Twitter API Retrieval System for Sentimental Analysis on Current Events

Introduction

Scope and Why?:

The idea that we came up with and wanted to research for a better understanding has to do with Tweets that individuals all over the U.S send out. Following the recent Boeing 737 Max 8 accidents was tragic and very unfortunate. We decided to implement a Twitter analysis of the tweets that people are saying about the tragedy.

Another topic we decided to touch base on was the New Zealand Attack. With this incident we also decided to gather tweets on the New Zealand attack.

The collection of tweets for two tragic incidents is the analysis of tragic events that we would like to analyze and see what kind of tweets are being said about them. Do people tweet Positive things about them ? Negative? Or just downright neutral?

For the tweet retrieval we used specific key words for gathering appropriate tweets that are relevant to what we are looking for. We used key words such as, "Boeing 737 Max 8" and "New Zealand Attack"

With these key words that we implement in our gathering of data, we will refine what we are looking for. Following the refinement of data acquisition, we can now begin our analyzing of Tweets.

Included in our gathering of data we searched for specifc words in the tweets that indicated two possible emotions:

Positive emotions - (ex. Proud, Happy, Content, Satisfied, Enjoyed, Great) and other possible references.
Negative emotions - (ex. Dissapointed, Angry, Upset, Unsatisfied, Horrible) and other references.

Relevant Data:

Variables-
The fields that we extracted from Tweepy, were (Created_At, Tweet, Likes, Username, Retweet_Count)
Specification of Variables:
Created_At: The initial time the tweet was uploaded to Twitter and retrieved by the API

Tweet: The actual tweet, sent by user.

Likes: The number of likes on the tweet.

Username: The name of the person's account.

Retweet_Count: The number of retweets on the tweet.

Program Features:

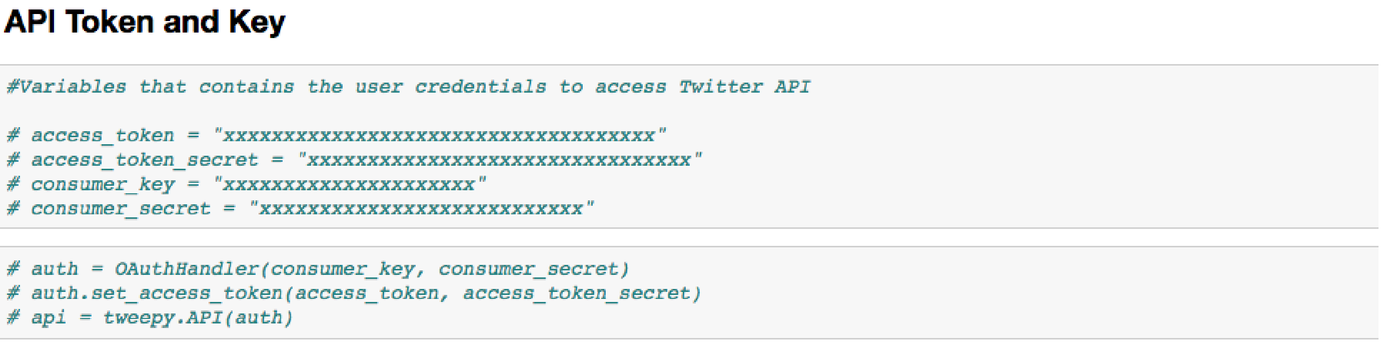
Accessing Twitter API –

To access the Twitter API, we had to request permission at Twitter Development, included in the request
we needed to include our Twitter account and explain what the reasoning was for the access of the API.

Once access was granted we were given:

Consumer API keys and Access Token.

With the keys and tokens we were able to request data from the API.

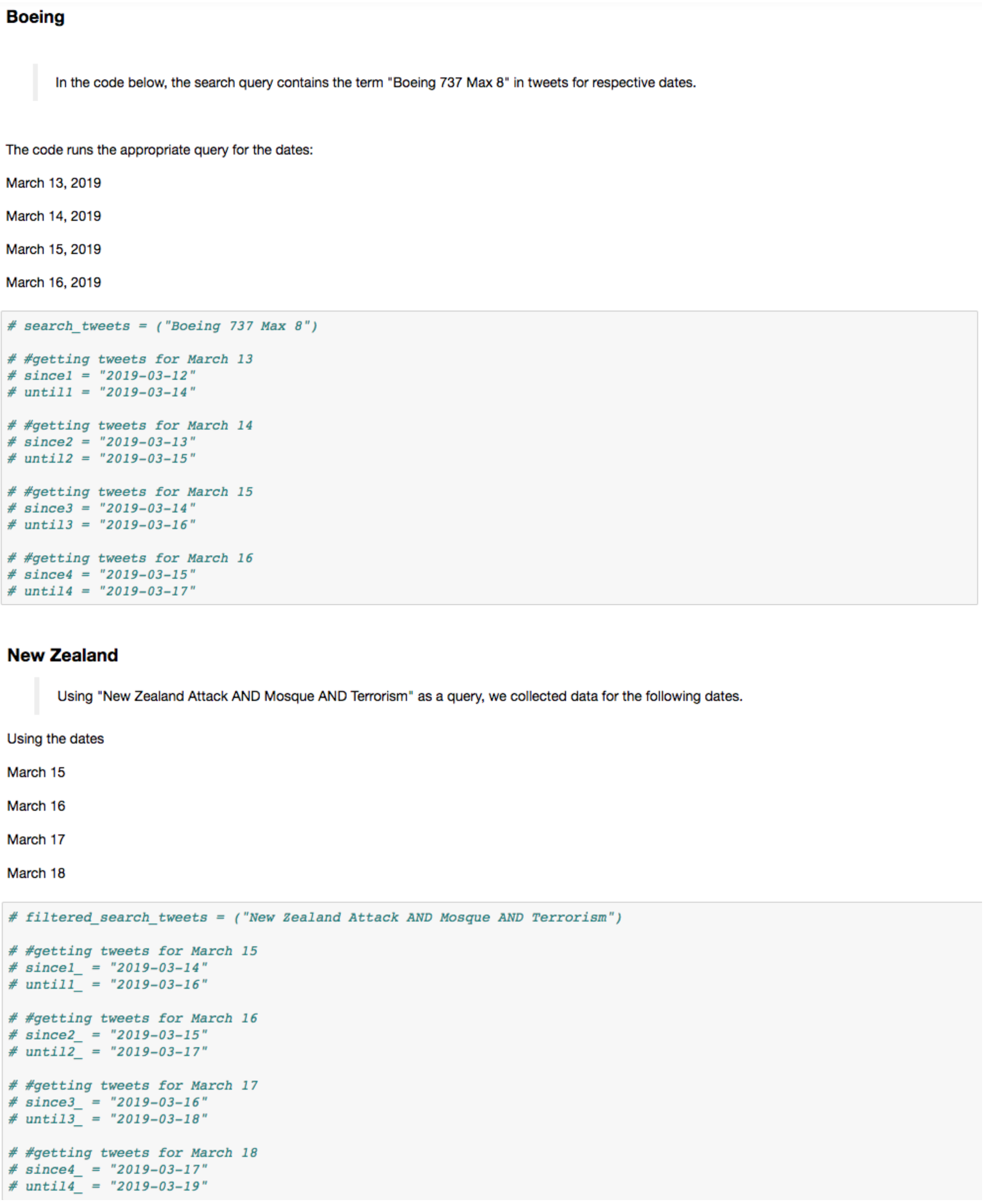


API Token and Key

```
#Variables that contains the user credentials to access Twitter API

# access_token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
# access_token_secret = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
# consumer_key = "xxxxxxxxxxxxxxxxxxxx"
# consumer_secret = "xxxxxxxxxxxxxxxxxxxxxxxxxx"
```

```
# auth = OAuthHandler(consumer_key, consumer_secret)
# auth.set_access_token(access_token, access_token_secret)
# api = tweepy.API(auth)
```

Search Query and Appropriate Dates:



Boeing

> In the code below, the search query contains the term "Boeing 737 Max 8" in tweets for respective dates.

The code runs the appropriate query for the dates:

March 13, 2019

March 14, 2019

March 15, 2019

March 16, 2019

```
# search_tweets = ("Boeing 737 Max 8")

# #getting tweets for March 13
# since1 = "2019-03-12"
# until1 = "2019-03-14"

# #getting tweets for March 14
# since2 = "2019-03-13"
# until2 = "2019-03-15"

# #getting tweets for March 15
# since3 = "2019-03-14"
# until3 = "2019-03-16"

# #getting tweets for March 16
# since4 = "2019-03-15"
# until4 = "2019-03-17"
```

New Zealand

> Using "New Zealand Attack AND Mosque AND Terrorism" as a query, we collected data for the following dates.

Using the dates

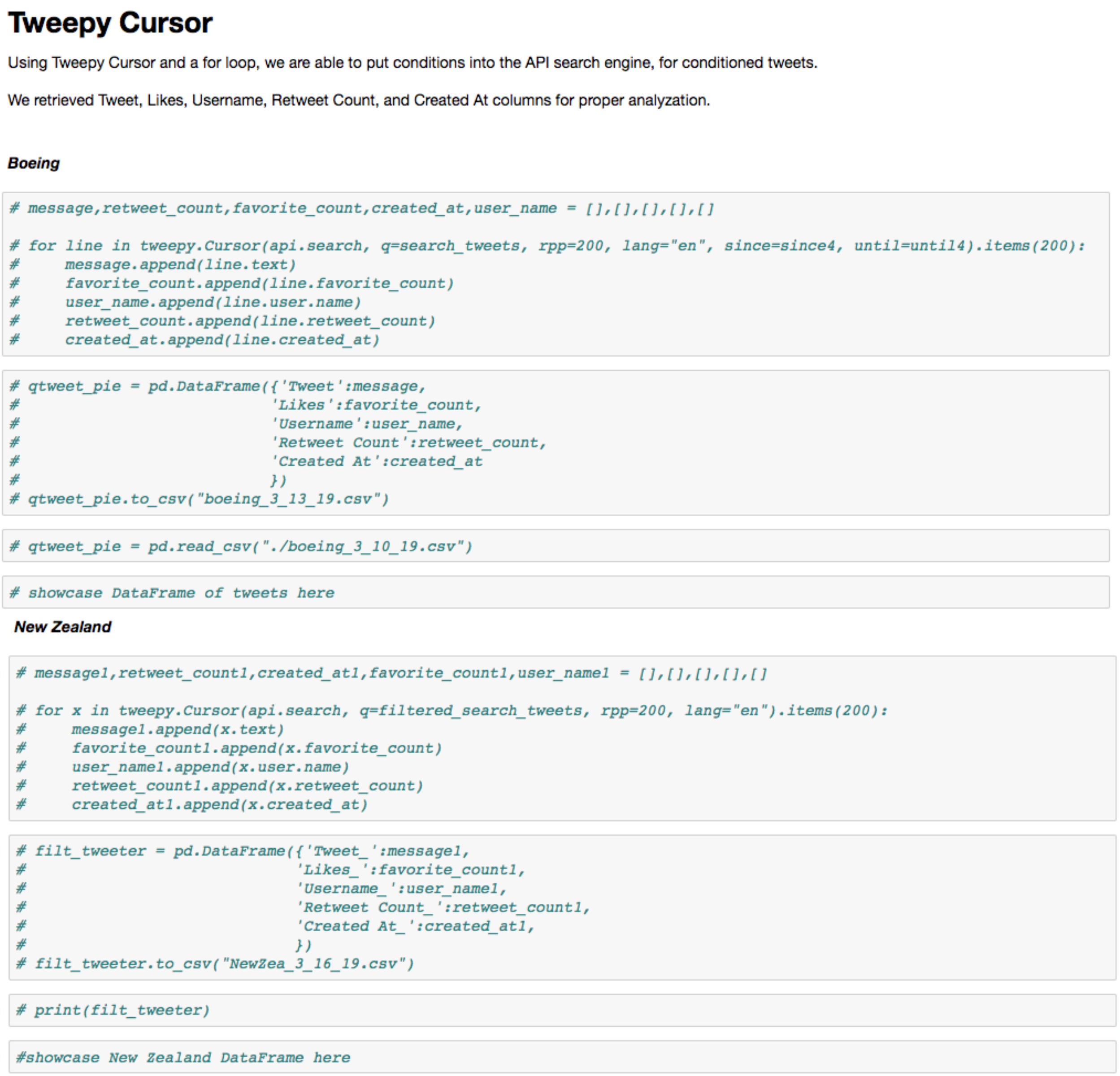
March 15

March 16

March 17

March 18

```
# filtered_search_tweets = ("New Zealand Attack AND Mosque AND Terrorism")

# #getting tweets for March 15
# since1_ = "2019-03-14"
# until1_ = "2019-03-16"

# #getting tweets for March 16
# since2_ = "2019-03-15"
# until2_ = "2019-03-17"

# #getting tweets for March 17
# since3_ = "2019-03-16"
# until3_ = "2019-03-18"

# #getting tweets for March 18
# since4_ = "2019-03-17"
# until4_ = "2019-03-19"
```



Tweepy Cursor

Using Tweepy Cursor and a for loop, we are able to put conditions into the API search engine, for conditioned tweets.

We retrieved Tweet, Likes, Username, Retweet Count, and Created At columns for proper analyzation.

Boeing

```
# message,retweet_count,favorite_count,created_at,user_name = [],[],[],[],[]

# for line in tweepy.Cursor(api.search, q=search_tweets, rpp=200, lang="en", since=since4, until=until4).items(200):
#     message.append(line.text)
#     favorite_count.append(line.favorite_count)
#     user_name.append(line.user.name)
#     retweet_count.append(line.retweet_count)
#     created_at.append(line.created_at)
```

```
# qtweet_pie = pd.DataFrame({'Tweet':message,
#                            'Likes':favorite_count,
#                            'Username':user_name,
#                            'Retweet Count':retweet_count,
#                            'Created At':created_at
#                            })
# qtweet_pie.to_csv("boeing_3_13_19.csv")
```

```
# qtweet_pie = pd.read_csv("./boeing_3_10_19.csv")
```

```
# showcase DataFrame of tweets here
```

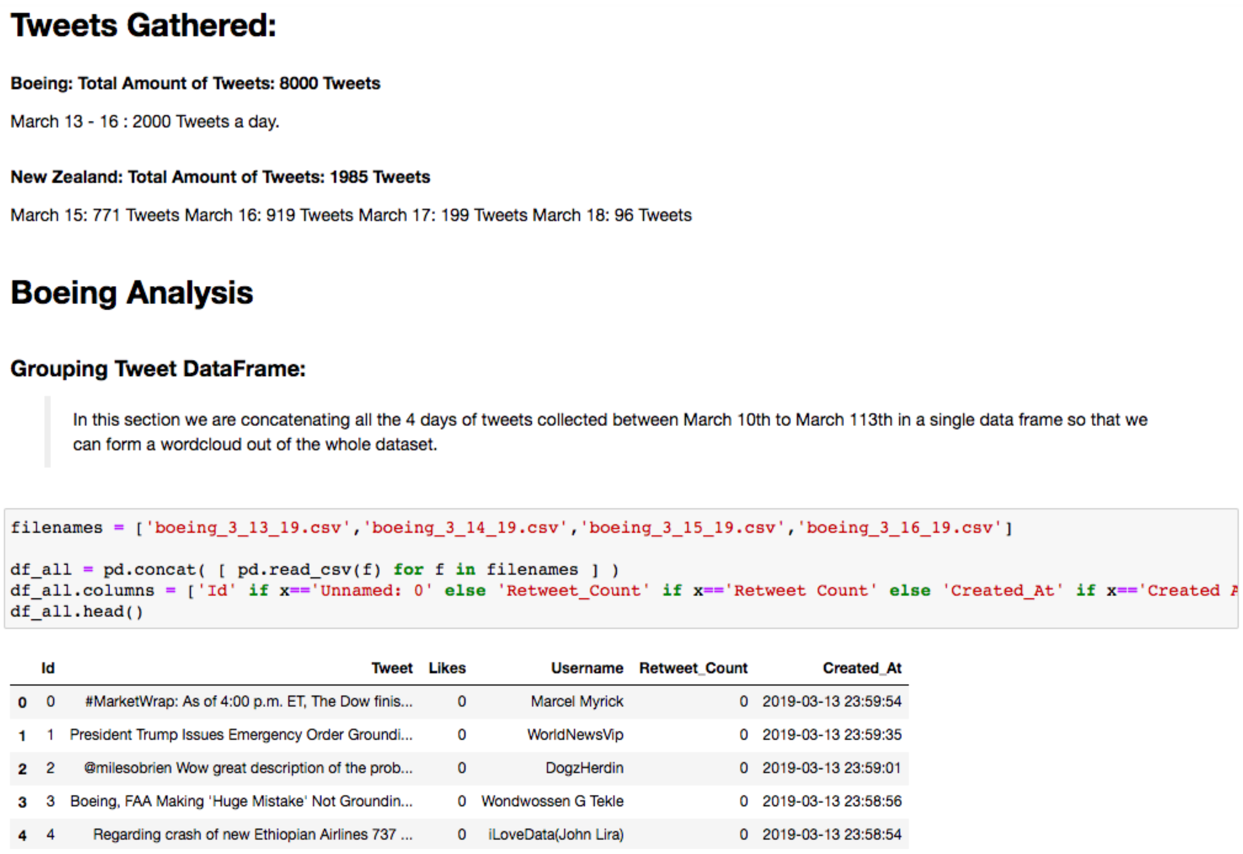
New Zealand

```
# message1,retweet_count1,created_at1,favorite_count1,user_name1 = [],[],[],[],[]

# for x in tweepy.Cursor(api.search, q=filtered_search_tweets, rpp=200, lang="en").items(200):
#     message1.append(x.text)
#     favorite_count1.append(x.favorite_count)
#     user_name1.append(x.user.name)
#     retweet_count1.append(x.retweet_count)
#     created_at1.append(x.created_at)
```

```
# filt_tweeter = pd.DataFrame({'Tweet_':message1,
#                              'Likes_':favorite_count1,
#                              'Username_':user_name1,
#                              'Retweet Count_':retweet_count1,
#                              'Created At_':created_at1,
#                              })
# filt_tweeter.to_csv("NewZea_3_16_19.csv")
```

```
# print(filt_tweeter)
```

```
#showcase New Zealand DataFrame here
```



Tweets Gathered:

Boeing: Total Amount of Tweets: 8000 Tweets

March 13 - 16 : 2000 Tweets a day.

