

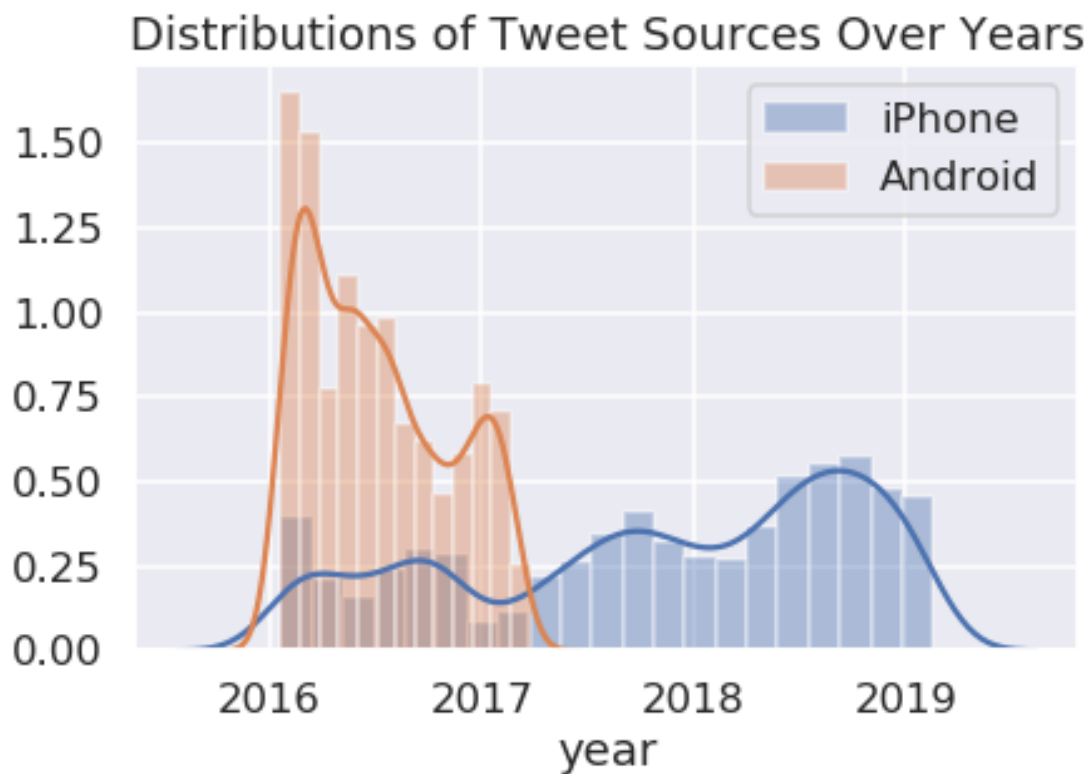
Notebook

February 25, 2019

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 like:

```
In [268]: android_source = trump.loc[trump['source'] == 'Twitter for Android']
         iphone_source = trump.loc[trump['source'] == 'Twitter for iPhone']
         _ = sns.distplot(iphone_source['year'])
         _ = sns.distplot(android_source['year'])
         _ = plt.legend(['iPhone', 'Android'])
         _ = plt.title('Distributions of Tweet Sources Over Years')
```

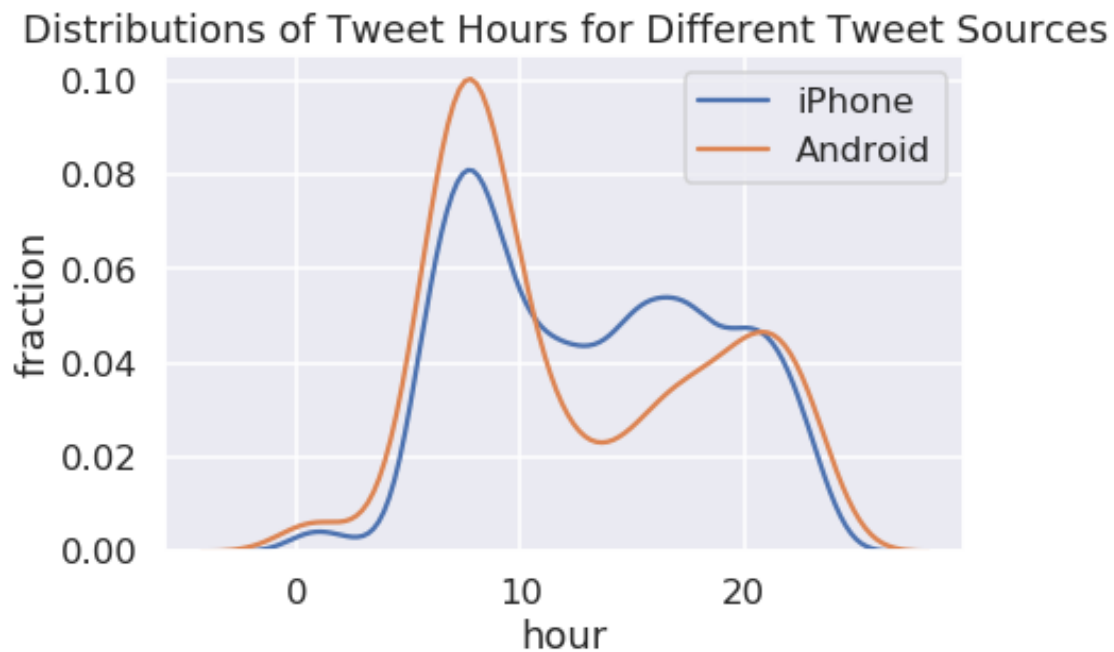
```
/srv/conda/envs/data100/lib/python3.6/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will result in a ValueError. Use the tuple form of indexing on new arrays.
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



0.0.1 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 plot should look similar to the following:

```
In [273]: ### make your plot here
android_hour = trump.loc[trump['source'] == 'Twitter for Android']
iphone_hour = trump.loc[trump['source'] == 'Twitter for iPhone']
_ = sns.distplot(iphone_hour['hour'], hist=False, label='iPhone')
_ = sns.distplot(android_hour['hour'], hist=False, label='Android')
_ = plt.legend()
_ = plt.ylabel('fraction')
_ = plt.title('Distributions of Tweet Hours for Different Tweet Sources')
```



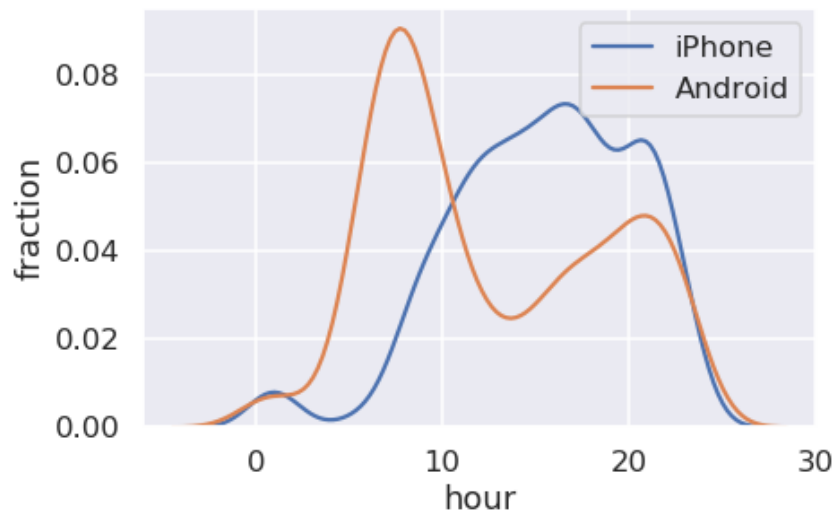
0.0.2 Question 4c

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 [274]: ### make your plot here
android_hour_2 = trump.loc[(trump['source'] == 'Twitter for Android') & (trump['year'] < 2017)]
iphone_hour_2 = trump.loc[(trump['source'] == 'Twitter for iPhone') & (trump['year'] < 2017)]
_ = sns.distplot(iphone_hour_2['hour'], hist=False, label='iPhone')
_ = sns.distplot(android_hour_2['hour'], hist=False, label='Android')
_ = plt.legend()
_ = plt.ylabel('fraction')
_ = plt.title('Distributions of Tweet Hours for Different Tweet Sources (pre-2017)')
```

Distributions of Tweet Hours for Different Tweet Sources (pre-2017)



0.0.3 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?

I believe the plot above supports this theory because there is a decrease in tweets from his Android during working hours, between 8am to 8pm, whereas there is an increase in tweets from his iPhone during the same time period. This observation reasonably supports the theory because staff members will likely be 'working' during the times they post tweets for Trump and Trump most likely has more time for personal tweets when he is away from work. In order to better support or reject this claim, we could, for instance, observe source behavior during holidays or weekends.

0.1 Question 6

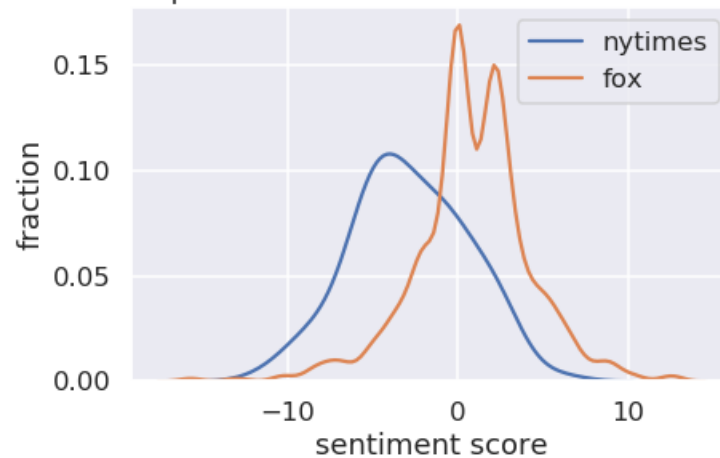
Now, let's try looking at the distributions of sentiments for tweets containing certain keywords.

0.1.1 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.

```
In [288]: nytimes = trump.loc[trump['no_punc'].str.contains('nytimes')][['polarity']]
          fox = trump.loc[trump['no_punc'].str.contains('fox')][['polarity']]
          _ = sns.distplot(nytimes, hist=False, label='nytimes')
          _ = sns.distplot(fox, hist=False, label='fox')
          _ = plt.ylabel('fraction')
          _ = plt.xlabel('sentiment score')
          _ = plt.title('Distribution of Trump\'s Tweets\' Sentiment Scores Containing Specific Words')
          _ = plt.legend()
```

Distribution of Trump's Tweets' Sentiment Scores Containing Specific Words



0.1.2 Question 6b

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

We're observing relatively positive sentiments (>0) in tweets containing the word 'fox' as opposed to negative sentiments (<0) in tweets contains the word 'nytimes'. It appears that the negative sentiment towards nytimes is stronger than is positive sentiment towards fox. A similar distribution can be observed using keywords such as 'immigrant' and 'america'.