In [1]:

```
import tweepy
from textblob import TextBlob
import pandas as pd
```

In [2]:

```
CONSUMER_KEY = ''
CONSUMER_SECRET = ''
ACCESS_KEY = ''
ACCESS_SECRET = ''

auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)
auth.secure = True
api = tweepy.API(auth, wait_on_rate_limit = True, wait_on_rate_limit_notify
```

In [4]:

```
searchQuery = 'Dogecoin'
retweet_filter = '-filter:retweets'

q = searchQuery+retweet_filter
tweetsPerQry = 100
fName = 'tweets.txt'
sinceId = None
```

In [5]:

```
x_id = -1
xTweets = 1000

eetCount = 0
int("Downloading max{0} tweets".format(maxTweets))
th open(fName, 'w') as f:
  while tweetCount < maxTweets:
      tweets = []
      try:
          if (max_id <=0):
              if (not sinceId):
                  new_tweets = api.search(q=q, lang = 'en', count = tweetsPe
              else:
                  new_tweets = api.search(q = q, lang = 'en', count = tweets
          else:
              if (not sinceId):
                  new_tweets = api.search(q=q, lang = 'en', count = tweetsPe
              else:
                  new_tweets = api.search(q=q, lang = 'en', count = tweetsPe

          if not new_tweets:
              print("No more tweets found")
              break
          for tweet in new_tweets:
              f.write(str(tweet.full_text.replace('\n', '').encode("utf-8"))

          tweetCount += len(new_tweets)
          print("Downloaded {0} tweets".format(tweetCount))
          max_id = new_tweets[-1].id

      except tweepy.TweepyError as e:
          print("some error: " + str(e) )
          break

int("downloaded {0} tweets, saved to {1}".format(tweetCount, fName))
```

```
Downloading max1000 tweets
Downloaded 100 tweets
Downloaded 200 tweets
Downloaded 300 tweets
Downloaded 400 tweets
Downloaded 500 tweets
Downloaded 600 tweets
Downloaded 700 tweets
Downloaded 800 tweets
Downloaded 900 tweets
Downloaded 1000 tweets
downloaded 1000 tweets, saved to tweets.txt
```

In [6]:

```
import re

def clean(tweet):
    tweet = re.sub(r'^RT[\s]+', '', tweet)
    tweet = re.sub(r'https?:\/\/.*[\r\n]*', '', tweet)
    tweet = re.sub(r'#', '', tweet)
    tweet = re.sub(r'@[A-Za-z0-9]+', '', tweet)
    return tweet

def read_tweets(file_name):
    with open(file_name, 'r') as f:
        tweets = [clean(line.strip()) for line in f]
    f.close()
    return tweets
```

In [7]:

```
tweets = read_tweets(fName)
```

In [ ]:

In [8]:

```
polarity = lambda x: TextBlob(x).sentiment.polarity
subjectivity = lambda x: TextBlob(x).sentiment.subjectivity
```

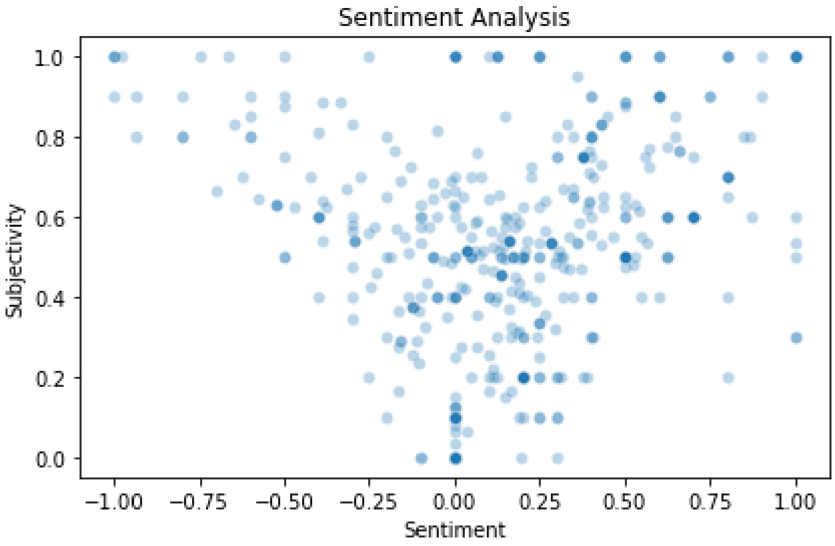
In [9]:

```
import numpy as np

tweet_polarity = np.zeros(len(tweets))
tweet_subjectivity = np.zeros(len(tweets))

for ID,tweet in enumerate(tweets):
    tweet_polarity[ID] = polarity(tweet)
    tweet_subjectivity[ID] = subjectivity(tweet)
```

In [11]:

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.scatterplot(tweet_polarity, tweet_subjectivity, alpha = .3)

plt.title("Sentiment Analysis")
plt.xlabel("Sentiment")
plt.ylabel("Subjectivity")
plt.tight_layout()
```

```
/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:36: Fut
ureWarning: Pass the following variables as keyword args: x, y. From vers
ion 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or mi
sinterpretation.
  warnings.warn(
```



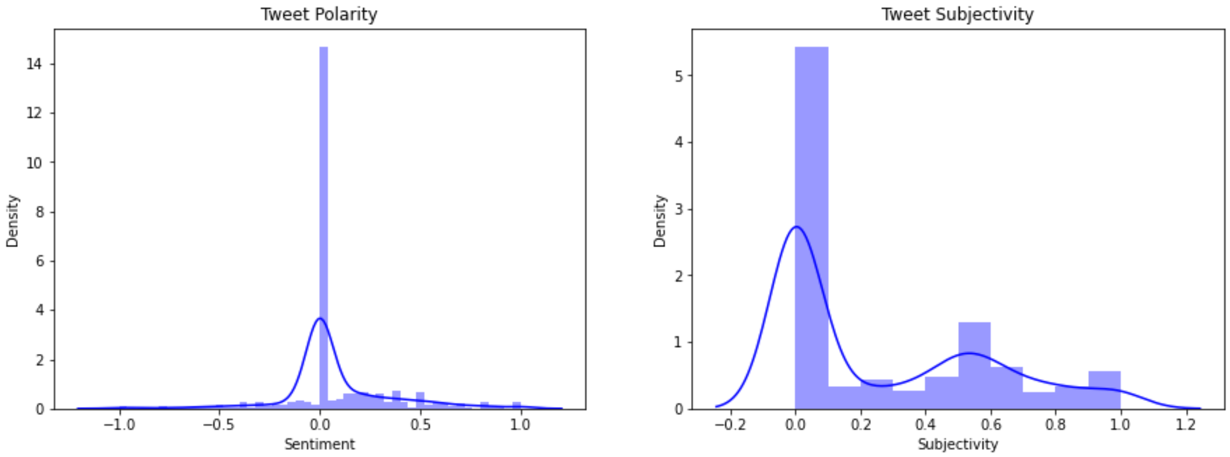
In [14]:

```
f, axs = plt.subplots(1,2, figsize = (15,5))

sns.distplot(tweet_polarity, color = 'b', ax = axs[0])
axs[0].set_title("Tweet Polarity")
axs[0].set_xlabel("Sentiment")

sns.distplot(tweet_subjectivity, color = 'b', ax = axs[1])
axs[1].set_title("Tweet Subjectivity")
axs[1].set_xlabel("Subjectivity")
```

```
/opt/anaconda3/lib/python3.8/site-packages/seaborn/distributions.py:2551:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figur
e-level function with similar flexibility) or `histplot` (an axes-level f
unction for histograms).
  warnings.warn(msg, FutureWarning)
/opt/anaconda3/lib/python3.8/site-packages/seaborn/distributions.py:2551:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figur
e-level function with similar flexibility) or `histplot` (an axes-level f
unction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[14]:

```
Text(0.5, 0, 'Subjectivity')
```



In [55]:

```
p = pd.DataFrame({'polarity': tweet_polarity})
p['polarity'].value_counts(bins = 10)
```

Out[55]:

```
(-0.2, 0.0]       623
(0.0, 0.2]        113
(0.2, 0.4]         90
(0.4, 0.6]         58
(0.6, 0.8]         34
(-0.4, -0.2]       28
(-0.6, -0.4]       18
(0.8, 1.0]         17
(-1.003, -0.8]     10
(-0.8, -0.6]        9
Name: polarity, dtype: int64
```

In [23]:

```python
import yfinance as yf

def yahoo_history_scraper(ticker, period, start, end):
    '''
    Creates a pandas dataframe for specified stock from start to end by spe
    Parameters:
        ticker (str) : stock ticker from yahoo
        period (str) : '1d', '1h', etc.
        start (str) : 'YYYY-MM-DD'
        end (str) : 'YYYY-MM-DD'

    Returns:
        DataFrame and csv with features Open High Low Close Volume Dividend
    '''
    ticker_symbol = ticker

    tickerData = yf.Ticker(ticker_symbol)

    tickerDf = tickerData.history(period = period, start = start, end = end

    return tickerDf


doge = yahoo_history_scraper('DOGE-USD', '1h', '2021-04-01', '2021-05-03')
```

In [24]:

```python
doge.head()
```

Out[24]:

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2021-04-01	0.053655	0.070111	0.053644	0.061986	5816046822	0	0
2021-04-02	0.061968	0.062249	0.057333	0.057664	2166925111	0	0
2021-04-03	0.057658	0.059484	0.055804	0.055804	1136931403	0	0
2021-04-04	0.055776	0.058107	0.055295	0.057404	938035097	0	0
2021-04-05	0.057411	0.060153	0.056435	0.059696	1513832721	0	0

In [26]:

```python
doge.dtypes
```

Out[26]:

```
Open            float64
High            float64
Low             float64
Close           float64
Volume            int64
Dividends         int64
Stock Splits      int64
dtype: object
```

In [31]:

```
# Make the date feature
doge.reset_index(level = 'Date')
```

Out[31]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2021-04-01	0.053655	0.070111	0.053644	0.061986	5816046822	0	0
1	2021-04-02	0.061968	0.062249	0.057333	0.057664	2166925111	0	0
2	2021-04-03	0.057658	0.059484	0.055804	0.055804	1136931403	0	0
3	2021-04-04	0.055776	0.058107	0.055295	0.057404	938035097	0	0
4	2021-04-05	0.057411	0.060153	0.056435	0.059696	1513832721	0	0
5	2021-04-06	0.059689	0.065050	0.058094	0.064458	2608613912	0	0
6	2021-04-07	0.064454	0.067299	0.057641	0.059028	2857060849	0	0
7	2021-04-08	0.059036	0.061745	0.058817	0.061464	1055257541	0	0
8	2021-04-09	0.061463	0.063910	0.060296	0.061684	1060465774	0	0
9	2021-04-10	0.061591	0.065068	0.060818	0.063845	1369793118	0	0
10	2021-04-11	0.063749	0.079243	0.063203	0.074649	6646627317	0	0
11	2021-04-12	0.074544	0.074986	0.068889	0.070767	2687720380	0	0
12	2021-04-13	0.070678	0.095181	0.070678	0.093445	7176815090	0	0
13	2021-04-14	0.094256	0.142510	0.092452	0.121517	20977679327	0	0
14	2021-04-15	0.121167	0.187326	0.120736	0.182207	17916620386	0	0
15	2021-04-16	0.181587	0.437700	0.180488	0.365870	69410680685	0	0
16	2021-04-17	0.366098	0.374455	0.236524	0.284173	31413404067	0	0
17	2021-04-18	0.285105	0.349918	0.246440	0.320475	23464350388	0	0
18	2021-04-19	0.320466	0.431751	0.309080	0.407318	30641432779	0	0
19	2021-04-20	0.408435	0.422347	0.271618	0.319500	28130646191	0	0
20	2021-04-21	0.316953	0.346966	0.297036	0.306925	12723463706	0	0
21	2021-04-22	0.305990	0.308757	0.256276	0.260967	11879823706	0	0
22	2021-04-23	0.261023	0.268378	0.163669	0.248509	16165566164	0	0
23	2021-04-24	0.249544	0.289390	0.229891	0.270212	11057578568	0	0
24	2021-04-25	0.269789	0.288771	0.226301	0.251111	6849914675	0	0
25	2021-04-26	0.251240	0.280452	0.248026	0.270674	5118886527	0	0
26	2021-04-27	0.271427	0.279629	0.264928	0.272188	3590611310	0	0
27	2021-04-28	0.272273	0.344662	0.256231	0.323682	14464343767	0	0
28	2021-04-29	0.323232	0.323881	0.296904	0.305169	5027354503	0	0
29	2021-04-30	0.304702	0.339757	0.302981	0.337561	5290390982	0	0
30	2021-05-01	0.337559	0.397135	0.328290	0.392987	10123343703	0	0
31	2021-05-02	0.391087	0.400029	0.364874	0.376046	8040406918	0	0

In [37]:

```
doge['Date'] = doge.index
```

```
In [43]: doge
```

Out[43]:

	Open	High	Low	Close	Volume	Dividends	Stock Splits	index	Date
Date									
2021-04-01	0.053655	0.070111	0.053644	0.061986	5816046822	0	0	2021-04-01	2021-04-01
2021-04-02	0.061968	0.062249	0.057333	0.057664	2166925111	0	0	2021-04-02	2021-04-02
2021-04-03	0.057658	0.059484	0.055804	0.055804	1136931403	0	0	2021-04-03	2021-04-03
2021-04-04	0.055776	0.058107	0.055295	0.057404	938035097	0	0	2021-04-04	2021-04-04
2021-04-05	0.057411	0.060153	0.056435	0.059696	1513832721	0	0	2021-04-05	2021-04-05
2021-04-06	0.059689	0.065050	0.058094	0.064458	2608613912	0	0	2021-04-06	2021-04-06
2021-04-07	0.064454	0.067299	0.057641	0.059028	2857060849	0	0	2021-04-07	2021-04-07
2021-04-08	0.059036	0.061745	0.058817	0.061464	1055257541	0	0	2021-04-08	2021-04-08
2021-04-09	0.061463	0.063910	0.060296	0.061684	1060465774	0	0	2021-04-09	2021-04-09
2021-04-10	0.061591	0.065068	0.060818	0.063845	1369793118	0	0	2021-04-10	2021-04-10
2021-04-11	0.063749	0.079243	0.063203	0.074649	6646627317	0	0	2021-04-11	2021-04-11
2021-04-12	0.074544	0.074986	0.068889	0.070767	2687720380	0	0	2021-04-12	2021-04-12
2021-04-13	0.070678	0.095181	0.070678	0.093445	7176815090	0	0	2021-04-13	2021-04-13
2021-04-14	0.094256	0.142510	0.092452	0.121517	20977679327	0	0	2021-04-14	2021-04-14
2021-04-15	0.121167	0.187326	0.120736	0.182207	17916620386	0	0	2021-04-15	2021-04-15
2021-04-16	0.181587	0.437700	0.180488	0.365870	69410680685	0	0	2021-04-16	2021-04-16
2021-04-17	0.366098	0.374455	0.236524	0.284173	31413404067	0	0	2021-04-17	2021-04-17
2021-04-18	0.285105	0.349918	0.246440	0.320475	23464350388	0	0	2021-04-18	2021-04-18
2021-04-19	0.320466	0.431751	0.309080	0.407318	30641432779	0	0	2021-04-19	2021-04-19
2021-04-20	0.408435	0.422347	0.271618	0.319500	28130646191	0	0	2021-04-20	2021-04-20

	Open	High	Low	Close	Volume	Dividends	Stock Splits	index	Date
Date									
2021-04-21	0.316953	0.346966	0.297036	0.306925	12723463706	0	0	2021-04-21	2021-04-21
2021-04-22	0.305990	0.308757	0.256276	0.260967	11879823706	0	0	2021-04-22	2021-04-22
2021-04-23	0.261023	0.268378	0.163669	0.248509	16165566164	0	0	2021-04-23	2021-04-23
2021-04-24	0.249544	0.289390	0.229891	0.270212	11057578568	0	0	2021-04-24	2021-04-24
2021-04-25	0.269789	0.288771	0.226301	0.251111	6849914675	0	0	2021-04-25	2021-04-25
2021-04-26	0.251240	0.280452	0.248026	0.270674	5118886527	0	0	2021-04-26	2021-04-26
2021-04-27	0.271427	0.279629	0.264928	0.272188	3590611310	0	0	2021-04-27	2021-04-27
2021-04-28	0.272273	0.344662	0.256231	0.323682	14464343767	0	0	2021-04-28	2021-04-28
2021-04-29	0.323232	0.323881	0.296904	0.305169	5027354503	0	0	2021-04-29	2021-04-29
2021-04-30	0.304702	0.339757	0.302981	0.337561	5290390982	0	0	2021-04-30	2021-04-30
2021-05-01	0.337559	0.397135	0.328290	0.392987	10123343703	0	0	2021-05-01	2021-05-01
2021-05-02	0.391087	0.400029	0.364874	0.376046	8040406918	0	0	2021-05-02	2021-05-02

```
sns.lineplot(x='Date', y='High', data = doge)
plt.xticks(rotation = 70);
```

I thought turning the Twitter data into .csv and .txt files would save me work but really just caused much more work in the end, so I scraped them and used this script.

We can see that dogecoin increased by approximately 900% over the first half of the month of April. The tweets revolving around Dogecoin for the majority part have a neutral sentiment, with approximately 73% falling within .2 of 0 sentiment. However, 623 are in the negative range which may explain the drop off from the peaks in the chart. The fact that the majority of these tweets are not subjective either makes the negativity hold more value. It would be very helpful to see how the shift of sentiment is directly affecting the price of Doge by having more segments of time. Twitter limiting only 7 days back certainly is an obstacle in this study. Being able to see sentiment values over more weeks would give a much better understanding of the relationship of crypto and Twitter. Even if the majority was still negative, but just less negative, that momentum we can assume would lead to a price increase at some point, but we would need more data to back that up.

In [ ]: