Beating election polls with *Twitter*
A visualization study

Suzanne Wetstein

Research Paper Business Analytics
Supervised by: Dr. Bartek Knapik

In partial fulfillment for the degree of:
MSc Business Analytics
Vrije Universiteit Amsterdam

July 6, 2016

Abstract

In this paper we compare the *Twitter* popularity of US presidential candidates in 2016 with primary results and election projections. For this purpose, we have collected Tweets containing some relevant search terms for one week with the *Twitter* Search API. These Tweets are then processed and only Tweets containing a geolocation are kept. The geolocated Tweets are used in making choropleth maps of the US by state and by county. The visualizations are made in *R* with packages like `ggplot2` and `tmap` and the code can be found in the appendix. The *Twitter* maps are compared to primary results and election projections, and we conclude that *Twitter* popularity is very different from the election projection. Some explanations for this difference and possible extensions of this study are discussed.

Contents

1 Introduction

Social media play a continuously more important role in our society. Topics discussed online reflect the interests of the community, and the US presidential elections are extensively discussed on *Twitter*. In this study we will uncover the social media popularity of US presidential candidates and compare this with primary outcomes and election projections. The results will be visualized on a map of the US.

1.1 *Twitter*

Twitter[1] is a free social media service on which registered members can post messages that consist of a maximum of 140 characters. These short broadcasted messages are called Tweets. Tweets are publicly available when searched for, but only users "following" the messenger see the messages in their message display called the timeline. Users on *Twitter* have a username starting with an @ followed by a self chosen name, like @realDonaldTrump. To connect Tweets to a general topic, like Donald Trump, members can add hashtags to a keyword in their post. This hashtag then works like a meta tag and is expressed as #trump. About 1% of tweets has a geolocation attached to it, this can be the coordinates of the home profile location of the user or if enabled the GPS location of the user's smartphone.

1.2 Visualizations on maps

To make vizualizations on maps using the geolocation data of a Tweet we use *R*[2]. *R* is a free software environment for statistical computing and graphics. Into *R* a lot of packages can be imported to aid in making data visualizations on maps. The most important packages we will use and describe in this paper are `ggmap`, `ggplot2`, `sp`, `maps`, `maptools` and `tmap`.

1.3 The US presidential elections

The US presidential elections are held every four years and are currently in progress. A replacement of Barack Obama (who has served his final term) will be in place starting January 20, 2017. At this time there are three candidates left, namely Donald Trump (Republican), Hillary Clinton (Democrat) and Bernie Sanders (Democrat). Donald Trump is already set as

[1] www.twitter.com

[2] www.r-project.org

the presidential candidate for the Republicans and it is almost certain that Hillary Clinton will become the presidential candidate for the Democratic party.

1.4 Paper overview

In this study the popularity of presidential candidates on *Twitter* will be compared with US election primary results and election projections, to see whether there is a difference in predicted outcomes of the US elections in polls and with *Twitter*. We will first focus on giving some background information on using *Twitter* data, making visualizations on maps and on the US presidential election voting system in section 2. Next we will present some related work in section 3 and then start with gathering, exploring and preparing our own data in section 4. In section 5 we will explain how we will visualize this data on maps. Our mapped results will be compared with previous voting outcomes in section 6. We conclude with a discussion and some ideas for further research in section 7.

2 Background

Data can be collected from *Twitter* and *Google Maps* using their application-programming interface (API). An API is a set of programming instructions for accessing a specific tool. Companies release their APIs to the public so that developers can design products that are powered by its service. In this section an overview will be given of the *Twitter* and *Google Maps* APIs, some *R* packages will be discussed and the procedure of the US presidential elections will be explained.

2.1 The *Twitter* API

The best way to access *Twitter* data depends on the type and amount of data you are trying to collect. There are two *Twitter* APIs: the REST API and the Streaming API, which both have their own characteristics. For social media monitoring and analytics, e.g. following the US elections, the latter is most appropriate.

2.1.1 OAuth authentication

Use of the *Twitter* API requires OAuth authentication. OAuth is an authentication protocol that allows users to approve the interaction of one aplication with another without giving away their password. To access the *Twitter* API you need to register with them.[3] Once registered, you will automatically be given a "Consumer Key", "Consumer Secret", "Access Token" and an "Access Token Secret". These credentials are needed to be able to establish a connection with the *Twitter* API from any programming language. We will explain how to use these credentials in section 4.

2.1.2 The REST API

The REST API is the most common way to access *Twitter* data. Using the credentials obtained via OAuth, your application makes requests to *Twitter* for specific data, and the response is available in a standard format: JSON (Javascript Object Notation). The REST API can be used to conduct singular searches, read user profile information, or post Tweets. Use of the REST API is subject to rate limits imposed by *Twitter*. These rate limits impact the ability to get full coverage streams for monitoring and analytics

[3] via apps.twitter.com

use cases. Applications are allowed to make 180 authenticated requests to an (unknown proportion) sampling of the REST API per 15 minutes.

The Search API, which is dedicated to running searches against the index of recent Tweets, is part of the REST API. This API behaves similarly to the Search feature available on *Twitter*, it searches against the sampling of Tweets in the REST API published in the past 7 days. The Search API is focused on relevance and not on completeness, because it is primarily intended to help surface interesting Tweets that are happening now. Since the API is focused on relevance, the sampling proportion is not given by *Twitter*.[23][18]

2.1.3 The Streaming API

The Streaming API gives near-real-time access to *Twitter* data. This API searches (without a rate limit per time frame) against a 1% random sampling of all Tweets, with the possibility to filter on up to 400 keywords or hashtags. Therefore the Streaming API can be used to monitor or process Tweets in real-time. There are three streaming endpoints: public streams, user streams and site streams. Public streams are streams of the public data flowing through *Twitter*. This stream can be used to follow specific users or topics or for data mining purposes. User streams containly almost all of the data corresponding to a single user's view. Site streams are the multi-user version of user streams. The Streaming API is not rate limited per time frame because data is pushed to the users' server as it comes in. [19]

2.2 Packages in *R*

The most used package in the field of data visualization is `ggplot2`[4], which can work with point and polygon data. This package can be used to divide a map of the US into states by using polygon data. Further the `ggmap`[5] or `maps`[6] package is needed to get static maps from online sources like the *Google Maps*[7] API discussed in section 2.2.1. The `sp` package contains classes and methods to work with spatial data, like points, lines, polygons and grids. This can be used to work with coordinates and spatial selection. Furthermore, the `maptools` package is used to read in geographic data and the `tmap` package is used to make thematic maps. `Tmap` is also useful for

[4] Manual: `https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf`

[5] Manual: `https://cran.r-project.org/web/packages/ggmap/ggmap.pdf`

[6] Manual: `https://cran.r-project.org/web/packages/maps/index.html`

[7] www.maps.google.com

reading and processing shapefiles. Many tutorials, like [20] and [13] on mapping in R using these packages are available. In sections 4 and 5 the application of all these packages on our data is discussed.

2.2.1 The *Google Maps* API

Google Maps is an online service that gives detailed information about geographical locations around the world. It offers road maps, and street, aerial and satellite views of many places. *Google Maps* has many different types of APIs, which are categorized by platform. There are Web, Android, iOS and HTTP Web service APIs. We will be working with the *Google* Static Maps API from the Web API category, to get simple map images with minimal code. Further from the HTTP Web service APIs, we will work with the *Google Maps* Geocoding API which converts between adresses and geographic coordinates and with the *Google Places* API Web Service which gives up-to-date information about millions of locations.[9]

2.3 The US presidential elections

The US elections consist of a complicated maze of caucasus, primaries and delegates. Here we will give you an introduction to some political jargon and an overview of the system.

2.3.1 Political jargon

In the US elections a lot of political jargon is used, here we give you an overview:
Caucus: A caucus is a meeting where registered members of a party in a city or county gather to show support for a candidate.
Primary: An election in which members of a party choose which candidates will run for office in the general election.
Delegate: A person authorized to act as a representitative.
Red state: A US state that typically votes Republican.
Blue state: A US state that typically votes Democratic.
Swing state: A US state that is closely split between Democratic and Republican voters, also called a ”Purple state”.
Electoral College: A group of 538 officials chosen by voters in each state to elect the American president.[11]

2.3.2 An overview of the voting system

The election process begins with candidates announcing their intention to run. In January to June of the election year the caucuses and primaries take place. In these elections states choose the political parties' nominees for the general election. At stake in each primary or caucus is a certain number of delegates, the candidate who receives a majority of his or her party's delegates wins the party nomination. During the nominating conventions from July till September each political party selects their party nominee. At this time the nominee also announces a Vice Presidential running mate. After nomination the candidates campaign accross the country to explain their views and plans to voters and to participate in debates with candidates from other parties. At the general elections in November, Americans cast their vote for the president. However, these votes do (not directly) determine the president but the composition of the Electoral College. To win the election, a candidate must receive the majority of electoral votes.[24]

3 Related work

This study consists of three parts: social media (*Twitter*), visualizations on maps (*Google Maps*) and elections. As little related work was found combining all three aspects we will also discuss some related work combining only two out of the three.

3.1 Election predictions with *Twitter*

A meta-analysis on electoral prediction from *Twitter* data [8] reveals that *Twitters'* presumed predictive power regarding electoral prediction may have been exaggerated. It concludes that although social media might provide a glimpse on the outcomes, research has not yet provided strong evidence that social media popularity can replace traditional polls. Other studies [14],[12], also find that electoral predictions using the published research methods on *Twitter* data are not better than chance. However, [15] says that the *Twitter* political index has generally correlated with well-known polls on the US elections, and that the index must not be seen as a substitute for polling but as a new sort of information that there was no way of accessing before.

There are two main methods of election predictions based on *Twitter* data. The first is making a prediction based solely on tweet counts mentioning a candidate or party as proposed in [22]. This method is appealing as it is easy to implement and fast. Also, they find that the number of messages mentioning a party reflect the election result. However, [12] also studied the same election and criticized the results in [22]. Especially the decisions regarding the selection of parties and the period of data collection were criticized. The second method of predicting elections from *Twitter* data is using sentiment analysis. Most of the sentiment analysis here is limited to lexicon-based sentiment analysis. Only [21],[1] did an analysis containing trained sentiment classifiers in a machine learning approach. Both studies find that predictions based on sentiment analysis give good results. Lexicon-based sentiment analysis has problems with the performance of the lexicon-based classifier. A study [7] uncovered that the precision of this classifier for one presidential candidate was much higher than for the other. Therefore results achieved with lexicon-based sentiment analysis should be carefully dealt with.

Some studies are done combining a type of election, *Twitter* data and geographical maps. For instance [3], studies the elimination of contestants in the American Idol TV show as an electoral phenomenon. They provide evidence that *Twitter* activity correlates with the contestants ranking and allows anticipation of the voting outcome. They also show that the fraction of tweets that contain geolocation allows them to map each contestants' fanbase and that strong regional polarizations occur in the fanbases.

3.2 Data visualization on maps

There are many blogs written about visualizing data on maps. However, not many scientific papers are dedicated to this topic. Eric Fischer is a software developer who has made a lot of data visualizations. He made the Geotagger's World Atlas, a series of maps linking interesting places around the world. He also made data visualizations using *flickr*[8] and *Twitter* data. Two of his *Twitter* maps are shown in figure 1 and 2. These figures contain true works of art made with real data. However, in figure 2 it can be seen that this data can be biased. The disproportionally high number of *Twitter* users in the Netherlands and England skewes the overall picture.

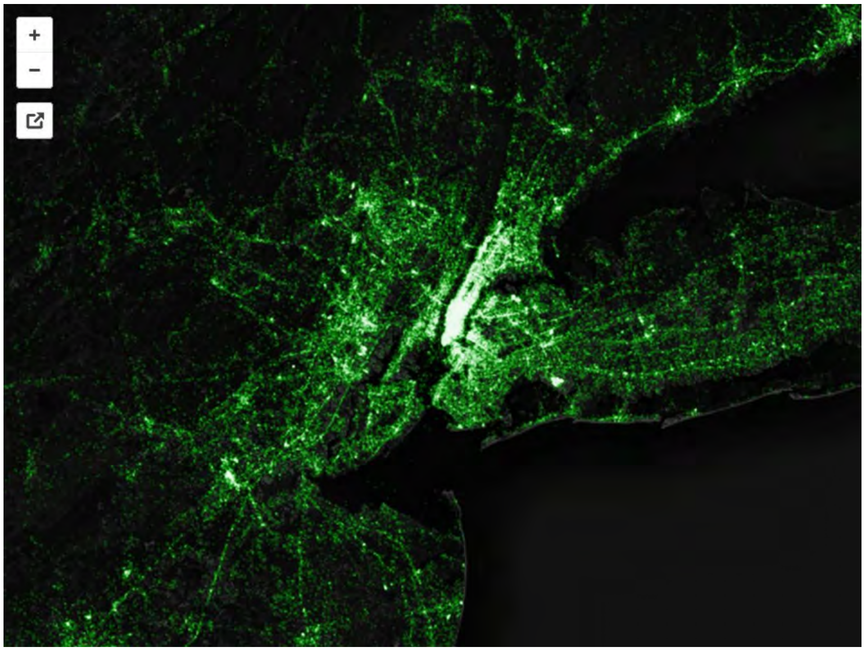


Figure 1: A *Twitter* map of every tweet in New York City from mid 2010 till 2014.[6]

[8] www.flickr.com

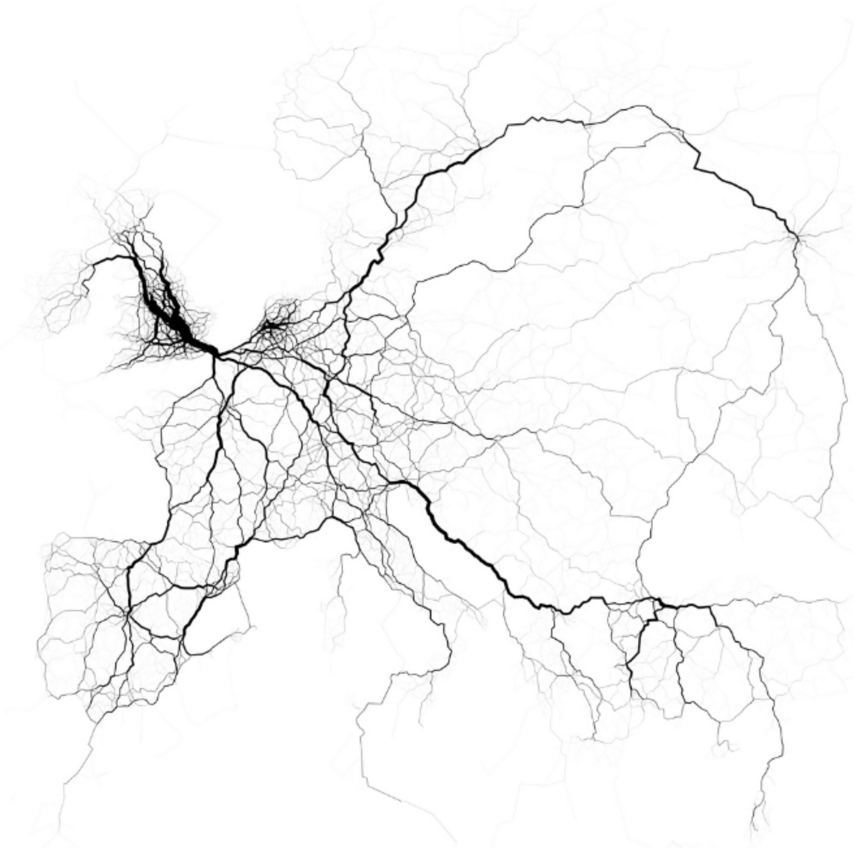


Figure 2: A *Twitter* map of 60,000 tweet-based trips in Europe in August 2011.[5]

To make sure that data visualizations really present unbiased data a lot of analyses have to been done on the dataset. In [17] a data visualization specialist shows how visualizations can lead to wrong conclusions when the data is not studied carefully before a visual map is made.

4 Data gathering, exploration and preparation

4.1 Data collection

We collected *Twitter* data containing the words ”Trump”, ”Clinton” and ”Bernie” from June 7, 2016 till June 16, 2016. This was done by first registering with *Twitter* and then making an OAuth signature as described in section 2.1.1. The *R* code for making this signature can be found in appendix A.1. Now we had to decide whether to gather data with the REST API or the Streaming API. Both methods were tried and the *R* codes for the REST API and the Streaming API can be found in appendix A.2 and A.3, respectively. As the REST API is rate limited and directed towards completeness, we decided to further use the Streaming API. Another benefit of this API is the fact that the data is realtime, making this study more relevant. The data collected was gathered from *Twitter* realtime and saved into one file per hour. The total collected data before processing consisted of 66GB of data.

4.2 Data exploration and preparation

To process the collected data all files were loaded into *R*, the Tweets were parsed and a dataset was created for each candidate. For each candidate only the Tweets with a geolocation attached to it were kept, other Tweets were disregarded. The geolocation used here is the user's home profile location and not their cellphone location, as the home profile location is connected to where (in which state) a person can vote. The *R* code with which we have created these datasets can be found in appendix A.4. These datasets consist of three attributes: home profile location longitude, latitude and the frequency with which this exact location tweeted something about the presidential candidate. As can be seen in table 1, the dataset with Tweets in which Trump was mentioned is the largest dataset. Trump was mentioned 142953 times in total, in 73122 unique locations. Clinton and Sanders were mentioned only 40465 and 33068 times, respectively.

To explore these datasets further we plotted the locations in each dataset on a map. The plots were made using the `ggmap` and `ggplot2` package in *R*. A map was downloaded and then the Tweets were added to this map as layers, the code can be found in appendix A.5. The maps can be seen in figure 3. We can clearly see that more Tweets mention Trump than Clinton or Sanders from comparing figure 3a, 3b and 3c. We can also see that in some states there are more Tweets about presidential candidates

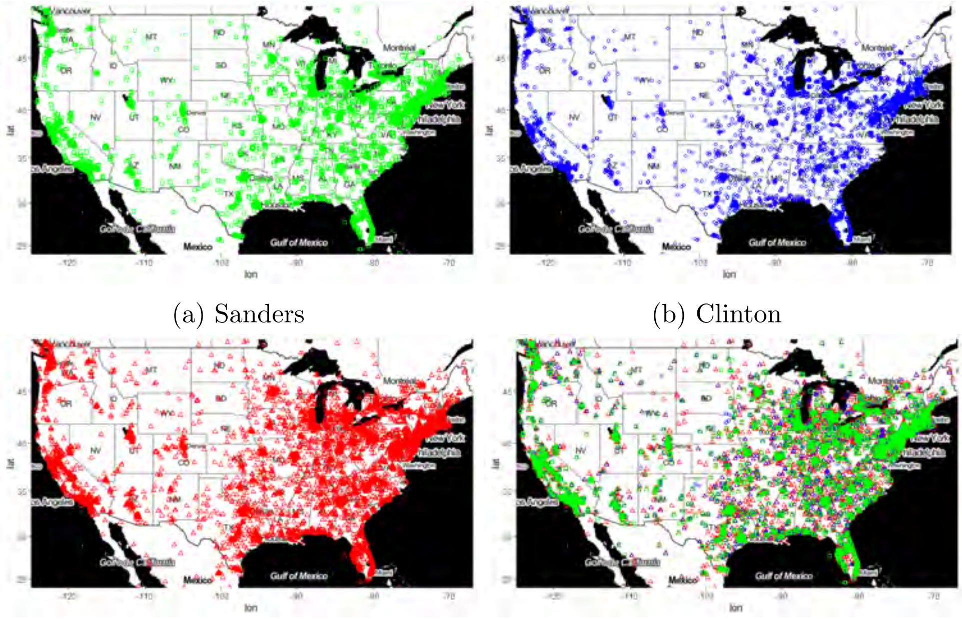
Candidate	Unique locations	Total Tweets
Trump	73122	142953
Clinton	26468	40465
Sanders	20193	33068

Table 1: Characteristics of the attributes of the presidential candidates' datasets.

than in other states. This can have multiple reasons, for example that these states have more inhabitants, or that the inhabitants Tweet more or that the inhabitants care more about the presidential elections.

4.3 Translating coordinates to states and counties

Presidential election results or forecasts are usually depicted on maps per state or per county. By giving the state, or county, a different shade in proportion to the percentage of the votes for a specific candidate the result is portrayed. This kind of map is called a choropleth map, an easy way to show how a measurement varies across a geographic area. To make a choropleth map with our data, we need to translate our longitudes and latitudes to states and counties. To do this we wrote a function in *R* using the `sp`, `maps` and `maptools` packages. The function prepares the spatial points (our longitude and latitute coordinates) and spatial polygons (the state (or county) shapes) to be used by the `over()` function in the `sp` package to calculate the intersection of points and polygons. Finally, as output the function returns the name of a state (or county) in which a latitude/longitude combination lies. The *R* code for this function can be found in appendix A.6 for the latitude/longitude to state conversion, however for conversion to counties all that has to be done is replace ”state” by ”county” in the function. The implementation of the function for states, to make a dataset containing the attributes state and the frequency each candidate is mentioned in that state is given in appendix A.7. While making this dataset we also throw away the geographic locations that do not belong to a state, that is to say we throw away the Tweets from outside the USA. The implementation for counties is a bit more complicated as some counties have the same names in different states, which is why we used the FIPS county code of the counties instead of their names. The FIPS code is a five-digit Federal Information Processing Standard code which uniquely identifies the counties in the US. The *R* implementation for counties can be found in appendix A.8.



(a) Sanders

(b) Clinton

(c) Trump

(d) All three candidates

Figure 3: Plots of all Tweets collected mentioning a certain candidate. Plot d is a combination of plot a-c, in which the Trump Tweets are the bottom layer, the Clinton Tweets are the middle layer and the Sanders Tweets are the top layer. This means that if one location mentions Trump and Sanders, in plot d only Sanders' green icon is visible.

5 Modelling

We have now created two datasets which we want to model: one with the amount of Tweets per candidate per state and one with the amount of Tweets per candidate per county. To make choropleth maps in *R* with this data we use the `tmap` package.

5.1 What to map

We first need to decide what we want to map. The data gives us the amount of Tweets per candidate per state, so we can calculate the difference in Tweets between two candidates for each state. Also, we can calculate the percentual difference between the amount of Tweets for the candidates in each state. We have decided to look at the percentage differences in Tweet amounts, because in some states there are much more Tweets than in oter states. Furthermore, we have decided to look at the difference between Sanders and Clinton, so we can compare this with the primary results, to look at the difference between Clinton and Trump, so we can compare this with the Electoral College projections and to look at the difference between the Democrats (Clinton and Sanders) and the Republicans (Trump), as people that have tweeted about Sanders will most likely vote for Clinton, as the only remaining Democratic candidate. In conclusion, we will map the percentage difference in Tweet amounts by state and by county between:

- Clinton and Sanders
- Clinton and Trump
- the Democrats (Clinton and Sanders) and the Republicans (Trump)

5.2 Collecting geographic data

Now that we know what to map we need to collect some geographic data for the USA. We used shapefiles with a scale of 1:5,000,000 from the United States Census Bureau. Shapefiles are easy to use because of their small filesize. We downloaded the shapefiles for states and counties and placed these in our working directory so that we can call the file with a command in *R*. The shapefile contains data for all areas of the US, thus also for outlying areas like Alaska, Hawaii and some small islands. We remove all these outlying areas to be able to use the shapefile for the lower 49 states.

5.3 Combining *Twitter* data with geographic data

The next step is to combine the Tweet data with the geographic data. In the case of states (counties) this is done by merging the data based on the state name (FIPS county code). One thing to check here is that we need to make sure that the state names and FIPS county codes are stored in exactly the same way in both datasets. We make sure that the key columns in the datasets are the same values, the same data type and in the same order. When this is the case, we join the two files.

5.4 Creating the map

To create the map we first make our own color palette, in which we use the usual colors for the presidential candidates: red for Trump, blue for Clinton and green for Sanders. The mapping is done with the `tmap` package, we specify the geodata file to be mapped and set which column to use for mapping color values. The resulting maps can be found in section 6. The *R* code for the modelling and mapping of the Tweets per state and per county can be found in appendix A.9 and A.10, respectively.

6 Results

To be able to compare the *Twitter* popularity of US presidential candidates to (expected) voting outcomes, we gathered some results from the primaries and caucuses of the Democratic party and we gathered some Electoral College projections for the race between the Democrats and the Republicans. The election outcomes in the primaries for Clinton versus Sanders can be found in figure 4a and the election projection for the Democrats versus the Republicans can be found in figure 5a.

6.1 Clinton v. Sanders

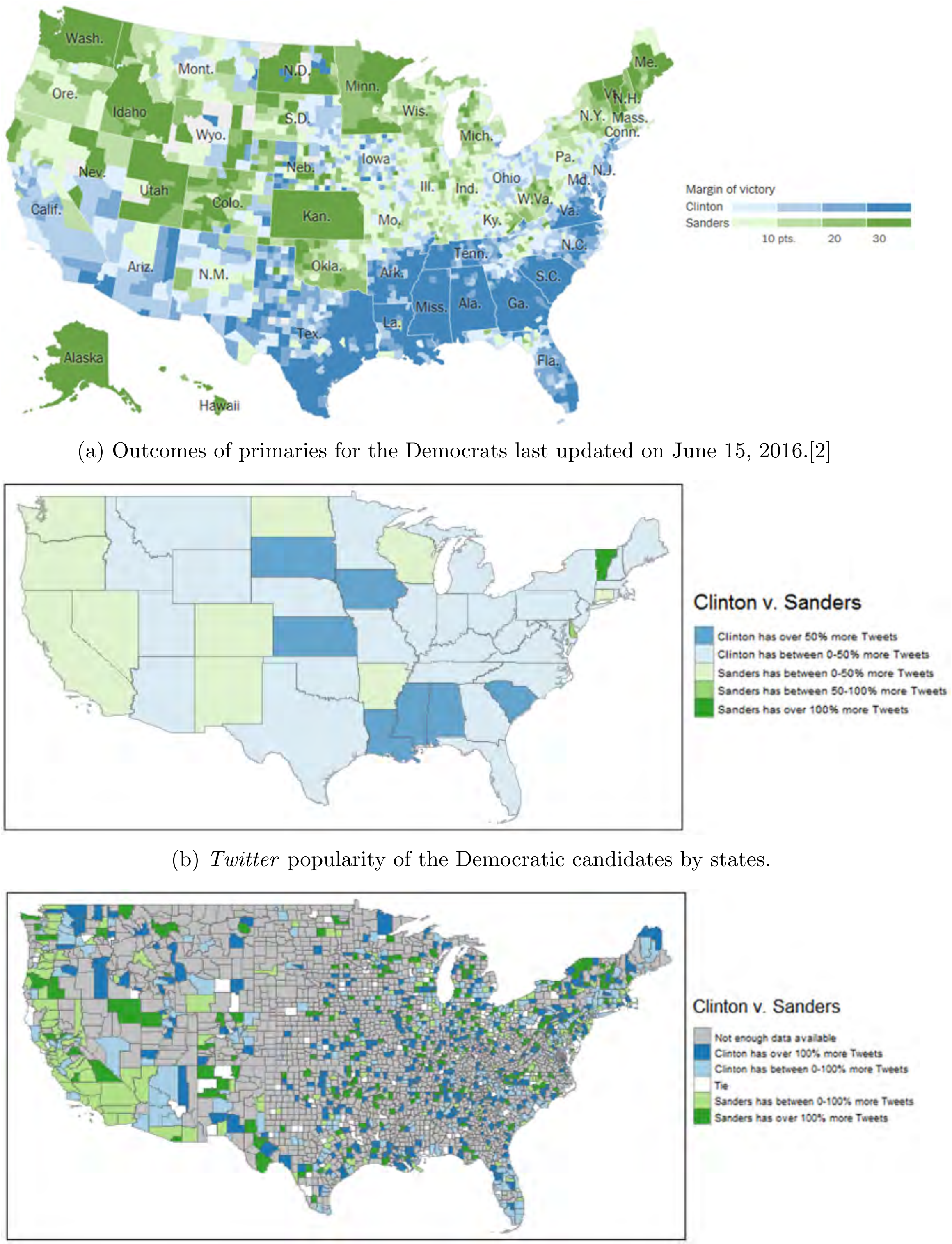
First we look at the *Twitter* and voting results between Bernie Sanders and Hillary Clinton in figure 4. When comparing the primary outcomes in figure 4a with the *Twitter* popularity of candidates by state in figure 4b we notice that a lot more states are in favour of Clinton by *Twitter* popularity. A state that stands out is Kansas, the voting results in this state show a strong preference for Sanders but the *Twitter* popularity shows a strong preference for Clinton. When comparing the *Twitter* results per county in figure 4c to the primary results we see that this is hard to compare as in many counties there is not enough data available. For the counties that do have enough data we see that in *Twitter* popularity most of the counties at the west coast Tweet about Sanders, and in the voting results Sanders is also preferred on the upper west coast but starting in California more people vote for Clinton. The voting results show that most states in the southeast of the US vote for Clinton, in the *Twitter* results we cannot really see this due to missing data. We can see that Clinton has more *Twitter* popularity in Florida.

6.2 Clinton v. Trump

In figure 5 we look at an election projection and the *Twitter* popularity of Clinton and Trump. What immediately stands out is that in the *Twitter* popularity by state in figure 5b Trump is more popular in every single state. When comparing the *Twitter* popularity by state with the election projection in figure 5a we see that states with a lower *Twitter* popularity for Trump (colored light red) are sometimes strong or even solid GOP in the election projection. The *Twitter* popularity by state is very different from the election projection. When comparing the *Twitter* popularity by county in figure 5c to the election projection we see that these are also very different. On the west coast Trump has a much higher *Twitter* popularity while the election projection shows that this region is expected to vote for the Democratic party. By county Trump also has a much higher *Twitter* popularity all over the US while the election projection shows that more states are in favour of Clinton.

6.3 Democrats v. Republicans

Since Clinton is quite definitely going to be the candidate for the Democratic party people who have Tweeted about Sanders need to vote for someone else. As people in the US usually vote for the same party every election it is likely that Sanders' Tweeters will vote for Hillary Clinton. In figure 6 we again see the election projection between the Democrats and the Republicans now compared with the *Twitter* popularity of the Democrats (Sanders and Clinton added up) and the Republicans (Trump). When comparing the election projection in figure 6a with the *Twitter* popularity by state in figure 6b we see that the Republicans have a far greater *Twitter* popularity than their popularity in the election projection. Only one state, Vermont, Tweets more about the Democratic candidates than about Trump and this state also is also predicted to vote for the Democratic party. In Kentucky Trump has the highest *Twitter* popularity in comparison to the Democratic candidates and the election pojection also strongly predicts that this state will vote for the Republican party. It is hard to draw any conclusions from the comparison of the election projection and the *Twitter* popularity by county in figure 6c as there is not enough data in many counties. We can see that the lower west coast Tweets more about Trump while the election projection indicates this region will vote Democratic. Moreover, many of the counties have a higher *Twitter* popularity for Trump while the election projection predicts that the Democrats will win the vote in more states.

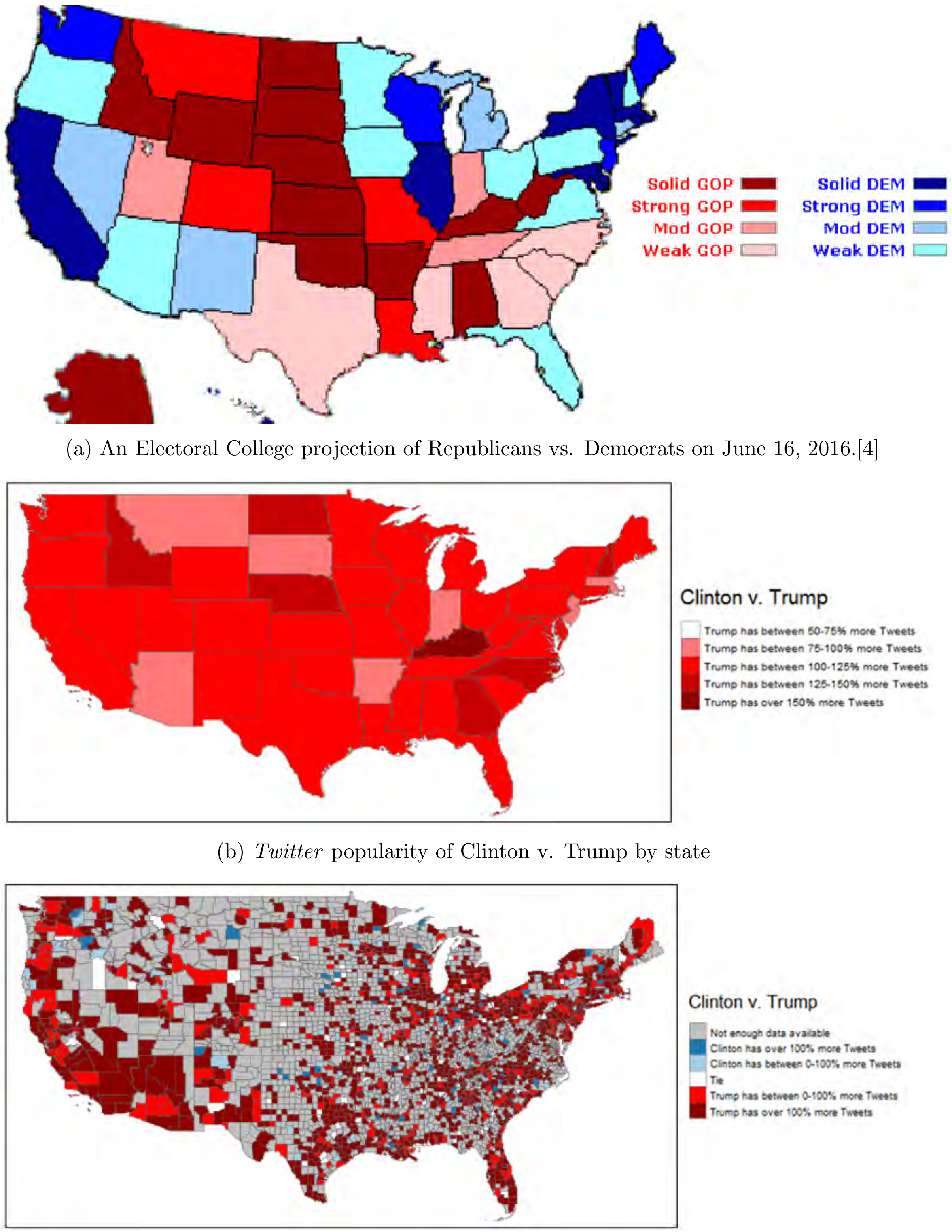

Wash.
Mont.
N.D.
Minn.
Ore.
Idaho
S.D.
Wis.
Wyo.
Mich.
Me.
Vt.
N.H.
N.Y.
Mass.
Conn.
Nev.
Neb.
Iowa
Pa.
N.J.
Utah
Ill.
Ind.
Ohio
Md.
Calif.
Colo.
W.Va.
Va.
Kan.
Mo.
Ky.
Tenn.
N.C.
Ariz.
N.M.
Okla.
Ark.
S.C.
Miss.
Ala.
Ga.
La.
Tex.
Fla.
Alaska
Hawaii
Margin of victory
Clinton
Sanders
10 pts.
20
30
Clinton v. Sanders
Clinton has over 50% more Tweets
Clinton has between 0-50% more Tweets
Sanders has between 0-50% more Tweets
Sanders has between 50-100% more Tweets
Sanders has over 100% more Tweets
Clinton v. Sanders
Not enough data available
Clinton has over 100% more Tweets
Clinton has between 0-100% more Tweets
Tie
Sanders has between 0-100% more Tweets
Sanders has over 100% more Tweets

(a) Outcomes of primaries for the Democrats last updated on June 15, 2016.[2]

(b) *Twitter* popularity of the Democratic candidates by states.

(c) *Twitter* popularity of the Democratic candidates by county.

Figure 4: *Twitter* and voting results between the Democratic candidates: Bernie Sanders and Hillary Clinton.

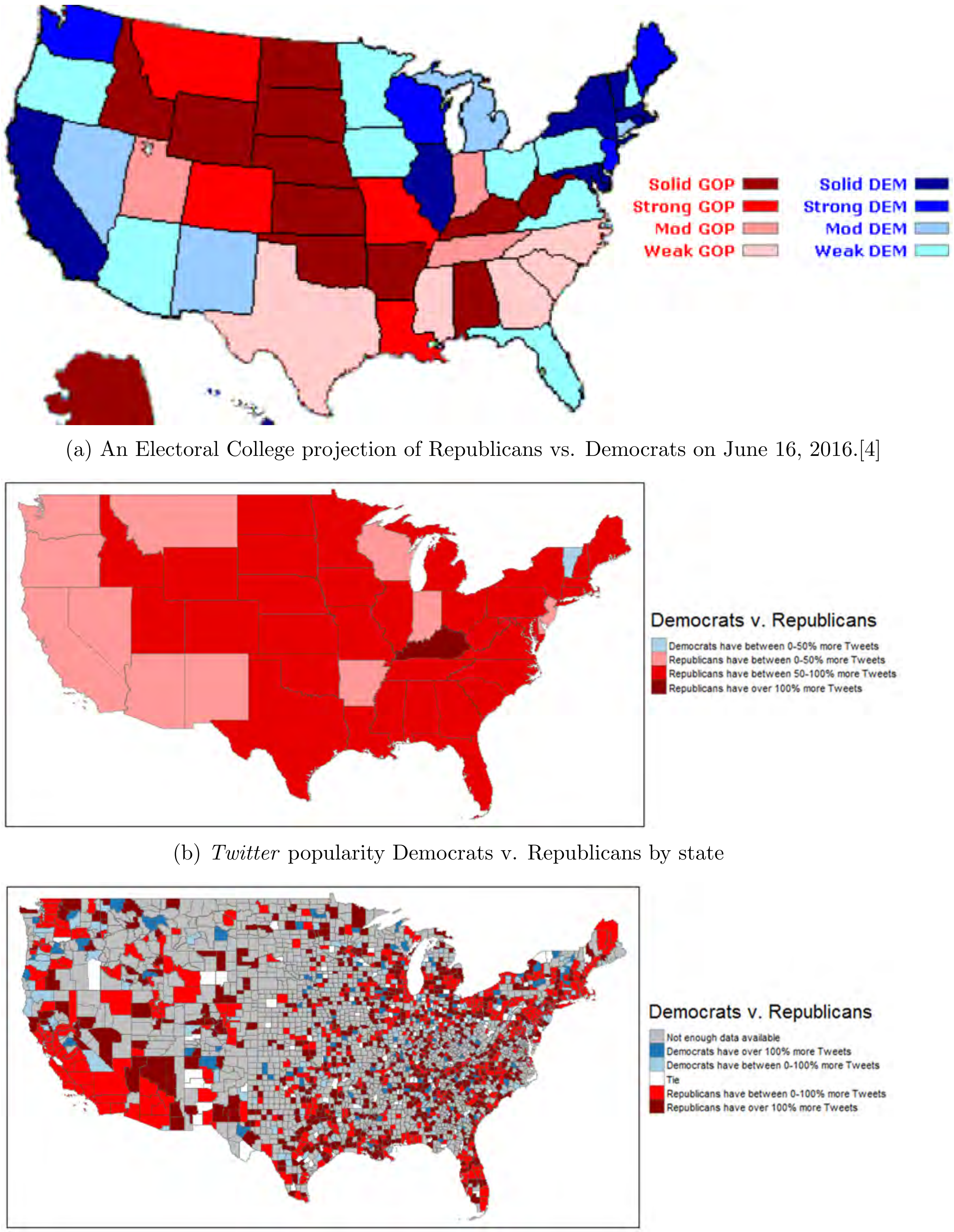

Solid GOP
Strong GOP
Mod GOP
Weak GOP
Solid DEM
Strong DEM
Mod DEM
Weak DEM
Clinton v. Trump
Trump has between 50-75% more Tweets
Trump has between 75-100% more Tweets
Trump has between 100-125% more Tweets
Trump has between 125-150% more Tweets
Trump has over 150% more Tweets
Clinton v. Trump
Not enough data available
Clinton has over 100% more Tweets
Clinton has between 0-100% more Tweets
Tie
Trump has between 0-100% more Tweets
Trump has over 100% more Tweets

(a) An Electoral College projection of Republicans vs. Democrats on June 16, 2016.[4]

(b) *Twitter* popularity of Clinton v. Trump by state

(c) *Twitter* popularity of Clinton v. Trump by county

Figure 5: *Twitter* popularity and an election projection for the race between Donald Trump and Hillary Clinton.


Solid GOP
Strong GOP
Mod GOP
Weak GOP
Solid DEM
Strong DEM
Mod DEM
Weak DEM
Democrats v. Republicans
Democrats have between 0-50% more Tweets
Republicans have between 0-50% more Tweets
Republicans have between 50-100% more Tweets
Republicans have over 100% more Tweets
Democrats v. Republicans
Not enough data available
Democrats have over 100% more Tweets
Democrats have between 0-100% more Tweets
Tie
Republicans have between 0-100% more Tweets
Republicans have over 100% more Tweets

(a) An Electoral College projection of Republicans vs. Democrats on June 16, 2016.[4]

(b) *Twitter* popularity Democrats v. Republicans by state

(c) *Twitter* popularity Democrats v. Republicans by county

Figure 6: *Twitter* popularity and an election projection for the race between the Democrats and the Republicans.

7 Conclusion and discussion

We have found that the *Twitter* popularity of US presidential candidates cannot predict the outcomes of the primary election and that it is not comparable with the election projection of the Electoral College. However, there are many other ways of measuring *Twitter* popularity which might influence these results. Some changes to our methods are discussed below.

7.1 Ways of collecting more (relevant) data

A possible extension to the present work would be collecting more Tweets about the elections by using more search terms. In this study we focused on the most used terms for each candidate, being ”Trump”, ”Clinton” and ”Bernie”. Adding search terms like ”Democrat”,”Republican”,”US election” and more might collect additional relevant data.

In this study we collected *Twitter* data for only one week. Our results for the *Twitter* popularity by county showed that in many counties there was not enough data. For further study it would be much better to collect data for a larger time frame. This will bring some new challenges to light as the data collected for one week was already 66GB. However, after dropping unnecessary attributes and keeping only data with geolocation we were left with only a few MB.

7.2 Sentiment analysis

Our results showed that Trump is immensely popular on *Twitter*. In this study we counted all Tweets about the candidates as popularity, as bad publicity is also publicity. However, people Tweeting about Trump might not necessarily vote for him and the Tweets could all be very negative. An extension of this study could be to analyze the sentiment of the collected Tweets, either by using trained sentiment classifiers in a machine learning approach or by using a lexicon-based sentiment analysis.

References

[1] Adam Bermingham and Alan F. Smeaton, *On Using Twitter to Monitor Political Sentiment and Predict Election Results*, paper presented at the Workshop on Sentiment Analysis where AI meets Psychology, 2011.

[2] Matthew Bloch et al., *Detailed Maps of Where Trump, Cruz, Clinton and Sanders Have Won*, [ONLINE] Available at: `http://www.nytimes.com/elections/2016/national-results-map`, New York Times, 2016.

[3] Fabio Ciulla et al., *Beating the news using social media: the case study of American Idol*, EPJ Data Science 1:8, 2012.

[4] Scott Elliott, *2016 Presidential elections*, [ONLINE] Available at: `http://www.electionprojection.com/presidential-elections.php`, 2016.

[5] Eric Fischer, *Shortest-path routing*, [ONLINE] Available at: `https://www.flickr.com/photos/walkingsf/sets/72157628993413851/with/6804680189`, Flickr, 2012.

[6] Eric Fischer, *Making the most detailed tweet map ever*, [ONLINE] Available at: `https://www.mapbox.com/blog/twitter-map-every-tweet/`, Mapbox, 2014.

[7] Daniel Gayo-Avello, *Don't turn social media into another "Literary Digest" poll*, Communications of the ACM 54 (10), 2011.

[8] Daniel Gayo-Avello, *A meta-analysis of state-of-the-art electoral prediction from Twitter data*, Social Science Computer Review, 2013.

[9] Google Developers, *Google Maps APIs*, [ONLINE] Available at: `https://developers.google.com/maps/get-started`, Google, 2016.

[10] Robert Harris, *#Election2016: US Presidential Candidate Twitter Buzz*, [ONLINE] Available at: `https://interactive.twitter.com/candidateRace16/?status=active`, Twitter, Inc., 2016.

[11] Nicholas Johnston, *American Political Jargon*, [ONLINE] Available at: `http://www.bloomberg.com/quicktake/american-political-jargon`, Bloomberg L.P., 2016.

[12] Andreas Jungherr et al., *Why the Pirate Party Won the German Election of 2009 or The Trouble With Predictions: A Response to Tumasjan, A., Sprenger, T. O., Sander, P. G., & Welpe, I. M. Predicting Elections With Twitter: What 140 Characters Reveal About Political Sentiment*, Social Science Computer Review, 2011.

[13] Mahbubul Majumder, *Displaying Maps and Spatial Data*, [ONLINE] Available at: http://www.unomaha.edu/mahbubulmajumder/data-science/fall-2014/lectures/06-display-spatial-data/06-display-spatial-data.html#/, 2014.

[14] Panagiotis T. Metaxas et al., *How (Not) To Predict Elections*, Proceedings of IEEE PASSAT/Conference on Social Computing, 2011.

[15] Martha T. Moore, *Twitter index tracks sentiment on Obama, Romney*, USA Today, 2012.

[16] The New York Times Co., *President Map*, [ONLINE] Available at: http://elections.nytimes.com/2012/results/president, New York Times, 2012.

[17] Jake Porway, *The Trials and Tribulations of Data Visualization for Good*, [ONLINE] Available at: https://marketsforgood.org/the-trials-and-tribulations-of-data-visualization-for-good/, DataKind, 2016.

[18] Samantha Quist, *Guide to the Twitter API - Part 2 of 3: An Overview of Twitter's Search API*, [ONLINE] Available at: https://blog.gnip.com/guide-to-the-twitter-api-part-2-of-3-an-overview-of-twitters-search-api, GNIP, Inc., 2011.

[19] Samantha Quist, *Guide to the Twitter API - Part 3 of 3: An Overview of Twitter's Streaming API*, [ONLINE] Available at: https://blog.gnip.com/guide-to-the-twitter-api-part-3-of-3-an-overview-of-twitters-streaming-api, GNIP, Inc., 2011.

[20] Zev Ross, *Mapping in R using the ggplot2 package*, [ONLINE] Available at: http://zevross.com/blog/2014/07/16/mapping-in-r-using-the-ggplot2-package/, ZevRoss, 2014.

[21] Tjong Kim Sang and Johan Bos, *Predicting the 2011 Dutch Senate Election Results with Twitter*, presented at the 13th Conference of the European Chapter of the Association for Computational Linguistics, 2012.

[22] Andranik Tumasjan et al., *Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment*, paper presented at the 4th International AAAI Conference on Weblogs and Social Media, 2010.

[23] Twitter, Inc., *Documentation*, [ONLINE] Available at: https://dev.twitter.com/overview/documentation, 2016.

[24] U.S. government, *Presidential Election Process*, [ONLINE] Available at: https://www.usa.gov/election, U.S. government's official web portal, 2016.

A *R* code

A.1 OAuth

```r
library(devtools)
library(streamR)
library(RCurl)
library(RJSONIO)
library(stringr)
library(ROAuth)

requestURL <- "https://api.twitter.com/oauth/request_token"
accessURL <- "https://api.twitter.com/oauth/access_token"
authURL <- "https://api.twitter.com/oauth/authorize"
consumerKey <- "ENTER_YOUR_CONSUMER_KEY_HERE"
consumerSecret <- "ENTER_YOUR_CONSUMER_SECRET_HERE"

OAuth <- OAuthFactory$new(consumerKey = consumerKey,
                             consumerSecret = consumerSecret,
                             requestURL = requestURL,
                             accessURL = accessURL,
                             authURL = authURL)

my_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl"))

### STOP HERE BEFORE RUNNING THE LINE BELOW!!! ###

# PART 2: Save the OAuth data to an .Rdata file
save(OAuth, file = "OAuth.Rdata")
```

A.2 Using the Search API

```r
libs <- c("twitteR", "ggmap", "ggplot2")
lapply(libs, library, character.only=TRUE)

consumer_key <- "ENTER_YOUR_CONSUMER_KEY_HERE" #This is my personal key from the app I created
      on apps.twitter.com
consumer_secret <- "ENTER_YOUR_CONSUMER_SECRET_HERE"
access_token <- "ENTER_YOUR_ACCESS_TOKEN_HERE"
access_secret <- "ENTER_YOUR_ACCESS_SECRET_HERE"
options(httr_oauth_cache=T) #This will enable the use of a local file to cache OAuth access
     credentials between R sessions.
setup_twitter_oauth(consumer_key,
                        consumer_secret,
                        access_token,
                        access_secret)

search_terms <- c("trump","clinton","bernie")

#Coordinates of Centre of the US
location <- "US"
longitude <- c("ENTER_LONGITUDE")
latitude <- c("ENTER_LATITUDE")
radius <- "30000km"
latlong <- paste(latitude,longitude,radius,sep=",")
latlong <- rep(latlong, length(search_terms))
search_terms <- as.data.frame(cbind(latlong, search_terms))
search_terms$search_terms <- as.character(search_terms$search_terms)
search_terms$latlong <- as.character(search_terms$latlong)
search_terms$location <- location

tweets <- data.frame()

for (i in 1:nrow(search_terms)){
  print(paste("Looking for",search_terms$search_terms[i], "in", search_terms[i,]$location))
  tw = searchTwitter(search_terms[i,]$search_terms,n=round(3000/nrow(search_terms)), geocode=search_terms[i,]$latlong)
  if (length(tw) == 0){
    print(paste("No tweets found for", search_terms$search_terms[i], "in", location))
  } else {
    tweets=rbind(twListToDF(tw), tweets)
  }
}

library(ROAuth)
library(plyr)

options(digits=10)
df <- count(tweets, c("longitude","latitude"))
df <- na.omit(df)
df$longitude <- as.numeric(df$longitude)
df$latitude <- as.numeric(df$latitude)

#Make the map
map <- get_map(location=c(lon=15.2551187,lat=54.5259614), zoom=4, maptype="hybrid")
finalmap <- ggmap(map)+geom_point(col="red", aes(x=longitude,y=latitude), data=df)
```

A.3 Gathering realtime *Twitter* data with the Streaming API

```
library(streamR)
load("C:/Users/Suzanne/Documents/OAuth.Rdata")
setwd("C:/Users/Suzanne/Documents/US Elections data")
filterStream(file.name =paste(format(Sys.time(), "%Y-%m-%d %H.%M"),".json",sep=""), # Save tweets in a json file
             track = c("trump","clinton","bernie"), # Collect all tweets with trump, clinton, bernie or a combination
             #location = c(-31.266001, 27.636311, 39.869301, 81.008797), # latitude/longitude pairs providing southwest and northeast corners of the bounding box.
             timeout = 3600, # Keep connection alive for 1 hour
             oauth = OAuth) # Use the OAuth file as the OAuth credentials
```

A.4 Data preparation

```
setwd("C:/Users/Suzanne/Documents/US Elections data/")
files <- list.files("C:/Users/Suzanne/Documents/US Elections data/")

library(streamR)
library(plyr)
library(data.table)
```

```

for (file in files){
  Parsed <- parseTweets(file, simplify = FALSE)
  Trump <- Parsed[grep("Trump",Parsed$text,ignore.case = TRUE),]
  Clinton <- Parsed[grep("clinton",Parsed$text,ignore.case = TRUE),]
  Sanders <- Parsed[grep("bernie",Parsed$text,ignore.case = TRUE),]

  options(digits=10)
  Trump <- count(Trump, c("place_lon","place_lat"))
  Trump <- na.omit(Trump)
  Trump$place_lon <- as.numeric(Trump$place_lon)
  Trump$place_lat <- as.numeric(Trump$place_lat)

  Clinton <- count(Clinton, c("place_lon","place_lat"))
  Clinton <- na.omit(Clinton)
  Clinton$place_lon <- as.numeric(Clinton$place_lon)
  Clinton$place_lat <- as.numeric(Clinton$place_lat)

  Sanders <- count(Sanders, c("place_lon","place_lat"))
  Sanders <- na.omit(Sanders)
  Sanders$place_lon <- as.numeric(Sanders$place_lon)
  Sanders$place_lat <- as.numeric(Sanders$place_lat)

  if (!exists("TrumpTweets")){
    TrumpTweets <- Trump
  }
  if (exists("TrumpTweets")){
  TrumpTweets <- rbind(TrumpTweets, Trump)
  }

  if (!exists("ClintonTweets")){
    ClintonTweets <- Clinton
  }
  if (exists("ClintonTweets")){
    ClintonTweets <- rbind(ClintonTweets, Clinton)
  }

  if (!exists("SandersTweets")){
    SandersTweets <- Sanders
  }
  if (exists("SandersTweets")){
    SandersTweets <- rbind(SandersTweets, Sanders)
  }

  rm(Parsed)
  rm(Trump)
  rm(Clinton)
  rm(Sanders)
}
```

A.5 Visualizing on maps

```
library(ggplot2)
library(ggmap)

#Download the map
map <- get_map(location=c(-125.0011,23.9493,-66.9326,49.5904), source="stamen", maptype="toner")

Sandersmap <- ggmap(map)+geom_point(col="green", shape=0, aes(x=place_lon,y=place_lat),data=SandersTweets)
Clintonmap <- ggmap(map)+geom_point(col="blue", shape=1, aes(x=place_lon,y=place_lat),data=ClintonTweets)
Trumpmap <- ggmap(map)+geom_point(col="red", shape=2, aes(x=place_lon,y=place_lat),data=TrumpTweets)
Totalmap <- ggmap(map)+geom_point(col="red", shape=2, aes(x=place_lon,y=place_lat),data=TrumpTweets)+geom_point(col="blue", shape=1, aes(x=place_lon,y=place_lat),data=ClintonTweets)+geom_point(col="green", shape=0, aes(x=place_lon,y=place_lat),data=SandersTweets)
```

A.6 Function for conversion of longitude/latitude to states (or counties)

```r
#Convert latitude / longitude to states

library(sp)
library(maps)
library(maptools)

# The input for this function, lonlat, is a dataframe in which the first column contains the longitude (in degrees) and the second column contains the latitude (in degrees).

lonlat2state <- function(lonlat) {
  # Prepare SpatialPolygons object with one SpatialPolygon per state (plus DC, minus HI & AK)
  states <- map(’state’, fill=TRUE, col="transparent", plot=FALSE)
  IDs <- sapply(strsplit(states$names, ":"), function(x) x[1])
  states_sp <- map2SpatialPolygons(states, IDs=IDs,
                                   proj4string=CRS("+proj=longlat +datum=WGS84"))

  # Convert lonlat to a SpatialPoints object
  pointsSP <- SpatialPoints(lonlat,
                            proj4string=CRS("+proj=longlat +datum=WGS84"))

  # Use ’over’ to get indices of the Polygons object containing each point
  indices <- over(pointsSP, states_sp)

  # Return the State names of the Polygons object containing each point
  stateName <- sapply(states_sp@polygons, function(x) x@ID)
  stateName[indices]
}
```

A.7 Dataset preparation for states

```r
# Implement the function for states
Input_trump <- data.frame(x = TrumpTweets$place_lon, y = TrumpTweets$place_lat)
Input_clinton <- data.frame(x = ClintonTweets$place_lon, y = ClintonTweets$place_lat)
Input_sanders <- data.frame(x = SandersTweets$place_lon, y = SandersTweets$place_lat)
States_trump <- data.frame(State = latlong2state(Input_trump), Trump_frequency = TrumpTweets$freq)
States_clinton <- data.frame(State = latlong2state(Input_clinton), Clinton_frequency = ClintonTweets$freq)
States_sanders <- data.frame(State = latlong2state(Input_sanders), Sanders_frequency = SandersTweets$freq)

#Add rows of same state
States_trump <- ddply(States_trump, .(State), summarize, Trump_frequency = sum(Trump_frequency))
States_trump <- States_trump[c(-50),]
States_clinton <- ddply(States_clinton, .(State), summarize, Clinton_frequency = sum(Clinton_frequency))
States_clinton <- States_clinton[c(-50),]
States_sanders <- ddply(States_sanders, .(State), summarize, Sanders_frequency = sum(Sanders_frequency))
States_sanders <- States_sanders[c(-50),]
States_all <- data.frame(State = States_trump$State, Trump = States_trump$Trump_frequency, Clinton = States_clinton$Clinton_frequency, Sanders = States_sanders$Sanders_frequency)

#Output states
write.csv(States_all,file="Statedata.csv")
```

A.8 Dataset preparation for counties

```r
Input_trump <- data.frame(x = TrumpTweets$place_lon, y = TrumpTweets$place_lat)
Input_clinton <- data.frame(x = ClintonTweets$place_lon, y = ClintonTweets$place_lat)
Input_sanders <- data.frame(x = SandersTweets$place_lon, y = SandersTweets$place_lat)
```

```
Counties_trump <- data.frame(county = latlong2county(Input_trump), Trump_frequency = TrumpTweets$freq)
Counties_clinton <- data.frame(county = latlong2county(Input_clinton), Clinton_frequency = ClintonTweets$freq)
Counties_sanders <- data.frame(county = latlong2county(Input_sanders), Sanders_frequency = SandersTweets$freq)

#Add rows of same county and throw away the Tweets that do not belong to counties (from outside USA)
Counties_trump <- ddply(Counties_trump, .(county), summarize, Trump_frequency = sum(Trump_frequency))
Counties_trump <- Counties_trump[c(-1520),]
Counties_clinton <- ddply(Counties_clinton, .(county), summarize, Clinton_frequency = sum(Clinton_frequency))
Counties_clinton <- Counties_clinton[c(-1090),]
Counties_sanders <- ddply(Counties_sanders, .(county), summarize, Sanders_frequency = sum(Sanders_frequency))
Counties_sanders <- Counties_sanders[c(-1001),]

#Solve problem of the candidates being mentioned in a different number of counties
Counties_all <- merge(x = Counties_trump, y = Counties_clinton, by = "county", all=TRUE)
Counties_all <- merge(x = Counties_all, y = Counties_sanders, by = "county", all=TRUE)
Counties_all[is.na(Counties_all)] <- 0

#Convert county names to fips codes
Counties_all$polyname <- Counties_all$county
Counties_all$polyname <- as.character(Counties_all$polyname)
Counties_all$polyname[Counties_all$polyname=="florida,okaloosa"] <- "florida,okaloosa:main"
Counties_all$polyname[Counties_all$polyname=="north carolina,currituck"] <- "north carolina,currituck:main"
Counties_all$polyname[Counties_all$polyname=="texas,galveston"] <- "texas,galveston:main"
Counties_all$polyname[Counties_all$polyname=="washington,pierce"] <- "washington,pierce:main"

Counties_all_test <- merge(x = county.fips, y = Counties_all, by = "polyname", all=TRUE)
Counties_all_test <- Counties_all_test[,c(-1,-3)]
Counties_all_test[is.na(Counties_all_test)] <- 0

#Merge double fips codes
Counties_all_test <- ddply(Counties_all_test, .(fips), summarize, Trump_frequency = sum(Trump_frequency), Clinton_frequency = sum(Clinton_frequency), Sanders_frequency = sum(Sanders_frequency))

#Output Counties
write.csv(Counties_all_test,file="countydatafips.csv")
```

A.9 Modelling and mapping by state

```
setwd("C:/Users/Suzanne/Documents/US Elections data")

library(tmap)

#Use loaded States_all file and create data to map
Data <- States_all
Data$TrumpMarginVotes <- Data$Trump - Data$Clinton
Data$TrumpPct <- (Data$Trump - Data$Clinton)/(Data$Trump + Data$Clinton)
Data$ClintonPct <- (Data$Clinton - Data$Trump)/(Data$Trump + Data$Clinton)
Data$TrumpMarginPctgPoints <- Data$TrumpPct - Data$ClintonPct

#Get geographic data
usstateshape <- "cb_2015_us_state_5m/cb_2015_us_state_5m.shp"
usstate <- read_shape(file = usstateshape)

# remove outlying areas like alaska and hawaii
usstates <- usstate[!usstate@data$STATEFP=="02" & !usstate@data$STATEFP=="15" & !usstate@data$STATEFP=="60" & !usstate@data$STATEFP=="81" & !usstate@data$STATEFP=="64" & !usstate@data$STATEFP=="66" & !usstate@data$STATEFP=="84" & !usstate@data$STATEFP=="86" & !usstate@data$STATEFP=="67" & !usstate@data$STATEFP=="89" & !usstate@data$STATEFP=="68" & !usstate@data$STATEFP=="71" & !usstate@data$STATEFP=="76" & !usstate@data$STATEFP=="69" & !usstate@data$STATEFP=="70" & !usstate@data$STATEFP=="95" & !usstate@data$STATEFP=="72" & !usstate@data$STATEFP=="74" & !usstate@data$STATEFP=="78" & !usstate@data$STATEFP=="79",]
#qtm(usstates)
```

```r

#Test the data formats
str(usstates@data$NAME)
str(Data$State)
usstates@data$NAME <- as.character(usstates@data$NAME)
Data$State <- as.character(Data$State)
Data$State2 <- usstates@data$NAME

#Solve problem with capitals vs no capitals
usstates@data$NAME <- tolower(usstates@data$NAME)
usstates <- usstates[order(usstates@data$NAME),]
Data <- Data[order(Data$State),]
identical(usstates@data$NAME,Data$State) #only continue if yes!!!

Map <- append_data(usstates, Data, key.shp = "NAME", key.data = "State")

#Create color palette
myownpaletteBernieClinton <- c("#1F78B4","#A6CEE3","#FFFFFF","#B2DF8A","#33A02C")
myownpaletteTrumpClinton <- c("#1F78B4","#A6CEE3","#FFFFFF","red","red4")

#Create map
tm_shape(Map)+
  tm_fill("TrumpMarginPctgPoints", title ="", labels = c("Democrats have between 0-50% more Tweets","Republicans have between 0-50% more Tweets","Republicans have between 50-100% more Tweets","Republicans have over 100% more Tweets"), palette = myownpaletteTrumpClinton)+
  tm_borders(alpha=.5)+
  tm_layout("Democrats v. Republicans",legend.outside = TRUE,legend.outside.position = "bottom")
```

A.10 Modelling and mapping by county

```r
setwd("C:/Users/Suzanne/Documents/US Elections data")

library(tmap)

#Use loaded States_all file and create data to map
Data <- Counties_all_test
Data$TrumpMarginVotes <- Data$Trump_frequency - Data$Clinton_frequency
Data$TrumpPct <- (Data$Trump_frequency - Data$Clinton_frequency)/(Data$Trump_frequency + Data$Clinton_frequency)
Data$ClintonPct <- (Data$Clinton_frequency - Data$Trump_frequency)/(Data$Trump_frequency + Data$Clinton_frequency)
Data$TrumpMarginPctgPoints <- Data$TrumpPct - Data$ClintonPct

#Get geographic data
uscountyshape <- "cb_2015_us_county_5m/cb_2015_us_county_5m.shp"
uscounty <- read_shape(file = uscountyshape)

# remove outlying areas like alaska and hawaii
uscounties <- uscounty[!uscounty@data$STATEFP=="02" & !uscounty@data$STATEFP=="15" & !uscounty@data$STATEFP=="60" & !uscounty@data$STATEFP=="81" & !uscounty@data$STATEFP=="64" & !uscounty@data$STATEFP=="66" & !uscounty@data$STATEFP=="84" & !uscounty@data$STATEFP=="86" & !uscounty@data$STATEFP=="67" & !uscounty@data$STATEFP=="89" & !uscounty@data$STATEFP=="68" & !uscounty@data$STATEFP=="71" & !uscounty@data$STATEFP=="76" & !uscounty@data$STATEFP=="69" & !uscounty@data$STATEFP=="70" & !uscounty@data$STATEFP=="95" & !uscounty@data$STATEFP=="72" & !uscounty@data$STATEFP=="74" & !uscounty@data$STATEFP=="78" & !uscounty@data$STATEFP=="79",]
#qtm(uscounties)

#Test the data formats
uscounties@data$fips <- paste(uscounties@data$STATEFP,uscounties@data$COUNTYFP,sep="")
uscounties@data$fips <- as.integer(uscounties@data$fips)

#Solve problem with missing fips codes
uscounties <- uscounties[order(uscounties@data$fips),]
Data <- Data[order(Data$fips),]
Data[is.na(Data)]<- -2.5 #For counties with no data
str(uscounties@data$fips)
str(Data$fips)
Map <- merge(x = uscounties@data, y = Data, by="fips", all=TRUE)
Map$Clinton_frequency[is.na(Map$Clinton_frequency)] <- 0
```

```r
Map$Trump_frequency[is.na(Map$Trump_frequency)] <- 0
Map[is.na(Map)] <- -2.5 #For counties with no data
Map2 <- append_data(uscounties, Map, key.shp = "fips", key.data = "fips")

#Create color palette
myownpaletteBernieClinton <- c("#C0C0C0","#1F78B4","#A6CEE3","#FFFFFF","#B2DF8A","#33A02C")
myownpaletteTrumpClinton <- c("#C0C0C0","#1F78B4","#A6CEE3","#FFFFFF","red","red4")
#Create static map
tm_shape(Map2)+
  tm_fill("TrumpMarginPctgPoints", title="", breaks = c(-3,-2,-1,0,0.0000001,1,2), labels=c("Not enough data available","Democrats have over 100% more Tweets","Democrats have between 0-100% more Tweets","Tie","Republicans have between 0-100% more Tweets","Republicans have over 100% more Tweets"), palette = myownpalette)+
  tm_borders(alpha=.5)+
  tm_layout("Democrats v. Republicans", frame=TRUE, legend.outside=TRUE, legend.outside.position = "bottom")
```