

final_project

November 12, 2021

1 Final Project

2 Exploratory data analysis of the Disney datasets

Author: Xuan Hong

3 Introduction

3.1 Question(s) of interests

In this analysis, I will be investigating a question associated with the collection of Disney datasets. I am interested in finding which animation director creates the most revenue gross for Disney. It is interesting because Disney is one of the most successful animation movie productions companies, and we can find some clues of how the animation industry changes over the years. We all know people have different tastes in the arts, especially the movies, but the economic benefits of a movie are straightforward.

My favourite Disney director is [Byron Howard](#). He directed the [Zootopia](#), which won the 2016 Oscar and is also my favourite Disney animation. Although Howard is an amazing director, he is more an animator but a director for many years. This means that Howard hasn't directed a lot of movies yet. In order to make a more reasonable guess, I did some research and found a "[Top Animation Directors Ever](#)" list.

Top Animation Directors Ever

by [khfjjuk](#) | created - 14 Aug 2013 | updated - 10 Jun 2015 | Public



10 names

Sort by:

List Order



View:



1. Walt Disney

[Snow White and the Seven Dwarfs](#)

Walter Elias Disney was born on December 5, 1901 in Chicago, Illinois, the son of [Flora Disney](#) (née Call) and [Elias Disney](#), a Canadian-born farmer and businessperson. He had Irish, German, and English ancestry. Walt moved with his parents to Kansas City at age seven, where he spent the majority of ...



2. Hayao Miyazaki

Director | [Sen to Chihiro no kamikakushi](#)

[Hayao Miyazaki](#) is one of Japan's greatest animation directors. The entertaining plots, compelling characters, and breathtaking animation in his films have earned him international renown from critics as well as public recognition within Japan. The Walt Disney Company's commitment to introduce the ...



3. John Lasseter

Writer | [Toy Story 2](#)

Although born in Hollywood John and his twin sister, Johanna were raised in Wittier near Los Angeles. His parents were Jewell Mae (Risley), an art teacher, and Paul Eual Lasseter, a parts manager at a Chevrolet dealership. His mother's profession contributed to his interest in animation and ...



4. Ron Clements

Writer | [Hercules](#)

Ron Clements was born on April 25, 1953 in Sioux City, Iowa, USA as Ronald Francis Clements. He is a writer and director, known for [Hercules](#) (1997), [Aladdin](#) (1992) and [The Princess and the Frog](#) (2009). He has been married to Tamara Lee Glumace since February 25, 1989. 2

The first is “[Walt Disney](#)”, who is great but we can not count him because he is all the Disney movies producer. The second and third are “[Hayao Miyazaki](#)” and “[John Lasseter](#)”, who are fabulous but not Disney director. The fourth is “[Ron Clements](#)” who directs “[The Little Mermaid](#)”, “[Hercules](#)”, “[The Princess and the Frog](#)” etc. and yes, a Disney director. Therefore, I would expect **Ron Clements** to be the animation director who creates the most revenue gross for Disney.

3.2 Dataset description

The below descriptions were taken directly from the [website](#) where the datasets were obtained.

“Disney characters, box office success & annual gross income” “What are the trends in the Walt Disney Studio’s box office data? How do certain characters contribute to the success or failure of a movie?”

The Disney dataset is composed of 5 tables, `disney_movies_total_gross.csv`, `disney_revenue_1991-2016.csv`, `disney-characters.csv`, `disney-director.csv`, and `disney-voice-actors.csv`. Each table is stored in a .csv file and contains different information about Disney movies including its revenue, directors, release date, characters, and voice actors. I will be using the `disney_movies_total_gross` and `disney-director` tables formally described below:

- **disney_movies_total_gross.csv**
 - Disney movie box office gross and inflation adjustments.
- **disney-director.csv**
 - Only scraped the first director’s name of each animation film.

4 Methods and Results

Since Disney produces not only animation but also real actors’ movies, I will need to clean the `disney_movies_total_gross` because I am only interested in animation directors. This means I will only consider the movies listed in `disney-director`.

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

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

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

Let’s see what the tables look like.

