Name : Diya Rathod
Roll No: ET2-41

PRN No : 202401070144

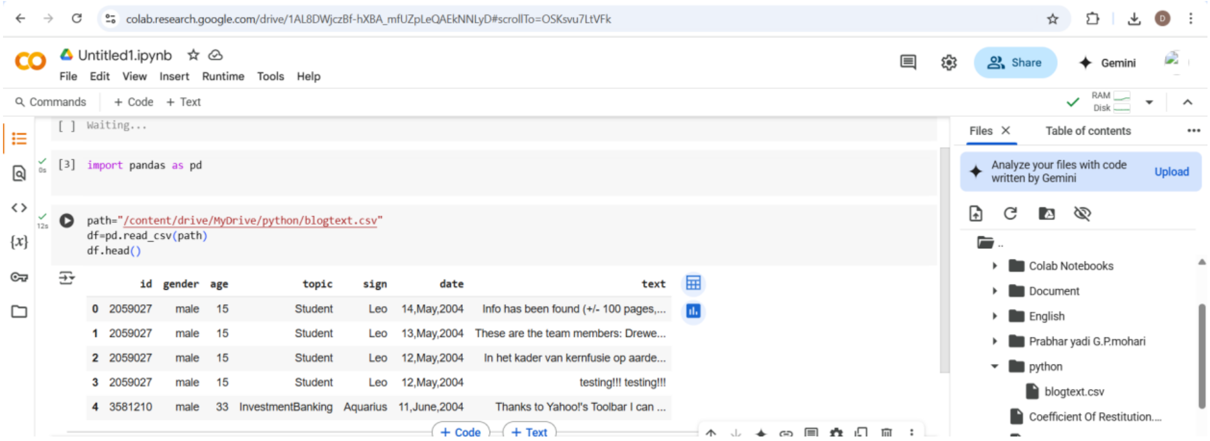
Class: ET2

SUB: EDS

Dataset :

The Blog Authorship Corpus

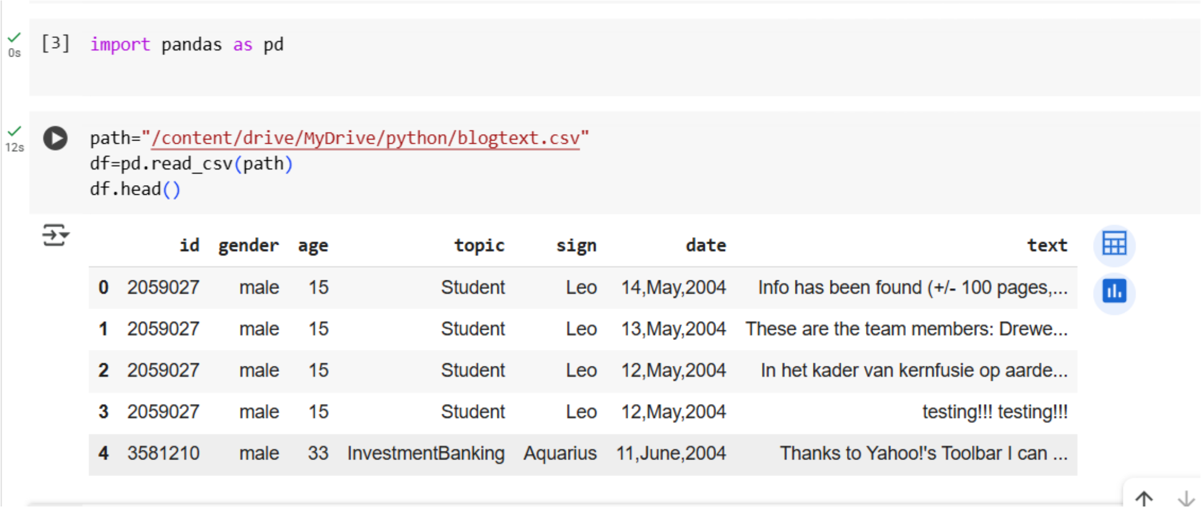
Read Dataset:


colab.research.google.com/drive/1AL8DWjczBf-hXBA_mfUZpLeQAEkNNLyD#scrollTo=OSKsvu7LtVFk
Untitled1.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Share
Gemini
RAM
Disk
Files
Table of contents
Analyze your files with code written by Gemini
Upload
Colab Notebooks
Document
English
Prabhar yadi G.P.mohari
python
blogtext.csv
Coefficient Of Restitution....
+ Code
+ Text

```
[ ] Waiting...
[3] import pandas as pd
```

```
path="/content/drive/MyDrive/python/blogtext.csv"
df=pd.read_csv(path)
df.head()
```

	id	gender	age	topic	sign	date	text
0	2059027	male	15	Student	Leo	14,May,2004	Info has been found (+/- 100 pages,...
1	2059027	male	15	Student	Leo	13,May,2004	These are the team members: Drewe...
2	2059027	male	15	Student	Leo	12,May,2004	In het kader van kernfusie op aarde...
3	2059027	male	15	Student	Leo	12,May,2004	testing!!! testing!!!
4	3581210	male	33	InvestmentBanking	Aquarius	11,June,2004	Thanks to Yahoo!'s Toolbar I can ...

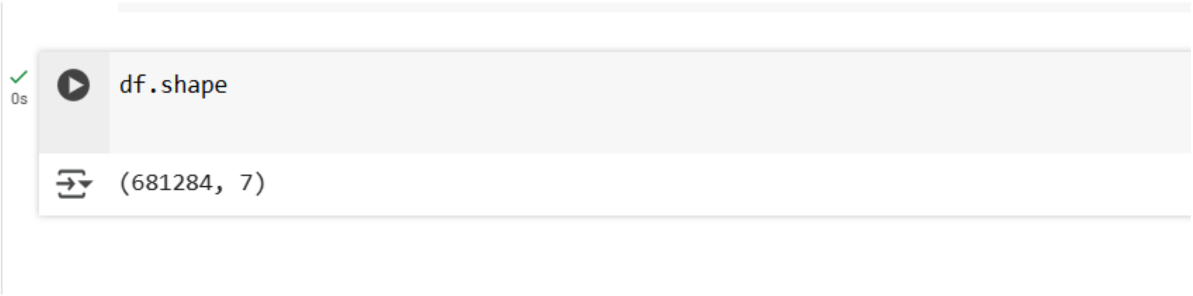


```
[3] import pandas as pd
```

```
path="/content/drive/MyDrive/python/blogtext.csv"
df=pd.read_csv(path)
df.head()
```

	id	gender	age	topic	sign	date	text
0	2059027	male	15	Student	Leo	14,May,2004	Info has been found (+/- 100 pages,...
1	2059027	male	15	Student	Leo	13,May,2004	These are the team members: Drewe...
2	2059027	male	15	Student	Leo	12,May,2004	In het kader van kernfusie op aarde...
3	2059027	male	15	Student	Leo	12,May,2004	testing!!! testing!!!
4	3581210	male	33	InvestmentBanking	Aquarius	11,June,2004	Thanks to Yahoo!'s Toolbar I can ...

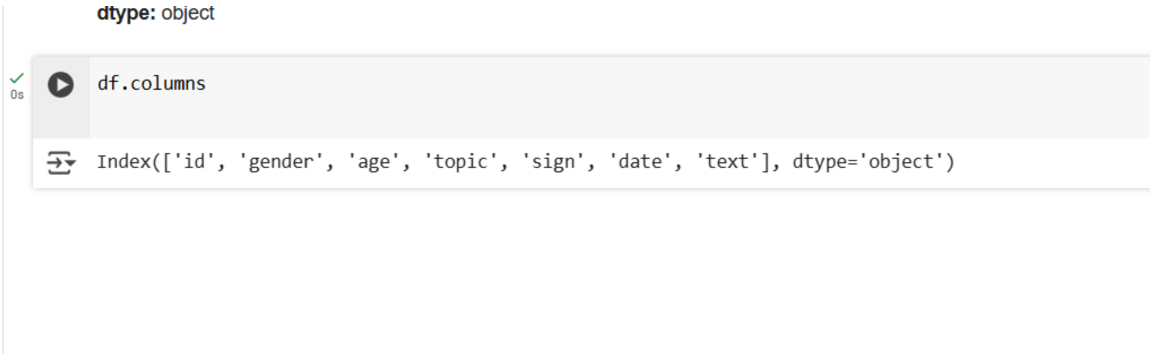
1. What is the shape of the dataset (rows, columns)?



```
df.shape
```

```
(681284, 7)
```

2. What are the columns available in the dataset?

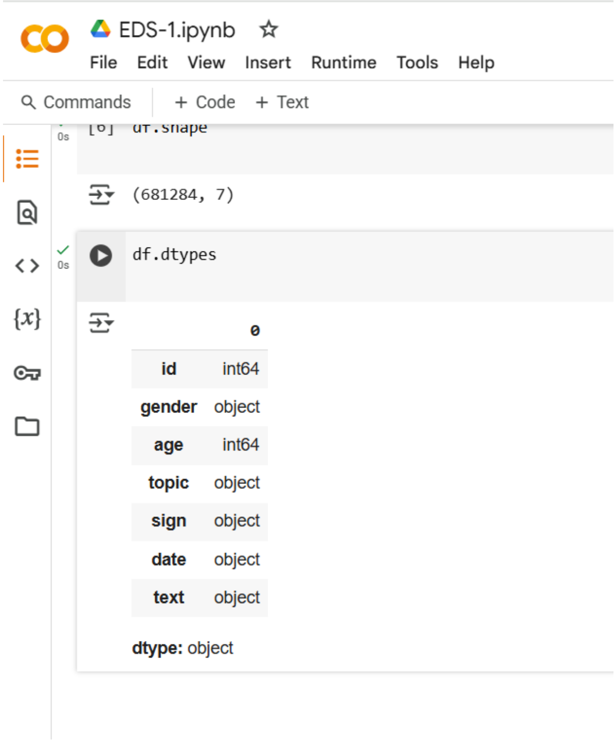


```
dtype: object
```

```
df.columns
```

```
Index(['id', 'gender', 'age', 'topic', 'sign', 'date', 'text'], dtype='object')
```

3. What is the data type of each column?



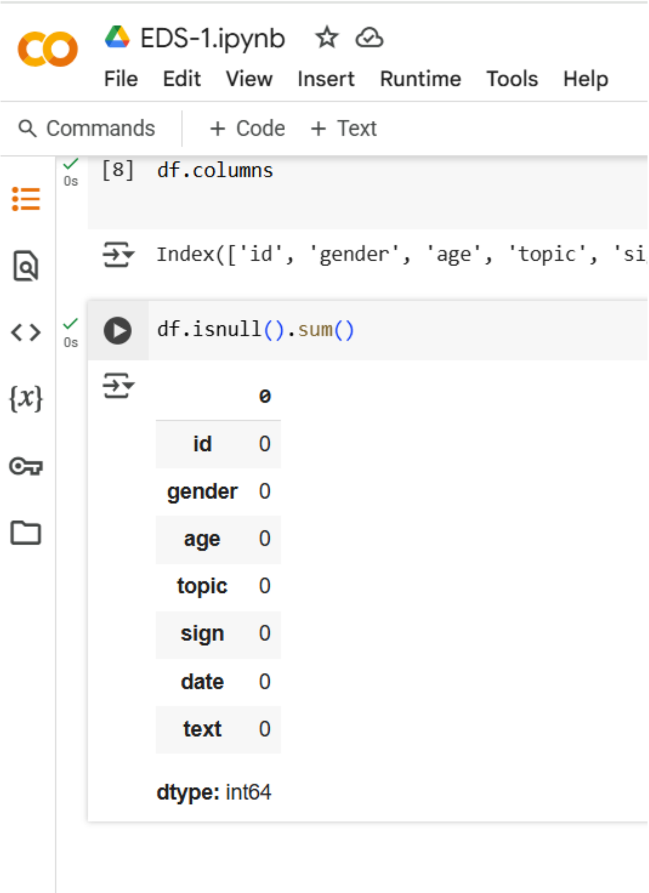
```
[6] df.shape
```

```
(681284, 7)
```

```
df.dtypes
```

```
             0
    id   int64
gender  object
   age   int64
 topic  object
  sign  object
  date  object
  text  object

dtype: object
```

4. How many missing values are in each column?



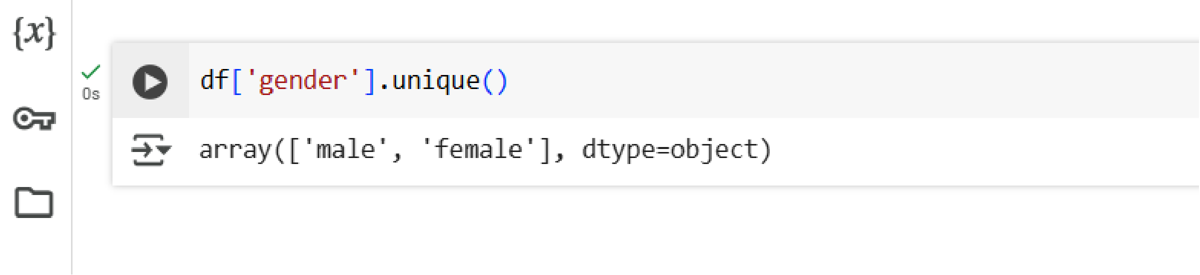
```
[8] df.columns
```

```
Index(['id', 'gender', 'age', 'topic', 'si
```

```
df.isnull().sum()
```

```
        0
id      0
gender  0
age     0
topic   0
sign    0
date    0
text    0

dtype: int64
```

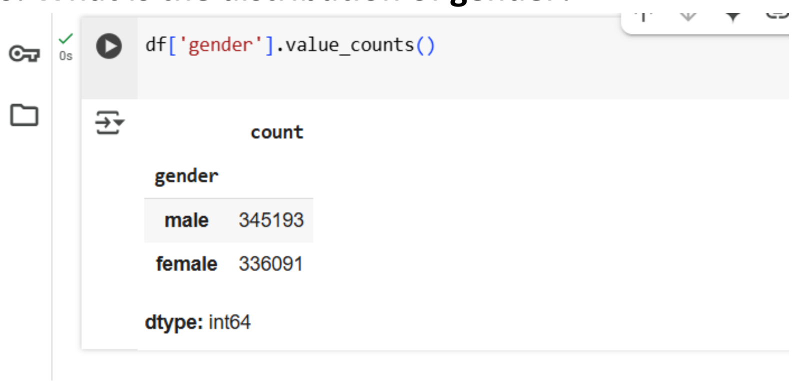
5. What are the unique genders available?



```
df['gender'].unique()
```

```
array(['male', 'female'], dtype=object)
```

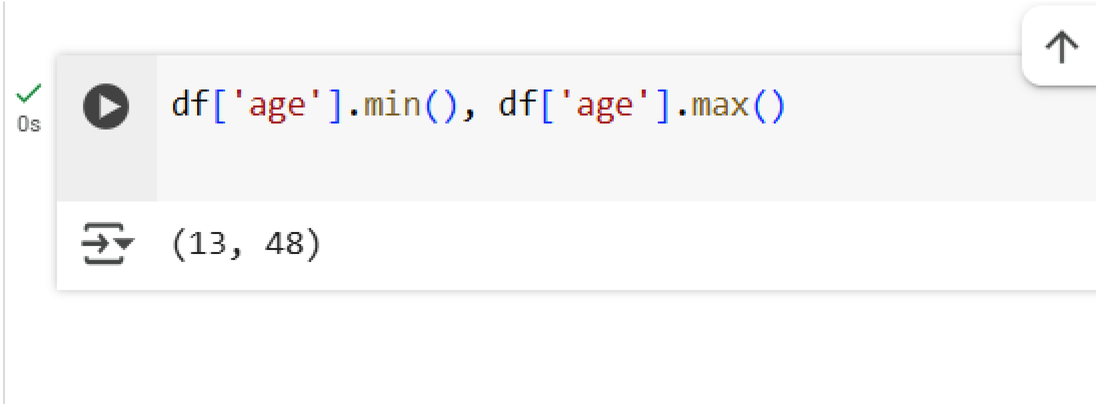
6. What is the distribution of gender?



```
df['gender'].value_counts()
```

```
         count
gender
male    345193
female  336091

dtype: int64
```

