Can Sentiment Analysis of financial textual media reveal an association between stock volatility and market sentiment?

Arsalan Khan

9/22/2021

Contents

Abstract

Over-utilization of market and accounting data over the last couple of decades by financial analysts has led to portfolio saturation in which all the major financial investment firms converge to more or less the same portfolio strategy thereby driving down returns. Recent advances in Natural language processing have allowed for as expansion of the types of data available to analysts for predicting market volatility. Most of the relevant data for explaining stock market volatility is now found in an unstructured format in the form of

textual data such as financial articles, tweets, company performance reports and minutes of board meetings among others. As a result a new sub-field of natural language processing, known as Natural language based financial forecasting, has emerged and promises to allow fund managers and individual investors to immediately change portfolio strategies as soon as there is any variation in market sentiment. This study aims to test the assumption that market sentiment is correlated with the market price of a stock. This is done by using a tidytext approach towards sentiment analysis of financial data and then using a Bayesian regression framework to find a correlation between price and sentiment.

Introduction

Predicting stock market volatility has always been a very challenging problem. Due to the highly stochastic nature of the markets, stock movement prediction has for long been considered an intractable problem *(Malkiel 1999)*. There are two dominant schools of thought on this matter. One side believes in the efficient market hypothesis which states that the market price of a stock/security reflects all the possible information one can know about the stock to accurately determine it's price. In other words this means that there is no hidden information that hasn't been accounted for in determining the value/price of stock/security and hence there is no possibility of excess returns beyond the market rate. The other side doesn't believe in this hypothesis and instead asserts that stocks can and do deviate from their fair market values. The opponents of the efficient market hypothesis believe that turbulence and sudden price shocks in the financial markets are evidence of stocks/securities being overvalued and thus deviating from their fair values. One way in which markets can be inefficient is through information asymmetry. Active investors seek to find undervalued stocks by finding information about factors that can affect stock prices such as news releases on earnings and profits, new product launches, employee layoffs and change in management among other factors. Traditionally financial analysts have had to restrict their technical analysis to structured tabular financial data since this data is easily ingested by financial models. However most of relevant financial data is in the form of unstructured textual data. For the purposes of natural language processing public news articles and social media are the two main sources for unstructured textual financial data.

Literature Review

Traditionally, financial forecasting relied primarily on tabular historical data about stocks and foreign exchange rates to make predictions about market volatility. More recently though financial analysts are looking to leverage the large volumes of financial textual data available on the internet for data mining. This emerging field at the intersection of Natural Language Processing (NLP) and Financial Forecasting is known as Natural Language Financial Forecasting NLFF. There is evidence suggesting that since the 1990's the correlation between the Dow Jones daily returns and it's historical data has receded therefore prompting analysts to look elsewhere for information for market data mining purposes *(Qian & Rasheed 2019)*. Recent studies have also shown that traditional financial econometric models for determining investment strategies are no longer a viable option. Frequentist based regression models fail to accurately take into account any uncertainty in the parameters of the model. There are a number of problems with these frequentist regression models. One of them is that they rely on strong assumptions that are not satisfied by financial phenomenon. Another problem is the strong reliance on p-values in Frequentist regression methods. Using p-values for deciding between models relies on a number of strong (unrealistic) assumptions about the model. These include correct model specification, mutually uncorrelated regressors and error terms that follow a Gaussian distribution. *(Marcos Lopez de Prado 2019)*.

In the last decade or so, social media websites such as Twitter, Facebook, etc. have generated an incredibly large volume of user content which the analytics community is now interested in mining for information. Researchers are providing evidence that social media sentiment positively and significantly predicts future stock returns, at least during less volatile stretches of market activity.*(Leung et al. 2019)*. Another study by *(CHOI & VARIAN 2012)* claimed that Google trends can be useful for forecasting short-term macroeconomic

factors. In financial forecasting there is a school of thought that adopts a connectionist perspective to the modeling problem which simply means that the movement of two stocks in the same industry is positively correlated. Data mining techniques of textual information can help with the discovery of these connections.

Financial forecasting can mean a bunch of things such as inflation rate prediction or credit score prediction but this paper will be focusing solely on employing textual data to explain stock market volatility due to a number of reasons. The first reason is the lack of accessibility for many other types of financial assets. Second reason is the nature of other financial products such as Treasury bills whose value is driven more by policy and so there is likely to be a weak correlation between textual financial data and the asset's value, on the other hand financial derivatives are more complicated financial products and have very limited information transparency therefore making them less amenable to textual data mining. The third reason is the transparency of stock and currency markets which means that there is a lot of publicly available information on stocks and currency market. There are lots of participants in these markets and thus there is a lot of weight given to the large volume of opinions generated by these market players.

When financial analysts started text mining for financial forecasting the most common method for importing these new predictors into their models was to use the bag-of-words method for representing textual information on a computer. The reason for this was that it's a very easy and simple method since it just involves using the set of words in the text and their frequencies. However the drawback with using the bag-of-words approach is that this representation of textual data does not take into consideration the order of words and this approach also fails to capture any semantic similarity between the words. One common alternative these days is to use a family of neural networks for word representations. *(Bengio, Y. et. all 2003)* This approach allows for more context information to be captured by the model at the expense of more computational complexity.

Once an appropriate model for text representation is decided, the next step in the analysis is to decide what information to extract from the text representation in order to use it in the model for stock price volatility prediction. There are a number of possible approaches that can be taken. These include sentiment analysis, named-entity recognition and Verb Analysis. A study found that by analyzing the verbs used in financial news articles and then using a machine learning algorithm, one can create an accurate model for stock price movement. The verb analysis method finds those words in financial articles that can move the price the most and whether those terms are positively correlated with the price or negatively.*(Robert P. Schumaker 2010)*

Most of the literature on the correlation between market sentiment as captured by sentiment analysis of financial text focuses on using complex non-linear models to accurately predict stock price volatility. The goal in most of these papers is predictive accuracy. One such paper is by *(Robert P. Schumaker, Hsinchun Chen 2006)* in which they use a support vector machine rather than a linear regression to find a statistically significant correlation between stock sentiment and stock price. They use financial articles as their source of financial textual data and employ a number of different text representation methods for sentiment mining. These include the standard bag-of-words approach and the Noun phrases approach. In their study the authors found that using the Noun phrases approach for text representation rather than the traditional bag-of-words approach resulted in greater predictive accuracy. Another study by *(Tay and Cao 2001)* also suggest using a Support Vector Regression model and demonstrate that relative to linear regression support vector regression results in better predictive accuracy. Some studies have also proposed a hybrid approach by combining machine learning methods with time series methods such as auto-regressive integrated moving average (ARIMA). *(Zhang, G.P. 2003)*

In light of the above literature on Natural Language Financial Forecasting, the goal of this paper is more about inference and less about prediction. Most studies in this sub-field of Natural Language Processing have focused more on finding better and more accurate predictive models for stock price movement. This paper aims to understand if there is indeed a significant positive relationship between stock price and stock movement, and to understand the exact nature of that relationship. The approach taken in this paper towards text representation of financial textual data will be that of bag-of-words. Despite it's shortcomings, the bag-of-words approach offers a simple and computationally feasible method for text representation. After that the text will be mined for sentiment using two different lexicons for sentiment mining. The relationship between the price and sentiment will be modeled using a Bayesian regression model. Given that the goal

of this paper is more about understanding the relationship between sentiment and price, it seems more appropriate to use a Bayesian regression framework since this will allow us to more accurately incorporate any uncertainty about the model parameters into the distribution of the model's parameters.

Methodology

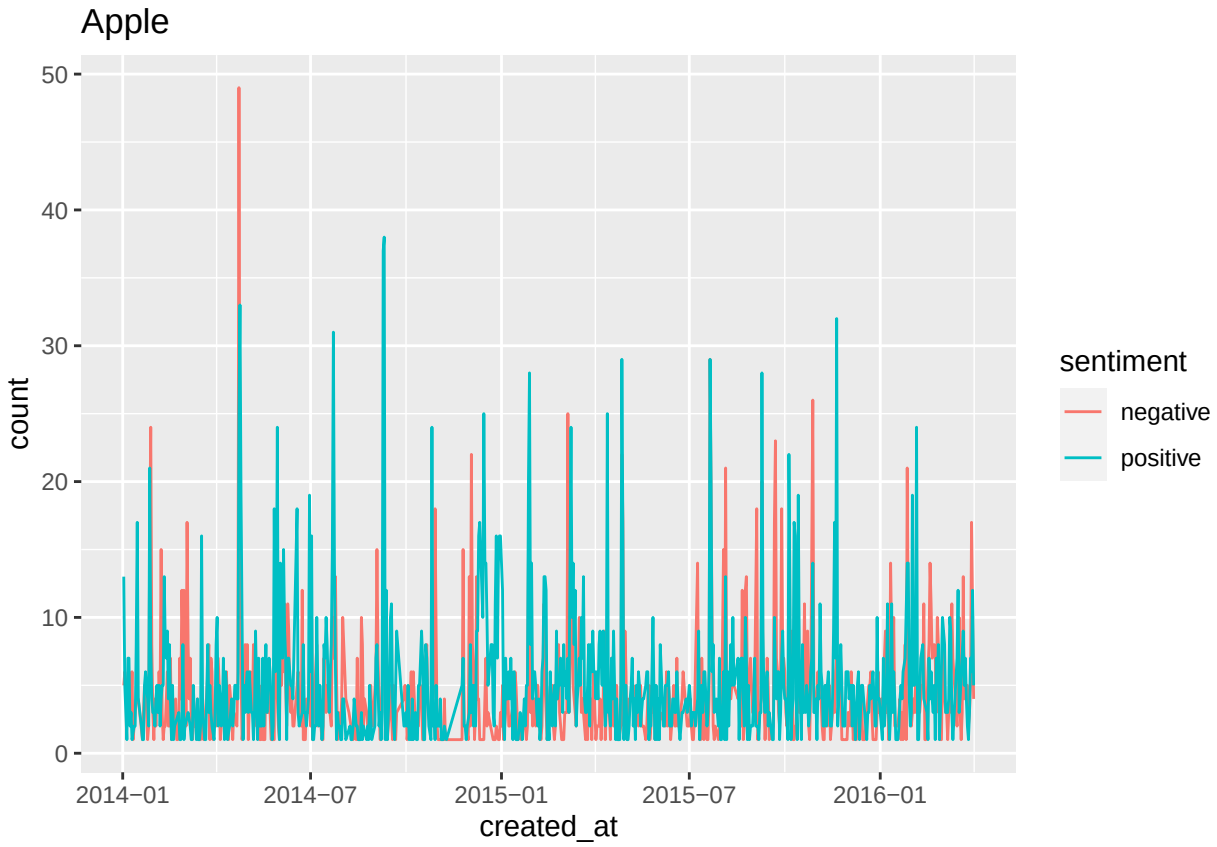
Most computational analysis of unstructured textual financial data can be divided into two main parts. The first part involves cleaning the corpus of textual data and scoring the tokenized text data for sentiment. This usually involves using a sentiment dictionary that has been precompiled from a general corpus of textual data. One also has to define how to tokenized the data. The simplest option is to use each seperate word as a single token (which is the approach taken in this paper). Another option is to use what is known as an n-gram approach where a single token of text can be a collection of n words. The approach, in this paper, taken towards this first part is what is known as tidytext *(Silge and Robinson 2016)*. The reason for taking this approach is that it follows the same tidy data principles *(Wickham 2014)* that are employed in a suite of other data wrangling and visualization packages in R and thus allows for a smooth data wrangling and visualization workflow. Basically the tidytext approach involves treating text as dataframes of individual words where each row is a single token (observation). The tidytext approach allows for a token to be a single word or a group of words (n-gram). This is in contrast to the general approach that is taken in most current analysis of textual data where the textual data is stored as either a string object or document-term matrix. Once the tweets are tokenzied as per the tidy text principles they are scored for sentiment using one of two sentiment dictionaries. One of the dictionaries is known as **bing** collected by *Bing Liu and collaborators*. This dictionary is a general sentiment dictionary that classifies a word into either positive or negative. Any words that cannot be fit into either bucket are by default classified as neutral. The second dictionary is a more context-based sentiment dictionary and is built specifically for financial texts. It's known as the Loughran and McDonald dictionary of financial sentiment terms. (Loughran and McDonald 2011). After each token(word) is scored for each tweet the average sentiment score is computed for each tweet. After the textual data is tokenized and scored for sentiment, the second part of the analysis is to use some form of machine learning (whether it's a basic regression model or a neural net regression model) for finding a function that can be used to map stock sentiment to stock price. Since the goal of this paper is more about inference and less about prediction, highly non-linear models are not used since the gain in predictive accuracy that these models provide comes at the cost of interpretability of the models parameters. Given the goal of the paper, Bayesian regression is much more suited to the task of finding an association between the market's sentiment of a stock and it's valuation, in terms of price, of the stock. The main difference of this approach from a regular frequentist regression model is that of estimating an entire posterior distribution for the parameters in the model rather than a single point estimate for each parameter. The benefit of this approach is that it allows us to more accuratley incorporate the uncertainty in our data about the parameters in the model and thus about the association between the stock price and the stock sentiment. Of course estimating a posterior distribution of parameters is a lot more computationally expensive than estimating a single point for the posterior distribution. A modern approach for posterior inference is known as Markov Chain Monte Carlo(MCMC). These are a class of numerical algorithms that approximate the posterior distribution rather than calculating it directly through multi-dimensional integrals (which is often impossible to do in practice). A well known computationally feasible MCMC algorithm for estimating the posterior is Hamiltonian Monte Carlo *(Betancourt 2017)*. This algorithm uses a physical analogy for the posterior distribution and tries to estimate it based on the analogous physical properties of the distribution. This algorithm is the workhorse engine of the Stan programming package which in this paper is interfaced through the BRMS package in R.

Data

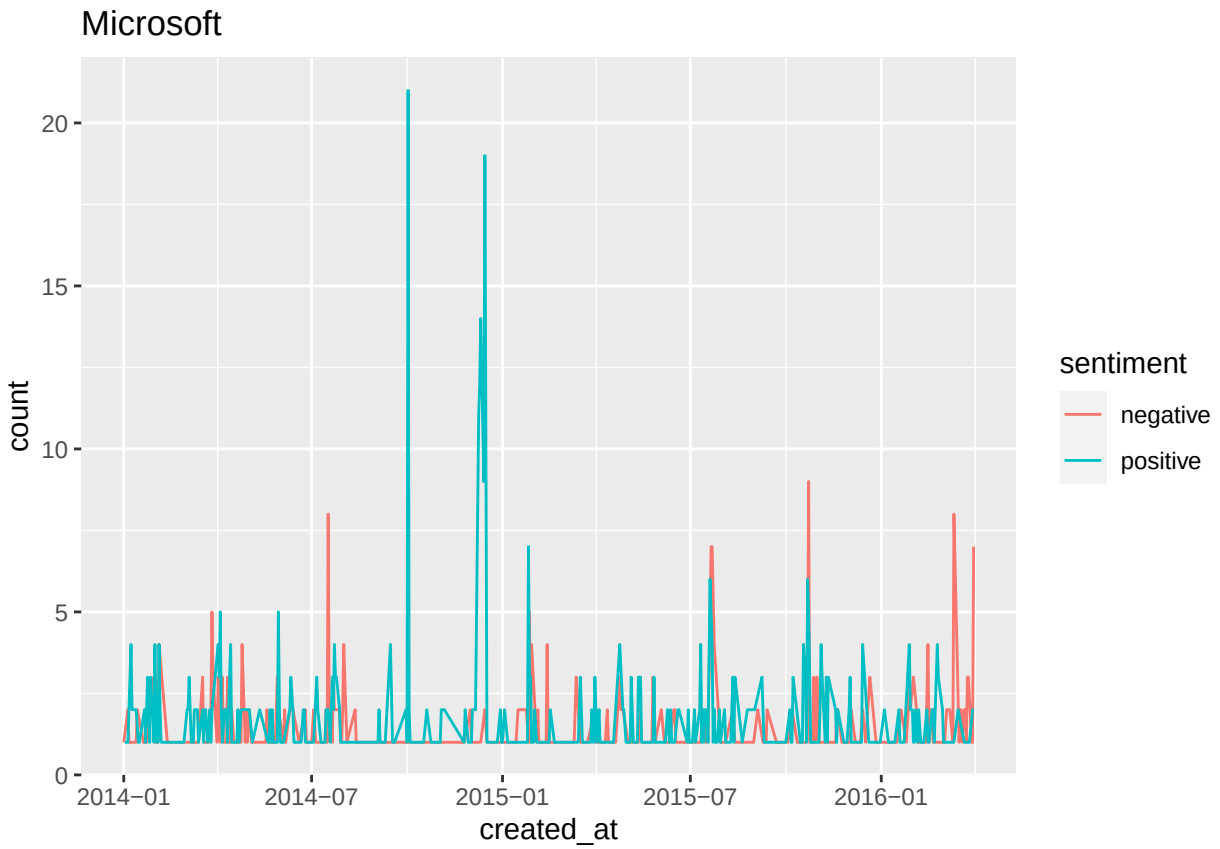
The textual financial data is in two part. The first is a dataset of financial tweets that was collected by Yumo Xu and Shay B. Cohen for a stock movement prediction paper *(Xu and Cohen 2018)*. The raw tweets are

contained in a JSON file for a number of different stocks from different industries. Each tweet has a unique ID for identifying it and also has a timestamp that contains date and time information for that particular tweet. These tweets were retrieved from twitter using the official twitter API and were searched for using regex expressions that were made of the NASDAQ ticker symbol for each stock. The second is a dataset acquired from Kaggle. This dataset contains financial articles for a large number of stocks that are traded on NYSE/NASDAQ. The original data has been collected from *investing.com* which is an archive of historical news on US equities. The stock price data has been retrieved from Yahoo finance for the appropriate years. Only the date and closing price variables are being retained because the date variable is needed to match the stock price with the appropriate tweets/articles and the closing price is being considered because any information regarding a stock in the media will tend to have a lagged effect on the price of that stock and thus the closing price will perhaps more accurately reflect that effect than the price of the product at the exact same time of the corresponding tweet/article. Lastly historical price information about the composite market index such as the NASDAQ index and NYSE index have also been downloaded from Yahoo finance. This is data is being used a control variable in the multiple regression models. Like the stock price data, only the Date and closing price columns are being retained since they are the only variables relevant for our model.

The following is a distribution over time of the number of negative and positive tweets for apple



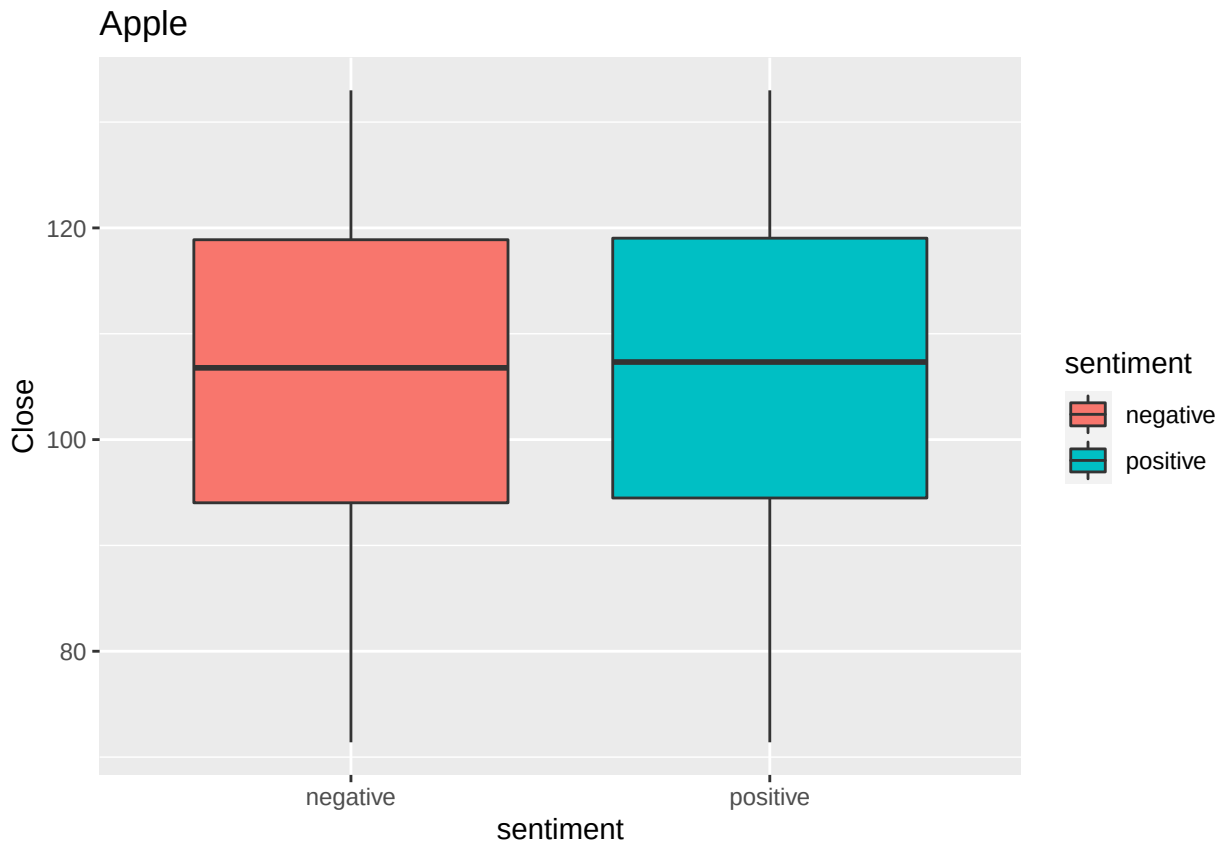
We can see from the plot above that peaks of negative sentiments rarely overlap with peaks of positive sentiment and that the peaks alternate over time with positive peaks quickly being followed by negative tweets. We can see the same plot below for tweets about Microsoft



Again we the same alternating peaks trend. This suggests that using sentiment scores of tweets about a company can, to some extent, reflect how the market feels about that company. Therefore we can use it as predictor variable in our stock volatility model.

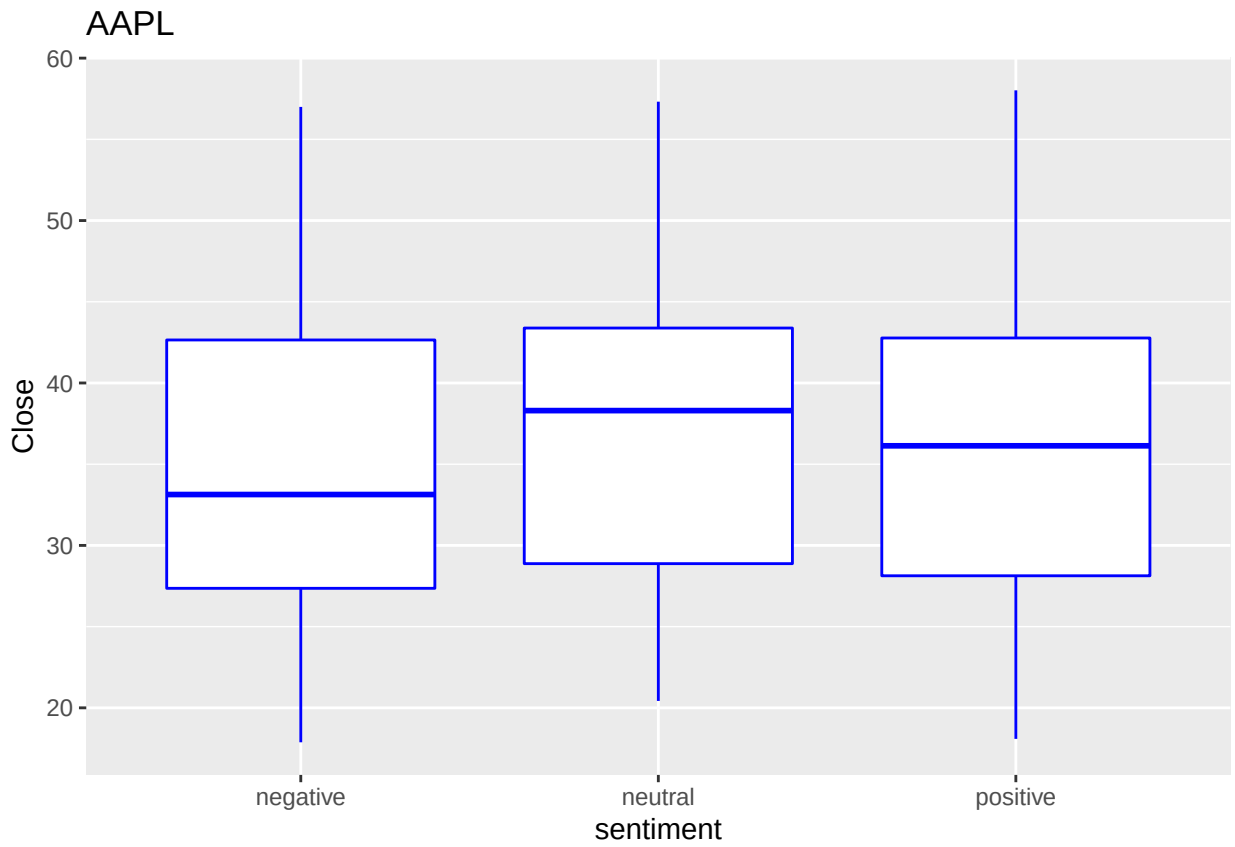
We also want to see if there is any difference in the distribution of stock prices across two groups(positive or negative) of tweets for a particular company.

Again we use the tweets about apple and see if there is any difference between the two groups.



We do see that there is a slight difference in the means of the two groups but not a very significant one.

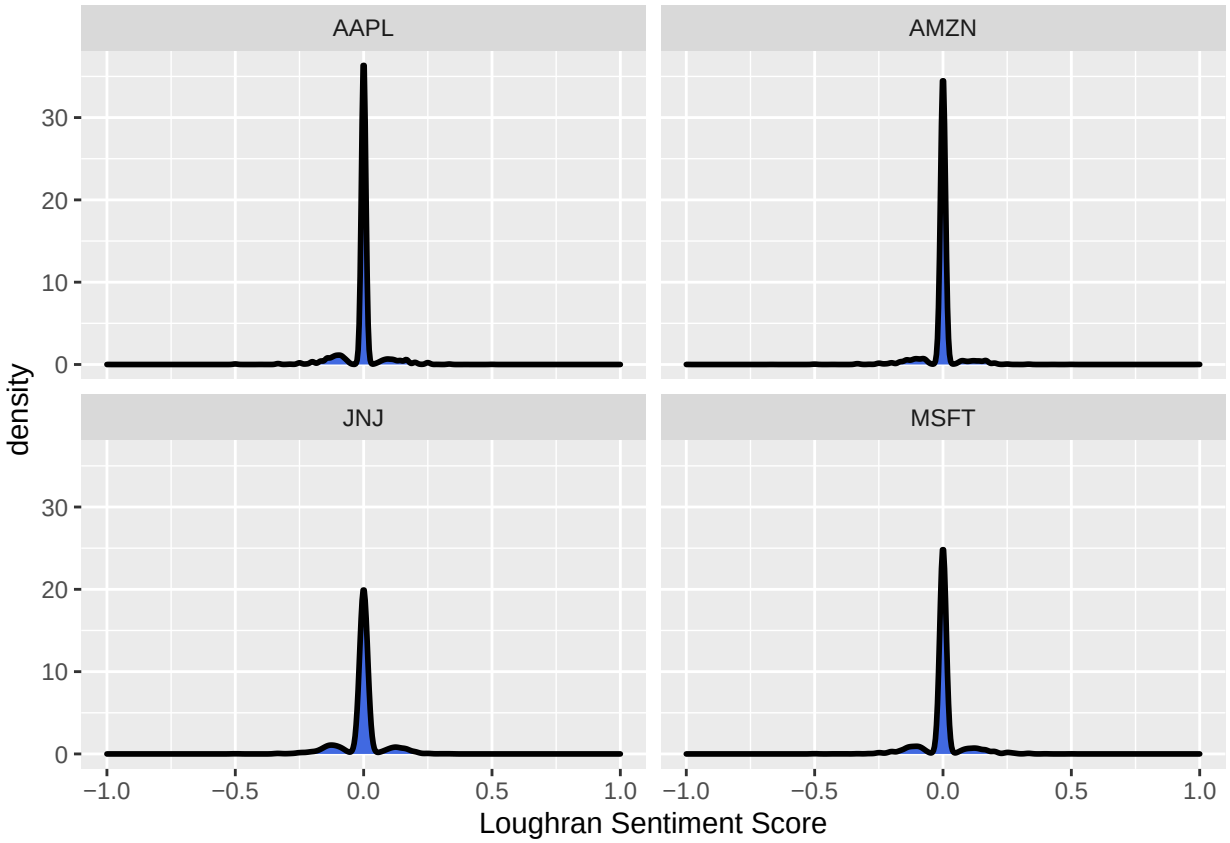
The same box plot can also be made using sentiment data from financial articles instead of financial tweets. The following is the same box plot only this time using financial articles instead of financial tweets.



We again see that average closing price for the positive sentiment group is higher than that of the negative sentiment group only this time it is somewhat more significant.

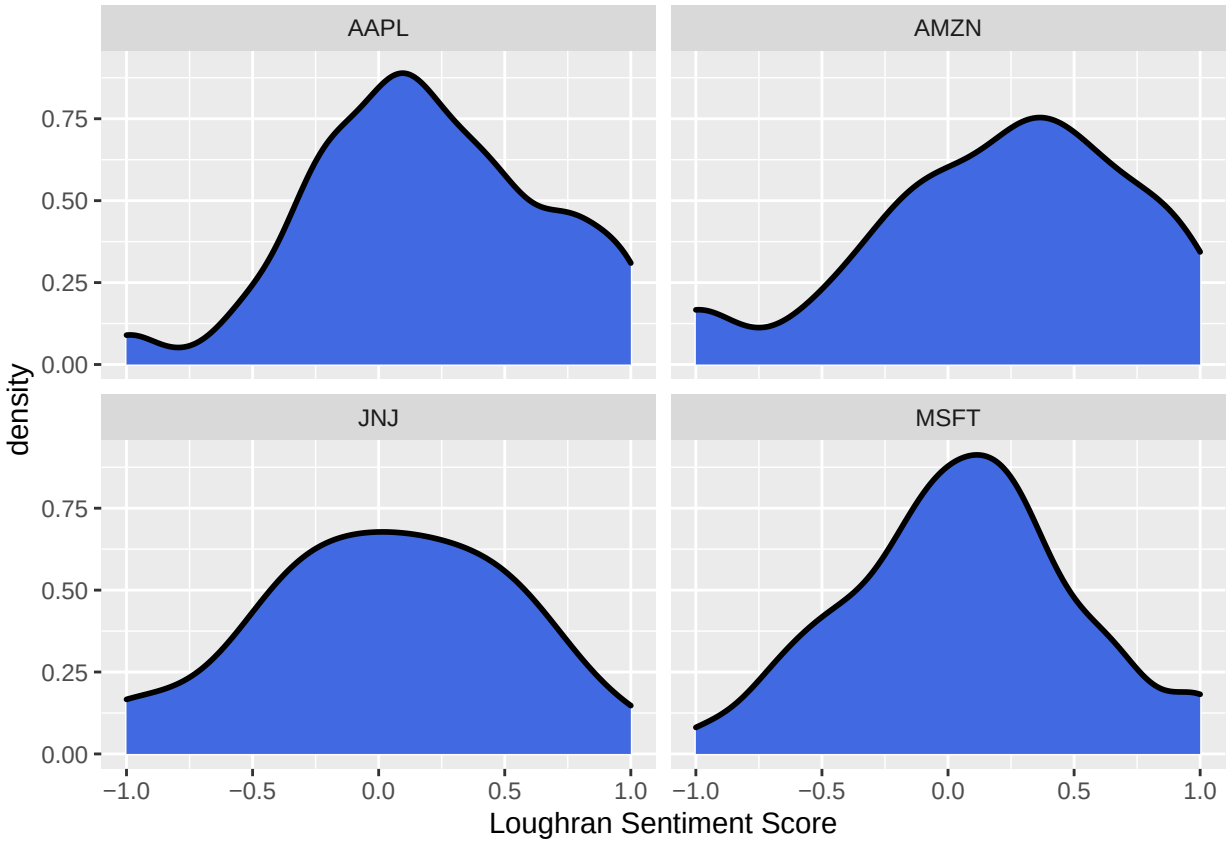
Measures and Variables

The dependent variable of interest is the closing price of a stock. This variable is standardized so as to allow for the Stan program to converge more easily and quickly on a posterior distribution. In the R code this variable is named as $Close_s$ for each stock. The predictor variable of interest for the financial tweets data is the proportion of positive tweets for a particular stock on a particular day. This proportion is calculated by dividing the number of positive tweets by the total number of positive and negative tweets. In the model/s this predictor is the variable *positive*. For each stock's corpus of tweets, two different sentiment scores based on the two different sentiment dictionaries(the Bing and Loughran dictionaries) have been generated. For the financial articles data, the predictor variable is the raw sentiment score. Again there are two different sentiment scores for each stock based on the two different sentiment dictionaries. This predictor is represented by the variable *Score* in the model. The reason for using the proportion of positive tweets as a predictor variable and not the raw scores itself is that since there is a character limit on a tweet, the overall distribution of words per tweet is very narrow and hence the distribution of possible average sentiment scores for each tweet is very narrow. This is illustrated below for the average scores per tweet for a few different stocks.



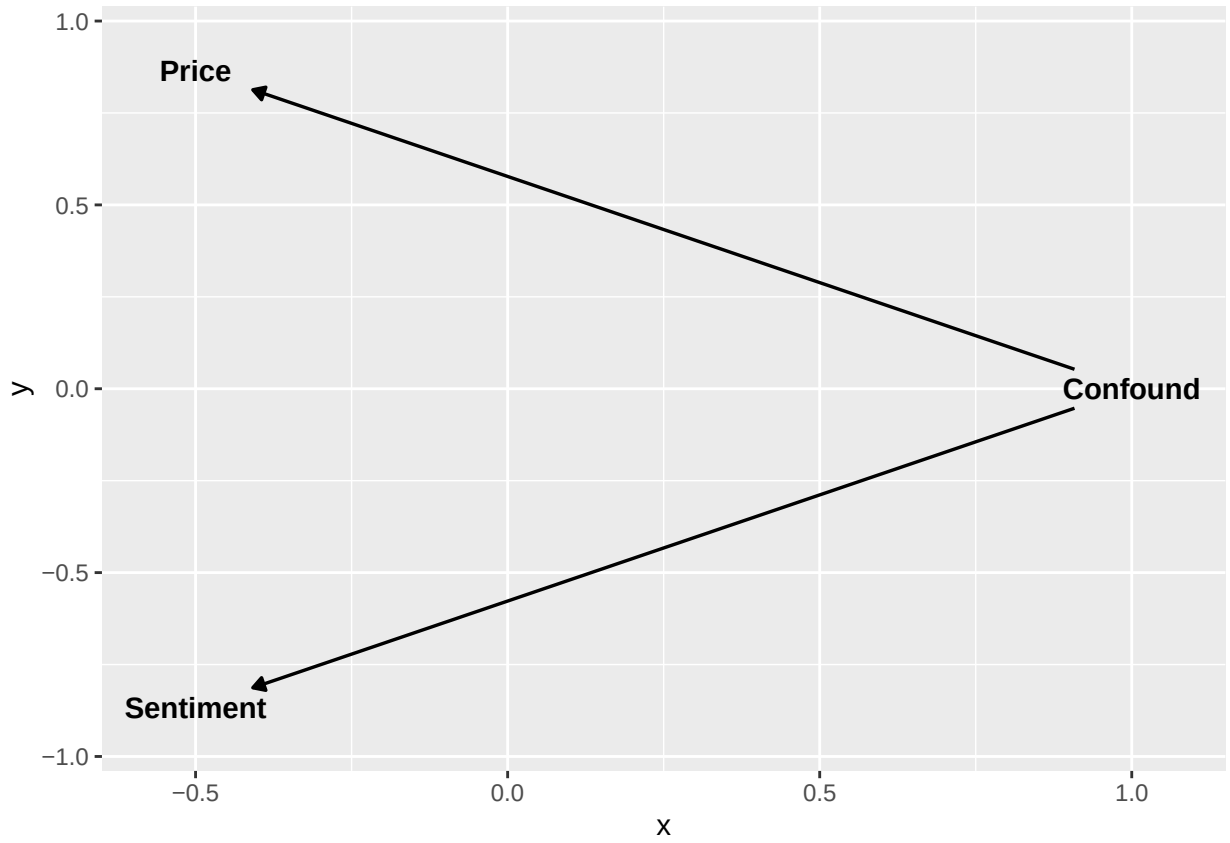
As the plots above show, the distribution of sentiment scores based on the Loughran dictionary is centered around 0 with very little variance. Therefore instead of using the raw scores, the proportion of positive tweets (tweets that have sentiment score greater than 0) are used as the predictor variable instead.

For the financial articles data, the distribution of raw scores have more variance and thus are more suited for regression. We can see this from the following density plots of the distribution of raw loughran sentiment scores for a few different stocks.



As illustrated by the plot above, the Loughran Sentiment scores for the financial articles for the same stocks have a lot more variance in their distribution than the financial tweets for the same stocks. Therefore the raw scores for the financial articles can be used as a predictor variable for the model.

Since most of these stocks are listed on NASDAQ and NYSE, it makes sense to use the values of these composite indices as control variables in our model. The main goal of this paper is to find causal link between stock sentiment and stock price. Unfortunately a regression can only tell us about an assosciation between two variables and if there is an assosciation between two variables, it does not immediately translate into a causal link between the two variables. Including control variables helps us to overcome this problem by conditioning on extra information. In the context of stock prices using the market index value (NASDAQ/NYSE) as control variables helps to eliminate any confounding effect that. For causal query inferences a DAG (Directed Acyclic Graph) is a very useful tool to help visually illustrate our causal questions/hypotheses. For this paper, the causal query is illustrated by the following DAG.



From the above Causal DAG we see that in order to check for dependence between the stock sentiment score and the stock price we'll have to condition on a confound variable or perhaps on several confound variables. The implied conditional independence from this graph is as follows.

```
## Pric _||_ Sntm | Cnfn
```

The goal in this paper of course is to check for conditional dependence between the price of a stock and it's sentiment score after controlling for a confound variable/s, which in our case is the market index value of NASDAQ and NYSE.

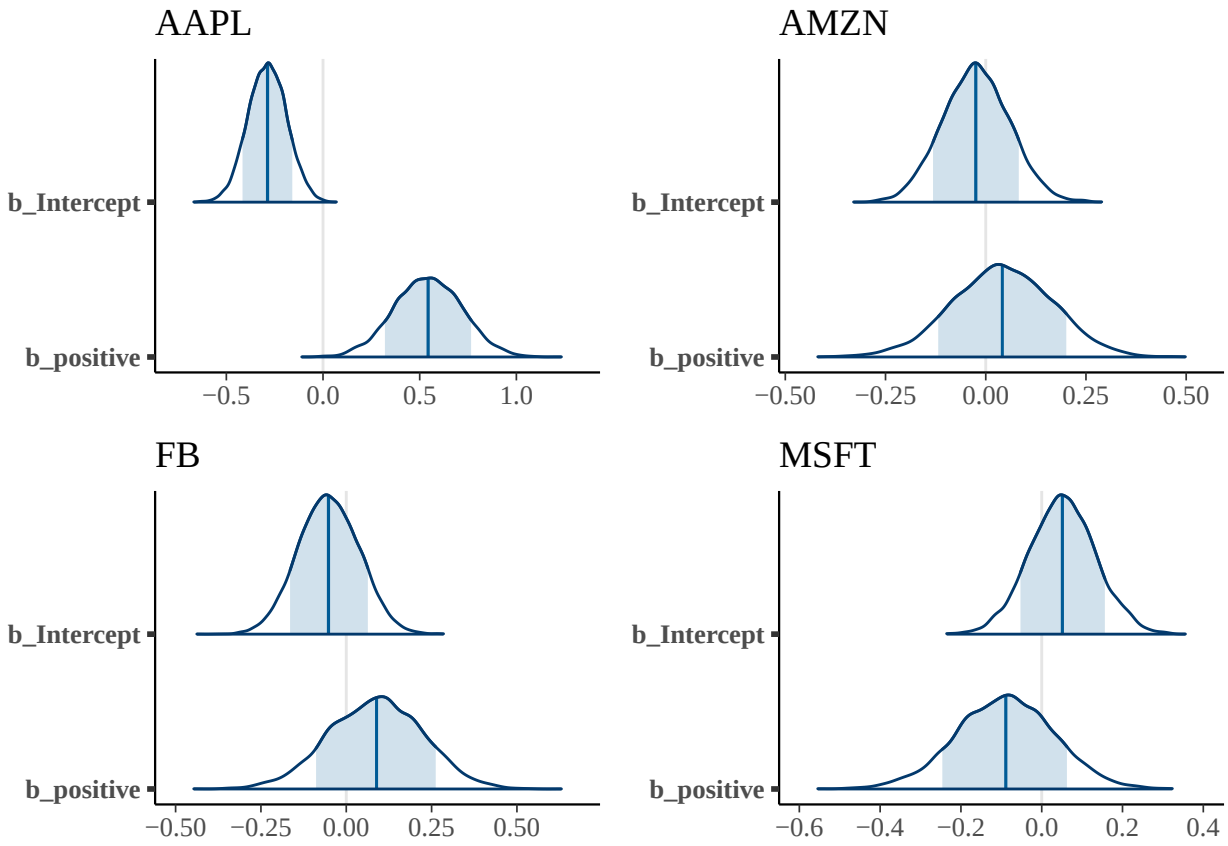
Results

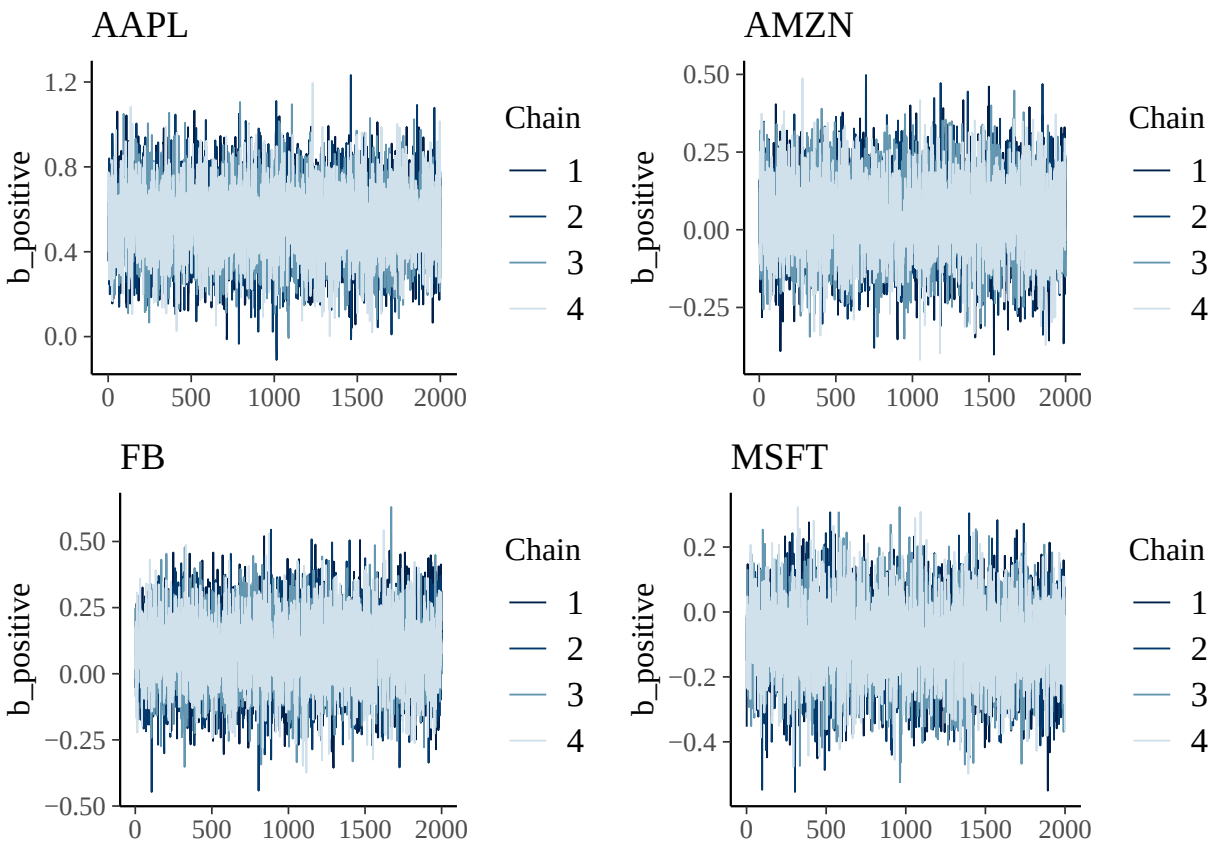
For each stock in our sample of stocks for analysis there are two regression models based on the two different sentiment scoring dictionaries. Furthermore for each type of score there are two different models. One is a simple linear regression model based on the following proposed relationship $Price = \alpha + \beta Sentiment$ (for the articles) and $Price = \alpha + \beta Positive$ (for the tweets). The other model is a multiple linear regression model based on the following proposed relationship, $Price = \alpha + \beta_1 Sentiment + \beta_2 Nasdaq$ (for the articles) and $Price = \alpha + \beta_1 Positive + \beta_2 Nasdaq$ (for the tweets).

We'll first analyse the results of our models for the financial tweets. We'll look at a number of different stocks and see the how the results of the models differ across different stocks/industries.

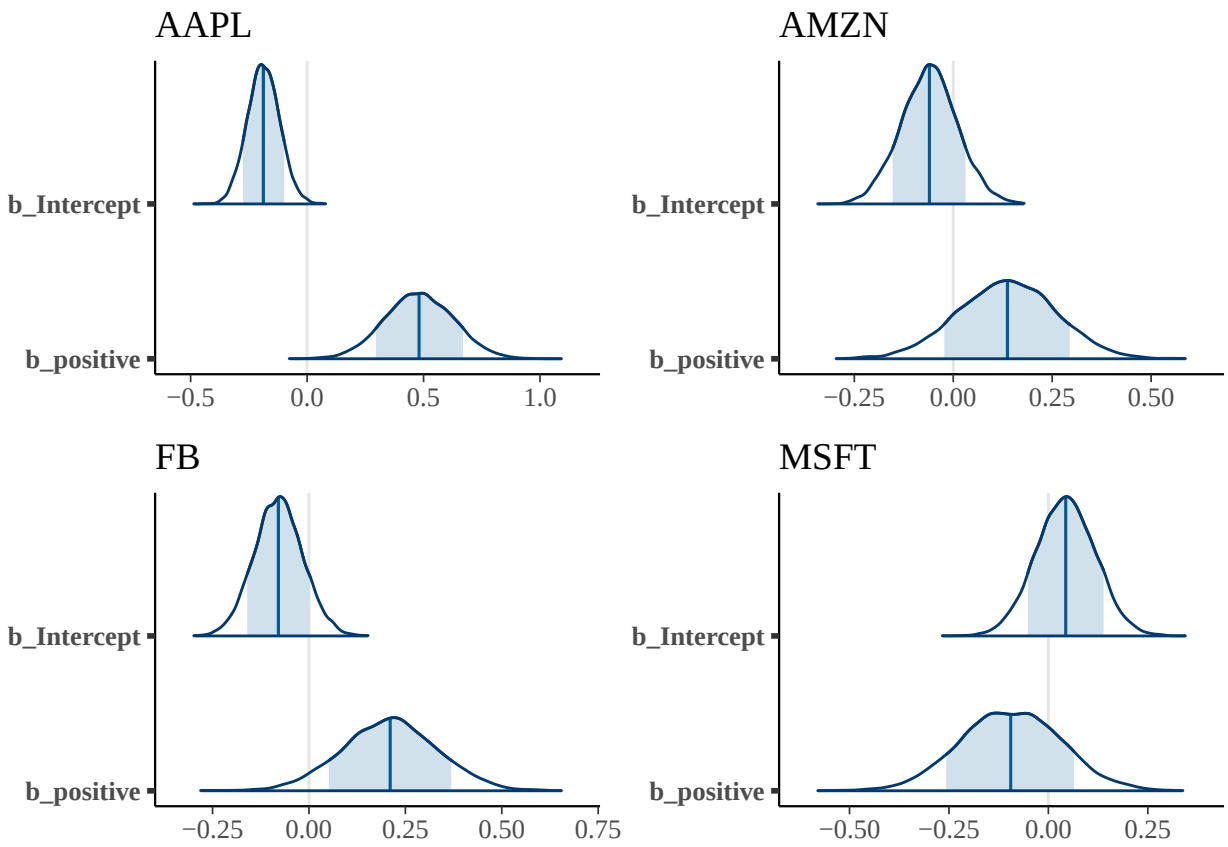
We'll start with analyzing stocks from the tech industry. These include stocks from Apple, Microsoft, Facebook, Amazon.

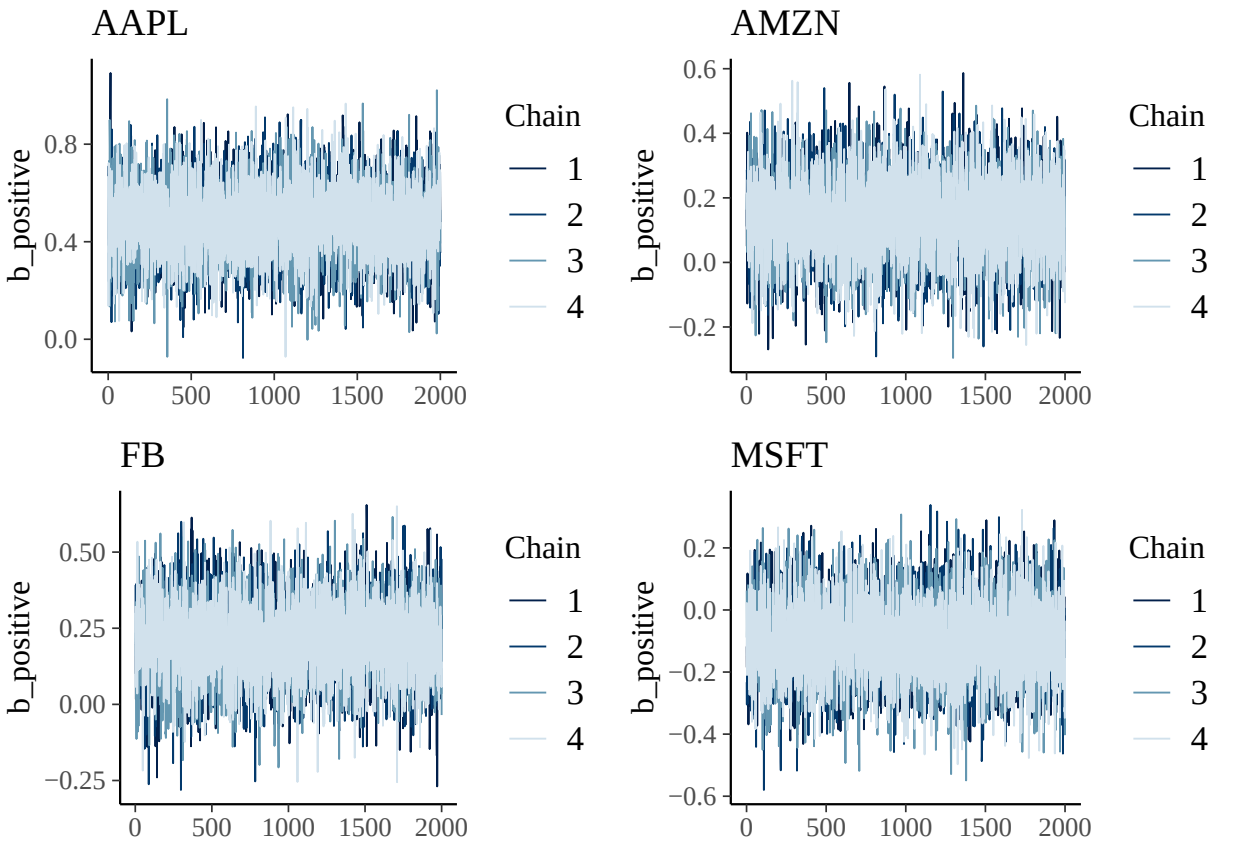
The results of the the simple linear regression model $Price = \alpha + \beta Positive$ based on the bing sentiment scoring dictionary are shown in the following plots.





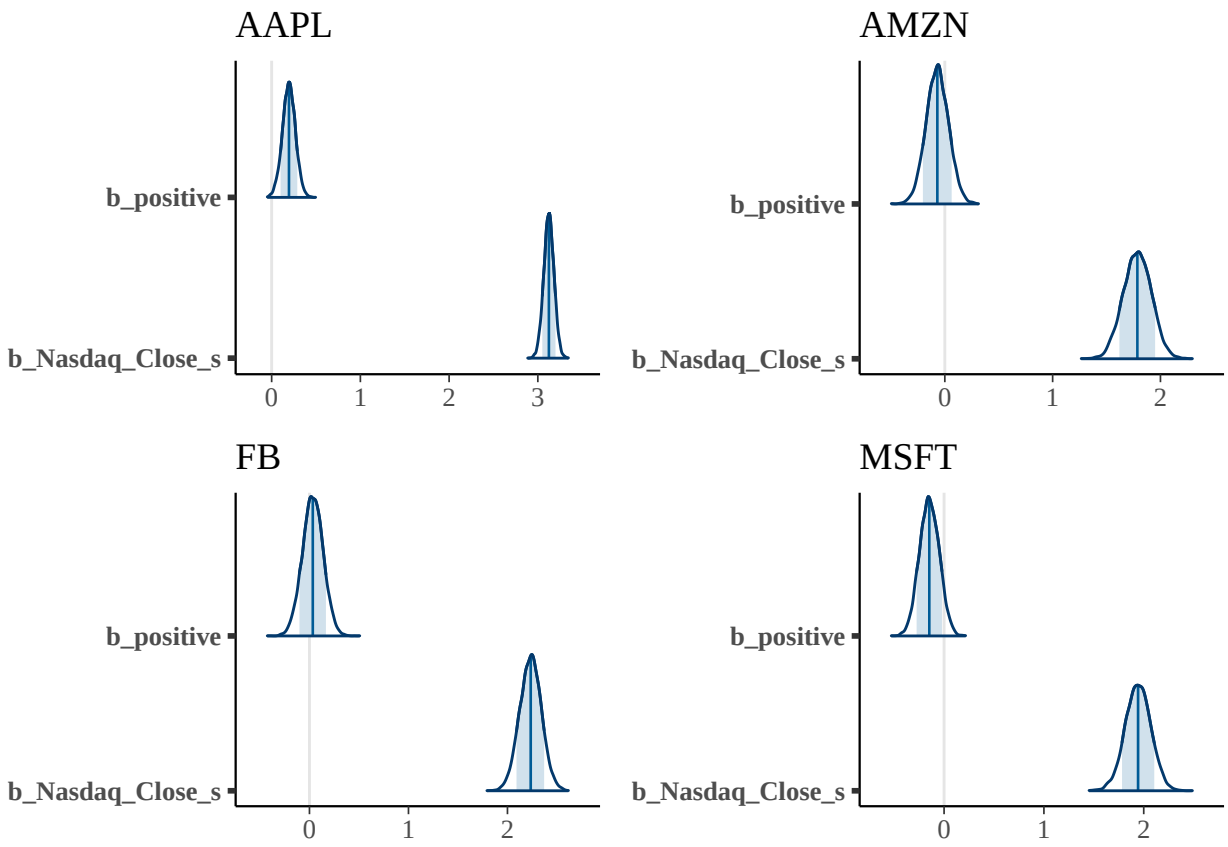
Using the Loughran sentiment dictionary, which is more suited for financial text, results in the following plots


AAPL
b_Intercept
b_positive
−0.5
0.0
0.5
1.0
AMZN
b_Intercept
b_positive
−0.25
0.00
0.25
0.50
FB
b_Intercept
b_positive
−0.25
0.00
0.25
0.50
0.75
MSFT
b_Intercept
b_positive
−0.50
−0.25
0.00
0.25

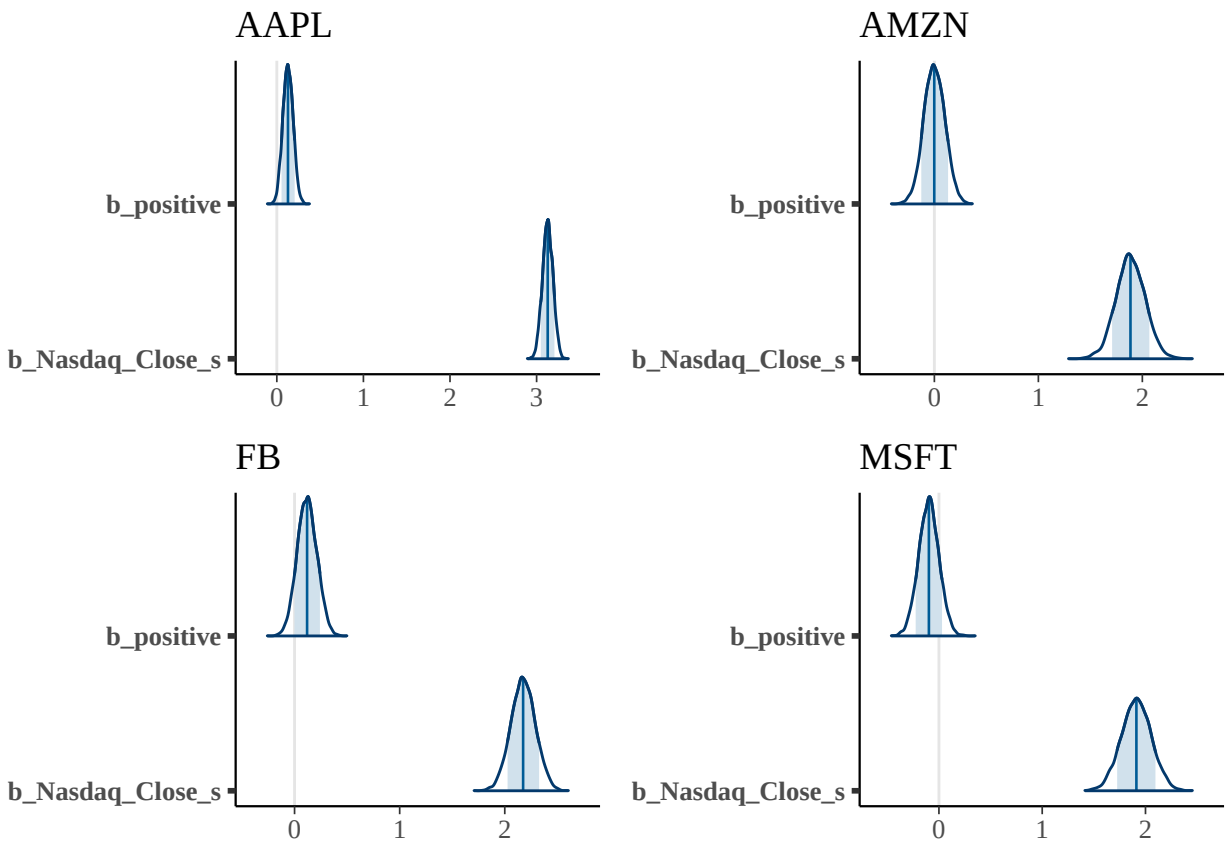


By comparing the two sets of plots we see that the distribution of the coefficient for the *Positive* variable (proportion of positive tweets) has shifted slightly to the right for each coefficient as the sentiment scoring dictionary is changed for each stock. Interestingly the distribution of the *Positive* variable coefficient for Microsoft , while also having shifted to the right, now has two peaks. Furthermore the markov chains (on the right in each graph) seems to converged and mixed well enough to suggest that model was able to find a posterior distribution in each case. For each stock the coefficient for *Positive* does take on positive values, indicating that there seems to positive correlation between the price of a stock and the proportion of positive tweets about that stock for that day. However the coefficient also takes on negative values meaning that the relationship can also be negative but since the distribution is skewed more towards the right (positive x-axis) in each case, there does seem to be some evidence of a positive relationship.

Having analyzed the simple regression models, we'll now look at the multiple regression models with the control variable $Nasdaq$, which is the composite index value of the market on which these stocks are traded. Our model equation is as follows $Price = \alpha + \beta_1 Positive + \beta_2 Nasdaq$. Including this control variable will help us eliminate any spurious correlations that the simple regression models might be suggesting to us. Again we'll first look at the MLR models based on the bing sentiment dictionary. The summary of these models is illustrated below.



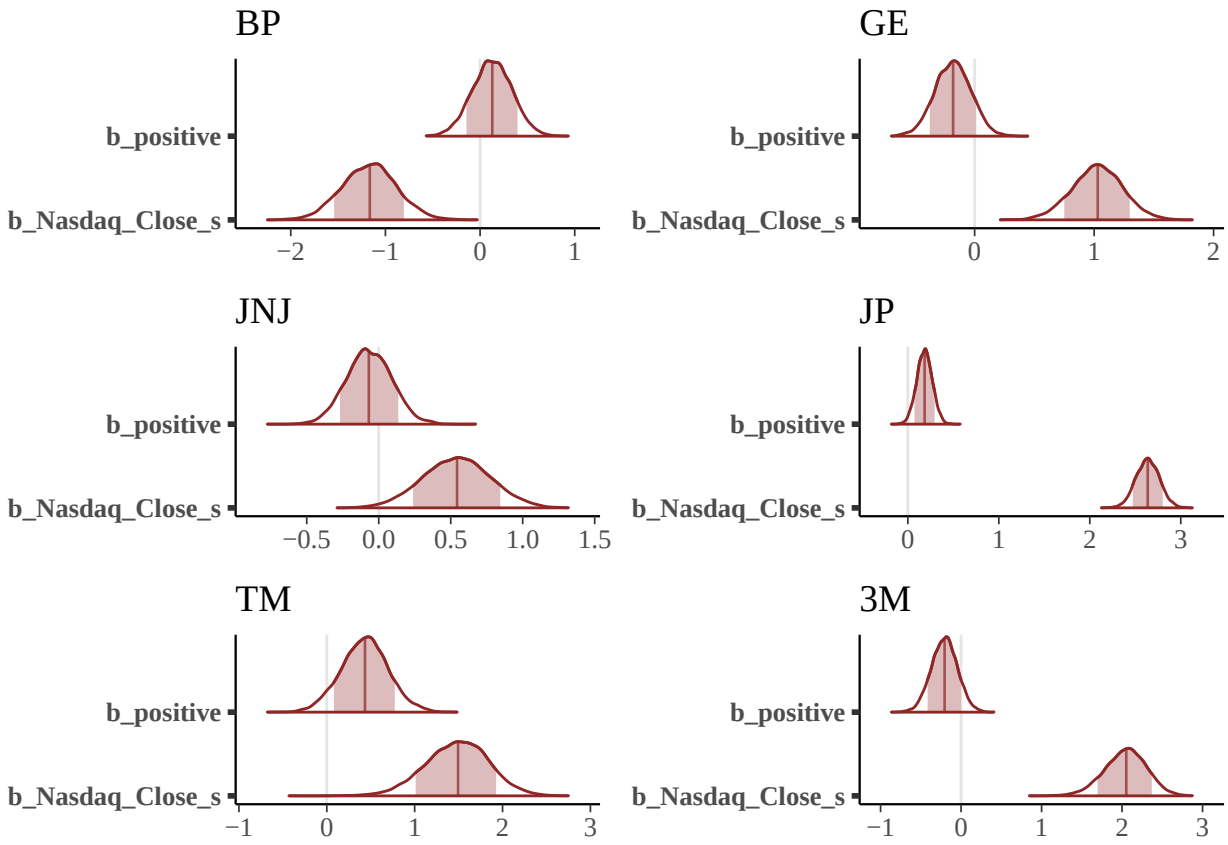
The following are the results based on the loughran dictionary



Again with the loughran dictionary the distribution for the coefficient of *Positive* is more skewed towards the right (positive x-axis). This could suggest that the loughran dictionary picks up positive words that are only seen as positive in a financial context.

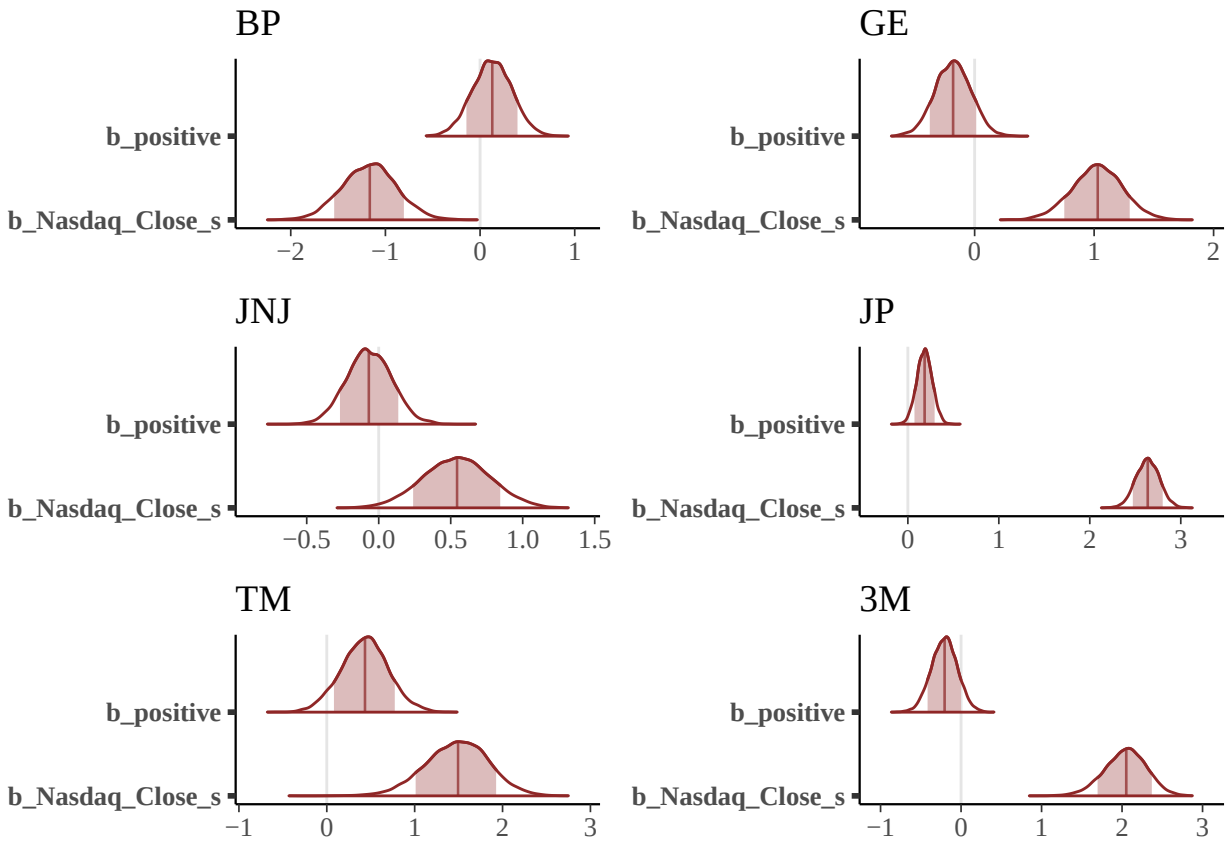
These were the results for all the models for Tech stocks (Apple, Microsoft, Amazon, Facebook). Now we'll look at the results for Non-Tech stocks. We'll only look at a representative sample of the Non-tech stocks in our data and we'll only look at the results of models that are based on the loughran sentiment dictionary since we've established now that the loughran dictionary is better suited for financial text data.

We'll first look at the results from the simple regression models. These are illustrated below.



We see a somewhat similar trend for the distribution of the coefficient for the variable *positive* in both these and the Tech stocks. For most of these stocks the distribution has a large variance with both positive and negative values for the coefficient but with positive values making a larger portion of the distribution.

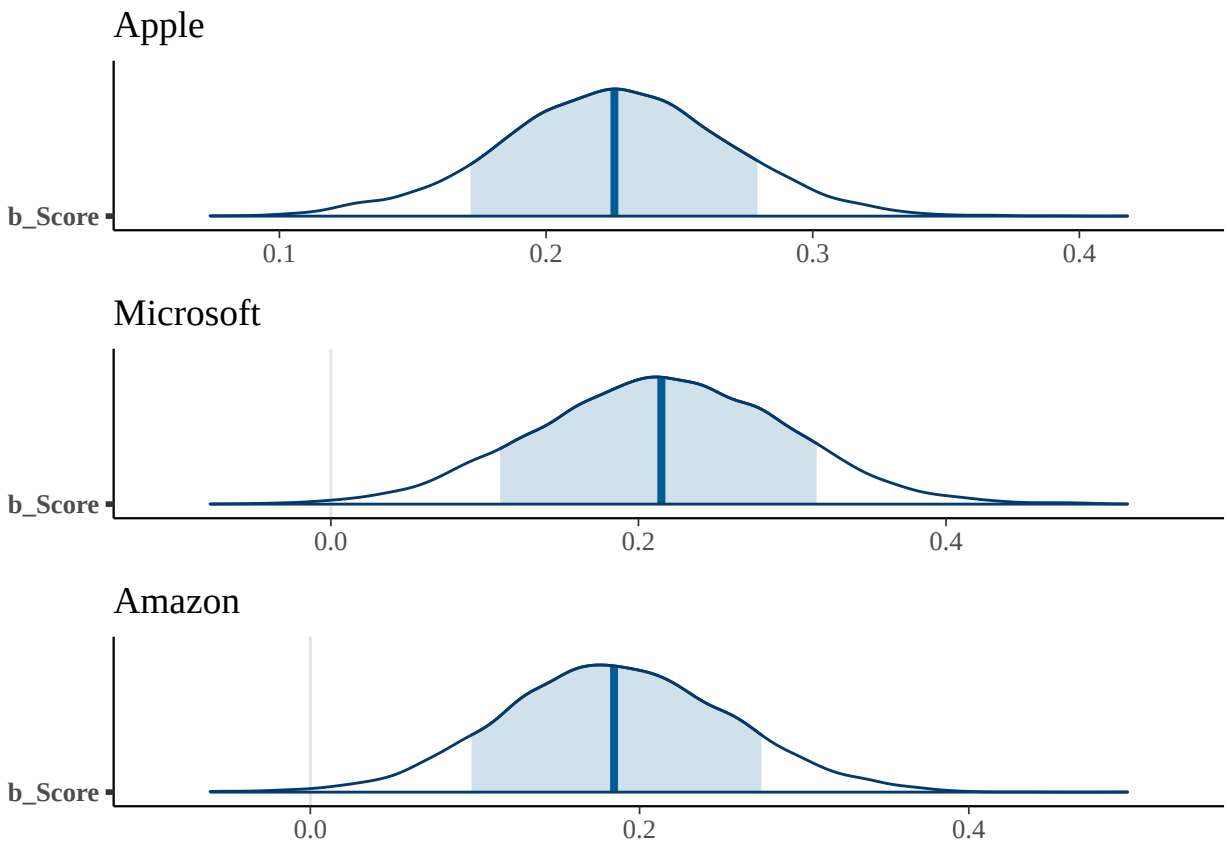
Again we'll include the results of the Multiple linear regression model for the Non-Tech stocks in order to check if the positive correlation that we see in the above plots isn't a spurious correlation



It appears that the inclusion of the control variable, $Nasdaq$, has caused the distribution for the coefficient of the variable *positive* to shift more towards the left (negative x-axis). The distributions also seem to have lower variance now indicating that the model is now relatively less uncertain about the assosciation between stock sentiment and stock price. Each of these distributions is now closely centered around 0 indicating that the model doesn't think there is a significant correlation between the stock sentiment and it's price.

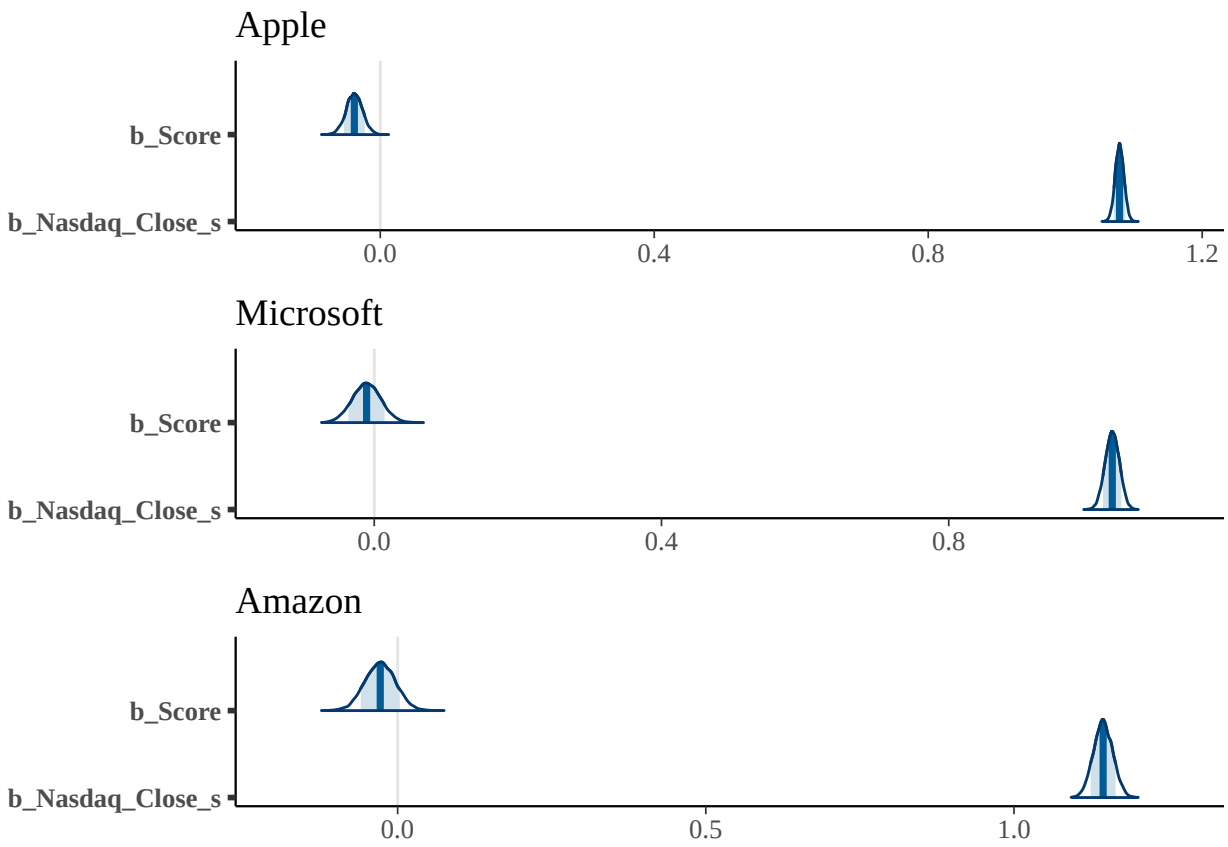
All of the results above are from models that gauged a stock's sentiment based on tweets about that stock. The main constraint with tweets is their character limit. This limit makes any sort of opinion mining very difficult and so it becomes more difficult to accuratley gauge any sentiment towards a stock from that particular tweet. Using financial articles allows us to resolve this problem since articles have a much larger body of text and thus allows for more accurate opinion mining. The following results are now from models that gauge a stock's sentiment using financial articles rather than financial tweets. There is a now slight change in the specification of the general model. The model equation is as follow, $Price = \alpha + \beta Score$ (for simple linear regression) and $Price = \alpha + \beta_1 Score + \beta_2 Nasdaq$ (for the multiple linear regression). We'll again look at Tech stocks first and also this time we'll only use the sentiment scores based on the loughran dictionary.

The results for the simple linear regression models are illustrated below.



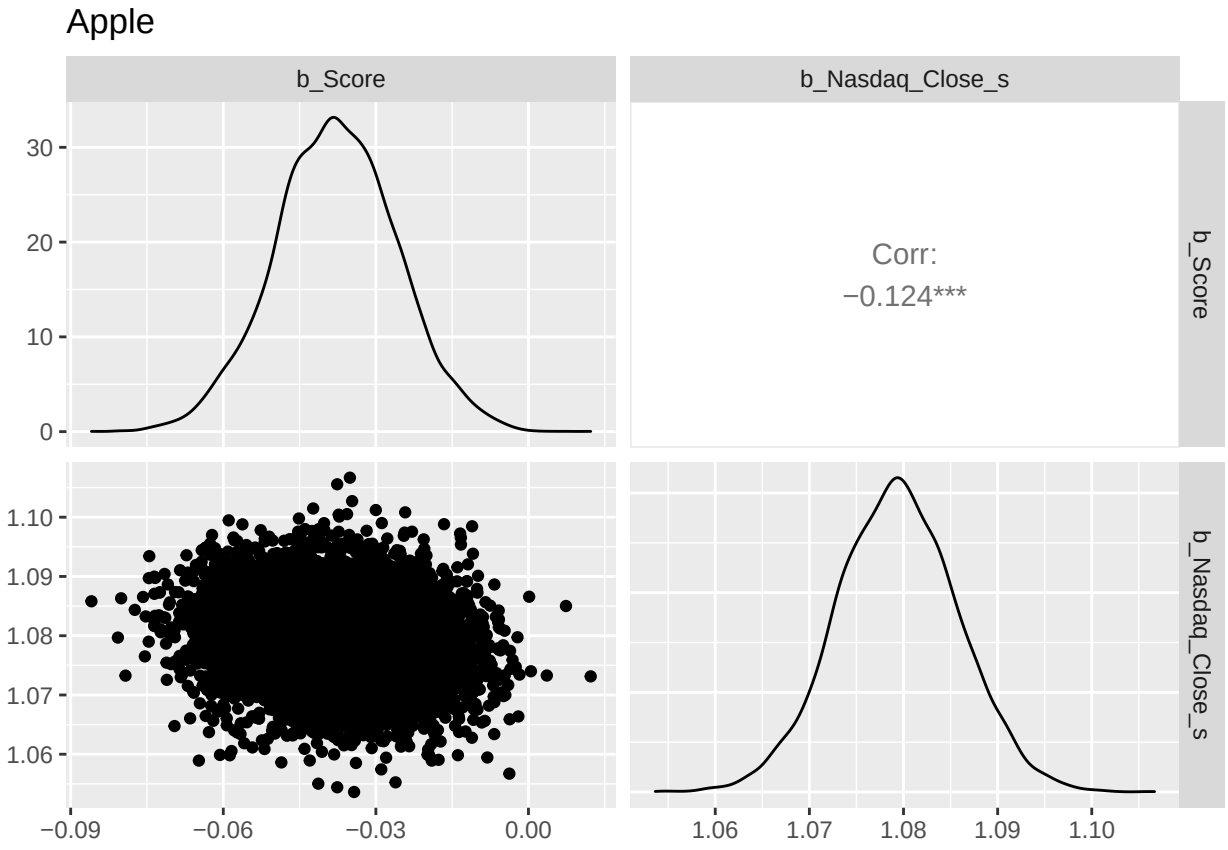
We see that the distributions for the coefficient of the variable *Score* have a large variance but almost all the values in the distribution are positive. This suggests that there maybe a positive correlation between the price and the sentiment score of a stock.

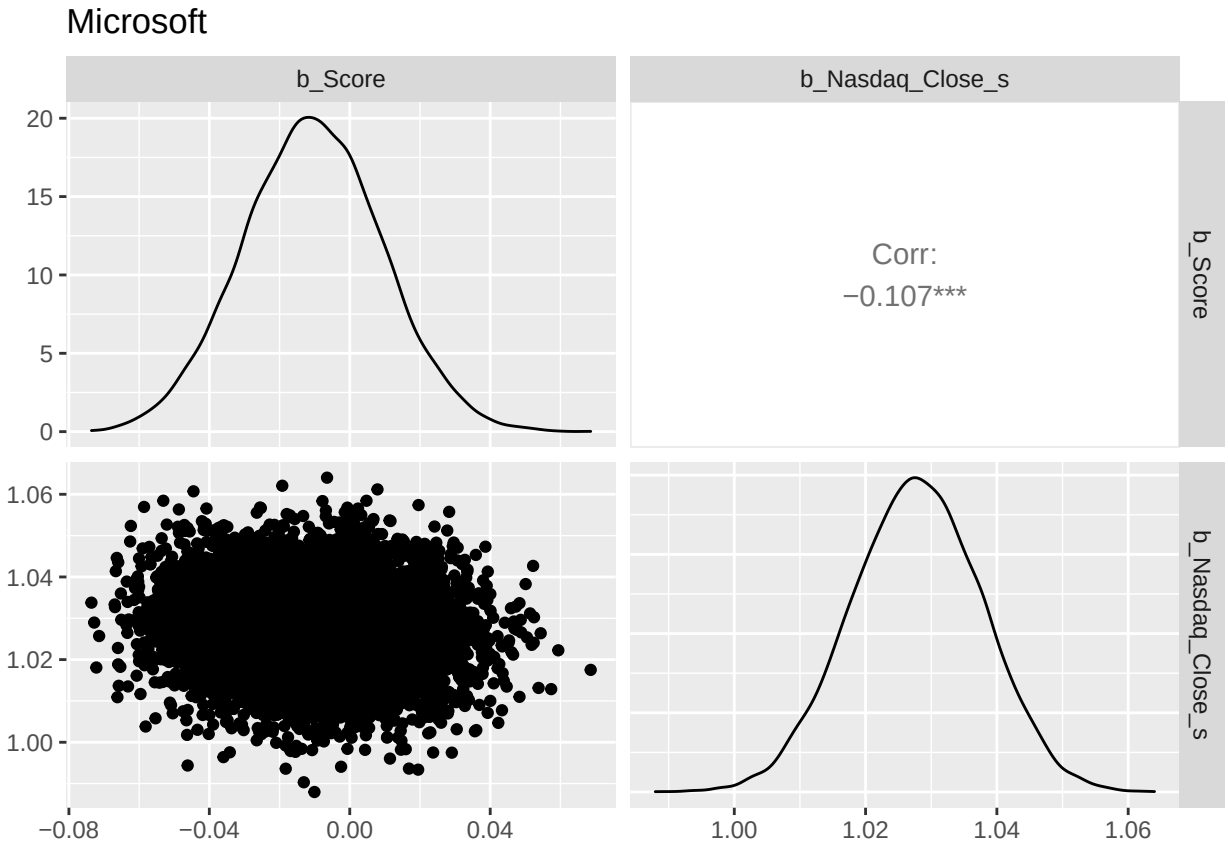
Now we'll look at the multiple linear regression models for the tech stocks and again we'll only look at models that use the loughran dictionary for sentiment scoring.

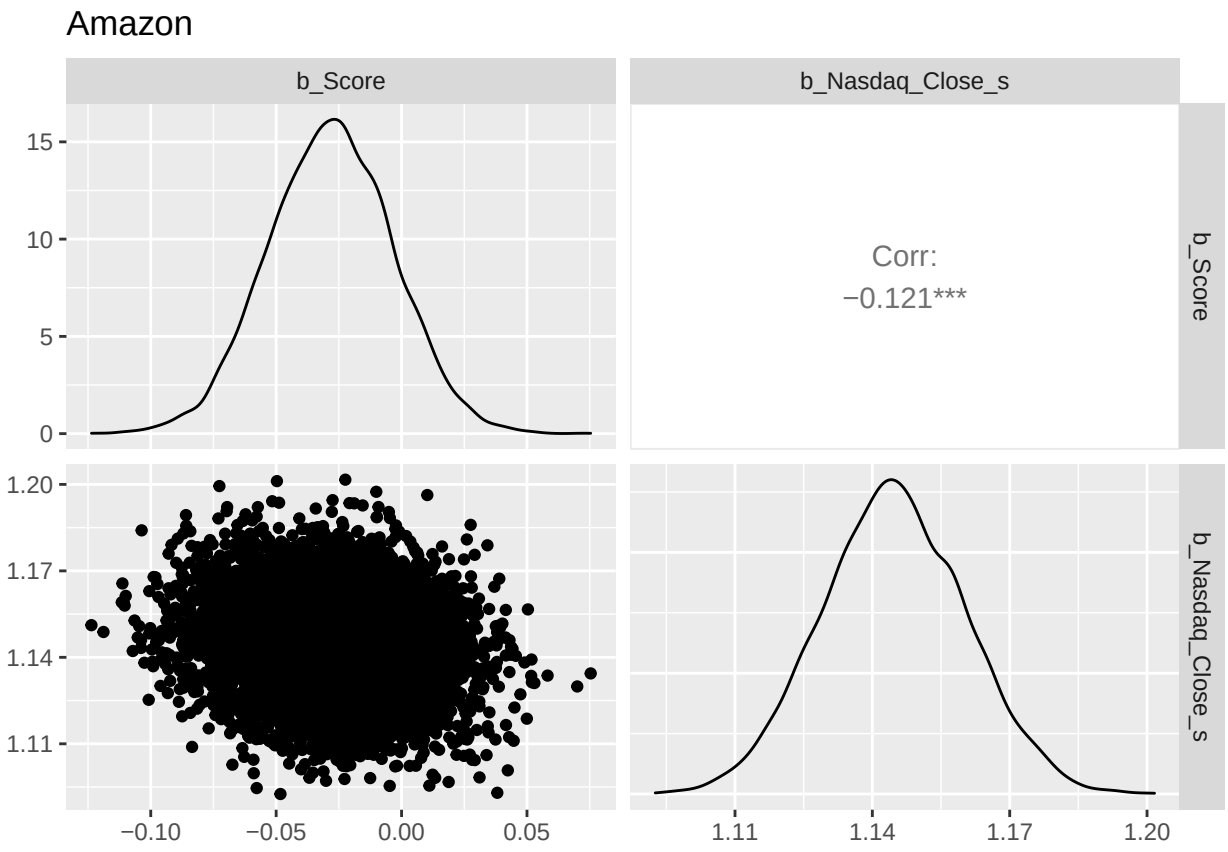


Like with the tweets data, including a control variable (*Nasdaq* in our model) the distribution of the coefficient of *Score* has shifted towards the left (negative x-axis) indicating that their are now fewer positive values for the coefficient and thus little evidence for the existence of a positive correlation between the stock price and the stock sentiment. Relative to the simple linear regression model the distribution of the coefficient of the variable *Score* has lower variance and is centered (in most cases) around 0, meaning that the model is more certain there isn't a positive assosciation between stock price and stock sentiment.

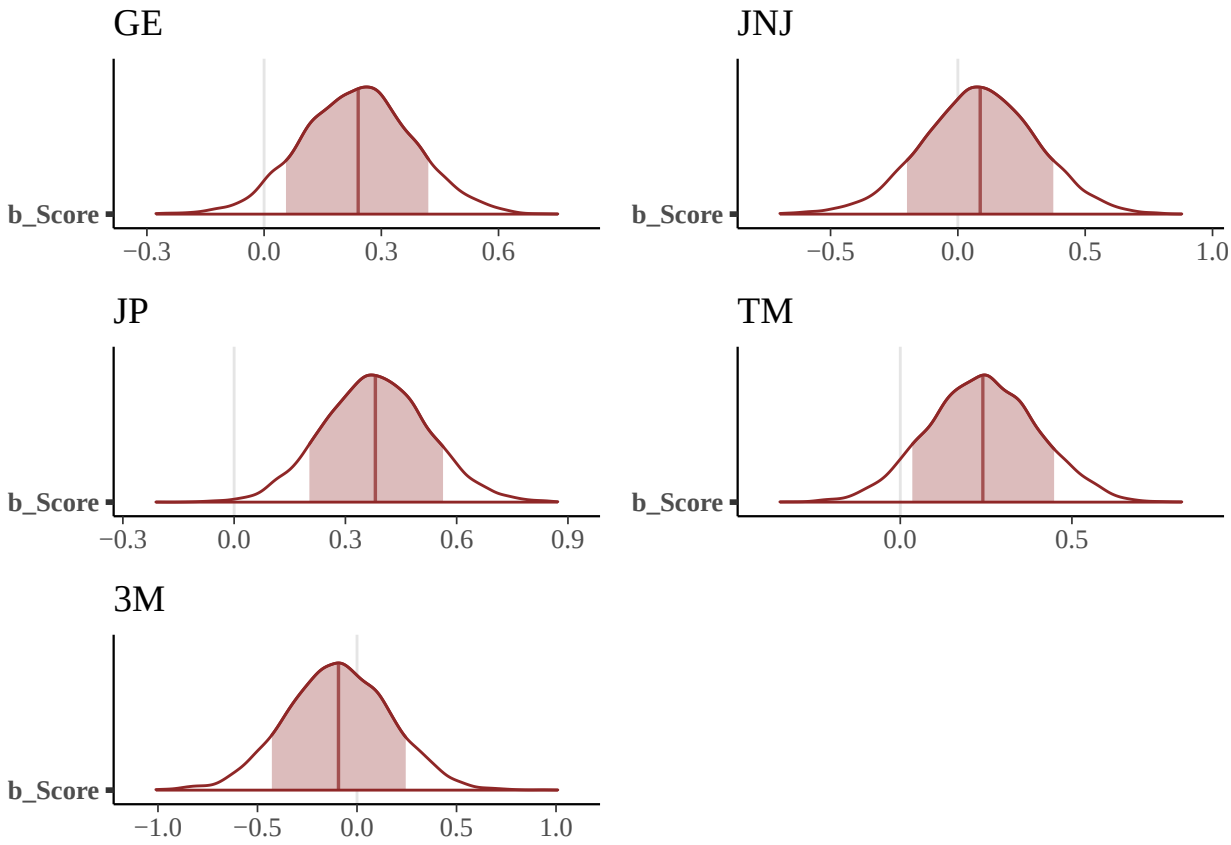
To check for any multi-collinearity problems the following plots reveal if there are any correlations between the predictor variables in our multiple linear regression models.


Apple
b_Score
b_Nasdaq_Close_s
Corr:
−0.124***
b_Score
b_Nasdaq_Close_s
30
20
10
0
1.10
1.09
1.08
1.07
1.06
−0.09
−0.06
−0.03
0.00
1.06
1.07
1.08
1.09
1.10

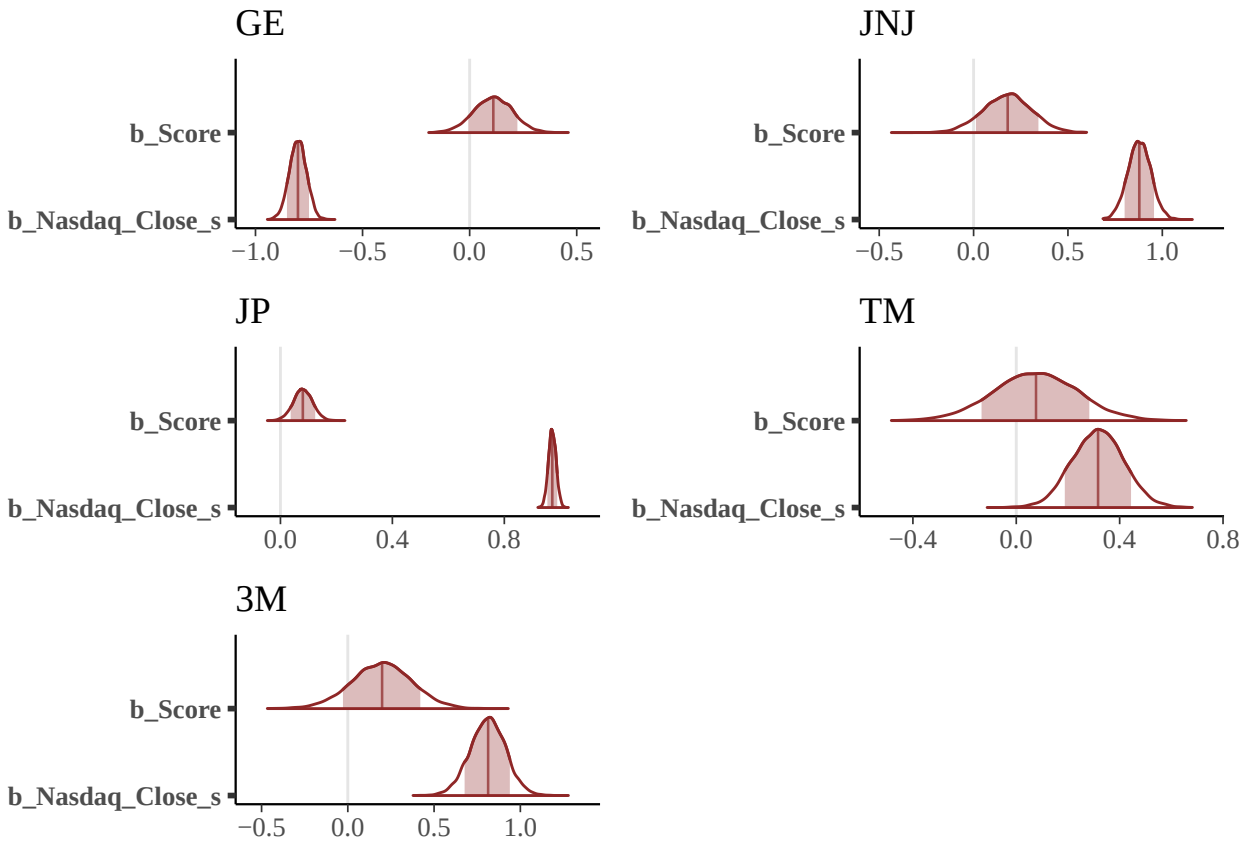

Microsoft
b_Score
b_Nasdaq_Close_s
Corr:
−0.107***
b_Score
b_Nasdaq_Close_s
20
15
10
5
0
1.06
1.04
1.02
1.00
−0.08
−0.04
0.00
0.04
1.00
1.02
1.04
1.06



The following plots show the distribution of the simple linear regression coefficient for Non-Tech stocks.



Now the plots for the distribution of the coefficients of the multiple linear regression models for Non-tech stocks.



Compared with the multiple linear regression for Tech stocks, the strength of the correlation between sentiment and price hasn't reduced very significantly. It is also interesting to note that in the case of the *3M* stock the distribution has shifted a little towards the right (positive x-axis), suggesting that the model now believes that a positive correlation is more plausible when conditioning on extra information like the Nasdaq index value.

Discussion

Based on our analysis we get a variety of different results depending on whether we've used the bing sentiment dictionary or the loughran sentiment dictionary. Our results also vary with the type of financial text data (tweets or articles) we've used and if we've included a control variable or not. Starting with the two sentiment dictionaries, our analysis revealed that the loughran dictionary seemed to do a better job of capturing sentiment, whether it was in the tweets data or the articles data. Furthermore the results from the models that used the loughran sentiment dictionary matched our expectation of a positive correlation between price and sentiment. Also when the loughran dictionary was used, the distribution for the coefficient was more narrower indicating that the model was more certain of the positive correlation between the price and sentiment. We can see this for the amazon stock as follows.

Bing Sentiment

```
##                   2.5%      97.5%
## b_Intercept -0.1057303 0.09879489
## b_Score     -0.1663336 0.32263638
## sigma        0.9238235 1.09420236
```

Loughran Sentiment

```
##                     2.5%      97.5%
## b_Intercept -0.06784615 0.06960458
## b_Score      0.05292970 0.32239162
## sigma        0.94583365 1.05097084
```

In our analysis we found that models based on financial articles seem to suggest a more stronger positive correlation between sentiment and price than those that were based on financial tweets instead. We illustrate this below again for the apple stock.

Tweets

```
##  b_Intercept   b_positive
## -0.187895667 -0.006759534
```

Articles

```
## b_Intercept     b_Score
##  0.01362913  0.22564707
```

Also the distribution for the coefficient is much narrower when using articles than when using tweets. This can be seen as follows. We use apple stocks again to illustrate this phenomenon.

Tweets

```
##                    2.5%       97.5%
## b_Intercept -0.3203897 -0.05048659
## b_positive   0.1942532  0.77105557
## sigma        0.9322432  1.05401362
```

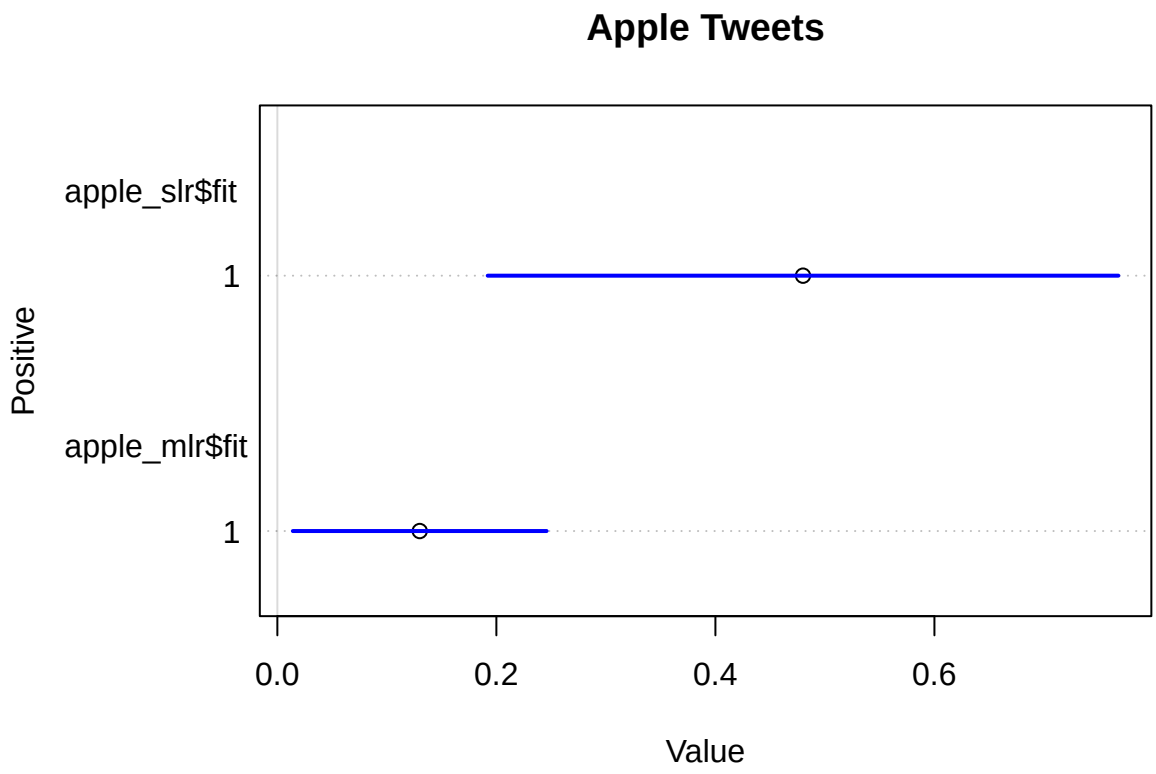
Articles

```
##                     2.5%     97.5%
## b_Intercept -0.02493437 0.0531607
## b_Score      0.13772566 0.3084565
## sigma        0.96749587 1.0229159
```

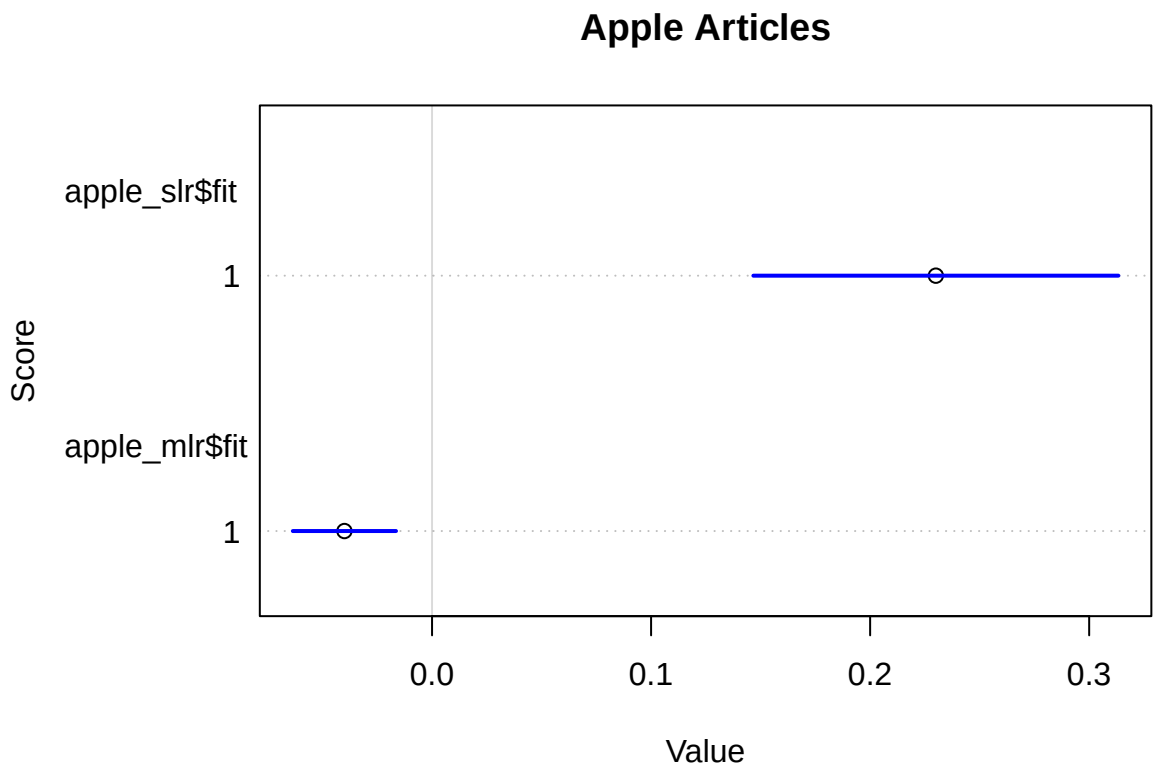
The distribution of the coefficient of the predictor variable for the model using the article data is more narrower and hence the model using the tweets data is more uncertain about the correlation between price and sentiment.

We also learned that including a control variable like $Nasdaq$ reduced the strength of any correlation between price and sentiment that had been found in earlier simple regression models.

Tweets

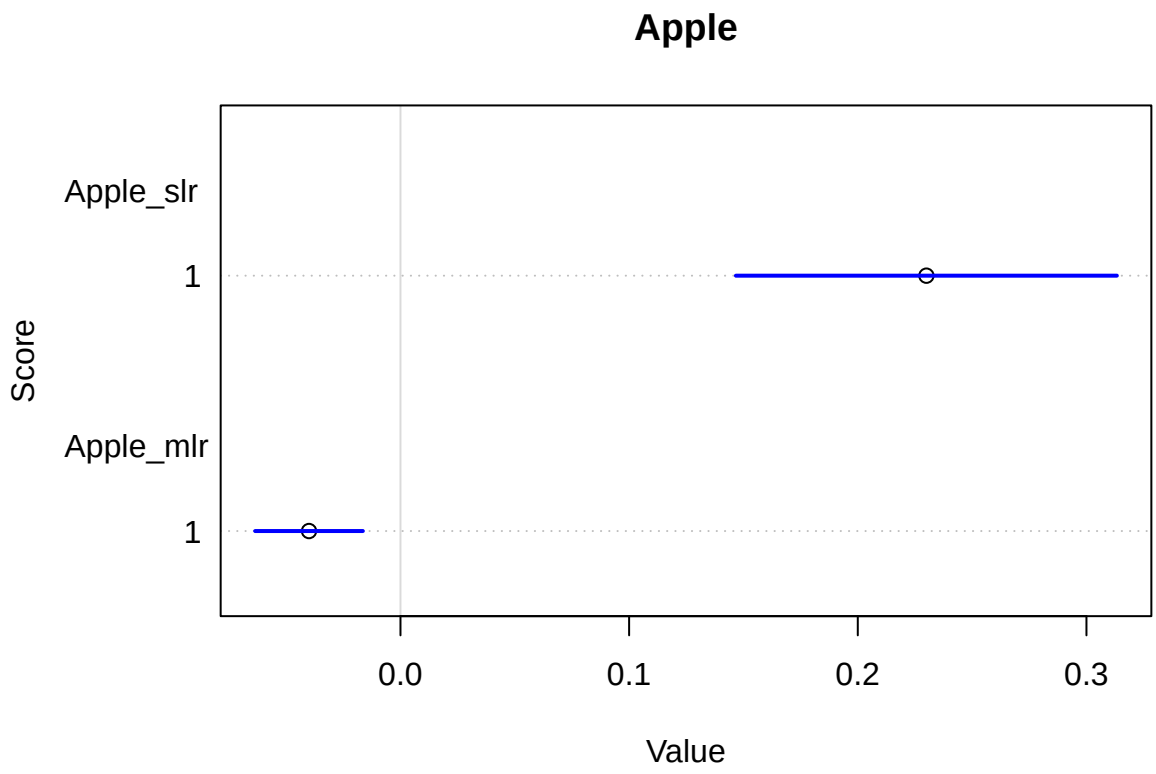

Apple Tweets
apple_slr$fit
1
apple_mlr$fit
1
Positive
0.0
0.2
0.4
0.6
Value

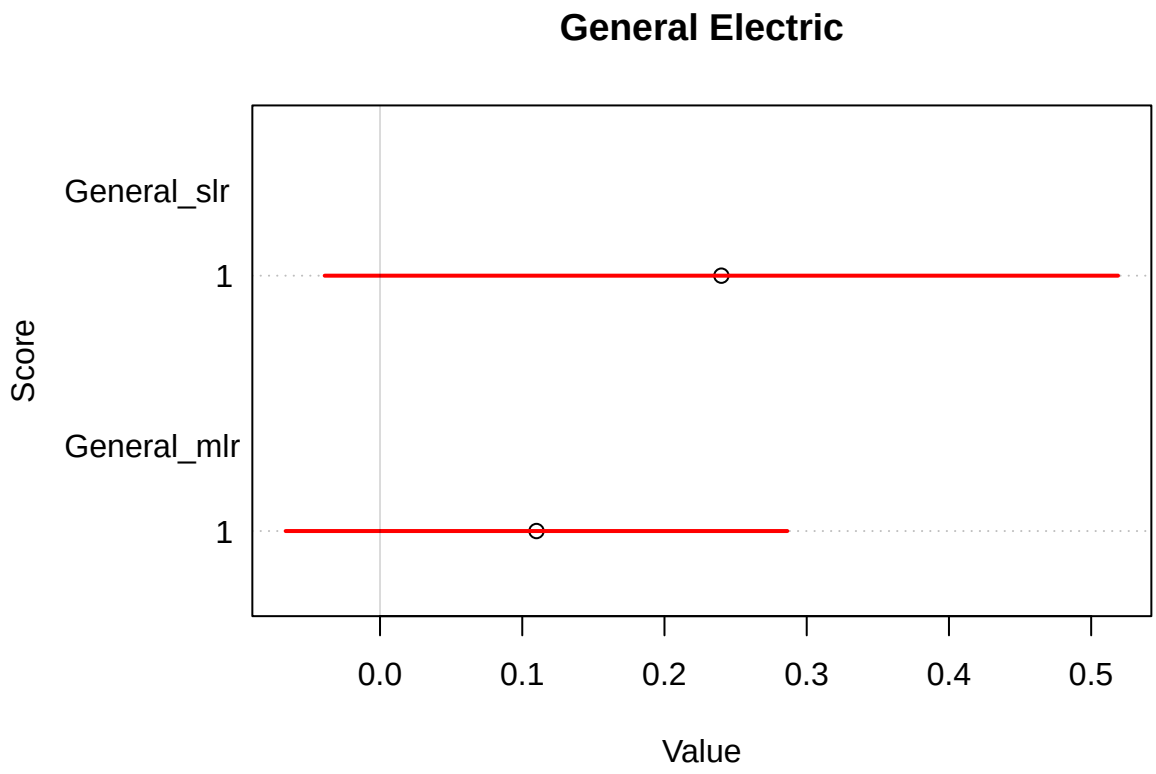
Articles



From the plots above we see that the strength of the correlation between price and sentiment is reduced for both types of data, tweets and articles, when a control variable like $Nasdaq$ is included in the regression.

An interesting result was the difference between multiple linear regression models for tech and non-tech stocks. In both cases, including a control variable like $Nasdaq$ reduced the correlation between price and sentiment, but the reduction was greater for tech stocks than non-tech stocks. This can be see in the following plots.


Apple
Apple_slr
1
Apple_mlr
1
Score
0.0
0.1
0.2
0.3
Value



In the plot above we can see that including a control variable like *Nasdaq* does indeed reduce the correlation between sentiment and price, but the reduction is far greater if it's a tech stock. This suggests that for Non-Tech stocks, even after conditioning on extra information, like a composite market index (Nasdaq in this case), there is still value in knowing the sentiment score of a stock in trying to explain some of the volatility in the stock's price.

One of the main problems with this study is that it uses a very simple approach of one word per token for sentiment scoring. Clearly a sentences sentiment is not simply the sum of it's individual words. It makes more sense to include more words in a single token of text. Although the single token per word approach taken in this study helps to simplify further analysis it compromises on the accuracy of the sentiment score that is eventually calculated. Furthermore when trying to gauge the overall sentiment of a corpus of text , at any point in the text, one cannot simply base it on the individual token at that particular point in the text or even on tokens in the local vicinity of that token of text. It makes sense to incorporate historical information as we go through the text. A further caveat with the approach taken in this study towards sentiment anaylsis is that for a large corpus of text, like the articles data, simply summing over all the uni-grams (one word tokens) can result in the sum averaging out to zero since a lot of positive and negative words can cancel each other out. In such cases it might make more sense to sum over portions of the text like a sentence or a paragraph and then take the average of all those different portions to find a composite sentiment score for the corpus.

Another limitation of this study is the model specification that was used for finding the assosciation between price and sentiment. The model used is a linear model and does not include any transformations of the predictor variables. In the context of this study's goal, which is to understand how sentiment influences(if at all) the price of a stock, a linear model makes sense since it allows for easier inference of the model parameters. However if the goal was more about prediction of stock price then more complex non-linear models are better suited for the task. These non-linear models make parameter inference difficult but they more than make up for it with their high prediction accuracy.

Conclusion

In this study, the goal was to determine whether there was a correlation between a stock's market sentiment and it's market price, and to check the strength of that correlation if it exists at all. This analysis was done in two main parts. The first part involved finding sentiment scores using two different dictionaries (bing and loughran) for the financial text data (articles and tweets). The approach taken towards sentiment analysis is what is known as a tidytext approach *(Silge and Robinson 2016)* which is essentially a framework for Natural language processing that is based on the tidy data principles in R *(Wickham 2014)*. Using this approach allows for easy integration of the sentiment analysis into the rest of the tidy data wrangling workflow.

The second part involves finding the assosciation between the sentiment scores ,computed in the first part, and the price of a stock. This was done in a Bayesian regression framework because in Bayesian regression one not only gets point estimates for our parameters of interest but can instead generate an entire distribution of plausible parameter values for our linear model. The main benefit of this approach is that one can more accurately incorporate any uncertainty about the assosciation between a dependent and independent variable in the parameters of the model. Interfacing with the Stan programming package in R allowed us to approximate the posterior distributions for our parameters.

In the end our analysis revealed that models that used sentiment data based on articles rather than tweets seemed to be more certain of a correlation between price and sentiment. We also found that conditioning on a control variable like the Nasdaq index value reduced the strength of any correlation between stock sentiment and stock price but this effect was more significant for tech stocks than non-tech stocks, suggesting that for non-tech stocks there is still value in knowing the sentiment of a stock even after conditioning on extra information like the Nasdaq index value.

Bibliographic References

1. Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. "A Neural Probabilistic Language Model." The Journal of Machine Learning Research 3 (null): 1137–55.

2. Zhang, G. Peter. 2003. "Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model." Neurocomputing 50 (January): 159–75. https://doi.org/10.1016/S0925-2312(01)00702-0.

3. Qian, Bo, and K. Rasheed. 2005. "HURST EXPONENT AND FINANCIAL MARKET PREDICTABILITY." 2005. https://www.semanticscholar.org/paper/HURST-EXPONENT-AND-FINANCIAL-MARKET-PREDICTABILITY-Qian-Rasheed/0816a5a989c8d2431a6d20076d27c4295c00fb77.

4. Malkiel, Burton Gordon. 1999. A Random Walk Down Wall Street: Including a Life-Cycle Guide to Personal Investing. Norton.

5. Silge, Julia, and David Robinson. 2016. "Tidytext: Text Mining and Analysis Using Tidy Data Principles in R." The Journal of Open Source Software 1 (July). https://doi.org/10.21105/joss.00037.

6. Wickham, Hadley. 2014. "Tidy Data." Journal of Statistical Software 59 (September): 1–23. https://doi.org/10.18637/jss.v059.i10.

7. Lopez de Prado, Marcos. 2019. "The 7 Reasons Most Econometric Investments Fail (Presentation Slides)." SSRN Scholarly Paper ID 3373116. Rochester, NY: Social Science Research Network. https://doi.org/10.2139/ssrn.3373116.

8. Leung, Xi, Jie Sun, and Billy Bai. 2019. "Thematic Framework of Social Media Research: State of the Art." Tourism Review 74 (January). https://doi.org/10.1108/TR-05-2018-0058.

9. Choi, Hyunyoung, and Hal Varian. 2012. "Predicting the Present with Google Trends." Economic Record 88 (s1): 2–9. https://doi.org/10.1111/j.1475-4932.2012.00809.x.

10. Schumaker, Rob. 2010. “An Analysis of Verbs in Financial News Articles and Their Impact on Stock Price,” January.

11. Schumaker, Robert P., and Hsinchun Chen. 2009. “Textual Analysis of Stock Market Prediction Using Breaking Financial News: The AZFin Text System.” ACM Transactions on Information Systems 27 (2): 12:1-12:19. https://doi.org/10.1145/1462198.1462204.

12. Cao, Lijuan, and Francis Tay. 2001. “Financial Forecasting Using Support Vector Machines.” Neural Computing and Applications 10 (May): 184–92. https://doi.org/10.1007/s005210170010.

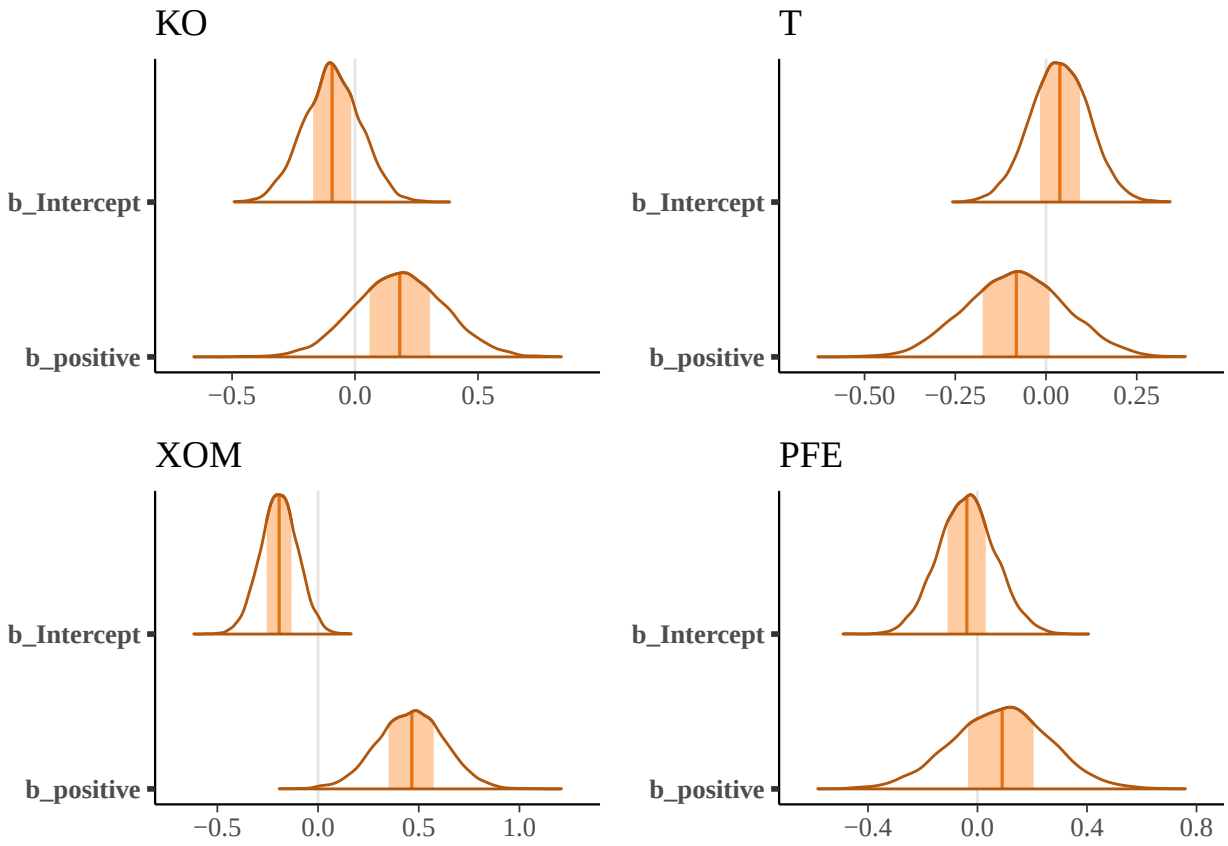
13. Betancourt, Michael. 2018. “A Conceptual Introduction to Hamiltonian Monte Carlo.” ArXiv:1701.02434 [Stat], July. http://arxiv.org/abs/1701.02434.

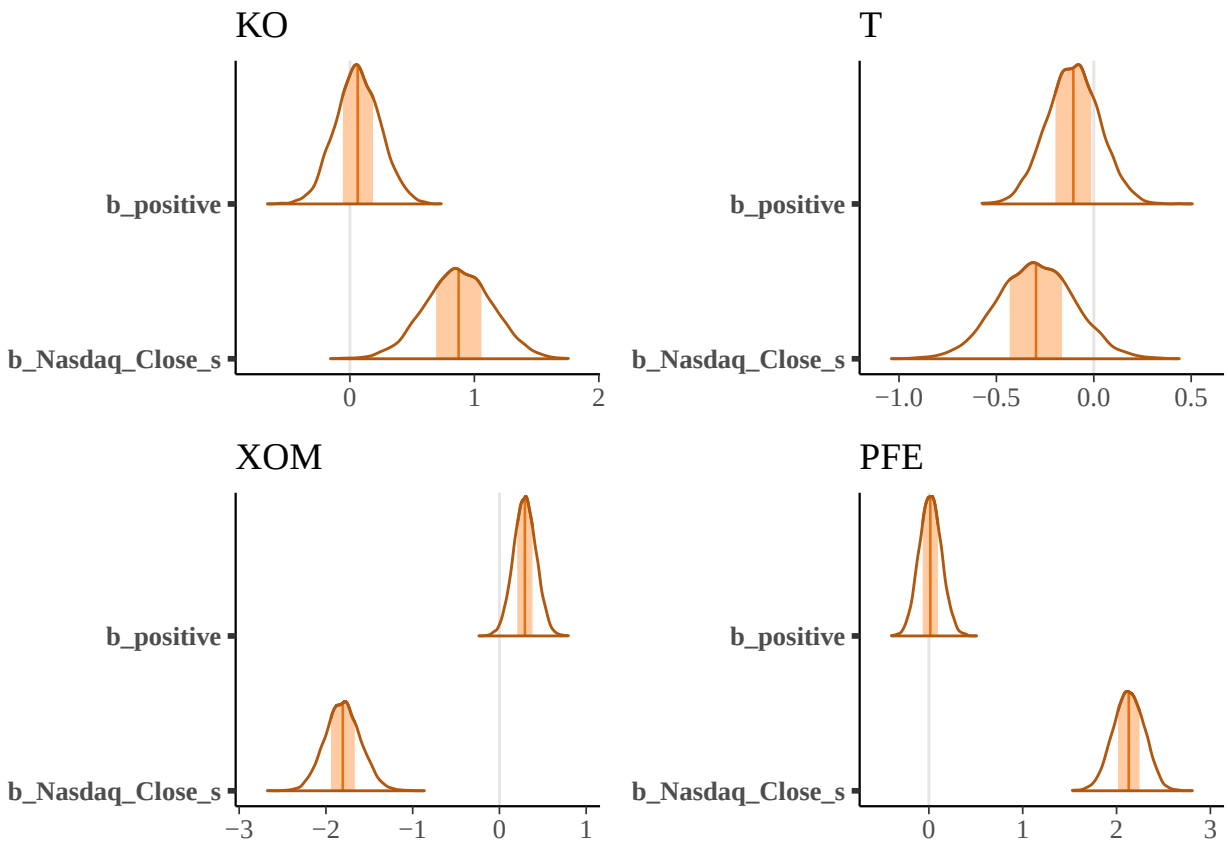
14. Xu, Yumo, and Shay B. Cohen. 2018. “Stock Movement Prediction from Tweets and Historical Prices.” In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1970–79. Melbourne, Australia: Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1183.

Appendices

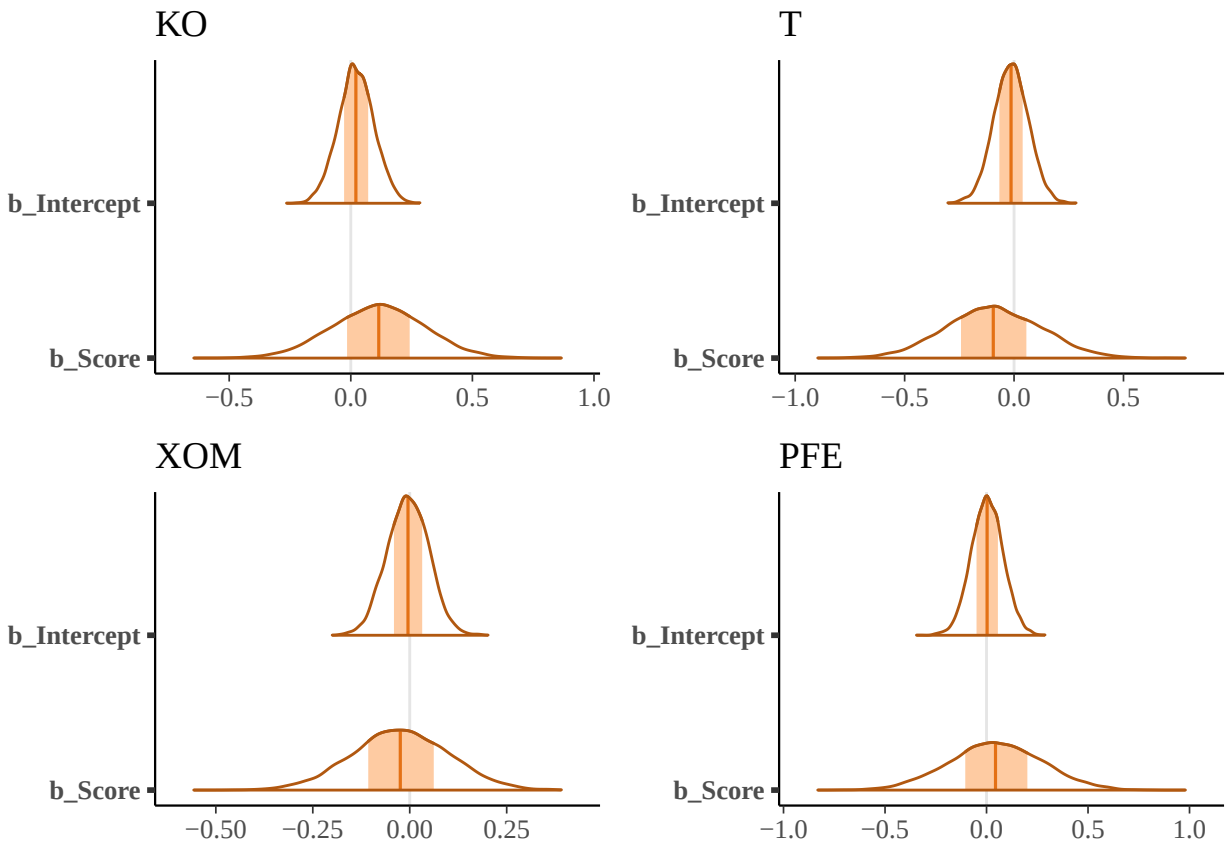
The following are the distributions of parameter values for the remaining stocks.

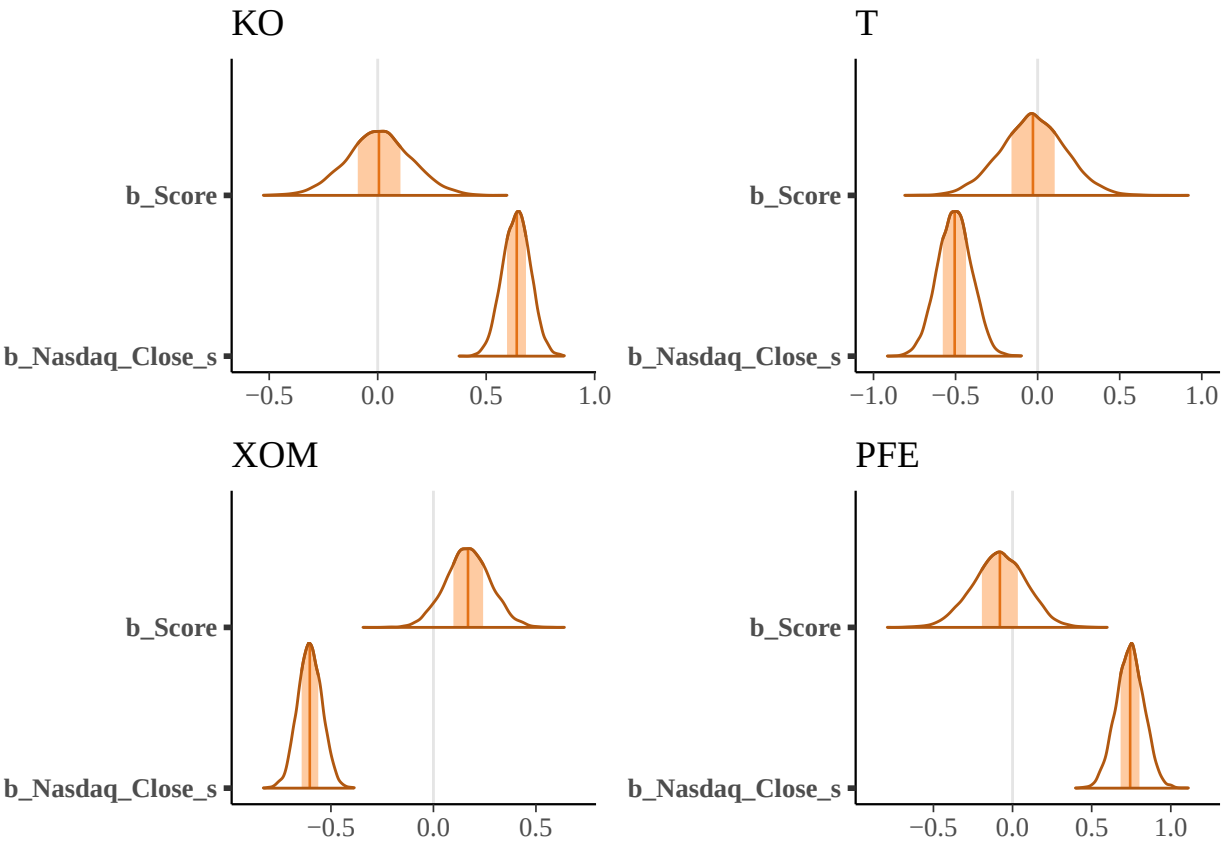
Tweets




KO
b_positive
b_Nasdaq_Close_s
0
1
2
T
b_positive
b_Nasdaq_Close_s
−1.0
−0.5
0.0
0.5
XOM
b_positive
b_Nasdaq_Close_s
−3
−2
−1
0
1
PFE
b_positive
b_Nasdaq_Close_s
0
1
2
3

Articles


KO
b_Intercept
b_Score
−0.5
0.0
0.5
1.0
T
b_Intercept
b_Score
−1.0
−0.5
0.0
0.5
XOM
b_Intercept
b_Score
−0.50
−0.25
0.00
0.25
PFE
b_Intercept
b_Score
−1.0
−0.5
0.0
0.5
1.0


KO
b_Score
b_Nasdaq_Close_s
−0.5
0.0
0.5
1.0
T
b_Score
b_Nasdaq_Close_s
−1.0
−0.5
0.0
0.5
1.0
XOM
b_Score
b_Nasdaq_Close_s
−0.5
0.0
0.5
PFE
b_Score
b_Nasdaq_Close_s
−0.5
0.0
0.5
1.0