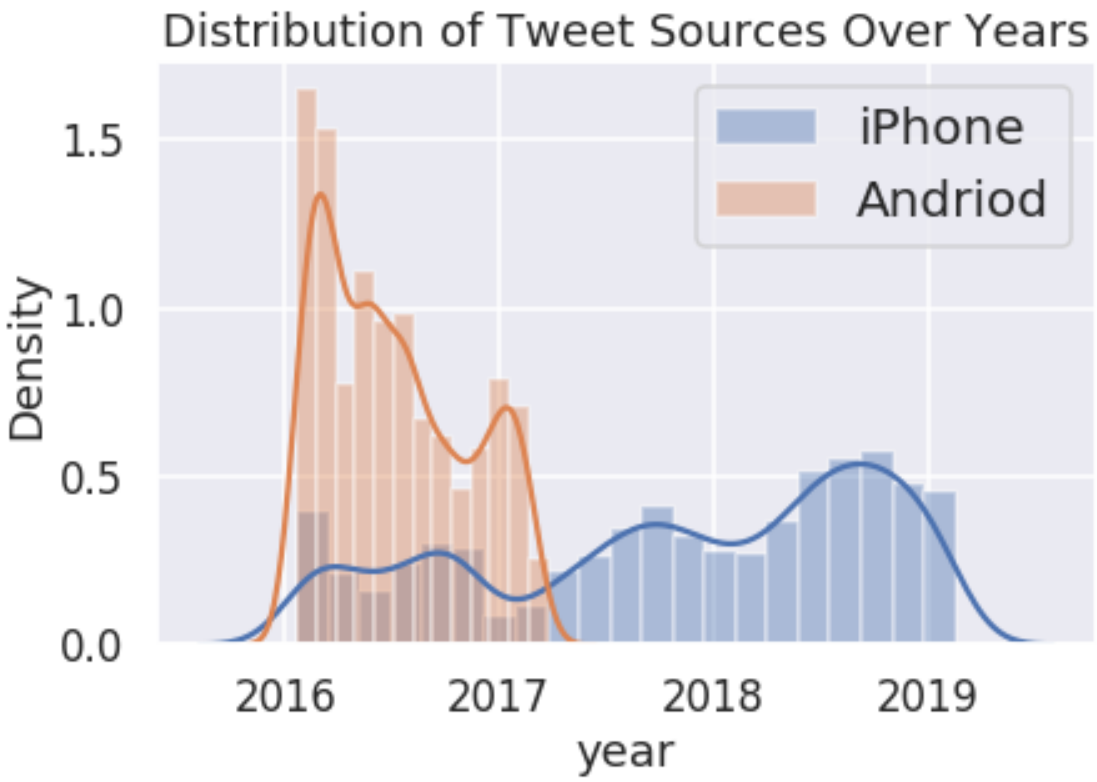
0.1 Question 0

There are many ways we could choose to read the President's tweets. Why might someone be interested in doing data analysis on the President's tweets? Name a kind of person or institution 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.

Someone might be interested in doing an analysis on Trump's tweets is to identify his word choice. As the president of the United States one might be interested in this because identifying certain words could be used to predict his behavior on policies or decisions. The media would be most interested in this as it would make for a good headline. Once they can identify and preform a meaningful analysis on his words, they can then communicate that to the public and everyone else interested in the president. This is especically helpful right now during a an election.

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:

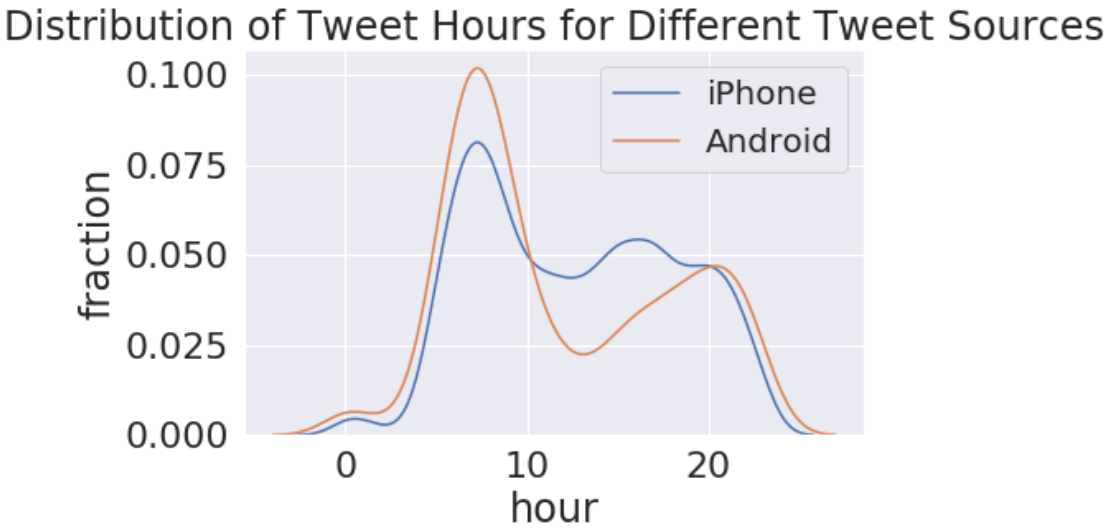
```
In [17]: iphone = trump[trump['source']=='Twitter for iPhone']['year']
         android = trump[trump['source']=='Twitter for Android']['year']
         sns.distplot(iphone, label='iPhone')
         sns.distplot(android, label='Andriod')
         plt.title('Distribution of Tweet Sources Over Years')
         plt.xlabel('year')
         plt.legend(prop={'size': 20})
         sns.set(font_scale =2)
```



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 [22]: ### make your plot here
         iphone_hour = trump[trump['source']=='Twitter for iPhone']['est_time'].dt.hour
         android_hour = trump[trump['source']=='Twitter for Android']['est_time'].dt.hour
         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(prop={'size':20})
         sns.set(font_scale=2);
```

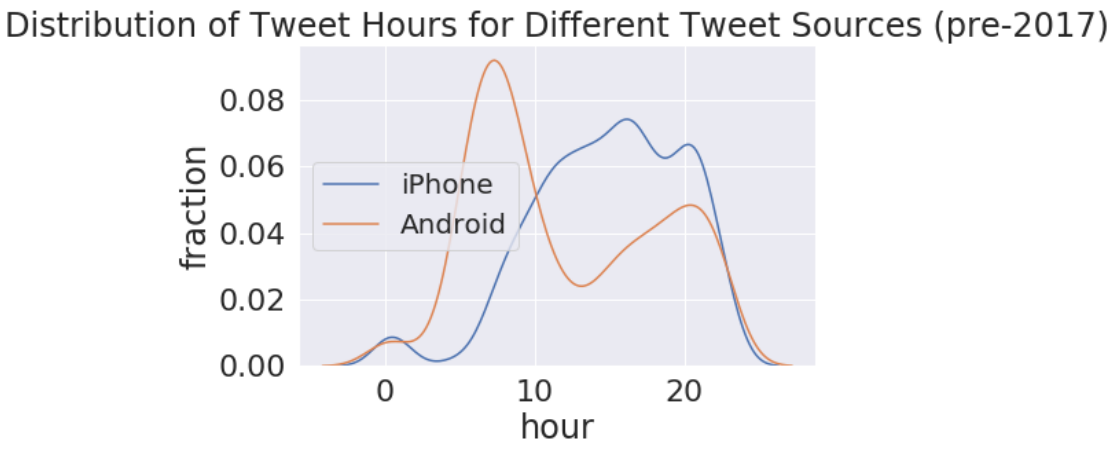


0.1.2 Question 4c

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 [23]:
```
### make your plot here
before2017_trump = trump[trump['year']<2017]
iphone_hrbefore2017 = before2017_trump[before2017_trump['source']=='Twitter for iPhone']['est_
android_hrbefore2017 = before2017_trump[before2017_trump['source']=='Twitter for Android']['es
sns.distplot(iphone_hrbefore2017, hist=False, label='iPhone')
sns.distplot(android_hrbefore2017, hist=False, label='Android')
plt.xlabel('hour')
plt.ylabel('fraction')
plt.title('Distribution of Tweet Hours for Different Tweet Sources (pre-2017)')
plt.legend(prop={'size': 20})
sns.set(font_scale=2);
```



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 claim seems to be supported by the comparison of pre-2017 graph and the plot in 4b. During the campaign the tweets sent from an iPhone versus an Android were on very different schedules and show different peak points. The Android tweets were sent mostly between 5am-10am. While the rest of the day the iPhone tweets were sent more steadily after 10am. We can also see that the graph in 4b shows that his tweeting is more consistent on both devices thus concluding he must be mananging his own twitter.

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

Please score the sentiment of one of the following words: - police - order - Democrat - Republican - gun - dog - technology - TikTok - security - face-mask - science - climate change - vaccine

What score did you give it and why? Can you think of a situation in which this word would carry the opposite sentiment to the one you've just assigned?

Usually the word police would be a positive number but in this case I would give it -0.7. Given the negative reprocussion of the police during this time its score is negative but also not fully -1 since there are still good positive things about the police.

0.2.2 Question 5b

VADER aggregates the sentiment of words in order to determine the overall sentiment of a sentence, and further aggregates sentences to assign just one aggregated score to a whole tweet or collection of tweets. This is a complex process and if you'd like to learn more about how VADER aggregates sentiment, here is the info at this link.

Are there circumstances (e.g. certain kinds of language or data) when you might not want to use VADER? What features of human speech might VADER misrepresent or fail to capture?

I dont think VADER would be particularly helpful with english slang as the positivety/negativety of the words/phrase usually depend on the tone it is said in. Some phrases have positive and negative connotations and typically depend on the WAY the sentence was said.

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 that the negative tweets are pretty negative in the sense that they use words such as: "outrageous, liar, attack, hoax etc." these words are typically not used in a positive context. Same with the positive tweets there are a lot of words like: "congratulations, thank you, beautiful, !". Even the puncuation in this case is positive using exclaimation points instead of periods.

0.4 Question 6

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

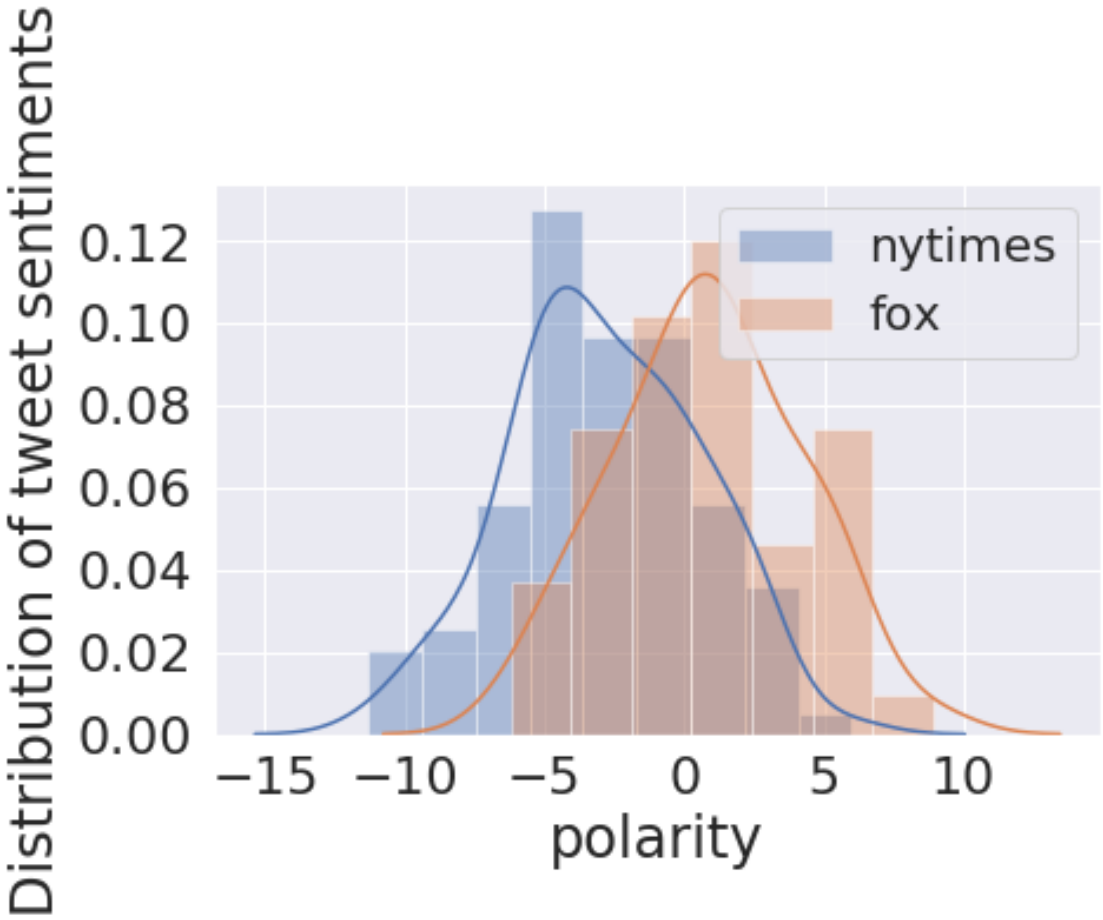
0.4.1 Question 6a

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 [37]: nytimes_id = tidy_format[tidy_format['word']=='nytimes'].index
         nytimes_polarity = trump[trump.index.isin(nytimes_id)]
         fox_id = tidy_format[tidy_format['word']=='fox'].index
         fox_polarity = trump[trump.index.isin(fox_id)]
         sns.distplot(nytimes_polarity['polarity'], label='nytimes')
         sns.distplot(fox_polarity['polarity'], label='fox')
         plt.legend()
         plt.legend('Distribution of tweet sentiments for tweets containing nytimes compared to contain
         plt.xlabel('polarity')
         plt.ylabel('Distribution of tweet sentiments')
         sns.set(font_scale=5)
         sns.set(rc={'figure.figsize':(20,10)})
         plt.legend(prop={'size':20})
```

```
Out[37]: <matplotlib.legend.Legend at 0x7f66b9d986d0>
```


Distribution of tweet sentiments
0.12
0.10
0.08
0.06
0.04
0.02
0.00
−15
−10
−5
0
5
10
polarity
nytimes
fox

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 nytimes sentiments show a more symetric distribution (bell curve). Majority of the polarity of the tweets lie in the negative values. Fox tends to have similar features but less negative. The overlap is roughly -6 to 6.

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

Based on the distributions the tweet sentiments show roughly the same spread but the blue (hashtag or link) skews to the right which shows more positive polarity. The peak for hashtags/links is presented at 0.0 and 1.0.