Data Manipulation in Python (CS2006 P2)

Introduction

On the 12th of November 2014, the European Space Agency lander Philae made the first ever soft landing of a spacecraft on the surface of a comet, 67P/Churyumov-Gerasimenko, having been carried there by the probe Rosetta. The news of the acheivement was disseminated through various social media platforms over the following days and weeks, including Twitter. This notebook analyses data independently gathered from Twitter to track the volume and nature of user activity related to the landing over the period of 3 weeks after the landing.

Functionality

All of the basic requirements were implemented as outlined in the practical specifaction. The following are any extensions which have been implemented:

```
Easy Extensions:
    Analysed Applications used to send Tweets

Medium Extensions:
    Analysed user patterens over the period of the dataset

Medium to hard Extensions:
    Added more interactive graphs

Other Extensions outwith the basic specifaction:
    Analysed different languages used in the dataset
    Analysed different Hashtags used in the dataset
```

In [4]:

```
#Imports used throughout the program
import pandas as pd
import matplotlib.pyplot as plt;
import math
import numpy as np

#Used for the more interactive graphs works towards HARD EXTENSION 1
from plotly import __version__
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)


import plotly.graph_objs as go
import operator
from collections import *

#Used for the word cloud
from wordcloud import WordCloud, STOPWORDS
```

In [5]:

```
#Reads the CSV File
df=pd.read_csv("../data/CometLanding.csv",encoding="UTF-8")
```

In [6]:

```
#Drops the duplicates from the dataset
df.drop_duplicates(['id_str'],inplace = True)
```

In [7]:

```
#Gets the total number of tweets
numTweets = len(df)
```

The raw data contained some duplicate tweets, which were removed, and the number of total remaining unique tweets is displayed below:

In [8]:

```
df = df[df['text'].notnull()]
```

The number of unique users is displayed below:

In [9]:

```
numUniqueUsers = len(df['from_user'].unique())
```

Language

The following section of the notebook looks at the different languages that appear throughout the set. This sections displays all of the different languages used throughout the dataset, along with a pie-chart displaying what share, the most used languages, make up in the dataset.

In [10]:

```
#Gets different languages from the data
language = df.groupby('user_lang')
```

In [11]:

```
#Used to loop through the different languages and counting how many times they a
ppear in the dataset
languages = []


for index, row in df.iterrows():
        text = (row['user_lang'])
        languages.append(text)


languageCount = {}

for lang in languages:
    if lang not in languageCount:
        languageCount[lang] = 1
    else:
        counter = languageCount.get(lang,'none')
        languageCount.update({lang: counter+1})
```

In [12]:

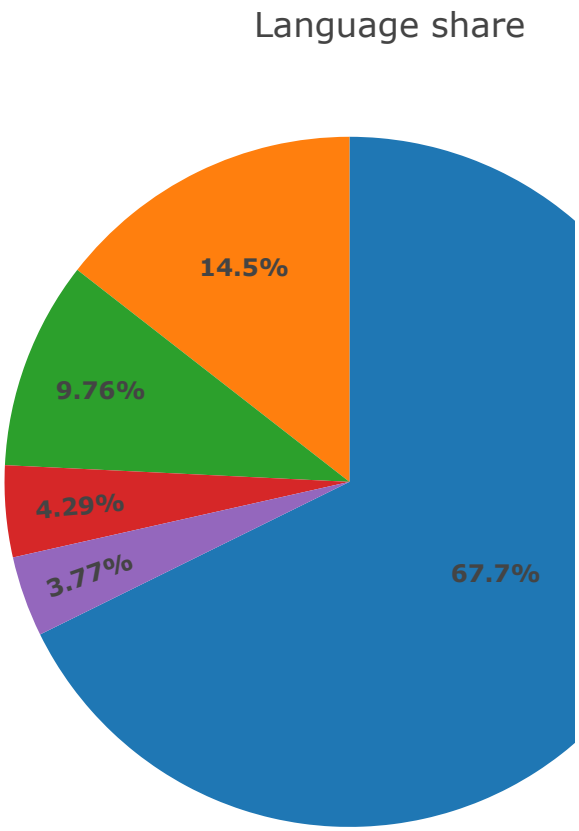
```
#Sorts the languages dictionary in order of the highest value at the top
topLangs = sorted(languageCount.items(), key=operator.itemgetter(1),reverse=True
)
```

In [13]:

```
#Gets the top 4 languages
topLanguage = topLangs[0][0]
topLanguageNum = topLangs[0][1]

secondLanguage = topLangs[1][0]
secondLanguageNum = topLangs[1][1]

thirdLanguage = topLangs[2][0]
thirdLanguageNum = topLangs[2][1]

fourthLanguage = topLangs[3][0]
fourthLanguageNum = topLangs[3][1]

totalLanguage = sum(languageCount.values())
otherLanguageNum = (totalLanguage-(topLanguageNum+secondLanguageNum+thirdLanguag
eNum+fourthLanguageNum))
```

In [14]:

```
#Displays the top languages in the pie-chart
fig = {
    'data': [{'labels': [topLanguage,secondLanguage,thirdLanguage,fourthLanguage
," \"Other\""],
               'values': [topLanguageNum,secondLanguageNum,thirdLanguageNum,fourt
hLanguageNum,otherLanguageNum],
               'type': 'pie'}],
    'layout': {'title': 'Language share'}
     }

iplot(fig)
```



The above pie chart shows the share each of language has on the tweets in the dataset. The chart clearly shows that english was the most used language taking over 2/3 of the set with 67.7% of the tweets in the set being written in english. This would be as expected as the comet launch was a eurpoean launch and english is the most spoken language in europe. This then shows that english would be the most used language as the majority of people who would be tweeting about the comet landing. Having english as the most used language would also be expected due to people from the US tweeting about the landing. users in america would be thought to be tweeting about the event due to many americans having an interest in space landings and probes, this would then increase the number of tweets being sent in english.

The next three popular languages are Spanish, French and German, this would also be expected to be the case due to the landing mission being european. This means that people in those countrys having a specific interest in the landing due to their country having something to do with the landing.

General Tweet Data

The following section will outline some of the more general aspects of the dataset. It highlights the total number of Tweets, Replies and Retweets and displays these values in a graph to show how the dataset is made up.

In [15]:

```
print("Total number of Tweets: " + str(numTweets))
```

```
Total number of Tweets: 77268
```

In [16]:

```
#Gets the number of tweets which
dfNoRT = df[~df.text.str.startswith('RT', na=False)]
```

In [17]:

```
print("Total number of unique users: "+ str(numUniqueUsers))
```

```
Total number of unique users: 50195
```

In [18]:

```
#Loops through the in_reply_to_screen_name column to see if a tweet is in reply
 to someon
dfReplies = df
numReplies = 0

for index, row in dfReplies.iterrows():
        text = (row['in_reply_to_screen_name'])
        if(not pd.isnull(text)):
            numReplies +=1
```

In [19]:

```
print("Total number of replies: " + str(numReplies))
```

```
Total number of replies: 1723
```

In [20]:

```
numReTweets = numTweets - ((len(dfNoRT)+numReplies))
```

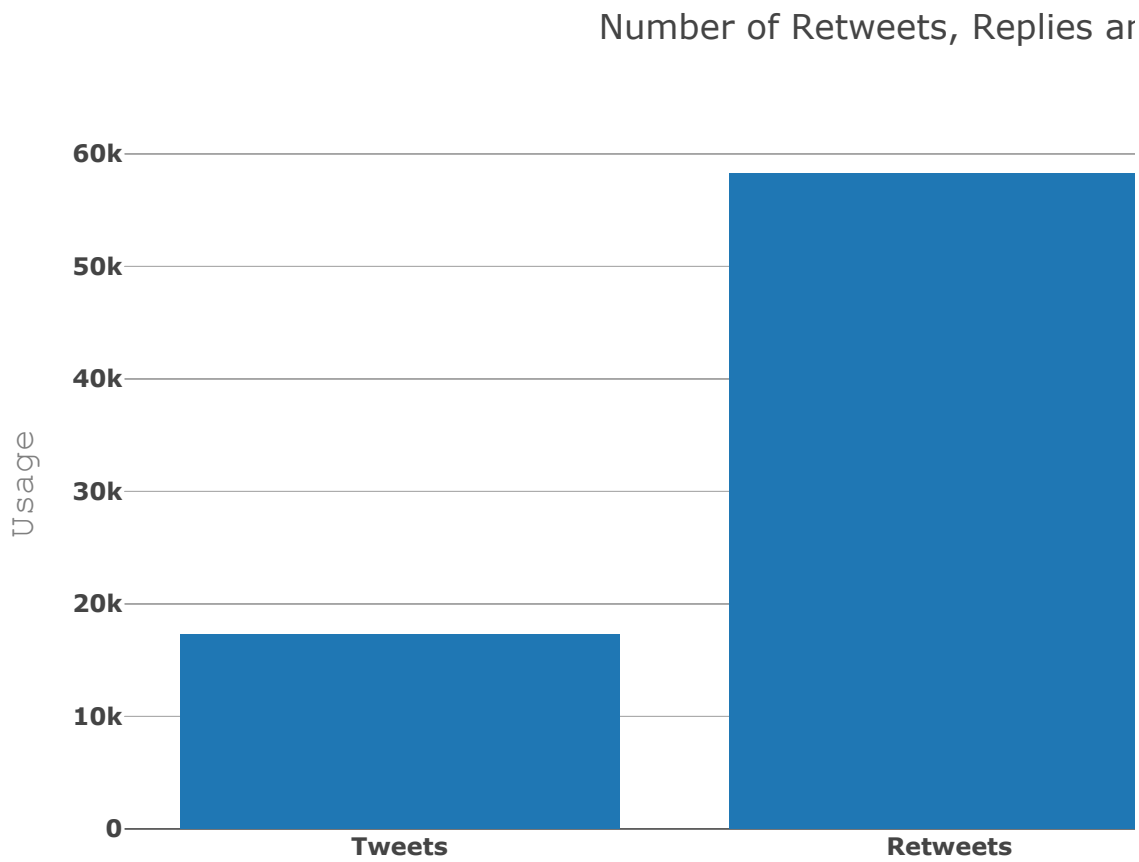
In [21]:

```
print("Total number of retweets: " + str(numReTweets))
```

```
Total number of retweets: 58276
```

In [22]:

```
#Displays bar-graph for tweets replies and retweets
numTweets = numTweets-(numReTweets+numReplies)
data = [go.Bar(x=['Tweets', 'Retweets', 'Replies'],y=[numTweets,numReTweets,numR
eplies])]
layout = go.Layout(
    title='Number of Retweets, Replies and Tweets',
    yaxis=dict(
        title='Usage',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    )
)

fig = go.Figure(data=data, layout=layout)
iplot(fig, filename='basic-bar')
```



The above graph shows the total number of unique Tweets that were in the data set. The graph then highlights how many of those tweets are Retweets and how many are replies to other tweets. As can be clearly seen from the graph the majority of the dataset is made up of Retweets with over 58 thousand Retweets being present. The set also includes a number of replies to tweets, however, when the amount of replies is compared to the number of Retweets it shows how many more Retweets there are over replies. This is shown with there only being 1723 replies in the set compared with the over 58 thousand Retweets.

One of the main reaons for this mass difference in the number of Retweets to replies is the nature of what each of them actually do. Retweeting a tweet is more designed to share that perticular tweet with your followers as it may be something interesting or something you agree with. Whereas a reply is used for the more conversational aspect of twitter it is there to add a response or to add more information to a tweet and in some cases it can be used as a convorsation tool between people.

Another possible reason for Retweets taking up a larger amount of the dataset over replies is that they are mush easier to utilise than replies. When a user goes to Retweet a tweet then they can simply just click Retweet which will then share this tweet with their followers. Whereas with a reply more thought must be put into the process as the actual content of the reply must be thought of and written out. This may be one of the reasons for Retweets being more prominent as simply they are quicker and easier to use.

Hashtags

This section of the notebook looks at the different hashtags which are present throughout the dataset. Highlighted is the most used hashtags with all of the hashtags which have been used over 150 times are displayed, a word cloud has also been created to display the different hashtags excluding the main hashtag '#CometLanding'. Finally a bar graph has been made to visually show how many times each of the top four hashtags have been used and how they compare to each other.

In [23]:

```
#Adds all the hashtags to an array
import re
hashtags = []
for index, row in dfNoRT.iterrows():
    text = (row['text'].split(" "))
    for token in text:
        re.sub('[\W_]', '', token)
        if token.startswith('#'):
            hashtags.append(str(token))
```

In [24]:

```
#Creates dictionary of hashtags and number of times they are used
hashtagCount = {}

for hashtag in hashtags:
    if hashtag not in hashtagCount:
        hashtagCount[hashtag] = 1
    else:
        counter = hashtagCount.get(hashtag,'none')
        hashtagCount.update({hashtag: counter+1})

#Prints hashtags which have been used more than 150 times
for key,val in hashtagCount.items():
    if val>150:
        print (repr(key) + "=>" + repr(val))
```

```
'#cometlanding'=>1834
'#CometLanding'=>12741
'#ESA'=>194
'#Rosetta'=>1471
'#Philae'=>734
'#CometLanding:'=>161
'#rosettamission'=>169
'#Cometlanding'=>165
'#67P'=>400
'#CometLanding.'=>244
'#rosetta'=>178
'#WishKoSaPasko'=>929
'#HappyBirthdaySandaraPark'=>928
```

The above shows the hashtags which have been used over 150 times. As can be seen from the set above hashtags relating to the comet landing make up some of the top hashtags used throughout the dataset.

In [25]:

```
#Sorts the hashtags by number of times used
topHashtags = sorted(hashtagCount.items(), key=operator.itemgetter(1),reverse=Tr
ue)
```

In [26]:

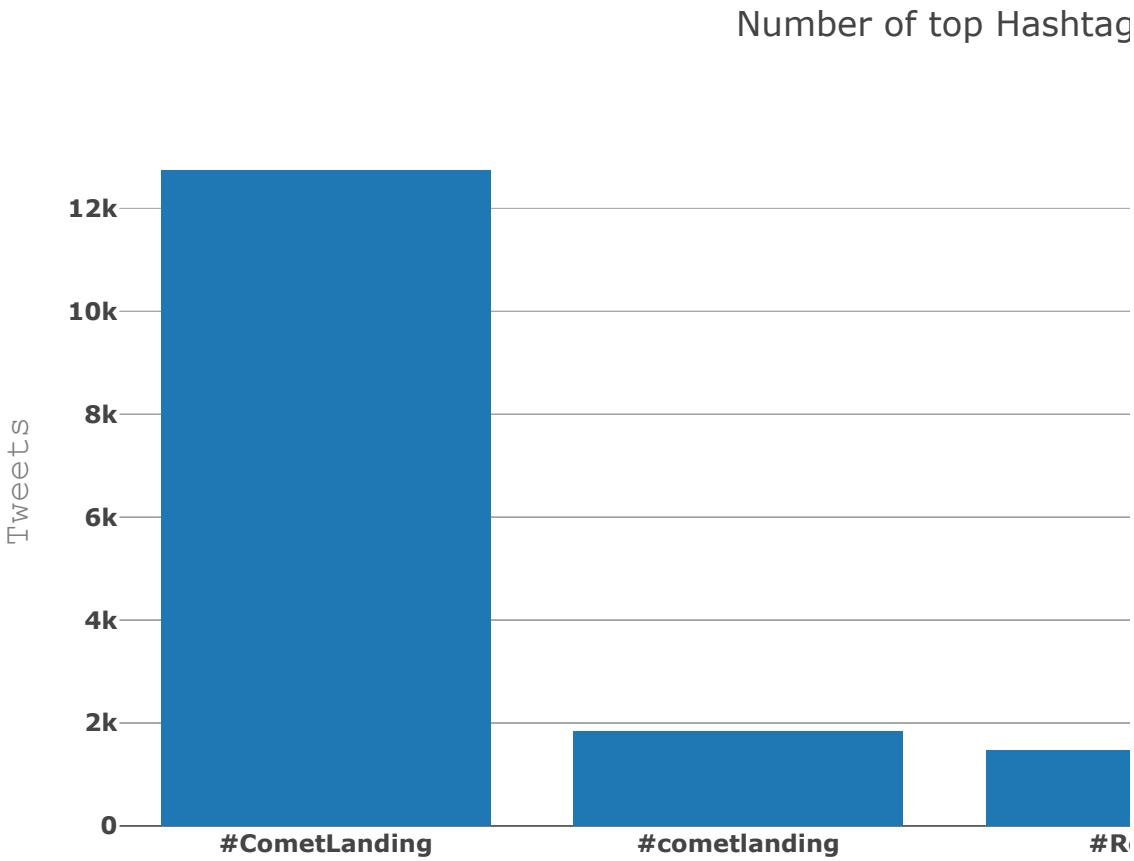
```
#Used to create the word cloud
words = []

for key,val in hashtagCount.items():
    words.append(key)

words = [e[1:] for e in words]
stopwords = set(STOPWORDS)
stopwords.add("CometLanding")


wordcloud = WordCloud(background_color='white',stopwords=stopwords,max_words=300
00,max_font_size=40, random_state=42).generate(str(hashtags))
plt.figure(figsize=(10,10))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

In [27]:

```
#Gets the top 4 hashtags
topHash = topHashtags[0][0]
topHashNum = topHashtags[0][1]

secondHash = topHashtags[1][0]
secondHashNum = topHashtags[1][1]

thirdHash = topHashtags[2][0]
thirdHashNum = topHashtags[2][1]

fourthHash = topHashtags[3][0]
fourthHashNum = topHashtags[3][1]
```

In [28]:

```
#Creates bar graph for top hashtags
data = [go.Bar(y=[topHashNum, secondHashNum, thirdHashNum,fourthHashNum],x=[topH
ash, secondHash,thirdHash,fourthHash])]
layout = go.Layout(
    title='Number of top Hashtags',
    yaxis=dict(
        title='Tweets',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    )
)

fig = go.Figure(data=data, layout=layout)
iplot(fig, filename='basic-bar')
```



The above chart shows the number of times each of the top four hashtags were used. As can be seen from the chart the top three hashtags directly relate to the Comet Landing. This does not come as a surprise as a large portion of the data set contains tweets about the Comet landing so it would make sense for users who are tweeting about the Comet Landing to include hashtags relating to it. Hashtags are a way for users to link their tweets into topics in which other users are talking about. This then means that when a user looks at a particular hashtag they will be able to see all of the tweets which have used the hashtag, which then allows users to see what different people have thought about the topic in question. This then shows why it would make sense for users who are tweeting about the Comet Landing to include '#CometLanding' in their tweet so users who are looking at tweets about the Comet Landing can see what the users sending the tweets are saying.

Applications (Easy Extension 1)

This section of the notebook examines the different applications which were used in the dataset to send tweets. A pie-chart has been created to highlight which devices were used to send the tweets and what percentage of tweets were sent from that application.

In [29]:

```
#Gets all the different applications and adds them to an array
dfSource = df
import re
items = []
for index, row in dfSource.iterrows():
    text = (row['source'])
    for token in str(text):
        if token.endswith('>'):
            split1 = text.split("</a>")
            split2 = str(split1).split(">")
            split2 = str(split2).split(",")
            items.append(str(split2[1]))
```

In [30]:

```
#Adds the different applications to a dictionary and counts number of times they
 have been used
appCount = {}

for device in items:
    if device not in appCount:
        appCount[device] = 1
    else:
        counter = appCount.get(device,'none')
        appCount.update({device: counter+1})
```

In [31]:

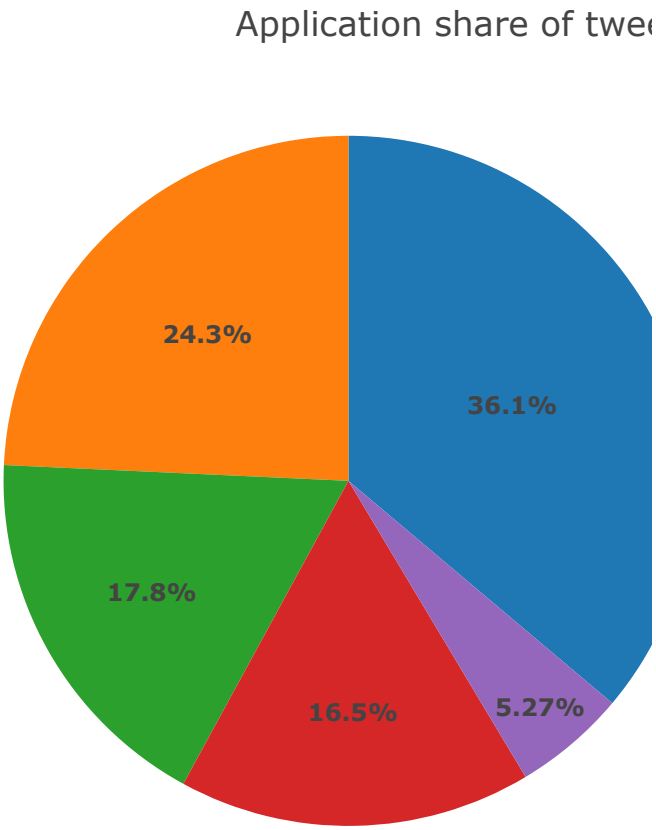
```
#Sorts the application dictionary by number of times used
topApplications = sorted(appCount.items(), key=operator.itemgetter(1),reverse=Tr
ue)
```

In [32]:

```
#Gets the top 4 applications
topApplication = topApplications[0][0]
topApplicationNum = topApplications[0][1]

secondApplication = topApplications[1][0]
secondApplicationNum = topApplications[1][1]

thirdApplication = topApplications[2][0]
thirdApplicationNum = topApplications[2][1]

fourthApplication = topApplications[3][0]
fourthApplicationNum = topApplications[3][1]

totalApplications = sum(appCount.values())
otherApplicationNum = (totalApplications-(topApplicationNum+secondApplicationNum
+thirdApplicationNum+fourthApplicationNum))
```

In [33]:

```
#Displays the pie-chart for the applications used
fig = {
    'data': [{'labels': [topApplication,secondApplication,thirdApplication,fourt
hApplication," \"Other\'"],
              'values': [topApplicationNum,secondApplicationNum,thirdApplication
Num,fourthApplicationNum,otherApplicationNum],
              'type': 'pie'}],
    'layout': {'title': 'Application share of tweets'}
     }

iplot(fig)
```



The above chart shows the top 4 applicaions which were used to used to send tweets in the dataset. The chart also shows what percentage of other applications were used to send tweets. As can be seen from the chart above the majority of tweets were sent from the "Twitter Web Client" as that was used to send 36.1% of tweets used in the set. The next two most popular applications used were mobile applications as "Twitter for iPhone" has 17.8% of the tweets and "Twitter for Android" having 16.5%. These figures show an interesting point as although the web application had overall more than all of the other applications, if you were to combine the two mobiles applications it would come out at 34.3% just under the 36.1% of the web application.

This shows that mobile applcations were infact very close to being the most used application for sending tweets in the set. It could also be agruged that mobile applications could have sent the most tweets as other mobile applications such as "Tweetbot for iOS" would be stored in the other section of the chart meaning that all considered it is no longer the case that traditional means such as web applications on computers will be the most used applications when it comes to social media.

Dates

In this section of the notebook it looks at how the number of tweets sent changed between the days in the dataset. The section looks at the number of tweets sent on each of the different days and then this is displayed in a bar graph. Finally this information is analysed to see what reasons there might be for some days having a higher number of tweets than others.

In [34]:

```
#Gets the dates used in the dataset
dfSource = df
import re
dates = []
for index, row in dfSource.iterrows():
    text = (row['created_at'])
    dates.append(text[0:10])
```

In [35]:

```
#Counts the number of tweets in each day
dateCount = {}
counter =0

for date in dates:
    if date not in dateCount:
        dateCount[date] = 1
    else:
        counter = dateCount.get(date,'none')
        dateCount.update({date: counter+1})
```

In [36]:

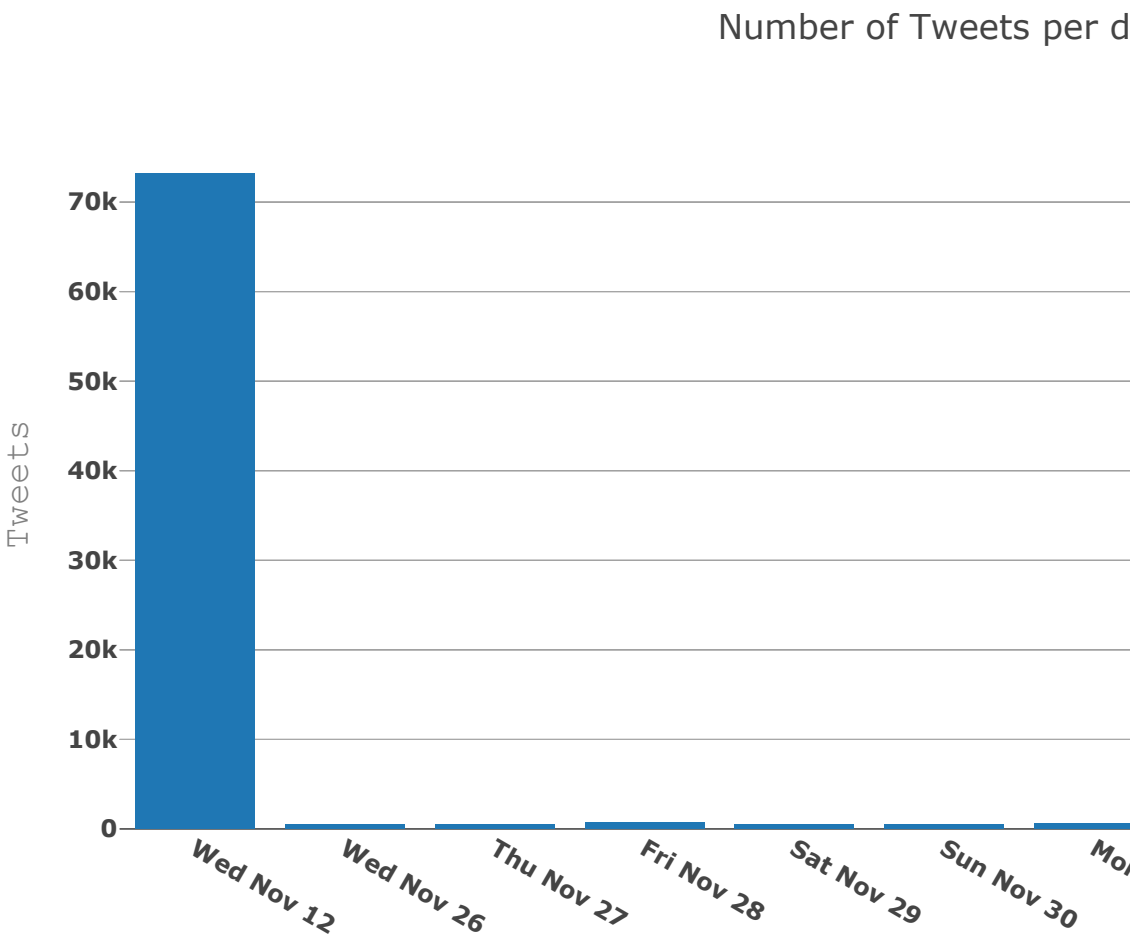
```
#Adds the dates and the number of tweets into arrays for use in the printing and
 in the grphs
values =[]
dateKeys = []

for key,val in dateCount.items():
    values.append(val)
    dateKeys.append(str(key))
    print("Date: "+ str(key) + " Number of Tweets: " + str(val))
```

```
Date: Fri Dec 05 Number of Tweets: 87
Date: Thu Dec 04 Number of Tweets: 200
Date: Wed Dec 03 Number of Tweets: 311
Date: Tue Dec 02 Number of Tweets: 475
Date: Mon Dec 01 Number of Tweets: 603
Date: Sun Nov 30 Number of Tweets: 343
Date: Sat Nov 29 Number of Tweets: 428
Date: Fri Nov 28 Number of Tweets: 711
Date: Thu Nov 27 Number of Tweets: 497
Date: Wed Nov 26 Number of Tweets: 400
Date: Wed Nov 12 Number of Tweets: 73212
```

Analysing patterens of user activity (Medium Extension 1)

In [37]:

```
#Sorts the dates and values and then displays them in a bar graph
dateKeys.reverse()
values.reverse()
data = [go.Bar(x=dateKeys,y=values)]
layout = go.Layout(
    title='Number of Tweets per day',
    yaxis=dict(
        title='Tweets',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    )
)

fig = go.Figure(data=data, layout=layout)
iplot(fig, filename='basic-bar')
```



The above graph shows the number of tweets which were sent on each day in the dataset. As can be seen from the graph the dataset contained a majority of tweets sent on 12th November, with 73 thousand tweets in the set being sent on that day. This would be some what expected as on that day was the lander reached its comet destination. As this was a day which had been greatly anticipated for nearly a decade it would be expected for the dataset to reflect this with the number of tweets.

The graph also shows us that on 1st December there was a slight increase in the number of tweets over previous days about the Comet landing. One of the resons for this could be that on that day some of the first images from the comet were released which may have then caused people to begining tweeting about the pictures. However this increase is only a fraction to the amount of tweets which were sent on the actual comet landing.

Observations

There are many observations which can be made about each of the sections of the dataset. It is clear from the analysis on the dataset that the Comet Landing created a lot of interest from users on twitter. This is shown by the mass spike in tweets on the 12th Novemeber (the day of the launch) and then this is also displayed on the 1st December where there is another slight spike in the number of tweets due to pictures being realsead from the probe. Hashtags also had an important part in the dataset with many users choosing to use #CometLanding to discuss the landing.

Another interesting observation which can be made about the applications which were used to send tweets. As the above information shows, the most used application was the "Twitter Web app" however the next two most used applications were mobile applications. Although the web client may have been the most used application for this set of data if the total number of mobile applications from both iOS and Android were added together it would most likely surpass the amount sent from the web application. This highlights the continuing trend more generally that mobile applications are or have even overtaken their PC counterparts. As more and more people continue to use their mobile devices for these kinda of tasks this different in use will only continue. For example if this twitter set was to be created in 2018 it would be assumed that the vast majority of tweets would have been sent from mobile applications.

Languages used to send the tweets was another area in which some interetsing points arose. It was expected that english was the most used language with it being used so widly in Europe and with how Americans are interested in space exploration. It was more the next most languages which were interesting with Spanish taking up nearly 10% of the tweets in the dataset.

Provanence

The majority of the code for this practical was implemented by use and was not taken from external sources. One aspect of the code which was adapted was the graph creation with plotly. When creating the pie-charts and bar graphs this was adapted from the tutorials on the plotly website. Another tutorial which was followed was for the creation of the word cloud. In order to create the word cloud a tutorial was followed on the word cloud for python website.

Conclusion

Overall for this project the Twitter dataset provided was examined and analysed. The notebook displays the most used hashtags, the languages which were used for the tweets and the types of applications which were used to send each of the tweets. Each of the findings in the notebook have been analysed and possible reasons for the findings have been discussed. If more time was to be dedicated for this project, it would be helpful to see how these findings change on a dataset with a larger number of tweets or on a more recent dataset.

References

[1]- https://plot.ly/ (https://plot.ly/) For graph creation [2]- https://github.com/amueller/word_cloud (https://github.com/amueller/word_cloud) For word cloud creation