NETFLIX

GROUP PROJECT

Mohammed Fardeen Khan	[20181CSE0429]
Mohammed Furqan Ahmed	[20181CSE0430]
Meher Taj	[20181CSE0415]
Mohamed Fahad Idrees	[20181CSE0422]
Mohammed Abdul Wahab Ahmed	[20181CSE0425]

INTRODUCTION

Data Analysis of "Netflix Movies & TV Shows" dataset using Python. The purpose of this project is to find out and visualize the data's main characteristics and trends using statistical methods and data visualization techniques.

Netflix was founded on August 29, 1997, as a mail-based rental business. In January 2007, the company launched a streaming media service, introducing video on demand via the Internet.

DATA PREPARATION

Importing Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
!pip install pywaffle
from pywaffle import Waffle
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

Loading the dataset with Pandas

```
netflix = pd.read_csv('netflix_titles.csv')
netflix_1 = pd.read_csv('netflix_titles.csv')
netflix
```

Column name and types

```
netflix.info()
```

PRE-PROCESSING

Checking for missing data

```
missing_data = netflix.isna().sum().sort_values(ascending=False)
missing_data

netflix_isna = pd.isna(netflix['director'])
netflix[netflix_isna]
```

Checking for duplicates

```
netflix['show_id'].duplicated().any()
```

Changing the date format of the column 'date_added' to 'datetime'

```
netflix['date_added']= pd.to_datetime(netflix['date_added'].str.strip(), format= "
%B %d, %Y")
netflix
```

Preparing Movie Rating variables

```
netflix_titles = pd.read_csv("netflix_titles.csv")
# preprocessing
sns.set_style('whitegrid') # plot with grid

movie = netflix_titles[netflix_titles['type'] == 'Movie']
rating_order =  ['G', 'TV-Y', 'TV-G', 'PG', 'TV-Y7', 'TV-Y7-FV', 'TV-PG', 'PG-
13', 'TV-14', 'R', 'NC-17', 'TV-MA']
movie_rating = movie['rating'].value_counts()[rating_order]
```

DATA ANALYSIS

Top 10 Rows from above

```
netflix.head(10)
```

Top 10 Rows from below

```
netflix.tail(10)
```

Number of Rows and Columns

```
netflix.shape
```

Describing the whole dataset

```
netflix.describe()
```

Number of movies/tv-shows added to the streaming platform by Year

```
netflix_release_year = netflix.date_added.dt.year.astype('Int64').value_counts()
netflix_release_year
```

The month with the most added movies/tv-shows

```
netflix_release_month = netflix.date_added.dt.month.astype('Int64').value_counts()
netflix_release_month
```

Day with the most added movies/tv-shows

```
netflix_release_day = netflix.date_added.dt.day.astype('Int64').value_counts()
netflix_release_day
```

About the Movies/TV-Shows

Number of Movies and TV-Shows

```
# create variables to count
mov=0
tv=0

# create a Dataframe
df = netflix_1

# loop for counting the unique
# values in type

for i in range(0, len(df['type'])):
  if "Movie" in df['type'][i]:
    mov += 1
print("No. of Movie values :",mov)

for i in range(0, len(df['type'])):
  if "TV Show" in df['type'][i]:
    tv += 1
print("No. of TV shows values :",tv)
```

Number of Movies vs. TV-Shows

```
netflix_type = netflix.type.value_counts()
netflix_type
```

The year with the most releases movies/tv-shows

```
movietv_release_year = netflix.release_year.value_counts()
movietv_release_year
```

The oldest movie/tv-show on streaming

```
netflix[netflix['release_year']== 1925]
```

Dropping 'NA' Records from the Column

```
#counting the number of 'NA' on the column 'date_added'
netflix['date_added'].isna().sum()
#dropping 'NA'
netflix = netflix.dropna(subset=['date_added'])
```

Top 10 Countries producing the most movies/tv-shows

```
country_count = netflix.copy()
country_count = pd.concat([country_count, netflix['country'].str.split(",", expand
=True)], axis=1)
country_count = country_count.melt(id_vars=["type","title"], value_vars=range(12),
 value_name="Country")
country_count = country_count[country_count["Country"].notna()]
country_count["Country"] = country_count["Country"].str.strip()
country_count

country_count.Country.unique()[:10]
```

Countries with the most number of content streaming

```
country_count.Country.value_counts()
```

Top 10 Cast members with the most content

```
cast_count = netflix.copy()
cast_count = pd.concat([cast_count, netflix['cast'].str.split(",", expand=True)],
axis=1)
cast_count = cast_count.melt(id_vars=["type","title"], value_vars=range(44), value
_name="Cast_name")
cast_count = cast_count[cast_count["Cast_name"].notna()]
cast_count["Cast_name"] = cast_count["Cast_name"].str.strip()
cast_count


cast_count.Cast_name.value_counts()[:10]
cast_count1=cast_count
cast_count1
```

Converting Pandas Series to Dictionary

```
pd.unique(cast_count.Cast_name)
sr = cast_count.Cast_name.value_counts()[:10]
print("Top 10 Cast in Series:\n", sr,"\n\n")

# convert to dictionary
n=sr.to_dict()
print("Coverted to Dictionary:\n", n)
```

Importing new Subplots library

```
import plotly.graph_objects as go
from plotly.subplots import make_subplots

a = cast_count.Cast_name.value_counts()[:10]
n = dict(list(n.items())[:10])
y = []
for i in range(0, len(a)):
  y.append(a[i])
u = sr.keys()

print("Top 10 Cast in Series:\n",a, "\n\n")
print("Top 10 Cast in List:\n",y, "\n\n")
print("Top 10 Cast in Dictionary:\n",n, "\n\n")
print("Top 10 Cast Dict(keys):\n",u)
```

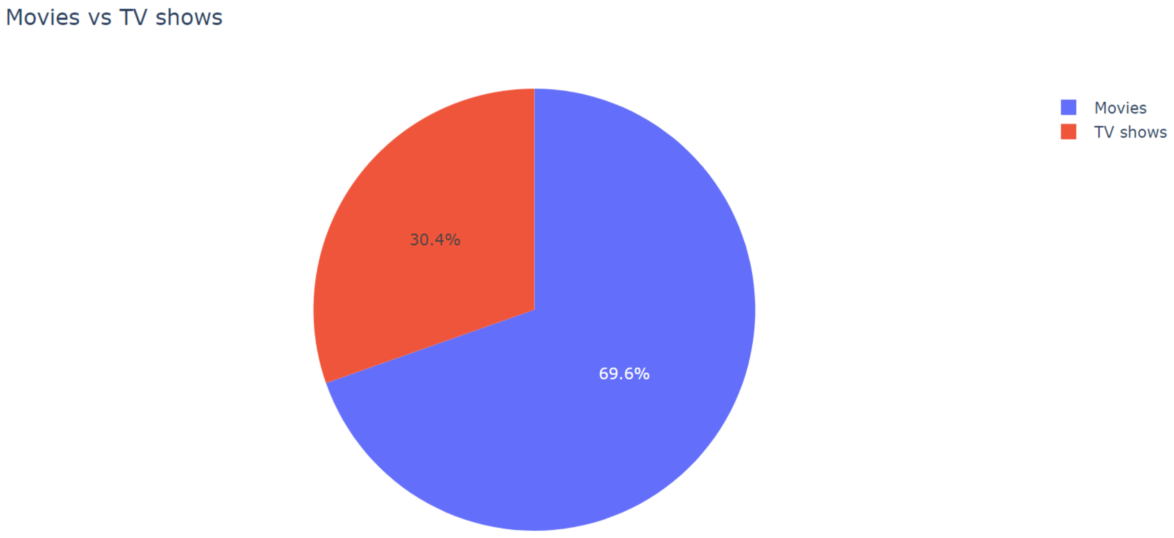
DATA VISUALIZATION

PIE CHART

One–Punch Man (Japanese: ワンパンマン
, Hepburn: Wanpanman) is a Japanese superhero franchise created by the artist ONE.

Movies vs TV shows

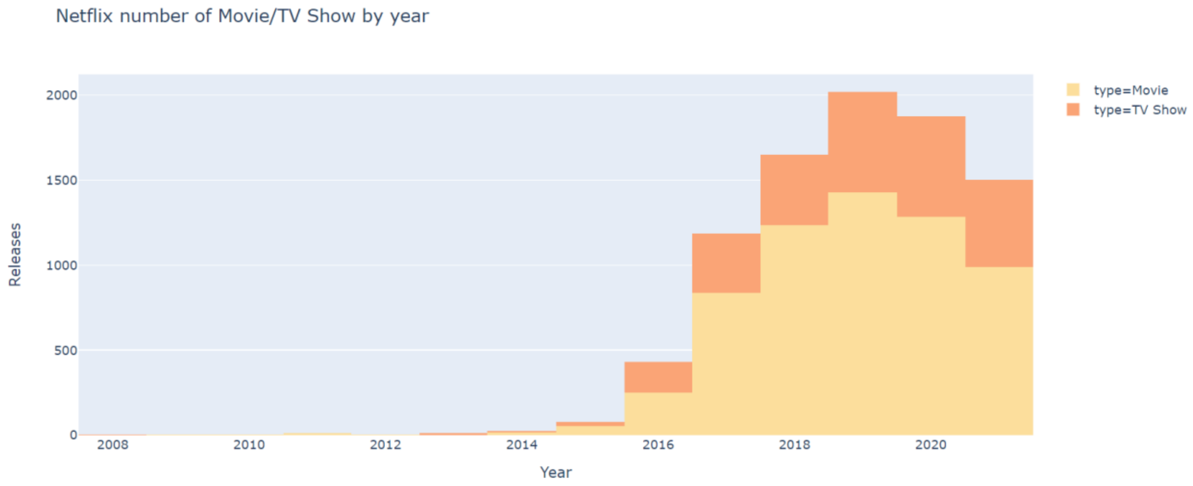
```
types=["Movies", "TV shows"]
fig = px.pie(netflix_type,
             values='type',
             names=types,
             title='Movies vs TV shows')
fig.show()
```



HISTOGRAMS

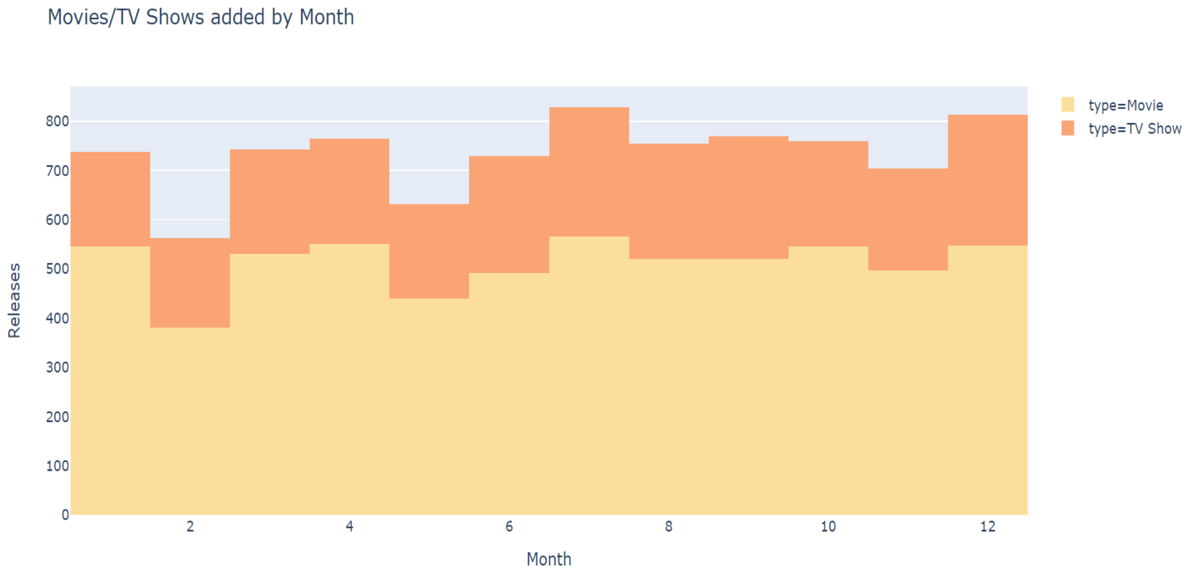
Number of Movies/TV-Shows added by Year (Netflix release)

```
fig=px.histogram(netflix,
                 x= netflix['date_added'].dt.year,
                 color= netflix['type'],
                 title="Netflix number of Movie/TV Show by year",
                 color_discrete_sequence= px.colors.sequential.Sunsetdark)
fig.update_layout(xaxis_title_text="Year",
                  yaxis_title_text="Releases")
fig.show()
```



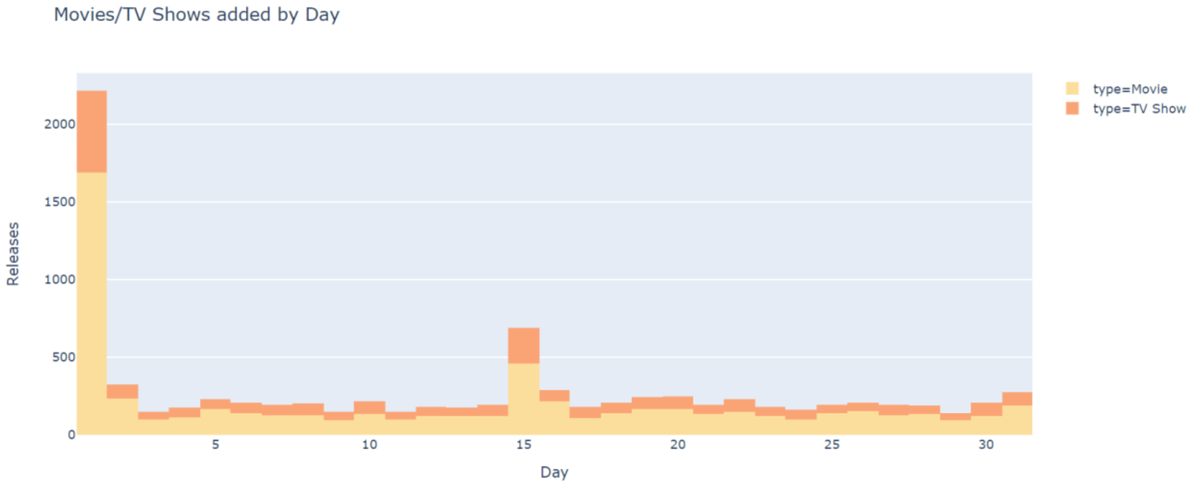
Number of Movies/TV-Shows added by Month

```
fig=px.histogram(netflix,
                 x= netflix['date_added'].dt.month,
                 color= netflix['type'],
                 color_discrete_sequence= px.colors.sequential.Sunsetdark,
                 title="Movies/TV Shows added by Month")
fig.update_layout(xaxis_title_text="Month",
                  yaxis_title_text="Releases")
fig.show()
```



Number of Movies/TV-Shows added by Day

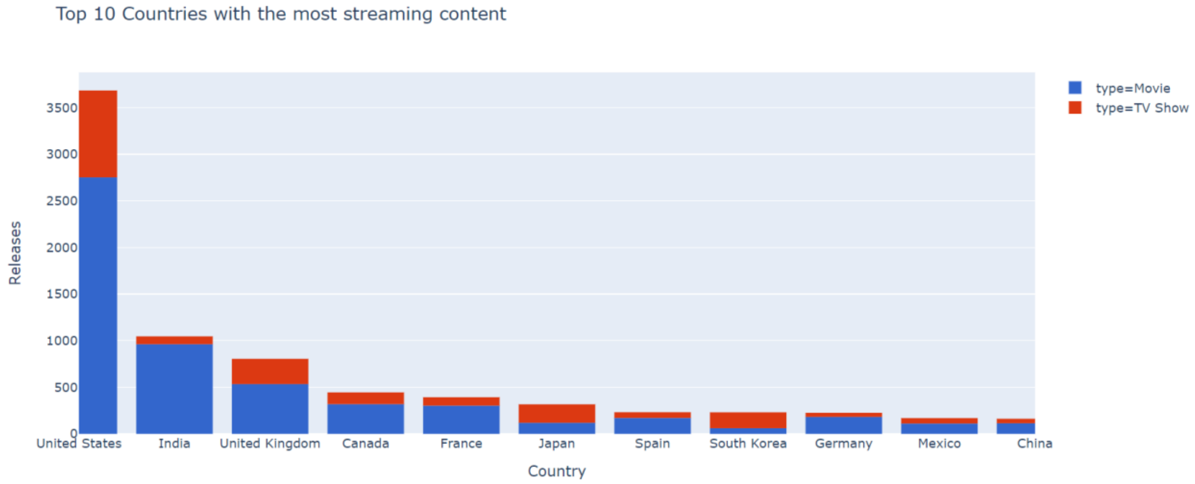
```
fig=px.histogram(netflix,
                 x= netflix['date_added'].dt.day,
                 color= netflix['type'],
                 color_discrete_sequence= px.colors.sequential.Sunsetdark,
                 title="Movies/TV Shows added by Day")
fig.update_layout(xaxis_title_text="Day",
                  yaxis_title_text="Releases")
fig.show()
```



Top 10 Countries with the most streaming content

```
fig=px.histogram(country_count,
                 x= 'Country',
                 color= 'type',
                 title="Top 10 Countries with the most streaming content",
                 color_discrete_sequence= px.colors.qualitative.G10).update_xaxes(
categoryorder="total descending",range=(0, 10))

fig.update_layout(xaxis_title_text="Country",
                  yaxis_title_text="Releases")
fig.show()
```

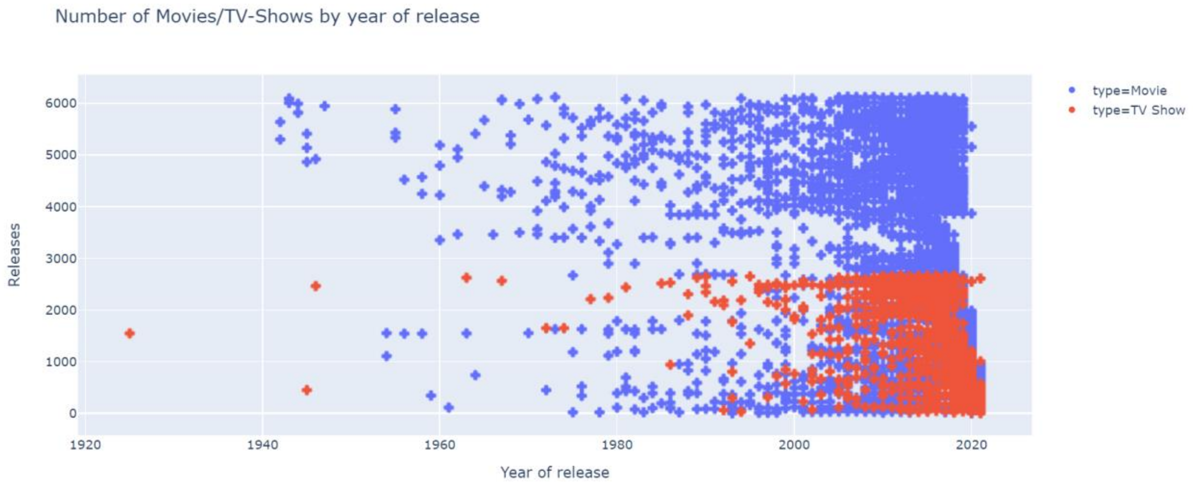


SCATTER PLOT

Squid Game (Korean: 오징어 게임; RR: Ojing-eo Geim) is a South Korean survival drama streaming television series created by Hwang Dong-hyuk for Netflix.

Number of Movies/TV-Shows by year of release (Global release)

```
fig = px.scatter(netflix_1,
                 x="release_year",
                 color="type",
                 title="Number of Movies/TV-Shows by year of release",
                 hover_data=['type'],
                 error_x="type",
                 error_y="type")
fig.update_layout(xaxis_title_text="Year of release",
                  yaxis_title_text="Releases")
fig.show()
```



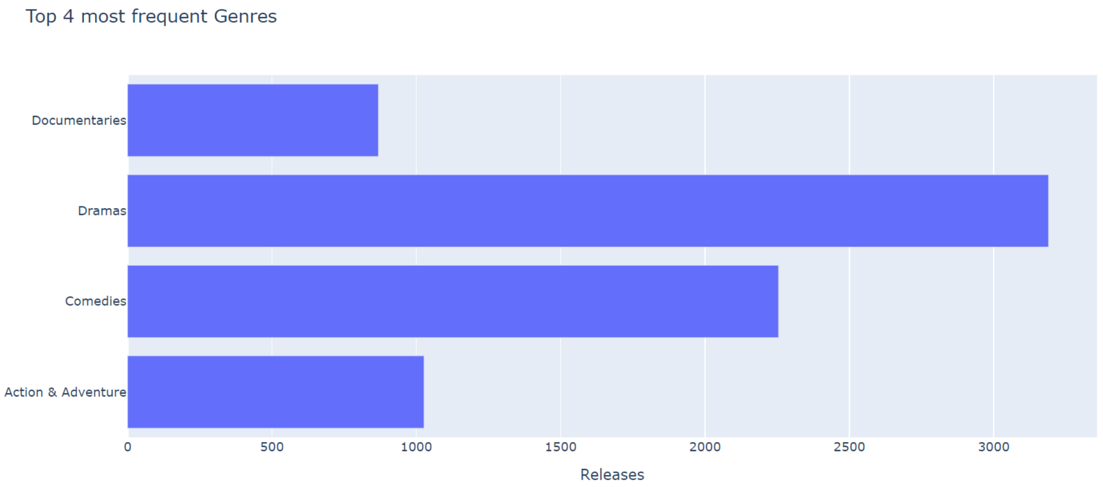
BAR CHART

The Witcher
is a fantasy drama streaming television series created by Lauren Schmidt Hissrich, based on the book series of the same name by Polish writer Andrzej Sapkowski.

Top 4 most frequent Genres

```
aa, co, dr, do = 0,0,0,0
df = netflix_1
for i in range(0, len(df['listed_in'])):
  if "Action & Adventure" in df['listed_in'][i]:
    aa += 1
print("No. of Action & Adventure values :",aa)
for i in range(0, len(df['listed_in'])):
  if "Comedies" in df['listed_in'][i]:
    co += 1
print("No. of Comedies values :",co)
for i in range(0, len(df['listed_in'])):
  if "Dramas" in df['listed_in'][i]:
    dr += 1
print("No. of Dramas values :",dr)
for i in range(0, len(df['listed_in'])):
  if "Documentaries" in df['listed_in'][i]:
    do += 1
print("No. of Documentaries values :",do)
netflix_listed = {"Action & Adventure":aa, "Comedies":co, "Dramas":dr, "Documentaries":do}
df=netflix_listed
names = list(df.keys())
values = list(df.values())
fig=px.bar(range(len(df)),
           values,
           names,
           orientation='h',
           title='Top 4 most frequent Genres')
fig.update_layout(xaxis_title_text="Releases",
                  yaxis_title_text="")
fig.show()
```

```
No. of Action & Adventure values : 1027
No. of Comedies values : 2255
No. of Dramas values : 3190
No. of Documentaries values : 869
```

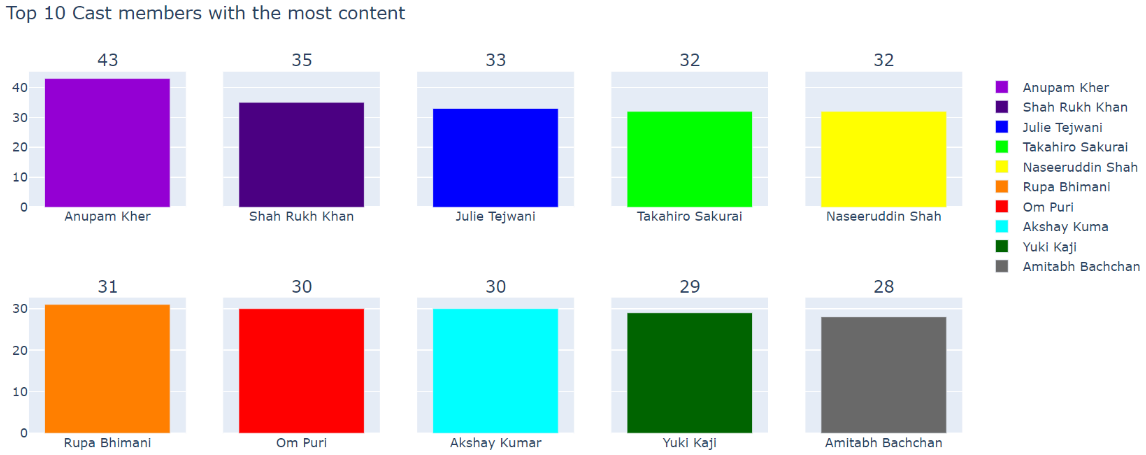


SUBPLOTS

Anupam Kher (born 7 March 1955) is an Indian actor and the former Chairman of Film and Television Institute of India.

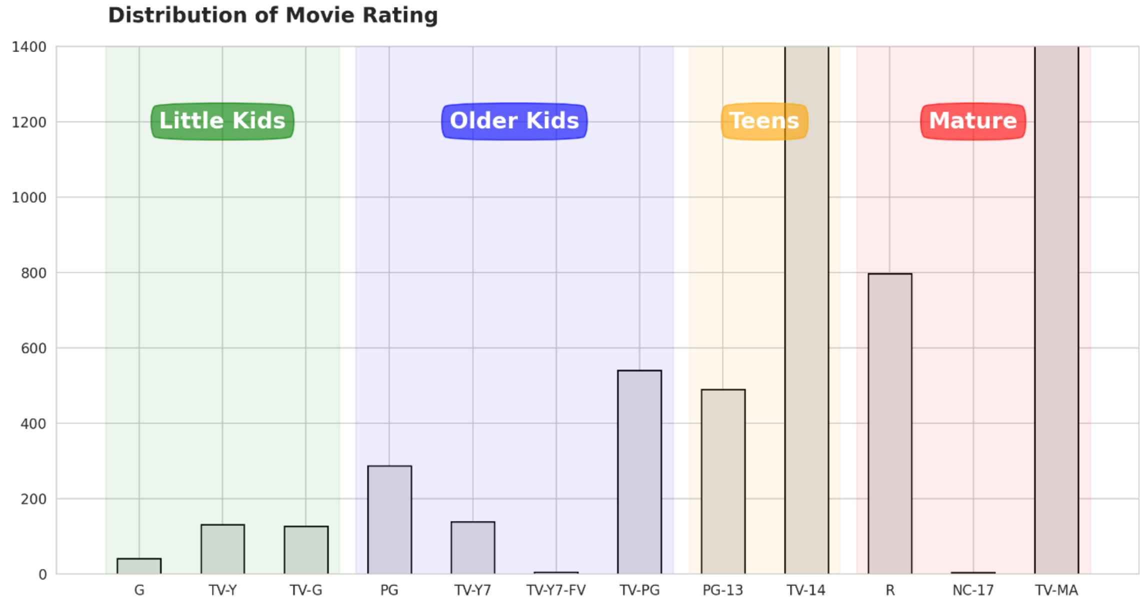
Top 10 Cast members with the most titles

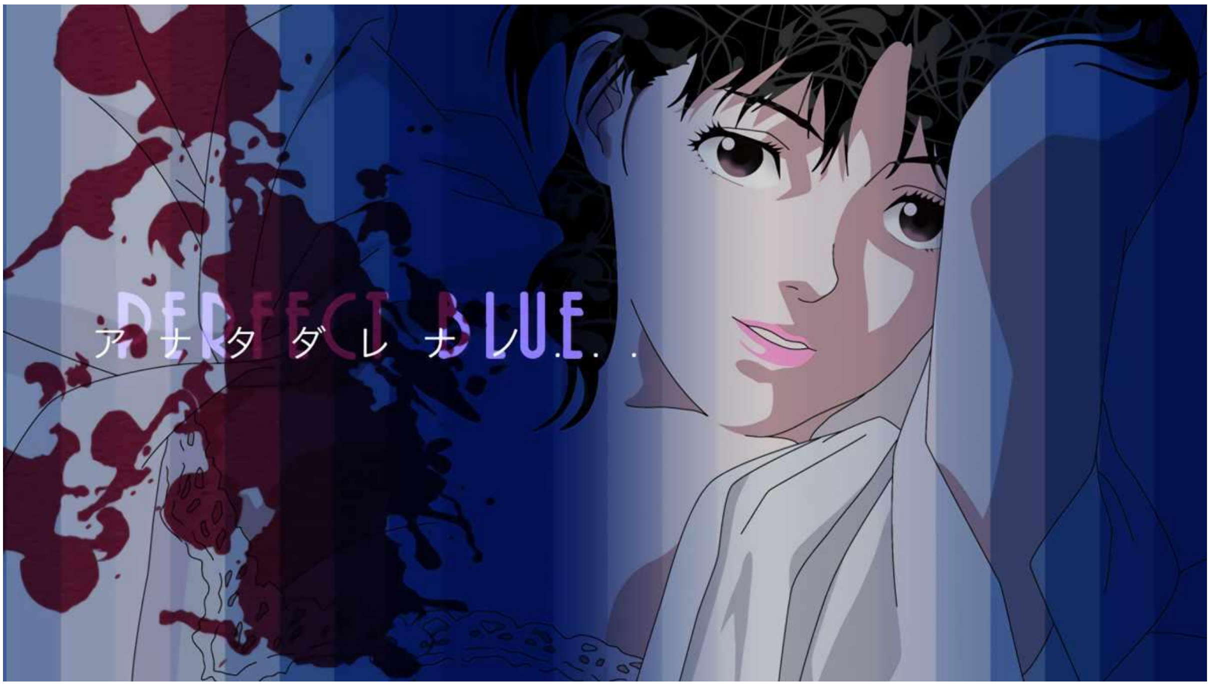
```
fig = make_subplots(rows=2, cols=5, shared_yaxes=True, subplot_titles=("43","35", "33", "32", "32", "31", "30", "30", "29", "28"))
fig.add_trace(go.Bar(x=['Anupam Kher'], y=[43], name="Anupam Kher", marker=dict(color='#9400D3', coloraxis="coloraxis")), 1, 1)
fig.add_trace(go.Bar(x=['Shah Rukh Khan'], y=[35], name="Shah Rukh Khan", marker=dict(color='#4B0082', coloraxis="coloraxis")), 1, 2)
fig.add_trace(go.Bar(x=['Julie Tejwani'], y=[33], name="Julie Tejwani", marker=dict(color='#0000FF', coloraxis="coloraxis")), 1, 3)
fig.add_trace(go.Bar(x=['Takahiro Sakurai'], y=[32], name="Takahiro Sakurai", marker=dict(color='#00FF00', coloraxis="coloraxis")), 1, 4)
fig.add_trace(go.Bar(x=['Naseeruddin Shah'], y=[32], name="Naseeruddin Shah", marker=dict(color='#FFFF00', coloraxis="coloraxis")), 1, 5)
fig.add_trace(go.Bar(x=['Rupa Bhimani'], y=[31], name="Rupa Bhimani", marker=dict(color='#FF7F00', coloraxis="coloraxis")), 2, 1)
fig.add_trace(go.Bar(x=['Om Puri'], y=[30], name="Om Puri", marker=dict(color='#FF0000', coloraxis="coloraxis")), 2, 2)
fig.add_trace(go.Bar(x=['Akshay Kumar'], y=[30], name="Akshay Kuma", marker=dict(color='#00FFFF', coloraxis="coloraxis")), 2, 3)
fig.add_trace(go.Bar(x=['Yuki Kaji'], y=[29], name="Yuki Kaji", marker=dict(color='#006400', coloraxis="coloraxis")), 2, 4)
fig.add_trace(go.Bar(x=['Amitabh Bachchan'], y=[28], name="Amitabh Bachchan", marker=dict(color='#696969', coloraxis="coloraxis")), 2, 5)
fig.update_layout(coloraxis=dict(colorscale='Bluered_r'), title_text="Top 10 Cast members with the most content", showlegend=True)
fig.show()
```



Distribution of Movie Ratings

```
def rating_barplot(data, title, height, h_lim=None):
    fig, ax = plt.subplots(1,1, figsize=(14, 7), dpi=200)
    if h_lim :
        ax.set_ylim(0, h_lim)
    ax.bar(data.index, data,  color="#e0e0e0", width=0.52, edgecolor='black')
    color =  ['green',  'blue',  'orange',  'red']
    span_range = [[0, 2], [3,  6], [7, 8], [9, 11]]
    for idx, sub_title in enumerate(['Little Kids', 'Older Kids', 'Teens', 'Mature
']):
        ax.annotate(sub_title,
                    xy=(sum(span_range[idx])/2 ,height),
                    xytext=(0,0), textcoords='offset points',
                    va="center", ha="center",
                    color="w", fontsize=16, fontweight='bold',
                    bbox=dict(boxstyle='round4', pad=0.4, color=color[idx], alpha=
0.6))
        ax.axvspan(span_range[idx][0]-
0.4,span_range[idx][1]+0.4,  color=color[idx], alpha=0.07)
    ax.set_title(f'Distribution of {title} Rating', fontsize=15, fontweight='bold'
, position=(0.20, 1.0+0.03))
    plt.show()
rating_barplot(movie_rating,'Movie', 1200, 1400)
```





Perfect Blue (Japanese: パーフェクトブルー, Hepburn: Pāfekuto Burū) is a 1997 Japanese animated psychological thriller film directed by Satoshi Kon.

The rating categories given in our dataset are as follows;

1. *Little Kids*

- **G** – G ratings are most notable for what the films don’t include: sex and nudity, substance abuse, or realistic/noncartoon violence.
- **TV-Y** – Content that is suitable for all children ages newborn-6, particularly those of preschool or kindergarten age.
- **TV-G**- Content that is suitable for all audiences.

2. *Older Kids*

- **PG** – Some material *may not be suitable for children*. The movie may have mildly strong language and some violence, but no substance use or physical abuse.
- **TV-Y7** – Content that is suitable for children who are at least *7 years old*.
- **TV-Y7-FV** – Programming rated TV-Y7-FV is recommended for *ages 7 and older*, with the unique advisory that the program contains fantasy violence.
- **TV-PG** – Content with parental guidance suggested.

3. *Teens*

- **PG-13** – Some Material May Be Inappropriate for Children Under 13.
- **TV-14** – Content may be inappropriate for children younger than 14 years of age.

4. *Mature*

- **R(restricted)** – *No one under 17* admitted without an accompanying parent or guardian.
- **NC-17** – Most parents would consider patently too adult for children 17 and under.
- **TV-MA** – Programs with this rating are usually not suitable for anyone under 17 years of age (under 18 in some cases).

VERDICT

About Netflix

- There are more Movies than TV-Shows available on streaming. 6131 movies and 2676 tv-shows.
- 2019 is the year with the most content addition on the streaming platform, 2016 movie/tv-shows added, followed by 2020 with 1879, and 2018 with 1649 total.
- July and December are the months with the most content addition, 827 and 813 movie/tv-shows added.
- Netflix adds content on the first day of the month more than any other day.

About the content

- Among the contents available 1147 of them were originally released in 2018 followed by 2017 with 1032, and 2019 with 1030 total.
- Pioneers: First Women Filmmakers is the oldest content available on streaming. It's a collection of restored films dating from 1925.
- The United States is the country that produces the most of the content with 3690 titles, followed by India 1046 titles and the United Kingdom 806 titles.
- Anupam Kher is the actor with the higher number of titles, 43 films. Anupam Kher is an Indian actor, director, and producer that has appeared in over 500 films.

SOURCE

The Netflix Movies & TV Shows dataset can be found on Kaggle. It contains all TV Shows and Movies metadata available on Netflix. The dataset is updated every month. It contains 8807 records and 12 columns.

https://colab.research.google.com/drive/18wn1ZQ1-bxcTY-5-S4PCas8b4ihTmgi0?usp=sharing