New Zealand: Total Amount of Tweets: 1985 Tweets

March 15: 771 Tweets March 16: 919 Tweets March 17: 199 Tweets March 18: 96 Tweets

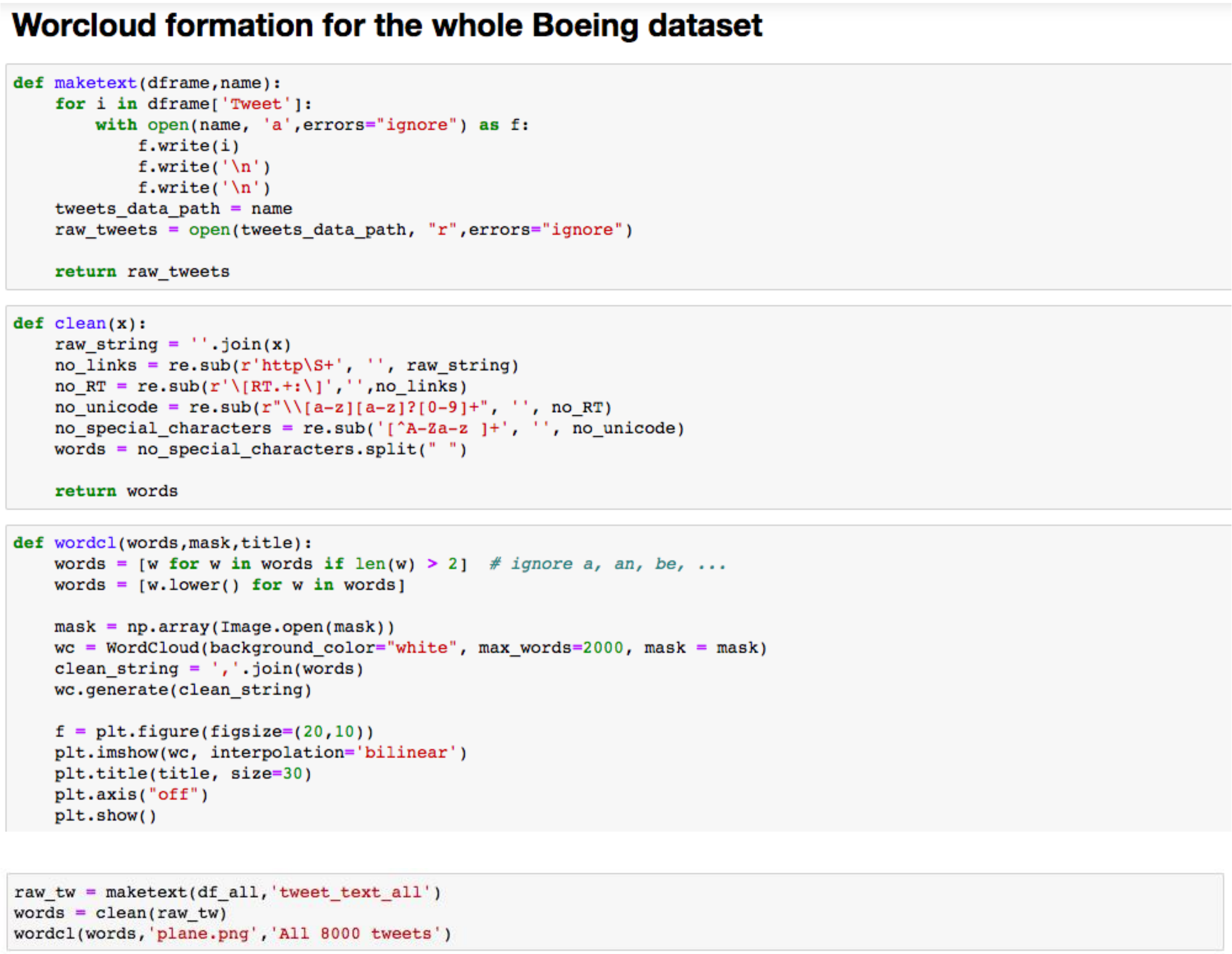
Boeing Analysis

Grouping Tweet DataFrame:

> In this section we are concatenating all the 4 days of tweets collected between March 10th to March 113th in a single data frame so that we can form a wordcloud out of the whole dataset.

```
filenames = ['boeing_3_13_19.csv','boeing_3_14_19.csv','boeing_3_15_19.csv','boeing_3_16_19.csv']

df_all = pd.concat( [ pd.read_csv(f) for f in filenames ] )
df_all.columns = ['Id' if x=='Unnamed: 0' else 'Retweet_Count' if x=='Retweet Count' else 'Created_At' if x=='Created A
df_all.head()
```

	Id	Tweet	Likes	Username	Retweet_Count	Created_At
0	0	#MarketWrap: As of 4:00 p.m. ET, The Dow finis...	0	Marcel Myrick	0	2019-03-13 23:59:54
1	1	President Trump Issues Emergency Order Groundi...	0	WorldNewsVip	0	2019-03-13 23:59:35
2	2	@milesobrien Wow great description of the prob...	0	DogzHerdin	0	2019-03-13 23:59:01
3	3	Boeing, FAA Making 'Huge Mistake' Not Groundin...	0	Wondwossen G Tekle	0	2019-03-13 23:58:56
4	4	Regarding crash of new Ethiopian Airlines 737 ...	0	iLoveData(John Lira)	0	2019-03-13 23:58:54



Worcloud formation for the whole Boeing dataset

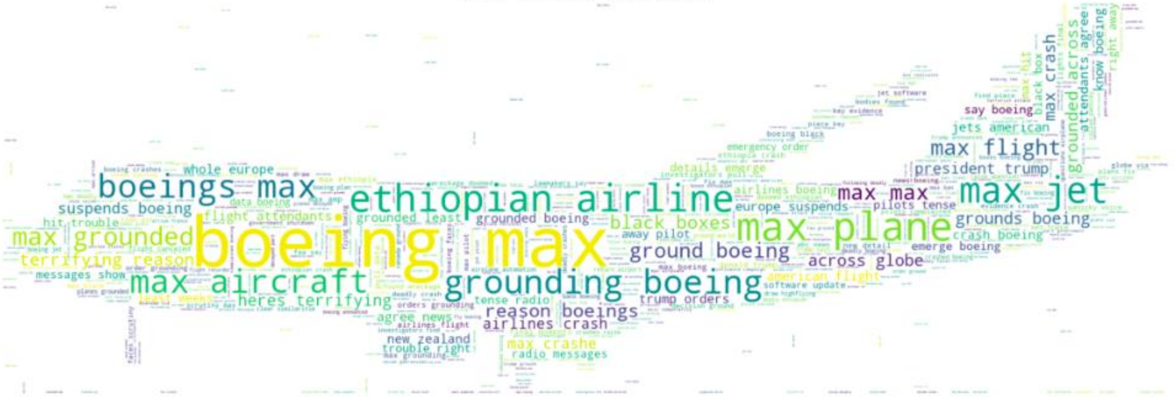
```python
def maketext(dframe,name):
    for i in dframe['Tweet']:
        with open(name, 'a',errors="ignore") as f:
            f.write(i)
            f.write('\n')
            f.write('\n')
    tweets_data_path = name
    raw_tweets = open(tweets_data_path, "r",errors="ignore")

    return raw_tweets
```

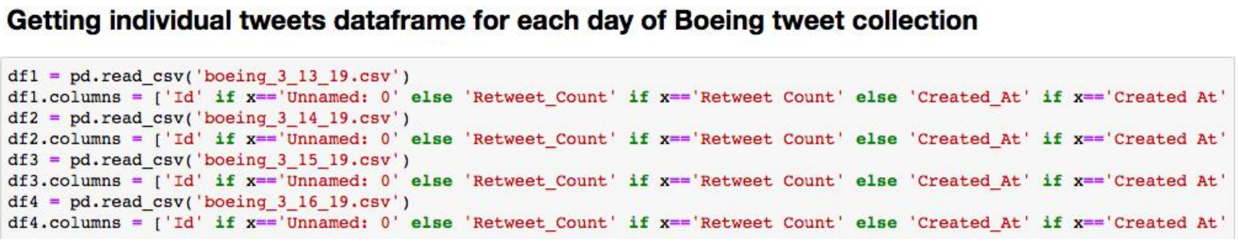
```python
def clean(x):
    raw_string = ''.join(x)
    no_links = re.sub(r'http\S+', '', raw_string)
    no_RT = re.sub(r'\[RT.+:\]',' ',no_links)
    no_unicode = re.sub(r"\\[a-z][a-z]?[0-9]+", '', no_RT)
    no_special_characters = re.sub('[^A-Za-z ]+', '', no_unicode)
    words = no_special_characters.split(" ")

    return words
```

```python
def wordcl(words,mask,title):
    words = [w for w in words if len(w) > 2]  # ignore a, an, be, ...
    words = [w.lower() for w in words]

    mask = np.array(Image.open(mask))
    wc = WordCloud(background_color="white", max_words=2000, mask = mask)
    clean_string = ','.join(words)
    wc.generate(clean_string)

    f = plt.figure(figsize=(20,10))
    plt.imshow(wc, interpolation='bilinear')
    plt.title(title, size=30)
    plt.axis("off")
    plt.show()
```

```python
raw_tw = maketext(df_all,'tweet_text_all')
words = clean(raw_tw)
wordcl(words,'plane.png','All 8000 tweets')
```

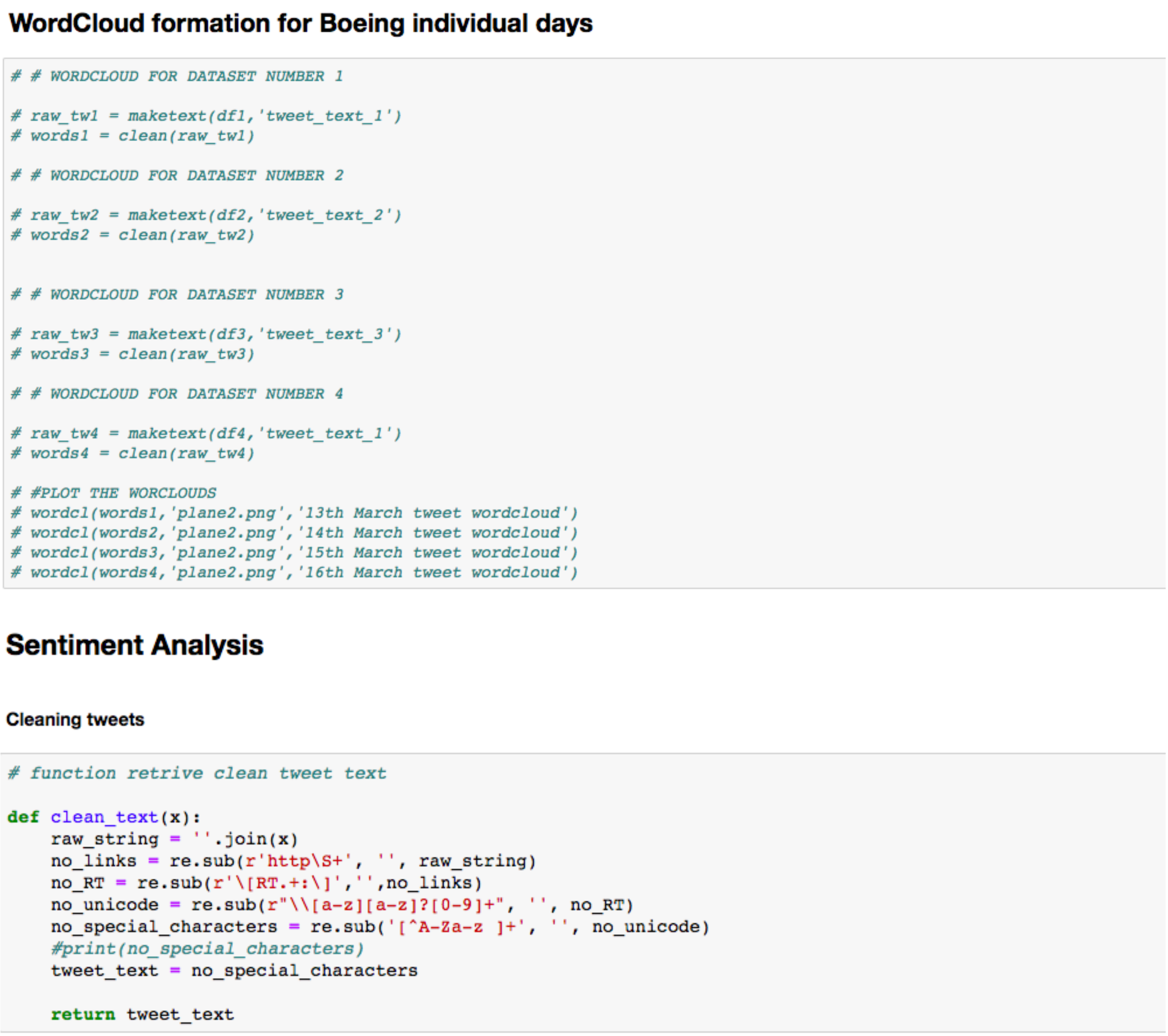
All 8000 tweets





Getting individual tweets dataframe for each day of Boeing tweet collection

```python
df1 = pd.read_csv('boeing_3_13_19.csv')
df1.columns = ['Id' if x=='Unnamed: 0' else 'Retweet_Count' if x=='Retweet Count' else 'Created_At' if x=='Created At'
df2 = pd.read_csv('boeing_3_14_19.csv')
df2.columns = ['Id' if x=='Unnamed: 0' else 'Retweet_Count' if x=='Retweet Count' else 'Created_At' if x=='Created At'
df3 = pd.read_csv('boeing_3_15_19.csv')
df3.columns = ['Id' if x=='Unnamed: 0' else 'Retweet_Count' if x=='Retweet Count' else 'Created_At' if x=='Created At'
df4 = pd.read_csv('boeing_3_16_19.csv')
df4.columns = ['Id' if x=='Unnamed: 0' else 'Retweet_Count' if x=='Retweet Count' else 'Created_At' if x=='Created At'
```



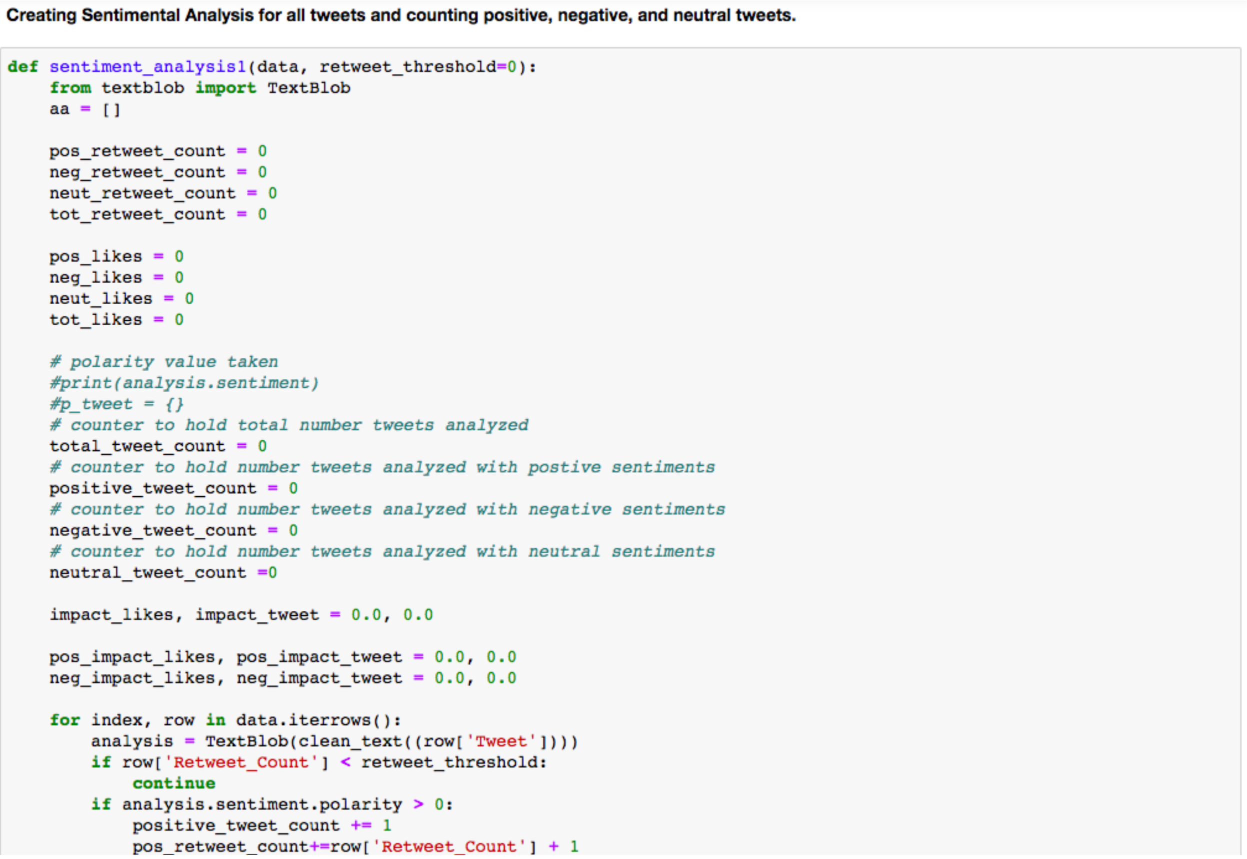
WordCloud formation for Boeing individual days

```
# # WORDCLOUD FOR DATASET NUMBER 1

# raw_tw1 = maketext(df1,'tweet_text_1')
# words1 = clean(raw_tw1)

# # WORDCLOUD FOR DATASET NUMBER 2

# raw_tw2 = maketext(df2,'tweet_text_2')
# words2 = clean(raw_tw2)


# # WORDCLOUD FOR DATASET NUMBER 3

# raw_tw3 = maketext(df3,'tweet_text_3')
# words3 = clean(raw_tw3)

# # WORDCLOUD FOR DATASET NUMBER 4

# raw_tw4 = maketext(df4,'tweet_text_1')
# words4 = clean(raw_tw4)

# #PLOT THE WORCLOUDS
# wordcl(words1,'plane2.png','13th March tweet wordcloud')
# wordcl(words2,'plane2.png','14th March tweet wordcloud')
# wordcl(words3,'plane2.png','15th March tweet wordcloud')
# wordcl(words4,'plane2.png','16th March tweet wordcloud')
```

Sentiment Analysis

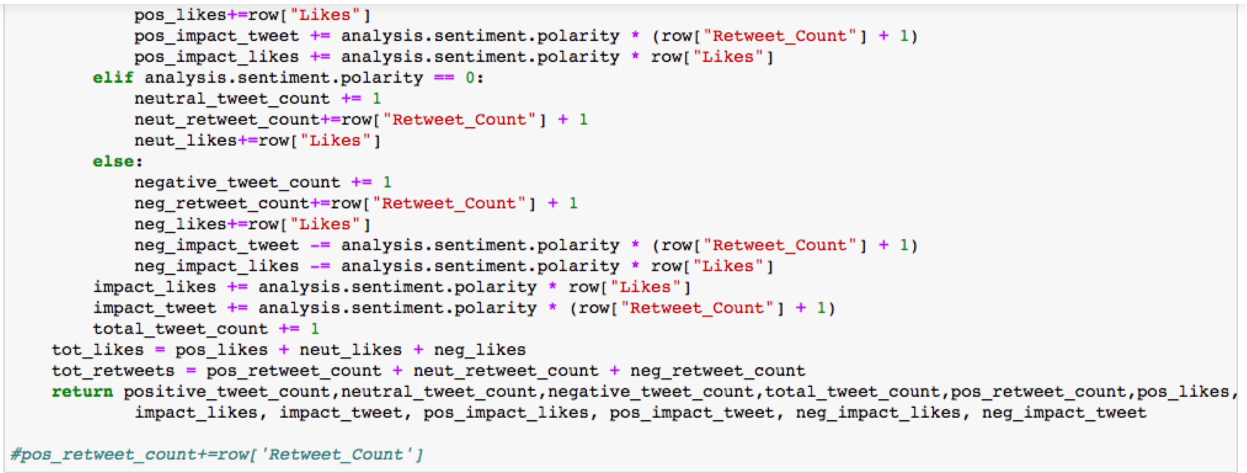
Cleaning tweets

```
# function retrive clean tweet text

def clean_text(x):
    raw_string = ''.join(x)
    no_links = re.sub(r'http\S+', '', raw_string)
    no_RT = re.sub(r'\[RT.+:\]','',no_links)
    no_unicode = re.sub(r"\\[a-z][a-z]?[0-9]+", '', no_RT)
    no_special_characters = re.sub('[^A-Za-z ]+', '', no_unicode)
    #print(no_special_characters)
    tweet_text = no_special_characters

    return tweet_text
```

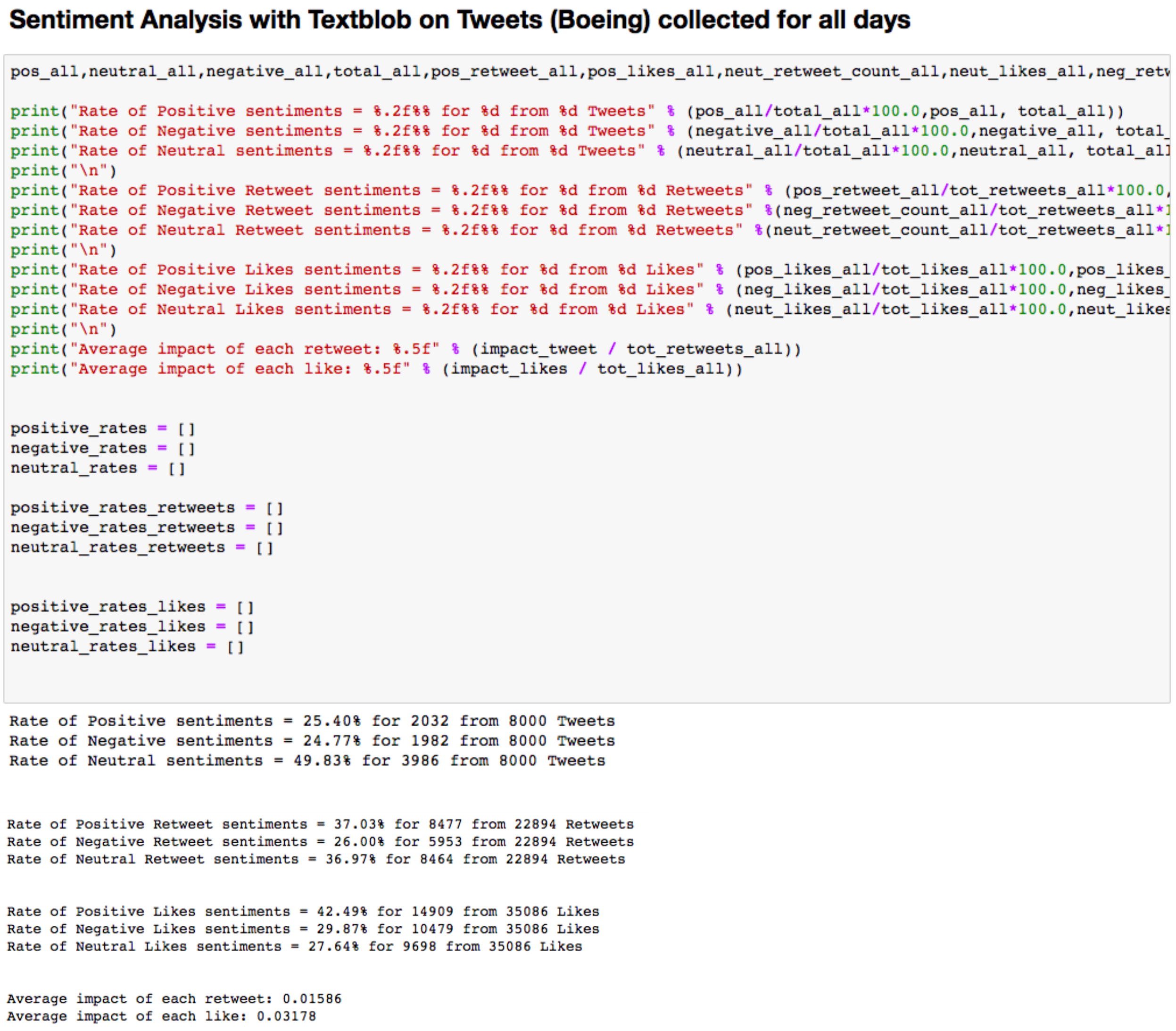


Creating Sentimental Analysis for all tweets and counting positive, negative, and neutral tweets.

```
def sentiment_analysis1(data, retweet_threshold=0):
    from textblob import TextBlob
    aa = []

    pos_retweet_count = 0
    neg_retweet_count = 0
    neut_retweet_count = 0
    tot_retweet_count = 0

    pos_likes = 0
    neg_likes = 0
    neut_likes = 0
    tot_likes = 0

    # polarity value taken
    #print(analysis.sentiment)
    #p_tweet = {}
    # counter to hold total number tweets analyzed
    total_tweet_count = 0
    # counter to hold number tweets analyzed with postive sentiments
    positive_tweet_count = 0
    # counter to hold number tweets analyzed with negative sentiments
    negative_tweet_count = 0
    # counter to hold number tweets analyzed with neutral sentiments
    neutral_tweet_count =0

    impact_likes, impact_tweet = 0.0, 0.0

    pos_impact_likes, pos_impact_tweet = 0.0, 0.0
    neg_impact_likes, neg_impact_tweet = 0.0, 0.0

    for index, row in data.iterrows():
        analysis = TextBlob(clean_text((row['Tweet'])))
        if row['Retweet_Count'] < retweet_threshold:
            continue
        if analysis.sentiment.polarity > 0:
            positive_tweet_count += 1
            pos_retweet_count+=row['Retweet_Count'] + 1
```



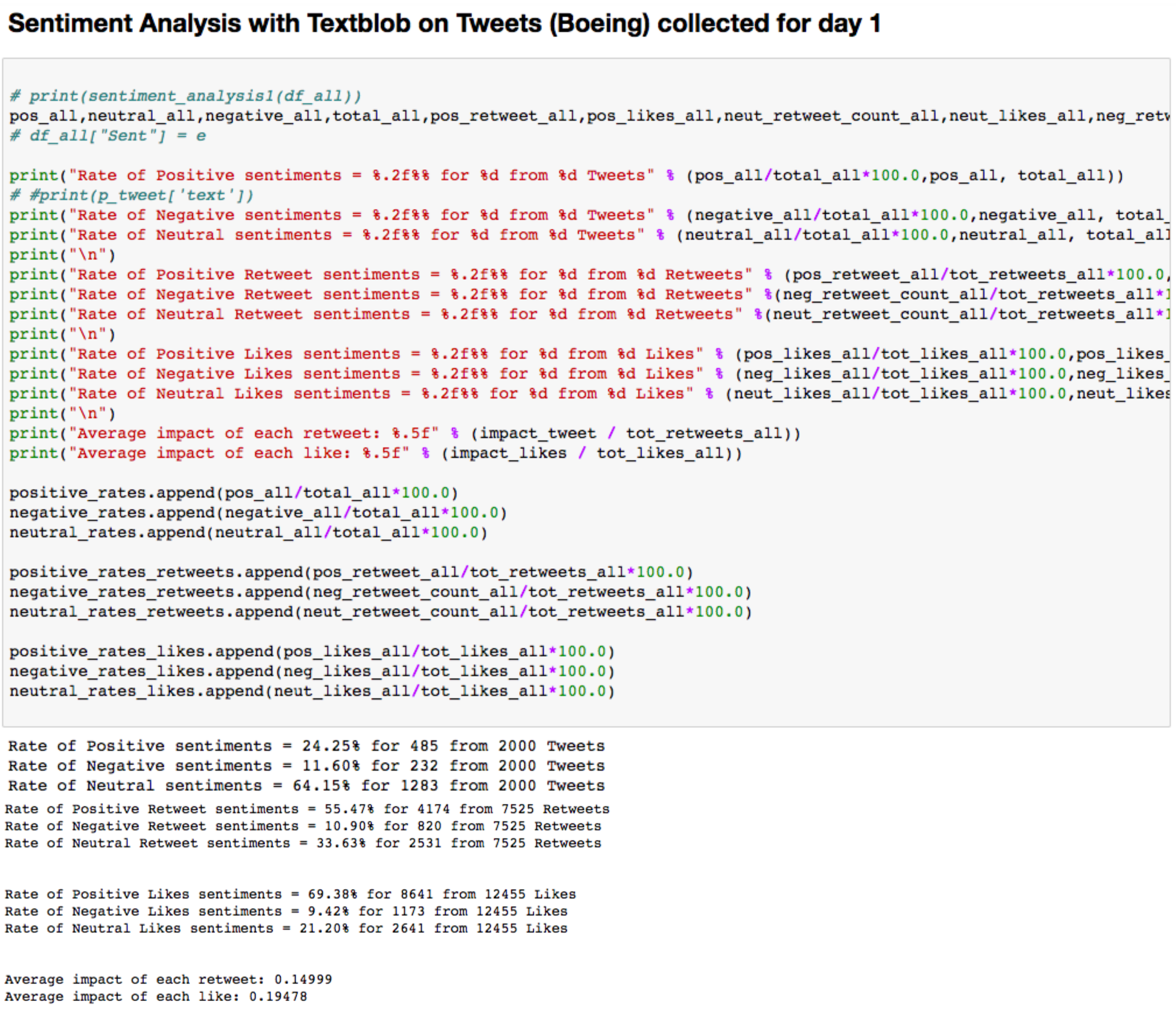
```
            pos_likes+=row["Likes"]
            pos_impact_tweet += analysis.sentiment.polarity * (row["Retweet_Count"] + 1)
            pos_impact_likes += analysis.sentiment.polarity * row["Likes"]
        elif analysis.sentiment.polarity == 0:
            neutral_tweet_count += 1
            neut_retweet_count+=row["Retweet_Count"] + 1
            neut_likes+=row["Likes"]
        else:
            negative_tweet_count += 1
            neg_retweet_count+=row["Retweet_Count"] + 1
            neg_likes+=row["Likes"]
            neg_impact_tweet -= analysis.sentiment.polarity * (row["Retweet_Count"] + 1)
            neg_impact_likes -= analysis.sentiment.polarity * row["Likes"]
        impact_likes += analysis.sentiment.polarity * row["Likes"]
        impact_tweet += analysis.sentiment.polarity * (row["Retweet_Count"] + 1)
        total_tweet_count += 1
    tot_likes = pos_likes + neut_likes + neg_likes
    tot_retweets = pos_retweet_count + neut_retweet_count + neg_retweet_count
    return positive_tweet_count,neutral_tweet_count,negative_tweet_count,total_tweet_count,pos_retweet_count,pos_likes,
            impact_likes, impact_tweet, pos_impact_likes, pos_impact_tweet, neg_impact_likes, neg_impact_tweet

#pos_retweet_count+=row['Retweet_Count']
```



Sentiment Analysis with Textblob on Tweets (Boeing) collected for all days

```
pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_retw

print("Rate of Positive sentiments = %.2f%% for %d from %d Tweets" % (pos_all/total_all*100.0,pos_all, total_all))
print("Rate of Negative sentiments = %.2f%% for %d from %d Tweets" % (negative_all/total_all*100.0,negative_all, total_
print("Rate of Neutral sentiments = %.2f%% for %d from %d Tweets" % (neutral_all/total_all*100.0,neutral_all, total_all
print("\n")
print("Rate of Positive Retweet sentiments = %.2f%% for %d from %d Retweets" % (pos_retweet_all/tot_retweets_all*100.0,
print("Rate of Negative Retweet sentiments = %.2f%% for %d from %d Retweets" %(neg_retweet_count_all/tot_retweets_all*1
print("Rate of Neutral Retweet sentiments = %.2f%% for %d from %d Retweets" %(neut_retweet_count_all/tot_retweets_all*1
print("\n")
print("Rate of Positive Likes sentiments = %.2f%% for %d from %d Likes" % (pos_likes_all/tot_likes_all*100.0,pos_likes_
print("Rate of Negative Likes sentiments = %.2f%% for %d from %d Likes" % (neg_likes_all/tot_likes_all*100.0,neg_likes_
print("Rate of Neutral Likes sentiments = %.2f%% for %d from %d Likes" % (neut_likes_all/tot_likes_all*100.0,neut_likes
print("\n")
print("Average impact of each retweet: %.5f" % (impact_tweet / tot_retweets_all))
print("Average impact of each like: %.5f" % (impact_likes / tot_likes_all))


positive_rates = []
negative_rates = []
neutral_rates = []

positive_rates_retweets = []
negative_rates_retweets = []
neutral_rates_retweets = []


positive_rates_likes = []
negative_rates_likes = []
neutral_rates_likes = []
```

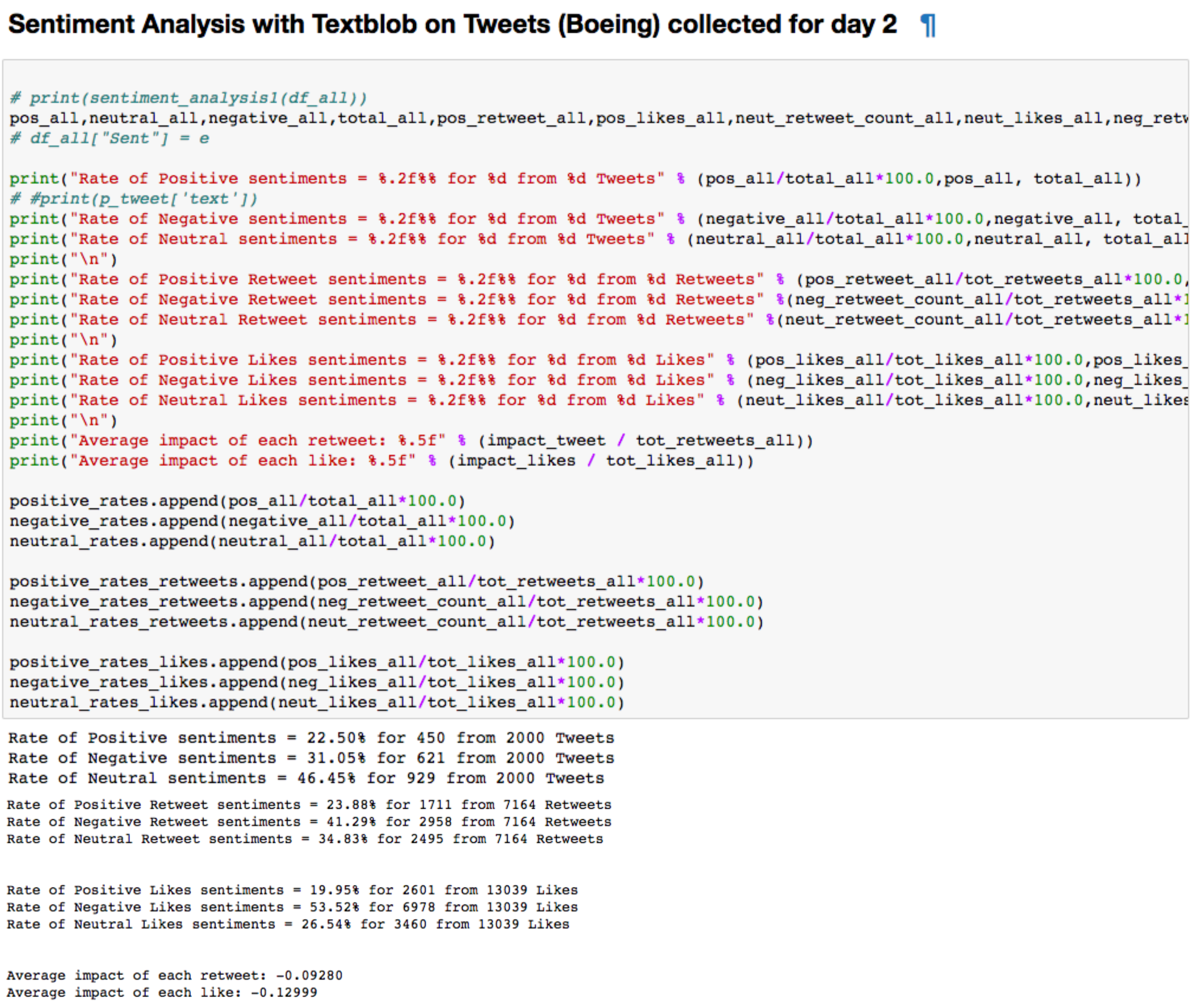
```
Rate of Positive sentiments = 25.40% for 2032 from 8000 Tweets
Rate of Negative sentiments = 24.77% for 1982 from 8000 Tweets
Rate of Neutral sentiments = 49.83% for 3986 from 8000 Tweets


Rate of Positive Retweet sentiments = 37.03% for 8477 from 22894 Retweets
Rate of Negative Retweet sentiments = 26.00% for 5953 from 22894 Retweets
Rate of Neutral Retweet sentiments = 36.97% for 8464 from 22894 Retweets


Rate of Positive Likes sentiments = 42.49% for 14909 from 35086 Likes
Rate of Negative Likes sentiments = 29.87% for 10479 from 35086 Likes
Rate of Neutral Likes sentiments = 27.64% for 9698 from 35086 Likes


Average impact of each retweet: 0.01586
Average impact of each like: 0.03178
```



Sentiment Analysis with Textblob on Tweets (Boeing) collected for day 1

```
# print(sentiment_analysis1(df_all))
pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_retw
# df_all["Sent"] = e

print("Rate of Positive sentiments = %.2f%% for %d from %d Tweets" % (pos_all/total_all*100.0,pos_all, total_all))
# #print(p_tweet['text'])
print("Rate of Negative sentiments = %.2f%% for %d from %d Tweets" % (negative_all/total_all*100.0,negative_all, total_
print("Rate of Neutral sentiments = %.2f%% for %d from %d Tweets" % (neutral_all/total_all*100.0,neutral_all, total_all
print("\n")
print("Rate of Positive Retweet sentiments = %.2f%% for %d from %d Retweets" % (pos_retweet_all/tot_retweets_all*100.0,
print("Rate of Negative Retweet sentiments = %.2f%% for %d from %d Retweets" %(neg_retweet_count_all/tot_retweets_all*1
print("Rate of Neutral Retweet sentiments = %.2f%% for %d from %d Retweets" %(neut_retweet_count_all/tot_retweets_all*1
print("\n")
print("Rate of Positive Likes sentiments = %.2f%% for %d from %d Likes" % (pos_likes_all/tot_likes_all*100.0,pos_likes_
print("Rate of Negative Likes sentiments = %.2f%% for %d from %d Likes" % (neg_likes_all/tot_likes_all*100.0,neg_likes_
print("Rate of Neutral Likes sentiments = %.2f%% for %d from %d Likes" % (neut_likes_all/tot_likes_all*100.0,neut_likes
print("\n")
print("Average impact of each retweet: %.5f" % (impact_tweet / tot_retweets_all))
print("Average impact of each like: %.5f" % (impact_likes / tot_likes_all))

positive_rates.append(pos_all/total_all*100.0)
negative_rates.append(negative_all/total_all*100.0)
neutral_rates.append(neutral_all/total_all*100.0)

positive_rates_retweets.append(pos_retweet_all/tot_retweets_all*100.0)
negative_rates_retweets.append(neg_retweet_count_all/tot_retweets_all*100.0)
neutral_rates_retweets.append(neut_retweet_count_all/tot_retweets_all*100.0)

positive_rates_likes.append(pos_likes_all/tot_likes_all*100.0)
negative_rates_likes.append(neg_likes_all/tot_likes_all*100.0)
neutral_rates_likes.append(neut_likes_all/tot_likes_all*100.0)
```

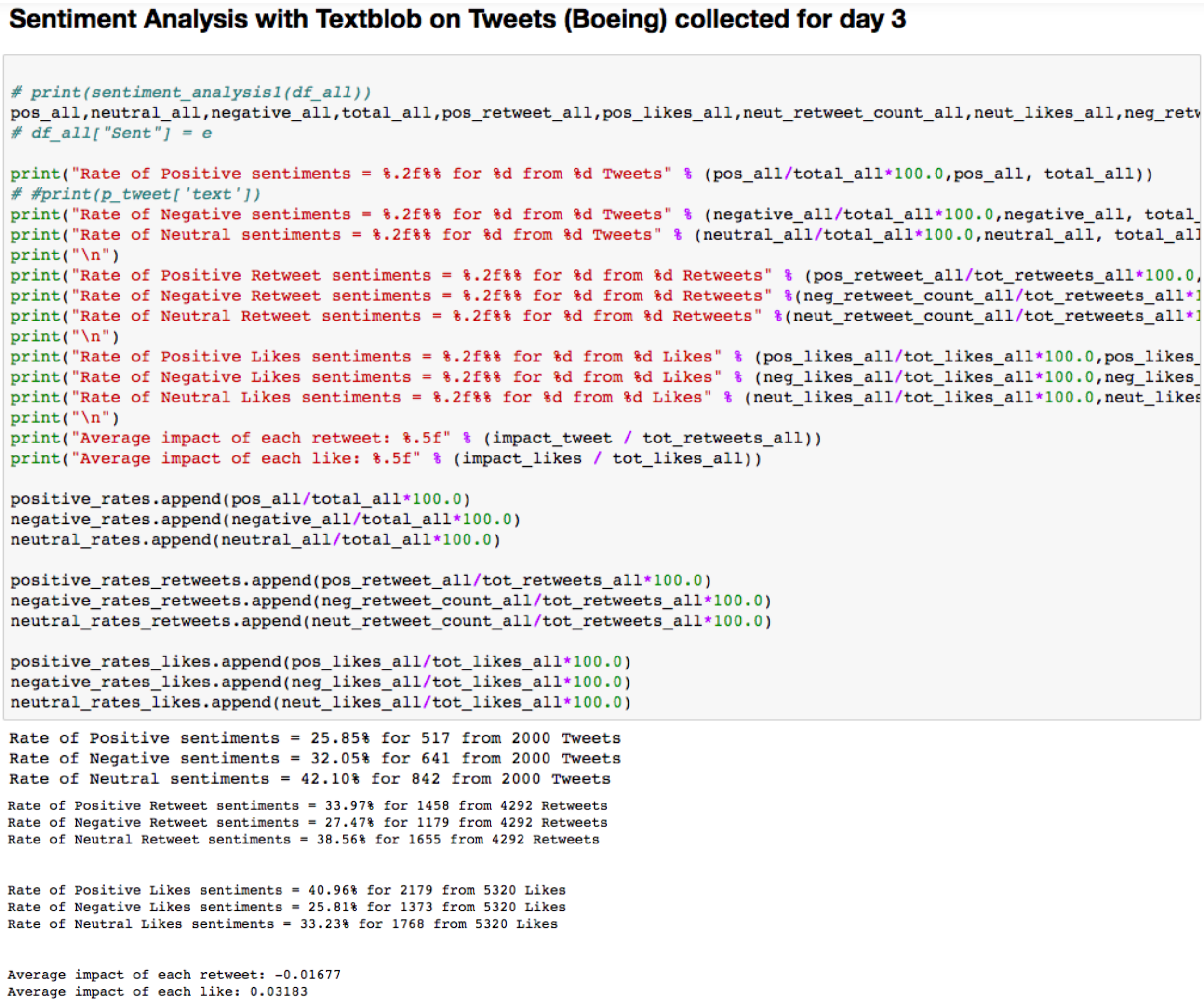
```
Rate of Positive sentiments = 24.25% for 485 from 2000 Tweets
Rate of Negative sentiments = 11.60% for 232 from 2000 Tweets
Rate of Neutral sentiments = 64.15% for 1283 from 2000 Tweets
Rate of Positive Retweet sentiments = 55.47% for 4174 from 7525 Retweets
Rate of Negative Retweet sentiments = 10.90% for 820 from 7525 Retweets
Rate of Neutral Retweet sentiments = 33.63% for 2531 from 7525 Retweets


Rate of Positive Likes sentiments = 69.38% for 8641 from 12455 Likes
Rate of Negative Likes sentiments = 9.42% for 1173 from 12455 Likes
Rate of Neutral Likes sentiments = 21.20% for 2641 from 12455 Likes


Average impact of each retweet: 0.14999
Average impact of each like: 0.19478
```



Sentiment Analysis with Textblob on Tweets (Boeing) collected for day 2 ¶

```
# print(sentiment_analysis1(df_all))
pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_retw
# df_all["Sent"] = e

print("Rate of Positive sentiments = %.2f%% for %d from %d Tweets" % (pos_all/total_all*100.0,pos_all, total_all))
# #print(p_tweet['text'])
print("Rate of Negative sentiments = %.2f%% for %d from %d Tweets" % (negative_all/total_all*100.0,negative_all, total_
print("Rate of Neutral sentiments = %.2f%% for %d from %d Tweets" % (neutral_all/total_all*100.0,neutral_all, total_all
print("\n")
print("Rate of Positive Retweet sentiments = %.2f%% for %d from %d Retweets" % (pos_retweet_all/tot_retweets_all*100.0,
print("Rate of Negative Retweet sentiments = %.2f%% for %d from %d Retweets" %(neg_retweet_count_all/tot_retweets_all*1
print("Rate of Neutral Retweet sentiments = %.2f%% for %d from %d Retweets" %(neut_retweet_count_all/tot_retweets_all*1
print("\n")
print("Rate of Positive Likes sentiments = %.2f%% for %d from %d Likes" % (pos_likes_all/tot_likes_all*100.0,pos_likes_
print("Rate of Negative Likes sentiments = %.2f%% for %d from %d Likes" % (neg_likes_all/tot_likes_all*100.0,neg_likes_
print("Rate of Neutral Likes sentiments = %.2f%% for %d from %d Likes" % (neut_likes_all/tot_likes_all*100.0,neut_likes
print("\n")
print("Average impact of each retweet: %.5f" % (impact_tweet / tot_retweets_all))
print("Average impact of each like: %.5f" % (impact_likes / tot_likes_all))

positive_rates.append(pos_all/total_all*100.0)
negative_rates.append(negative_all/total_all*100.0)
neutral_rates.append(neutral_all/total_all*100.0)

positive_rates_retweets.append(pos_retweet_all/tot_retweets_all*100.0)
negative_rates_retweets.append(neg_retweet_count_all/tot_retweets_all*100.0)
neutral_rates_retweets.append(neut_retweet_count_all/tot_retweets_all*100.0)

positive_rates_likes.append(pos_likes_all/tot_likes_all*100.0)
negative_rates_likes.append(neg_likes_all/tot_likes_all*100.0)
neutral_rates_likes.append(neut_likes_all/tot_likes_all*100.0)
```

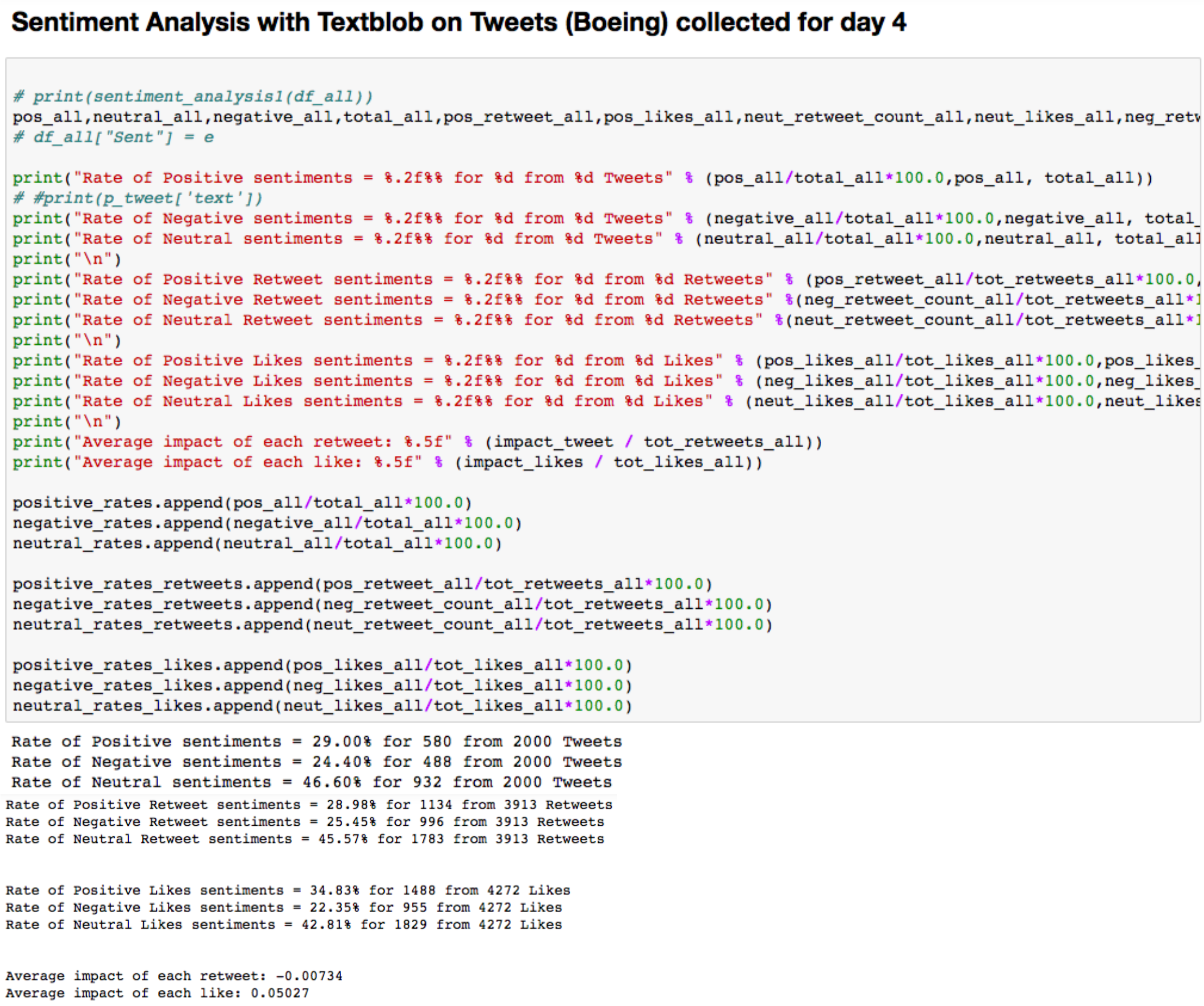
```
Rate of Positive sentiments = 22.50% for 450 from 2000 Tweets
Rate of Negative sentiments = 31.05% for 621 from 2000 Tweets
Rate of Neutral sentiments = 46.45% for 929 from 2000 Tweets

Rate of Positive Retweet sentiments = 23.88% for 1711 from 7164 Retweets
Rate of Negative Retweet sentiments = 41.29% for 2958 from 7164 Retweets
Rate of Neutral Retweet sentiments = 34.83% for 2495 from 7164 Retweets


Rate of Positive Likes sentiments = 19.95% for 2601 from 13039 Likes
Rate of Negative Likes sentiments = 53.52% for 6978 from 13039 Likes
Rate of Neutral Likes sentiments = 26.54% for 3460 from 13039 Likes


Average impact of each retweet: -0.09280
Average impact of each like: -0.12999
```



Sentiment Analysis with Textblob on Tweets (Boeing) collected for day 3

```
# print(sentiment_analysis1(df_all))
pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_retw
# df_all["Sent"] = e

print("Rate of Positive sentiments = %.2f%% for %d from %d Tweets" % (pos_all/total_all*100.0,pos_all, total_all))
# #print(p_tweet['text'])
print("Rate of Negative sentiments = %.2f%% for %d from %d Tweets" % (negative_all/total_all*100.0,negative_all, total_
print("Rate of Neutral sentiments = %.2f%% for %d from %d Tweets" % (neutral_all/total_all*100.0,neutral_all, total_all
print("\n")
print("Rate of Positive Retweet sentiments = %.2f%% for %d from %d Retweets" % (pos_retweet_all/tot_retweets_all*100.0,
print("Rate of Negative Retweet sentiments = %.2f%% for %d from %d Retweets" %(neg_retweet_count_all/tot_retweets_all*1
print("Rate of Neutral Retweet sentiments = %.2f%% for %d from %d Retweets" %(neut_retweet_count_all/tot_retweets_all*1
print("\n")
print("Rate of Positive Likes sentiments = %.2f%% for %d from %d Likes" % (pos_likes_all/tot_likes_all*100.0,pos_likes_
print("Rate of Negative Likes sentiments = %.2f%% for %d from %d Likes" % (neg_likes_all/tot_likes_all*100.0,neg_likes_
print("Rate of Neutral Likes sentiments = %.2f%% for %d from %d Likes" % (neut_likes_all/tot_likes_all*100.0,neut_likes
print("\n")
print("Average impact of each retweet: %.5f" % (impact_tweet / tot_retweets_all))
print("Average impact of each like: %.5f" % (impact_likes / tot_likes_all))

positive_rates.append(pos_all/total_all*100.0)
negative_rates.append(negative_all/total_all*100.0)
neutral_rates.append(neutral_all/total_all*100.0)

positive_rates_retweets.append(pos_retweet_all/tot_retweets_all*100.0)
negative_rates_retweets.append(neg_retweet_count_all/tot_retweets_all*100.0)
neutral_rates_retweets.append(neut_retweet_count_all/tot_retweets_all*100.0)

positive_rates_likes.append(pos_likes_all/tot_likes_all*100.0)
negative_rates_likes.append(neg_likes_all/tot_likes_all*100.0)
neutral_rates_likes.append(neut_likes_all/tot_likes_all*100.0)
```

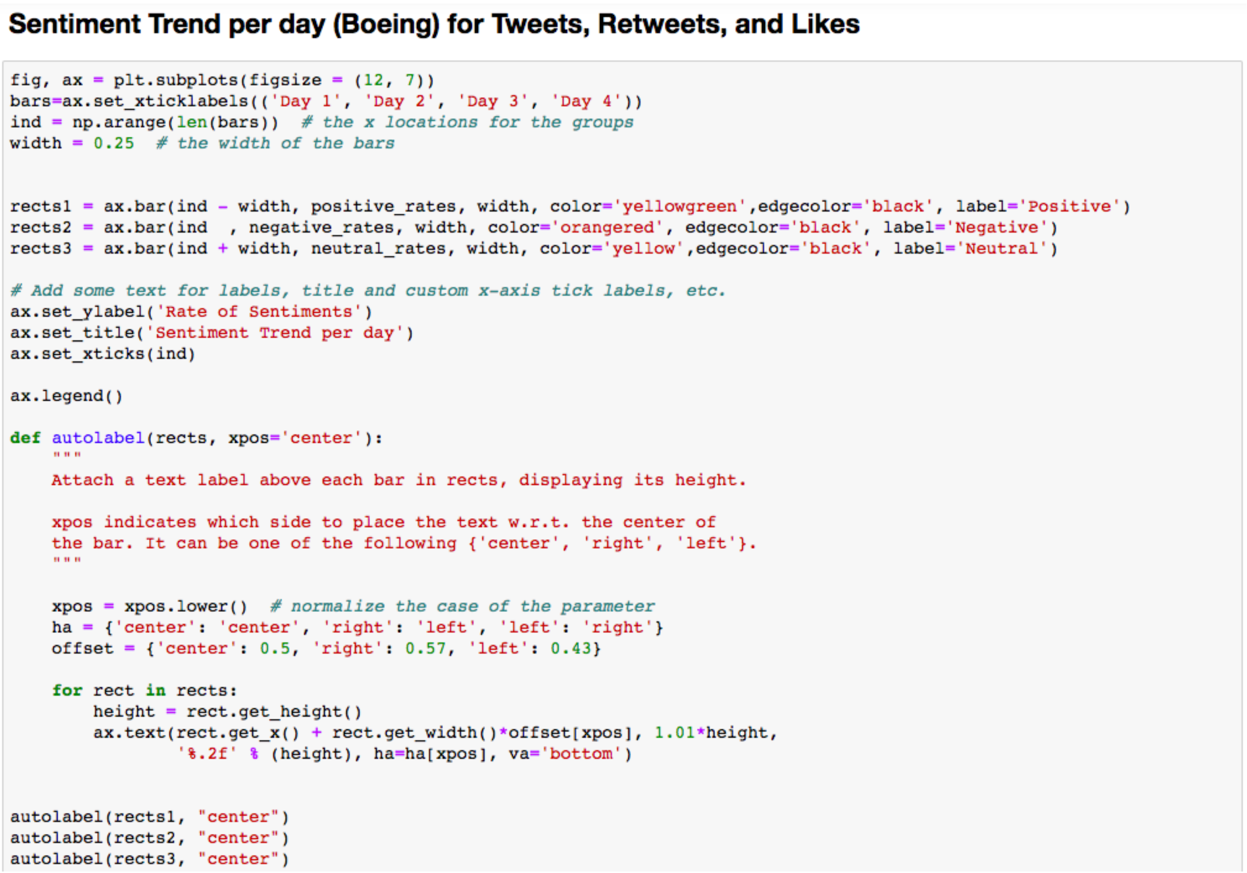
```
Rate of Positive sentiments = 25.85% for 517 from 2000 Tweets
Rate of Negative sentiments = 32.05% for 641 from 2000 Tweets
Rate of Neutral sentiments = 42.10% for 842 from 2000 Tweets

Rate of Positive Retweet sentiments = 33.97% for 1458 from 4292 Retweets
Rate of Negative Retweet sentiments = 27.47% for 1179 from 4292 Retweets
Rate of Neutral Retweet sentiments = 38.56% for 1655 from 4292 Retweets


Rate of Positive Likes sentiments = 40.96% for 2179 from 5320 Likes
Rate of Negative Likes sentiments = 25.81% for 1373 from 5320 Likes
Rate of Neutral Likes sentiments = 33.23% for 1768 from 5320 Likes


Average impact of each retweet: -0.01677
Average impact of each like: 0.03183
```



Sentiment Analysis with Textblob on Tweets (Boeing) collected for day 4

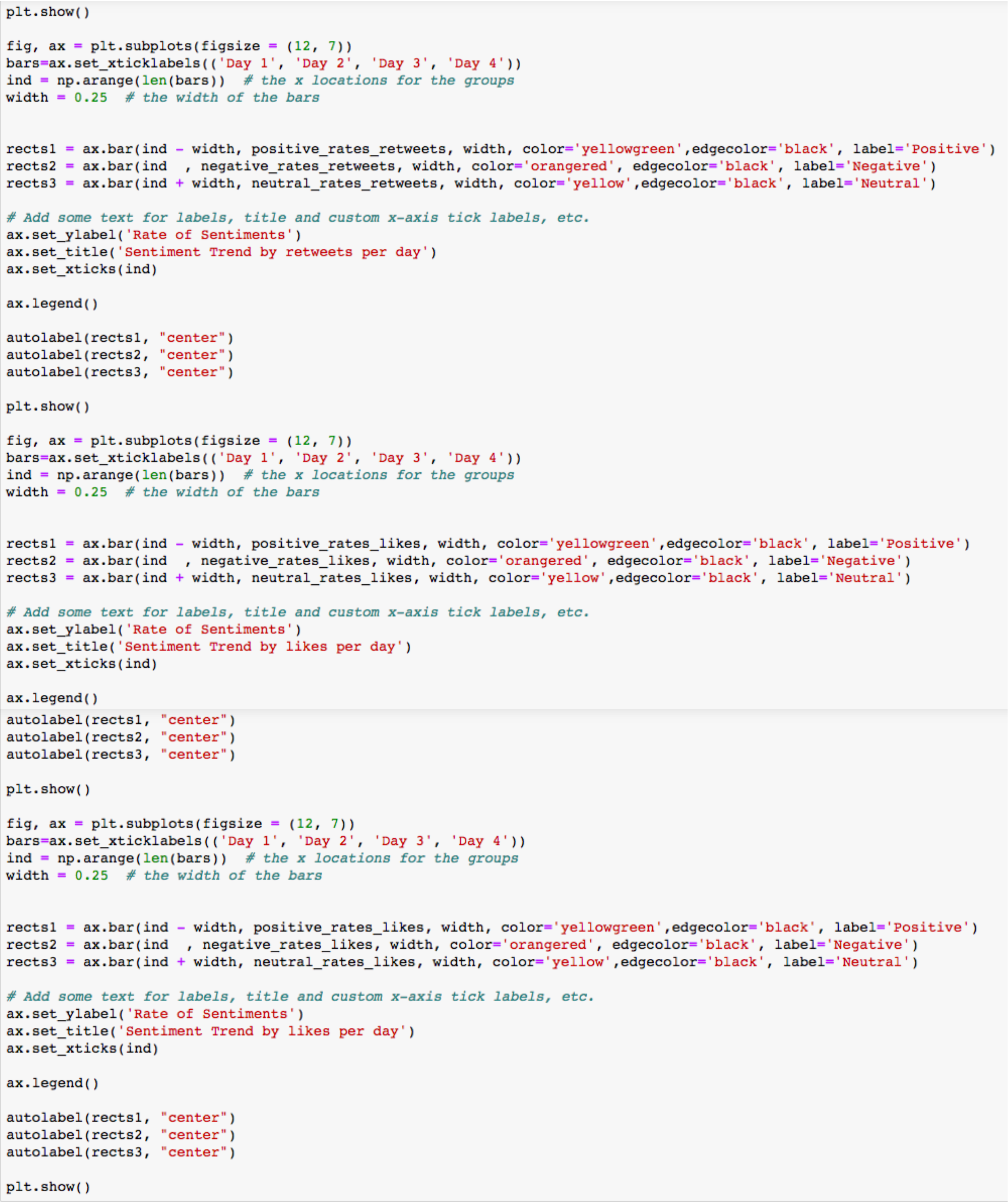
```
# print(sentiment_analysis1(df_all))
pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_retw
# df_all["Sent"] = e

print("Rate of Positive sentiments = %.2f%% for %d from %d Tweets" % (pos_all/total_all*100.0,pos_all, total_all))
# #print(p_tweet['text'])
print("Rate of Negative sentiments = %.2f%% for %d from %d Tweets" % (negative_all/total_all*100.0,negative_all, total_
print("Rate of Neutral sentiments = %.2f%% for %d from %d Tweets" % (neutral_all/total_all*100.0,neutral_all, total_all
print("\n")
print("Rate of Positive Retweet sentiments = %.2f%% for %d from %d Retweets" % (pos_retweet_all/tot_retweets_all*100.0,
print("Rate of Negative Retweet sentiments = %.2f%% for %d from %d Retweets" %(neg_retweet_count_all/tot_retweets_all*1
print("Rate of Neutral Retweet sentiments = %.2f%% for %d from %d Retweets" %(neut_retweet_count_all/tot_retweets_all*1
print("\n")
print("Rate of Positive Likes sentiments = %.2f%% for %d from %d Likes" % (pos_likes_all/tot_likes_all*100.0,pos_likes_
print("Rate of Negative Likes sentiments = %.2f%% for %d from %d Likes" % (neg_likes_all/tot_likes_all*100.0,neg_likes_
print("Rate of Neutral Likes sentiments = %.2f%% for %d from %d Likes" % (neut_likes_all/tot_likes_all*100.0,neut_likes
print("\n")
print("Average impact of each retweet: %.5f" % (impact_tweet / tot_retweets_all))
print("Average impact of each like: %.5f" % (impact_likes / tot_likes_all))

positive_rates.append(pos_all/total_all*100.0)
negative_rates.append(negative_all/total_all*100.0)
neutral_rates.append(neutral_all/total_all*100.0)

positive_rates_retweets.append(pos_retweet_all/tot_retweets_all*100.0)
negative_rates_retweets.append(neg_retweet_count_all/tot_retweets_all*100.0)
neutral_rates_retweets.append(neut_retweet_count_all/tot_retweets_all*100.0)

positive_rates_likes.append(pos_likes_all/tot_likes_all*100.0)
negative_rates_likes.append(neg_likes_all/tot_likes_all*100.0)
neutral_rates_likes.append(neut_likes_all/tot_likes_all*100.0)
```

```
Rate of Positive sentiments = 29.00% for 580 from 2000 Tweets
Rate of Negative sentiments = 24.40% for 488 from 2000 Tweets
Rate of Neutral sentiments = 46.60% for 932 from 2000 Tweets
Rate of Positive Retweet sentiments = 28.98% for 1134 from 3913 Retweets
Rate of Negative Retweet sentiments = 25.45% for 996 from 3913 Retweets
Rate of Neutral Retweet sentiments = 45.57% for 1783 from 3913 Retweets


Rate of Positive Likes sentiments = 34.83% for 1488 from 4272 Likes
Rate of Negative Likes sentiments = 22.35% for 955 from 4272 Likes
Rate of Neutral Likes sentiments = 42.81% for 1829 from 4272 Likes


Average impact of each retweet: -0.00734
Average impact of each like: 0.05027
```

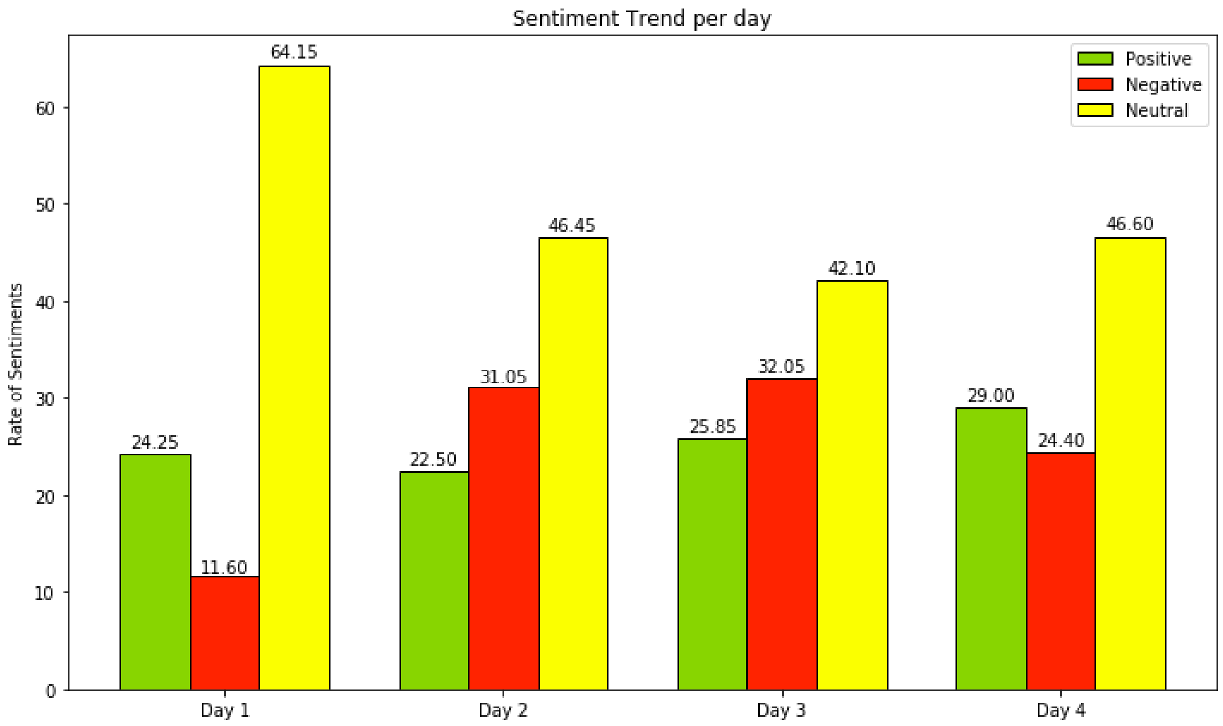


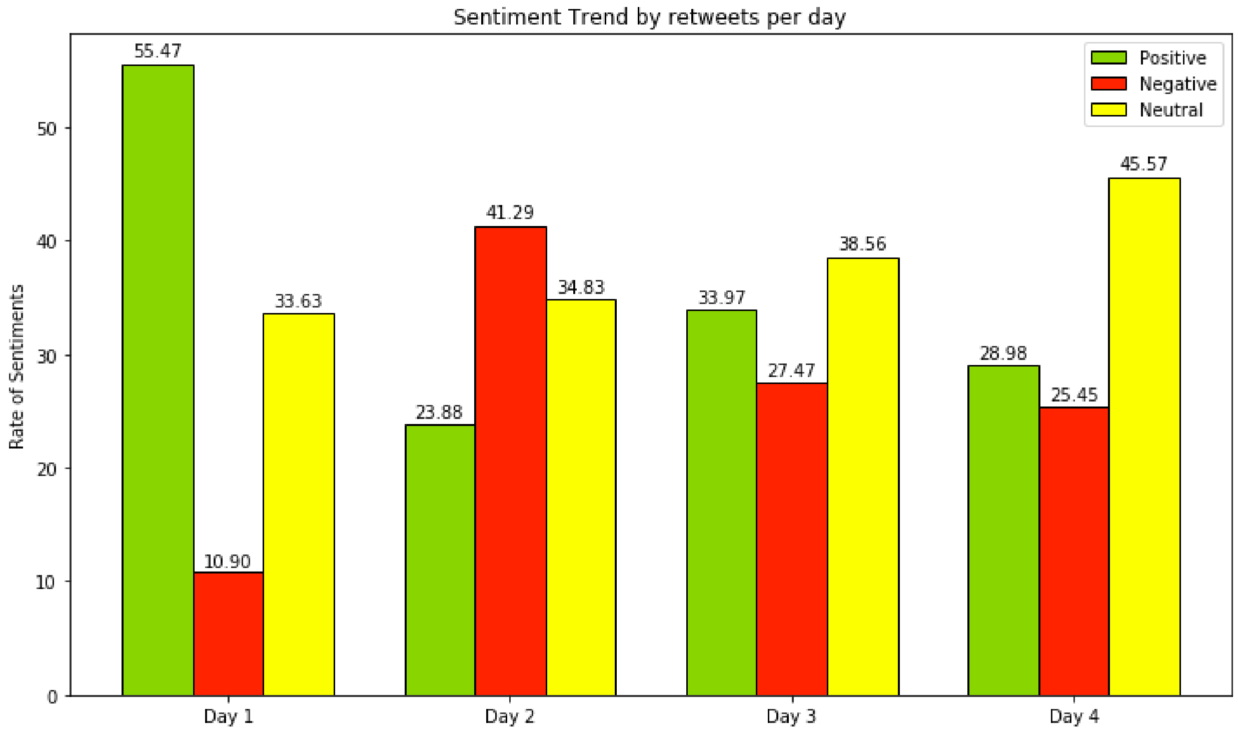
Sentiment Trend per day (Boeing) for Tweets, Retweets, and Likes

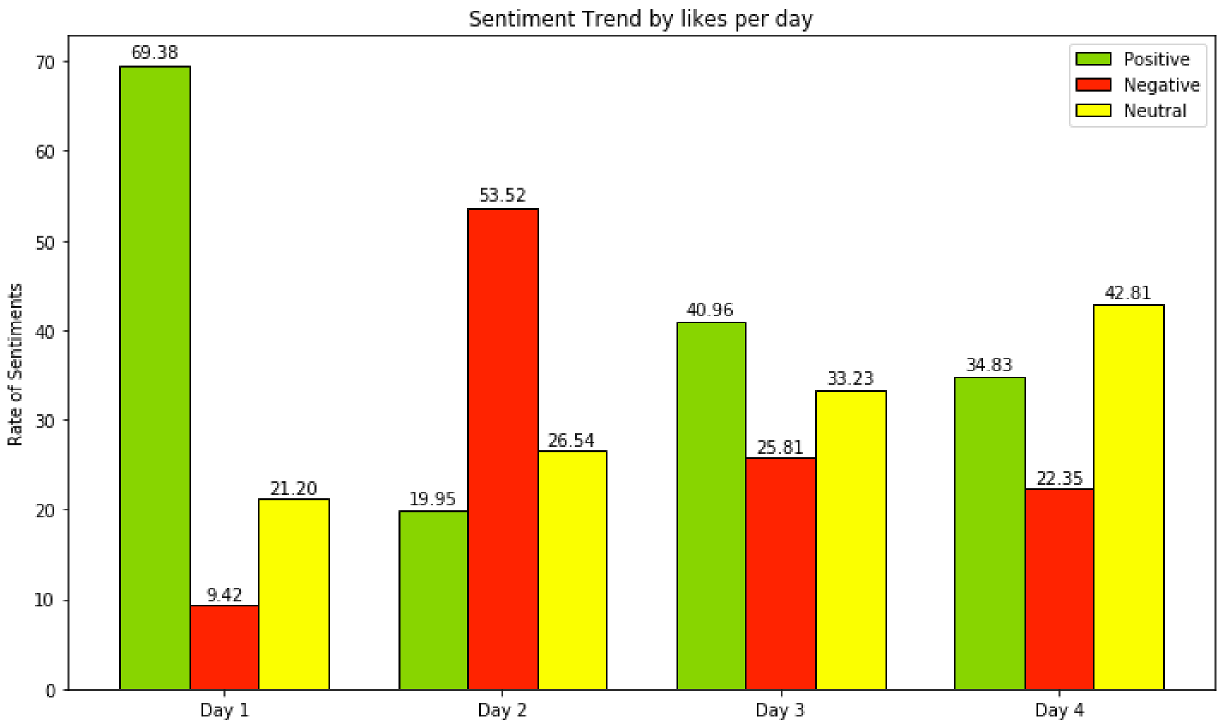
```
fig, ax = plt.subplots(figsize = (12, 7))
bars=ax.set_xticklabels(('Day 1', 'Day 2', 'Day 3', 'Day 4'))
ind = np.arange(len(bars))  # the x locations for the groups
width = 0.25  # the width of the bars


rects1 = ax.bar(ind - width, positive_rates, width, color='yellowgreen',edgecolor='black', label='Positive')
rects2 = ax.bar(ind  , negative_rates, width, color='orangered', edgecolor='black', label='Negative')
rects3 = ax.bar(ind + width, neutral_rates, width, color='yellow',edgecolor='black', label='Neutral')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Rate of Sentiments')
ax.set_title('Sentiment Trend per day')
ax.set_xticks(ind)

ax.legend()

def autolabel(rects, xpos='center'):
    """
    Attach a text label above each bar in rects, displaying its height.

    xpos indicates which side to place the text w.r.t. the center of
    the bar. It can be one of the following {'center', 'right', 'left'}.
    """

    xpos = xpos.lower()  # normalize the case of the parameter
    ha = {'center': 'center', 'right': 'left', 'left': 'right'}
    offset = {'center': 0.5, 'right': 0.57, 'left': 0.43}

    for rect in rects:
        height = rect.get_height()
        ax.text(rect.get_x() + rect.get_width()*offset[xpos], 1.01*height,
                '%.2f' % (height), ha=ha[xpos], va='bottom')


autolabel(rects1, "center")
autolabel(rects2, "center")
autolabel(rects3, "center")
```



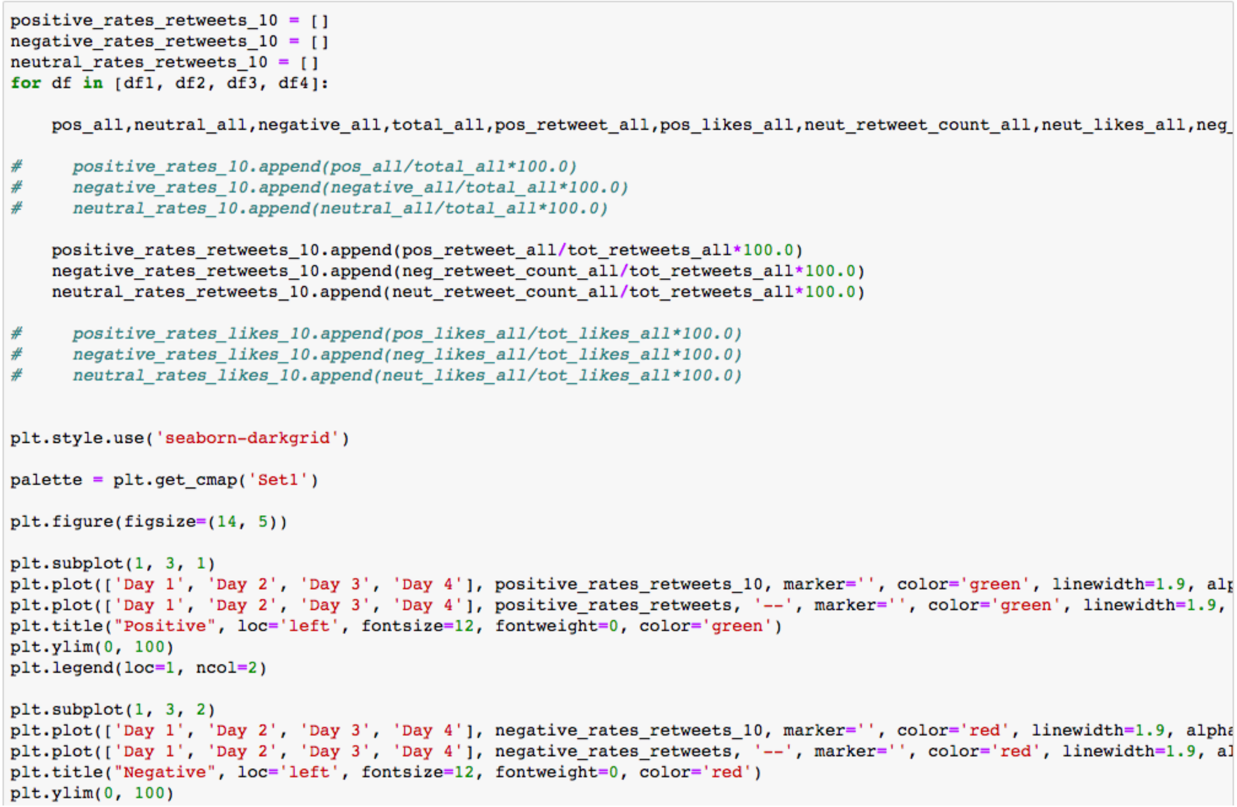
```
plt.show()

fig, ax = plt.subplots(figsize = (12, 7))
bars=ax.set_xticklabels(('Day 1', 'Day 2', 'Day 3', 'Day 4'))
ind = np.arange(len(bars))  # the x locations for the groups
width = 0.25  # the width of the bars


rects1 = ax.bar(ind - width, positive_rates_retweets, width, color='yellowgreen',edgecolor='black', label='Positive')
rects2 = ax.bar(ind  , negative_rates_retweets, width, color='orangered', edgecolor='black', label='Negative')
rects3 = ax.bar(ind + width, neutral_rates_retweets, width, color='yellow',edgecolor='black', label='Neutral')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Rate of Sentiments')
ax.set_title('Sentiment Trend by retweets per day')
ax.set_xticks(ind)

ax.legend()

autolabel(rects1, "center")
autolabel(rects2, "center")
autolabel(rects3, "center")

plt.show()

fig, ax = plt.subplots(figsize = (12, 7))
bars=ax.set_xticklabels(('Day 1', 'Day 2', 'Day 3', 'Day 4'))
ind = np.arange(len(bars))  # the x locations for the groups
width = 0.25  # the width of the bars


rects1 = ax.bar(ind - width, positive_rates_likes, width, color='yellowgreen',edgecolor='black', label='Positive')
rects2 = ax.bar(ind  , negative_rates_likes, width, color='orangered', edgecolor='black', label='Negative')
rects3 = ax.bar(ind + width, neutral_rates_likes, width, color='yellow',edgecolor='black', label='Neutral')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Rate of Sentiments')
ax.set_title('Sentiment Trend by likes per day')
ax.set_xticks(ind)

ax.legend()
autolabel(rects1, "center")
autolabel(rects2, "center")
autolabel(rects3, "center")

plt.show()

fig, ax = plt.subplots(figsize = (12, 7))
bars=ax.set_xticklabels(('Day 1', 'Day 2', 'Day 3', 'Day 4'))
ind = np.arange(len(bars))  # the x locations for the groups
width = 0.25  # the width of the bars


rects1 = ax.bar(ind - width, positive_rates_likes, width, color='yellowgreen',edgecolor='black', label='Positive')
rects2 = ax.bar(ind  , negative_rates_likes, width, color='orangered', edgecolor='black', label='Negative')
rects3 = ax.bar(ind + width, neutral_rates_likes, width, color='yellow',edgecolor='black', label='Neutral')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Rate of Sentiments')
ax.set_title('Sentiment Trend by likes per day')
ax.set_xticks(ind)

ax.legend()

autolabel(rects1, "center")
autolabel(rects2, "center")
autolabel(rects3, "center")

plt.show()
```


Sentiment Trend per day
Positive
Negative
Neutral
Rate of Sentiments
0
10
20
30
40
50
60
24.25
11.60
64.15
22.50
31.05
46.45
25.85
32.05
42.10
29.00
24.40
46.60
Day 1
Day 2
Day 3
Day 4

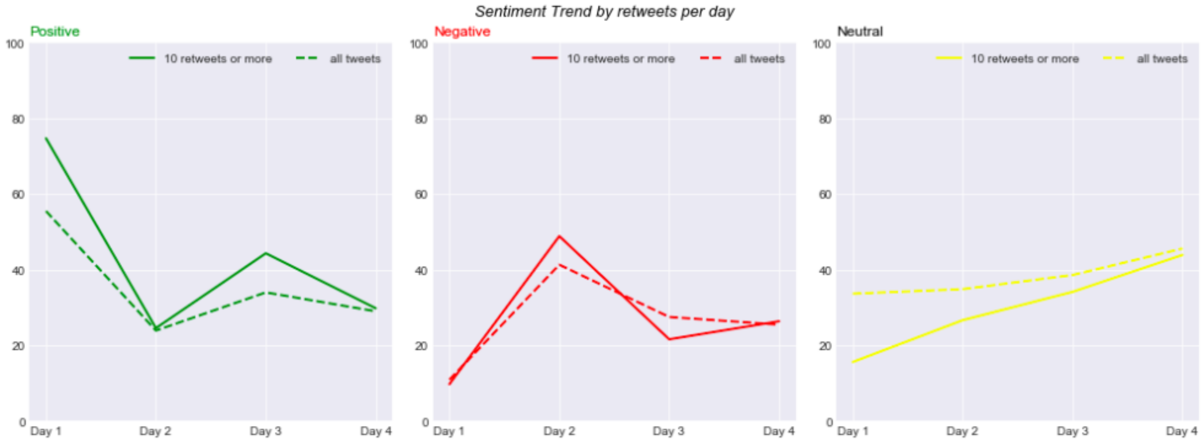

Sentiment Trend by retweets per day
Positive
Negative
Neutral
Rate of Sentiments
0
10
20
30
40
50
55.47
10.90
33.63
23.88
41.29
34.83
33.97
27.47
38.56
28.98
25.45
45.57
Day 1
Day 2
Day 3
Day 4



Line Chart for Sentimental Analysis for Boeing for 4 days



```
positive_rates_retweets_10 = []
negative_rates_retweets_10 = []
neutral_rates_retweets_10 = []
for df in [df1, df2, df3, df4]:

    pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_

#     positive_rates_10.append(pos_all/total_all*100.0)
#     negative_rates_10.append(negative_all/total_all*100.0)
#     neutral_rates_10.append(neutral_all/total_all*100.0)

    positive_rates_retweets_10.append(pos_retweet_all/tot_retweets_all*100.0)
    negative_rates_retweets_10.append(neg_retweet_count_all/tot_retweets_all*100.0)
    neutral_rates_retweets_10.append(neut_retweet_count_all/tot_retweets_all*100.0)

#     positive_rates_likes_10.append(pos_likes_all/tot_likes_all*100.0)
#     negative_rates_likes_10.append(neg_likes_all/tot_likes_all*100.0)
#     neutral_rates_likes_10.append(neut_likes_all/tot_likes_all*100.0)


plt.style.use('seaborn-darkgrid')

palette = plt.get_cmap('Set1')

plt.figure(figsize=(14, 5))

plt.subplot(1, 3, 1)
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], positive_rates_retweets_10, marker='', color='green', linewidth=1.9, alp
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], positive_rates_retweets, '--', marker='', color='green', linewidth=1.9,
plt.title("Positive", loc='left', fontsize=12, fontweight=0, color='green')
plt.ylim(0, 100)
plt.legend(loc=1, ncol=2)

plt.subplot(1, 3, 2)
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], negative_rates_retweets_10, marker='', color='red', linewidth=1.9, alpha
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], negative_rates_retweets, '--', marker='', color='red', linewidth=1.9, al
plt.title("Negative", loc='left', fontsize=12, fontweight=0, color='red')
plt.ylim(0, 100)
plt.ylim(0, 100)
plt.legend(loc=1, ncol=2)

plt.subplot(1, 3, 3)
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], neutral_rates_retweets_10, marker='', color='yellow', linewidth=1.9, alp
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], neutral_rates_retweets, '--', marker='', color='yellow', linewidth=1.9,
plt.title("Neutral", loc='left', fontsize=12, fontweight=0, color='black')
plt.ylim(0, 100)
plt.legend(loc=1, ncol=2)

# general title
plt.suptitle("Sentiment Trend by retweets per day", fontsize=13, fontweight=0, color='black', style='italic', y=1.02)
plt.tight_layout()
```



NEW ZEALAND Analysis

Grouping Tweet DataFrame

> Concatenating all 4 csv files into one dataframe.

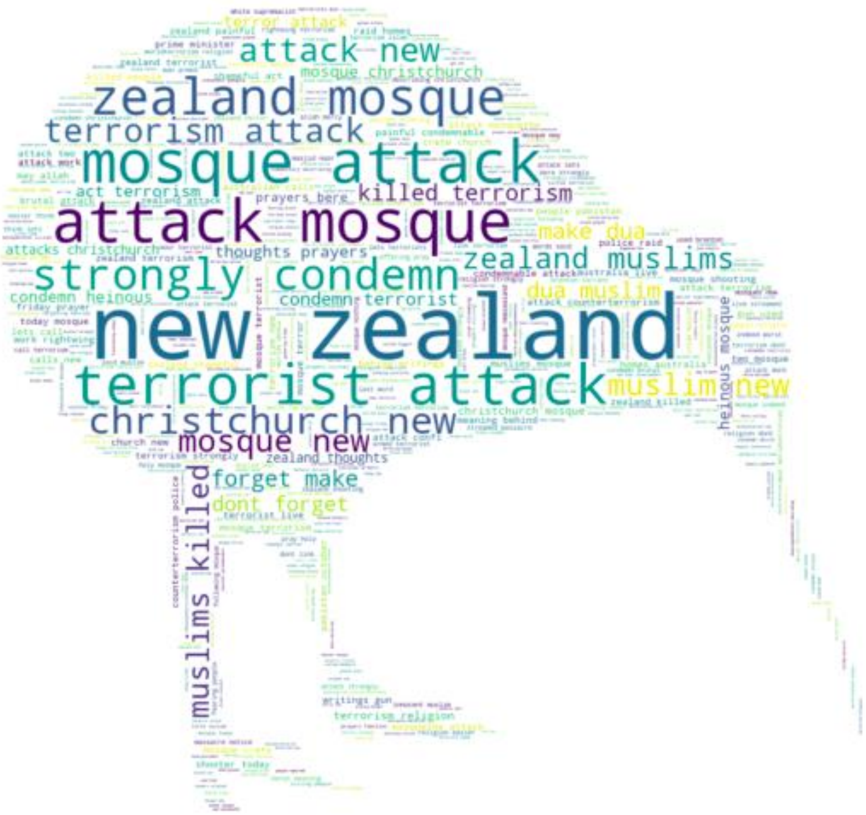
```
filenames = ['NewZealand_3_15_19.csv','NewZealand_3_16_19.csv','NewZealand_3_17_19.csv','NewZealand_3_18_19.csv']

df_all = pd.concat( [ pd.read_csv(f) for f in filenames ] )
df_all.columns = ['Id' if x=='Unnamed: 0' else 'Retweet_Count' if x=='Retweet Count' else 'Created_At' if x=='Created A
df_all.head()
```

	Id	Tweet	Likes	Username	Retweet_Count	Created_At
0	0	Thoughts out to the families and friends of al...	1	Thunda Banton	0	2019-03-15 23:48:55
1	1	PM Imran Khan condemns New Zealand attack, say...	1	Lombok Hebat	0	2019-03-15 23:42:45
2	2	Don't forget, make dua for muslim in New Zeala...	4	Hudayfi Abdirashid Abdirahman	0	2019-03-15 23:38:06
3	3	Extremist appears in court charged with New Ze...	0	Pir Ahmad Shah Qureshi 🇵🇰	0	2019-03-15 23:21:25
4	4	COMMENTARY: When describing the Christchurch m...	20	Kamil Karamali	6	2019-03-15 23:19:48

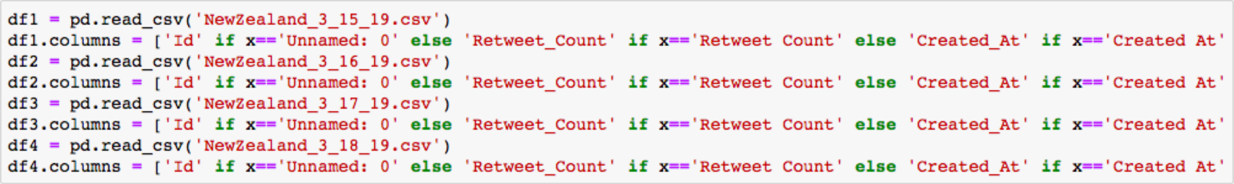
WordCloud Formation for the whole New Zealand dataset

```
raw_tw = maketext(df_all,'tweet_text_all3')
words = clean(raw_tw)
wordcl(words,'kiwi.jpg','All tweets')
```

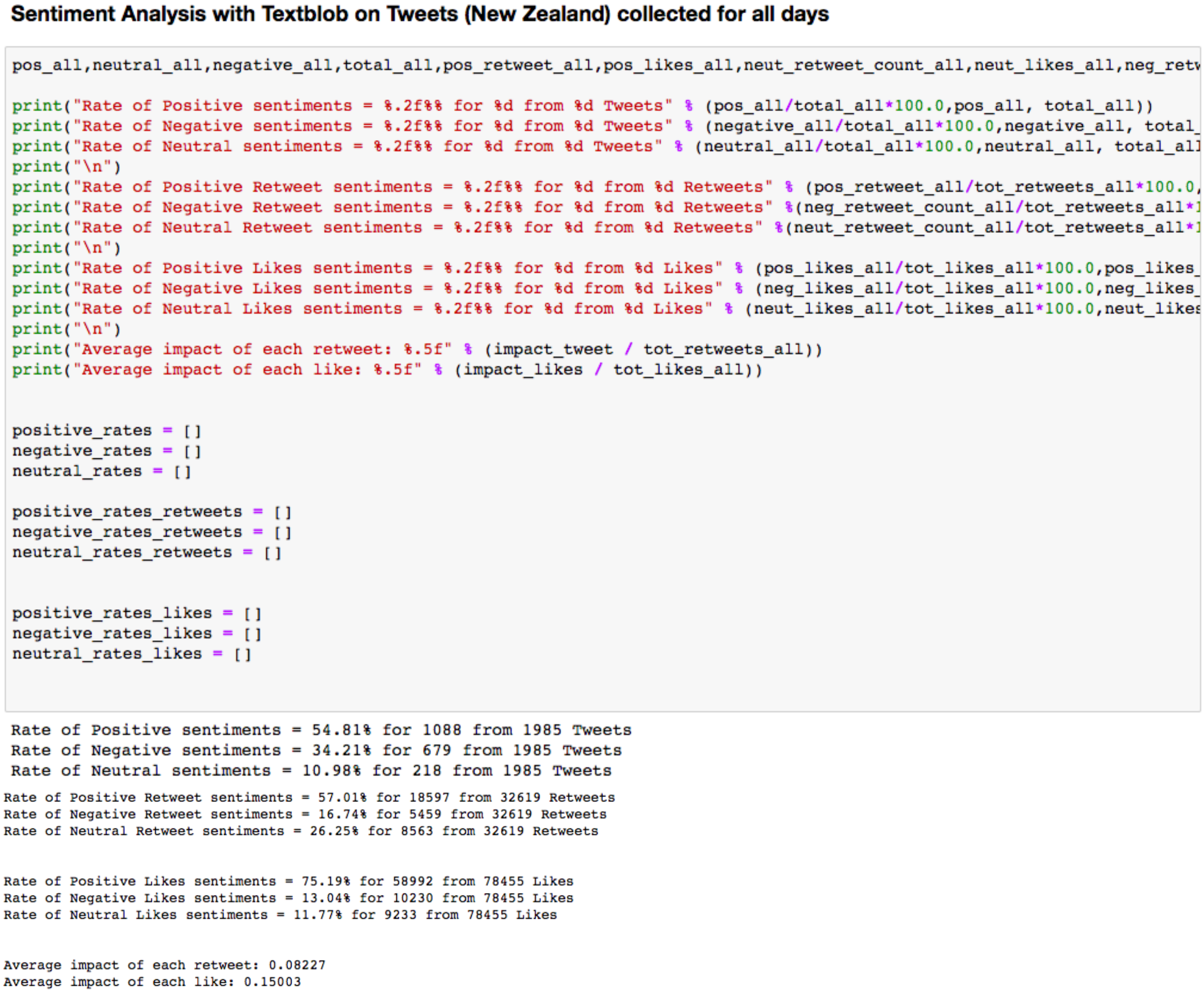
All tweets



Getting individual tweets dataframe for each day of New Zealand tweet collection



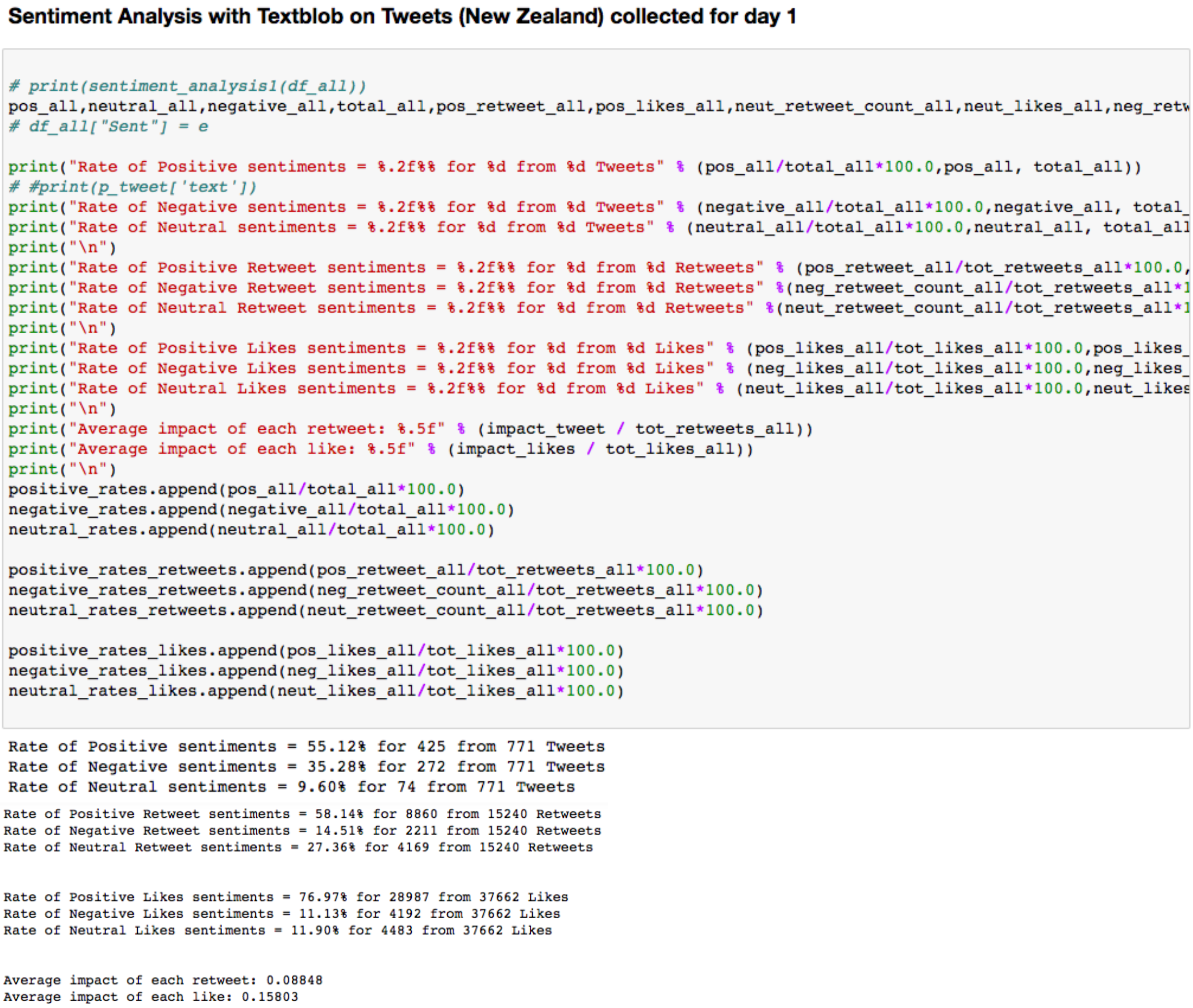
```
df1 = pd.read_csv('NewZealand_3_15_19.csv')
df1.columns = ['Id' if x=='Unnamed: 0' else 'Retweet_Count' if x=='Retweet Count' else 'Created_At' if x=='Created At'
df2 = pd.read_csv('NewZealand_3_16_19.csv')
df2.columns = ['Id' if x=='Unnamed: 0' else 'Retweet_Count' if x=='Retweet Count' else 'Created_At' if x=='Created At'
df3 = pd.read_csv('NewZealand_3_17_19.csv')
df3.columns = ['Id' if x=='Unnamed: 0' else 'Retweet_Count' if x=='Retweet Count' else 'Created_At' if x=='Created At'
df4 = pd.read_csv('NewZealand_3_18_19.csv')
df4.columns = ['Id' if x=='Unnamed: 0' else 'Retweet_Count' if x=='Retweet Count' else 'Created_At' if x=='Created At'
```



Sentiment Analysis with Textblob on Tweets (New Zealand) collected for all days

```
pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_retw

print("Rate of Positive sentiments = %.2f%% for %d from %d Tweets" % (pos_all/total_all*100.0,pos_all, total_all))
print("Rate of Negative sentiments = %.2f%% for %d from %d Tweets" % (negative_all/total_all*100.0,negative_all, total_
print("Rate of Neutral sentiments = %.2f%% for %d from %d Tweets" % (neutral_all/total_all*100.0,neutral_all, total_all
print("\n")
print("Rate of Positive Retweet sentiments = %.2f%% for %d from %d Retweets" % (pos_retweet_all/tot_retweets_all*100.0,
print("Rate of Negative Retweet sentiments = %.2f%% for %d from %d Retweets" %(neg_retweet_count_all/tot_retweets_all*1
print("Rate of Neutral Retweet sentiments = %.2f%% for %d from %d Retweets" %(neut_retweet_count_all/tot_retweets_all*1
print("\n")
print("Rate of Positive Likes sentiments = %.2f%% for %d from %d Likes" % (pos_likes_all/tot_likes_all*100.0,pos_likes_
print("Rate of Negative Likes sentiments = %.2f%% for %d from %d Likes" % (neg_likes_all/tot_likes_all*100.0,neg_likes_
print("Rate of Neutral Likes sentiments = %.2f%% for %d from %d Likes" % (neut_likes_all/tot_likes_all*100.0,neut_likes
print("\n")
print("Average impact of each retweet: %.5f" % (impact_tweet / tot_retweets_all))
print("Average impact of each like: %.5f" % (impact_likes / tot_likes_all))


positive_rates = []
negative_rates = []
neutral_rates = []

positive_rates_retweets = []
negative_rates_retweets = []
neutral_rates_retweets = []


positive_rates_likes = []
negative_rates_likes = []
neutral_rates_likes = []
```

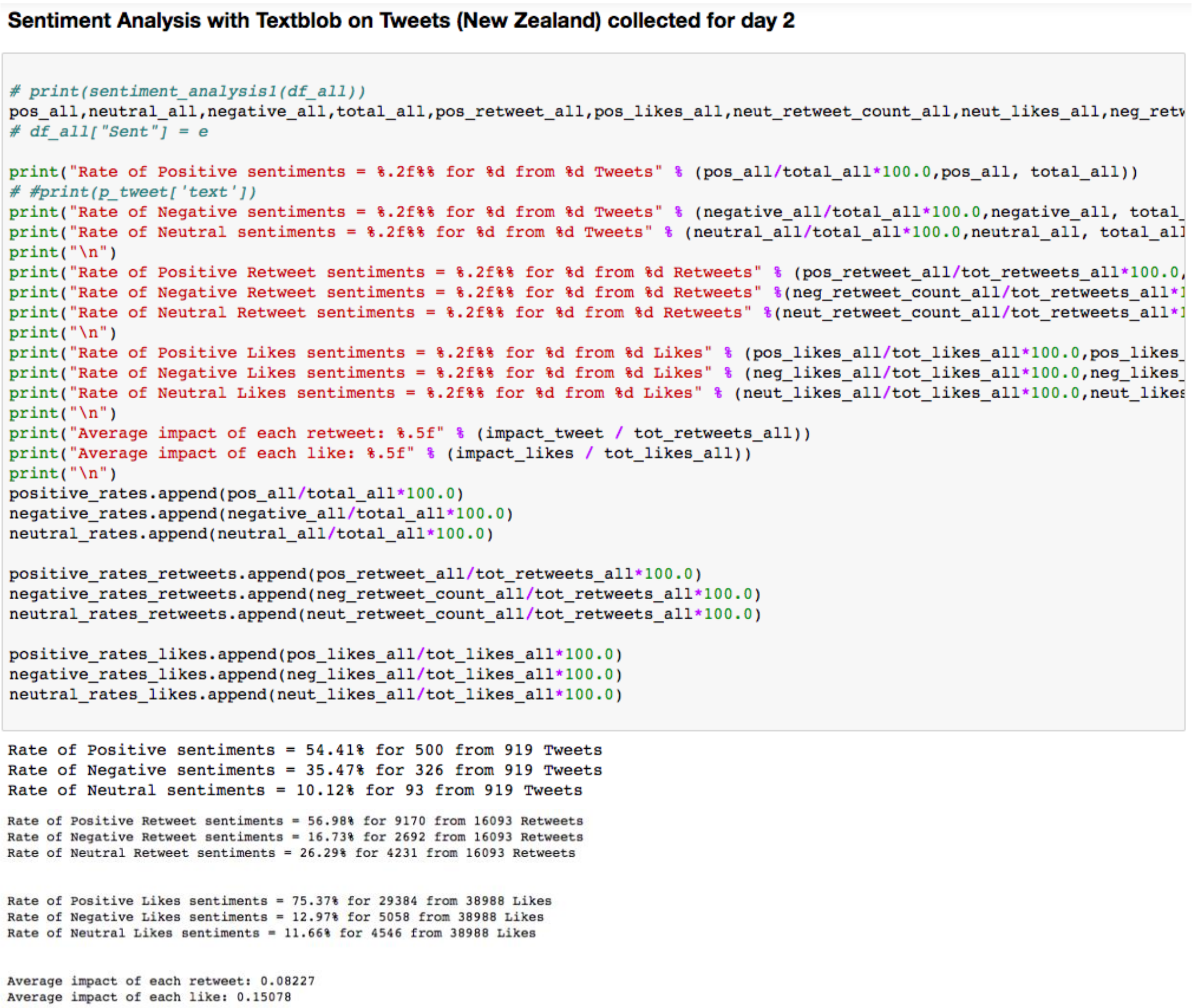
```
Rate of Positive sentiments = 54.81% for 1088 from 1985 Tweets
Rate of Negative sentiments = 34.21% for 679 from 1985 Tweets
Rate of Neutral sentiments = 10.98% for 218 from 1985 Tweets
Rate of Positive Retweet sentiments = 57.01% for 18597 from 32619 Retweets
Rate of Negative Retweet sentiments = 16.74% for 5459 from 32619 Retweets
Rate of Neutral Retweet sentiments = 26.25% for 8563 from 32619 Retweets


Rate of Positive Likes sentiments = 75.19% for 58992 from 78455 Likes
Rate of Negative Likes sentiments = 13.04% for 10230 from 78455 Likes
Rate of Neutral Likes sentiments = 11.77% for 9233 from 78455 Likes


Average impact of each retweet: 0.08227
Average impact of each like: 0.15003
```



Sentiment Analysis with Textblob on Tweets (New Zealand) collected for day 1

```
# print(sentiment_analysis1(df_all))
pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_retw
# df_all["Sent"] = e

print("Rate of Positive sentiments = %.2f%% for %d from %d Tweets" % (pos_all/total_all*100.0,pos_all, total_all))
# #print(p_tweet['text'])
print("Rate of Negative sentiments = %.2f%% for %d from %d Tweets" % (negative_all/total_all*100.0,negative_all, total_
print("Rate of Neutral sentiments = %.2f%% for %d from %d Tweets" % (neutral_all/total_all*100.0,neutral_all, total_all
print("\n")
print("Rate of Positive Retweet sentiments = %.2f%% for %d from %d Retweets" % (pos_retweet_all/tot_retweets_all*100.0,
print("Rate of Negative Retweet sentiments = %.2f%% for %d from %d Retweets" %(neg_retweet_count_all/tot_retweets_all*1
print("Rate of Neutral Retweet sentiments = %.2f%% for %d from %d Retweets" %(neut_retweet_count_all/tot_retweets_all*1
print("\n")
print("Rate of Positive Likes sentiments = %.2f%% for %d from %d Likes" % (pos_likes_all/tot_likes_all*100.0,pos_likes_
print("Rate of Negative Likes sentiments = %.2f%% for %d from %d Likes" % (neg_likes_all/tot_likes_all*100.0,neg_likes_
print("Rate of Neutral Likes sentiments = %.2f%% for %d from %d Likes" % (neut_likes_all/tot_likes_all*100.0,neut_likes
print("\n")
print("Average impact of each retweet: %.5f" % (impact_tweet / tot_retweets_all))
print("Average impact of each like: %.5f" % (impact_likes / tot_likes_all))
print("\n")
positive_rates.append(pos_all/total_all*100.0)
negative_rates.append(negative_all/total_all*100.0)
neutral_rates.append(neutral_all/total_all*100.0)

positive_rates_retweets.append(pos_retweet_all/tot_retweets_all*100.0)
negative_rates_retweets.append(neg_retweet_count_all/tot_retweets_all*100.0)
neutral_rates_retweets.append(neut_retweet_count_all/tot_retweets_all*100.0)

positive_rates_likes.append(pos_likes_all/tot_likes_all*100.0)
negative_rates_likes.append(neg_likes_all/tot_likes_all*100.0)
neutral_rates_likes.append(neut_likes_all/tot_likes_all*100.0)
```

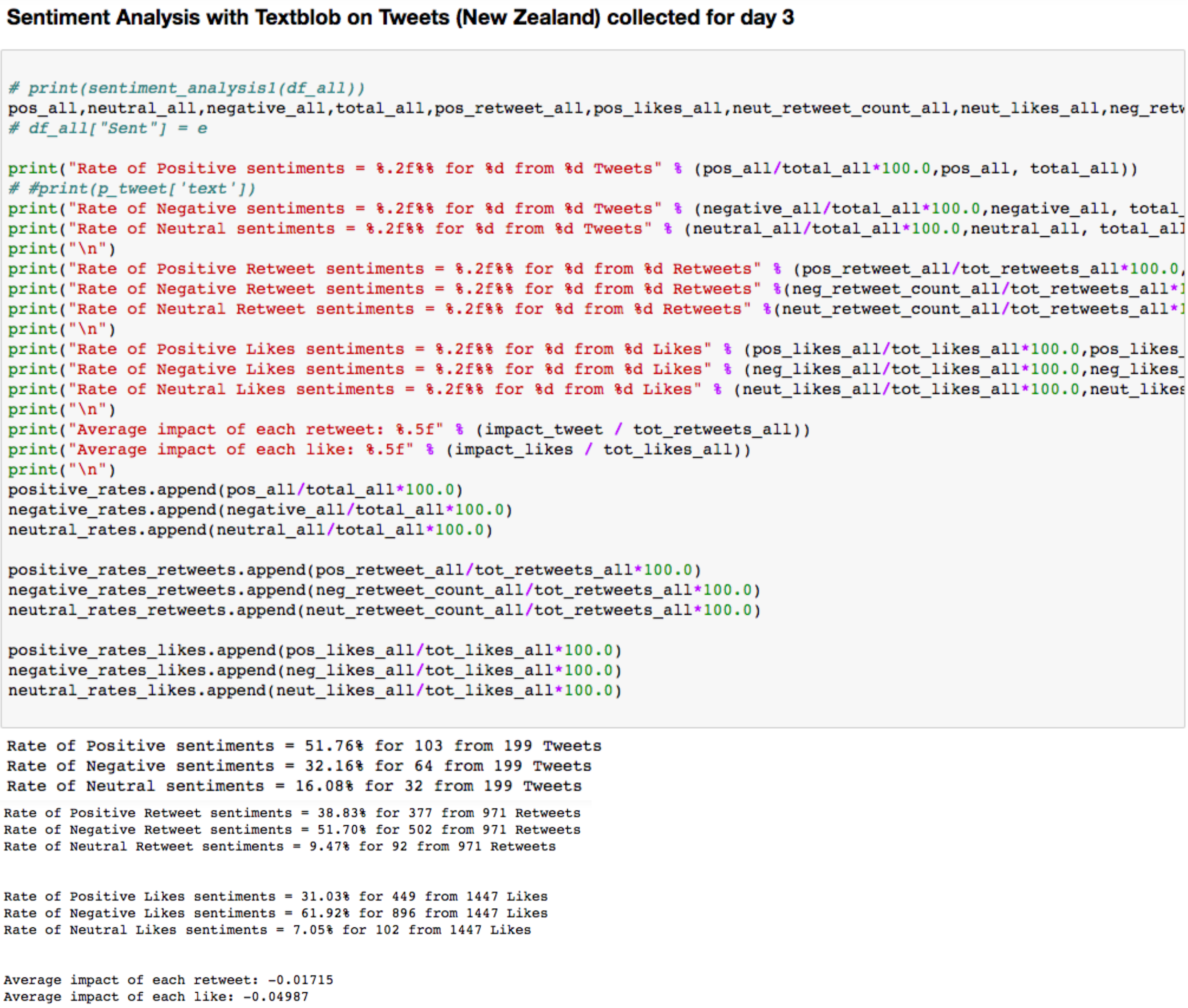
```
Rate of Positive sentiments = 55.12% for 425 from 771 Tweets
Rate of Negative sentiments = 35.28% for 272 from 771 Tweets
Rate of Neutral sentiments = 9.60% for 74 from 771 Tweets

Rate of Positive Retweet sentiments = 58.14% for 8860 from 15240 Retweets
Rate of Negative Retweet sentiments = 14.51% for 2211 from 15240 Retweets
Rate of Neutral Retweet sentiments = 27.36% for 4169 from 15240 Retweets


Rate of Positive Likes sentiments = 76.97% for 28987 from 37662 Likes
Rate of Negative Likes sentiments = 11.13% for 4192 from 37662 Likes
Rate of Neutral Likes sentiments = 11.90% for 4483 from 37662 Likes


Average impact of each retweet: 0.08848
Average impact of each like: 0.15803
```



Sentiment Analysis with Textblob on Tweets (New Zealand) collected for day 2

```
# print(sentiment_analysis1(df_all))
pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_retw
# df_all["Sent"] = e

print("Rate of Positive sentiments = %.2f%% for %d from %d Tweets" % (pos_all/total_all*100.0,pos_all, total_all))
# #print(p_tweet['text'])
print("Rate of Negative sentiments = %.2f%% for %d from %d Tweets" % (negative_all/total_all*100.0,negative_all, total_
print("Rate of Neutral sentiments = %.2f%% for %d from %d Tweets" % (neutral_all/total_all*100.0,neutral_all, total_all
print("\n")
print("Rate of Positive Retweet sentiments = %.2f%% for %d from %d Retweets" % (pos_retweet_all/tot_retweets_all*100.0,
print("Rate of Negative Retweet sentiments = %.2f%% for %d from %d Retweets" %(neg_retweet_count_all/tot_retweets_all*1
print("Rate of Neutral Retweet sentiments = %.2f%% for %d from %d Retweets" %(neut_retweet_count_all/tot_retweets_all*1
print("\n")
print("Rate of Positive Likes sentiments = %.2f%% for %d from %d Likes" % (pos_likes_all/tot_likes_all*100.0,pos_likes_
print("Rate of Negative Likes sentiments = %.2f%% for %d from %d Likes" % (neg_likes_all/tot_likes_all*100.0,neg_likes_
print("Rate of Neutral Likes sentiments = %.2f%% for %d from %d Likes" % (neut_likes_all/tot_likes_all*100.0,neut_likes
print("\n")
print("Average impact of each retweet: %.5f" % (impact_tweet / tot_retweets_all))
print("Average impact of each like: %.5f" % (impact_likes / tot_likes_all))
print("\n")
positive_rates.append(pos_all/total_all*100.0)
negative_rates.append(negative_all/total_all*100.0)
neutral_rates.append(neutral_all/total_all*100.0)

positive_rates_retweets.append(pos_retweet_all/tot_retweets_all*100.0)
negative_rates_retweets.append(neg_retweet_count_all/tot_retweets_all*100.0)
neutral_rates_retweets.append(neut_retweet_count_all/tot_retweets_all*100.0)

positive_rates_likes.append(pos_likes_all/tot_likes_all*100.0)
negative_rates_likes.append(neg_likes_all/tot_likes_all*100.0)
neutral_rates_likes.append(neut_likes_all/tot_likes_all*100.0)
```

```
Rate of Positive sentiments = 54.41% for 500 from 919 Tweets
Rate of Negative sentiments = 35.47% for 326 from 919 Tweets
Rate of Neutral sentiments = 10.12% for 93 from 919 Tweets

Rate of Positive Retweet sentiments = 56.98% for 9170 from 16093 Retweets
Rate of Negative Retweet sentiments = 16.73% for 2692 from 16093 Retweets
Rate of Neutral Retweet sentiments = 26.29% for 4231 from 16093 Retweets


Rate of Positive Likes sentiments = 75.37% for 29384 from 38988 Likes
Rate of Negative Likes sentiments = 12.97% for 5058 from 38988 Likes
Rate of Neutral Likes sentiments = 11.66% for 4546 from 38988 Likes


Average impact of each retweet: 0.08227
Average impact of each like: 0.15078
```

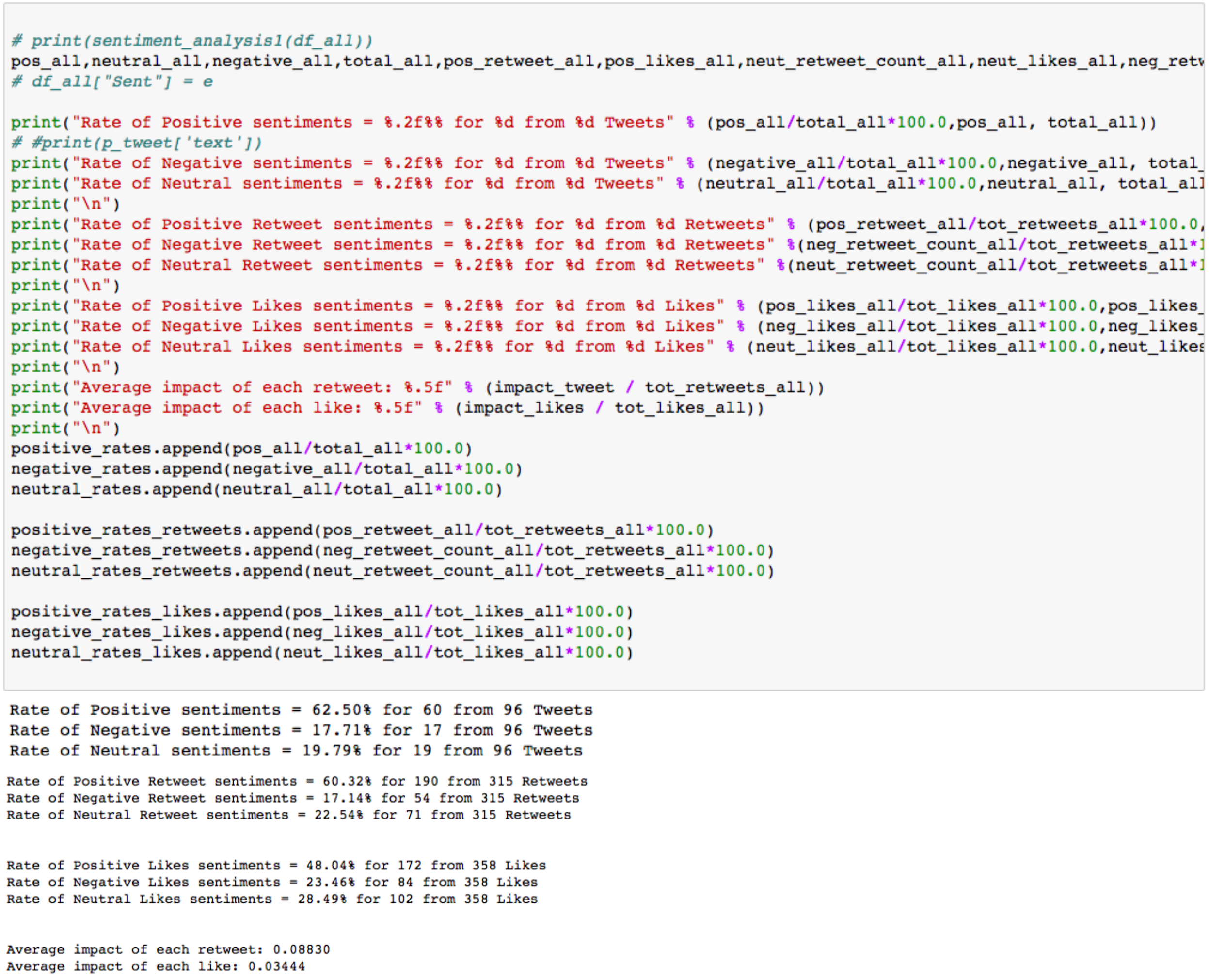


Sentiment Analysis with Textblob on Tweets (New Zealand) collected for day 3

```
# print(sentiment_analysis1(df_all))
pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_retw
# df_all["Sent"] = e

print("Rate of Positive sentiments = %.2f%% for %d from %d Tweets" % (pos_all/total_all*100.0,pos_all, total_all))
# #print(p_tweet['text'])
print("Rate of Negative sentiments = %.2f%% for %d from %d Tweets" % (negative_all/total_all*100.0,negative_all, total_
print("Rate of Neutral sentiments = %.2f%% for %d from %d Tweets" % (neutral_all/total_all*100.0,neutral_all, total_all
print("\n")
print("Rate of Positive Retweet sentiments = %.2f%% for %d from %d Retweets" % (pos_retweet_all/tot_retweets_all*100.0,
print("Rate of Negative Retweet sentiments = %.2f%% for %d from %d Retweets" %(neg_retweet_count_all/tot_retweets_all*1
print("Rate of Neutral Retweet sentiments = %.2f%% for %d from %d Retweets" %(neut_retweet_count_all/tot_retweets_all*1
print("\n")
print("Rate of Positive Likes sentiments = %.2f%% for %d from %d Likes" % (pos_likes_all/tot_likes_all*100.0,pos_likes_
print("Rate of Negative Likes sentiments = %.2f%% for %d from %d Likes" % (neg_likes_all/tot_likes_all*100.0,neg_likes_
print("Rate of Neutral Likes sentiments = %.2f%% for %d from %d Likes" % (neut_likes_all/tot_likes_all*100.0,neut_likes
print("\n")
print("Average impact of each retweet: %.5f" % (impact_tweet / tot_retweets_all))
print("Average impact of each like: %.5f" % (impact_likes / tot_likes_all))
print("\n")
positive_rates.append(pos_all/total_all*100.0)
negative_rates.append(negative_all/total_all*100.0)
neutral_rates.append(neutral_all/total_all*100.0)

positive_rates_retweets.append(pos_retweet_all/tot_retweets_all*100.0)
negative_rates_retweets.append(neg_retweet_count_all/tot_retweets_all*100.0)
neutral_rates_retweets.append(neut_retweet_count_all/tot_retweets_all*100.0)

positive_rates_likes.append(pos_likes_all/tot_likes_all*100.0)
negative_rates_likes.append(neg_likes_all/tot_likes_all*100.0)
neutral_rates_likes.append(neut_likes_all/tot_likes_all*100.0)
```

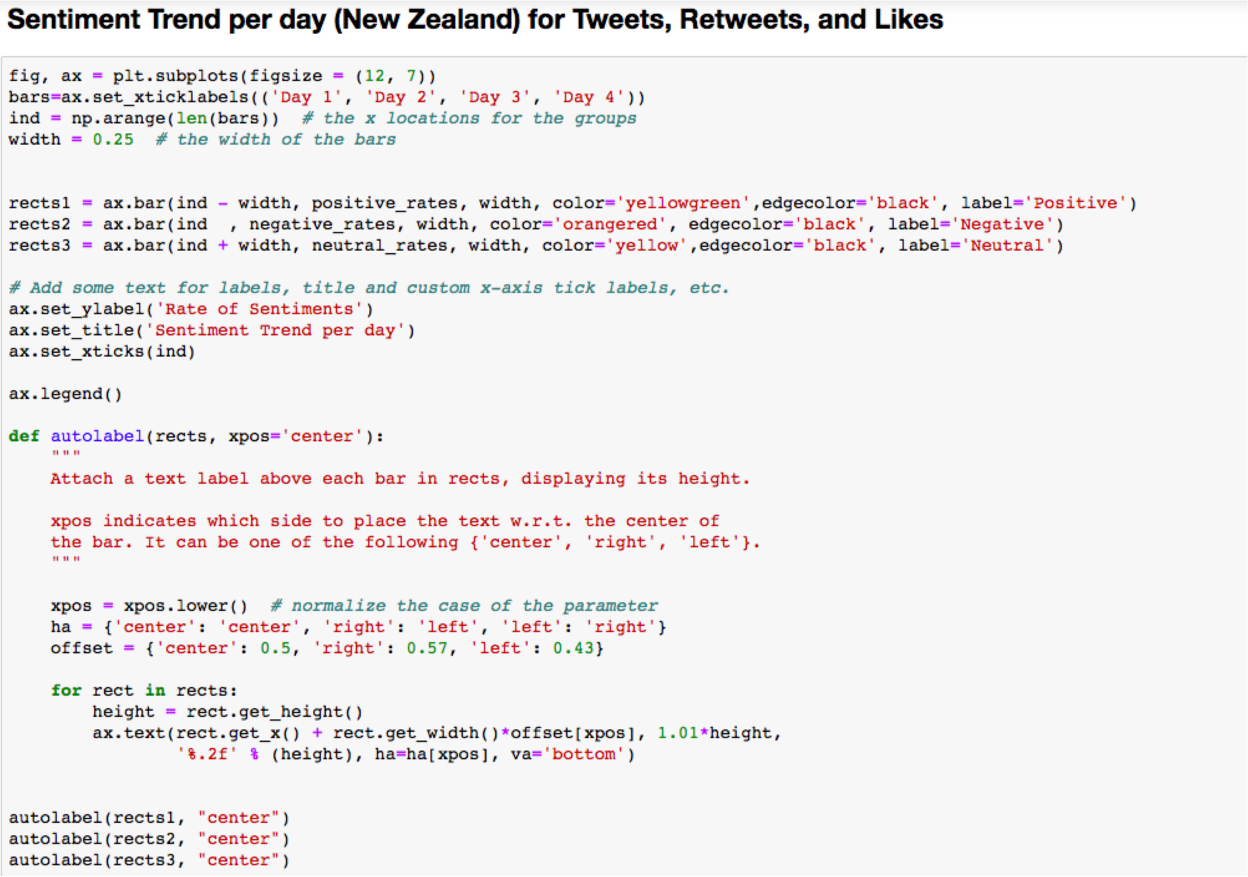
```
Rate of Positive sentiments = 51.76% for 103 from 199 Tweets
Rate of Negative sentiments = 32.16% for 64 from 199 Tweets
Rate of Neutral sentiments = 16.08% for 32 from 199 Tweets
Rate of Positive Retweet sentiments = 38.83% for 377 from 971 Retweets
Rate of Negative Retweet sentiments = 51.70% for 502 from 971 Retweets
Rate of Neutral Retweet sentiments = 9.47% for 92 from 971 Retweets


Rate of Positive Likes sentiments = 31.03% for 449 from 1447 Likes
Rate of Negative Likes sentiments = 61.92% for 896 from 1447 Likes
Rate of Neutral Likes sentiments = 7.05% for 102 from 1447 Likes


Average impact of each retweet: -0.01715
Average impact of each like: -0.04987
```

Sentiment Analysis with Textblob on Tweets (New Zealand) collected for day 4



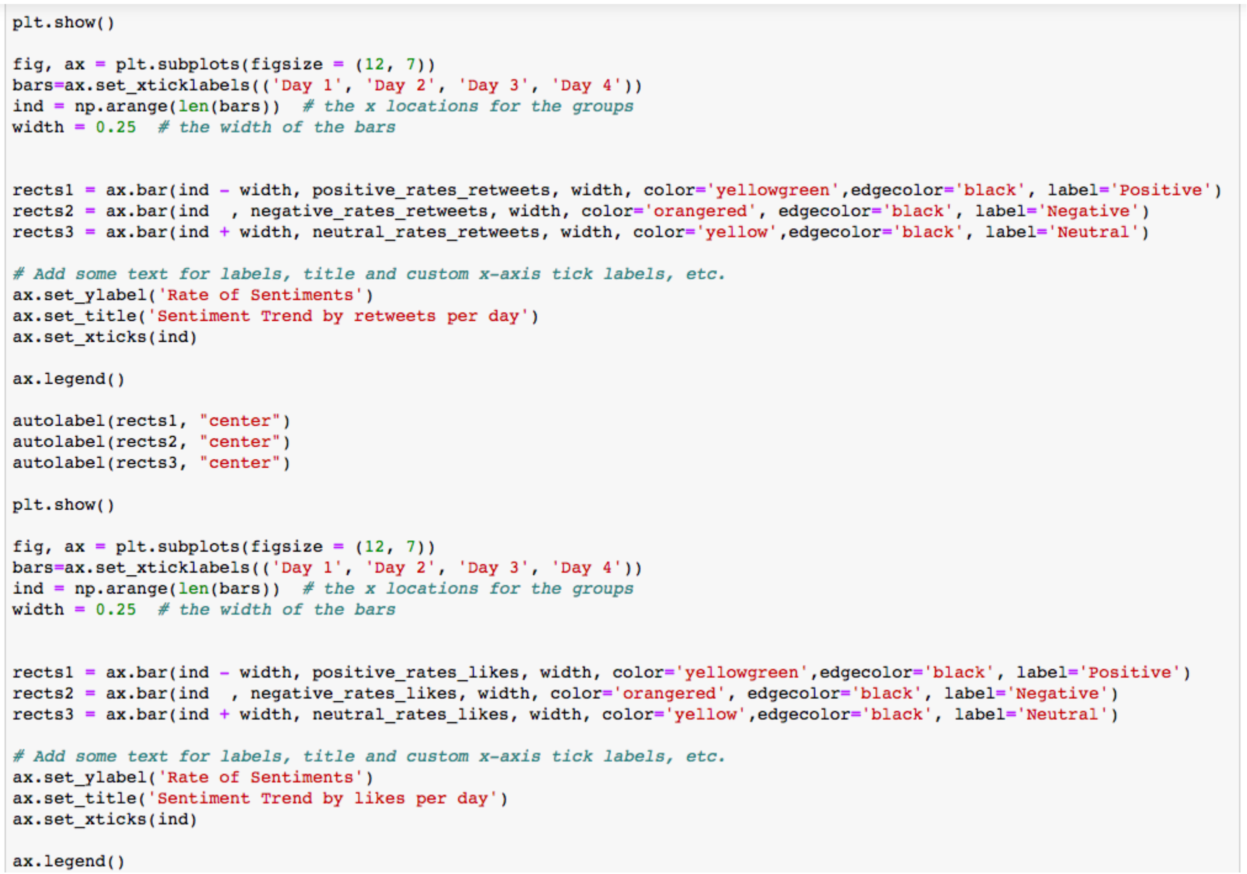
```
# print(sentiment_analysis1(df_all))
pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_retw
# df_all["Sent"] = e

print("Rate of Positive sentiments = %.2f%% for %d from %d Tweets" % (pos_all/total_all*100.0,pos_all, total_all))
# #print(p_tweet['text'])
print("Rate of Negative sentiments = %.2f%% for %d from %d Tweets" % (negative_all/total_all*100.0,negative_all, total_
print("Rate of Neutral sentiments = %.2f%% for %d from %d Tweets" % (neutral_all/total_all*100.0,neutral_all, total_all
print("\n")
print("Rate of Positive Retweet sentiments = %.2f%% for %d from %d Retweets" % (pos_retweet_all/tot_retweets_all*100.0,
print("Rate of Negative Retweet sentiments = %.2f%% for %d from %d Retweets" %(neg_retweet_count_all/tot_retweets_all*1
print("Rate of Neutral Retweet sentiments = %.2f%% for %d from %d Retweets" %(neut_retweet_count_all/tot_retweets_all*1
print("\n")
print("Rate of Positive Likes sentiments = %.2f%% for %d from %d Likes" % (pos_likes_all/tot_likes_all*100.0,pos_likes_
print("Rate of Negative Likes sentiments = %.2f%% for %d from %d Likes" % (neg_likes_all/tot_likes_all*100.0,neg_likes_
print("Rate of Neutral Likes sentiments = %.2f%% for %d from %d Likes" % (neut_likes_all/tot_likes_all*100.0,neut_likes
print("\n")
print("Average impact of each retweet: %.5f" % (impact_tweet / tot_retweets_all))
print("Average impact of each like: %.5f" % (impact_likes / tot_likes_all))
print("\n")
positive_rates.append(pos_all/total_all*100.0)
negative_rates.append(negative_all/total_all*100.0)
neutral_rates.append(neutral_all/total_all*100.0)

positive_rates_retweets.append(pos_retweet_all/tot_retweets_all*100.0)
negative_rates_retweets.append(neg_retweet_count_all/tot_retweets_all*100.0)
neutral_rates_retweets.append(neut_retweet_count_all/tot_retweets_all*100.0)

positive_rates_likes.append(pos_likes_all/tot_likes_all*100.0)
negative_rates_likes.append(neg_likes_all/tot_likes_all*100.0)
neutral_rates_likes.append(neut_likes_all/tot_likes_all*100.0)
```

```
Rate of Positive sentiments = 62.50% for 60 from 96 Tweets
Rate of Negative sentiments = 17.71% for 17 from 96 Tweets
Rate of Neutral sentiments = 19.79% for 19 from 96 Tweets

Rate of Positive Retweet sentiments = 60.32% for 190 from 315 Retweets
Rate of Negative Retweet sentiments = 17.14% for 54 from 315 Retweets
Rate of Neutral Retweet sentiments = 22.54% for 71 from 315 Retweets


Rate of Positive Likes sentiments = 48.04% for 172 from 358 Likes
Rate of Negative Likes sentiments = 23.46% for 84 from 358 Likes
Rate of Neutral Likes sentiments = 28.49% for 102 from 358 Likes


Average impact of each retweet: 0.08830
Average impact of each like: 0.03444
```

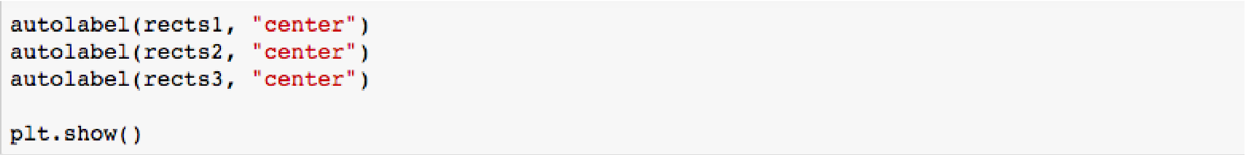


Sentiment Trend per day (New Zealand) for Tweets, Retweets, and Likes

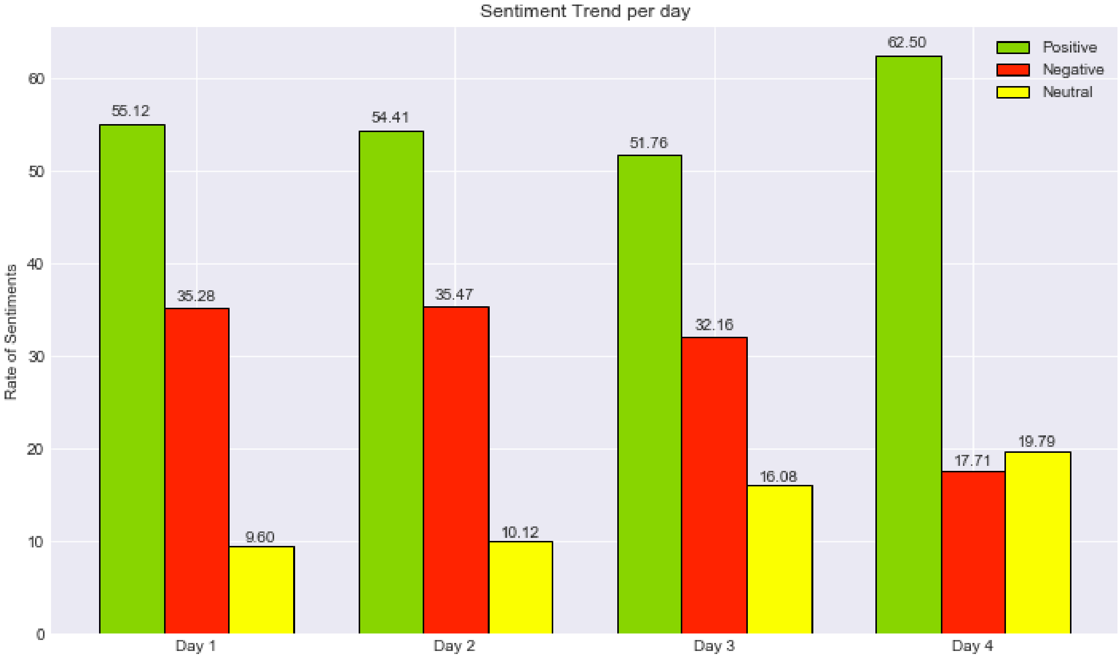
```
fig, ax = plt.subplots(figsize = (12, 7))
bars=ax.set_xticklabels(('Day 1', 'Day 2', 'Day 3', 'Day 4'))
ind = np.arange(len(bars))  # the x locations for the groups
width = 0.25  # the width of the bars


rects1 = ax.bar(ind - width, positive_rates, width, color='yellowgreen',edgecolor='black', label='Positive')
rects2 = ax.bar(ind  , negative_rates, width, color='orangered', edgecolor='black', label='Negative')
rects3 = ax.bar(ind + width, neutral_rates, width, color='yellow',edgecolor='black', label='Neutral')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Rate of Sentiments')
ax.set_title('Sentiment Trend per day')
ax.set_xticks(ind)

ax.legend()

def autolabel(rects, xpos='center'):
    """
    Attach a text label above each bar in rects, displaying its height.

    xpos indicates which side to place the text w.r.t. the center of
    the bar. It can be one of the following {'center', 'right', 'left'}.
    """

    xpos = xpos.lower()  # normalize the case of the parameter
    ha = {'center': 'center', 'right': 'left', 'left': 'right'}
    offset = {'center': 0.5, 'right': 0.57, 'left': 0.43}

    for rect in rects:
        height = rect.get_height()
        ax.text(rect.get_x() + rect.get_width()*offset[xpos], 1.01*height,
                '%.2f' % (height), ha=ha[xpos], va='bottom')


autolabel(rects1, "center")
autolabel(rects2, "center")
autolabel(rects3, "center")
```

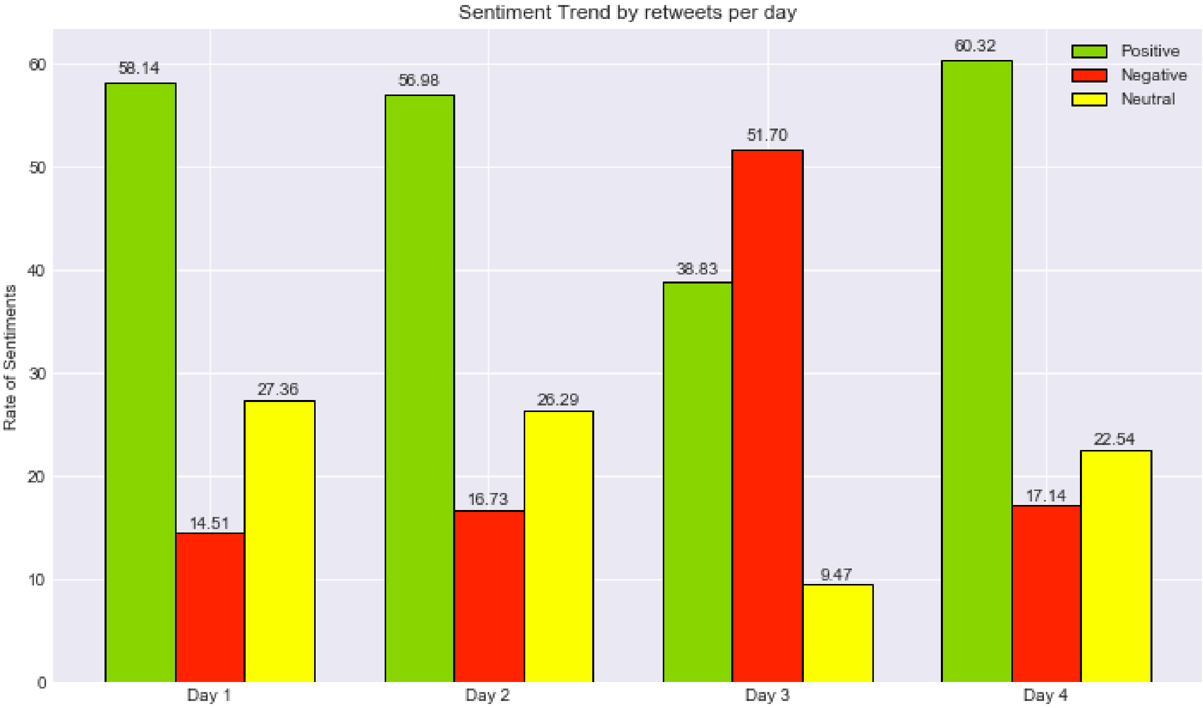


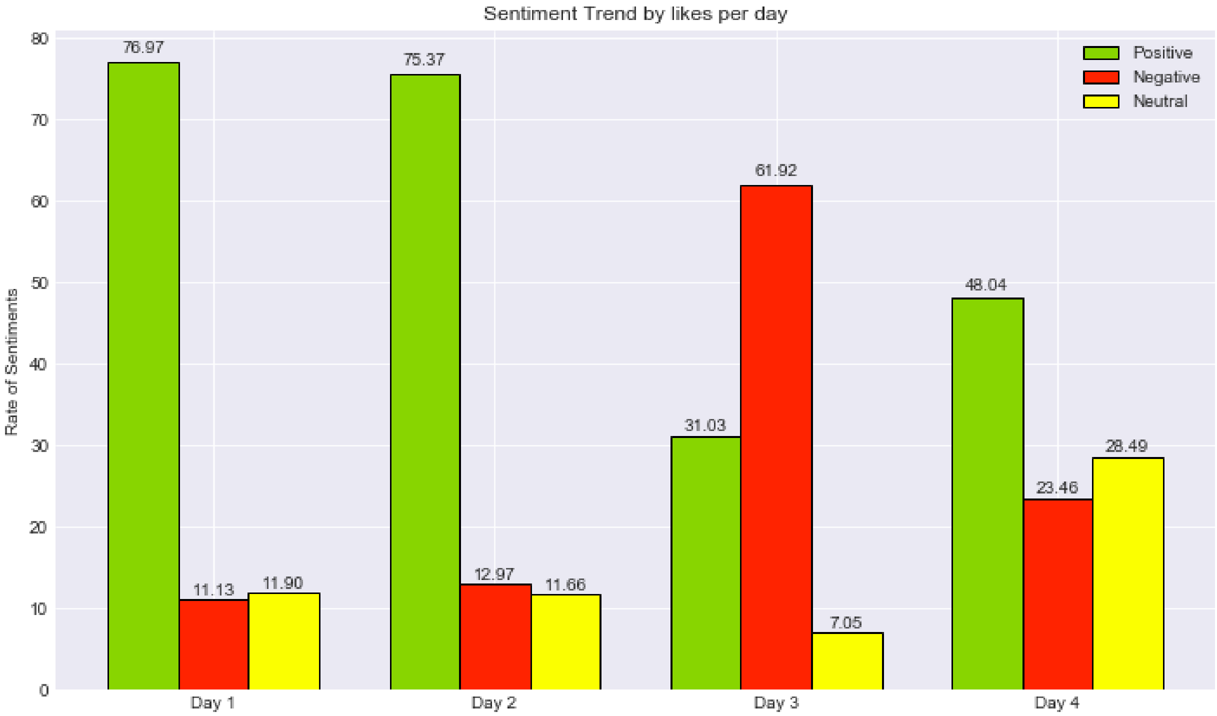
```
plt.show()

fig, ax = plt.subplots(figsize = (12, 7))
bars=ax.set_xticklabels(('Day 1', 'Day 2', 'Day 3', 'Day 4'))
ind = np.arange(len(bars))  # the x locations for the groups
width = 0.25  # the width of the bars


rects1 = ax.bar(ind - width, positive_rates_retweets, width, color='yellowgreen',edgecolor='black', label='Positive')
rects2 = ax.bar(ind  , negative_rates_retweets, width, color='orangered', edgecolor='black', label='Negative')
rects3 = ax.bar(ind + width, neutral_rates_retweets, width, color='yellow',edgecolor='black', label='Neutral')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Rate of Sentiments')
ax.set_title('Sentiment Trend by retweets per day')
ax.set_xticks(ind)

ax.legend()

autolabel(rects1, "center")
autolabel(rects2, "center")
autolabel(rects3, "center")

plt.show()

fig, ax = plt.subplots(figsize = (12, 7))
bars=ax.set_xticklabels(('Day 1', 'Day 2', 'Day 3', 'Day 4'))
ind = np.arange(len(bars))  # the x locations for the groups
width = 0.25  # the width of the bars


rects1 = ax.bar(ind - width, positive_rates_likes, width, color='yellowgreen',edgecolor='black', label='Positive')
rects2 = ax.bar(ind  , negative_rates_likes, width, color='orangered', edgecolor='black', label='Negative')
rects3 = ax.bar(ind + width, neutral_rates_likes, width, color='yellow',edgecolor='black', label='Neutral')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Rate of Sentiments')
ax.set_title('Sentiment Trend by likes per day')
ax.set_xticks(ind)

ax.legend()
```



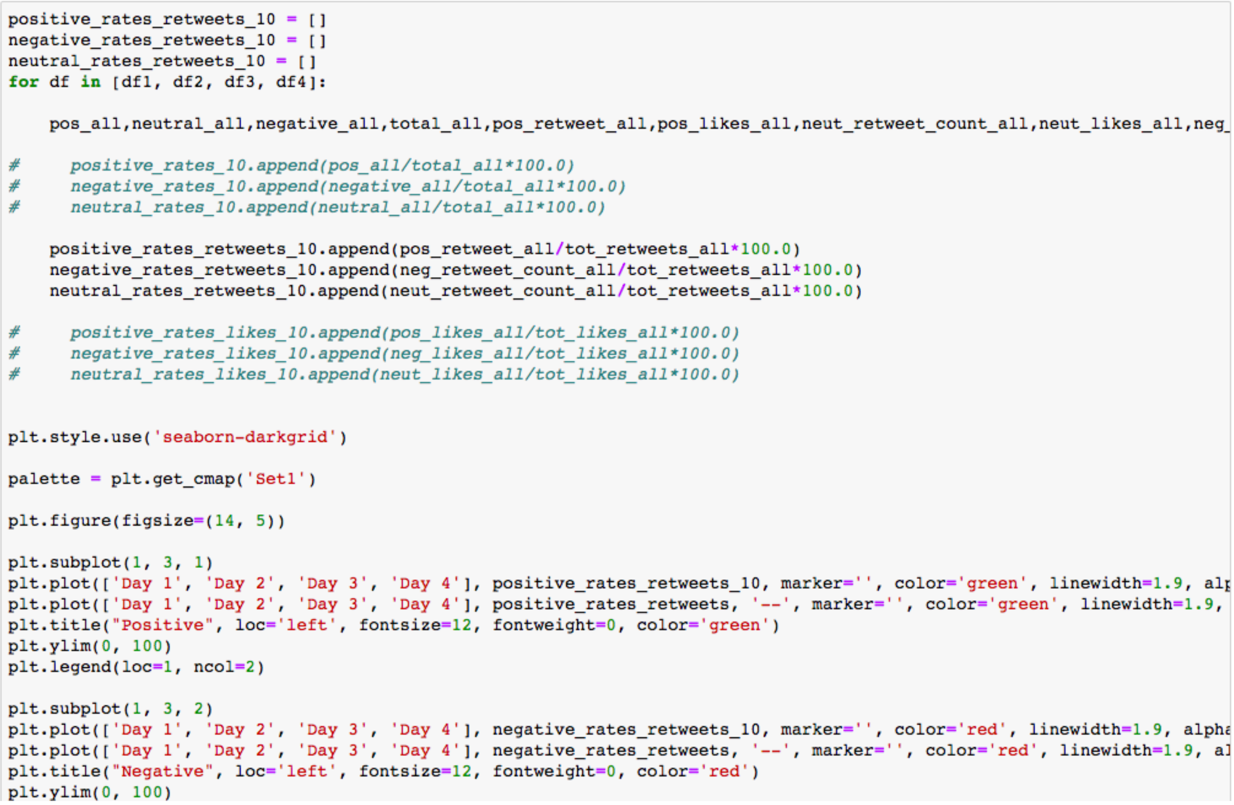
```
autolabel(rects1, "center")
autolabel(rects2, "center")
autolabel(rects3, "center")

plt.show()
```



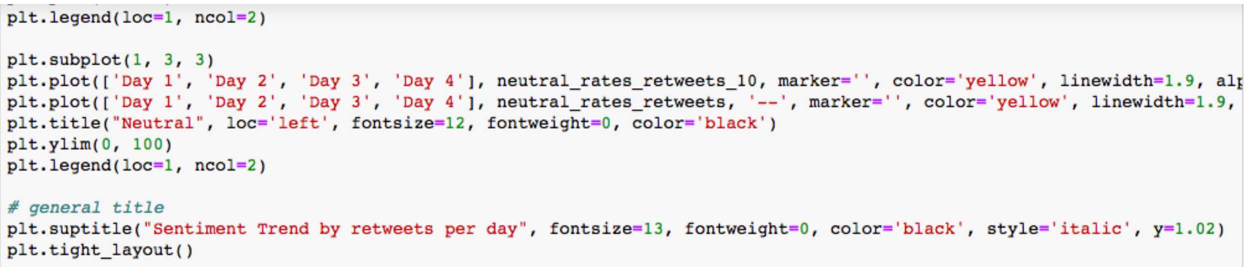




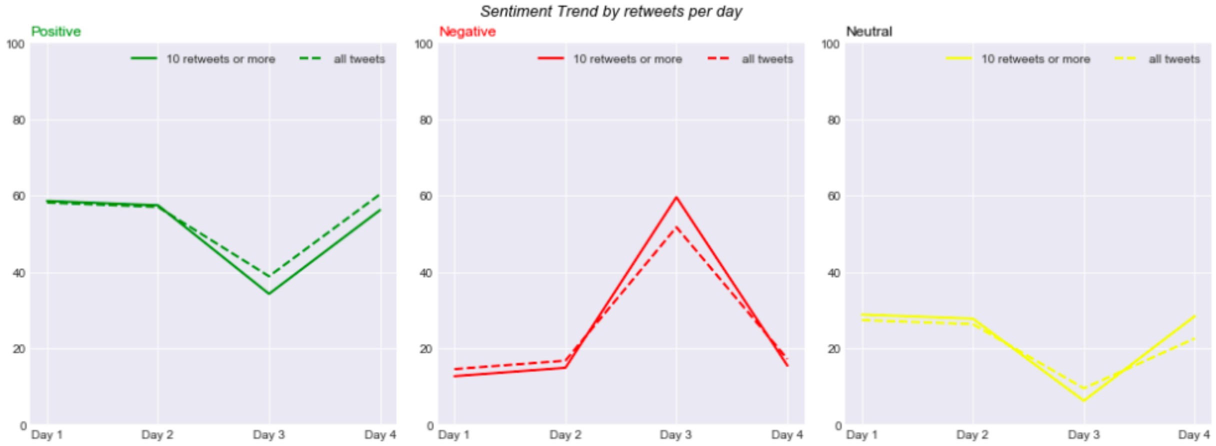
Line Chart Sentiment Analysis for New Zealand for 4 days



```
positive_rates_retweets_10 = []
negative_rates_retweets_10 = []
neutral_rates_retweets_10 = []
for df in [df1, df2, df3, df4]:

    pos_all,neutral_all,negative_all,total_all,pos_retweet_all,pos_likes_all,neut_retweet_count_all,neut_likes_all,neg_

#     positive_rates_10.append(pos_all/total_all*100.0)
#     negative_rates_10.append(negative_all/total_all*100.0)
#     neutral_rates_10.append(neutral_all/total_all*100.0)

    positive_rates_retweets_10.append(pos_retweet_all/tot_retweets_all*100.0)
    negative_rates_retweets_10.append(neg_retweet_count_all/tot_retweets_all*100.0)
    neutral_rates_retweets_10.append(neut_retweet_count_all/tot_retweets_all*100.0)

#     positive_rates_likes_10.append(pos_likes_all/tot_likes_all*100.0)
#     negative_rates_likes_10.append(neg_likes_all/tot_likes_all*100.0)
#     neutral_rates_likes_10.append(neut_likes_all/tot_likes_all*100.0)


plt.style.use('seaborn-darkgrid')

palette = plt.get_cmap('Set1')

plt.figure(figsize=(14, 5))

plt.subplot(1, 3, 1)
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], positive_rates_retweets_10, marker='', color='green', linewidth=1.9, alp
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], positive_rates_retweets, '--', marker='', color='green', linewidth=1.9,
plt.title("Positive", loc='left', fontsize=12, fontweight=0, color='green')
plt.ylim(0, 100)
plt.legend(loc=1, ncol=2)

plt.subplot(1, 3, 2)
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], negative_rates_retweets_10, marker='', color='red', linewidth=1.9, alpha
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], negative_rates_retweets, '--', marker='', color='red', linewidth=1.9, al
plt.title("Negative", loc='left', fontsize=12, fontweight=0, color='red')
plt.ylim(0, 100)
```



```
plt.legend(loc=1, ncol=2)

plt.subplot(1, 3, 3)
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], neutral_rates_retweets_10, marker='', color='yellow', linewidth=1.9, alp
plt.plot(['Day 1', 'Day 2', 'Day 3', 'Day 4'], neutral_rates_retweets, '--', marker='', color='yellow', linewidth=1.9,
plt.title("Neutral", loc='left', fontsize=12, fontweight=0, color='black')
plt.ylim(0, 100)
plt.legend(loc=1, ncol=2)

# general title
plt.suptitle("Sentiment Trend by retweets per day", fontsize=13, fontweight=0, color='black', style='italic', y=1.02)
plt.tight_layout()
```



Conclusion :

In hind sight, our acquisition and exploratory data analysis, showed very interesting and informative results of data.

Word Cloud and Sentiment Analysis with Textblob:

Boeing 737 Max 8:

> The word cloud that was formed with all four dates (3-13/3-16-19) showed a big plane containing the most frequent amount of words. The words that populated a lot of the stencil, were of neutral sentiment. This being said, most of the tweets being tweeted, contained neutral words that were neither good nor bad.

> In the Textblob Sentiment Analysis it is proven that the majority of all 8000 tweets were of neutral value. Neutral - (49.83%) or (3986 Tweets) out of 8000 Tweets. The sentiment analysis of Retweets and Likes were more Positive. Positive Retweets - (37.03%) or (8477 Retweets) out of 22,894 Retweets. Positive Likes - (42.49%) or (14909 Likes) out of 35,086 Likes.

New Zealand Attack: ¶

> The word cloud that was formed with all four dates (3-15/3-18-19) showed a kiwi containing the most frequent amount of words. The words that populated a lot of the stencil, at first look full of negative tweets. The analysis of the sentiments, proved that there is more positive sentiment in the WordCloud than negative.

> In the Textblob Sentiment Analysis it is proven that the majority of all 1985 tweets were of positive value. Positive - (54.81%) or (1088 Tweets) out of 1985 Tweets. The sentiment analysis of Retweets and Likes were more Positive as well. Positive Retweets - (57.01%) or (18,597 Retweets) out of 32,619 Retweets. Positive Likes - (75.19%) or (58,992 Likes) out of 78,455 Likes.