Disney production house performance analysis

Introduction

Question(s) of interests

In this analysis, I will be investigating a questions associated with the collection of disney datasets. I am interested in find out how disney movies performed per year, moreover, as you already know disney is a big production house which worked multiple directors, it will interesting to determine how their movies perfomed in box office.

Dataset description

he disney data set is composed of five tables, `disney_movies_total_gross.csv` , `disney_revenue_1991-2016.csv` , `disney-characters.csv` , `disney-director.csv` , `disney_voice-actor` . Each table is stored in a `.csv` file and contains information about disney including the movies, their release dates, genre, total revenue, charactors and voice-actors. I'll be using `disney_movies_total_gross` , `disney-directors` , `disney-director` tables formally described below:

- **disney_movies_total_gross.csv**
  - This file contains information on disney including unique movie titles, their release dates, heros, villains and songs which are featured in these movies.
- **disney_revenue_1991-2016.csv**
  - This file includes information on the revenues generated from 1991 to 2016.These

Methods and Results

Since I am only interested in computing the gross of the movies and directors who worked with Disney from the available datasets, I will need to use tables that contain information on gross and directors. This implies that I will need to use the **gross** and the **director** tables.

However, before moving further, let us import the tables and do some basic visualizations.

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

# import all the required files
gross = pd.read_csv("data/disney_movies_total_gross.csv")
directors = pd.read_csv("data/disney-director.csv")
```

Lets see what the tables look like.

In [2]:
```
gross.head()
```

Out[2]:

	movie_title	release_date	genre	MPAA_rating	total_gross	inflation_adjusted_gross
0	Snow White and the Seven Dwarfs	Dec 21, 1937	Musical	G	$184,925,485	$5,228,953,251
1	Pinocchio	Feb 9, 1940	Adventure	G	$84,300,000	$2,188,229,052
2	Fantasia	Nov 13, 1940	Musical	G	$83,320,000	$2,187,090,808
3	Song of the South	Nov 12, 1946	Adventure	G	$65,000,000	$1,078,510,579
4	Cinderella	Feb 15, 1950	Drama	G	$85,000,000	$920,608,730

In [3]:
```
directors.head()
```

Out[3]:

	name	director
0	Snow White and the Seven Dwarfs	David Hand
1	Pinocchio	Ben Sharpsteen
2	Fantasia	full credits
3	Dumbo	Ben Sharpsteen
4	Bambi	David Hand

Lets get some other information about the **gross** table.

In [4]:
```
#Enter the code here to derive the information
gross.info()
print(gross['movie_title'].dtype)
print(gross['release_date'].dtype)
print(gross['total_gross'].dtype)
```

```
<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
object
object
object
```

The total_gross table has 579 rows and 6 columns. Every **movie_title** has a **release_date**, a **genre**, the **MPAA_rating** , **total_gross** , **inflation_adjusted_gross**.

Lets get some other information about the **directors** table. In this table we get the **name** of the movie and the **director** who directed that movie.

In [27]:
```
#Entered the code here to derive the information
directors.info()
print(directors['name'].dtype)
print(directors['director'].dtype)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 56 entries, 0 to 55
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   name      56 non-null     object
 1   director  56 non-null     object
dtypes: object(2)
memory usage: 1.0+ KB
object
object
```

The table has 56 rows with 2 columns. Movie's **name** and the **director** who directed the movie.

Before we start with further analysis we need to wrangle the data by first merging **gross** and **directors** table w.r.t. the name of the movie as a common element in to one table **movie_df**. This new table will contain only the required columns with correct datatypes.

In [6]:
```
# Outer Merge the gross and directors data Sets into a new dataframe object
merged_movie_df = gross.merge(directors, left_on = 'movie_title',right_on = 'name',how='outer')
movie_df = merged_movie_df.loc[:, merged_movie_df
                               .columns.drop(['name'])]

# Step 2 Convert the release date column to datetime, total_gross to int64

# Converted release_date to datetime
movie_df['release_date'] =  pd.to_datetime(movie_df['release_date'])

# Remove rows with Null total_Gross
movie_df = movie_df[movie_df['total_gross'].notna()]

# Converted total_gross to Int removed "$" and ","
movie_df['total_gross'] = movie_df['total_gross'].str.replace(',', '', regex=True).str.replace('$', '', regex=True).astype(int)

#Add a release_year column dataFrame
movie_df['release_year'] = pd.DatetimeIndex(movie_df['release_date']).year

# Place required columns in correct order
movie_df = movie_df[['release_year', 'release_date', 'movie_title', 'genre', 'total_gross','director']]

# Display the final merged dataframe
display(movie_df)
```

	release_year	release_date	movie_title	genre	total_gross	director
0	1937	1937-12-21	Snow White and the Seven Dwarfs	Musical	184925485	David Hand
1	1940	1940-02-09	Pinocchio	Adventure	84300000	Ben Sharpsteen
2	1940	1940-11-13	Fantasia	Musical	83320000	full credits
3	1946	1946-11-12	Song of the South	Adventure	65000000	NaN
4	1950	1950-02-15	Cinderella	Drama	85000000	Wilfred Jackson
...	...	...	...	...	...	...
574	2016	2016-09-02	The Light Between Oceans	Drama	12545979	NaN
575	2016	2016-09-23	Queen of Katwe	Drama	8874389	NaN
576	2016	2016-11-04	Doctor Strange	Adventure	232532923	NaN
577	2016	2016-11-23	Moana	Adventure	246082029	Ron Clements
578	2016	2016-12-16	Rogue One: A Star Wars Story	Adventure	529483936	NaN

579 rows × 6 columns

As a first visualization, lets look at the average number of movies released in each year. To do this, I will use the **gross** table. I will group by **release year** and then compute the 10 grossing year in analysis.

This will help us to identify the Top 10 highest grossing years of Disney

Let us create a table which groups by the movie release year and returns corresponding gross revenue

In [7]:
```
import final_script as s

# Create a Dataframe Grouped by Year, count number of movies, sum the revenue released by year

# group by year and find the total gross and movie count
release_year_group = movie_df.groupby('release_year').agg({'total_gross':sum, 'movie_title':'count'}).rename(columns={'movie_titl

# Sortby top 10 highest grossing years
top_10_years = s.custom_sort(release_year_group, 'total_gross', 10)

# Print Outputs i.e Top 10 Grossing year Analysis, Highest Grossing Year is Analysis, Performance of Year with MOST number of rel
top_N = str(top_10_years.shape[0])
print ("Top "+top_N+" grossing year is Analysis: ")
display(top_10_years)

print("Highest Grossing Year is Analysis: ")
display(release_year_group.loc[release_year_group['total_gross'] == top_10_years['total_gross'].max()])

print("Performance of Year with MOST number of released movies : ")
display(release_year_group.loc[release_year_group['movie_count'] == release_year_group['movie_count'].max()])

print("Performance of Year with LEAST number of released Mmovies : ")
display(release_year_group.loc[release_year_group['movie_count'] == release_year_group['movie_count'].min()])

average_movie_count = str(release_year_group['movie_count'].mean().round(0))
print ("Average Number of movies released in an year is " + average_movie_count)
```

```
Top 10 grossing year is Analysis:
```

	release_year	total_gross	movie_count
53	2016	2873393105	14
52	2015	2495662696	11
50	2013	1821352070	11
40	2003	1564114393	19
47	2010	1518975880	14
51	2014	1514179473	12
49	2012	1452972057	10
44	2007	1436787754	14
43	2006	1427356974	19
35	1998	1229279167	22

```
Highest Grossing Year is Analysis:
```

	release_year	total_gross	movie_count
53	2016	2873393105	14

```
Performance of Year with MOST number of released movies :
```

	release_year	total_gross	movie_count
32	1995	1131964294	32

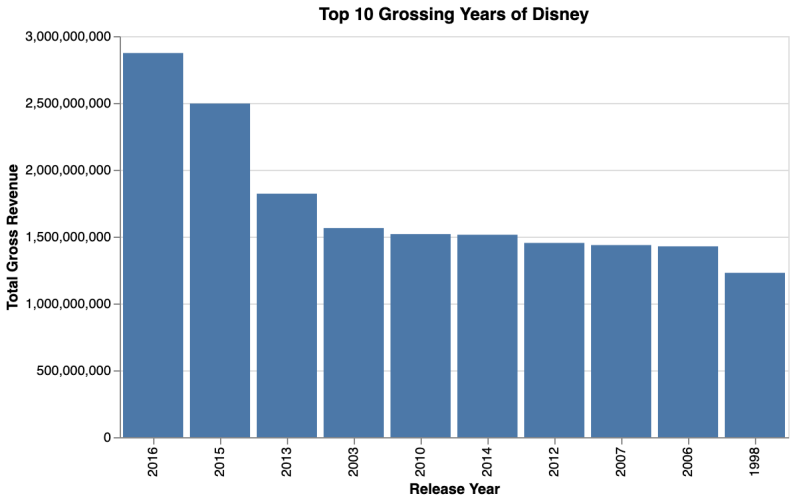
```
Performance of Year with LEAST number of released Mmovies :
```

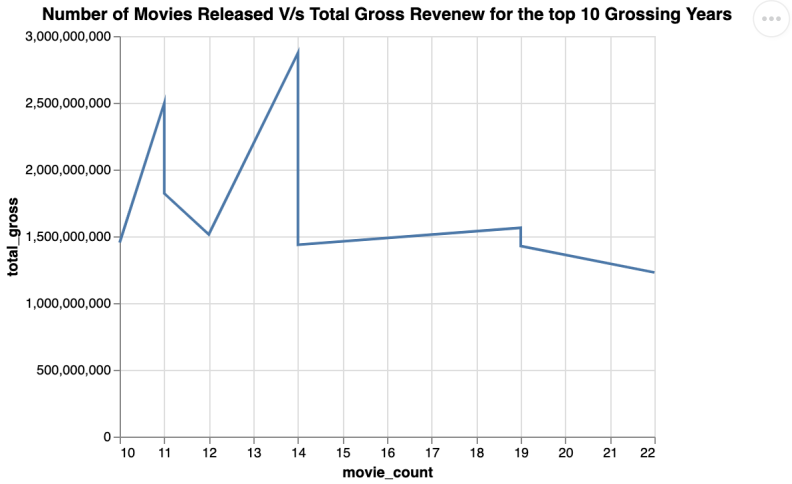
	release_year	total_gross	movie_count
0	1937	184925485	1
2	1946	65000000	1
3	1950	85000000	1
4	1954	28200000	1
5	1955	93600000	1
6	1959	9464608	1
8	1962	9230769	1
9	1963	22182353	1
10	1967	141843000	1
11	1968	21540050	1
13	1971	17871174	1
14	1975	31916500	1
16	1979	35841901	1

```
Average Number of movies released in an year is 11.0
```

Now let's plot a chart which demonstrates the relationship between movies released and the revenue generated by the top 10 grossing years individually. In our analysis we are not accounting for the inflated adjusted gross revenue.

In [16]:
```
# Enter your Step2 code here

# Use altair to generate a bar plot for top 10 grossing years
top_10_years_plot = (alt.Chart(top_10_years, width=500, height=300)
                     .mark_bar()
                     .encode(
                         x=alt.X("release_year:O", title="Release Year",sort="-y"),
                         y=alt.Y("total_gross:Q", title="Total Gross Revenue"),
                     )
                     .properties(title="Top 10 Grossing Years of Disney"))

display(top_10_years_plot)

# Use altair to generate a line plot to show relationship between movies released and total revenue for the top 10 grossing years

line_plot = (alt.Chart(top_10_years)
             .mark_line()
             .encode(
                x='movie_count',
                y='total_gross'
            )
            .properties(title="Number of Movies Released V/s Total Gross Revenue for the top 10 Grossing Years"))

display(line_plot)
```





From the above analysis we could identify the top 10 highest grossing years and can conclude that year **2016** was the highest grossing year with the revenue of **USD 2,873,393,105** and **14** movie releases. Whereas, gross revenue of the year **1995** which had the highest number of movie releases i.e **32 movies** was **USD 1,131,964,294**. We also identified that there were **13 years** with only **1** release and **1937** was the highest grossing year i.e **USD 184,925,485**. From the analysis of the relationship between count of movie release and gross revenue earned we realised that they are not directly proportional to each other and number or movie releases doesn't gurantee increased revenue.

As a second visualization, lets take a look at the count of movies and the gross revenue earned by each **director** who worked for Disney from the available dataset . To do this, only movies without **Null** values in the **director** column will be considered.

In [17]:
```
# group by year and find the total gross and movie count
director_group = movie_df.groupby('director').agg({'total_gross':sum, 'movie_title':'count'}).rename(columns={'A':'count','movie_
```

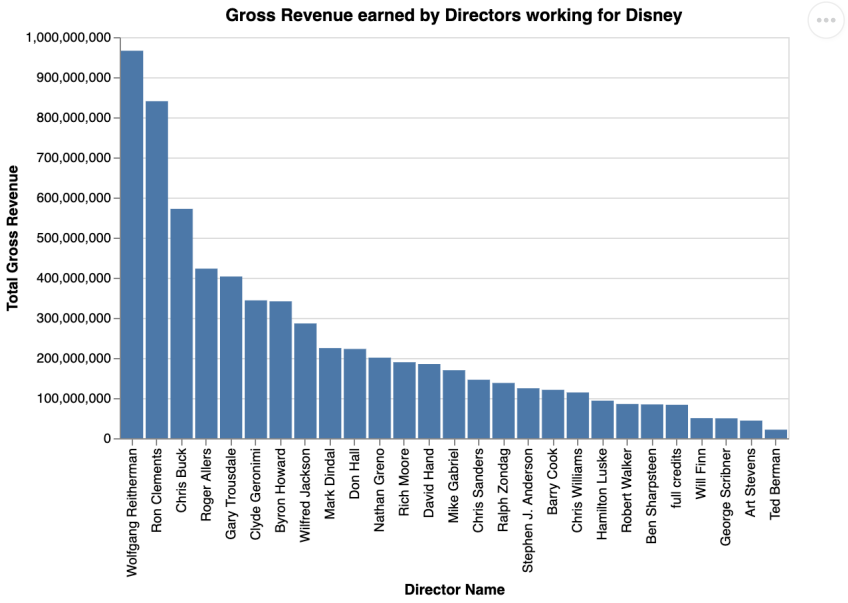
```
# Find the count of Directors Disney worked with
director_count = str(director_group['director'].count())
print("Number of directors who worked with Disney during this period : " + director_count)

# Find the performance of Director finding highest revenue
print("Details of the Highest revenue generating Director : ")
display(director_group.loc[director_group['movie_count'] == director_group['movie_count'].max()])
```

```
Number of directors who worked with Disney during this period : 27
Details of the Highest revenue generating Director :
```

	director	total_gross	movie_count
25	Wolfgang Reitherman	966009582	9

Now, lets plot a graph which demonstrates the gross revenue earned by each directors working for Disney

In [21]:
```
# Create a Histogram Plot which plots Revenue Geneated by Directors V/s Number of Movies
director_plot = (alt.Chart(director_group, width=500, height=300).mark_bar().encode(
        x=alt.X("director:O", title="Director Name",sort="-y"),
        y=alt.Y("total_gross:Q", title="Total Gross Revenue"),
    ).properties(title="Gross Revenue earned by Directors working for Disney"))

display(director_plot)
```



From the above analysis we identified that there were **27** directors who worked for Disney in the given dataset. **Wolfgang Reitherman** was the highest grossing director with the revenue of **USD 966,009,582** with total of **9** movie releases. With the help of the chart we are able to identify the performance of the directors sorted decendingly based on the gross revenue earned from their directed movies

Apply Black Formatting to the Scripting file containing Custom Sort function used in our analysis

In [22]:
```
!black final_script.py
```

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

Appling Black formatting to the Test script used to test the python file

In [31]:
```
!black test_final_script.py
```

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

Discussions

In this work, I analyzed the disney movies dataset and tried to compute the performance of Disney Production house w.r.t. the years range and directors who worked for disney. Before answering this question, I did some exploratory data analysis to see how Number of Movies Released in an year impacts the total gross revenue, for the top 10 grossing years of Disney. To our surprise we identifies, there were a few years where the average gross decreased drastically even though the number of movies released were comparatively higher. We were also able to determine which movie earned the most revenue.

It was interesting to learn that director **Wolfgang Reithrman** earned the highest gross revenue for Disney . My guess would have been **Chris Buck** given the popularity of the director. Surprisingly, **Chris Buck** earned only around USD $400,000,000$ from his directed movies which places him to the 3rd position out of all the directors who worked for Disney.

Another question that could be looked at given this dataset is the the top 10 most earned movies over the years. This is interesting because Disney's highest earning year was 2016. It was great to see progressinve they have been since 1998 which was deemed to be the least profitable year.

References

Example :

Not all the work in this notebook is original. Some parts were borrowed from online resources. I take no credit for parts that are not mine. They were soley used for illustration purposes. Lets give to **Ceasar** what belongs to **Ceasar**.

Resources used

- Data Tables
  - This Disney database used in this work was provided by UBC Extended learning team for exploratory purpose

Image obtained from here