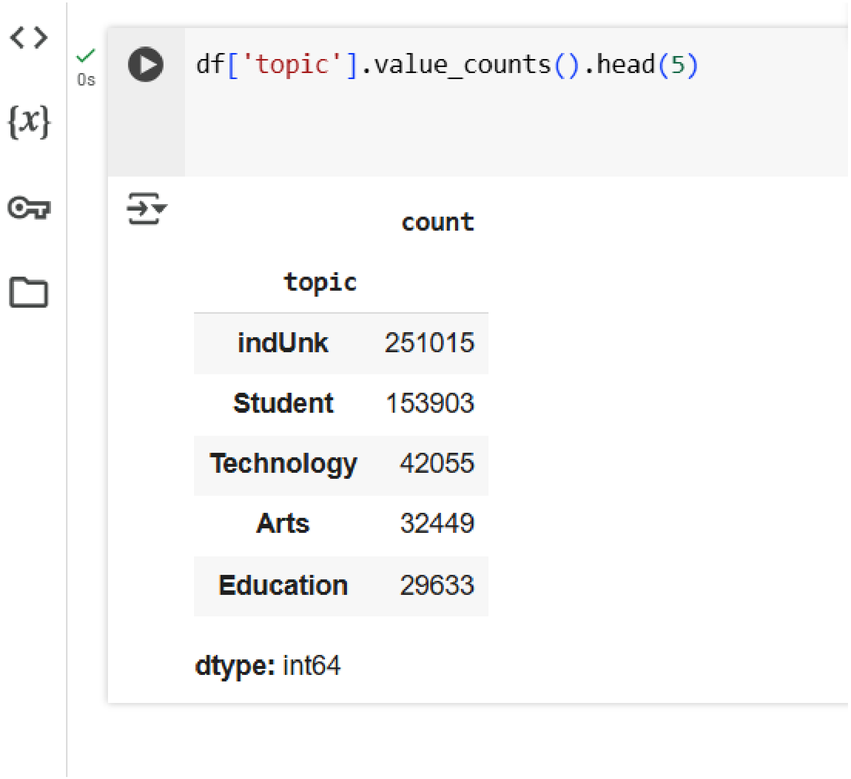
7. What is the minimum and maximum age of bloggers?



```
df['age'].min(), df['age'].max()
```

```
(13, 48)
```

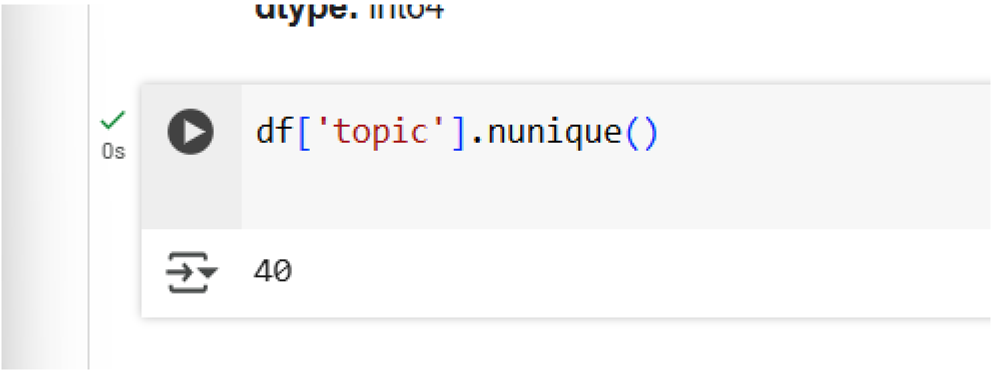
8. What are the most common blogging topics?



```
df['topic'].value_counts().head(5)
```

```
                count
      topic
     indUnk    251015
    Student    153903
 Technology     42055
       Arts     32449
  Education     29633

dtype: int64
```

9. How many distinct topics are there?

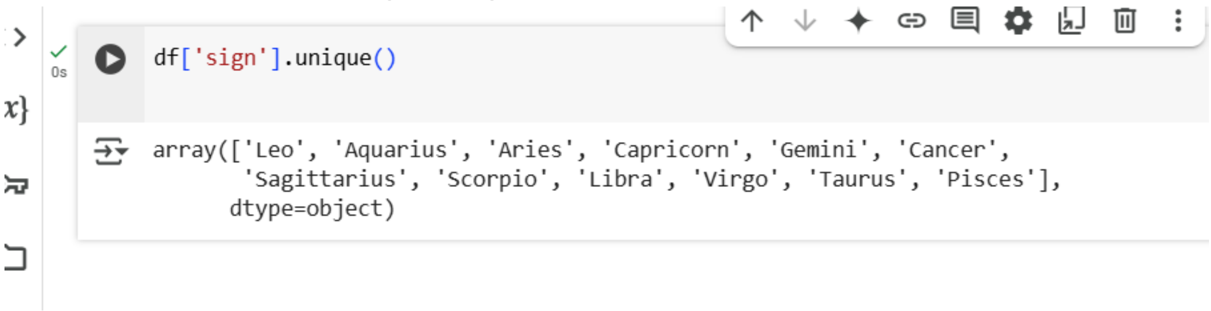


```
dtype: int64
```

```
df['topic'].nunique()
```

```
40
```

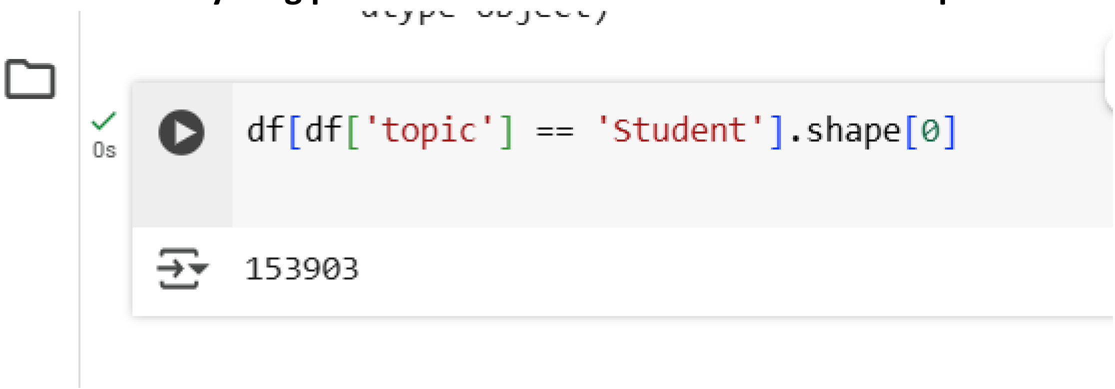
10. What are the astrological signs available?



```
df['sign'].unique()
```

```
array(['Leo', 'Aquarius', 'Aries', 'Capricorn', 'Gemini', 'Cancer',
       'Sagittarius', 'Scorpio', 'Libra', 'Virgo', 'Taurus', 'Pisces'],
      dtype=object)
```

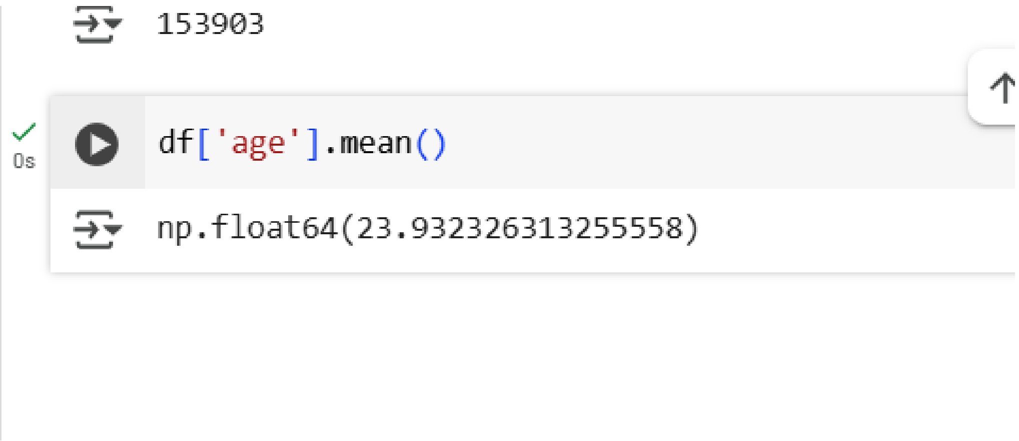
11. How many blog posts are written under the "Student" topic?



```
df[df['topic'] == 'Student'].shape[0]
```

```
153903
```

12. What is the average age of bloggers?

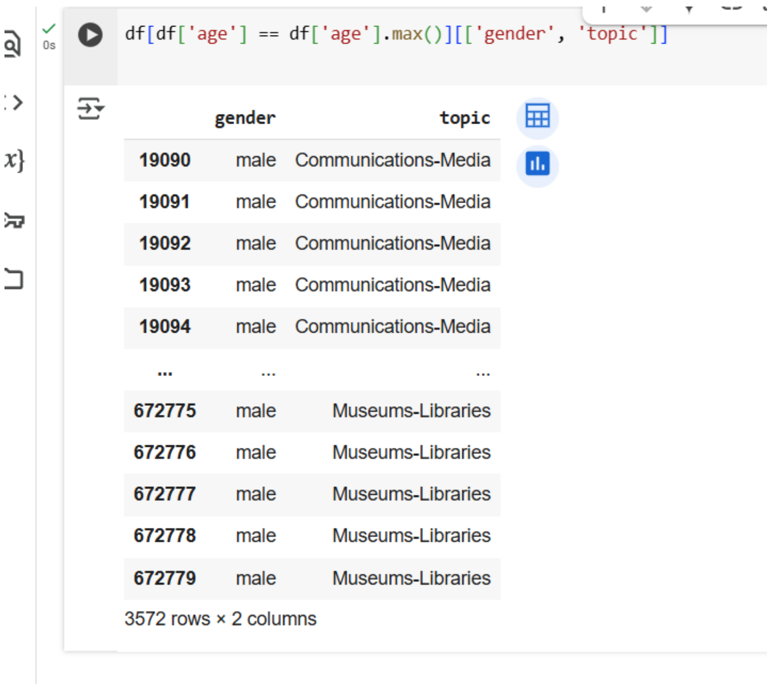


```
153903
```

```
df['age'].mean()
```

```
np.float64(23.932326313255558)
```

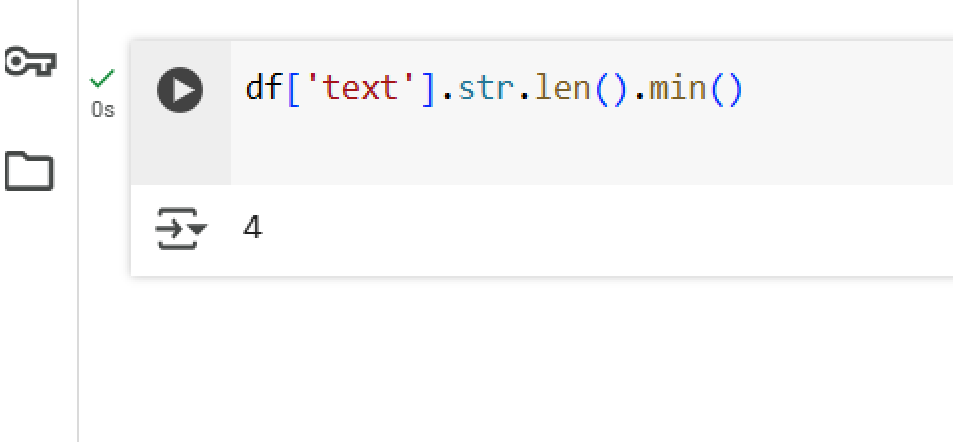
13. What is the oldest blogger's gender and topic?



```
df[df['age'] == df['age'].max()][['gender', 'topic']]
```

```
        gender                topic
19090     male  Communications-Media
19091     male  Communications-Media
19092     male  Communications-Media
19093     male  Communications-Media
19094     male  Communications-Media
  ...      ...                   ...
672775    male     Museums-Libraries
672776    male     Museums-Libraries
672777    male     Museums-Libraries
672778    male     Museums-Libraries
672779    male     Museums-Libraries
3572 rows × 2 columns
```

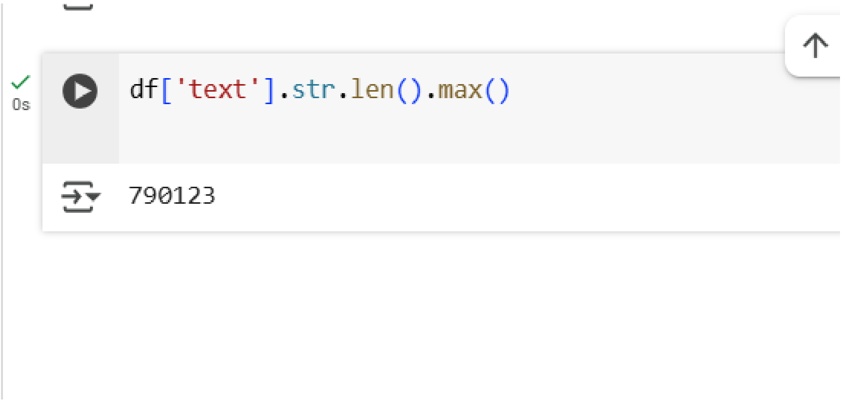
14. What is the shortest blog post?



```
df['text'].str.len().min()
```

```
4
```

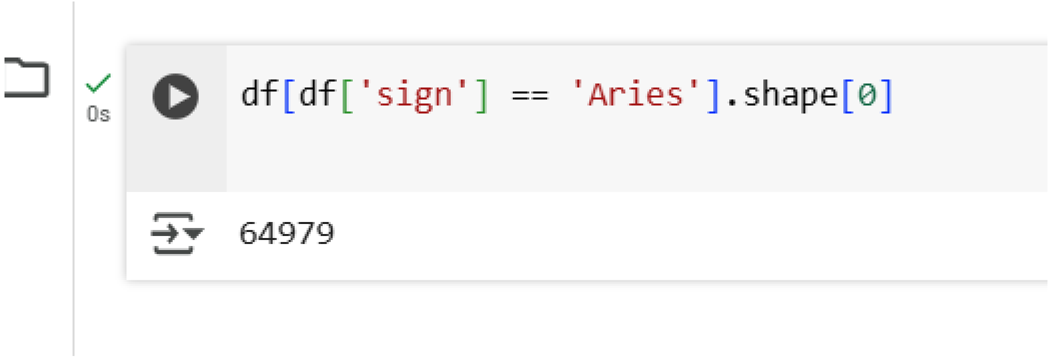
15. What is the longest blog post?



```
df['text'].str.len().max()
```

```
790123
```

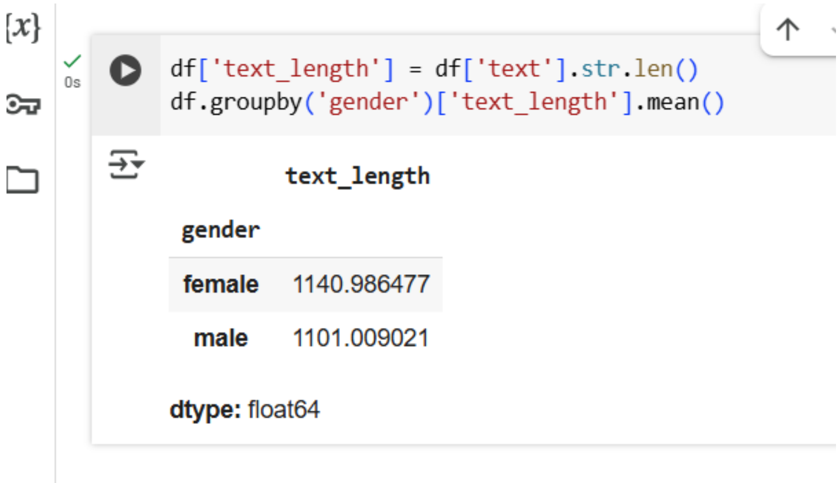
16. How many posts were made by Aries bloggers?



```
df[df['sign'] == 'Aries'].shape[0]
```

```
64979
```

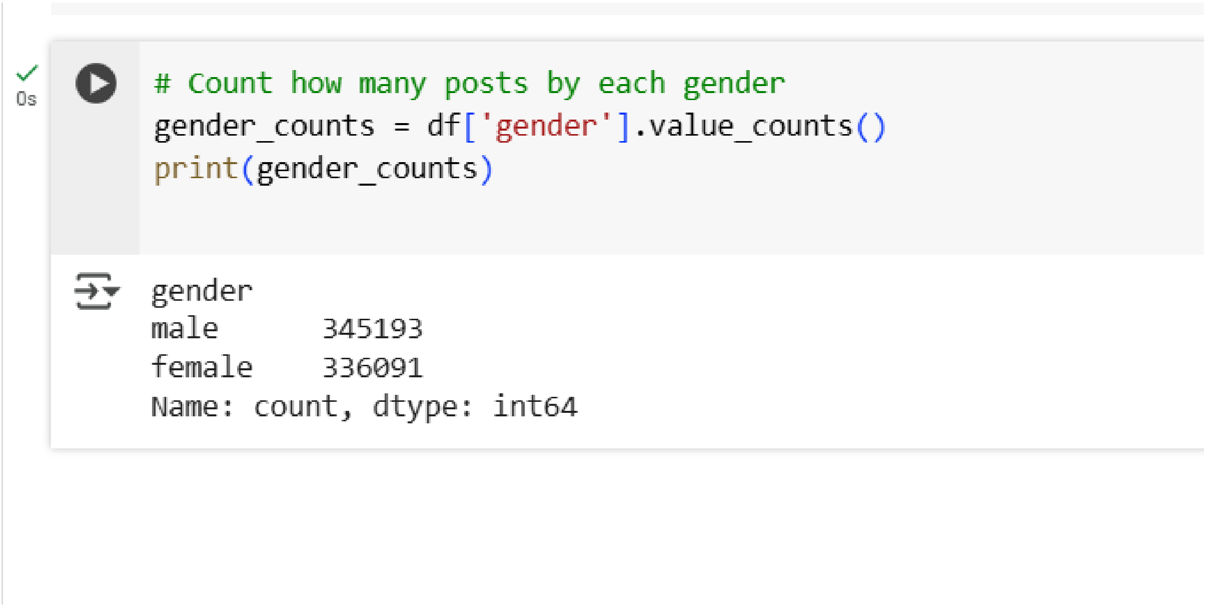
17. Who writes longer posts on average: male or female?



```
df['text_length'] = df['text'].str.len()
df.groupby('gender')['text_length'].mean()
```

```
         text_length
gender
female   1140.986477
  male   1101.009021

dtype: float64
```

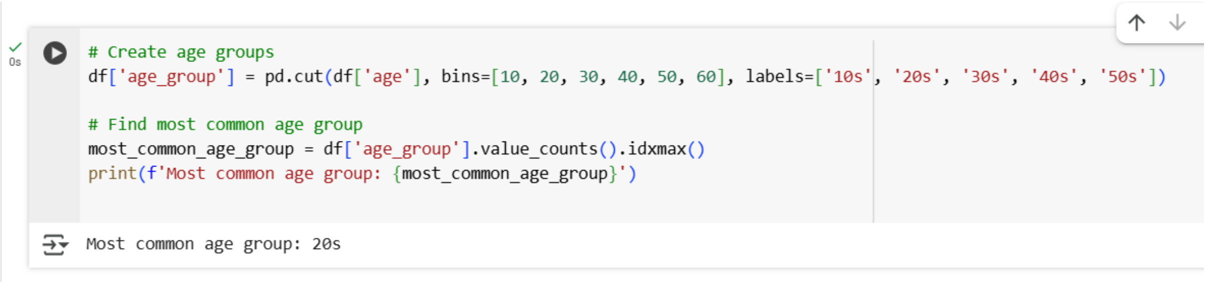
18. How many blog posts were written by male vs female authors?



```
# Count how many posts by each gender
gender_counts = df['gender'].value_counts()
print(gender_counts)
```

```
gender
male      345193
female    336091
Name: count, dtype: int64
```

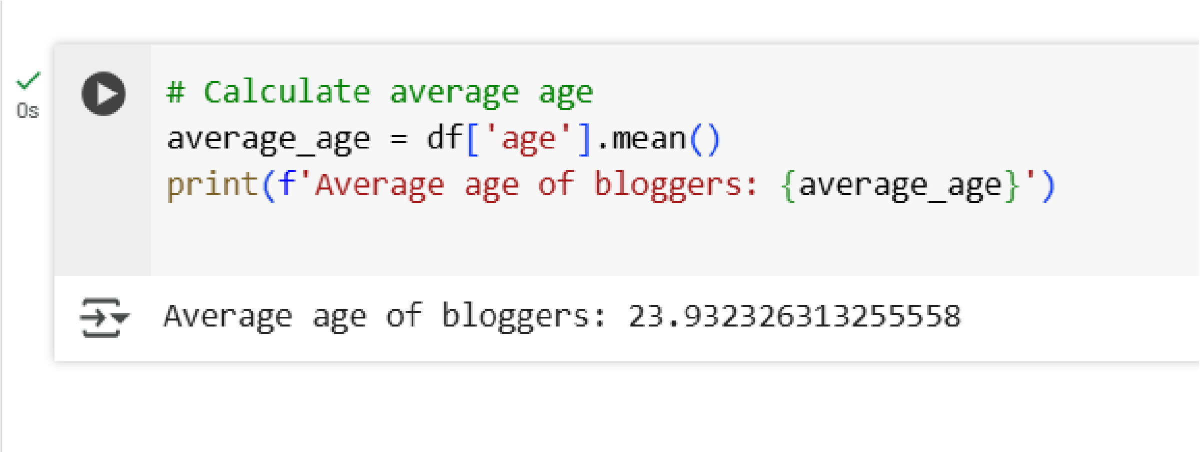
19. Which age group has written the most blog posts?



```python
# Create age groups
df['age_group'] = pd.cut(df['age'], bins=[10, 20, 30, 40, 50, 60], labels=['10s', '20s', '30s', '40s', '50s'])

# Find most common age group
most_common_age_group = df['age_group'].value_counts().idxmax()
print(f'Most common age group: {most_common_age_group}')
```

```
Most common age group: 20s
```

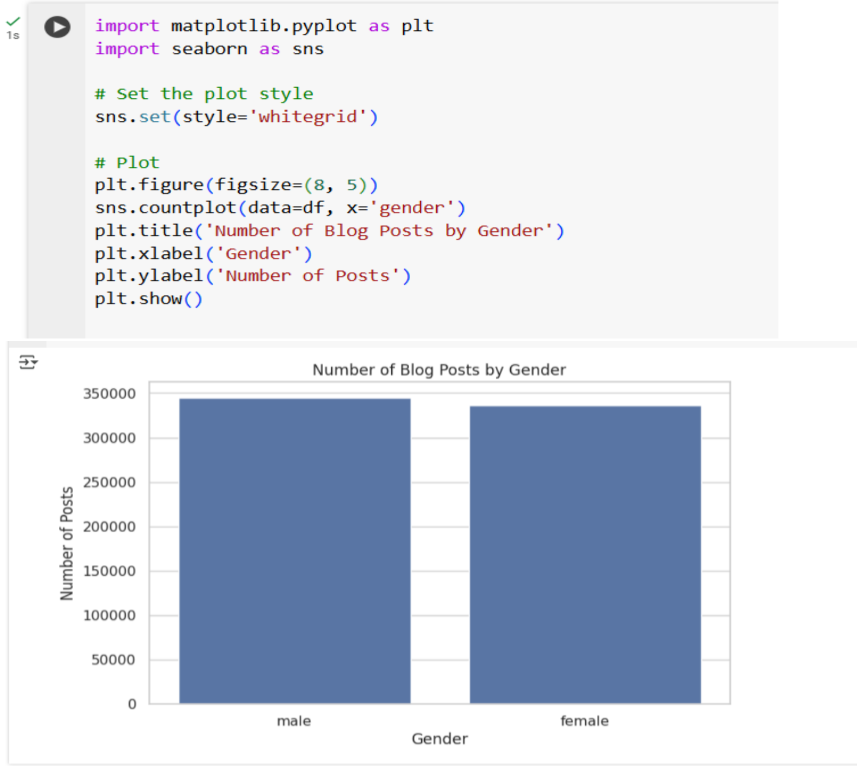
20. Find the average age of all bloggers.



```python
# Calculate average age
average_age = df['age'].mean()
print(f'Average age of bloggers: {average_age}')
```

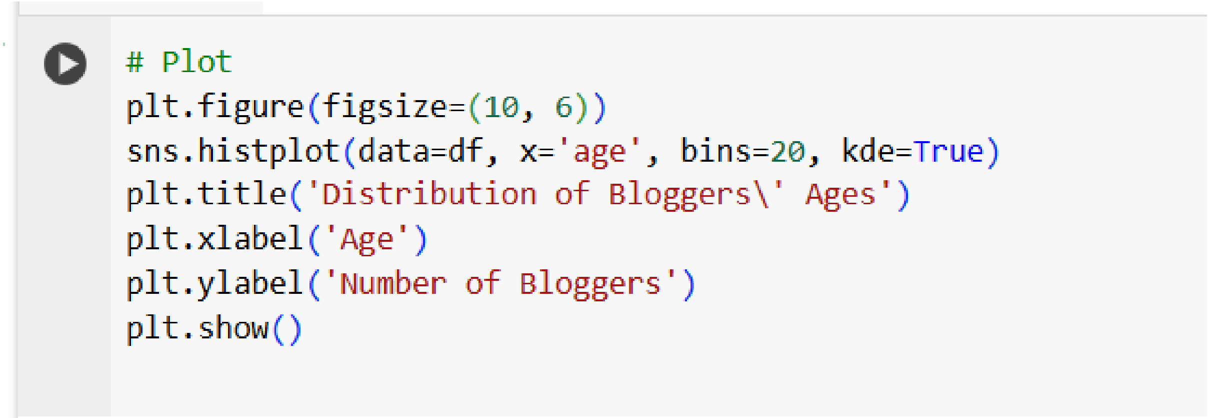
```
Average age of bloggers: 23.932326313255558
```

21. Plot the number of blog posts by gender.

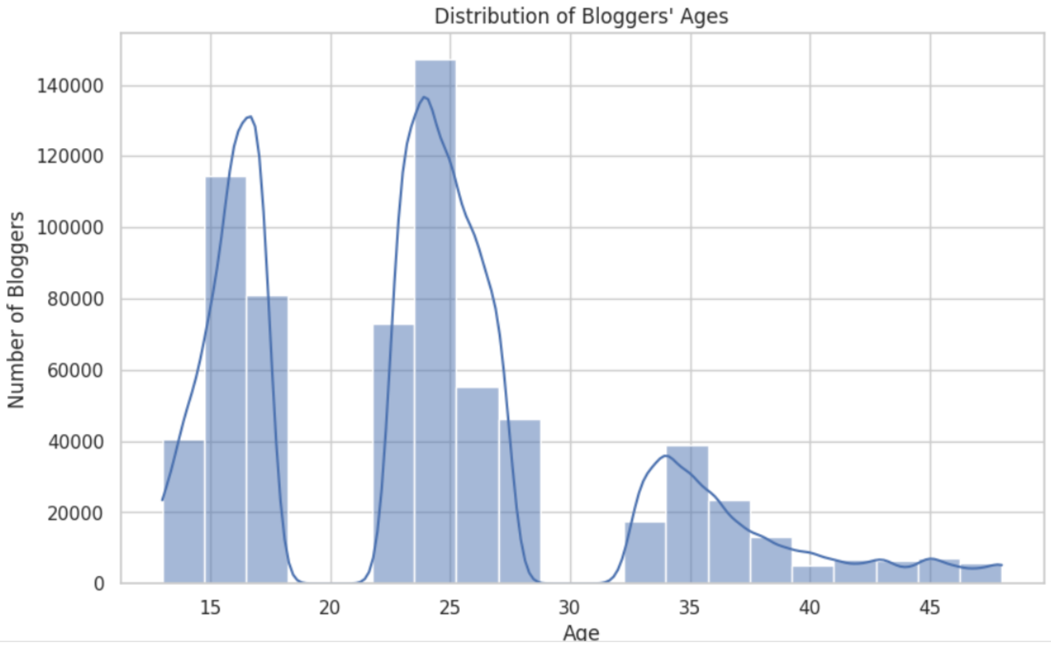


```python
import matplotlib.pyplot as plt
import seaborn as sns

# Set the plot style
sns.set(style='whitegrid')

# Plot
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='gender')
plt.title('Number of Blog Posts by Gender')
plt.xlabel('Gender')
plt.ylabel('Number of Posts')
plt.show()
```

22. Plot the distribution of bloggers' ages.



```
# Plot
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='age', bins=20, kde=True)
plt.title('Distribution of Bloggers\' Ages')
plt.xlabel('Age')
plt.ylabel('Number of Bloggers')
plt.show()
```


Distribution of Bloggers' Ages
Number of Bloggers
Age