In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
df=pd.read_csv("netflix.csv")
```

In [4]:
```
# Filtering  Data Column wise so that each row has one director,one genre,o

# 1. Separting Casts in each column
c2=df["cast"].apply(lambda x: str(x).split(', ')).tolist()
df2=pd.DataFrame(c2,index=df['title'])
df2=df2.stack()
df2=pd.DataFrame(df2.reset_index())
df2.rename(columns={0:'Actors'},inplace=True)
df2.drop(['level_1'],axis=1,inplace=True)
df2.head(20)

# 2.Separting Country in each column
c3=df["country"].apply(lambda x:str(x).split(', ')).tolist()
df3=pd.DataFrame(c3,index=df["title"])
df3=df3.stack()
df3=pd.DataFrame(df3.reset_index())
df3.rename(columns={0:"country"},inplace=True)
df3.drop(columns="level_1",axis=1,inplace=True)

# 3. Separting Genres in each column
c4=df["listed_in"].apply(lambda x:str(x).split(', ')).tolist()
df4=pd.DataFrame(c4,index=df["title"])
df4=df4.stack()
df4=pd.DataFrame(df4.reset_index())
df4.rename(columns={0:"Genre"},inplace=True)
df4.drop(columns="level_1",axis=1,inplace=True)

# 5. Separting Directors in each column
c5=df["director"].apply(lambda x:str(x).split(', ')).tolist()
df5=pd.DataFrame(c5,index=df["title"])
df5=df5.stack()
df5=pd.DataFrame(df5.reset_index())
df5.rename(columns={0:"director"},inplace=True)
df5.drop(columns="level_1",axis=1,inplace=True)

#merging actor and country
df_1=pd.merge(df2,df3,on="title",how="inner")
#merging df1 and genre
df_2=pd.merge(df_1,df4,on="title",how="inner")
#merging df2 and director
df_3=pd.merge(df_2,df5,on="title",how="inner")

# Replace NAN values
df_3["director"].replace(["nan"],["Director"],inplace=True)
df_3["Actors"].replace(["nan"],["Actor"],inplace=True)
df_3["country"].replace(["nan"],[np.nan],inplace=True)

# JOining above merged data with original data i.e df
df_final=df3.merge(df[["title","show_id","type","date_added","release_year"

# making new column Month i,e extracting month from Date_added column
df_final["month"]=df_final["date_added"].apply(lambda x:str(x).split(','))
df_final["month"]=df_final["month"].str[-2]
```

In [5]:
```
# now Separting above filtered data in two types i.e MOVIES & TV-SHOWS
def myfunc(data):
    if data== "Movie":
        return 1
    else:
        return 0

df_final["new"]=df_final["type"].apply(myfunc)
# MOVIES
df_final_movies=df_final[df_final["new"]==1]
# TV-SHOWS
df_final_season=df_final[df_final["new"]==0]

# Dropping column new from movies data and TV-SHOW data
df_final_season.drop(columns=["new"],inplace=True)
df_final_movies.drop(columns=["new"],inplace=True)
```

```
C:\Users\aayus\AppData\Local\Temp\ipykernel_12972\1102067653.py:15: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  df_final_season.drop(columns=["new"],inplace=True)
C:\Users\aayus\AppData\Local\Temp\ipykernel_12972\1102067653.py:16: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  df_final_movies.drop(columns=["new"],inplace=True)
```

In [6]:
```
# droping  show_id and date_added column from both filtered movies and tv-s
df_final_movies=df_final_movies.drop(columns=["show_id","date_added"])
df_final_season=df_final_season.drop(columns=["show_id","date_added"])
```

In [86]:
```
df_final_movies
```

Out[86]:

	title	Actors	country	Genre	director	type	release_year	rating
0	Dick Johnson Is Dead	Unknown Actor	United States	Documentaries	Kirsten Johnson	Movie	2020	PG-13
159	My Little Pony: A New Generation	Vanessa Hudgens	NaN	Children & Family Movies	Robert Cullen	Movie	2021	PG
160	My Little Pony: A New Generation	Vanessa Hudgens	NaN	Children & Family Movies	José Luis Ucha	Movie	2021	PG
161	My Little Pony: A New Generation	Kimiko Glenn	NaN	Children & Family Movies	Robert Cullen	Movie	2021	PG
162	My Little Pony: A New Generation	Kimiko Glenn	NaN	Children & Family Movies	José Luis Ucha	Movie	2021	PG
...	...	...	...	...	...	...	...	...
201986	Zubaan	Anita Shabdish	India	International Movies	Mozez Singh	Movie	2015	TV-14
201987	Zubaan	Anita Shabdish	India	Music & Musicals	Mozez Singh	Movie	2015	TV-14
201988	Zubaan	Chittaranjan Tripathy	India	Dramas	Mozez Singh	Movie	2015	TV-14
201989	Zubaan	Chittaranjan Tripathy	India	International Movies	Mozez Singh	Movie	2015	TV-14
201990	Zubaan	Chittaranjan Tripathy	India	Music & Musicals	Mozez Singh	Movie	2015	TV-14

145843 rows × 10 columns

In [87]:
```
df_final_season
```

Out[87]:

	title	Actors	country	Genre	director	type	release_year	rating	durati
1	Blood & Water	Ama Qamata	South Africa	International TV Shows	UnKnown Director	TV Show	2021	TV-MA	Seasc
2	Blood & Water	Ama Qamata	South Africa	TV Dramas	UnKnown Director	TV Show	2021	TV-MA	Seasc
3	Blood & Water	Ama Qamata	South Africa	TV Mysteries	UnKnown Director	TV Show	2021	TV-MA	Seasc
4	Blood & Water	Khosi Ngema	South Africa	International TV Shows	UnKnown Director	TV Show	2021	TV-MA	Seasc
5	Blood & Water	Khosi Ngema	South Africa	TV Dramas	UnKnown Director	TV Show	2021	TV-MA	Seasc
...	...	...	...	...	...	...	...	...	
201864	Zindagi Gulzar Hai	Hina Khawaja Bayat	Pakistan	Romantic TV Shows	UnKnown Director	TV Show	2012	TV-PG	Seas
201865	Zindagi Gulzar Hai	Hina Khawaja Bayat	Pakistan	TV Dramas	UnKnown Director	TV Show	2012	TV-PG	Seas
201932	Zombie Dumb	Unknown Actor	NaN	Kids' TV	UnKnown Director	TV Show	2018	TV-Y7	Seasc
201933	Zombie Dumb	Unknown Actor	NaN	Korean TV Shows	UnKnown Director	TV Show	2018	TV-Y7	Seasc
201934	Zombie Dumb	Unknown Actor	NaN	TV Comedies	UnKnown Director	TV Show	2018	TV-Y7	Seasc

56148 rows × 10 columns

Defining Problem Statement and Analysing basic metrics

PROBLEM STATEMENT : Analyze the data and generate insights that could help Netflix in deciding which type of shows/movies to produce and how they can grow the business in different countries. By seeing the Data of netflix I observed Netflix is one of the most popular media and video streaming platforms. They have over 10000 movies or tv shows available on their platform, as of mid-2021, they have over 222M Subscribers globally. 1.How has the number of movies released per year changed over the last 20-30 years? 2.Comparison of tv shows vs. movies? 3.What is the best time to launch a TV show? 4.Analysis of actors/directors of different types of shows/movies. 5.Does Netflix has more focus on TV Shows than movies in recent years 6.Understanding what content is available in different countries

In [3]:
```
df
```

Out[3]:

	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021
...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015

8807 rows × 12 columns

In [4]:
```
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
```

In [38]:
```
# only Movies data
df_final_movies
```

Out[38]:

	title	Actors	country	Genre	director	show_id	type	date_adde
0	Dick Johnson Is Dead	Unknown Actor	United States	Documentaries	Kirsten Johnson	s1	Movie	Septembe 25, 202
159	My Little Pony: A New Generation	Vanessa Hudgens	NaN	Children & Family Movies	Robert Cullen	s7	Movie	Septembe 24, 202
160	My Little Pony: A New Generation	Vanessa Hudgens	NaN	Children & Family Movies	José Luis Ucha	s7	Movie	Septembe 24, 202
161	My Little Pony: A New Generation	Kimiko Glenn	NaN	Children & Family Movies	Robert Cullen	s7	Movie	Septembe 24, 202
162	My Little Pony: A New Generation	Kimiko Glenn	NaN	Children & Family Movies	José Luis Ucha	s7	Movie	Septembe 24, 202
...	...	...	...	...	...	...	...	.
201986	Zubaan	Anita Shabdish	India	International Movies	Mozez Singh	s8807	Movie	March 2 201
201987	Zubaan	Anita Shabdish	India	Music & Musicals	Mozez Singh	s8807	Movie	March 2 201
201988	Zubaan	Chittaranjan Tripathy	India	Dramas	Mozez Singh	s8807	Movie	March 2 201
201989	Zubaan	Chittaranjan Tripathy	India	International Movies	Mozez Singh	s8807	Movie	March 2 201
201990	Zubaan	Chittaranjan Tripathy	India	Music & Musicals	Mozez Singh	s8807	Movie	March 2 201

145843 rows × 12 columns

In [41]:
```
# only TV-SHOW DATA
df_final_season
```

Out[41]:

	title	Actors	country	Genre	director	show_id	type	date_added	relea
1	Blood & Water	Ama Qamata	South Africa	International TV Shows	UnKnown Director	s2	TV Show	September 24, 2021	
2	Blood & Water	Ama Qamata	South Africa	TV Dramas	UnKnown Director	s2	TV Show	September 24, 2021	
3	Blood & Water	Ama Qamata	South Africa	TV Mysteries	UnKnown Director	s2	TV Show	September 24, 2021	
4	Blood & Water	Khosi Ngema	South Africa	International TV Shows	UnKnown Director	s2	TV Show	September 24, 2021	
5	Blood & Water	Khosi Ngema	South Africa	TV Dramas	UnKnown Director	s2	TV Show	September 24, 2021	
...	...	...	...	...	...	...	...	...	
201864	Zindagi Gulzar Hai	Hina Khawaja Bayat	Pakistan	Romantic TV Shows	UnKnown Director	s8801	TV Show	December 15, 2016	
201865	Zindagi Gulzar Hai	Hina Khawaja Bayat	Pakistan	TV Dramas	UnKnown Director	s8801	TV Show	December 15, 2016	
201932	Zombie Dumb	Unknown Actor	NaN	Kids' TV	UnKnown Director	s8804	TV Show	July 1, 2019	
201933	Zombie Dumb	Unknown Actor	NaN	Korean TV Shows	UnKnown Director	s8804	TV Show	July 1, 2019	
201934	Zombie Dumb	Unknown Actor	NaN	TV Comedies	UnKnown Director	s8804	TV Show	July 1, 2019	

56148 rows × 12 columns

Non-Graphical Analysis: Value counts and unique attributes

In [7]:
```
# UNIQUE ATTRIBUTES
print("No. of ratings - ",df["rating"].nunique())
print("Total Titles - ",df["title"].nunique())
print("Total Directors - ",df["director"].nunique())
print("Total country - ",df["country"].nunique())
print("Total years  - ",df["release_year"].nunique())
print("Total Genres - ",df["listed_in"].nunique())
```

```
No. of ratings -  17
Total Titles -  8807
Total Directors -  4528
Total country -  748
Total years  -  74
Total Genres -  514
```

In [8]:
```
# ATTRIBUTES VALUE COUNTS
print("********* Name of types on Netflix ************ ")
print(pd.DataFrame(df["type"].value_counts().reset_index()))
print()
print("********* Name of country having shows on Netflix ************ ")
print(pd.DataFrame(df["country"].value_counts().reset_index()))
print("********* Name of titles on Netflix ************ ")
print(pd.DataFrame(df["title"].value_counts().reset_index()))
print()
print("********* Name of Directors on Netflix from MOVIES ************ ")
print(pd.DataFrame(df_final_movies["director"].value_counts().reset_index()
print()
print("********* Name of Directors on Netflix  From TV-SHOWS ************
print(pd.DataFrame(df_final_season["director"].value_counts().reset_index()
print()
print("********* Name of Actors on Netflix  From Movies ************ ")
print(pd.DataFrame(df_final_movies["Actors"].value_counts().reset_index()))
print()
print("********* Name of Actors on Netflix  From TV-SHOWS ************ ")
print(pd.DataFrame(df_final_season["Actors"].value_counts().reset_index()))
```

```
********* Name of types on Netflix *************
     index  type
0    Movie  6131
1  TV Show  2676

********* Name of country having shows on Netflix *************
                                    index  country
0                           United States     2818
1                                   India      972
2                          United Kingdom      419
3                                   Japan      245
4                             South Korea      199
..                                    ...      ...
743             Romania, Bulgaria, Hungary        1
744                     Uruguay, Guatemala        1
745               France, Senegal, Belgium        1
746  Mexico, United States, Spain, Colombia        1
747           United Arab Emirates, Jordan        1

[748 rows x 2 columns]
********* Name of titles on Netflix *************
                                    index  title
0                     Dick Johnson Is Dead      1
1                                  Ip Man 2      1
2          Hannibal Buress: Comedy Camisado      1
3                               Turbo FAST      1
4                            Masha's Tales      1
...                                    ...    ...
8802                       Love for Sale 2      1
8803                          ROAD TO ROMA      1
8804                             Good Time      1
8805  Captain Underpants Epic Choice-o-Rama      1
8806                                Zubaan      1

[8807 rows x 2 columns]

********* Name of Directors on Netflix from MOVIES *************
                    index  director
0        UnKnown Director      1285
1         Martin Scorsese       419
2         Youssef Chahine       409
3     Cathy Garcia-Molina       356
4        Steven Spielberg       355
...                   ...       ...
4773        John Smithson         1
4774         Alex Coletti         1
4775       Michael Govier         1
4776       Sabaah Folayan         1
4777      Kirsten Johnson         1

[4778 rows x 2 columns]

********* Name of Directors on Netflix  From TV-SHOWS *************
                index  director
0    UnKnown Director     49358
1          Noam Murro       189
2       Thomas Astruc       160
3     Houda Benyamina       104
4     Damien Chazelle       104
..                ...       ...
295     Rashida Jones         1
```

```
296   Sharon Grimberg         1
297   Garrett Bradley         1
298       Alex Gibney         1
299  Padraic McKinley         1

[300 rows x 2 columns]

********* Name of Actors on Netflix  From Movies *************
                index  Actors
0       Unknown Actor    1328
1         Liam Neeson     161
2       Alfred Molina     157
3      John Krasinski     138
4         Salma Hayek     130
...               ...     ...
25947   Bill Goldberg       1
25948        BJ Verot       1
25949      Sean Skene       1
25950   Marrese Crump       1
25951    Rebekah Graf       1

[25952 rows x 2 columns]

********* Name of Actors on Netflix  From TV-SHOWS *************
                    index  Actors
0           Unknown Actor     818
1      David Attenborough      82
2        Takahiro Sakurai      56
3                Yuki Kaji     45
4                Ai Kayano     41
...                    ...     ...
14859       Jimmy O. Yang       1
14860       Diana Silvers       1
14861      John Malkovich       1
14862      Sassy Bermudez       1
14863       Telma Hopkins       1

[14864 rows x 2 columns]
```

In [52]:
```
print("********* Name of Genres on Netflix from movies ************* ")
print(pd.DataFrame(df_final_movies["Genre"].value_counts().reset_index()))
print()
print("********* Name of Genres on Netflix from TV-SHOWS ************* ")
print(pd.DataFrame(df_final_season["Genre"].value_counts().reset_index()))
```

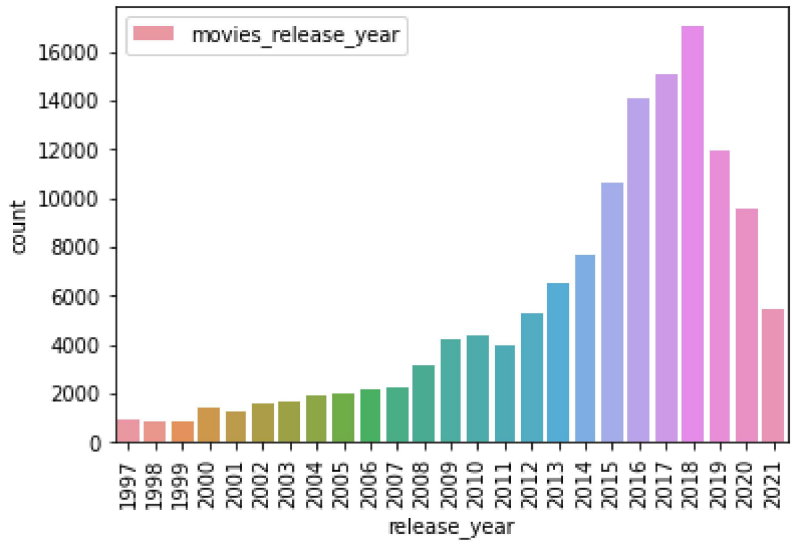
```
********* Name of Genres on Netflix from movies *************
                        index  Genre
0                      Dramas  29775
1       International Movies  28211
2                    Comedies  20829
3          Action & Adventure  12216
4          Independent Movies   9834
5   Children & Family Movies   9771
6                   Thrillers   7107
7             Romantic Movies   6412
8               Horror Movies   4571
9            Sci-Fi & Fantasy   4037
10           Music & Musicals   3077
11              Documentaries   2407
12              Sports Movies   1531
13             Classic Movies   1434
14                Cult Movies   1077
15             Anime Features   1045
16               LGBTQ Movies    838
17      Faith & Spirituality    719
18            Stand-Up Comedy    540
19                     Movies    412

********* Name of Genres on Netflix from TV-SHOWS *************
                           index  Genre
0         International TV Shows  12845
1                      TV Dramas   8942
2                    TV Comedies   4963
3                  Crime TV Shows   4733
4                       Kids' TV   4568
5              Romantic TV Shows   3049
6                   Anime Series   2313
7          TV Action & Adventure   2288
8      Spanish-Language TV Shows   2126
9                British TV Shows   1808
10                   TV Mysteries   1281
11               Korean TV Shows   1122
12           TV Sci-Fi & Fantasy   1045
13                      TV Horror    941
14                    Docuseries    845
15                  TV Thrillers    768
16                 Teen TV Shows    742
17                    Reality TV    735
18                      TV Shows    337
19             Classic & Cult TV    272
20  Stand-Up Comedy & Talk Shows    268
21           Science & Nature TV    157
```

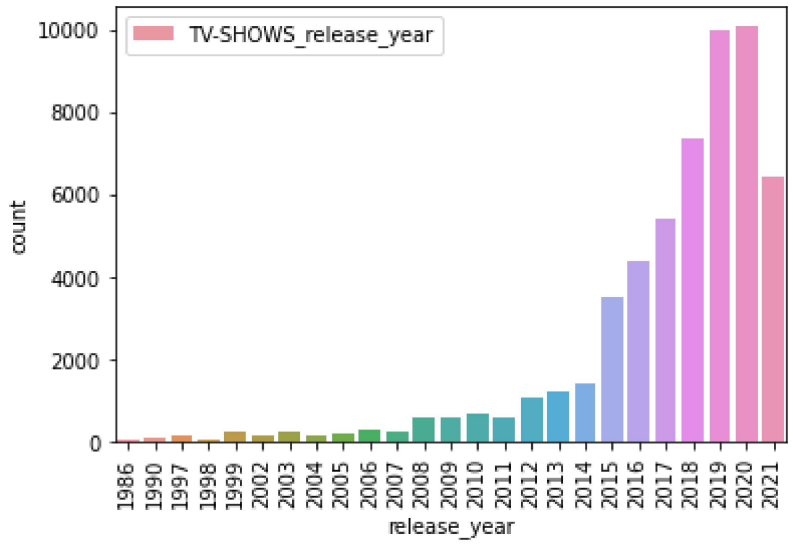
Visual Analysis - Univariate, Bivariate after pre-processing of the data

In [38]:
```
# considering the top datas from both Movies and TV-SHOWS
#1. Movies
top_3_genres=df_final_movies["Genre"].value_counts().index[:3]
top_3_titles=df_final_movies["title"].value_counts().index[:10]
top_3_actors=df_final_movies["Actors"].value_counts().index[:4]
top_3_directors=df_final_movies["director"].value_counts().index[:4]
top_3_months=df_final_movies["month"].value_counts().index[:3]
top_3_countries=df_final_movies["country"].value_counts().index[:3]
top_3_ratings_movies=df_final_movies["rating"].value_counts().index[:3]
top_25_years=df_final_movies["release_year"].value_counts().index[:25]
top_10_duration=df_final_movies["duration"].value_counts().index[:10]
#2. TV-SHOWS
top_3_genres1=df_final_season["Genre"].value_counts().index[:3]
top_3_titles1=df_final_season["title"].value_counts().index[:10]
top_3_actors1=df_final_season["Actors"].value_counts().index[:4]
top_3_directors1=df_final_season["director"].value_counts().index[:4]
top_3_months1=df_final_season["month"].value_counts().index[:3]
top_3_countries1=df_final_season["country"].value_counts().index[:3]
top_3_ratings_seasons1=df_final_season["rating"].value_counts().index[:3]
top_25_years1=df_final_season["release_year"].value_counts().index[:25]
top_10_duration1=df_final_season["duration"].value_counts().index[:10]
```

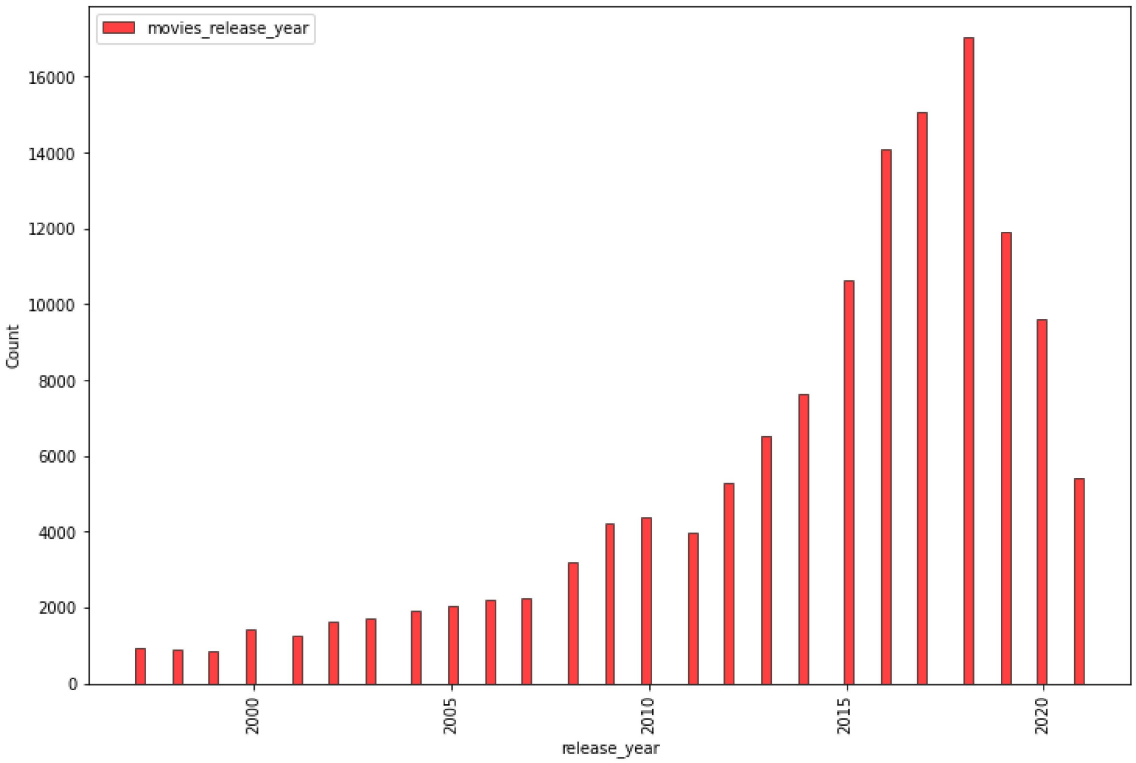
For continuous variable(s): Distplot, countplot, histogram for univariate analysis

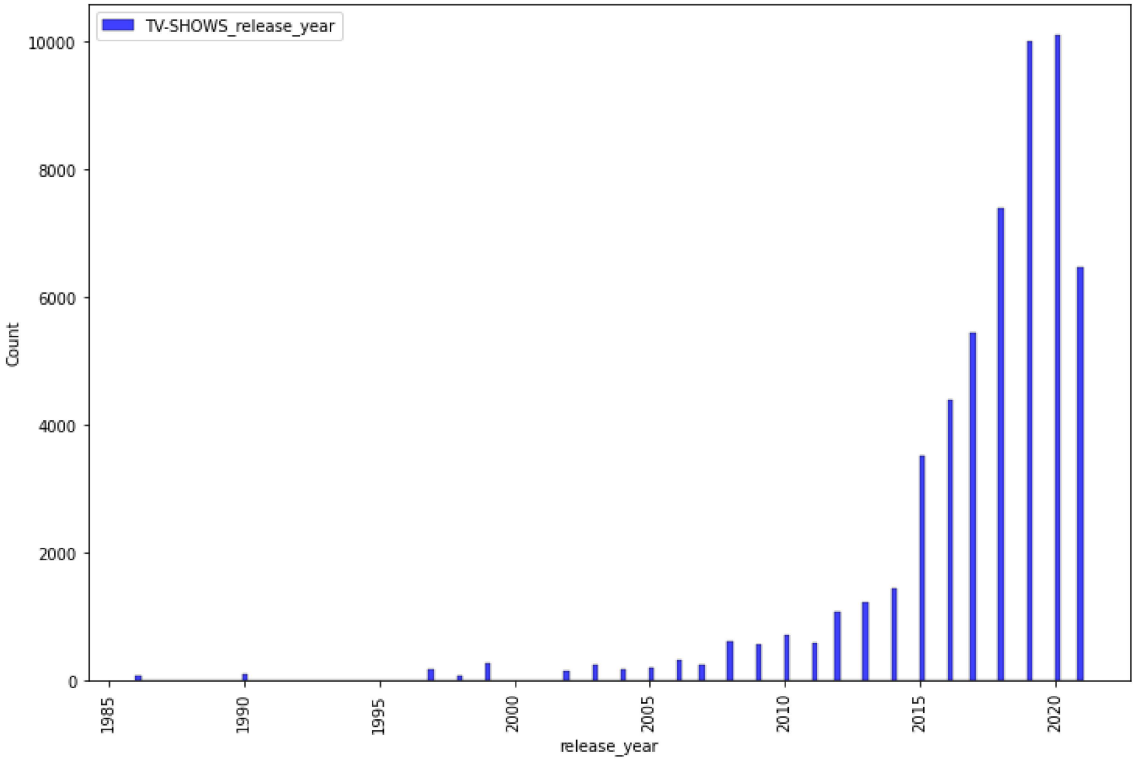
In [133]:
```
                                        # countplot for release_year
#movies
top_25_releaseyears=df_final_movies.loc[(df_final_movies["release_year"].is
sns.countplot(data=top_25_releaseyears,x="release_year")
plt.xticks(rotation=90)
plt.legend(["movies_release_year"])
plt.show()
# seasons
top_25_releaseyears1=df_final_season.loc[(df_final_season["release_year"].i
sns.countplot(data=top_25_releaseyears1,x="release_year")
plt.xticks(rotation=90)
plt.legend(["TV-SHOWS_release_year"])
plt.show()
```





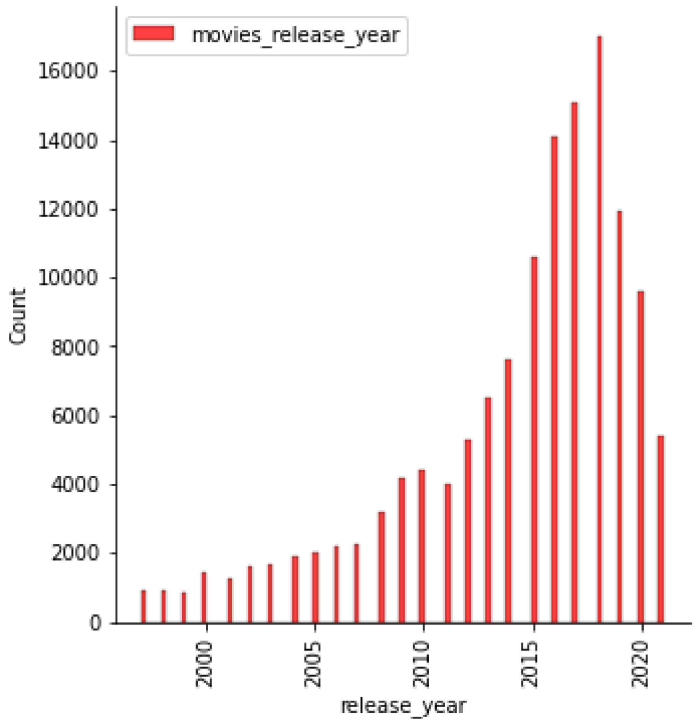
In [134]:
```
                                        # histplot for release_year
#movies
plt.figure(figsize=(12,8))
sns.histplot(data=top_25_releaseyears,x="release_year",color="red")
plt.xticks(rotation=90)
plt.legend(["movies_release_year","counts"])
plt.show()
#seasons
plt.figure(figsize=(12,8))
top_25_releaseyears1=df_final_season.loc[(df_final_season["release_year"].i
sns.histplot(data=top_25_releaseyears1,x="release_year",color="blue")
plt.xticks(rotation=90)
plt.legend(["TV-SHOWS_release_year"])
plt.show()
```



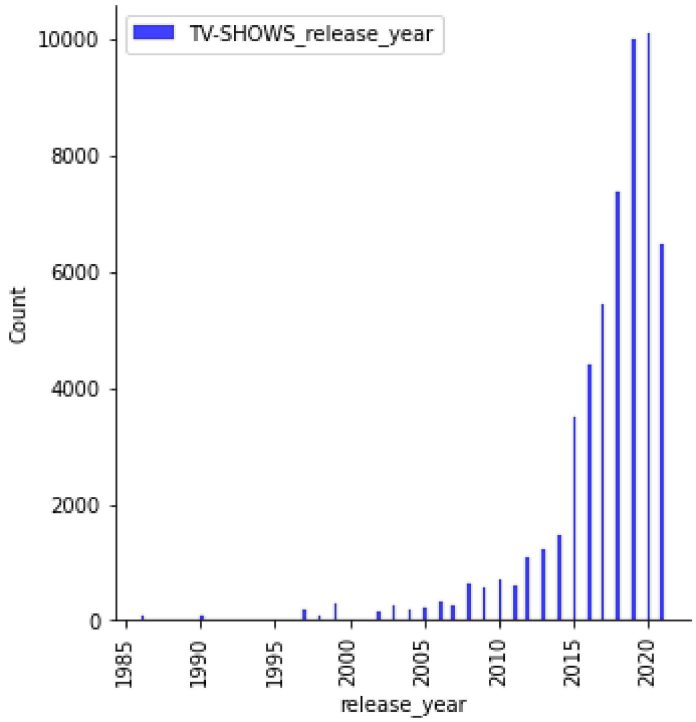

TV-SHOWS_release_year
Count
release_year
0
2000
4000
6000
8000
10000
1985
1990
1995
2000
2005
2010
2015
2020

In [135]:
```
                                    # Displot for Release_years
#movies
plt.figure(figsize=(12,8))
sns.displot(data=top_25_releaseyears,x="release_year",color="red")
plt.xticks(rotation=90)
plt.legend(["movies_release_year","counts"])
plt.show()
#seasons
plt.figure(figsize=(12,8))
top_25_releaseyears1=df_final_season.loc[(df_final_season["release_year"].i
sns.displot(data=top_25_releaseyears1,x="release_year",color="blue")
plt.xticks(rotation=90)
plt.legend(["TV-SHOWS_release_year"])
plt.show()
```

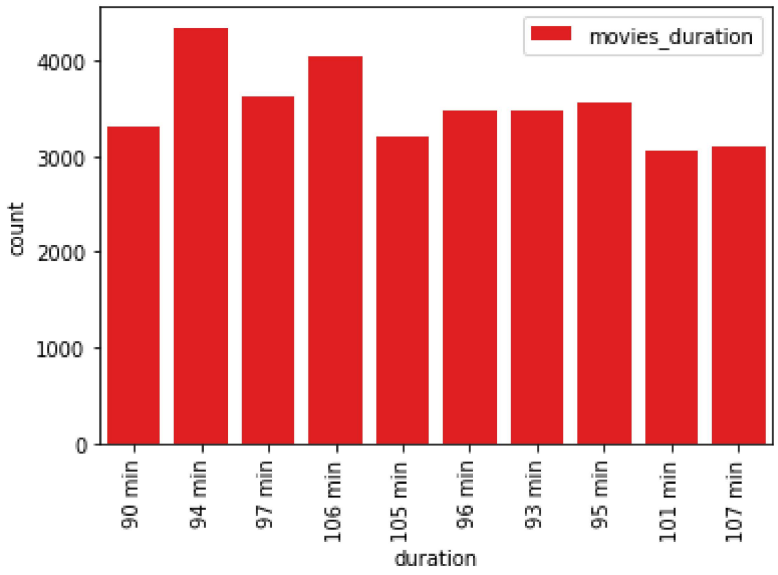
```
<Figure size 864x576 with 0 Axes>
```

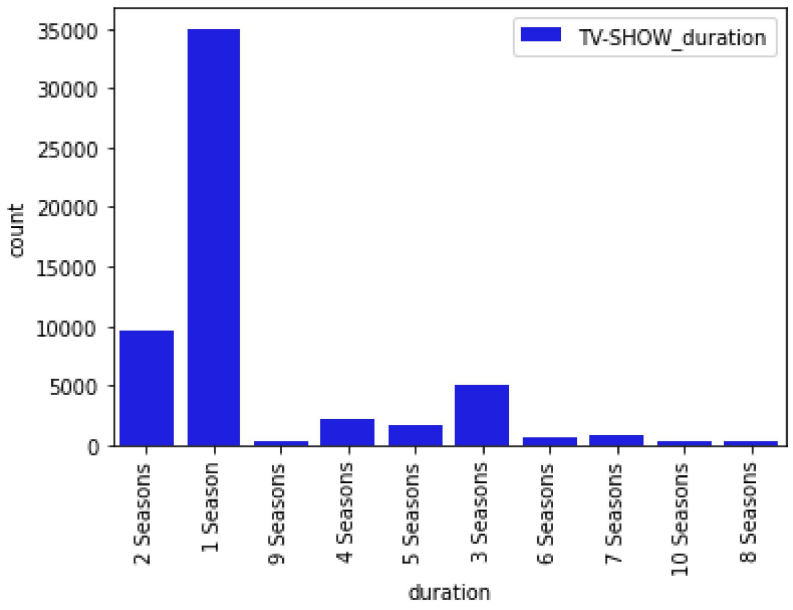



```
<Figure size 864x576 with 0 Axes>
```

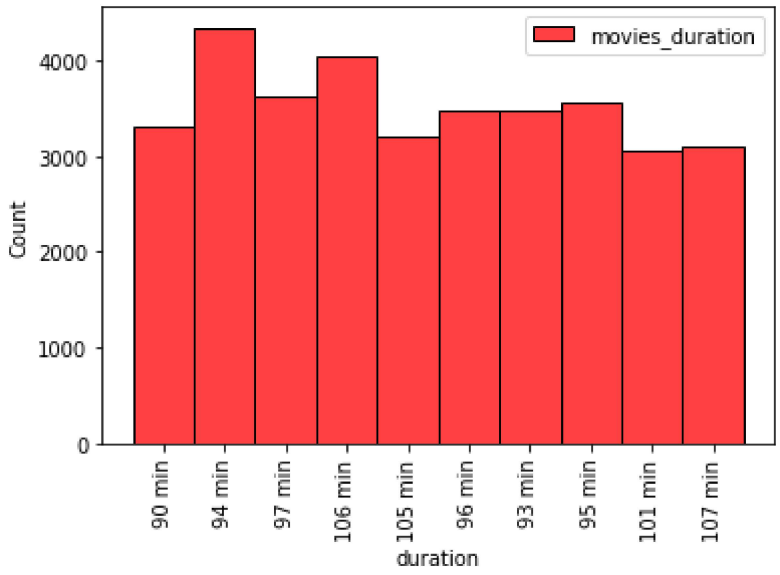

TV-SHOWS_release_year
10000
8000
6000
4000
2000
0
Count
1985
1990
1995
2000
2005
2010
2015
2020
release_year

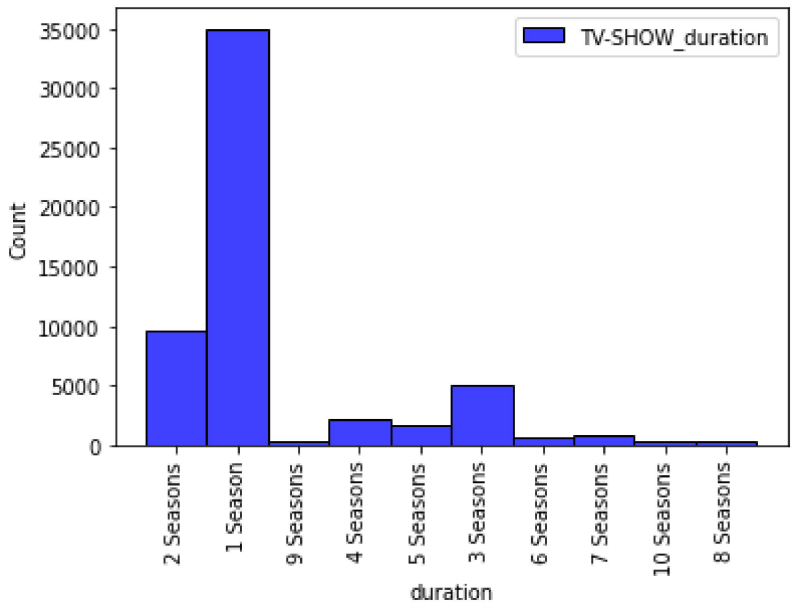
In [136]:
```
                                        #Countplot for duration
#movies
top_10_duration_movies=df_final_movies.loc[(df_final_movies["duration"].isi
sns.countplot(data=top_10_duration_movies,x="duration",color="red")
plt.xticks(rotation=90)
plt.legend(["movies_duration"])
plt.show()
#seaons
top_10_duration_season=df_final_season.loc[(df_final_season["duration"].isi
sns.countplot(data=top_10_duration_season,x="duration",color="blue")
plt.xticks(rotation=90)
plt.legend(["TV-SHOW_duration"])
plt.show()
```





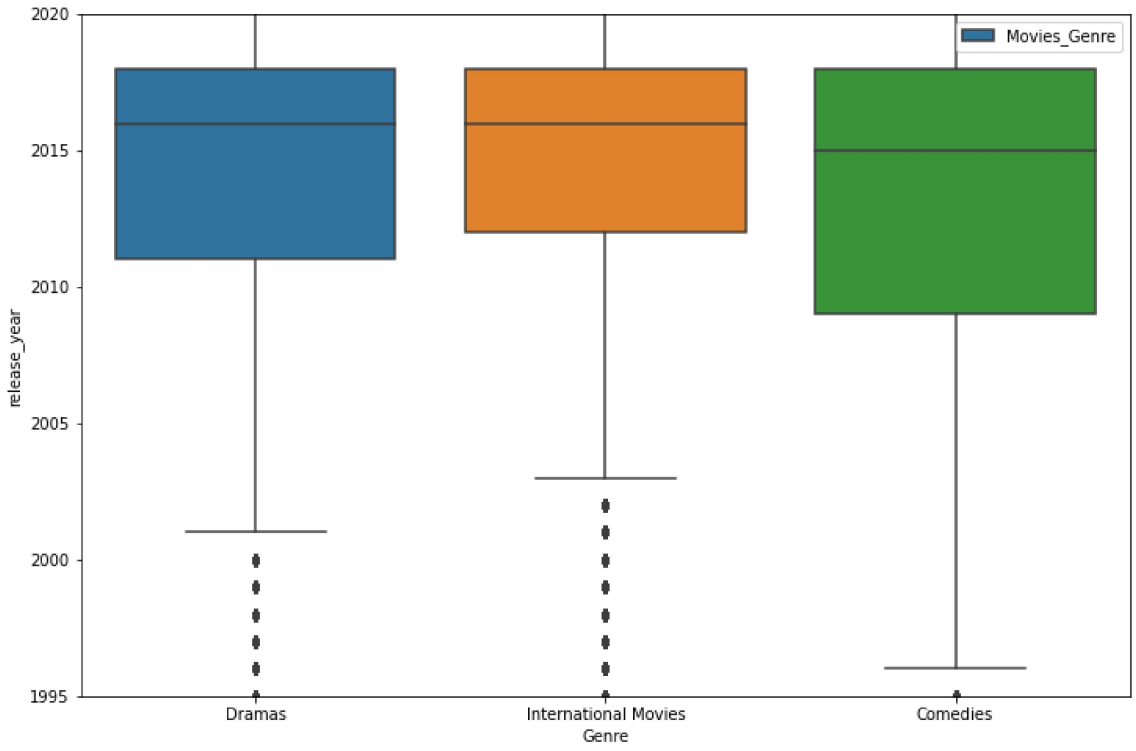
In [137]:
```
                                                # histplot for release_year
#movies
top_10_duration_movies=df_final_movies.loc[(df_final_movies["duration"].isi
sns.histplot(data=top_10_duration_movies,x="duration",color="red")
plt.xticks(rotation=90)
plt.legend(["movies_duration"])
plt.show()
#seaons
top_10_duration_season=df_final_season.loc[(df_final_season["duration"].isi
sns.histplot(data=top_10_duration_season,x="duration",color="blue")
plt.xticks(rotation=90)
plt.legend(["TV-SHOW_duration"])
plt.show()
```

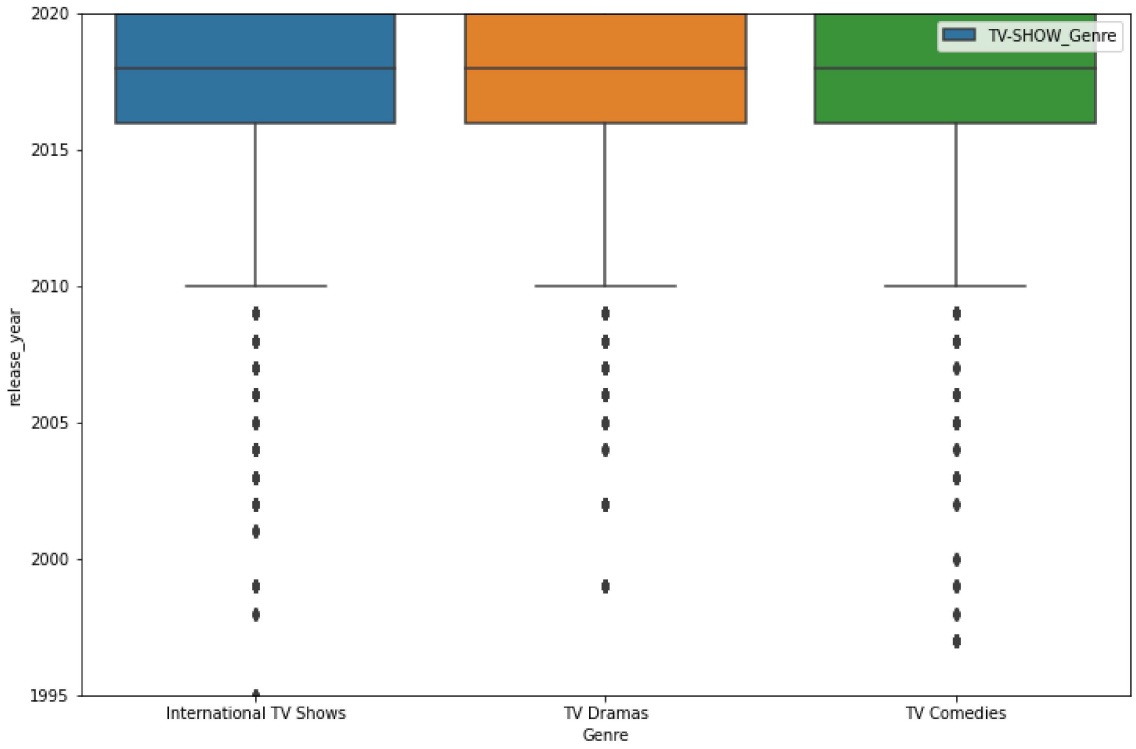




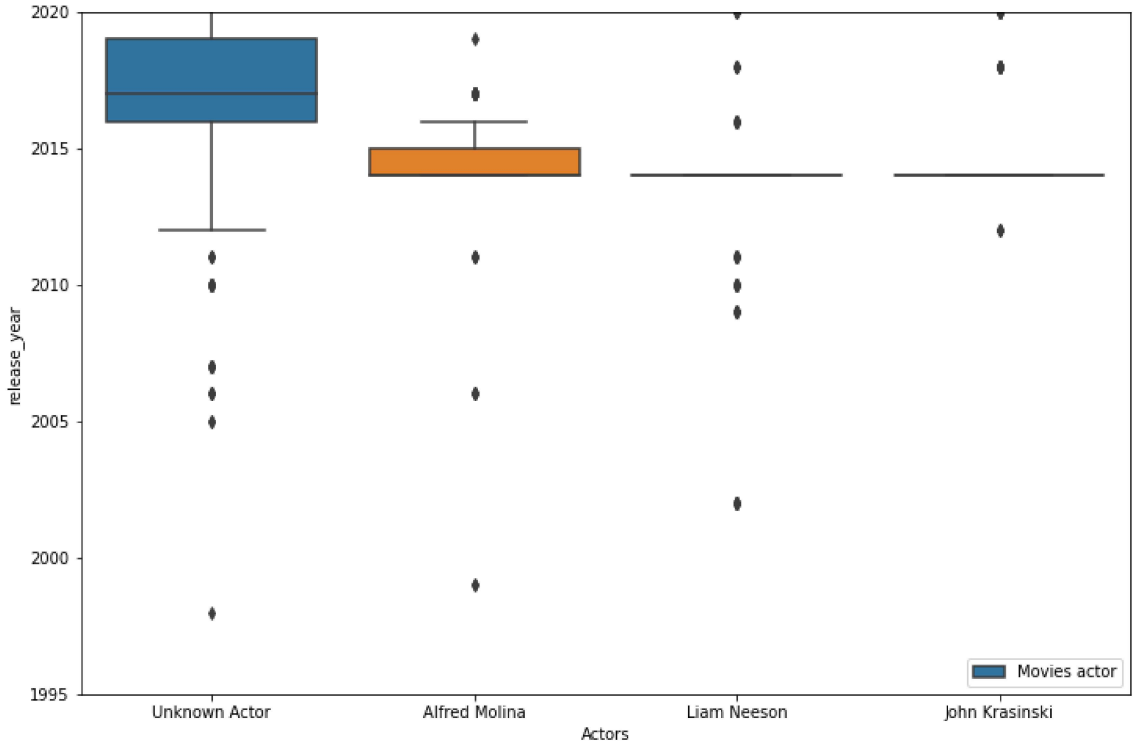
For categorical variable(s): Boxplot

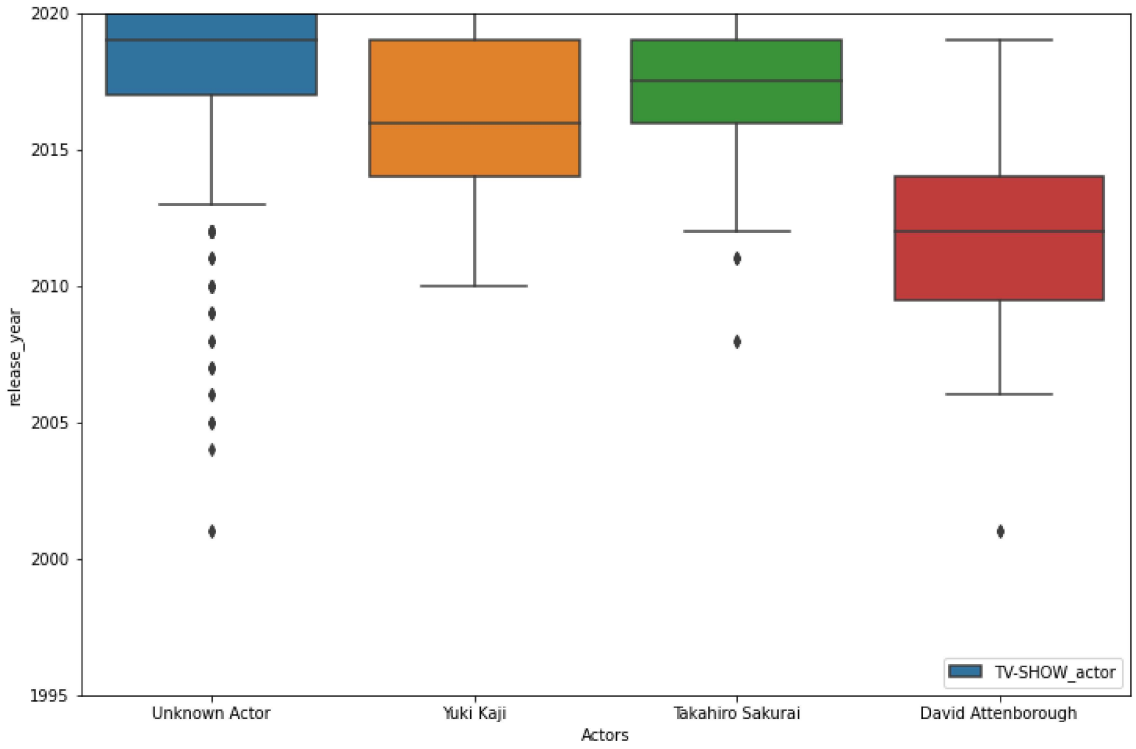
In [138]:
```
                                        # TOP 3 Genre v/s last 25 years
#Movies
top_3_data_Genres=df_final_movies.loc[(df_final_movies["Genre"].isin(top_3_
plt.figure(figsize=(12,8))
sns.boxplot(data=top_3_data_Genres,x="Genre",y="release_year")
plt.ylim(bottom=1995,top=2020)
plt.legend(["Movies_Genre"])
plt.show()
#TV-SHOWS
top_3_data_Genres1=df_final_season.loc[(df_final_season["Genre"].isin(top_3
plt.figure(figsize=(12,8))
sns.boxplot(data=top_3_data_Genres1,x="Genre",y="release_year")
plt.ylim(bottom=1995,top=2020)
plt.legend(["TV-SHOW_Genre"])
plt.show()
```



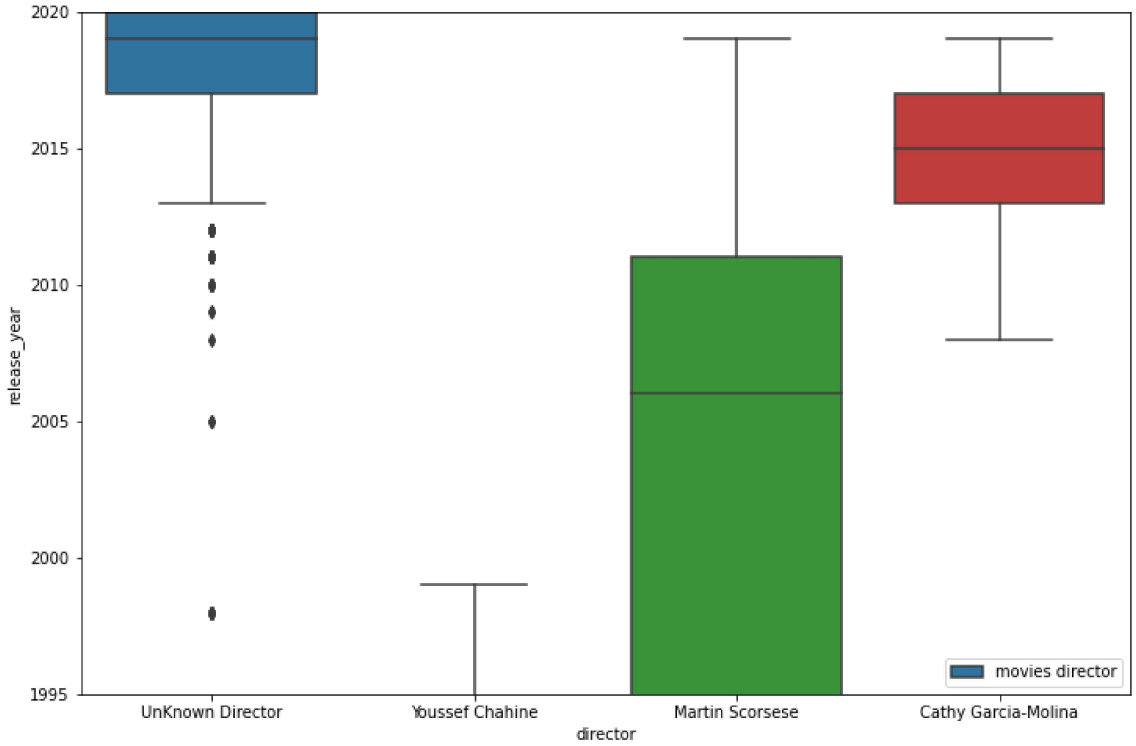

TV-SHOW_Genre
release_year
2020
2015
2010
2005
2000
1995
International TV Shows
TV Dramas
TV Comedies
Genre

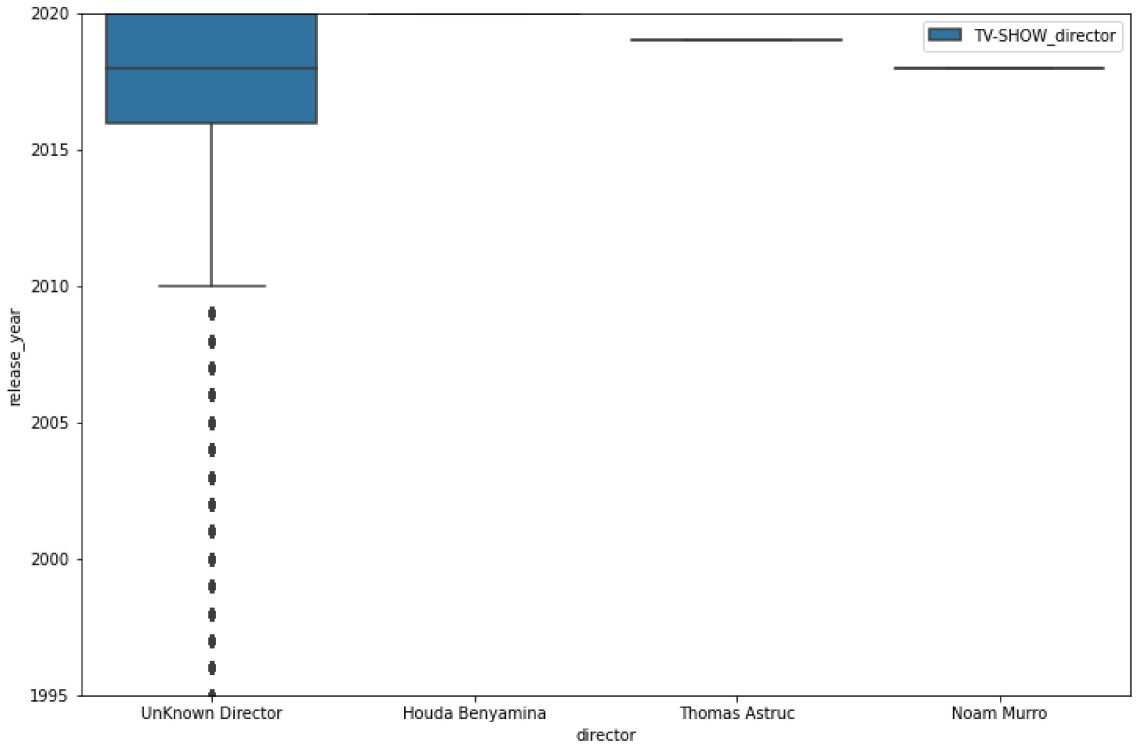
In [14]:
```
                                    # TOP 4 Actor v/s last 25 years
#Movies
top_3_data_actors=df_final_movies.loc[(df_final_movies["Actors"].isin(top_3
plt.figure(figsize=(12,8))
sns.boxplot(data=top_3_data_actors,x="Actors",y="release_year")
plt.ylim(bottom=1995,top=2020)
plt.legend(["Movies actor"])
plt.show()
#TV-SHOWS
top_3_data_actors1=df_final_season.loc[(df_final_season["Actors"].isin(top_
plt.figure(figsize=(12,8))
sns.boxplot(data=top_3_data_actors1,x="Actors",y="release_year")
plt.ylim(bottom=1995,top=2020)
plt.legend(["TV-SHOW_actor"])
plt.show()
```



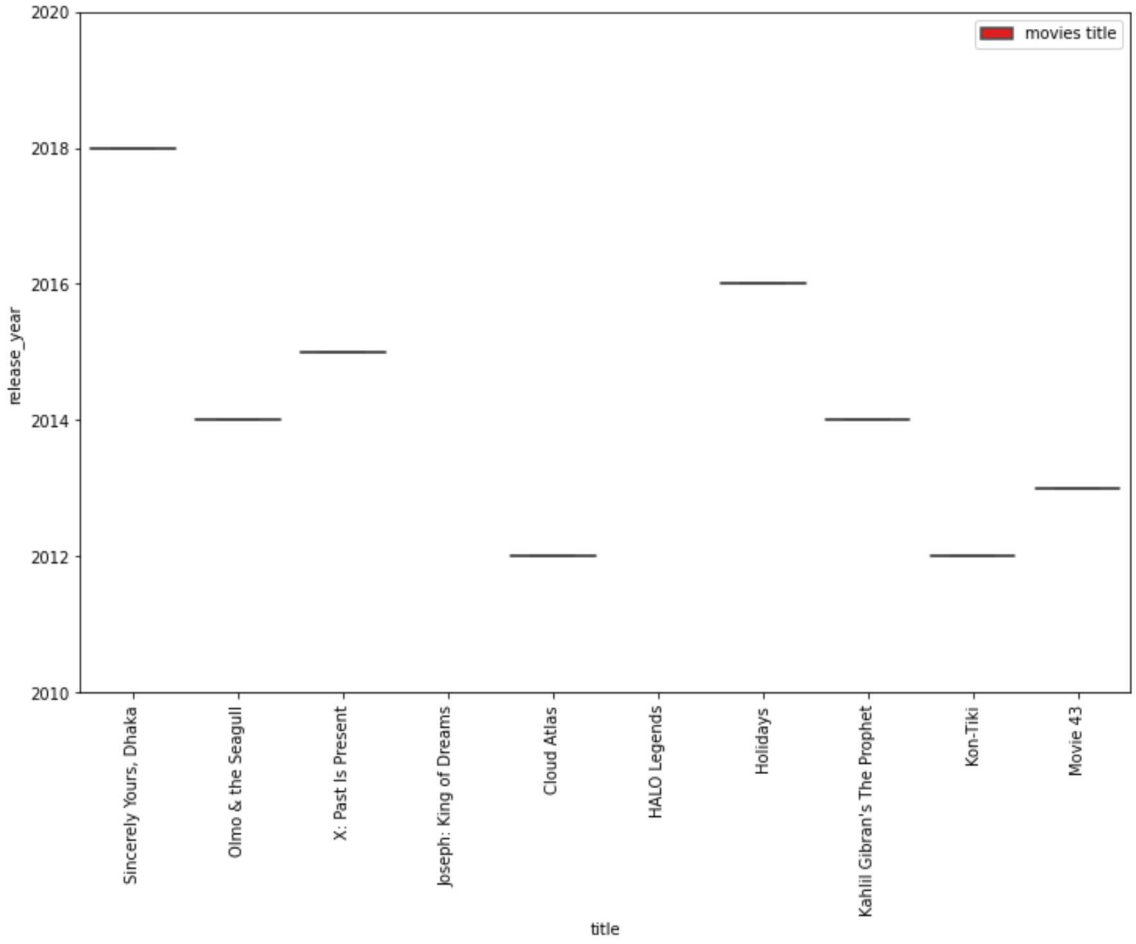

2020
2015
2010
2005
2000
1995
release_year
Unknown Actor
Yuki Kaji
Takahiro Sakurai
David Attenborough
Actors
TV-SHOW_actor

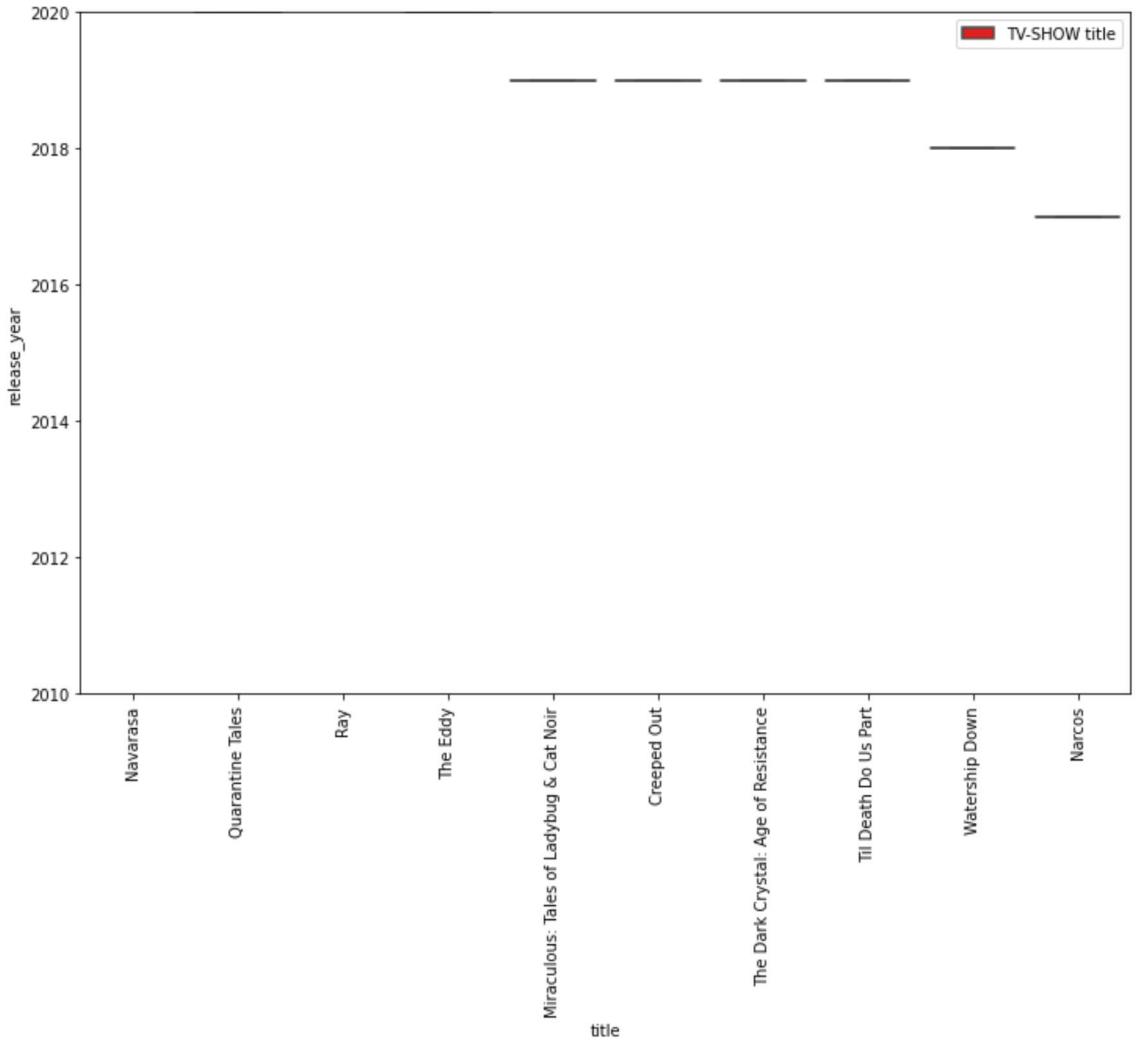
In [15]:
```
                                    # Directors v/s last 25 years
#Movies
top_3_data_directors=df_final_movies.loc[(df_final_movies["director"].isin(
plt.figure(figsize=(12,8))
sns.boxplot(data=top_3_data_directors,x="director",y="release_year")
plt.ylim(bottom=1995,top=2020)
plt.legend(["movies director"])
plt.show()
#TV-SHOWS
top_3_data_directors1=df_final_season.loc[(df_final_season["director"].isin
plt.figure(figsize=(12,8))
sns.boxplot(data=top_3_data_directors1,x="director",y="release_year")
plt.ylim(bottom=1995,top=2020)
plt.legend(["TV-SHOW_director"])
plt.show()
```




TV-SHOW_director
release_year
2020
2015
2010
2005
2000
1995
UnKnown Director
Houda Benyamina
Thomas Astruc
Noam Murro
director

In [45]:
```
                                                    # top Titles v/s last 10 yea
#Movies
top_3_data_titles=df_final_movies.loc[(df_final_movies["title"].isin(top_3_
plt.figure(figsize=(12,8))
sns.boxplot(data=top_3_data_titles,x="title",y="release_year",color="red")
plt.ylim(bottom=2010,top=2020)
plt.legend(["movies title"])
plt.xticks(rotation=90)
plt.show()

# TV-SHOWS
top_3_data_titles1=df_final_season.loc[(df_final_season["title"].isin(top_3
plt.figure(figsize=(12,8))
sns.boxplot(data=top_3_data_titles1,x="title",y="release_year",color="red")
plt.ylim(bottom=2010,top=2020)
plt.legend(["TV-SHOW title"])
plt.xticks(rotation=90)
plt.show()
```





For correlation: Heatmaps, Pairplots (10 Points)

In [57]:
```
top_data_actors_directors=df_final_movies.loc[(df_final_movies["Actors"].is
top_data_actors_directors
```

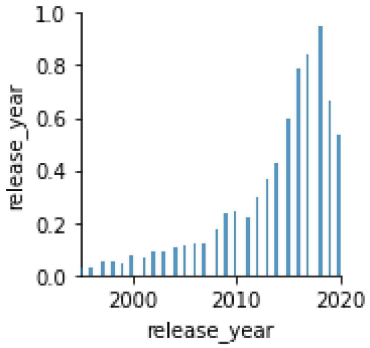
Out[57]:

	title	Actors	country	Genre	director	type	release_year	rating
10052	9to5: The Story of a Movement	Unknown Actor	NaN	Documentaries	UnKnown Director	Movie	2021	TV-MA
16652	Sisters on Track	Unknown Actor	NaN	Documentaries	UnKnown Director	Movie	2021	PG
16653	Sisters on Track	Unknown Actor	NaN	Sports Movies	UnKnown Director	Movie	2021	PG
18759	Trese After Dark	Unknown Actor	NaN	Anime Features	UnKnown Director	Movie	2021	TV-14
18760	Trese After Dark	Unknown Actor	NaN	Documentaries	UnKnown Director	Movie	2021	TV-14
...	...	...	...	...	...	...	...	...
183414	Smash: Motorized Mayhem	Unknown Actor	United States	Documentaries	UnKnown Director	Movie	2017	TV-MA
183415	Smash: Motorized Mayhem	Unknown Actor	United States	Sports Movies	UnKnown Director	Movie	2017	TV-MA
189358	The Darkest Dawn	Unknown Actor	United Kingdom	Action & Adventure	UnKnown Director	Movie	2016	TV-MA
189359	The Darkest Dawn	Unknown Actor	United Kingdom	Independent Movies	UnKnown Director	Movie	2016	TV-MA
189360	The Darkest Dawn	Unknown Actor	United Kingdom	International Movies	UnKnown Director	Movie	2016	TV-MA

116 rows × 10 columns

In [85]:
```
plt.figure(figsize=(20,15))
sns.pairplot(data=df_final_movies)
plt.xlim(left=1995,right=2020)
# plt.ylim(bottom=0,top=1)
```

Out[85]:
```
(1995.0, 2020.0)

<Figure size 1440x1080 with 0 Axes>
```



Insights based on Non-Graphical and Visual Analysis (10 Points)

1 Comments on the range of attributes

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

3 Comments for each univariate and bivariate plot

1

Comments on the range of attributes

1.Based on the given data we observe that there are wide variety of Genres nowdays 2.The platform helps the user to display the most watched shows

2

1. Through the data we have have observe there are relationship betwwen directors , casting and actors.
2. As the years are moving forward the Rating depend directly on cast and type of Genre title belogs too

3

For univariate plots 1.As the years are moving craze in people for seeing movies and tv-shows are increasing 2.For the duration I see that in movies people like to watch movies duration between 90 to 100 minutes time interval and for tv shows people has high craze for shows having 1 or 2 seasons

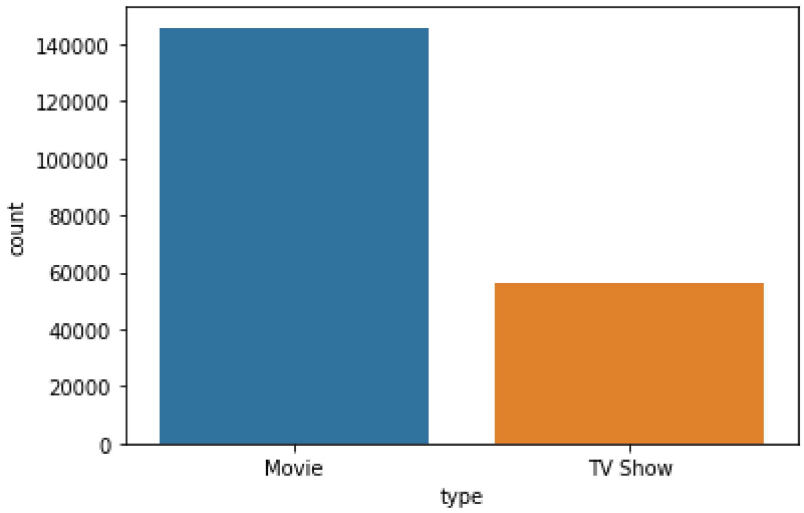
For bivariate plots 1.People like to watch high rating movies and shows

Business Insights - Should include patterns observed in the data along with what you can infer from it

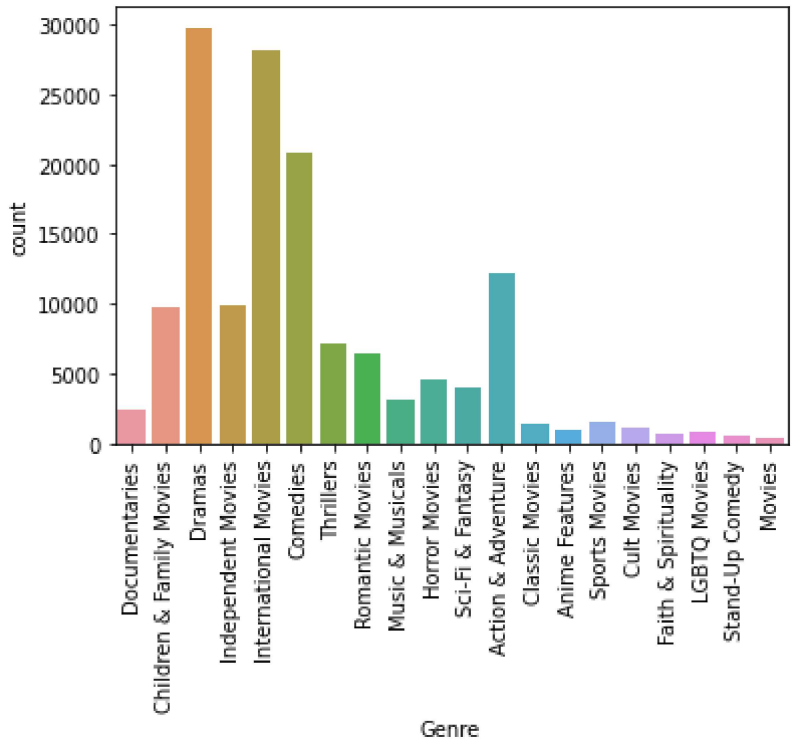
For every below plots insight number and result is mentioned

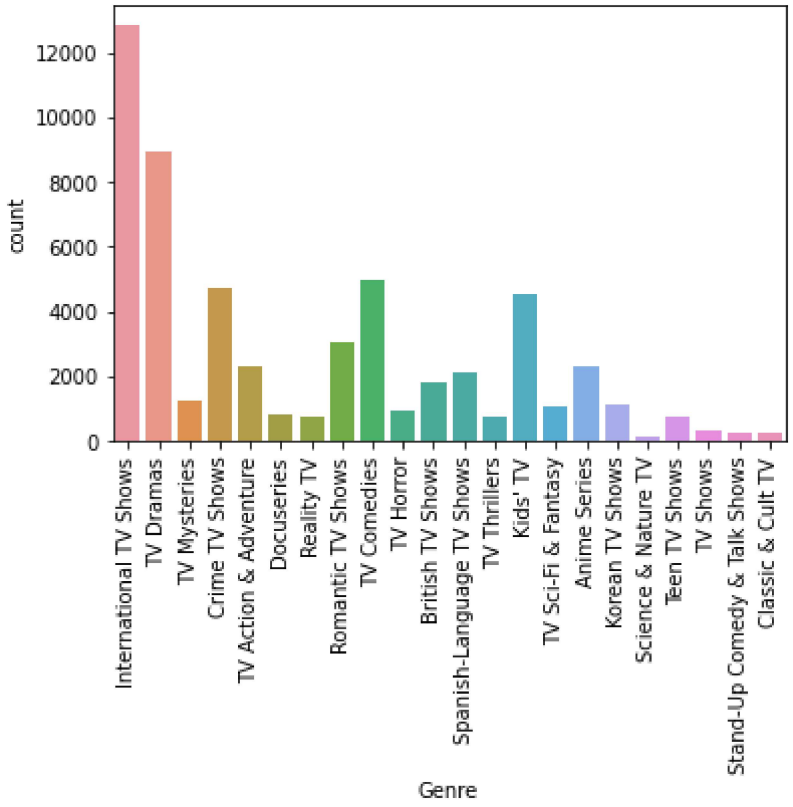
In [142]:
```
# INSIGHT 1
sns.countplot(data=df_final,x="type")

# RESULT--SHOWS DEMAND OF MOVIES IS MORE THEN TV SHOWS
```

Out[142]: <AxesSubplot: xlabel='type', ylabel='count'>

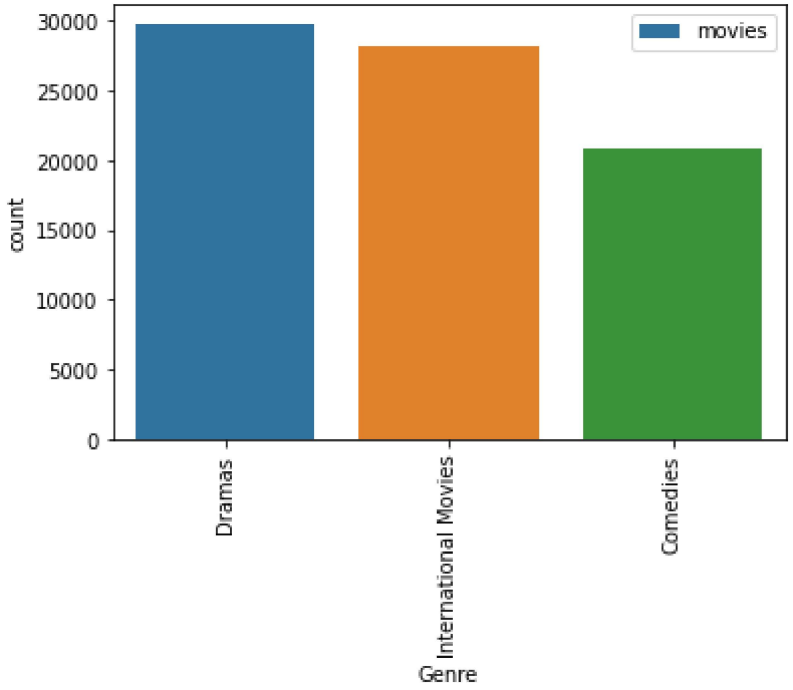


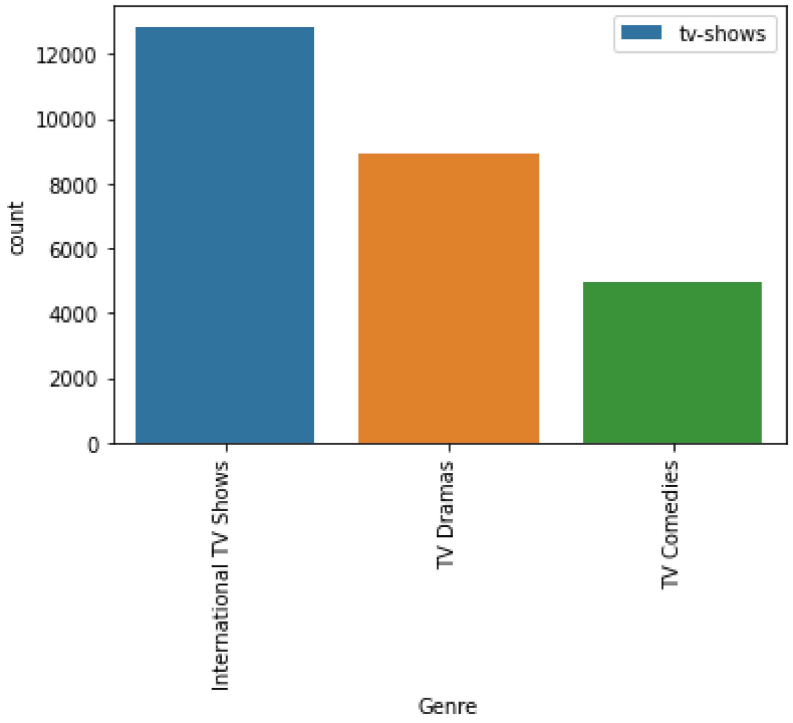
In [89]:
```
# INSIGHT 2
#movies
sns.countplot(data=df_final_movies,x="Genre")
plt.xticks(rotation=90)
plt.show()

#seasons
sns.countplot(data=df_final_season,x="Genre")
plt.xticks(rotation=90)
plt.show()

## RESULT--
## From the below data we can observe people needs more dramas movies and i
```



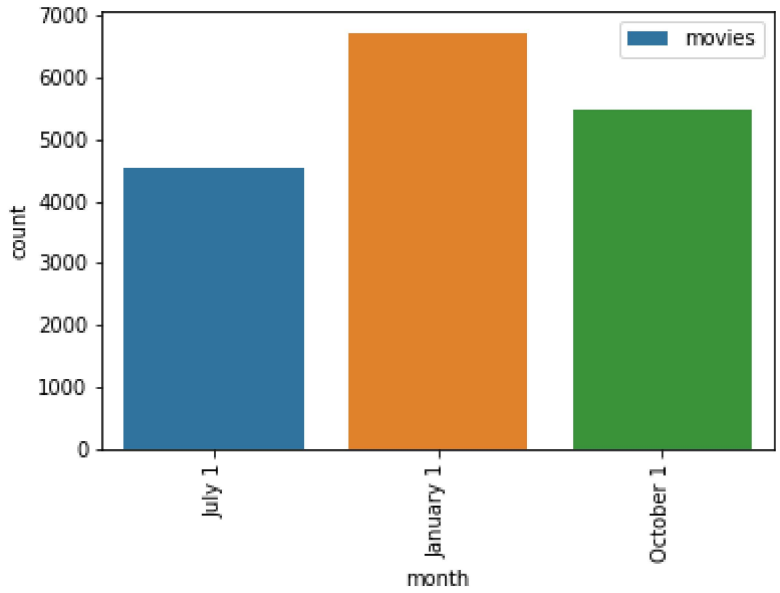

12000
10000
8000
6000
4000
2000
0
count
International TV Shows
TV Dramas
TV Mysteries
Crime TV Shows
TV Action & Adventure
Docuseries
Reality TV
Romantic TV Shows
TV Comedies
TV Horror
British TV Shows
Spanish-Language TV Shows
TV Thrillers
Kids' TV
TV Sci-Fi & Fantasy
Anime Series
Korean TV Shows
Science & Nature TV
Teen TV Shows
TV Shows
Stand-Up Comedy & Talk Shows
Classic & Cult TV
Genre

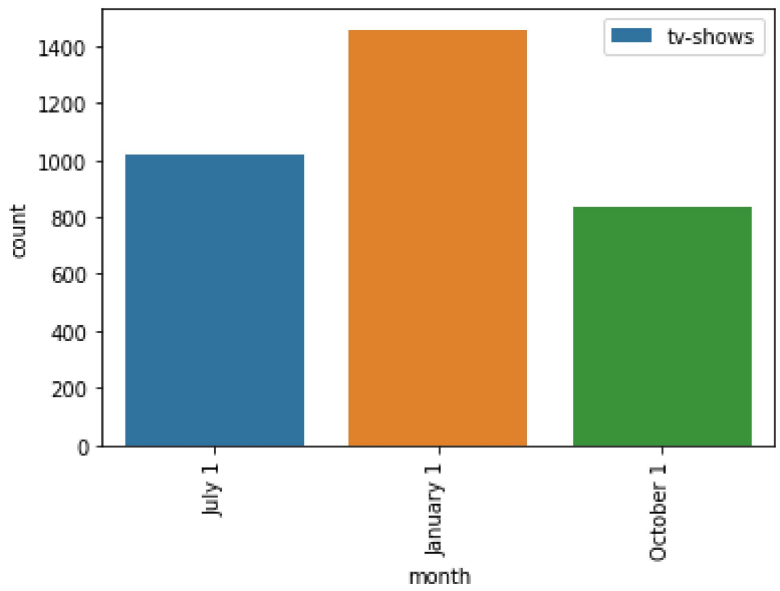
In [135]:
```
# INSIGHT 3
#movies
top_3_data_Genres=df_final_movies.loc[(df_final_movies["Genre"].isin(top_3_
sns.countplot(data=top_3_data_Genres,x="Genre")
plt.xticks(rotation=90)
plt.legend(["movies"])
plt.show()
#season
top_3_data_Genres1=df_final_season.loc[(df_final_season["Genre"].isin(top_3
sns.countplot(data=top_3_data_Genres1,x="Genre")
plt.xticks(rotation=90)
plt.legend(["tv-shows"])
plt.show()

# RESULT--
## FROM THE BELOW GRAPH WE SEE TOP 3 GENRES IN MOVIES ARE DRAMAS, INTERNATI
## TOP 3 GENRES IN TV SHOWS ARE INTERNATIONAL TV SHOWS,DRAMAS, TV COMEDIES
```



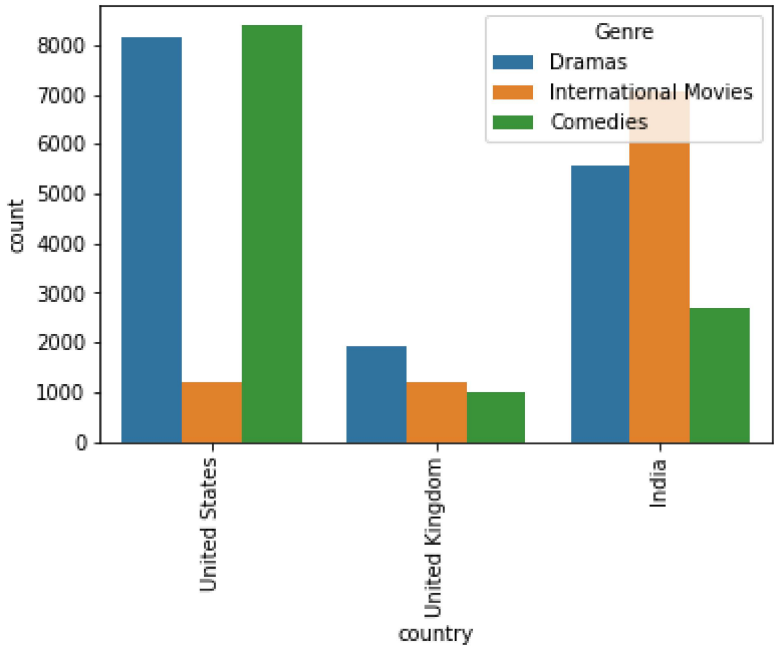

tv-shows
12000
10000
8000
6000
4000
2000
0
count
International TV Shows
TV Dramas
TV Comedies
Genre

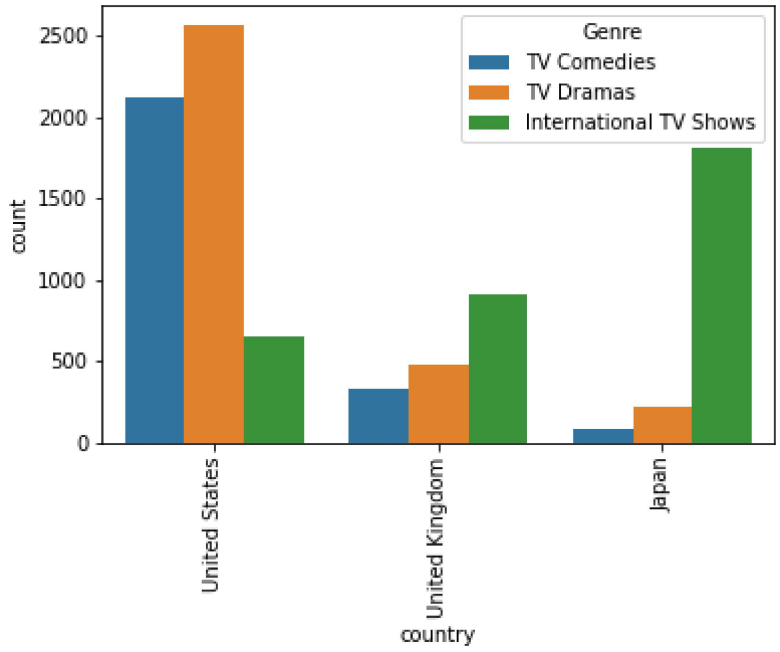
In [111]:
```
# INSIGHT 4
# movies
top_3_data_months=df_final_movies.loc[(df_final_movies["month"].isin(top_3_
sns.countplot(data=top_3_data_months,x="month")
plt.xticks(rotation=90)
plt.legend(["movies"])
plt.show()
#seasons
top_3_data_months1=df_final_season.loc[(df_final_season["month"].isin(top_3
sns.countplot(data=top_3_data_months1,x="month")
plt.xticks(rotation=90)
plt.legend(["tv-shows"])
plt.show()

# RESULT--
# FROM THE BELOW WE SEE PEOPLE LIKE MORE MOVIES AND TV-SHOWS TO RELEASE IN
```





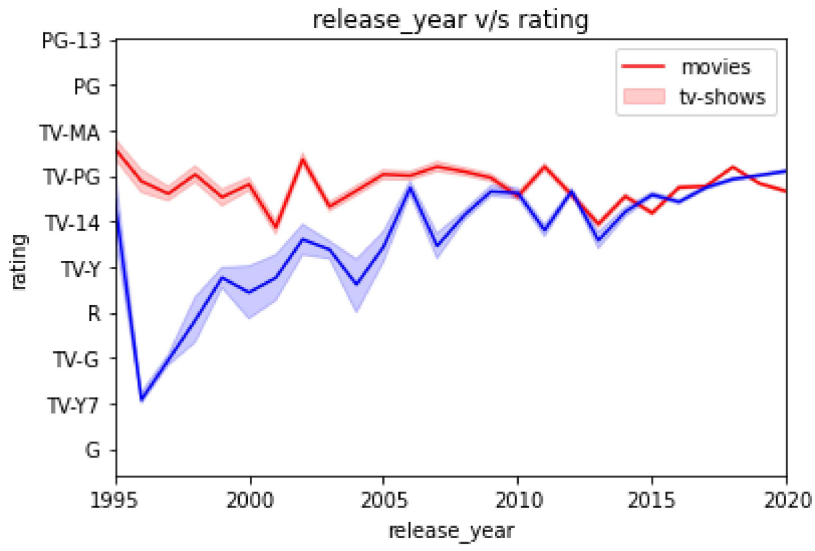
In [102]:
```
# INSIGHT 5

#movies
top_3_data_Genres_countries=df_final_movies.loc[(df_final_movies["Genre"].i
sns.countplot(data=top_3_data_Genres_countries,x="country",hue="Genre")
plt.xticks(rotation=90)
# plt.legend(["movies"])
plt.show()
#seasons
top_3_data_Genres_countries=df_final_season.loc[(df_final_season["Genre"].i
sns.countplot(data=top_3_data_Genres_countries,x="country",hue="Genre")
plt.xticks(rotation=90)
plt.show()


# RESULT--
# FROM BELOW WE SEE TOP 3 GENRES IN TOP 3 COUNTRIES I.E IN US PEOPLE LIKE T
## FOR TV SHOWS US--TV DRAMAS, UK--INTERNATIONAL TV SHOWS, JAPAN--INTERNATI
## NOTE IN BOTH MOVIES AND TV SHOWS UNITED STATES AND UNITED KINGDOM ARE TO
```





In [110]:
```
# INSIGHT 6

#movies and seasons
sns.lineplot(data=df_final_movies,
             x="release_year",
             y="rating",color="red")
plt.xlim(left=1995,right=2020)
sns.lineplot(data=df_final_season,
             x="release_year",
             y="rating",color="blue")
plt.xlim(left=1995,right=2020)
plt.legend(["movies","tv-shows"])
plt.title("release_year v/s rating")

# RESULT--
# IN CASE OF MOVIES RATINGS OVER LAST 25 YEARS REAMINS AT SAME LEVEL ALMOST
# IN CASE OF TV SHOWS RATING LEVEL INCREASE DRASTICALLY AND IT IS GOOD TO S
```

Out[110]:
```
Text(0.5, 1.0, 'release_year v/s rating')
```



Recommendations

1.Netfix should focus on more movies content then Tv-shows

2.Netflix should more focus for content from countries like united states, UK, japan, India

3.Netflix should also focus on rating of movies as for lst 25 years rating have no growth, so it is important to change movies content to increase level so that rating in coming years increase as same as ratings of tv-shows

4.Netflix should more focus on duaration of movies between 90-110 minutes and for tv-shows it should be 1-2 seasons for better growth

5.Netflix should upload more movies of actors like Alfred Molina,Liam Neeson, John Krasinski and for TV-SHOWS it should be of actors like Yuki Kaji,Takahiro Sakurai,David Attenborogh

6.Netflix for better revenue should ask directors to release shows more in months of january,july,october

7.Netflix should more focus on Comedy Genre type movies and international tv shows for TV-SHOWS

8.Netflix should upload more movies of directors like Youssef Chahine,Martin Scorsese,cathy garcia and for tv shows it should be Houda Benyamina,Thomas Astruc, Noam Murro

In [ ]: