



HW7

PLOT กราฟแสดงการกระจายของข้อมูลใน INDIA IMDB

[CLICK HERE](#)

นำเข้าข้อมูล



▼ นำเข้าข้อมูล

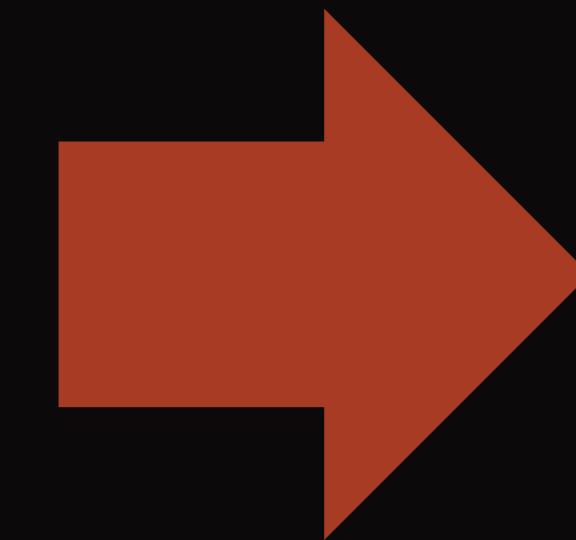
```
[1] import pandas as pd  
#  
[2] import os  
#  
[3] from google.colab import drive  
drive.mount('/content/drive')  
→ Mounted at /content/drive  
[4] path_to_movie = '/content/drive/MyDrive/data_viz_2024_DATA/IMDb_Movies_India.csv'  
#  
▶ movie_df = pd.read_csv(path_to_movie, encoding='latin-1')  
movie_df
```

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3	grid
0			Nan	Nan	Drama	Nan	J.S. Randhawa	Manmauji	Birbal	Rajendra Bhatia	
1	#Gadhvi (He thought he was Gandhi)	(2019)	109 min	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid	
2	#Homecoming	(2021)	90 min	Drama, Musical	Nan	Nan	Soumyajit Majumdar	Sayani Gupta	Plabita Borthakur	Roy Angana	
3	#Yaaram	(2019)	110 min	Comedy, Romance	4.4	35	Ovais Khan	Prateik	Ishita Raj	Siddhant Kapoor	
4	...And Once Again	(2010)	105 min	Drama	Nan	Nan	Amol Palekar	Rajat Kapoor	Rituparna Sengupta	Antara Mali	
...	
15504	Zulm Ko Jala Doonga	(1988)	Nan	Action	4.6	11	Mahendra Shah	Naseeruddin Shah	Sumeet Saigal	Suparna Anand	
15505	Zulmi	(1999)	129 min	Action, Drama	4.5	655	Kuku Kohli	Akshay Kumar	Twinkle Khanna	Aruna Irani	
15506	Zulmi Raj	(2005)	Nan	Action	Nan	Nan	Kiran Thej	Sangeeta Tiwari	NaN	NaN	
15507	Zulmi Shikari	(1988)	Nan	Action	Nan	Nan	NaN	NaN	NaN	NaN	
15508	Zulm-O-Sitam	(1998)	130 min	Action, Drama	6.2	20	K.C. Bokadia	Dharmendra	Jaya Prada	Arjun Sarja	

15509 rows × 10 columns

จัดการข้อมูล





[] movie_df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15509 entries, 0 to 15508
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   Name      15509 non-null   object  
 1   Year       14981 non-null   object  
 2   Duration   7240 non-null   object  
 3   Genre      13632 non-null   object  
 4   Rating     7919 non-null   float64 
 5   Votes      7920 non-null   object  
 6   Director   14984 non-null   object  
 7   Actor 1    13892 non-null   object  
 8   Actor 2    13125 non-null   object  
 9   Actor 3    12365 non-null   object  
dtypes: float64(1), object(9)
memory usage: 1.2+ MB
```

ตรวจสอบค่า null
movie_df.isnull().sum()

	Count
Name	0
Year	528
Duration	8269
Genre	1877
Rating	7590
Votes	7589
Director	525
Actor 1	1617
Actor 2	2384
Actor 3	3144

dtype: int64

ตรวจสอบข้อมูลเบื้องต้น

- DataFrame มี 15,509 แถว และ 10 คอลัมน์
- คอลัมน์สำคัญ เช่น Name, Year, Duration, Genre, Rating
- ตรวจสอบประเภทข้อมูลและจำนวนข้อมูลที่ไม่เป็นค่า null ในแต่ละคอลัมน์

ตรวจสอบค่าที่หายไป

- มีข้อมูลที่หายไปในหลายคอลัมน์ เช่น Year, Duration, Genre, Rating
- ค่าที่หายไปจำเป็นต้องจัดการเพื่อไม่ให้กระทบต่อการวิเคราะห์ในขั้นต่อไป

```
[ ] movie_df = movie_df.dropna(subset=['Rating', 'Votes','Duration'])

▶ movie_df['Year'] = movie_df['Year'].str.replace('(', '').str.replace(')', '').astype(int)
movie_df['Duration'] = movie_df['Duration'].str.replace(' min', '').astype(int)
movie_df['Votes'] = movie_df['Votes'].str.replace(',', '')

☒ <ipython-input-14-3ea375ecc81f>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
movie_df['Year'] = movie_df['Year'].str.replace('(', '').str.replace(')', '').astype(int)
<ipython-input-14-3ea375ecc81f>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
movie_df['Duration'] = movie_df['Duration'].str.replace(' min', '').astype(int)
<ipython-input-14-3ea375ecc81f>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
movie_df['Votes'] = movie_df['Votes'].str.replace(',', '')

[ ] # ตรวจสอบและแปลงคอลัมน์ Votes เป็น float
movie_df['Votes'] = pd.to_numeric(movie_df['Votes'], errors='coerce')

☒ <ipython-input-15-cdd5c73ce1e2>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
movie_df['Votes'] = pd.to_numeric(movie_df['Votes'], errors='coerce')
```

จัดการค่าที่หายไปและการแปลงข้อมูล

- ใช้ dropna() เพื่อลบแถวที่มีค่า Rating, Votes, Duration ที่หายไป
- แปลงข้อมูลในคอลัมน์ Year, Duration, Votes เป็นตัวเลข โดยลบอักขระที่ไม่เกี่ยวข้อง
- ใช้ pd.to_numeric() เพื่อแปลงคอลัมน์ Votes เป็นตัวเลข

[] movie_df

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
1	#Gadhvi (He thought he was Gandhi)	2019	109	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arwind Jangid
3	#Yaaram	2019	110	Comedy, Romance	4.4	35	Ovais Khan	Prateik	Ishita Raj	Siddhant Kapoor
5	...Aur Pyaar Ho Gaya	1997	147	Comedy, Drama, Musical	4.7	827	Rahul Rawail	Bobby Deol	Aishwarya Rai Bachchan	Shammi Kapoor
6	...Yahaan	2005	142	Drama, Romance, War	7.4	1086	Shoojit Sircar	Jimmy Sheirgill	Minissha Lamba	Yashpal Sharma
8	? A Question Mark	2012	82	Horror, Mystery, Thriller	5.6	326	Allyson Patel	Yash Dave	Muntazir Ahmad	Kiran Bhatia
...
15493	Zubaan	2015	115	Drama	6.1	408	Mozez Singh	Vicky Kaushal	Sarah Jane Dias	Raaghav Chanana
15494	Zubeidaa	2001	153	Biography, Drama, History	6.2	1496	Shyam Benegal	Karisma Kapoor	Rekha	Manoj Bajpayee
15503	Zulm Ki Zanjeer	1989	125	Action, Crime, Drama	5.8	44	S.P. Muthuraman	Chiranjeevi	Jayamalini	Rajinikanth
15505	Zulmi	1999	129	Action, Drama	4.5	655	Kuku Kohli	Akshay Kumar	Twinkle Khanna	Aruna Irani
15508	Zulm-O-Sitam	1998	130	Action, Drama	6.2	20	K.C. Bokadia	Dharmendra	Jaya Prada	Arjun Sarja

5851 rows × 10 columns

ข้อมูลหลังการจัดการ

แสดงตัวอย่างข้อมูลที่จัดการแล้วใน DataFrame ซึ่งประกอบด้วยคอลัมน์ Name, Year, Duration, Genre, Rating, Votes, Director, และนักแสดง

```
movie_df.shape  
[5851, 10]  
# หั่งdropกอแล้วข้อมูลเหลือกี่%จากเดิม  
print(f"ข้อมูลเหลือ {(len(movie_df) / len(pd.read_csv('/content/drive/MyDrive/data_viz_2024_DATA/IMDb_Movies_India.csv', encoding='latin-1')) * 100:.2f}% จากเดิม")  
ข้อมูลเหลือ 37.73% จากเดิม  
movie_df.duplicated().sum()  
0
```

ข้อมูลหลังการลบค่าที่หายไป

- ข้อมูลมีขนาด 5,851 แถว และ 10 คอลัมน์หลังจากการลบค่าที่หายไป (shape)
- ข้อมูลเหลือเพียง 37.73% จากข้อมูลเดิม
- ไม่มีข้อมูลซ้ำ (duplicated)

```
[ ] movie_df.isnull().sum()
```

	0
Name	0
Year	0
Duration	0
Genre	31
Rating	0
Votes	0
Director	1
Actor 1	75
Actor 2	117
Actor 3	163

dtype: int64

ค่าที่หายไปหลังการจัดการ

ตรวจสอบค่าที่หายไปในคอลัมน์ Genre, Director, และ Actor โดยมี Genre หายไป 31 ค่า, Director หายไป 1 ค่า, และ Actor หายไปในบางคอลัมน์

```
# แยกประเภทหนังในคอลัมน์ Genre และเลือกประเภทแรกที่พบ
movie_df['Primary Genre'] = movie_df['Genre'].str.split(',').str[0]

# ดูประเภทหนังที่ไม่ซ้ำกัน
unique_genres = movie_df['Primary Genre'].unique()

# แสดงจำนวนและประเภทหนังที่ไม่ซ้ำกัน
print(f"จำนวนประเภทหนังทั้งหมด: {len(unique_genres)}")
print("ประเภทหนังที่ไม่ซ้ำกัน:")
print(unique_genres)
```

```
จำนวนประเภทหนังทั้งหมด: 21
ประเภทหนังที่ไม่ซ้ำกัน:
['Drama' 'Comedy' 'Horror' 'Action' 'Crime' 'Thriller' 'Adventure' 'Sport'
 'Biography' 'Documentary' 'Mystery' 'Musical' 'Romance' 'Fantasy'
 'Sci-Fi' 'Family' 'History' 'Animation' nan 'War' 'Music']
<ipython-input-22-c1d4aec5ee13>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
movie_df['Primary Genre'] = movie_df['Genre'].str.split(',').str[0]
```

ตรวจสอบและนับจำนวนประเภทหนังที่มีในข้อมูล

- ดำเนินการแยกประเภทหนังหลัก (Primary Genre) จากคอลัมน์ 'Genre'
- พบประเภทหนังที่ไม่ซ้ำกันทั้งหมด 21 ประเภท
- ประเภทหนังที่พบ ได้แก่: 'Drama', 'Comedy', 'Horror', 'Action', 'Crime', 'Thriller',
'Adventure', 'Sport', 'Biography', 'Documentary', 'Mystery', 'Musical',
'Romance', 'Fantasy', 'Sci-Fi', 'Family', 'History', 'Animation', 'nan', 'War', 'Music'



```
# สร้าง index สำหรับแทนที่ประเภทหนังด้วยตัวเลข
genre_to_num = {genre: idx for idx, genre in enumerate(movie_df['Primary Genre'].unique())}
movie_df['Genre_Num'] = movie_df['Primary Genre'].map(genre_to_num)

# สรุปข้อมูล
summary = movie_df.groupby('Primary Genre').agg({
    'Rating': ['mean', 'median', 'min', 'max', 'count']
}).sort_values([('Rating', 'mean')], ascending=False)

summary.columns = ['Average Rating', 'Median Rating', 'Min Rating', 'Max Rating', 'Number of Movies']
summary = summary.reset_index()
summary
```

	Primary Genre	Average Rating	Median Rating	Min Rating	Max Rating	Number of Movies
0	Documentary	7.586154	7.70	4.1	9.3	130
1	Music	7.466667	6.80	5.9	9.7	3
2	History	7.333333	7.10	6.0	9.4	9
3	Biography	6.697619	6.95	2.4	8.9	84
4	Family	6.391228	6.40	2.6	9.3	57
5	Sci-Fi	6.320000	6.30	4.3	9.3	5
6	Drama	6.255253	6.50	1.6	10.0	1875
7	Fantasy	6.251613	6.60	2.4	8.0	31
8	Adventure	6.181905	6.50	2.4	8.4	105
9	Crime	6.132721	6.40	2.1	8.9	272
10	Musical	6.095556	6.55	1.8	8.2	90
11	Animation	6.014286	6.25	2.3	8.6	56
12	Mystery	5.883607	6.20	2.7	8.8	61
13	Comedy	5.844121	6.10	1.6	9.3	995
14	Sport	5.800000	5.80	5.5	6.1	2
15	Romance	5.612500	5.70	1.8	8.6	160
16	Action	5.513907	5.60	1.1	9.2	1661
17	Thriller	5.330435	5.50	2.5	8.7	92
18	Horror	4.696899	4.60	1.9	8.0	129
19	War	4.333333	3.50	2.8	6.7	3

```
▶ # หา Genre ที่มีค่าเฉลี่ย Rating สูงสุดและต่ำสุด  
top_genre = summary.iloc[0]  
bottom_genre = summary.iloc[-1]  
  
print(f"\nGenre ที่มีค่าเฉลี่ย Rating สูงสุด: {top_genre['Primary Genre']} (Average Rating: {top_genre['Average Rating']:.2f})")  
print(f"Genre ที่มีค่าเฉลี่ย Rating ต่ำสุด: {bottom_genre['Primary Genre']} (Average Rating: {bottom_genre['Average Rating']:.2f})")  
  
# หา Genre ที่มีจำนวนหนังมากที่สุดและน้อยที่สุด  
most_movies = summary.loc[summary['Number of Movies'].idxmax()]  
least_movies = summary.loc[summary['Number of Movies'].idxmin()]  
  
print(f"\nGenre ที่มีจำนวนหนังมากที่สุด: {most_movies['Primary Genre']} ({most_movies['Number of Movies']} movies)")  
print(f"Genre ที่มีจำนวนหนังน้อยที่สุด: {least_movies['Primary Genre']} ({least_movies['Number of Movies']} movies)")  
  
# คำนวณค่าเฉลี่ย Rating ทั้งหมด  
overall_avg_rating = movie_df['Rating'].mean()  
print(f"\nค่าเฉลี่ย Rating ของหนังทั้งหมด: {overall_avg_rating:.2f}")
```



Genre ที่มีค่าเฉลี่ย Rating สูงสุด: Documentary (Average Rating: 7.59)

Genre ที่มีค่าเฉลี่ย Rating ต่ำสุด: War (Average Rating: 4.33)

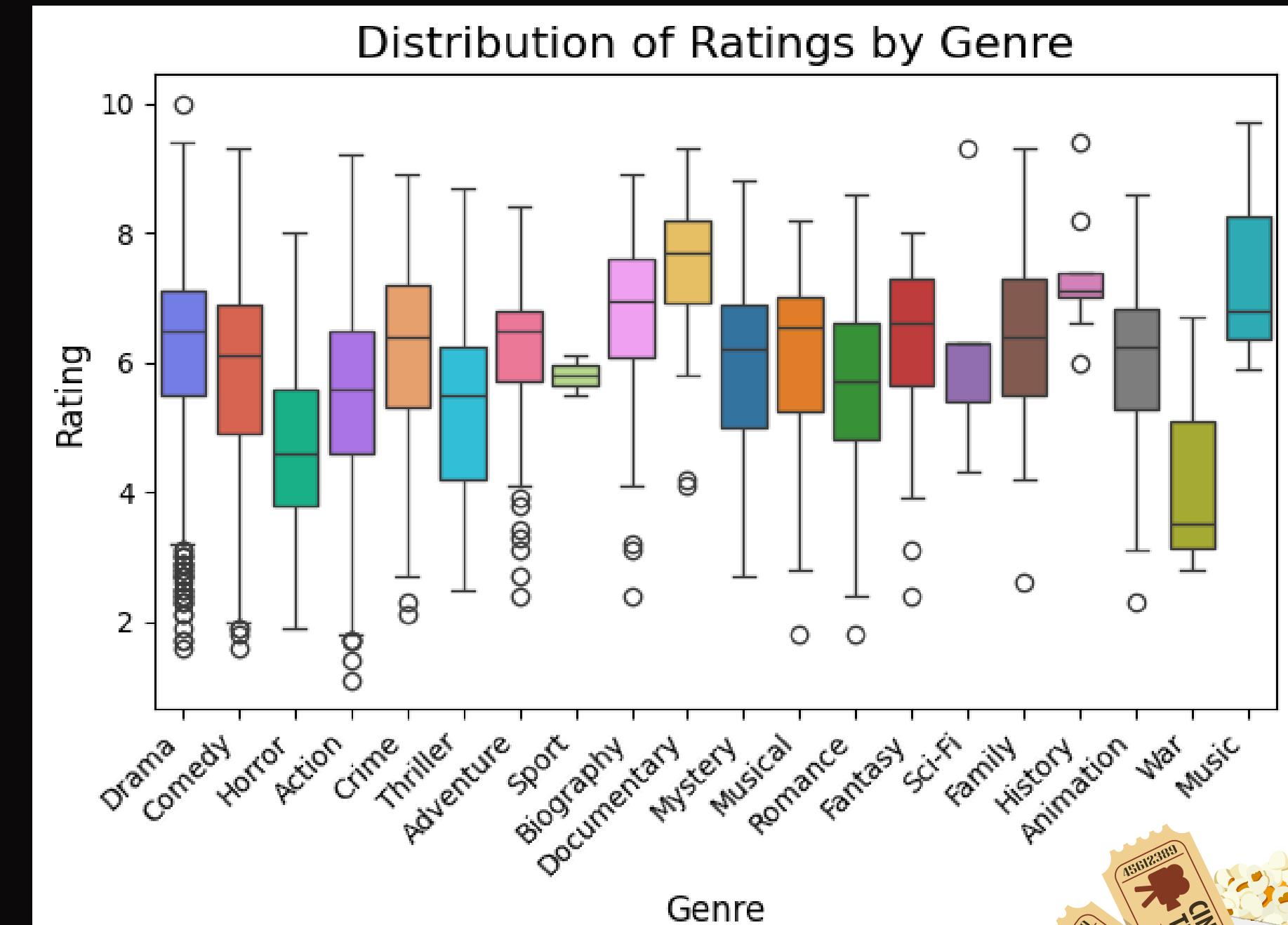
Genre ที่มีจำนวนหนังมากที่สุด: Drama (1875 movies)

Genre ที่มีจำนวนหนังน้อยที่สุด: Sport (2 movies)

ค่าเฉลี่ย Rating ของหนังทั้งหมด: 5.93



```
1 # สร้างพาเลทสีที่มีจำนวนสีเท่ากับจำนวน Genre
2 colors = ['#636EFA', '#EF553B', '#00CC96', '#AB63FA', '#FFA15A',
3           '#19D3F3', '#FF6692', '#B6E880', '#FF97FF', '#FECB52',
4           '#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
5           '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf',
6           '#FF5733']
7 # ใช้พาเลทสีที่กำหนด
8 sns.boxplot(
9     x='Primary Genre',
10    y='Rating',
11    data=movie_df,
12    palette=colors # ใช้พาเลทสีที่กำหนดเอง
13 )
14
15 # ตั้งชื่อและป้ายกำกับกราฟ
16 plt.title('Distribution of Ratings by Genre', fontsize=16)
17 plt.xlabel('Genre', fontsize=12)
18 plt.ylabel('Rating', fontsize=12)
19 plt.xticks(rotation=45, ha='right')
20 plt.tight_layout()
21
22 # แสดงกราฟ
23 plt.show()
```



```
[41] 1 # สร้าง Scatter Plot ด้วย Plotly
2 colors = ['#636EFA', '#EF553B', '#00CC96', '#AB63FA', '#FFA15A',
3           '#19D3F3', '#FF6692', '#B6E880', '#FF97FF', '#FECB52',
4           '#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
5           '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf',
6           '#FF5733']
7 fig = px.scatter(
8     movie_df,
9     x='Rating',
10    y='Votes',
11    color='Primary Genre', # ใช้สีเพื่อแยกประเภท
12    hover_name='Name', # แสดงชื่อภาพยนตร์เมื่อ hover
13    hover_data={'Year': True, 'Genre': True, 'Duration': True, 'Rating': True, 'Votes': True, 'Primary Genre': True},
14    title='Scatter Plot of Rating vs. Votes',
15    labels={'Rating': 'Rating', 'Votes': 'Votes'},
16    color_discrete_sequence=colors # กำหนดสี
17 )
18
19 # ปรับแต่งกราฟ
20 fig.update_layout(
21     xaxis_title='Rating',
22     yaxis_title='Votes',
23     template='plotly_white'
24 )
25
26 # แสดงกราฟ
27 fig.show()
```

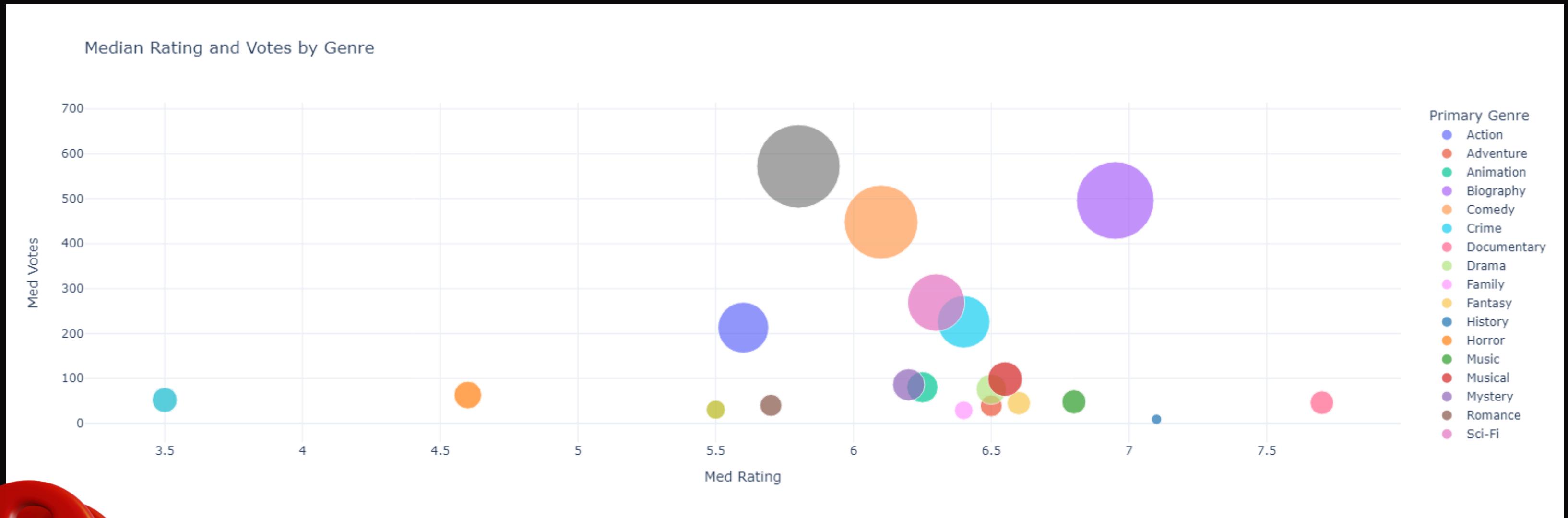


Plot 1



```
1 # ข้อมูลค่าเฉลี่ย
2 avg_data = movie_df.groupby('Primary Genre')[['Rating', 'Votes']].median().reset_index()
3
4 # สร้างกราฟ scatter plot ด้วย Plotly
5 colors = ['#636EFA', '#EF553B', '#00CC96', '#AB63FA', '#FFA15A',
6           '#19D3F3', '#FF6692', '#B6E880', '#FF97FF', '#FECB52',
7           '#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
8           '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf',
9           '#FF5733']
10 fig = px.scatter(
11     avg_data,
12     x='Rating',
13     y='Votes',
14     color='Primary Genre', # สีแยกตาม Primary Genre
15     size='Votes', # ขนาดของฟองอากาศตามจำนวน Votes
16     size_max=60, # ขนาดสูงสุดของฟองอากาศ (ปรับให้เล็กลง)
17     hover_name='Primary Genre', # ข้อมูลที่แสดงเมื่อ hover
18     hover_data={'Rating': True, 'Votes': True}, # ข้อมูลเพิ่มเติมที่แสดงเมื่อ hover
19     title='Median Rating and Votes by Genre',
20     labels={'Rating': 'Med Rating', 'Votes': 'Med Votes'},
21     color_discrete_sequence=colors # กำหนดสี
22 )
23
24 # ปรับแต่ง Layout
25 fig.update_layout(
26     xaxis_title='Med Rating',
27     yaxis_title='Med Votes',
28     template='plotly_white'
29 )
30
31 # แสดงกราฟ
32 fig.show()
```

Median Rating and Votes by Genre



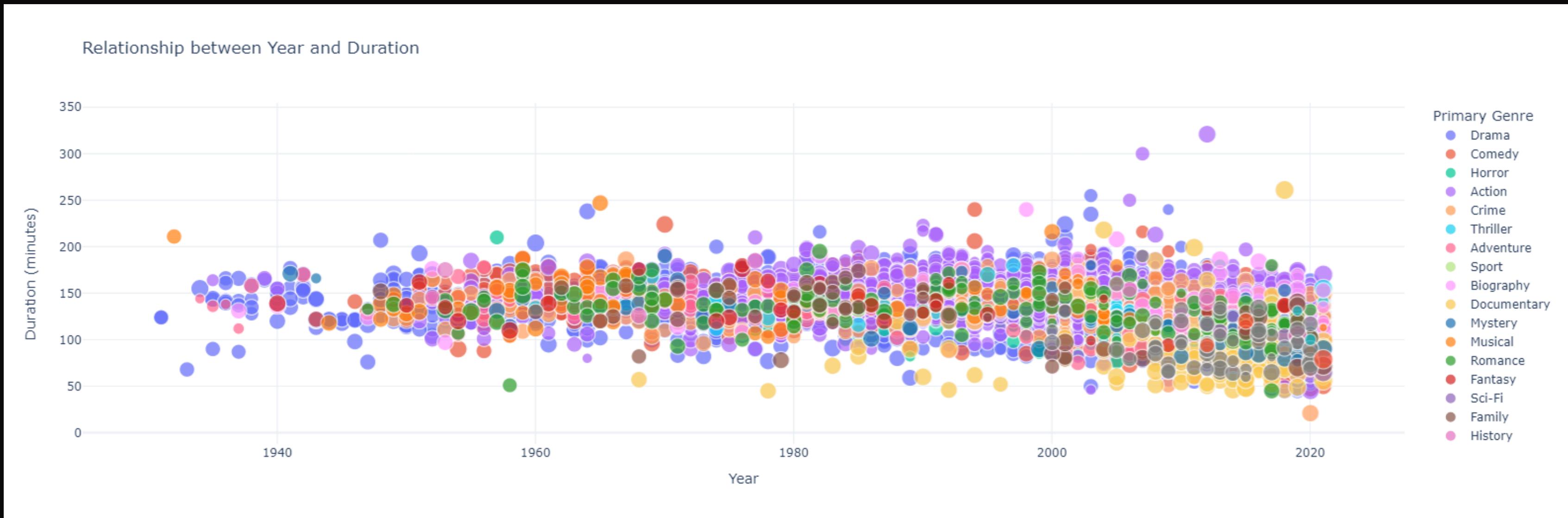
Plot 2

```
1 # สร้างกราฟ
2 colors = ['#636EFA', '#EF553B', '#00CC96', '#AB63FA', '#FFA15A',
3           '#19D3F3', '#FF6692', '#B6E880', '#FF97FF', '#FEBC52',
4           '#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
5           '#8c564b', '#e377c2', '#7f7f7f', '#bcbd22', '#17becf',
6           '#FF5733']
7 fig = px.scatter(
8     movie_df,
9     x='Year',
10    y='Duration',
11    color='Primary Genre', # ใช้สีเพื่อแยกประเภท
12    size='Rating', # ขนาดของฟองอากาศแสดงถึงคะแนน
13    hover_name='Name', # แสดงชื่อภาพยนตร์เมื่อ hover
14    hover_data={'Year': True, 'Duration': True, 'Rating': True, 'Primary Genre': True},
15    title='Relationship between Year and Duration',
16    labels={'Duration': 'Duration (minutes)', 'Rating': 'Rating'},
17    color_discrete_sequence=colors # กำหนดสี
18 )
19
20 # ปรับแต่งกราฟ
21 fig.update_layout(
22     xaxis_title='Year',
23     yaxis_title='Duration (minutes)',
24     coloraxis_colorbar=dict(title='Primary Genre'),
25     template='plotly_white'
26 )
27
28 # แสดงกราฟ
29 fig.show()
```

```
1 # คำนวณค่าสหสัมพันธ์ระหว่าง Year และ Duration
2 correlation = movie_df['Year'].corr(movie_df['Duration'])
3
4 print(f'Correlation between Year and Duration: {correlation}')
5
```

Correlation between Year and Duration: -0.3374252032853612

RELATIONSHIP BETWEEN YEAR AND DURATION





สมาชิก

นายนายภูริบง สายเชื้อ¹
นายปานัชญ์ โจนพัฒนาเดชา²
นางสาวพรพรรณ ราชคุมน์³
นางสาวสุวรรณเกตุ สุมาลี⁴

รหัสนักศึกษา 653020216-6
รหัสนักศึกษา 653020210-8
รหัสนักศึกษา 653020213-2
รหัสนักศึกษา 653020578-2