Notebook

March 2, 2020

0.1 Question 0

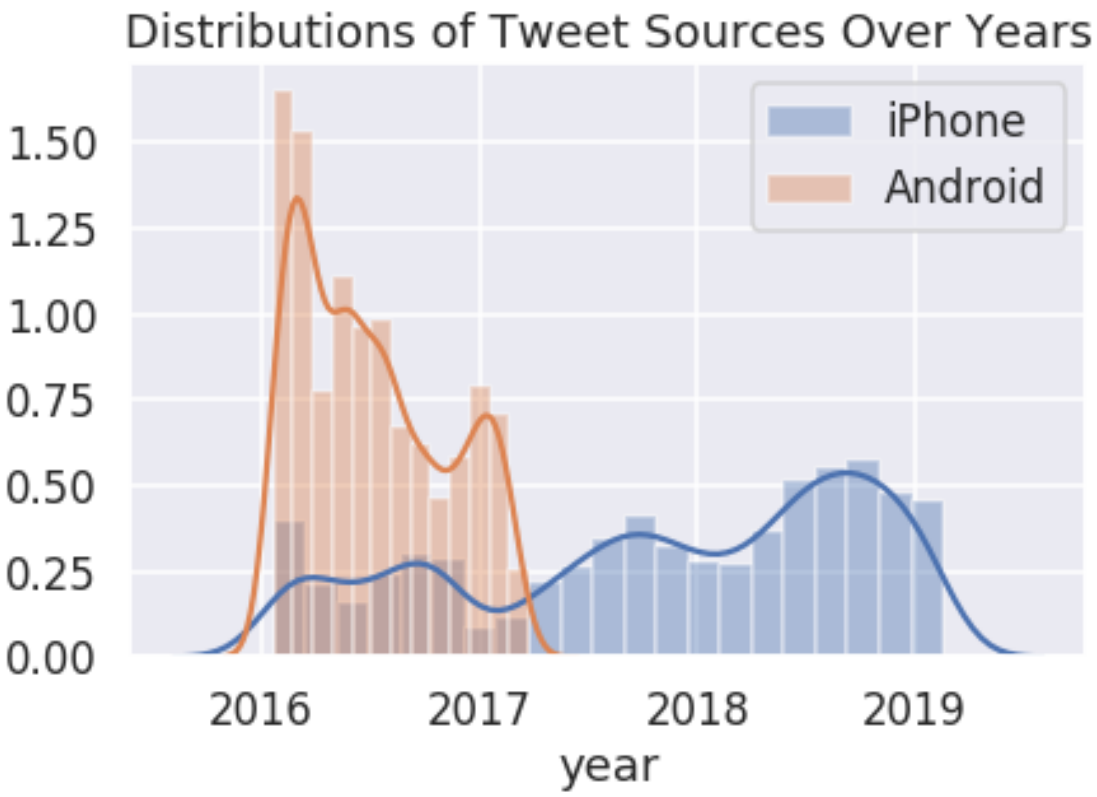
Why might someone be interested in doing data analysis on the President's tweets? Name one person or entity which might be interested in this kind of analysis. Then, give two reasons why a data analysis of the President's tweets might be interesting or useful for them. Answer in 2-3 sentences.

A person who is interested in the stock market might be interested in this kind of analysis because tweets from the President may have implicit effects on the rise or fall of stock prices. Given the scenario that the President tweets insults or anything to the Chinese government that may hurt the U.S.'s relations with China, stocks in Chinese brands or companies may fall because of it. Doing data analysis on the President's tweets may give these people more insight on how the President affects the stock market.

Now, use `sns.distplot` to overlay the distributions of Trump's 2 most frequently used web technologies over the years. Your final plot should look similar to the plot below:

```
In [178]: iphone = trump[trump['source'] == 'Twitter for iPhone']
          android = trump[trump['source'] == 'Twitter for Android']
          sns.distplot(iphone[['year']], label = 'iPhone')
          sns.distplot(android[['year']], label = 'Android')
          plt.xlabel('year')
          plt.title('Distributions of Tweet Sources Over Years')
          plt.legend()

Out[178]: <matplotlib.legend.Legend at 0x7fb4505b7710>
```

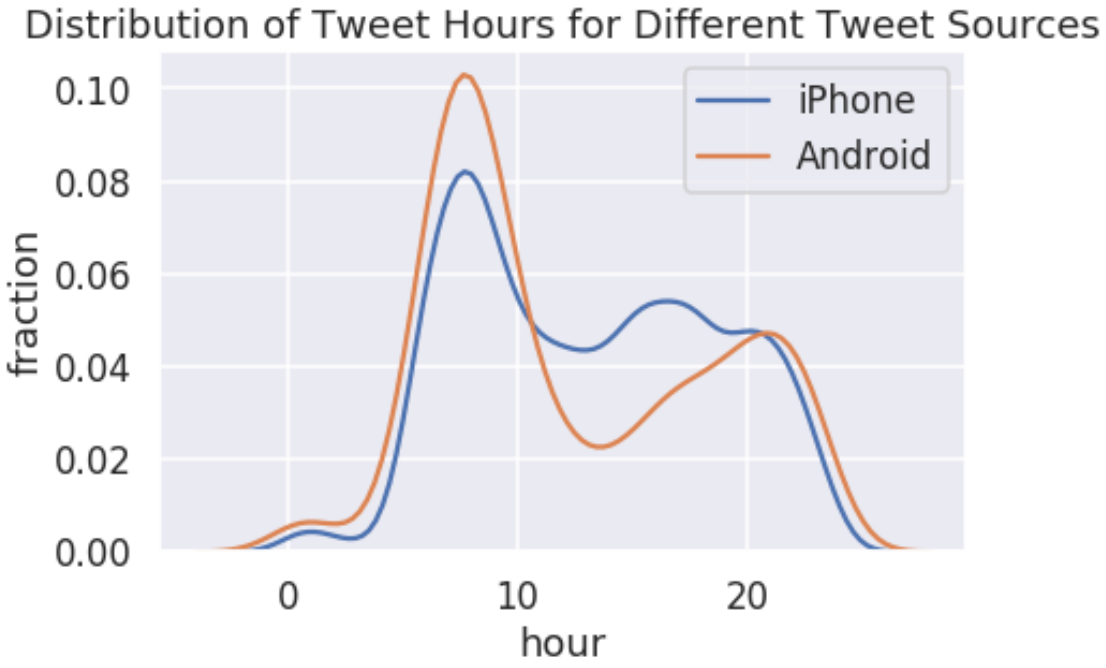


0.1.1 Question 4b

Use this data along with the seaborn `distplot` function to examine the distribution over hours of the day in eastern time that Trump tweets on each device for the 2 most commonly used devices. Your final plot should look similar to the following:

```
In [199]: ### make your plot here
          iphone = trump[trump['source'] == 'Twitter for iPhone']
          android = trump[trump['source'] == 'Twitter for Android']
          sns.distplot(iphone[['hour']], hist = False, label = 'iPhone')
          sns.distplot(android[['hour']], hist = False, label = 'Android')
          plt.xlabel('hour')
          plt.ylabel('fraction')
          plt.title('Distribution of Tweet Hours for Different Tweet Sources')
          plt.legend()

Out[199]: <matplotlib.legend.Legend at 0x7fb44a3d9ba8>
```



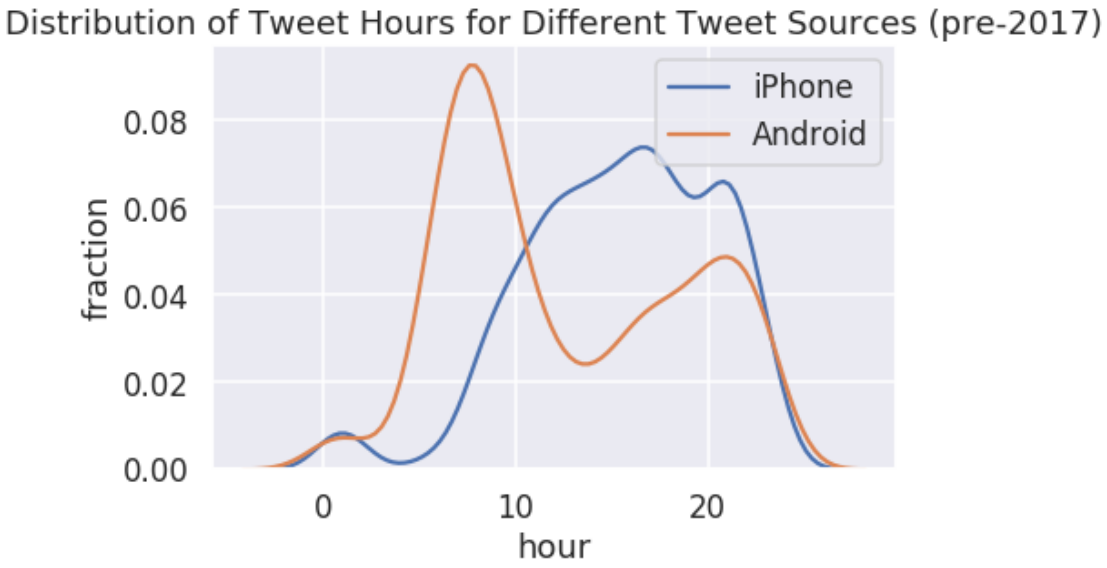
0.1.2 Question 4c

According to this Verge article, Donald Trump switched from an Android to an iPhone sometime in March 2017.

Let's see if this information significantly changes our plot. Create a figure similar to your figure from question 4b, but this time, only use tweets that were tweeted before 2017. Your plot should look similar to the following:

```
In [203]: ### make your plot here
          iphone = trump[trump['source'] == 'Twitter for iPhone']
          iphone = iphone[iphone['year'] < 2017]
          android = trump[trump['source'] == 'Twitter for Android']
          android = android[android['year'] < 2017]
          sns.distplot(iphone[['hour']], hist = False, label = 'iPhone')
          sns.distplot(android[['hour']], hist = False, label = 'Android')
          plt.xlabel('hour')
          plt.ylabel('fraction')
          plt.title('Distribution of Tweet Hours for Different Tweet Sources (pre-2017)')
          plt.legend()

Out[203]: <matplotlib.legend.Legend at 0x7fb450313cc0>
```



0.1.3 Question 4d

During the campaign, it was theorized that Donald Trump's tweets from Android devices were written by him personally, and the tweets from iPhones were from his staff. Does your figure give support to this theory? What kinds of additional analysis could help support or reject this claim?

This figure does support this theory because this shows that Trump was only active very early in the morning and late at night. Given that it was during the campaign, it makes sense that he was not posting much in the day, and having his staff curate tweets for him that may improve his standings with the people.

0.2 Question 5

The creators of VADER describe the tool's assessment of polarity, or "compound score," in the following way:

"The compound score is computed by summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive). This is the most useful metric if you want a single unidimensional measure of sentiment for a given sentence. Calling it a 'normalized, weighted composite score' is accurate."

As you can see, VADER doesn't "read" sentences, but works by parsing sentences into words assigning a preset generalized score from their testing sets to each word separately.

VADER relies on humans to stabilize its scoring. The creators use Amazon Mechanical Turk, a crowdsourcing survey platform, to train its model. Its training set of data consists of a small corpus of tweets, New York Times editorials and news articles, Rotten Tomatoes reviews, and Amazon product reviews, tokenized using the natural language toolkit (NLTK). Each word in each dataset was reviewed and rated by at least 20 trained individuals who had signed up to work on these tasks through Mechanical Turk.

0.2.1 Question 5a

Given the above information about how VADER works, name one advantage and one disadvantage of using VADER in our analysis.

One advantage of using VADER is that it will be quite efficient in analyzing tweets from social media which mainly uses words with more extreme valence. One disadvantage is that since it is not reading the entire sentence, but rather by each word, sentences that have a positive valence in its total meaning may have a negative valence as a result.

0.2.2 Question 5b

Are there circumstances (e.g. certain kinds of language or data) when you might not want to use VADER? Please answer "Yes," or "No," and provide 1 reason for your answer.

Yes, VADER will not be useful for foreign languages and data that are purely numerical.

0.3 Question 5h

Read the 5 most positive and 5 most negative tweets. Do you think these tweets are accurately represented by their polarity scores?

Yes I think these tweets accurately represent their polarity scores.

0.4 Question 6

Now, let's try looking at the distributions of sentiments for tweets containing certain keywords.

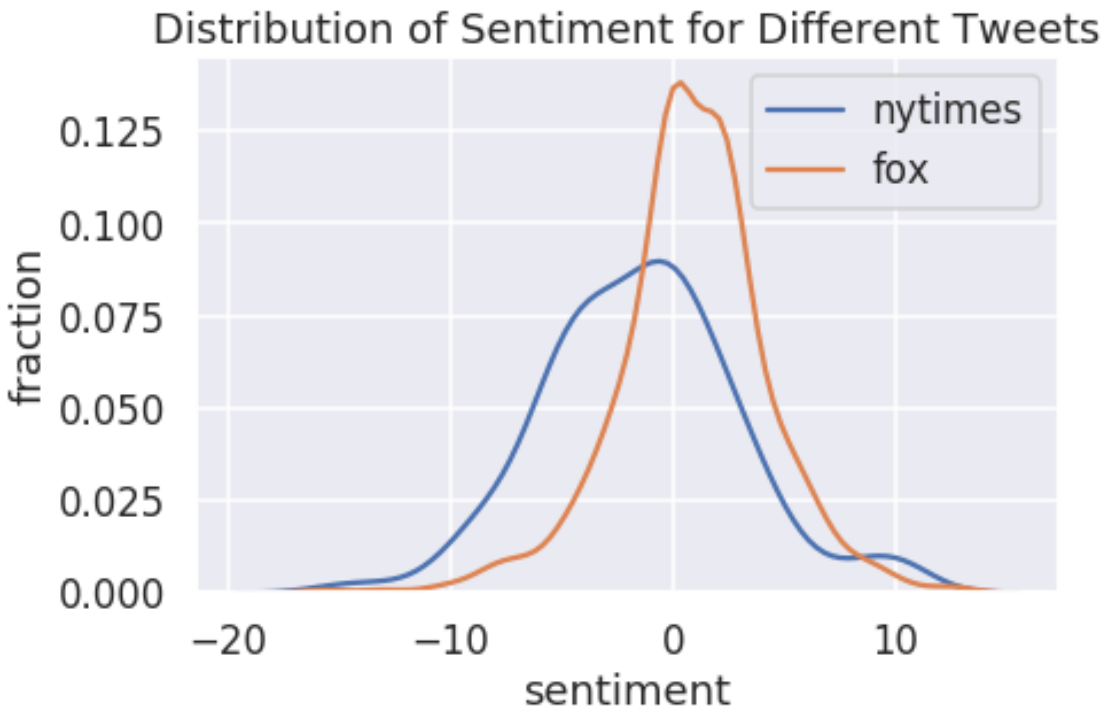
0.4.1 Question 6a

In the cell below, create a single plot showing both the distribution of tweet sentiments for tweets containing `nytimes`, as well as the distribution of tweet sentiments for tweets containing `fox`.

Be sure to label your axes and provide a title and legend. Be sure to use different colors for `fox` and `nytimes`.

```
In [237]: nyt = trump[trump['text'].str.lower().str.contains('nyt')]
          fox = trump[trump['text'].str.lower().str.contains('fox')]
          sns.distplot(nyt[['polarity']], hist = False, label = 'nytimes')
          sns.distplot(fox[['polarity']], hist = False, label = 'fox')
          plt.xlabel('sentiment')
          plt.ylabel('fraction')
          plt.title('Distribution of Sentiment for Different Tweets')
          plt.legend()
```

```
Out[237]: <matplotlib.legend.Legend at 0x7fb450957ef0>
```



0.4.2 Question 6b

Comment on what you observe in the plot above. Can you find another pair of keywords that lead to interesting plots? Describe what makes the plots interesting. (If you modify your code in 6a, remember to change the words back to `nytimes` and `fox` before submitting for grading).

What makes this plot interesting is how much more skewed to the right tweets about fox are compared to nytimes. Trump's tweets about fox has a much higher chance of having a positive valence than nytimes.

What do you notice about the distributions? Answer in 1-2 sentences.

From this distribution, I noticed that tweets with hastags or links generally have more positive sentiment compared to tweets without hashtags or links. Tweets without hastags or links are generally more spread out across -10 to 10 in terms of polarity.