DS110 Project 2 - Facebook Online Sales

Nathan Gonyo, Paul Ritter, and Elliot Schendel
Professor Rasitha Jayasekare

DS 110
12/04/2020

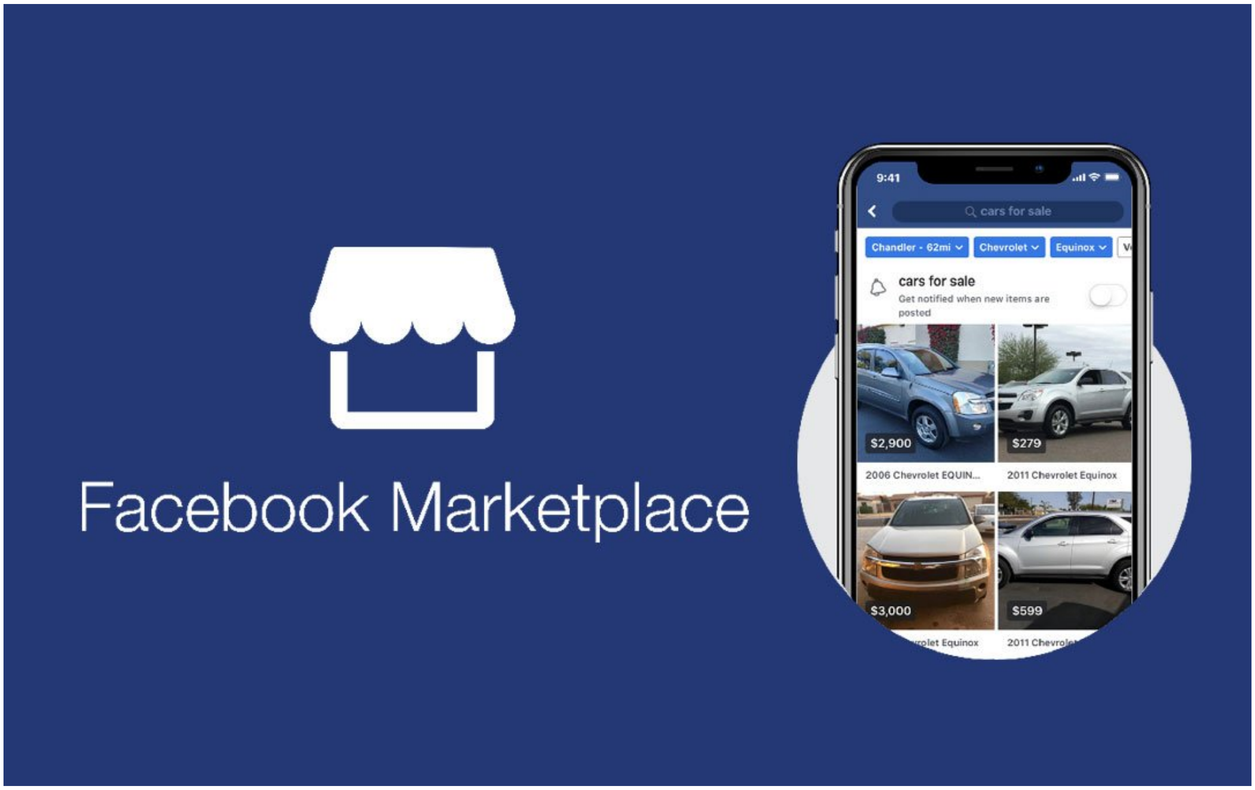
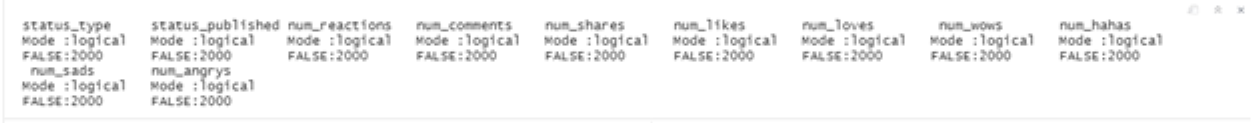


Table of Contents

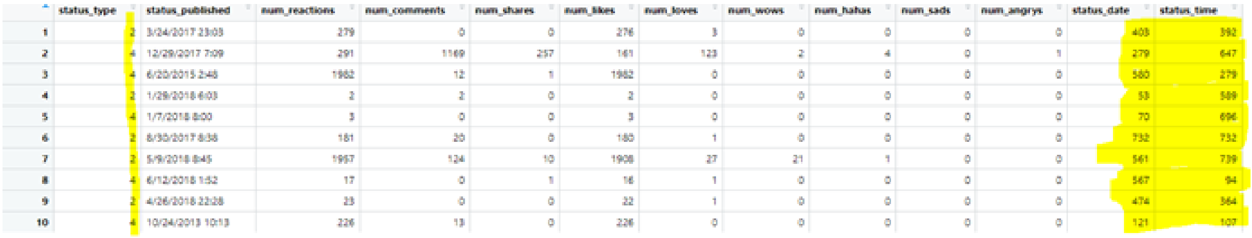
Preprocessing

Before preprocessing occurred, our team had to take a random sample of 2000 records from the 'FBOnlineSalesData.txt' file's 7050 records. Once completed, our team was able to begin preprocessing. The first step of preprocessing we took was identifying any and all rows whose records contained missing values. This process began with using a unique command in R known as 'is.na', which identifies rows where missing values are present and identifying them for our team in a new data table we called 'fb2000_na'. If a record contained a value that wasn't there, it would be identified as 'TRUE' (otherwise it would appear as 'FALSE' for not being a missing value). After completing this step, we identified zero records within our random sample which contained any records with missing values. Seen below is R output which shows how many records contained missing values based on each variable. Again, 'FALSE' indicates records which do not have any missing values…



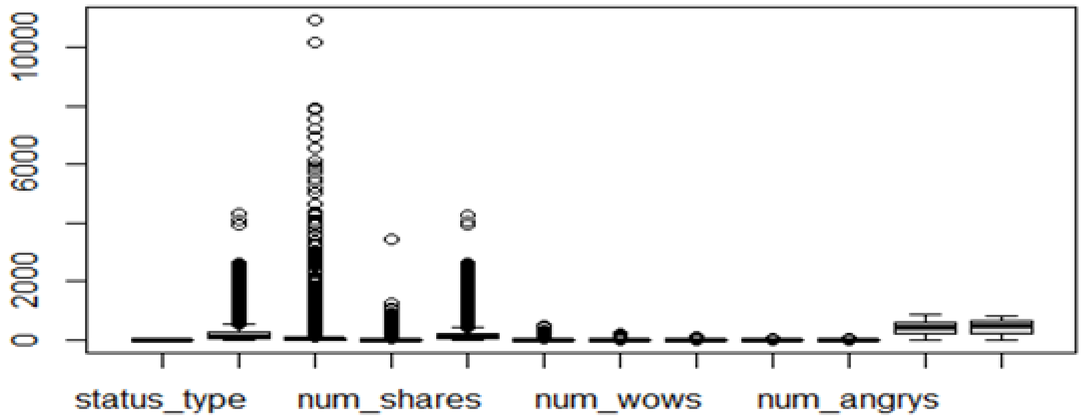
```
 status_type     status_published num_reactions   num_comments    num_shares      num_likes       num_loves        num_wows        num_hahas
 Mode :logical   Mode :logical    Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:2000      FALSE:2000       FALSE:2000      FALSE:2000      FALSE:2000      FALSE:2000      FALSE:2000      FALSE:2000      FALSE:2000
  num_sads       num_angrys
 Mode :logical   Mode :logical
 FALSE:2000      FALSE:2000
```

Once completed, we moved onto recoding several variables in our data set for the purposes of cluster analysis in this project. Firstly, we separated the 'status_published' variable into 'status_date' and 'status_time'. Originally, values within the original variable would be seen in the following format: "3/24/2017 23:03". These two new variables would split this value separately based on our new variables ("3/24/2017" would be the new value for 'status_date' and "23:03" would be the new variable for 'status_time'). We further changed these new variables into numeric variables; this was done so that we could use only factor or numeric variables for cluster analysis (avoiding the use of 'char' variables when possible). We also converted the 'status_type' variable into a factor variable for the same previously stated above; as such we changed "link", "photo", "status", and "video" to be coded as "1", "2", "3" and "4" respectively. Seen below is an example of the records in our sample dataset which reflect these changes…

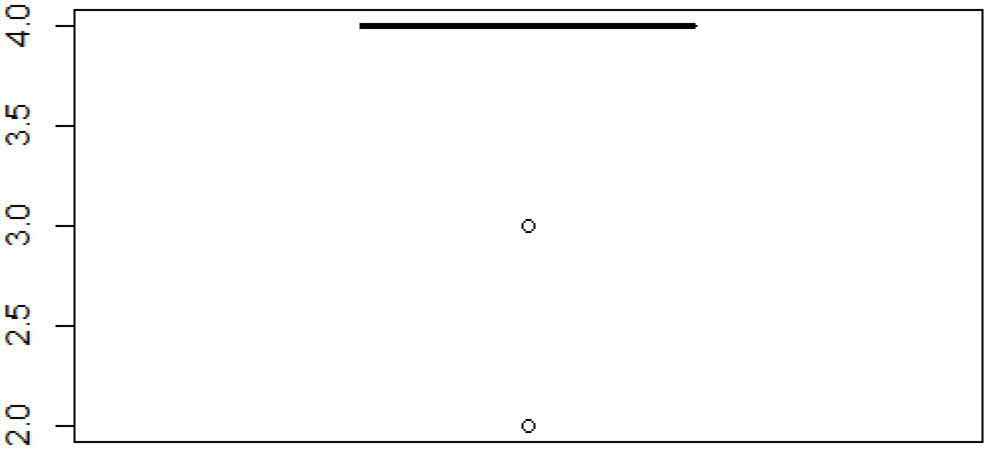


	status_type	status_published	num_reactions	num_comments	num_shares	num_likes	num_loves	num_wows	num_hahas	num_sads	num_angrys	status_date	status_time
1	2	3/24/2017 23:03	279	0	0	276	3	0	0	0	0	403	392
2	4	12/29/2017 7:09	291	1169	257	161	123	2	4	0	1	279	647
3	4	6/20/2015 2:48	1982	12	1	1982	0	0	0	0	0	580	279
4	2	1/29/2018 6:03	2	2	0	2	0	0	0	0	0	53	589
5	4	1/7/2018 8:00	3	0	0	3	0	0	0	0	0	70	696
6	2	8/30/2017 8:38	181	20	0	180	1	0	0	0	0	732	732
7	2	5/9/2018 8:45	1957	124	10	1908	27	21	1	0	0	561	739
8	4	6/12/2018 1:52	17	0	1	16	1	0	0	0	0	567	94
9	2	4/26/2018 22:28	23	0	0	22	1	0	0	0	0	474	364
10	4	10/24/2013 10:13	226	13	0	226	0	0	0	0	0	121	107

The last step we undertook for preprocessing was identifying outliers. It is important to mention that our team was not directed to remove the outliers but to merely identify them and report on them as they appeared. We used a boxplot to visualize the outliers present in our dataset based on each variable. Seen below is the boxplot which visualizes outlier values based on each variable…



Our team identified 405 records in our sample which contained outlier values (representing 20.25% of all sampled records). As a result, our team was able to get a couple of interesting insights about the outliers that are present in our sample. One of the most astonishing insights we found was that there was no outlier record which were of the ‘link’ status type; 23 of our outlier records were ‘photo’s, 11 were ‘status’ and those remaining were ‘videos’ See below the boxplot depicting the outlier records based on the ‘status_type’ variable…



Another interesting insight we found is regarding ‘num_comments’, the variable which contained the most volatile values for all 405 outlier records. We found that each record had on average around 1102 comments; one record has a staggering 10,960 comments. The 1st quartile, median and 3rd quartiles were 253, 579 and 1307 comments, respectively. We gleaned that there is a large skew on the number of comments beyond the 3rd quartile. See below the boxplot for outlier records based on the 'num_comments' variable…

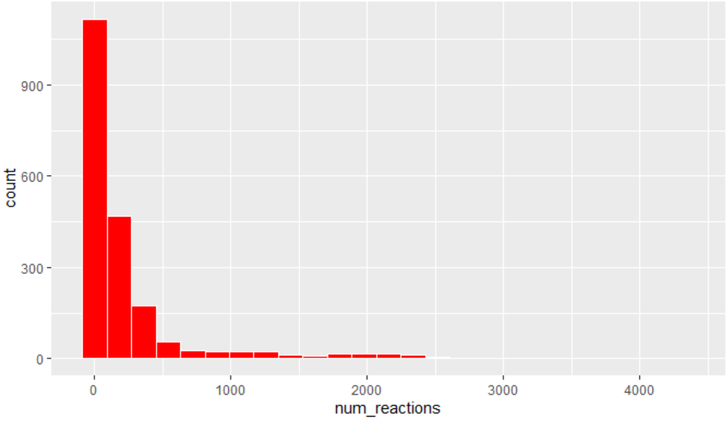
Numerical & Visual Descriptions of Dataset

After we completed analyzing the presence of outliers in the data, we were done with the majority of preprocessing before moving on to performing cluster analysis. Before that however, we need to describe the variables of the dataset both numerically/statistically, as well as visually. It is important to note that because we were instructed not to remove the outliers we found, they would remain in this portion of the preprocessing, which could affect the way the variables are distributed.To start with the data description, we Provided Numerical Summaries of each of the Variables to get an idea of how they are distributed, including mean, max, median, etc.

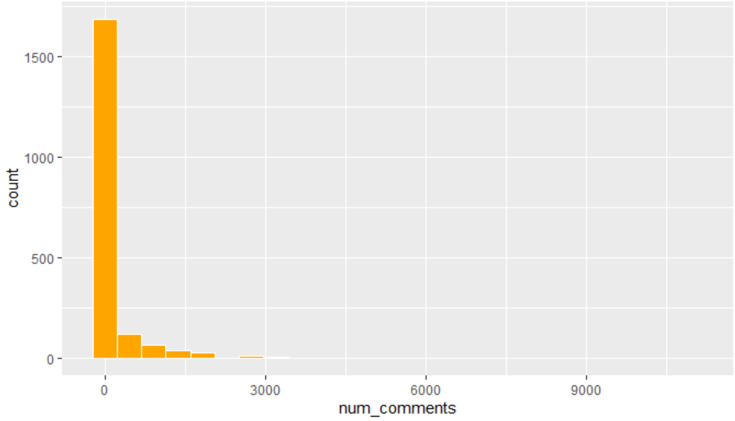
	Numerical Descriptions of Variables			
	Min	**Median**	**Mean**	**Max**
Reactions	0	65	223.8	4315
Comments	0	5	228.4	10960
Shares	0	0	41.58	3424
Likes	0	62	218	4241
Loves	0	0	13.34	482
Wows	0	0	1.41	200
Hahas	0	0	.673	76
Sads	0	0	.303	51
Angrys	0	0	.109	8

After we completed the numerical descriptions of each of the variables, we moved on to visualizing the data. To start with data visualization, we visualized each of the Variables through a simple histogram in order to get an idea of how they are distributed. After this we will provide multivariate plots to compare variables, especially focusing on how each of the variables were affected by the form of post / type.

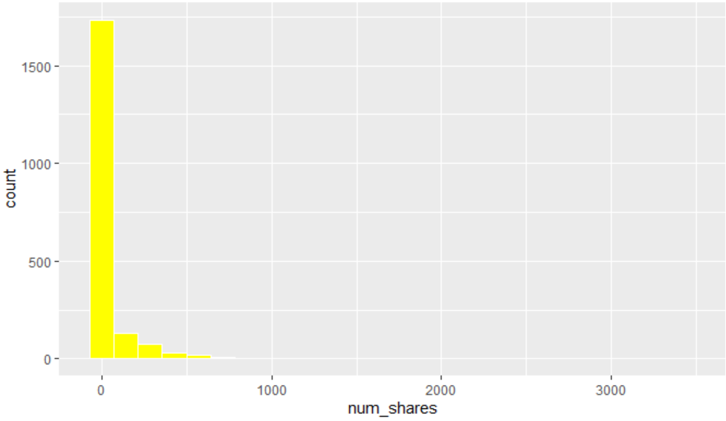
Distribution of Number of Reactions



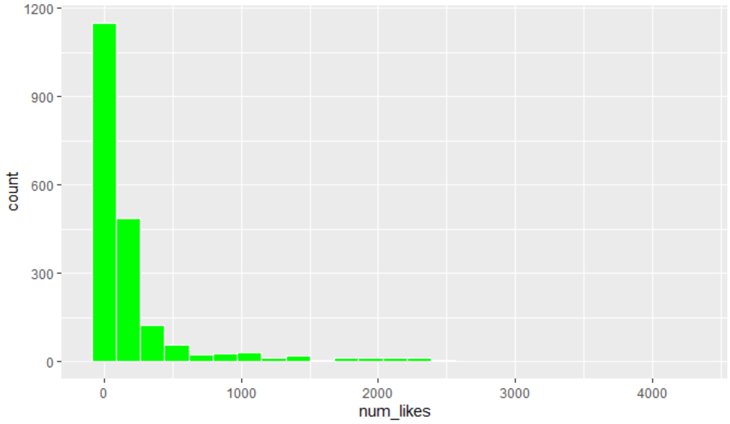
Distribution of Number of Comments



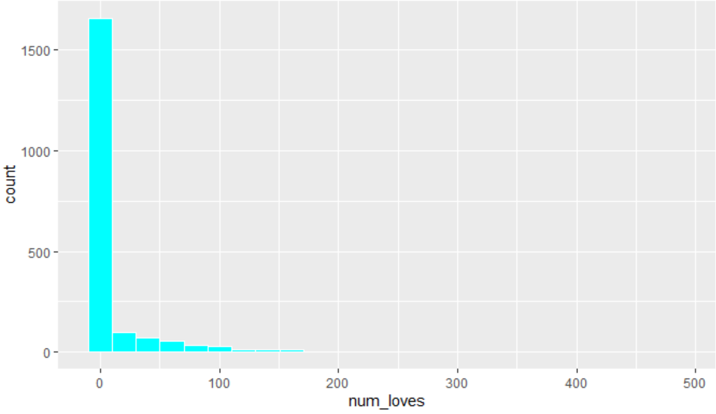
Distribution of Number of Shares



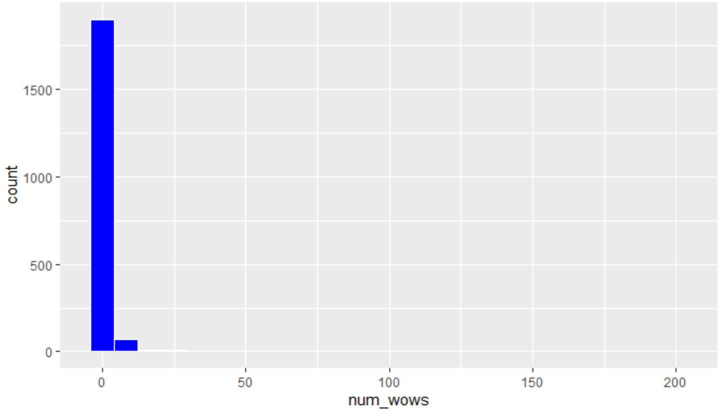
Distribution of Number of Likes



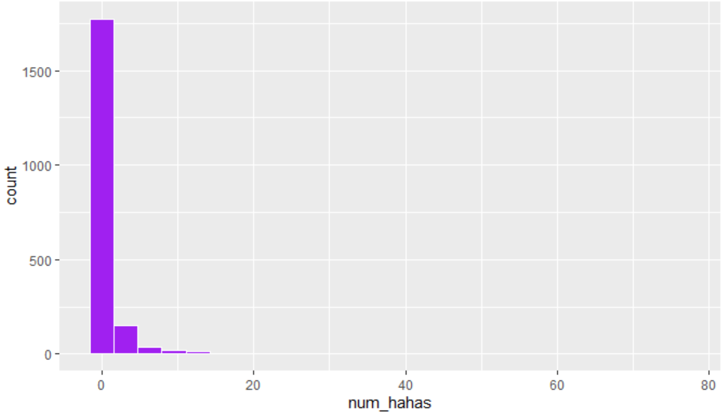
Distribution of Number of Loves



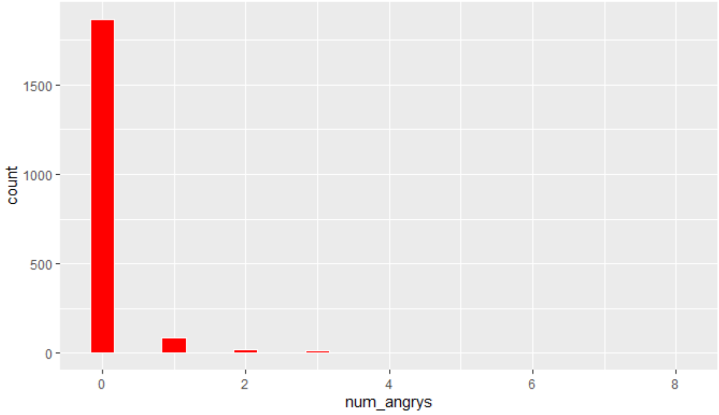
Distribution of Number of Wows



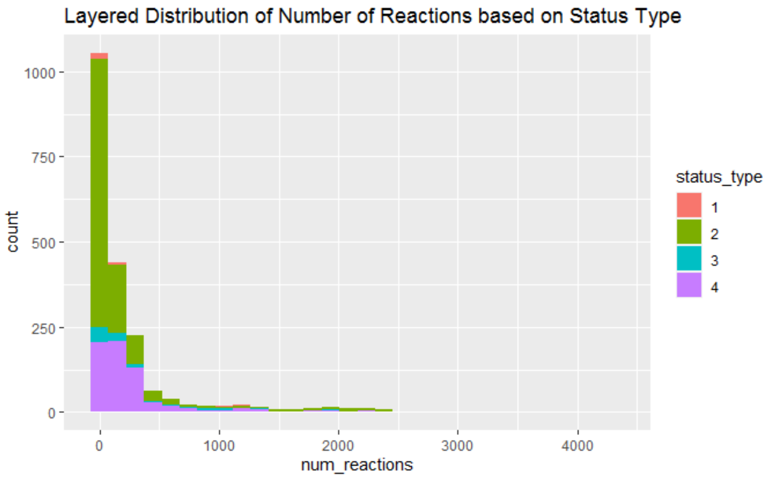
Distribution of Number of Hahas

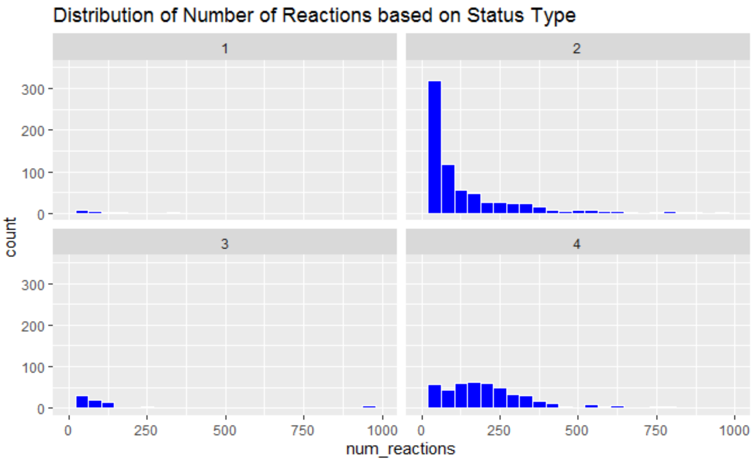


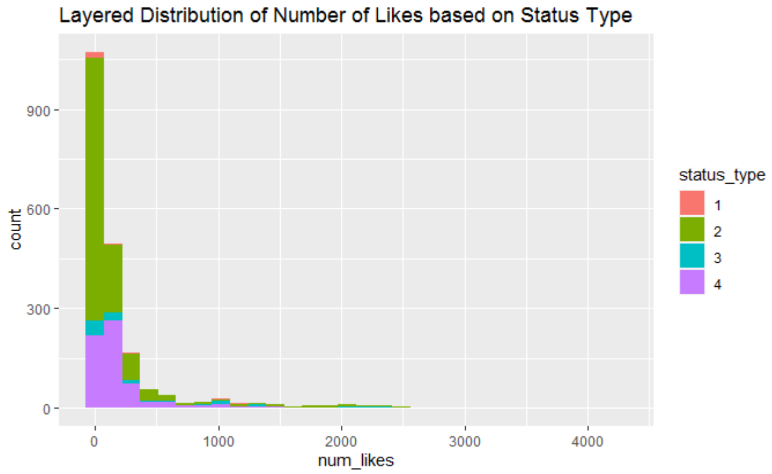
Distribution of Number of Angrys

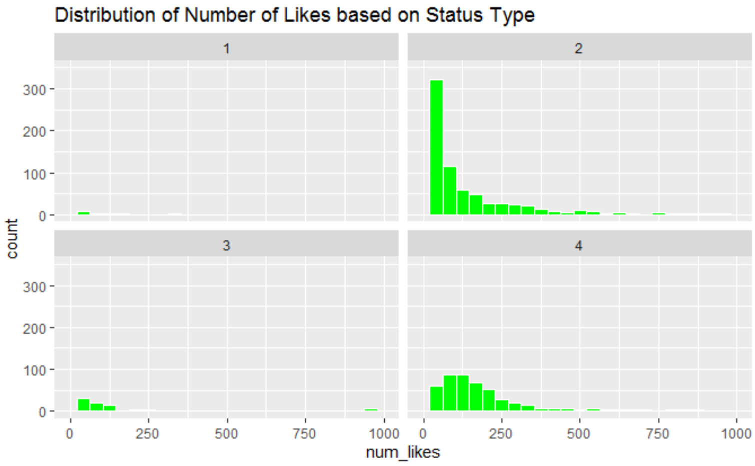


After Creating a simple univariate histogram of each of the variables, we could clearly see that each variable followed a right / positive skewed distribution, with the vast majority of observations close to the y axis. We can also see that there are variables which are more spread out than others, these being the variables with the most observations, resulting in generally more outliers. After these univariate plots were created, we went on to examine how variables shifted depending on the status type of the post. Keep in mind that we converted status type to be numerical, with numbers from 1-4 representing "link", "photo", "status", and "video" respectively.

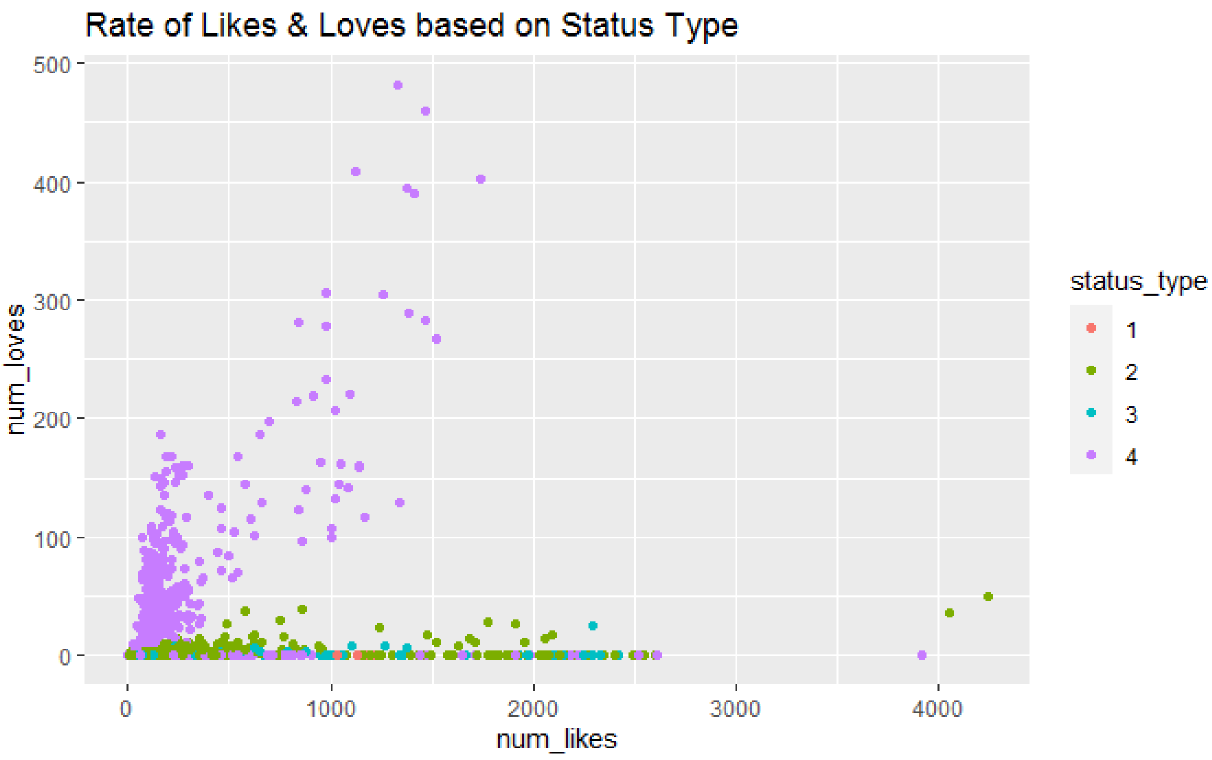

Layered Distribution of Number of Reactions based on Status Type


Distribution of Number of Reactions based on Status Type


Layered Distribution of Number of Likes based on Status Type


Distribution of Number of Likes based on Status Type

The above two sets of plots were created to illustrate how the frequency of reactions & likes were based on the status types mentioned above. From these plots, it becomes clear that status types of photo and video are much more likely to receive reactions (status_type 2 & 4) than posts with links or a simple status (status_type 1 & 3). This makes sense as media such as photos and videos are generally more likely to draw attention from users on social media, as opposed to a simple post containing text.

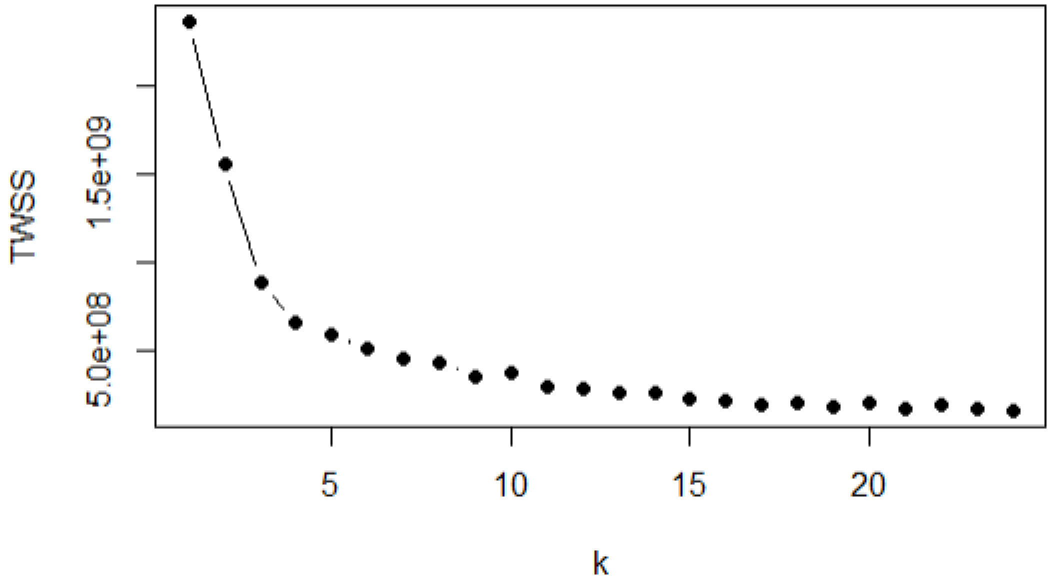


The above plot further illustrates the positive correlation between posts containing photo and video (status_type 2 & 4) and rates of interaction from users. This plot is looking at the variables num_loves and num_likes, which are different variables from the sets of plots above, but hold the same relationship. This plot also illustrates the extent to which video (purple, 4) outperforms the other forms of posts, even photos (green, 2), as evident by how high the video's number of "loves" is on the y axis.

Now that we have an idea of how each of the variables in the dataset are distributed, as well as how different forms of statuses/posts affect the rates of engagement, we can begin to perform cluster analysis of the data. This was the final step in preprocessing.

Applying K Means Clustering

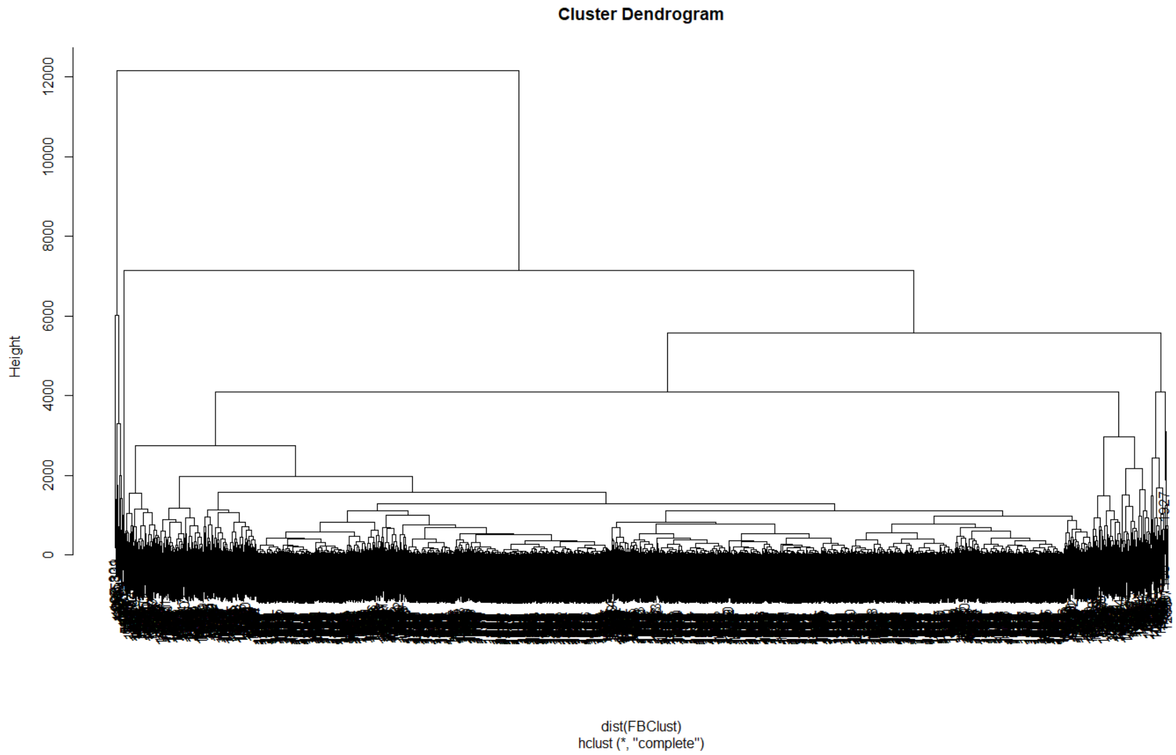
To perform K means clustering we first needed to determine the best number of clusters. To determine this we created a graph that plots the Total within sum of squares value (TWSS) vs the number of clusters. Below is the TWSS graph.



As you can see the TWSS doesn't experience a significant drop after k=4, so we will use 4 clusters. After performing the K-Means clustering algorithm our 1st, 2nd, 3rd, and 4th clusters had 42, 139, 722, 1047 data points respectively. We will explain these clusters in more detail later.

Applying Hierarchical Clustering

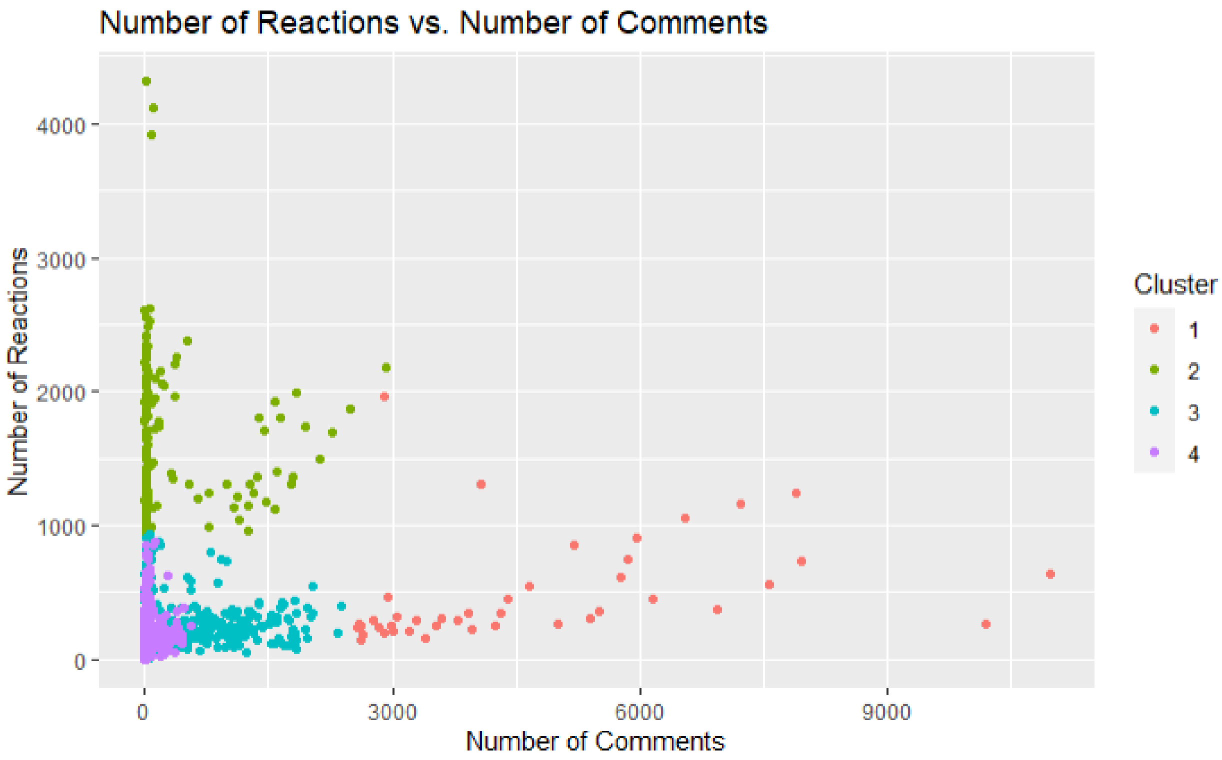
Similar to K Means to perform Hierarchical clustering we also need to determine the optimal number of clusters. To do this we will need to look at the dendrogram below.



When looking at this dendrogram we see a set of two clear tall branches between 12,000 and 7,000. Therefore we will cut between those, giving us two clusters. The 1st and 2nd clusters had 1984, and 16 data points respectively.

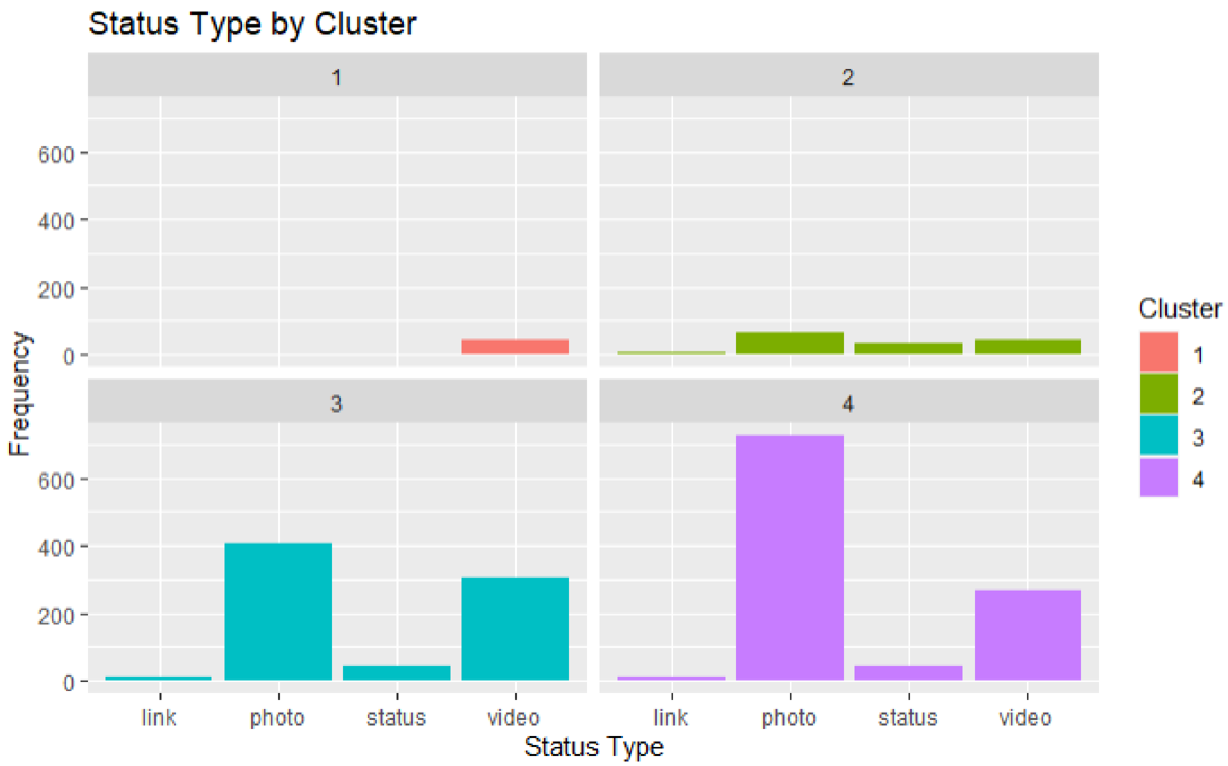
Post Analysis

Since the hierarchical clustering had only 2 clusters and one of them only contained 16 data points, we decided to do our post-analysis on the K-means clustering. K-means had 4 different clusters and when comparing the clusters amongst each other we found that the number of comments vs the number of reactions seemed to be a good indicator of what cluster a datapoint would be assigned. The graph and table below show that cluster 1 contains data points with more comments than reactions. Cluster 2 on the other hand seems to have more reactions than comments. Cluster 3 has about an even number of reactions and comments, while cluster also has an even amount, but less than cluster 4



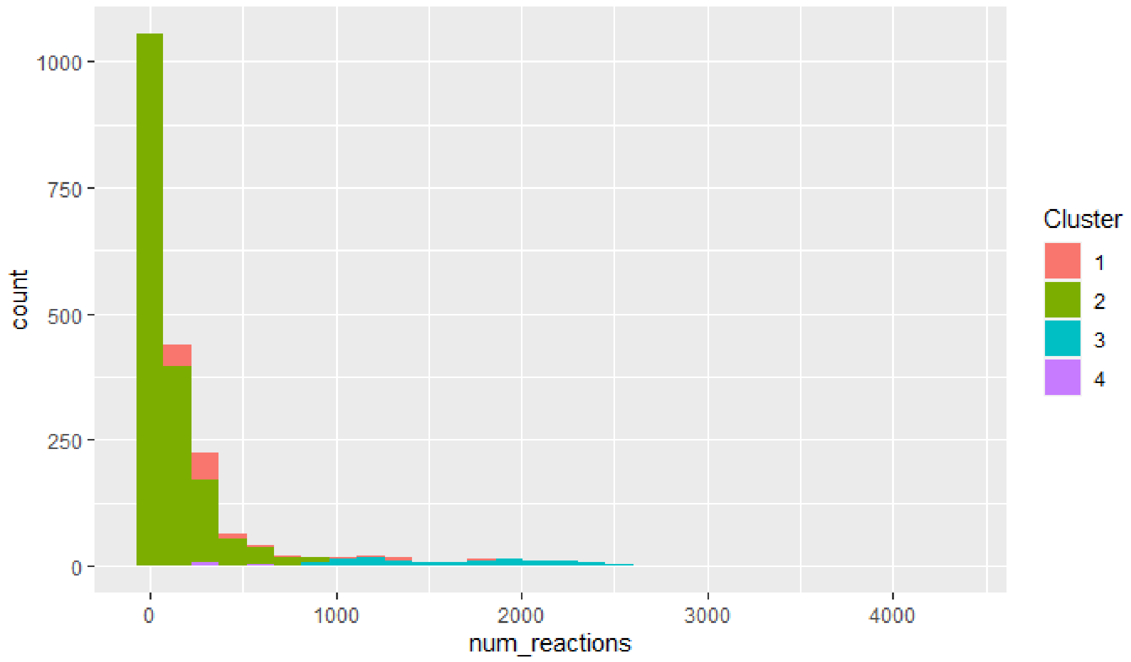
	Average Number of Comments	Average Number of Reactions
Cluster 1	4,686	488.7
Cluster 2	340	1,677
Cluster 3	245.7	148.9
Cluster 4	21.99	94.58

Next we decided to take a look at the status_types contained in each cluster. In the below graph we can see the distribution of status type for each cluster.

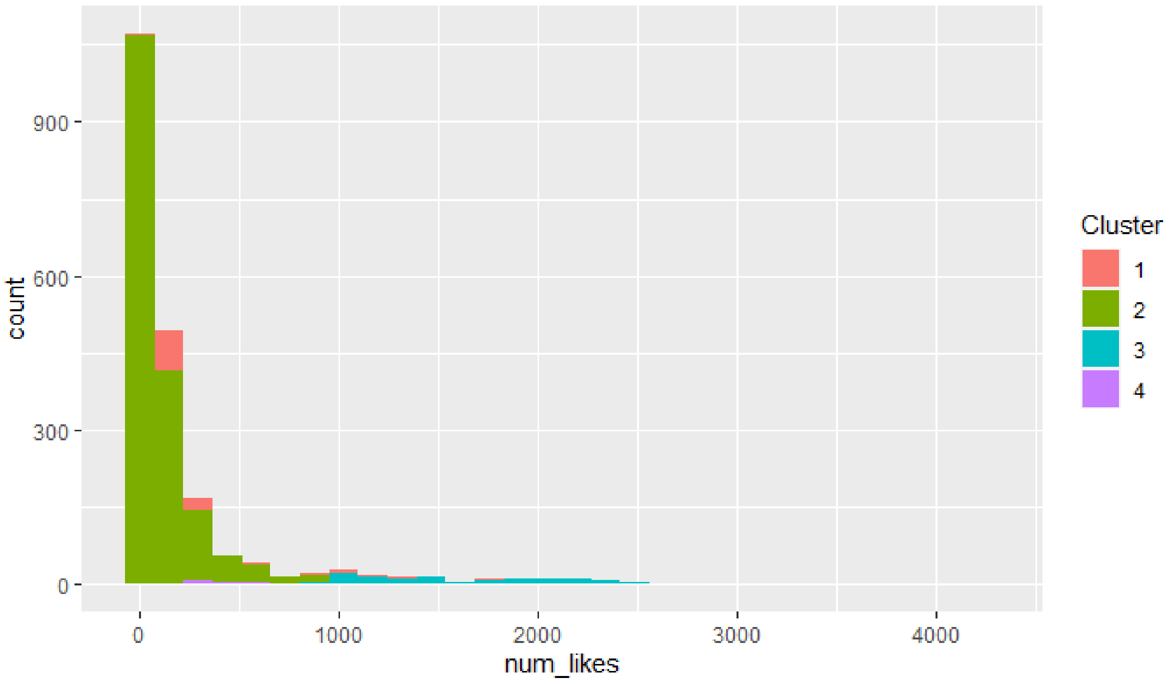


As you can see cluster 1 contains almost all videos, while the other 3 clusters each have a similar distribution of status types, photo being the highest, followed by videos, then status, and finally link.

Next, we decided to look closer at reactions and likes for each cluster. The following two graph are a layered histogram showing the reactions and likes by cluster.

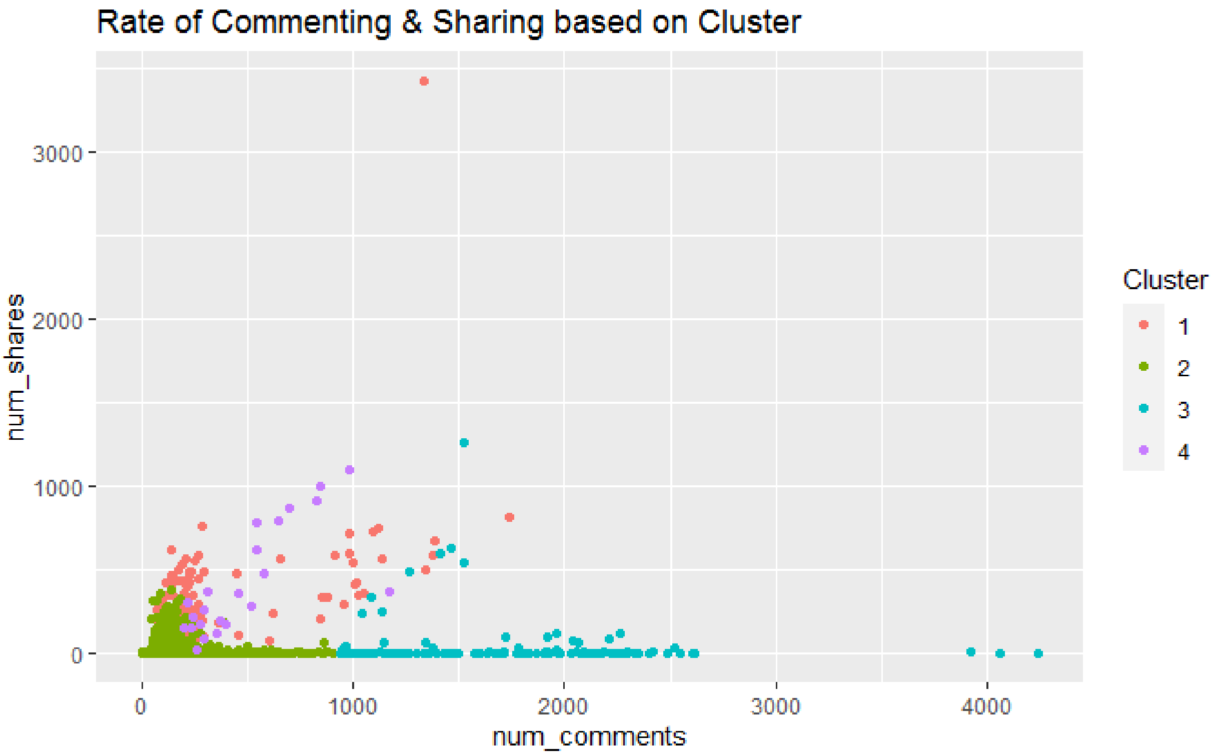


Layered Distrubition of Number of Reactions based on Cluster

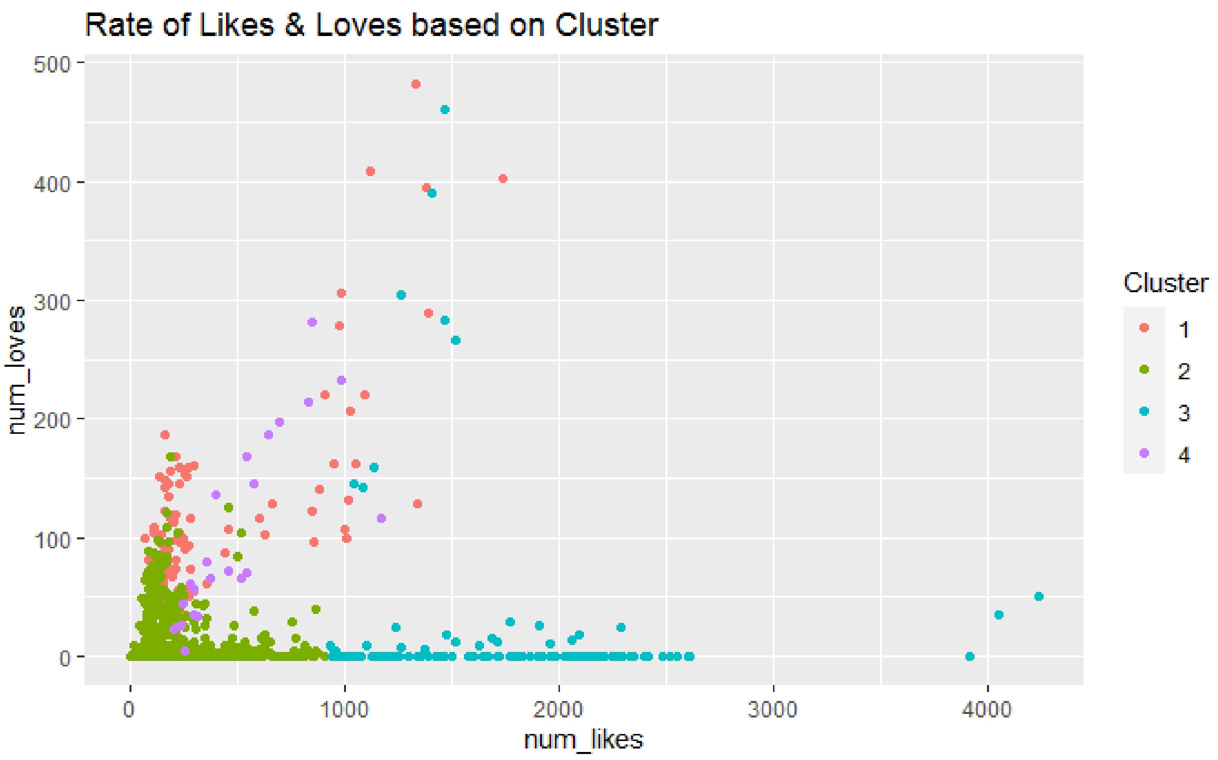


Layered Distrubition of Number of Likes based on Cluster

From the above two layered histograms, we are able to see how different observations were distributed to clusters according to the number of reactions posts received (top plot), as well as the number of likes (bottom plot). It is evident that generally posts with a lower number of reactions & likes, which were the vast majority of posts, were assigned with cluster 2 most frequently. We also can see that as the levels of engagement increased, posts were more likely to be assigned to cluster 3 (blue). This relationship is further illustrated by the following two plots, which illustrates how clusters varied depending on the number of shares & comments (top plot) as well as the number of likes and loves (bottom plot).



In this plot, it seems cluster 3 generally had high levels of comments, while cluster 2 had lower frequencies of comments. Cluster 4 was a smaller number of observations, but generally had high levels of engagement. Cluster 1 was similar to cluster 4, except observations tended to be more broadly spread over the plot, with cluster 4 more closely spread out.



This plot compared the number of likes and loves across each of the clusters. Cluster 3 and 2 were distributed similarly to the above graph, with cluster 3 being centered around high levels of x (likes) and low levels of y (loves) Cluster 2 was centered near the origin, as in the graph above. Clusters 1 and 4 were generally more broadly spread out compared to clusters 2&3, which is what we observed in the commenting vs sharing plot. Generally, the visualization of these clusters allowed us to gain a much better understanding of how clusters were related to each other, and how observations that shared a cluster had characteristics in common. One thing to consider is the possibility that clusters 1 and 4 would have had even fewer observations present if we were to have removed outliers, rather than keeping them in the dataset after preprocessing. This would make sense as clusters 1 and 4 seemed to be much more spread out from the other observations, as shown in the two graphs above.