netfilx-buss-case-study

February 20, 2024

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

[ ]:
```
#Importing Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#loading DataSet
data = pd.read_csv('F:\\buss_cass\\data\\netflix.csv')
```

[2]:
```
#checking the Dataset
data.head()
```

[2]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…	NaN	September 24, 2021	2021	TV-MA	1 Season	
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…	India	September 24, 2021	2021	TV-MA	2 Seasons	

```
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…
```

[109]:
```
data.columns
```

[109]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

[3]:
```
#exploring the data

#shape
data.shape
```

[3]:
```
(8807, 12)
```

[110]:
```
data.ndim
```

[110]:
```
2
```

[4]:
```
#data types
data.dtypes
```

[4]:
```
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
```

[ ]:

[5]:
```
#finding null values

data.T.apply(lambda x: x.isnull().sum(),axis =1)
```

[5]:
```
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
```

[6]:
```
#statistical summary
data.describe()
```

[6]:
```
       release_year
count   8807.000000
mean    2014.180198
std        8.819312
min     1925.000000
25%     2013.000000
50%     2017.000000
75%     2019.000000
max     2021.000000
```

[ ]:
```

```

[7]:
```
#Data Cleaning and Unnesting
```

[8]:
```
# the columns of director, cast, country value NaN updated to Unkown

dict = {'director': 'Unknown Director', 'cast' : 'Unknown Actor', 'country' :␣
 ↪'Unknown country'}
data.fillna(value = dict , inplace = True)
data.head(10)
```

[8]:
```
  show_id     type                              title  \
0      s1    Movie               Dick Johnson Is Dead
1      s2  TV Show                       Blood & Water
2      s3  TV Show                           Ganglands
3      s4  TV Show             Jailbirds New Orleans
```

```
4      s5  TV Show                        Kota Factory
5      s6  TV Show                       Midnight Mass
6      s7    Movie  My Little Pony: A New Generation
7      s8    Movie                              Sankofa
8      s9  TV Show     The Great British Baking Show
9     s10    Movie                         The Starling

                      director  \
0               Kirsten Johnson
1             Unknown Director
2               Julien Leclercq
3             Unknown Director
4             Unknown Director
5                Mike Flanagan
6  Robert Cullen, José Luis Ucha
7                 Haile Gerima
8               Andy Devonshire
9                Theodore Melfi

                                          cast  \
0                                Unknown Actor
1  Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…
2  Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi…
3                                Unknown Actor
4  Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…
5  Kate Siegel, Zach Gilford, Hamish Linklater, H…
6  Vanessa Hudgens, Kimiko Glenn, James Marsden, …
7  Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D…
8  Mel Giedroyc, Sue Perkins, Mary Berry, Paul Ho…
9  Melissa McCarthy, Chris O'Dowd, Kevin Kline, T…

                                       country          date_added  \
0                                United States  September 25, 2021
1                                South Africa  September 24, 2021
2                              Unknown country  September 24, 2021
3                              Unknown country  September 24, 2021
4                                        India  September 24, 2021
5                              Unknown country  September 24, 2021
6                              Unknown country  September 24, 2021
7  United States, Ghana, Burkina Faso, United Kin…  September 24, 2021
8                               United Kingdom  September 24, 2021
9                                United States  September 24, 2021

   release_year rating   duration  \
0          2020  PG-13     90 min
1          2021  TV-MA  2 Seasons
2          2021  TV-MA   1 Season
```

```
3          2021  TV-MA   1 Season
4          2021  TV-MA  2 Seasons
5          2021  TV-MA   1 Season
6          2021     PG     91 min
7          1993  TV-MA    125 min
8          2021  TV-14  9 Seasons
9          2021  PG-13    104 min

                                          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 …
5                TV Dramas, TV Horror, TV Mysteries
6                          Children & Family Movies
7   Dramas, Independent Movies, International Movies
8                       British TV Shows, Reality TV
9                                    Comedies, Dramas

                                          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…
5  The arrival of a charismatic young priest brin…
6  Equestria's divided. But a bright-eyed hero be…
7  On a photo shoot in Ghana, an American model s…
8  A talented batch of amateur bakers face off in…
9  A woman adjusting to life after a loss contend…
```

[9]:
```
#statistical summary
data.describe()
```

[9]:
```
       release_year
count   8807.000000
mean    2014.180198
std        8.819312
min     1925.000000
25%     2013.000000
50%     2017.000000
75%     2019.000000
max     2021.000000
```

[ ]:
```

```

[10]:
```
data.T.apply(lambda x: x.isnull().sum(),axis =1)
```

[10]:
```
show_id          0
type             0
title            0
director         0
cast             0
country          0
date_added      10
release_year     0
rating           4
duration         3
listed_in        0
description      0
dtype: int64
```

Note: Some columns contain null values in 'date_added', 'release_year', and 'rating' fields. These null values are systematically handled to ensure data integrity and to optimize data analysis time.

[ ]:
```

```

[11]:
```
#unnesting the listed_in

gen = pd.DataFrame(data[['show_id', 'title', 'listed_in']])
gen['listed_in'] = gen['listed_in'].str.split(', ')
gen = gen.explode('listed_in').reset_index()
gen.head()
```

[11]:

	index	show_id	title	listed_in
0	0	s1	Dick Johnson Is Dead	Documentaries
1	1	s2	Blood & Water	International TV Shows
2	1	s2	Blood & Water	TV Dramas
3	1	s2	Blood & Water	TV Mysteries
4	2	s3	Ganglands	Crime TV Shows

[12]:
```
#count of movies based on gerne top 10
count_gen = gen['listed_in'].value_counts()
#top 10
count_gen.head(10)
```

[12]:
```
listed_in
International Movies        2752
Dramas                      2427
Comedies                    1674
International TV Shows      1351
Documentaries                869
Action & Adventure           859
```

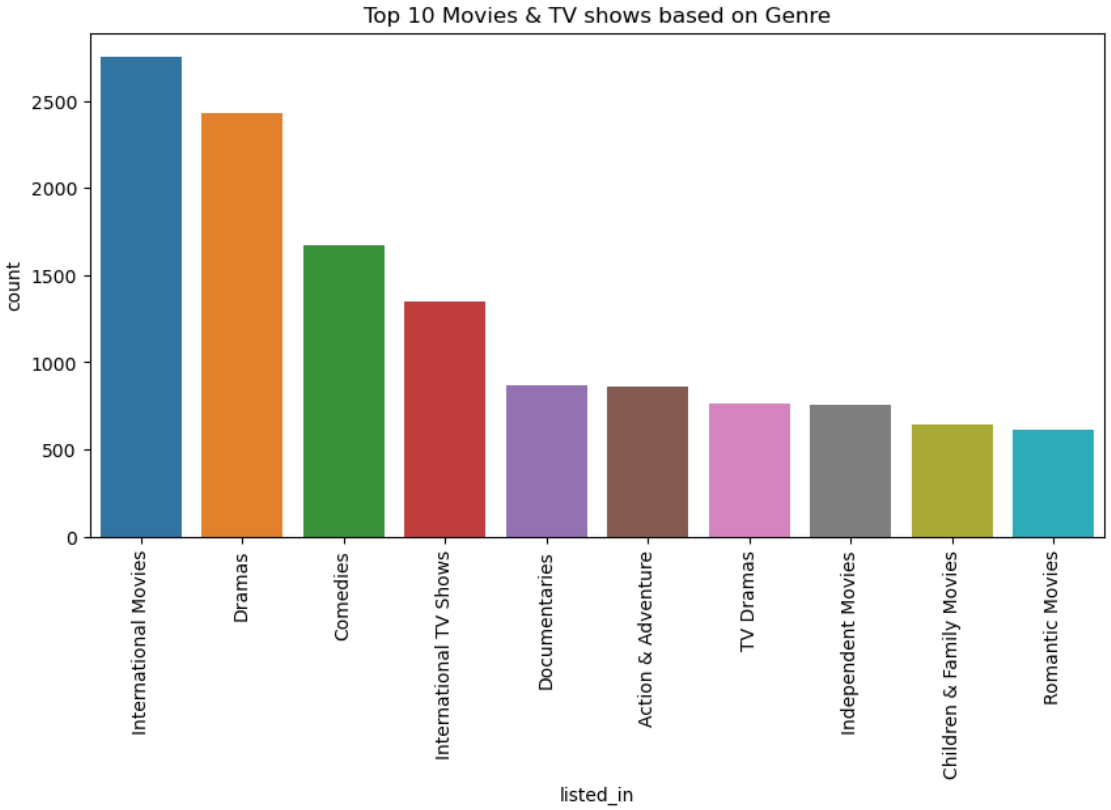
```
TV Dramas                     763
Independent Movies            756
Children & Family Movies      641
Romantic Movies               616
Name: count, dtype: int64
```

