Amazone TV SHows and Movies EDA

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

In [2]:
```
import sys
if not sys.warnoptions:
    import warnings
    warnings.simplefilter('ignore')
```

In [5]:
```
# load csv files
credit_df=pd.read_csv(r'C:\Users\Dell\Downloads\credits.csv')
title_df=pd.read_csv(r'C:\Users\Dell\Downloads\titles.csv')
```

Explore Credit Data

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

Out[6]:

	person_id	id	name	character	role
0	59401	ts20945	Joe Besser	Joe	ACTOR
1	31460	ts20945	Moe Howard	Moe	ACTOR
2	31461	ts20945	Larry Fine	Larry	ACTOR
3	21174	tm19248	Buster Keaton	Johnny Gray	ACTOR
4	28713	tm19248	Marion Mack	Annabelle Lee	ACTOR

In [7]:
```
#shape of a data
credit_df.shape
```

Out[7]:
```
(124235, 5)
```

In [8]:
```
#information of data
credit_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 124235 entries, 0 to 124234
Data columns (total 5 columns):
 #   Column     Non-Null Count   Dtype 
---  ------     --------------   ----- 
 0   person_id  124235 non-null  int64 
 1   id         124235 non-null  object
 2   name       124235 non-null  object
 3   character  107948 non-null  object
 4   role       124235 non-null  object
dtypes: int64(1), object(4)
memory usage: 4.7+ MB
```

In [11]:
```
#sum the null Value
credit_df.isnull().sum()
```

Out[11]:
```
person_id        0
id               0
name             0
character    16287
role             0
dtype: int64
```

#character have 16287 null value

In [12]:
```
#percentage of missing value
round(100*(credit_df.isnull().sum()/len(credit_df.index)),2)
```

Out[12]:
```
person_id     0.00
id            0.00
name          0.00
character    13.11
role          0.00
dtype: float64
```

character column has 13% missing value

In [13]:
```
# Replace the missing value with 'no data'
credit_df.replace(np.nan,'No Data',inplace=True)
```

In [14]:
```
credit_df.duplicated().sum()
```

Out[14]:
```
56
```

Explore the title detail

In [15]:
```
1 title_df.head()
```

Out[15]:

description	release_year	age_certification	runtime	genres	production_countries	seasons	imd
The Three Stooges were an American vaudeville ...	1934	TV-PG	19	['comedy', 'family', 'animation', 'action', 'f...	['US']	26.0	tt085
During America's Civil War, Union spies steal ...	1926	NaN	78	['action', 'drama', 'war', 'western', 'comedy'...	['US']	NaN	tt001
It's the hope that sustains the spirit of ever...	1946	NaN	171	['romance', 'war', 'drama']	['US']	NaN	tt003
Hildy, the journalist former wife of newspaper...	1940	NaN	92	['comedy', 'drama', 'romance']	['US']	NaN	tt003
An aspiring actress begins to suspect that her...	1950	NaN	94	['thriller', 'drama', 'romance']	['US']	NaN	tt004

In [16]:
```
1 # shape of data
2 title_df.shape
```

Out[16]: (9871, 15)

In [17]:
```
#information of data
title_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9871 entries, 0 to 9870
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    9871 non-null   object 
 1   title                 9871 non-null   object 
 2   type                  9871 non-null   object 
 3   description           9752 non-null   object 
 4   release_year          9871 non-null   int64  
 5   age_certification     3384 non-null   object 
 6   runtime               9871 non-null   int64  
 7   genres                9871 non-null   object 
 8   production_countries  9871 non-null   object 
 9   seasons               1357 non-null   float64
 10  imdb_id               9204 non-null   object 
 11  imdb_score            8850 non-null   float64
 12  imdb_votes            8840 non-null   float64
 13  tmdb_popularity       9324 non-null   float64
 14  tmdb_score            7789 non-null   float64
dtypes: float64(5), int64(2), object(8)
memory usage: 1.1+ MB
```

In [18]:
```
# missing value
title_df.isnull().sum()
```

Out[18]:
```
id                       0
title                    0
type                     0
description            119
release_year             0
age_certification     6487
runtime                  0
genres                   0
production_countries     0
seasons               8514
imdb_id                667
imdb_score            1021
imdb_votes            1031
tmdb_popularity        547
tmdb_score            2082
dtype: int64
```

In [21]:
```
#percentage of missing value in a columns
round(100*(title_df.isnull().sum()/len(title_df.index)),2)
```

Out[21]:
```
id                       0.00
title                    0.00
type                     0.00
description              1.21
release_year             0.00
age_certification       65.72
runtime                  0.00
genres                   0.00
production_countries     0.00
seasons                 86.25
imdb_id                  6.76
imdb_score              10.34
imdb_votes              10.44
tmdb_popularity          5.54
tmdb_score              21.09
dtype: float64
```

age_certification and seasons more than 60 % missing value.drop those columns

In [22]:
```
title_df=title_df.drop(columns=['seasons','age_certification'])
```

In [26]:
```
#INPUT THE VALUE REST OF NULL COLUMNS
title_df['imdb_id'].replace(np.nan,'No Data',inplace=True)
title_df['description'].replace(np.nan,'No Data',inplace=True)
title_df['imdb_score']=title_df['imdb_score'].fillna(title_df['imdb_score'].
title_df['imdb_votes']=title_df['imdb_votes'].fillna(title_df['imdb_votes'].
title_df['tmtb_popularity']=title_df['tmdb_popularity'].fillna(title_df['tmd
title_df['tmdb_score']=title_df['tmdb_score'].fillna(title_df['tmdb_score'].
```

In [27]:
```
title_df['production_countries'] = title_df['production_countries'].str[2:4]
for i in range(len(title_df['production_countries'])):
    if title_df['production_countries'][i] == '':
        title_df['production_countries'][i] = 'Unknown'
```

In [28]:
```
credit_df.head()
```

Out[28]:

	person_id	id	name	character	role
0	59401	ts20945	Joe Besser	Joe	ACTOR
1	31460	ts20945	Moe Howard	Moe	ACTOR
2	31461	ts20945	Larry Fine	Larry	ACTOR
3	21174	tm19248	Buster Keaton	Johnny Gray	ACTOR
4	28713	tm19248	Marion Mack	Annabelle Lee	ACTOR

In [29]:
```
1 title_df.head()
```

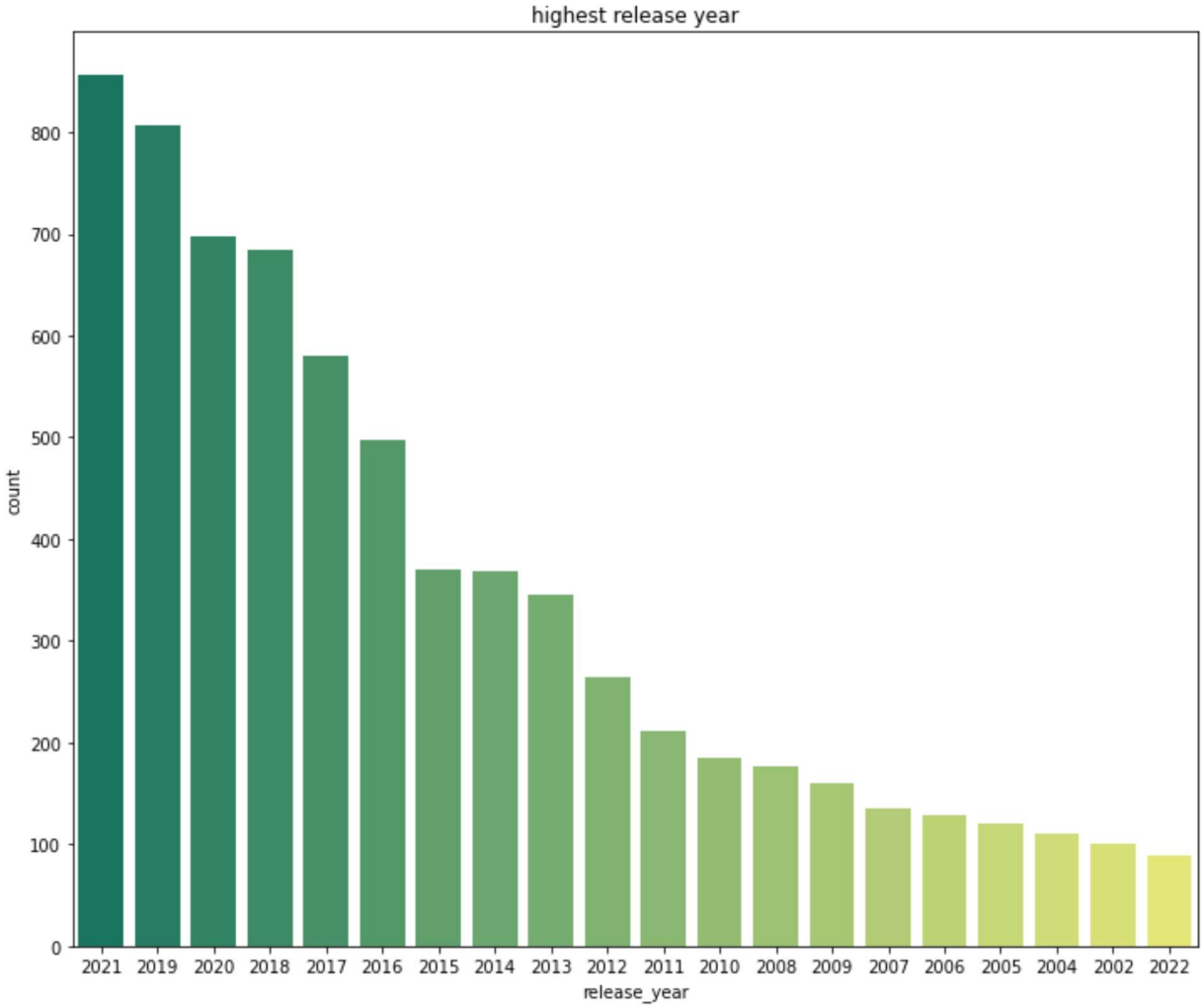
Out[29]:

	type	description	release_year	runtime	genres	production_countries	imdb_id	imdb_score
	SHOW	The Three Stooges were an American vaudeville ...	1934	19	['comedy', 'family', 'animation', 'action', 'f...	US	tt0850645	8.6
	MOVIE	During America's Civil War, Union spies steal ...	1926	78	['action', 'drama', 'war', 'western', 'comedy'...	US	tt0017925	8.2
	MOVIE	It's the hope that sustains the spirit of ever...	1946	171	['romance', 'war', 'drama']	US	tt0036868	8.1
	MOVIE	Hildy, the journalist former wife of newspaper...	1940	92	['comedy', 'drama', 'romance']	US	tt0032599	7.8
	MOVIE	An aspiring actress begins to suspect that her...	1950	94	['thriller', 'drama', 'romance']	US	tt0042593	7.9

EDA

In [30]:
```
#top 20 years maximum movies and tv shows were added
plt.figure(figsize=(12,10))
plt.title('highest release year')
sns.countplot(x='release_year',data=title_df,order=title_df['release_year'].
```

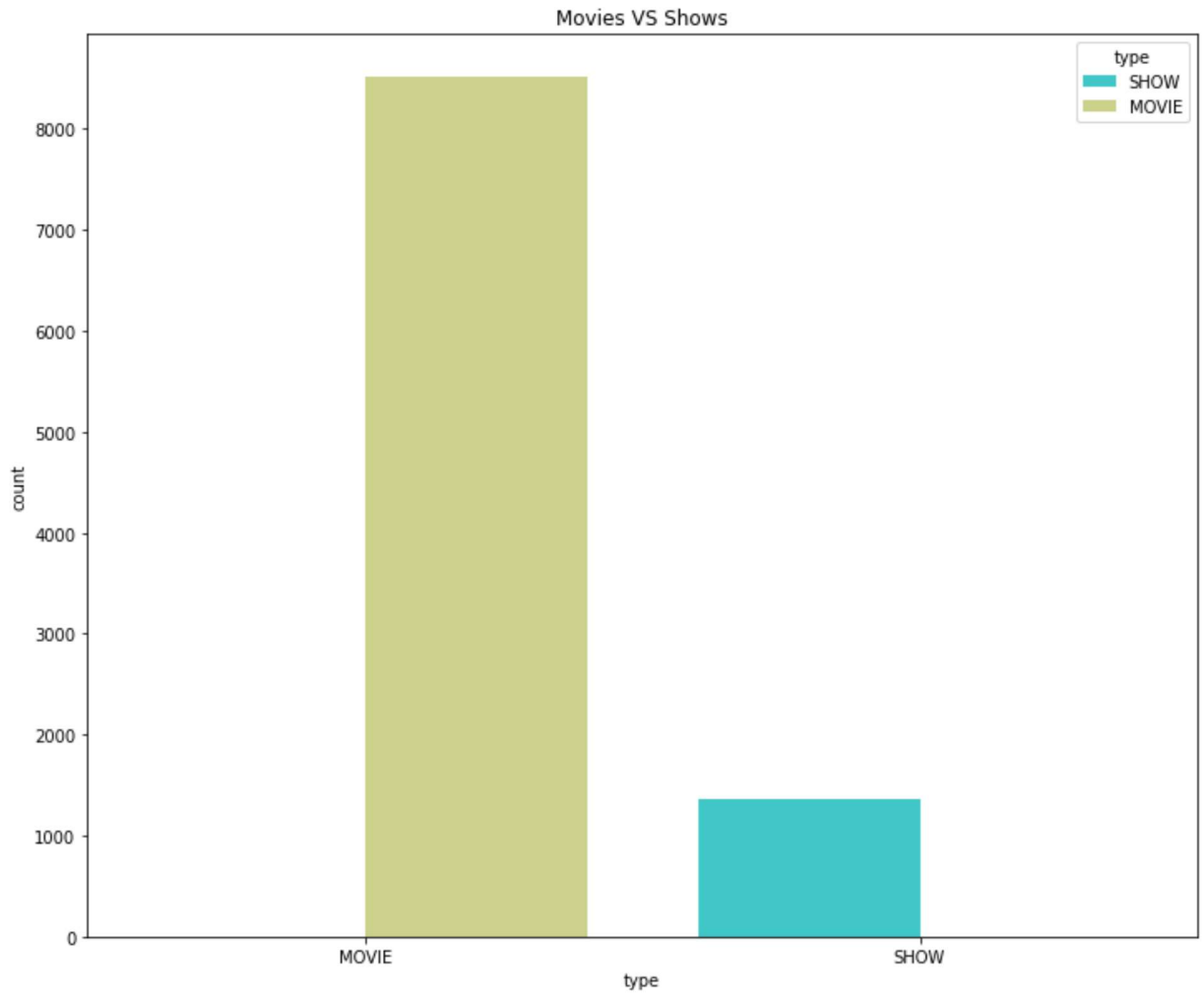
Out[30]:
```
<AxesSubplot:title={'center':'highest release year'}, xlabel='release_year', yl
abel='count'>
```



shows that 2021 maximum number of realeses movies and shows

In [31]:
```
# Top 20 Years Maximum movies or shows added
plt.figure(figsize=(12,10))
plt.title('Movies VS Shows')
sns.countplot(x='type',data=title_df,order=title_df['type'].value_counts().i
```

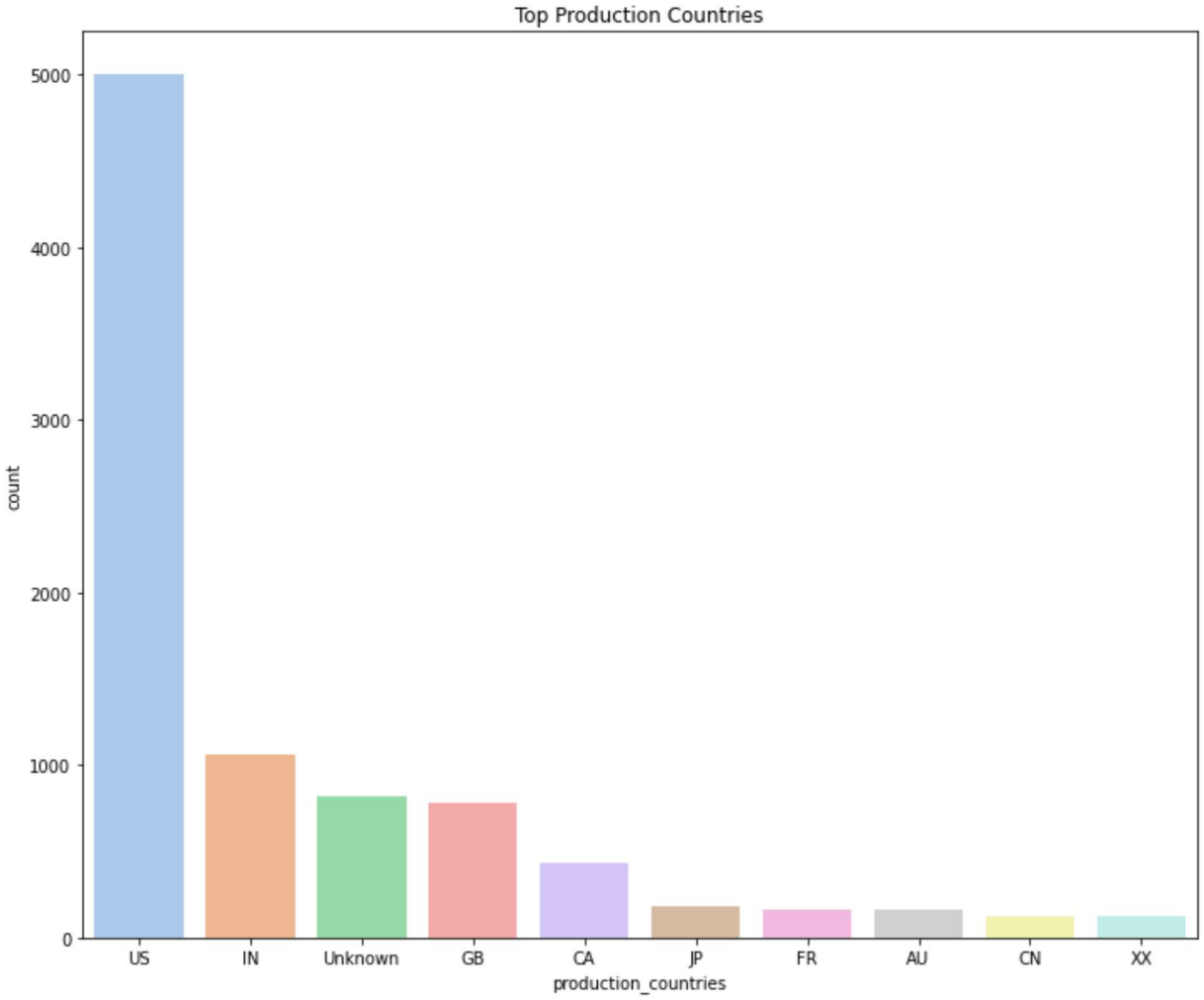
Out[31]:
```
<AxesSubplot:title={'center':'Movies VS Shows'}, xlabel='type', ylabel='count'>
```



it shows that maximum no of moview=s release as compair to shows

In [33]:
```
# Top 10 production countries
plt.figure(figsize=(12,10))
plt.title('Top Production Countries')
sns.countplot(x='production_countries',data=title_df,order=title_df['product
        palette='pastel')
```

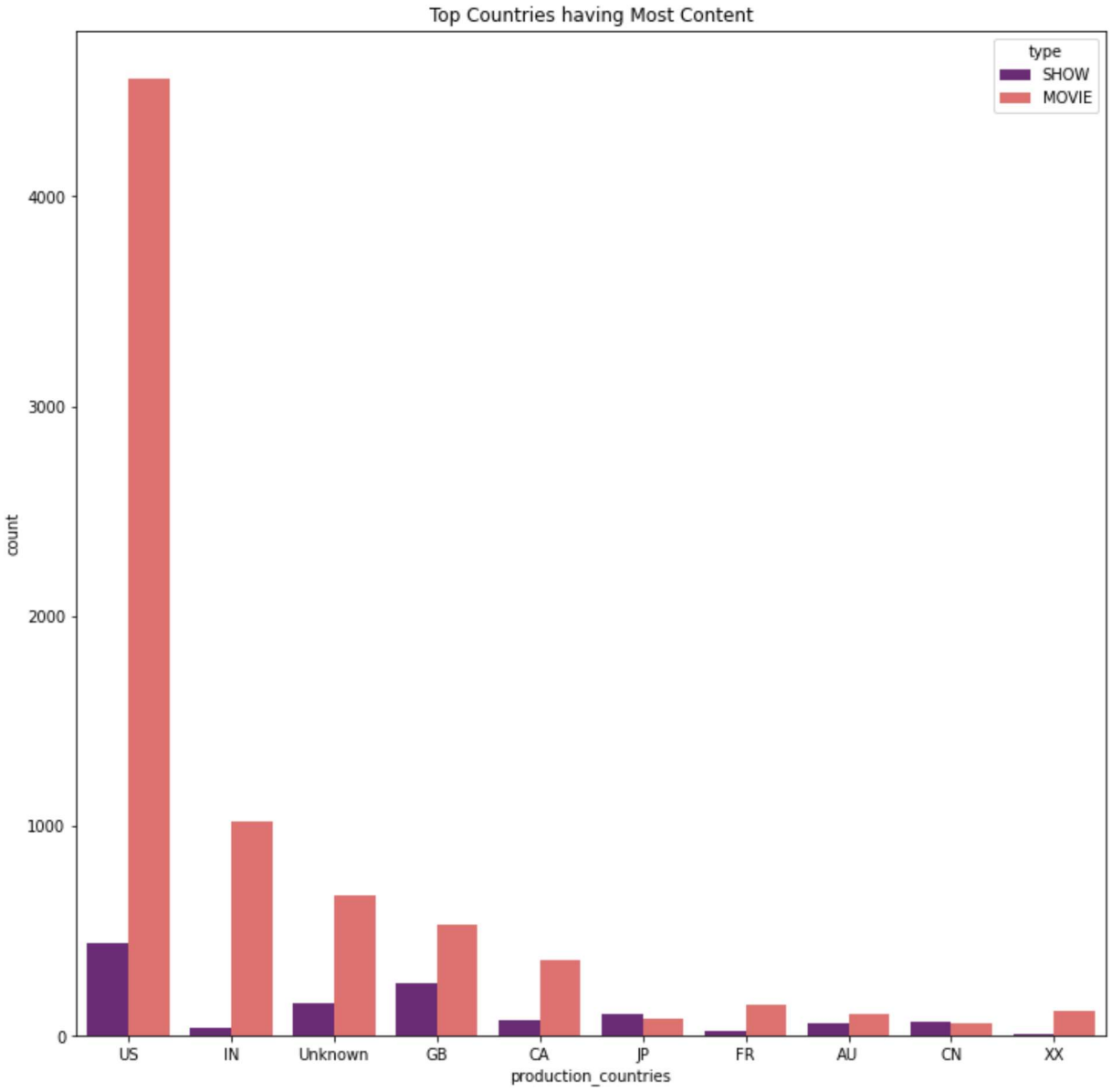
Out[33]:
```
<AxesSubplot:title={'center':'Top Production Countries'}, xlabel='production_co
untries', ylabel='count'>
```



it shows that US maximum produce movies and shows

In [34]:
```
#top countries having most content
plt.figure(figsize=(12,12))
plt.title('Top Countries having Most Content')
sns.countplot(x='production_countries',data=title_df,order=title_df['product
```

Out[34]:
```
<AxesSubplot:title={'center':'Top Countries having Most Content'}, xlabel='prod
uction_countries', ylabel='count'>
```



Popularity Based

In [35]:
```
new_df=title_df[['title','type','genres','production_countries','release_yea
```

In [36]:
```
new_df=new_df[new_df['imdb_votes']>500]
```

In [41]:
```
new_df=new_df.sort_values(by=['imdb_votes','imdb_score'],ascending=False)
```

In [42]:
```
new_df.head()
```

Out[42]:

	title	type	genres	production_countries	release_year	imdb_score	imdb_votes
2220	Titanic	MOVIE	['drama', 'romance']	US	1997	7.9	1133692.0
2230	The Usual Suspects	MOVIE	['thriller', 'crime', 'drama']	US	1995	8.5	1059480.0
2237	Braveheart	MOVIE	['drama', 'history', 'war']	US	1995	8.4	1016629.0
2229	The Sixth Sense	MOVIE	['thriller', 'drama']	US	1999	8.2	967864.0
1814	The Terminator	MOVIE	['thriller', 'action', 'scifi']	US	1984	8.1	841706.0

In [45]:
```
popular_types=new_df[new_df['release_year']==2022]
popular_Movies=popular_types[popular_types['type']=='MOVIE']
popular_Movies
```

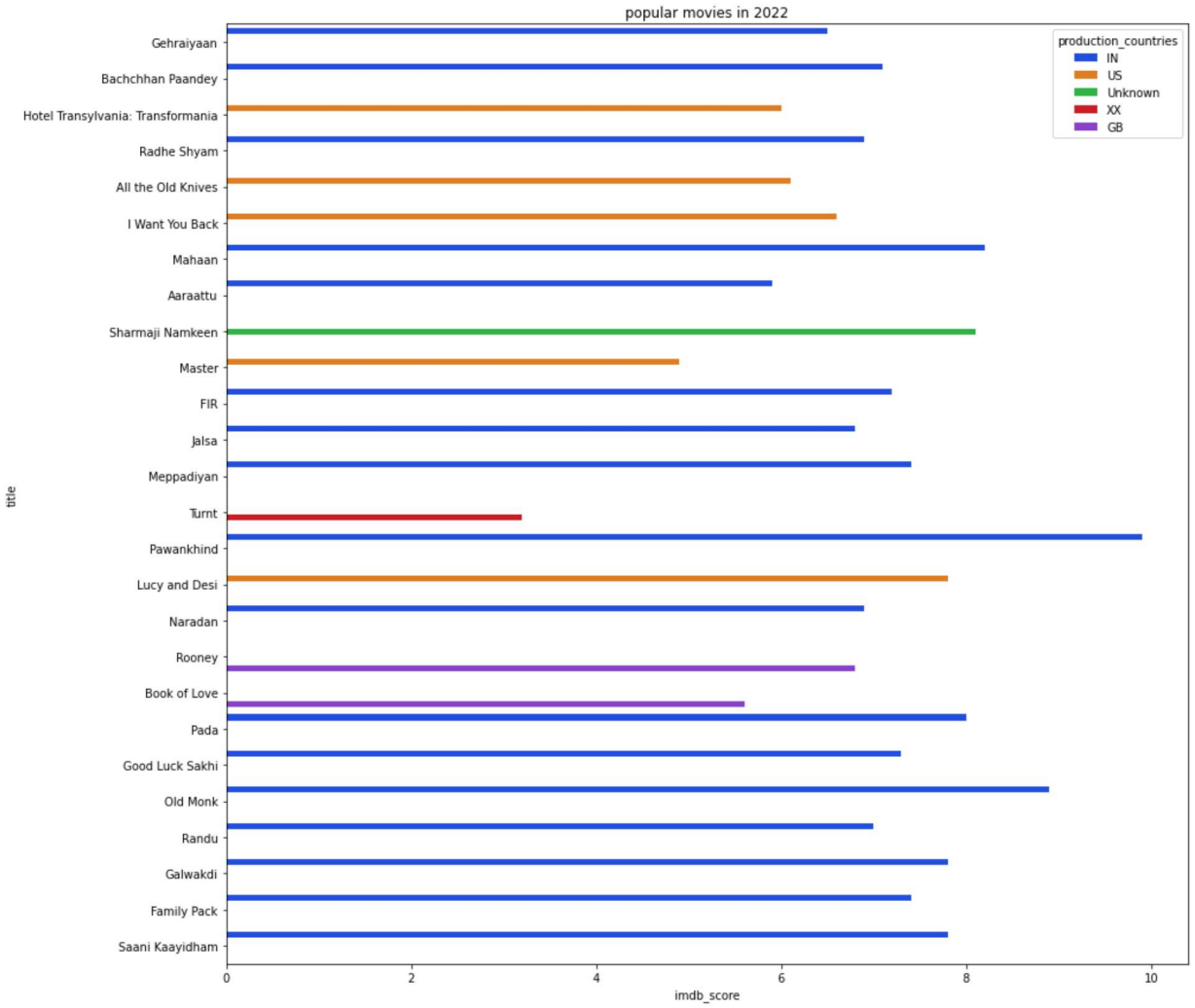
Out[45]:

	title	type	genres	production_countries	release_year	imdb_score	imdb_
9167	Gehraiyaan	MOVIE	['romance', 'drama', 'comedy']	IN	2022	6.5	4
8988	Bachchhan Paandey	MOVIE	['action', 'crime', 'drama', 'comedy']	IN	2022	7.1	3
8964	Hotel Transylvania: Transformania	MOVIE	['fantasy', 'romance', 'animation', 'comedy', ...	US	2022	6.0	2(
9009	Radhe Shyam	MOVIE	['romance', 'drama']	IN	2022	6.9	2
8934	All the Old Knives	MOVIE	['thriller']	US	2022	6.1	1
8952	I Want You Back	MOVIE	['romance', 'family', 'comedy']	US	2022	6.6	1
9276	Mahaan	MOVIE	['action', 'drama', 'thriller', 'crime']	IN	2022	8.2	1
9330	Aaraattu	MOVIE	['action', 'drama']	IN	2022	5.9	
9097	Sharmaji Namkeen	MOVIE	['comedy', 'drama', 'family']	Unknown	2022	8.1	
8960	Master	MOVIE	['thriller', 'horror', 'drama']	US	2022	4.9	
9241	FIR	MOVIE	['thriller', 'action']	IN	2022	7.2	
9062	Jalsa	MOVIE	['thriller', 'drama']	IN	2022	6.8	
9212	Meppadiyan	MOVIE	['drama', 'thriller']	IN	2022	7.4	
9338	Turnt	MOVIE	['drama']	XX	2022	3.2	
9135	Pawankhind	MOVIE	['action', 'drama', 'history']	IN	2022	9.9	
8967	Lucy and Desi	MOVIE	['comedy', 'drama', 'history', 'documentation']	US	2022	7.8	
9098	Naradan	MOVIE	['drama', 'thriller']	IN	2022	6.9	
9127	Rooney	MOVIE	['documentation', 'sport']	GB	2022	6.8	
9001	Book of Love	MOVIE	['romance', 'comedy']	GB	2022	5.6	
9121	Pada	MOVIE	['thriller', 'crime', 'drama']	IN	2022	8.0	
9522	Good Luck Sakhi	MOVIE	['drama', 'romance', 'comedy']	IN	2022	7.3	

	title	type	genres	production_countries	release_year	imdb_score	imdb_
9311	Old Monk	MOVIE	['comedy', 'romance']	IN	2022	8.9	
9316	Randu	MOVIE	['comedy']	IN	2022	7.0	
9146	Galwakdi	MOVIE	['drama', 'comedy', 'romance']	IN	2022	7.8	
9519	Family Pack	MOVIE	['comedy', 'romance', 'drama']	IN	2022	7.4	
9058	Saani Kaayidham	MOVIE	['drama', 'action', 'crime']	IN	2022	7.8	

In [47]:
```
plt.figure(figsize=(15,15))
plt.title('popular movies in 2022')
sns.barplot(y='title',data=popular_Movies,x='imdb_score',hue='production_cou
```

Out[47]:
```
<AxesSubplot:title={'center':'popular movies in 2022'}, xlabel='imdb_score', yl
abel='title'>
```



In [48]:
```
#Popular Show in 2022
popular_shows=popular_types[popular_types['type']=='SHOW']
popular_shows
```

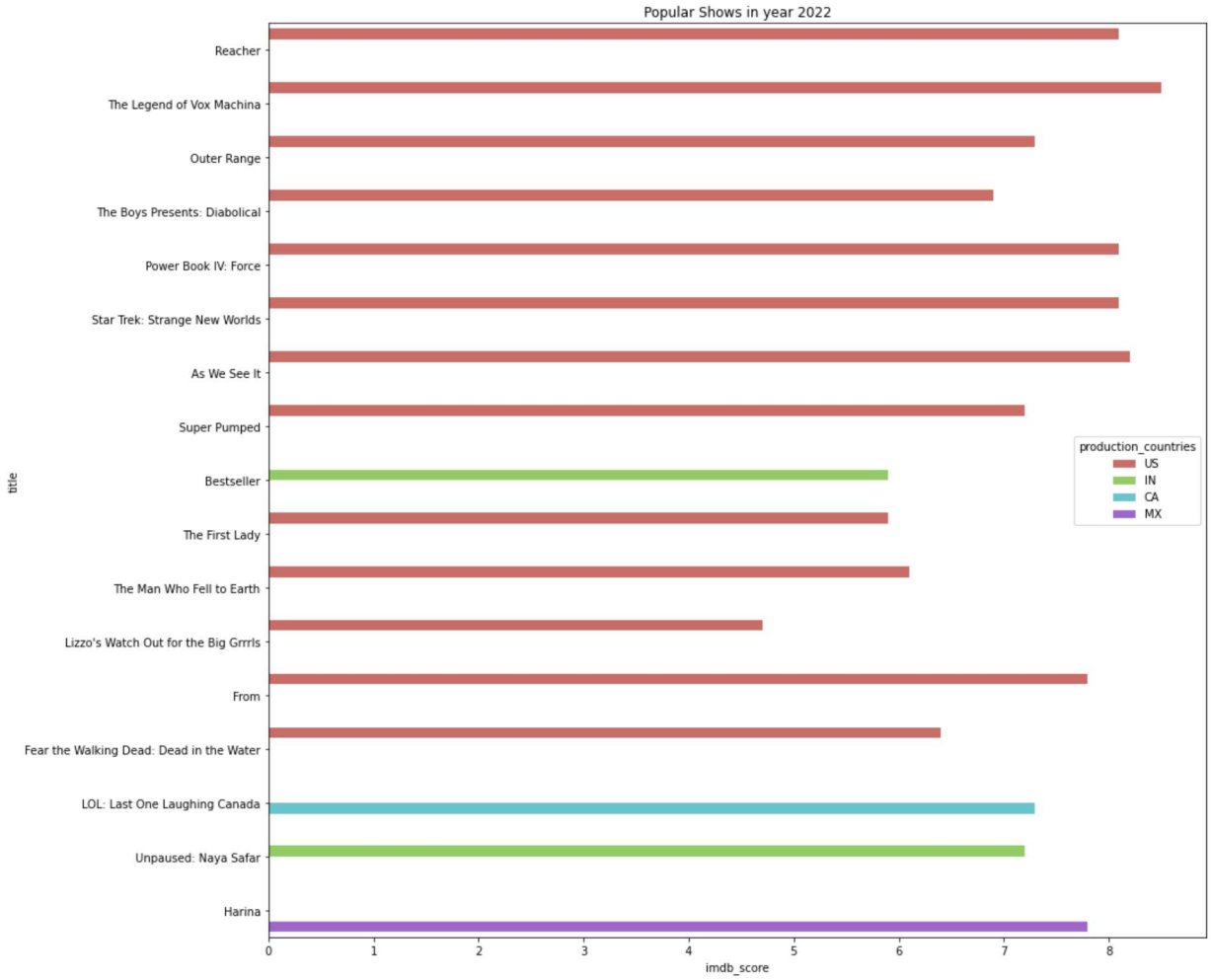
Out[48]:

	title	type	genres	production_countries	release_year	imdb_score	imdb_votes
8939	Reacher	SHOW	['action', 'crime', 'drama', 'thriller']	US	2022	8.1	95704.0
8955	The Legend of Vox Machina	SHOW	['animation', 'action', 'comedy', 'fantasy', '...	US	2022	8.5	18406.0
8926	Outer Range	SHOW	['western', 'drama', 'thriller']	US	2022	7.3	11642.0
8970	The Boys Presents: Diabolical	SHOW	['drama', 'scifi', 'animation', 'action', 'com...	US	2022	6.9	7846.0
8974	Power Book IV: Force	SHOW	['crime', 'drama']	US	2022	8.1	4066.0
8930	Star Trek: Strange New Worlds	SHOW	['scifi', 'action']	US	2022	8.1	3625.0
8969	As We See It	SHOW	['comedy', 'drama']	US	2022	8.2	2818.0
8935	Super Pumped	SHOW	['drama']	US	2022	7.2	2756.0
9253	Bestseller	SHOW	['drama', 'thriller']	IN	2022	5.9	1630.0
8929	The First Lady	SHOW	['drama', 'history']	US	2022	5.9	1502.0
8933	The Man Who Fell to Earth	SHOW	['scifi', 'drama', 'thriller']	US	2022	6.1	1317.0
8984	Lizzo's Watch Out for the Big Grrrls	SHOW	['reality']	US	2022	4.7	1059.0
8928	From	SHOW	['scifi', 'thriller', 'drama', 'horror']	US	2022	7.8	662.0
9029	Fear the Walking Dead: Dead in the Water	SHOW	['action']	US	2022	6.4	615.0

	title	type	genres	production_countries	release_year	imdb_score	imdb_votes
9015	LOL: Last One Laughing Canada	SHOW	['comedy', 'reality']	CA	2022	7.3	583.0
9192	Unpaused: Naya Safar	SHOW	['drama']	IN	2022	7.2	574.0
9040	Harina	SHOW	['comedy']	MX	2022	7.8	509.0

In [50]:
```
plt.figure(figsize=(15,15))
plt.title('Popular Shows in year 2022')
sns.barplot(y='title',data=popular_shows,x='imdb_score',hue='production_coun
```

Out[50]:
```
<AxesSubplot:title={'center':'Popular Shows in year 2022'}, xlabel='imdb_scor
e', ylabel='title'>
```



In [ ]:
```
1
```