About Netflix

- **Netflix** is one of the world's leading media and video streaming platforms.
- It offers a vast library of **over 10,000 movies and TV shows** across various genres.
- As of mid-2021, Netflix has **222 million subscribers** globally.
- Available in **190+ countries**, with content in multiple languages.
- Known for its award-winning **original content**, including movies, series, and documentaries.
- Provides a personalized experience using **advanced recommendation algorithms**.
- Offers flexible subscription plans, including **ad-supported and ad-free options**.
- Continually expands its library with **new releases and exclusive content**.
- Compatible with a wide range of devices, including **smart TVs, mobile phones, and gaming consoles**.

Business Problem

- Analyze Netflix's data to uncover **key trends and patterns** in viewership.
- Identify which **genres, themes, and formats** perform best across different regions.
- Provide insights to help Netflix **decide on new content production** strategies.
- Understand audience preferences to optimize **content recommendation systems**.
- Assess the impact of **release timing and duration** on viewer engagement.
- Explore how Netflix can **expand its presence in different countries**.
- Identify potential markets for **localized and regional content expansion**.
- Evaluate the effectiveness of **marketing strategies** based on content performance.
- Provide data-driven recommendations for **subscription model improvements**.

Dataset

Link: Dataset_link

The dataset contains a comprehensive list of all TV shows and movies available on Netflix, with the following key attributes:

- **Show_id** – Unique ID for each movie/TV show.
- **Type** – Identifies whether the content is a movie or TV show.
- **Title** – Name of the movie/TV show.
- **Director** – Director of the movie.
- **Cast** – List of actors involved.
- **Country** – Country where the content was produced.
- **Date_added** – Date when it was added to Netflix.
- **Release_year** – Actual release year of the content.
- **Rating** – TV rating (e.g., PG, R, etc.).
- **Duration** – Length in minutes or number of seasons.
- **Listed_in** – Genre classification of the content.
- **Description** – A brief summary of the content.

By -Govardhan

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import plotly.io as pio
pio.renderers.default = 'svg'
pio.templates["plotly_dark_custom"] = pio.templates["plotly_dark"]
pio.templates["plotly_dark_custom"].layout.width = 950
pio.templates["plotly_dark_custom"].layout.height = 600
pio.templates.default = "plotly_dark_custom"
```

```
df = pd.read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/ass
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...

Next steps: Generate code with df | View recommended plots | New interactive sheet

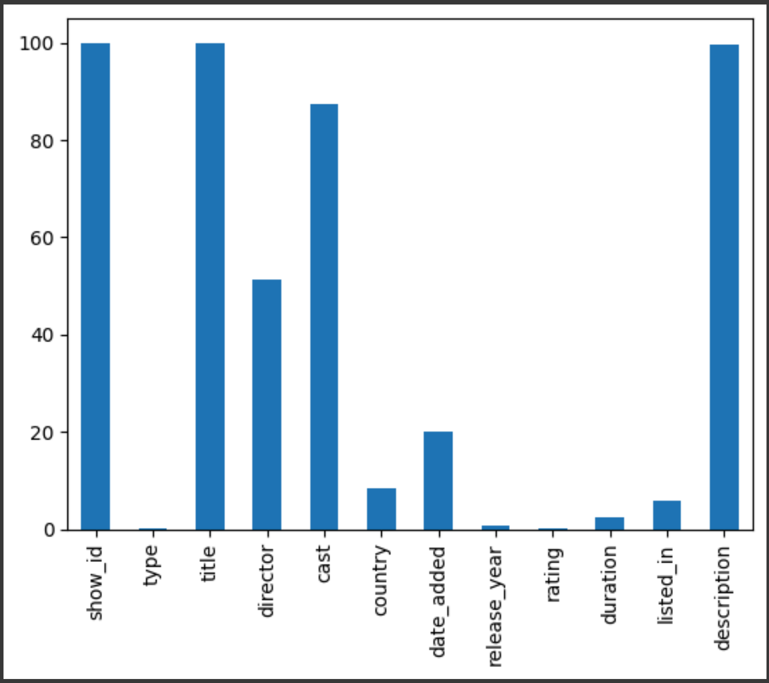
Understand The Data

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64 
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

```
unique_count = df.nunique()
print(unique_count)
(unique_count/df.shape[0]*100).plot(kind='bar')
```

```
show_id         8807
type               2
title           8807
director        4528
cast            7692
country          748
date_added      1767
release_year      74
rating            17
duration         220
listed_in        514
description     8775
dtype: int64
<Axes: >
```



```
repeted_columns = ['type', 'director', 'cast', 'country', 'date_added','
def plot_value_counts(i,repeted_column):
    a = i.value_counts()
    print(a)
    # fig = px.histogram(a,title=repeted_column,histnorm='percent')
    # fig.show()
    print('*'*100)
for i in repeted_columns:
  plot_value_counts(df[i],i)
```

```
duration
1 Season     1793
2 Seasons     425
3 Seasons     199
90 min        152
94 min        146
             ... 
16 min          1
186 min         1
193 min         1
189 min         1
191 min         1
Name: count, Length: 220, dtype: int64
****************************************************************************************************
listed_in
Dramas, International Movies                             362
Documentaries                                            359
Stand-Up Comedy                                          334
Comedies, Dramas, International Movies                   274
Dramas, Independent Movies, International Movies         252
                                                        ... 
Kids' TV, TV Action & Adventure, TV Dramas                 1
TV Comedies, TV Dramas, TV Horror                          1
Children & Family Movies, Comedies, LGBTQ Movies           1
Kids' TV, Spanish-Language TV Shows, Teen TV Shows         1
Cult Movies, Dramas, Thrillers                             1
Name: count, Length: 514, dtype: int64
****************************************************************************************************
```

Analysis Of Missing Data

1)Is data missing randomly or with a specific pattren

```
#Data Columns With Missing Data
df.isna().sum()/df.shape[0]*100
```

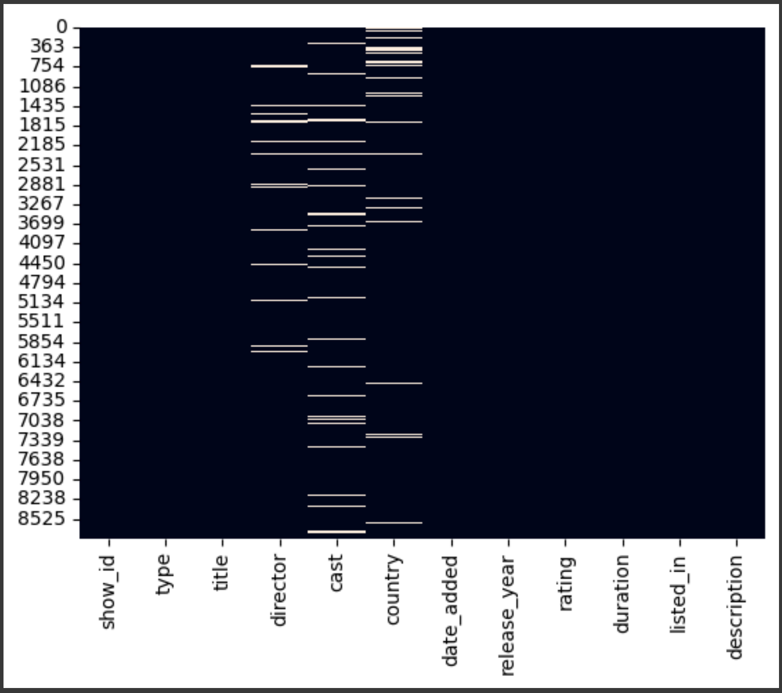
	0
show_id	0.000000
type	0.000000
title	0.000000
director	29.908028
cast	9.367549
country	9.435676
date_added	0.113546
release_year	0.000000
rating	0.045418
duration	0.034064
listed_in	0.000000
description	0.000000

dtype: float64

--> How My Data Is Missing

```
sns.heatmap(df[df['type']=='Movie'].isna(),cbar=False)
```

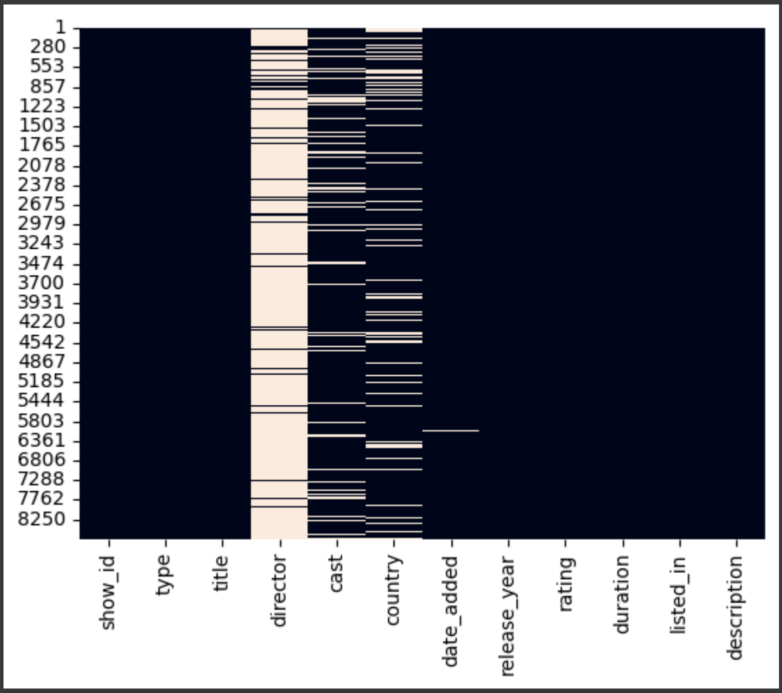
```
<Axes: >
```



less missing values in movies

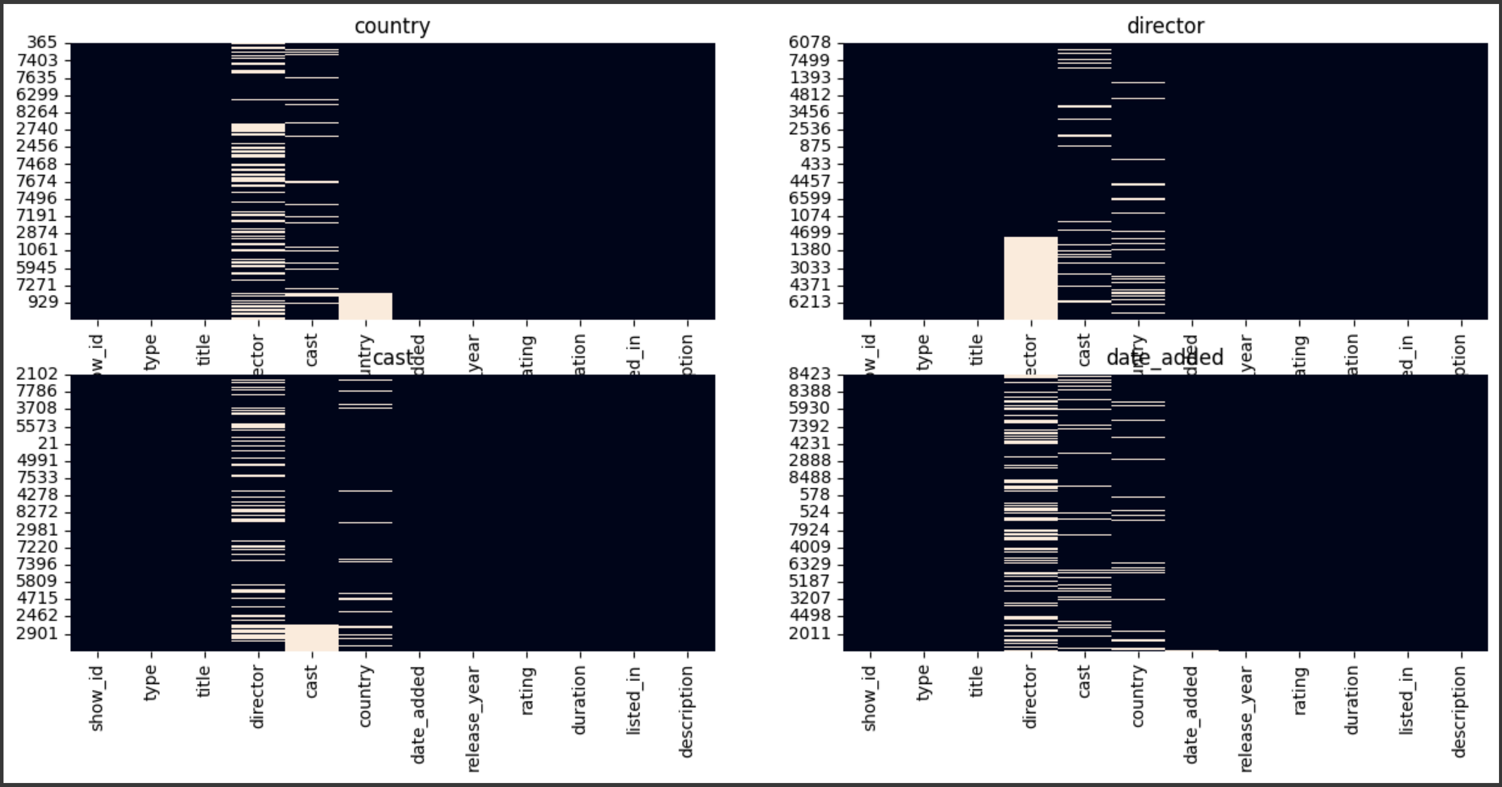
```
sns.heatmap(df[df['type']=='TV Show'].isna(),cbar=False)
```

```
<Axes: >
```



lot of missing values in series than movies

```
loc = 1
fig = plt.figure(figsize=(15,10))
for i in ['country','director','cast','date_added']:
  plt.subplot(3,2,loc)
  sns.heatmap(df.sort_values(i).isna(),cbar=False)
  plt.title(i)
  loc+=1

plt.show()
```



1. The Data Is Missing In Random
2. Lot Of missing data in Director Column

```
#duration--
df.loc[df['duration'].isna()] #3 Nulls But Mis-Enterd in rating
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	April 4, 2017	2017	74 min	NaN	Movies	Louis C.K. muses on religion, eternal love, gi...
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010	84 min	NaN	Movies	Emmy-winning comedy writer Louis C.K. brings h...
5813	s5814	Movie	Louis C.K.: Live at the	Louis	Louis	United	August 15,	2015	66 min	NaN	Movies	The comic puts his trademark

```
df['duration1'] = df['duration']
```

```
df.loc[df['duration'].isna(),'duration'] = df.loc[df['duration'].isna(),
```

```
#Replace MisEnterd Rating to None
df.loc[df['duration1'].isna(),'rating'] = None
```

```
df['duration'].isna().sum()
```

```
0
```

```
#rating
df['rating'].isna().sum()
```

```
7
```

```
df[df['rating'].isna()].head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	dura
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	April 4, 2017	2017	None	74 min	Movies	Louis C.K. muses on religion, eternal love, gi...	
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010	None	84 min	Movies	Emmy-winning comedy writer Louis C.K. brings h...	
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	August 15, 2016	2015	None	66 min	Movies	The comic puts his trademark hilarious/thought...	
5989	s5990	Movie	13TH: A Conversation with Oprah Winfrey & Ava ...	NaN	Oprah Winfrey, Ava DuVernay	NaN	January 26, 2017	2017	NaN	37 min	Movies	Oprah Winfrey sits down with director Ava DuVe...	3
6827	s6828	TV Show	Gargantia on the Verdurous Planet	NaN	Kaito Ishikawa, Hisako Kanemoto, Ai Kayano, Ka...	Japan	December 1, 2016	2013	NaN	1 Season	Anime Series, International TV Shows	After falling through a wormhole, a space-dwel...	1 S

```
#Understanding How listed_in is Related to rating
df.groupby('listed_in')['rating'].sum().head()
```

	rating
listed_in	
Action & Adventure	RRRRRTV-MARPG-13RRRRPG-13PG-13RPG-13RRRRRRRTV-...
Action & Adventure, Anime Features	TV-MA
Action & Adventure, Anime Features, Children & Family Movies	TV-PGPGTV-PGTV-14
Action & Adventure, Anime Features, Classic Movies	TV-14PG-13
Action & Adventure, Anime Features, Horror Movies	TV-MA

dtype: object

```
#date_added
df['date_added'].value_counts().head()
```

	count
date_added	
January 1, 2020	109
November 1, 2019	89
March 1, 2018	75
December 31, 2019	74
October 1, 2018	71

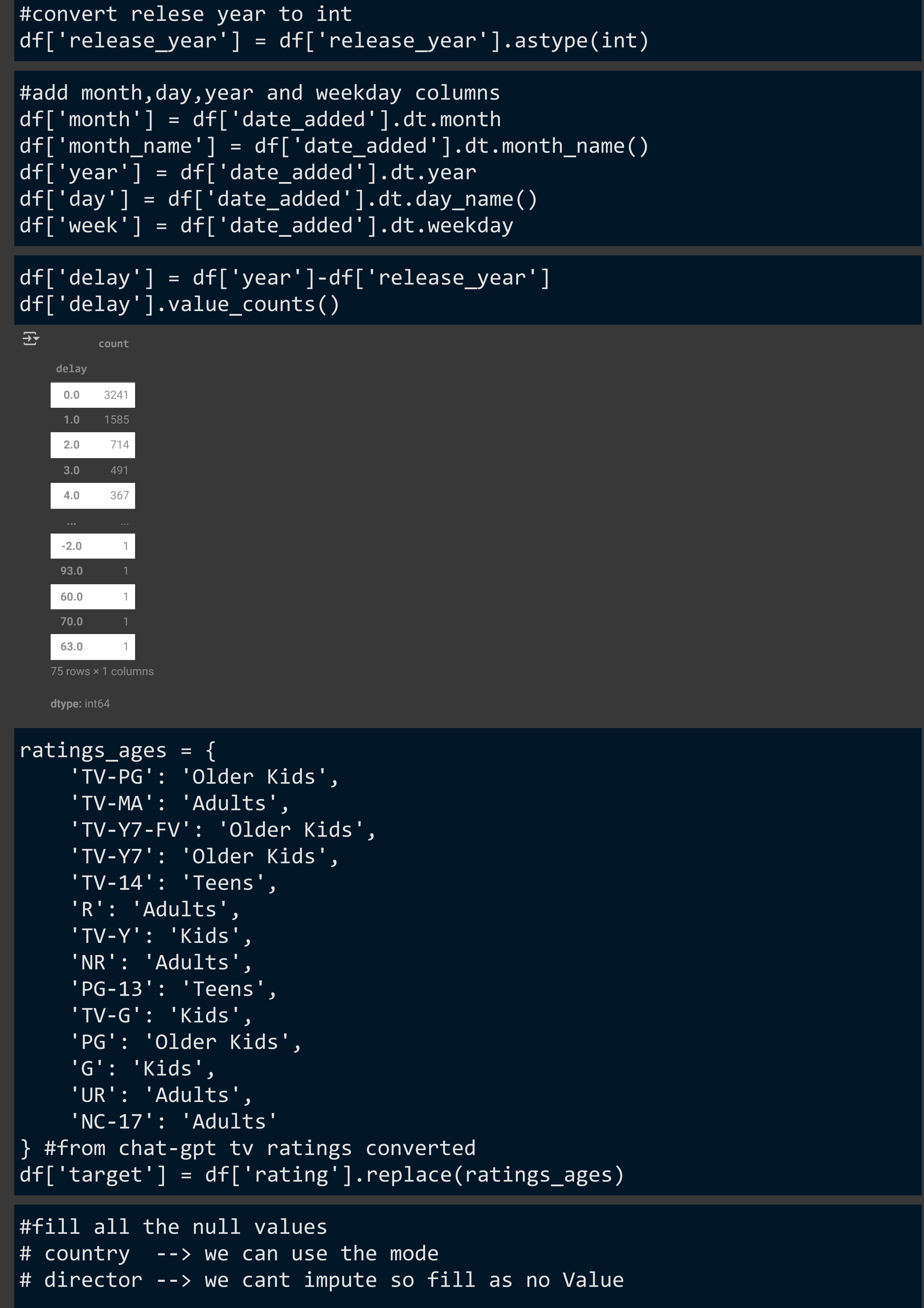
dtype: int64

```
#Convert Date TO DateTime
df['date_added'] = pd.to_datetime(df['date_added'].str.strip(),errors='co
```

```
df['date_added'].isna().sum()
```

```
10
```

We Can Ignore The Nulls has They are small in number



```
#convert relese year to int
df['release_year'] = df['release_year'].astype(int)
```

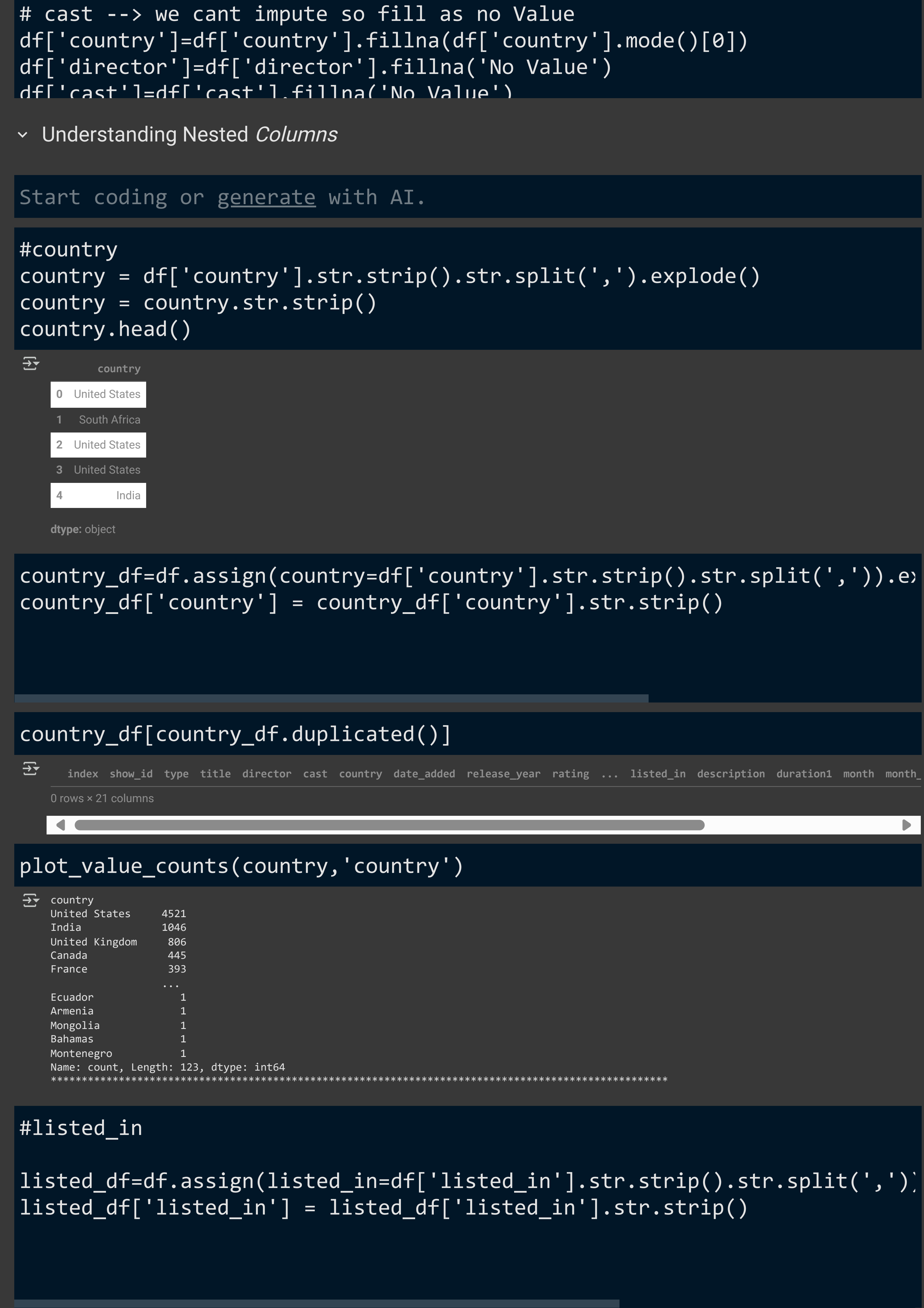
```
#add month,day,year and weekday columns
df['month'] = df['date_added'].dt.month
df['month_name'] = df['date_added'].dt.month_name()
df['year'] = df['date_added'].dt.year
df['day'] = df['date_added'].dt.day_name()
df['week'] = df['date_added'].dt.weekday
```

```
df['delay'] = df['year']-df['release_year']
df['delay'].value_counts()
```

```
        count
delay
  0.0    3241
  1.0    1585
  2.0     714
  3.0     491
  4.0     367
  ...     ...
 -2.0       1
 93.0       1
 60.0       1
 70.0       1
 63.0       1
75 rows × 1 columns

dtype: int64
```

```
ratings_ages = {
    'TV-PG': 'Older Kids',
    'TV-MA': 'Adults',
    'TV-Y7-FV': 'Older Kids',
    'TV-Y7': 'Older Kids',
    'TV-14': 'Teens',
    'R': 'Adults',
    'TV-Y': 'Kids',
    'NR': 'Adults',
    'PG-13': 'Teens',
    'TV-G': 'Kids',
    'PG': 'Older Kids',
    'G': 'Kids',
    'UR': 'Adults',
    'NC-17': 'Adults'
} #from chat-gpt tv ratings converted
df['target'] = df['rating'].replace(ratings_ages)
```

```
#fill all the null values
# country  --> we can use the mode
# director --> we cant impute so fill as no Value
```



```
# cast --> we cant impute so fill as no Value
df['country']=df['country'].fillna(df['country'].mode()[0])
df['director']=df['director'].fillna('No Value')
df['cast']=df['cast'].fillna('No Value')
```

Understanding Nested *Columns*

```
Start coding or generate with AI.
```

```
#country
country = df['country'].str.strip().str.split(',').explode()
country = country.str.strip()
country.head()
```

	country
0	United States
1	South Africa
2	United States
3	United States
4	India

dtype: object

```
country_df=df.assign(country=df['country'].str.strip().str.split(',')).ex
country_df['country'] = country_df['country'].str.strip()
```

```
country_df[country_df.duplicated()]
```

index	show_id	type	title	director	cast	country	date_added	release_year	rating	...	listed_in	description	duration1	month	month_

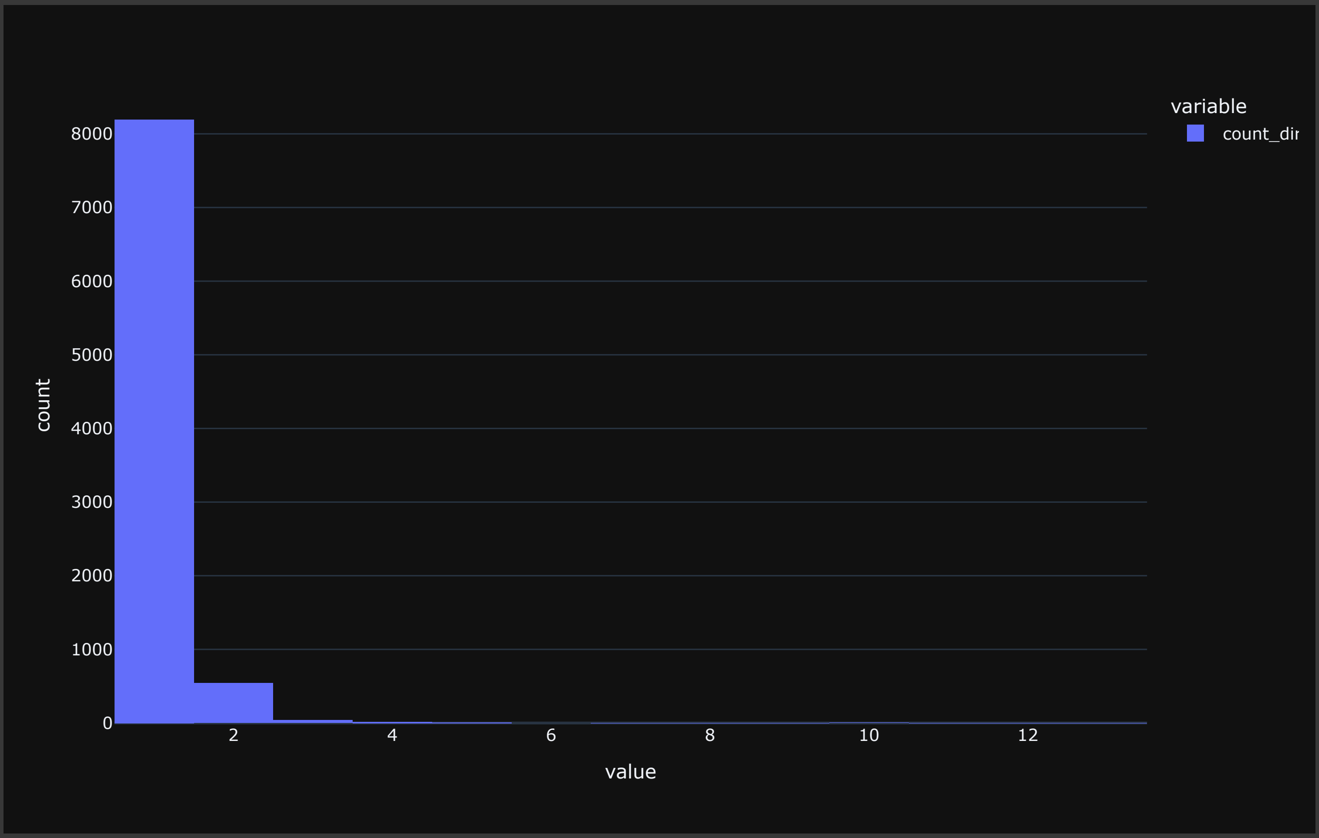
0 rows × 21 columns

```
plot_value_counts(country,'country')
```

```
country
United States     4521
India             1046
United Kingdom     806
Canada             445
France             393
                  ...
Ecuador              1
Armenia              1
Mongolia             1
Bahamas              1
Montenegro           1
Name: count, Length: 123, dtype: int64
****************************************************************************************************
```

```
#listed_in

listed_df=df.assign(listed_in=df['listed_in'].str.strip().str.split(','))
listed_df['listed_in'] = listed_df['listed_in'].str.strip()
```

```
#director
df['director'].value_counts()
df['count_dir']=df['director'].str.split(',').apply(lambda x:len(x))
px.histogram(df['count_dir'])
```



we can observe lot of values so needs unnesting

```
#director unnest
director_df=df[['director','show_id']].assign(director=df['director'].str
director_df['director'] = director_df['director'].str.strip()

director_df.head()
```

	index	director	show_id
0	0	Kirsten Johnson	s1
1	1	No Value	s2
2	2	Julien Leclercq	s3
3	3	No Value	s4
4	4	No Value	s5

Next steps: Generate code with director_df View recommended plots New interactive sheet

```
#Cast
cast_df=df[['cast','show_id']].assign(cast=df['cast'].str.strip().str.spl
cast_df['cast'] = cast_df['cast'].str.strip()
cast_df.head()
```

	index	cast	show_id
0	0	No Value	s1
1	1	Ama Qamata	s2
2	1	Khosi Ngema	s2
3	1	Gail Mabalane	s2
4	1	Thabang Molaba	s2

Next steps: Generate code with cast_df View recommended plots New interactive sheet

Divide The Date into movies and series

```
movies = df[df['type']=='Movie']
series = df[df['type']=='TV Show']
```

['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added', 'release_year', 'rating', 'duration', 'listed_in', 'description']

```
df['type'].value_counts()/df.shape[0]*100
```

	count
type	
Movie	69.615079
TV Show	30.384921

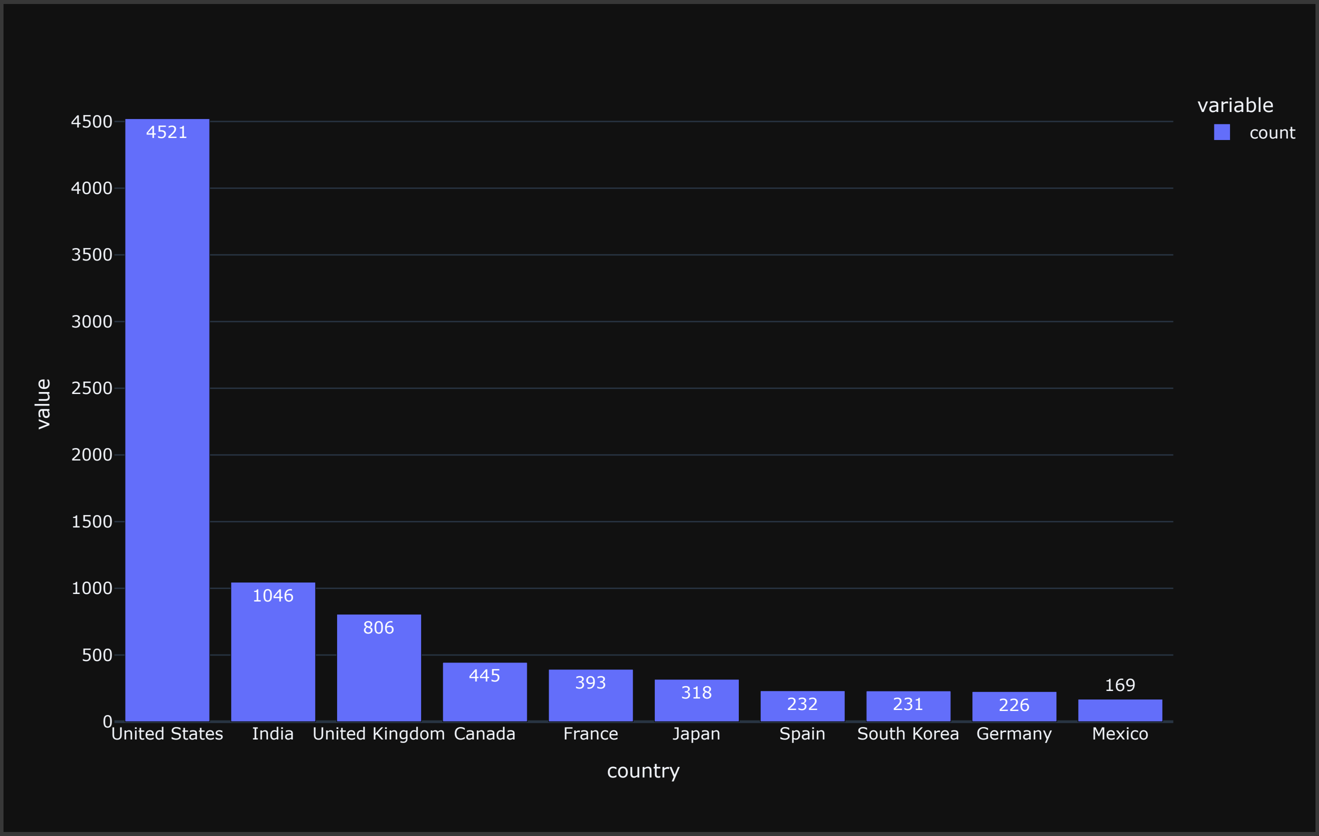
dtype: float64

70% movies and 30% series

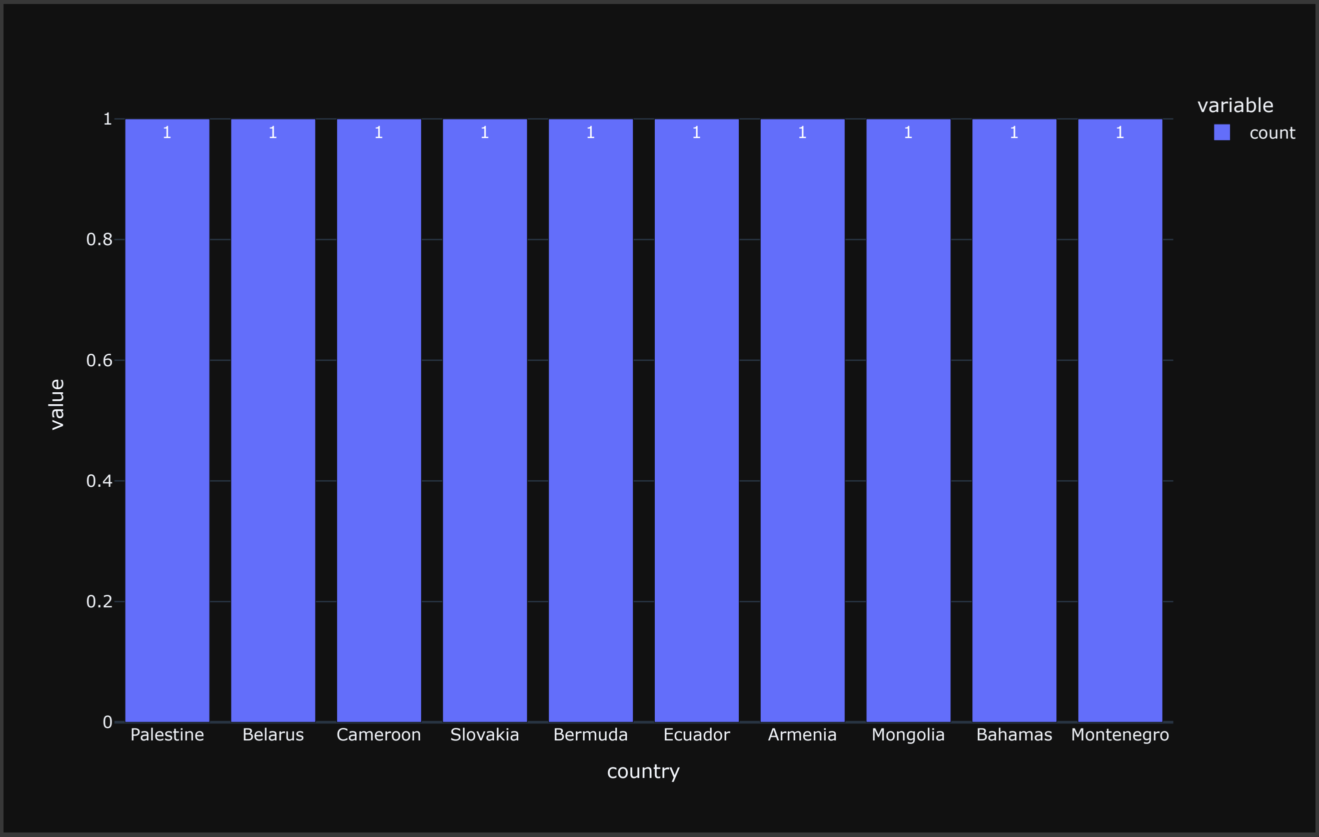
```
Start coding or generate with AI.
```

*Analysis *

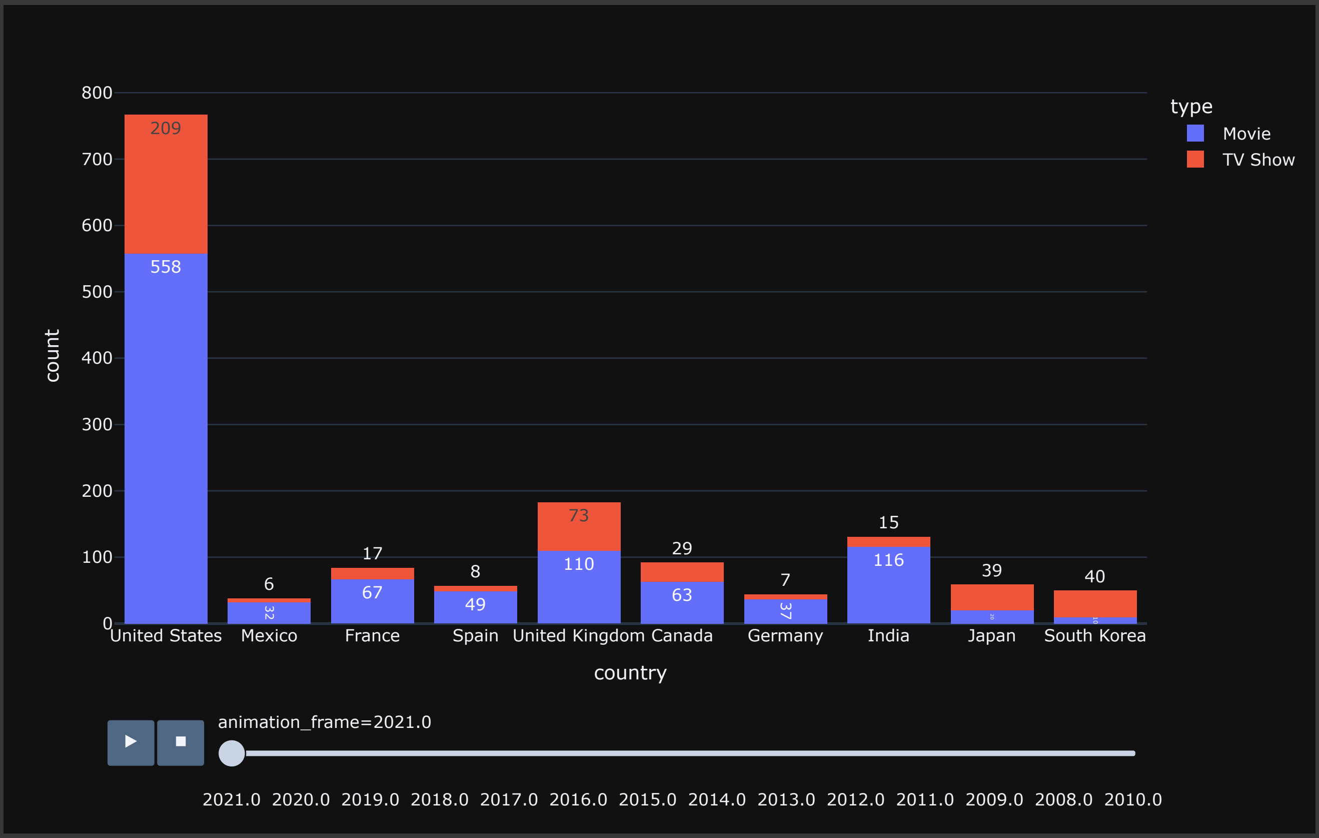
```
#top 10 Country's:
top_10_country = country.value_counts()[:10]
px.bar(top_10_country,text_auto=True)
```



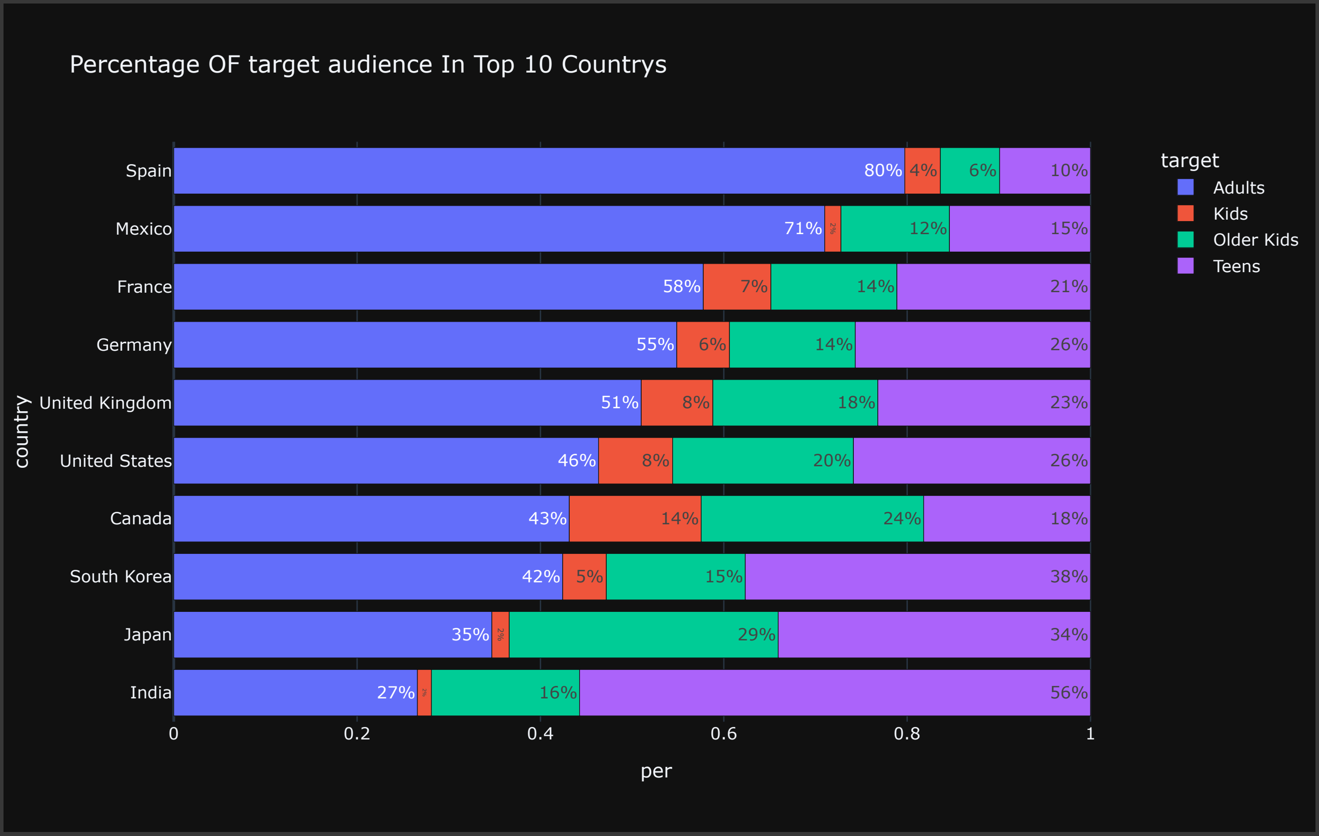
```
#top 10 Country's:
top_10_bottom_country = country.value_counts()[-10:]
fig = px.bar(top_10_bottom_country,text_auto=True)
fig.show()
```



```
#How Top_10 is Related TO Last 20 to 40 Years
top_10_country_df = country_df[country_df['country'].isin(top_10_country.
fig = px.histogram(top_10_country_df.sort_values('date_added'),x='country
fig.show()
```

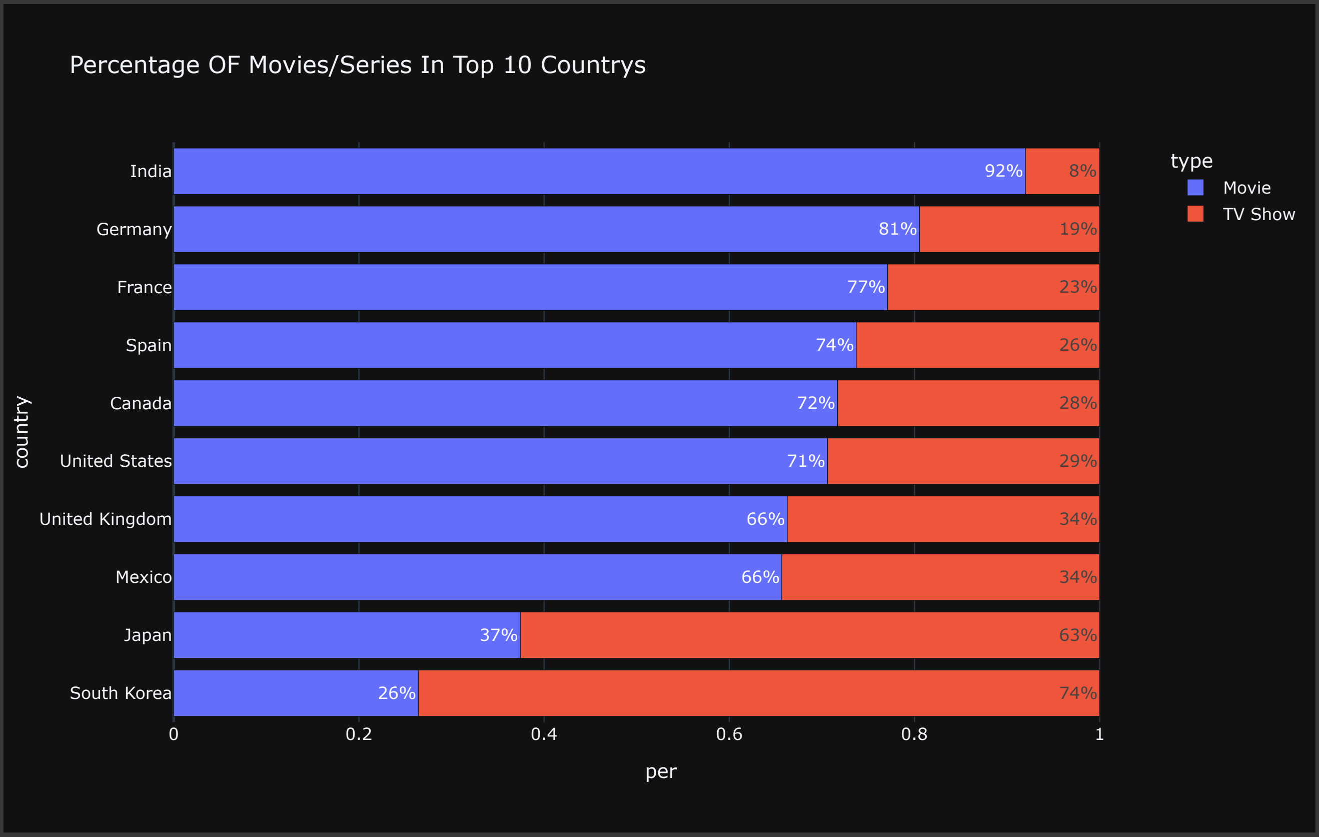


```
title = "Percentage OF target audience In Top 10 Countrys"
a = top_10_country_df.groupby(['country','target'])['show_id'].count().re
a['sum'] = a.groupby(['country'])['show_id'].transform('sum')
a['per']=a['show_id']/a['sum']
px.bar(a.sort_values(['target','per']),y='country',x='per',color='target'
```



- india has lot of teen shows
- spain focus on adult shows

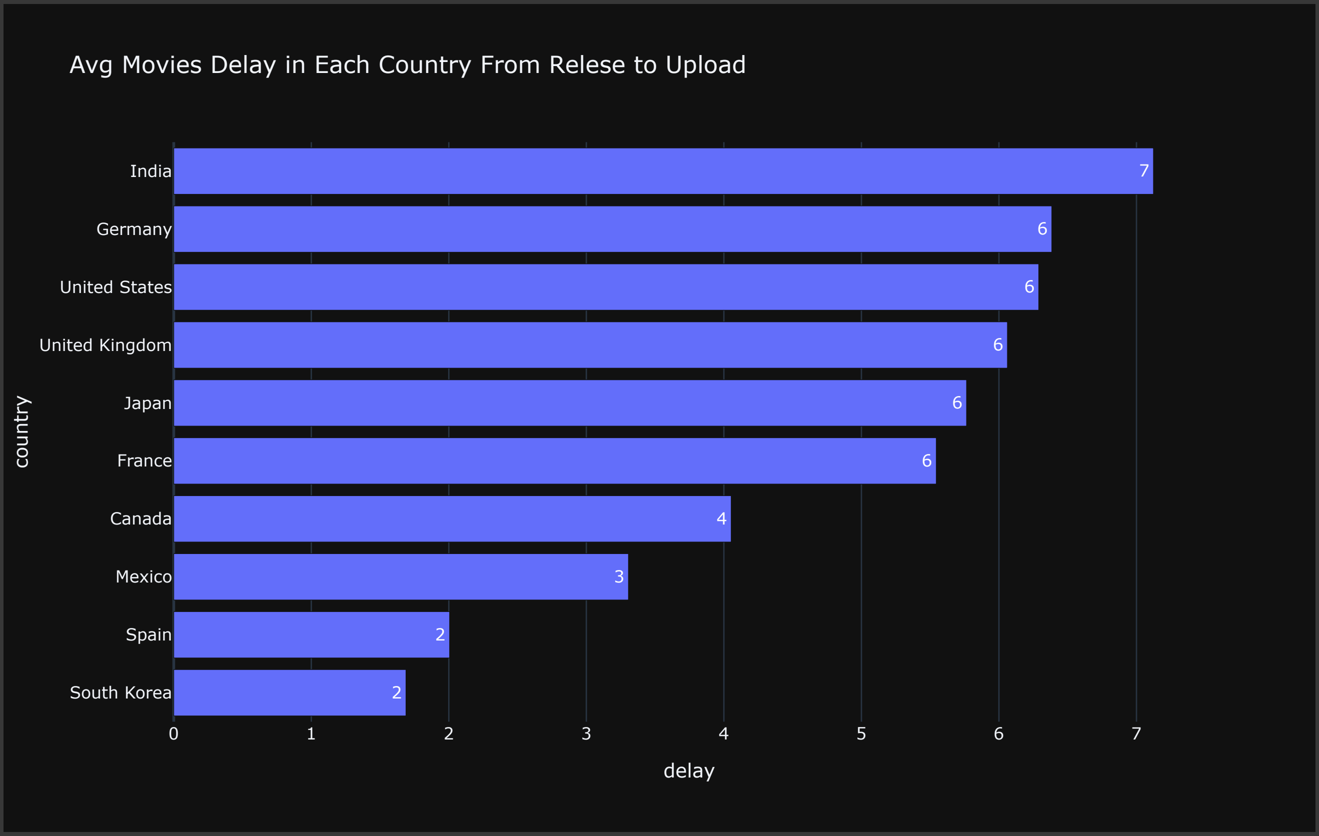
```
title = "Percentage OF Movies/Series In Top 10 Countrys"
a = top_10_country_df.groupby(['country','type'])['show_id'].count().rese
a['sum'] = a.groupby(['country'])['show_id'].transform('sum')
a['per']=a['show_id']/a['sum']
px.bar(a.sort_values(['type','per']),y='country',x='per',color='type',tex
```



- we can see different countries have different distribution
- south korea has more series than movies
- in india Netflix has to focus on releasing tv shows

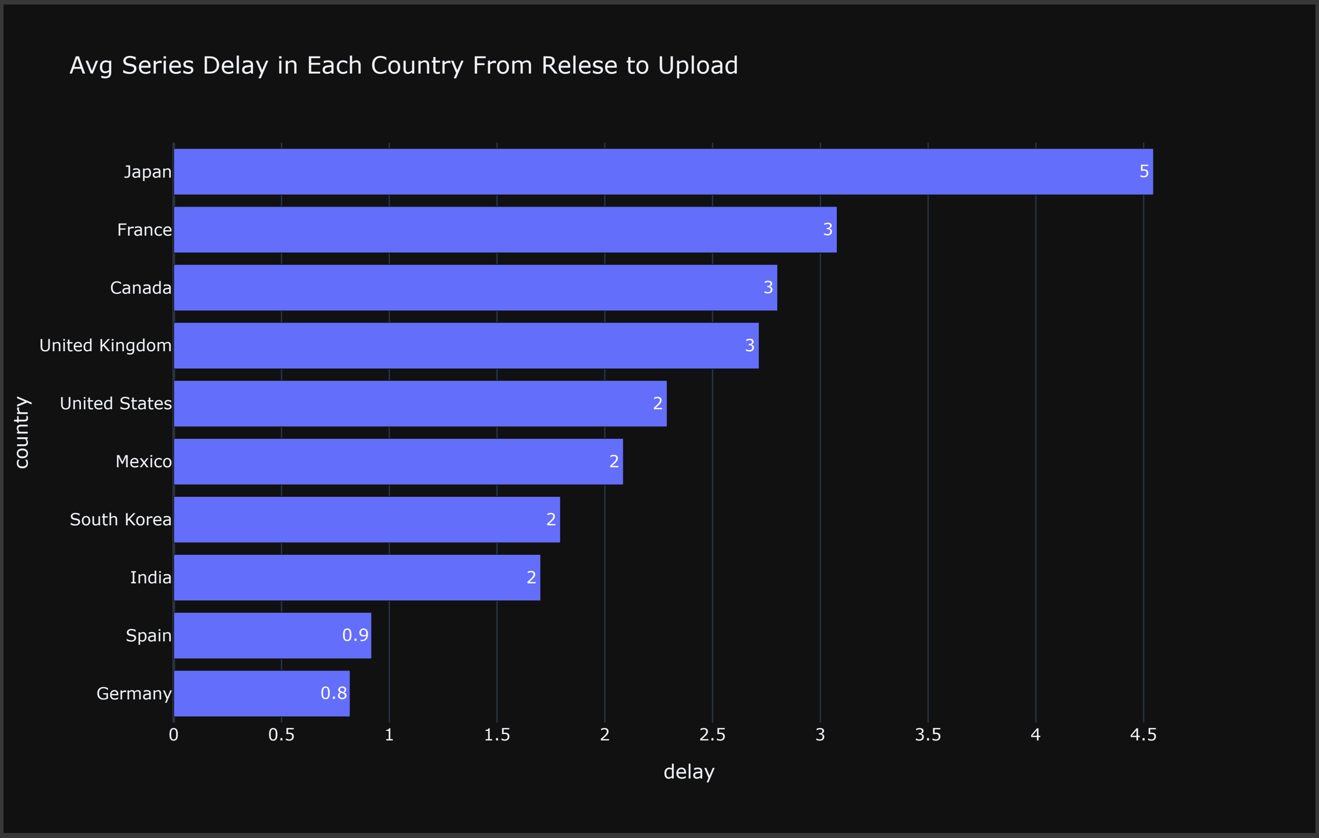
```
Start coding or generate with AI.
```

```
def abc(x):
  return x.drop_duplicates(['show_id'])['delay'].mean()
title = "Avg Movies Delay in Each Country From Relese to Upload"
a = top_10_country_df[top_10_country_df['type']=='Movie'].groupby(['count
a.columns = ['country','delay']
px.bar(a.sort_values('delay'),y='country',x='delay',text_auto='.0',title=
```



India and Germany has the 7 years gap in movie relese and upload

```
title = "Avg Series Delay in Each Country From Relese to Upload"
a = top_10_country_df[top_10_country_df['type']=='TV Show'].groupby('cour
a.columns = ['country','delay']
px.bar(a.sort_values('delay'),y='country',x='delay',text_auto='.0',title=
```



- In Series release and Upload gap is low compared to movies
- In Japan the delay is high

```
exp = '% of the movies are relesed in top_10 Countryes {}'.format(top_10_
print(top_10_country_df.shape[0]/df.shape[0]*100,exp)
```

```
95.23106619734301 % of the movies are relesed in top_10 Countryes Index(['United States', 'India', 'United Kingdom', 'Canada', 'France', 'Jap
       'Spain', 'South Korea', 'Germany', 'Mexico'],
      dtype='object', name='country')
```

```
count_country= country.value_counts()
print("In {} number of movies released are {} which is {}%".format(count_
```

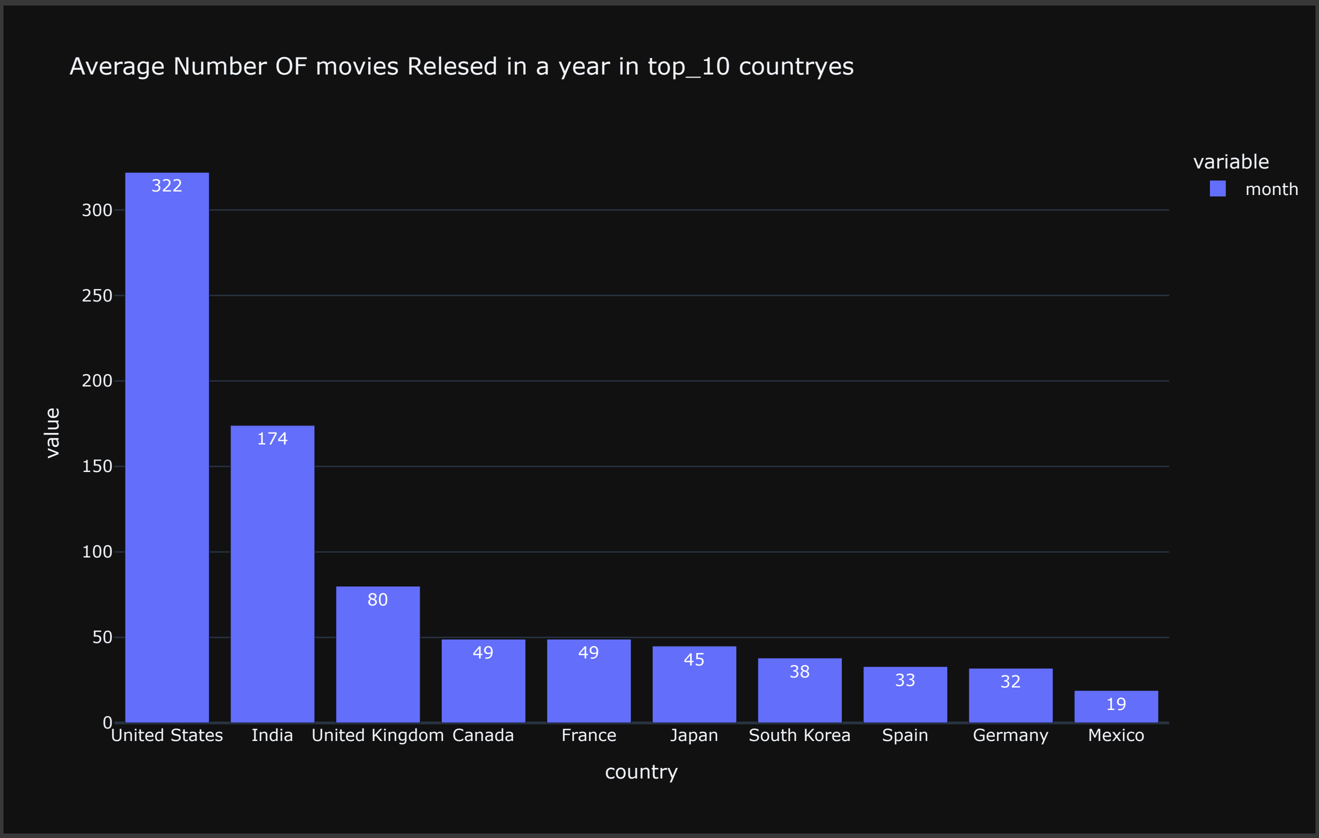
```
In United States number of movies released are 4521 which is 51.33416600431475%
```

```
per_country = count_country/df.shape[0]*100
print("Top 2 Countries relese {}% of Movies".format(per_country[:2].sum()
```

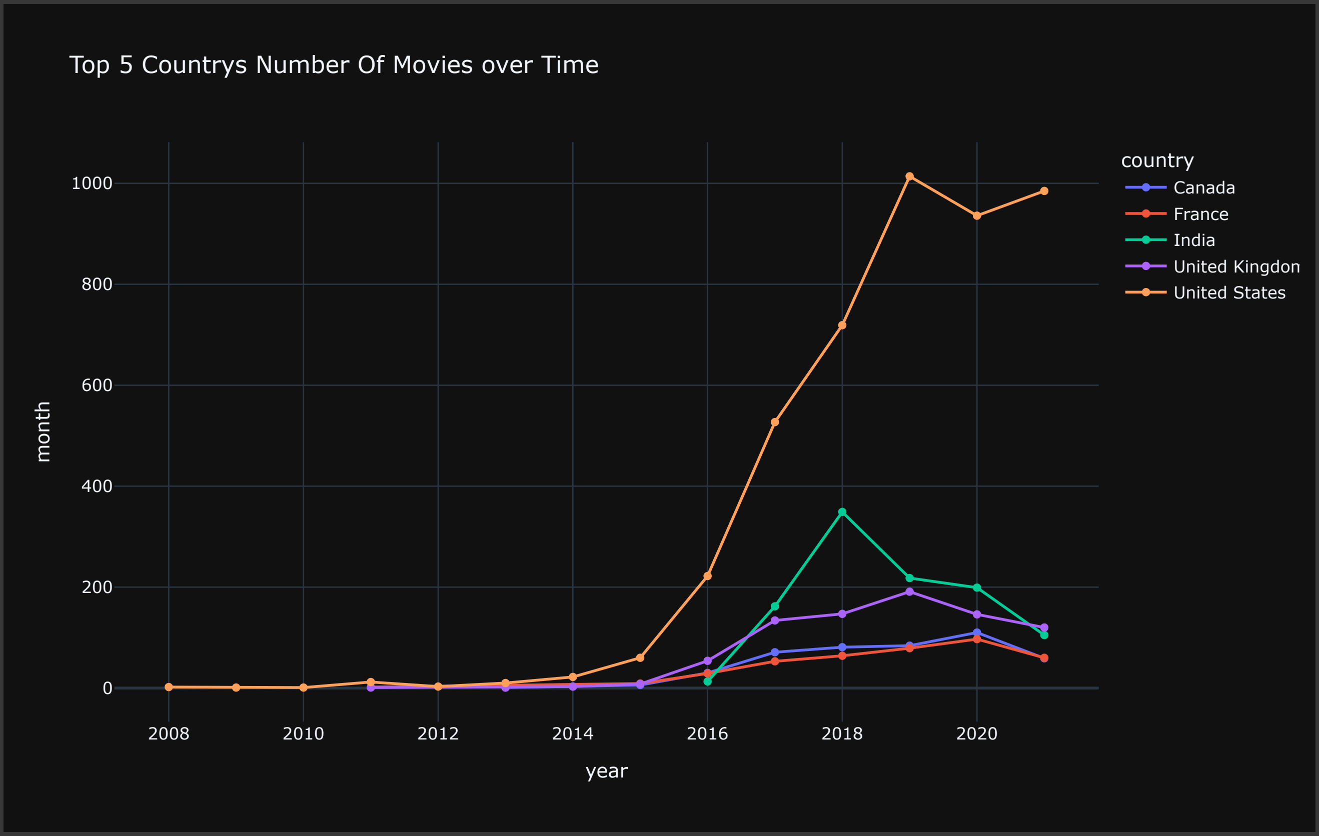
```
Top 2 Countries relese 63.21108209378903% of Movies
```

```
title = "Average Number OF movies Relesed in a year in top_10 countryes"
num_mov = top_10_country_df.groupby(['country','year'])['month'].count().
```

```
px.bar(num_mov.groupby('country')['month'].mean().round(0).sort_values(as
```



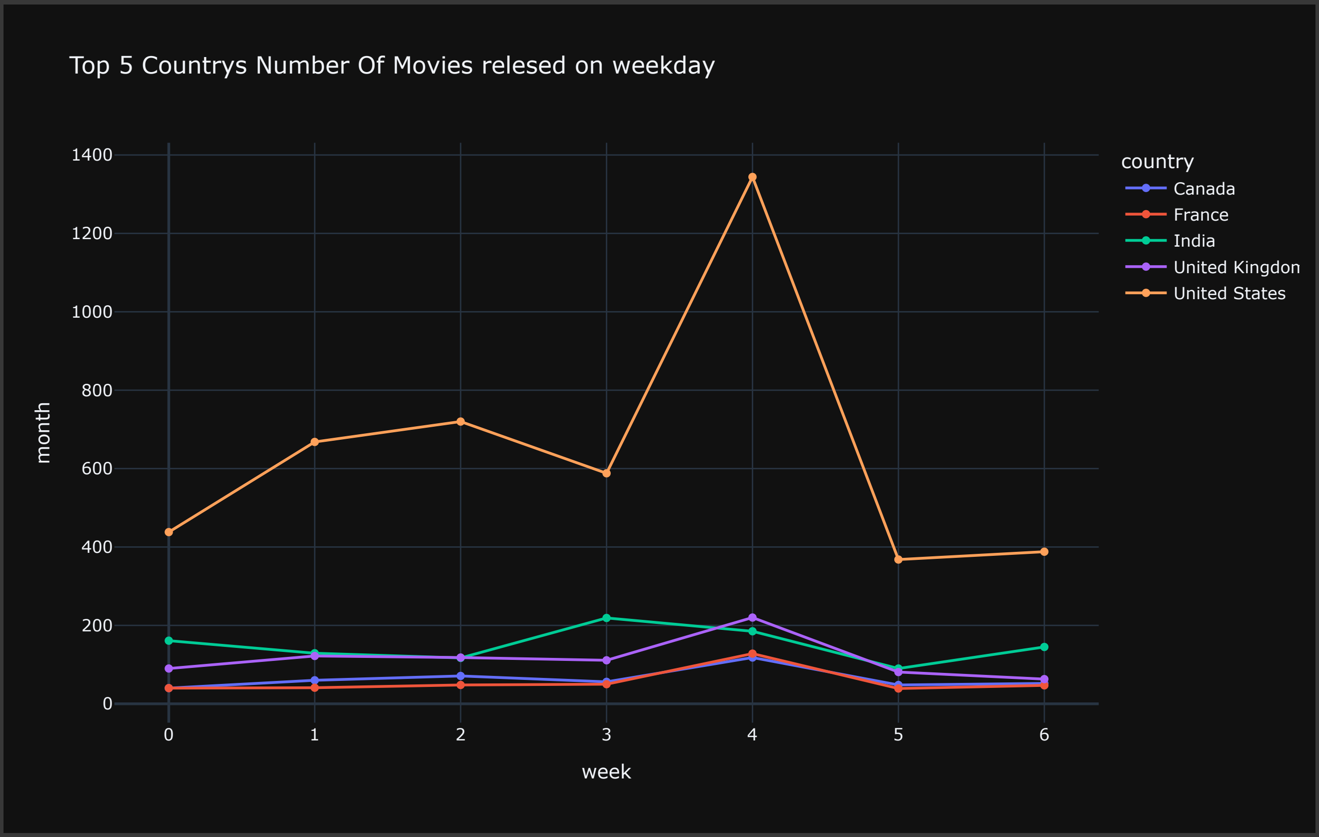
```
title = "Top 5 Countrys Number Of Movies over Time"
fig = px.line(num_mov[num_mov['country'].isin(count_country[:5].index)],x

fig.show()
```



- We Can Cearly see that in pandamin the number goes down (trend)
- Peak is in in 2019

```
num_week = top_10_country_df.groupby(['country','week'])['month'].count(
```

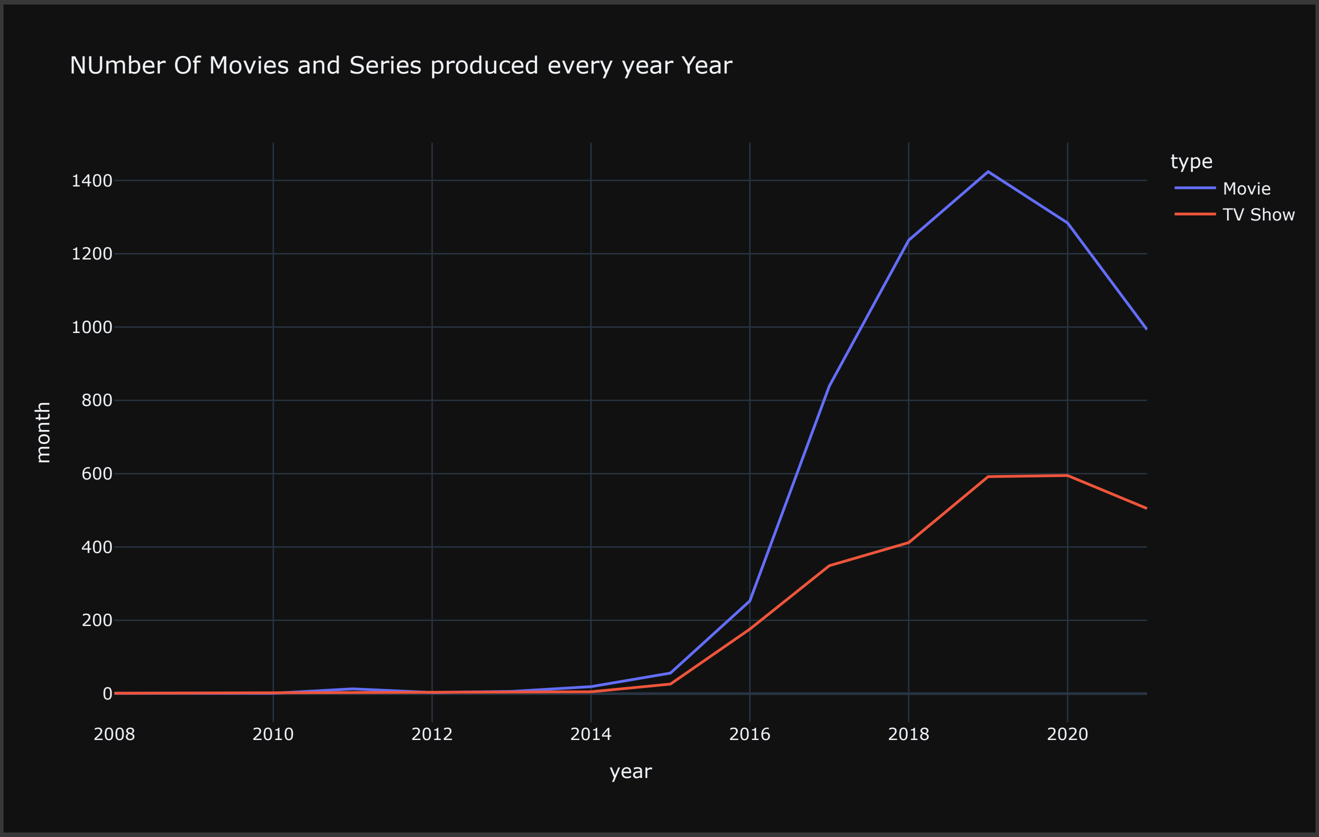
```
title = "Top 5 Countrys Number Of Movies relesed on weekday"
px.line(num_week[num_week['country'].isin(count_country[:5].index)],x='we
```



- we can observe sudden spike in number of movies released on Friday

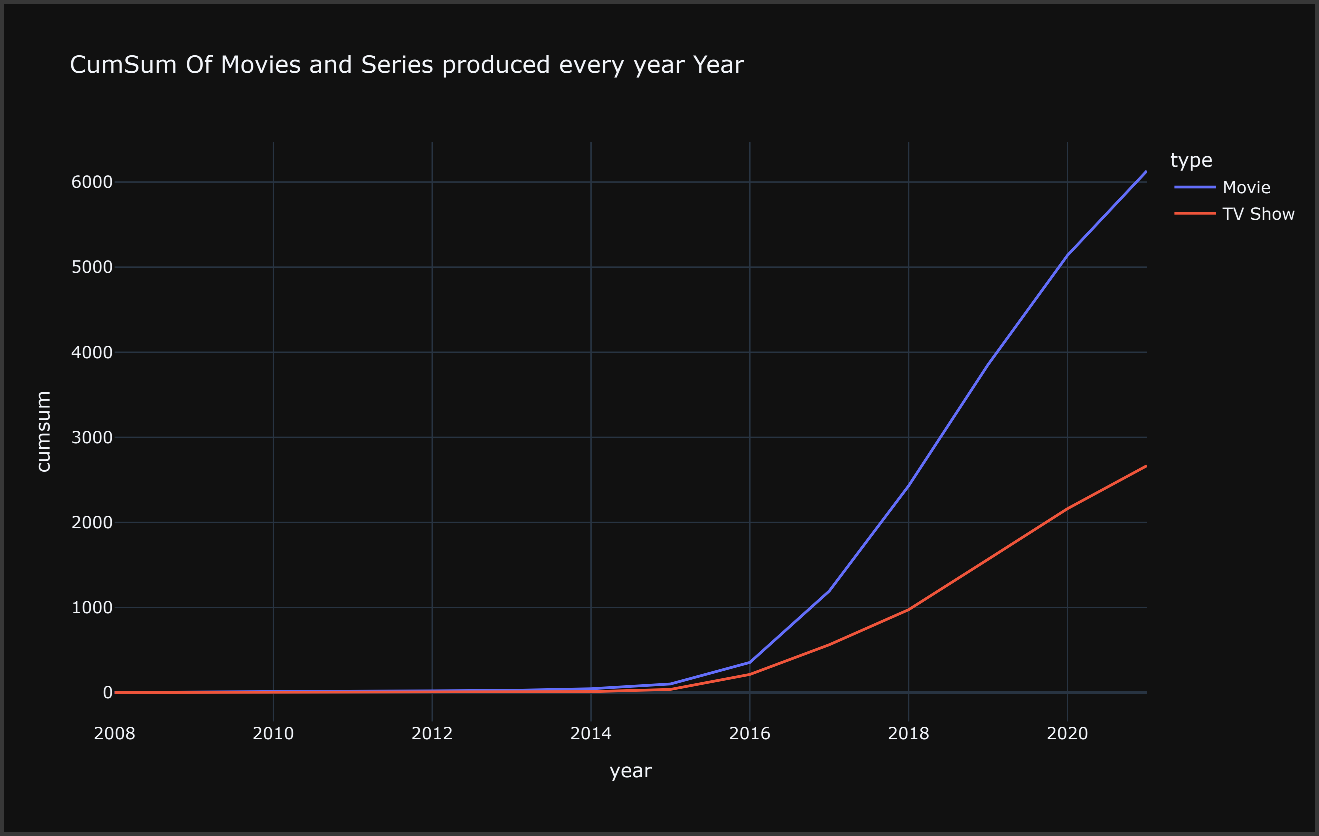
```
#Lets Understand How ratio of movies and series changes over time
type_group = df.groupby(['year','type'])['month'].count().reset_index()
```

```
title = "NUmber Of Movies and Series produced every year Year"
px.line(type_group,x='year',y='month',color='type',title=title)
```



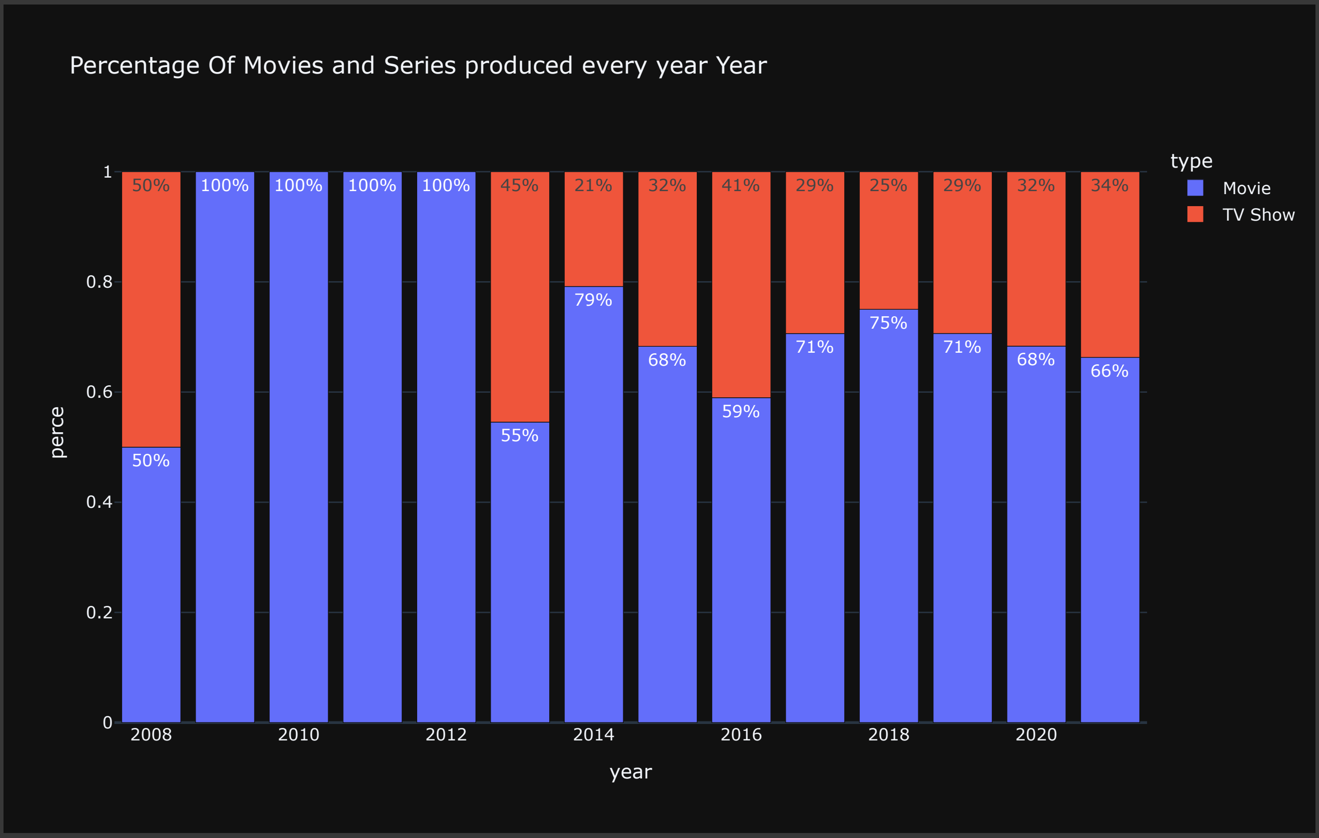
number movies and series are increasing over time

```
type_group['cumsum'] = type_group.groupby('type')['month'].cumsum()
title = "CumSum Of Movies and Series produced every year Year"
px.line(type_group,x='year',y='cumsum',color='type',title=title)
```



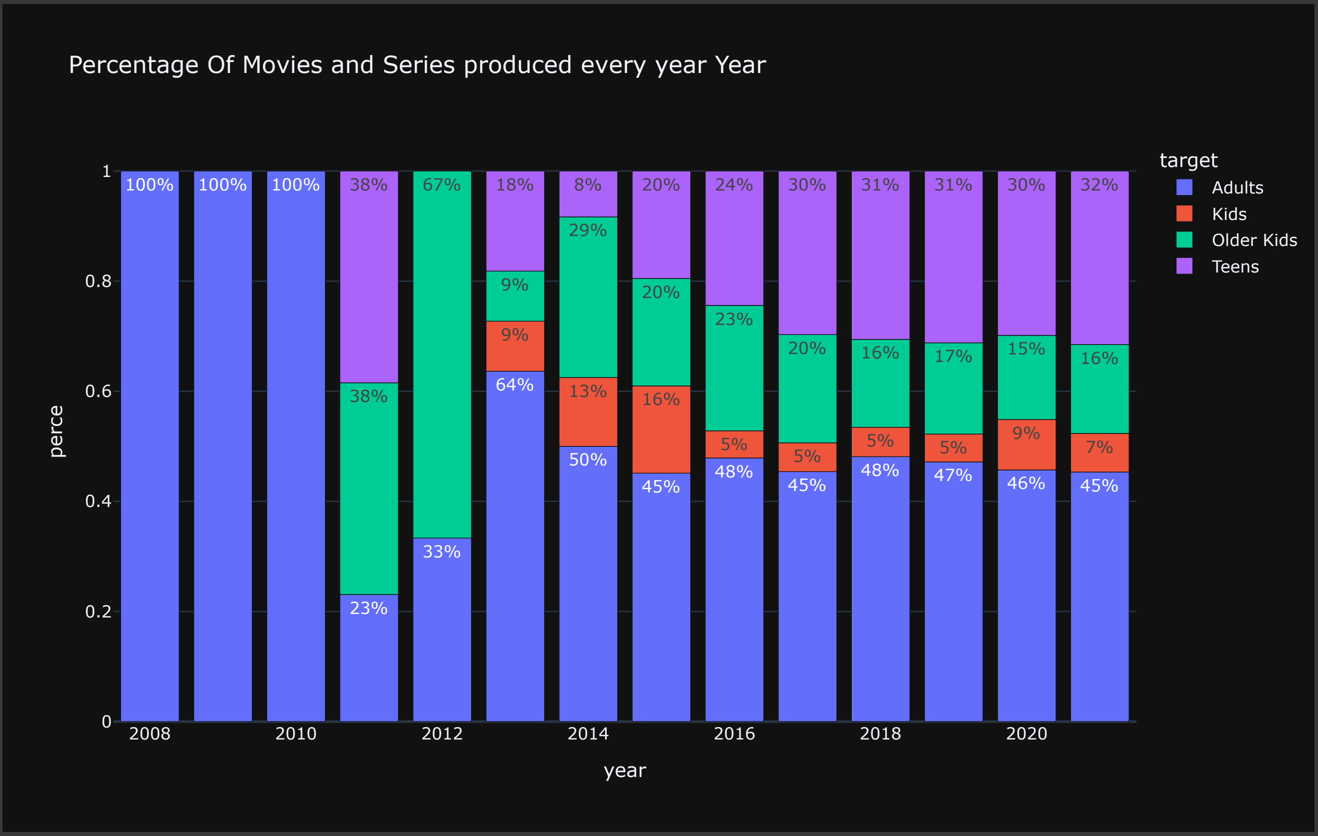
Movies Are Increasing Rapidly Than Series

```
type_group['perce']=type_group.groupby('year')['month'].transform(sum)
type_group['perce']=(type_group['month']/type_group['perce'])
title = "Percentage Of Movies and Series produced every year Year"
px.bar(type_group.sort_values(['type','perce']),x='year',y='perce',color=
```



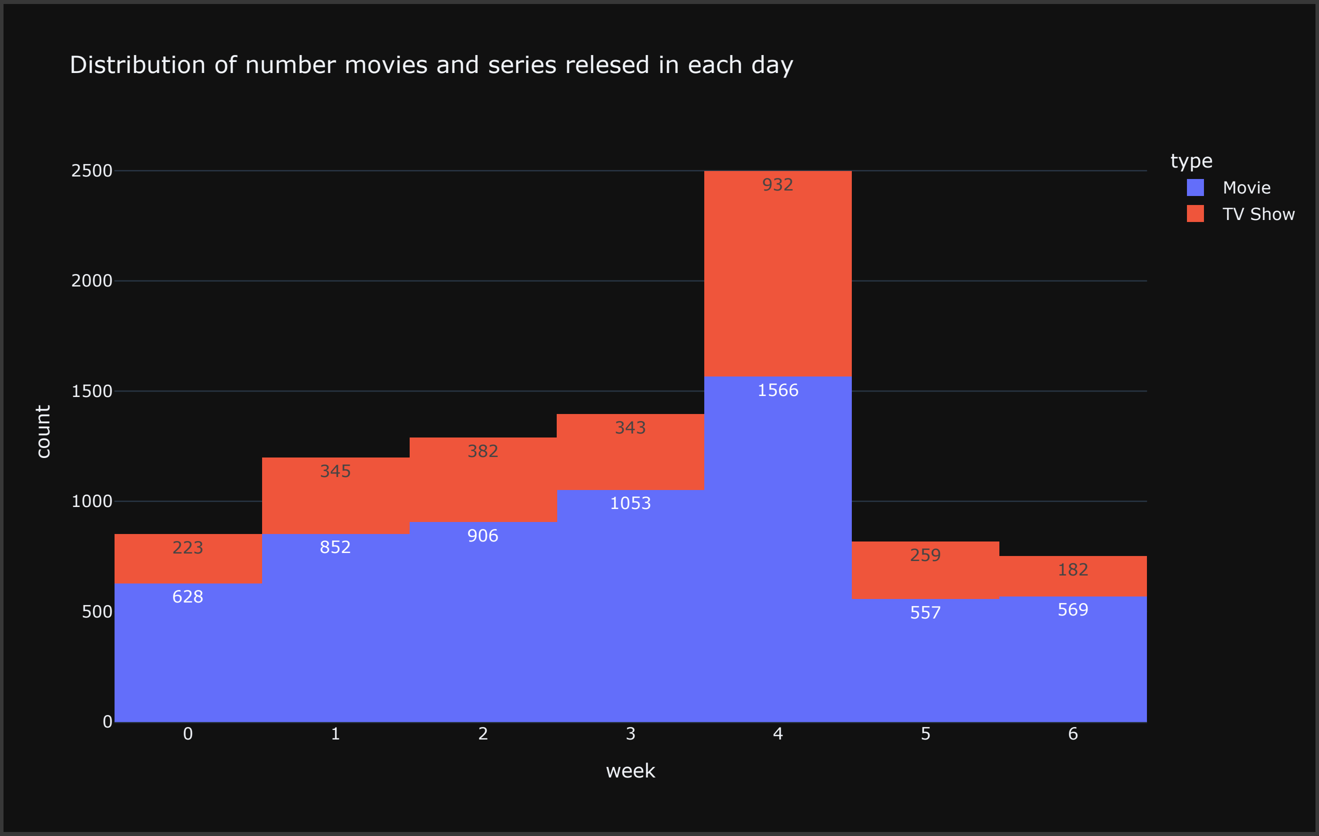
all the years we have series less than 40% netflix uploading more movies compared to series

```
title="Percentage Of Target audians for each year"
type_group = df.groupby(['year','target'],as_index=True)['month'].count(
type_group['perce']=type_group.groupby('year')['month'].transform(sum)
type_group['perce']=(type_group['month']/type_group['perce'])
title = "Percentage Of Movies and Series produced every year Year"
px.bar(type_group.sort_values(['target','perce']),x='year',y='perce',colo
```



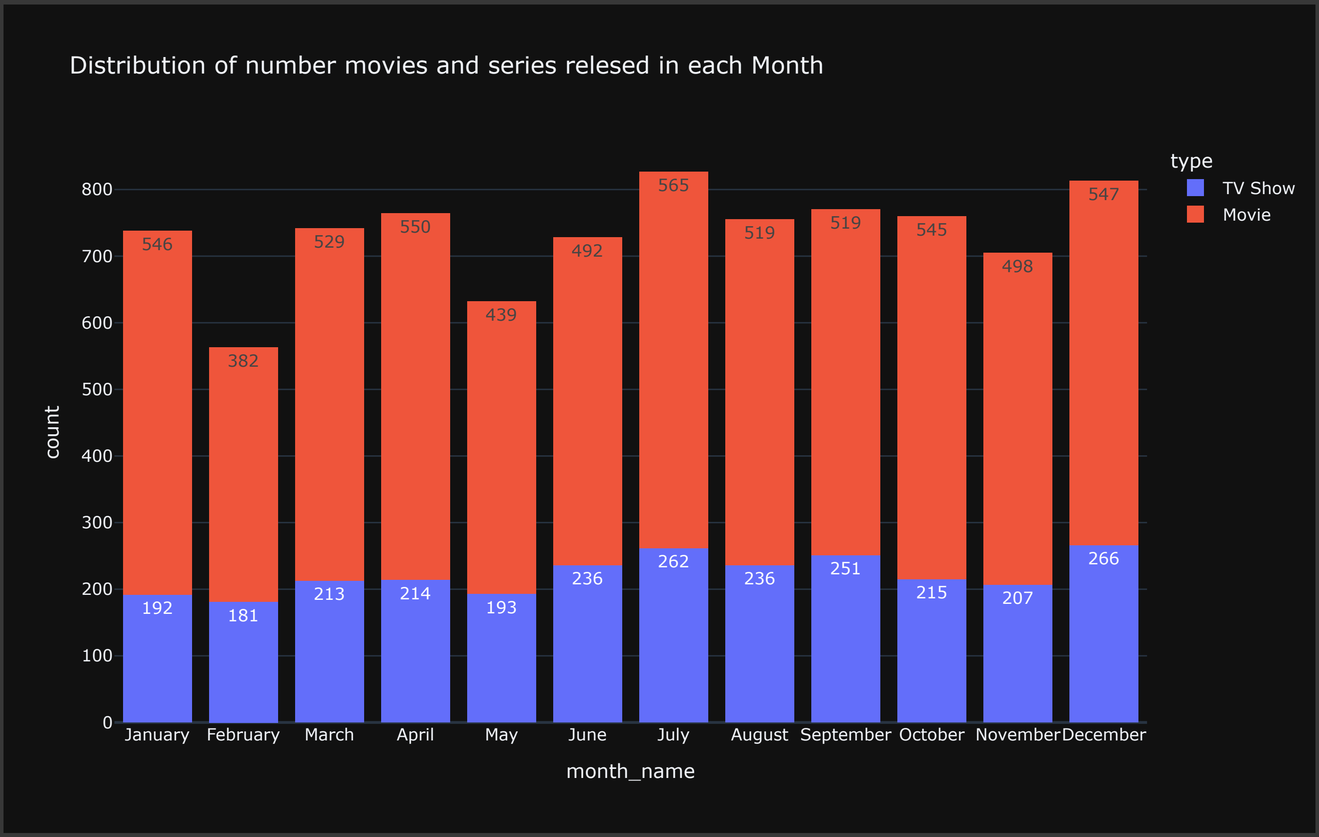
in recent years the distribution remains same

```
#What is the best time to launch a TV show?
title="Distribution of number movies and series relesed in each day"
px.histogram(df,x='week',color='type',title=title,text_auto=True)
```



On Friday We Are Seeing lot Of releases

```
title="Distribution of number movies and series relesed in each Month"
px.histogram(df.sort_values('month'),x='month_name',color='type',title=ti
```



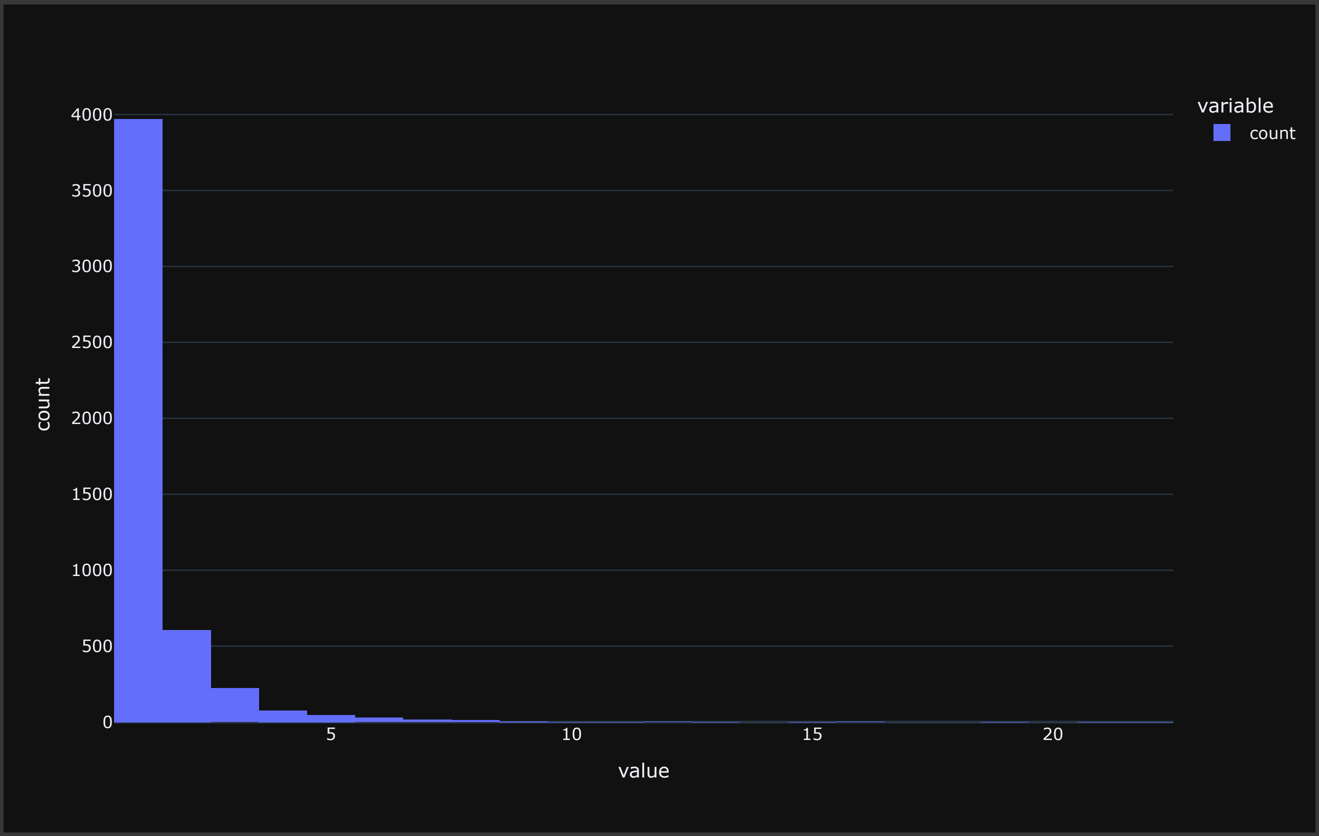
Netflix Is Uploading more in jan,july,dec

```
#Director Mearge
dir_df = director_df.merge(df,on='show_id',how='left')
dir_df.head()
```

	index	director_x	show_id	type	title	director_y	cast	country	date_added	release_year	...	description	duration1	month
0	0	Kirsten Johnson	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No Value	United States	2021-09-25	2020	...	As her father nears the end of his life, filmm...	90 min	9.0
1	1	No Value	s2	TV Show	Blood & Water	No Value	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	...	After crossing paths at a party, a Cape Town t...	2 Seasons	9.0
2	2	Julien Leclercq	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	2021-09-24	2021	...	To protect his family from a powerful drug lor...	1 Season	9.0
3	3	No Value	s4	TV Show	Jailbirds New Orleans	No Value	No Value	United States	2021-09-24	2021	...	Feuds, flirtations and toilet talk go down amo...	1 Season	9.0
4	4	No Value	s5	TV Show	Kota Factory	No Value	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	...	In a city of coaching centers known to train I...	2 Seasons	9.0

5 rows × 23 columns

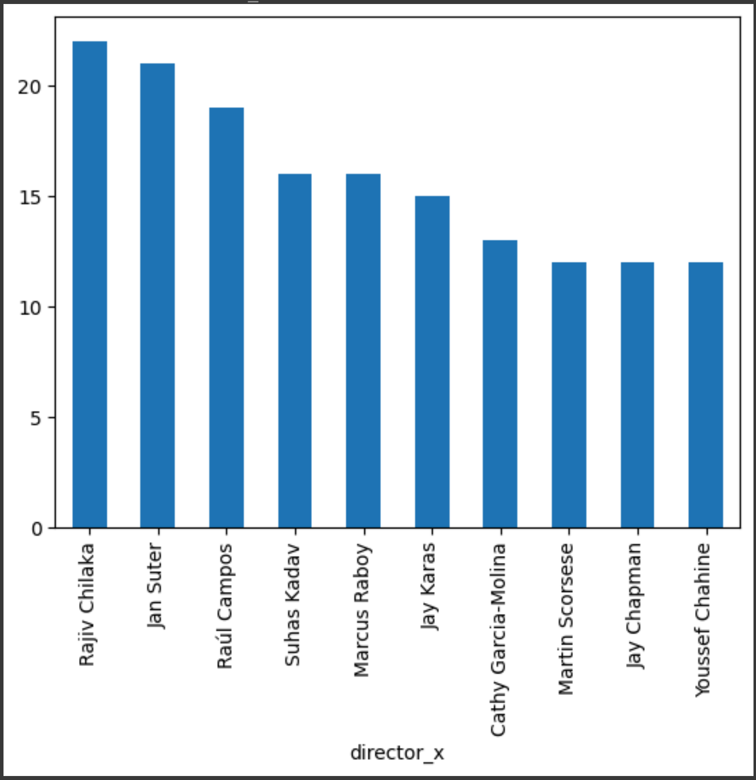
```
#Analysis of actors/directors of different types of shows/movies.
count_dir=dir_df['director_x'].value_counts()
px.histogram(count_dir[1:])
```



Most Of The Directors Directed single Movie or series

```
top_10_dir=count_dir[1:11]
top_10_dir.plot(kind='bar')
```

```
<Axes: xlabel='director_x'>
```



```
top_10_dir_df = dir_df[dir_df['director_x'].isin(top_10_dir.index)]
top_10_dir_df['type'].value_counts()
```

```
            count
     type
    Movie     157
  TV Show       1

dtype: int64
```

```
Start coding or generate with AI.
```

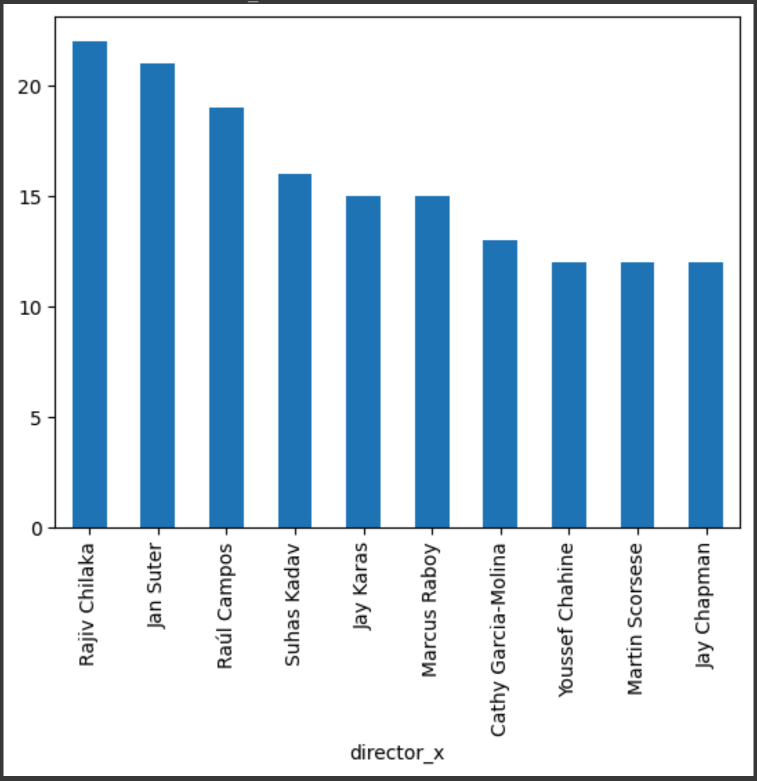
so Needs A seperate Movies Top 10 And Series TOp 10

```
#Movies Top 10 Directors
movies_top_10_dir = dir_df[dir_df['type']=='Movie']['director_x'].value_c
series_top_10_dir = dir_df[dir_df['type']=='TV Show']['director_x'].value
```

```
movies_top_dir_df = dir_df[dir_df['director_x'].isin(movies_top_10_dir.ir
series_top_dir_df = dir_df[dir_df['director_x'].isin(series_top_10_dir.ir
```

```
movies_top_10_dir.plot(kind='bar')
```

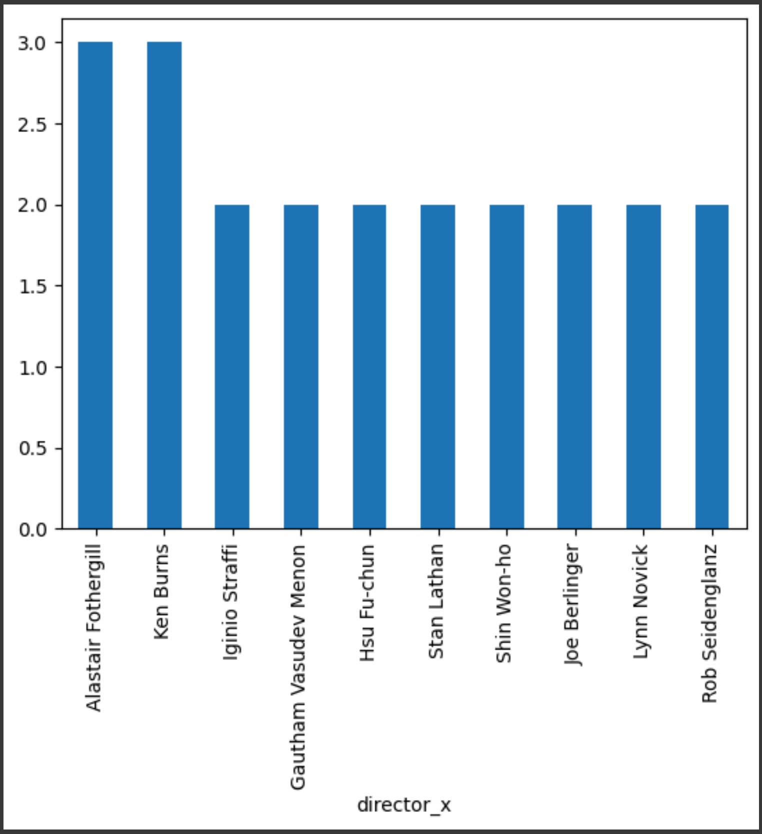
```
<Axes: xlabel='director_x'>
```



Top Directors

```
series_top_10_dir.plot(kind='bar')
print((series['director']=='No Value').sum(),(movies['director']=='No Va
```

```
2446 188
```

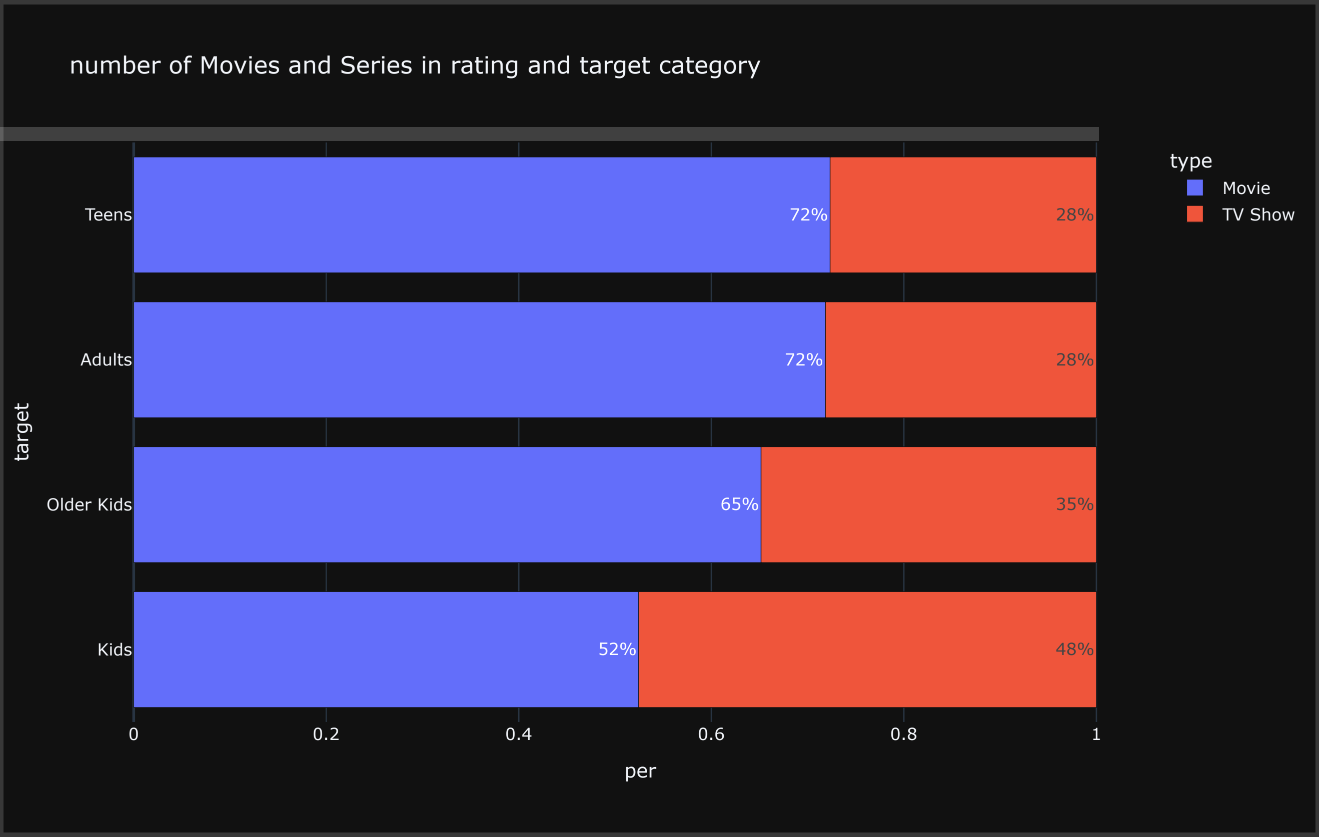


lot of null values so above may not be true

Start coding or generate with AI.

```
#rating and target
title = "number of Movies and Series in rating and target category"
a = df.groupby(['type','target'],as_index=False)['rating'].count()
```

```
a['sum'] = a.groupby(['target'])['rating'].transform('sum')
a['per']=(a['rating']/a['sum'])
px.bar(a.sort_values(['type','per']),x='per',y='target',color='type',text
```


number of Movies and Series in rating and target category

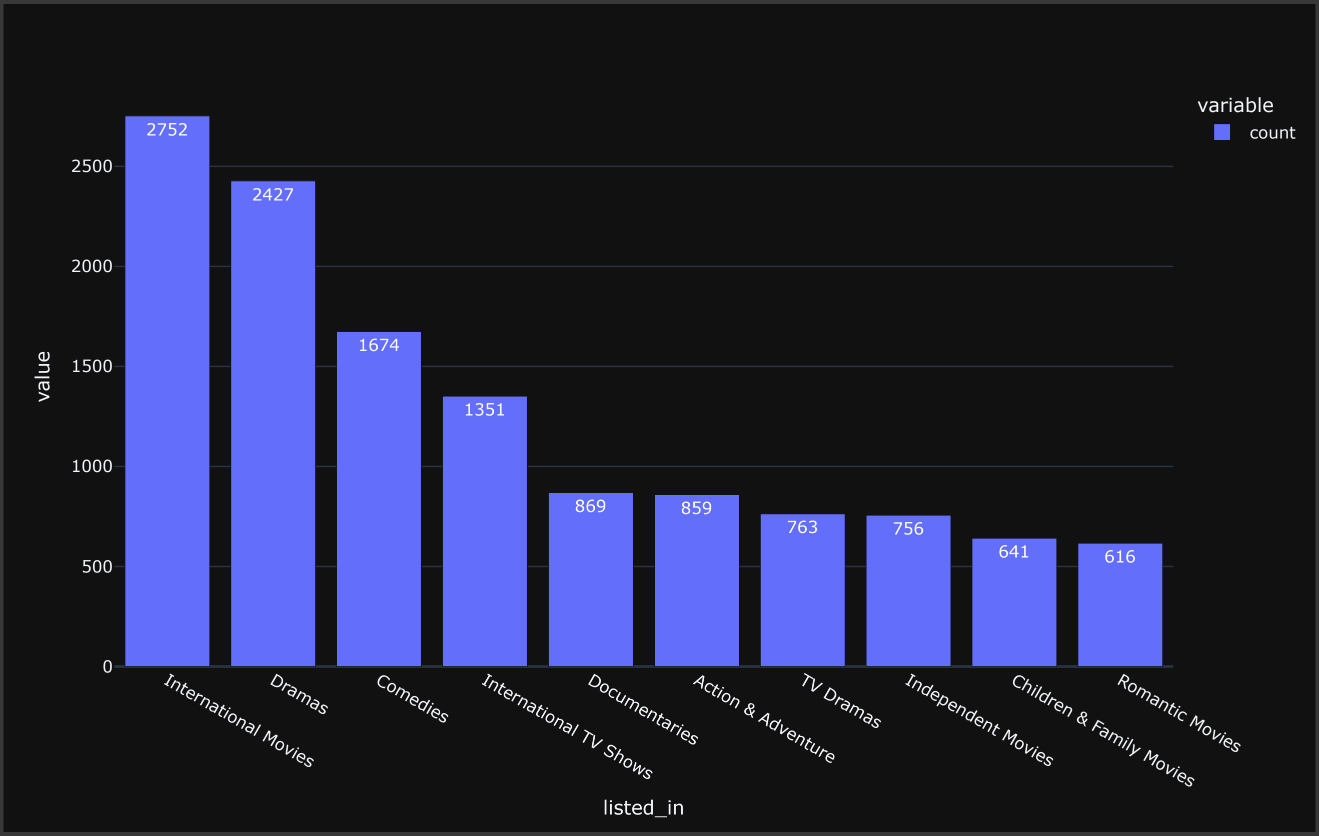
- We Have more adult and teen movies and series
- In Kids category series are more common than movies

```
count_genre = listed_df['listed_in'].value_counts()
count_genre.head()
```

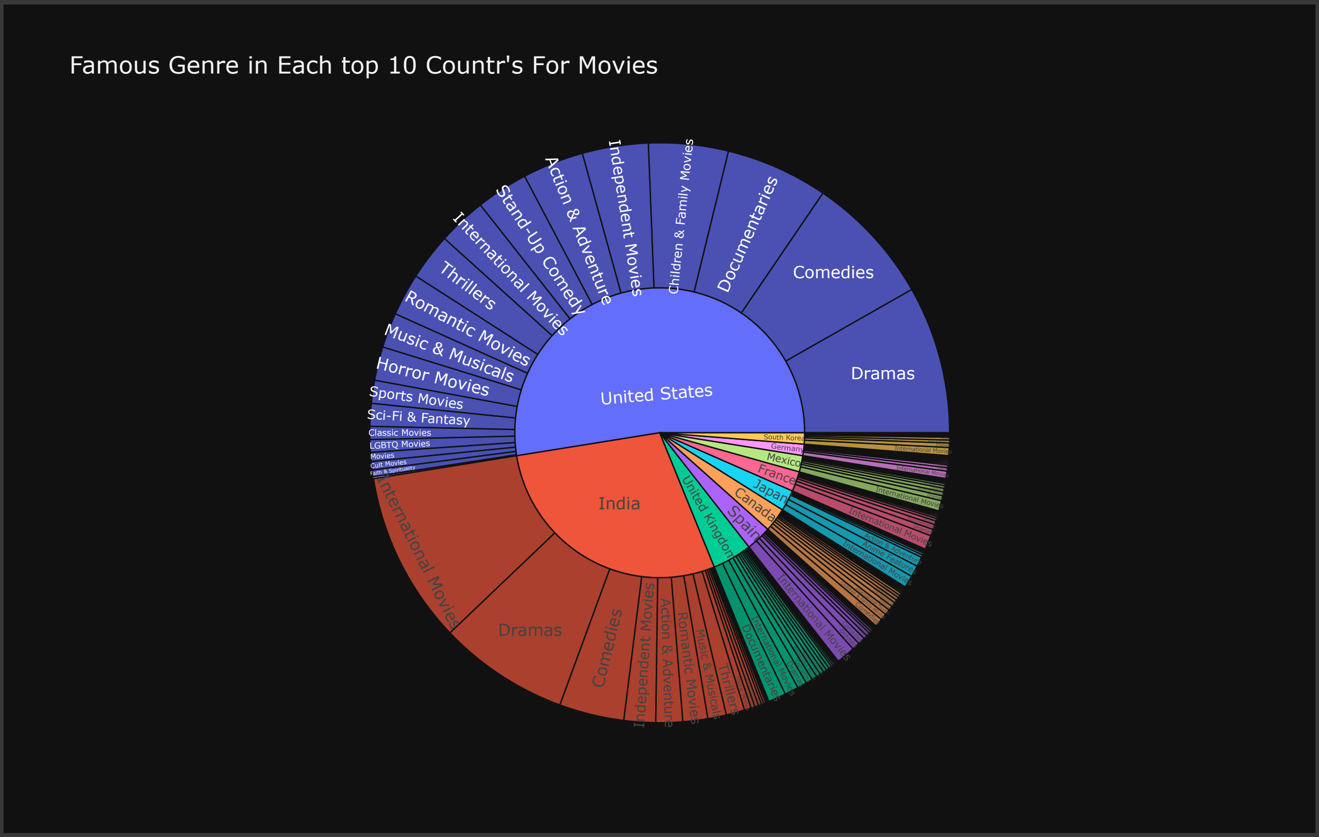
listed_in	count
International Movies	2752
Dramas	2427
Comedies	1674
International TV Shows	1351
Documentaries	869

dtype: int64

```
px.bar(count_genre[:10],text_auto=True)
```

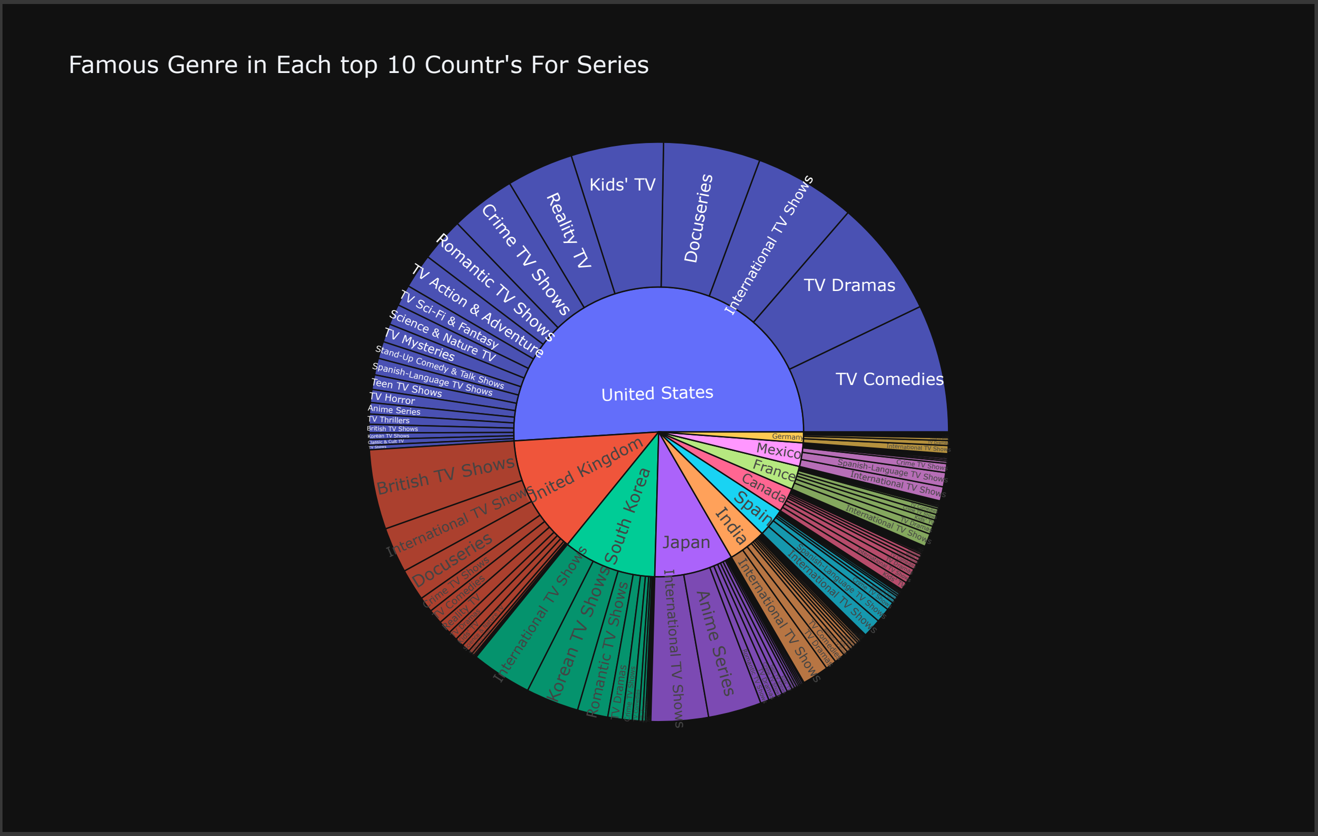


```
title = "Famous Genre in Each top 10 Countr's For Movies"
a = listed_df[(listed_df['type']=='Movie') & (listed_df['country'].isin(t
px.sunburst(a,path=['country','listed_in'],values='title',title=title)
```



- International Movies and Drams are popular in India
- Dramas Comedy and Documentary are Popular Genre In US

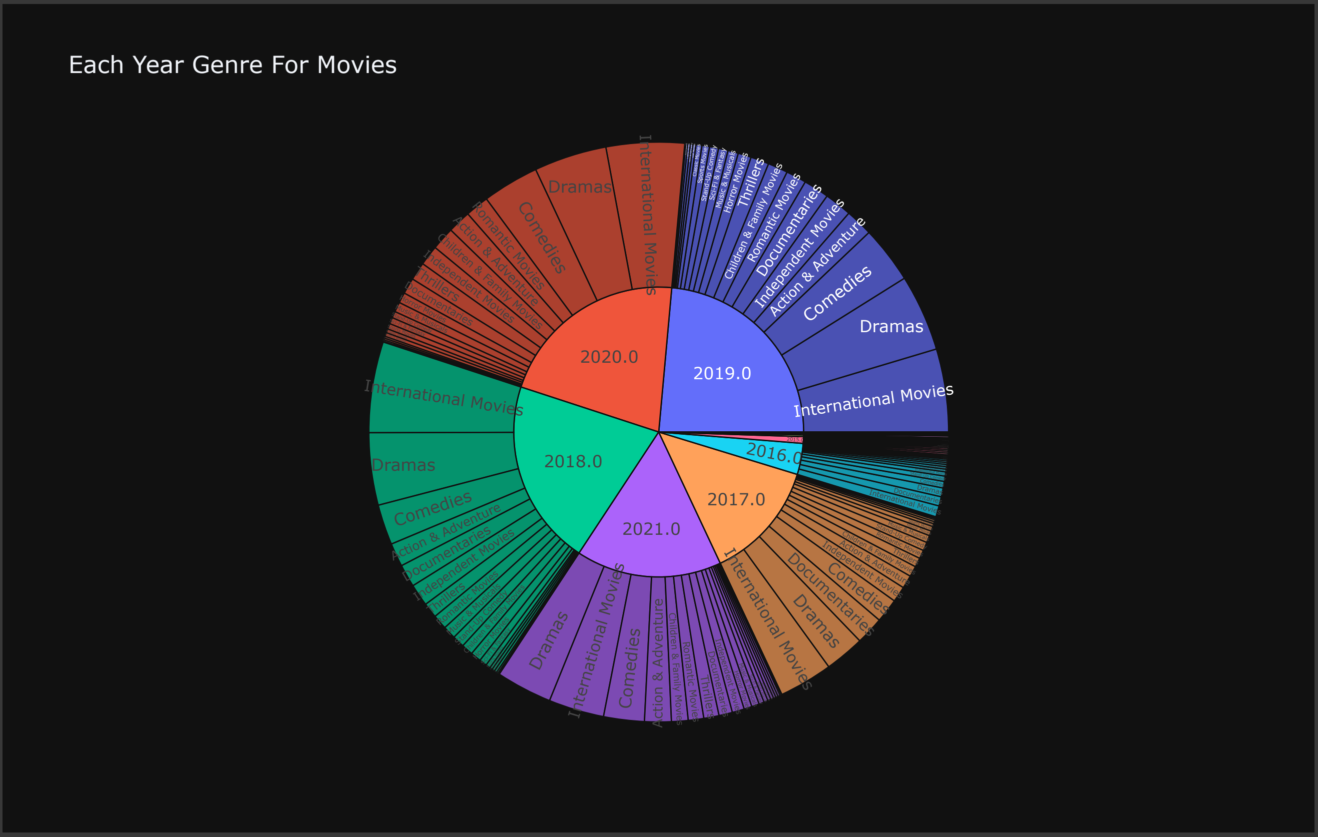
```
title = "Famous Genre in Each top 10 Countr's For Series"
a = listed_df[(listed_df['type']=='TV Show') & (listed_df['country'].isir
px.sunburst(a,path=['country','listed_in'],values='title',title=title)
```



Famous Genre in Each top 10 Countr's For Series

- TV Comedies and Dramas are Popular in US
- Brtish TV Shows in UK

```
title = "Each Year Genre For Movies"
a = listed_df[listed_df['type']=='Movie'].groupby(['year','listed_in'])['
px.sunburst(a,path=['year','listed_in'],values='title',title=title)
```

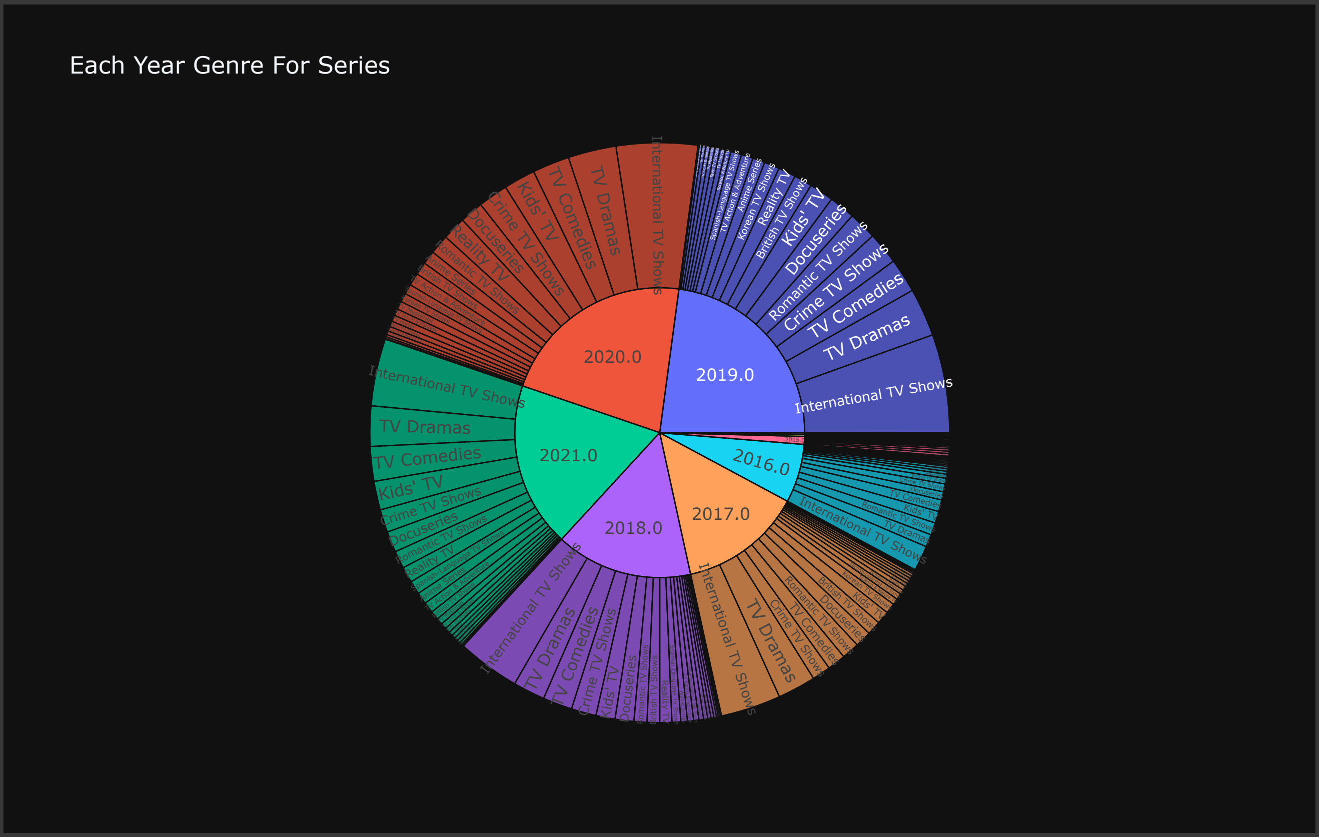


Each Year Genre For Movies

- International Movies
- Drama
- Comedy
- Action and Adventures

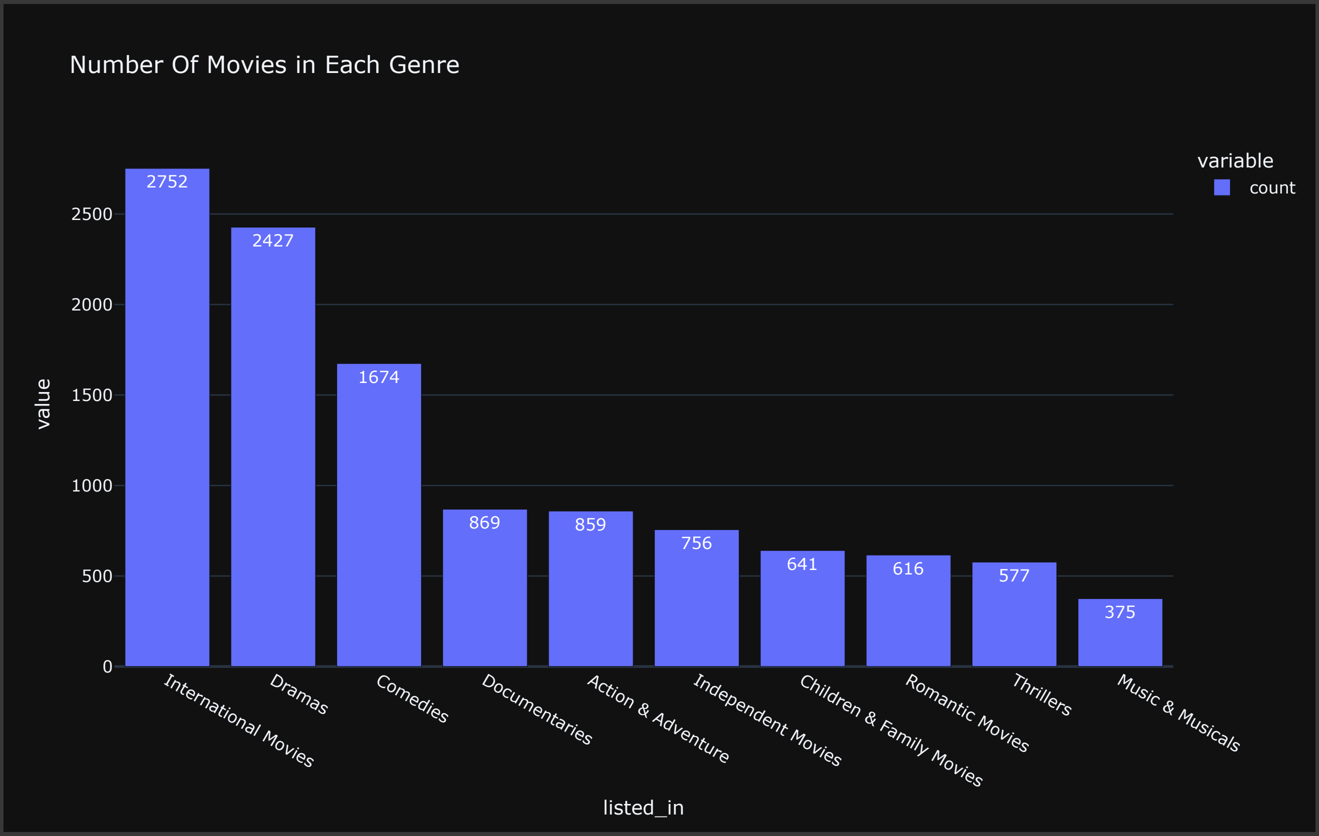
Are popular in Movies all Years

```
title = "Each Year Genre For Series"
a = listed_df[listed_df['type']=='TV Show'].groupby(['year','listed_in']
px.sunburst(a,path=['year','listed_in'],values='title',title=title)
```

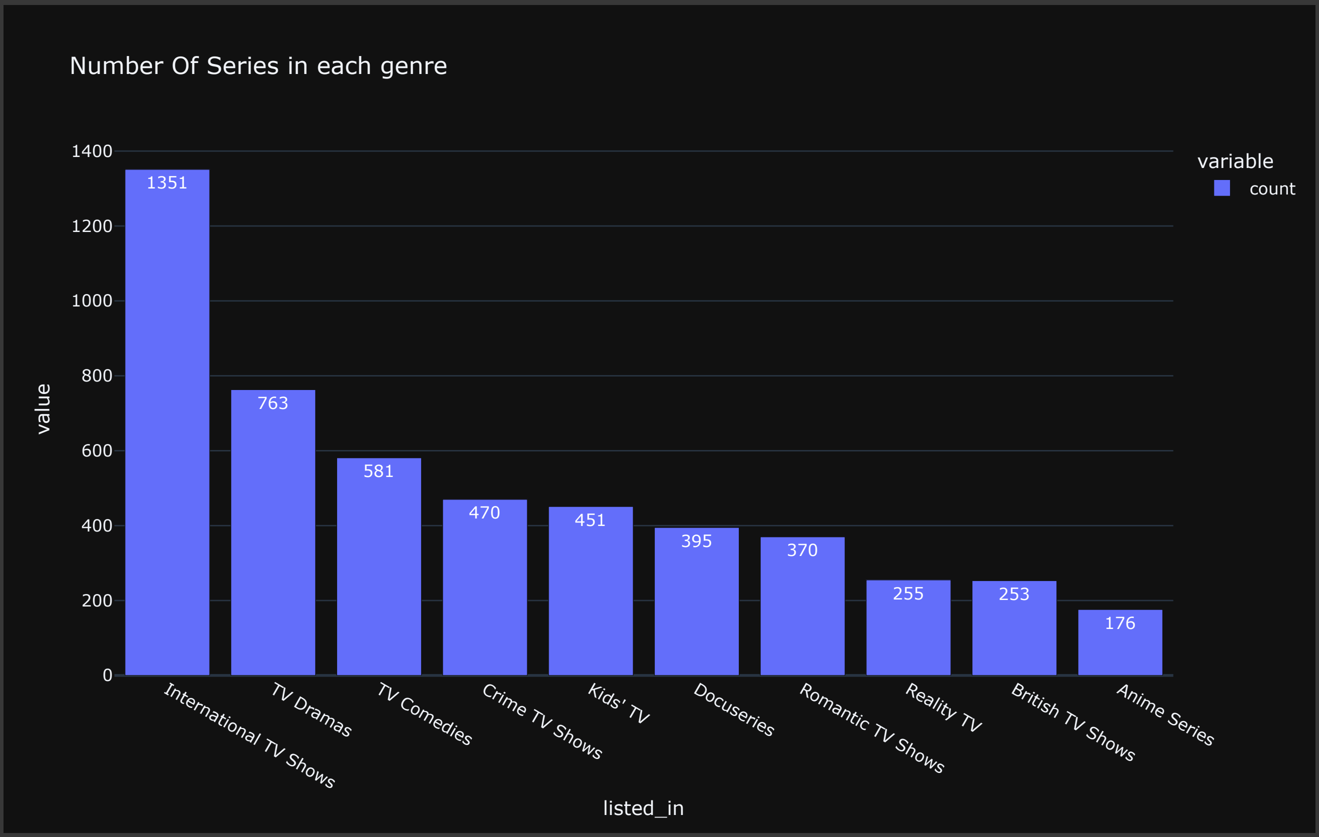


- International TV Shows
- TV Dramas
- TV comedys
- TV Kinds Are Popouler in Series All Years

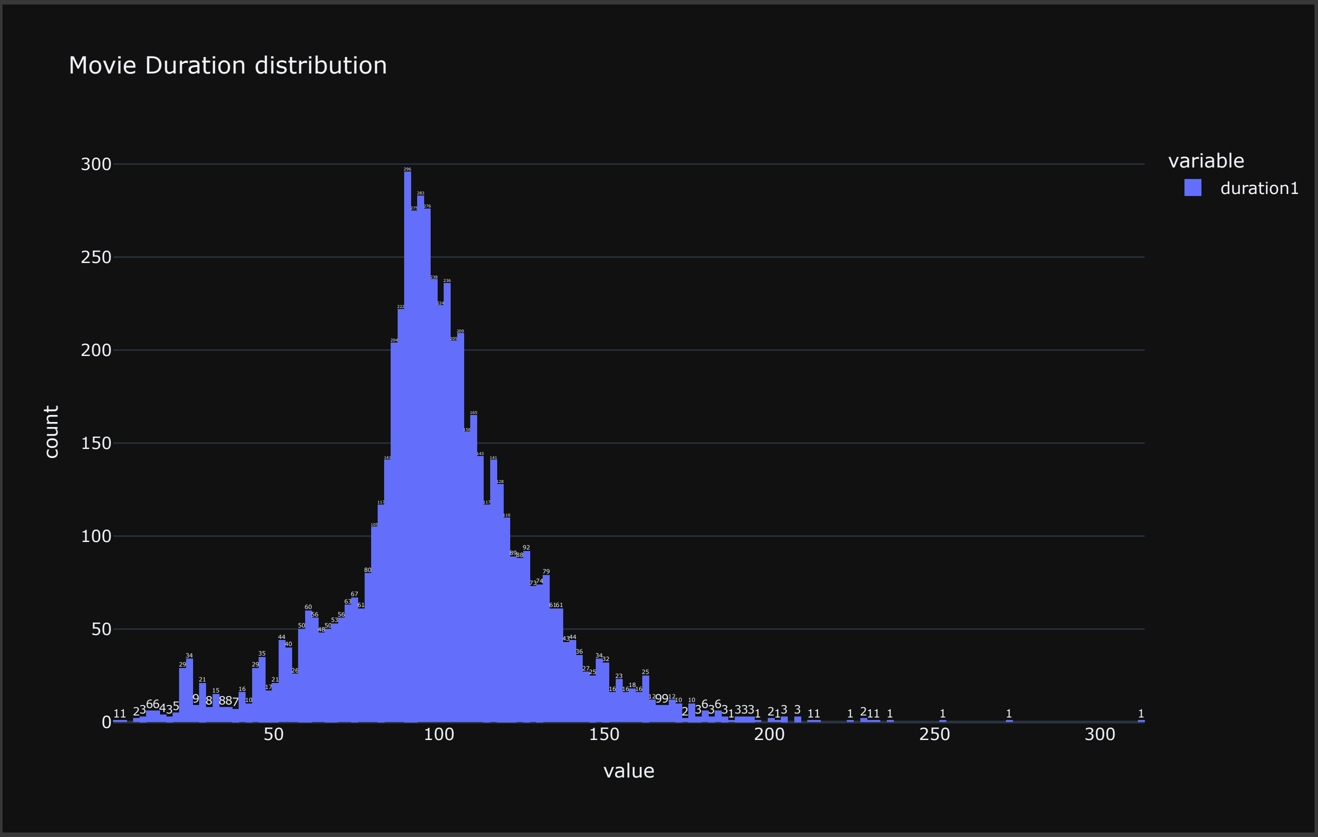
```
title = "Number Of Movies in Each Genre"
movies_genre = df[df['type']=='Movie']['listed_in'].str.replace(', ',','
top_movies_genre = movies_genre.value_counts()[:10]
px.bar(top_movies_genre,text_auto=True,title=title)
```


Number Of Movies in Each Genre

```
title = "Number Of Series in each genre"
series_genre = df[df['type']=='TV Show']['listed_in'].str.replace(', ',','
top_series_genre = series_genre.value_counts()[:10]
px.bar(top_series_genre,text_auto=True,title=title)
```

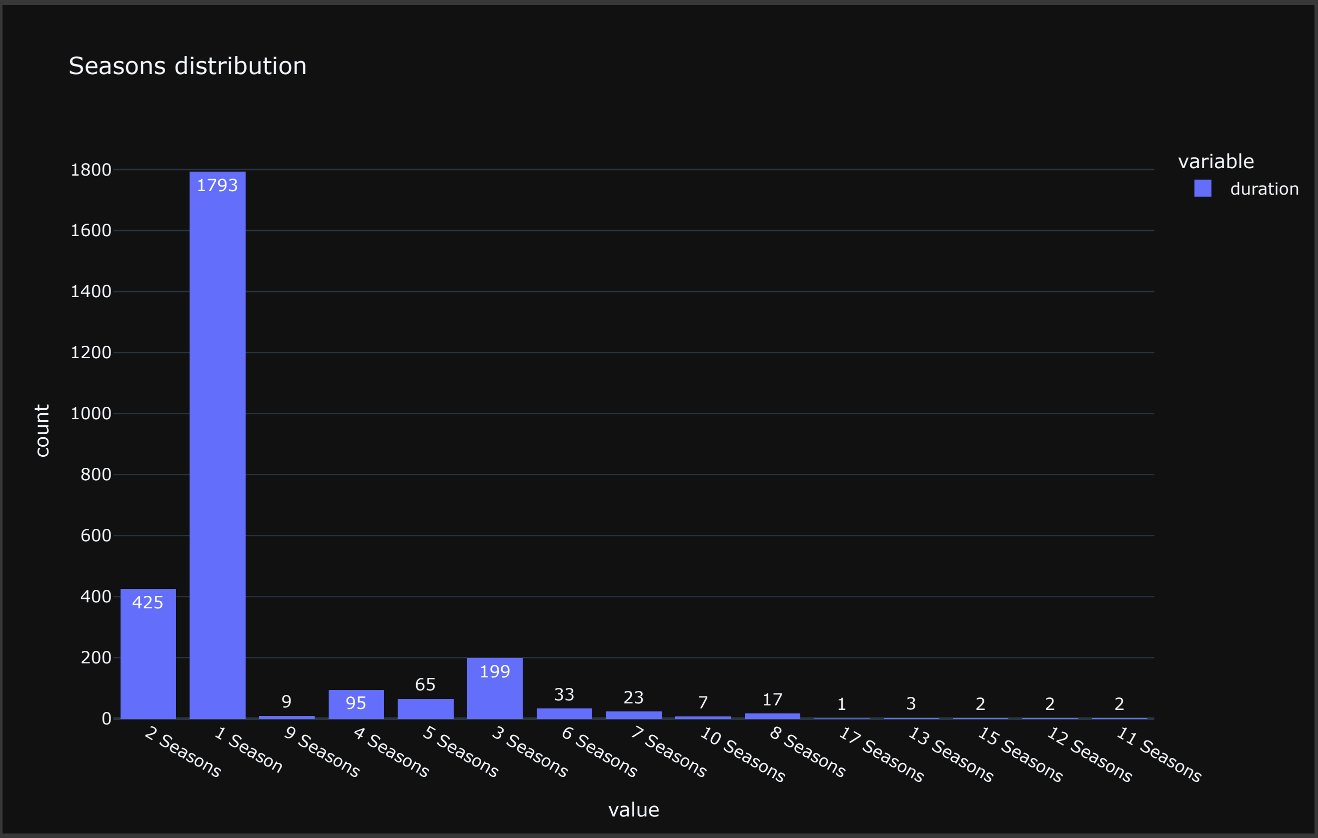

Number Of Series in each genre

```
#duration
title = "Movie Duration distribution"
movies['duration1'] = movies['duration'].str.split(' ').apply(lambda x:in
px.histogram(movies['duration1'],text_auto=True,title=title)
```



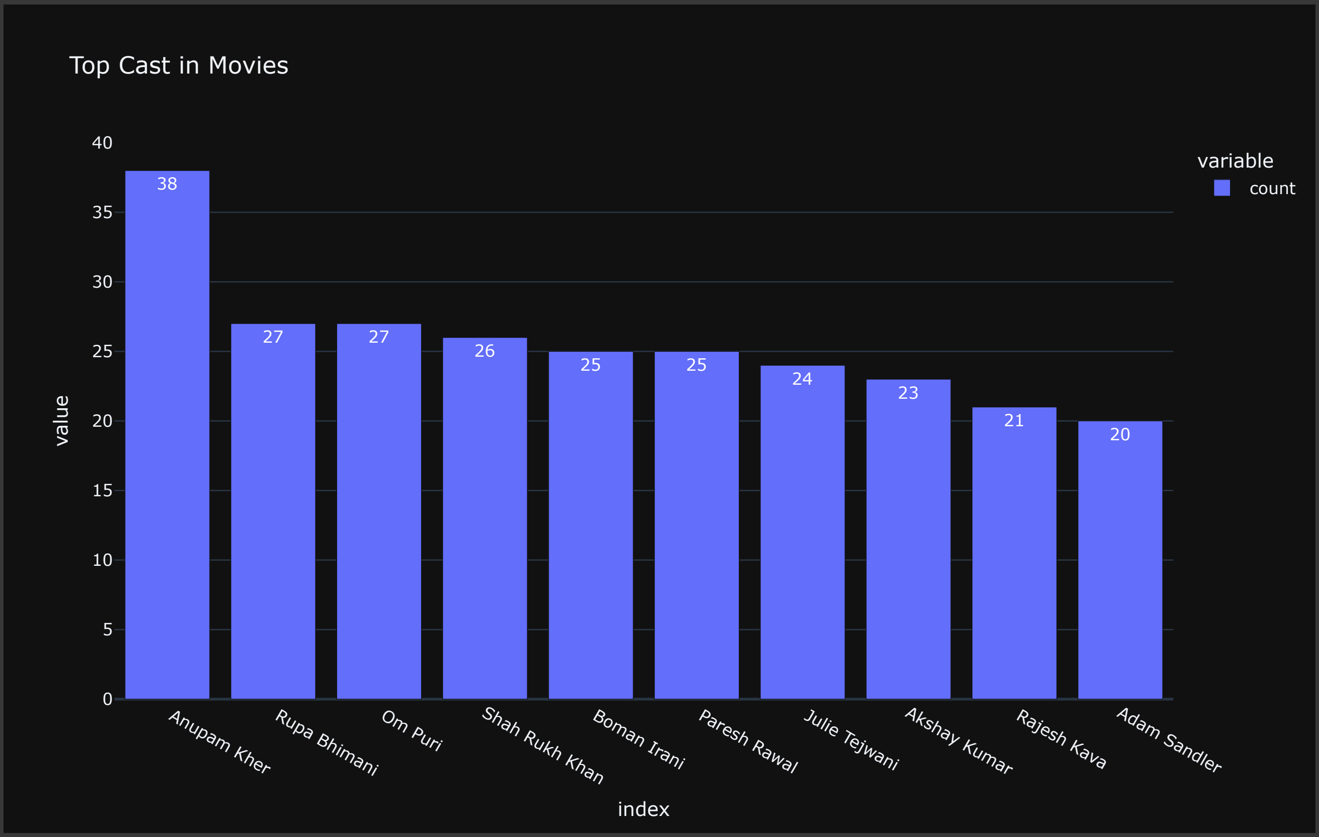
More Movies Are more Than around 100 min

```
title = "Seasons distribution"
px.histogram(series['duration'],text_auto=True,title=title)
```

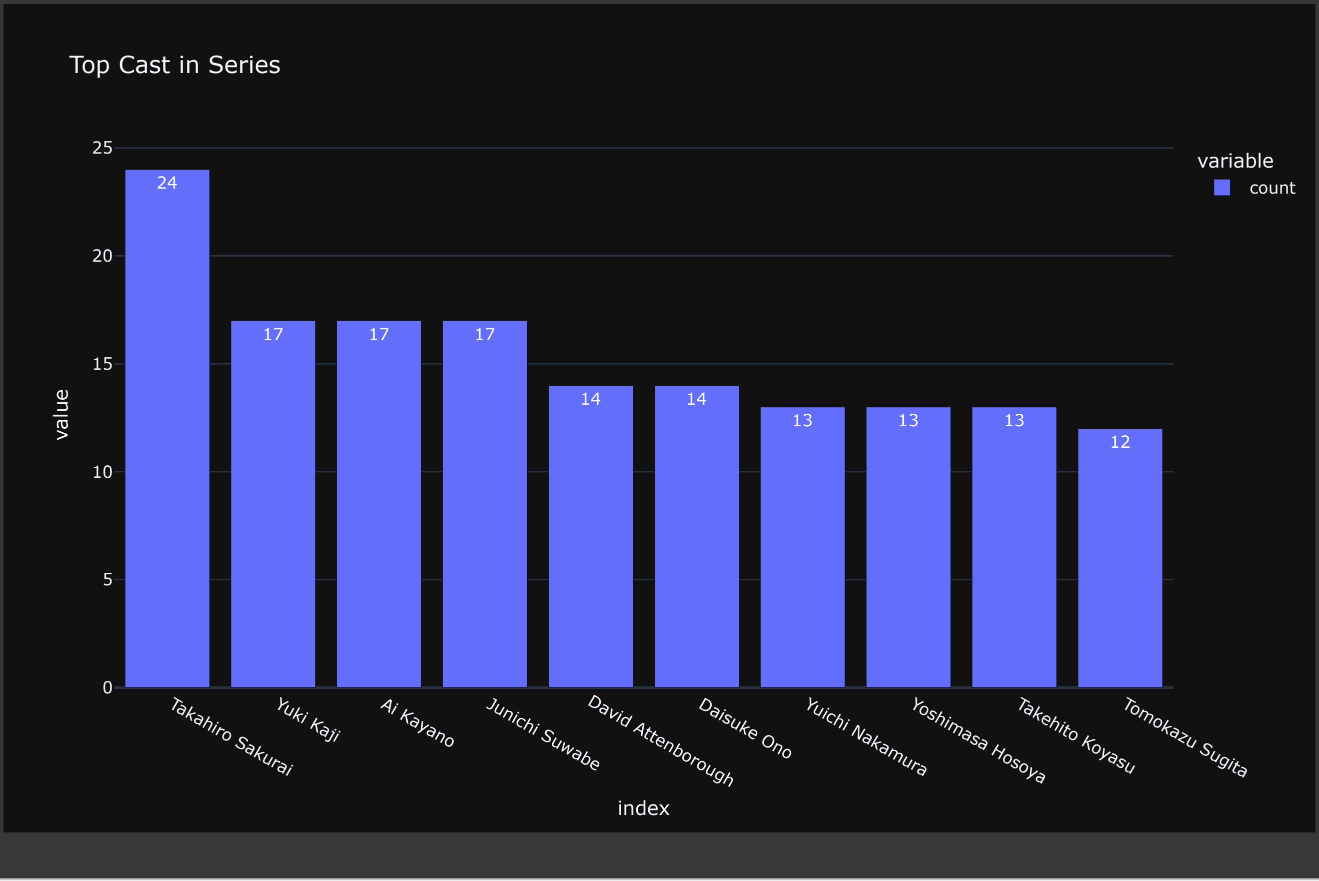


Most number of series are 1,2,3,4 seasons long

```
#Cast
cast_list = df[df['type']=='Movie']['cast'].dropna().str.split(',').sum()
top_cast = pd.Series(cast_list).value_counts()[1:11]
title = "Top Cast in Movies"
px.bar(top_cast,title = title,text_auto=True)
```



```
#Cast
cast_list = df[df['type']=='TV Show']['cast'].dropna().str.split(',').sum
top_cast = pd.Series(cast_list).value_counts()[1:11]
title = "Top Cast in Series"
px.bar(top_cast,title = title,text_auto=True)
```



Insights 📊

1. **Content Composition:**
   - Netflix hosts more **movies (70%)** than TV shows (30%), indicating a preference for single-consumption content.
2. **Peak Additions:**
   - Content additions **peaked in 2020**, likely influenced by the COVID-19 pandemic.
3. **Regional Dominance:**
   - The **USA contributes 60%** of the content, highlighting a Western-centric focus.
4. **Popular Genres:**
   - The most common genres include **Drama, Comedy, and International TV Shows**, appealing to a broad audience.
5. **Audience Focus:**
   - "TV-MA" rated content dominates, reflecting a strong **adult audience base**.
6. **Duration Trends:**
   - Movies typically range between **90-120 minutes**, while most TV shows have **1-2 seasons**.
7. **Korean Influence:**
   - South Korea stands out in **Thriller and Romance**, driven by the global rise of K-dramas.
8. **Content Strategy Shift:**
   - Recent additions include more **family-friendly titles**, signaling a strategic shift.
9. **Content Era:**
   - A majority of the content originates from the **2010s**, indicating a modern content focus.

Recommendations 💡

1. **Target Emerging Markets:**
   - Increase investment in rapidly growing markets like **South Korea and India**.
2. **Expand TV Show Offerings:**
   - Develop more episodic content to enhance **viewer retention** and engagement.
3. **Genre Diversification:**
   - Focus on underrepresented genres to attract **niche audiences** and expand reach.

4. **Family-Oriented Content:**
    - Strengthen the library with more **kid-friendly and family-oriented shows** to attract diverse age groups.

```
print("Thank You")
```

```
Thank You
```

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.