
HOLLYWOOD

Nhóm 1 – Khóa D45

Phân tích và tìm ra những yếu tố giúp lọt top 1,000 phim điện ảnh có doanh thu cao nhất trên toàn cầu.

Dữ liệu phân tích:

Top

1,000

Phim có doanh thu phòng vé cao nhất.

+

500

Phim có chi phí sản xuất cao nhất.

Nguồn: https://kaggle.com/

Chi phí sản xuất

Production cost

Doanh số tuần công chiếu đầu tiên

Opening weekly

Thời điểm công chiếu

Release date

Mục lục

01 Giới thiệu về Dataset

02 Xử lý bằng Python

03 Phân tích bằng Power BI

04 Kết luận

Giới thiệu về Dataset

Nguồn dữ liệu: *https://kaggle.com/*
Thời gian: 1991 - 2022

1 Top 500

phim có chi phí sản xuất cao nhất

- ❑ Tên phim
- ❑ Ngày công chiếu
- ❑ Chi phí sản xuất
- ❑ Doanh thu nội địa
- ❑ Doanh thu toàn cầu
- ❑ Doanh thu tuần mở bán đầu tiên
- ❑ Phân loại
- ❑ Thể loại
- ❑ Thời lượng phim

2 Top 1,000

phim có doanh thu phòng vé cao nhất

- ❑ Tên phim
- ❑ Rating phim (do chuyên gia đánh giá)
- ❑ Lượt bình chọn của khán giả

Xử lý bằng Python

1. Upload file dữ liệu lên colab.research.google
2. Đọc file dữ liệu

+ Mã + Văn bản

[2] 18 giây

```
import pandas as pd #Import thư viện pandas
from google.colab import files
uploaded = files.upload() #Import files
```

Choose Files 2 files

- **Top_1000_Highest_Grossing_Movies_Of_All_Time.csv**(text/csv) - 252215 bytes, last modified: 10/16/2022 - 100% done
- **top-500-movies.csv**(text/csv) - 69811 bytes, last modified: 10/15/2022 - 100% done

```
Saving Top_1000_Highest_Grossing_Movies_Of_All_Time.csv to Top_1000_Highest_Grossing_Movies_Of_All_Time.csv
Saving top-500-movies.csv to top-500-movies.csv
```

[3] 0 giây

```
df1 = pd.read_csv('Top_1000_Highest_Grossing_Movies_Of_All_Time.csv')
df2 = pd.read_csv('top-500-movies.csv') #Đọc files
```

Xử lý bằng Python

3. Xử lý file dữ liệu top 500 phim có chi phí sản xuất cao nhất

```
[4]  df2.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   rank             500 non-null    int64
 1   release_date     499 non-null    object
 2   title            500 non-null    object
 3   url              500 non-null    object
 4   production_cost  500 non-null    int64
 5   domestic_gross   500 non-null    int64
 6   worldwide_gross  500 non-null    int64
 7   opening_weekend  479 non-null    float64
 8   mpaa             492 non-null    object
 9   genre            495 non-null    object
 10  theaters         479 non-null    float64
 11  runtime          487 non-null    float64
 12  year             499 non-null    float64
dtypes: float64(4), int64(4), object(5)
memory usage: 50.9+ KB
```

```
[5]  #Giữ lại các cột cần cho mục đích project
     df2 = df2.drop(['rank','url', 'theaters', 'year'], axis=1)
```

```
[6]  df2.isna().sum() #Kiểm tra giá trị null

release_date        1
title               0
production_cost     0
domestic_gross      0
worldwide_gross     0
opening_weekend    21
mpaa                8
genre               5
runtime            13
dtype: int64
```

Xử lý bằng Python

3. Xử lý file dữ liệu top 500 phim có chi phí sản xuất cao nhất

```
[11] df2 = df2.dropna() #Xoá các dòng chứa giá trị null
```

```
df2.isna().sum()
```

```
release_date       0
title              0
production_cost    0
domestic_gross     0
worldwide_gross    0
opening_weekend    0
mpaa               0
genre              0
runtime            0
dtype: int64
```

Xử lý bằng Python

3. Xử lý file dữ liệu top 500 phim có chi phí sản xuất cao nhất

```
[22] #Đổi datatype của runtime và opening_weekend
     df2['opening_weekend'] = df2['opening_weekend'].astype(int)
     df2['runtime'] = df2['runtime'].astype(int)
```

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 474 entries, 0 to 498
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   release_date     474 non-null    object
 1   title            474 non-null    object
 2   production_cost  474 non-null    int64 
 3   domestic_gross   474 non-null    int64 
 4   worldwide_gross  474 non-null    int64 
 5   opening_weekend  474 non-null    int64 
 6   mpaa             474 non-null    object
 7   genre            474 non-null    object
 8   runtime          474 non-null    int64 
dtypes: int64(5), object(4)
memory usage: 37.0+ KB
```

Xử lý bằng Python

4. Xử lý file dữ liệu top 1,000 phim có doanh thu phòng vé cao nhất

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Movie Title         1000 non-null   object 
 1   Year of Realease    1000 non-null   object 
 2   Genre               1000 non-null   object 
 3   Movie Rating        1000 non-null   float64
 4   Duration            1000 non-null   int64  
 5   Gross               1000 non-null   object 
 6   Worldwide LT Gross  1000 non-null   object 
 7   Metascore           1000 non-null   object 
 8   Votes               1000 non-null   object 
 9   Logline             1000 non-null   object 
dtypes: float64(1), int64(1), object(8)
memory usage: 78.2+ KB
```

```
df1.isna().sum() #Kiểm tra giá trị null
```

```
Movie Title           0
Year of Realease      0
Genre                 0
Movie Rating          0
Duration              0
Gross                 0
Worldwide LT Gross    0
Metascore             0
Votes                 0
Logline               0
dtype: int64
```

```
#Giữ lại các cột cần cho mục đích project
df1 = df1.drop(['Genre','Duration', 'Gross', 'Worldwide LT Gross', 'Metascore', 'Logline', 'Year of Realease'], axis=1)
```

Xử lý bằng Python

4. Xử lý file dữ liệu top 1,000 phim có doanh thu phòng vé cao nhất

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 3 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Movie Title   1000 non-null   object 
 1   Movie Rating  1000 non-null   float64
 2   Votes         1000 non-null   object 
dtypes: float64(1), object(2)
memory usage: 23.6+ KB
```

```
[31] #Xoá dấu , của cột
     df1['Votes'] = df1['Votes'].str.replace(',', '')

     # Chuyển đổi kiểu dữ liệu của cột 'Votes' sang số nguyên
     df1['Votes'] = df1['Votes'].astype(int)
```

```
df1 = df1.rename(columns={'Movie Title': 'title'})
```

Xử lý bằng Python

5. Kết hợp hai bảng

```
[35] df_combined = pd.merge(df1, df2, on='title') #Nối 2 table theo chiều ngang sử dụng key là title (tên phim)
```

```
print(df_combined) #In final ra
```

```
                        title  Movie Rating    Votes release_date  \
0                      Avatar           7.8  1236962   2009-12-17
1           Avengers: Endgame           8.4  1108641   2019-04-23
2                     Titanic           7.9  1162142   1997-12-18
3       Avengers: Infinity War           8.4  1062517   2018-04-25
4      Spider-Man: No Way Home           8.3   735006   2021-12-14
..                         ...           ...      ...          ...
346                      Hugo           7.5   323110   2011-11-23
347                  Watchmen           7.6   550223   2009-03-06
348         Jupiter Ascending           5.3   188877   2015-02-06
349              Flushed Away           6.6   126561   2006-11-03
350                The A-Team           6.7   259316   2010-06-11

     production_cost  domestic_gross  worldwide_gross  opening_weekend   mpaa  \
0          237000000       785221649       2910370905         77025481  PG-13
1          400000000       858373000       2797800564        357115007  PG-13
2          200000000       659363944       2207986545         28638131  PG-13
3          300000000       678815482       2048359754        257698183  PG-13
4          200000000       814108407       1912775610        260138569  PG-13
..               ...             ...              ...              ...    ...
346        180000000        73864507        180047784         11364505     PG
347        138000000       107509799        186976250         55214334      R
348        179000000        47482519        181982519         18372372  PG-13
```

Xử lý bằng Python

6. Xóa các dòng dữ liệu bị trùng lặp trong bảng dữ liệu

```
[37] duplicates = df_combined['title'].duplicated()
     print("Số lượng giá trị bị trùng lặp trong cột 'name':", duplicates.sum()) #Kiểm tra giá trị dulicate trong cột title

     Số lượng giá trị bị trùng lặp trong cột 'name': 13

[38] df_combined.drop_duplicates(subset='title', keep='first', inplace=True) #Xoá những dòng bị dulicate và giữ lại dòng đầu tiên
```

Xử lý bằng Python

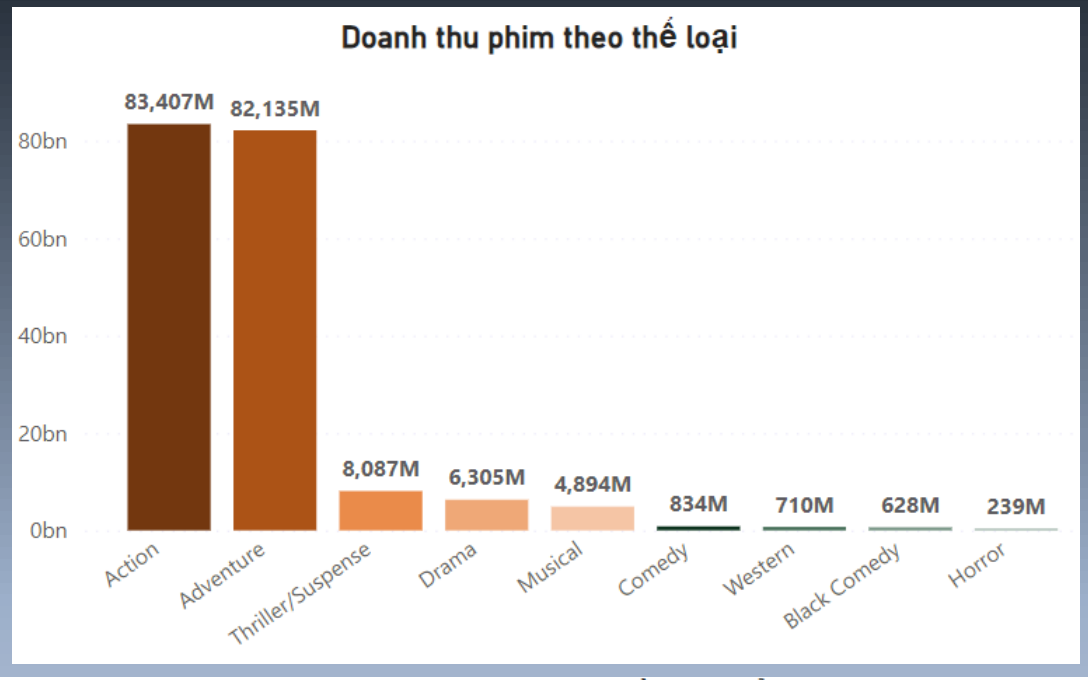
7. Download bảng dữ liệu cuối cùng về

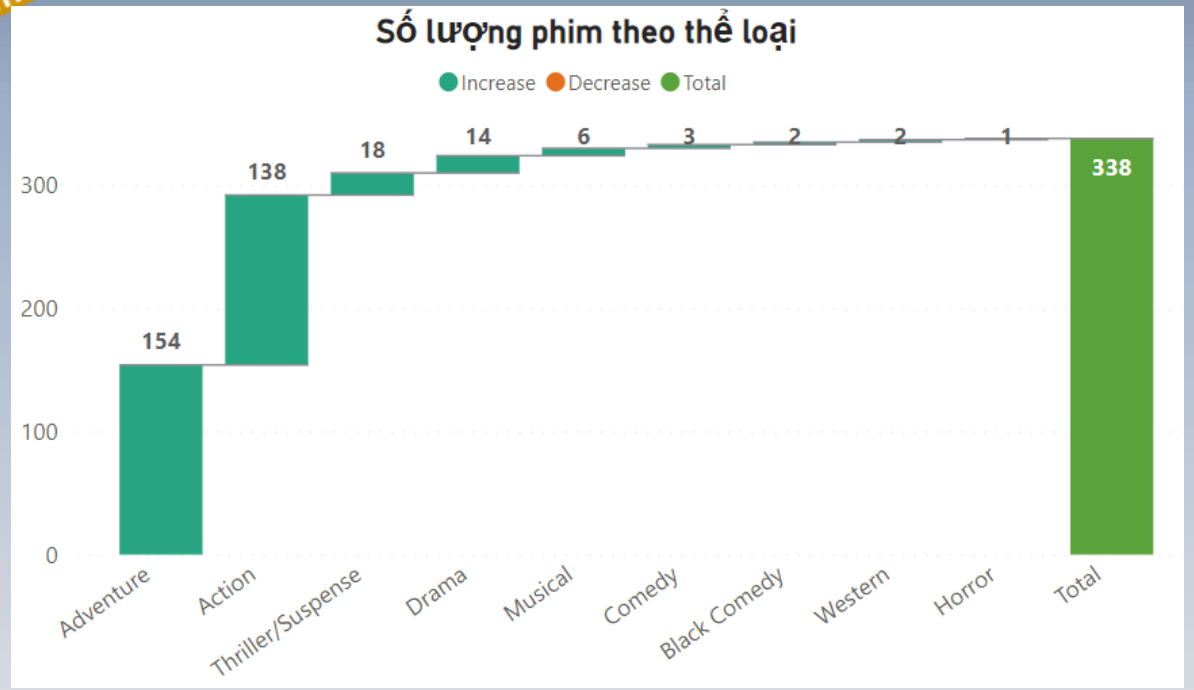
```
[35] df_combined.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 338 entries, 0 to 350
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   title            338 non-null    object
 1   Movie Rating     338 non-null    float64
 2   Votes            338 non-null    int64
 3   release_date     338 non-null    object
 4   production_cost  338 non-null    int64
 5   domestic_gross   338 non-null    int64
 6   worldwide_gross  338 non-null    int64
 7   opening_weekend  338 non-null    int64
 8   mpaa             338 non-null    object
 9   genre            338 non-null    object
 10  runtime          338 non-null    int64
dtypes: float64(1), int64(6), object(4)
memory usage: 31.7+ KB
```

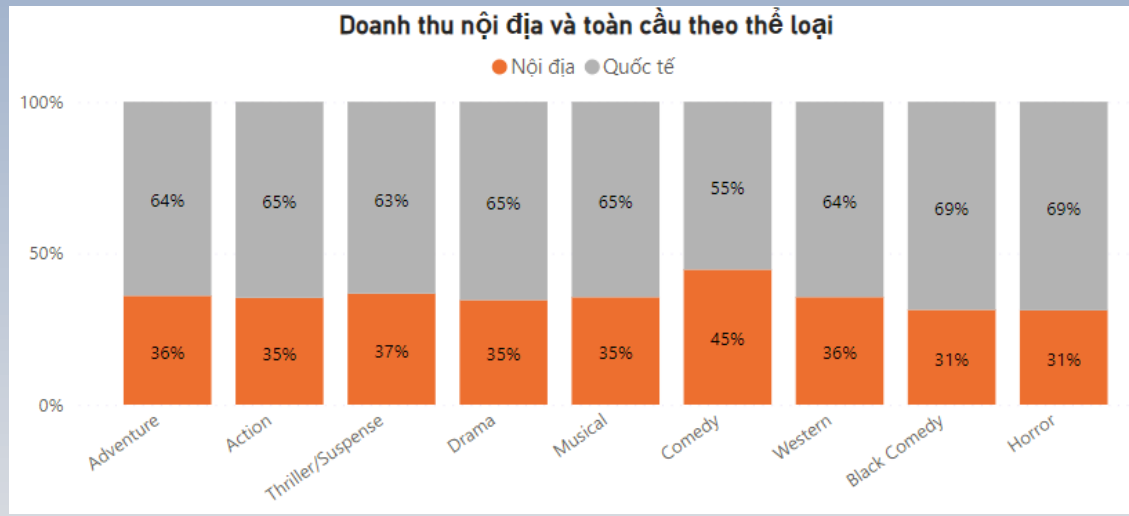
```
df_combined.to_csv('final.csv', index=False)
files.download('final.csv')
```

Phân tích bằng Power BI

Sơ lược về bảng dữ liệu

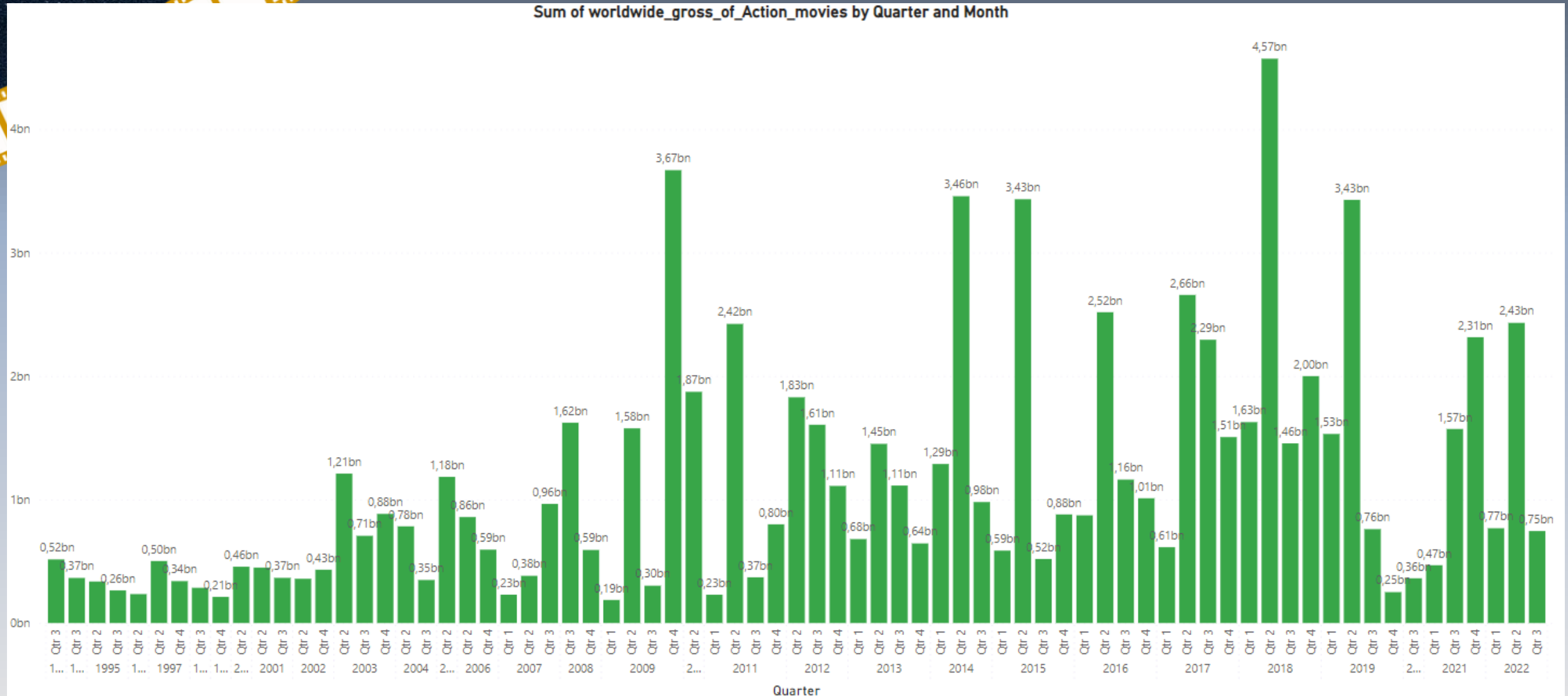






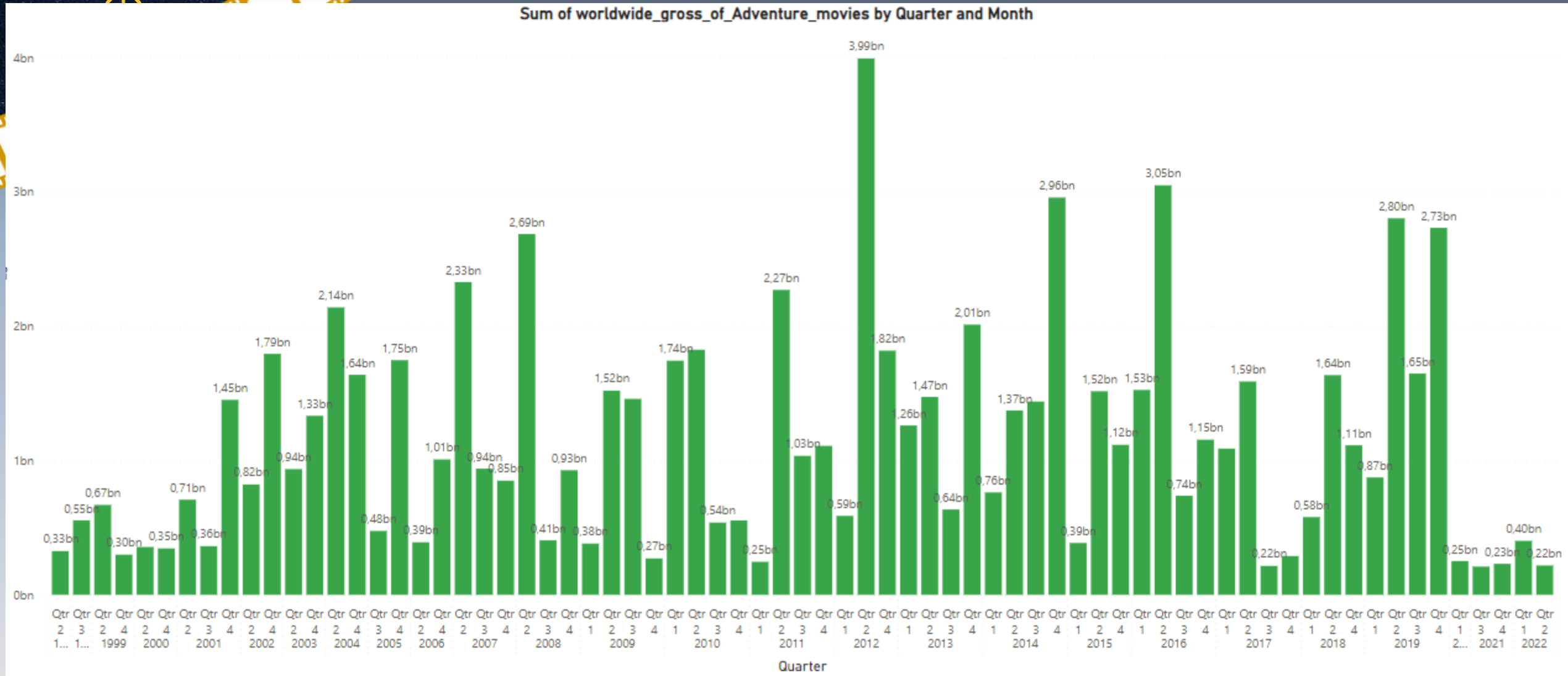
Phân tích bằng Power BI

Thể loại Action



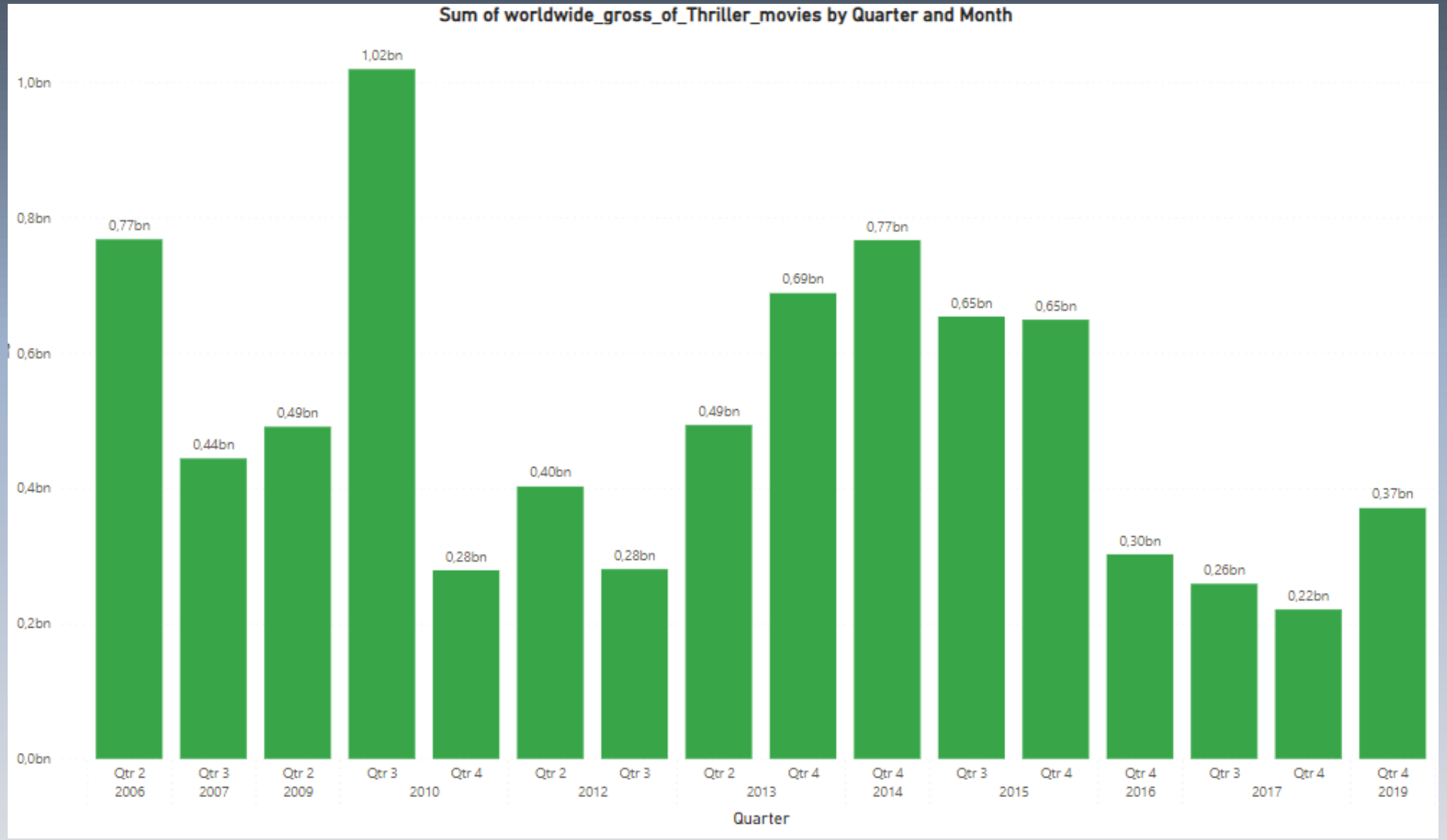
Phân tích bằng Power BI

Thể loại Adventure



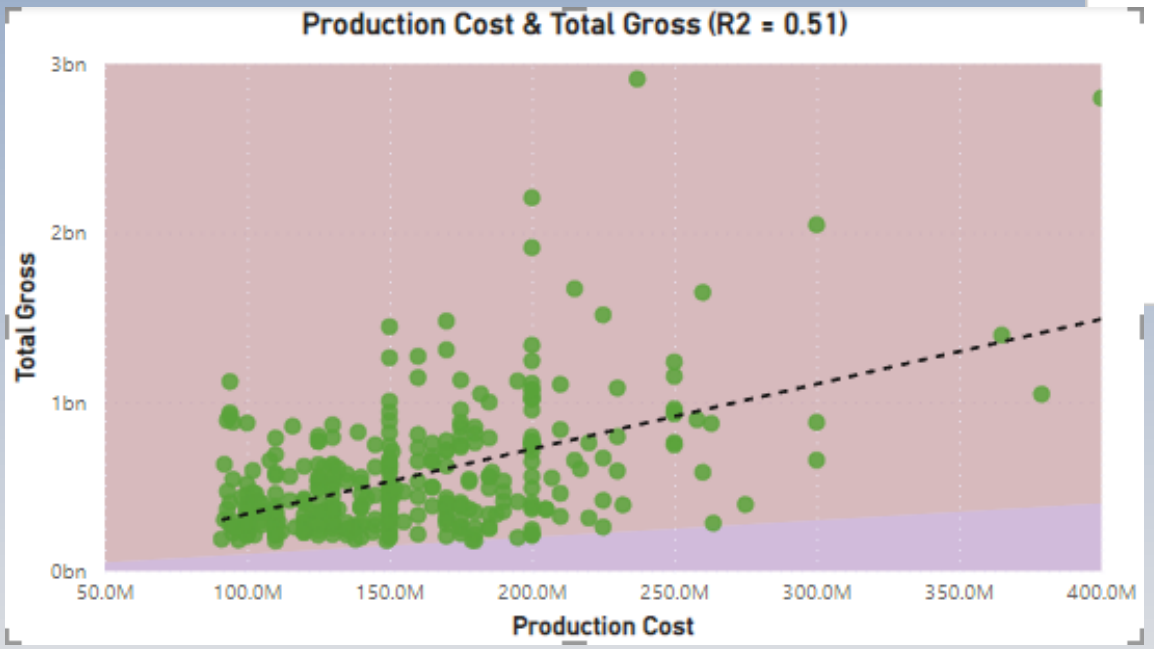
Phân tích bằng Power BI

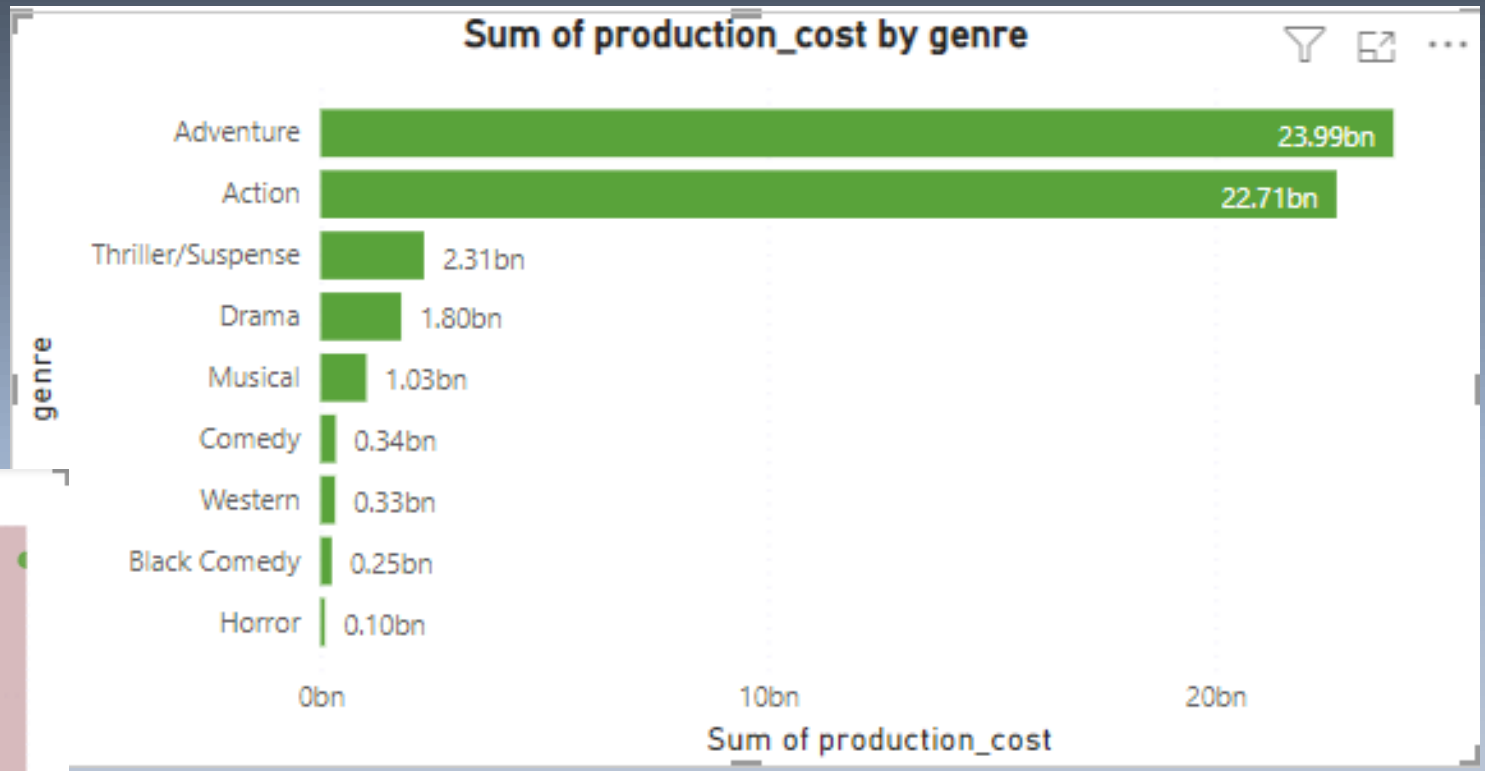
Thể loại Thriller



Phân tích bằng Power BI

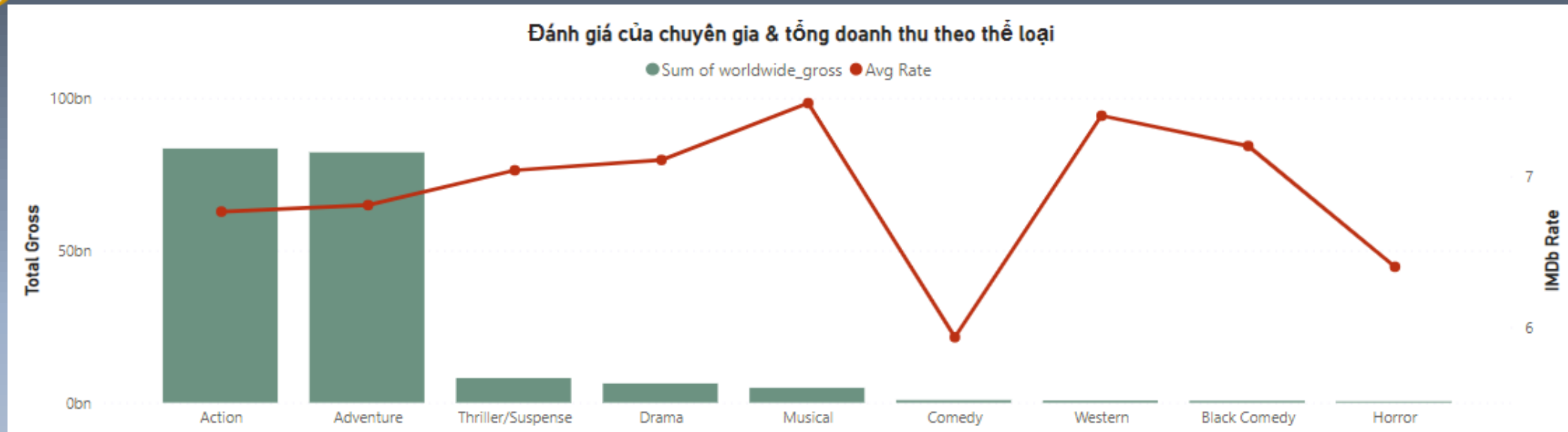
Mối tương quan giữa doanh thu & chi phí sản xuất

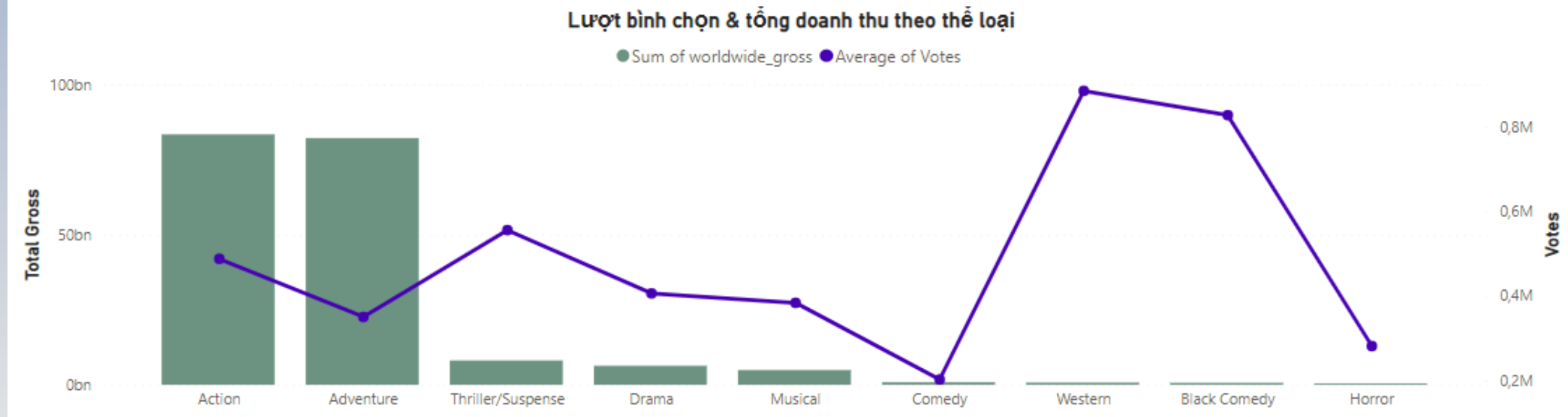




Phân tích bằng Power BI

Doanh thu – IMDb – Lượt bình chọn

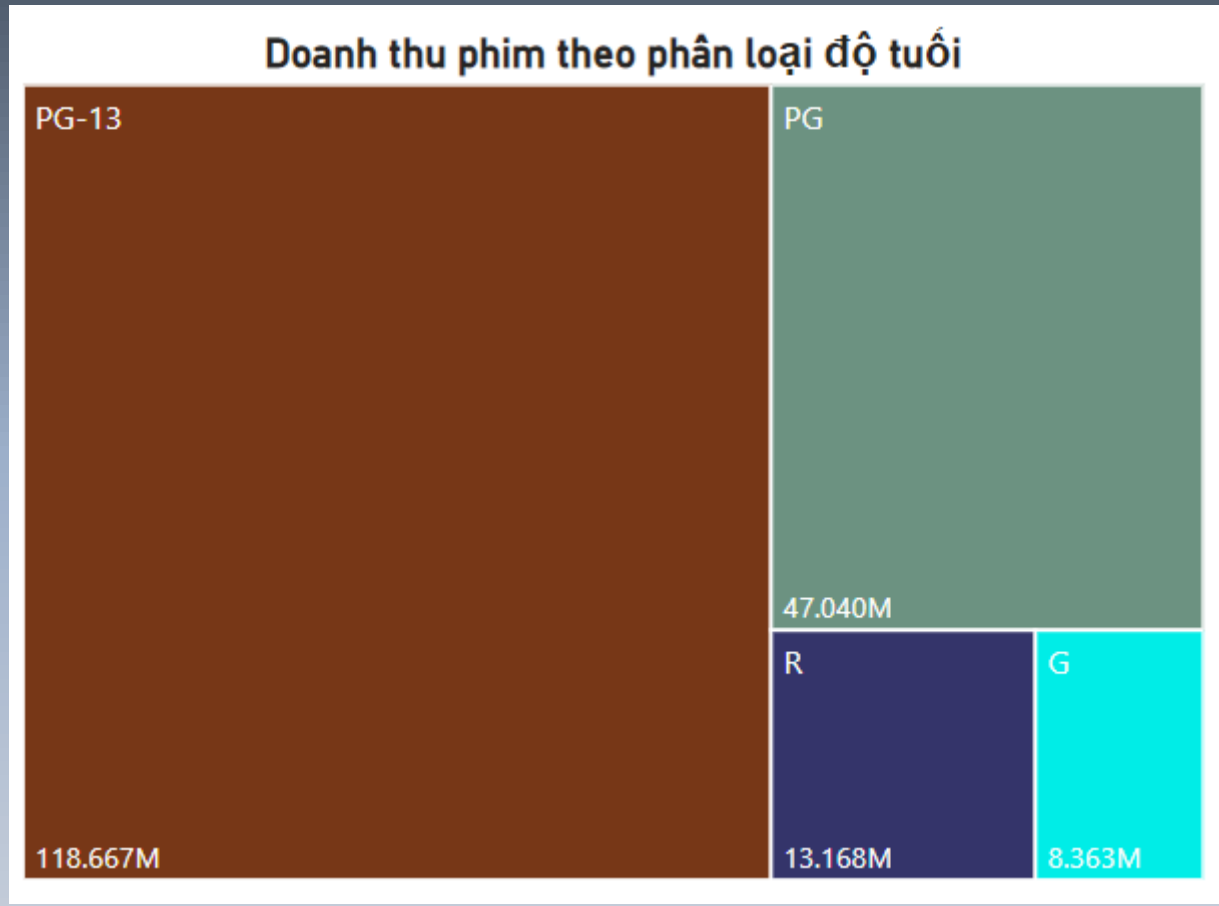




Phân tích bằng Power BI

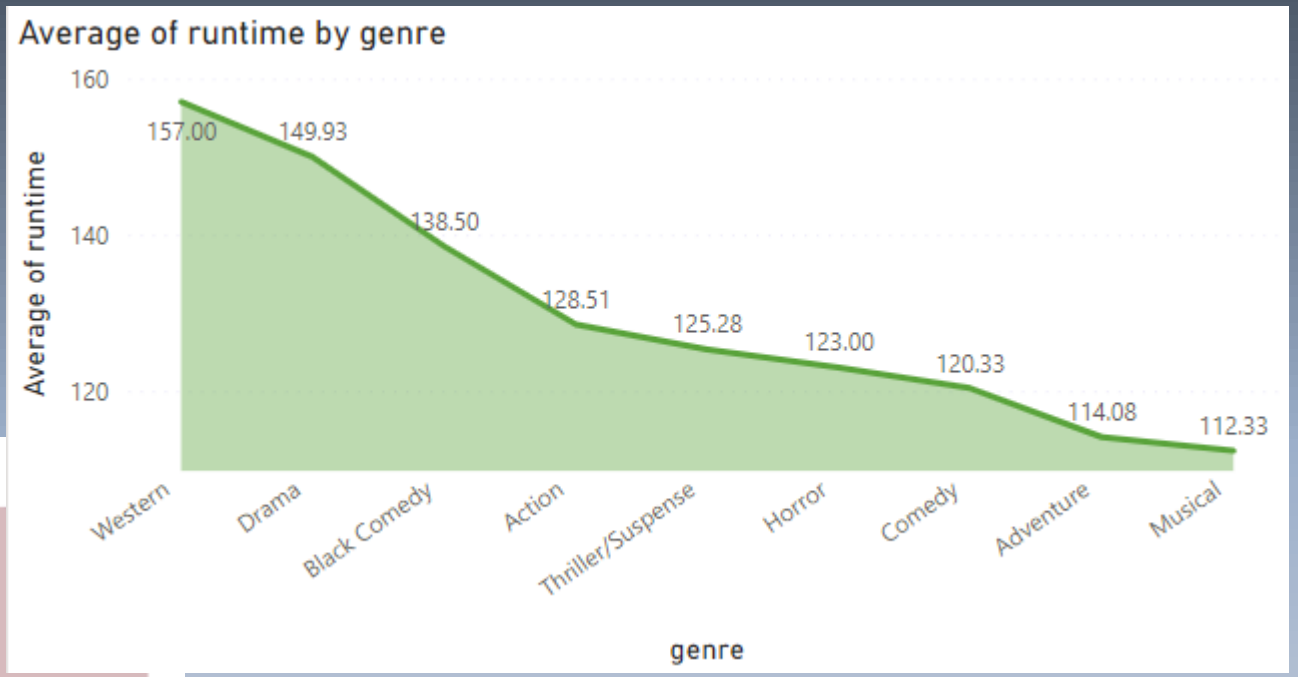
Doanh thu – Phân loại phim theo độ tuổi

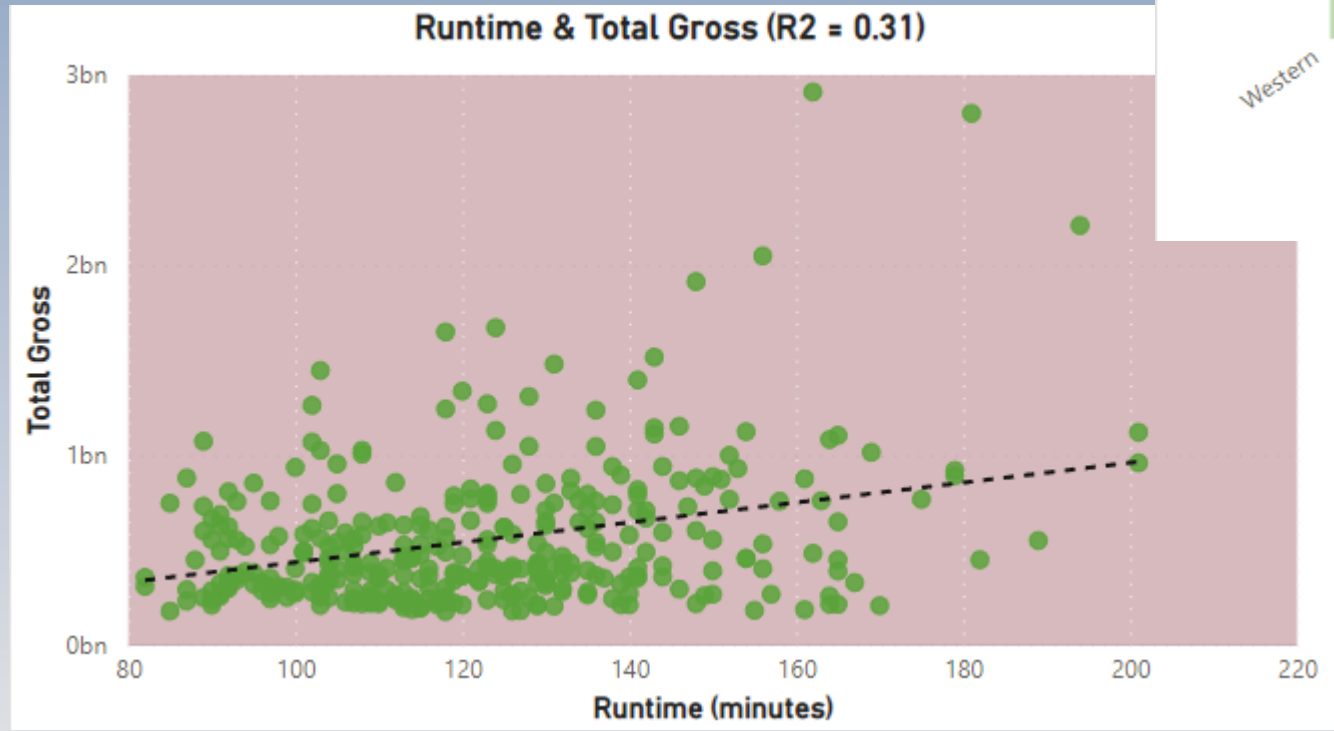
Biểu tượng phân loại	Nội dung phân loại
G	**G (General Audiences) – Phim dành cho mọi lứa tuổi** Mọi người đều có thể xem.
PG	**PG (Parental Guidance Suggested) – Phim có thể có một số chi tiết (hình ảnh, từ ngữ) không phù hợp với trẻ nhỏ. Bố mẹ cần cân nhắc khi cho con cái xem phim.** Một số hình ảnh có thể không thích hợp cho trẻ em.
PG-13	**PG-13 (Parents Strongly Cautioned) – Phim có một số chi tiết không phù hợp với trẻ dưới 13 tuổi.** Một số hình ảnh không thích hợp cho trẻ em dưới 13 tuổi.
R	**R (Restricted) – Thanh thiếu niên dưới 17 tuổi không được xem phim nếu không có sự đồng ý của người lớn.** Không dành cho người dưới 17 tuổi mà không có cha mẹ hoặc người giám hộ đi cùng do có thể gây hoảng loạn hoặc ảnh hưởng xấu đến tư duy, đạo đức của trẻ em. Mức này ở nhiều nước khác (nhất là các nước châu Á) sẽ bị xếp ở hạng "cấm trẻ em dưới 18 tuổi", đồng thời phải cắt bớt một số hình ảnh, nội dung không phù hợp với văn hóa bản địa. Ví dụ như phim điện ảnh *Sex and the City* ở Mỹ được dán nhãn R, nhưng khi chiếu ở Singapore thì bị xếp ở mức "cấm trẻ em dưới 18 tuổi", đồng thời phải cắt bỏ hết các cảnh khỏa thân, lộ ngực hay văng tục do *"không phù hợp với văn hóa của người Hoa, người Mã Lai và người Ấn"* (3 dân tộc chính của Singapore).



Phân tích bằng Power BI

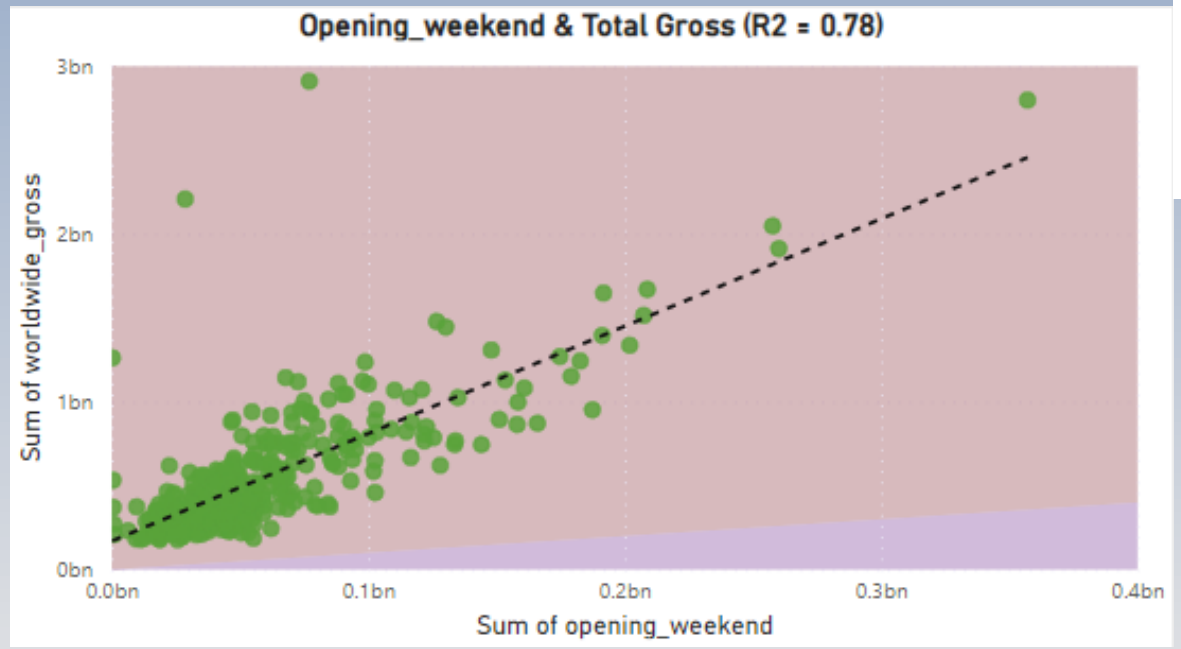
Doanh thu & độ dài phim

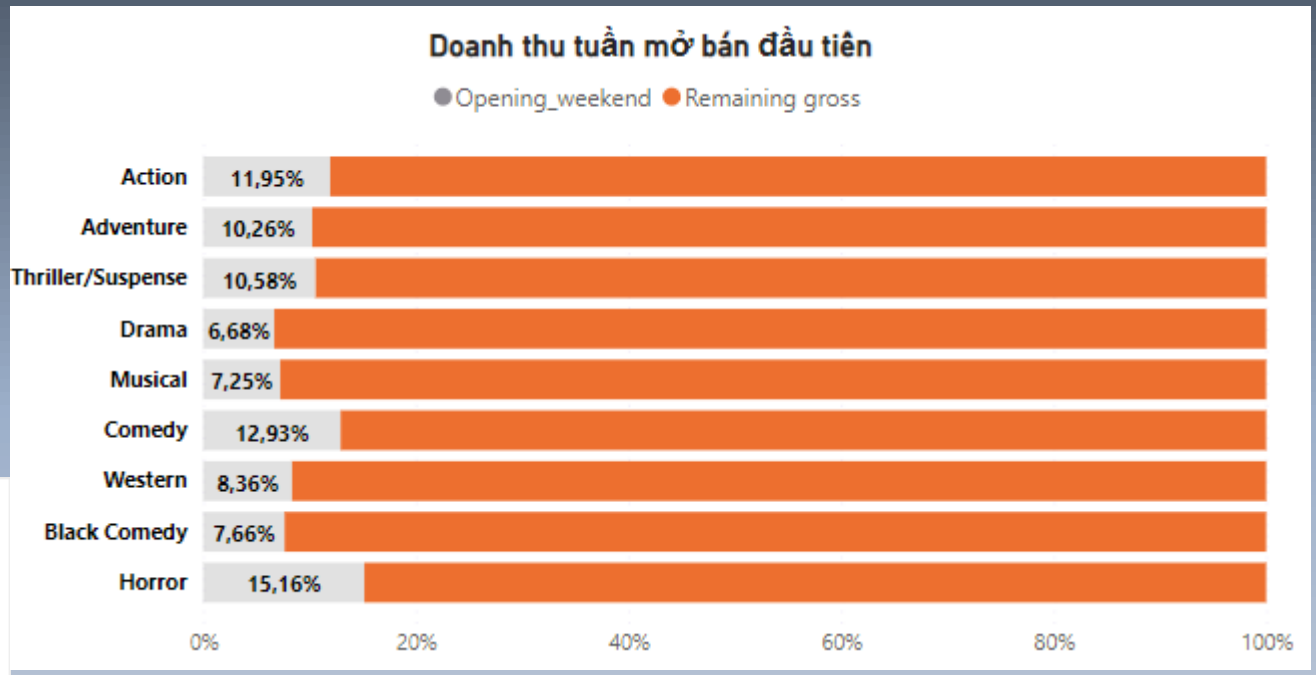




Phân tích bằng Power BI

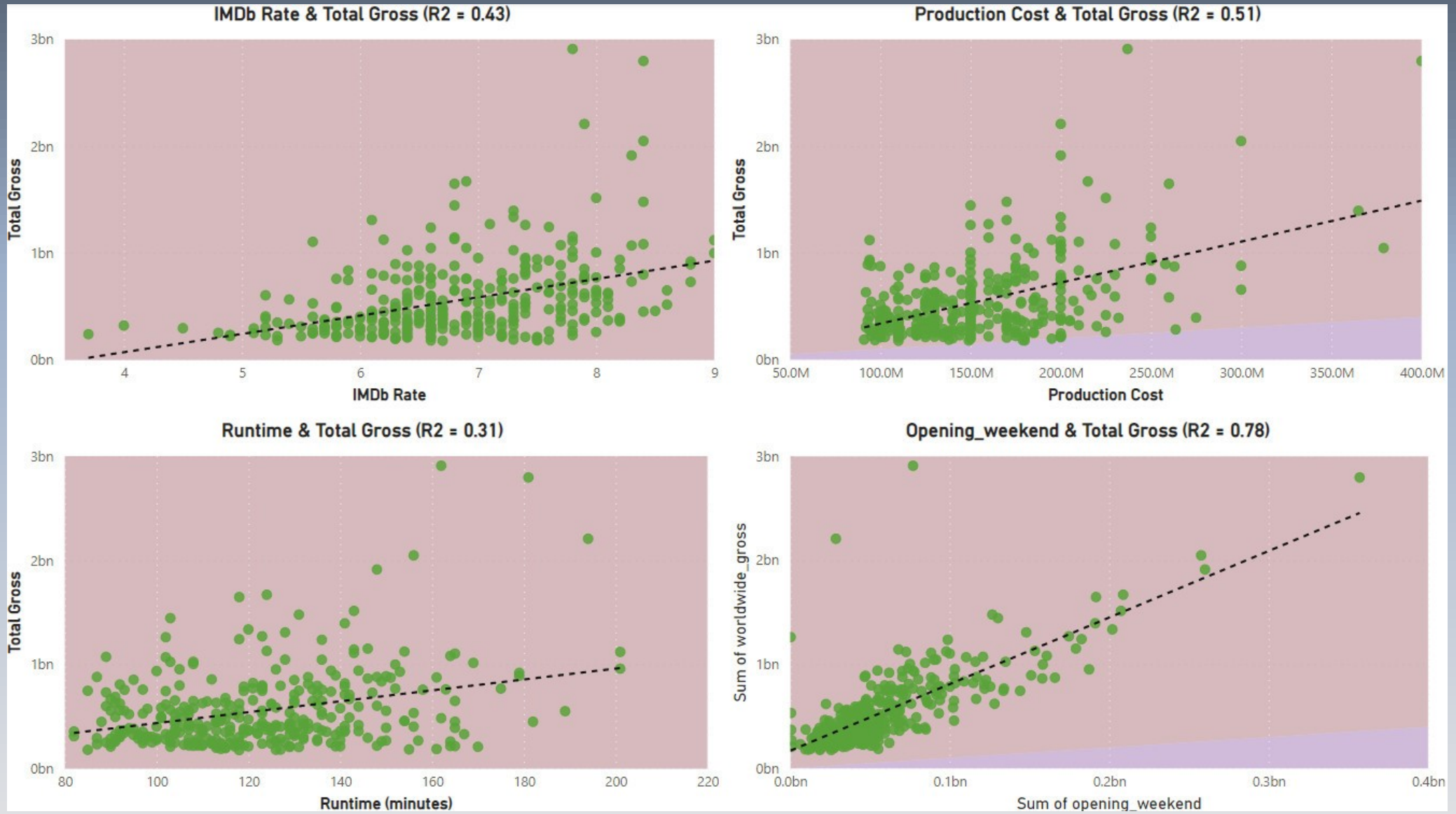
Doanh thu toàn cầu - Doanh thu tuần đầu công chiếu





Phân tích bằng Power BI

Kết luận mối tương quan giữa các biến



Kết luận

Kịch bản hay

Đạo diễn, diễn viên giỏi

Hậu kỳ tốt

…..

Lựa chọn thời điểm công chiếu thích hợp

Marketing đẩy mạnh doanh số tuần công chiều đầu tiên

Thank You!