Bar Plots and Mastodon Toots
ISTA 131 Hw8, Due 4/11/2024 at 11:59 pm

Introduction. This homework introduces you to sentiment analysis and making bar plots from `pandas` data structures. You will do this by analyzing the sentiment expressed in toots from the Twitter-like platform Mastodon containing the keywords `'Trump'`, and `'Biden'` from collections of about 40 toots each made on January 29 and January 30. This technique (by pure chance, one would think) has exactly predicted the last 2 presidential elections (with Twitter data) when done right before the elections. We will compare how people are feeling about Trump and Biden (according to this dubious method) at this very early point in the presidential election cycle. Then you will make a bar plot summarizing the results.

Instructions. Create a module named `hw8.py`. Below is the spec for four functions. Implement them and upload your module to D2L Assignments. **Don't scrape tweets, they are provided in `json` files.**

Testing. Download `hw8_test.py` and auxiliary testing files and put them in the same folder as your `hw8.py` module. Each of the first two functions is worth 30% of your correctness score. The figure you create is worth 40%. You do not directly receive points for `main`, but it has to work correctly or you will get no points for your plot. You can examine the test module in a text editor to understand better what your code should do. The test module is part of the spec. The test file we will use to grade your program will be different and may uncover failings in your work not evident upon testing with the provided file. Add any necessary tests to make sure your code works in all cases.

Documentation. Your module must contain a header docstring containing your name, your section leader's name, the date, `ISTA 131 Hw8`, and a brief summary of the module. Each function must contain a docstring. Each docstring should include a description of the function's purpose, the name, type, and purpose of each parameter, and the type and meaning of the function's return value.

Grading. Your module will be graded on correctness, documentation, and coding style. Code should be clear and concise. You will only lose style points if your code is a real mess. Include inline comments to explain tricky lines and summarize sections of code.

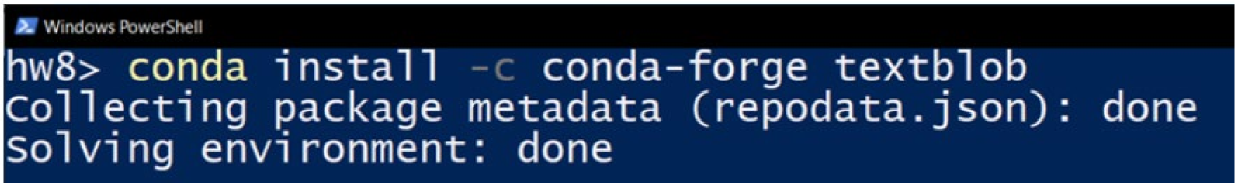
Collaboration. Collaboration is allowed. You are responsible for your learning. Depending too much on others will hurt you on the exams. "Helping" others too much harms them in reality. Cite any sources/collaborators in your header docstring. Leaving this out is dishonest.

Resources.
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.html
https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.T.html
https://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.errorbar
https://matplotlib.org/examples/api/barchart_demo.html
https://matplotlib.org/api/_as_gen/matplotlib.pyplot.bar.html
https://matplotlib.org/examples/color/named_colors.html
https://developer.twitter.com/en/docs/tweets/search/guides/standard-operators
https://developer.twitter.com/en/docs/tweets/tweet-updates
https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/post-statuses-update

https://textblob.readthedocs.io/en/dev/
For future reference, not necessarily this assignment:
https://matplotlib.org/api/patches_api.html#matplotlib.patches.Rectangle

`get_sentiment`: This function takes a filename as its sole argument. The file contains a `json` string representing a list of the text of a number of tweets. Open the file in read mode and use the `json` module to load its contents into a Python list (see the relevant notebook). Use `textblob` to perform sentiment analysis on each of the tweets, rejecting any that have both polarity and subjectivity equal to `0.0`. Return a two-element list containing the mean of the remaining polarities and the sample standard deviation of those polarities.

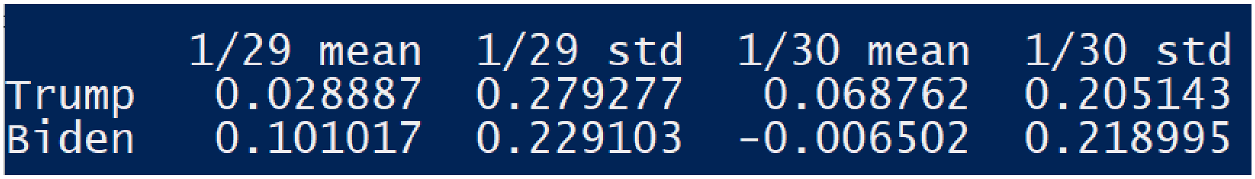
In order to use `textblob`, you will need to install it from the command line:



```
hw8> conda install -c conda-forge textblob
Collecting package metadata (repodata.json): done
Solving environment: done
```

For more information, check out the relevant notebook. If possible, use `conda` to install it, not `pip`.

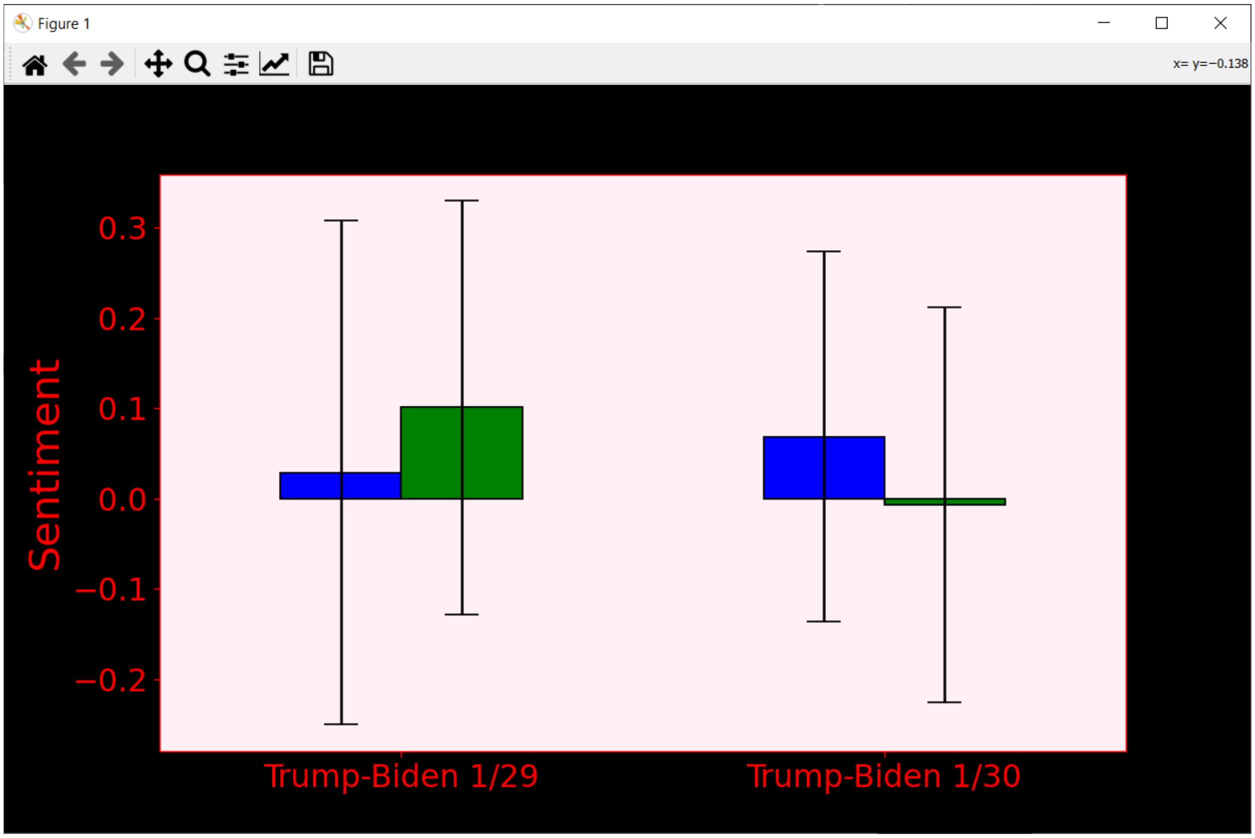
`get_ct_sentiment_frame`: This function takes no arguments and returns a `DataFrame` that looks like the image below. The `'pre_mean'` column contains the mean of the polarities of the tweet sample for each president scraped on the two days as determined by `get_sentiment`, etc.



```
       1/29 mean  1/29 std  1/30 mean  1/30 std
Trump   0.028887  0.279277   0.068762  0.205143
Biden   0.101017  0.229103  -0.006502  0.218995
```

`make_fig`: This function takes a sentiment frame (the return value from `get_ct_sentiment_frame`), and creates a figure that looks like the image at the top of the next page, immediately (without resizing). The rubric for this figure is:

- no points unless the sentiment frame passes its test
- +3 each bar (+12 total): correct size (+1) and color (+1).
- +6 error bars: all have correct vertical size (+3) and cap size (+3).
- +1 each (+3 total): *x* and *y* tick labels, *y* label are correct values.
- +1 each (+3 total): *x* and *y* tick labels, *y* label are correct sizes (tick labels 18, *y* label 24).
- +1 each (+3 total): *x* and *y* tick labels, *y* label are correct color.
- +4: spine colors are correct.
- +4: figure face color is correct.
- +3: plot background color is almost correct.
- +5: plot background color is exactly correct.
- -3: plot needs resizing to grade.



`main`: Get the sentiment frame, print a blank line, print the frame, make the figure, and display it.