[13]:
```
#bottom 10
count_gen.tail(10)
```

[13]:
```
listed_in
TV Horror                       75
Anime Features                  71
Cult Movies                     71
Teen TV Shows                   69
Faith & Spirituality            65
TV Thrillers                    57
Movies                          57
Stand-Up Comedy & Talk Shows    56
Classic & Cult TV               28
TV Shows                        16
Name: count, dtype: int64
```

[ ]:
```

```

[108]:
```
#graphical representation
plt.figure(figsize = (10, 5))
sns.countplot(data = gen, x= gen['listed_in'], order= gen['listed_in'].
 ↪value_counts().index[:10])
plt.xticks(rotation=90)
plt.title('Top 10 Movies & TV shows based on Genre')
plt.show()
```



[ ]:

[15]:
```
#unnesting the country

con = pd.DataFrame(data[['show_id', 'title', 'country','type' ]])
con['country'] = con['country'].str.split(', ')
con = con.explode('country')
con.head()
```

[15]:

	show_id	title	country	type
0	s1	Dick Johnson Is Dead	United States	Movie
1	s2	Blood & Water	South Africa	TV Show
2	s3	Ganglands	Unknown country	TV Show
3	s4	Jailbirds New Orleans	Unknown country	TV Show
4	s5	Kota Factory	India	TV Show

[ ]:

[16]:
```
# dropping the duplicates
con = con.drop_duplicates(subset=['country', 'title'])
```

[17]:
```
con
```

[17]:
```
     show_id                  title          country     type
0         s1   Dick Johnson Is Dead    United States    Movie
1         s2          Blood & Water     South Africa  TV Show
2         s3              Ganglands  Unknown country  TV Show
3         s4  Jailbirds New Orleans  Unknown country  TV Show
4         s5           Kota Factory            India  TV Show
…       …                      …              …      …
8802   s8803                Zodiac    United States    Movie
8803   s8804            Zombie Dumb  Unknown country  TV Show
8804   s8805             Zombieland    United States    Movie
8805   s8806                  Zoom    United States    Movie
8806   s8807                Zubaan            India    Movie

[10845 rows x 4 columns]
```

[151]:
```
#count of available movies by country
movies = con[(con['type'] == 'Movie') & (con['title'].nunique()) &␣
 ↪(con['country'] != 'Unknown country')]
top_country = movies['country'].value_counts().head(10)
top_country
```

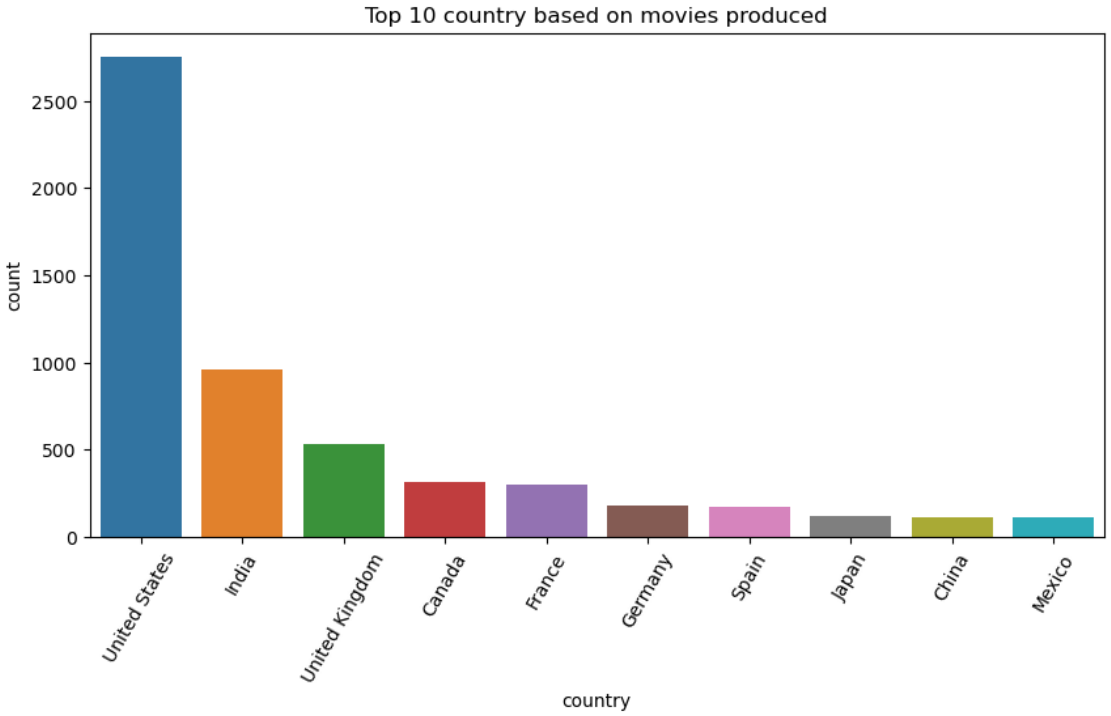
[151]:
```
country
United States     2751
India              962
United Kingdom     532
Canada             319
France             303
Germany            182
Spain              171
Japan              119
China              114
Mexico             111
Name: count, dtype: int64
```

[150]:
```
top_country.index
```

[150]:
```
Index(['United States', 'India', 'United Kingdom', 'Canada', 'France',
       'Germany', 'Spain', 'Japan', 'China', 'Mexico'],
      dtype='object', name='country')
```

[19]:
```
# top 10 country produced movies

#graphical representation
plt.figure(figsize= (10,5))
```

```
sns.countplot(data = movies , x = 'country',order= movies['country'].
 ↪value_counts().index[:10])
plt.xticks(rotation=60)
plt.title('Top 10 country based on movies produced')
plt.show()
```



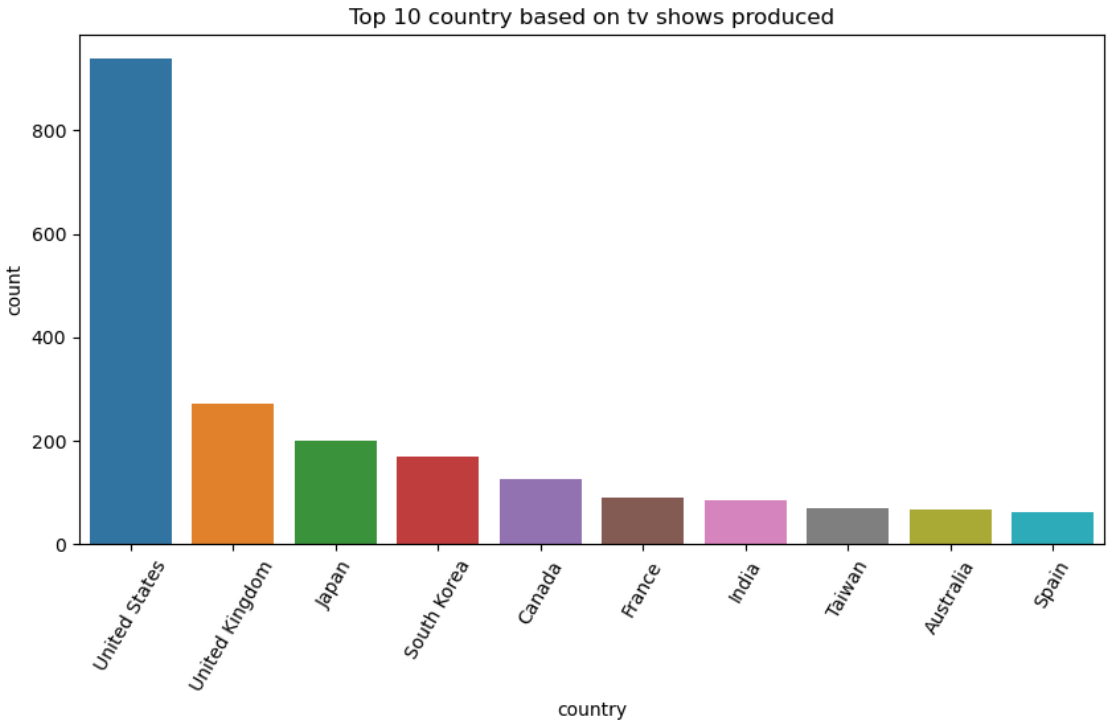
[ ]:

[20]:
```
#count of available Tv shows by country
tv_shows = con[(con['type'] == 'TV Show') & (con['title'].nunique()) &␣
 ↪(con['country'] != 'Unknown country')]
tv_shows['country'].value_counts().head(10)
```

[20]:
```
country
United States     938
United Kingdom    272
Japan             199
South Korea       170
Canada            126
France             90
India              84
Taiwan             70
Australia          66
```

```
Spain                61
Name: count, dtype: int64
```

[21]:
```
# top 10 country produced TV Shows

#graphical represention
plt.figure(figsize= (10,5))
sns.countplot(data = tv_shows , x = 'country',order= tv_shows['country'].
 ↪value_counts().index[:10])
plt.xticks(rotation=60)
plt.title('Top 10 country based on tv shows produced')
plt.show()
```



[177]:
```
# mergeing the countrys dataframe and gener dataframe
merged_df = pd.merge(gen, con, how='inner', on='title')
#dropping columns
merged_df = merged_df.drop(columns = ['index','show_id_x', 'show_id_y'])
# removing duplicates
merged_df = merged_df.drop_duplicates(subset=['listed_in','country', 'title'])
```

[178]:
```
merged_df.head()
```

[178]:

	title	listed_in	country	type
0	Dick Johnson Is Dead	Documentaries	United States	Movie
1	Blood & Water	International TV Shows	South Africa	TV Show
2	Blood & Water	TV Dramas	South Africa	TV Show
3	Blood & Water	TV Mysteries	South Africa	TV Show
4	Ganglands	Crime TV Shows	Unknown country	TV Show

[179]:
```
# filtering the top 10 country by produced more movies
top_country.index
```

[179]:
```
Index(['United States', 'India', 'United Kingdom', 'Canada', 'France',
       'Germany', 'Spain', 'Japan', 'China', 'Mexico'],
      dtype='object', name='country')
```

[180]:
```
merged_df = merged_df[merged_df['country'].isin(top_country.index)]
merged_df
```

[180]:

	title	listed_in	country	type
0	Dick Johnson Is Dead	Documentaries	United States	Movie
9	Kota Factory	International TV Shows	India	TV Show
10	Kota Factory	Romantic TV Shows	India	TV Show
11	Kota Factory	TV Comedies	India	TV Show
16	Sankofa	Dramas	United States	Movie
…	…	…	…	…
23749	Zoom	Children & Family Movies	United States	Movie
23750	Zoom	Comedies	United States	Movie
23751	Zubaan	Dramas	India	Movie
23752	Zubaan	International Movies	India	Movie
23753	Zubaan	Music & Musicals	India	Movie

[15787 rows x 4 columns]

[181]:
```
# Group the data by Country and Genre
grouped = merged_df.groupby(['country', 'listed_in']).size().
 ↪reset_index(name='Count')
```

[182]:
```
grouped = grouped.sort_values(by=['country', 'Count'], ascending=False)
```

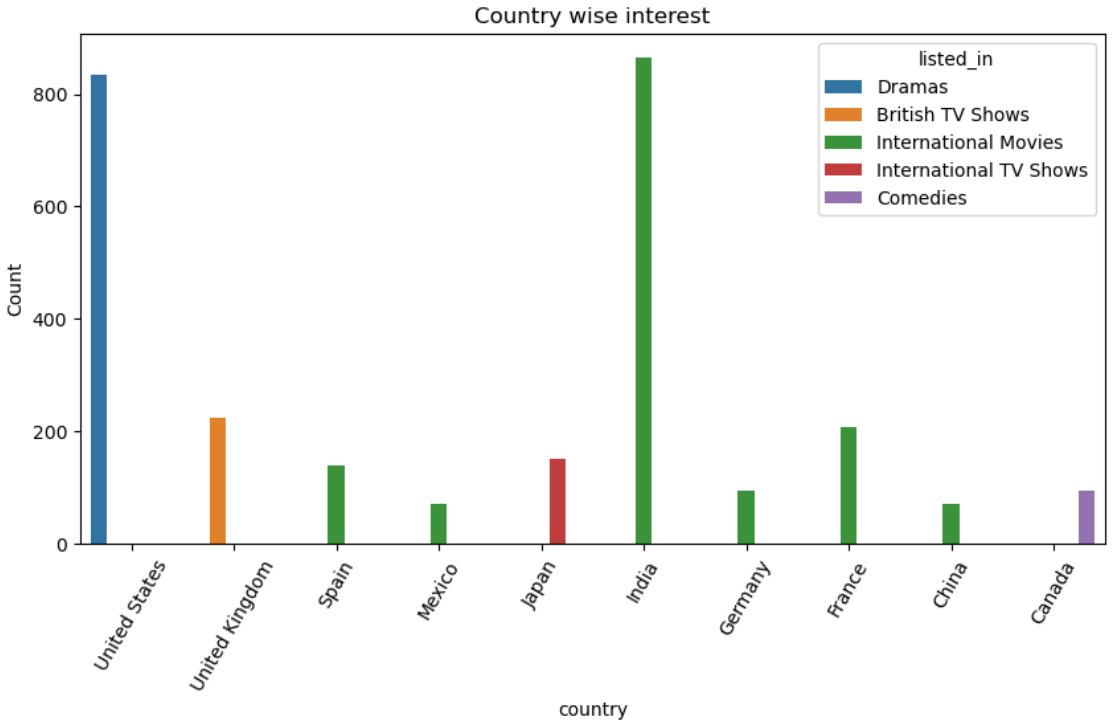
[183]:
```
top_genre_by_country = grouped.groupby('country').head(1)
top_genre_by_country
```

[183]:

	country	listed_in	Count
317	United States	Dramas	835
269	United Kingdom	British TV Shows	225
249	Spain	International Movies	140
217	Mexico	International Movies	70
185	Japan	International TV Shows	151

```
148           India     International Movies    864
115         Germany     International Movies     94
81           France     International Movies    207
51            China     International Movies     71
5            Canada                   Comedies     94
```

[184]:
```
#graph
plt.figure(figsize= (10,5))
sns.barplot(x='country', y='Count', hue='listed_in', data=top_genre_by_country)
plt.xticks(rotation=60)
plt.title('Country wise interest')
plt.show()
```



[ ]:

[ ]:

[22]:
```
#finding total nuumber of MOVIES & TV SHOWS
total_count = data[['title', 'type']]
```

[23]:
```
total_count
```

[23]:
```
                      title     type
0      Dick Johnson Is Dead    Movie
1              Blood & Water  TV Show
2                  Ganglands  TV Show
3     Jailbirds New Orleans  TV Show
4               Kota Factory  TV Show
…                        …      …
8802                Zodiac    Movie
8803            Zombie Dumb  TV Show
8804             Zombieland    Movie
8805                   Zoom    Movie
8806                 Zubaan    Movie

[8807 rows x 2 columns]
```

[ ]:

[24]:
```
#count of movies
mov_count = total_count[(total_count['type'] == 'Movie')].nunique()

x = mov_count['title']
x
```

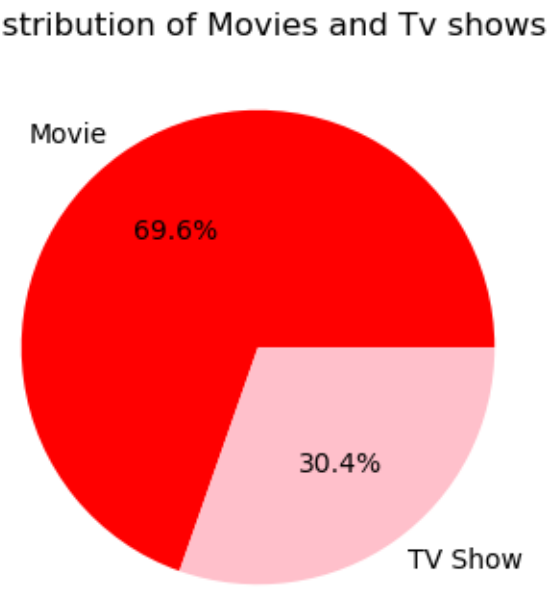
[24]:
```
6131
```

[25]:
```
#count of TV shows
tv_count = total_count[(total_count['type'] == 'TV Show')].nunique()

y = tv_count['title']
y
```

[25]:
```
2676
```

[26]:
```
li = [x, y]
```

[27]:
```
# total movies vs Tv Show availble
plt.figure(figsize = (5,4))
plt.pie(li, labels = total_count['type'].unique(), colors = ['red', 'pink'],␣
 ↪autopct = "%2.1f%%")
plt.title("Distribution of Movies and Tv shows")
plt.show()
```



[ ]:

[28]:
```
# duration of movies & TV shows
total_mv_tv = data[['title', 'type', 'duration']]
```

[29]:
```
total_mv_tv.head(5)
```

[29]:

	title	type	duration
0	Dick Johnson Is Dead	Movie	90 min
1	Blood & Water	TV Show	2 Seasons
2	Ganglands	TV Show	1 Season
3	Jailbirds New Orleans	TV Show	1 Season
4	Kota Factory	TV Show	2 Seasons

[30]:
```
movies = total_mv_tv[total_mv_tv['type'] == 'Movie'].copy()
```

[31]:
```
movies.head(5)
```

[31]:

	title	type	duration
0	Dick Johnson Is Dead	Movie	90 min
6	My Little Pony: A New Generation	Movie	91 min
7	Sankofa	Movie	125 min
9	The Starling	Movie	104 min
12	Je Suis Karl	Movie	127 min

[32]:
```
#find null values
movies.T.apply(lambda x: x.isnull().sum(),axis =1)
```

[32]:
```
title       0
type        0
duration    3
dtype: int64
```

[33]:
```
#droping the null values
movies = movies.dropna()
```

[34]:
```
#find null values
movies.T.apply(lambda x: x.isnull().sum(),axis =1)
```

[34]:
```
title       0
type        0
duration    0
dtype: int64
```

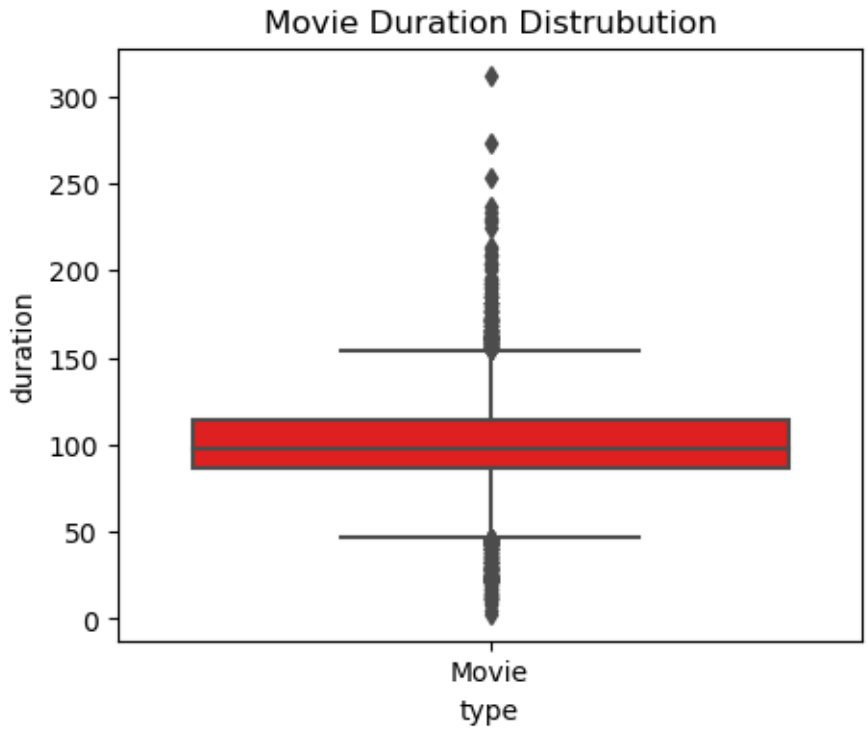
[35]:
```
#extracting duration and coverting to int type
movies['duration'] = movies['duration'].str.extract(r'(\d+)').astype(int)
```

[36]:
```
movies.head()
```

[36]:

	title	type	duration
0	Dick Johnson Is Dead	Movie	90
6	My Little Pony: A New Generation	Movie	91
7	Sankofa	Movie	125
9	The Starling	Movie	104
12	Je Suis Karl	Movie	127

[37]:
```
#box plot for movie duration
plt.figure(figsize = (5,4))
sns.boxplot(data = movies, x= 'type', y ='duration', color = 'red')
plt.title("Movie Duration Distrubution")
plt.show()
```



[38]:
```
tv_shows = total_mv_tv[total_mv_tv['type'] == 'TV Show'].copy()
```

[39]:
```
tv_shows
```

[39]:

	title	type	duration
1	Blood & Water	TV Show	2 Seasons
2	Ganglands	TV Show	1 Season
3	Jailbirds New Orleans	TV Show	1 Season
4	Kota Factory	TV Show	2 Seasons
5	Midnight Mass	TV Show	1 Season
…	…	…	…
8795	Yu-Gi-Oh! Arc-V	TV Show	2 Seasons
8796	Yunus Emre	TV Show	2 Seasons
8797	Zak Storm	TV Show	3 Seasons
8800	Zindagi Gulzar Hai	TV Show	1 Season
8803	Zombie Dumb	TV Show	2 Seasons

[2676 rows x 3 columns]

[40]:
```
#find null values
tv_shows.T.apply(lambda x: x.isnull().sum(),axis =1)
```

[40]:
```
title       0
type        0
duration    0
dtype: int64
```

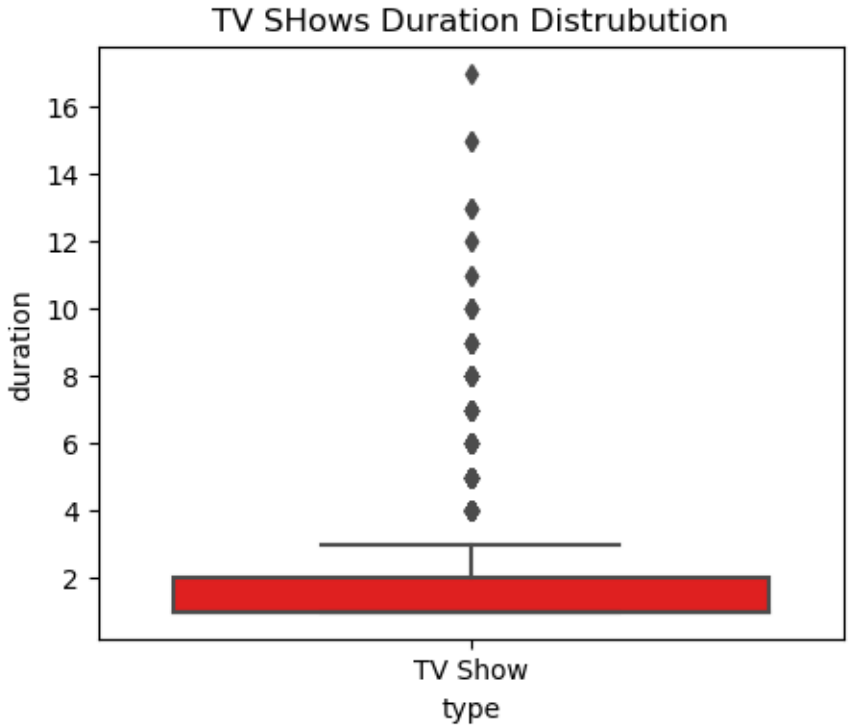
[41]:
```
#extracting duration and coverting to int type
tv_shows['duration'] = tv_shows['duration'].astype(str).str.extract(r'(\d+)').
 ↪astype(int)
```

[42]:
```
tv_shows.head()
```

[42]:

	title	type	duration
1	Blood & Water	TV Show	2
2	Ganglands	TV Show	1
3	Jailbirds New Orleans	TV Show	1
4	Kota Factory	TV Show	2
5	Midnight Mass	TV Show	1

[43]:
```
#box plot for TV show duration
plt.figure(figsize = (5,4))
sns.boxplot(data = tv_shows, x= 'type', y ='duration', color = 'red')
plt.title("TV SHows Duration Distrubution")
plt.show()
```



[ ]:

[44]:
```
#Unnesting the director

dir = pd.DataFrame(data[['show_id', 'title', 'director']])
dir['director'] = dir['director'].str.split(',')
dir = dir.explode('director')
dir.head()
```

[44]:

	show_id	title	director
0	s1	Dick Johnson Is Dead	Kirsten Johnson
1	s2	Blood & Water	Unknown Director
2	s3	Ganglands	Julien Leclercq
3	s4	Jailbirds New Orleans	Unknown Director
4	s5	Kota Factory	Unknown Director

[45]:
```
# dropping the duplicates
dir_unique = dir.drop_duplicates(subset=['director', 'title'])
```

[46]:
```
dir_unique.head(5)
```

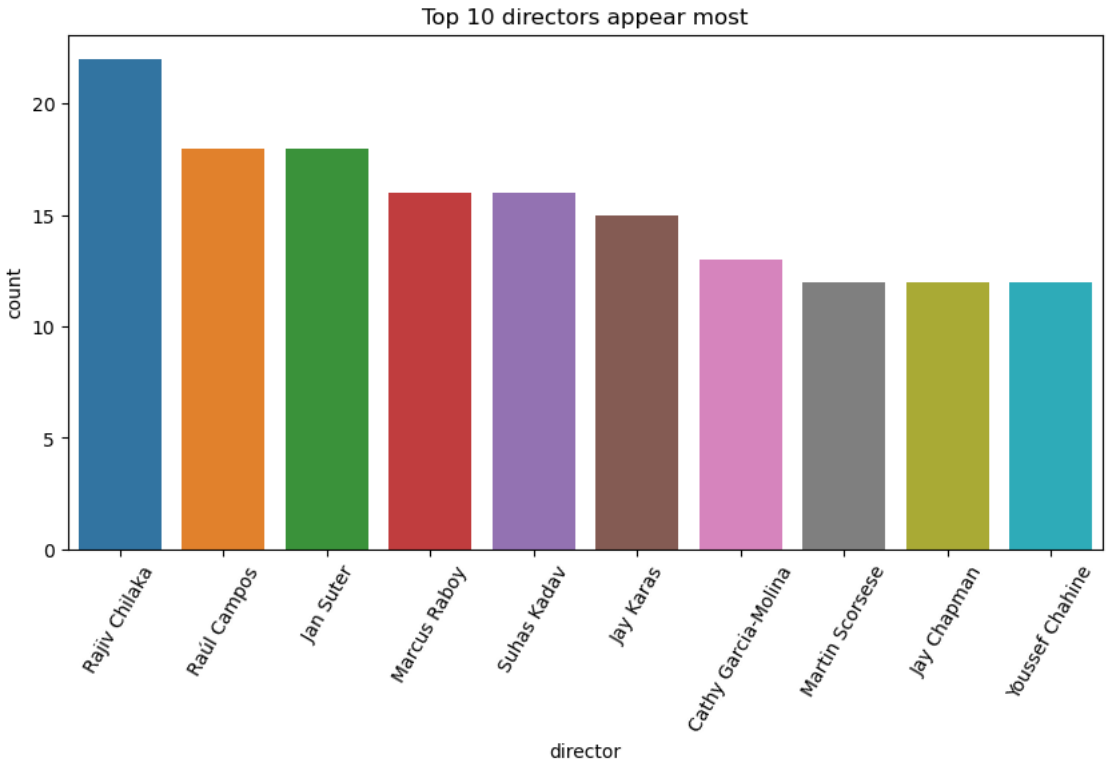
[46]:

	show_id	title	director
0	s1	Dick Johnson Is Dead	Kirsten Johnson
1	s2	Blood & Water	Unknown Director
2	s3	Ganglands	Julien Leclercq
3	s4	Jailbirds New Orleans	Unknown Director
4	s5	Kota Factory	Unknown Director

[47]:
```
#Top 10 popular Directors appered most
dir_unique = dir_unique[dir_unique['director'] != 'Unknown Director']
dir_cout = dir_unique['director'].value_counts(ascending = False).head(10)
dir_cout
```

[47]:
```
director
Rajiv Chilaka          22
Raúl Campos            18
 Jan Suter             18
Marcus Raboy           16
Suhas Kadav            16
Jay Karas              15
Cathy Garcia-Molina    13
Martin Scorsese        12
Jay Chapman            12
Youssef Chahine        12
Name: count, dtype: int64
```

[48]:
```
# top 10 directors
#graphical representation
plt.figure(figsize= (10,5))
sns.countplot(data = dir_unique , x = 'director',order= dir_unique['director'].
 ↪value_counts().index[:10])
plt.xticks(rotation=60)
plt.title('Top 10 directors appear most')
plt.show()
```



[ ]:

[49]:
```
#unnesting the cast
act = pd.DataFrame(data[['show_id', 'title', 'cast']])
act['cast'] = act['cast'].str.split(',')
act = act.explode('cast').reset_index()
act.head()
```

[49]:

	index	show_id	title	cast
0	0	s1	Dick Johnson Is Dead	Unknown Actor
1	1	s2	Blood & Water	Ama Qamata
2	1	s2	Blood & Water	Khosi Ngema
3	1	s2	Blood & Water	Gail Mabalane

```
4      1      s2          Blood & Water    Thabang Molaba
```

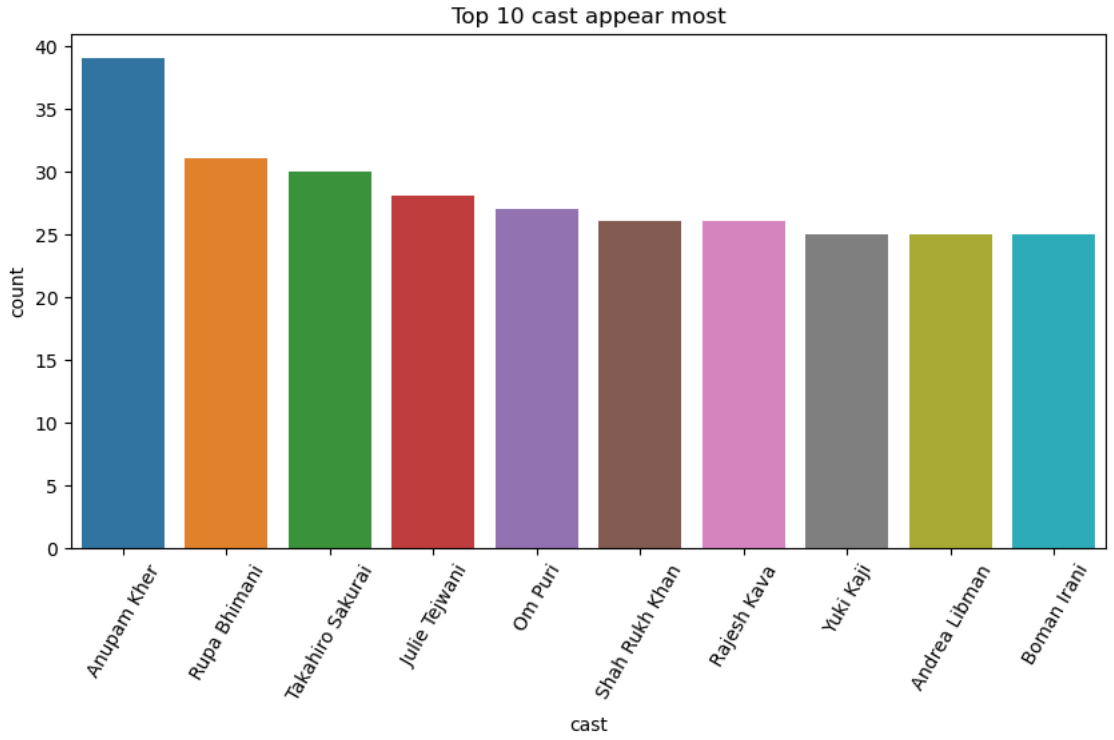
[50]:
```
# dropping the duplicates
act_unique = act.drop_duplicates(subset=['cast', 'title'])
```

[51]:
```
#Top 10 popular cast appered most
act_unique = act_unique[act_unique['cast'] != 'Unknown Actor']
act_cout = act_unique['cast'].value_counts(ascending = False).head(10)
act_cout
```

[51]:
```
cast
 Anupam Kher          39
 Rupa Bhimani         31
 Takahiro Sakurai     30
 Julie Tejwani        28
 Om Puri              27
Shah Rukh Khan        26
 Rajesh Kava          26
 Yuki Kaji            25
 Andrea Libman        25
 Boman Irani          25
Name: count, dtype: int64
```

[ ]:
```

```

[107]:
```
# top 10 cast
#graphical represention
plt.figure(figsize= (10,5))
sns.countplot(data = act_unique , x = 'cast',order= act_unique['cast'].
 ↪value_counts().index[:10])
plt.xticks(rotation=60)
plt.title('Top 10 cast appear most')
plt.show()
```



[ ]:

[219]:
```
# relase week
res = pd.DataFrame(data[['title', 'date_added', 'type', 'rating']])
res
```

[219]:

	title	date_added	type	rating
0	Dick Johnson Is Dead	September 25, 2021	Movie	PG-13
1	Blood & Water	September 24, 2021	TV Show	TV-MA
2	Ganglands	September 24, 2021	TV Show	TV-MA
3	Jailbirds New Orleans	September 24, 2021	TV Show	TV-MA
4	Kota Factory	September 24, 2021	TV Show	TV-MA
…	…	…	…	…
8802	Zodiac	November 20, 2019	Movie	R
8803	Zombie Dumb	July 1, 2019	TV Show	TV-Y7
8804	Zombieland	November 1, 2019	Movie	R
8805	Zoom	January 11, 2020	Movie	PG
8806	Zubaan	March 2, 2019	Movie	TV-14

[8807 rows x 4 columns]

[220]:
```
#changinge the type to datetime
res['date_added'] = res['date_added'].str.strip()
```

[221]:
```
res['date_added'] = pd.to_datetime(res['date_added'], format='%B %d, %Y')
```

[222]:
```
res['week'] = res['date_added'].dt.day_name()
res['month'] = res['date_added'].dt.month_name()
res['year'] = res['date_added'].dt.year
```

[223]:
```
#checking the null

res.T.apply(lambda x: x.isnull().sum(),axis =1)
```

[223]:
```
title          0
date_added    10
type           0
rating         4
week          10
month         10
year          10
dtype: int64
```

[224]:
```
#droping the null values
res = res.dropna()
```

[225]:
```
#checking the null

res.T.apply(lambda x: x.isnull().sum(),axis =1)
```

[225]:
```
title         0
date_added    0
type          0
rating        0
week          0
month         0
year          0
dtype: int64
```

[226]:
```
res
```

[226]:

	title	date_added	type	rating	week	month
0	Dick Johnson Is Dead	2021-09-25	Movie	PG-13	Saturday	September
1	Blood & Water	2021-09-24	TV Show	TV-MA	Friday	September
2	Ganglands	2021-09-24	TV Show	TV-MA	Friday	September
3	Jailbirds New Orleans	2021-09-24	TV Show	TV-MA	Friday	September
4	Kota Factory	2021-09-24	TV Show	TV-MA	Friday	September
…	…	…	…	…	…	…
8802	Zodiac	2019-11-20	Movie	R	Wednesday	November
8803	Zombie Dumb	2019-07-01	TV Show	TV-Y7	Monday	July
8804	Zombieland	2019-11-01	Movie	R	Friday	November

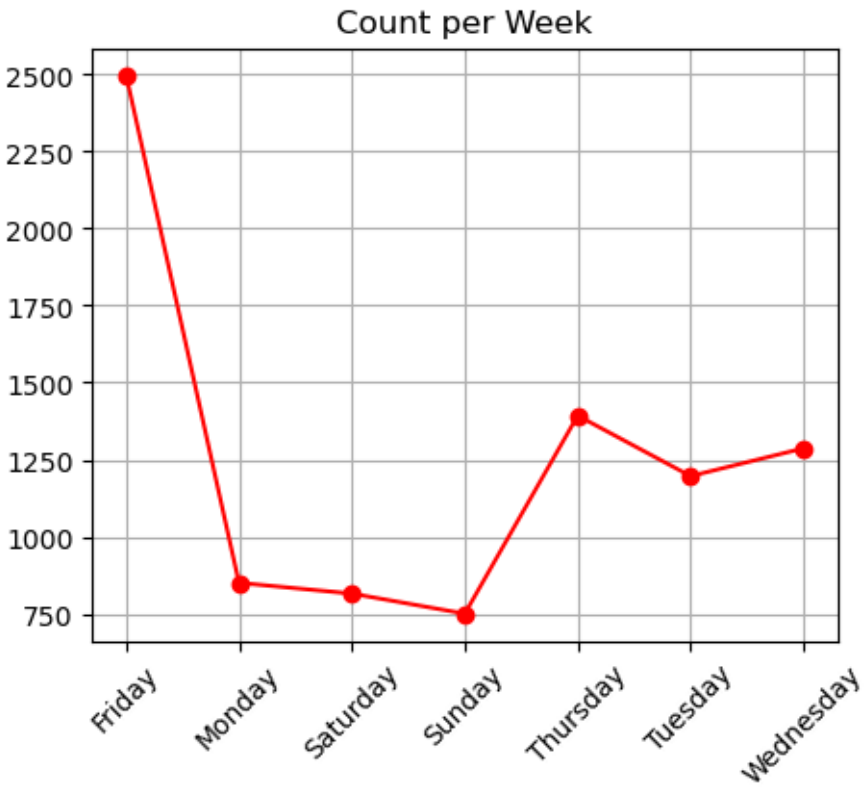
```
8805                  Zoom 2020-01-11    Movie     PG   Saturday    January
8806                Zubaan 2019-03-02    Movie  TV-14   Saturday      March

        year
0     2021.0
1     2021.0
2     2021.0
3     2021.0
4     2021.0
…      …
8802  2019.0
8803  2019.0
8804  2019.0
8805  2020.0
8806  2019.0

[8793 rows x 7 columns]
```

[ ]:

[227]:
```
# week day where movies & TV shows added
grouped_df = res.groupby('week').size().reset_index(name='title_count')
grouped_df
```

[227]:
```
         week  title_count
0      Friday         2498
1      Monday          851
2    Saturday          816
3      Sunday          751
4    Thursday         1393
5     Tuesday         1197
6   Wednesday         1287
```

[228]:
```
plt.figure(figsize=(5, 4))
plt.plot(grouped_df['week'], grouped_df['title_count'], marker='o', color =␣
 ↪'red')
plt.title('Count per Week')
plt.grid(True)
plt.xticks(rotation=45)
plt.show()
```



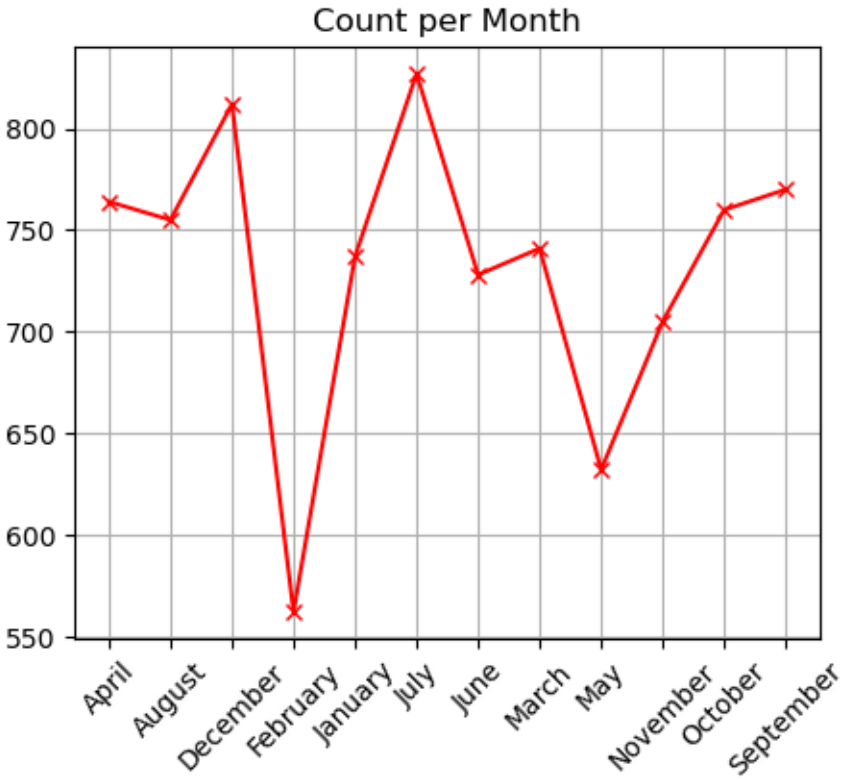
[229]:
```
# month where movies & TV shows added
grouped_m = res.groupby('month').size().reset_index(name='title_count')
grouped_m
```

[229]:

	month	title_count
0	April	764
1	August	755
2	December	812
3	February	562
4	January	737
5	July	827
6	June	728
7	March	741
8	May	632
9	November	705
10	October	760
11	September	770

[230]:
```
plt.figure(figsize=(5, 4))
plt.plot(grouped_m['month'], grouped_m['title_count'], marker='x', color =␣
↪'red')
```

```
plt.title('Count per Month')
plt.grid(True)
plt.xticks(rotation=45)
plt.show()
```



[ ]:

[231]:
```
#Filtering the MOVIES and TVshows
df_movies = res[res['type'] == 'Movie']
df_tv_shows = res[res['type'] == 'TV Show']
```

[232]:
```
#counting movies and TVSHOWS
movies_count = df_movies['year'].value_counts().sort_index()
movies_count
```

[232]:
```
year
2008.0       1
2009.0       2
2010.0       1
2011.0      13
2012.0       3
```

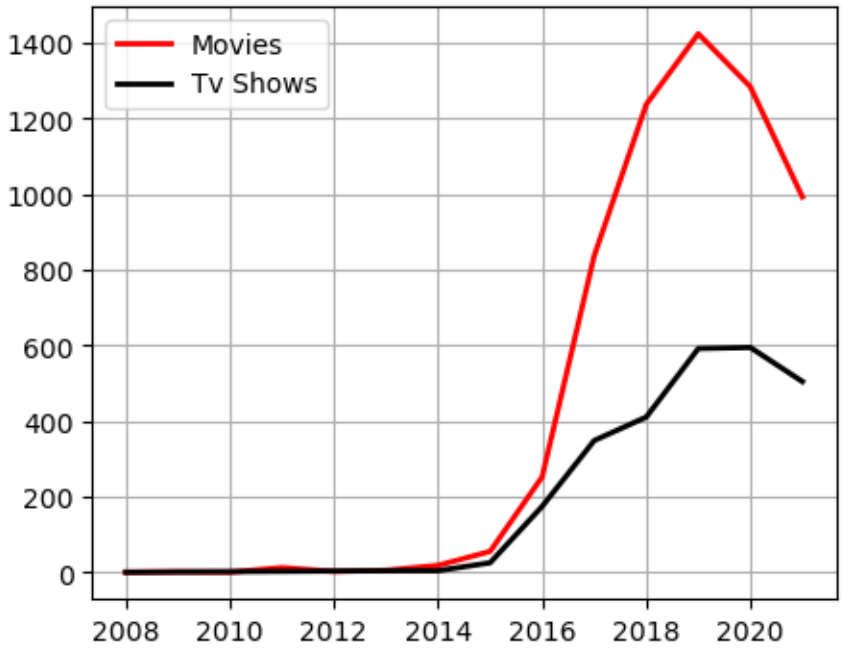
```
2013.0       6
2014.0      19
2015.0      56
2016.0     253
2017.0     837
2018.0    1237
2019.0    1424
2020.0    1284
2021.0     993
Name: count, dtype: int64
```

[233]:
```
tv_show_count =  df_tv_shows['year'].value_counts().sort_index()
tv_show_count
```

[233]:
```
year
2008.0      1
2013.0      5
2014.0      5
2015.0     26
2016.0    175
2017.0    349
2018.0    411
2019.0    592
2020.0    595
2021.0    505
Name: count, dtype: int64
```

[ ]:
```

```

[234]:
```
#ploting graph based on year added
plt.figure(figsize=(5, 4))
plt.plot(movies_count.index, movies_count.values, color='red',
label='Movies', linewidth=2)
plt.plot(tv_show_count.index, tv_show_count.values, color='black',
label='TV Shows', linewidth=2)
plt.grid(True)
plt.legend(labels=['Movies', 'Tv Shows'])
plt.show()
```



[241]:
```
#checking for nulls
res.T.apply(lambda x: x.isnull().sum(),axis =1)
```

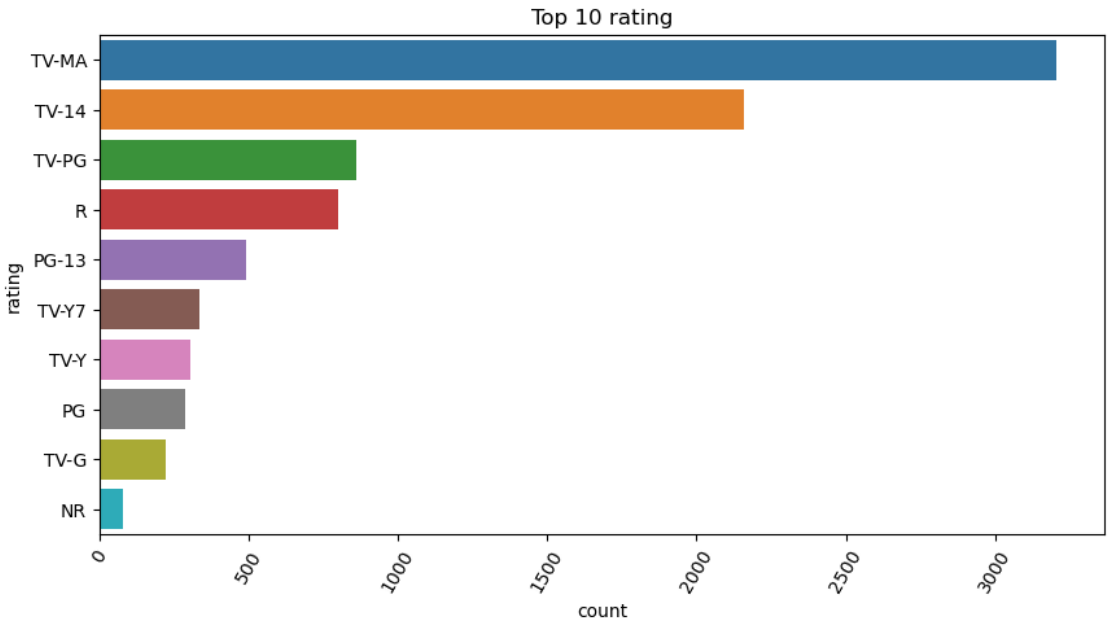
[241]:
```
title         0
date_added    0
type          0
rating        0
week          0
month         0
year          0
dtype: int64
```

[240]:
```
#find the top rating
res['rating'].value_counts()
```

[240]:
```
rating
TV-MA       3205
TV-14       2157
TV-PG        861
R            799
PG-13        490
TV-Y7        333
TV-Y         306
PG           287
TV-G         220
```

```
NR            79
G             41
TV-Y7-FV       6
NC-17          3
UR             3
74 min         1
84 min         1
66 min         1
Name: count, dtype: int64
```

[243]:
```
#graphical representation
plt.figure(figsize= (10,5))
sns.countplot(data = res , y = 'rating',order= res['rating'].value_counts().
 ↪index[:10])
plt.xticks(rotation=60)
plt.title('Top 10 rating')
plt.show()
```



[ ]:

[65]:
```
# Extracting unique genres from the 'listed_in' column
genres = gen['listed_in'].unique()
genres
```

[65]:
```
array(['Documentaries', 'International TV Shows', 'TV Dramas',
       'TV Mysteries', 'Crime TV Shows', 'TV Action & Adventure',
```

```
       'Docuseries', 'Reality TV', 'Romantic TV Shows', 'TV Comedies',
       'TV Horror', 'Children & Family Movies', 'Dramas',
       'Independent Movies', 'International Movies', 'British TV Shows',
       'Comedies', 'Spanish-Language TV Shows', 'Thrillers',
       'Romantic Movies', 'Music & Musicals', 'Horror Movies',
       'Sci-Fi & Fantasy', 'TV Thrillers', "Kids' TV",
       'Action & Adventure', 'TV Sci-Fi & Fantasy', 'Classic Movies',
       'Anime Features', 'Sports Movies', 'Anime Series',
       'Korean TV Shows', 'Science & Nature TV', 'Teen TV Shows',
       'Cult Movies', 'TV Shows', 'Faith & Spirituality', 'LGBTQ Movies',
       'Stand-Up Comedy', 'Movies', 'Stand-Up Comedy & Talk Shows',
       'Classic & Cult TV'], dtype=object)
```

[66]:
```
# Create a new DataFrame to store the genre data
genre_data = pd.DataFrame(index=genres, columns=genres, dtype=float)
genre_data.head()
```

[66]:
```
                        Documentaries  International TV Shows  TV Dramas  \
Documentaries                     NaN                    NaN        NaN
International TV Shows            NaN                    NaN        NaN
TV Dramas                         NaN                    NaN        NaN
TV Mysteries                      NaN                    NaN        NaN
Crime TV Shows                    NaN                    NaN        NaN

                        TV Mysteries  Crime TV Shows  TV Action & Adventure  \
Documentaries                    NaN             NaN                    NaN
International TV Shows           NaN             NaN                    NaN
TV Dramas                        NaN             NaN                    NaN
TV Mysteries                     NaN             NaN                    NaN
Crime TV Shows                   NaN             NaN                    NaN

                        Docuseries  Reality TV  Romantic TV Shows  \
Documentaries                  NaN         NaN                NaN
International TV Shows         NaN         NaN                NaN
TV Dramas                      NaN         NaN                NaN
TV Mysteries                   NaN         NaN                NaN
Crime TV Shows                 NaN         NaN                NaN

                        TV Comedies  …  Science & Nature TV  Teen TV Shows  \
Documentaries                   NaN  …                 NaN            NaN
International TV Shows          NaN  …                 NaN            NaN
TV Dramas                       NaN  …                 NaN            NaN
TV Mysteries                    NaN  …                 NaN            NaN
Crime TV Shows                  NaN  …                 NaN            NaN

                        Cult Movies  TV Shows  Faith & Spirituality  \
Documentaries                   NaN       NaN                   NaN
```

```
International TV Shows          NaN       NaN                    NaN
TV Dramas                       NaN       NaN                    NaN
TV Mysteries                    NaN       NaN                    NaN
Crime TV Shows                  NaN       NaN                    NaN

                        LGBTQ Movies  Stand-Up Comedy  Movies  \
Documentaries                    NaN              NaN     NaN
International TV Shows           NaN              NaN     NaN
TV Dramas                        NaN              NaN     NaN
TV Mysteries                     NaN              NaN     NaN
Crime TV Shows                   NaN              NaN     NaN

                        Stand-Up Comedy & Talk Shows  Classic & Cult TV
Documentaries                                    NaN                NaN
International TV Shows                           NaN                NaN
TV Dramas                                        NaN                NaN
TV Mysteries                                     NaN                NaN
Crime TV Shows                                   NaN                NaN

[5 rows x 42 columns]
```

[67]:
```
# Fill the genre data DataFrame with zeros
genre_data.fillna(0, inplace=True)
```

[68]:
```
genre_data.head()
```

[68]:
```
                        Documentaries  International TV Shows  TV Dramas  \
Documentaries                     0.0                    0.0        0.0
International TV Shows            0.0                    0.0        0.0
TV Dramas                         0.0                    0.0        0.0
TV Mysteries                      0.0                    0.0        0.0
Crime TV Shows                    0.0                    0.0        0.0

                        TV Mysteries  Crime TV Shows  TV Action & Adventure  \
Documentaries                    0.0             0.0                    0.0
International TV Shows           0.0             0.0                    0.0
TV Dramas                        0.0             0.0                    0.0
TV Mysteries                     0.0             0.0                    0.0
Crime TV Shows                   0.0             0.0                    0.0

                        Docuseries  Reality TV  Romantic TV Shows  \
Documentaries                  0.0         0.0                0.0
International TV Shows         0.0         0.0                0.0
TV Dramas                      0.0         0.0                0.0
TV Mysteries                   0.0         0.0                0.0
Crime TV Shows                 0.0         0.0                0.0
```

```
                         TV Comedies  …  Science & Nature TV  Teen TV Shows  \
Documentaries                    0.0  …                  0.0            0.0
International TV Shows           0.0  …                  0.0            0.0
TV Dramas                        0.0  …                  0.0            0.0
TV Mysteries                     0.0  …                  0.0            0.0
Crime TV Shows                   0.0  …                  0.0            0.0

                         Cult Movies  TV Shows  Faith & Spirituality  \
Documentaries                    0.0       0.0                   0.0
International TV Shows           0.0       0.0                   0.0
TV Dramas                        0.0       0.0                   0.0
TV Mysteries                     0.0       0.0                   0.0
Crime TV Shows                   0.0       0.0                   0.0

                         LGBTQ Movies  Stand-Up Comedy  Movies  \
Documentaries                     0.0              0.0     0.0
International TV Shows            0.0              0.0     0.0
TV Dramas                         0.0              0.0     0.0
TV Mysteries                      0.0              0.0     0.0
Crime TV Shows                    0.0              0.0     0.0

                         Stand-Up Comedy & Talk Shows  Classic & Cult TV
Documentaries                                     0.0                0.0
International TV Shows                            0.0                0.0
TV Dramas                                         0.0                0.0
TV Mysteries                                      0.0                0.0
Crime TV Shows                                    0.0                0.0

[5 rows x 42 columns]
```

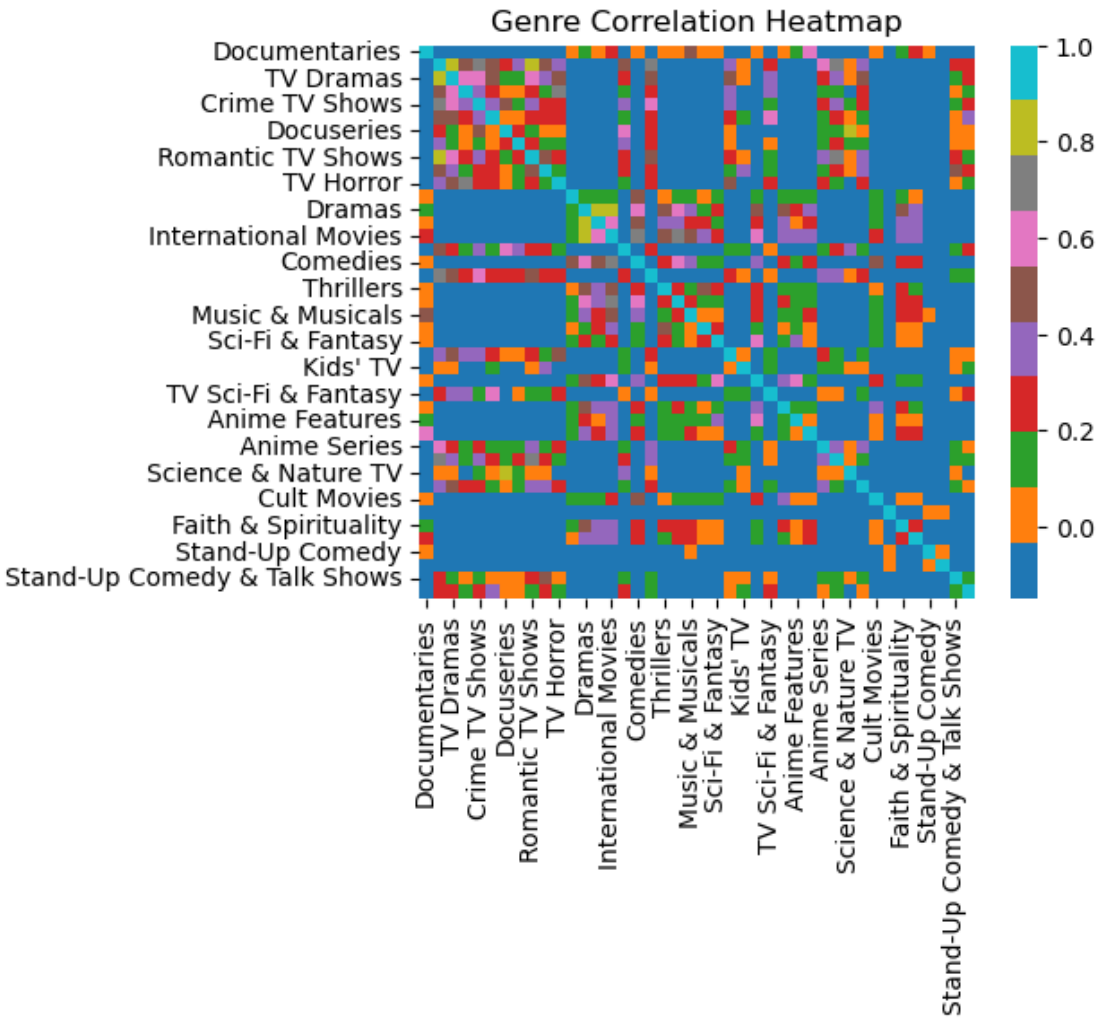
[111]:
```
# Iterate over each row in the original DataFrame and update the genre data␣
↪DataFrame
for _, row in data.iterrows():
    listed_in = row['listed_in'].split(', ')
    for genre1 in listed_in:
        for genre2 in listed_in:
            genre_data.at[genre1, genre2] += 1
```

[70]:
```
# Create a correlation matrix using the genre data
correlation_matrix = genre_data.corr()
correlation_matrix.head()
```

[70]:
```
                         Documentaries  International TV Shows  TV Dramas  \
Documentaries                 1.000000               -0.100995  -0.087762
International TV Shows       -0.100995                1.000000   0.794050
TV Dramas                    -0.087762                0.794050   1.000000
TV Mysteries                 -0.094797                0.434699   0.610177
```

```
Crime TV Shows              -0.092723                0.686674   0.629871

                      TV Mysteries  Crime TV Shows  TV Action & Adventure  \
Documentaries            -0.094797       -0.092723              -0.100440
International TV Shows    0.434699        0.686674               0.435123
TV Dramas                 0.610177        0.629871               0.467544
TV Mysteries              1.000000        0.411654               0.298558
Crime TV Shows            0.411654        1.000000               0.416212

                      Docuseries  Reality TV  Romantic TV Shows  \
Documentaries          -0.079243   -0.072131          -0.086693
International TV Shows  0.292292    0.348752           0.779347
TV Dramas               0.117222    0.138931           0.590389
TV Mysteries            0.069630    0.043834           0.236665
Crime TV Shows          0.435597    0.139077           0.347744

                      TV Comedies  …  Science & Nature TV  Teen TV Shows  \
Documentaries           -0.086350  …            -0.074008      -0.092535
International TV Shows   0.496281  …             0.075633       0.408244
TV Dramas                0.393326  …            -0.019922       0.480982
TV Mysteries             0.128915  …            -0.039656       0.222069
Crime TV Shows           0.229995  …             0.116984       0.204605

                      Cult Movies  TV Shows  Faith & Spirituality  \
Documentaries           -0.031538 -0.039866              0.158488
International TV Shows  -0.146541 -0.061790             -0.130955
TV Dramas               -0.127341 -0.053694             -0.113797
TV Mysteries            -0.137548 -0.057998             -0.122919
Crime TV Shows          -0.134538 -0.056729             -0.120229

                      LGBTQ Movies  Stand-Up Comedy    Movies  \
Documentaries             0.305317        -0.030151 -0.039866
International TV Shows   -0.142630        -0.063438 -0.061790
TV Dramas                -0.123942        -0.055126 -0.053694
TV Mysteries             -0.133877        -0.059545 -0.057998
Crime TV Shows           -0.130948        -0.058242 -0.056729

                      Stand-Up Comedy & Talk Shows  Classic & Cult TV
Documentaries                            -0.072087          -0.101227
International TV Shows                    0.253982           0.223413
TV Dramas                                 0.123506           0.206246
TV Mysteries                              0.020653           0.119857
Crime TV Shows                            0.082847           0.217355

[5 rows x 42 columns]
```

[78]:
```
# Create the heatmap
plt.figure(figsize=(5, 4))
color = sns.color_palette("tab10")
sns.heatmap(correlation_matrix, annot=False, cmap = color)
plt.title('Genre Correlation Heatmap')
plt.xticks(rotation=90)
plt.yticks(rotation=0)
plt.show()
```



[124]:

Insights: • Analysis shows that there are more Movies then TV show • Most of the content was added in Netfilx at July and December • Most of the movies and TV show are produced by USA • Postive Co-relation was observer on genre between tv dramas and crime tv shows, romantic tv show and tv dramas etc. • Movie most of the duration are 100mins • TV shows duration are mostly 1season

Recommendations: • The release in Netflix should focus on the year end and weekends which is to be mainly focussed • Along with Movies we can focus on TV show which help in getting more content to increase the popularity • Top director and cast we can plan some more movies/tv shows in order to increase the popularity • Based on the country and Genre we can added content which help to increase popularity • We have seen most no of international movies genre so need to give priority another genre • There is a linear growth in adding content up to 2019, therefore we need maintain that so attract more people • We can focus on the native Tv shows and Movies as most of the movies and TV show genre are International

[ ]: