

# EDS THEORY ASSIGNMENT 1

Name: Sameer Sahebrao Sirsath

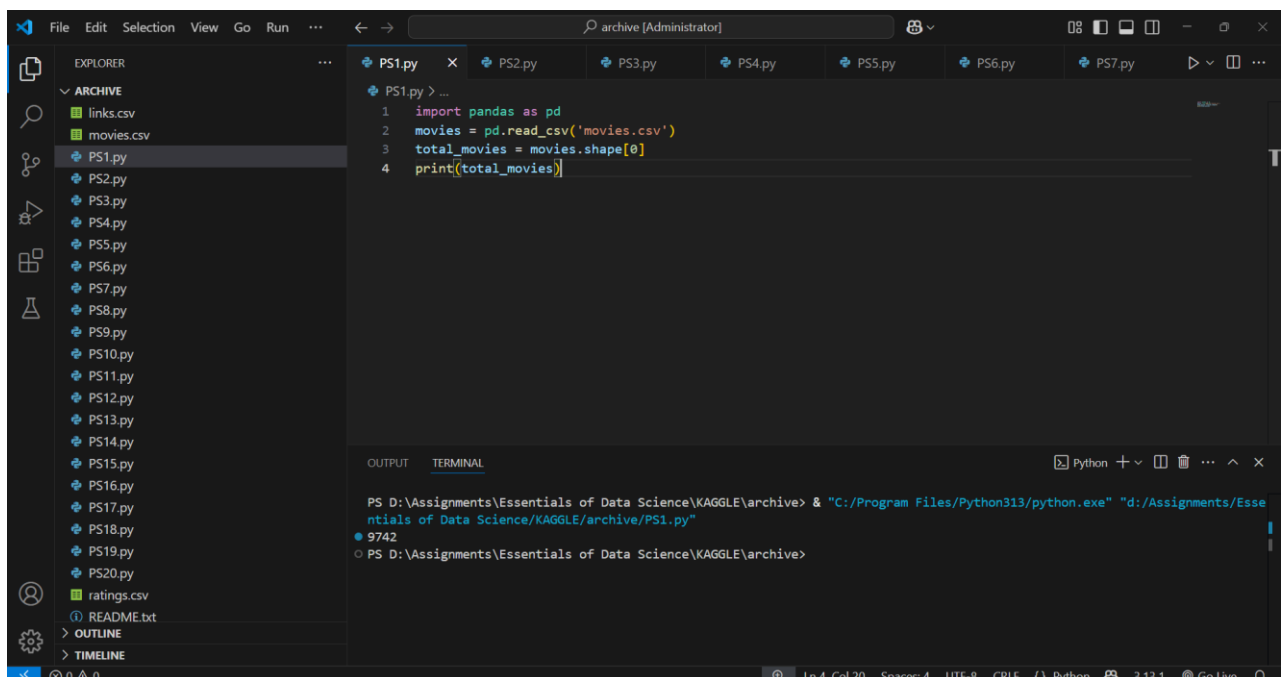
DIV: CS7

Roll No: CS7-75

PRN: 202401110023

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

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

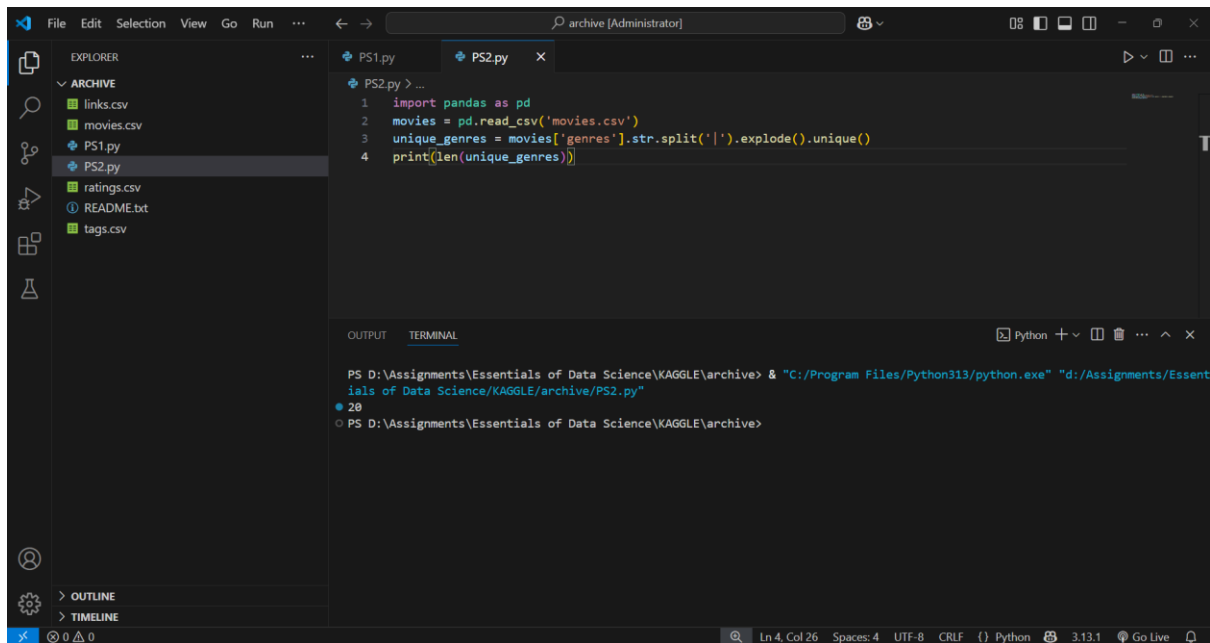


The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor in the center. The file explorer shows a folder named 'ARCHIVE' containing several files, including 'links.csv', 'movies.csv', and 'ratings.csv'. The code editor displays a Python script in a file named 'PS1.py'. The script imports pandas, reads the 'movies.csv' file, and prints the total number of movies. The output of the script is shown in the terminal at the bottom, indicating that there are 9742 movies in the dataset.

```
1 import pandas as pd
2 movies = pd.read_csv('movies.csv')
3 total_movies = movies.shape[0]
4 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.



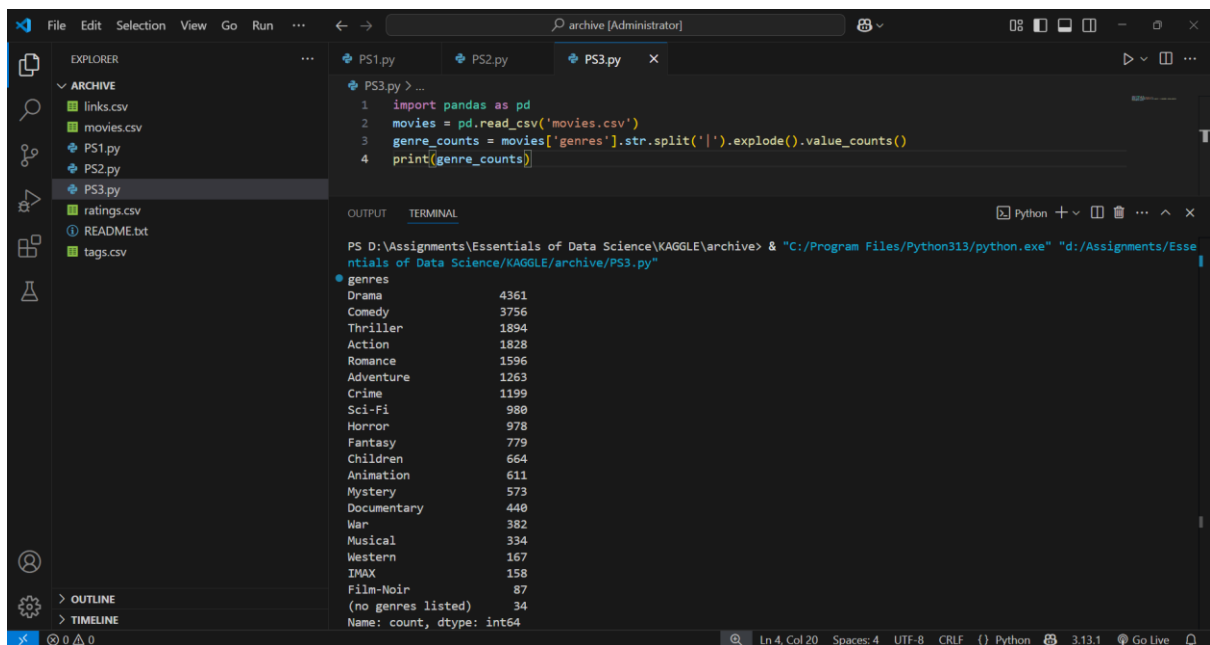
The screenshot shows a VS Code editor with a file explorer on the left containing 'links.csv', 'movies.csv', 'PS1.py', 'PS2.py', 'ratings.csv', 'README.txt', and 'tags.csv'. The main editor displays 'PS2.py' with the following code:

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

The terminal at the bottom shows the command execution and output:

```
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.



The screenshot shows a VS Code editor with a file explorer on the left containing 'links.csv', 'movies.csv', 'PS1.py', 'PS2.py', 'PS3.py', 'ratings.csv', 'README.txt', and 'tags.csv'. The main editor displays 'PS3.py' with the following code:

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

The terminal at the bottom shows the command execution and output:

```
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.

The screenshot shows a VS Code editor with a file explorer on the left containing files like links.csv, movies.csv, ratings.csv, and README.txt. The main editor displays PS4.py with the following code:

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

The terminal at the bottom shows the command to run the script and its output:

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

5. Extract the release year from the movie titles.

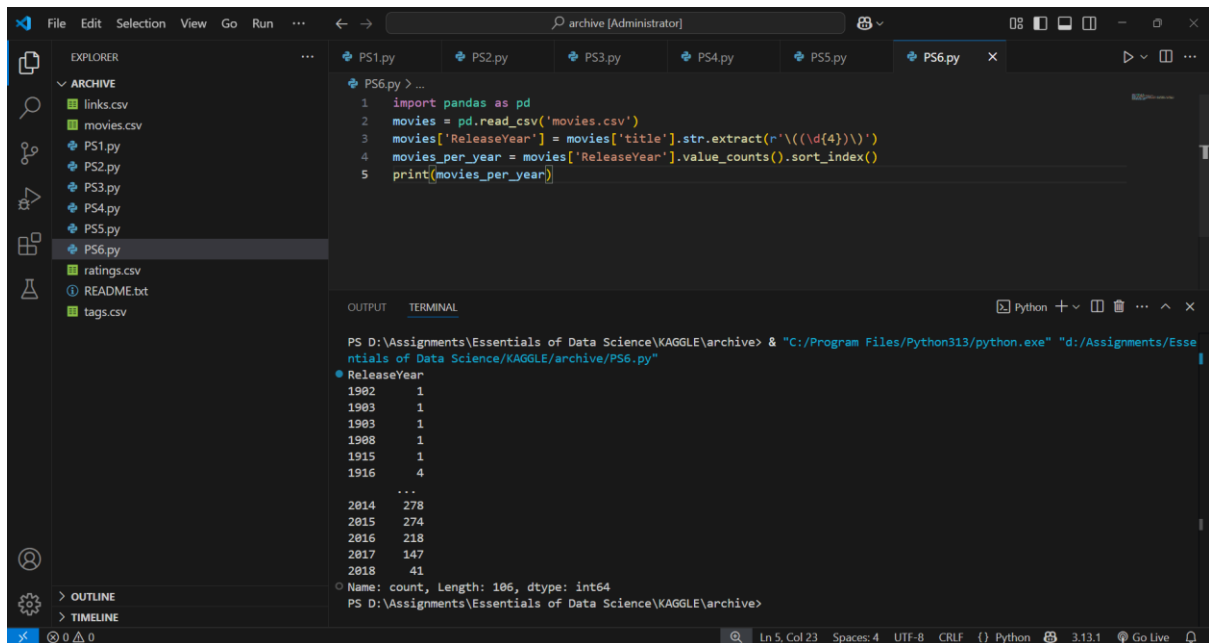
The screenshot shows a VS Code editor with a file explorer on the left. The main editor displays PS5.py with the following code:

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

The terminal at the bottom shows the command to run the script and its output:

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

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

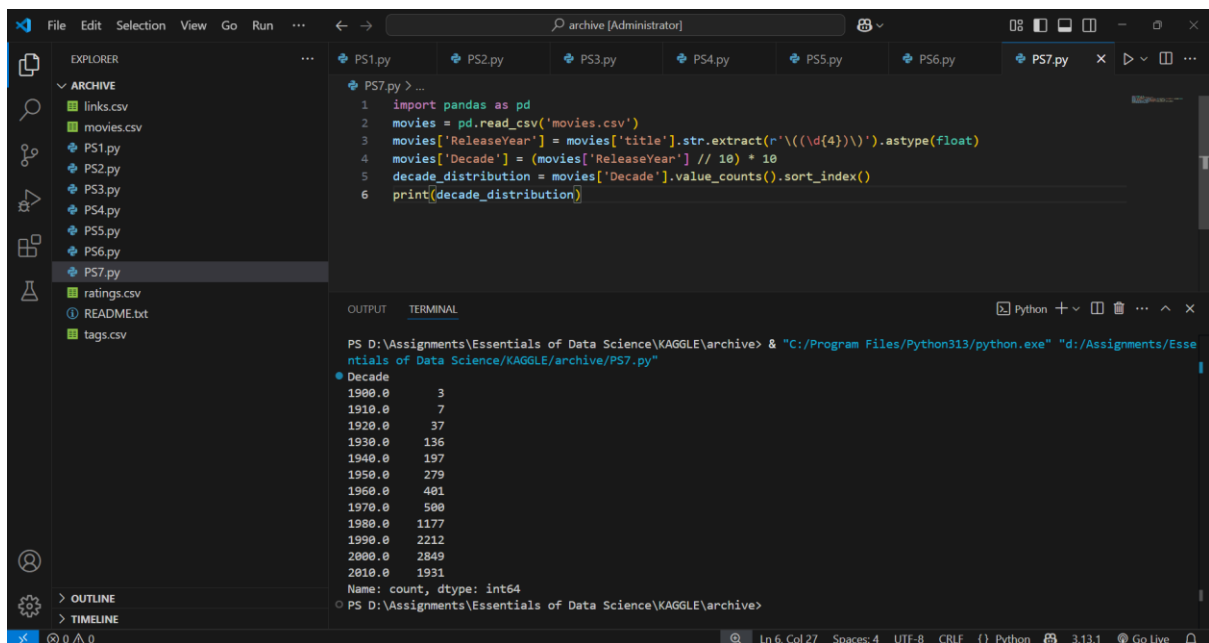


```
1 import pandas as pd
2 movies = pd.read_csv('movies.csv')
3 movies['ReleaseYear'] = movies['title'].str.extract(r'\((\d{4})\)').astype(float)
4 movies_per_year = movies['ReleaseYear'].value_counts().sort_index()
5 print(movies_per_year)
```

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

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



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

PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials 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    481
1970.0    580
1980.0   1177
1990.0   2212
2000.0   2849
2010.0    1931
Name: count, dtype: int64
```

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

The screenshot shows a VS Code editor with a file explorer on the left containing files like links.csv, movies.csv, PS1.py through PS8.py, ratings.csv, README.txt, and tags.csv. The main editor displays PS8.py with the following code:

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

The terminal output shows the execution of the script, displaying a DataFrame of comedy movies with columns movieId, title, and genres. The output is as follows:

movieId	title	genres
0	1	Toy Story (1995)
2	3	Grumpier Old Men (1995)
3	4	Waiting to Exhale (1995)
4	5	Father of the Bride Part II (1995)
6	7	Sabrina (1995)
...	...	...
9732	193565	Gintama: The Movie (2010)
9734	193571	Silver Spoon (2014)
9737	193581	Black Butler: Book of the Atlantic (2017)
9738	193583	No Game No Life: Zero (2017)
9741	193609	Andrew Dice Clay: Dice Rules (1991)

The terminal also shows the command used to run the script: `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"`.

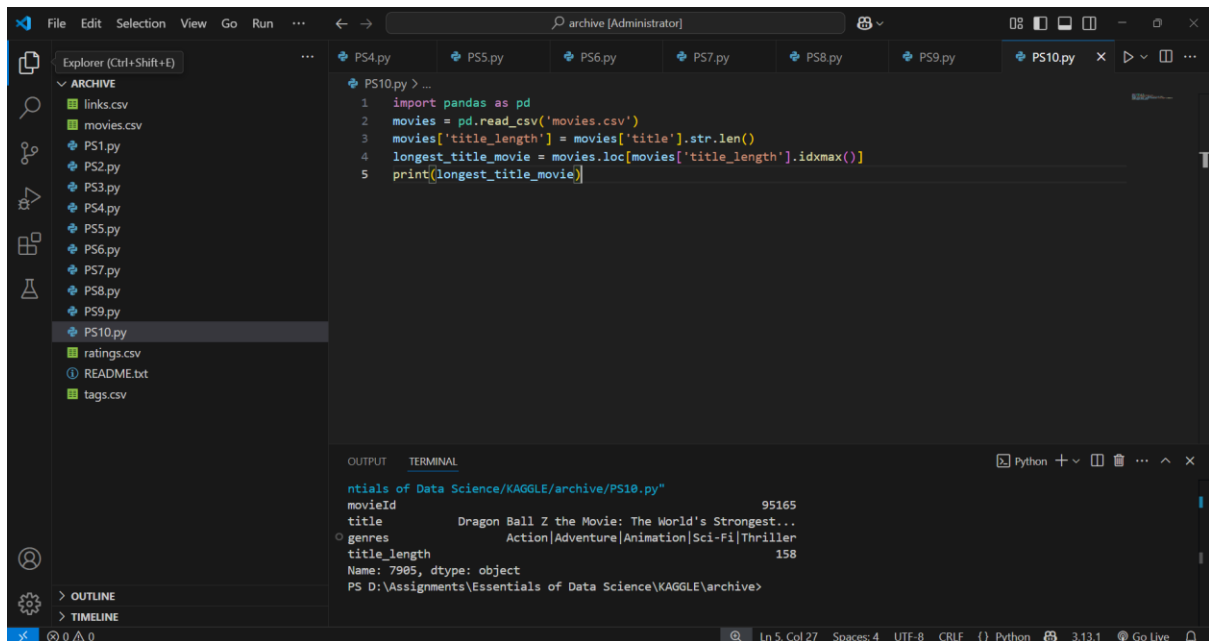
9. Count the number of movies that have more than one genre listed.

The screenshot shows a VS Code editor with a file explorer on the left containing files like links.csv, movies.csv, PS1.py through PS9.py, ratings.csv, README.txt, and tags.csv. The main editor displays PS9.py with the following code:

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

The terminal output shows the execution of the script, displaying the count of movies with more than one genre listed: 6891. The command used to run the script is: `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"`.

10. Identify the movie with the longest title.

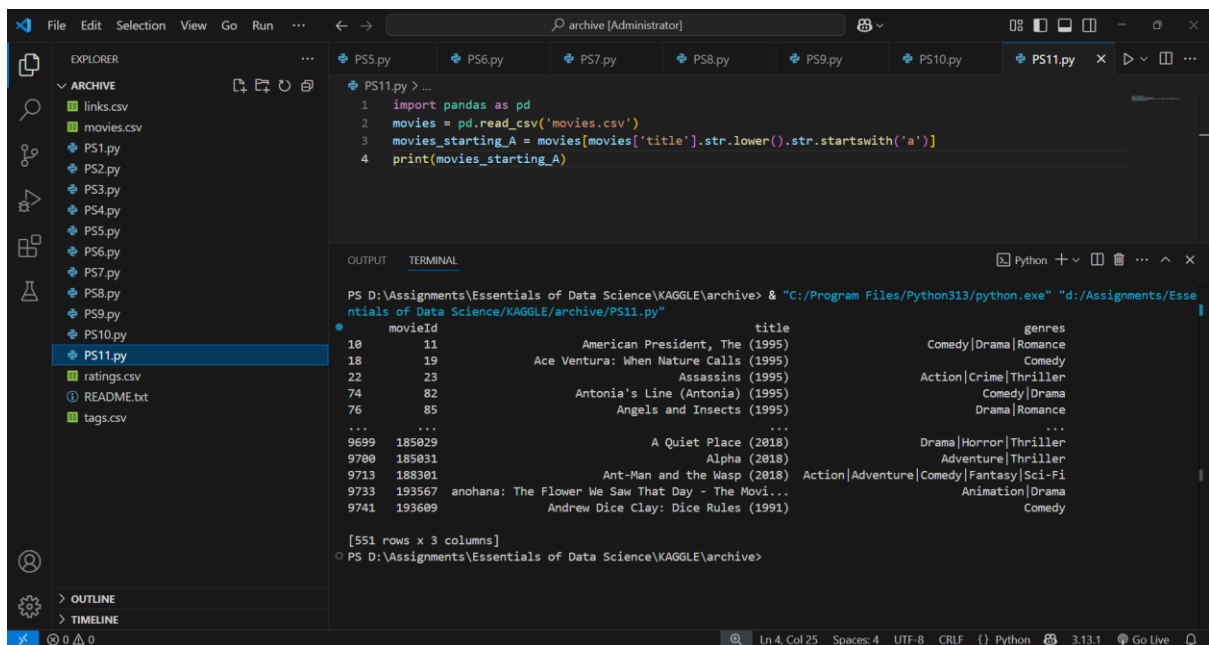


```
1 import pandas as pd
2 movies = pd.read_csv('movies.csv')
3 movies['title_length'] = movies['title'].str.len()
4 longest_title_movie = movies.loc[movies['title_length'].idxmax()]
5 print(longest_title_movie)
```

OUTPUT TERMINAL

```
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”.



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

OUTPUT TERMINAL

```
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.

```

1 import pandas as pd
2 movies = pd.read_csv('movies.csv')
3 duplicate_titles = movies[movies.duplicated(subset='title', keep=False)]
4 print(duplicate_titles)

```

movieId	title	genres
650	Emma (1996)	Comedy Drama Romance
2141	Saturn 3 (1980)	Adventure Sci-Fi Thriller
4169	Confessions of a Dangerous Mind (2002)	Comedy Crime Drama Thriller
5601	Emma (1996)	Romance
5854	Eros (2004)	Drama
5931	War of the Worlds (2005)	Action Adventure Sci-Fi Thriller
6932	War of the Worlds (2005)	Action Sci-Fi
9106	Confessions of a Dangerous Mind (2002)	Comedy Crime Drama Romance Thriller
9135	Eros (2004)	Drama Romance
9468	Saturn 3 (1980)	Sci-Fi Thriller

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

```

1 import pandas as pd
2 movies = pd.read_csv('movies.csv')
3 no_genre_movies = movies[movies['genres'] == "(no genres listed)"]
4 print(no_genre_movies.shape[0])

```

```

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

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

OUTPUT TERMINAL

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

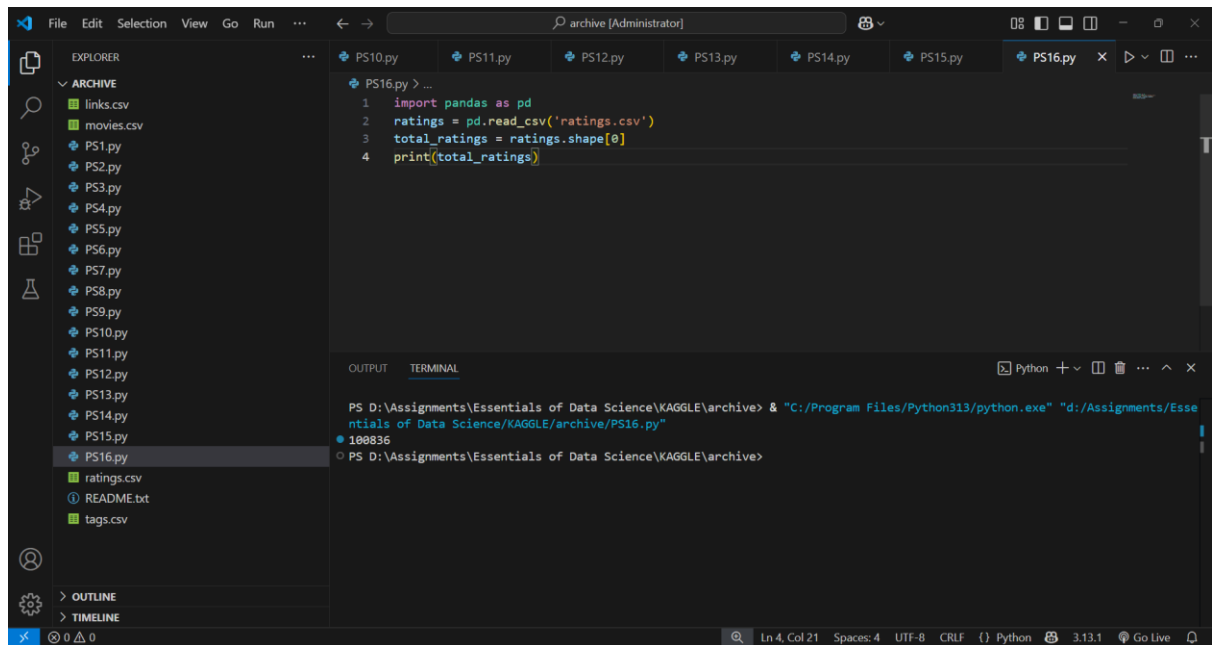
```
1 import pandas as pd
2 movies = pd.read_csv('movies.csv')
3 multi_genre = movies[movies['genres'].str.contains(r'|')]
4 percentage_multi = (multi_genre.shape[0] / movies.shape[0]) * 100
5 print(percentage_multi)
```

OUTPUT TERMINAL

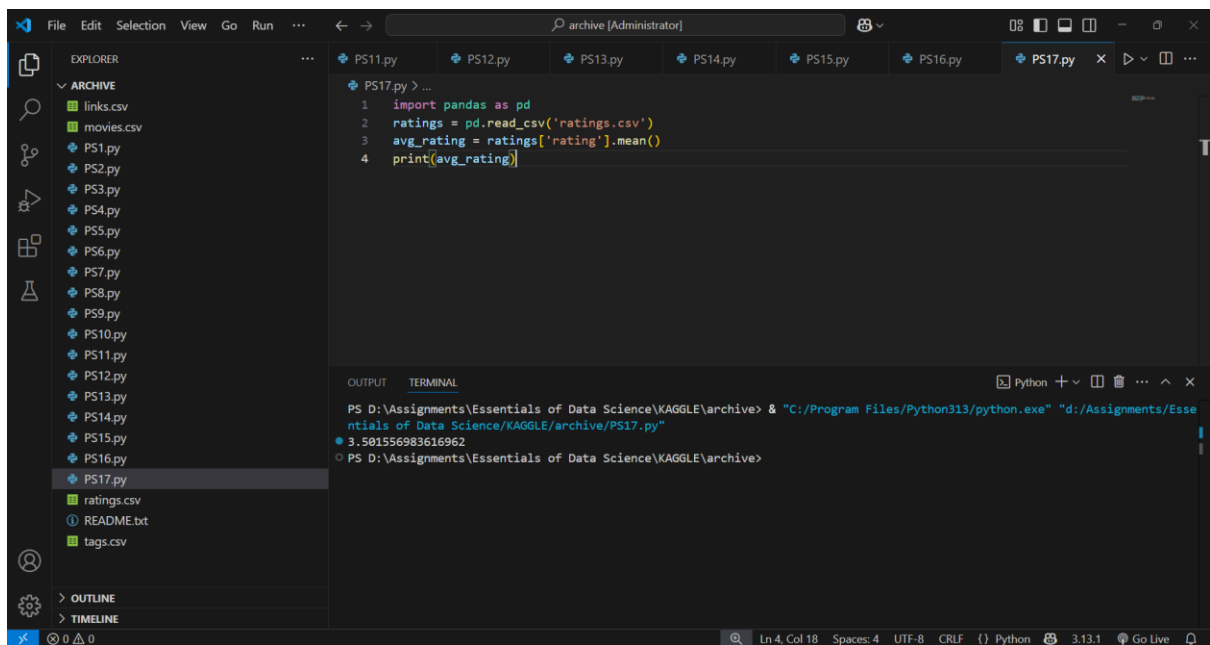
```
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> "C:/Program Files/Python313/python.exe" "d:/Assignments/Essentials of Data Science/KAGGLE/archive/PS15.py"
70.73496202011907
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>
```

16. Count the total number of ratings

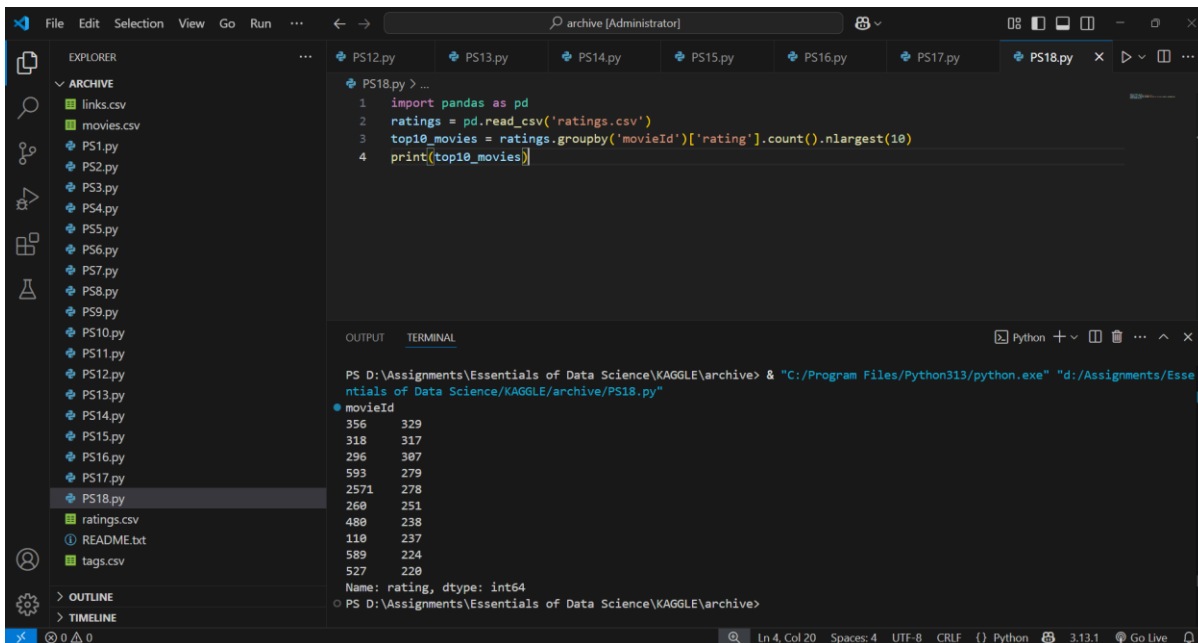




17. Calculate the average rating overall



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

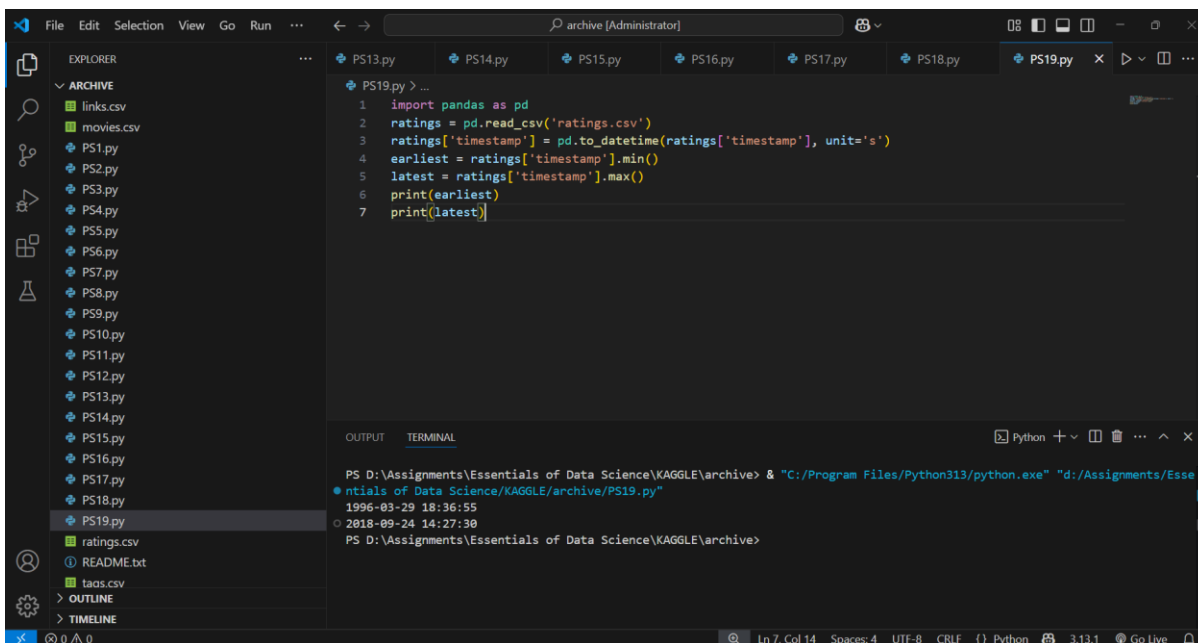


```
File Edit Selection View Go Run ... archive [Administrator]
EXPLORER
ARCHIVE
links.csv
movies.csv
PS1.py
PS2.py
PS3.py
PS4.py
PS5.py
PS6.py
PS7.py
PS8.py
PS9.py
PS10.py
PS11.py
PS12.py
PS13.py
PS14.py
PS15.py
PS16.py
PS17.py
PS18.py
ratings.csv
README.txt
tags.csv
OUTLINE
TIMELINE

PS18.py > ...
1 import pandas as pd
2 ratings = pd.read_csv('ratings.csv')
3 top10_movies = ratings.groupby('movieId')['rating'].count().nlargest(10)
4 print(top10_movies)

OUTPUT TERMINAL
Python
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Esse
ntials of Data Science/KAGGLE/archive/PS18.py"
movieId
356 329
318 317
296 307
593 279
2571 278
269 251
489 238
118 237
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



```
File Edit Selection View Go Run ... archive [Administrator]
EXPLORER
ARCHIVE
links.csv
movies.csv
PS1.py
PS2.py
PS3.py
PS4.py
PS5.py
PS6.py
PS7.py
PS8.py
PS9.py
PS10.py
PS11.py
PS12.py
PS13.py
PS14.py
PS15.py
PS16.py
PS17.py
PS18.py
PS19.py
ratings.csv
README.txt
tags.csv
OUTLINE
TIMELINE

PS19.py > ...
1 import pandas as pd
2 ratings = pd.read_csv('ratings.csv')
3 ratings['timestamp'] = pd.to_datetime(ratings['timestamp'], unit='s')
4 earliest = ratings['timestamp'].min()
5 latest = ratings['timestamp'].max()
6 print(earliest)
7 print(latest)

OUTPUT TERMINAL
Python
PS D:\Assignments\Essentials of Data Science\KAGGLE\archive> & "C:/Program Files/Python313/python.exe" "d:/Assignments/Esse
ntials 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

File Edit Selection View Go Run ...

archive [Administrator]

PS14.py PS15.py PS16.py PS17.py PS18.py PS19.py PS20.py

EXPLORER

ARCHIVE

links.csv movies.csv

PS1.py PS2.py PS3.py PS4.py PS5.py PS6.py PS7.py PS8.py PS9.py PS10.py PS11.py PS12.py PS13.py PS14.py PS15.py PS16.py PS17.py PS18.py PS19.py PS20.py

ratings.csv README.txt

OUTLINE

TIMELINE

PS20.py > ...

1 import pandas as pd

2 ratings = pd.read\_csv('ratings.csv')

3 above\_four = (ratings['rating'] > 4).sum()

4 total\_ratings = ratings.shape[0]

5 percentage = (above\_four / total\_ratings) \* 100

6 print(percentage)

OUTPUT TERMINAL

Python + -

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

21.581578007854336

PS D:\Assignments\Essentials of Data Science\KAGGLE\archive>

Ln 6, Col 18 Spaces: 4 UTF-8 CRLF {} Python 3.13.1 Go Live