In [1]:
```
import pandas as pd
```

In [2]:
```
# import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
import plotly.express as px
warnings.filterwarnings("ignore")
```

In [3]:
```
df = pd.read_csv(r"original_netflix.csv")
```

In [4]:
```
df.head(2)
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA

In [5]:
```
df
```

Out[5]:

	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021
...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006

	show_id	type	title	director	cast	country	date_added	release_year
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015

8807 rows × 12 columns

Upon initial examination of the dataset, we observed that it consists 8807 rows and 12 columns. A preliminary overview indicates that the dataset predominantly represents movies and TV shows, with a notable absence of ratings for some entries. Additionally, NaN values are present in certain columns, suggesting missing or incomplete data.

#Comments on the range of attributes Show_id: A unique identifier assigned to each movie or TV show in the dataset. Type: Identifies whether the entry is a movie or a TV show. Title: The title or name of the movie or TV show. Director: The director(s) associated with the movie. Cast: The actors or individuals involved in the movie or TV show. Country: The country where the movie or TV show was produced. Date_added: The date when the movie or TV show was added to Netflix. Release_year: The actual year when the movie or TV show was released. Rating: The TV rating assigned to the movie or TV show. Duration: The total duration of the movie in minutes or the number of seasons for TV shows. Listed_in: The genre or categories in which the movie or TV show is listed. Description: A summary or brief description of the movie or TV show.

In [6]:
```
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
```

In [7]:
```
df.shape
```

Out[7]: (8807, 12)

In [8]:
```
df.describe()
```

Out[8]:

	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

In [9]:
```
df.describe(include='object').T
```

Out[9]:

	count	unique	top	freq
show_id	8807	8807	s1	1
type	8807	2	Movie	6131
title	8807	8807	Dick Johnson Is Dead	1
director	6173	4528	Rajiv Chilaka	19
cast	7982	7692	David Attenborough	19
country	7976	748	United States	2818
date_added	8797	1767	January 1, 2020	109
rating	8803	17	TV-MA	3207
duration	8804	220	1 Season	1793
listed_in	8807	514	Dramas, International Movies	362
description	8807	8775	Paranormal activity at a lush, abandoned prope...	4

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

In [12]:
```
df['type'].value_counts()
```

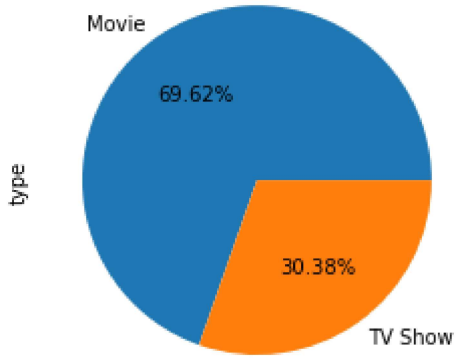
Out[12]:
```
Movie      6131
TV Show    2676
Name: type, dtype: int64
```

In [13]:
```
df['type'].value_counts(normalize=True)*100
```

Out[13]:
```
Movie      69.615079
TV Show    30.384921
Name: type, dtype: float64
```

In [14]:
```
df['type'].value_counts().plot(
kind='pie',autopct='%.2f%%')
```

Out[14]:
```
<AxesSubplot:ylabel='type'>
```



Based on the provided chart, it is evident that the majority of the audience, accounting for 69.62%, prefers movies compared to TV shows

In [15]:
```
Rating = df.groupby("rating").rating.count().sort_values(ascending=False)
Rating
```

Out[15]:
```
rating
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
UR             3
NC-17          3
74 min         1
84 min         1
66 min         1
Name: rating, dtype: int64
```

In [16]:
```
fig = px.histogram(df, y='rating', color='type')
fig.update_yaxes(showgrid=False, categoryorder='total ascending',
ticksuffix=' ', showline=False)
fig.show()
```

The audience prefers TV-MA and TV-14 shows more and the least preferred rating shows are Nc□17. Most of the content watched by the audience is for a mature audience. The second largest type of rating watched by the audience is TV-14 which is inappropriate for children younger than age 14. The conclusion is drawn here is most of the audience is of mature age. chart for comparison between TV Shows and Movies. This chart tells us that Netflix's audience prefers to watch movies rather than TV Shows. The Highest rating is given to Movies and TV Show having a TV-MA

In [17]:
```
movie_count = df[df["type"]=="Movie"]
movie_release = movie_count.groupby("release_year").size()
movie_release.tail()
```

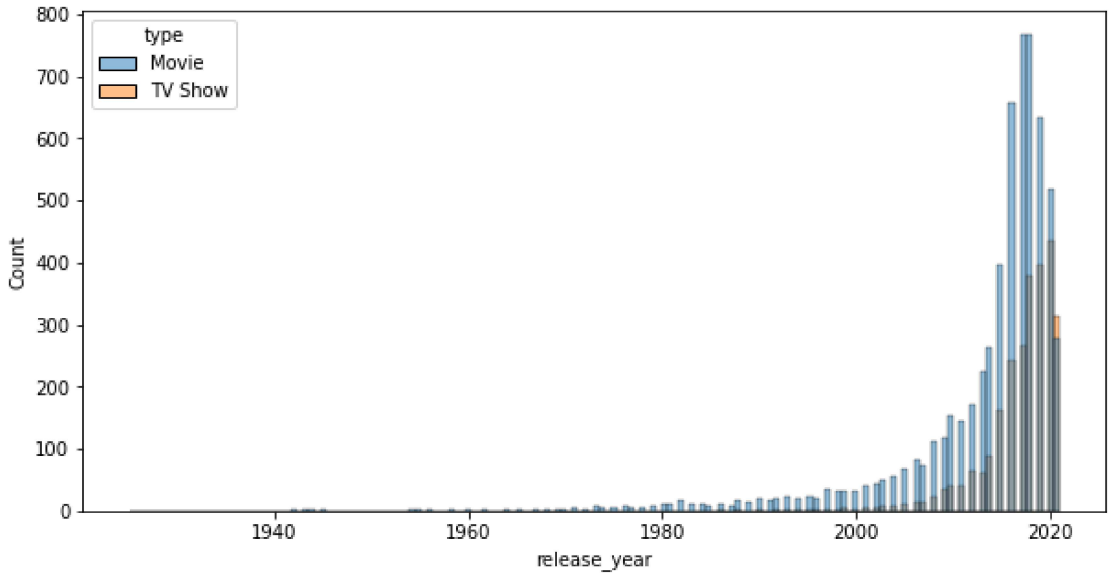
Out[17]:
```
release_year
2017    767
2018    767
2019    633
2020    517
2021    277
dtype: int64
```

In [18]:
```
show_count = df[df["type"]=="TV Show"]
show_release = show_count.groupby("release_year").size()
show_release.tail()
```

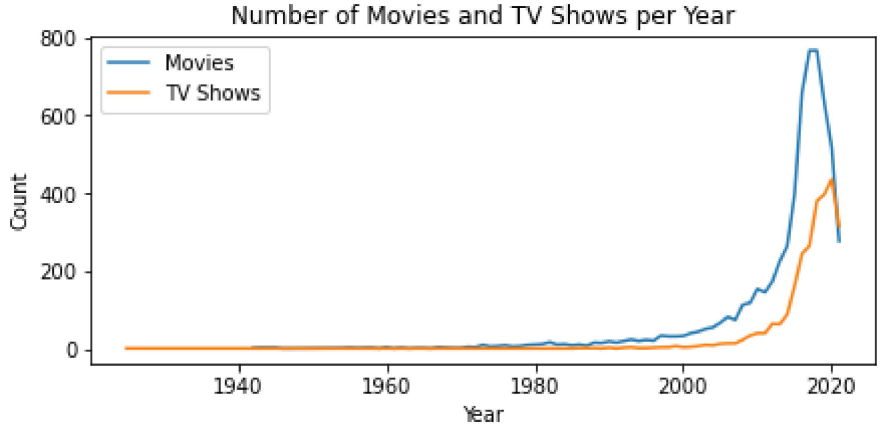
Out[18]:
```
release_year
2017    265
2018    380
2019    397
2020    436
2021    315
dtype: int64
```

In [19]:
```
plt.figure(figsize=(10,5))
sns.histplot(x='release_year',hue='type',data=df)
```

Out[19]: <AxesSubplot:xlabel='release_year', ylabel='Count'>



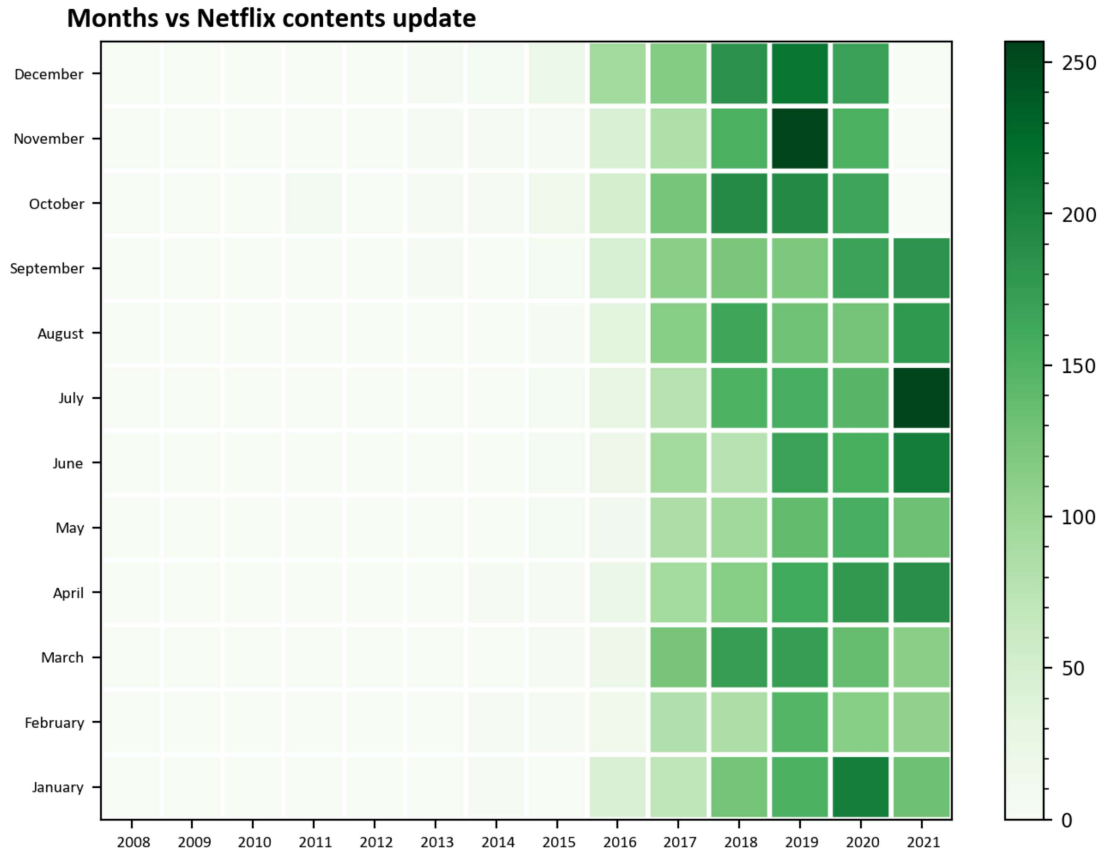
In [20]:
```
df_movie = df[df["type"] == "Movie"]
movie_by_year=df_movie.groupby("release_year").size()
df_tvshow = df[df["type"] == "TV Show"]
tvshow_by_year=df_tvshow.groupby("release_year").size()
plt.figure(figsize=(7,3))
plt.plot(movie_by_year.index, movie_by_year.values, label='Movies')
plt.plot(tvshow_by_year.index, tvshow_by_year.values, label='TV Shows')
plt.xlabel('Year')
plt.ylabel('Count')
plt.title('Number of Movies and TV Shows per Year')
plt.legend()
plt.show()
```



As shown, 2020 was a high water mark with the greatest number of shows 436 released that year. The steep decline from 2020 could be due to the fact that competitors began introducing their own streamong services. For instance, all Four Marvel shows were not continued with

Netflix, with the rights reverting to Marvel/Disney. same shows for Movies aswell 2017-2018 was a high water mark with the greatest number of shows 767 released that year, but after

In [21]:
```
netflix_date = df[['date_added']].dropna()
netflix_date['year']= netflix_date['date_added'].apply(lambda x:x.split('
netflix_date['month']= netflix_date['date_added'].apply(lambda x:x.lstrip(
month_order = ['January', 'February', 'March', 'April', 'May', 'June', 'Ju
new_df = netflix_date.groupby('year')['month'].value_counts().unstack().fi
plt.figure(figsize=(8,6), dpi=200)
plt.pcolor(new_df, cmap ='Greens',edgecolors='white',linewidths=2)
plt.xticks(np.arange(0.5, len(new_df.columns),1), new_df.columns, fontsize
plt.yticks(np.arange(0.5, len(new_df.index),1), new_df.index, fontsize=7,f
plt.title('Months vs Netflix contents update',fontsize=12,fontfamily='cali
cbar=plt.colorbar()
cbar.ax.tick_params(labelsize=8)
cbar.ax.minorticks_on()
plt.show()
```



If a producer wants to release a show which month is the best month to release it. The best month to release content so the producer can gain much revenue. Most of the holidays came in December month so to releases a Movie or TV show in December is the best way to earn a lot of profit as the whole family will be spending time with each other and watching shows

In [22]:
```
df.country.value_counts().head(10) # top 10 countries
```

Out[22]:
```
United States     2818
India              972
United Kingdom     419
Japan              245
South Korea        199
Canada             181
Spain              145
France             124
Mexico             110
Egypt              106
Name: country, dtype: int64
```

In [23]:
```
top_10_countries = df['country'].value_counts().head(10)
df_top_10 = df[df['country'].isin(top_10_countries.index)]
fig = px.histogram(df_top_10, y='country', color='type')
fig.update_yaxes(showgrid=False, categoryorder='total ascending',
ticksuffix=' ', showline=False)
fig.update_layout(
title={
'text': 'Top 10 Countries - Country Distribution',
'x': 0.5,
'y': 0.95,
'xanchor': 'center',
'yanchor': 'top'
}
)
fig.show()
```

In the given chart, Upon closer examination, it is evident that the USA has the highest number of movies. Following closely is India, which holds the second position. However, when considering TV shows, the United Kingdom (UK) takes the second position after the USA, with a significant representation.

In [24]:
```
cast_df = df[['type','cast']]
actor = cast_df.set_index('type').cast.str.split(', ', expand=True).stack(
d3 = actor.reset_index()
d3.rename({0: "Actor"}, axis=1, inplace=True)
```

In [25]:
```
d3[d3['type']=='Movie'].Actor.value_counts().head(10)
```

Out[25]:
```
Anupam Kher          42
Shah Rukh Khan       35
Naseeruddin Shah     32
Akshay Kumar         30
Om Puri              30
Amitabh Bachchan     28
Julie Tejwani        28
Paresh Rawal         28
Rupa Bhimani         27
Boman Irani          27
Name: Actor, dtype: int64
```

In [26]:
```
top_15_cast = d3['Actor'].value_counts().head(15)
df_top_15 = d3[d3['Actor'].isin(top_15_cast.index)]
fig = px.histogram(df_top_15, y='Actor', color='type')
fig.update_yaxes(showgrid=False, categoryorder='total ascending',
ticksuffix=' ', showline=False)
fig.update_layout(
title={
'text': 'Top 15 Actors - Actor Distribution',
'x': 0.5,
'y': 0.95,
'xanchor': 'center',
'yanchor': 'top'
}
)
fig.show()
```

Based on the provided plot, it is apparent that Takahiro Sakurai emerges as the leading actor in TV series, securing the topmost position. Following closely behind in the second position is Yuki Kaji. Anupam Kher emerges as the leading actor in Movies, securing the topmost position. Following closely behind in the second position is Shah Rukh Khan

In [27]:
```
dir_df = df[['type','director']]
actor = dir_df.set_index('type').director.str.split(', ', expand=True).sta
d4 = actor.reset_index()
d4.rename({0: "Director"}, axis=1, inplace=True)
```

In [28]:
```
d4[d4['type']=='Movie'].Director.value_counts().head(10)
```

Out[28]:
```
Rajiv Chilaka          22
Jan Suter              21
Raúl Campos            19
Suhas Kadav            16
Marcus Raboy           15
Jay Karas              15
Cathy Garcia-Molina    13
Martin Scorsese        12
Youssef Chahine        12
Jay Chapman            12
Name: Director, dtype: int64
```

In [29]:
```
d4[d4['type']=='TV Show'].Director.value_counts().head(10)
```

Out[29]:
```
Alastair Fothergill      3
Ken Burns                3
Jung-ah Im               2
Gautham Vasudev Menon    2
Iginio Straffi           2
Hsu Fu-chun              2
Stan Lathan              2
Joe Berlinger            2
Shin Won-ho              2
Lynn Novick              2
Name: Director, dtype: int64
```

In [30]:
```
top_20_Dir = d4['Director'].value_counts().head(20)
df_top_20 = d4[d4['Director'].isin(top_20_Dir.index)]
fig = px.histogram(df_top_20, y='Director', color='type')
fig.update_yaxes(showgrid=False, categoryorder='total ascending',
ticksuffix=' ', showline=False)
fig.update_layout(
title={
'text': 'Top 20 Directors - Director Distribution',
'x': 0.5,
'y': 0.95,
'xanchor': 'center',
'yanchor': 'top'
}
)
fig.show()
```

Based on the plot, it is apparent that Alastair Fothergill & Ken Burns are the leading diector in TV series, securing the topmost position. Rajiv Chilaka emerges as the leading diector in Movies, securing the topmost position. Following closely behind in the second position is Jan Suter.

In [31]:
```
gen = df[['type','listed_in']]
Genre = gen.set_index('type').listed_in.str.split(', ', expand=True).stack
d2 = Genre.reset_index()
d2.rename({0: "Genre"}, axis=1, inplace=True)
```

In [32]:
```
d2[d2['type']=='Movie'].Genre.value_counts().head(10)
```

Out[32]:
```
International Movies        2752
Dramas                      2427
Comedies                    1674
Documentaries                869
Action & Adventure           859
Independent Movies           756
Children & Family Movies     641
Romantic Movies              616
Thrillers                    577
Music & Musicals             375
Name: Genre, dtype: int64
```

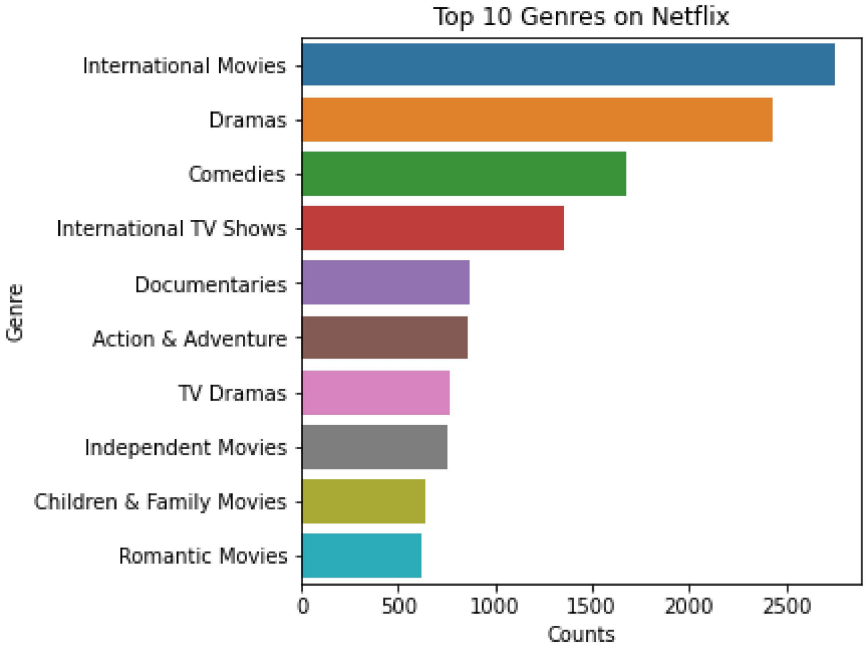
In [33]:
```
d2[d2['type']=='TV Show'].Genre.value_counts().head(10)
```

Out[33]:
```
International TV Shows    1351
TV Dramas                  763
TV Comedies                581
Crime TV Shows             470
Kids' TV                   451
Docuseries                 395
Romantic TV Shows          370
Reality TV                 255
British TV Shows           253
Anime Series               176
Name: Genre, dtype: int64
```

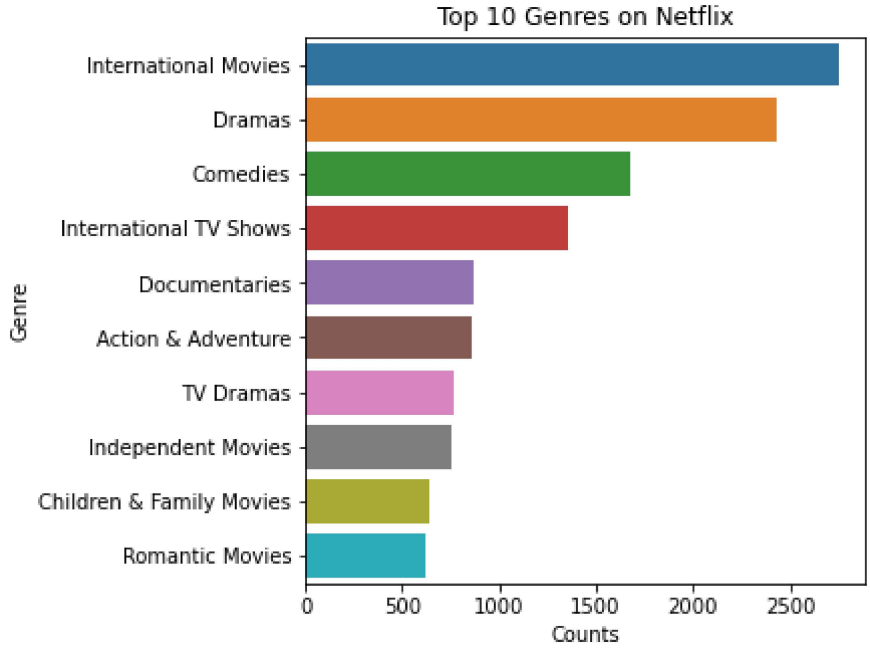
In [34]:
```
top_10_genre = d2['Genre'].value_counts().head(10)
df_top_10 = d2[d2['Genre'].isin(top_10_genre.index)]
fig = px.histogram(df_top_10, y='Genre', color='type')
fig.update_yaxes(showgrid=False, categoryorder='total ascending',
ticksuffix=' ', showline=False)
fig.update_layout(
title={
'text': 'Top 10 Genres - Genre Distribution',
'x': 0.5,
'y': 0.95,
'xanchor': 'center',
'yanchor': 'top'
}
)
fig.show()
```

Based on the provided plot, it is apparent that International Movies & International TV Shows are the top gener in movie & tv show.

In [35]:
```
Genre = df.set_index('title').listed_in.str.split(', ', expand=True).stack
plt.figure(figsize=(5,5))
g = sns.countplot(y = Genre, order=Genre.value_counts().index[:10])
plt.title('Top 10 Genres on Netflix')
plt.xlabel('Counts')
plt.ylabel('Genre')
plt.show()
```



In [36]:
```
Genre = df.set_index('title').listed_in.str.split(', ', expand=True).stack
plt.figure(figsize=(5,5))
g = sns.countplot(y = Genre, order=Genre.value_counts().index[:10])
plt.title('Top 10 Genres on Netflix')
plt.xlabel('Counts')
plt.ylabel('Genre')
plt.show()
```



In [37]:
```
 # droping rows for small percentages of null
df.dropna(subset=['rating','duration'],axis=0,inplace=True)
df.dropna(subset=['date_added'],axis=0,inplace=True)
```

In [38]:
```
df.shape
```

Out[38]: (8790, 12)

In [39]:
```
round(df.isnull().sum()/df.shape[0]*100,2).sort_values(ascending=False)
```

Out[39]:
```
director        29.82
country          9.43
cast             9.39
show_id          0.00
type             0.00
title            0.00
date_added       0.00
release_year     0.00
rating           0.00
duration         0.00
listed_in        0.00
description      0.00
dtype: float64
```

In [40]:
```
#Data Wrangling
df['country']=df['country'].fillna(df['country'].mode()[0])
df['cast'].replace(np.nan, 'No Data', inplace=True)
df['director'].replace(np.nan, 'No Data', inplace=True)
df.dropna(inplace=True)
#drop duplicates
df.drop_duplicates(inplace=True)
```

In [41]:
```
df.isnull().sum()
```

Out[41]:
```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
dtype: int64
```

In [42]:
```
show_df = df.loc[(df['type']=='TV Show')]
show_df.head(2)
```

Out[42]:

	show_id	type	title	director	cast	country	date_added	release_year	ratin
1	s2	TV Show	Blood & Water	No Data	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV M
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	September 24, 2021	2021	TV M

In [43]:
```
show_df.duration=show_df.duration.apply(lambda x:x.replace(" Season","")i
show_df.head(2) # still here 1 data with 's'
```

Out[43]:

	show_id	type	title	director	cast	country	date_added	release_year	ratin
1	s2	TV Show	Blood & Water	No Data	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV M
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	September 24, 2021	2021	TV M

In [47]:
```
show_df.duration=show_df.duration.apply(lambda x:x.replace("s","")if 's' i
show_df.head(2)
```

Out[47]:

	show_id	type	title	director	cast	country	date_added	release_year	ratin
1	s2	TV Show	Blood & Water	No Data	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV M
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	September 24, 2021	2021	TV M

In [48]:
```
show_df.loc[:,['duration']]=show_df.loc[:,['duration']].apply(lambda x: x.
```

In [49]:
```
show_df.describe()
```

Out[49]:

	release_year	duration
count	2664.000000	2664.000000
mean	2016.627628	1.751877
std	5.735194	1.550622
min	1925.000000	1.000000
25%	2016.000000	1.000000
50%	2018.000000	1.000000
75%	2020.000000	2.000000
max	2021.000000	17.000000

In [50]:
```
longest_shows=show_df.loc[(show_df['duration']>=17)]
longest_shows.head()
```

Out[50]:

	show_id	type	title	director	cast	country	date_added	release_year	ratin
548	s549	TV Show	Grey's Anatomy	No Data	Ellen Pompeo, Sandra Oh, Katherine Heigl, Just...	United States	July 3, 2021	2020	TV-1

In [52]:
```
# shorted tv show
longest_shows=show_df.loc[(show_df['duration']==1)]
longest_shows.head()
```

Out[52]:

	show_id	type	title	director	cast	country	date_added	release_year	ra
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	September 24, 2021	2021	
3	s4	TV Show	Jailbirds New Orleans	No Data	No Data	United States	September 24, 2021	2021	
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	United States	September 24, 2021	2021	
10	s11	TV Show	Vendetta: Truth, Lies and The Mafia	No Data	No Data	United States	September 24, 2021	2021	
11	s12	TV Show	Bangkok Breaking	Kongkiat Komesiri	Sukollawat Kanarot, Sushar Manaying, Pavarit M...	United States	September 23, 2021	2021	

Based on the available data, it has been determined that the shortest TV show in the dataset consists of only 1 season, while the longest TV show spans an impressive 17 seasons.

In [53]:
```
movie_df = df.loc[(df['type']=='Movie')]
movie_df.head(2)
```

Out[53]:

	show_id	type	title	director	cast	country	date_added	release_year	ratin
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No Data	United States	September 25, 2021	2020	PG 1
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	United States	September 24, 2021	2021	P(

In [54]:
```
movie_df.duration=movie_df.duration.apply(lambda x:x.replace(" min","")if
movie_df.head(2)
```

Out[54]:

	show_id	type	title	director	cast	country	date_added	release_year	ratin
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No Data	United States	September 25, 2021	2020	PG 1
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	United States	September 24, 2021	2021	P(

In [55]:
```
movie_df.loc[:,['duration']]=movie_df.loc[:,['duration']].apply(lambda x:
```

In [56]:
```
movie_df.describe()
```

Out[56]:

	release_year	duration
count	6126.000000	6126.000000
mean	2013.120144	99.584884
std	9.681723	28.283225
min	1942.000000	3.000000
25%	2012.000000	87.000000
50%	2016.000000	98.000000
75%	2018.000000	114.000000
max	2021.000000	312.000000

In [57]:
```
# short movie
shortest_movie=movie_df.loc[(movie_df['duration']==np.min(movie_df.duratic
shortest_movie
```

Out[57]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	d
3777	s3778	Movie	Silent	Limbert Fabian, Brandon Oldenburg	No Data	United States	June 4, 2019	2014	TV-Y	

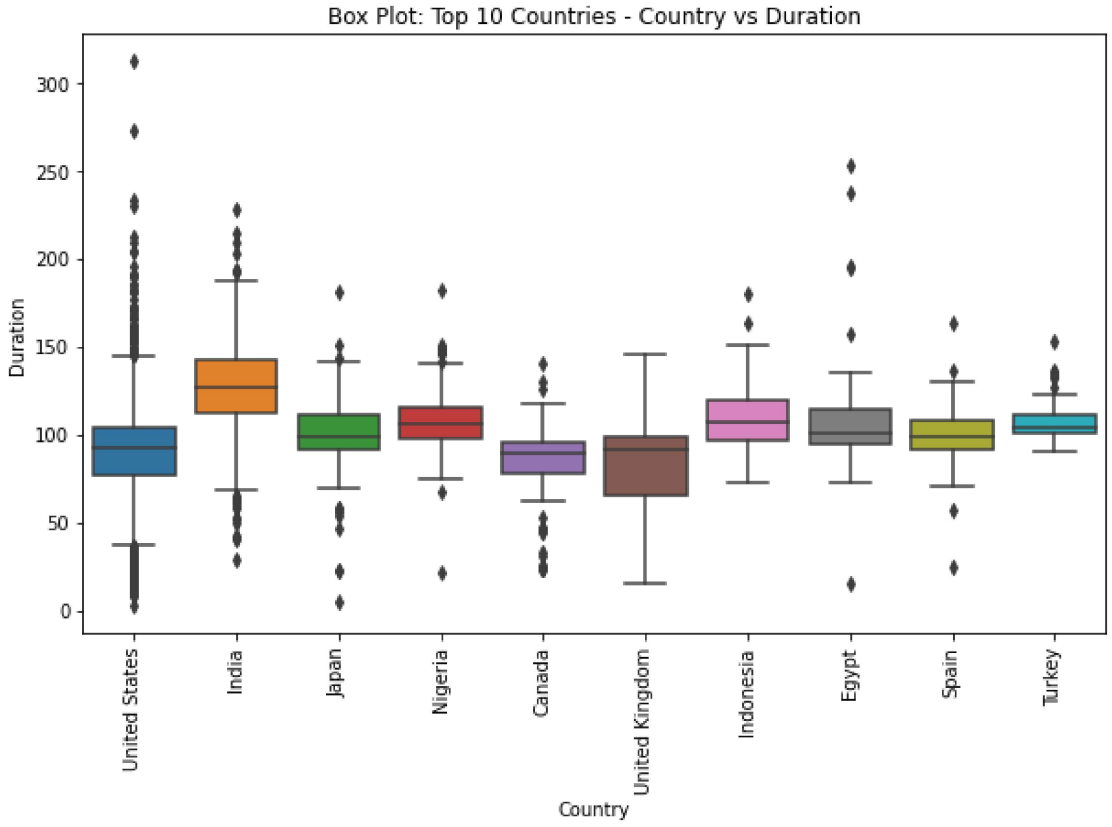
In [58]:
```
# Longest movie
longest_movie=movie_df.loc[(movie_df['duration']==np.max(movie_df.duratior
longest_movie
```

Out[58]:

	show_id	type	title	director	cast	country	date_added	release_yea
4253	s4254	Movie	Black Mirror: Bandersnatch	No Data	Fionn Whitehead, Will Poulter, Craig Parkinson...	United States	December 28, 2018	201

After analyzing the provided data, it has been ascertained that the shortest movie in the dataset has a duration of 3 minutes, whereas the longest movie stretches out for an impressive 312 minutes.

In [62]:
```
top_10_countries = movie_df['country'].value_counts().head(10).index
# Filter the data for the top 10 countries
filtered_data = movie_df[movie_df['country'].isin(top_10_countries)]
plt.figure(figsize=(10, 6))
sns.boxplot(x='country', y='duration', data=filtered_data)
plt.title('Box Plot: Top 10 Countries - Country vs Duration')
plt.xlabel('Country')
plt.ylabel('Duration')
plt.xticks(rotation=90)
plt.show()
```

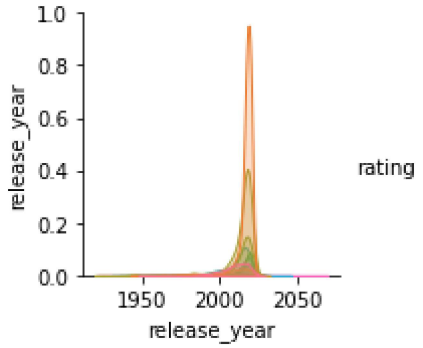


There are several movies from the USA with relatively longer durations compared to other countries. These outliers are represented as individual data points beyond the whiskers of the box plot, indicating extreme values

In [64]:
```
df1 = df.copy()
df1["date_added"] = pd.to_datetime(df1['date_added']) # add 2 columns of y
df1['year_added'] = df1['date_added'].dt.year
df1['month_added']=df1['date_added'].dt.month
```

In [65]:
```
#Drop rows with missing values and drop duplicates:
df1.dropna(inplace=True)
df1.drop_duplicates(inplace=True)
plt.figure(figsize=(10,6))
#visualize pairplot of df:
sns.pairplot(df, hue='rating')
#plot the graph:
plt.show()
```

```
<Figure size 720x432 with 0 Axes>
```



with the given given plot its shows that in 2018 released content have TV-MA rating, same is for year and month date added on Netflix

In [67]:
```
df1.head(2)
```

Out[67]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No Data	United States	2021-09-25	2020	PG-13
1	s2	TV Show	Blood & Water	No Data	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA

In [69]:
```
filtered_country = df1[df1.country != 'No Data'].set_index('title').countr
country_df = filtered_country.reset_index()
country_df.rename({0:"Country"},axis=1,inplace=True)
```

In [74]:
```
filtered_genre = df1.set_index('title').listed_in.str.split(', ', expand=T
genre_df = filtered_genre.reset_index()
genre_df.rename({0:"genre"},axis=1,inplace=True)
```

In [75]:
```
df_county=df1.reset_index().merge(country_df, on='title', how='inner')
```

In [76]:
```
df_new=df_county.reset_index().merge(genre_df, on='title', how='inner')
```

In [77]:
```
df_new.head()
```

Out[77]:

	level_0	index	show_id	type	title	director	cast	country	date_added	re
0	0	0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No Data	United States	2021-09-25	
1	1	1	s2	TV Show	Blood & Water	No Data	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	
2	1	1	s2	TV Show	Blood & Water	No Data	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	
3	1	1	s2	TV Show	Blood & Water	No Data	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	
4	2	2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	2021-09-24	

In [78]:
```
df_new[df_new['Country']=='United States'].head()
```

Out[78]:

	level_0	index	show_id	type	title	director	cast	country	date_added	rele
0	0	0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	No Data	United States	2021-09-25	
4	2	2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	2021-09-24	
5	2	2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	2021-09-24	
6	2	2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	United States	2021-09-24	
7	3	3	s4	TV Show	Jailbirds New Orleans	No Data	No Data	United States	2021-09-24	

In [79]:
```
top_10_country = df_new['Country'].value_counts().head(10)
df_top_10 = df_new[df_new['Country'].isin(top_10_country.index)]
fig = px.histogram(df_top_10, y='Country', color='genre')
fig.update_yaxes(showgrid=False, categoryorder='total ascending',
ticksuffix=' ', showline=False)
fig.update_layout(
title={
'text': 'Top 10 country - Genre Distribution',
'x': 0.5,
'y': 0.95,
'xanchor': 'center',
'yanchor': 'top'
}
)
fig.show()
```

From the provided plot, it is evident that in the United States, the most frequently watched genres are documentaries, dramas, and comedies. On the other hand, in India, the preferred genres include 27 international movies, dramas, and comedies.

In [ ]: