Assignment 8: Event Studies and Sentiment Analysis using Python

This assignment is split into multiple parts or steps to make it easier for you to execute in step by step fashion. They are not separate assignments.

- In the end, it is one assignment and it is due on **Dec 3**. But I will allow late submissions without any penalty till **Dec 9th**
- I need to grade and submit final grades by Dec 14th, so no late submissions beyond Dec 9th would be entertained

Submission Details

- Submit through canvas
- Due on `<Dec 3rd, 11:59 pm>`
- This assignment can be done in Python, PERL, SAS, R, etc. Python is recommended for this particular assignment.
- But I am OK if you want to use R (see `https://cran.r-project.org/web/packages/tm/index.html`)
- There are a number of resources available online (some of which are highlighted in class) that can help with the assignment
- You have to submit ONLY
  - The program used to download the data
  - Note that you do not need to submit any datasets for this assignment, but please save the data as it will be used in latter parts of this assignment

Assignment 8.1

Step 1: Download Data from the SEC

The main task for the first part of the assignment is simple. You need to download 100 (random) 8-Ks for each year-quarter (not year) from the SEC website for the time period 1995:Q1 through 2018:Q4 (resulting in approximately $10,000$ 8-K documents). These documents will be analyzed in the next assignment. Note that the task is to write a script or program to do this automatically, and not to download all files manually for every year-quarter.

- The 8-K files are available on the SEC website: `https://www.sec.gov/`
- More information about accessing SEC data can be found here: `https://www.sec.gov/edgar/searchedgar/accessing-edgar-data.htm`
- Please be sure to use the *index* file (full-index)
- The directory is organized by year and quarter; thus, if you want to access data from 1995:Q2, you will have to navigate to: `https://www.sec.gov/Archives/edgar/full-index/1995/QTR2/`
- You can either download the master.idx file or master.zip (preferred)
- Note that you will need to *clean* the master.idx file (after unzipping), in order to remove the first few lines
- Use regular expressions (*regex*) to filter the forms so that you can keep only the 8-K filings
- Extract 100 random lines or companies for each year-quarter
- Extract the path name or link for the 8-K download
- Download the corresponding 8-Ks
- Create a .CSV file that keeps track of the company identifier (CIK) and the 8-K filing date (this will be used for a later assignment)
- Submit the program (but not the downloaded 8-K files)

Step 2: Event Studies

- note: You need to submit ONLY
  - The program(s) used to merge data, run event studies, and generate descriptive statistics
  - A report highlighting descriptive statistics of the event studies

The main task for this assignment is to compute abnormal stock returns and abnormal trading volume around 8-K filings. In later assignments, we will refine the information in 8-Ks further. For this assignment, however, use all the 8-K filings.

- Use the .CSV file that contains information on CIKs and 8-K filing dates (from step 1 of Assignment 8.1)

- Note: For this assignment, you just need the date of the 8-K filing for each company
- You need to compute the cumulative abnormal returns (CARs) and cumulative abnormal volumes (CAVs) around 8-K filings date; the windows are, in relation to the event date, 0, -1,+1, -2,+2, -3,+3, -5,+5
- Information for computing CARs and CAVs can be found from the following papers:
  - Are Credit Ratings Still Relevant?
    (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2023998)
  - December Doldrums, Investor Distraction, and Stock Market Reaction to Unscheduled News Events
    (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2962476)
- Once CAR and CAV is computed, check if it is different from 0 and compute some descriptive statistics of the CAR and CAV, along with the plotting the distribution of both measures
- The daily stock return information can be accessed from WRDS. The relevant file is the daily stock file (DSF) from the CRSP database.
- The key identifier in the CRSP DSF file is the PERMNO or CUSIP; one needs to go through the COMPUSTAT database (for the funda.sas7bdat file) to find a link between CIK and CUSIP
- Submit the program and the descriptive statistics for both the CAR and the CAV
- Be sure to save the final dataset that contains CAR and CAV information for each CIK-filing date pair – we will be using it for the next assignment

Assignment 8.2: Sentiment Analysis

- Note: You have to submit ONLY
  - The program used to merge data, run event studies, perform sentiment analysis, and generate descriptive statistics
  - A report highlighting descriptive statistics of the event studies organized by different quintiles of positivity

Step 1: Rudimentary Sentiment Analysis

The main task for this assignment is to compute abnormal stock returns and abnormal trading volume around 8-K filings, and study how these measures vary depending on the positivity/negativity of the filing.

- Please utilize the words list from Bill McDonald's website:
  `https://www3.nd.edu/~mcdonald/Word_Lists.html`
  - Please use the latest version of the "Master Dictionary" available
- For each downloaded 8-K filing (from Assignment I), calculate the difference bewteen the number of positive words and the number of negative words, and scale this difference by the total number of words in the document
- Sort the measure constructed in the above step into quintiles at an annual frequency
  - Note that the lowest (highest) quintile represents the most negative (most positive) document
- Generate a report highlighting the difference in descriptive statistics for CAR and CAV for the different quintiles of 8-K filings

Step 2: Advanced Sentiment Analysis

- note: You have to submit ONLY
  - The program used to merge data, run event studies, perform sentiment analysis, and generate descriptive statistics
  - A report highlighting descriptive statistics of the event studies organized by different quintiles of positivity

- **IMPORTANT**: Please note that this assignment requires significant self-study and research; it is imperative that students begin working on this assignment early

In the previous step, the “tone” of the 8-K filing was captured through simple counts of the number of negative and positive words in the filing. However, this sentiment measure is fairly imprecise. In this assignment, we will make use of natural language processing in order to more accurately capture the sentiment of the 8-K filing.

- For each downloaded 8-K filing (from Assignment I), identify the informational component of the filing (i.e., ignore the header section, and only focus on the filing information section)
    - Note that this informational component of the filing is usually in the form of paragraphs
- Break the paragraph of information into sentences, and “tokenize” these individual sentences
- Assign a “tone” value to each sentence
    - Note that NLTK assigns ‘weights’ of negativity, neutrality, and positivity to input data. The weights sum up to 1. For our example, the input data is individual sentences from the information section of the 8-K filing. You can use the compounded weight assigned to each sentence as a measure of the sentence’s tone.
- For each document, calculate the average “tone” value of all sentences in the document.
- Sort the measure constructed in the above step into quintiles at an annual frequency
    - Note that the lowest (highest) quintile represents the most negative (most positive) document
- Generate a report highlighting the difference in descriptive statistics for CAR and CAV for the different quintiles of 8-K filings
- **HINT**:
    - Using the VADER sentiment analysis toolkit from NLTK may be a good starting point

Important Resources and References

- Tutorials on "Text Classification for Sentiment Analysis" using the Natural Language Toolkit (NLTK) for Python available on StreamHacker; relevant articles and tutorials:
  - `https://streamhacker.com/` (StreamHacker website)
  - `https://streamhacker.com/2010/05/10/text-classification-sentiment-analysis-naive-bayes-classifier/` (for information regarding the Naive Bayes Classifier)
  - `https://streamhacker.com/2010/05/17/text-classification-sentiment-analysis-precision-recall/` (precision and recall)
  - `https://streamhacker.com/2010/05/24/text-classification-sentiment-analysis-stopwords-collocations/` (stopwords and collocations)
  - `https://streamhacker.com/2010/06/16/text-classification-sentiment-analysis-eliminate-low-information-features/` (eliminate low information features)
- Kaggle: `https://www.kaggle.com/ngyptr/python-nltk-sentiment-analysis`
- NLTK official documentation, with examples (most relevant):
  - `http://www.nltk.org/api/nltk.sentiment.html`
  - `http://www.nltk.org/howto/sentiment.html`
- the most straightforward implementation of the NLTK assignment can be done through the VADER sentiment analysis toolkit (available as part of NLTK): `https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/viewPaper/8109`
- VADER is based on social media data, but it's OK for the purpose of this assignment even though it is not perfectly suited for this assignment