# final_project

August 31, 2021

#

Exploratory Data Analysis of Disney Movies

##

Author: Pankti Shah || August 2021

---

---

# 1 Introduction

## 1.1 Question(s) of Interests

In this analysis, I will be investigating few questions associated with the collection of disney movie datasets. I am interested in finding to what extend key factors like the movie release month, genre, rating, and the movie director have had influence on the overall box office success of disney movies over the years.

I am interested in exploring these questions as it important to understand what factors influences audience to watch disney movies, and how has audiences' interests evolved over the years. I would expect release month would not have too much of an impact, unless it's during public holiday months. Since disney movies are released across the world, and each country has public holidays during different months, it should not have a significant impact on the box office. I am expecting comedy genres, and PG rated movies to have greater earnings as it would reach to a larger audience across the world.

## 1.2 Dataset Description

This notebook will be showing some exploratory data analysis for the `Disney Movies` dataset located here.

Following two disney movie dataset will be used: `disney-directors.csv`, and `disney_movies_total_gross.csv`. Each table is stored in a `.csv` file and contains different information about Disney movies including release date, genre, MPAA rating, total gross earning, inflation adjusted gross earning, and movie director.

1. **disney-directors.csv**
   - This file contains information on director(s) that has made Disney movies.
2. **disney_movies_total_gross.csv**

- This file includes information on disney movies gross_earning, and inflated_adjusted gross earning. Each movie is given a unique ID number, along with information about its release date, genre, and rating.

---

---

## 2 Methods and Results

Before beginning the analysis, releavant libraries, functions, and data tables that will be needed for analysis will be imported.

[512]:
```
# Import releavant libraries
import pandas as pd
import altair as alt
```

[513]:
```
# Import function
from frequency import frequency_table
from frequency_tests import test_sample_dataframe

# Unit testing to ensure imported function works
test_sample_dataframe()
```

[514]:
```
# Import relevant data tables
director = pd.read_csv('data/disney-director.csv')
movie_gross_earning = pd.read_csv('data/disney_movies_total_gross.csv')
```

In the table `movie_gross_earning`, release_date is an object. In order to analyze how earnings have changed over years, and to what extend movie release date influences gross_earnings, data in `release_date` column needs to be cleaned-up into appropriate format.

[515]:
```
# Data cleaning

# Date split into day, month, year for the table name movie_gross_earning
gross_earning_dates = movie_gross_earning['release_date'].str.split(
    ' ', expand = True).rename(
    columns = {0: 'Release_Month', 1: 'Release_Day', 2: 'Release_Year'})

gross_earning = pd.concat([movie_gross_earning, gross_earning_dates], axis = 1)

# Dropping release_date variable as it is redudant and not required
gross_earning = gross_earning.drop(columns = 'release_date')

# Converting total_gross and inflation_adjusted_gross values from object type␣
↪to int64
gross_earning['total_gross'] = pd.to_numeric(gross_earning['total_gross'].
↪replace('[^0-9\.-]','',regex=True))
```

```
gross_earning['inflation_adjusted_gross'] = pd.
 ↪to_numeric(gross_earning['total_gross'].replace('[^0-9\.-]','',regex=True))
```

Using the function `frequency_table()`, we will first find percent of movies that were released in each month.

More information about the function `frequency_table()` can be found in DocString in `frequency.py`. Recall, unit test for this function have already been validated earlier, but can be found in `frequency_test.py` as well.

Throughout this exploratory analysis, `Inflation adjusted gross` column data is used to normalize earnings. Next step is to find the total Disney movie earnings based on the movie release month. Filtered dataframe for this is saved as `earning_month`.

`combined_monthly_plot` displays the data results to help visualize our findings. Bar graph shows the percentage of movies that were released in each month (y-axis on left). Line graph shows total earnings for all the movies that were released in each of the respective months (y-axis on right).

[516]:
```
# Percentage of movies released in each months
movies_month = frequency_table(gross_earning, 'Release_Month','Month',␣
 ↪'Frequency')

# Total earning based on movie release month
earning_month = gross_earning.
 ↪drop(columns=['movie_title','genre','MPAA_rating','Release_Day']).groupby(
    by = 'Release_Month').sum().reset_index().rename(columns={'Release_Month':
 ↪'Month'})

# Figure 1 to show percentage of movies released each month along with their␣
 ↪inflation adjusted gross earnings
month_freq = alt.Chart(movies_month).mark_bar(color = 'lavender').encode(
            alt.X('Month', title = 'Movie Release Month'),
            alt.Y('frequency_percent', title = 'Movie Release Percentage %'))

month_earning = alt.Chart(earning_month).mark_line(
    color = 'red', point=True, size=3, radius=8).encode(
            alt.X('Month', title = 'Movie Release Month'),
            alt.Y('inflation_adjusted_gross', title = 'Inflation Adjusted Gross␣
 ↪Revenue ($)'))

combined_monthly_plot = alt.layer(month_freq, month_earning).properties(
    height=600, width=800, title="Figure 1: Influence of the Movie Release␣
 ↪Month on Earnings").resolve_scale(
    y = 'independent').configure_axis(
    labelFontSize=16, titleFontSize=16).configure_title(fontSize=24)

combined_monthly_plot
```

[516]: alt.LayerChart(…)

Next, we explore number of Disney movies that were released each year, and how their earnings have changed over the years. Has Disney movies gained more popularity over the years?

`combined_yearly_plot` displays percent of movies that have released each year in the bar graph (y-axis on left), and yearly total earnings in the line graph (y-axis on right). Important to note, bucketing for this analysis is based on the year of the movie release.

[517]:
```
# Percentage of movies released each year
movie_years = frequency_table(gross_earning, 'Release_Year', 'Year',␣
↪'Frequency')

# Total earning based on the year of the movie release
earning_year = gross_earning.
↪drop(columns=['movie_title','genre','MPAA_rating','Release_Day','Release_Month']).
↪groupby(
    by = 'Release_Year').sum().reset_index().rename(columns={'Release_Year':␣
↪'Year'})

# Figure 2 to show percentage of movies released each year along with their␣
↪inflation adjusted gross earnings
year_freq = alt.Chart(movie_years).mark_bar(color = 'greenyellow').encode(
            alt.X('Year', title = 'Release Year', sort='x'),
            alt.Y('frequency_percent', title = 'Movie Release (%)'))

year_earning = alt.Chart(earning_year).mark_line(color = 'blue', point=True).
↪encode(
            alt.X('Year', title = 'Release Year'),
            alt.Y('inflation_adjusted_gross', title = 'Inflation Adjusted Gross␣
↪Revenue ($)'))

combined_yearly_plot = alt.layer(year_freq, year_earning).properties(
    height=600, width=800, title="Figure 2: Influence of the Movie Release Year␣
↪on Earnings").resolve_scale(
    y = 'independent').configure_axis(
    labelFontSize=16, titleFontSize=16).configure_title(fontSize=24)

combined_yearly_plot
```

[517]: alt.LayerChart(…)

Next question is to what extend does the movie genre impact overall movie earnings? Are certain genre more popular than others?

To answer these, once again `frequency_table()` function is used to find percent of Disney movies released per each genre. `gross_earning` dataframe is used to find total movie earnings for each genre.

`combined_genre_plot` shows breakdown of movie genres (y-axis on left), and total earnings for the respective genre using the line graph (y-axis on right).

[518]:
```
# Percentage of movies released for each genre
movies_genre = frequency_table(gross_earning, 'genre','Genre', 'Frequency')

# Total earning for each genre
genre_earning = gross_earning.drop(columns=['movie_title', 'Release_Day',␣
↪'MPAA_rating']).groupby(
    by = 'genre').sum().reset_index()

# Figure 3 to show percentage of movies released for each genre along with␣
↪their inflation adjusted gross earnings
genre_freq = alt.Chart(movies_genre).mark_bar(color = 'lightcoral').encode(
            alt.X('Genre', title = 'Movie Genres', sort='-y'),
            alt.Y('frequency_percent', title = 'Movie Release (%)'))

genre_earning = alt.Chart(genre_earning).mark_line(
    color = 'blue', point=True, strokeWidth= 3).encode(
            alt.X('genre', title = 'Movie Genres'),
            alt.Y('inflation_adjusted_gross', title = 'Inflation Adjusted Gross␣
↪Revenue ($)'))

combined_genre_plot = alt.layer(genre_freq, genre_earning).properties(
    height=600, width=800, title="Figure 3: Influence of Movie Genre on␣
↪Earnings").resolve_scale(
    y = 'independent').configure_axis(
    labelFontSize=16, titleFontSize=16).configure_title(fontSize=24)

combined_genre_plot
```

[518]:
```
alt.LayerChart(…)
```

Using similar analysis techniques as above, we will now be exploring impact of movie rating on earnings.

`combined_rating_plot` shows percent breakdown of the `MPAA_rating` (y-axis on left), and total earnings for the respective rating using the line graph (y-axis on right).

[519]:
```
# Percent breakdown of Disney movies based on their ratings
movies_rating = frequency_table(gross_earning, 'MPAA_rating','Rating',␣
↪'Frequency')

# Total earning for each rating
rating_earning = gross_earning.drop(columns=['movie_title', 'genre',␣
↪'Release_Day']).groupby(
    by = 'MPAA_rating').sum().reset_index()
```

```
# Figure 4 shows percent of movies released by their rating along with their␣
 ↪inflation adjusted gross earnings
rating_freq = alt.Chart(movies_rating).mark_bar(color = 'khaki').encode(
            alt.X('Rating', title = 'Movie Rating'),
            alt.Y('frequency_percent', title = '% Disney Movie Releases'))

rating_earning = alt.Chart(rating_earning).mark_line(color='blue', point=True,␣
 ↪strokeWidth= 3).encode(
            alt.X('MPAA_rating', title = 'Movie Rating'),
            alt.Y('inflation_adjusted_gross', title = 'Inflation Adjusted Gross␣
 ↪Revenue ($)'))

combined_rating_plot = alt.layer(rating_freq, rating_earning).properties(
    height=600, width=800, title="Figure 4: Influence of Movie Rating on␣
 ↪Earnings").resolve_scale(
    y = 'independent', color='independent',
    shape='independent').configure_axis(labelFontSize=16, titleFontSize=16).
 ↪configure_title(fontSize=24)

combined_rating_plot
```

[519]:
```
alt.LayerChart(…)
```

To find influence of movie director on earnings, we first need to merge following two dataframe:`director_earning`,`gross_earning`. Inner merging is done based on common movie titles found in both of the dataframes. Movie titles that are not found in the `gross_earning` dataframe are excluded for the purpose of the analysis.

[520]:
```
# Dataframe merge to link movie_title with their respective director, and␣
 ↪earnings
director_earning = gross_earning.merge(director, left_on='movie_title',␣
 ↪right_on='name',how='inner',indicator=True).drop(
    columns=['Release_Day'])

# Percent breakdown of Disney movies based on their director
movies_director = frequency_table(director_earning, 'director','Movie␣
 ↪Director', 'Frequency')

# Amount of inflation adjusted gross_profit associated with each direactor
director_earn = director_earning.groupby(by = 'director').sum().reset_index()

# Figure 5 shows percent of movies released by their rating along with their␣
 ↪inflation adjusted gross earnings
director_freq = alt.Chart(movies_director).mark_bar(color = 'darkblue').encode(
            alt.X('Movie Director', title = 'Movie Directors'),
            alt.Y('frequency_percent', title = '% Disney Movie Releases'))
```

```
director_earnings = alt.Chart(director_earn).mark_line(color='darkred',␣
↪point=True, strokeWidth= 3.5).encode(
            alt.X('director', title = 'Movie Directors'),
            alt.Y('inflation_adjusted_gross', title = 'Inflation Adjusted Gross␣
↪Revenue ($)'))

combined_director_plot = alt.layer(director_freq, director_earnings).properties(
    height=600, width=800, title="Figure 5: Influence of Movie Director on␣
↪Earnings").resolve_scale(
    y = 'independent', color='independent',
    shape='independent').configure_axis(labelFontSize=16, titleFontSize=16).
↪configure_title(fontSize=24)

combined_director_plot
```

[520]: alt.LayerChart(…)

---

---

# 3 Discussions

## 3.1 Summary of Findings

Earlier, we did analysis to find to what extend key factors like the movie release month, genre, rating, and the movie director have had influence on the overall box office success of disney movies over the years.

Figure 1 shows as expected, there is no clear relationship between when the movie is released (month) and its gross earnings. Disney movies are popular watch all year round.

Figure 2 shows Disney movies have gained nearly exponential popularity (or earnings) over the years. As seen in the figure, prior to 1985 not many Disney movies were released. Most movies were released from mid 1980s to early 2000s, but Disney movie earning grew the most (nearly doubled) post 2010.

Figure 3 shows majority of the Disney movies have been comedy, and adventure (close-second). As expected, these genres are also higher earners than others.

Figure 4 shows nearly 35% of movies have been rated PG, and this rating has been the highest earner. This is aligned with my earlier expectation as PG movies can be viewed by larger and more diverse audience. PG-13 movies are the second highest earners, and the second most popular Disney movie category.

Figure 5 shows the most popular Disney movie director has been Will Finn, and then Ron Clements with accumulated movie earnings of ~$950 M and ~$850 M, respectively.

Overall, the data analysis shows the most popular directors, genre, and ratings trend with higher movie earnings.

### 3.2 Impact of Findings and conluding remarks

Disney movie exploratory analysis shows genres and ratings that have been most popular over the years. It provides insight into what sorts of movies audience enjoys watching. It provides opportunity to make similar themed movies for future. By knowing popular movie directors, future producers and directors can take inspiration from their films regarding techniques, characters, plot development as it has proven to appeal to wide range of audience. Because Disney movies have been earning the greatest amount in recent years, it shows that our audience today is still very engaged and invested with Disney, and that there is a huge market still waiting to be captured. For a movie marketing team, it also helps to understand a new Disney movie launch month will not have a significant impact on its overall revenue.

In conclusion, Disney movie genre, rating and movie director will have an influence on success of a disney movie. Other questions that I would be interested in answering includes which character or movies have been the most popular that it's worth making a sequel for, or how can we better commericalize less successful disney movies by re-making them to fit popular characterization, genre and rating. I would also be interested to know how has audience age-group changed over the years.

---

---

## 4 References

Not all the work in this notebook is original. Data were analyzed from online resources.

### 4.1 Resources used

- Data Source
  - Disney movie database used in this work was uploaded by **Kelly Garrett** on the sourced website.
  - Disney character success - dataset by Kgarrett. data.world. (2018, June 13). https://data.world/kgarrett/disney-character-success-00-16.
- Data Visualization
  - Inspiration for generating the plotting the average number of parts over the years was taken from **Andrea Sandico**.
  - Asindico. (2017, July 19). Data exploration. Kaggle. https://www.kaggle.com/asindico/data-exploration.

---

---

## 5 Miscellaneous

[522]:
```
# Bloack formatting
!black final_project.ipynb
!black frequency.py
!black frequency_tests.py
```

```
reformatted final_project.ipynb
All done!
1 file reformatted.
reformatted frequency.py
All done!
1 file reformatted.
reformatted frequency_tests.py
All done!
1 file reformatted.
```

[ ]: