In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("netflix.csv")
df.head()
```

Out[1]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	du
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	9
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	Se
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 S
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 S
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	Se

In [2]:
```
print('No. rows -> ',len(df),'\n')
print('Types of columns ->\n',df.dtypes)
```

```
No. rows ->  8807

Types of columns ->
 show_id         object
type             object
title            object
director         object
cast             object
country          object
date_added       object
release_year      int64
rating           object
duration         object
listed_in        object
description      object
dtype: object
```

In [3]:
```
# no. unique values in our data
for i in df.columns:
    print(i,':',df[i].nunique())
```

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

In [4]:
```
# checking null values in every column
df.isnull().sum()
```

Out[4]:
```
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 [5]:
```
# checking the occurance of each rating
df['rating'].value_counts()
```

Out[5]:
```
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
Name: rating, dtype: int64
```

In [6]:
```
df
```

Out[6]:

	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
...	...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R

	show_id	type	title	director	cast	country	date_added	release_year	rating
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14

8807 rows × 12 columns

In [7]:
```
c_director = df['director'].apply(lambda x: str(x).split(', ')).to_list()
df1 = pd.DataFrame(c_director, index=df['title'])
df1 = df1.stack()
df1 = pd.DataFrame(df1.reset_index())
df1.rename(columns={0:'Directors'},inplace=True)
df1.drop(['level_1'],axis=1,inplace=True)
df1
```

Out[7]:

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

9612 rows × 2 columns

In [8]:
```
# Repeating the same for cast
c_cast = df['cast'].apply(lambda x: str(x).split(', ')).to_list()
df2 = pd.DataFrame(c_cast, 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
```

Out[8]:

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

64951 rows × 2 columns

In [9]:
```
# Doing same with listed_in
c_listed = df['listed_in'].apply(lambda x: str(x).split(', ')).to_list()
df3 = pd.DataFrame(c_listed, index=df['title'])
df3 = df3.stack()
df3 = pd.DataFrame(df3.reset_index())
df3.rename(columns={0:'Genre'},inplace=True)
df3.drop(['level_1'],axis=1,inplace=True)
df3
```

Out[9]:

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

19323 rows × 2 columns

In [10]:
```
# Doing same with country
c_country = df['country'].apply(lambda x: str(x).split(', ')).to_list()
df4 = pd.DataFrame(c_country, index=df['title'])
df4 = df4.stack()
df4 = pd.DataFrame(df4.reset_index())
df4.rename(columns={0:'Country'},inplace=True)
df4.drop(['level_1'],axis=1,inplace=True)
df4
```

Out[10]:

	title	Country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	nan
3	Jailbirds New Orleans	nan
4	Kota Factory	India
...	...	...
10840	Zodiac	United States
10841	Zombie Dumb	nan
10842	Zombieland	United States
10843	Zoom	United States
10844	Zubaan	India

10845 rows × 2 columns

In [11]:
```
# merging director and actor
df5 = df2.merge(df1, on=['title'], how='inner')

# merging above data with genre
df6 = df5.merge(df3, on=['title'], how='inner')

# merging the above data woth country
df7 = df6.merge(df4, on=['title'], how='inner')

# replacing NaN values with 'unknown'
df7['Actors'].replace(['nan'],['Unknown Actor'], inplace=True)
df7['Directors'].replace(['nan'],['Unknown Directors'], inplace=True)
df7['Country'].replace(['nan'],[np.nan], inplace=True)

df7
```

Out[11]:

	title	Actors	Directors	Genre	Country
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States
1	Blood & Water	Ama Qamata	Unknown Directors	International TV Shows	South Africa
2	Blood & Water	Ama Qamata	Unknown Directors	TV Dramas	South Africa
3	Blood & Water	Ama Qamata	Unknown Directors	TV Mysteries	South Africa
4	Blood & Water	Khosi Ngema	Unknown Directors	International TV Shows	South Africa
...	...	...	...	...	...
201986	Zubaan	Anita Shabdish	Mozez Singh	International Movies	India
201987	Zubaan	Anita Shabdish	Mozez Singh	Music & Musicals	India
201988	Zubaan	Chittaranjan Tripathy	Mozez Singh	Dramas	India
201989	Zubaan	Chittaranjan Tripathy	Mozez Singh	International Movies	India
201990	Zubaan	Chittaranjan Tripathy	Mozez Singh	Music & Musicals	India

201991 rows × 5 columns

In [12]:
```
df_final = df7.merge(df[['show_id','type','title','date_added','release_year','rati
                    on=['title'], how='left')

df_final
```

Out[12]:

	title	Actors	Directors	Genre	Country	show_id	type	date_added	re
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	
1	Blood & Water	Ama Qamata	Unknown Directors	International TV Shows	South Africa	s2	TV Show	September 24, 2021	
2	Blood & Water	Ama Qamata	Unknown Directors	TV Dramas	South Africa	s2	TV Show	September 24, 2021	
3	Blood & Water	Ama Qamata	Unknown Directors	TV Mysteries	South Africa	s2	TV Show	September 24, 2021	
4	Blood & Water	Khosi Ngema	Unknown Directors	International TV Shows	South Africa	s2	TV Show	September 24, 2021	
...	...	...	...	...	...	...	...	...	
201986	Zubaan	Anita Shabdish	Mozez Singh	International Movies	India	s8807	Movie	March 2, 2019	
201987	Zubaan	Anita Shabdish	Mozez Singh	Music & Musicals	India	s8807	Movie	March 2, 2019	
201988	Zubaan	Chittaranjan Tripathy	Mozez Singh	Dramas	India	s8807	Movie	March 2, 2019	
201989	Zubaan	Chittaranjan Tripathy	Mozez Singh	International Movies	India	s8807	Movie	March 2, 2019	
201990	Zubaan	Chittaranjan Tripathy	Mozez Singh	Music & Musicals	India	s8807	Movie	March 2, 2019	

201991 rows × 11 columns

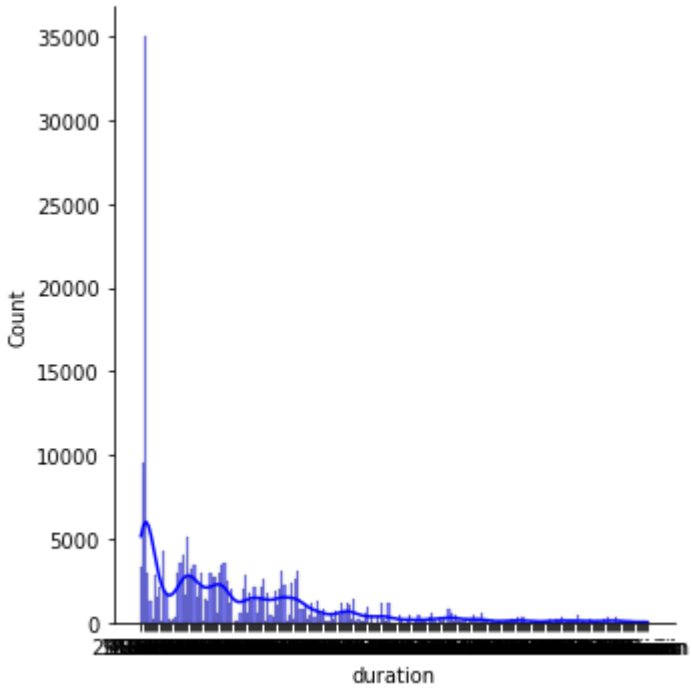
In [13]:
```
df_final.isnull().sum()
```

Out[13]:
```
title              0
Actors             0
Directors          0
Genre              0
Country        11897
show_id            0
type               0
date_added       158
release_year       0
rating            67
duration           3
dtype: int64
```

In [14]:
```
df_final['duration'].value_counts()
```

Out[14]:
```
1 Season      35035
2 Seasons      9559
3 Seasons      5084
94 min         4343
106 min        4040
               ...
3 min             4
5 min             3
11 min            2
8 min             2
9 min             2
Name: duration, Length: 220, dtype: int64
```

In [15]:
```
import seaborn as sns

sns.displot(data=df_final['duration'], kde=True,
bins = int(5), color = 'blue')

plt.show()
```



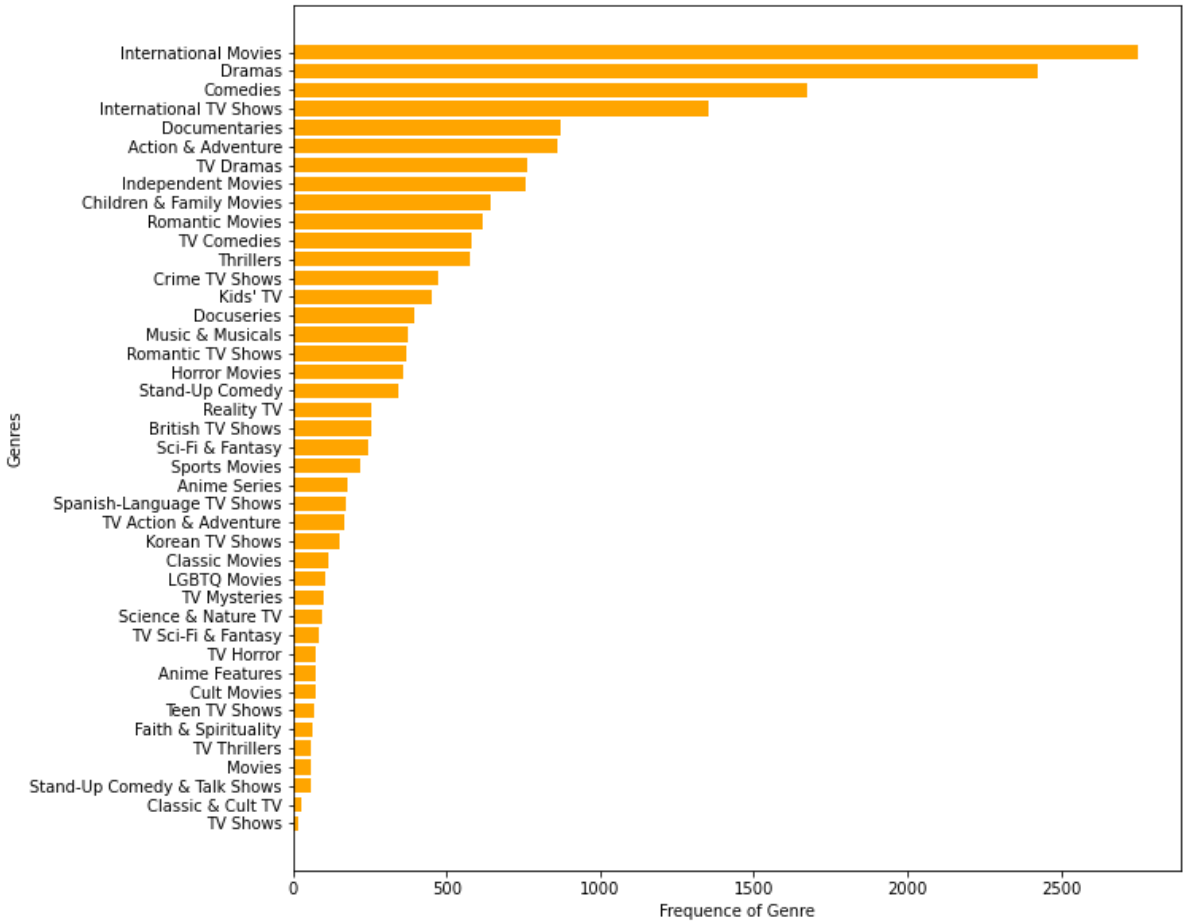
In [16]:
```
df_final.groupby(['Genre']).agg({'title':'nunique'})
```

Out[16]:

Genre	title
Action & Adventure	859
Anime Features	71
Anime Series	176
British TV Shows	253
Children & Family Movies	641
Classic & Cult TV	28
Classic Movies	116
Comedies	1674
Crime TV Shows	470
Cult Movies	71
Documentaries	869
Docuseries	395
Dramas	2427
Faith & Spirituality	65
Horror Movies	357
Independent Movies	756
International Movies	2752
International TV Shows	1351
Kids' TV	451
Korean TV Shows	151
LGBTQ Movies	102
Movies	57
Music & Musicals	375
Reality TV	255
Romantic Movies	616
Romantic TV Shows	370
Sci-Fi & Fantasy	243
Science & Nature TV	92
Spanish-Language TV Shows	174
Sports Movies	219
Stand-Up Comedy	343
Stand-Up Comedy & Talk Shows	56

Genre	title
TV Action & Adventure	168
TV Comedies	581
TV Dramas	763
TV Horror	75
TV Mysteries	98
TV Sci-Fi & Fantasy	84
TV Shows	16
TV Thrillers	57
Teen TV Shows	69
Thrillers	577

In [17]:
```
df_genre = df_final.groupby(['Genre']).agg({'title':'nunique'}).reset_index().sort_

plt.figure(figsize=(10,10))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'], color=['orange'])
plt.xlabel('Frequence of Genre')
plt.ylabel('Genres')
plt.show()
```

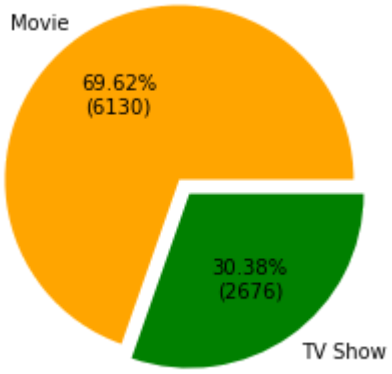


In [18]:
```
df_final.groupby(['type']).agg({'title':'nunique'}).reset_index()
```

Out[18]:

	type	title
0	Movie	6131
1	TV Show	2676

In [19]:
```
def func(pct, allvalues):
    absolute = int(pct / 100.*np.sum(allvalues))
    return f"{pct:.2f}%\n({absolute})"

df_type = df_final.groupby(['type']).agg({'title':'nunique'}).reset_index()
data = df_type['title'].to_list()
plt.pie(df_type['title'],explode=(0.05,0.05), labels=df_type['type'], colors=['oran
plt.show()
```



In [20]:
```
#number of distinct titles on the basis of countries
pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)
df_final.groupby(['Country']).agg({'title':'nunique'})
```

Out[20]:

Country	title
	2
Afghanistan	1
Albania	1
Algeria	3
Angola	1
Argentina	91
Armenia	1
Australia	160
Austria	12
Azerbaijan	1
Bahamas	1
Bangladesh	4
Belarus	1
Belgium	90
Bermuda	1
Botswana	1
Brazil	97
Bulgaria	10
Burkina Faso	1
Cambodia	5
Cambodia,	1
Cameroon	1
Canada	445
Cayman Islands	2
Chile	29
China	162
Colombia	52
Croatia	4
Cuba	1
Cyprus	1
Czech Republic	22
Denmark	48

