Question 3

Feed the following paragraph into your favourite data analytics tool, and answer the following;

''' As a term, data analytics predominantly refers to an assortment of applications, from basic business intelligence (BI), reporting and online analytical processing (OLAP) to various forms of advanced analytics. In that sense, it's similar in nature to business analytics, another umbrella term for approaches to analyzing data -- with the difference that the latter is oriented to business uses, while data analytics has a broader focus. The expansive view of the term isn't universal, though: In some cases, people use data analytics specifically to mean advanced analytics, treating BI as a separate category. Data analytics initiatives can help businesses increase revenues, improve operational efficiency, optimize marketing campaigns and customer service efforts, respond more quickly to emerging market trends and gain a competitive edge over rivals -- all with the ultimate goal of boosting business performance. Depending on the particular application, the data that's analyzed can consist of either historical records or new information that has been processed for real-time analytics uses. In addition, it can come from a mix of internal systems and external data sources. At a high level, data analytics methodologies include exploratory data analysis (EDA), which aims to find patterns and relationships in data, and confirmatory data analysis (CDA), which applies statistical techniques to determine whether hypotheses about a data set are true or false. EDA is often compared to detective work, while CDA is akin to the work of a judge or jury during a court trial -- a distinction first drawn by statistician John W. Tukey in his 1977 book Exploratory Data Analysis. Data analytics can also be separated into quantitative data analysis and qualitative data analysis. The former involves analysis of numerical data with quantifiable variables that can be compared or measured statistically. The qualitative approach is more interpretive -- it focuses on understanding the content of non-numerical data like text, images, audio and video, including common phrases, themes and points of view. '''

- a. What is the probability of the word “data” occurring in each line ?
- b. What is the distribution of distinct word counts across all the lines ?
- c. What is the probability of the word “analytics” occurring after the word “data” ?

Question **3.a** and **3.c** are almost similar but **3.c** bigrams must be applied.

a. What is the probability of the word “data” occurring in each line ?

- 1. investigate how many the word "data".
- 2. check the number of lines
- 3. find the number of times the word "data" appears in the text.
- 4. calculate the probability

```
# Start with importing libraries
import re
from collections import Counter
import pandas as pd
import nltk
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

```
given_string = """As a term, data analytics predominantly refers to an assortment of applications, from basic \nbusiness intelligence (BI), reporting
print(given_string)
```

```
    As a term, data analytics predominantly refers to an assortment of applications, from basic
    business intelligence (BI), reporting and online analytical processing (OLAP) to various forms of
    advanced analytics. In that sense, it's similar in nature to business analytics, another umbrella
    term for approaches to analyzing data -- with the difference that the latter is oriented to
    business uses, while data analytics has a broader focus. The expansive view of the term isn't
    universal, though: In some cases, people use data analytics specifically to mean advanced
    analytics, treating BI as a separate category. Data analytics initiatives can help businesses
    increase revenues, improve operational efficiency, optimize marketing campaigns and customer
    service efforts, respond more quickly to emerging market trends and gain a competitive edge
    over rivals -- all with the ultimate goal of boosting business performance. Depending on the
    particular application, the data that's analyzed can consist of either historical records or new
    information that has been processed for real-time analytics uses. In addition, it can come from a
    mix of internal systems and external data sources. At a high level, data analytics methodologies
    include exploratory data analysis (EDA), which aims to find patterns and relationships in data,
    and confirmatory data analysis (CDA), which applies statistical techniques to determine whether
    hypotheses about a data set are true or false. EDA is often compared to detective work, while
    CDA is akin to the work of a judge or jury during a court trial -- a distinction first drawn by
    statistician John W. Tukey in his 1977 book Exploratory Data Analysis. Data analytics can also
    be separated into quantitative data analysis and qualitative data analysis. The former involves
    analysis of numerical data with quantifiable variables that can be compared or measured
    statistically. The qualitative approach is more interpretive -- it focuses on understanding the
    content of non-numerical data like text, images, audio and video, including common phrases,
    themes and points of view.
```

```
given_string = given_string.lower()
```

```
line_list = given_string.splitlines()
```

```
nlines = len(line_list)
nlines
```

```
nlines
```

```
23
```

```
def tokenize(string):
    return re.compile('\w+').findall(string)

def word_freq(string):
    text = tokenize(string.lower())
    c = Counter(text)           # count the words
    d = Counter(''.join(text))  # count all letters
    return (dict(c))    # return a tuple of counted words and letters

def return_word_freq(string):
    text = tokenize(string.lower())
    c = Counter(text)           # count the words
    d = Counter(''.join(text))  # count all letters
    return (dict(c), dict(d))    # return a tuple of counted words and letters

words = word_freq(given_string) # count and get dicts with counts
words_, letters = return_word_freq(given_string)

sumWords = sum(words.values())       # sum total words
sumLetters = sum(letters.values())   # sum total letters
```

```
len(words)
```

```
194
```

```
{k:v for (k,v) in words.items() if v>8 }
```

```
{'a': 10, 'analytics': 10, 'and': 9, 'data': 18, 'of': 10, 'the': 11, 'to': 11}
```

```
# The probability of an event A is the number of ways event A can occur divided by the total number of possible outcomes.
data_count = words['data']
analytics_count = words['analytics']
```

```
probablity_of_data_appearing_in_every_line = data_count/nlines
```

```
probablity_of_data_appearing_in_every_line
```

```
0.782608695652174
```

```
# GOT IT ! 0.78 it is......
```

```
probablity_of_data_appearing_in_full_text = data_count/sumWords
```

```
probablity_of_data_appearing_in_full_text
```

```
0.05625
```

B. Now figure out this -> What is the distribution of distinct word counts across all the lines ?

- 1. We shall first figure out the distinct word counts in every line,
- 2. Create a histogram, bar plot or something similar to see the distribution.

```
line_unique_counts = []
for line in line_list:
    line_unique_counts.append(len(word_freq(line)))
```

```
line_unique_counts
```

```
[14,
 13,
 13,
 13,
 16,
 13,
 12,
 10,
 14,
 13,
 15,
 17,
 14,
 13,
 12,
 17,
```

```
        18,
        14,
        11,
        13,
        11,
        14,
        5]
```

```
df_unique_words = pd.DataFrame(line_unique_counts)
```

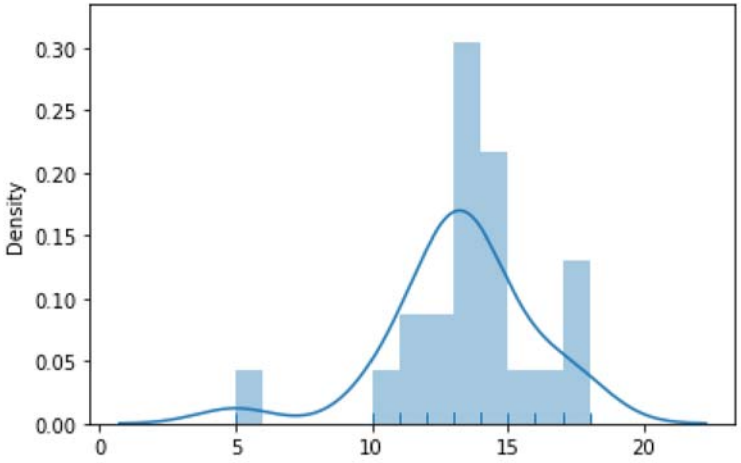
```
df_unique_words.rename(columns={0: "Unique_Words"})
```

	Unique_Words
0	14
1	13
2	13
3	13

```
import seaborn as sns
```

```
sns.distplot(df_unique_words, kde=True, rug=True);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2056: FutureWarning: The `axis` variable is no longer used and will be removed.
  warnings.warn(msg, FutureWarning)
```




```
# NORMALLY DISTRIBUTED - to an extent.
```

20	11

C. Finally time to figure out this -> What is the probability of the word “analytics” occurring after the word “data” ?

- 1. draw the bigrams,

- 2. determine their counts
- 3. find "data analytics" and only "analytics" counts
- 4. calculate the probability.

```
# import nltk
# import re
# from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
# from nltk.corpus import stopwords
# from nltk.tokenize import word_tokenize
# import pandas as pd

# Getting bigrams
vectorizer = CountVectorizer(ngram_range =(2, 2))
X1 = vectorizer.fit_transform(line_list)
features = (vectorizer.get_feature_names())
# print("\n\nX1 : \n", X1.toarray())

# Applying TFIDF
# You can still get n-grams here
vectorizer = TfidfVectorizer(ngram_range = (2, 2))
X2 = vectorizer.fit_transform(line_list)
scores = (X2.toarray())
# print("\n\nScores : \n", scores)

# Getting top ranking features
sums = X2.sum(axis = 0)
data1 = []
for col, term in enumerate(features):
    data1.append( (term, sums[0, col] ))
ranking = pd.DataFrame(data1, columns = ['term', 'rank'])
words = (ranking.sort_values('rank', ascending = False))
print ("\n\nWords : \n", words.head(7))
```

```
Words :
                  term      rank
80     data analytics  1.121468
79      data analysis  1.074633
100  exploratory data  0.509671
159    numerical data  0.507052
235        themes and  0.500000
```

```
32         and points  0.500000
186         points of  0.500000
```

```
#The probability of an event A is the number of ways event A can occur divided by the total number of possible outcomes.
import re
from collections import Counter
bigramwords = re.findall('\w+', given_string)
bigram_counts = Counter(zip(bigramwords,bigramwords[1:]))
data_analytics_count = bigram_counts[("data", "analytics")]
print(f"Number of times data analytics appear together: {data_analytics_count}")
print(f"Number of times only analytics appear in complete text: {analytics_count}")
```

```
Number of times data analytics appear together: 6
Number of times only analytics appear in complete text: 10
```

```
print(f"Probablity of analytics appearing after data: {data_analytics_count/analytics_count}")
```

```
Probablity of analytics appearing after data: 0.6
```

```

```

```

```

```

```

✓ 0s completed at 23:26