1. Netflix EDA

May 21, 2024

0.1 Table of Contents:

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
from tqdm import tqdm
```

1 Problem Statement, Insights and Recommendations:

- Problem Statement :
    - Analyze the data and generate insights that could help Netflix in deciding **which type of shows/movies to produce** and how they can **grow the business in different countries**
- Insights :

  - Content creation has exploded post 2015, reaching maximum in 2018 (graph here)
  - The fraction of TV shows created has been steadily increasing since 2017, surpassing 50% mark in 2021 (graph here)
  - Content added on netflix has grown exponentially since 2015, reaching maximum in 2019 (graph here)
  - Content has been added uniformly across all months (graph here)
  - Netflix has a strong preference of adding content on either 1st or 15th of a month (graph here)
  - Best Movie Director : Rajiv Chilaka (Animator for Chota Bheem) (graph here)
  - Best Movie Cast : Anupam Kher (Big Bollywood Star) (graph here)
  - Best TV Show Director : Alastair Fothergill (Nature Documentaries like Our Planet) (graph here)
  - Best TV Show Cast : Takahiro Sakurai (Japanese Voice actor eg Jujutsu Kaisen) (graph here)
  - Top 3 movie Genres : Dramas, Comedies, Documentaries (graph here)
  - Top 3 TV Show Genres : Dramas, Comedies, Crime (graph here)
  - US is top producer leading in 35 out of 42 Genres (graph here)
- Recommendations :
  - Netflix should keep on adding content which has been produced in US as it has its strongest holding there
  - In order to expand into more countries, Netflix needs to look beyond its biggest producers ie US. It needs to add content from countries which shows the highest promise for a particular genre, and also needs to hire the best native talent in terms of the director and the cast. Following are some my recommendations based on data : (supporting data and all recommendations)
    - Thrillers : Directed by Anurag Kashyap casting Nawazuddin Siddiqui produced in India
    - Docuseries : Directed by Alastair Fothergill casting David Attenborough produced in UK
    - TV Dramas : Directed by Jeon Go-woon casting Cho Seong-ha produced in South Korea
    - Sports Movies : Directed by Clay Porter casting Usain Bolt produced in UK

2 Non Graphical Eda

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

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

[3]:
```
  show_id     type                  title         director  \
0      s1    Movie   Dick Johnson Is Dead  Kirsten Johnson
1      s2  TV Show          Blood & Water              NaN
2      s3  TV Show              Ganglands  Julien Leclercq
3      s4  TV Show  Jailbirds New Orleans              NaN
4      s5  TV Show           Kota Factory              NaN
```

```
                                                cast        country  \
0                                                NaN  United States
1  Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…   South Africa
2  Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…            NaN
3                                                NaN            NaN
4  Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…          India

           date_added  release_year rating   duration  \
0  September 25, 2021          2020  PG-13     90 min
1  September 24, 2021          2021  TV-MA  2 Seasons
2  September 24, 2021          2021  TV-MA   1 Season
3  September 24, 2021          2021  TV-MA   1 Season
4  September 24, 2021          2021  TV-MA  2 Seasons

                                           listed_in  \
0                                      Documentaries
1    International TV Shows, TV Dramas, TV Mysteries
2  Crime TV Shows, International TV Shows, TV Act…
3                             Docuseries, Reality TV
4  International TV Shows, Romantic TV Shows, TV …

                                         description
0  As her father nears the end of his life, filmm…
1  After crossing paths at a party, a Cape Town t…
2  To protect his family from a powerful drug lor…
3  Feuds, flirtations and toilet talk go down amo…
4  In a city of coaching centers known to train I…
```

[4]:
```
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
```

Missing values! - Missing values present in : director, cast, country, date_added, rating, duration
- Director data has highest nulls : ~2.5k movies/tv shows data is missing

---

[5]:

```
# adding a year column will help with analysis further
df['date_added'] = pd.to_datetime(df['date_added'])
df['date_added_year'] = df['date_added'].dt.year
```

2.1 Numercial Columns Distribution

[6]:

```
df.describe()
```

[6]:

```
       release_year  date_added_year
count   8807.000000      8797.000000
mean    2014.180198      2018.871888
std        8.819312         1.574243
min     1925.000000      2008.000000
25%     2013.000000      2018.000000
50%     2017.000000      2019.000000
75%     2019.000000      2020.000000
max     2021.000000      2021.000000
```

2.2 Categorical Columns Distribution

[7]:

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

[7]:

```
       show_id   type                 title       director  \
count     8807   8807                  8807           6173
unique    8807      2                  8807           4528
top         s1  Movie  Dick Johnson Is Dead  Rajiv Chilaka
freq         1   6131                     1             19

                      cast        country rating  duration  \
count                 7982           7976   8803      8804
unique                7692            748     17       220
top     David Attenborough  United States  TV-MA  1 Season
freq                    19           2818   3207      1793

                           listed_in  \
count                           8807
unique                           514
top     Dramas, International Movies
```

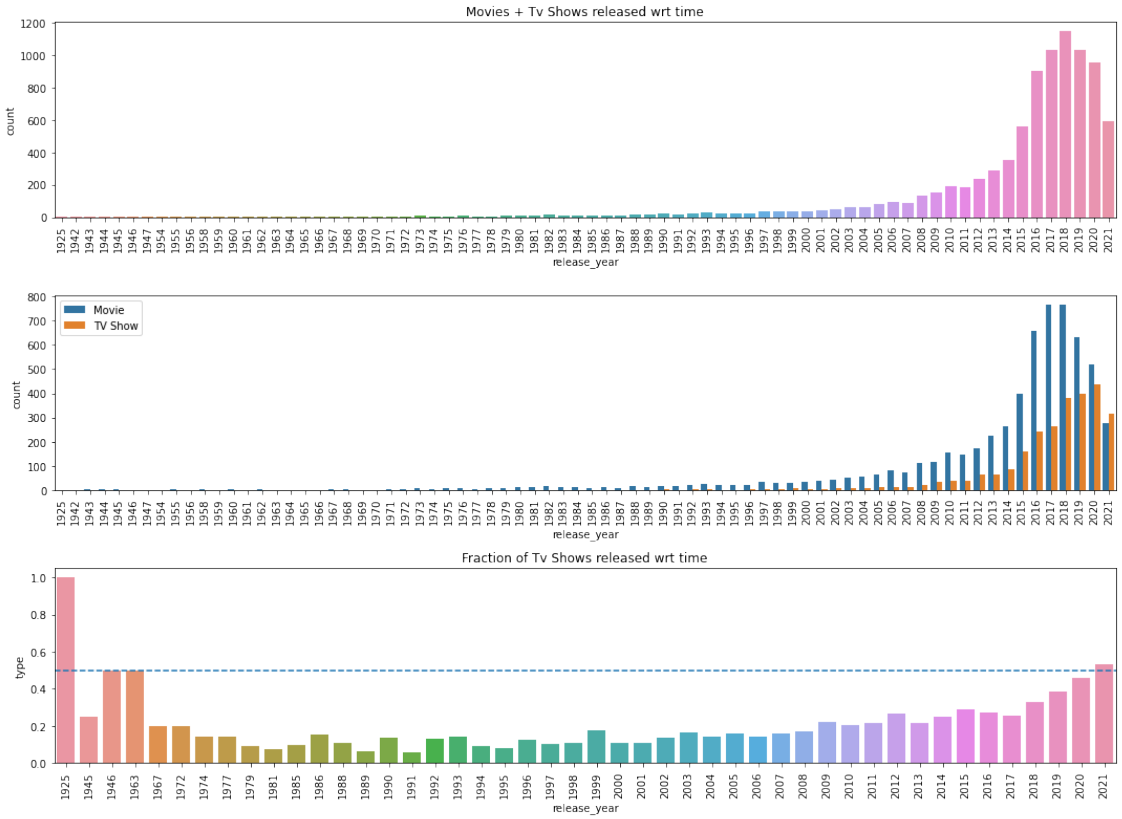
```
freq                             362

                                              description
count                                                8807
unique                                               8775
top     Paranormal activity at a lush, abandoned prope…
freq                                                    4
```

3 Graphical Eda

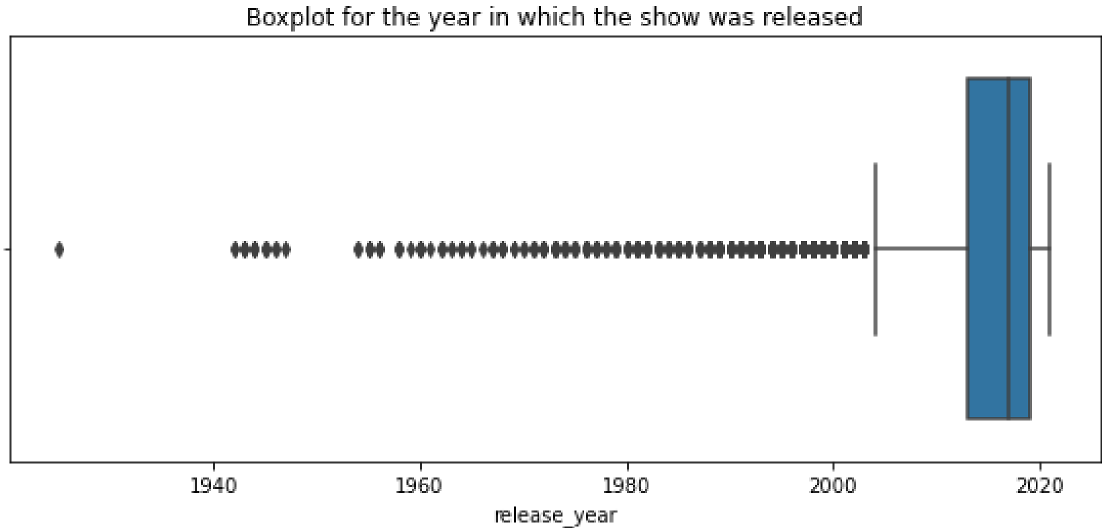
3.1 Show Release Year and Show Added on Netflix Year Analysis

3.1.1 Trend of Movies/TV Shows released :

[8]:
```
fig, ax = plt.subplots(3,1,figsize=(15,11))
sns.countplot(x='release_year',data=df,ax=ax[0])
ax[0].set_xticklabels(ax[0].get_xticklabels(),rotation=90)
ax[0].set_title('Movies + Tv Shows released wrt time')
sns.countplot(x='release_year',hue='type',data=df,ax=ax[1])
ax[1].set_xticklabels(ax[1].get_xticklabels(),rotation=90)
ax[1].legend(loc='upper left')
tmp = pd.DataFrame(df.groupby('release_year')['type'].value_counts(1))
tmp = tmp.iloc[tmp.index.get_level_values(1) == 'TV Show'].droplevel(1).
 ↪reset_index()
sns.barplot(x='release_year',y='type',data=tmp,ax=ax[2])
ax[2].axhline(0.5,ls='--')
ax[2].set_xticklabels(ax[2].get_xticklabels(),rotation=90)
ax[2].set_title('Fraction of Tv Shows released wrt time')
fig.tight_layout()
print()
```



[9]:
```
plt.rcParams['figure.figsize'] = 10,4
g = sns.boxplot(df['release_year'])
g.set_title('Boxplot for the year in which the show was released')
print()
```

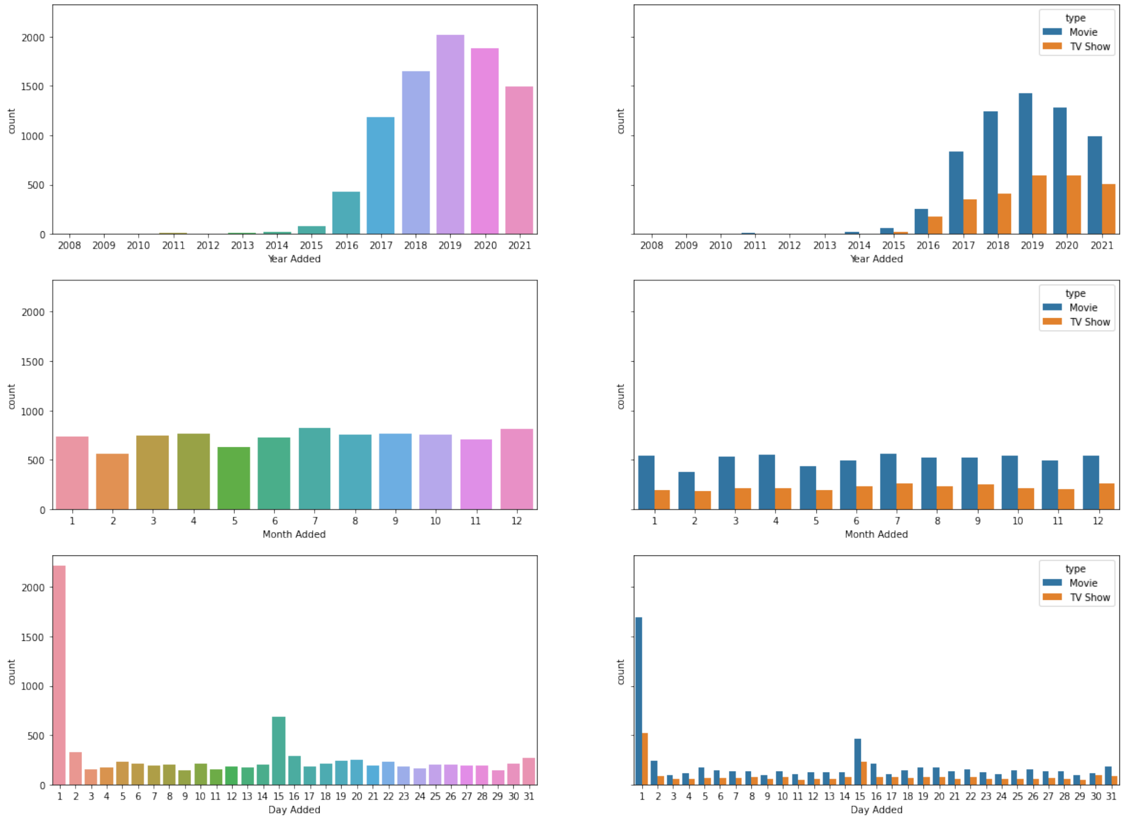


Insights : - Content creation has exploded post 2015, reaching maximum in 2018 - The fraction of TV shows has been steadily increasing since 2017, surpassing 50% mark in 2021

3.1.2 Trend of Movies/TV Shows added on Netflix :

[10]:

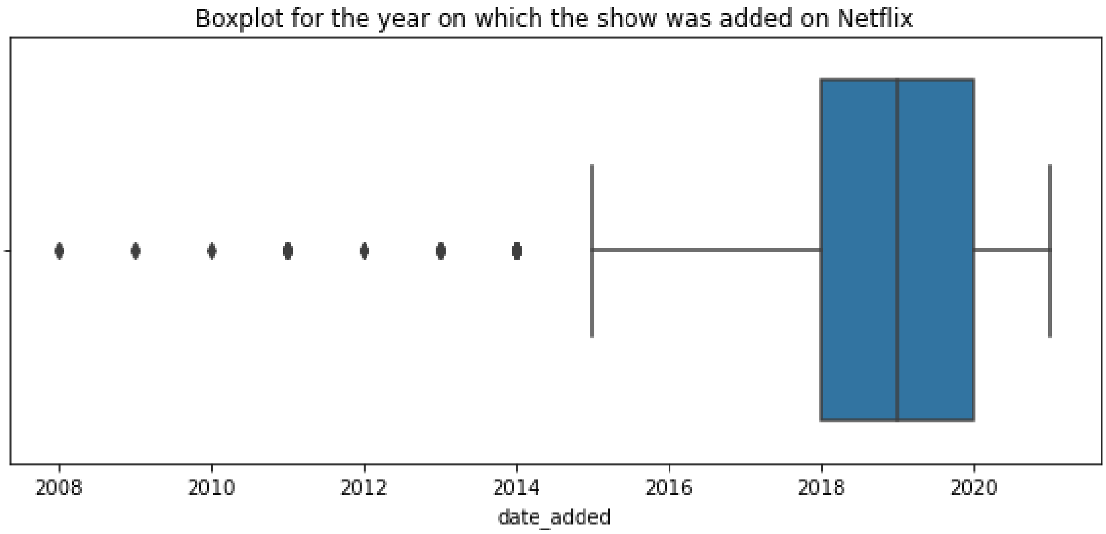
```
fig,ax = plt.subplots(3,2,figsize=(20,15), sharey=True)
sns.countplot(df['date_added'].dt.year.dropna().astype(int),ax=ax[0][0])
ax[0][0].set_xlabel('Year Added')
sns.countplot(df['date_added'].dt.month.dropna().astype(int),ax=ax[1][0])
ax[1][0].set_xlabel('Month Added')
sns.countplot(df['date_added'].dt.day.dropna().astype(int),ax=ax[2][0])
ax[2][0].set_xlabel('Day Added')

sns.countplot(df['date_added'].dt.year.dropna().
 ↪astype(int),ax=ax[0][1],hue=df['type'])
ax[0][1].set_xlabel('Year Added')
sns.countplot(df['date_added'].dt.month.dropna().
 ↪astype(int),ax=ax[1][1],hue=df['type'])
ax[1][1].set_xlabel('Month Added')
sns.countplot(df['date_added'].dt.day.dropna().
 ↪astype(int),ax=ax[2][1],hue=df['type'])
ax[2][1].set_xlabel('Day Added')
print()
```



Insights : - Content added on netflix has grown exponentially since 2015, reaching maximum in 2019 - Content has been added uniformly across all months - Netflix has a strong preference of adding content on either 1st or 15th of a month

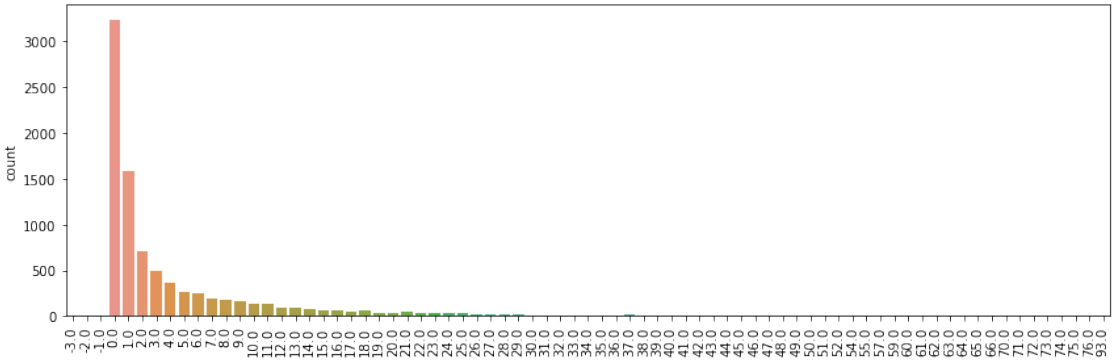
[11]:

```
plt.rcParams['figure.figsize'] = 10,4
g = sns.boxplot(df['date_added'].dt.year)
g.set_title('Boxplot for the year on which the show was added on Netflix')
print()
```



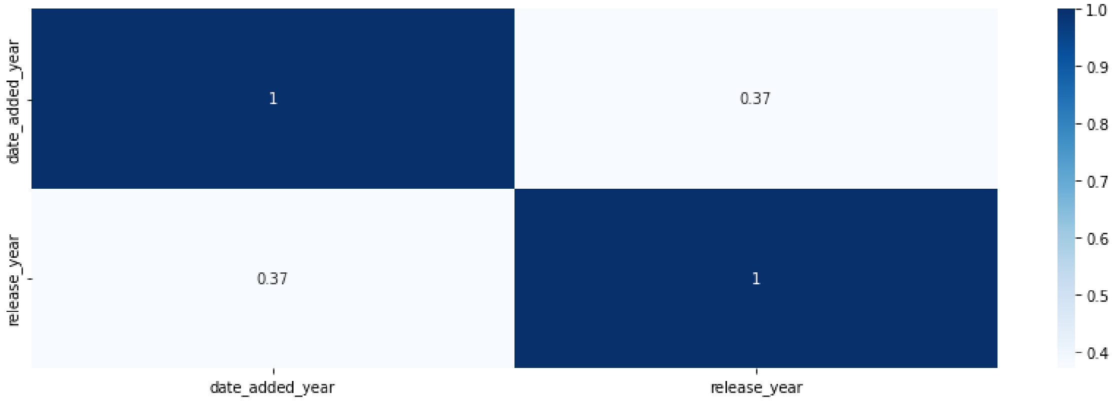
3.1.3 Release Date vs Added date :

[12]:
```
plt.rcParams['figure.figsize'] = 12,4
plt.rcParams['figure.autolayout'] = True
g = sns.countplot((df['date_added'].dt.year - df['release_year']))
g.set_xticklabels(g.get_xticklabels(), rotation=90)
print()
```

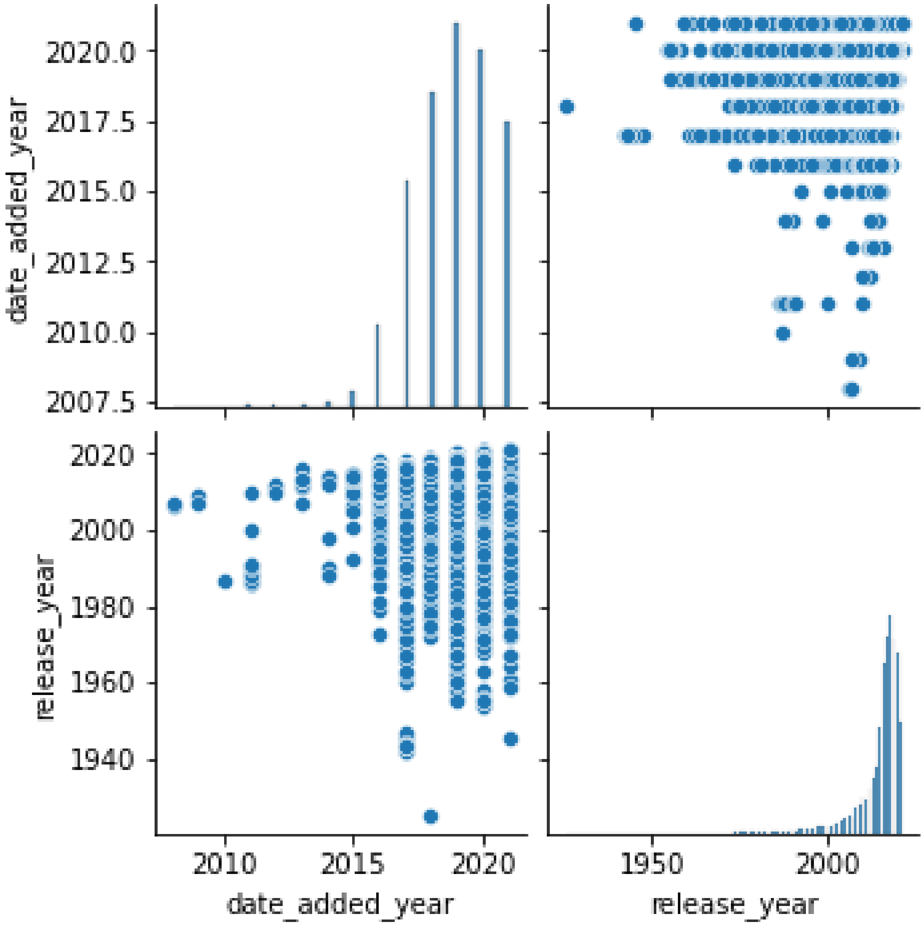


[13]:
```
# We dont use Pearson correlation because columns are not normally distributed
sns.heatmap(df[['date_added_year','release_year']].corr(method='spearman'),␣
 ↪annot=True, cmap='Blues')
```

[13]:
```
<AxesSubplot:>
```



```
[14]: sns.pairplot(df[['date_added_year','release_year']])
      print()
```



Insight : - We can see most of the content is added the same year as it was released

---

3.2 Director/Cast/Genre Analysis :

3.2.1 Top director/cast/genres :

[15]:
```
print(f"Mean null values in director column : {round(df['director'].isna().
 ↪mean() * 100)}%")
print(f"Mean null values in cast column : {round(df['cast'].isna().mean() *␣
 ↪100)}%")
df.groupby('type')[['director','cast']].apply(lambda x : round(x.isna().
 ↪mean()*100))
```

```
Mean null values in director column : 30%
Mean null values in cast column : 9%
```

[15]:
```
         director  cast
type
Movie         3.0   8.0
TV Show      91.0  13.0
```

Insight : - TV shows have 91% missing directors data!

---

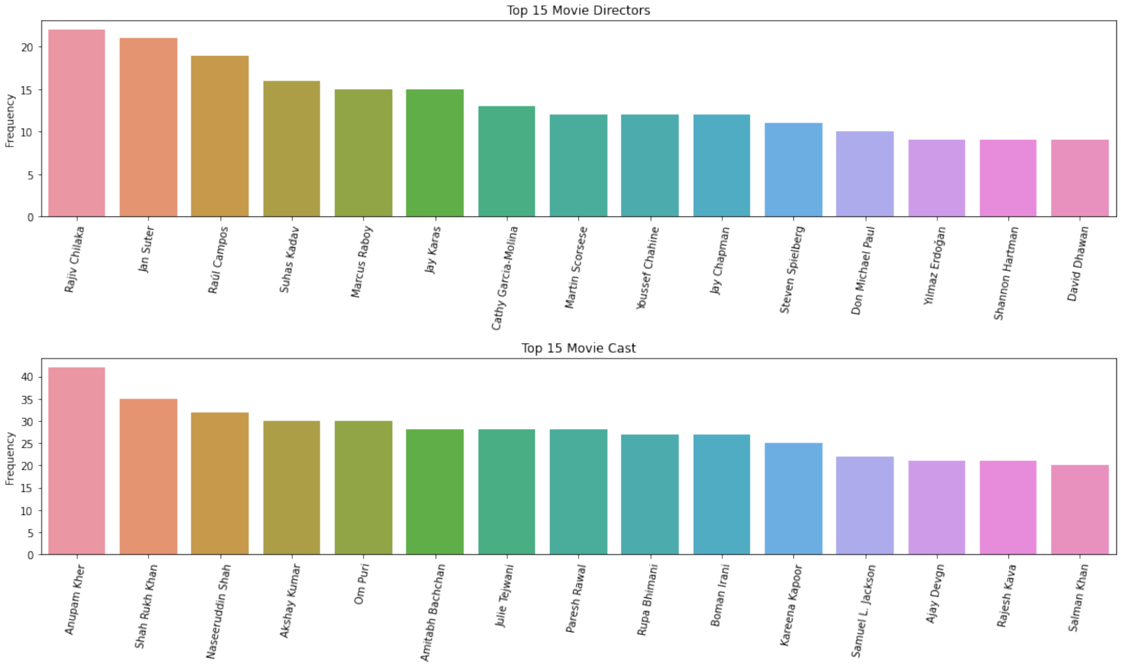
[29]:
```
def flatten_list(og_list):
    """ Flattens the input list.
    input -> ['raunak','abhinav','abhjieet, rupesh, aryan']
    output -> ['raunak','abhinav','abhjieet', 'rupesh', 'aryan']"""
    flattend_list = []
    for i in og_list:
        if isinstance(i,str):
            flattend_list.extend([i.strip() for i in i.split(',')])
    return flattend_list
```

[17]:
```
#movie_dirs contains a list of all the movie directors
movie_dirs = flatten_list(df.loc[df['type'] == 'Movie','director'].tolist())
movie_cast = flatten_list(df.loc[df['type'] == 'Movie','cast'].tolist())

tv_dirs = flatten_list(df.loc[df['type'] == 'TV Show','director'].tolist())
tv_cast = flatten_list(df.loc[df['type'] == 'TV Show','cast'].tolist())
```

Top 15 directors/casts in movies :

[18]:

```
fig, ax = plt.subplots(2,1,figsize=(15,9))
sns.barplot(x=pd.Series(movie_dirs).value_counts().head(15).index, y=pd.
 ↪Series(movie_dirs).value_counts().head(15).values, ax=ax[0])
ax[0].set_xticklabels(ax[0].get_xticklabels(), rotation=80)
ax[0].set_title('Top 15 Movie Directors')
ax[0].set_ylabel('Frequency')
sns.barplot(x=pd.Series(movie_cast).value_counts().head(15).index, y=pd.
 ↪Series(movie_cast).value_counts().head(15).values, ax=ax[1])
ax[1].set_xticklabels(ax[1].get_xticklabels(), rotation=80)
ax[1].set_title('Top 15 Movie Cast')
ax[1].set_ylabel('Frequency')
fig.tight_layout()
print()
```

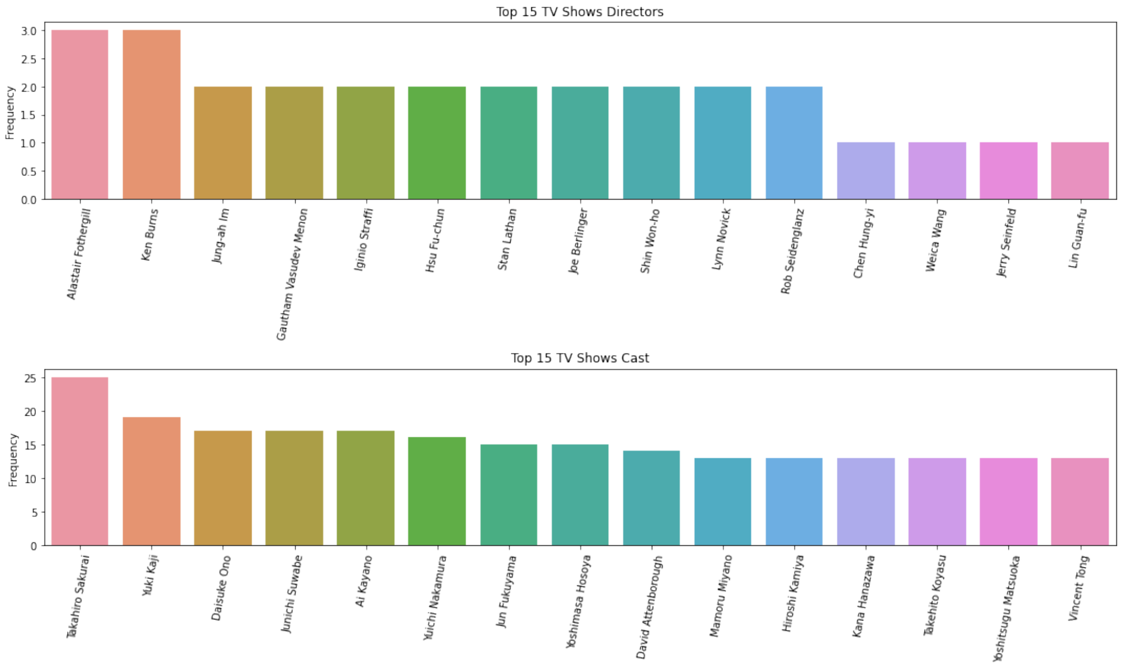


Top 15 directors/casts in TV Shows :

[19]:

```
fig, ax = plt.subplots(2,1,figsize=(15,9))
sns.barplot(x=pd.Series(tv_dirs).value_counts().head(15).index, y=pd.
 ↪Series(tv_dirs).value_counts().head(15).values, ax=ax[0])
ax[0].set_xticklabels(ax[0].get_xticklabels(), rotation=80)
ax[0].set_title('Top 15 TV Shows Directors')
ax[0].set_ylabel('Frequency')
```

```
sns.barplot(x=pd.Series(tv_cast).value_counts().head(15).index, y=pd.
 ↪Series(tv_cast).value_counts().head(15).values, ax=ax[1])
ax[1].set_xticklabels(ax[1].get_xticklabels(), rotation=80)
ax[1].set_title('Top 15 TV Shows Cast')
ax[1].set_ylabel('Frequency')
fig.tight_layout()
print()
```



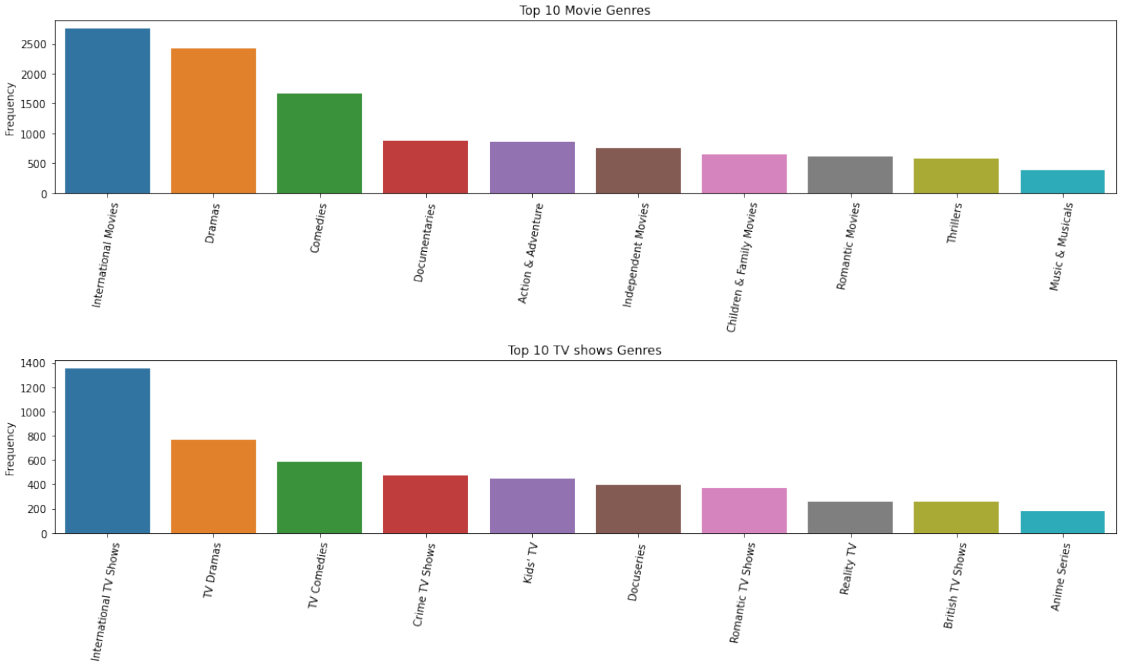
Insight : - Best Movie Director : Rajiv Chilaka (Animator for Chota Bheem) - Best Movie Cast : Anupam Kher (Big Bollywood Star) - Best TV Show Director : Alastair Fothergill (Nature Documentaries like Our Planet) - Best TV Show Cast : Takahiro Sakurai (Japanese Voice actor eg Jujutsu Kaisen)

[20]:
```
movie_genres = flatten_list(df.loc[df['type'] == 'Movie','listed_in'].tolist())

tv_genres = flatten_list(df.loc[df['type'] == 'TV Show','listed_in'].tolist())
```

Top 10 genres for movies/TV Shows :

[21]:
```
fig, ax = plt.subplots(2,1,figsize=(15,9))
sns.barplot(x=pd.Series(movie_genres).value_counts().head(10).index, y=pd.
 ↪Series(movie_genres).value_counts().head(10).values, ax=ax[0])
ax[0].set_xticklabels(ax[0].get_xticklabels(), rotation=80)
```

```
ax[0].set_title('Top 10 Movie Genres')
ax[0].set_ylabel('Frequency')
sns.barplot(x=pd.Series(tv_genres).value_counts().head(10).index, y=pd.
 ↪Series(tv_genres).value_counts().head(10).values, ax=ax[1])
ax[1].set_xticklabels(ax[1].get_xticklabels(), rotation=80)
ax[1].set_title('Top 10 TV shows Genres')
ax[1].set_ylabel('Frequency')
fig.tight_layout()
print()
```



Insight : - Drama and comedy are top Genres for both movies/tv shows

---

3.2.2 Countries leading in content genre-wise:

[22]:
```
tmp = df[['country','listed_in']].astype(str)
```

[23]:
```
show_genre_dict = {}
for _, row in tqdm(tmp.iterrows()): #iterate through all shows
    for genre in row['listed_in'].split(','): #iterate through all genres
        if genre.strip() not in show_genre_dict.keys(): #if genre is not in␣
 ↪dict create an empty list
```

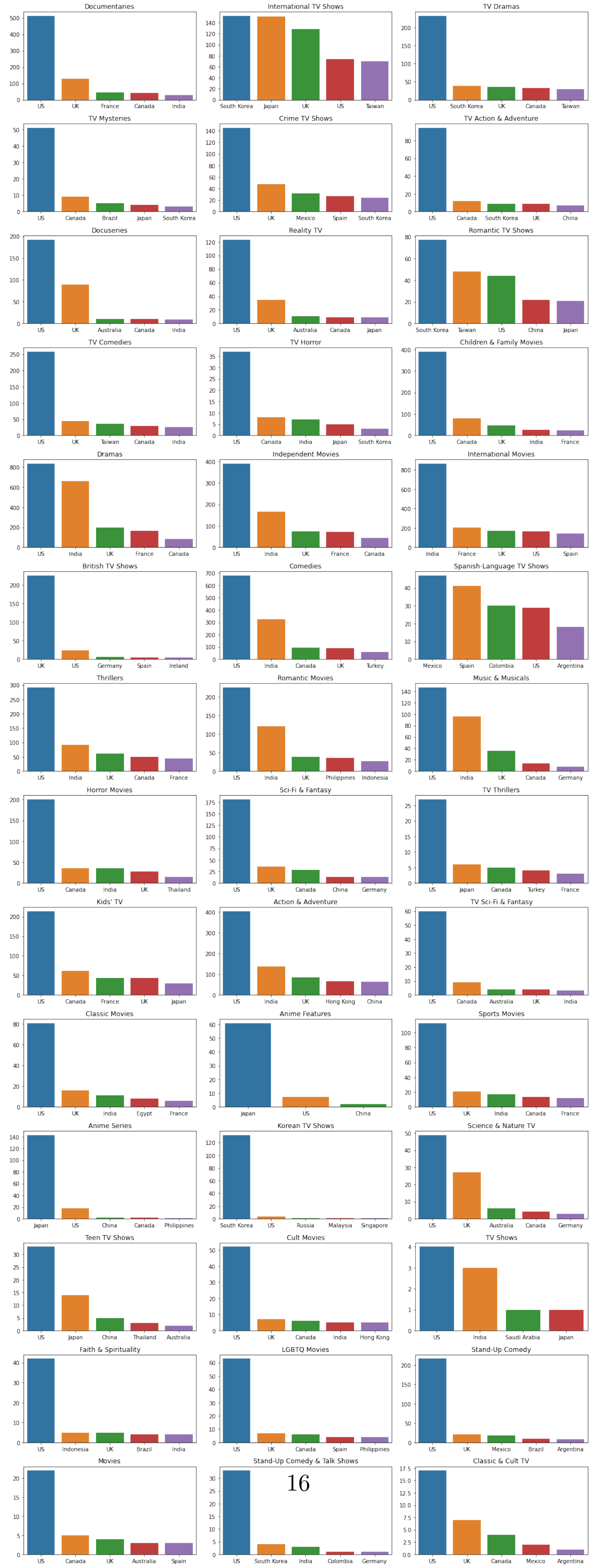
```
            show_genre_dict[genre.strip()] = []
        for country in row['country'].split(','): #iterate through all countries
            if country.strip() != 'nan':
                show_genre_dict[genre.strip()].append(country.strip())
```

```
8807it [00:01, 7508.05it/s]
```

show_genre_dict : - Keys are all the different genres (genres are extracted from listed_in column) - Values are the countries where the genre was produced (extracted from country column) - eg show_genre_dict['Dramas'] -> ['United States', 'Ghana', 'Burkina Faso', 'United Kingdom', 'Germany'] - The values can have duplicates. Finding the mode of the list we can get the country where this genre is most produced

[24]:

```
plt.rcParams['figure.figsize'] = 15,40
plt.rcParams['figure.autolayout'] = True
for i,key in enumerate(show_genre_dict.keys()):
    ax = plt.subplot(14,3,i+1)
    sns.barplot(pd.Series(show_genre_dict[key]).replace({'United States':
 ↪'US','United Kingdom':'UK'}).value_counts().head().index,
               pd.Series(show_genre_dict[key]).replace({'United States':
 ↪'US','United Kingdom':'UK'}).value_counts().head().values,
               ax=ax)
    ax.set_title(key)
```


Documentaries
0 100 200 300 400 500
US UK France Canada India
International TV Shows
0 20 40 60 80 100 120 140
South Korea Japan UK US Taiwan
TV Dramas
0 50 100 150 200
US South Korea UK Canada Taiwan
TV Mysteries
0 10 20 30 40 50
US Canada Brazil Japan South Korea
Crime TV Shows
0 20 40 60 80 100 120 140
US UK Mexico Spain South Korea
TV Action & Adventure
0 20 40 60 80
US Canada South Korea UK China
Docuseries
0 50 100 150 200
US UK Australia Canada India
Reality TV
0 20 40 60 80 100 120
US UK Australia Canada Japan
Romantic TV Shows
0 20 40 60 80
South Korea Taiwan US China Japan
TV Comedies
0 50 100 150 200 250
US UK Taiwan Canada India
TV Horror
0 5 10 15 20 25 30 35
US Canada India Japan South Korea
Children & Family Movies
0 100 200 300 400
US Canada UK India France
Dramas
0 200 400 600 800
US India UK France Canada
Independent Movies
0 100 200 300 400
US India UK France Canada
International Movies
0 200 400 600 800
India France UK US Spain
British TV Shows
0 50 100 150 200
UK US Germany Spain Ireland
Comedies
0 100 200 300 400 500 600 700
US India Canada UK Turkey
Spanish-Language TV Shows
0 10 20 30 40
Mexico Spain Colombia US Argentina
Thrillers
0 50 100 150 200 250 300
US India UK Canada France
Romantic Movies
0 50 100 150 200
US India UK Philippines Indonesia
Music & Musicals
0 20 40 60 80 100 120 140
US India UK Canada Germany
Horror Movies
0 50 100 150 200
US Canada India UK Thailand
Sci-Fi & Fantasy
0 25 50 75 100 125 150 175
US UK Canada China Germany
TV Thrillers
0 5 10 15 20 25
US Japan Canada Turkey France
Kids' TV
0 50 100 150 200
US Canada France UK Japan
Action & Adventure
0 100 200 300 400
US India UK Hong Kong China
TV Sci-Fi & Fantasy
0 10 20 30 40 50 60
US Canada Australia UK India
Classic Movies
0 20 40 60 80
US UK India Egypt France
Anime Features
0 10 20 30 40 50 60
Japan US China
Sports Movies
0 20 40 60 80 100
US UK India Canada France
Anime Series
0 20 40 60 80 100 120 140
Japan US China Canada Philippines
Korean TV Shows
0 20 40 60 80 100 120
South Korea US Russia Malaysia Singapore
Science & Nature TV
0 10 20 30 40 50
US UK Australia Canada Germany
Teen TV Shows
0 5 10 15 20 25 30
US Japan China Thailand Australia
Cult Movies
0 10 20 30 40 50
US UK Canada India Hong Kong
TV Shows
0 1 2 3 4
US India Saudi Arabia Japan
Faith & Spirituality
0 10 20 30 40
US Indonesia UK Brazil India
LGBTQ Movies
0 10 20 30 40 50 60
US UK Canada Spain Philippines
Stand-Up Comedy
0 50 100 150 200
US UK Mexico Brazil Argentina
Movies
0 5 10 15 20
US Canada UK Australia Spain
Stand-Up Comedy & Talk Shows
0 5 10 15 20 25 30
US South Korea India Colombia Germany
Classic & Cult TV
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
US UK Canada Mexico Argentina

Insight : - US is a clear leader across all genres except : - Japan in Anime Series, Anime Features - South Korea in Korean movies, International TV show, Romantic TV Shows - Mexico in spanish language tv shows - India in international movies - US is #1 in 35/42 genres

- Although netflix should continue to invest in US across all genres we will look at what is the most promising country other than US/genre leader for each genre

3.2.3 Recommendation for best upcoming country for each genre :

upcoming_country_for_each_genre : - is a dictonary - keys are genres - values is a list with 3 elements : [upcoming country, best director, best cast]

[26]:

```
upcoming_country_for_each_genre = {}
for k,v in show_genre_dict.items():
    upcoming_country = pd.Series(show_genre_dict[k]).value_counts().index[1]
    upcoming_country_for_each_genre[k] = [upcoming_country]
```

[27]:

```
for k,v in upcoming_country_for_each_genre.items():

    # Subset on country
    tmp = df[df['country'] == v[0]]
    # Subset on genre
    tmp = tmp[tmp['listed_in'].str.contains(k)]

    #best director
    dirs = flatten_list(tmp['director'])
    try:
        best_dir = max(set(dirs), key=dirs.count)
    except:
        best_dir = 'NA'

    #best cast
    cast = flatten_list(tmp['cast'])
    try:
        best_cast = max(set(cast), key=cast.count)
    except:
        best_cast = 'NA'

    upcoming_country_for_each_genre[k].extend([best_dir, best_cast])
```

format -> genre : [country , best director, best cast]

[28]:

```
upcoming_country_for_each_genre
```

```
[28]: {'Documentaries': ['United Kingdom', 'Edward Cotterill', 'Samuel West'],
       'International TV Shows': ['Japan', 'Hayato Date', 'Takahiro Sakurai'],
       'TV Dramas': ['South Korea', 'Lee Kyoungmi', 'Cho Seong-ha'],
       'TV Mysteries': ['Canada', 'NA', 'Jim Watson'],
       'Crime TV Shows': ['United Kingdom', 'Ellena Wood', 'Charlie Creed-Miles'],
       'TV Action & Adventure': ['Canada', 'NA', 'Brianna Daguanno'],
       'Docuseries': ['United Kingdom', 'Alastair Fothergill', 'David Attenborough'],
       'Reality TV': ['United Kingdom', 'Andy Devonshire', 'Nadiya Hussain'],
       'Romantic TV Shows': ['Taiwan', 'Chang Chin-jung', 'Amanda Chou'],
       'TV Comedies': ['United Kingdom', 'Gordon Anderson', 'Ruth Bratt'],
       'TV Horror': ['Canada', 'NA', 'Greyston Holt'],
       'Children & Family Movies': ['Canada', 'Vivieno Caldinelli', 'Michela Luci'],
       'Dramas': ['India', 'Anurag Kashyap', 'Shah Rukh Khan'],
       'Independent Movies': ['India', 'Qaushiq Mukherjee', 'Naseeruddin Shah'],
       'International Movies': ['France', 'Thierry Donard', 'Wille Lindberg'],
       'British TV Shows': ['United States', 'NA', 'Celine Buckens'],
       'Comedies': ['India', 'David Dhawan', 'Anupam Kher'],
       'Spanish-Language TV Shows': ['Spain', 'Mateo Gil', 'José Sacristán'],
       'Thrillers': ['India', 'Anurag Kashyap', 'Nawazuddin Siddiqui'],
       'Romantic Movies': ['India', 'Imtiaz Ali', 'Akshay Kumar'],
       'Music & Musicals': ['India', 'Mastan Alibhai Burmawalla', 'Akshay Kumar'],
       'Horror Movies': ['Canada', 'Clay Staub', 'Booboo Stewart'],
       'Sci-Fi & Fantasy': ['United Kingdom', 'Johnny Kevorkian', 'Welile Nzunza'],
       'TV Thrillers': ['Japan', 'NA', 'Minami Takayama'],
       "Kids' TV": ['Canada', 'NA', 'Jordan Clark'],
       'Action & Adventure': ['India', 'Ram Gopal Varma', 'Amitabh Bachchan'],
       'TV Sci-Fi & Fantasy': ['Canada', 'NA', 'Brianna Daguanno'],
       'Classic Movies': ['United Kingdom', 'Terry Jones', 'Eric Idle'],
       'Anime Features': ['United States', 'Koji Morimoto', 'NA'],
       'Sports Movies': ['United Kingdom', 'Daniel Kontur', 'Ryan Howard'],
       'Anime Series': ['United States', 'NA', 'Richard Armitage'],
       'Korean TV Shows': ['United States', 'NA', 'Jung Sun-hye'],
       'Science & Nature TV': ['United Kingdom', 'NA', 'David Attenborough'],
       'Teen TV Shows': ['Japan', 'Takuya Igarashi', 'Takahiro Sakurai'],
       'Cult Movies': ['United Kingdom', 'Danny Boyle', 'Peter Mullan'],
       'TV Shows': ['India', 'Gautham Vasudev Menon', 'Chandan Anand'],
       'Faith & Spirituality': ['Indonesia', 'Chairun Nissa', 'Fedi Nuril'],
       'LGBTQ Movies': ['United Kingdom', 'Jon Carey', 'Rory J. Saper'],
       'Stand-Up Comedy': ['United Kingdom', 'Chris Howe', 'Bill Hicks'],
       'Movies': ['Canada', 'Justin G. Dyck', 'Robb Wells'],
       'Stand-Up Comedy & Talk Shows': ['South Korea',
        'Jung-ah Im',
        'Si-kyung Sung'],
       'Classic & Cult TV': ['United Kingdom', 'Michael Cumming', 'Matt Berry']}
```