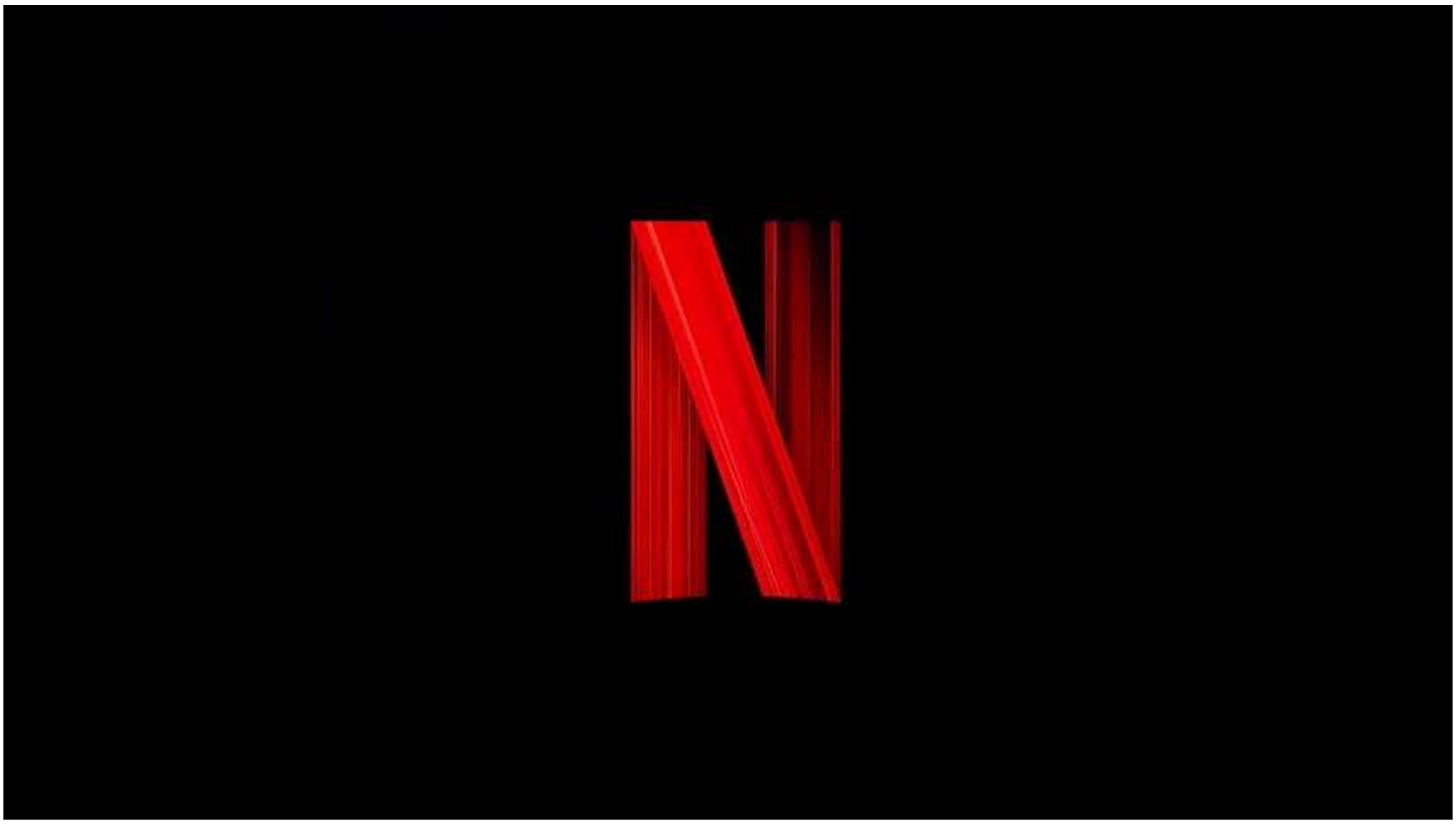


🌎 ABOUT:

Netflix, Inc. is an American streaming service and production company, offering a vast library of films and TV series, including Netflix Originals. As of March 31, 2023, it has 232.5 million paid memberships in over 190 countries. Founded by Reed Hastings and Marc Randolph in 1997, Netflix started as a DVD rental business before shifting to streaming in 2007, transforming home entertainment.

Features of the dataset:

- **Show_id:** Unique ID for every Movie / Tv Show
- **Type:** Identifier - A Movie or TV Show
- **Title:** Title of the Movie / Tv Show
- **Director:** Director of the Movie
- **Cast:** Actors 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:** TV Rating of the movie/show
- **Duration:** Total Duration - in minutes or number of seasons
- **Listed_in:** Genre
- **Description:** The summary description

In [1]:
```
!wget "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv" -O netflix.csv # downloading the data
```

```
--2024-06-28 19:16:50--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 13.224.9.24, 13.224.9.129, 13.224.9.181, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|13.224.9.24|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3399671 (3.2M) [text/plain]
Saving to: ‘netflix.csv’

netflix.csv         100%[===================>]   3.24M  --.-KB/s    in 0.08s

2024-06-28 19:16:50 (41.4 MB/s) - ‘netflix.csv’ saved [3399671/3399671]
```

The dataset provided to us consists of a list of all the TV shows/movies available on Netflix

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
data = pd.read_csv("netflix.csv")
```

In [4]:
```
data.head(3)
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...

In [5]:
```
data.tail(3)
```

Out[5]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 min	Comedies, Horror Movies	Looking to survive in a world taken over by zo...
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG	88 min	Children & Family Movies, Comedies	Dragged from civilian life, a former superhero...
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

In [6]:
```
data.shape # Shape of the dataset
```

Out[6]:
```
(8807, 12)
```

In [7]:
```
data.info() # Checking the data type and non null cout of every columns in the data set.
```

```
<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
```

🔍 Insights:

- From the analysis above, we can see that the dataset.

contains 12 features with a mix of alphanumeric data.

- Additionally, five of the columns have missing data.
- There are a significant number of missing values in the "cast" and "director" columns.
- The "type of rating" and "date_added" columns are currently classified as "object" data types, but they should be converted to categorical and datetime types, respectively.

🧬 Statistical Summary

In [8]:
```
data.describe() # statistical summary of numerical data
```

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

🔍 Insights:

- The "release_year" column has 8,807 entries, with an average year of 2014. The release years range from 1925 to 2021.
- Most releases are recent, with 25% before 2013, 50% before 2017, and 75% before 2019.

In [9]:
```
data.describe(include = object) # statistical summary of categorical data
```

Out[9]:

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

🔍 Insights:

- The dataset includes 8,807 entries, primarily consisting of movies (6,131).
- Rajiv Chilaka and David Attenborough appear most frequently as director and cast member, respectively.
- Most entries are from the United States, with January 1, 2020, being the most common date added.

🕵️ Detecting Missing Values

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

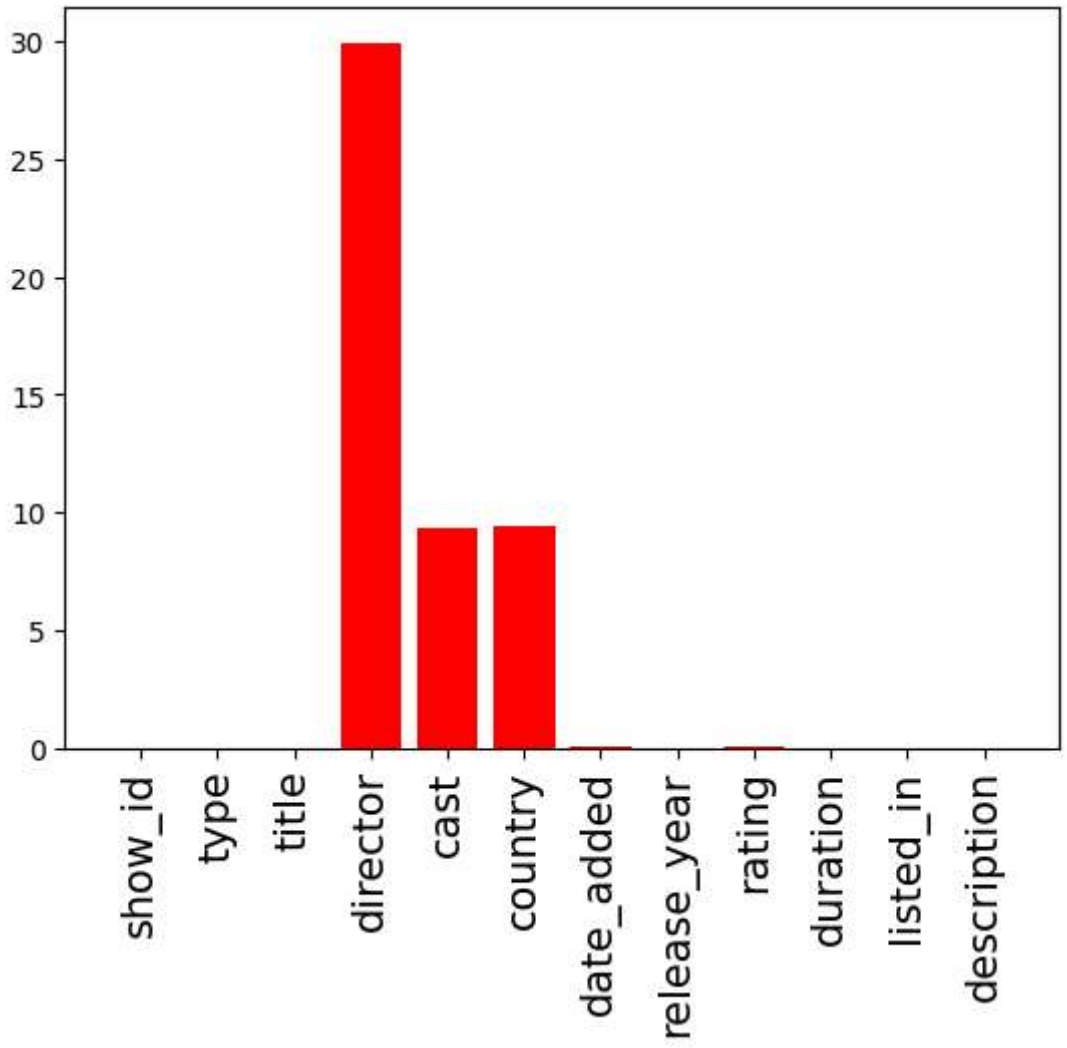
Out[10]:
```
show_id          0
type             0
title            0
director      2634
cast           825
country        831
date_added      10
release_year     0
rating           4
duration         3
listed_in        0
description      0
dtype: int64
```

In [11]:
```
def null_rate(df,col):
  null_rate = (df[col].isnull().sum() / df.shape[0])*100
  return round(null_rate,2)

null_rate(data, data.columns)
```

Out[11]:
```
show_id         0.00
type            0.00
title           0.00
director       29.91
cast            9.37
country         9.44
date_added      0.11
release_year    0.00
rating          0.05
duration        0.03
listed_in       0.00
description     0.00
dtype: float64
```

In [12]:
```
x = null_rate(data, data.columns)
plt.bar(data.columns,x,color = "red")
plt.xticks(rotation = 90, fontsize = 15)
plt.show()
```



🔍 **Insights:**

- Our analysis shows six columns with missing values: 'director' has the most, followed by 'cast' and 'country'. 'Date added', 'ratings', and 'duration' have significantly fewer missing values (<1%).

📜Filling the missing Values

In [13]:
```
data["director"].fillna("Unknown Director", inplace = True)
data["cast"].fillna("Unknown Cast", inplace = True)
data["country"].fillna("Unknown Country", inplace = True)
```

In [14]:
```
data.isnull().sum()
```

Out[14]:
```
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
```

In [15]:
```
data["rating"].value_counts()
```

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
NC-17          3
UR             3
74 min         1
84 min         1
66 min         1
Name: count, dtype: int64
```

In [16]:
```
data[(data["rating"] == "74 min") | (data["rating"] == "84 min") | (data["rating"] == "66 min")]
```

Out[16]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
5541	s5542	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	April 4, 2017	2017	74 min	NaN	Movies	Louis C.K. muses on religion, eternal love, gi...
5794	s5795	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	September 16, 2016	2010	84 min	NaN	Movies	Emmy-winning comedy writer Louis C.K. brings h...
5813	s5814	Movie	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	United States	August 15, 2016	2015	66 min	NaN	Movies	The comic puts his trademark hilarious/thought...

🔁 Replacing values of duration column :

- Move the last three values (74 min, 84 min, 66 min) from the 'rating' column to the 'duration' column, as the dataset shows the 'duration' column missing exactly 3 values, matching the respective rows.

In [ ]:
```
data["duration"][[5541,5794,5813]] = data["rating"][[5541,5794,5813]] # moving wrong column data to the right one
```

In [18]:
```
data["duration"][[5541,5794,5813]]
```

Out[18]:
```
5541    74 min
5794    84 min
5813    66 min
Name: duration, dtype: object
```

In [ ]:
```
data["rating"][[5541,5794,5813]] = "Unknown Rating" # replacing existing wrong rating columns with unknown rating
```

In [20]:
```
data["rating"][[5541,5794,5813]]
```

Out[20]:
```
5541    Unknown Rating
5794    Unknown Rating
5813    Unknown Rating
Name: rating, dtype: object
```

In [21]:
```
data["rating"].fillna("Unknown Rating", inplace = True)
```

In [22]:
```
data.isnull().sum() # Checking to confirm changes
```

Out[22]:
```
show_id          0
type             0
title            0
director         0
cast             0
country          0
date_added      10
release_year     0
rating           0
duration         0
listed_in        0
description      0
dtype: int64
```

📅 Converting date_added Column for better analysis and categorical attributes to category

In [23]:
```
data = data.astype({"type": "category"})
```

In [24]:
```
data["date_added"] = data["date_added"].str.strip() #As there was a leading space in some of the "date_added" column values.
```

In [25]:
```
data["date_added"] = pd.to_datetime(data["date_added"])
data["date_added"].dtype
```

Out[25]:
```
dtype('<M8[ns]')
```

In [26]:
```
#Adding new columns year_added (year) , month_added(name of the month) , day_added(name of the day)

data["year_added"] = data["date_added"].dt.year
data["month_added"] = data["date_added"].dt.month_name()
data["day_added"] = data["date_added"].dt.day_name()
data["week_added"] = data["date_added"].dt.isocalendar().week

data.head(3)
```

Out[26]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	year_added	mont
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown Cast	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	2021.0	Se
1	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	2021.0	Se
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown Country	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	2021.0	Se

📂➡📄📄📄 Un-nesting the required Columns

- Unnesting columns 'director,' 'cast,' and 'country' allows each nested value to be expanded into individual rows for clearer visualization and analysis.

In [27]:
```
data_new = data.copy()
```

In [28]:
```
data_new.head(3)
```

Out[28]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	year_added	mont
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown Cast	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	2021.0	Se
1	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	2021.0	Se
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown Country	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	2021.0	Se

In [29]:
```
data_new["director"] = data_new["director"].str.split(", ")
data_new["cast"] = data_new["cast"].str.split(", ")
data_new["country"] = data_new["country"].str.split(", ")
```

In [30]:
```
data_new = data_new.explode("cast")
data_new = data_new.explode("director")
data_new = data_new.explode("country")
```

In [31]:
```
data_new.head(3)
```

Out[31]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	year_added	month_ac
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown Cast	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	2021.0	Septer
1	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	2021.0	Septer
1	s2	TV Show	Blood & Water	Unknown Director	Khosi Ngema	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	2021.0	Septer

In [32]:
```
data_new.shape
```

Out[32]:
```
(89382, 16)
```

In [33]:
```
data_new.drop_duplicates(keep = "first", inplace = True)
```

💡 Data Visualization & Analysis

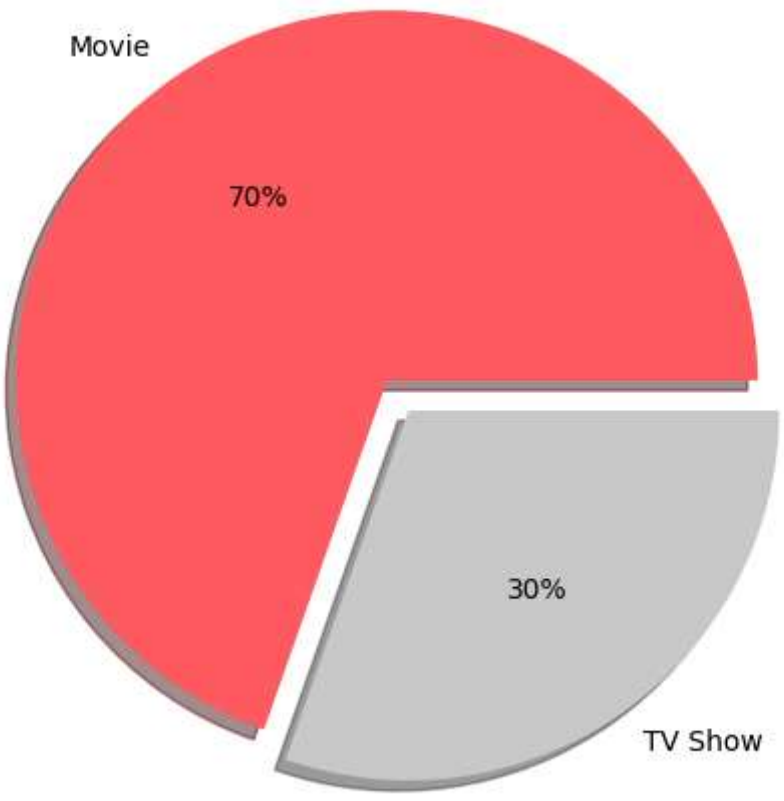
🎬 Content Distribution

In [34]:
```
cont_dis = data["type"].value_counts().reset_index()
cont_dis
```

Out[34]:

	type	count
0	Movie	6131
1	TV Show	2676

In [35]:
```
colors = ['#FF5A5F', '#CCCCCC']
sizes = cont_dis["count"]
labels = cont_dis["type"]
explode = (0.1,0)
plt.figure(figsize=(6,6))
plt.pie(sizes,labels = labels , explode = explode , colors = colors , autopct = "%1.f%%", shadow = True)
plt.show()
```



In [36]:
```
data_duration = data[["type","duration"]].copy()
```

In [37]:
```
data_duration["duration"] = data["duration"].apply(lambda x : x.split()[0]).astype("int")
```

In [38]:
```
data_duration.head()
```

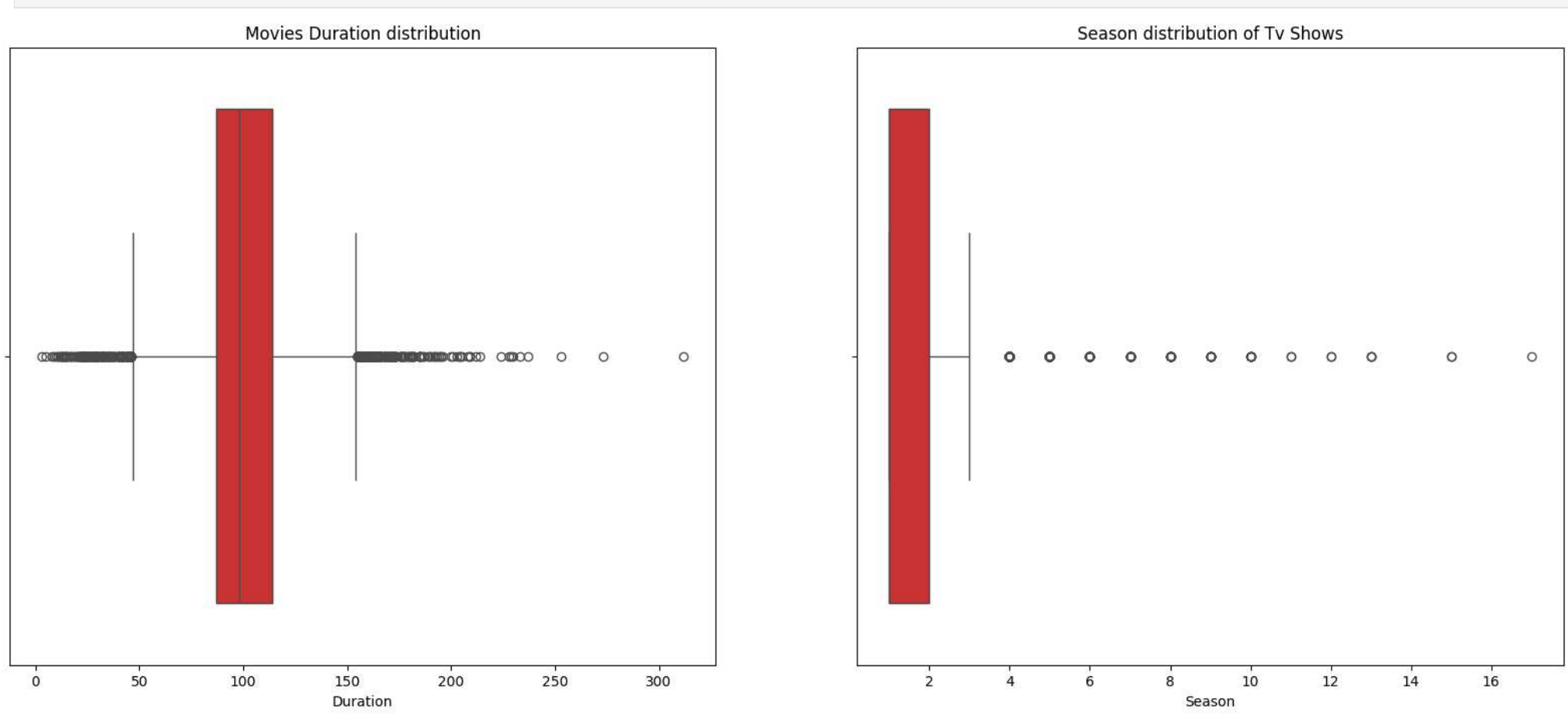
Out[38]:

	type	duration
0	Movie	90
1	TV Show	2
2	TV Show	1
3	TV Show	1
4	TV Show	2

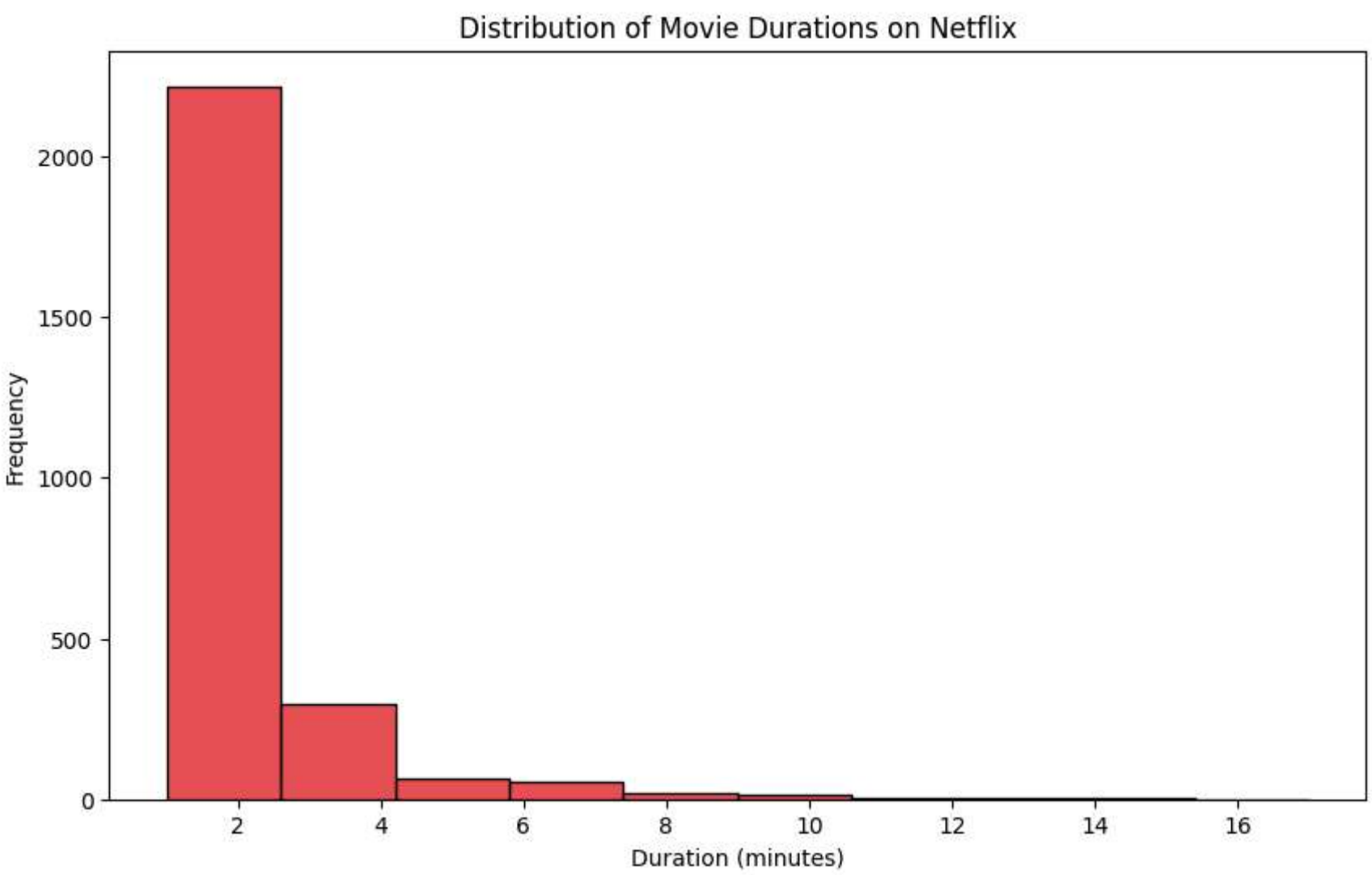
In [39]:
```
movie = data_duration[data_duration["type"] == "Movie"]
tv = data_duration[data_duration["type"] == "TV Show"]
```

🕐 Duration of Movies and TV Shows on Netflix

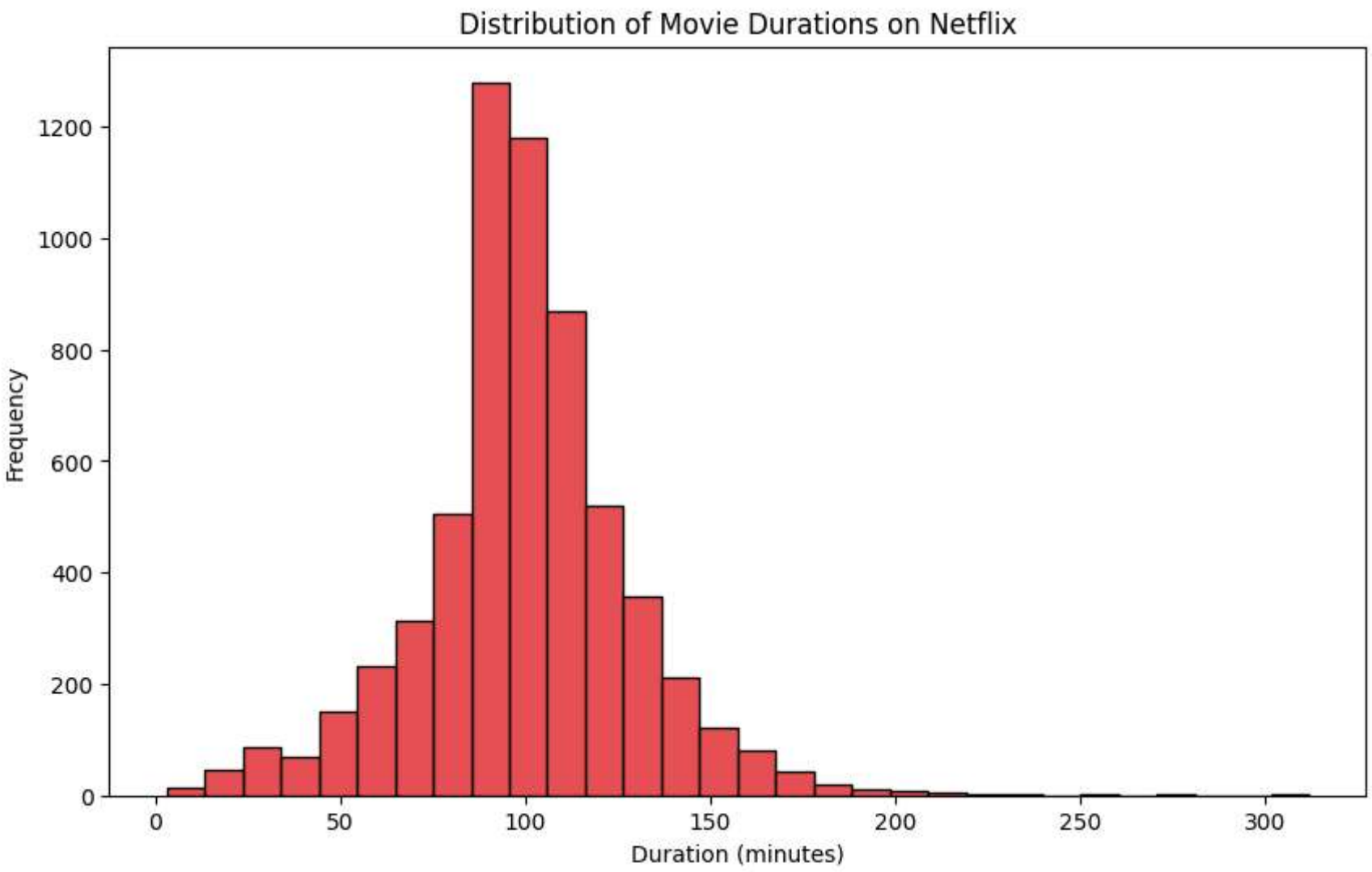
In [40]:
```
sns.set_palette("Set1")
plt.figure(figsize = (20,8))
plt.subplot(1,2,1)
sns.boxplot(x = "duration" , data = movie )
plt.title("Movies Duration distribution")
plt.xlabel("Duration")
plt.subplot(1,2,2)
sns.boxplot(x = "duration" , data = tv )
plt.xlabel("Season")
plt.title("Season distribution of Tv Shows")
plt.show()
```



In [41]:
```
sns.set_palette("Set1")
plt.figure(figsize=(10, 6))
sns.histplot(x = 'duration',data = tv, bins=10, kde=False, edgecolor='black')
plt.title('Distribution of Movie Durations on Netflix')
plt.xlabel('Duration (minutes)')
plt.ylabel('Frequency')
plt.show()
```



In [42]:
```
sns.set_palette("Set1")
plt.figure(figsize=(10, 6))
sns.histplot(x = 'duration',data = movie, bins=30, kde=False, edgecolor='black')
plt.title('Distribution of Movie Durations on Netflix')
plt.xlabel('Duration (minutes)')
plt.ylabel('Frequency')
plt.show()
```



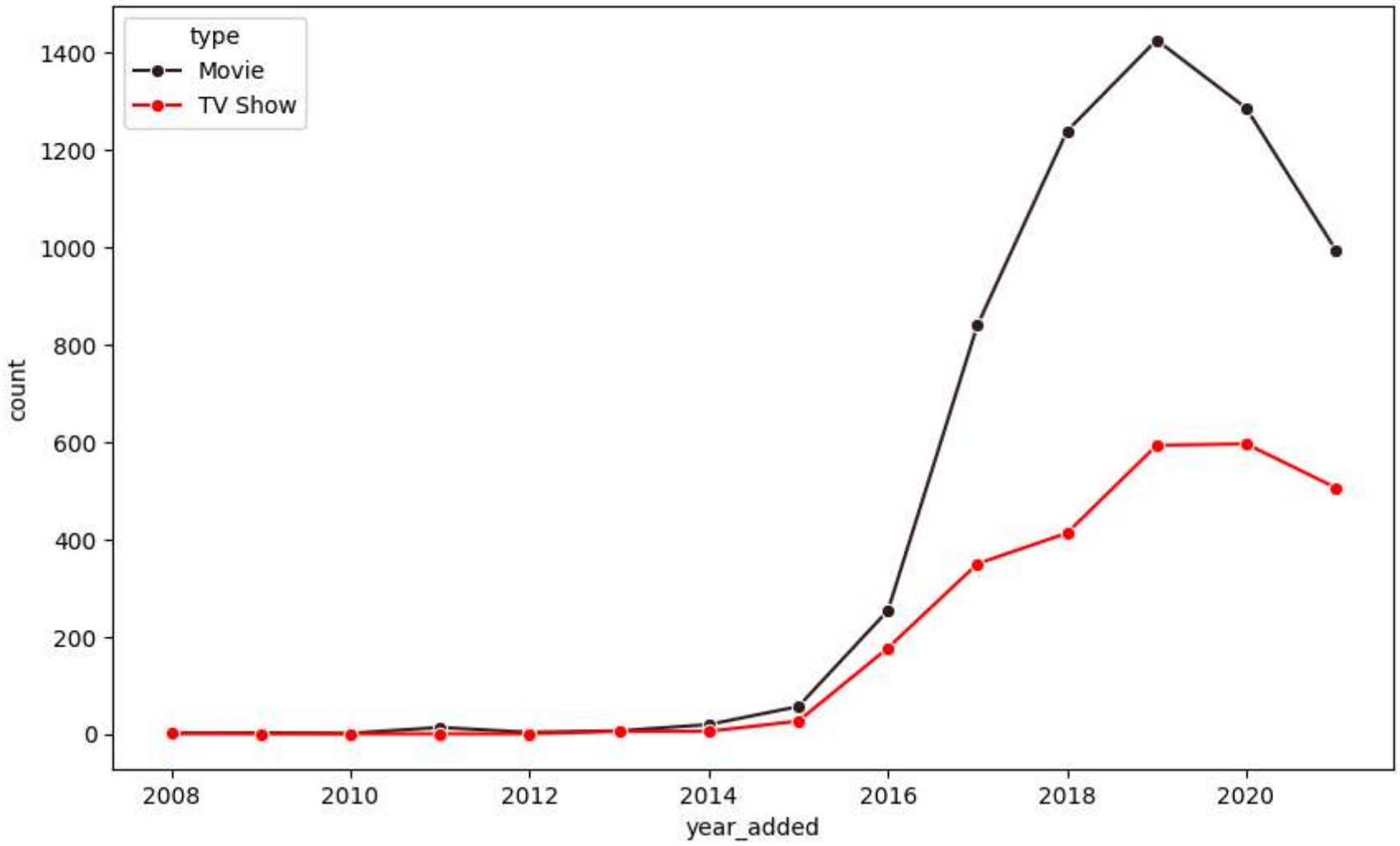
🔍 Insights:

- Movies typically have an average duration of around 100 minutes, with more outliers compared to TV shows.
- TV shows usually have 1 or 2 seasons.

📅 Movies and Tv Shows Added and Released Over time

In [43]:
```
yearwise_added = data.groupby("type")["year_added"].value_counts().reset_index()
```

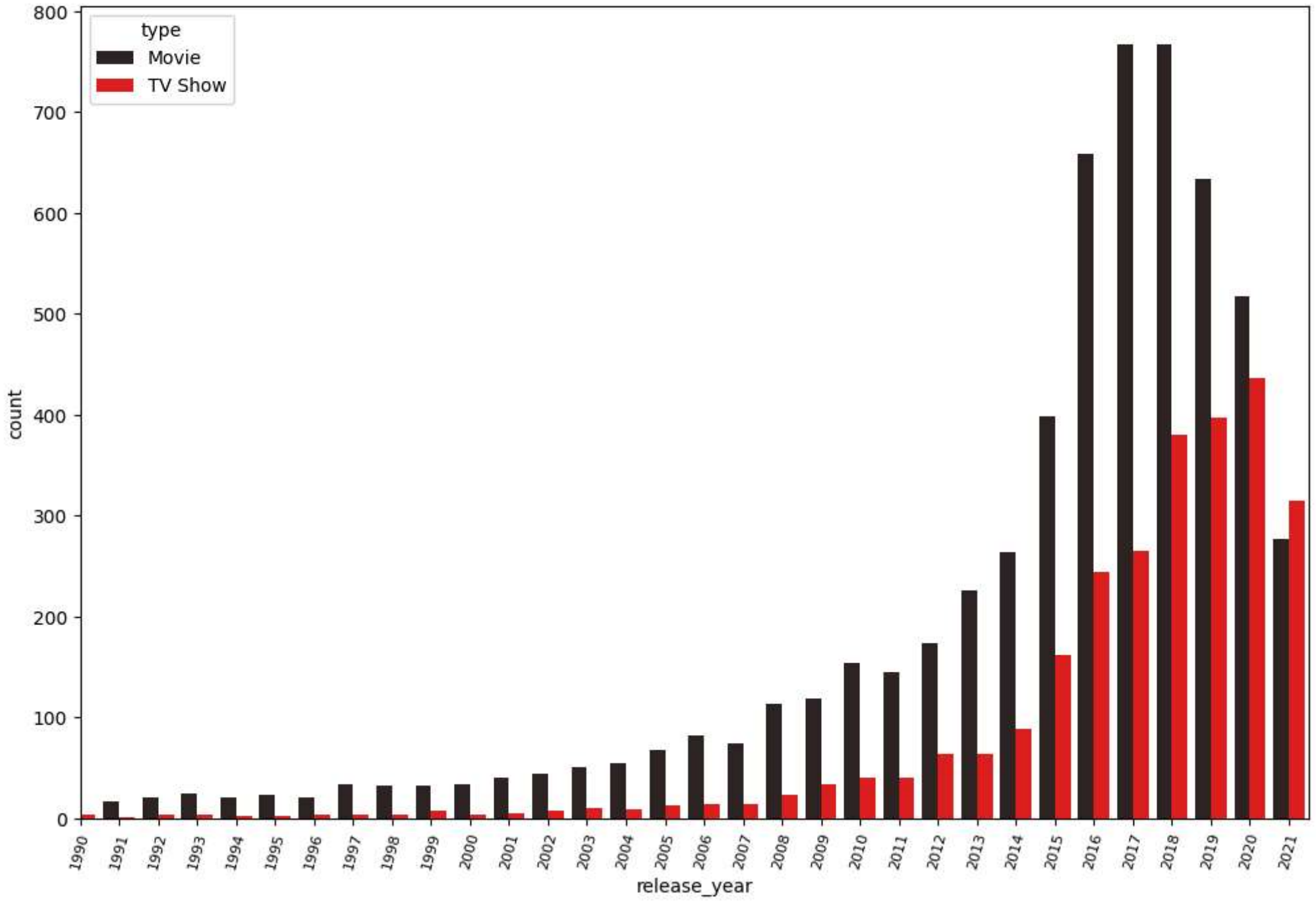
In [44]:
```
plt.figure(figsize=(10,6))
sns.lineplot( x = "year_added", y = "count", hue = "type" ,data = yearwise_added, palette = "dark:red", marker = "o")
plt.suptitle("Movies and Tv Shows Added Over time")
plt.show()
```



🔍 **Insights:**

- Netflix's growth was gradual over several years, with a noticeable uptick starting in 2015. From 2016 onwards, there was a significant surge.

In [45]:
```
plt.figure(figsize=(12,8))
sns.countplot( x = "release_year", hue = "type" ,data = data, palette = "dark:red")
left,right = plt.xlim()
plt.xlim(left = "1990" ,right = right)
plt.xticks(rotation = 75, fontsize = 8)
plt.suptitle("Movies and Tv Shows Released Over time",fontsize= 20)
plt.show()
```



🔍 **Insights:**

- TV Shows Peak in 2020:
- 2020 had the most TV shows released, followed by 2019 and 2021, driven by pandemic-induced demand.
- Movie Trends Post-2015:
- Movie releases increased after 2015, but dropped significantly in 2021, likely due to COVID-19 impacts.

🎬 Directors with the Most Movies or TV shows

In [46]:
```
# We'll use the unnested data frame for this visualization.
# Creating a dataframe for top directors.
data_dir = data_new.groupby("director")["title"].nunique().sort_values(ascending = False).head(11).reset_index()
```

In [47]:
```
data_dir
```

Out[47]:

	director	title
0	Unknown Director	2634
1	Rajiv Chilaka	22
2	Jan Suter	21
3	Raúl Campos	19
4	Marcus Raboy	16
5	Suhas Kadav	16
6	Jay Karas	15
7	Cathy Garcia-Molina	13
8	Jay Chapman	12
9	Martin Scorsese	12
10	Youssef Chahine	12

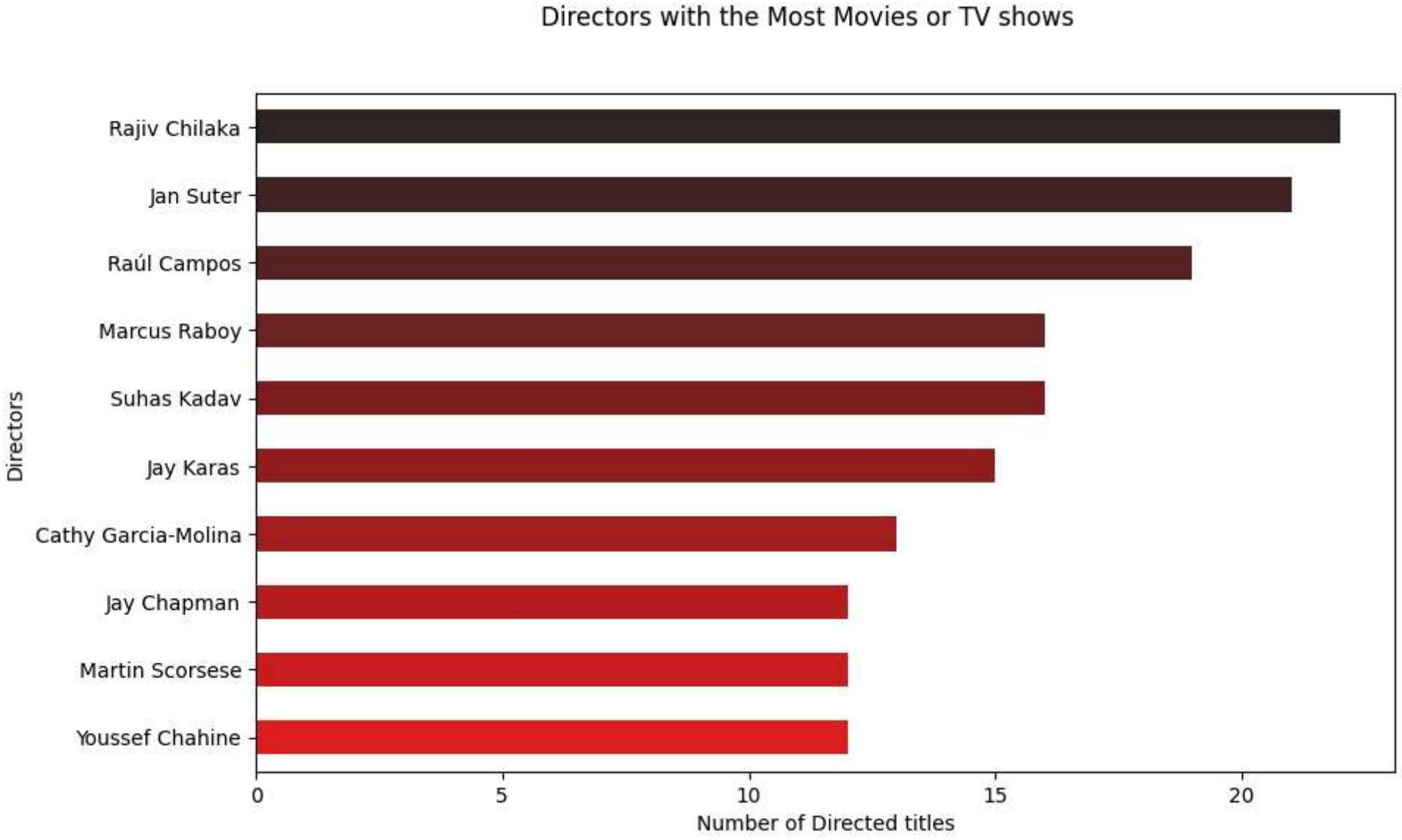
In [48]:
```
data_dir.drop(0 , inplace = True) # droping the unknown director row
```

In [49]:
```
data_dir # Will work with top 10 directors
```

Out[49]:

	director	title
1	Rajiv Chilaka	22
2	Jan Suter	21
3	Raúl Campos	19
4	Marcus Raboy	16
5	Suhas Kadav	16
6	Jay Karas	15
7	Cathy Garcia-Molina	13
8	Jay Chapman	12
9	Martin Scorsese	12
10	Youssef Chahine	12

In [50]:
```
plt.figure(figsize=(10,6))
sns.barplot(y = "director", x = "title", hue = "director" ,data = data_dir, legend = False, palette = "dark:red",  width=0.5)
plt.suptitle("Directors with the Most Movies or TV shows")
plt.ylabel("Directors")
plt.xlabel("Number of Directed titles")
plt.show()
```



🚶 Most Appeared Actors

In [51]:
```
# Creating a dataframe for top actors
data_act = data_new.groupby("cast")["title"].nunique().sort_values(ascending = False).reset_index().head(11)
data_act
```

Out[51]:

	cast	title
0	Unknown Cast	825
1	Anupam Kher	43
2	Shah Rukh Khan	35
3	Julie Tejwani	33
4	Naseeruddin Shah	32
5	Takahiro Sakurai	32
6	Rupa Bhimani	31
7	Om Puri	30
8	Akshay Kumar	30
9	Yuki Kaji	29
10	Paresh Rawal	28

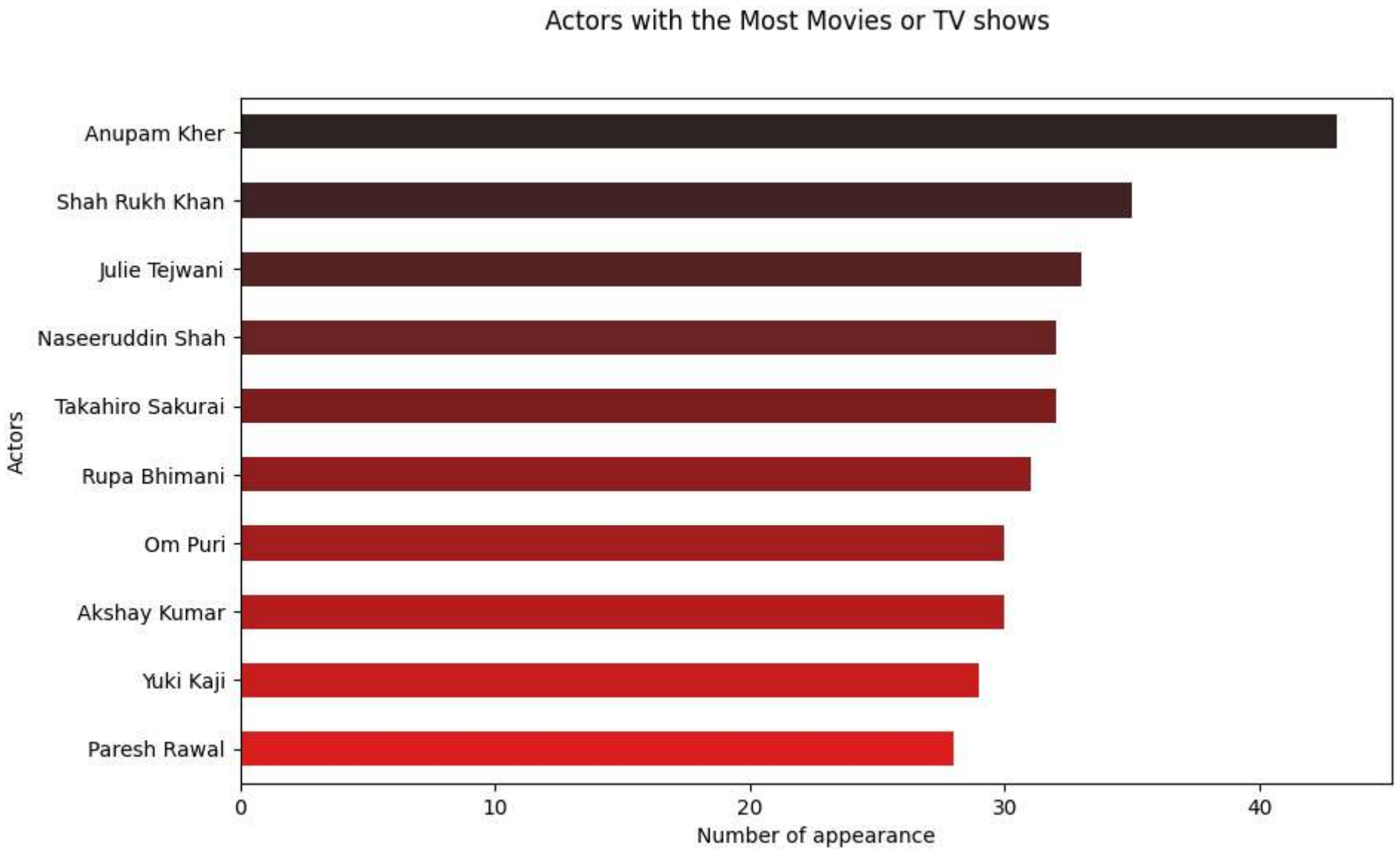
In [52]:
```
data_act.drop(0,inplace = True) # droping the unknown cast row
```

In [53]:
```
data_act # Will work with top 10 cast
```

Out[53]:

	cast	title
1	Anupam Kher	43
2	Shah Rukh Khan	35
3	Julie Tejwani	33
4	Naseeruddin Shah	32
5	Takahiro Sakurai	32
6	Rupa Bhimani	31
7	Om Puri	30
8	Akshay Kumar	30
9	Yuki Kaji	29
10	Paresh Rawal	28

In [54]:
```
plt.figure(figsize=(10,6))
sns.barplot(y = "cast", x = "title", hue = "cast" ,data = data_act, legend = False, palette = "dark:red",  width=0.5)
plt.suptitle("Actors with the Most Movies or TV shows")
plt.ylabel("Actors")
plt.xlabel("Number of appearance")
plt.show()
```



🔍 Insights:

- It is noticeable that the majority of actors are of Indian origin.

🌎 Overview of Movies and TV Shows by Country

In [55]:
```
# Creating a DataFrame for the Top Movie and Tv Shows Producing Countries Separately

data_movie = data_new[(data_new["type"] == "Movie") & (data_new["country"]!= "Unknown Country")] # Excluding unknown country data for movies
data_tv = data_new[(data_new["type"] == "TV Show") & (data_new["country"]!= "Unknown Country")] # Excluding unknown country data for TV Shows
```

In [56]:
```
data_movie_sort = data_movie.groupby("country")["title"].nunique().sort_values(ascending = False).reset_index()
data_movie_sort2 = data_movie_sort.head(10)
data_movie_sort2 # top 10 country producing movies
```

Out[56]:

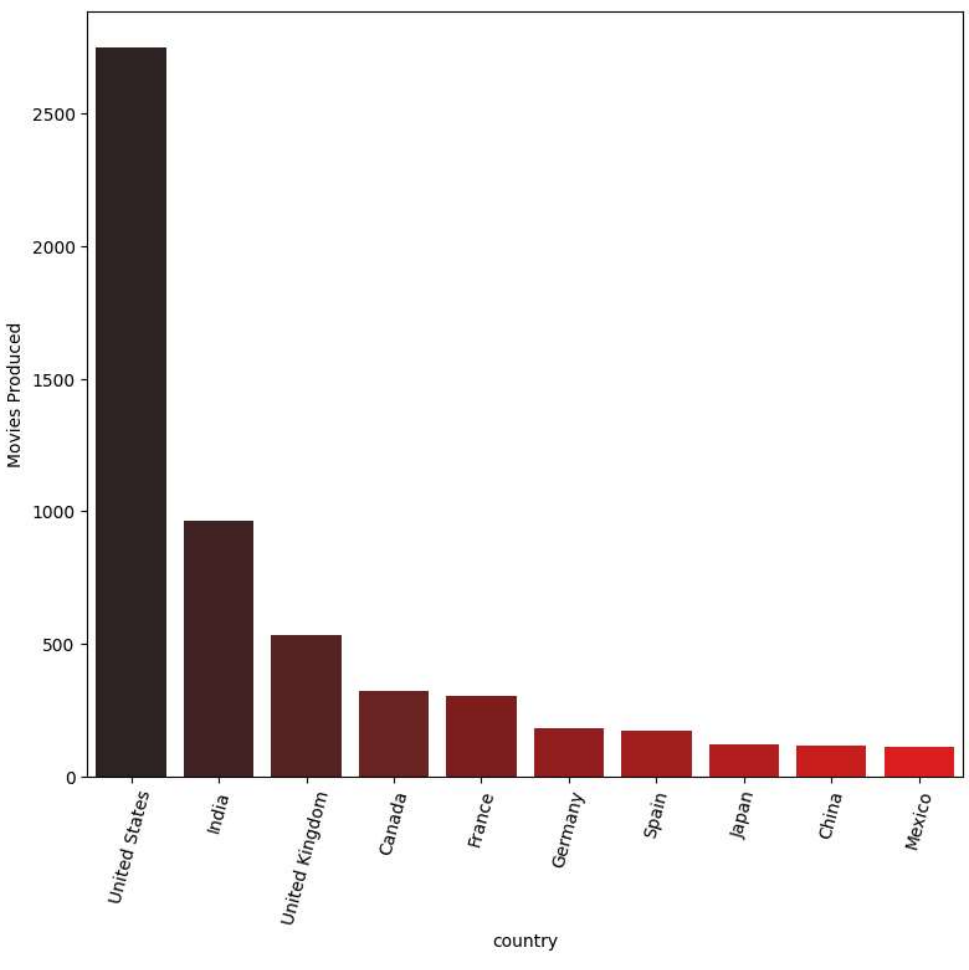
	country	title
0	United States	2751
1	India	962
2	United Kingdom	532
3	Canada	319
4	France	303
5	Germany	182
6	Spain	171
7	Japan	119
8	China	114
9	Mexico	111

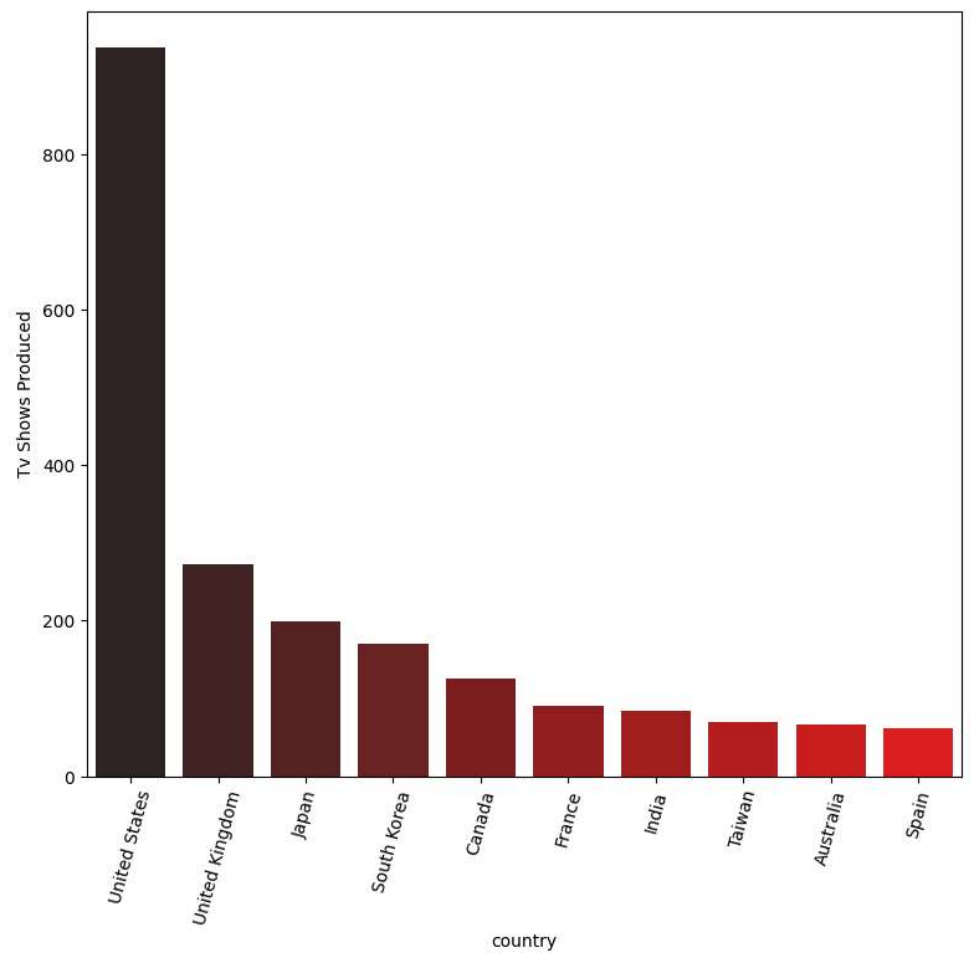
In [57]:
```
data_tv_sort = data_tv.groupby("country")["title"].nunique().sort_values(ascending = False).reset_index()
data_tv_sort2 = data_tv_sort.head(10)
data_tv_sort2 # top 10 country producing Tv Shows
```

Out[57]:

	country	title
0	United States	938
1	United Kingdom	272
2	Japan	199
3	South Korea	170
4	Canada	126
5	France	90
6	India	84
7	Taiwan	70
8	Australia	66
9	Spain	61

In [58]:
```
plt.figure(figsize = (20 , 8))
plt.subplot(1,2,1)
sns.barplot(x = "country", y = "title", hue = "country", legend = False ,data = data_movie_sort2 , palette = "dark:red")
plt.xticks(rotation = 75)
plt.ylabel("Movies Produced")
plt.subplot(1,2,2)
sns.barplot(x = "country", y = "title", hue = "country", legend = False ,data = data_tv_sort2 , palette = "dark:red")
plt.xticks(rotation = 75)
plt.ylabel("Tv Shows Produced")
plt.show()
```





🔍 **Insights:**

- **Strategic Content Investment:** Netflix emphasizes content

production in the USA and India, reflecting its focus on key markets for subscriber growth.

- **Diverse Global Content:** Netflix curates a diverse catalog with shows from the UK, Canada, France, Japan, etc., catering to varied cultural preferences worldwide.
- **Regional Viewing Preferences:** Preferences in India lean towards movies over TV shows, contrasting with South Korea's preference for TV shows, shaping Netflix's regional content strategies.
- **Cultural Relevance:** By featuring content from various countries, Netflix enhances cultural resonance and audience engagement globally.
- **Market Adaptation:** Netflix's strategy adapts to local preferences, leveraging regional content consumption habits to optimize user experience and retention.

Comparison between Movies 🎞 & Tv Shows 📺 On Country Basis

In [59]:
```
data_content = data_new[["country","type","title"]][data_new["country" ]!= "Unknown Country"]
```

In [60]:
```
data_content.duplicated().sum()
```

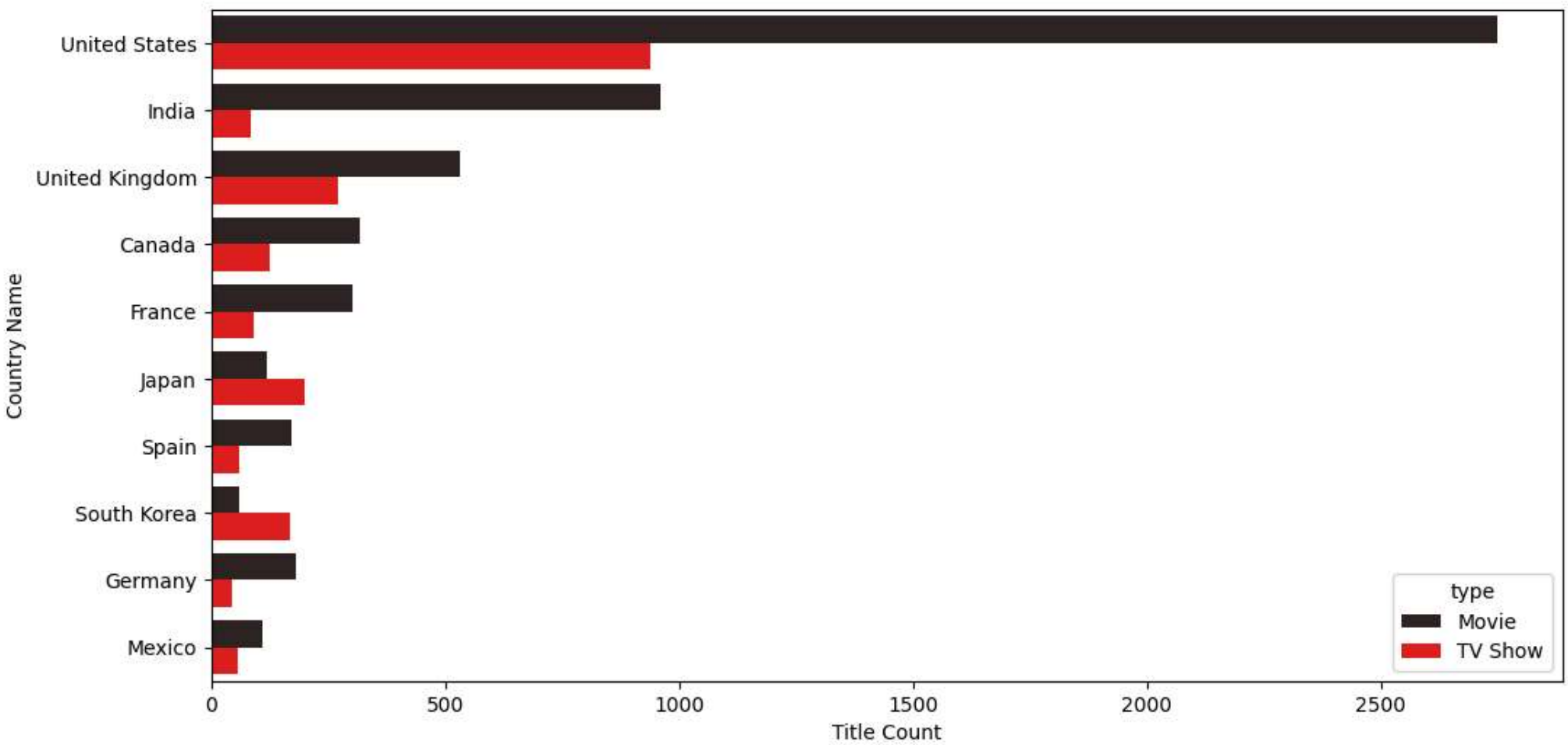
Out[60]: 73927

In [61]:
```
data_content.drop_duplicates(inplace = True)
```

In [62]:
```
top_country = data_content["country"].value_counts().reset_index().head(10)
```

In [63]:
```
data_content_f = data_content.merge(top_country , on = "country" , how = "right")
```

In [64]:
```
plt.figure(figsize = (12,6))
sns.countplot(y = "country", hue = "type", data= data_content_f , palette = "dark:red")
plt.ylabel("Country Name")
plt.xlabel("Title Count")
plt.suptitle("Movies And TV Shows Country Wise Comparision (TOP 10)",fontsize = 15)
plt.show()
```



🔍 **Insights:**

Content Preferences

- Asia: TV shows are more popular than movies, especially in South Korea and Japan.
- Europe: Movies are preferred over TV shows.
- India and North America: India has a strong preference for movies, while North American countries show a balanced interest in both movies and TV shows.

📅 Best Month to launch a TV Show / Movie?

In [65]:
```
data_month = data.groupby("month_added")["type"].value_counts().reset_index()
```

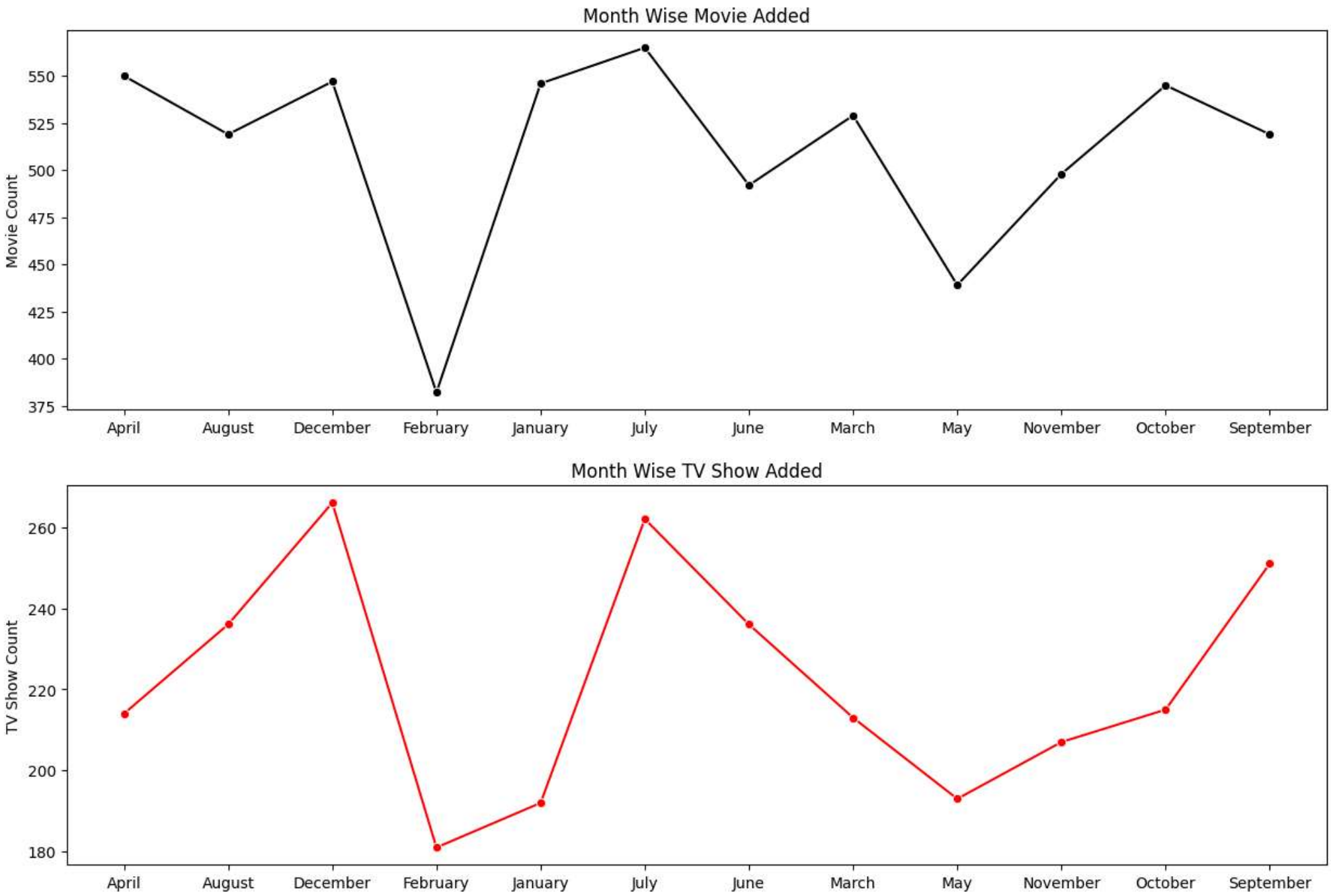
In [66]:
```
movie_month = data_month[data_month["type"] == "Movie"]
```

In [67]:
```
tv_month = data_month[data_month["type"] == "TV Show"]
```

In [68]:
```
plt.figure(figsize =(15, 10))
plt.subplot(2,1,1)
sns.lineplot(x = "month_added", y = "count" , data = movie_month, marker = "o", color = "black")
plt.xlabel("")
plt.ylabel("Movie Count")
plt.title("Month Wise Movie Added")

plt.subplot(2,1,2)
sns.lineplot(x = "month_added", y = "count" , data = tv_month, marker = "o", color = "red")
plt.xlabel("")
plt.ylabel("TV Show Count")
plt.title("Month Wise TV Show Added")

plt.suptitle("Movie and TV Show Launch On Platform , Month", fontsize = 25)
plt.show()
```



🔍 **Insights:**

- **Seasonal Trends:** Both movies and TV shows experience a significant drop in February, with peaks in July and December, indicating a seasonal content addition strategy.
- **High Activity Months:** July and December are the peak months for adding both movies and TV shows, suggesting a coordinated effort to boost content during these periods.
- **Low Activity Months:** February and May are low activity months for content addition, presenting opportunities to increase content during these times to maintain user engagement.
- **Consistent Trends:** The similar trends in content addition for both movies and TV shows imply a unified content strategy, potentially based on user engagement data and seasonal considerations.

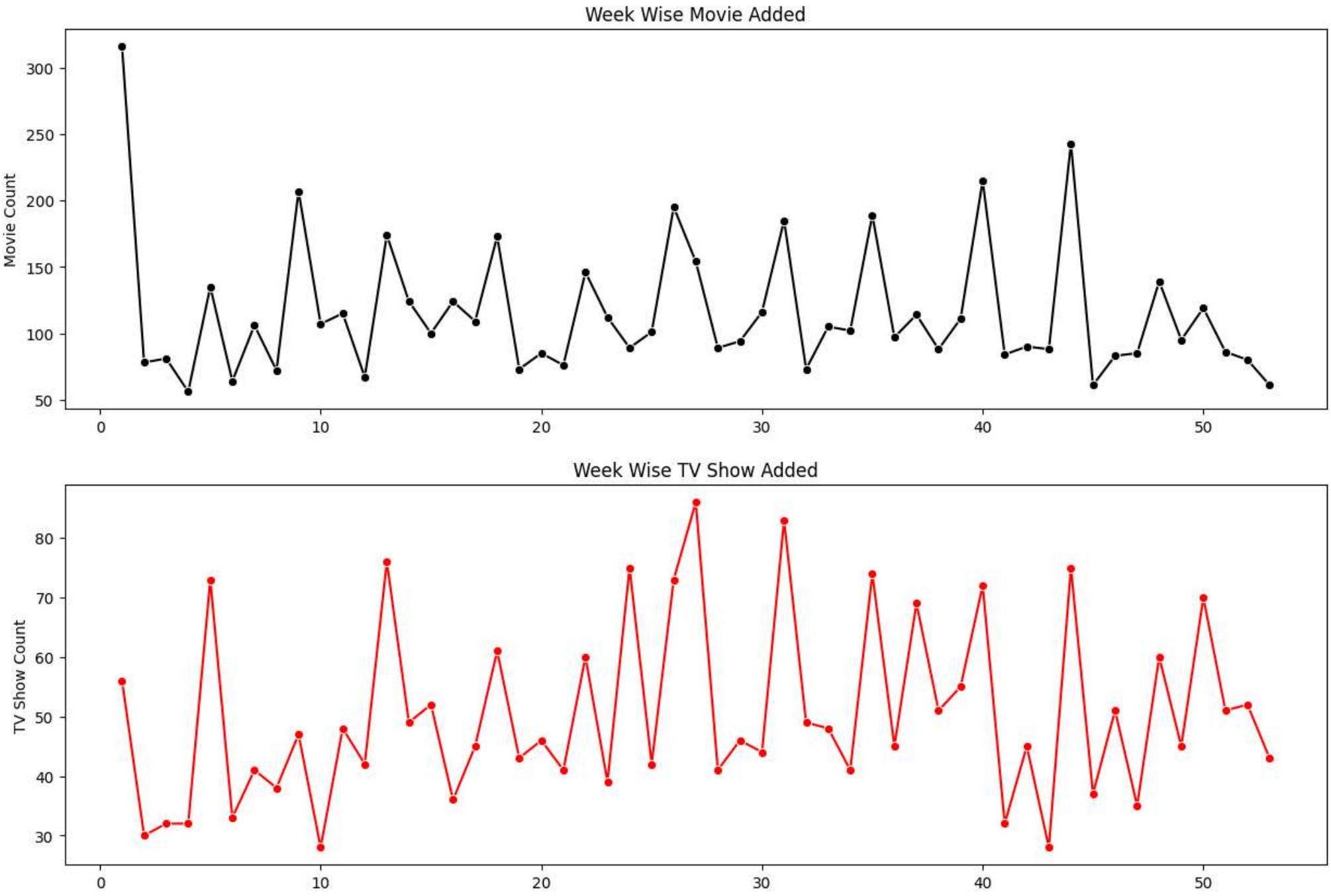
📅 Best Week to Launch a Movie / TV Show

In [69]:
```
data_week = data.groupby("week_added")["type"].value_counts().reset_index()
movie_week = data_week[data_week["type"]=="Movie"]
tv_week = data_week[data_week["type"]=="TV Show"]
```

In [70]:
```
plt.figure(figsize =(15, 10))
plt.subplot(2,1,1)
sns.lineplot(x = "week_added", y = "count" , data = movie_week, marker = "o", color = "black")
plt.xlabel("")
plt.ylabel("Movie Count")
plt.title("Week Wise Movie Added")

plt.subplot(2,1,2)
sns.lineplot(x = "week_added", y = "count" , data = tv_week, marker = "o", color = "red")
plt.xlabel("")
plt.ylabel("TV Show Count")
plt.title("Week Wise TV Show Added")

plt.suptitle("Movie and TV Show Launch On Platform , Week", fontsize = 25)
plt.show()
```



🔍 **Insights:**

- **Consistent Patterns:** Both Movies and TV shows on Netflix exhibit consistent weekly upload patterns, showing noticeable spikes and dips.

These patterns suggest a cyclic trend where content uploads surge during specific weeks, followed by quieter periods.

- **Peak Upload Times:** Movies tend to have their highest volume of uploads during the initial week of the year.

TV shows see their peak uploads around the 26th week, typically towards the end of June.

📅 Best Day to Launch a Movie / TV Show

In [71]:
```
data_day = data.groupby("day_added")["type"].value_counts().reset_index()
```

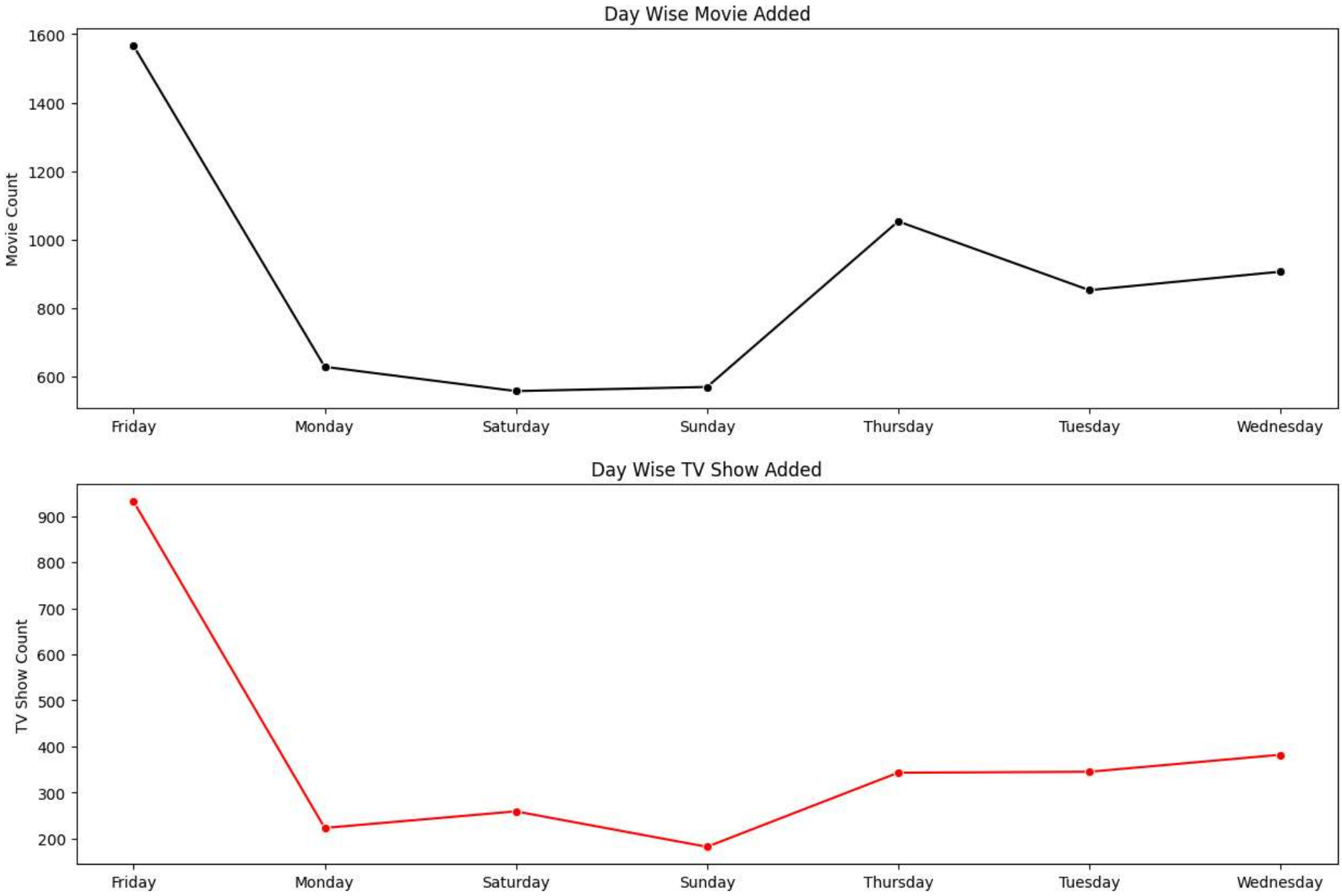
In [72]:
```
movie_day = data_day[data_day["type"] == "Movie"]
```

In [73]:
```
tv_day = data_day[data_day["type"] == "TV Show"]
```

In [74]:
```
plt.figure(figsize =(15, 10))
plt.subplot(2,1,1)
sns.lineplot(x = "day_added", y = "count" , data = movie_day, marker = "o", color = "black")
plt.xlabel("")
plt.ylabel("Movie Count")
plt.title("Day Wise Movie Added")

plt.subplot(2,1,2)
sns.lineplot(x = "day_added", y = "count" , data = tv_day, marker = "o", color = "red")
plt.xlabel("")
plt.ylabel("TV Show Count")
plt.title("Day Wise TV Show Added")

plt.suptitle("Movie and TV Show Launch On Platform, Day", fontsize = 25)
plt.show()
```



🔍 **Insights:**

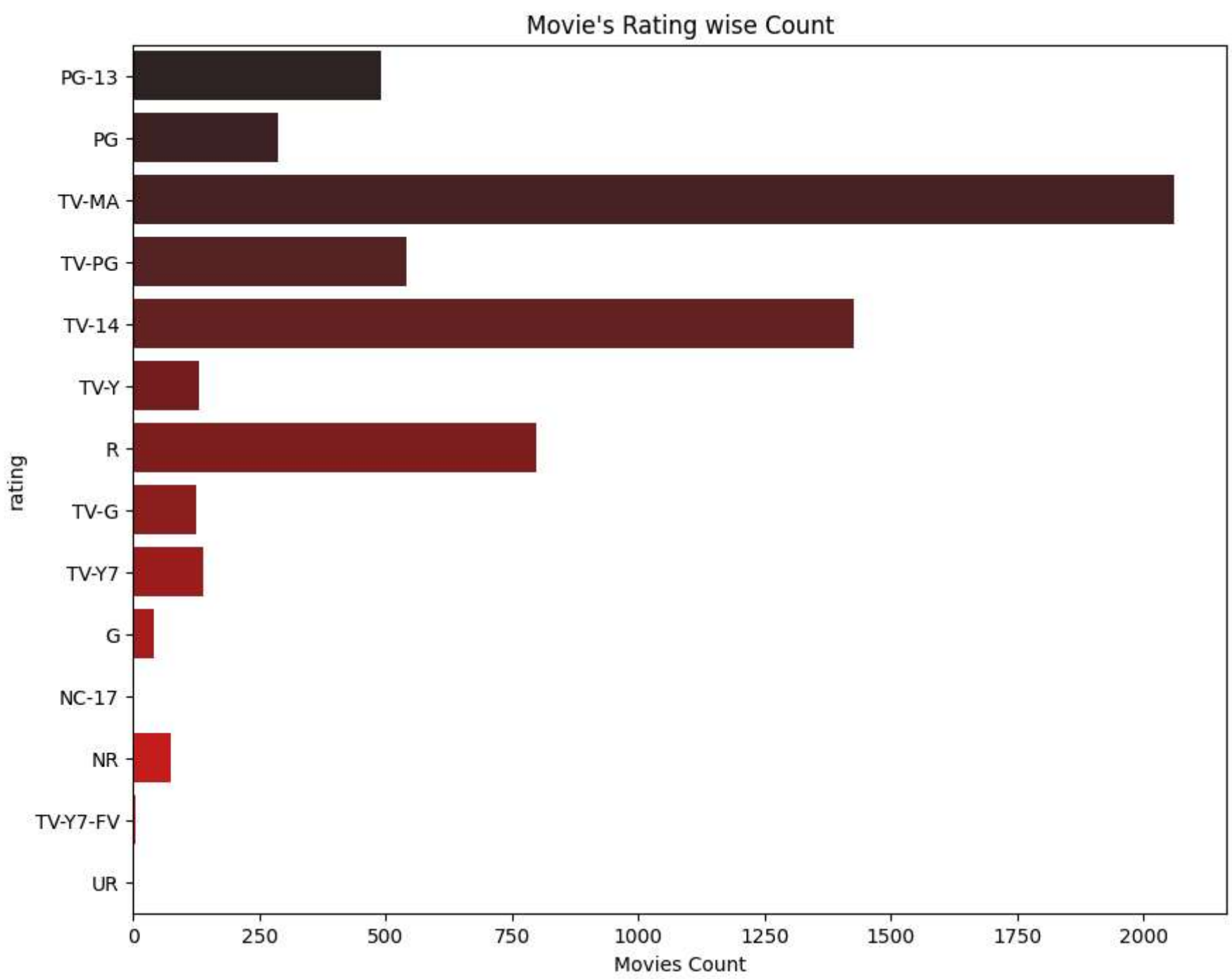
- **Peak Content Addition Day:** Most movies and TV shows are added on Fridays, indicating a strategic end-of-week release.
- **Low Activity Days:** Sundays see the fewest movies, and Mondays have the fewest TV shows, marking these as less active days for content addition.
- **Midweek Trends:** Content additions increase midweek, with significant upticks for both movies and TV shows on Thursdays.
- **Consistent Patterns:** The similar patterns in content release for movies and TV shows suggest a coordinated strategy to maximize engagement towards the end of the week.

🍿 Content Ratings

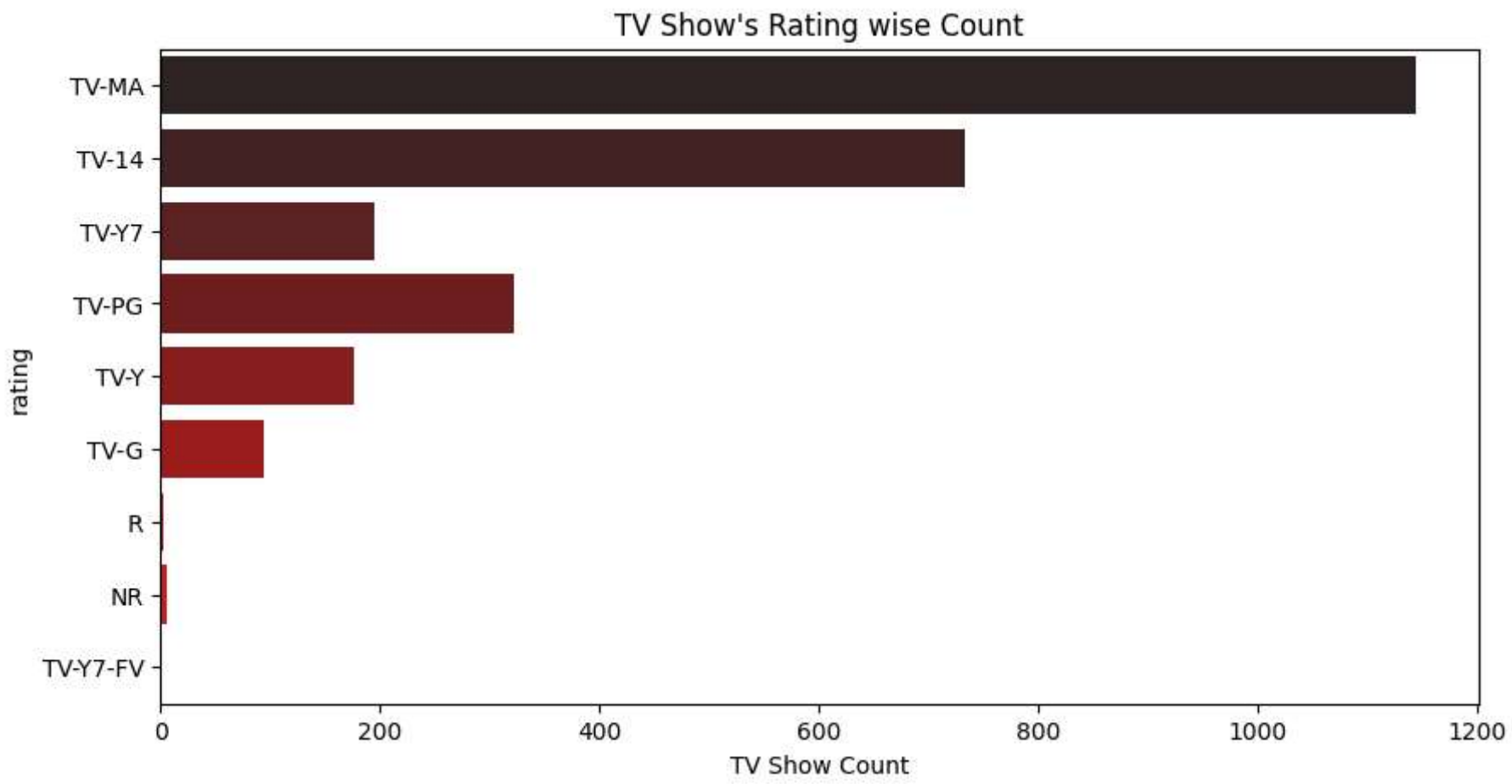
In [75]:
```
movie_rating = data[["type" , "rating"]][(data["type"]=="Movie") & (data["rating"]!= "Unknown Rating")]
```

In [76]:
```
tv_rating = data[["type" , "rating"]][(data["type"]=="TV Show") & (data["rating"]!= "Unknown Rating")]
```

In [77]:
```
plt.figure(figsize = (10,8))
sns.countplot(y = "rating", data = movie_rating, hue = "rating", legend = False , palette = "dark:red")
plt.xlabel("Movies Count")
plt.title("Movie's Rating wise Count")
plt.show()
```



In [78]:
```
plt.figure(figsize = (10,5))
sns.countplot(y = "rating", data = tv_rating, hue = "rating", legend = False , palette = "dark:red")
plt.xlabel("TV Show Count")
plt.title("TV Show's Rating wise Count")
plt.show()
```



🔍 **Insights :**

- The majority of Netflix's movies and TV shows are aimed at adult audiences, with significant content rated TV-MA.
- Following adult content, a considerable amount of Netflix's offerings are rated TV-14, targeting teenage viewers.
- There is also a notable portion of content rated TV-PG, suitable for older children.
- Content rated TV-Y and TV-Y7, designed for younger children, makes up a smaller segment of Netflix's library.
- This distribution indicates Netflix's strategic focus on appealing to a wide audience, with a particular emphasis on adults and teens due to their substantial purchasing power.

🎦 Various Genres of Movies and TV Shows available on Netflix.

In [79]:
```
text_movie = data["listed_in"].str.split(", ").explode()[data["type"] == "Movie"].value_counts()
text_tv = data["listed_in"].str.split(", ").explode()[data["type"] == "TV Show"].value_counts()
```

In [80]:
```
from wordcloud import WordCloud # importing wordcloud

color = sns.color_palette("dark:red", as_cmap=True)
wordcld = WordCloud(width=1500, height=400, background_color='white', colormap=color).generate_from_frequencies(text_movie)

plt.figure(figsize=(12, 4))
plt.imshow(wordcld, interpolation='bilinear')
plt.axis('off')
plt.show()
```

In [81]:
```
color = sns.color_palette("dark:red", as_cmap=True)
wordcld = WordCloud(width=1500, height=400, background_color='white', colormap=color).generate_from_frequencies(text_tv)

plt.figure(figsize=(12, 4))
plt.imshow(wordcld, interpolation='bilinear')
plt.axis('off')
plt.show()
```

🔍 Insights:

- Netflix's popular movie genres span International Movies, Comedies, Dramas,Family, Action, and Romantic films.
- Among Netflix International series, TV shows, popular genres include Drama, Crime, Romance, Kids' content, Comedies,Anime, and Reality Shows.

👨‍👩‍👧 Targeted Audience

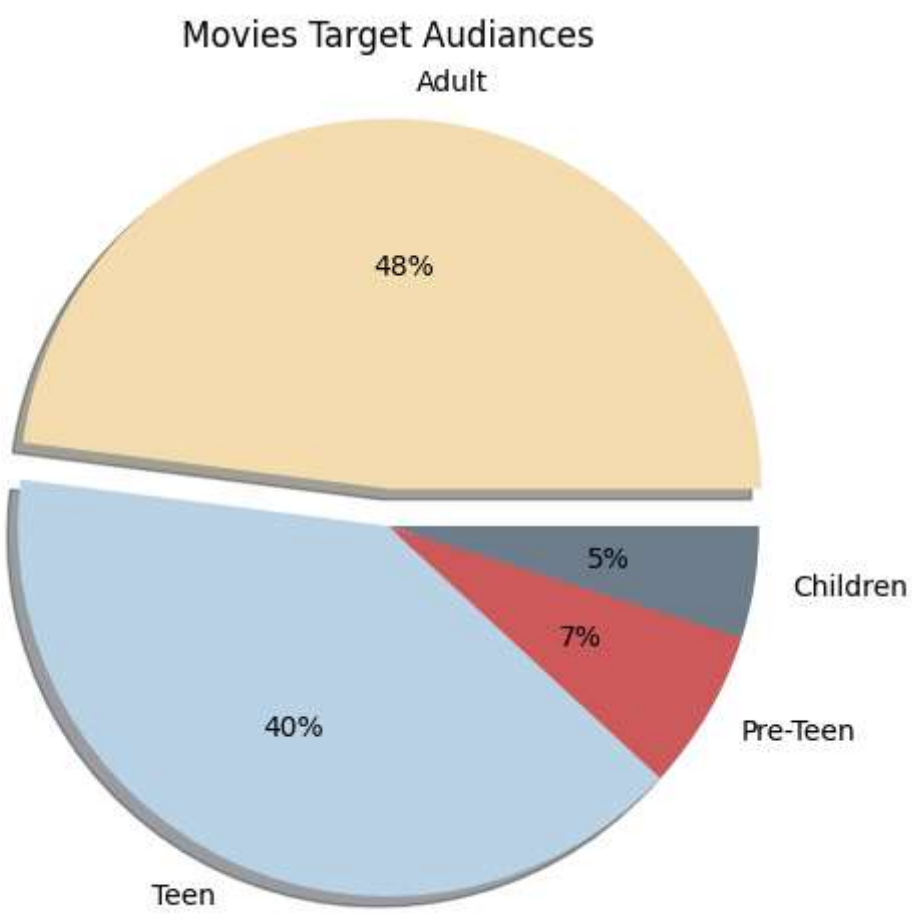
In [82]:
```
def target_aud(rating):
  if rating in ["TV-Y","G","TV-G"]:
    return "Children"
  elif rating in ["TV-Y7","TV-Y7-FV","PG"]:
    return "Pre-Teen"
  elif rating in ["TV-PG","PG-13","TV-14"]:
    return "Teen"
  else:
    return "Adult"

movie_rating["age_group"] = movie_rating["rating"].apply(lambda x:target_aud(x))
tv_rating["age_group"] =  tv_rating["rating"].apply(lambda x:target_aud(x))
```

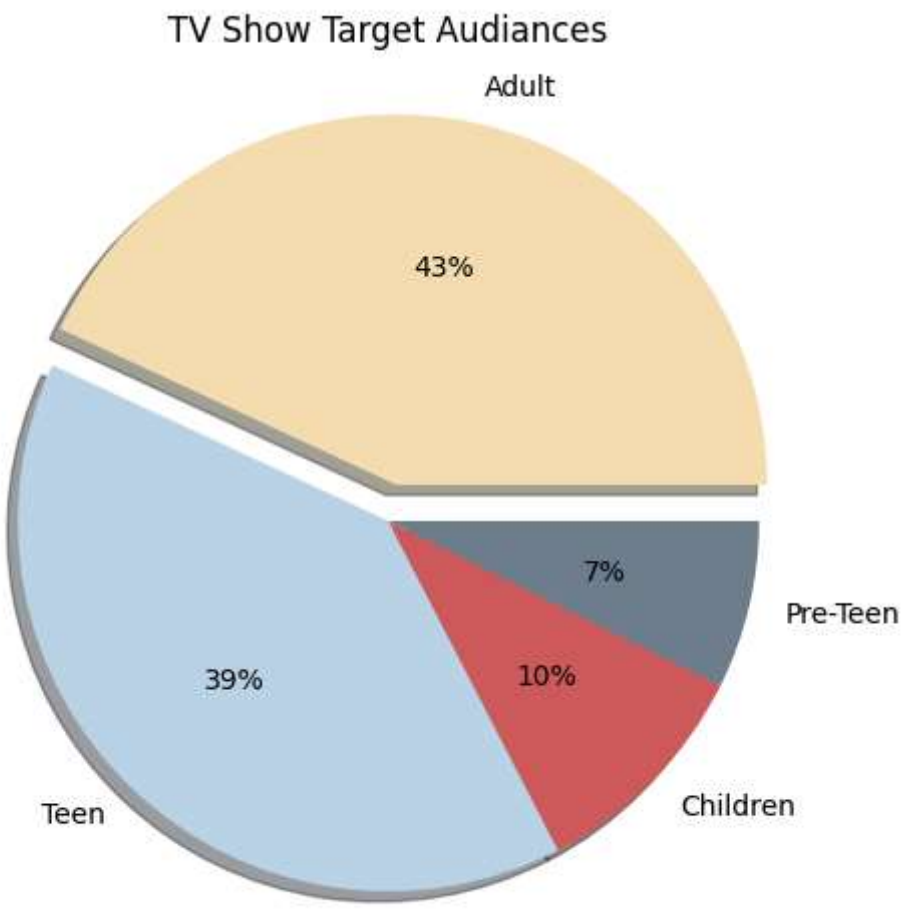
In [83]:
```
movie_agegroup = movie_rating["age_group"].value_counts().reset_index()
```

In [84]:
```
tv_agegroup = tv_rating["age_group"].value_counts().reset_index()
```

In [85]:
```
plt.figure(figsize = (6,6))
colors = ['#F5DEB3', '#BCD4E6', '#CD5C5C', '#708090']
explode = (0.1 , 0 , 0 , 0)
plt.pie(movie_agegroup["count"], labels = movie_agegroup["age_group"],autopct='%1.f%%', shadow=True, colors = colors,explode = explode)
plt.title("Movies Target Audiances")
plt.show()
```



In [86]:
```
plt.figure(figsize = (6,6))
colors = ['#F5DEB3', '#BCD4E6', '#CD5C5C', '#708090']
explode = (0.1 , 0 , 0 , 0)
plt.pie(tv_agegroup["count"], labels = tv_agegroup["age_group"],autopct='%1.f%%', shadow=True, colors = colors,explode = explode)
plt.title("TV Show Target Audiances")
plt.show()
```



🔍 **Insights:**

- **Movies:** Netflix's movie collection is predominantly geared towards adult viewers, with a significant portion also catering to teenagers and a smaller segment tailored for children.
- **TV Shows:** Netflix's TV show lineup follows a similar trend, with a notable emphasis on content for children alongside offerings for adult and teenage audiences, including popular anime series.

📊 Rating Wise Countries Demographic Analysis

In [87]:
```
country_content = data_new[["country","title","rating"]].drop_duplicates(keep = "first")
```

In [88]:
```
country_content_top = country_content[country_content["country"].isin(top_country["country"])]
country_content_top =  country_content_top[country_content_top["rating"]!= "Unknown Rating"]
```

In [89]:
```
country_content_top2 = country_content_top.groupby("country")["rating"].value_counts().reset_index().sort_values(by = "count", ascending = 
```

In [90]:
```
country_content_top2.head(3)
```

Out[90]:

	country	rating	count
102	United States	TV-MA	1100
103	United States	R	660
37	India	TV-14	572

- # Non Graphical Analysis

- Top 10 country Most demanded Content Ratings

In [91]:
```
top_rating_country = country_content_top2.sort_values(by= "count", ascending = False).drop_duplicates("country",keep  = "first")
```

In [92]:
```
top_rating_country[["country","rating"]]
```

Out[92]:

	country	rating
102	United States	TV-MA
37	India	TV-14
90	United Kingdom	TV-MA
78	Spain	TV-MA
13	France	TV-MA
0	Canada	TV-MA
58	Mexico	TV-MA
48	Japan	TV-MA
67	South Korea	TV-MA
26	Germany	TV-MA

- # Graphical Analysis

In [93]:
```
country_total_rating = country_content_top2.groupby("country")["count"].sum().reset_index()
data_country_rating = country_content_top2.merge(country_total_rating , on = "country")
data_country_rating.rename(columns = {"count_x":"count","count_y":"total_count"},inplace = True)
```

In [94]:
```
data_country_rating["percent"] = round(data_country_rating["count"] / data_country_rating["total_count"] * 100,1)
```

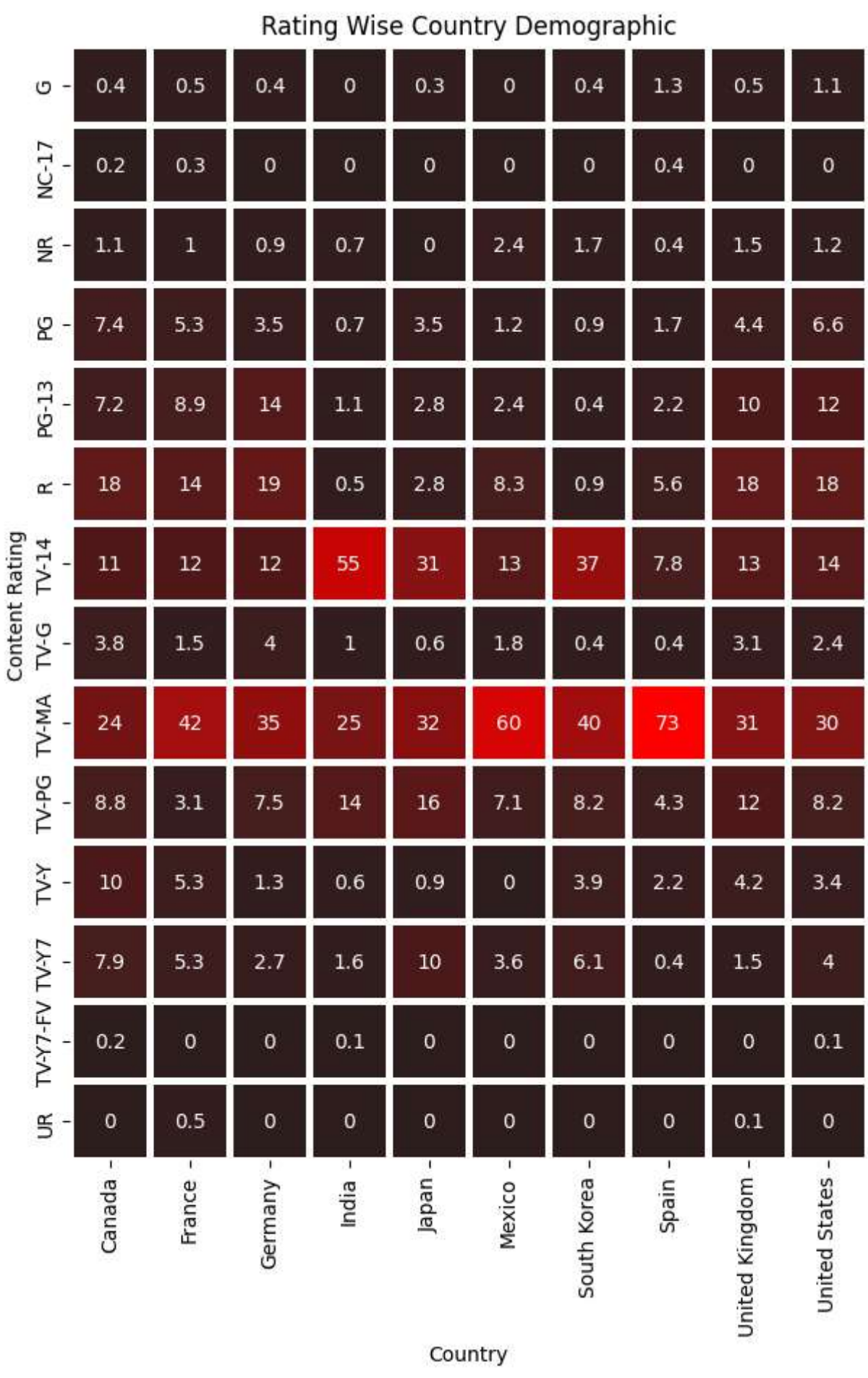
In [95]:
```
country_heatmap = data_country_rating.pivot(index = "rating", columns = "country",values ="percent").fillna(0)
```

In [96]:
```
country_heatmap
```

Out[96]:

country rating	Canada	France	Germany	India	Japan	Mexico	South Korea	Spain	United Kingdom	United States
G	0.4	0.5	0.4	0.0	0.3	0.0	0.4	1.3	0.5	1.1
NC-17	0.2	0.3	0.0	0.0	0.0	0.0	0.0	0.4	0.0	0.0
NR	1.1	1.0	0.9	0.7	0.0	2.4	1.7	0.4	1.5	1.2
PG	7.4	5.3	3.5	0.7	3.5	1.2	0.9	1.7	4.4	6.6
PG-13	7.2	8.9	13.7	1.1	2.8	2.4	0.4	2.2	10.4	11.7
R	17.8	14.5	19.0	0.5	2.8	8.3	0.9	5.6	18.0	17.9
TV-14	11.0	12.2	11.9	54.7	31.2	13.0	37.2	7.8	12.8	13.5
TV-G	3.8	1.5	4.0	1.0	0.6	1.8	0.4	0.4	3.1	2.4
TV-MA	24.0	41.5	35.0	25.4	31.9	60.4	39.8	73.3	31.2	29.8
TV-PG	8.8	3.1	7.5	13.8	15.8	7.1	8.2	4.3	12.2	8.2
TV-Y	10.1	5.3	1.3	0.6	0.9	0.0	3.9	2.2	4.2	3.4
TV-Y7	7.9	5.3	2.7	1.6	10.1	3.6	6.1	0.4	1.5	4.0
TV-Y7-FV	0.2	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.1
UR	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0

In [97]:
```
plt.figure(figsize= (10,10))
color = sns.color_palette("dark:red", as_cmap=True)
sns.heatmap(data = country_heatmap, cmap = color,square = True ,linewidth = 2.5,cbar = False,annot = True )
plt.ylabel("Content Rating")
plt.xlabel("Country")
plt.title("Rating Wise Country Demographic")
plt.show()
```



🔍 **Insights:**

- **High TV-MA Content in the U.S. and Spain:** The United States (73%) and Spain (60%) have the highest percentages of TV-MA content, indicating a preference for mature content in these countries.
- **Dominance of TV-14 in India:** India has an exceptionally high percentage (55%) of TV-14 rated content, suggesting a significant focus on content suitable for teenagers and older audiences.
- **R and TV-MA Content Popularity:** TV-MA and R-rated content are notably popular across multiple countries, with the U.S., France, and Spain showing high percentages, highlighting a trend towards mature and adult content.
- **Low NC-17 Content:** NC-17 content is almost negligible across all countries, indicating either low production or limited acceptance of this rating category globally.

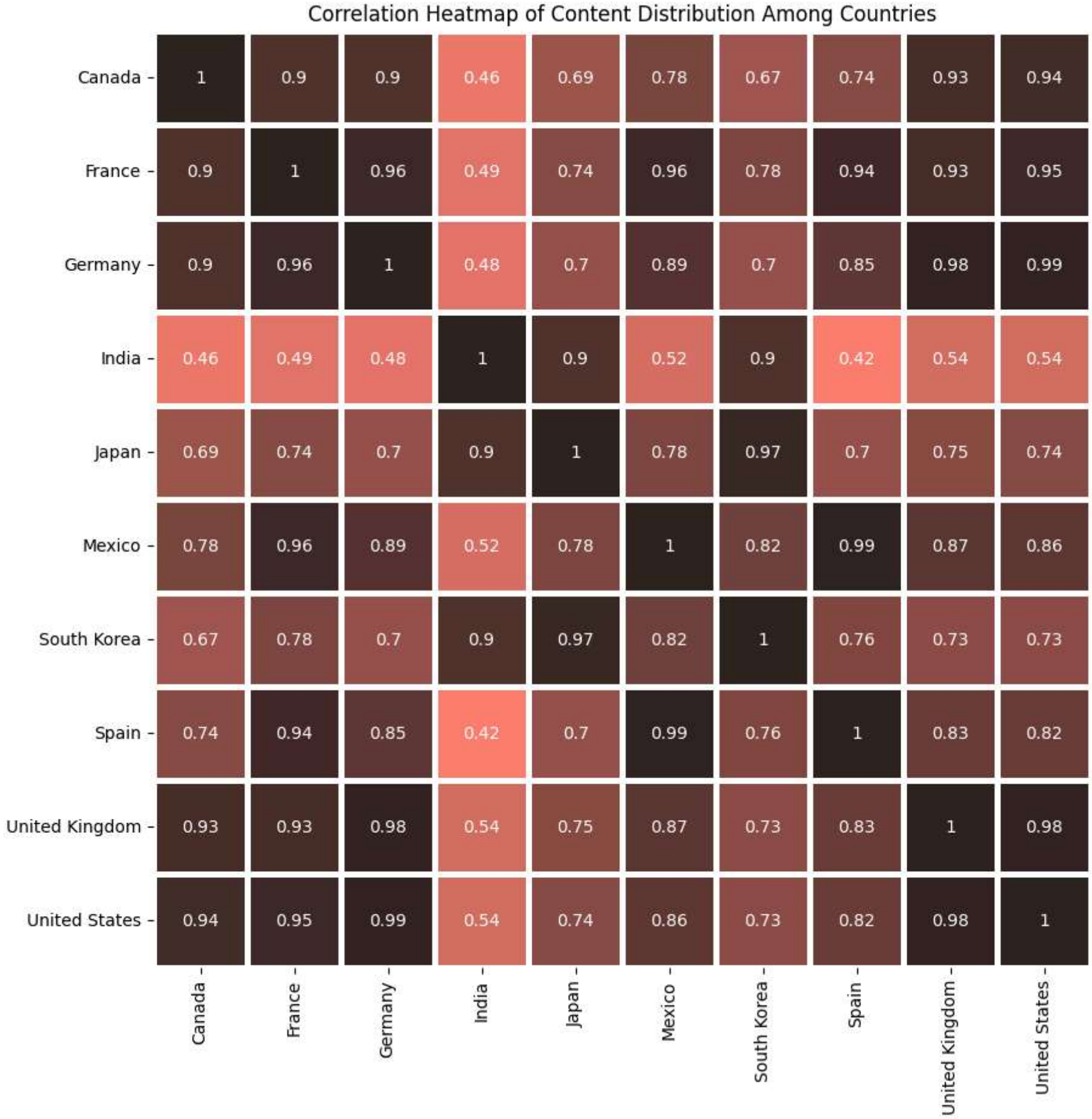
📈 Correlation Heatmap of Content Distribution Among Countries

In [98]:
```
country_content_corr = country_heatmap.corr()
country_content_corr
```

Out[98]:

country	Canada	France	Germany	India	Japan	Mexico	South Korea	Spain	United Kingdom	United States
country										
Canada	1.000000	0.895932	0.901699	0.460418	0.687366	0.783531	0.667889	0.742837	0.930021	0.939630
France	0.895932	1.000000	0.955247	0.493830	0.739459	0.957098	0.779570	0.942297	0.930751	0.949305
Germany	0.901699	0.955247	1.000000	0.480059	0.700963	0.887102	0.702935	0.853470	0.979157	0.986962
India	0.460418	0.493830	0.480059	1.000000	0.898933	0.516328	0.902944	0.424994	0.539193	0.538169
Japan	0.687366	0.739459	0.700963	0.898933	1.000000	0.775855	0.965921	0.699409	0.747979	0.744045
Mexico	0.783531	0.957098	0.887102	0.516328	0.775855	1.000000	0.815371	0.989398	0.872571	0.862778
South Korea	0.667889	0.779570	0.702935	0.902944	0.965921	0.815371	1.000000	0.755457	0.730817	0.734115
Spain	0.742837	0.942297	0.853470	0.424994	0.699409	0.989398	0.755457	1.000000	0.831288	0.820822
United Kingdom	0.930021	0.930751	0.979157	0.539193	0.747979	0.872571	0.730817	0.831288	1.000000	0.984827
United States	0.939630	0.949305	0.986962	0.538169	0.744045	0.862778	0.734115	0.820822	0.984827	1.000000

In [99]:
```
plt.figure(figsize= (10,10))
color = sns.color_palette("dark:salmon_r", as_cmap=True)
sns.heatmap(data = country_content_corr, cmap = color,square = True ,linewidth = 2.5,cbar = False,annot = True)
plt.xlabel("")
plt.ylabel("")
plt.title("Correlation Heatmap of Content Distribution Among Countries")
plt.show()
```



🔍 **Insights:**

- High Correlation Among Western Countries: The United States, United Kingdom, Canada, France, and Germany exhibit very high correlations in content distribution, indicating similar content preferences.

- Distinct Content Preferences in India: India shows lower correlations with other countries, especially Western ones, highlighting unique content consumption patterns.
- Similar Content Preferences Between Mexico and Spain: Mexico and Spain have a high correlation (0.99), reflecting similar content distributions, likely due to shared language and cultural ties.
- Moderate Correlation for Japan and South Korea: Japan and South Korea show moderate correlations with Western countries, suggesting a blend of unique and shared content preferences.

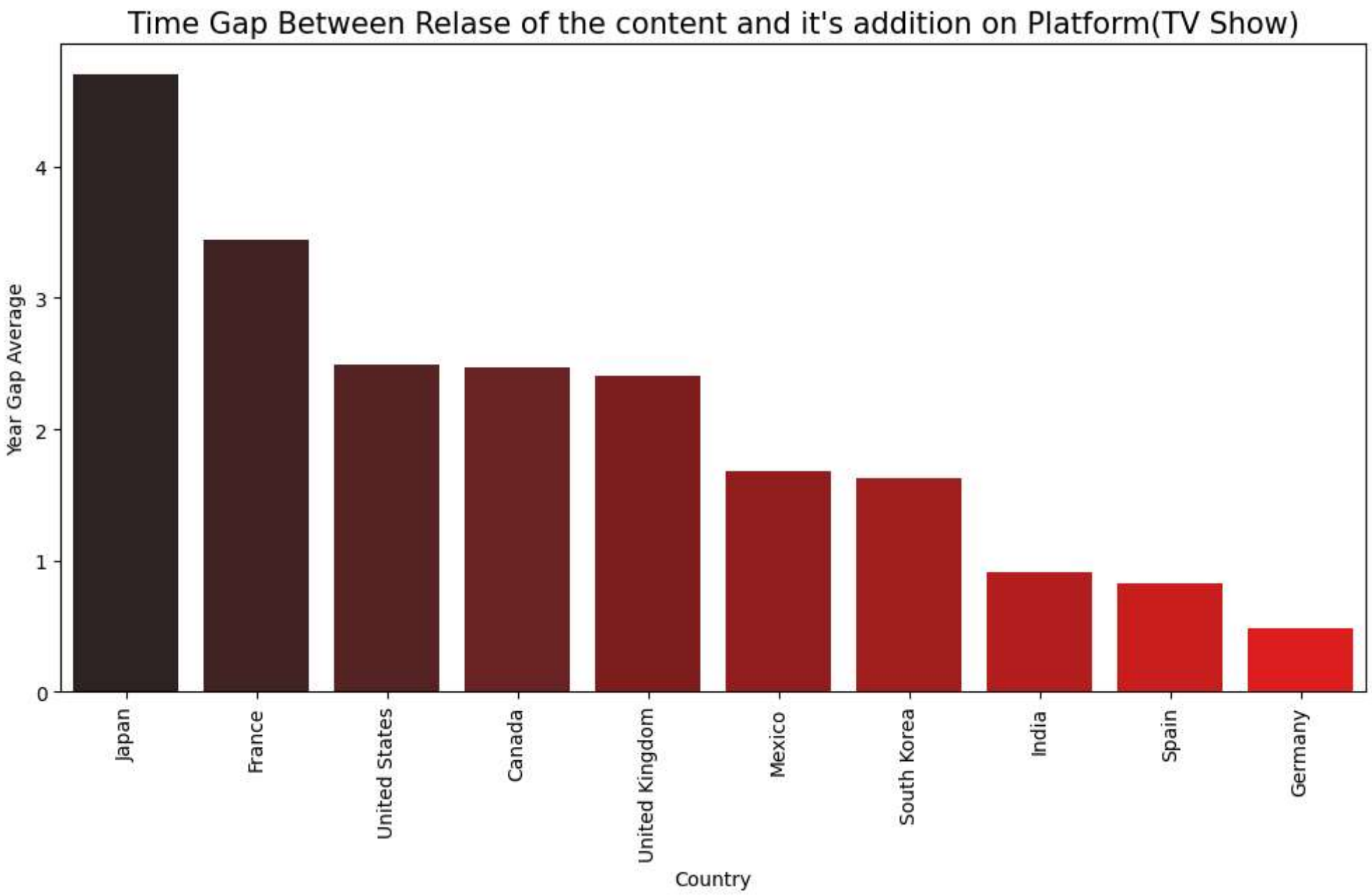
⏳ Time Gap Between Release of the content and it's addition on Platform

In [100…
```
data_year_tv = data_new[["country","release_year","year_added"]][(data_new["type"]== "TV Show") & (data_new["country"]!= "Unknown Country")
data_year_tv["gap"] = data_year_tv["year_added"] - data_year_tv["release_year"]
year_tv = data_year_tv.groupby("country")["gap"].mean().round(2).reset_index().sort_values(by="gap",ascending = False)
year_tv = year_tv[year_tv["country"].isin(top_country["country"])]
year_tv
```

Out[100…

	country	gap
30	Japan	4.70
19	France	3.44
63	United States	2.49
8	Canada	2.47
62	United Kingdom	2.41
38	Mexico	1.68
52	South Korea	1.63
25	India	0.91
53	Spain	0.83
20	Germany	0.48

In [101…
```
plt.figure(figsize=(12,6))
sns.barplot(x = "country", y= "gap",hue = "country",legend = False , data= year_tv,palette="dark:red")
plt.xlabel("Country")
plt.ylabel("Year Gap Average")
plt.title("Time Gap Between Relase of the content and it's addition on Platform(TV Show)",fontsize = 15)
plt.xticks(rotation = 90)
plt.show()
```



In [102…
```
data_year_movie = data_new[["country","release_year","year_added"]][(data_new["type"]== "Movie") & (data_new["country"]!= "Unknown Country"
data_year_movie["gap"] = data_year_movie["year_added"] - data_year_movie["release_year"]
```

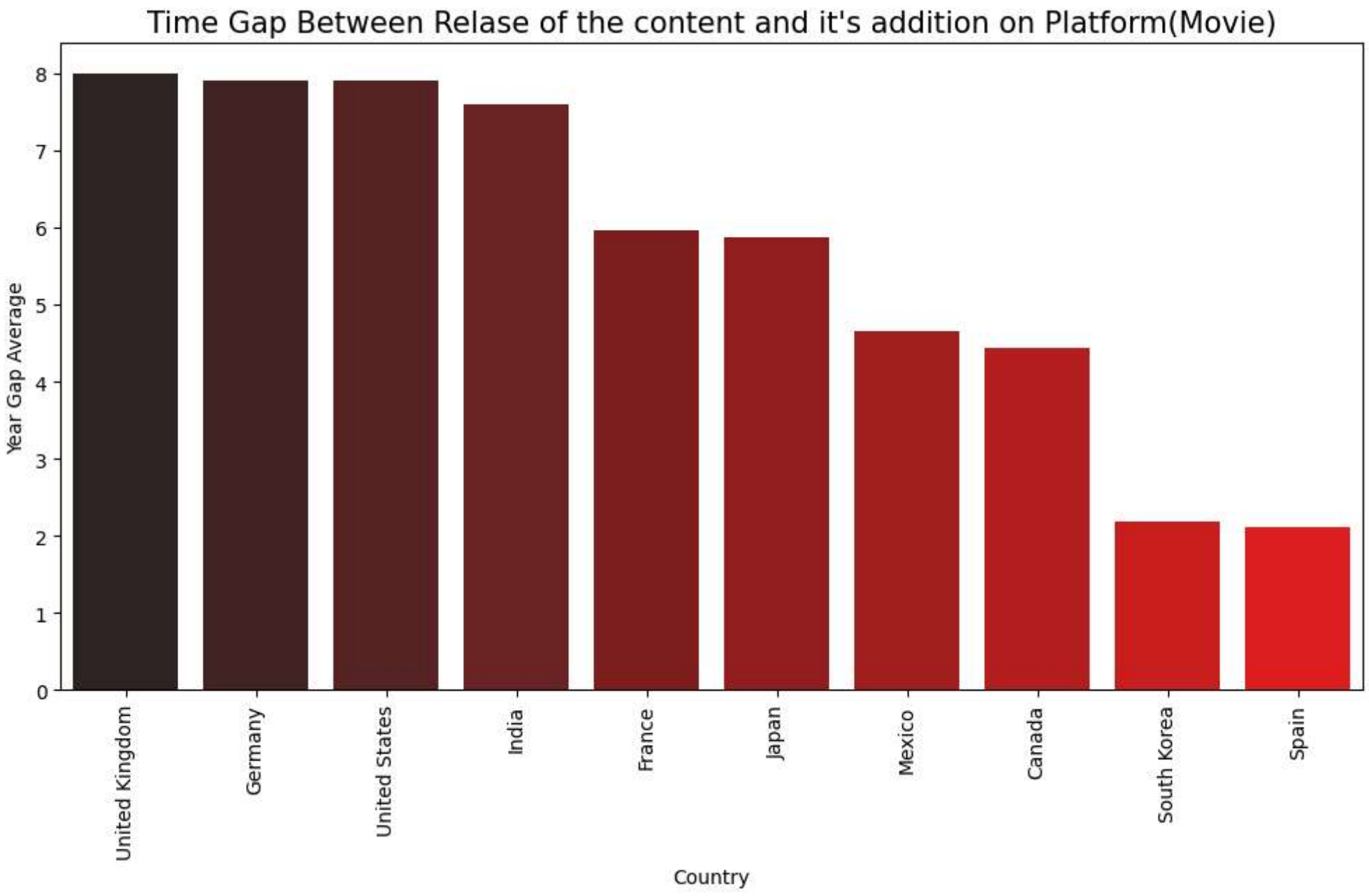
```
year_movie = data_year_movie.groupby("country")["gap"].mean().round(2).reset_index().sort_values(by="gap",ascending = False)
year_movie = year_movie[year_movie["country"].isin(top_country["country"])]
year_movie
```

Out[102…

	country	gap
112	United Kingdom	7.99
36	Germany	7.91
114	United States	7.91
43	India	7.60
34	France	5.96
51	Japan	5.88
65	Mexico	4.66
20	Canada	4.43
98	South Korea	2.18
100	Spain	2.12

In [103…

```
plt.figure(figsize=(12,6))
sns.barplot(x = "country", y= "gap",hue = "country",legend = False , data= year_movie,palette="dark:red")
plt.xlabel("Country")
plt.ylabel("Year Gap Average")
plt.title("Time Gap Between Relase of the content and it's addition on Platform(Movie)",fontsize = 15)
plt.xticks(rotation = 90)
plt.show()
```



🔍 Insights:

- In general, movies are added to Netflix with a longer delay compared to TV shows, suggesting a higher demand for recent TV shows among viewers.
- Spain stands out with a minimal delay of 2 years for movies and 1 year for TV shows, indicating a preference for more recent content.
- India, the United Kingdom, the United States, and France show significant delays of 6 to 8 years for movies and only 1 to 3 years for TV shows, reflecting a preference for recent TV content over older movies.
- Japan maintains a consistent delay of 5 to 6 years for both movies and TV shows, possibly influenced by considerations related to dubbing or language barriers, ensuring a stable release pattern.

Strategic Business Recommendations for Netflix

1. Enhanced Investment in Localized Content :

- Cultural Relevance: Continue investing in content that aligns with diverse cultural and linguistic preferences globally.

- Asian Market Focus: Increase production and acquisition of Asian TV shows, particularly from South Korea and Japan.
- European Movie Curation: Prioritize building a diverse and engaging movie library tailored to European tastes.
- Market Emphasis: Maintain focus on content production in the USA and India due to significant investment and market potential.

2. Optimized Release Timing Strategies

- Monthly Focus: Concentrate on high-quality releases during peak viewer demand months (January, July, August, October, December).
- Weekly Highlight: Designate the first week of each month as a "Featured Release Week" to launch major TV shows or movies. Use subsequent weeks to promote existing content effectively.

3. Targeted Age-Specific Content Expansion

- Teen-Centric Content: Partner with local studios in India and Japan to develop original series and movies that resonate with teenage audiences.
- Adult-Centric Content: Create sophisticated and culturally aligned original content for mature audiences in Spain, Mexico, Germany, and France.

4. Diverse Content Runtimes

- Movie Formats: Continue producing standard-length films while exploring medium-length formats to cater to varied viewer preferences.
- TV Series Formats: Focus on producing limited series and shorter seasons to accommodate audience preferences for manageable episodic content.

5. Streamlined Content Acquisition Strategy

- Timely Access: Improve acquisition of recent movies to reduce the gap between theatrical release and availability on the platform, aligning with viewer expectations for up-to-date content.

Additional Point:

- Data-Driven Decision Making

Analytics Integration: Utilize viewer data to inform content acquisition, production, and release strategies, ensuring decisions are aligned with audience preferences and market trends.