Canopy Data Analysis Report

Ehsan Keikhavani (105665705)

1. Introduction and Understanding of Data Analysis and Python in this project:

Through this project, I gained a deeper understanding of how Python and data visualization can be powerful tools for analysing and presenting data to support business decisions. Python's flexibility allowed for seamless data cleaning, transformation, and analysis using libraries like Pandas, Matplotlib, Seaborn, and Bokeh. These tools provided an efficient way to manage large datasets and create clear, insightful visualizations.

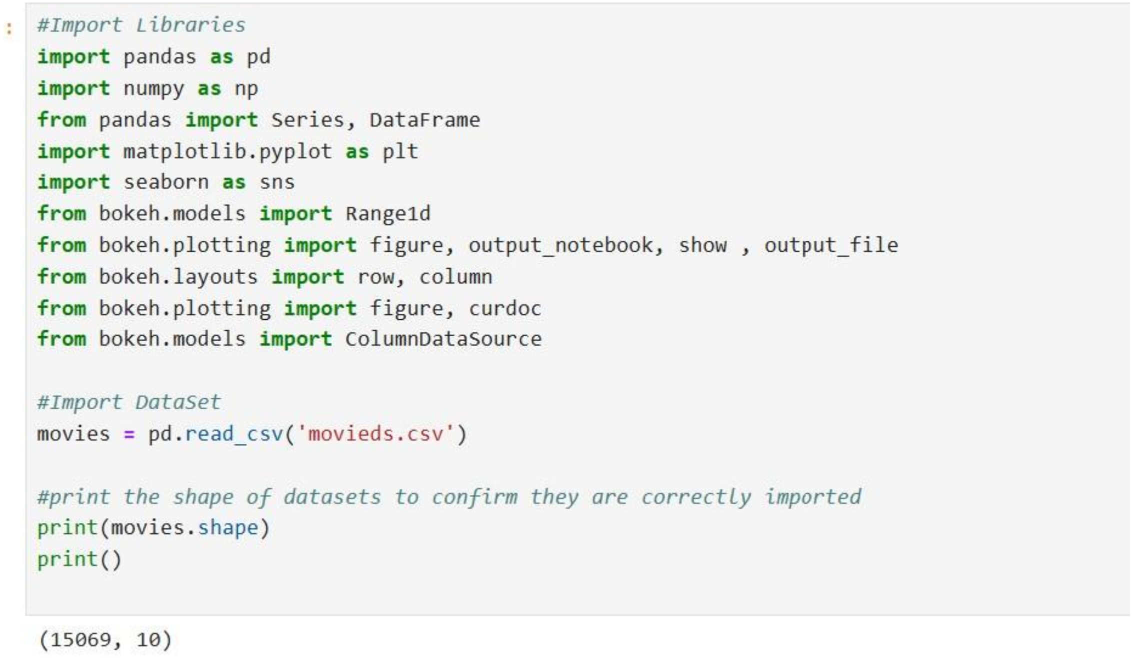
In the context of Canopy's goals, data visualization was essential to uncover trends and patterns in French-language movies, such as identifying high-rated films, understanding age-appropriate content, and exploring niche genres. For example, scatter plots highlighted relationships between variables like IMDb ratings and runtime, while bar plots clearly showed genre diversity and age group distribution. These visualizations not only made the data more accessible but also provided actionable insights aligned with Canopy's business objectives.

This experience reinforced the importance of systematic data preparation, such as handling missing values and standardizing formats, to ensure accuracy and reliability. It also highlighted the role of visual storytelling in communicating findings effectively. Overall, the project demonstrated how Python's data analysis and visualization capabilities can support evidence-based decision-making, making it an invaluable tool for businesses like Canopy seeking to establish a unique identity in competitive markets.

2. Approach Used to Create a Data Processing Pipeline:

2.1 Data Loading

A structured data processing pipeline in Jupyter Notebook, Using Python's Pandas and Matplotlib libraries to cleanse, transform, and analyze Canopy's movie data has been developed. Data and Libraries are imported, Print and Shape function been used to check if data imported correctly.

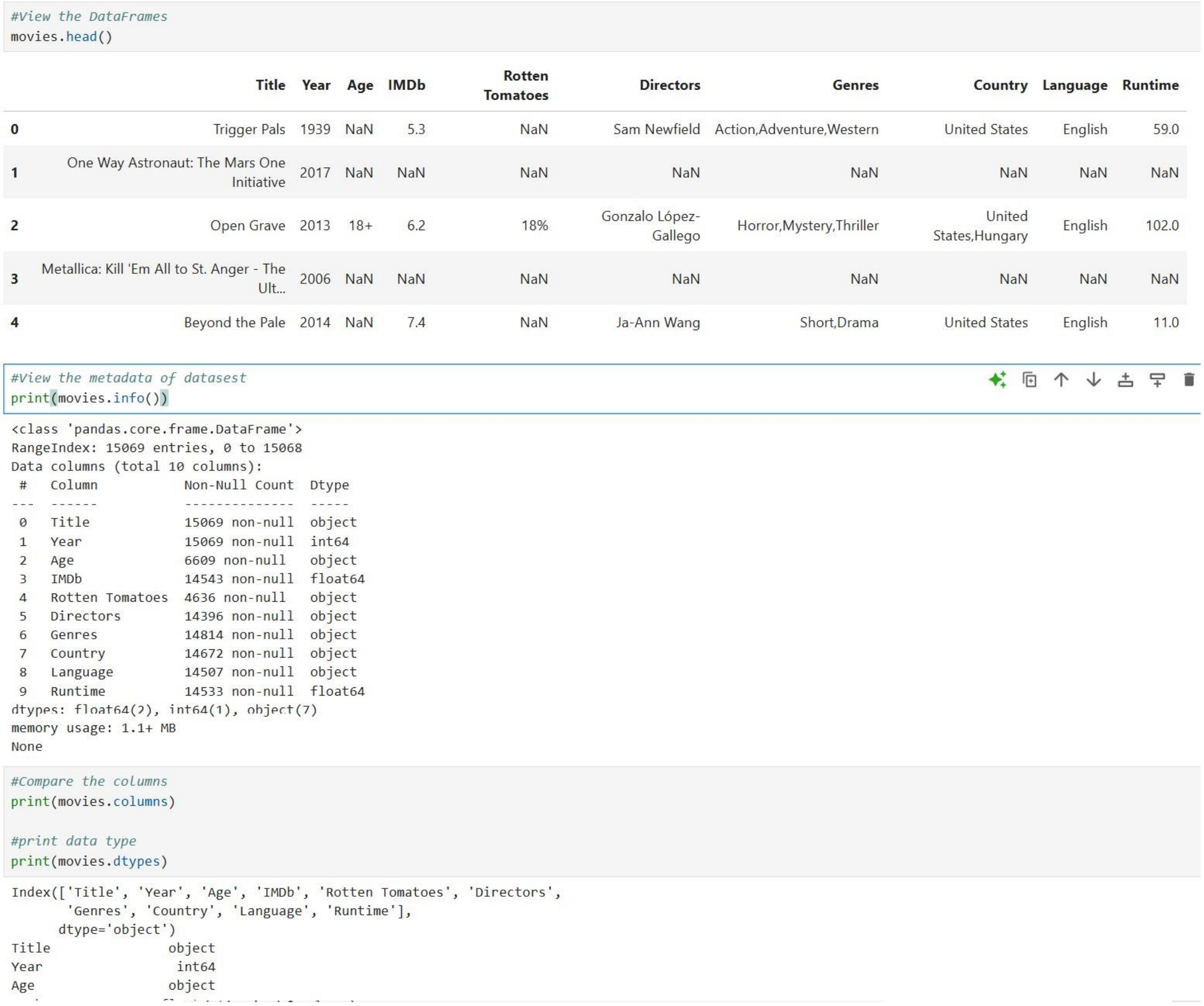


```
#Import Libraries
import pandas as pd
import numpy as np
from pandas import Series, DataFrame
import matplotlib.pyplot as plt
import seaborn as sns
from bokeh.models import Range1d
from bokeh.plotting import figure, output_notebook, show , output_file
from bokeh.layouts import row, column
from bokeh.plotting import figure, curdoc
from bokeh.models import ColumnDataSource

#Import DataSet
movies = pd.read_csv('movieds.csv')

#print the shape of datasets to confirm they are correctly imported
print(movies.shape)
print()
```

```
(15069, 10)
```

2.2 Initial Inspection

The first task was to load the datasets into Pandas data frames. This required using functions like info() ,head(),Columns and dtype to verify the data structure and identify critical columns for analysis, including Title, Age, IMDb, Genres, Language and Runtime.



```
#View the DataFrames
movies.head()
```

	Title	Year	Age	IMDb	Rotten Tomatoes	Directors	Genres	Country	Language	Runtime
0	Trigger Pals	1939	NaN	5.3	NaN	Sam Newfield	Action,Adventure,Western	United States	English	59.0
1	One Way Astronaut: The Mars One Initiative	2017	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Open Grave	2013	18+	6.2	18%	Gonzalo López-Gallego	Horror,Mystery,Thriller	United States,Hungary	English	102.0
3	Metallica: Kill 'Em All to St. Anger - The Ult...	2006	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Beyond the Pale	2014	NaN	7.4	NaN	Ja-Ann Wang	Short,Drama	United States	English	11.0

```
#View the metadata of datasest
print(movies.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15069 entries, 0 to 15068
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Title            15069 non-null  object 
 1   Year             15069 non-null  int64  
 2   Age              6609 non-null   object 
 3   IMDb             14543 non-null  float64
 4   Rotten Tomatoes  4636 non-null   object 
 5   Directors        14396 non-null  object 
 6   Genres           14814 non-null  object 
 7   Country          14672 non-null  object 
 8   Language         14507 non-null  object 
 9   Runtime          14533 non-null  float64
dtypes: float64(2), int64(1), object(7)
memory usage: 1.1+ MB
None
```

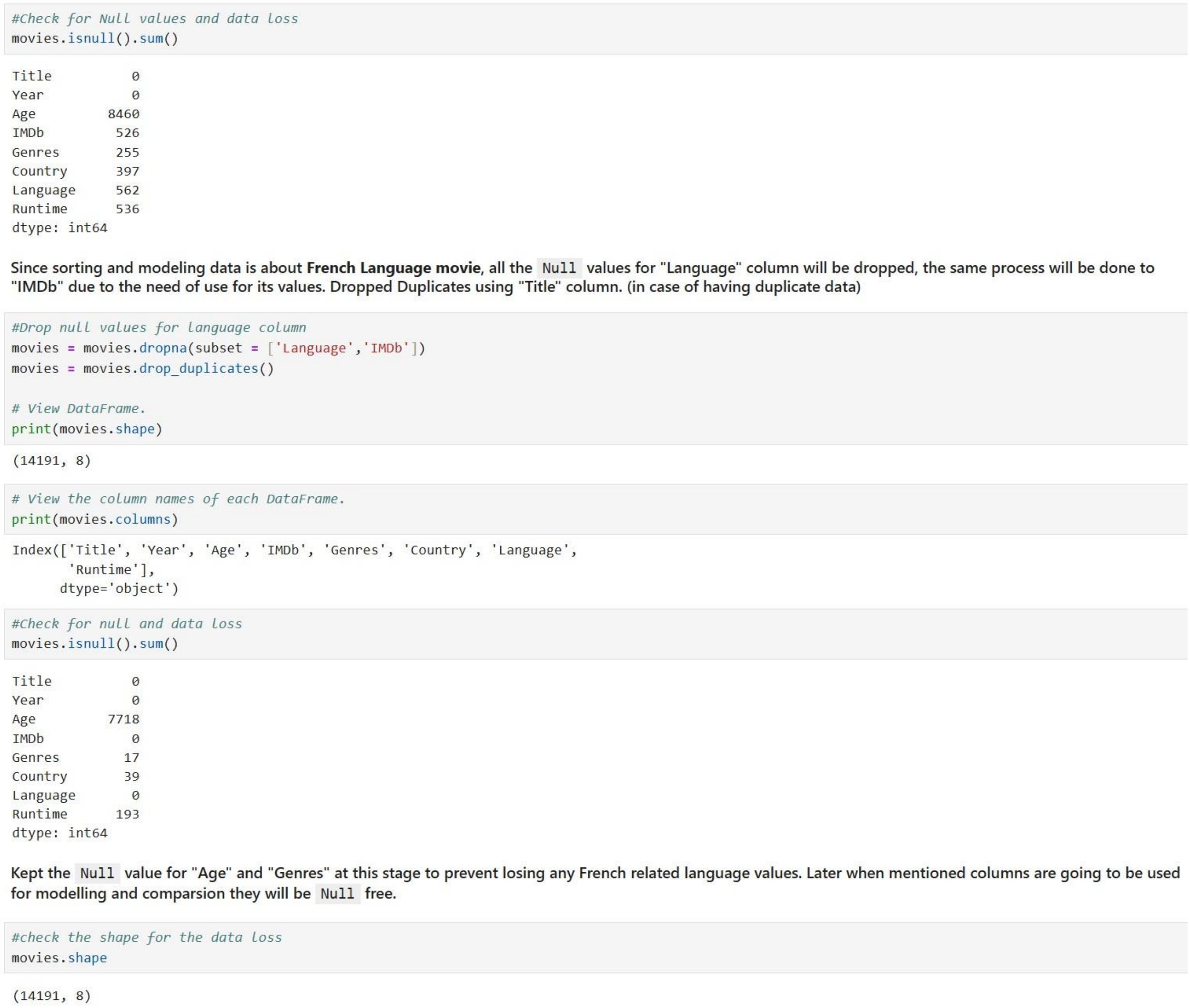
```
#Compare the columns
print(movies.columns)

#print data type
print(movies.dtypes)
```

```
Index(['Title', 'Year', 'Age', 'IMDb', 'Rotten Tomatoes', 'Directors',
       'Genres', 'Country', 'Language', 'Runtime'],
      dtype='object')
Title               object
Year                 int64
Age                 object
```

2.3 Data Wrangling

Data cleansing focused on addressing missing and inconsistent values across critical columns such as Age, Genres, IMDb, Language.

Key transformations included filtering for French-language content, handling missing data, and duplicates.



```
#Check for Null values and data loss
movies.isnull().sum()
```

```
Title          0
Year           0
Age         8460
IMDb         526
Genres       255
Country      397
Language     562
Runtime      536
dtype: int64
```

Since sorting and modeling data is about **French Language movie**, all the `Null` values for "Language" column will be dropped, the same process will be done to "IMDb" due to the need of use for its values. Dropped Duplicates using "Title" column. (in case of having duplicate data)

```
#Drop null values for language column
movies = movies.dropna(subset = ['Language','IMDb'])
movies = movies.drop_duplicates()

# View DataFrame.
print(movies.shape)
```

```
(14191, 8)
```

```
# View the column names of each DataFrame.
print(movies.columns)
```

```
Index(['Title', 'Year', 'Age', 'IMDb', 'Genres', 'Country', 'Language',
       'Runtime'],
      dtype='object')
```

```
#Check for null and data loss
movies.isnull().sum()
```

```
Title          0
Year           0
Age         7718
IMDb           0
Genres        17
Country       39
Language       0
Runtime      193
dtype: int64
```

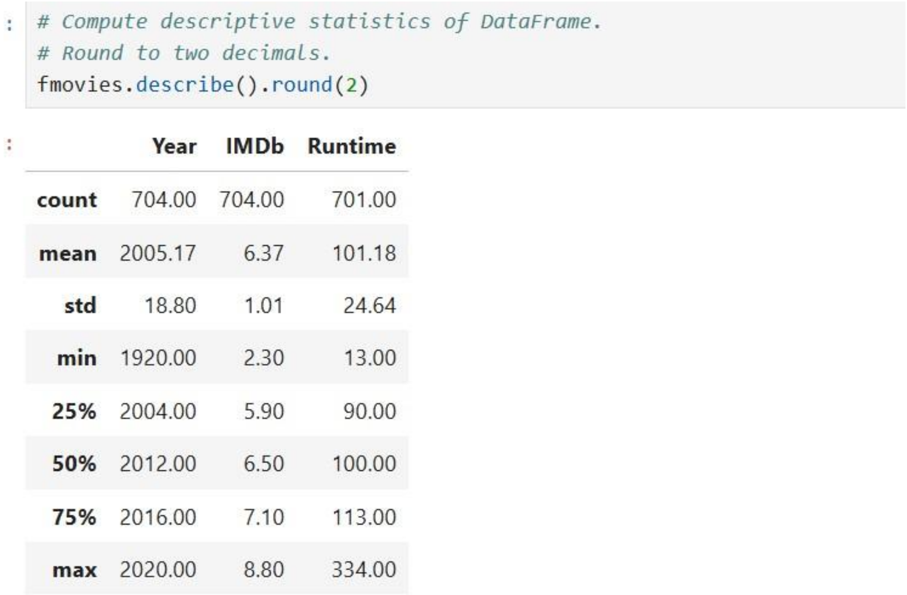
Kept the `Null` value for "Age" and "Genres" at this stage to prevent losing any French related language values. Later when mentioned columns are going to be used for modelling and comparsion they will be `Null` free.

```
#check the shape for the data loss
movies.shape
```

```
(14191, 8)
```

2.4 Data Description

A new dataset Created from the merged datasets consists of essential columns and specifically for French Language movies. Furthermore, calculating summary statistics.

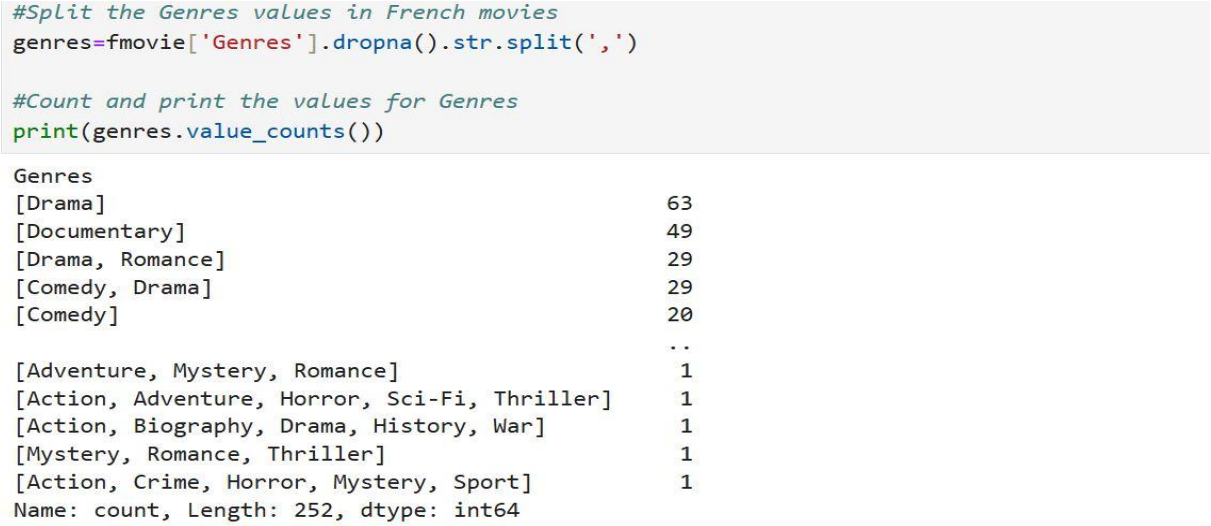


```
# Compute descriptive statistics of DataFrame.
# Round to two decimals.
fmovies.describe().round(2)
```

	Year	IMDb	Runtime
count	704.00	704.00	701.00
mean	2005.17	6.37	101.18
std	18.80	1.01	24.64
min	1920.00	2.30	13.00
25%	2004.00	5.90	90.00
50%	2012.00	6.50	100.00
75%	2016.00	7.10	113.00
max	2020.00	8.80	334.00

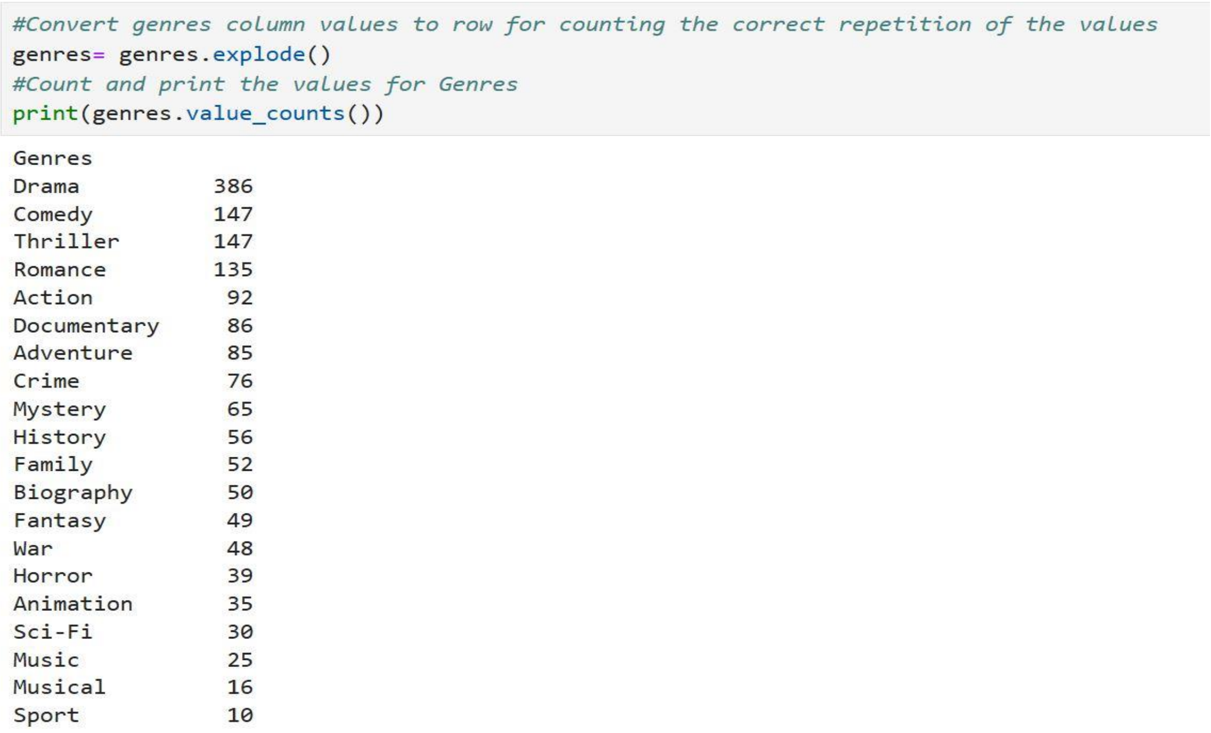
2.5 Data Modeling and Transformation

To enable more detailed analysis of genres, entries in the Genres column that contained multiple categories were separated by splitting and counting each genre individually. This transformation allowed for an accurate calculation of genre repetition, highlighting the most available and underrepresented genres.



```
#Split the Genres values in French movies
genres=fmovie['Genres'].dropna().str.split(',')

#Count and print the values for Genres
print(genres.value_counts())
```

```
Genres
[Drama]                                           63
[Documentary]                                     49
[Drama, Romance]                                  29
[Comedy, Drama]                                   29
[Comedy]                                          20
                                                  ..
[Adventure, Mystery, Romance]                      1
[Action, Adventure, Horror, Sci-Fi, Thriller]      1
[Action, Biography, Drama, History, War]           1
[Mystery, Romance, Thriller]                       1
[Action, Crime, Horror, Mystery, Sport]            1
Name: count, Length: 252, dtype: int64
```



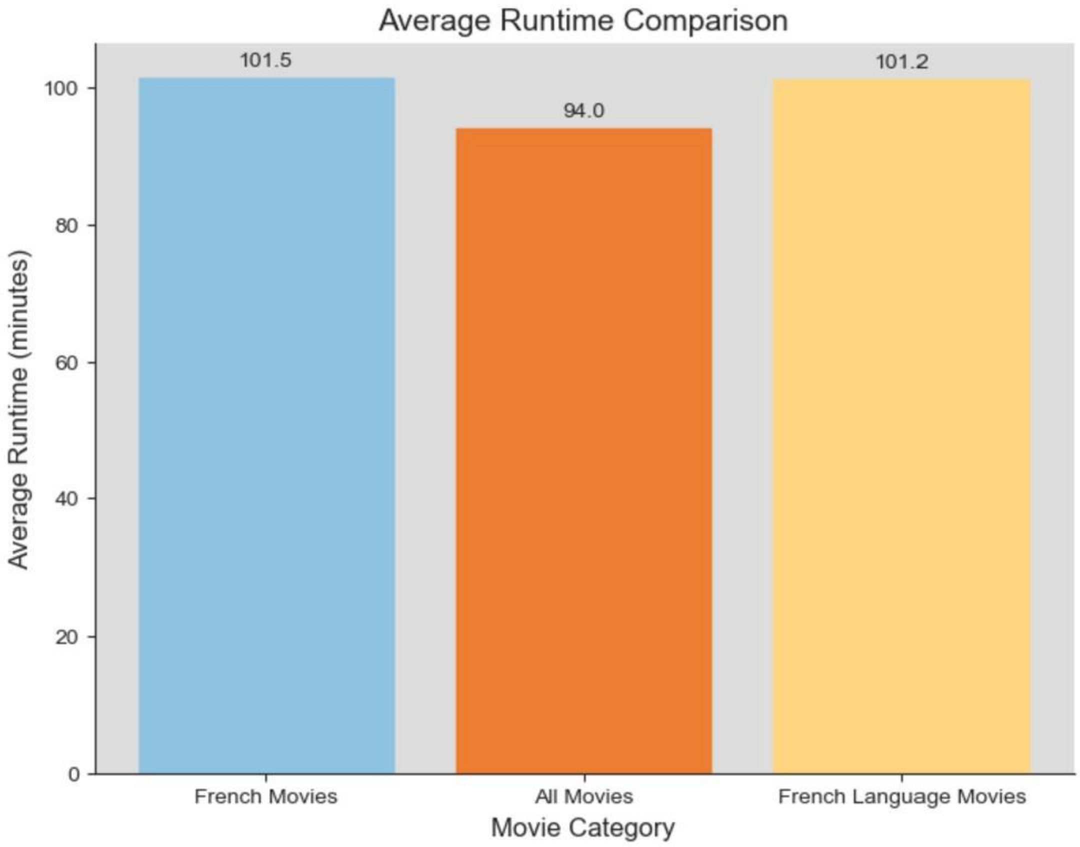
```
#Convert genres column values to row for counting the correct repetition of the values
genres= genres.explode()
#Count and print the values for Genres
print(genres.value_counts())
```

```
Genres
Drama          386
Comedy         147
Thriller       147
Romance        135
Action          92
Documentary     86
Adventure       85
Crime           76
Mystery         65
History         56
Family          52
Biography       50
Fantasy         49
War             48
Horror          39
Animation       35
Sci-Fi          30
Music           25
Musical         16
Sport           10
```

2.6 Data visualization Exploratory Visualizations:

Purpose: Used during the analysis phase to uncover patterns, trends, and relationships in the data.

Audience: Typically, the data analyst or team working on the dataset. Focused on discovery rather than communication. Allowing for comparisons and deeper dives into the data.



Bar Plot for Average Runtime and IMDb vs Movie Category

- **Categorical Data**

Movie categories (e.g., "French Movies" vs. "All Movies") are discrete, categorical variables.

A bar plot is ideal for visualizing comparisons across such categories because it clearly shows the differences.

- **Easy Comparison**

A bar plot allows for quick and intuitive comparisons of the average runtimes between the categories.

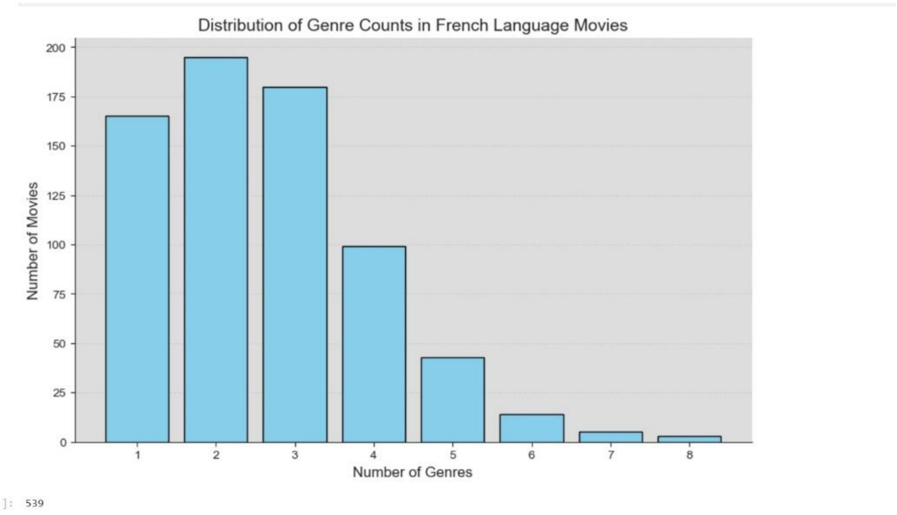
Are French movies generally longer or shorter than the overall average?

This comparison is immediately visible from the heights of the bars.

- **Simplistic and Focused**

A bar plot is simple and avoids unnecessary clutter, making it ideal for audiences who want straightforward insights into runtime differences. It provides a focused view on averages without the distraction of individual data points.

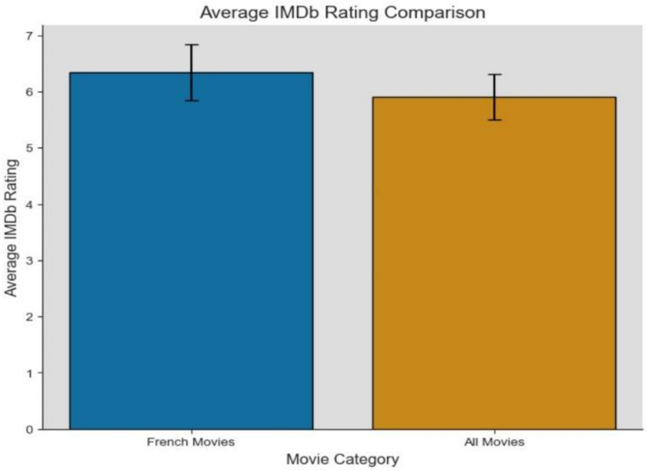
In addition, Annotation been used to pinpoint the important data amount. Different color assigned to the bars for better visualization.



```
]: 539
```

Understanding the distribution of genre counts helps Canopy identify trends in movie diversity:

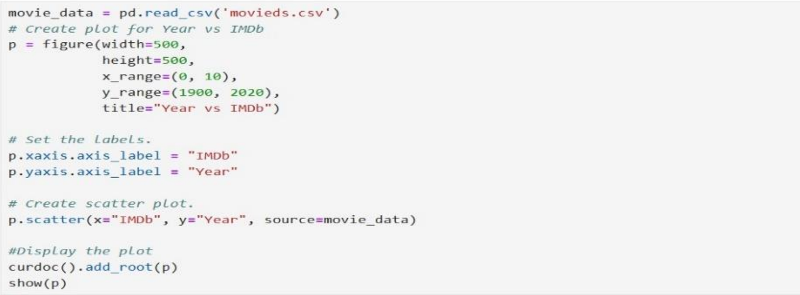
If most successful movies are multi-genre, Canopy might focus on acquiring or producing such content. Conversely, if single-genre movies dominate, Canopy can emphasize their niche appeal.



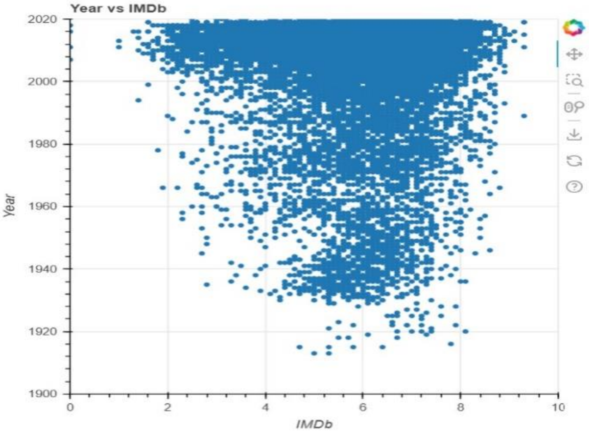
In addition, for Average Rating and Movie Category, to cover any possible error, standard deviation been calculated and used in the chart.

Why Use a Scatter Plot for Year and IMDb Rating?

- **Visualizing Trends Over Time A scatter plot helps identify patterns in movie ratings over the years.**



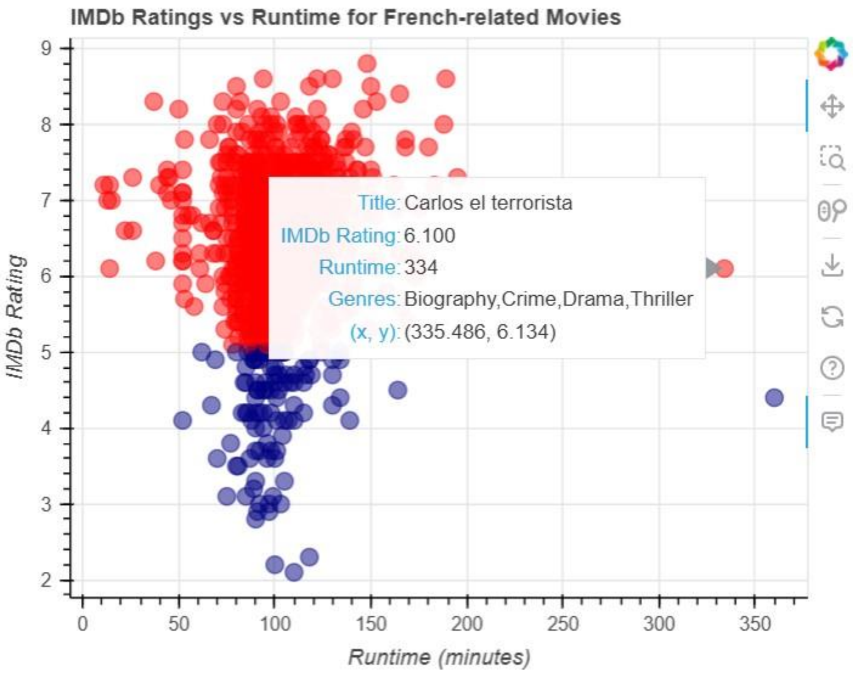
```
movie_data = pd.read_csv('movieds.csv')
# Create plot for Year vs IMDb
p = figure(width=500,
           height=500,
           x_range=(0, 10),
           y_range=(1900, 2020),
           title="Year vs IMDb")

# Set the labels.
p.xaxis.axis_label = "IMDb"
p.yaxis.axis_label = "Year"

# Create scatter plot.
p.scatter(x="IMDb", y="Year", source=movie_data)

#Display the plot
curdoc().add_root(p)
show(p)
```



Are older movies generally rated higher or lower compared to newer ones?

Does a particular time period show a spike in highly rated movies?

- **Understanding Audience Preferences: IMDb ratings reflect audience approval and preferences. By plotting them against release years: We can assess whether audiences prefer recent movies or classics. It provides insight into whether the top-rated movies belong to a specific period, helping curate content for viewers seeking nostalgic or contemporary experiences.**
- **Spotting Outliers A scatter plot easily highlights outliers:**
  For instance, a highly rated movie from an unexpected era might stand out, offering insight into unique patterns. Similarly, poorly rated movies in a strong year might suggest anomalies or trends to avoid.
- **Identifying Gaps: The scatter plot reveals gaps where there are few or no high-rated movies by certain runtime. The chart highlights movies with IMDb ratings above 5 for better visualization, while also showcasing areas with significantly high or low runtimes.**



Title: Carlos el terrorista
IMDb Rating: 6.100
Runtime: 334
Genres: Biography,Crime,Drama,Thriller
(x, y): (335.486, 6.134)

Canopy can use this information to decide if their curated French-language films should focus on certain time periods. If high-rated movies are concentrated in specific decades (or runtime), Canopy might target films from those years (or durations) to attract a larger audience.

Final data outputs exported as HTML and Jupyter notebook file for the final step which is Visualization.

3. Conclusions

By addressing the following gaps, Canopy can better fulfill its objectives of being a versatile platform for French-language film lovers while also standing out with unique, original offerings.

"In the U.S. alone, revenue from video-on-demand services was approximately $70 billion in 2023, highlighting the critical role of streaming in daily life ". (Statista, 2023)

The scatter plot between runtime and IMDb ratings indicates that movies with runtimes between 90 and 120 minutes tend to receive higher ratings, suggesting an optimal runtime range for well-received films.

The bar plot for IMDb ratings and category provides a measure of audience satisfaction and quality. Comparing categories helps Canopy, determine French-language (average rating6.34) movies are performing better than the overall market (average rating 5.90).

The bar plot for runtime and category demonstrate on average French movies (101.5 min) has higher time than other categories (94 min) which are in the high rated area of IMDb.

Recommendations

- **Expand the catalogue to include family-friendly and teen-focused content to attract a broader demographic, particularly families and younger viewers.**
- **Understanding the relationship between Runtime and IMDb ratings helps Canopy decide what type of movies to prioritize: If higher-rated movies are consistently within a specific runtime range, Canopy can focus on curating films that align with this range. This insight helps tailor content to match audience preferences for both quality and runtime.**
- **Understanding the distribution of genre counts helps Canopy identify trends in movie diversity, if most successful movies are two-genre, Canopy might focus on acquiring or producing such content.**

- **Continuously use data analytics to monitor trends in audience preferences, genre popularity, and viewership behaviour.**

---

4. <u>Ethical Obligations</u>

  - **Ensure that content reflects diverse voices, cultures, and perspectives, avoiding stereotypes or biases in curated and original films.**

  - **Use audience data responsibly by adhering to privacy laws like GDPR, ensuring transparency and consent in data collection and usage.**

  - **Avoid exploiting personal data for invasive marketing tactics.**

  - **Offer equitable opportunities for emerging filmmakers and ensure ethical practices in collaborations and partnerships.**

  - **Promote content without misleading viewers about its quality or appeal.**

Implementing data-driven strategies in streaming services necessitates careful ethical considerations, particularly concerning user privacy and data usage. Netflix, for example, collects extensive user data to inform content recommendations and production decisions, raising concerns about how this data is utilized and protected. (Blinks and Buttons

5. References

Grand View Research. (2023). *Video streaming market size, share & trends analysis report by solution (Internet Protocol TV, Over-the-top), by platform (gaming consoles, laptops & desktops), by region, and segment forecasts, 2024 - 2030*. Retrieved from https://www.grandviewresearch.com/industry-analysis/video-streaming-market

User privacy and data usage Blinks and Buttons