WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

Masters Programmes

Dissertation Cover Sheet

Degree Course: MSc Business Analytics

Student ID Number: 2092650

Title: The Influence of President Trump's Twitter Sentiment on his Approval Ratings

Dissertation Code: IB93Y0

Submission Deadline: 17 September 2021

Word Count: 8,803

Number of Pages: 79

"I declare that this work is entirely my own in accordance with the University's Regulation 11 and the WBS guidelines on plagiarism and collusion. All external references and sources are clearly acknowledged and identified within the contents.

No substantial part(s) of the work submitted here has also been submitted by me in other assessments for accredited courses of study, and I acknowledge that if this has been done it may result in me being reported for self-plagiarism and an appropriate reduction in marks may be made when marking this piece of work."

The Influence of President Trump's Twitter Sentiment on his Approval Ratings

By

James Graham Chalfant, V

"I am officially running for President of the United States. #MakeAmericaGreatAgain"

— Donald J. Trump (@realDonaldTrump) Jun 16, 2015

A dissertation submitted in partial fulfilment for the degree of MSc in Business Analytics at Warwick Business School – University of Warwick.

Coventry, September 2021

Acknowledgments

I would like to extend my immense gratitude to my supervisor, Associate Professor Kathryn Hoad, for helping me with this dissertation. Her enthusiasm and obvious love for business analytics was incredibly inspiring. I am also extremely grateful for the effort she put forth to review my work. Additionally, I would like to thank my professors and coursemates at Warwick Business School for constantly being a source of knowledge and help throughout the academic year.

Abstract

The 45th President of the United States of America, Donald J. Trump, showed the world a new form of presidential communication via the social media platform Twitter. Understanding how Twitter affected Trump's presidency holds many implications for future Presidents. This research uncovers the relationship between Trump's Twitter sentiment and his presidential approval ratings.

Understanding which characteristics of Trump's Twitter sentiment influenced his approval ratings is essential because 1) approval ratings give a general idea of how the American populous feel about the President and 2) approval ratings are an excellent attribute to judge presidents for future re-elections (The Effects of Public Opinion, 2021). Understanding the relationship between Trump's approval ratings and his Twitter sentiment first required calculating his average tweet sentiment. Then, Trump's Twitter sentiment was used as a predictor variable in a regression model to predict approval ratings. Furthermore, different subsets of Trump's tweets based on prominent presidential topics were also tested as predictor variables. Regression analysis revealed no significant relationship between Trump's average tweet sentiment or any tweet subsets and his approval ratings.

Contents

Index of Tables

Index of Figures

1. Introduction

1.1 Background

The internet is reshaping many major segments of life, including how presidents manage their terms. For example, in 2009, Bush and Gore used the power of the internet to sway the populous in their favor using email (Kaid, Holtz-Bacha, 2008, 210). Today, American politicians are using a similar tactic via Twitter.

Twitter contains tweets[1], a message posted to Twitter that can have a maximum of 280 characters. These tweets can have text, links, photos, and videos, and anyone can post them with the click of a button. Once someone posts a tweet, anyone who follows the tweet owner or another account that retweeted[2] the original tweet can view it.

President Trump's social media use has set him apart from the earlier presidents. Trump tweeted[8] an astounding 16,433 original tweets during his time in office. In addition, he used his Twitter handle[3] @realDonalTrump to break news, feud with critics, and share opinions. Trump's use of Twitter generated immense media coverage and left many researchers pondering the impact of such presidential conversations (Francia, P.L., 2018) (Enli, G., 2017).

1.2 Objective

In the past five years, there has been no shortage of research covering the impact Trump's use of Twitter had during his presidential campaign. His ability to captivate the U.S. population and generate free media has led many to believe that Twitter is now necessary for presidential candidates.

Table 1 The invisible Primary and the initial Long Odds of the Trump Campaign (Francia, P.L., 2018)

Candidate	Rank: National Endorsements	Rank: Money Raised
Jeb Bush	1	1
Marco Rubio	2	3
Ted Cruz	3	2
Rand Paul	4	6
Chris Christie	5	8
Carly Fiorina	8	5
John Kasich	7	7
Ben Carson	10	4
Mike Huckabee	5	10
Donald Trump	10	9

Trump displayed the power of Twitter during his 2016 campaign. *Table 1* shows that Trump was at a significant disadvantage regarding endorsements and money raised early in the presidential

election. For instance, Hillary Clinton, Trump's opponent for the presidential seat in the White House, raised USD 1.2 Billion compared to Trump's USD 647 Million (Francia, P.L., 2018). Consequently, this enabled Clinton to pour considerable sums of money into television advertising. Despite this, Trump was able to secure his position as the 45th president of the United States. Many academics consider his success to be partially because of his use of Twitter (Enli, G., 2017).

When Trump took office on Jan 20, 2017, his aggressive use of Twitter did not slow. On the contrary, Trump's use of Twitter only increased the longer he was in office. Since the impact of Trump's Twitter during the 45th Presidential Election is well studied and understood, the next logical frontier would be the impact of Twitter on Trump's presidential term. Understanding how Twitter affected Trump's presidency is essential information for future candidates. In addition, understanding which parts of Trump's Twitter influenced his approval ratings is essential because 1) approval ratings give a general idea of how the American people feel about the President, and 2) approval ratings are an excellent attribute to judge presidents for future re-elections (The Effects of Public Opinion | American Government, 2021).

This research aims to uncover the impact that Trump's Twitter sentiment had on his approval ratings.
The Analysis will use sentiment analysis and regression analysis in order to draw conclusions. Additionally, different subsets of Trump's tweets will be identified using automatic keyword extraction and tested using regression analysis.

1.3 Contribution

This research builds upon K. Sahu, Y. Bai and Y. Choi's, (2020) and Colonescu's (2018) work. Using a cross-correlation approach, K. Sahu, Y. Bai and Y. Choi examined the relationship between Trump's Twitter sentiment and his approval ratings. Similarly, Colonescu researched the relationship Trump's Twitter sentiment had on financial markets using regression analysis. Additionally, Colonescu recommended measuring the influence different tweet topics had on some variables for future work. So, using the general research ideas for both studies and the recommendation made by Colonescu, this research will analyze the different influences that subsets of Trump's tweets had on his approval ratings.

1.4 Outline

The rest of the study is organized as follows:
Section 2 examines the literature that helped guide the research. This section will discuss general text analytics, sentiment analysis, keyword extraction, time series analysis, and regression analysis.

Section 3 covers the methodologies used for this research and is structured using the CRISP-DM methodology.

Section 4 covers the analysis used to evaluate the relationship between Trump's average Twitter sentiment and his approval ratings.

Section 5 will discuss the results of the analysis and present ideas for future work.

The bellow diagram shows the flow of the report. Each arrow has an embedded link and can be used to navigate to each respective section.

Figure 1 Dissertation Outline

2 Literature Review and Background

This section discusses past studies and research which parallel the work at hand or influenced the techniques used. Ideas and techniques are introduced at the beginning of each section, and then applications from past studies follow.

2.1 Text Data

Growth in online social media platforms has resulted in a massive surge of textual data (Ittoo, A., Nguyen, L. M., & van den Bosch, A. 2016). Textual data is semi-structured or unstructured data, two of the three general forms of data (Sumathy, K.L. and Chidambaram, M., 2013). Examples of semi-structured and unstructured data include full text, tweets, emails, and other such forms (Sumathy, K.L. and Chidambaram, M., 2013). In contrast, structured data, as the name suggests, is well organized, cleaned, and commonly stored in databases (Sumathy, K.L. and Chidambaram, M., 2013). When working with text data, common issues are synonyms, links, misspelled words, abbreviations, and random punctuations (Provost and Fawcett, 2021). Semi-structured and unstructured data makeup 80% of all electronic data and have become highly valued by most industries (Kaushik, A. and Naithani, S., 2016). The challenges and high value placed on semi-structured data have spawned many advanced algorithms to extract insights from text data, including summarization, extraction, sentiment analysis, and others (Kaushik, A. and Naithani, S., 2016).

Donald J. Trump's presidency has resulted in an explosion of textual data. From the time Trump took office until the end of his term in 2021, Trump tweeted 16,433 times. Trump used this platform to "break news, feud with critics, and even conduct diplomacy," which resulted in an abundance of media attention (Pain, P. and Masullo Chen, G., 2019). Trump's use of Twitter as a political device attracted masses of academics trying to understand the relationship between Trump’s Twitter use and other variables (Pain, P. and Masullo Chen, G., 2019). Typical research topics include the relationship between Trump's Twitter and approval ratings, financial markets, polarization, techno populism, and free media theory (Pain, P. and Masullo Chen, G., 2019). The current study builds upon these past papers by examining how different subsets of Trump’s tweets affected his approval ratings.

2.2 Sentiment Analysis

Sentiment analysis is a subgroup of text analytics that focuses on discovering the hidden emotion behind textual data. Recent sentiment analysis algorithms can detect emotion in text, such as positive and negative, with a reasonable degree of success (Akkaya and Mihalcea, 2009). Analyzing

Twitter data using sentiment analysis is a common area of study. Additionally, a plethora of research explicitly investigates sentiment predictability on some factors (Grgić, D., Karaula, M., Babac, M.B. and Podobnik, V., 2020.). For example, they are using user Twitter sentiment to predict elections.

Sentiment analysis enables researchers to uncover hidden trends and factors in tweets that contribute to Twitter user's influence and popularity. For example, Lauretta-Otero and Cordero-Gutierrz (2016) used sentiment analysis to estimate which factors led to more significant social influence on Twitter. They quantified social influence as the amount of reach[4] and engagement[5] an influencer[6] had (Lahuerta-Otero, E. and Cordero-Gutiérrez, R., 2016.). Next, they compared engagement and reach against the users' average tweet sentiment. The results showed that Twitter users with a more positive Twitter sentiment had a greater social influence (Lahuerta-Otero, E. and Cordero-Gutiérrez, R., 2016.).

Additionally, Grgic, Karaula, Babac, and Podobnik (2020) analyzed the impact of social communication sentiment on public figure approval ratings from 2017 – 2018 (Grgić, D., Karaula, M., Babac, M.B. and Podobnik, V., 2020.). They used tweet sentiment to predict public figures' approval ratings. The study found that Twitter sentiment is a significant predictor of future approval ratings (Grgić, D., Karaula, M., Babac, M.B. and Podobnik, V., 2020.). Furthermore, positive tweets affected future approval ratings for a much shorter time than negative ones (Grgić, D., Karaula, M., Babac, M.B. and Podobnik, V., 2020.).

2.2.1 Lexicon Dictionary

The lexicon dictionary[7] used to evaluate the sentiment of the text is vital to the analysis (Rice, D.R. and Zorn, C., 2021). Typically, for sentiment analysis, a standard dictionary-based approach is used. Pre-constructed dictionaries are easy to use and thoroughly validated, although they have limitations (Rice, D.R. and Zorn, C., 2021). For instance, these standard lexicons may have trouble classifying sentiment in a specific context (Rice, D.R. and Zorn, C., 2021). An example illustrated by Rice and Zorn (2019) would be the interpretation of "bagel" in general and tennis contexts. In tennis, the word "bagel" means losing 6 – 0, which would carry a negative sentiment. For this reason, the business problem needs thorough consideration because each dictionary scales sentiment in its own way.

Constantin Colonescu (2018) employed the text analytics method described by Sigle and Robinson (2017) to measure a relationship between Trump's Twitter sentiment and financial markets. Colonescu uses the "AFFIN" dictionary to evaluate the sentiment of Trump's tweets. Colonescu chose the "AFFIN" dictionary because of the granularity it offers. Unlike other dictionaries, the AFFIN

dictionary measures sentiment on a scale from -5 to 5. In contrast, other dictionaries classify terms as either negative or positive – more on this in the Methodology. Colonescu found a significant correlation between Trump's Twitter sentiment and financial markets.

Another study by Kalyan Sahu, Yu Bai, and Yoonsuk Choi (2020) analyzed the relationship between Donald Trump's Twitter sentiment and his Approval Ratings. They used a machine learning approach to analyze Trump's Twitter sentiment and then evaluated the relationship between Trump's sentiment and his approval ratings with cross-correlation. Kalyan Sahu, Yu Bai, and Yoonsuk Choi found that there is not a strong correlation between Trump's Twitter sentiment and his approval ratings. This research builds upon their work by evaluating how the sentiment of different tweet subsets affects Trump's approval ratings.

2.3 Keyword Extraction

Constantin Colonescu (2018) recommends examining how the sentiment of specific topics affects some variables. Colonescu's recommendation comes from the work of Nguyen et al. (2015). Nguyen, like Colonescu, studied the effect of Twitter sentiment on stock price movements. Although, Nguyen proposed a topic model to classify tweets into different segments and then use the segments as predictor variables. As a result of Nguyen's topic–sentiment model, he achieved 2.7% greater accuracy when determining stock price movements (Nguyen, T.H., Shirai, K. and Velcin, J., 2015).

Nguyen's model is beyond the scope of the current research, although a similar approach will be applied, namely, keyword extraction by parts of speech tagging (POS). Because of the volume of text data available, it is not feasible for an analyst to identify different subjects by hand (Wu, X. and Bolivar, A., 2008, April.). Instead, keyword extraction creates value by identifying relevant information and topics in documents (Bordoloi, M. and Biswas, S.K., 2018.). Keyword extraction algorithms can identify keywords by examining the frequency, centrality, position, and strength of neighboring words in text (Rose, S., Engel, D., Cramer, N. and Cowley, W., 2010.). Discovering the keywords or themes in Trump's tweets via keyword extraction will present terms for segmentation. The analysis for this study uses the 'Udpipe' package for keyword extraction. Additional information about Udpipe is in the Methodology.

Many different industries use keyword extraction. For instance, keyword extraction of customer reviews is a common use of the algorithm. Analyzing customer reviews by hand is an infeasible response to the overwhelming number of responses companies face. As a result, companies must leverage a tool like keyword extraction is to generate prominent topics within product reviews (Raja, 2019). This enables companies to quickly understand the key themes their customers are discussing.

Additionally, keyword extraction is used with microblogging websites. There is a multitude of studies regarding the text analysis of presidential debates. One of interest is Shaban, Hexter, and Choi's (2017) analysis, which aimed to quantify specific topics' impact on the 2016 presidential debate. Their research focused on which topics, in particular, had a lasting impact on the election for Clinton and Trump. They mined these topics from Twitter using many different techniques, including keyword extraction (Shaban, T.A., Hexter, L. and Choi, J.D., 2017).

2.4 Time Series Analysis – The Average Sentiment Method

A time series of Trump's average sentiment was constructed to analyze his Twitter sentiment's impact on his approval ratings. Time series data is dependent on specific points in time. For instance, yearly weather forecasts are time-series data. Time series data of user sentiment on microblogging sites can garner many unique insights (Thelwall, M., Buckley, K. and Paltoglou, G., 2011). Thelwall (2011) explains that microblogging time series are useful because 1) they are reliably time-stamped, which means that temporal analysis can be confidently completed, 2) a large portion of public opinions can be captured because of the ease of use for microblogging sites, and 3) they are public information, so they are easily accessible for researchers.

To analyze the effect of Trump's Twitter sentiment on financial markets, Colonescu (2018) constructed a time series of the President's Twitter sentiment against the movement in specific stocks. Colonescu constructed the time series using the tidyverse package (Wickham and Grolemund 2017) and the average sentiment method. These methods are discussed more thoroughly in the Methodology section. Colonescu used the time series data to measure the correlation between Trump's average Twitter sentiment and financial exchange aggregates, such as the Dow Jones Industrial Average.

Additionally, Sahu, Bai, and Choi (2020) used time series analysis to analyze the relationship between Trump's Twitter sentiment and his approval ratings (K. Sahu, Y. Bai and Y. Choi, 2020). The research found no direct relationship between Trump's Twitter sentiment and job approval ratings (K. Sahu, Y. Bai and Y. Choi, 2020).

2.5 Regression Analysis

Regression analysis was used to analyze the impact that Trump's Twitter had on his approval ratings. Regression analysis is a statistical technique that can analyze the relationship between a dependent and independent variable or predict a change in the dependent variable given a change in the independent variable.

Regression analysis was selected for this study because of its use in Colonescu's analysis of Trump's Twitter and financial markets. First, Colonescu found Trump's daily average sentiment using a dictionary-based sentiment analysis and average sentiment approaches. Then, he used Trump's average sentiment and the Dow Jones Industrial Average Index in a regression model to determine the influence of Trump's average sentiment. Colonescu found that Trump's Twitter sentiment had a short-term effect on the Dow Jones Industrial Average Index (Colonescu, 2018).

2.6 Summary of Literature Review

The literature review covered techniques from other research used in the Analysis, such as text mining, sentiment analysis, keyword extraction, time series analysis, and regression analysis. These techniques were used to determine the influence that Trump's Twitter sentiment had on his approval ratings. Additionally, subsets of Trump's tweets were also tested for influence.

3. Methodology

The CRISP-DM (Cross Industry Standard Process for Data Mining) is used to guide this research methodology. CRISP-DM is a widely used methodology that guides data mining processes (Wirth, R. and Hipp, J., 2000, April.). CRISP-DM is used because of its ability to create structure, consistency, repeatability, and objectiveness within projects (Provost, F. and Fawcett, T., 2013). This research is primarily a text mining problem, which differs from data mining because of the data structure (Provost, F. and Fawcett, T., 2013). The difference in mining methods is seen below in *Figure 3*. Although the difference in structure, CRISP-DM is still a viable methodology for text mining (Silva, E.M., Do Prado, H.A. and Ferneda, E., 2002). Silva, Prado, and Ferneda (2002) argued that CRISP-DM was a promising approach for text mining projects. They made their point by utilizing CRISP-DM for a text mining project. The project encompassed text mining techniques such as word frequencies and counts, keyword extraction, and visualizing themes over time, all of which are present in this research.

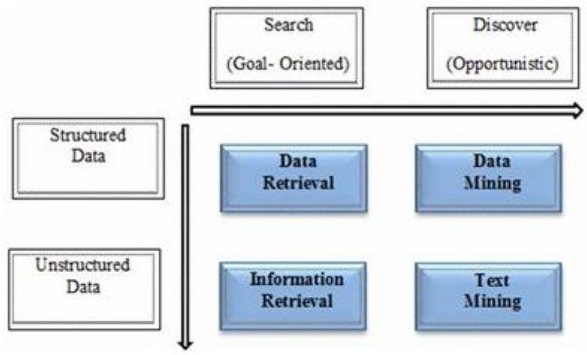


Figure 2 Distinguishing Text Mining from Data Mining (Silva, E.M., Do Prado, H.A. and Ferneda, E., 2002.)

The CRISP-DM process is seen below in *Figure 3*. As depicted, the process diagram is an iterative diagram with six steps, namely, 1) Business Understanding, 2) Data Understanding, 3) Data Preparation, 4) Modelling, 5) Evaluation, and 6) Deployment. The direction and number of arrows denote that iteration is expected. In fact, iteration is encouraged as it helps refine the project. For instance, alterations may be made in the business understanding if limitations are found in the data understanding (Provost, F. and Fawcett, T., 2013.).

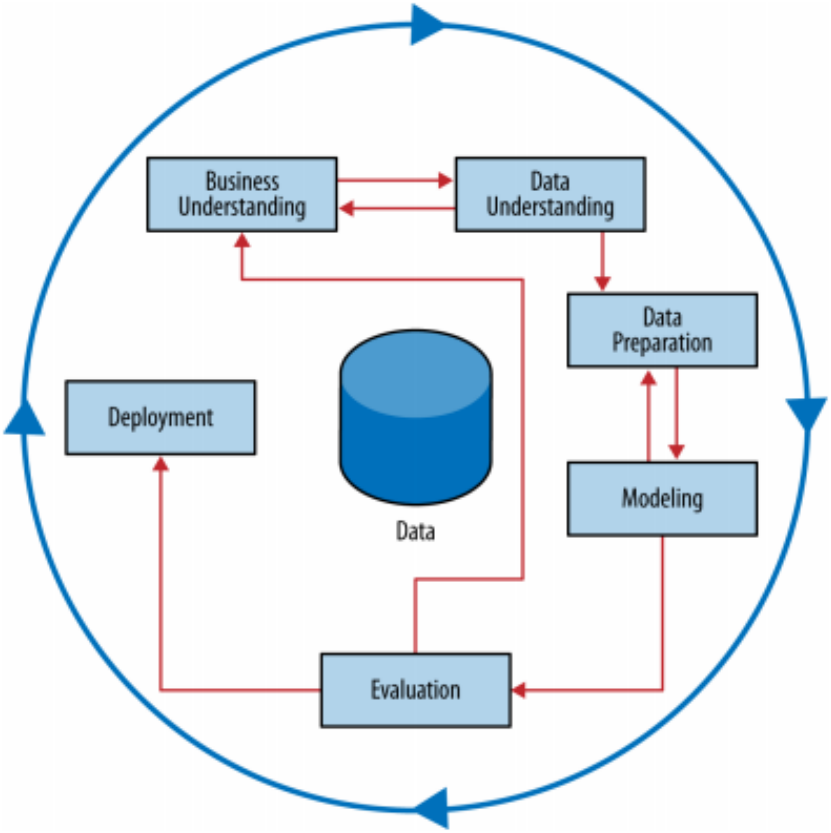


Figure 3 CRISP-DM Cycle (Provost, F. and Fawcett, T., 2013.)

3.1 Business Understanding

The business understanding is the first step in CRISP-DM and is often considered the most important. The business understanding involves framing the business problem as a data science problem (Provost, F. and Fawcett, T., 2013.). Framing the business problem seems straightforward, but as *Figure 3* illustrates, it typically is not completed on the first attempt. The refinement of the business problem is crucial as it forms the foundation for the remaining analysis. In order to develop an adequate business understanding, Provost and Fawcett (2013) recommend the analyst asking themselves what exactly they want to do and how exactly they would do it.

a) What

The Introduction section outlined and discussed the business understanding for this research. However, to reiterate, Trump's use of Twitter as a political device is what many believe partially enabled him to secure his position in the White House, despite being at a significant disadvantage. Moreover, this new form of presidential talk holds many implications for future candidates. So, this research aims to better understand which aspects of Trump's Twitter held influence.

This research aims to understand the connection between Trump's average Twitter sentiment and its influence on his approval ratings. In addition to this, the research will take a more granular approach to past research and investigate how specific subsets of Trump's tweets influenced his approval ratings.

b) How

Trump, who tweeted 16,344 original tweets during his term, has demonstrated a new way to communicate with the populous. Trump's tweets contain many different quantifiable characteristics such as length, polarity, and topic. These characteristics can garner insight into what exactly enabled Trump's Twitter to gain influence. Therefore, the 45th President's Twitter sentiment is of interest to this study.

Trump's Twitter sentiment, or the emotion behind his tweets, gives insight into how he tweeted. A regression of Trump's Twitter sentiment and his approval ratings will determine if the emotion behind his tweets influenced his popularity. Numerous political polling companies produce presidential approval ratings. There are more details on his approval rating data in the Data Understanding section.

Text mining, time series, and data mining techniques are suitable approaches to reach this project's aims. The tidytext approach, created by Sigle and Robinson (2017), is utilized to analyze Trump's Twitter sentiment. Colonescu (2018) and Sahu, Bai, and Choi (2017) used the tidytext approach to analyze Trump's Twitter sentiment on some dependent variables. Additionally, time series analysis was used to analyze and visualize the sentiment of Trump's Twitter and his presidential approval ratings over time. The "average sentiment methods" proposed by Colonescu (2018) were also implemented. Finally, regression was leveraged to determine the impact of Trump's Twitter sentiment on his approval ratings. The techniques discussed above were chosen because of their use in past research, which parallels the current study, allowing for a direct comparison of results.

Furthermore, after the initial analysis has was conducted, the tweets were segmented based on keywords to analyze the impact of different topics' sentiment on Trump's approval ratings. Topic keywords were determined based on prior knowledge and the keyword extraction algorithm provided by Udpipe. Details on this algorithm are discussed in section 3.4.3.

3.2 Data Understanding

The data understanding step in CRISP-DM is for understanding the strengths and weaknesses of the data. Data understanding is an essential step in the data mining process because the data is the

means to reach the desired solution. Other data sets may need to be acquired if the data is inadequate or has too many limitations (Provost, F. and Fawcett, T., 2013.)

Data Collection

This study used two datasets: Trump's approval ratings and tweets written between Jan 20, 2017, and Jan 20, 2021. The website fivethirtyeight.com supplied Trump's approval rating data. FiveThirtyEight is an opinion polling company that focuses on politics, economics, and sports blogging (How Popular Is Donald Trump?, 2021). This particular dataset was of interest because Sahu, Bai, and Choi's (2017) used it to cross-check the dataset they used. Additionally, FiveThirtyEight uses multiple polling agencies' results and averages their results using a weighting system, which considers authenticity (Silver, 2021). So, using the same data set as past research allows for comparability, and FiveThirtyEight's polling data is quite robust as it used the results of other polls.

Trump's tweets were acquired from the Trump Twitter Archive (2021). Trump's Twitter account was permanently banned on Jan 8, 2021, and all of his tweets were removed from Twitter as a result. Because of this, scarping[9] Trump's tweets straight from Twitter was not possible. Fortunately, since 2016, every tweet Trump made was recorded in the Trump Twitter Archive. Thus, a dataset containing all of Trump's tweets was acquired by simply downloading it from thetrumparchive.com as a CSV. This dataset was also used in Colonescu's (2018) analysis of Trump's Twitter influence on exchange markets.

Data Exploration

The approval rating dataset contained 4,377 observations and 13 columns, 10 of which were not relevant to this analysis. From the data set, only the following columns were of value:

- **subgroup:** The Group of individuals who were surveyed. There are "Voters," "Adults," and "All polls." All polls contain both registered voters and likely voters.

- **approve_estimate:** Numeric approval estimates from polls that have been deemed authentic, scientific surveys. Represents the percentage of U.S. citizens who approve of Trump's actions in office. The data is smoothed using a polynomial regression with weights based on each poll's past forecasting accuracy in elections since 1998 (Silver, 2021).

- **modeldate:** the date that the model was built using polling data

The subgroup chosen for the analysis of this study was "All polls," which includes registered voters and likely voters. FiveThirtyEight recommended using All polls because past approval ratings were calculated using all adults, which included likely voters and registered voters.

Of the total approval rating observations, 1,459 are accounted for by "All polls," which was used in the Analysis. Approval ratings began on 2017-01-23, three days after Trump was sworn into office, and ended on 2021-01-20, the last day of his term. The approve_estimate of "All polls" can be seen below in *Figure 4.* From the graph, it is evident that the data is not stationary[12]. Additionally, there does not seem to be any seasonality. Although, there is evidence of a cyclical trend at the beginning of the time series.

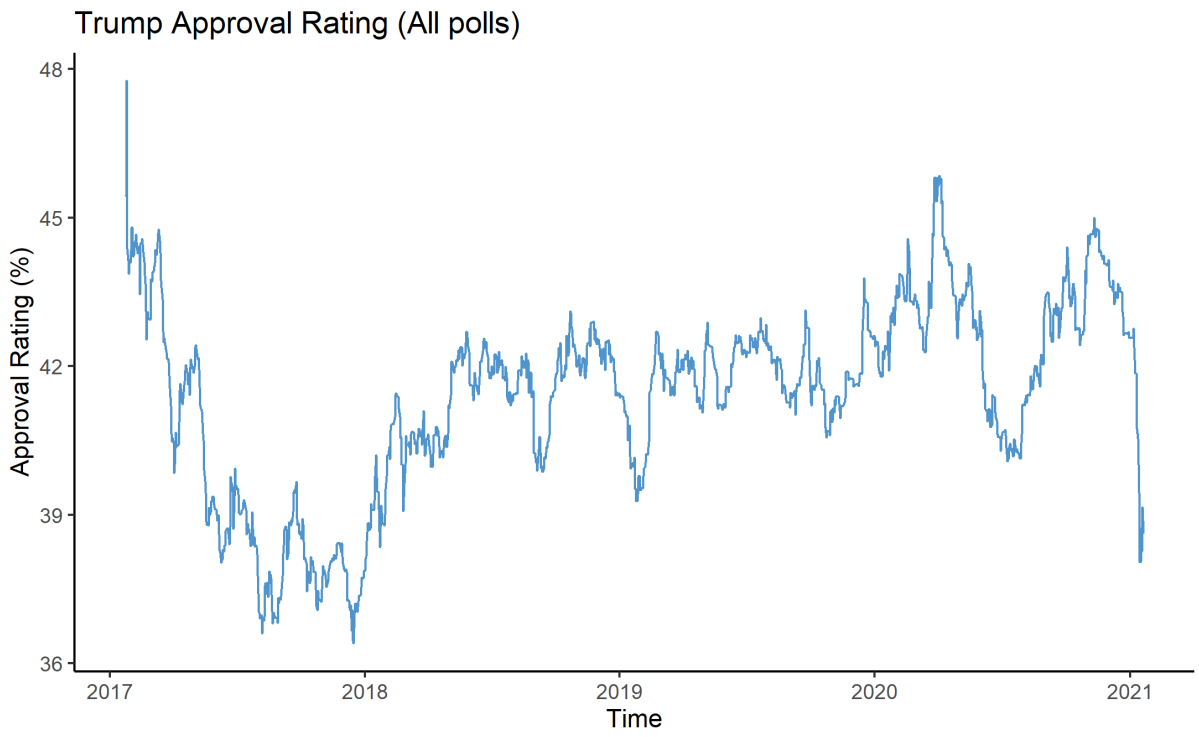


Figure 4 Trump's Approval Ratings with Major Events

Data from the Trump Twitter Archive had 56,571 observations and nine columns. Of the nine columns, only the following were of importance:

Id: The id of the tweet posted.

text: The text within a specified Twitter post.

date: The date the tweet was posted.

The remaining columns, such as "retweets" and "favorites," were irrelevant for the analysis. The analysis of Twitter sentiment over time only required each tweet's text and the time stamp for when it was posted.

3.3 Data Preparation

This section outlines the processes which were used to prepare the data for analysis. Because the approval rating data from FiveThirtyEight is already in a cleaned format, this section focuses on Trump's Twitter dataset.

3.3.1 The Tidytext Approach

Trump's tweets were first cleaned and then formatted in the tidytext format for sentiment analysis (Sigle and Robinson, 2021). Twitter data requires extensive cleaning because it contains many unnecessary words and symbols that do not impact the analysis. *Figures 5 and 6* show a sample of Trump's tweets before and after cleaning. The figures show the removal of unnecessary symbols, conversion of all text to lowercase, and the removal of punctuation, all of which are necessary for the tidytext format (Sigle and Robinson, 2021). Specific examples of unnecessary values are retweets, mentions[13], and image files.

text <chr>
Republicans and Democrats have both created our economic problems.
I was thrilled to be back in the Great city of Charlotte, North Carolina with thousands of hardworking American Patric
RT @CBS_Herridge: READ: Letter to surveillance court obtained by CBS News questions where there will be further dis
The Unsolicited Mail In Ballot Scam is a major threat to our Democracy, & the Democrats know it. Almost all rec
RT @MZHemingway: Very friendly telling of events here about Comey's apparent leaking to compliant media. If you r
RT @WhiteHouse: President @realDonaldTrump announced historic steps to protect the Constitutional right to pray i
Getting a little exercise this morning! https://t.co/fyAAcbhbgk
https://t.co/4qwCKQOiOw
https://t.co/VlEu8yyovv
https://t.co/z5CRqHO8vg

Figure 5 Uncleaned Twitter Data

text <chr>
republicans and democrats have both created our economic problems
i was thrilled to be back in the great city of charlotte north carolina with thousands of hardworking american patriots
the unsolicited mail in ballot scam is a major threat to our democracy the democrats know it almost all recent electio
getting a little exercise this morning

Figure 6 Cleaned Twitter Data

Trump's tweets were also filtered in addition to being cleaned. First, all of Trump's retweets were filtered out. This study intends to understand Trump's publications and their impact on his popularity, not other authors' influence. Second, Trump's tweets were filtered to keep only tweets

that were created during his term. This study pertains solely to Trump's presidency, so tweets that were not between the dates 2017-01-23 and 2021-01-20 were removed. After Trump's tweets were filtered for retweets and his term dates, the initial 56,571 tweets were distilled down to 16,433.

Trump tweeted 16,433 original tweets while he was in office between 2017-01-23 and 2021-01-20. *Figure 7* illustrates Trump's Twitter behavior over his time in office. *Figure 7* shows that Trump became more active on Twitter over his term. Additionally, the multiplicative nature of his Twitter activity indicates that his Twitter behavior also became more variable.

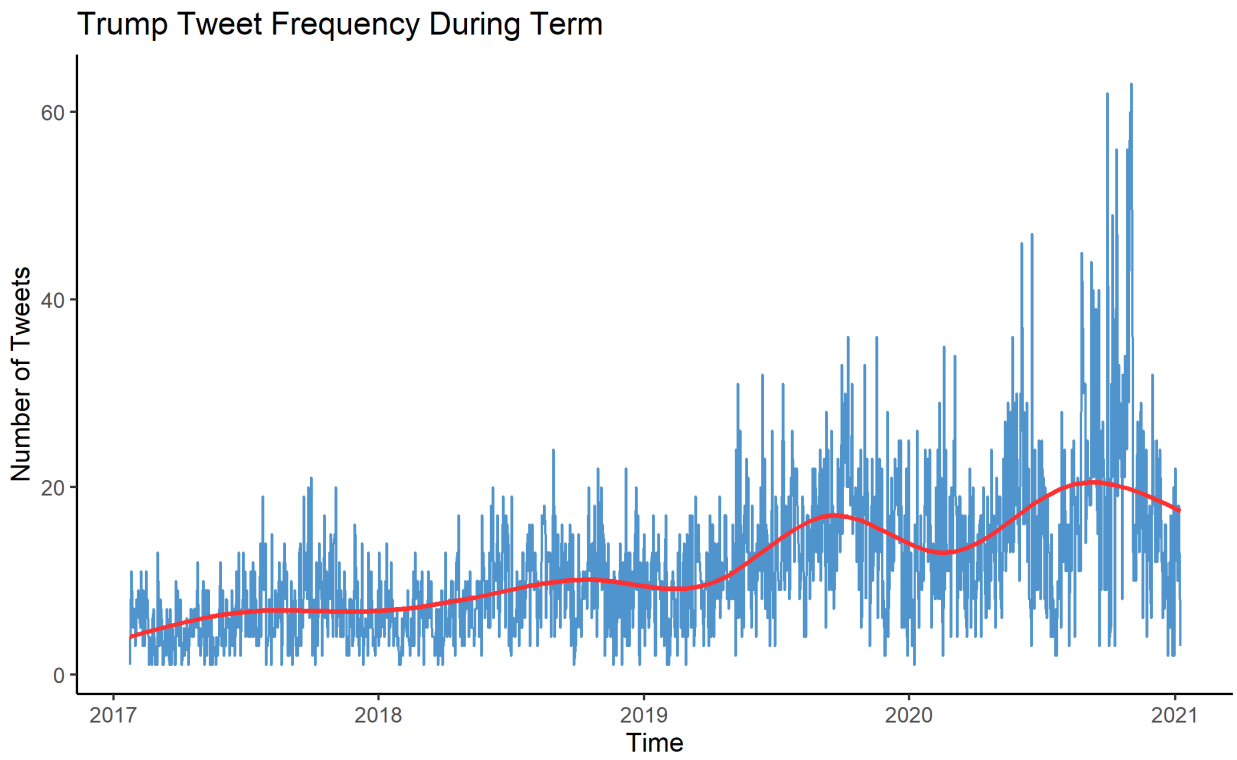


Figure 7 Trump's Tweet Frequency During His Term

After cleaning Trump's tweets, Sigle and Robinson's (2017) tidytext approach was applied. This same method was used in Constantin Colonescu's (2018) analysis of Trump's Twitter sentiment on financial markets. Below, *Figure 8* shows a flow chart of the tidytext format.

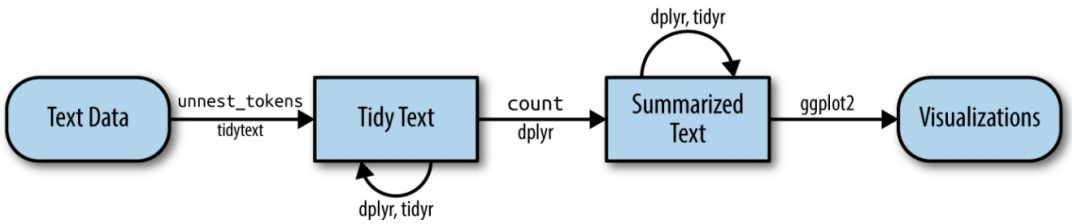


Figure 8 Flow of Tidytext Format (Sigle and Robinson, 2021)

The tidytext format is described as one-token-per-row. A token is a word from a text. Tokenization is the process of splitting text into tokens. Each token has an associated ID which indicates the tweet each token belongs to. This conforms with the tidytext format and allows the data to be easily manipulated with a consistent set of tools. Once all tweets are in the tidytext format, the data can quickly be processed. Stop words are removed after tokenizing. Stop words are words that frequently appear in text and do not add value to the analysis. Common stop words are “the,” “and,” “a.” The tidytext package comes equipped with a list of stopwords. This list is anti-joined[10] with the tokenized words to easily remove stop words.

3.4 Modelling

The Modeling stage of CRISP-DM discusses the data mining and text mining techniques used in the Analysis. This stage outlines algorithms used, parameters chosen, and a general assessment of models and techniques.

As mentioned previously, the modeling section will examine tools and techniques used to measure the impact Trump’s Twitter sentiment had on his presidential approval ratings. First, sentiment analysis was used to measure the emotions behind Trump’s Tweets. Next, his average Twitter sentiment was used as a predictor variable in a regression with approval ratings as what was being predicted. Furthermore, after the initial regression analysis, Trump’s tweets were segmented based on significant events or topics from his presidency. Prominent topics were identified using previous studies of Trump’s presidency, word frequencies, and keywords extracted with UdPipe. The segmented tweets were then used as predictor variables in a regression model. This drew inspiration from Colonescu’s (2018) analysis of Trump’s Twitter sentiment on financial markets. Colonescu recommended future research to focus on different segments of Trump’s tweets.

3.4.1 Sentiment Analysis

Sentiment analysis is a very useful tool that can automatically measure the emotion in online text. Sentiment analysis’s usefulness has promoted its widespread study, which has resulted in the creation of many different methods and algorithms (Thelwall, M., Buckley, K. and Paltoglou, G., 2011). There are three common sentiment analysis approaches: full-text machine learning, lexicon-based methods, and linguistic analysis (Thelwall, M., Buckley, K. and Paltoglou, G., 2011). The machine learning approach involves an analyst annotating a set of text that is used to train the algorithm. Once trained, the algorithm is given unseen text and tasked with predicting its sentiment (Thelwall, M., Buckley, K. and Paltoglou, G., 2011). A lexicon approach utilizes a list of text which has

a preassigned sentiment from an analyst. The linguistic approach often utilizes a lexicon in conjunction with the grammatical structure of text to predict polarity. The grammatical structure leverages sentence context to predict polarity (Thelwall, M., Buckley, K. and Paltoglou, G., 2011). For this research, the lexicon approach was used. The lexicon approach was chosen because Colonescu (2018) used this approach. Following a similar method as Colonescu allows for comparability and consistency in work.

Performing sentiment analysis using a lexicon approach is a straightforward task. Following the tidytext approach, the analyst must choose the appropriate lexicon or sentiment dictionary. There are three general lexicons, namely the "AFINN," "Bing," and "NRC" lexicons. The NRC and Bing lexicons both categorize words in a binary fashion. For instance, the Bing lexicon categorizes words as either "positive" or "negative" (Sigle and Robinson, 2021). This research used the AFINN dictionary. The AFINN library assigns sentiment on a scale of -5 to 5, therefore allowing for a more granular quantification of sentiment.

Additionally, this scale is the most adequate for constructing a continuous time series (Colonescu, 2018). The tidytext sentiment analysis approach can be seen below in *Figure 9*. The only difference between the tidytext sentiment analysis approach and the tidytext approach is that the tokens are inner-joined[11] with the analyst's specified lexicon.

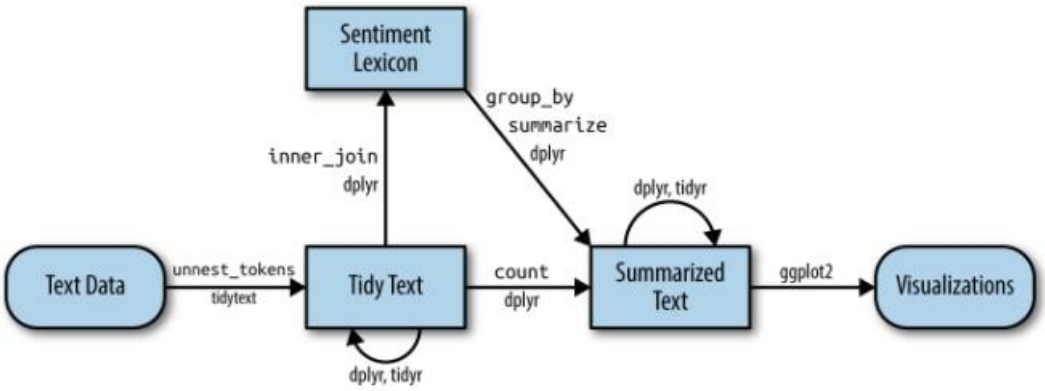


Figure 9 Tidytext Sentiment Analysis (Sigle and Robinson, 2021)

After tokens are joined with the specified lexicon dictionary, the tokens can be rearranged to explore trends in sentiment.

3.4.2 The Average Sentiment Method

The average sentiment of each day is found after the sentiment of each token is computed. Colonescu (2018) recommends finding the average sentiment for each day, thus allowing the average sentiment to be plotted over time.

Positive and negative sentiment for each date is summed together to compute the total sentiment per day. Then, the summed sentiment for each day is divided by that day's number of tokens to compute the average sentiment.

3.4.3 Topic Segmentation

Trump's tweets were segmented to determine if specific topics have a more significant impact on his approval ratings than others. Trump's prominent Twitter topics were found using the topic analysis results from past research, Twitter word frequencies, and keyword extraction algorithms.

First, significant topics were identified by examining past research regarding Trump's prominent topics. Topic analysis of Trump's presidency is a common area of research. Topics from past research did not determine the topics used to segment Trump's tweets. Instead, they were used to complement the keyword extraction, thereby adding legitimacy to the keywords found.

Next, Trump's most frequently used words were identified by rearranging his Twitter tokens. The reason for doing this was identical to examining research that analyzed Trump's major topics. Trump's most frequently used words did not determine what his major topics were. Instead, they added legitimacy to the topics found through keyword extraction.

Lastly, keyword extraction using the UDpipe package was used to identify keywords. The Udpipe package offers robust algorithms which can rapidly identify significant keywords within a document. Keyword extraction is done by an analyst who specifies which kinds of co-occurrences to extract, for instance, noun and adjective combinations. This study used simple noun phrases to find key phrases as noun phrases comprise vital information within text documents (Kaur, J. and Gupta, V., 2010).

Once prominent topics were identified, Trump's tweets were subset using the "grep" function. The grep function searches strings for character matches specified by an analyst. So, Trump's tweets were filtered by words pertaining to the major topics discovered.

3.4.4 Regression Model

Regression analysis measured the relationship strength between Trump's Twitter sentiment and his approval ratings. Regression was chosen because of its use and success in Colonescu's (2018)

research about the influence of Trump's Twitter sentiment on financial markets. *Figure 10* displays a simple linear regression with only a dependent and independent variable has five parts.

$$y = a + bx + \varepsilon$$

Figure 10 Regression Equation(Storm, K., 2019)

$\boldsymbol{y}$ is the dependent variable or what is being predicted. This value would be Trump's approval ratings.

$\boldsymbol{a}$ is a constant value that indicates the value of y when x is equal to zero. To illustrate, a is Trump's approval ratings when his average Twitter sentiment is zero ($x = 0$).

$\boldsymbol{b}$ is the coefficient of x. b is the slope of the regression line and determines how much y changes for a given change in x. A larger b value would result in a more significant change in y for a given x value.

$\boldsymbol{x}$ is the value of the independent variable. $\boldsymbol{x}$ is what is predicting or explaining Trump's approval ratings, y.

$\boldsymbol{\varepsilon}$ is the error term when predicting y given a value of x. Regression analysis relies heavily on the error terms for assumption checking. Additional information on this is in the Analysis.

Fahrmeir, L., Kneib, T., Lang, S. and Marx, B. (2007) book, Regression, is an excellent resource for additional information on regression analysis.

Below, *Figure 11* shows a scatterplot of Trump's average sentiment plotted against his approval ratings. There is also a line of best fit plotted through the points. This line was created using a linear model (LM). These are the same points the regression model in the Analysis uses.

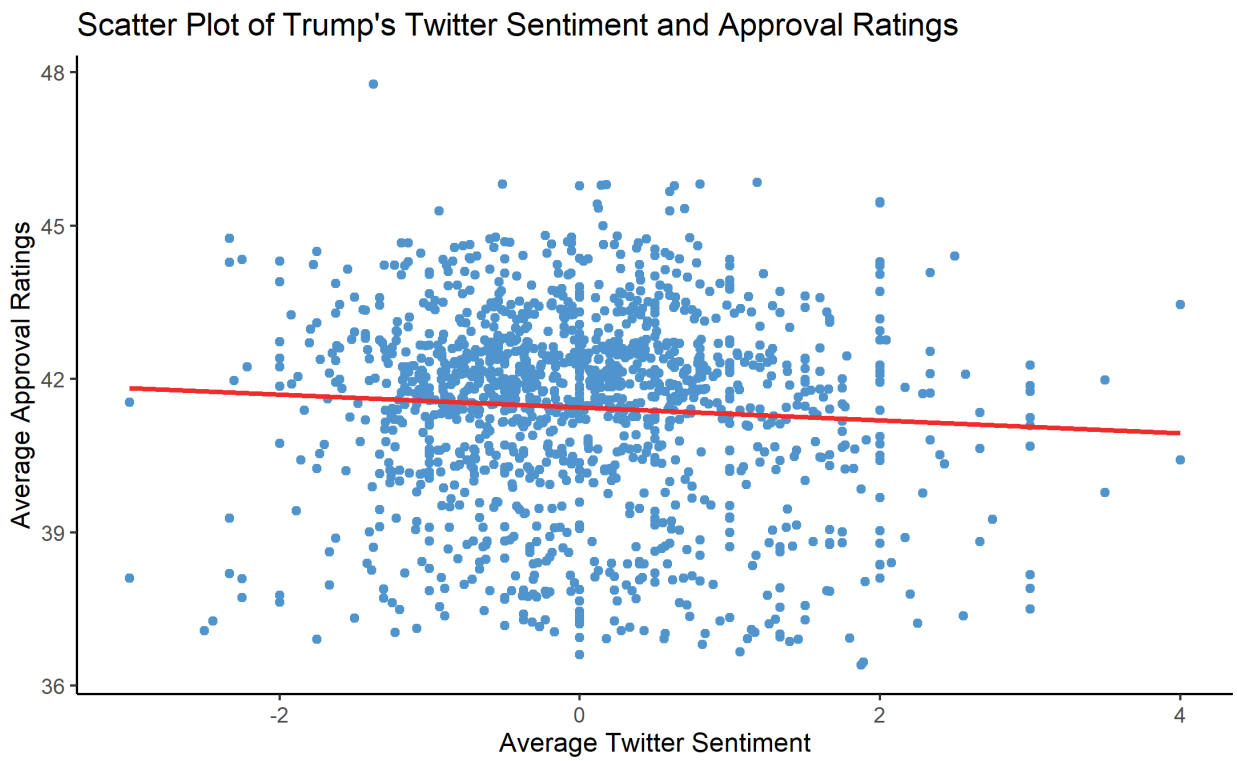


Figure 11 Scatterplot of Trump's Twitter Sentiment and Approval Ratings

3.4.5 Regression Model Output

RStudio's Linear Model (LM) produces a p-value, F-statistic, R-squared, and Residual Standard Error, all of which are important when interpreting a model's strength.

First, regression analysis measures the importance of each independent variable using a p-value. P-values test the null hypothesis that a model's coefficient equals zero or that the independent variable does not affect the dependent variable (Rego, 2021). If an independent variable has a p-value < 0.05, the null hypothesis is rejected at the 5% level of significance, and there is strong evidence that the independent variable explains some of the variations in the dependent variable. In summary, a p-value < 0.05 shows that the independent variable is an excellent addition to the regression model.

Second, the regression model produces an F-statistic value. Similar to the P-value, the F-statistic is a good measure to determine if there is a relationship between the predictor and response variable. The F-statistic compares a model with no predictors to the model the analyst specifies. This comparison tests the null hypothesis that the model specified by the analyst and an intercept-only model are equal. If the two models are the same, then the F-statistic would be equal to one. So, a large F-statistic greater than one is preferred.

Third, a regression model produces an output that describes the amount of variance in the dependent variable explained by the independent variable. This measure is known as the r-squared value (R^2) and it measures how well the model fits the data. The r-squared measure has a range of 0% - 100%. Generally, a higher r-squared is preferred. Although, a low r-squared does not mean conclusions cannot be drawn.

Lastly, the standard error shows how well the model fits the data. The standard error represents the average distance between the observed values and the regression line. Thus, the standard error measure explains how wrong the regression is on average using the response variable's units.

3.5 Evaluation

Conclusions from the model results cannot be drawn without a proper evaluation of modeling techniques. Therefore, model results were evaluated using regression residual plots. These plots display the errors associated with a regression model. Residual errors must be normally distributed, have a zero mean, be homoscedastic, and be independent (Rego, 2021).

Model residuals were evaluated after modeling. There are multiple different remedies to rectify a failed residual test, including a log or square root transformation of the data. Specific remedies for each broken assumption were not discussed. Instead, if an assumption was broken in the Analysis section, proper action was taken to fix that issue.

3.6 Methodology Summary

Using the CRISP-DM framework, the methodology explained how the aims for this study came to fruition and how the research will reach those aims. More specifically, the methodology explained how sentiment analysis, time series analysis, keyword extraction, and regression analysis were used to measure the influence Trump's Twitter had on his approval ratings.

4. Analysis

The analysis implements the techniques discussed in the methodology. In addition, the analysis closely mirrors Colonescu's (2018) and Sahu, Bai, and Choi's (2020) work for comparability. Results produced in the Analysis will be discussed further in the Conclusion.

4.1 Sentiment Analysis

Following the methods outlined in the Sentiment Analysis section of the Methodology, Trumps' tweets were tokenized and then inner-joined using the AFINN dictionary. Then, the tokens' average sentiment was plotted over Trump's term by date, thereby adhering to Colonescu's Average Sentiment method.

Figures 12 and 13 show two different visualizations of Trump's Twitter. The first is a word cloud which illustrated Trump's most frequently used positive and negative words. The Wordcloud displays word frequency as the size of the word and polarity with color. Thus, "fake" was the most frequently used negative word Trump tweeted.

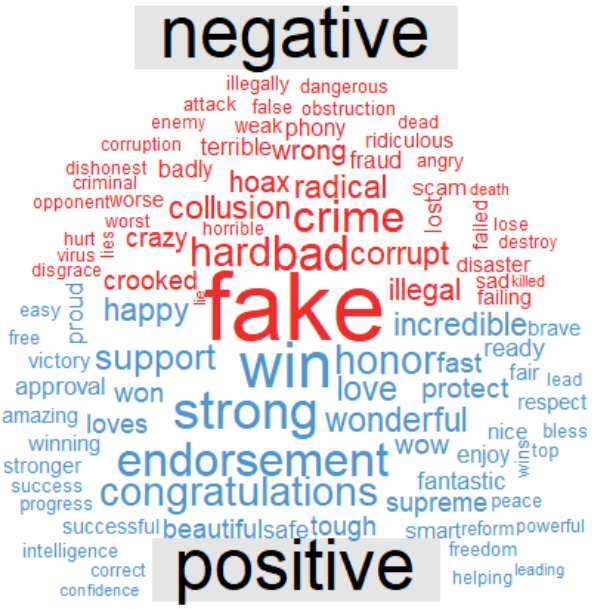


Figure 12 Word Cloud of Trump's Most Frequently used Positive and Negative Words

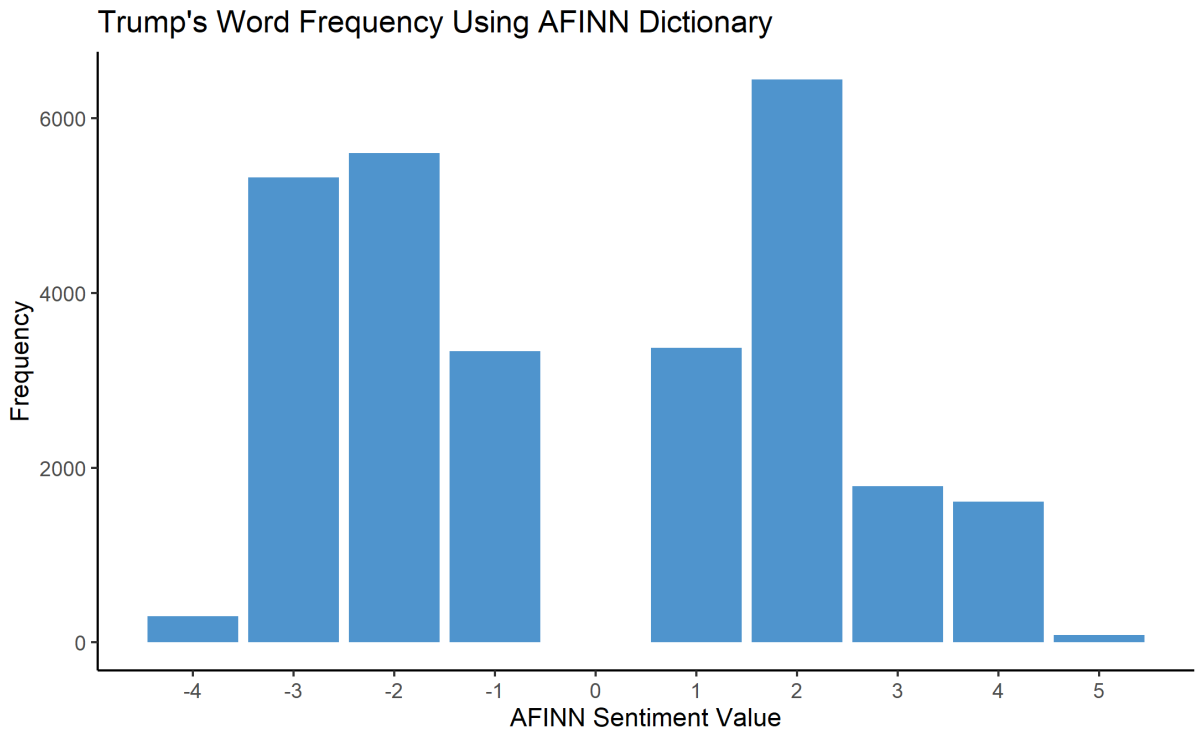


Figure 13 Trump's Word Frequency Using AFINN Dictionary

Figure 13 shows the distribution of Trump's word frequencies scaled by their sentiment value. The AFINN dictionary scales words on a range of -5 to 5. Negative values are associated with words with negative sentiment. *Figure 13* shows that Trump tweeted positive and negative words almost equally. Additionally, the presence of the value '5' indicates that Trump tweeted some extremely positive words. Conversely, the absence of '-5' indicates that he did not tweet any extremely negative words.

4.2 Regression Assumptions

The data used in a time series regression model must be stationary. Following Colonescu's (2018) work, stationarity is tested through visual inspection, augmented Dickey-Fuller test (ADF), and ACF plots. Stationarity is needed because of Granger and Newbold's (1947) work. They found that regression modules that use non-stationary data give spurious results (Granger, C.W. and Newbold, P., 1974).

Trump's approval rating data in *Figure 14* displays a trend. The first half of the time series displays a strong cyclical trend. Then, the data after ~ 2018-06-01 displays positive and negative runs. Overall, the data does not seem random, nor does it seem to have a constant mean. Next, to confirm the visual inspection, an augmented Dickey-Fuller test is was used. The augmented Dickey-Fuller test is a unit root test that tests stationarity. If the test returns a p-value < 0.05, the null hypothesis is

rejected, and the time series is deemed stationary. An ADF test of Trump's approval ratings resulted in a p-value of 0.03425, which is considered significant. Finally, an autocorrelation plot was produced. The ACF plot in Appendix I indicates a clear trend. Thus, the visual inspection and ACF plots both confirmed non-stationarity. Next, Trump's approval rating data was differenced to rectify the issue of non-stationarity. First-order differencing of Trump's approval ratings created stationary data and can be seen in Appendix II.

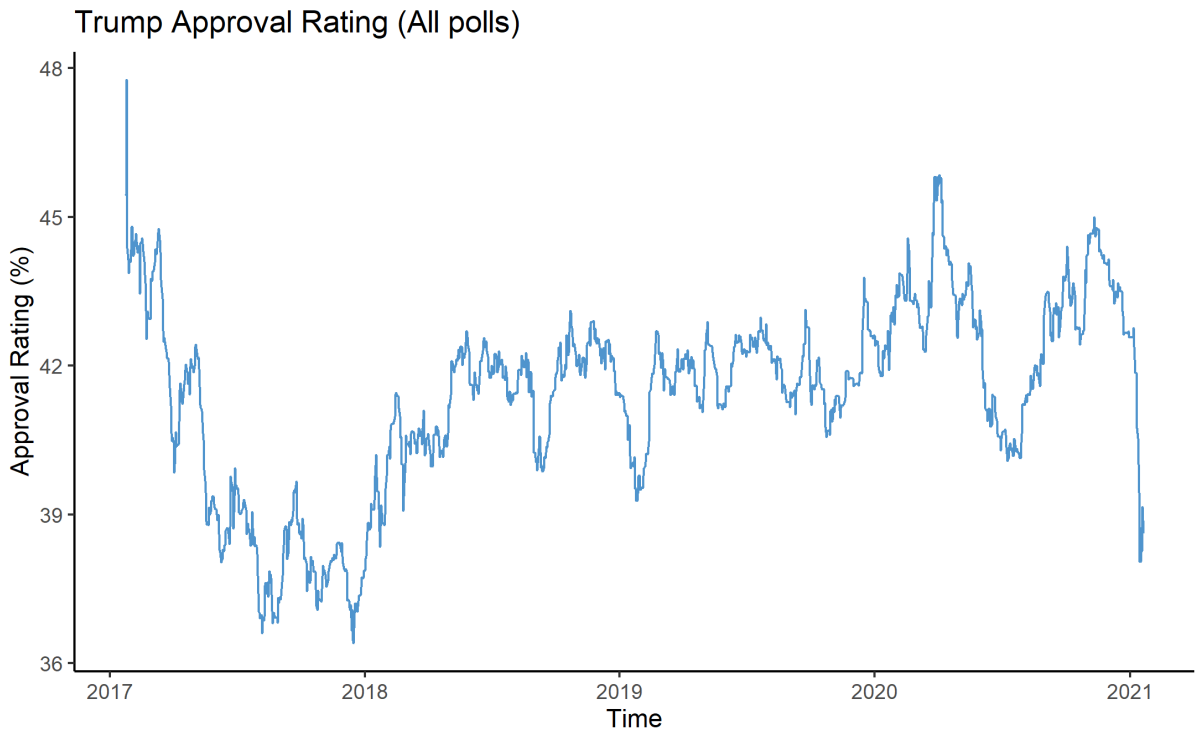


Figure 14 Trump's Approval Ratings

Figure 15 plots Trump's daily average sentiment over time, and inspection reveals that the time series is stationary. The series is random with no predictable trend. The volatility in the data seems constant. The sum of the average sentiment per day was 118.3038, showing that there were more positive tweets than negative on average. The ADF and ACF plots in Appendix III both show that the average sentiment data is stationary.

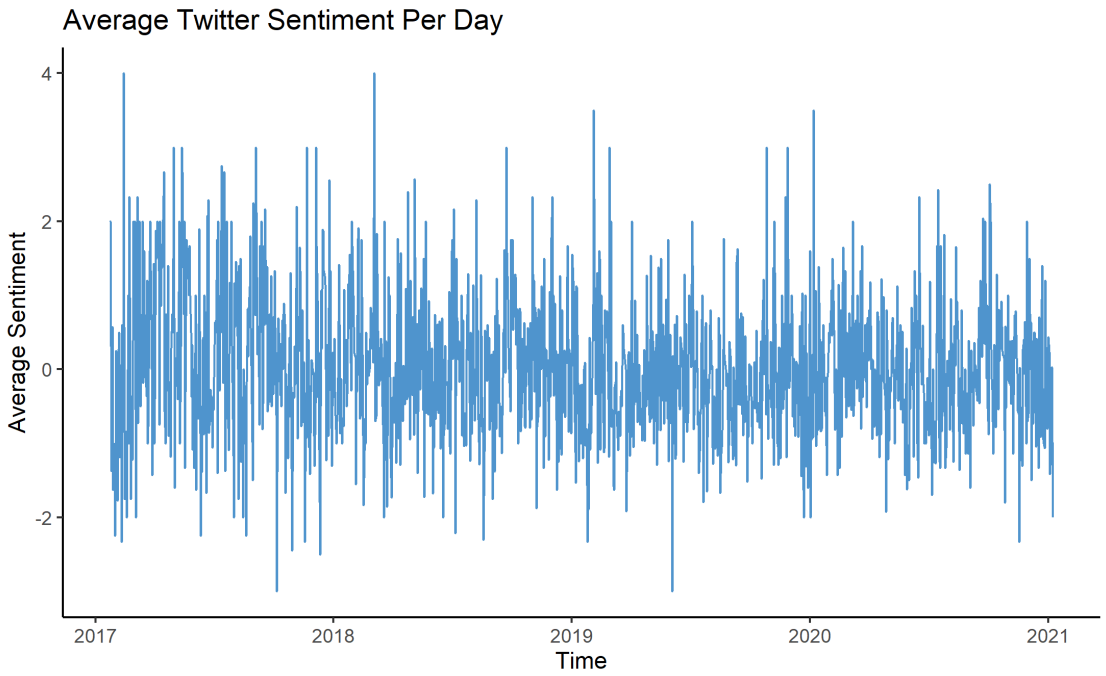


Figure 15 Trump's Average Twitter Sentiment per Day

Figure 16 below shows five-day moving averages of Trump's approval ratings and average daily sentiment. The sentiment and approval ratings seem to move together in a small number of cases after ~ 2018-06-01. For example, the sharp decline after 2019 shows sentiment and approval ratings moving in unison.

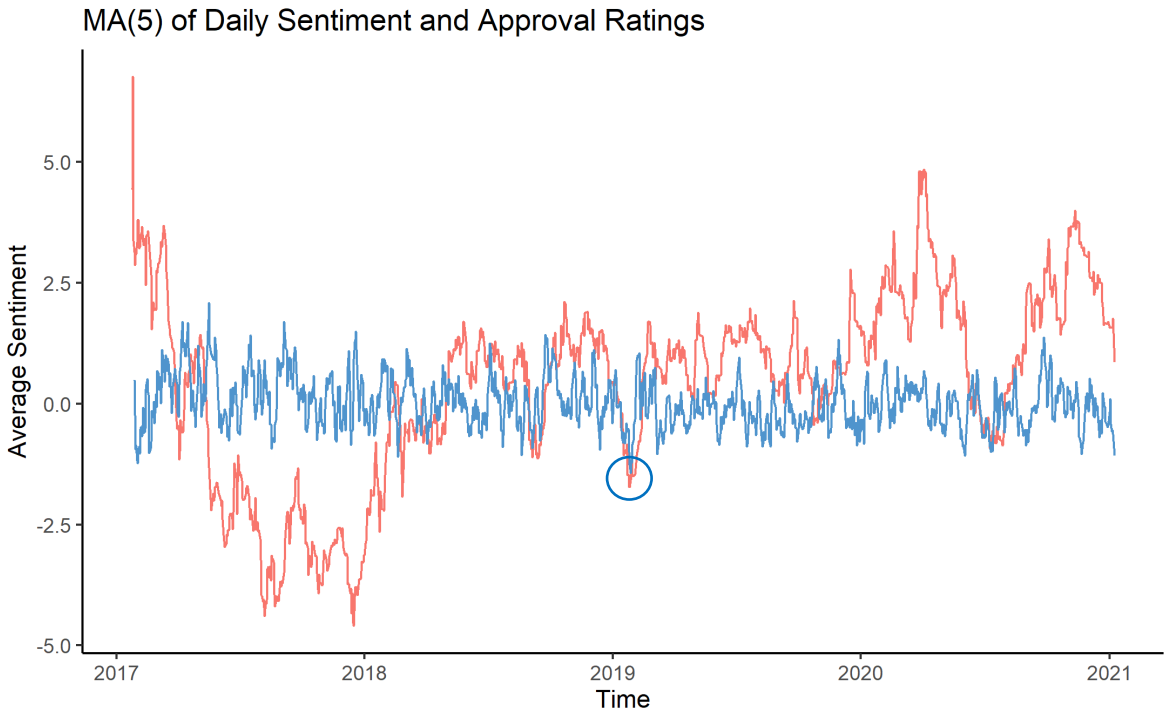


Figure 16 Trumps Approval Ratings and Average Twitter Sentiment Plotted Together

4.3 Regression Analysis of All Presidential Tweets

The regression model was built using the understanding of regression analysis in section 3.4.4. The dependent variable, what is being predicted, is Trump's approval ratings. The independent variable, the predictor, is Trump's daily average sentiment. *Figure 17* shows an example regression built with Trump's Twitter sentiment and approval ratings.

$$approval\ ratings = intercept + \beta(avg.sentiment)$$

Figure 17 Regression Equation

The model summary in Appendix IV and *Table 2* show that Trump's average daily sentiment is not statistically significant, p-value = 0.832. Additionally, the models F-statistic and R-squared are very small, indicating that Trump's average Twitter sentiment is an insignificant predictor of his approval ratings. In fact, his average Twitter sentiment only explains 0.00005761% of the variance in Trump's approval ratings.

Table 2 Regression Output Using all of Trump's Tweets

	P-Value	F-Statistic	R-squared	Standard Error
Regression All Tweets	0.776	0.081	5.761e-05	0.2675

These results are congruent with Sahu, Bai, and Choi's (2020) findings, "there is no direct correlation between the President's Twitter activity and his approval rating" (K. Sahu, Y. Bai, and Y. Choi, 2020).

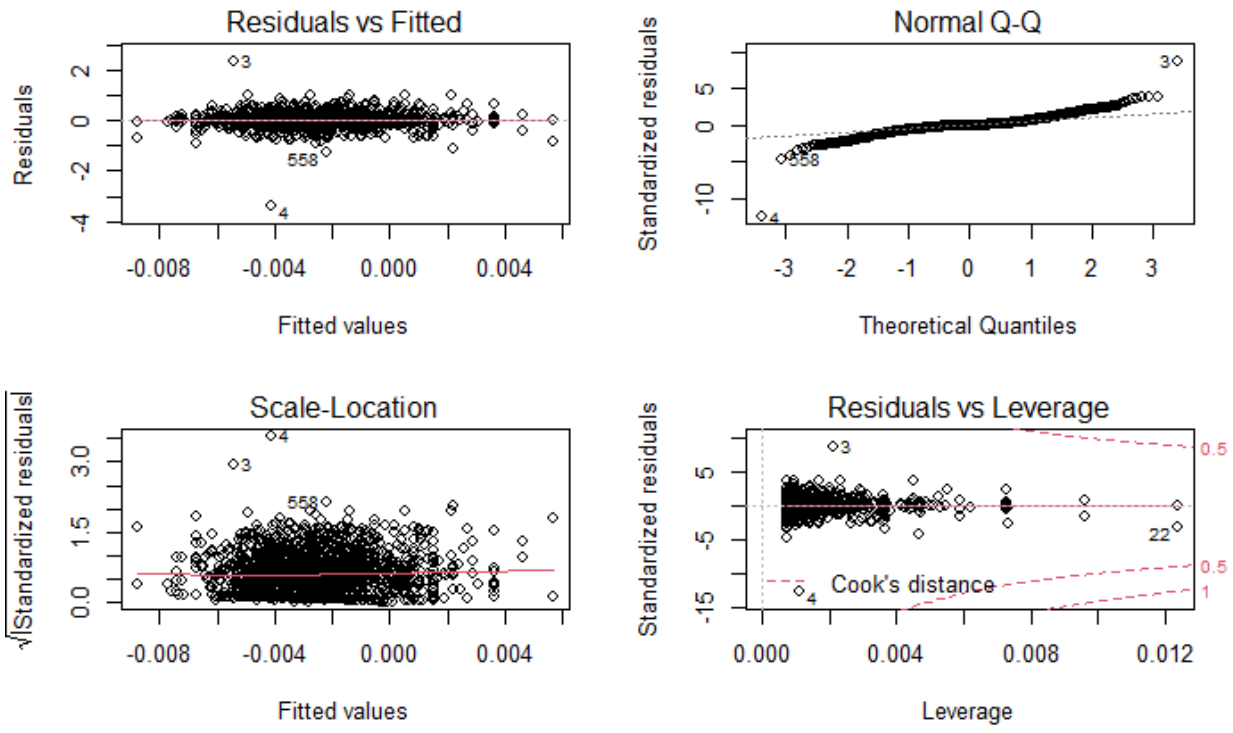


Figure 18 Residuals of Regression Using All of Trump's Tweets

The residual plots in *Figure 18* display normality and homoscedasticity. Although, Cook's distance shows clear outliers.

Residuals vs. Fitted confirms the linear relationship assumption. All residuals are very close to the mean. Additionally, the mean line is almost perfectly horizontal and straight.

Scale-Location confirms that the residuals are homogeneous. Almost all of the residuals are spread equally along the range of the predictors. Although, observations 3 and 4 look like outliers.

The Q-Q Plot is fairly normal, although observations 3 and 4 seem very influential and are pulling the points away from the reference line.

Residuals vs. Leverage indicates that there are outliers. Furthermore, these outliers have a cooks distance > 5, which strongly influences the model.

Cooks distance identifies influential outliers in the predictor variables. These points hurt the model. Generally, points with a Cook's Distance greater than three times the mean should be removed (*Cook's Distance*, 2021).

Figure 19 below shows all observations and their associated Cooks Distance. The blue horizontal line is the threshold for values that are three times greater than the mean. Additionally, values that are labeled are ten times greater than the mean, showing extreme influence.

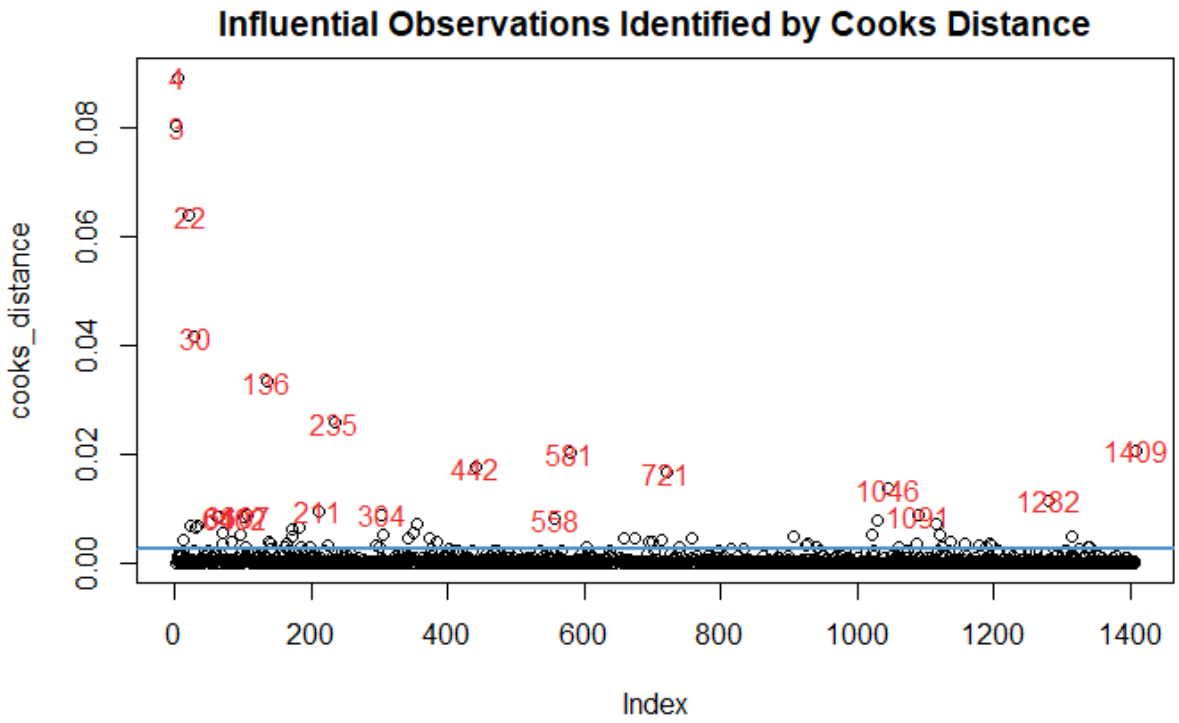


Figure 19 Outliers Identified by Cook's Distance

Figure 20 below shows Trump's approval rating data with and without outliers found with Cook's Distance. Outliers were values that were three times greater than the mean. The plots below show a clear difference before and after the outliers were removed. Two points in particular that had a significant influence on the model were points 3 and 4. The same outliers found in *Figure 19* can be seen at the beginning of the first plot in *Figure 20.*

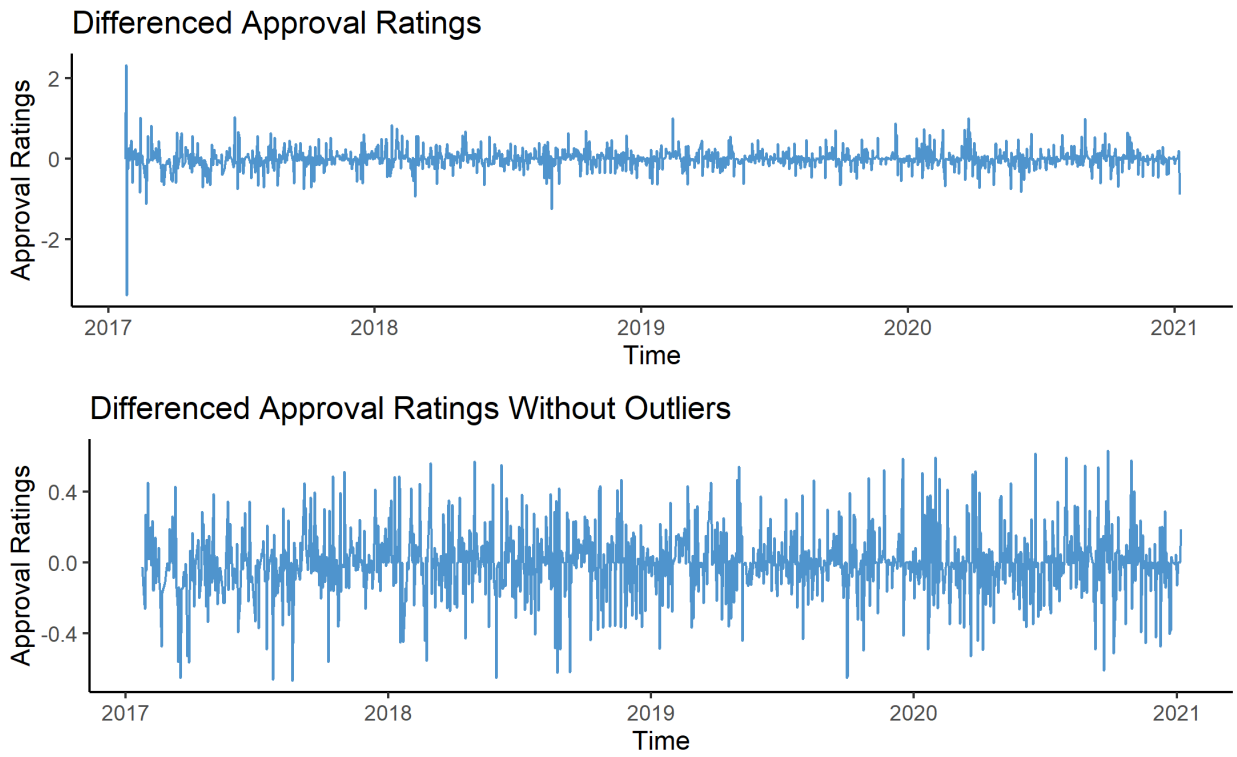


Figure 20 Trumps First Order Differenced Approval Ratings with and Without Outliers

The model was recreated without observations that were considered outliers. Despite removing these outliers, the model fit did not improve. From *Table 3* below, it can be seen that the model without outliers was worse than the initial model. Additionally, The Normal Q-Q plot seen in Appendix V did not improve. This perhaps indicates that the values removed were pertinent to the analysis. With this in mind, the remaining analysis used Trump's differenced approval rating data with the potential outliers.

Table 3 Regression Using All Tweets Without Outliers

	P-Value	F-Statistic	R-squared	Standard Error
Regression All Tweets (No Outliers)	0.8322	0.04489	3.388e-05	0.1894

4.4 Topic Segmentation

As discussed in the Methodology, Trump's major topics were found by first identifying candidate topics using past research regarding Trump's prominent topics and his most frequently used words. Then, these candidate topics were confirmed and validated using keyword extraction.

First, important topics from Trump's presidency were found using past press releases and research. Significant topics from Trump's presidency include "China," "jobs," and "immigration" (Ecker, Jetter

and Lewandowsky, 2020). Donald J. Trump was constantly making remarks about his work in job creation (Chiu, 2020). Additionally, China was the focal point of Trump's international policy discussions (Boucher, J.C. and Thies, C.G., 2019). Finally, bringing an end to illegal immigration was one of the main discussions during Trump's presidential campaign. This topic also followed the President into office (Jim VandeHei, 2017).

Second, word frequencies were used to identify key political topics and legitimize potential topics. For example, manipulating tokens using the dplyr package and visualizing with ggplot2 reveals Trump's top 20 most frequently used words.

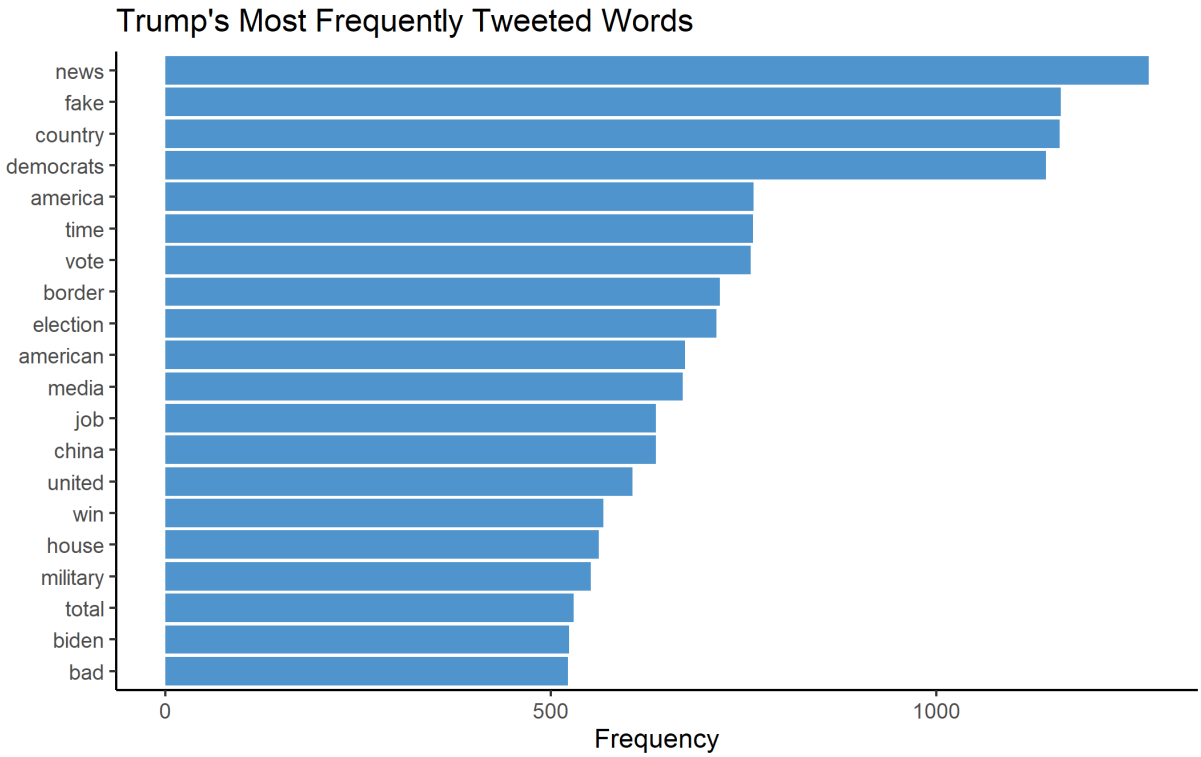


Figure 21 Trump's Most Frequently Used Words

Words with high frequencies included "border," "job," and "china." This finding added legitimacy to the topics discussed above. Other themes such as fake news, the democratic party, elections, and the military were also present. These new themes were considered for potential topics.

Lastly, keyword extraction with the UDpipe package was used to identify keywords. As discussed in the Methodology, Udpipe's keyword phrases function annotates the text and then searches co-occurrences. In this case, co-occurrences of simple noun phrases were extracted because noun phrases comprise vital information within text documents (Kaur, J. and Gupta, V., 2010).

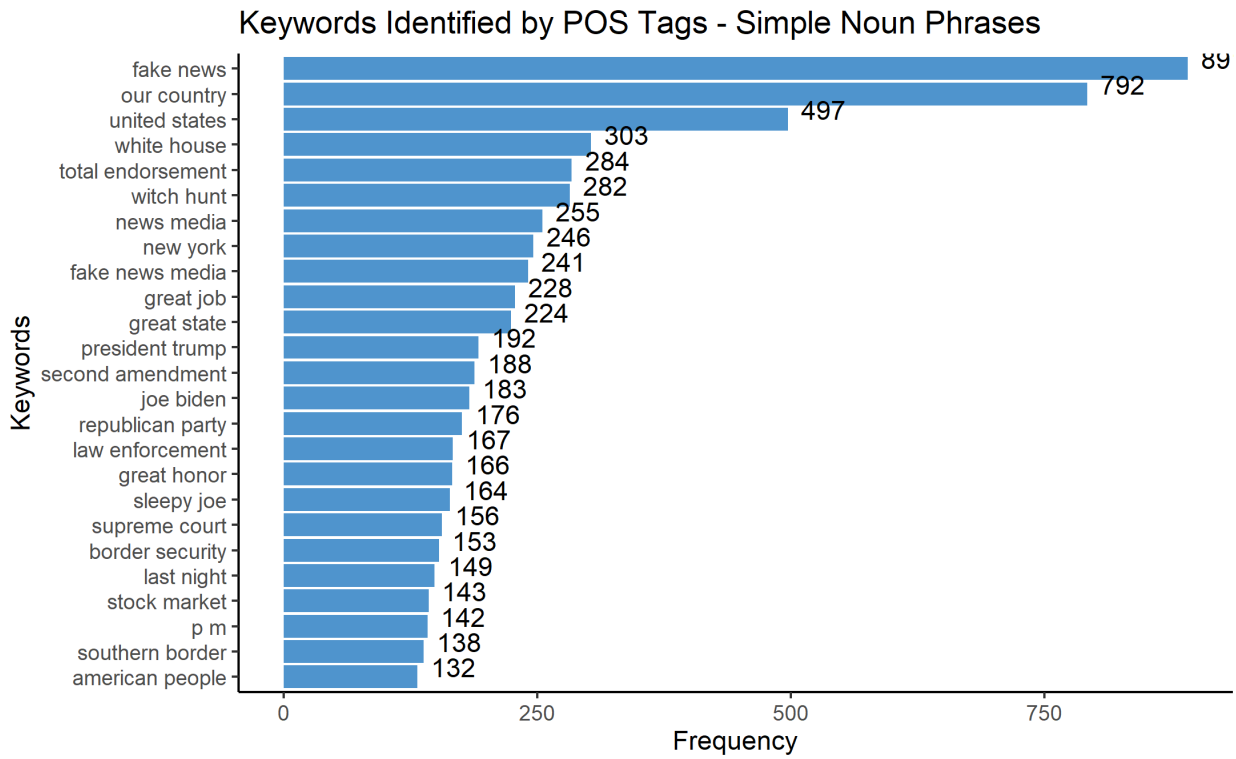


Figure 22 Keywords Identified Using Parts of Speech Tagging - Simple Noun Phrases

From *Figure 22*, high-frequency simple noun phrases are fake news, democratic party, border security, and discussing how great America is.

Based on the analysis above, the three topics which were used to subset Trump's tweets are:

1. Elections and the democratic party
2. Border security
3. News and fake news

Topics were chosen based on past research regarding Trump's major topics, word frequency analysis, and keyword extraction by simple noun phrases. Trump's Tweets were subset using the "grep" function.

4.5 Regression Analysis using Topic Subsets

Based on Colonescu's (2018) recommendation, Trump's tweets were segmented based on prominent topics during his presidency.

Trump's tweets were subset using the topics described in section 4.4. These subsets were then used as predictor variables in three different regression models. Trump's tweets were subset using

different keywords associated with the topics above. The code in *Figure 23* is an example of how tweets about “news” or “fake news” were segmented.

```
news_tweets <- clean_tweets[grep("fake.*news fake.*media|cnn|fake|news", clean_tweets$text, value=FALSE), ]
```

Figure 23 Segmenting Trump’s Tweets Using Keywords

Figure 24 shows five-day moving averages of the average daily sentiment for the three different topic segments. The three charts seem to display stationarity, but the border tweets could potentially be showing a trend. Although, the sample size is far smaller than the democrat and news tweets. Consequently, many days do not have sentiment scores, so ggplot2 filled the space by connecting all points.

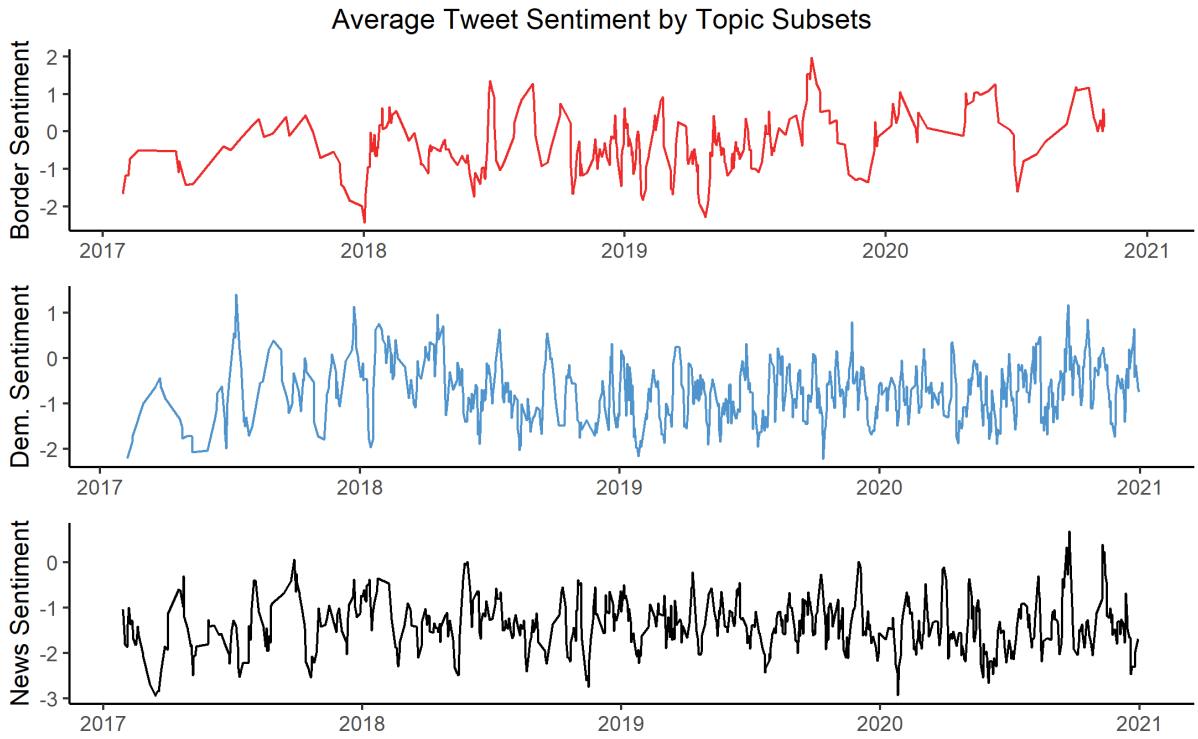


Figure 24 MA(5) Average Sentiment by Twitter Topic Subset

Table 4 shows that the news tweets have the most negative sentiment. Border tweets contain the most positive tweets, although the average is still negative.

Table 4 Average Sentiment of Tweet Subsets

Tweet Subset	**Average Sentiment**
Border Tweets	-0.366215

Left Tweets	-0.73916
News Tweets	-1.385137

Table 5 shows the regression model results for each tweet subset. The different tweet subsets about the democratic party, border security, and news and fake news are all considered insignificant predictors of Trump's approval ratings (p-value > 0.05). Additionally, all models display poor F-statistics and R-squared values. Also, all models display a similar standard error. In this case, the average distance from the regression line to each data point is ~0.27 approval rating points.

Table 5 Regression Model Summary Values by Tweet Subset

Model	P-value	F-statistic	R-squared	Standard Error
News	0.6086	0.2625	0.0003701	0.282
Left	0.9807	0.0005865	0.0000007424	0.2334
Border	0.9127	0.01203	0.0000355	0.2931

The residual plots and model output for the regression models built using subset tweets can be seen in Appendix VI, Appendix VII, and Appendix VIII. All three models have very similar residuals plots.

The residuals for said plots are interpreted below:

Residuals vs. Fitted: All three models display plots with a constant mean and points equally distributed around that mean. This confirms the linear relationship assumption

Scale-Location: All three models display plots with constant means and are all reasonably homogenous. There is nothing too concerning in these plots.

The Q-Q Plot: All plots but the one associated with tweets subset by news keywords are normal. The residuals of the news tweets model are not very normal.

Residuals vs. Leverage: All plots indicate that there are outliers. The same points which were flagged in the regression model containing all of Trump's tweets are present. These points will not be removed because it cannot be confirmed that they are officially outliers.

5. Conclusion and Discussion

5.1 Conclusions

This study measured the influence of Trump's 16,433 presidential tweets on his approval ratings using sentiment analysis, keyword extraction, and regression analysis. The resulting analysis concluded that Trump's average Twitter sentiment was not a strong predictor of his approval ratings. In addition, all subsets of Trump's tweets were deemed insignificant predictors using regression analysis.

Trump's Twitter sentiment was calculated using a dictionary sentiment analysis approach. Trump's tweets were deconstructed using the tidytext method and then averaged over his presidential term following Colonescu's (2018) average sentiment method. His average sentiment was then utilized as a predictor variable in a linear regression model. Following the ideology outlined in Colonescu's (2018) work, Trump's presidential approval ratings were used as the dependent variable in the regression model. Its p-value determined the influence of Trump's Twitter sentiment. The Analysis revealed that Trump's Twitter sentiment was statistically insignificant in predicting Trump's approval ratings. The first regression model's output revealed that Trump's Twitter sentiment was statistically insignificant, P-value = 0.832. These results are congruent with Sahu, Bai, and Choi (2020) findings.

Following Colonescu's (2018) recommendation, additional regression models were constructed using subsets of Trump's presidential tweets. Subsets of Trump's tweets were constructed considering three things: 1) past research examining Trump's prominent topics, 2) tweet word frequencies and 3) automatic keyword extraction. This segmentation methodology indicated that topics that dominated Trump's presidency were:

1. Border security
2. The democratic party
3. News and fake news

Three additional regression models were created using subsets of Trump's tweets based on the above topics. The methodology and construction of the models were congruent with the initial model. The Analysis revealed that the three subsets of Trump's tweets were also poor predictors of Trump's approval ratings. All three subsets were classified as statistically insignificant, P-value > 0.05. These findings indicate that Trump's average Twitter sentiment was a poor predictor of his approval ratings.

5.2 Limitations and Future Work

Regarding topic segmentation, filtering by words assumed to be associated with a specific topic was one shortcoming of this research. Deciding which words are associated with specific topics manually was time-consuming and created room for human error and bias. For future research, a more robust technique such as topic analysis is recommended. Various research has been conducted on hybrid topic-based approaches for sentiment analysis (Ficamos, P. and Liu, Y., 2016). A technique like topic modeling would increase the legitimacy of the analysis by removing human error and bias.

Second, future work could include regression models with additional predictor variables. Examples of additional predictor variables could be the Dow Jones Industrial Average, unemployment rate, and even the COVID-19 death rate.

Lastly, it may be of value to evaluate the relationship between Trump's approval ratings and his Twitter sentiment plus other variables during specific periods. This makes sense not only because certain events did not encompass Trump's entire presidency but also because word frequencies can have high temporal dependencies. For instance, using border security tweets as a predictor variable may not have been wise considering the uneven distribution of border tweets throughout Trump's term.

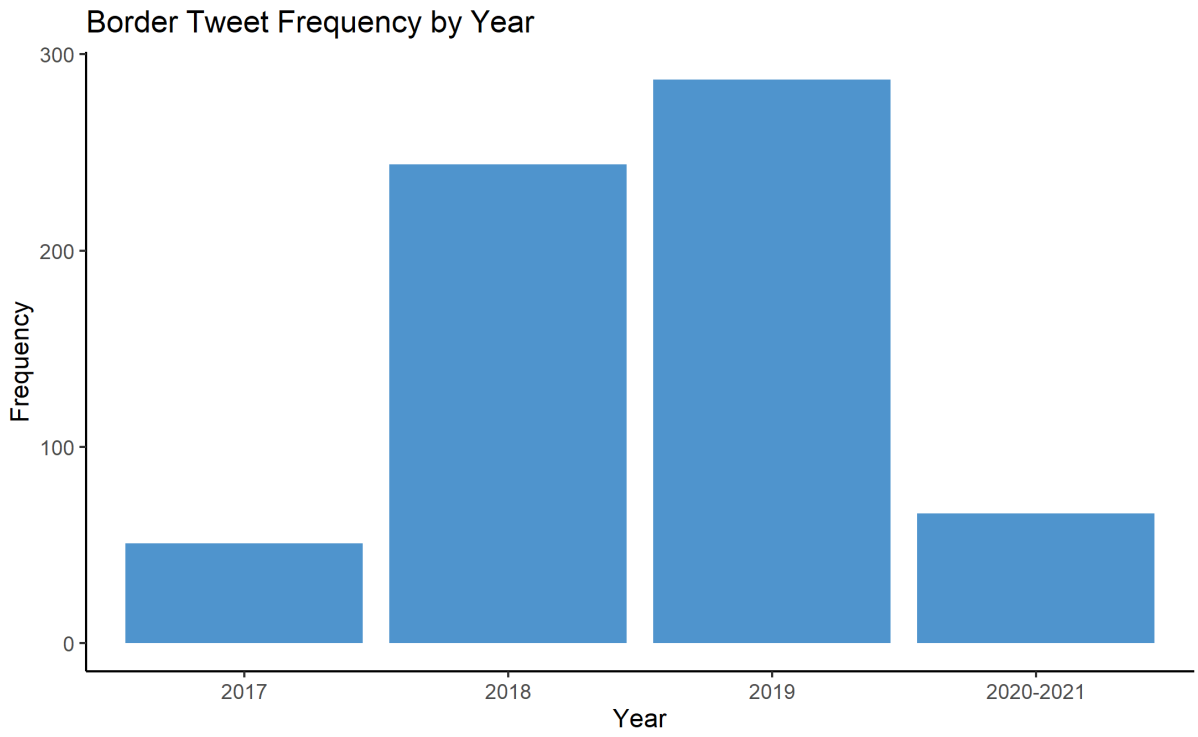


Figure 25 Border Tweet Frequency by Year

6. References

(Rice, D.R. and Zorn, C., 2021) Corpus based dictionaries for sentiment analysis of specialized vocabularies

Agarwal, K., 2020. RAKE: Rapid Automatic Keyword Extraction Algorithm. [online] Medium. Available at: <https://medium.datadriveninvestor.com/rake-rapid-automatic-keyword-extraction-algorithm-f4ec17b2886c> [Accessed 30 August 2021].

Akkaya, C. and Mihalcea, R., 2009. Subjectivity Word Sense Disambiguation. [online] Aclanthology.org. Available at: <https://aclanthology.org/D09-1020.pdf> [Accessed 30 August 2021].

Akkaya, C., Wiebe, J. and Mihalcea, R., 2009. Subjectivity Word Sense Disambiguation. [online] Aclanthology.org. Available at: <https://aclanthology.org/D09-1020.pdf> [Accessed 30 August 2021].

Bordoloi, M. and Biswas, S.K., 2018. Keyword extraction from micro-blogs using collective weight. Social Network Analysis and Mining, 8(1), pp.1-16.

Boucher, J.C. and Thies, C.G., 2019. “I am a tariff man”: the power of populist foreign policy rhetoric under President Trump. The Journal of Politics, 81(2), pp.712-722.

Brown, B., 2021. Trump Twitter Archive. [online] Thetrumparchive.com. Available at: <https://www.thetrumparchive.com/> [Accessed 16 September 2021].

Chakraborty, G., Pagolu, M. and Garla, S., 2021. [online] Support.sas.com. Available at: <http://support.sas.com/publishing/pubcat/chaps/65646.pdf> [Accessed Aug 30 2021]

Chiu, A., 2020. Obama pushes Trump’s button on economy. Trump responds: ‘Did you hear the latest con job?’. [online] The Washington Post. Available at: <https://www.washingtonpost.com/nation/2020/02/18/trump-obama-economy/> [Accessed 16 September 2021].

Colonescu, C., 2018. The effects of Donald Trump’s Tweets on U.S. financial and foreign exchange markets. Athens Journal of Business & Economics, 4(4), pp.375-388.

Ecker, U., Jetter, M. and Lewandowsky, S., 2020. How Trump uses Twitter to distract the media – new research. [online] The Conversation. Available at: <https://theconversation.com/how-trump-uses-twitter-to-distract-the-media-new-research-149847> [Accessed 16 September 2021].

Fahrmeir, L., Kneib, T., Lang, S. and Marx, B., 2007. Regression. Springer-Verlag Berlin Heidelberg.

Fan, W., Wallace, L., Rich, S., & Zhang, Z. (2006). Tapping the power of text mining. Communications of the ACM, 49(9), 76-82.

Ficamos, P. and Liu, Y., 2016. A topic based approach for sentiment analysis on Twitter data. International Journal of Advanced Computer Science and Applications, 7(12), pp.201-205.

Firoozeh, N., Nazarenko, A., Alizon, F. and Daille, B., 2020. Keyword extraction: Issues and methods. Natural Language Engineering, 26(3), pp.259-291.

FiveThirtyEight. 2021. How Popular Is Donald Trump?. [online] Available at: <https://projects.fivethirtyeight.com/trump-approval-ratings/> [Accessed 16 September 2021].

Francia, P.L., 2018. Free media and Twitter in the 2016 presidential election: The unconventional campaign of Donald Trump. Social Science Computer Review, 36(4), pp.440-455.

Granger, C.W. and Newbold, P., 1974. Spurious regressions in econometrics. Journal of econometrics, 2(2), pp.111-120.

Grgić, D., Karaula, M., Babac, M.B. and Podobnik, V., 2020. Predicting dependency of approval rating change from twitter activity and sentiment analysis. In Agents and Multi-Agent Systems: Technologies and Applications 2020 (pp. 103-112). Springer, Singapore.

Ittoo, A., Nguyen, L. M., & van den Bosch, A. (2016). Text analytics in industry: Challenges, desiderata and trends. Computers in Industry, 78, 96–107. https://doi.org/10.1016/J.COMPIND.2015.12.001

Jim VandeHei, M., 2017. What Trump gets most right and most wrong. [online] Axios. Available at: <https://www.axios.com/what-trump-gets-most-right-and-most-

wrong-1513300535-3d67e185-55b0-43ce-8b83-d6287e1dbfc1.html> [Accessed 16 September 2021].

K. Sahu, Y. Bai and Y. Choi, 2020. "Supervised Sentiment Analysis of Twitter Handle of President Trump with Data Visualization Technique," 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, pp. 0640-0646, doi: 10.1109/CCWC47524.2020.9031237.

Wang, N., Zeng, J., Ye, M. and Chen, M., 2018. Text mining and sustainable clusters from unstructured data in cloud computing. Cluster Computing, 21(1), pp.779-788.

Wallsten, S., 2021. Do Likes of Trump's Tweets Predict His Popularity? | Publications | The Technology Policy Institute. [online] The Technology Policy Institute. Available at: <https://techpolicyinstitute.org/publications/miscellaneous/do-likes-of-trumps-tweets-predict-his-popularity/> [Accessed 26 August 2021].

Wallsten, S., 2018. Do Likes of Trump's Tweets Predict His Popularity? | Publications | The Technology Policy Institute. [online] The Technology Policy Institute. Available at: <https://techpolicyinstitute.org/publications/miscellaneous/do-likes-of-trumps-tweets-predict-his-popularity/> [Accessed 30 August 2021].

Thomas, D. R. (2006) 'A General Inductive Approach for Analyzing Qualitative Evaluation Data', American Journal of Evaluation, 27(2), pp. 237–246. doi: 10.1177/1098214005283748.

Thelwall, M., Buckley, K. and Paltoglou, G., 2011. Sentiment in Twitter events. Journal of the American Society for Information Science and Technology, 62(2), pp.406-418.

Sumathy, K.L. and Chidambaram, M., 2013. Text mining: concepts, applications, tools and issues-an overview. International Journal of Computer Applications, 80(4).

Storm, K., 2019. Introduction to construction statistics using Excel. Industrial Process Plant Construction Estimating and Man-Hour Analysis; Gulf Professional Publishing: Houston, TX, USA, pp.1-21.

Sky News. 2021. Trump's tweets: Infamous, offensive and bizarre posts by @realDonaldTrump. [online] Available at: <https://news.sky.com/story/trumps-tweets-infamous-offensive-and-bizarre-posts-by-atrealdonaldtrump-12182992> [Accessed 30 August 2021].

Silver, N., 2021. How We're Tracking Donald Trump's Approval Ratings. [online] FiveThirtyEight. Available at: <https://fivethirtyeight.com/features/how-were-tracking-donald-trumps-approval-ratings/> [Accessed 16 September 2021].

Silver, N., 2021. How We're Tracking Donald Trump's Approval Ratings. [online] FiveThirtyEight. Available at: <https://fivethirtyeight.com/features/how-were-tracking-donald-trumps-approval-ratings/> [Accessed 7 September 2021].

Silva, E.M., Do Prado, H.A. and Ferneda, E., 2002. Text mining: crossing the chasm between the academy and the industry. WIT Transactions on Information and Communication Technologies, 28.

Sigle, J. and Robinson, D., 2021. Preface | Text Mining with R. [online] Tidytextmining.com. Available at: <https://www.tidytextmining.com/preface.html> [Accessed 30 August 2021].

Rstudio-pubs-static.s3.amazonaws.com. 2021. Basics of Text Mining in R - Bag of Words. [online] Available at: <http://rstudio-pubs-static.s3.amazonaws.com/256588_57b585da6c054349825cba46685d8464.html> [Accessed 30 August 2021].

Rice, D.R. and Zorn, C., 2021. Corpus-based dictionaries for sentiment analysis of specialized vocabularies. Political Science Research and Methods, 9(1), pp.20-35.

Rego, F., 2021. Quick Guide: Interpreting Simple Linear Model Output in R. [online] Feliperego.github.io. Available at: <https://feliperego.github.io/blog/2015/10/23/Interpreting-Model-Output-In-R> [Accessed 16 September 2021].

Raja, A., 2019. How to do Topic Extraction from Customer Reviews in R · Programming with R. [online] Programmingwithr.com. Available at: <https://www.programmingwithr.com/how-to-do-topic-extraction-from-customer-reviews-in-r/> [Accessed 16 September 2021].

Raja, A., 2018. Text Analysis in R made easy with Udpipe. [online] Medium. Available at: <https://towardsdatascience.com/easy-text-analysis-on-abc-news-headlines-b434e6e3b5b8> [Accessed 30 August 2021].

Provost, F. and Fawcett, T., 2021. Data Science for Business. [online] O'Reilly Online Learning. Available at: <https://www.oreilly.com/library/view/data-science-for/9781449374273/> [Accessed 26 August 2021].

Provost, F. and Fawcett, T., 2013. Data Science for Business: What you need to know about data mining and data-analytic thinking. " O'Reilly Media, Inc.".

Pain, P. and Masullo Chen, G. (2019) 'The President Is in: Public Opinion and the Presidential Use of Twitter', Social Media + Society. doi: 10.1177/2056305119855143.

Ouyang, Y. and Waterman, R.W., 2020. Trump Tweets: How Often and on What Topics. In Trump, Twitter, and the American Democracy (pp. 53-87). Palgrave Macmillan, Cham.

Nguyen, T.H., Shirai, K. and Velcin, J., 2015. Sentiment analysis on social media for stock movement prediction. Expert Systems with Applications, 42(24), pp.9603-9611.

Mumtaz, D. and Ahuja, B., 2016. A lexical approach for opinion mining in twitter. International Journal of Education and Management Engineering, 6(4), pp.20-29.

Monkeylearn.com. 2021. [online] Available at: <https://monkeylearn.com/keyword-extraction/> [Accessed 30 August 2021].

Misuraca, M., Forciniti, A., Scepi, G. and Spano, M., 2020. Sentiment Analysis for Education with R: packages, methods and practical applications. arXiv preprint arXiv:2005.12840.

Maegan Vazquez, C., 2020. Donald Trump's presidency by the numbers. [online] CNN. Available at: <https://edition.cnn.com/2020/12/18/politics/trump-presidency-by-the-numbers/index.html> [Accessed 30 August 2021].

Lee, S. and Kim, H.J., 2008, September. News keyword extraction for topic tracking. In 2008 Fourth International Conference on Networked Computing and Advanced Information Management (Vol. 2, pp. 554-559). IEEE

Lahuerta-Otero, E. and Cordero-Gutiérrez, R., 2016. Looking for the perfect tweet. The use of data mining techniques to find influencers on twitter. Computers in Human Behavior, 64, pp.575-583.

Kaushik, A. and Naithani, S., 2016. A comprehensive study of text mining approach. International Journal of Computer Science and Network Security (IJCSNS), 16(2), p.69. [10] Tweeting during the Covid: sentiment analysis of twitter messages by President Trump

Kaur, J. and Gupta, V., 2010. Effective approaches for extraction of keywords. International Journal of Computer Science Issues (IJCSI), 7(6), p.144.

Rose, S., Engel, D., Cramer, N. and Cowley, W., 2010. Automatic keyword extraction from individual documents. Text mining: applications and theory, 1, pp.1-20.

Courses.lumenlearning.com. 2021. The Effects of Public Opinion | American Government. [online] Available at: <https://courses.lumenlearning.com/os-government2e/chapter/the-effects-of-public-opinion/> [Accessed 17 September 2021].

Enli, G., 2017. Twitter as arena for the authentic outsider: exploring the social media campaigns of Trump and Clinton in the 2016 US presidential election. European journal of communication, 32(1), pp.50-61.

Shaban, T.A., Hexter, L. and Choi, J.D., 2017, September. Event Analysis on the 2016 US Presidential Election using social media. In *International conference on social informatics* (pp. 201-217). Springer, Cham.

Statistics How To. 2021. *Cook's Distance / Cook's D: Definition, Interpretation*. [online] Available at: <https://www.statisticshowto.com/cooks-distance/> [Accessed 17 September 2021].

Wirth, R. and Hipp, J., 2000, April. CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (Vol. 1). London, UK: Springer-Verlag.

Wu, X. and Bolivar, A., 2008, April. Keyword extraction for contextual advertisement. In *Proceedings of the 17th international conference on World Wide Web* (pp. 1195-1196).

Yaqub, U., 2020. Tweeting During the Covid-19 Pandemic: Sentiment Analysis of Twitter Messages by President Trump. *Digital Government: Research and Practice*, *2*(1), pp.1-7.

7. RStudio References

Alboukadel Kassambara (2020). ggpubr: 'ggplot2' Based Publication Ready Plots. R package version 0.4.0. https://CRAN.R-project.org/package=ggpubr

Dirk Eddelbuettel and James Joseph Balamuta (2018). Extending R with C++: A Brief Introduction to Rcpp. The American Statistician.

Dirk Eddelbuettel and Romain Francois (2011). Rcpp: Seamless R and C++ Integration. Journal of Statistical Software, 40(8), 1-18. URL https://www.jstatsoft.org/v40/i08/.

Eddelbuettel, Dirk (2013) Seamless R and C++ Integration with Rcpp. Springer, New York. ISBN 978-1-4614-6867-7.

Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL https://www.jstatsoft.org/v40/i03/.

Hadley Wickham (2019). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0. https://CRAN.R-project.org/package=stringr

Hadley Wickham and Jennifer Bryan (2019). readxl: Read Excel Files. R package version 1.3.1. https://CRAN.R-project.org/package=readxl

Hans W. Borchers (2021). pracma: Practical Numerical Math Functions. R package version 2.3.3. https://CRAN.R-project.org/package=pracma

Ian Fellows (2018). wordcloud: Word Clouds. R package version 2.6. https://CRAN.R-project.org/package=wordcloud

Jan Wijffels (2021). udpipe: Tokenization, Parts of Speech Tagging, Lemmatization and Dependency Parsing with the 'UDPipe' 'NLP' Toolkit. R package version 0.8.6. https://CRAN.R-project.org/package=udpipe

Kearney, M. W. (2019). rtweet: Collecting and analyzing Twitter data, Journal of Open Source Software, 4, 42. 1829 doi:10.21105/joss.01829 (R package version 0.7.0)

Matt Dancho and Davis Vaughan (2021). tidyquant: Tidy Quantitative Financial Analysis. R package version 1.0.3. https://CRAN.R-project.org/package=tidyquant

Matt Dowle and Arun Srinivasan (2021). data.table: Extension of `data.frame`. R package version 1.14.0. https://CRAN.R-project.org/package=data.table

Nicolas Proellochs and Stefan Feuerriegel (2021). SentimentAnalysis: Dictionary-Based Sentiment Analysis. R package version 1.3-4.

Rinker, T. W. (2020). qdap: Quantitative Discourse Analysis Package. 2.4.2. Buffalo, New York. https://github.com/trinker/qdap

Silge J, Robinson D (2016). "tidytext: Text Mining and Analysis Using Tidy Data Principles in R." _JOSS_, *1*(3). doi: 10.21105/joss.00037 (URL: https://doi.org/10.21105/joss.00037), <URL: http://dx.doi.org/10.21105/joss.00037>.

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

8. Glossary

Tweet message posted on Twitter (See [1])

Retweet reposting a tweet (See [2])

Twitter Handle Twitter username (See [3])

Reach is the estimated amount of people you can contact (See [4])

Engagement user interactions (See [5])

Influencer someone who has the power to influence people's purchasing decisions (See [6])

Lexicon Dictionary prebuilt dictionary with sentiment scores (See [7])

Tweeted past tense of a message posted on Twitter (See [8])

Scraping extracting data from websites (See [9])

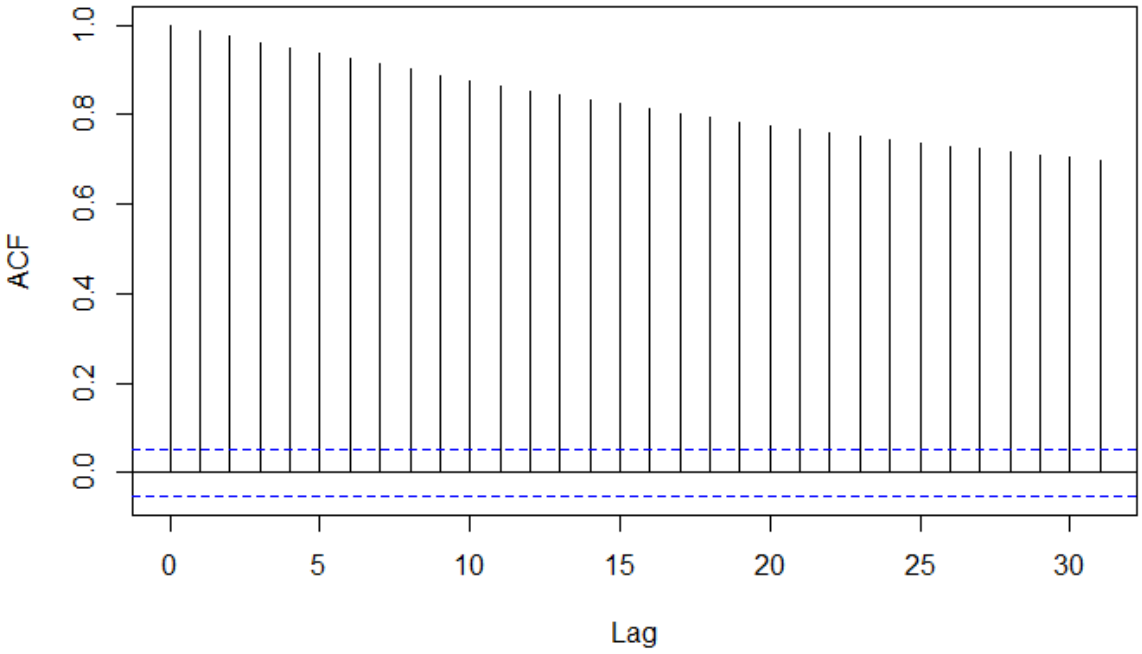
Anit-Join removing common rows from two datasets (See [10])

Inner-Join combining two datasets on common rows (See [11])

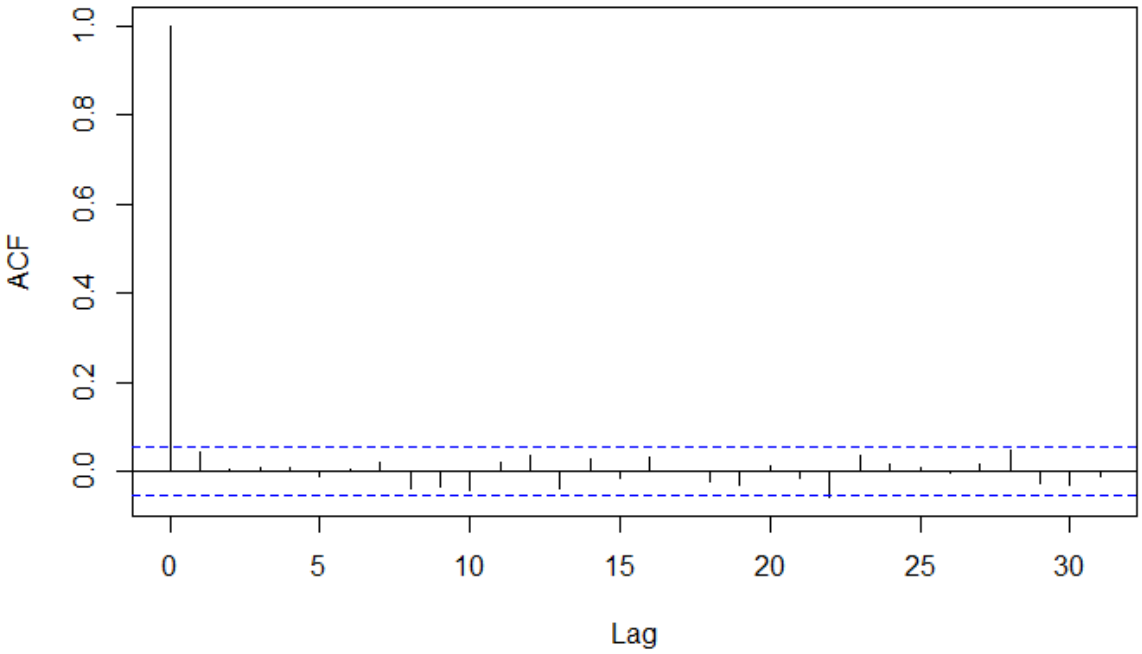
Stationary Data time series data with a 0 mean and constant variance (See [12])

Mention when another user makes a tweet containing another user's twitter handle (See[13])

Appendix I – ACF Plot of Trump's Approval Ratings



Appendix II – ACF Plot of Trump's Approval Ratings



Appendix III – Trump Average Twitter Sentiment Stationarity Tests

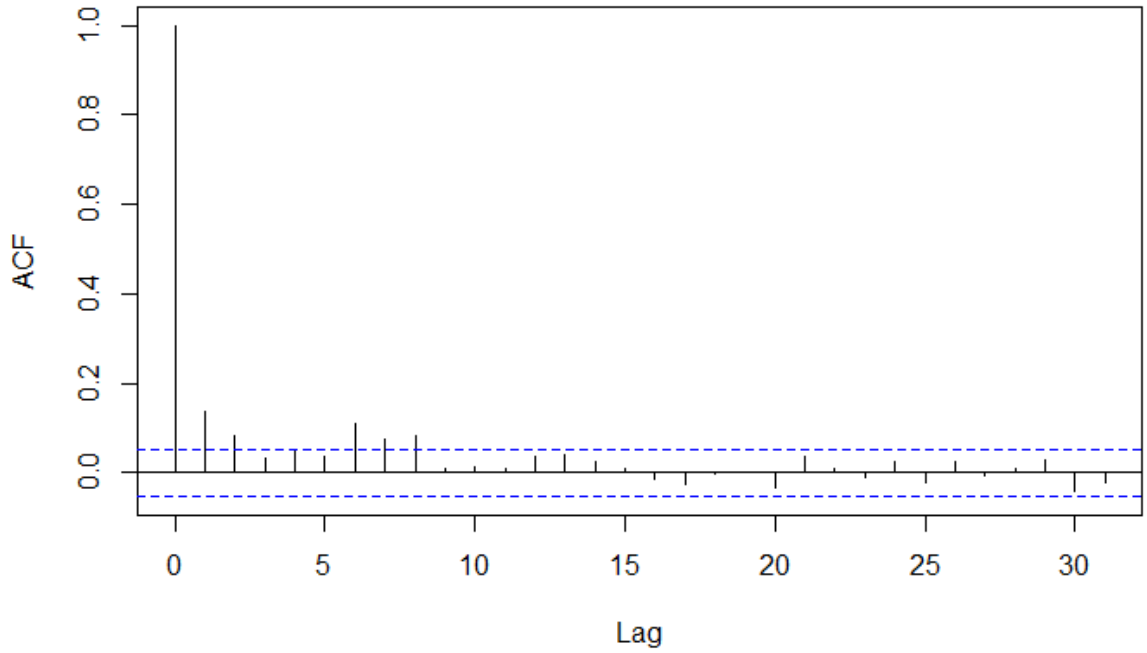


Table 6 Augmented Dickey-Fuller of Trump's Twitter Sentiment

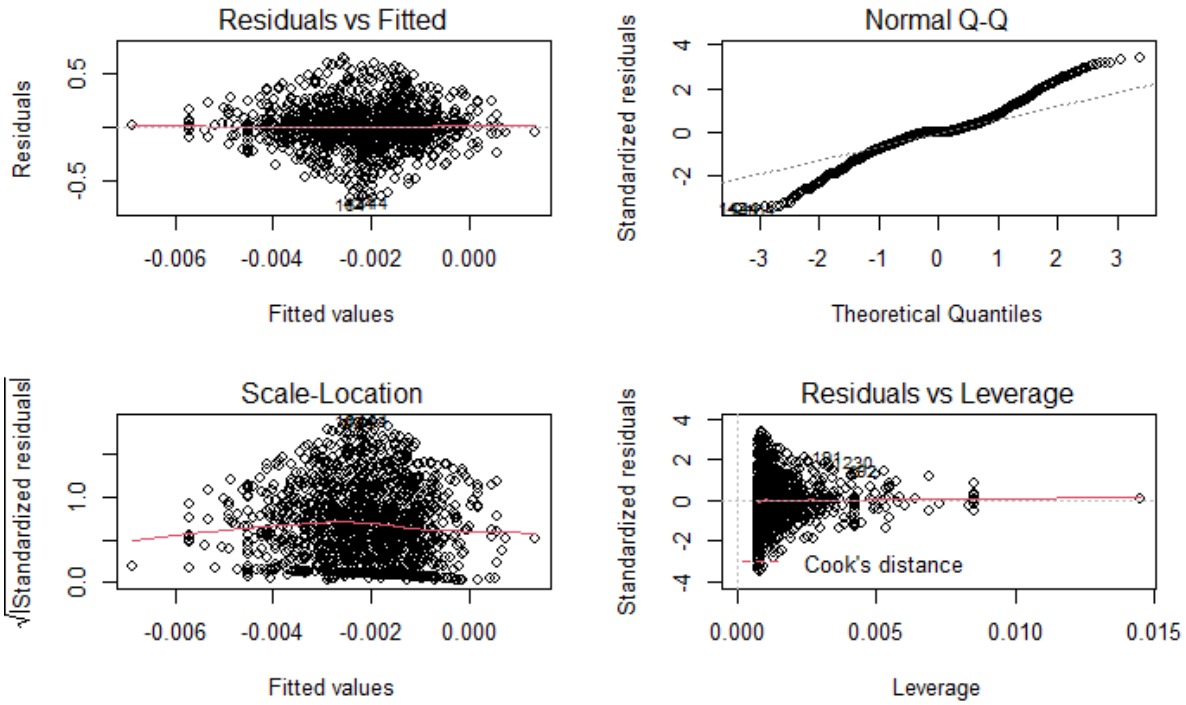
	P-value
Trump Twitter Sentiment	0.01

Appendix IV – Regression Model using All of Trump's Tweets

```
Call:
lm(formula = approval_diff_1 ~ avg_sent, data = app_and_sent_outliers)

Residuals:
     Min       1Q   Median       3Q      Max
-0.66729 -0.08610  0.00152  0.07317  0.63706

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.002180   0.005201  -0.419    0.675
avg_sent    -0.001171   0.005528  -0.212    0.832

Residual standard error: 0.1894 on 1325 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  3.388e-05, Adjusted R-squared:  -0.0007208
F-statistic: 0.04489 on 1 and 1325 DF,  p-value: 0.8322
```

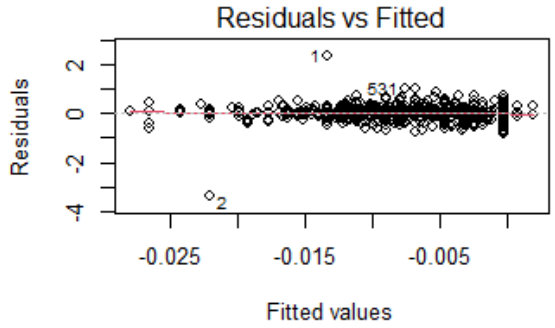
Appendix V – All Tweet Regression Model Without Outliers

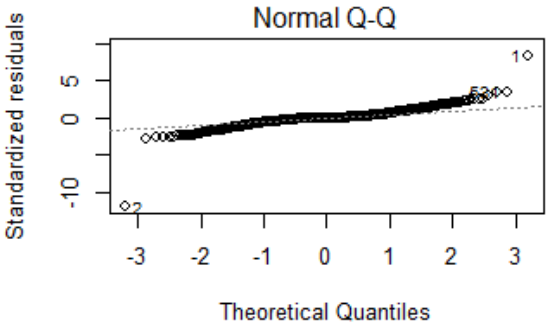
```
Call:
lm(formula = approval_diff_1 ~ avg_sent, data = app_and_sent_outliers)

Residuals:
     Min       1Q   Median       3Q      Max
-0.66729 -0.08610  0.00152  0.07317  0.63706

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.002180   0.005201  -0.419    0.675
avg_sent    -0.001171   0.005528  -0.212    0.832

Residual standard error: 0.1894 on 1325 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  3.388e-05, Adjusted R-squared:  -0.0007208
F-statistic: 0.04489 on 1 and 1325 DF,  p-value: 0.8322
```

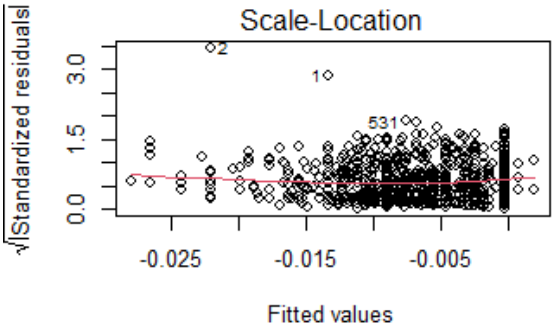


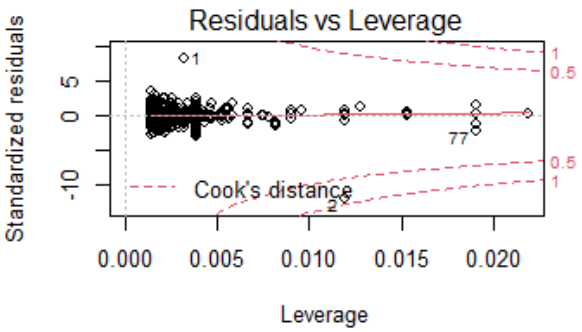
Appendix VI – Regression Built with Tweets Subset by News Keywords

```
lm(formula = approval_diff_1 ~ news_avg_sent, data = app_and_sentDT_news)

Residuals:
    Min      1Q  Median      3Q     Max
-3.3668 -0.0927  0.0059  0.0817  2.3357

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.013388   0.015879  -0.843    0.399
news_avg_sent -0.004381   0.008551  -0.512    0.609

Residual standard error: 0.282 on 709 degrees of freedom
Multiple R-squared:  0.0003701, Adjusted R-squared:  -0.00104
F-statistic: 0.2625 on 1 and 709 DF,  p-value: 0.6086
```



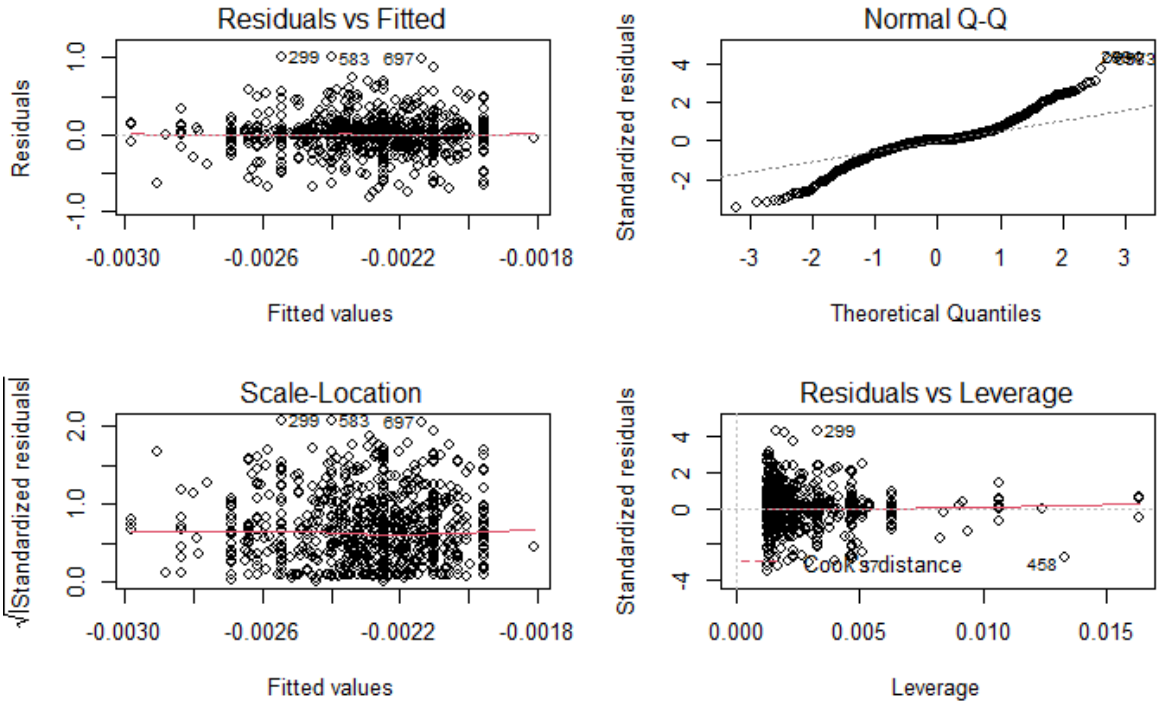






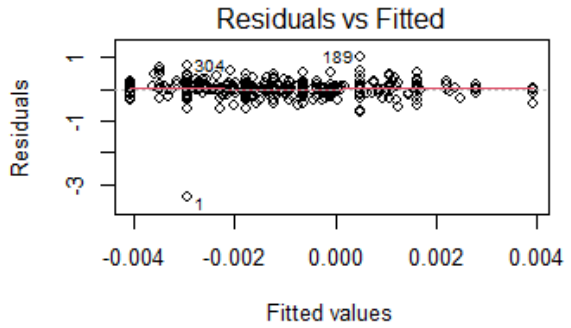
Appendix VII – Regression Built with Tweets Subset by Democratic Keywords

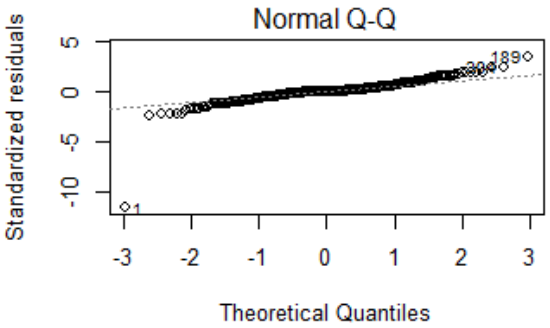
```
lm(formula = approval_diff_1 ~ left_avg_sent, data = app_and_sentDT_left)

Residuals:
     Min       1Q   Median       3Q      Max
-0.82020 -0.09134  0.00221  0.07618  1.00545

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.0023959  0.0094183  -0.254    0.799
left_avg_sent -0.0001463  0.0060399  -0.024    0.981

Residual standard error: 0.2334 on 790 degrees of freedom
Multiple R-squared:  7.424e-07, Adjusted R-squared:  -0.001265
F-statistic: 0.0005865 on 1 and 790 DF,  p-value: 0.9807
```

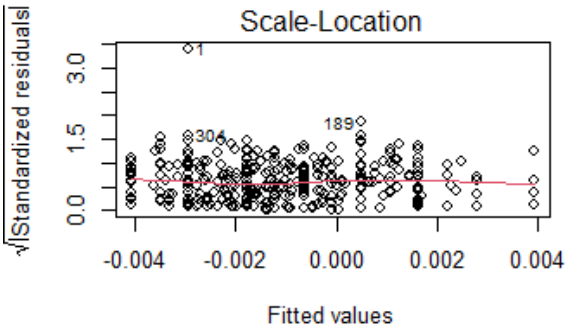


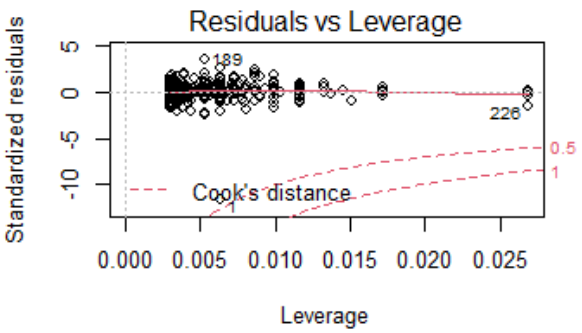
Appendix VIII – Regression Built with Tweets Subset by Border Keywords

```
lm(formula = approval_diff_1 ~ Border_avg_sent, data = app_and_sentDT_Border)

Residuals:
    Min      1Q  Median      3Q     Max
-3.3861 -0.0978  0.0018  0.1064  1.0024

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.000648   0.016322   -0.04    0.968
Border_avg_sent  0.001139   0.010384    0.11    0.913

Residual standard error: 0.2931 on 339 degrees of freedom
Multiple R-squared:  3.55e-05,  Adjusted R-squared:  -0.002914
F-statistic: 0.01203 on 1 and 339 DF,  p-value: 0.9127
```









Appendix IX – R Code

Data Preparation

Reading in Data and Cleaning

```
#read in csv
tweets <- read_csv("all_trump_tweets.csv")

column types or set `show_col_types = FALSE` to quiet this message.

#clean data by remoivng twitter specific text
clean_tweets <- tweets

clean_tweets$text <- tolower(clean_tweets$text)
# Remove mentions, urls, emojis, numbers, punctuations, etc.
clean_tweets$text <- gsub("@\\w+", "", clean_tweets$text)
clean_tweets$text <- gsub("https?://.+", "", clean_tweets$text)
clean_tweets$text <- gsub("\\d+\\w*\\d*", "", clean_tweets$text)
clean_tweets$text <- gsub("#\\w+", "", clean_tweets$text)
clean_tweets$text <- gsub("[^\x01-\x7F]", "", clean_tweets$text)
clean_tweets$text <- gsub("[[:punct:]]", " ", clean_tweets$text)
clean_tweets$text <- gsub("amp", " ", clean_tweets$text)
# Remove spaces and newlines
clean_tweets$text <- gsub("\n", " ", clean_tweets$text)
clean_tweets$text <- gsub("^\\s+", "", clean_tweets$text)
clean_tweets$text <- gsub("\\s+$", "", clean_tweets$text)
clean_tweets$text <- gsub("[ |\t]+", " ", clean_tweets$text)

#Removing retweets
clean_tweets <- clean_tweets[!grepl("^rt", clean_tweets$text),]

#Checking for NA and structure errors

summary(clean_tweets)
str(clean_tweets)
```

Filtering for Tweets During Trump's Term

```
clean_tweets$date <- as.Date(clean_tweets$date)

clean_tweets <- clean_tweets %>% filter(date >= as.Date("2017-01-23") & date <= as.Date("2021-01-20"))
```

Tokenizing Tweets and Removing Stopwords

```
#Unesting Tokens
tidy_tweets <- clean_tweets %>% mutate(linenumber = row_number()) %>%  unnest_tokens(word, text)

#Creating custom stopwords
custom_stop_words <- tribble(
  ~word,     ~lexicon,
  "íí","CUSTOM",
  "get","CUSTOM",
  "like","CUSTOM",
```

```
  "just","CUSTOM",
  "yes","CUSTOM",
  "know","CUSTOM",
  "will","CUSTOM",
  "good","CUSTOM",
  "day","CUSTOM",
  "people","CUSTOM",
  "amp", "CUSTOM",
  "dont", "CUSTOM",
  "trump", "CUSTOM",
  "president", "CUSTOM"

)

#Adding custom stopwords
stop_words2 <- stop_words %>%
bind_rows(custom_stop_words)

#Anit-joining tokenized tweets with stopwords
tidy_tweets <- tidy_tweets %>% anti_join(stop_words2)

## Joining, by = "word"
```

Data Discovery

Visualizing Trump's Twitter Usage

```
tweet_plot <- clean_tweets %>% select(date) %>% group_by(date) %>% summarize(freq = n()) %>%
arrange(date)

ggplot(tweet_plot, aes(x = date, y = freq)) +
 geom_line(color = "steelblue3")+
 geom_smooth(color = "firebrick1", se = FALSE)+
 labs(title = "Trump Tweet Frequency During Term", x = "Time", y = "Number of Tweets") +
 theme_classic() +
 theme(legend.position = "none")

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

#ggsave("Trump Tweet Frequency During Term.png")
```

Plotting Trump's Most Frequently Used Words

```
#Arranging trumps most frequently used words in descending order
word_count <- tidy_tweets %>% count(word) %>% mutate(id = row_number()) %>% arrange(desc(n)
) %>% mutate(word2 = fct_reorder(word, n)) %>% anti_join(stop_words2)

## Joining, by = "word"

#Sorting for top 20 most frequent words
top_20 <- word_count[1:20,]

ggplot(top_20, aes(x = word2, y = n)) +
 geom_bar(fill = "steelblue3", stat = "identity") +
 coord_flip() +
 theme_classic()+
```

```
 ggtitle("Trump's Most Frequently Tweeted Words")+
 guides(fill = FALSE) +
 xlab("") +
 ylab("Frequency")+
 theme(legend.position = "none")
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

```
#ggsave("Trump's Most Frequently Tweeted Words.png")
```

Plotting Trumps Word Frequency by Year

```
#Arranging words by year
word_count_year <- tidy_tweets %>% mutate(year = year(date)) %>% select(year, word)%>%  group
_by(word, year) %>%  summarize(freq = n()) %>%  arrange(desc(freq))
```

```
## `summarise()` has grouped output by 'word'. You can override using the `.groups` argument.
```

```
#Assigning words by year to their own variable
first_word_count <- word_count_year %>% filter(year == 2017) %>% arrange(desc(freq)) %>% head(
10)
second_word_count <- word_count_year %>% filter(year == 2018) %>% arrange(desc(freq)) %>% he
ad(10)
third_word_count <- word_count_year %>% filter(year == 2019) %>% arrange(desc(freq)) %>% head
(10)
fourth_word_count <- word_count_year %>% filter(year == 2020) %>% arrange(desc(freq)) %>% hea
d(10)
fifth_word_count <- word_count_year %>% filter(year == 2021) %>% arrange(desc(freq)) %>% head(
10)
```

```
#Plotting each years most frequently used words seperately
a <- ggplot(first_word_count, aes(x = reorder(word, freq), y = freq, fill = word)) +
 geom_bar(stat = "identity", fill = "steelblue3") +
 coord_flip() +
 theme_classic()+
 guides(fill = FALSE) +
 ylab("") +
 xlab("")+
 theme(axis.text.x = element_blank())
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

```
b <- ggplot(second_word_count, aes(x = reorder(word, freq), y = freq, fill = word)) +
 geom_bar(stat = "identity", fill = "steelblue3") +
 coord_flip() +
 theme_classic()+
 guides(fill = FALSE) +
 ylab("") +
 xlab("")+
 theme(axis.text.x = element_blank())
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

c <- ggplot(third_word_count, aes(x = reorder(word, freq), y = freq, fill = word)) +
 geom_bar(stat = "identity", fill = "steelblue3") +
 coord_flip() +
 theme_classic()+
 guides(fill = FALSE) +
 ylab("") +
 xlab("")+
 theme(axis.text.x = element_blank())

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

d <- ggplot(fourth_word_count, aes(x = reorder(word, freq), y = freq, fill = word)) +
 geom_bar(stat = "identity", fill = "steelblue3") +
 coord_flip() +
 theme_classic()+
 guides(fill = FALSE) +
 ylab("") +
 xlab("")+
 theme(axis.text.x = element_blank())

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

e <- ggplot(fifth_word_count, aes(x = reorder(word, freq), y = freq, fill = word)) +
 geom_bar(stat = "identity", fill = "steelblue3") +
 coord_flip() +
 theme_classic()+
 guides(fill = FALSE) +
 ylab("")+
 xlab("")+
 theme(axis.text.x = element_blank())

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.

#Combining all plots into one
figure <- ggarrange(a, b, c, d, e +
                      rremove("x.text"),
                    title = "Most Frequent Words Per Year",
                    labels = c("2017", "2018", "2019", "2020", "2021"),
                    ncol = 3, nrow = 2)

## Warning in as_grob.default(plot): Cannot convert object of class character into
## a grob.

annotate_figure(figure,
                top = text_grob("Tweet Word Frequency by Year", face = "bold", size = 13))

#ggsave("Tweet Word Frequency by Year.png")
```

Wordcloud of Trump's Most Frequently Used Words

```
set.seed(111)
pal = brewer.pal(9,"Blues")
word_count %>% with(wordcloud(word2, n, random.order = FALSE, random.color = FALSE, max.words = 25, colors = pal, scale = c(4, .5)))
```

```
## Warning in wordcloud(word2, n, random.order = FALSE, random.color = FALSE, :
## republican could not be fit on page. It will not be plotted.
```

Preparing Data To Visualize Using the 'Bing' Dictionary

```
#Creating Custom Stopwords
custom_bing_stop_words <- tribble(
  ~word,     ~lexicon,
  "íí","CUSTOM",
  "get","CUSTOM",
  "like","CUSTOM",
  "just","CUSTOM",
  "yes","CUSTOM",
  "know","CUSTOM",
  "will","CUSTOM",
  "good","CUSTOM",
  "day","CUSTOM",
  "people","CUSTOM",
  "amp", "CUSTOM",
  "dont", "CUSTOM",
  "trump", "CUSTOM" #Added "trump" because the bing library treats it as a positive word

)

#Adding custom stop words to stopwords
stop_words_bing <- stop_words %>%
bind_rows(custom_bing_stop_words)


#Joining Trump's tokens with 'Bing' dictionary
trump_tweet_bing <- tidy_tweets %>%
 anti_join(custom_bing_stop_words) %>%
 inner_join(get_sentiments("bing")) %>%
 count(word, sentiment, sort = TRUE) %>%
 ungroup()
```

```
## Joining, by = "word"
## Joining, by = "word"
```

Plotting Trump's most Frequently Used Positive and Negative Words Using 'Bing' Tokens

```
#Creating colors for chart
my_colors <- c("firebrick2", "steelblue3")

#Plotting trumps most frequently used positive and negative words
trump_tweet_bing %>%
 group_by(sentiment) %>%
 slice_max(n, n = 15) %>%
```

```
 ungroup() %>%
 mutate(word = reorder(word, n)) %>%
 ggplot(aes(n, word, fill = sentiment)) +
 geom_col(show.legend = FALSE) +
 facet_wrap(~sentiment, scales = "free_y") +
 labs(x = "Contribution to sentiment", y = NULL, title = "Trump's Most Frequently Used Positive and
Negative Words") +
 theme_classic()+
 scale_fill_manual(values= my_colors)

#ggsave("Trump's Most Frequently Used Positive and Negative Words.png")
```

Creating Wordcloud with 'Bing' Tokens

```
#Creating word cloud
set.seed(123)

tidy_tweets %>%
 anti_join(stop_words_bing) %>%
 inner_join(get_sentiments("bing")) %>%
 count(word, sentiment, sort = TRUE) %>%
 reshape2::acast(word ~ sentiment, value.var = "n", fill = 0) %>%
 comparison.cloud(colors = c("firebrick2", "steelblue3"),
                  max.words = 100)
```

Sum of Trump's Twitter Sentiment Per Day

```
tidy_tweets %>%
 inner_join(get_sentiments("afinn")) %>%
 group_by(date) %>%
 summarize(avg_sent = sum(value)) %>%
 ggplot(aes(x = date, y = avg_sent))+
 geom_line(color = "steelblue3", linetype = "solid")+
 labs(title = "Sum of Trump's Twitter Sentiment by Day", x = "Time", y = "Average Sentiment")+
 theme_classic()

## Joining, by = "word"

#ggsave("Sum of Trump's Twitter Sentiment by Day.png")
```

Trump's Approval Rating Data

Reading in Trump's Approval Rating Data

```
#Reading in the approval rating data
app_rate_data <- read_excel("approval_topline_five_thirty_eight_USE_THIS_ONE_cross_checked.xls
x")

## New names:
## * `` -> ...11
## * `` -> ...12
## * `` -> ...13
```

Filtering for Voter Type and Extracting Relevant Columns

```
#Filtering for the relevant columns
app_data <- app_rate_data %>% select(subgroup, modeldate, approve_estimate) %>% filter(subgro
up == "All polls") %>% rename(date = modeldate) %>% arrange(date)
```

```
#Plotting Trump's Approval Ratings
ggplot(app_data, aes(x = date, y = approve_estimate)) +
 geom_line(color = "steelblue3") +
 labs(title = "Trump Approval Rating (All polls)", x = "Time", y = "Approval Rating (%)") +
 theme_classic()

#ggsave("Trump Approval Rating.png")
```

Data Prep for Approval Ratings with Major Events

```
#Variable containing dates and associated names of major events
major_events <- data.frame("date" = as.Date(c("2017-06-01", "2017-08-15", "2018-03-01", "2019-10
-26", "2019-12-18", "2020-03-13", "2020-07-06", "2020-11-07", "2021-01-06", "2021-01-13", "2019-
01-25")), "Name" = c("Paris Agreement", "Charlottesville", "China Trade War Begins", "ISIS chief Bag
hdadi Killed", "Impeachment", "COVID national emergency", "WHO withdrawl", "Loses re-election",
"Capitol riot", "Second impeachment", "Border wall battle"))

#Converting event dates to 'date' format
app_data$date <- as.Date(app_data$date)
#Joining approval ratings data with major events data
app_with_points <- full_join(app_data, major_events, by = "date")
#Created new variable to add dates to special event rows
app_with_points <- app_with_points %>% mutate(point_date = ifelse(is.na(Name), NA, as.Date(date
)))
#Converted date to 'date' format
app_with_points$point_date <- as.Date(app_with_points$point_date)
```

Plotting Major Events

```
#Library for adding labels to graphs
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 4.0.5
```

```
#ggplot for major events
ggplot(app_with_points, aes(x = date, y = approve_estimate)) +
 geom_line(color = "steelblue3") +
 geom_point(aes(x = point_date, y = approve_estimate, color = "red"))+
 geom_label_repel(aes(label = Name),
                  box.padding = 0.35,
                  point.padding = .5,
                  max.overlaps = Inf,
                  direction = "both",
                  nudge_y = 2,
                  segment.size = .5,
                  ylim = c(-Inf, Inf),
                  segment.color = "firebrick2") +
 labs(y = "Approval Estimate (%)", x = "Time", title = "Trump's Approval Rating with Major Events")+
 theme_classic()+
 theme(legend.position = "None")

#ggsave("Trump's Approval Rating with Major Events.png")
```

Computing average sentiment time series

```
#Joining twitter tokens with 'afinn' dictionary
tidy_tweets <- tidy_tweets %>%
 inner_join(get_sentiments("afinn"))

## Joining, by = "word"

#Computin mean sentiment per day
avg_sentiment_by_day <- tidy_tweets %>% group_by(date) %>% summarize(avg_sent = mean(value
))


#Average sentiment per day plotted
ggplot(avg_sentiment_by_day, aes(x = date, y = avg_sent)) +
 geom_line(color = "steelblue3")  +
 labs(title = "Average Twitter Sentiment Per Day", x = "Time", y = "Average Sentiment") +
 theme_classic()+
 theme(legend.position = "none")

#ggsave("average twitter sentiment per day.png")
```

Distribution of Trumps Twitter Sentimetn using the AFFIN dictionary

```
tidy_tweets
ggplot(tidy_tweets, aes(x = value)) +
 geom_bar(fill = "steelblue3")+
 theme_classic()+
 labs(title = "Trump's Word Frequency Using AFINN Dictionary", y = "Frequency", x = "AFINN Sentim
ent Value")+
 scale_x_continuous(breaks = seq(-4, 5, by = 1))

ggsave("WordFrequencyUSingAFINNDictionary.png")

## Saving 5 x 4 in image
```

Creating MA(5) of Average Daily Sentiment and Plotting with Approval Ratings

```
#Calculating 5 day moving average of Twitter sentiment
avg_sentiment_by_day <-  avg_sentiment_by_day %>% mutate(avg_sent_ma_05 = zoo::rollmean(av
g_sent, k = 5, fill = NA))

#Joining approval ratings with avg_sentiment
app_and_sent <- inner_join(avg_sentiment_by_day, app_data)

## Joining, by = "date"

#Subtracting 35 from approval rating in order to have sentiment and approval near eachother
app_and_sent$approve_estimate_lower <- app_and_sent$approve_estimate - 41

app_and_sent %>% select(date, avg_sent, approve_estimate_lower) %>%
 ggplot(aes(x = date))+
 geom_line(aes(y = approve_estimate_lower, color = "firebrick2"))+
 geom_ma(aes(y = avg_sent), ma_fun = SMA, n = 5, color = "steelblue3", linetype = "solid") +
 labs(title = "MA(5) of Daily Sentiment and Approval Ratings", x = "Time", y = "Average Sentiment")+
 theme_classic()+
 theme(legend.position = "none")
```

```
#ggsave("five_day_moving_average_plot.png")
```

Scatter Plot of approval ratings and average daily sentiment

```
app_and_sent

ggplot(app_and_sent, aes(x = avg_sent, y = approve_estimate)) +
  geom_point(color = "steelblue3") +
  geom_smooth(method = "lm", color = "firebrick2", se = FALSE) +
  labs(title = "Scatter Plot of Average Daily Sentiment and Daily Approval Ratings", x = "Average Daily
Sentiment", y = "Daily Approval Ratings")+
  theme_classic()

## `geom_smooth()` using formula 'y ~ x'

#ggsave("Scatterplot_Daily_Sentiment_and_Approval_Ratings.png")
```

Testing Stationary of Approval Ratings and Average Sentiment

```
library(tseries)

## Warning: package 'tseries' was built under R version 4.0.5

#Stationarity test of average sentiment
adf.test(avg_sentiment_by_day$avg_sent, k = 0)

## Warning in adf.test(avg_sentiment_by_day$avg_sent, k = 0): p-value smaller than
## printed p-value

#Stationarity test of approval ratings
adf.test(app_data$approve_estimate, k = 0)
```

ACF Plots of Approval Ratings and Average Sentiment

```
#Autocorrelation plots of approval ratings
acf(app_data$approve_estimate, plot = TRUE) #Clear trend

#Autocorrelation plots of average sentiment
acf(avg_sentiment_by_day$avg_sent, plot = TRUE)
```

First Order Differencing of Approval Rating Data

```
#Converting appa and sent to a DT for differencing
app_and_sentDT <- data.table(app_and_sent)

#Differencing approval estimate
app_and_sentDT[, approval_diff_1 := approve_estimate - shift(approve_estimate, type = "lag", n = 1)
]

#Data is now stationary
adf.test(app_and_sentDT$approve_estimate, k = 0)

## Warning in adf.test(app_and_sentDT$approve_estimate, k = 0): p-value smaller
## than printed p-value

#plotting the differenced approval ratings
ggplot(app_and_sentDT, aes(x = date, y = approval_diff_1)) +
```

```
 geom_line(color = "steelblue3")+
 labs(title = "Differenced Approval Ratings", y = "Approval Ratings", x = "Time")+
 theme_classic()

## Warning: Removed 1 row(s) containing missing values (geom_path).

#ggsave("differenced_approval_ratings.png")
```

Modeling

Model 1 - All Tweets

```
#Regression model containing all tweets
model.1 <- lm(approval_diff_1~avg_sent, data = app_and_sentDT)
#model 1 output
summary(model.1)

#Residual Plots
par(mfrow = c(2, 2))  # Create plotting columns
plot(model.1)  # Plot the model information

#Scatter plot of model 1
ggplot(app_and_sentDT, aes(x = avg_sent, y = approve_estimate)) +
 geom_point(color = "steelblue3")+
 geom_smooth(method = "lm", se = FALSE, color = "firebrick2") +
 labs(title = "Scatter Plot of Trump's Twitter Sentiment and Approval Ratings", x = "Average Twitter S
entiment", y = "Average Approval Ratings")+
 theme_classic()

## `geom_smooth()` using formula 'y ~ x'

#ggsave("Scatter Plot of Trump's Twitter Sentiment and Approval Ratings.png")
```

Model 1 Cooks Distance

```
#Calculating cooks distance
cooks_distance <- cooks.distance(model.1)

#Plotting Cooks Distance
plot(cooks_distance, main = "Influential Observations Identified by Cooks Distance")
#Creating horizontal line for values greater than 3 times the mean
abline(h = 3*mean(cooks_distance), lty = 1, col = "steelblue3", lwd = 2)
#Labeling values which are greater than 10 times the mean
text(x=1:length(cooks_distance)+1, y=cooks_distance, labels=ifelse(cooks_distance>10*mean(cooks
_distance), names(cooks_distance),""), col="firebrick2")
```

Removing Outliers Found by Cooks Distance

```
cooksD <- cooks.distance(model.1)
influential <- cooksD[(cooksD > (3 * mean(cooksD, na.rm = TRUE)))]#Sorting observations with Cook's
distance greater than 3 times the mean

#Turning back to dataframe so I can sort for outliers and keep the differenced data
app_and_sent_outliers <- as.data.frame(app_and_sentDT)

#Identifying row index for outliers
names_of_influential <- names(influential)
```

```
outliers <- app_and_sent_outliers[names_of_influential, ]

#Anti-joining to remove outliers
app_and_sent_outliers <- anti_join(app_and_sent_outliers, outliers)

## Joining, by = c("date", "avg_sent", "avg_sent_ma_05", "subgroup", "approve_estimate", "approve
_estimate_lower", "approval_diff_1")

#ACF of data wihtout outliers
acf(na.omit(app_and_sent_outliers$approval_diff_1), plot = TRUE)
```

Model 2 - Regression Model Without Outliers

```
#New model without outliers
model.2 <- lm(approval_diff_1~avg_sent, data = app_and_sent_outliers)
summary(model.2)

par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(model.2)
```

Differenced Approval Ratings With and Without Outliers

```
#Using grid arrange to combine plots
approval_with_and_without_outliers <- gridExtra::grid.arrange(

#plotting the differenced approval ratings
ggplot(app_and_sentDT, aes(x = date, y = approval_diff_1)) +
 geom_line(color = "steelblue3")+labs(title = "Differenced Approval Ratings", y = "Approval Ratings",
x = "Time")+
 theme_classic(),


#Approval ratings without outliers
ggplot(app_and_sent_outliers, aes(x = date, y = approval_diff_1)) +
 geom_line(color = "steelblue3")+labs(title = "Differenced Approval Ratings Without Outliers", y = "A
pproval Ratings", x = "Time")+
 theme_classic()
)

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 1 row(s) containing missing values (geom_path).

#ggsave("approval_ratings_with_and_without_outliers.png", approval_with_and_without_outliers)
```

Segmenting tweets

```
#Segmenting tweets based on news keywords
news_tweets <- clean_tweets[grep("fake.*news|fake.*media|cnn|fake|news", clean_tweets$text,
value=FALSE, ignore.case = TRUE), ] #Time trump tweets about fake news

#Segmenting tweets based on democratic keywords
```

```
left_tweets <- clean_tweets[grep("democrats|joe|biden|hillary|left|sleepy joe", clean_tweets$text,
value=FALSE, ignore.case = TRUE), ]

#Segmenting tweets based on Border keywords
Border_tweets <- clean_tweets[grep("immigration|immigrant|Border.*wall|wall.*Border|southern.
*Border|illegal.*immigrants|Border.*security|mexican[s]|mexico", clean_tweets$text, value=FALSE
, ignore.case = TRUE), ]
```

Visualizing Border Tweets Per Year

```
#Creating column to group tweets by year
Border_tweets_year <- Border_tweets %>% mutate("Year" = ifelse(date <= "2017-12-31", "2017",
                                        ifelse(date <= "2018-12-31", "2018",
                                            ifelse(date <= "2019-12-31", "2019", "2020-2021"))))

#Plotting Border tweet frequncy by year
board_plot <- Border_tweets_year %>% count(Year)
 ggplot(board_plot, aes(x = Year, y = n)) +
 geom_bar(stat = "identity", fill = "steelblue3")+
 labs(title = "Border Tweet Frequency by Year", y = "Frequency")+
 theme_classic()+
 theme(legend.position = "none")

#ggsave("Border_tweet_frequency_by_year.png")
```

News tweets

Tokenizing News Tweets

```
#Unest tokens
tidy_news_tweets <- news_tweets %>% mutate(linenumber = row_number()) %>%  unnest_tokens(
word, text)

#Creating custom stop words
custom_stop_words <- tribble(
   ~word,     ~lexicon,
   "íí","CUSTOM",
   "get","CUSTOM",
   "like","CUSTOM",
   "just","CUSTOM",
   "yes","CUSTOM",
   "know","CUSTOM",
   "will","CUSTOM",
   "good","CUSTOM",
   "day","CUSTOM",
   "people","CUSTOM",
   "amp", "CUSTOM",
   "dont", "CUSTOM",
   "trump", "CUSTOM",
   "president", "CUSTOM"

)
#Combining custom stop words with predifined stopwords
```

```
stop_words2 <- stop_words %>%
bind_rows(custom_stop_words)

#Anti-joining news tweets with stop words
tidy_news_tweets <- tidy_news_tweets %>% anti_join(stop_words2)

## Joining, by = "word"
```

Calculating Average News Tweet Sentiment

```
#Joining news tweet tokens with 'afinn' dictionary
tidy_news_tweets <- tidy_news_tweets %>%
 inner_join(get_sentiments("afinn"))

## Joining, by = "word"

#Calculating averae daily sentiment
avg_news_sentiment_by_day <- tidy_news_tweets %>% group_by(date) %>% summarize(news_avg
_sent = mean(value))

#Average sentiment per day plotted
ggplot(avg_news_sentiment_by_day, aes(x = date, y = news_avg_sent)) +
 geom_line(color = "steelblue3")+
 labs(title = "Average News Tweet Sentiment Per Day", x = "Time", y = "Average Sentiment") +
 theme_classic()+
 theme(legend.position = "none")

#ggsave("average news tweet sentiment.png")
```

Creating MA(5) of News Tweet Sentiment

```
#Creating 5 day moving average data from sentimetn data
avg_news_sentiment_by_day <-  avg_news_sentiment_by_day %>% mutate(avg_news_sent_ma_05
= zoo::rollmean(news_avg_sent, k = 5, fill = NA))

app_and_sentDT_news <- inner_join(app_and_sentDT, avg_news_sentiment_by_day)

## Joining, by = "date"
```

Regression Model with News Tweets

```
#Regression Model
model_news <- lm(approval_diff_1 ~ news_avg_sent, app_and_sentDT_news)
summary(model_news)

#Scatter Plot of Regression Model
ggplot(app_and_sentDT_news, aes(x = approve_estimate, y = news_avg_sent)) +
 geom_point(color = "steelblue3") +
 geom_smooth(method = "lm", se = FALSE, color = "firebrick1")+
 labs(x = "Approval Ratings", y = "Average Sentiment", title = "Scatter Plot of Approval Ratings and A
verage News Sentiment")+
 theme_classic()

## `geom_smooth()` using formula 'y ~ x'

#ggsave("scatterplot_approval_ratings_and_average_news_sentiment.png")
```

```
#Plotting model residuals
par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(model_news)
```

Democrat tweets

Tokenizing Segmented Democratic Tweets

```
#Unest tokens
tidy_left_tweets <- left_tweets %>% mutate(linenumber = row_number()) %>%  unnest_tokens(wor
d, text)

#Creating custom stop words
custom_stop_words <- tribble(
  ~word,     ~lexicon,
  "íí","CUSTOM",
  "get","CUSTOM",
  "like","CUSTOM",
  "just","CUSTOM",
  "yes","CUSTOM",
  "know","CUSTOM",
  "will","CUSTOM",
  "good","CUSTOM",
  "day","CUSTOM",
  "people","CUSTOM",
  "amp", "CUSTOM",
  "dont", "CUSTOM",
  "trump", "CUSTOM",
  "president", "CUSTOM"

)
#Combining custom stopwords with predefined stopwords
stop_words2 <- stop_words %>%
bind_rows(custom_stop_words)

#Removing stopwords from democratic tweets
tidy_left_tweets <- tidy_left_tweets %>% anti_join(stop_words2)

## Joining, by = "word"
```

Calculating Average Sentiment of Democratic Tweets

```
#Joining democratic tokens with 'afinn' dictionary
tidy_left_tweets <- tidy_left_tweets %>%
 inner_join(get_sentiments("afinn"))

## Joining, by = "word"

#Groupoing tokens by date and summing sentiment
avg_left_sentiment_by_day <- tidy_left_tweets %>% group_by(date) %>% summarize(left_avg_sent
= mean(value))

#Average sentiment per day plotted
```

```
ggplot(avg_left_sentiment_by_day, aes(x = date, y = left_avg_sent)) +
 geom_line(color = "steelblue3")  +
 labs(title = "Average Democratic Tweet Sentiment Per Day", x = "Time", y = "Average Sentiment") +
 theme_classic() +
 theme(legend.position = "none")

#ggsave("Average Democratic Tweet Sentiment Per Day.png")
```

Creating MA(5) of Democratic Tweet Average Daily Sentiment

```
#Creating 5 day moving average data from sentimetn data
avg_left_sentiment_by_day <-  avg_left_sentiment_by_day %>% mutate(avg_left_sent_ma_05 = zoo
::rollmean(left_avg_sent, k = 5, fill = NA))

#Joining democratic moving average with main DF
app_and_sentDT_left <- inner_join(app_and_sentDT, avg_left_sentiment_by_day)

## Joining, by = "date"
```

Regression Model with Democratic Tweets

```
#Regression Model
model_left <- lm(approval_diff_1 ~ left_avg_sent, app_and_sentDT_left)
summary(model_left)

#Scatterplot of LM
ggplot(app_and_sentDT_left, aes(x = approve_estimate, y = left_avg_sent)) +
 geom_point(color = "steelblue3") +
 geom_smooth(method = "lm", se = FALSE, color = "firebrick1")+
 labs(title ="Scatter Plot of Approval Ratings and Average Democratic Tweet Sentiment", x = "Approv
al Rating", y = "Average Sentiment")+
 theme_classic()

## `geom_smooth()` using formula 'y ~ x'

#ggsave("Scatter Plot of Approval Ratings and Average Democratic Tweet Sentiment.png")

#Plotting Model residuals
par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(model_left)
```

Border tweets

Tokenizing Segmented Border Tweets

```
#Unest tokens
tidy_Border_tweets <- Border_tweets %>% mutate(linenumber = row_number()) %>%  unnest_toke
ns(word, text)

#Creating custom stop words
custom_stop_words <- tribble(
   ~word,     ~lexicon,
   "íí","CUSTOM",
   "get","CUSTOM",
   "like","CUSTOM",
```

```
  "just","CUSTOM",
  "yes","CUSTOM",
  "know","CUSTOM",
  "will","CUSTOM",
  "good","CUSTOM",
  "day","CUSTOM",
  "people","CUSTOM",
  "amp", "CUSTOM",
  "dont", "CUSTOM",
  "trump", "CUSTOM",
  "president", "CUSTOM"

)
#Joining custom stopwords with predefined stopwords
stop_words2 <- stop_words %>%
bind_rows(custom_stop_words)

#Removing stopwords from Border tweet tokens
tidy_Border_tweets <- tidy_Border_tweets %>% anti_join(stop_words2)

## Joining, by = "word"
```

Finding average sentiment for Border tweets

```
#Joining Border tweet tokens with 'afinn' dictionary
tidy_Border_tweets <- tidy_Border_tweets %>%
 inner_join(get_sentiments("afinn"))

## Joining, by = "word"

#Grouping sentiment by date and summing
avg_Border_sentiment_by_day <- tidy_Border_tweets %>% group_by(date) %>% summarize(Border
_avg_sent = mean(value))



#Average sentiment per day plotted
ggplot(avg_Border_sentiment_by_day, aes(x = date, y = Border_avg_sent, color = "red")) +
 geom_line(color = "steelblue3")  +
 labs(title = "Average Border Sentiment Per Day", x = "Time", y = "Average Sentiment") +
 theme_classic()+
 theme(legend.position = "none")

#ggsave("average Border sentiment tweets.png")
```

Creating MA(5) of Border Tweets

```
#Creating 5 day moving average data from sentimetn data
avg_Border_sentiment_by_day <-  avg_Border_sentiment_by_day %>% mutate(avg_Border_sent_m
a_05 = zoo::rollmean(Border_avg_sent, k = 5, fill = NA))
#Joining Border tweet moving average with main DF
app_and_sentDT_Border <- inner_join(app_and_sentDT, avg_Border_sentiment_by_day)

## Joining, by = "date"
```

Regression Model with Border Tweets

```
#Regression model with Border tweets
model_Border <- lm(approval_diff_1 ~ Border_avg_sent, app_and_sentDT_Border)
summary(model_Border)

#Scatter plot of Border tweet regression model
ggplot(app_and_sentDT_Border, aes(x = approve_estimate, y = Border_avg_sent)) +
 geom_point(color = "steelblue3") +
 geom_smooth(method = "lm", se = FALSE, color = "firebrick1")+
 labs(x = "Approval Rating", y = "Average Sentiment", title = "Scatter Plot of Approval Ratings and Av
erage Border Tweet Sentiment")

## `geom_smooth()` using formula 'y ~ x'

#ggsave("Scatter Plot of Approval Ratings and Average Border Tweet Sentiment.png")

#Redidual Plot of Border Tweet regression
par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(model_Border)
```

Moving Average Plot of All Tweet Subset Average Daily Sentiment

```
#Combining all plots with grid.arrange
all_tweet_plot <- gridExtra::grid.arrange(

#plot of Border tweets
ggplot(avg_Border_sentiment_by_day, aes(x = date, y  = avg_Border_sent_ma_05)) +
 geom_line(color = "firebrick2") +
 labs(y = "Border Sentiment")+
 theme_classic()+
 theme(legend.position = "none", axis.title.x = element_blank()),
#Plot of democratic tweets
ggplot(avg_left_sentiment_by_day, aes(x = date, y  = avg_left_sent_ma_05)) +
 geom_line(color = "steelblue3")+
 labs(y = "Dem. Sentiment")+
 theme_classic()+
 theme(legend.position = "none", axis.title.x = element_blank()),
#Plot of news tweets
ggplot(avg_news_sentiment_by_day, aes(x = date, y  = avg_news_sent_ma_05)) +
 geom_line()+
 labs(y = "News Sentiment")+
 theme_classic()+
 theme(legend.position = "none", axis.title.x = element_blank()),

top = "Average Tweet Sentiment by Topic Subsets"


ggsave("ma of all tweet subsets.png", all_tweet_plot)

## Saving 5 x 4 in image
```

Calculating Sum of Sentiment by Subset

```
avg_Border_sentiment_by_day %>% summarise("Aveage Border Sentiment" = mean(Border_avg_se
nt))

avg_left_sentiment_by_day %>% summarise("Aveage Left Sentiment" = mean(left_avg_sent))

avg_news_sentiment_by_day %>% summarise("Aveage News Sentiment" = mean(news_avg_sent))
```

Parts of Speech Tagging

Downloading Pretrained Model

```
#download and load the pre-trained models
#udmodel <- udpipe_download_model(language = "english")
#udmodel <- udpipe_load_model(file = udmodel$file_model)

#annotate the data frame with udpipe model
#annotated_tweets <- udpipe_annotate(udmodel, x = clean_tweets$text)
#annotated_tweets <- as.data.frame(annotated_tweets)
```

Saving Annotated Tweets as RDS object

```
#saveRDS(annotated_tweets, file = "annotated_tweets.rds")
annotated_tweets <- readRDS(file = "annotated_tweets.rds")
```

Extracting Keyword Phrases using Simple Noun Phrases

```
annotated_tweets$phrase_tag <- as_phrasemachine(annotated_tweets$upos, type = "upos")
stats2 <- keywords_phrases(x = annotated_tweets$phrase_tag, term = tolower(annotated_tweets$t
oken),
                           pattern =  "(A|N)*N(P+D*(A|N)*N)*",
                           is_regex = TRUE, detailed = FALSE)

stats2 <- subset(stats2, ngram > 1 & freq > 125)
stats2$key <- factor(stats2$keyword, levels = rev(stats2$keyword))
stats2 %>%
 ggplot(aes(x = reorder(keyword, freq), y =  freq)) +
 geom_col(fill = "steelblue3") +
 coord_flip() +
 geom_text(aes(label = freq, vjust = 0, hjust = -0.3 )) +
 xlab("Keywords")+
 ylab("Frequency")+
 ggtitle("Keywords Identified by POS Tags - Simple Noun Phrases") +
 theme_classic()+
 theme(legend.position = "none")

#ggsave("Keywords Identified by POS Tags - Simple Noun Phrases.png")
```