Netflix data analysis

Problem statement : Analyse the Netflix data and find the type of shows to produce and how to grow the business

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

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

In [4]:

```
df.shape
```

Out[4]:

```
(8807, 12)
```

In [5]:

```
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 [6]:

```
df.describe(include='all')
```

Out[6]:

	show_id	type	title	director	cast	country	date_added	release_year	ra
count	8807	8807	8807	6173	7982	7976	8797	8807.000000	8
unique	8807	2	8807	4528	7692	748	1767	NaN	
top	s2060	Movie	Tremors: Shrieker Island	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	NaN	
freq	1	6131	1	19	19	2818	109	NaN	3
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2014.180198	
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	8.819312	
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1925.000000	
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2013.000000	
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2017.000000	
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2019.000000	
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2021.000000	

In [8]:

```
df.isna().sum()
```

Out[8]:

```
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
```

TV Shows Vs Movies

In [9]:

```
df['type'].value_counts()
```

Out[9]:

```
Movie      6131
TV Show    2676
Name: type, dtype: int64
```

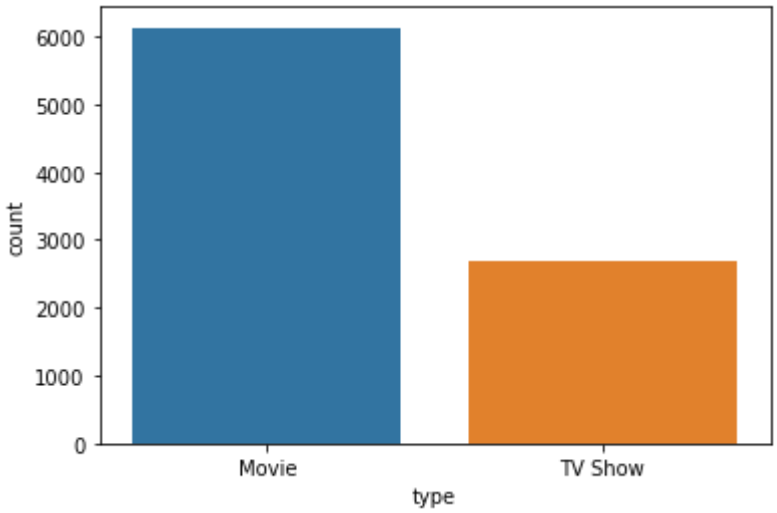
The type of show found to be more in number is Movie when compared to TV Shows.

In [13]:

```
sns.countplot(data=df, x='type')
```

Out[13]:

```
<AxesSubplot:xlabel='type', ylabel='count'>
```



In [91]:

```
df['year_added'] = pd.to_datetime(df['date_added'])
```

In [92]:

```
df['year_added'] = df['year_added'].dt.year
```

In [40]:

