In [1]:
```
# Dependencies
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import requests
import tweepy
import json
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-1-06a2388099f4> in <module>()
      5 import requests
      6 import time
----> 7 import tweepy
      8 import json

ModuleNotFoundError: No module named 'tweepy'
```

In [ ]:
```
# Import vaderSentiment Analyzer
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()
```

In [ ]:
```
# Twitter API Keys
consumer_key = "ho0syaVOcYapNnj2gtCFfF5io"
consumer_secret = "WN5xsDa2ufNVPR9MPJkHxVwTSxREYY8JxMG6sI8tQ8wrHDmITW"
access_token = "907733914470567937-7Us4vjLpARCIOrJCZoYLqBxVMqYPUUU"
access_token_secret = "ypNiR5CCZQn8clBmeX25glfGi07ZuEUiJw4xxcX319KuV"
```

In [ ]:
```
# Setup Tweepy API Authentication
#auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
#auth.set_access_token(access_token, access_token_secret)
#api = tweepy.API(auth, parser=tweepy.parsers.JSONParser())
```

In [5]:
```
# Target User
target_user = ("@BBCWorld", "@CBSNews", "@CNN", "@FoxNews", "@nytimesworld")

# Loop through each user
for user in target_user:
    # Variables for holding sentiments
    compound_list = []
    positive_list = []
    negative_list = []
    neutral_list = []
```

In [ ]:
```
# Loop through 10 pages of tweets (total 200 tweets)
    for page in tweepy.Cursor(api.user_timeline, id=user).pages(20):

        # Get all tweets from home feed
        #public_tweets = api.user_timeline(user)
        page = page[0]
        tweet = json.dumps(page._json, indent=3)
        tweet = json.loads(tweet)
        text = tweet['text']
```

In [ ]:
```
# Complete sentiment analysis
compound = analyzer.polarity_scores(target_string)
#print(compound)
compound = analyzer.polarity_scores(target_string)["compound"]
pos = analyzer.polarity_scores(target_string)["pos"]
neu = analyzer.polarity_scores(target_string)["neu"]
neg = analyzer.polarity_scores(target_string)["neg"]
```

In [ ]:
```
# Print analysis
print(target_string)
print("Compound Score: %s" % compound)
print("Positive Score: %s" % pos)
print("Neutral Score: %s" % neu)
print("Negative Score: %s" % neg)
```