Taylor Wynn

SI 206

11/4/17

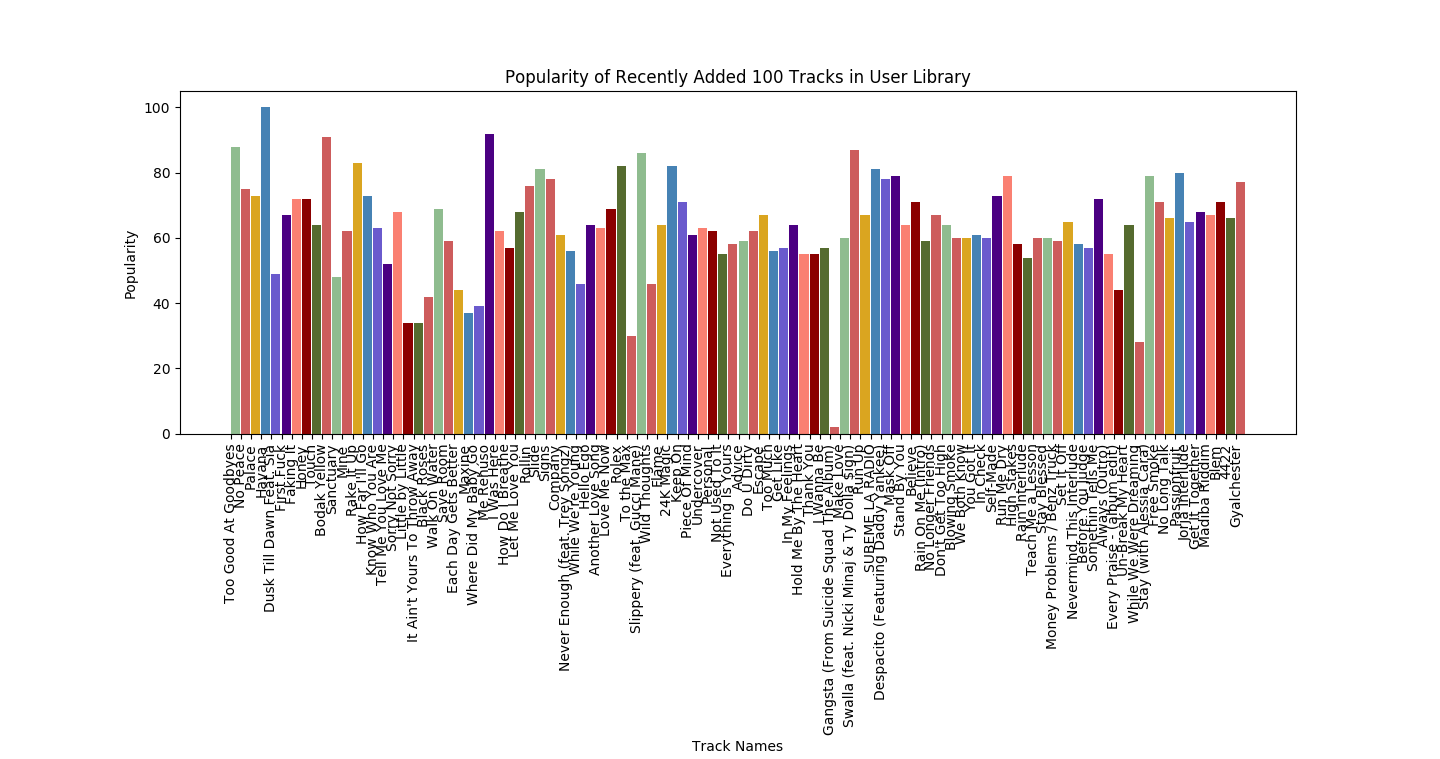
206 Final Project Report

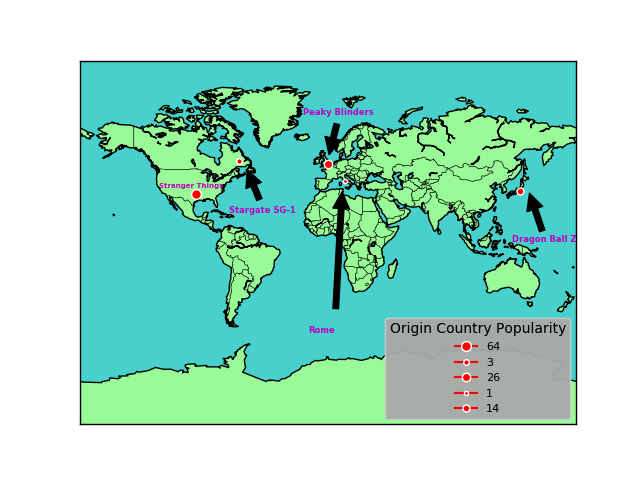
I had several goals for the Final Project. One, I wanted to learn to access some of the more challenging APIs such as YouTube and Spotify. Normally, people tend to shy away from using these APIs because of their difficult authentication requirements. Second, I wanted to retrieve my user data with all the APIs I used. I thought this would make the project more interesting in the end because I could learn more information about myself e.g. how active I am on Facebook or how often I listen to certain songs on Spotify. Third, I wanted to create different visualizations from the data gathered from these APIs.

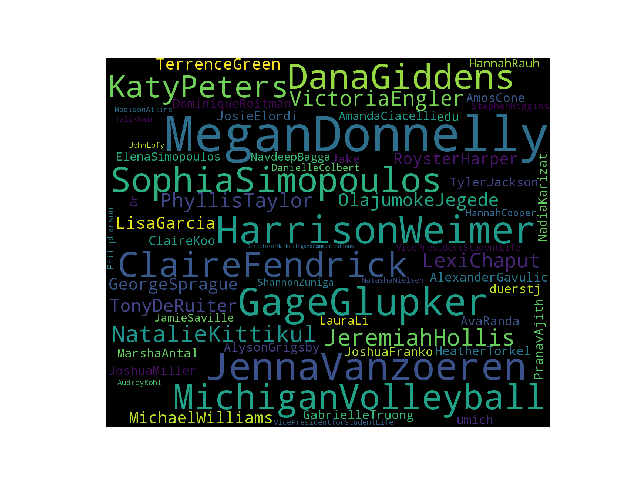
In the end, I achieved two out of my three goals. I was able to accomplish my first goal and successfully accessed and retrieved data from some of the more challenging APIs such as Google Mail, YouTube, and Spotify. I was able to accomplish my third goal and learned how to make a Google Map, WordCloud, and bar graph visualizations. I was not able to accomplish my second goal mainly because I was not active enough on certain sites. For example, I had no information to gather from the YouTube API because I never post videos on my channel.

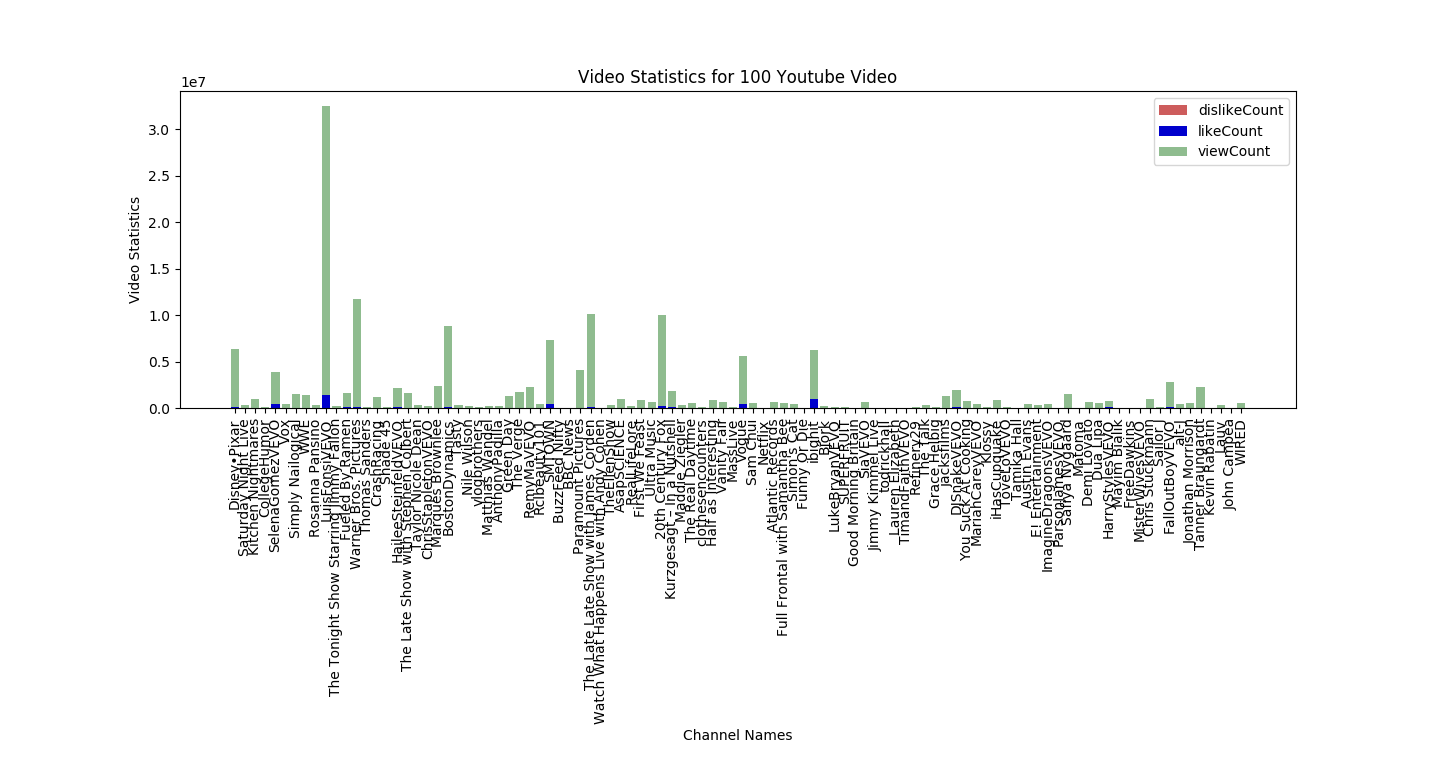
I faced many problems along the way with this project. At the beginning I had a lot of problems accessing the YouTube and Google Mail APIs. Most of the authentication code I found for OAuth was not easy to follow (until I found one website!) which made it difficult to start the project. I also ran into problems learning how to page certain APIs so that I could get exactly 100 interactions. Luckily, most APIs already have a paging feature, which made things easier. After accessing the APIs, I ran into problems trying to read through the nested data and pull out what I wanted. For example, the Google Mail API returned to me a dictionary with a key that held a list with several dictionaries that had the same keys. So I had to figure out how to access the data I wanted. Lastly, I ran into problems with creating the visualizations. The first visualization I made was the Google Map where I ran into problems trying to annotate it. When it came to the bar graph, I ran into problems learning how to write the bars in different locations on the graph. In the end after a lot of trial and error I was able to overcome these problems and create the following social media “report”.

**Social Media Report**

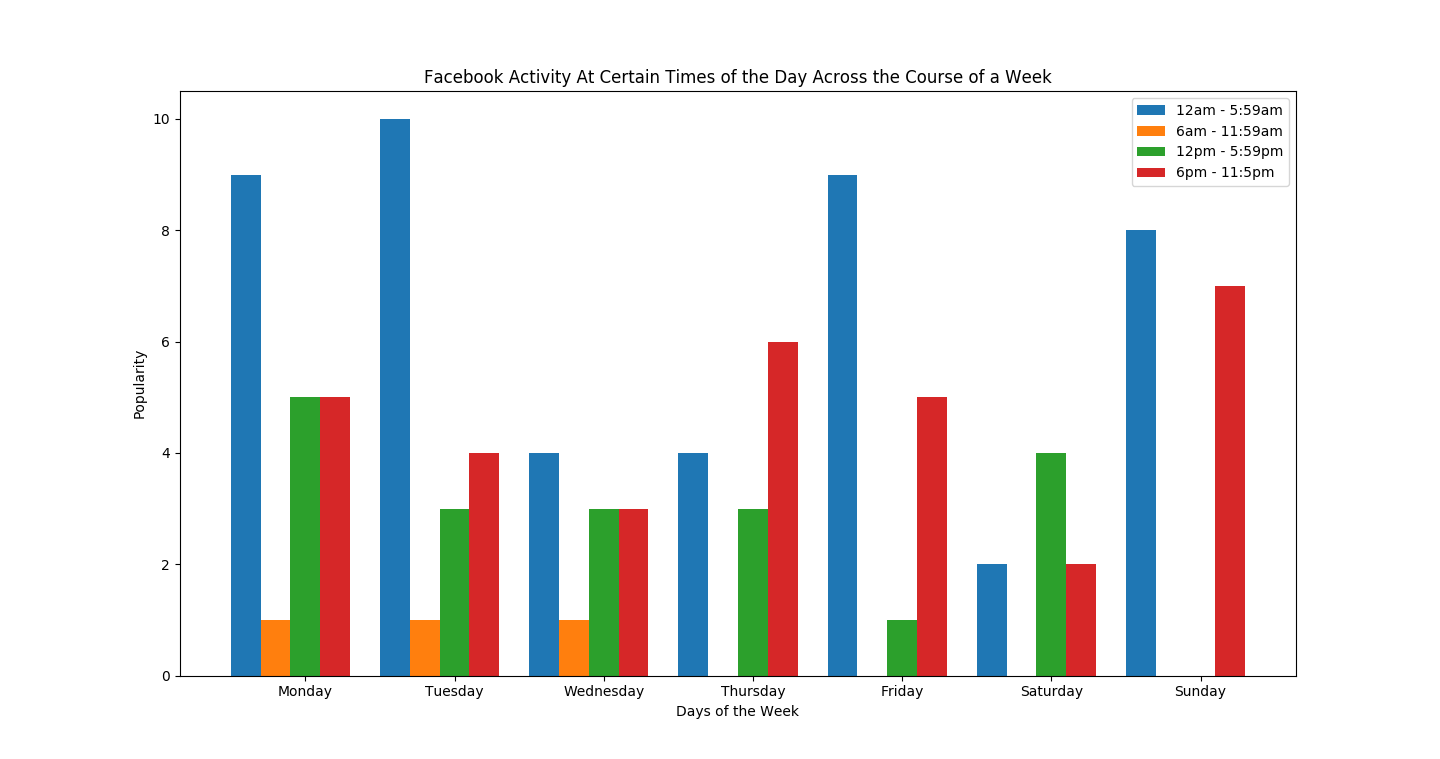
1. Using the ‘Spotify API’ (more specifically the spotipy wrapper) I ultimately created a bar graph that shows the popularity of the first 100 tracks in my user library. It might not be extremely clear enough to read all the text at the bottom, but according to the graph the most popular song out of the top 100 in my library is *Havana*.

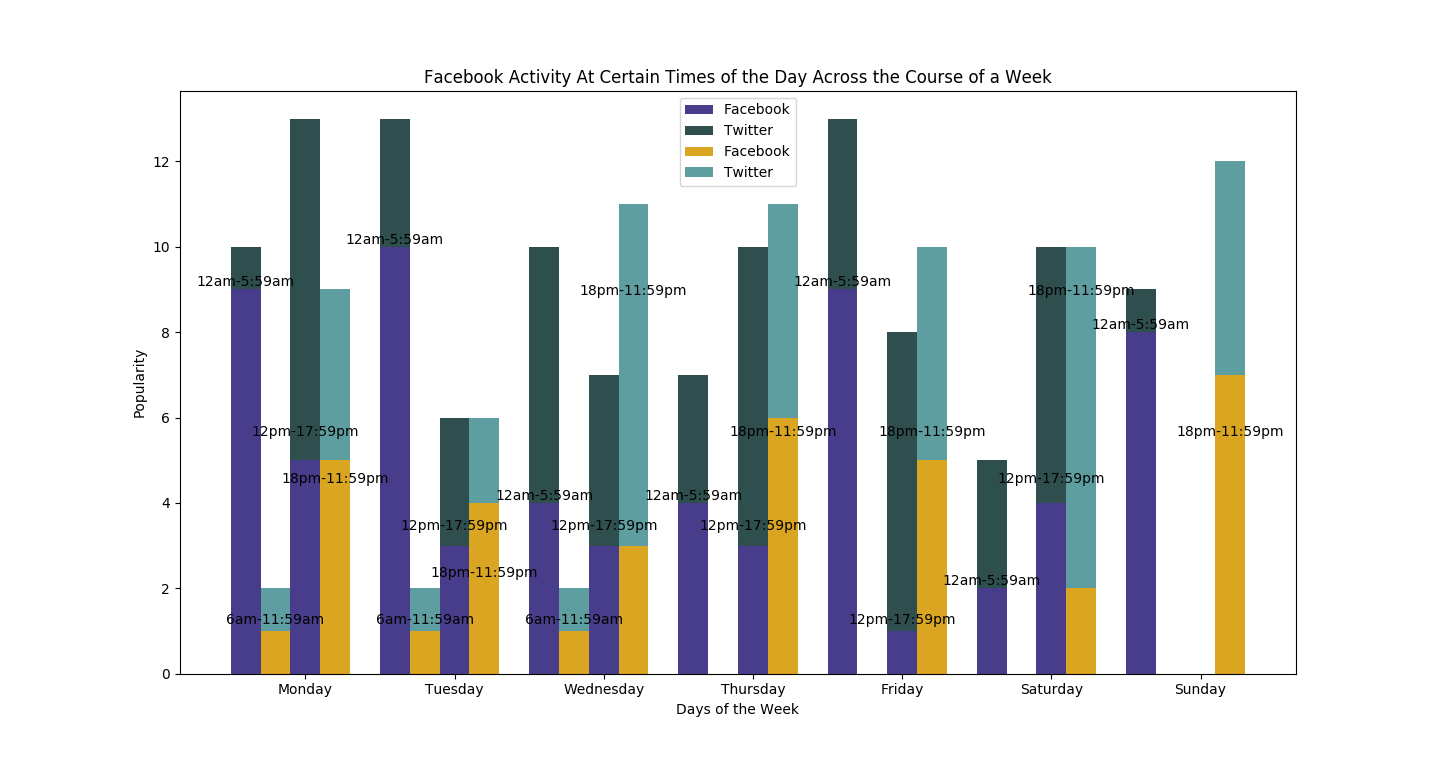
2. Using the ‘The Movie Database API’, I ultimately created a Google Map that shows the popularity of each origin country for the top\_rated 100 TV Shows on the database along with the most popular TV show in each country mentioned. According to my findings, the most popular origin country for TV Shows is the United States where the most popular TV show is ‘Stranger Things’ while the least popular origin country for TV Shows is Italy where the most popular TV Show is Rome.

3. Using the ‘Gmail API’, I ultimately created a WordCloud that shows who I get the most emails from (the bigger the name, the more emails I get from them). In order for WordCloud to not divide people’s first and last names, I combined them into one name. According to my findings, I get the most emails from Megan Donnelly.

4. Using the ‘YouTube API’, I ultimately created a Stacked Bar Graph that shows which videos get the most views, likes, and dislikes combined. Interestingly enough, the dislikes for the videos were not significant enough (compared to the viewCount and likeCount) to show up on the graph. This can mean that people tend to watch lots of YoutTube vidos but do not react to them online. In order for the titles to fit on the graph, I had to use the channel names instead of the video names (because they were too long). According to my findings, the music video Échame La Culpa with Luis Fonsi and Demi Lovato is the most viewed and liked video in the top 100 most popular YouTube videos.

5. Lastly, using the ‘Facebook API’, I ultimately created two bar graphs. One that shows only my Facebook activity over the course of a week during four periods of the day: 12:00am - 5:59am, 6:00am - 11:59pm, 12pm - 5:59 pm, and 6:00pm - 11:59pm. Along with a second one that is a comparison stacked bar graph that shows my Facebook activity compared to my Twitter activity. In order for the bars in my graph to not have the same color, I gave both Facebook and Twitter two different colors. An interesting finding with both of these visualizations is that I have no Facebook or Twitter activity during some hours of the day (e.g. between 6am-11:59am Thursday’s – Sunday’s).

 For my Facebook activity, my most popular times appear to be between either 12am-5:59am or 6pm-11:59pm over the course of a week. For my Twitter activity, my most popular times appear to be between either 12pm-5:59pm or 6pm-11:59pm over the course of a week. When comparing my Facebook activity to my Twitter activity, I am overall more active on Facebook.

**Instructions on how to run my code (accounts and links):**

For each API I created different “secret files”, code files, and database files. Therefore, I will go through how to run each of them. For all of these files, the code ‘plt.show()’ at the bottom will have to be uncommented in order to see the image that the code generates.

1. The Movie Database API

* In order to run my themoviedatabase.py file, you will need to create an account on the movie database and then register for an API key. The first link below takes you to the homepage where you can click the SIGN UP button on the right hand side of the screen to create an account. The second links walks you through how to generate an API key.
  + https://www.themoviedb.org/?\_dc=1512456593
  + https://developers.themoviedb.org/3/getting-started/introduction

2. Gmail and Youtube API

* In order to run both my gmail.py and youtube.py files, you will need to create a client\_secret.json file that includes all of your credentials. Following the first link below, you only have to complete Step 1. Specifically for the Gmail visualization, you need to fork and clone the git repository for WordCloud to your local repository. The second link will take you to that repository.
  + <https://developers.google.com/gmail/api/quickstart/python>
  + <https://github.com/amueller/word_cloud>

3. Facebook API

* In order to run my fbapi.py file, you will need a Facebook access token. The first link below will take you to the Graph Explorer API where you can generate your access token (make sure you have the user\_posts box checked when you generate your token). Currently, the visualization is set for the second example shown above in the visualization portion. Thus, you will also need your twitter information (consumer key, consumer secret, access token, and access token secret) to run the twitter.py file. This file will cache your data into a twitter\_data.txt file that I open in the fbapi.py file to pull the twitter data from it. The second link is where you can go to create a twitter application in order to get the necessary credentials.
  + <https://developers.facebook.com/tools/explorer>
  + apps.twitter.com

4. Spotify (spotipy) API

* In order to run my spotify.py file, you will need to log into your Spotify account and create and developer app on the Spotify website. The link to that website is below. You will need your client id and client secret from the application. Along with those credentials, you will need to specify your username, a redirect uri, and a scope (in this case “user-library-read”) to run the file.
  + <https://beta.developer.spotify.com/dashboard/>

**Documentation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Issue Description** | **Location of Resource** | **Results (did it solve the issue)?** |
| 11/13/17 | Accessing Gmail and YouTube API | <https://developers.google.com/api-client-library/python/auth/installed-app> | Yes! |
| 11/13/15 | Methods to access specific data I want from Gmail API | <https://developers.google.com/gmail/api/v1/reference/users/messages> | Yes! In the end I had to use both the .threads() and .messages() methods to get more information about the emails. |
| 11/13/17 | Methods to access most popular videos on YouTube | <https://developers.google.com/youtube/v3/guides/implementation/videos> | Yes! |
| 11/15/17 | Using the Spotipy wrapper to access Spotify API | <https://spotipy.readthedocs.io/en/latest/> | Yes! |
| 11/15/17 | Figure out pagination for spotipy wrapper | <https://stackoverflow.com/questions/39086287/spotipy-how-to-read-more-than-100-tracks-from-a-playlist> | Yes! Although I had to modify it because I did not write it as a function. |
| 11/16/17 | Accessing the top rated TV shows for the movie database | <https://developers.themoviedb.org/3/tv/get-top-rated-tv> | Yes! |
| 11/19/17 | Creating the Google Map visual and plotting data points | <http://basemaptutorial.readthedocs.io/en/latest/utilities.html> | Yes! |
| 11/20/17 | Annotating the Google Map | <http://matplotlib.org/users/annotations.html> | A little bit – it doesn’t particularly tell you that the axis are only between 0 and 1 for the .annotate method (so that took me a while to figure that out) |
| 11/26/17 | Creating a WordCloud | <https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/9781784390150/11/ch11lvl1sec71/creating-a-wordcloud> | Yes! |
| 11/27/17 | Creating a bar graph | <https://matplotlib.org/devdocs/api/_as_gen/matplotlib.pyplot.bar.html> | A little bit – it isn’t clear about how to insert the data you are using (from a list or dictionary) |
| 11/28/17 | Creating a bar graph | <https://stackoverflow.com/questions/40575067/matplotlib-bar-chart-space-out-bars> | Yes! Helped with figuring out how to add bars to the chart if my data is in a dictionary. |
| 11/28/17 | Coloring a bar graph | <http://matthiaseisen.com/pp/patterns/p0178/> | Yes! Showed me how to make the bars different colors. |
| 11/29/17 | Creating a stacked bar graph | <http://benalexkeen.com/bar-charts-in-matplotlib/> | Yes! |
| 11/29/17 | Creating a stacked bar graph – with multiple bars in one area | <https://stackoverflow.com/questions/14270391/python-matplotlib-multiple-bars> | Yes! |
| 12/1/17 | Labeling the tops of the bars in a bar graph | <https://stackoverflow.com/questions/40489821/how-to-write-text-above-the-bars-on-a-bar-plot-python> | Yes! |