# Query based Sentiment Analysis using Python

In [1]:

```
# General:
import tweepy           # To consume Twitter's API
import pandas as pd     # To handle data
import numpy as np      # For number computing

# For plotting and visualization:
from IPython.display import display
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

# Creating a Twitter App

In [2]:

```
# Twitter App access keys for @user

# Consume:
CONSUMER_KEY    = 'yzxNyCyjkVJJrpNY0L3AphSVV'
CONSUMER_SECRET = 'e7xhbuIqTM520PCtwGrSrSIXWyrQbH7ckr9s0ToOlQE6tek4OP'

# Access:
ACCESS_TOKEN  = '2186056028-BPexxHqPrML7hcvwIuLfX3itD8IZPqguuksDHtn'
ACCESS_SECRET = 'vvhbgIqa7jK5nBAO9P4Mh1MTFvJdWH3Fl5bbzGhai9Dah'
```

In [3]:

```
# We import our access keys:
from credentials import *    # This will allow us to use the keys as variables

# API's setup:
def twitter_setup():
    """
    Utility function to setup the Twitter's API
    with our access keys provided.
    """
    # Authentication and access using keys:
    auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
    auth.set_access_token(ACCESS_TOKEN, ACCESS_SECRET)

    # Return API with authentication:
    api = tweepy.API(auth)
    return api
```

In [4]:

```
extractor = twitter_setup()
trends1 = extractor.trends_place(1) # from the end of your code
# trends1 is a list with only one element in it, which is a
# dict which we'll put in data.
data = trends1[0]
# grab the trends
trends = data['trends']
# grab the name from each trend
names = [trend['name'] for trend in trends]
# put all the names together with a ' ' separating them
trendsName = "\n".join(names)
print(trendsName)
```

```
Dória
#ال_سعود_ساده_وقاده
#DíaDelMédico
#وش_بتسوي_لو_صرت_الملك
#23Oct
Megyn Kelly
#TEMPO_KAI
Eli Apple
Parejo
Lucas Vázquez
Baby One More Time
Rodrigo de la Serna
Juremir Machado
Sandra Day O'Connor
Polichacao
Scorpio
辻谷さん
#OBomDesseAnoFoi
#FelizMartes
#AEKFCB
#يامحمد_الهمه_شعبك_معك_للقمه
#YaşadıkçaTürkçüyüz
#ExpectationsAtMidnight
#AtentadoFakeDoPT
#TuesdayThoughts
#BielNoPânico
#Güneşİle3YılDaha
#EverybodyShouldGet
#غرد_وكانك_في_التسعينات
#يكرهك_الشخص_اللي
#herkesyoluna
#SomethingWickedIn3Words
#MUNJUV
#پرسپولیس
#نطالب_بتجديد_عقد_عموري
#الغا_قناه_الجزيره
#MehmetçikKutluZafer
#Commissione
#PatisserieSongs
#عموري_رباط_صليبي
#HappyKrystalDay
#seoul
#السد_بيرسبوليس
#خمسين_هكتار_لصغار_المزارعين
#9DaysForSRKDay
#ViswasamPongal2019
```

```
#MVTAznarCasado
#افريقيا_يا_اهلي
#AdayımızYaşarAydın
#3YearsofSGFG
```

In [5]:

```
query="narendramodi" # Enter user or query
```

In [6]:

```
# We create an extractor object:
extractor = twitter_setup()

# We create a tweet list based on query:

# tweets = extractor.search(q=query, count=100)

# We create a tweet list based on user

tweets = extractor.user_timeline(screen_name=query, count=100)

print("Number of tweets extracted: {}.\n".format(len(tweets)))

# We print the most recent 5 tweets:
print("5 recent tweets:\n")
for tweet in tweets[:5]:
    print(tweet.text)
    print()
```

```
Number of tweets extracted: 100.

5 recent tweets:

The 'मैं नहीं, हम' portal is based on the principle of #Self4Society, whic
h will bring together IT professionals an… https://t.co/ynkHM8h9Ib (ht
tps://t.co/ynkHM8h9Ib)

Tomorrow afternoon, I eagerly look forward to taking part in an intere
sting programme, a Town Hall-style interactio… https://t.co/MX8UtCeGcl
 (https://t.co/MX8UtCeGcl)

#MahatmaGandhi continua a servir de inspiração enquanto a 🇮🇳 e 🇵🇹 cons
troem uma parceria no século XXI. #Gandhi150

O famoso flautista Português Rão Kyao interpreta a Bhajan favorita de
 Mahatma Gandhi, Vaishnav Jan To, em frente à… https://t.co/9s899jqmaA
 (https://t.co/9s899jqmaA)

Renowned Portuguese flutist Rão Kyao plays Gandhi Ji's favourite 'Vais
hnava Jana To', in front of iconic Torre de B… https://t.co/MZsF3so5NC
 (https://t.co/MZsF3so5NC)
```

In [7]:

```
# We create a pandas dataframe as follows:
data = pd.DataFrame(data=[tweet.text for tweet in tweets], columns=['Tweets'])

# We display the first 10 elements of the dataframe:
display(data.head(10))
```

| | Tweets |
|---|---|
| 0 | The ‘मैं नहीं, हम’ portal is based on the prin... |
| 1 | Tomorrow afternoon, I eagerly look forward to ... |
| 2 | #MahatmaGandhi continua a servir de inspiração... |
| 3 | O famoso flautista Português Rão Kyao interpre... |
| 4 | Renowned Portuguese flutist Rão Kyao plays Gan... |
| 5 | ...تكريما لذكرى مهاتما غاندي، شاركت فلسطين العالم |
| 6 | Joining the world in paying homage to Bapu, Pa... |
| 7 | La Huaca Pucllana, pirámide de adobe (400 d.C)... |
| 8 | El Perú contemporáneo rinde tributo a la canci... |
| 9 | Hear ‘Vaishnava Jana To' in the voice of one o... |

In [8]:

```
# Internal methods of a single tweet object:
print(dir(tweets[0]))
```

```
['__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq_
_', '__format__', '__ge__', '__getattribute__', '__getstate__', '__gt_
_', '__hash__', '__init__', '__le__', '__lt__', '__module__', '__ne_
_', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr_
_', '__sizeof__', '__str__', '__subclasshook__', '__weakref__', '_ap
i', '_json', 'author', 'contributors', 'coordinates', 'created_at', 'd
estroy', 'entities', 'favorite', 'favorite_count', 'favorited', 'geo',
'id', 'id_str', 'in_reply_to_screen_name', 'in_reply_to_status_id', 'i
n_reply_to_status_id_str', 'in_reply_to_user_id', 'in_reply_to_user_id
_str', 'is_quote_status', 'lang', 'parse', 'parse_list', 'place', 'pos
sibly_sensitive', 'retweet', 'retweet_count', 'retweeted', 'retweets',
'source', 'source_url', 'text', 'truncated', 'user']
```

# Adding relevant info to our dataframe

In [9]:

```
# We add relevant data:
data['len']  = np.array([len(tweet.text) for tweet in tweets])
data['ID']   = np.array([tweet.id for tweet in tweets])
data['Date'] = np.array([tweet.created_at for tweet in tweets])
data['Source'] = np.array([tweet.source for tweet in tweets])
data['Likes']  = np.array([tweet.favorite_count for tweet in tweets])
data['RTs']    = np.array([tweet.retweet_count for tweet in tweets])
data['Language'] = np.array([tweet.lang for tweet in tweets])
```

In [10]:

```
# Display of first 10 elements from dataframe:
display(data.head(10))
```

| | Tweets | len | ID | Date | Source | Likes | RTs | Language |
|---|---|---|---|---|---|---|---|---|
| **0** | The ‘मैं नहीं, हम’ portal is based on the prin... | 140 | 1054747482884759553 | 2018-10-23 14:52:50 | Twitter Web Client | 4482 | 1276 | en |
| **1** | Tomorrow afternoon, I eagerly look forward to ... | 140 | 1054747470528303104 | 2018-10-23 14:52:47 | Twitter Web Client | 2592 | 814 | en |
| **2** | #MahatmaGandhi continua a servir de inspiração... | 114 | 1054736897552003072 | 2018-10-23 14:10:47 | Twitter Web Client | 1234 | 323 | pt |
| **3** | O famoso flautista Português Rão Kyao interpre... | 139 | 1054736895358525442 | 2018-10-23 14:10:46 | Twitter Web Client | 1213 | 320 | pt |
| **4** | Renowned Portuguese flutist Rão Kyao plays Gan... | 140 | 1054736893387124736 | 2018-10-23 14:10:46 | Twitter Web Client | 2509 | 610 | en |
| **5** | تكريما لذكرى مهاتما غاندي، شاركت فلسطين العالم... | 139 | 1054736519766913024 | 2018-10-23 14:09:17 | Twitter Web Client | 765 | 220 | ar |
| **6** | Joining the world in paying homage to Bapu, Pa... | 140 | 1054736517611040770 | 2018-10-23 14:09:16 | Twitter Web Client | 1879 | 468 | en |
| **7** | La Huaca Pucllana, pirámide de adobe (400 d.C)... | 119 | 1054736142153760768 | 2018-10-23 14:07:46 | Twitter Web Client | 903 | 241 | es |
| **8** | El Perú contemporáneo rinde tributo a la canci... | 140 | 1054736139712712705 | 2018-10-23 14:07:46 | Twitter Web Client | 948 | 246 | es |
| **9** | Hear ‘Vaishnava Jana To' in the voice of one o... | 140 | 1054736137296797696 | 2018-10-23 14:07:45 | Twitter Web Client | 1746 | 433 | en |

In [11]:

```
# We extract the mean of lenghts:
mean = np.mean(data['len'])

print("The lenght's average in tweets: {}".format(mean))
```

```
The lenght's average in tweets: 136.84
```

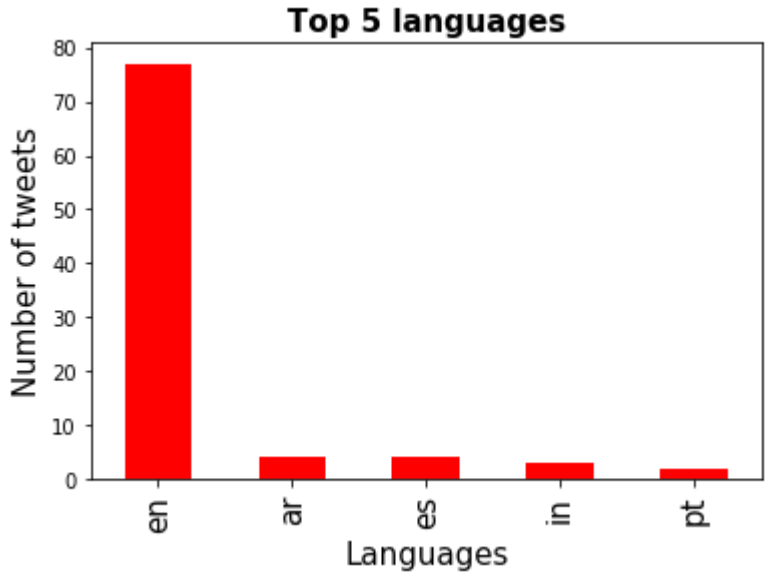
In [12]:

```
tweets_by_lang = data['Language'].value_counts()

fig, ax = plt.subplots()
ax.tick_params(axis='x', labelsize=15)
ax.tick_params(axis='y', labelsize=10)
ax.set_xlabel('Languages', fontsize=15)
ax.set_ylabel('Number of tweets' , fontsize=15)
ax.set_title('Top 5 languages', fontsize=15, fontweight='bold')
tweets_by_lang[:5].plot(ax=ax, kind='bar', color='red')
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1133e9ba8>
```



In [13]:

```
# We extract the tweet with more FAVs and more RTs:

fav_max = np.max(data['Likes'])
rt_max  = np.max(data['RTs'])

fav = data[data.Likes == fav_max].index[0]
rt  = data[data.RTs == rt_max].index[0]

# Max FAVs:
print("The tweet with more likes is: \n{}".format(data['Tweets'][fav]))
print("Number of likes: {}".format(fav_max))
print("{} characters.\n".format(data['len'][fav]))

# Max RTs:
print("The tweet with more retweets is: \n{}".format(data['Tweets'][rt]))
print("Number of retweets: {}".format(rt_max))
print("{} characters.\n".format(data['len'][rt]))
```

```
The tweet with more likes is:
Greetings to @BJP4India President Shri @AmitShah on his birthday. Unde
r Amit Bhai’s leadership, the Party has expan… https://t.co/hQFLlUiw95
 (https://t.co/hQFLlUiw95)
Number of likes: 31846
140 characters.

The tweet with more retweets is:
Extremely saddened by the train accident in Amritsar. The tragedy is h
eart-wrenching. My deepest condolences to the… https://t.co/6jiATjWfM6
 (https://t.co/6jiATjWfM6)
Number of retweets: 7646
140 characters.
```

# Time series of Tweets

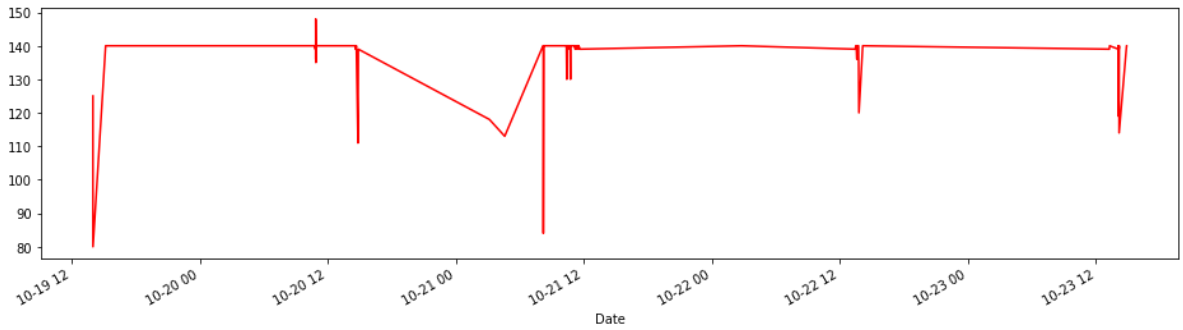
In [14]:

```
# We create time series for data:

tlen = pd.Series(data=data['len'].values, index=data['Date'])
tfav = pd.Series(data=data['Likes'].values, index=data['Date'])
tret = pd.Series(data=data['RTs'].values, index=data['Date'])
```

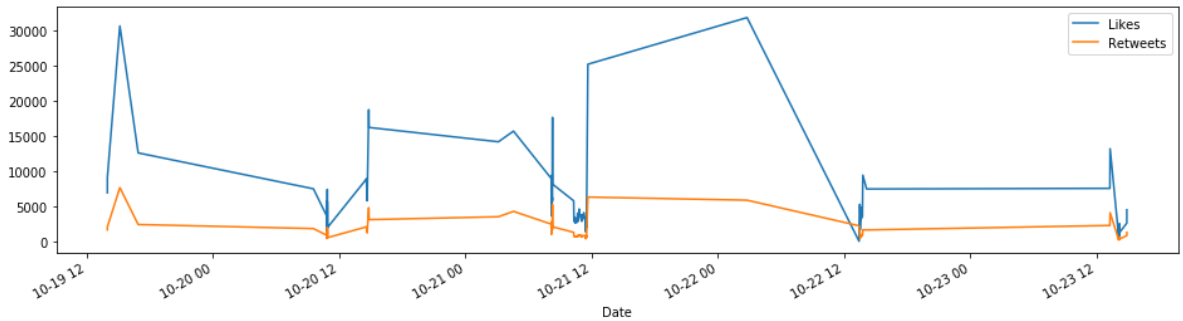
In [15]:

```
# Lenghts along time:
tlen.plot(figsize=(16,4), color='r');
```




In [16]:

```
# Likes vs retweets visualization:
tfav.plot(figsize=(16,4), label="Likes", legend=True)
tret.plot(figsize=(16,4), label="Retweets", legend=True);
```




# Pie charts of sources

In [17]:

```
# We obtain all possible sources:
sources = []
for source in data['Source']:
    if source not in sources:
        sources.append(source)

# We print sources list:
print("Creation of content sources:")
for source in sources:
    print("* {}".format(source))
```

```
Creation of content sources:
* Twitter Web Client
* Twitter for iPhone
* Media Studio
* Periscope
```

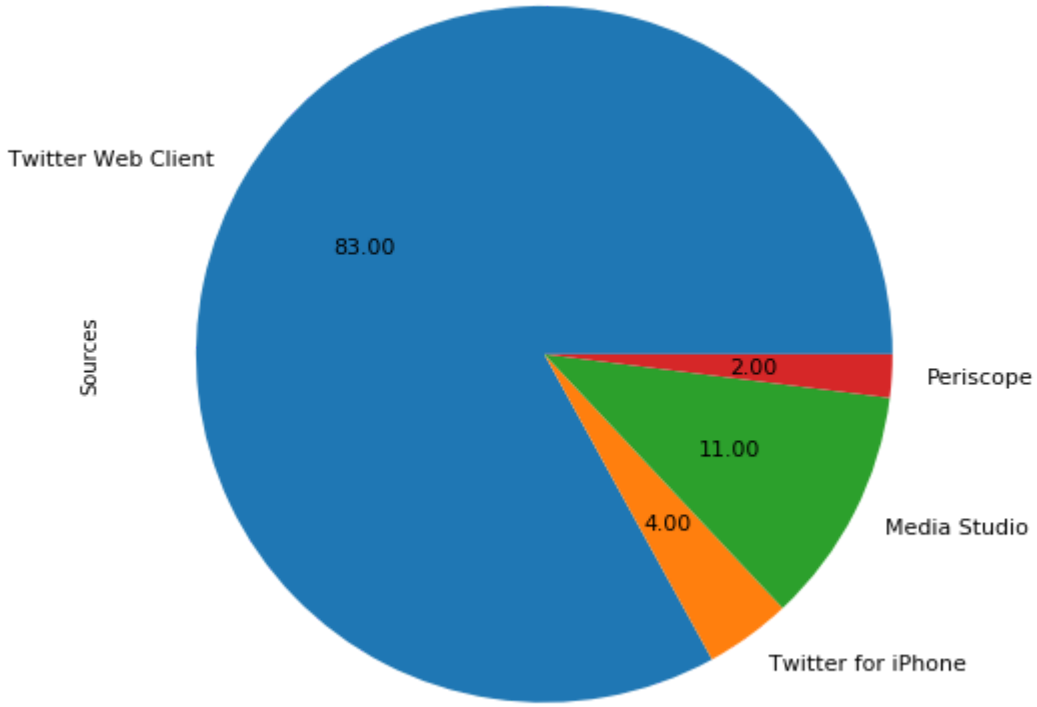
In [18]:

```
# We create a numpy vector mapped to labels:
percent = np.zeros(len(sources))

for source in data['Source']:
    for index in range(len(sources)):
        if source == sources[index]:
            percent[index] += 1
            pass

percent /= 100

# Pie chart:
pie_chart = pd.Series(percent, index=sources, name='Sources')
pie_chart.plot.pie(fontsize=11, autopct='%.2f', figsize=(10,8));
```



# Sentiment analysis

In [19]:

```
from textblob import TextBlob
import re

def clean_tweet(tweet):
    '''
    Utility function to clean the text in a tweet by removing
    links and special characters using regex.
    '''
    return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", t

def analize_sentiment(tweet):
    '''
    Utility function to classify the polarity of a tweet
    using textblob.
    '''
    analysis = TextBlob(clean_tweet(tweet))
    if analysis.sentiment.polarity > 0:
        return 1
    elif analysis.sentiment.polarity == 0:
        return 0
    else:
        return -1
```

In [20]:

```
# We create a column with the result of the analysis:
data['SA'] = np.array([ analize_sentiment(tweet) for tweet in data['Tweets'] ])

# We display the updated dataframe with the new column:
display(data.head(10))
```

| | Tweets | len | ID | Date | Source | Likes | RTs | Language | SA |
|---|---|---|---|---|---|---|---|---|---|
| 0 | The 'मैं नहीं, हम' portal is based on the prin... | 140 | 1054747482884759553 | 2018-10-23 14:52:50 | Twitter Web Client | 4482 | 1276 | en | 0 |
| 1 | Tomorrow afternoon, I eagerly look forward to ... | 140 | 1054747470528303104 | 2018-10-23 14:52:47 | Twitter Web Client | 2592 | 814 | en | 1 |
| 2 | #MahatmaGandhi continua a servir de inspiração... | 114 | 1054736897552003072 | 2018-10-23 14:10:47 | Twitter Web Client | 1234 | 323 | pt | 0 |
| 3 | O famoso flautista Português Rão Kyao interpre... | 139 | 1054736895358525442 | 2018-10-23 14:10:46 | Twitter Web Client | 1213 | 320 | pt | 0 |
| 4 | Renowned Portuguese flutist Rão Kyao plays Gan... | 140 | 1054736893387124736 | 2018-10-23 14:10:46 | Twitter Web Client | 2509 | 610 | en | 1 |
| 5 | تكريما لذكرى مهاتما غاندي، شاركت فلسطين العالم... | 139 | 1054736519766913024 | 2018-10-23 14:09:17 | Twitter Web Client | 765 | 220 | ar | 0 |
| 6 | Joining the world in paying homage to Bapu, Pa... | 140 | 1054736517611040770 | 2018-10-23 14:09:16 | Twitter Web Client | 1879 | 468 | en | 0 |
| 7 | La Huaca Pucllana, pirámide de adobe (400 d.C)... | 119 | 1054736142153760768 | 2018-10-23 14:07:46 | Twitter Web Client | 903 | 241 | es | 0 |
| 8 | El Perú contemporáneo rinde tributo a la canci... | 140 | 1054736139712712705 | 2018-10-23 14:07:46 | Twitter Web Client | 948 | 246 | es | 0 |
| 9 | Hear 'Vaishnava Jana To' in the voice of one o... | 140 | 1054736137296797696 | 2018-10-23 14:07:45 | Twitter Web Client | 1746 | 433 | en | 1 |

In [21]:

```
# We construct lists with classified tweets:

pos_tweets = [ tweet for index, tweet in enumerate(data['Tweets']) if data['SA'][inc
neu_tweets = [ tweet for index, tweet in enumerate(data['Tweets']) if data['SA'][inc
neg_tweets = [ tweet for index, tweet in enumerate(data['Tweets']) if data['SA'][inc
```

In [22]:

```
# We print percentages:

print("Percentage of positive tweets: {}%".format(len(pos_tweets)*100/len(data['Twee
print("Percentage of neutral tweets: {}%".format(len(neu_tweets)*100/len(data['Tweet
print("Percentage of negative tweets: {}%".format(len(neg_tweets)*100/len(data['Twee
```

```
Percentage of positive tweets: 41.0%
Percentage of neutral tweets: 53.0%
Percentage of negative tweets: 6.0%
```

In [23]:

```
# Data to plot
labels = 'Positive Tweets', 'Neutral Tweets','Negative Tweets'
sizes = [len(pos_tweets)*100/len(data['Tweets']), len(neu_tweets)*100/len(data['Twee
colors = ['gold', 'yellowgreen', 'lightcoral']
explode = (0.07, 0.07, 0.07)  # explode 1st slice

# Plot
plt.figure(figsize=(10,6))
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=140)
plt.axis('equal')
plt.show()
```