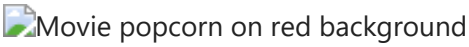


Netflix! What started in 1997 as a DVD rental service has since exploded into one of the largest entertainment and media companies.

Given the large number of movies and series available on the platform, it is a perfect opportunity to flex your exploratory data analysis skills and dive into the entertainment industry. Our friend has also been brushing up on their Python skills and has taken a first crack at a CSV file containing Netflix data. They believe that the average duration of movies has been declining. Using your friends initial research, you'll delve into the Netflix data to see if you can determine whether movie lengths are actually getting shorter and explain some of the contributing factors, if any.

You have been supplied with the dataset `netflix_data.csv`, along with the following table detailing the column names and descriptions:

The data

netflix_data.csv

Column	Description
`show_id`	The ID of the show
`type`	Type of show
`title`	Title of the show
`director`	Director of the show
`cast`	Cast of the show
`country`	Country of origin
`date_added`	Date added to Netflix
`release_year`	Year of Netflix release
`duration`	Duration of the show in minutes
`description`	Description of the show
`genre`	Show genre

In [71]:
```
# Importing pandas and matplotlib
import pandas as pd
import matplotlib.pyplot as plt
```

Let's load the CSV file and rename it

In [72]:
```
# read file using read_csv and renaming dataframe
netflix_df = pd.read_csv('netflix_data.csv')
```

Lets familiarize ourselves with the data first

In [73]:
```
# view first 10 rows
netflix_df.head()
```

Out[73]:

	show_id	type	title	director	cast	country	date_added	release_year	duration
0	s1	TV Show	3%	NaN	João Miguel, Bianca Comparato, Michel Gomes, R...	Brazil	August 14, 2020	2020	4
1	s2	Movie	7:19	Jorge Michel Grau	Demián Bichir, Héctor Bonilla, Oscar Serrano, ...	Mexico	December 23, 2016	2016	93
2	s3	Movie	23:59	Gilbert Chan	Tedd Chan, Stella Chung, Henley Hii, Lawrence ...	Singapore	December 20, 2018	2011	78
3	s4	Movie	9	Shane Acker	Elijah Wood, John C. Reilly, Jennifer Connelly...	United States	November 16, 2017	2009	80
4	s5	Movie	21	Robert Luketic	Jim Sturgess, Kevin Spacey, Kate Bosworth, Aar...	United States	January 1, 2020	2008	123

In [74]:
```
# some general information about our data
netflix_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7787 entries, 0 to 7786
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       7787 non-null   object
 1   type          7787 non-null   object
 2   title         7787 non-null   object
 3   director      5398 non-null   object
 4   cast          7069 non-null   object
 5   country       7280 non-null   object
 6   date_added    7777 non-null   object
 7   release_year  7787 non-null   int64 
 8   duration      7787 non-null   int64 
 9   description   7787 non-null   object
 10  genre         7787 non-null   object
dtypes: int64(2), object(9)
memory usage: 669.3+ KB
```

Now since we're researching only Movies, let us filter out the TV shows

In [75]:
```
# remove TV shows and save as netflix_subset
netflix_subset = netflix_df[netflix_df['type'] == 'Movie']
```

Now let's check the last 10 rows just for confirmation

In [76]:
```
# check if our filter did what we expected it to do
netflix_subset.tail()
```

Out[76]:

	show_id	type	title	director	cast	country	date_added	release_year	durati
7781	s7782	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	
7782	s7783	Movie	Zozo	Josef Fares	Imad Creidi, Antoinette Turk, Elias Gergi, Car...	Sweden	October 19, 2020	2005	
7783	s7784	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	1
7784	s7785	Movie	Zulu Man in Japan	NaN	Nasty C	NaN	September 25, 2020	2019	
7786	s7787	Movie	ZZ TOP: THAT LITTLE OL' BAND FROM TEXAS	Sam Dunn	NaN	United Kingdom	March 1, 2020	2019	

It looks good! our table still has a lot of information so we are going to trim it down to leave 5 columns "title", "country", "genre", "release_year", "duration", and save this into a new DataFrame called netflix_movies.

In [77]:
```
#smaller table

netflix_movies = netflix_subset[["title", "country", "genre", "release_year", "d
```

Now let us check to confirm the accuracy of this code..

In [78]:
```
# display first 5 rows

netflix_movies.head(5)
```

Out[78]:

	title	country	genre	release_year	duration
1	7:19	Mexico	Dramas	2016	93
2	23:59	Singapore	Horror Movies	2011	78
3	9	United States	Action	2009	80
4	21	United States	Dramas	2008	123
6	122	Egypt	Horror Movies	2019	95

It looks good! Now we need to just filter netflix_movies to find the movies that are shorter than 60 minutes, saving the resulting DataFrame as short_movies

In [79]:
```
#create a dataframe short_movies

short_movies = netflix_movies[netflix_movies["duration"] < 60]
```

let's inspect them..

In [80]:
```
#checking first 10 rows

short_movies.head()
```

Out[80]:

	title	country	genre	release_year	duration
35	#Rucker50	United States	Documentaries	2016	56
55	100 Things to do Before High School	United States	Uncategorized	2014	44
67	13TH: A Conversation with Oprah Winfrey & Ava ...	NaN	Uncategorized	2017	37
101	3 Seconds Divorce	Canada	Documentaries	2018	53
146	A 3 Minute Hug	Mexico	Documentaries	2019	28

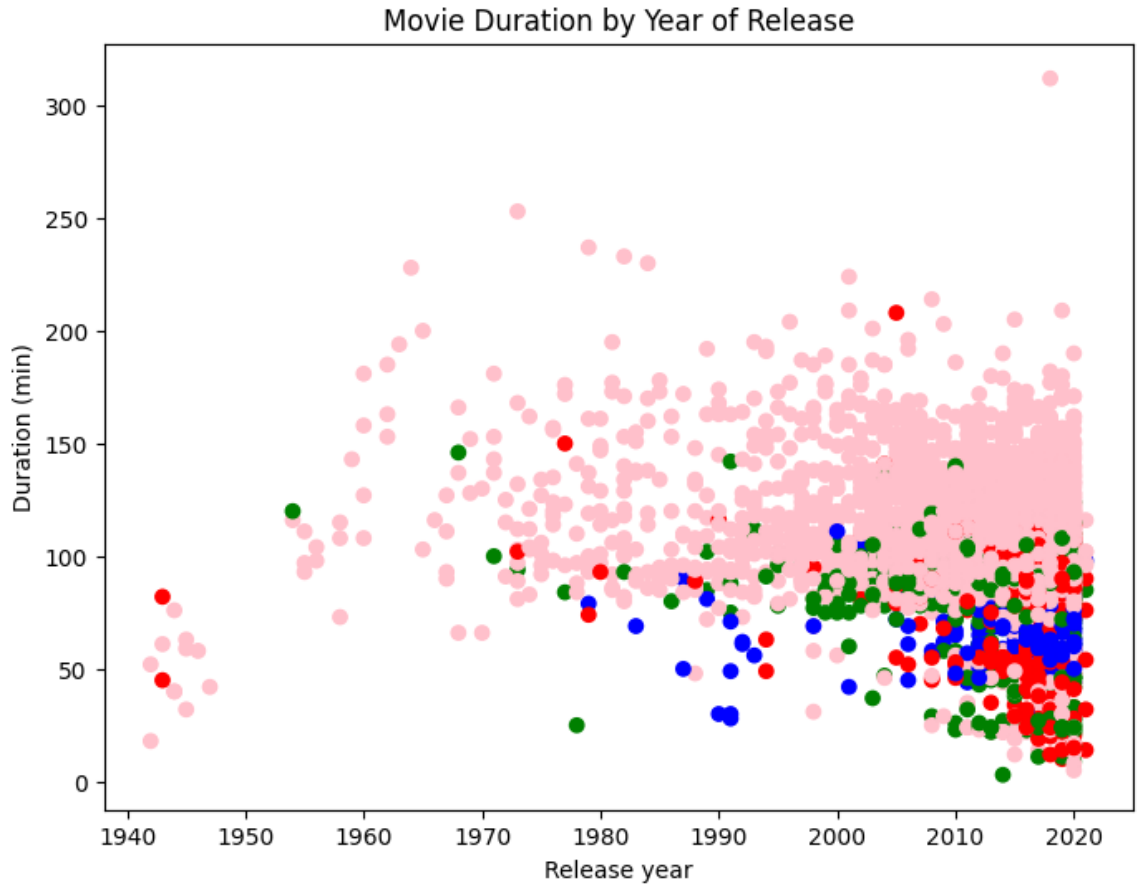
great! there are about 420 of these entries using the count() function. what could be causing this? Let us make a graph then.

In [81]:
```
# Initial empty list
colors = []

# For loop based on genres
for lab, row in netflix_movies.iterrows():
    if row['genre'] == "Children":
        colors.append('green')
    elif row['genre'] == "Documentaries":
        colors.append('red')
    elif row['genre'] == "Stand-Up":
        colors.append('blue')
    else:
```

```
        colors.append('pink')

# Plot
fig = plt.figure(figsize=(8, 6))  # Set the figure size
plt.scatter(netflix_movies['release_year'], netflix_movies['duration'], c=colors
plt.xlabel("Release year")
plt.ylabel("Duration (min)")
plt.title("Movie Duration by Year of Release")
plt.show()
```



Are we certain that movies are getting shorter?

No, we definitely need additional analysis

In [82]:
```
answer = "maybe"
```