ASSIGNMENT DEVELOPMENT REPORT

PREPARED FOR

Vidhika Lonare

VP - Product Development and Strategy, Pucho

PREPARED BY

Vishal Sharma

Intern applicant of Data Analysis and Visualization

EXECUTIVE SUMMARY

LIBRARIES USED:

- Pandas
- Ploty
- Matplotlib
- Scikit-learn
- Numpy
- textblob

Programming Language: Python3

Dataset :
fivethirtyeight-presidential-commencement-speeches

Dataset Format : CSV File

Dataset File Name:
transcripts.csv,commencement_speeches.csv

HOW I SOLVE THE ASSIGNMENT?

Dataset file have to two column :

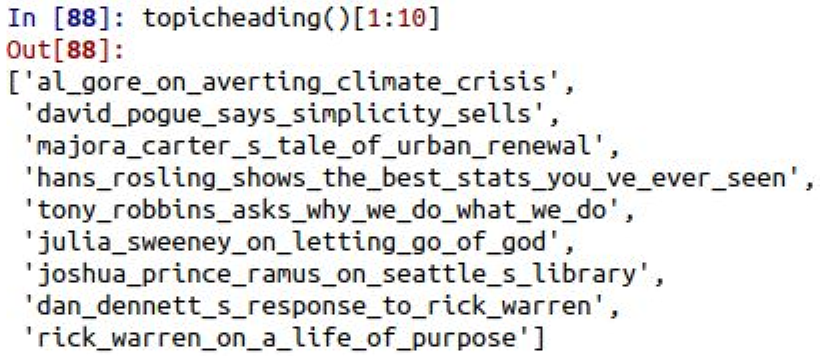
- **url**
- **transcript**

QUES:How I get the Name of the Speech from URL ?

Ans : For getting the name of speech, i created a **topicheading()** function to clean the url. I have used **replace()** function for cleaning purpose. **topicheading()** return the list of name of speech after the whole process gets over.

For testing the result, i used topicheading()[1:10] to see the names of the speech whether it is cleared or not.

ILLUSTRATION:

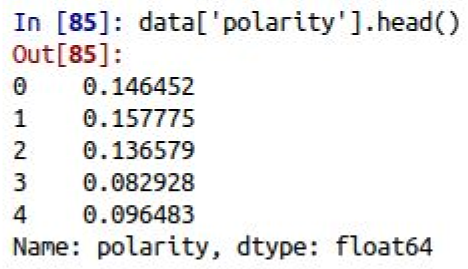


```
In [88]: topicheading()[1:10]
Out[88]:
['al_gore_on_averting_climate_crisis',
 'david_pogue_says_simplicity_sells',
 'majora_carter_s_tale_of_urban_renewal',
 'hans_rosling_shows_the_best_stats_you_ve_ever_seen',
 'tony_robbins_asks_why_we_do_what_we_do',
 'julia_sweeney_on_letting_go_of_god',
 'joshua_prince_ramus_on_seattle_s_library',
 'dan_dennett_s_response_to_rick_warren',
 'rick_warren_on_a_life_of_purpose']
```

Ques: How does I find out the sentiment polarity distribution?

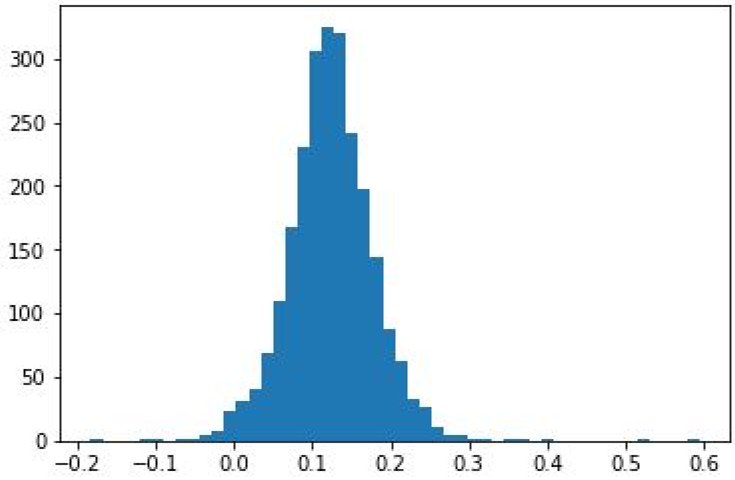
Ans: *For the this purpose, I have used* **textblob** *library to find out sentiment polarity distribution.*

Illustration of Result:



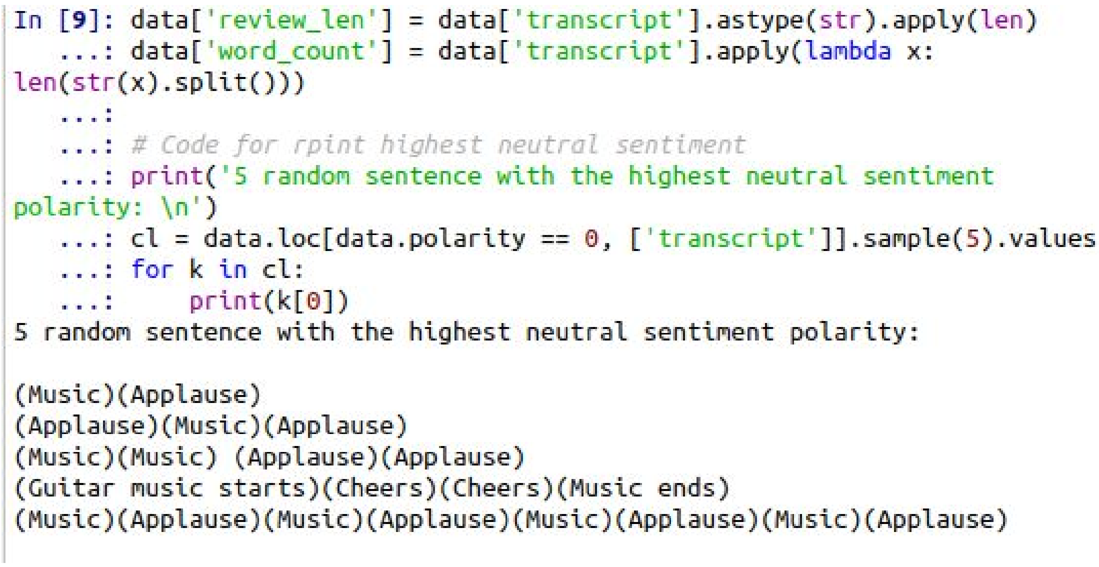
```
In [85]: data['polarity'].head()
Out[85]:
0    0.146452
1    0.157775
2    0.136579
3    0.082928
4    0.096483
Name: polarity, dtype: float64
```

BAR CHART FOR SENTIMENT POLARITY DISTRIBUTION



Polarity range: -1 to +1

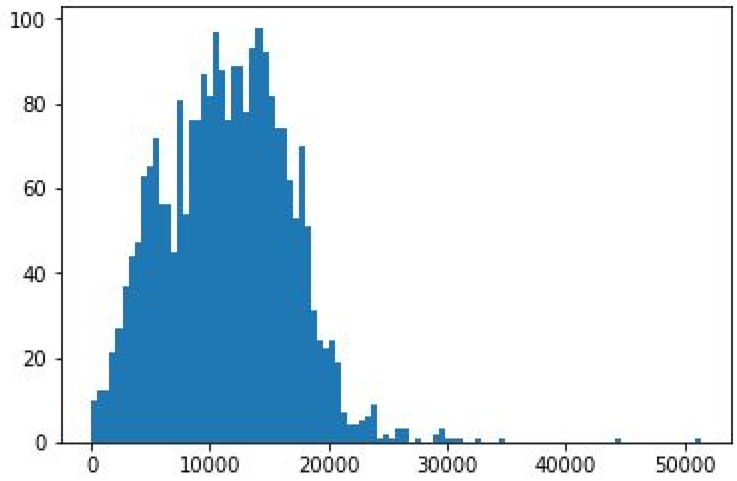
FIVE RANDOM SENTENCE WITH THE HIGHEST NEUTRAL SENTIMENT POLARITY



```
In [9]: data['review_len'] = data['transcript'].astype(str).apply(len)
   ...: data['word_count'] = data['transcript'].apply(lambda x:
len(str(x).split()))
   ...:
   ...: # Code for rpint highest neutral sentiment
   ...: print('5 random sentence with the highest neutral sentiment
polarity: \n')
   ...: cl = data.loc[data.polarity == 0, ['transcript']].sample(5).values
   ...: for k in cl:
   ...:     print(k[0])
5 random sentence with the highest neutral sentiment polarity:

(Music)(Applause)
(Applause)(Music)(Applause)
(Music)(Music) (Applause)(Applause)
(Guitar music starts)(Cheers)(Cheers)(Music ends)
(Music)(Applause)(Music)(Applause)(Music)(Applause)(Music)(Applause)
```

GRAPH FOR SPEECH TEXT LENGTH DISTRIBUTION



GRAPH FOR SPEECH TEXT WORD COUNT DISTRIBUTION

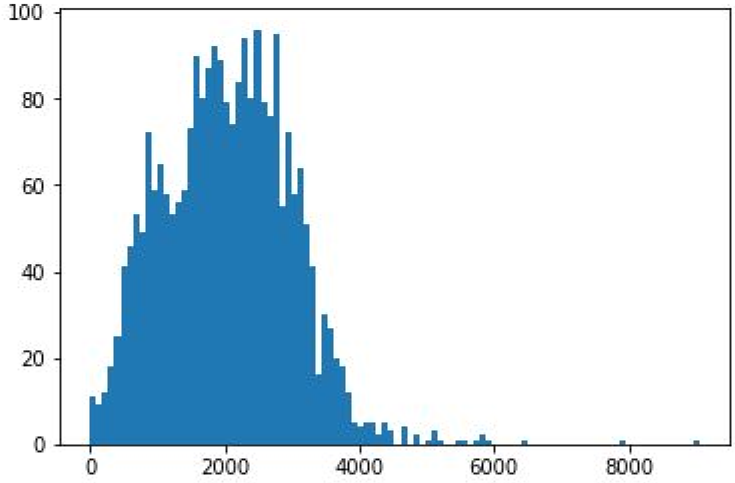
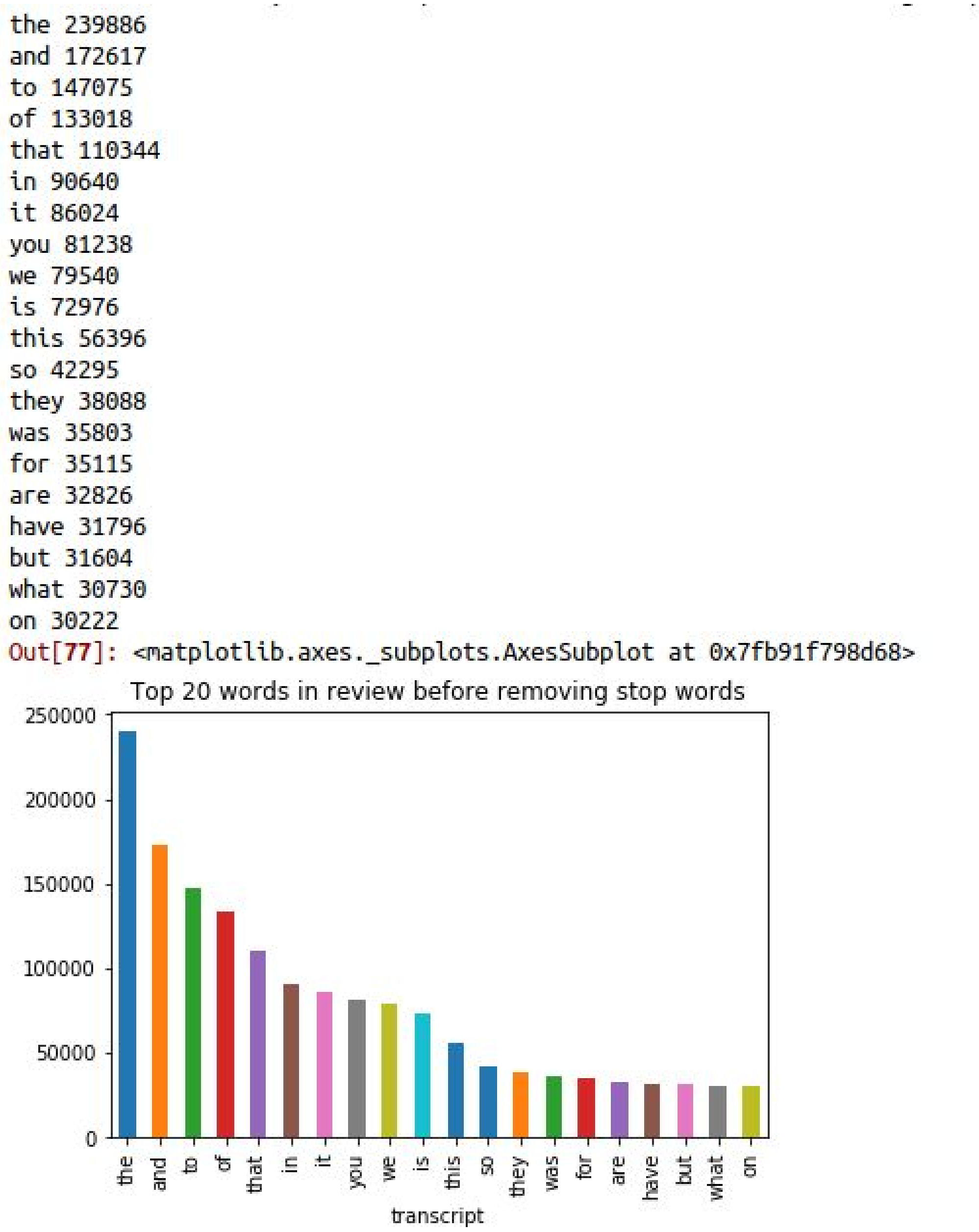
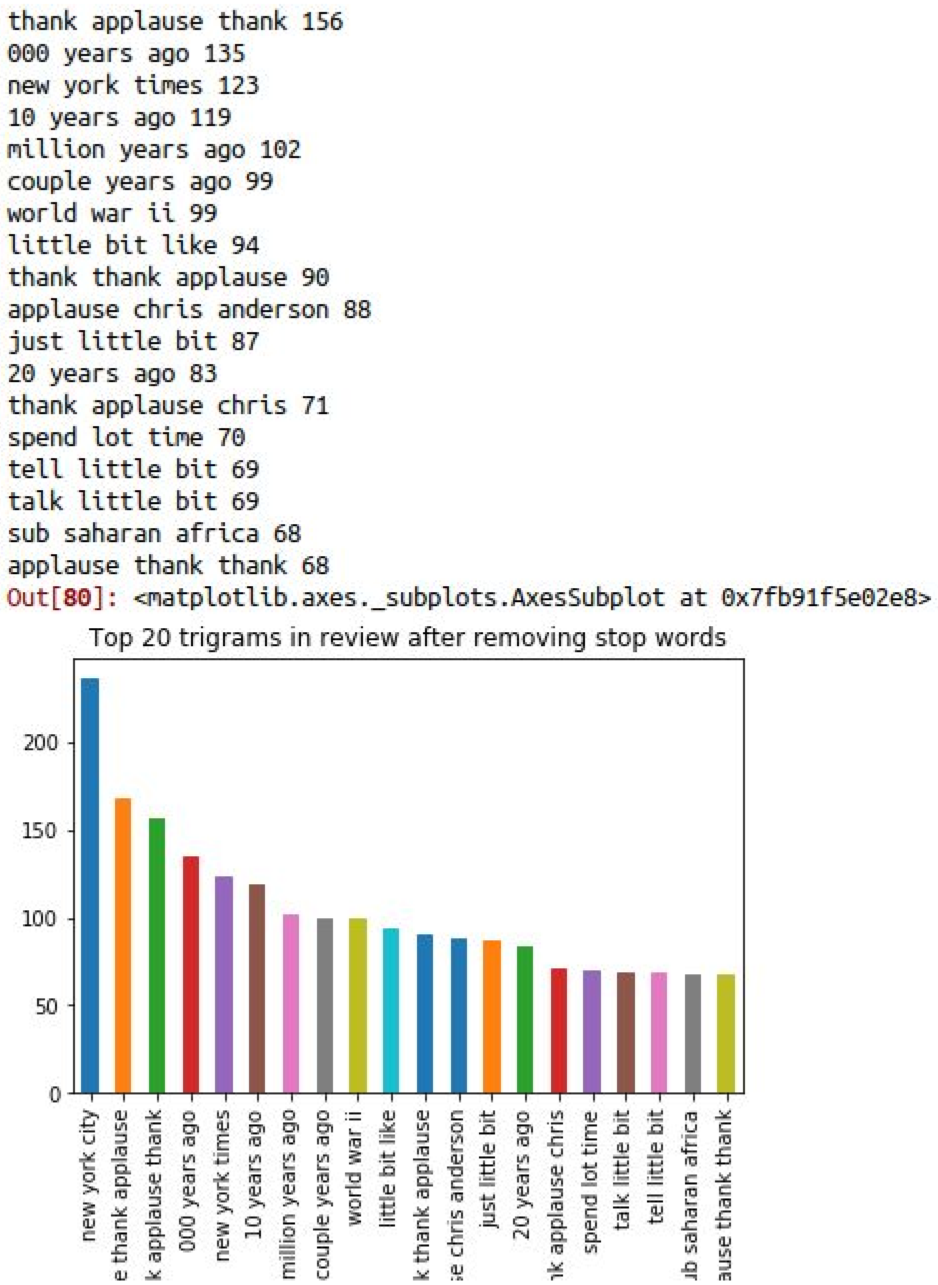


ILLUSTRATION OF TOP 20 WORDS IN REVIEW BEFORE REMOVING STOP WORDS



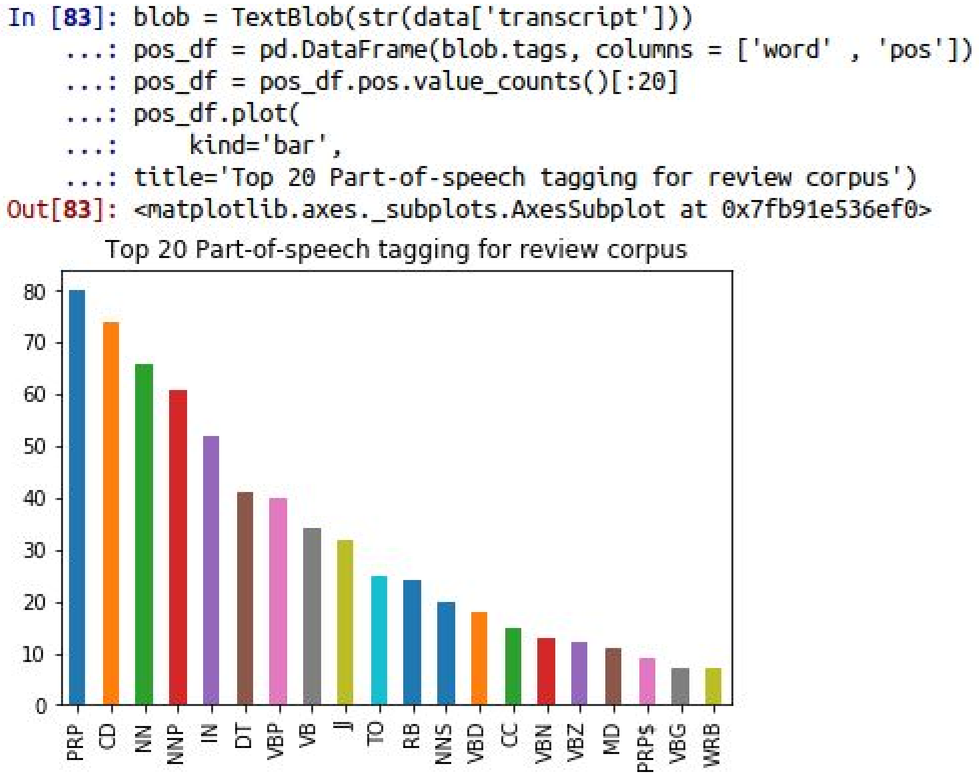
```
the 239886
and 172617
to 147075
of 133018
that 110344
in 90640
it 86024
you 81238
we 79540
is 72976
this 56396
so 42295
they 38088
was 35803
for 35115
are 32826
have 31796
but 31604
what 30730
on 30222
Out[77]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb91f798d68>
```

ILLUSTRATION OF TOP 20 TRIGRAMS IN REVIEW AFTER REMOVING STOP WORDS



```
thank applause thank 156
000 years ago 135
new york times 123
10 years ago 119
million years ago 102
couple years ago 99
world war ii 99
little bit like 94
thank thank applause 90
applause chris anderson 88
just little bit 87
20 years ago 83
thank applause chris 71
spend lot time 70
tell little bit 69
talk little bit 69
sub saharan africa 68
applause thank thank 68
Out[80]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb91f5e02e8>
```

ILLUSTRATION OF TOP 20 PART-OF-SPEECH TAGGING FOR REVIEW CORPUS



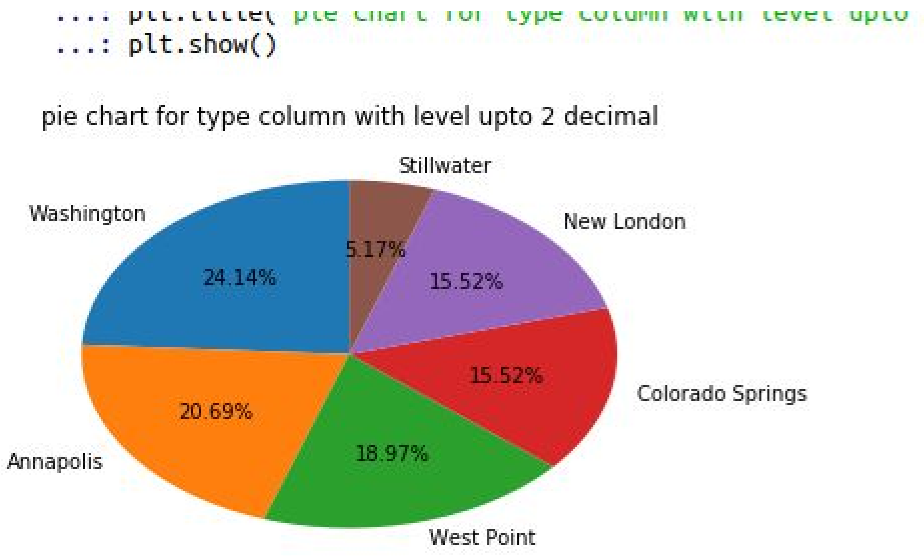
```
In [83]: blob = TextBlob(str(data['transcript']))
    ...: pos_df = pd.DataFrame(blob.tags, columns = ['word' , 'pos'])
    ...: pos_df = pos_df.pos.value_counts()[:20]
    ...: pos_df.plot(
    ...:     kind='bar',
    ...: title='Top 20 Part-of-speech tagging for review corpus')
Out[83]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb91e536ef0>
```

QUES: **For which purpose, I Have used scikit-learn library?**

Ans: I have used it for tokenization of sentence of speech using several function of library such as CountVectorizer(),transform(),fit()...etc.

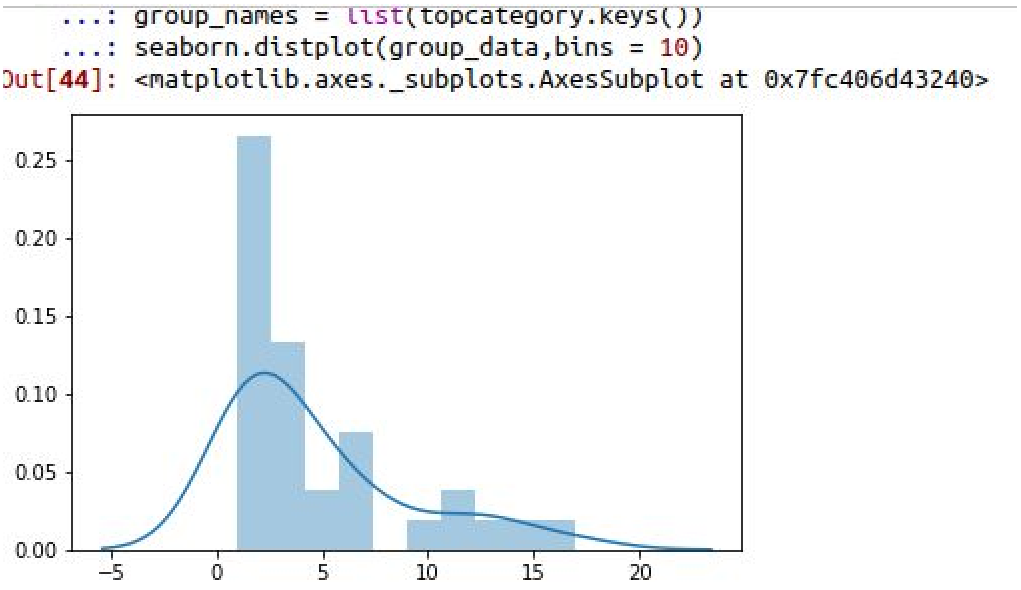
ANALYSIS REPORT FOR commencement_speeches.csv

Pie chart for Representing Top-6 President according to number of speeches



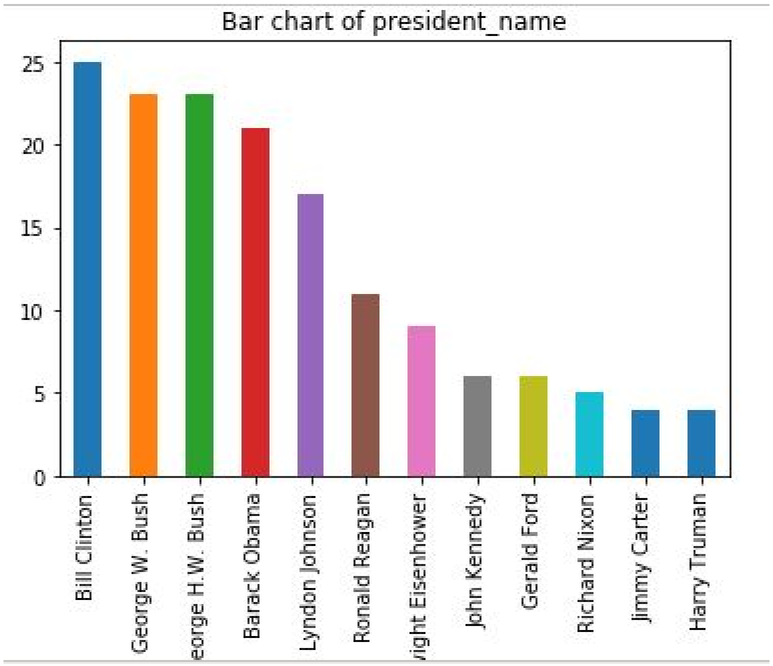
```
   ....: plt.title('pie chart for type column with level upto
   ...: plt.show()
```

Distplot graph for different states of USA

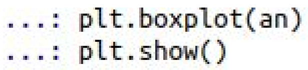


```
   ...: group_names = list(topcategory.keys())
   ...: seaborn.distplot(group_data,bins = 10)
Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc406d43240>
```

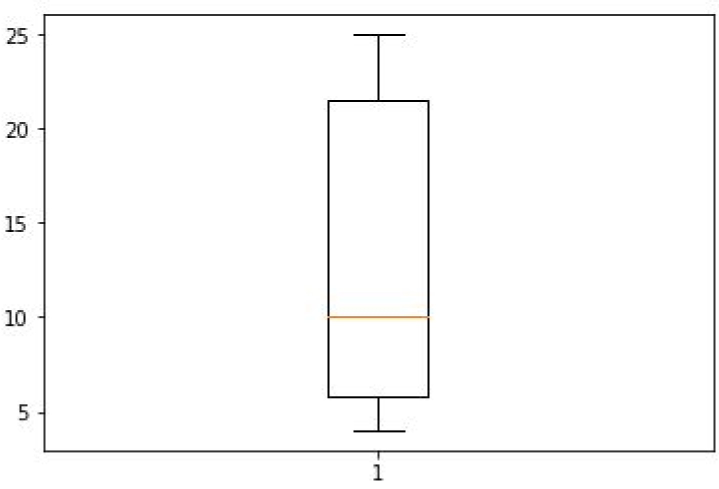
Bar chart for top President according to their number of speech



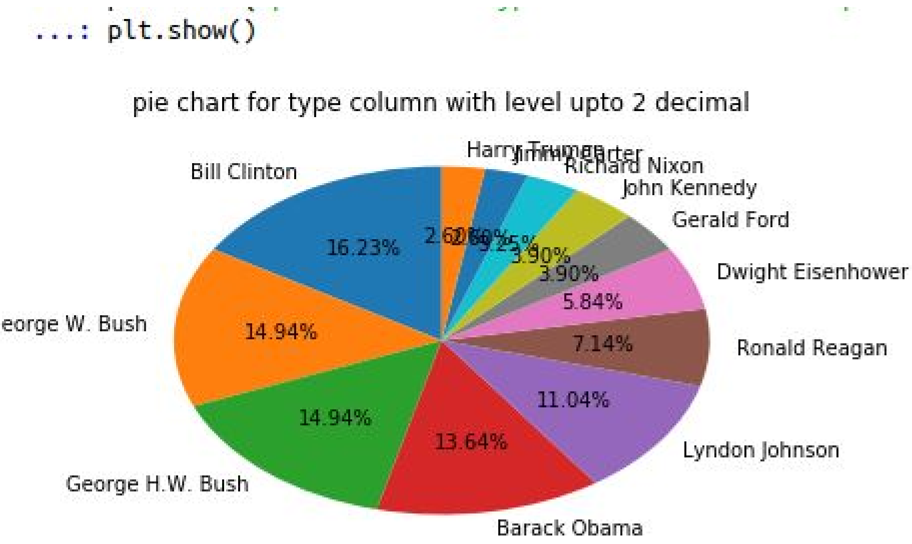
boxplot graph with president_name and president



```
...: plt.boxplot(an)
...: plt.show()
```



Pie Chart for representing president name with their contribution to number of speech



```
   ...: plt.show()
```

HERE MY REPORT ENDS, YOU CAN SEE WHOLE CODE ON GOOGLE DRIVE LINK WITH DATASET.

THANK YOU