INDIA MOVIES

HW 7

DATA CLEANING

```
movie_df = movie_df.dropna()
movie_df = movie_df.drop_duplicates()
print(movie_df.info())
movie_df
```

การลบแถวที่มีค่า NaN หรือค่าที่ขาดหายไปจาก DataFrame และ ลบแถวที่ซ้ำกันออกจาก DataFrame

DATA CLEANING

```
# Delete min character from 'Duration' column on movie_df
movie_df['Duration'] = movie_df['Duration'].str.replace('min','')
# prompt: delete parentheses in 'Year' column on movie_df
movie_df['Year'] = movie_df['Year'].str.replace('(', '').str.replace(')', '')
print(movie_df.info())
movie_df
```

ลบคำว่า "min" ออกจากข้อมูลในคอลัมน์ Duration โดยใช้ฟังก์ชัน .str.replace() และวงเล็บออกจากคอลัมน์ Year

DATA CLEANING

```python
# Convert 'Votes' column to string type before using .str accessor
movie_df['Votes'] = movie_df['Votes'].astype(str).str.replace(',', '').astype(float)
# prompt: Convert 'Duration' column to float type
movie_df['Duration'] = movie_df['Duration'].astype(float)
# prompt: convert 'Year' column in movie_df to date type
movie_df['Year'] = pd.to_datetime(movie_df['Year'], format='%Y')
print(movie_df.info())
print(movie_df.describe())
movie_df
```

แปลงคอลัมน์ Votes ให้เป็นชนิดข้อมูล string เพื่อให้สามารถใช้ .str accessor ในการจัดการกับข้อมูลจากนั้น จะลบเครื่องหมายจุลภาค (,) ที่อยู่ในค่าของคอลัมน์ Votes ซึ่งอาจเป็นรูปแบบที่แสดงจำนวนโหวต เช่น "1,234" จะถูกแปลงเป็น "1234แปลงกลับเป็นชนิด float

แปลงค่าของคอลัมน์ Duration ให้เป็นชนิดข้อมูล float

แปลงค่าของคอลัมน์ Year ให้เป็นชนิดข้อมูล datetime โดยใช้รูปแบบ %Y

DATA CLEANING

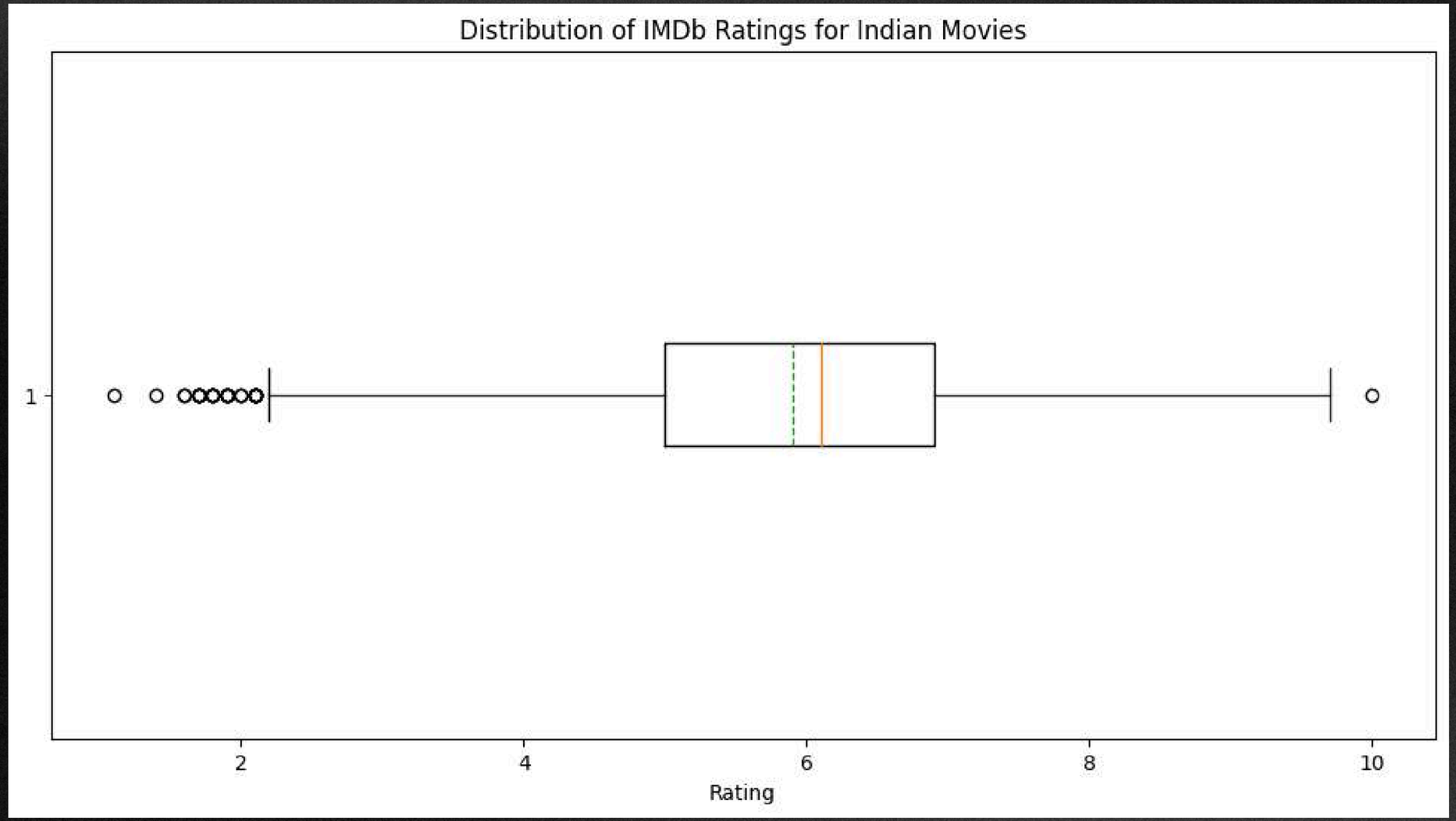
	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
1	#Gadhvi (He thought he was Gandhi)	2019-01-01	109.0	Drama	7.0	8.0	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid
3	#Yaaram	2019-01-01	110.0	Comedy, Romance	4.4	35.0	Ovais Khan	Prateik	Ishita Raj	Siddhant Kapoor
5	...Aur Pyaar Ho Gaya	1997-01-01	147.0	Comedy, Drama, Musical	4.7	827.0	Rahul Rawail	Bobby Deol	Aishwarya Rai Bachchan	Shammi Kapoor
6	...Yahaan	2005-01-01	142.0	Drama, Romance, War	7.4	1086.0	Shoojit Sircar	Jimmy Sheirgill	Minissha Lamba	Yashpal Sharma
8	?: A Question Mark	2012-01-01	82.0	Horror, Mystery, Thriller	5.6	326.0	Allyson Patel	Yash Dave	Muntazir Ahmad	Kiran Bhatia
...	...	...	...	...	...	...	...	...	...	...
15493	Zubaan	2015-01-01	115.0	Drama	6.1	408.0	Mozez Singh	Vicky Kaushal	Sarah Jane Dias	Raaghavv Chanana
15494	Zubeidaa	2001-01-01	153.0	Biography, Drama, History	6.2	1496.0	Shyam Benegal	Karisma Kapoor	Rekha	Manoj Bajpayee
15503	Zulm Ki Zanjeer	1989-01-01	125.0	Action, Crime, Drama	5.8	44.0	S.P. Muthuraman	Chiranjeevi	Jayamalini	Rajinikanth
15505	Zulmi	1999-01-01	129.0	Action, Drama	4.5	655.0	Kuku Kohli	Akshay Kumar	Twinkle Khanna	Aruna Irani
15508	Zulm-O-Sitam	1998-01-01	130.0	Action, Drama	6.2	20.0	K.C. Bokadia	Dharmendra	Jaya Prada	Arjun Sarja

CREATE A BOXPLOT

```
# ขยายขนาดกราฟ
plt.figure(figsize=(12, 6))  # กำหนดขนาดของกราฟ (กว้าง, สูง)

# สร้าง boxplot
O = plt.boxplot(movie_df['Rating'], showmeans=True, meanline=True, vert=False)

# เพิ่มชื่อแกน
plt.xlabel('Rating')
plt.title('Distribution of IMDb Ratings for Indian Movies')

# แสดงกราฟ
plt.show()
```

CREATE A BOXPLOT

DISTRIBUTION OF IMDB RATINGS FOR INDIAN MOVIES



NEW DATAFRAME FOR SCATTER

ดู Top 3 ของ Genre

```
df_genre = pd.DataFrame(movie_df, columns=['Genre'])
df_genre = df_genre.groupby('Genre').size() # Use size() to count occurrences in each group
df_genre = df_genre.reset_index(name='Count')
df_genre = df_genre.sort_values(by='Count', ascending=False)
df_genre
```

	Genre	Count
229	Drama	844
284	Drama, Romance	332
28	Action, Crime, Drama	329
38	Action, Drama	206
151	Comedy, Drama	205
...	...	...
113	Animation, Action, Comedy	1
111	Animation, Action	1
252	Drama, Fantasy, History	1
108	Adventure, Romance	1
188	Comedy, Sci-Fi	1

376 rows × 2 columns

NEW DATAFRAME FOR SCATTER

- สร้าง dataframe ใหม่ที่เก็บ top 3 genres

```
[15] # Filter the DataFrame to include rows where the 'Genre' column is one of the desired values
     movie_genre_top3_df = movie_df[movie_df['Genre'].isin(['Drama', 'Drama, Romance', 'Action, Crime, Drama'])]
     movie_genre_top3_df = movie_genre_top3_df[['Genre', 'Duration','Rating', 'Votes']] # Pass column names as strings in a list. Changed 'Rating' to 'Ratings'.
     print(movie_genre_top3_df.info())
     movie_genre_top3_df
```

movie_genre_top3_df

	Genre	Duration	Rating	Votes
1	Drama	109.0	7.0	8.0
10	Drama	96.0	6.2	17.0
30	Drama	116.0	7.1	1002.0
32	Drama	168.0	5.6	15.0
36	Drama	94.0	4.5	16.0
...	...	...	...	...
15466	Drama	134.0	6.0	5.0
15482	Drama	140.0	5.7	7.0
15488	Drama	100.0	5.7	78.0
15493	Drama	115.0	6.1	408.0
15503	Action, Crime, Drama	125.0	5.8	44.0

1505 rows × 4 columns

NEW DATAFRAME FOR SCATTER

- - สร้าง dataframe ใหม่ที่เก็บ top 3 genres โดยสุ่มข้อมูลมา 300 จุด

```
movie_genre_top3_rand_df = movie_genre_top3_df.sample(300) # Use sample() to get a random row from the DataFrame
print(movie_genre_top3_rand_df.info())
movie_genre_top3_rand_df
```

	Genre	Duration	Rating	Votes
4480	Drama	135.0	6.9	43.0
12283	Drama	128.0	7.3	6.0
8466	Drama	116.0	8.6	155.0
5387	Drama	102.0	8.4	52.0
14952	Action, Crime, Drama	110.0	3.8	11.0
...	...	...	...	...
13872	Drama	120.0	7.5	27.0
3767	Drama	153.0	6.7	158.0
14270	Drama	133.0	4.4	133.0
8408	Drama, Romance	120.0	7.2	12.0
4380	Drama	122.0	2.9	554.0

300 rows × 4 columns

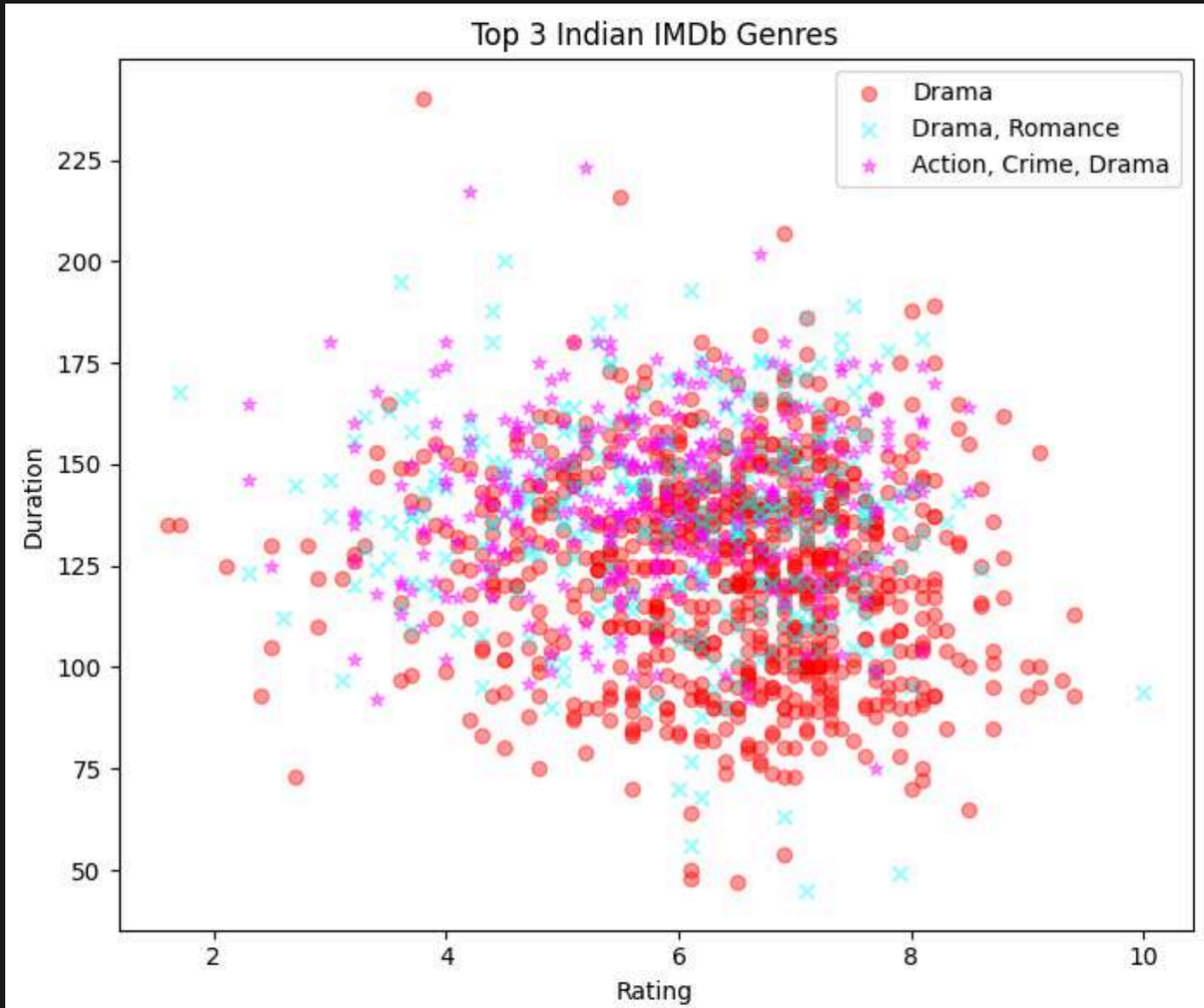
CREATE SCATTER PLOT FOR EACH GENRE

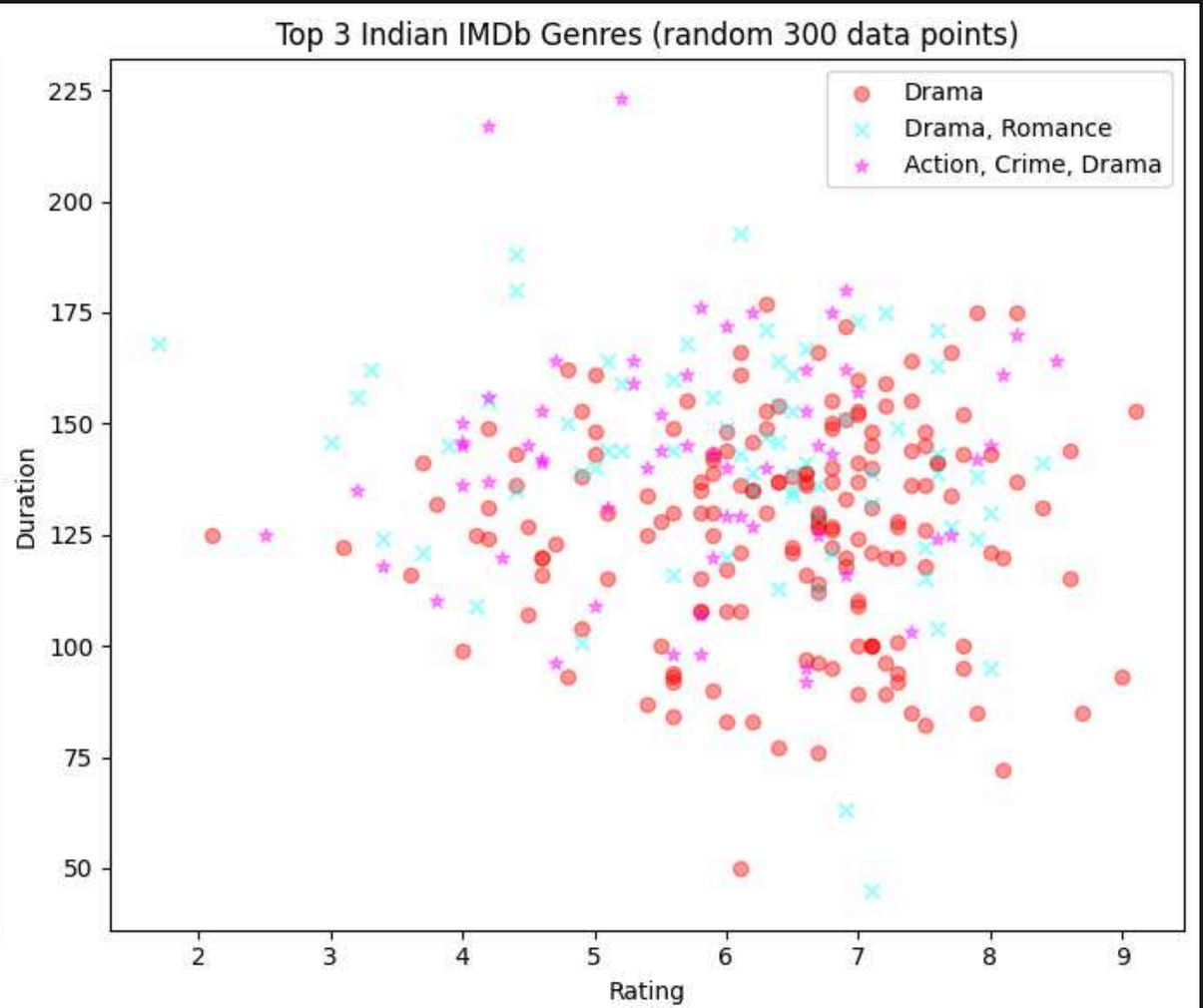
```
[21] # Create a figure with 2 subplots
     fig, axes = plt.subplots(1, 2, figsize=(14, 6))

     # First scatter plot
     axes[0].scatter(movie_genre_top3_df[movie_genre_top3_df['Genre'] == 'Drama']['Rating'],
                     movie_genre_top3_df[movie_genre_top3_df['Genre'] == 'Drama']['Duration'],
                     color='red', marker='o', alpha=0.4, label='Drama')

     axes[0].scatter(movie_genre_top3_df[movie_genre_top3_df['Genre'] == 'Drama, Romance']['Rating'],
                     movie_genre_top3_df[movie_genre_top3_df['Genre'] == 'Drama, Romance']['Duration'],
                     color='cyan', marker='x', alpha=0.4, label='Drama, Romance')

     axes[0].scatter(movie_genre_top3_df[movie_genre_top3_df['Genre'] == 'Action, Crime, Drama']['Rating'],
                     movie_genre_top3_df[movie_genre_top3_df['Genre'] == 'Action, Crime, Drama']['Duration'],
                     color='magenta', marker='*', alpha=0.4, label='Action, Crime, Drama')

     # Add labels and title to the first plot
     axes[0].legend()
     axes[0].set_title('Top 3 Indian IMDb Genres')
     axes[0].set_xlabel('Rating')
     axes[0].set_ylabel('Duration')
```

CREATE SCATTER PLOT FOR EACH GENRE

```
# Second scatter plot
axes[1].scatter(movie_genre_top3_rand_df[movie_genre_top3_rand_df['Genre'] == 'Drama']['Rating'],
                movie_genre_top3_rand_df[movie_genre_top3_rand_df['Genre'] == 'Drama']['Duration'],
                color='red', marker='o', alpha=0.4, label='Drama')

axes[1].scatter(movie_genre_top3_rand_df[movie_genre_top3_rand_df['Genre'] == 'Drama, Romance']['Rating'],
                movie_genre_top3_rand_df[movie_genre_top3_rand_df['Genre'] == 'Drama, Romance']['Duration'],
                color='cyan', marker='x', alpha=0.4, label='Drama, Romance')

axes[1].scatter(movie_genre_top3_rand_df[movie_genre_top3_rand_df['Genre'] == 'Action, Crime, Drama']['Rating'],
                movie_genre_top3_rand_df[movie_genre_top3_rand_df['Genre'] == 'Action, Crime, Drama']['Duration'],
                color='magenta', marker='*', alpha=0.4, label='Action, Crime, Drama')

# Add labels and title to the second plot
axes[1].legend()
axes[1].set_title('Top 3 Indian IMDb Genres (random 300 data points)')
axes[1].set_xlabel('Rating')
axes[1].set_ylabel('Duration')

# Adjust the layout to prevent overlapping
plt.tight_layout()

# Show the plots
plt.show()
```

CREATE SCATTER PLOT FOR EACH GENRE





MEMBERS

1.นายทินราช เพ็งเภา 653020208-5

2.นายรัชชานนท์ พันกาฬสินธุ์ 653020217-4

3.นางสาวณัฏฐ์กฤตา ไชยโกฏิ 653020206-9

4.นางสาวพรชนิตว์ เหล่าโยธี 653020212-4

THANK YOU!