In [1]:
```
import numpy as np
import pandas as pd
```

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

In [3]:
```
df.head()
```

Out[3]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	dura
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	Sea
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	Sea
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	Sea
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	Sea

1.Defining Problem Statement

```
1. Explore the Data to identify the Patterns/Insights from various attributes
of data for business improvements like recommendations and to increase
customizations
2. Visualizing how different parameters have correlation/dependency on another
parameters by Univariate, Bi-variate and Multivariate Analysis
```

1.Analyzing the basic metrics

Since it is the recommenadtion systems, the success/business metrics can be reviewed by analysing

1. Click-Through Rates
2. Salves and Revenue
3. User Behavior and Engagement
4. Basic Metrics [a. Precision b. Recall c.F1-measure d.False-positive rate e.Mean average precision f.Mean absolute error g.The area under the ROC curve (AUC)]

Data Cleaning

Data cleaning involves

1. Drop Unnecessary columns(In this case description)
2. Changing data column types(date str --> date datetime)
3. Adding Few required columns
4. Cleaning Corrupt data (date added < release.year)

In [4]:
```
# 1. Drop Unnecessary columns(In this case description)
# 2. changing data column types(date str --> date datetime)
df = df.drop(['description'], axis=1)
df['date_added'] = pd.to_datetime(df['date_added'])
df.head()
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	dura
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	Sea
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	Sea
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	Sea
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	Sea

In [5]:
```
# 3. Adding seasons columns
def season_split(season):
    season = str(season)
    if "Seasons" in season:
        season_list.append(int(season.split(" ")[0]))
    elif "Season" in season:
        season_list.append(int(season.split(" ")[0]))
    else:
        season_list.append(0)

season_list = []
df["duration"].apply(season_split)

df["seasons"] = season_list
```

In [6]:
```
df.head()
```

Out[6]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	dura
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	Sea
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	Sea
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	Sea
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	Sea

In [28]:
```
# 4. dropping corrupt data
df = df.drop(df[(pd.to_datetime(df['date_added']).dt.year < df['release_year'])].
df = df.drop(df[df['rating'].str.contains('min')].index)
df.shape
```

Out[28]:
```
(8776, 12)
```

5. Missing Value & Outlier check

In [29]:
```
# Missing values

df.director.fillna("Unknown", inplace=True)
df.cast.fillna("Unknown", inplace=True)
df.country.fillna("Unknown", inplace=True)
df.dropna(subset=["date_added", "rating"], inplace=True)

df.head()
```

Out[29]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	du
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown	United States	2021-09-25	2020	PG-13	
1	s2	TV Show	Blood & Water	Unknown	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	Se
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown	2021-09-24	2021	TV-MA	S
3	s4	TV Show	Jailbirds New Orleans	Unknown	Unknown	Unknown	2021-09-24	2021	TV-MA	S
4	s5	TV Show	Kota Factory	Unknown	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	Se

In [189]:
```
# outliers check on seasons and release year
cols = ['seasons', 'release_year'] # one or more

Q1 = df[cols].quantile(0.25)
Q3 = df[cols].quantile(0.75)
IQR = Q3 - Q1

df_outliers = df[((df[cols] < (Q1 - 1.5 * IQR)) |(df[cols] > (Q3 + 1.5 * IQR))).a
df_outliers.shape
```

Out[189]: (1141, 12)

2. Observations on the shape of data, data types

of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, statistical summary

In [190]:
```
df.head()
```

Out[190]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	du
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown	United States	2021-09-25	2020	PG-13	
1	s2	TV Show	Blood & Water	Unknown	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	Se
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown	2021-09-24	2021	TV-MA	S
3	s4	TV Show	Jailbirds New Orleans	Unknown	Unknown	Unknown	2021-09-24	2021	TV-MA	S
4	s5	TV Show	Kota Factory	Unknown	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	Se

In [191]:
```
# shape of data
df.shape
```

Out[191]:
```
(8776, 12)
```

In [192]:
```
# data types of all the attributes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8776 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       8776 non-null   object        
 1   type          8776 non-null   object        
 2   title         8776 non-null   object        
 3   director      8776 non-null   object        
 4   cast          8776 non-null   object        
 5   country       8776 non-null   object        
 6   date_added    8776 non-null   datetime64[ns]
 7   release_year  8776 non-null   int64         
 8   rating        8776 non-null   object        
 9   duration      8776 non-null   object        
 10  listed_in     8776 non-null   object        
 11  seasons       8776 non-null   int64         
dtypes: datetime64[ns](1), int64(2), object(9)
memory usage: 1.2+ MB
```

```
# conversion of categorical attributes to 'category' (If required)
After conversion, the data become huge so i did the conversion while analysing
it individually in plotting section ,please kindly check it
```

In [193]:
```
# missing value detection
df.isnull().sum()
```

Out[193]:
```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
seasons         0
dtype: int64
```

In [194]:
```
# statistical summary
df.describe()
```

Out[194]:

	release_year	seasons
count	8776.000000	8776.000000
mean	2014.175137	0.528145
std	8.830018	1.169851
min	1925.000000	0.000000
25%	2013.000000	0.000000
50%	2017.000000	0.000000
75%	2019.000000	1.000000
max	2021.000000	17.000000

In [195]:
```
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import plotly.figure_factory as ff
import seaborn as sns
import matplotlib.pyplot as plt
```

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

In [196]:
```
df['type'].value_counts()
```

Out[196]:
```
Movie      6124
TV Show    2652
Name: type, dtype: int64
```

In [197]:
```
df['type'].unique()
```

Out[197]:
```
array(['Movie', 'TV Show'], dtype=object)
```

In [198]:
```
df['country'].value_counts()
```

Out[198]:
```
United States                                     2803
India                                              972
Unknown                                            829
United Kingdom                                     418
Japan                                              243
                                                  ...
Bulgaria, United States, Spain, Canada               1
Cambodia                                             1
United States, United Kingdom, Canada, China         1
United States, United Kingdom, India                 1
United States, Venezuela                             1
Name: country, Length: 746, dtype: int64
```

In [199]:
```
df['country'].unique()
```

Out[199]:
```
array(['United States', 'South Africa', 'Unknown', 'India',
       'United States, Ghana, Burkina Faso, United Kingdom, Germany, Ethiopi
a',
       'United Kingdom', 'Germany, Czech Republic', 'Mexico', 'Turkey',
       'Australia', 'United States, India, France', 'Finland',
       'China, Canada, United States',
       'South Africa, United States, Japan', 'Nigeria', 'Japan',
       'Spain, United States', 'France', 'Belgium',
       'United Kingdom, United States', 'United States, United Kingdom',
       'France, United States', 'South Korea', 'Spain',
       'United States, Singapore', 'United Kingdom, Australia, France',
       'United Kingdom, Australia, France, United States',
       'United States, Canada', 'Germany, United States',
       'South Africa, United States', 'United States, Mexico',
       'United States, Italy, France, Japan',
       'United States, Italy, Romania, United Kingdom',
       'Australia, United States', 'Argentina, Venezuela',
       'United States, United Kingdom, Canada', 'China, Hong Kong',
       'Russia', 'Canada', 'Hong Kong', 'United States, China, Hong Kong',
```

In [200]:
```
df['rating'].value_counts()
```

Out[200]:
```
TV-MA       3195
TV-14       2156
TV-PG        860
R            799
PG-13        490
TV-Y7        332
TV-Y         305
PG           287
TV-G         220
NR            79
G             41
TV-Y7-FV       6
UR             3
NC-17          3
Name: rating, dtype: int64
```

In [201]:
```
df['rating'].unique()
```

Out[201]:
```
array(['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', 'TV-Y', 'TV-Y7', 'R',
       'TV-G', 'G', 'NC-17', 'NR', 'TV-Y7-FV', 'UR'], dtype=object)
```

In [202]:
```
duration_min_counts = df[df['duration'].str.contains('min')]
print("The Movie show counts are", len(duration_min_counts))

documentary_counts = df[~df['duration'].str.contains('min')]
print("The documentary counts are", len(documentary_counts))
```

```
The Movie show counts are 6124
The documentary counts are 2652
```

In [203]:
```
df['date_added'].dt.year.value_counts()
```

Out[203]:
```
2019    2012
2020    1876
2018    1645
2021    1498
2017    1184
2016     424
2015      82
2014      24
2011      13
2013      10
2012       3
2009       2
2008       2
2010       1
Name: date_added, dtype: int64
```

In [204]:
```
df['date_added'].dt.year.unique()
```

Out[204]:
```
array([2021, 2020, 2019, 2018, 2017, 2016, 2015, 2014, 2013, 2012, 2011,
       2009, 2008, 2010], dtype=int64)
```

In [205]:
```
directors_filtered = df[df.director != 'Unknown'].set_index('title').director.str
```

In [206]:
```
directors_filtered.value_counts()
```

Out[206]:
```
Rajiv Chilaka         22
Jan Suter             21
Raúl Campos           19
Marcus Raboy          16
Suhas Kadav           16
                      ..
Maïwenn                1
Mitch Gould            1
Steve Rash             1
Daniel Noah            1
Juan Carlos Medina     1
Length: 4988, dtype: int64
```

In [207]:
```
directors_filtered.unique()
```

Out[207]:
```
array(['Kirsten Johnson', 'Julien Leclercq', 'Mike Flanagan', ...,
       'Majid Al Ansari', 'Peter Hewitt', 'Mozez Singh'], dtype=object)
```

In [208]:
```
casts_filtered = df[df.cast != 'Unknown'].set_index('title').cast.str.split(', ',
casts_filtered.value_counts()
```

Out[208]:
```
Anupam Kher          43
Shah Rukh Khan       35
Julie Tejwani        33
Takahiro Sakurai     32
Naseeruddin Shah     32
                     ..
Muhammad Ali          1
Dozie Onyiriuka       1
Allison Williams      1
Heather Lawless       1
Leander Vyvey         1
Length: 36353, dtype: int64
```

In [209]:
```
casts_filtered.unique()
```

Out[209]:
```
array(['Ama Qamata', 'Khosi Ngema', 'Gail Mabalane', ..., 'Malkeet Rauni',
       'Anita Shabdish', 'Chittaranjan Tripathy'], dtype=object)
```

In [210]:
```
genre_filtered = df[df.listed_in != 'Unknown'].set_index('title').listed_in.str.s
genre_filtered.value_counts()
```

Out[210]:
```
International Movies            2752
Dramas                          2426
Comedies                        1674
International TV Shows          1346
Documentaries                    869
Action & Adventure               858
TV Dramas                        759
Independent Movies               756
Children & Family Movies         641
Romantic Movies                  616
Thrillers                        577
TV Comedies                      568
Crime TV Shows                   467
Kids' TV                         446
Docuseries                       393
Music & Musicals                 375
Romantic TV Shows                368
Horror Movies                    357
Stand-Up Comedy                  342
Reality TV                       254
British TV Shows                 252
Sci-Fi & Fantasy                 242
Sports Movies                    219
Anime Series                     174
Spanish-Language TV Shows        172
TV Action & Adventure            167
Korean TV Shows                  151
Classic Movies                   116
LGBTQ Movies                     102
TV Mysteries                      97
Science & Nature TV               92
TV Sci-Fi & Fantasy               83
TV Horror                         75
Cult Movies                       71
Anime Features                    71
Teen TV Shows                     69
Faith & Spirituality              65
TV Thrillers                      57
Stand-Up Comedy & Talk Shows      56
Movies                            53
Classic & Cult TV                 26
TV Shows                          16
dtype: int64
```

In [211]:
```
genre_filtered.unique()
```

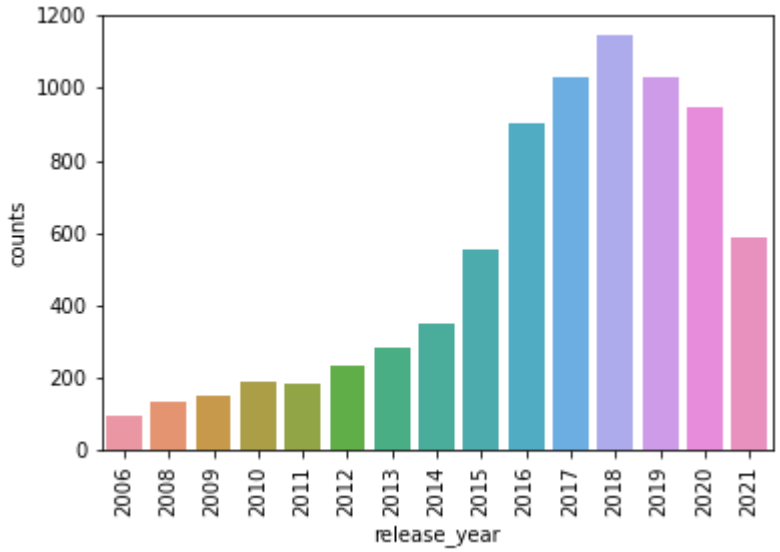
Out[211]:
```
array(['Documentaries', 'International TV Shows', 'TV Dramas',
       'TV Mysteries', 'Crime TV Shows', 'TV Action & Adventure',
       'Docuseries', 'Reality TV', 'Romantic TV Shows', 'TV Comedies',
       'TV Horror', 'Children & Family Movies', 'Dramas',
       'Independent Movies', 'International Movies', 'British TV Shows',
       'Comedies', 'Spanish-Language TV Shows', 'Thrillers',
       'Romantic Movies', 'Music & Musicals', 'Horror Movies',
       'Sci-Fi & Fantasy', 'TV Thrillers', "Kids' TV",
       'Action & Adventure', 'TV Sci-Fi & Fantasy', 'Classic Movies',
       'Anime Features', 'Sports Movies', 'Anime Series',
       'Korean TV Shows', 'Science & Nature TV', 'Teen TV Shows',
       'Cult Movies', 'TV Shows', 'Faith & Spirituality', 'LGBTQ Movies',
       'Stand-Up Comedy', 'Movies', 'Stand-Up Comedy & Talk Shows',
       'Classic & Cult TV'], dtype=object)
```

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

Univariate Analysis

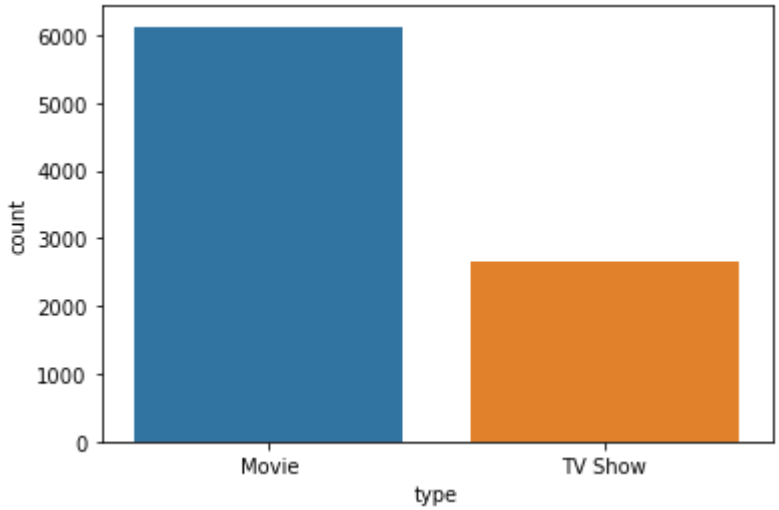
In [212]:
```
def univariate_analysis(col, top = 2):
    top_data = df.groupby(col).count().sort_values("type", ascending = False)[:to
    top_data["counts"] = top_data["type"]
    sns.barplot(data=top_data, x=col , y='counts')
    plt.xticks(rotation = 90);
```

In [213]:
```
univariate_analysis("release_year", 15)
# 6.3 comment: Movies are increasing every year being peak at 2018
```



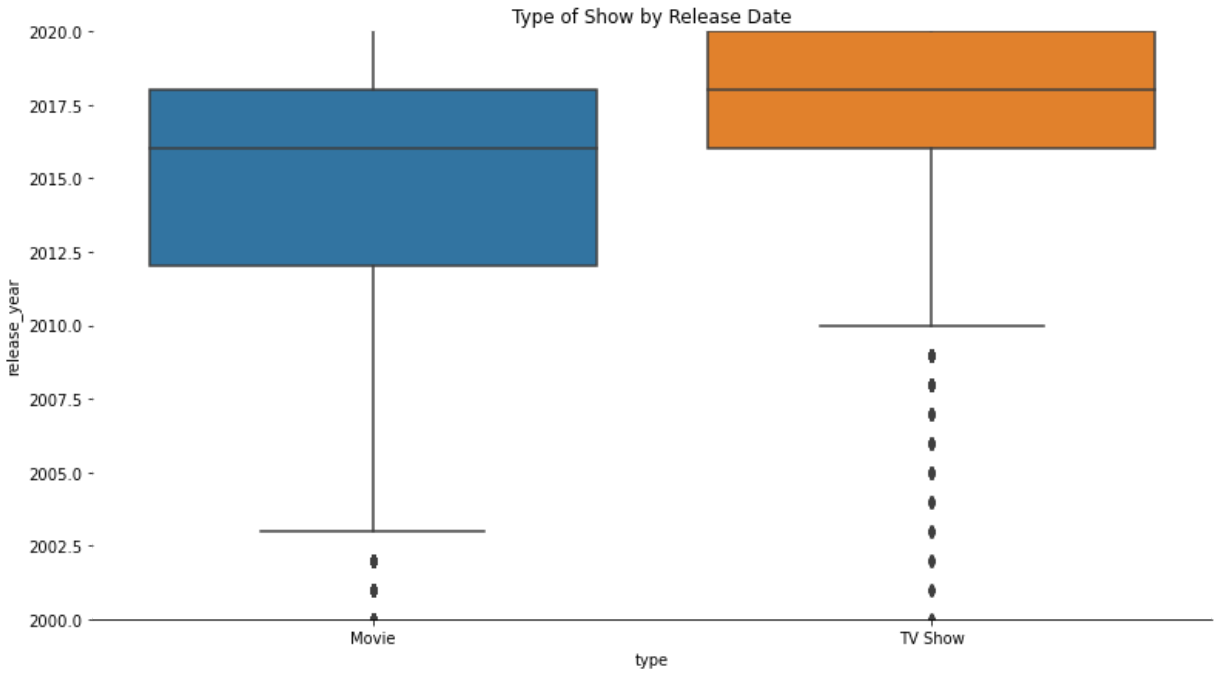
In [215]:
```
sns.countplot(x='type', data =df)
# 6.3 comment: 68.5% of data are movies and 31.5% are Tv shows
```

Out[215]: <matplotlib.axes._subplots.AxesSubplot at 0x19d7484f250>

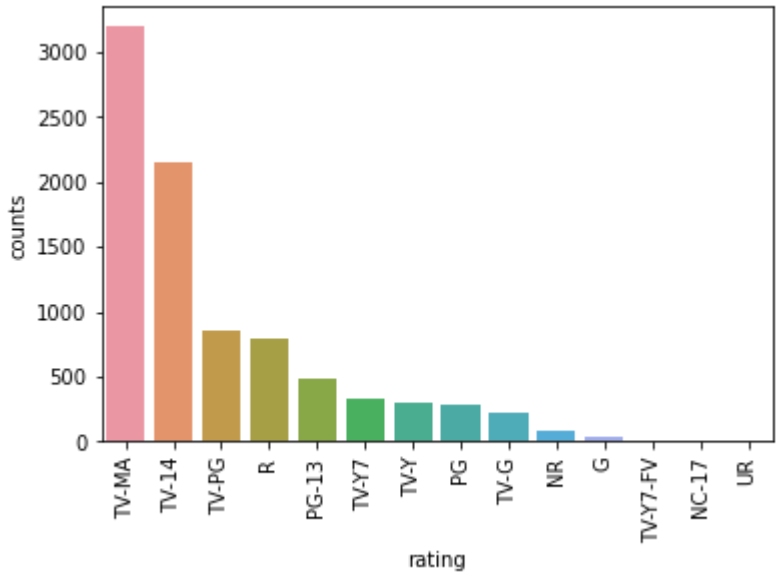


In [216]:
```
# Box Plot
plt.figure(figsize=(13,7))
sns.boxplot(x='type', y='release_year', data=df, )
sns.despine(left=True)
plt.title('Type of Show by Release Date')
plt.ylim(2000,2020)
```

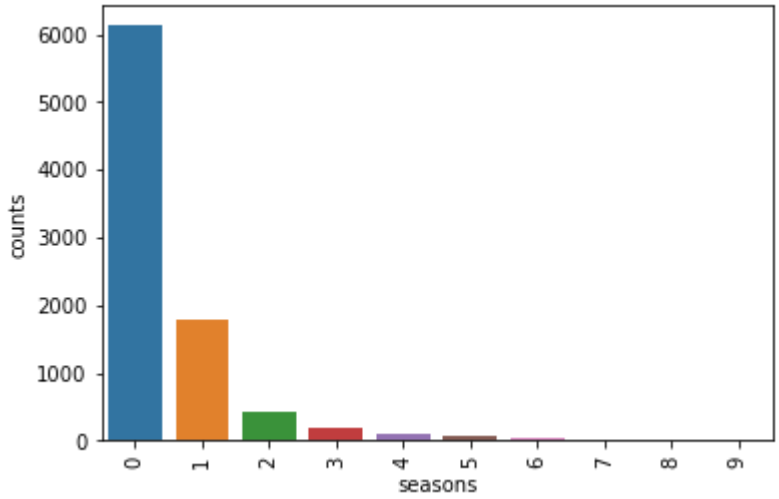
Out[216]: (2000.0, 2020.0)



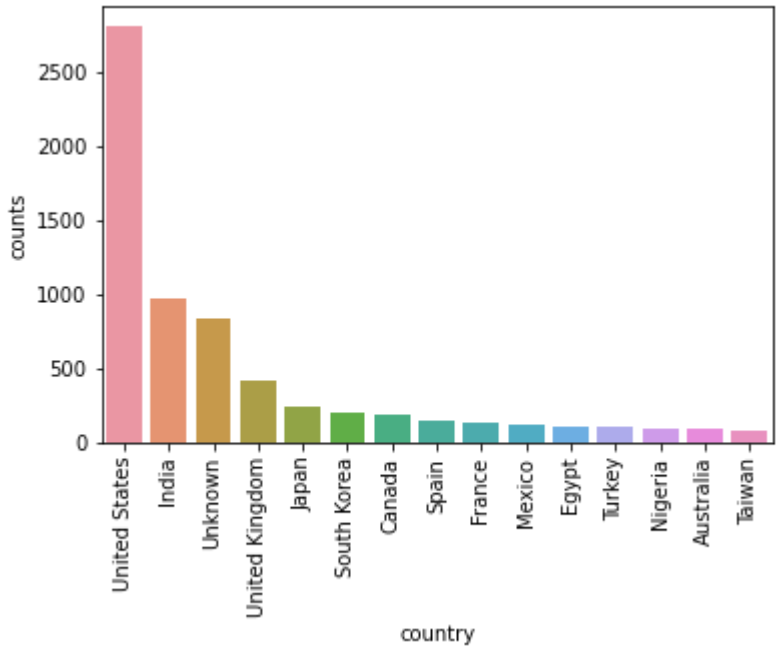
In [217]:
```
univariate_analysis("rating" , 15)
# 6.3 Comment: Tv-MA and TV-14 has more rating
```



In [218]:
```
univariate_analysis("seasons" , 10)
# 6.3 Comment: Tv shows with 1 season and 2 seasons are the most
```

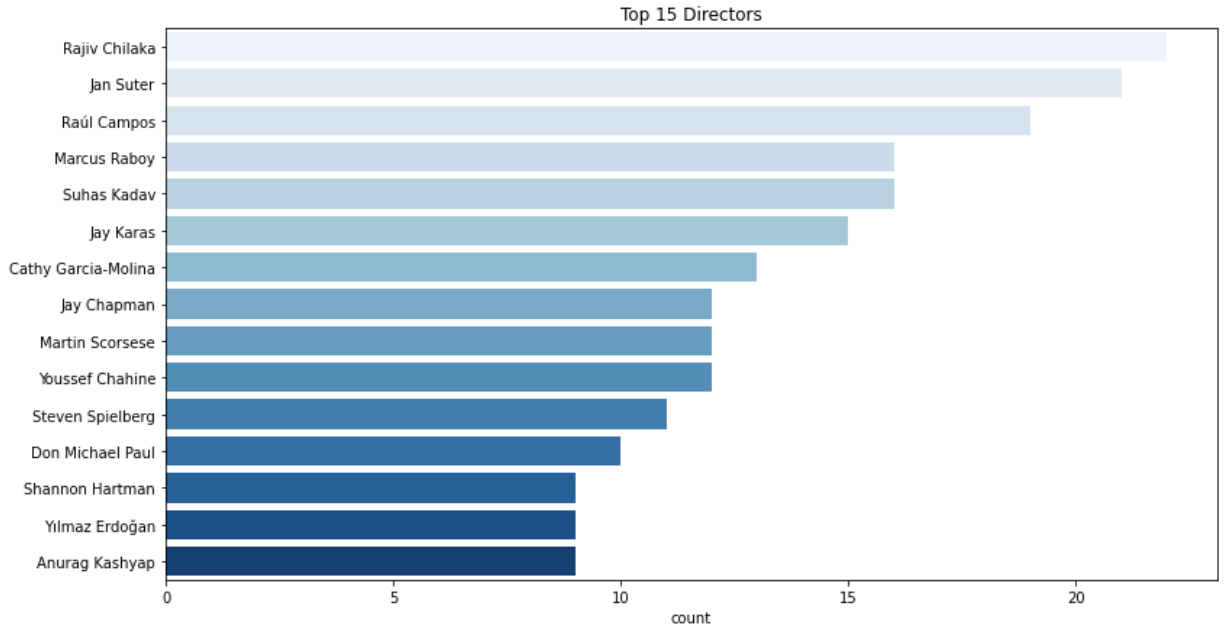


In [219]:
```
univariate_analysis("country" , 15)
# 6.3 Comment: US, Indian, UK and Japan has produced more movies
```

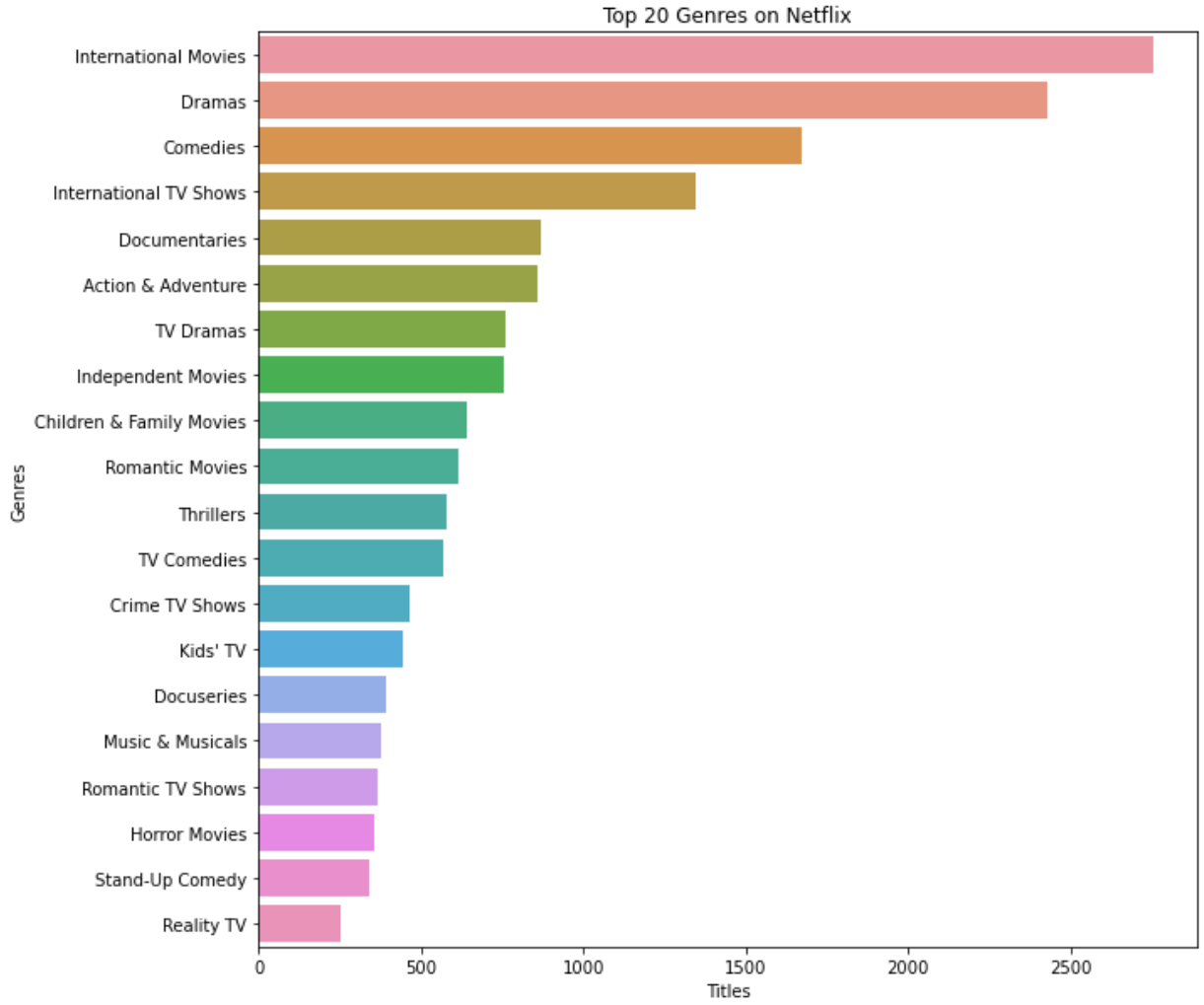


Bivariate Analysis

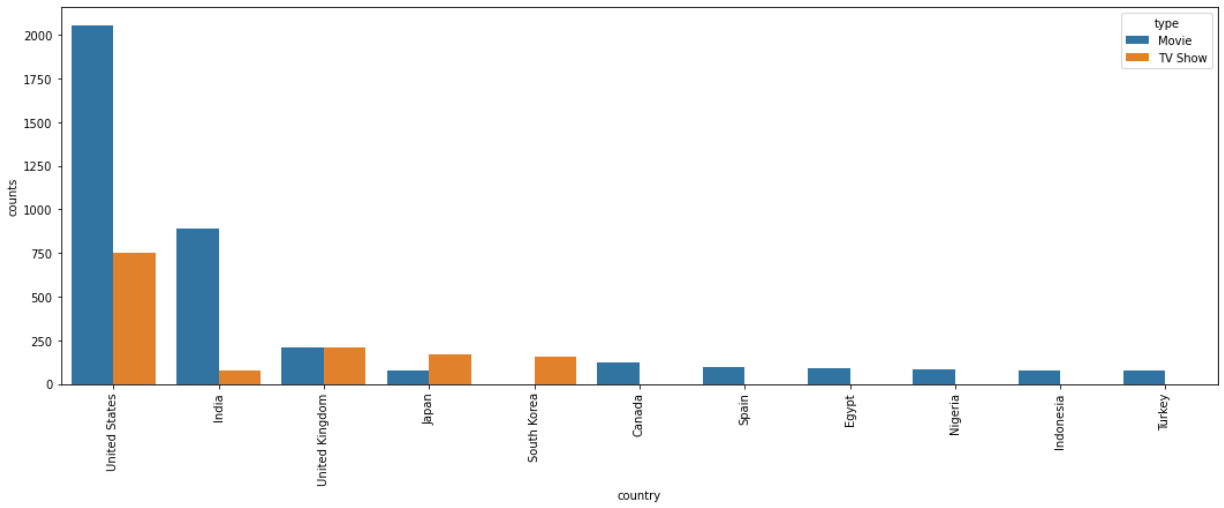
In [220]:
```
filtered_directors = df[df.director != 'Unknown'].set_index('title').director.str
plt.figure(figsize=(13,7))
plt.title('Top 15 Directors')
sns.countplot(y = filtered_directors, order=filtered_directors.value_counts().ind
plt.show()

# 6.3 Comment: The most popular director on Netflix, with the most titles, is mai
```



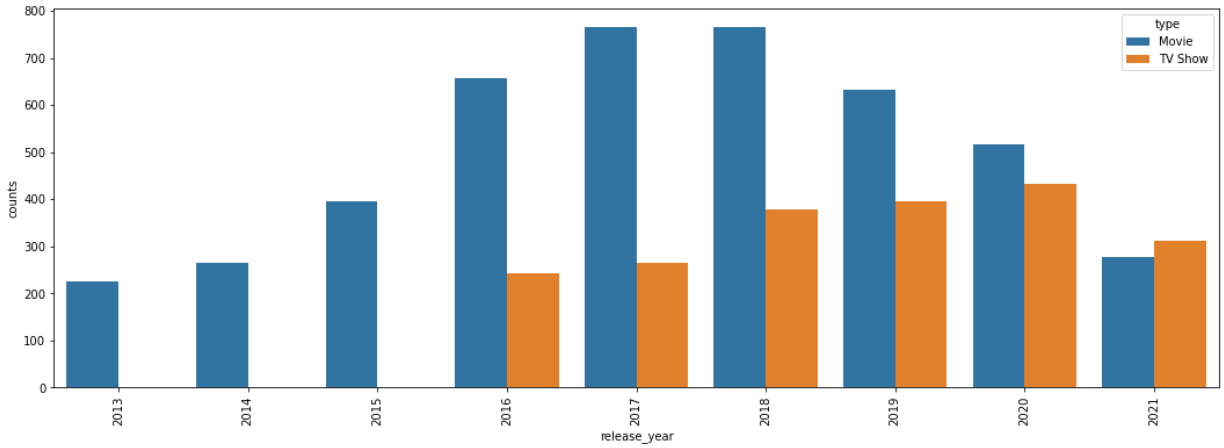
In [221]:
```
filtered_genres = df.set_index('title').listed_in.str.split(', ', expand=True).st
plt.figure(figsize=(10,10))
g = sns.countplot(y = filtered_genres, order=filtered_genres.value_counts().index
plt.title('Top 20 Genres on Netflix')
plt.xlabel('Titles')
plt.ylabel('Genres')
plt.show()

# 6.3 Comment: From the graph, we know that International Movies take the first p
```



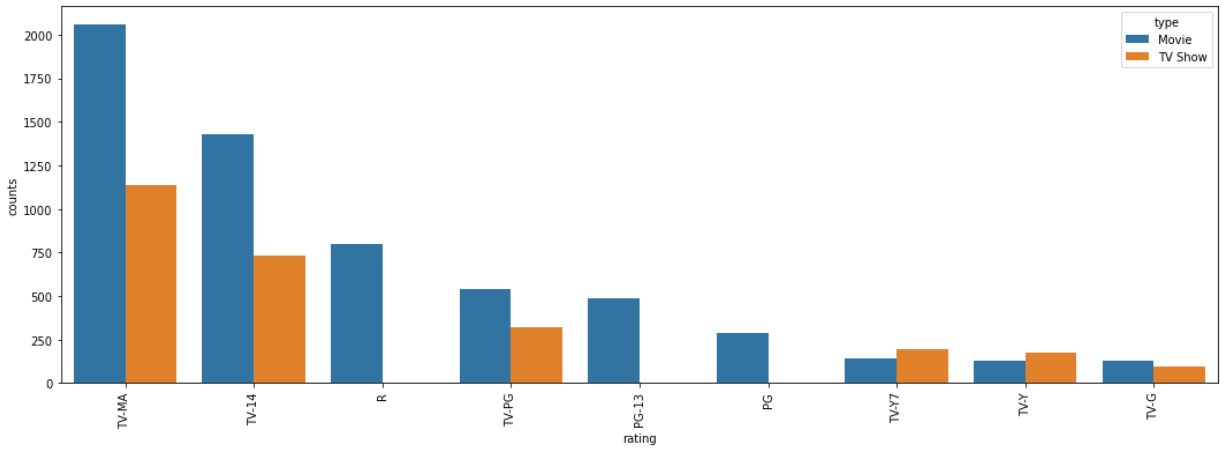
In [222]:
```
country_type = df[df.country != "Unknown"][['country','type']]
country_type = country_type.groupby(['country','type']).size().reset_index(name='

plt.figure(figsize = [18,6]);
sns.barplot(data=  country_type , x = 'country' , y = 'counts' , hue = 'type');
plt.xticks(rotation = 90);

# 6.3 Comment: japan has more TV shows than movies, only few countries has more m
```



In [223]:
```
release_year = df[['release_year','type']]
release_year = release_year.groupby(['release_year','type']).size().reset_index(r

plt.figure(figsize = [18,6]);
sns.barplot(data=  release_year , x = 'release_year' , y = 'counts' , hue = 'type
plt.xticks(rotation = 90);

# 6.3 comment: Every year more movies are getting released than TV shows
```



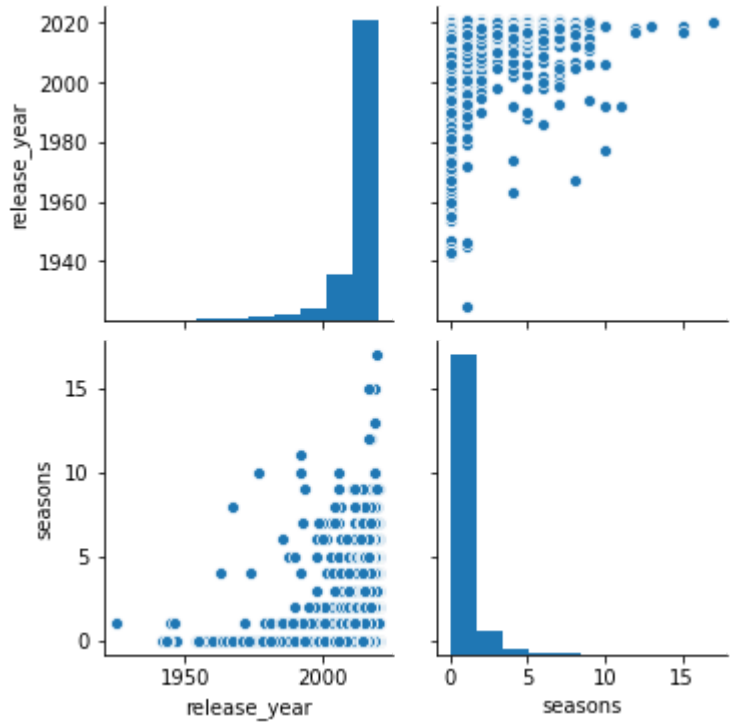
In [224]:
```
ating','type']]
.groupby(['rating','type']).size().reset_index(name='counts').sort_values(['counts

ize = [18,6]);
a=  rating , x = 'rating' , y = 'counts' , hue = 'type');
tion = 90);

TV-MA and TV-14 has more ratings than any other ratings
```



4.3 For correlation: Heatmaps, Pairplots

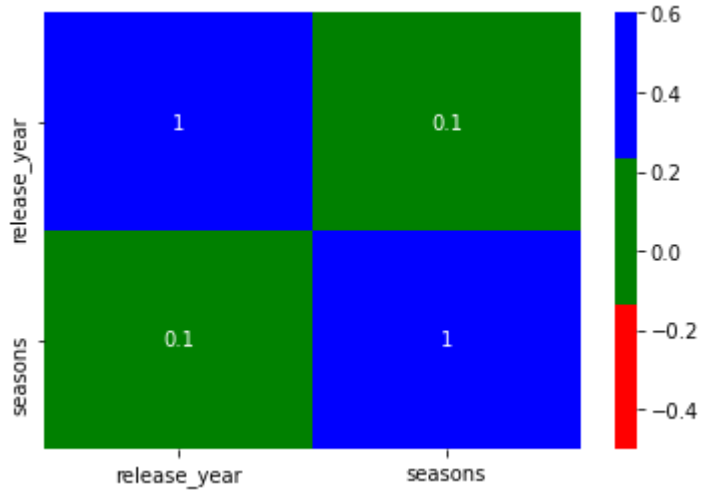
In [225]:
```
plt.figure(figsize=(30,30))
sns.pairplot(df)
```

Out[225]:
```
<seaborn.axisgrid.PairGrid at 0x19d72b3de50>

<Figure size 2160x2160 with 0 Axes>
```



In [226]:
```
sns.heatmap(df.corr(),cmap=['red','green','blue'],
vmin = -.5 , vmax = 0.6,annot = True)

# 6.3 comment: There is very little co-relation about release_year and season
```

Out[226]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x19d743ff1c0>
```



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

6.1 Comments on the range of attributes

1. So there are about 4,000++ movies and almost 2,000 TV shows, with movies being the majority. There are far more movie titles (68,5%) that TV shows titles (31,5%) in terms of title
2. About 1,000+ new movies were added in both 2019 and 2020. Besides, we can know that Netflix has increasingly focused on movies rather than TV shows in recent years
3. About 50% of movies/TV shows get's produced from US, Indian, UK and Japan
4. 47% of movies and Tv shows are coming from International Movies, Dramas, Comedies, International TV Shows and Documentaries

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

1. No of movies/Tv shows getting added to netflix is increasing being maximum in 2020 and 2019
2. No of movies/Tv shows getting released is also increasing being maximum in 2017 and 2018
3. US, Indian, UK and Japan has produced more movies
4. Rajiv chilka and jan suter had directed more movies than the others
5. The most popular actor on Netflix movie, based on the number of titles, is Anupam Kher.

6.2 Comments for each univariate and bivariate plot

1.I have commented it out in individual plots, please check it

In [ ]: