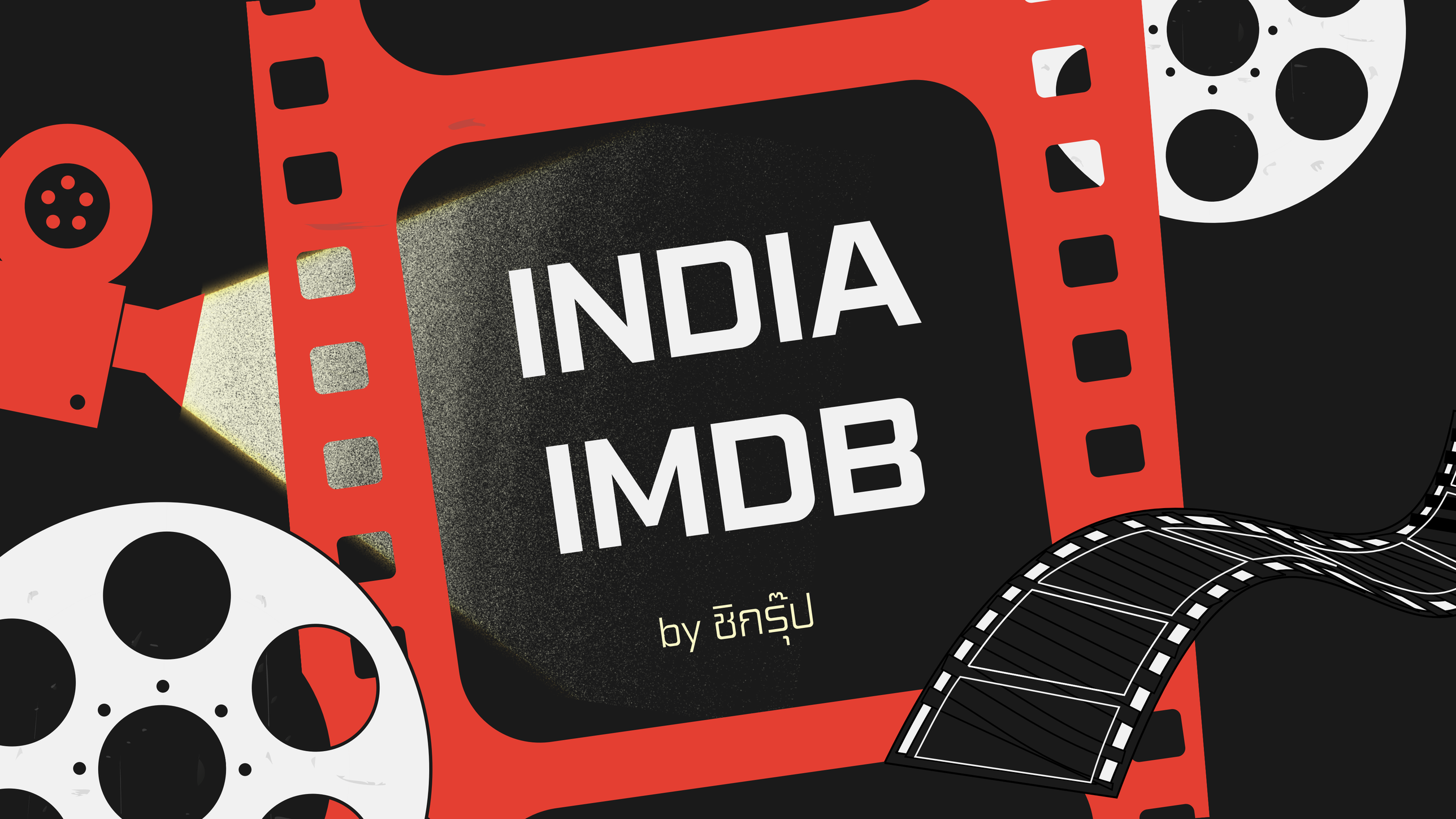
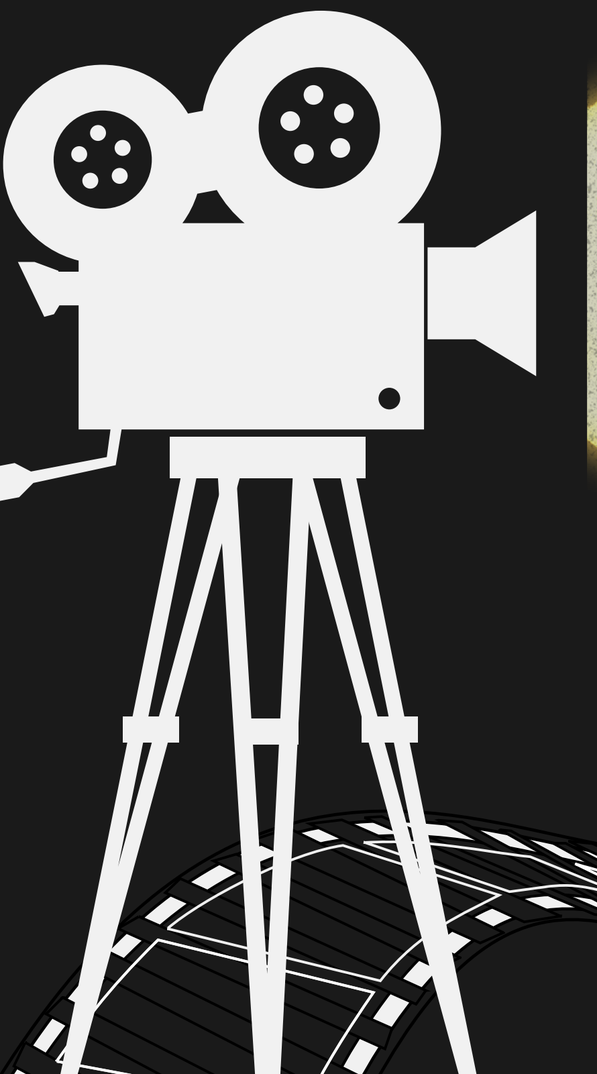

INDIA
IMDB
by ชิกรุ๊ป

สมาชิก

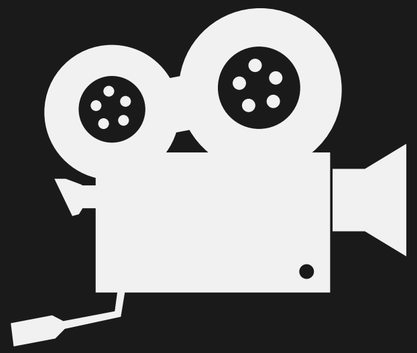


5 ณัชชา ประทุมชาติ

12 พรรณรมณ ราชคมน์

15 รัชชานนท์ พันกาฬสินธุ์

27 อริสรา ยังอยู่



Hw. 9

สร้าง RADAR CHART

ของ TOP DIRECTOR 3 อันดับแรก จากข้อมูลหนัง INDIA

การจัดการข้อมูล

```
import pandas as pd
```

```
import os
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
path_to_movie = '/content/drive/MyDrive/data_viz_2024/IMDb Movies India.csv'
```

```
data_india = pd.read_csv(path_to_movie, encoding='latin-1')
data_india
```

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
0		NaN	NaN	Drama	NaN	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

การจัดการข้อมูล

```
#ตรวจสอบค่า null
data_india.isnull().sum()
```

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

```
data_india = data_india.dropna(subset=['Rating', 'Votes','Duration','Genre', 'Director'])
```

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

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

การจัดการข้อมูล

```
data_india
```

	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	Arvind 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

```
#หลังdropnaแล้วข้อมูลเหลือกี่%จากเดิม

print(f"ข้อมูลเหลือ {(len(data_india) / len(pd.read_csv('/content/drive/MyDrive/data_viz_2024/IMDb Movies India.csv', encoding='latin-1'))) * 100:.2f}% จากเดิม")
```

```
ข้อมูลเหลือ 37.52% จากเดิม
```

```
data_india.isnull().sum()
```

	0
Name	0
Year	0
Duration	0
Genre	0
Rating	0
Votes	0
Director	0
Actor 1	73
Actor 2	114
Actor 3	160

การจัดการข้อมูล

```
# prompt: split value in column genre by ',' and make more column for each of those

import pandas as pd
# Split the 'Genre' column by ',' and create new columns
genre_split = data_india['Genre'].str.split(',', expand=True)

# Rename the new columns
genre_split.columns = ['Genre1', 'Genre2', 'Genre3']

# Concatenate the new columns with the original DataFrame
data_india = pd.concat([data_india, genre_split], axis=1)

# Display the updated DataFrame
data_india.head()
```

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3	Genre1	Genre2	Genre3
1	#Gadhvi (He thought he was Gandhi)	2019	109	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid	Drama	None	None
3	#Yaaram	2019	110	Comedy, Romance	4.4	35	Ovais Khan	Prateik	Ishita Raj	Siddhant Kapoor	Comedy	Romance	None
5	...Aur Pyaar Ho Gaya	1997	147	Comedy, Drama, Musical	4.7	827	Rahul Rawail	Bobby Deol	Aishwarya Rai Bachchan	Shammi Kapoor	Comedy	Drama	Musical

```
# prompt: delete space in the value in Genre1 Genre2 Genre3

for col in ['Genre1', 'Genre2', 'Genre3']:
  data_india[col] = data_india[col].str.strip() if data_india[col].dtype == 'object' else data_india[col]
```

```
data_india
```

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3	Genre1	Genre2	Genre3
1	#Gadhvi (He thought he was Gandhi)	2019	109	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid	Drama	None	None
3	#Yaaram	2019	110	Comedy, Romance	4.4	35	Ovais Khan	Prateik	Ishita Raj	Siddhant Kapoor	Comedy	Romance	None
5	...Aur Pyaar Ho Gaya	1997	147	Comedy, Drama, Musical	4.7	827	Rahul Rawail	Bobby Deol	Aishwarya Rai Bachchan	Shammi Kapoor	Comedy	Drama	Musical
6	...Yahaan	2005	142	Drama, Romance, War	7.4	1086	Shoojit Sircar	Jimmy Sheirgill	Minissha Lamba	Yashpal Sharma	Drama	Romance	War
8	?: A Question Mark	2012	82	Horror, Mystery, Thriller	5.6	326	Allyson Patel	Yash Dave	Muntazir Ahmad	Kiran Bhatia	Horror	Mystery	Thriller

การจัดการข้อมูล

เกณฑ์การเลือกผู้กำกับ

```python
# prompt: เลือกผู้กำกับที่มี Rating และ Votes มากกว่า 75%
# top_directors = movie_df[(movie_df['Rating'] >= rating_q3) & (movie_df['Votes'] >= tes_q3)]['Director'].value_counts()
#และ คิดเกณฑ์คะแนนถ่วงน้ำหนัก Votes และ rating ให้ rating 60% และ Votes 40% ถ่วงให้ไม่เกิน 100%
# Calculate the 75th percentile for Rating and Votes
rating_q3 = data_india['Rating'].quantile(0.75)
tes_q3 = data_india['Votes'].quantile(0.75)

# Filter the DataFrame to include only directors with Rating and Votes above the 75th percentile
top_directors = data_india[(data_india['Rating'] >= rating_q3) & (data_india['Votes'] >= tes_q3)]['Director'].value_counts()
top_directors
```

```python
top_directors.head(10)
```

Director	count
Yash Chopra	11
Hrishikesh Mukherjee	10
Anurag Kashyap	9
Ram Gopal Varma	8
Nagesh Kukunoor	8
Vishal Bhardwaj	7
Sanjay Leela Bhansali	7
Raj Kapoor	7
Priyadarshan	7
Gulzar	6

การจัดการข้อมูล

```
# prompt: อยากสร้างตัวแปร experience โดยคิดจาก ปี(Year)ใหม่สุดของหนัง(Name)ลบกับปี(Year)เก่าสุดของหนัง(Name) ของDirectorแต่ละคน

# Group by director and get the minimum and maximum year for each director
director_experience = data_india.groupby('Director').agg({'Year': ['min', 'max']})

# Calculate the experience by subtracting the minimum year from the maximum year
director_experience['Experience'] = director_experience['Year']['max'] - director_experience['Year']['min']

# Display the experience for each director
print(director_experience['Experience'])

# You can merge this experience data back into your top DataFrame if needed
# top = top.merge(director_experience['Experience'], on='Director', how='left')
```

```
Director
A. Bhimsingh       16
A. Jagannathan     10
A. Majid            0
A. Muthu            0
A. Salaam          13
                   ..
Zia Sarhadi         5
Ziaullah Khan       0
Zoya Akhtar        11
Zubair Khan         0
Zunaid Memon        0
Name: Experience, Length: 2540, dtype: int64
```

```
director_experience['Experience']
```

Director	Experience
A. Bhimsingh	16
A. Jagannathan	10
A. Majid	0
A. Muthu	0
A. Salaam	13
...	...
Zia Sarhadi	5
Ziaullah Khan	0
Zoya Akhtar	11
Zubair Khan	0
Zunaid Memon	0

2540 rows × 1 columns

การจัดการข้อมูล

```
# prompt: นับจำนวน genre ที่director แต่ละคนทำ เช่นทำ Drama กับ Action นับเป็น 2 genre พร้อมบอกด้วยมีอะไรบ้าง

# Group by director and collect unique genres
director_genres = data_india.groupby('Director').agg({'Genre1': lambda x: set(x.dropna()),
                                                      'Genre2': lambda x: set(x.dropna()),
                                                      'Genre3': lambda x: set(x.dropna())})

# Combine genres from all three columns into a single set
director_genres['All_Genres'] = director_genres.apply(lambda row: row['Genre1'].union(row['Genre2']).union(row['Genre3']), axis=1)

# Calculate the number of unique genres for each director
director_genres['Genre_Count'] = director_genres['All_Genres'].apply(len)

# Display the results
print(director_genres[['Genre_Count', 'All_Genres']].to_string())
```

```
                    Genre_Count                                                                                                All_Genres
Director
Hrishikesh Mukherjee            6                                                                                 {Musical, Romance, Drama, Family, Mystery, Comedy}
Ram Gopal Varma                11  {Musical, Romance, Drama, Biography, Thriller, Horror, Adventure, Mystery, Action, Crime, Comedy}
Yash Chopra                    11        {History, Musical, Romance, Drama, Family, Thriller, Music, Mystery, Action, Crime, Comedy}
```

การจัดการข้อมูล

```
allindiagenre = list(data_india['Genre1'])+list(data_india['Genre2'])+list(data_india['Genre3'])
allindiagenre
```

```
 'Action',
 'Drama',
 'Drama',
 'Thriller',
 'Action',
 'Drama',
 'Action',
 'Comedy',
 'Action',
```

```
unique_genres = list(set([genre for genre in allindiagenre if genre is not None]))
print(unique_genres)
```

```
['Biography', 'Horror', 'Family', 'Sport', 'Fantasy', 'Animation', 'Mystery', 'Crime'
```

```
len(set(allindiagenre))
```

```
23
```

การจัดการข้อมูล

```
# prompt: อยากสร้างคะแนน diversity โดยคิดจาก director_genres[['Genre_Count', 'All_Genres'] (ของdata_india) สเกลคะแนนเต็ม10

# Calculate the maximum possible genre count (diversity)
max_genre_count = len(set(allindiagenre))  # Use allindiagenre to get the maximum possible genre count

# Create a new column for the diversity score
director_genres['Diversity_Score'] = (director_genres['Genre_Count'] / max_genre_count) * 10

# Display the results with the diversity score
print(director_genres[['Genre_Count', 'All_Genres', 'Diversity_Score']].to_string())
```

Director	Genre_Count	All_Genres	Diversity_Score
A. Bhimsingh	8	{Musical, Romance, Drama, Biography, Family, Thriller, Action, Comedy}	3.478261
A. Jagannathan	5	{Drama, Romance, Mystery, Action, Crime}	2.173913
A. Majid	2	{Drama, Action}	0.869565
A. Muthu	2	{Romance, Musical}	0.869565
A. Salaam	4	{Drama, Thriller, Action, Family}	1.739130
A. Shamsheer	4	{Romance, Drama, Action, Adventure}	1.739130
A. Veerappan	3	{Drama, Action, Crime}	1.304348
A.C. Trilogchander	3	{Romance, Drama, Family}	1.304348
A.K. Bir	1	{Drama}	0.434783
A.L. Vijay	2	{Horror, Comedy}	0.869565

```
director_genres['Diversity_Score']
```

Director	Diversity_Score
A. Bhimsingh	3.478261
A. Jagannathan	2.173913
A. Majid	0.869565
A. Muthu	0.869565
A. Salaam	1.739130
...	...
Zia Sarhadi	1.739130
Ziaullah Khan	0.434783
Zoya Akhtar	2.608696
Zubair Khan	0.434783
Zunaid Memon	1.304348

2540 rows × 1 columns

การจัดการข้อมูล

```
# prompt: ทำตารางรวม rating (ค่าเฉลี่ย)
# duration (ค่าเฉลี่ย)
# diversity
# movie count
# experience ของ director แต่ละคน

# Create a new DataFrame to store the aggregated data
director_summary = pd.DataFrame()

# Calculate the average rating for each director
director_summary['Average_Rating'] = data_india.groupby('Director')['Rating'].mean()

# Calculate the average duration for each director
director_summary['Average_Duration'] = data_india.groupby('Director')['Duration'].mean()

# Get the diversity score from the director_genres DataFrame
director_summary['Diversity_Score'] = director_genres['Diversity_Score']

# Count the number of movies for each director
director_summary['Movie_Count'] = data_india.groupby('Director')['Name'].count()

# Get the experience from the director_experience DataFrame
director_summary['Experience'] = director_experience['Experience']


# Display the summary table
print(director_summary)
```

Director	Average_Rating	Average_Duration	Diversity_Score	Movie_Count	Experience
A. Bhimsingh	6.785714	151.214286	3.478261	14	16
A. Jagannathan	5.833333	141.000000	2.173913	3	10
A. Majid	5.700000	162.000000	0.869565	1	0
A. Muthu	3.000000	143.000000	0.869565	1	0
A. Salaam	5.575000	130.250000	1.739130	4	13
...	...	...	...	...	...
Zia Sarhadi	6.366667	146.000000	1.739130	3	5
Ziaullah Khan	5.600000	110.000000	0.434783	1	0
Zoya Akhtar	6.800000	146.571429	2.608696	7	11
Zubair Khan	5.400000	113.000000	0.434783	1	0
Zunaid Memon	6.400000	144.000000	1.304348	1	0

การจัดการข้อมูล

director_summary

Director	Average_Rating	Average_Duration	Diversity_Score	Movie_Count	Experience
A. Bhimsingh	6.785714	151.214286	3.478261	14	16
A. Jagannathan	5.833333	141.000000	2.173913	3	10
A. Majid	5.700000	162.000000	0.869565	1	0
A. Muthu	3.000000	143.000000	0.869565	1	0
A. Salaam	5.575000	130.250000	1.739130	4	13
...	...	...	...	...	...
Zia Sarhadi	6.366667	146.000000	1.739130	3	5
Ziaullah Khan	5.600000	110.000000	0.434783	1	0
Zoya Akhtar	6.800000	146.571429	2.608696	7	11
Zubair Khan	5.400000	113.000000	0.434783	1	0
Zunaid Memon	6.400000	144.000000	1.304348	1	0

การจัดการข้อมูล

```
# prompt: ทำNormalizeของdirector_summary

from sklearn.preprocessing import MinMaxScaler

# Create a scaler object
scaler = MinMaxScaler()

# Select the columns to normalize
columns_to_normalize = ['Average_Rating', 'Average_Duration', 'Diversity_Score', 'Movie_Count', 'Experience']

# Fit and transform the selected columns
director_summary[columns_to_normalize] = scaler.fit_transform(director_summary[columns_to_normalize])

# Display the normalized summary table
print(director_summary)
```

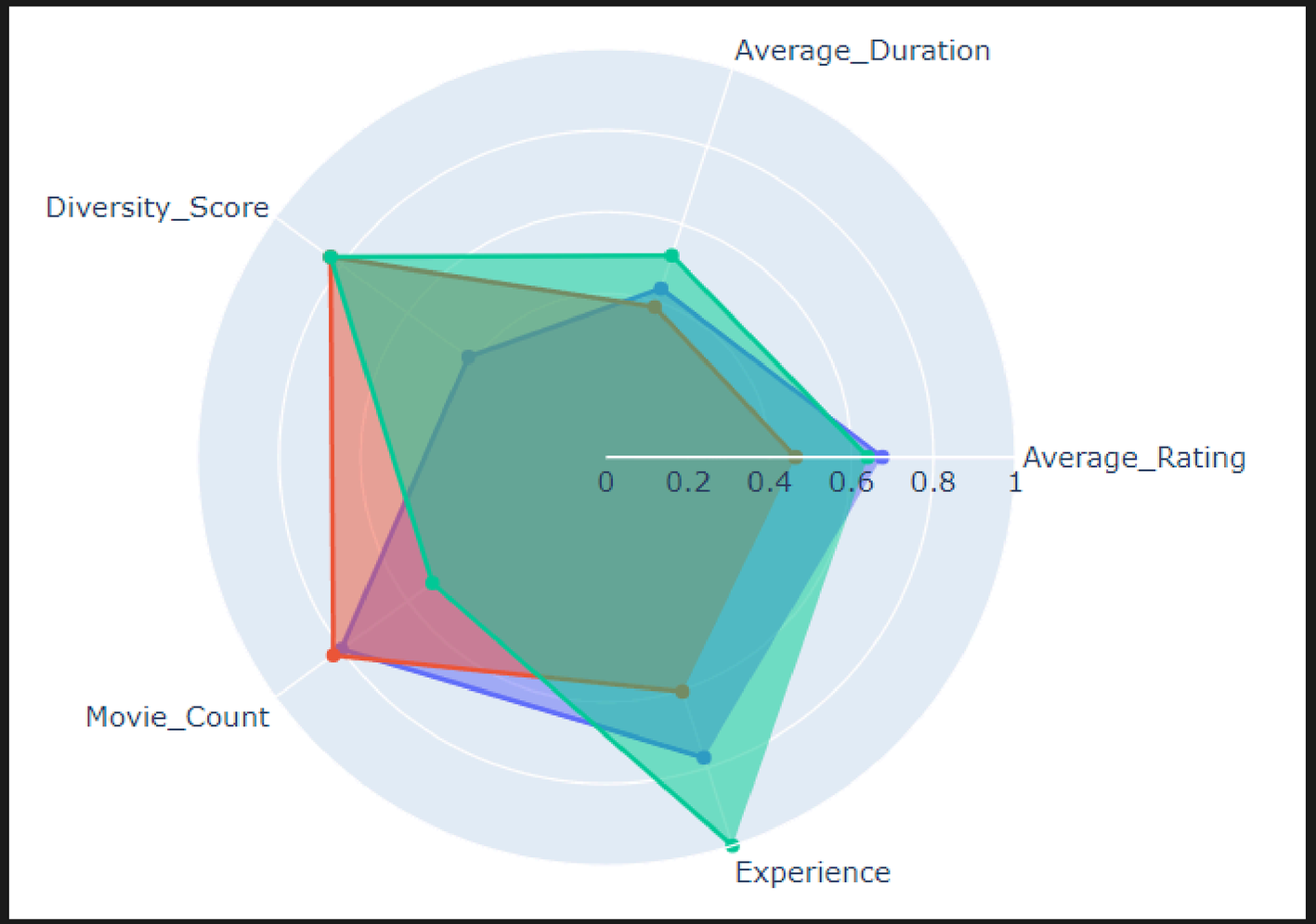
Director	Average_Rating	Average_Duration	Diversity_Score
A. Bhimsingh	0.617347	0.466718	0.583333
A. Jagannathan	0.503968	0.430108	0.333333
A. Majid	0.488095	0.505376	0.083333
A. Muthu	0.166667	0.437276	0.083333
A. Salaam	0.473214	0.391577	0.250000
...	...	...	...
Zia Sarhadi	0.567460	0.448029	0.250000
Ziaullah Khan	0.476190	0.318996	0.000000
Zoya Akhtar	0.619048	0.450077	0.416667
Zubair Khan	0.452381	0.329749	0.000000
Zunaid Memon	0.571429	0.440860	0.166667

Director	Movie_Count	Experience
A. Bhimsingh	0.325	0.301887
A. Jagannathan	0.050	0.188679
A. Majid	0.000	0.000000
A. Muthu	0.000	0.000000
A. Salaam	0.075	0.245283
...	...	...
Zia Sarhadi	0.050	0.094340
Ziaullah Khan	0.000	0.000000
Zoya Akhtar	0.150	0.207547
Zubair Khan	0.000	0.000000
Zunaid Memon	0.000	0.000000

Radar chart

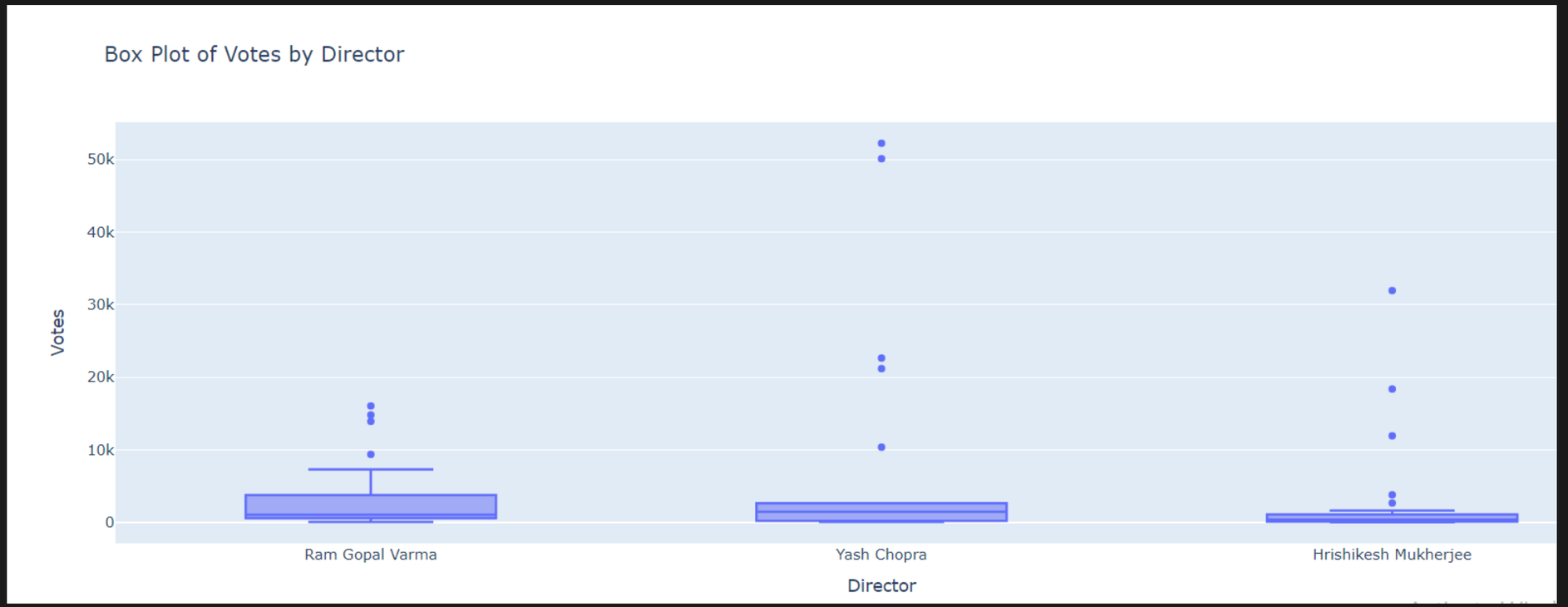
```
# prompt: ทำ radar chart ของ director_summary โดยสนใจแค่ top director

# Select only the top directors from director_summary
top_directors_summary = director_summary[director_summary.index.isin(directors_to_extract)]

categories = ['Average_Rating', 'Average_Duration', 'Diversity_Score', 'Movie_Count', 'Experience']
fig = go.Figure()

for director in top_directors_summary.index:
  fig.add_trace(go.Scatterpolar(
        r=top_directors_summary.loc[director].values,
        theta=categories,
        fill='toself',
        name=director
  ))

fig.update_layout(
  polar=dict(
    radialaxis=dict(
      visible=True,
      range=[0, 1]
    )),
  showlegend=True,
  title="Radar Chart of Top Directors"
)

fig.show()
```

Radar chart


Radar Chart of Top Directors
Average_Duration
Diversity_Score
Movie_Count
Experience
Average_Rating
0
0.2
0.4
0.6
0.8
1
Hrishikesh Mukherjee
Ram Gopal Varma
Yash Chopra

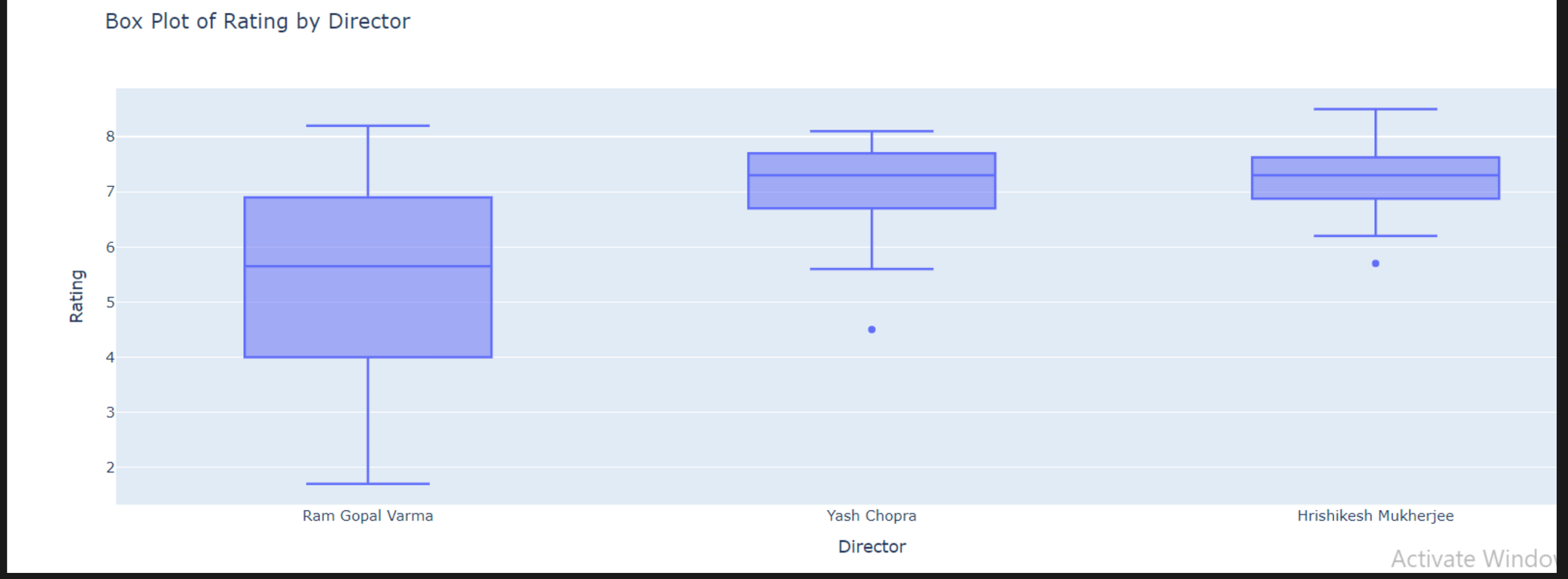
การจัดการข้อมูล

```
import plotly.express as px

fig = px.box(top, x='Director', y='Votes', title='Box Plot of Votes by Director')
fig.show()
```



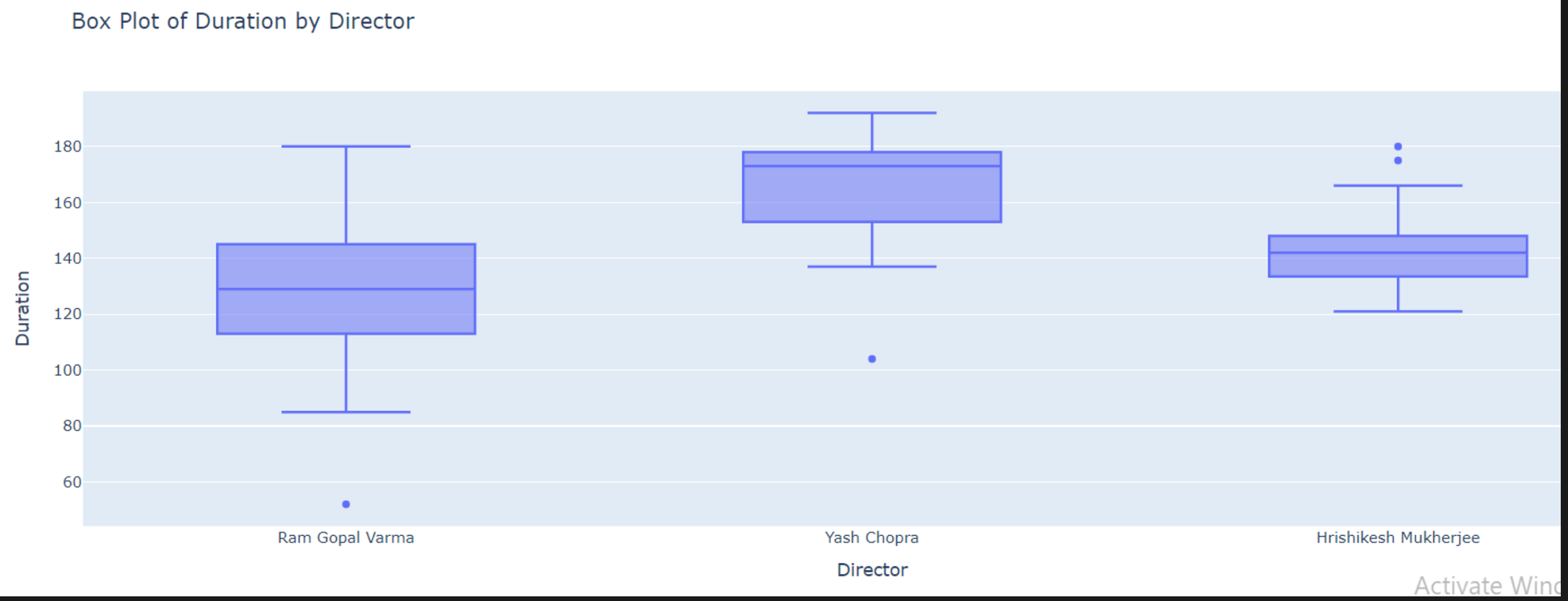
การจัดการข้อมูล

```
# prompt: box plot ของ rating ของ director แต่ละคน

fig = px.box(top, x='Director', y='Rating', title='Box Plot of Rating by Director')
fig.show()
```



การจัดการข้อมูล

```
# prompt: box plot ของ duration ของ director แต่ละคน

fig = px.box(top, x='Director', y='Duration', title='Box Plot of Duration by Director')
fig.show()
```




THANK
YOU