0.1 Question 0

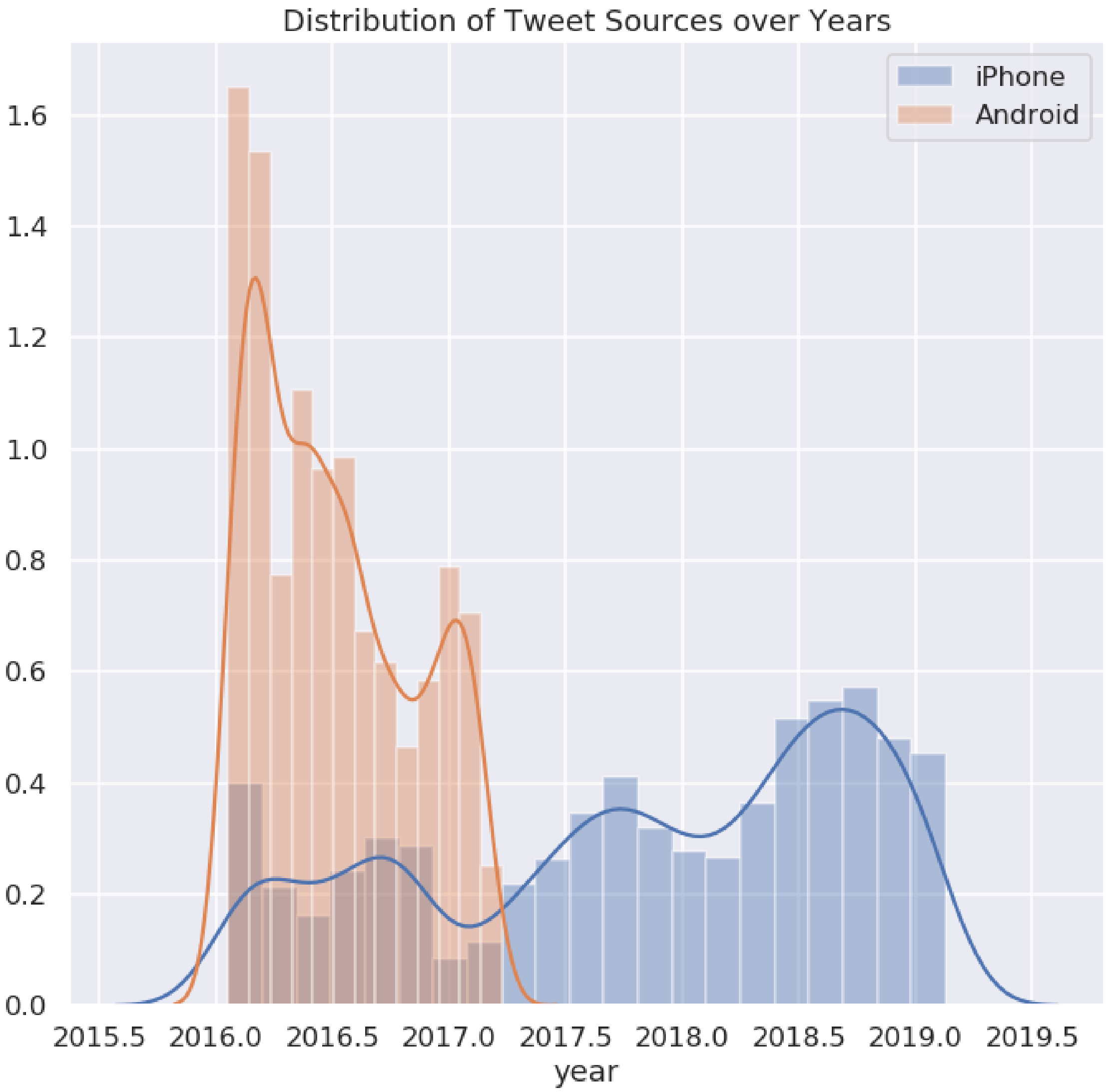
Why might someone be interested in doing data analysis on the President's tweets? Name one person or entity which might be interested in this kind of analysis. Then, give two reasons why a data analysis of the President's tweets might be interesting or useful for them. Answer in 2-3 sentences.

If one had access to a database of the US government's actions, it could be possible to correlate Trump's tweeting to certain government actions, enabling foreign state actors to predict US policy movements. In addition, it could be possible to correlate Trump's tweets with his mental state, giving the same actors the ability to understand his state of mind before negotiations.

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 [14]: plt.figure(figsize=(10, 10))
         sns.distplot(trump.loc[trump["source"].str.contains("iPhone"), "year"], label="iPhone")
         sns.distplot(trump.loc[trump["source"].str.contains("Android"), "year"], label="Android")
         plt.xlabel="year"
         plt.ylabel="proportion"
         plt.title("Distribution of Tweet Sources over Years")
         plt.legend()
```

```
Out[14]: <matplotlib.legend.Legend at 0x7f55dd2ec6d0>
```

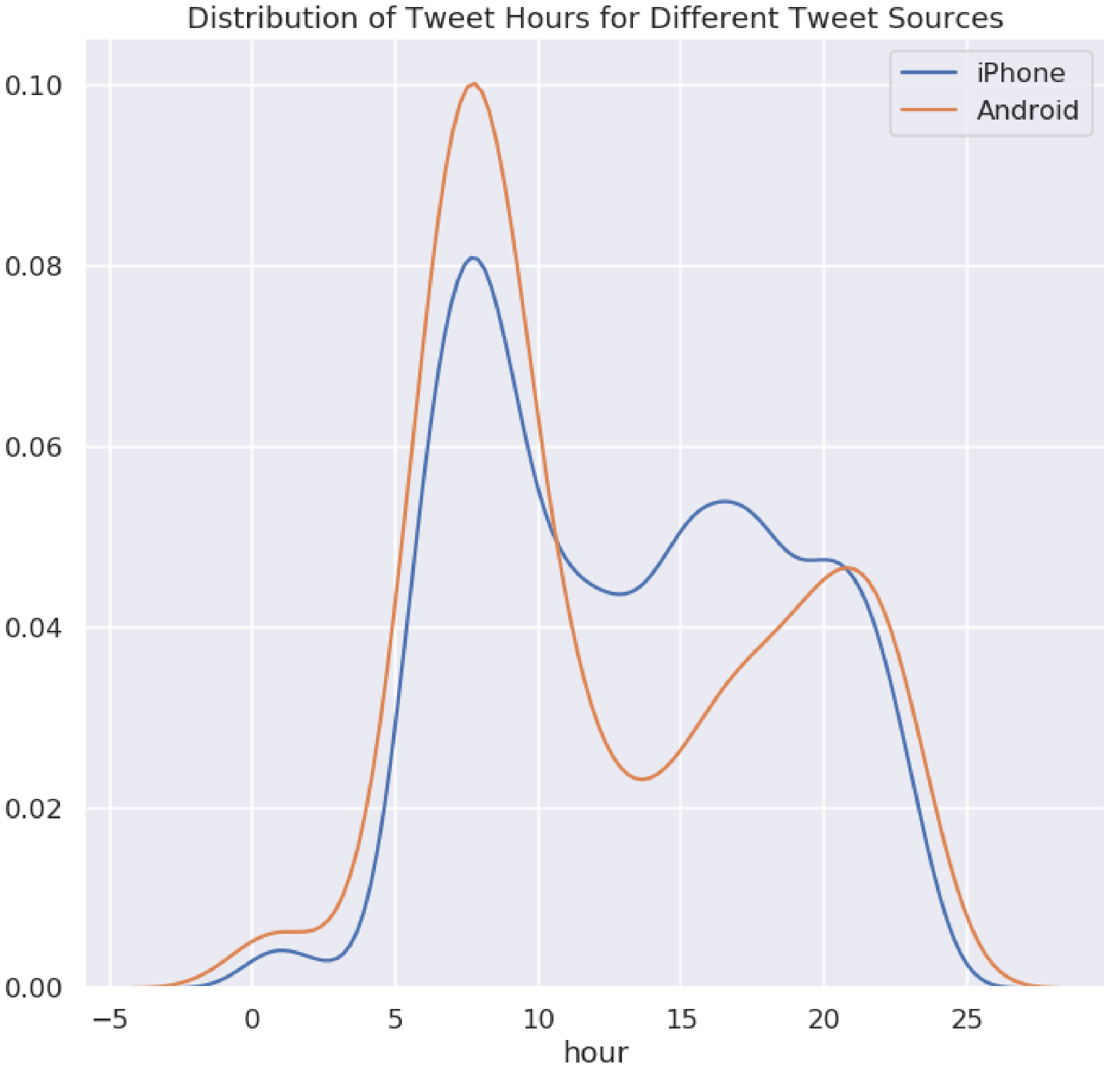


0.1.1 Question 4b

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 [19]: ### make your plot here
         plt.figure(figsize=(10, 10))
         sns.distplot(trump.loc[trump["source"].str.contains("iPhone"), "hour"],
                      hist=False, kde=True, label="iPhone")
         sns.distplot(trump.loc[trump["source"].str.contains("Android"), "hour"],
                      hist=False, kde=True, label="Android")
         plt.xlabel="hour"
         plt.ylabel="fraction"
         plt.title("Distribution of Tweet Hours for Different Tweet Sources")
```

```
Out[19]: Text(0.5, 1.0, 'Distribution of Tweet Hours for Different Tweet Sources')
```


Distribution of Tweet Hours for Different Tweet Sources
iPhone
Android
0.10
0.08
0.06
0.04
0.02
0.00
−5
0
5
10
15
20
25
hour

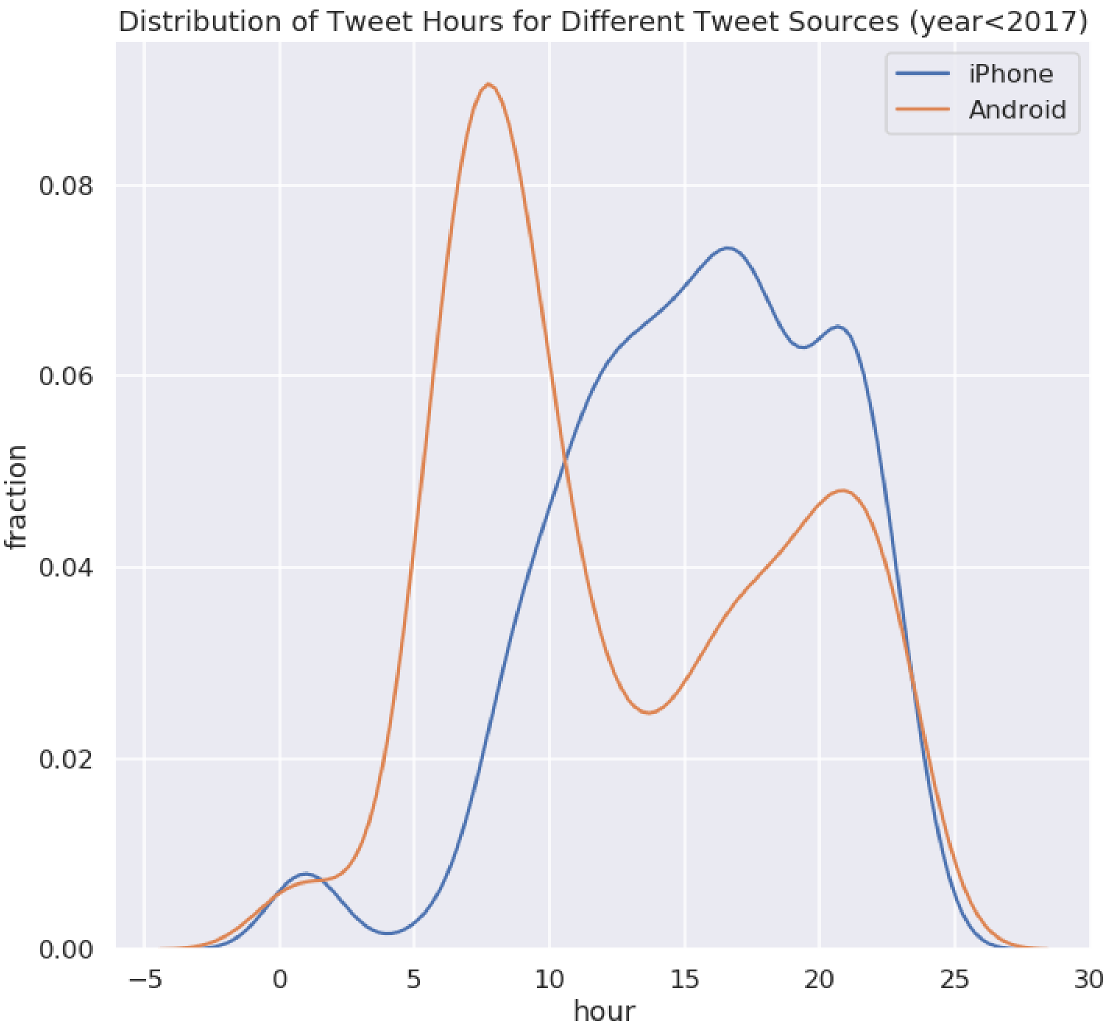
0.1.2 Question 4c

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 [20]: ### make your plot here
         plt.figure(figsize=(10, 10))
         sns.distplot(trump.loc[(trump["source"].str.contains("iPhone"))
                                 & (trump["year"] < 2017), "hour"],
                      hist=False, kde=True, label="iPhone")
         sns.distplot(trump.loc[(trump["source"].str.contains("Android"))
                                 & (trump["year"] < 2017), "hour"],
                      hist=False, kde=True, label="Android").set(ylabel="fraction")
         plt.title("Distribution of Tweet Hours for Different Tweet Sources (year<2017)")
```

```
Out[20]: Text(0.5, 1.0, 'Distribution of Tweet Hours for Different Tweet Sources (year<2017)')
```


Distribution of Tweet Hours for Different Tweet Sources (year<2017)
iPhone
Android
0.08
0.06
0.04
0.02
0.00
fraction
−5
0
5
10
15
20
25
30
hour

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?

Given that the tweets from Android prior to 2017 occur earlier in the morning, it makes sense that they would be from Trump as opposed to his campaign staff, who would likely not be working. To additionally investigate this claim, lexical/semantic analysis could be done on the tweet content to see if there's a statistically significant difference between the content of the iPhone tweets and the Android tweets.

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.

VADER seems like a very straightforwards and simple ranking tool, and obtaining exploratory information about sentiment using it seems valuable. However, VADER is not context-aware, because it ranks individual words and not sentences as a whole, meaning it could lose out on pragmatic implicatures that are not immediately present in the written text. With that said, Trump's language seems to be simple and straightforwards to parse since he does not make excessive use of idiomatic prose, making VADER more valuable.

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: text that is heavily prosaic and context-heavy would not be easily parseable by VADER; for instance, using positive statements in a sarcastic manner would results in a high VADER score, but it would pragmatically imply a negative sentiment. In addition, it would be possible to skew VADER scores with infelicitous language, that is, a slew of positive words in a nonsensical order may get a high score, when in fact analysis on such a string would be meaningless.

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?

It does seem so: the negative tweets express on-brand Trump disapproval with sacked officials, immigrants, and China, while the positive tweets express approval about his followers or mundane football games.

0.4 Question 6

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

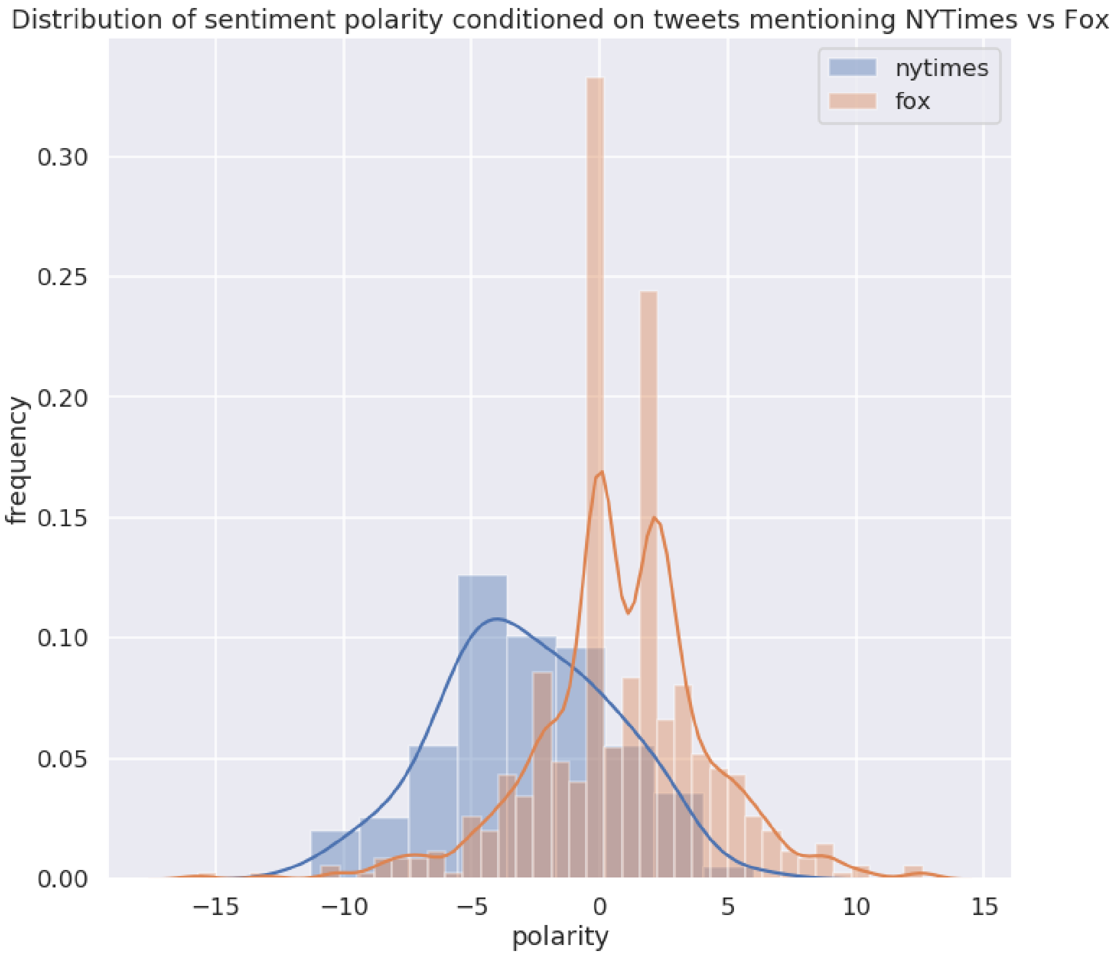
0.4.1 Question 6a

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 [35]: plt.figure(figsize=(10, 10))
         sns.distplot(
             trump.loc[trump["text"].str.contains("nytimes")]["polarity"],
             label="nytimes")
         sns.distplot(
             trump.loc[trump["text"].str.contains("fox")]["polarity"],
             label="fox").set(ylabel="frequency")
         plt.title("Distribution of sentiment polarity conditioned on tweets mentioning NYTimes vs Fox"
         plt.legend()
```

```
Out[35]: <matplotlib.legend.Legend at 0x7f55e3295d50>
```



```
In [36]: from scipy.stats import ttest_ind

         tstat, pvalue = ttest_ind(
             trump.loc[trump["text"].str.contains("nytimes")]["polarity"],
             trump.loc[trump["text"].str.contains("fox")]["polarity"],
             equal_var=False
         )

         print(f"t-statistic:\t{tstat}\np-value:\t{pvalue}\
             \nRejecting H_0 that the two distributions are equal.")
```

```
t-statistic:        -9.967093526312498
p-value:        3.2209745301693646e-18
Rejecting H_0 that the two distributions are equal.
```

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).

The Fox distribution appears bimodal and heavily centred, whilst the NYT distribution is unimodal and relatively flat. Clearly the distribution of sentiment for tweets mentioning the NYT has a mean to the left of that of the distribution of tweets mentioning Fox. More interestingly, the tweets mentioning Fox are *more overwhelmingly positive*: the mode of the Fox distribution is not only higher in polarity thatn that of the NYT distribution, but there are also more values clustered around the mode. This manifests as a higher peak in the KDE plot as well as higher bars on the histogram.

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

The sentiment distribution for tweets without hashtags and links is *far* wider than that for tweets with hashtags or links. In addition, the sentiment distribution for tweets *with* hashtags or links seems to be heavily bimodal.