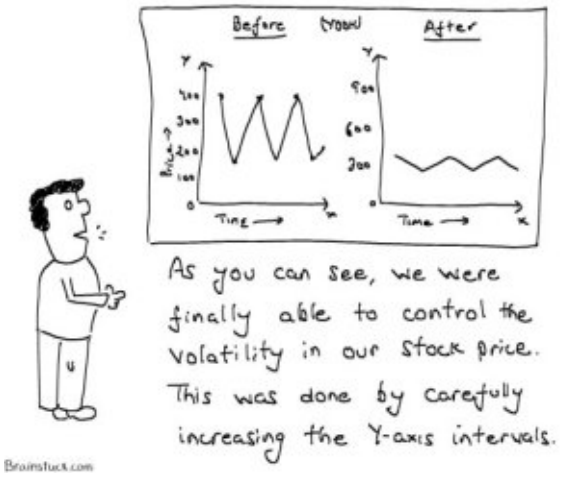
# Predicting Financial Market Volatility with Google Domestic Trends

*Wei Wu*

*November 14, 2018*

## Contents





# Introduction

When stock traders talk about "the market", they are often referring to the movement of the largest publicly traded stocks. Indices are metrics used by the financial world to track markets, or subsets of the market, in order to determine the aggregate movement of the selected companies. While there are many ways to track this so called "market", the most commonly used metric for measuring the performance of the American stock market is the S&P 500.

For this project, I propose to predict S&P 500 market volatility on a daily granularity. Volatility is the range of price change a security (a tradable financial instrument) experiences over a given period of time. If the price stays relatively stable, the security has low volatility. A highly volatile security is one that hits new highs and lows, moves erratically, and experiences rapid increases and dramatic falls.

The S&P 500 market data, publicly available on Yahoo Yah, comprises of high, low, close, open (HLCO), adjusted closed, and volume of each trading day. A daily volatility is estimated using HLCO with the following equations:

$u = log(Hi/Op), d = log(Lo/Op), c = log(Cl/Op)$

$\sigma = 0.511(u-d)^2 - 0.019[c(u+d) - 2ud] - 0.383c^2$

HLCO, adjusted closed, and volume will serves as our model features. In addition, inspisred by Xiong et al. [2015], I will use Google Trend, Google's collection of search queries popularities, as indicators of macroeconomics to supplement the financial features. Rolling linear regressions and LASSO regression with lag by one day will be applied to predict daily volatility. Models performnace will be compared visually and quantatively.

## Data Collection and Data Wrangling

The Yahoo Finance data and Google Trends data are public available online. However, for this project, we obtained the data from the following Github Repo: https://github.com/philipperemy/stock-volatility-google-trends. Goole has stopped providing daily search trends data for period larger than 90 days. A convinient Python script to scrap Google Trends data can be found at this Github Repo: https://github.com/GeneralMills/pytrends.

The collected SP500 data contains Open, High, Low, Close, and Volume for each market day. There are in total 2756 data points, ranging from 2006-10-23 to 2017-10-05. After calculating the daily volatility *sigma*, our daily volatility is typically at the scale of $10^{-6}$. For this project, we later scaled daily volatility by $10^6$. We also later normalized trade volume by comuting the z-score.

Table 1: Tickers Data

| | Date | Open | High | Low | Close | Volume | sigma |
|---|---|---|---|---|---|---|---|
| 10 | 2017-09-22 | 249.05 | 249.63 | 249.02 | 249.44 | 51214032 | 2.00e-06 |
| 11 | 2017-09-21 | 249.88 | 249.98 | 249.18 | 249.39 | 48211398 | 3.60e-06 |
| 12 | 2017-09-20 | 250.07 | 250.19 | 248.92 | 250.06 | 59574083 | 1.31e-05 |
| 13 | 2017-09-19 | 250.00 | 250.07 | 249.60 | 249.97 | 47108148 | 1.80e-06 |
| 14 | 2017-09-18 | 249.61 | 250.12 | 249.28 | 249.72 | 46235238 | 5.60e-06 |
| 15 | 2017-09-15 | 248.69 | 249.29 | 248.57 | 249.19 | 95432382 | 2.60e-06 |

The Google Trends data indicates relavtive search intensity for a certain keyword during a period of time. The trend intensity are normalized. We included 24 Goolge Trends, listed below:

Table 2: Google Trends & Abbreviation

| Trend | Abbreviation |
|---|---|
| advertising & marketing | advert |
| air travel | airtvl |
| auto buyers | autoby |
| auto financing | autofi |
| business & industrial | bizind |
| bankruptcy | bnkrpt |
| computers & electronics | comput |
| credit cards | crcard |
| durable goods | durble |
| education | educat |
| finance & investing | invest |
| financial planning | finpln |
| furniture | furntr |
| insurance | insur |
| jobs | jobs |
| luxury goods | luxury |
| mobile & wireless | mobile |
| mortgage | mrtge |
| real estate | rlest |
| rental | rental |
| shopping | shop |
| small business | smallbiz |
| travel | travel |
| unemployment | unempl |

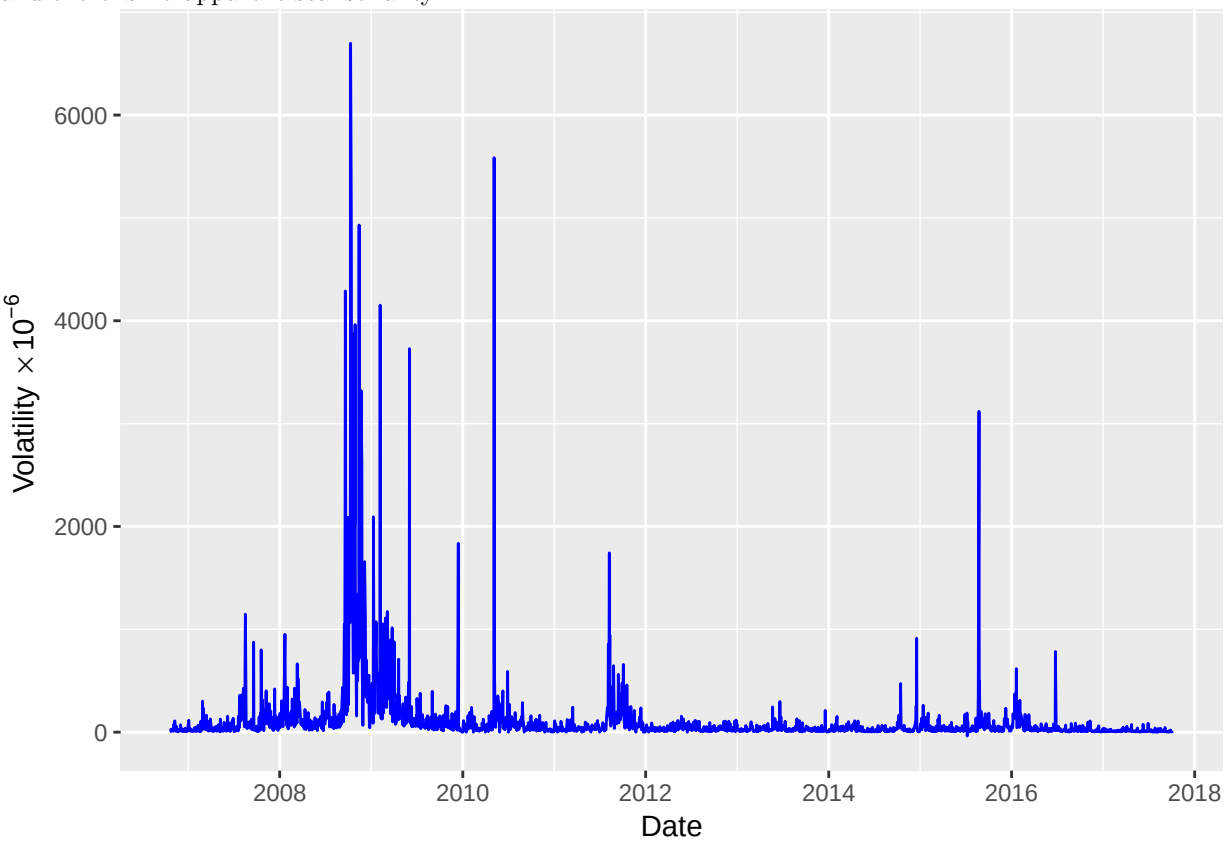
| Trend | Abbreviation |
|---|---|

Table 3: Google Trends Data

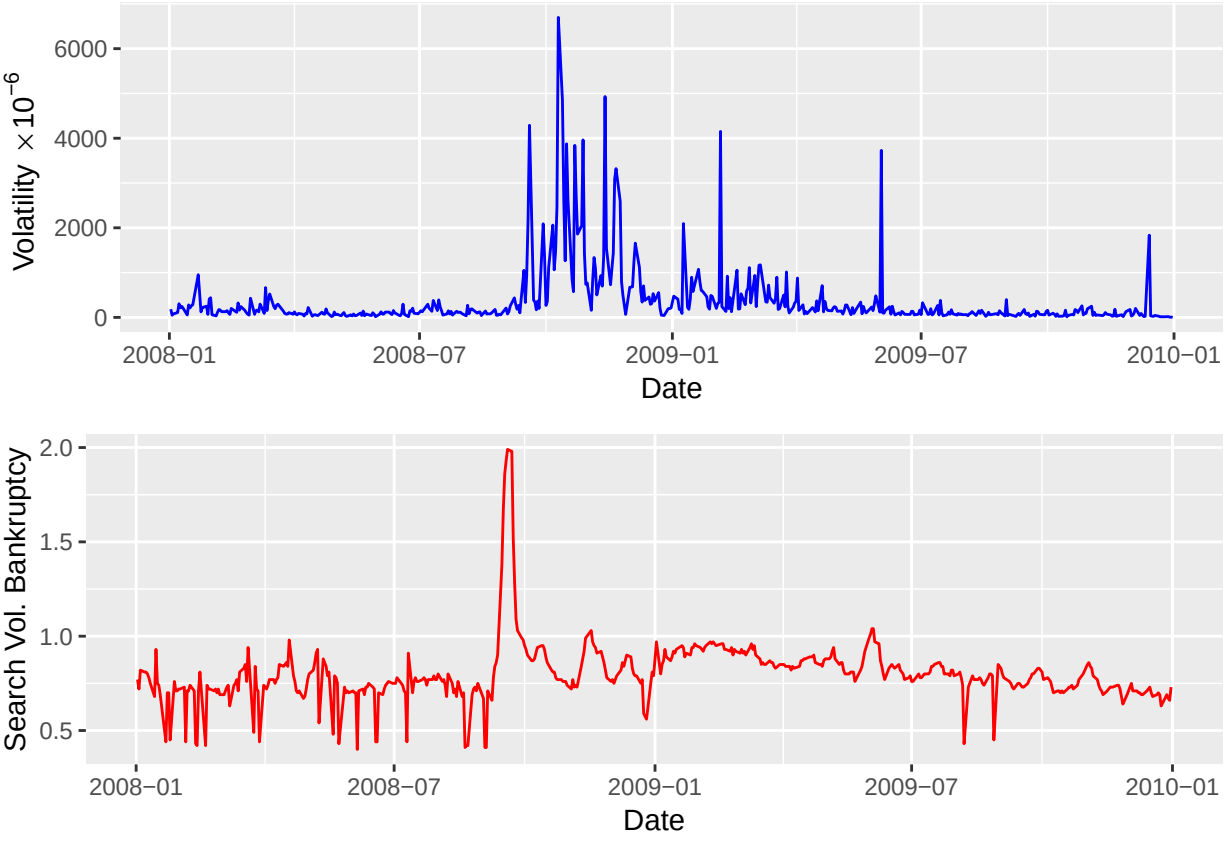
| Date | advert | airtvl | autoby | autofi | bizind | bnkrpt | comput |
|---|---|---|---|---|---|---|---|
| 2006-10-22 | 0.71 | 0.97 | 0.79 | 0.73 | 1.60 | 0.58 | 1.03 |
| 2006-10-23 | 0.71 | 0.97 | 0.79 | 0.91 | 1.60 | 0.58 | 1.03 |
| 2006-10-24 | 0.70 | 0.97 | 0.79 | 0.90 | 1.60 | 0.59 | 1.04 |
| 2006-10-25 | 0.71 | 0.96 | 0.80 | 0.89 | 1.59 | 0.55 | 1.04 |
| 2006-10-26 | 0.71 | 0.96 | 0.80 | 0.87 | 1.59 | 0.58 | 1.03 |

## Exploratory Data Analysis

We first examine our entire volatility data. We can see from the plot that the volatility is highly non-stationary, and there is no apparent seansonality.



Next we visually examine the correlation of market volatility and Goolge Trend. Below are the plots for volatility and search volume for keyword “Bankruptcy”" between years 2008 and 2010. As the market became volatile, the search volume for “Bankruptcy” surged.



# Modeling

## Spliting Data

We first subset our data set into Training and Testing. Training data ranges from 2006-10-23 to 2017-10-05, and Testing ranges from 2016-01-02 to 2017-10-05. The Training data was further subset to have a portion Validation, ranging from 2015-01-02 to 2016-01-01. The Validation data is used by LASSO to select an optimal penalty constant. The Testing data set is held out to compare models. A rolling scheme is applied when fitting the data. The R package "caret" comes in handy for performing rolling regression.

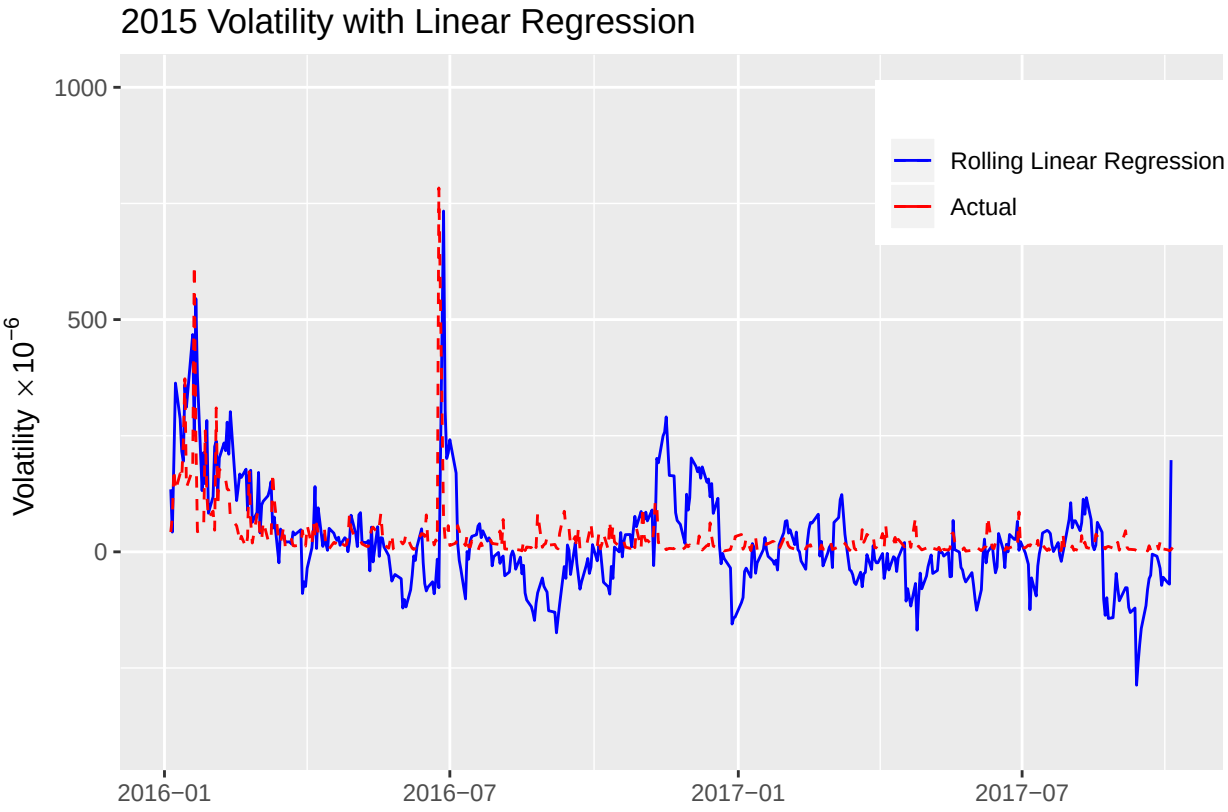
## Rolling Linear Regression

We first train a linear regression model with no intercept on the the entire training set, and test our linear regression model on testing data set. The significant coefficients, with each's p-value smaller than 0.05, is described in the following table.

Table 4: Significant Coeffcients

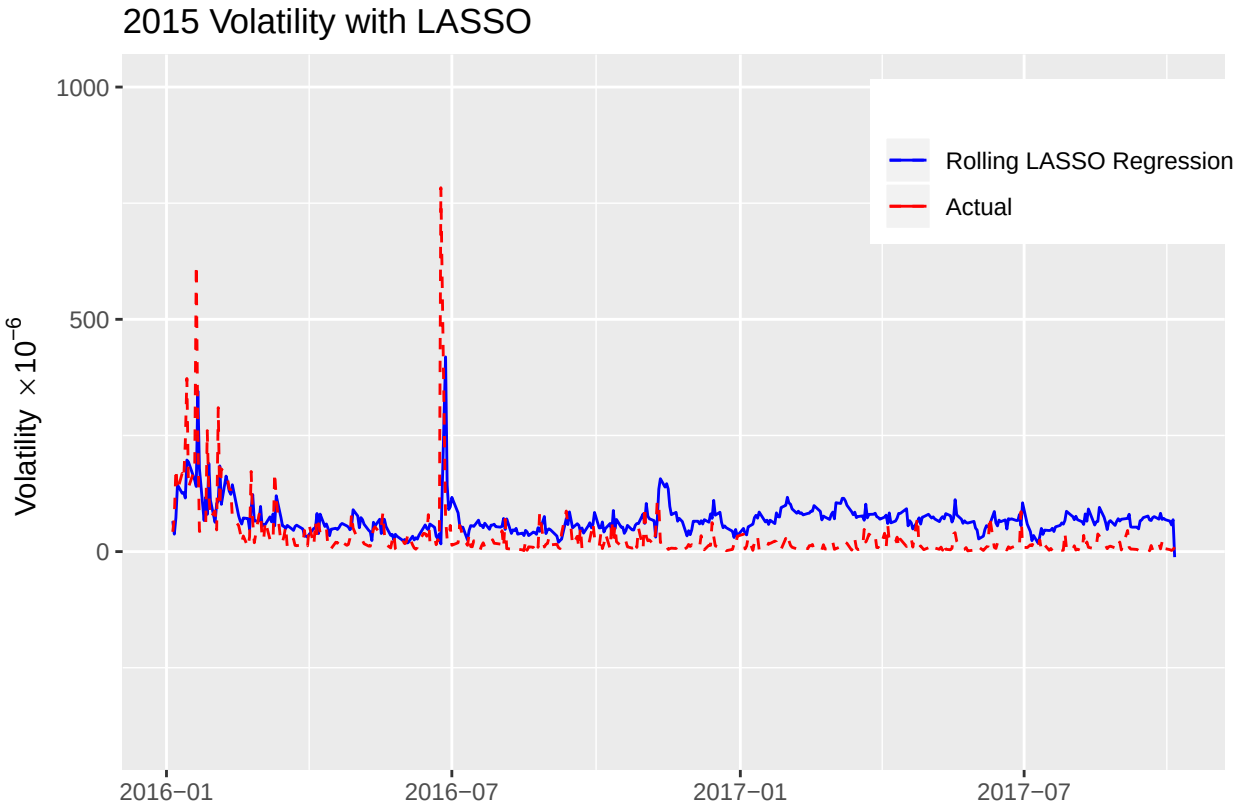
| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| Open_Lag1 | -38.5229776 | 9.1575237 | -4.206702 | 0.0000267 |
| Low_Lag1 | 26.5416828 | 8.4791868 | 3.130216 | 0.0017653 |
| Close_Lag1 | -2.6002630 | 0.7176076 | -3.623516 | 0.0002959 |
| sigma_Lag1 | 0.3669127 | 0.0234253 | 15.663100 | 0.0000000 |
| advert_Lag1 | -662.5757800 | 229.1524734 | -2.891419 | 0.0038654 |
| bnkrpt_Lag1 | 198.2234439 | 63.7043802 | 3.111614 | 0.0018800 |
| comput_Lag1 | -586.0776511 | 206.6148933 | -2.836570 | 0.0045938 |
| furntr_Lag1 | 600.5104399 | 248.8655294 | 2.412992 | 0.0158878 |
| invest_Lag1 | 982.7352501 | 107.0335178 | 9.181565 | 0.0000000 |
| jobs_Lag1 | 627.3233936 | 119.8630835 | 5.233666 | 0.0000002 |
| shop_Lag1 | 258.7715583 | 85.3294101 | 3.032619 | 0.0024473 |
| smallbiz_Lag1 | -621.9196982 | 314.3925104 | -1.978163 | 0.0480109 |
| unempl_Lag1 | -144.6565010 | 36.6249721 | -3.949669 | 0.0000802 |
| return_Lag1 | -5337.6806762 | 503.2758815 | -10.605874 | 0.0000000 |

For a visual examiniation of the prediction, we plot the prediction results versus the actual sigma of the testing data. Even though linear regression model does a good job at predicting near the high volatilty regions, e.g. around 2016-02, it performs poorly when the actual volatlity is low. We also observe that for the high volatility day, the prediction of high volatility often lags by a small time window.



## Rolling LASSO

Next we train and validate a LASSO linear regression model with no intercept. The penalization constant $\lambda$ is selected so that the RMSE (root-mean-squared-erro) is the smallest on the validation data set. After $\lambda$ is selected, we fit a LASSO model on the testing set, with the selected penalty constant. Visually, the LASSO regression model does a better job at flecting the overall trend of the market volatility, even though near the high volatility region the model underperforms. Again we observe that the prediction of high volatility lags by a couple of days.




## Features Selections by LASSO

The LASSO regression shrinks the coeffcients of most features to zeros. For the final model, the selected feautures and their coeffcients are described in the following table. The historical volatility (sigma), unsurprisingly, has been selected by LASSO. Other interesting features are daily return and trend investing, which also appeared in the feature selection by Xiong et al. [2015].

Table 5: Selected Features by LASSO

| features | coefs |
|---|---|
| Close_Lag1 | -0.4957916 |
| sigma_Lag1 | 0.4065497 |
| invest_Lag1 | 234.8123290 |
| return_Lag1 | -1292.9102592 |

## Models Comparision

To compare our mode quantatively, we compute two scoring metrics, RMSE and MAE(mean-absolute-error). The LASSO model performs better than the Linear Regression model in both terms.

Table 6: Daily Realized Variance Prediction Performance

| | RMSE | MAE |
|---|---|---|
| LASSO | 71.27 | 50.4 |
| Linear Regression | 112.98 | 73.4 |

## Summary and Future Work

This project indicates promise in applying rolling regression scheme to predcit market volatility. Future work awaits to be done on using different time lags when fitting the models. Xiong et al. [2015] provides an optimal normalization observation and scheme, which could be studied further as a continuation fo this project. Another observation worth noting is that our prediction of high volatility often lags by a small time window. This was also reflected in the prediction results of Xiong et al. [2015]. However at this point we are unsure if this is a characteristics of our models.

## Reflection

Predicting any financial market quantity is a daunting task, given the highly volatile nature of the markets. Google search trends have served as our macroeconomics indicator in this project, yet there are other data from financial instruments that could be used, such as EuroDollars, Nasdaq, etc.

# Bibliography

# Appendix

```
knitr::opts_chunk$set(echo = F)
knitr::opts_chunk$set(warning = F)
#knitr::opts_knit$set(root.dir=normalizePath('~/getDocuments/2018Fall/STAT410/project'))

# pacman is a neat little package that automates packages installation and importing
if (!require("pacman")) install.packages("pacman")

pacman::p_load('stringr',
               'tinytex',
               'zoo',
               'xtable',
               'knitr',
               'GGally',
               'ggplot2',
               'astsa',
               'reshape',
               'scales',
               'gridExtra',
               'forecast',
               'tseries',
               'bibtex',
               'rpivotTable',
               'party',
               'DataCombine',
               'MLmetrics',
               'quantmod',
               'data.table',
               'glmnet',
               'knitr',
               'caret',
               'anytime',
               'tools',
               'latex2exp')
knitr::include_graphics("./controlling-volatility.jpg")
# Load Market Data
tickers <- read.csv("./all data/SP500.csv",stringsAsFactors = FALSE)
tickers$Date <- as.Date(tickers$Date,"%d-%B-%y")
tickers$Open <- as.numeric(tickers$Open)
tickers$Close <- as.numeric(tickers$Close)
tickers$High <- as.numeric(tickers$High)
tickers$Low <- as.numeric(tickers$Low)
# Compute sigma
tickers$u <- log(tickers$High/tickers$Open)
tickers$d <- log(tickers$Low/tickers$Open)
tickers$c <- log(tickers$Close/tickers$Open)
tickers$sigma <- 0.511*(tickers$u-tickers$d)^2 - 0.019*(tickers$c*(tickers$u+tickers$d)-2*tickers$u*tick
tickers <- tickers[, !(names(tickers) %in% c("c","u","d"))]
kable(tickers[10:15,], caption = "Tickers Data")
```

```r
trend_abbr <- matrix(c(
'advertising & marketing',
'air travel',
"auto buyers",
"auto financing",
"business & industrial",
"bankruptcy",
"computers & electronics",
"credit cards",
"durable goods",
"education",
"finance & investing",
"financial planning",
"furniture",
"insurance",
"jobs",
"luxury goods",
"mobile & wireless",
"mortgage",
"real estate",
"rental",
"shopping",
"small business",
"travel",
"unemployment",
"advert",
"airtvl",
"autoby",
"autofi",
"bizind",
"bnkrpt",
"comput",
"crcard",
"durble",
"educat",
"invest",
"finpln",
"furntr",
"insur",
"jobs",
"luxury",
"mobile",
"mrtge",
"rlest",
"rental",
"shop",
"smallbiz",
"travel",
"unempl"), ncol =2)

colnames(trend_abbr) <- c("Trend","Abbreviation")
kable(trend_abbr, caption = "Google Trends & Abbreviation")
# Load Google Trends Data
```

```
myFiles <- file_path_sans_ext(list.files(path = "./all data/", pattern = "*.csv"))
L <- setdiff(myFiles, c("SP500"))

O = lapply(L, function(x) {
           DF <- read.csv(paste("./all data/",x,".csv", sep =""),stringsAsFactors = FALSE)
           #DF$Date <- as.character(CAN$Date)
           DF$Date <- as.Date(DF$Date, format ="%d-%B-%y")
           DF <- DF[c("Date", "Close")]
           colnames(DF) <- c("Date", x)
           #DF_Merge <- merge(all.dates.frame, CAN, all = T)
           #DF_Merge$Bid.Yield.To.Maturity <- NULL
           return(DF)})
all_trends <- Reduce(function(x,y) merge(x,y,by="Date"),O)


# Get all Data
all_data <- merge(tickers, all_trends, by="Date", all.x=TRUE)
dates <- all_data$Date
#format(all_data$Date, format = "%B %d %Y")

# Fill Missing Value
all_data[is.na(all_data)] <- 0

# Compute Return
all_data$return <- c(0, (all_data$Close[-1] - all_data$Close[-length(all_data$Close)]) / all_data$Close
# Scale sigma
sigma_sclae <- 10^6
all_data$sigma <- all_data$sigma*sigma_sclae

# normalize volume
all_data$Volume <- (all_data$Volume - mean(all_data$Volume))/sd(all_data$Volume)
kable(all_trends[1:5,1:8], caption = "Google Trends Data")
# Function used to lag data
lag_data <- function(df, lags){
  sigma <- df$sigma[-c(1:lags)]
  lagged_data <- as.data.frame(sigma)
  for (lag in 1:lags) {
    x <- df[-c((0:(lags - lag)), (nrow(df) - lag + 1):nrow(df)),]
    colnames(x) <- paste0(colnames(x), "_Lag", as.character(lag))
    lagged_data <- as.data.frame(cbind(lagged_data, x))
  }
  lagged_data[,] <- lapply(lagged_data[,], as.character)
  lagged_data[,] <- lapply(lagged_data[,], as.numeric)
  lagged_data <- Filter(function(x)!all(is.na(x)), lagged_data)
  return(lagged_data)
}

daily_sigma <- ggplot(all_data, aes(x = Date, y = sigma)) + geom_line(color = "blue") + ylab(TeX("Volati
daily_sigma
daily_sigma <- ggplot(subset(all_data, Date < as.Date("2010-01-01") & Date > as.Date("2008-01-01")), aes
bankrupt_trend <- ggplot(subset(all_data, Date < as.Date("2010-01-01") & Date > as.Date("2008-01-01")),

#daily_sigma
```

```r
grid.arrange(daily_sigma, bankrupt_trend, nrow = 2, ncol=1)
training <- subset(all_data, Date <= as.Date("2015-01-01"))
validation <- subset(all_data, Date >= as.Date("2015-01-01") & Date <= as.Date("2016-01-01"))
testing <- subset(all_data, Date >= as.Date("2016-01-01"))
intcpt = 0 # change this to 1 to include intercept
num_of_lags <- 1
training_and_validation <- rbind(training,validation)

fitControl_fixedWindow <- trainControl(method = "timeslice",
                               initialWindow = dim(training_and_validation)[1],
                               horizon = 1,
                               fixedWindow = TRUE,
                               savePredictions = T)

rolling_LM_Fit <- train(sigma ~ ., data = lag_data(all_data, num_of_lags),
                 method = "lm",
                 trControl = fitControl_fixedWindow,
                 tuneGrid  = expand.grid(intercept = intcpt))

rolling_LM_Model <- rolling_LM_Fit$finalModel
rolling_LM_pred <- rolling_LM_Fit$pred$pred
#rolling_LM_Fit
lm_summary <- summary(rolling_LM_Model)
sig_coef <- lm_summary$coefficients[which(lm_summary$coefficients[,4] < 0.05),]
kable(sig_coef, caption = "Significant Coeffcients")
lm_plot <- ggplot() +
  geom_line(aes(testing$Date[-(1:num_of_lags)],
                rolling_LM_pred,
                colour = "Rolling Linear Regression")) +
  geom_line(aes(testing$Date[-(1:num_of_lags)],
                testing$sigma[-(1:num_of_lags)],
                colour = "Actual"
                ),linetype=2) +
  ylab(TeX("Volatility $\\times 10^{-6$} ")) +
  xlab("") +
  ggtitle("2015 Volatility with Linear Regression") +
  scale_colour_manual("",
                      breaks = c("Rolling Linear Regression",
                                 "Actual"),
                      values = c("red","blue")) +
  theme(legend.position = c(0.85,0.85)) +
  ylim(-400,1000)

lm_plot
lagged_training <- lag_data(training,num_of_lags)

fitControl_fixedWindow <- trainControl(method = "timeslice",
                               initialWindow = dim(training)[1],
                               horizon = 1,
                               fixedWindow = TRUE,
                               savePredictions = T)

rolling_LASSO_Fit <- train(sigma ~ ., data = lag_data(training_and_validation, num_of_lags),
```

```
                  method = "glmnet",
                  trControl = fitControl_fixedWindow,
                  tuneGrid = data.frame(alpha = 1, lambda = seq(0,100,1)),
                  intercept=intcpt)

rolling_LASSO_Model <- rolling_LASSO_Fit$finalModel
# LASSO predictions
new_fitControl_fixedWindow <- trainControl(method = "timeslice",
                               initialWindow = dim(training_and_validation)[1],
                               horizon = 1,
                               fixedWindow = TRUE,
                               savePredictions = T)

rolling_LASSO_best_Fit <- train(sigma ~ ., data = lag_data(all_data, num_of_lags),
                  method = "glmnet",
                  trControl = new_fitControl_fixedWindow,
                  tuneGrid = data.frame(alpha = 1, lambda = rolling_LASSO_Fit$bestTune$lambda),
                  intercept=intcpt)

rolling_LASSO_pred <- rolling_LASSO_best_Fit$pred$pred

lasso_plot <- ggplot() +
  geom_line(aes(testing$Date[-(1:num_of_lags)],
                rolling_LASSO_pred,
                colour = "Rolling LASSO Regression")) +
  geom_line(aes(testing$Date[-(1:num_of_lags)],
                testing$sigma[-(1:num_of_lags)],
                colour = "Actual"),linetype=2) +
  ylab(TeX("Volatility $\\times 10^{-6$} ")) +
  xlab("") +
  ggtitle("2015 Volatility with LASSO") +
  scale_colour_manual("",
                      breaks = c("Rolling LASSO Regression",
                                 "Actual"),
                      values = c("red","blue")) +
  theme(legend.position = c(0.85,0.85)) +
  ylim(-400,1000)

lasso_plot
lasso_coeff = coef(rolling_LASSO_Model, s = rolling_LASSO_Fit$bestTune$lambda)
myResults <- data.frame(
  features = lasso_coeff@Dimnames[[1]][ which(lasso_coeff != 0 ) ], #intercept included
  coefs    = lasso_coeff              [ which(lasso_coeff != 0 ) ]  #intercept included
)
kable(myResults, caption = "Selected Features by LASSO")
nonzero_coeff_names = as.character(myResults$features)
testing_actual <- lag_data(testing,num_of_lags)$sigma
accuracy_lasso <- forecast::accuracy(rolling_LASSO_pred, testing_actual)
accuracy_lm <- forecast::accuracy(rolling_LM_pred, testing_actual)
daily_accuracy <- as.data.frame(rbind(accuracy_lasso,
                                      accuracy_lm))
rownames(daily_accuracy) <- c("LASSO", "Linear Regression")
kable(round(daily_accuracy, 2)[-c(1,4:5)], caption = "Daily Realized Variance Prediction Performance")
```

# References

Ruoxuan Xiong, Eric P Nichols, and Yuan Shen. Deep learning stock volatility with google domestic trends. *arXiv preprint arXiv:1512.04916*, 2015.