Database Course Homework 7: Trump's Tweets

In this homework, I will figure out the following 2 questions:

1. On what kind of device (iPhone or an Andriod phone, etc.) were these tweets sent?
2. On what time were these tweets sent?

by Fenglei Gu fg1121

> Data Scource: Baumgartner, Jason, 2019, "realdonaldtrump.ndjson", *Twitter Tweets for Donald J. Trump (@realdonaldtrump)*, https://doi.org/10.7910/DVN/KJEBIL/ADNH3U (https://doi.org/10.7910/DVN/KJEBIL/ADNH3U), Harvard Dataverse, V1

Preparation Works

1: Basic configurations

In [1]:
```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('fivethirtyeight')
import seaborn as sns
sns.set()
sns.set_context("talk")
import re
```

2: Load Data

The original dataset is about 110 MB, please be patient

In [2]:
```
def load_tweets(path):
    with open(path,'r',encoding="utf-8") as f:
        import ndjson
        return ndjson.load(f)
```

In [3]:
```
file_path='realdonaldtrump.ndjson'
trump_tweets_all = load_tweets(file_path)
```

In [4]:
```
trump_tweets_all[0].keys()
```

Out[4]:
```
dict_keys(['contributors', 'coordinates', 'created_at', 'entities', 'favorite
_count', 'favorited', 'geo', 'id', 'id_str', 'in_reply_to_screen_name', 'in_r
eply_to_status_id', 'in_reply_to_status_id_str', 'in_reply_to_user_id', 'in_r
eply_to_user_id_str', 'is_quote_status', 'lang', 'place', 'retrieved_utc', 'r
etweet_count', 'retweeted', 'source', 'text', 'truncated', 'user'])
```

3: Transform data into Pandas Dataframe

Note that I am only interested Trump's tweets since 2016 (under universal time zone).

Still, the dataset is big, so it might take some time

In [5]:
```
idx=[]
dt={'time':[],'source':[]}
for twt in trump_tweets_all:
    if eval(twt['created_at'][-4:]) < 2016:
        continue
    else:
        idx.append(twt['id'])
        dt['time'].append(pd.to_datetime(twt['created_at']))
        dt['source'].append(twt.get('source',''))
trump = pd.DataFrame(data=dt,index=idx)
trump.head()
```

Out[5]:

	time	source
682723973449289728	2016-01-01 00:44:14+00:00	<a href="http://twitter.com/download/android" ...
682764544402440192	2016-01-01 03:25:27+00:00	<a href="http://twitter.com/download/iphone" r...
682792967736848385	2016-01-01 05:18:23+00:00	<a href="http://twitter.com/download/iphone" r...
682805320217980929	2016-01-01 06:07:28+00:00	<a href="http://twitter.com/download/iphone" r...
682805477168779264	2016-01-01 06:08:06+00:00	<a href="http://twitter.com/download/android" ...

4: Data Cleaning

1. Remove the HTML tags in `source`
2. Transform UTC to Eastern Standard Time. The two entries showing EST in 2015 are actually in 2016 under UTC.
3. Create continuous parameter `year`.

In [6]:
```
r = lambda x : x.group(0).split("</")[-2].split('>')[-1]
trump['source'] = trump['source'].str.replace(r'<a.*</a>',repl=r)
```

In [7]:
```
import datetime

def year_fraction(date):
    start = datetime.date(date.year, 1, 1).toordinal()
    year_length = datetime.date(date.year+1, 1, 1).toordinal() - start
    return date.year + float(date.toordinal() - start) / year_length

trump['est'] = (trump['time'].dt.tz_convert("EST"))
trump['hour'] = trump['est'].apply(lambda x : (x.hour + x.minute /60 + x.secon
d /3600))
trump['year'] = trump['time'].apply(year_fraction)
trump.head()
```

Out[7]:

	time	source	est	hour	year
682723973449289728	2016-01-01 00:44:14+00:00	Twitter for Android	2015-12-31 19:44:14-05:00	19.737222	2016.0
682764544402440192	2016-01-01 03:25:27+00:00	Twitter for iPhone	2015-12-31 22:25:27-05:00	22.424167	2016.0
682792967736848385	2016-01-01 05:18:23+00:00	Twitter for iPhone	2016-01-01 00:18:23-05:00	0.306389	2016.0
682805320217980929	2016-01-01 06:07:28+00:00	Twitter for iPhone	2016-01-01 01:07:28-05:00	1.124444	2016.0
682805477168779264	2016-01-01 06:08:06+00:00	Twitter for Android	2016-01-01 01:08:06-05:00	1.135000	2016.0

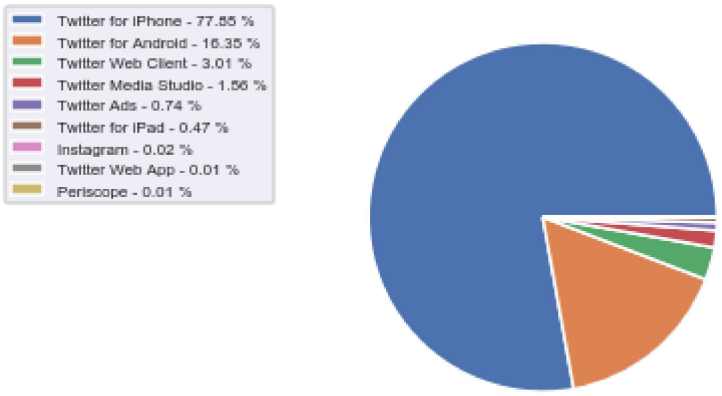
What kind of devices were used?

In [8]:
```
# Source: https://doi.org/10.7910/DVN/KJEBIL/ADNH3U

source = trump['source'].value_counts()

x = source.index.tolist()
y = source.values
percent = 100.*y/y.sum()

patches, texts = plt.pie(y)
labels = ['{0} - {1:1.2f} %'.format(i,j) for i,j in zip(x, percent)]
patches, labels, dummy =  zip(*sorted(zip(patches, labels, y),key=lambda x: x[
2],reverse=True))
plt.legend(patches, labels, bbox_to_anchor=(-0.1, 1.),fontsize=8)
plt.show()
```

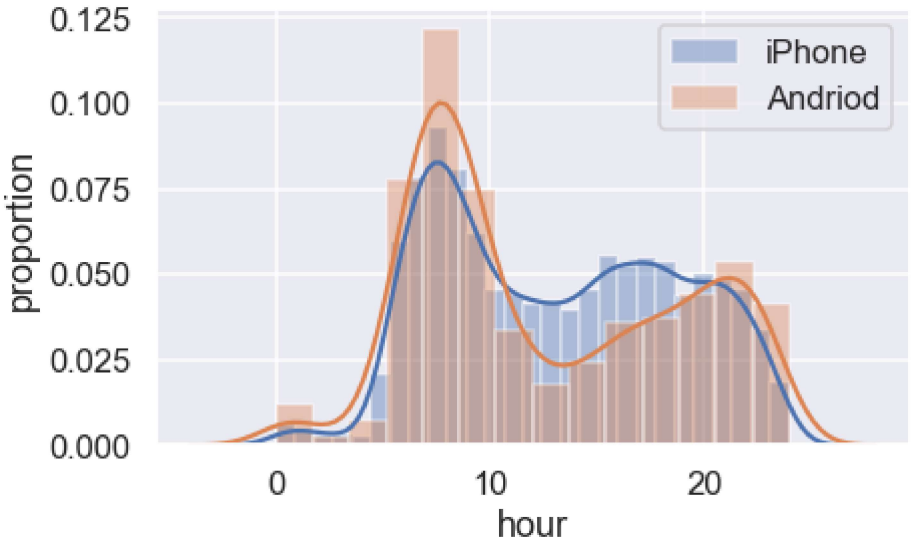


The pie chart above shows that most of Trumps' tweets were sent on iPhone, and a significant proportion were sent on an Android device, whereas there are also several other platforms. This pie chart might be useful for those intersted in Trump's behavior.

Since just a few of his tweets were not sent through iPhone nor Android, we'll focuse on iOS and Android below.

At what time of a day were Trump's tweets sent?

In [9]:
```
# Source: https://doi.org/10.7910/DVN/KJEBIL/ADNH3U
sns.distplot(trump[trump['source']=='Twitter for iPhone']['hour'], label = 'iPhone')
sns.distplot(trump[trump['source']=='Twitter for Android']['hour'], label = 'Andriod')
plt.xlabel('hour')
plt.ylabel('proportion')
plt.legend()
plt.show()
```



It indicates that very few of Trump's tweets sent during day, especially in the afternoon, were sent by an Android device. This plot may be helpful for researches on Trump's tweeting routine.

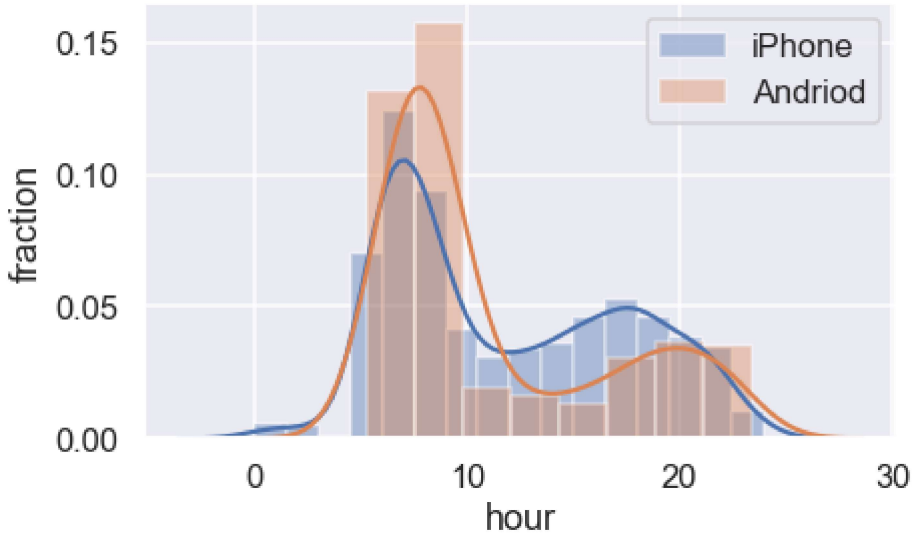
with interactive time

Above are the result of all data since 2016. Now, it's time for the user to input a year (2016,2017,2018,2019) to see the result within that year!

In [1]:
```
while True:
    s=input('Enter the year you want:')
    try:
        yr=eval(s)
    except:
        print('Not a number!')
        continue
    else:
        if type(yr)==type(2016):
            if ( yr<2016 or yr>2020 ):
                print('Not in 2016~2019!')
            else:
                break
        else:
            print('Not a valid year number!')
print('You selected Year {}!'.format(yr))
```

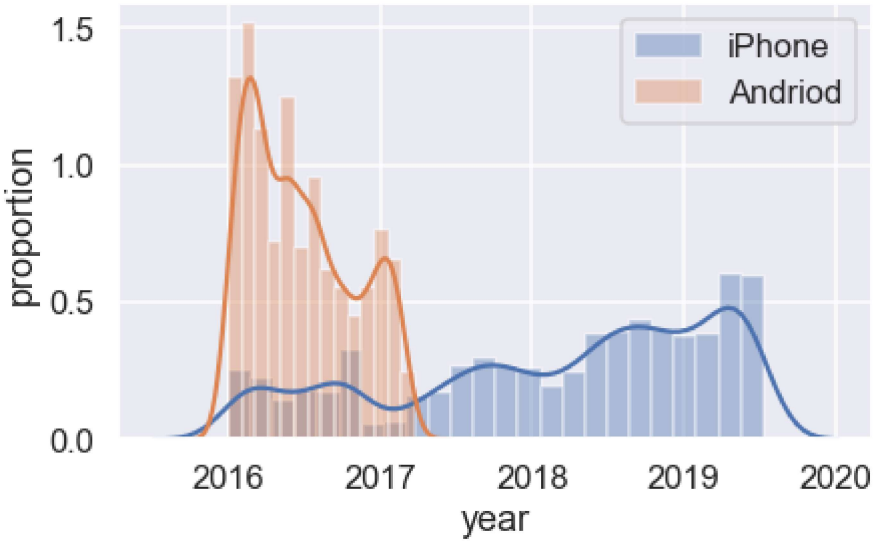
```
Enter the year you want:2017.0
Not a valid year number!
Enter the year you want:'17
Not a number!
Enter the year you want:2021
Not in 2016~2019!
Enter the year you want:2017
You selected Year 2017!
```

In [14]:
```
# Source: https://doi.org/10.7910/DVN/KJEBIL/ADNH3U
trump2 = trump[(trump['year'] >= yr) & (trump['year'] < yr+1)]
sns.distplot(trump2[trump2['source']=='Twitter for iPhone']['hour'], label =
'iPhone')
sns.distplot(trump2[trump2['source']=='Twitter for Android']['hour'], label =
'Andriod')
plt.xlabel('hour')
plt.ylabel('fraction')
plt.legend()
plt.show()
```



What about distribution on a long scale of time?

In [12]:
```
# Source: https://doi.org/10.7910/DVN/KJEBIL/ADNH3U
sns.distplot(trump[trump['source']=='Twitter for iPhone']['year'], label = 'iP
hone')
sns.distplot(trump[trump['source']=='Twitter for Android']['year'], label = 'A
ndriod')
plt.xlabel('year')
plt.ylabel('proportion')
plt.legend()
plt.show()
```



The plot indicates that Trump used to be using an Android device, yet has switched to an iPhone.

In [ ]:
```
```