Exploratory Data Analysis

Introduction to the data

The data consisted of 1000 rows of data spread on 7 columns:

1. id: id is a unique value for row and is of type int64
2. tit: an unique id for each book and is of type object/string
3. title: is the title of the book that is entered and is of type object/string
4. date: I assume is the data the book has been published and is of type object/string
5. author: provides the name of the author of the book and has duplicated names and is of type object/string
6. story: is a value of type object/string that provides more information about the book
7. topic: a value of type object/string that is the same throughout one data set

1.Data Preprocessing

Data from the 11 stories data sets provided was first loaded into different data frames to check for each data frame's values and types. Also checking for nan values or missing values within all the data frames was necessary. Consequently, all data frames were merged into a single data frame. Each data frame consisted of 1000 rows x 7 columns of data all with matching columns and types, and after finding no missing values, merging was completed successfully. The merged data frame consisted of 11000 rows x 7 columns of data with no missing values.

▾ Merging the data frames

```
merged_df = pd.concat([df, df2, df3, df4, df5, df6, df7, df8, df9, df10, df11])
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11000 entries, 0 to 999
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  11000 non-null  int64
 1   id          11000 non-null  object
 2   title       11000 non-null  object
 3   date        11000 non-null  object
 4   author      11000 non-null  object
 5   story       11000 non-null  object
 6   topic       11000 non-null  object
dtypes: int64(1), object(6)
memory usage: 687.5+ KB
```

```
[99] merged_df.dtypes
```

```
Unnamed: 0     int64
id            object
title         object
date          object
author        object
story         object
topic         object
dtype: object
```

2.Data processing

Each column in the merged data frame was then checked for unique values. The topics column only consisted of 10 unique topics, so I used label encoding to change the categorical value into a numerical value for further analysis. The author columns also consisted of 773 unique values which was also encoded using label encoding. Other columns shown a huge number of unique values which made it infeasible to encode.

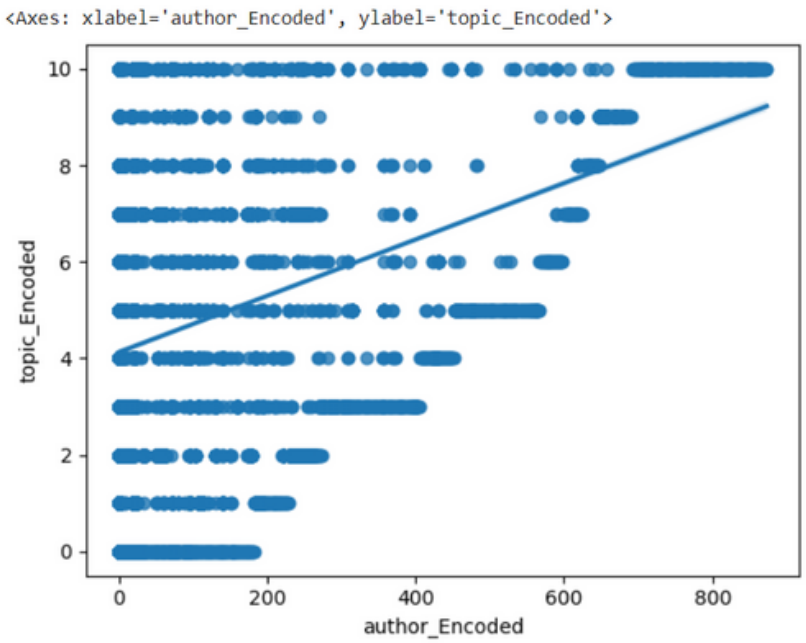
I then checked for correlation between numeric categories made through the encoding of the two columns mentioned above and found a strong correlation between topics and authors with a Pearson Correlation Coefficient of 0.45670151243298396 with a P-value of P = 0.0. Such numbers proved the correlation between authors and topics chosen as shown below .

```
[86] from scipy import stats
     pearson_coef, p_value = stats.pearsonr(merged_df['author_Encoded'], merged_df['topic_Encoded'])
     print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value of P =", p_value)

     The Pearson Correlation Coefficient is 0.45670151243298396  with a P-value of P = 0.0
```

since p value is less than <0.001 then there is a strong correlation between topic and author

the graph below is a scatter plot of the correlation between topic_Encoded and author_Encoded



The plot below shows the count of occurrences of each topic after being encoded with "tamazight" showing the highest number of occurrences.