Importing Libraries

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

Uploading Dataset

```
from google.colab import files
uploaded = files.upload()
```

Choose Files netflix.csv

- **netflix.csv**(text/csv) - 3399671 bytes, last modified: 2/11/2023 - 100% done

```
Saving netflix.csv to netflix.csv
```

Preprocessing of Data

unnesting cast column

```
df = pd.read_csv("netflix.csv")
```

```
temp_df=df['cast'].apply(lambda x: str(x).split(', ')).tolist()
df_cast=pd.DataFrame(temp_df,index=df['title'])
df_cast=df_cast.stack()
df_cast=pd.DataFrame(df_cast)
df_cast.reset_index(inplace=True)
df_cast=df_cast[['title',0]]
df_cast.columns=['title','cast']
df_cast
```

	title	cast
0	Dick Johnson Is Dead	nan
1	Blood & Water	Ama Qamata

```
np.all(df_cast['title'].isin(df['title']))
```

```
True
```

4	Blood & Water	Thabang Molaba

Merging df_cast and df dataframe as "data1"

64946	Zubaan	Manish Chaudhary

```
data1=df.merge(df_cast,how="left",left_on="title",right_on="title")
data1.head()
```

	show_id	type	title	director	cast_x	country	date_added	release_year	ra
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	F
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	T
2	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	T
					Ama				

Dropping unnecessary columns and renaming cast_y column to cast

```
data1.drop(['cast_x'],axis=1,inplace=True)
data1.rename(columns={'cast_y':'cast'},inplace=True)
data1.head()
```

	show_id	type	title	director	country	date_added	release_year	rating	dura
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	September 25, 2021	2020	PG-13	90
1	s2	TV	Blood &	NaN	South	September	2021	TV-MA	

Unnesting Listed_in column

```
temp_df=df['listed_in'].apply(lambda x: str(x).split(', ')).tolist()
df_listed=pd.DataFrame(temp_df,index=df['title'])
df_listed=df_listed.stack()
df_listed=pd.DataFrame(df_listed)
df_listed.reset_index(inplace=True)
df_listed=df_listed[['title',0]]
df_listed.columns=['title','listed_in']
df_listed
```

	title	listed_in
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

```
np.all(df_listed['title'].isin(data1['title']))
```

```
True
```

Merging data1 and df_listed dataframe as "data2"

```
data2=data1.merge(df_listed,how="left",left_on="title",right_on="title")
data2.head()
```

	show_id	type	title	director	country	date_added	release_year	rating	dura
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	September 25, 2021	2020	PG-13	90
1	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Sea
2	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Sea
3	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Sea

Dropping unnecessary columns and renaming listed_in_y column to genre

```
data2.drop(['listed_in_x'],axis=1,inplace=True)
data2.rename(columns={'listed_in_y':'genre'},inplace=True)
data2.head()
```

	show_id	type	title	director	country	date_added	release_year	rating	dura
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	United States	September 25, 2021	2020	PG-13	90
1	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Sea
2	s2	TV Show	Blood & Water	NaN	South Africa	September 24, 2021	2021	TV-MA	Sea

Unnesting country column

```
temp_df=df['country'].apply(lambda x: str(x).split(', ')).tolist()
df_country=pd.DataFrame(temp_df,index=df['title'])
df_country=df_country.stack()
df_country=pd.DataFrame(df_country)
```

```
df_country=pd.DataFrame(df_country)
df_country.reset_index(inplace=True)
df_country=df_country[['title',0]]
df_country.columns=['title','country']
df_country
```

	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

```
np.all(df_country['title'].isin(data2['title']))
```

```
True
```

Merging data2 and df_country data frame as "data3"

```
data3=data2.merge(df_country,how="left",left_on="title",right_on="title")
data3.head()
```

	show_id	type	title	director	country_x	date_added	release_year	rating	du
0	s1	Movie	Johnson	Johnson	States	25, 2021	2020	PG-13	

Dropping unneccesary columns and renaming country_y to country

```
data3.drop(['country_x'],axis=1,inplace=True)
data3.rename(columns={'country_y':'country'},inplace=True)
data3.head()
```

	show_id	type	title	director	date_added	release_year	rating	duration	des
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	September 25, 2021	2020	PG-13	90 min	As nea
1	s2	TV Show	Blood & Water	NaN	September 24, 2021	2021	TV-MA	2 Seasons	par
2	s2	TV Show	Blood & Water	NaN	September 24, 2021	2021	TV-MA	2 Seasons	par

Unnesting director column

```
temp_df=data3['director'].apply(lambda x: str(x).split(', ')).tolist()
df_director=pd.DataFrame(temp_df,index=data3['title'])
df_director=df_director.stack()
df_director=pd.DataFrame(df_director)
df_director.reset_index(inplace=True)
df_director=df_director[['title',0]]
df_director.columns=['title','director']
df_director
```

	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	nan
2	Blood & Water	nan

```
np.all(df_director['title'].isin(data3['title']))
```

```
True
```

...	...	...

Merging data3 and df_director dataframe as "netflix_df"

201987	Zubaan	Mozez Singh

```
netflix_df = data3.merge(df_director,how="left",left_on="title",right_on="title")
netflix_df.head()
```

	show_id	type	title	director_x	date_added	release_year	rating	duration	d
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	September 25, 2021	2020	PG-13	90 min	A n
1	s2	TV Show	Blood & Water	NaN	September 24, 2021	2021	TV-MA	2 Seasons	p
2	s2	TV Show	Blood & Water	NaN	September 24, 2021	2021	TV-MA	2 Seasons	p

Dropping unneccesary columns and replacing director_y to director

```
netflix_df.drop(['director_x'],axis=1,inplace=True)
netflix_df.rename(columns={'director_y':'director'},inplace=True)
netflix_df.head()
```

	show_id	type	title	date_added	release_year	rating	duration	description
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min	As her father nears the end of his life, filmm...
1	s2	TV	Blood &	September	2021	TV-MA	2	paths at a

Checking for duplicate rows

```
netflix_df.duplicated()
```

```
0          False
1          False
2           True
3           True
4           True
           ...  
8417142     True
8417143     True
8417144     True
8417145     True
8417146     True
Length: 8417147, dtype: bool
```

Deleting duplicate rows

```
netflix_df = netflix_df.drop_duplicates()
np.any(netflix_df.duplicated())
```

```
False
```

Rearranging columns

```
netflix_df = netflix_df[["title","genre","type","director","date_added","release_year","ra
netflix_df.head()
```

	title	genre	type	director	date_added	release_year	rating	durati
0	Johnson	Documentaries	Movie	Kirsten Johnson	September 25, 2021	2020	PG-13	90 r

Missing value

```
netflix_df.isna().head()
```

	title	genre	type	director	date_added	release_year	rating	duration	cast
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
58	False	False	False	False	False	False	False	False	False
115	False	False	False	False	False	False	False	False	False
172	False	False	False	False	False	False	False	False	False

```
netflix_df.isna().sum()
```

```
title             0
genre             0
type              0
director          0
date_added      158
release_year      0
rating           67
duration          3
cast              0
country           0
description       0
dtype: int64
```

Replacing null with zero for numerical data

```
netflix_df.fillna(0)
```

	title	genre	type	director	date_added	release_year	rating	du
0	Dick Johnson Is Dead	Documentaries	Movie	Kirsten Johnson	September 25, 2021	2020	PG-13	
1	Blood & Water	International TV Shows	TV Show	nan	September 24, 2021	2021	TV-MA	
58	Blood & Water	TV Dramas	TV Show	nan	September 24, 2021	2021	TV-MA	
115	Water	TV Mysteries	Show	nan	24, 2021	2021	TV-MA	

Replacing null values for column with dtype - 'object'

```
netflix_df[['rating','duration']].fillna('-')
```

	rating	duration
0	PG-13	90 min
1	TV-MA	2 Seasons
58	TV-MA	2 Seasons
115	TV-MA	2 Seasons
172	TV-MA	2 Seasons
...	...	...
8417027	TV-14	111 min
8417051	TV-14	111 min
8417075	TV-14	111 min
8417099	TV-14	111 min
8417123	TV-14	111 min

201936 rows × 2 columns

```
netflix_df["date_added"] = pd.to_datetime(df["date_added"])
mean_date = netflix_df['date_added'].astype(int).mean().astype('datetime64[ns]')
netflix_df['date_added'].fillna(mean_date, inplace=True)
```

```
<ipython-input-108-388c2d970cf1>:2: FutureWarning: casting datetime64[ns] values to i
  mean_date = netflix_df['date_added'].astype(int).mean().astype('datetime64[ns]')
```

```
netflix_df['rating'].fillna('',inplace=True)
```

```
netflix_df['duration'].fillna('',inplace=True)
```

```
netflix_df.isnull().sum()
```

```
    title           0
    genre           0
    type            0
    director        0
    date_added      0
    release_year    0
    rating          0
    duration        0
    cast            0
    country         0
    description     0
    dtype: int64
```

Problem Statement and Analysing basic metrics

==> Identify business insights and make recommendations to improve Netflix's business by understanding Netflix content over the years.

==> Basic metrics here can be like finding countries with more content, identifying if a country puts emphasis on TV shows or movies etc.

```
# checking first 5 rows
netflix_df.head(5)
```

	title	genre	type	director	date_added	release_year	rating	durati
0	Dick Johnson Is Dead	Documentaries	Movie	Kirsten Johnson	2021-09-25	2020	PG-13	90 r
1	Blood & Water	International TV Shows	TV Show	nan	2021-09-24	2021	TV-MA	Seaso
58	Blood & Water	TV Dramas	TV Show	nan	2021-09-15	2021	TV-MA	Seaso

```
type(netflix_df)
```

```
    pandas.core.frame.DataFrame
```

```
netflix_df.dtypes
```

```
    title                   object
    genre                   object
    type                    object
    director                object
    date_added      datetime64[ns]
    release_year             int64
    rating                  object
    duration                object
    cast                    object
    country                 object
    description             object
    dtype: object
```

```
netflix_df.shape
```

```
    (201936, 11)
```

```
netflix_df.columns
```

```
    Index(['title', 'genre', 'type', 'director', 'date_added', 'release_year',
           'rating', 'duration', 'cast', 'country', 'description'],
          dtype='object')
```

```
netflix_df.info()
```

```
    <class 'pandas.core.frame.DataFrame'>
    Int64Index: 201936 entries, 0 to 8417123
    Data columns (total 11 columns):
     #   Column        Non-Null Count   Dtype
    ---  ------        --------------   -----
     0   title         201936 non-null  object
     1   genre         201936 non-null  object
     2   type          201936 non-null  object
     3   director      201936 non-null  object
     4   date_added    201936 non-null  datetime64[ns]
     5   release_year  201936 non-null  int64
     6   rating        201936 non-null  object
     7   duration      201936 non-null  object
     8   cast          201936 non-null  object
     9   country       201936 non-null  object
     10  description   201936 non-null  object
    dtypes: datetime64[ns](1), int64(1), object(9)
    memory usage: 18.5+ MB
```

```
netflix_df.describe()
```

	release_year
count	201936.000000
mean	2013.452277
std	9.003761
min	1925.000000

```
netflix_df.describe(include="all")
```

```
<ipython-input-113-51f2f71230af>:1: FutureWarning: Treating datetime data as categori
  netflix_df.describe(include="all")
```

	title	genre	type	director	date_added	release_year	rating	d
count	201936	201936	201936	201936	201936	201936.000000	201936	
unique	8807	42	2	4994	174	NaN	18	
top	Kahlil Gibran's The Prophet	Dramas	Movie	nan	1678-01-17 15:00:19.229707264	NaN	TV-MA	
freq	700	29756	145788	50643	201744	NaN	73819	
first	NaN	NaN	NaN	NaN	1678-01-17 15:00:19.229707264	NaN	NaN	
last	NaN	NaN	NaN	NaN	2021-09-25 00:00:00	NaN	NaN	
mean	NaN	NaN	NaN	NaN	NaN	2013.452277	NaN	
std	NaN	NaN	NaN	NaN	NaN	9.003761	NaN	
min	NaN	NaN	NaN	NaN	NaN	1925.000000	NaN	
25%	NaN	NaN	NaN	NaN	NaN	2012.000000	NaN	
50%	NaN	NaN	NaN	NaN	NaN	2016.000000	NaN	
75%	NaN	NaN	NaN	NaN	NaN	2019.000000	NaN	
max	NaN	NaN	NaN	NaN	NaN	2021.000000	NaN	

```
netflix_df["director"].value_counts()
```

```
nan                       50643
Martin Scorsese             419
Youssef Chahine             409
Cathy Garcia-Molina         356
Steven Spielberg            355
                          ...  
Richard Maurice               1
Richard E. Norman             1
Spencer Williams              1
Oscar Micheaux                1
```

```
Kirsten Johnson            1
Name: director, Length: 4994, dtype: int64
```

```
netflix_df.sort_values(by="release_year").head()
```

	title	genre	type	director	date_added	release_year
3992289	Pioneers: First Women Filmmakers*	TV Shows	TV Show	nan	1678-01-17 15:00:19.229707264	1925
7771206	The Battle of Midway	Documentaries	Movie	John Ford	1678-01-17 15:00:19.229707264	1942
7771202	The Battle of Midway	Classic Movies	Movie	John Ford	1678-01-17 15:00:19.229707264	1942

```
netflix_df.isnull().sum()
```

```
title           0
genre           0
type            0
director        0
date_added      0
release_year    0
rating          0
duration        0
cast            0
country         0
description     0
dtype: int64
```

Observations on dataframe

shape of data

```
netflix_df.shape # Processed dataset has 201936 rows and 11 columns
```

```
(201936, 11)
```

data types of all the attributes

```
netflix_df.info() # Apart from release year, all other columns are of type 'object'
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 201936 entries, 0 to 8417123
Data columns (total 11 columns):
 #   Column        Non-Null Count   Dtype         
---  ------        --------------   -----         
 0   title         201936 non-null  object        
 1   genre         201936 non-null  object        
 2   type          201936 non-null  object        
 3   director      201936 non-null  object        
 4   date_added    201936 non-null  datetime64[ns]
 5   release_year  201936 non-null  int64         
 6   rating        201936 non-null  object        
 7   duration      201936 non-null  object        
 8   cast          201936 non-null  object        
 9   country       201936 non-null  object        
 10  description   201936 non-null  object        
dtypes: datetime64[ns](1), int64(1), object(9)
memory usage: 18.5+ MB
```

missing value detection

```
netflix_df.isna().sum(axis = 0) # No null values as we have replaced them earlier.
```

```
title           0
genre           0
type            0
director        0
date_added      0
release_year    0
rating          0
duration        0
cast            0
country         0
description     0
dtype: int64
```

statistical summary

```
netflix_df.describe() # Retuns data for columns which are not type of bool/object. Let us
```

```
release_year
```

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

	show_id	type	title	director	cast	country	date_added	release_
count	8807	8807	8807	6173	7982	7976	8797	8807.00
unique	8807	2	8807	4528	7692	748	1767	
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	
freq	1	6131	1	19	19	2818	109	
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2014.18
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	8.81
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1925.00
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2013.00
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2017.00
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2019.00
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2021.00

categorical attributes - In the given data set, there are some categorical values. Let us find some insights on them.

```
lst = ["type", "release_year", "director", "cast", "country"]
def new_unique(i):
  return pd.value_counts(netflix_df[i]).head() #netflix_data[i].unique()

for i in lst:
  print("Column Name: "+i)
  print("-------------------------------------")
  print(new_unique(i))
  print()
```

```
Column Name: type
-------------------------------------
Movie      145788
TV Show     56148
Name: type, dtype: int64

Column Name: release_year
-------------------------------------
2018    24413
2019    21883
2017    20516
2020    19679
```

```
2016    18465
Name: release_year, dtype: int64

Column Name: director
-------------------------------------
nan                    50643
Martin Scorsese          419
Youssef Chahine          409
Cathy Garcia-Molina      356
Steven Spielberg         355
Name: director, dtype: int64

Column Name: cast
-------------------------------------
nan               2146
Liam Neeson        161
Alfred Molina      160
John Krasinski     139
Salma Hayek        130
Name: cast, dtype: int64

Column Name: country
-------------------------------------
United States     59324
India             22814
United Kingdom    12945
nan               11897
Japan              8679
Name: country, dtype: int64
```

3. Non-Graphical Analysis

```
netflix_df["release_year"].nunique()
```

```
74
```

```
pd.value_counts(netflix_df["release_year"])
```

```
2018    24413
2019    21883
2017    20516
2020    19679
2016    18465
        ...
1947        8
1946        6
1942        6
1943        5
1925        1
Name: release_year, Length: 74, dtype: int64
```

```
netflix_df[netflix_df['type']=='TV Show']['duration'].unique()
```

```
     array(['2 Seasons', '1 Season', '9 Seasons', '4 Seasons', '5 Seasons',
            '3 Seasons', '6 Seasons', '7 Seasons', '10 Seasons', '8 Seasons',
            '17 Seasons', '13 Seasons', '15 Seasons', '12 Seasons',
            '11 Seasons'], dtype=object)
```

```
df["director"].value_counts()
```

```
     Rajiv Chilaka                    19
     Raúl Campos, Jan Suter           18
     Marcus Raboy                     16
     Suhas Kadav                      16
     Jay Karas                        14
                                      ..
     Raymie Muzquiz, Stu Livingston    1
     Joe Menendez                      1
     Eric Bross                        1
     Will Eisenberg                    1
     Mozez Singh                       1
     Name: director, Length: 4528, dtype: int64
```

4. Visual Analysis

```
genre_count = netflix_df[['title','country','type']].drop_duplicates().groupby(by=['countr
```

```
genre_count.sort_values('title',inplace=True,ascending=False)
```

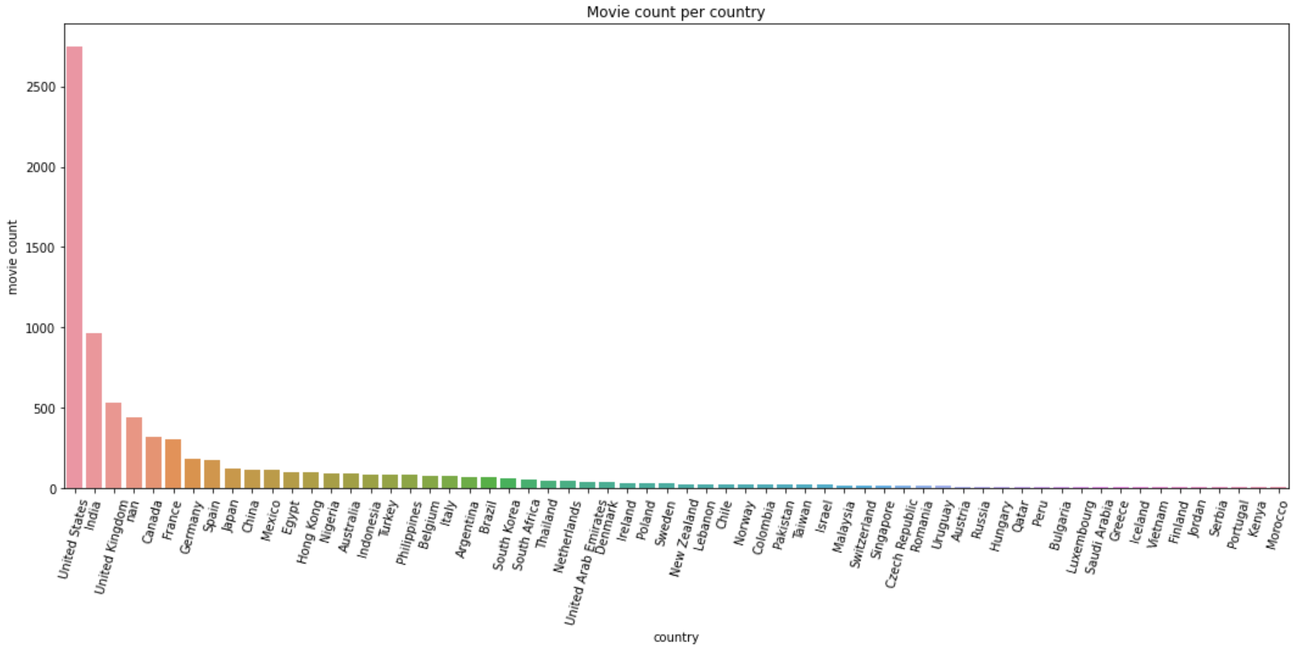
```
genre_count
```

	country	type	title
177	United States	Movie	2751
68	India	Movie	962
178	United States	TV Show	938
174	United Kingdom	Movie	532
188	nan	Movie	440
...	...	...	...
65	Hungary	TV Show	1
155	Sri Lanka	Movie	1
156	Sudan	Movie	1
123	Panama	Movie	1
0		Movie	1

190 rows × 3 columns

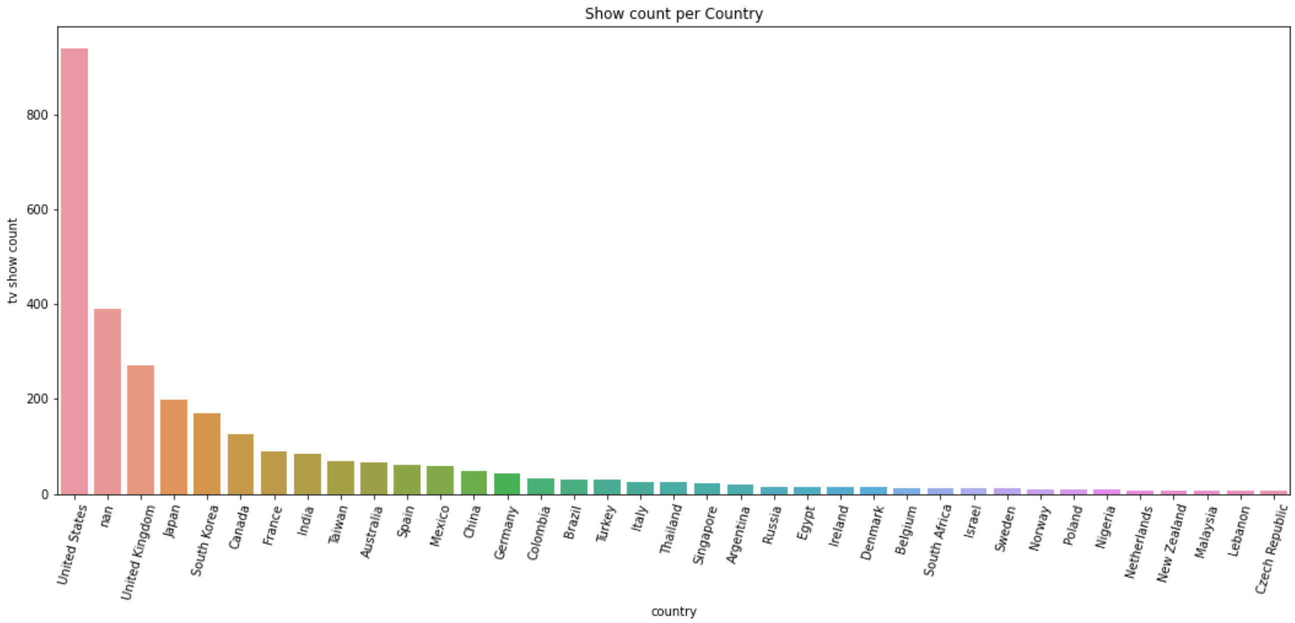
Movies count unique to country > 5

```
plt.figure(figsize=(18,7))
plt.title('Movie count per country')
sns.barplot(x='country',y='title',data=genre_count[(genre_count['type']=='Movie')&(genre_c
plt.xticks(rotation=75)
plt.ylabel('movie count')
plt.show()
```



TV shows based on country > 5

```
plt.figure(figsize=(18,7))
plt.title('Show count per Country')
sns.barplot(x='country',y='title',data=genre_count[(genre_count['type']=='TV Show')&(genre
plt.ylabel('tv show count')
plt.xticks(rotation=75)
plt.show()
```



```
release_yr_cnt = netflix_df[['title','release_year','type']].drop_duplicates().groupby(by=
```

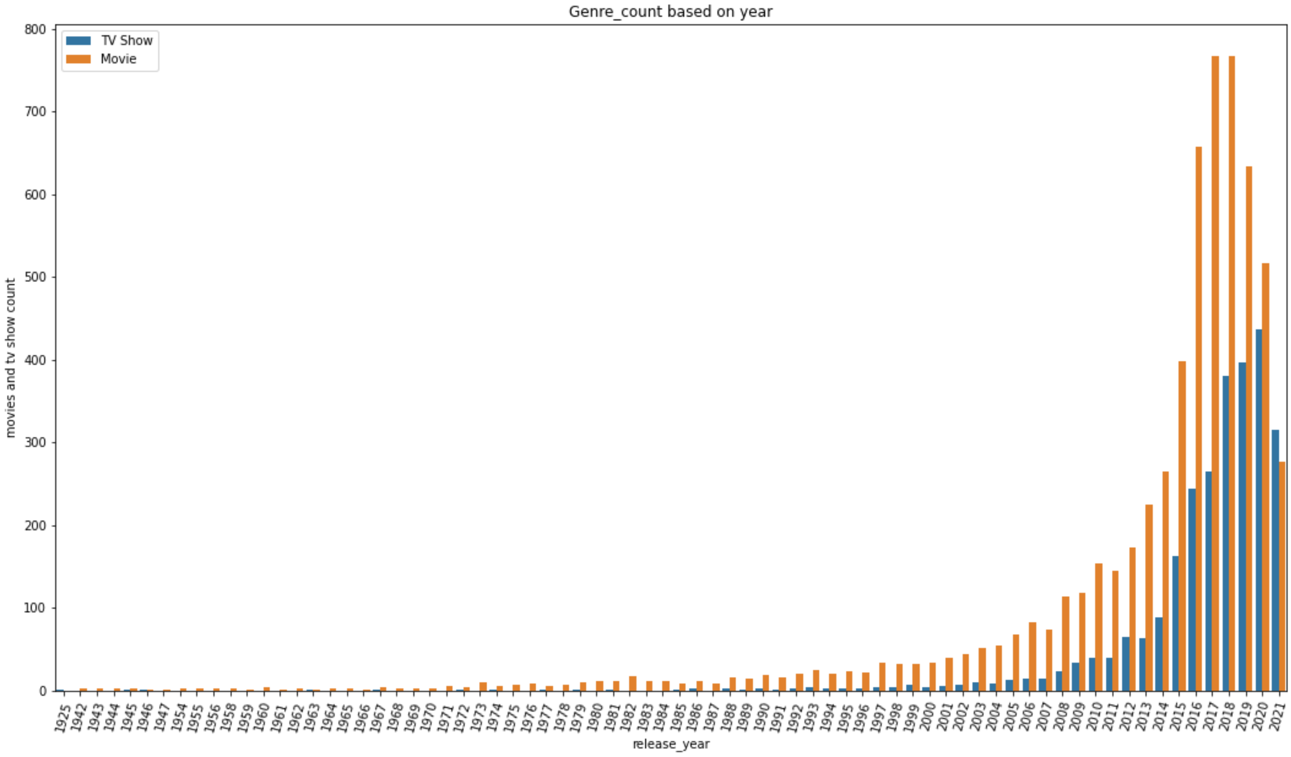
```
release_yr_cnt
```

	release_year	type	title
0	1925	TV Show	1
1	1942	Movie	2
2	1943	Movie	3
3	1944	Movie	3
4	1945	Movie	3
...	...	...	...
114	2019	TV Show	397
115	2020	Movie	517
116	2020	TV Show	436
117	2021	Movie	277
118	2021	TV Show	315

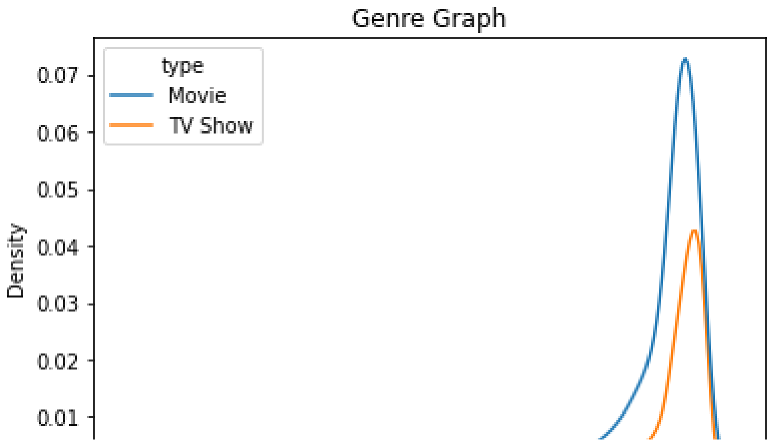
119 rows × 3 columns

Genre_count based on year

```
plt.figure(figsize=(18,10))
plt.title('Genre_count based on year')
sns.barplot(x='release_year',y='title',data=release_yr_cnt,hue='type')
plt.ylabel('movies and tv show count')
plt.xticks(rotation=75)
plt.legend(loc='upper left')
plt.show()
```



```
plt.title('Genre Graph')
sns.kdeplot(data=df,x='release_year',hue='type')
plt.show()
```



```
rating_count = netflix_df[['title','rating','country','type']].drop_duplicates().groupby(b
```

```
rating_count
```

	country	rating	type	title
0		TV-14	Movie	1
1		TV-MA	TV Show	1
2	Afghanistan	TV-MA	Movie	1
3	Albania	TV-MA	Movie	1
4	Algeria	TV-14	Movie	1
...	...	...	...	...
691	nan	TV-Y	Movie	47
692	nan	TV-Y	TV Show	33
693	nan	TV-Y7	Movie	56
694	nan	TV-Y7	TV Show	42
695	nan	TV-Y7-FV	Movie	1

696 rows × 4 columns

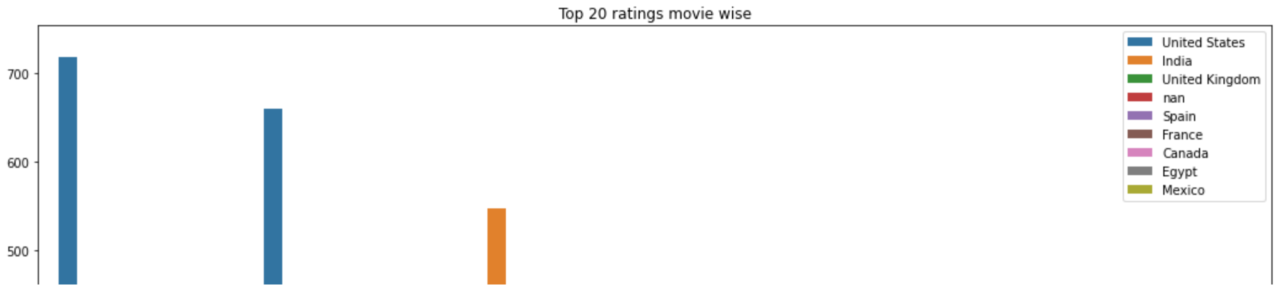
```
rating_count[rating_count.country=='South Korea']
```

	country	rating	type	title
524	South Korea	G	Movie	1
525	South Korea	NR	Movie	4
526	South Korea	PG	Movie	2
527	South Korea	PG-13	Movie	1
528	South Korea	R	Movie	2
529	South Korea	TV-14	Movie	4
530	South Korea	TV-14	TV Show	82
531	South Korea	TV-G	TV Show	1
532	South Korea	TV-MA	Movie	37

```
rating_count.sort_values(by=['title','country'],ascending=[False,True],inplace=True)
```

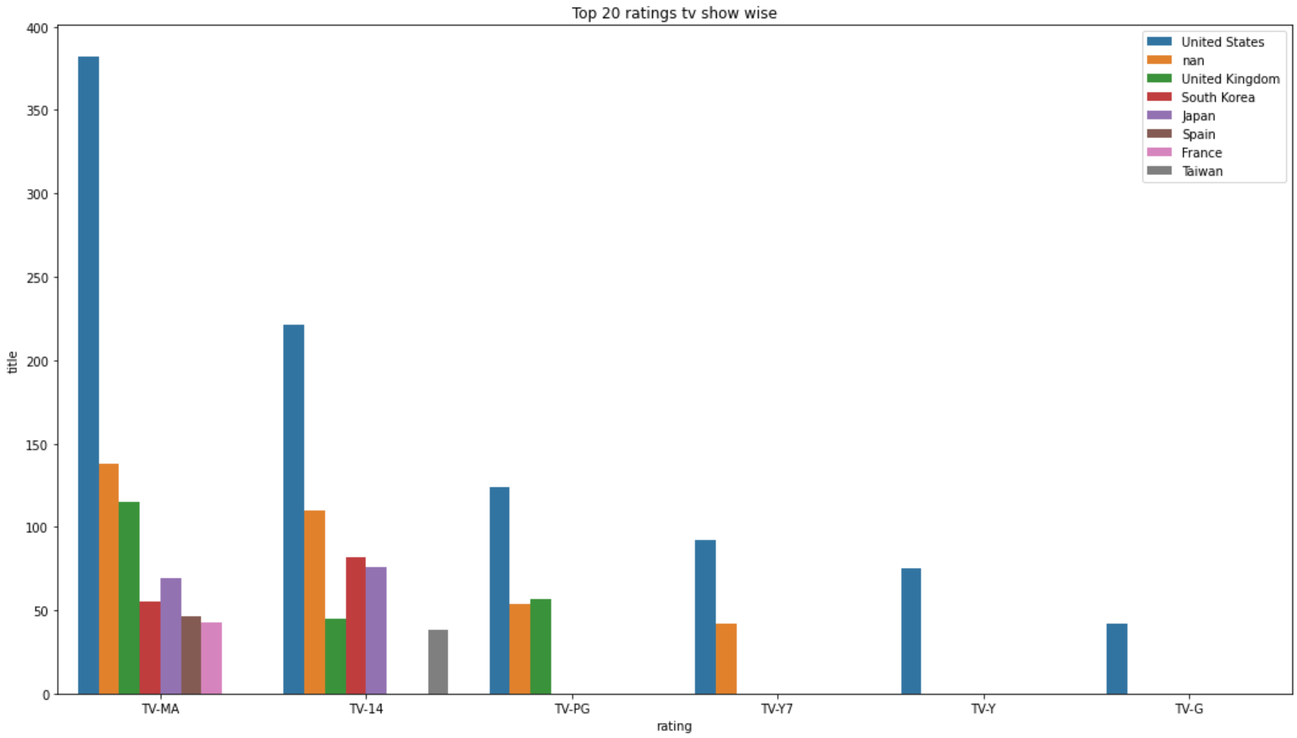
Top20 ratings for movies

```
plt.figure(figsize=(18,10))
plt.title('Top 20 ratings for movies')
sns.barplot(x='rating',y='title',data=rating_count[rating_count['type']=='Movie'].head(20)
plt.legend(loc='upper right')
plt.show()
```

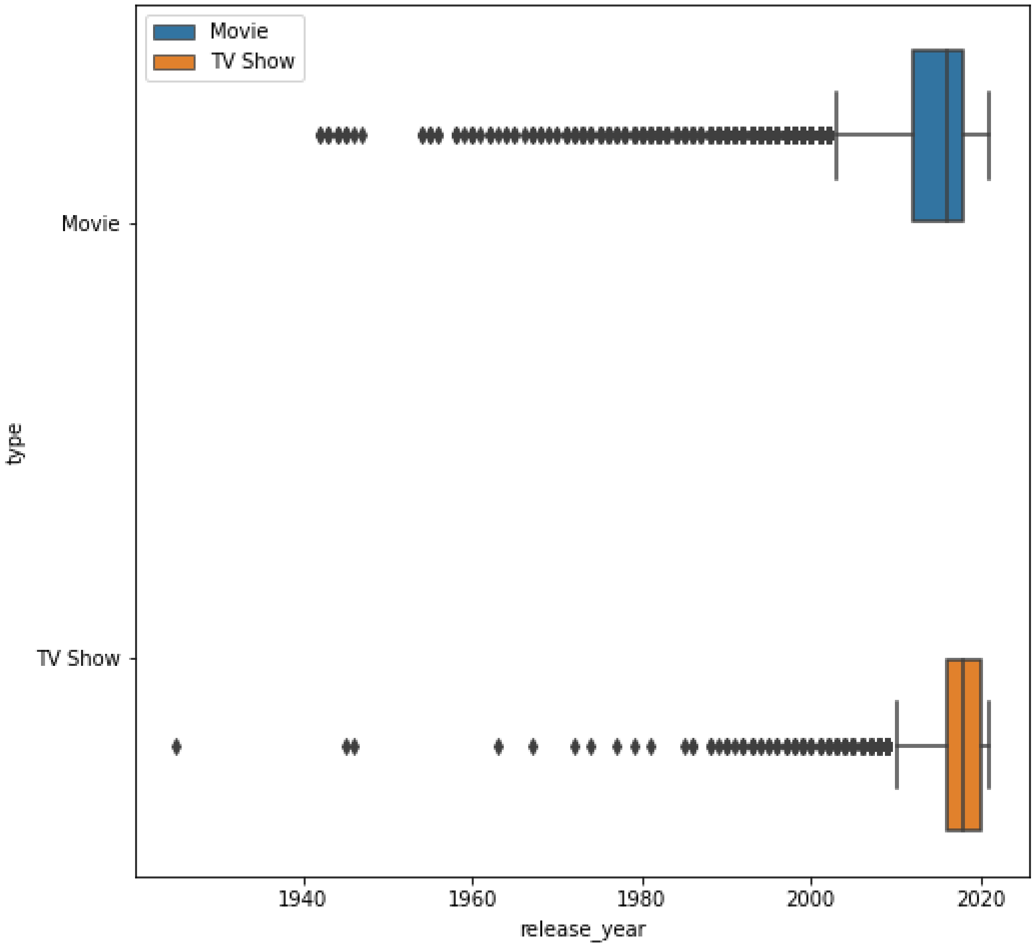


Top 20 TV shows

```
plt.figure(figsize=(18,10))
plt.title('Top 20 ratings for shows')
sns.barplot(x='rating',y='title',data=rating_count[rating_count['type']=='TV Show'].head(2
plt.legend(loc='upper right')
plt.show()
```



```
plt.figure(figsize=(8,8))
sns.boxplot(data=df,x='release_year',y='type',hue='type')
plt.legend(loc='upper left')
plt.show()
```



Insights based on Non-Graphical and Visual Analysis

Non-Graphical Analysis Insights

==> The data given indicates Netflix has more movies when compared with TV shows

==> The data is also incorrect for the duration field because it cannot be null. The field values for 3 rows are swapped with the wrong column and the 3 rows are for the same cast and director which indicates faulty data

==> The cast and director and country are unknown in some cases which can hinder further analysis

==> The date_added field also has null values and it should not be possible because the field indicates when content has been added to Netflix.

Graphical Analysis Insights

==> From the plot with the title 'Movie count per country' we can see that USA is number one followed by India. There are few countries which have less than 5 movies which is not good

==> From the plot with the title 'Show count per country' we can see that USA is number one followed by Unknown[None] where as India is at 8th position. This can indicate 2 things either Indians like to watch more movies or the majority TV shows content available in India is being sold to some other OTT services.

==> From the plot with the title 'Genre_count based on year' we can see that the movie and TV show trend was growing though there is a reduction in 2021. This can be due to external reasons like the absence of Covid in the dataset

==> It is clear from the plot with the title 'Top 20 ratings for movies' that USA telecasts a majority type of movie, whereas R-rated movies aren't telecast as often because there are fewer of them in other nations.

==> From the plot with the title 'Top 20 ratings for shows' we can see that USA telecasts the majority kind and UK and south korea. We can also see South Korea put more emphasis on TV shows when compared to movies.

Business Insights

==> Majority of the netflix content is available for USA and UK and Canada so netflix should make sure they keep adding content in those regions more

==> South korea has more emphasis over TV shows which is good and netflix should keep on supporting them since netflix has less TV shows when compared with movies

Recommendations

==> To improve tv show content over netflix a research need to be done in respective countries where movie content is really good but the tv show content is not good to find out the other vendors who host the tv show content and take action accordingly

==> Netflix should also focus on its reach to other countries from where the content is too low

✓ 1s completed at 12:57 AM ● ×