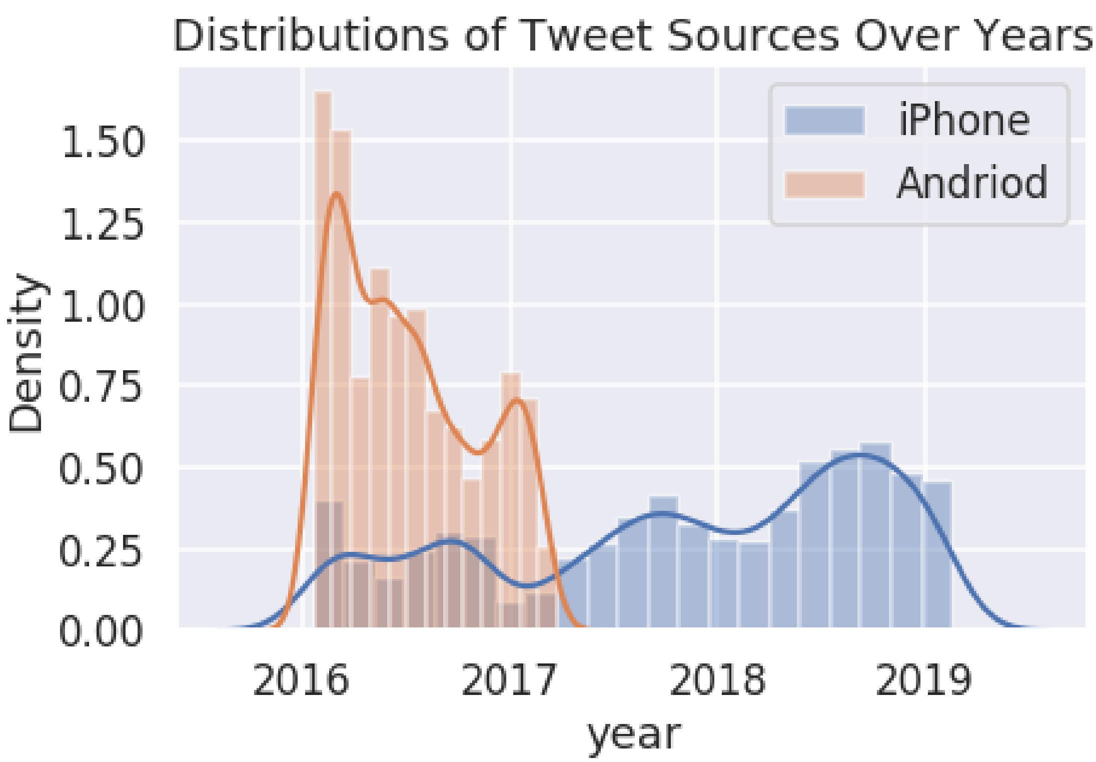
0.1 Question 0

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 data analysis on the President's tweets to know what words or phrases he prefers to use to understand his general view/response to certain happenings or responses to individuals, this might be useful to his political opponents, such as the Biden campaign. First of all, it would be useful for former Vice President Biden to know what kind of words or phrases Trump might use in debates and his campaign to use his words in advertisments targeting Trump.

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:

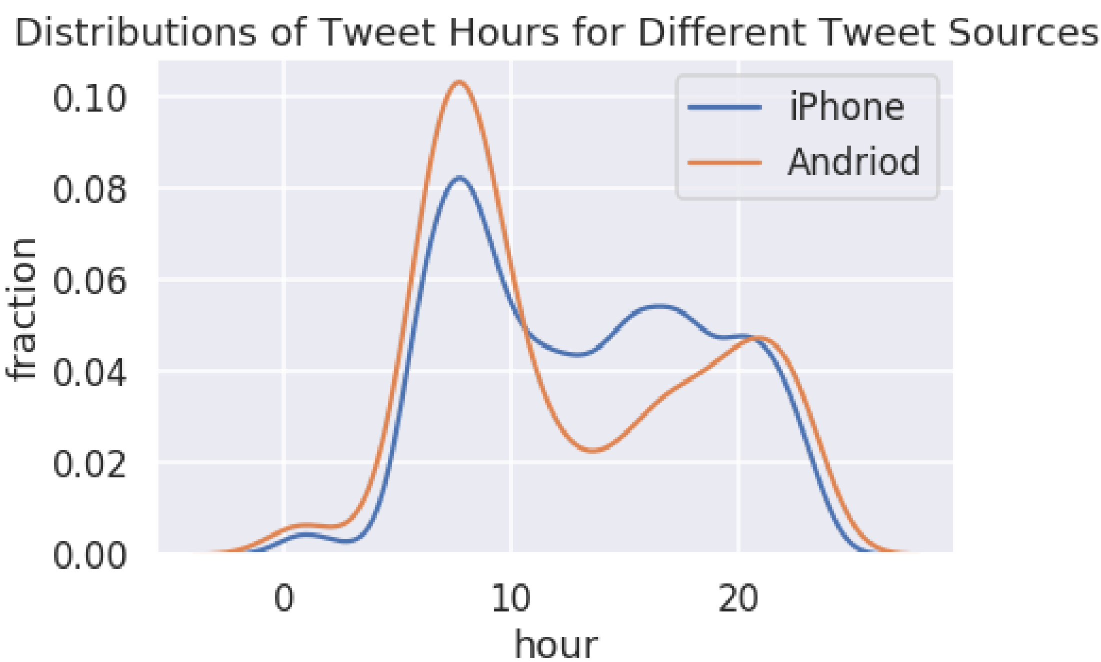
In [14]:
```
iphone = trump[trump['source'] == 'Twitter for iPhone']['year']
android = trump[trump['source'] == 'Twitter for Android']['year']
sns.distplot(iphone, label = 'iPhone')
sns.distplot(android, label = 'Andriod')
plt.xlabel('year')
plt.title('Distributions of Tweet Sources Over Years')
plt.legend();
```



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
         hours_iphone = trump[trump['source'] == 'Twitter for iPhone']['hour']
         hours_android = trump[trump['source'] == 'Twitter for Android']['hour']
         sns.distplot(hours_iphone, hist = False, label = 'iPhone')
         sns.distplot(hours_android, hist= False,  label = 'Andriod')
         plt.xlabel('hour')
         plt.ylabel('fraction')
         plt.title('Distributions of Tweet Hours for Different Tweet Sources')
         plt.legend();
```

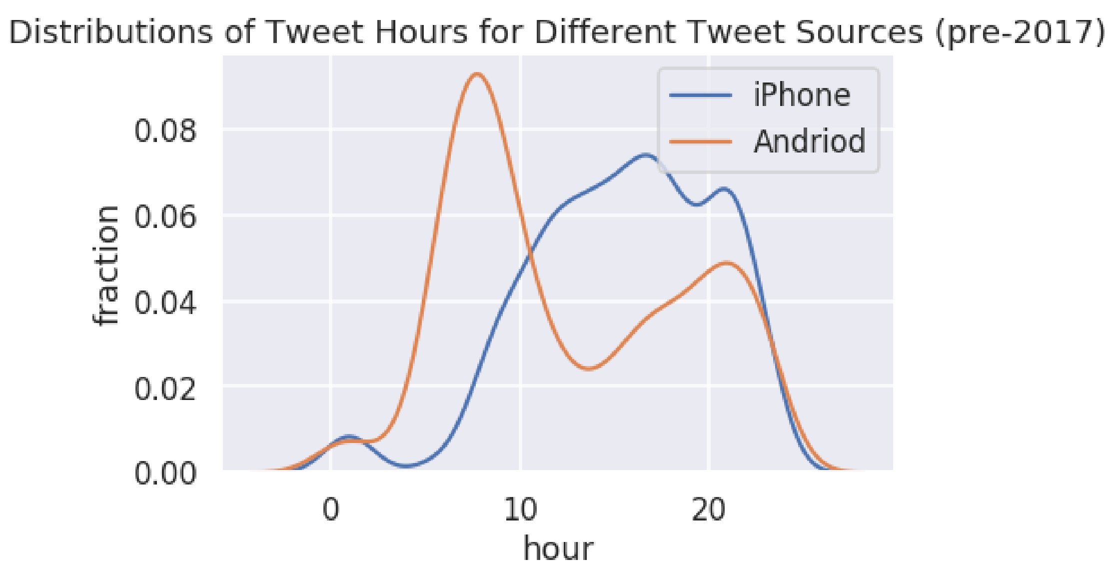


0.1.2 Question 4c

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
         pre_2017 = trump[trump['year'] < 2017]
         pre_2017_iphone = pre_2017[pre_2017['source'] == 'Twitter for iPhone']['hour']
         pre_2017_android = pre_2017[pre_2017['source'] == 'Twitter for Android']['hour']
         sns.distplot(pre_2017_iphone, hist = False, label = 'iPhone')
         sns.distplot(pre_2017_android, hist= False,  label = 'Andriod')
         plt.xlabel('hour')
         plt.ylabel('fraction')
         plt.title('Distributions of Tweet Hours for Different Tweet Sources (pre-2017)')
         plt.legend();
```



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?

Yes, the figure supports the theory, because it can be observed that tweets from Android devices were usually in the morning, most likely when Donald Trump himself is tweeting and the tweets from the iPhone devices were more frequent throughout the day, which is when his staff are usually working (writing the tweets). To add support for this claim it would be helpful to analyze

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?

For police I gave a score of -0.6, due to the current events that are centered around police brutality and violence. A situation in which the word police could carry the opposit sentiment maybe news of a police helping someone, or maybe involving themselves in improving the relationship between the police force and the victims of police brutality.

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?

Sarcasm, references to different times (past, present, future) and emphasis are some features that VADER will most likely fail to capture because these are features that can be understood only through context or tone.

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, these tweets were accuraetly represented by 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`.

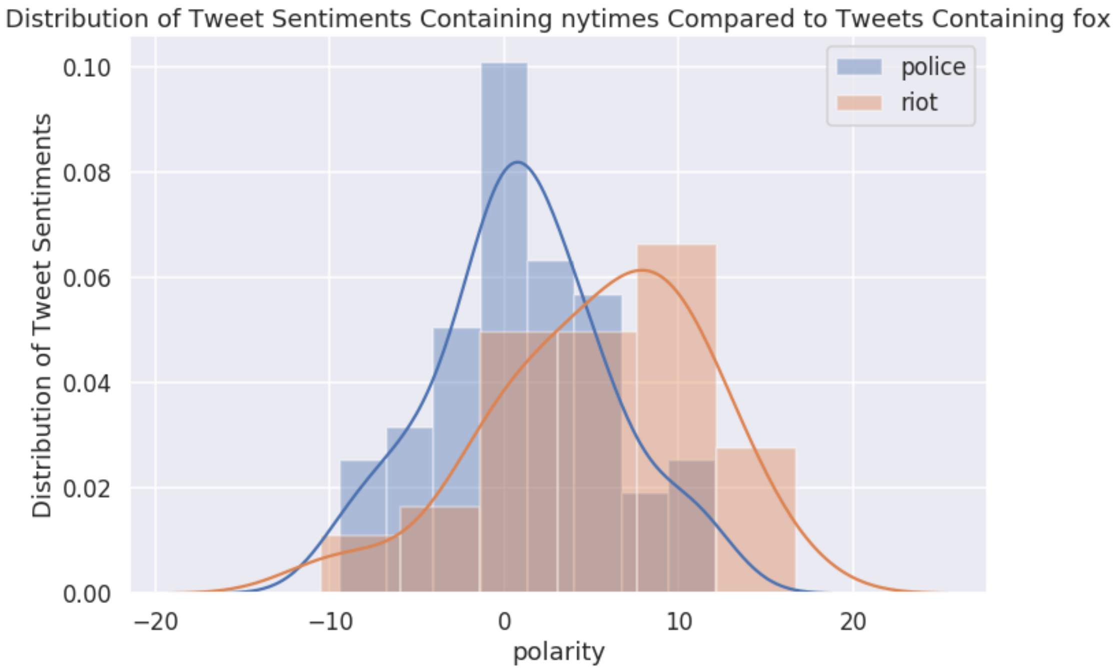
In [34]:
```
trump
```

Out[34]:
```
                                          time               source  \
690171032150237184  2016-01-21 13:56:11+00:00  Twitter for Android
690171403388104704  2016-01-21 13:57:39+00:00  Twitter for Android
690173226341691392  2016-01-21 14:04:54+00:00  Twitter for Android
690176882055114758  2016-01-21 14:19:26+00:00  Twitter for Android
690180284189310976  2016-01-21 14:32:57+00:00  Twitter for Android
…                                         …                   …
1096547516290543617 2019-02-15 23:11:15+00:00   Twitter for iPhone
1096812333333184512 2019-02-16 16:43:32+00:00   Twitter for iPhone
1096856815810342912 2019-02-16 19:40:18+00:00   Twitter for iPhone
1096924708132581377 2019-02-17 00:10:04+00:00   Twitter for iPhone
1096926633708134406 2019-02-17 00:17:44+00:00   Twitter for iPhone


690171032150237184
690171403388104704
690173226341691392
690176882055114758
690180284189310976
…
1096547516290543617
1096812333333184512
1096856815810342912
1096924708132581377
1096926633708134406  trade negotiators have just returned from china where the meetings on tra

                     retweet_count         year                   est_time  \
690171032150237184            1059  2016.054645 2016-01-21 08:56:11-05:00
690171403388104704            1339  2016.054645 2016-01-21 08:57:39-05:00
690173226341691392            2006  2016.054645 2016-01-21 09:04:54-05:00
690176882055114758            2266  2016.054645 2016-01-21 09:19:26-05:00
690180284189310976            2886  2016.054645 2016-01-21 09:32:57-05:00
…                                …            …                          …
```

```
1096547516290543617          21296  2019.123288 2019-02-15 18:11:15-05:00
1096812333333184512          17134  2019.126027 2019-02-16 11:43:32-05:00
1096856815810342912          29569  2019.126027 2019-02-16 14:40:18-05:00
1096924708132581377          21811  2019.128767 2019-02-16 19:10:04-05:00
1096926633708134406           8325  2019.128767 2019-02-16 19:17:44-05:00

                          hour  \
690171032150237184    8.936389
690171403388104704    8.960833
690173226341691392    9.081667
690176882055114758    9.323889
690180284189310976    9.549167
…                          …
1096547516290543617  18.187500
1096812333333184512  11.725556
1096856815810342912  14.671667
1096924708132581377  19.167778
1096926633708134406  19.295556


690171032150237184
690171403388104704
690173226341691392
690176882055114758
690180284189310976
…
1096547516290543617
1096812333333184512
1096856815810342912
1096924708132581377
1096926633708134406  trade negotiators have just returned from china where the meetings on trac

                     polarity
690171032150237184        0.0
690171403388104704       -2.6
690173226341691392       -6.0
690176882055114758        4.3
690180284189310976       -2.6
…                         …
1096547516290543617       4.3
1096812333333184512       0.0
1096856815810342912       0.0
1096924708132581377       0.0
1096926633708134406       3.2

[10370 rows x 9 columns]
```

In [35]:
```python
plt.figure(figsize = [10,7])
nytimes = trump[trump['text'].str.contains('police')]['polarity']
fox = trump[trump['text'].str.contains('riot')]['polarity']
sns.distplot(nytimes, label = 'police')
sns.distplot(fox, label = 'riot')
plt.ylabel('Distribution of Tweet Sentiments')
```

```
plt.title('Distribution of Tweet Sentiments Containing nytimes Compared to Tweets Containing fo
plt.legend();
```



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 distribution of sentiments for both nytimes and fox are somewhat symmetrical. Nytimes was symmetrical at around a polarity of -5, while fox was symmetrical at around a polarity of -1. Furthermore, the distribution of the two words overlapped. I tried the words police and riot in reference to recent events. The distribution of sentiment for police was symmetrical at around 0 and also had a mode at 0. Meanwhile, riot had a mode at around 10 and is skewed to the right. Both overlapped, with police overall having more weight on negative polarity and riot leaning towards a positive polarity.

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

Hastag or link is bimodal, while both hastag or link and no hastag and link have a mode at 0. Hastag or link is somewhat symmetrical while slightly skewed to the left, while no hastag nad link is symmetry and has a somewhat normal curve.