Project Summary

Purpose

The purpose of this assignment is to better understand how sentiment within a 10-K effects the stock return around the release of the 10-K. In a high-level overview, we accomplished this by measuring the sentiment scores of the 10-Ks against multiple dictionaries to indicate if the 10-K had a more positive or negative sentiment. We then evaluated the correlation between each sentiment dictionary and the return. In the end, I found no significant correlation between my sentiment dictionaries and stock returns on the filing date.

Process

To accomplish this task, we took the following steps:

1. Extracted the list of firms listed on the S&P 500 from the S&P 500 page on wikipedia
2. Downloaded 10-K files from the SEC website during the year 2022
3. Loaded five sentiment dictionaries
4. Looped over the sample list to add accession number, filing date, positive sentiment scores per dictionary, and negative sentiment scores per dictionary
5. Downloaded list of returns for each stock in 2022
6. Merged sample list with returns list by ticker, filing date, and return date
7. Evaluated correlation per sentiment dictionary in both positive and negative by return

The Data

The Sample

Our sample is the list of stocks listed on the S&P 500 from December 28, 2022. Our total sample size is 503.

Return Variables

To find the return variables, we used the following steps:

1. Import urlopen from the rllib.request library
2. Import BytesIO from the io library
3. Copy the "crsp_2022_only.zip" file URL from the Stock Returns (CRSP) folder

4. Use the following code to read the zip file and create a varaible to save the data

```
url =
'https://github.com/LeDataSciFi/data/blob/main/Stock%20Returns%20(CRSP
raw=true'

with urlopen(url) as request:
    data = BytesIO(request.read())

with ZipFile(data) as archive:
    with archive.open(archive.namelist()[0]) as stata:
        returns = pd.read_stata(stata)
```

This saved dataset includes daily returns for each stock for 2022. For my analysis, I only want returns for the filing date of the 10-K. To do this I used a left merge on the S&P 500 data where ticker symbol matched and the return date matched the filing date.

To create a more comprehensive analysis, it's best to look at the return data on the day of the 10-K filing and at return data on days surrounding the filing date.

Sentiment Variables

To create our sentiment variables, we used five sentiment dictionaries. Two of these libraries were given to us and the other three we created ourselves. The two dictionaries given to us were the LM sentiment dictionary from researchers Loughran and McDonald and the ML sentiment dictionary from the Journal of Financial Economics.

To create the sentiment variables, we used the following steps:

1. Read the CSV files of the LM dictionary
2. Divide the LM dictionary into a positive and negative by creating a list of positive words where the positive column is greater than zero and a list of negative words where the negative column is greater than zero
3. Load the ML negative dictionary text file
4. Load the ML positive dictionary text file
5. Create text files of our sentiment dictionaries, ensuring various uses of relevant words are included (ie. -s, -ed, etc.)
6. Load the text files of the personalized sentiment dictionaries
7. For consistency, make all dictionaries lower case
8. Use len() to find the length of the document
9. Use NEAR_finder() to create positive and negative sentiment scores for each personalized dictionary
10. Divide the NEAR_finder() value by the document length to get the sentiment score. Here is an example of this code

```
sp500.loc[index,'pos_rep'] = NEAR_finder(reputation,
BHR_positive,document)[0]/doc_length
```

11. To compute sentiment scores for the LM and ML dictionaries, use the findall() function and divide by the document length. Here is an example of this code

```
LM_pos = r'\b('+'|'.join(LM_positive)+r')\b'
sp500.loc[index, 'LMpos'] = len(re.findall(LM_pos,
document))/doc_length
```

LM and ML Dictionary Statistics

Dictionary	Word Count
LM Positive	347
LM Negative	2345
ML Positive	75
ML Negative	94

Contextual Sentiments

For my contextual sentiment measures, I chose weather, natural disasters, and reputation. I chose these topics because I was particularly interested in how, if at all, these topics influence the overall sentiment of a 10-K.

Summary Statistics

Using .describe(), I can see that the mean and standard deviation of my sentiment scores and the returns are not zero. This is significant because it indicates there is variability in the data. Please see the summary statistics for each sentiment score and the returns below.

Passing the Smell Test

I do believe my contextual measures pass the "smell" test. I have variation in my measurements and many scores which were not zero. With that being said, refining the contextual sentiment dictionaries could help strengthen these results and further improve the analysis, resulting in overall less zeros in sentiment scores for individual stocks.

In [1]:

```
# Summary statistics of the sentiment scores and returns
import numpy as np
import pandas as pd
import csv
data = pd.read_csv('output/analysis_sample.csv')
data[['pos_weather', 'neg_weather', 'pos_disaster', 'neg_disaster', 'pos_rep
    'BHRpos', 'BHRneg', 'ret']].describe()
```

Out[1]:

	pos_weather	neg_weather	pos_disaster	neg_disaster	pos_rep	neg_re
count	501.000000	501.000000	501.000000	501.000000	501.000000	501.00000
mean	0.000065	0.000215	0.000117	0.000287	0.001059	0.00118
std	0.000084	0.000182	0.000086	0.000220	0.000316	0.00035
min	0.000000	0.000000	0.000000	0.000000	0.000244	0.00000
25%	0.000017	0.000085	0.000057	0.000150	0.000835	0.00094
50%	0.000042	0.000168	0.000097	0.000242	0.001020	0.00117
75%	0.000085	0.000295	0.000155	0.000376	0.001263	0.00141
max	0.000866	0.001137	0.000676	0.002705	0.002172	0.00244

Results

In [2]:

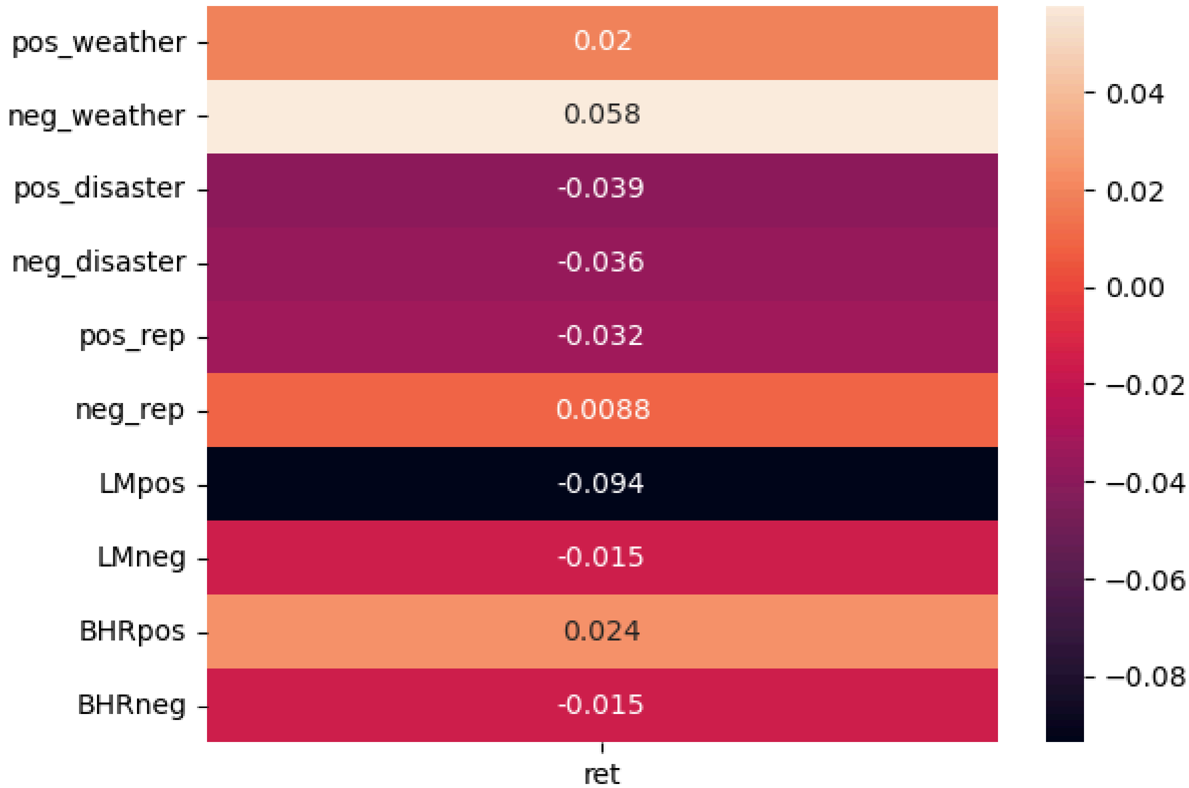
```
# Correlation between each sentiment measure and the return

import matplotlib.pyplot as plt
import seaborn as sns
corr_table = data[['pos_weather', 'neg_weather', 'pos_disaster', 'neg_disast
corr_table = corr_table.loc[['pos_weather', 'neg_weather', 'pos_disaster', '
                             'pos_rep', 'neg_rep', 'LMpos', 'LMneg', 'BHRpos

print(corr_table)

print(sns.heatmap(corr_table, annot=True))
```

```
                   ret
pos_weather   0.019569
neg_weather   0.057847
pos_disaster -0.039282
neg_disaster -0.036025
pos_rep      -0.032220
neg_rep       0.008762
LMpos        -0.093838
LMneg        -0.015421
BHRpos        0.023518
BHRneg       -0.015421
Axes(0.125,0.11;0.62x0.77)
```



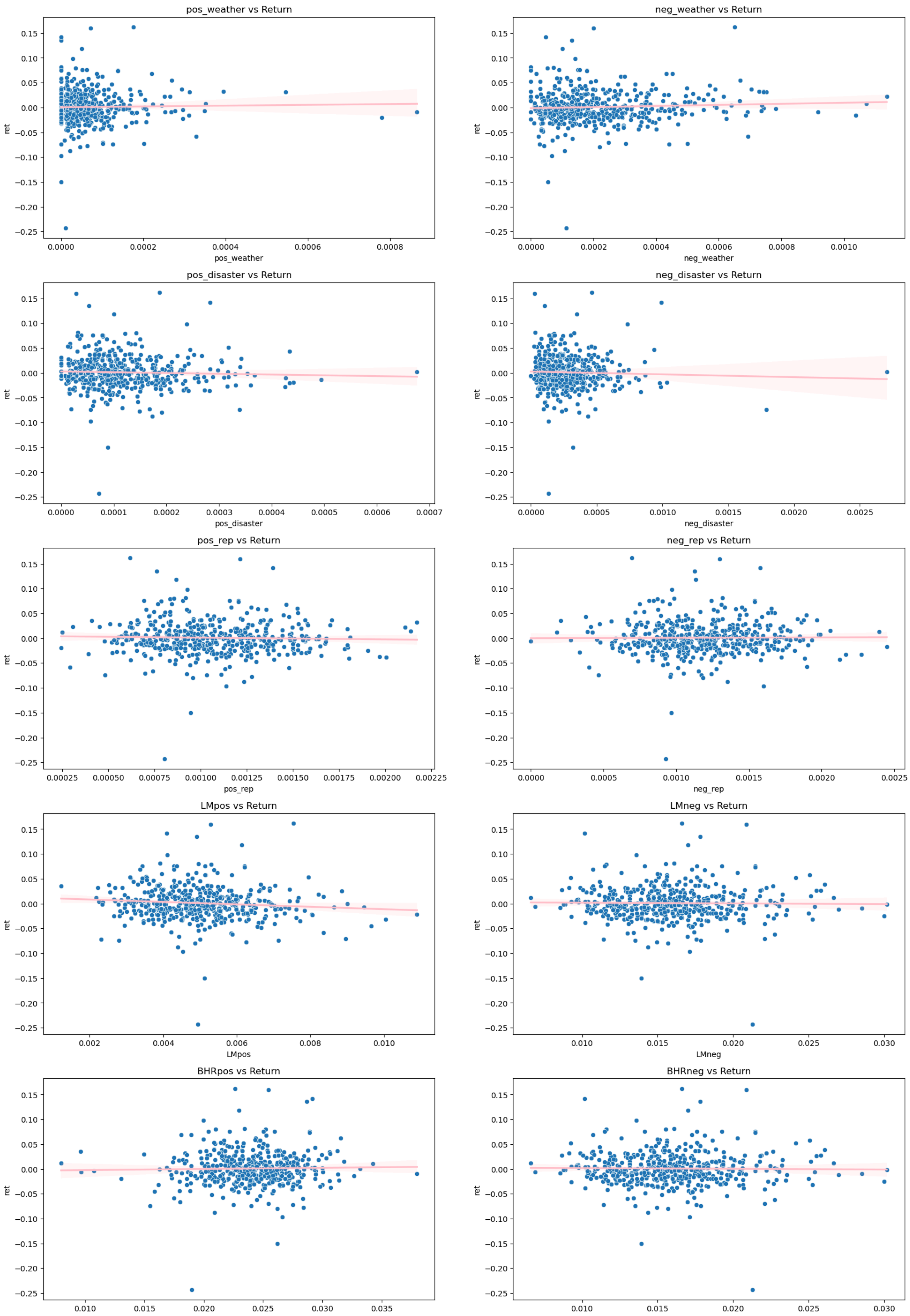
In [3]:
```
# Scatterplot of each sentiment measure vs return with regression lines

sent = ['pos_weather', 'neg_weather', 'pos_disaster', 'neg_disaster',
             'pos_rep', 'neg_rep', 'LMpos', 'LMneg', 'BHRpos', 'BHRneg']

fig, axes = plt.subplots(5, 2, figsize=(20, 30))

axes = axes.flatten()

for i, sent in enumerate(sent):
    sns.scatterplot(data=data, x=sent, y='ret', ax=axes[i])
    sns.regplot(data=data, x=sent, y='ret', ax=axes[i], scatter = False, col
    axes[i].set_title(f"{sent} vs Return")
```



Let's Explore the Results

Return Variables and LM Sentiment vs Return Variables and ML Sentiment

The return variable on the filing date has a negative correlation with LM positive, LM negative, and ML negative dictionaries. The return variable on the filing date has a positive correlation with ML positive. Additionally, LM positive has the greatest magnitude of correlation while LM negative and ML negative share the same correlation.

Garcia, Hu, and Rohrer Paper Results

My results were slightly different than the results in the Garcia, Hu, and Rohrer paper. This is likely because of a difference in the number of firms observed and controls in place that Garcia, Hu, and Rohrer used compared to us. This was likely because they wanted more data points to perform analysis on to produce more reliable results. My magnitude dispersion was generally the same, as were my signs.

Contextual Sentiment Measures

My three contextual sentiment measures do look different enough from zero that more investigation can be done. This is likely because my sentiment measures can have direct result on firm performance which may effect stock return. Refining these sentiment dictionaries would lead to more accurate results.

ML Returns

ML positive has a higher magnitude in correlation to returns compared to ML negative. ML negative is negative while ML positive is positive in signs.