BAC (Bank of America) Stock Data & Tweet Volume Analysis

This code will take an existing dataset (A count of Tweets by day across BAC owned handles)

And combine it with BAC Stock data pulled from yFinance (Primarily Closing Prices and Stock Volume)

It will go through conversions and adjustments to accurately combine these data frames, to then explore possible relationships in the data

yFinance is used to pull 2022 Stock Data for BAC.

```
# Grabbing historical data for 2022 BAC.
# It is too large for one pull, so we make two calls and then concatenate them
BAC_hist = BAC_.history(start='2022-01-01',end='2022-06-30')
BAC_hist2 = BAC_.history(start='2022-06-30',end='2022-12-31')
# followed by removing fields we wont be looking at
BAC_2022=pd.concat([BAC_hist,BAC_hist2])
BAC_2022.drop(['Stock Splits','Dividends'],axis:=1,inplace=True)
```

Date	Open	High	Low	Close	Volume
2022-01-03 00:00:00-05:00	44.064006	45.432151	43.956508	45.129204	58587900
2022-01-04 00:00:00-05:00	45.989179	47.328005	45.969634	46.898018	74606700
2022-01-05 00:00:00-05:00	47.044600	47.269366	46.086900	46.106445	57791600
2022-01-06 00:00:00-05:00	46.888244	47.064148	46.116219	47.034832	52040900

```
# Localize the time format to remove the unnecessary time-zone information
BAC_2022['Date']=BAC_2022['Date'].dt.tz_localize(None)
```

```
BAC_2022.reset_index(inplace=True)
```

Grab existing dataset of Daily Tweet Volume Count for BAC owned handles in 2022.

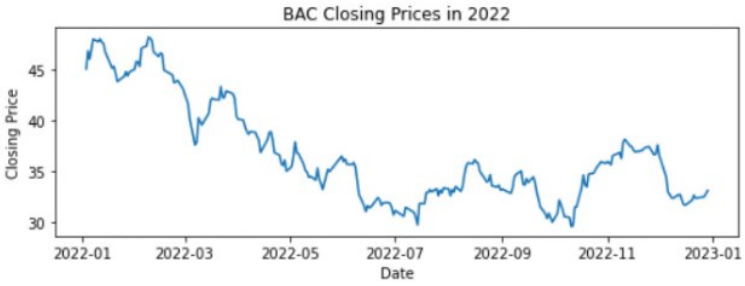
```
pd.read_csv('Counts - Sheet1.csv')
```

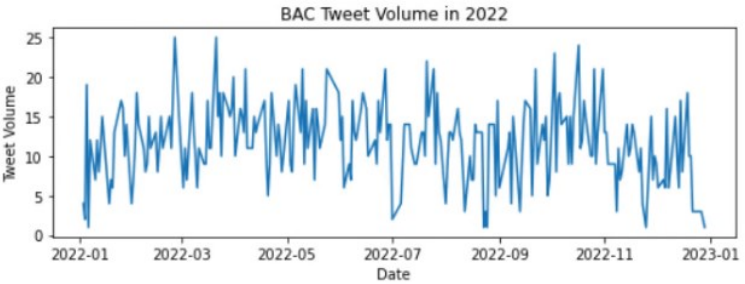
Row Labels	Count of URL
1/1/2022	2
1/3/2022	4
1/4/2022	2

Then some transformations (resetting the index of the second data frame, renaming columns, and applying datetime conversions to both dataframes) are applied.

After which, the dataframes can now be merged or joined together. Here they are joined on date, so the new dataframe can retain Tweet Volume data for days when the stock market was closed. At this time, the frame will be left to retain *NaN* values, as opposed to filling them in with zeros or any other value.
Furthermore, dates will be converted to DayOfYear for easier use with Seaborn

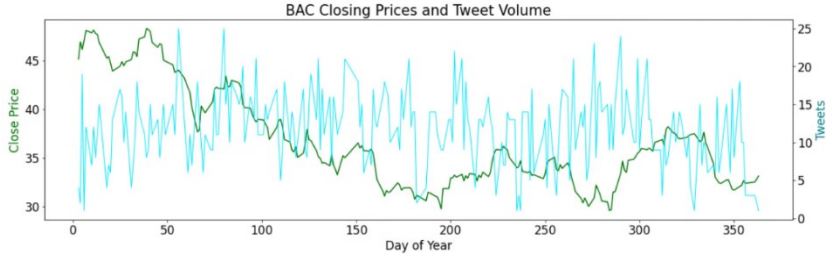
Using MatPlotLib allows for a cursory analysis of the Closing Prices and Stock volume. It appears that stock prices in 2022 for BAC followed an overall downward trend, while Tweet Volume was quite erratic.

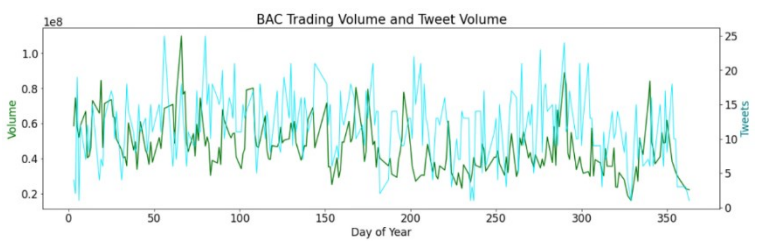




Date	Tweet_Count	Open	High	Low	Close	Volume
2022-01-01	2	NaN	NaN	NaN	NaN	NaN
2022-01-03	4	44.064006	45.432151	43.956508	45.129204	58587900.0
2022-01-04	2	45.989179	47.328005	45.969634	46.898018	74606700.0
2022-01-05	19	47.044600	47.269366	46.086900	46.106445	57791600.0

Plotting Closing Prices and Tweet Volume together does not immediately indicate a correlation between the two. This also appears to be the case for Trading Volume and Tweet Volume





However, the possibility of a lagged effect still exists:

```
# function to shift data frame and introduce a lag
def df_shift(df, target=None, lag=0):
    if not lag and not target:
        return df
    new = {}
    for c in df.columns:
        if c == target:
            new[c] = df[target]
        else:
            new[c] = df[c].shift(periods=lag)
    return  pd.DataFrame(data=new)
```

```
# Create empty lists to track results
lags=[]
volcorrs=[]
closecorrs=[]

for x in range(0,11):
    # first need to set the index to date
    # lag the counts data using the  function
    shifted_counts=df_shift(c3.set_index('Date'),lag=x)

    # Join the lagged data on the date index
    lagframe=shifted_counts.join(BAC3.set_index('Date'))

    # The next three lines take the lag amount,
    # and the correlation coefficients between
    # Tweet_Count and the other variables
    # ( That is, Count vs Volume  and Count vs Close Price)
    # and add them to the lists

    lags.append(x)
    volcorrs.append(lagframe.corr()["Tweet_Count"]['Volume'])
    closecorrs.append(lagframe.corr()["Tweet_Count"]['Close'])
```

Lags	Volume Correlation	Close Price Correlation
0	0.167527	-0.042306
1	0.068533	-0.081734
2	0.054570	-0.051190
3	0.063447	-0.052072
4	-0.007644	-0.058288
5	-0.008457	-0.009096
6	-0.029897	-0.079482
7	0.074747	-0.008018
8	0.023922	-0.073690
9	-0.041121	-0.041573
10	0.069767	-0.032279

As it can be seen from the above output:

There is no significant lagged correlation between the presented variables

Conclusion:

Here, fail to reject the null hypothesis that is:

"Social Media Volume Outputs for the BAC have no effect on Stock Performance"