Name : Sarthak Galande
PRN:202401080074
Batch:CS6
Roll No: 84

EDS ACTIVITY NO.1

Dataset : The Blog Authorship Corpus

Problem Statements :-

```python
import pandas as pd
import numpy as np

df = pd.read_csv('blog_authorship_dataset(1).csv')

# 1. Find the total number of records (rows)
total_rows = df.shape[0]
print("1. Total rows:", total_rows)
print('\n')
print('\n')


# 2. Find the total number of unique authors
unique_authors = df['author'].nunique()
print("2. Unique authors:", unique_authors)
print('\n')
print('\n')


# 3. Number of missing values in each column
missing_values = df.isnull().sum()
print("3. Missing values per column:\n", missing_values)
print('\n')
print('\n')


# 4. Top 5 authors with the most blog posts
top_5_authors = df['author'].value_counts().head(5)
print("4. Top 5 authors:\n", top_5_authors)
print('\n')
print('\n')


# 5. Average length of blog titles
title_lengths = df['title'].astype(str).apply(len)
average_title_length = title_lengths.mean()
```

```python
43
44    # 6. Maximum and minimum lengths of content_snippet
45    content_lengths = df['content_snippet'].astype(str).apply(len)
46    print("6. Max content_snippet length:", content_lengths.max())
47    print("6. Min content_snippet length:", content_lengths.min())
48    print('\n')
49    print('\n')
50
51
52
53    # 7. Blog posts with title length > 100
54    long_titles = df[title_lengths > 100]
55    print("7. Posts with title length > 100:\n", long_titles[['id', 'title']])
56    print('\n')
57    print('\n')
58
59
60    # 8. Number of blog posts with empty (NaN) titles
61    empty_titles_count = df['title'].isna().sum()
62    print("8. Number of empty titles:", empty_titles_count)
63    print('\n')
64    print('\n')
65
66
67
68    # 9. Create a new column counting words in content_snippet
69    content_word_counts = df['content_snippet'].astype(str).apply(lambda x: len(x.split()))
70    df['content_snippet_word_count'] = content_word_counts
71    print("9. Word counts added to DataFrame.")
72    print('\n')
73    print('\n')
74
75
76
77    # 10. Top 10 blog posts with the longest content_snippet
78    top_10_content = df.sort_values(by='content_snippet_word_count', ascending=False).head(10)
79    print("10. Top 10 longest content_snippet posts:\n", top_10_content[['id', 'content_snippet_word_count']])
```

```python
     85   # 11. Average number of words per content_snippet for each author
     86   avg_words_per_author = df.groupby('author')['content_snippet_word_count'].mean()
     87   print("11. Average words per content_snippet per author:\n", avg_words_per_author)
     88   print('\n')
     89   print('\n')
     90
     91
     92   # 13. Author who wrote the blog post with the longest content_snippet
     93   longest_content_index = df['content_snippet_word_count'].idxmax()
     94   longest_content_author = df.loc[longest_content_index, 'author']
     95   print("13. Author with longest content_snippet:", longest_content_author)
     96   print('\n')
     97   print('\n')
     98
     99
    100   # 14. Proportion of blog posts by top 5 authors
    101   top5_total_posts = top_5_authors.sum()
    102   proportion_top5 = top5_total_posts / total_rows
    103   print("14. Proportion of posts by top 5 authors:", proportion_top5)
    104   print('\n')
    105   print('\n')
    106
    107
    108   # 15. Mean and standard deviation of title lengths
    109   mean_title_length = title_lengths.mean()
    110   std_title_length = title_lengths.std()
    111   print(f"15. Mean title length: {mean_title_length}, Std Dev title length: {std_title_length}")
    112   print('\n')
    113   print('\n')
    114
    115
    116
    117   # 16. Title with maximum number of words
    118   df['title_word_count'] = df['title'].astype(str).apply(lambda x: len(x.split()))
    119   max_title_words_row = df.loc[df['title_word_count'].idxmax()]
    120   print("16. Title with maximum words:\n", max_title_words_row[['id', 'title', 'title_word_count']])
    121   print('\n')
```

```python
123
124    # 17. List authors who have written more than 50 blog posts
125    authors_post_counts = df['author'].value_counts()
126    authors_more_than_50 = authors_post_counts[authors_post_counts > 50]
127    print("17. Authors with more than 50 posts:\n", authors_more_than_50)
128    print('\n')
129    print('\n')
130
131
132    # 18. Correlation between title length and content_snip word count
133    correlation = np.corrcoef(title_lengths, content_word_counts)[0, 1]
134    print("18. Correlation between title length and content_snip word count:", correlation)
135    print('\n')
136    print('\n')
137
138
139
140    # 19. Number of blog posts containing the word "data" in title
141    posts_with_data = df['title'].astype(str).str.contains('data', case=False, na=False)
142    num_posts_with_data = posts_with_data.sum()
143    print("19. Number of posts mentioning 'data' in title:", num_posts_with_data)
144    print('\n')
145    print('\n')
146
147
148
149    # 20. Find the ID of blog posts where content_snip is exactly empty ("")
150    empty_content_ids = df[df['content_snippet'].astype(str).str.strip() == '']['id']
151    print("20. IDs with empty content_snip:\n", empty_content_ids.values)
152    print('\n')
153    print('\n')
154
155
```

Solutions:-

```
PS C:\Users\Engeers\Desktop\python_practice\python_practice> python -u "c:\Users\Engeers\Desktop\python_practice\python_practice\EDS\EDS_ACT_NO1.py"
1. Total rows: 299


2. Unique authors: 46



3. Missing values per column:
 id                  0
author              0
title               0
content_snippet     0
word_count          0
avg_word_length     0
punctuation_count   0
post_date           0
sentiment           0
topic               0
device_used         0
reading_time_min    0
dtype: int64



4. Top 5 authors:
 author
Gregory       16
Christopher   16
Karen         12
Derek         12
Patrick       11
Name: count, dtype: int64
```

```
Derek         12
Patrick       11
Name: count, dtype: int64


5. Average title length: 29.401337792642142


6. Max content_snippet length: 99
6. Min content_snippet length: 14


7. Posts with title length > 100:
 Empty DataFrame
Columns: [id, title]
Index: []


8. Number of empty titles: 0


9. Word counts added to DataFrame.


10. Top 10 longest content_snippet posts:
      id  content_snippet_word_count
90    91                          17
73    74                          17
6      7                          16
54    55                          16
27    28                          16
142  143                          16
156  157                          16
166  167                          16
189  190                          16
36    37                          15


11. Average words per content_snippet per author:
 author
Aaron          12.250000
```

```
11. Average words per content_snippet per author:
 author
Aaron          12.250000
Alex            8.000000
Amber          11.125000
Andrew         12.000000
Anthony        10.000000
Bernard        11.444444
Brianna        12.500000
Brittney       10.500000
Carolyn         8.250000
Christopher    11.875000
Colleen        10.600000
Crystal        13.500000
Danielle        9.444444
Denise         11.428571
Derek          10.666667
Erika          11.250000
Felicia         7.800000
George         10.333333
Grant          13.333333
Gregory        10.375000
Jacob          11.000000
Jessica         8.000000
John            9.000000
Jon             9.500000
Joseph         10.625000
Joshua         10.000000
Kara           11.888889
Karen          12.083333
Katherine      10.428571
Kenneth        11.500000
Kimberly       12.090909
Mark           10.000000
Michelle       11.285714
Nicole         11.250000
Null           13.500000
Patrick        10.363636
Paul           13.333333
Peggy          12.800000
```

```
Nicole         11.250000
Null           13.500000
Patrick        10.363636
Paul           13.333333
Peggy          12.800000
Robert         11.428571
Ryan            9.222222
Sandra         10.800000
Sharon         12.222222
Stephen        10.750000
Tammy          11.142857
Theodore       12.428571
Tyler           9.000000
Name: content_snippet_word_count, dtype: float64


13. Author with longest content_snippet: Robert


14. Proportion of posts by top 5 authors: 0.22408026755852842


15. Mean title length: 29.401337792642142, Std Dev title length: 8.491970045359619



16. Title with maximum words:
 id                                                    135
title             Until interest investment sister TV heart such.
title_word_count                                        7
Name: 134, dtype: object



17. Authors with more than 50 posts:
 Series([], Name: count, dtype: int64)
```

14. Proportion of posts by top 5 authors: 0.22408026755852842

15. Mean title length: 29.401337792642142, Std Dev title length: 8.491970045359619

16. Title with maximum words:
 id   135
title   Until interest investment sister TV heart such.
title_word_count   7
Name: 134, dtype: object

17. Authors with more than 50 posts:
 Series([], Name: count, dtype: int64)

18. Correlation between title length and content_snip word count: 0.09305850521353563

19. Number of posts mentioning 'data' in title: 2

20. IDs with empty content_snip:
 []

PS C:\Users\Engeers\Desktop\python_practice\python_practice>