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

```
[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

	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

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

```
[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

Let's get some other information about the **gross** table.

```
[4]: 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 **gross** table has 579 rows and 6 columns. Every **movie_title** has a **release_date**, the number of its **total_gross** in the release year, and the number of **inflation_adjusted_gross**, which adjust the revenue regarding to the 2016 value. The **genre** and the **MPAA_rating** have some missing entries, but they do not relate to our analysis.

Let's get some other information about the **directors** table.

```
[5]: directors.info()
```

```
<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

```

The **directors** table has 56 rows with 2 columns. Every movie has a **name**, and a **director**.

As a first visualization, let's look at the gross ranking of Disney movies from 1937 to 2016 regarding the 2016 adjusted value. To do this, I will use the **gross** table.

```

[6]: #convert the currency $ string in the "total_gross" and
      ↳ "inflation_adjusted_gross" to float
gross[gross.columns[4:]] = gross[gross.columns[4:]].replace('[\$,]', '',
      ↳ regex=True).astype(float)

#sort the gross value based on the "inflation_adjusted_gross"
gross_rank = gross.sort_values(by="inflation_adjusted_gross", ascending=False)

#Reset the index so we can plot using altair
gross_rank = gross_rank.reset_index(drop=True)
gross_rank.head(10)

```

```

[6]:
      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       101 Dalmatians      Jan 25, 1961  Comedy      G
4  Lady and the Tramp      Jun 22, 1955  Drama      G
5  Song of the South      Nov 12, 1946  Adventure      G
6  Star Wars Ep. VII: The Force Awakens  Dec 18, 2015  Adventure  PG-13
7                Cinderella      Feb 15, 1950  Drama      G
8    The Jungle Book      Oct 18, 1967  Musical  Not Rated
9    The Lion King      Jun 15, 1994  Adventure      G

      total_gross  inflation_adjusted_gross
0  184925485.0      5.228953e+09
1   84300000.0      2.188229e+09
2   83320000.0      2.187091e+09
3  153000000.0      1.362871e+09
4   93600000.0      1.236036e+09
5   65000000.0      1.078511e+09
6  936662225.0      9.366622e+08
7   85000000.0      9.206087e+08
8  141843000.0      7.896123e+08
9  422780140.0      7.616409e+08

```

Now that we have it in the proper format, we can generate a bar plot to visualize it.

```
[7]: # Use altair to generate a bar plot
gross_rank_plot = (
    alt.Chart(gross_rank, width=7000, height=300)
    .mark_bar(size=10)
    .encode(
        x=alt.X("movie_title:N", title="Disney Movie"),
        y=alt.Y("inflation_adjusted_gross:Q", title="Gross Value (adjusted up_
↳to 2016)"),
    )
    .properties(title="Gross Ranking of Disney Movies from 1937 to 2016")
)
gross_rank_plot
```

```
[7]: alt.Chart(...)
```

From this plot, we know that the top 5 animation is “Snow White and the Seven Dwarfs”, “Pinocchio”, “Fantasia”, “101 Dalmatians”, and “Lady and the Tramp”. It looks like they are all from the last century almost 70 years ago. Also, this plot is long and messy. In this case, I think I need to take a deeper look at the **gross** data more about its time.

Let’s create a second visualization of the yearly gross value of Disney movies. Before doing that, I need to get the year value separately for every movie and rearrange the columns we need.

```
[8]: #get the year, month, day separately of each movie
dates = (gross['release_date'].str.split(' ', expand=True).rename(columns = {0:
↳'month',
1: 'day',
2: 'year'}))

dates
```

```
[8]:      month  day  year
0      Dec  21,  1937
1      Feb   9,  1940
2      Nov  13,  1940
3      Nov  12,  1946
4      Feb  15,  1950
..      ...  ...  ...
574    Sep   2,  2016
575    Sep  23,  2016
576    Nov   4,  2016
577    Nov  23,  2016
578    Dec  16,  2016

[579 rows x 3 columns]
```

```
[9]: #assign the seperate year to the gross DataFrame
gross_year = gross.assign(year = dates['year'].astype(int))

#rearrange the gross_year columns
gross_year = gross_year.loc[:, ['year', 'movie_title', 'total_gross',
    ↪ 'inflation_adjusted_gross']]
gross_year
```

```
[9]:      year      movie_title  total_gross \
0    1937  Snow White and the Seven Dwarfs  184925485.0
1    1940                Pinocchio      84300000.0
2    1940                Fantasia      83320000.0
3    1946      Song of the South      65000000.0
4    1950        Cinderella      85000000.0
..    ...
574  2016  The Light Between Oceans      12545979.0
575  2016      Queen of Katwe       8874389.0
576  2016      Doctor Strange      232532923.0
577  2016                Moana      246082029.0
578  2016  Rogue One: A Star Wars Story  529483936.0

      inflation_adjusted_gross
0          5.228953e+09
1          2.188229e+09
2          2.187091e+09
3          1.078511e+09
4          9.206087e+08
..          ...
574         1.254598e+07
575         8.874389e+06
576         2.325329e+08
577         2.460820e+08
578         5.294839e+08
```

[579 rows x 4 columns]

Now, let's group by 'year' and count the total number of the 'inflation_adjusted_gross' and sort it in a descending way. To do this, I will import and use the script I created with a custom function that takes in a data frame and groups it by a certain column and then sort it by the values we interest.

```
[10]: # import the custom script
import script as ps

# run it on the data
gross_year_group = ps.custom_sort(gross_year, 'year', 'inflation_adjusted_gross')
gross_year_group
```

```

[10]:      year  inflation_adjusted_gross
0    1937      5.228953e+09
1    1940      4.375320e+09
2    2016      2.873393e+09
3    2015      2.495663e+09
4    1998      2.189031e+09
5    1995      2.188599e+09
6    2003      2.171245e+09
7    1996      2.157239e+09
8    1994      2.140691e+09
9    1999      1.981583e+09
10   2013      1.887864e+09
11   2006      1.834745e+09
12   1961      1.797807e+09
13   2007      1.757480e+09
14   1992      1.736350e+09
15   2000      1.711676e+09
16   2002      1.680600e+09
17   2010      1.622535e+09
18   2004      1.570219e+09
19   2014      1.560791e+09
20   2012      1.537669e+09
21   1997      1.474848e+09
22   1993      1.427313e+09
23   2009      1.359909e+09
24   2001      1.296292e+09
25   2005      1.273983e+09
26   1955      1.236036e+09
27   1990      1.226461e+09
28   1991      1.221190e+09
29   1987      1.186650e+09
30   2011      1.172579e+09
31   2008      1.135356e+09
32   1989      1.097851e+09
33   1946      1.078511e+09
34   1988      9.868419e+08
35   1950      9.206087e+08
36   1967      7.896123e+08
37   1986      5.774724e+08
38   1954      5.282800e+08
39   1977      3.636592e+08
40   1970      3.563622e+08
41   1981      1.816369e+08
42   1984      1.781327e+08
43   1985      1.731242e+08
44   1963      1.538708e+08
45   1968      1.386127e+08

```


46	1975	1.312469e+08
47	1979	1.203774e+08
48	1982	1.107312e+08
49	1962	1.095816e+08
50	1983	1.005495e+08
51	1971	9.130545e+07
52	1980	4.356021e+07
53	1959	2.150583e+07

Now that we have reasonable yearly values in the proper format, we can generate a bar plot to visualize it.

```
[11]: # Use altair to generate a bar plot
gross_year_plot = (
    alt.Chart(gross_year_group, width=700, height=300)
    .mark_bar()
    .encode(
        x=alt.X("year:Q", title="Year"),
        y=alt.Y("inflation_adjusted_gross:Q", title="Gross Value(adjusted up to_
→2016)"),
    )
    .properties(title="Total Gross Revenue of Disney Movies from 1937 to 2016")
)
gross_year_plot
```

```
[11]: alt.Chart(...)
```

From the above plot, we can see the historical development of Disney. In 1937 and 1940, Disney made its highest value achievement. Then it slows down and creates one animation almost every 5 years. Since 1980, Disney started to create movies every year and get a high number gross every 3 – 4 years. Generally, the number keeps growing from 1980 and doubles quickly in 1987. It reaches its first peak around 1995 and decreases slightly from 2000 to 2012. After that, the number increases again and reaches the third-highest gross in Disney history in 2016. This plot gives me more confidence in recent 30 years animations.

As a third visualization, let's take a look at the number of animations directed by every director. To do this, I will use **directors** DataFrame.

Now let's group by 'director' name and count the frequency of the animation 'name'.

```
[12]: # group by director and count the frequency of movies
animation_with_directors_group = pd.DataFrame(directors.
→groupby('director')['name'].count())

# Reset the index so we can plot using altair
animation_with_directors_group = animation_with_directors_group.reset_index()
animation_with_directors_group
```

```
[12]:
```

	director	name
0	Art Stevens	1
1	Barry Cook	1
2	Ben Sharpsteen	2
3	Byron Howard	1
4	Chris Buck	2
5	Chris Sanders	1
6	Chris Williams	1
7	Clyde Geronimi	3
8	David Hand	2
9	Don Hall	1
10	Gary Trousdale	3
11	George Scribner	1
12	Hamilton Luske	2
13	Jack Kinney	4
14	Mark Dindal	2
15	Mike Gabriel	2
16	Nathan Greno	1
17	Norman Ferguson	1
18	Ralph Zondag	1
19	Rich Moore	1
20	Robert Walker	1
21	Roger Allers	1
22	Ron Clements	7
23	Stephen J. Anderson	2
24	Ted Berman	1
25	Wilfred Jackson	1
26	Will Finn	1
27	Wolfgang Reitherman	7
28	full credits	2

Let's do a scatter plot of the directors with the most animations.

```
[13]: directors_plot = (
    alt.Chart(animation_with_directors_group, width=500, height=300)
    .mark_circle()
    .encode(
        x=alt.X("director:O", sort="y", title="Disney directors"),
        y=alt.Y("name:Q", title="Numbers of animations"),
    )
    .properties(title="Numbers of animations directed by Disney directors")
)
directors_plot
```

```
[13]: alt.Chart(...)
```

It seems that both our expectation **Ron Clements**, and a director **Wolfgang Reitherman**, who directed “**101 Dalmatians**”, “**The Jungle Books**” etc. have created 7 animations. What a fierce

competition!

Now, it is time to answer our original questions. First, I need to change the column “name” in **directors** to “movie_title” to match the **gross** for merging the data.

```
[14]: # change the column name to match
directors = directors.rename({'name': 'movie_title'}, axis=1)
directors
```

```
[14]:
```

	movie_title	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
5	Saludos Amigos	Jack Kinney
6	The Three Caballeros	Norman Ferguson
7	Make Mine Music	Jack Kinney
8	Fun and Fancy Free	Jack Kinney
9	Melody Time	Clyde Geronimi
10	The Adventures of Ichabod and Mr. Toad	Jack Kinney
11	Cinderella	Wilfred Jackson
12	Alice in Wonderland	Clyde Geronimi
13	Peter Pan	Hamilton Luske
14	Lady and the Tramp	Hamilton Luske
15	Sleeping Beauty	Clyde Geronimi
16	101 Dalmatians	Wolfgang Reitherman
17	The Sword in the Stone	Wolfgang Reitherman
18	The Jungle Book	Wolfgang Reitherman
19	The Aristocats	Wolfgang Reitherman
20	Robin Hood	Wolfgang Reitherman
21	The Many Adventures of Winnie the Pooh	Wolfgang Reitherman
22	The Rescuers	Wolfgang Reitherman
23	The Fox and the Hound	Art Stevens
24	The Black Cauldron	Ted Berman
25	The Great Mouse Detective	Ron Clements
26	Oliver & Company	George Scribner
27	The Little Mermaid	Ron Clements
28	The Rescuers Down Under	Mike Gabriel
29	Beauty and the Beast	Gary Trousdale
30	Aladdin	Ron Clements
31	The Lion King	Roger Allers
32	Pocahontas	Mike Gabriel
33	The Hunchback of Notre Dame	Gary Trousdale
34	Hercules	Ron Clements
35	Mulan	Barry Cook
36	Tarzan	Chris Buck
37	Fantasia 2000	full credits

38	Dinosaur	Ralph Zondag
39	The Emperor's New Groove	Mark Dindal
40	Atlantis: The Lost Empire	Gary Trousdale
41	Lilo & Stitch	Chris Sanders
42	Treasure Planet	Ron Clements
43	Brother Bear	Robert Walker
44	Home on the Range	Will Finn
45	Chicken Little	Mark Dindal
46	Meet the Robinsons	Stephen J. Anderson
47	Bolt	Chris Williams
48	The Princess and the Frog	Ron Clements
49	Tangled	Nathan Greno
50	Winnie the Pooh	Stephen J. Anderson
51	Wreck-It Ralph	Rich Moore
52	Frozen	Chris Buck
53	Big Hero 6	Don Hall
54	Zootopia	Byron Howard
55	Moana	Ron Clements

Ok, let's merge it with the **gross** dataframe and we will only focus on the movies listed in **directors** since we want the animation directors.

