Factors to Being Successful on Reddit

Statistics 133 Final Project Report

By Sam Zhou, Ken Chen, and Christian Choi (The SnooSquad)

Introduction

Reddit is a popular website that is at its essence an online forum. It is designed to facilitate sharing of user created content, news articles, personal opinions, and in general, anything fun people find on the internet. The heart of Reddit is its voting system and the community input. Anyone can post or comment using any content they would like to and anyone who browses Reddit can vote on posts and comments they personally find interesting - upvote if you enjoyed the content or agree with it, downvote if you find it unpleasant or negative. The net upvotes minus downvotes is a metric that Reddit refers to as Karma. It is a point system tied to a person's account that accrues as people vote on their new posts and comments on existing posts. Essentially, it is a measure of how often a person contributes to the community and whether or not their contributions are generally positive or negative. High Karma posts end up on the front page of Reddit and receive more exposure than posts with low Karma. Along the same line of thought, comments with high Karma appear at the top of the thread of replies to a post. Frequent users of Reddit look to create content that results in high Karma because it is an indication that the community approves of what they have produced. The ultimate question of this report is to ask what leads to successful or high Karma posts and comments as well as how to get the most exposure for your content.

In order to answer this question we will look at what to post, when to post, and where to post. The question of where has to do with Reddit's subreddit system. Subreddits are essentially categories that posts are filled under which allow users to browse content related to what they are interested. For example, the subreddit "science" is filled with links to scientific articles and discussions of new advancements in the scientific field. The analysis will also differentiate between posts and comments at various points because some metrics are unique to one or the other and the scale of Karma is different for both groups. We will begin with methods used to gather our data and then move into how we analyzed and visualized our data. Finally, we will conclude with the results and findings of our analysis and final remarks on what leads to success on Reddit.

Gathering Data

Data from Reddit was gathered in two ways: Big Query and web scraping. Reddit is a massive website with almost 200 million posts and about 8 billion comments. The full corpus of Reddit data consists of over 1TB of plain text which is far too much to be processed locally. Thankfully, this data has been published on Google's Cloud Service Platform, Big Query. It is a service that hosts enormous data sets that can be queried using SQL code. This returns a table with results that can be downloaded as a csv if the file is small enough. The restriction on the file, however, is incredibly small and would only contain about a week or two of only comments. The workaround for this was to use a package in R called *bigrquery*. It allows R to interface directly with Big Query and return the table as a dataframe in R. This also has its limitations but allows us to pull a few full months of both posts and comments directly into R. These posts and comments are from April-August 2014 and is the data set we primarily used in all of our visualizations. The code placed at the top of our scripts to pull this data from Big Query can be found in **APPENDIX A.1**.

The only visualizations that do not use this data set are the **comment networks** of top posts in some of the most popular subreddits. This data was instead scraped from Reddit using their API and a Python library called PRAW (Python Reddit API Wrapper). This was done because the desired comments were from very specific posts and the goal was to encompass the entirety of a post's comment tree. Because of the limitations of the Reddit API, comments are only associated with their parents, and not the post to which the highest-level comment belongs. Thus, the filtering of all comments from a particular post given a sample of the entire corpus is extremely difficult and would require deep recursion into the comment trees. Instead, we extracted the entire comment trees (as JSON/XML, which is handled by PRAW) and flattened the result to get an iterable of all relevant comments. The Python script found in **APPENDIX A.2** writes the result into a csv file, which is read into R and further cleaned. The wrangling of this data will be discussed in the next section.

Data Visualizations and Analysis

Establishing a Model for Success

In order to begin analyzing what leads to success in Reddit, we must first establish what is considered to be successful and who to look at as a model for success. The quickest solution is to look at the list of Top Users found on the website itself. The issue with this, however, is that the list is updated to reflect current standings in 2016 while the data set we are looking through is from 2014. The quickest remedy for this was to look at our data set and isolate the users with the most Karma accumulated over the timespan of our data set. Figures 1 and 2 show the top users by first looking at posts and then at comments.

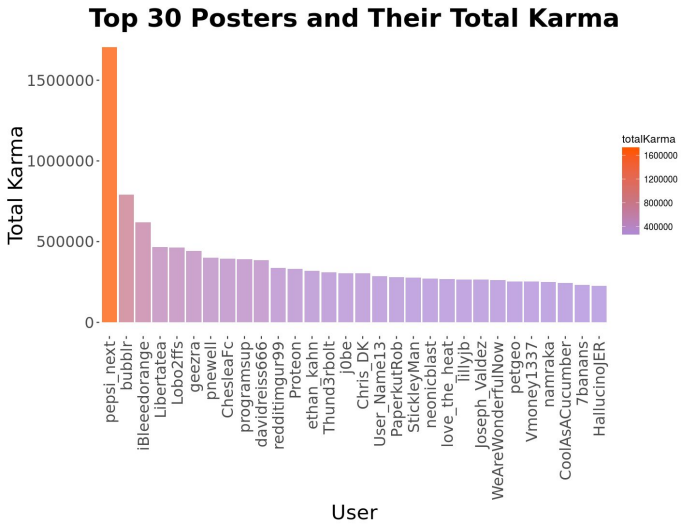

Figure 1**. Top users based on their post Karma. Generated by code found in **APPENDIX B.1

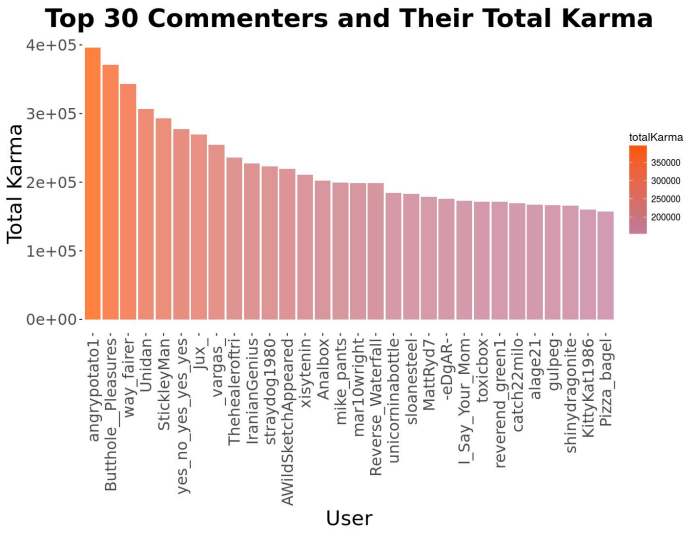

Figure 2**. Top users based on their comment Karma. Generated by code found in **APPENDIX B.2

These users have earned the most Karma within the scope of our data set and any future references to “top users” in this paper will be in reference to these lists of users.

Where to Post and Comment

In order to get as much exposure as possible for a submission it is important to display it somewhere that receives as much attention as possible. The more people that are active on a subreddit, the more likely it is to be seen at least a few times just by sheer chance alone. Although our data does not include the number of views a post or comment recieves, there are other metrics for judging the activity of a subreddit. In terms of activity that is relevant to posts, votes are likely to be the best metric. People browsing Reddit have the ability to skim pages and pages of posts while only looking at thumbnails and titles without clicking into the thread. They can also vote on these posts from the front page very quickly. This means that many of the detectable interactions on Reddit is only through voting. Figure 3 shows the subreddits with the most activity based on total votes where total votes are calculated by tallying all upvotes and downvotes.

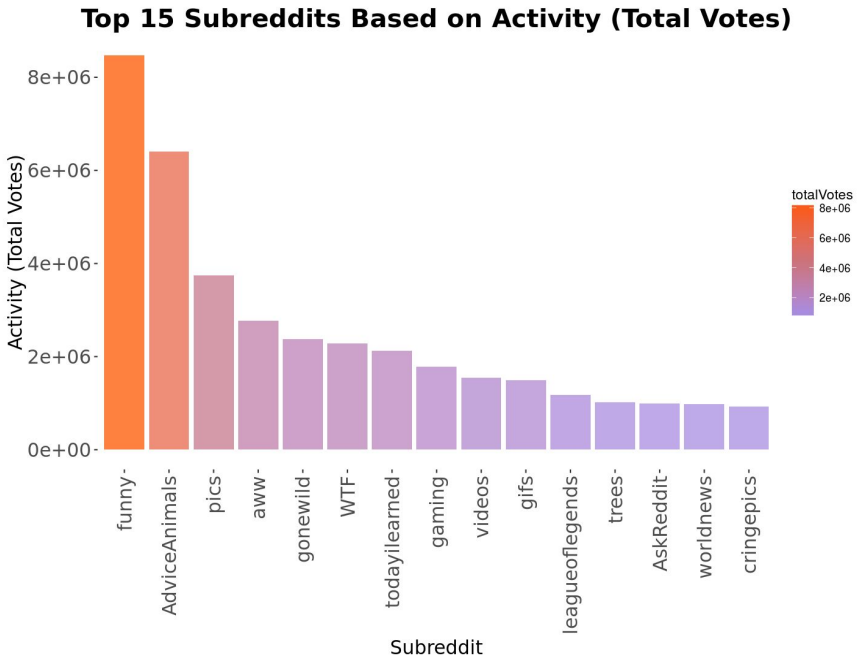

Figure 3**. Displays the top 15 subreddits based on the total count of votes on posts in the subreddit. Generated by code found in **APPENDIX C.1

This was not, however, the only metric used to try and determine which subreddits lead to the most successful posts. Another analysis was done on which subreddits led to the most viral posts. Viral posts are ones that earn a significant amount of Karma, gain more visibility from that Karma, and exponentially earn more

Karma. The threshold for what we term a viral post is above 500 Karma since any post above this point is on its way to the front page where almost everyone will see it. Figure 4 was created by filtering out only viral posts and counting how many of these posts existed in each subreddit.

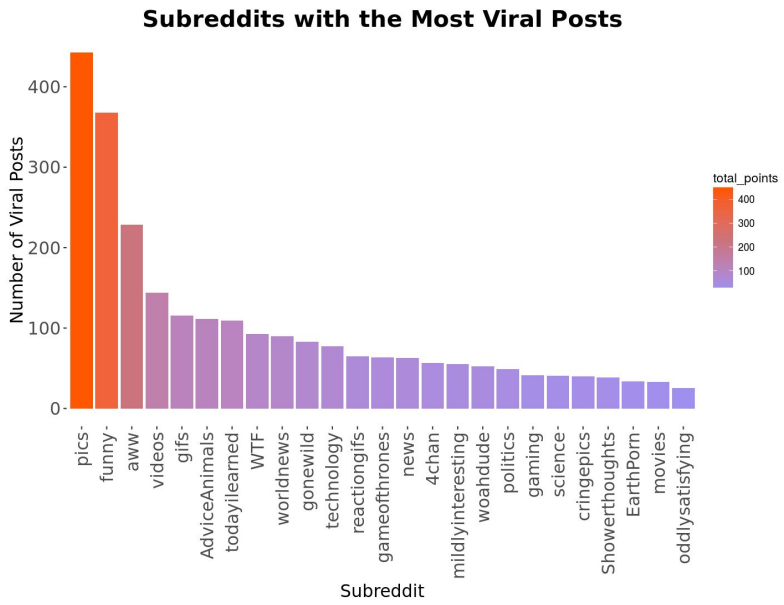

Figure 4**. Displays the count of viral posts in each subreddit. Generated by code found in **APPENDIX C.2

Based on these plots, the subreddits “pics”, “funny”, and “aww” appear to be the subreddits with the most activity and potential for a successful post. This could be due to a few factors including the brevity of posts in these subreddits, the inclusion of pictures, and the most importantly their status of “default subreddit”. A default subreddit is any subreddit that users are automatically subscribed to when they create an account. They can be easily removed, but new users to Reddit often leave the settings as they are upon creation. This means these subreddits are essentially an introduction to Reddit and will be many people’s first experience with different subreddits.

Activity was also analyzed in respect to comments but in this situation votes and virality were not the metric used. In order to comment on Reddit, a user must click into a thread and either reply to the original post or to other comments. This means that every commenter has looked at the thread and seen at least a few of the comments. This led us to use number of comments in a subreddit as the metric to determine comment visibility. Figure 5 is similar to Figure 3 except the analysis uses total comments in each subreddit instead of total votes.

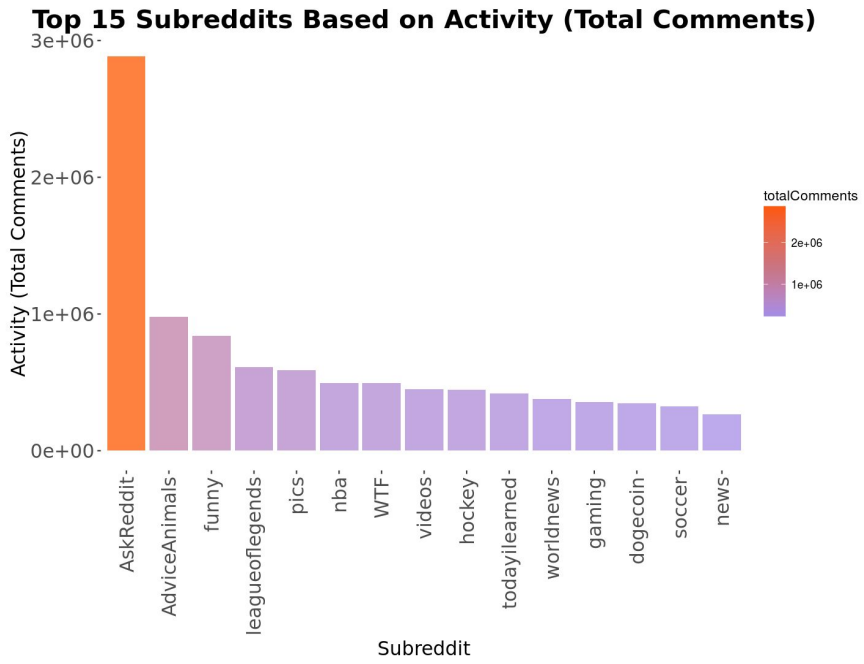

Figure 5**. Displays the top 15 subreddits based on the total count of comments on posts in the subreddit. Generated by code found in **APPENDIX C.3

This plot would indicate that “AskReddit” is far beyond its competitors in terms of activity for commenting. It has far more comments than other subreddits which makes it a top contender for submitting comments. This makes sense as “AskReddit” is a discussion based subreddit where posts are questions and people comment with their responses. In order to confirm and look further into the accuracy of this metric, a network of posts and comments was created to help visualize the depth of discussions in popular subreddits. This process was more intensive than the other graphics in this report and the following section will be thorough sub-report on this specific analysis.

Comment and Post Networks

Overview

Reddit posts are links of text, images, YouTube videos, or other forms of media submitted to a particular subreddit (subforum). Users on Reddit can choose to reply to a post by submitting a comment. These comments can gain upvotes and downvotes by voting from the rest of the Reddit community. Users can also reply to direct comments (we’ll call these **top-level** comments), and subsequently those can be replied to as well.

The result is often that a comment chain can continue to quite significant depths, thus forming a very long, spindly branch in the comment tree.

We are interested in investigating the particular shape of these comment trees, depending on which subreddit their posts belong to. We hypothesize that comment trees from posts submitted to more discussion-focused subreddits like “AskReddit” will have longer, more spindly branches due to extensive replies and discussion. On the other hand, posts submitted to subreddits like “funny” might be short and humor-driven, so we might expect more bushy, wider branching structures.

The knowledge of these structures may provide key insights to ways of earning more Karma (or upvotes) on the platform. The presence of long, spindly trees with large nodes (higher Karma) might suggest a stronger potential for earning Karma, while bushy trees with many more tiny nodes suggest that most top-level comments fall short of considerable discussion and upvoting.

Methods

We considered first the top 25 posts of all time on Reddit, from no particular subreddit, and tried to analyze the network from that. The result was too highly connected, with many overlapping regions, so in our second iteration we looked at the top 10 posts individually from three top subreddits: “AskReddit”, “funny”, and “pics”.

After we ran the Python script that wrote large csv files for both posts and comments, we read those into R and proceeded with data cleaning and wrangling. Most of the cleaning was straightforward from the dplyr package: converting variables into factors, times into POSIXct date values, and selecting the necessary columns. The next step was to convert the data frames of posts and comments into the proper formats to be understood by the igraph library.

We utilized the igraph and network packages to generate all of the visuals for these comment trees. Each graph must be defined by two separate data frames before they can be processed:

- **Nodes:** a data frame of all unique node id’s, and several auxiliary variables to determine node weights, color, etc.
- **Edges:** a data frame of adjacencies, which contains all edges connecting nodes and auxiliary variables for determining edge weights and color

The Nodes table was generated by first creating separate data frames for all subreddits, posts, and comments, and finally joining the three together. The techniques required for this step were basic data wrangling manipulations using mutate, group_by, summarize, rbind, etc. The Edges table was generated by filtering the Nodes table and collecting all parent id's of each entry in addition to the row id's. Additionally, a caveat of the LGL (Large Graph Layout) algorithm we used for igraph is that the entire graph must be connected, so we had to do further processing to connect each subreddit node with every other subreddit node. After these two tables were completed the rest of the task was to use the auxiliary variables within the tables to alter colors, sizes, and widths of the resulting elements in the network.

Results and Discussion

First we will define some terminology that we will use for the rest of this section. Networks are connected directional graphs, which can be represented as G(V, E), where **V** is the set of vertices and **E** is the set of all edges. The **depth** of a vertex *v* is a measure of the number of iterations needed of getting *v*'s parent (and so on) until reaching the top-level comment. Every vertex (node) is incident to a number of edges (there may be multiple edges coming out, but there can be at most one edge pointing in, since every node has at most one parent). Then, we will define the number of edges coming out of a vertex *v* as the **outdegree** of *v*. Each edge points from a **source** vertex to a **target** vertex, but arrows indicating direction are not portrayed in the results to more clearly emphasize vertices. The legend for coloring is as such: Blue - Subreddit, Yellow - Post, Orange - Comment. Vertex size represents the total Karma, where for a post or comment, this is the Karma of the submission; for a subreddit, this is the total Karma of all submissions connecting to it. The algorithm used to generate the final graph is LGL, which uses force repulsions between nodes determined by their total outdegree.

An initial network graph of the top 25 posts on the reddit platform showcases the highly connected network of posts and comments. For every post, there is a swarm of connected comments, particularly right next to the source. These small top level comments are so prevalent that they often completely cover the vertex that actually represents the post. This is clear in Figure 6, where small orange vertices (comments with relatively less Karma) cluster closely around large yellow vertices (a top post). The branching structure of comments is also evident: a comment vertex can have any finite outdegree, depending on the number of users who responded to that particular comment. It can be seen from Figure 7 that larger comment vertices usually correspond to a higher outdegree, but very few vertices in this particular network have as high of an outdegree as the originating post vertex. Comments with higher Karma are also

generally more shallow in the comment tree, and this may be directly correlated with Reddit's algorithm of showing which comments to display on the initial page (further comments can be expanded by clicking "see more comments").

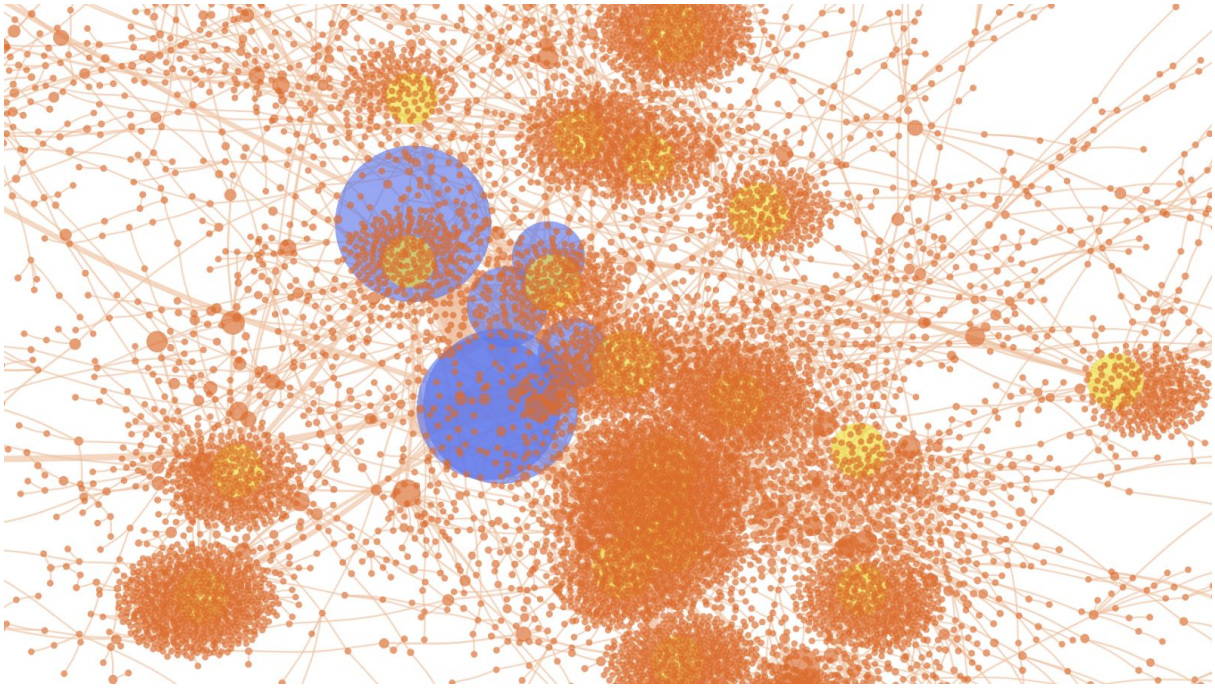

Figure 6. *An initial network of the top 25 Reddit posts of all time. Subreddits are clustered in the central region because force repulsions between vertices v and w are calculated by the total outdegree of v and w, not their visual size. Visual size instead represents the total Karma of the vertex. Generated by code found in* ***APPENDIX A.2*** *and* ***D*** *with settings* ***A.3.1.***

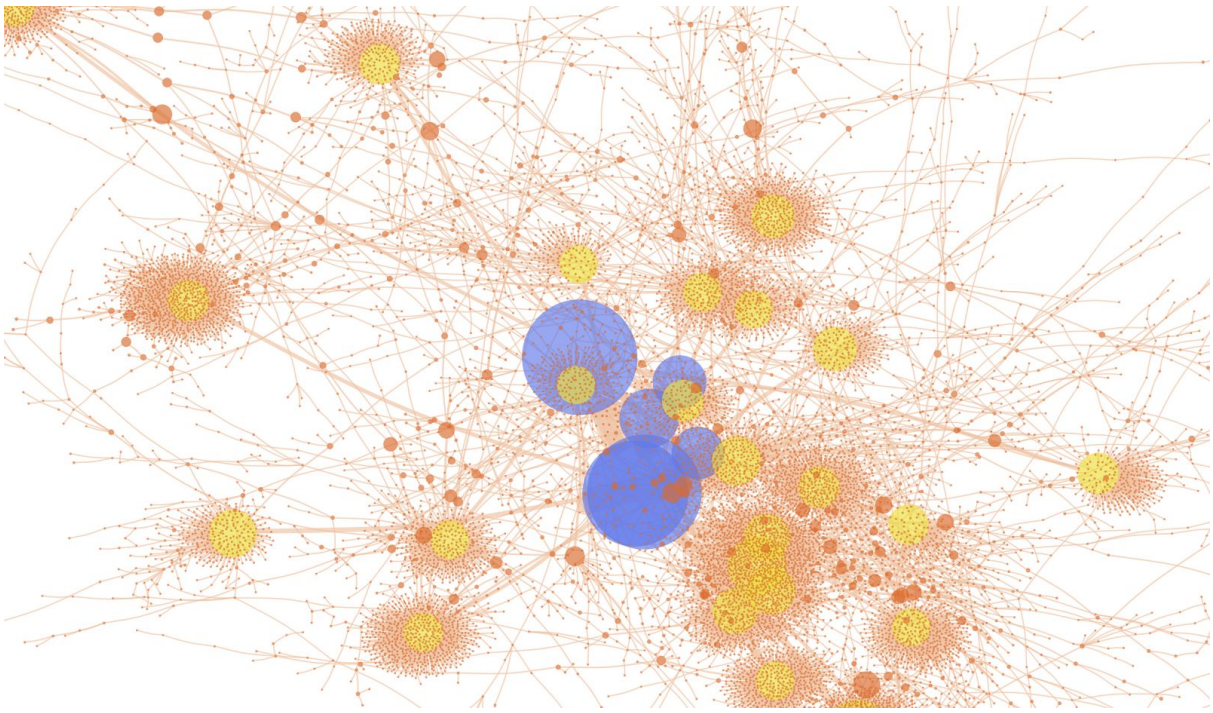

Figure 7. *The same network of top 25 posts, but drawn with thinner weights on E, and smaller sizes on V. This more effectively portrays the clustering nature of high-Karma comments, which are generally top-level or at least very shallow in any comment tree. Generated by code found in* ***APPENDIX A.2*** *and* ***D*** *with settings* ***A.3.1.***

These initial graphs do not give much more insight to the correlation between Karma density and locations on the Reddit network, but they serve as a good primer to recognize some important structures. First, the clustering of large vertices (high Karma posts) is of particular importance, since we would like to know specifically where in the network it would be optimal to write a comment. Then, the length of a particular branch is also a necessary measure to indicate the amount and length of discussion stemming from a particular comment. Optimally, we would be able to find a location with a high degree of clustering in addition to very long comment branches.

Results: *AskReddit* Network of Top Ten Posts

Perhaps the most elucidating data set we looked at was the set of top posts from the subreddit “AskReddit”. In our hypothesis, we predicted that a discussion-focused subreddit like “AskReddit” would be a good candidate for writing high-Karma responses, since people would more likely be actively browsing the comments section.

As seen from Figure 8, the branching structures from posts are extremely broad and deep. This is highlighted by how far the upper-left cluster is from the central region of the graph. This is due to the repulsion between that cluster and the rest of the network, and such an extreme degree of repulsion is indicative of densely connected clusters with high outdegrees. The depth of comment trees is also evident in the network: the number of long, spindly chains of comment vertices is much greater in this graph than in Figure 7, where bushier, shorter branches were more prevalent.

In Figure 9, which is a zoomed-in display of the upper left cluster in Figure 8, it is clear that the network of comments is both wide and deep. The large majority of comments fail to have any responses, remaining closely clustered towards the center of the post. However, particularly insightful or interesting comments have a huge potential to stir up conversation, in the form of long comment chains. It is particularly interesting to note that the larger comment vertices are clustered near the center, but comment vertices beyond a certain depth rarely have more than one or two children. This model is descriptive of a tree that has significant branching at shallow depths, but very sparse branching at higher depths. In context, top-level conversations on the “AskReddit” forum are plenty, and they stir up the most replies (generating the most Karma as well). Comments that are embedded below many other comments, however, rarely receive more than one or two replies, partly owing to Reddit’s own algorithm of displaying comments and hiding deep comments below “see more comments” tags.

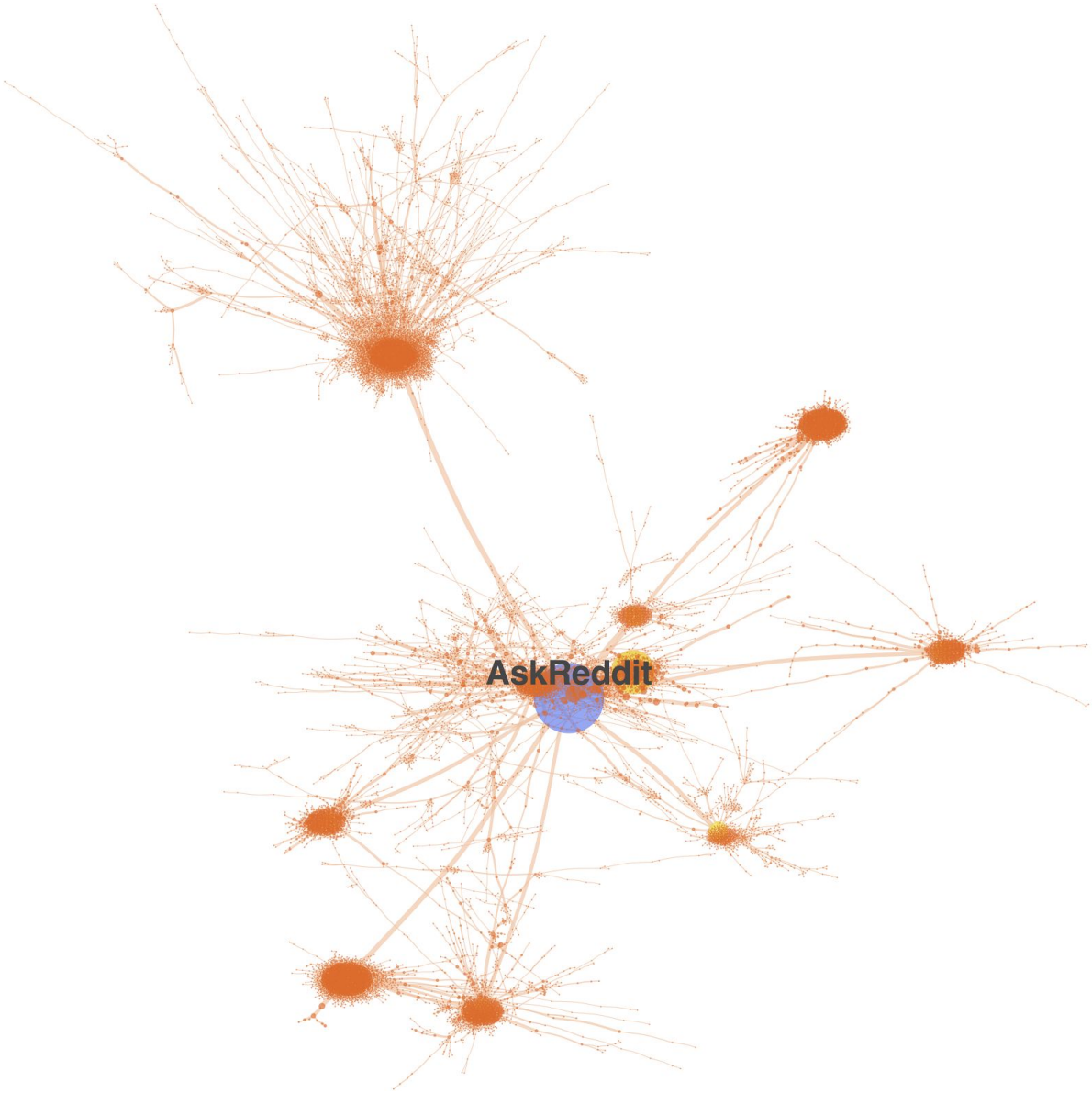

Figure 8. *The complete network of top ten all-time posts from "AskReddit". Elaborate and deep branching structures from all posts can be seen, in addition to large comment clusters. Generated by code found in* ***APPENDIX A.2*** *and* ***D*** *with settings* ***A.3.2.***

Figure 9. *A closer look at the branching structure of one post extending from "AskReddit". While the majority of comments are top-level low-Karma vertices clustered in the center of the post, there are also many extensive spindly branches extending from the root. Additionally, many comment vertices at relatively shallow depths are large, representing higher-Karma responses. Generated by code found in* ***APPENDIX A.2*** *and* ***D*** *with settings* ***A.3.2.***

Results: *funny* and *pics* Networks of Top Ten Posts

"funny" and "pics" are subreddits that are known to have more humorous content, often images or videos. These particular forums rarely have beyond superficial discussion of the subject material, and often top-level comments are brief, sarcastic reactions. As seen in Figure 10, the full network of top posts from "funny", the network branches highly but has few comment chains that are very deep. The deeper chains are mostly characterized by a series of high-Karma responses to one another, suggesting that for a deep comment chain, subsequent replies are all of a comparable size to the origin to the maintain the magnitude and visibility of the response. A noticeable aspect of Figure 10 (shown in more detail in Figure 11) is the cluster of large comment vertices at the bottom right, which effectively showcases this inference.

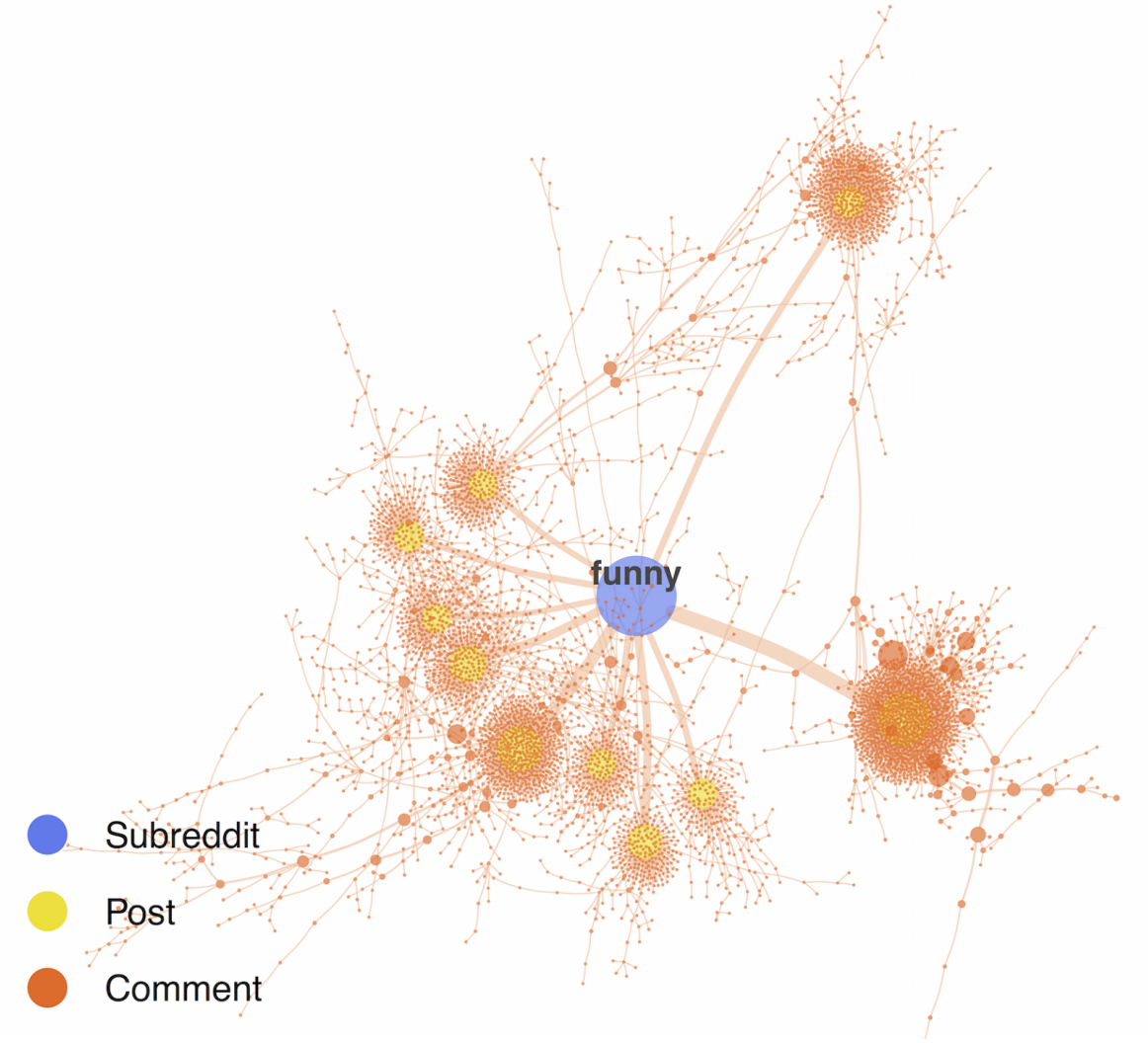

Figure 10. *The full network of posts and comments extending from "funny". Deep branching structures are sparse, but a significant aspect is the subnetwork of large comment vertices at the lower right. Generated by code found in* ***APPENDIX A.2*** *and* ***D*** *with settings* ***A.3.3.***

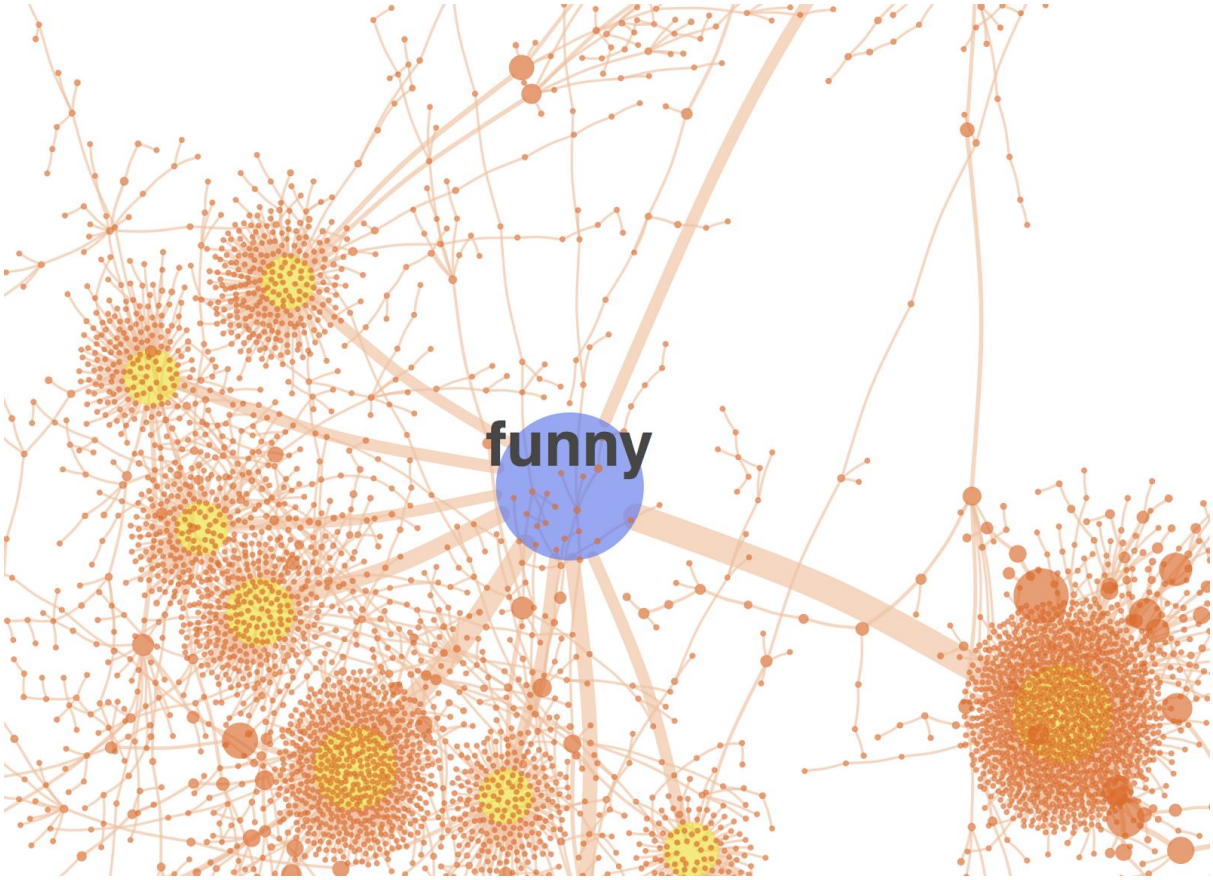

Figure 11. *A closer look at the central point of the network. Almost all comment replies are top-level vertices that don't garner a large amount of Karma. Branches are more dense, since the average depth of comments is much lower than that of "AskReddit". Generated by code found in* ***APPENDIX A.2*** *and* ***D*** *with settings* ***A.3.3.***

Likewise, the network for "pics", shown in Figure 12, has very extensive branching at lower depths, but very few tendrils representing longer, more deeper conversations. Top-level small comment vertices are even more prevalent in this network, greedily clustering around the original posts. An observation more clear in this network than that of Figure 11 is that the posts typically have one or two top comments that are dominant over all other responses. These appear as large auxiliary nodes to the posts, and one or two of these can be seen around every post vertex. Very few large vertices beyond these auxiliary nodes exist elsewhere around the graph, unless there is a continuous chain of high-mass comments, as discussed in the "funny" network.

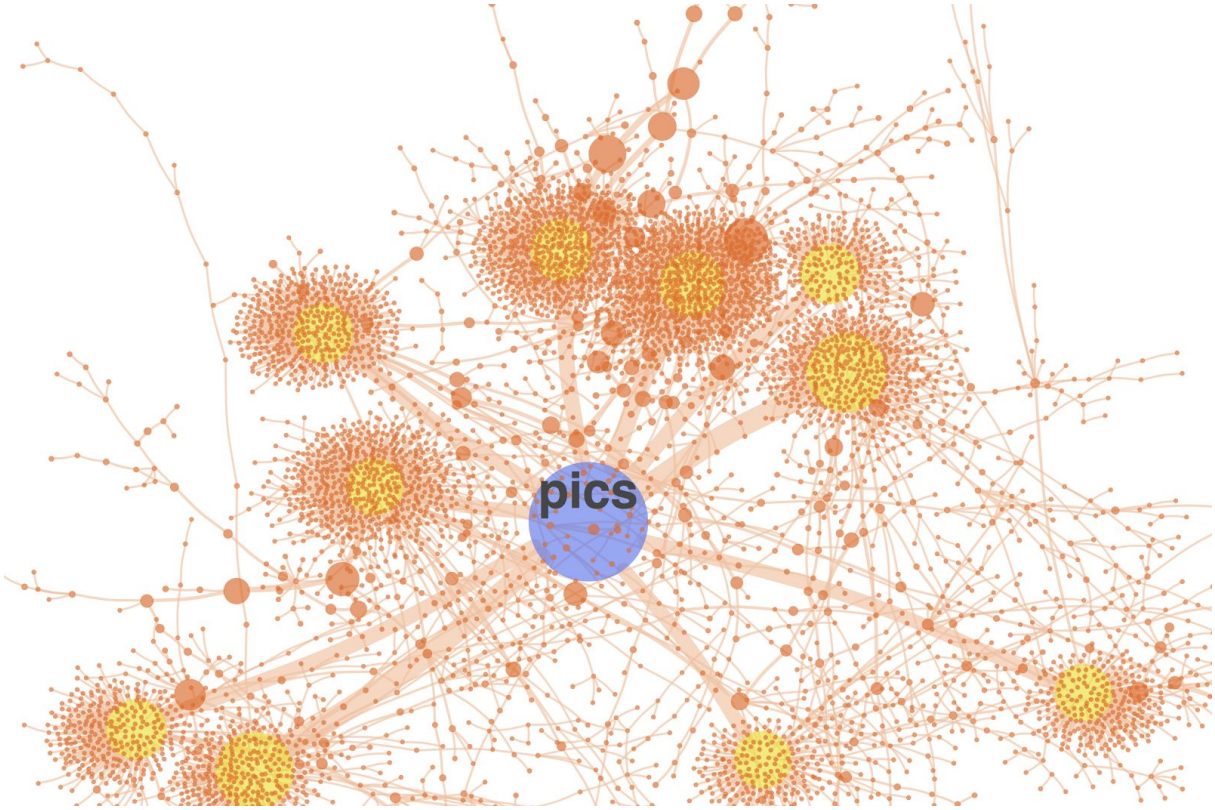

Figure 12. The "pics" network is very similar to the "funny" network, with very few long branches, and very dense clustering of top-level comments that don't score a large amount of Karma. Near by every post, there is at least one or two comments that have significant mass (these are the top replies), but very few beyond that have comparable mass. Generated by code found in ***APPENDIX A.2*** *and* ***D*** *with settings* ***A.3.4.***

Where to Post and Comment: Closing Thoughts

As the network demonstrated, "AskReddit" contains much deeper and lengthy discussions where people read comments and respond to each other. This makes it a perfect environment to post comments since it is one of the few subreddits where people frequently look through the comment section.

In conclusion, the best subreddits to post in based on both number of votes and number of viral posts are "funny", "pics", and "aww", while the best subreddit to comment in is "AskReddit". These subreddits are popular subreddits that are characteristically conducive for high traffic and visibility which leads to a better chance at being successful.

When to Post and Comment

Beyond posting to an appropriate subreddit, it is important to consider when to post. The time that a post is created has strong implications on whether or not it will be successful in the end. This is because despite Reddit being a global community of users, there are trends in the amount of traffic or activity on the website. Posting when there is a lot of activity improves the chance that someone will see the it and give it the boost it needs to start accumulating Karma. First, we took a look at hour of the day and tried to determine if it resulted in a difference in Karma. Figure 13 is a bar graph that shows the variation of average Karma based on what hour it was created.

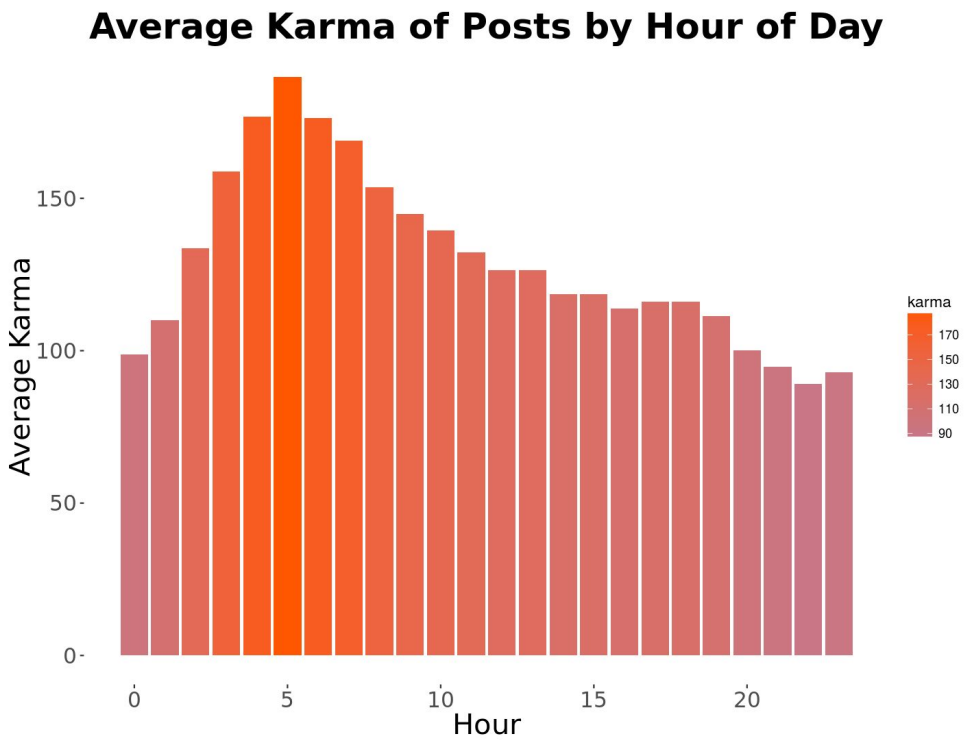

Figure 13**. Displays the average Karma of posts at given hours in the day where Hour is given in 24hr format based on GMT. Generated by code found in **APPENDIX E.1

From Figure 13, there appears to be a peak in Karma for posts made around 4-6 am GMT (9-11 pm PST). The increase around this time is significant but unexplained by any immediately obvious factor. One potential factor we looked into is whether or not this increase correlated to common work schedules. In order to analyze this, we broke the groups down further into day of week as well of hour of the day. If our assumption about work schedules was correct, this trend would be highly visible on weekdays but

less noticeable on weekends when fewer people are working. Figure 14 uses size and color to relay this information about average Karma based on two metrics of time.

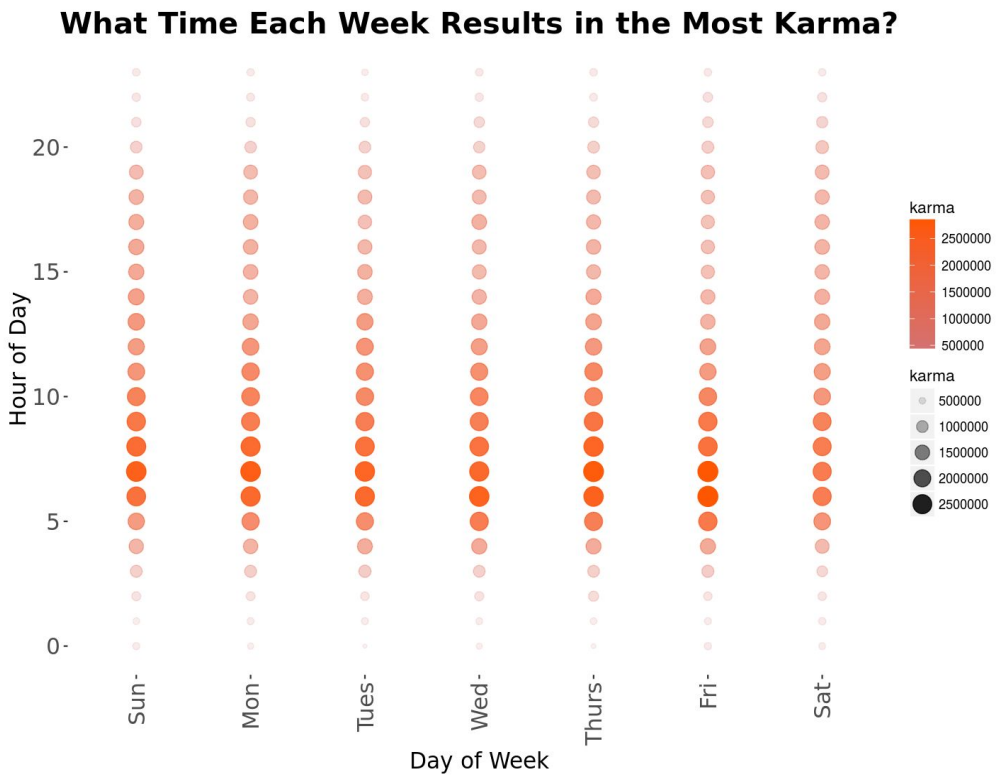

Figure 14. *Graphic which displays a heatmap of total Karma for two categorical variables, Day of the Week and Hour of the Day. Generated by code found in* ***APPENDIX E.2***

Contrary to our hypothesis, the trend of Karma based on hour of day is consistent across every day of the week. The peak in Karma is still around 5 am GMT for every day of the week. This means that work schedule is unlikely to be the primary influence on this spike in Karma. Regardless, this time period appears to have a correlation with successful Karma posts.

Another approach we took to this question is to look at the top users in Reddit and observe their habits for posting and commenting. The motive for this is to learn how to be successful from those who have already been successful. This was done for both posts and comments and the results of the analysis is seen in Figures 15 and 16.

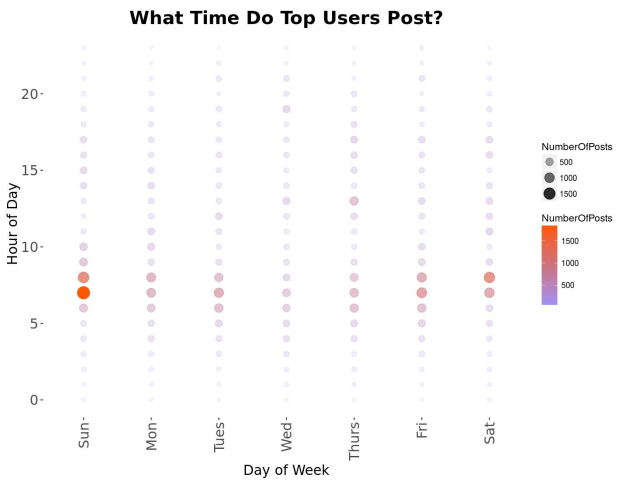

Figure 15. *Graphic that is similar to Figure 14 except isolated to posts from top users. Generated by code found in* ***APPENDIX E.3***

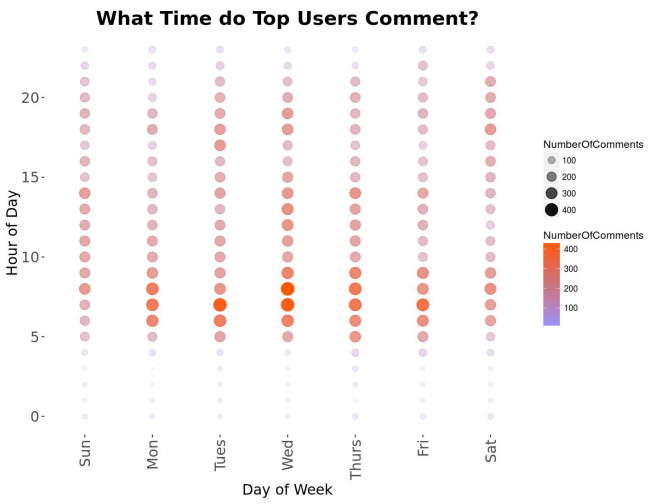

Figure 16. *Graphic that is similar to Figure 14 except isolated to comments from top users. Generated by code found in* ***APPENDIX E.4***

Figure 15 and 16 were generated very similarly to Figure 14 where both day of the week and hour of the day were considered. For posts, there appears to be a much weaker trend in amount of Karma across the hours of the day. All the dots are approximate in value except for peak time on Sunday which appears to be much greater than other times. This suggests that posts from top users are very heavily concentrated at this point in time on Sunday. Every other time is faint and has very few posts from top users so there must be some attribute that makes this time much more successful. Comments on the other hand generally hold the same trend as posts in general except shifted about an hour or two later. This indicates that comments are dependent on post times and the most successful comments appear and hour or two after the post is created. There is no single sweet spot to comment like there is for posting since comments are dependent on successful posts to be successful comments. This is supported by the fact that top commenters have comments that follow closely after top posts; since the posts from that time are successful, the comments in those posts have a greater visibility and are more likely to be successful as well.

In conclusion, the best time to post appears to be around 4-6 am GMT on Sunday specifically while the best time to comment appears to be an hour or two after a post is made. This time has a tendency to lead to successful posts while Sunday is a popular time to post among the top users on Reddit. This specific time is still unexplained, but we have inferred that it does not have to do with working hours. The best time to comment appears to be directly related to the best time to post which suggests that

successful comments piggyback off of successful posts and comments made shortly after the post was made are some of the best.

What to Post and Comment

Possibly the most important factor to a successful submission to Reddit is the content of the submission. A lot of variables play into what makes a post or comment successful, such as context or relevance to recent news, but this analysis will only look at submissions in general due to limitations on computing ability and scope of data.

The heart of many posts and comments is the text that make up the bulk of the content. While it is true many posts are mainly pictures, there is still a title which is what grabs the attention of users skimming the front page. And so our first analysis is a look at the top users of Reddit and the most frequent words in their submissions as seen in Figures 17 and 18.

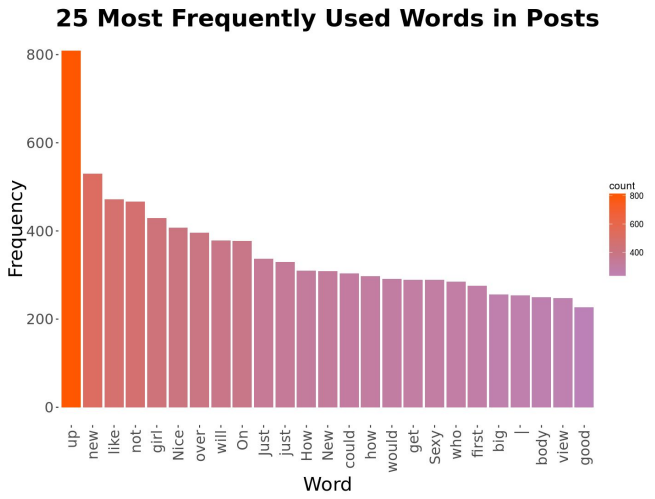

Figure 17**. Most frequently used words in posts where small articles such as "the", "a", and "of" are removed. Generated by code found in **APPENDIX F.1

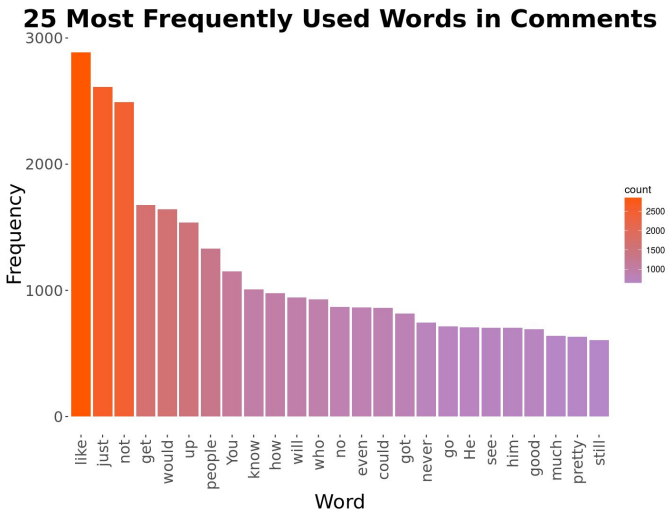

Figure 18**. Most frequently used words in comments where small articles such as "the", "a", and "of" are removed. Generated by code found in **APPENDIX F.2

Despite best efforts to remove as many small articles as possible from this analysis, some of the words in these lists are uninformative. This does not mean, however, that all of these short descriptions are unrelated. In terms of posts, generally positive words such as "up", "new", and "nice" are frequently used in titles of posts. This is not surprising as titles are meant to grab attention and make viewers interested. Negatively connotated words upset some users and make titles disdainful. Also among

the top words in posts are “girl”, “sexy”, and “body”. This is also unsurprising since Reddit’s user base is primarily males and there are many subreddits dedicated to images of the human body and associated media. Comments do not have themes that are as clear, but many of the words are pronouns. This is partially due to the fact that comments are replies to other content so it is common for them to reference another user or person in their reply.

Comments are in general too based on context to find general trends in success since they are all inherently replies. Because of this, we will focus on posts and specifically factors that affect the title since that is what attracts the most views. First we looked at the length of titles since we believed shorter titles to be more friendly towards users just skimming the page. The plot of title length vs Karma is seen in Figure 19.

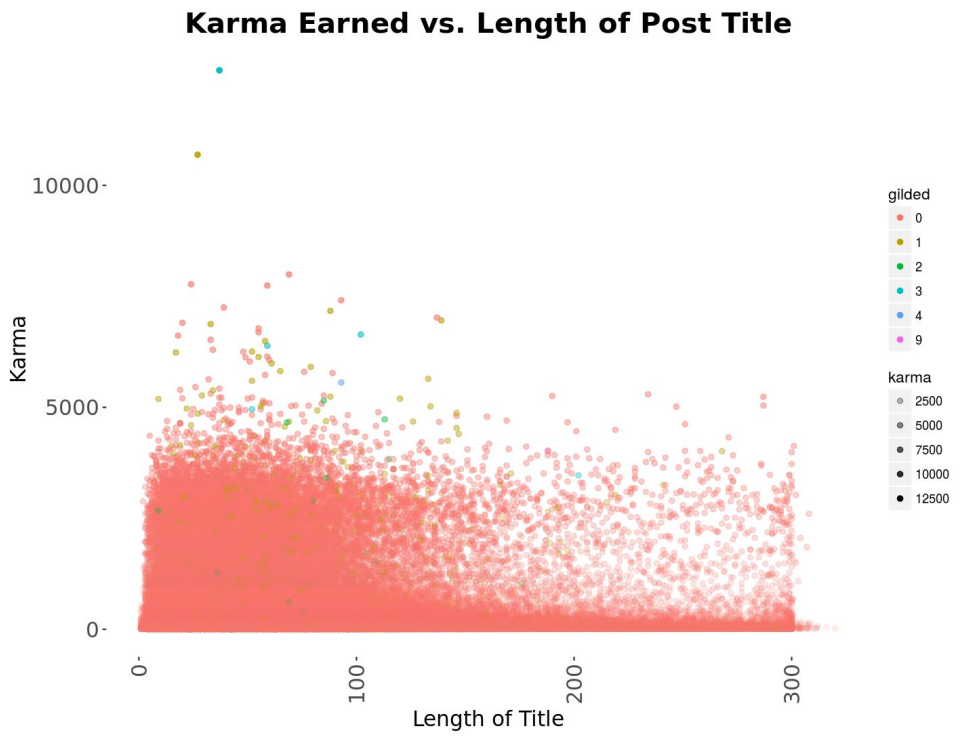

Figure 19. *Scatter plot of the number of characters in the title and how much Karma was earned by the post. Generated by code found in* ***APPENDIX F.3***

At first glance, Figure 19 appears to show that longer titles have worse Karma values. However further analysis showed that there are simply more titles with fewer character and the distribution is approximately the expected amount of randomness.

We conclude that there is no significant relationship between length of the title and Karma earned.

The analysis of post title continued with a look at the structuring of the title and the inclusion of specific key phrases. By structure of the title, we are referring to the usage of punctuation and proper capitalization. We assume that a well structured title is more pleasing to viewers and would lead to more views and greater Karma. Figure 20 is the graphic we created of the distribution of posts based on whether or not they had titles that contained punctuation and capitalization.

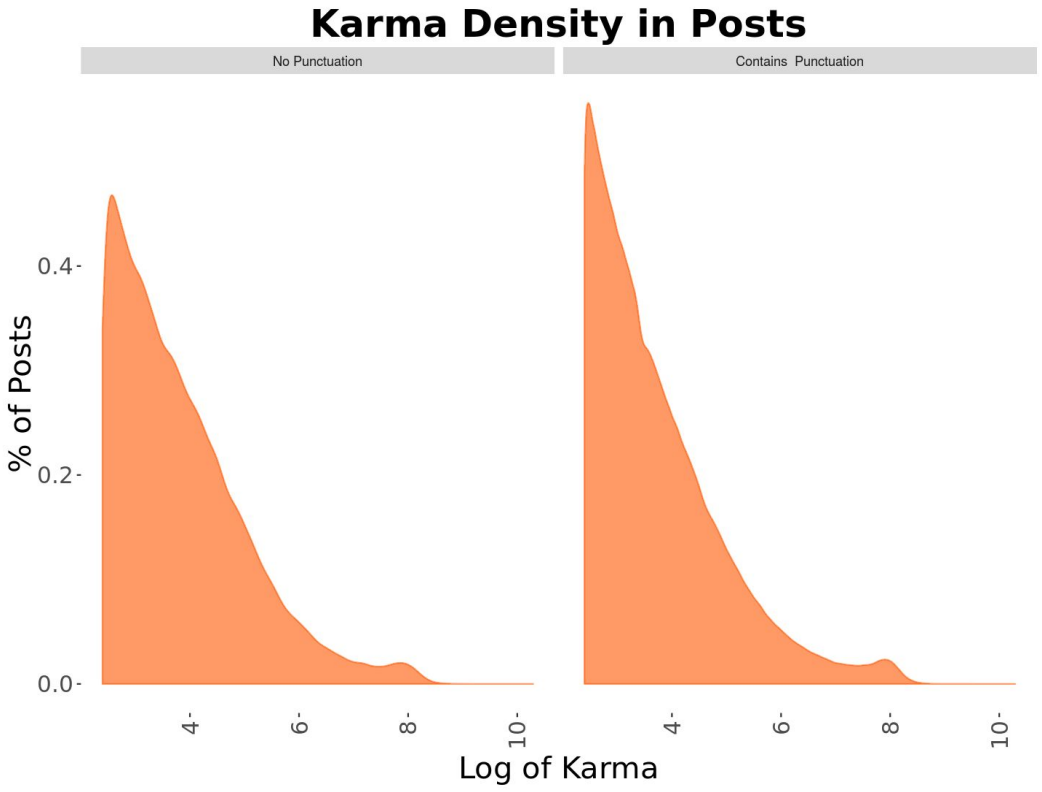

Figure 20. *The distribution of posts on a log scale that is faceted by whether or not the title is well structured. Generated by code found in* ***APPENDIX F.4***

The distributions are very similar in shape and the difference would appear to be insignificant. For this test we also calculated the mean and median Karma for both categories and found that a structured title resulted in the following:

Table 1. *Mean and Median for Structured vs Unstructured Titles in Posts*		
	Mean	**Median**
Structured Titles	130.67	31
Unstructured Titles	132.50	35

If anything, this would indicate structured titles perform worse, but the difference is small and the difference in number of posts in each category make the distinction irrelevant in our analysis. After completing this test we thought to check whether or not certain phrases in the title led to more success using a similar method as we used for structured vs unstructured. We chose a phrase, “TIL” meaning “Today I Learned”, since it appears frequently and is easy to detect since it is relatively unique to Reddit and is not part of common English. Figure 21 is a distribution created similarly to Figure 20 except we used the presence of “TIL” as our faceting factor.

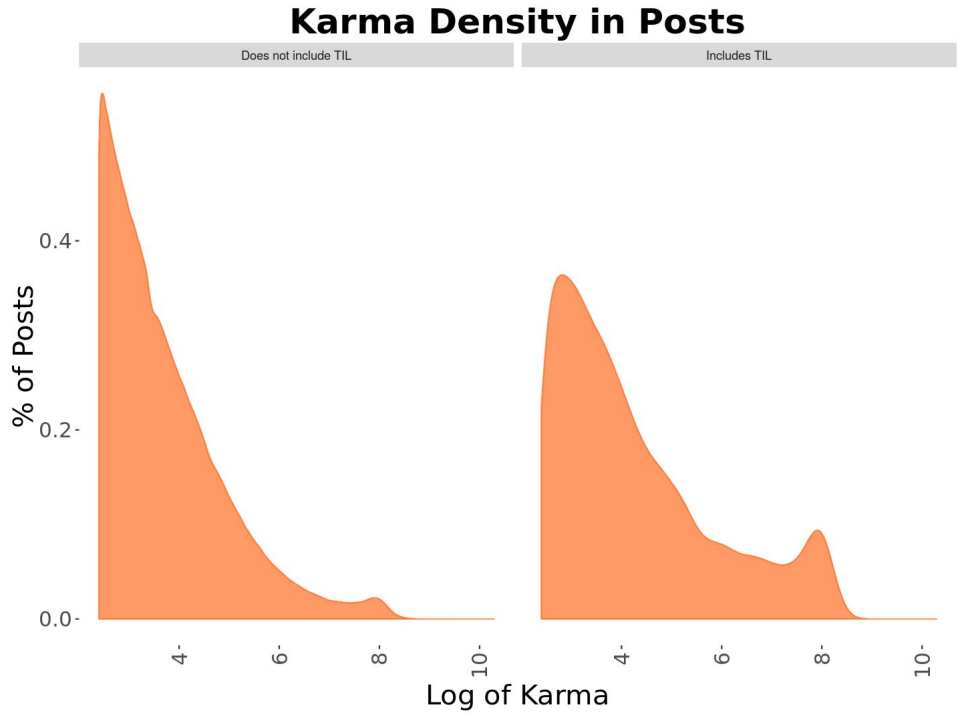

Figure 21. *Distribution of posts based on their Karma and faceted by whether or not the phrase “TIL” was contained in the title. Generated by code found in* ***APPENDIX F.5***

Unlike our analysis on the structure of titles, this analysis on the presence of a key phrase resulted in significantly different distributions. Including “TIL” appears to shift the distribution towards higher Karma values and a look at the values of the mean and median confirm this.

Table 2. *Mean and Median for Includes vs Does not Include “TIL”*		
	Mean	**Median**
Includes “TIL”	336.38	46
Does not Include “TIL”	128.93	31

It appears that including “TIL” tends to result in a more successful post. This could be because adding “TIL” to a title is essentially a tag that is quickly recognizable and indicative of the content of the post. Anyone who frequently uses Reddit and sees “TIL” will know a little bit about what will follow. Also, “TIL” is a phrase found mostly in Reddit and creates a sense of community since it is practically exclusive to this environment. It is a phrase that certainly makes a difference and helps lead to high Karma posts.

In conclusion, the best content to submit to Reddit is hard to pinpoint since so much is situational and contextual, however there are a few general trends that seem to help. For posts, using positive words in the title and using short phrases unique to the community that also relays a lot of information leads to generally higher Karma. For comments, there is a much looser theme since it is even more referential and contextual since they are replies.

Summary, Conclusion, and Looking Forward

The overarching goal of this report was to answer the question of how to be successful on Reddit. This was accomplished by answering the questions of where, when, and what to post and comment. We began this paper by explaining the process we used to collect our data, from BigQuery to data scraping with Reddit’s API. Our data is primarily from April-August 2014 and is taken from 1TB+ of data which is the entire corpus of all Reddit posts and comments. The networks did not use this data and instead used the data that was scraped directly from Reddit.

Next, we established our metric for success and identified what we considered to be the most successful users from our data set. This led into the discussion of which subreddits are the best to post and comment in. Our simple bar graphs indicated that “funny”, “pics”, and “aww” are the best to post in and “AskReddit” is the best to comment in. This was analyzed much more deeply in our creation of the comment and post network. From our subreport analyzing the networks of subreddits, their posts, and comments, it is clear that while the structure of the interaction network on Reddit is very sophisticated, there clear distinctions between different parts of the website that can be capitalized upon to acquire more Karma. The majority of Reddit’s top subreddits discourage significant back-and-forth discussion within the comments. The result is widely branching and shallow comment networks, with few top comments taking the majority of the Karma mass generated by posts. On the other hand, discussion-driven forums like “AskReddit” encourage active browsing of comments and responding to them. The result we desire, a dense network of high-Karma comments and very long comment chains, is precisely exhibited by the “AskReddit” network. While it is possible for one to post the top-scoring comment in “funny” or “pics”, the potential is much higher for consistently high Karma scores in “AskReddit”.

After looking at where to post and comment, we looked at when the best times to submit content was. This was done by looking at the times that resulted in the most Karma across the hours of the day and the days of the week. Our first analysis using all posts led us to conclude 4-6 am GMT had a tendency to have high Karma posts across every day of the week. We then considered only top users and found that Sunday specifically had an incredibly high concentration of posts from these people. The best time to comment was determined to be related to the time of the original post and fell about one or two hours after the post was made. This is because comments ride on the success of the post they belong to so the better the post the better the comment is likely to be.

Finally, we looked at what content made a post or comment exceptional and found a few aspects to be helpful. Comments were found to be hard to pinpoint because the material is entirely contextual and is dependent on the original post or other comments. Posts on the other hand had interesting trends in what made for a compelling title. We found that positive words were helpful and that the phrase “TIL” led to significantly more Karma while the structure of the title did not seem to matter too much. In the end it is hard to generalize what content is superior, but our analysis brought interesting insights on what made titles more inviting than others.

In the end, Reddit is a very complex website and it is next to impossible to pinpoint success on it to a single word or good timing on posts. It consists of the continual interaction between millions and millions of users which is hard to generalize in any sense of the word. But still, we try our best to find trends and argue what tendencies lead to better results. As a whole, we are excited about the results we found but our analysis only leaves us with more questions to ask. It would be interesting to create more networks of subreddits and to do a deeper analysis of all subreddits vs default subreddits. Also, another look at when to comment in respect to when posts are made could lead to more insights and good commenting. Finally, one of our more compelling findings is related to the phrase “TIL”. It would be useful to conduct similar tests on other phrases and to identify all of the phrases that Reddit seems to love. Reddit has been incredibly difficult to wrangle, but taking a deep look at any society is challenging and incredibly rewarding in the end. We hope to be able to continue some of our research and to bring even more insightful tips on how to be successful on Reddit. Also don’t forget to send us an upvote!

Appendix

Appendix A - Data Collection

A.1: Big Query

```
projectID = "elated-coil-129122"

sqlPosts = "SELECT subreddit, ups, downs, num_comments, title, gilded,
created_utc, author, subreddit_id, id, name
FROM [Reddit.AllPosts]
WHERE YEAR(SEC_TO_TIMESTAMP(INTEGER(created_utc))) == 2014 and
(MONTH(SEC_TO_TIMESTAMP(INTEGER(created_utc))) > 4 and
MONTH(SEC_TO_TIMESTAMP(INTEGER(created_utc))) < 9) and score > 10"

sqlComments = "SELECT author, ups, downs, subreddit, created_utc, name, id,
parent_id, subreddit_id, gilded, body
FROM [Reddit.2014Comments]
WHERE YEAR(SEC_TO_TIMESTAMP(INTEGER(created_utc))) == 2014 and
(MONTH(SEC_TO_TIMESTAMP(INTEGER(created_utc))) > 4 and
MONTH(SEC_TO_TIMESTAMP(INTEGER(created_utc))) < 9) and score > 10"

Posts = query_exec(query = sqlPosts, project = projectID, max_pages = Inf,
destination_table = "Reddit.SamsPosts")

Comments = query_exec(query = sqlComments, project = projectID, max_pages = Inf,
destination_table = "Reddit.SamsComments")
```

A.2: Python scraping with PRAW and writing post/comment data to csv

```
__author__ = 'kenchen'

import praw
import csv
```

```
def combine_subreddits(subs):
    return "+".join(subs)


r = praw.Reddit("Network visualization data scraper v1.0 by u/k-a-n")

subreddits = ["funny", "AdviceAnimals", "pics", "aww", "WTF",
              "todayilearned", "gaming", "videos", "gifs",
"leagueoflegends"]

# Getting top [limit] posts from multi-reddit defined from subreddits
above
top_subreddits = r.get_subreddit(combine_subreddits(subreddits))
top_posts = top_subreddits.get_top_from_all(limit=25)

# Write posts csv table
with open('postssmall.csv', 'wb') as csv_posts:
    post_attrs = ["id", "subreddit_id", "subreddit", "author", "title",
"created_utc", "ups", "downs", "gilded"]
    pwriter = csv.writer(csv_posts, delimiter=',')
    pwriter.writerow(post_attrs)

    # Write comments csv table
    with open('commentssmall.csv', 'wb') as csv_comments:
        comment_attrs = ["id", "parent_id", "subreddit", "author",
"created_utc", "ups", "downs", "gilded"]
        cwriter = csv.writer(csv_comments, delimiter=',')
        cwriter.writerow(comment_attrs)

        for post in top_posts:
            # Write this post with variables post_attrs to file
            pwriter.writerow([getattr(post, attr) for attr in post_attrs])

            # Replace MoreComments objects with more Comments (expand
comment tree deeper)
            post.replace_more_comments(limit=post.num_comments // 100)
            post_comments = praw.helpers.flatten_tree(post.comments)

            # Write all Comment objects in the flattened comment tree
            for comment in post_comments:
                if type(comment) != praw.objects.MoreComments:
                    cwriter.writerow([getattr(comment, attr) for attr in
comment_attrs])
```

A.3: Inputs and settings for generating graphs

1. As is, with the code above in A.2
2. ```
   subreddits = ["AskReddit”]
   top_posts = top_subreddits.get_top_from_all(limit=10)
   ```
3. ```
   subreddits = ["funny”]
   top_posts = top_subreddits.get_top_from_all(limit=10)
   ```
4. ```
   subreddits = ["pics”]
   top_posts = top_subreddits.get_top_from_all(limit=10)
   ```

Appendix B - Determining the Top Users

B.1: Top Posters

```
TopUsers = Posts %>%
  filter(author != "[deleted]") %>%
  group_by(author) %>%
  summarize(totalKarma = sum(karma)) %>%
  arrange(desc(totalKarma)) %>%
  head(30)


TopUsers %>%
  transform(author = reorder(author, order(totalKarma, decreasing = TRUE))) %>%
  ggplot(aes(x = author, y = totalKarma, fill = totalKarma)) +
  geom_bar(stat = "identity", alpha = 0.75) +
  scale_fill_gradient2(low = reddit.blue, mid = "#CA7580", high = reddit.red,
midpoint = max(TopUsers$totalKarma)/2) +
  ggtitle("Top 30 Posters and Their Total Karma") + xlab("User") + ylab("Total
Karma") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
25),
```
```

```
        axis.title.x = element_text(family = "NimbusSanCond", size = 20),
        axis.title.y = element_text(family = "NimbusSanCond", size = 20))
```

B.2: Top Commenters

```
TopUsers = Comments %>%
  filter(author != "[deleted]") %>%
  group_by(author) %>%
  summarize(totalKarma = sum(karma)) %>%
  arrange(desc(totalKarma)) %>%
  head(30)

TopUsers %>%
  transform(author = reorder(author, order(totalKarma, decreasing = TRUE))) %>%
  ggplot(aes(x = author, y = totalKarma, fill = totalKarma)) +
  geom_bar(stat = "identity", alpha = 0.75) +
  scale_fill_gradient2(low = reddit.blue, mid = "#CA7580", high = reddit.red,
midpoint = max(TopUsers$totalKarma)/2) +
  ggtitle("Top 30 Commenters and Their Total Karma") + xlab("User") + ylab("Total
Karma") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
25),
        axis.title.x = element_text(family = "NimbusSanCond", size = 20),
        axis.title.y = element_text(family = "NimbusSanCond", size = 20))
```

Appendix C - Where to Submit

C.1: Activity Based on Votes

```
SubActivity = Posts %>%
  select(subreddit, num_comments, activity) %>%
  group_by(subreddit) %>%
  summarize(totalVotes = sum(activity), totalComments = sum(num_comments))

SubActivity %>%
  arrange(desc(totalVotes)) %>%
  head(15) %>%
  transform(subreddit = reorder(subreddit, order(totalVotes, decreasing = TRUE)))
%>%
  ggplot(aes(x = subreddit, y = totalVotes)) +
  geom_bar(stat = "identity", fill = "#FF5700", alpha = 0.75) +
  ggtitle("Top 15 Subreddits Based on Activity (Total Votes)") +
xlab("Subreddit") + ylab("Activity (Total Votes)") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
25),
        axis.title.x = element_text(family = "NimbusSanCond", size = 20),
        axis.title.y = element_text(family = "NimbusSanCond", size = 20))
```

C.2: Subreddits with Viral Posts

```
ViralSubreddits <-
  Posts %>%
  filter(karma >= karmaThreshold) %>%
  mutate(points = karma / karmaThreshold) %>%
  group_by(subreddit) %>%
  summarize(total_points = sum(points)) %>%
  arrange(desc(total_points)) %>%
  head(25)
```

```
ViralSubreddits %>%
  arrange(desc(total_points)) %>%
  transform(subreddit = reorder(subreddit, order(total_points, decreasing =
TRUE))) %>%
  ggplot(aes(x = subreddit, y = total_points, fill = total_points)) +
geom_bar(stat = "identity") +
  scale_fill_gradient2(low = reddit.blue, mid = "#CA7580", high = reddit.red,
midpoint = max(ViralSubreddits$total_points)/2) +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
20),
        axis.title.x = element_text(family = "NimbusSanCond", size = 15),
        axis.title.y = element_text(family = "NimbusSanCond", size = 15)) +
  ggtitle("Subreddits with the Most Viral Posts") + xlab("Subreddit") +
ylab("Number of Viral Posts")
```

C.3: Activity Based on Comments

```
SubActivity = Posts %>%
  select(subreddit, num_comments, activity) %>%
  group_by(subreddit) %>%
  summarize(totalVotes = sum(activity), totalComments = sum(num_comments))

SubActivity %>%
  arrange(desc(totalComments)) %>%
  head(15) %>%
  transform(subreddit = reorder(subreddit, order(totalComments, decreasing =
TRUE))) %>%
  ggplot(aes(x = subreddit, y = totalComments)) +
  geom_bar(stat = "identity", fill = "#9494FF", alpha = 0.75) +
```

```
  ggtitle("Top 15 Subreddits Based on Activity (Total Comments)") +
xlab("Subreddit") + ylab("Activity (Total Comments)") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
25),
        axis.title.x = element_text(family = "NimbusSanCond", size = 20),
        axis.title.y = element_text(family = "NimbusSanCond", size = 20))
```

Appendix D - Network Visualization

R cleaning, building vertex/adjacency lists for graphs, and drawing networks

```
library(DataComputing)
library(igraph)
library(network)

wd <- "/Users/kenchen/MDST/visual_reddit/python/"
Posts <- read.file(paste0(wd, "postssmall.csv"))
Comments <- read.file(paste0(wd, "commentssmall.csv"))

# Cleaning up Posts and Comments

Posts <- Posts %>%
  mutate(created_utc = as.POSIXct(as.numeric(created_utc),
origin="1970-01-01"),
         hour = lubridate::hour(created_utc),
         karma = ups - downs,
         gilded = as.factor(gilded))

Comments <- Comments %>%
  mutate(created_utc = as.POSIXct(as.numeric(created_utc),
origin="1970-01-01"),
         hour = lubridate::hour(created_utc),
         karma = ups - downs,
         gilded = as.factor(gilded))

# Creating data frame of network nodes
```

```r
# 1 - subreddit
# 2 - post
# 3 - comment

NetworkSubreddits <- Posts %>%
  select(-c(author, title, ups, downs, gilded)) %>%
  group_by(subreddit) %>%
  summarize(id = head(subreddit_id, 1),
            karma = sum(karma)) %>%
  mutate(type = 1,
         subreddit_id = NA,
         parent_id = NA,
         created_utc = NA,
         hour = NA)

NetworkPosts <- Posts %>%
  select(-c(author, title, ups, downs, gilded)) %>%
  mutate(type = 2,
         id = paste0("t3_", id),
         parent_id = subreddit_id,
         subreddit_id = NA)

NetworkComments <- Comments %>%
  select(-c(author, ups, downs, gilded)) %>%
  mutate(type = 3,
         id = paste0("t1_", id),
         subreddit_id = NA)

NetworkNodes <- NetworkSubreddits %>% rbind(NetworkPosts) %>%
rbind(NetworkComments) %>%
  select(id, subreddit, karma, type, subreddit_id, parent_id, created_utc,
hour)

# Creating table of network adjacencies

SubredditEdges <- data.frame(from = c(), to = c(), weight = c(),
created_utc = c())

for (i in 1:nrow(NetworkSubreddits)) {
  for (j in i:nrow(NetworkSubreddits)) {
    if (i != j) {
      SubredditEdges <- SubredditEdges %>% rbind(
        data.frame(from = c(NetworkSubreddits$id[i]),
                   to = c(NetworkSubreddits$id[j]),
                   weight = c(NetworkSubreddits$karma[i]), created_utc =
```

```
NA)
       )
     }
   }
}

NetworkEdges <- NetworkNodes %>%
  filter(type != 1) %>%
  select(from = parent_id, to = id, weight = karma, created_utc) %>%
  rbind(SubredditEdges)

# Creating network

redditNetwork <- graph.data.frame(NetworkEdges,
                                  NetworkNodes %>%
                                    mutate(subreddit = ifelse(type == 1,
subreddit, "")),
                                  directed = T) %>%
  simplify(remove.multiple = F, remove.loops = T)

colors <- c("#557BED", "#F2E11F", "#FA6C23")
V(redditNetwork)$color <- colors[V(redditNetwork)$type] %>%
adjustcolor(alpha.f = 0.66)

max_subreddit_karma <- max((V(redditNetwork)[type==1])$karma)
max_post_karma <- max((V(redditNetwork)[type==2])$karma)
max_comment_karma <- max((V(redditNetwork)[type==3])$karma)

V(redditNetwork)$size <- ifelse(V(redditNetwork)$type == 1,
                                5 + 8 * V(redditNetwork)$karma /
max_subreddit_karma, # subreddit
                                ifelse(V(redditNetwork)$type == 2,
                                       2 + 6 * V(redditNetwork)$karma /
max_post_karma, # post
                                       0.25 + 2.8 * V(redditNetwork)$karma
/ max_comment_karma)) # comment

max_weight <- max(E(redditNetwork)$weight)
E(redditNetwork)$width <- 0.4 + 7 * E(redditNetwork)$weight / max_weight

set.seed(44414693)
plot(redditNetwork,
     layout=layout.lgl,
     vertex.frame.color = NA,
     edge.color = adjustcolor("#FFC3A3", alpha.f = 0.66),
```

```
      edge.arrow.size = 0,
      edge.arrow.width = 0,
      edge.lty = 1,
      # edge.width = 0.6,
      edge.curved = 0.1,
      vertex.label = V(redditNetwork)$subreddit,
      vertex.label.family = "Helvetica",
      vertex.label.font = 2,
      vertex.label.cex= 0.8,
      vertex.label.color = "#474747",
      vertex.label.dist = 0.2,
      vertex.label.degree = -pi/2,
      main = "Network of subreddits, posts, and comments")

legend(x=-1.5, y=-0.6, c("Subreddit", "Post", "Comment"), pch=21,
       col="#ffffff", pt.bg=colors, pt.cex=3, cex=.9, bty="n", ncol=1)
```

Appendix E - When to Submit

E.1: Average Karma by Hour

```
KarmaByHour <-
  Posts %>%
  group_by(hour) %>%
  summarize(karma = mean(karma))


KarmaByHour %>%
  ggplot(aes(x = hour, y = karma, fill = karma)) + geom_bar(stat = "identity") +
  scale_fill_gradient2(low = reddit.blue, mid = "#CA7580", high = reddit.red,
midpoint = max(KarmaByHour$karma)/2) +
  ggtitle("Average Karma of Posts by Hour of Day") + xlab("Hour") + ylab("Average
Karma") +
  theme(axis.text.x = element_text(family = "NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
25),
        axis.title.x = element_text(family = "NimbusSanCond", size = 20),
        axis.title.y = element_text(family = "NimbusSanCond", size = 20))
```

E.2: Average Karma by Hour and Day of Week

```
Posts %>%
  filter(karma >= 500) %>%
  group_by(day_of_week, hour) %>%
  summarize(karma = sum(karma)) %>%
  ggplot(aes(x = day_of_week, y = hour)) +
  geom_point(aes(size = karma, col = karma, alpha = karma)) +
  scale_color_gradient2(low = reddit.blue, mid = "#CA7580", high = reddit.red,
midpoint = max(Ken$karma)/2) +
  # geom_smooth(data = HelperHourTable, aes(x = weekday, y = avg_hour)) +
  ggtitle("What Time Each Week Results in the Most Karma?") + xlab("Day of Week")
+ ylab("Hour of Day") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
20),
        axis.title.x = element_text(family = "NimbusSanCond", size = 15),
        axis.title.y = element_text(family = "NimbusSanCond", size = 15))
```

E.3: Time to Post by Top Users

```
BestUserComments = Comments %>%
  filter(author %in% TopUsers$author)

Words = strsplit(BestUserComments$body, " ")
str(Words)
temp = c()
for (i in 1:length(Words)) {
  for (x in 1:length(Words[[i]])) {
    temp = c(temp, Words[[i]][x])
  }
}
```

```
Words = data.frame(temp) %>%
  filter(!grepl(toIgnore, temp)) %>%
  group_by(temp) %>%
  tally() %>%
  mutate(count = n) %>%
  arrange(desc(n))

Words = Words[-1,]
BestUserComments = BestUserComments %>%
  group_by(day_of_week, hour) %>%
  tally() %>%
  mutate(NumberOfComments = n)

BestUserComments %>%
  ggplot(aes(x = day_of_week, y = hour)) +
  geom_point(aes(size = NumberOfComments, col = NumberOfComments, alpha =
NumberOfComments)) +
  scale_color_gradient2(low = reddit.blue, mid = "#CA7580", high = reddit.red,
midpoint = max(BestUserComments$NumberOfComments)/2) +
  # geom_smooth(data = HelperHourTable, aes(x = weekday, y = avg_hour)) +
  ggtitle("What Time do Top Users Comment?") + xlab("Day of Week") + ylab("Hour
of Day") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
20),
        axis.title.x = element_text(family = "NimbusSanCond", size = 15),
        axis.title.y = element_text(family = "NimbusSanCond", size = 15))
```

E.4: Time to Comment by Top Users

```
BestUserPosts = Posts %>%
  filter(author %in% TopUsers$author)

Words = strsplit(BestUserPosts$body, " ")
str(Words)
temp = c()
for (i in 1:length(Words)) {
  for (x in 1:length(Words[[i]])) {
    temp = c(temp, Words[[i]][x])
  }
}

Words = data.frame(temp) %>%
  filter(!grepl(toIgnore, temp)) %>%
  group_by(temp) %>%
  tally() %>%
  mutate(count = n) %>%
  arrange(desc(n))

Words = Words[-1,]
BestUserComments = BestUserComments %>%
  group_by(day_of_week, hour) %>%
  tally() %>%
  mutate(NumberOfComments = n)

BestUserComments %>%
  ggplot(aes(x = day_of_week, y = hour)) +
  geom_point(aes(size = NumberOfComments, col = NumberOfComments, alpha =
NumberOfComments)) +
  scale_color_gradient2(low = reddit.blue, mid = "#CA7580", high = reddit.red,
midpoint = max(BestUserComments$NumberOfComments)/2) +
  # geom_smooth(data = HelperHourTable, aes(x = weekday, y = avg_hour)) +
```

```
  ggtitle("What Time do Top Users Comment?") + xlab("Day of Week") + ylab("Hour
of Day") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
20),
        axis.title.x = element_text(family = "NimbusSanCond", size = 15),
        axis.title.y = element_text(family = "NimbusSanCond", size = 15))
```

Appendix F - What to Submit

F.1: Frequently Used Words in Posts

```
Words %>%
  head(25) %>%
  transform(temp = reorder(temp, order(count, decreasing = TRUE))) %>%
  ggplot(aes(x = temp, y = count, fill = count)) +
  geom_bar(stat = "identity") +
  ggtitle("25 Most Frequently Used Words in Comments") + xlab("Word") +
ylab("Frequency") +
  scale_fill_gradient2(low = reddit.blue, mid = "#CA7580", high = reddit.red,
midpoint = max(Words$n)/2) +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
25),
        axis.title.x = element_text(family = "NimbusSanCond", size = 20),
        axis.title.y = element_text(family = "NimbusSanCond", size = 20))
```

F.2: Frequently Used Words in Comments

```
Words %>%
  head(25) %>%
  transform(temp = reorder(temp, order(count, decreasing = TRUE))) %>%
  ggplot(aes(x = temp, y = count, fill = count)) +
  geom_bar(stat = "identity") +
  ggtitle("25 Most Frequently Used Words in Comments") + xlab("Word") +
ylab("Frequency") +
  scale_fill_gradient2(low = reddit.blue, mid = "#CA7580", high = reddit.red,
midpoint = max(Words$n)/2) +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
25),
        axis.title.x = element_text(family = "NimbusSanCond", size = 20),
        axis.title.y = element_text(family = "NimbusSanCond", size = 20))
```

F.3: Karma Earned vs. Length of Post Title

```
PostTitles <-
  Ken %>%
  mutate(len_title = nchar(title))

PostTitles %>%
  ggplot(aes(x = len_title, y = karma)) + geom_point(aes(col = gilded, alpha =
karma)) +
  ggtitle("Karma Earned vs. Length of Post Title") + xlab("Length of Title") +
ylab("Karma") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
```

```
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
20),
        axis.title.x = element_text(family = "NimbusSanCond", size = 15),
        axis.title.y = element_text(family = "NimbusSanCond", size = 15))
```

F.4: Karma Density in Posts by Punctuation

```
puncuation = "([:upper:])"

PuncuationCheckPosts <- Posts %>%
  extractMatches(puncuation, title) %>%
  mutate(hasPuncuation = as.factor(!is.na(match1)))

levels(PuncuationCheckPosts$hasPuncuation) <- c("No Punctuation", "Contains
Punctuation")
head(PuncuationCheckPosts)

PuncuationCheckPosts %>%
  ggplot(aes(x = log(karma), group = hasPuncuation)) +
  geom_density(fill = reddit.red, col = reddit.red, alpha = 0.6) +
  facet_wrap(~hasPuncuation) +
  ggtitle("Karma Density in Posts") + xlab("Log of Karma") + ylab("% of Posts") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
25),
        axis.title.x = element_text(family = "NimbusSanCond", size = 20),
        axis.title.y = element_text(family = "NimbusSanCond", size = 20))

averageKarmaPuncPosts = PuncuationCheckPosts %>%
  group_by(hasPuncuation) %>%
  summarize(average = mean(karma), med = median(karma) * 1.0)
```

```
PuncuationCheckPosts %>%
  group_by(hasPuncuation) %>%
  tally() %>%
  head()
```

F.5: Karma Density in Posts by Phrase "TIL"

```
##look at "Here is"
puncuation = "(TIL)"

PhraseCheckPosts <- Posts %>%
  extractMatches(puncuation, title) %>%
  mutate(hasPuncuation = as.factor(!is.na(match1)))

levels(PhraseCheckPosts$hasPuncuation) <- c("Does not include TIL", "Includes
TIL")

PhraseCheckPosts %>%
  ggplot(aes(x = log(karma), group = hasPuncuation)) +
  geom_density(fill = reddit.red, col = reddit.red, alpha = 0.6) +
  facet_wrap(~hasPuncuation) +
  ggtitle("Karma Density in Posts") + xlab("Log of Karma") + ylab("% of Posts") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.5, family =
"NimbusSanCond", size = 15),
        axis.text.y = element_text(family = "NimbusSanCond", size = 15),
        panel.background = element_blank(),
        plot.title = element_text(family = "NimbusSanCond", face = "bold", size =
25),
        axis.title.x = element_text(family = "NimbusSanCond", size = 20),
        axis.title.y = element_text(family = "NimbusSanCond", size = 20))

averageKarmaPhrasePosts = PhraseCheckPosts %>%
  group_by(hasPuncuation) %>%
```

```
  summarize(average = mean(karma), med = median(karma))

PhraseCheckPosts %>%
  group_by(hasPuncuation) %>%
  tally() %>%
  head()
```