question.md

Assignment 3

Question

Scrape tweets of a given hashtag from a given time period and perform a basic word frequency analysis. Show your results in a bar graph.

Code Snippets

Code snippets from previous exercises will be useful here. For the following snippets, packages needed are : matplotlib, twint, seaborn.

- To scrape tweets that contain a string using twint:

```
import twint

c = twint.Config()

# to search for tweets with a hashtag, search_string = "#hashtag"
c.Search = search_string

# format : yyyy-mm-dd (example : 2021-05-31)
c.Since = start_date

# format : yyyy-mm-dd, Until date is not included in the interval for scraping
c.Until = end_date

# maximum number of tweets to be scraped. Once limit is reached, scraping stops
c.Limit = 2000

# only consider tweets with 20 minimum likes
c.Min_likes = 20
c.Lang = "en"

# True if tweets need to be stored in a pandas dataframe
c.Pandas = True

# run the search
twint.run.Search(c)

# dataframe with tweets in it
Tweets_df = twint.storage.panda.Tweets_df
```

- To iterate through dataframe, process each tweet and get a dictionary with word counts:

```
counts = {}                                     # dictionary to maintain overall word count
for index, row in Tweets_df.iterrows():
        text = row["tweet"]

        # process text here (split, lowercase, get alphanumeric tokens, removing stop words and stem)
        # to get a list of tokens

        # iterate through list of words and update dictionary
# sort counts dictionary
```

- To plot frequency counts

```
import matplotlib.pyplot as plt
import seaborn as sns
limit = 20      # number of words to plot
keys = list(counts.keys())[0:limit]
items = [counts[key] for key in keys]
plt.figure(figsize=(10,5))
sns.barplot(items, keys, alpha=0.8)
```

```
plt.title("Top Words Overall")
plt.ylabel("Words", fontsize=12)
plt.xlabel("Counts", fontsize=12)
plt.show()
```

Helpful Tutorials

- Twint: Twitter Scraping Without Twitter’s API
- Scraping tweets using twint and analyzing with NLP