```
[15]: directors_gross_merged = pd.merge(gross, directors, on="movie_title")
directors_gross_merged
```

```
[15]:
```

	movie_title	release_date	genre \
0	Snow White and the Seven Dwarfs	Dec 21, 1937	Musical
1	Pinocchio	Feb 9, 1940	Adventure
2	Fantasia	Nov 13, 1940	Musical
3	Cinderella	Feb 15, 1950	Drama
4	Cinderella	Mar 13, 2015	Drama
5	Lady and the Tramp	Jun 22, 1955	Drama
6	Sleeping Beauty	Jan 29, 1959	Drama
7	101 Dalmatians	Jan 25, 1961	Comedy
8	101 Dalmatians	Nov 27, 1996	Comedy
9	The Sword in the Stone	Dec 25, 1963	Adventure
10	The Jungle Book	Oct 18, 1967	Musical
11	The Jungle Book	Dec 25, 1994	Adventure
12	The Jungle Book	Apr 15, 2016	Adventure
13	The Aristocats	Apr 24, 1970	Musical
14	The Many Adventures of Winnie the Pooh	Mar 11, 1977	NaN
15	The Rescuers	Jun 22, 1977	Adventure
16	The Fox and the Hound	Jul 10, 1981	Comedy
17	The Black Cauldron	Jul 24, 1985	Adventure
18	The Great Mouse Detective	Jul 2, 1986	Adventure
19	Oliver & Company	Nov 18, 1988	Adventure
20	The Little Mermaid	Nov 15, 1989	Adventure

21	The Rescuers Down Under	Nov 16, 1990	Adventure
22	Beauty and the Beast	Nov 13, 1991	Musical
23	Aladdin	Nov 11, 1992	Comedy
24	The Lion King	Jun 15, 1994	Adventure
25	Pocahontas	Jun 10, 1995	Adventure
26	The Hunchback of Notre Dame	Jun 21, 1996	Adventure
27	Hercules	Jun 13, 1997	Adventure
28	Mulan	Jun 19, 1998	Adventure
29	Tarzan	Jun 16, 1999	Adventure
30	Dinosaur	May 19, 2000	Adventure
31	The Emperor's New Groove	Dec 15, 2000	Adventure
32	Atlantis: The Lost Empire	Jun 8, 2001	Adventure
33	Lilo & Stitch	Jun 21, 2002	Adventure
34	Treasure Planet	Nov 27, 2002	Adventure
35	Brother Bear	Oct 24, 2003	Adventure
36	Home on the Range	Apr 2, 2004	Comedy
37	Chicken Little	Nov 4, 2005	Adventure
38	Meet the Robinsons	Mar 30, 2007	Adventure
39	Bolt	Nov 21, 2008	Comedy
40	The Princess and the Frog	Nov 25, 2009	Adventure
41	Alice in Wonderland	Mar 5, 2010	Adventure
42	Tangled	Nov 24, 2010	Adventure
43	Winnie the Pooh	Jul 15, 2011	Adventure
44	Wreck-It Ralph	Nov 2, 2012	Adventure
45	Frozen	Nov 22, 2013	Adventure
46	Big Hero 6	Nov 7, 2014	Adventure
47	Zootopia	Mar 4, 2016	Adventure
48	Moana	Nov 23, 2016	Adventure

	MPAA_rating	total_gross	inflation_adjusted_gross	director
0	G	184925485.0	5.228953e+09	David Hand
1	G	84300000.0	2.188229e+09	Ben Sharpsteen
2	G	83320000.0	2.187091e+09	full credits
3	G	85000000.0	9.206087e+08	Wilfred Jackson
4	PG	201151353.0	2.011514e+08	Wilfred Jackson
5	G	93600000.0	1.236036e+09	Hamilton Luske
6	NaN	9464608.0	2.150583e+07	Clyde Geronimi
7	G	153000000.0	1.362871e+09	Wolfgang Reitherman
8	G	136189294.0	2.587289e+08	Wolfgang Reitherman
9	NaN	22182353.0	1.538708e+08	Wolfgang Reitherman
10	Not Rated	141843000.0	7.896123e+08	Wolfgang Reitherman
11	PG	44342956.0	8.893032e+07	Wolfgang Reitherman
12	PG	364001123.0	3.640011e+08	Wolfgang Reitherman
13	G	55675257.0	2.551615e+08	Wolfgang Reitherman
14	NaN	0.0	0.000000e+00	Wolfgang Reitherman
15	NaN	48775599.0	1.597439e+08	Wolfgang Reitherman
16	NaN	43899231.0	1.331189e+08	Art Stevens

17	NaN	21288692.0	5.055314e+07	Ted Berman
18	NaN	23605534.0	5.363737e+07	Ron Clements
19	G	49576671.0	1.022545e+08	George Scribner
20	G	111543479.0	2.237260e+08	Ron Clements
21	G	27931461.0	5.579673e+07	Mike Gabriel
22	G	218951625.0	3.630177e+08	Gary Trousdale
23	G	217350219.0	4.419692e+08	Ron Clements
24	G	422780140.0	7.616409e+08	Roger Allers
25	G	141579773.0	2.743710e+08	Mike Gabriel
26	G	100138851.0	1.909888e+08	Gary Trousdale
27	G	99112101.0	1.820294e+08	Ron Clements
28	G	120620254.0	2.168078e+08	Barry Cook
29	G	171091819.0	2.839003e+08	Chris Buck
30	PG	137748063.0	2.154390e+08	Ralph Zondag
31	G	89296573.0	1.367893e+08	Mark Dindal
32	PG	84052762.0	1.251881e+08	Gary Trousdale
33	PG	145771527.0	2.115067e+08	Chris Sanders
34	PG	38120554.0	5.518914e+07	Ron Clements
35	G	85336277.0	1.192183e+08	Robert Walker
36	PG	50026353.0	6.791017e+07	Will Finn
37	G	135386665.0	1.779547e+08	Mark Dindal
38	G	97822171.0	1.198606e+08	Stephen J. Anderson
39	PG	114053759.0	1.337025e+08	Chris Williams
40	G	104400899.0	1.163165e+08	Ron Clements
41	PG	334191110.0	3.570635e+08	Clyde Geronimi
42	PG	200821936.0	2.143885e+08	Nathan Greno
43	G	26692846.0	2.837587e+07	Stephen J. Anderson
44	PG	189412677.0	2.003550e+08	Rich Moore
45	PG	400738009.0	4.149972e+08	Chris Buck
46	PG	222527828.0	2.292492e+08	Don Hall
47	PG	341268248.0	3.412682e+08	Byron Howard
48	PG	246082029.0	2.460820e+08	Ron Clements

We can see some duplicated rows in “movie_title” like “Cinderella”, “The Jungle Book”, and “101 Dalmatians”. When we check the release date, they are all different. The first one of these movies is animation, the rest are all real-actor movies filmed in recent years. We will need to delete them in case it affects our results.

```
[16]: # Drop rows with dupliacted movie_title
directors_gross_merged = directors_gross_merged.
↳drop_duplicates(subset="movie_title")

# Reset the index to get the right order
directors_gross_merged = directors_gross_merged.reset_index(drop=True)
```

Finally, let’s group the directors and sum up the revenue of all the movies directed by every director to find who creates the most revenue. Here, we can use our custom sort function to save time again!

```
[17]: # run custom sort on the data
directors_gross_merged_group = ps.
    ↳ custom_sort(directors_gross_merged, 'director', 'inflation_adjusted_gross')
directors_gross_merged_group
```

```
[17]:
```

	director	inflation_adjusted_gross
0	David Hand	5.228953e+09
1	Wolfgang Reitherman	2.721260e+09
2	Ben Sharpsteen	2.188229e+09
3	full credits	2.187091e+09
4	Ron Clements	1.318950e+09
5	Hamilton Luske	1.236036e+09
6	Wilfred Jackson	9.206087e+08
7	Roger Allers	7.616409e+08
8	Chris Buck	6.988974e+08
9	Gary Trousdale	6.791946e+08
10	Clyde Geronimi	3.785693e+08
11	Byron Howard	3.412682e+08
12	Mike Gabriel	3.301677e+08
13	Mark Dindal	3.147439e+08
14	Don Hall	2.292492e+08
15	Barry Cook	2.168078e+08
16	Ralph Zondag	2.154390e+08
17	Nathan Greno	2.143885e+08
18	Chris Sanders	2.115067e+08
19	Rich Moore	2.003550e+08
20	Stephen J. Anderson	1.482365e+08
21	Chris Williams	1.337025e+08
22	Art Stevens	1.331189e+08
23	Robert Walker	1.192183e+08
24	George Scribner	1.022545e+08
25	Will Finn	6.791017e+07
26	Ted Berman	5.055314e+07

It seems that in fact it is **David Hand** is the director who creates the most revenue. Our expectation **Ron Clements** is the forth if we ignore the “all credits” one.

```
[18]: # Visualize the gross created by the directors using a bar plot.
directors_gross_plot = (
    alt.Chart(directors_gross_merged_group, width=500, height=300)
    .mark_bar()
    .encode(
        x=alt.X("director:N", title="Disney Directors", sort="-y"),
        y=alt.Y("inflation_adjusted_gross:Q", title="Gross Revenue of their_
    ↳ Animations(adjusted up to 2016)"),
    )
```

```
.properties(title="Total Gross Revenue of Disney Animations Created by  
↪Every Director")  
)  
directors_gross_plot
```

```
[18]: alt.Chart(...)
```

5 Discussions

In this project, I analyzed the Disney dataset and tried to find which director creates the most revenue. Before answering this question, I did some exploratory data analysis to see which movies gain the most revenue, how is the total revenue of Disney movies changes over the year. Generally, there is an increasing trend in the total number since 1980. However, the animations released in 1930 – 1960 are so popular and gorgeous and win the top revenue. That nearly determined the result because the top three directors are all active in 1930 – 1960. The top first, **David Hand** is the director of “[Snow White and the Seven Dwarfs](#)” and “[Bambi](#)”. The second, **Wolfgang Reitherman**, I have introduced that he directed “[101 Dalmatians](#)”, “[The Jungle Books](#)” etc. The third, **Ben Sharpsten**, is the director of “[Pinocchio](#)” and “[Dumbo](#)”.

My assumption **Ron Clements** is still the director who creates the most revenue in the recent 40 years since 1980. If we ignore the effects of the inflation-adjusted rates, he can indeed win the first price. However, it will be unfair to compare the revenue without the adjustment. Let’s still follow the inflation adjust gross in this project.

In this project, I did a lot of research on animations, and they bring my memory back to when I was a kid. Regardless of the inflation rate, one of the reasons that the animations in 1930 – 1960 got that much achievement is because people at that time were longing for pure and childish happiness. Those directors, in a way, create some lovely dreams for both children and adults.

Another question that could be looked at given this dataset is to find the most welcome genre movies. One could look at which genre has created the most revenue over the years. This is interesting because Disney has produced different types of movies. It is meaningful for us to have some ideas about which genre is the most popular one. We can also see how the audience’s taste changes over the years. Will the audience in 70 years ago like a different kind of movies with us. Or are people the same? This might also influent Disney’s filming decision.

6 References

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 solely used for illustration purposes. Let’s give to **Ceasar** what belongs to **Ceasar**.

6.1 Resources used

- [Data Source](#)
 - This Disney database used in this work was curated by **Kelly Garrett**.
- [Data Visualization & Question Of Interest](#)

- Inspiration for generating the plotting the total revenue over the years and consider the inflation rate influence was taken from **Avinash Reddy Munnangi**.
- [WikiPedia](#)
 - Introductions of the animations and directors I mentioned.
- [IMDB](#)
 - Inspiration of the achievement of animators in the 30s-60s.

[]: