DATA ANALYSIS AND VISUALIZATION PROJECT

Netflix Data Analysis and Visualization

Submitted by:	*Submitted to:*
Ashish Sah (Team Leader) – 3436	Dr. Manju
Dhruv Singh – 3438	
Durgesh Kumar – 3446	
Ankit – 3456	
Pankaj – 3463	
Tanmay Chowdhary – 3483	

Department of Computer Science

Hansraj College, University Of Delhi

NETFLIX DATA ANALYSIS AND VISUALIZATION

Domain:

OTT Platforms/Entertainment

Objective:

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?

Source of data:

Link: Netflix Movies and TV Shows | Kaggle

Size of Dataset:

6234 rows , 12 columns

Description:

Netflix is a popular service that people across the world use for entertainment.

Attributes information:

1. *Show_id* - Unique ID for every Movie / TV Show
2. *Type* – Categorizes the content as - A Movie or TV Show
3. *Title* - Title of the Movie / TV Show
4. *Director* – Name of the person(s) who directed the movie
5. *Cast* – Actors/Actresses involved in the movie/show
6. *Country* - Country where the movie/show was produced
7. *Date_added* - Date it was added on Netflix
8. *Release_year* - Actual Release year of the movie/show
9. *Rating*: Rating of the movie/show
10. *Duration* - Total Duration in minutes or number of seasons
11. *Listed_in* – Genre of the movie/show
12. *Description* - The summary description of the movie/show

Team Members:

1. Ashish Sah (3436)

2. Dhruv Singh (3438)

3. Durgesh Kumar (3446)

4. Ankit (3456)

5. Pankaj (3463)

6. Tanmay Chowdhary (3483)

NDA

December 2, 2023

1 Netflix Data Analysis and Visualisation

Team

- Ashish Sah (3436) (Team Leader)
- Dhruv Singh (3438)
- Durgesh Kumar (3446)
- Ankit (3456)
- Pankaj (3463)
- Tanmay Chowdhary (3483)

Objective 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.

1.1 Introduction

Netflix is a streaming platform with thousands of movies and TV shows. This project dives into the depths of Netflix viewership, aiming to uncover consumption trends, genre preferences, and the influence of popular releases. This project uses the dataset imported from Kaggle. The dataset is available at: https://www.kaggle.com/datasets/shivamb/netflix-shows

1.2 1. Import Libraries

[2]:

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

1.3 2. Load Datasets

[3]:

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

[4]:

```
netflix.head()
```

[4]:

```
  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…
```

```
[5]: print('Shape of Netflix dataset : ' , netflix.shape);
```

```
Shape of Netflix dataset :  (8807, 12)
```

1.4 3. Dataset description

Description of columns in the files:

- Show_id - Unique ID for every Movie / TV Show
- Type – Categorizes the content as - A Movie or TV Show
- Title - Title of the Movie / TV Show
- Director – Name of the person(s) who directed the movie
- Cast – Actors/Actresses involved in the movie/show
- Country - Country where the movie/show was produced
- Date_added - Date it was added on Netflix
- Release_year - Actual Release year of the movie/show
- Rating: Rating of the movie/show
- Duration - Total Duration in minutes or number of seasons

- Listed_in – Genre of the movie/show
- Description - The summary description of the movie/show

1.5 4. EDA of Netflix Dataset

[6]:

```
#Info about the Netflix dataset
netflix.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
```

Conclusion: * Dataset is having 11 Object Data type columns and 1 Integer dataType Column. * Some columns contain missing values and we have to handle them. * We can see date_added column is of object type which should be of datetime. * We can see type and rating columns are also of object type which should be of category.

[7]:

```
#Description of numerical data in Netflix dataset
netflix.describe()
```

[7]:

```
       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
```

1.5.1 4.1. Handling missing values

[8]:

```
#Count of null values in each column
netflix.isnull().sum()
```

[8]:

```
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
```

[9]:

```
#Percentage of null values in each column
for col in netflix.columns:
    nullValue = netflix[col].isnull().sum()
    percentage = (nullValue/len(netflix))*100
    print(col,":", percentage,"%")
```

```
show_id : 0.0 %
type : 0.0 %
title : 0.0 %
director : 29.908027705234474 %
cast : 9.367548540933349 %
country : 9.435676166685592 %
date_added : 0.11354604292040424 %
release_year : 0.0 %
rating : 0.045418417168169 %
duration : 0.034063812876121265 %
listed_in : 0.0 %
description : 0.0 %
```

Conclusion: * Columns cast, country, date_added, duration and rating are having less than 10% null values, dropping these NaN values wouldn't affect the dataset much. * director column has approx 30% null value , so instead of dropping them we will fill them with not known.

[10]:

```
#Dropping null values from cast, country, date_added, duration and rating␣
  ↪columns
netflix.
  ↪dropna(subset=['cast','country','date_added','duration','rating'],how='any',␣
  ↪inplace=True)
```

[11]:

```
#Filling null values of director column by 'Not Known'
netflix['director'].fillna(value='Not Known', inplace=True)
```

[12]:

```
netflix.isnull().sum()
```

[12]:

```
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
```

Result: Our Netflix dataset is clean now i.e. it doesn't contain any null value. Now lets check the size of updated Netflix dataset.

[13]:

```
netflix.shape
```

[13]:

```
(7290, 12)
```

1.5.2 4.2. Handling data type of columns

[14]:

```
#Converting 'date_added' column to datetime
netflix['date_added'] = pd.to_datetime(netflix['date_added'], format='%B %d,␣
 ↪%Y', errors='coerce')

#Converting apropriate columns to category type.
netflix = netflix.astype({'type': 'category',
                           'rating':'category'})
```

[15]:

```
netflix.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 7290 entries, 1 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       7290 non-null   object
 1   type          7290 non-null   category
 2   title         7290 non-null   object
 3   director      7290 non-null   object
 4   cast          7290 non-null   object
 5   country       7290 non-null   object
 6   date_added    7213 non-null   datetime64[ns]
 7   release_year  7290 non-null   int64
 8   rating        7290 non-null   category
 9   duration      7290 non-null   object
 10  listed_in     7290 non-null   object
 11  description   7290 non-null   object
dtypes: category(2), datetime64[ns](1), int64(1), object(8)
memory usage: 641.5+ KB
```

1.5.3 4.3. Handling duplicate values

[16]:

```python
netflix.duplicated().sum()
```

[16]:

```
0
```

Conclusion: There is no duplicate value in dataset.

1.5.4 4.4. Handling irrelevant columns

[17]:

```python
#Dropping description as it is of no use in analysis
netflix.drop('description', axis=1, inplace=True)
```

[18]:

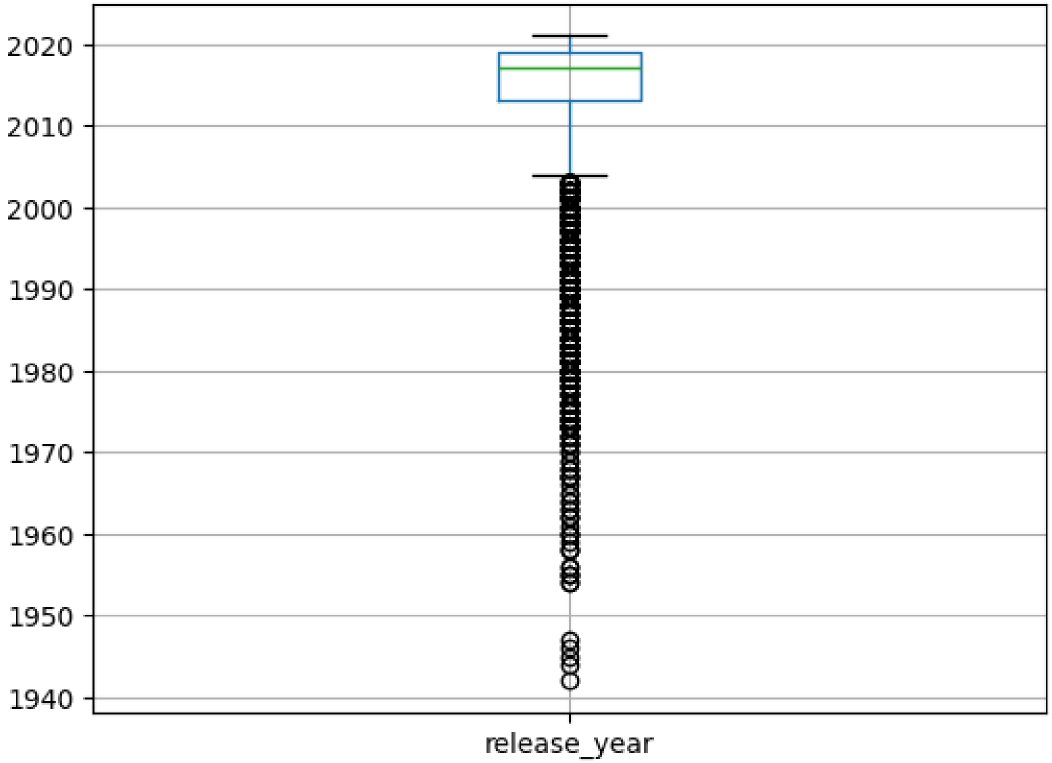
```python
netflix.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 7290 entries, 1 to 8806
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       7290 non-null   object
 1   type          7290 non-null   category
 2   title         7290 non-null   object
 3   director      7290 non-null   object
 4   cast          7290 non-null   object
 5   country       7290 non-null   object
 6   date_added    7213 non-null   datetime64[ns]
 7   release_year  7290 non-null   int64
 8   rating        7290 non-null   category
 9   duration      7290 non-null   object
 10  listed_in     7290 non-null   object
dtypes: category(2), datetime64[ns](1), int64(1), object(7)
memory usage: 584.5+ KB
```

1.5.5 4.5. Handling outliers

[19]:
```
#Viewing boxplot of release_year column as it has quantitative data
netflix.boxplot(column='release_year')
```

[19]:
```
<Axes: >
```



Conclusion: We can clearly see here that there are many outlier values in ‘release_year’ column but we can’t manipulate or remove these outlier values because there is no limit on the releasing years. Hence we will not operate on these outlier values.

1.6 5. Queries

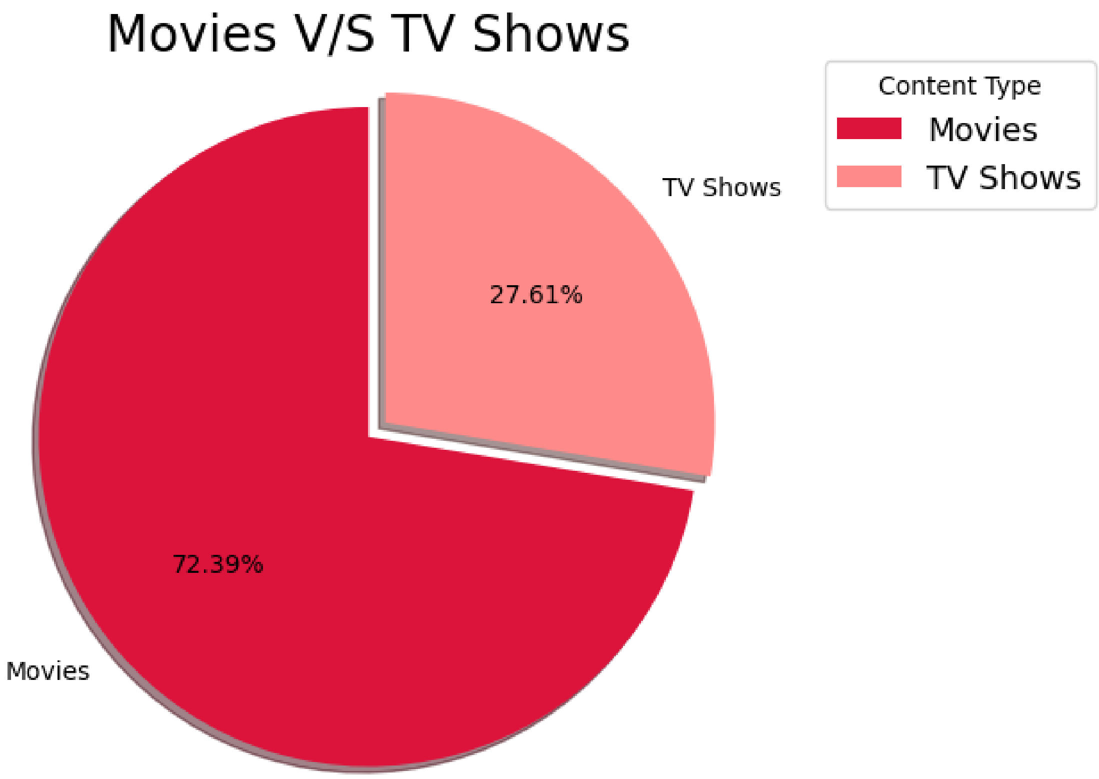
1.6.1 5.1 Total number of Movies and TV shows added on Netflix

[20]:
```
# Getting the count of total movies and TV shows
count = netflix['type'].value_counts()
print('Total number of Movies : ',count['Movie'])
print('Total number of TV shows : ',count['TV Show'])
```

```
Total number of Movies :  5277
Total number of TV shows :  2013
```

[21]:

```
# Pie chart for comparison between total number of movies and total number of␣
 ↪tv shows
plt.figure(figsize = (4,4))
plt.pie(count,
        labels=['Movies', 'TV Shows'],
        shadow=True,
        radius= 1.5,
        colors = ["#DC143C","#FF8A8A"],
        explode=[0,0.1],
        startangle=90,
        autopct= "%0.2f%%")
plt.legend(['Movies', 'TV Shows'], fontsize=13,
          title ="Content Type",
          loc ="center left",
          bbox_to_anchor =(1.3, 0.5, 2.5, 1.1))
plt.title('Movies V/S TV Shows',fontsize = 20,loc='center', pad=45)
plt.show()
```



Conclusion: Large number of Movies are added on Netflix as comapred to TV Shows.

1.6.2 5.2. Checking the type of content added on Netflix over the years

[22]:
```
#Adding required columns in the dataset
netflix['year_added']=netflix['date_added'].dt.year
netflix['month_added']=netflix['date_added'].dt.month_name()
netflix['day_added']=netflix['date_added'].dt.day_name()


#Dropping date_added column
netflix.drop('date_added', axis=1, inplace=True)
netflix.head()
```

[22]:
```
  show_id     type                          title         director  \
1      s2  TV Show                  Blood & Water        Not Known   
4      s5  TV Show                   Kota Factory        Not Known   
7      s8    Movie                        Sankofa     Haile Gerima   
8      s9  TV Show  The Great British Baking Show  Andy Devonshire   
9     s10    Movie                   The Starling   Theodore Melfi   

                                                cast  \
1  Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban…   
4  Mayur More, Jitendra Kumar, Ranjan Raj, Alam K…   
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  release_year rating  \
1                                       South Africa          2021  TV-MA   
4                                              India          2021  TV-MA   
7  United States, Ghana, Burkina Faso, United Kin…          1993  TV-MA   
8                                     United Kingdom          2021  TV-14   
9                                      United States          2021  PG-13   

    duration                                          listed_in  year_added  \
1  2 Seasons    International TV Shows, TV Dramas, TV Mysteries      2021.0   
4  2 Seasons  International TV Shows, Romantic TV Shows, TV …      2021.0   
7    125 min   Dramas, Independent Movies, International Movies      2021.0   
8  9 Seasons                       British TV Shows, Reality TV      2021.0   
9    104 min                                   Comedies, Dramas      2021.0   

  month_added day_added
1   September    Friday
4   September    Friday
7   September    Friday
8   September    Friday
9   September    Friday
```

[23]:

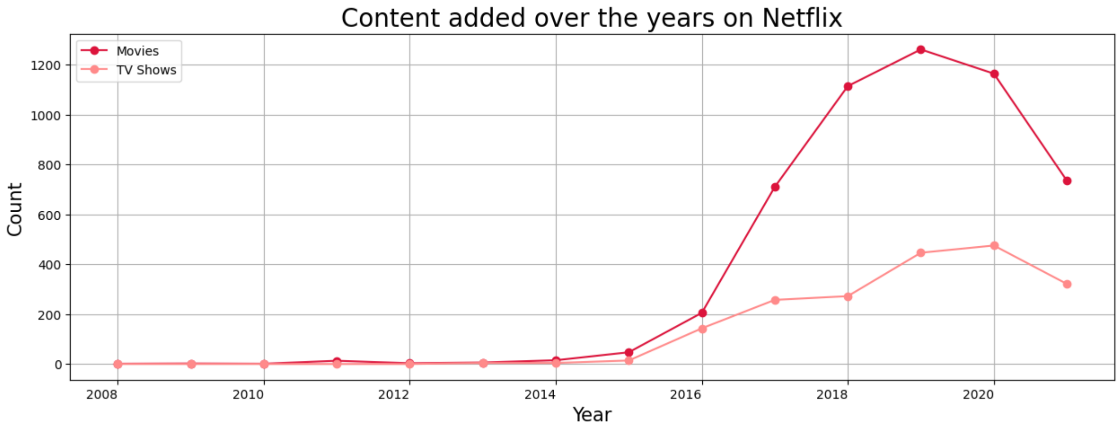
```python
#grouping the data by column 'year_added'
grouped = netflix.groupby('year_added')['type'].value_counts().unstack()
grouped
```

[23]:

```
type        Movie  TV Show
year_added
2008.0          1        0
2009.0          2        0
2010.0          1        0
2011.0         13        0
2012.0          3        0
2013.0          6        4
2014.0         15        4
2015.0         47       14
2016.0        205      143
2017.0        710      257
2018.0       1114      272
2019.0       1261      446
2020.0       1164      475
2021.0        735      321
```

[24]:

```python
#Plotting
plt.figure(figsize = (15,5))
plt.plot(grouped['Movie'], marker='o', color='#DC143C')
plt.plot(grouped['TV Show'], marker='o', color='#FF8A8A')
plt.xlabel('Year',fontsize=15)
plt.ylabel('Count',fontsize=15)
plt.xticks(size = 10, ha="right")
plt.yticks(size = 10)
plt.title('Content added over the years on Netflix', fontsize=20)
plt.legend(['Movies', 'TV Shows'], loc='upper left')
plt.grid()
plt.show()
```



Conclusion: * After year 2015 a lot of content is added on Netflix. * The number of movies added become significantly larger than TV shows since 2017. * There is a significant drop in content addition in both type after 2019 that is because of Covid.

1.6.3 5.3 Checking the number of movies and TV Shows added on Netflix every month

[25]:

```
#Splitting the dataset into separate groups of Month:
grouped = netflix['month_added'].groupby(netflix['type']).value_counts().
 ↪unstack('type')
grouped
```

```
C:\Users\ashis\AppData\Local\Temp\ipykernel_21692\1676305996.py:2:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped =
netflix['month_added'].groupby(netflix['type']).value_counts().unstack('type')
```

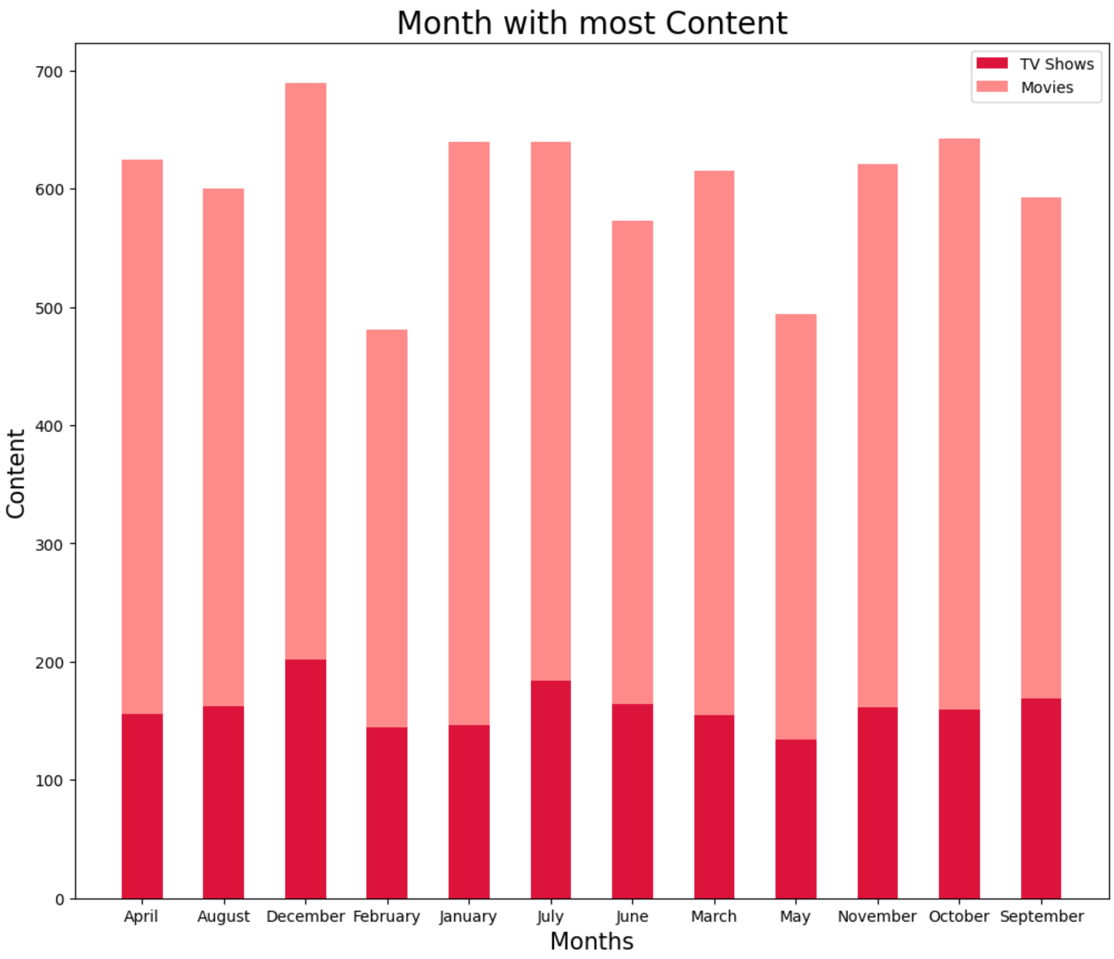
[25]:

```
type          Movie  TV Show
month_added
April           469      156
August          438      162
December        487      202
February        337      144
January         494      146
July            456      184
June            409      164
March           460      155
May             360      134
November        460      161
October         483      159
September       424      169
```

[26]:

```
#Plotting Bar plot to check the count of content added each month:
plt.figure(figsize = (15,5))
fig = plt.subplots(figsize =(12, 10))
p1 = plt.bar(grouped.index, grouped['TV Show'], 0.5, color='#DC143C')
p2 = plt.bar(grouped.index, grouped['Movie'], 0.5,
             bottom = grouped['TV Show'], color='#FF8A8A')
plt.ylabel('Content',fontsize = 15)
plt.xlabel('Months',fontsize = 15)
plt.title('Month with most Content',fontsize = 20)
plt.legend((p1[0], p2[0]), ('TV Shows', 'Movies'))
plt.show()
```

```
<Figure size 1500x500 with 0 Axes>
```



Conclusion: * Most content was added on Netflix in the month of December, January and October in total. * Also the number of Movie content added is much higher than TV Show for every month.

1.6.4 5.4. The most common runtime of Movies.

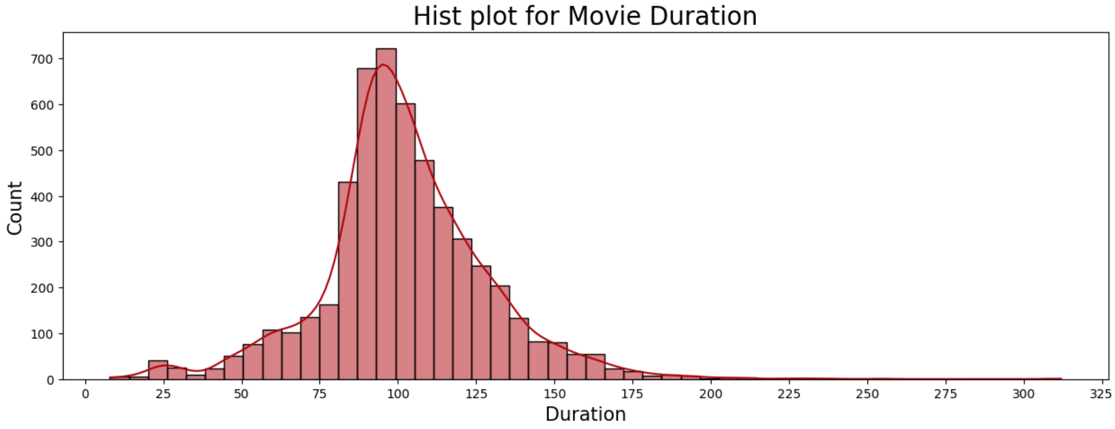
[27]:
```
# Getting the runtime of movies
runtime = netflix[netflix['duration'].str.contains('min')]['duration'].
  ↪apply(lambda x: x.split()[0]).astype('int')
runtime
```

[27]:
```
7         125
9         104
12        127
24        166
27        103
         …
8801       96
8802      158
```

```
8804     88
8805     88
8806    111
Name: duration, Length: 5277, dtype: int32
```

[28]:

```
#plotting histogram
fig, ax1 = plt.subplots(1, 1, figsize=(15,5))
g = sns.histplot(runtime, kde=True, color="#B00710", bins=50, ax = ax1)
ax1.set_title('Hist plot for Movie Duration', fontsize=20)
ax1.set_xlabel('Duration', fontsize=15)
ax1.set_ylabel('Count', fontsize=15)
g.set(xticks=np.arange(0, 350, 25))
plt.show()
```



Conclusion: Generally movies have runtime range around 100 mins.

1.6.5 5.5. The most common number of seasons released on Netflix

[29]:

```
#Getting the number of seasons for each TV Show on netflix
seasons = netflix[netflix['duration'].str.contains('Season')]['duration'].
  ↪value_counts().sort_values()
seasons
```

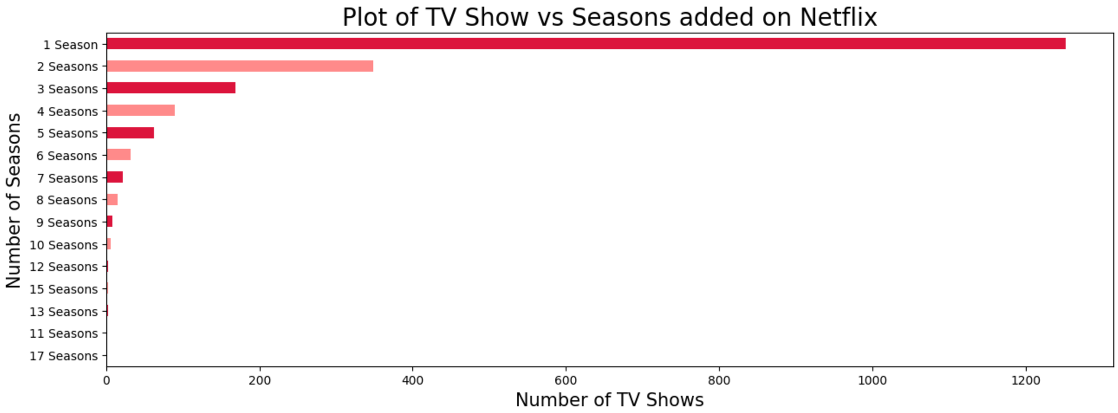
[29]:

```
duration
17 Seasons      1
11 Seasons      1
13 Seasons      2
15 Seasons      2
12 Seasons      2
10 Seasons      6
9 Seasons       8
8 Seasons      15
```

```
7 Seasons       22
6 Seasons       32
5 Seasons       63
4 Seasons       90
3 Seasons      169
2 Seasons      348
1 Season      1252
Name: count, dtype: int64
```

[30]:

```
#Plotting a bar graph of No of seasons vs No of TV shows
plt.figure(figsize=(15,5))
ax = seasons.plot(y='duration', kind='barh',color=["#DC143C", '#FF8A8A'])
ax.set_title('Plot of TV Show vs Seasons added on Netflix', fontsize=20)
ax.set_xlabel('Number of TV Shows', fontsize=15)
ax.set_ylabel('Number of Seasons', fontsize=15)
plt.show()
```



Conclusion: * Most of the TV shows have only 1-2 season. * We can also observe that there are very rare TV shows having more than 10 seasons.

1.6.6 5.6. Content as per rating

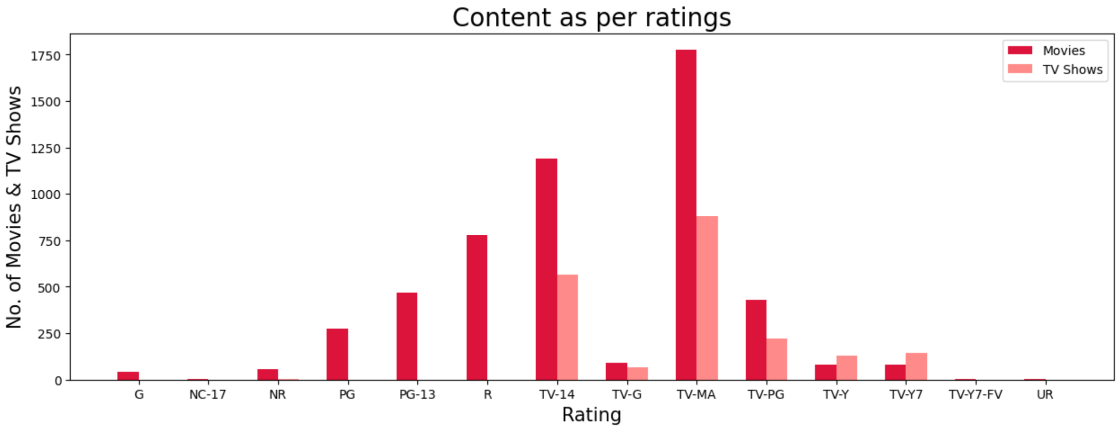
[31]:

```
#Getting the no. of Movies and TV Shows for a particular rating
grouped = netflix['rating'].groupby(netflix['type']).value_counts().
  ↪unstack('type')
grouped
```

```
C:\Users\ashis\AppData\Local\Temp\ipykernel_21692\447544373.py:2: FutureWarning:
The default of observed=False is deprecated and will be changed to True in a
future version of pandas. Pass observed=False to retain current behavior or
observed=True to adopt the future default and silence this warning.
  grouped =
netflix['rating'].groupby(netflix['type']).value_counts().unstack('type')
```

[31]:
```
type       Movie  TV Show
rating
G             40        0
NC-17          3        0
NR            58        4
PG           275        0
PG-13        470        0
R            778        1
TV-14       1191      564
TV-G          90       68
TV-MA       1776      881
TV-PG        430      223
TV-Y          80      129
TV-Y7         80      142
TV-Y7-FV       3        1
UR             3        0
```

[32]:
```
#Plotting a bar graph to show the No of Movies and TV Shows for each Rating type
X_axis = np.arange(len(grouped.index))
plt.figure(figsize = (15,5))
plt.bar(X_axis-0.15, grouped['Movie'], 0.3, label = 'Movies', color='#DC143C')
plt.bar(X_axis+0.15, grouped['TV Show'], 0.3, label = 'TV Shows',␣
  ↪color='#FF8A8A')
plt.title("Content as per ratings", fontsize = 20)
plt.xlabel("Rating", fontsize = 15)
plt.ylabel("No. of Movies & TV Shows",fontsize=15)
plt.xticks(X_axis, grouped.index, size = 10)
plt.yticks(size = 10)
plt.legend()
plt.show()
```



Conclusion: * Movies with rating TV-MA are mostly added and movies with rating NC-17, TV-

Y7-FV, UR are negligible on netflix. * TV Shows with rating TV-MA are mostly added and there are no TV Shows with rating NC-17, TV-Y7-FV, UR, NR, R, PG-13,PG, G on netflix. * Along with that TV-14 is also a commonly occurring rating in both Movies and TV Shows.

1.6.7 5.7. The count of content added as per audience type

[33]:

```
#Dividing the Netflix content into 'Adult', 'Kid' and 'Teen' categories␣
 ↪according to the rating
Kids = ['TV-Y', 'TV-Y7', 'G', 'TV-G', 'PG', 'TV-PG', 'TV-Y7-FV']
Teens = ['PG-13', 'TV-14']
Adults = ['R', 'TV-MA', 'NC-17', 'UR', 'NR']

def rate(x):
    if x in Kids:
        return 'Kid'
    elif x in Teens:
        return 'Teen'
    elif x in Adults:
        return 'Adult'

df1 = netflix['rating'].apply(rate)
df2 = pd.concat([df1, netflix['type']], axis=1)
df2
```

[33]:

```
     rating     type
1     Adult  TV Show
4     Adult  TV Show
7     Adult    Movie
8      Teen  TV Show
9      Teen    Movie
…       …       …
8801  Adult    Movie
8802  Adult    Movie
8804  Adult    Movie
8805    Kid    Movie
8806   Teen    Movie

[7290 rows x 2 columns]
```

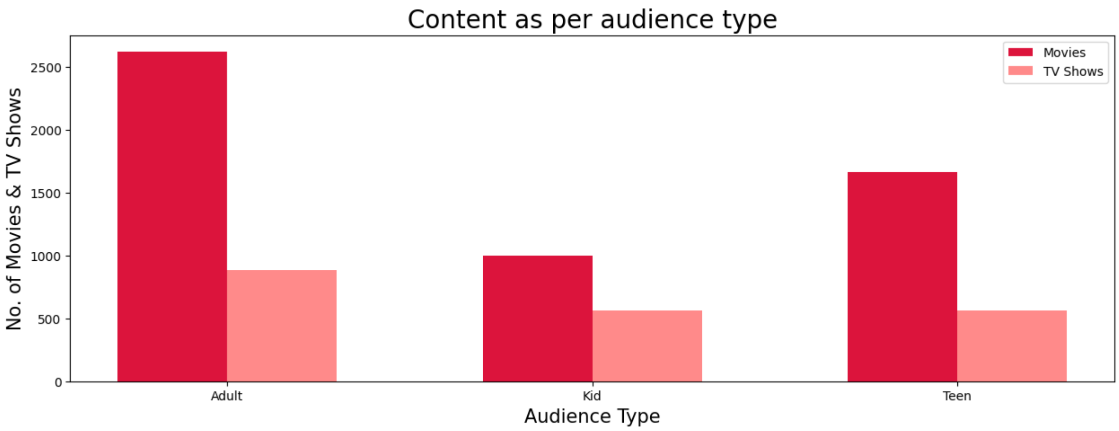
[34]:

```
#Getting the total no. of Movies and TV Shows for each audience type
grouped = df2['rating'].groupby(df2['type']).value_counts().unstack('type')
grouped
```

```
C:\Users\ashis\AppData\Local\Temp\ipykernel_21692\2519060405.py:2:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped = df2['rating'].groupby(df2['type']).value_counts().unstack('type')
```

[34]:
```
type    Movie  TV Show
rating
Adult    2618      886
Kid       998      563
Teen     1661      564
```

[35]:
```
#Plotting a bar graph showing the No of Movies and TV Shows for each of the␣
  ↪audience type
X_axis = np.arange(len(grouped.index))
plt.figure(figsize = (15,5))
plt.bar(X_axis-0.15, grouped['Movie'], 0.3, label = 'Movies', color='#DC143C')
plt.bar(X_axis+0.15, grouped['TV Show'], 0.3, label = 'TV Shows',␣
  ↪color='#FF8A8A')
plt.title("Content as per audience type", fontsize = 20)
plt.xlabel("Audience Type", fontsize = 15)
plt.ylabel("No. of Movies & TV Shows",fontsize=15)
plt.xticks(X_axis, grouped.index, size = 10)
plt.yticks(size = 10)
plt.legend()
plt.show()
```



Conclusion: * Netflix added a lot of adult movies as compared to adult TV shows. * Adult and Teen content is the most common content on netflix. * Kid content is the least on Netflix for both movies and TV Shows.

1.6.8 5.8 The count of content added for Top 20 countries

[36]:
```
def explode_data(df, col, name, along):
    return (df[col].replace(' ,', ',').replace(', ',',').str.split(',')
                   .to_frame()
                   .set_index(df[along])
                   .explode(col)
```

```
                        .replace('',np.nan)
                        .replace('NA',np.nan)
                        .dropna()
                        .reset_index())
```

[37]:
```
countries = explode_data(netflix, 'country', 'country', 'type')
countries['country']=countries['country'].str.strip()
countries
```

[37]:
```
         type        country
0     TV Show   South Africa
1     TV Show          India
2       Movie  United States
3       Movie          Ghana
4       Movie   Burkina Faso
…         …            …
9142    Movie         Jordan
9143    Movie  United States
9144    Movie  United States
9145    Movie  United States
9146    Movie          India

[9147 rows x 2 columns]
```

[38]:
```
# Getting the no. of movies, TV Shows individually and both of them together␣
  ↪for each country:
grouped = countries['country'].groupby(countries['type']).value_counts().
  ↪unstack('type').replace(np.nan,0)
grouped.columns = grouped.columns.astype(str)
grouped['sum'] = grouped.sum(axis=1)
grouped=grouped.sort_values(by='sum', ascending=False)
grouped
```

```
C:\Users\ashis\AppData\Local\Temp\ipykernel_21692\2835970324.py:2:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped = countries['country'].groupby(countries['type']).value_counts().unsta
ck('type').replace(np.nan,0)
```

[38]:
```
type            Movie  TV Show   sum
country
United States    2492      782  3274
India             942       65  1007
United Kingdom    476      232   708
Canada            295      119   414
France            285       76   361
…                 …       …    …
```

```
Mozambique          1        0     1
Namibia             1        0     1
Nicaragua           1        0     1
Panama              1        0     1
Zimbabwe            1        0     1

[115 rows x 3 columns]
```

[39]:

```
#Top 20 countries producing content on Netflix
group=grouped.head(20)
group
```

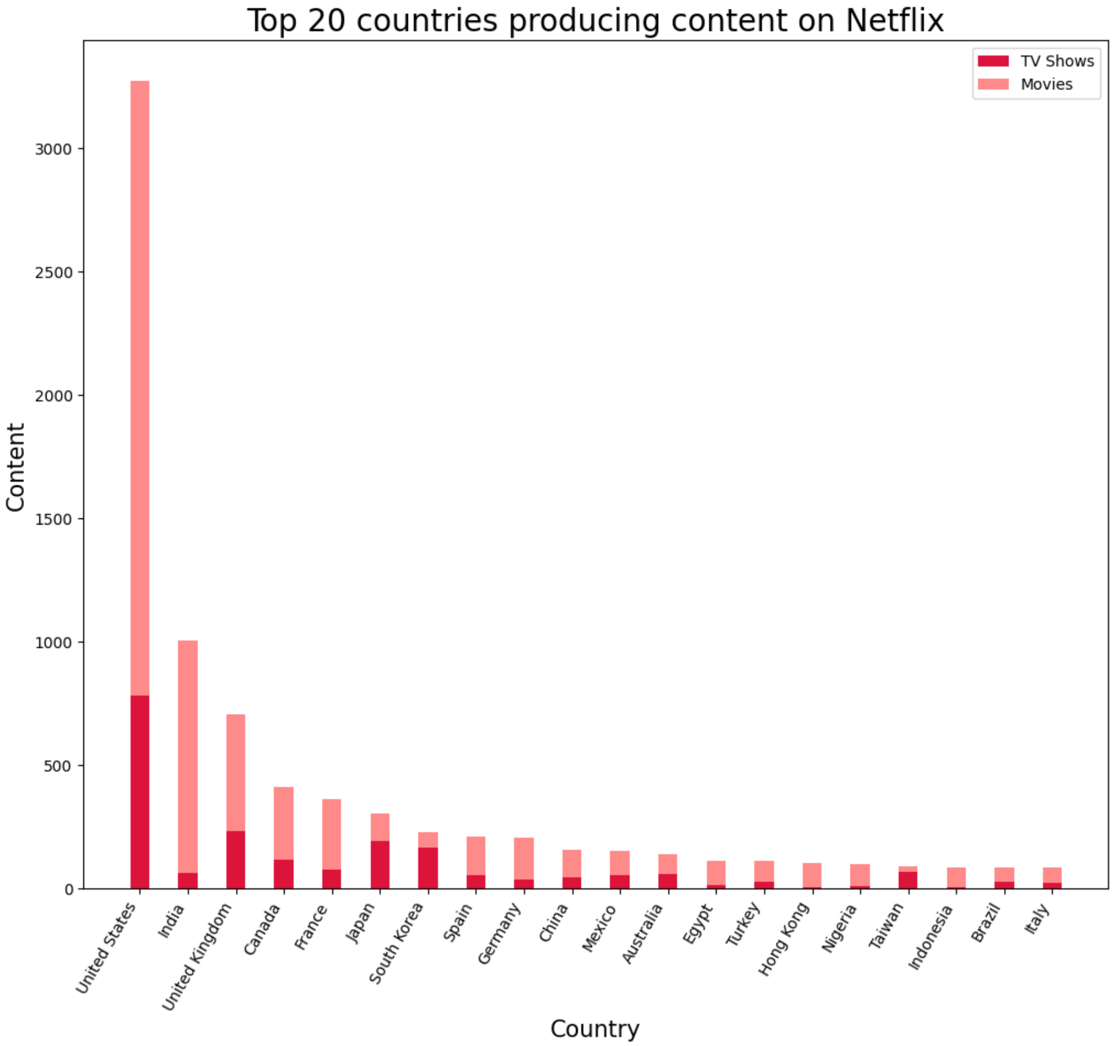
[39]:

```
type            Movie  TV Show   sum
country
United States    2492      782  3274
India             942       65  1007
United Kingdom    476      232   708
Canada            295      119   414
France            285       76   361
Japan             114      193   307
South Korea        59      168   227
Spain             156       53   209
Germany           168       39   207
China             108       48   156
Mexico             98       55   153
Australia          83       58   141
Egypt             100       15   115
Turkey             83       30   113
Hong Kong         100        5   105
Nigeria            92        9   101
Taiwan             19       70    89
Indonesia          84        4    88
Brazil             58       27    85
Italy              62       23    85
```

[40]:

```
# Plotting bar graph of netflix content country-wise:
plt.figure(figsize = (20,10))
fig = plt.subplots(figsize =(12, 10))
p1 = plt.bar(group.index, group['TV Show'], 0.4, color='#DC143C')
p2 = plt.bar(group.index, group['Movie'], 0.4,
             bottom = group['TV Show'], color='#FF8A8A')
plt.ylabel('Content',fontsize = 15)
plt.xlabel('Country',fontsize = 15)
plt.xticks(size=10, rotation=60, ha='right')
plt.title('Top 20 countries producing content on Netflix',fontsize = 20)
plt.legend((p1[0], p2[0]), ('TV Shows', 'Movies'))
plt.show()
```

```
<Figure size 2000x1000 with 0 Axes>
```



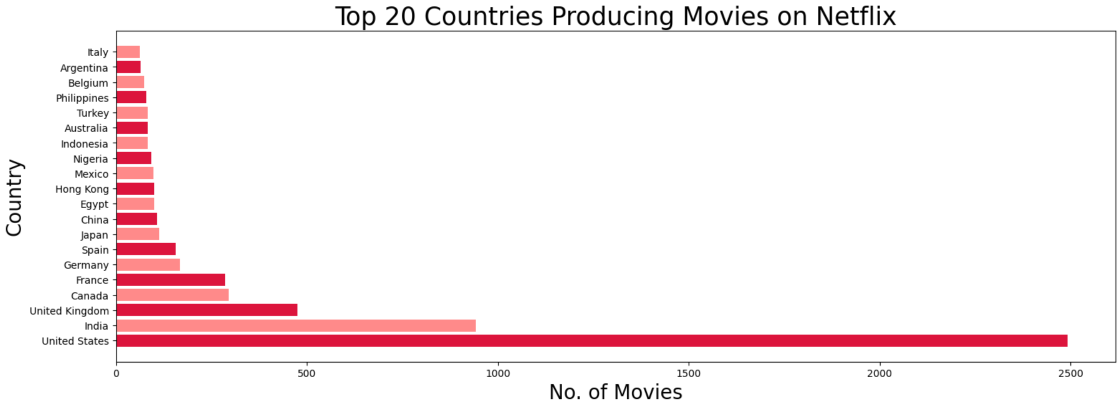
Conclusion: * United States is producing much more content on netflix than any other country.* After United States, India and United Kingdom are the most content producing countries on Netflix.

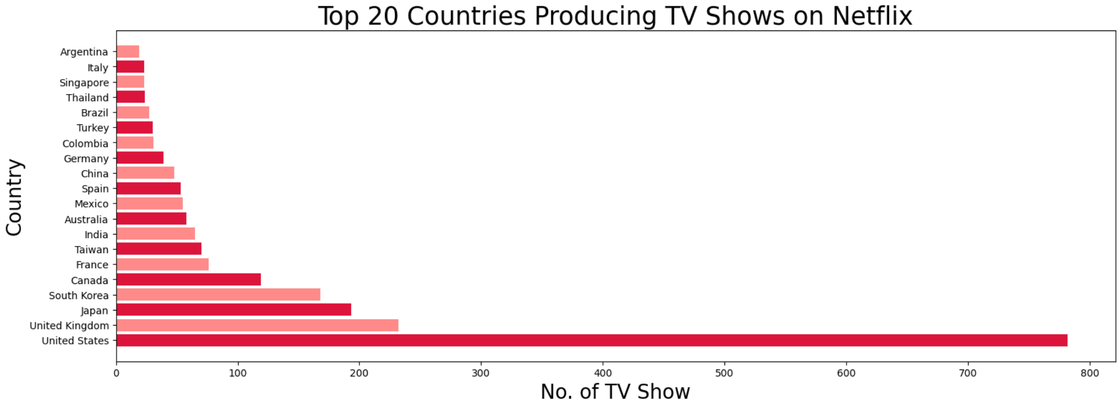
1.6.9 5.9 Top 20 countries producing Movies and TV Shows respectively

[41]:
```
#Top 20 countries producing Movies
grouped.sort_values(by='Movie',inplace=True, ascending=False)
group=grouped.head(20)
#Plotting
fig, (ax1, ax2) = plt.subplots(2,1,figsize=(18,15))
fig.subplots_adjust(hspace=0.5)
ax1.barh(group.index, group['Movie'], color=['#DC143C', '#FF8A8A'])
ax1.set_xlabel('No. of Movies', fontsize=20)
```

```
ax1.set_ylabel('Country', fontsize=20)
ax1.set_title('Top 20 Countries Producing Movies on Netflix', fontsize=25)

#Top 20 countries producing TV Shows
grouped.sort_values(by='TV Show', ascending=False, inplace=True)
group=grouped.head(20)
#Plotting
ax2.barh(group.index, group['TV Show'], color=['#DC143C', '#FF8A8A'])
ax2.set_xlabel('No. of TV Show', fontsize=20)
ax2.set_ylabel('Country', fontsize=20)
ax2.set_title('Top 20 Countries Producing TV Shows on Netflix', fontsize=25)
plt.show()
```





Conclusion: * The order of top 20 Movie and TV Show separately is different. * In countries like India, Spain, Germany Movies are more popular than TV Shows. * In countries like Japan, South Korea, Taiwan TV Shows are more popular than Movies.

1.6.10 5.10. Top 10 genres popular on Netflix

[42]:

```
genre=explode_data(netflix, 'listed_in', 'genre', 'type')
genre['listed_in']=genre['listed_in'].str.strip()
genre
```

[42]:

```
          type                 listed_in
0      TV Show    International TV Shows
1      TV Show                 TV Dramas
2      TV Show              TV Mysteries
3      TV Show    International TV Shows
4      TV Show          Romantic TV Shows
…          …                        …
16363    Movie  Children & Family Movies
16364    Movie                  Comedies
16365    Movie                    Dramas
16366    Movie      International Movies
16367    Movie          Music & Musicals

[16368 rows x 2 columns]
```

[43]:

```
#Displaying the number of movies and tv shows according to different genres
grouped = genre['listed_in'].groupby(genre['type']).value_counts().
↪unstack('type').replace(np.nan,0)
grouped
```

```
C:\Users\ashis\AppData\Local\Temp\ipykernel_21692\800436048.py:2: FutureWarning:
The default of observed=False is deprecated and will be changed to True in a
future version of pandas. Pass observed=False to retain current behavior or
observed=True to adopt the future default and silence this warning.
  grouped = genre['listed_in'].groupby(genre['type']).value_counts().unstack('ty
pe').replace(np.nan,0)
```

[43]:

```
type                        Movie  TV Show
listed_in
Action & Adventure            811        0
Anime Features                 61        0
Anime Series                    0      150
British TV Shows                0      201
Children & Family Movies      521        0
Classic & Cult TV               0       25
Classic Movies                108        0
Comedies                     1574        0
Crime TV Shows                  0      358
Cult Movies                    69        0
Documentaries                 410        0
Docuseries                      0      162
Dramas                       2309        0
```

```
Faith & Spirituality             58        0
Horror Movies                   338        0
Independent Movies              742        0
International Movies           2392        0
International TV Shows            0     1047
Kids' TV                          0      342
Korean TV Shows                   0      133
LGBTQ Movies                     82        0
Movies                           30        0
Music & Musicals                299        0
Reality TV                        0      141
Romantic Movies                 583        0
Romantic TV Shows                 0      295
Sci-Fi & Fantasy                239        0
Science & Nature TV               0       52
Spanish-Language TV Shows         0      139
Sports Movies                   157        0
Stand-Up Comedy                 310        0
Stand-Up Comedy & Talk Shows      0       41
TV Action & Adventure             0      151
TV Comedies                       0      480
TV Dramas                         0      657
TV Horror                         0       69
TV Mysteries                      0       88
TV Sci-Fi & Fantasy               0       76
TV Shows                          0        5
TV Thrillers                      0       50
Teen TV Shows                     0       64
Thrillers                       549        0
```

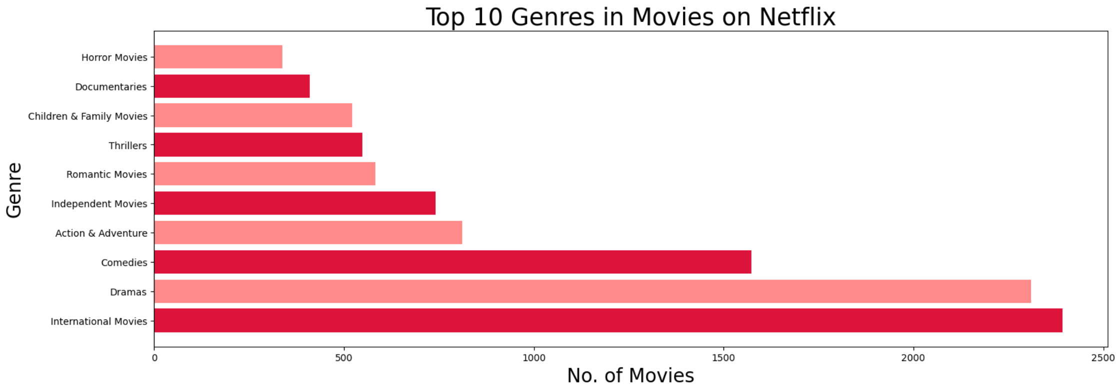
[44]:

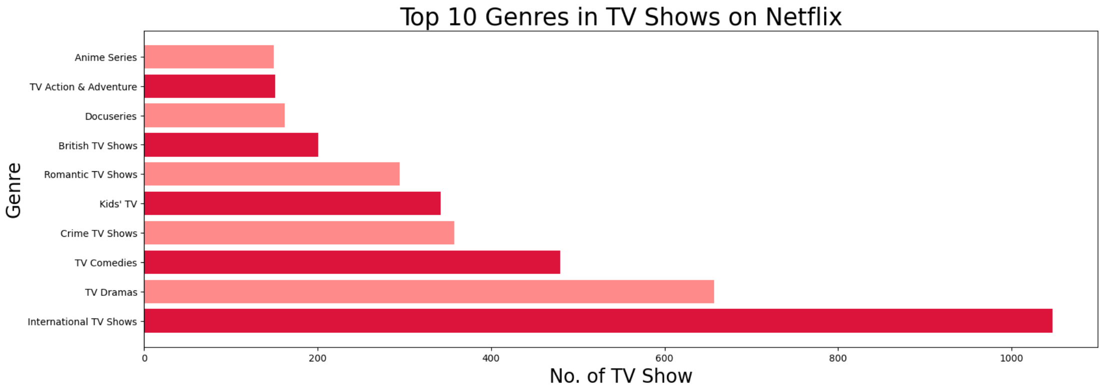
```python
#Top 10 genres in Movies
grouped.sort_values(by='Movie', ascending=False, inplace=True)
group=grouped.head(10)
#Plotting
fig, (ax1, ax2) = plt.subplots(2,1,figsize=(18,15))
fig.subplots_adjust(hspace=0.5)
ax1.barh(group.index, group['Movie'], color=['#DC143C', '#FF8A8A'])
ax1.set_xlabel('No. of Movies', fontsize=20)
ax1.set_ylabel('Genre', fontsize=20)
ax1.set_title('Top 10 Genres in Movies on Netflix', fontsize=25)

#Top 10 genres in TV Shows
grouped.sort_values(by='TV Show', ascending=False, inplace=True)
group=grouped.head(10)

#Plotting
ax2.barh(group.index, group['TV Show'], color=['#DC143C', '#FF8A8A'])
```

```
ax2.set_xlabel('No. of TV Show', fontsize=20)
ax2.set_ylabel('Genre', fontsize=20)
ax2.set_title('Top 10 Genres in TV Shows on Netflix', fontsize=25)
plt.show()
```





Conclusion: * TV Show genres Crime TV Shows and Kids' TV are more popular than their similar genre in Movies. * Movie genres Documentaries and Action & Adventure are more popular than their similar genre in TV Show.

1.6.11 5.11. Top 10 Movie genre per country

[45]:
```
def make_data(df):
    return (
        df[df['country'].isin(df['country'].value_counts().head(10).index)]
        .value_counts()
        .reset_index()
        .rename(columns={0:'count'})
        .pivot(index='listed_in', columns='country', values='count')
        .fillna(0)
        )
```

[46]:

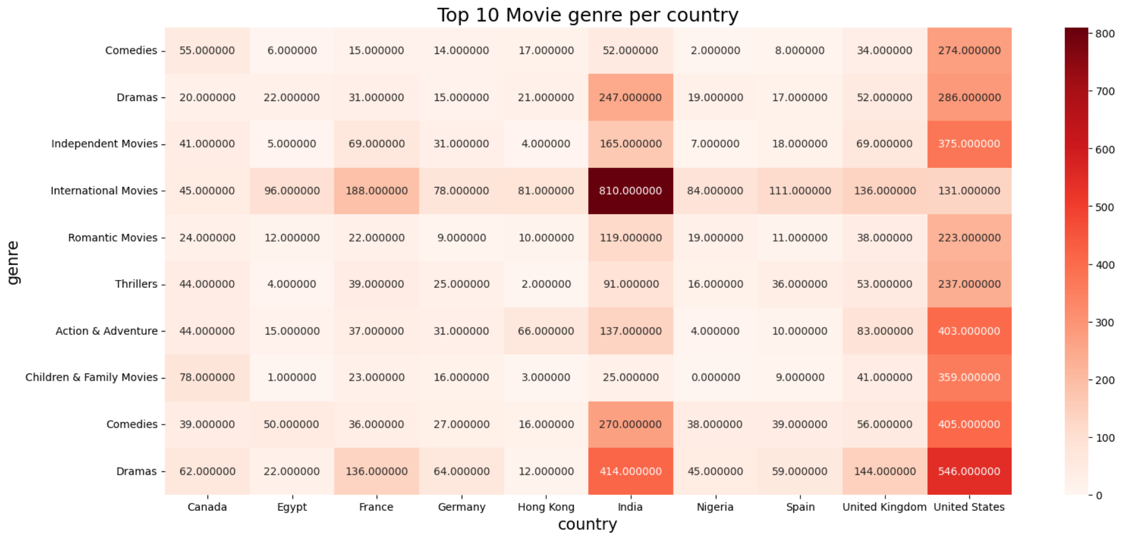
```
country_data = explode_data(netflix, 'country', 'country', 'title')
country_data['country'] = country_data['country'].str.strip()
genre_data = explode_data(netflix, 'listed_in', 'listed_in', 'title')
genre_data_type = explode_data(netflix, 'listed_in', 'listed_in', 'type')

genre_data_type = genre_data_type.value_counts().reset_index(level=1)
top_movie_genres = list(genre_data_type.loc['Movie'].head(10)['listed_in'].
  ↪values)

df = country_data.merge(genre_data).drop('title', axis=1)
df1 = df[df['listed_in'].isin(top_movie_genres)]
df1 = make_data(df1)
```

[47]:

```
#plotting heatmap for top 10 movie genre per country
plt.figure(figsize=(18,8))
ax = sns.heatmap(df1, annot=True, fmt="f", cmap='Reds')
plt.ylabel('genre',fontsize = 15)
plt.xlabel('country',fontsize = 15)
plt.title('Top 10 Movie genre per country',fontsize = 18)
plt.show()
```



Conclusion: > Top 3 genres in movies for top 3 countries are:- * United States: Dramas, Comedies, Action & Adventure * India: International Movies, Dramas and Comedies * United Kingdom: Dramas, International Movies, Documentries

1.6.12 5.12. Top 10 TV show genre per country

[48]:

```
top_tv_genres = list(genre_data_type.loc['TV Show'].head(10)['listed_in'].
 ↪values)
df2 = df[df['listed_in'].isin(top_tv_genres)]
df2 = make_data(df2)
df2
```

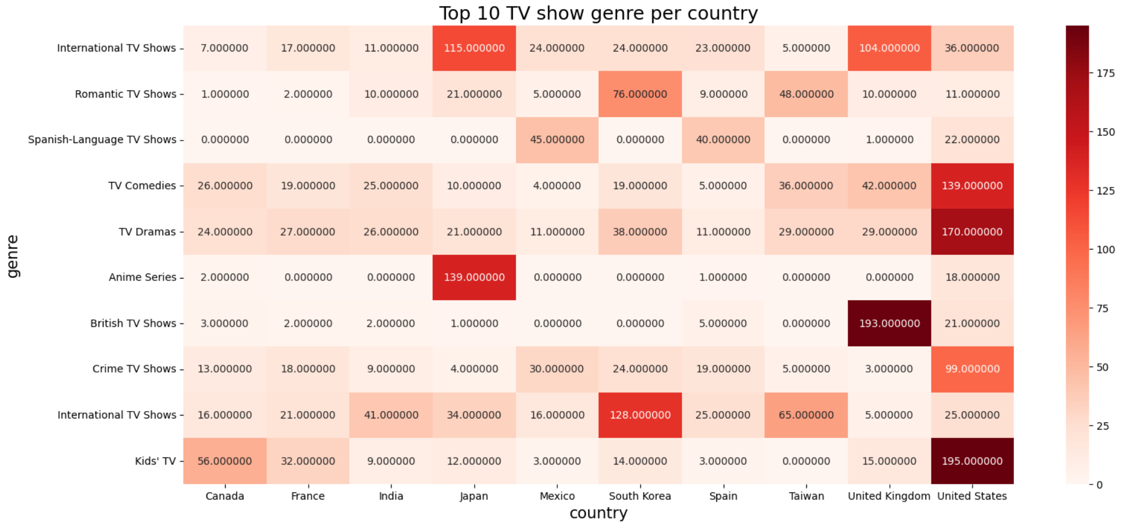
[48]:

```
country                     Canada  France  India  Japan  Mexico  South Korea  \
listed_in
 International TV Shows        7.0    17.0   11.0  115.0    24.0         24.0
 Romantic TV Shows             1.0     2.0   10.0   21.0     5.0         76.0
 Spanish-Language TV Shows     0.0     0.0    0.0    0.0    45.0          0.0
 TV Comedies                  26.0    19.0   25.0   10.0     4.0         19.0
 TV Dramas                    24.0    27.0   26.0   21.0    11.0         38.0
Anime Series                   2.0     0.0    0.0  139.0     0.0          0.0
British TV Shows               3.0     2.0    2.0    1.0     0.0          0.0
Crime TV Shows                13.0    18.0    9.0    4.0    30.0         24.0
International TV Shows        16.0    21.0   41.0   34.0    16.0        128.0
Kids' TV                      56.0    32.0    9.0   12.0     3.0         14.0

country                     Spain  Taiwan  United Kingdom  United States
listed_in
 International TV Shows      23.0     5.0           104.0           36.0
 Romantic TV Shows            9.0    48.0            10.0           11.0
 Spanish-Language TV Shows   40.0     0.0             1.0           22.0
 TV Comedies                  5.0    36.0            42.0          139.0
 TV Dramas                   11.0    29.0            29.0          170.0
Anime Series                  1.0     0.0             0.0           18.0
British TV Shows              5.0     0.0           193.0           21.0
Crime TV Shows               19.0     5.0             3.0           99.0
International TV Shows       25.0    65.0             5.0           25.0
Kids' TV                      3.0     0.0            15.0          195.0
```

[49]:

```
#plotting heatmap for top 10 TV show genre per country
plt.figure(figsize=(18,8))
ax = sns.heatmap(df2, annot=True, fmt="f", cmap='Reds')
plt.ylabel('genre',fontsize = 15)
plt.xlabel('country',fontsize = 15)
plt.title('Top 10 TV show genre per country',fontsize = 18)
plt.show()
```



Conclusion: > Top 3 genres in TV Shows for top 3 countries are:- * United States: TV Comedies, TV Dramas, Kids' TV * United Kingdom: British TV Shows, International TV Shows, Docuseries * Japan: International TV Shows, Anime Series, Kids' TV

1.6.13 5.13. Checking overall content added on Netflix monthly per year

[50]:
```
year_month_count = (netflix.loc[:,['year_added','month_added']]
                    .value_counts()
                    .reset_index()
                    .rename(columns={0:'count'})
                    .pivot(index="month_added", columns="year_added",␣
 ↪values="count")
                    .fillna(0)
                    .apply(lambda x: x.astype('int')))
year_month_count
```

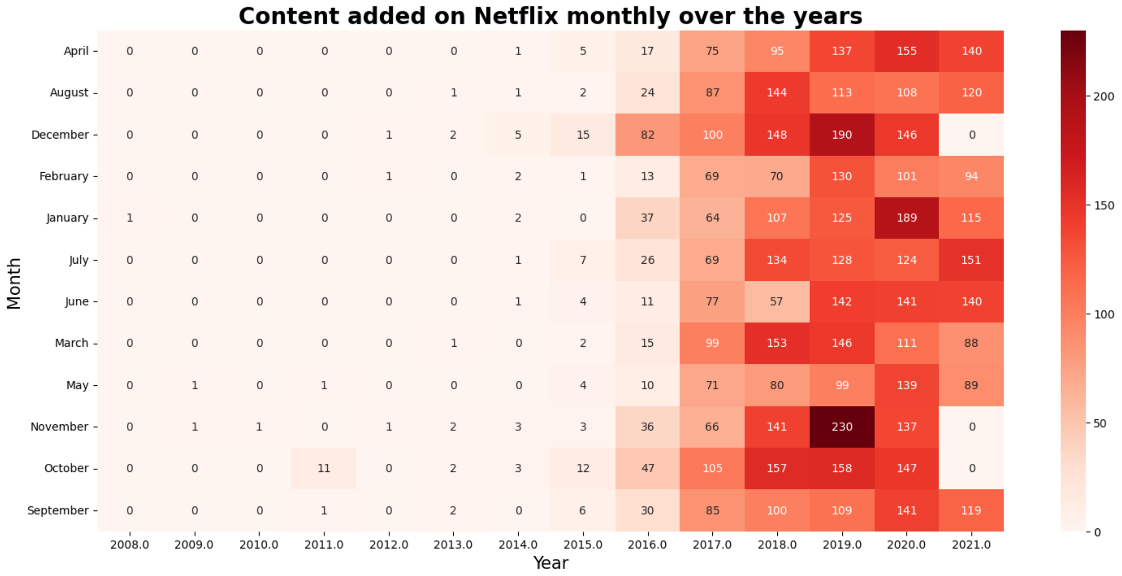
[50]:
```
year_added   2008.0  2009.0  2010.0  2011.0  2012.0  2013.0  2014.0  2015.0  \
month_added
April             0       0       0       0       0       0       1       5
August            0       0       0       0       0       1       1       2
December          0       0       0       0       1       2       5      15
February          0       0       0       0       1       0       2       1
January           1       0       0       0       0       0       2       0
July              0       0       0       0       0       0       1       7
June              0       0       0       0       0       0       1       4
March             0       0       0       0       0       1       0       2
May               0       1       0       1       0       0       0       4
November          0       1       1       0       1       2       3       3
October           0       0       0      11       0       2       3      12
```

```
September          0       0       0       1       0       2       0       6

year_added    2016.0  2017.0  2018.0  2019.0  2020.0  2021.0
month_added
April             17      75      95     137     155     140
August            24      87     144     113     108     120
December          82     100     148     190     146       0
February          13      69      70     130     101      94
January           37      64     107     125     189     115
July              26      69     134     128     124     151
June              11      77      57     142     141     140
March             15      99     153     146     111      88
May               10      71      80      99     139      89
November          36      66     141     230     137       0
October           47     105     157     158     147       0
September         30      85     100     109     141     119
```

Plotting Heatmap to check the content added on Netflix monthly over the years:

[51]:
```
#plotting heatmap
plt.figure(figsize=(18,8))
ax = sns.heatmap(year_month_count, annot=True, fmt="d", cmap='Reds')
plt.ylabel('Month',fontsize = 15)
plt.xlabel('Year',fontsize = 15)
plt.title('Content added on Netflix monthly over the years',fontsize = 20,␣
  ↪fontweight = 'bold')
plt.show()
```



Conclusion: * The month of July has the highest content in 2021 but before 2021 content added in July was not very high. * If we do not consider the content added in 2021 as we don't have data for all months, then more content is added in October, November and December. * For year 2021 we do not have the data after september.

1.6.14 5.14. Overall content added by netflix each week day

Splitting the dataset into separate groups of weekdays:

```
[52]: grouped = netflix['day_added'].groupby(netflix['type']).value_counts().
        ↪unstack('type')
      grouped
```

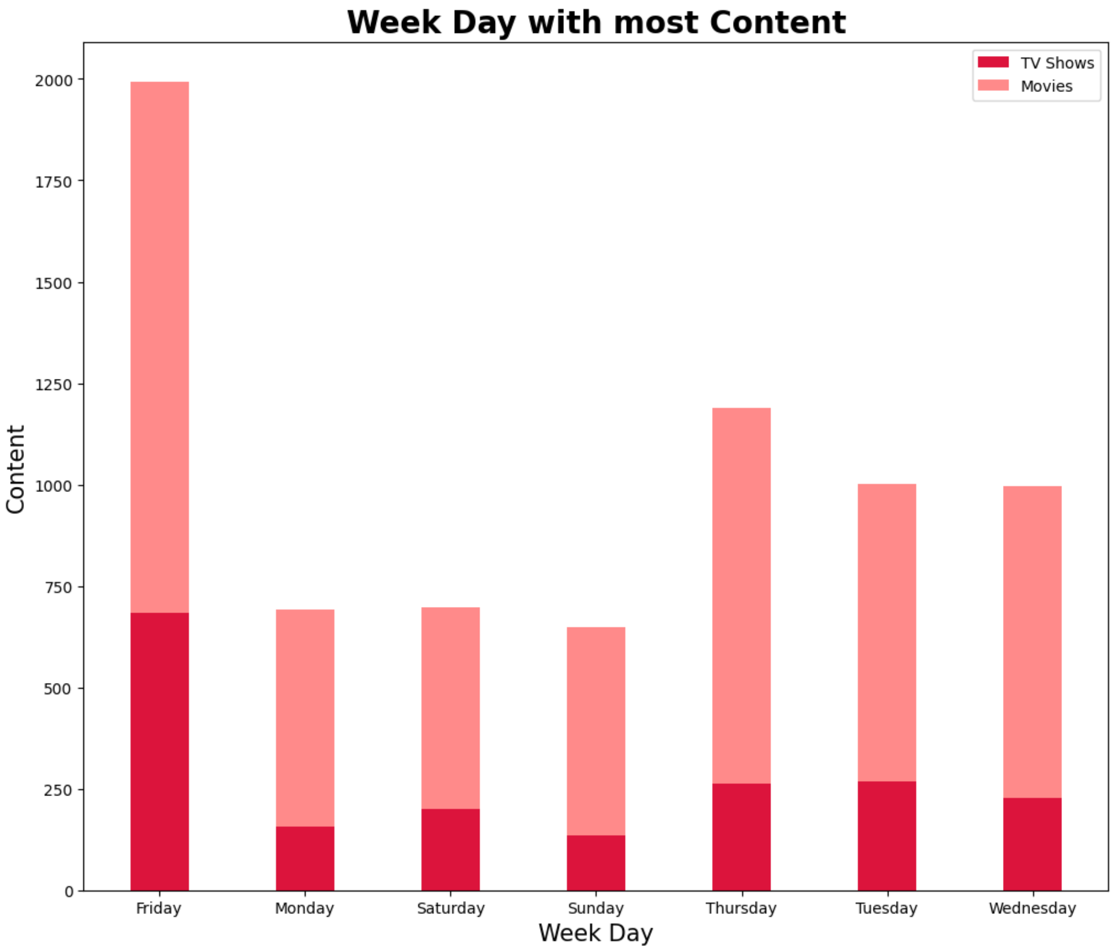
```
C:\Users\ashis\AppData\Local\Temp\ipykernel_21692\2212942706.py:1:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped =
netflix['day_added'].groupby(netflix['type']).value_counts().unstack('type')
```

```
[52]: type        Movie  TV Show
      day_added
      Friday       1308      684
      Monday        533      158
      Saturday      495      201
      Sunday        514      135
      Thursday      924      264
      Tuesday       734      267
      Wednesday     769      227
```

Plotting bar graph to analyze the week day having most content:

```
[53]: plt.figure(figsize = (15,5))
      fig = plt.subplots(figsize =(12, 10))
      p1 = plt.bar(grouped.index, grouped['TV Show'], 0.4, color='#DC143C')
      p2 = plt.bar(grouped.index, grouped['Movie'], 0.4,
                   bottom = grouped['TV Show'], color='#FF8A8A')
      plt.ylabel('Content',fontsize = 15)
      plt.xlabel('Week Day',fontsize = 15)
      plt.title('Week Day with most Content',fontsize = 20, fontweight = 'bold')
      plt.legend((p1[0], p2[0]), ('TV Shows', 'Movies'))
      plt.show()
```

```
<Figure size 1500x500 with 0 Axes>
```



Conclusion: * Large number of content added on Thursday after Friday. * On average Netflix added more movies on Friday and Thursday. * On an average significant number of TV Show are added on Friday than other days.

1.6.15 5.15. The count of content added for Top 20 countries per year.

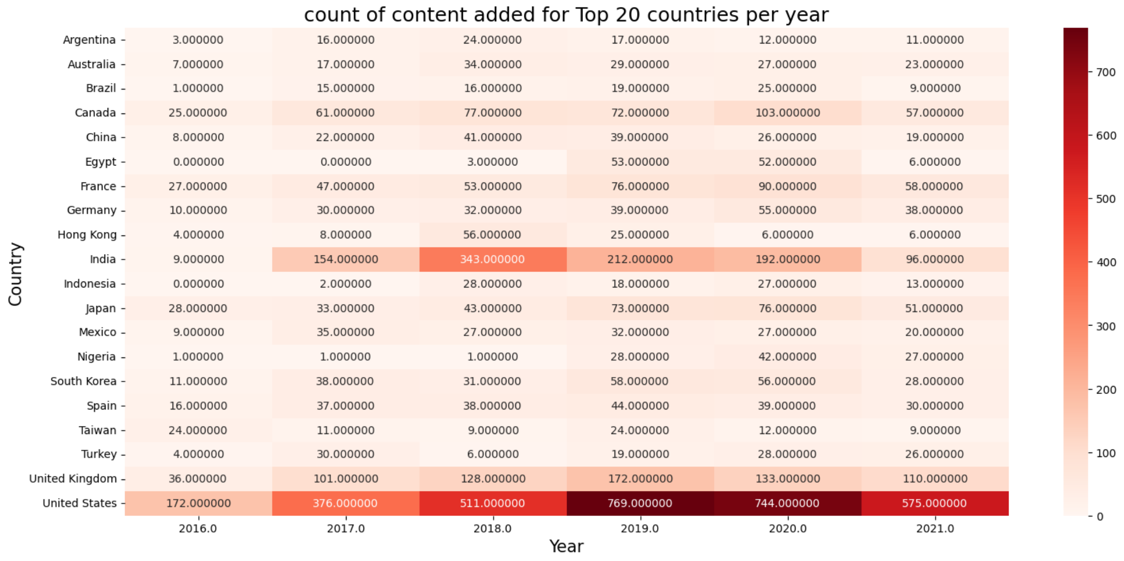
Getting the amount of netflix content of top 20 countries per year:

[54]:
```
country_data = explode_data(netflix, 'country', 'country', 'year_added')
country_data['country']=country_data['country'].str.strip()
#As netflix content is grown after 2015, so we are considering that data
country_data = country_data[country_data['year_added']>2015]
country_data = (
    country_data
    [country_data['country'].isin(country_data['country'].value_counts().
 ↪head(20).index)]
    .value_counts()
    .reset_index()
    .rename(columns={0:'count'})
    .pivot(index='country', columns = 'year_added', values='count')
    .fillna(0)
    )
country_data
```

[54]:
```
year_added      2016.0  2017.0  2018.0  2019.0  2020.0  2021.0
country
Argentina          3.0    16.0    24.0    17.0    12.0    11.0
Australia          7.0    17.0    34.0    29.0    27.0    23.0
Brazil             1.0    15.0    16.0    19.0    25.0     9.0
Canada            25.0    61.0    77.0    72.0   103.0    57.0
China              8.0    22.0    41.0    39.0    26.0    19.0
Egypt              0.0     0.0     3.0    53.0    52.0     6.0
France            27.0    47.0    53.0    76.0    90.0    58.0
Germany           10.0    30.0    32.0    39.0    55.0    38.0
Hong Kong          4.0     8.0    56.0    25.0     6.0     6.0
India              9.0   154.0   343.0   212.0   192.0    96.0
Indonesia          0.0     2.0    28.0    18.0    27.0    13.0
Japan             28.0    33.0    43.0    73.0    76.0    51.0
Mexico             9.0    35.0    27.0    32.0    27.0    20.0
Nigeria            1.0     1.0     1.0    28.0    42.0    27.0
South Korea       11.0    38.0    31.0    58.0    56.0    28.0
Spain             16.0    37.0    38.0    44.0    39.0    30.0
Taiwan            24.0    11.0     9.0    24.0    12.0     9.0
Turkey             4.0    30.0     6.0    19.0    28.0    26.0
United Kingdom    36.0   101.0   128.0   172.0   133.0   110.0
United States    172.0   376.0   511.0   769.0   744.0   575.0
```

Plotting a heatmap of amount of content for the top 20 countries for years 2017-2021:

[55]:
```
#plotting heatmap
plt.figure(figsize=(18,8))
ax = sns.heatmap(country_data, annot=True, fmt="f", cmap='Reds')
plt.ylabel('Country',fontsize = 15)
plt.xlabel('Year',fontsize = 15)
plt.title('count of content added for Top 20 countries per year',fontsize = 18)
plt.show()
```



Conclusion: * Top 5 countries where netflix is adding more content per year except United States are India, United Kingdom, Canada, France and Japan.

1.6.16 5.16. What are the most popular genres added mostly per year on Netflix?

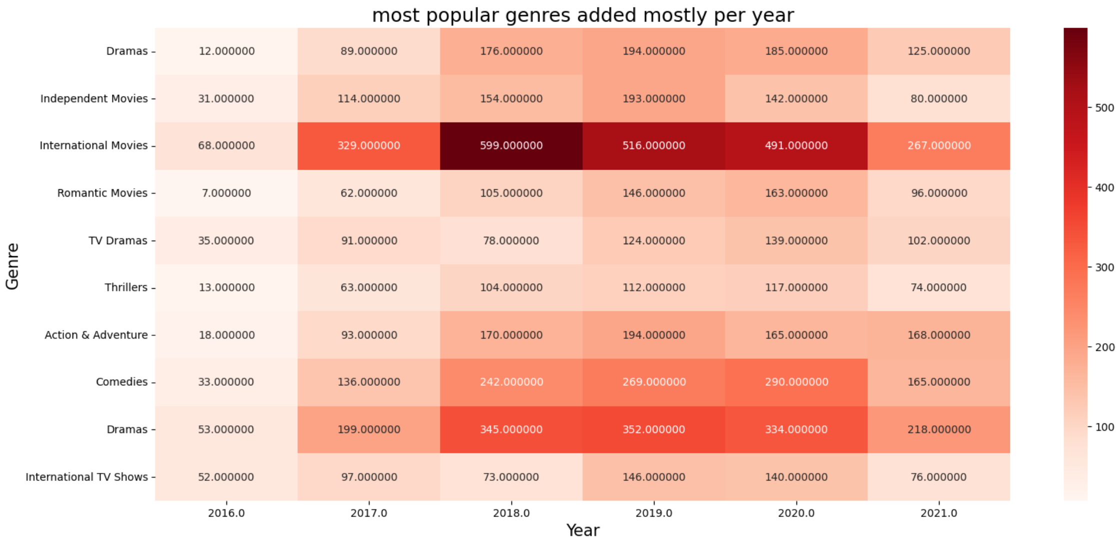
Displaying the count of overall content according to different genres from 2016 to 2021:

[56]:
```
genre_data = explode_data(netflix, 'listed_in', 'listed_in', 'year_added')
#As netflix content is grown after 2015, so we are considering that data
genre_data = genre_data[genre_data['year_added']>2015]
genre_data = (
    genre_data
    [genre_data['listed_in'].isin(genre_data['listed_in'].value_counts().
 ↪head(10).index)]
    .value_counts()
    .reset_index()
    .rename(columns={0:'count'})
    .pivot(index='listed_in', columns='year_added', values='count')
    .fillna(0)
    )
genre_data
```

[56]:
```
year_added              2016.0  2017.0  2018.0  2019.0  2020.0  2021.0
listed_in
 Dramas                     12      89     176     194     185     125
 Independent Movies         31     114     154     193     142      80
 International Movies       68     329     599     516     491     267
 Romantic Movies             7      62     105     146     163      96
 TV Dramas                  35      91      78     124     139     102
 Thrillers                      13      63     104     112     117      74
Action & Adventure              18      93     170     194     165     168
Comedies                        33     136     242     269     290     165
Dramas                          53     199     345     352     334     218
International TV Shows          52      97      73     146     140      76
```

Plotting heatmap for the most popular genres added mostly per year:

[57]:
```python
plt.figure(figsize=(18,8))
ax = sns.heatmap(genre_data, annot=True, fmt="f", cmap='Reds')
plt.ylabel('Genre',fontsize = 15)
plt.xlabel('Year',fontsize = 15)
plt.title('most popular genres added mostly per year',fontsize = 18)
plt.show()
```

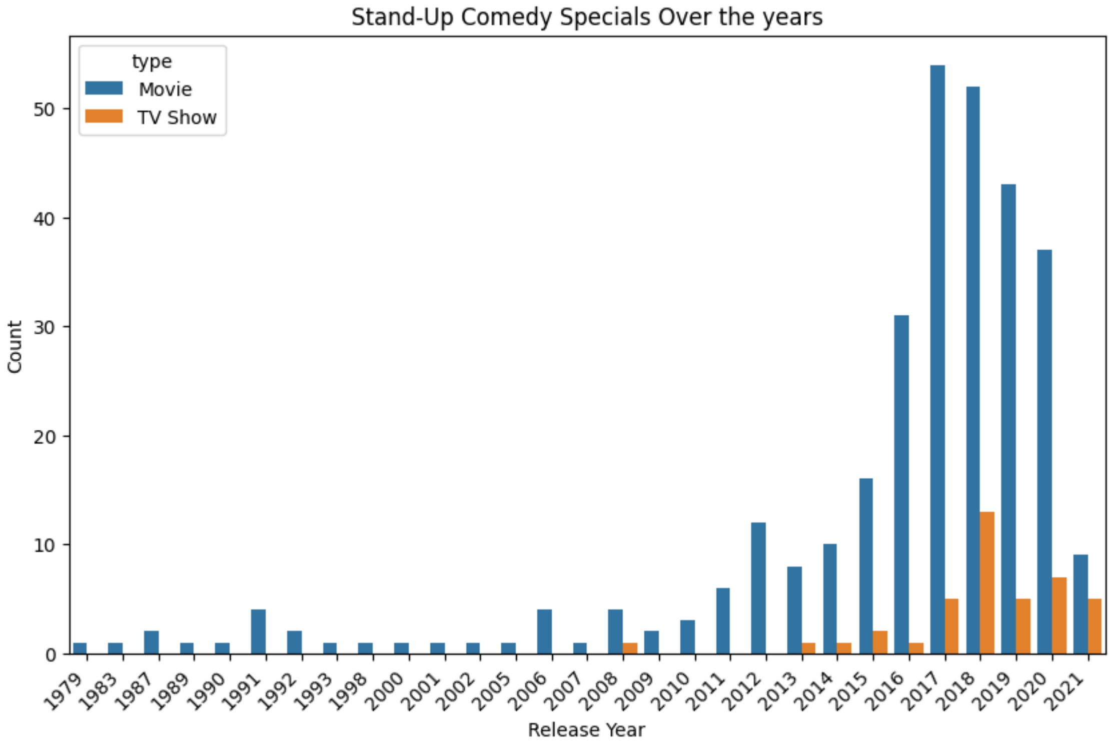


Conclusion: * Five most popular genres in recent years are International movies, Dramas, Comedies, International TV Shows and Action & Adventure.

1.6.17 5.17. Analysis of Trend of Stand-Up Comedy Over the Years.

[58]:

```
# Analysis of Stand-Up Comedy Specials:
plt.figure(figsize=(10,6))
standup_comedy = netflix[netflix['listed_in'].str.contains('Stand-Up Comedy',na=False)]
sns.countplot(data=standup_comedy, x='release_year', hue='type')
plt.title('Stand-Up Comedy Specials Over the years')
plt.xticks(rotation=45, ha="right")
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.show()
```



[59]:

```
# Prints the data
print("Stand-Up Comedy Specials Over the Years:")
standup_comedy.groupby(['release_year','type']).size().unstack().fillna(0)
```

```
Stand-Up Comedy Specials Over the Years:
```

```
C:\Users\ashis\AppData\Local\Temp\ipykernel_21692\4028681054.py:3:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  standup_comedy.groupby(['release_year','type']).size().unstack().fillna(0)
```

[59]:

```
type          Movie  TV Show
release_year
1979              1        0
1983              1        0
1987              2        0
1989              1        0
1990              1        0
1991              4        0
1992              2        0
1993              1        0
1998              1        0
2000              1        0
2001              1        0
2002              1        0
2005              1        0
2006              4        0
2007              1        0
2008              4        1
2009              2        0
2010              3        0
2011              6        0
2012             12        0
2013              8        1
2014             10        1
2015             16        2
2016             31        1
2017             54        5
2018             52       13
2019             43        5
2020             37        7
2021              9        5
```

Conclusion: * Stand-Up Comedy movies sharply increases after 2009 and more gradually after 2015. * Stand-Up Comedy in TV Show is less as compared to movies but it comes into play 2013 onwards.

1.6.18 5.18. Identify the genres that often co-occur in Netflix content and the Directors names who directed both Movie and TV Shows.

a) Identify the genres that often co-occur in Netflix content:

[60]:

```
from itertools import combinations
genre_combinations = netflix['listed_in'].str.split(', ').apply(lambda x:␣
  ↪list(combinations(x, 2)))
genre_combinations = [genre for sublist in genre_combinations for genre in␣
  ↪sublist]
common_genre_pairs = pd.Series(genre_combinations).value_counts().head(5)
print(common_genre_pairs)
```

```
(Dramas, International Movies)         1380
(Comedies, International Movies)        748
(Dramas, Independent Movies)            577
(Comedies, Dramas)                      484
(International TV Shows, TV Dramas)     416
Name: count, dtype: int64
```

b) Identify the directors who have worked on both movies and TV shows:

[61]:
```
directors_both = netflix.groupby('director')['type'].nunique()
directors_both = directors_both[directors_both > 1].index.tolist()
print(directors_both)
```

```
['BB Sasore', 'Eli Roth', 'Jay Chandrasekhar', 'Jerry Seinfeld', 'Kemi Adetiba',
'Kobun Shizuno, Hiroyuki Seshita', 'Kyran Kelly', 'Marcus Raboy', 'Mark
Tonderai', 'Masaaki Yuasa', 'Mateo Gil', 'Michael Simon', 'Noam Murro', 'Not
Known', 'Obi Emelonye', 'Oliver Stone', 'Ryan Polito', 'Sion Sono', 'Soumendra
Padhi', 'Stan Lathan', 'Tensai Okamura', 'Thomas Astruc', 'Tosin Coker',
'Tsutomu Mizushima', 'Ziad Doueiri']
```

Conclusion: * It gives the genres that comes often together means that the content belonging to netflix having such a genre will also belong to other one. * It gives the name of directors who worked on both type of content Movies and TV Shows.

1.7 Summary

- **Netflix added more Movies than TV Shows.**
- **After 2019 there is a drop in content added. Drop in Movie content is more than TV Show content. It was due to Covid.**
- **If we not consider the content added in 2021 as we don't have data for all months then more content is added in October, November and December.**
- **More content for adults is there on Netflix.**
- **Most of the content is added on Friday and Thursday, respectively.**
- **United states has added most content on Netflix.**
- **Top 5 countries where netflix is adding more content per year except United States are India, United Kingdom, Canada, France and Japan.**
- **Five most popular genres in recent years are International movies, Dramas, Comedies, International TV Shows and Action & Adventure.**

1.8 Movie

- **Almost same count of movies added on Netflix monthly.**
- **In countries like India, Spain, Germany Movies are more popular than TV Shows.**
- **Movies are of duration around 100 minutes.**
- **Top 5 countries where movies added are United States, India, United Kingdom, Canada and France.**
- **Top 3 genres in Movies are International Movies, Dramas and Comedies.**

1.9 TV Show

- **Large number of TV Show added on Friday than other weekday.**
- **TV Shows have mostly season 1 and season 2 respectively.**
- **In countries like United Kingdom, Japan, South Korea, Taiwan TV Shows are more popular than Movies.**
- **Top 5 countries where movies added are United States, United Kingdom, Japan, South Korea and Canada.**
- **Top 3 genres in Movies are International TV Shows, TV Dramas and TV Comedies.**

1.10 Recommendations:

1.10.1 Movie

- **Netflix should be focusing on adding more movies in emerging countries like India, United Kingdom, Canada and France for Adult audience.**
- **Preferred Duration of movies will be from 80-120 minutes.**
- **International Movies, Dramas, Comedies should be the preferred genres for Movies.**

1.10.2 TV Show

- **For TV Shows Netflix should focus on countries like Japan, South Korea, Canada and France.**
- **TV Show seasons can be up to 3 preferably.**
- **International TV Shows, TV Dramas, TV Comedies should be the preferred genres for TV Shows.**