Madeleine Singh, Elaina Tiller, Victoria Stahl

APIs, SQL, and Visualizations

https://github.com/toriastahl/Final-Project

Original Goals

We initially planned to use APIs from Twitter, iTunes, and Spotify. We planned to determine the top podcasts on itunes and Spotify, and then look at its mentions for hashtags and shares on Twitter. We thought we could find a correlation of twitter popularity versus Spotify/Itunes. Are iTunes charts or Spotify charts more indicative of which podcasts are actually the most popular/talked about? Most importantly, we wanted to hone our programming skills and practice using APIs and SQL.

Goals Achieved

APIs/websites used: YouTube, Spotify, Reddit (website)

During our project, we changed our scope to accomodate what was available through the APIs. Twitter's API was limited with its hashtag data, and Spotify does give any data for popularity or streams on any podcast. We ended Spotify, Reddit, and YouTube as our three sources. We specifically used the top podcast, Joe Rogan Experience because it had a lot of potential content that we could find and analyze. Through Youtube's API, we were able to get guest appearances, views, likes, and dislikes for the 1600+ podcasts. Since JRP is often a controversial podcast due to the various guests, it was interesting to see how the views and likes varied by guest. We used Spotify and Reddit to get dates of the episodes and discussions on Joe Rogan's Reddit page. We ended up with a handful of visualizations that will serve as good recommendations of quality JRE episodes, and in doing so advanced our programming skills.

Problems

Problem 1: The first major problem we faced was that our bulk of API data from YouTube was removed due to Joe Rogan's $100 million deal with Spotify. I didn't realize this meant that his videos would be removed from YouTube, but luckily I had a cached csv file from early in the project I worked with before I knew how to limit the API return items.

- Solution: With permission, I then changed all the code to work with the csv and limited the csv file data to mimic an API. Although this means we will not be getting JRP episodes form the last week or two, using this cached file gives us access to make visualizations with views, likes, dislikes, and ratings which are not accessible by using the Spotify API.

Problem 2: Other major problems were encountered in making the bar graph with guests of the highest viewed shows. Our database that has the guests names and their id's only has the ids for shows with one guest, because it would not be possible to put two guest id's in the youtube database. So, when selecting the top 8 videos by views and JOIN'ing the table with the id's, some of the videos didnt give guest names. So for the videos with a None value, I put the title of the video on the chart and split it to find the two people in the episode. However, this gave another problem since the two names were a category on the table, the text was long and cut off

- Solution: I used a wrap function from an importanted library to automatically text into multiple lines after a character limit. This made a neat looking graph with lines of text for the bar graph categories.

```
guestname = ['\n'.join(wrap(x, 16)) for x in  guestname]
```

Problem 3: When scraping Reddit for the discussion posts for Joe Rogan's podcast with the most comments by date, there were hundreds of discussion dates available once the webpage was loaded, but when scraping using .requests, it would only return the 17 most recent dates.

- Solution: In office hours we were recommended to download the webpage at a .htm file like in Project 2 and we then limited the data grabbed to 100 entries.

Calculations

youtube.txt

```
Guest,Number Apperances
Brendan Schaub,8
Tim Dillon,6
Tom Papa,6
Joey Diaz,6
Donnell Rawlings,5
Tony Hinchcliffe,5
Mike Baker,5
Brian Redban,5
Lex Fridman,4
Duncan Trussell,4
Greg Fitzsimmons,4
Andrew Santino,4
Steven Rinella,3
Bridget Phetasy,3
Jeremy Corbell,3
Bill Burr,3
Andrew Schulz,3
Michael Yo,3
Tim Pool,3
Eric Weinstein,3
Michael Malice,3
Bryan Callen,3
Tom Green,2
Nicholas Christakis,2
Gad Saad,2
```

fileOutputEpisodes.txt ×

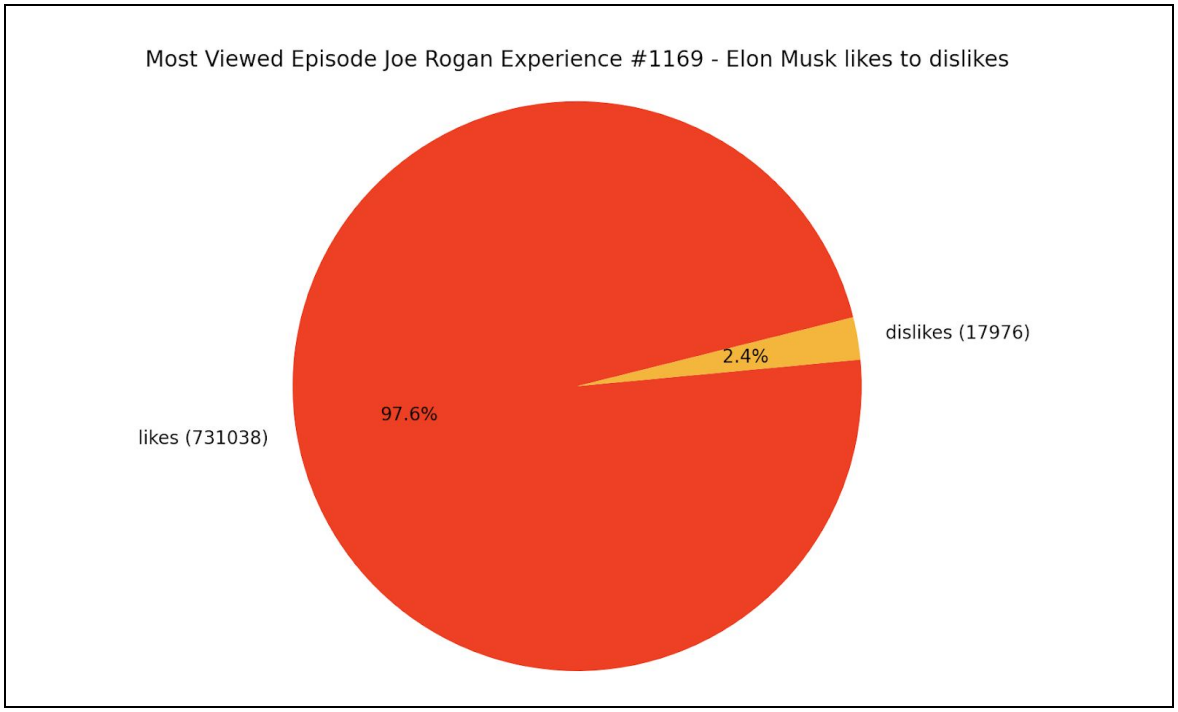
fileOutputEpisodes.txt

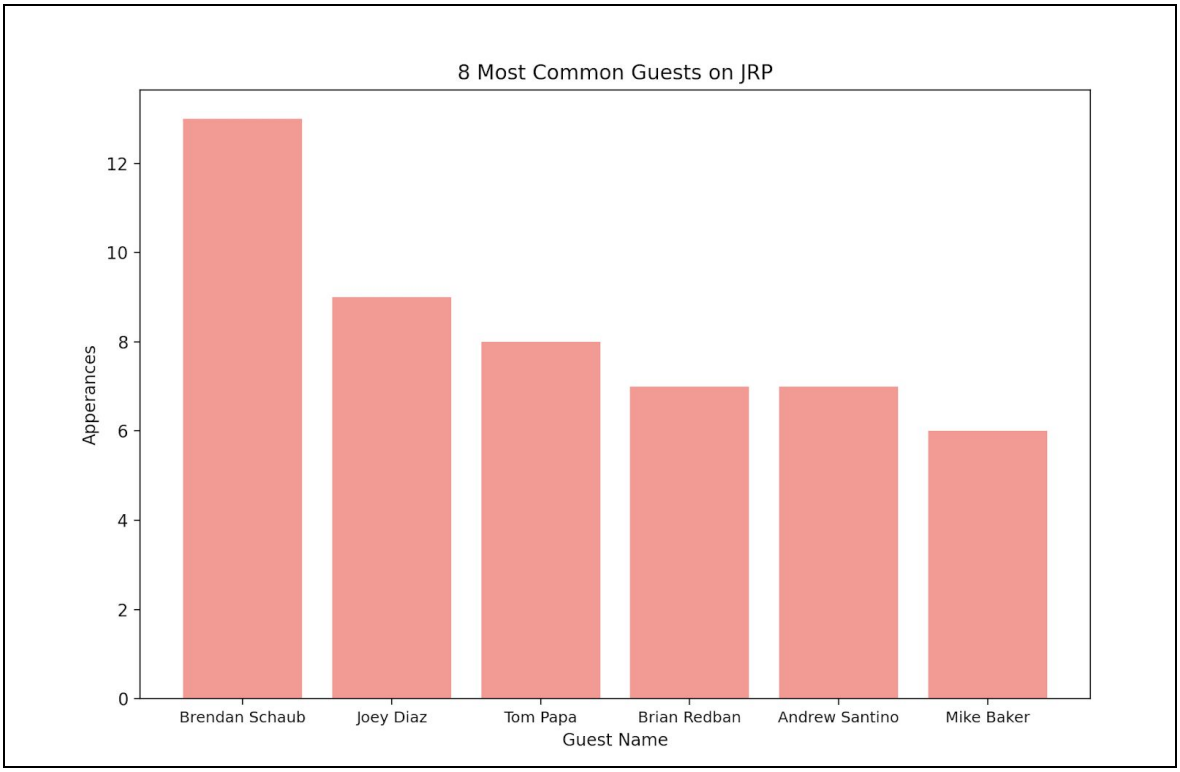
```
Month (2020),Number of Episodes
January,20
February,18
March,18
April,17
May,20
June,17
July,20
August,12
September,14
October,15
November,15
December,5

```

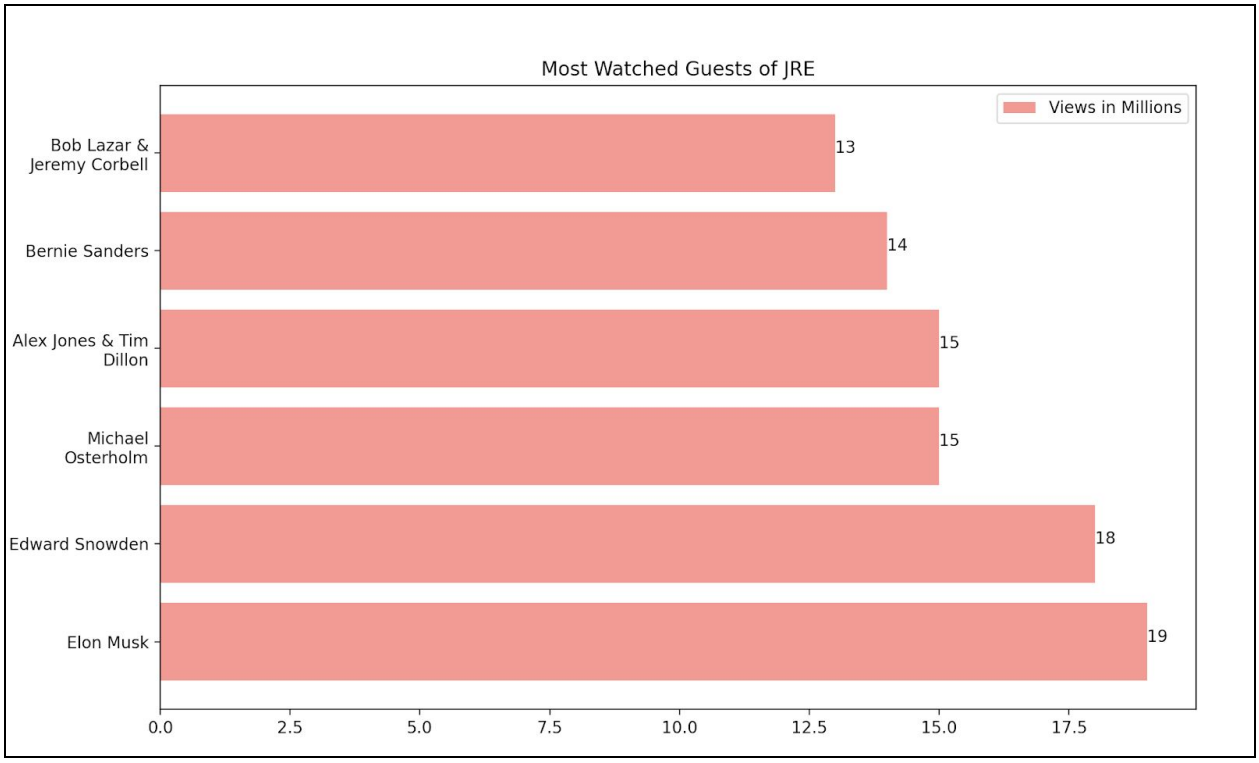
reddit.txt

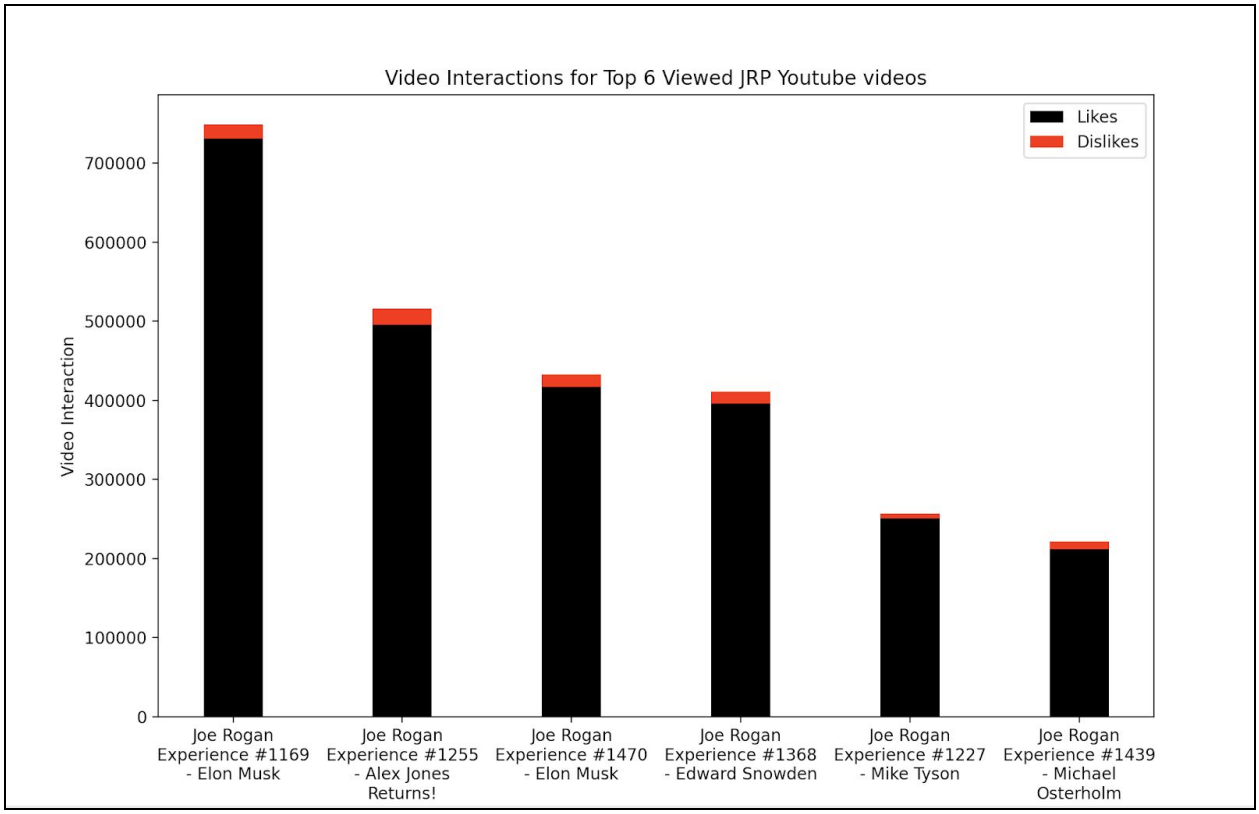
```
Average Number of Comments is 90.69
```

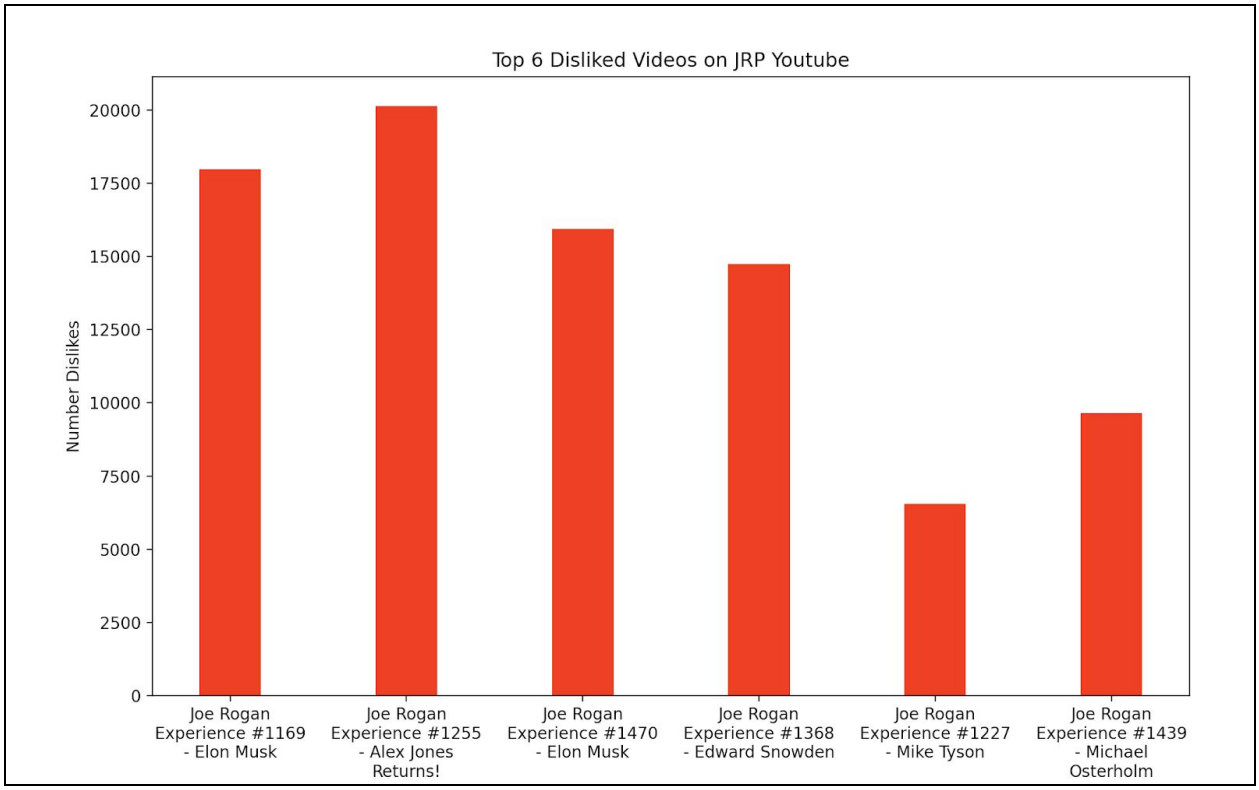
Visualizations

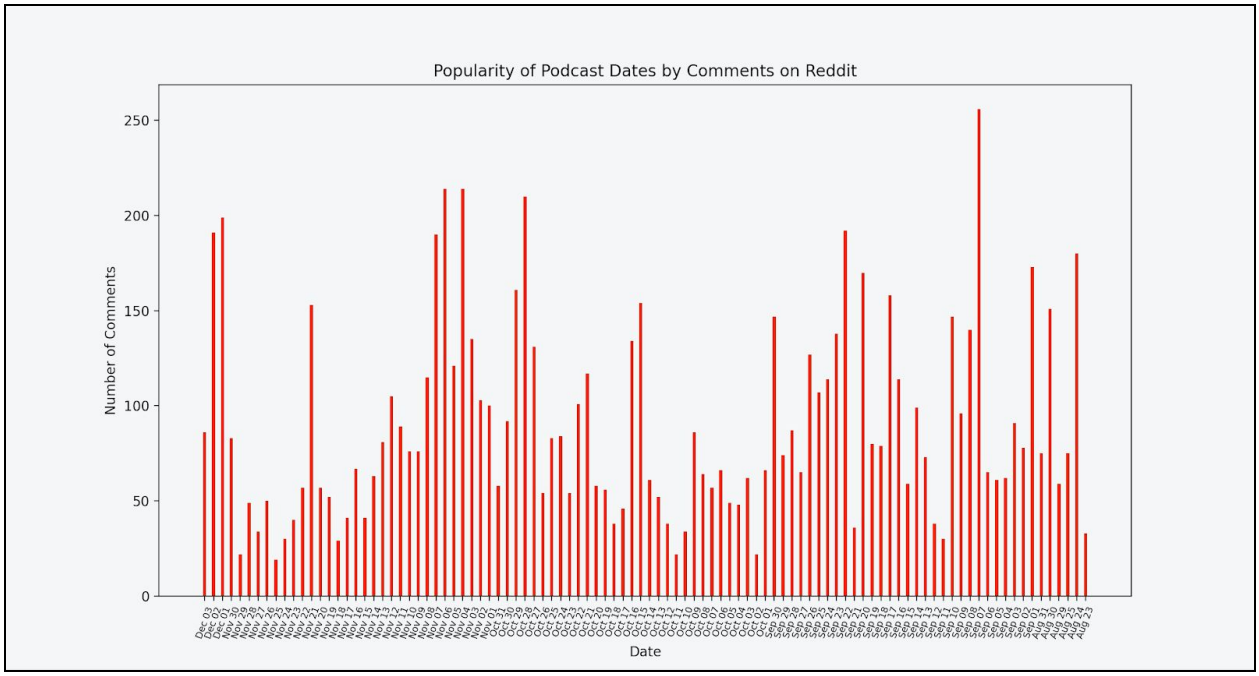


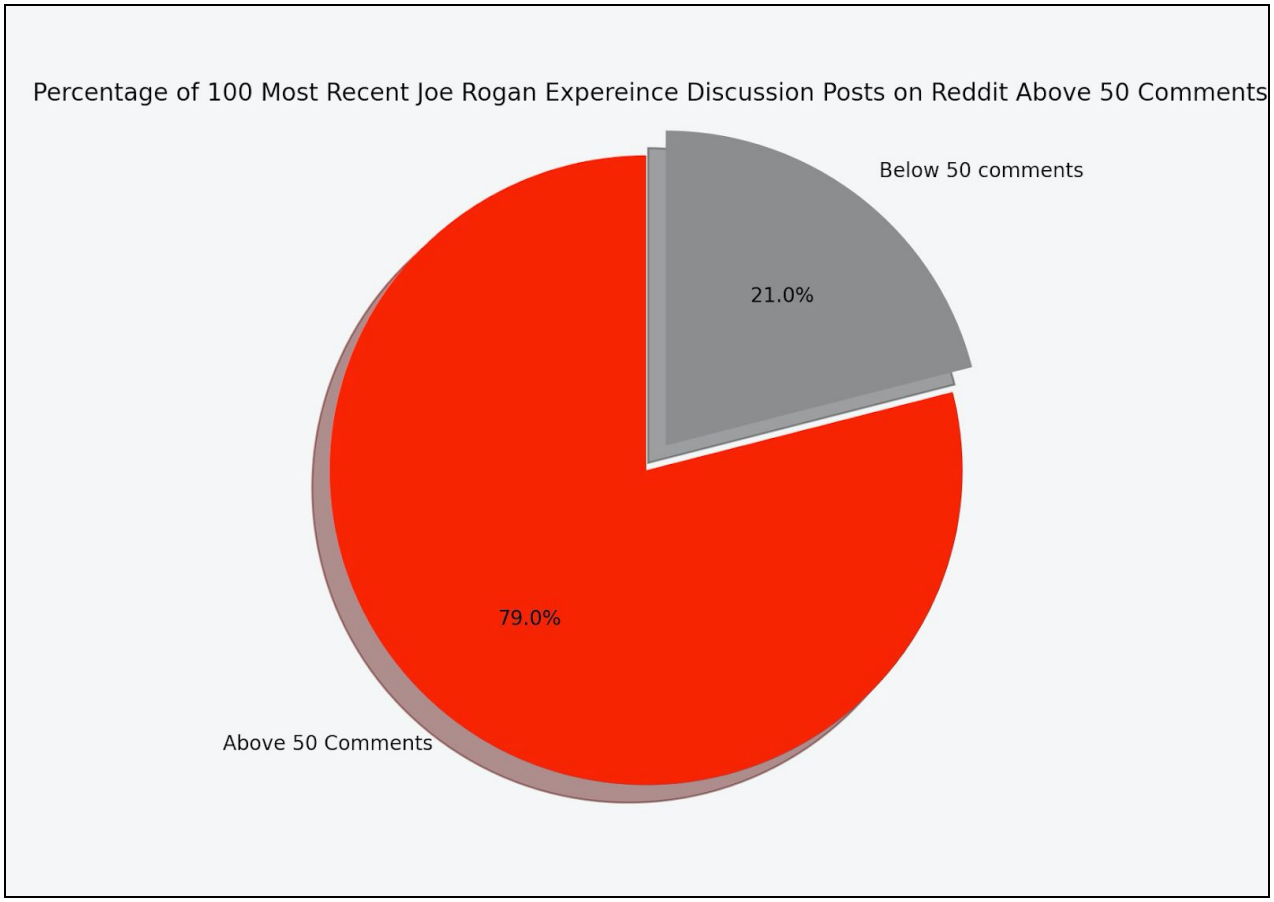


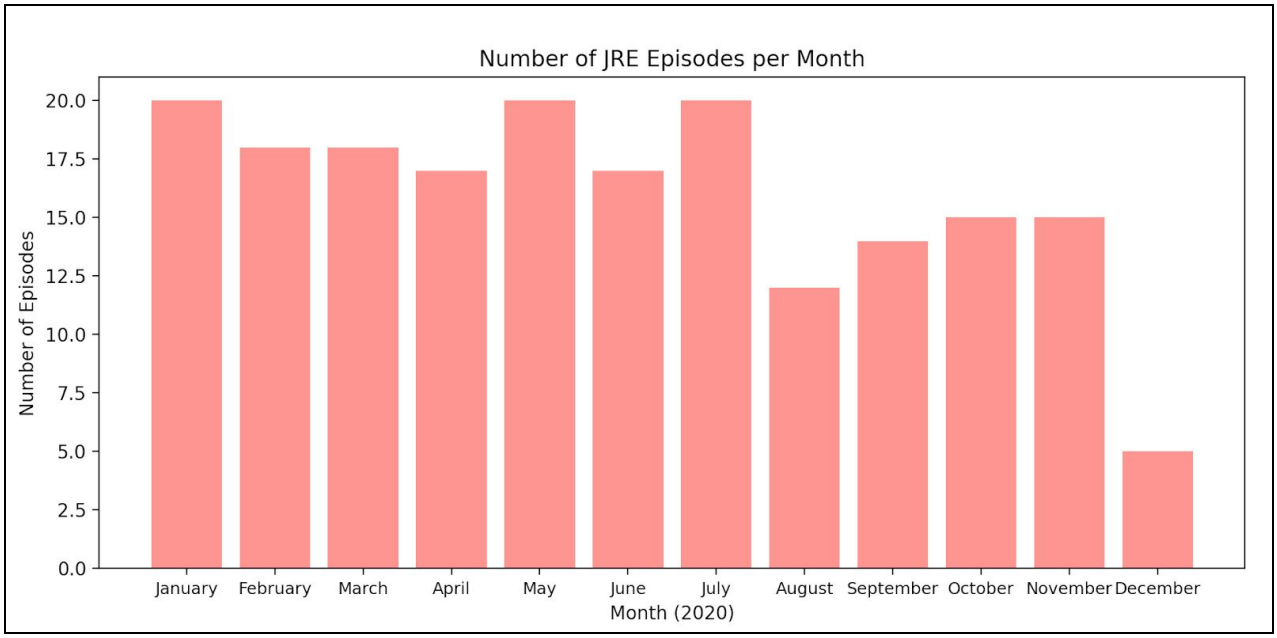

Most Watched Guests of JRE
Views in Millions
Bob Lazar & Jeremy Corbell
Bernie Sanders
Alex Jones & Tim Dillon
Michael Osterholm
Edward Snowden
Elon Musk
13
14
15
15
18
19
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

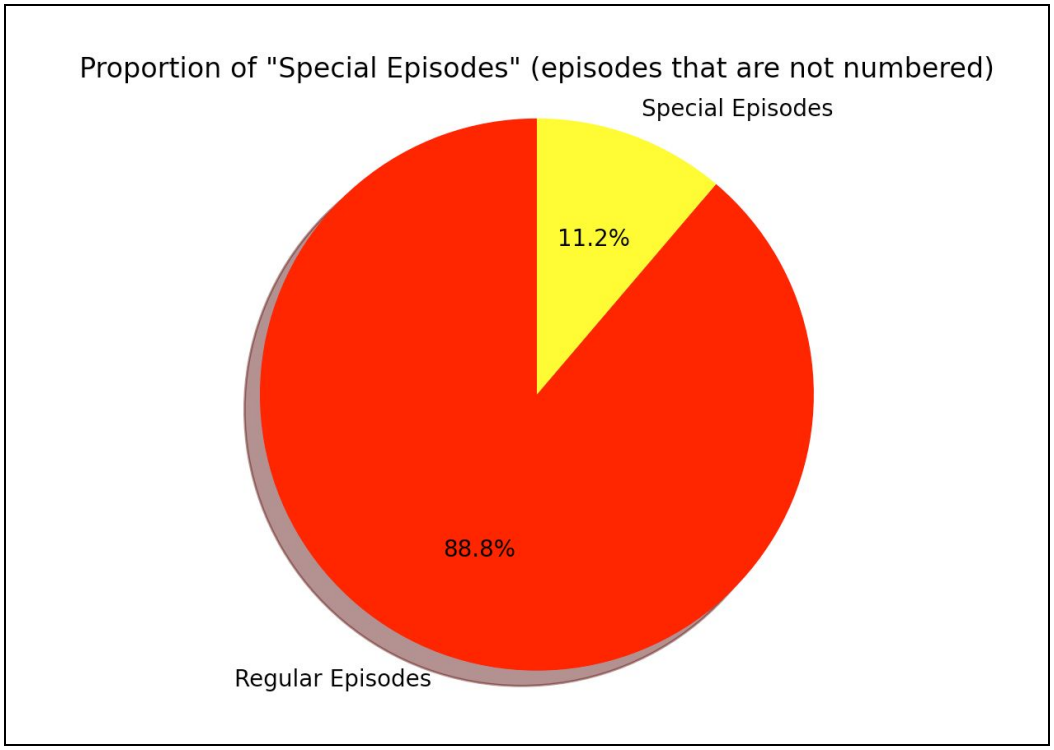

Video Interactions for Top 6 Viewed JRP Youtube videos
Likes
Dislikes
Video Interaction
0
100000
200000
300000
400000
500000
600000
700000
Joe Rogan Experience #1169 - Elon Musk
Joe Rogan Experience #1255 - Alex Jones Returns!
Joe Rogan Experience #1470 - Elon Musk
Joe Rogan Experience #1368 - Edward Snowden
Joe Rogan Experience #1227 - Mike Tyson
Joe Rogan Experience #1439 - Michael Osterholm


Top 6 Disliked Videos on JRP Youtube
Number Dislikes
0
2500
5000
7500
10000
12500
15000
17500
20000
Joe Rogan Experience #1169 - Elon Musk
Joe Rogan Experience #1255 - Alex Jones Returns!
Joe Rogan Experience #1470 - Elon Musk
Joe Rogan Experience #1368 - Edward Snowden
Joe Rogan Experience #1227 - Mike Tyson
Joe Rogan Experience #1439 - Michael Osterholm


Popularity of Podcast Dates by Comments on Reddit
Number of Comments
0
50
100
150
200
250
Date


Percentage of 100 Most Recent Joe Rogan Expereince Discussion Posts on Reddit Above 50 Comments
Below 50 comments
21.0%
79.0%
Above 50 Comments


Number of JRE Episodes per Month
Number of Episodes
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
January
February
March
April
May
June
July
August
September
October
November
December
Month (2020)


Proportion of "Special Episodes" (episodes that are not numbered)
Special Episodes
11.2%
88.8%
Regular Episodes

Instructions for running code

youtube_JRP.py must be run first. Next, reddit.py and spotify.py can be run in any order.

youtube_JRP.py
Scroll to the bottom for main. Run section 1 8+ times to get at least 200 people in the db (makes calculations easier with more values). If you want to see visuals, uncomment below graphs.

reddit.py
Run setUpComments 4 times to add all 100 entries to the database Popularity (adds 25 at a time). Then to see visuals, uncomment below calculations and graphs.

spotify.py
First, follow the instructions at the top of the file to get a fresh token to use with Spotify's API. Paste this token into the variable 'token' on line 15. Then, run the code to get data - 8 or more times is recommended to create accurate visualizations. Comment out Section 1 and uncomment Section 2 if you want to see visualizations or calculations.

Documentation

youtube_JRE.py

```
def readDataFromFile(filename):
'''Takes in the file youtube_data.csv to read it and return file data'''

def setUpDatabase(db_name):
    '''Takes in the database 'JRP.bd' as a paramater and sets it up, also returning cur and conn'''

def uploadDataJRE(cur,conn):
    '''Takes in cur and conn to create JRP and JRP_guest_count table in JRP.db with all the information for
    each podcasts on Rogans youtube channel. Finds the last inserted id for both tabls, and inserts another 25
    more podcasts for the JRP table from the csv file that has not been added. For the JRP_guest_count it
    adds all guests that have not yet been added, or increments their apperances. This function also links the
    tables together with shared key of guestid in JRP and id in JRP_guest_count. For all
    episodes with two guests, or an irregular episode, the guestid is defaulted to 0 in JRP'''

def printNamesPretty(file):
    '''Takes in a file and writes out calculated apperances of each guest in form guest,apperances'''

def barChart1(cur):
    '''Bar chart of the top 6 guests by number of apperances in the database'''

def barChart2(cur):
    ''' Shows number of views in millions for the top 6 watched
    videos in the database'''

def barChart3(cur):
    '''Show number of likes and disliked stacked on top of eachother
    for top 6 most viewed videos in the database '''

def barChart4(cur):
    '''Makes a pie chart of the Top 6 disliked videos in database and
    their number of dislikes'''


def pieChartMostViewedEps(cur):
    '''Makes a pie chart of the most viewed episode in the database
    with percentage and number of likes to dislikes displayed'''
```

reddit.py

```
def getDates(filename):
    '''This function takes the .htm file as a parameter and scrapes the file for the date of the discussion post
    and the number of comments on that discussion post. It returns a list of 100 tuples that include the date,
    number of comments, and count which we use later as discussion_id'''

def setUpDatabase(db_name):
    '''This function takes the database 'JRP.db' as a parameter, sets up the database, and returns cur and conn.'''

def setUpComments(dates_comments, cur, conn):
    '''This function takes the list of tuples dates_comments, cur, and conn as parameters and inserts the data from
    dates_comments into the database set up above (JRP.db). This code needs to be run 4 times since it inputs data
    25 rows at a time and has a total of 100 rows.'''

def getAverageComments(cur):
    '''This function selects the dates from the Popularity table to calculate the total number of dates and the comments
    from the Popularity table to calculate the total number of comments for the 100 dates. It then returns the average
    number of comments for each discussion post.'''

def printAverageComments(comments,file):
    '''This function writes the average number of comments to reddit.txt file as a calculation.'''

def makeVisualizations(cur):
    '''This function selects the dates and number of comments from the Popularity table and returns a bar chart
    with the number of comments on each of the 100 most recent Joe Rogan Experience discussion posts on Reddit.'''

def vizualizationByComments(cur):
    '''This function selects the dates from the Popularity table to calculate the total number of dates gathered
    and the number of comments from the Popularity table that are above 50 to make a percentage of discussion
    posts that have above 50 comments since those posts are considered 'popular'.'''

def main():
    '''This function calls the above functions.'''
```

spotify.py

```
def episodes_search(id, offset,cur):
    '''this function uses the Spotify API to grab information for 25 episodes at a time (starting with the most
    recent, but can be offset by updating the offset parameter), and then finds the name and release date of each
    episode grabbed. it returns a list of 25 tuples of (name, release date)'''

def setUpDatabase(db_name):
    '''this function takes the database 'JRP.db' as a parameter, sets up the database, and returns cur and conn'''

def setUpEpisodes(data, cur, conn):
    '''this function sets up the Episodes table that will go into the JRP.db database. It takes the list of tuples
    returned by episodes_search and put them in a table that has values episode_id (which is a count), name,
    and release data. it adds 25 items at a time because that is how many is returned by episodes_search'''

def createPieChart(cur):
    '''Once the above 3 functions are run and an appropriate amount of data is gathered, run this function to
    create a pie chart that shows the percentage of 'Special Episodes.' These are episodes that are not numbered
    with Joe Rogan's normal numbering sequence. It uses the Spotify Episodes table in JRP.db to get the data
    and counts how many episodes start with a # (which is his 'normal' numbering sequence) how many don't'''

def createBarGraph(cur,file):
    '''this function, which should be run at the same time as createPieChart, creates a bar graph showing the number
    of episodes released in each month of 2020. It uses Spotify Episodes and splits the release date into its respective
    parts, which are then used to create counts for each month. It then displays these findings in a bar graph AND
    writes them to a text file, which can be named with the 'file' variable.'''

def main():
    '''this function is used to call the above functions. it is recommended to call the first three at once
    multiple times, and then the last two only once.'''
```

Resources Used

Date	Issue Description	Location of Resource	Result
11/29	Needed to extract name from complex string statement	https://www.datacamp.com/community/tutorials/python-regular-expression-tutorial	Used match.group(...) to get the group by name it the defined regex expression
12/1	Needed to change a specific cell in SQL database to put a shared id value for a whole column	`https://stackoverflow.com/questions/3024546/change-one-cells-data-in-mysql`	Using a loop and (`UPDATE my_table SET my_column='new value' WHERE something='some value'`) I was able to change a whole column's values to s shared guestid
12/1	Needed to zip two lists together to make a list of tuples	`https://www.kite.com/python/answers/how-to-zip-two-lists-in-python`	Showed me how to zip the two lists together and then cast the zip to a list
12/1	Needed help using Spotify's API to get specific pieces of episode information	`https://developer.spotify.com/console/get-show-episodes/?id=4rOoJ6Egrf8K2IrywzwOMk&market=&limit=50&offset=1`	Was able to get token, understand parameters, and retrieve data successfully from the API
12/2	Needed to start reading a csv file after a certain row(imitating the 25 upload maximum of an API)	https://stackoverflow.com/questions/26464567/csv-read-specific-row	Resource allowed me to use a csv reader and next() function to get a row containing a specific row number
12/5	Y axis of 'views' for a bar graph was being put in exponential form and it was not intuitive to look at	`https://stackoverflow.com/questions/14711655/how-to-prevent-numbers-being-changed-to-exponential-form-in-python-matplotlib-fi`	Showed how to change the ticklabels to `useOffset=False` which shows the views in by pure millions instead
12/5	X axis names of a bar chart were crossing over each other making it hard to read	https://stackoverflow.com/questions/43152502/how-can-i-rotate-xticklabels-in-matplotlib-so-that-the-spacing-between-each-xtic	Showed how to rotate the x-axis labels by 45 and make the font smaller to make it easier to read with `plt.setp(ax.get_xticklabels(), ha="right", rotation=45)`

12/5	X-axis labels were long, cut off and unreadable even when shifted 45 degrees	https://stackoverflow.com/questions/59466109/how-to-get-x-axis-labels-in-multiple-line-in-matplotlib	Used `from text wrap import wrap` To wrap lines after a certain amount of characters making graph easier to read
12/5	Needed to select top values for a certain column based on values	https://www.w3schools.com/sql/sql_top.asp	Using SQl command : `"SELECT views FROM JRP ORDER BY views DESC LIMIT 6"` I was able to select top 6 viewed videos
12/5	Wanted to customize the graph's to include the title of the video it got from the data, but forgot how to use the %s or %d format	https://stackoverflow.com/questions/997797/what-does-s-mean-in-a-python-format-string	Easily explained the python format string specifier syntax
12/7	Needed to make a pie chart using matplotlib	https://matplotlib.org/3.1.1/gallery/pie_and_polar_charts/pie_features.html	I was able to create a more complex pie chart
12/9	Write my calculation to a text file even though it wasn't in a list	https://www.geeksforgeeks.org/reading-writing-text-files-python/	I was able to write my string to my .txt file