UNIDAD DE MAESTRÍAS Y POSGRADOS EN ECONOMÍA

# PFIZER: Everybody's Talkin'

## Sentiment Analysis of Pfizer-Related Tweets and Their Correlation to Pfizer Stock's Movements (2020-2022)

by

**María Victoria Friss de Kereki Tosar**

Final project presented to qualify
for the degree of MSc Finance

Thesis tutor
Jonhatan Lacuesta

Montevideo, Uruguay
2023

UNIVERSIDAD DE MONTEVIDEO

UNIDAD DE MAESTRÍAS Y POSGRADOS EN ECONOMÍA

**María Victoria Friss de Kereki Tosar**

# PFIZER: EVERYBODY’S TALKIN’

## Sentiment Analysis of Pfizer-Related Tweets and Their Correlation to Pfizer Stock’s Movements (2020-2022)

Tutor Jonhatan Lacuesta

TFC presentado para aspirar al título de Máster en Finanzas

Juicio del Tribunal:

---

Recomendación para su publicación en el Repositorio de la UM:

……………………………………………………………………

……………………………………………………………………

……………………………………………………………………

Presidente: …………………………………………………………….

(Firma) (Aclaración)

Secretario: …………………………………………………………….

(Firma) (Aclaración)

Vocal: …………………………………………………………….

(Firma) (Aclaración)

Montevideo, …. de …. de 2023

## Disclaimer

The author of this final degree project declares that she is the only one responsible for its content, in particular for the opinions expressed in it, which the University of Montevideo does not necessarily share; in addition, she declares that no third-party right is infringed, be it intellectual, industrial or other. Consequently, she is the only person responsible and can exclusively assume any claims from third parties (individuals or legal entities) that refer to the work's authorship or other related aspects, including the plagiarism claim.

## Descargo de responsabilidad

La autora de este trabajo final de carrera declara que es la única responsable de su contenido, y en particular de las opiniones expresadas en él, las que no necesariamente son compartidas por la Universidad de Montevideo; asimismo, declara que no se infringe ningún derecho de terceros, ya sea de propiedad intelectual, industrial o cualquier otro. En consecuencia, es la única responsable y de manera exclusiva puede asumir eventuales reclamaciones de terceros (personas físicas o jurídicas) que refieran a la autoría de la obra y a otros aspectos vinculados a ésta, incluido el reclamo por plagio.

# Acknowledgements

I want to express my sincere gratitude to the University of Montevideo, and especially to Alejandro Cid and Daniel Ferrés, for providing me with the opportunity to pursue this Master's degree. The guidance and support from my professors and advisors have been invaluable throughout my academic journey, and the University's commitment to excellence in education has played a crucial role in shaping me into the person I am today.

Moreover, I am thankful to my parents, Sylvia Tosar and Federico Friss de Kereki, for their continuous support, love and encouragement throughout my academic journey. Their forever patience, guidance, and understanding have been instrumental in helping me achieve my goals. I would also like to express a special thank you to the latter for his exceptional help and guidance throughout the process of writing this project. His dedication and expertise have truly made a significant impact, and I am deeply grateful for his contributions.

Additionally, I extend my heartfelt thanks to my friends, who have been a constant source of encouragement and support. Your unwavering support and presence have helped me overcome difficult times, and I am grateful to have you in my life. Your belief in me has motivated me to strive for excellence in all walks of life.

Lastly, I want to convey my sincere thank you to all who contributed to my academic and personal growth, including my classmates, colleagues and mentors. Your insights, knowledge, and experiences have been invaluable, and I am grateful for the opportunity to have worked with and learned from each of you.

Once again, thank you to everyone who has played a role in my academic journey. Your support and encouragement have been invaluable, and I look forward to the next chapter of my life with confidence and gratitude.

“*Everybody’s talkin’ at me*
*I don’t hear a word they’re sayin’*”
Fred Neil in “*Everybody’s Talkin’*”, 1966

## Abstract

This project presents a comprehensive exploration of Twitter data, focusing on discussions surrounding the Pfizer biopharmaceutical company, a developer of a COVID-19 vaccine. Spanning from January 2020 to December 2022, the study involves the analysis of tweets that refer to the company, utilising diverse sentiment analysis techniques (Bing, AFINN, NRC, SentimentR), each examined individually and compared to one another. The project further investigates the interplay between sentiment indexes and Pfizer stock's trading volume and daily returns, illuminating potential connections with the market. Employing data manipulation, visualisation, and statistical analysis, these analyses enhance understanding of sentiment dynamics, market interactions, and the relationship between public sentiment and Pfizer stock's performance.

*Keywords: Correlation, Natural Language Processing, Pfizer, Sentiment Analysis, Stocks, Twitter, Twitter Sentiment Analysis*

## Resumen

Este proyecto presenta una exploración integral de datos de Twitter, centrada en torno a la empresa biofarmacéutica Pfizer, desarrolladora de una vacuna contra COVID-19. Abarcando desde enero de 2020 hasta diciembre de 2022, el estudio implica un análisis de tweets referidos a la empresa utilizando diversas técnicas de análisis de sentimiento (Bing, AFINN, NRC, SentimentR), cada una examinada individualmente y comparada con las demás. Además, el proyecto investiga la interacción entre los índices de sentimiento y el volumen de negociación y rendimientos diarios de las acciones de Pfizer, arrojando luz sobre posibles conexiones con el mercado. Mediante manipulación de datos, visualización y análisis estadístico, estos análisis mejoran la comprensión de las dinámicas de sentimiento, las interacciones del mercado y la relación entre el sentimiento público y el rendimiento de las acciones de Pfizer.

*Palabras Clave: Acciones, Análisis de Sentimiento, Análisis de Sentimiento de Twitter, Correlación, Pfizer, Procesamiento de Lenguaje Natural, Twitter*

# Table of Contents

# List of Tables

# List of Figures

# 1. Introduction

This chapter describes the investigation's background, the statement of the problem, the purpose of the study, the research questions, and the document's structure.

## 1.1 Background

On March 11th, 2020, the World Health Organization (WHO) characterised the COVID-19 outbreak as a pandemic, which meant that the public eye started turning to a particular industry: the pharmaceutical one.[1] While in the labs the race to create the first effective and approved vaccine started gaining momentum, on social media everyone around the world would have something to say about companies like Pfizer,[2] BioNTech,[3] Moderna,[4] and others from the same industry.

Despite its vast volume of 500 million daily tweets,[5] which positions Twitter[6] as an immense source of raw data, it remains vastly underutilised in its potential.[7] This is mainly because processing vast amounts of tweets in an efficient way that may enable the obtention of valuable insights can be a troublesome and time-consuming task, which companies are usually not willing to handle. Does this volume of tweets and ideas expressed in them relate to financial indexes? Or are they completely independent and non-related events?

## 1.2 Statement of the Problem

This project will focus on Pfizer, one of the world's top multinational biopharmaceutical companies. Founded in 1849, their "*commitment to finding innovations that help*

[1] WHO (2020)

[2] https://www.pfizer.com/

[3] https://www.biontech.com/

[4] https://www.modernatx.com/

[5] A "*tweet*" is a short message posted on the social media platform Twitter. This will be developed in Section 2.5 *Twitter*.

[6] https://twitter.com/

[7] Ahlgren, M., & Team, W. (2023)

*people*"has withstood. Its vision, as declared on its web page, is to be "*in relentless pursuit of breakthroughs that change patients' lives*".[8]

Pfizer is a publicly traded company on *The New York Stock Exchange* (NYSE)[9] under the ticker PFE.[10] In this project, the problem to be studied is the relationship between tweets about the company and movements in its stock price and trading volume.

## 1.3 Purpose of the Study

The main objective of this investigation is to examine the possible correlation between the tweets related to Pfizer and the company's stock price and traded volume. Specifically, it will be analysed whether fluctuations in public opinion, as extracted from tweets shared by users worldwide, have a measurable relationship to the company's stock market performance. By studying the trends in tweets (all languages) related to Pfizer between 2020 and 2022 and the sentiment expressed by tweets (in English only), the aim is to gain insights into the dynamics between social media sentiment and financial markets and contribute to the ongoing discussion on the relationship between online chatter and financial indexes.

## 1.4 Research Questions

The research questions that will be answered with this investigation are the following:

- What trends can be identified in the number and contents of tweets containing the word Pfizer between 2020 and 2022, primary years of the COVID-19 pandemic?[11]
- Do the previously analysed trends relate to Pfizer stock's trading volume and price movements?[12]

[8] Pfizer (2023)

[9] https://www.nyse.com

[10] https://finance.yahoo.com/quote/PFE/

[11] This question is answered in Section 6.1 *Tweets Dataset Descriptive Analysis*.

[12] This question is answered in Chapter 9 *Interpret*, through the analysis carried out in Chapter 7 *Explore – Sentiment Analysis* and Chapter 8 *Model*.

- How can the answers to the previous questions enhance the activity of the Pfizer Finance Team?[13]

## 1.5 Structure

The structure of this document is the following:

- Chapter 1 *Introduction* provides the background and states the problem while also defining the purpose of the study and presenting the research questions;
- Chapter 2 *Literature Review* covers critical concepts such as technical analysis of stock pricing, alternative data for stock pricing, natural language processing, sentiment analysis, Twitter, Twitter sentiment analysis, and correlation analysis;
- Chapter 3 *Methodology and Data* describes the type of investigation and the OSEMN methodology employed, discusses ethical considerations related to data extraction from Twitter and of the Pfizer Stock Dataset, and outlines the tools and packages used;
- Chapter 4 *Obtain* presents the extraction of the Tweets Dataset and of the Pfizer Stock Dataset;
- Chapter 5 *Scrub* focuses on the importation of the Tweets Dataset into R and the conduction of quality revision;
- Chapter 6 *Explore – Descriptive Analysis* includes the descriptive analysis of the Tweets Dataset and the Pfizer Stock Dataset, exploring various aspects such as general numbers, trends, outliers, tweet types, tweet sources, languages, users, and most frequent words in English tweets;
- Chapter 7 *Explore – Sentiment Analysis* presents the sentiment analysis of tweets, including methods Bing, AFINN, NRC, and SentimentR, to analyse the sentiment of tweets, while an evaluation and comparison of these methods is also conducted;
- Chapter 8 *Model* shows the correlation analysis between positivity indexes and the number of tweets, and PFE's traded volume and daily returns;

[13] This question is answered in Section 10.2 *Applications for Pfizer*.

- Chapter 9 *Interpret* and Chapter 10 *Conclusions, Applications and Future Work* provide insights and conclusions drawn from the analysis, and also future work suggestions.

The project concludes with a bibliography, a glossary, and several appendixes containing code snippets related to the Tweets Dataset extraction, importation and quality revision, the Pfizer Stock Dataset download, the descriptive analysis and sentiment analysis, and more.

# 2. Literature Review

This chapter presents a systematic review of the existing literature on technical analysis of stock pricing, alternative data for stock pricing, natural language processing, sentiment analysis, Twitter, Twitter sentiment analysis, and correlation analysis. The various techniques and algorithms used in these fields are explored, as well as the applications and limitations of these techniques. The review provides a foundation for the research undertaken in this project, which used these techniques to analyse the sentiment of tweets related to Pfizer.

## 2.1 Technical Analysis of Stock Pricing

*Technical analysis* is a method used in finance to evaluate securities by analysing statistics generated by market activity, for example past prices and volume. This method is used to identify patterns, trends, and signals that can help investors make better-informed decisions regarding buying or selling securities.[14]

In the context of the present project, understanding technical analysis in finance is crucial for drawing connections with sentiment analysis.[15] Technical analysis involves evaluating securities based on historical market data and identifying patterns, trends, and signals to make informed decisions about buying or selling securities. In contrast, sentiment analysis focuses on assessing public opinions and emotions expressed in textual data, such as social media posts, particularly tweets. A more comprehensive understanding of stock price movements emerges by combining technical and sentiment analysis. This integration allows investors to gain insights from past market activity while also considering real-time sentiments and perceptions of the public, investors, and customers regarding a specific stock, company, or financial event. Such a holistic approach can lead to more informed investment decisions.[16]

[14] Petrusheva, N., & Jordanoski, I. (2016)

[15] Though technical analysis is not withing the scope of this investigation, references to the inclusion of it as complimentary to sentiment indexes are included principally in Chapter 10 *Conclusions, Applications and Future Work*.

[16] Deng, S., Mitsubuchi, T., Shioda, K., Shimada, T., & Sakurai, A. (2011)

The literature on technical analysis offers different perspectives on its usefulness and effectiveness.

- Taylor and Allen explain that a questionnaire survey conducted in November 1988 among chief foreign exchange dealers based in London revealed that at least 90% of them use technical analysis to some extent when forming their views on currency markets. The dealers relied more on technical analysis for shorter time horizons and gradually shifted towards fundamental analysis for longer horizons. Most viewed technical and fundamental analysis as complementary, and some suggested that technical advice may become self-fulfilling.[17]
- One study by Brock et al. tests the effectiveness of two popular trading strategies, moving average and trading range break, by analysing a long-term data series of the *Dow Jones index* from 1897 to 1986. The authors use bootstrap techniques to analyse the data and find that both strategies generate positive returns. However, the study suggests that transaction costs should be carefully considered before implementing these strategies, and the returns-generating process of stocks is more complicated than suggested by linear models. The paper concludes that technical rules may pick up some hidden patterns, and more elaborate rules may generate even more considerable differences in returns.[18]
- Bessembinder and Chan explore the finding that basic technical analysis techniques can help predict US equity indices' returns, as Brock et al. reported in 1992. They discover that the predictive ability may be partially due to measurement errors in trading returns, but not entirely. They also argue that this evidence does not necessarily contradict the idea of market efficiency. The study calculates the estimated trading costs to be relatively low compared to actual trading costs, suggesting that technical analysis can still be profitable even after accounting for transaction costs.[19]
- Park and Irwin suggest that early studies indicated profitability in foreign exchange and futures markets for technical trading strategies but not in stock

[17] Taylor, M. P., & Allen, H. (1992)

[18] Brock, W. A., Lakonishok, J., & LeBaron, B. (1992)

[19] Bessembinder, H., & Chan, K. (1998)

markets before the 1980s. In contrast, modern studies indicate that technical trading strategies were profitable in various speculative markets, at least until the early 1990s. Although there is supportive evidence regarding the profitability of technical trading strategies, the paper proposes that numerous empirical investigations encounter challenges in their testing methodologies. These issues encompass concerns like data snooping, ex-post selection of trading rules or search methodologies, as well as complications in accurately estimating risk and transaction expenses. Therefore, the paper suggests that future research must address these deficiencies in testing to offer decisive proof regarding the profitability of technical trading strategies.[20]

- Blume et al. investigate the informational role of volume in the stock market and its applicability to technical analysis. They develop an equilibrium model where traders receive signals of different qualities and show that volume provides information on information quality that cannot be deduced from the price statistic. The paper also demonstrates how traders who use information in market statistics, including volume, do better than those who do not. The authors argue that technical analysis is a natural component of the agents' learning process.[21]
- Wong et al. examine the effectiveness of technical analysis in predicting stock market entry and exit points using two popular indicators, the *Moving Average* and the *Relative Strength Index*. The study used data from the *Singapore Stock Exchange* (SES) and found that technical indicators could generate significant positive returns, with member firms of the SES enjoying substantial profits using technical analysis. The paper suggests that member firms' trading teams' widespread use of technical analysis could be attributed to its success in generating profits.[22]

In summary, the literature on technical analysis of stock pricing suggests that it can be a valuable tool for investors in evaluating securities and making informed decisions. Studies have shown positive returns from trading strategies based on technical analysis,

[20] Park, C. H., & Irwin, S. H. (2004)

[21] Blume, L. E., Easley, D., & O'Hara, M. (1994)

[22] Wong, W., Manzur, M., & Chew, B. (2003)

although transaction costs should be considered. While empirical research acknowledges some testing issues, there is evidence of profitability in using technical analysis, particularly in speculative markets. Overall, technical analysis complements fundamental analysis and can provide valuable insights for traders and investors.[23]

## 2.2 Alternative Data for Stock Pricing

*Alternative data* has become a hot topic in the financial industry in recent years, with the rise of big data and the proliferation of data sources available for analysis. Nowadays, it is as easy as ever to access infinite data sources that, with adequate processing, can become alternative sources of information for stock analysis and pricing.[24] This can be as varied as one's imagination can conceive, looking beyond companies' formal statements and books. Alternative data sources include social media sentiment, news sentiment, product reviews, web traffic, app usage, surveys, and satellite imagery, among others.[25]

In the project developed in this document, the primary focus centres on sentiment analysis, which involves the analysis of social media sentiment, specifically tweets related to Pfizer.[26] Sentiment analysis serves as a crucial alternative data source, providing valuable insights into public opinions and emotions surrounding the company.

The Deloitte Center for Financial Services discussed the potential benefits of using alternative data for investment decision-making. Their report notes that more than traditional data sources may be needed for investors to gain a competitive edge in the market and that alternative data, such as social media sentiment, satellite imagery, and transactional data, can provide new insights into companies and industries. The authors note that while alternative data has the potential to revolutionise investment decision-

[23] Similar ideas will be presented in Chapter 9 *Interpret* and Chapter 10 *Conclusions, Applications and Future Work*.

[24] Hansen, K. B., & Borch, C. (2022)

[25] Shahbaznezhad, H., Dolan, R., & Rashidirad, M. (2021)

[26] Literature review for Sentiment analysis and Twitter Sentiment Analysis will be developed in Section 2.4 *Sentiment Analysis* and Section 2.6 *Twitter Sentiment Analysis*, respectively.

making, there are also significant challenges, such as data quality, privacy concerns, and the need for sophisticated analytics tools.[27]

Cao et al. investigate the integration of artificial intelligence (AI) and traditional stock analysis methods to create more accurate investment strategies. The authors argue that AI technologies can provide a wealth of new data sources and analytics techniques that can lead to more profitable investment decisions when combined with human insights.[28]

Brown and Cliff examine the relationship between investor sentiment and asset valuation. The authors argue that investor sentiment plays a crucial role in asset pricing, mainly when there is significant uncertainty or ambiguity about the future. They found that a survey-based measure of investor sentiment can predict market returns for the next 1-3 years and can explain deviations from intrinsic value as measured by other researchers' models of stock prices. The significance of the results remained strong even after accounting for rational factors and changes in methodology.[29]

Li et al. prove that social media platforms like Twitter can be powerful social signals for predicting price movements in the highly speculative altcoin market. The authors analyse Twitter signals as a means to predict price fluctuations of a small-cap alternative cryptocurrency called ZClassic.[30] Tweets were extracted and classified as positive, neutral, or negative to create hourly sentiment indices, which were used to train a model for price prediction.[31] The resulting predictions had a high correlation with testing data and were statistically significant.[32]

Baker and Wurgler challenge the classical finance theory, in which investors' sentiment does not impact stock prices, expected returns, or actual returns. Their study shows that investor sentiment has significant cross-sectional effects. The authors found that the

[27] Deloitte (2018)

[28] Cao, S., Jiang, W., Wang, J., & Yang, B. (2021)

[29] Brown, G. M., & Cliff, M. T. (2005)

[30] https://coinmarketcap.com/currencies/zclassic/

[31] A similar approach is used in the present document and is detailed in the next chapters. However, while Li et al. used hourly indexes for price prediction, in the present investigation daily indexes were created and correlation to prices was analysed.

[32] Li, T., Chamrajnagar, A. S., Fong, X. R., Rizik, N. R., & Fu, F. (2019)

investor sentiment level at the period's beginning influenced the cross-section of future stock returns. When sentiment is high, stocks attractive to optimists and speculators but unattractive to arbitrageurs tend to have lower returns. This pattern reverses when sentiment is low. Surprisingly, firm characteristics that do not have predictive power show strong patterns when sentiment is considered. The results suggest that more than the explanation based on systematic risks is needed to account for these patterns.[33]

In a later study, Baker et al. create investor sentiment indices for six major stock markets and break them down into a global and six local indices. They find that the relative sentiment is linked to the relative prices of dual-listed companies. Global sentiment is a contrarian predictor of country-level returns, and both global and local sentiments are contrarian predictors of future market returns. When sentiment is high, future returns tend to be low on stocks that are harder to arbitrage and value. Private capital flows seem to be a way in which sentiment spreads across markets and creates global sentiment.[34]

## 2.3 Natural Language Processing

*Natural Language Processing* (NLP) is a rapidly evolving field concerned with developing algorithms and computational models that enable computers to comprehend, interpret, and generate human language.[35] In recent years, NLP has gained increasing attention due to its potential to revolutionise various industries, such as healthcare, finance, and marketing.[36]

According to Manning and Schütze, NLP involves several subfields:

- *Text categorisation* is concerned with identifying topics or themes in texts.
- *Information extraction* involves identifying specific pieces of information from unstructured text, such as named entities and events.

[33] Baker, M., & Wurgler, J. (2006)

[34] Baker, M., Wurgler, J., & Yuan, Y. (2012)

[35] Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011)

[36] Khurana, D., Koli, A. C., Khatter, K., & Singh, S. (2022)

- *Sentiment analysis* is concerned with determining the sentiment expressed in a text, such as positive or negative.[37]

NLP has roots in different fields. Bird et al. note that it encompasses a blend of computer science, artificial intelligence, and linguistics, and covers an extensive array of technologies, from simple tasks like word frequency analysis to more advanced tasks like understanding and responding to human utterances. Having its origins in several subfields, such as machine translation and speech recognition, NLP technologies are increasingly prevalent in our daily lives, powering predictive text, web search engines, and machine translation.[38]

Khurana et al. provide an overview of the state of the art, current trends, and challenges in NLP. According to them, NLP has two main components:

- *Natural Language Understanding* (NLU), and
- *Natural Language Generation* (NLG), which is a step ahead as it involves generating text.[39]

NLU allows machines to comprehend and process natural language by identifying concepts, entities, emotions, and keywords. It finds applications in customer care, where it helps to understand customer complaints, feedback, and queries communicated through speech or writing. NLG involves the production of phrases, sentences and paragraphs with a meaning. It "*happens in four phases: identifying the goals, planning on how goals may be achieved by evaluating the situation and available communicative sources and realizing the plans as a text*".[40]

[37] Manning, C. D., & Schütze, H. (1999)

[38] Bird, S., Klein, E., & Loper, E. (2009)

[39] Khurana, D., Koli, A. C., Khatter, K., & Singh, S. (2022)

[40] Ibid.

## 2.4 Sentiment Analysis

*Sentiment Analysis* (SA), also known as *opinion mining*, is a field of study involving NLP techniques to extract and identify subjective information from textual data.[41] The goal of sentiment analysis usually is to classify the sentiment of a given piece of text as either positive, negative, or neutral.[42] [43] For example:

- "*I love Pfizer*" → Positive
- "*I'm not sure if I like Pfizer*" → Neutral
- "*Pfizer is awful*" → Negative

Over the years, numerous researchers have studied and contributed to the development of sentiment analysis.

Pang and Lee were pioneers in SA, helping establish the foundation of sentiment analysis as a distinct study area within natural language processing and computational linguistics. The authors describe various approaches towards sentiment analysis, which "*all share the common theme of mapping a given piece of text, such as a document, paragraph, or sentence, to a label drawn from a pre-specified finite set or to a real number*".[44]

Machine learning techniques have been extensively utilised in sentiment analysis. Turney proposed a method called "*thumbs up or thumbs down"* that uses unsupervised learning to classify movie reviews as positive or negative. The sentiment of a review can be determined by calculating the average sentiment of the phrases within it that contain adjectives or adverbs.[45]

Nasukawa and Yi present a sentiment analysis method that focuses on extracting sentiments related to positive or negative polarities for particular subjects within a document rather than just classifying the overall sentiment of the entire document.[46]

[41] Saad, S., & Saberi, B. (2017)

[42] Wankhade, M., Rao, A. C. S., & Kulkarni, C. A. (2022)

[43] Sentiment Analysis is applied in Chapter 7 *Explore – Sentiment Analysis*.

[44] Pang, B., & Lee, L. (2008)

[45] Turney, P. D. (2002)

[46] Nasukawa, T., & Yi, J. (2003)

Esuli and Sebastiani introduced *SentiWordNet*, a tool that assigns three numerical scores to each WORDNET synset, indicating the degree of objectivity, positivity, and negativity associated with the terms in the synset. The scores are generated by combining the outputs of eight classifiers.[47]

Recently Hutto and Gilbert proposed the *VADER* (*Valence Aware Dictionary and sEntiment Reasoner*) lexicon, "*a gold-standard sentiment lexicon that is especially attuned to microblog-like contexts*". The authors conducted a study to evaluate the performance of VADER compared to human raters in categorising the sentiment of tweets. According to the authors, VADER was found to perform better than individual human raters at correctly categorising the sentiment of tweets as positive, neutral, or negative.[48]

## 2.5 Twitter

*Twitter*[49] is a microblogging social networking platform where users can post and interact with short messages, known as "*tweets*".[50] It was launched in March 2006, and since then has grown to 368.4 million active users worldwide by December 2022, who post around 500 million messages daily.[51] [52] On Twitter, users can share short messages, images, videos and links, and with them interact with other users around the globe.

Twitter allows users to follow other users and view their tweets chronologically. Users can also engage with tweets by liking, retweeting, and replying to them. The following are the key features around which Twitter operates, as defined by Twitter:

[47] Esuli, A., & Sebastiani, F. (2006)

[48] Hutto, C. J., & Gilbert, E. (2014)

[49] As of July 2023, Twitter is undergoing rebranding, changing its name to X. However, for consistency and clarity, the nomenclature used during the studied period (*Twitter*, *tweet*, *retweet*, etc.) will be retained in this document. For additional information, refer to McCallum, S. (2023).

[50] Twitter Developer Platform (2023)

[51] Statista (2022a)

[52] Sayce, D. (2022)

- *Tweet*: A tweet is any message posted on Twitter. It may have up to 280 characters and can be related to any topic. Also, it can include not only text but also links, photos, GIFs, and videos.

  Tweets can be either organic (new authentic publications made on Twitter and not based on any previous publication) or non-organic (replies to other tweets – tweets sent as an answer to another tweet published previously).[53]
- *User*: A user is someone who has registered to use Twitter, so can post and read tweets.
- *Username*: Each user has a unique username, which is the way he is identified on Twitter.
- *Follower*: A follower is someone who has chosen to subscribe to another user's Twitter update. When one user follows another one, every time the other user posts something it will appear on the former's Twitter home page.
- *Mention*: A mention in a tweet is a direct reference to another user. To include a mention a tweet must have the @ symbol followed by the other user's username (@username).
- *Reply*: A reply is a response to another user's tweet.
- *Retweet*: A retweet is a tweet that someone has forwarded to his followers.
- *Like*: A like is the result of tapping on the heart icon included in a tweet. It is a way of expressing appreciation for a tweet.
- *Hashtag*: A hashtag is any word or phrase immediately preceded by the # symbol. It is used in tweets to refer to a specific topic being commented on in the tweet.[54] [55]

Figure 1 shows an example of a tweet. Posted by *@Twitter*[56] (official Twitter account), it includes a hashtag (#*Tweetups*), a link (*sharedstudios.com/tweetups*) and four images.[57]

[53] This will be used in Section 6.1.3 *Tweet Types*.

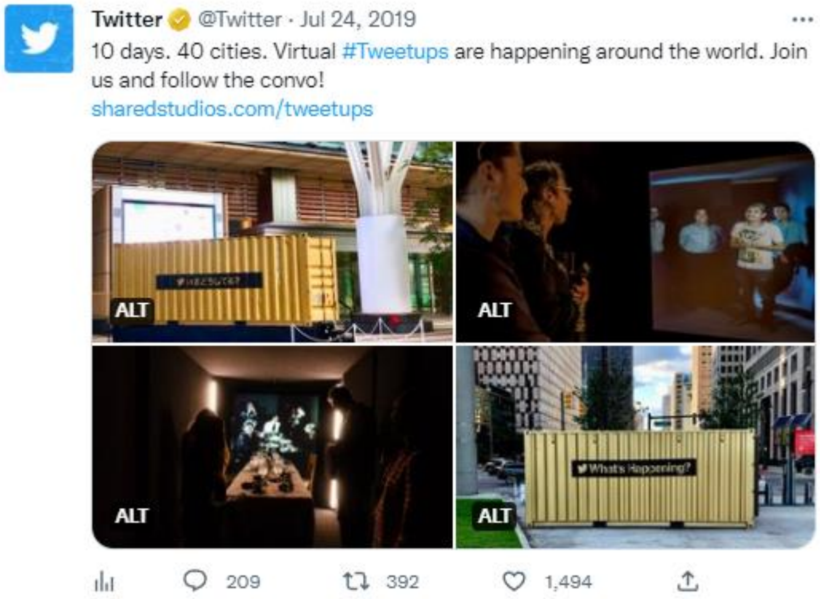
[54] Twitter (2023a)

[55] Twitter (2023b)

[56] https://twitter.com/Twitter

[57] https://twitter.com/Twitter/status/1154172324599537665

**Figure 1**
*Sample tweet from @Twitter*



Twitter @Twitter · Jul 24, 2019

10 days. 40 cities. Virtual #Tweetups are happening around the world. Join us and follow the convo!
sharedstudios.com/tweetups

Also, the previous tweet got 209 replies, was mentioned 392 times, and was liked 1,494 times.

Twitter has become a popular platform for news and information sharing, with many individuals and organisations using it to disseminate information quickly and widely. Due to its real-time and public nature, it has also become a popular platform for sentiment analysis, which involves analysing the opinions and emotions expressed in tweets about a particular topic or entity. Sentiment analysis on Twitter can be used for a variety of purposes, such as understanding public opinion on a product, brand, or political candidate, and monitoring customer satisfaction or brand reputation.

## 2.6 Twitter Sentiment Analysis

Microblogging has emerged as a widely used communication tool, with millions of users expressing their opinions on various topics daily.[58] So, with social media platforms like Twitter becoming increasingly popular, so has the use of sentiment analysis to extract insights from the vast amounts of data generated by these platforms.[59]

[58] Pak, A., & Paroubek, P. (2010)

[59] Statista (2022b)

*Twitter Sentiment Analysis* (TSA) is a subfield of Sentiment Analysis that involves analysing the sentiment of tweets to understand public opinion on a particular topic.[60] Over the years, researchers have explored different techniques and methods to improve sentiment classification accuracy. This literature review summarises the findings of several studies related to TSA.

Zimbra et al. provide an overview of the state-of-the-art techniques for TSA. They discuss the challenges of sentiment analysis on Twitter due to the brevity of tweets and novel language, sentiment class imbalance and poor sentiment recall, and stream-based tweet generation and temporal dependency. The article provides a taxonomy of the techniques developed to address these challenges, including sentiment information propagation, feature representation expansion, Twitter-specific pre-processing, Twitter-specific features, training set expansion, multiple classifier methods, sentiment-topic model techniques, and stream-based classifiers. The article also provides a list of representative studies for each technique.[61] [62]

According to Giachanou and Crestani, most TSA methods use a machine-learning technique called *classifier*. The TSA process involves collecting and labelling tweets for training data, selecting features to train the classifier, building the classifier model, and finally, using the classifier to assign labels to unlabelled tweets. The correctness of the annotations determines the classifier's performance, and evaluation metrics are used to measure performance. Collecting and labelling tweets can be challenging, and selecting features can impact classifier performance.[63]

Kouloumpis et al. look into using language features to identify the sentiment of tweets. The authors assess how adequate existing language resources are and also consider features that capture the unique language style of microblogging. They use a supervised learning method and use hashtags in Twitter data to create a training dataset. Part-of-

[60] Wang, Y., Guo, J., Yuan, C., & Li, B. (2022)

[61] Zimbra, D., Abbasi, A., Zeng, D., & Chen, H. (2018)

[62] In the present document *multiple classifier methods* are applied. This is presented in Chapter 7 *Explore – Sentiment Analysis*, where four different sentiment indexes are created with unrelated lexicons or methods, and then contrasted.

[63] Giachanou, A., & Crestani, F. (2016)

speech features were found to be valuable, while microblogging features such as intensifiers and emoticons were most effective. Using hashtags and emoticons as training data was useful, but their effectiveness may depend on the type of features used. The study suggests that the use of emoticon training data is less effective when microblogging features are included.[64]

Moreover, Mittal and Goel examined the relationship between public mood on Twitter and the values of the *Dow Jones Industrial Average* (DJIA). Their results indicate that natural language processing techniques can be used to capture public mood from Twitter feeds, and that calmness and happiness are causative factors of DJIA values 3-4 days later. However, the study notes that the dataset only considers English-speaking Twitter users and does not necessarily represent genuine public sentiment or direct investment decisions.[65] [66]

As explained by Giachanou and Crestani, TSA approaches can be broadly categorised into four types:[67]

- *Machine Learning*: This approach uses machine learning algorithms to classify the sentiment of a tweet. The algorithm is trained on labelled data, where each tweet is labelled as positive, negative, or neutral.[68] The features used in this approach can be either traditional NLP features (e.g., bag of words, part-of-speech tags, etc.) or domain-specific features (e.g., emoticons, hashtags, etc.). Some popular machine learning algorithms used for TSA are *Support Vector Machines* (SVMs), *Naïve Bayes*, and *Maximum Entropy*.[69]
- *Lexicon-Based*: This approach uses sentiment lexicons (also known as opinion lexicons), which are dictionaries containing words or phrases and their corresponding sentiment scores. The sentiment score can be either binary

[64] Kouloumpis, E., Wilson, T., & Moore, J. D. (2011)

[65] Mittal, A. & Goel, A. (2020)

[66] A similar approach is employed in this document, as only tweets in English were used for sentiment analysis. This is developed in Chapter 7 *Explore – Sentiment Analysis*.

[67] Giachanou, A., & Crestani, F. (2016)

[68] Pang, B., & Lee, L. (2008)

[69] Bhavitha, B. K., Rodrigues, A. P., & Chiplunkar, N. N. (2017)

(positive or negative) or continuous (ranging from very negative to very positive). The sentiment of a tweet, which can be positive, negative, or neutral, is computed by aggregating the sentiment scores of its words or phrases. Some popular sentiment lexicons used in this approach are *SentiWordNet*, *Opinion Lexicon*, and *AFINN*.[70] [71]

- *Hybrid* (Machine Learning & Lexicon-Based): This approach combines machine learning and lexicon-based methods. It uses machine learning algorithms to classify the sentiment of a tweet and then uses sentiment lexicons to refine the classification. For example, if the machine learning algorithm classifies a tweet as neutral, the sentiment lexicon can be used to further determine if the tweet has any positive or negative sentiment.[72]
- *Graph-Based*: This class of approaches uses graph-based algorithms to model the relationships between words in tweets and to classify their sentiment. These approaches represent tweets as graphs, where nodes correspond to words and edges correspond to the relationships between words. Graph-based algorithms are then used to classify the sentiment of tweets based on the graph's topology.[73]

## 2.7 Correlation Analysis

*Correlation analysis* is a statistical technique that explores the relationship between different numerical variables, helping understand how changes in one variable correspond to changes in another.[74] This literature review aims to provide an overview of correlation analysis and its various measures.

- *Pearson's Correlation Coefficient* assesses the linear relationship between two continuous variables.[75] It is one of the most widely used correlation measures and

[70] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011)

[71] In the present project a lexicon-based approach was employed for most of the sentiment analysis (see Chapter 7 *Explore – Sentiment Analysis*).

[72] Ahmad, M., Aftab, S., Ali, I., & Hameed, N. (2017)

[73] Aisopos, F., Papadakis, G., & Varvarigou, T. (2011)

[74] Gogtay, N., & Thatte, U. (2017)

[75] Pearson, K. (1909)

has been extensively studied and applied in various fields.[76] [77] This correlation coefficient assumes a linear relationship between the variables, meaning that they tend to change together consistently, so it is a measure that summarises the strength and direction of the relationship between the variables. Its values range from -1 to +1:

- Positive values indicate a direct correlation, meaning that both variables tend to increase or decrease together. A correlation coefficient of 1 corresponds to a perfect positive correlation, which means that the variables have a strong linear relationship, so when one variable increases, the other increases proportionally, moving in perfect synchronisation.
- Zero values indicate no linear relationship between the variables, meaning that changes in one variable do not correspond to changes in the other.
- Negative values indicate an inverse correlation, implying that as one variable rises, the other tends to decline. A correlation coefficient of -1 signifies a perfect negative correlation. This means the variables have a strong inverse relationship, so when one variable increases, the other decreases proportionally. A value of -1 suggests that the variables move in perfect opposition.[78]

Figure 2 graphically shows how three different correlation values look. The first scatter plot has a 0.6 positive correlation between its variables, while the second one has no correlation (0.0) between its variables, and the third one has a negative correlation of -0.6.

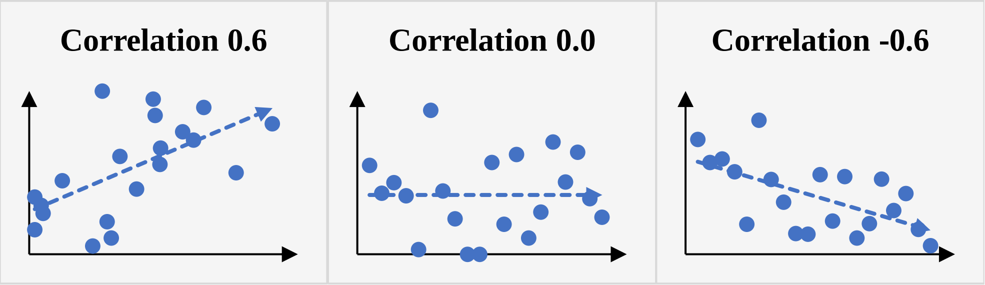
[76] Cohen, J., Cohen, P., West, S.G., & Aiken, L.S. (2002)

[77] In the present investigation this approach will be used in Section 7.5 *Evaluation and Comparison of Sentiment Analysis Methods* and Chapter 8 *Model*.

[78] JMP (2020)

**Figure 2**

*Scatter plots – Pearson's Correlation*



- *Spearman's Rank Correlation Coefficient* is a measure of correlation that assesses the *monotonic* relationship between variables.[79] [80] It is particularly useful when the relationship between variables is not linear or when the data are ranked or ordinal. It has been applied in studies involving ordinal or ranked data, such as surveys, ranking systems, and preference studies.[81]
- *Kendall's Rank Correlation Coefficient* is another measure of correlation that assesses the strength and direction of the relationship between variables.[82] It is based on the concordance and discordance of ranks between paired observations. Kendall's coefficient is applied for testing hypotheses of independence of random variables.[83]
- *Point-Biserial Correlation Coefficient* measures the relationship between a dichotomous variable and a continuous one.[84]
- *Phi Coefficient* is a measure of association used for binary variables. It assesses the strength and direction of the relationship between two dichotomous variables.[85]

[79] See *GLOSSARY* for a definition of *monotonic relationship between variables*.

[80] Spearman, C. (1961)

[81] In the present investigation this approach will be used in Section 7.5 *Evaluation and Comparison of Sentiment Analysis Methods* and Chapter 8 *Model*.

[82] Kendall, M. (1938)

[83] Encyclopedia of Mathematics (2011)

[84] Cureton, E. E. (1956)

[85] Ekström, J. (2011)

These different correlation measures offer valuable tools for analysing the relationship between variables in diverse research contexts. By selecting the appropriate measure based on the nature of the variables and the research question, meaningful insights can be gained into the associations and dependencies between variables.

# 3. Methodology and Data

This chapter describes and supports the methodology and data used in the research process. The chapter begins by describing the investigation type, and explaining the chosen methodology and its rationale. To continue, an assessment of the ethical considerations is included. Finally, the tools and packages utilised throughout the study are introduced.

By providing a detailed explanation of the research methods, ethical considerations, and tools used, this chapter aims to ensure the validity and reliability of the research findings.

## 3.1 Type of Investigation

This investigation is an *empirical research project* that uses web scraping of tweets available online and publicly available data on Pfizer stock.[86] The research design involves the manipulation of independent variables related to the data extracted through web scraping to observe its effect on dependent variables related to the evolution of the Pfizer stock, with all available data used for analysis.

Moreover, this is a *quantitative research study*, which includes descriptive analysis, correlation analysis, and regression analysis techniques.[87]

## 3.2 Methodology Employed: OSEMN

For this investigation, the OSEMN framework will be followed, a methodology for data analysis that provides a structured approach to working with data. It was introduced by Mason and Wiggins as a framework for Data Science projects and has been widely adopted in the field.[88]

> *I'm sure this seems completely obvious to everybody in this room, that you get some data, you clean it up, you look at it, you interpret it, model it, and then you visualise it, or communicate it in some way. That's what it was. But in 2010, that was not obvious, and we wrote it down* (…) *and said, "We're going to say, 'This is the process. This is what you do when*

[86] Bouchrika, I (2023)

[87] Sukamolson, S. (2007)

[88] Brandt, P. (2016)

> *you are data science-ing.'" And it is really funny to talk about it here today, because now you look at it and you think, "Oh, this is so obvious. Like, of course." But it wasn't obvious in 2010.*
>
> *Hilary Mason*[89]

The OSEMN framework provides a methodical procedure for data analysis that can help ensure that all relevant factors are taken into account and that the analysis is comprehensive while being a flexible and adaptable methodology, which allows for customisation to fit the project's specific needs.[90]

It is essential to acknowledge that the OSEMN methodology is not inherently superior or inferior to other approaches, as each method has strengths and limitations. However, its widespread adoption and demonstrated effectiveness in numerous Data Science projects establish it as a reliable and credible approach to data analysis. By following OSEMN, researchers can achieve a comprehensive understanding of the research question by thoroughly analysing all pertinent aspects of the data.[91]

The OSEMN methodology includes five steps: *Obtain*, *Scrub*, *Explore*, *Model*, and *iNterpret*.[92]

1) *Obtain*: Data are collected from various sources such as databases, APIs, or web scraping.
2) *Scrub*: Data are cleaned and pre-processed, including dealing with missing values, handling outliers, and transforming the data into a suitable format for analysis.
3) *Explore*: Data are visualised and analysed to gain insights and identify patterns or relationships.
4) *Model*: Statistical or machine learning models are built to make predictions or classify data.
5) *iNterpret*: Results of the analysis are communicated to stakeholders clearly and concisely, often through data visualisation or storytelling techniques.

Figure 3 graphically illustrates the previously described steps.

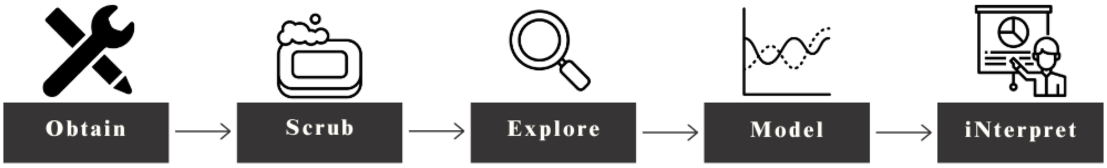
[89] Mason, H. (2022)

[90] Lau, C. H. (2019)

[91] Lao, R. (2017)

[92] Mason, H., & Wiggins, C. (2010)

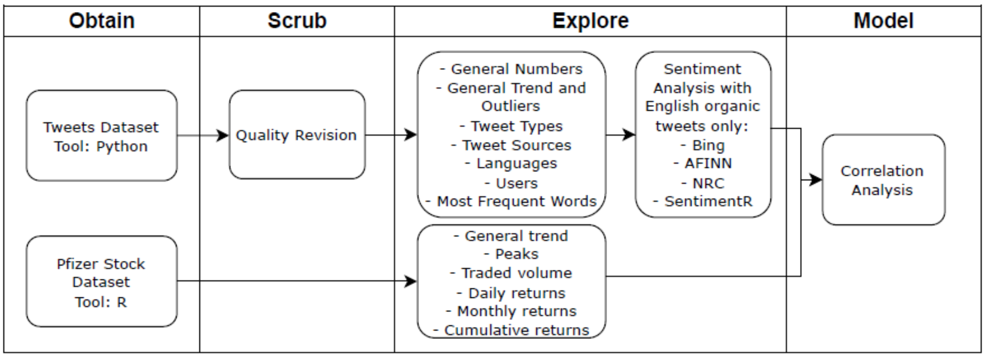
**Figure 3**
*Data Science OSEMN methodology*



This sequence is not meant to be a strict linear process but rather a flexible framework that can be adapted to different DS projects, including the one developed in this document.

Figure 4 illustrates the initial four stages of OSEMN as implemented in the project presented in this document. It is worth noting that the interpret stage, which is a more comprehensive phase, is not included in the diagram due to its broader scope, as it involves making sense of the data and extracting insights from it rather than just processing it.

**Figure 4**
*Applied OSEMN methodology*



## 3.3 Ethical Considerations

Ethical considerations were taken into account during the data collection and analysis stages, mainly when extracting the Tweets Dataset.

### 3.3.1 Tweets Dataset Extraction

Several ethical aspects need to be considered when collecting information from tweets:

- *Informed consent*: It is impossible to obtain informed consent from Twitter users, as tweets are publicly available and users do not expect privacy when posting on the platform. However, in this investigation transparency is being shown related to how the data are being used and credit to the original authors of the tweets is given whenever it corresponds.[93]

- *Anonymity*: While tweets are publicly available, protecting the anonymity of the users who posted them is crucial. No private information about users was collected, as only what is publicly shared by each user forms part of the downloaded database.[94]
- *Twitter Terms of Service*: These have been considered for this investigation. They mention, "*This license authorizes us to make your Content available to the rest of the world and to let others do the same*".[95]
- *Uruguayan Law*: In Uruguay, where this investigation was carried out, there are no specific laws or regulations related to the use of social media data for research purposes. The Laws and Regulations related to personal data do not include information from Twitter, as it is not related to specific or identifiable persons.[96] [97]

### 3.3.2 Pfizer Stock Dataset Extraction

Pfizer stock data has been extracted from Yahoo Finance[98] and is publicly available data regarding a publicly traded company, Pfizer, listed on the New York Stock Exchange (NYSE) under the ticker symbol PFE. Then, the Yahoo Terms of Service were considered for the use of this data.[99] [100]

[93] For examples of how credit to the original authors of tweets is given, see Section 6.1.2 *General Trends and Outliers*.

[94] To see what information about each tweet and its author was downloaded, see Appendix A – *Tweets' Download*.

[95] Twitter (2023c)

[96] IMPO (2008)

[97] Uruguay Presidencia & Agesic (n.d.)

[98] https://finance.yahoo.com/

[99] Yahoo (2023)

[100] NYSE (2022)

## 3.4 Tools and Packages Used

This project was carried out using both *Python* and *R* programming languages. The first part (collection of tweets) was done with *Python*, while the second part (analysis) was made with *R*.[101] [102]

### 3.4.1 Python

This section introduces the programming language Python and its packages used to carry out the present investigation.

> *"Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.* (…) *Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed".*[103]

In this project Python was used for the web scraping of all the tweets database. The following packages were used:[104]

- *datetime*[105] provides classes for working with dates and times. In this project its *timedelta* function was used in the scraping loop to add one day to the date for each iteration.
- *os*[106] supplies a portable way of employing functionalities that depend on the operating system. In this project it was used to save the tweets into a JSON stored in the working directory.
- *pandas*[107] is a library that offers adaptable, rapid, and expressive data structures intended to simplify and make working with "relational" or "labelled" data

[101] https://www.python.org/

[102] https://www.r-project.org/

[103] Python.org (2023)

[104] See Appendix A – *Tweets' Download*.

[105] https://docs.python.org/es/3/library/datetime.html

[106] https://docs.python.org/3/library/os.html

[107] https://pandas.pydata.org/

straightforward and intuitive. In this project it was used various times: to create a date range, to transform a date into a date-time format and to read the JSON stored in the working directory.

- *snscrape*[108] is a scraper for social networking services (SNS).[109] It gathers items, such as relevant posts, through processes like scraping user profiles, hashtags, or search queries. In this project it was used to scrape all the tweets which fulfil chosen filters (tweet count, text query, since date and until date) and their related information.

### 3.4.2 R

In this project the programming language R was used for part of the data extraction process, and the whole exploration and modelling stages. So, in this section it is introduced, along with its packages that were used to carry out the present investigation.

> *"R is a language and environment for statistical computing and graphics. (…) R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible".*[110]

The following packages were used:[111]

- *dplyr*[112] is employed for manipulating data, offering a uniform set of actions that address typical data manipulation tasks (select, mutate, filter, summarise, group_by, arrange). Many of these verbs, and others, were used along this project, for instance, to create dataframes which group the information contained in another one by the category present in one of its columns (*group_by*).
- *ggplot2*[113] is used for generating statistical, or data, graphics. Unlike most graphic packages, *ggplot2* possesses a foundational structure rooted in the *Grammar of*

[108] https://github.com/JustAnotherArchivist/snscrape

[109] Saleem, A. (2023)

[110] R (2022)

[111] See Appendix B – *R Setup*.

[112] https://dplyr.tidyverse.org/

[113] https://ggplot2-book.org/introduction.html

*Graphics*, enabling the creation of graphs through the integration of distinct elements. In this project most graphs were created using this package.[114]

- *ggpubr*[115] offers user-friendly functions for tailoring publication-ready plots based on *ggplot2*. In this project this package was used to present *ggplot2* graphs together in the same image.
- *ggrepel*[116] supplies text and label geoms for *ggplot2* to prevent text label overlap. Labels repel away from each other and from data points. In this project this package was used so that the labels present in the *ggplot2* graphs do not overlap.
- *lubridate*[117] encompasses functionalities for handlings date-times and time spans. It offers efficient and user-friendly parsing of date-time data, extraction and modification of date-time components, and algebraic operations involving date-time and time-span objects. In this project the *lubridate* package was used to work with dates.
- *psych*[118] serves as a versatile toolkit for personality, psychometric theory, and experimental psychology. In this project the *psych* package was used to visualise correlations between variables.
- *purrr*[119] comprises a comprehensive and cohesive set of functional programming tools for R. In this project this package was used to import the tweets CSVs which are then joined in one large CSV.
- *readr*[120] offers an efficient and user-friendly method to read tabular data from delimited files, like comma-separated values (CSV) and tab-separated values (TSV). In this project this package was used each time a CSV is read.

[114] Wilkinson, L. (2005)

[115] https://cran.r-project.org/web/packages/ggpubr

[116] https://cran.r-project.org/web/packages/ggrepel

[117] https://cran.r-project.org/web/packages/lubridate

[118] https://cran.r-project.org/web/packages/psych

[119] https://cran.r-project.org/web/packages/purrr

[120] https://readr.tidyverse.org/

- *reshape2*[121] allows for versatile data restructuring and aggregation with the utilisation of only two functions. In this project the *melt* function was used to organise the data so it can be exposed correctly by graphs.
- *scales*[122] supplies the underlying scaling infrastructure employed by *ggplot2* and offers mechanisms to override the default breaks, transformations, labels, and palettes. In this project the *scale_x_date* function of the *scales* package was used to make *ggplot2* graphs look clearer.
- *sentimentr*[123] computes sentiment polarity of text at the sentence level and has the option to aggregate by rows or based on grouping variables. In this project the *sentimentr* package was used to conduct a sentiment analysis that analyses whole sentences instead of independent words.
- *stringr*[124] provides an extensive array of functions designed to streamline string manipulation as much as possible. In this project its *str_detect* function was used to remove the term "Pfizer" from the tweets' content to then find the most frequent words in the tweets.
- *textdata*[125] offers a framework for downloading, parsing, and storing text datasets on the disk, and load them when required. Includes various sentiment lexicons and labelled text datasets for classification and analysis. In this project the *textdata* package was used to download the *AFINN lexicon* for sentiment analysis.
- *tibble*[126] offers utilities for working with *tibbles*.[127] In this project the *tibble* package was used to convert the row names into the first column of a dataset.[128]

[121] https://cran.r-project.org/web/packages/reshape2

[122] https://scales.r-lib.org/

[123] https://cran.r-project.org/web/packages/sentimentr

[124] https://stringr.tidyverse.org/

[125] https://cran.r-project.org/web/packages/textdata

[126] https://tibble.tidyverse.org/reference/tibble-package.html

[127] See *GLOSSARY* for a definition of *tibble*.

[128] Tidyverse.org (2023)

- *tidyquant*[129] integrates tools for gathering and examining financial data with the tidy data infrastructure. In this project this package was used to download the Pfizer stock data from Yahoo Finance.
- *tidyr*[130] aims to enable the creation of *tidy data*.[131] In this project this package was used to create dummies for the weekdays via the *spread* function.
- *tidytext*[132] offers functions and supporting datasets for converting text to and from tidy formats facilitating a seamless transition between tidy tools and existing text mining packages. In this project this package was used to remove *stopwords*[133] from tweets and also for certain functionalities used in the sentiment analysis.[134] [135]

[129] https://cran.r-project.org/web/packages/tidyquant

[130] https://tidyr.tidyverse.org/

[131] See *GLOSSARY* for a definition of *tidy data*.

[132] https://cran.r-project.org/web/packages/tidytext/vignettes/tidytext.html

[133] See *GLOSSARY* for a definition of *stopword*.

[134] https://cran.r-project.org/web/packages/tidytext/vignettes/tidytext.html

[135] Wickham, H. (2014)

# 4. Obtain

To begin, data on tweets and the Pfizer stock was downloaded, covering the period from January 2020 to December 2022. The tweets data was acquired using the *snscrape* package in Python, while the Pfizer stock data was obtained from Yahoo Finance using the *tidyquant* package in R.[136] In this chapter this data extraction process is described in detail.

## 4.1 Tweets Dataset Extraction

The complete set of tweets dated between January 1st, 2020, and December 31st, 2022, containing the term "Pfizer" (case-insensitive), was extracted using Python. The script retrieved the tweet content and supplementary details, compiling them into a CSV file with each tweet represented as a distinct entry.[137] In addition to the tweet's text, the following fields were downloaded: URL, date and time, tweet id, user, reply count, retweet count, like count, quote count, conversation id, language, source, source URL, source label, outlinks, retweeted tweet, quoted tweet, in reply to tweet, in reply to user, mentioned users and hashtags.

Various difficulties were faced during this process. Firstly, though until early November 2020, the daily number of tweets was on average smaller than 320, meaning the complete download of all these days was quick and presented no inconveniences, on November 9th of the mentioned year, Pfizer and BioNTech announced their vaccine was successful in first phase 3 analysis. This announcement meant that on that day more than 180,000 tweets were published, making the average amount of tweets on the subsequent week larger than 59,000 – many orders of magnitude larger than before the announcement.

The workaround found so that all the tweets could be downloaded was to download one day at a time and save each in an independent CSV. This meant that if issues during the process were faced (i.e., lost internet connection), only the day running at the moment of the issue would be affected. However, November 9th, 2020's CSV file has a record size

[136] DD (2018)

[137] See Appendix A – *Tweets' Download*.

of 520MB. Other issues were identified in the quality assurance step, and are described in detail in Chapter 5 *Scrub*.

## 4.2 Pfizer Stock Dataset Extraction

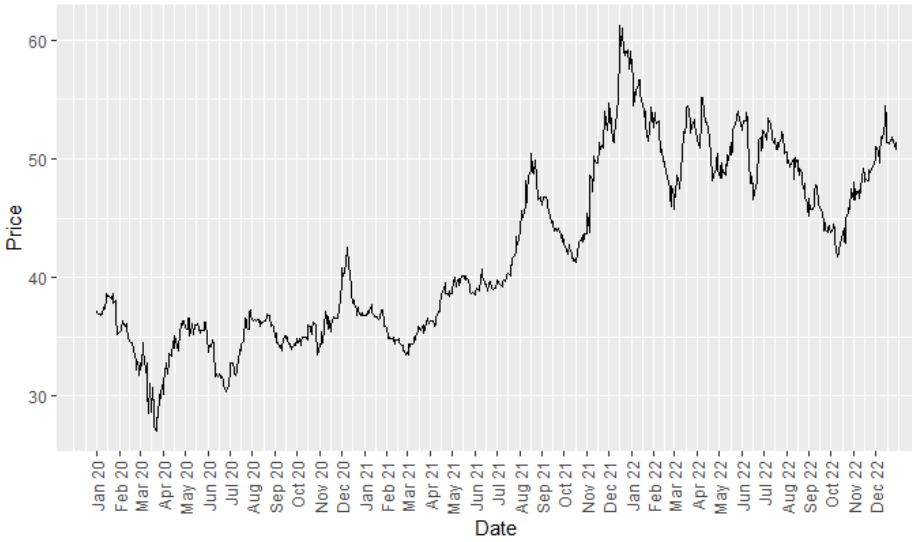
All the data about the PFE stock was downloaded by using tools included in the R programming language. The download includes, for each business day, the opening and closing price, the adjusted closing price, the highest and lowest price, and the traded volume.[138] As stock markets only work on business days, there is no trading and no information for non-business days. This shall be considered in later steps when carrying out the analysis. The tail of the downloaded dataset is presented in Table 1.

**Table 1**
*PFE Stock Database*

| | PFE.Open | PFE.High | PFE.Low | PFE.Close | PFE.Volume | PFE.Adjusted |
|---|---|---|---|---|---|---|
| 23/12/2022 | 51.56 | 51.95 | 51.24 | 51.83 | 10,666,500 | 51.3585 |
| 27/12/2022 | 51.86 | 51.93 | 51.05 | 51.13 | 12,033,800 | 50.6649 |
| 28/12/2022 | 51.05 | 51.39 | 50.75 | 50.80 | 10,053,900 | 50.3379 |
| 29/12/2022 | 51.02 | 51.67 | 50.99 | 51.33 | 8,971,300 | 50.8630 |
| 30/12/2022 | 51.29 | 51.40 | 50.75 | 51.24 | 11,394,800 | 50.7739 |

The downloaded series of prices is presented in Figure 5.

**Figure 5**
*Pfizer stock price series 2020-2022*



[138] See Appendix C – *Pfizer Stock Dataset Download*.

# 5. Scrub

In this stage, the data was prepared for analysis by performing a quality revision and correcting any errors that may have occurred in the previous stage. The objective was to have the data ready for exploration and analysis.

Once all the CSV files with tweets were ready, they were imported into the R environment.[139] While saving one day's worth of tweets in separate CSV files was advantageous in minimising the potential impact of issues such as lost internet connections during the downloading process, it did introduce its own challenges. Detecting errors in these circumstances proved to be demanding. Occasionally, some CSV files had a tiny size of 1KB, which made the error evident, allowing for swift remediation by re-downloading the affected day's tweets.

However, there were instances where the issue wasn't as obvious. At times, only a portion of a day's downloads was affected, leaving some hours incomplete while others remained intact. To address this issue comprehensively and ensure the integrity of the downloaded data, a quality assurance procedure was employed after importing all the tweets into the R environment. This process involved grouping tweets by date and hour, revealing that several dates exhibited stretches of subsequent hours with zero tweets. This anomaly hinted at incomplete downloads during those specific hours, necessitating further investigation and action.

The analysis revealed that forty-nine days of data were affected by the missing-hour issue, and as a remedy, these specific days had to be re-downloaded to rectify the gaps in the dataset, emphasising the importance of robust quality assurance procedures in handling and validating large-scale data acquisitions.[140]

Following the identification of dates with missing tweets, a targeted approach was employed to rectify the situation. Specifically, the Python code for tweets download, as outlined in Section 4.1 *Tweets Dataset Extraction*, was rerun exclusively for the dates requiring correction. This strategy allowed for a more efficient resolution of the issue by

[139] See Appendix D – *Tweets Dataset Importation*.

[140] See Appendix E – *Tweets Dataset Quality Revision*.

focusing solely on the affected time periods. Subsequently, the revised datasets were imported into the R environment, incorporating the re-downloaded tweets. Finally, the comprehensive quality revision process was performed again to ensure data accuracy and completeness. This iterative approach not only addressed the missing-hour issue but also reinforced the credibility of the dataset by verifying that all necessary data points were successfully acquired and integrated.

# 6. Explore – Descriptive Analysis

The analysis commenced with an exploratory phase, where the tweeting patterns and trends observed throughout the examined period were studied, and then went into a purely quantitative part, where volumes and variations were contrasted. The characteristics of the data were thoroughly examined, laying the foundation for subsequent analyses. In this chapter, the descriptive analysis of the Tweets Dataset and the Pfizer Stock Dataset is presented.

## 6.1 Tweets Dataset Descriptive Analysis

First, after checking the dataset for missing data (as explained in the previous chapter), an exploratory analysis was conducted to identify the dataset's key characteristics, including tweet distribution, trends in tweet volume, popular languages, outliers, most frequent words, and influential users. In the following sections the main findings are presented.[141]

### 6.1.1 General Numbers

The total amount of tweets including the term "Pfizer" published between January 1st, 2020, and December 31st, 2022, and available on Twitter on December 31st, 2022, is 13,077,597.[142] This number was then analysed further and that is included in the next sections of this document.

### 6.1.2 General Trends and Outliers

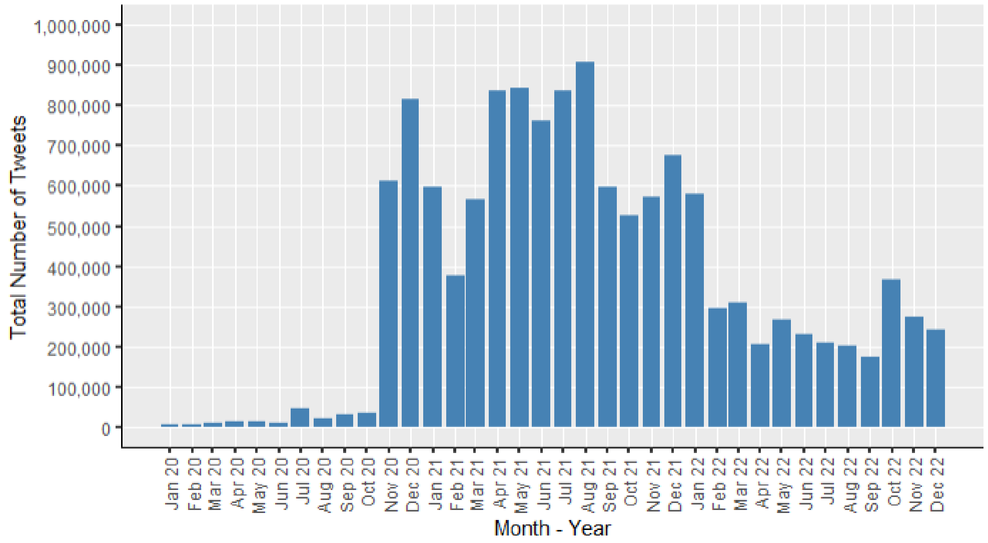
In this stage the goal was to understand how the Twitter world behaved in big numbers, in relation to the key term "Pfizer". To start with: was the tweeting trend stable? To answer this question data was first grouped by month and year, and the total number of tweets in each period was calculated.

[141] This section answers the first question presented in Section 1.4 *Research Questions*: *What trends can be identified in the number and contents of tweets containing the word Pfizer between 2020 and 2022, primary years of the COVID 19 pandemic?*

[142] See Appendix F – *Tweets Dataset: General Numbers Analysis*.

**Figure 6**

*Tweets per month with the term "Pfizer"*



As Figure 6 shows, in early 2020 the number of tweets per month related to Pfizer was really low, not even surpassing the 50,000 mark. However, in November 2020 the number boomed, maintaining a growing trend until mid-2021, when the number of tweets started getting smaller, though always considerably more significant than the numbers of early 2020.

Why did the number of tweets suddenly boom? To answer this question an *outlier analysis* was carried out.[143] [144]

The chosen method for outlier identification involves treating all days equally in the selected period without restrictions. This approach aims to identify the main outstanding points in the data by capturing significant deviations or exceptional occurrences. By avoiding smoothing techniques like moving averages, the analysis becomes more precise and focused, ensuring that important outliers are not masked, leading to a more accurate understanding of the dataset's key features and peculiarities over time.

[143] See *GLOSSARY* for a definition of *outlier*. For additional information, refer to Aggarwal, C. C. (2017).

[144] See Appendix G – *Outlier Identification*.

Then, to find outliers all the values of the number of tweets on a particular day were ordered from smaller to bigger, and the *median* was identified.[145] Then the *first quartile* (Q1) and *third quartile* (Q3) were calculated, creating four groups containing each the same number of values. Having identified Q1 and Q3, the *interquartile range* (IQR) is the distance between those values: Q3 – Q1. Outliers are observations more than 1.5 times IQR bigger than Q3, or more than 1.5 times IQR smaller than Q1.

All the previously explained values are shown in a box plot in Figure 7. A box plot shows a box from Q1 to Q3, showing the median inside it. Then, to both sides goes a line right until -1.5 IQR from Q1 and +1.5 IQR from Q3. All values outside these lower and upper whiskers are considered outliers.

**Figure 7**
*Boxplot – tweets per day 2020-2022*

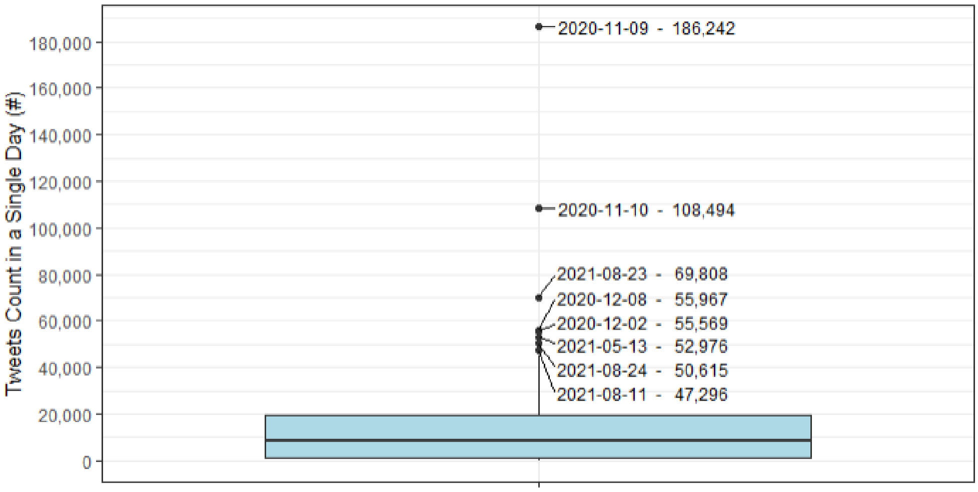


Figure 7 shows a boxplot[146] where each day's number of tweets is a value. The median is marked in 8,536, which corresponds to April 26th, 2022. This value separates the daily tweet count into two groups of the same size: each group has precisely 538 values (537 + the median).

[145] See *GLOSSARY* for a definition of *minimum value, 1st quartile, median, mean, 3rd quartile,* and *maximum value*. For additional information, refer to James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017).

[146] See *GLOSSARY* for a definition of *box plot*. For additional information, refer to Frigge, M., Hoaglin, D. C., & Iglewicz, B. (1989).

Moreover, Q1 is 1,122 (the average between the value in position 269 and the value in position 270) and Q3 is 19,459 (the average between the value in position 806 and the value in position 807). So, 1,122 and 19,459 are the lower and upper limits of the box, respectively, being its difference, 18,337, the IQR. As all values must be positive, the lower limit is marked in 0 and the lower whisker is smaller and almost imperceptible. However, the upper one, which is the one that goes from Q3 to Q3 + 1.5 IQR, is bigger and goes up to number 46,964.5. All the values bigger than that are considered outliers.

Then, as seen in Figure 7, several outliers are on the box plot's top end. These correspond to 8 dates which are shown in Table 2.

**Table 2**
*Outliers*

| Date | Number of Tweets |
|---|---:|
| 09/11/2020 | 186,242 |
| 10/11/2020 | 108,494 |
| 02/12/2020 | 55,569 |
| 08/12/2020 | 55,967 |
| 13/05/2021 | 52,976 |
| 11/08/2021 | 47,296 |
| 23/08/2021 | 69,808 |
| 24/08/2021 | 50,615 |

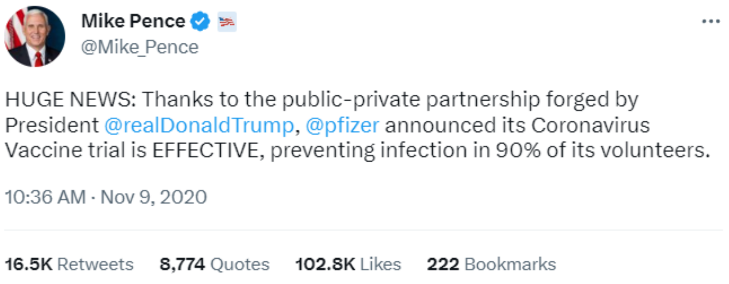
Most of these dates are associated with special events related to Pfizer and the race for the COVID-19 vaccine, as follows, presented in chronological order:

- November 9th, 2020, had the most significant number of tweets in a single day: 186,242. That day, Pfizer and BioNTech announced that their vaccine against COVID-19 had succeeded in the first analysis from Phase 3 Study.[147] That day the most quoted tweet, shown in Figure 8, was one by @Mike_Pence (Mike Pence, 48th Vice President of the United States from 2017 to 2021) announcing Pfizer's effective vaccine trial.[148]

[147] Pfizer (2020)

[148] https://twitter.com/Mike_Pence/status/1325794339335921667

**Figure 8**
*Most quoted tweet – Nov 9, 2020*



Mike Pence
@Mike_Pence

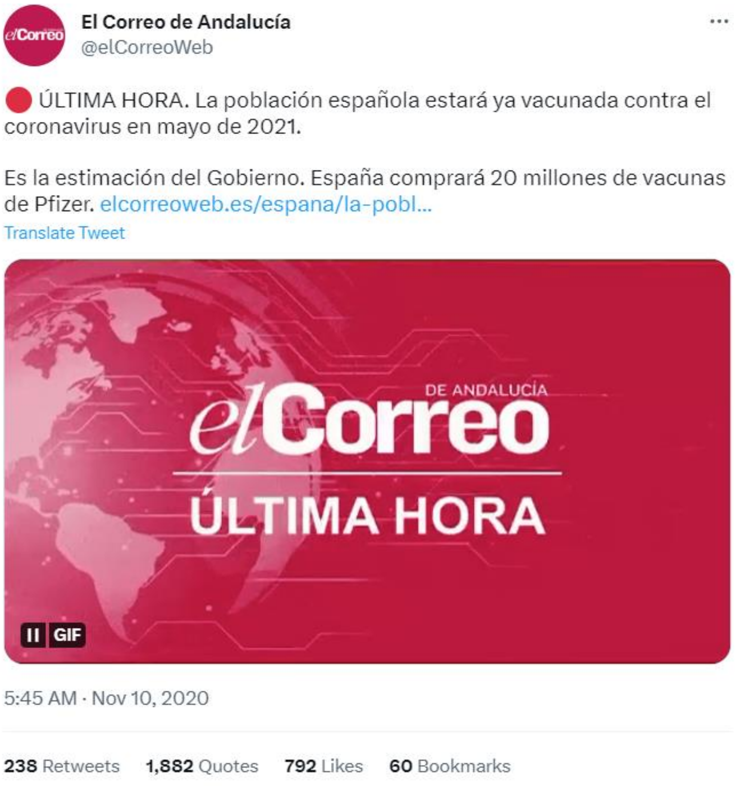
HUGE NEWS: Thanks to the public-private partnership forged by President @realDonaldTrump, @pfizer announced its Coronavirus Vaccine trial is EFFECTIVE, preventing infection in 90% of its volunteers.

10:36 AM · Nov 9, 2020

16.5K Retweets 8,774 Quotes 102.8K Likes 222 Bookmarks

- November 10th, 2020, presented the second largest number of tweets on a single day: 108,494. On that day the most quoted tweet is the one shown in Figure 9 which was tweeted in Spanish by @elCorreoWeb (*El Correo de Andalucía*)[149] and mentions that the Spanish population would be vaccinated against coronavirus by May 2021.[150]

**Figure 9**
*Most quoted tweet – Nov 10, 2020*



El Correo de Andalucía
@elCorreoWeb

🔴 ÚLTIMA HORA. La población española estará ya vacunada contra el coronavirus en mayo de 2021.

Es la estimación del Gobierno. España comprará 20 millones de vacunas de Pfizer. elcorreoweb.es/espana/la-pobl...
Translate Tweet

5:45 AM · Nov 10, 2020

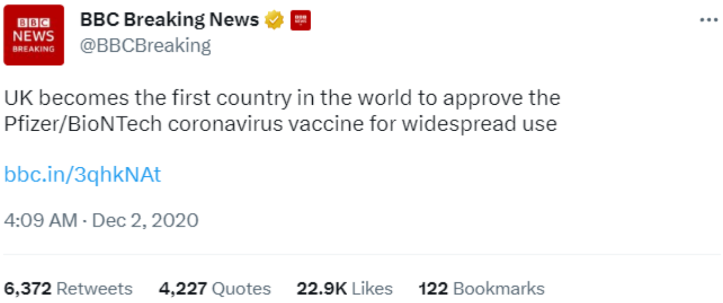
238 Retweets 1,882 Quotes 792 Likes 60 Bookmarks

[149] https://elcorreoweb.es/

[150] https://twitter.com/elCorreoWeb/status/1326083569710718976

- December 2nd, 2020, had 55,569 published tweets. On that day the United Kingdom became the first sovereign state to approve Pfizer and BioNTech's COVID-19 vaccine.[151] This is evidenced by that day's most quoted tweet, shown in Figure 10, which is one tweeted by @BBCBreaking (*BBC Breaking News*),[152] which states that the UK became the first nation to authorise the widespread use of the Pfizer/BioNTech COVID-19 vaccine.[153]

**Figure 10**
*Most quoted tweet – Dec 2, 2020*



BBC Breaking News
@BBCBreaking

UK becomes the first country in the world to approve the Pfizer/BioNTech coronavirus vaccine for widespread use

bbc.in/3qhkNAt

4:09 AM · Dec 2, 2020

**6,372** Retweets **4,227** Quotes **22.9K** Likes **122** Bookmarks

- December 8th, 2020, had 55,967 published tweets. On that day the first person ever received a Pfizer COVID-19 vaccine. It happened in the United Kingdom. This is seen in that day's most quoted tweet, shown in Figure 11. Tweeted by @ABC (*ABC News*),[154] it gives the news of William Shakespeare, who, at 81 years old, received the Pfizer-BioNTech COVID-19 vaccine, becoming the second person to get it.[155]

[151] Ellyatt, H. (2020)

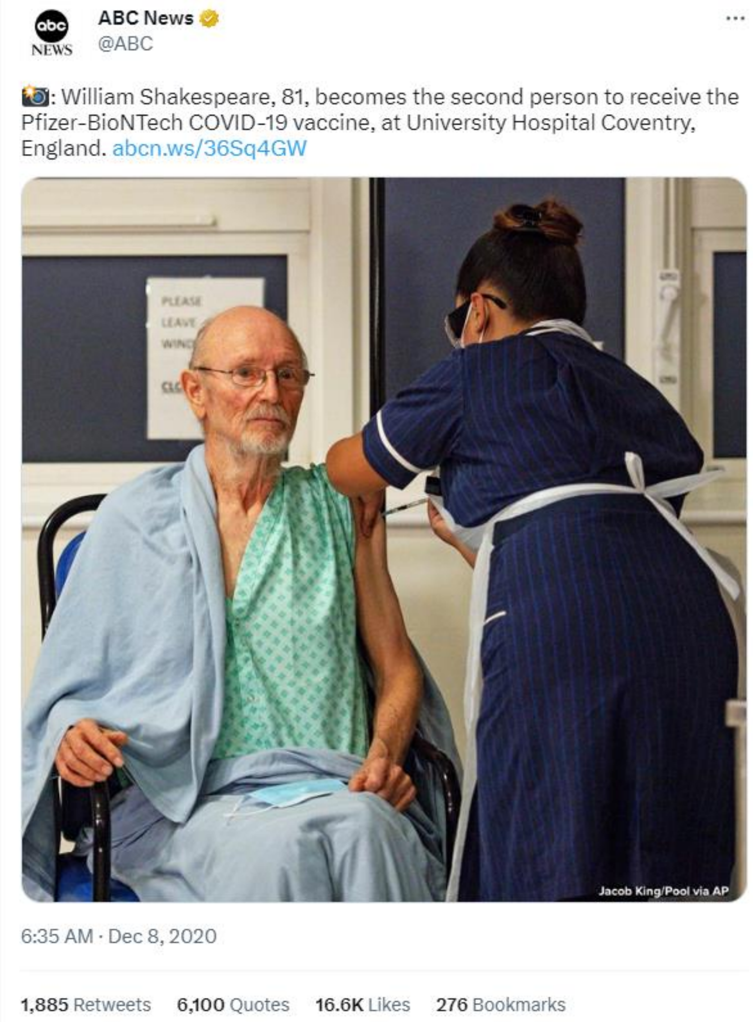
[152] https://www.bbc.com/news/uk

[153] https://twitter.com/BBCBreaking/status/1334032020947734535

[154] https://abcnews.go.com/

[155] https://twitter.com/ABC/status/1336242888167067650

**Figure 11**
*Most quoted tweet – Dec 8, 2020*



ABC News
@ABC

📸: William Shakespeare, 81, becomes the second person to receive the Pfizer-BioNTech COVID-19 vaccine, at University Hospital Coventry, England. abcn.ws/36Sq4GW

6:35 AM · Dec 8, 2020

1,885 Retweets 6,100 Quotes 16.6K Likes 276 Bookmarks

- May 13th, 2021, presented the next outlier: 52,976 tweets. On that day there was a combination of events about which everyone had something to say, being one the most popular: FDA (*U.S. Food and Drug Administration*)[156] approved the use of Pfizer and BioNTech's COVID-19 vaccine for kids from 12 to 15 years old on May 10th.[157]
- August 11th, 2021, had 47,296 tweets in one day. Though this is identified as an outlier, it is the smallest one. Also, there was no special Pfizer event on that day. However, what did occur is that Vladimir Putin (president of Russia) announced

[156] https://www.fda.gov/

[157] Pfizer (2021)

that Russia approved its first vaccine against COVID-19, a direct competitor to Pfizer and BioNTech's vaccine.[158] Also, on that day it was announced that HIV+ volunteers would be included in the final stage of Pfizer vaccine trials.[159]

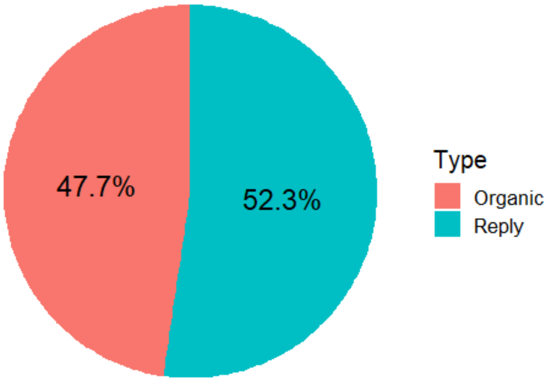
- August 23rd and 24th, 2021, present the last two outliers with 52,751 and 50,615 tweets, respectively. What occurred at that moment, specifically on the first date, is that the FDA approved the first COVID-19 vaccine, the Pfizer-BioNTech one.[160]

### 6.1.3 Tweet Types

As seen in Section 2.5 *Twitter*, tweets can be either organic or non-organic. In the Tweets Dataset, the second type can be identified as there is a column which mentions whether a tweet is a reply to another tweet, and if it was, to which tweet (*tweet id*). Then, all tweets which contain information in that column were classified as non-organic.

From the total 13,077,597 tweets, 6,234,476 are organic.[161] This is 47.7% of the total tweets. The other 6,843,121, 52.3% of the tweets, are replies. This is presented graphically in Figure 12.

**Figure 12**
*Organic tweets and replies (%) – 2020-2022*



[158] BBC News (2020)

[159] Nigam, A. (2020)

[160] U.S. Food and Drug Administration (2021)

[161] See Appendix H – *Organic Tweets and Replies*.

### 6.1.4 Tweet Sources

Tweets can be sent from various sources, with Android app, web app and iPhone app being the most popular ones. In Table 3 the ten most popular tweeting sources are shown, along with an average of how many likes each tweet from that source had. Also, the total for all sources is shown in the last row of the table.[162]

**Table 3**
*Top tweet sources*

| Source | Number of Tweets | Percentage of Total Tweets | Number of Likes in Tweets | Likes Per Tweet |
|---|---:|---:|---:|---:|
| Twitter for Android | 3,986,924 | 30.47% | 36,334,879 | 9.11 |
| Twitter Web App | 3,691,192 | 28.27% | 53,898,157 | 14.6 |
| Twitter for iPhone | 3,607,532 | 27.61% | 74,606,724 | 20.68 |
| Twitter for iPad | 380,374 | 2.86% | 3,201,683 | 8.42 |
| TweetDeck | 246,507 | 1.87% | 6,598,377 | 26.77 |
| WordPress.com | 133,230 | 0.99% | 145,571 | 1.09 |
| dlvr.it | 115,122 | 0.88% | 358,919 | 3.12 |
| Hootsuite Inc. | 82,955 | 0.66% | 1,325,814 | 15.98 |
| IFTTT | 58,664 | 0.44% | 40,496 | 0.69 |
| SocialFlow | 48,017 | 0.33% | 3,084,397 | 64.24 |
| **Total** | **13,077,597** | **100%** | **187,924,424** | **14.37** |

### 6.1.5 Languages

The original dataset includes a language column, and this was utilised to determine the languages of the tweets. For the little more than 13 million tweets, the distribution is as follows:[163]

- 6,530,264 tweets are in English,
- 2,178,435 tweets are in Spanish,
- 1,135,268 tweets are in Portuguese, and
- 937,697 tweets are in French.

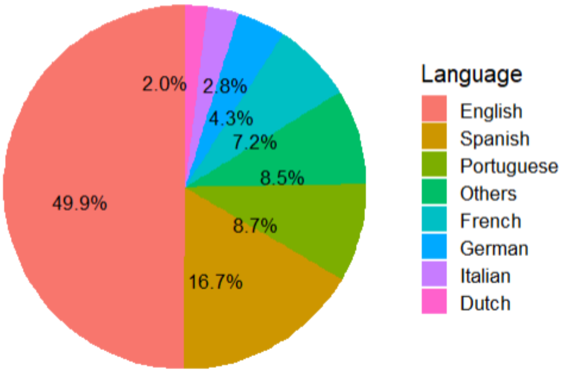
[162] See Appendix I – *Tweets' Source Analysis*.

[163] See Appendix J – *Tweets' Languages*.

- The remaining tweets are written in 66 different languages identified within the dataset.

A visual representation of the language proportions can be found in Figure 13.

**Figure 13**
*Tweets' languages (%) – 2020-2022*



### 6.1.6 Users

The 13,077,597 tweets were tweeted by 3,725,375 different users. So, on average, each of these users posted 3.51 tweets mentioning Pfizer between 2020 and 2022. However, the distribution of tweets per user is different from a regular one. While 2,221,709 users had only one tweet and 602,737 users published only 2, one user (@ScienceCareers)[164] tweeted 11,175 times, and another (@VaccineCa)[165] did so on 17,392 occasions.[166]

Figure 14 presents three histograms illustrating the distribution of the number of users per number of tweets related to Pfizer published between 2020 and 2022. The histogram has been divided into three sections to enhance visual clarity due to the substantial disparity in scales, as the range spans from over 2 million users with a single tweet to fewer than 5,000 users with 21 tweets and only two users with over 11,000 tweets each.

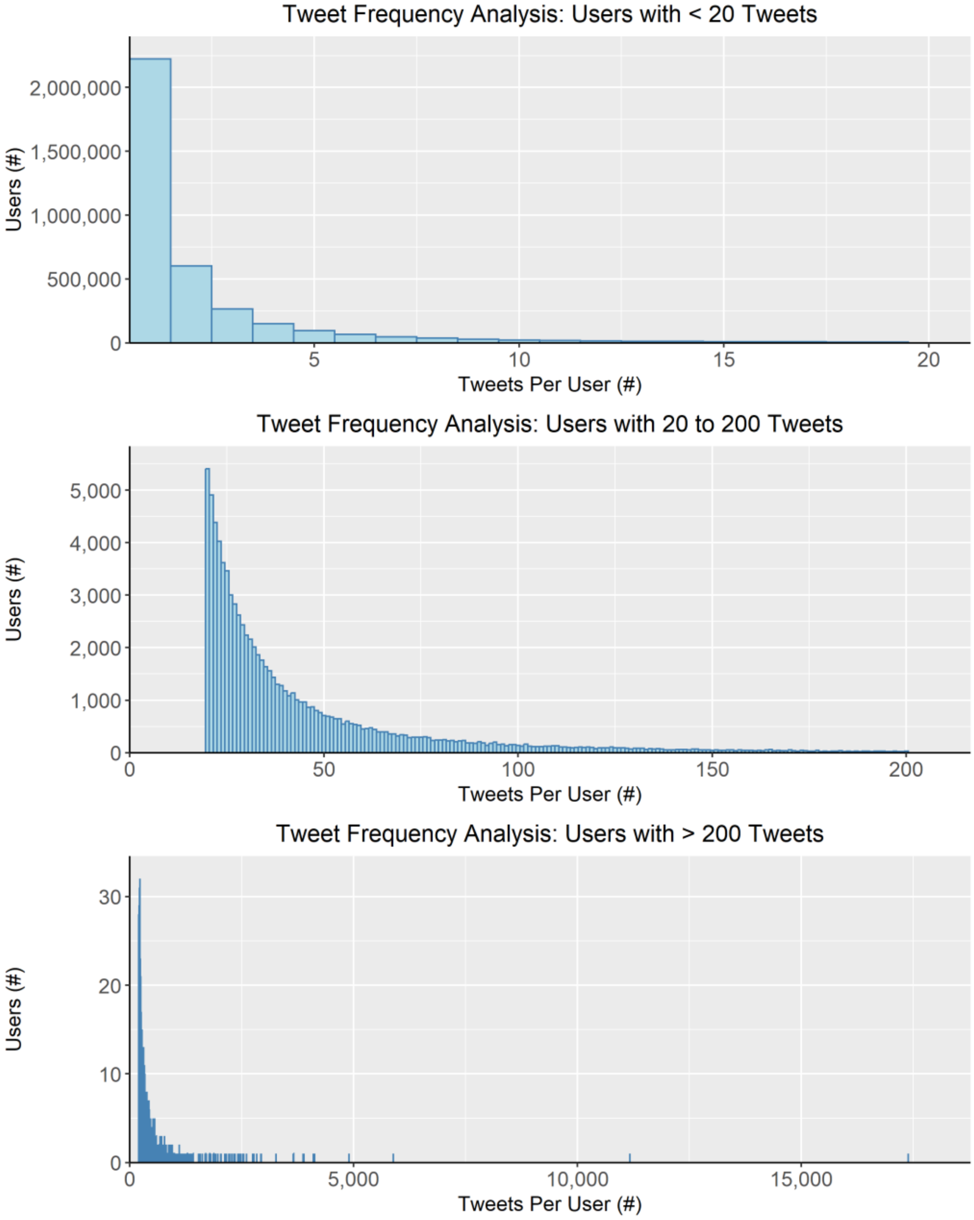
[164] https://twitter.com/ScienceCareers

[165] https://twitter.com/VaccineCa

[166] See Appendix K – *Tweeting Users Analysis*.

**Figure 14**

*Histograms – tweets per user*

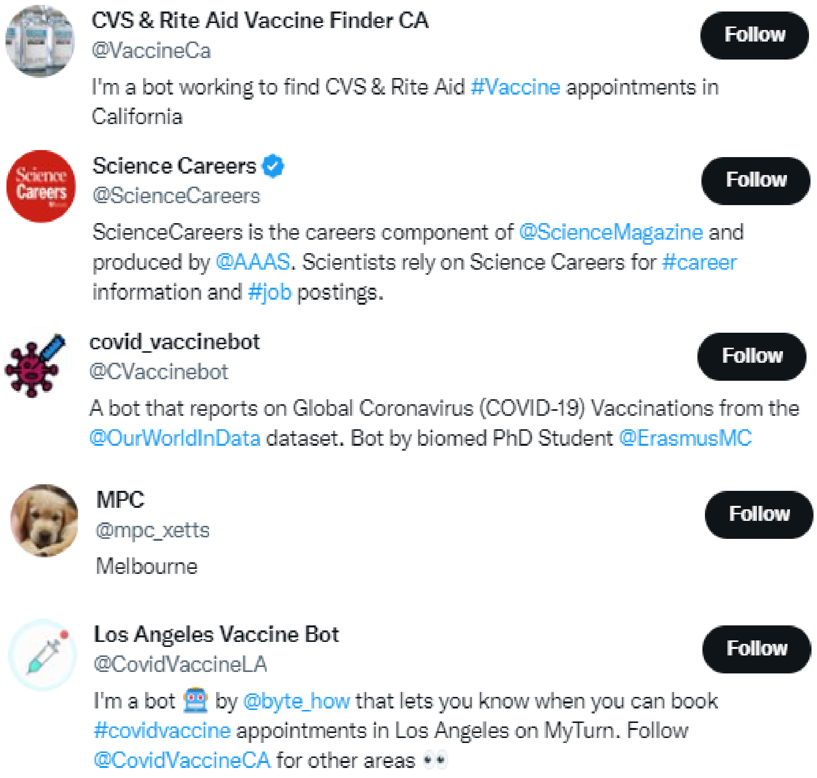


Moreover, it is noteworthy that certain users, on average, posted more than twice a day. Upon closer examination, it was noticed that these users are primarily automated bots sharing pre-programmed news. This observation is supported by Table 4 and Figure 15, which highlight the top five users with the highest number of tweets related to Pfizer. Among them, three users are explicitly identified as bots in their profile descriptions (see Figure 15). Additionally, their tweets consistently adhere to a fixed format and frequency, further reinforcing their automated nature.

**Table 4**
*Top 5 users – Pfizer tweets*

| User | Number of Tweets | Bot? |
|---|---:|---|
| @VaccineCa | 17,392 | Yes |
| @ScienceCareers | 11,175 | No |
| @CVaccinebot | 5,887 | Yes |
| @mpc_xetts | 4,895 | No |
| @CovidVaccineLA | 4,127 | Yes |

**Figure 15**
*Top 5 users – Pfizer tweets*



CVS & Rite Aid Vaccine Finder CA
@VaccineCa
I'm a bot working to find CVS & Rite Aid #Vaccine appointments in California

Science Careers
@ScienceCareers
ScienceCareers is the careers component of @ScienceMagazine and produced by @AAAS. Scientists rely on Science Careers for #career information and #job postings.

covid_vaccinebot
@CVaccinebot
A bot that reports on Global Coronavirus (COVID-19) Vaccinations from the @OurWorldInData dataset. Bot by biomed PhD Student @ErasmusMC

MPC
@mpc_xetts
Melbourne

Los Angeles Vaccine Bot
@CovidVaccineLA
I'm a bot 🤖 by @byte_how that lets you know when you can book #covidvaccine appointments in Los Angeles on MyTurn. Follow @CovidVaccineCA for other areas 👀

## 6.1.7 Most Frequent Words in Tweets in English

In this document's investigation, which centres on sentiment analysis, the specific analysis is exclusively conducted on English tweets.[167] Then, for language consistency in this section the identification of the most frequently used words in tweets mentioning "Pfizer" will be confined to English tweets too.

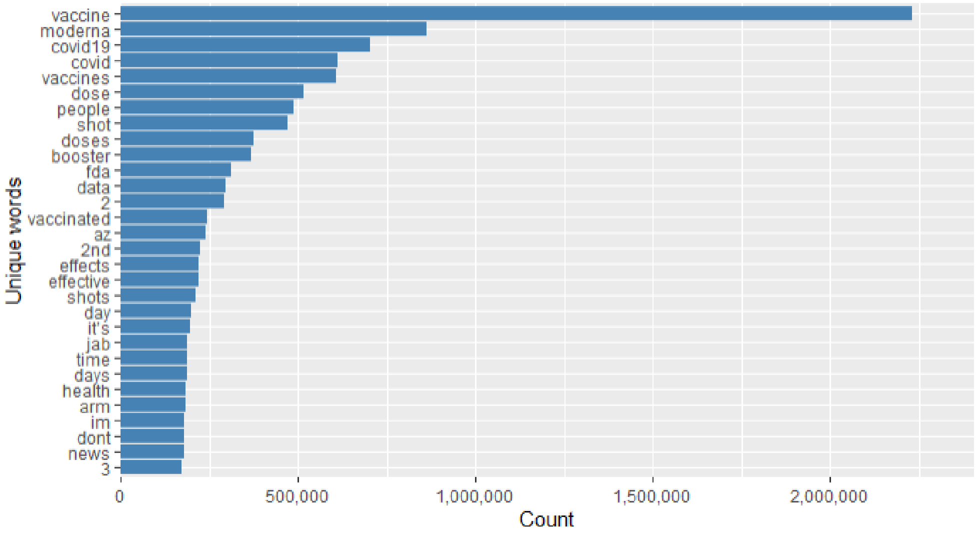
[167] See Chapter 7 *Explore – Sentiment Analysis* for detail about this.

So, to accomplish the section's objective the first step involved cleaning the tweets by removing hyperlinks, mentions, and punctuation marks.[168] Next, stop words were removed too. To remove these R's *tidytext* package was used, which includes a list of 1,149 stop words such as: "across", "allow", "certainly", "her", "particular", "they're", "would", "young", etc. Also, the word "Pfizer" was removed from the list as, by definition, it is present in every tweet at least once. Finally, the tweets were separated into individual tokens, and the frequency of each word was counted.

Despite appearing straightforward, this task was time-consuming due to the sheer volume of data, as every word in the 6.5 million English tweets had to be analysed.

The top 30 most frequent words found in tweets in English, excluding stopwords and "Pfizer", are shown graphically in Figure 16.

**Figure 16**
*30 Most frequent words (stopwords and "Pfizer" excluded)*



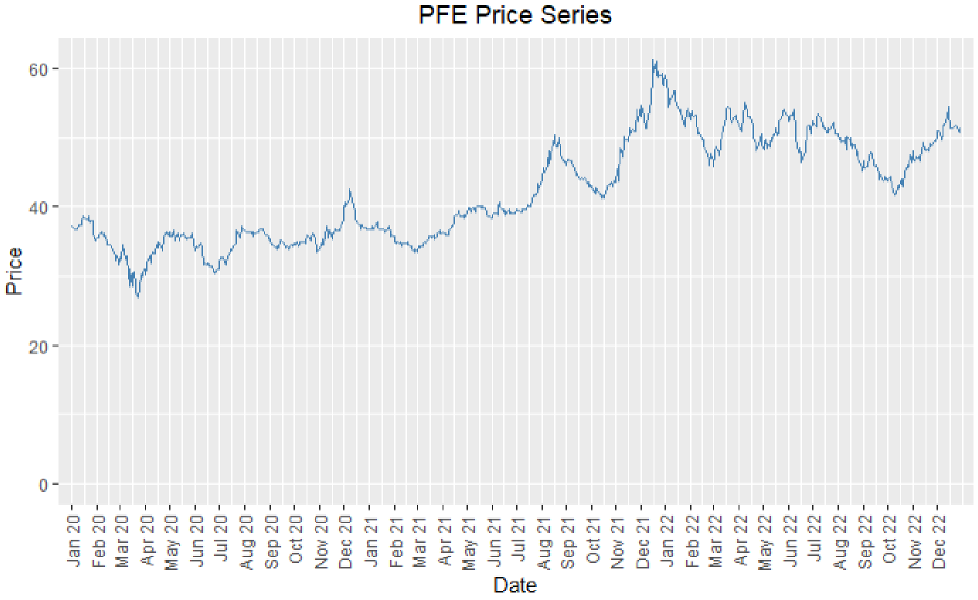
## 6.2 Pfizer Stock Descriptive Analysis

In this section, a descriptive analysis of Pfizer stock's values was carried out to gain deeper insights into its behaviour. By examining various aspects and patterns, the understanding of how the stock has performed over time was enhanced.

[168] See Appendix L – *Most Frequent Words Determination*.

Then, having extracted the data in previous steps,[169] the first thing done was to visualise the evolution of prices from 2020 until 2022.[170] Figure 17 illustrates PFE's price series in the studied period.

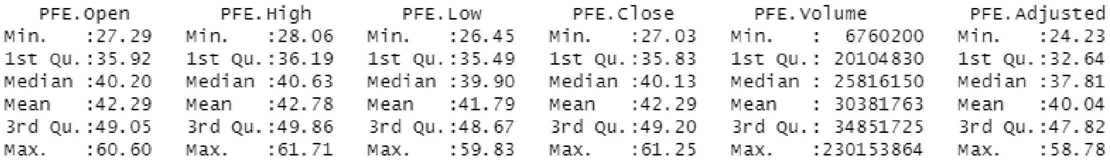
**Figure 17**
*PFE price series 2020-2022*



As can be seen in the figure, the general trend is growing until around the end of 2021, when the price peaks and starts a decreasing trend.

The following descriptive analysis was one of the main descriptive statistics (for each attribute present in the data: minimum value, 1st quartile, median, mean, 3rd quartile, and maximum value), and the results are shown in Figure 18.

**Figure 18**
*Descriptive statistics – PFE 2020-2022*



```
    PFE.Open          PFE.High          PFE.Low           PFE.Close          PFE.Volume           PFE.Adjusted
 Min.   :27.29    Min.   :28.06    Min.   :26.45    Min.   :27.03    Min.   :  6760200    Min.   :24.23
 1st Qu.:35.92    1st Qu.:36.19    1st Qu.:35.49    1st Qu.:35.83    1st Qu.: 20104830    1st Qu.:32.64
 Median :40.20    Median :40.63    Median :39.90    Median :40.13    Median : 25816150    Median :37.81
 Mean   :42.29    Mean   :42.78    Mean   :41.79    Mean   :42.29    Mean   : 30381763    Mean   :40.04
 3rd Qu.:49.05    3rd Qu.:49.86    3rd Qu.:48.67    3rd Qu.:49.20    3rd Qu.: 34851725    3rd Qu.:47.82
 Max.   :60.60    Max.   :61.71    Max.   :59.83    Max.   :61.25    Max.   :230153864    Max.   :58.78
```

The lowest low in the studied period is 26.45; as was found out by looking at the complete dataset, this happened on 23/03/2020. On this same day, the lowest closing value

[169] See Section 4.2 *Pfizer Stock Dataset Extraction*.

[170] See Appendix C – *Pfizer Stock Dataset Download* and Appendix M – *Pfizer Stock Descriptive Analysis*.

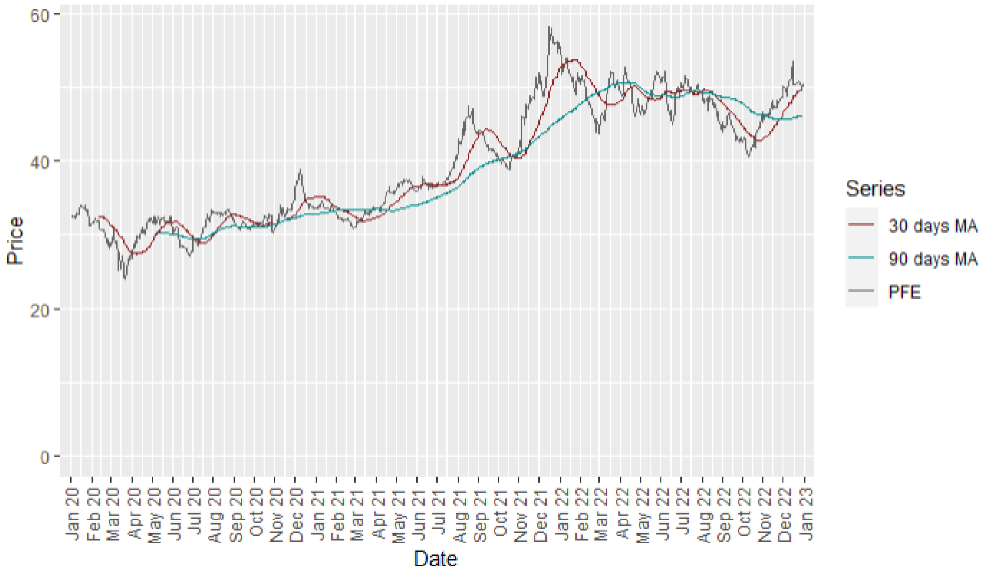
occurred, 27.03. Then, the upwards trend peaks at 61.71 on 20/12/2021, while the highest closing value is 61.25 on 16/12/2021, identifiable in Figure 17 as the highest price in the period.

Moreover, the traded volume presents an outstanding maximum. This occurred on 09/11/2020, the day Pfizer and BioNTech announced their vaccine against COVID-19 had succeeded in the first analysis from Phase 3 Study.[171]

To provide a smoother representation of the price movements, Figure 19 displays the PFE price along with its 30-day and 90-day moving averages. By incorporating these moving averages, the short-term and long-term trends of the PFE price can be observed more effectively.[172]

**Figure 19**

*PFE price series with moving averages 2020-2022*



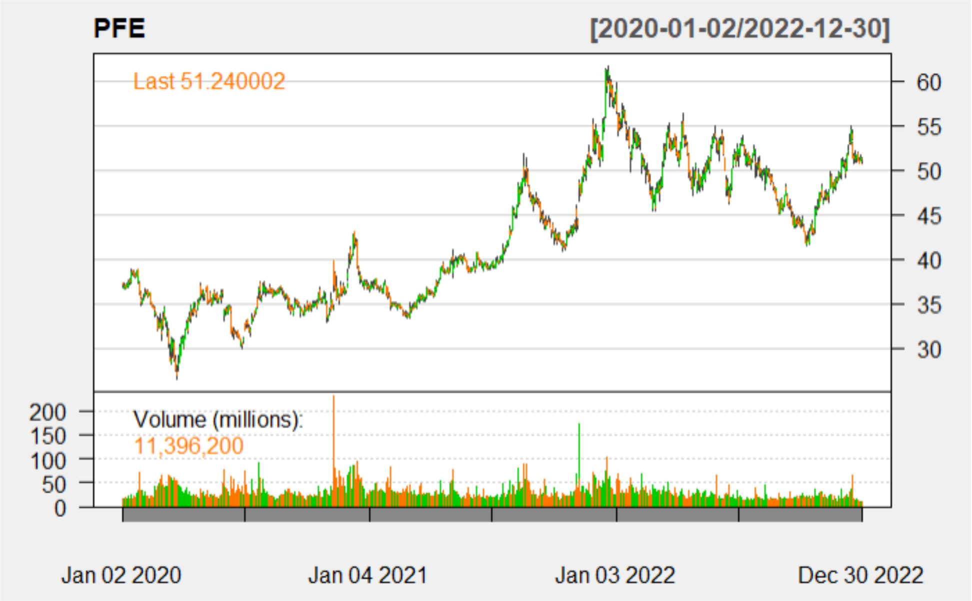
Moreover, Figure 20 shows movements in PFE’s price along with traded volume in millions. Green bars represent days when the stock’s closing price was higher than the opening price, while red bars represent the opposite. Moreover, in this figure two days in which the volume is abnormal can be spotted: 09/11/2020, explained in Section 6.1.2

[171] Pfizer (2020)

[172] See *GLOSSARY* for a definition of *simple moving average*. For additional information, refer to Raudys, A., Lenčiauskas, V., & Malčius, E. (2013).

*General Trends and Outliers*, and 05/11/2021, with no apparent reason for its high traded volume.

**Figure 20**
*PFE price series and traded volume 2020-2022*



The next step carried out to get a better understanding of the behaviour of the PFE stock in the studied period was to calculate returns. First the daily returns were calculated.[173] This was done by calculating the difference between one day's adjusted price and the previous day's one and dividing that between the previous day's adjusted price. Table 5 shows a fragment of the resulting table of daily returns.

**Table 5**
*PFE daily returns*

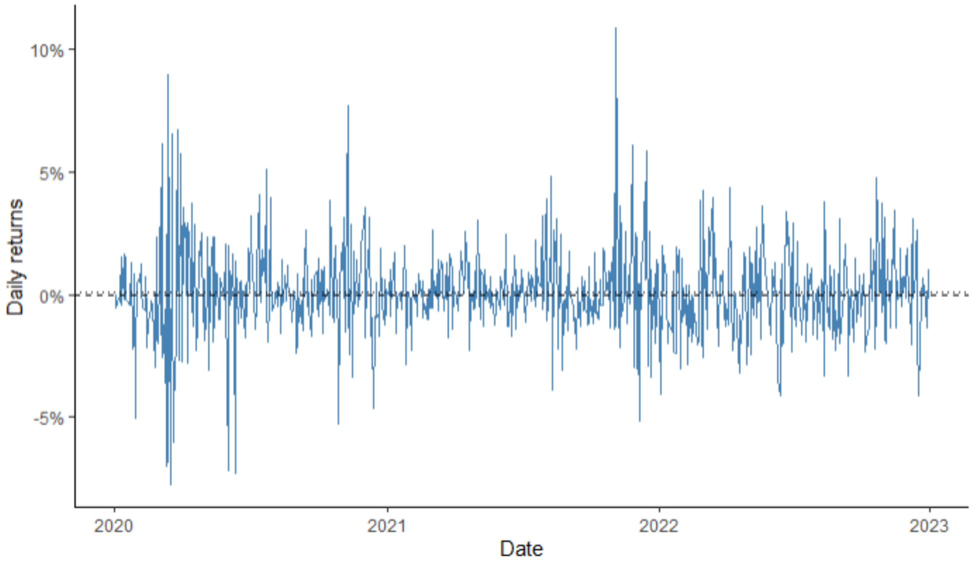
| Date | Daily Return |
|---|---:|
| 25/03/2021 | 0.0017 |
| 26/03/2021 | 0.0162 |
| 29/03/2021 | 0.0102 |
| 30/03/2021 | -0.0139 |
| 31/03/2021 | 0.0033 |

The daily returns are graphically presented in Figure 21. Daily returns are continuous data points, so a line chart was chosen to allow the visualisation of the trend and changes in

[173] See *GLOSSARY* for a definition of *daily return*. For additional information, refer to Kennan, M. (2010).

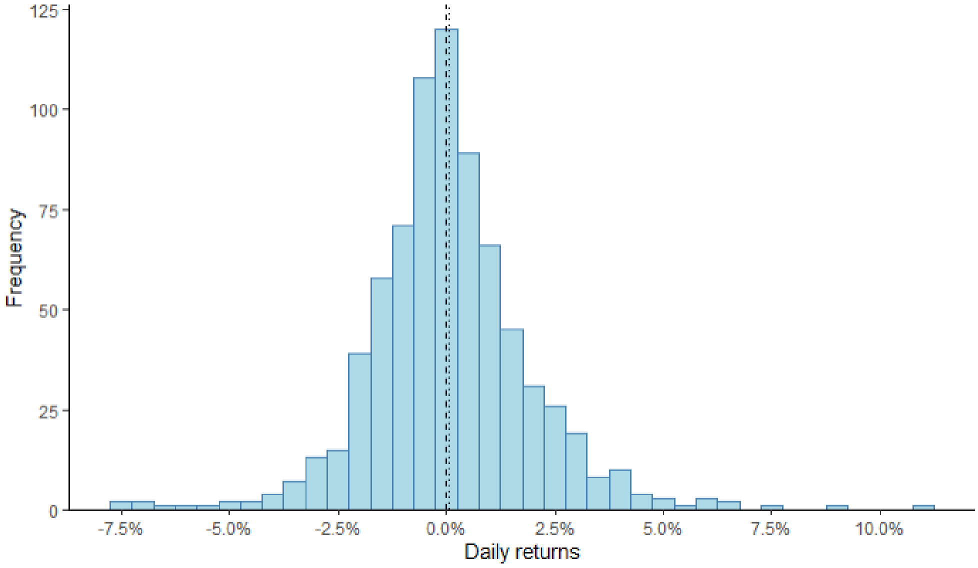
returns over time more smoothly. To enhance visual clarity two horizontal lines were drawn: a dashed line marks 0% returns, while a dotted line marks the mean return, slightly above 0%.

**Figure 21**
*PFE daily returns*



Looking at Figure 21 it can be concluded that daily returns are usually within the -4% to 4% range. To see it more clearly, a histogram of the daily returns is presented in Figure 22. To enhance visual clarity two vertical lines were drawn: a dashed line marks 0% returns, while a dotted line marks the mean return, slightly above 0%. From further calculations, the value is 0.7487%.

**Figure 22**
*PFE daily returns histogram*



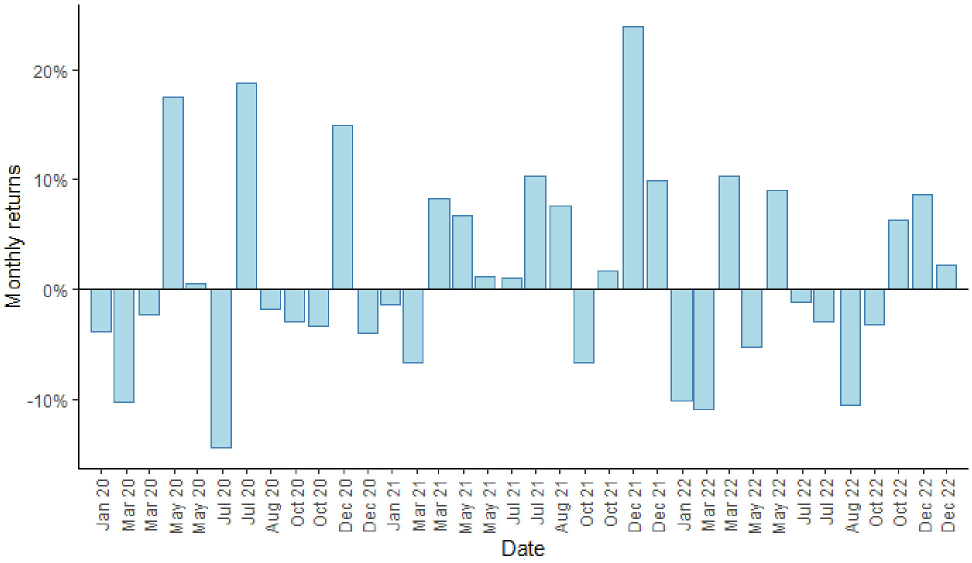
Then, the monthly returns were calculated just as with the daily returns. Table 6 shows a fragment of the resulting table of monthly returns, having in the Date column the last banking date of the month for which the monthly return was calculated.

**Table 6**
*PFE monthly returns*

| Date | Monthly Return |
|---|---:|
| 29/01/2021 | -0.0141 |
| 26/02/2021 | -0.0671 |
| 31/03/2021 | 0.0818 |
| 30/04/2021 | 0.0668 |
| 28/05/2021 | 0.0119 |

The monthly returns are graphically presented in Figure 23. Monthly returns are discrete data points, representing aggregated returns for each month. A bar chart was chosen as it allows the visualisation and comparison of the returns for different months side by side. Each bar represents a specific month, making it clearer to see the variation in returns between months.

**Figure 23**
*PFE monthly returns*



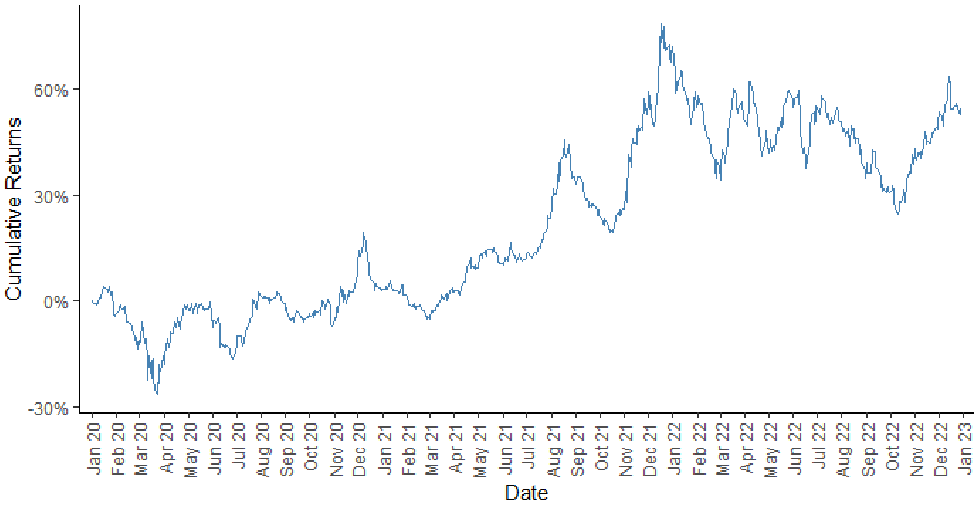
Having analysed the volatility of returns through the daily and monthly returns data, the cumulative returns were pending.[174] Cumulative returns are shown in Figure 24. $1

[174] See *GLOSSARY* for a definition of *cumulative return*. For additional information, refer to Chen, J. (2020).

investment at the beginning of 2020 turned into $1.54 by the end of 2022. However, for a long time in 2020 the cumulative return was negative, meaning that the person who had invested $1 would have less than that for some time.

**Figure 24**

*PFE cumulative returns 2020-2022*



# 7. Explore – Sentiment Analysis

In this chapter the sentiment analysis which was conducted is detailed, as well as the comparative analysis between the four sentiment indexes that were created.

From this section onwards only tweets in English were considered.[175] This is because of two main reasons: firstly, sentiment analysis packages are more developed in English and, secondly, most tweets in the considered dataset are in English.

Also, only organic tweets were considered. This is for various reasons:

- *Capturing original sentiment*: the primary aim of sentiment analysis is to understand the sentiment of the original content posted by users, so by excluding answers and comments, the analysis can focus specifically on the sentiment expressed in the initial tweets, which provides a more direct and accurate reflection of users' opinions and emotions.
- *Avoiding duplication or repetition*: answer or comment tweets often contain similar or repetitive sentiments as the original tweet they are responding to, so including these tweets in the analysis may introduce redundancy and skew the sentiment analysis results by overemphasising certain sentiments.
- *Streamlining analysis process*: analysing only non-answer or non-comment tweets simplifies the analysis process and reduces computational complexity. This approach allows for more efficient and targeted sentiment analysis, enabling the processing of a larger volume of relevant tweets within available computational resources.

From the total 13.1 million tweets, 2,962,736 are both organic and in English.

Several lexicon-based dictionaries assign sentiments and categories to words. In this project three of them were explored:

- *Bing lexicon* assigns either a positive, neutral, or negative score to each word, and has 6,786 labelled ones as either positive or negative (unlabelled words are

[175] This approach was used before in Section 6.1.7 *Most Frequent Words in Tweets in English*.

considered neutral).[176] This list was created gradually over a long period, beginning with the lexicon's authors' first paper.[177] [178]

The following are examples of Bing's labelled words:

- Amazing → positive
- Beauty → positive
- Cheap → negative
- Delay → negative
- Hot → positive
- Ready → positive
- Ridiculous → negative
- Runaway → negative
- Sorry → negative
- Thank → positive

- *AFINN lexicon* gives each word a score between -5 and 5, being negative scores related to negative sentiment and positive scores indicating positive sentiment.[179] It has 2,477 labelled words, manually labelled by the lexicon's author between 2009 and 2011. Most of the positive words were assigned a +2 score while most negative ones were assigned a -2 score, with strong obscene words rated -4 or -5. The word list is biased towards negative words (1598 words, 65% of the total) compared to positive words (878 words, 35% of the total). Only one phrase was labelled as neutral.[180] [181]

  The following are examples of AFINN's labelled words:

  - Outstanding → 5

[176] Liu, B. (2004)

[177] Hu, M., & Liu, B. (2004)

[178] The application of this method to the project developed in the present document is detailed in Section 7.1 *Sentiment Analysis with Bing*.

[179] Nielsen, F. Å. (2011)

[180] Wilson, T., Wiebe, J., & Hoffmann, P. (2005)

[181] The application of this method to the project developed in the present document is detailed in Section 7.2 *Sentiment Analysis with AFINN*.

    - Win → 4
    - Luck → 3
    - Exclusive → 2
    - United → 1
    - Ghost → -1
    - Pathetic → -2
    - Victim → -3
    - Hell → -4
    - Bastard → -5

- *NRC Emotion Lexicon* has sentiments positive and negative, and also emotions anger, anticipation, disgust, fear, joy, sadness, surprise and trust, and to each word in its list is assigned either "yes" or "no" to each sentiment and emotion.[182] It has 13,875 labelled words, which were classified via crowdsourcing.[183] [184]

  The following are examples of NRC's labelled words:

    - Award → anticipation, joy, positive, surprise, trust
    - Crazy → anger, fear, negative
    - Depression → negative, sadness
    - Fool → disgust, negative
    - Government → fear, negative
    - Hate → disgust, fear, negative, sadness
    - Invite → anticipation, joy, positive, surprise, trust
    - Money → anger, anticipation, joy, positive, surprise, trust
    - Ridiculous → anger, disgust, negative
    - Vote → anger, anticipation, joy, negative, positive, sadness, surprise, trust

The previously described methods for sentiment analysis consider each word that forms each tweet independently. However, words put together in a sentence can make the

[182] Mohammad, S. M. (2011)

[183] Mohammad, S. M., & Turney, P. D. (2013)

[184] The application of this method to the project developed in the present document is detailed in Section 7.3 *Sentiment Analysis with NRC*.

sentiment of a tweet vary a lot from the one resulting from the words checked separately. Negation is a widely used language construct that impacts the polarity of text and hence it is essential to consider negation in sentiment analysis.[185] [186] [187]

For instance, the following three tweets: "*I am loving Pfizer*", "*I am not loving Pfizer*" and "*I am hardly loving Pfizer*" convey different meanings. While the first one expresses a positive feeling, the other two convey the opposite as they have the words "not" and "hardly" in them. However, when analysed with Bing, all of them have one positive point and nothing else. This is because the only labelled word in the three tweets is "loving", which has a positive label. So, though one would interpret the three tweets differently, a sentiment analysis that does not consider negation may assume the three tweets convey the same feeling towards Pfizer.

The *SentimentR* package in R allows the extraction of negation-sensitive polarity scores for each tweet.[188] It is designed to quickly calculate text polarity sentiment in English at the sentence level, considering valence while maintaining speed. The *SentimentR* package has a lexicon of 11,709 words with scores ranging from -2 to 1, which can be used for sentiment analysis. The package also has a set of 140 valence shifters that take an integer value from 1 to 4.[189] Valence shifters are the following ones:

- *Negators* flip the sign of polarised words. For instance, the word "*not*" in the sentence "*Pfizer is not lovely*".
- *Amplifiers* (*intensifiers*) increase the impact of a polarised word. For instance, the word "extremely" in "Pfizer is extremely lovely".
- *De-amplifiers* (*downtoners*) reduce the impact of a polarised word. For instance, the words "*kind of*" in "*Pfizer is kind of lovely*".

[185] Schweinberger, M. (2022)

[186] Wiegand, M., Balahur, A., Roth, Klakow, D., & Montoyo, A. (2009)

[187] The application of this method to the project developed in the present document is detailed in Section 7.4 *Sentiment Analysis with SentimentR*.

[188] Trinker (2021)

[189] Naldi, M. (2019)

- *Adversative conjunctions* overrule the previous clause containing a polarised word. For instance, the word "*although*" in "*Pfizer is lovely although not worth it*".

Table 7 shows the result of analysing the previous examples of sentences with valence shifters using the SentimentR package.

**Table 7**
*Valence shifters – SentimentR examples*

| Sentence | Word Count | Sentiment |
|---|---:|---:|
| Pfizer is lovely. | 3 | 0.43301270 |
| Pfizer is not lovely. | 4 | -0.37500000 |
| Pfizer is extremely lovely. | 4 | 0.67500000 |
| Pfizer is kind of lovely. | 5 | 0.06708204 |
| Pfizer is lovely although not worth it. | 7 | -0.63809852 |

As can be seen, the base case ("*Pfizer is lovely*", which has no valence shifters) has a positive sentiment. However, the second one, which includes the word "*not*" is assigned a negative sentiment. Moreover, the third one, which has the word "*extremely*" in it, has a stronger positive sentiment than the first one. Furthermore, the fourth sentence, which has "*kind of*" in it, also has a positive sentiment but a weaker one than the one in the base case. Finally, the fifth sentence, which includes "*although not worth it*", has a negative sentiment as the presence of the adversative conjunction is stronger than what is said before it.

## 7.1 Sentiment Analysis with Bing

The first method for sentiment analysis employed was Bing.[190] As was mentioned previously in this chapter, this method assigns "negative" or "positive" to each labelled word found in a tweet and has 6,776 labelled words.

To assign a positivity score to each tweet the steps are the following:

1. The first step is to break all tweets into their tokens so each token can be analysed.
2. Following, each token is analysed and either a positive, a negative or no value is assigned.

[190] See Appendix O – *Bing Positivity Index Calculation*.

3. Then all the tokens from the same tweet are combined so that a total positive and negative score for each tweet can be calculated. Some tweets may have no score at all. These are tweets that Bing has classified as neutral.
4. Finally, a positivity index for each tweet is built. This is done by adding all the positive points and subtracting the negative ones.

An example of the resulting table is shown in Table 8; the used tweets are listed afterwards.

**Table 8**
*Bing Positivity Index per tweet*

| Tweet # | Positive | Negative | Positivity Index |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 4 | 1 | 1 | 0 |
| 6 | 3 | 4 | -1 |
| 9 | 0 | 1 | -1 |
| 10 | 3 | 0 | 3 |

The table shows the first five tweets with words labelled in the Bing lexicon as either positive or negative, while, as is developed later, the missing tweets are those in which none of the words are labelled in the lexicon. The labelled tweets are shown next:

- Tweet #1:[191]

  *NLRB Concludes Recusal Procedure **Strong**, But Potential For 'Circular Firing Squad' Persists: https://t.co/3KXf1nMHo7 ATT #pfizer #glaxosmithkline #utilities #gasutilities #utilities #gasutilities #attjobs #jobs **layoff***

  Positive word: strong

  Negative word: layoff

- Tweet #4:[192]

  *Pfizer, in 2013, quietly asked the F.D.A. and regulators in other countries to ban Depo-Medrol for epidural use. If they were injured by a Depo-Medrol epidural within the Statute of **Limitations** there may be an attorney near them that would **like** to meet them. https://t.co/ezsEiIju2Q*

[191] https://twitter.com/StafferTech/status/1212516376222609412

[192] https://twitter.com/Ted_LQ4L/status/1212510289855471617

Positive word: like

Negative word: limitations

- Tweet #6:[193]

  *Drugmakers from Pfizer to GSK to hike prices on over 200 drugs http://a.msn.com/00/en-us/BBYvfkV?ocid=st This is called a* ***fake Trump*** *moment! Goes a little* ***like*** *this, Will raise our price to a* ***crazy*** *amount to get the public attention then* ***wobbles*** *in the cape crusader aka* ***Trump*** *to say no! New* ***BS*** 🕵

  Positive words: trump, like, trump

  Negative words: fake, crazy, wobbles, bs

- Tweet # 9:[194]

  *They got a huge Taxscam in 2017, drug prices are higher in USA than any other country and now they raise prices for no apparent reason - Big Pharma* ***Greed*** *Drugmakers from Pfizer to GSK to hike prices on over 200 drugs https://t.co/b9OL7ELYLw*

  Positive word: greed

- Tweet #10:[195]

  *Empowered for Equality: How Women Can* ***Advocate*** *for Their* ***Worth*** *at* ***Work****: https://t.co/2Lke3XIJg3 ATT #pfizer #glaxosmithkline telecom utilities gasutilities #OMP #PMP #fired*

  Positive words: advocate, worth, work

Many tweets, such as numbers 2, 3, 5, 7 and 8, have no value. This is because none of the words included in the tweet are one of Bing's labelled words. For instance,

- Tweet #2:[196]

  *Exclusive: Drugmakers from Pfizer to GSK to hike U.S. prices on over 200 drugs https://t.co/hLHxaWc1H5*

[193] https://twitter.com/SSGEricB/status/1212507712145068032

[194] https://twitter.com/MrBill_Resists/status/1212500986104811520

[195] https://twitter.com/StafferTech/status/1212497502198521856

[196] https://twitter.com/granny713212/status/1212515805155536896

None of the words in the tweet are labelled, so neither a negative nor a positive score is assigned.

After analysing every tweet, and once the positivity index is ready, tweets are assembled by date so that a positivity index for each date of the period can be calculated. The Bing Positivity Index was calculated as an average, which is the result of dividing the sum of positivity indexes of each date's tweets by the total number of organic tweets in English in the day, and some lines of the resulting table are shown in Table 9.

**Table 9**
*Bing Positivity Index per day*

| Date | Positivity Index | Tweets Count | Positivity Index (Average) |
|---|---:|---:|---:|
| 2020-01-01 | 6 | 315 | 0.0190 |
| 2020-01-02 | 27 | 181 | 0.1492 |
| 2020-11-08 | 1 | 123 | 0.0081 |
| 2020-11-09 | 31,065 | 57,797 | 0.5375 |
| 2022-06-10 | -335 | 1098 | -0.3051 |

Figure 25 shows the daily evolution of the Bing Positivity Index, while Figure 26 shows its monthly evolution.

**Figure 25**
*Bing Positivity Index per day*

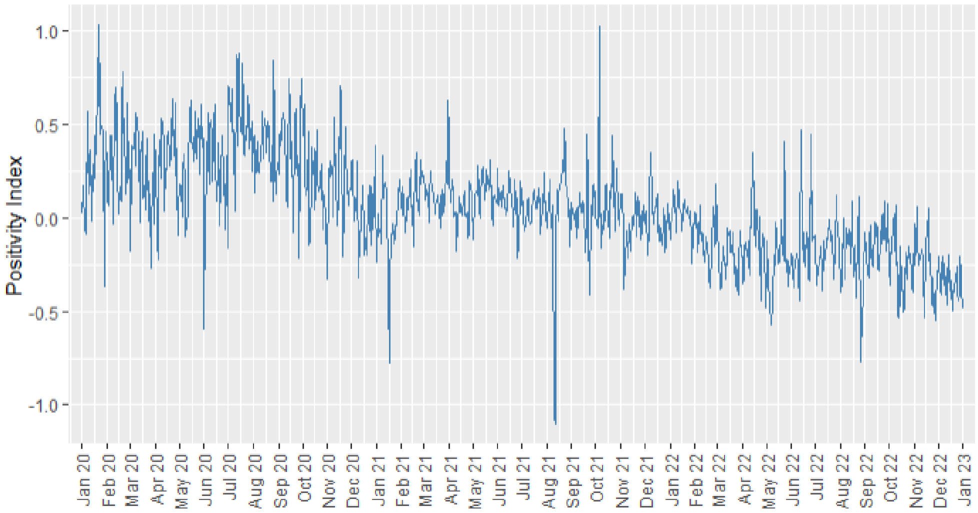
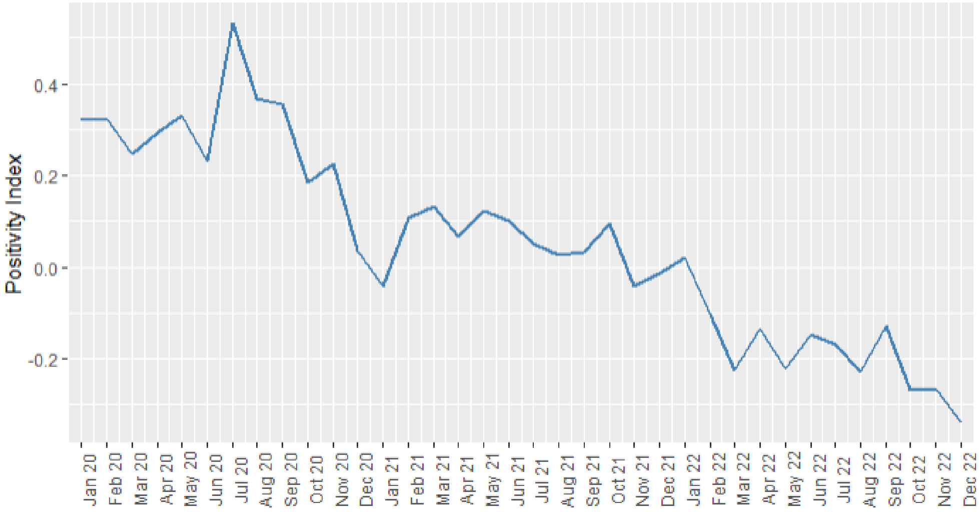


**Figure 26**

*Bing Positivity Index per month*



Moreover, Figure 27 shows the total number of analysed tweets with their resulting sentiment classified as negative, neutral and positive. Also, Figure 28 shows the same but with percentages. It is particularly noticeable in this graph that the proportion of positive tweets gets smaller over time while the one of negative tweets gets bigger.

**Figure 27**

*Bing tweets by sentiment per month*

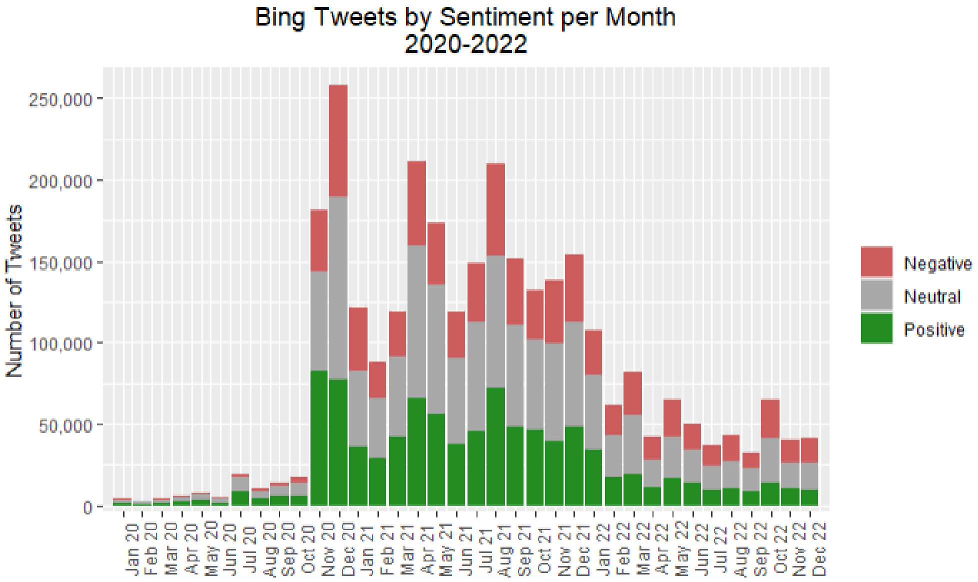
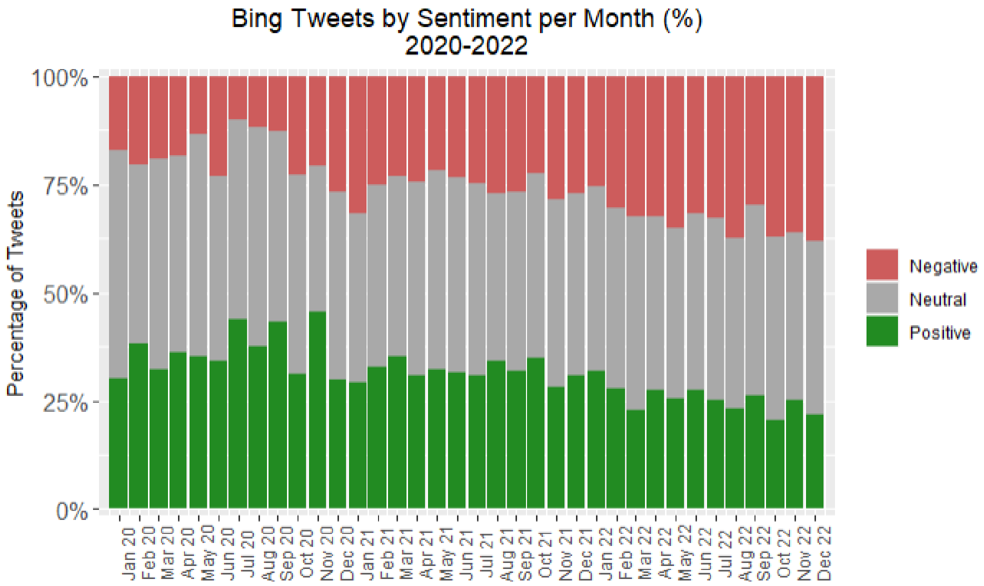


**Figure 28**
*Bing tweets by sentiment per month (%)*



## 7.2 Sentiment Analysis with AFINN

The second method for sentiment analysis employed is AFINN.[197] As was elaborated previously in this chapter, this method assigns a score between -5 and 5 to each word in a tweet and has 2,477 labelled words.

To assign a positivity score to each tweet the steps are similar to the ones carried out for the Bing sentiment analysis:

1. The first step is to break all tweets into their tokens so each token can be analysed.
2. Following, each token is analysed and a score between -5 and 5 is assigned.
3. Then all the tokens from the same tweet are combined so that a total score for each tweet can be calculated. Some tweets may have no score at all. These are tweets which have been classified as neutral according to AFINN.
4. Finally, a positivity index for each tweet is built. This is done by adding all the positive points and subtracting the negative ones.

An example of the resulting table is shown in Table 10; the used tweets are listed afterwards.

[197] See Appendix P – *AFINN Positivity Index Calculation*.

**Table 10**
*AFINN Positivity Index per tweet*

| Tweet # | Positivity Index |
|---|---|
| 1 | 0 |
| 2 | 2 |
| 3 | 2 |
| 4 | -2 |
| 6 | -4 |

The table shows the first five tweets with words labelled in the AFINN lexicon as either positive or negative, while, as is developed later, the missing tweets are those in which none of the words are labelled in the lexicon. The labelled tweets are:

- Tweet #1:[198]

  *NLRB Concludes Recusal Procedure **Strong**, But Potential For 'Circular **Firing** Squad' Persists: https://t.co/3KXf1nMHo7 ATT #pfizer #glaxosmithkline #utilities #gasutilities #utilities #gasutilities #attjobs #jobs layoff*

  Scored words: strong (2), firing (-2)

- Tweet #2:[199]

  ***Exclusive**: Drugmakers from Pfizer to GSK to hike U.S. prices on over 200 drugs https://t.co/hLHxaWc1H5*

  Scored words: exclusive (2)

- Tweet #3:[200]

  ***Exclusive**: Drugmakers from Pfizer to GSK to hike U.S. prices on over 200 drugs https://t.co/jWXtTcOywb*

  Scored words: exclusive (2)

- Tweet #4:[201]

  *Pfizer, in 2013, quietly asked the F.D.A. and regulators in other countries to **ban** Depo-Medrol for epidural use. If they were **injured** by a Depo-Medrol epidural*

[198] https://twitter.com/StafferTech/status/1212516376222609412

[199] https://twitter.com/granny713212/status/1212515805155536896

[200] https://twitter.com/HasidPuentes/status/1212514011046133760

[201] https://twitter.com/Ted_LQ4L/status/1212510289855471617

*within the Statute of Limitations there may be an attorney near them that would **like** to meet them. https://t.co/ezsEiIju2Q*

Scored words: ban (-2), injured (-2), like (2)

- Tweet #6:[202]

  *Drugmakers from Pfizer to GSK to hike prices on over 200 drugs http://a.msn.com/00/en-us/BBYvfkV?ocid=st This is called a **fake** Trump moment! Goes a little **like** this, Will raise our price to a **crazy** amount to get the public attention then wobbles in the cape crusader aka Trump to say **no**! New BS* 🕵️

  Scored words: fake (-3), like (2), crazy (-2), no (-1)

Some tweets, such as number 5, have no value. This is because none of the words included in the tweet are one of AFINN's labelled words. For instance,

- Tweet #5:[203]

  *https://t.co/HELrU4TV7i: Drugmakers from Pfizer to GSK to hike US prices on over 200 drugs. https://t.co/GTYXytTvtC via @GoogleNews*

None of the words in the tweet are labelled, so neither a negative nor a positive score is assigned.

After analysing every tweet, and once the positivity index is ready, tweets are assembled by date so that a positivity index for each date of the period can be calculated. The AFINN Positivity Index was calculated as an average, which is the result of dividing the sum of positivity indexes of each date's tweets by the total number of organic tweets in English in the day, and some lines of the resulting table are shown in Table 11.

[202] https://twitter.com/SSGEricB/status/1212507712145068032

[203] https://twitter.com/PersianKittenz/status/1212509546087944192

**Table 11**
*Bing Positivity Index per day*

| Date | Positivity Index | Tweets Count | Positivity Index (Average) |
|---|---:|---:|---:|
| 2020-01-01 | 296 | 315 | 0.9397 |
| 2020-01-02 | 169 | 181 | 0.9337 |
| 2020-11-08 | 78 | 123 | 0.6341 |
| 2020-11-09 | 76,597 | 57,797 | 1.3253 |
| 2022-06-10 | -471 | 1098 | -0.4290 |

Figure 29 shows the daily evolution of the AFINN Positivity Index while Figure 30 shows its monthly evolution.

**Figure 29**
*AFINN Positivity Index per day*

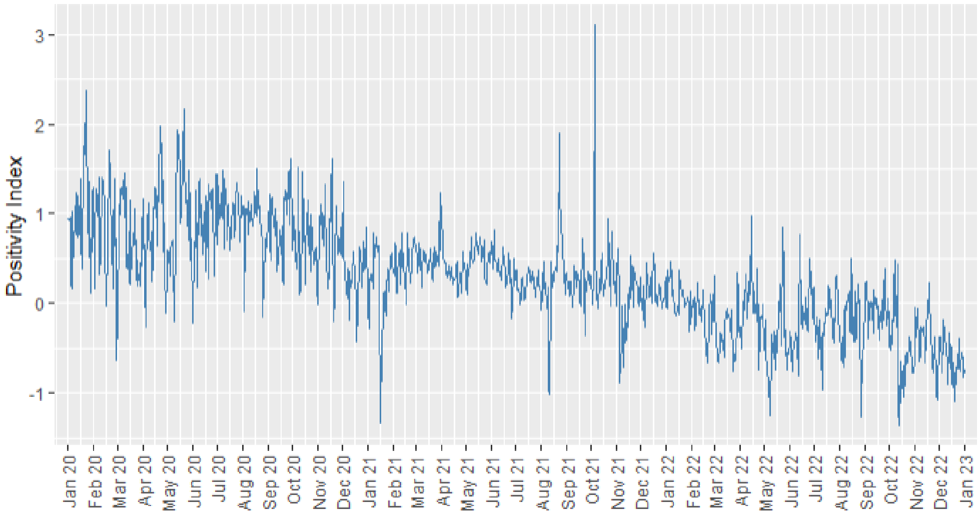
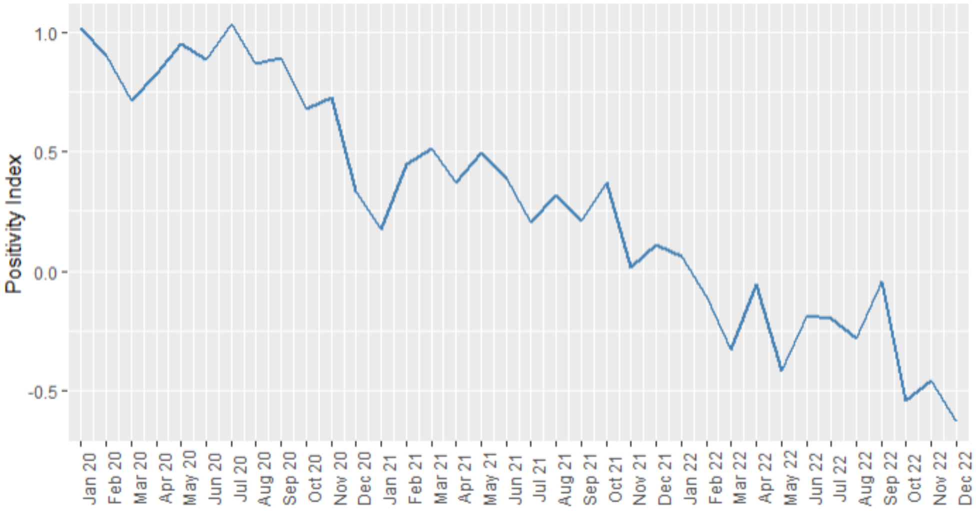


**Figure 30**
*AFINN Positivity Index per month*



Moreover, Figure 31 shows the total number of analysed tweets with their resulting sentiment classified as negative, neutral and positive. Moreover, Figure 32 shows the same but with percentages. It is particularly noticeable in this graph that the proportion of positive tweets gets smaller over time while the one of negative tweets gets bigger.

**Figure 31**

*AFINN tweets by sentiment per month*

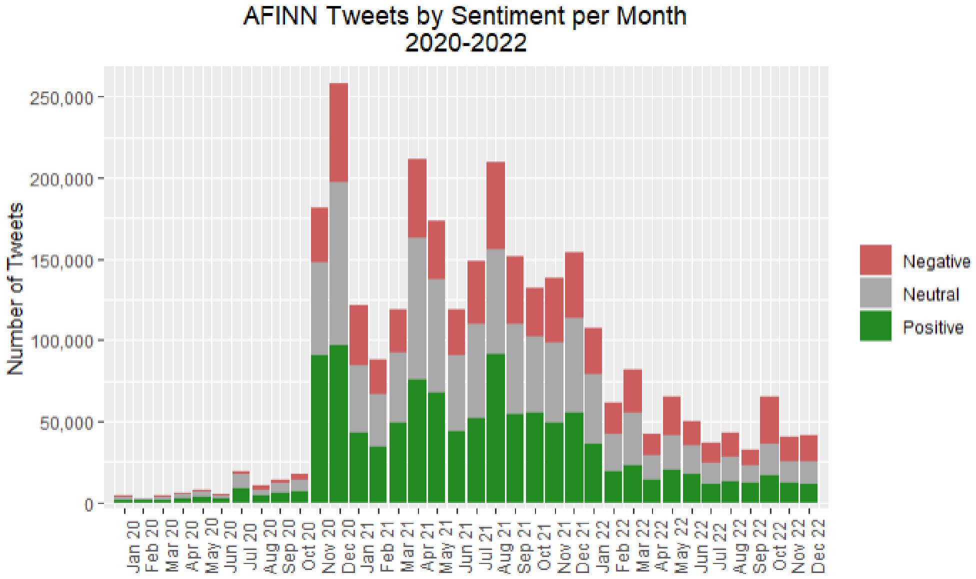
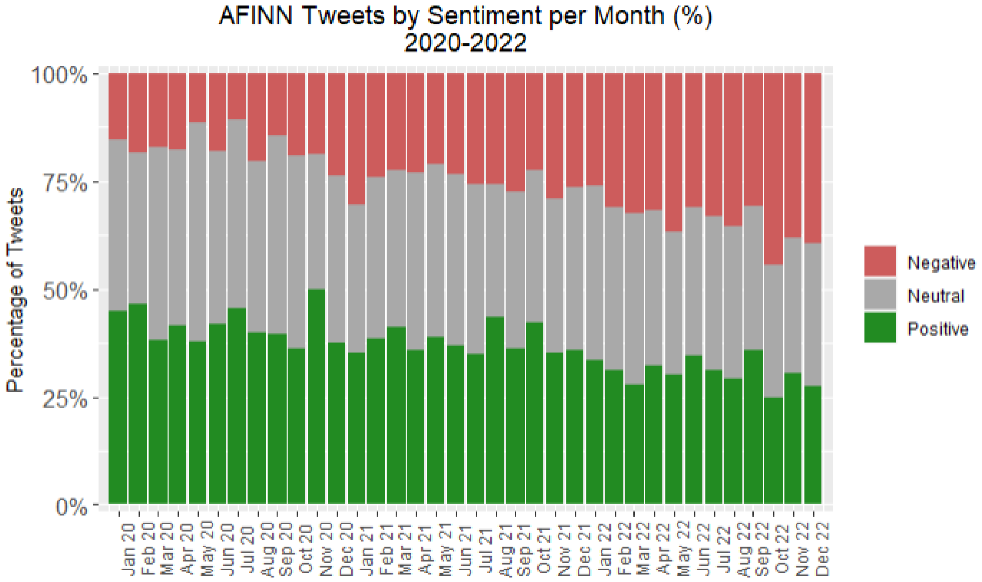


**Figure 32**

*AFINN tweets by sentiment per month (%)*



## 7.3 Sentiment Analysis with NRC

The third method for sentiment analysis employed is NRC.[204] As was mentioned previously in this chapter, this method assigns categories positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise and trust to each word in a tweet.

To carry out the analysis the steps are similar to the ones followed for the other sentiment analysis:

1. The first step is to break all tweets into their tokens so each token can be analysed.
2. Following, each token is analysed and a sentiment or none is assigned.
3. Then all the tokens from the same tweet are combined so that a total score for each sentiment for each tweet can be calculated. Some tweets may have no score at all. These are tweets which are neutral according to the analysis with NRC.
4. Finally, a positivity index for each tweet is built. This is done by adding all the positive sentiment points and subtracting all negative sentiment points.

An example of the resulting table is shown in Table 12; the used tweets are listed afterwards.

**Table 12**
*NRC Positivity Index per tweet*

| Tweet # | Positive | Negative | Anger | Anticipation | Disgust | Fear | Joy | Sadness | Surprise | Trust | Positivity Index |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 4 | 1 | 2 | 1 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | -1 |
| 6 | 2 | 2 | 1 | 1 | 0 | 1 | 0 | 1 | 2 | 0 | 0 |
| 9 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 2 |

The table shows the first five tweets with words labelled in the NRC lexicon as either positive or negative, while, as is developed later, the missing tweets are those in which none of the words are labelled in the lexicon. The labelled tweets are:

[204] See Appendix Q – *NRC Positivity Index Calculation*.

- Tweet #1:[205]

  *NLRB Concludes Recusal* ***Procedure*** *Strong, But Potential For 'Circular Firing Squad' Persists: https://t.co/3KXf1nMHo7 ATT #pfizer #glaxosmithkline #utilities #gasutilities #utilities #gasutilities #attjobs #jobs layoff*

  Scored words: procedure (fear, positive)

- Tweet #4:[206]

  *Pfizer, in 2013, quietly asked the F.D.A. and regulators in other countries to* ***ban*** *Depo-Medrol for epidural use. If they were* ***injured*** *by a Depo-Medrol epidural within the Statute of Limitations there may be an* ***attorney*** *near them that would like to meet them. https://t.co/ezsEiIju2Q*

  Scored words: ban (negative), injured (fear, negative, sadness), attorney (anger, fear, positive, trust)

- Tweet #6:[207]

  *Drugmakers from Pfizer to GSK to hike prices on over 200 drugs http://a.msn.com/00/en-us/BBYvfkV?ocid=st This is called a* ***fake Trump*** *moment! Goes a little like this, Will raise our price to a* ***crazy*** *amount to get the* ***public attention*** *then wobbles in the cape crusader aka* ***Trump*** *to say no! New BS* 🕵️

  Scored words: fake (negative), trump (surprise), crazy (anger, fear, negative, sadness), public (anticipation, positive), attention (positive), trump (surprise)

- Tweet #9:[208]

  *They got a huge Taxscam in 2017, drug prices are higher in USA than any other country and now they raise prices for no apparent* ***reason*** *- Big Pharma* ***Greed*** *Drugmakers from Pfizer to GSK to hike prices on over 200 drugs https://t.co/b9OL7ELYLw*

  Scored words: reason (positive), greed (anger, disgust, negative)

[205] https://twitter.com/StafferTech/status/1212516376222609412

[206] https://twitter.com/Ted_LQ4L/status/1212510289855471617

[207] https://twitter.com/SSGEricB/status/1212507712145068032

[208] https://twitter.com/MrBill_Resists/status/1212500986104811520

- Tweet #10:[209]

  *Empowered for* ***Equality****: How Women Can* ***Advocate*** *for Their* ***Worth*** *at Work: https://t.co/2Lke3XIJg3 ATT #pfizer #glaxosmithkline telecom utilities gasutilities #OMP #PMP #fired*

  Scored words: equality (joy, positive, trust), advocate (trust), worth (positive)

Many tweets, such as numbers 2, 3, 5, 7 and 8, have no value. This is because none of the words included in the tweets are one of NRC's labelled words. For instance,

- Tweet #2:[210]

  *Exclusive: Drugmakers from Pfizer to GSK to hike U.S. prices on over 200 drugs https://t.co/hLHxaWc1H5*

None of the words in the tweet are labelled, so no sentiment is assigned.

After analysing every tweet, the positivity index is calculated by summing positive sentiments and subtracting negative ones. Once each tweet's positivity index is ready, tweets are assembled by date so that a positivity index for each date of the period can be calculated. The NRC Positivity Index was calculated as an average, which is the result of dividing the sum of positivity indexes of each date's tweets by the total number of organic tweets in English in the day, and some lines of the resulting table are shown in Table 13.

**Table 13**
*NRC Positivity Index per day*

| Date | Positivity Index | Tweets Count | Positivity Index (Average) |
|---|---:|---:|---:|
| 2020-01-01 | 53 | 315 | 0.1683 |
| 2020-01-02 | 95 | 181 | 0.5249 |
| 2020-11-08 | 42 | 123 | 0.3415 |
| 2020-11-09 | 82,571 | 57,797 | 1.4286 |
| 2022-06-10 | 414 | 1098 | 0.3770 |

Figure 33 shows the daily evolution of the NRC Positivity Index, while Figure 34 shows its monthly evolution.

[209] https://twitter.com/StafferTech/status/1212497502198521856

[210] https://twitter.com/granny713212/status/1212515805155536896

**Figure 33**

*NRC Positivity Index per day*

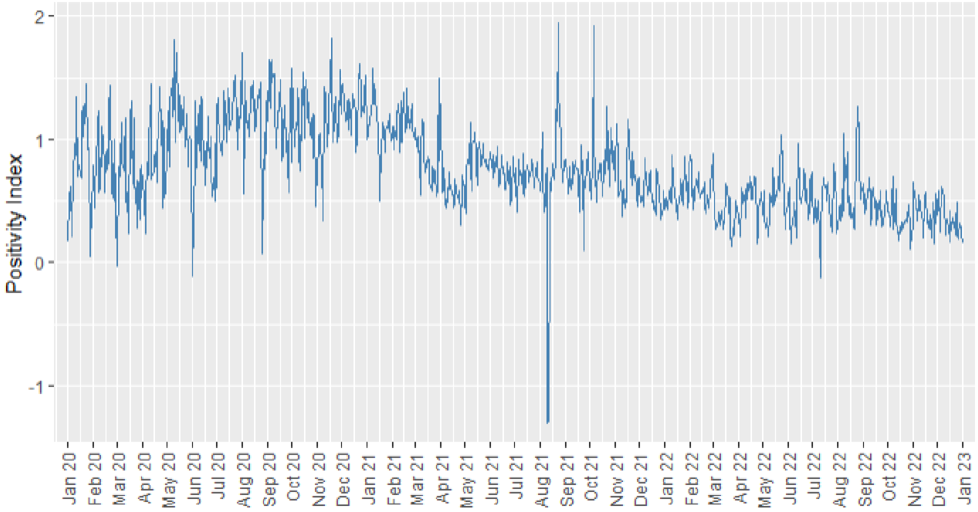
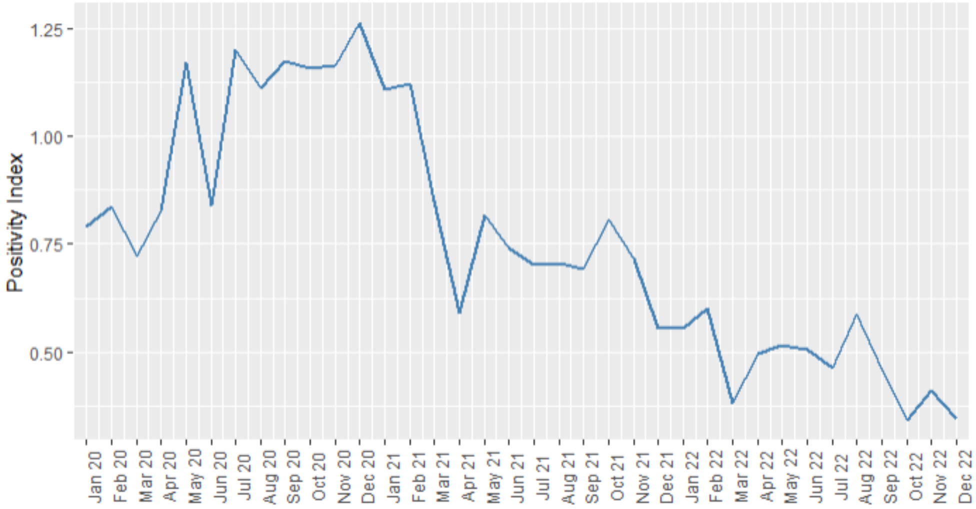


**Figure 34**

*NRC Positivity Index per month*



Moreover, Figure 35 shows the total number of analysed tweets with their resulting sentiment classified as negative, neutral and positive. Also, Figure 36 shows the same but with percentages. It is particularly noticeable in this graph that the proportion of positive tweets gets smaller over time while the one of negative tweets gets bigger.

*Figure 35*

*NRC tweets by sentiment per month*

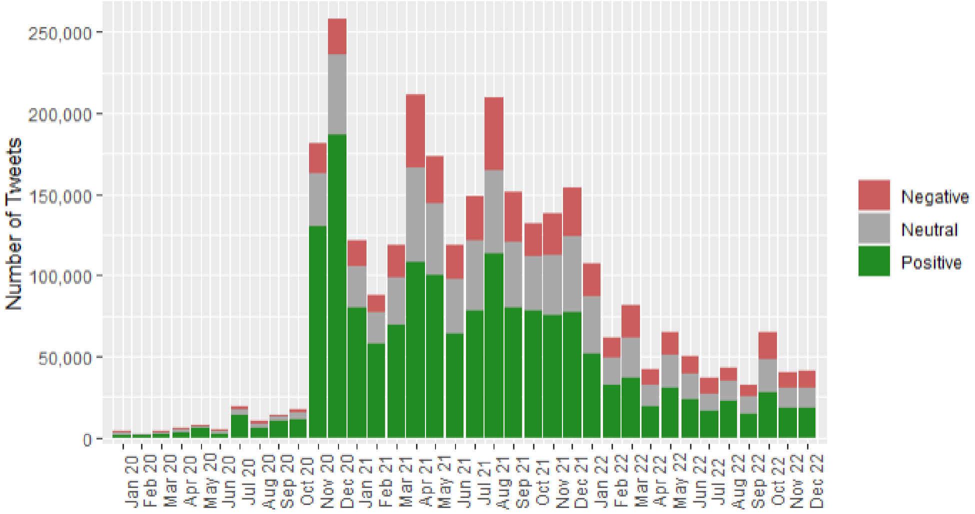
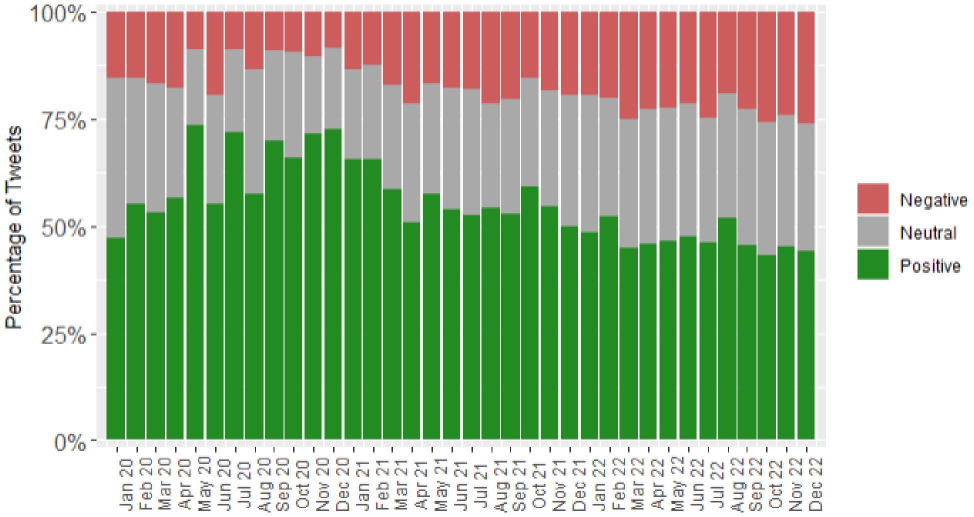


**Figure 36**

*NRC tweets by sentiment per month (%)*



## 7.4 Sentiment Analysis with SentimentR

The fourth method for sentiment analysis employed is SentimentR.[211] As was elaborated previously in this chapter, the SentimentR package calculates a sentiment score for each tweet not by checking each word individually, but rather by analysing sentences. By doing so, it considers valence while maintaining speed.

[211] See Appendix R – *SentimentR Positivity Index Calculation*.

Sentiment analysis with SentimentR was carried out, and Table 14 shows an example of the result. The used tweets are listed afterwards.

**Table 14**
*Sentiment analysis – SentimentR*

| Tweet # | Sentence Id | Word Count | Sentiment |
|---|---|---|---|
| 1 | 1 | 27 | -0.2791 |
| 2 | 1 | 19 | 0.0000 |
| 3 | 1 | 18 | 0.0000 |
| 4 | 1 | 20 | -0.1118 |
| 4 | 2 | 27 | -0.1443 |

The table shows the first five assessed sentences, which correspond to the first four tweets (note tweet #4 consists of two sentences). The previously assessed tweets are:

- Tweet #1:[212]
  *NLRB Concludes Recusal Procedure Strong, But Potential For 'Circular Firing Squad' Persists: https://t.co/3KXf1nMHo7 ATT #pfizer #glaxosmithkline #utilities #gasutilities #utilities #gasutilities #attjobs #jobs layoff*
- Tweet #2:[213]
  *Exclusive: Drugmakers from Pfizer to GSK to hike U.S. prices on over 200 drugs https://t.co/hLHxaWc1H5*
- Tweet #3:[214]
  *Exclusive: Drugmakers from Pfizer to GSK to hike U.S. prices on over 200 drugs https://t.co/jWXtTcOywb*
- Tweet #4:[215]
  *Pfizer, in 2013, quietly asked the F.D.A. and regulators in other countries to ban Depo-Medrol for epidural use. If they were injured by a Depo-Medrol epidural within the Statute of Limitations there may be an attorney near them that would like to meet them. https://t.co/ezsEiIju2Q*

[212] https://twitter.com/StafferTech/status/1212516376222609412

[213] https://twitter.com/granny713212/status/1212515805155536896

[214] https://twitter.com/HasidPuentes/status/1212514011046133760

[215] https://twitter.com/Ted_LQ4L/status/1212510289855471617

In contrast to the previously employed methods, which made it possible to analyse how each result was calculated, SentimentR is less clear to interpret and more of a black box.[216]

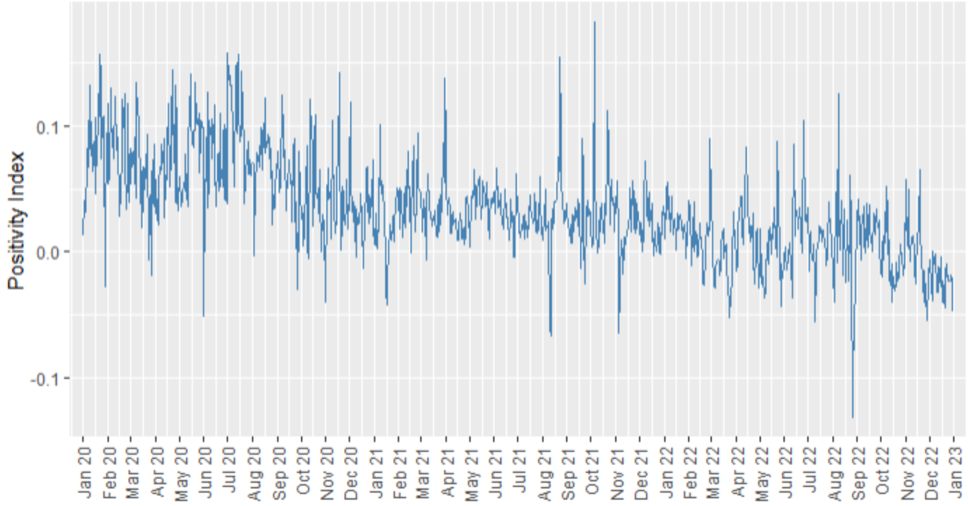
After grouping per tweet and calculating an average sentiment for each one, the head of the resulting table is what is shown in Table 15.

**Table 15**
*Sentiment for each tweet – SentimentR*

| Tweet # | Sentiment |
|:---:|---:|
| 1 | -0.2791 |
| 2 | 0.0000 |
| 3 | 0.0000 |
| 4 | -0.0854 |

Figure 37 shows the daily evolution of the SentimentR Positivity Index, while Figure 38 shows a monthly evolution of the same index.

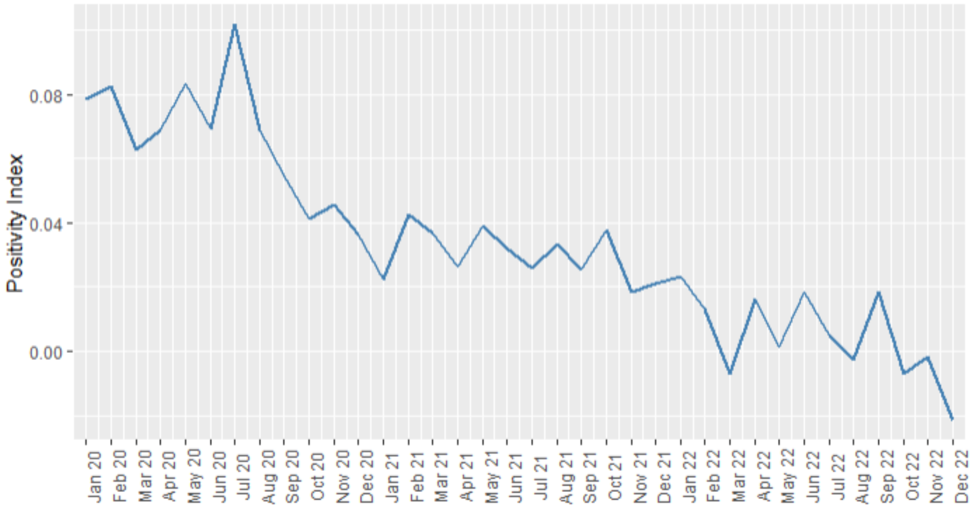
**Figure 37**
*SentimentR Positivity Index per day*



[216] See *GLOSSARY* for a definition of *black box model*. For additional information, refer to Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018).

**Figure 38**

*SentimentR Positivity Index per month*



Moreover, Figure 39 shows the total number of analysed tweets with their resulting sentiment classified as negative, neutral and positive. Also, Figure 40 shows the same but with percentages. It is particularly noticeable in this graph that the proportion of positive tweets gets smaller over time while the one of negative tweets gets bigger. Also, it can be seen that when using SentimentR to classify, fewer tweets are identified as neutral, and many more as either positive or negative.

**Figure 39**

*SentimentR tweets by sentiment per month*

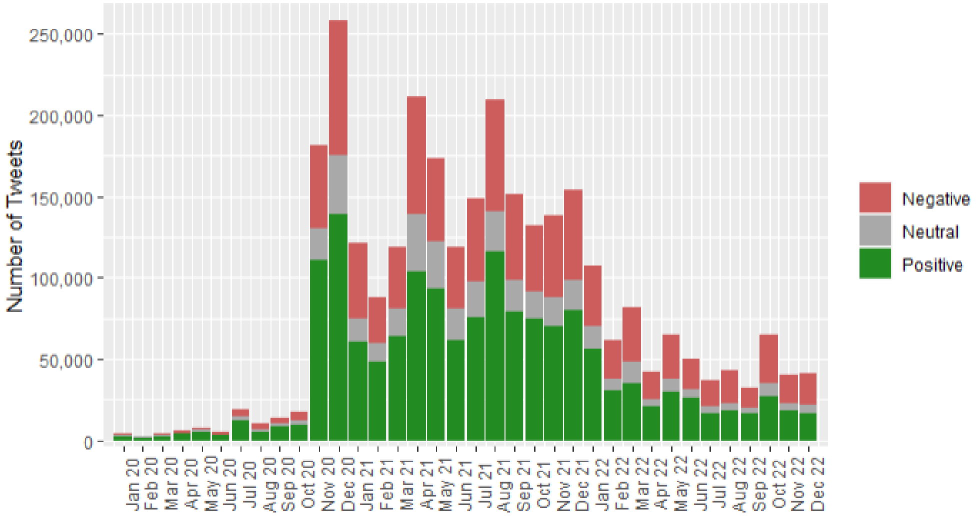
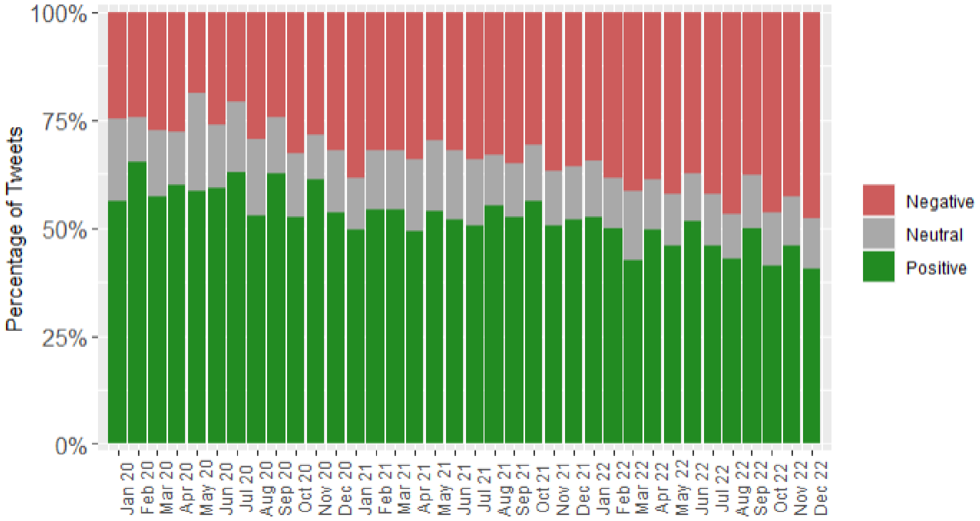


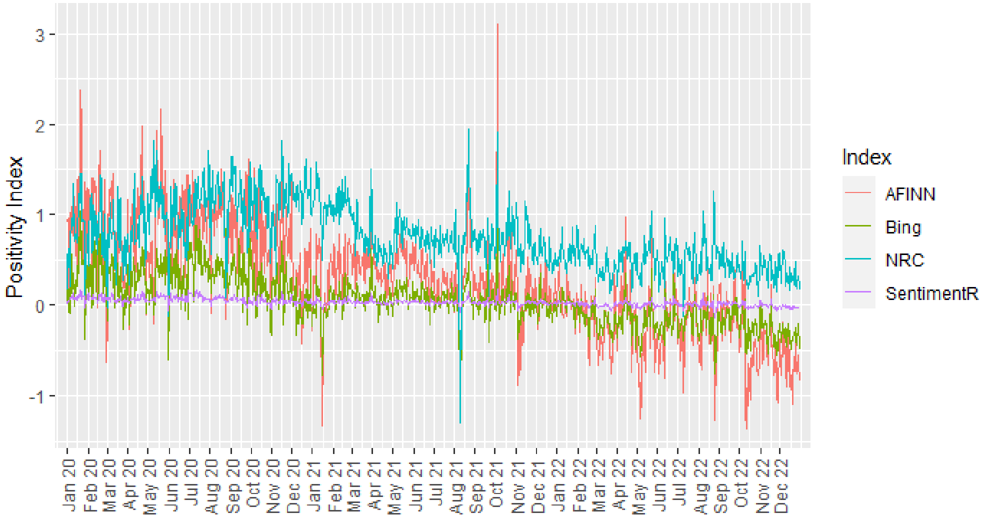
**Figure 40**
*SentimentR tweets by sentiment per month (%)*



## 7.5 Evaluation and Comparison of Sentiment Analysis Methods

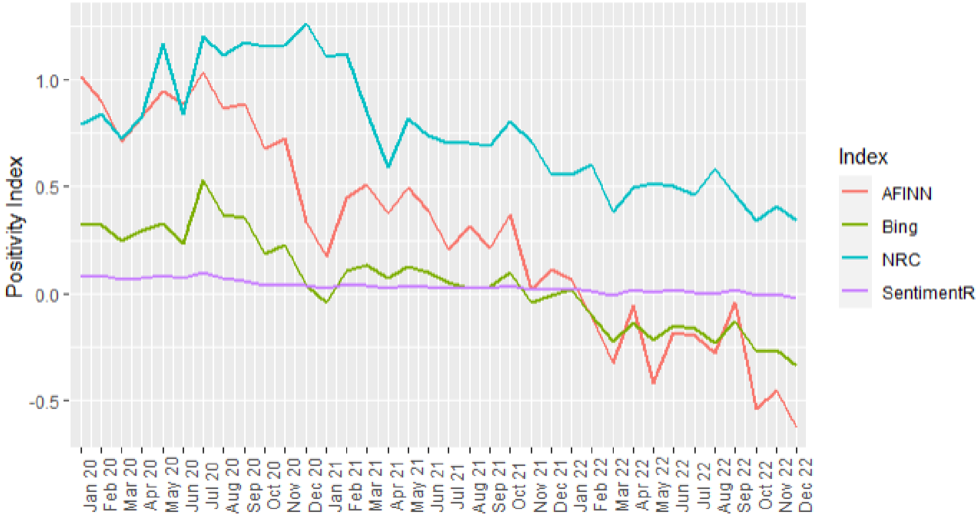
To start with, a visual analysis was developed.[217] In Figure 41 the four previously calculated daily positivity indexes are shown, while in Figure 42 the monthly positivity indexes are displayed.

**Figure 41**
*Daily positivity indexes*



[217] See Appendix S – *Comparison of Sentiment Indexes*.

**Figure 42**
*Monthly positivity indexes*



As can be seen in the figures, they all share a similar decreasing tendency, which starts with a more positive value in January 2020 and then gradually becomes less positive until it has its lowest values by the end of 2022. Also, the figures show that NRC has more positive results than AFINN, Bing and SentimentR, being the latest the less positive one.

Moreover, in the graphs, it can be appreciated that SentimentR's gradients are much more horizontal than the others, and Bing's gradient follows, which means they have less extreme positive and extreme negative values. This can be appreciated in Figure 41, where the more extreme values are usually from AFINN (seen in Figure 42 with the biggest gradient) and sometimes from NRC. Besides, the mentioned figures show that the SentimentR index showed results always nearer to 0, consistent with what can be seen in Figure 37.

To continue*, Pearson's Correlation Coefficients* between the daily positivity indexes were calculated. As described in Section 2.7 *Correlation Analysis*, *Pearson's Correlation* expresses the extent to which two variables are linearly related (meaning they change together at a constant rate). Table 16 shows the coefficients between the Bing, AFINN, NRC and SentimentR positivity indexes.

**Table 16**
*Pearson's Correlation – positivity indexes*

| | Bing | AFINN | NRC | SentimentR |
|---|---|---|---|---|
| **Bing** | 1.0000 | 0.9068 | 0.7136 | 0.8900 |
| **AFINN** | 0.9068 | 1.0000 | 0.7080 | 0.8777 |
| **NRC** | 0.7136 | 0.7080 | 1.0000 | 0.6790 |
| **SentimentR** | 0.8900 | 0.8777 | 0.6791 | 1.0000 |

The four indexes have a strong positive correlation with each other. This means that as the one positivity index increases, the rest also tends to increase, and vice versa. The high correlation coefficients suggest that the sentiment analysis results obtained from these four methods align closely with each other in terms of measuring positivity. The strongest correlation is between the Bing Positivity Index and the AFINN one, which is 0.9068, indicating the closest relationship is between these two indexes.

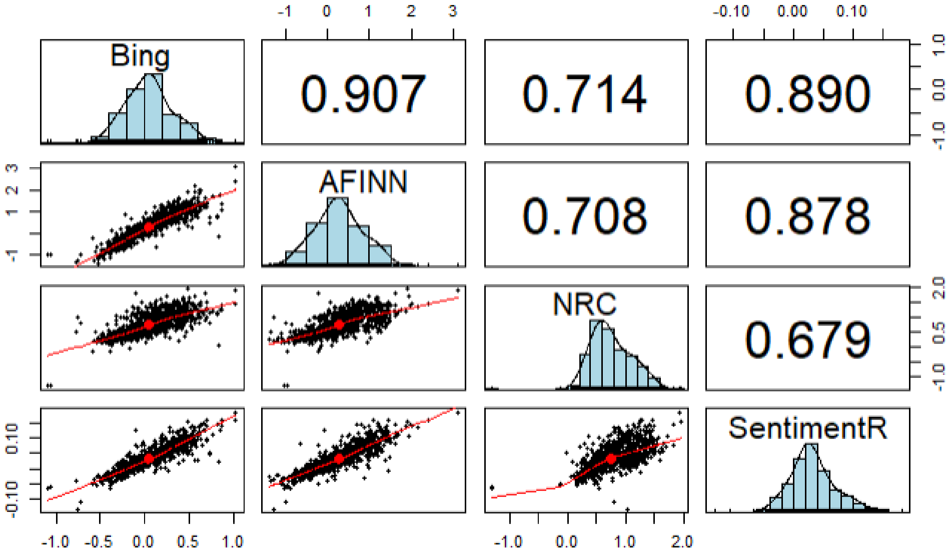
The scatter plot of matrices (SPLOM) in Figure 43 shows bivariate scatter plots below the diagonal, histograms on the diagonal, and the correlation coefficients above the diagonal.[218]

- In the scatter plots each day is represented with a point, and on each axis that day's positivity index is shown for each of the two positivity indexes represented in the plot.
- The histograms show the distribution of each positivity index's values. Each bar's height represents the number of days in which the index had a value in the respective bin's range.

[218] R (n.d.)

**Figure 43**

*SPLOM – positivity indexes (Pearson)*



Next, *Spearman's Rank Correlation Coefficients* were calculated. As described in Section 2.7 *Correlation Analysis*, *Spearman's Rank Correlation* assesses the monotonic relationship between variables (meaning that as one variable increases or decreases, the other variable consistently follows the same pattern, either increasing or decreasing). Table 17 shows the coefficients between the Bing, AFINN, NRC and SentimentR positivity indexes.

**Table 17**

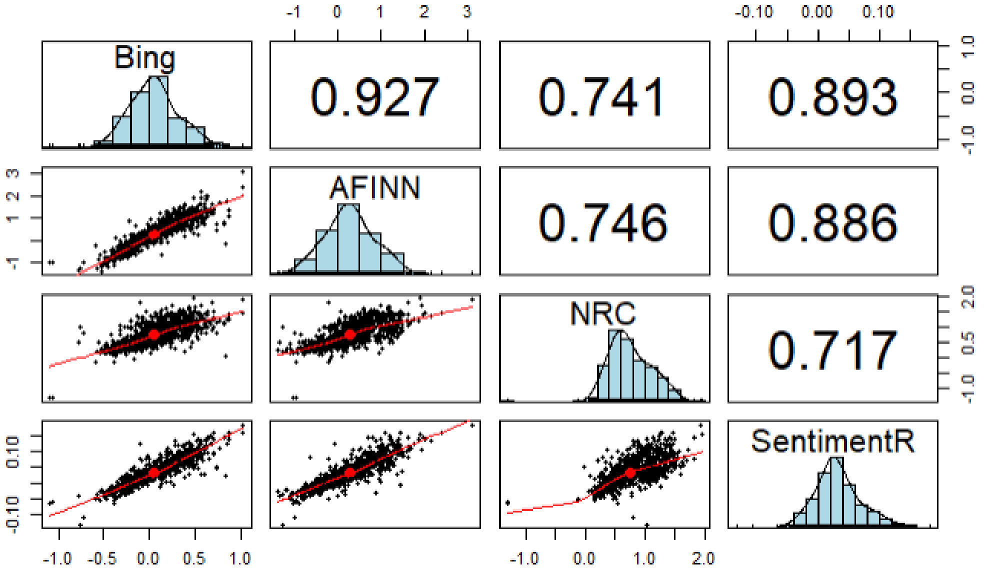
*Spearman's Rank Correlation – positivity indexes*

| | Bing | AFINN | NRC | SentimentR |
|---|---|---|---|---|
| **Bing** | 1.0000 | 0.9269 | 0.7407 | 0.8933 |
| **AFINN** | 0.9269 | 1.0000 | 0.7464 | 0.8861 |
| **NRC** | 0.7407 | 0.7464 | 1.0000 | 0.7170 |
| **SentimentR** | 0.8933 | 0.8861 | 0.7170 | 1.0000 |

Just as with the previous coefficients, these four indexes have a strong positive correlation with each other. This means that as the one positivity index increases, the rest also tends to increase, and vice versa. The high correlation coefficients suggest that the sentiment analysis results obtained from these four methods align closely with each other in terms of measuring positivity. The strongest correlation is between the Bing Positivity Index and the AFINN one, which is 0.9269, indicating the closest relationship is between these two indexes.

The scatter plot of matrices (SPLOM) in Figure 44 shows bivariate scatter plots below the diagonal, histograms on the diagonal, and the correlation coefficients above the diagonal.

**Figure 44**
*SPLOM – positivity indexes (Spearman's Rank)*



*Spearman's Rank correlation coefficients* and *Pearson's correlation coefficients* generally show similar patterns of association between the sentiment indexes. Both sets of coefficients indicate positive correlations, suggesting that as one index increases, the others also tend to increase. However, the magnitudes of the correlations may differ slightly between the two methods. This is because while the former captures the monotonic relationships between the indexes, the latter assesses the linear relationships.

# 8. Model

This chapter developed a correlation analysis between the four sentiment indexes created before and PFE's price and traded volume data. As explained in Section 2.7 *Correlation Analysis*, correlation is a statistical measure expressing the extent to which two variables are related or associated. Using quantitative methods allowed for the measurement of the strength and direction of relationships between variables, which helped provide more precise and objective results, as opposed to relying solely on subjective interpretations. Furthermore, quantitative methods enable the replication and verification of results by other researchers, increasing the study's validity and reliability.

Two correlation coefficients were calculated for each analysis: Pearson's Correlation Coefficient and Spearman's Rank Correlation Coefficient. Analysing these coefficients helped gain initial insights into the relationship between sentiment and PFE's traded volume, guiding further investigation into sentiment-market dynamics. First the correlation analysis with PFE's traded volume was carried out.[219] Then the same analysis was done with PFE's daily returns.[220]

## 8.1 Sentiment Indexes vs PFE's Traded Volume

First the correlation coefficients between each of the sentiment indexes and PFE's traded volume were calculated. This was an initial step to explore the potential relationship between sentiment and trading volume. By examining the correlation coefficients, the degree and direction of the relationship between these variables was assessed.

Pearson's and Spearman's Rank Correlation Coefficients were calculated and are shown in Table 18.

[219] See Appendix T – *Correlation Analysis: Sentiment Indexes and PFE Traded Volume*.

[220] See Appendix U – *Correlation Analysis: Sentiment Indexes and PFE Daily Returns*.

**Table 18**
*Correlation – positivity indexes and PFE's traded volume*

| | PFE Volume (Pearson's) | PFE Volume (Spearman's Rank) |
|---|---:|---:|
| **Bing** | 0.1571 | 0.2096 |
| **AFINN** | 0.1635 | 0.1944 |
| **NRC** | 0.1797 | 0.2260 |
| **SentimentR** | 0.1306 | 0.1825 |

The correlation between all positivity indexes and the traded volume is positive, meaning that more positive values of the indexes are related to bigger traded volumes. However, while there is some level of association between the variables, the strength of the correlation is relatively low (coefficients are small, near zero), meaning the correlation is weak.

## 8.2 Absolute Values of Sentiment Indexes vs PFE's Traded Volume

Next, the calculation of correlation coefficients was performed using the absolute values of the sentiment indexes instead of considering them in their original form. This approach aimed to derive a polarisation value that captured the overall intensity or strength of the sentiment expressed in the tweets, irrespective of its positive or negative nature. By focusing on the magnitude of the sentiment without distinguishing between its direction, valuable insights into the overall sentiment intensity can be gained.

The justification for using absolute values lies in the recognition that the intensity or strength of sentiment, regardless of its polarity, can significantly relate to trading activity and market behaviour. In some cases, sentiments with higher intensity, regardless of whether they are positive or negative, can drive substantial shifts in trading volume. Then, this analysis widens the understanding of the potential relationship between sentiment, beyond its direction, and market dynamics.

Table 19 shows an example of how the absolute Bing index looks for a randomly chosen five-day period.

**Table 19**
*Example – Bing Absolute Sentiment Index*

| Date | Bing Index | Bing Absolute Index |
|---|---|---|
| 01/12/2021 | -0.0127 | 0.0127 |
| 02/12/2021 | -0.0519 | 0.0519 |
| 03/12/2021 | 0.0383 | 0.0383 |
| 04/12/2021 | -0.1879 | 0.1879 |
| 05/12/2021 | -0.1958 | 0.1958 |

Pearson's and Spearman's Rank Correlation Coefficients between the absolute sentiment indexes and PFE's Traded Volume were calculated and shown in Table 20.

**Table 20**
*Correlation – absolute positivity indexes and PFE's traded volume*

| | PFE Volume (Pearson's) | PFE Volume (Spearman's Rank) |
|---|---|---|
| **Bing_abs** | 0.0440 | -0.0398 |
| **AFINN_abs** | 0.0248 | 0.0314 |
| **NRC_abs** | 0.1998 | 0.2355 |
| **SentimentR_abs** | 0.1062 | 0.1299 |

The newly calculated correlation coefficients between the absolute values of the sentiment indexes and PFE's traded volume are smaller than the previous coefficients, except for the NRC index, which shows a slightly higher correlation.

## 8.3 Sentiment Indexes vs PFE's Traded Volume (only banking days)

To conduct a more accurate correlation analysis, it is important to address the presence of zero values in the trading volume data on weekends and banking holidays. Since no stocks can be traded on these days, the trading volume is consistently zero. Therefore, removing weekends from the dataset is necessary to avoid any potential distortion in the correlation analysis.

By excluding weekends and banking holidays from the dataset, the correlation analysis is ensured to focus solely on the trading activity during weekdays when the market is open and trading volume is recorded. This adjustment helps eliminate any spurious correlations that may arise due to zero values on non-trading days. So, for the next correlation analysis weekends and banking holidays were removed from the dataset.

**Table 21**
*Correlation – positivity indexes and PFE's traded volume (banking days)*

| | PFE Volume (Pearson's) | PFE Volume (Spearman's Rank) |
|---|---|---|
| **Bing** | 0.0871 | 0.1552 |
| **AFINN** | 0.0955 | 0.1532 |
| **NRC** | 0.1480 | 0.2151 |
| **SentimentR** | 0.0751 | 0.1379 |

The correlation coefficients between the positivity indexes and PFE's traded volume, excluding weekends and banking holidays, are presented in Table 21. Despite removing non-trading days from the dataset, the correlations between the positivity indexes and trading volume remain weak and almost null.

## 8.4 Number of Tweets per Day vs PFE's Traded Volume

Next, the correlation between the number of tweets on each given day and the traded volume was studied. This analysis aimed to explore whether there is any observable relationship between the volume of tweets and the trading volume of the Pfizer stock. Understanding this relationship can provide insights into how social media activity, represented by the number of tweets, may relate to trading activity and market trends.

Furthermore, it is vital to consider the influence of weekends on this analysis. As mentioned earlier, trading volume is typically zero on weekends and banking holidays, which may affect the correlation results. Therefore, conducting the correlation analysis both with and without weekends in the dataset allowed for a more comprehensive understanding of the relationship between tweet volume and trading volume while accounting for the impact of non-trading days.

**Table 22**
*Correlation – number of tweets per day and PFE's traded volume*

| | PFE Volume (Pearson's) | PFE Volume (Spearman's Rank) |
|---|---|---|
| **Tweets count (with non-trading days)** | 0.4053 | 0.2030 |
| **Tweets count (without non-trading days)** | 0.4486 | 0.2276 |

Table 22 reveals that there is a positive correlation between the studied variables, indicating that on days when a higher number of tweets was published, there was typically a larger volume of PFE stock being traded.

## 8.5 Positivity Indexes vs PFE’s Daily Returns

The study of the correlation between positivity indexes and PFE’s daily returns contributes to the broader understanding of the interplay between sentiment, social media engagement, and stock market behaviour. It provides insights into the dynamics of sentiment-driven markets and enhances knowledge of how social media data can be utilised in financial analysis.

This analysis sheds light on whether there is a noticeable association between the sentiment expressed in tweets, as reflected by the positivity indexes, and the subsequent daily returns of PFE stock. This knowledge can benefit investors, traders, and financial analysts who seek to leverage social media data for market insights and decision-making. Moreover, this correlation analysis helps to validate the potential utility of sentiment analysis and social media data in financial analysis.

To ensure the accuracy and relevance of the calculations, this section focused exclusively on trading days, excluding weekends and banking holidays. This decision was based on the understanding that zero trading volume and daily returns on non-trading days are solely due to market rules and regulations, rather than being influenced by sentiment or other indicators.

Pearson’s and Spearman’s Rank Correlation Coefficients between each sentiment index and PFE’s daily returns were calculated and are shown in Table 23.

**Table 23**
*Correlation – positivity indexes and PFE’s daily returns*

| | PFE Daily Returns (Pearson’s) | PFE Daily Returns (Spearman’s Rank) |
|---|---|---|
| **Bing** | 0.0154 | 0.0046 |
| **AFINN** | -0.0226 | -0.0175 |
| **NRC** | 0.0118 | 0.0174 |
| **SentimentR** | 0.0279 | 0.0245 |

Both Pearson’s and Spearman’s Rank correlation coefficients are nearly zero.

## 8.6 Absolute Values of Sentiment Indexes vs PFE’s Daily Returns

Next, the correlation coefficients between the absolute values of the sentiment indexes and PFE’s daily returns were calculated. This approach focuses on the magnitude or intensity of sentiment without considering the positive or negative direction. The purpose is to explore whether the overall strength of sentiment, irrespective of its polarity, has any meaningful association with the stock's daily returns.

Table 24 presents the correlation coefficients between the absolute values of the sentiment indexes and PFE's daily returns. It is important to note that these correlation coefficients should be interpreted differently from the previous ones, as they now reflect the relationship between the intensity of sentiment and the daily returns, rather than the direction of sentiment.

**Table 24**
*Correlation – absolute positivity indexes and PFE's daily returns*

| | PFE Daily Returns<br>(Pearson's) | PFE Daily Returns<br>(Spearman's Rank) |
|---|---:|---:|
| **Bing_abs** | 0.0423 | 0.0548 |
| **AFINN_abs** | 0.0229 | 0.0213 |
| **NRC_abs** | 0.0133 | 0.0175 |
| **SentimentR_abs** | 0.0423 | 0.0279 |

The correlation coefficients in Table 24 indicate that the absolute values of the sentiment indexes have negligible correlations with PFE's daily returns.

## 8.7 Number of Tweets vs PFE's Daily Returns

Next, the correlation between the number of tweets on each given day and the daily returns was studied, and the result is presented in Table 25.

**Table 25**
*Correlation – number of tweets and PFE's daily returns*

| | PFE Daily Returns<br>(Pearson's) | PFE Daily Returns<br>(Spearman's Rank) |
|---|---:|---:|
| **Tweets count** | 0.1248 | 0.0659 |

Considering the correlation coefficients between the number of tweets and PFE's daily returns (0.1248 for Pearson's correlation and 0.0659 for Spearman's Rank correlation), there is a very weak or negligible correlation between social media activity, as measured by the number of tweets, and the day-to-day fluctuations in PFE's returns.

# 9. Interpret

The analysis presented in the previous chapters, particularly the correlation analysis conducted in Chapter 8 *Model*, provides valuable insights into the relationship between sentiment, social media activity, and Pfizer (PFE) stock market behaviour.[221]

First, in Section 8.1 *Sentiment Indexes vs PFE's Traded Volume,* the correlation analysis findings revealed a weak correlation between the sentiment indexes (Bing, AFINN, NRC, SentimentR) and PFE's traded volume. While the positive correlation coefficients suggest that higher sentiment values are associated with increased trading volume, indicating that sentiment may have some relationship to trading activity for PFE stock, the correlation coefficients are weak, with Pearson's coefficients ranging from 0.1306 to 0.1797 and Spearman's Rank coefficients ranging from 0.1825 to 0.2260. These values indicate a limited and weak relationship between sentiment and trading volume for PFE stock, suggesting that there are probably other factors more strongly related to traded volume than sentiment.

Moreover, when comparing Spearman's Rank correlation coefficients to Pearson's correlation coefficients between the positivity indexes and PFE's traded volume, some differences in the strength of the correlations can be observed. Looking at all the coefficients, the former show slightly stronger correlations than the latter. This can be attributed to the fact that Spearman's Rank coefficient considers the rank order relationship between variables, which can capture nonlinear relationships or associations that may not be captured by Pearson's coefficient.

Second, in Section 8.2 *Absolute Values of Sentiment Indexes vs PFE's Traded Volume*, when considering the absolute values of the sentiment indexes, which represent the overall intensity or strength of sentiment, the correlation with PFE's traded volume becomes even weaker. The correlation coefficients range from 0.0248 to 0.1998 for Pearson's coefficients and from -0.0398 to 0.2355 for Spearman's Rank coefficients. This implies that sentiment intensity may have a minimal association with trading volume for

[221] This chapter answers the second question presented in Section 1.4 *Research Questions*: *Do the previously analysed trends relate to Pfizer stock's trading volume and price movements?*

PFE stock, and the overall intensity of sentiment expressed in tweets may not be a significant predictor of trading volume for the mentioned stock.

These findings emphasise the importance of considering both the direction and intensity of sentiment when examining its relationship to market behaviour. While the sentiment indexes may provide insights into the sentiment expressed in tweets, their overall intensity alone may not be significantly related to trading volume in the context of PFE's stock. Further analysis and investigation are warranted to better understand the complex relationship between sentiment and trading activity in this particular context.

Third, in Section 8.3 *Sentiment Indexes vs PFE's Traded Volume (only banking days)* none of the indexes shows a strong correlation with PFE's traded volume. For Pearson's and Spearman's Rank Correlation Coefficients, the highest values are observed for the NRC index, which are 0.1480 and 0.2151 respectively, indicating a very weak positive correlation. The other indexes exhibit even lower correlation coefficients ranging from 0.0751 to 0.0955 and 0.1379 to 0.1552, suggesting an even weaker relationship.

Fourth, in Section 8.4 *Number of Tweets per Day vs PFE's Traded Volume*, when the number of tweets with non-trading days included is analysed, Pearson's correlation coefficient is 0.4053, indicating a moderate positive correlation between the variables. This suggests that as the number of tweets increases, there tends to be a corresponding increase in PFE's traded volume. Spearman's Rank correlation coefficient for the same relationship is 0.2030, which is noticeably lower. This indicates a weaker monotonic relationship between the variables, emphasising that the strength of the relationship may not be solely linear. Nevertheless, it is essential to note that this correlation coefficient includes weekends and banking holidays, where there are tweets but no traded volume due to non-trading days.

When non-trading days are excluded from the analysis, the correlation coefficients slightly increase. Pearson's correlation coefficient becomes 0.4486, indicating a stronger positive correlation between the number of tweets and PFE's traded volume. Similarly, Spearman's Rank correlation coefficient increases to 0.2276, reflecting a relatively stronger monotonic relationship between the variables.

Overall, these results suggest a potential link between social media activity and trading activity in the context of Pfizer stock, and that the number of tweets can serve as an indicator of market activity and reflect the sentiment or interest surrounding Pfizer stock. It is important to note that correlation does not imply causation and further analysis is needed to uncover the underlying mechanisms driving this relationship.[222]

Fifth, regarding PFE's daily returns, in Section 8.5 *Positivity Indexes vs PFE's Daily Returns*, the correlation coefficients between the sentiment indexes and daily returns are nearly zero. This indicates no significant linear relationship between sentiment and the day-to-day fluctuations in PFE's returns. Pearson's coefficients for daily returns range from -0.0226 to 0.0279, and Spearman's Rank coefficients range from -0.0175 to 0.0046. These values suggest that there is neither a linear nor a monotonic relationship between positivity indexes and PFE daily returns. Daily stock returns are influenced by a myriad of factors, and sentiment expressed in tweets is not a robust predictor of these fluctuations. Overall, both Pearson's and Spearman's Rank correlation coefficients indicate negligible relationships between the sentiment indexes and PFE's daily returns.

Sixth, the correlation coefficients calculated in Section 8.6 *Absolute Values of Sentiment Indexes vs PFE's Daily Returns* indicate that the absolute values of the sentiment indexes have weak and nearly negligible correlations with PFE's daily returns. Despite considering the intensity of sentiment, the overall relation of sentiment and the day-to-day fluctuations of the stock's returns remains minimal.

Finally, in Section 8.7 *Number of Tweets vs PFE's Daily Returns* the correlation coefficients close to zero suggest that there is neither a substantial linear relationship nor a monotonic relation between the number of tweets and PFE's daily returns. The limited correlation indicates that changes in social media activity, as represented by tweet volume, are not strongly associated with changes in the stock's daily returns. Therefore, it can be inferred that the number of tweets alone may not be a reliable indicator of PFE's daily return performance.

In summary, the analysis highlights the complexity of the relationship between sentiment, social media activity, and PFE's stock market performance. While some correlations exist, being the strongest one by far the correlation between the number of tweets and PFE's traded volume, they are generally weak, and further investigation is needed to fully

[222] Vigen, T. (2015)

understand the underlying dynamics driving these relationships. The results underscore the importance of considering sentiment and social media activity in conjunction with other factors when analysing stock market behaviour for PFE.

# 10. Conclusions, Applications and Future Work

This chapter highlights the key conclusions drawn and offers recommendations for future research and projects.

## 10.1 Conclusions

Based on the correlation analysis conducted in this study, it is evident that the relationship between social media sentiment, and daily returns and traded volume for PFE stock, is generally weak and nearly negligible. The only correlation of considerable strength is observed between the number of tweets per day and PFE's traded volume, indicating that social media activity may have some relation to trading activity for the mentioned stock. However, in other cases, such as sentiment indexes vs daily returns and traded volume, the effect size is relatively small and insignificant.

As a result, caution should be exercised when using sentiment indexes as sole predictors for stock returns: while they may offer some insights, they should not be solely relied upon as reliable indicators. It is crucial to incorporate other relevant factors and approaches to conduct a comprehensive analysis of stock market behaviour.

Additionally, it is essential to acknowledge the limitations of sentiment analysis, including potential data noise and biases that can influence the results. Careful interpretation and analysis are necessary to avoid drawing misleading conclusions from sentiment data.

In summary, while social media sentiment analysis can provide valuable insights, the correlation between sentiment and daily returns and traded volume for PFE stock is minimal. To gain a more accurate understanding of stock market dynamics, it is recommended to integrate sentiment analysis with other factors and methodologies to achieve a robust analysis. This comprehensive approach will help mitigate the limitations associated with relying solely on sentiment analysis and contribute to improved accuracy in understanding stock market trends and behaviour.

## 10.2 Applications for Pfizer

This study on sentiment analysis and its correlation with PFE's daily returns and traded volume can provide valuable insights and potential benefits to Pfizer itself, particularly to its finance team. By analysing the relationship between social media sentiment and stock market behaviour, the company's financial department can better understand how public sentiment relates to the company's stock performance.[223]

Incorporating sentiment analysis into their financial analysis toolkit can aid Pfizer's finance team in identifying market trends and investor sentiment patterns that might not be captured through traditional financial metrics alone. This could give them a broader perspective on the dynamics driving the stock market's response to its actions, announcements, and overall business performance.

Additionally, the insights gained from this study can be used to enhance the pharmaceutical firm's investor relations efforts. The financial department can use sentiment analysis to identify areas of concern or positive sentiment among investors and address them proactively. By responding to investor sentiment effectively, Pfizer can build and maintain stronger relationships with its stakeholders and influence market sentiment positively.

Overall, this study can offer Pfizer's finance team a data-driven approach to gauge market sentiment and its impact on the company's stock performance. By integrating sentiment analysis into their decision-making processes, they can improve their understanding of market behaviour, investor sentiment, and factors influencing stock prices. Ultimately, this could strengthen its financial strategies and communication with investors, contributing to more informed and prosperous financial decisions.

## 10.3 Future Work

This section presents some potential avenues for future research that builds upon the results of this thesis.

One area of future research is to evaluate the impact of including data from additional languages on the sentiment analysis of Pfizer tweets. Currently, only tweets in English are included in the analysis. However, Twitter is a global platform and tweets in other languages may provide valuable insights into the sentiment towards Pfizer. Therefore, it

[223] This section answers the third question presented in Section 1.4 *Research Questions*: *How can the answers to the previous questions enhance the activity of the Pfizer Finance Team?*

would be interesting to explore whether including tweets in additional languages results in different sentiment analysis outcomes.

Another potential area for future research is to assess the impact of considering different stocks on the sentiment analysis results. The present project's sentiment analysis was conducted for Pfizer tweets only. However, Pfizer is one of the companies in the S&P 500 index. Therefore, it would be interesting to investigate whether the sentiment analysis results would be the same if the analysis was conducted for the S&P 500 index as a whole or if a different company was considered.

Moreover, exploring alternative sentiment analysis techniques in future research would be beneficial. The present project employed AFINN, NRC, Bing, and SentimentR, for sentiment analysis. However, other methods, such as machine learning techniques, could further enhance the accuracy and robustness of sentiment analysis results.

Furthermore, future research could investigate the relationship between the sentiment towards Pfizer on Twitter and Pfizer's financial performance. This could be accomplished by comparing the sentiment analysis results to Pfizer's financial reports.

Lastly, expanding the analysis beyond Twitter and including other social media platforms, such as Facebook, Instagram, or LinkedIn, would be valuable. This could provide a more comprehensive understanding of the sentiment towards Pfizer across different social media platforms.

# BIBLIOGRAPHY

Aggarwal, C. C. (2017) *Outlier Analysis* (2nd ed.). Springer Cham. https://doi.org/10.1007/978-3-319-47578-3

Ahlgren, M., & Team, W. (2023) *Más de 55 estadísticas, hechos y tendencias de Twitter para 2023*. Website Rating. https://www.websiterating.com/es/research/twitter-statistics/

Ahmad, M., Aftab, S., Ali, I., & Hameed, N. (2017). *Hybrid Tools and Techniques for Sentiment Analysis: A Review*. International Journal of Multidisciplinary Sciences and Engineering, 8(4). https://www.academia.edu/34007886/Hybrid_Tools_and_Techniques_for_Sentiment_Analysis_A_Review

Aisopos, F., Papadakis, G., & Varvarigou, T. (2011) *Sentiment analysis of social media content using N-Gram graphs*. In Proceedings of the 3rd ACM SIGMM International Workshop on Social Media (WSM'11). ACM, New York, NY, 9-14. DOI: http://dx.doi.org/10.1145/2072609.2072614

Baker, M., & Wurgler, J. (2006) *Investor Sentiment and the Cross-Section of Stock Returns.* Journal of Finance, 61(4), 1645-1680. https://doi.org/10.1111/j.1540-6261.2006.00885.x

Baker, M., Wurgler, J., & Yuan, Y. (2012) *Global, local, and contagious investor sentiment*. Journal of Financial Economics, 104(2), 272-287. https://doi.org/10.1016/j.jfineco.2011.11.002

BBC News (2020) *Coronavirus: Putin says vaccine has been approved for use*. BBC News. https://www.bbc.com/news/world-europe-53735718

Bessembinder, H., & Chan, K. (1998) *Market Efficiency and the Returns to Technical Analysis on JSTOR*. https://www.jstor.org/stable/3666289

Bhavitha, B. K., Rodrigues, A. P., & Chiplunkar, N. N. (2017) *Comparative study of machine learning techniques in sentimental analysis,* 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2017, pp. 216-221, doi: 10.1109/ICICCT.2017.7975191

Bird, S., Klein, E., & Loper, E. (2009) *Natural Language Processing with Python*. O'Reilly Media, Inc.. https://www.oreilly.com/library/view/natural-language-processing/9780596803346/

Blume, L. E., Easley, D., & O'Hara, M. (1994) *Market Statistics and Technical Analysis: The Role of Volume.* Journal of Finance, 49(1), 153-181. https://doi.org/10.1111/j.1540-6261.1994.tb04424.x

Bouchrika, I. (2023) *What Is Empirical Research? Definition, Types & Samples*. Research.com. https://research.com/research/what-is-empirical-research

Brandt, P. (2016) *The emergence of the data science profession.* Columbia University. https://doi.org/10.7916/d8bk1ckj

Brock, W. A., Lakonishok, J., & LeBaron, B. (1992) *Simple Technical Trading Rules and the Stochastic Properties of Stock Returns.* Journal of Finance, 47(5), 1731-1764. https://doi.org/10.1111/j.1540-6261.1992.tb04681.x

Brown, G. M., & Cliff, M. T. (2005) *Investor Sentiment and Asset Valuation*. The Journal of Business, 78(2), 405-440. https://doi.org/10.1086/427633

Cao, S., Jiang, W., Wang, J., & Yang, B. (2021) *From Man vs Machine to Man + Machine: The Art and AI of Stock Analyses*. https://doi.org/10.3386/w28800

Chen, J. (2020) *Cumulative Return: definition, calculation, and example*. Investopedia. https://www.investopedia.com/terms/c/cumulativereturn.asp

Cohen, J., Cohen, P., West, S.G., & Aiken, L.S. (2002) *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences* (3rd ed.). Routledge. https://doi.org/10.4324/9780203774441

Cureton, E. E. (1956) *Rank-biserial correlation*. Psychometrika, 21(3), 287–290. https://doi.org/10.1007/bf02289138

DD (2018) *How to download stock prices in R*. Coding Finance. Retrieved July 19, 2023, from https://www.codingfinance.com/post/2018-03-27-download-price/

Deloitte (2018) *Collective intelligence investing: Alpha generation via alternative data brings new risks*. Deloitte Center for Financial Services.

https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-dcfs-im-collective-intel-investing.pdf

Deng, S., Mitsubuchi, T., Shioda, K., Shimada, T., & Sakurai, A. (2011) *Combining Technical Analysis with Sentiment Analysis for Stock Price Prediction*, 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, Sydney, NSW, Australia, 2011, pp. 800-807, doi: 10.1109/DASC.2011.138. https://ieeexplore.ieee.org/abstract/document/6118898/

Ekström, J. (2011) *The Phi-coefficient, the Tetrachoric Correlation Coefficient, and the Pearson-Yule Debate*. https://escholarship.org/uc/item/7qp4604r

Ellyatt, H. (2020) *UK becomes the first to approve Pfizer-BioNTech Covid vaccine, rollout due next week.* CNBC. https://www.cnbc.com/2020/12/02/uk-approves-pfizer-biontech-coronavirus-vaccine-for-use.html

Encyclopedia of Mathematics (2011) *Kendall coefficient of rank correlation*. Encyclopedia of Mathematics. https://encyclopediaofmath.org/index.php?title=Kendall_coefficient_of_rank_correlation&oldid=13189

Esuli, A., & Sebastiani, F. (2006) *SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining*. In Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06), Genoa, Italy. European Language Resources Association (ELRA).

Frigge, M., Hoaglin, D. C., & Iglewicz, B. (1989) *Some Implementations of the Boxplot*. The American Statistician. 43:1. 50-54. DOI: 10.1080/00031305.1989.10475612

Giachanou, A., & Crestani, F. (2016) *Like It or Not. ACM Computing Surveys*, 49(2), 1-41. https://doi.org/10.1145/2938640

Gogtay, N., & Thatte, U. (2017) *Principles of Correlation Analysis.* Journal of the Association of Physicians of India, Vol. 65.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018) *A Survey of Methods for Explaining Black Box Models*. ACM Comput. Surv. 51, 5, Article 93 (September 2019), 42 pages. https://doi.org/10.1145/3236009

Hansen, K. B., & Borch, C. (2022) *Alternative data and sentiment analysis: Prospecting non-standard data in machine learning-driven finance*. Big Data & Society, 9(1). https://doi.org/10.1177/20539517211070701

Hu, M., & Liu, B. (2004) *Mining and summarising customer reviews.* Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004, full paper), Seattle, Washington, USA, Aug 22-25, 2004.

Hutto, C. J., & Gilbert, E. (2014) *VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text*. International Conference on Weblogs and Social Media, 8(1), 216-225. https://doi.org/10.1609/icwsm.v8i1.14550

IMPO (2008) *Law 18.331, Art. 4, numeral D: Personal data: information of any kind related to specific or identifiable natural or legal persons.* https://www.impo.com.uy/bases/leyes/18331-2008

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017) *An Introduction to Statistical Learning: with Applications in R* (7th ed.). Springer.

JMP (2020) *Correlation*. Introduction to Statistics. JMP. Retrieved on May 2, 2023, from https://www.jmp.com/en_ca/statistics-knowledge-portal/what-is-correlation.html

Kendall, M. (1938) *A New Measure of Rank Correlation*. Biometrika, 30, 81-89. https://doi.org/10.1093/biomet/30.1-2.81

Kennan, M. (2010) *How to Calculate Daily Stock Return*. Sapling. Retrieved July 15, 2023, from https://www.sapling.com/6543683/calculate-daily-stock-return

Khurana, D., Koli, A. C., Khatter, K., & Singh, S. (2022) *Natural language processing: state of the art, current trends and challenges*. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-022-13428-4

Kouloumpis, E., Wilson, T., & Moore, J. D. (2011) *Twitter Sentiment Analysis: The Good the Bad and the OMG!* International Conference on Weblogs and Social Media, 5(1), 538-541. https://doi.org/10.1609/icwsm.v5i1.14185

Lao, R. (2017) *Life of Data | Data Science is OSEMN* - Randy Lao - Medium. Medium. https://medium.com/@randylaosat/life-of-data-data-science-is-osemn-f453e1febc10

Lau, C. H. (2019) *5 steps of a Data Science Project Lifecycle* - towards Data Science. Medium. https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492

Li, T., Chamrajnagar, A. S., Fong, X. R., Rizik, N. R., & Fu, F. (2019) *Sentiment-Based Prediction of Alternative Cryptocurrency Price Fluctuations Using Gradient Boosting Tree Model*. Frontiers in Physics, 7. https://doi.org/10.3389/fphy.2019.00098

Liu, B. (2004) *Opinion Mining, Sentiment Analysis, Opinion Extraction*. https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

Manning, C. D., & Schütze, H. (1999) *Foundations of Statistical Natural Language Processing.* The MIT Press. ISBN 978-0-262-13360-9.

Mason, H. (2022). *The OSEMN ("Awesome") Data Science process* (J. Krohn [Super Data Science: ML & AI Podcast with Jon Krohn], Interviewer) [Video]. YouTube. https://www.youtube.com/watch?v=eaGeoOleq1c

Mason, H., & Wiggins, C. (2010) *A taxonomy of data science.* https://introdatasci.dlilab.com/pdf/A_Taxonomy_of_Data_Science.pdf

McCallum, S. (2023) *Elon Musk: Twitter rebrands as X and kills off blue bird logo.* BBC News. https://www.bbc.com/news/business-66284304

Mittal, A. & Goel, A. (2020) *Stock Prediction Using Twitter Sentiment Analysis.* http://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf

Mohammad, S. M. (2011) *NRC Word-Emotion Association Lexicon.* http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

Mohammad, S. M., & Turney, P. D. (2013) *Crowdsourcing a Word-Emotion Association Lexicon*. Institute for Information Technology, National Research Council Canada. http://arxiv.org/pdf/1308.6297.pdf

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011) *Natural language processing: an introduction*. Journal of the American Medical Informatics Association, Volume 18, Issue 5, September 2011, Pages 544–551, https://doi.org/10.1136/amiajnl-2011-000464

Naldi, M. (2019) *A review of sentiment computation methods with R packages*. https://arxiv.org/pdf/1901.08319.pdf

Nasukawa, T., & Yi, J. (2003) *Sentiment analysis. International Conference on Knowledge Capture*. https://doi.org/10.1145/945645.945658

Nielsen, F. Å. (2011) *A new ANEW: Evaluation of a word list for sentiment analysis in microblogs*. arXiv.org. http://arxiv.org/abs/1103.2903

Nigam, A. (2020) *COVID-19: Moderna, Pfizer to include HIV+ volunteers in final stage of vaccine trials.* Republic World. https://www.republicworld.com/world-news/global-event-news/covid-19-moderna-pfizer-to-include-hiv-volunteers-in-final-stage-of.html

NYSE (2022) *Listings Directory for NYSE Stocks*. Retrieved on May 22, 2023, from https://www.nyse.com/listings_directory/stock

Pak, A., & Paroubek, P. (2010) *Twitter as a Corpus for Sentiment Analysis and Opinion Mining*. In Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10) (pp. 1320-1326).

Pang, B., & Lee, L. (2008) *Opinion Mining and Sentiment Analysis*, Foundations and Trends® in Information Retrieval: Vol. 2: No. 1–2, pp 1-135. http://dx.doi.org/10.1561/1500000011

Park, C. H., & Irwin, S. H. (2004) *The Profitability of Technical Analysis: A Review.* Social Science Research Network. https://doi.org/10.2139/ssrn.603481

Pearson, K. (1909) *Determination of the Coefficient of Correlation*. Science, 30(757), 23–25. https://doi.org/10.1126/science.30.757.23

Petrusheva, N., & Jordanoski, I. (2016) *Comparative analysis between the fundamental and technical analysis of stocks*. Journal of Process Management. New Technologies, 4(2), 26–31. https://doi.org/10.5937/jpmnt1602026p

Pfizer (2020) *Pfizer and BioNTech Announce Vaccine Candidate Against COVID-19 Achieved Success in First Interim Analysis from Phase 3 Study.* Pfizer. https://www.pfizer.com/news/press-release/press-release-detail/pfizer-and-biontech-announce-vaccine-candidate-against

Pfizer (2021) *Pfizer and BioNTech Receive First U.S. Authorization for Emergency Use of COVID-19 Vaccine in Adolescents.* Pfizer. https://www.pfizer.com/news/press-release/press-release-detail/pfizer-and-biontech-receive-first-us-authorization

Pfizer (2023) *History.* Pfizer. Retrieved July 19, 2023, from https://www.pfizer.com/about/history.

Python.org (2023) *What is Python? Executive Summary*. Python.org. Retrieved on February 25, 2023, from https://www.python.org/doc/essays/blurb/

R (2022) *What is R?* Retrieved on January 15, 2023, from https://www.r-project.org/about.html

R (n.d.) *R: SPLOM, histograms and correlations for a data matrix*. Retrieved on May 2, 2023, from https://search.r-project.org/CRAN/refmans/psych/html/pairs.panels.html

Raudys, A., Lenčiauskas, V., & Malčius, E. (2013) *Moving Averages for Financial Data Smoothing*. In: Skersys, T., Butleris, R., Butkiene, R. (eds) Information and Software Technologies. ICIST 2013. Communications in Computer and Information Science, vol 403. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-41947-8_4

Saad, S., & Saberi, B. (2017) *Sentiment Analysis or Opinion Mining: A Review*, International Journal on Advanced Science, Engineering and Information Technology, vol. 7, no. 5, pp. 1660-1666, 2017. [Online]. Available: http://dx.doi.org/10.18517/ijaseit.7.5.2137

Saleem, A. (2023) *Scrape Twitter data without Twitter API using SNScrape for timeseries analysis*. Data Science Dojo. https://datasciencedojo.com/blog/scrape-twitter-data-using-sncrape/

Sayce, D. (2022) *The Number of tweets per day in 2022*. David Sayce. https://www.dsayce.com/social-media/tweets-day/

Schweinberger, M. (2022) *Sentiment Analysis in R*. Brisbane: The University of Queensland. https://ladal.edu.au/sentiment.html (Version 2022.10.30)

Shahbaznezhad, H., Dolan, R., & Rashidirad, M. (2021) *The Role of Social Media Content Format and Platform in Users' Engagement Behavior*. Journal of Interactive Marketing, 53(1), 47–65. https://doi.org/10.1016/j.intmar.2020.05.001

Spearman, C. (1961) *The Proof and Measurement of Association Between Two Things*. In J. J. Jenkins & D. G. Paterson (Eds.), Studies in individual differences: The search for intelligence (pp. 45–58). Appleton-Century-Crofts. https://doi.org/10.1037/11491-005

Statista (2022a) *Twitter: number of worldwide users 2019-2024*. https://www.statista.com/statistics/303681/twitter-users-worldwide/

Statista (2022b) *Number of monthly active Twitter users worldwide from 1st quarter 2010 to 1st quarter 2019* Statista. https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

Sukamolson, S. (2007) *Fundamentals of quantitative research*. Language Institute Chulalongkorn University. https://researchgate.net

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011) *Lexicon-Based Methods for Sentiment Analysis*. Computational Linguistics, 37(2), 267-307. https://doi.org/10.1162/coli_a_00049

Taylor, M. P., & Allen, H. (1992) *The use of technical analysis in the foreign exchange market.* Journal of International Money and Finance, 11(3), 304-314. https://doi.org/10.1016/0261-5606(92)90048-3

Tidyverse.org (2023) *tbl_df class — tbl_df-class*. Retrieved on July 4, 2023, from https://tibble.tidyverse.org/reference/tbl_df-class.html

Trinker (2021) GitHub - *trinker/sentimentr: Dictionary based sentiment analysis that considers valence shifters*. GitHub. https://github.com/trinker/sentimentr

Turney, P. D. (2002) *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*. arXiv.org. https://arxiv.org/abs/cs/0212032

Twitter (2023a) *New user FAQ*. Retrieved March 4, 2023, from https://help.twitter.com/en/resources/new-user-faq

Twitter (2023b) *Glossary*. Retrieved March 4, 2023, from https://help.twitter.com/en/resources/glossary

Twitter (2023c) *Twitter Terms of Service*. Twitter Terms of Service. Retrieved May 20, 2023, from https://twitter.com/en/tos

Twitter Developer Platform (2023) *Counting characters*. Docs | Twitter Developer Platform. Retrieved March 2, 2023, from https://developer.twitter.com/en/docs/counting-characters.

U.S. Food and Drug Administration (2021) *FDA Approves First COVID-19 Vaccine*. U.S. Food and Drug Administration. https://www.fda.gov/news-events/press-announcements/fda-approves-first-covid-19-vaccine

Uruguay Presidencia & Agesic (n.d.) *Guía General de Protección de Datos Personales en Uruguay* (Updated with articles 37 to 40 of Law 19.670, from October 15, 2018, and the Decree 64/020, from February 17, 2020). Unidad Reguladora y de Control de Datos Personales. https://www.gub.uy/unidad-reguladora-control-datos-personales/sites/unidad-reguladora-control-datos-personales/files/documentos/publicaciones/Guia%20Protecci%C3%B3n%20de%20Datos%20Personales.pdf

Vigen, T. (2015) *Spurious Correlations* (Gift edition). Hachette Books.

Wang, Y., Guo, J., Yuan, C., & Li, B. (2022) *Sentiment Analysis of Twitter Data*. Applied Sciences. 12. 11775. 10.3390/app122211775.

Wankhade, M., Rao, A. C. S., & Kulkarni, C. A. (2022) *A survey on sentiment analysis methods, applications, and challenges*. Artificial Intelligence Review, 55(7), 5731–5780. https://doi.org/10.1007/s10462-022-10144-1

WHO (2020) *WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020.* https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020

Wickham, H. (2014) *Tidy Data*. Journal of Statistical Software, 59(10). https://doi.org/10.18637/jss.v059.i10

Wiegand, M., Balahur, A., Roth, Klakow, D., & Montoyo, A. (2009) *A survey on the role of negation in sentiment analysis*. https://aclanthology.org/W10-3111.pdf

Wilkinson, L. (2005) *The grammar of graphics*. In Springer eBooks (2nd ed.). Springer New York, NY. https://doi.org/10.1007/0-387-28695-0

Wilson, T., Wiebe, J., & Hoffmann, P. (2005) *Recognizing contextual polarity in phrase level sentiment analysis*. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, Association for Computational Linguistics

Wong, W., Manzur, M., & Chew, B. (2003) *How rewarding is technical analysis? Evidence from Singapore stock market.* Applied Financial Economics, 13(7), 543-551. https://doi.org/10.1080/09603100220000020906

Yahoo (2023) *Yahoo Terms of Service*. Yahoo. Retrieved on March 25, 2023, from https://legal.yahoo.com/us/en/yahoo/terms/otos

Zimbra, D., Abbasi, A., Zeng, D., & Chen, H. (2018) *The State-of-the-Art in Twitter Sentiment Analysis*. ACM Transactions on Management Information Systems, 9(2), 1-29. https://doi.org/10.1145/3185045

# GLOSSARY

*Adjusted R-squared* (*Adjusted $R^2$*): Modified version of $\mathrm{R}^2$ that has been adjusted for the number of predictors in the model. It is always lower than the $\mathrm{R}^2$. The Adjusted $\mathrm{R}^2$ can be helpful for comparing the fit of different regression models that use different numbers of predictor variables.

*Amplifier* (*intensifier*): In a sentence, increases the impact of a polarised word, making the expressed sentiment stronger.

*Bivariate scatter plot*: Visual representation of the relationship between two variables. Shows individual data points as dots on a graph, with one variable on the x-axis and the other on the y-axis.

*Black box model*: In *Data Science* it is a model which's way of reaching outputs is difficult, or usually not possible at all, to interpret. Though it hands an output, it does not show how the output was reached.

*Box plot*: Graphical way of showing statistical data such as *minimum* (value at the bottom), *first quartile* (bottom end of the box), *median* (horizontal line around the middle of the box), *third quartile* (top end of the box) and *maximum* (value at the top). The values marked as independent dots are *outliers*.

*Correlation*: Statistical measure that expresses the extent to which two variables are linearly related (meaning they change together at a constant rate). *Correlation coefficients* range between -1 and 1.

*Cumulative return*: Total change in the price of an asset over a period of time, calculated as the difference between the price at the end of the period and at the beginning of the period, divided between the price at the beginning, and expressed as a percentage.

*Daily return*: Percentage change in the value of an investment or asset from one trading day to the next, calculated by subtracting the opening price from the closing price and then dividing the result of the subtraction between the opening price.

*De-amplifier* (*downtoner*): In a sentence, reduces the impact of a polarised word, making the expressed sentiment weaker.

*Dummy variable*: A variable that can take only two possible values: 1 or 0. For instance, a dummy for Wednesday would take the value “1” if the day is Wednesday and “0” if it is not.

*First quartile*: In a dataset, the value below which 25% of the data falls.

*Follower*: In *Twitter*, someone who has chosen to subscribe to another user’s *Twitter* updates.

*Hashtag*: In Twitter, any word or phrase immediately preceded by the # symbol. It is used in *tweets* to refer to a specific topic being commented on in the *tweet*.

*Histogram*: Graphical representation that organises and displays data by dividing it into intervals or bins and showing the frequency or count of observations within each interval.

*Lag variable*: A variable based on the past values of a time series. For instance, for a given day a lag variable of 1 day for the sentiment index will have the value of the previous day’s index.

*Like*: On *Twitter*, the result of tapping on the heart icon included in a *tweet*. It is a way of expressing appreciation for a *tweet*.

*Mean*: Average value of a dataset, calculated by summing all the values and dividing by the total number of values.

*Median*: Middle value in a dataset when the data are arranged in ascending or descending order.

*Mention*: In a *tweet*, a direct reference to another user. Must have the @ symbol followed by the other user’s username (*@username*).

*Monotonic relationship between variables*: As one variable increases or decreases, the other variable consistently follows the same pattern, either increasing or decreasing. However, the rate or magnitude of change may vary.

*Natural Language Processing* (NLP): Rapidly evolving field concerned with developing algorithms and computational models that enable computers to understand, interpret, and generate human language.

*Negators*: In a sentence, a word that flips the sign of polarised words, making a positive word turn into a negative phrase, or a negative word turn into a positive phrase.

*Outlier*: In statistics, an outlier is an observation that is significantly different from the rest of the observations found in the same population. In this analysis, an outlier is a day in which the number of tweets is notoriously different to the number of tweets of the rest of the days.

*P-value*: Statistical measurement used to validate a hypothesis against observed data. It measures the probability of obtaining the observed results, assuming that the null hypothesis is true. The lower the p-value, the greater the statistical significance of the observed difference. A p-value of 0.05 or lower is generally considered statistically significant.

*Reply*: In *Twitter,* a response to another user's tweet.

*Retweet*: On *Twitter,* a *tweet* that someone has forwarded to his followers.

*R-squared* ($R^2$): Statistical measure in a regression model that represents the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, $R^2$ shows how well the data fit the regression model (the goodness of fit).

*Scatter plot of matrices* (SPLOM): Combined graphs that show *bivariate scatter plots* below the diagonal, *histograms* on the diagonal, and *correlation coefficients* above the diagonal.

*Scatter plot*: Graph that displays the relationship between two variables. With each variable represented in one axis, dots are used to represent individual data points on the two-dimensional coordinate system.

*Sentiment analysis*: Also known as *opinion mining,* is a field of study that involves the use of *natural language processing* (NLP) techniques to extract and identify subjective information from textual data.

*Simple moving average*: Arithmetic average of the last q days of a time series. For instance, the 60-day moving average calculated for a specific day considers that day and the previous 59 days with information, considering then a total of 60 days. This indicator helps identify trends and smooths noise in prices.

*Stopword*: A word that is extremely common and frequently used in a language, such as “a”, “and”, “in”, and “to”. These words are often excluded by computer search engines or when creating a concordance, as they don’t carry significant meaning and can be found in almost every text.

*Tbl_df class:* Special case of dataframe.

*Third quartile*: In a dataset, the value below which 75% of the data falls.

*Tibble*: Colloquial term for the *S3 tbl_df class*.

*Tidy data*: Data that meets the following criteria: each variable has its column, each observation has its row, and each cell contains a single value. In other words, *tidy data* is a well-organized and standardised form of data that allows for easy analysis and interpretation.

*Tweet*: Short message posted on the social media platform *Twitter*.

*Twitter Sentiment Analysis*: Subfield of sentiment analysis that involves analysing the sentiment of tweets to understand public opinion on a particular topic.

*Twitter*: Microblogging social media platform where users can post and interact with short messages, known as “*tweets*”, that are limited to 280 characters.

*User*: On *Twitter*, someone who has registered to use the platform, so can post and read tweets.

*Username*: On *Twitter*, unique name taken by each user, which is an exclusive identifier of the user in the platform.

# Appendix A – Tweets' Download

The Python code in this Appendix was designed to download the complete set of tweets containing the word "Pfizer" within a specified date range (2020-01-01 to 2022-12-31). The code ensures that no tweets are left behind by setting a high maximum daily tweet count (99,000,000).[224] Moreover, the script processes the data day by day, using a loop that iterates through each date within the chosen period, and for each one, it retrieves its tweets and saves them along with additional information into a CSV file.

Using this code, the tweets that include the term "Pfizer" (case-insensitive) across the specified time frame were downloaded, enabling the creation of the tweets database analysed in this document.[225]

```
import snscrape
import os
import pandas as pd
from datetime import timedelta, date

daterange = pd.date_range("2020-01-01", "2022-12-31")

for single_date in daterange:

      tweet_count = 99000000
      text_query = "Pfizer"
      since_date = str(single_date)[0:10]
      until_date = str(pd.to_datetime(single_date)
                   + timedelta(days=1))[0:10]

      # Use the OS library to call CLI commands in Python.

      os.system('snscrape --jsonl --max-results {} --since {} twitter-
      search "{} until:{} "> text-query-
      tweets.json'.format(tweet_count, since_date, text_query,
      until_date))

      # Read the JSON generated from the CLI command above and create
      a Pandas dataframe.

      tweets_df2 = pd.read_json('text-query-tweets.json', lines=True)

      # Export dataframe into a CSV

      tweets_df2.to_csv( ('text-query-
      tweets_'+str(single_date)[0:10]+'.csv'), sep=';', index=False)
```

[224] After downloading the complete tweets database it was verified that no day had reached the 99,000,000 limit, nor was even close.

[225] See Section 4.1 *Tweets Dataset Extraction*.

An intermediate result is generated as a JSON file during each loop iteration in the provided Python code. This file contains all the tweets related to the word “Pfizer” for a specific date. Each tweet is represented as a JSON object, which includes various pieces of information about the tweet. For example, a single tweet in one of these JSON files may look like this:

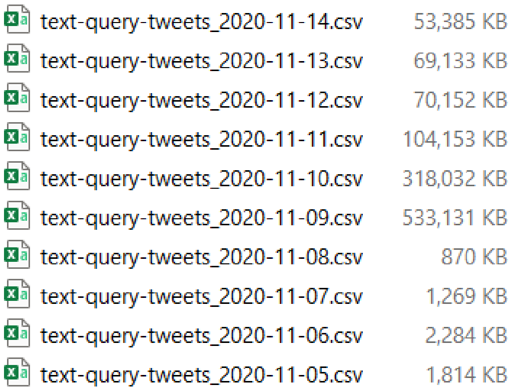
```
{"_type": "snscrape.modules.twitter.Tweet",
"url":
"https://twitter.com/BobWilk73230025/status/1608989835343298561",
"date": "2022-12-31T00:54:04+00:00",
"content": "The \"vaccine'\" does not work.\n\nThe unvaxxed know
it.\nThe vaxxed know it\nBiden knows it \nDoctors know it. \nNurses
know it.\nScientists know it.\nPoliticians know it.\nFauci knows
it.\nPfizer knows it.\nThe media knows it.\n\nYet nine of the ten
above are still doubling down on the lies.",
"renderedContent": "The \"vaccine'\" does not work.\n\nThe unvaxxed
know it.\nThe vaxxed know it\nBiden knows it \nDoctors know it.
\nNurses know it.\nScientists know it.\nPoliticians know it.\nFauci
knows it.\nPfizer knows it.\nThe media knows it.\n\nYet nine of the
ten above are still doubling down on the lies.",
"id": 1608989835343298561,
"user": {"_type": "snscrape.modules.twitter.User",
         "username": "BobWilk73230025",
         "id": 1528966033284993026,
         "displayname": "Bob Wilkinson",
         "description": "Christian,Husband, Father, Conservative,
         continually censored/suspended and banned by big tech and
         it's lies.",
         "rawDescription": "Christian,Husband, Father, Conservative,
         continually censored/suspended and banned by big tech and
         it's lies.",
         "descriptionUrls": null,
         "verified": false,
         "created": "2022-05-24T05:08:15+00:00",
         "followersCount": 978,
         "friendsCount": 1280,
         "statusesCount": 5660,
         "favouritesCount": 2541,
         "listedCount": 1,
         "mediaCount": 637,
         "location": "",
         "protected": false,
         "linkUrl": null,
         "linkTcourl": null,
         "profileImageUrl":
         "https://pbs.twimg.com/profile_images/1528967033576112128/
         SU_SyPrc_normal.jpg",
         "profileBannerUrl":
         "https://pbs.twimg.com/profile_banners/1528966033284993026/
         1653369179",
         "label": null,
         "url": "https://twitter.com/BobWilk73230025"},
"replyCount": 1,
"retweetCount": 3,
"likeCount": 18,
"quoteCount": 0,
"conversationId": 1608989835343298561,
"lang": "en",
"source": "<a href=\"https://mobile.twitter.com\"
rel=\"nofollow\">Twitter Web App</a>",
"sourceUrl": "https://mobile.twitter.com",
"sourceLabel": "Twitter Web App",
"outlinks": null,
"tcooutlinks": null,
"media": null,
"retweetedTweet": null,
"quotedTweet": null,
"inReplyToTweetId": null,
"inReplyToUser": null,
"mentionedUsers": null,
"coordinates": null,
"place": null,
"hashtags": null,
"cashtags": null}
```

The JSON object provides details such as the unique tweet ID, the username of the author, the timestamp of when the tweet was posted, the actual text of the tweet, the number of retweets and likes it received, and any relevant hashtags used in the tweet.

By generating these intermediate JSON files, the code ensures that tweets are processed efficiently and sequentially, allowing for further analysis and aggregation of the data into a comprehensive CSV file containing all the day's tweets and associated information.

During each iteration of the loop, the Python code then converts the JSON file, which contains tweets related to the word “Pfizer” for a specific date, into a dataframe. The data is then exported as a CSV file for that particular day. This process is repeated for each date in the specified range (2020-01-01 to 2022-12-31). As a result, after running the entire loop, a folder is filled with numerous CSV files, each corresponding to one day and containing the whole collection of tweets and associated information for that date. In Figure 45 a subset of these CSV files is displayed, giving an overview of the data gathered from various days.

**Figure 45**
*CSV files – downloaded tweets*



| | |
|---|---|
| text-query-tweets_2020-11-14.csv | 53,385 KB |
| text-query-tweets_2020-11-13.csv | 69,133 KB |
| text-query-tweets_2020-11-12.csv | 70,152 KB |
| text-query-tweets_2020-11-11.csv | 104,153 KB |
| text-query-tweets_2020-11-10.csv | 318,032 KB |
| text-query-tweets_2020-11-09.csv | 533,131 KB |
| text-query-tweets_2020-11-08.csv | 870 KB |
| text-query-tweets_2020-11-07.csv | 1,269 KB |
| text-query-tweets_2020-11-06.csv | 2,284 KB |
| text-query-tweets_2020-11-05.csv | 1,814 KB |

Within each CSV file, a tweet is represented in a structured format containing various attributes that provide valuable insights into the tweet's content, user information, and engagement metrics. The following text corresponds to a randomly chosen tweet in its original CSV format, with some of its features, detailed next, highlighted in bold:

- Link: `https://twitter.com/mcrispinmiller/status/1304568706744414211`
- Date: `2020-09-11 23:53:12+00:00`
- Message: `Pfizer CEO pre-demonizes those who won't get the #COVID19 shot, calling them the "weak link" in #BigPharma's grand defense against "the virus" https://t.co/AcmXV9UYnN`
- Username: `mcrispinmiller`
- User's display name (`'displayname'`): `Mark Crispin Miller`

```
snscrape.modules.twitter.Tweet;https://twitter.com/mcrispinmiller/stat
us/1304568706744414211;2020-09-11 23:53:12+00:00;"Pfizer CEO pre-
demonizes those who won't get the #COVID19 shot, calling them the
""weak link"" in #BigPharma's grand defense against ""the virus""
https://t.co/AcmXV9UYnN";"Pfizer CEO pre-demonizes those who won't get
the #COVID19 shot, calling them the ""weak link"" in #BigPharma's
grand defense against ""the virus""
markcrispinmiller.com/2020/09/pfizer…";1304568706744414211;{'_type':
'snscrape.modules.twitter.User', 'username': 'mcrispinmiller', 'id':
18644548, 'displayname': 'Mark Crispin Miller', 'description':
'Professor of Media Studies at NYU. Founder of News from Underground.
Author of numerous books, including Boxed In and Fooled Again.',
'rawDescription': 'Professor of Media Studies at NYU. Founder of News
from Underground. Author of numerous books, including Boxed In and
Fooled Again.', 'descriptionUrls': None, 'verified': False, 'created':
'2009-01-05T20:30:35+00:00', 'followersCount': 26522, 'friendsCount':
43, 'statusesCount': 11807, 'favouritesCount': 75, 'listedCount':
1035, 'mediaCount': 180, 'location': '', 'protected': False,
'linkUrl': 'https://markcrispinmiller.substack.com', 'linkTcourl':
'https://t.co/zDZ0MYAtw7', 'profileImageUrl':
```

```
'https://pbs.twimg.com/profile_images/69726368/MCMiller_1_normal.jpg',
'profileBannerUrl':
'https://pbs.twimg.com/profile_banners/18644548/1602297032', 'label':
None, 'url':
'https://twitter.com/mcrispinmiller'};0;0;0;0;1304568706744414211;en;"
<a href=""https://mobile.twitter.com"" rel=""nofollow"">Twitter Web
App</a>";https://mobile.twitter.com;Twitter Web
App;['http://markcrispinmiller.com/2020/09/pfizer-ceo-pre-demonizes-
those-who-wont-get-the-covid-19-shot-calling-them-the-weak-link-in-
big-pharmas-grand-defense-against-the-
virus/'];['https://t.co/AcmXV9UYnN'];;;;;;;;;['COVID19', 'BigPharma'];
```

# Appendix B – R Setup

Before running any R code, an essential selection of libraries was (and must be) loaded.[226] The following code snippet was utilised to enable all the essential libraries employed in the forthcoming appendixes, and as a result, it was executed beforehand. Additionally, the language was set to English to ensure that all graphs would be generated in that language, and a working directory was established.

```r
library(dplyr)
library(ggplot2)
library(ggpubr)
library(ggrepel)
library(lubridate)
library(psych)
library(purrr)
library(readr)
library(reshape2)
library(scales)
library(sentimentr)
library(stringr)
library(textdata)
library(tibble)
library(tidyquant)
library(tidyr)
library(tidytext)
library(tm)
library(wordcloud)

Sys.setlocale("LC_ALL", "English")

setwd("C:/Users/Victoria/Desktop/Pfizer Twitter project/")
```

[226] See Section 3.4.2 *R*.

# Appendix C – Pfizer Stock Dataset Download

The extraction of information related to the Pfizer stock was accomplished by executing the following R code.[227] [228]

First deactivate warning messages to prevent them from being displayed in the R console (for cleanliness of the output).

```
options("getSymbols.warning4.0"=FALSE)

options("getSymbols.yahoo.warning"=FALSE)
```

Download PFE data using the *quantmod* package.

```
getSymbols("PFE", from = "2020-01-01",
           to = "2022-12-31",warnings = FALSE,
           auto.assign = TRUE)
```

The *quantmod* package was utilised to obtain comprehensive data concerning Pfizer (PFE) within the specified date range from 2020-01-01 to 2022-12-31, and various charts were generated based on this data to check that data was wholly and correctly downloaded.

```
chart_Series(PFE)

chart_Series(PFE['2022-01/2022-06'])
```

The time series was transformed into a dataframe, and an additional column indicating the day of the week was added for further analysis.

```
PFE_df <- as.data.frame(PFE)

PFE_df <- tibble::rownames_to_column(PFE_df, "date")

PFE_df$week_day <- wday(PFE_df$date)
```

[227] See Section 4.2 *Pfizer Stock Dataset Extraction*.

[228] Before executing this code, run the one in Appendix B – *R Setup*.

## Appendix D – Tweets Dataset Importation

To combine all the CSV files containing tweets into a single unified dataset named "tweets.csv", the R code in this appendix was employed.[229] [230]

The importation of the Tweets Dataset involved several steps:

- Firstly, the code listed all the CSV files present in the directory where the tweets CSV files were downloaded.
- Then, each CSV file was read and its contents were processed.
- Subsequently, a comprehensive dataframe was created, incorporating all the information from the previously read CSV files.
- In the final dataframe design step, a new date column was generated to aid subsequent analyses.
- Finally, the combined dataset was exported to a CSV file named "tweets.csv", allowing for convenient and rapid data importation without the need to rerun the entire script.

After executing the following code, the complete Tweets Dataset can be accessed by simply uploading the tweets.csv file, and this method proves more efficient than re-executing the complete script.

```r
files <- list.files(path = "C:/Users/Victoria/Desktop/Pfizer Twitter
project/Tweets", pattern = "*.csv", full.names = TRUE)

tweets <- files %>%
    set_names() %>%
    map_dfr(.f = read_delim,
            delim = ";",
            .id = "file_name")

tweets <- as.data.frame(tweets)

tweets$date2 <- as.Date(tweets$date, "%Y/%m/%d", tz="UTc")

write.csv(tweets, file='tweets.csv' , row.names = F)
```

[229] See Chapter 5 *Scrub*.

[230] Before executing this code, run the ones in Appendix A – *Tweets' Download* and Appendix B – *R Setup*.

The outcome of this process was a Tweets Dataset comprising 13,077,597 observations, each with 30 variables. The same dataset was also made available in CSV format, taking up 36.9 GB of storage.

For the rest of the appendixes the code in this one must have been run previously. Alternatively, if the tweets.csv file is available, the following two lines of code may be run, producing the same result.

```
tweets <- read.csv('tweets.csv')

tweets_date_time$date2 <- as.Date(tweets_date_time$date2
```

# Appendix E – Tweets Dataset Quality Revision

The R code provided in this appendix served as a critical step in ensuring the quality of the Tweets Dataset.[231]

The initial step involved creating a new dataframe where each row represents a combination of date and hour, starting on January 1, 2020, at 00hs and ending on December 31, 2022, at 23hs. This dataframe served as a reference for all the expected date-hour combinations.

Next, the Tweets Dataset, previously generated using the code from Appendix D – *Tweets Dataset Importation*, was grouped by date and hour. This grouping helped organise and analyse the data on a granular level.

Finally, the two dataframes were combined using a left join, linking the expected date-hour combinations with the actual Tweets Dataset. A filter was applied to retain only those date-hour combinations with zero tweets. The presence of zero tweets for certain date-hour combinations suggests potential errors in data collection. Therefore, further investigation and action were necessary for the dates with multiple hours containing zero tweets. These dates were likely erroneous and were subsequently re-downloaded to ensure data accuracy and completeness.[232]

Create a dataframe with dates and times.

```
dates <- as.data.frame(
            seq(from=as.POSIXct("2020-01-01 00:00:00", tz="UTC"),
                to=as.POSIXct("2022-12-31 23:00:00", tz="UTC"),
                by="hour")
                      )

colnames(dates) <- 'Date'

dates$Time <- hour(dates$Date)

dates$Date <- date(dates$Date)
```

[231] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix D – *Tweets Dataset Importation*.

[232] See Chapter 5 *Scrub*.

Group tweets by dates and times, counting the number of tweets in each.

```
tweets$time <- hour(tweets$date)

tweets_date_time <- tweets %>% group_by(date2,time) %>%
                    summarise(tweets_count = n())
```

Join dataframes and check for days with zero tweets.

```
tweets_dates_check <- left_join(dates, tweets_date_time,
                                by= c("Date"="date2", "Time"="time")
                                )

tweets_dates_check <- subset(tweets_dates_check,
                             is.na(tweets_dates_check$tweets_count))
```

After conducting quality assurance and addressing the dataset's incomplete days, the resulting data's first lines are presented in Table 26.

**Table 26**
*Date-time combinations with zero tweets after quality revision*

| Date | Time | Tweets_count |
|---|---|---|
| 2020-01-04 | 7 | NA |
| 2020-01-05 | 6 | NA |
| 2020-01-06 | 5 | NA |
| 2020-01-11 | 6 | NA |
| 2020-01-12 | 4 | NA |

Despite some combinations of dates and times still having no tweets at all, it was verified that these instances correspond to periods when the number of tweets related to Pfizer was relatively low. Extensive manual checks were performed to ensure the accuracy of the data, confirming that these cases are not errors but rather a reflection of the actual situation: no tweets were published during those specific moments.

This meticulous verification process has validated the authenticity of the dataset, and the absence of tweets during certain periods is a genuine representation of the online activity related to Pfizer. Consequently, the dataset can be relied upon for further analysis and research, providing valuable insights into tweet activity concerning Pfizer over the specified time frame.

## Appendix F – Tweets Dataset: General Numbers Analysis

The code below was utilised to conduct the initial comprehensive Tweets Dataset analysis.[233] [234]

Create a dataframe with the total number of tweets on each date.

```
tweets_date <- tweets %>% group_by(date2) %>%
               summarise(tweets_count = n())
```

Create a dataframe with the number of tweets on each month.

```
tweets_month <- tweets_date %>%
                group_by(year(date2), month(date2)) %>%
                summarise(tweets_month = sum(tweets_count))

tweets_month$month_year <-
                as.Date(paste(tweets_month$'year(date2)',
                tweets_month$'month(date2)', "01",sep="-"))
```

Create a bar plot showing the number of tweets per month.[235]

```
ggplot(tweets_month, aes(y=tweets_month, x=month_year)) +
   geom_bar(stat="identity", fill="steelblue") +
   scale_x_date(breaks=seq(as.Date("2020-01-01"),
                           as.Date("2022-12-31"),
                           by = "1 month"),
                           date_labels="%b %y"
                           ) +
   theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
                          panel.grid.minor = element_blank(),
                          panel.border = element_blank(),
                          axis.line = element_line(colour="black"),
                          axis.ticks = element_line(linewidth = 1)) +
   scale_y_continuous(labels = scales::comma,
                      limits=c(0,1000000), n.breaks=10) +
   xlab("Month - Year") +
   ylab("Total Number of Tweets") +
   ggtitle("Tweets Per Month\n2020-2022") +
   theme(plot.title = element_text(hjust = 0.5))
```

Save the tweets_dates dataframe as CSV for future use.

```
write.csv(tweets_date, file='tweets_date.csv' , row.names = F)
```

[233] See Section 6.1.1 *General Numbers*.

[234] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix D – *Tweets Dataset Importation*.

[235] See Figure 6.

For some of the next appendixes the code in this one must have been run previously. Alternatively, if the tweets_date.csv file is available, the following line of code may be run, producing the same result.

```
tweets_date <- read.csv('tweets_date.csv')
```

## Appendix G – Outlier Identification

The provided R code served the purpose of identifying outliers in the dataset.[236] Also, by applying the boxplot formula, this code allowed for the visualisation and detection of potential outliers in the dataset, and the generation of Figure 7.[237]

Detect outliers and create labels for the ggplot graph.

```
is_outlier <- function(x) {
                 return(x < quantile(x, 0.25) - 1.5 * IQR(x) | x >
                 quantile(x, 0.75) + 1.5 * IQR(x))
                          }

tweets_date <- tweets_date %>% group_by() %>%
               mutate(is_outlier=ifelse(is_outlier(tweets_count),
               paste(date2, " - ",format(tweets_count, big.mark=",")),
               as.numeric(NA)))
```

Identify main statistics: median, Q1, Q3, interquartile range (IQR) and outlier minimum value (Q3 + 1.5 times IQR).

```
median(tweets_date$tweets_count)

quantile(tweets_date$tweets_count, 0.25)

quantile(tweets_date$tweets_count, 0.75)

IQR(tweets_date$tweets_count)

quantile(tweets_date$tweets_count, 0.75) + 1.5 *
         IQR(tweets_date$tweets_count)
```

[236] See Section 6.1.2 *General Trends and Outliers*.

[237] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix D – *Tweets Dataset Importation* (or, instead of the latter, import the tweets.csv file by running the first line of code in this appendix).

Create a boxplot with outliers marked and identified.[238]

```
ggplot(tweets_date) +
      aes(x = "", y = tweets_count) +
      ylab("Tweets Count in a Single Day (#)") +
      ggtitle("Tweets Per Day\n2020-2022") + xlab("") +
      geom_boxplot(fill = "lightblue") +
      theme_bw() +
      scale_y_continuous(labels = scales::comma,
                         limits=c(0,NA), n.breaks=12) +
      scale_x_discrete() +
      theme(plot.title = element_text(hjust = 0.5)) +
      geom_text_repel(aes(label=is_outlier), hjust=-.1, size=3.2,
                      direction = "y", box.padding = 0.18,
                      min.segment.length = unit(0, 'lines')
                     )
```

Extract outliers and identify each one’s date.[239]

```
outliers <- boxplot.stats(tweets_date$tweets_count)$out

outliers_row <- which(tweets_date$tweets_count %in% c(outliers))

tweets_date[outliers_row, c(1:2)] %>% arrange(1)
```

[238] See Figure 7.

[239] See Table 2.

# Appendix H – Organic Tweets and Replies Analysis

The R code in this appendix was designed to analyse and determine the proportion of organic tweets and replies within the dataset.[240] [241]

Keep only organic tweets (remove tweets that are replies).

```
tweets_organic_all <- subset(tweets, is.na(tweets$inReplyToTweetId))
```

Calculate the total number and proportion of organic tweets and replies.

```
Total <- nrow(tweets)

Organic <- nrow(tweets_organic_all)

organic_nonorganic <- data.frame(Type = c("Organic","Reply"),
                                 Value = c(Organic, Total-Organic))

organic_nonorganic <- organic_nonorganic %>%
             group_by() %>%
             mutate(percent = Value / sum(Value)) %>%
             mutate(percent = scales::percent(percent, accuracy=0.1L))
```

Draw a pie chart showing the proportion of organic tweets and replies.[242]

```
ggplot(organic_nonorganic, aes(x="", y=Value, fill=Type))+
      geom_bar(width = 1, stat = "identity") +
      coord_polar("y", start=0) +
      theme_void() +
      geom_text(aes(label = percent),
                position = position_stack(vjust = 0.5) , size=6 ) +
      coord_polar(theta = "y") +
      ggtitle("Proportion of Organic Tweets and Replies\n2020-2022") +
      theme(plot.title = element_text(size=16, hjust=0.5),
            legend.title=element_text(size=15),
            legend.text = element_text(size=13)
      )
```

[240] See Section 6.1.3 *Tweet Types*.

[241] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix D – *Tweets Dataset Importation*.

[242] See Figure 12.

# Appendix I – Tweets' Source Analysis

The provided R code was used to analyse the sources of tweets, getting insights into the platforms or applications used for posting content.[243] [244]

Identify tweets' sources and the number of tweets tweeted from each.

```
tweets_source <- tweets %>%
                 group_by(sourceLabel) %>%
                 summarize(n_tweets=n(),
                           n_likes=sum(likeCount),
                           likes_per_tweet =
                           round((sum(likeCount)/n()),2)
                 ) %>%
                 arrange(-n_tweets)

tweets_source <- as.data.frame(tweets_source)

tweets_source <- tweets_source %>%
                 group_by() %>%
                 mutate(PercentageOfTotal = n_tweets /
                        sum(n_tweets)
                       ) %>%
                 mutate(PercentageOfTotal =
                        scales::percent(PercentageOfTotal,
                        accuracy=0.11L)
                       )

head(tweets_source,10)
```

Calculate totals.[245]

```
tweets_source_total <- tweets_source %>%
                       group_by() %>%
                       summarize(n_tweets= sum(n_tweets),
                                 n_likes = sum(n_likes)
                                ) 	%>%
                       mutate(likes_per_tweet = n_likes/n_tweets)
```

[243] See Section 6.1.4 *Tweet Sources*.

[244] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix D – *Tweets Dataset Importation*.

[245] See Table 3.

# Appendix J – Tweets' Languages Analysis

The given R code was used to analyse the languages used in tweets and determine the distribution of languages in which content was published.[246] [247]

Group tweets by language and count how many tweets there are in each language.

```
tweets_languages <- tweets[c(14,4)] %>%
                    group_by(lang) %>%
                    summarise(amount = n())

tweets_languages <- tweets_languages[order(-tweets_languages$amount),]
```

Include languages with less than 200,000 tweets in the "Others" group.

```
tweets_languages$Language <- ifelse(tweets_languages$amount > 200000,
                                    tweets_languages$lang, "Others")

tweets_languages_2 <- tweets_languages %>%
                      group_by(Language) %>%
                      summarise(TotalTweets = sum(amount))
```

Replace language names' abbreviations with complete language names.

```
tweets_languages_2$Language <-
      ifelse(tweets_languages_2$Language == "en", "English",
      ifelse(tweets_languages_2$Language == "de", "German",
      ifelse(tweets_languages_2$Language == "es", "Spanish",
      ifelse(tweets_languages_2$Language == "fr", "French",
      ifelse(tweets_languages_2$Language == "it", "Italian",
      ifelse(tweets_languages_2$Language == "pt", "Portuguese",
      ifelse(tweets_languages_2$Language == "nl", "Dutch",
             tweets_languages_2$Language )))))))
```

Calculate the percentage of tweets in each language out of the total number of tweets.

```
tweets_languages_2 <- tweets_languages_2 %>%
                      group_by() %>%
                      mutate(percent_of_total =
                             TotalTweets/sum(TotalTweets)) %>%
                      mutate(percent_of_total =
                             scales::percent(percent_of_total,
                                             accuracy=0.1L)
                            )
```

[246] See Section 6.1.5 *Languages*.

[247] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix D – *Tweets Dataset Importation*.

Create a pie chart showing the previously calculated proportions.[248]

```
ggplot(tweets_languages_2,
       aes(x="", y=TotalTweets, fill=reorder(Language, -TotalTweets))
      ) +
       geom_bar(width = 1, stat = "identity") +
       coord_polar("y", start=0) +
       labs(fill = "Language") +
       theme_void() +
       ggtitle("Languages of Tweets\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       geom_text_repel(aes(label = format(percent_of_total)),
                       position = position_stack(vjust = 0.5),
                       box.padding=0.2, size=4.5) +
       theme(plot.title = element_text(size=16),
             legend.title=element_text(size=15),
             legend.text = element_text(size=13)
            )
```

[248] See Figure 13.

# Appendix K – Tweeting Users Analysis

The provided R code was used to analyse the distribution of tweets among the different users in the dataset.[249] [250]

Keep only the date and user columns.

```
tweets_users <- tweets[c(8,4)]
```

Extract the username from the user column, as the column has a lot of information apart from the username.

```
tweets_users$user <- sub(".*, 'username': '", "", tweets_users$user)

tweets_users$user <- sub("', 'id.* ", "", tweets_users$user)
```

Group the dataframe by username and count each user's tweets.[251]

```
tweets_users_2 <- tweets_users %>%
                  group_by(user) %>%
                  summarise(amount = n())

tweets_users_2 <- arrange(tweets_users_2, by = -amount)
```

Group the dataframe by number of tweets tweeted to see how many users tweeted each number of times.

```
tweets_users_3 <- tweets_users_2 %>%
                  group_by(amount) %>%
                  summarise(number_users = n())
```

Creates three graphs to show results: one with the number of users with less than 20 tweets, another one with the number of users with a number of tweets between 20 and 200, and the last one with those users with over 200 tweets.

[249] See Section 6.1.6 *Users*.

[250] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix D – *Tweets Dataset Importation*.

[251] See Table 4.

First graph, showing the number of users with less than 20 tweets.

```
tweets_users_4 <- filter(tweets_users_2, amount<20)

p1 <- ggplot(tweets_users_4, aes(x = amount)) +
             geom_histogram(binwidth = 1, color = "steelblue",
                            fill = "lightblue") +
             ggtitle("Tweet Frequency Analysis: Users with < 20
                     Tweets") +
             theme(plot.title = element_text(hjust = 0.5, size = 14),
                   axis.line = element_line(color = "black"),
                   axis.title = element_text(size=12),
                   axis.text = element_text(size=12)
                 )+
             labs(x="Tweets Per User (#)", y= "Users (#)") +
             scale_y_continuous(labels = scales::comma, expand =
                                expansion(mult = c(0, 0.08))
                               ) +
             scale_x_continuous(labels = scales::comma, expand =
                                expansion(mult = c(0, 0.08)))
```

Second graph, showing the number of users with between 20 and 200 tweets.

```
tweets_users_5 <- filter(tweets_users_2, amount>19, amount<201)

p2 <- ggplot(tweets_users_5, aes(x=amount)) +
             geom_histogram(binwidth=1, color="steelblue",
                            fill="lightblue") +
             ggtitle("Tweet Frequency Analysis: Users with 20 to 200
                     Tweets") +
             theme(plot.title = element_text(hjust = 0.5, size = 14),
                   axis.line = element_line(color = "black"),
                   axis.title = element_text(size=12),
                   axis.text = element_text(size=12)
                  )+
             labs(x="Tweets Per User (#)", y= "Users (#)") +
             scale_y_continuous(labels = scales::comma, expand =
                                expansion(mult = c(0, 0.08))
                               ) +
             scale_x_continuous(labels = scales::comma, expand =
                                expansion(mult = c(0, 0.08)),
                                limits = c(0, NA))
```

Third graph, showing the number of users with over 200 tweets.

```
tweets_users_6 <- filter(tweets_users_2, amount>200)

p3 <- ggplot(tweets_users_6, aes(x=amount)) +
             geom_histogram(binwidth=1, color="steelblue",
                            fill="lightblue") +
        ggtitle("Tweet Frequency Analysis: Users with > 200 Tweets") +
  theme(plot.title = element_text(hjust = 0.5, size = 14), axis.line =
element_line(color = "black"),
        axis.title = element_text(size=12),
        axis.text = element_text(size=12) )+
        labs(x="Tweets Per User (#)", y= "Users (#)") +
        scale_y_continuous(labels = scales::comma, expand =
                           expansion(mult = c(0, 0.08))
                          ) +
        scale_x_continuous(labels = scales::comma, expand =
                           expansion(mult = c(0, 0.08)),
                           limits = c(0, NA))
```

Present the three graphs (p1, p2 and p3) together.[252]

```
gg_plot <- ggarrange(p1, p2, p3,
                     ncol = 1,
                     heights = c(5, 5, 5),
                     align = "v"
                    )
```

Show a list of the users with more tweets.

```
head(tweets_users_2)
```

[252] See Figure 14.

# Appendix L – Most Frequent Words Determination

The R code in this appendix was used to identify the most frequently used words in English tweets.[253] [254]

Filter only English language tweets and clean the original dataset: remove hyperlinks, mentions (@) and punctuation from the tweets' content.

```
tweets_words <- tweets

tweets_words <- filter(tweets, lang == "en")

tweets_words$content <- gsub("https\\S*", "", tweets_words$content)
tweets_words$content <- gsub("@\\S*", "", tweets_words$content)
tweets_words$content <- gsub("amp", "", tweets_words$content)
tweets_words$content <- gsub("[\r\n]", "", tweets_words$content)
tweets_words$content <- gsub("[[:punct:]]", "", tweets_words$content)
```

From the dataset keep only the list of words included in the tweets' content and remove both stopwords and the term "Pfizer".

```
tweets_words_2 <- tweets_words %>%
                  select(content) %>%
                  unnest_tokens(word, content) %>%
                  anti_join(stop_words) %>%
                  filter(!str_detect(word, "pfizer"))
```

Creates a bar chart showing the thirty words with the highest frequency in the dataset.[255]

```
tweets_words_2 %>% count(word, sort = TRUE) %>%
                   top_n(30) %>%
                   mutate(word = reorder(word, n)) %>%
                   ggplot(aes(x = word, y = n)) +
                   geom_col(fill="steelblue") +
                   xlab(NULL) +
                   coord_flip() +
                   labs(y = "Count",
                        x = "Unique words",
                        title = "Most frequent words in tweets in
                                 English related to Pfizer",
                        subtitle = "Stopwords and \"Pfizer\" removed
                                    from the list") +
                   scale_y_continuous(labels = scales::comma, expand =
                                      expansion(mult = c(0, 0.08)),
                                      limits = c(0, NA)
                                     )
```

[253] See Section 6.1.7 *Most Frequent Words in Tweets in English*.

[254] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix D – *Tweets Dataset Importation*.

[255] See Figure 16.

## Appendix M – Pfizer Stock Descriptive Analysis

The R code in this appendix was used to perform the descriptive analysis of the Pfizer stock values. This analysis involved exploring and summarising key statistical measures and characteristics of the stock's price series.[256] [257]

The first part of the code creates a plot showing the PFE price series for the period.[258]

```
ggplot(PFE, aes(x = index(PFE), y = PFE[, 4])) +
       geom_line(color = "steelblue", size=0.7) +
       ggtitle("PFE Price Series") +
       xlab("Date") +
       ylab("Price") +
       theme(plot.title = element_text(hjust = 0.5)) +
       scale_x_date(breaks = seq(as.Date("2020-01-01"),
                                 as.Date("2022-12-31"),
                                 by = "1 month"),
                    date_labels = "%b %y",
                    expand = expansion(mult = c(0.015, 0.015))
                   ) +
       theme(axis.text.x = element_text(angle = 90,
                                        vjust = 0.5, hjust = 1)
                                       ) +
       scale_y_continuous(limits = c(0, NA))
```

Convert data to dataframe format.

```
PFE_df <- as.data.frame(PFE)
```

Make the dataframe's row names become the first column of the dataframe instead.

```
PFE_df <- tibble::rownames_to_column(PFE_df, "date")
```

Create a new column with the weekday number, being Sunday day 1.

```
PFE_df$week_day <- wday(PFE_df$date)
```

Calculate the descriptive statistics.[259]

```
summary(PFE_df[2:7])
```

[256] See Section 6.2 *Pfizer Stock Descriptive Analysis*.

[257] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix C – *Pfizer Stock Dataset Download*.

[258] See Figure 17.

[259] See Figure 18.

Calculate moving averages in a new dataset called “pfe_mm”.

```
pfe_mm <- PFE

pfe_mm10 <- rollmean(pfe_mm[,6], 10,
                     fill = list(NA, NULL, NA),
                     align = "right")

pfe_mm30 <- rollmean(pfe_mm[,6], 30,
                     fill = list(NA, NULL, NA),
                     align = "right")

pfe_mm60 <- rollmean(pfe_mm[,6], 60,
                     fill = list(NA, NULL, NA),
                     align = "right")

pfe_mm90 <- rollmean(pfe_mm[,6], 90,
                     fill = list(NA, NULL, NA),
                     align = "right")

pfe_mm$mm10 <- coredata(pfe_mm10)

pfe_mm$mm30 <- coredata(pfe_mm30)

pfe_mm$mm60 <- coredata(pfe_mm60)

pfe_mm$mm90 <- coredata(pfe_mm90)
```

Create a line graph with PFE price and moving averages (30 days and 90 days).[260]

```
ggplot(pfe_mm, aes(x = index(pfe_mm))) +
       geom_line(aes(y = pfe_mm[,6], color = "PFE")) +
       ggtitle("PFE Price Series") +
       geom_line(aes(y = pfe_mm$mm30, color = "30 days MA")) +
       geom_line(aes(y = pfe_mm$mm90, color = "90 days MA")) +
       xlab("Date") + ylab("Price") +
       theme(plot.title = element_text(hjust = 0.5),
             panel.border = element_blank()) +
       scale_colour_manual("Series",
                           values=c("PFE"="gray40",
                                    "30 days MA"="firebrick4",
                                    "90 days MA"="darkcyan")
                                   ) +
       scale_x_date(breaks=seq(as.Date("2020-01-01"),
                               as.Date("2023-01-01"),
                               by = "1 month"),
                               date_labels="%b %y",
                               expand = expansion(mult = c(0.015,
                                                           0.015))
                   ) +
       theme(axis.text.x = element_text(angle = 90,
                                        vjust = 0.5,
                                        hjust=1)) +
       scale_y_continuous(limits = c(0, NA))
```

[260] See Figure 19.

Create a chart with PFE's price series and traded volume.[261]

```
chartSeries(PFE, theme = 'white')
```

Calculate daily returns.

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

pfe_daily_returns <- PFE_df %>%
                     tq_transmute(select = PFE.Adjusted,
                                  mutate_fun = periodReturn,
                                  period = "daily",
                                  col_rename = "pfe_returns")
```

Create a line chart for daily returns.[262]

```
pfe_daily_returns %>% ggplot(aes(x = date, y = pfe_returns)) +
                      geom_line(color="steelblue") +
                      theme_classic() +
                      labs(x = "Date", y = "Daily returns") +
                      ggtitle("PFE Daily Returns") +
                      scale_x_date(date_breaks = "years",
                                   date_labels = "%Y") +
                      scale_y_continuous(breaks = seq(-0.5,0.6,0.05),
                                         labels = scales::percent) +
                      geom_hline(yintercept = 0,
                                 color = "black",
                                 linetype = "dashed",
                                 size=0.5) +
                      geom_hline(aes(yintercept=mean(pfe_returns)),
                                 color="black",
                                 linetype="dotted",
                                 size=0.5)
```

[261] See Figure 20.

[262] See Figure 21.

Create a histogram showing PFE's daily returns.[263]

```
pfe_daily_returns %>% ggplot(aes(x = pfe_returns)) +
                      geom_histogram(binwidth = 0.005,
                                     color="steelblue",
                                     fill="lightblue") +
                      theme_classic() +
                      labs(x = "Daily returns", y = "Frequency") +
                      ggtitle("PFE Daily Returns Histogram") +
                      theme(plot.title = element_text(hjust = 0.5)) +
                      scale_x_continuous(n.breaks=10,
                                         labels = scales::percent) +
                      scale_y_continuous(expand = expansion(mult =
                                         c(0, 0.05))) +
                      geom_vline(xintercept = 0, linetype = "dashed",
                                 color = "black", size=0.5) +
                      geom_vline(aes(xintercept=mean(pfe_returns)),
                                 color="black",
                                 linetype="dotted",
                                 size=0.5)
```

Calculate monthly returns.

```
pfe_monthly_returns <- PFE_df %>%
                       tq_transmute(select = PFE.Adjusted,
                                    mutate_fun = periodReturn,
                                    period = "monthly",
                                    col_rename = "pfe_returns")
```

Create a bar chart showing PFE's monthly returns.[264]

```
pfe_monthly_returns %>% ggplot(aes(x = date, y = pfe_returns)) +
                        geom_bar(stat = "identity",
                                 color="steelblue",
                                 fill="lightblue") +
                        theme_classic() +
                        labs(x = "Date", y = "Monthly returns") +
                        ggtitle("PFE Monthly Returns") +
                        theme(plot.title = element_text(hjust = 0.5)) +
                        geom_hline(yintercept = 0) +
                        scale_y_continuous(breaks = seq(-0.6,0.8,0.1),
                                           labels = scales::percent) +
                        scale_x_date(breaks =
                                     seq(as.Date("2019-12-31"),
                                         as.Date("2022-12-31"),
                                         by = "1 month"),
                                         date_labels = "%b %y",
                                         expand = expansion(mult =
                                                  c(0.015, 0.015))
                                    ) +
                        theme(axis.text.x = element_text(angle = 90,
                                                         vjust = 0.5,
                                                         hjust = 1))
```

[263] See Figure 22.

[264] See Figure 23.

Calculate the cumulative returns for PFE stock by using the cumprod() function.

```
pfe_cum_returns <- pfe_daily_returns %>%
                   mutate(cr = cumprod(1 + pfe_returns)) %>%
                   mutate(cumulative_returns = cr - 1)
```

Create a line graph showing the calculated cumulative returns.[265]

```
pfe_cum_returns %>% ggplot(aes(x = date, y = cumulative_returns)) +
                    geom_line(color="steelblue", size=0.7) +
                    theme_classic() +
                    labs(x = "Date", y = "Cumulative Returns") +
                    ggtitle("PFE Cumulative Returns\n2020-2022",
                            subtitle = "$1 investment in 2020 grew to
                                       $1.54 by the end of 2022") +
                    theme(plot.title = element_text(hjust = 0.5),
                          plot.subtitle = element_text(hjust = 0.5)) +
                    scale_y_continuous(labels = scales::percent) +
                    scale_x_date(breaks = seq(as.Date("2020-01-01"),
                                              as.Date("2023-01-01"),
                                              by = "1 month"),
                                              date_labels = "%b %y",
                                              expand = expansion(mult =
                                                       c(0.015,
                                                         0.015)
                                              )
                                 ) +
                    theme(axis.text.x = element_text(angle = 90,
                                                     vjust = 0.5,
                                                     hjust = 1)
                                                   )
```

Calculate the mean daily return.

```
pfe_daily_returns %>% select(pfe_returns) %>%
                      .[[1]] %>%
                      mean(na.rm = TRUE)
```

[265] See Figure 24.

## Appendix N – Subset of Organic English Tweets

The provided R code was used to prepare the database for the upcoming sentiment analysis by creating three separate dataframes, each serving a specific purpose:[266] [267]

- *tweets_organic_en.csv* containing all tweets that are organic and in English,
- *tweets_date_organic_english.csv* providing the count of organic English tweets for each day within the studied period, and
- *tweets_sentiment.csv* including one line for each token in each tweet, along with the corresponding tweet number based on the original order of the tweets_organic_en database.

First subset tweets, which are organic and in English, and export to CSV.

```
tweets_organic_en <- subset(tweets,
                            is.na(tweets$inReplyToTweetId) &
                            tweets$lang == "en"
                           )

write.csv(tweets_organic_en,
          file='tweets_organic_en.csv',
          row.names = F)
```

Create a dataframe with the number of organic English tweets each day, and export it to CSV.

```
tweets_date_organic_english <- tweets_organic_en %>%
                               group_by(date2) %>%
                               summarise(tweets_count = n())

colnames(tweets_date_organic_english)[1] <- "date"

tweets_date_organic_english$date <-
      as.Date(tweets_date_organic_english$date)

write.csv(tweets_date_organic_english,
      file='tweets_date_organic_english.csv',
      row.names = F)
```

[266] See Chapter 7 *Explore – Sentiment Analysis*.

[267] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix D – *Tweets Dataset Importation*.

Create a dataframe where each row represents a word found in a tweet, along with the corresponding tweet number in which that word is found.

```
tweets_sentiment <- tibble(line = 1:nrow(tweets_organic_en),
                   text = tweets_organic_en[,5])

tweets_sentiment <- tweets_sentiment %>% unnest_tokens(word, text)

write.csv(tweets_sentiment, file='tweets_sentiment.csv',
      row.names = F)
```

For the rest of the appendixes the code in this one must have been run previously. Alternatively, if the tweets_organic_en.csv, tweets_date_organic_english.csv and tweets_sentiment.csv files are available, the following three lines of code may be run, producing the same result.

```
tweets_organic_en.csv <- read.csv('tweets.csv')

tweets_date_organic_english.csv <- read.csv('tweets.csv')

tweets_sentiment.csv <- read.csv('tweets.csv')
```

## Appendix O – Bing Positivity Index Calculation

The provided R code was used to calculate and visualise the Bing Positivity Index by performing an inner join between the tokens (words) in each tweet and the lexicons included in Bing. This process allowed the determination of the overall positivity of the tweets based on the words they contain, and the result was plotted to provide insights into the sentiment distribution across the dataset.[268] [269]

Calculate the Bing positivity index for each tweet.

```
get_sentiments("bing")

tweets_sentiment_bing <- tweets_sentiment %>%
                         inner_join(get_sentiments("bing")) %>%
                         group_by(line) %>%
                         summarise(positive = sum(sentiment ==
                                                  "positive"),
                                   negative = sum(sentiment ==
                                                  "negative"))

tweets_sentiment_bing$positivity_index <-
      tweets_sentiment_bing$positive - tweets_sentiment_bing$negative
```

Integrate the positivity index of each tweet with the original dataset to obtain the respective tweet dates and compute the positivity index for each date.

```
tweets_organic_en$line <- seq.int(nrow(tweets_organic_en))

tweets_organic_en_2 <- tweets_organic_en %>%
                       left_join(tweets_sentiment_bing)

tweets_organic_en_2 <- tweets_organic_en_2[,c("date", "lang",
                                              "positivity_index")]

tweets_organic_en_2[is.na(tweets_organic_en_2)] <- 0

colnames(tweets_organic_en_2)[1] <- "date"

tweets_organic_en_2$date <- as.Date(tweets_organic_en_2$date)

tweets_organic_en_2 <- tweets_organic_en_2 %>%
                       group_by(date) %>%
                       summarise(
                         positivity_index_bing = sum(positivity_index),
                         bing_pos = sum(positivity_index > 0),
                         bing_neg = sum(positivity_index < 0),
                         bing_zero = sum(positivity_index == 0))
```

[268] See Section 7.1 *Sentiment Analysis with Bing*.

[269] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix N – *Subset of Organic English*.

Add to the dataframe the number of tweets per day that are organic and in English to then calculate an average positivity index per day.

```
tweets_organic_en_2 <- tweets_date_organic_english %>%
                       left_join(tweets_organic_en_2)

tweets_organic_en_2$positivity_index_avg <-
                           tweets_organic_en_2$positivity_index_bing /
                           tweets_organic_en_2$tweets_count
```

Plot the positivity index per day.[270]

```
ggplot(tweets_organic_en_2, aes(x=date, y=positivity_index_avg)) +
       geom_line(color = "steelblue") +
       xlab("") +
       ggtitle("Bing Positivity Index per Day\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(y= "Positivity Index") +
       scale_x_date(breaks = seq(as.Date("2020-01-01"),
                                 as.Date("2023-01-01"),
                                 by = "1 month"),
                                 date_labels = "%b %y",
                                 expand = expansion(mult = c(0.015,
                                                             0.015))
                   ) +
       theme(axis.text.x = element_text(angle = 90,
                                        vjust = 0.5,
                                        hjust = 1))
```

Create a monthly positivity index.

```
tweets_sample_organic_language_month <- tweets_organic_en_2 %>%
          group_by( year(date),month(date)) %>%
          summarise(positivity_index_avg = mean(positivity_index_avg),
                    bing_pos = sum(bing_pos),
                    bing_neg = sum(bing_neg),
                    bing_zero = sum(bing_zero)
                   )

tweets_sample_organic_language_month$year_month <-
   as.yearmon(paste(tweets_sample_organic_language_month$'year(date)',
                    tweets_sample_organic_language_month$'month(date)'
                   ),
              "%Y %m")
```

[270] See Figure 25.

Draw a line graph showing the monthly positivity index.[271]

```
ggplot(tweets_sample_organic_language_month,
       aes(x=year_month, y=positivity_index_avg)) +
       geom_line(color="steelblue",size=0.8) +
       xlab("") +
       scale_x_yearmon(
          labels=date_format("%b %y"),
          breaks = seq(from =
                 min(tweets_sample_organic_language_month$year_month),
                       to =
                 max(tweets_sample_organic_language_month$year_month),
                       by = 1/12),
          expand = expansion(mult = c(0.015, 0.015))) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("Bing Positivity Index Per Month\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(x="", y= "Positivity Index")
```

To draw stacked bar charts showing the monthly positivity index, first convert the dataframe to long format and reorder the levels of the sentiment variable.

```
tweets_sample_organic_language_month_long <-
                    pivot_longer(tweets_sample_organic_language_month,
                    cols = c("bing_pos", "bing_neg", "bing_zero"),
                    names_to = "Sentiment",
                    values_to = "Number of Tweets")

tweets_sample_organic_language_month_long <-
                    tweets_sample_organic_language_month_long %>%
                    mutate(Sentiment = recode(Sentiment,
                           "bing_pos" = "Positive",
                           "bing_neg" = "Negative",
                           "bing_zero" = "Neutral")
                          )

tweets_sample_organic_language_month_long$Sentiment <-
           factor(tweets_sample_organic_language_month_long$Sentiment,
                  levels = c("Negative", "Neutral", "Positive")
                 )
```

Create a stacked bar chart.[272]

```
ggplot(tweets_sample_organic_language_month_long,
       aes(x=year_month, y=`Number of Tweets`, fill=Sentiment)) +
       geom_col() +
       scale_fill_manual(values=c("Positive"="forestgreen",
                                  "Negative"="indianred",
                                  "Neutral"="darkgrey")
                        ) +
       xlab("") +
       ylab("Number of Tweets") +
```

[271] See Figure 26.

[272] See Figure 27.

```
       scale_x_yearmon(labels=date_format("%b %y"),
                       breaks = seq(from =
            min(tweets_sample_organic_language_month_long$year_month),
                                    to =
            max(tweets_sample_organic_language_month_long$year_month),
                                    by = 1/12),
                       expand = expansion(mult = c(0.015, 0.015))
                      ) +
       scale_y_continuous(labels = scales::comma,
                          expand = expansion(mult = c(0.015, 0.04))
                         ) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("Bing Tweets by Sentiment per Month\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(fill="Sentiment") +
       guides(fill=guide_legend(title=NULL))
```

Create a 100% stacked bar chart.[273]

```
ggplot(tweets_sample_organic_language_month_long,
       aes(x=year_month, y=`Number of Tweets`, fill=Sentiment)) +
       geom_col(position = position_fill(reverse = FALSE)) +
       scale_fill_manual(values=c("Positive"="forestgreen",
                                  "Negative"="indianred",
                                  "Neutral"="darkgrey")
                        ) +
       xlab("") +
       ylab("Percentage of Tweets") +
       scale_x_yearmon(labels=date_format("%b %y"),
                  breaks = seq(from =
            min(tweets_sample_organic_language_month_long$year_month),
                               to =
            max(tweets_sample_organic_language_month_long$year_month),
                               by = 1/12),
                  expand = expansion(mult = c(0.015, 0.015))
                      ) +
       scale_y_continuous(labels = scales::percent_format(),
                          expand = expansion(mult = c(0.015, 0.015))
                         ) +
       theme(axis.text.y = element_text(size = 12)) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("Bing Tweets by Sentiment per Month (%)\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(fill="Sentiment") +
       guides(fill=guide_legend(title=NULL))
```

[273] See Figure 28.

# Appendix P – AFINN Positivity Index Calculation

The given R code was used to calculate and visualise the AFINN Positivity Index by performing an inner join between the tokens (words) in each tweet and the lexicons included in AFINN. This process allowed the determination of the overall positivity of the tweets based on the words they contain, and the result was plotted to provide insights into the sentiment distribution across the dataset.[274] [275]

Calculate the AFINN positivity index for each tweet.

```
tweets_sentiment_afinn <- tweets_sentiment %>%
                          inner_join(get_sentiments("afinn")) %>%
                          group_by(line) %>%
                          summarise(sentiment_afinn = sum(value))
```

Join the positivity index of each tweet to the original dataset to get each tweet's date and calculate each date's positivity index.

```
tweets_organic_en$line <- seq.int(nrow(tweets_organic_en))

tweets_organic_en_3 <- tweets_organic_en %>%
                       left_join(tweets_sentiment_afinn)

tweets_organic_en_3 <-tweets_organic_en_3[,c("date", "lang",
                                             "sentiment_afinn")]

tweets_organic_en_3[is.na(tweets_organic_en_3)] <- 0

tweets_organic_en_3$date <- as.Date(tweets_organic_en_3$date)

tweets_organic_en_3 <- tweets_organic_en_3 %>%
                      group_by(date) %>%
                      summarise(sent_afinn = sum(sentiment_afinn),
                                afinn_pos = sum(sentiment_afinn > 0),
                                afinn_neg = sum(sentiment_afinn < 0),
                                afinn_zero = sum(sentiment_afinn == 0)
                               )
```

[274] See Section 7.2 *Sentiment Analysis with AFINN*.

[275] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix N – *Subset of Organic English*.

Add to the dataframe the number of tweets per day that are organic and in English to then calculate an average positivity index per day.

```
tweets_organic_en_3 <- tweets_date_organic_english %>%
                       left_join(tweets_organic_en_3)

tweets_organic_en_3$sentiment_afinn_avg <-
                                     tweets_organic_en_3$sent_afinn /
                                     tweets_organic_en_3$tweets_count
```

Plot the positivity index per day.[276]

```
ggplot(tweets_organic_en_3, aes(x=date, y=sentiment_afinn_avg)) +
       geom_line(color = "steelblue") + xlab("") +
       ggtitle("AFINN Positivity Index per Day\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(y= "Positivity Index") +
       scale_x_date(breaks = seq(as.Date("2020-01-01"),
                                 as.Date("2023-01-01"),
                                 by = "1 month"),
                                 date_labels = "%b %y",
                                 expand = expansion(mult = c(0.015,
                                                             0.015))
                   ) +
       theme(axis.text.x = element_text(angle = 90,
                                        vjust = 0.5,
                                        hjust = 1))
```

Create a monthly positivity index.

```
tweets_sample_organic_language_month_AFINN <- tweets_organic_en_3 %>%
            group_by( year(date),month(date)) %>%
            summarise(sentiment_afinn_avg = mean(sentiment_afinn_avg),
                      afinn_pos = sum(afinn_pos),
                      afinn_neg = sum(afinn_neg),
                      afinn_zero = sum(afinn_zero))

tweets_sample_organic_language_month_AFINN$year_month <-
      as.yearmon(paste(tweets_sample_organic_language_month_AFINN$'yea
      r(date)',
      tweets_sample_organic_language_month_AFINN$'month(date)'),
      "%Y %m")
```

[276] See Figure 29.

Draw a line graph showing the monthly positivity index.[277]

```
ggplot(tweets_sample_organic_language_month_AFINN,
       aes(x=year_month, y=sentiment_afinn_avg)) +
       geom_line(color="steelblue",size=0.8) + xlab("") +
       scale_x_yearmon(labels=date_format("%b %y"),
                       breaks = seq(from =
           min(tweets_sample_organic_language_month_AFINN$year_month),
                                    to =
           max(tweets_sample_organic_language_month_AFINN$year_month),
                                    by = 1/12),
                       expand = expansion(mult = c(0.015, 0.015))
                      ) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("AFINN Positivity Index per Month\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(x="", y= "Positivity Index")
```

To draw stacked bar charts showing the monthly positivity index, first convert the dataframe to long format and reorder the levels of the sentiment variable.

```
tweets_sample_organic_language_month_AFINN_long <-
              pivot_longer(tweets_sample_organic_language_month_AFINN,
              cols = c("afinn_pos", "afinn_neg", "afinn_zero"),
              names_to = "Sentiment",
              values_to = "Number of Tweets")

tweets_sample_organic_language_month_AFINN_long <-
                   tweets_sample_organic_language_month_AFINN_long %>%
                   mutate(Sentiment = recode(Sentiment,
                          "afinn_pos" = "Positive",
                          "afinn_neg" = "Negative",
                          "afinn_zero" = "Neutral"))

tweets_sample_organic_language_month_AFINN_long$Sentiment <-
     factor(tweets_sample_organic_language_month_AFINN_long$Sentiment,
            levels = c("Negative", "Neutral", "Positive"))
```

Create a stacked bar chart.[278]

```
ggplot(tweets_sample_organic_language_month_AFINN_long,
       aes(x=year_month, y='Number of Tweets', fill=Sentiment)) +
       geom_col() +
       scale_fill_manual(values=c("Positive"="forestgreen",
                                  "Negative"="indianred",
                                  "Neutral"="darkgrey")) +
       xlab("") +
       ylab("Number of Tweets") +
       scale_x_yearmon(labels=date_format("%b %y"),
                  breaks = seq(from =
      min(tweets_sample_organic_language_month_AFINN_long$year_month),
                               to =
```

[277] See Figure 30.

[278] See Figure 31.

```
      max(tweets_sample_organic_language_month_AFINN_long$year_month),
                               by = 1/12),
                  expand = expansion(mult = c(0.015, 0.015))) +
       scale_y_continuous(labels = scales::comma,
                          expand = expansion(mult = c(0.015, 0.04))) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("AFINN Tweets by Sentiment per Month\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(fill="Sentiment") +
       guides(fill=guide_legend(title=NULL))
```

Create a 100% stacked bar chart.[279]

```
ggplot(tweets_sample_organic_language_month_AFINN_long,
       aes(x=year_month, y='Number of Tweets', fill=Sentiment)) +
       geom_col(position = position_fill(reverse = FALSE)) +
       scale_fill_manual(values=c("Positive"="forestgreen",
                                  "Negative"="indianred",
                                  "Neutral"="darkgrey")) +
       xlab("") +
       ylab("Percentage of Tweets") +
       scale_x_yearmon(labels=date_format("%b %y"),
                       breaks = seq(from =
      min(tweets_sample_organic_language_month_AFINN_long$year_month),
                                    to =
      max(tweets_sample_organic_language_month_AFINN_long$year_month),
                                    by = 1/12),
                                    expand = expansion(mult = c(0.015,
                                                               0.015))
                      ) +
       scale_y_continuous(labels = scales::percent_format(),
                          expand = expansion(mult = c(0.015, 0.015))
                         ) +
       theme(axis.text.y = element_text(size = 12)) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("AFINN Tweets by Sentiment per Month (%)\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(fill="Sentiment") +
       guides(fill=guide_legend(title=NULL))
```

[279] See Figure 32.

# Appendix Q – NRC Positivity Index Calculation

The R in this appendix was used to calculate and visualise the NRC Positivity Index by performing an inner join between the tokens (words) in each tweet and the lexicons included in NRC. This process allowed the determination of the overall positivity of the tweets based on the words they contain, and the result was plotted to provide insights into the sentiment distribution across the dataset.[280] [281]

Calculate the NRC positivity index for each tweet.

```
get_sentiments("nrc")

tweets_sentiment_nrc <- tweets_sentiment %>%
             inner_join(get_sentiments("nrc")) %>%
             group_by(line) %>%
                   summarise(positive = sum(sentiment == "positive"),
                             negative = sum(sentiment == "negative"),
                             anger = sum(sentiment == "anger"),
                             anticipation = sum(sentiment ==
                                                "anticipation"),
                             disgust = sum(sentiment == "disgust"),
                             fear = sum(sentiment == "fear"),
                             joy = sum(sentiment == "joy"),
                             sadness = sum(sentiment == "sadness"),
                             surprise = sum(sentiment == "surprise"),
                             trust = sum(sentiment == "trust")
                            )

tweets_sentiment_nrc$positivity_index_nrc <-
                                tweets_sentiment_nrc$positive -
                                tweets_sentiment_nrc$negative
```

Join the positivity index of each tweet to the original dataset to get each tweet’s date and calculate each date’s positivity index.

```
tweets_organic_en$line <- seq.int(nrow(tweets_organic_en))

tweets_organic_en_4 <- tweets_organic_en %>%
                       left_join(tweets_sentiment_nrc)

tweets_organic_en_4 <- tweets_organic_en_4[,c("date", "lang",
                                              "positivity_index_nrc")]

tweets_organic_en_4[is.na(tweets_organic_en_4)] <- 0

tweets_organic_en_4$date <- as.Date(tweets_organic_en_4$date)
```

[280] See Section 7.3 *Sentiment Analysis with NRC*.

[281] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix N – *Subset of Organic English*.

```
tweets_organic_en_4 <- tweets_organic_en_4 %>%
                       group_by(date) %>%
                       summarise(sent_nrc = sum(positivity_index_nrc),
                                 nrc_pos = sum(positivity_index_nrc >
                                               0),
                                 nrc_neg = sum(positivity_index_nrc <
                                               0),
                                 nrc_zero = sum(positivity_index_nrc
                                                == 0))
```

Add to the dataframe the number of tweets per day that are organic and in English to then calculate an average positivity index per day.

```
tweets_organic_en_4 <- tweets_date_organic_english %>%
                       left_join(tweets_organic_en_4)

tweets_organic_en_4$tweets_sentiment_nrc_avg <-
                       tweets_organic_en_4$sent_nrc /
                       tweets_organic_en_4$tweets_count
```

Plot the positivity index per day.[282]

```
ggplot(tweets_organic_en_4,
       aes(x=date, y=tweets_sentiment_nrc_avg)) +
       geom_line(color = "steelblue") + xlab("") +
       ggtitle("NRC Positivity Index per Day\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(y= "Positivity Index") +
       scale_x_date(breaks = seq(as.Date("2020-01-01"),
                                 as.Date("2023-01-01"),
                                 by = "1 month"),
                    date_labels = "%b %y",
                    expand = expansion(mult = c(0.015, 0.015))
                   ) +
       theme(axis.text.x = element_text(angle = 90,
                                        vjust = 0.5,
                                        hjust = 1))
```

Create a monthly positivity index.

```
tweets_sample_organic_language_month_nrc <- tweets_organic_en_4 %>%
  group_by( year(date),month(date)) %>%
  summarise(tweets_sentiment_nrc_avg = mean(tweets_sentiment_nrc_avg),
            nrc_pos = sum(nrc_pos),
            nrc_neg = sum(nrc_neg),
            nrc_zero = sum(nrc_zero))

tweets_sample_organic_language_month_nrc$year_month <-
as.yearmon(paste(tweets_sample_organic_language_month_nrc$'year(date)'
, tweets_sample_organic_language_month_nrc$'month(date)'),
  "%Y %m")
```

[282] See Figure 33.

Draw a line graph showing the monthly positivity index.[283]

```
ggplot(tweets_sample_organic_language_month_nrc,
       aes(x=year_month, y=tweets_sentiment_nrc_avg)) +
       geom_line(color="steelblue",size=0.8) + xlab("") +
       scale_x_yearmon(labels=date_format("%b %y"),
                       breaks = seq(from =
             min(tweets_sample_organic_language_month_nrc$year_month),
                                      to =
             max(tweets_sample_organic_language_month_nrc$year_month),
                                      by = 1/12),
                                      expand = expansion(mult =
                                                      c(0.015, 0.015))
                      ) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("NRC Positivity Index per Month\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(x="", y= "Positivity Index")
```

To draw stacked bar charts showing the monthly positivity index, first convert the dataframe to long format and reorder the levels of the sentiment variable.

Finally, create a stacked bar chart and a 100% stacked bar chart.[284]

```
tweets_sample_organic_language_month_nrc_long <-
            pivot_longer(tweets_sample_organic_language_month_nrc,
                         cols = c("nrc_pos", "nrc_neg", "nrc_zero"),
                         names_to = "Sentiment",
                         values_to = "Number of Tweets")

tweets_sample_organic_language_month_nrc_long <-
                    tweets_sample_organic_language_month_nrc_long %>%
                    mutate(Sentiment = recode(Sentiment,
                           "nrc_pos" = "Positive",
                           "nrc_neg" = "Negative",
                           "nrc_zero" = "Neutral"))

tweets_sample_organic_language_month_nrc_long$Sentiment <-
  factor(tweets_sample_organic_language_month_nrc_long$Sentiment,
         levels = c("Negative", "Neutral", "Positive"))
```

Create a stacked bar chart.[285]

```
ggplot(tweets_sample_organic_language_month_nrc_long,
       aes(x=year_month, y=`Number of Tweets`, fill=Sentiment)) +
       geom_col() +
       scale_fill_manual(values=c("Positive"="forestgreen",
                                  "Negative"="indianred",
                                  "Neutral"="darkgrey")) +
       xlab("") +
```

[283] See Figure 34.

[284] See Figure 35.

[285] See Figure 36.

```
       ylab("Number of Tweets") +
       scale_x_yearmon(labels=date_format("%b %y"),
                       breaks = seq(from =
        min(tweets_sample_organic_language_month_nrc_long$year_month),
                                    to =
        max(tweets_sample_organic_language_month_nrc_long$year_month),
                                    by = 1/12),
                                    expand = expansion(mult = c(0.015,
                                                                0.015))
                      ) +
       scale_y_continuous(labels = scales::comma,
                          expand = expansion(mult = c(0.015, 0.04))) +

       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("NRC Tweets by Sentiment per Month\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(fill="Sentiment") +
       guides(fill=guide_legend(title=NULL))
```

Create a 100% stacked bar chart.

```
ggplot(tweets_sample_organic_language_month_nrc_long,
       aes(x=year_month, y=`Number of Tweets`, fill=Sentiment)) +
       geom_col(position = position_fill(reverse = FALSE)) +
       scale_fill_manual(values=c("Positive"="forestgreen",
                                  "Negative"="indianred",
                                  "Neutral"="darkgrey")) +
       xlab("") +
       ylab("Percentage of Tweets") +
       scale_x_yearmon(labels=date_format("%b %y"),
                       breaks = seq(from =
        min(tweets_sample_organic_language_month_nrc_long$year_month),
                                    to =
        max(tweets_sample_organic_language_month_nrc_long$year_month),
                                    by = 1/12),
                       expand = expansion(mult = c(0.015, 0.015))
                      ) +
       scale_y_continuous(labels = scales::percent_format(),
                          expand = expansion(mult = c(0.015, 0.015))
                         ) +
       theme(axis.text.y = element_text(size = 12)) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("NRC Tweets by Sentiment per Month (%)\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(fill="Sentiment") +
       guides(fill=guide_legend(title=NULL))
```

## Appendix R – SentimentR Positivity Index Calculation

The given R code was used to calculate the SentimentR Positivity Index and create a visualisation of it, allowing for a clear understanding of the positivity trends within the analysed tweets.[286] [287]

Calculate the SentimentR positivity index for each sentence and for each tweet.

```
sentimentr_sentiment <- get_sentences(tweets_organic_en$content)

sentimentr_sentiment <- sentiment(sentimentr_sentiment)

sentimentr_grouped <- sentimentr_sentiment %>%
                      group_by(element_id) %>%
                      summarise(sentiment = mean(sentiment))

colnames(sentimentr_grouped)[1] <- "line"
```

Join the positivity index of each tweet to the original dataset to get each tweet's date and calculate each date's positivity index.

```
tweets_organic_en_5 <- tweets_organic_en %>%
                       left_join(sentimentr_grouped)

tweets_organic_en_5 <- tweets_organic_en_5[,c("date","sentiment")]

tweets_organic_en_5[is.na(tweets_organic_en_5)] <- 0

tweets_organic_en_5$date <- as.Date(tweets_organic_en_5$date)

tweets_organic_en_5 <- tweets_organic_en_5 %>%
                       group_by(date) %>%
                       summarise(sentiment_sentimentr =
                                 sum(sentiment),
                       sentimentr_pos = sum(sentiment > 0),
                       sentimentr_neg = sum(sentiment < 0),
                       sentimentr_zero = sum(sentiment == 0))
```

[286] See Section 7.4 *Sentiment Analysis with SentimentR*.

[287] Before executing this code, run the ones in Appendix B – *R Setup* and Appendix N – *Subset of Organic English*.

Add to the dataframe the number of tweets per day that are organic and in English to then calculate an average positivity index per day.

```
tweets_organic_en_5 <- tweets_date_organic_english %>%
                       left_join(tweets_organic_en_5)

tweets_organic_en_5$sentiment_sentimentr_avg <-
                           tweets_organic_en_5$sentiment_sentimentr /
                           tweets_organic_en_5$tweets_count
```

Plot the positivity index per day.[288]

```
ggplot(tweets_organic_en_5, aes(x=date, y=sentiment_sentimentr_avg)) +
       geom_line(color = "steelblue") +
       xlab("") +
       ggtitle("SentimentR Positivity Index per Day\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(y= "Positivity Index") +
       scale_x_date(breaks = seq(as.Date("2020-01-01"),
                                 as.Date("2023-01-01"),
                                 by = "1 month"),
                                 date_labels = "%b %y",
                                 expand = expansion(mult = c(0.015,
                                                             0.015))
                   ) +
       theme(axis.text.x = element_text(angle = 90,
                                        vjust = 0.5,
                                        hjust = 1))
```

Create a monthly positivity index.

```
tweets_sample_organic_language_month_sentimentr <-
  tweets_organic_en_5 %>%
  group_by( year(date),month(date)) %>%
  summarise(sentiment_sentimentr_avg = mean(sentiment_sentimentr_avg),
  sentimentr_pos = sum(sentimentr_pos),
  sentimentr_neg = sum(sentimentr_neg),
  sentimentr_zero = sum(sentimentr_zero))

tweets_sample_organic_language_month_sentimentr$year_month <-
      as.yearmon(paste(tweets_sample_organic_language_month_sentimentr
                       $'year(date)',
                       tweets_sample_organic_language_month_sentimentr
                       $'month(date)'),
                 "%Y %m")
```

[288] See Figure 37.

Draw a line graph showing the monthly positivity index.[289]

```
ggplot(tweets_sample_organic_language_month_sentimentr,
       aes(x=year_month, y=sentiment_sentimentr_avg)) +
       geom_line(color="steelblue",size=0.8) +
       xlab("") +
       scale_x_yearmon(labels=date_format("%b %y"),
                       breaks = seq(from =
      min(tweets_sample_organic_language_month_sentimentr$year_month),
                                    to =
      max(tweets_sample_organic_language_month_sentimentr$year_month),
                                    by = 1/12),
                       expand = expansion(mult = c(0.015, 0.015))) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("SentimentR Positivity Index per Month\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(x="", y= "Positivity Index")
```

To draw stacked bar charts showing the monthly positivity index, first convert the dataframe to long format and reorder the levels of the sentiment variable.

```
tweets_sample_organic_language_month_sentimentr_long <-
         pivot_longer(tweets_sample_organic_language_month_sentimentr,
                      cols = c("sentimentr_pos",
                               "sentimentr_neg",
                               "sentimentr_zero"),
                      names_to = "Sentiment",
                      values_to = "Number of Tweets")

tweets_sample_organic_language_month_sentimentr_long <-
              tweets_sample_organic_language_month_sentimentr_long %>%
              mutate(Sentiment = recode(Sentiment,
                                        "sentimentr_pos" = "Positive",
                                        "sentimentr_neg" = "Negative",
                                        "sentimentr_zero" = "Neutral")
                    )

tweets_sample_organic_language_month_sentimentr_long$Sentiment <-
factor(tweets_sample_organic_language_month_sentimentr_long$Sentiment,
       levels = c("Negative", "Neutral", "Positive"))
```

Create a stacked bar chart.[290]

```
ggplot(tweets_sample_organic_language_month_sentimentr_long,
       aes(x=year_month, y=`Number of Tweets`, fill=Sentiment)) +
       geom_col() +
       scale_fill_manual(values=c("Positive"="forestgreen", 3
                                  "Negative"="indianred",
                                  "Neutral"="darkgrey")) +
       xlab("") +
       ylab("Number of Tweets") +
       scale_x_yearmon(labels=date_format("%b %y"),
```

[289] See Figure 38.

[290] See Figure 39.

```
                        breaks = seq(from =
 min(tweets_sample_organic_language_month_sentimentr_long$year_month),
                                      to =
 max(tweets_sample_organic_language_month_sentimentr_long$year_month),
                                      by = 1/12),
                        expand = expansion(mult = c(0.015, 0.015))) +
       scale_y_continuous(labels = scales::comma,
                           expand = expansion(mult = c(0.015, 0.04))) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("SentimentR Tweets by Sentiment per Month\n2020-2022")+
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(fill="Sentiment") +
       guides(fill=guide_legend(title=NULL))
```

Create a 100% stacked bar chart.[291]

```
ggplot(tweets_sample_organic_language_month_sentimentr_long,
       aes(x=year_month, y=`Number of Tweets`, fill=Sentiment)) +
       geom_col(position = position_fill(reverse = FALSE)) +
       scale_fill_manual(values=c("Positive"="forestgreen",
                                  "Negative"="indianred",
                                  "Neutral"="darkgrey")) +
       xlab("") +
       ylab("Percentage of Tweets") +
       scale_x_yearmon(labels=date_format("%b %y"),
                        breaks = seq(from =
 min(tweets_sample_organic_language_month_sentimentr_long$year_month),
                                      to =
 max(tweets_sample_organic_language_month_sentimentr_long$year_month),
                                      by = 1/12),
                        expand = expansion(mult = c(0.015, 0.015))) +
       scale_y_continuous(labels = scales::percent_format(),
                           expand = expansion(mult = c(0.015, 0.015))
                          ) +
       theme(axis.text.y = element_text(size = 12)) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("SentimentR Tweets by Sentiment per Month (%)\n2020-
               2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(fill="Sentiment") +
       guides(fill=guide_legend(title=NULL))
```

[291] See Figure 40.

## Appendix S – Comparison of Sentiment Indexes

The R code in this appendix was utilised to compare and analyse the sentiment indexes derived from four methods: Bing, AFINN, NRC, and SentimentR. The code was also used to create visualisations to present a clear comparison of the sentiment scores obtained from each method, allowing for a straightforward interpretation of the variations in sentiment across the tweets. Additionally, correlation analysis was performed to understand the degree of agreement or divergence between the different sentiment indexes, providing insights into the level of consistency among the methods in assessing sentiment in the dataset.[292] [293]

First create a dataset that combines the four previously generated positivity indexes (Bing, AFINN, NRC, and SentimentR) for further analysis and comparison. One dataset per day and another dataset per month.

```
positivity_index_combined <- left_join(tweets_organic_en_2,
                                       tweets_organic_en_3)

positivity_index_combined <- left_join(positivity_index_combined,
                                       tweets_organic_en_4)

positivity_index_combined <- left_join(positivity_index_combined,
                                       tweets_organic_en_5)

positivity_index_combined_month <- left_join(
                      tweets_sample_organic_language_month,
                      tweets_sample_organic_language_month_AFINN)

positivity_index_combined_month <- left_join(
                      positivity_index_combined_month,
                      tweets_sample_organic_language_month_nrc)

positivity_index_combined_month <- left_join(
                      positivity_index_combined_month,
                      tweets_sample_organic_language_month_sentimentr)

positivity_index_combined_month <-
                      positivity_index_combined_month[c(3,7,8,12,16)]
```

[292] See Section 7.5 *Evaluation and Comparison of Sentiment Analysis Methods*.

[293] Before executing this code, run the ones in Appendix B – *R Setup*, Appendix O – *Bing Positivity Index Calculation*, Appendix P – *AFINN Positivity Index Calculation*, Appendix Q – *NRC Positivity Index Calculation* and Appendix R – *SentimentR Positivity Index Calculation*.

Create a monthly dataframe and plot the four monthly positivity indexes together.[294]

```
positivity_index_combined_month_melt <-
                      melt(positivity_index_combined_month,
                      id = "year_month")

positivity_index_combined_month_melt$variable <-
                  ifelse(positivity_index_combined_month_melt$variable
                         == "positivity_index_avg", "Bing",
                  ifelse(positivity_index_combined_month_melt$variable
                         == "tweets_sentiment_nrc_avg", "NRC",
                  ifelse(positivity_index_combined_month_melt$variable
                         == "sentiment_afinn_avg", "AFINN",
                         "SentimentR")))

colnames(positivity_index_combined_month_melt)[2] <- "Index"

ggplot(positivity_index_combined_month_melt,
       aes(x=year_month, y=value, color = Index)) +
       geom_line(size=0.75) +
       scale_x_yearmon(labels=date_format("%b %y"),
                       breaks = seq(from =
                 min(positivity_index_combined_month_melt$year_month),
                                    to =
                 max(positivity_index_combined_month_melt$year_month),
                                    by = 1/12),
                       expand = expansion(mult = c(0.015, 0.015))
                      ) +
       theme(axis.text.x = element_text(angle = 90)) +
       ggtitle("Positivity Index per Month AFINN, Bing, NRC &
               SentimentR\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       labs(x="", y= "Positivity Index")
```

Create a daily dataframe and plot the four daily positivity indexes together.[295]

```
positivity_index_combined_melt <- melt(positivity_index_combined,
                                       id = "date")

positivity_index_combined_melt$variable <-
                   ifelse(positivity_index_combined_melt$variable
                          == "positivity_index_avg", "Bing",
                   ifelse(positivity_index_combined_melt$variable
                          == "tweets_sentiment_nrc_avg", "NRC",
                   ifelse(positivity_index_combined_melt$variable
                          == "sentiment_afinn_avg", "AFINN",
                   ifelse(positivity_index_combined_melt$variable
                          == "sentiment_sentimentr_avg", "SentimentR",
                          positivity_index_combined_melt))))

positivity_index_combined_melt <-
                   filter(positivity_index_combined_melt,
                          positivity_index_combined_melt$variable %in%
```

[294] See Figure 42.

[295] See Figure 41.

```
                          c("Bing","NRC","AFINN","SentimentR")
                         )

colnames(positivity_index_combined_melt)[2] <- "Index"

positivity_index_combined_melt$Index <-
                   as.character(positivity_index_combined_melt$Index)

ggplot(positivity_index_combined_melt,
       aes(x=date, y=value, color = Index)) +
       geom_line() +
       xlab("") +
       ggtitle("Positivity Index per Day AFINN, Bing, NRC &
                SentimentR\n2020-2022") +
       theme(plot.title = element_text(hjust = 0.5)) +
       scale_x_date(labels=date_format("%b %y"),
                    breaks = seq(from = as.Date('2020-01-02'),
                                 to = as.Date('2022-12-31'),
                                 by = '1 month'),
                    expand = expansion(mult = c(0.015, 0.015))
                   ) +
       theme(axis.text.x = element_text(angle = 90,
                                        vjust = 0.5,
                                        hjust = 1)) +
       labs(x="", y= "Positivity Index")
```

Adjust column names.

```
colnames(positivity_index_combined)[4] <- "Bing"

colnames(positivity_index_combined)[6] <- "AFINN"

colnames(positivity_index_combined)[8] <- "NRC"

colnames(positivity_index_combined)[10] <- "SentimentR"
```

Calculate Pearson’s correlation between the four indexes.[296]

```
cor(positivity_index_combined[,
    c('Bing', 'AFINN', 'NRC', 'SentimentR')],
    use = "pairwise.complete.obs")

pairs.panels(positivity_index_combined[,
             c('Bing', 'AFINN', 'NRC', 'SentimentR')],
             digits = 3,
             hist.col = 'lightblue')
```

[296] See Table 16 and Figure 43.

Calculate Spearman’s Rank correlation between the 4 indexes.[297]

```
cor(positivity_index_combined[,
    c('Bing', 'AFINN', 'NRC', 'SentimentR')],
    method = "spearman")

pairs.panels(positivity_index_combined[,
             c('Bing', 'AFINN', 'NRC', 'SentimentR')],
             digits = 3,
             hist.col = 'lightblue',
             method="spearman")
```

[297] See Table 17 and Figure 44.

## Appendix T – Correlation Analysis: Sentiment Indexes and PFE Traded Volume

The given R code was utilised to perform a correlation analysis between the sentiment indexes (Bing, AFINN, NRC, and SentimentR) and the traded volume of the PFE stock, aiming to investigate potential relationships between social media sentiment and stock trading activity.[298] [299]

Prepare the dataframe for analysis by assuring the correct date formats, combining datasets, creating a weekday column (Sunday becomes day 1), and replacing NAs in the Volume and Positivity Index columns with 0s.

```r
tweets_date$date <- as.Date(tweets_date$date2)

PFE_df$date <- as.Date(PFE_df$date)

lm_1_df <- left_join(tweets_date, PFE_df)

lm_1_df$week_day <- weekdays(lm_1_df$date)

lm_1_df["PFE.Volume"][is.na(lm_1_df["PFE.Volume"])] <- 0

lm_1_df <- left_join(lm_1_df, positivity_index_combined, by= "date")

lm_1_df["positivity_index"][is.na(lm_1_df["positivity_index"])] <- 0
```

Calculate correlation coefficients between the four positivity indexes and PFE's traded volume. First Pearson's correlation is calculated, and then Spearman's Rank.[300]

```r
cor(lm_1_df[,c('Bing','AFINN','nrc','sentimentr','PFE.Volume')])

cor(lm_1_df[,c('Bing','AFINN','nrc','sentimentr','PFE.Volume')],
    method = "spearman")
```

[298] See Chapter 8 *Model*.

[299] Before executing this code, run the ones in Appendix B – *R Setup*, Appendix C – *Pfizer Stock Dataset Download*, Appendix F – *Tweets Dataset: General Numbers Analysis*, and Appendix S – *Comparison of Sentiment Indexes*.

[300] See Table 18.

Create absolute indexes from the positivity indexes.[301]

```
lm_1_df <- lm_1_df %>% mutate(Bing_abs = abs(Bing))

lm_1_df <- lm_1_df %>% mutate(AFINN_abs = abs(AFINN))

lm_1_df <- lm_1_df %>% mutate(nrc_abs = abs(nrc))

lm_1_df <- lm_1_df %>% mutate(sentimentr_abs = abs(sentimentr))
```

Calculate correlation coefficients between the four absolute positivity indexes and PFE's traded volume. First Pearson's correlation is calculated, and then Spearman's Rank.[302]

```
cor(lm_1_df[,c('Bing_abs','AFINN_abs','nrc_abs','sentimentr_abs',
               'PFE.Volume')])

cor(lm_1_df[,c('Bing_abs','AFINN_abs','nrc_abs','sentimentr_abs',
               'PFE.Volume')], method = "spearman")
```

Remove weekends and banking holidays.

```
lm_1_df_only_weekdays <- subset(lm_1_df, !is.na(lm_1_df[,'PFE.Open']))
```

Calculate correlation coefficients between the four positivity indexes (weekends and banking holidays excluded) and PFE's traded volume. First Pearson's correlation is calculated and then Spearman's Rank.[303]

```
cor(lm_1_df_only_weekdays[,c('Bing','AFINN','nrc','sentimentr',
                             'PFE.Volume')])

cor(lm_1_df_only_weekdays[,c('Bing','AFINN','nrc','sentimentr',
                             'PFE.Volume')], method = "spearman")
```

Calculate correlation coefficients between the number of tweets published and PFE's traded volume, both including and excluding weekends and banking holidays. First Pearson's correlation is calculated and then Spearman's Rank.[304]

```
cor(lm_1_df_only_weekdays[,c('tweets_count','PFE.Volume')])

cor(lm_1_df[,c('tweets_count','PFE.Volume')])

cor(lm_1_df_only_weekdays[,c('tweets_count','PFE.Volume')],
    method = "spearman")

cor(lm_1_df[,c('tweets_count','PFE.Volume')],method = "spearman")
```

[301] See Table 19.

[302] See Table 20.

[303] See Table 21.

[304] See Table 22.

# Appendix U – Correlation Analysis: Sentiment Indexes and PFE Daily Returns

The provided R code was used to perform a correlation analysis between the sentiment indexes (Bing, AFINN, NRC, and SentimentR) and the daily returns of the PFE stock, aiming to investigate potential relationships between social media sentiment and stock trading activity.[305] [306]

Prepare the dataframe for analysis by calculating PFE daily returns, assuring the correct date formats, and joining the positivity indexes dataframe.

```
daily_returns <- dailyReturn(PFE)

daily_returns <- data.frame(date2 = index(daily_returns),
                            daily_return = coredata(daily_returns))

lm_1_df_only_weekdays$date2 <- as.Date(lm_1_df_only_weekdays$date2)

lm_1_df_only_weekdays <- left_join(lm_1_df_only_weekdays,
                                   daily_returns)
```

Calculate correlation coefficients between the four positivity indexes and PFE's daily returns. First Pearson's correlation is calculated and then Spearman's Rank.[307]

```
cor(lm_1_df_only_weekdays[,c('Bing','AFINN','nrc','sentimentr',
                             'daily.returns')])

cor(lm_1_df_only_weekdays[,c('Bing','AFINN','nrc','sentimentr',
                             'daily.returns')], method = "spearman")
```

Calculate correlation coefficients between the four absolute positivity indexes and PFE's daily returns. First Pearson's correlation is calculated and then Spearman's Rank.[308]

```
cor(lm_1_df_only_weekdays[,c('Bing_abs','AFINN_abs','nrc_abs',
                             'sentimentr_abs','daily.returns')])

cor(lm_1_df_only_weekdays[,c('Bing_abs','AFINN_abs','nrc_abs',
    'sentimentr_abs','daily.returns')], method="spearman")
```

[305] See Chapter 8 *Model*.

[306] Before executing this code, run the ones in Appendix B – *R Setup*, Appendix C – *Pfizer Stock Dataset Download*, Appendix S – *Comparison of Sentiment Indexes* and Appendix T – *Correlation Analysis: Sentiment Indexes and PFE Traded Volume*.

[307] See Table 23.

[308] See Table 24.

Calculate correlation coefficients between the number of tweets published and PFE's daily returns, excluding weekends and banking holidays. First Pearson's correlation is calculated and then Spearman's Rank.[309]

```
cor(lm_1_df_only_weekdays[,c('tweets_count','daily.returns')])

cor(lm_1_df_only_weekdays[,c('tweets_count','daily.returns')],
    method = "spearman")
```

[309] See Table 25.