A Statistical Analysis of Network Data from Reddit

Abstract

Network structures are everywhere, from social networks to health epidemics. When making statistical models, it is important to be able to account for a network structure, since network data violates the assumption of independence that a regular linear model requires. In this paper, we explore descriptive statistics of networks, purely mathematical models of networks, and Exponential Random Graph Models (ERGMs) – a statistical modeling method that accounts for network structures. Finally, we construct two ERGMs from network datasets that we gathered from two politically charged subreddits: the subreddit for Donald Trump supporters and the subreddit for Bernie Sanders supporters. Using sentiment and text analysis of Reddit comments in our model, we are able to quantify how likely a user is to comment positively in a comments section that consists largely of positive comments, and vice-versa. We find that commenting positively in an already negative discussion is less likely than commenting negatively in an already negative discussion and vice versa. Additionally, we discuss structural and numerical characteristics of our networks, in the hope of better understanding how redditors interact with each other in these political environments.

1. Introduction

Networks are all around us. The concept of entities sharing links with each other is one that explains the structure behind many phenomena, from social groups to the spread of illnesses. Since this structure is so pervasive, statistical methods have been created in order to account for it when modeling network-structured phenomena. When examining network data, the typical assumptions required for a basic linear model are violated — the data points are not necessarily independent (for example, we would expect a group of friends to have similar characteristics, and so each cluster of data points would not be independent).

In our paper, we will apply statistical methods to network data that we gathered from Reddit.com. Reddit is a social news aggregator where users can post and comment on links, text, and images from around the internet. In particular, we examine the Reddit communities dedicated to discussing two political figures: Bernie Sanders and Donald Trump. Using Exponential Random Graph Models (ERGMs), we examine what factors make Reddit users within these communities more likely to interact with each other in an internet discussion.

We begin by introducing some basic network concepts, while also discussing our data gathering process and the construction of our networks. Then, we perform exploratory data analysis on the resulting network datasets. After this, we introduce some purely mathematical models of networks and compare our data to these models. Finally, we construct our statistical models (using ERGMs) and interpret them.

2. Constructing Our Network Data

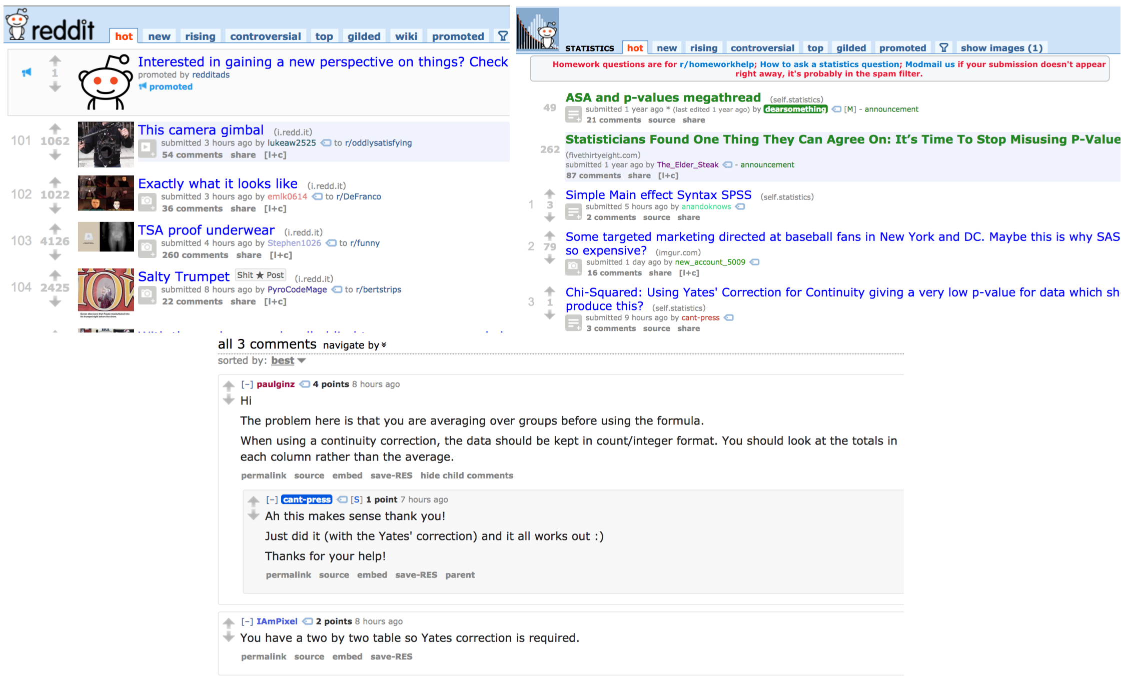
To introduce our dataset, we must explain two concepts: Reddit and networks. We will begin by discussing Reddit, and then we will introduce network terminology to formalize the structure of the site and the data we gathered from the site. Finally, we will describe our data scraping process and the limitations of our data.

2.1 Reddit

The website from which we gathered our dataset, Reddit, is a social news and content discussion site. The site's users (called "redditors") can create posts that either contain a link or plaintext, and other users can vote on these posts and change their position in the news feed by "upvoting" or "downvoting." Upvotes increase a post's score and move it up in the feed, whereas downvotes do the opposite. In addition to voting on a post, users can also add comments. Furthermore, users can reply to any comment on a post, creating a tree of comments that adds even more structure to the site. An image of the Reddit homepage is shown in the leftmost screenshot of Figure 1, and the comment structure for a Reddit post is shown in the rightmost screenshot.

Reddit is composed of many smaller communities, called "subreddits." Each subreddit pertains to a specific topic — some popular subreddits include "funny," "pics," "news," and "videos." Many subreddits develop into active communities in which users interact with each other frequently. These communities often generate their own customs, memes, and other pieces of internet culture. To denote a subreddit, it is common practice to write `/r/` followed by the name of the subreddit. This notation originated from the URLs of subreddits, which are always given by `reddit.com/r/subreddit_name`. Thus, to refer to the "videos" subreddit, we would write `/r/videos`. The center screenshot in Figure 1 shows an example of a subreddit's homepage (in this screenshot, we display `/r/statistics`).

Figure 1: *(From left to right, top to bottom) The Reddit homepage, the statistics subreddit, and the comment section of a post.*


reddit
hot new rising controversial top gilded wiki promoted
STATISTICS
hot new rising controversial top gilded promoted show images (1)

1 Interested in gaining a new perspective on things? Check
promoted by redditads
promoted

101 1062 This camera gimbal (i.redd.it)
submitted 3 hours ago by lukeaw2525 to r/oddlysatisfying
54 comments share [l+c]

102 1022 Exactly what it looks like (i.redd.it)
submitted 3 hours ago by emlk0614 to r/DeFranco
36 comments share [l+c]

103 4126 TSA proof underwear (i.redd.it)
submitted 4 hours ago by Stephen1026 to r/funny
260 comments share [l+c]

104 2425 Salty Trumpet Shit ★ Post (i.redd.it)
submitted 8 hours ago by PyroCodeMage to r/bertstrips
22 comments share [l+c]

Homework questions are for r/homeworkhelp; How to ask a statistics question; Modmail us if your submission doesn't appear right away, it's probably in the spam filter.

49 ASA and p-values megathread (self.statistics)
submitted 1 year ago * (last edited 1 year ago) by dearsomething [M] - announcement
21 comments source share

262 Statisticians Found One Thing They Can Agree On: It's Time To Stop Misusing P-Value
(fivethirtyeight.com)
submitted 1 year ago by The_Elder_Steak - announcement
87 comments share [l+c]

1 3 Simple Main effect Syntax SPSS (self.statistics)
submitted 5 hours ago by anandoknows
2 comments source share

2 79 Some targeted marketing directed at baseball fans in New York and DC. Maybe this is why SAS so expensive? (imgur.com)
submitted 1 day ago by new_account_5009
16 comments share [l+c]

3 1 Chi-Squared: Using Yates' Correction for Continuity giving a very low p-value for data which sh produce this? (self.statistics)
submitted 9 hours ago by cant-press
3 comments source share

all 3 comments navigate by
sorted by: best

[–] paulginz 4 points 8 hours ago
Hi

The problem here is that you are averaging over groups before using the formula.

When using a continuity correction, the data should be kept in count/integer format. You should look at the totals in each column rather than the average.

permalink source embed save-RES hide child comments

[–] cant-press [S] 1 point 7 hours ago
Ah this makes sense thank you!

Just did it (with the Yates' correction) and it all works out :)

Thanks for your help!

permalink source embed save-RES parent

[–] IAmPixel 2 points 8 hours ago
You have a two by two table so Yates correction is required.

permalink source embed save-RES

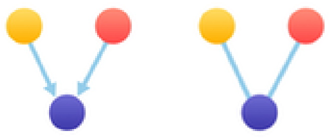
2.2 Reddit as a network

In mathematical terms, a graph G is an ordered pair consisting of a set V of vertices (also known as nodes) and a set E of edges: $G = (V, E)$. The set of edges E consists of pairs $\{u, v\}$ of vertices from V, indicating that vertices u and v are associated by an edge. In this paper, we use the terms *network* and *graph* interchangeably.

There are two main types of graphs: directed and undirected. In a *directed graph*, the edges are ordered pairs — that is, the order matters. In an *undirected graph*, the edges have no specified order. As a simple example, consider a graph of family members. We can formulate a directed graph or an undirected graph in this situation as follows:

1. **Directed graph.** Let each vertex represent a person. We will define our edges such that some person A shares a directed edge with another person B if A is a parent of B. Thus, parents will have edges pointing to their children, but their children will not have edges pointing back to the parents (see Figure 2, left side).
2. **Undirected graph.** Again, let each vertex represent a person. Now, define the edges such that some person A shares an edge with another person B if A and B are in the same family. Now, a parent and their child will have a single edge between them (see Figure 2, right side).

Figure 2: *A directed graph and an undirected graph, respectively.*



There are many possible approaches for structuring a Reddit dataset in the form of a network. For this paper, we are interested in looking at how redditors interact with each other within a subreddit.

To do so, we set up our network in a way that connects users when they have commented on the same post. Our network is a simple undirected graph. More formally, our network's vertices are users who comment on posts in a subreddit. Two user vertices will share an edge if those two users have commented on the same post.

As mentioned above, there are other approaches we could have taken. For instance, all comments on a post are not simply a flat list of comments — there is a nesting structure: a comment can be a reply to another comment. Thus, we could have made our network somehow involve edges between users who had commented in reply to one another. The reason we did not use this approach has to do with efficiency. This would have added an extra layer of complexity to our data scraping, which would leave room for more failures of the API (see discussion of limitations below) and would also cause our scrapes to take much more time to run. The goal of our choice of structure was to allow us to gather data efficiently and relatively easily, while also still capturing some information about how redditors interact within the subreddit.

2.3 Gathering the data

We gathered the Reddit network data using our own script (written in Kotlin) and the Reddit API. For a given subreddit, the script goes through the subreddit's "front page" (the top posts at the time) and looks at each post. For each post, it goes through each comment and records all the users who commented on that post (along with some other variables about the users and the comments). With all this information, we can create an edge list for users who comment on the same post as other users.

We set up the script to run once each day for eight days for a few subreddits of interest. The subreddits we ended up analyzing are the unofficial Bernie Sanders subreddit (`/r/SandersForPresident`) with approximately 215,000 subscribers and the unofficial Donald Trump subreddit (`/r/the_donald`) with approximately 391,000 subscribers. We compiled each of our daily subreddit scrapes into a single edgelist for each subreddit, spanning multiple days.

We can actually consider this edgelist to be a projection of a *bipartite graph*. A bipartite graph is a graph whose vertices can be partitioned into two disjoint sets, in which no vertices from the same vertex subset are connected by an edge. That is, a bipartite graph $G = (V, E)$ can have its vertex set V split into two disjoint subsets $U \subseteq V$ and $W \subseteq V$, where, for all edges $\{u, w\} \in E$, u and w are in different vertex subsets.

We can think of the vertices in a bipartite as each having a type, and thus we can divide the vertex set by type, giving us two disjoint sets. In our case, our two types are users and Reddit posts. Thus, if we say that our vertex set is made up of sets U and W (such that $U \cup W = V$), we can think of, for example, the U set as the set of vertices that represent users and W as the set of vertices that represent Reddit posts. For a user node u and a post node w, if $\{u, w\} \in E$, then we know that the user represented by node u has commented on the post represented by node w.

To get our desired simple graph structure in which each node represents a user (and no nodes represent Reddit posts), we first remove duplicate edges between users and posts (in case a user comments on a post multiple times). We then *project* the resulting bipartite graph into a simpler user-only structure. Projecting a bipartite graph removes the vertices of a certain type (in this case, the Reddit post vertices). When removing a Reddit post vertex, edges are formed between all the user vertices connected to that vertex. Thus, the process translates edges that represent comments on a post into edges that represent a shared post that both users have commented on. Bipartite graph projections are easily performed using the `igraph` package [3]. This process of projecting the graph forces each post to be a *clique* in our graph. A clique is a subgraph in which all vertices share edges with each other.

Due to limitations of the Reddit API, the scraper did not always function correctly during all eight of the days we scraped. Our dataset for the Sanders subreddit thus contains data scraped on April 3rd, 4th, 5th, 7th, and 10th (2017), and our dataset for the Trump subreddit contains data scraped on April 3rd, 4th, 6th, 10th, and 11th (2017). After our scraping period, we had gathered 143 posts, with a total of 7,908 comments from 3,767 unique redditors from the Trump subreddit and 86 posts with a total of 1,209 comments from 570 unique redditors from the Sanders subreddit.

Table 1: *Number of comments and commenters in our subreddit datasets*

Subreddit	Posts in dataset	Comments in dataset	Commenters in dataset
`/r/the_donald`	143	7,908	3,767
`/r/SandersForPresident`	86	1,209	570

2.4 Network attributes

In gathering and analyzing our network data, we included a few vertex attributes (i.e. variables) for each user. These were gathered or constructed from the Reddit API. Redditors are fairly anonymous on the site – users do not need to provide any personal information such as name, date of birth, or location. It is possible for a redditor's only public information to be their username, their posts, and their comments. Thus, we had a fairly limited selection of attributes to choose from. We gathered all the information we could easily obtain, including information specific to the user's comments on the subreddits of interest.

Each node (representing a user) in our dataset is associated with the following variables:

- The user's average comment "karma" (a score based on the number of upvotes and downvotes a comment has received) over all comments in our scraped data
- The user's average number of comments per post in our scraped data
- The user's average response time (minutes between post submission and comment submission)
- The user's average comment sentiment score (defined below in 2.4.1)
- The user's average "real word" count proportion, using text analysis (defined below in 2.4.1).

2.4.1 Sentiment and text analysis

As a further point of analysis, we generated sentiment scores for each comment in order to assess the relative positivity or negativity of the commenter's words. Sentiment analysis is a technique for quantifying how positive or negative a string of words is, by assigning each word a sentiment score and computing an overall score for the entire collection of words. We applied this technique to the comments in our dataset.

Each comment's sentiment is calculated using the Afinn word list in the `tidytext` R package [8]. Afinn contains a list of words, where each word is associated with a score on a continuous -5 to 5 scale (-5 being most negative and 5 being most positive) [7]. We found a user's average sentiment score by parsing the words in each of their comments and calculating the average Afinn numeric score over all the user's comments. This score is then assigned as a user vertex attribute.

In addition to sentiment, we included an average "real word" proportion attribute for each user. A real word count is obtained by subtracting the number of stop words (commonly used words like "the", "or", and "and") from the total number of words in a comment. We obtain our average real word proportion attribute by calculating the real word count for each comment and dividing it by the number of total words in a comment. Then, for each user, we take the average of the real word proportions of all that user's comments and set it as a user-level attribute.

2.5 Limitations of the data

A main limitation of our data is the Reddit API. Our scraper script ran once daily, but it occasionally ran into some hiccups. If the Reddit server was down for even a small period of time (which happens often enough for it to be relevant to our data collection), our scrape could fail to get any data. We rely on our data being spread out over eight days to help smooth the occasional gaps.

Due to our daily scraping, there is room for further caveats with the data. If a post were on the front page for a few hours of the day, but it was not on the front page when our scraper ran, we would miss those comments. Thus, there could be more edges that are not being taken into account, but for the most part our approach gives us a good idea of who is interacting with whom. Additionally, if a post was on the front page for multiple days, it would have more comments scraped from it, whereas a new post that was scraped in one of our later scrapes would have fewer comments (even if it gained more traction after our scraping period was complete).

Our sentiment analysis allows us to get a general sense of the type of language users are using, but it also has some limitations. For one, any misspelled words would not be included in our sentiment analysis because they could not be referenced in the Afinn word list. Furthermore, the Afinn list is extensive but not exhaustive so some words that could have informed us of the sentiment of a user may not have been included. It is also possible that a commenter used a word that is traditionally associated with a particular sentiment, but within the context of their post it had a different connotation. For example, if a user were to describe something as "not bad" this would recieve a score of -3 when in fact the user means something more similar to "good" which has a score of 3.

3. Exploratory Data Analysis

With our dataset gathered, we are ready to explore structural and numerical differences found between the subreddits. We will do this by examining visualizations as well as numerical statistics associated with the subreddit networks. Later, we will formalize our analysis using statistical methods.

In the tumultuous political climate of today, Bernie Sanders and Donald Trump's followers strike us as two social networks that could behave differently. Our networks can show how comment sentiment, percentage of real words, and popularity can promote different interaction between supporters. Our goal is to use vertex attributes to explain patterns in network interactions. The structural analysis of network graphs has traditionally been treated primarily as a descriptive task, as opposed to an inferential task [5]. In this section, it is our focus to identify key differences through descriptors in the Sanders and the Trump network graphs. It is helpful to use visuals to help us understand the environment that redditors interacting in the subreddits are creating, and visualizations will help us see distinct visual differences that will inspire us to explore certain characteristics of the network data further.

With so many vertices in our dataset, Section 3.1 relies on subsets of the data so that the visuals are easier to understand. The subset was obtained by taking a timed sample that would roughly reflect the amount of nodes in a day of the Sanders network. The subset should loosely reflect network characteristics of the entire data set, because it is not a random sample, and it would be easier to identify trends if the dataset was smaller. All statistical summary tables in this section reflect the sample of the dataset.

3.1 Comment attributes and sentiment analysis

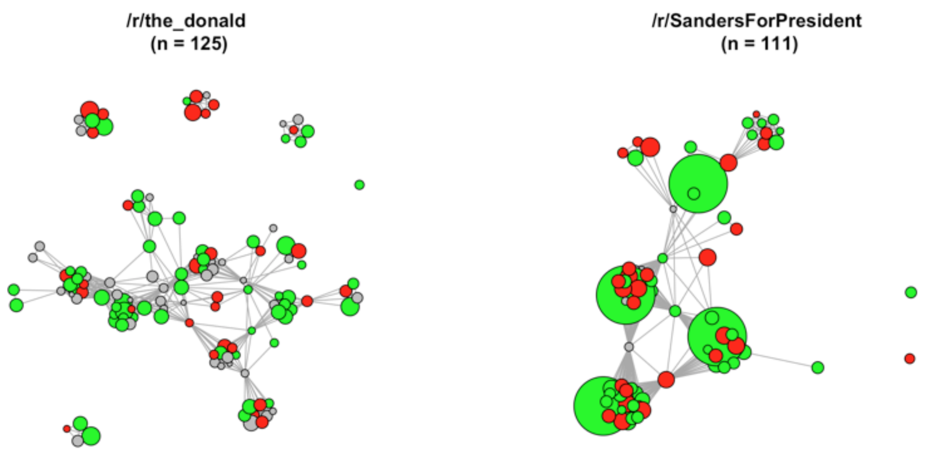
3.1.1 Assessing comment length and sentiment

As a first step in our analysis, we compare the average length of comments in both subreddits, incorporating sentiment. In Figure 3, the size of each vertex is proportional to the length of that user's average comment length, and the color indicates sentiment (red for negative, gray for neutral, and green for positive). We notice from Figure 3 how longer comments constitute more positivity in Bernie's subreddit. We also see how relatively shorter the Trump subreddit user comments are. In our sample, the Trump subreddit median comment length is 13.67 words, while the Sanders subreddit median comment length is 23.58 words. For the entire dataset, the median comment length is 17 and 30 for the Trump and Sanders combined data, respectively, showing this trend persists on a larger scale.

Table 2: *Average length of words, by sentiment.*

Subreddit	Positive	Neutral	Negative
`/r/SandersForPresident`	540.02	15.29	49.05
`/r/the_donald`	35.36	11.66	48.55

Figure 3: *The sampled networks, with each node sized by average comment length and color indicating sentiment (green = positive, red = negative).*



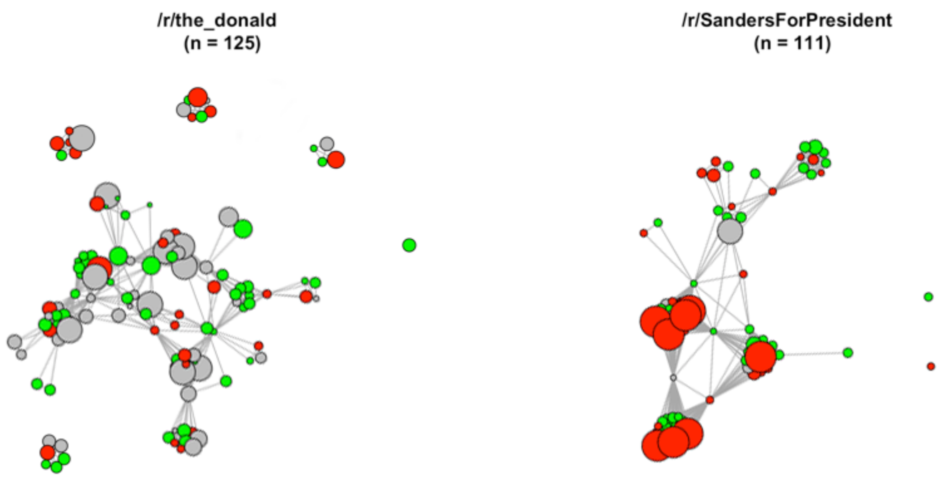
Also, while neutral-sentiment comments make up 39% of the total comments for Trump's subreddit and 22% of the total comments for Sanders's subreddit, neutral comments are much more rarely seen in the visualization in Figure 3, suggesting that neutral comments tend to be shorter in length and thus hidden by longer comments in the visualization. We can also see from Table 2 how much shorter neutral comments are compared to other sentiments, and how long the Sanders positive comments are on average in this sample. Finally, we notice by network structure that nodes are much more clustered in the Sanders network than the Trump network. We will formalize this observation in section 3.2.1 and 3.2.2, but for now it is important to note that the Sanders network looks more clustered than the Trump network. The structure of the network can tell us a lot about the type of interactions that occur in each network.

3.1.2 Assessing comment knowledgeability and sentiment

We now look at our real word count attribute with the goal of assessing how knowledgeable a comment may seem. Note that we will describe comments with more real words as more "intelligent," even though this is not necessarily the case.

Figure 4 displays the average ratio of real words used to total words used as node size. We see that neutral nodes dominate the Trump subreddit, suggesting that the most intelligent comments are neutral in sentiment. In Figure 3, we saw how short neutral comments are. This, combined with the prevalence of neutral comments in `/r/the_donald` suggests that the most intelligent comments on Trump's subreddit must be also short. Table 3 shows the average real word ratio for positive, neutral, and negative commenters.

Figure 4: *The sampled networks, with node size proportional to ratio of real words to words, and color indicating sentiment (green = positive, red = negative).*



On the other hand, the Sanders subreddit shows that knowledgeable users are more likely to interact and also to use negative language in their interactions, as Table 3 shows. The negative comments are clustered together, suggesting, perhaps unsurprisingly, that people with similar attitudes are more likely to interact with each other.

Table 3: *Average ratio of non-stop words to words, by sentiment.*

Subreddit	Positive	Neutral	Negative
`/r/SandersForPresident`	.371	.403	.510
`/r/the_donald`	.409	.581	.441

3.1.3 Assessing comment score (karma) and sentiment

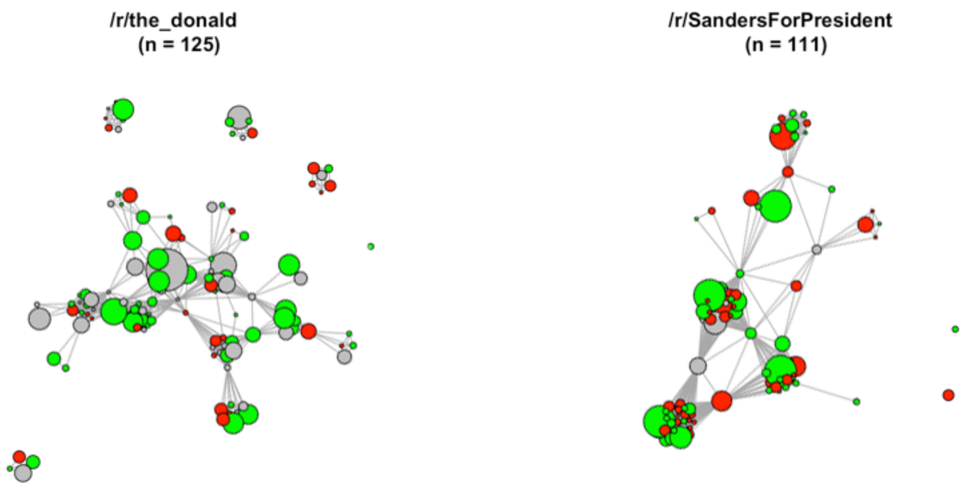
We now look at the "karma" of each comment (a score based on the number of upvotes or downvotes the comment has). The size of the nodes in Figure 5 represents the quantity of karma the user averaged. Table 4 shows that positive comments yield higher karma (on average 21.71 points) in Trump's subreddit, while negative comments yield higher karma (on average 32.169 points) in Sanders's subreddit.

From the network graph, we see that higher degree can yield higher comment karma. Using a plot of the log of degree and log of comment karma (Figure 6), we see a positive correlation between degree

Table 4: *Mean karma by sentiment*

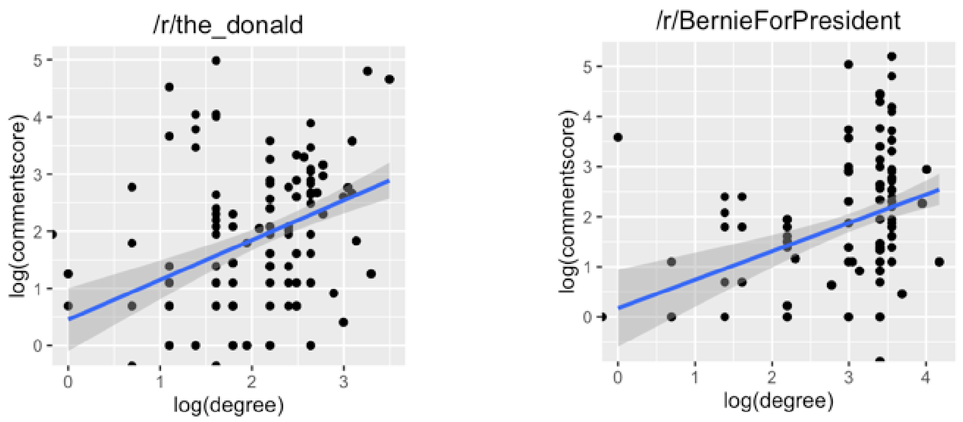
Subreddit	Positive	Neutral	Negative
`/r/SandersForPresident`	14.02	9.52	32.17
`/r/the_donald`	21.71	13	9.56

Figure 5: *Comparing average karma per user, and relating that to sentiment (green = positive, red = negative)*



and karma. This is not unexpected, since having a higher degree means that the user has most likely commented on posts with more commenters, and thus the user would have more opportunity to earn karma (from fellow commenters' upvotes).

Figure 6: *User degree and karma comparison*



3.2 Numerical and structural data analysis

Now that we have a good idea of how nodes generally behave with one another, we can look at more concrete network statistics to characterize the graphs.

3.2.1 Transitivity

Various types of basic social dynamics can be represented by triplets of vertices. We are interested in the proportion of triads that are "transitive" (i.e. where if x has an edge with y, and y has an edge with z, then x also has an edge with z). With undirected data, there are four possible types of triadic relations: no edges, one edge, two edges, or all three edges. The transitivity measure of a graph gives an idea of how clustered the graph is.

We can think of a triad as a small circle of friends. The interactions between these three people are significant to their dynamic as a group. The idea of transitivity connects triads to the broader context of a graph. As social network scholar Nicholas A. Christakis puts it, "Too much transitivity would mean the group was cut off from the rest of the world, and too little transitivity would mean the group was too disorganized to reinforce its own members' behavior. People might not know exactly how all their friends are connected, but they probably do have a sense of whether they might be able to reach people beyond their group" [2].

The transitivity of a graph G can be expressed as

$$T(G) = \frac{3\tau_{\Delta}(G)}{\tau_3(G)}$$

where $\tau_{\Delta}(G)$ is the total number of closed triplets in the graph G, and $\tau_3(G)$ represents the number of connected three nodes.

Table 5 shows that the transitivity of the Trump network is .497 and the transitivity of the Sanders network is .897. As we've already noted in Section 3.1, the Sanders network visualizations suggested a higher level of clustering than Trump and these network statistics support this claim.

Table 5: *Global network statistics*

Subreddit	Vertices	Edges	Density	Transitivity	Average Path Length	Modularity
`/r/SandersForPresident`	570	16,424	.101	.897	2.298	.707
`/r/the_donald`	3,767	219,331	.031	.497	2.226	.505

3.2.2 Density

We now assess the density of our network data in order to examine the dyadic connections in a population. Density is simply the ratio of the number of realized edges to the number of potential edges. That is, in a graph G with no self-loops and no multiple edges, the density of $G = (V, E)$ is

$$den(G) = \frac{|E|}{|V| \times \frac{1}{2}(|V| - 1)}$$

The value of $den(G)$ will lie between zero and one and provides a measure of how close G is to being a clique (a graph in which all vertices are connected to all others). The closer density is to one - the closer the network is to being a clique.

The density of the Trump subreddit is 0.031, whereas the density of the Sanders subreddit is 0.101. The Sanders community is shown to be more clustered and more than three times as dense the Trump community.

3.2.3 Average Path Length

We can also look at movement through average path length and assess how efficiently information can spread in each network. Path length gives insight to how far each nodes is separated from other nodes. Path length is simply the shortest number of edges it takes to travel from one node in the network to another; average path length is calculated by averaging this value over each possible path in the network. This is one statistic that is very similar between the Sanders network and the Trump network. The Sanders subreddit users average 2.298 steps between one another, while the Trump subreddit users average 2.226 steps between one another.

This statistic shows that the different density and transitivity between the subreddits do not have a particularly large effect on average path length. Diameter is another statistic similar for both networks, diameter is the maximum path length between any two nodes on the graph. The diameter of the Trump network is 8 and the Diameter of the Sanders network is 10. This makes sense because the Sanders network is more clustered than the Trump network, so it will be slightly harder to connect two "far apart" nodes in the Sanders network compared to the Trump network.

We see both similarities (diameter and average path length) and differences (transitivity and density) between the two network graphs, and a look at mathematical models in Section 4. will help us understand if those graph characteristics are simply appearing by chance. For now, we will look at partitioning and modularity to study how modular the subgraphs in our model are.

3.2.4 Partitioning & Modularity

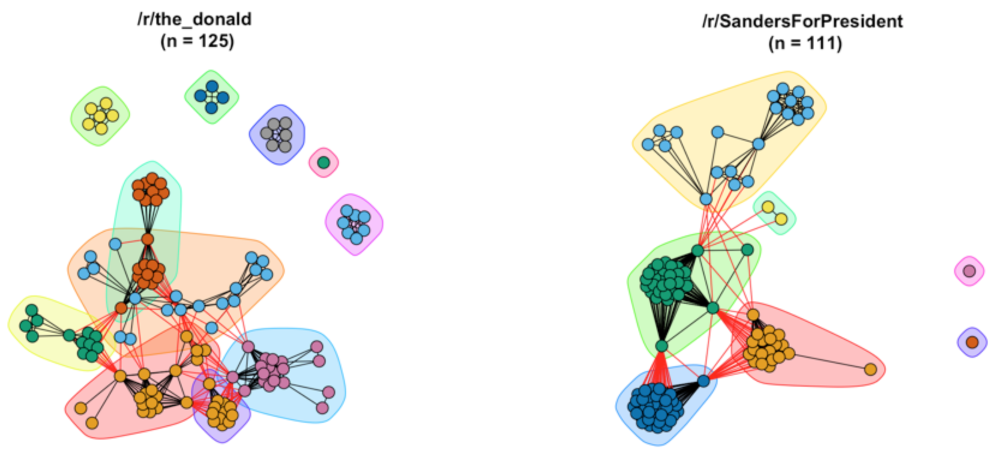
Partitioning refers to the segmentation of a set of elements into "natural" subsets. More formally, a partition $C = (C_1, ..., C_K)$ of a finite set V is a decomposition of V into k disjoint, nonempty subsets C_K such that $\bigcup C_K = V$ [5]. In the analysis of network graphs, partitioning is a useful tool for finding subsets of vertices that demonstrate a "cohesiveness" with respect to underlying relational patterns.

From Figures 3, 4, and 5 we see relatively more dense subgraphs that form when multiple users interact on a post. To formalize this notion, we can apply a hierarchical clustering algorithm to partition the graph into clusters. For this example, we use the fast greedy algorithm. This algorithm is also called the Clauset-Newman-Moore algorithm [1], and at each step two groups or nodes merge based on modularity. Modularity measures how mutually exclusive or "modular" the subcommunities as a whole are. If modularity is closer to one, then the subcommunities are more modular, while if the modularity is closer to zero, there is more overlap between communities. Fast greedy aims to optimize modularity with as many steps until the modularity does not get any higher. Initially, every vertex belongs to a separate community, and communities are merged iteratively such that each merge is locally optimal (i.e. yields the largest increase in the current value of modularity). The algorithm stops when it is not possible to increase the modularity any more, and it produces the partitioned networks in Figure 7. Modularity can tell us a lot about how separated the subcommunities in our networks are.

The sample Trump network was found to have 11 clusters and a corresponding modularity of 0.705, while the Sanders network was found to have 7 clusters with a modularity of 0.536. These statistics are visually represented in Figure 7. We can observe that the Trump network has seemingly less red edges in the graph (indicating overlap between communities) and further showing that the communities are more "pure" in this sample. However, on a much larger scale, the higher transitivity in the Sanders subreddit starts to come into play and the network takes on a higher modularity than the Trump network.

When exploring the entire data set, the Sanders subreddit exhibited 25 communities with a modularity of .707, while the Trump subreddit had 13 communities and a modularity of .505. Furthermore, we see only a few nodes that were seriously distressing the modularity of the sample Sanders subreddit in Figure 7, so with a larger sample size, the algorithm was able to find a higher modularity amongst the Sanders network. (Note that Figure 7 does not reflect these results, as it is only displaying a sample of the dataset).

Figure 7: *Sample hierarchical clustering partition visual*



4. Mathematical Models of Networks

When performing a statistical analysis of networks, we rely on statistical models. However, there are also purely mathematical models of networks – networks that can be generated according to a set of rules and parameters such that the networks have certain characteristics. Before introducing our statistical model, we will introduce a few mathematical models of networks, and, when applicable, we will compare these models to our data to see if our data displays any characteristics of a mathematical model.

4.1 Random network

The first and most basic type of mathematically modeled network is the random network. There are two ways to construct a random network:

1. You can specify a number of vertices and a number of edges for the network to have, and then generate such a network, where the edges are randomly distributed between pairs of vertices.
2. You can specify a number of vertices and an edge probability p, and then generate a network where each pair of vertices has probability p of sharing an edge.

We can easily compare our network data to randomly generated graphs using the first technique: since we know the number of vertices and edges in our dataset, we can generate a large number of random graphs with the same number of vertices and edges, and then compare our network's statistics to the random network distribution. We will compare the networks' transitivities, average path lengths, and diameters. This comparison takes the form of a randomization test: we will simulate distributions of these statistics over 500 random graphs, and we will then compare our observed network data to these distributions.

For the /r/SandersForPresident network data, we find that the transitivity (0.898) is far greater than the maximum transitivity of the random network distribution (0.103). The average path length for the Sanders data (2.298) is also much larger than the maximum of the random network's average path length distribution (1.901). Finally, the Sanders graph has a diameter of 10, which is higher than the random network diameters (all of which were 3). All of these extreme statistics indicate that the Sanders network is, unsurprisingly, not random — there are other factors at work in the structure of the graph. We see a very similar lack of randomness in the Trump network data. The tables with the distributions of our simulated random graph statistics along with the observed values of those statistics are given in tables 6 and 7 (with the observed statistics in the final column). Note that although it may seem striking that the diameter and average path length of the simulated networks for both /r/SandersForPresident and /r/the_donald are fairly similar, this result is unsurprising given that diameter scales logarithmically (as a function of the size of the vertex set V) and that the networks have many vertices and edges [5].

Table 6: */r/SandersForPresident - Statistics of interest for simulated graphs, as well the observed values (in the last column)*

Statistic	Min.	1st Qu.	Median	Mean	3rd Qu.	Max	**Observed**
Transitivity	0.100	0.101	0.101	0.101	0.102	0.103	**0.898**
Avg. Path Length	1.901	1.901	1.901	1.901	1.901	1.901	**2.298**
Diameter	3	3	3	3	3	3	**10**

Table 7: */r/the_donald - Statistics of interest for simulated graphs, as well the observed values (in the last column)*

Statistic	Min.	1st Qu.	Median	Mean	3rd Qu.	Max	**Observed**
Transitivity	0.031	0.031	0.031	0.031	0.031	0.031	**0.498**
Avg. Path Length	1.995	1.995	1.996	1.996	1.996	1.996	**2.226**
Diameter	3	3	3	3	3	3	**8**

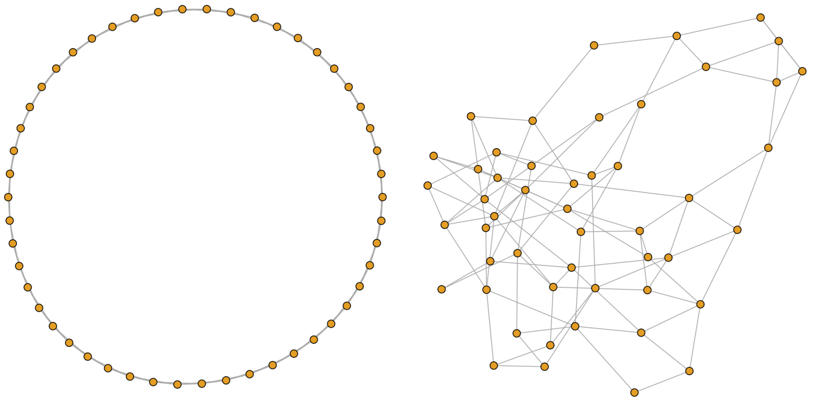
4.2 Small-world networks

Many networks in the real world exhibit high levels of clustering while also exhibiting small distances between most nodes. As an example, consider a network of friends. It is not hard to imagine distinct clusters for closer groups of friends joined to other clusters by a few nodes who have friends in other groups. Thus, this network would have a high level of clustering, and all nodes (even across friend groups) would be fairly accessible to other nodes, indicating small distances between most nodes.

These characteristics do not show up in the randomly generated graphs discussed in the previous section. So, to generate random graphs with these characteristics, Watts and Strogatz proposed the following process: create a network with a lattice structure, and then "rewire" a given (small) percentage of edges in the graph [9]. When generating these networks in R, there are four main parameters to set: the dimension of the lattice, the number of vertices to use, the "rewiring" probability, and the number of neighbors each node should initially be connected to in the lattice. As an example, in Figure 8, we see two different graphs generated by the Watts-Strogatz algorithm: the graph on the left has a rewiring probability of zero (it is simply a lattice), and the graph on the right is of the same size, but with a rewiring probability of 0.2.

To check whether our network data has small-world characteristics, we can refer to our random network distributions in tables 6 and 7 from Section 4.1. By comparing the transitivity (which is a measure of clustering) and average path length of our observed networks to the distributions from simulated networks, we can see whether a subreddit network has a high level of clustering while

Figure 8: *(Left to right) A small-world network with 50 vertices and a rewiring probability of zero, and a small-world network with 50 vertices and a rewiring probability of 0.2.*



maintaining a regular average path length.

Referring back to Table 6, we see that the transitivity of our Sanders network data (0.898) is much higher than the maximum transitivity from a distribution of random graphs of the same size (0.103), by a factor of 8.7. Thus, there is significantly more clustering in our dataset than in a randomly generated graph. Also, the average path length for the Sanders graph (2.299) is only 1.2 times longer than the maximum of the random graph distribution (1.901). Therefore, our observed network data has higher clustering but a relatively similar path length when compared to the random networks.

As for the Trump network data, in Table 7, we see similar (and stronger) results. The transitivity of the Trump data (0.498) is nearly 16 times higher than the maximum of the distribution of random graphs (0.031), indicating very high clustering relative to a random graph. The average path length for the Trump data (2.226) is only 1.1 times greater than the maximum of the average path lengths in our distribution of random graphs (1.995). Therefore, we can draw a conclusion similar to the one we drew from the Sanders data: the similar average path length combined with the increased clustering compared to the random networks suggests the presence of small-world characteristics. This conclusion is stronger in the Trump case due to how much higher the transitivity is, relative to the random networks.

5. Statistical Models

Typical statistical models rely on the assumption that observations are independent of one another. This assumption of independence is violated when looking at networks. As an example, consider a social network: two people who are friends, are more likely to have similar characteristics than two people who are not friends. Thus, there is an inherent dependence between vertices in the network. Therefore, to account for this dependence, we need a model that takes into account the structure of a network. An exponential random graph model (ERGM) is analogous to a generalized linear model (GLM) but takes into account the underlying structure of the network. ERGMs are a generative model, which express the probability of an edge between two nodes as a function of node attributes and network structure properties.

Using the `ergm` package in `R`[4], models are fit using Monte Carlo Markov Chain maximum-likelihood estimation methods (MCMC MLE). We can interpret the coefficients of the model as the log-odds of an edge occurring between two nodes based on the node characteristics of each, conditional on the rest of the graph. Since ERGMs consider the probability of an edge between two nodes rather than a single observation, they allow for the inclusion of four predictor types:

- Node attributes (main order effects), which is defined using an additive form. For our model we will use the main effect between two nodes x_i and x_j as $\alpha(x_i + x_j)$
- Dyadic predictors (second order effects), which is defined through a relation between two nodes. For our model we will use the absolute difference between two node variables, x_i and x_j would be $\beta|x_i - x_j|$
- Structural predictors

Node attributes are predictors wherein the characteristic of an individual node affects the likelihood of observing an edge incident to it. Dyadic predictors measure the similarity or dissimilarity of node attributes, which affects the likelihood of an edge between them. *Homophily* is the tendency of vertices to associate with vertices with similar characteristics, and *heterophily* is the tendency of vertices to associate with vertices with more dissimilar characteristics. Structural predictors can be included which include structural factors in the model (such as degree distribution or transitivity).

In building our models of our `/r/the_donald` and `/r/SandersForPresident` subreddit networks, we will focus on node and dyad predictors (main and second order effects) as well as testing the effects of structural predictors to see how these improve our model's ability to accurately reflect our observed network. For the model used on our subreddit data, a specified formula reduced from general formula for ERGM is

$$\mathbb{P}(Y = y) = \frac{1}{k}\exp\{\sum_{H}(\theta|E| + \sum_{i,j}(\alpha(x_i + x_j) + \beta|x_i - x_j|))g_H(y)\}, 0 \leq i \leq j \leq |V|$$

where,

1. Y represents the adjacency matrix of the network and y is a special instance of the adjacency matrix that we want to calculate the probability about.
2. Each H represents a subset of vertices in G with a configuration of edges. And $g_H(y) = \prod_{y_{ij} \in H} y_{ij}$, where $g_H(y) = 0$ indicates that H is not a possible configuration with the y adjacency matrix, otherwise $g_H(y) = 1$. Therefore, the formula would only need to consider the cases where H is consistent with y adjacency matrix.
3. $\sum_{i,j}(\alpha(x_i+x_j)+\beta|x_i-x_j|)$ is the sum of evaluation on each pair of vertices within H configuration. For each pair x_i and x_j, $\alpha(x_i + x_j)$ is the evaluation on the pair's main order effects with α being the coefficient. $\beta|x_i - x_j|$ is the evaluation on second order effects with β being the coefficient.
4. $\theta|E|$ is an intercept term that controls the density of our modeled networks.
5. k is a constant for normalization, so that we can ensure that the probability is between 0 and 1.

This ERGM defines a joint distribution for all network edges. But for the model interpretations, we evaluate the probability that an edge appears in a network given the rest of the network structure. The following formula is similar to a generalized linear model interpretation:

$$D = \theta|E| + \sum_{i,j}(\alpha(x_i + x_j) + \beta|x_i - x_j|)$$

$$\mathbb{P}(Y_{ij} = 1|Y_{i'j'}) = \frac{e^D}{1 + e^D}$$

where $Y_{i'j'}$ represents the edges in the network that follows the corresponding position of $y_{i'j'}$ in the adjacency matrix instance y,

1. Given x_i, x_j and the rest of the network edges, D is the log odds of an edge between thesetwo vertices.
2. e^D is the odds of the edge appearing and therefore we use $\frac{e^D}{1+e^D}$ to calculate out the actual probability of an edge between nodes i and j.

5.1 Model Building

We began by constructing a null model that had no substantive predictors. We can use this model as a point of comparison for subsequent models. The only term in our null model is for the edges, which means the model would produce a random graph with the same number of nodes and edges as our observed graph. Subsequent graphs with additional terms are compared to this model to verify that the predictors added are useful and do not result in overfitting of the model. From our null model, we iteratively added significant attributes to the model, checking that our AIC decreased at each step, until we had exhausted our set of attributes and arrived at a model with significant predictors. We considered including all node attributes as main order effects, and left in only those that improved the model. For dyadic predictors we considered sentiment score since we were interested in observing the relationship between commenters based on the similarity of their sentiment. We looked at the goodness of fit of our models using node and dyadic predictors before determining whether any structural predictors were necessary.

5.2 Model fit analysis

We need to determine whether the model is actually a good fit for the observed networks. A good feature of ERGMs is that they are generative; that is given a set of coefficients, we can simulate networks based on their maximum likelihood realization. Note that the simulated networks will have the same node attributes as the actual observed network.

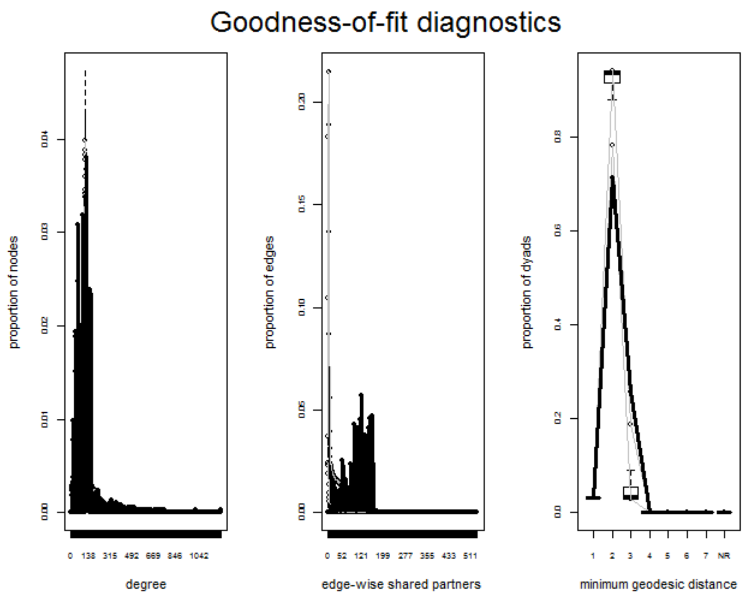
We use a goodness of fit (GOF) test on these simulated networks to check the GOF of an ERGM model built from our observed network. We are running, by default, 100 randomly simulated networks in the GOF test. GOF works by comparing the network characteristics of these simulated networks with the given network. The characteristics being compared are the distributions of the network statistics node degree, edge-wise shared partners, and geodesic distance for each node. We define these terms in the following list:

- **Node degree** is the number of edges incident to a node.
- **Edge-wise shared partners** is the number of dyads (a connected pair of nodes) who are themselves connected and who have a fixed number of shared partners.
- **Geodesic distance** between two vertices in a graph is the number of edges in a shortest path connecting them.

Plotting the statistics for simulated networks and the actual network can be very informative as shown in Figures 9 and 10. As we can see, the plot produces one panel for each of the three network statistics, including a box-plot and 95% empirical confidence intervals (indicated by light grey lines) that show the variability of the individual network statistic across the simulated networks. The black lines reflect the distribution of the three statistics for the observed network. Furthermore, we can compare the specific statistics for each simulated network with the statistics of our observed network, in order to check if our model is doing a good job of describing the observed network. The statistics for these 100 simulated networks are collected with minimum, mean, and maximum values for comparison with our network.

5.2.1 Trump network goodness of fit

Figure 9: *Goodness of fit for Trump network*



As we can see in Figure 9, our black line (representing the observed network) sits inside the confidence-boxes for degree, geodesic distance, and the edge-wise shared partners (although it may be hard to see, there is a black line in the peak of its distribution). In the simulated data result, out of hundreds of GOF statistics, at least 80% of the statistics have large p-values (at least 0.5), each indicating that the model can generate our original network characteristics on the modeled network. Therefore, there is no need for further improvement for this model (that is, we do not need to consider including a structural predictor).

5.2.2 Sanders network goodness of fit

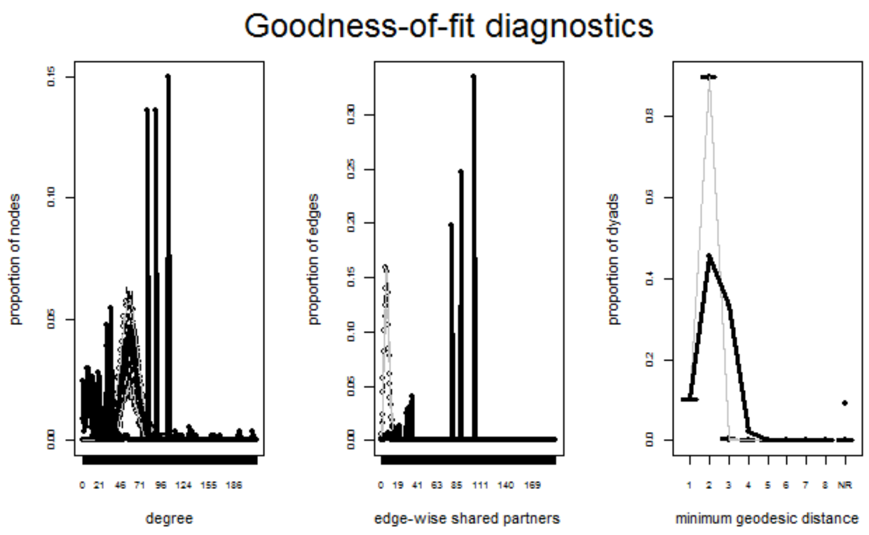
As we can see in Figure 10, our black line sits inside the confidence-boxes for degree and geodesic distance. But it is much more off with the edge-wise shared partners statistics. So our model is not doing a good job, as it tends to produce a network that underestimates the number of dyads that are themselves related and have a fixed number of shared partners. In the simulated data result, out of hundreds of GOF statistics, at least 50% of the statistics have very low p-values (below 0.05), each indicating that the fitted model cannot regenerate the corresponding characteristics in our original network. Therefore, we need to find an improved model.

5.2.3 Improving the Sanders model using structural predictors

To improve the model, we can add in another type of predictor — a structural predictor. Because this predictor provides information about local structures in the network to the model, such as shared edges between some local pairs of vertices, the new model would likely better fit the observed network. There are three most commonly used terms of structural predictors: geometrically weighted degree distribution (GWDegree), geometrically weighted edgewise shared partnerships (GWESP), and geometrically weighted dyadwise shared partnerships (GWDSP).

Since the goodness of fit test specifically indicates a bad fit on the statistics of edge-wise shared partners and good fit on statistics of degree and maximum geodesic distance, we decide to only add

Figure 10: *Goodness of fit for Sanders network*



in GWESP to improve this aspect of our model. This predictor measures transitivity of the network. In another sense, this approach models the degree distribution and how likely that the higher-degree vertices would occur in the model.

In addition to the drastically increased time complexity on running models with structural predictors, there are two potential problems: model degeneracy and convergence failure. Model degeneracy occurs when the model overestimates the likelihood of a small set of unlikely outcomes and does a poor job of fitting the observed network. This often results in networks generated using the model being much too dense or an empty graph, with no edges. According to the goodness of fit test so far, we have not encountered this issue. Typically, after we successfully add a structural predictor into the model, we would have to check back on this model degeneracy issue again.

However, convergence failure did pose a problem. Since ERGMs use MCMC MLE, the model needs to reach a converged maximum log-likelihood value to give a final estimates. The first 20 iterations of estimation (each iteration can take up to a few hours) with GWESP added did not show any convergence. Due to limited computational capacity, even if there is an actual convergence point after 20 iterations, we are not able to use more iterations to find it. Also, considering that the model examples in the books [5] [6] that successfully incorporated GWESP usually achieved convergence in less than 20 iterations, we decide to stop further effort on this new model and call it unsuccessful due to a convergence failure. Since we cannot build a new model with GWESP, model degeneracy is not our concern here.

It is likely that the GWESP term does not converge on our `/r/SandersForPresident` data due to the highly clustered nature of how our network was constructed. Since users are linked to all other users who commented on the same post, any two users connected by a post are connected to all of the same other users from that post. We see that the values for edge-wise shared partners are discrete, since each user's number of edge-wise shared partners is some linear combination based on how many posts they commented on and the number of comments on each of those posts. This unique clique structure of our data makes it much more difficult for the model to converge and the inability to accurately reflect the global property of edge-wise shared partners is a limitation of our model.

5.3 Model Interpretation

We can now interpret our final ERGM models. The interpretations for the Sanders subreddit model may be inaccurate due to the lack-of-fit, but we will proceed as attempts to include structural predictors to improve the model were unsuccessful.

5.3.1 Sanders Network Model

Our final model for the Sanders subreddit has three significant node predictors (Table 8): a user's average number of comments per post, the log of average comment score, and average response time. Our model also has one dyadic predictor, which predicts ties based on how similar two nodes are with respect to their average sentiment score. For the Sanders network, the AIC score decreased from the null model (105,483) to our final model (105,047).

In the Sanders network we see that, holding all else equal, a user is more likely to share an edge with users with more similar sentiment scores. That is, we see homophily. Increasing the distance between two users' sentiment scores by 1 results in a 4% decrease in the odds of an edge between them.

Also, a user with a lower average number of comments, a lower average comment karma score, or a slower response time would be more likely to share an edge with another user by commenting on a post. If the total average number of comments between two users increases by 1, then the odds of an edge between them decreases by 2.6%, holding all other terms constant. Increasing the combined comment karma score between two users by a factor of 10 decreases the odds of an edge between them by only 0.29%, holding all other terms constant. Finally, if the combined average response time between two users increases by 60 minutes, we see a 1.6% increase in the odds of an edge between them, holding all other terms constant.

Table 8: *`/r/SandersForPresident` - Model Terms*

Predictor Type	Predictor	Coefficient	Std Error	P-Value
Structural	Edges	-2.026	2.80e-02	<1e-04
Main Order	Avg Number of Comments	-2.60e-02	1.87e-03	<1e-04
Main Order	log(Avg Comment Score)	-1.26e-03	1.98e-04	<1e-04
Main Order	Avg Response Time	2.57e-04	2.12e-05	<1e-04
Second Order	Avg Sentiment	-4.21e-02	9.14e-03	<1e-04

5.3.2 Trump Network Model

The final model for the Trump subreddit network has two main order node predictors (Table 9): average number of comments and the log of average comment score. We also have one dyad predictor for the average sentiment score. For the Trump network we see that the AIC has decreased in comparison with our null model, from 1,949,242 to 1,948,873 suggesting an improved model fit and that that we are not overfitting our model.

We see a significant dyadic predictor for average sentiment score. We see homophily amongst the sentiment of the Trump subreddit users as well. This means that given the rest of the edges in our network, if the difference between two users' sentiment scores increases by 1, the odds of an edge between them decreases by 2%. If the sum of the average number of comments between two users decreases by 1, then the odds of a link between the two increases by 1.4%. Increasing the sum of the of the average comment score between two users by a factor of 2, results in the odds of a link between the two decreasing by nearly 1%, holding all other terms constant.

Table 9: */r/the_donald - Model Terms*

Predictor Type	Predictor	Coefficient	Std Error	P-Value
Structural	Edges	-3.43	8e-03	<1e-04
Main order	Avg Number of Comments	-1.4e-02	1e-03	<1e-04
Main order	log(Avg Comment Score)	1.4e-02	1e-03	<1e-04
Second order	Avg Sentiment	-2e-2	2e-03	<1e-04

6. Conclusion

In this paper, we have introduced some basic network concepts (specifically, those related to network statistics), discussed mathematical models of networks, and modeled our data using the exponential random graph model. We find that our data was not random, and, in fact, that our data exhibits the small-world characteristics of high clustering and normal-sized paths. Upon examining the data in our exploratory data analysis, we observe the potential for homophily with regard to sentiment. We also notice interesting trends with regard to comment length and comment karma score; namely, sentimentally neutral comments tend to be shorter in both subreddits, and we see that the Sanders subreddit favors positive comments, whereas the Trump subreddit favors negative comments. When we build a statistical model to test these hypotheses, our hypothesis about sentiment homophily is confirmed. We also see a number of other interesting results, regarding comment timing, comment karma, and number of comments a user posts.

Although the results may seem obvious or inconsequential, we note that the modeling techniques used in this paper have potential for further analysis of subreddit data. With more resources, one could obtain a dataset that retains the nested structure of comments in a Reddit discussion, which would allow the model to include more information about interactions within the comments section of a post. Furthermore, there are other variations on the ERGM which could allow for different sorts of analyses. In the end, the analysis in this paper functions as a starting point for network analysis of comment discussions on internet forums, and we are excited to see where further research may take this idea.

References

[1] Aaron Clauset, M. E. J. Newman, C. M. Finding community structure in very large networks.

[2] Christakis N, F. J. *The Surprising Power of Our Social Networks and How They Shape Our Lives*. 2009.

[3] Csardi, G., and Nepusz, T. The igraph software package for complex network research. *InterJournal Complex Systems* (2006), 1695.

[4] Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M., Morris, and Martina. ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software 24*, 3 (2008), 1–29.

[5] Kolaczyk, E. D., and Csárdi, G. *Statistical Analysis of Network Data with R*. Springer, 2014.

[6] Luke, D. A. *A User's Guide to Network Analysis in R*. Springer, 2015.

[7] Nielsen, F. Å. Afinn, March 2011.

[8] Silge, J., and Robinson, D. tidytext: Text mining and analysis using tidy data principles in r. *JOSS 1*, 3 (2016).

[9] Watts, D. J., and Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature 393*, 6684 (June 1998), 440–442.