final_project

May 24, 2024

1 Final Project - Exploration of movie genre relation to gross revenue - Chris Taylor

2 Introduction

2.1 Question of interest

I am interested in finding out which genre of Disney movie generates the highest domestic box office gross revenue using a collection of Disney datasets. I am interested to see if the genre that generates the highest gross revenue is not necessarily the genre with the most movies produced.

I expect the **Musical** genre to have the highest gross revenue due to the box office success of Disney musicals, although I expect the **Musical** genre to have fewer movies produced than most genres.

2.2 Dataset description

The below descriptions were taken directly from the project where the datasets were obtained.

"This project seeks to find the relationship between box office gross and MPAA ratings in Disney movies. The common assumption is that G-rated movies generate the most revenue because the largest portion of viewers are allowed admittance to these movies, children and adults alike. Our project includes five CSVs from four different sources, all of which we found in the form of HTML tables.

1. Sugarcane, "Walt Disney Animation Studios Films" - The link provides a list of Disney animated movies and the hero/villain character names in each movie.
2. The Numbers, "Movies Released by Walt Disney" - It is a chart and provides a list of Disney movies, and their genre, gross, and MPAA ratings.
3. Wikipedia, "List of Disney animated universe characters" - The link provides a complete list of Disney characters and their voice actors.
4. Wikipedia, "List of Walt Disney Animation Studios films" - The link provides a list of Disney animated movies and the director of each movie.
5. Wikipedia, "Annual gross revenues of The Walt Disney Company" - This is a Disney financial data chart which contains annual gross revenues by sections (includes studio entertainment, parks and resorts, etc.) from 1991-2016.The data are collected from the Disney annual report."

Each table is stored in a `.csv` file and contains information about Disney including movies, genres, ratings, revenue, characters, songs, actors and directors. I will be using the `disney_movies_total_gross` table formally described below:

disney_movies_total_gross.csv * This file contains information on Disney movies from 1937 to 2016, including the movie title, the release date, the genre, the MPAA rating, the gross revenue in the release year, and the gross revenue adjusted for inflation to 2016.

3 Methods and Results

Since I am only interested in analysing the genre and revenue of movies, I only need to use the **disney_movies_total_gross** table.

Let's import the table and determine information about the table.

[1]:
```python
# Lets import all the required libraries needed for this analysis
import altair as alt
import pandas as pd

# import all the required files
movie_genre_gross = pd.read_csv("data/disney_movies_total_gross.csv")
```

Table 1: Dataset for disney_movies_total_gross

[2]:
```python
movie_genre_gross
```

[2]:
```
                         movie_title  release_date      genre MPAA_rating  \
0    Snow White and the Seven Dwarfs  Dec 21, 1937    Musical           G
1                          Pinocchio   Feb 9, 1940  Adventure           G
2                           Fantasia  Nov 13, 1940    Musical           G
3                  Song of the South  Nov 12, 1946  Adventure           G
4                         Cinderella  Feb 15, 1950      Drama           G
..                               …           …         …         …
574         The Light Between Oceans   Sep 2, 2016      Drama       PG-13
575                   Queen of Katwe  Sep 23, 2016      Drama          PG
576                   Doctor Strange   Nov 4, 2016  Adventure       PG-13
577                            Moana  Nov 23, 2016  Adventure          PG
578     Rogue One: A Star Wars Story  Dec 16, 2016  Adventure       PG-13

      total_gross inflation_adjusted_gross
0    $184,925,485           $5,228,953,251
1     $84,300,000           $2,188,229,052
2     $83,320,000           $2,187,090,808
3     $65,000,000           $1,078,510,579
4     $85,000,000             $920,608,730
..            …                      …
574   $12,545,979              $12,545,979
575    $8,874,389               $8,874,389
576  $232,532,923             $232,532,923
577  $246,082,029             $246,082,029
578  $529,483,936             $529,483,936
```

```
[579 rows x 6 columns]
```

Table 2: Columns types for dataset disney_movies_total_gross

[3]:
```python
movie_genre_gross.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 579 entries, 0 to 578
Data columns (total 6 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   movie_title               579 non-null    object
 1   release_date              579 non-null    object
 2   genre                     562 non-null    object
 3   MPAA_rating               523 non-null    object
 4   total_gross               579 non-null    object
 5   inflation_adjusted_gross  579 non-null    object
dtypes: object(6)
memory usage: 27.3+ KB
```

The table has 579 rows and 6 columns. Every movie (**movie_title**) has the year it was released in (**release_year**), a **genre**, a MPAA rating (**MPAA_rating**), the total domestic gross revenue in the year it was released (**total_gross**, see notes 1 and 2), and total gross revenue adjusted to 2016 for inflation (**inflation_adjusted_gross**, see note 3).

Note: 1. Only domestic revenue is tracked. 2. Total gross revenue is calculated based on box office ticket sales and may include revenue from the next calendar year. Movies released late in the calendar year may stay in the box office into the next calendar year. 3. Inflation adjusted gross revenue is based on estimated ticket sales.

3.1 Clean the table for analysis

Let's start by cleaning the table for this analysis by: 1. Removing columns not of interest 2. Transforming columns to only the data of interest 3. Formatting columns for analysis 4. Addressing null entries

3.1.1 Remove columns not of interest

These columns are not of interest to this analysis, let's remove them: * **MPAA_rating**: The report that inspired this analysis studied the MPAA rating, thus I will not study it. * **total_gross**: I will use inflation adjusted gross revenue as the comparison of genres will span multiple years.

Table 3: Columns remaining after removing columns not of interest

[4]:
```python
clean_movie_genre_gross = movie_genre_gross.drop(columns=["MPAA_rating",␣
 ↪"total_gross"])
clean_movie_genre_gross.dtypes
```

[4]:
```
movie_title                 object
release_date                object
```

```
genre                          object
inflation_adjusted_gross       object
dtype: object
```

3.1.2 Transform columns to only the data of interest

My analysis will only need the year the movie was released. I will extract the **year** from the **release_date** and drop the column.

Table 4: Columns after year extracted from release_date

[5]:
```
clean_movie_genre_gross["release_date"] = pd.
↪to_datetime(clean_movie_genre_gross["release_date"])
clean_movie_genre_gross["year"] = clean_movie_genre_gross["release_date"].dt.
↪year
clean_movie_genre_gross = clean_movie_genre_gross.drop(columns=["release_date"])
clean_movie_genre_gross.dtypes
```

[5]:
```
movie_title                    object
genre                          object
inflation_adjusted_gross       object
year                            int64
dtype: object
```

3.1.3 Format columns for analysis

I will convert **inflation_adjusted_gross** to an integer for numerical comparisons. This will require stripping '$', replacing ',' then converting the column type. Now the clean dataframe will have 4 columns, with **year** and **inflation_adjusted_gross** converted to **int64**.

Table 5: Columns after converting to numeric types

[6]:
```
clean_movie_genre_gross = clean_movie_genre_gross.
↪assign(inflation_adjusted_gross =                                      ␣
↪clean_movie_genre_gross["inflation_adjusted_gross"].
                                                                         str.strip("$").str.
↪replace(',', '').astype(int))
clean_movie_genre_gross.dtypes
```

[6]:
```
movie_title                    object
genre                          object
inflation_adjusted_gross        int64
year                            int64
dtype: object
```

3.1.4 Address null entries

Last step to cleaning the data is to address null entries. Let's example the **clean_movie_genre_gross** table for null entries.

Table 6: Null entries per column

[7]:
```
clean_movie_genre_gross.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 579 entries, 0 to 578
Data columns (total 4 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   movie_title               579 non-null    object
 1   genre                     562 non-null    object
 2   inflation_adjusted_gross  579 non-null    int64
 3   year                      579 non-null    int64
dtypes: int64(2), object(2)
memory usage: 18.2+ KB
```

There are 17 null entries in the **genre** column. There are 3 options for handling movies with no **genre** specified: 1. Exclude from my analysis 2. Assign a genre by searching for the movie online 3. Analyse these movies as a group by assigning them a genric genre

Let's explore option 1 by calculating the percentage of gross revenue these movies represent to decide to include or exclude them from my analysis.

Table 7: Total gross revenue if genre specified

[8]:
```
no_genre_sum = clean_movie_genre_gross.loc[clean_movie_genre_gross['genre'].
 ↪isna(), "inflation_adjusted_gross"].sum()
gross_sum = clean_movie_genre_gross['inflation_adjusted_gross'].sum()
print("Genre not specified gross sum:", no_genre_sum)
print("Genre specified gross sum:", gross_sum)

no_genre_percentage = round((no_genre_sum / gross_sum) * 100, 2)
print("Percentage: ", no_genre_percentage, '%')
```

```
Genre not specified gross sum: 367603384
Genre specified gross sum: 68763500997
Percentage:  0.53 %
```

Thus I will exclude these movies with no genre as they represent $< 1\%$ of gross revenue.

Table 8: Null entries per column after cleaning

[9]:
```
clean_movie_genre_gross = clean_movie_genre_gross.dropna()
clean_movie_genre_gross.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 562 entries, 0 to 578
```

```
Data columns (total 4 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   movie_title              562 non-null    object
 1   genre                    562 non-null    object
 2   inflation_adjusted_gross 562 non-null    int64
 3   year                     562 non-null    int64
dtypes: int64(2), object(2)
memory usage: 22.0+ KB
```

3.2 Analysis of Genre and Gross Revenue

Now that the table is clean, I will analyze how movie genre and gross revenue are related by calculating for each genre: 1. The number of movies produced 2. The total gross revenue for all movies 3. The mean gross revenue per movie

I will use a function to calculate these stats as they may need to be calculated again for further analysis.

Before using the function: * Ensure the function and associated unit tests are PEP 8 compliant using the Black formatter * Ensure the unit tests pass

[10]:
```
!black script.py
```

```
All done!
1 file left unchanged.
```

[11]:
```
!black test_script.py
```

```
All done!
1 file left unchanged.
```

[12]:
```
!pytest test_script.py
```

```
============================= test session starts
==============================
platform linux -- Python 3.8.5, pytest-6.2.4, py-1.10.0, pluggy-0.13.1
rootdir: /home/jupyter/prog-python-ds-students/release/final_project
plugins: anyio-3.2.1, dash-1.20.0
collected 1 item


test_script.py .
[100%]

============================== 1 passed in 0.82s
===============================
```

Now calculate the stats for each genre.

Table 9: Stats per genre

[13]:
```python
from script import group_stats

stats_df = group_stats(clean_movie_genre_gross, "genre",␣
↪"inflation_adjusted_gross")
stats_df
```

[13]:

	genre	sum	mean	count
0	Action	5498936786	137473419	40
1	Adventure	24561266158	190397412	129
2	Black Comedy	156730475	52243491	3
3	Comedy	15409526913	84667730	182
4	Concert/Performance	114821678	57410839	2
5	Documentary	203488418	12718026	16
6	Drama	8195804484	71893021	114
7	Horror	140483092	23413848	6
8	Musical	9657565776	603597861	16
9	Romantic Comedy	1788872933	77777084	23
10	Thriller/Suspense	2151690954	89653789	24
11	Western	516709946	73815706	7

I see there are a few genres that most moviegoers would consider similar: * Comedy, Romantic Comedy, Black Comedy * Horror, Thriller/Suspense

I will combine these genres to make the analysis more relatable to the average moviegoer. * Comedy * Horror/Thriller/Suspense

Table 10: Stats per genre simplified

[14]:
```python
# Update all genres of Comedy to just Comedy
clean_movie_genre_gross.loc[clean_movie_genre_gross["genre"].str.
↪contains("Comedy"), "genre"] = "Comedy"

# Combine Horror and Thriller/Suspense
clean_movie_genre_gross.loc[clean_movie_genre_gross["genre"].str.
↪fullmatch("Horror"), "genre"] = "Horror/Thriller/Suspense"
clean_movie_genre_gross.loc[clean_movie_genre_gross["genre"].str.
↪fullmatch("Thriller/Suspense"), "genre"] = "Horror/Thriller/Suspense"

stats_df = group_stats(clean_movie_genre_gross, "genre",␣
↪"inflation_adjusted_gross")
stats_df
```

[14]:

	genre	sum	mean	count
0	Action	5498936786	137473419	40
1	Adventure	24561266158	190397412	129
2	Comedy	17355130321	83438126	208
3	Concert/Performance	114821678	57410839	2

```
4               Documentary    203488418    12718026     16
5                     Drama   8195804484    71893021    114
6  Horror/Thriller/Suspense   2292174046    76405801     30
7                   Musical   9657565776   603597861     16
8                   Western    516709946    73815706      7
```

Now I will chart the table to see which genre has the highest: 1. Number of movies 2. Total gross revenue for all movies 3. Mean gross revenue per movie

[15]:
```
movie_count_plot = (
    alt.Chart(stats_df, width=500, height=300)
    .mark_bar()
    .encode(
        x=alt.X("genre:N", title="Genre", sort="-y"),
        y=alt.Y("count:Q", title="Number of Movies Produced"),
    )
    .properties(title="Figure 1. Movies Produced per Genre")
)
movie_count_plot
```

[15]: alt.Chart(…)

The **Comedy** genre has the **highest** number of movies, while the **Musical** genre has the **3rd fewest** number of movies.

[16]:
```
total_gross_plot = (
    alt.Chart(stats_df, width=500, height=300)
    .mark_bar()
    .encode(
        x=alt.X("genre:N", title="Genre", sort="-y"),
        y=alt.Y("sum:Q", title="Total Gross Revenue ($)"),
    )
    .properties(title="Figure 2. Total Gross Revenue per Genre")
)
total_gross_plot
```

[16]: alt.Chart(…)

The **Adventure** genre has the highest total gross revenue, while the **Musical** genre has the **3rd highest** total gross revenue.

[17]:
```
mean_gross_plot = (
    alt.Chart(stats_df, width=500, height=300)
    .mark_bar()
    .encode(
        x=alt.X("genre:N", title="Genre", sort="-y"),
        y=alt.Y("mean:Q", title="Mean Gross Revenue ($)"),
    )
```

```
    .properties(title="Figure 3. Mean Gross Revenue per Genre")
)
mean_gross_plot
```

[17]: alt.Chart(…)

The **Musical** genre has the **highest** mean gross revenue. This is inline with my initial expectation that the **Musical** genre would have the highest gross revenue (highest mean gross revenue, 3rd highest total gross revenue) while having fewer movies produced than most genres (3rd fewest).

It is quite surprising that the **Musical** genre generates 3x higher mean gross revenue than the next genre despite representing only 16 of 579 movies. I suspect this is related to how gross revenue has changed over time. Let's examine the 16 movies in the **Musical** genre and plot how total gross revenue has changed for each decade and over time.

Table 11: Movies in the Musical genre

[18]:
```
musical_df = clean_movie_genre_gross.loc[clean_movie_genre_gross["genre"].str.
↪fullmatch("Musical")].sort_values(by="year")
musical_df
```

[18]:
```
                                  movie_title    genre  \
0            Snow White and the Seven Dwarfs  Musical
2                                    Fantasia  Musical
10                           Babes in Toyland  Musical
13                            The Jungle Book  Musical
15                             The Aristocats  Musical
17                   Bedknobs and Broomsticks  Musical
114                      Beauty and the Beast  Musical
142                                Swing Kids  Musical
161           The Nightmare Before Christmas  Musical
254                                     Evita  Musical
321                      Fantasia 2000 (IMAX)  Musical
330      Fantasia 2000 (Theatrical Release)  Musical
354               Beauty and the Beast (IMAX)  Musical
446  Tim Burton's The Nightmare Before Chr…  Musical
474      High School Musical 3: Senior Year  Musical
553                             Into the Woods  Musical

     inflation_adjusted_gross  year
0                  5228953251  1937
2                  2187090808  1940
10                  124841160  1961
13                  789612346  1967
15                  255161499  1970
17                   91305448  1971
114                 363017667  1991
142                  11468231  1993
```

```
161                  100026637  1993
254                   92077628  1996
321                   94852354  2000
330                   14238144  2000
354                   36980311  2002
446                   30737517  2006
474                  106308538  2008
553                  130894237  2014
```

[19]:
```python
musical_plot = (
    alt.Chart(musical_df, width=500, height=300)
    .mark_bar()
    .encode(
        x=alt.X("year:N", title="Year", bin=alt.Bin(maxbins=8)),
        y=alt.Y("inflation_adjusted_gross:Q", title="Gross Revenue", sort="x"),
    )
    .properties(title="Figure 4. Musical Total Gross Revenue by decade")
)
musical_plot
```

[19]: alt.Chart(...)

[20]:
```python
musical_plot = (
    alt.Chart(musical_df, width=500, height=300)
    .mark_circle()
    .encode(
        x=alt.X("year:N", title="Year"),
        y=alt.Y("inflation_adjusted_gross:Q", title="Gross Revenue", sort="x"),
    )
    .properties(title="Figure 5. Musical Gross Revenue by movie")
)
musical_plot
```

[20]: alt.Chart(...)

Note that total gross revenue is dominated by the 2 early Disney musicals **Snow White and the Seven Dwarfs** and **Fantasia** and has declined significantly over time.

4 Discussions

I analysed the Disney dataset for which genre had the highest gross revenue compared to movies produced. Before answering this question, I did some exploratory analysis on the different genres of movies. I observed that movies with no genre generate < 1% of total gross revenue and thus can be excluded from this analysis. I observed that there are a few genres of movies that most moviegoers would consider similar like Comedy and Romantic Comedy that should be combined for this analysis.

I found there are two ways to look that which genre has the highest gross revenue, either the **total** for all movies or the **mean** per movie. Generally the genres with the most movies (**Comedy** and **Adventure**) had the highest total gross revenue. However, the **Musical** genre had highest mean gross revenue, nearly 3x more than the next genre (**Adventure**), and 6x more than the genre with the most movies (**Comedy**) despite the **Musical** genre representing only 16 of 579 movies!

I further explored the **Musical** genre and found that 2 outliers (**Fantasia** and **Snow White and the Seven Dwarfs**) make up most of gross revenue, and that gross revenue per movie has decreased significantly over time. I was surprised to see this as High School Musical was a huge hit for Disney in the 2000s and missing from the Disney dataset. Turns out that High School Musical and the sequel High School Musical 2 were not theatrical releases and thus not included in the Disney dataset.

Another question that could be looked at given this dataset is how gross revenue changes over time with respect to genre. This is interesting to show how trends in audience tastes are related to the number of movies produced. For the past decade superhero (**Adventure**) movies have dominated the summer box office, has this resulted in superhero movies dominating the number of movies produced?

5 References

Not all the work in this notebook is original. Parts were borrowed from online resources and I take no credit for parts that are not mine. They were soley used for illustration purposes.

5.1 Resources used

- Data Source
  - This Disney database used in this work was curated by **Kelly Garrett**.
- Data Visualization
  - Inspiration for plotting the average gross revenue over the years was taken from **Kelly Garrett and Lichen Zhen**.
- Question Of Interest
  - The question of interest was inspired by **Kelly Garrett and Lichen Zhen**.
- High School Musical (franchise)
  - Reason why High School Musical and High School Musical 2 are not included in the Disney database.