Name: Pratimesh Shivaji Gaykawad

Task:Stock Market Prediction using Numerical and Textual Analysis

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pylab as pl
%matplotlib inline
import seaborn as sns
```

In [2]:
```
import matplotlib as mpl
from matplotlib import style

#Adjusting the size of matplotlib

mpl.rc('figure', figsize = (15 , 15))
mpl.__version__

#Adjusting the style of matplotlib

style.use('ggplot')
```

In [3]:
```
!pip install pandas-datareader
```

```
Requirement already satisfied: pandas-datareader in c:\users\hp\anaconda3\lib\site-packages (0.9.0)
Requirement already satisfied: lxml in c:\users\hp\anaconda3\lib\site-packages (from pandas-datareader) (4.5.0)
Requirement already satisfied: requests>=2.19.0 in c:\users\hp\anaconda3\lib\site-packages (from pandas-datareader) (2.22.0)
Requirement already satisfied: pandas>=0.23 in c:\users\hp\anaconda3\lib\site-packages (from pandas-datareader) (1.0.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (2019.11.28)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (1.25.8)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (2.8)
Requirement already satisfied: pytz>=2017.2 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=0.23->pandas-datareader) (2019.3)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=0.23->pandas-datareader) (2.8.1)
Requirement already satisfied: numpy>=1.13.3 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=0.23->pandas-datareader) (1.18.1)
Requirement already satisfied: six>=1.5 in c:\users\hp\anaconda3\lib\site-packages (from python-dateutil>=2.6.1->pandas>=0.23->pandas-datareader) (1.14.0)
```

In [4]:
```
import datetime
import pandas_datareader.data as web
from pandas import Series, DataFrame
```

Pandas web data reader is an extension of pandas library to communicate with most updated Financial Data

In [5]:
```
start = datetime.datetime(2010, 1, 1)
end = datetime.datetime(2020, 1, 11)
```

In [6]:
```
df = web.DataReader("AAPL", 'yahoo', start, end)
```

In [7]:
```
#Take a look at dataset

df.head()
```

Out[7]:

Date	High	Low	Open	Close	Volume	Adj Close
2009-12-31	7.619643	7.520000	7.611786	7.526072	352410800.0	6.503574
2010-01-04	7.660714	7.585000	7.622500	7.643214	493729600.0	6.604801
2010-01-05	7.699643	7.616071	7.664286	7.656428	601904800.0	6.616219
2010-01-06	7.686786	7.526786	7.656428	7.534643	552160000.0	6.510980
2010-01-07	7.571429	7.466072	7.562500	7.520714	477131200.0	6.498945

In [8]:
```
#Take a look at dataset

df.tail()
```

Out[8]:

Date	High	Low	Open	Close	Volume	Adj Close
2020-01-06	74.989998	73.187500	73.447502	74.949997	118387200.0	74.436470
2020-01-07	75.224998	74.370003	74.959999	74.597504	108872000.0	74.086395
2020-01-08	76.110001	74.290001	74.290001	75.797501	132079200.0	75.278160
2020-01-09	77.607498	76.550003	76.809998	77.407501	170108400.0	76.877136
2020-01-10	78.167503	77.062500	77.650002	77.582497	140644800.0	77.050926

Rolling Mean (Moving Average)

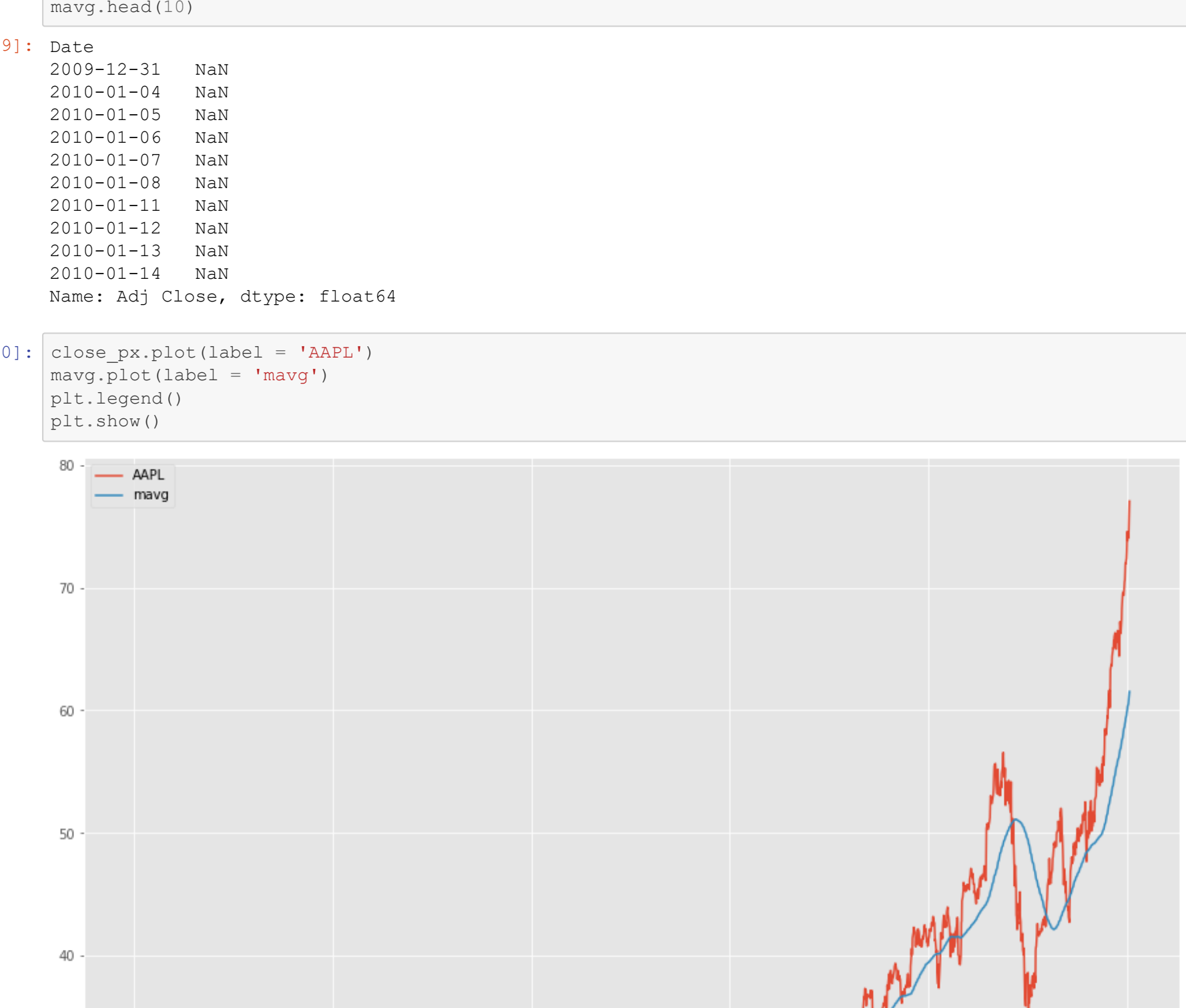
To determine trend

The Moving Average makes the line smooth and showcase the increasing or decreasing trend of stocks price.

In [9]:
```
close_px = df['Adj Close']
mavg = close_px.rolling(window = 100).mean()
mavg.head(10)
```

Out[9]:
```
Date
2009-12-31    NaN
2010-01-04    NaN
2010-01-05    NaN
2010-01-06    NaN
2010-01-07    NaN
2010-01-08    NaN
2010-01-11    NaN
2010-01-12    NaN
2010-01-13    NaN
2010-01-14    NaN
Name: Adj Close, dtype: float64
```

In [10]:
```
close_px.plot(label = 'AAPL')
mavg.plot(label = 'mavg')
plt.legend()
plt.show()
```



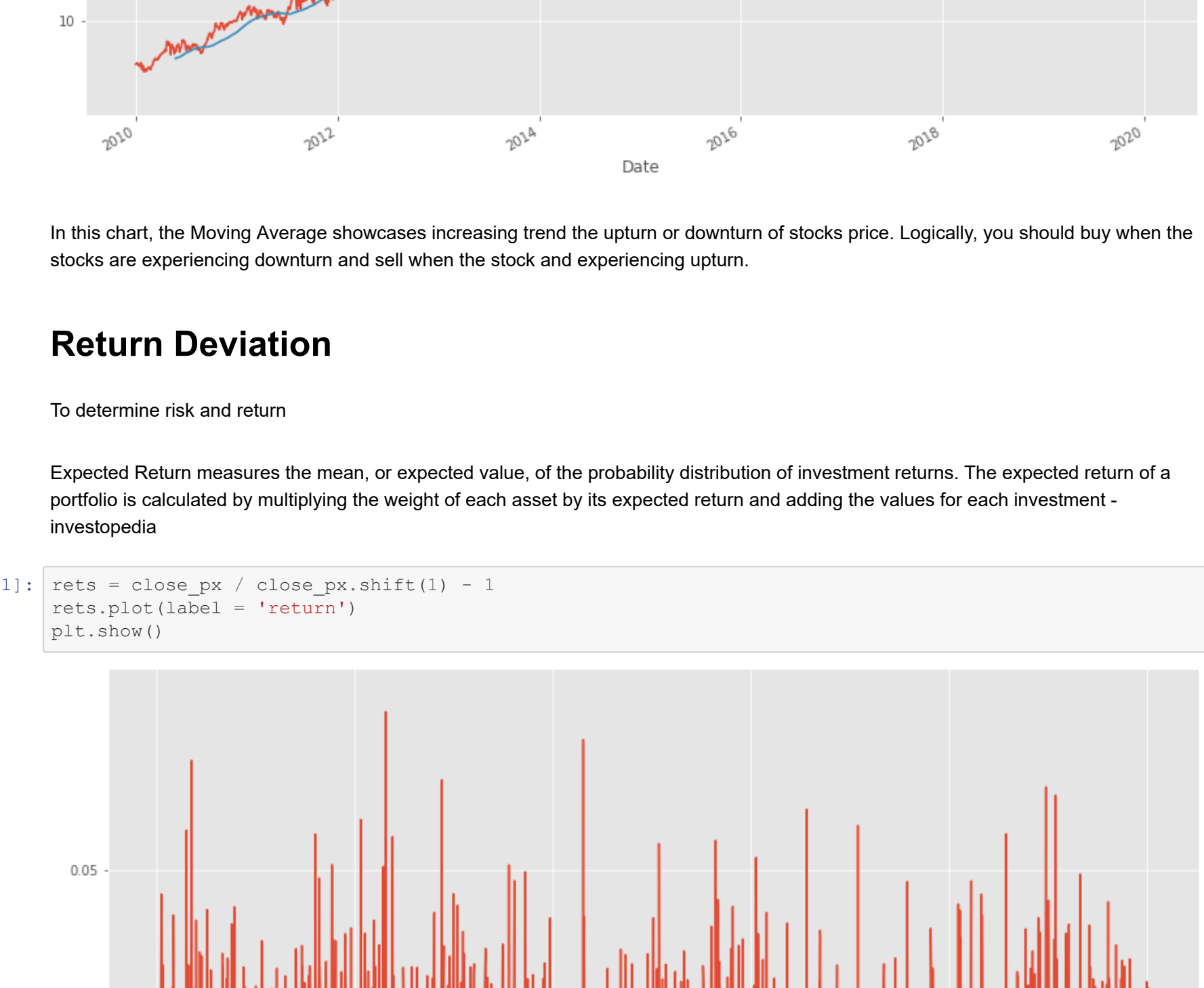
In this chart, the Moving Average showcases increasing trend the upturn or downturn of stocks price. Logically, you should buy when the stocks are experiencing downturn and sell when the stock and experiencing upturn.

Return Deviation

To determine risk and return

Expected Return measures the mean, or expected value, of the probability distribution of investment returns. The expected return of a portfolio is calculated by multiplying the weight of each asset by its expected return and adding the values for each investment - investopedia

In [11]:
```
rets = close_px / close_px.shift(1) - 1
rets.plot(label = 'return')
plt.show()
```



logically, our ideal stocks should return as high and stable as possible. if you are risk averse, you might want to avoid this stocks as you saw the 10% drop in 2013. This decision is heavily subjected to your general sentiment of stocks and competitor analysis.

Analysis Competitors Stocks

In this segment , we are going to analyse on how one company performs in relative with its competitors. Let'sassume we are intrested in technology companies and want to compare the big guns: Apple, GE, Google, IBM, and Microsoft.

In [12]:
```
dfcomp = web.DataReader(['AAPL', 'GE', 'GOOG', 'IBM', 'MSFT'], 'yahoo', start = start, end = end)['Adj Close']
dfcomp.head(10)
```

```
C:\Users\HP\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Pandas doesn't allow columns to be created via a new attribute name - see https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access
  """Entry point for launching an IPython kernel.
```

Out[12]:

Date	AAPL	GE	GOOG	IBM	MSFT
2009-12-31	6.503574	10.493408	308.832428	92.406113	23.801456
2010-01-04	6.604801	10.715343	312.204773	93.500313	24.168472
2010-01-05	6.616219	10.770829	310.829926	92.370834	24.176279
2010-01-06	6.510980	10.715343	302.994293	91.770790	24.027906
2010-01-07	6.498945	11.270185	295.940735	91.453102	23.778025
2010-01-08	6.542150	11.512927	299.885956	92.370834	23.942017
2010-01-11	6.484439	11.623892	299.432648	91.403702	23.637472
2010-01-12	6.410679	11.630828	294.137512	92.130806	23.481291
2010-01-13	6.501104	11.672442	292.448822	91.933167	23.699940
2010-01-14	6.463451	11.582279	293.823669	93.401497	24.176279

Correlation Analysis

Does one competitor affect others?

We can analyse the competition by running the percentage change and correlation function in pandas. Percentage change will find how the price changes compared to the previous day which defines returns. Knowing the correlation will help us see whether the returns are affected by other stocks returns

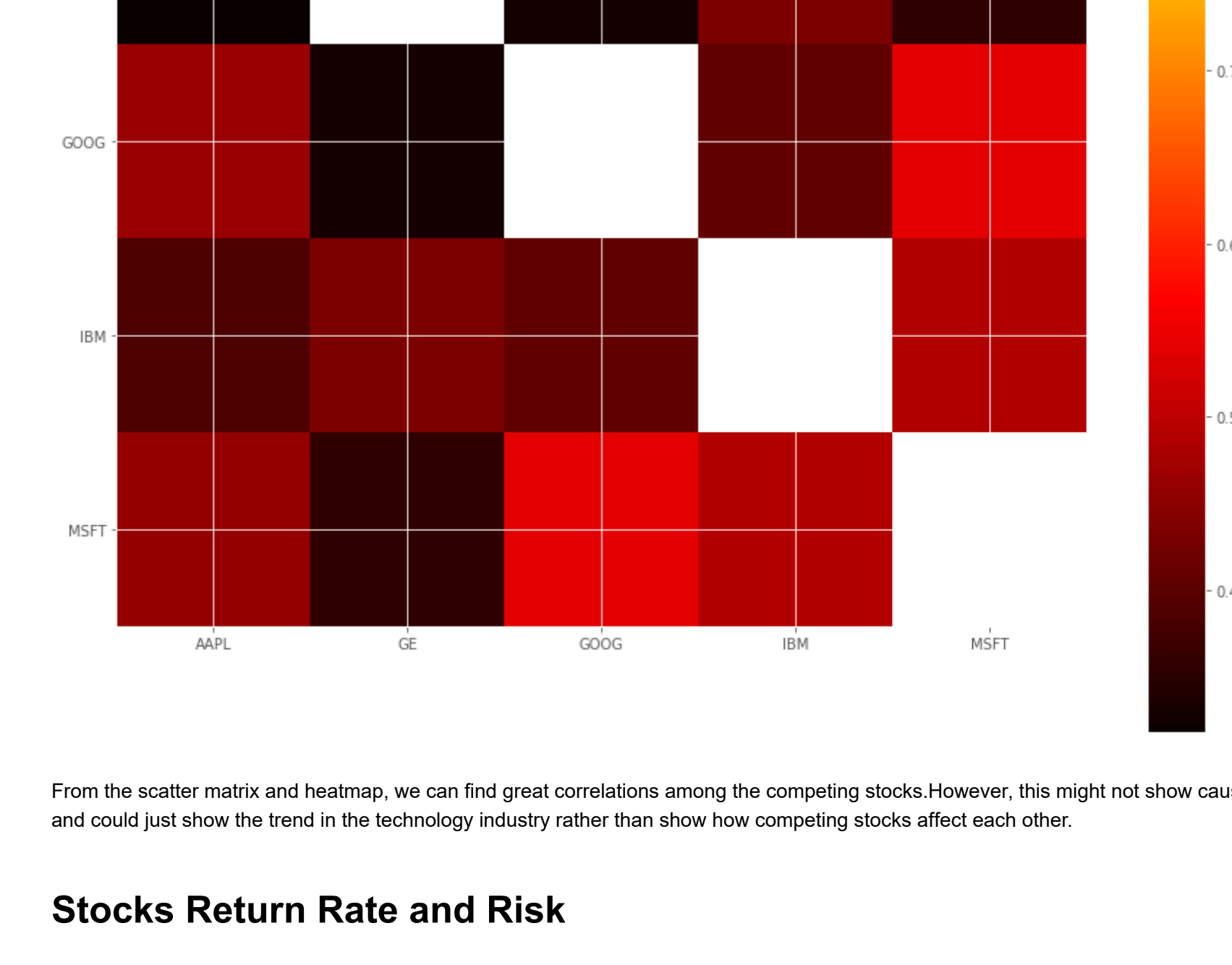
In [13]:
```
retscomp = dfcomp.pct_change()
corr = retscomp.corr()
retscomp.head(10)
```

Out[13]:

Date	AAPL	GE	GOOG	IBM	MSFT
2009-12-31	NaN	NaN	NaN	NaN	NaN
2010-01-04	0.015565	0.021150	0.010920	0.011841	0.015420
2010-01-05	0.001729	0.005178	-0.004404	-0.012080	0.000323
2010-01-06	-0.015906	-0.005152	-0.025209	-0.006496	-0.006137
2010-01-07	-0.001848	0.051780	-0.023280	-0.003462	-0.010400
2010-01-08	0.006648	0.021538	0.013331	0.010035	0.006897
2010-01-11	-0.008821	0.009638	-0.001512	-0.010470	-0.012720
2010-01-12	-0.011375	0.000597	-0.017684	0.007955	-0.006607
2010-01-13	0.014105	0.003578	-0.005741	-0.002145	0.009312
2010-01-14	-0.005792	-0.007724	0.004701	0.015972	0.020099

To prove the positive correlations, we will use heat map to visualize the correlation ranges among the competing stocks. Notice that the lighter the color, the more correlated the two stocks are.

In [14]:
```
plt.imshow(corr, cmap = 'hot', interpolation = 'none')
plt.colorbar()
plt.xticks(range(len(corr)), corr.columns)
plt.yticks(range(len(corr)), corr.columns);
```

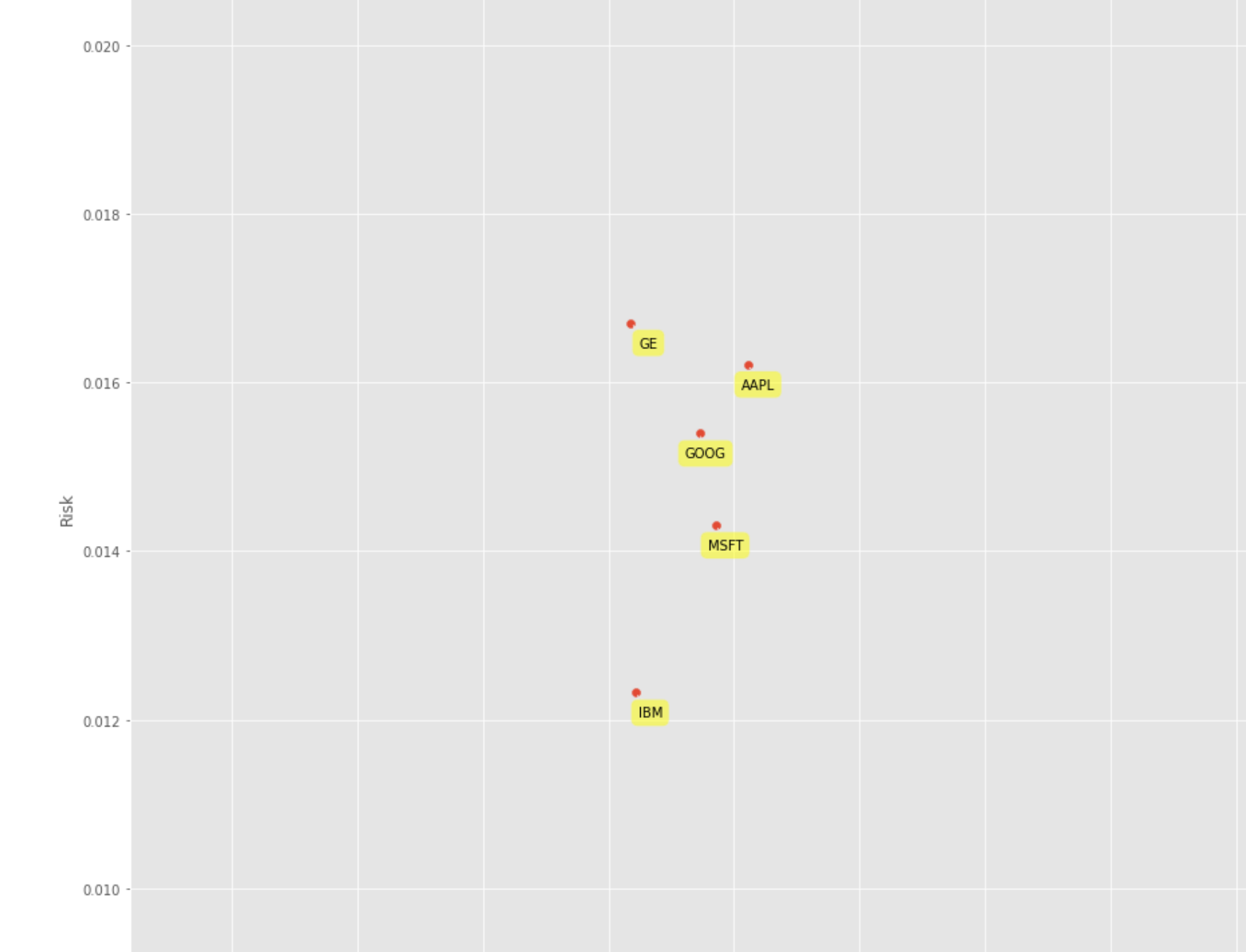


From the scatter matrix and heatmap, we can find great correlations among the competing stocks.However, this might not show causally, and could just show the trend in the technology industry rather than show how competing stocks affect each other.

Stocks Return Rate and Risk

Apart from correlation, we also analyse each stock's risks and returns. In this case we are extracting the average of returns (Return Rate) and the standard deviation of returns (Risk)

In [15]:
```
plt.scatter(retscomp.mean(), retscomp.std())
plt.xlabel('Expected returns')
plt.ylabel('Risk')

for label, x, y in zip(retscomp.columns, retscomp.mean(), retscomp.std()):
    plt.annotate(label, xy = (x, y), xytext = (20 , -20), textcoords = 'offset points',
    ha = 'right', va = 'bottom', bbox = dict(boxstyle = 'round, pad = 0.5', fc = 'yellow',
    alpha = 0.5), arrowprops = dict(arrowstyle = '->', connectionstyle = 'arc3, rad = 0'))
```



India-News-Headlines file imported and doing Numerical Analysis

In [16]:
```
df1 = pd.read_csv("E:\india-news-headlines.csv")
```

In [17]:
```
#Take a look at dataset

df1.head(20)
```

Out[17]:

	publish_date	headline_category	headline_text
0	20010101	sports.wwe	win over cena satisfying but defeating underta...
1	20010102	unknown	Status quo will not be disturbed at Ayodhya; s...
2	20010102	unknown	Fissures in Hurriyat over Pak visit
3	20010102	unknown	America's unwanted heading for India?
4	20010102	unknown	For bigwigs; it is destination Goa
5	20010102	unknown	Extra buses to clear tourist traffic
6	20010102	unknown	Dilute the power of transfers; says Riberio
7	20010102	unknown	Focus shifts to teaching of Hindi
8	20010102	unknown	IT will become compulsory in schools
9	20010102	unknown	Move to stop freedom fighters' pension flayed
10	20010102	unknown	Gilani claims he applied for passport 2 years ago
11	20010102	unknown	Parivar dismisses PM's warning
12	20010102	unknown	India; Pak exchange lists of N-plants
13	20010102	unknown	Will Qureshi's return really help the govt?
14	20010102	unknown	PM's tacit message: Put Ram tample on hold
15	20010102	unknown	Text of the Prime Minister's article
16	20010102	unknown	NCW to focus on violence against women
17	20010102	unknown	BBC's reality TV will focus on AIIMS
18	20010102	unknown	Jaitley firm on legal reforms
19	20010102	unknown	Hoshangabad farmers have enough water

In [18]:
```
df1.tail(20)
```

Out[18]:

	publish_date	headline_category	headline_text
3297152	20200630	entertainment.hindi.bollywood	sanjay dutt remembers jagdeep on his demise sa...
3297153	20200630	home.education	telangana intermediate second year advanced su...
3297154	20200630	entertainment.hindi.bollywood	salim khan comedy ke liye timing bohot zaroori...
3297155	20200630	india	clinical trials of coronavirus vaccine begin i...
3297156	20200630	india	india again rejects chinas claim over galwan v...
3297157	20200630	india	pm modi to inaugurate asias largest 750 mw rew...
3297158	20200630	city.bengaluru	what bengaluru can do to tackle covid surge
3297159	20200630	entertainment.hindi.bollywood	prabhas gets nostalgic as baahubali the beginn...
3297160	20200630	home.education	cbse course cut to impact jee neet too
3297161	20200630	city.bengaluru	karnataka may adopt keralas triple lockdown plan
3297162	20200630	city.kanpur	vehicle of up elf team bringing gangster vikas...
3297163	20200630	entertainment.hindi.bollywood	sushant singh rajputs demise fans trend clinfor...
3297164	20200630	entertainment.hindi.bollywood	amitabh bachchans grandson agastya nanda prepp...
3297165	20200630	home.education	icse isc result 2020 when where to check class...
3297166	20200630	city.lucknow	up govt imposes weekend restrictions from tonight
3297167	20200630	gadgets-news	why tiktok removed 1 65 crore videos in india
3297168	20200630	entertainment.hindi.bollywood	apurva asrani calls alia bhatts mother soni ra...
3297169	20200630	entertainment.hindi.bollywood	kangana ranaut gets a doll version of herself ...
3297170	20200630	entertainment.hindi.bollywood	meezaan jaffrey reminisces his childhood days ...
3297171	20200630	entertainment.telugu.movies.news	prabhas20 titled as radhe shyam prabhas and po...

Data Exploration

In [19]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3297172 entries, 0 to 3297171
Data columns (total 3 columns):
 #   Column             Dtype
---  ------             -----
 0   publish_date       int64
 1   headline_category  object
 2   headline_text      object
dtypes: int64(1), object(2)
memory usage: 75.5+ MB
```

In [20]:
```
df1.describe()
```

Out[20]:

	publish_date
count	3.297172e+06
mean	2.012470e+07
std	4.896213e+04
min	2.001010e+07
25%	2.009101e+07
50%	2.013071e+07
75%	2.016110e+07
max	2.020063e+07

In [21]:
```
df1.max()
```

Out[21]:
```
publish_date                                                   20200630
headline_category                                           young-turks
headline_text        ~!The 7 Plexus Knots story in a Galaxy called ...
dtype: object
```

In [22]:
```
df1.min()
```

Out[22]:
```
publish_date                                  20010101
headline_category                     2008-in-pictures
headline_text        15cr funds for two auditoriums
dtype: object
```

In [23]:
```
df1['headline_category'].unique()
```

Out[23]:
```
array(['sports.wwe', 'unknown', 'entertainment.hindi.bollywood', ...,
       'elections.assembly-elections.delhi-assembly-elections',
       'elections.assembly-elections.delhi', 'life-style.coronavirus'],
      dtype=object)
```

In [24]:
```
#In case of null values

df1.isnull().any()
```

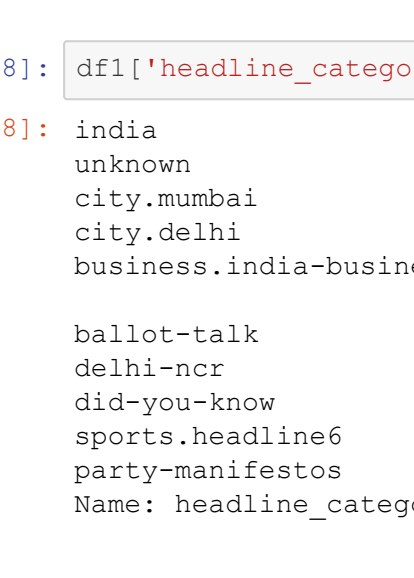
Out[24]:
```
publish_date         False
headline_category    False
headline_text        False
dtype: bool
```

In [25]:
```
df1.corr()
```

Out[25]:

	publish_date
publish_date	1.0

In [26]:
```
sns.set_palette('viridis')
sns.pairplot(df1)
plt.show()
```



In [27]:
```
df1['headline_text'].value_counts()
```

Out[27]:
```
Sunny Leone HOT photos                                                98
Watch top news in one-minute                                          90
Official Site                                                         89
Football: Italian Serie A table                                       88
Football: Spanish La Liga table                                       86
                                                                      ..
Petrapole: on Indo-Bangla border; gateway of terror?                   1
Golf course at Arambol opposed                                         1
Journalist; actor; play writer Cho Ramaswamy passes away               1
Kolkata Police prepare top cop for CBI interview with list of 80-100 questions    1
Only Congress can create Vidarbha state: Ranjit                        1
Name: headline_text, Length: 3082589, dtype: int64
```

In [28]:
```
df1['headline_category'].value_counts()
```

Out[28]:
```
india                    285619
unknown                  207732
city.mumbai              132649
city.delhi               124658
business.india-business  115246
                          ...
ballot-talk                   7
delhi-ncr                     7
did-you-know                  6
sports.headline6              3
party-manifestos              2
Name: headline_category, Length: 1016, dtype: int64
```

Exploratory data analysis using NLP and NLTK tools

In [29]:
```
df1['headline_text'].str.len().hist()
plt.show()
```

This histogram shows that news headlines range from 10 to 70 characters and generally, it is between 25 to 55 characters. now, we will move on to data exploration at a word-level. let's plot the number of words appearing in each news headline.

In [30]:
```
df1['headline_text'].str.split().\
    apply(lambda x : [len(i) for i in x]).\
    map(lambda x: np.mean(x)).hist()
plt.show()
```

The average word length ranges between 1 to 18 with 10 being the most common length. Does it mean that people are using really short words in news headlines? Lets find out. One reason, why this may not be true is stopwords. Stopwords are the words that are most commonly used in any language such as "the", "an", "a" etc. As these words probably small in length these words may have caused the above graph to be left-skewed.

Analyzing the amount and the types of stopwords can give us some good insights into the data. To get the corpus containing stopwords you can use the nltk library. Nltk contains stopwords from many languages, since we are only dealing with english news. I will filter the english stopwords from the corpus

In [31]:
```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
stop = set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

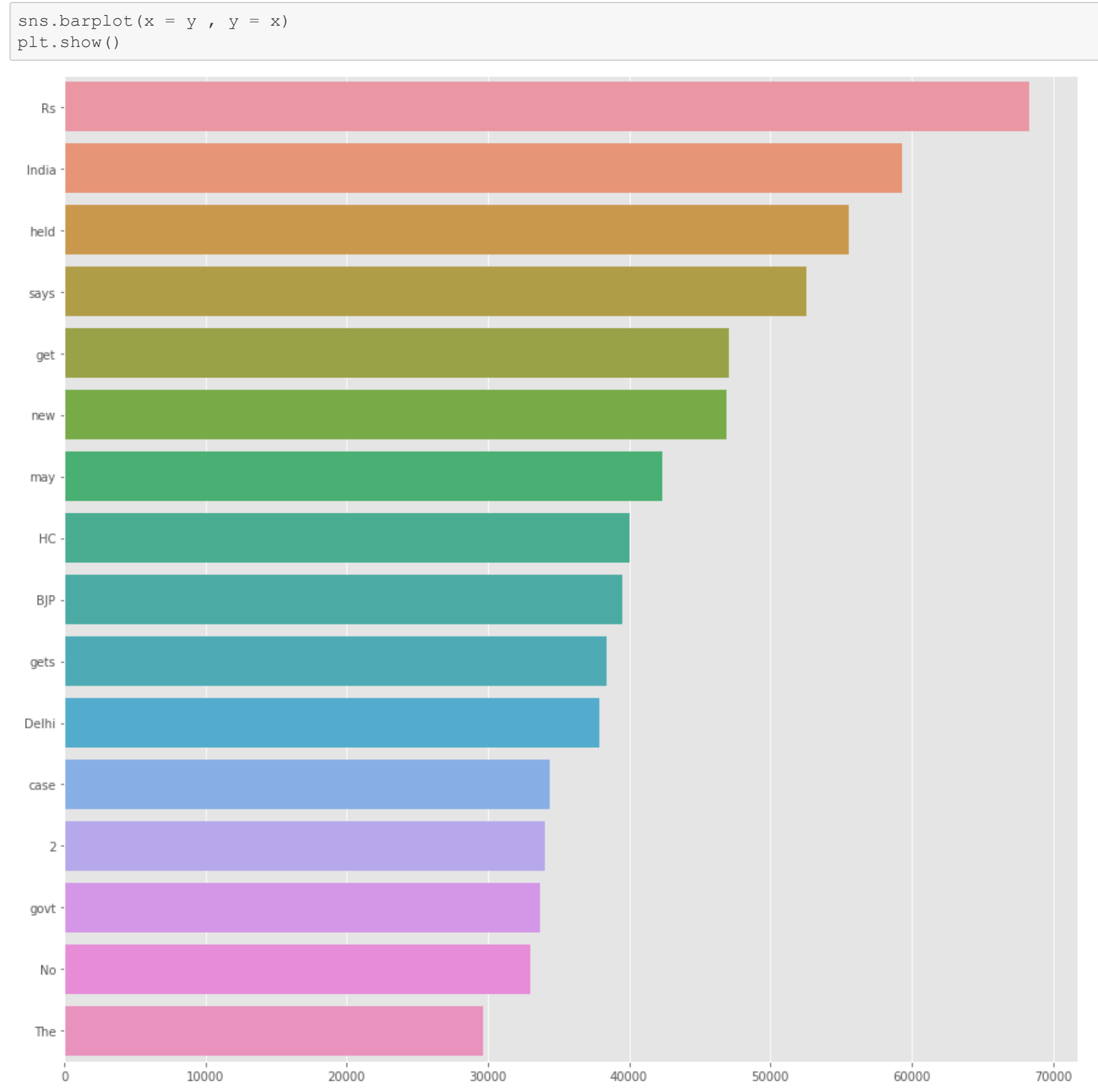
In [42]:
```
corpus = []
new = df1['headline_text'].str.split()
new = new.values.tolist()
corpus = [word for i in new for word in i]
```

In [33]:
```
from collections import defaultdict

dict = defaultdict(int)
for word in corpus:
    if word in stop:
        dict[word]+=1
```

Now we know which stopwords occur frquently in our text, let's inspect which words other than these stopwords occur frequently. We will use the counter function from the collection library to count and store the occurrences of each word in a list of tupples.

In [34]:
```
from collections import Counter

counter = Counter(corpus)
most = counter.most_common()
x, y = [], []
for word, count in most[:40]:
    if (word not in stop):
        x.append(word)
        y.append(count)
```

In [35]:
```
sns.barplot(x = y , y = x)
plt.show()
```



wow! the "Rs", "India" and "held" dominate the headlines over the last 15 year but when we observe other words in the graph they are all related to the RS - India

Ngram Exploration

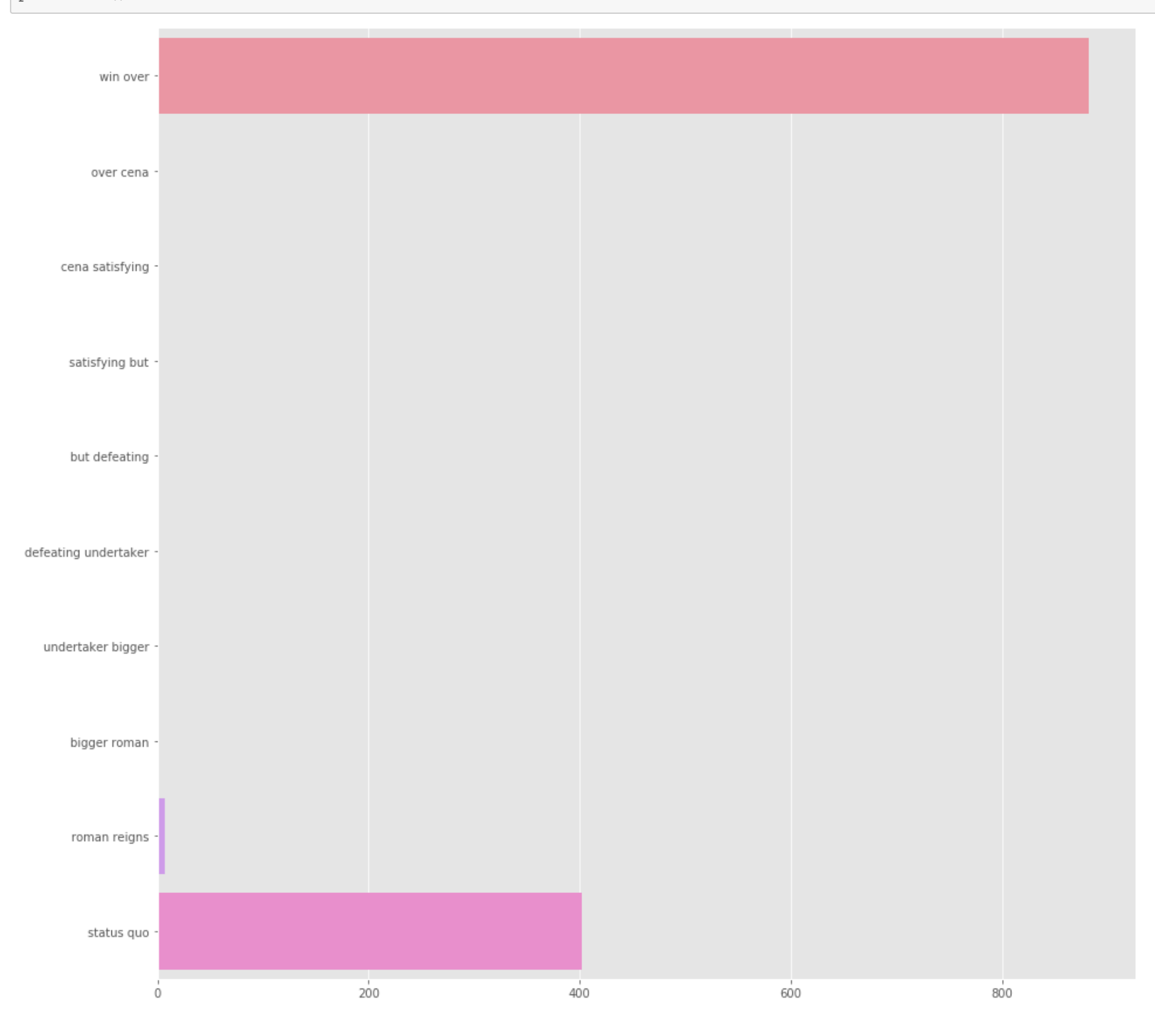
Ngram are simply contiguous sequences of n words. For example "riverbank", "The three muskedteers" etc. If the number of words is 2, It is called bigram. For 3 words it is called a trigram and so on.

In [36]:
```
from nltk.util import ngrams
list(ngrams(['I' , 'went', 'to', 'the', 'river', 'bank'], 2))
```

Out[36]:
```
[('I', 'went'),
 ('went', 'to'),
 ('to', 'the'),
 ('the', 'river'),
 ('river', 'bank')]
```

In [37]:
```
from sklearn.feature_extraction.text import CountVectorizer
def get_top_ngram(corpus, n = None):
    vec = CountVectorizer(ngram_range = (n , n)).fit(corpus)
    bag_of_words = vec.transform(corpus)
    sum_words = bag_of_words.sum(axis = 0)
    words_frq = [(word, sum_words[0, idx])
                 for word, idx in vec.vocabulary_.items()]
    words_feq = sorted(words_frq, key = lambda x : x[1], reverse = True)
    return words_frq[:10]
```

In [46]:
```
top_n_bigrams = get_top_ngram(df1['headline_text'], 2)[:10]
x,y = map(list, zip(*top_n_bigrams))
sns.barplot(x = y, y = x)
plt.show()
```



Textblob

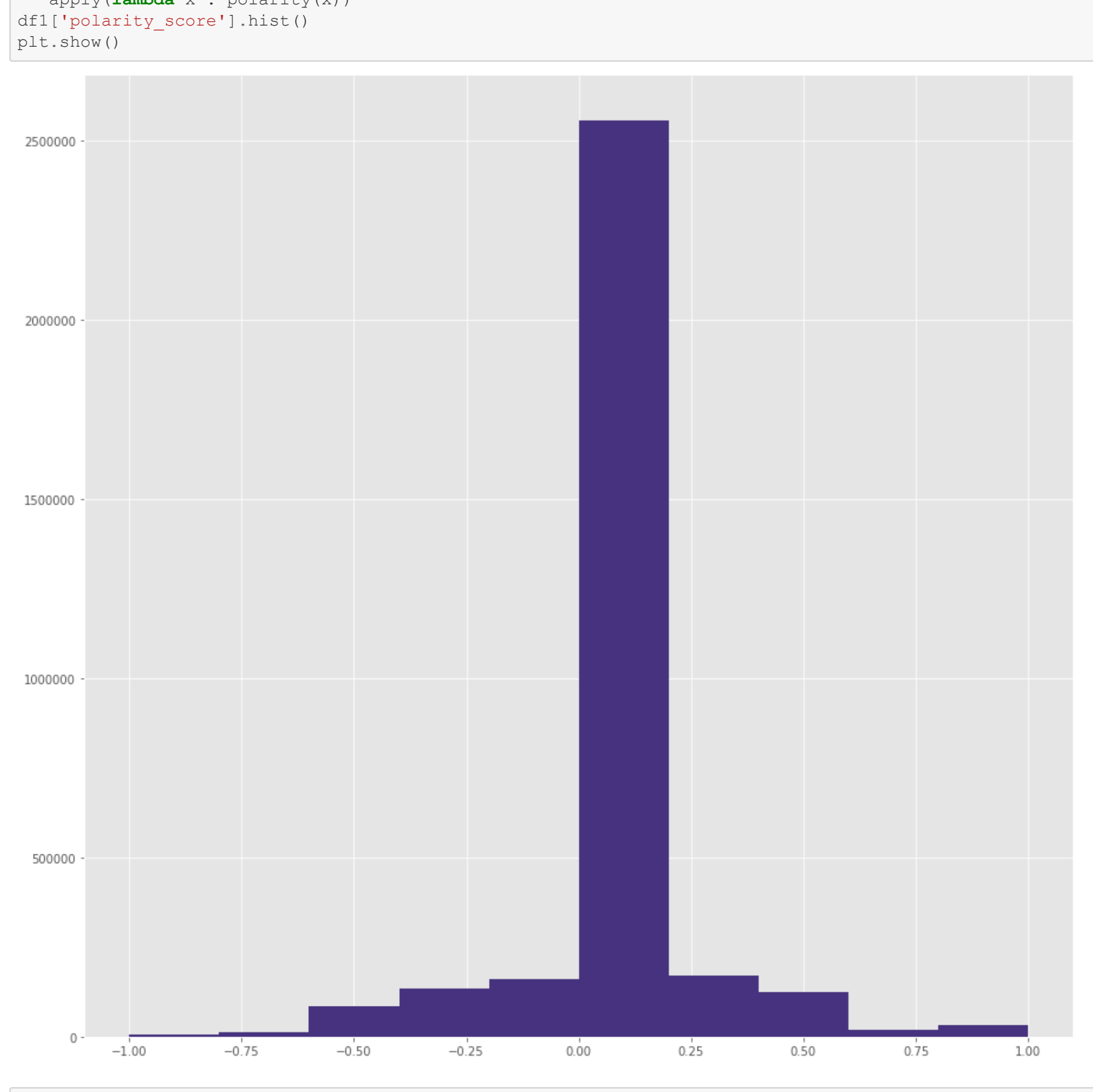
In [43]:
```
!pip install textblob
```

```
Requirement already satisfied: textblob in c:\users\hp\anaconda3\lib\site-packages (0.15.3)
Requirement already satisfied: nltk>=3.1 in c:\users\hp\anaconda3\lib\site-packages (from textblob) (3.4.5)
Requirement already satisfied: six in c:\users\hp\anaconda3\lib\site-packages (from nltk>=3.1->textblob) (1.14.0)
```

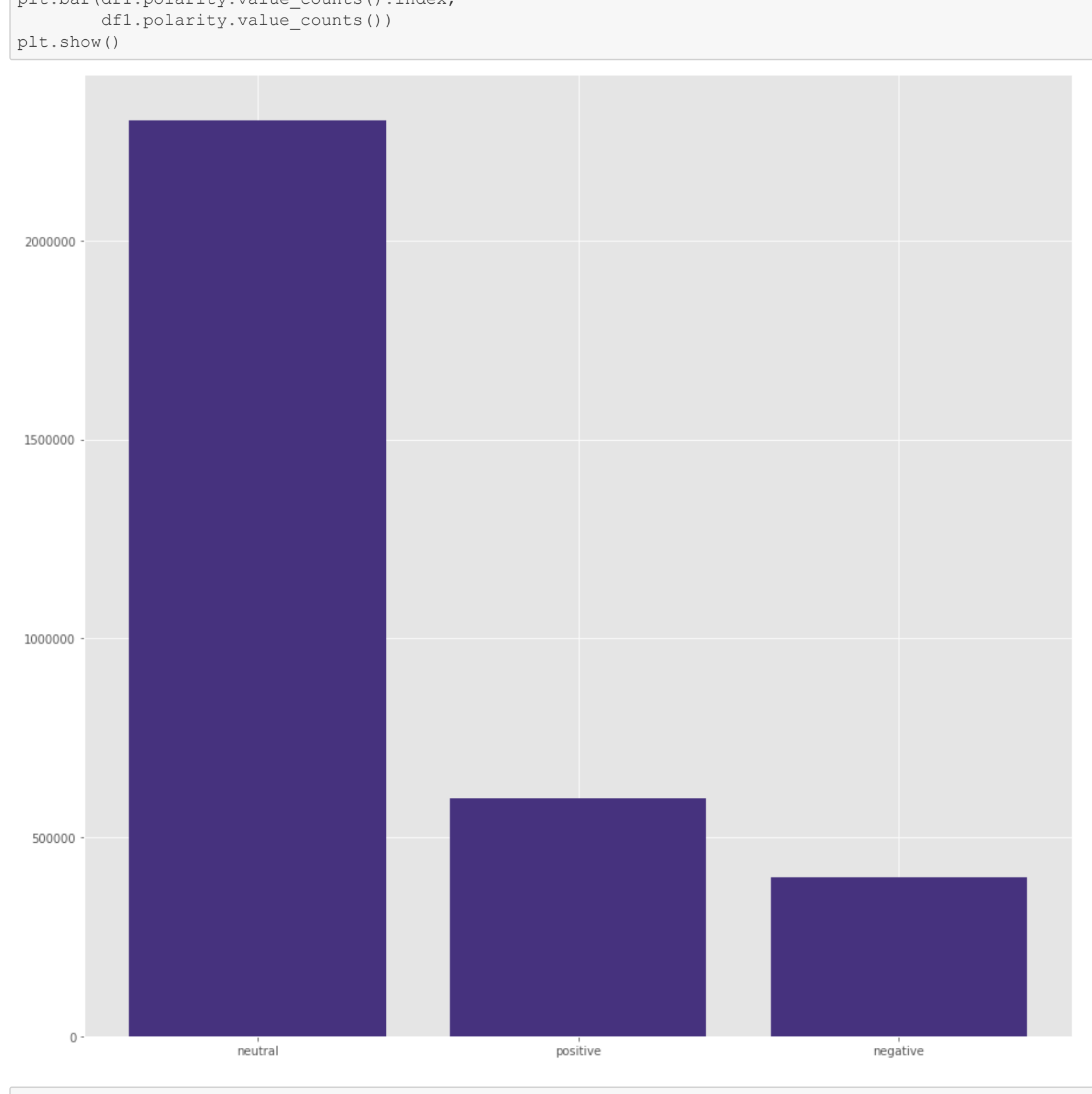
In [40]:
```
from textblob import TextBlob
TextBlob('100 people killed in Iraq').sentiment
```

Out[40]:
```
Sentiment(polarity=-0.2, subjectivity=0.0)
```

In [41]:
```
def polarity(text):
    return TextBlob(text).sentiment.polarity
df1['polarity_score'] = df1['headline_text'].\
   apply(lambda x : polarity(x))
df1['polarity_score'].hist()
plt.show()
```



In [44]:
```
def sentiment(x):
    if x < 0:
        return 'negative'
    elif x == 0:
        return 'neutral'
    else:
        return 'positive'

df1['polarity'] = df1['polarity_score'].\
   map(lambda x : sentiment(x))
plt.bar(df1.polarity.value_counts().index,
        df1.polarity.value_counts())
plt.show()
```



In [45]:
```
df1[df1['polarity'] == 'positive']['headline_text'].head()
```

Out[45]:
```
0     win over cena satisfying but defeating underta...
5                  Extra buses to clear tourist traffic
13          Will Qureshi's return really help the govt?
31                 Extra buses to clear tourist traffic
39          Will Qureshi's return really help the govt?
Name: headline_text, dtype: object
```

In [48]:
```
df1[df1['polarity'] == 'negative']['headline_text'].head()
```

Out[48]:
```
66         Destroying myths and doubts on sexuality
87         Powerless north India gropes in the dark
105                          10-year-old girl missing
132              Net lottery: A winner or a sucker?
143    Mental illness can pass from parent to child
Name: headline_text, dtype: object
```