over time for each of the major models and variables. For HIG, the general trend among our residuals is that our accuracy increased between 2012 and 2016. Regression tree had the lowest residual variance, followed by polynomial regression, and then linear regression.

Predicted vs. Actual Returns for TRV

We found that these models, while having some of the same issues as the HIG model, also had lower absolute ranges of variability. The tree model continues to be inaccurate around 2012 and improve in accuracy by 2013. TRV models still do not accurately predict sudden large increases or decreases in actual value for any return type.

Residual Values for TRV

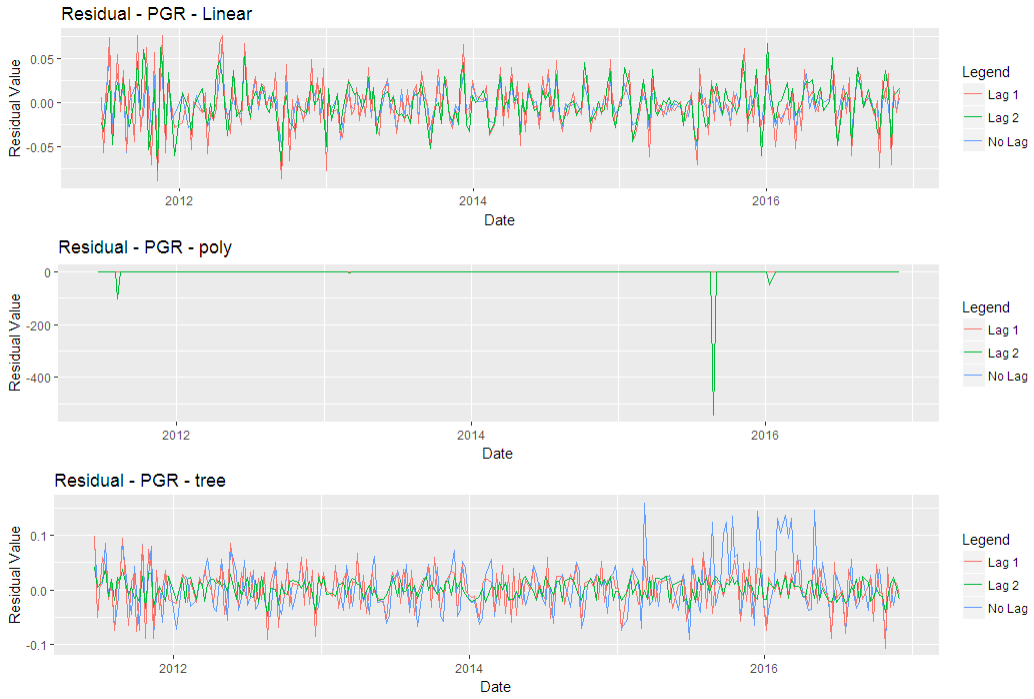


Residual values for TRV have substantially less variance than the HIG model. While HIG models generally had +/-.2 in residual variance, TRV has about +/- .1 in residual variance. While TRV is also less accurate around 2012, more than HIG, TRV is less accurate with negative residuals in 2012 and is less accurate in late 2016.

Predicted vs. Actual Returns for PGR

There appeared to be some significant residual values in the later dates for the PGR models with very strange results for our polynomial predictions on lag 2 weeks. Our Lag 2 model had unusually large predicted values from the polynomial regression. While this needs further analysis, this graphic hides the fact that our Lag 2 polynomial analysis for PGR actually produced one of our lowest MSEs, of .00122.

Residual Values for PGR

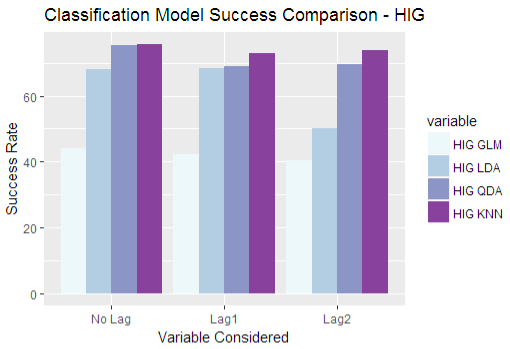


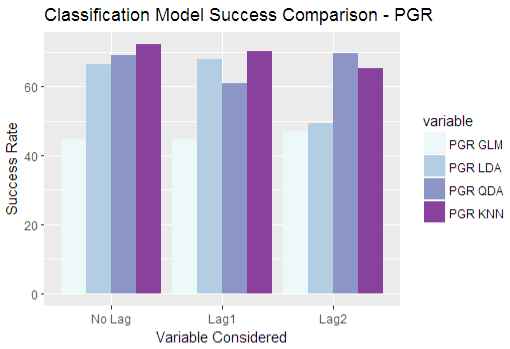
Our residual analysis for PGR shows that our accuracy was substantially better than HIG and slightly better than TRV. Excluding the outlier polynomial analysis, residuals for PGRs tree and linear regressions averaged between +/- .1 and +/- .05. More than either HIG or TRV, PGR suffered from less accuracy later in our time series, around 2016.

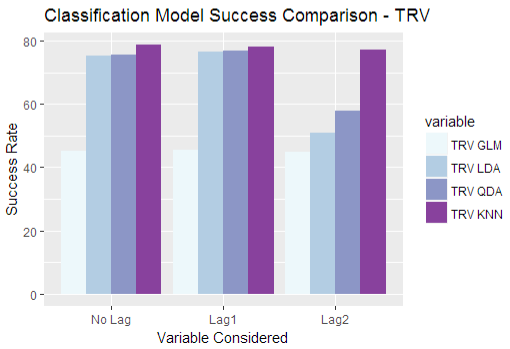
Prediction Success Rates for Classification Methods

We also compared the prediction accuracy of our classification models across the 3 stocks with analyzed. There were four models – logistic regression, LDA, QDA, and KNN.

Generally speaking, logistic regression yielded the lowest accuracy while KNN yielded the highest. LDA and QDA varied across variables and stocks.







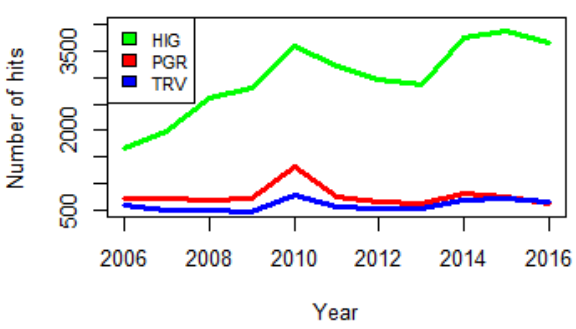
Between our three stocks, prediction success rates did not always respond similarly when measuring No Lag, Lag 1, or Lag 2. HIG had its best success rates for No Lag, and second best for Lag 2. PGR was best in No Lag, and second best in Lag 1. TRV was best in Lag 1 and second best in No Lag.

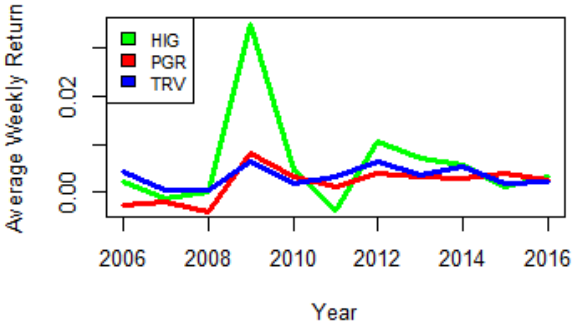
Comparison of Insurance Companies Analyzed

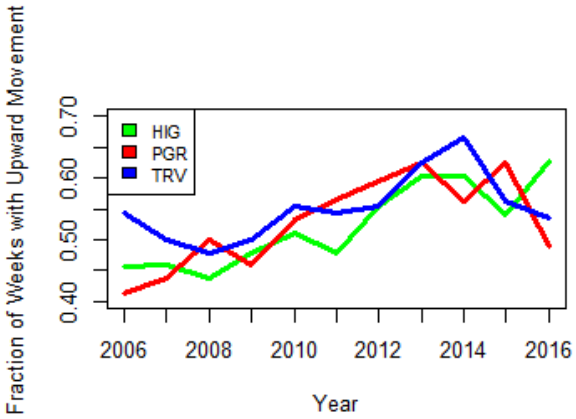
Apart from using various analytical techniques for predictions, we also conducted an analysis of our variables inputs and selected stocks to better interpret the results.

Comparison of Hits, Returns, and Movements over time

The chart below shows the year-to-year comparison of the three companies in terms of the number of hits, the average weekly return, and the fraction of weeks during which the stock went up.







As can be seen from the above, the year-to-year pattern of the number hits detected by gtredsR about the three companies is similar from 2009-2013. HIG has considerably more hits than the PGR or TRV, but the year-to-year pattern is comparable until 2014. In 2014 and 2015 there was an uptick in the number of HIG hits that was not there for PGR or TRV.

The average weekly returns for PGR and TRV are comparable after 2008; in 2006 and 2007 TRV had a higher return. HIG's weekly return spiked in 2009 and then was below the other two in 2011; otherwise it was comparable.

The fraction of weeks with upward movement in the stock price tells a different story than the average weekly return. TRV generally had the most weeks with an upward return, and PGR had more weeks with an upward return than HIG, except in 2006, 2009, 2014, and 2016. In 2016 HIG showed an upward trend while the other two declined.

One possible explanation for the discrepancy between the cross-company average weekly return and the cross-company fraction of weeks with upward movement could be that the percentages in the upward and downward weeks were of different magnitudes, so that the average weekly return could be influenced by a few weeks during which there were extreme movements up or down.

Comparison of Return Correlations

The sections from the spread sheets shown below indicate the most significant correlations among the eighteen variables: HIG, PGR, and TRV return and upward/downward movement for the week, the

previous week, and two weeks earlier. We are showing the top five positive correlations, and the top three negative correlations, as there were very few significantly negative correlations.

hig_return stat	hig_return corr	pgr_return stat	pgr_return corr	trv_return stat	trv_return corr
hig_r_Lag2	-0.1979	pgr_r_Lag1	-0.1908	trv_r_Lag2	-0.0821
hig_r_Lag1	-0.0894	trv_r_Lag1	-0.1249	hig_r_Lag1	-0.0669
hig_r_Lag1.d.num	-0.0556	pgr_r_Lag1.d.num	-0.0953	hig_r_Lag2	-0.0523
trv_return.d.num	0.2259	hig_return.d.num	0.3729	pgr_return.d.num	0.3755
pgr_return.d.num	0.2343	hig_return	0.3928	hig_return.d.num	0.3769
trv_return	0.3846	trv_return.d.num	0.4133	hig_return	0.3846
pgr_return	0.3928	trv_return	0.5183	pgr_return	0.5183
hig_return.d.num	0.4609	pgr_return.d.num	0.6973	trv_return.d.num	0.6707

hig lag 1 stat	hig_r_Lag1 corr	hig lag 2 stat	hig_r_Lag2 corr	pgr lag 1 stat	pgr_r_Lag1 corr
hig_r_Lag2	-0.0895	hig_return	-0.1979	pgr_return	-0.1908
hig_return	-0.0894	hig_r_Lag1	-0.0895	pgr_r_Lag2	-0.1892
trv_return	-0.0669	hig_return.d.num	-0.0715	trv_r_Lag2	-0.1244
trv_r_Lag1.d.num	0.2265	trv_r_Lag2.d.num	0.2263	hig_r_Lag1.d.num	0.3740
pgr_r_Lag1.d.num	0.2349	pgr_r_Lag2.d.num	0.2343	hig_r_Lag1	0.3929
trv_r_Lag1	0.3846	trv_r_Lag2	0.3848	trv_r_Lag1.d.num	0.4141
pgr_r_Lag1	0.3929	pgr_r_Lag2	0.3919	trv_r_Lag1	0.5188
hig_r_Lag1.d.num	0.4609	hig_r_Lag2.d.num	0.4611	pgr_r_Lag1.d.num	0.6981

pgr lag 2 stat	pgr_r_Lag2 corr	trv lag 1 stat	trv_r_Lag1 corr	trv lag 2 stat	trv_r_Lag2 corr
pgr_r_Lag1	-0.1892	pgr_return	-0.1249	pgr_r_Lag1	-0.1244
trv_r_Lag1.d.num	-0.0894	hig_r_Lag2	-0.0673	trv_return	-0.0821
pgr_r_Lag1.d.num	-0.0604	trv_hit	-0.0524	trv_hit	-0.0566
hig_r_Lag2.d.num	0.3696	pgr_r_Lag1.d.num	0.3771	hig_r_Lag2.d.num	0.3794
hig_r_Lag2	0.3919	hig_r_Lag1.d.num	0.3783	pgr_r_Lag2.d.num	0.3803

trv_r_Lag2.d.num	0.4175	hig_r_Lag1	0.3846	hig_r_Lag2	0.3848
trv_r_Lag2	0.5216	pgr_r_Lag1	0.5188	pgr_r_Lag2	0.5216
pgr_r_Lag2.d.num	0.6982	trv_r_Lag1.d.num	0.6719	trv_r_Lag2.d.num	0.6716

hig movement stat	**hig_return.d.num corr**	**pgr movement stat**	**pgr_return.d.num corr**	**trv movement stat**	**pgr_return.d.num corr**
hig_r_Lag2	-0.0715	pgr_r_Lag1	-0.0642	pgr_r_Lag1	-0.0642
hig_r_Lag1.d.num	-0.0351	pgr_r_Lag1.d.num	-0.0633	pgr_r_Lag1.d.num	-0.0633
pgr_r_Lag1.d.num	-0.0277	hig_r_Lag2	-0.0488	hig_r_Lag2	-0.0488
pgr_return.d.num	0.3601	hig_return	0.2343	hig_return	0.2343
pgr_return	0.3729	hig_return.d.num	0.3601	hig_return.d.num	0.3601
trv_return	0.3769	trv_return	0.3755	trv_return	0.3755
trv_return.d.num	0.4061	trv_return.d.num	0.4168	trv_return.d.num	0.4168
hig_return	0.4609	pgr_return	0.6973	pgr_return	0.6973

hig lag 1 movement stat	**hig_r_Lag1.d.num corr**	**hig lag 2 movement stat**	**hig_r_Lag2.d.num corr**	**pgr lag 1 movement stat**	**pgr_r_Lag1.d.num corr**
hig_return	-0.0556	hig_r_Lag1	-0.0563	pgr_return	-0.0953
hig_r_Lag2.d.num	-0.0392	hig_r_Lag1.d.num	-0.0392	trv_return.d.num	-0.0817
hig_return.d.num	-0.0351	trv_r_Lag1	-0.0267	pgr_return.d.num	-0.0633
pgr_r_Lag1.d.num	0.3639	pgr_r_Lag2.d.num	0.3599	hig_r_Lag1	0.2349
pgr_r_Lag1	0.3740	pgr_r_Lag2	0.3696	hig_r_Lag1.d.num	0.3639
trv_r_Lag1	0.3783	trv_r_Lag2	0.3794	trv_r_Lag1	0.3771
trv_r_Lag1.d.num	0.4097	trv_r_Lag2.d.num	0.4097	trv_r_Lag1.d.num	0.4168
hig_r_Lag1	0.4609	hig_r_Lag2	0.4611	pgr_r_Lag1	0.6981

pgr lag 2 movement stat	**pgr_r_Lag2.d.num corr**	**trv lag 1 movement stat**	**trv_r_Lag1.d.num corr**	**trv lag 2 movement stat**	**trv_r_Lag2.d.num corr**
pgr_r_Lag1	-0.0941	pgr_r_Lag2	-0.0894	pgr_r_Lag1	-0.0333
trv_r_Lag1.d.num	-0.0783	pgr_r_Lag2.d.num	-0.0783	trv_hit	-0.0307
pgr_r_Lag1.d.num	-0.0597	trv_r_Lag2	-0.0381	trv_r_Lag1	-0.0294
hig_r_Lag2	0.2343	hig_r_Lag1	0.2265	hig_r_Lag2	0.2263
hig_r_Lag2.d.num	0.3599	hig_r_Lag1.d.num	0.4097	hig_r_Lag2.d.num	0.4097
trv_r_Lag2	0.3803	pgr_r_Lag1	0.4141	pgr_r_Lag2	0.4175
trv_r_Lag2.d.num	0.4204	pgr_r_Lag1.d.num	0.4168	pgr_r_Lag2.d.num	0.4204
pgr_r_Lag2	0.6982	trv_r_Lag1	0.6719	trv_r_Lag2	0.6716

A few observations on the above:

- The UP and DOWN movement does not seem to have a strong negative correlation with any of the other variables.

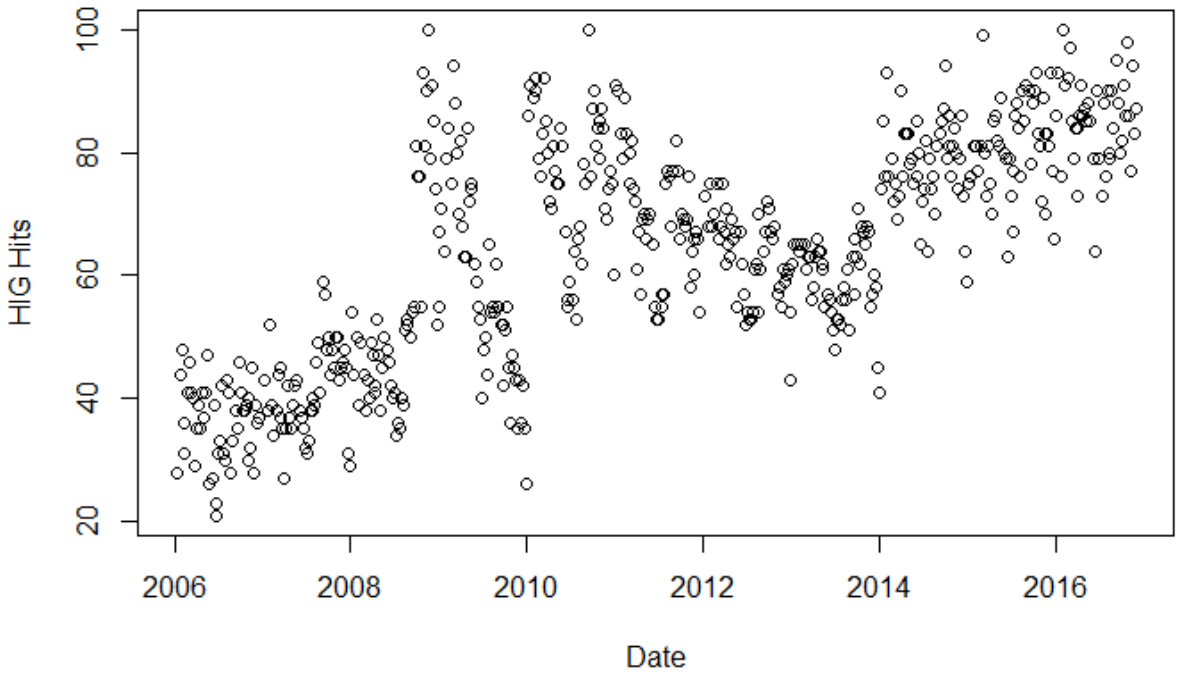
- The lag 1 and lag 2 variables for the weekly return are not correlated with the current-week return for TRV and are negatively correlated with the weekly return for HIG (lag 1) and PGR (lag 2) at almost the 20% level.
- The lag 1 and lag 2 variables for the UP/DOWN movement are not strongly correlated with the weekly movement.

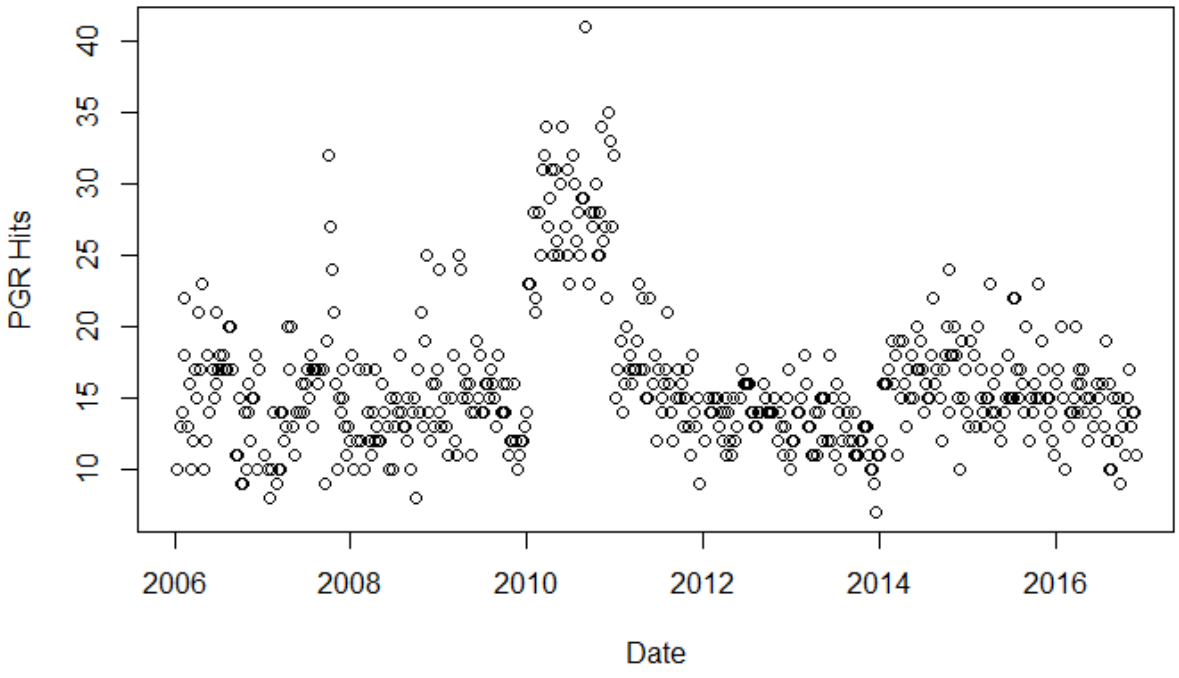
The above 2 bullets indicate that there is probably not much auto-correlation on a week-to-week basis, so the previous weeks by themselves would not be useful for predicting the current week's return.

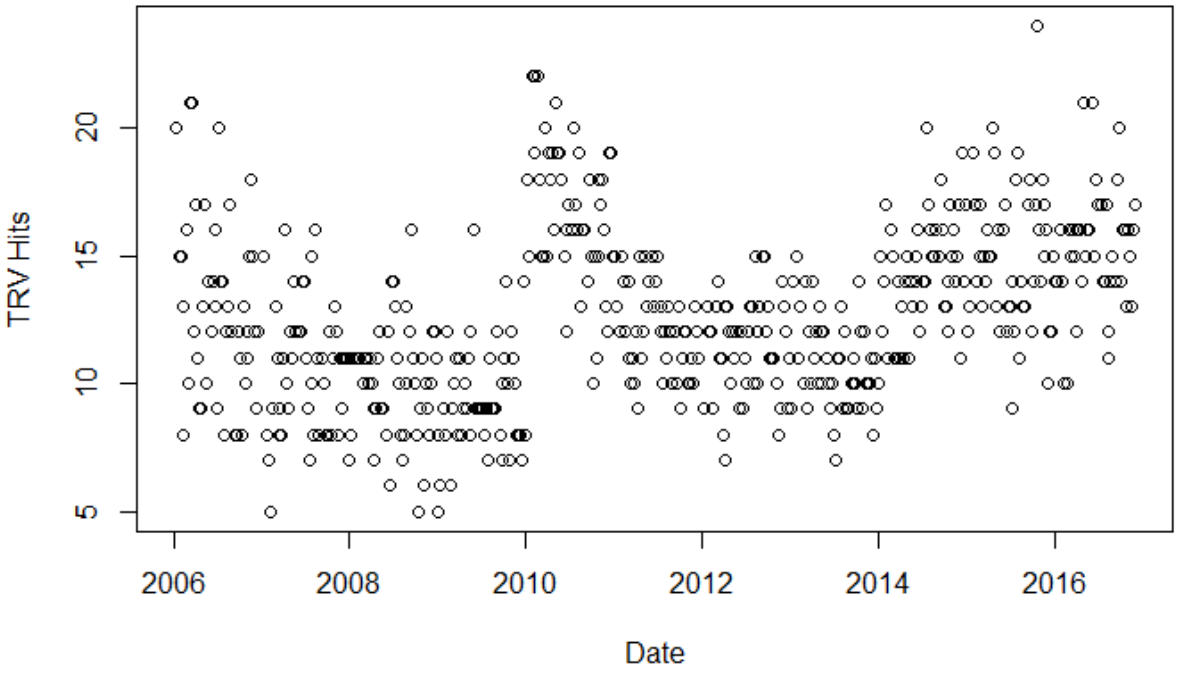
- The weekly, lag 1, and lag 2 returns and UP/DOWN movements are strongly correlated (from 30-50%) between the three stocks, with coefficients of 38-51% for the returns and 36-42% for the UP/DOWN movements. This is a further illustration of what the charts above showed visually.
- The correlations between the returns and the UP/DOWN movements were strong for all 3, but were much higher for PGR and TRV (67-70%) than for HIG (46%).
- There does not appear to be any correlation between the amount of hits in gtrendsR and the weekly returns or the weekly UP/DOWN movement.

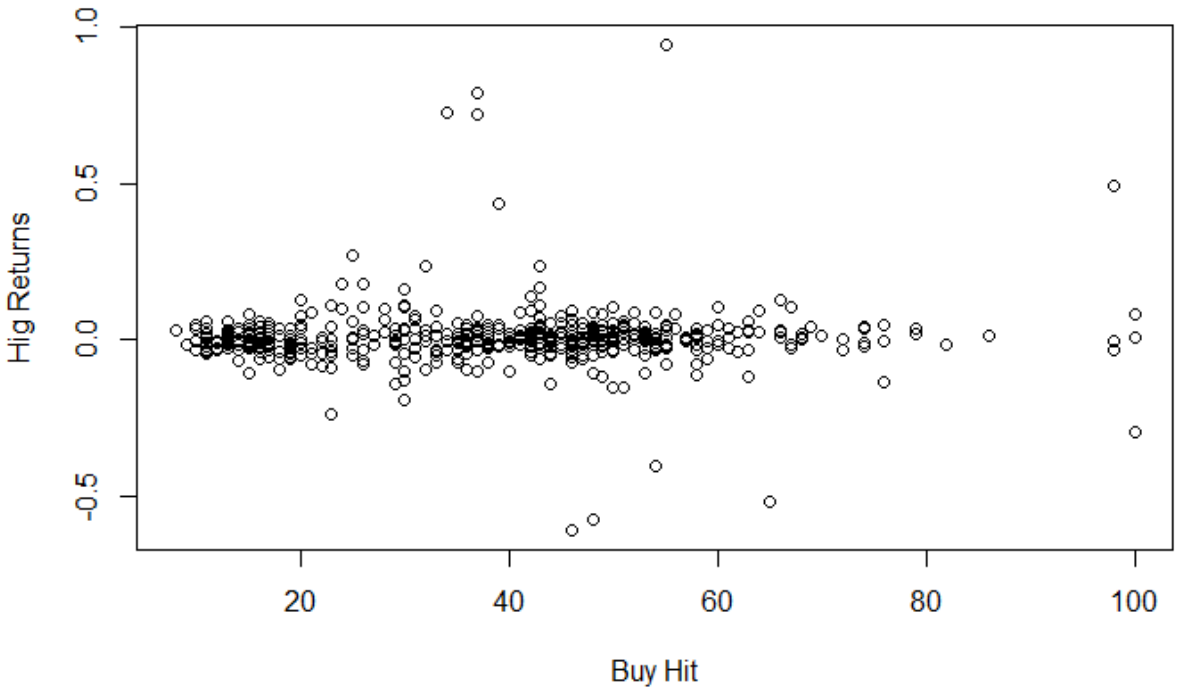
Comparison by Google Trend Keyword/Phrase

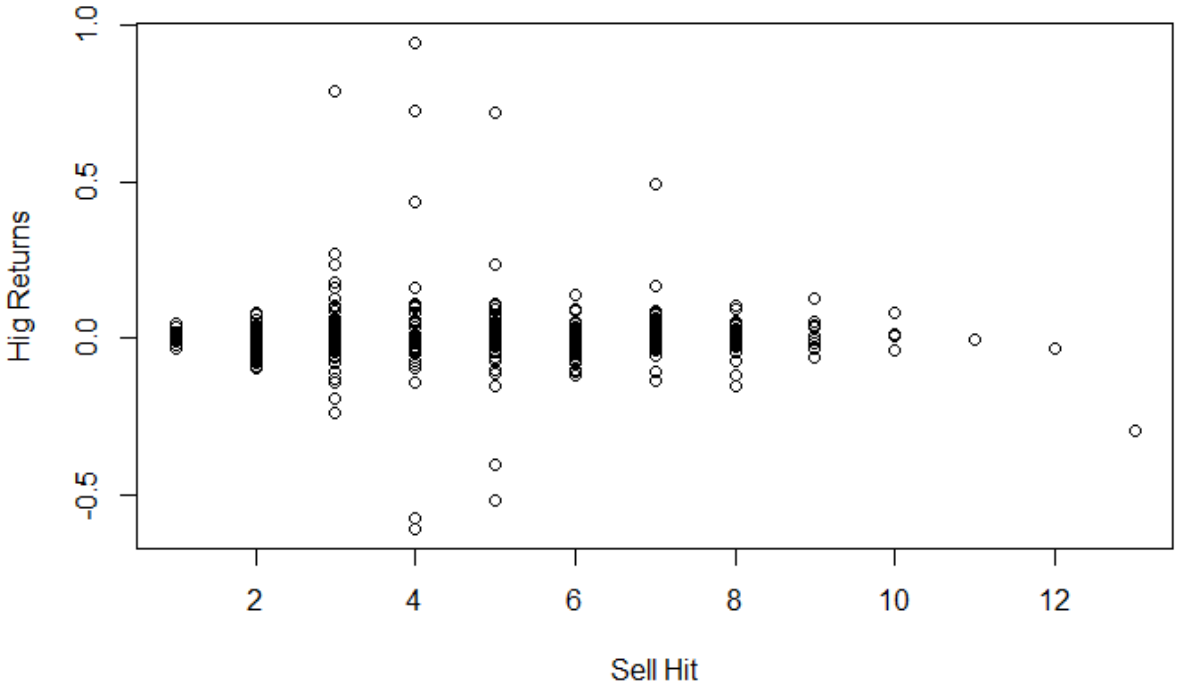
Comparing the three companies, HIG has considerably more searches on Google than the other two insurance companies. The number of HIG hits over our data is 33000 compared to the number of PGR hits (8301) and TRV hits (6495). This could be due to how similar the word HIG is to other words like hug, high, hog, etc. The other two stock abbreviations wouldn't come across as typos with normal everyday searches. Looking at additional data on Google Trends, Connecticut contains the highest amount of searches for HIG which makes sense as the Hartford Insurance Group is located in Hartford, CT. The odd thing about this data is that prior to 2010, HIG as a keyword performed only slightly higher than PGR and TRV in terms of number of searches. After 2010, it spiked to intervals close to 100. All three company keywords experienced a spike in 2010 but HIG showed the most significance in their spike of search. This is seen through the three graphs below on their hit intervals across the 10 years of data from Google Trends.

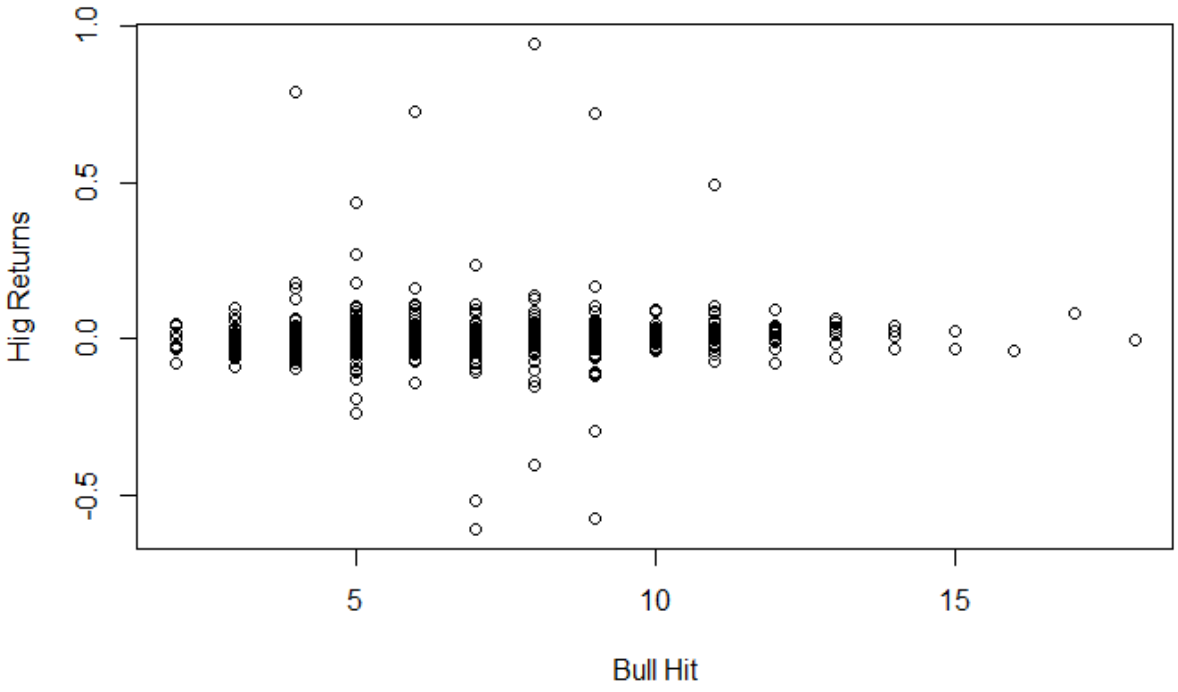

HIG Hits
20
40
60
80
100
2006
2008
2010
2012
2014
2016
Date

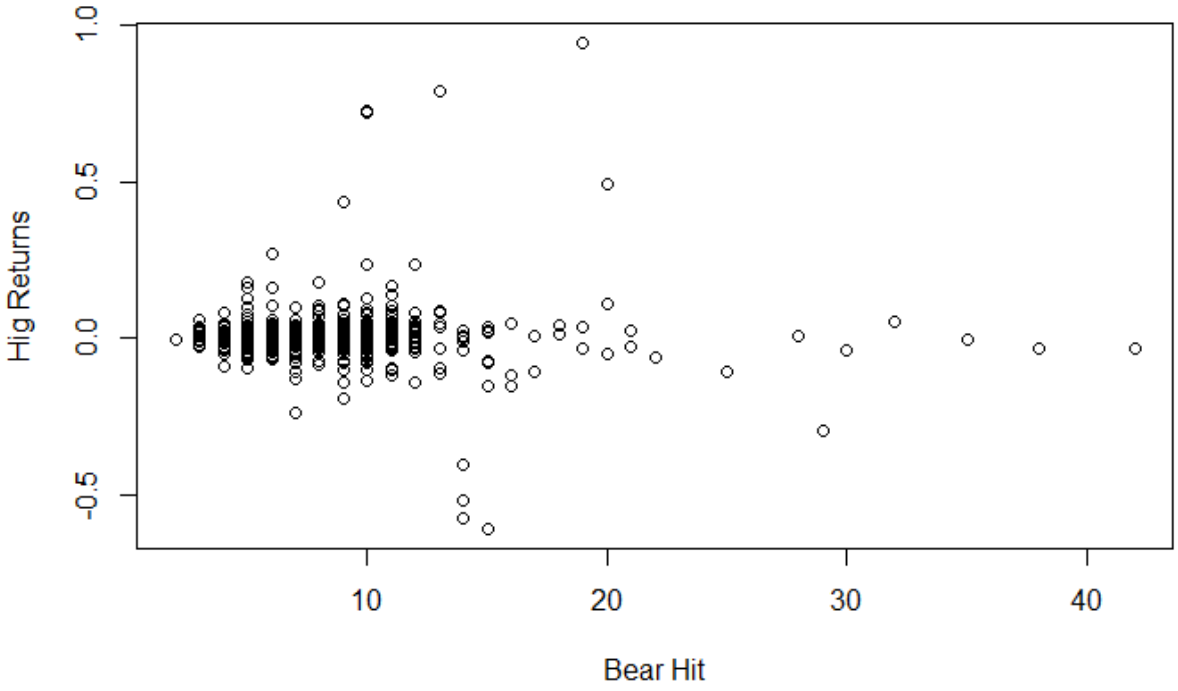

PGR Hits
10
15
20
25
30
35
40
2006
2008
2010
2012
2014
2016
Date

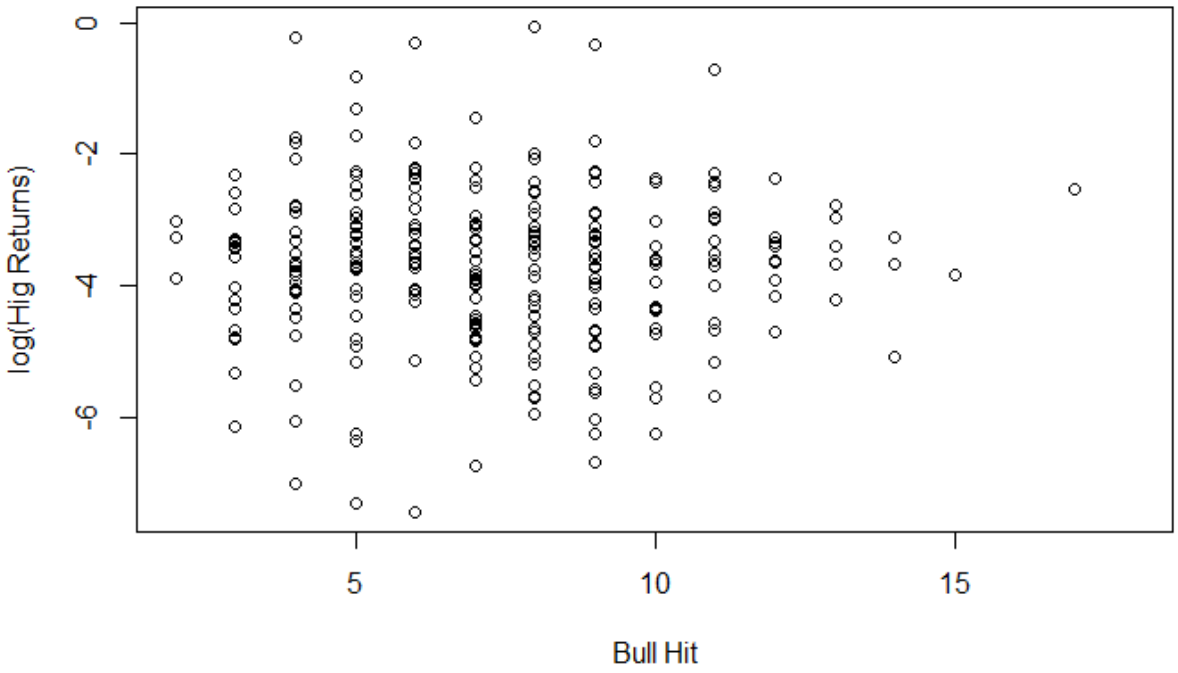

TRV Hits
5
10
15
20
2006
2008
2010
2012
2014
2016
Date

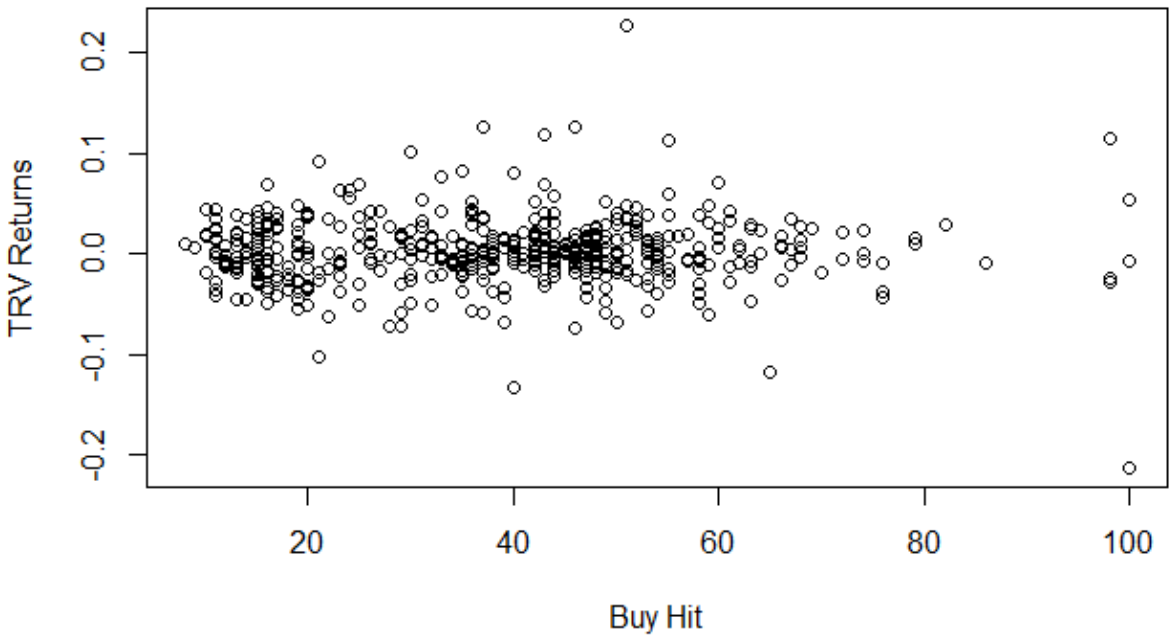

Hig Returns
-0.5
0.0
0.5
1.0
20
40
60
80
100
Buy Hit

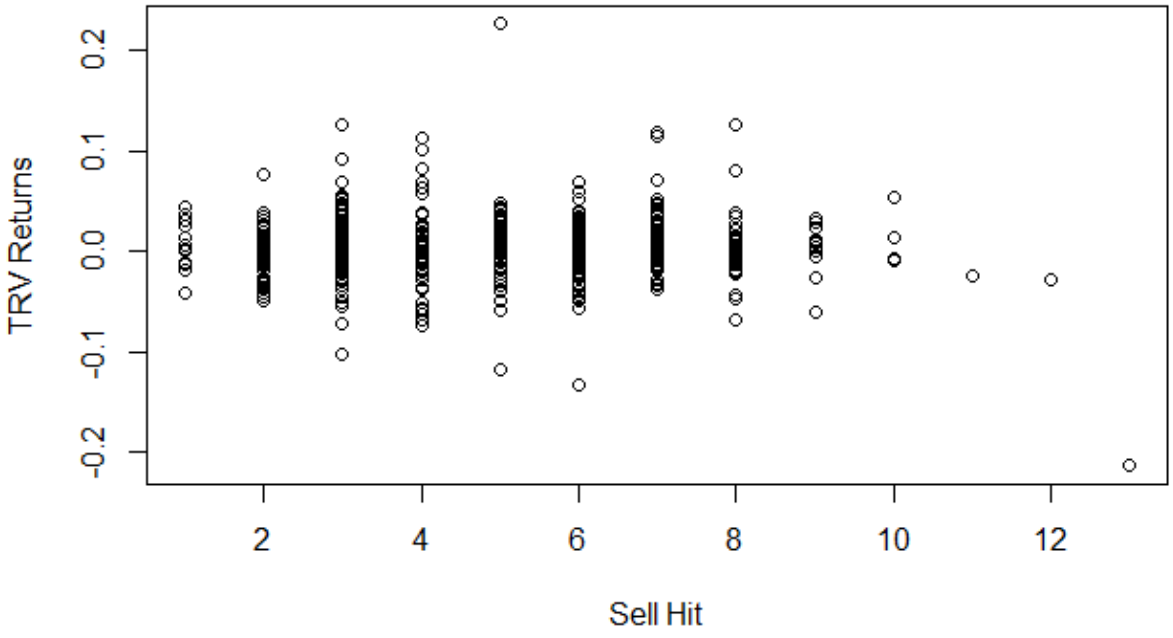

Hig Returns
1.0
0.5
0.0
-0.5
2
4
6
8
10
12
Sell Hit

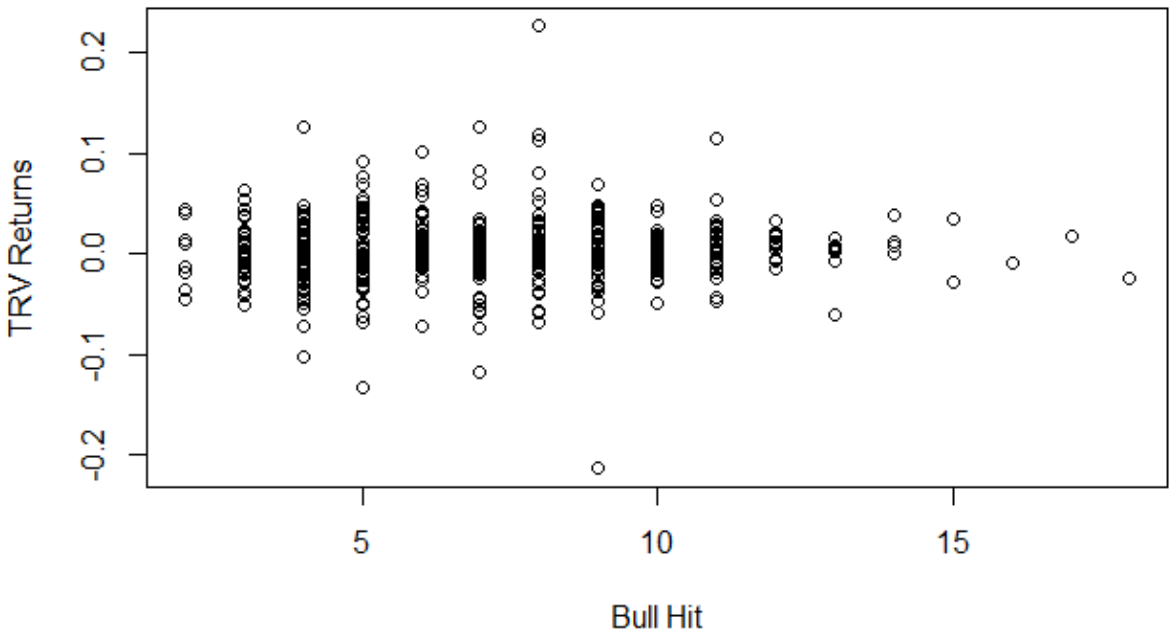

Hig Returns
1.0
0.5
0.0
-0.5
5
10
15
Bull Hit

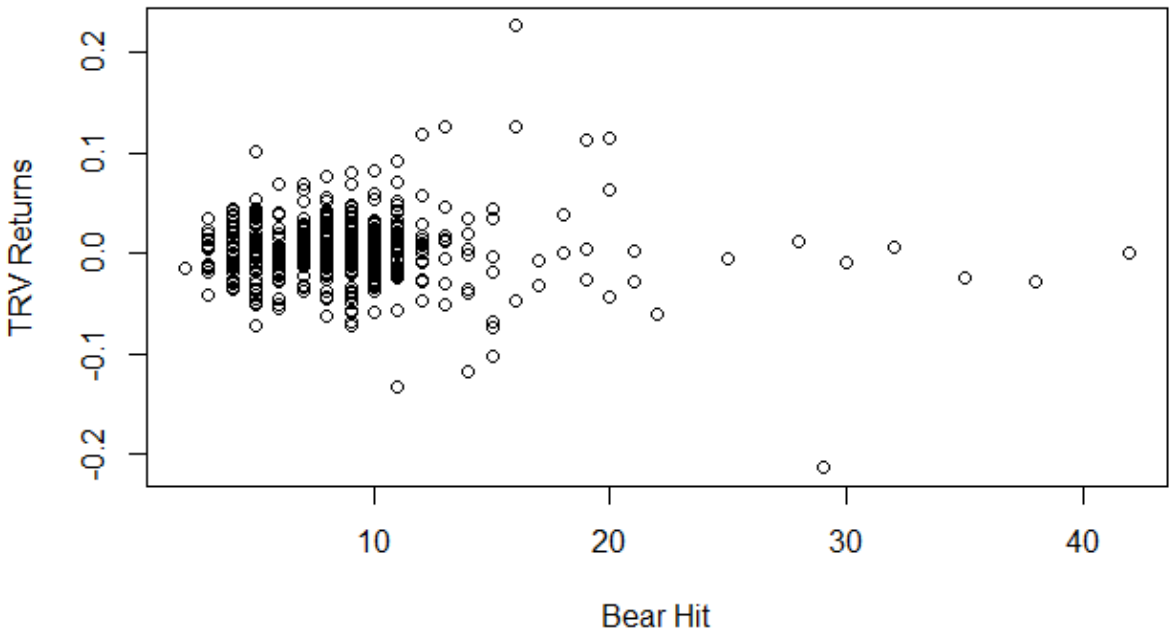

Hig Returns
1.0
0.5
0.0
-0.5
10
20
30
40
Bear Hit

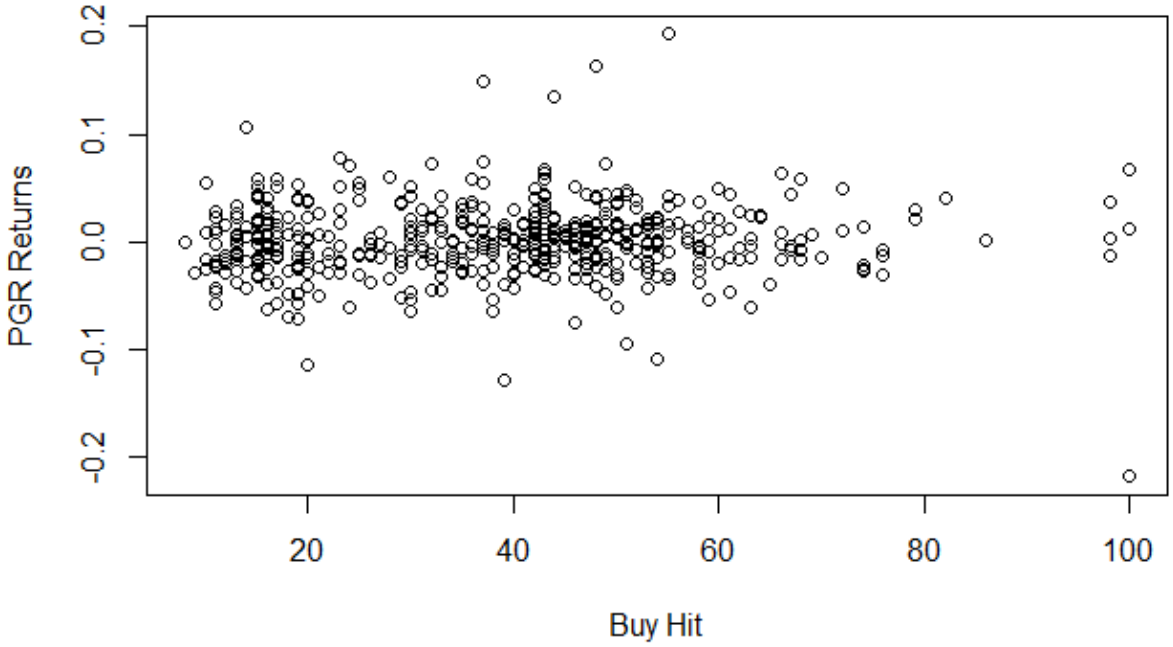

log(Hig Returns)
0
-2
-4
-6
5
10
15
Bull Hit

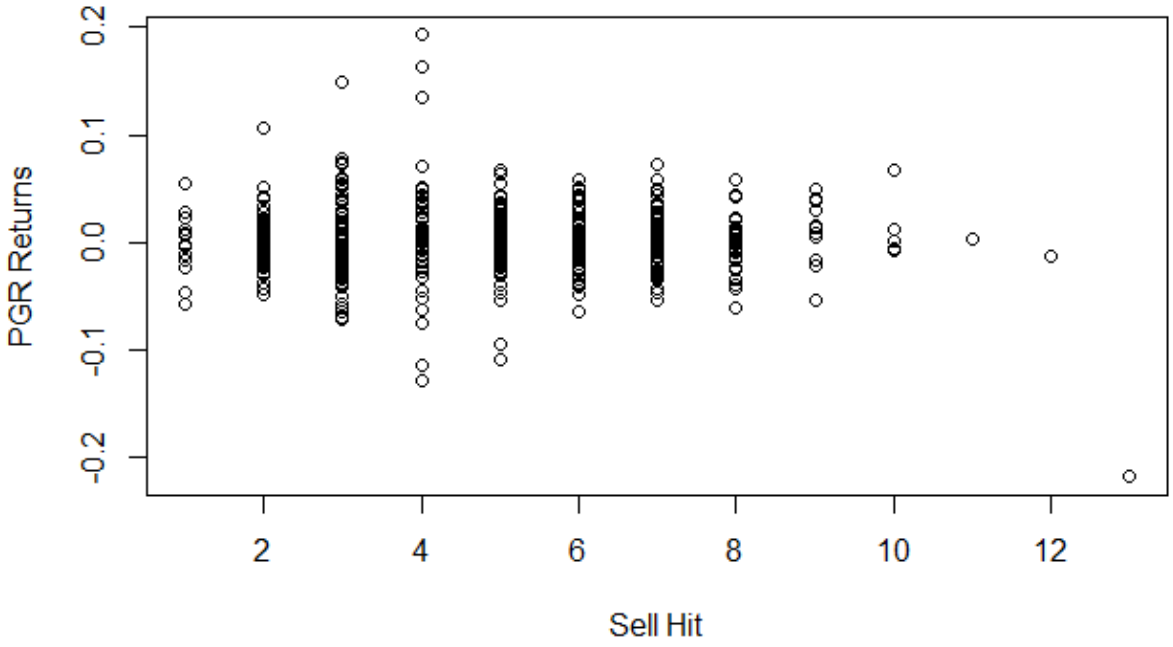

TRV Returns
0.2
0.1
0.0
-0.1
-0.2
20
40
60
80
100
Buy Hit

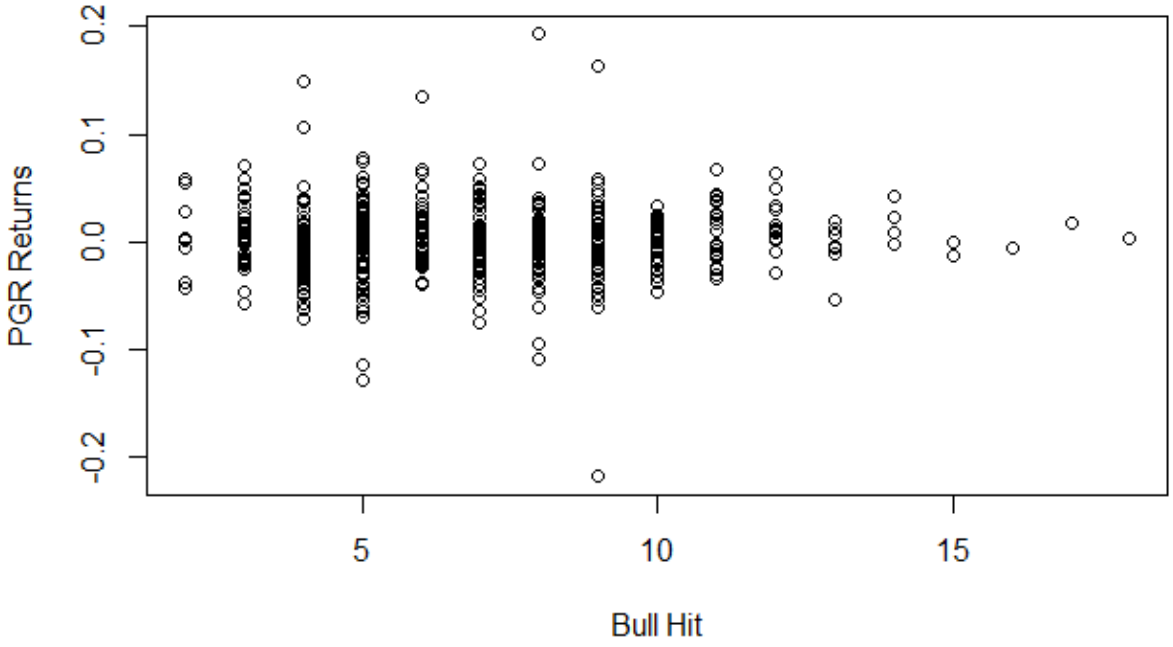

TRV Returns
0.2
0.1
0.0
-0.1
-0.2
2
4
6
8
10
12
Sell Hit

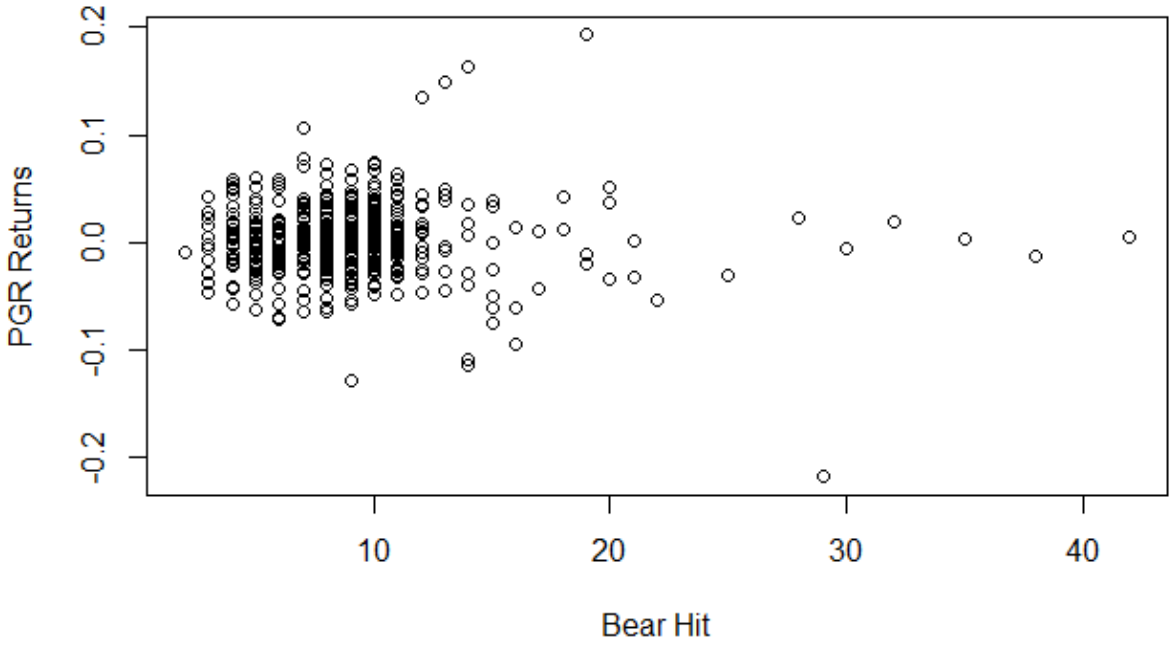

TRV Returns
0.2
0.1
0.0
-0.1
-0.2
5
10
15
Bull Hit


TRV Returns
0.2
0.1
0.0
-0.1
-0.2
10
20
30
40
Bear Hit


PGR Returns
0.2
0.1
0.0
-0.1
-0.2
20
40
60
80
100
Buy Hit

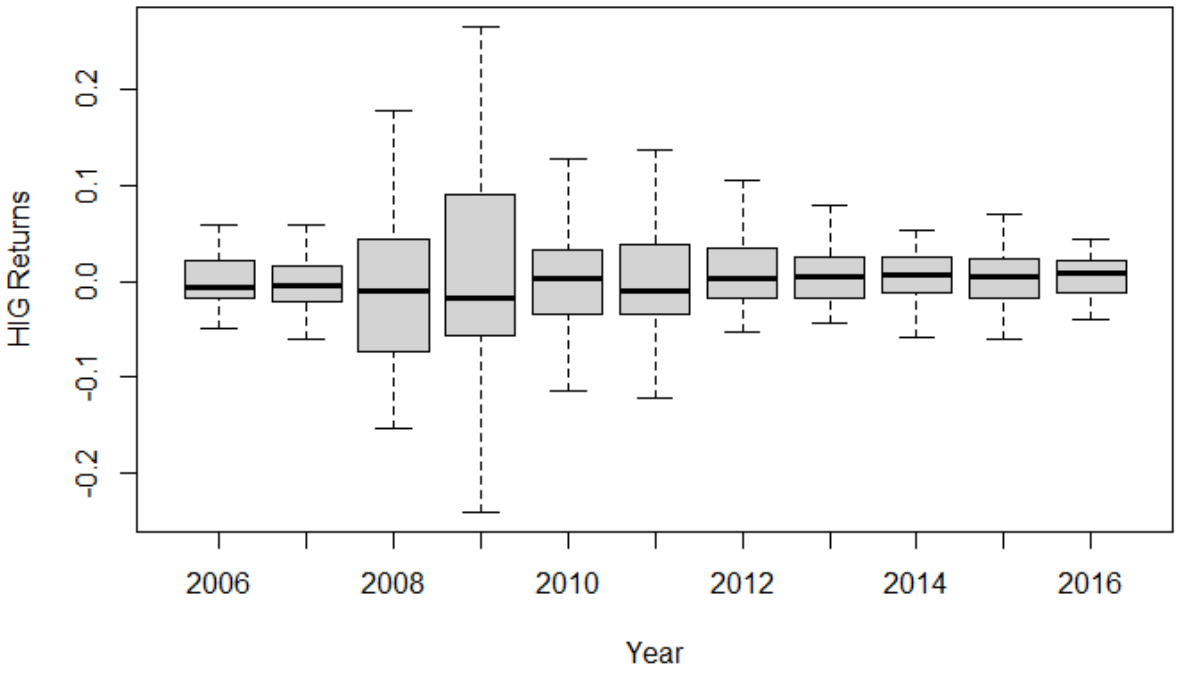

PGR Returns
0.2
0.1
0.0
-0.1
-0.2
2
4
6
8
10
12
Sell Hit

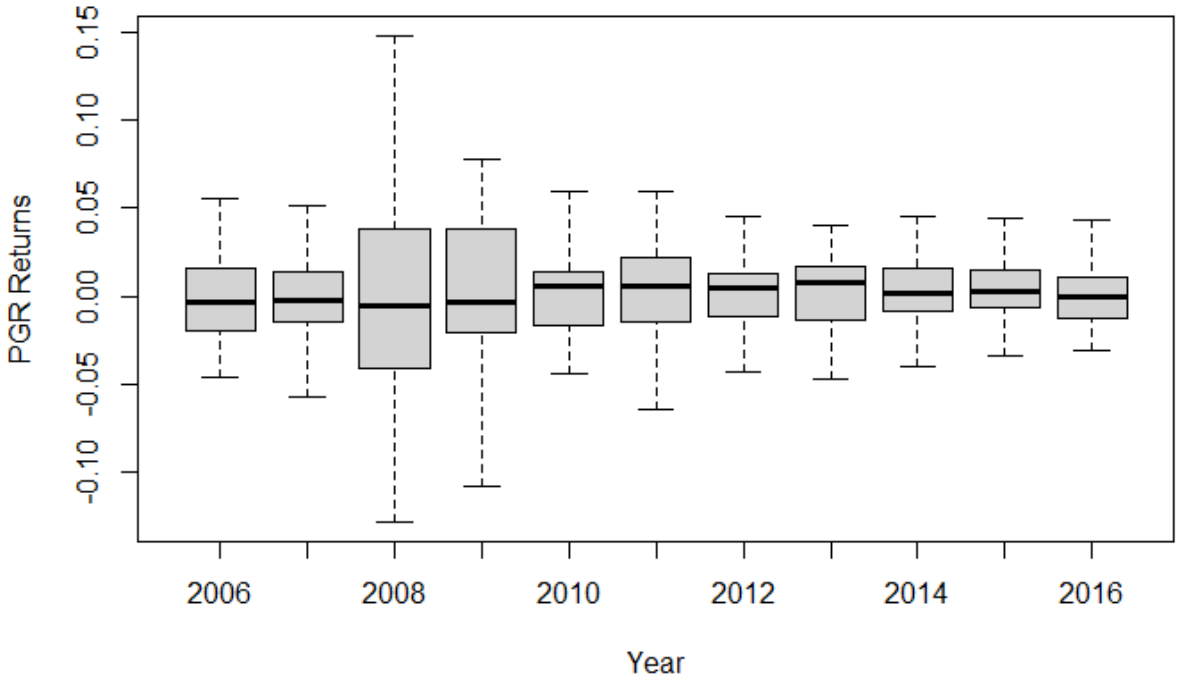

PGR Returns
0.2
0.1
0.0
-0.1
-0.2
5
10
15
Bull Hit

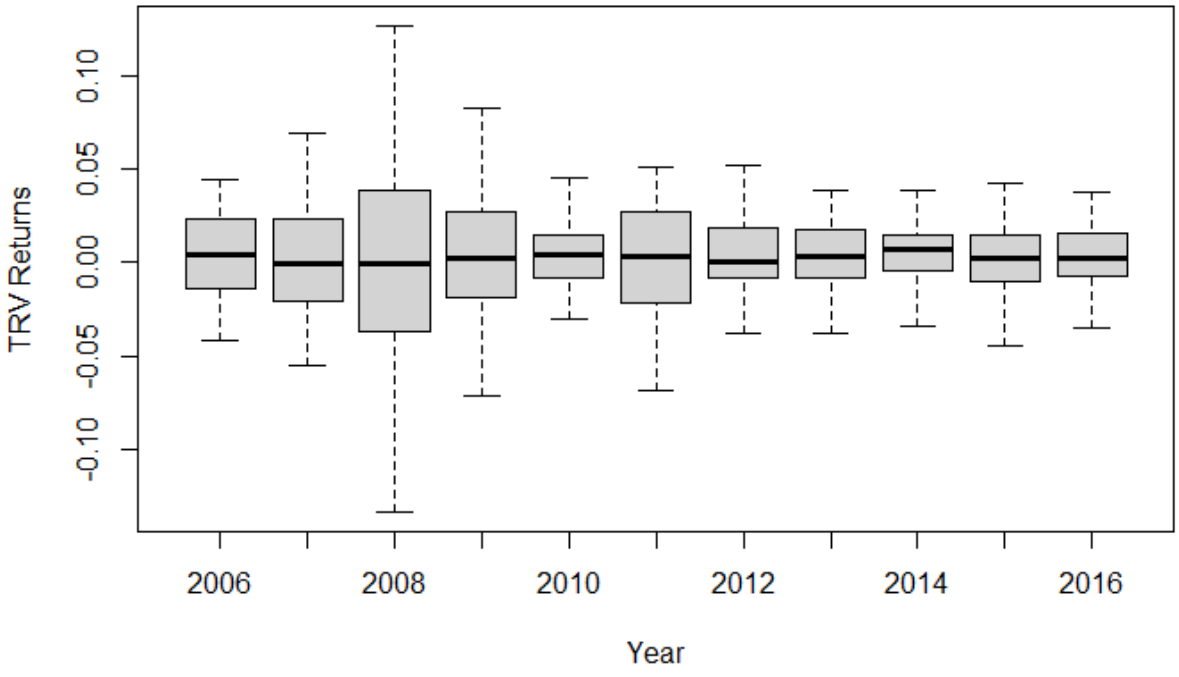

PGR Returns
0.2
0.1
0.0
-0.1
-0.2
10
20
30
40
Bear Hit

Looking at the insurance company returns below, we see the dynamic times around 2008 and 2009 when the US economy was going through a recession. HIG was impacted the most in terms of their stock performance. The other two companies experienced volatility during those years but not to the extent that HIG did. In 2010, the three stocks were returning to a more consistent performance but in 2011, they appeared to have gone through another period of volatility. Looking at news from that year, the overall performance of the market ended almost exactly where it started but it experienced a number of shifts before getting back to where it was when the year started.

From 2013 to 2016, the performance of all three stocks has not experienced as many swings and seems to be performing as it did before the 2008 recession of the United States. With TRV and PGR, it looks as their performance has experienced more consistency in the more recent years than it did prior to the 2008 recession.

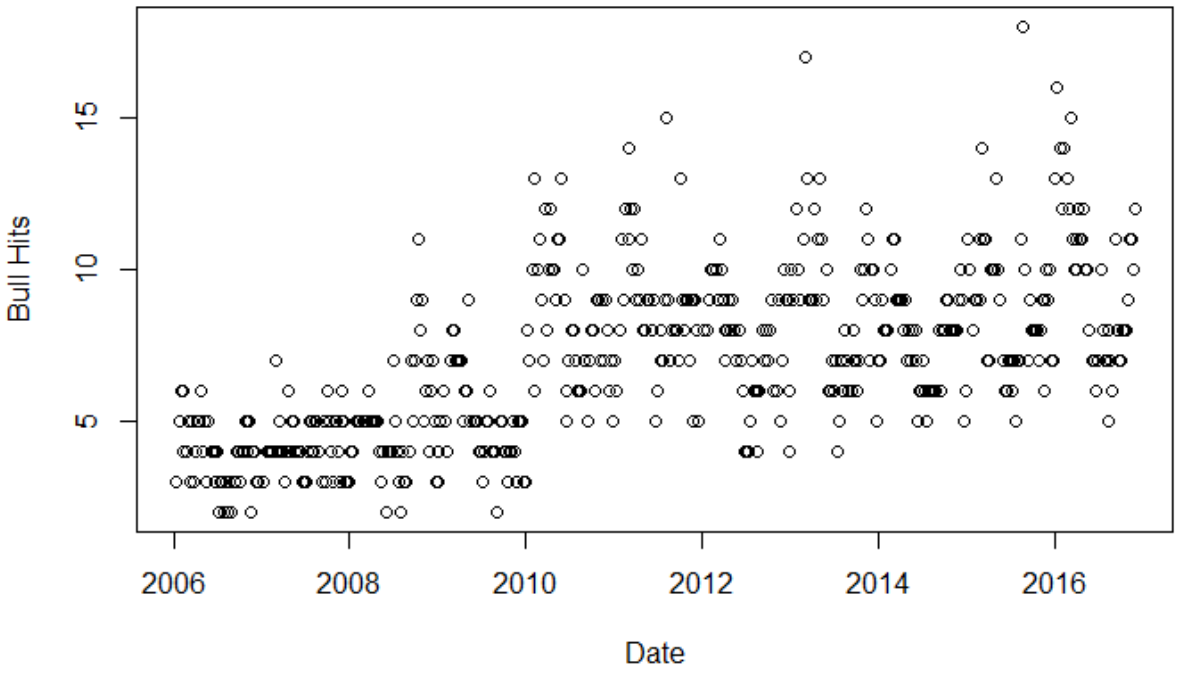


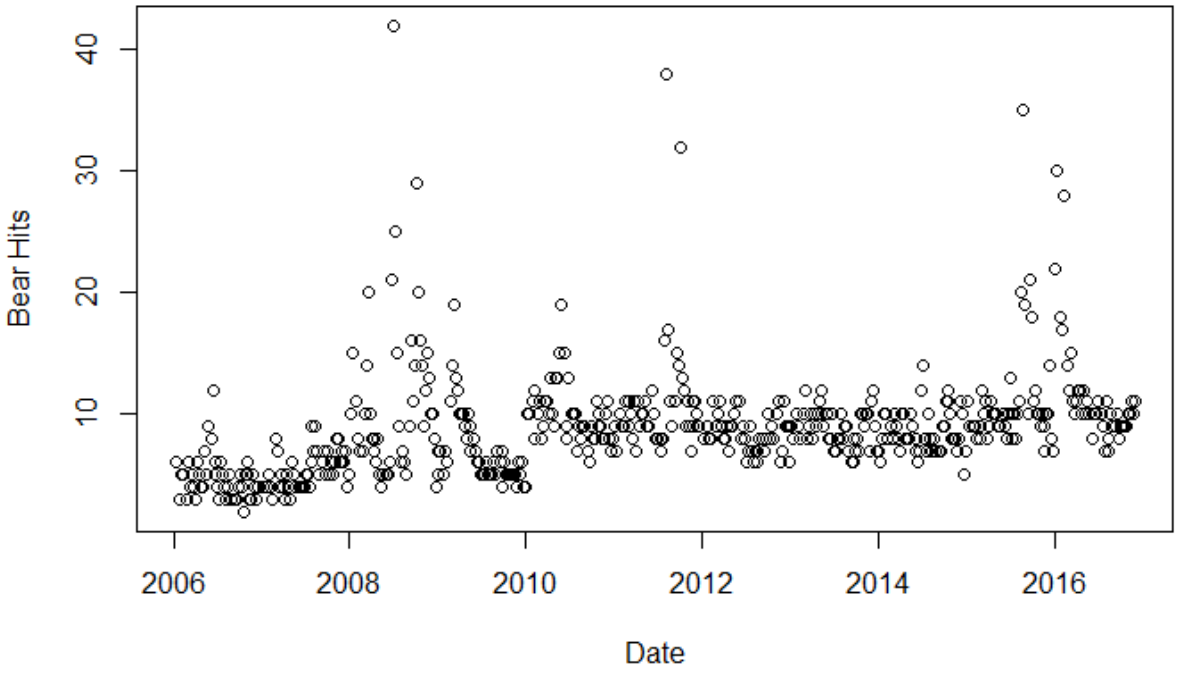

PGR Returns
0.15
0.10
0.05
0.00
-0.05
-0.10
2006
2008
2010
2012
2014
2016
Year

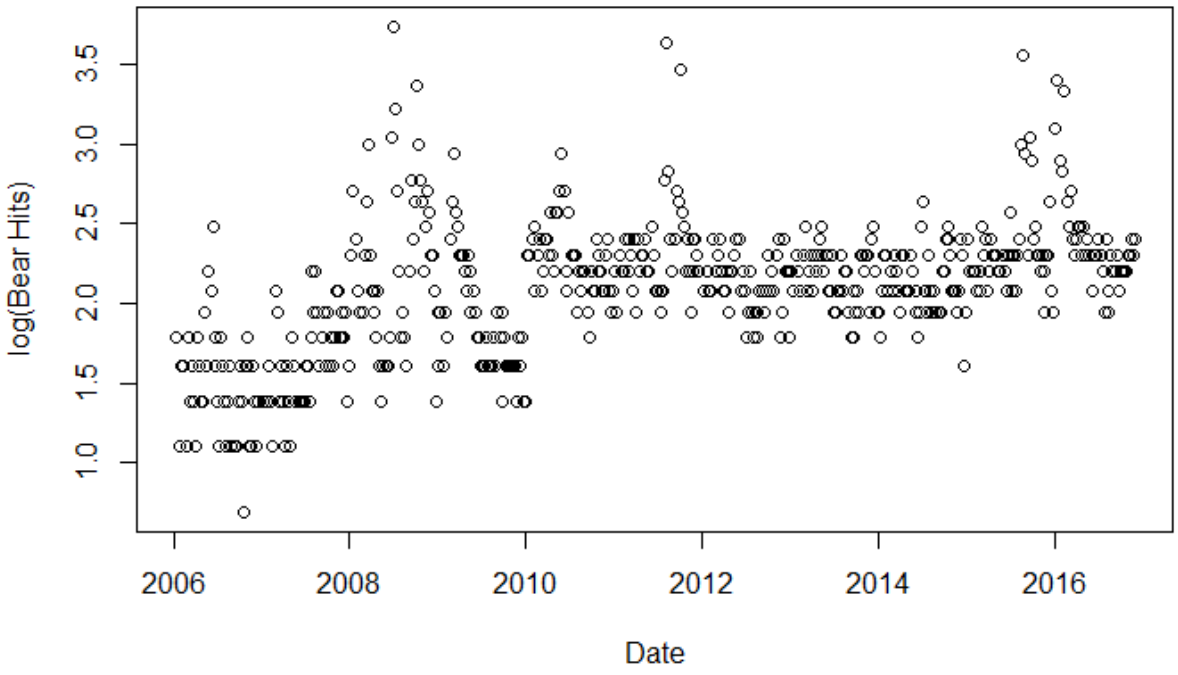

TRV Returns
0.10
0.05
0.00
-0.05
-0.10
2006
2008
2010
2012
2014
2016
Year

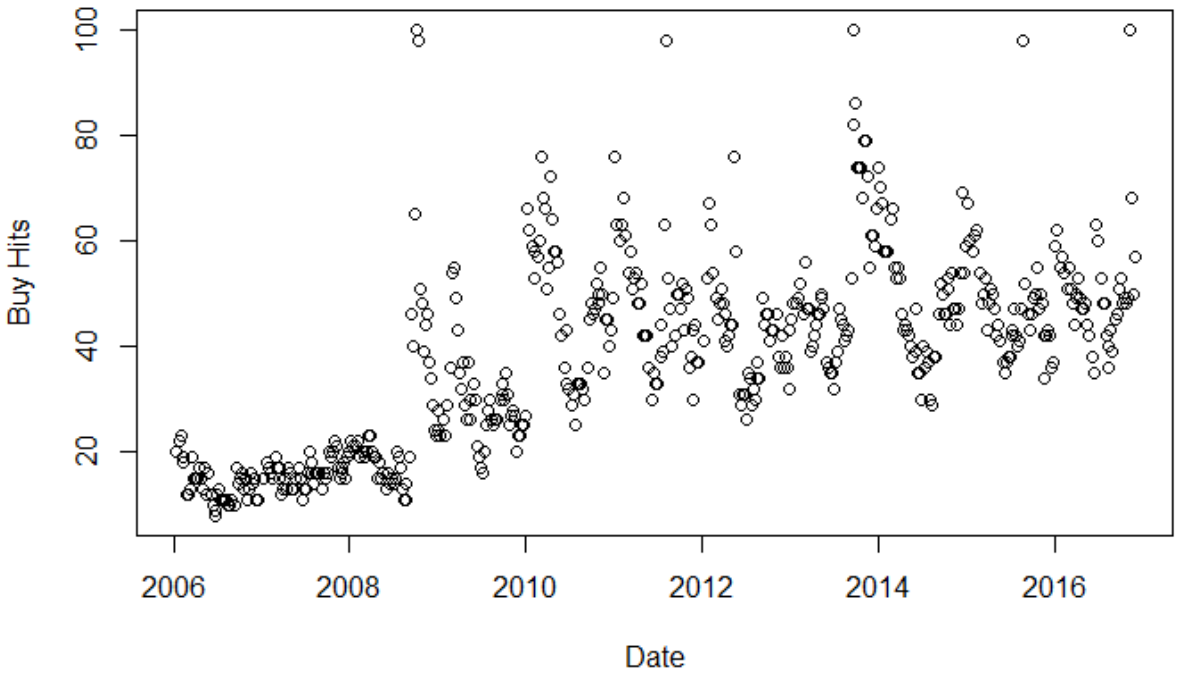
Looking at the keywords that we chose, bear and bull were both keywords that had roughly the same number of hits. There are outliers with bear that have it being search at a higher interval but overall both of the keywords hover around the 10 – 15 hits on a weekly interval. The total number of hits overall vary by a thousand between the two keywords (bear and bull) but they hover roughly around the same number of hits on a regular basis. As the bear hits looked a bit condensed, a logarithm calculation was run on the to see if any findings could be determined with a more spread out view of the information. Looking at the log chart of the bear hits, there was a much lower frequency of the data in the past and it stabilized after 2010.

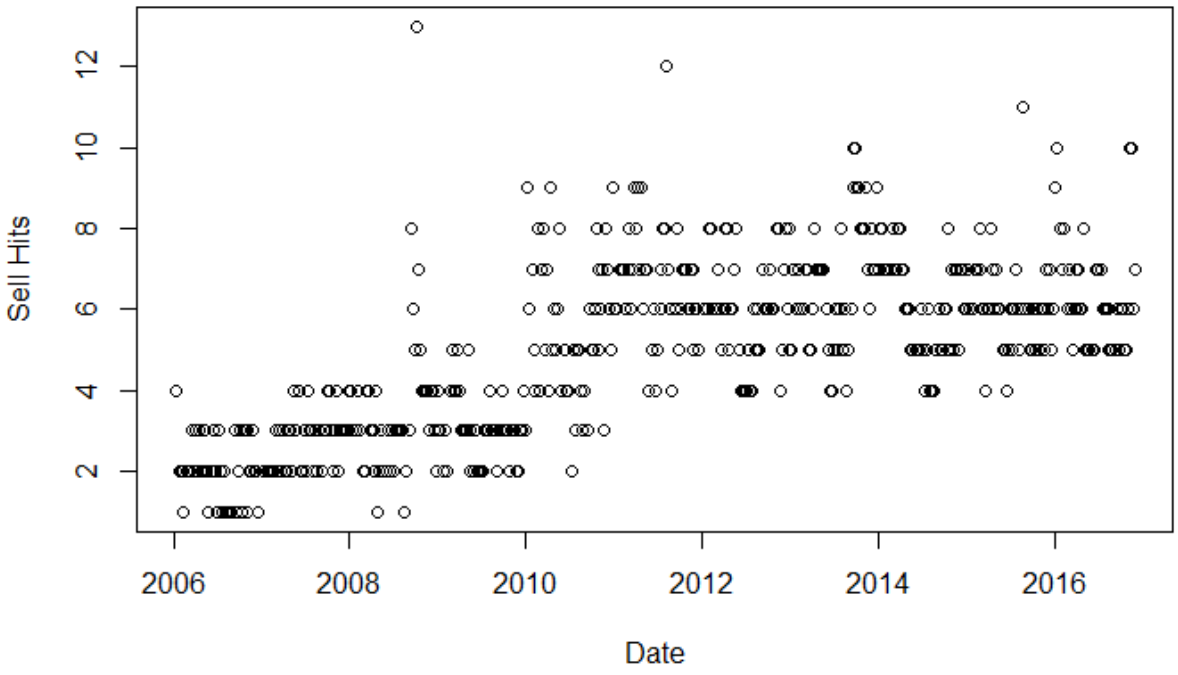
As for the buy and sell keywords, they differ drastically. The buy keyword has a sum total of 19894 where as our sell keyword has 2566. We attribute this to behavior in the users where buying products occurs regularly through Google searches. As you can see from the plots below, buy showed significant searches after 2010 where internet purchases became more prominent. The sell keyword remains the same in terms of search by internal across our 10 years of data. Overall, looking at the keywords that we chose, the majority of them seemed to have shown an increase in searches after 2010.




Bear Hits
10
20
30
40
2006
2008
2010
2012
2014
2016
Date


log(Bear Hits)
1.0
1.5
2.0
2.5
3.0
3.5
2006
2008
2010
2012
2014
2016
Date


Buy Hits
20
40
60
80
100
2006
2008
2010
2012
2014
2016
Date


Sell Hits
2
4
6
8
10
12
2006
2008
2010
2012
2014
2016
Date

With our models, we looked at the overall significance of our keywords and none of them was counted as a significant variable in our models. Our inputs into our models for logistical regression, LDA, QDA, KNN all were adjusted to use the variables that played a significance to our data. Originally, we used all of our keywords as inputs but after viewing our models, we removed them to ensure accuracy of our model. There was one instance where the bull hit had a small significance to the PGR lag 2 returns but the coefficient didn't have enough of a significance to warrant it to include in the model. The significant coefficients within our models were consistently the coefficients from the other stocks and their weekly performance data. The summaries showing the significance of each coefficient is displayed below.

```
Call:
glm(formula = hig_return.d ~ buy_hit + sell_hit + bull_hit +
    bear_hit + hig_hit + pgr_hit + trv_hit + trv_return + pgr_return,
    family = binomial, data = clean.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0232  -0.9328   0.3858   0.9636   2.9537

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.328733   0.500104  -0.657    0.511
buy_hit      0.006563   0.013220   0.496    0.620
sell_hit    -0.015666   0.099184  -0.158    0.874
bull_hit     0.039542   0.065364   0.605    0.545
bear_hit    -0.046340   0.033284  -1.392    0.164
hig_hit      0.011289   0.008849   1.276    0.202
pgr_hit     -0.011103   0.021434  -0.518    0.604
trv_hit     -0.021571   0.034822  -0.619    0.536
trv_return  24.021886   4.577587   5.248 1.54e-07 ***
pgr_return  22.682057   4.457926   5.088 3.62e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 715.69  on 516  degrees of freedom
Residual deviance: 583.81  on 507  degrees of freedom
AIC: 603.81

Number of Fisher Scoring iterations: 5
```

```
Call:
glm(formula = hig_r_Lag1.d ~ buy_hit + sell_hit + bull_hit +
    bear_hit + hig_hit + pgr_hit + trv_hit + trv_return + pgr_return +
    hig_return + trv_r_Lag1 + pgr_r_Lag1, family = binomial,
    data = clean.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8987  -0.9176   0.2987   0.9304   2.1909

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.063e-01  5.166e-01  -0.593  0.55324
```

```
buy_hit      3.031e-05  1.309e-02   0.002  0.99815
sell_hit     1.275e-01  1.026e-01   1.242  0.21412
bull_hit    -6.590e-02  6.485e-02  -1.016  0.30955
bear_hit    -5.111e-03  3.048e-02  -0.168  0.86684
hig_hit      4.057e-03  9.011e-03   0.450  0.65258
pgr_hit     -2.614e-02  2.196e-02  -1.190  0.23390
trv_hit      2.625e-02  3.472e-02   0.756  0.44957
trv_return  -6.392e+00  4.396e+00  -1.454  0.14594
pgr_return   1.814e+01  4.267e+00   4.250 2.14e-05 ***
hig_return  -4.078e+00  1.299e+00  -3.141  0.00169 **
trv_r_Lag1   2.538e+01  4.356e+00   5.825 5.70e-09 ***
pgr_r_Lag1   2.704e+01  4.422e+00   6.114 9.72e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 715.50  on 516  degrees of freedom
Residual deviance: 562.49  on 504  degrees of freedom
AIC: 588.49

Number of Fisher Scoring iterations: 5


Call:
glm(formula = hig_r_Lag2.d ~ buy_hit + sell_hit + bull_hit +
    bear_hit + hig_hit + pgr_hit + trv_hit + trv_return + pgr_return +
    hig_return + trv_r_Lag1 + pgr_r_Lag1 + hig_r_Lag1 + trv_r_Lag2 +
    pgr_r_Lag2, family = binomial, data = clean.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7399  -0.9192   0.3305   0.9324   2.2352

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.7079479  0.5146896  -1.375  0.16898
buy_hit      0.0008308  0.0129797   0.064  0.94896
sell_hit    -0.0101129  0.1011466  -0.100  0.92036
bull_hit    -0.0010023  0.0654182  -0.015  0.98778
bear_hit    -0.0097559  0.0296797  -0.329  0.74238
hig_hit      0.0069793  0.0092031   0.758  0.44823
pgr_hit      0.0124733  0.0221942   0.562  0.57411
trv_hit      0.0117104  0.0353581   0.331  0.74050
trv_return   3.9747450  4.0272894   0.987  0.32367
pgr_return   2.9877427  4.2409809   0.704  0.48113
hig_return  -0.5784615  1.3486690  -0.429  0.66799
trv_r_Lag1  -6.1388060  4.4278638  -1.386  0.16562
pgr_r_Lag1  17.5654730  4.3772954   4.013 6.00e-05 ***
hig_r_Lag1  -4.0101369  1.3832567  -2.899  0.00374 **
trv_r_Lag2  27.1198229  4.4507564   6.093 1.11e-09 ***
pgr_r_Lag2  24.6486051  4.4538173   5.534 3.13e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 715.50  on 516  degrees of freedom
Residual deviance: 565.27  on 501  degrees of freedom
AIC: 597.27

Number of Fisher Scoring iterations: 5


Call:
glm(formula = trv_return.d ~ buy_hit + sell_hit + bull_hit +
    bear_hit + hig_hit + pgr_hit + trv_hit + hig_return + pgr_return,
    family = binomial, data = clean.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6074  -0.9703   0.4571   0.9131   2.9101

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.096650   0.500127  -0.193  0.84676
buy_hit     -0.018693   0.013236  -1.412  0.15789
sell_hit     0.035222   0.098575   0.357  0.72086
bull_hit     0.084417   0.065764   1.284  0.19927
bear_hit    -0.010937   0.030417  -0.360  0.71916
hig_hit      0.009802   0.008878   1.104  0.26953
pgr_hit     -0.005376   0.021779  -0.247  0.80502
trv_hit     -0.017299   0.034568  -0.500  0.61677
hig_return   5.778607   2.046069   2.824  0.00474 **
pgr_return  34.195217   4.481726   7.630 2.35e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 711.27  on 516  degrees of freedom
Residual deviance: 583.30  on 507  degrees of freedom
AIC: 603.3

Number of Fisher Scoring iterations: 5


Call:
glm(formula = trv_r_Lag1.d ~ buy_hit + sell_hit + bull_hit +
    bear_hit + hig_hit + pgr_hit + trv_hit + trv_return + pgr_return +
    hig_return + hig_r_Lag1 + pgr_r_Lag1, family = binomial,
    data = clean.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4477  -0.9866   0.4359   0.9108   3.0319

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.0241873  0.5008041   0.048  0.96148
buy_hit     -0.0070251  0.0131024  -0.536  0.59184
sell_hit     0.0520366  0.1020048   0.510  0.60995
bull_hit     0.0478317  0.0660804   0.724  0.46916
bear_hit    -0.0159401  0.0323110  -0.493  0.62178
```

```
hig_hit      0.0118523  0.0089036   1.331  0.18313
pgr_hit     -0.0006743  0.0214317  -0.031  0.97490
trv_hit     -0.0600788  0.0348824  -1.722  0.08501 .
trv_return  -3.0014863  4.2837003  -0.701  0.48351
pgr_return   3.4029749  4.2838023   0.794  0.42697
hig_return  -1.4631364  1.3266912  -1.103  0.27009
hig_r_Lag1   5.1452273  1.8932794   2.718  0.00658 **
pgr_r_Lag1  35.5180461  4.5730688   7.767 8.05e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 711.27  on 516  degrees of freedom
Residual deviance: 578.53  on 504  degrees of freedom
AIC: 604.53

Number of Fisher Scoring iterations: 5


Call:
glm(formula = trv_r_Lag2.d ~ buy_hit + sell_hit + bull_hit +
    bear_hit + hig_hit + pgr_hit + trv_hit + trv_return + pgr_return +
    hig_return + trv_r_Lag1 + pgr_r_Lag1 + hig_r_Lag1 + hig_r_Lag2 +
    pgr_r_Lag2, family = binomial, data = clean.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4673  -0.9670   0.4180   0.9088   2.8205

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.0731603  0.5048748   0.145  0.88478
buy_hit     -0.0005039  0.0132018  -0.038  0.96955
sell_hit     0.0888196  0.1013784   0.876  0.38096
bull_hit    -0.0001201  0.0653647  -0.002  0.99853
bear_hit    -0.0238301  0.0308334  -0.773  0.43960
hig_hit      0.0024814  0.0092214   0.269  0.78786
pgr_hit      0.0142932  0.0218744   0.653  0.51348
trv_hit     -0.0382949  0.0351106  -1.091  0.27541
trv_return  -6.2212735  4.5011196  -1.382  0.16692
pgr_return  -1.5207545  4.1815533  -0.364  0.71610
hig_return   3.6153177  1.6274056   2.222  0.02632 *
trv_r_Lag1  -3.6358638  4.4578698  -0.816  0.41473
pgr_r_Lag1   4.5016708  4.3920895   1.025  0.30539
hig_r_Lag1  -1.1085351  1.4615628  -0.758  0.44818
hig_r_Lag2   5.8970471  1.9000177   3.104  0.00191 **
pgr_r_Lag2  36.5310019  4.7758501   7.649 2.02e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 711.27  on 516  degrees of freedom
Residual deviance: 572.29  on 501  degrees of freedom
AIC: 604.29
```

```
Number of Fisher Scoring iterations: 5


Call:
glm(formula = pgr_return.d ~ buy_hit + sell_hit + bull_hit +
    bear_hit + hig_hit + pgr_hit + trv_hit + hig_return + trv_return,
    family = binomial, data = clean.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6490  -1.0148   0.4318   0.9901   2.1521

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.6557773  0.4904000  -1.337  0.18115
buy_hit      0.0139931  0.0130752   1.070  0.28453
sell_hit    -0.0393723  0.0984566  -0.400  0.68923
bull_hit     0.0588618  0.0644847   0.913  0.36135
bear_hit     0.0007913  0.0292834   0.027  0.97844
hig_hit     -0.0056201  0.0086155  -0.652  0.51419
pgr_hit     -0.0072910  0.0207639  -0.351  0.72548
trv_hit      0.0301353  0.0339966   0.886  0.37539
hig_return   5.1265889  1.6839034   3.044  0.00233 **
trv_return  32.1642638  4.6330948   6.942 3.86e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 715.09  on 516  degrees of freedom
Residual deviance: 602.44  on 507  degrees of freedom
AIC: 622.44

Number of Fisher Scoring iterations: 5


Call:
glm(formula = pgr_r_Lag1.d ~ buy_hit + sell_hit + bull_hit +
    bear_hit + hig_hit + pgr_hit + trv_hit + trv_return + pgr_return +
    hig_return + hig_r_Lag1 + trv_r_Lag1, family = binomial,
    data = clean.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.3232  -0.9997   0.4154   0.9571   2.3119

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.007380   0.496657  -2.028 0.042527 *
buy_hit     -0.009231   0.012800  -0.721 0.470823
sell_hit     0.046969   0.100473   0.467 0.640156
bull_hit     0.100288   0.065518   1.531 0.125848
bear_hit    -0.016212   0.031077  -0.522 0.601903
hig_hit      0.005597   0.008807   0.635 0.525133
pgr_hit     -0.017949   0.021068  -0.852 0.394221
trv_hit      0.041002   0.034310   1.195 0.232078
trv_return   1.432422   4.328273   0.331 0.740686
```

```
pgr_return  -9.327380   4.163510  -2.240 0.025074 *
hig_return  -0.064366   1.280935  -0.050 0.959924
hig_r_Lag1   6.265734   1.879168   3.334 0.000855 ***
trv_r_Lag1  31.416408   4.695586   6.691 2.22e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 715.09  on 516  degrees of freedom
Residual deviance: 592.72  on 504  degrees of freedom
AIC: 618.72

Number of Fisher Scoring iterations: 5


Call:
glm(formula = pgr_r_Lag2.d ~ buy_hit + sell_hit + bull_hit +
    bear_hit + hig_hit + pgr_hit + trv_hit + trv_return + pgr_return +
    hig_return + trv_r_Lag1 + pgr_r_Lag1 + hig_r_Lag1 + hig_r_Lag2 +
    trv_r_Lag2, family = binomial, data = clean.data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0165  -0.9617   0.3650   0.9365   2.4238

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.825586   0.504443  -1.637 0.101708
buy_hit      0.002997   0.012545   0.239 0.811185
sell_hit    -0.145715   0.099298  -1.467 0.142254
bull_hit     0.133957   0.066646   2.010 0.044434 *
bear_hit     0.031176   0.033240   0.938 0.348291
hig_hit      0.008432   0.009092   0.927 0.353695
pgr_hit     -0.004514   0.021290  -0.212 0.832092
trv_hit     -0.018069   0.034776  -0.520 0.603356
trv_return   6.880198   4.034482   1.705 0.088129 .
pgr_return   1.691949   4.413776   0.383 0.701472
hig_return  -0.298902   1.497510  -0.200 0.841794
trv_r_Lag1   0.332006   4.132108   0.080 0.935961
pgr_r_Lag1  -9.555694   4.319927  -2.212 0.026966 *
hig_r_Lag1   0.034466   1.271332   0.027 0.978372
hig_r_Lag2   6.211118   1.796399   3.458 0.000545 ***
trv_r_Lag2  32.814171   4.835704   6.786 1.15e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 714.85  on 516  degrees of freedom
Residual deviance: 579.62  on 501  degrees of freedom
AIC: 611.62

Number of Fisher Scoring iterations: 5
```

Conclusion

Between the various non classification and classification models used in this analysis, we were able to accurately predict both actual return values (the lowest significant MSE was .0024 for PGR when using regression tree on the Lag 1 return) and return directions (the highest prediction rate was 78.76% for TRV when using KNN on the No Lag return). Using our combined methods, we accurately predicted No Lag, Lag 1, and Lag 2 returns for all three stocks. Upon further analysis, however, we could not prove our hypothesis. The most significant variables in our prediction models were actually the returns of similar stocks and the returns for the measured stock in previous weeks. A majority of Google trend data was insignificant in our final prediction models. When predicting No Lag returns between the tree stocks, no Google trend data was significant. For Lag 2 returns, only four Google trend variables between all stocks were significant and those were all ticker abbreviation searches. Only Lag 2 return predictions for PGR included significant Google key phrase variables. In the Lag 2 non classification prediction for PGR, "Sell Stocks" and "Bull Market" were significant.

The three companies had a consistent pattern in terms of the trend lines for their mentions in gtrendsR, but there was little correlation between their performance and the amount of hits in gtrendsR. HIG seemed to have higher average returns overall, particularly in 2009, while TRV had the biggest percentage of weeks with upward movement in the stock. The returns on the three stocks were correlated, as were the number of weeks with upward price movement. Within a stock, there was strong correlation between the return and the number of weeks when the stock price moved up. Interestingly, there were not strong correlations between the price of the return on the stock and the return in subsequent weeks, indicating week-to-week volatility in the return. The non-correlation methods did, however, find a statistically significant relationship between a stock's performance and its return in subsequent weeks. This was also true for the movement up or down on a week-to-week basis.

The discrepancy between the pattern of the weekly return and the number of weeks with upward movement is most likely due to the fact that there were weeks with unusually large or small returns compared to the other weeks, and that influenced the average. The difference in the pattern between the average weekly return and the percentage of weeks when the stock moved up or down indicates that there will be different strategies depending on the investor's goals. If interested in making profits from rapid turnover, the investor would do well to focus on short-term movements in the price, while a long-term investor would do better looking at the average return over an extended period of time.

With the analysis of the raw data, there were no correlations between the information returned from Google Trends and the performance of each stock. Within each plot between our keywords and the returns of each stock, the number of hits didn't increase the return or decrease the return of any stock performance. The performance remained the same regardless of the number of hits for a keyword. This was further proven out by our models and their summaries indicating that significant coefficients in our models.

Looking at the insurance company performance, we can see there were periods of volatility in 2008, 2009, and 2011. Those periods are represented by economic events during those time periods and are expected in the data. All three stocks followed had these periods of volatility but in varying ranges of performance. As the stocks showed this correlation, our models indicated their level of significance as coefficients. While we found that the keywords did not affect the stock's performance, we did see that stocks affected each other's performance and could be used as a good indication. When starting the

analysis on the project, we expected the keywords to contribute to the performance of a stock but we discovered that not to be true and that the market has its own correlation patterns within each industry. The three stocks we choose to analyze all seemed to follow the industry trend of performance.

References

Google. (n.d.). Google Trends. *Normalized hit data 2006 to 2016*.

How Trends Data is Adjusted. (2017). Retrieved from Google: https://support.google.com/trends/answer/4365533?hl=en

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An Introduction to Statistical Learning with Applications in R.* New York: Springer.

Package 'gtrendsR'. (2016, 11 03). Retrieved from cran.r-project.org: https://cran.r-project.org/web/packages/gtrendsR/gtrendsR.pdf

Package 'stockPortfolio'. (2015, 02 20). Retrieved from cran.r-project: https://cran.r-project.org/web/packages/stockPortfolio/stockPortfolio.pdf

Appendix A: R Packages Used

- library("gtrendsR")
- library("quantmod")
- library("stockPortfolio")
- library("lubridate")
- library("zoo")
- library("ggplot2")
- library("class")
- library("readr")
- library("data.table")
- library("plyr")
- library("leaps")
- library("tree")
- library("MASS")
- library("class")
- library(randomForest)
- library(dplyr)
- library(reshape2)

Appendix B: Attached Data Files

1. **FE582_Group_Project_R_Code_Master**

The master base code file for this project. This includes the initial data pulling, cleaning up, and non classification and classification analysis.

2. **Cleandata.rda**

The final cleaned data set as it appears in the master code file.

3. **FE582_Group_Project_SML_Updates_20170502.r**

A modified version of the master code to adjust variable names for use in classification and non classification plots.

4. **FE582_Group_Project_SML_Updates_20170428_Classification_Plots.R**

Classification method plots created using variables defined in file 3.

5. **FE582_Group_Project_SML_Updates_20170502_NonClass_Plots.R**

Non classification method plots using variables defined in file 3.

6. **FE582_Group_Project_SML_Updates_20170502_Classification_Plots_Residuals.R**

Non classification residual plots using variables defined in file 3.

7. **FE582_Group_Project_R_Code_MASTER_5.4.17 SS.r**

Plots, tables, and analysis used in the comparison by stocks section.

8. **FE582_Group_Project_R_Code_MASTER_5.5.17.r**

Plots, tables, and analysis used in the comparison by Google keyword and phrase section

9. **Correlations.xlsx**

Table of correlations between model variables and stock No Lag, Lag 1 and Lag 2 return values. Used in the comparison by stocks section.