Country	title
Dominican Republic	1
East Germany	1
Ecuador	1
Egypt	117
Ethiopia	1
Finland	11
France	393
Georgia	2
Germany	226
Ghana	5
Greece	11
Guatemala	2
Hong Kong	105
Hungary	11
Iceland	11
India	1046
Indonesia	90
Iran	4
Iraq	2
Ireland	46
Israel	30
Italy	100
Jamaica	1
Japan	318
Jordan	9
Kazakhstan	1
Kenya	6
Kuwait	8
Latvia	1
Lebanon	31
Liechtenstein	1
Lithuania	1

Country	title
Luxembourg	12
Malawi	1
Malaysia	26
Malta	3
Mauritius	2
Mexico	169
Mongolia	1
Montenegro	1
Morocco	6
Mozambique	1
Namibia	2
Nepal	2
Netherlands	50
New Zealand	33
Nicaragua	1
Nigeria	103
Norway	30
Pakistan	24
Palestine	1
Panama	1
Paraguay	1
Peru	10
Philippines	83
Poland	40
Poland,	1
Portugal	6
Puerto Rico	1
Qatar	10
Romania	14
Russia	27
Samoa	1
Saudi Arabia	13

Country	title
Senegal	3
Serbia	7
Singapore	41
Slovakia	1
Slovenia	3
Somalia	1
South Africa	62
South Korea	231
Soviet Union	3
Spain	232
Sri Lanka	1
Sudan	1
Sweden	42
Switzerland	19
Syria	3
Taiwan	89
Thailand	70
Turkey	113
Uganda	1
Ukraine	3
United Arab Emirates	37
United Kingdom	804
United Kingdom,	2
United States	3689
United States,	1
Uruguay	14
Vatican City	1
Venezuela	4
Vietnam	7
West Germany	5
Zimbabwe	3

In [21]:
```
#eliminate duplicates like united-states
df_final['Country'] = df_final['Country'].str.replace(',','')
df_country = df_final.groupby(['Country']).agg({'title':'nunique'})
#country wise unique shows
```

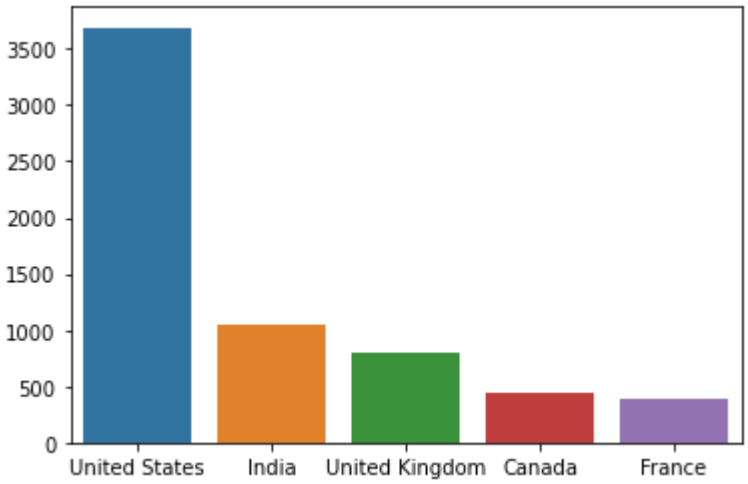
In [22]:
```
d_country = df_country.sort_values(by=['title'], ascending=False).head()['title'].t
d_country
# df = sns.load_dataset("df_final")
# sns.barplot(data=df, x="Country", y="title")
```

Out[22]:
```
{'United States': 3690,
 'India': 1046,
 'United Kingdom': 806,
 'Canada': 445,
 'France': 393}
```

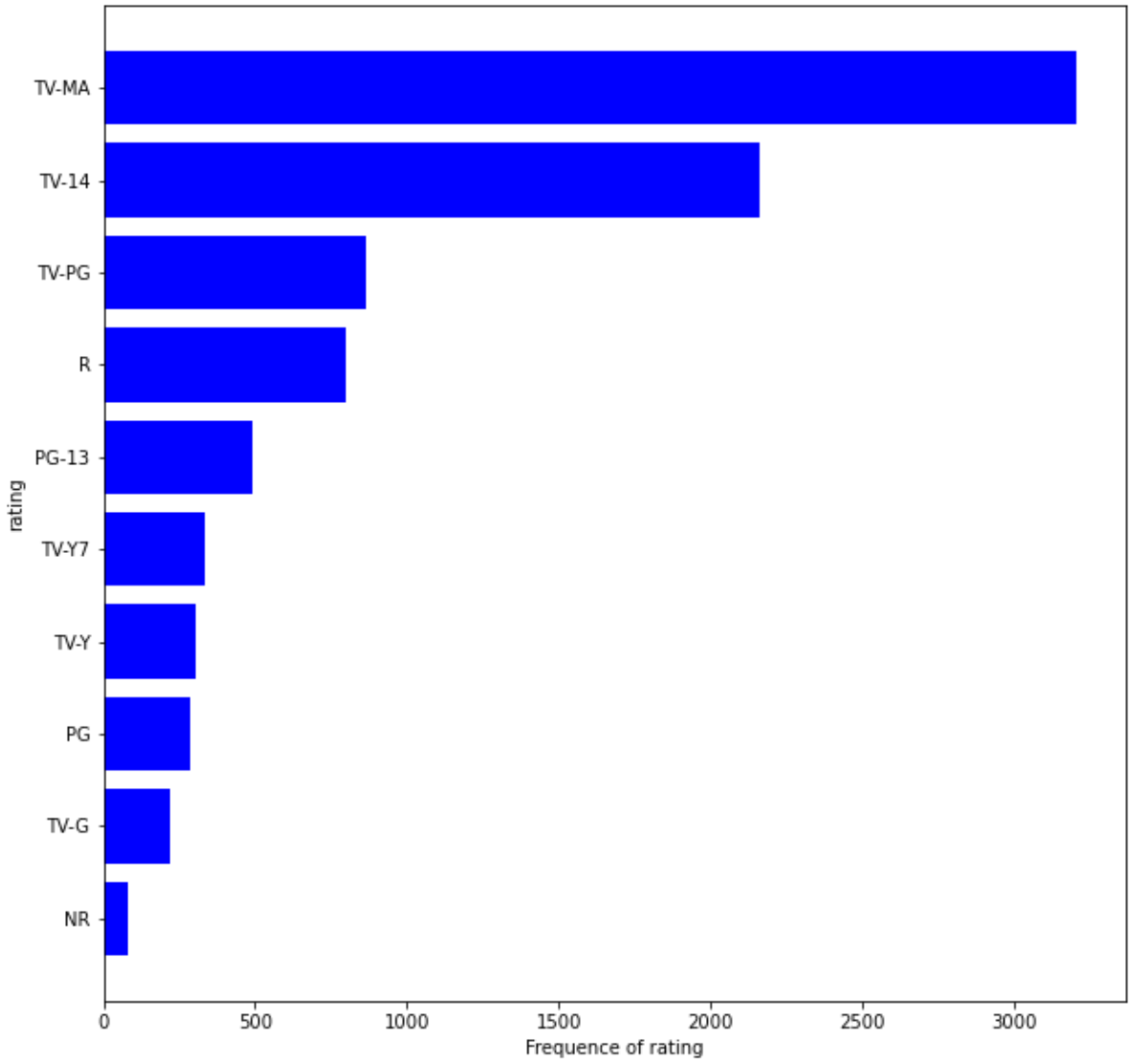
In [23]:
```
keys = list(d_country.keys())
vals = list(d_country.values())
print(keys,vals)
sns.barplot(x=keys, y=vals)
```

```
['United States', 'India', 'United Kingdom', 'Canada', 'France'] [3690, 1046, 806,
445, 393]
```

Out[23]:
```
<AxesSubplot:>
```

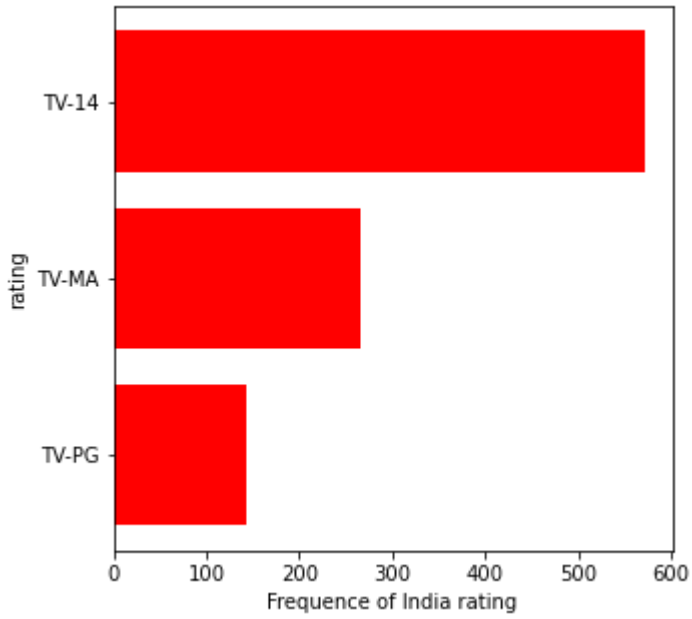


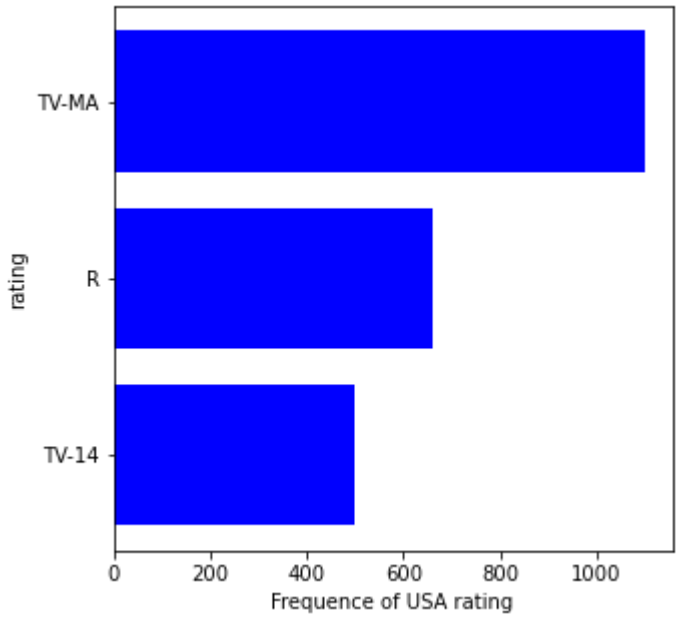
In [24]:
```
df_rating = df_final.groupby(['rating']).agg({'title':'nunique'}).reset_index().sor
df_rating = df_rating.head(10)
plt.figure(figsize=(10,10))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'], color=['blue'])
plt.xlabel('Frequence of rating')
plt.ylabel('rating')
plt.show()
#Top 10 rating wise categories
```



In [25]:

```
df_ind_rate = df_final[df_final['Country'] == 'India']
df_usa_rate = df_final[df_final['Country'] == 'United States']


df_rating = df_ind_rate.groupby(['rating']).agg({'title':'nunique'}).reset_index().
df_rating = df_rating.head(3)
plt.figure(figsize=(5,5))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'], color=['red'])
plt.xlabel('Frequence of India rating')
plt.ylabel('rating')
plt.show()

df_rating = df_usa_rate.groupby(['rating']).agg({'title':'nunique'}).reset_index().
df_rating = df_rating.head(3)
plt.figure(figsize=(5,5))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'], color=['blue'])
plt.xlabel('Frequence of USA rating')
plt.ylabel('rating')
plt.show()

#Top 10 rating wise categories
```





In [26]:
```
df_final.head()
```

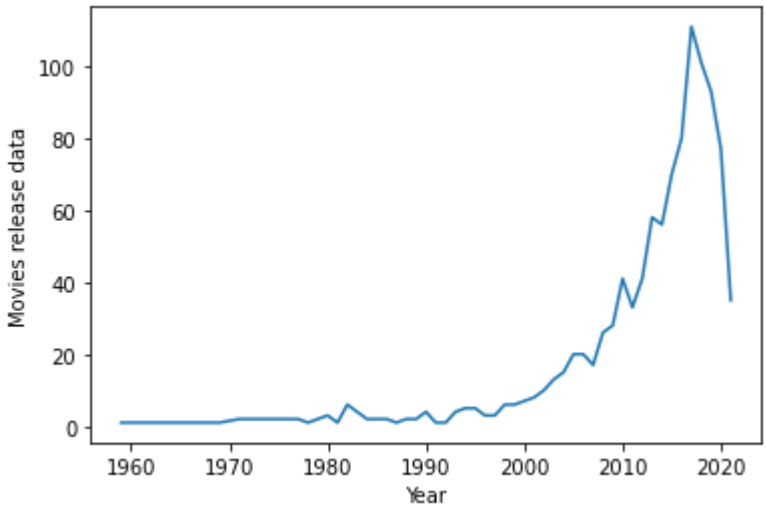
Out[26]:

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_yea
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	202
1	Blood & Water	Ama Qamata	Unknown Directors	International TV Shows	South Africa	s2	TV Show	September 24, 2021	202
2	Blood & Water	Ama Qamata	Unknown Directors	TV Dramas	South Africa	s2	TV Show	September 24, 2021	202
3	Blood & Water	Ama Qamata	Unknown Directors	TV Mysteries	South Africa	s2	TV Show	September 24, 2021	202
4	Blood & Water	Khosi Ngema	Unknown Directors	International TV Shows	South Africa	s2	TV Show	September 24, 2021	202

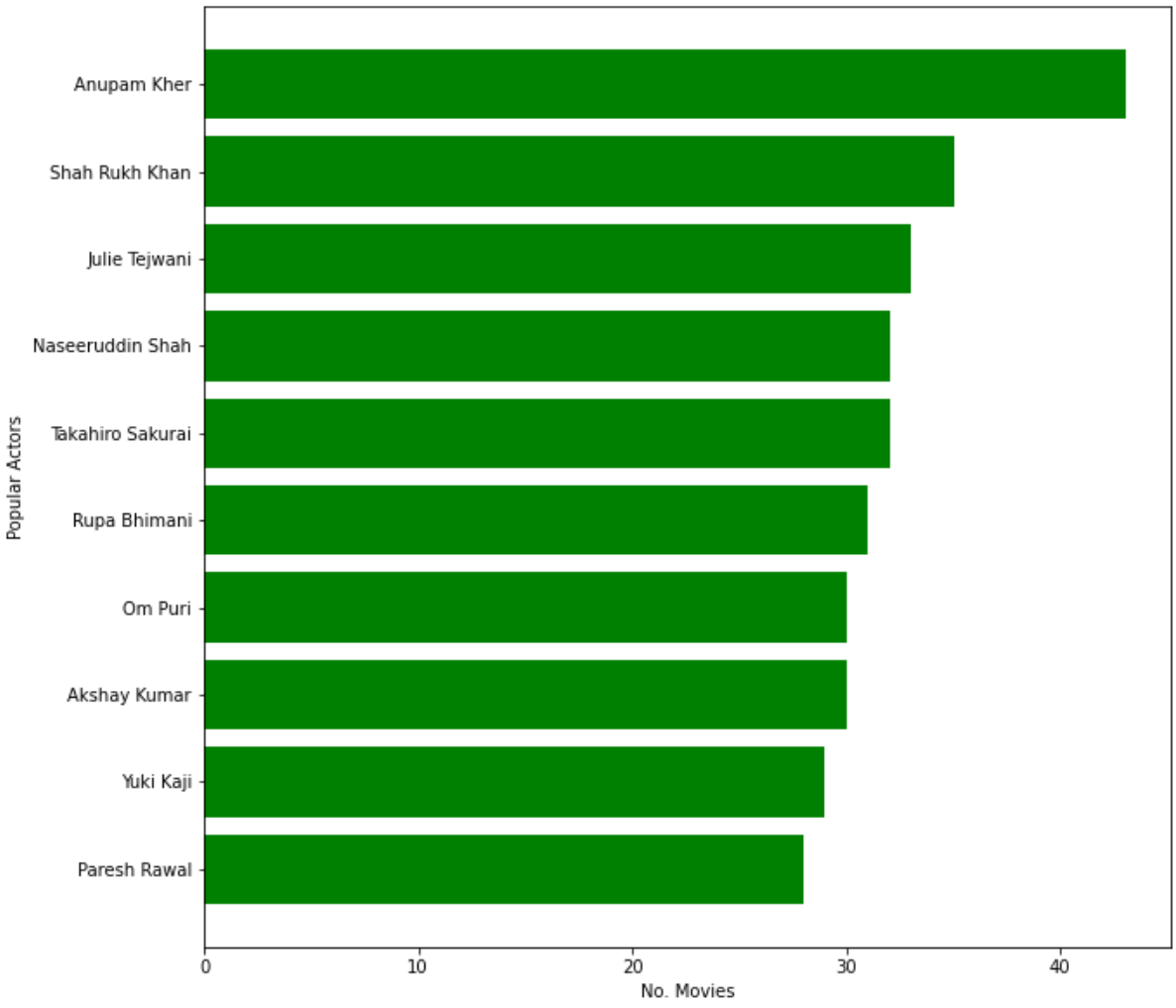
In [27]:
```
df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 201991 entries, 0 to 201990
Data columns (total 11 columns):
 #   Column        Non-Null Count   Dtype 
---  ------        --------------   ----- 
 0   title         201991 non-null  object
 1   Actors        201991 non-null  object
 2   Directors     201991 non-null  object
 3   Genre         201991 non-null  object
 4   Country       190094 non-null  object
 5   show_id       201991 non-null  object
 6   type          201991 non-null  object
 7   date_added    201833 non-null  object
 8   release_year  201991 non-null  int64 
 9   rating        201924 non-null  object
 10  duration      201988 non-null  object
dtypes: int64(1), object(10)
memory usage: 18.5+ MB
```

In [30]:
```
df_ind_show = df_final[df_final['Country'] == 'India']

df_release_year = df_ind_show.groupby(['release_year']).agg({'title':'nunique'}).re
sns.lineplot(data=df_release_year, x='release_year', y='title')
plt.ylabel('Movies release data')
plt.xlabel('Year')
plt.show()
# df_release_year

# The number of TV Shows in India reached a peak between 2010-2015. Since that
# it has reduced in 2015 from 2020. From 1960-2000 the graph is pretty steady and l
# but after the year 2000 no. releases started to sky rocket
```



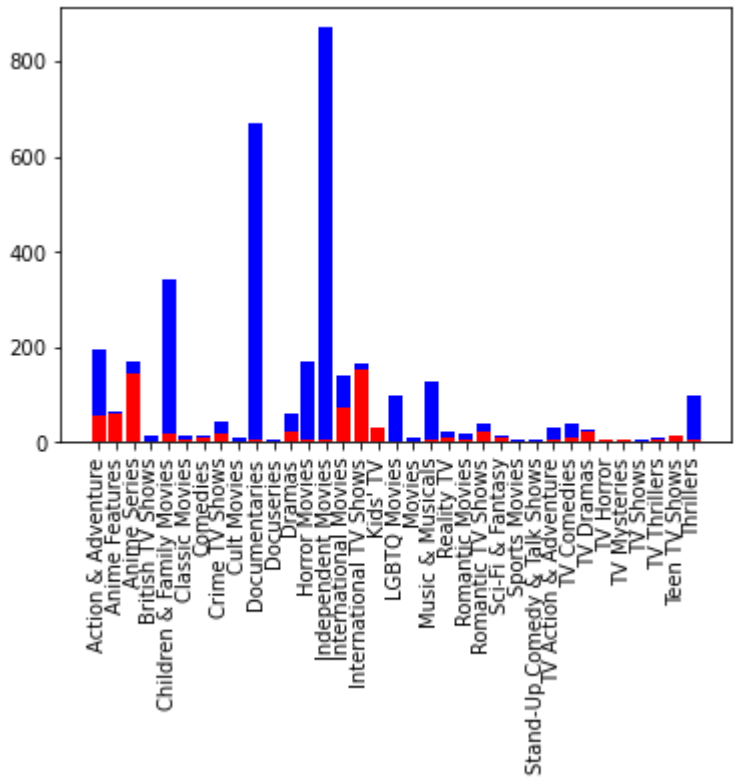
In [42]:
```
df_actor = df_final.groupby(['Actors']).agg({'title':'nunique'}).reset_index().sort
df_actor = df_actor[df_actor['Actors'] != 'Unknown Actor']
df_actor = df_actor.head(10).reset_index(drop=True)
plt.figure(figsize=(10,10))
plt.barh(df_actor[::-1]['Actors'], df_actor[::-1]['title'], color=['g'])
plt.xlabel('No. Movies')
plt.ylabel('Popular Actors')
plt.show()
```



In [32]:

```
xx = df_final['Genre'].to_list()
listxx = list(set(xx))
xx1 = listxx

df_jpn = df_final[df_final['Country']=='Japan'].groupby(['Genre']).agg({'title':'nu
yy1 = df_jpn['title'].to_list()
xx1 = df_jpn['Genre'].to_list()
# df_jpn = df_final[df_final['Country']=='South Korea'].groupby(['Genre']).agg({'ti
# yy2 = df_jpn['title'].to_list()

df_jpn = df_final[df_final['Country']=='India'].groupby(['Genre']).agg({'title':'nu
yy3 = df_jpn['title'].to_list()

print(xx1)
print(yy1)
# print(yy2)
print(yy3)

print(len(xx1))
print(len(yy1))
# print(len(yy2))
print(len(yy3))


# plot bars in stack manner
plt.bar(xx1, yy1, color='r')
plt.bar(xx1, yy3, bottom=yy1, color='b')
plt.xticks(rotation=90)

# plt.bar(xx1, yy3, bottom=yy2, color='g')
plt.show()

# # create data
# x = ['A', 'B', 'C', 'D']
# y1 = [10, 20, 10, 30]
# y2 = [20, 25, 15, 25]
# y3 = [20, 25, 15, 25]

# # plot bars in stack manner
# plt.bar(x, y1, color='r')
# plt.bar(x, y2, bottom=y1, color='b')
# plt.bar(x, y3, bottom=y2, color='g')
# plt.show()
```

```
['Action & Adventure', 'Anime Features', 'Anime Series', 'British TV Shows', 'Chil
dren & Family Movies', 'Classic Movies', 'Comedies', 'Crime TV Shows', 'Cult Movie
s', 'Documentaries', 'Docuseries', 'Dramas', 'Horror Movies', 'Independent Movie
s', 'International Movies', 'International TV Shows', "Kids' TV", 'LGBTQ Movies',
'Movies', 'Music & Musicals', 'Reality TV', 'Romantic Movies', 'Romantic TV Show
s', 'Sci-Fi & Fantasy', 'Sports Movies', 'Stand-Up Comedy & Talk Shows', 'TV Actio
n & Adventure', 'TV Comedies', 'TV Dramas', 'TV Horror', 'TV Mysteries', 'TV Show
s', 'TV Thrillers', 'Teen TV Shows', 'Thrillers']
[57, 61, 143, 1, 19, 3, 9, 16, 1, 7, 2, 23, 4, 7, 72, 151, 29, 1, 2, 6, 9, 7, 21,
9, 1, 1, 5, 10, 21, 5, 4, 1, 6, 14, 5]
[137, 3, 26, 11, 323, 9, 5, 27, 9, 662, 4, 35, 167, 864, 66, 12, 2, 96, 6, 120, 1
2, 12, 17, 6, 3, 5, 26, 28, 7, 2, 3, 3, 3, 1, 92]
35
35
35
```



In [33]:
```
# Recommendations

# 1. The most popular genres across the countries and in both TV shows and Movies a
# Show/Movies, so content aligning to that is recommended

# 2. USA, India & UK are the biggest consumers for netflix having unique shows with
# so its better to focus and optimize for these countries first.

# 3. The target audience in USA and India is recommended to be 14+ and above rating
# completely Mature/R content. Nevertheless both 14+ & Mature content are in top 3

# 4. Add movies for Indian Audience, it has been declining after 2015

# 5. Anime Series, Anime Features & Internation TV shows Genre for Japan and Docume
# Movies(Independent, Thriller, Music & Musicals, Horror) Romantic Genre in TV Show
```

```
# 6. While creating content, take into consideration the popular actors/directors f
# Anupam Kher  & Shahrukh Khan are most popular actors
```

In [ ]:
```
jupyter nbconvert --to webpdf --allow-chromium-download Untitled.ipynb
```