Stat Madness

STAT 385 FA2019 - Team 18

Jovan Krcadinac - krcadin2
Ameya Shahane - ameyaps2
Danny Annese - annese2
Do Yeop Kwon - dykwon2

March 15, 2019

Abstract

In this project, we will be analyzing live time tweets for the upcoming March Madness college basketball tournament. From this analysis, we hope to understand the series of associations between variables conflicting with the game and user tweet responses.

Contents

1 Introduction

Founded in 2006, Twitter is a popular social networking service that allows individuals to share information within and beyond one's network by composing tweets with 280 characters or fewer. It currently has more than 321 million monthly active users (Twitter, 2018). There is widespread recognition by both sports teams and sports media that Twitter is a powerful and revolutionary tool for publishing, promotion, and relationship management (Anderson, 2018). As a result, more and more sports organizations have adopted Twitter accounts to enhance their levels of interaction with fans worldwide. Our idea for this project was to analyze live-time tweets revolving around the upcoming March Madness college basketball tournament. We made several conclusions about this analysis including: does team location influence the number of tweets coming from a particular area, is there any point in the games that there is a spike in the twitter usage, monitoring followers for both winning and losing teams, and other connections we found throughout the data.

Our purpose for this project was to analyze a series of associations between variables conflicting with the game and user tweet responses. A few examples of these associations were an analysis of spikes of tweets during the game, as well as a monitoring of most frequently tweeted words during the game. We sought an increase in social media activity during major sports events. In this case, we focused on Twitter for social media and March Madness 2019 for the sports events. In our data collection, we had at least 10000 observations in the form of at least 10000 tweets and at least 80 variable names such as username, the date the account was created, time of the tweet, geodata, content, device used, link (URL), hashtags, followers, retweet counts, and favorite counts, etc. By employing a wide variety of packages and through the use of close evaluation techniques, we brought our course's focus on statistical programming into a close perspective.

2 Related Work

One idea that had been attempted while connecting sports and Twitter was the effect that Twitter had on ads and fan engagement during sporting events. Our idea differed from this study because we focused on the effect the sport itself has on Twitter and how events during the games impacted Twitter activity. Moreover, we were more interested in fan reactions to in-game events rather than the number of fans that were tuning in.

Another idea that had been attempted connecting sports and Twitter was the frequency of NBA teams tweeting during the games; this showed the difference in the number of conversations during the games between the NBA teams on Twitter. Our idea differed from this study as we were mostly focusing on the tweets by fans, not specific to NBA teams or in this case, the college basketball teams.

We decided to use the following links: https://cran.r-project.org/web/packages/rtweet/rtweet.pdf ,https://cran.r-project.org/web/packages/rtweet/index.html, and https://developer.twitter.com/en/docs/basics/authentication/overview/oauth.

We used these links in order to analyze scholarly articles that referenced data analysis using rtweet and to refer to prior attempts at this process.

3 Methods

We have used the Shiny interface in Rstudio to build an interactive visualization display. Some of the required packages we used are Rtweet (Kearney 2018), which allows us to connect a Twitter API to R, ggplot2 (Hadley Wickham 2018) to help us easily visualize our results, stats package (R Core Team 2018b) to help us analyze the data, as well as some other basic packages to make our app and report smoother. These include tidyr (Wickham and Henry 2019), dplyr(Wickham et al. 2018), Matrix(Douglas Bates 2019), plyr(Wickham 2011), MatrixModels (Bates and Maechler 2015), stats (R Core Team 2018b), leaps (Fortran code by Alan Miller

2017), stringR(Wickham 2018), tidytext (Silge and Robinson 2016), httpuv (Cheng et al. 2019), tm(Ingo Feinerer and Meyer 2018), wordcloud2(Lang and Chien 2018), and ggthemes(Arnold 2019).

We first made our developer accounts that gave us access to Twitter's API and have dabbled around in the Shiny interface. The process of obtaining a developer's account took approximately a week maximum. Also, the Twitter API had events shortlisted that are based on current affairs. This is the reason why we decided to focus on March Madness which is a recent event from the time of our writing.

In terms of the statistical method, we first used the Twitter API to get a downloaded database of tweets respective to March Madness. Then, in order to understand the points in the game where there is a spike in Twitter usage, we employed the gtrends API to parse through buzzwords and rank the severity of fan engagement based on key factors such as the number of game-related words. This is because pivotal parts of a game day such as the final score had a significant effect on the spike of number of tweets and produced key data to study. In the process of using the gtrends API, we ensured that our other variables such as username, age, gender, and the device used would be filtered. The limitation to keep note of is the fact that our Twitter API would only be active up to 3 days from the sports event. This entitled us to act fast in terms of the data retrieval process. However, there is also a live implementation of the Twitter API which would require at least one group member to be active at the scene (watching a live March Madness Game) and take note of live user data tweet trends.

To elaborate, our packages, such as rtweet package (Kearney 2018) allowed us to complement the API key with our live search of the twitter data based on specified hash tags to filter certain data pertaining to the March Madness games. This is because our rtweet package (Kearney 2018) contains the stream_tweets and parse_tweets functions that allowed us to to first specify a filename to parse our data and then store our parsed json data into a local data frame in our environment. In addition, our stream_tweets method of the rtweet package (Kearney 2018) allowed us to specify hashtags to filter the data with while our search_tweets located data with these specific hashtags. Our write_as_csv method from rtweet package (Kearney 2018) allowed us to convert the local json data frame parsed into a csv file to share among our team members for further extraction and analysis of data using the ggplot package (Hadley Wickham 2018) to plot it. Our httpuv package (Cheng et al. 2019) allowed us to extract twitter data via a web server in a live setting and complement our API structure. Namely, this helped us parse our HTTP requests from twitter for our college basketball games. In addition, our tidytext package (Silge and Robinson 2016) used the dplyr package to tidy the data and help us parse it into a json via our live setting use of the API. The tidytext package (Silge and Robinson 2016) in itself is a text mining tool for loading the twitter data into a json format as per R's JSON format. Moreover, we used plyr package (Wickham 2011) to cut the time intervals in duration of five miniutes. Not to forget, our baseR package (R Core Team 2018a) allowed us to add columns and tidy the data frame in a way that allowed us to identify which game the data frame belonged to via an additional "GameName" column.

We then decided to make a shiny application for our basketball games. This shiny app includes the following for each elite basketball game: a graph of the number of tweets over time, a summary tab using a multi-regression linear model calculating the number of twitter followers, and a wordcloud. For this application's graph, we decided to retrieve the number of followers and time using our tidied data frames through the methodology afforementioned. The summary tab in the app was made using a multi-regression linear model tryig to estimate the number of twitter followers for a user with their statuses count, friends count, and favourties count. Perhaps the most strenous part of the shiny application was crafting the wordcloud. Reason being is that we had to use a new text mining package called "tm" (Ingo Feinerer and Meyer 2018) which allowed our parsed vectorized game inputs with twitter information outputted as a text. We also had to use the wordcloud2 package (Lang and Chien 2018) to superimpose a template in the form of a circle for our outputted text values. In order to make sure our text output was valid, we also decided to use the hunspell package (Ooms 2018). We used this hunspell package (Ooms 2018) to change each word into its root word for finalizing the output for the wordcloud.

In order to preserve our data, we ensured that each JSON reading of the game was stored locally as a parsed data frame. This was done by using rtweet's parse_stream method. The predefined hashtags we used were the following comma separated values: #marchmadness, #NCAA, #NCAA2019, #collegebasketball,

#collegebball, #hoops, #upset, #upsetalert, #ncaatournament, #ncaachampionship, #ncaahoops, #ncaabball, #ncaabasketball, #gameon, #bracketology, #bracket, #bracketbusted, #bracketbuster, #cinderella, #underdog, march madness, ncaa, basketball, college basketball, bball, hoops, upset alert, bracket buster, bracket busted. We also had round specific values such as the following: #RoundOf64, #RoundOf32, #Sweet16, #Elite8, #FinalFour, and #Final4. In addition, we will be using game specific hashtags and wordings such as "-#RedRaiders, #TexasTech, #TTech, redraiders, texastech, ttech" for Texas Tech.

#Results

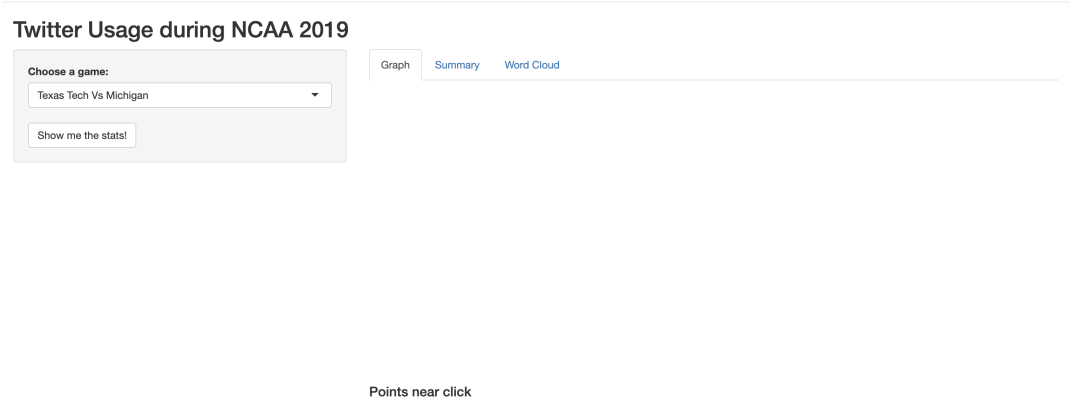


Figure 1: Start-up page of the Shiny-app

This is how our app looks upon launching it. As you can see, we have a side panel with a drop-down menu for selecting the games and a button that will show the results for the selected game upon clicking it. In the main panel, we have 3 tabs for graph, summary statistics and wordcloud.

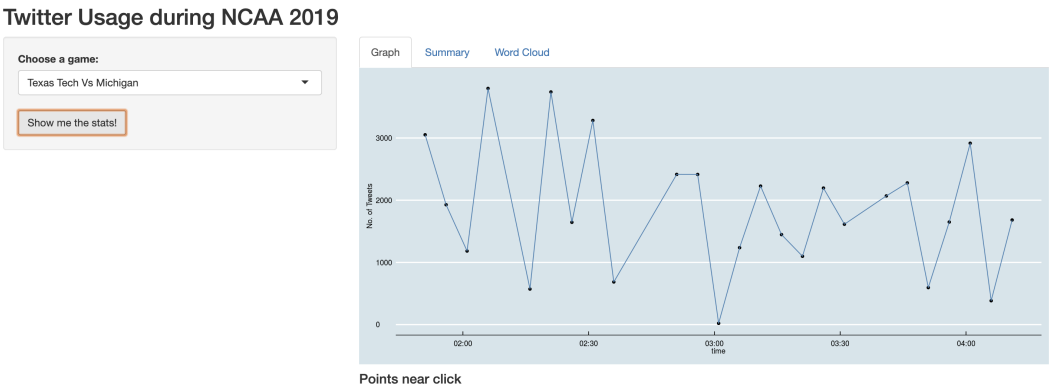


Figure 2: After clicking Show me the stats button

After clicking the "Show me the stats!" button, the statistics for the game selected (eg. Texas Tech Vs Michigan) should show up in the main panel.

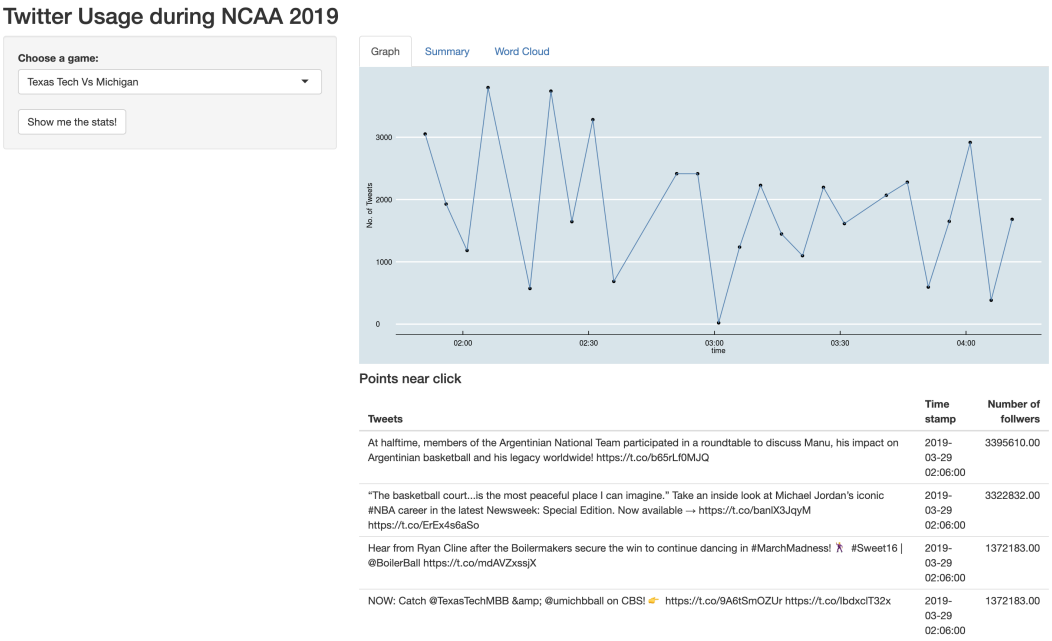


Figure 3: Graph tab of the Shiny-app

This is the graph tab of our Shiny-app. It shows the number of tweets against the time interval of the game. When the user clicks on one of the existing points on the graph, five tweets that were tweeted during that time interval will be shown at the bottom of the graph.

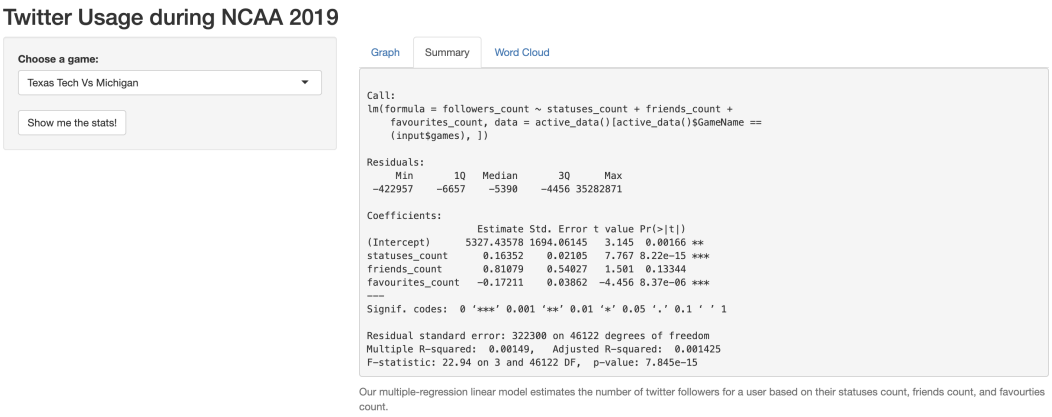


Figure 4: Summary tab of the Shiny-app

This is the summary tab of our Shiny-app. We tried to estimate the number of "followers_count" based on "statuses_count", "friends_count", and "favourites_count" using linear regression model.

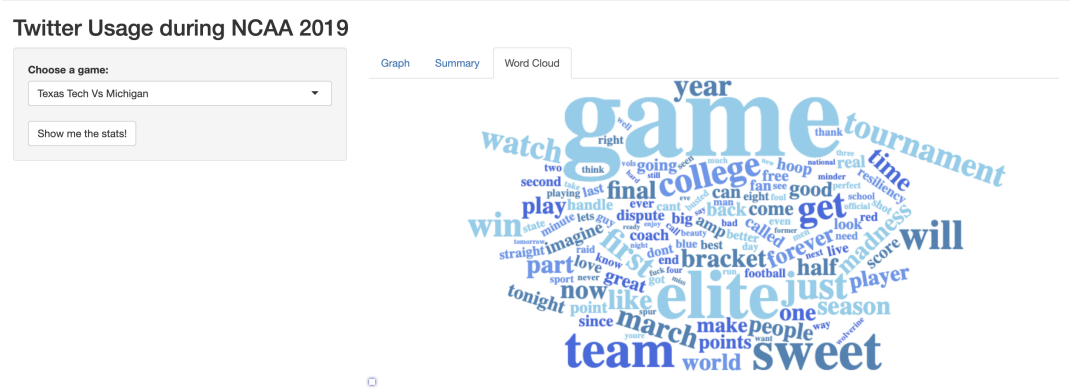
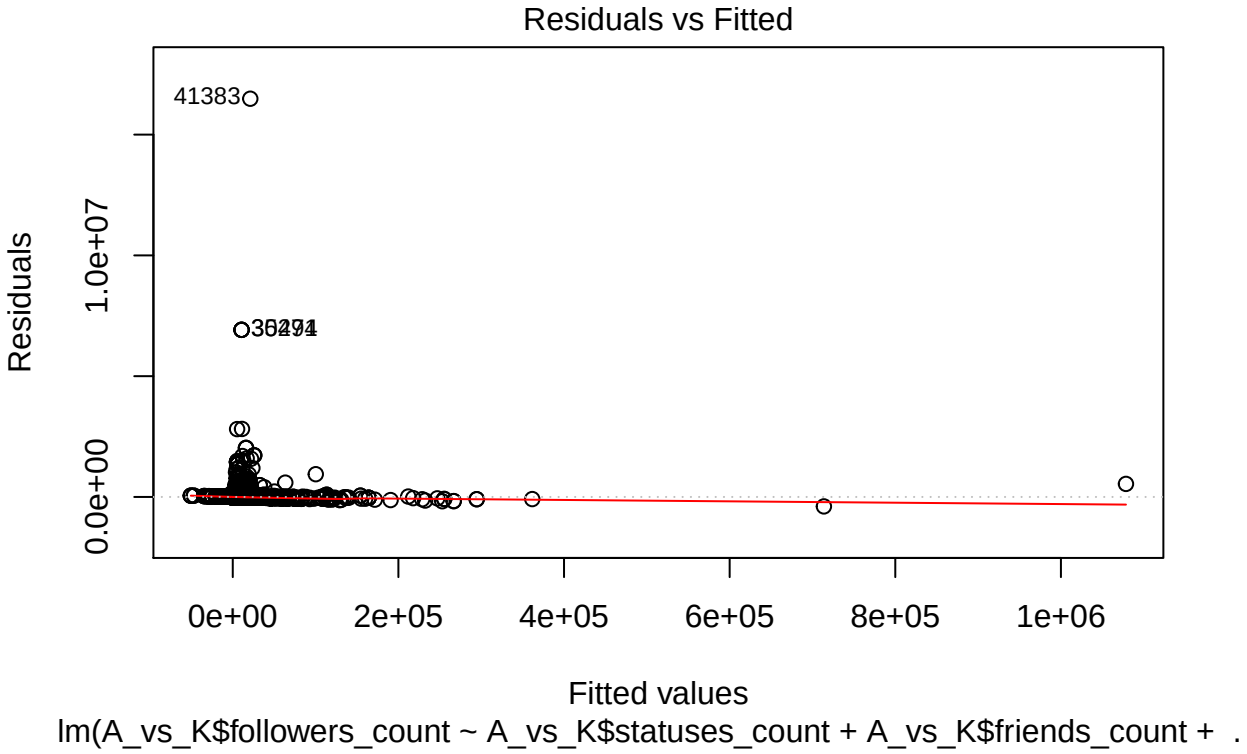


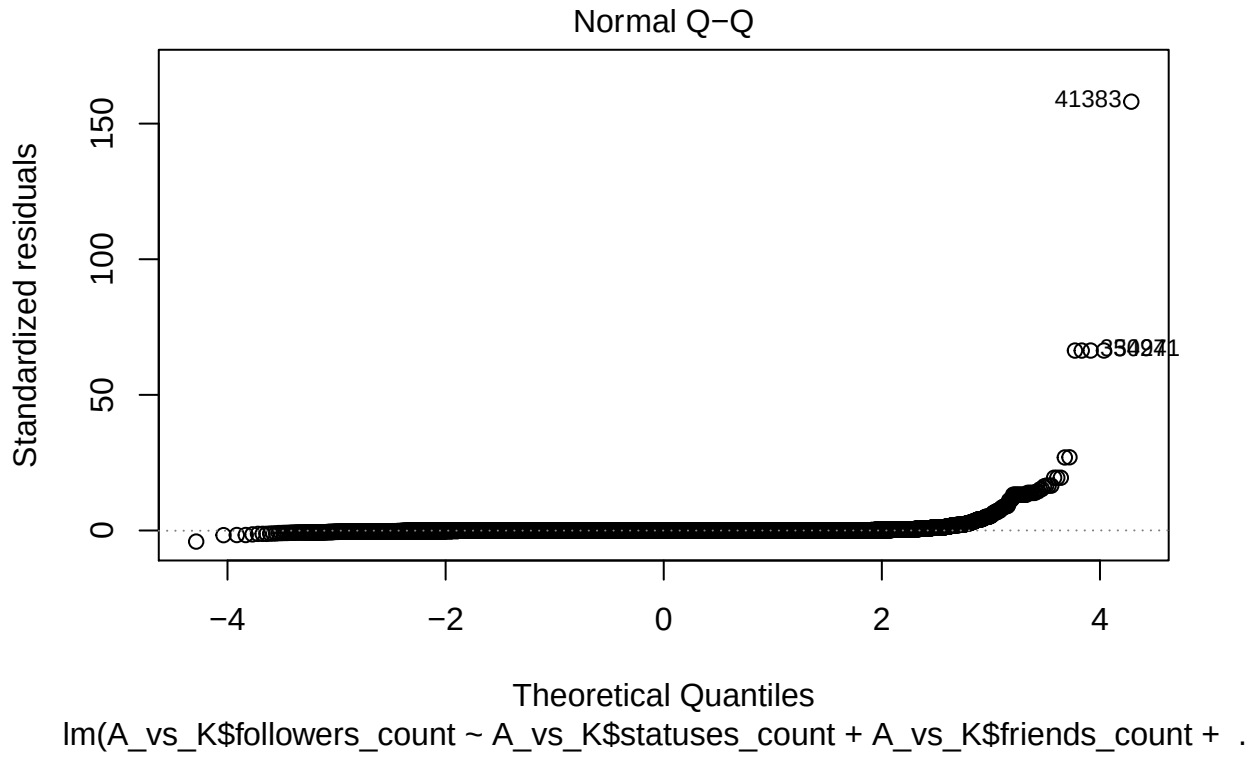
Figure 5: Wordcloud tab of the Shiny-app

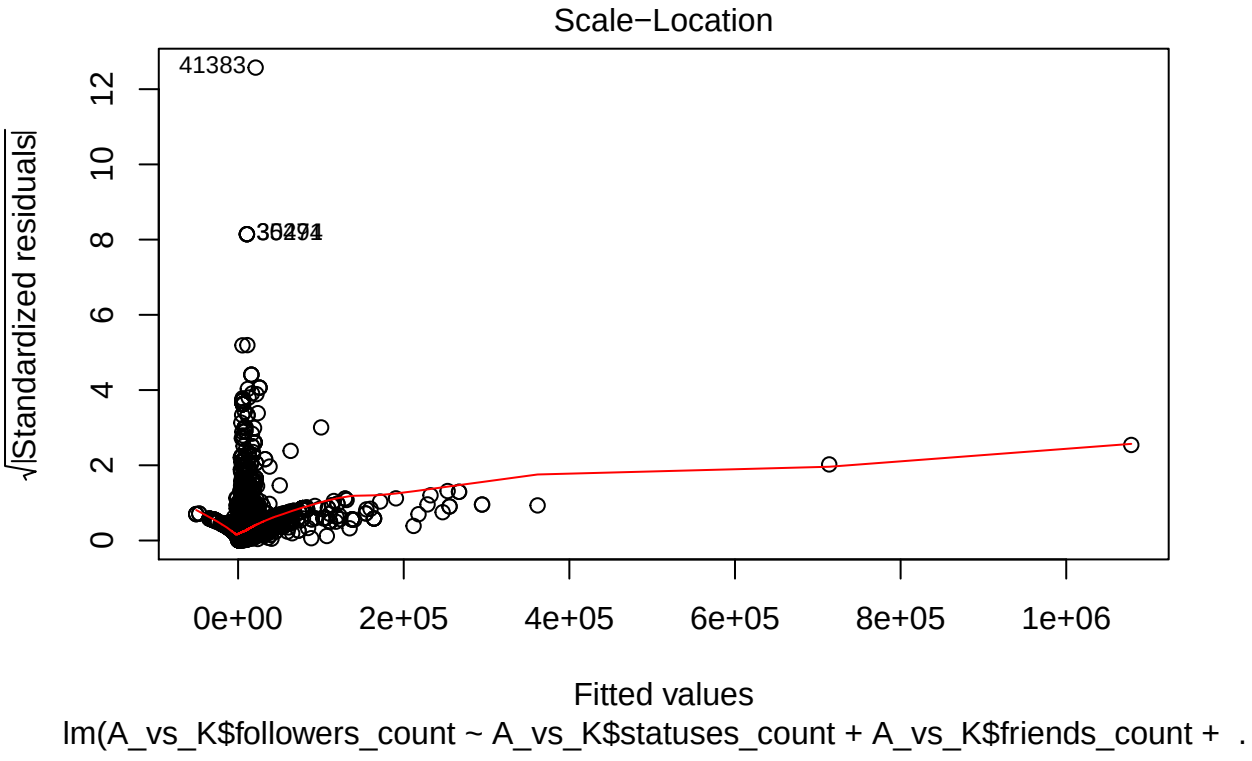
This is the wordcloud tab of our Shiny-app. It shows the words in tweets that were tweeted during the selected game. Moreover, if the users hover their cursor above the word, they are able to see the frequency count of the word.

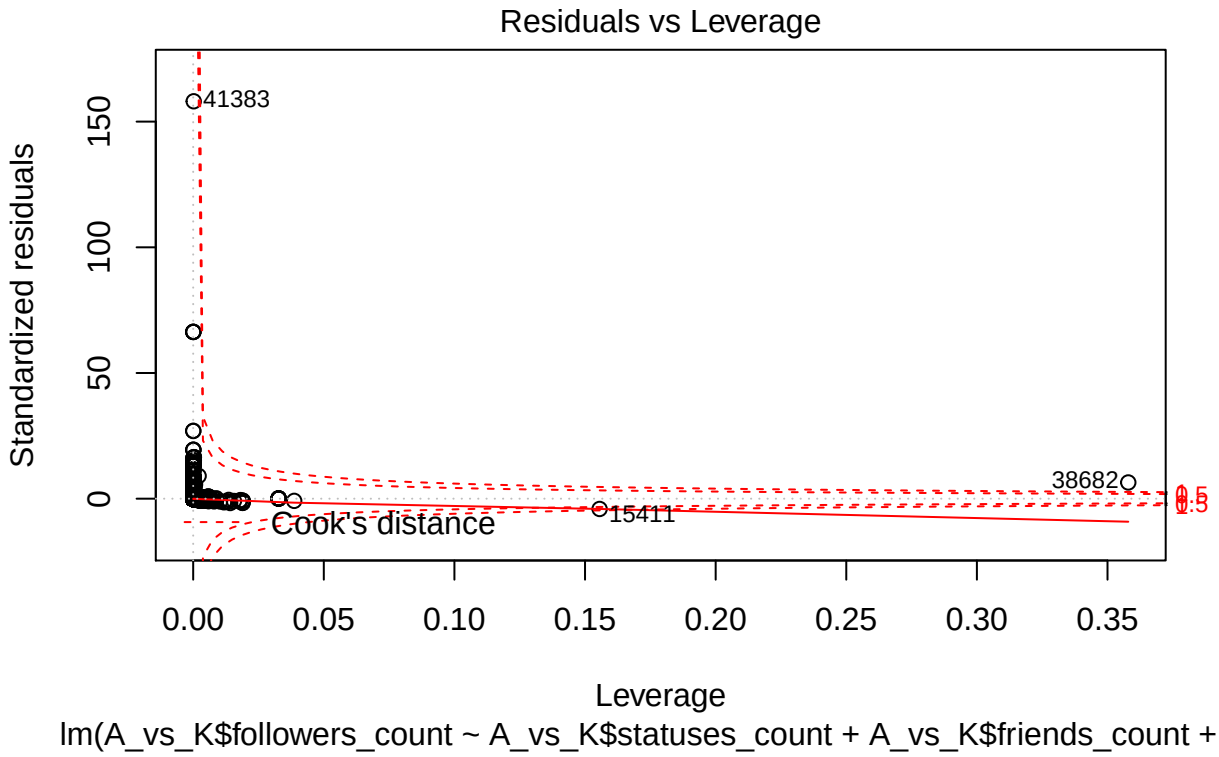
After collecting the live data and cleaning it, we were left with 534,481 observations and 34 columns. This data was collected from a total of 7 games during the NCAA March Madness Tournament.

The model we have chosen for our project tries to explain the relationship between the amount of followers the account has, in relation to the amount of statuses uploaded, the amount of people the user follows, and the amount of favorites the account received. By doing the games individually, we were able to see that all variables were significant within our model. However, The highest r-squared that was found was in the Auburn vs. Kentucky game [r2 = 0.008107]. Since all of our models had r-squared values under 1%, we can conclude that the models explain almost none of the variability of the response data around its mean. Furthermore, our smallest residual standard error was 104300 on 55256 degrees of freedom. This means that the points of our data are far away from the fit. To better understand our model we plotted it using the plot() function in r. Our results below are based off the following model: lm(followers_count ~ statuses_count + friends_count + favourites_count)




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(A_vs_K$followers_count ~ A_vs_K$statuses_count + A_vs_K$friends_count + .


Scale−Location
√|Standardized residuals|
Fitted values
lm(A_vs_K$followers_count ~ A_vs_K$statuses_count + A_vs_K$friends_count + .



Our results show us that there is are clear patterns in the Residual vs Fitted, Scale-Location, and Residuals vs Leverage plots. This potentially tells us that a linear model for this data may not be the best choice. Also, the Q-Q plot indicates that the residuals are not normally distributed. These results however do seem to make sense, since the amount of followers on an account can be random, even if the account is inactive.

4 Discussion

From the results in our shiny app users are able to see how many tweets were sent out during different points in the different games that we extracted data from. When viewing this line graph, we also included points that are able to be clicked on in order to display a couple of tweets that were sent at that time. This allows users to view what may have been happening at that time in the game to understand why that number of tweet may have been sent out. We also created a summary tab that holds our linear regression model which shows the effect that "statuses_count", "favorite_count", and "friends_count" had on "followers_count". This is a model that we created to try and determine if there was a way to predict a person's follower count, but this model yielded low R-squared numbers, indicating that this is not a particularly strong model in determining someone's followers count - this is understandable because there are a lot of factors that go into a person's follower count. Our final tab contained our word bubble which displayed the most frequent word that was tweeted during each individual game. This also allows users to hover over each word to see the exact number of times it was used in tweets during that game. All of these results achieve the goal of what we wanted to accomplish in order to create a comprehensive way to understand how twitter is used during the NCAA College Basketball March Madness Tournament in 2019. Our app gives people a way to see how twitter is used during different games in the tournament in order to see what kind of events may cause a spike or decline in twitter usage. These results also fulfilled the overall goal of the project by demonstrating our knowledge of R and ability to properly use its capabilities with real data and then to implement it in a way that any user would be able to navigate.

5 Conclusion

In conclusion, our group set out to use Twitter API in order to stream live tweets during the 2019 NCAA March Madness Tournament. The goal we laid was to create a visualization for people to be able to see how Twitter was used during these high profile events. This is exactly what we created using R and the shiny app to create a comprehensive app in order for people to navigate the Twitter usage during the games that we selected to display. The line graph, word cloud, and summary all provide a look into how users were interacting with Twitter while watching the March Madness Tournament.

