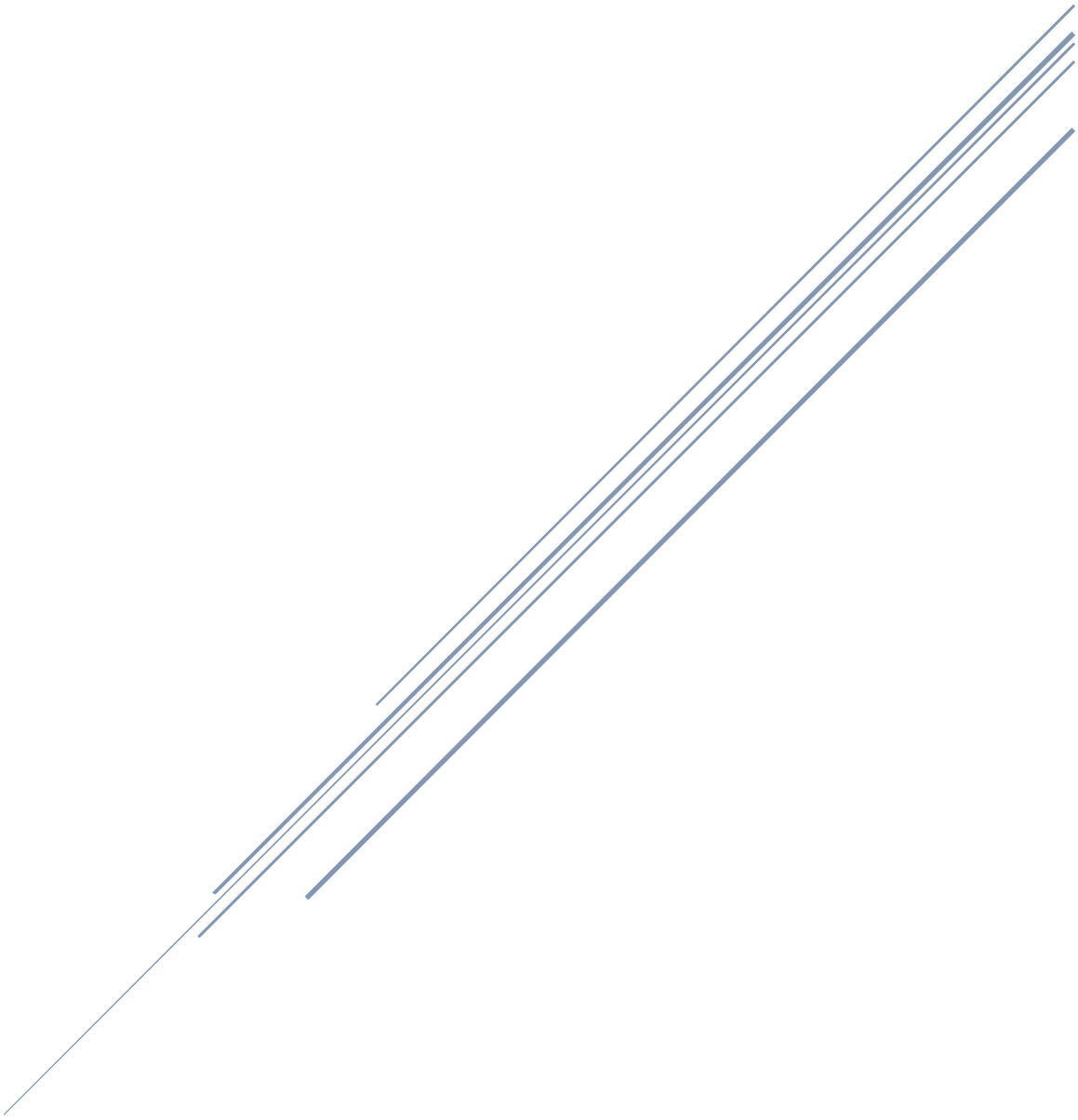
AN INSIDE LOOK AT BARSTOOL SPORTS THROUGH STOOLIE TWEETS

A Twitter Sentiment Analysis



Nicholas Curci
Bryant University

Table of Contents

Abstract:

This report will offer an inside look into Barstool Sports and Stoolie culture by analyzing Tweet sentiment and assigning Sentiment Scores to each tweet. The findings of this report can be used to guide company direction, influence online content, and observe sentiment change over time. The experiments and queries done in this report are not limited and can be replicated at any time. The purpose of this report is not to make predictions but rather to observe how Stoolies think, react, and tweet about Barstool Sports.

Problem Statement:

Born in 2006, Twitter has grown and changed. Today, Twitter is considered America's forum, where news, ideas, thoughts, and even jokes are shared. Twitter is home to over 100 million active daily users who are Tweeting more than 500 million times per day. These statistics are as of 2014. (Hootsuite). For this project I have been tasked with developing, querying, and visualizing insight from twitter data over a two-week time period. I will be doing this to analyze the sentiment of tweets relating to the topic of Barstool Sports or "Barstool" for short. I am challenged with determining if users are tweeting positive, negative, or neutral tweets relating to Barstool. Founded as a free black and white Boston newspaper in 2003, Barstool Sports is now a Manhattan-based digital media company that applies comedy and satire to pop culture, politics, gorgeous women, trending Internet topics and of course, sports (Forbes). Today, Barstool Sports is the homepage to two million (2,000,000) Twitter followers, 400 branded college accounts (yes, there is one for Bryant), and 7.4 million Instagram followers. Followers of Barstool Sports are often ravaging die-hard fans whom traditional media outlets often liken to a cult. These fans

are known as “Stoolies”. Stoolies are often at the heart of Barstool’s most controversial issues. Reports titled; “*Barstool Sports and the persistence of traditional masculinity in sports culture*” (NBC), “*Inside Barstool Sports’ Culture of Online Hate*” (DailyBeast), and “*NFL pulls credentials from Barstool Sports*” (PFT) all point to the massive presence of online “Stoolies” that will protect the Barstool brand at all costs. For this report we will be going to Twitter’s front-line to hear exactly what people are saying on twitter about Barstool Sports.

Hypothesis:

Based on knowledge gained from experience, I expect to find the average sentiment to be positive. I suspect that the majority of users tweeting about Barstool will be the Stoolies who are wary to speak negatively about the company. Although, if there is a current controversy at the time of tweet-collection, then it is very possibly that the overall sentiment could be negative, id suspect this negative sentiment to come from user accounts such as news outlets with many followers that agree with their message. I am also interested to see the content of user tweets, specifically related to pizza. One of Barstool Sports’ main engagement drivers is their pizza reviews. And often, these pizza reviews reflect negatively about the pizza place. This could skew sentiment because a negative sentiment tweet may not be about barstool at all, it may just be about pizza.

Data:

The data for this report was compiled over a two-week period between December 2019 and January 2020 using a python-based tweet listener. This tweet listener scanned twitter for all newly created tweets about “Barstool” and recorded the data about the tweet into a JSON file. A total of 2 GB of tweets were collected. The data that was recorded includes;

create_date – Day and time tweet was created

tweet_id – The ID of the tweet

source – How the tweet was sent

user_screen_name – The user who sent it

user_location – The sender’s location

user_followers – The sender’s followers

user_friends - The sender’s friends

user_language – The sender’s language

user_coordiantes – The sender’s coordinates

quoted_user_name – If it was a quote-tweet then the user who was quoted

retweeted_user_name – If it was a retweet then the user who was retweeted

tweet_language – The language of the tweet

tweet_text – The text of the tweet

During the data collection process, the was an unexpected error. The cause of this error is unknown. This error resulted in one day of collection data to be unusable due to internal JSON errors. Overall, an estimate of 10% of data was lost due to these unknown errors. However, the

good news is that the project and report was still able to be completed with the data that was uncorrupted.

Discussion of Findings:

A NOTE BEFORE DISCUSSION OF FINDINGS:

After going through the data dictionary and learning what fields I had to use, I decided to investigate three different areas of my data. In the first area, I wanted to examine timetable data. This includes data about days of the week and time of day. In the second category, I wanted to investigate the users. And in the third category I decided to investigate sentiment. For these three categories I developed 10 unique queries that would help be understand and analyze this data. The questions and categories that I developed and the analysis of them are as follows:

Category 1: Date/Time

1. **On which weekday are the most tweets created?**
    a. This was one of the most simple queries that I could have run and surprisingly, it gave me some of the most interesting data. Of the 266,000+ tweets that were collected, over 57,000 came from Wednesday. This proved to be over 10,000 tweets higher than the next closest day (Thursday). Surprisingly, Saturday proved to hold the least number of tweets at 21,360. Weekdays attributed to over 80% of all tweets collected.
2. **What time of day are most tweets sent?**
    a. This information can be crucial to a business and their social media operations. Knowing when users are online and engaged with their brand can help boost promotions, content, and engagement. To do this I had to create a new time table

and substring out the time values from create_date. After doing this I was able to assign an hourly time to each tweet and count and group them more easily. After doing this I created the visualizations in tableau and found that most tweets about Barstool are spent in the evening, between the hours of 10pm and 4am. This seems to coincide with Barstool's main demographic, young-adults, who are often up later in the night and less active during the day. There is also a spike in tweets with the simple max coming at 4pm.

3. **Does day of the week affect sentiment?**
    a. Coming full circle from the query one results, this query attaches a sentiment to the tweets grouped by weekday. The results of these followed the results of query one with Thursday holding the highest sentiment near 2.0 and Monday holding the only negative sentiment. This low sentiment could be due to a negative event that happened on Monday that went viral or simply just it being Monday. To make this query better more data is needed, and this data needs to be clean and uncorrupted.

Category 2: Users

1. **Is one user responsible for many tweets, is this skewing the data?**
    a. I wanted to see if one user was responsible for creating many tweets about Barstool. But before I did this, I needed to filter out the "Barstool Sports" main twitter page as it was showing up frequently in my results. After filtering this out I did a simple count of tweeted and grouped by the username. I found that even over the span of just a couple of weeks, there were a few users who were tweeting significantly more than others. User "TalibKweli" tweeted over 3x more often

than the next highest, “OldManKirk1”. “DodgeRamOwner” was also among the most tweeting users.

2. **Does the number of followers a user have relate to their tweet sentiment?**
    a. I wanted to ask this question to see if popular accounts were speaking highly or lowly of Barstool. I found that users with a high number of followers tweeted with the highest sentiment. These could be celebrities or other people of interest. I also found that those with a very low number of followers spoke the least positively of Barstool, with an average sentiment score of .42 compared to the 1.03 of users with high-level followers. To do this query I created a follower category and joined in the sentiment of tweets from the user, then grouped by the follower category.
3. **What sentiment are highly active users giving?**
    a. To do this query I found who my users are with the highest engagement, for this example I took the top 6. I then added in their number of tweets and the average sentiment for these tweets. Once again, the “TalibKweli” username came up, leading me to suspect this use to be a major critic of Barstool Sports. I found that even though these users may be tweeting about Barstool more often than others, their message may not be positive since only 1 of the top 6 users had a positive average sentiment score. I suspect that the negative users could be either online trolls or employees of competing companies.

Category 3: Sentiment

1. **Do longer tweet texts have higher sentiment?**

   a. This was one of the questions that I was most excited to answer. To answer this, I first grouped tweet length into categories “Short”, “Medium”, and “Long”. Then I grouped sentiment into groups “Positive”, “Neutral”, and “Negative”. After grouping these I counted the number of tweets that were in each category and found that there was only a small difference in the number of positive and negative tweets in the category of Short and Medium. Medium and Long length tweets tended to be more positive while Short tweets were near equal. Neutral tweets looked to be a non-factor except for Medium Length tweets.

2. **What is the sentiment of pizza tweets vs non-pizza tweets?**
   a. This question gave me the most surprising results. Usually pizza reviews are negative. It is very uncommon to see President Dave Portnoy offer a glimmering review for a pizza. This would lead one to believe that the sentiment around these tweets would be negative. This was not the case. Tweets that mentioned “pizza” rated very high on the sentiment scale, over .80 sentiment points while regular tweets were just higher, closer to .90. I am still unsure what to make of these results but can offer an educated guess as to the lower sentiment is users defending their local pizza place against the review of President Portnoy.
3. **Do quote-tweets or retweets offer higher sentiment?**
   a. I wanted to see which form of reposting yielded higher sentiment. I did this by making two queries, one with the result of quote tweet sentiment and the other with the retweet. I expected to find quote tweets with higher sentiment,

because they offer the user a more personalized message and my hypothesis was incorrect. Quoted tweets offered a wicked high sentiment score of .23 while retweets were only .003. Like the last query, I found this my making two queries and joining them together in tableau to visualize the results.

4. **Do different geographic locations offer starkly different sentiment?**
    a. I'll be honest, this is the one that I was most excited to see. I wanted to see if Barstool was seen in a positive light nationally. To execute this query, I grouped tweets with similar locations and averaged their sentiment scores. I limited this query to U.S. Cities and States and took the top 7 cities with the most tweets. Surprisingly, I found that Barstool holds a positive sentiment for users in Boston, California, Chicago, and Houston. Philly was the only major city with Negative sentiment. This is most could be due to recent controversies with New England and Philly sports teams as well as underperforming Philly sports teams.

These queries show just the surface of how powerful Twitter data can be. Even with only minimal data I was able to derive invaluable insight for a company that relies on social media to operate. Now that I know how this data works and is held, it will make future data analysis more seamless and conjure an even higher level of integration and insight.

Limitations:

A massive limitation to this study was the limited amount of data. Working with under 300,000 records resulted in a small sample size of data. This small sample size opens the door to

many issues. The biggest of which is replication. Since the data collected was primarily from weekdays, then it is possible that data collected over a longer period would return drastically different results. However, I believe that the data that was used for this report offers an honest snapshot into the daily sentiment of Barstool. Not every day can be a high-content day with highly positive sentiment. If the data was collected on only a Friday or a Saturday then I believe it would be heavily skewed positive.

Another limitation of this report was the time. Two weeks is simply not enough time to study a growing company like Barstool. With new followers and new content delivered every day, it is impossible to get an accurate definition of sentiment over only two weeks.

A third and final limitation was software. It was my first time using python to tweet-listen, first time using Jupyter Notebooks to collect data, first time using Hive, and only second time using Tableau. Using these foreign softwares resulted in an incredibly steep learning curve that was not completely surpassed in the time period allotted for this project. While there were not mistakes made in using these softwares and the integrity of this study remains firm, I did find it difficult to learn and adapt to these softwares on-the-fly.

Future Direction:

In the future, if I were to replicate this study, I would focus on three things. The first of which would be Data Collection. A successful and meaningful project relies on strong data. For a replication of this project I would put a heavy emphasis on strong data collection and making sure that files remained uncorrupted and held their integrity.

The second aspect of this project that I would alter if I were to replicate it would be the timeline. I would start the data collection process much earlier, allotting double or triple the time to collect data and apply stronger collection practices to keep the data clean.

The third and final facet of this project that I would update would be personal to my software experience. Before embarking on a replication of this project I would spend more time practicing how to use the different features that this project demands. Possibly create smaller, practice, projects before trying my hand at this one again. LinkedIn Learning is also another great tool that I would utilize before trying this project again.

Overall this project generated some fascinating insights that I was not expecting to find. The application of these insights to a company like Barstool Sports I believe to be invaluable. Using these findings, I believe that Barstool can better situate and prepare themselves for the future while they stay on the frontlines of America's Forum, standing as one of the most impactful, influential, and controversial companies in the United States.

Amazon Web Services, Hive, and Pig:

AWS (Amazon Web Services) is a comprehensive, evolving cloud computing platform provided by Amazon that includes a mixture of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings. AWS services can offer an organization tools such as compute power, database storage and content delivery services.

Hive is an open-source, data warehouse, and analytic package that runs on top of a Hadoop cluster. Hive scripts use an SQL-like language called Hive QL. Hive extends the SQL paradigm by including serialization formats. You can also customize query processing by

creating table schema that match your data, without touching the data itself. In contrast to SQL (which only supports primitive value types such as dates, numbers, and strings), values in Hive tables are structured elements, such as JSON objects.

Apache Pig is an open-source Apache library that runs on top of Hadoop, providing a scripting language that you can use to transform large data sets without having to write complex code. Pig works with structured and unstructured data in a variety of formats. You can also execute Pig commands interactively or in batch mode.

For this project I was tasked with querying my tweets through Amazon Web Services. To do this, I first created an AWS account and linked to the ISA 360 classroom. After linking to this classroom, I created an S3 bucket. Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web. I then created folder for input, jars, logs, output, and scripts that I then filled with eh appropriate files from the ones given to me on Blackboard. After creating an S3 bucket, I changed the output in my PIG script to reflect the new destinations (input, scripts, and jars) as well.

Following the creating of my S3 bucket, I then went to services and launched the EMR. Amazon Web Services EMR (Elastic Map Reduce) is their proprietary tool for big data processing and analysis. Within the EMR I then migrated to Step Execution and added the proper Script S3 destinations for location, input, and output for the pig file that I planned to run using the default software configurations.

After setting up the EMR steps, I then went to clusters and created a unique cluster. A cluster is just a logical grouping of tasks or services. From the cluster I went to my S3 output folder and downloaded the results of my query. Now I am ready to visualize my data!

To start, I created a folder on my desktop to store my results. I then downloaded the SQL script with my queries and made sure that all of my locations were correct. Then, on Amazon Web Services, I went back to my bucket output that I created earlier to see the location of the results. Then I downloaded the results (q0, q1, q2, q3, q4, q5) one by one and put them in the desktop folder that I just created.

Once all of the files were downloaded, I then created an Excel spreadsheet and imported the results for analysis. I saved the excel file into the same folder under the file name tweets_analysis. Each sheet became home to its own query (or a combination of queries). I created labels for each query and imported the data through excels import feature. For language I did something special, I imported the twitter language code book found on their developer's website and then cross referenced them to my excel data using a V lookup. After adding in the language name, I decided to sort my languages that were unknown and got rid of anything that was not supported by twitter. Just to do a quick visualization of this V lookup, I created a pie chart which I have included below. The pie chart is just one of many visualizations that I created while experimenting with AWS data and the visualization of this data in Excel. After creating these graphs, I shut down my instances on AWS and double checked that everything was saved correctly.

Even though my sample size was small, Amazon Web Services allows me to execute my queries and download my data much faster than if I was to use HiveSQL. What would have taken me over 7 minutes per query with Hive executed significantly faster on AWS. (7 minutes vs 44 seconds). While AWS can be a bit tricky to understand at first, once a user is comfortable with the environment and executing queries, data can be queried, processed, and visualized much

faster than using a local or even a VM cloud system. This can save an organization or user time, money, and resources.

Overall, this project allowed me the opportunity to explore interesting and business critical data while also learning and experimenting with new collection methods (Python tweet streaming through Jupyter Notebooks and Twitter Developer), data querying methods (Amazon Web Services, and Hive SQL/Cloudera), along with different visualization methods (Tableau and Excel). This experience has strengthened my skills in these fields and I am looking forwards to further exploration and development in these skills.

Appendix:

This appendix includes queries and graphs that are and will be used for the data visualization portion of this project. Each query is important in the grand scheme of this project.

Create Barstool_Tweets Table

```
CREATE EXTERNAL TABLE IF NOT EXISTS barstool_tweets (create_date STRING, tweet_id
STRING, source STRING, user_screen_name STRING, user_location STRING, user_followers
INT, user_friends INT, user_language STRING,
user_coordiantes STRING, quoted_user_name STRING, retweeted_user_name String,
tweet_language STRING, tweet_text String)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
LOCATION '/user/tweetsanalytics1/output';
```

Create B_Tweets Table

```
create table b_tweets as
select w.tweet_id as tweet_id, avg(a.rating) as sentiment_score,
      case when avg(a.rating)>0 then 'Positive'
          when avg(a.rating)=0 then 'Neutral'
          else 'Negative'
          end as sentiment
from (select tweet_id, tweet_language, single_word
    from barstool_tweets b
    LATERAL VIEW explode(split(lower(tweet_text),' ')) adTable AS single_word) w
 inner join afinn a
 on (w.single_word =a.word)
 group by w.tweet_id;
```

Create Followers Table

```
Create table Followers as
Select tweet_id, user_screen_name, user_followers, Case
When user_followers < 100 then 'very low followers'
When user_followers between 101 and 1000 then 'low followers'
When user_followers between 101 and 1000 then 'medium followers'
When user_followers between 1001 and 10000 then 'high followers'
When user_followers > 10000 then 'influencer followers'
Else 'unknown number of followers'
end as FollowerCategory
From barstool_tweets;
```

Create Tweet_Length Table

```
create table tweet_length as
Select
case
when length(tweet_text) between 0 and 62 then 'Short'
when length(tweet_text) between 63 and 115 then 'Medium'
when length(tweet_text) > 115 then 'Long'
Else 'Broken'
end as LengthCategory, bar.tweet_id, sentiment
from barstool_tweets bar
inner join b_tweets b on bar.tweet_id = b.tweet_id;
```

Create Bar_Tweets Table

```
create table bar_tweets as
select w.tweet_id as tweet_id, avg(a.rating) as sentiment_score,
      case when avg(a.rating)>0 then 'Positive'
          when avg(a.rating)=0 then 'Neutral'
          else 'Negative'
          end as sentiment
from (select tweet_id, tweet_language, single_word
    from barstool_tweets b
    LATERAL VIEW explode(split(lower(tweet_text),' ')) adTable AS single_word) w
 inner join afinn a
 on (w.single_word =a.word)
 where w.tweet_language= 'en'
 group by w.tweet_id;
```

Create DayOfTheWeek Table

```
create table dayoftheweek as
select tweet_id, substr(create_date,0,3) as DayOfWeek
from barstool_tweets;
```

Create Time Table

```
Create table time as
select tweet_id,substr(create_date,12,8) as tod
from barstool_tweets;
```

Create TimeTable Table

```
create table timetable as
select tweet_id,
case
when tod like "00:%" then "24"
when tod like "01:%" then "1"
when tod like "02:%" then "2"
when tod like "03:%" then "3"
when tod like "04:%" then "4"
when tod like "05:%" then "5"
when tod like "06:%" then "6"
when tod like "07:%" then "7"
when tod like "08:%" then "8"
when tod like "09:%" then "9"
when tod like "10:%" then "10"
when tod like "11:%" then "11"
when tod like "12:%" then "12"
when tod like "13:%" then "13"
when tod like "14:%" then "14"
when tod like "15:%" then "15"
when tod like "16:%" then "16"
when tod like "17:%" then "17"
when tod like "18:%" then "18"
when tod like "19:%" then "19"
when tod like "20:%" then "20"
when tod like "21:%" then "21"
when tod like "22:%" then "22"
when tod like "23:%" then "23"
else "unknown"
end as tweet_time
from time;
```

Create Stop_words Table

```
CREATE EXTERNAL TABLE IF NOT EXISTS stop_words (
word STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '/n'
LOCATION '/user/tweetsanalytics5/stop_words'
TBLPROPERTIES ("skip.header.line.count"="1");
```

Create Afinn Table

```
CREATE EXTERNAL TABLE IF NOT EXISTS afinn (
word STRING, rating INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
LOCATION '/user/tweetsanalytics5/affin'
TBLPROPERTIES ("skip.header.line.count"="1");
```

Hive SQL Queries:

Category 1:

1. **What days create the most tweets?**

```
select count(tweet_id) as numberoftweets, dayofweek
from dayoftheweek
group by dayofweek
order by numberoftweets;
```

2. **Time of day when most tweets are sent?**

```
select count(tweet_id) as numberoftweets, tweet_time as timeofday
from timetable
group by tweet_time
order by numberoftweets desc;
```

3. **Does day of the week affect sentiment?**

```
select avg(sentiment_score) as avgscore, dayofweek
from b_tweets b
inner join dayoftheweek d on b.tweet_id = d.tweet_idgroup by dayofweek;
```

Category 2:

4. **Is one user responsible for many tweets?**

```
select user_screen_name, count(tweet_id) as numoftweets
from barstool_tweets
where user_screen_name not like "Barstoolsports"
group by user_screen_name
order by numoftweets desc
limit 15;
```

5. **Does higher followers = higher sentiment?**

```
select avg(sentiment_score), followercategory
from b_tweets b
inner join followers f on f.tweet_id = b.tweet_id
group by followercategory;
```

6. **What sentiment are highly active users giving (must have more than 2 tweets in time period)?**

```
select avg(sentiment_score) as sentscore, user_screen_name, count(b.tweet_id) as numoftweets
from b_tweets b
inner join followers f on f.tweet_id = b.tweet_id
group by user_screen_name
order by numoftweets desc, sentscore desc
limit 7 ;
```

Category 3:

7. **Do longer texts have better sentiment?**

```
Select count(tweet_id) as numoftweets, sentiment, lengthcategory
from tweet_length
group by sentiment, lengthcategory
order by numoftweets desc;
```

8. **Pizza sentiment?**

```
select avg(sentiment_score) as pizzasentscore
from b_tweets b
inner join barstool_tweets bar on b.tweet_id = bar.tweet_id
where tweet_text LIKE '%pizza%';
```

```
select avg(sentiment_score) as regular
from b_tweets b
inner join barstool_tweets bar on b.tweet_id = bar.tweet_id;
```

9. **Do quotes or retweets have higher sentiment?**

```
select avg(sentiment_score) as QuotedTweetSentimen
from barstool_tweets bar
inner join b_tweets b on bar.tweet_id = b.tweet_id
where quoted_user_name != ''
and retweeted_user_name = '';
```

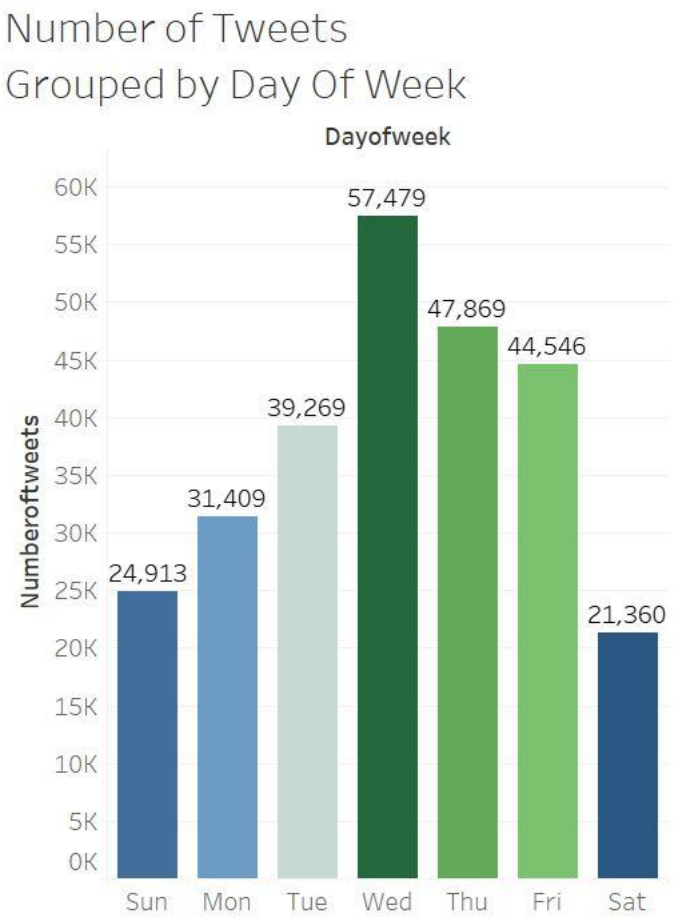
```
select avg(sentiment_score) as ReTweetSentiment
from barstool_tweets bar
inner join b_tweets b on bar.tweet_id = b.tweet_id
where retweeted_user_name != ''
and quoted_user_name = '';
```

10. Geographic sentiment?

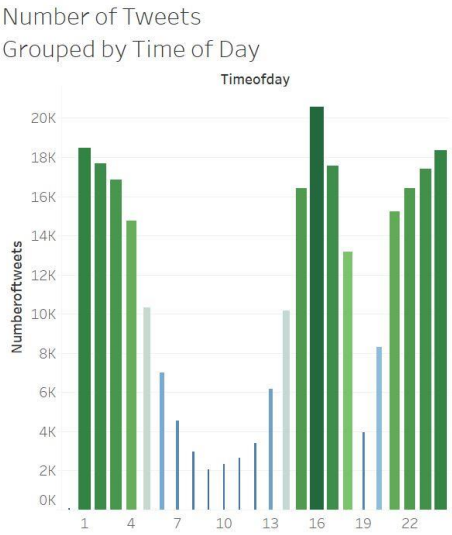
```
select user_location, avg(sentiment_score) as avgsent, count(bar.tweet_id) as numoftweets
from barstool_tweets bar
inner join b_tweets b on b.tweet_id = bar.tweet_id
group by user_location
order by numoftweets desc, avgsent desc
limit 10;
```

Tableau Graphs:

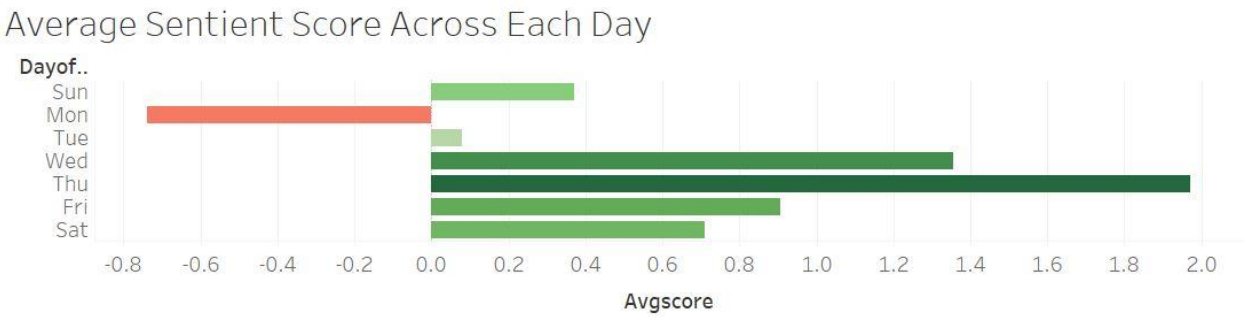
Query 1:



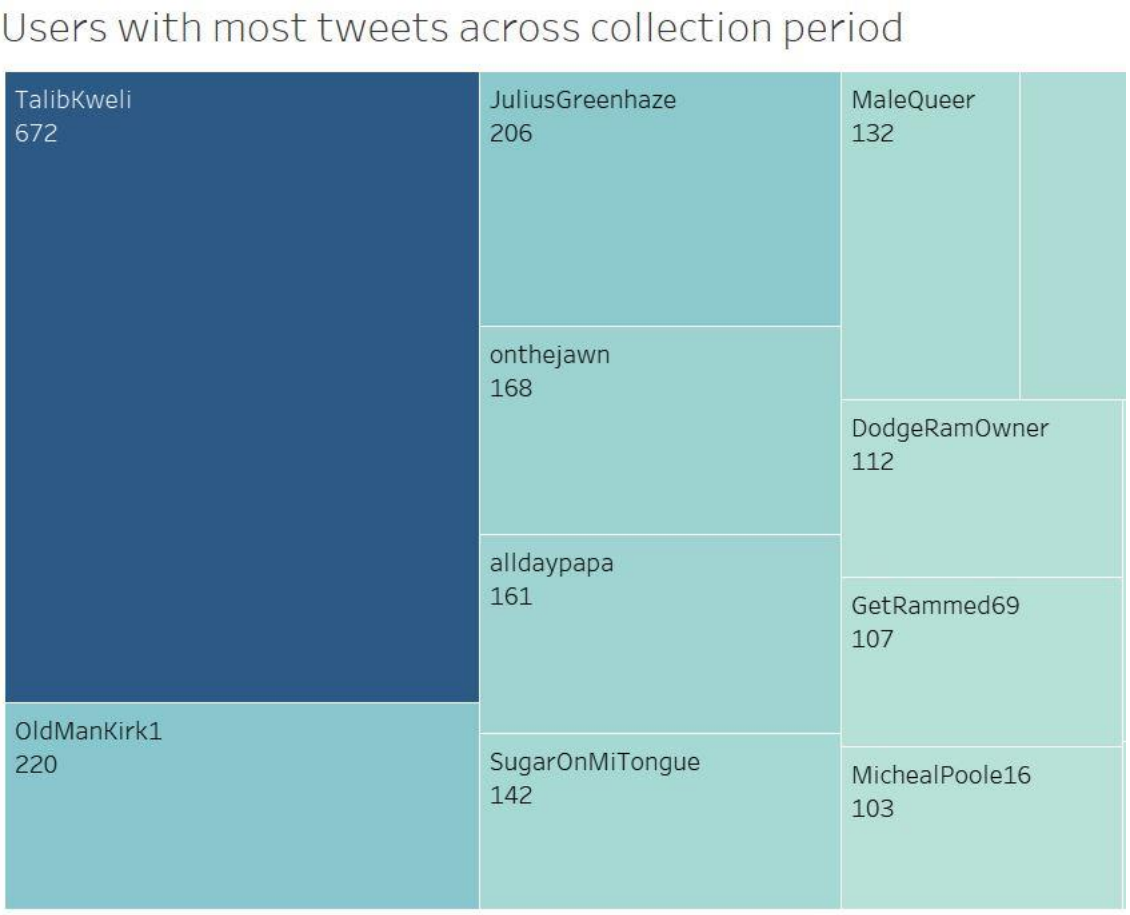
Query 2:



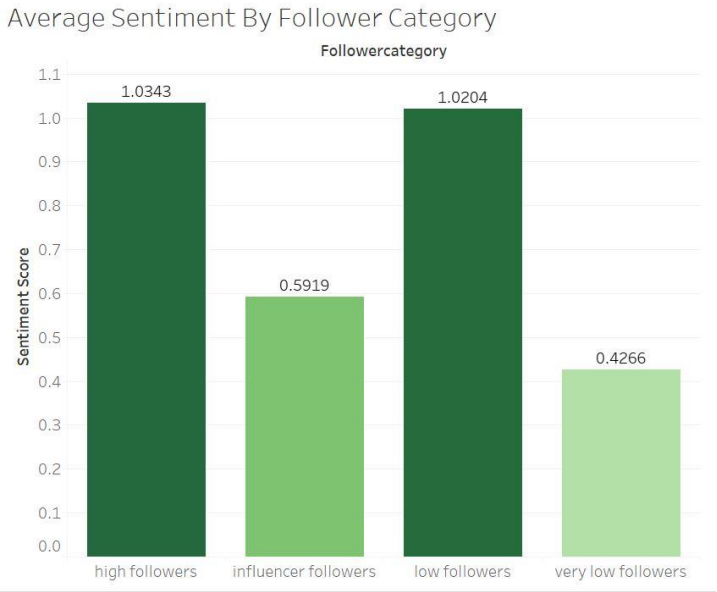
Query 3:



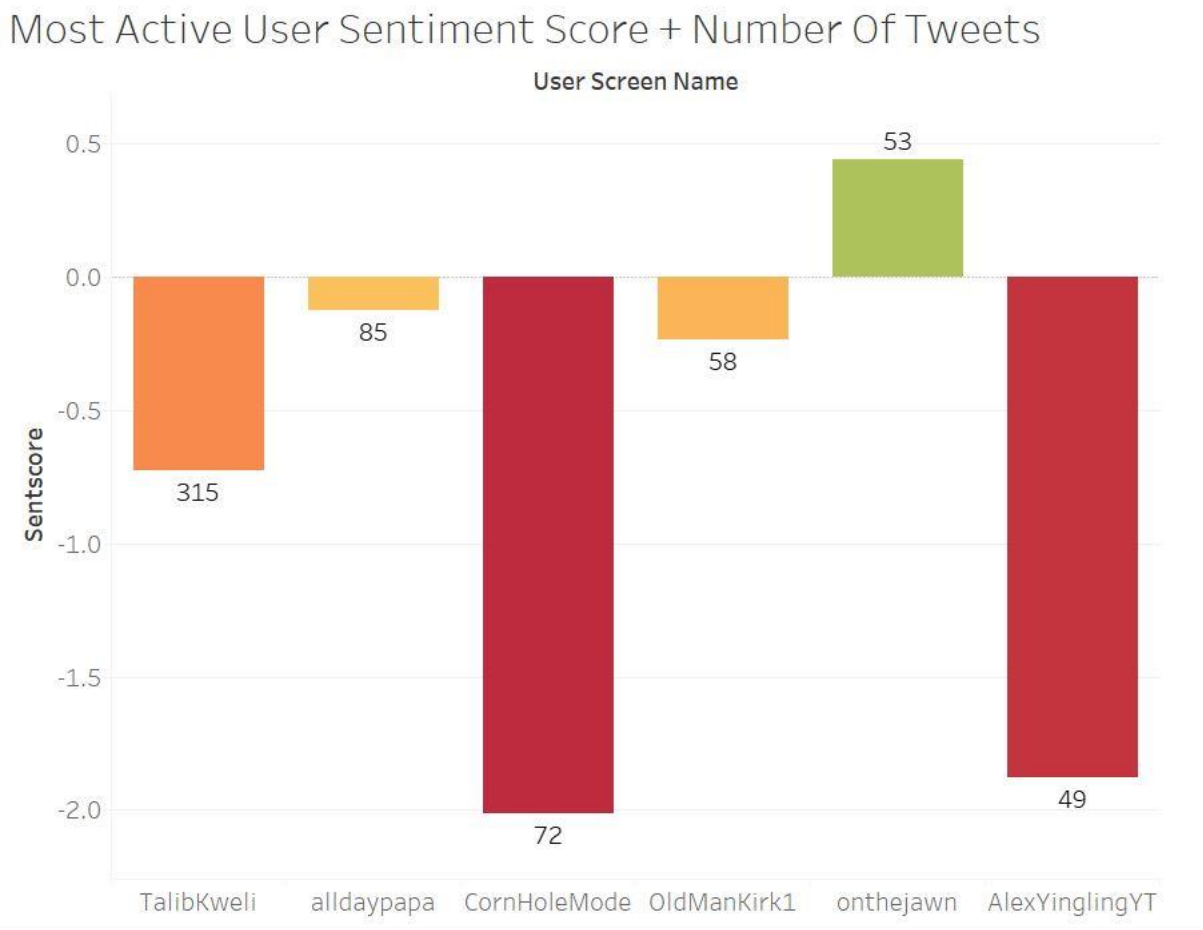
Query 4:



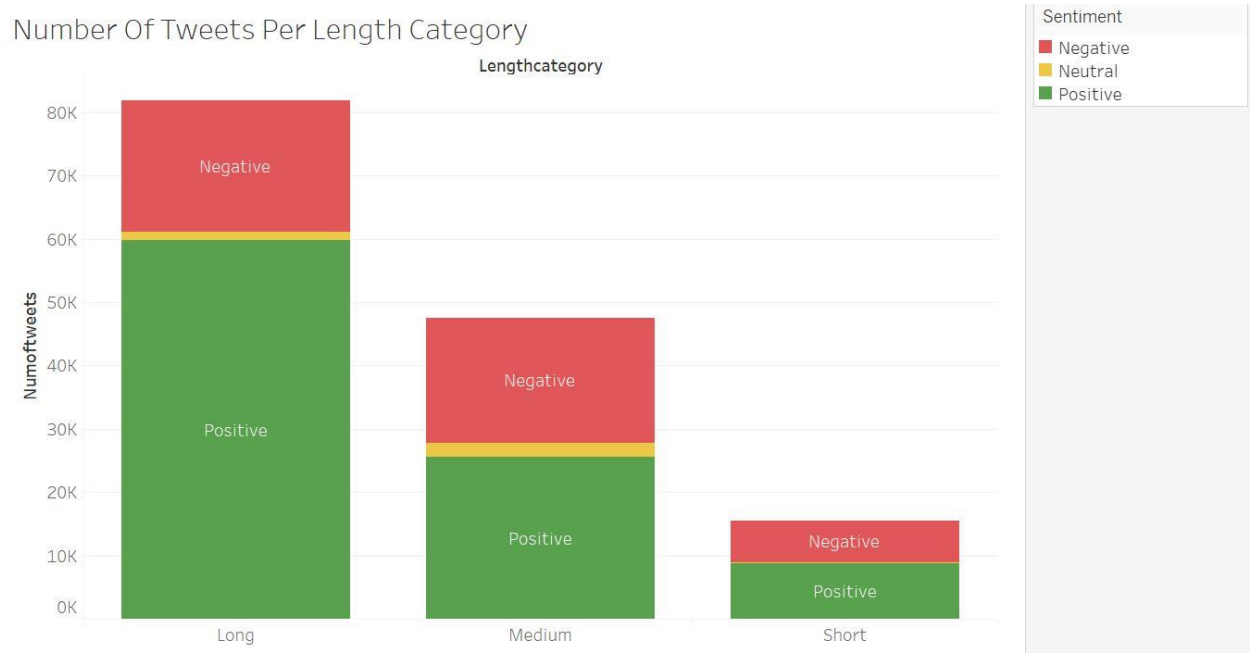
Query 5:



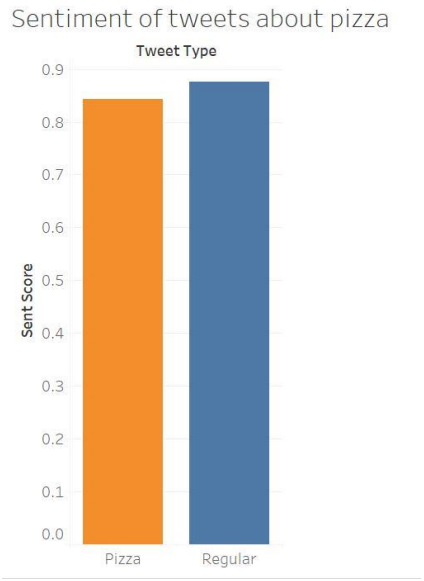
Query 6:



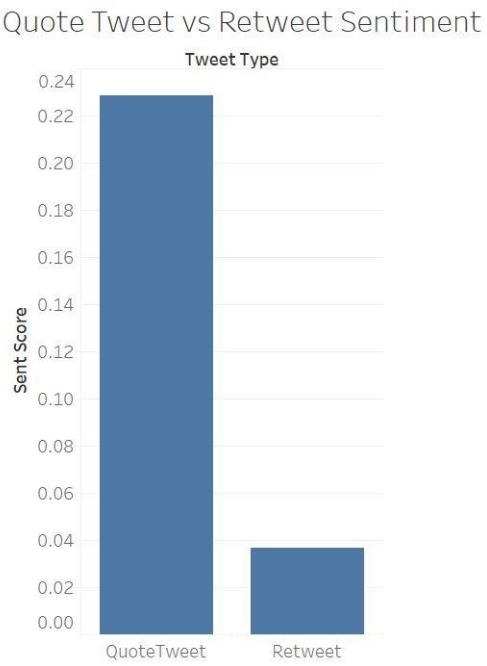
Query 7:



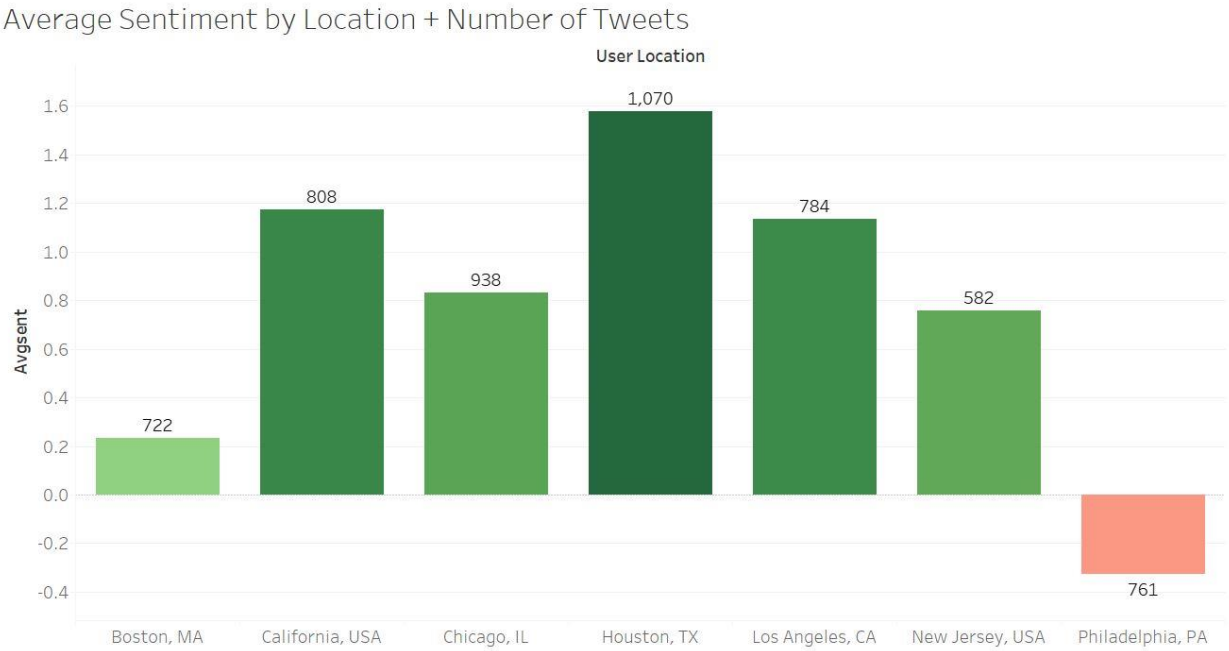
Query 8:



Query 9:



Query 10:



Amazon Web Services Queries:

There were 6 queries run to sample the data from the time period. They are as follows:

Query 0:

This is the original Table that we will create and use:

CREATE EXTERNAL TABLE IF NOT EXISTS barstool_tweets (create_date STRING, tweet_id STRING, source STRING, user_screen_name STRING, user_location STRING, user_followers INT, user_friends INT, user_language STRING,
user_coordiantes STRING, quoted_user_name STRING, retweeted_user_name String, tweet_language STRING, tweet_text String)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
LOCATION 's3://ncurcibarstool/output/barstool_tweets';

1. **How many tweets have been collected about Barstool?**

INSERT OVERWRITE DIRECTORY
's3://ncurcibarstool/output/barstool_results/q0_TotalTweets'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
select count(*)
from barstool_tweets;

Our sample data yields us with 4234 tweets about Barstool.

2. **How many unique users tweeted about barstool?**

INSERT OVERWRITE DIRECTORY 's3://ncurcibarstool/output/barstool_results/q0_TotalUsers'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
select count(distinct user_screen_name)
from barstool_tweets;

Our sample data yields us with 946 users. Which tells us that users are tweeting multiple times about barstool.

Query 1:

What are the number of tweets by type (original tweets, retweets, or quoted tweets)?

```
INSERT OVERWRITE DIRECTORY
's3://ncurcibarstool/output/barstool_results/q1_tweetsByType'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
Select "Original", count(*) as Numoftweets
from barstool_tweets
where quoted_user_name="" and retweeted_user_name=""
union all
Select "Retweet_only", count(*) as Numoftweets
from barstool_tweets
where retweeted_user_name<>"" and quoted_user_name=""
union all
Select "Quoted_only", count(*) as Numoftweets
from barstool_tweets
where quoted_user_name<>"" and retweeted_user_name=""
union all
Select "Both Retweet&quoted", count(*) as Numoftweets
from barstool_tweets
where quoted_user_name<>"" and retweeted_user_name<>"";
```

There have been 946 Original Tweets, 2531 Retweet Only, 577 Quote Only, and 645 both Retweet and Quote tweeted. This tells us that most users are retweeting things related to Barstool.

Query 2:

What are the number of tweets by language?

```
INSERT OVERWRITE DIRECTORY
's3://ncurcibarstool/output/barstool_results/q2_tweetsByLanguage'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
select tweet_language, count(*) as NumofTweets
from barstool_tweets
group by tweet_language
order by NumofTweets desc;
```

The most dominant language was English with 4574 tweets. Dutch and Russian also had tweets.

Query 3:

What are the top 100 hashtags in barstool topic (limit to the tweets in English, remove barstool hashtag).

```
INSERT OVERWRITE DIRECTORY
's3://ncurcibarstool/output/barstool_results/q3_tophashtags'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
select temp.word, count(*) as NumberofWords
from
(select explode(split(lower(tweet_text)," ")) as word
from barstool_tweets
where tweet_language='en')temp
where instr(temp.word,"#")==1 and temp.word <> '#barstool'
group by temp.word
order by NumberofWords desc
limit 100;
```

#barstool and #spittinchicklets were two of the most popular hashtags used in conjunction with the keyword Barstool with 19 and 10 tweets respectively. SpittinChicklets is a podcast by Barstool Sports.

Query 4:

What are the top 100 words in barstool topic? (limit to the tweets in English, remove stopwords, all hashtags)

```
CREATE EXTERNAL TABLE IF NOT EXISTS stopwords (
word STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '/n'
LOCATION 's3://ncurcibarstool/input/stopwords'
TBLPROPERTIES ("skip.header.line.count"="1");

INSERT OVERWRITE DIRECTORY 's3://ncurcibarstool/output/barstool_results/q4_topwords'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
select t2.single_word, count(*) as NumberofWords
from
(Select t1.single_word as single_word
from (select explode(split(lower(tweet_text), " ")) as single_word
    from barstool_tweets
    where tweet_language='en') t1
```

```
 left outer join stopwords w
 on (t1.single_word = w.word)
 where w.word is NULL and instr(t1.single_word,"#")<>1) t2
 group by t2.single_word
 Order by NumberofWords DESC
 LIMIT 100;

-- remove all words starting with extra characters( number, #, *, &)

INSERT OVERWRITE DIRECTORY
's3://ncurcibarstool/output/barstool_results/q4_topwords_revised'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
select t2.single_word as word, count(*) as NumberofWords
from
(Select t1.single_word as single_word
from (select explode(split(lower(tweet_text), " ")) as single_word
     from barstool_tweets
     where tweet_language='en') t1
 left outer join stopwords w
 on (t1.single_word = w.word)
 where w.word is NULL and regexp_extract(t1.single_word, '^[0-9@#&$]*', 0) == "") t2
 group by t2.single_word
 Order by NumberofWords DESC
 LIMIT 100;
```

Barstool was the most tweeted about word with 3195 tweets, followed by “rt”, “pizza”, “review”, “pizzeria”, and “@stoolpresidente”. Barstool Sports hosts daily pizza reviews at local pizzerias hosted by Dave Portny A.K.A. “@stoolpresidente”

Query 5 – Part1:

For each tweet, calculate the total sentiment words, average sentiment score and total sentiment score.

```
CREATE EXTERNAL TABLE IF NOT EXISTS afinn (
word STRING, rating INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
LOCATION 's3://ncurcibarstool/input/afinn'
TBLPROPERTIES ("skip.header.line.count"="1");
```

```
-- show overall sentiment

INSERT OVERWRITE DIRECTORY
's3://ncurcibarstool/output/barstool_results/q5_overallsentiment'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
select avg(numberofs_words) as AvgSentimentWords, avg(total_sentiment_score) as
AvgTotalSentimentScore, avg(average_sentiment_score) AvgSentimentScore
from
(select w.tweet_id, count(a.word) as numberofs_words, sum(a.rating) as total_sentiment_score,
avg(a.rating) as average_sentiment_score
from (select tweet_id, tweet_language, single_word
    from barstool_tweets b
    LATERAL VIEW explode(split(lower(tweet_text),' ')) adTable AS single_word) w
 inner join afinn a
 on (w.single_word =a.word)
 where w.tweet_language= 'en'
 group by w.tweet_id)temp;
```

Positive tweets had an average sentiment score of 1.85 while negative tweets averaged -1.54 and neutral averaged a -.37 sentiment score.

Query 5 Finished:

Calculate number of positive, neutral and negative tweets about Barstool.

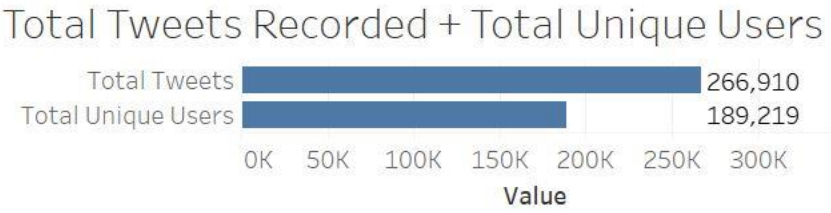
```
INSERT OVERWRITE DIRECTORY
's3://ncurcibarstool/output/barstool_results/q5_sentimentbytype'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
select sentiment, count(*) as numOfTweets
from
(select w.tweet_id, avg(a.rating) as average_sentiment_score,
      case when avg(a.rating)>0 then 'Positive'
          when avg(a.rating)=0 then 'Neutral'
          else 'Negative'
          end as Sentiment
from (select tweet_id, tweet_language, single_word
    from barstool_tweets b
    LATERAL VIEW explode(split(lower(tweet_text),' ')) adTable AS single_word) w
 inner join afinn a
 on (w.single_word =a.word)
```

```
where w.tweet_language= 'en'
group by w.tweet_id) temp
group by sentiment;
```

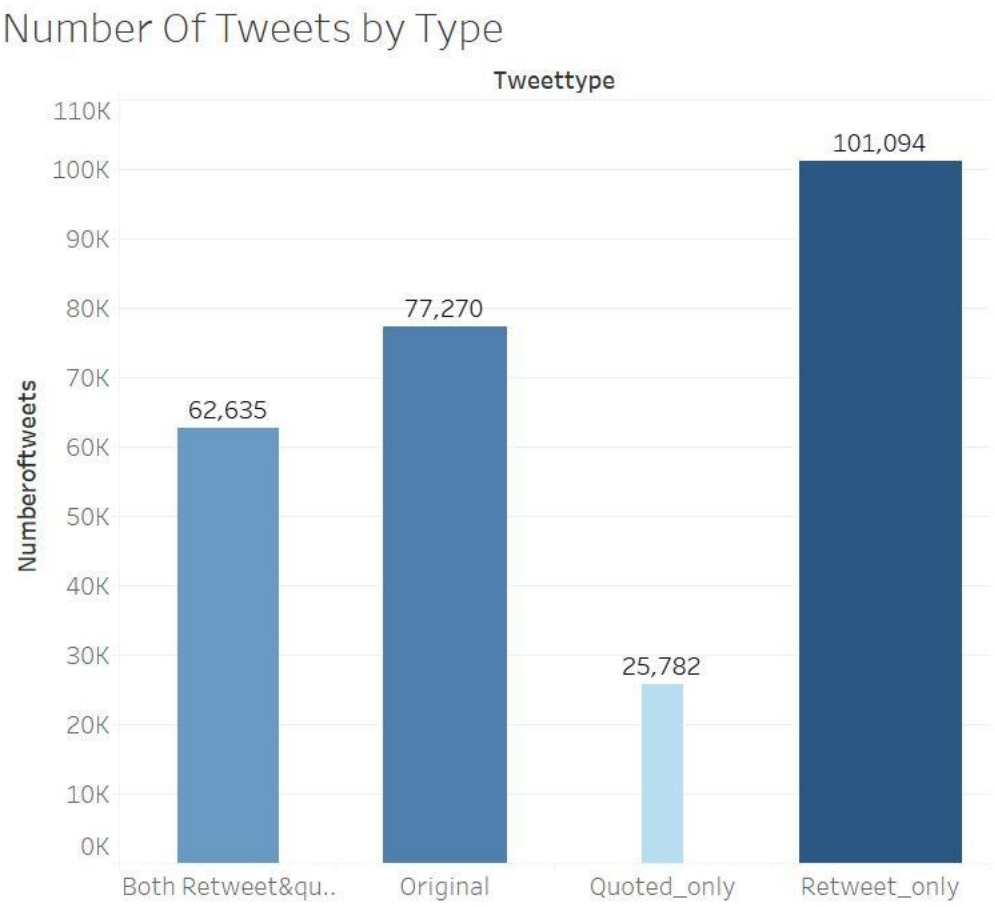
There were 834 negative, 660 positive, and 49 neutral tweets about Barstool from our sample data.

Amazon Web Services Query Graphs:

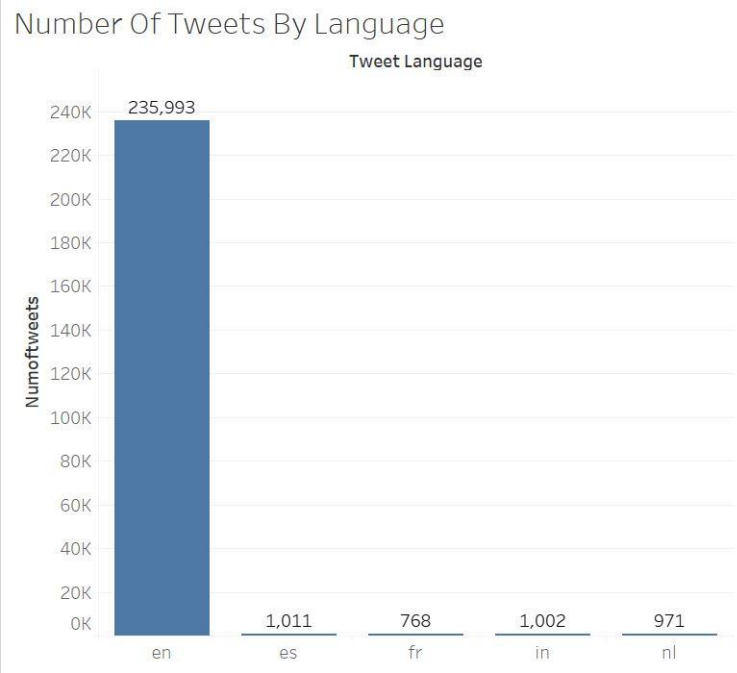
Query 0:



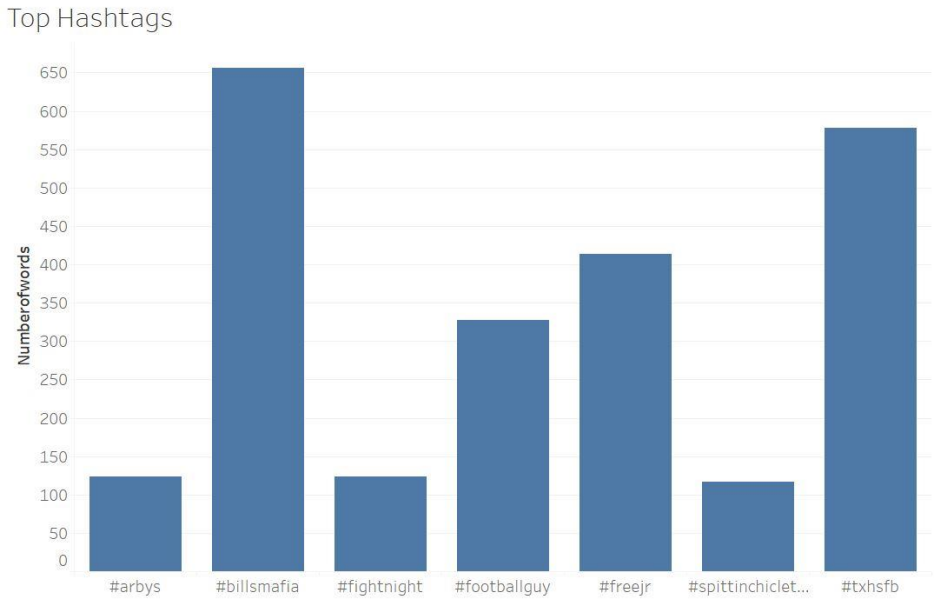
Query 1:



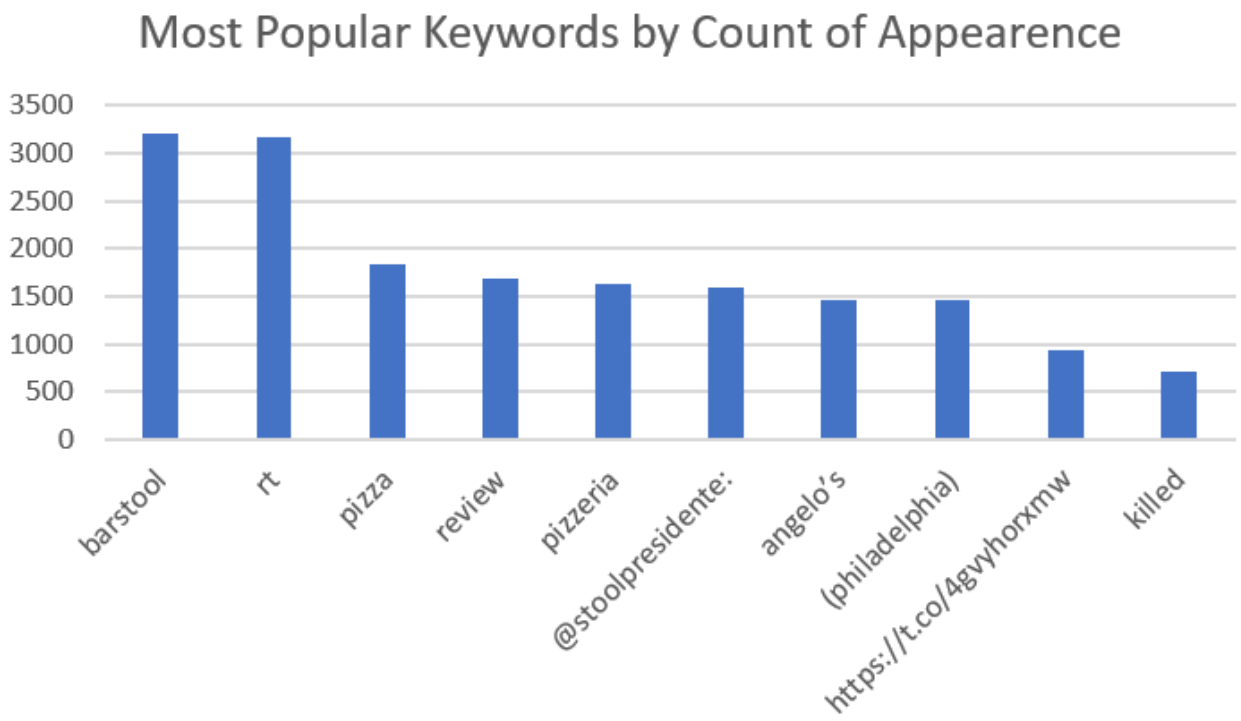
Query 2:



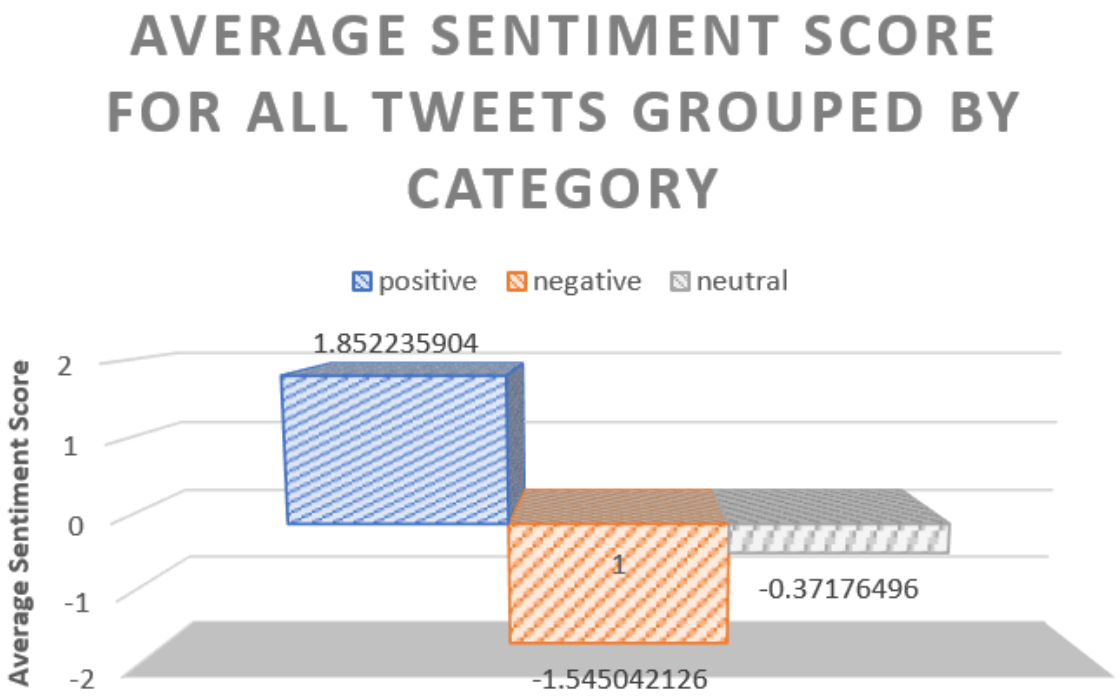
Query 3:



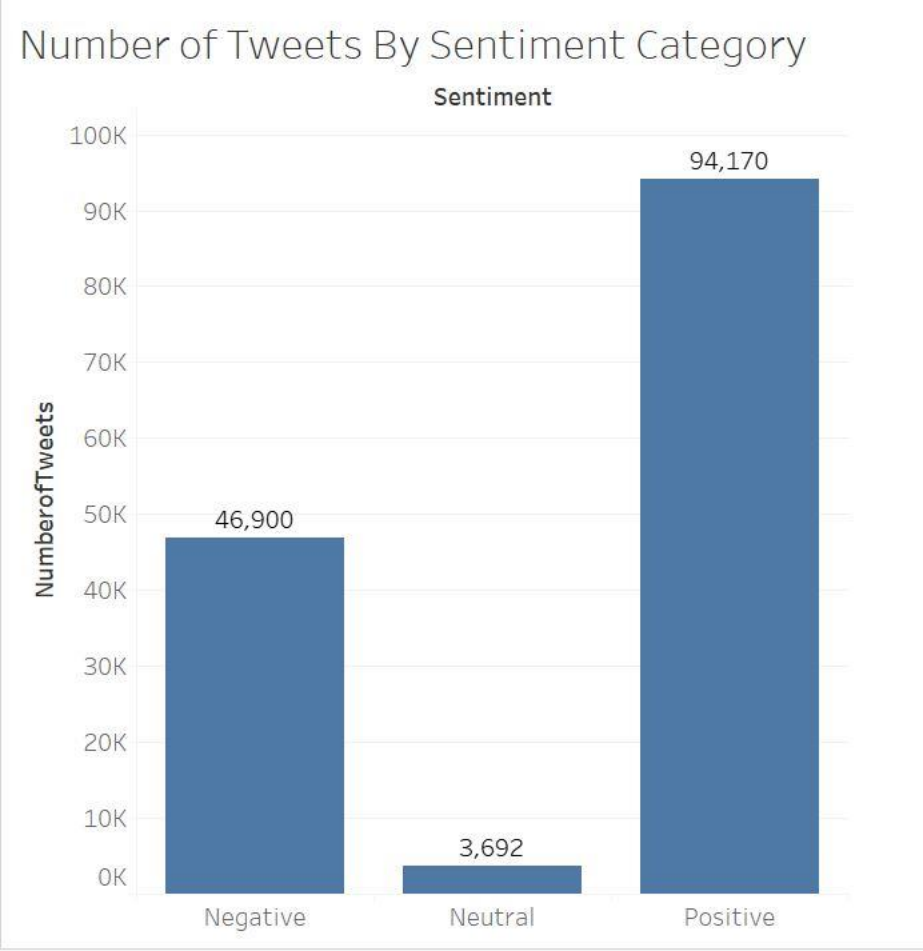
Query 4:



Query 5 Part 1:



Query 5 Finished:



References:

AWS. (2015). AWS Glossary. Retrieved December 12, 2019, from https://docs.aws.amazon.com/general/latest/gr/glos-chap.html.

Barstool Sports. (n.d.). Barstool Sports (@barstoolsports) | Twitter. Retrieved December 8, 2019, from https://twitter.com/barstoolsports?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr

Burns, M. (2017, July 18). Past, Present, Future: How Barstool Sports Is Swinging For The Fences In Digital Media. Retrieved December 8, 2019, from https://www.forbes.com/sites/markjburns/2017/07/11/past-present-future-how-barstool-sports-is-swinging-for-the-fences-in-digital-media/#1dc98f554edb

DailyBeast. (n.d.). Barstool and Sexual Harassment. Retrieved December 8, 2019, from https://www.thedailybeast.com/inside-barstool-sports-culture-of-online-hate-they-treat-sexual-harassment-and-cyberbullying-as-a-game

Hootsuite. (2019, October 31). 25 Twitter Statistics All Marketers Should Know in 2020. Retrieved December 8, 2019, from https://blog.hootsuite.com/twitter-statistics/

NBC. (2017, January 31). NFL pulls credentials from Barstool Sports. Retrieved December 8, 2019, from https://profootballtalk.nbcsports.com/2017/01/31/nfl-pulls-credentials-from-barstool-sports/

NBC. (2019, September 21). Barstool Sports and the persistence of traditional masculinity in sports culture. Retrieved December 8, 2019, from https://www.nbcnews.com/news/us-news/barstool-sports-persistence-traditional-masculinity-sports-culture-n1057061

Acknowledgements / Author Note:

I'd like to extend a thank you to Bryant University and Dr. Suhong Li for creating and holding an extremely informative and instructional class. I would also like to thank Dr. Suhong Li for her aid in structuring queries and her advice in the direction of this project. I would also like to extend my appreciation to Barstool Sports and Tyler O'Day for making this report and accompanying presentation possible. Without the guidance and counsel of the Viceroy program, this analysis would not hold the same attention to detail or level of passion that it does. Again, thank you to all involved in the making of this report and I am looking forwards to discussion of these findings shortly.

Nicholas Curci