```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
To: /content/netflix.csv
100% 3.40M/3.40M [00:01<00:00, 2.64MB/s]
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('netflix.csv')
df
```

	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...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

```
2# Observations on the shape of data
df.shape
```

```
(8807, 12)
```

OBSERVATION: Given data has 8807 rows and 12 attributes/columns

```
# data types of all the attributes
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
```

All the columns are of the type object expect one that is release_year which is integer type

```
#2.3missing value detection
df.isnull().sum()
```

```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

In above data shows null values present in the dataframe, the highest being in the director column with 2634 missing directors name.

```
df.iloc[]
```

	count
rating	
TV-MA	3207
TV-14	2160
TV-PG	863
R	799
PG-13	490
TV-Y7	334
TV-Y	307
PG	287
TV-G	220
NR	80
G	41
TV-Y7-FV	6
NC-17	3
UR	3
74 min	1
84 min	1
66 min	1

dtype: int64

```
#2.4conversion of categorical attributes to 'category'
df['type'] = df['type'].astype('category')
df['rating'] =df['rating'].astype('category')
```

```
#2.5  statistical summary
num_stat1 = df['release_year'].describe()
num_stat1
```

```
count    8807.000000
mean     2014.180198
std         8.819312
min      1925.000000
```

```
25%      2013.000000
50%      2017.000000
75%      2019.000000
max      2021.000000
Name: release_year, dtype: float64
```

```
num_stat2= df['duration'].describe()
num_stat2
```

```
count         8804
unique         220
top       1 Season
freq          1793
Name: duration, dtype: object
```

OBSERVATION: there were total of 1793 Tv shows that aired in netflix as per the data.

Double-click (or enter) to edit

3. Non-Graphical Analysis: Value counts and unique attributes

```
#Non-Graphical Analysis: Value counts and unique attributes
cast_df = df[["title",'cast','show_id','duration','rating']]
cast_df["list_of_cast"] = cast_df['cast'].apply(lambda x: str(x).split(", "))
cast_df = cast_df.explode("list_of_cast")
cast_df
```

```
<ipython-input-43-e53d8f87ea45>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  cast_df["list_of_cast"] = cast_df['cast'].apply(lambda x: str(x).split(", "))
```

	title	cast	show_id	duration	rating	list_of_cast
0	Dick Johnson Is Dead	NaN	s1	90 min	PG-13	nan
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Ama Qamata
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Khosi Ngema
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Gail Mabalane
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Thabang Molaba
...	...	...	...	...	...	...
8806	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	s8807	111 min	TV-14	Manish Chaudhary
8806	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	s8807	111 min	TV-14	Meghna Malik
8806	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	s8807	111 min	TV-14	Malkeet Rauni
8806	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	s8807	111 min	TV-14	Anita Shabdish
8806	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	s8807	111 min	TV-14	Chittaranjan Tripathy

64951 rows × 6 columns

```
cast_df['list_of_cast'].value_counts()
```

```
nan                        825
Anupam Kher                 43
Shah Rukh Khan              35
Julie Tejwani               33
Naseeruddin Shah            32
                          ...
Melanie Straub               1
Gabriela Maria Schmeide      1
Helena Zengel                1
Daniel Valenzuela            1
Chittaranjan Tripathy        1
Name: list_of_cast, Length: 36440, dtype: int64
```

OBSERVATION: Anupam kher is the most popular actor with 43 projects

```
country_df = df[["title",'country','show_id']]
country_df["list_of_countries"] = country_df['country'].apply(lambda x: str(x).split(", "))
country_df = country_df.explode("list_of_countries")
country_df['list_of_countries'].nunique()
```

```
<ipython-input-13-f21f9f4577d4>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  country_df["list_of_countries"] = country_df['country'].apply(lambda x: str(x).split(", "))
128
```

OBSERVATION Netflix was streamed in 128 countries

```
merge1 = pd.merge(cast_df,country_df, left_on ='show_id',right_on='show_id', how= 'left')
merge1
```

	title_x	cast	show_id	duration	rating	list_of_cast	title_y	country	list_of_countries
0	Dick Johnson Is Dead	NaN	s1	90 min	PG-13	nan	Dick Johnson Is Dead	United States	United States
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Ama Qamata	Blood & Water	South Africa	South Africa
2	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Khosi Ngema	Blood & Water	South Africa	South Africa
3	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Gail Mabalane	Blood & Water	South Africa	South Africa
4	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Thabang Molaba	Blood & Water	South Africa	South Africa
...	...	...	...	...	...	...	...	...	...
81703	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	s8807	111 min	TV-14	Manish Chaudhary	Zubaan	India	India
81704	Zubaan	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	s8807	111 min	TV-14	Meghna Malik	Zubaan	India	India

```
merge1[['list_of_cast','list_of_countries']].value_counts().head(20)
```

```
list_of_cast       list_of_countries
nan                United States        406
                   nan                  154
                   United Kingdom        96
Anupam Kher        India                 40
nan                India                 39
Shah Rukh Khan     India                 34
nan                France                32
Naseeruddin Shah   India                 31
nan                Canada                31
Akshay Kumar       India                 29
Takahiro Sakurai   Japan                 29
Om Puri            India                 29
Amitabh Bachchan   India                 28
Paresh Rawal       India                 28
Yuki Kaji          Japan                 28
Boman Irani        India                 27
Julie Tejwani      nan                   26
Rupa Bhimani       nan                   25
Kareena Kapoor     India                 25
nan                Spain                 23
dtype: int64
```

OBSERVATION: In INDIA Anupam kher is the most popular actor with 40 projects.

```
directors_df = df[["title",'director','show_id']]
directors_df["list_of_directors"] = directors_df['director'].apply(lambda x: str(x).split(", "))
directors_df = directors_df.explode("list_of_directors")
directors_df
```

```
<ipython-input-16-33785e7a0cae>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  directors_df["list_of_directors"] = directors_df['director'].apply(lambda x: str(x).split(", "))
```

	title	director	show_id	list_of_directors
0	Dick Johnson Is Dead	Kirsten Johnson	s1	Kirsten Johnson
1	Blood & Water	NaN	s2	nan
2	Ganglands	Julien Leclercq	s3	Julien Leclercq
3	Jailbirds New Orleans	NaN	s4	nan
4	Kota Factory	NaN	s5	nan
...	...	...	...	...
8802	Zodiac	David Fincher	s8803	David Fincher
8803	Zombie Dumb	NaN	s8804	nan
8804	Zombieland	Ruben Fleischer	s8805	Ruben Fleischer
8805	Zoom	Peter Hewitt	s8806	Peter Hewitt
8806	Zubaan	Mozez Singh	s8807	Mozez Singh

9612 rows × 4 columns

```
adc_df = pd.merge(merge1,directors_df,on ="show_id",how="inner")
adc_df.drop(columns=["title_y","title"],inplace=True)
adc_df
```

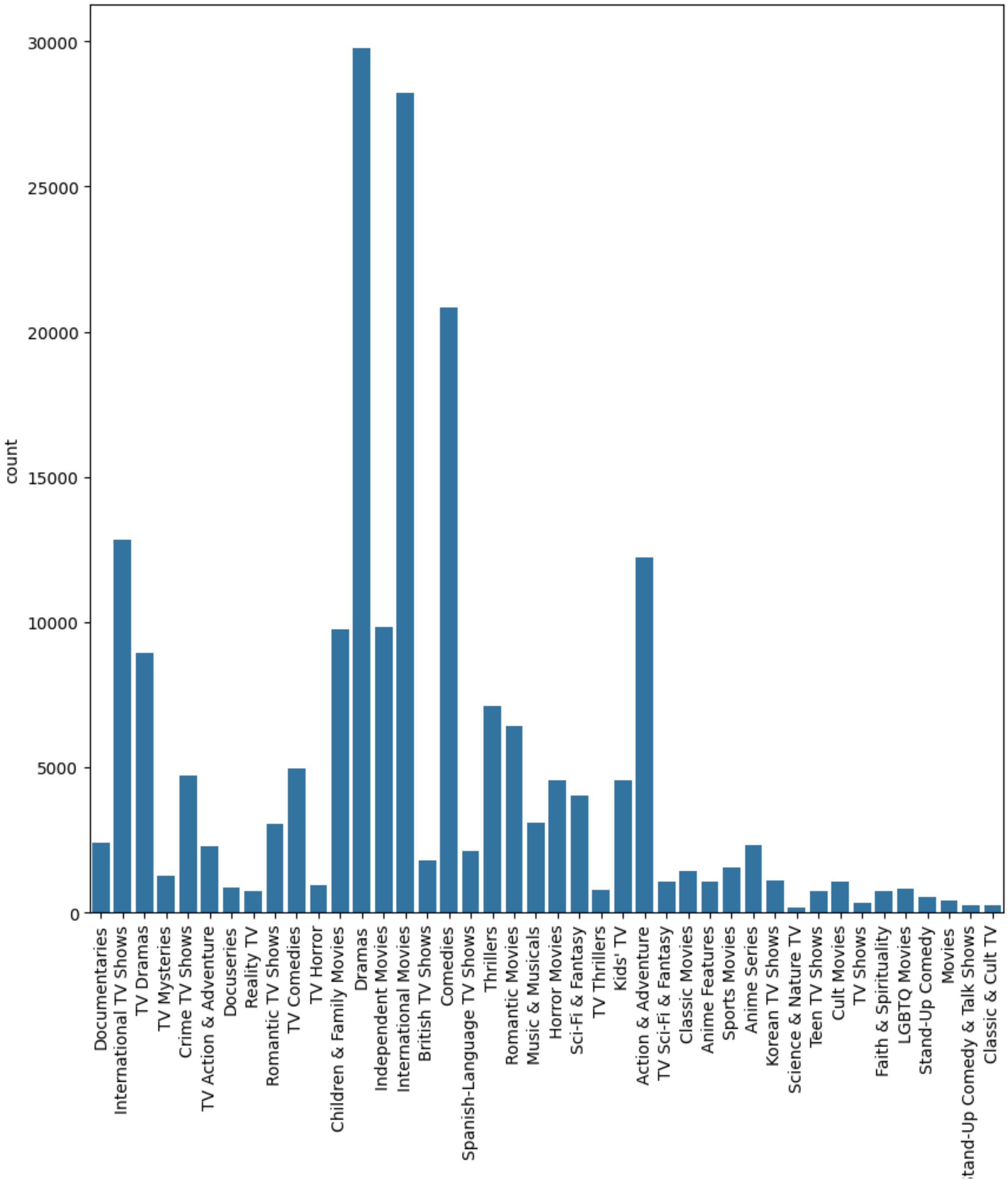
	title_x	cast	show_id	duration	rating	list_of_cast	country	list_of_countries	director	list_of_directors
0	Dick Johnson Is Dead	NaN	s1	90 min	PG-13	nan	United States	United States	Kirsten Johnson	Kirsten Johnson
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Ama Qamata	South Africa	South Africa	NaN	nan
2	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Khosi Ngema	South Africa	South Africa	NaN	nan
3	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Gail Mabalane	South Africa	South Africa	NaN	nan
4	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane	s2	2 Seasons	TV-MA	Thabang Molaba	South Africa	South Africa	NaN	nan

```
genre_df = df[["title",'listed_in','show_id']]
genre_df["list_of_genres"] = genre_df['listed_in'].apply(lambda x: str(x).split(", "))
genre_df = genre_df.explode("list_of_genres")
genre_df

final_df= pd.merge(adc_df,genre_df,on="show_id",how="inner")
final_df.drop(columns='title',inplace=True)
final_df
```

```
<ipython-input-18-ca923058ea20>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cc
  genre_df["list_of_genres"] = genre_df['listed_in'].apply(lambda x: str(x).split(", "))
```

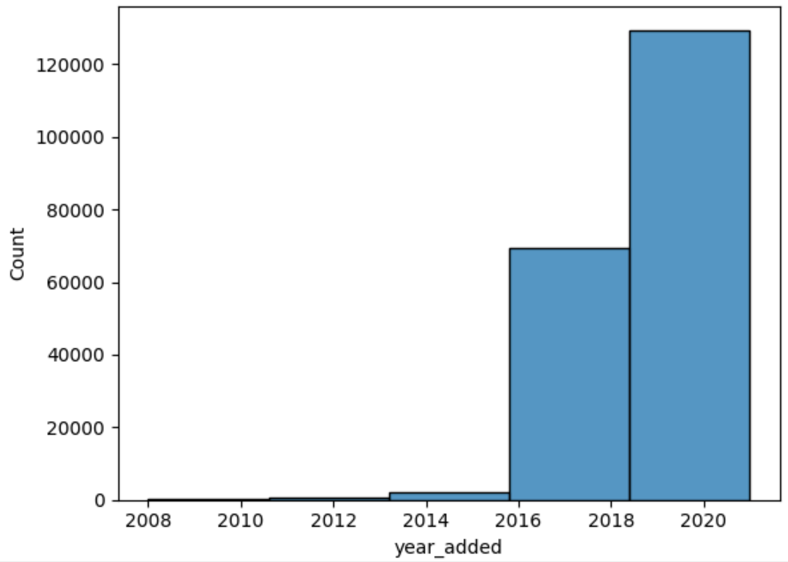
	title_x	cast	show_id	duration	rating	list_of_cast	country	list_of_countries	director	list_of_directors	listed_i
0	Dick Johnson Is Dead	NaN	s1	90 min	PG-13	nan	United States	United States	Kirsten Johnson	Kirsten Johnson	Documentarie
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Ama Qamata	South Africa	South Africa	NaN	nan	Internationa TV Shows, T Dramas, T Mysterie
2	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Ama Qamata	South Africa	South Africa	NaN	nan	Internationa TV Shows, T Dramas, T Mysterie
3	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2	2 Seasons	TV-MA	Ama Qamata	South Africa	South Africa	NaN	nan	Internationa TV Shows, T Dramas, T Mysterie
4	Blood &	Ama Qamata, Khosi Ngema,	s2	2	TV-MA	Khosi Ngema	South	South Africa	NaN	nan	Internationa TV Shows, T

```
#4.1 For continuous variable(s): Distplot, countplot, histogram for univariate analysis
#Countplot
plt.figure(figsize=(10,10))
sns.countplot(data=final_df,x='list_of_genres')
plt.xticks(rotation=90)
plt.show()
```



OBSERVATION: Top genres are Dramas followed by International movies least being Science & Nature TV , Classic & Cult TV

```
#Histogram
df["date_added"] = pd.to_datetime(df['date_added'])
df['year_added'] = df['date_added'].dt.year
df['month_added'] = df['date_added'].dt.month
date_df = df[['show_id','year_added','month_added']]
final_df1 = pd.merge(final_df,date_df,on ='show_id', how='inner')
sns.histplot(data=final_df1,x='year_added',bins=5)
plt.show()
```



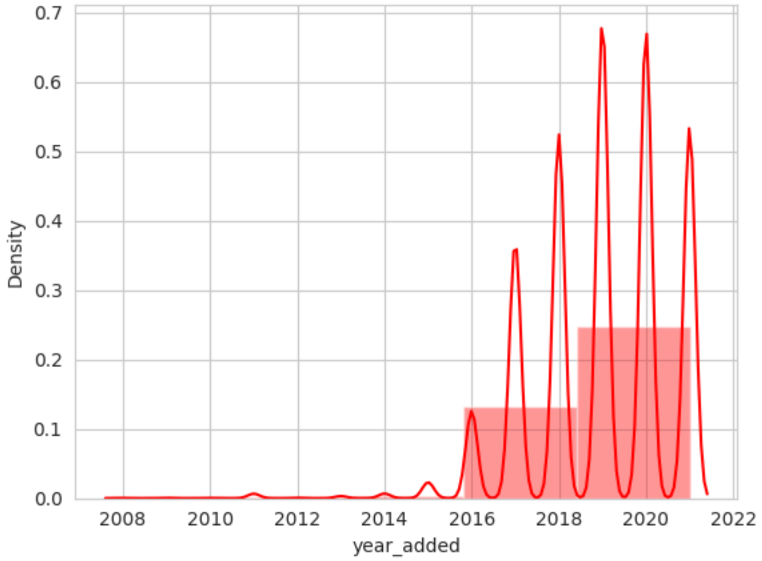
OBSERVATION: The number of shows added were higher in between 2018-2021

```
final_df1['season_count'] = df.apply(lambda x: x['duration'].split(" ")[0] if pd.notna(x['duration']) and "Season" in x['duration'] else "",
final_df1['duration'] = df.apply(lambda x : x['duration'].split(" ")[0] if pd.notna(x['duration']) and "Season" not in x['duration'] else ""
final_df1
```

	title_x	cast	show_id	duration	rating	list_of_cast	country	list_of_countries	director	list_of_directors	listed_i
0	Dick Johnson Is Dead	NaN	s1	90	PG-13	nan	United States	United States	Kirsten Johnson	Kirsten Johnson	Documentarie
1	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2		TV-MA	Ama Qamata	South Africa	South Africa	NaN	nan	Internationa TV Shows, T Dramas, T Mysterie
2	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2		TV-MA	Ama Qamata	South Africa	South Africa	NaN	nan	Internationa TV Shows, T Dramas, T Mysterie
3	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2		TV-MA	Ama Qamata	South Africa	South Africa	NaN	nan	Internationa TV Shows, T Dramas, T Mysterie
4	Blood & Water	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	s2		TV-MA	Khosi Ngema	South Africa	South Africa	NaN	nan	Internationa TV Shows, T Dramas, T Mysterie
...	...	...	...	...	...	...	...	...	...	...	.

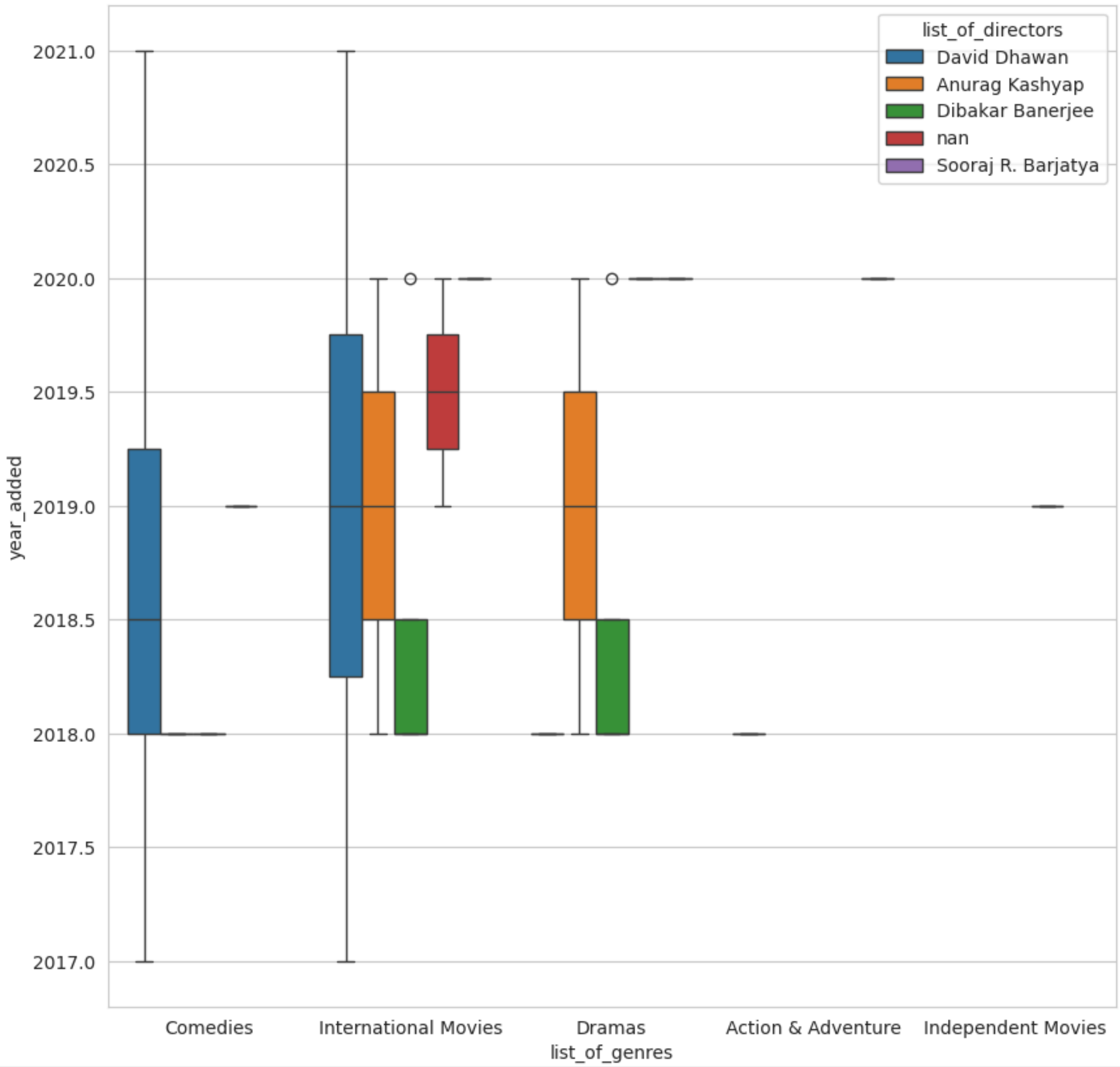
```
#Distplot
sns.set_style('whitegrid')
sns.distplot(final_df1['year_added'], kde = True, color ='red', bins = 5)
plt.show()
```

```
<ipython-input-31-7ad6962ba14c>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(final_df1['year_added'], kde = True, color ='red', bins = 5)
```



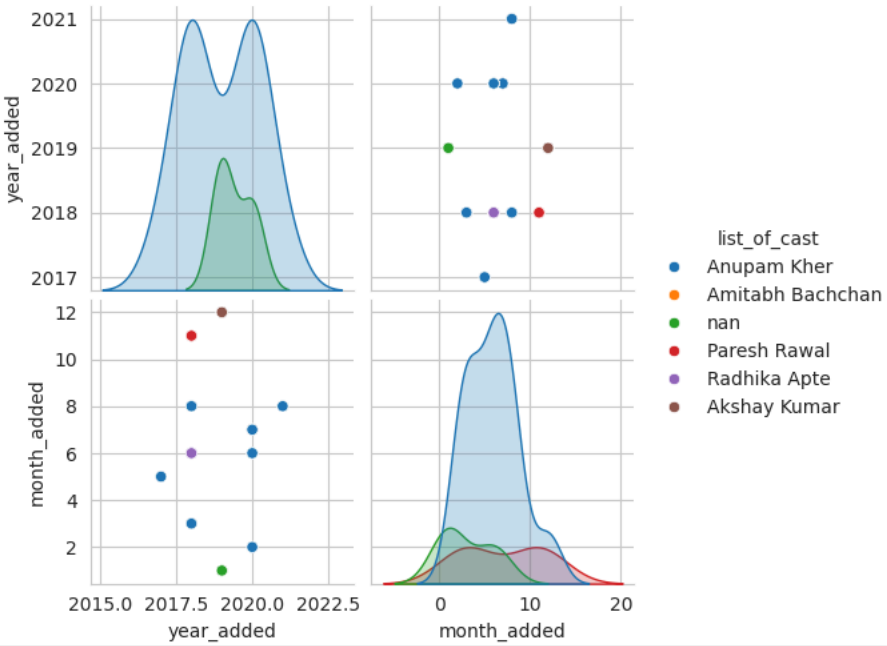
OBSERVATION: The number of shows added were higher in 2019 followed by 2020

```
#BoxPlot
india_df = final_df1[final_df1['list_of_countries']== 'India']
top_dir = india_df['list_of_directors'].value_counts().index[:5]
top_genre= india_df['list_of_genres'].value_counts().index[:5]
top_actors= india_df['list_of_cast'].value_counts().index[:9]
top_data = india_df[(india_df['list_of_directors'].isin(top_dir))&(india_df['list_of_genres'].isin(top_genre))&(india_df['list_of_cast'].isi
plt.figure(figsize=(10,10))

sns.boxplot(data = top_data,x='list_of_genres',y='year_added',hue='list_of_directors')
plt.show()
```



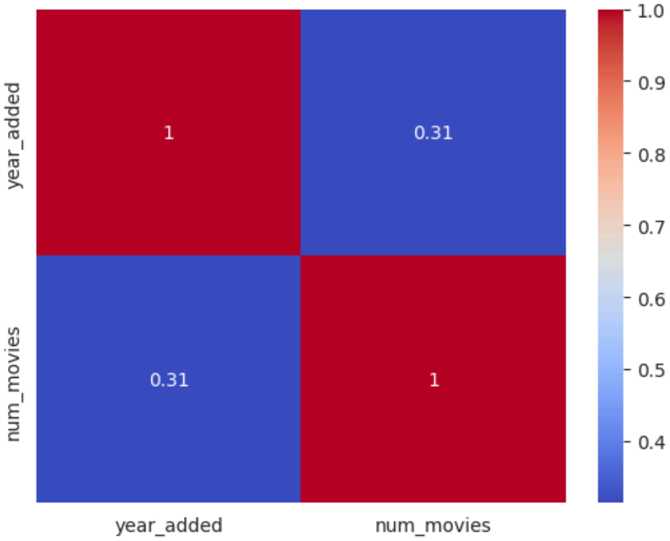
OBSERVATION: David dhawan directed comedy movies were added starting form year 2017 .25% of the total added comedy movies of david dhawan were added by year 2018, 50% by 2018.5, 75% in between 2019-2019.5 and total 100% by 2021.

```
#PAIRPLOT
sns.pairplot(data=top_data,hue="list_of_cast")
plt.show()
```



```
#HEATMAP


year_gen = final_df1.groupby(['list_of_genres', 'year_added']).size().reset_index(name='num_movies')

sns.heatmap(year_gen.corr(),cmap='coolwarm',annot=True)
plt.show()
```

```
<ipython-input-34-f41670f8e39e>:6: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version
  sns.heatmap(year_gen.corr(),cmap='coolwarm',annot=True)
```



OBSERVATION: number of movies/tv shows added through out the years have 0.31 correlation

1. Defining Problem Statement and Analysing basic metrics

Double-click (or enter) to edit

```
#1. Defining Problem Statement and Analysing basic metrics
dir_act = final_df1.groupby(['list_of_directors', 'list_of_cast']).size().reset_index(name='num_movies')
sorted_dir_act = dir_act.sort_values(by='num_movies',ascending=False)
sorted_dir_act.head(20)
```

	list_of_directors	list_of_cast	num_movies
62528	nan	nan	738
51531	nan	David Attenborough	72
61040	nan	Takahiro Sakurai	54
62313	nan	Yuki Kaji	43
62297	nan	Yuichi Nakamura	38
54960	nan	Jun Fukuyama	38
55225	nan	Kate Harbour	37
54999	nan	Junichi Suwabe	37
48811	nan	Ai Kayano	37
51290	nan	Daisuke Ono	36
55032	nan	Justin Fletcher	35
54572	nan	John Sparkes	34
59429	nan	Richard Webber	33
24461	Lars von Trier	Shia LaBeouf	33
53374	nan	Hiroshi Kamiya	33
62231	nan	Yoshimasa Hosoya	33
24466	Lars von Trier	Uma Thurman	33
24465	Lars von Trier	Stellan Skarsgård	33
24456	Lars von Trier	Christian Slater	33
24455	Lars von Trier	Charlotte Gainsbourg	33

```
Idir_act = india_df.groupby(['list_of_directors', 'list_of_cast']).size().reset_index(name='num_movies')
Isorted_dir_act = Idir_act.sort_values(by='num_movies',ascending=False)
Isorted_dir_act.head(20)
```

	list_of_directors	list_of_cast	num_movies
8118	nan	nan	40
1541	David Dhawan	Anupam Kher	18
6525	Sooraj R. Barjatya	Alok Nath	15
6550	Sooraj R. Barjatya	Salman Khan	15
6542	Sooraj R. Barjatya	Mohnish Bahl	12
1577	David Dhawan	Salman Khan	12
4558	Priyadarshan	Rajpal Yadav	11
4358	Prakash Jha	Ajay Devgn	10
2620	Karan Johar	Rani Mukerji	10
7232	Umesh Mehra	Gulshan Grover	10
7245	Umesh Mehra	Mithun Chakraborty	10
6688	Subhash Ghai	Amrish Puri	9
1581	David Dhawan	Shakti Kapoor	9
6547	Sooraj R. Barjatya	Reema Lagoo	9
4552	Priyadarshan	Paresh Rawal	9
2750	Ken Ghosh	Shahid Kapoor	9
3851	Neeraj Pandey	Anupam Kher	9
2203	Hrishikesh Mukherjee	David Abraham	9
2201	Hrishikesh Mukherjee	Asrani	9
7733	Zoya Akhtar	Farhan Akhtar	9

```
act_gen = final_df1.groupby(['list_of_genres', 'list_of_cast']).size().reset_index(name='num_movies')
sorted_act_gen = act_gen.sort_values(by='num_movies',ascending=False)
sorted_act_gen.head(10)
```

	list_of_genres	list_of_cast	num_movies
28796	Documentaries	nan	694
65428	International Movies	nan	334
29285	Docuseries	nan	267
74200	International TV Shows	nan	124
80817	Reality TV	nan	95
27026	Crime TV Shows	nan	89
92258	Sports Movies	nan	85
10133	Children & Family Movies	John Krasinski	66
36827	Dramas	Liam Neeson	65
8437	Children & Family Movies	Alfred Molina	63

```
Iact_gen = india_df.groupby(['list_of_genres', 'list_of_cast']).size().reset_index(name='num_movies')
Isorted_act_gen = Iact_gen.sort_values(by='num_movies',ascending=False)
Isorted_act_gen.head(20)
```

	list_of_genres	list_of_cast	num_movies
7183	International Movies	Anupam Kher	38
9419	International Movies	Shah Rukh Khan	31
4206	Dramas	Naseeruddin Shah	29
6986	International Movies	Akshay Kumar	28
8097	International Movies	Kareena Kapoor	28
4940	Dramas	Shah Rukh Khan	28
8700	International Movies	Om Puri	28
8570	International Movies	Naseeruddin Shah	27
7044	International Movies	Amitabh Bachchan	27
8741	International Movies	Paresh Rawal	27
2963	Dramas	Anupam Kher	26
7436	International Movies	Boman Irani	25
4521	Dramas	Radhika Apte	25
8925	International Movies	Radhika Apte	24
2832	Dramas	Amitabh Bachchan	23
4324	Dramas	Om Puri	21
6965	International Movies	Ajay Devgn	21
4224	Dramas	Nawazuddin Siddiqui	20
9258	International Movies	Salman Khan	20
2659	Documentaries	nan	20

```
con_dur= final_df1.groupby(['list_of_countries', 'duration']).size().reset_index(name='num_movies')
sorted_con_dur = con_dur.sort_values(by='num_movies',ascending=False)

sorted_con_dur.head(20)
```

	list_of_countries	duration	num_movies
1396	United States		816
1583	nan		524
546	India		277
759	Japan		237
1301	United Kingdom		135
168	Canada		86
981	Nigeria		71
876	Mexico		62
356	France		50
1089	South Africa		46
1566	United States	96	45
1565	United States	95	43
1560	United States	90	42
1561	United States	91	41
1738	nan	94	40
67	Australia		40
1563	United States	93	39
1562	United States	92	37
1585	nan	101	37
1398	United States	100	37

OBSERVATION: Director Lars von Trier and actor Shia LaBeouf together they worked on the highest 33 projects and in INDIA David Dhawan and Anupam Kher together did 18 projects . Projects of these pair of director and actors are suggested for NETFLIX. While duration of 95-96 minutes of movies is suggested for United states.

```
#5.Missing Value & Outlier check (Treatment optional)
df.isnull().sum()
```

```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
year_added        10
month_added       10
dtype: int64
```

```
#Treatment
df['country'].fillna('Missing',inplace=True)
df['description'].fillna('Missing',inplace=True)
df['cast'].fillna('Missing',inplace=True)
```

6.1 Comments on the range of attributes

```
Idir_act = india_df.groupby(['list_of_directors', 'list_of_cast']).size().reset_index(name='num_movies')
Isorted_dir_act = Idir_act.sort_values(by='num_movies',ascending=False)
Isorted_dir_act.head()
```

	list_of_directors	list_of_cast	num_movies
8118	nan	nan	40
1541	David Dhawan	Anupam Kher	18
6525	Sooraj R. Barjatya	Alok Nath	15
6550	Sooraj R. Barjatya	Salman Khan	15
6542	Sooraj R. Barjatya	Mohnish Bahl	12

```
Igen_yr = india_df.groupby(['list_of_genres', 'year_added']).size().reset_index(name='num_movies')
Isorted_gen_yr = Igen_yr.sort_values(by='num_movies',ascending=False)
Isorted_gen_yr.head()
```

	list_of_genres	year_added	num_movies
59	International Movies	2018.0	2555
40	Dramas	2018.0	1996
60	International Movies	2019.0	1337
61	International Movies	2020.0	1291
58	International Movies	2017.0	1107

In india director david dhawan and actor anupam kher pair, in generes international movie and dramas are popular.so netflix in india should go for movies which are directed by david dhawan in which anupam kher is playing role which should be a drama or a international movie.

6.2 Comments on the distribution of the variables and relationship between them

```
year_gen = final_df1.groupby(['list_of_genres', 'year_added']).size().reset_index(name='num_movies')
sorted_year_gen = year_gen.sort_values(by='num_movies',ascending=False)
sorted_year_gen.head()
```

	list_of_genres	year_added	num_movies
103	Dramas	2019.0	6758
136	International Movies	2018.0	6466
137	International Movies	2019.0	6418
104	Dramas	2020.0	6202
102	Dramas	2018.0	6170

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

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

OBSERVATION: There are a total of 6758 number of movies/tv shows added in year 2019 which are in drama genre and in year 2018, 6466 were added which were International Movies genre.

```
rating_gen = final_df1.groupby(['list_of_genres', 'rating','list_of_countries','year_added']).size().reset_index(name='num_movies')
sorted_rating_gen = rating_gen.sort_values(by='num_movies',ascending=False)
sorted_rating_gen.head(5)
```

	list_of_genres	rating	list_of_countries	year_added	num_movies
504220	International Movies	TV-14	India	2018.0	1609
382364	Dramas	TV-14	India	2018.0	1144
381581	Dramas	R	United States	2019.0	936
229261	Comedies	R	United States	2019.0	919
504221	International Movies	TV-14	India	2019.0	797

OBSERVATION: Popular among TV-14 rated audience in India in 2018 was international movies followed by dramas. And in united stated among R rated audience in 2019 were drama followed by comedy.

6.3 Comments for each univariate and bivariate plot

1.Distplot: The number of shows/movies added were higher in 2019 followed by 2020.

2.Countplot: Top genres are Dramas followed by International movies least being Science & Nature TV , Classic & Cult TV

3.Histogram: The number of shows added were higher in between 2018-2021

4.Boxplot: In netflix-india David dhawan directed comedy movies were added starting form year 2017 .25% of the total added comedy movies of david dhawan were added by year 2018, 50% by 2018.5, 75% in between 2019-2019.5 and total 100% by 2021.

5.Heatmaps: Number of movies/tv shows added through out the years have 0.31 correlation.

7. Business Insights Performance Analysis

```
rating_gen = final_df1.groupby(['list_of_genres', 'rating','list_of_countries','year_added']).size().reset_index(name='num_movies')
sorted_rating_gen = rating_gen.sort_values(by='num_movies',ascending=False)
sorted_rating_gen.head(5)
```

	list_of_genres	rating	list_of_countries	year_added	num_movies
504220	International Movies	TV-14	India	2018.0	1609
382364	Dramas	TV-14	India	2018.0	1144
381581	Dramas	R	United States	2019.0	936
229261	Comedies	R	United States	2019.0	919
504221	International Movies	TV-14	India	2019.0	797

OBSERVATION: Popular among TV-14 rated audience in India in 2018 and 2019 was international movies followed by dramas. And in united stated among R rated audience in 2019 were drama followed by comedy.

```
Idir_act = india_df.groupby(['list_of_directors', 'list_of_cast']).size().reset_index(name='num_movies')
Isorted_dir_act = Idir_act.sort_values(by='num_movies',ascending=False)
Isorted_dir_act.head()
```

	list_of_directors	list_of_cast	num_movies
8118	nan	nan	40
1541	David Dhawan	Anupam Kher	18
6525	Sooraj R. Barjatya	Alok Nath	15
6550	Sooraj R. Barjatya	Salman Khan	15
6542	Sooraj R. Barjatya	Mohnish Bahl	12

OBSERVATION: In India director david dhawan and actor Anupam kher together worked in 18 projects making the pair as popular director-actor pair in india.

```
con_dur= final_df1.groupby(['list_of_countries', 'duration']).size().reset_index(name='num_movies')
sorted_con_dur = con_dur.sort_values(by='num_movies',ascending=False)

sorted_con_dur.head(20)
```

	list_of_countries	duration	num_movies
1396	United States		816
1583	nan		524
546	India		277
759	Japan		237
1301	United Kingdom		135
168	Canada		86
981	Nigeria		71
876	Mexico		62
356	France		50
1089	South Africa		46
1566	United States	96	45
1565	United States	95	43
1560	United States	90	42
1561	United States	91	41
67	Australia		40

OBSERVATION: duration of 95-96 minutes of movies is suggested for United states.

```
#export data to csv file
```

```
dir_act = final_df1.groupby(['list_of_directors', 'list_of_cast']).size().reset_index(name='num_movies')
sorted_dir_act = dir_act.sort_values(by='num_movies',ascending=False)
sorted_dir_act.head(20)
```

	list_of_directors	list_of_cast	num_movies
62528	nan	nan	738