```
df.head()
```

Out[40]:

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

In [42]:

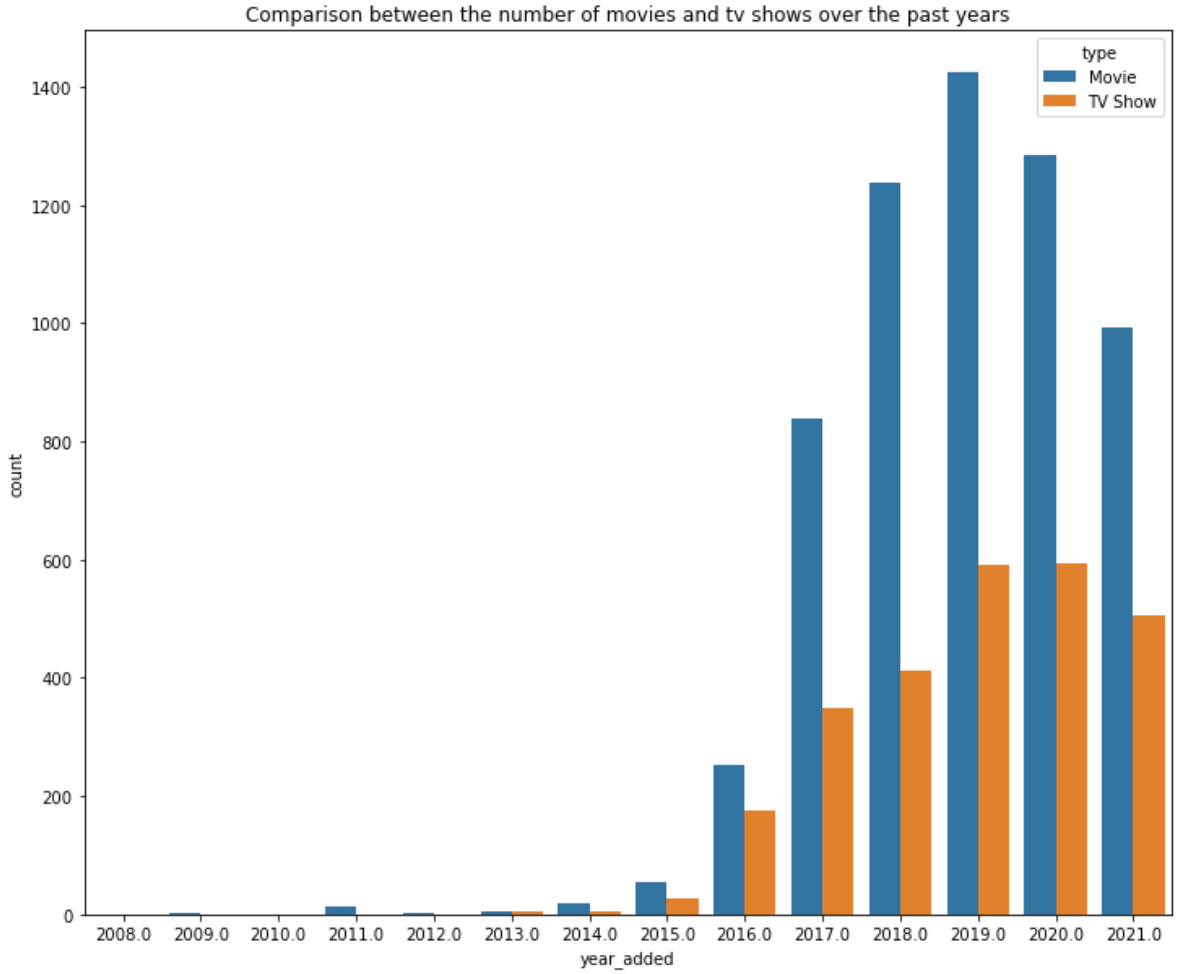
```
df.groupby(by='year_added')['type'].value_counts()
```

Out[42]:

```
year_added  type
2008.0      Movie         1
            TV Show       1
2009.0      Movie         2
2010.0      Movie         1
2011.0      Movie        13
2012.0      Movie         3
2013.0      Movie         6
            TV Show       5
2014.0      Movie        19
            TV Show       5
2015.0      Movie        56
            TV Show      26
2016.0      Movie       253
            TV Show     176
2017.0      Movie       839
            TV Show     349
2018.0      Movie      1237
            TV Show     412
2019.0      Movie      1424
            TV Show     592
2020.0      Movie      1284
            TV Show     595
2021.0      Movie       993
            TV Show     505
Name: type, dtype: int64
```

In [41]:

```
plt.figure(figsize=(12,10))
sns.countplot(x='year_added', hue='type', data=df)
plt.title('Comparison between the number of movies and tv shows over the past years')
plt.show()
```



Over the past years, the highest number of movies and tv shows were added to Netflix in the year 2019. Even though the tv shows are having increased popularity these days, movies are still higher in number.

Type of content available in different countries

In [43]:

```
df['country'].value_counts().sort_values(ascending=False)
```

Out[43]:

```
United States                              2818
India                                       972
United Kingdom                              419
Japan                                       245
South Korea                                 199
                                           ...
United Kingdom, China                         1
United States, Germany, Mexico                1
United States, Australia, Mexico              1
France, Belgium, Luxembourg, Cambodia,        1
Denmark, France, United States, Sweden        1
Name: country, Length: 748, dtype: int64
```

In [44]:

```
df['country'].nunique()
```

Out[44]:

```
748
```

Among the 748 countries, let's consider the top countries from above analysis - United States, India, United Kingdom, Japan, South Korea

In [46]:

```
top_countries = df[(df['country']=='United States') | (df['country']=='India') | (df['count
                  | (df['country']=='Japan') | (df['country']=='South Korea')]
```

In [49]:

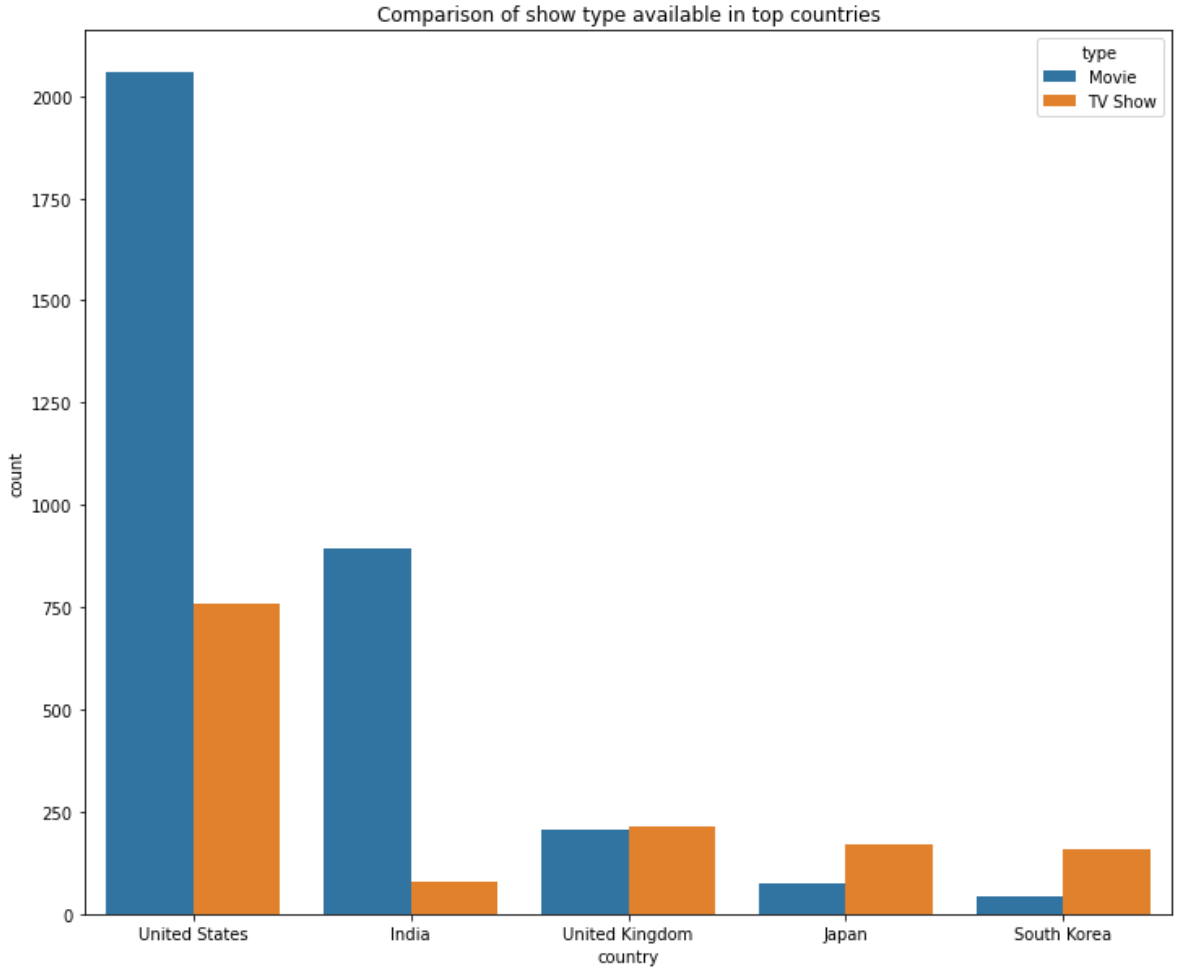
```
top_countries['country'].value_counts()
```

Out[49]:

```
United States     2818
India              972
United Kingdom     419
Japan              245
South Korea        199
Name: country, dtype: int64
```

In [52]:

```
plt.figure(figsize=(12,10))
sns.countplot(x='country', hue='type', data=top_countries)
plt.title('Comparison of show type available in top countries')
plt.show()
```



United States has the highest no. of movies and tv shows and India has the second highest. Rest three countries have lesser in number when compared to United states and India. Also, these 3 countries, UK, Japan and South Korea is found to be more focused on TV shows than movies.

Content update analysis

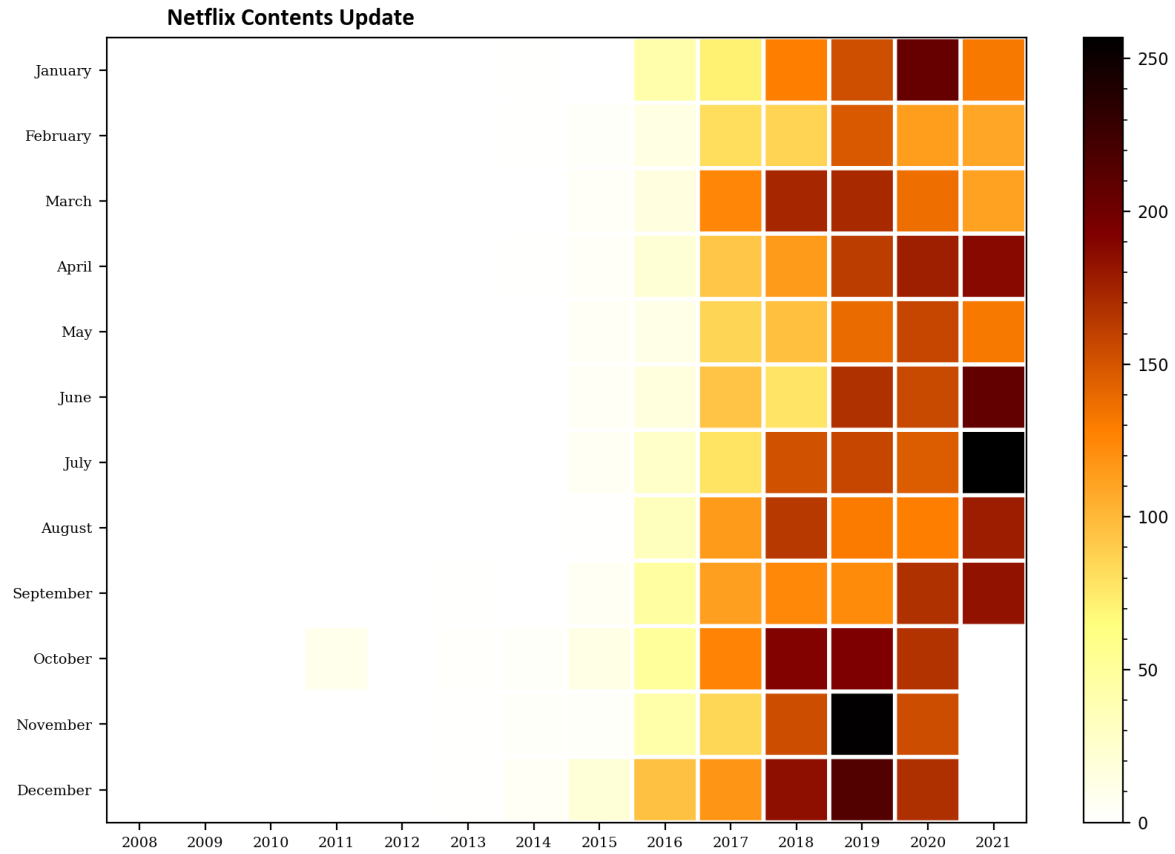
In [55]:

```
date_update = df[['date_added']].dropna()
date_update['year'] = date_update['date_added'].apply(lambda x : x.split(', ')[-1])
date_update['month'] = date_update['date_added'].apply(lambda x : x.lstrip().split(' ')[0])

months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'Septem
content_update = date_update.groupby('year')['month'].value_counts().unstack().fillna(0)[mo
plt.figure(figsize=(10, 7), dpi=200)
plt.pcolor(content_update, cmap='afmhot_r', edgecolors='white', linewidths=2)
plt.xticks(np.arange(0.5, len(content_update.columns), 1), content_update.columns, fontsize
plt.yticks(np.arange(0.5, len(content_update.index), 1), content_update.index, fontsize=7,

plt.title('Netflix Contents Update', fontsize=12, fontfamily='calibri', fontweight='bold',
cbar = plt.colorbar()

cbar.ax.tick_params(labelsize=8)
cbar.ax.minorticks_on()
plt.show()
```



From 2018 to 2020, frequent content update is found from the month of October to January. But in 2021, more content update is seen starting from mid year.

Type of ratings

In [57]:

```
df['rating'].value_counts()
```

Out[57]:

```
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
66 min         1
84 min         1
Name: rating, dtype: int64
```

In [58]:

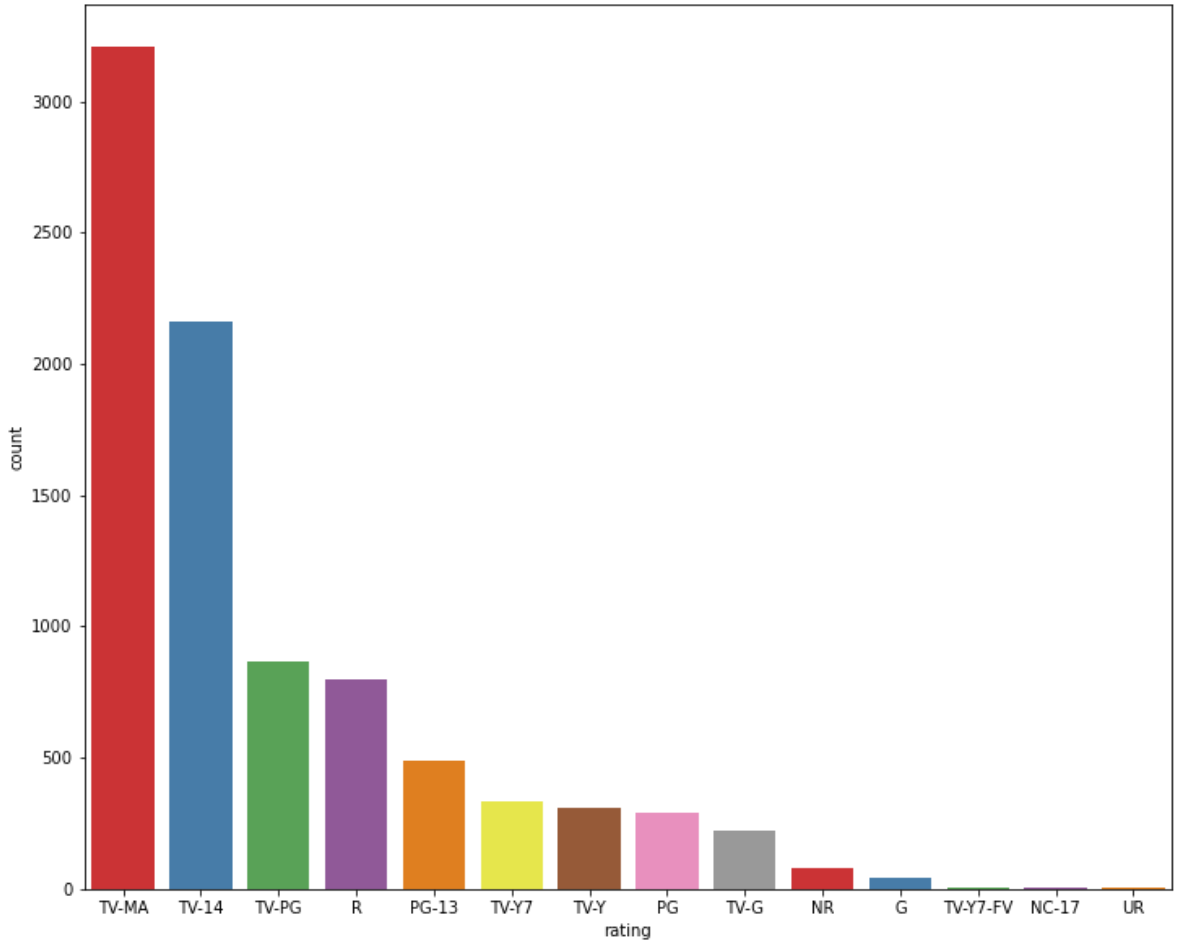
```
df['rating'].nunique()
```

Out[58]:

```
17
```

In [61]:

```
plt.figure(figsize=(12,10))
sns.countplot(x="rating", data=df, palette="Set1", order=df['rating'].value_counts().index[
plt.show()
```



Among the 14 types of ratings, TV-MA(TV Mature Audience Only) has the highest category followed by TV-14(Parents Strongly Cautioned) and TV-PG(Parental Guidance Suggested). This shows that Netflix has more content for Adults.

Ratings vs Show type

In [68]:

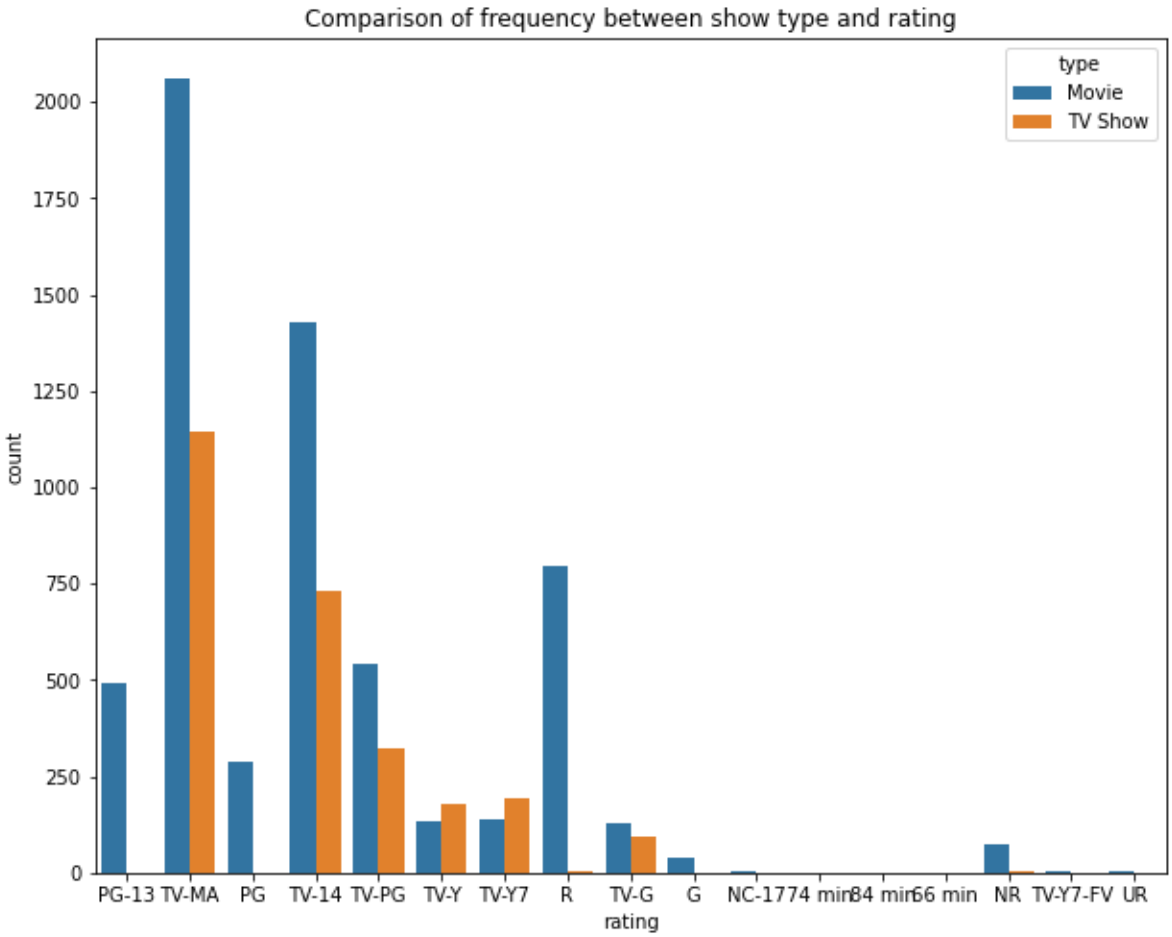
```
df.groupby(by='type')['rating'].value_counts()
```

Out[68]:

```
type     rating
Movie    TV-MA       2062
         TV-14       1427
         R            797
         TV-PG        540
         PG-13        490
         PG           287
         TV-Y7        139
         TV-Y         131
         TV-G         126
         NR            75
         G             41
         TV-Y7-FV       5
         NC-17          3
         UR             3
         66 min         1
         74 min         1
         84 min         1
TV Show  TV-MA       1145
         TV-14        733
         TV-PG        323
         TV-Y7        195
         TV-Y         176
         TV-G          94
         NR             5
         R              2
         TV-Y7-FV       1
Name: rating, dtype: int64
```

In [65]:

```
plt.figure(figsize=(10,8))
sns.countplot(x='rating',hue='type',data=df)
plt.title('Comparison of frequency between show type and rating')
plt.show()
```



TV-MA, TV-14, TV-PG, TV-Y, TV-G only have both movie and tv shows. And the other ratings have only show type has movie. It says that Movies is the majority category.

Analysis of actors/directors of different types of shows/movies

As the cast, director and listed_in columns have nested data, let's unnest them

In [70]:

```
cast_constraint=df['cast'].apply(lambda x : str(x).split(', ')).tolist()
cast_df = pd.DataFrame(cast_constraint, index=df['title'])
cast_df = cast_df.stack()
cast_df = pd.DataFrame(cast_df)
cast_df.reset_index()
```

Out[70]:

	title	level_1	0
0	Dick Johnson Is Dead	0	nan
1	Blood & Water	0	Ama Qamata
2	Blood & Water	1	Khosi Ngema
3	Blood & Water	2	Gail Mabalane
4	Blood & Water	3	Thabang Molaba
...	...	...	...
64946	Zubaan	3	Manish Chaudhary
64947	Zubaan	4	Meghna Malik
64948	Zubaan	5	Malkeet Rauni
64949	Zubaan	6	Anita Shabdish
64950	Zubaan	7	Chittaranjan Tripathy

64951 rows × 3 columns

In [71]:

```
director_constraint=df['director'].apply(lambda x : str(x).split(', ')).tolist()
director_df = pd.DataFrame(director_constraint, index=df['title'])
director_df = director_df.stack()
director_df = pd.DataFrame(director_df)
director_df.reset_index()
```

Out[71]:

	title	level_1	0
0	Dick Johnson Is Dead	0	Kirsten Johnson
1	Blood & Water	0	nan
2	Ganglands	0	Julien Leclercq
3	Jailbirds New Orleans	0	nan
4	Kota Factory	0	nan
...	...	...	...
9607	Zodiac	0	David Fincher
9608	Zombie Dumb	0	nan
9609	Zombieland	0	Ruben Fleischer
9610	Zoom	0	Peter Hewitt
9611	Zubaan	0	Mozez Singh

9612 rows × 3 columns

In [72]:

```
listedin_constraint = df['listed_in'].apply(lambda x : str(x).split(', ')).tolist()
listedin_df = pd.DataFrame(listedin_constraint, index=df['title'])
listedin_df = listedin_df.stack()
listedin_df = pd.DataFrame(listedin_df)
listedin_df.reset_index()
```

Out[72]:

	title	level_1	0
0	Dick Johnson Is Dead	0	Documentaries
1	Blood & Water	0	International TV Shows
2	Blood & Water	1	TV Dramas
3	Blood & Water	2	TV Mysteries
4	Ganglands	0	Crime TV Shows
...	...	...	...
19318	Zoom	0	Children & Family Movies
19319	Zoom	1	Comedies
19320	Zubaan	0	Dramas
19321	Zubaan	1	International Movies
19322	Zubaan	2	Music & Musicals

19323 rows × 3 columns

In [73]:

```
cast_df = cast_df.reset_index()
cast_df.drop(columns='level_1', inplace=True)
director_df = director_df.reset_index()
director_df.drop(columns='level_1', inplace=True)
listedin_df = listedin_df.reset_index()
listedin_df.drop(columns='level_1', inplace=True)
```

In [77]:

```
from functools import reduce
merged_df = reduce(lambda x,y: pd.merge(x,y, on='title', how='inner'), [cast_df, director_d
```

Out[77]:

	title	0_x	0_y	0
0	Dick Johnson Is Dead	nan	Kirsten Johnson	Documentaries
1	Blood & Water	Ama Qamata	nan	International TV Shows
2	Blood & Water	Ama Qamata	nan	TV Dramas
3	Blood & Water	Ama Qamata	nan	TV Mysteries
4	Blood & Water	Khosi Ngema	nan	International TV Shows

In [78]:

```
merged_df.columns = ['title','cast','director','listed_in']
merged_df.head()
```

Out[78]:

	title	cast	director	listed_in
0	Dick Johnson Is Dead	nan	Kirsten Johnson	Documentaries
1	Blood & Water	Ama Qamata	nan	International TV Shows
2	Blood & Water	Ama Qamata	nan	TV Dramas
3	Blood & Water	Ama Qamata	nan	TV Mysteries
4	Blood & Water	Khosi Ngema	nan	International TV Shows

In [79]:

```
merged_df.shape
```

Out[79]:

```
(161216, 4)
```

In [80]:

```
netflix_df = reduce(lambda x,y: pd.merge(x,y, on='title', how='inner'), [merged_df, df])
```

In [82]:

```
netflix_df.drop(columns=['director_y','cast_y','listed_in_y','description'], inplace=True)
```

In [84]:

```
netflix_df['cast_x'].value_counts()
```

Out[84]:

```
nan                 1733
Anupam Kher          120
Shah Rukh Khan        99
Naseeruddin Shah      98
Radhika Apte          89
                    ...
Denzel Whitaker        1
Nanao                  1
Grant McFarland        1
Jennifer L. Yen        1
Nicole Richie          1
Name: cast_x, Length: 36440, dtype: int64
```

Netflix has shows of the actors Anupam Kher, Shah Rukh Khan, Naseeruddin Shah and Radhika Apte more in number when compared other thousands of actors.

In [85]:

```
netflix_df['director_x'].value_counts()
```

Out[85]:

```
nan                     44621
Cathy Garcia-Molina       356
Youssef Chahine           288
Martin Scorsese           273
David Dhawan              270
                        ...
Caio Cobra                  1
Matthew Cooke               1
Marc-Antoine Hélard         1
Jon Olb                     1
David Batra                 1
Name: director_x, Length: 4994, dtype: int64
```

Cathy Garcia-Molina, Youssef Chahine, Martin Scorsese, David Dhawan are the top directors

In [86]:

```
netflix_df['listed_in_x'].value_counts()
```

```
Sci-Fi & Fantasy                2603
Anime Series                    2126
Documentaries                   1937
Spanish-Language TV Shows       1853
TV Action & Adventure           1796
British TV Shows                1434
Sports Movies                   1275
TV Mysteries                    1138
Korean TV Shows                 1086
Classic Movies                  1085
Anime Features                   928
TV Sci-Fi & Fantasy              866
TV Horror                        841
Cult Movies                      781
Docuseries                       753
Teen TV Shows                    742
LGBTQ Movies                     727
TV Thrillers                     650
Reality TV                       578
```

In [87]:

```
netflix_df['listed_in_x'].nunique()
```

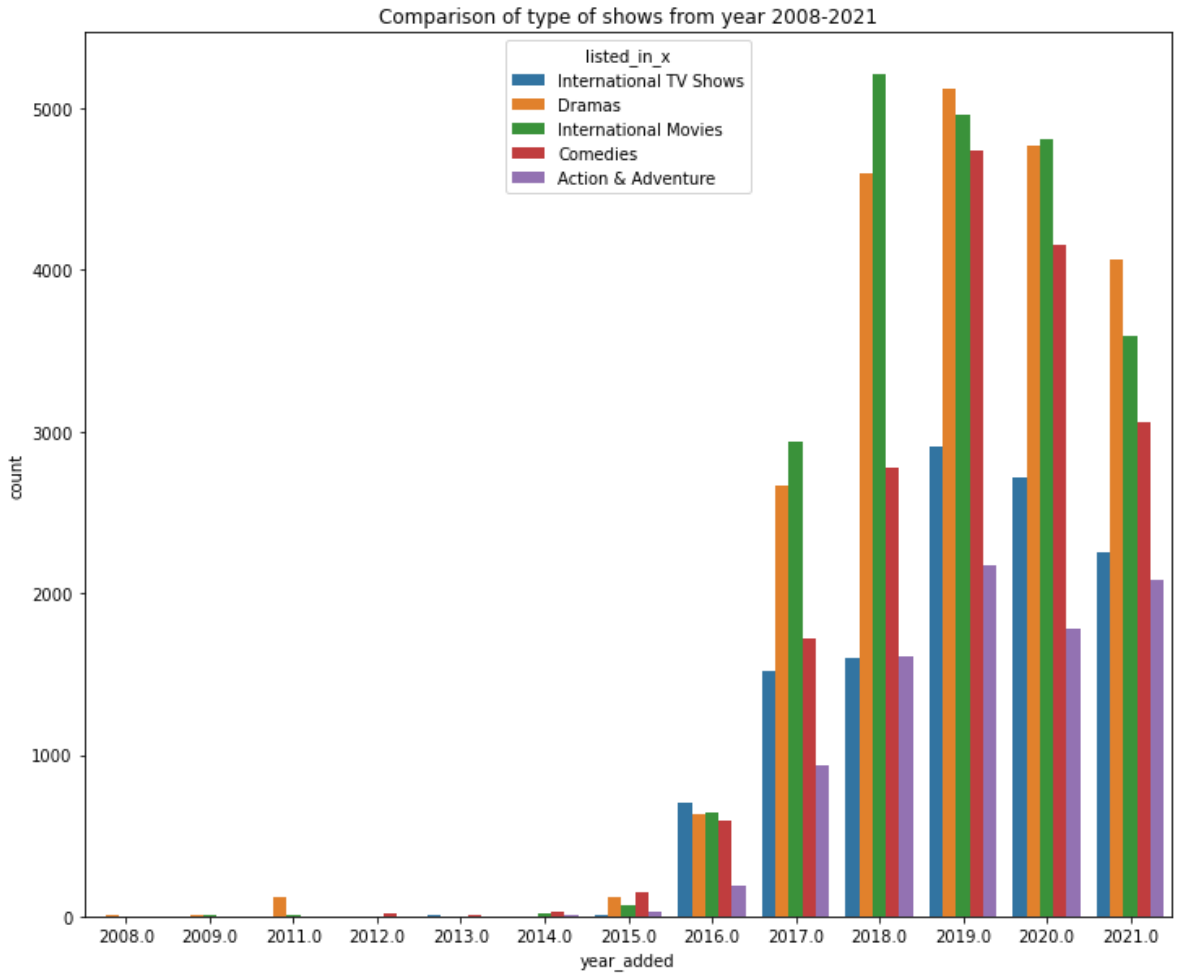
Out[87]:

```
42
```

Netflix has 42 types of shows. Among them Internation movies, dramas, comedies, International TV shows are the top categories.

In [99]:

```
top_categories = netflix_df[(netflix_df['listed_in_x']=='International Movies') | (netflix_
                 | (netflix_df['listed_in_x']=='International TV Shows') | (netflix_df['li
plt.figure(figsize=(12,10))
sns.countplot(x='year_added', hue='listed_in_x', data=top_categories)
plt.title('Comparison of type of shows from year 2008-2021')
plt.show()
```



Above graph clearly shows that Netflix should produce more shows in the above five categories to gain more business.

In [ ]: