

**Assignment 3**

**2015 GOP Debate Tweets Sentiment Analysis**

Theodore Fitch

Department of Data Analytics, University of Maryland Global Campus

DATA 650: Big Data Analytics

Dr. Ozan Ozcan

July 16<sup>th</sup>, Summer 2024

## **Introduction:**

Sentiment analysis, a subfield of natural language processing (NLP), is an essential tool in understanding public opinion and emotions expressed in text data, such as social media posts, product reviews, or news articles (Medhat et al., 2014). This text data is ubiquitous around us nowadays whether expressed by Google/Amazon reviews, Tweets, or surveys. By leveraging sentiment analysis, we can categorize text into positive, negative, or neutral sentiments. This offers valuable insights into how people perceive events, brands, or public figures. For instance, businesses can use sentiment analysis to gauge customer satisfaction from reviews, while political analysts can assess public opinion from social media posts during an election campaign. This powerful technique transforms unstructured text into actionable insights, allowing organizations to make data-driven decisions based on the sentiments expressed by their audience.

The first Republican Primary debate, which took place in August of 2015, was a pivotal event in the United States political landscape (Cornfield, 2018). It marked a significant moment in the race for the Republican nomination for the presidential election. GOP stands for the Grand Old Party, another name for the Republican Party. This debate was the first in a series of many where prospective candidates for president discussed their policies, defended their records, and aimed to win voter support. It was a hectic and highly charged time, characterized by intense media scrutiny and public interest (Cornfield, 2018). The inclusion of Donald Trump, a high-profile businessman, television personality, and political outsider added an unprecedented level of attention and a unique dynamic to the debates. His candidacy challenged traditional political norms and brought significant controversy and media coverage. This made the debates

particularly significant and a rich source of data for sentiment analysis, as public reactions and opinions were sharply divided and highly vocal across social media platforms.

Thus, applying sentiment analysis to the context of the first 2015 GOP debate provides a fascinating glimpse into public opinion during a critical political event. By analyzing tweets related to the debate, we can discern how different candidates were perceived by the public and identify the key issues that resonated with viewers. The real-time and widespread nature of Twitter makes it an invaluable resource for capturing the immediate reactions and sentiments of a diverse audience. In this analysis, the aim is to uncover patterns and trends in the sentiment expressed by Twitter users, shedding light on the overall reception of the debate and its participants. This approach not only offers insights into the political landscape of that period but may be applicable to the landscape today. It also demonstrates the practical application of sentiment analysis in understanding and interpreting large volumes of social media data.

In the era of big data, the ability to analyze vast amounts of information efficiently is crucial for deriving meaningful insights. For our sentiment analysis of tweets related to the 2015 GOP debate, we utilized DB2 on the IBM Cloud environment, a robust database management system designed for handling large-scale data operations, to store and query our dataset (IBM, 2024a). By employing SQL (Structured Query Language) within DB2, we can ensure the data was loaded correctly, perform exploratory data analysis (EDA) to understand the structure, identify key characteristics, and clean the dataset. Once the dataset is explored, we can turn to IBM Watson Studio's RStudio, an integrated development environment (IDE) for R, which offers powerful statistical and graphical capabilities (IBM, 2024b). Using RStudio, we can perform sentiment analysis to classify and interpret the sentiments expressed in the tweets,

leveraging its extensive libraries and tools tailored for text mining and natural language processing. This combination of DB2 for data management and RStudio for analysis allows us to transform raw data into actionable insights, demonstrating the effectiveness of these technologies in extracting valuable information from large datasets.

### **Method - Exploratory Data Analysis (EDA):**

Prior to performing any analysis, the data must be fully understood. The dataset was provided by the classroom as a CSV file, along with a word file containing a key for the dataset:

A	ID	Row ID
B	CANDIDATE	Candidate mentioned
C	CANDIDATE_CONFIDENCE	Confidence of the candidate mentioned
D	RELEVANT_YN	"no" means that the tweet was meant to be part of the dataset but was not available when contributors went to judge it
E	RELEVANT_YN_CONFIDENCE	confidence in the existence/non-existence of the tweet
F	SENTIMENT	Tweet Sentiment
G	SENTIMENT_CONFIDENCE	Confidence of the sentiment
H	SUBJECT_MATTER	Tweet subject
I	SUBJECT_MATTER_CONFIDENCE	Confidence of the subject matter
J	CANDIDATE_GOLD	whether the candidate was included in the gold standard for the model
K	NAME	the user who tweeted
L	RELEVANT_YN_GOLD	whether the tweet yn value is golden
M	RETWEET_COUNT	number of times the user has retweeted
N	SENTIMENT_GOLD	if the profile is golden, what is the sentiment
O	SUBJECT_MATTER_GOLD	whether the subject matter was included in the gold standard for the model
P	TEXT	the text of the tweet
Q	TWEET_COORD	if the user in column K has location turned on, the coordinates as a string with the format "[latitude, longitude]"
R	TWEET_CREATED	When tweet was created
S	TWEET_ID	Tweet identification number
T	TWEET_LOCATION	User's country, city, state
U	USER_TIMEZONE	User time zone

Figure 1. 2015 GOP Debate Tweet dataset variables show 21 variables.

The dataset was uploaded to the DB2 environment:

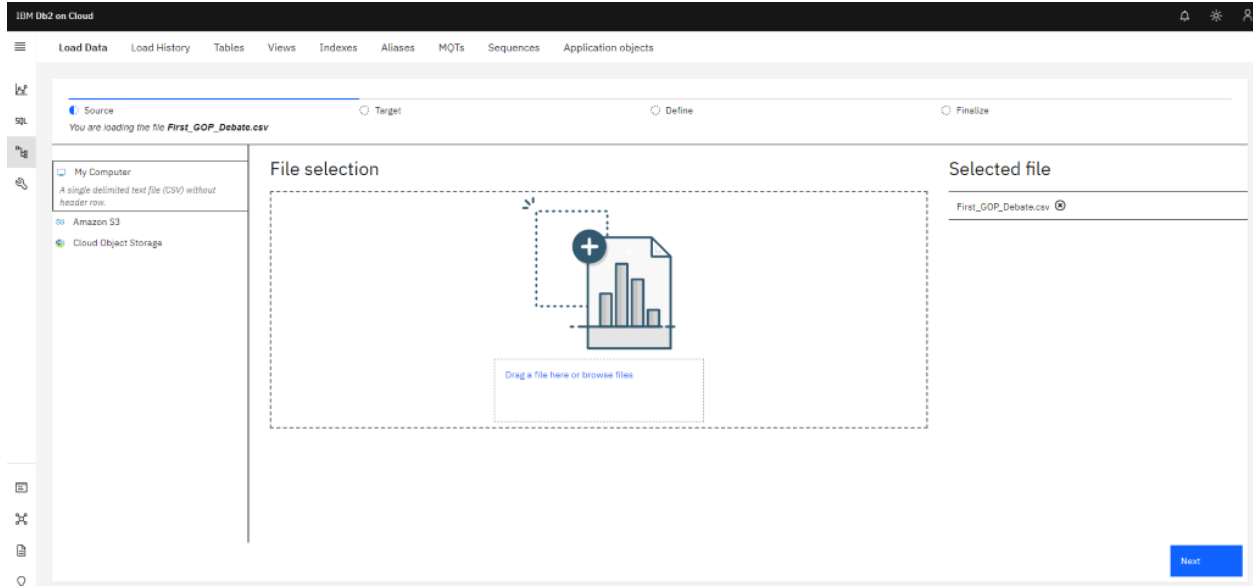


Figure 2. GOP Debate dataset being uploaded to DB2 Cloud environment.

The statement failed because the authorization ID does not have the required authorization or privilege to perform the operation. Authorization ID: "TYG86489", Operation: "EXECUTE", Object: "SYSPROC.MON\_GET\_CONTAINER". [Show logs](#)

ID	CANDIDATE	CANDIDATE_CONFIDENCE	RELEVANT_YN	RELEVANT_YN_CONFIDENCE	SENTIMENT	SENTIMENT_CONFIDENCE	SUBJECT_MATTER
1	No candidate mentioned	1	yes	1	Neutral	0.6578	None of f
2	Scott Walker	1	yes	1	Positive	0.6333	None of f
3	No candidate mentioned	1	yes	1	Neutral	0.6629	None of f
4	No candidate mentioned	1	yes	1	Positive	1	None of f
5	Donald Trump	1	yes	1	Positive	0.7045	None of f
6	Tea Cruz	0.6332	yes	1	Positive	0.6332	None of f
7	No candidate mentioned	1	yes	1	Negative	0.6761	FOX New
8	No candidate mentioned	1	yes	1	Neutral	1	None of f
9	Ben Carson	1	yes	1	Negative	0.6889	None of f
10	No candidate mentioned	0.4594	yes	0.6778	Negative	0.6778	None of f

Figure 3. Once uploaded, the GOP Debate dataset variables length were edited in DB2 Cloud environment.

Column	Number of characters
CANDIDATE_GOLD	22
SENTIMENT_GOLD	16
SUBJECT_MATTER_GOLD	45
TEXT	185
TWEET_COORD	28
TWEET_CREATED	14
TWEET_LOCATION	110
USER_TIMEZONE	28

Figure 4. The GOP Debate dataset variables length were edited in DB2 Cloud environment as per the above character lengths.

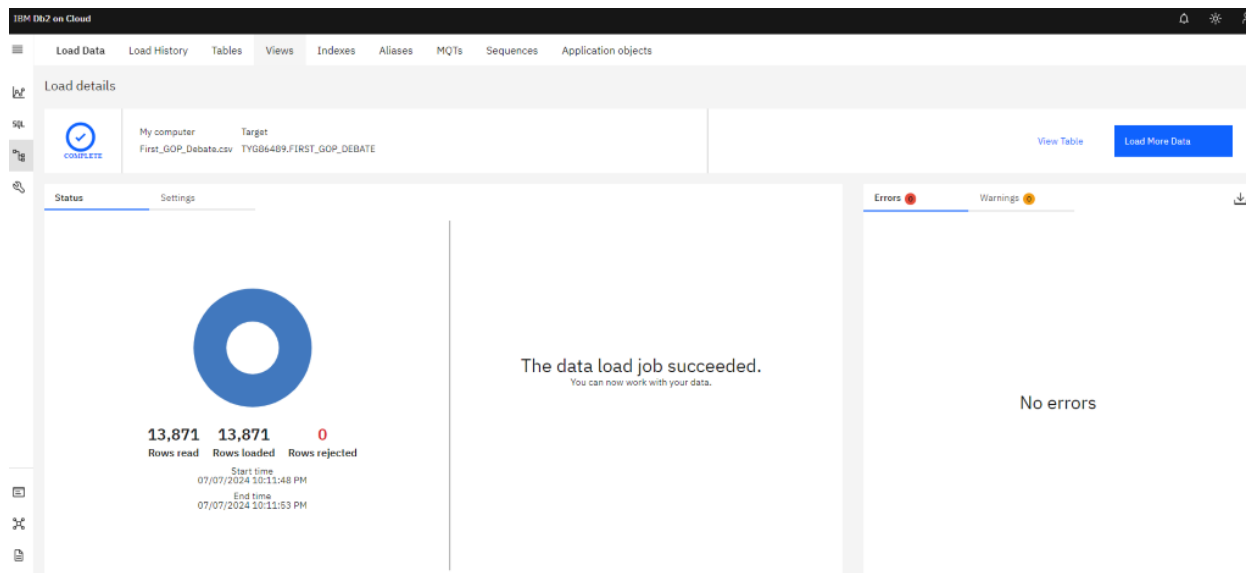


Figure 5. The GOP Debate dataset was successfully loaded to the DB2 Cloud environment with no errors, no warnings, and 13,871 read and loaded (with 0 rows rejected). The start and end load times can be observed as well.

The screenshot shows the first 18 rows of the dataset 'TYG86489.FIRST\_GOP\_DEBATE'. The table has 13 columns: ID, CANDIDATE, CANDIDATE\_CO., RELEVANT\_YN, RELEVANT\_YN\_CO., SENTIMENT, SENTIMENT\_CO., SUBJECT\_MATT, SUBJECT\_MATT\_CO., CANDIDATE\_G., NAME, RELEVANT\_YN\_CO., and RETWEET\_CO. The data includes various candidates like Scott Walker, Donald Trump, Ted Cruz, and Ben Carson, with their respective sentiment scores and tweet counts.

ID	CANDIDATE	CANDIDATE_CO.	RELEVANT_YN	RELEVANT_YN_CO.	SENTIMENT	SENTIMENT_CO.	SUBJECT_MATT	SUBJECT_MATT_CO.	CANDIDATE_G.	NAME	RELEVANT_YN_CO.	RETWEET_CO.
1	No candidate menti	1.0000	yes	1.0000	Neutral	0.6578	None of the above	1.0000		I_Am_Kenzi		5
2	Scott Walker	1.0000	yes	1.0000	Positive	0.6333	None of the above	1.0000		PeacefulQuest		26
3	No candidate menti	1.0000	yes	1.0000	Neutral	0.6629	None of the above	0.6629		PussayCrook		27
4	No candidate menti	1.0000	yes	1.0000	Positive	1.0000	None of the above	0.7039		MattFromTexas31		138
5	Donald Trump	1.0000	yes	1.0000	Positive	0.7045	None of the above	1.0000		sharonDay5		156
6	Ted Cruz	0.6332	yes	1.0000	Positive	0.6332	None of the above	1.0000		DRJohnson11		228
7	No candidate menti	1.0000	yes	1.0000	Negative	0.6761	FOX News or Modern	1.0000		DebWilliams57		17
8	No candidate menti	1.0000	yes	1.0000	Neutral	1.0000	None of the above	1.0000		RaulAReyes		0
9	Ben Carson	1.0000	yes	1.0000	Negative	0.6889	None of the above	0.6444		kengpdx		0
10	No candidate menti	0.4594	yes	0.6778	Negative	0.6778	None of the above	0.4594		Kathielarsyn		1
11	Donald Trump	1.0000	yes	1.0000	Negative	1.0000	None of the above	1.0000		jnjsmom		0
12	Mike Huckabee	1.0000	yes	1.0000	Positive	1.0000	Foreign Policy	0.6520		KLWorster		188
13	No candidate menti	1.0000	yes	1.0000	Negative	0.6957	None of the above	1.0000		wiggerblonde		0
14	Jeb Bush	0.6947	yes	1.0000	Neutral	0.6421	Foreign Policy	1.0000		NCBleThumper		5
15	Scott Walker	1.0000	yes	1.0000	Positive	1.0000	None of the above	1.0000		In_Related_News		215
16	Chris Christie	1.0000	yes	1.0000	Negative	1.0000	None of the above	0.6778		MDiner92		0
17	Donald Trump	0.3923	yes	0.6264	Negative	0.6264	Women's Issues (no	0.3923		gina_catch22gg		45
18	Donald Trump	0.6679	yes	1.0000	Negative	0.6679	FOX News or Modern	1.0000		FDNPT7orn		7

Figure 6. The GOP Debate dataset rows could be read to confirm the dataset appeared to be uploaded correctly and each variable type appears as it should.

One can easily see from the above workflow, the dataset was chosen to be uploaded to a DB2 environment from the CSV file (Figure 2). Then, a few of the variable's character lengths were adjusted (per classroom instructions)(Figure 3). It's important to note that variable titles could also be adjusted at this junction. Next, the upload itself occurred with no warnings, errors, or failed rows. The time start and end of the load can be seen, as well as the row count of 13,871

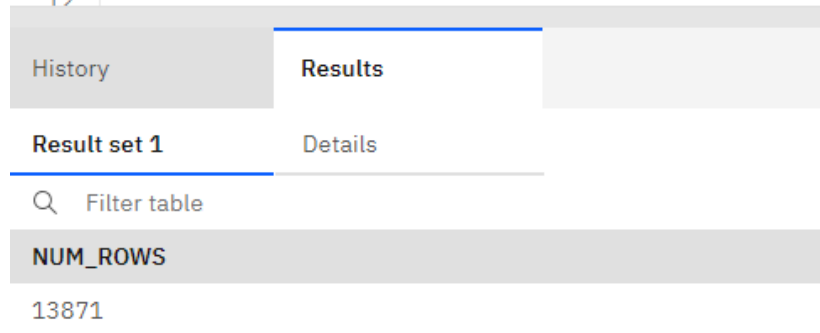
(Figure 5). Then the dataset could be observed to visually check that the variables were uploaded correctly (Figure 6). After performing visual checks to ensure the data loaded correctly and seeing the IBM DB2 automated messaging to see how the data was loaded, there are several programmatic approaches within the IBM DB2 environment that can be taken. These were focusing on verifying row counts and table metadata.

Initially, the `SELECT COUNT(*)` SQL query was used to read the total number of rows in the dataset, confirming that the number of entries matched our expectations based on the source file.

```

8  -- Check the rows count to ensure proper data load
9  SELECT count(*) num_rows
10  FROM FIRST_GOP_DEBATE;
11
12

```



The screenshot shows a SQL query execution interface. The query is: `SELECT count(*) num_rows FROM FIRST_GOP_DEBATE;`. The results are displayed in a table with one column, `NUM_ROWS`, and one row containing the value `13871`. The interface includes tabs for 'History' and 'Results', and a 'Result set 1' section with a 'Details' link. A search bar labeled 'Filter table' is also visible.

NUM_ROWS
13871

Figure 7. SQL code and output of counting all rows to ensure proper data load.

This step was crucial to ensure that the entire dataset was imported correctly without any omissions. Next, the table metadata was checked to verify the structure and integrity of the data. This involved querying the `SYSIBM.SYSTABLES` system catalog to retrieve information about the table, including the number of columns and their respective data types. By running `SELECT NAME, COLCOUNT FROM SYSIBM.SYSTABLES WHERE NAME = 'FIRST_GOP_DEBATE'`, it was confirmed that the table contained the correct number of columns as defined in the schema.

```

14 -- SYSIBM.SYSTABLES to see all tables uploaded, the creator, the time created, and column count
15 SELECT NAME, CREATOR, CTIME, COLCOUNT
16 FROM SYSIBM.SYSTABLES
17 WHERE CREATOR = 'TYG86489';
18

```

NAME	CREATOR	CTIME	COLCOUNT
AIRLINE_SENTIMENT	TYG86489	2024-06-30 21:14:12.576536	15
FIRST_GOP_DEBATE	TYG86489	2024-07-08 02:11:38.66694	21

Figure 8. SQL code and output of seeing all tables uploaded by user – Airline Sentiment table was from a previous analysis.

The SYSIBM.SYSCOLUMNS system catalog was utilized to examine detailed column metadata. The query `SELECT NAME, COLTYPE, LENGTH FROM SYSIBM.SYSCOLUMNS WHERE TBNAME = 'FIRST_GOP_DEBATE'` allowed verification that each column was correctly defined with the appropriate data type and length. This step was essential to ensure that the data would be accurately interpreted during subsequent analyses.

```

21 -- SYSIBM.SYSCOLUMNS to check metadata about each variable
22 SELECT NAME, TBNAME, TBcreator, COLTYPE, NULLS, LENGTH
23 FROM SYSIBM.SYSCOLUMNS
24 WHERE TBcreator='TYG86489'
25 AND TBNAME='FIRST_GOP_DEBATE';
26

```

NAME	TBNAME	TBcreator	COLTYPE	NULLS	LENGTH
CANDIDATE	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	22
CANDIDATE_CONFIDENCE	FIRST_GOP_DEBATE	TYG86489	DECIMAL	Y	8
CANDIDATE_GOLD	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	22
ID	FIRST_GOP_DEBATE	TYG86489	SMALLINT	Y	2
NAME	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	15
RELEVANT_YN	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	3
RELEVANT_YN_CONFIDENCE	FIRST_GOP_DEBATE	TYG86489	DECIMAL	Y	8
RELEVANT_YN_GOLD	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	5
RETWEET_COUNT	FIRST_GOP_DEBATE	TYG86489	SMALLINT	Y	2
SENTIMENT	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	8
SENTIMENT_CONFIDENCE	FIRST_GOP_DEBATE	TYG86489	DECIMAL	Y	8
SENTIMENT_GOLD	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	16
SUBJECT_MATTER	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	36
SUBJECT_MATTER_CONFIDENCE	FIRST_GOP_DEBATE	TYG86489	DECIMAL	Y	8
SUBJECT_MATTER_GOLD	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	45
TEXT	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	185
TWEET_COORD	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	28
TWEET_CREATED	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	14
TWEET_ID	FIRST_GOP_DEBATE	TYG86489	BIGINT	Y	8
TWEET_LOCATION	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	110
USER_TIMEZONE	FIRST_GOP_DEBATE	TYG86489	VARCHAR	Y	28

Figure 9. SQL code and output of checking metadata of each column/variable to ensure all data from each column was loaded correctly.



Additionally, the data was checked for any duplicate entries or missing values in key columns using SQL queries such as `SELECT tweet_id, COUNT(*) FROM FIRST_GOP_DEBATE GROUP BY tweet_id HAVING COUNT(*) > 1` to identify duplicates and `SELECT COUNT(*) - COUNT(column_name) FROM FIRST_GOP_DEBATE` to find null values. These checks helped to confirm data consistency and integrity, ensuring that the dataset was ready for further analysis. By systematically verifying row counts and table metadata through these SQL queries, it was ensured that the data was loaded correctly and maintained its expected structure, providing a reliable foundation for the subsequent sentiment analysis.

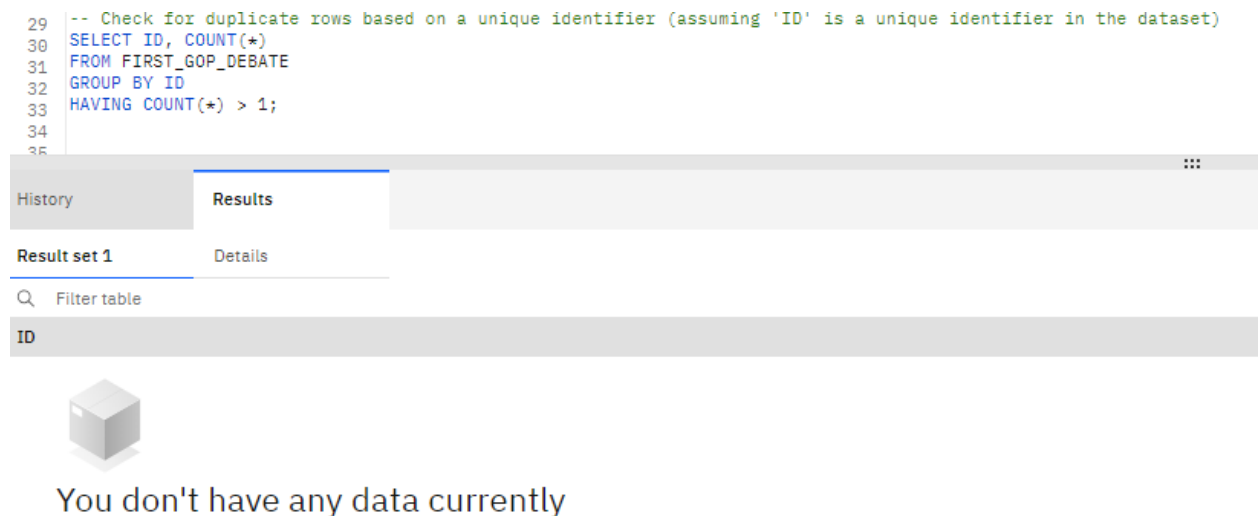


Figure 10. SQL code and output of checking duplicate rows did not exist to ensure all data from each column was loaded correctly.

```

37 -- Retrieve the first few rows of the table to verify data integrity and types
38 SELECT *
39 FROM FIRST_GOP_DEBATE
40 FETCH FIRST 10 ROWS ONLY;
41

```

History		Results						
Result set 1		Details						
Q		Filter table						
ID	CANDIDATE	CANDIDATE_CONFIDENCE	RELEVANT_YN	RELEVANT_YN_CONFIDENCE	SENTIMENT	SENTIMENT_CONFIDENCE	SUBJECT_MATTER	SUBJECT_MATTER_CONFIDENCE
1	No candidate mentioned	1.0000	yes	1.0000	Neutral	0.6578	None of the above	1.0000
2	Scott Walker	1.0000	yes	1.0000	Positive	0.6333	None of the above	1.0000
3	No candidate mentioned	1.0000	yes	1.0000	Neutral	0.6629	None of the above	0.6629
4	No candidate mentioned	1.0000	yes	1.0000	Positive	1.0000	None of the above	0.7039
5	Donald Trump	1.0000	yes	1.0000	Positive	0.7045	None of the above	1.0000
6	Ted Cruz	0.6332	yes	1.0000	Positive	0.6332	None of the above	1.0000
7	No candidate mentioned	1.0000	yes	1.0000	Negative	0.6761	FOX News or Moderators	1.0000
8	No candidate mentioned	1.0000	yes	1.0000	Neutral	1.0000	None of the above	1.0000
9	Ben Carson	1.0000	yes	1.0000	Negative	0.6889	None of the above	0.6444
10	No candidate mentioned	0.4594	yes	0.6778	Negative	0.6778	None of the above	0.4594

Figure 11. SQL code and output of pulling the first 10 rows of the dataset to visually ensure all data from each column was loaded correctly. This performs the same check as viewing the table in DB2 environment but via SQL.

Exploratory Data Analysis (EDA) is a critical step in data analysis that involves examining datasets to summarize their main characteristics, often using visual methods. It helps in understanding the data's structure, detecting patterns, spotting anomalies, and testing hypotheses. One of the initial EDA queries performed on the GOP debate tweets dataset focused on determining how many times each candidate was mentioned. This query is highly relevant as it provides insights into the level of public attention and interest each candidate received during the debate. By counting the mentions, the analysis reveals which candidates dominated the public discourse on social media, indicating their visibility and potential influence in the debate's context. Understanding the frequency of mentions can highlight key figures in the political landscape and guide further detailed sentiment analysis. The top contenders for the 2016 primary election were Trump, Cruz, Kasich, Rubio, Carson, Bush in that order with Trump boasting over 14 million votes and Bush emerging with 286,000 (Berg-Andersson, 2024). All other candidates earned less than 100,000 votes each. This correlates fairly well with the results of this query where Trump is by far the most mentioned candidate at 2,800 followed by Bush, Cruz, and

Carson in that order. In fact, all other candidates range from 242-705 mentions. All other mentions combined only amounts to 3,400, 600 more than Trump's total alone. This shows the way Trump dominated the online discourse during this time. There are also a whopping 7,491 Tweets with no candidate mentioned, in addition to another 96 with null value (though most of these null values were explored and they mentioned multiple candidates).

```

46 -- Query to count how many times each candidate is mentioned in column B (candidate) and order the results by the count of mentions
47 SELECT candidate, COUNT(*) AS mention_count
48 FROM FIRST_GOP_DEBATE
49 GROUP BY candidate
50 ORDER BY mention_count DESC;
51

```

CANDIDATE	MENTION_COUNT
No candidate mentioned	7491
Donald Trump	2813
Jeb Bush	705
Ted Cruz	637
Ben Carson	404
Mike Huckabee	393
Chris Christie	293
Marco Rubio	275
Rand Paul	263
Scott Walker	259
John Kasich	242
-	96

Figure 12. SQL code and output of querying the count of rows which mention each candidate.

A key exploratory query involves examining the overall count of each sentiment type—positive, negative, and neutral—in the dataset. This query provides an initial understanding of the general public mood during the 2015 GOP debate. By categorizing tweets based on sentiment, the analysis can identify the predominant sentiment expressed by the audience, offering a snapshot of the collective emotional response to the event. Following this, a more granular query breaks down these sentiment counts for each candidate. This detailed analysis is

crucial as it reveals not only the volume of attention each candidate received but also the nature of the sentiments associated with them. Comparing sentiment types across candidates can uncover public favorability or criticism directed towards specific individuals, shedding light on their perceived performance and impact during the debate.

It's interesting but not surprising that negative sentiment Tweets dominate at 8,500, followed by neutral at 3,100, and then positive at 2,200. This follows along with the old adage “bad news sells”. It’s critical to highlight that all candidates except for 3 had more negative sentiment Tweet mentions than positive ones. Those 3 exceptions are Cruz, Rubio, and Kasich. This is reflective of the fact that while Jeb Bush dominated much of the public discourse in the early campaign (trailing in second far after Trump), he eventually fell far behind where he received less than a 10<sup>th</sup> of the votes received by Cruz, Rubio, or Kasich (Berg-Andersson, 2024). By common knowledge, he had a notably failed campaign. It’s also critical to denote that a Tweet is far more likely to be neutral when no candidate is mentioned.

```

55 -- Query for # of rows for each sentiment level (excluding nulls)
56 SELECT
57     sentiment,
58     COUNT(*) AS Sentiment_Count
59 FROM
60     FIRST_GOP_DEBATE
61 WHERE
62     sentiment IS NOT NULL
63 GROUP BY
64     sentiment
65 ORDER BY
66     Sentiment_Count DESC:

```

SENTIMENT	SENTIMENT_COUNT
Negative	8493
Neutral	3142
Positive	2236

Figure 13. SQL code and output of querying the count of sentiment type: negative, neutral, and positive.

```

71 -- Query to return the candidate name, number of negative, positive, neutral mentions, and total mentions for each candidate
72 SELECT
73     candidate AS Candidate_Name,
74     SUM(CASE WHEN sentiment = 'Negative' THEN 1 ELSE 0 END) AS Negative_Mentions,
75     SUM(CASE WHEN sentiment = 'Positive' THEN 1 ELSE 0 END) AS Positive_Mentions,
76     SUM(CASE WHEN sentiment = 'Neutral' THEN 1 ELSE 0 END) AS Neutral_Mentions,
77     COUNT(*) AS Total_Mentions
78 FROM
79     FIRST_GOP_DEBATE
80 GROUP BY
81     candidate
82 ORDER BY
83     Total_Mentions DESC;
84

```

CANDIDATE_NAME	NEGATIVE_MENTIONS	POSITIVE_MENTIONS	NEUTRAL_MENTIONS	TOTAL_MENTIONS
No candidate mentioned	4724	680	2087	7491
Donald Trump	1758	609	446	2813
Jeb Bush	589	44	72	705
Ted Cruz	221	290	126	637
Ben Carson	186	164	54	404
Mike Huckabee	237	73	83	393
Chris Christie	218	33	42	293
Marco Rubio	105	119	51	275
Rand Paul	148	55	60	263
Scott Walker	179	42	38	259
John Kasich	82	113	47	242
-	46	14	36	96

Figure 14. SQL code and output of querying the count of sentiment type: negative, neutral, and positive for each candidate.

Another insightful query in the exploratory data analysis involves identifying the most retweeted tweet for each candidate and calculating the average number of retweets they received. This query is significant as it highlights the most impactful and widely shared statements made by each candidate, indicating the moments that resonated most with the audience. The most retweeted tweets can reveal key points or messages that gained traction and sparked widespread engagement. Additionally, analyzing the average number of retweets provides a broader understanding of each candidate's overall influence and engagement on social media during the debate. High average retweet counts suggest consistent interest and engagement from the audience, reflecting the candidate's ability to capture and maintain public attention (for better or worse). This dual analysis of peak and average retweet activity offers a comprehensive view of the candidates' social media performance and influence. It's unique to point out that Bush

outperformed Trump in highest retweet count (though by a mere 24). Huckabee also was in the top 4 highest retweet count despite his overall poor voting turnout. It is interesting that there is not a clear emergent pattern from highest retweet or average retweet count amongst the candidates (aside from Bush dominating both categories). This appears to be indicative that despite voting turnout and general public discourse, the retweet patterns don't indicate any of the candidates as standout except Bush indicating he was more of a controversial candidate within the GOP.

```

88 -- Query to find the most retweeted tweet count for each candidate and the average retweet count for each candidate
89 SELECT
90     candidate AS Candidate_Name,
91     MAX(retweet_count) AS Most_Retweeted_Tweet_Count,
92     AVG(retweet_count) AS Average_Retweet_Count
93 FROM
94     FIRST_GOP_DEBATE
95 GROUP BY
96     candidate
97 ORDER BY
98     Most_Retweeted_Tweet_Count DESC;
99

```

CANDIDATE_NAME	MOST_RETWEETED_TWEET_COUNT	AVERAGE_RETWEET_COUNT
No candidate mentioned	4965	44
Jeb Bush	3427	80
Donald Trump	3403	46
Ben Carson	2209	46
Mike Huckabee	1736	42
Ted Cruz	1449	68
Rand Paul	1322	23
Marco Rubio	842	24
Chris Christie	791	32
John Kasich	717	22
Scott Walker	653	27
-	449	29

Figure 15. SQL code and output of querying the highest retweet number and average retweet number for each candidate.

Analyzing the average retweet count for each sentiment type—negative, neutral, and positive—provides valuable insights into how different types of content engage audiences on social media. In this case, tweets with a negative sentiment had the highest average retweet count at 49, followed by positive tweets with an average of 44 retweets, and neutral tweets with an

average of 37 retweets. This pattern suggests that negative tweets garnered the most engagement, reflecting a tendency for social media users to interact more with content expressing criticism or discontent. Positive tweets also attracted significant attention, though slightly less than negative ones, indicating that uplifting or favorable messages were also widely shared. Neutral tweets, while still engaging, elicited the least interaction, possibly due to their less emotionally charged nature. This analysis underscores the impact of sentiment on social media engagement, highlighting how emotional content can drive higher levels of user interaction.

```

104 -- Query to calculate the average number of retweets and favorites for each sentiment category
105 SELECT
106     SENTIMENT,
107     AVG(RETWEET_COUNT) AS Average_Retweets
108 FROM
109     FIRST_GOP_DEBATE
110 GROUP BY
111     SENTIMENT
112 ORDER BY
113     SENTIMENT;
114

```

SENTIMENT	AVERAGE_RETWEETS
Negative	49
Neutral	37
Positive	44

Figure 16. SQL code and output of querying the average retweet number for each sentiment.

Then, a SQL query was designed to find the most active days, based on tweet counts, for each candidate mentioned in the dataset. By extracting the date portion from the TWEET\_CREATED field and grouping the data by both candidate and date, the query calculates the total number of tweets for each candidate on each day. The output lists candidates along with the dates on which they were most frequently mentioned and the corresponding tweet counts.

There were only 2 dates in the dataset: August 6<sup>th</sup> and August 7<sup>th</sup> 2015 representing before and after the first primary debate. From the results, it's clear that August 7, 2015, was a much more active day, likely corresponding to capturing the public's interest in the debate. Candidates like Donald Trump, Jeb Bush, and Marco Rubio received significant attention on this date, with Donald Trump having the highest tweet count at 1044. This indicates a high level of public engagement and discourse surrounding these candidates during the event. Other candidates like Ben Carson and Chris Christie also saw substantial mentions, suggesting that they were key figures in the discussions. The data shows that specific events drive spikes in social media activity, reflecting the immediate public interest and reactions to these political figures during critical moments in the campaign. This analysis helps in understanding the dynamics of public engagement with different candidates over time.



```

117 -- Query to find the most active day (with the highest number of tweets) for each candidate
118
119 SELECT
120     CANDIDATE AS Candidate_Name,
121     LEFT(TWEET_CREATED, LOCATE(' ', TWEET_CREATED) - 1) AS Tweet_Date,
122     COUNT(*) AS Total_Tweets
123 FROM
124     FIRST_GOP_DEBATE
125 GROUP BY
126     CANDIDATE, LEFT(TWEET_CREATED, LOCATE(' ', TWEET_CREATED) - 1)
127 ORDER BY
128     Candidate_Name, Total_Tweets DESC;

```

History Results

Result set 1 Details

Q Filter table

CANDIDATE_NAME	TWEET_DATE	TOTAL_TWEETS
Ben Carson	8/7/2015	213
Ben Carson	8/6/2015	191
Chris Christie	8/7/2015	177
Chris Christie	8/6/2015	116
Donald Trump	8/7/2015	1844
Donald Trump	8/6/2015	969
Jeb Bush	8/6/2015	478
Jeb Bush	8/7/2015	227
John Kasich	8/7/2015	145
John Kasich	8/6/2015	97
Marco Rubio	8/7/2015	203
Marco Rubio	8/6/2015	72
Mike Huckabee	8/6/2015	212
Mike Huckabee	8/7/2015	181
No candidate mentioned	8/7/2015	4842
No candidate mentioned	8/6/2015	2649
Rand Paul	8/7/2015	187
Rand Paul	8/6/2015	76
Scott Walker	8/7/2015	166
Scott Walker	8/6/2015	93

Figure 17. SQL code and output of querying the tweet count for each candidate and for each date in the dataset.

Subsequently, a SQL query then retrieved the top 10 most retweeted tweets from the dataset. This provides valuable insights into which messages garnered the most engagement during the first 2015 Primary GOP debate. This query is particularly useful for (EDA as it highlights the content that resonated most with the audience, indicating key moments or statements that triggered significant reactions from the nation's collective attention. By ordering the tweets by retweet\_count in descending order, the query identifies tweets that were widely shared, reflecting the most influential or provocative content.

The results reveal that the most retweeted tweets often did not mention specific candidates but rather discussed broader topics or reactions to the debate. For instance, tweets with sentiments about how the debate was handled or reactions from notable figures like Hillary

Clinton and Bernie Sanders received high retweet counts. This indicates that the public discourse was not solely focused on the candidates but also on the overall event and its reception by prominent personalities. The presence of both positive and negative sentiments among the top retweeted tweets suggests that highly engaging content can be polarizing, capturing attention regardless of the sentiment. However, 7 of the top 10 tweets were all negative, followed by 2 positive, and finally 1 neutral. This continues the earlier exposed trend showing that most tweets were eliciting negative sentiment, followed by positive, and then followed by neutral. Critically, the results show that while tweets mentioning candidates like Bush and Trump were among the most retweeted, they were fewer compared to broader discussion tweets. This highlights the multifaceted nature of social media engagement, where not just candidate-specific statements but also general reactions and commentary play a significant role in driving public interest and interaction. Such insights are crucial for understanding the dynamics of social media engagement and can guide further detailed analysis of sentiment and topics related to the debate. It is also of note that almost all of these tweets include #GOPdebate in them showing the importance of including hashtags for widest audience reach. They also include several other prominent hashtags and several include links in them. In addition, every tweet is a retweet towards one or more major politician or celebrity.

```

1 -- Query to find the top 10 most retweeted tweets
2 SELECT
3   id,
4   candidate,
5   sentiment,
6   retweet_count,
7   text
8 FROM
9   FIRST_GOP_DEBATE
10 ORDER BY
11   retweet_count DESC
12 FETCH FIRST 10 ROWS ONLY;

```

ID	CANDIDATE	SENTIMENT	RETWEET_COUNT	TEXT
30	No candidate mentioned	Negative	4965	RT @HillaryClinton: Watch the #GOPDebate? Bet you feel like donating to a Democrat right about now. <a href="http://t.co/pGIQCqQgOP">http://t.co/pGIQCqQgOP</a> <a href="http://t.co/QP1e...">http://t.co/QP1e...</a>
46	No candidate mentioned	Negative	4416	RT @LeKarmaSucre: How the #GOPDebate handled #BlackLivesMatter <a href="http://t.co/8ZGdY6iPCT">http://t.co/8ZGdY6iPCT</a>
250	No candidate mentioned	Positive	4270	RT @BernieSanders: Tom Hanks. Finally. Somebody who makes some sense. #GOPDebate #DebateWithBernie
42	No candidate mentioned	Negative	4006	RT @JanelleMyBelle: Meanwhile, in the White House... #GOPDebate <a href="http://t.co/nouUUT5hKq">http://t.co/nouUUT5hKq</a>
803	No candidate mentioned	Negative	3946	RT @AdamSmith_USA: democrats watching the #GOPDebate <a href="http://t.co/MuSUto1IRh">http://t.co/MuSUto1IRh</a>
132	No candidate mentioned	Neutral	3847	RT @HillaryClinton: Missing Jon Stewart already. #GOPDebate #JonVoyage -H
410	No candidate mentioned	Positive	3469	RT @RealBenCarson: May the Lord guide my words tonight, let His wisdom be my thoughts. #GOPDebate
38	Jeb Bush	Negative	3427	RT @kvrdashian: Jeb Bush: "Obama is at fault, not my brother, because Obama didn't clean up the mess my brother made." #GOPDebate
414	Donald Trump	Negative	3403	RT @deray: Trump literally said that he donates to politicians and then calls them later to get what he wants. He literally just said that....
256	No candidate mentioned	Negative	3385	RT @BernieSanders: Oh. It was just a movie trailer. #GOPDebate #DebateWithBernie

Figure 18. SQL code and output of querying the top 10 most retweeted tweets in the dataset, which candidate they pertained to, the sentiment, and the text.

The final EDA query focused on identifying the number of missing values for each variable in the dataset. The SQL code used a combination of the COUNT function and conditional logic to determine the number of null values in each column. The query involved multiple SELECT statements for each column, with each statement calculating the total number of rows minus the count of non-null entries for that specific column. These results were then combined using the UNION ALL function to produce a comprehensive list of columns along with their respective missing value counts (note: the image only shows the repeating code for the first variable, but it continues using a JOIN statement for each variable). In the output displayed, the query results show the number of missing values for each column in the dataset. This output is crucial for understanding data quality and completeness, which are essential for any subsequent analysis. Variables such as CANDIDATE, SENTIMENT, RETWEET\_COUNT, and TEXT have no missing values, indicating that these fields were consistently recorded across all

entries, ensuring reliability in analyses involving these variables. These are the variables that matter most to this analysis.

However, some columns have significant numbers of missing values. For example, the TWEET\_COORD, CANDIDATE\_GOLD, and SUBJECT\_MATTER\_GOLD columns show substantial gaps, with each having over 13,000 missing values. High levels of missing data in these columns may limit their usefulness in certain analyses or require imputation strategies to address the gaps. The presence of missing values, particularly in columns, suggests incomplete data collection or optional fields. These gaps might impact the accuracy of insights derived from these variables and necessitate careful handling during data preprocessing. Of the variables which have many missing values, there are 4 ending in \_GOLD in reference to being included in a model previously made. These will be irrelevant to our analysis. In addition, there are 3 variables with a handful of missing values: TWEET\_COORD, TWEET\_LOCATION, AND USER\_TIMEZONE. These will be kept as part of the dataset as they may yield further information during analysis.

```

144 -- Query to count the number of missing (NULL) values for each variable in the dataset
145
146 SELECT
147     'ID' AS Column_Name,
148     COUNT(*) - COUNT(ID) AS Missing_Values
149 FROM
150     FIRST_GOP_DEBATE
151 UNION ALL
152

```

COLUMN_NAME	MISSING_VALUES
ID	0
CANDIDATE	96
CANDIDATE_CONFIDENCE	0
RELEVANT_YN	0
RELEVANT_YN_CONFIDENCE	0
SENTIMENT	0
SENTIMENT_CONFIDENCE	0
SUBJECT_MATTER	326
SUBJECT_MATTER_CONFIDENCE	0
CANDIDATE_GOLD	13843
NAME	0
RELEVANT_YN_GOLD	13839
RETWEET_COUNT	0
SENTIMENT_GOLD	13865
SUBJECT_MATTER_GOLD	13853
TEXT	0
TWEET_COORD	13860
TWEET_CREATED	0
TWEET_ID	0
TWEET_LOCATION	3919
USER_TIMEZONE	4403

Figure 19. SQL code and output of querying the number of missing values for each variable.

### **Method – RStudio Analysis:**

The next major section of this analysis focused on using R to perform sentiment analysis in Watson Studio. The R code involved setting up the environment and verifying that the data is properly loaded and ready for EDA. First, various libraries such as tm, wordcloud, and SnowballC were loaded. These libraries provided functions for text mining, data visualization, and natural language processing, essential for analyzing the sentiment of tweets. Next, the CSV file was read and loaded. The tweets were put into a dataframe named SENTIMENT. The structure and unique values of key columns were then examined to ensure data integrity. This step was akin to initial data checks performed in SQL, where row counts and table metadata are inspected to confirm successful data loading. After the number of rows, the number of tweets for each sentiment, and number of tweets for each candidate were confirmed, followed by confirmation of the number of tweets by candidate and sentiment, further analysis proceeded.

```

> nrow(SENTIMENT)
[1] 13871
>
> # Number of tweets per sentiment
> table(SENTIMENT$SENTIMENT)

Negative Neutral Positive
 8493    3142    2236
>
> # Number of tweets per candidate
> table(SENTIMENT$CANDIDATE)

          Ben Carson      Chris Christie      Donald Trump      Jeb Bush
          96          404          293          2813          705
John Kasich      Marco Rubio      Mike Huckabee      No candidate mentioned      Rand Paul
 242          275          393          7491          263
Scott Walker      Ted Cruz
 259          637
\

> # Number of tweets per sentiment by candidate
> table(SENTIMENT$CANDIDATE, SENTIMENT$SENTIMENT)

          Negative Neutral Positive
Ben Carson          46          36          14
Chris Christie      218          42          33
Donald Trump        1758         446         609
Jeb Bush            589          72          44
John Kasich         82          47         113
Marco Rubio         105          51         119
Mike Huckabee       237          83          73
No candidate mentioned 4724        2087        680
Rand Paul           148          60          55
Scott Walker        179          38          42
Ted Cruz            221         126         290

```

Figure 20. R code and output of querying the number of rows, tweets per sentiment, tweets per candidate, and tweets per sentiment and candidate to confirm proper data load.

Unlike the SQL EDA, it was also checked to see the number of tweets per subject matter. This gives some key insights into the public's general interest per topic. Some of the top topics included Fox News/moderators, Women's Issues, and Abortion. There are also a few standout outliers for example Racial Issues being very high for Dr. Carson.

```
> # Number of tweets by candidate per subject matter
> table(SENTIMENT$SUBJECT_MATTER, SENTIMENT$CANDIDATE)
```

	Ben Carson	Chris Christie	Donald Trump	Jeb Bush	John Kasich	Marco Rubio
Abortion	48	8	5	66	12	3
Foreign Policy	0	1	2	17	12	2
FOX News or Moderators	2	9	35	26	21	1
Gun Control	3	16	11	849	156	4
Healthcare (including Medicare)	0	0	0	2	2	0
Immigration	0	4	2	6	3	7
Jobs and Economy	4	0	6	62	23	0
LGBT issues	0	6	7	40	19	10
None of the above	1	5	1	1	1	31
Racial issues	37	274	221	1604	445	179
Religion	0	70	1	12	3	1
Women's Issues (not abortion though)	0	10	1	14	1	4
	1	1	1	114	7	0

	Mike Huckabee	No candidate mentioned	Rand Paul	Scott Walker	Ted Cruz
Abortion	7	153	3	9	7
Foreign Policy	18	181	1	29	4
FOX News or Moderators	45	155	13	25	31
Gun Control	4	1775	5	11	62
Healthcare (including Medicare)	4	43	9	0	1
Immigration	6	31	1	2	4
Jobs and Economy	1	87	0	3	6
LGBT issues	14	120	6	10	2
None of the above	28	53	2	0	1
Racial issues	247	4095	220	145	500
Religion	6	248	1	9	0
Women's Issues (not abortion though)	3	330	1	12	19
	10	220	1	4	0

Figure 21. R code and output of querying the number tweets per subject per candidate.

Next, a pie chart was generated to visualize the number of tweets per candidate.

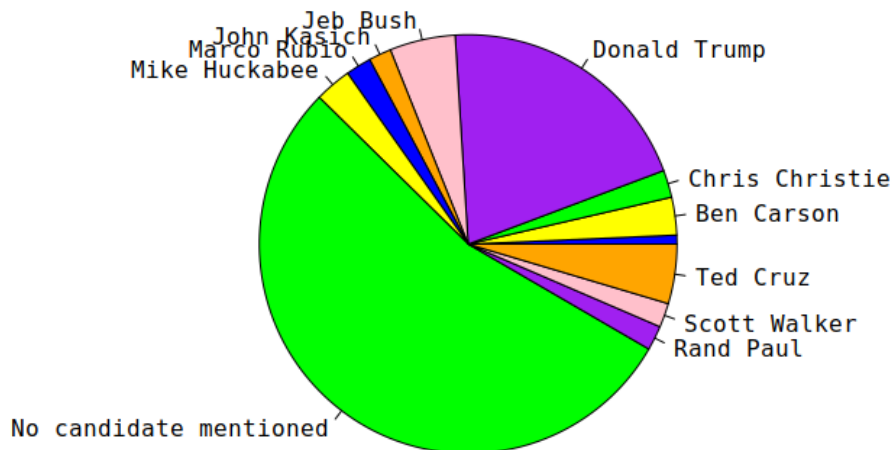


Figure 22. Pie chart of number of tweets per candidate.

Subsequently, a table was generated to show the count of tweets per each topic in descending order. The topic of Fox News/Moderators was by far the most common highlighting

how many people cared about how the debate was handled by Fox News. The volume of tweets is nearly 10x any other category. The next most popular topics were Religion, Foreign Policy, and Women's issues all in the 360-410 range.

```
> # List the most common subjects
> reasonCounts <- na.omit(plyr::count(SENTIMENT$SUBJECT_MATTER))
> reasonCounts <- reasonCounts[order(reasonCounts$freq, decreasing=TRUE), ]
> reasonCounts
```

		x	freq
10	None of the above	8148	
4	FOX News or Moderators	2900	
12	Religion	407	
3	Foreign Policy	366	
13	Women's Issues (not abortion though)	362	
11	Racial issues	353	
1		326	
2	Abortion	293	
8	Jobs and Economy	251	
7	Immigration	211	
9	LGBT issues	126	
6	Healthcare (including Medicare)	67	
5	Gun Control	61	

Figure 23. R code and output of querying the number tweets per subject in descending order.

A similar bar chart of the output was generated to visualize the same thing showing the vast difference between the most popular topic versus all others.

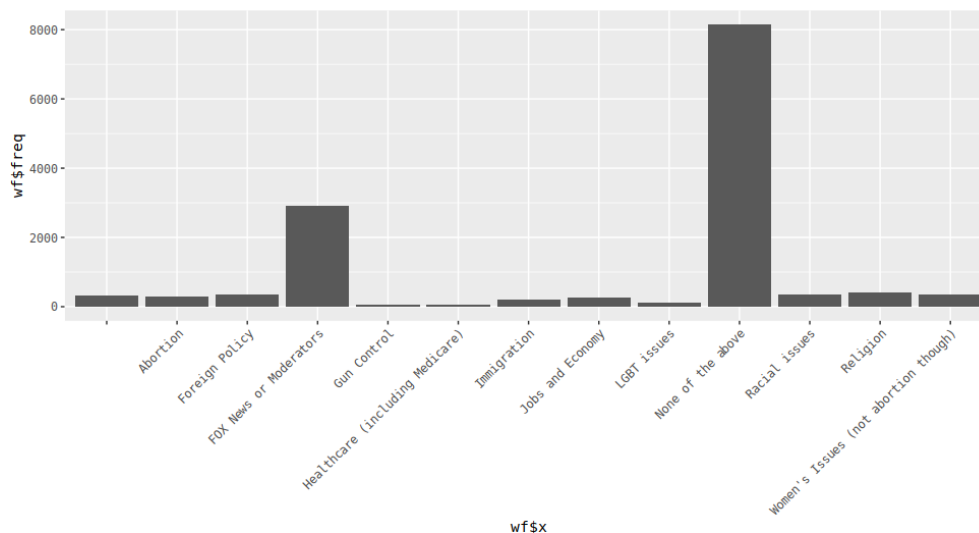


Figure 24. Bar chart showing the number tweets per subject.



Next, a table was generated to show the total number of tweets per each category.

This was thought to be useful to show how each topic resonated not just within this dataset but with other users retweeting the data here. Interestingly, Fox News still led the pack followed by Racial Issues, Religion, and Abortion (in opposition to the previous tweet analysis).

```
> # Number of retweets per subject matter
> ddply(SENTIMENT, ~ SUBJECT_MATTER, summarize, numRetweets = sum(RETWEET_COUNT, na.rm = TRUE))
```

	SUBJECT_MATTER	numRetweets
1		8738
2	Abortion	17044
3	Foreign Policy	14974
4	FOX News or Moderators	194129
5	Gun Control	4106
6	Healthcare (including Medicare)	922
7	Immigration	11820
8	Jobs and Economy	8305
9	LGBT issues	4878
10	None of the above	314759
11	Racial issues	19935
12	Religion	18973
13	Women's Issues (not abortion though)	16755

Figure 25. R code and output of querying the total number of retweets per subject.

The top 4 most retweeted tweets were then queried in order to visually see and understand what users cared most about.

```
> # Posts that have more than 4000 retweets
> as.character(subset(SENTIMENT, RETWEET_COUNT > 4000)$TEXT)
```

```
[1] "RT @HillaryClinton: Watch the #GOPdebate? Bet you feel like donating to a Democrat right about now. http://t.co/pG1QCqQgOP http://t.co/QP1e..."
[2] "RT @JanelleMyBelle: Meanwhile, in the White House... #GOPDebate http://t.co/nouUUt5hKq"
[3] "RT @LeKarmaSucre: How the #GOPDebate handled #BlackLivesMatter http://t.co/8ZGdY6lPCT"
[4] "RT @BernieSanders: Tom Hanks. Finally. Somebody who makes some sense. #GOPDebate #DebateWithBernie"
```

Figure 26. R code and output of querying the top 4 most retweeted tweets.

Next, the number of tweets per day were queried along with confirming which day had the most tweets. August 6<sup>th</sup> having 5,303 versus August 7<sup>th</sup> having 8,568 show a slight uptick from the first debate occurring on the former date and users interacting with posts, retweeting, and tweeting more in the day after as they heard the news and digested what occurred.

```

> # Number of posts per day
> posts <- as.Date(SENTIMENT$TWEET_CREATED)
> table(posts)
posts
2015-08-06 2015-08-07
      5303      8568
> # Day with the maximum number of posts
> table(posts)[which.max(table(posts))]
2015-08-07
      8568

```

Figure 27. R code and output of querying the number of tweets per day and confirming which day had the most tweets.

Plotting the number of tweets per day by sentiment provided a visual representation of how sentiment trends fluctuated over the two days. The TWEET\_CREATED field was converted to a date format, and the data was grouped by both sentiment and date. By using ggplot2, a line plot was created where each line represented a different sentiment category. This visualization helped to that the rate of change appears similar for negative and neutral while it appears slightly lower for positive sentiment. This shows again how people are less interested in positive sentiment tweets as shown by engaging with less positive tweets versus more negative and neutral tweets from Aug 6<sup>th</sup> to Aug 7<sup>th</sup>.

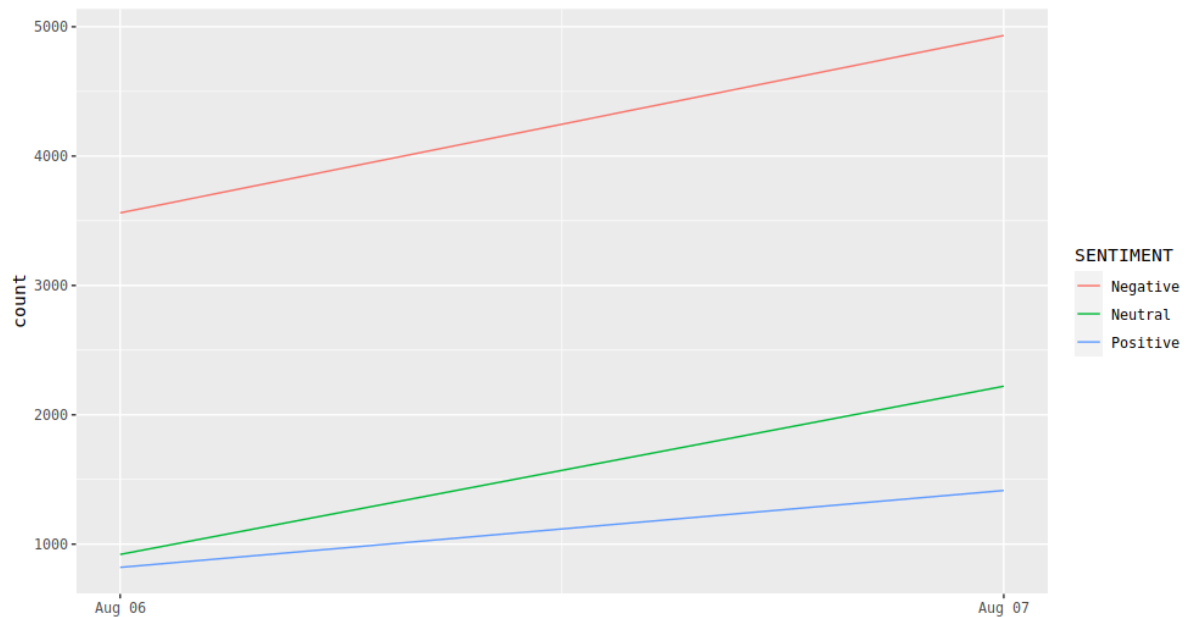


Figure 28. R code and output of querying the number of tweets per day and confirming which day had the most tweets.

For the Apriori approach, association rule mining was applied to discover relationships between different variables within the dataset. Relevant variables like CANDIDATE, SENTIMENT, SUBJECT\_MATTER, and RETWEET\_COUNT were transformed into factors. The Apriori algorithm was then used to generate association rules based on specified support and confidence levels. These rules were inspected to identify interesting and significant patterns, such as common combinations of sentiments and subjects associated with specific candidates. Visualization of the rules using arulesViz provided a graphical representation of the associations, allowing for easier interpretation and analysis of the underlying data relationships. This method revealed important insights into the interplay between various factors within the tweets, highlighting key trends and associations.

The Apriori rules identified strong associations between certain subjects, sentiments, and retweet counts. For instance, tweets mentioning specific subject matters such as “Gun Control” and “Women's Issues” frequently resulted in negative sentiments and higher retweet counts (2+). Additionally, time zones like “Mazatlan” and “Dublin” were also associated with tweets that had higher retweet counts (though it’s unclear if this is due to socioeconomic status, popularity, or something else like using a VPN). The visualization further highlighted these associations, showing that certain candidates, such as Jeb Bush and Ted Cruz, were often linked with specific subject matters and sentiments. The lift values in the rules indicate the strength of these associations, with higher lift values suggesting stronger relationships.

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{USER_TIMEZONE=Mazatlan}	=> {RETWEET_COUNT=2+}	0.001009300	0.8750000	0.001153486	1.657398	14
[2]	{USER_TIMEZONE=Dublin}	=> {RETWEET_COUNT=0}	0.001513950	0.8750000	0.001730229	2.387788	21
[3]	{USER_TIMEZONE=Rome}	=> {RETWEET_COUNT=2+}	0.001802321	0.8333333	0.002162786	1.578474	25
[4]	{SUBJECT_MATTER=Gun Control}	=> {SENTIMENT=Negative}	0.003604643	0.8196721	0.004397664	1.338711	50
[5]	{CANDIDATE=Rand Paul}	=> {SUBJECT_MATTER=None of the above}	0.015860428	0.8365019	0.018960421	1.424045	220
[6]	{SUBJECT_MATTER=Racial issues}	=> {SENTIMENT=Negative}	0.021051114	0.8271955	0.025448778	1.350998	292
[7]	{SUBJECT_MATTER=Women's Issues (not abortion though)}	=> {SENTIMENT=Negative}	0.023358085	0.8950276	0.026097614	1.461784	324
[8]	{SUBJECT_MATTER=Religion}	=> {CANDIDATE=No candidate mentioned}	0.023790642	0.8108108	0.029341792	1.501369	330
[9]	{CANDIDATE=Jeb Bush}	=> {SENTIMENT=Negative}	0.042462692	0.8354610	0.050825463	1.364498	589
[10]	{SUBJECT_MATTER=None of the above, USER_TIMEZONE=Dublin}	=> {RETWEET_COUNT=0}	0.001153486	0.8421053	0.001369764	2.298021	16
[11]	{SUBJECT_MATTER=FOX News or Moderators, USER_TIMEZONE=Rome}	=> {RETWEET_COUNT=2+}	0.001081393	1.0000000	0.001081393	1.894169	15
[12]	{SENTIMENT=Negative, USER_TIMEZONE=Rome}	=> {RETWEET_COUNT=2+}	0.001081393	0.9375000	0.001153486	1.775783	15
[13]	{SUBJECT_MATTER=Gun Control, RETWEET_COUNT=2+}	=> {SENTIMENT=Negative}	0.002379064	0.8461538	0.002811621	1.381962	33
[14]	{CANDIDATE=No candidate mentioned, SUBJECT_MATTER=Gun Control}	=> {SENTIMENT=Negative}	0.002739529	0.8837209	0.003099993	1.443317	38
[15]	{CANDIDATE=Jeb Bush, USER_TIMEZONE=America/New York}	=> {SENTIMENT=Negative}	0.001009300	1.0000000	0.001009300	1.633227	14

Figure 29. R code output of querying the top 15 Apriori rules.

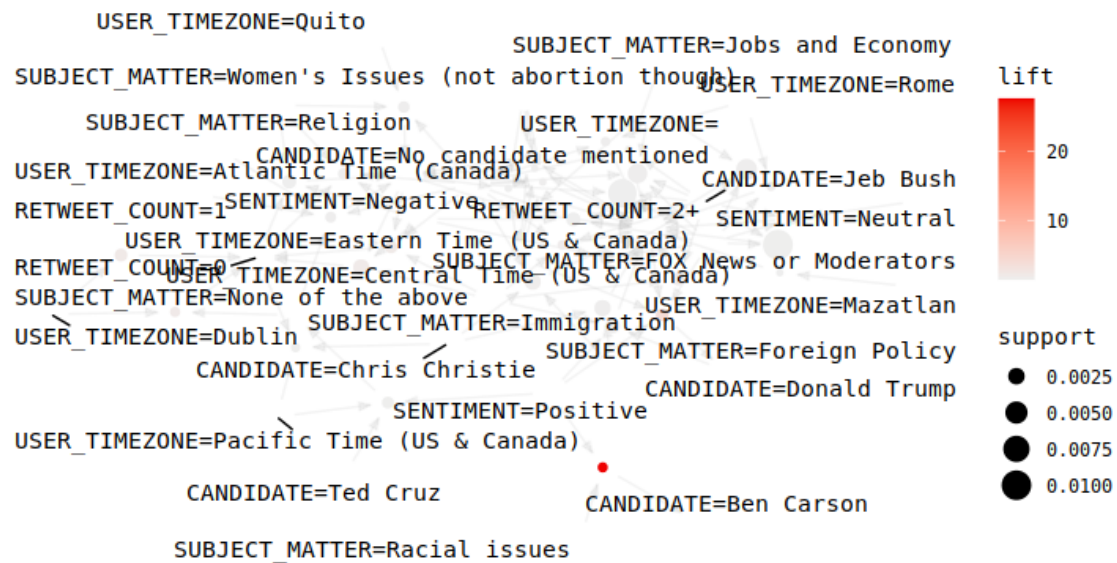


Figure 30. Apriori plot showing associations between different rules (only top 33 rules were selected for this plot).

Subsequently, text mining was performed to extra the key words from the tweets. Text mining involved several preprocessing steps to clean and prepare the textual data for analysis. The initial steps included converting text to a consistent format by removing whitespace, URLs, non-ASCII characters, punctuation, and numbers. Functions like `tm_map()` in R were utilized to apply these transformations systematically. Additionally, converting all text to lowercase ensured uniformity, which helped in accurately counting word frequencies. Stop words, which are common words that did not add significant meaning (such as “and”, “the”, “is”), were removed to reduce noise in the data. Stemming was another crucial preprocessing step, where words were reduced to their root form (e.g., “running” to “run”), enabling the analysis to treat different forms of a word as a single term. These preprocessing techniques ensured that the text data was clean and standardized, providing a reliable foundation for further analysis. Once the data was preprocessed, advanced techniques such as creating a Document-Term Matrix (DTM) were employed. A DTM is a matrix representation of the corpus, where rows represented documents

and columns represented terms. The entries in the matrix indicated the frequency of terms within each document. This matrix served as the basis for various analyses, including frequency analysis, association rule mining, and clustering.

	catch	didnt	full	gopdeb	line	night	scott	scottwalk	second	walker
1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0
5	0	0	0	1	1	1	0	0	0	0
6	0	0	0	1	0	0	1	0	0	1
7	0	0	0	1	0	0	0	0	0	0

Figure 31. DTM showing only the preview of the first several rows/columns of many, many rows/columns.

After text mining was performed, the wordcloud generated from the positive sentiment tweets provides a visual representation of the most frequently used words. Prominent terms like “Trump”, “GOPDebate”, “Fox”, “Hillary”, and “President” stand out, indicating high relevance and frequency in the context of the GOP debate. Names of several candidates, including “Ben Carson”, “John Kasich”, and “Marco Rubio”, also appear frequently, reflecting their significant mention during the debate. Other notable words include “leader”, “immigration”, and “winner”, suggesting key themes and topics discussed in the tweets. The wordcloud helps to quickly identify the main subjects and individuals that captured the public's attention, highlighting the focal points of positive sentiment during the debate (despite its obvious limitation being that it is not a easily comparable quantitative tool).

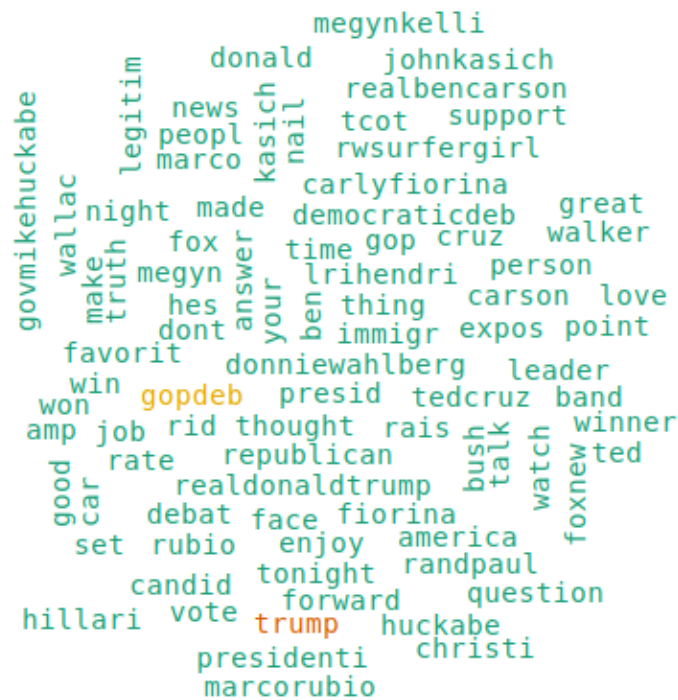


Figure 32. Wordcloud for the top used words from positive sentiment tweets.

Subsequently, a term-term adjacency matrix was generated from the Term Document Matrix (TDM). It showed the co-occurrence frequencies of terms within the same tweets. Each cell represented the number of times the terms in the corresponding row and column appeared together in the dataset. For example, “love” and “realdonaldtrump” co-occurred 13 times, indicating they were frequently mentioned together in tweets about the GOP debate. This matrix helps identify which terms are most commonly associated with each other allowing further analysis to occur revealing patterns and relationships within the discussion.

	gopdeb	night	debat	rate	realdonaldtrump	trump	foxnew	tedcruz	love
gopdeb	2092	191	241	95	267	388	97	160	93
night	191	196	25	3	18	17	12	6	8
debat	241	25	253	11	66	34	12	63	4
rate	95	3	11	97	77	7	5	0	0
realdonaldtrump	267	18	66	77	301	28	26	28	13
trump	388	17	34	7	28	413	7	3	10
foxnew	97	12	12	5	26	7	112	2	2
tedcruz	160	6	63	0	28	3	2	166	5
love	93	8	4	0	13	10	2	5	96

Figure 33. Term-term matrix (showing the co-occurrence of words) displaying only the preview of the first several rows/columns of many, many rows/columns.

A dendrogram is a tree-like diagram used to visualize the arrangement of clusters produced by hierarchical clustering (Podani & Schmera, 2006). A dendrogram was created to illustrate the hierarchical relationships between terms in the tweets related to the GOP debate. The terms are arranged on the x-axis, while the y-axis represents the distance or dissimilarity between clusters of terms. The height of the branches indicates the level of similarity between the terms or clusters, with shorter branches signifying higher similarity. The dendrogram shows distinct clusters of terms that frequently appear together in the dataset. For example, terms like “Rubio”, “Cruz”, “Job”, and “Bush” form a close cluster, indicating that these terms were often mentioned together in the tweets. This suggests that discussions about the debate frequently included references to these topics and individuals. Another cluster includes terms such as “Fox”, “News”, “Rate”, and “realdonaldtrump” which are associated with evaluating the performance of the host of the debate, Fox News. Rwsurfergirl (the handle of Right Wing Surfer Girl) also appears in this cluster noting how she was a notable (>600,000 followers) and vocal supporter of Donald Trump on Twitter at the time (before being suspended)(Nash, 2016).



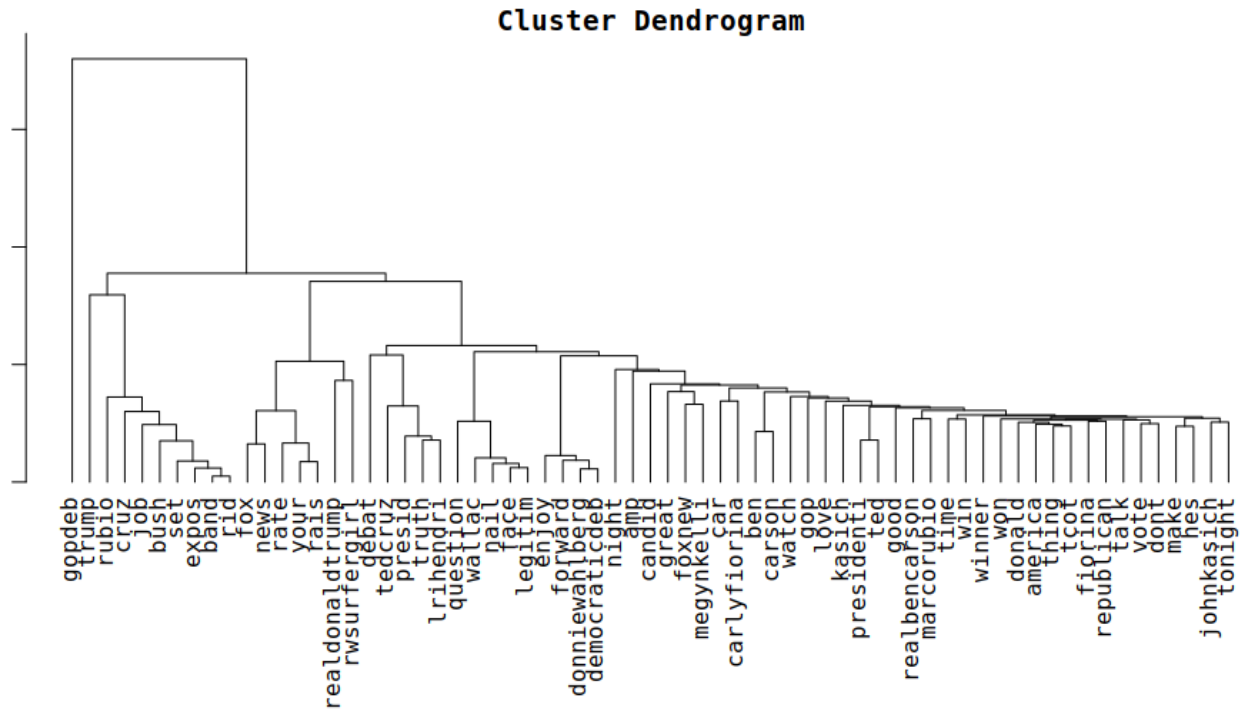


Figure 34. Dendrogram showing relationships of words from positive sentiment tweets.

The network graph using the Fruchterman-Reingold layout showed a clear, structured visualization of the relationships between terms (left). Key terms like “realDonaldTrump,” “GOPDebate,” and “debate” are prominently positioned at the center, indicating their central role in the discussions. The dense network of connections surrounding these terms suggests that they were frequently mentioned together with various other terms. Insights from this plot highlight the central figures and topics of the debate, showing how discussions were highly interconnected, particularly around Donald Trump and the debate itself. In the network graph utilizing the Gem layout, the terms are arranged in a visually appealing manner, emphasizing the interconnectedness and groupings of terms. Terms like “realDonaldTrump,” “GOPDebate,” and “debate” still remain central, with strong connections to other significant terms such as “Bush” and “question.” This layout highlights the relationships between specific candidates and key debate themes, providing insights into the focal points of public interest during the debate.

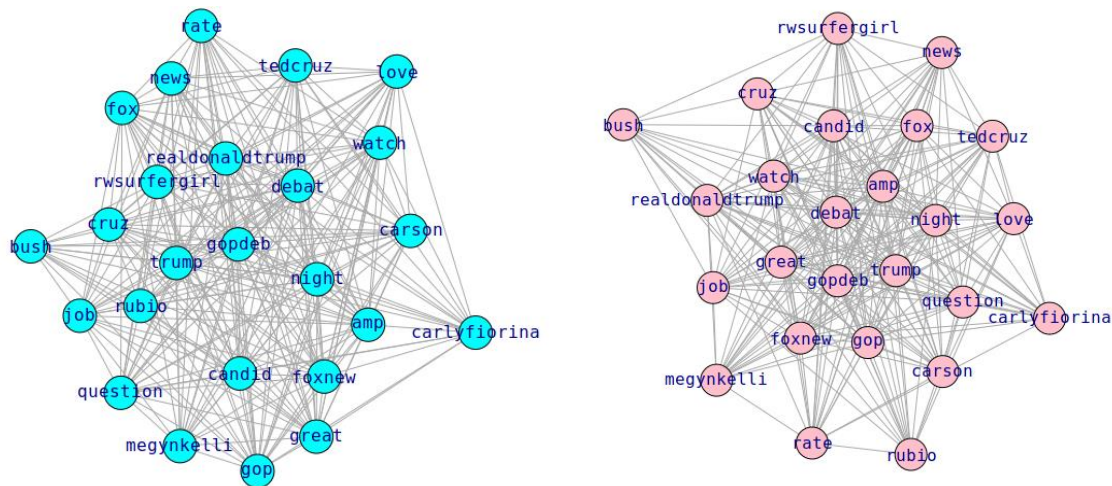


Figure 35. Fruchterman-Reingold and Gem plots showing interconnectedness of terms.

The Star layout arranged the nodes in a circular pattern with “GOPDebate” at the center and other terms radiating outward. This visualization emphasizes the hierarchical structure and the prominence of the central term, since this was used as the most popular hashtag to make posts more popular. Key terms like “realDonaldTrump,” “debate,” and “CarlyFiorina” are directly connected to the central node, indicating their significant role in the discussions. This layout provides a clear view of the main topics and figures that dominated the conversation, highlighting their central importance. While there is so much interconnectedness between each term, one can see that the connections towards the top are thicker while the ones towards the bottom are less dense.

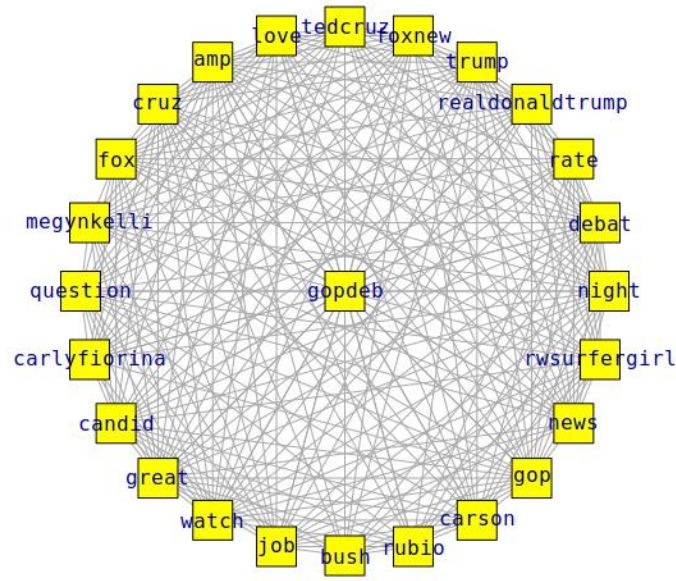


Figure 36. Star plot showing interconnectedness of terms and their relation to central term GOPdebate.

The network plot with the Random layout positions the nodes randomly within the plot area, providing a more chaotic view of the network. Despite the randomness, central terms like “GOPDebate,” “rwsurfergirl,” and “news” still stand out due to their many connections. While this appears similar to the Fruchterman-Reingold and Gem plots, those plots arranged key nodes in clusters in regards to similarity to each other while this plot truly displays a random layout.

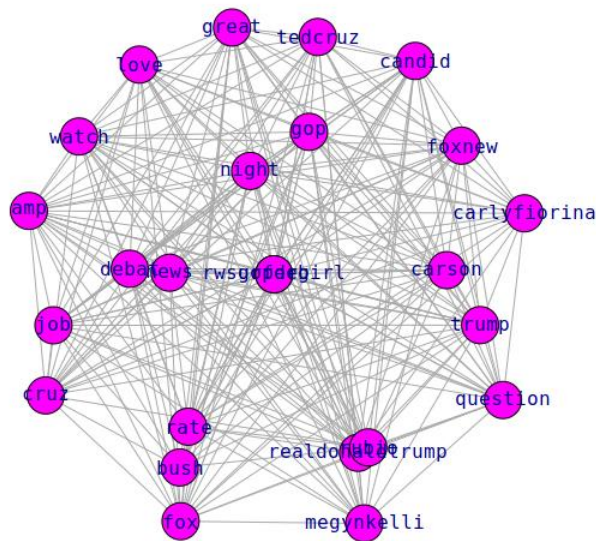


Figure 37. Random layout plot showing interconnectedness of terms and their relation to central term GOPdebate.

The circle layout plot arranged nodes evenly around a circle, emphasizing equal visibility and non-hierarchical organization. It was useful for visualizing the overall connectivity of the network and comparing the number of connections between nodes, though it does not highlight clusters or structural relationships as force-directed layouts do. The primary insights come from examining the edges and connections between nodes, rather than their positions. It's clear from viewing the plot that there appears a similar pattern as the star plot where the bottom right quadrant has less connections as the rest of the plot showing those terms were less used compared to the rest of the terms listed.

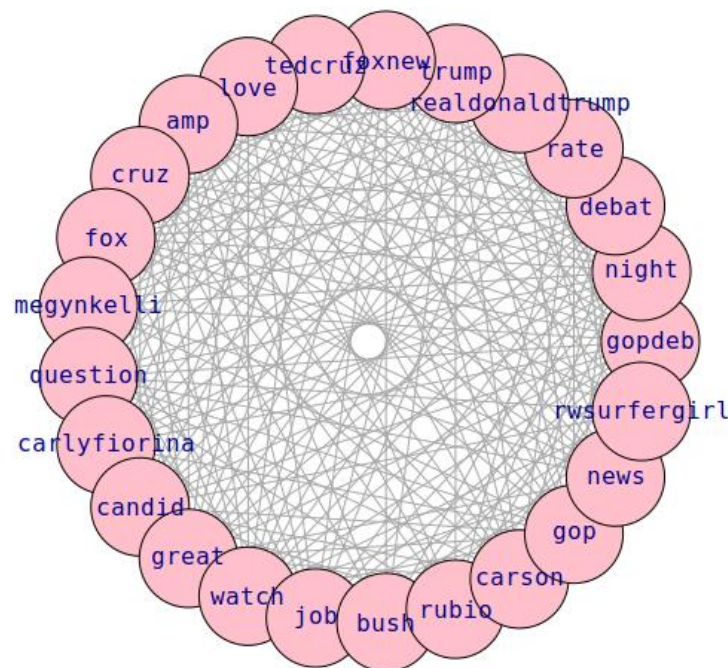


Figure 38. Circle layout plot showing interconnectedness of terms.

Next up, the Nicely layout should have positioned the nodes in a non-overlapping manner, optimizing for readability and visual appeal (though it's immediately clear that there is still some overlap). It is however more readable than the Fruchterman-Reingold, Gem, or Star plots. The central terms were "GOPDebate," "realDonaldTrump," and "debate," with clear connections to other key terms like "rwsurfergirl" and "news." It does not appear to show further insight than its similar predecessor plots.



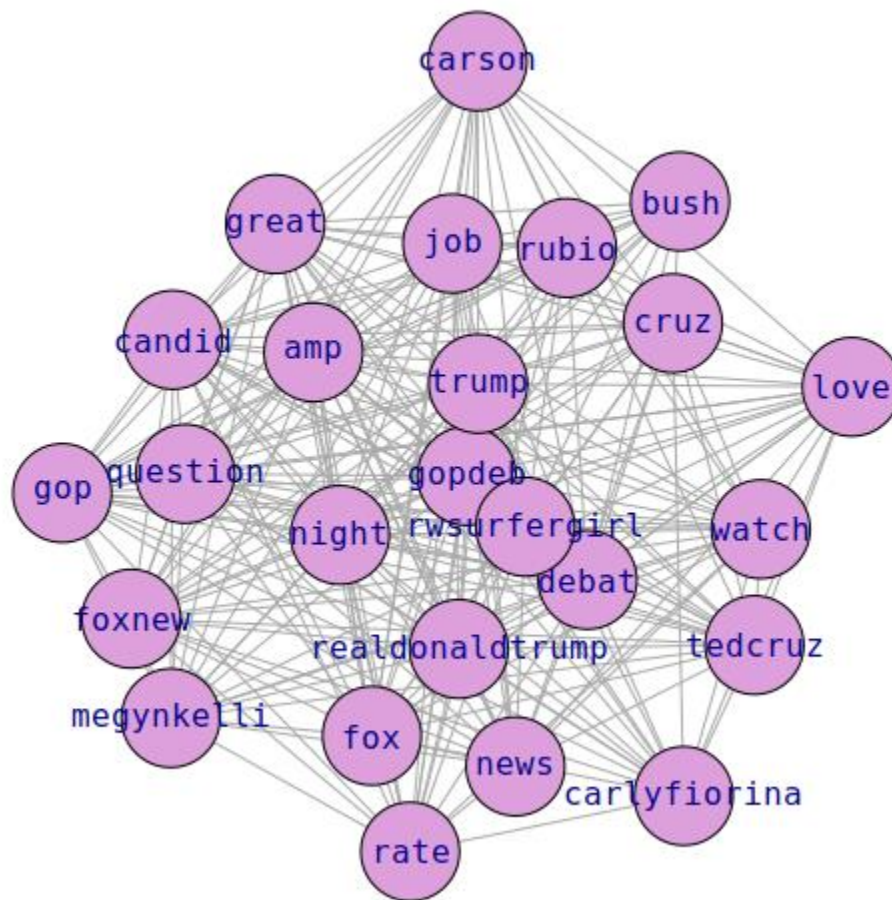


Figure 39. Nicely layout plot showing interconnectedness of terms.

Next, a force-directed layout plot was made, with nodes representing key terms from the GOP debate tweets. Each node was labeled with a term, and the edges between nodes indicated the co-occurrence of terms within the same tweets. The color-coded regions highlighted clusters of terms that frequently appeared together, showing distinct groups of related terms. “Trump” and “GOPDebate” were central, connected to a dense network of terms like “realDonaldTrump,” “debate,” “night,” and “foxnews”. Clusters around specific candidates, such as “CarlyFiorina,” “TedCruz,” and “Bush,” indicated focused discussions on these individuals. The clustering patterns revealed insights into the primary topics of interest and the relationships between

different terms, highlighting how discussions were organized around key figures and themes from the debate. It is notable that “Bush” and “Carson” are on the opposite side of the cluster for “Love”.

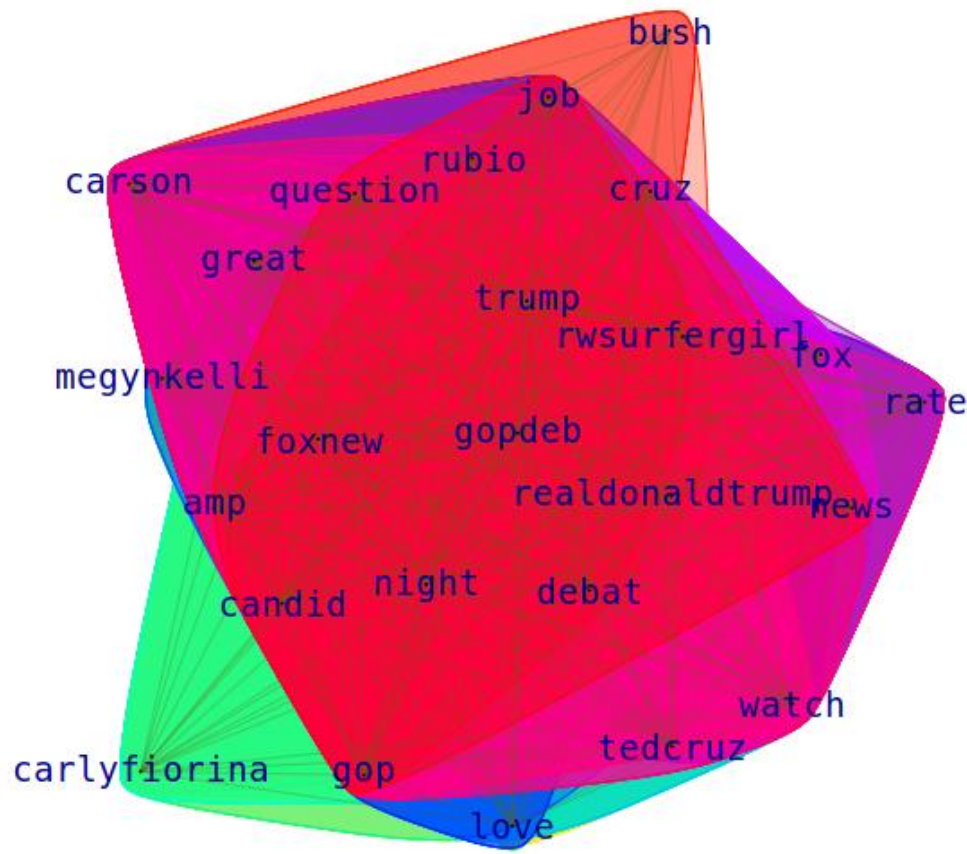


Figure 40. Force-directed layout plot showing interconnectedness of terms with similar nodes being pulled closely together and dissimilar nodes being pushed farther apart.

Next, a cluster plot (clusplot) was generated from a k-means clustering analysis of the term-term distance matrix. Each point represented a term, and the ellipses depicted clusters of terms identified by the k-means algorithm (Syakur et al., 2018). The numbers within the ellipses indicated the cluster numbers assigned by the algorithm. The plot showed several clusters, with each grouping terms that frequently co-occurred in the same tweets. For instance, Cluster 5

included terms like “trump”, “rwsurfergirl”, and “realDonaldTrump” suggesting these terms were often mentioned together. Cluster 3 contained terms like “bush”, “cruz”, “foxnews”, and “rubio” indicating these terms were commonly associated with each other. Notably, the term “gopdeb” appeared to form its own cluster, suggesting it was mentioned frequently but not strongly associated with other specific terms. The density and size of the clusters provided insights into the strength of the relationships among the terms (Syakur et al., 2018). Dense clusters with closely packed points indicated strong associations, while more spread-out clusters suggested weaker or more varied associations. Terms far from the main clusters or in isolated clusters, like “gopdeb”, could be considered outliers, indicating frequent but broad usage across various contexts. Though, it would come as a large surprise that “gopdeb” would be considered an outlier and have such a low relationship with many of the other terms displayed here. It is unfortunate that cluster 2 and 4 are so tightly packed that they are difficult to read making interpretation difficult.

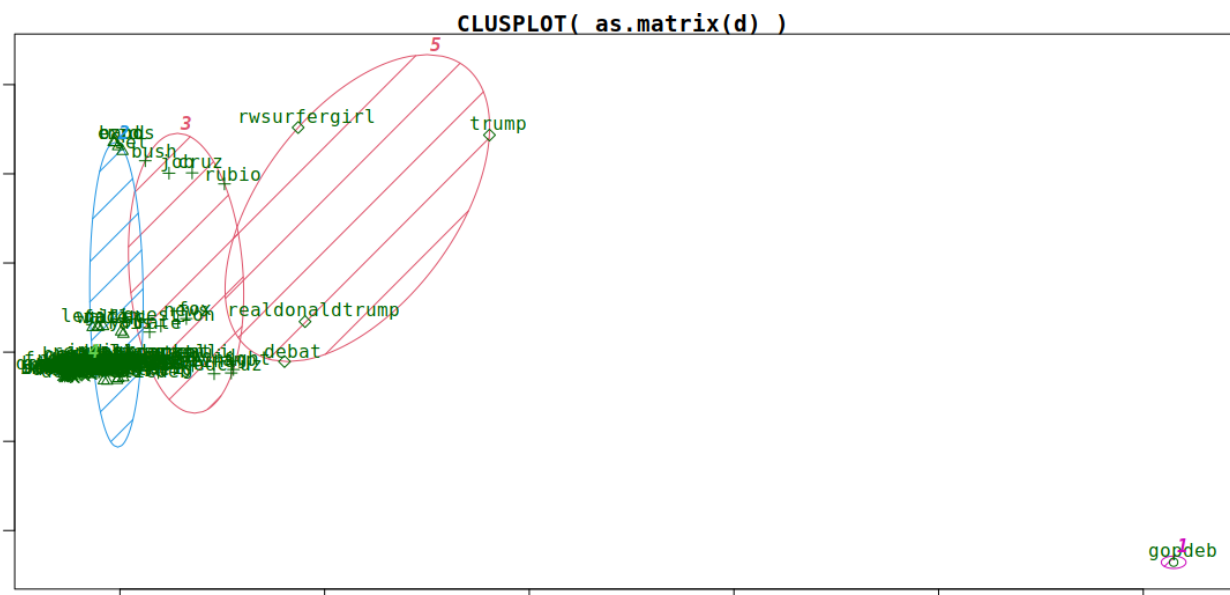


Figure 41. K-means cluster plot showing interconnectedness and co-occurrence of words.



Lastly, an elbow plot was generated. These are used in k-means clustering to determine the optimal number of clusters (Syakur et al., 2018). The x-axis represents the number of clusters, while the y-axis shows the sum of squared distances. Two lines are plotted: the blue line represents the between-cluster sum of squares (BSS), which indicates the variance between the clusters, and the black line represents the within-cluster sum of squares (WSS), which indicates the variance within each cluster. From the plot, we can see that the “elbow” point, where the rate of decrease sharply changes, suggests the optimal number of clusters. In this case, the elbow point appears to be around 3 to 5 clusters. This indicates that using around 3 to 5 clusters provides a good balance between explaining variance between clusters and maintaining low variance within clusters. This plot helps in selecting the most appropriate number of clusters for the k-means algorithm to produce meaningful and interpretable groupings.

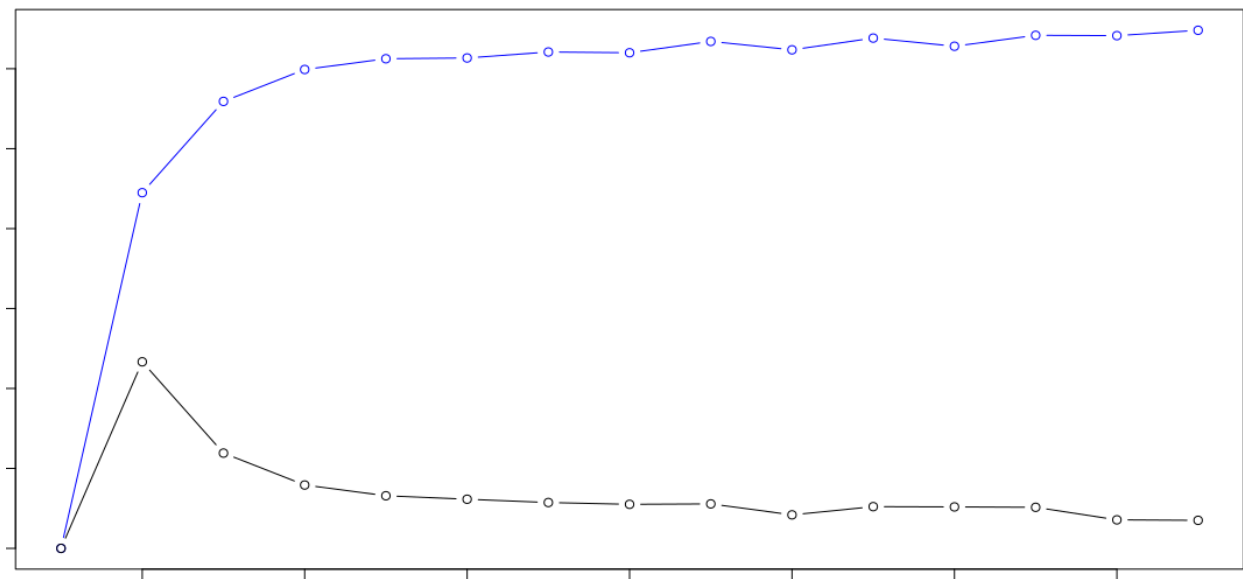


Figure 42. Elbow plot used to determine ideal number of clusters.

### **Discussion:**

The sentiment analysis of tweets related to the first primary GOP debate of 2015 yielded several actionable insights into public opinion and the focal points of the discourse during the

event. The term-term adjacency matrix revealed frequent co-occurrences between key terms, highlighting the central themes and figures discussed. The strong connections between terms like “Trump”, “GOPDebate”, “love”, and “realDonaldTrump” indicated that discussions were heavily centered around Donald Trump and the debate itself. While this is less actionable now, this suggests that any strategic communication or campaign efforts should have instantly considered Trump's significant influence and the central role of the debate in shaping public opinion. Performing this analysis on August 8<sup>th</sup> 2015 using this dataset would have given deep insight into showing how much the public generally enjoyed Trump’s participation in this debate.

The k-means clustering analysis provided deeper insights into the structure of these discussions. Cluster 5, which included “Trump”, “rwsurfergirl”, and “realDonaldTrump”, showed that discussions about Trump were not only frequent but also tied to specific influential users and topics. Cluster 3, with terms like “debate”, “foxnews”, and “rubio” pointed to a segment of the conversation focused on debate performance and media coverage. These clusters suggest targeted communication strategies: focusing on perhaps using influential users for spreading messages about Trump and engaging with media-related discussions to influence perceptions of debate performance. Related to the k-means clustering, the elbow plot helped determine the optimal number of clusters, indicating that around 3 to 5 clusters would balance the variance between and within clusters. It would be interesting to perform the analysis again with only 3 clusters. This optimization ensured that the identified clusters were both distinct and cohesive, providing a clear segmentation of the data. From a strategic standpoint, understanding these distinct clusters allows for more focused and effective messaging, as it identifies the primary areas of interest and concern among the public.

The network plots using various layouts, such as Fruchterman-Reingold, Gem, and circle layouts, provided different perspectives on the relationships between terms. These plots consistently showed the centrality of terms like “GOPDebate” and “Trump” highlighting their pivotal role in the discussions. The Fruchterman-Reingold layout, with its dense connections around central terms, underscored the strong interconnectedness of discussions around Trump and the debate. This insight can guide campaign efforts to maintain a strong presence in these central discussions, leveraging the dense network of connections to amplify their message. It can also be determined which candidates were more enjoyed by the public with stronger associations with “love” being there for “trump”, “rubio”, “cruz”, and “carson” but not for “bush”. This would have been an early predictor that Jeb Bush would not likely advance as a candidate, and conversely that Trump was the favorite after this debate.

The force-directed layout yielded a few interesting insights as well. While it still clustered together some of the most central terms already previously discussed, it also showed some key outliers of the most popular terms. Carly Fiorina was exceptionally popular as a topic however she was noticeably distant from the central cluster. This pattern holds for Carson and Bush clearly giving evidence that they certainly had a niche for the public however they were not nearly as central/popular as Cruz, Rubio, or Trump. “Rate” also appeared much closer to Trump, Cruz, News, but far from Megyn Kelly, Carson, and Fiorina. This also gives insight into how the former are mostly attributable to how the debate was perceived by the public (as in the ratings) whereas the latter were not attributable to the ratings.

Overall, the analysis provided clear, actionable insights into the structure and focal points of public discourse during the GOP debate. By identifying key terms, most discussed candidates, influential users, and the centrality of specific topics, strategic communication efforts can be

better tailored to engage with the most relevant and impactful aspects of the conversation. This targeted approach ensures that messages resonate more effectively with the audience, leveraging the identified clusters and central nodes to maximize influence and engagement.

**Limitations:**

Despite the valuable insights garnered in this analysis, studies like this one have inherent limitations. One significant limitation is the dataset itself. Social media data, particularly from platforms like Twitter, can be unrepresentative of the general population. Twitter users tend to skew younger and more technologically savvy, which may not reflect the broader demographic distribution. Additionally, the dataset might be limited in its variables; for instance, more detailed metadata such as user demographics, tweet geolocation, and interaction metrics like likes and replies could provide a richer context for analysis. The absence of these variables can restrict the depth of insights that can be derived, potentially leading to an incomplete understanding of the data. While sentiment analysis and text mining techniques are powerful, they can sometimes struggle with the nuances of human language, such as sarcasm, idiomatic expressions, or cultural references, leading to potential inaccuracies in sentiment classification. The reliance on predefined lexicons or machine learning models trained on different datasets might not perfectly capture the sentiments expressed in tweets about the GOP debate. Moreover, the analysis primarily focuses on the frequency of terms and basic sentiment categorization, which might overlook the deeper, more complex interactions and influences present in the data.

Furthermore, the study's conclusions are based on data that captures public opinion over only two days, which limits the ability to observe trends and changes in sentiment over time. Analyzing sentiment from multiple debates and tracking how public opinion evolves throughout the entire election cycle would provide more valuable and comprehensive insights. This

approach would help identify long-term trends and shifts in voter perception, offering a deeper understanding of how various factors influence public sentiment over extended periods. Finally, the analysis is limited by the quality of the data preprocessing steps; any accidentally missed errors or omissions in cleaning and preparing the data could lead to biased or inaccurate results. Overall, while the study offers valuable insights, it is important to acknowledge these limitations and consider them when interpreting the findings.

### **Further Research:**

Further research could first improve this study by diving further into the relationships already discovered. It would be interesting to adjust the number of clusters for the k-means clustering approach (adjusting to 3 or 4). It could also focus on understanding relationships between key words better (like the clusters of Trump and rwsurfergirl, or Fox, News, and Rating). It could also delve into more granular aspects of sentiment analysis and text mining to uncover deeper insights from the dataset. Techniques such as topic modeling using Latent Dirichlet Allocation (LDA) could help identify the main themes and issues discussed in the tweets, providing a clearer picture of what topics drove public discourse (Murel and Kavlakoglu, 2024). Additionally, using advanced sentiment analysis techniques, such as emotion detection or aspect-based sentiment analysis, could reveal not just the overall sentiment but also the specific emotions and opinions tied to particular topics or candidates (De Bruyne, 2022). This would allow for a more nuanced understanding of how different candidates were perceived on various issues.

Another promising avenue for further research involves focusing sentiment analysis on the top-performing candidates, such as Trump, Cruz, Kasich, and Rubio. By tracking public sentiment and opinions on each of their debate performances, researchers can identify trends and

shifts in voter perception over time. This targeted analysis could reveal how specific statements, policies, or debate tactics influence public opinion, providing detailed insights into each candidate's strengths and weaknesses. Moreover, given Donald Trump's significant influence and his position as the presumed GOP candidate for the upcoming 2024 presidential election, tracking public sentiment specifically related to him is crucial. Analyzing the public's views on Trump's debate performances, policies, and overall campaign (compared from 2016 to 2024) could provide a comprehensive understanding of his standing with voters. This analysis could be extended to include reactions to his social media activity and public appearances, offering a real-time gauge of public opinion. Such focused research would not only shed light on the dynamics of Trump's support base but also help predict potential outcomes and voter behavior in the forthcoming election.

Expanding the scope of the analysis to include new data could also yield significant insights. For instance, collecting and analyzing tweets from the recent debate between Biden and Trump on June 27th, 2024, could offer a comparative perspective on how public sentiment has evolved over different election cycles. Combining this with other social media platforms, such as Facebook or Instagram, would provide a more holistic view of public opinion. Additionally, conducting network analysis to understand how information and sentiment propagate through social networks could uncover patterns of influence and information spread. Further, integrating demographic data could help identify how different segments of the population perceive and react to political events, offering valuable insights for targeted campaign strategies and understanding voter behavior.

**Conclusion:**

In conclusion, the combined use of SQL via IBM DB2 for EDA and RStudio for sentiment analysis on tweets provides a comprehensive approach to understanding public

opinion and discourse. The SQL queries allow for efficient data validation, cleaning, and initial exploration, ensuring that the dataset is well-prepared for detailed analysis. Several techniques were demonstrated to validate proper data loading. By leveraging the advanced text mining and sentiment analysis capabilities of RStudio, it's possible to gain deeper insights into the cultural and social trends reflected in the tweets. This method of analysis is particularly effective in revealing how the public's attention is distributed among GOP candidates, highlighting key moments that generated significant reactions. Through sentiment analysis, trends and patterns in the public's perception of the candidates can be identified, providing valuable insights into the dynamics of political engagement and the broader cultural context. Overall, this approach demonstrates the power of integrating database management systems and advanced analytical tools to extract meaningful information from large, unstructured datasets like social media posts.

### **References:**

- Berg-Andersson, R. (2024). *The Green Papers*. The Green Papers. <https://www.thegreenpapers.com/P16/R>
- Cornfield, M. (2018). Empowering the party-crasher: Donald J. Trump, the first 2016 GOP presidential debate, and the Twitter marketplace for political campaigns. In *Social Media, Political Marketing and the 2016 US Election* (pp. 6-37). Routledge.
- De Bruyne, L., Karimi, A., De Clercq, O., Prati, A., & Hoste, V. (2022). *Aspect-based emotion analysis and multimodal coreference: A case study of customer comments on adidas Instagram posts*. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (pp. 574-580).
- IBM. (2024a). *Db2*. IBM. <https://cloud.ibm.com/catalog/services/db2>
- IBM. (2024b). *RStudio*. IBM. <https://datapatform.cloud.ibm.com/docs/content/wsj/analyze-data/rstudio-overview>
- Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4), 1093-1113.

Murel, J. and Kavlakoglu, E. (2024). *What is Latent Dirichlet allocation?* IBM.

<https://www.ibm.com/topics/latent-dirichlet-allocation>

Nash, C. (2016). Twitter's Attempt to Silence Generation Trump. Breitbart.

<https://www.breitbart.com/tech/2016/02/08/twitters-attempt-to-silence-generation-trump/>

Podani, J., & Schmera, D. (2006). On dendrogram-based measures of functional diversity. *Oikos*, 115(1), 179-185.

Syakur, M. A., Khotimah, B. K., Rochman, E. M. S., & Satoto, B. D. (2018). Integration k-means clustering method and elbow method for identification of the best customer profile cluster. In *IOP conference series: materials science and engineering* (Vol. 336, p. 012017). IOP Publishing.