EDS THEORY ASSIGNMENT 1

Name: Komal Dhanajay Gangawane

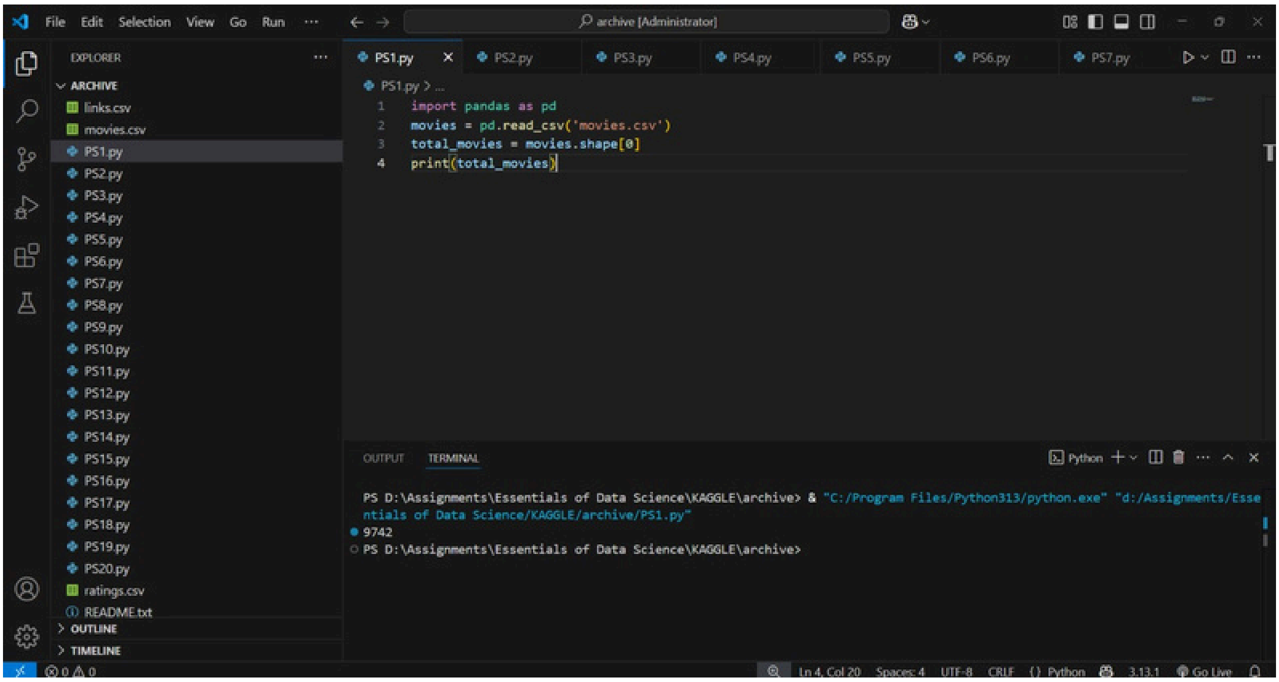
DIV: ET2

Roll No: ET2-75

PRN: 202401070218

URL: https://www.kaggle.com/datasets/shubhammehta21/movie-lens-small-latest-dataset

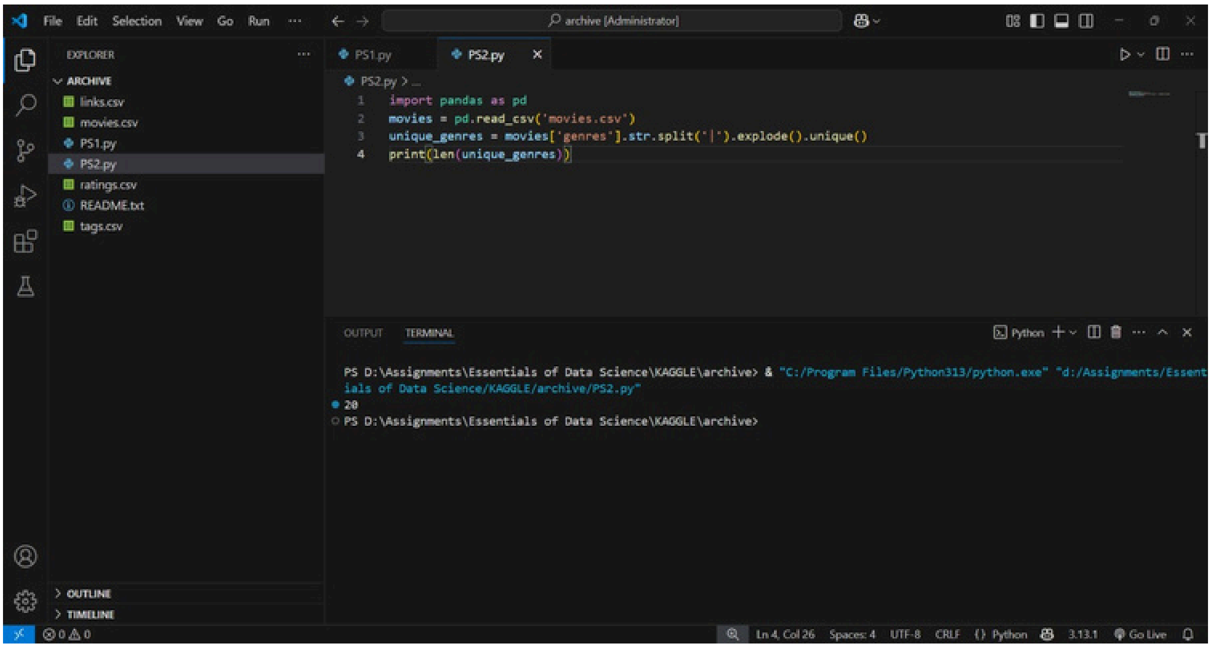
1. Count the total number of movies in the dataset.



```python
import pandas as pd
movies = pd.read_csv('movies.csv')
total_movies = movies.shape[0]
print(total_movies)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS1.py"
9742
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

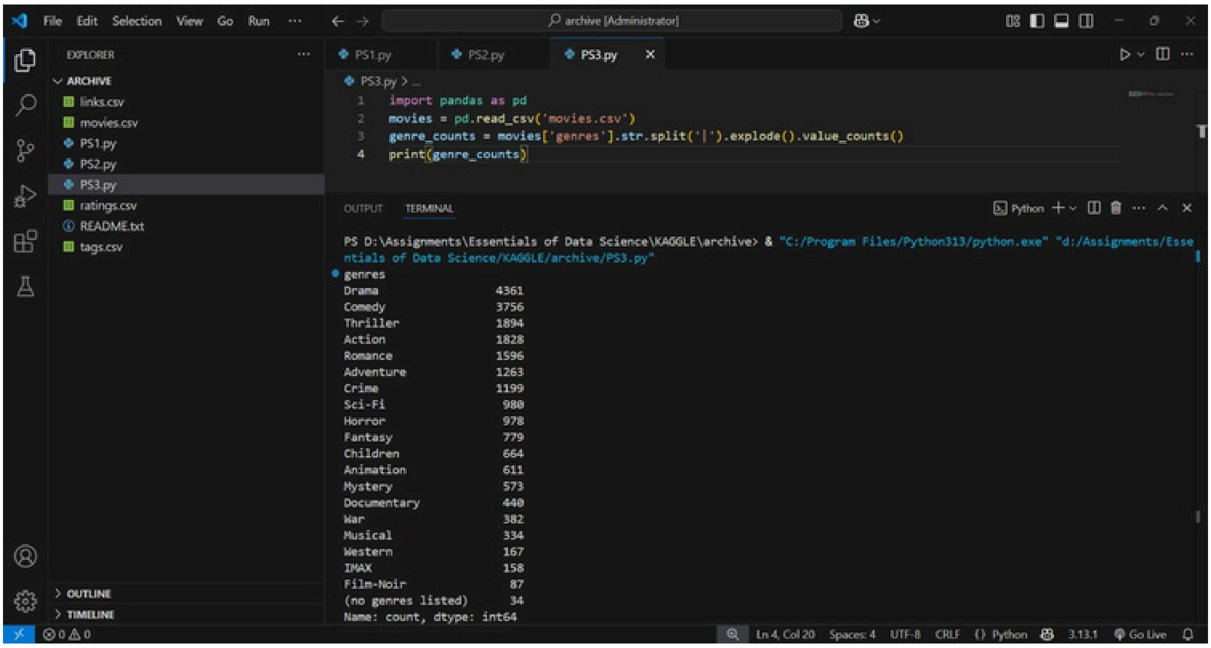
2. Determine the number of unique genres spanning all movies.



```python
import pandas as pd
movies = pd.read_csv('movies.csv')
unique_genres = movies['genres'].str.split('|').explode().unique()
print(len(unique_genres))
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS2.py"
20
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

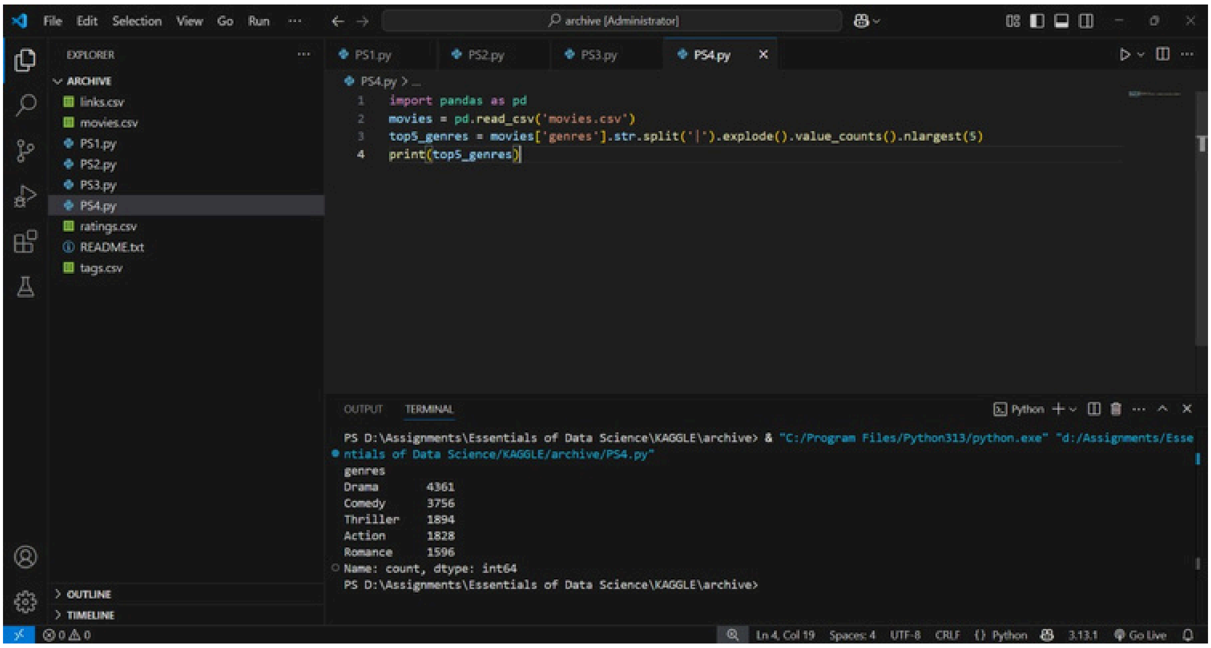
3. Compute how many times each genre appears by splitting and exploding the genre strings.



```python
import pandas as pd
movies = pd.read_csv('movies.csv')
genre_counts = movies['genres'].str.split('|').explode().value_counts()
print(genre_counts)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS3.py"
genres
Drama                 4361
Comedy                3756
Thriller              1894
Action                1828
Romance               1596
Adventure             1263
Crime                 1199
Sci-Fi                 980
Horror                 978
Fantasy                779
Children               664
Animation              611
Mystery                573
Documentary            440
War                    382
Musical                334
Western                167
IMAX                   158
Film-Noir               87
(no genres listed)      34
Name: count, dtype: int64
```

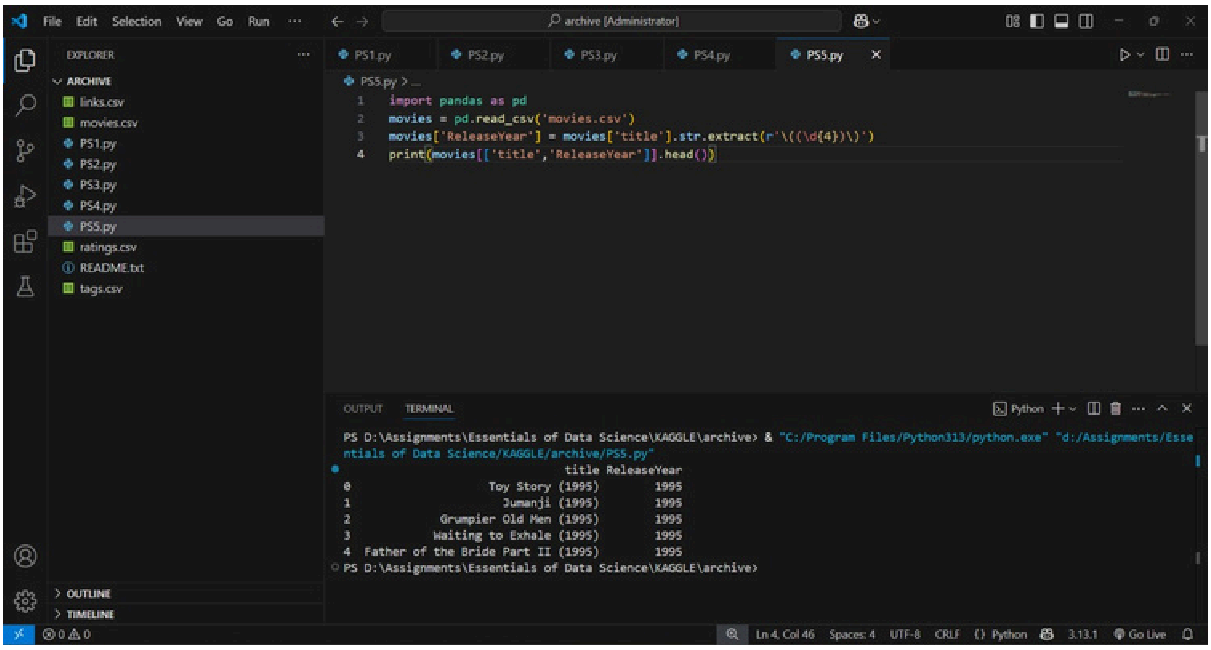
4. Identify the five most frequent genres in the dataset.



```python
import pandas as pd
movies = pd.read_csv('movies.csv')
top5_genres = movies['genres'].str.split('|').explode().value_counts().nlargest(5)
print(top5_genres)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Esse
ntials of Data Science/KAGGLE/archive/PS4.py"
genres
Drama       4361
Comedy      3756
Thriller    1894
Action      1828
Romance     1596
Name: count, dtype: int64
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

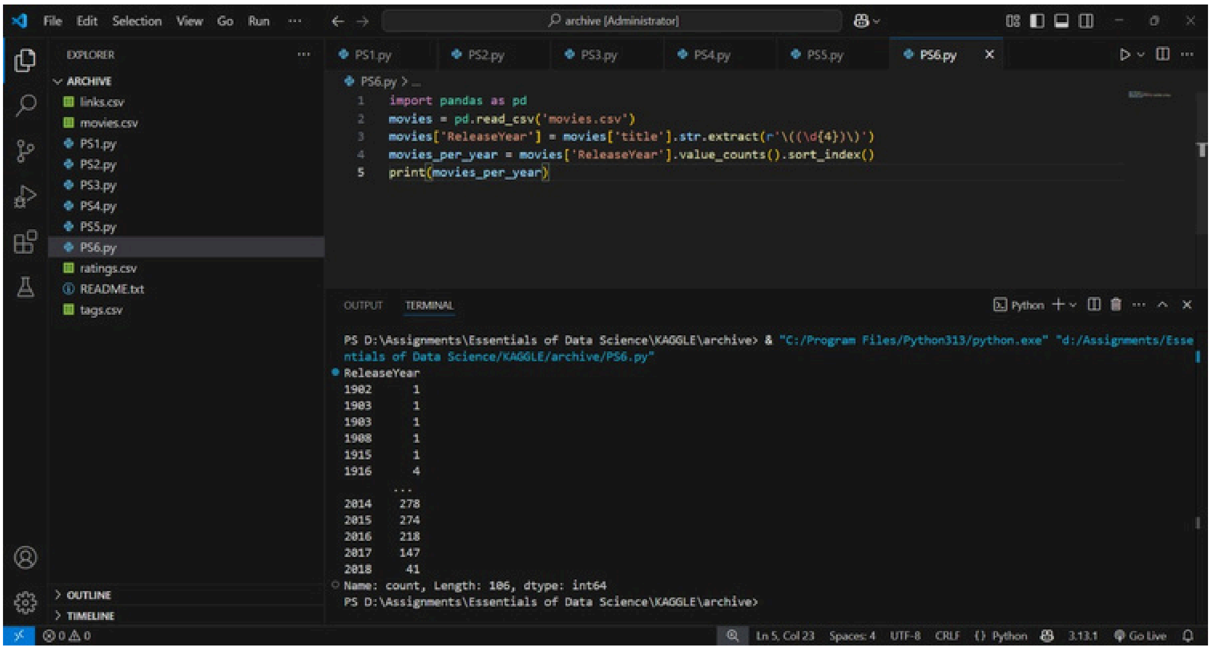
5. Extract the release year from the movie titles.



```python
import pandas as pd
movies = pd.read_csv('movies.csv')
movies['ReleaseYear'] = movies['title'].str.extract(r'\((\d{4})\)')
print(movies[['title','ReleaseYear']].head())
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Esse
ntials of Data Science/KAGGLE/archive/PS5.py"
                        title ReleaseYear
0                Toy Story (1995)        1995
1                  Jumanji (1995)        1995
2         Grumpier Old Men (1995)        1995
3        Waiting to Exhale (1995)        1995
4  Father of the Bride Part II (1995)        1995
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

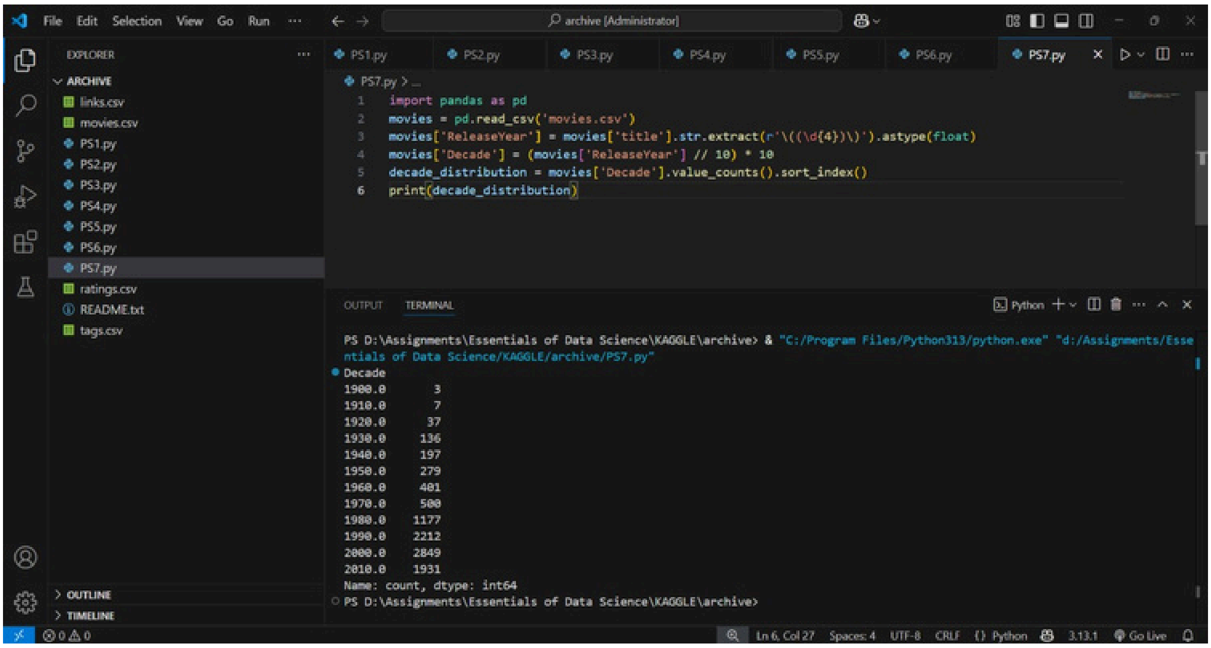
6. Count how many movies were released in each year.



```python
import pandas as pd
movies = pd.read_csv('movies.csv')
movies['ReleaseYear'] = movies['title'].str.extract(r'\((\d{4})\)')
movies_per_year = movies['ReleaseYear'].value_counts().sort_index()
print(movies_per_year)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Esse
ntials of Data Science/KAGGLE/archive/PS6.py"
ReleaseYear
1902      1
1903      1
1903      1
1908      1
1915      1
1916      4
        ...
2014    278
2015    274
2016    218
2017    147
2018     41
Name: count, Length: 106, dtype: int64
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

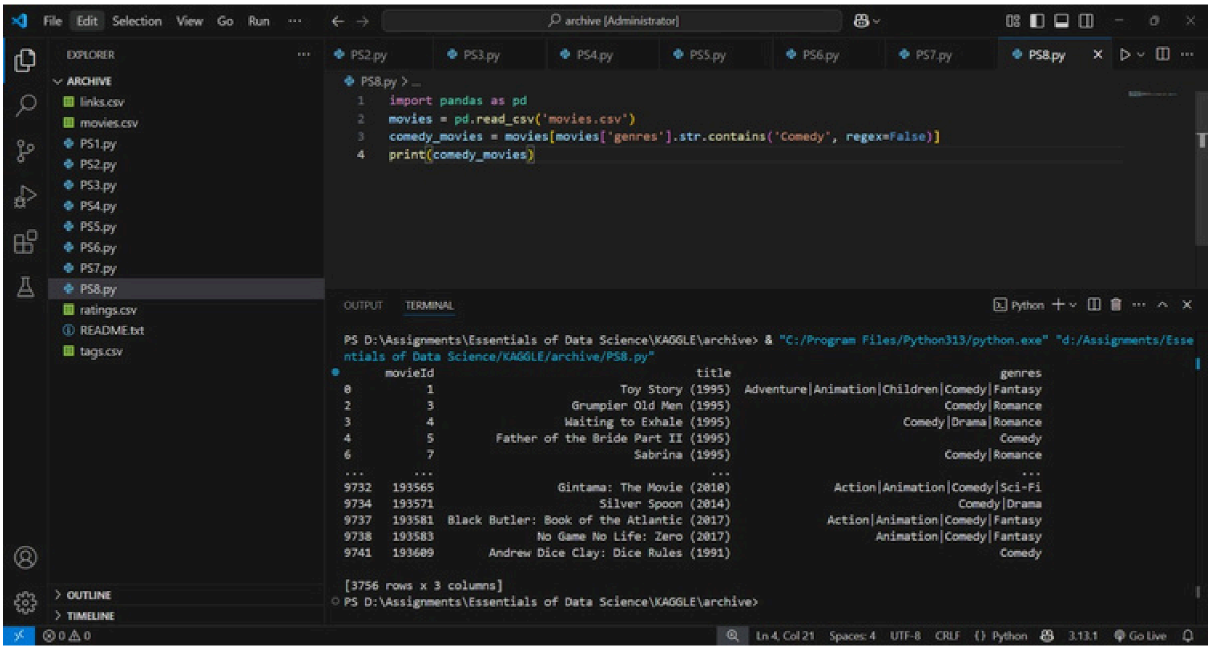
7. Derive each movie's decade (based on the release year) and show the distribution.



```python
import pandas as pd
movies = pd.read_csv('movies.csv')
movies['ReleaseYear'] = movies['title'].str.extract(r'\((\d{4})\)').astype(float)
movies['Decade'] = (movies['ReleaseYear'] // 10) * 10
decade_distribution = movies['Decade'].value_counts().sort_index()
print(decade_distribution)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Esse
ntials of Data Science/KAGGLE/archive/PS7.py"
Decade
1900.0       3
1910.0       7
1920.0      37
1930.0     136
1940.0     197
1950.0     279
1960.0     401
1970.0     500
1980.0    1177
1990.0    2212
2000.0    2849
2010.0    1931
Name: count, dtype: int64
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

8. Filter out the movies that belong to the "Comedy" genre.

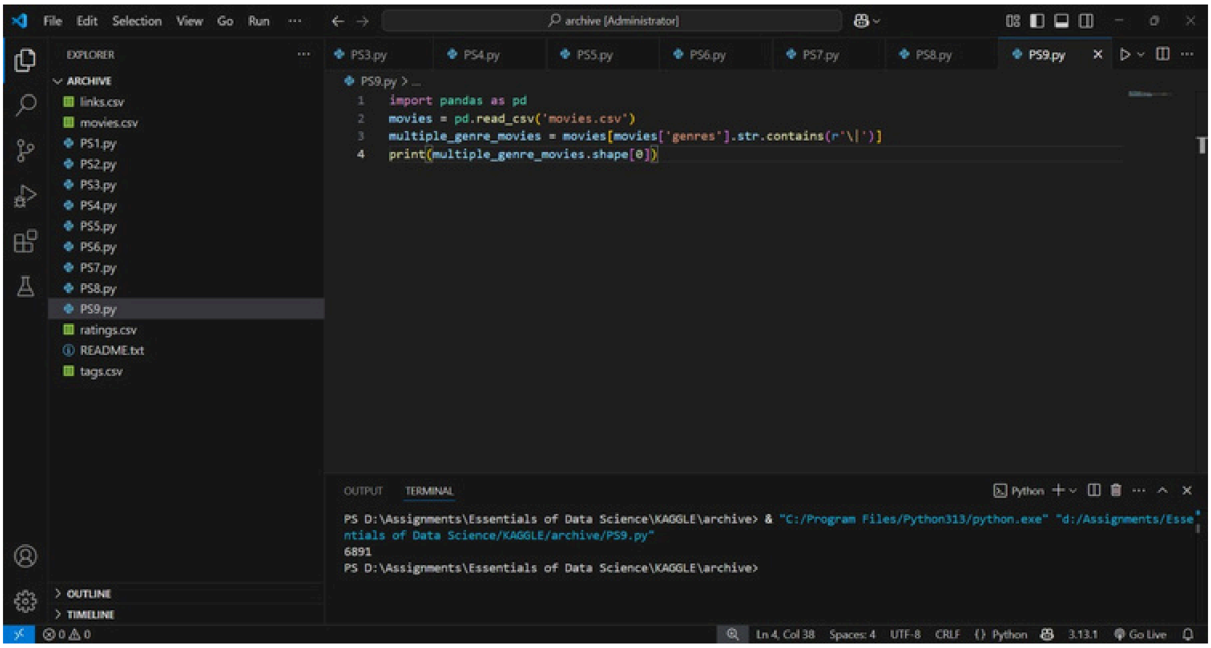


```python
import pandas as pd
movies = pd.read_csv('movies.csv')
comedy_movies = movies[movies['genres'].str.contains('Comedy', regex=False)]
print(comedy_movies)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS8.py"
      movieId                                      title                                        genres
0           1                           Toy Story (1995)  Adventure|Animation|Children|Comedy|Fantasy
2           3                    Grumpier Old Men (1995)                               Comedy|Romance
3           4                   Waiting to Exhale (1995)                         Comedy|Drama|Romance
4           5         Father of the Bride Part II (1995)                                       Comedy
6           7                             Sabrina (1995)                               Comedy|Romance
...       ...                                        ...                                          ...
9732   193565                   Gintama: The Movie (2010)                  Action|Animation|Comedy|Sci-Fi
9734   193571                        Silver Spoon (2014)                                 Comedy|Drama
9737   193581  Black Butler: Book of the Atlantic (2017)           Action|Animation|Comedy|Fantasy
9738   193583                 No Game No Life: Zero (2017)                 Animation|Comedy|Fantasy
9741   193609          Andrew Dice Clay: Dice Rules (1991)                                       Comedy

[3756 rows x 3 columns]
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

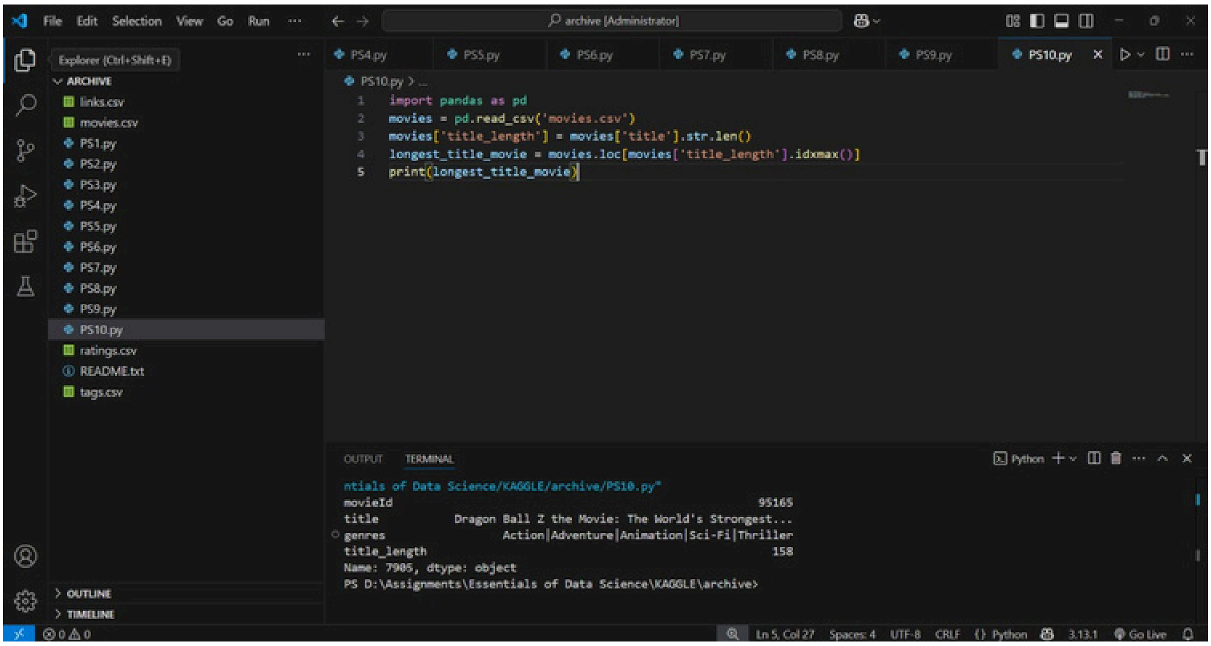
9. Count the number of movies that have more than one genre listed.



```python
import pandas as pd
movies = pd.read_csv('movies.csv')
multiple_genre_movies = movies[movies['genres'].str.contains(r'\|')]
print(multiple_genre_movies.shape[0])
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS9.py"
6891
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

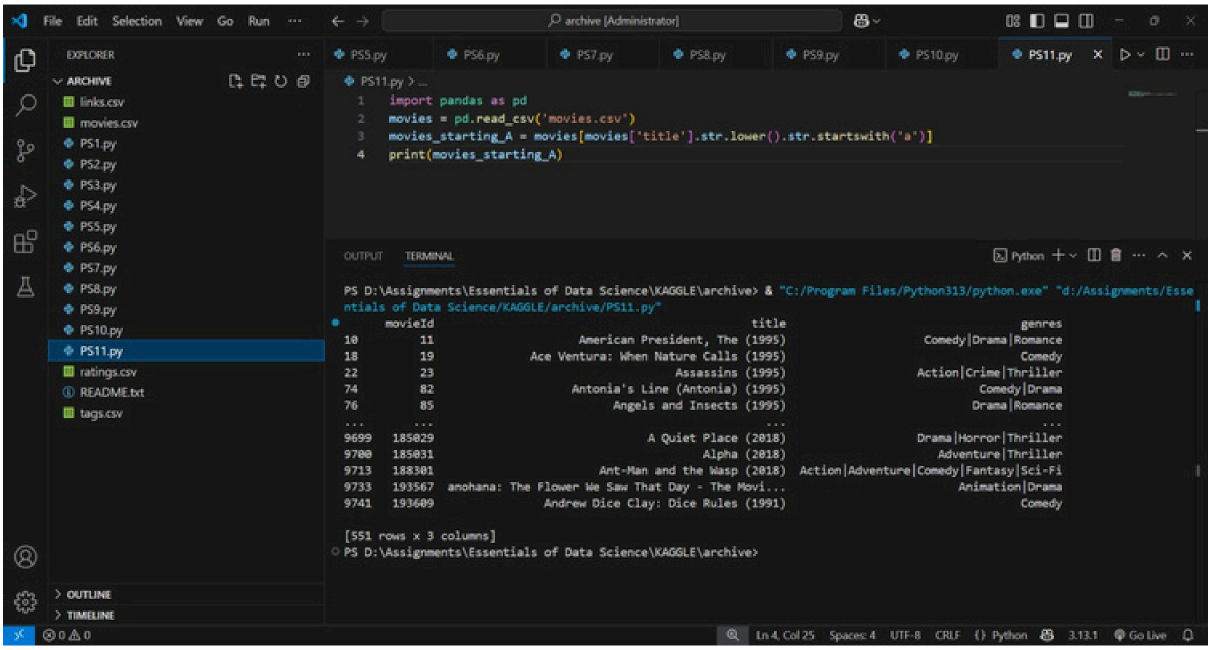
10. Identify the movie with the longest title.



```
import pandas as pd
movies = pd.read_csv('movies.csv')
movies['title_length'] = movies['title'].str.len()
longest_title_movie = movies.loc[movies['title_length'].idxmax()]
print(longest_title_movie)
```

```
ntials of Data Science/KAGGLE/archive/PS10.py"
movieId                                                     95165
title           Dragon Ball Z the Movie: The World's Strongest...
genres                Action|Adventure|Animation|Sci-Fi|Thriller
title_length                                                  158
Name: 7905, dtype: object
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

11. Filter movies that have titles starting with the letter “A”.

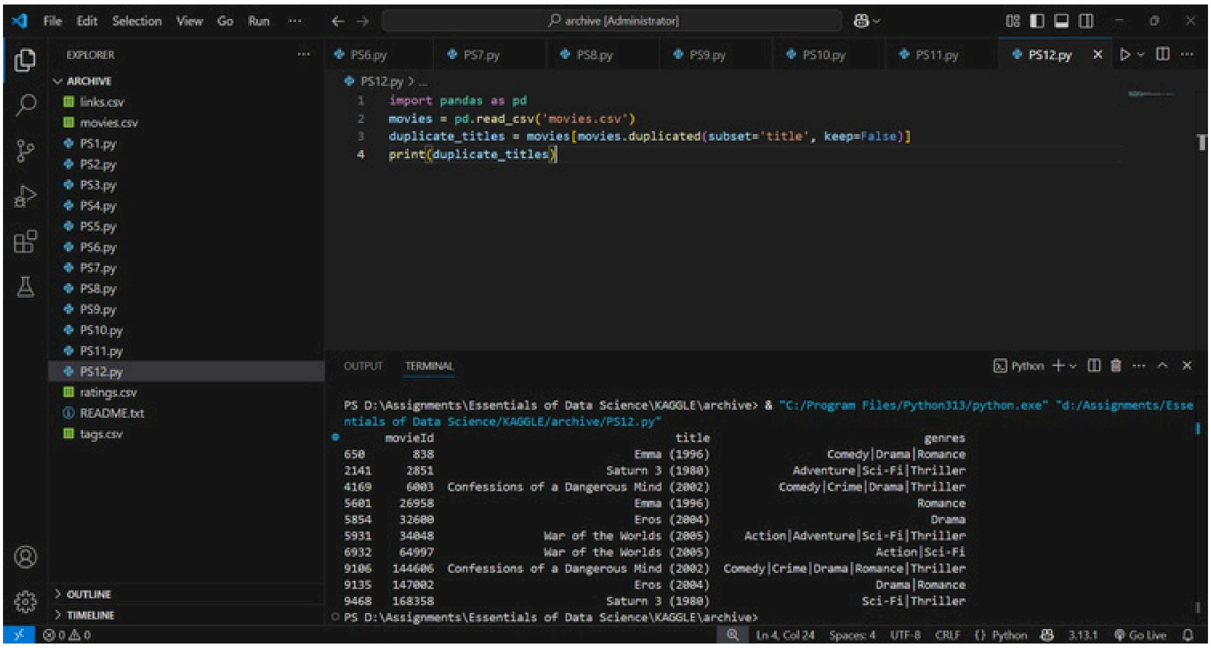


```
import pandas as pd
movies = pd.read_csv('movies.csv')
movies_starting_A = movies[movies['title'].str.lower().str.startswith('a')]
print(movies_starting_A)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Esse
ntials of Data Science/KAGGLE/archive/PS11.py"
      movieId                                           title                                genres
10         11                 American President, The (1995)                  Comedy|Drama|Romance
18         19             Ace Ventura: When Nature Calls (1995)                            Comedy
22         23                              Assassins (1995)                 Action|Crime|Thriller
74         82                  Antonia's Line (Antonia) (1995)                         Comedy|Drama
76         85                       Angels and Insects (1995)                        Drama|Romance
...       ...                                             ...                                   ...
9699   185029                          A Quiet Place (2018)                 Drama|Horror|Thriller
9700   185031                                  Alpha (2018)                    Adventure|Thriller
9713   188301                   Ant-Man and the Wasp (2018)  Action|Adventure|Comedy|Fantasy|Sci-Fi
9733   193567  anohana: The Flower We Saw That Day - The Movi...                    Animation|Drama
9741   193609              Andrew Dice Clay: Dice Rules (1991)                            Comedy

[551 rows x 3 columns]
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

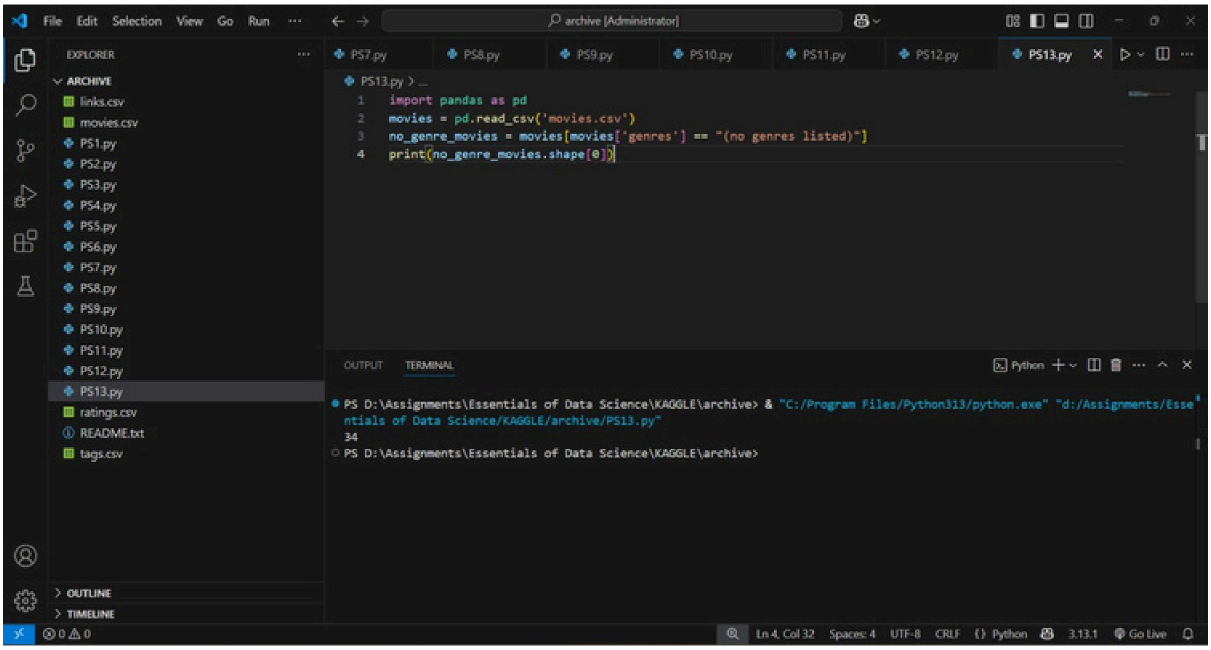
12. Identify movie titles that appear more than once.



```python
import pandas as pd
movies = pd.read_csv('movies.csv')
duplicate_titles = movies[movies.duplicated(subset='title', keep=False)]
print(duplicate_titles)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS12.py"
      movieId                                 title                                   genres
650       838                           Emma (1996)                     Comedy|Drama|Romance
2141     2851                       Saturn 3 (1980)                Adventure|Sci-Fi|Thriller
4169     6003  Confessions of a Dangerous Mind (2002)            Comedy|Crime|Drama|Thriller
5601    26958                           Emma (1996)                                  Romance
5854    32600                           Eros (2004)                                    Drama
5931    34048                 War of the Worlds (2005)      Action|Adventure|Sci-Fi|Thriller
6932    64997                 War of the Worlds (2005)                          Action|Sci-Fi
9106   144606  Confessions of a Dangerous Mind (2002)  Comedy|Crime|Drama|Romance|Thriller
9135   147002                           Eros (2004)                            Drama|Romance
9468   168358                       Saturn 3 (1980)                          Sci-Fi|Thriller
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

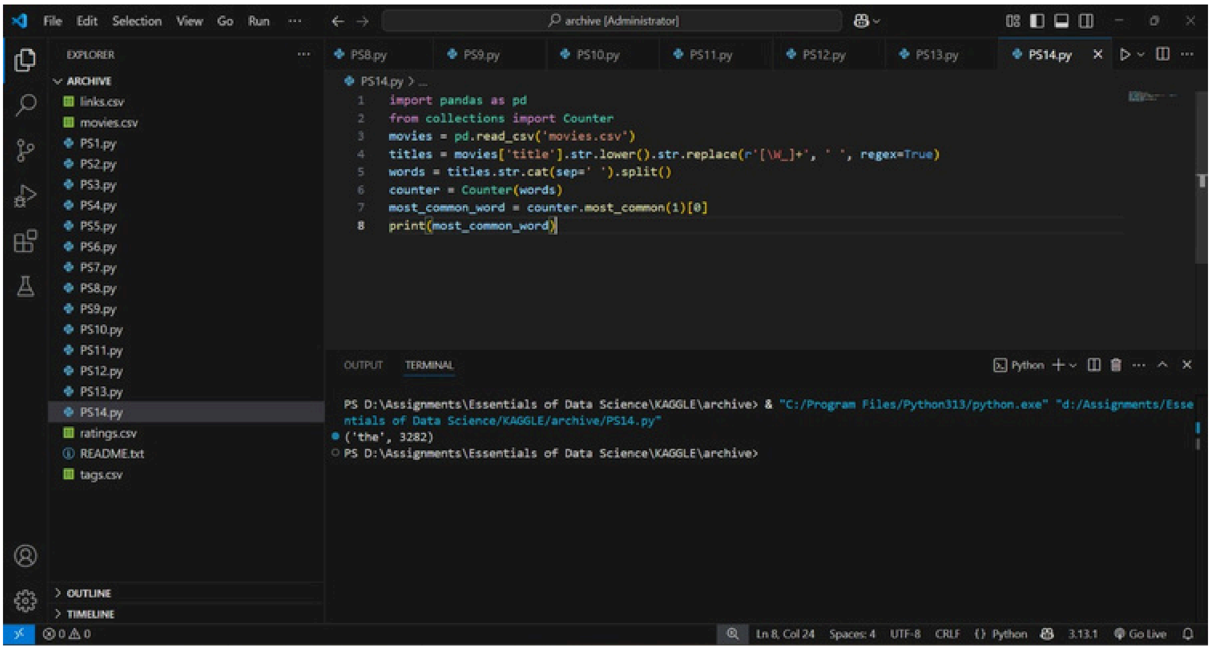
13. Count how many movies are label with “(no genres listed)” in the genres field.



```python
import pandas as pd
movies = pd.read_csv('movies.csv')
no_genre_movies = movies[movies['genres'] == "(no genres listed)"]
print(no_genre_movies.shape[0])
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS13.py"
34
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

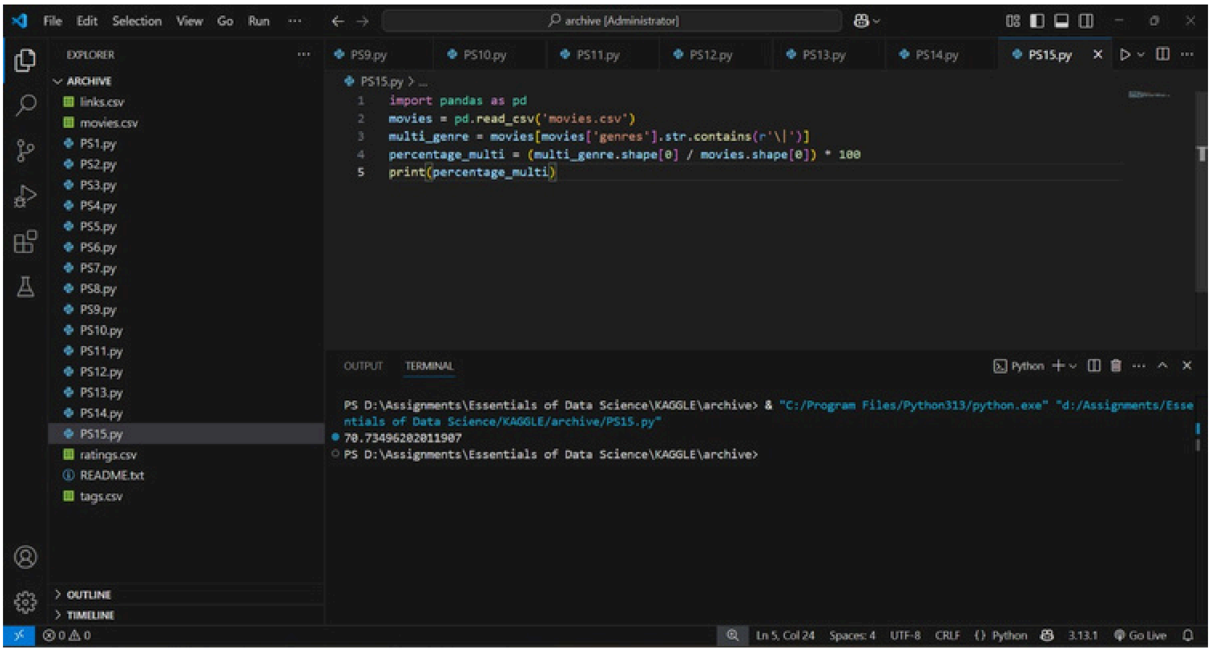
14. Determine the single most common word found in movie titles.



```python
import pandas as pd
from collections import Counter
movies = pd.read_csv('movies.csv')
titles = movies['title'].str.lower().str.replace(r'[\W_]+', ' ', regex=True)
words = titles.str.cat(sep=' ').split()
counter = Counter(words)
most_common_word = counter.most_common(1)[0]
print(most_common_word)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Esse
ntials of Data Science/KAGGLE/archive/PS14.py"
('the', 3282)
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

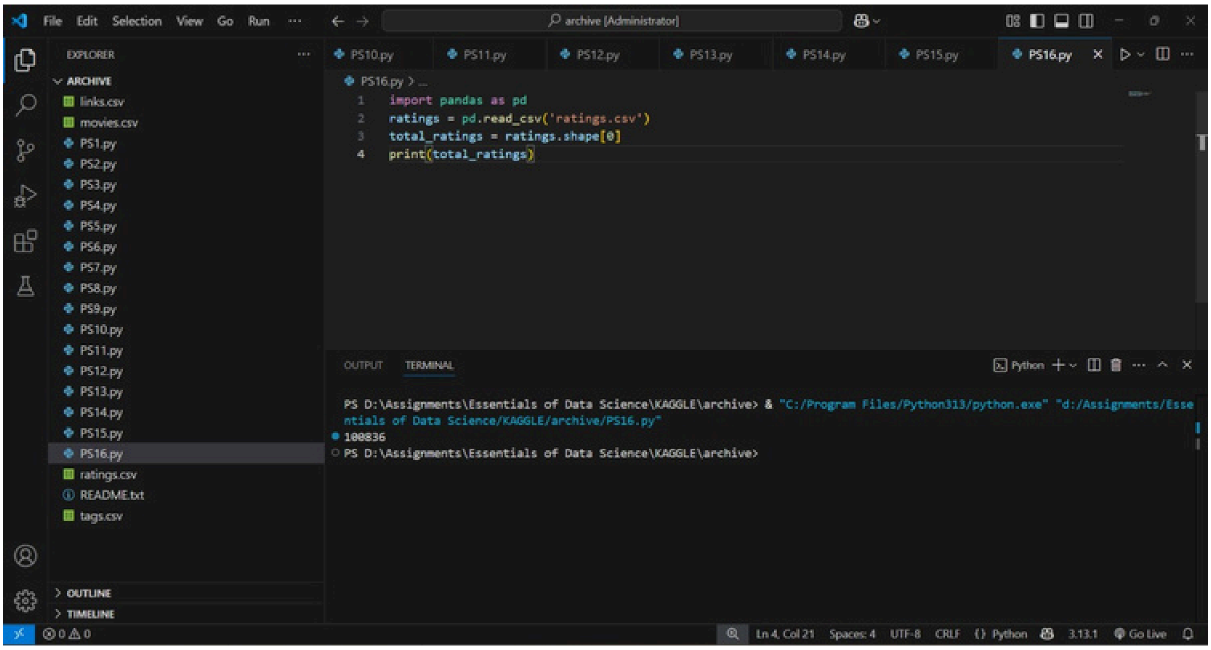
15. Calculate the percentage of movies that belong to multiple genres.



```python
import pandas as pd
movies = pd.read_csv('movies.csv')
multi_genre = movies[movies['genres'].str.contains(r'\|')]
percentage_multi = (multi_genre.shape[0] / movies.shape[0]) * 100
print(percentage_multi)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Esse
ntials of Data Science/KAGGLE/archive/PS15.py"
70.73496202011907
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

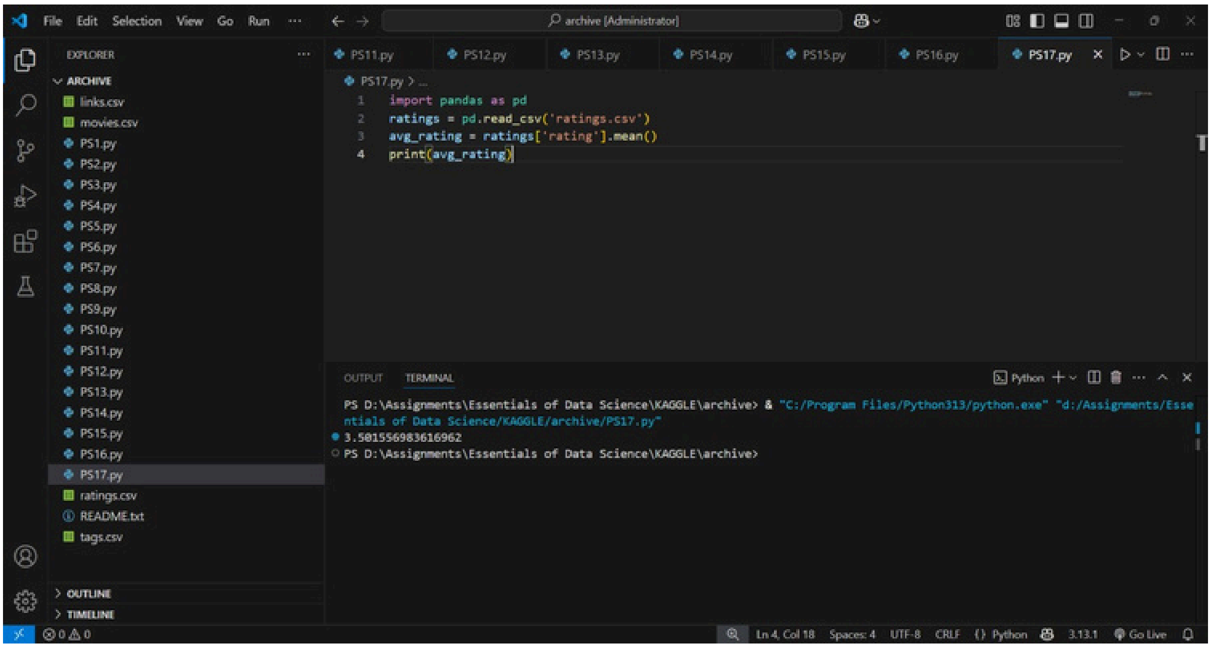
16. Count the total number of ratings



```python
import pandas as pd
ratings = pd.read_csv('ratings.csv')
total_ratings = ratings.shape[0]
print(total_ratings)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS16.py"
100836
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

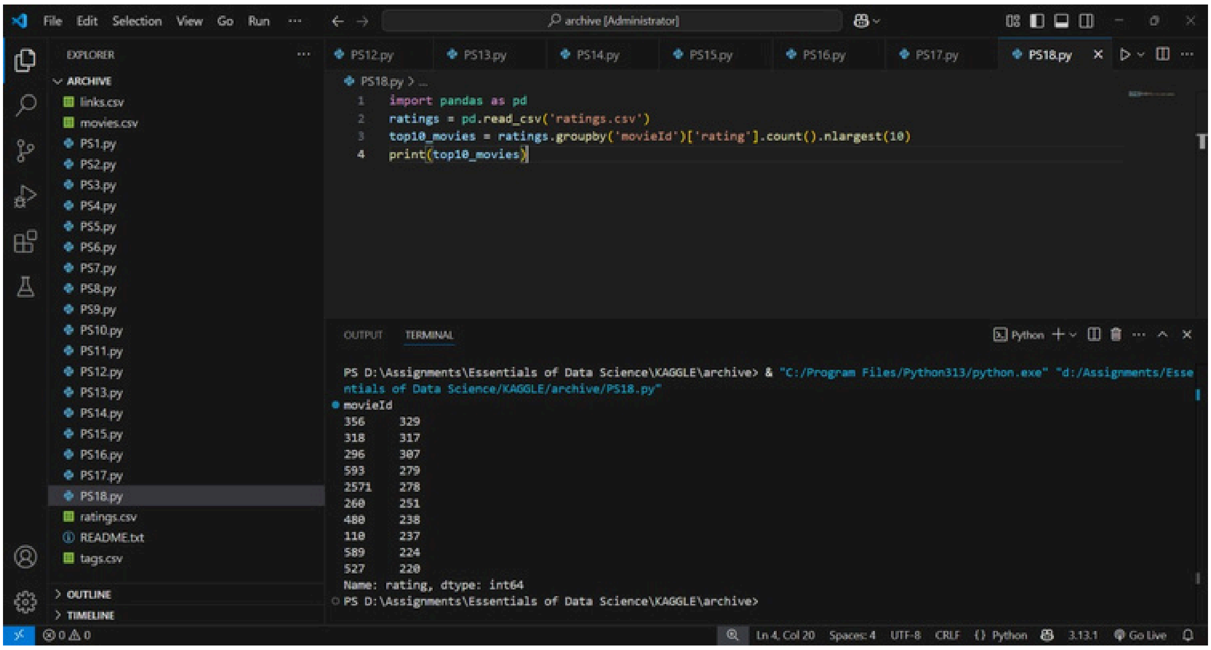
17. Calculate the average rating overall



```python
import pandas as pd
ratings = pd.read_csv('ratings.csv')
avg_rating = ratings['rating'].mean()
print(avg_rating)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS17.py"
3.501556983616962
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

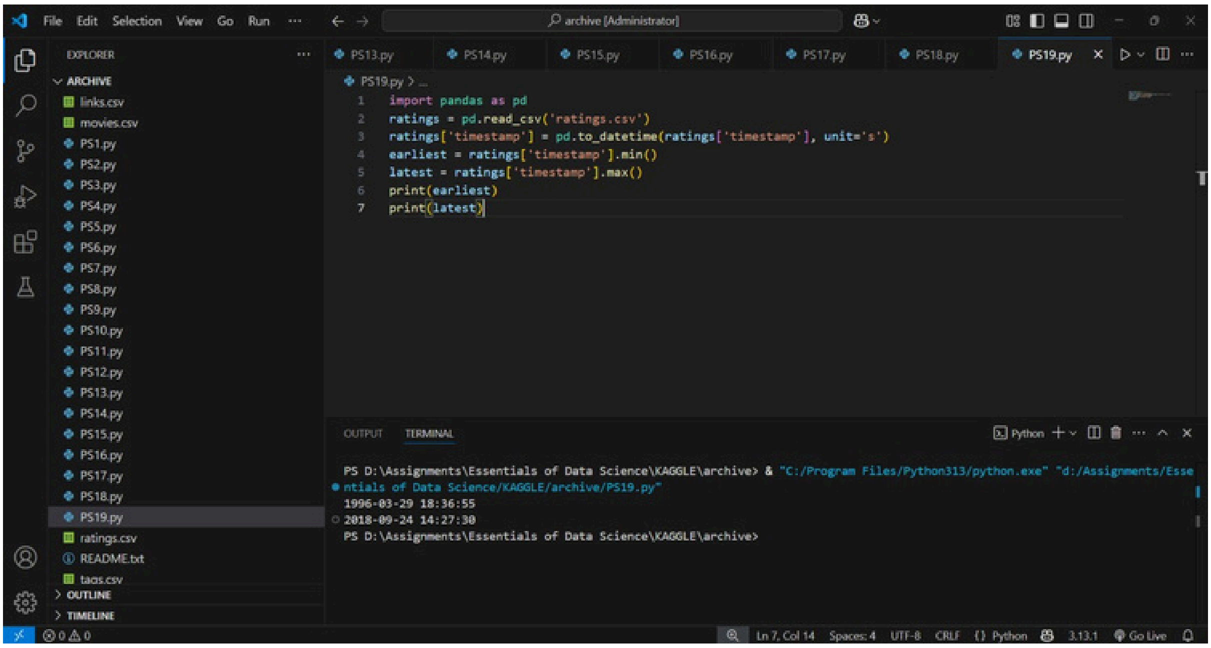
18. Identify the top 10 movies with the highest number of ratings



```python
import pandas as pd
ratings = pd.read_csv('ratings.csv')
top10_movies = ratings.groupby('movieId')['rating'].count().nlargest(10)
print(top10_movies)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS18.py"
movieId
356     329
318     317
296     307
593     279
2571    278
260     251
480     238
110     237
589     224
527     220
Name: rating, dtype: int64
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

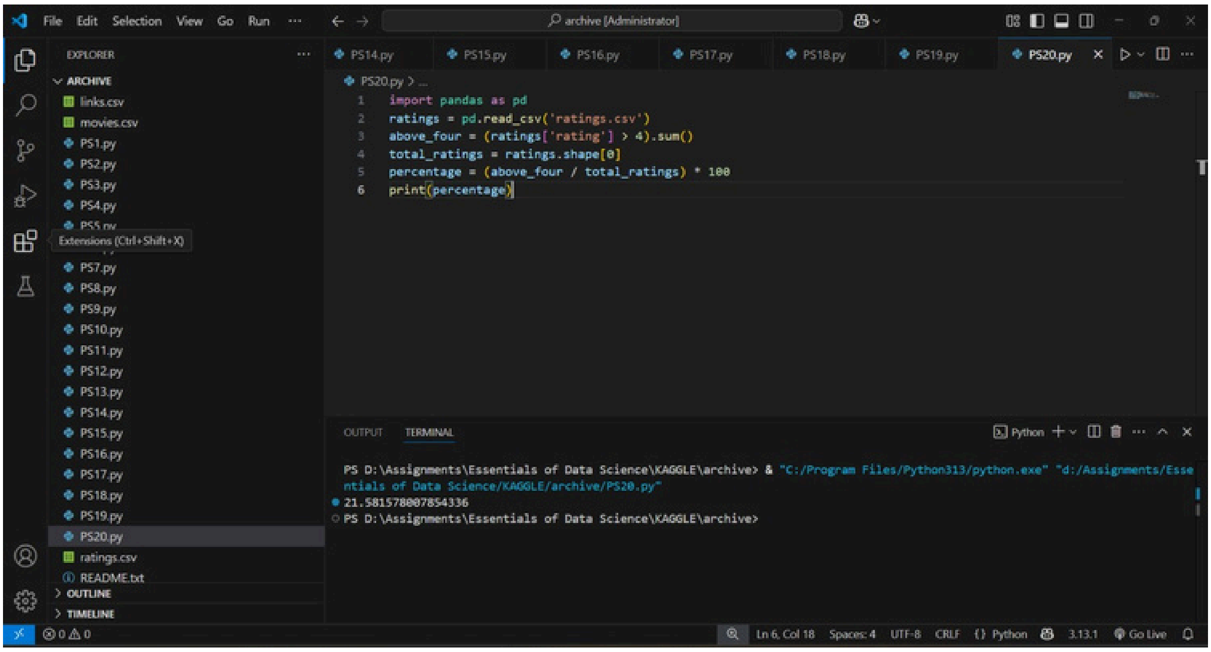
19. Find the earliest and latest rating timestamps



```python
import pandas as pd
ratings = pd.read_csv('ratings.csv')
ratings['timestamp'] = pd.to_datetime(ratings['timestamp'], unit='s')
earliest = ratings['timestamp'].min()
latest = ratings['timestamp'].max()
print(earliest)
print(latest)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS19.py"
1996-03-29 18:36:55
2018-09-24 14:27:30
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

20. Calculate the percentage of ratings that are above 4



```python
import pandas as pd
ratings = pd.read_csv('ratings.csv')
above_four = (ratings['rating'] > 4).sum()
total_ratings = ratings.shape[0]
percentage = (above_four / total_ratings) * 100
print(percentage)
```

```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Esse
ntials of Data Science/KAGGLE/archive/PS20.py"
21.581578007854336
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```