6 Appendix

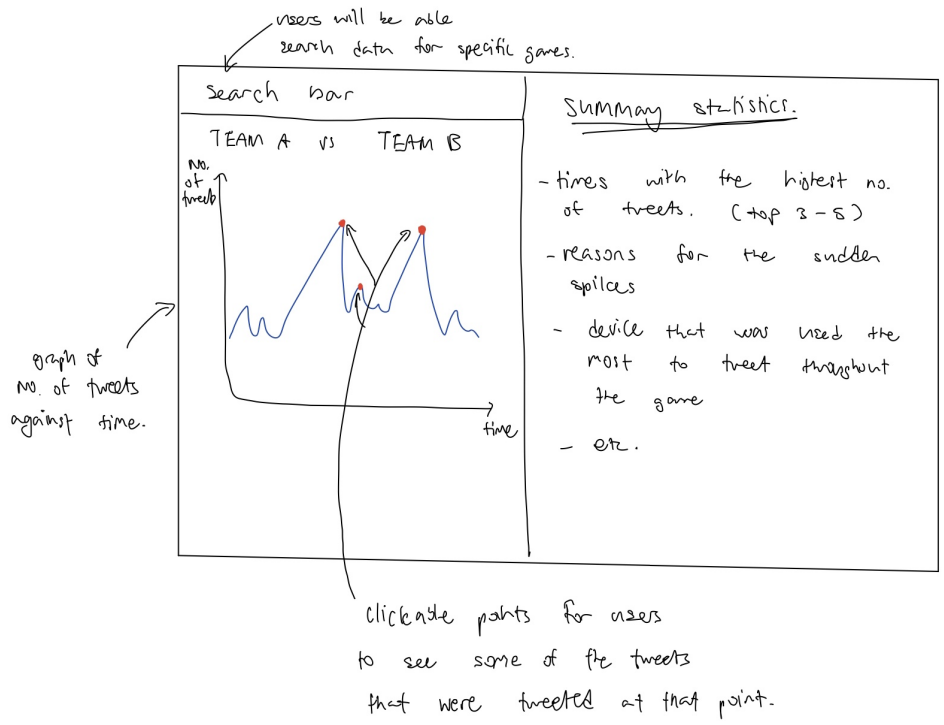


Figure 6: Heres an example of how we want our Shiny interface to look like

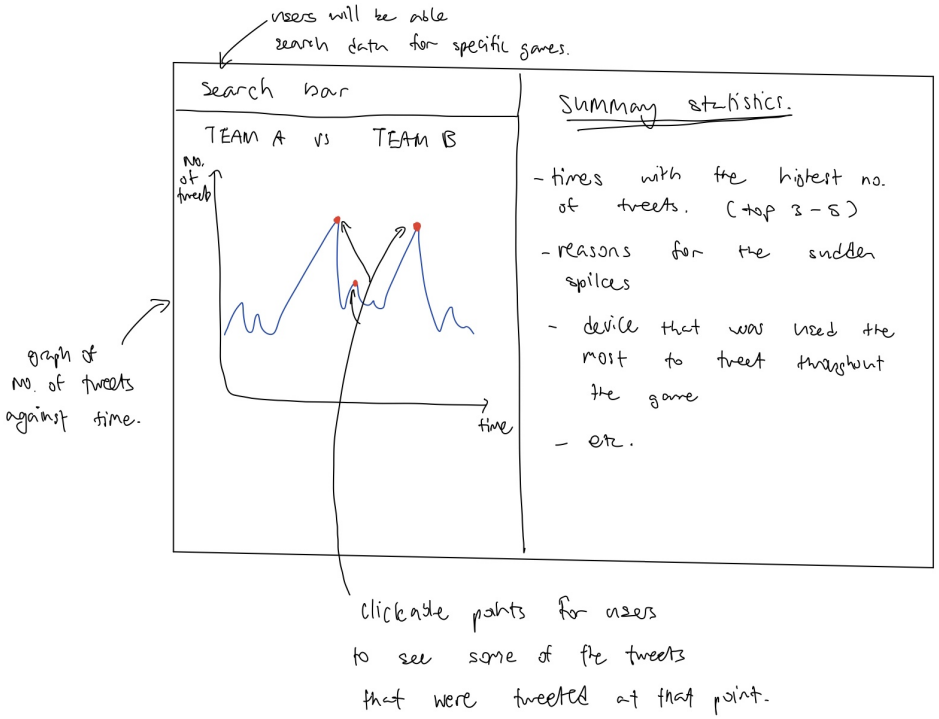


Figure 7: Heres a sample graph of how we want it to be displayed in our Shiny app

7 References


```
@article{kearney2016rtweet,
  title={rtweet: Collecting Twitter data},
  author={Kearney, Michael W},
  journal={Comprehensive R Archive Network. Available at: https://cran. r-project. org/package= rtweet}
}

@article{go2009twitter,
  title={Twitter sentiment classification using distant supervision},
  author={Go, Alec and Bhayani, Richa and Huang, Lei},
  journal={CS224N Project Report, Stanford},
  volume={1},
  number={12},
  pages={2009},
  year={2009}
}

@article{benhardus2013streaming,
  title={Streaming trend detection in twitter},
  author={Benhardus, James and Kalita, Jugal},
  journal={International Journal of Web Based Communities},
  volume={9},
  number={1},
  pages={122--139},
  year={2013},
  publisher={Inderscience Publishers Ltd}
}

@Manual{ggplot2-package,
    title = {ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics},
    author = {Hadley Wickham, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske
              Takahashi, Claus Wilke, Kara Woo},
    year = {2018},
    note = {R package version 3.1.0},
    url = {https://cran.r-project.org/web/packages=ggplot2},
}

@Manual{tidyr:2019,
    title = {tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions},
    author = {Hadley Wickham and Lionel Henry},
    year = {2019},
    note = {R package version 0.8.3},
    url = {https://cran.r-project.org/web/package=tidyr},
}

@Manual{dplyr:2018,
    title = {dplyr: A Grammar of Data Manipulation},
    author = {Hadley Wickham and Romain François and Lionel Henry and Kirill Müller},
    year = {2018},
    note = {R package version 0.7.7},
    url = {https://cran.r-project.org/web/package=dplyr},
```

```
}

@Manual{Matrix:2019,
    title = {Matrix: Sparse and Dense Matrix Classes and Methods},
    author = {Douglas Bates, Martin Maechler, Timothy A. Davis, Jens Oehlschlägel, Jason Riedy },
    year = {2019},
    note = {R package version   1.2-16},
    url = {https://cran.r-project.org/web/package=Matrix},
}

@Article{plyr-package,
    title = {The Split-Apply-Combine Strategy for Data Analysis},
    author = {Hadley Wickham},
    journal = {Journal of Statistical Software},
    year = {2011},
    volume = {40},
    number = {1},
    pages = {1--29},
    url = {http://www.jstatsoft.org/v40/i01/},
  }

@Manual{MatrixModels:2015,
    title = {MatrixModels: Modelling with Sparse And Dense Matrices},
    author = {Douglas Bates and Martin Maechler},
    year = {2015},
    note = {R package version 0.4-1},
    url = {https://CRAN.R-project.org/package=MatrixModels},
}

@Manual{Stats,
    title = {Stats: R Statistical Functions},
    author = {{R Core Team}},
    organization = {R Foundation for Statistical Computing},
    address = {Vienna, Austria},
    year = {2018},
    url = {https://www.R-project.org/},
}

@Manual{leaps:2017,
    title = {leaps: Regression Subset Selection},
    author = {Thomas Lumley based on Fortran code by Alan Miller},
    year = {2017},
    note = {R package version 3.0},
    url = {https://CRAN.R-project.org/package=leaps},
}

@article{wickham2014tidy,
  title={Tidy data},
  author={Wickham, Hadley and others},
  journal={Journal of Statistical Software},
  volume={59},
  number={10},
  pages={1--23},
```

```
  year={2014},
  publisher={Foundation for Open Access Statistics}
}

@Manual{stringR-package,
    title = {stringr: Simple, Consistent Wrappers for Common String Operations},
    author = {Hadley Wickham},
    year = {2018},
    note = {http://stringr.tidyverse.org, https://github.com/tidyverse/stringr},
  }
}

@Article{tidytext-tool,
    title = {tidytext: Text Mining and Analysis Using Tidy Data
      Principles in R},
    author = {Julia Silge and David Robinson},
    doi = {10.21105/joss.00037},
    url = {http://dx.doi.org/10.21105/joss.00037},
    year = {2016},
    publisher = {The Open Journal},
    volume = {1},
    number = {3},
    journal = {JOSS},
  }

@Manual{rtweet-package,
    title = {rtweet: Collecting Twitter Data},
    author = {Michael W. Kearney},
    year = {2018},
    note = {R package version 0.6.7},
    url = {https://cran.r-project.org/package=rtweet},
  }

@Manual{httpuv-package,
    title = {httpuv: HTTP and WebSocket Server Library},
    author = {Joe Cheng and Hector Corrada Bravo and Jeroen Ooms and Winston Chang},
    year = {2019},
    note = {R package version 1.5.1},
    url = {https://CRAN.R-project.org/package=httpuv},
  }

@Manual{baseR-package,
    title = {R: A Language and Environment for Statistical Computing},
    author = {{R Core Team}},
    organization = {R Foundation for Statistical Computing},
    address = {Vienna, Austria},
    year = {2018},
    url = {https://www.R-project.org/},
  }

@Manual{ggthemes-package,
    title = {ggthemes: Extra Themes, Scales and Geoms for 'ggplot2'},
    author = {Jeffrey B. Arnold},
```

```
    year = {2019},
    note = {R package version 4.1.1},
    url = {http://github.com/jrnold/ggthemes},
  }

@Manual{wordcloud2-package,
    title = {wordcloud2: Create Word Cloud by 'htmlwidget'},
    author = {Dawei Lang and Guan-tin Chien},
    year = {2018},
    note = {R package version 0.2.1},
    url = {https://github.com/lchiffon/wordcloud2},
  }

@Manual{tm-package,
    title = {tm: Text Mining Package},
    author = { Ingo Feinerer, Kurt Hornik, and David Meyer},
    year = {2018},
    note = {R package version 0.7-6},
    url = {https://github.com/lchiffon/wordcloud2},
  }
}
```

Educational Links:

Anderson, B. (2018). Winning over Fans: How Sports Teams Use Live-Tweeting to Maximize Engagement [PDF file]. Journalism & Media Analytics Elon University. Retrieved from https://www.elon.edu/u/academics/communications/journal/wp-content/uploads/sites/153/2018/05/06_Anderson_Livetweeting.pdf

Jeyakumar, K. & AR. Mohamed, S. & Sridhar, S. (2017). Twitter Sports: Real Time Detection of Key Events from Sports Tweets [PDF file]. Retrieved from scholarpublishing.org/index.php/TMLAI/article/download/3729/2275/

Corney, D. & Martin, C. & Göker, A. (2014) Spot the Ball: Detecting Sports Events on Twitter. In: de Rijke M. et al. (eds) Advances in Information Retrieval [PDF file]. Retrieved from https://link.springer.com/content/pdf/10.1007%2F978-3-319-06028-6.pdf

Murphy, G. (2018). Twitter Changes the Live TV Sports Viewing Experience [Web]. Retrieved from https://marketing.twitter.com/na/en/insights/twitter-changes-the-live-tv-sports-viewing-experience.html/

Gluck, J. (2018). The hidden Twitter relationships of NBA teams [Web]. CNS Data Lab. Retrieved from https://cnsmaryland.org/2018/06/03/the-relationships-between-nba-teams-twitter-accounts/

Arnold, Jeffrey B. 2019. *Ggthemes: Extra Themes, Scales and Geoms for 'Ggplot2'.* http://github.com/jrnold/ggthemes.

Bates, Douglas, and Martin Maechler. 2015. *MatrixModels: Modelling with Sparse and Dense Matrices.* https://CRAN.R-project.org/package=MatrixModels.

Cheng, Joe, Hector Corrada Bravo, Jeroen Ooms, and Winston Chang. 2019. *Httpuv: HTTP and Websocket Server Library.* https://CRAN.R-project.org/package=httpuv.

Douglas Bates, Timothy A. Davis, Martin Maechler. 2019. *Matrix: Sparse and Dense Matrix Classes and Methods.* https://cran.r-project.org/web/package=Matrix.

Fortran code by Alan Miller, Thomas Lumley based on. 2017. *Leaps: Regression Subset Selection.* https://CRAN.R-project.org/package=leaps.

Hadley Wickham, Lionel Henry, Winston Chang. 2018. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics.* https://cran.r-project.org/web/packages=ggplot2.

Ingo Feinerer, Kurt Hornik, and David Meyer. 2018. *Tm: Text Mining Package*. https://github.com/lchiffon/wordcloud2.

Kearney, Michael W. 2018. *Rtweet: Collecting Twitter Data*. https://cran.r-project.org/package=rtweet.

Lang, Dawei, and Guan-tin Chien. 2018. *Wordcloud2: Create Word Cloud by 'Htmlwidget'*. https://github.com/lchiffon/wordcloud2.

Ooms, Jeroen. 2018. *Hunspell: High-Performance Stemmer, Tokenizer, and Spell Checker*.

R Core Team. 2018a. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

———. 2018b. *Stats: R Statistical Functions*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Silge, Julia, and David Robinson. 2016. "Tidytext: Text Mining and Analysis Using Tidy Data Principles in R." *JOSS* 1 (3). The Open Journal. https://doi.org/10.21105/joss.00037.

Wickham, Hadley. 2011. "The Split-Apply-Combine Strategy for Data Analysis." *Journal of Statistical Software* 40 (1): 1–29. http://www.jstatsoft.org/v40/i01/.

———. 2018. *Stringr: Simple, Consistent Wrappers for Common String Operations*.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2018. *Dplyr: A Grammar of Data Manipulation*. https://cran.r-project.org/web/package=dplyr.

Wickham, Hadley, and Lionel Henry. 2019. *Tidyr: Easily Tidy Data with 'Spread()' and 'Gather()' Functions*. https://cran.r-project.org/web/package=tidyr.