**Importing Libraries:**

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

**Reading csv file:**

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

```
df
```

| | 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... |
| | | | Jailbirds | | | | | | | | | Feuds, |

**Step 0:** Basic understanding and clean up

```
#To get started, let us look at the shape of the raw data.
df.shape
```

```
(8807, 12)
```

```
#Looking at the random data to get the feel of variation in data
df.sample(5)
```

| | show_id | type | title | director | cast | country | date_added | relea |
|---|---|---|---|---|---|---|---|---|
| **2895** | s2896 | Movie | System Crasher | Nora Fingscheidt | Helena Zengel, Albrecht Schuch, Gabriela Maria... | Germany | February 21, 2020 | |
| **6046** | s6047 | Movie | A Most Violent Year | J.C. Chandor | Oscar Isaac, Jessica Chastain, David Oyelowo, ... | United Arab Emirates, United States | July 22, 2018 | |

```
#Taking a look at the null count in entire data -
df.isnull().sum()
```

```
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
```

```
#Cleaning - There are multiple names in the form of list in director, cast columns etc. If we need to find how many movies a cast person has
df_director_r = pd.DataFrame(df["director"].apply(lambda x: str(x).split(",")).tolist(), index = df["title"])
df_director = df_director_r.stack().reset_index()
df_director.drop("level_1", axis = 1, inplace = True)
df_director.rename(columns = {0: "director"}, inplace = True)
df_director.head()
```

| | title | director |
|---|---|---|
| 0 | Dick Johnson Is Dead | Kirsten Johnson |
| 1 | Blood & Water | nan |
| 2 | Ganglands | Julien Leclercq |
| 3 | Jailbirds New Orleans | nan |
| 4 | Kota Factory | nan |

```
#Repeat same for cast -
df_cast_r = pd.DataFrame(df['cast'].apply(lambda x: str(x).split(',')).tolist(), index =df['title'])
df_cast = df_cast_r.stack().reset_index()
df_cast.drop('level_1', axis = 1, inplace = True)
df_cast.rename(columns ={0:'cast'}, inplace = True)
```

```
df_cast.head()
#note that Nan has been converted as a string nan, this needs to be removed.
```

| | title | cast |
|---|---|---|
| 0 | Dick Johnson Is Dead | nan |
| 1 | Blood & Water | Ama Qamata |
| 2 | Blood & Water | Khosi Ngema |
| 3 | Blood & Water | Gail Mabalane |
| 4 | Blood & Water | Thabang Molaba |

```
#Repeat same for country -
df_country_r = pd.DataFrame(df['country'].apply(lambda x: str(x).split(',')).tolist(), index =df['title'])
df_country = df_country_r.stack().reset_index()
df_country.drop('level_1', axis = 1, inplace = True)
df_country.rename(columns ={0:'country'}, inplace = True)
```

```
df_country.head()
```

| | title | country |
|---|---|---|
| 0 | Dick Johnson Is Dead | United States |
| 1 | Blood & Water | South Africa |
| 2 | Ganglands | nan |
| 3 | Jailbirds New Orleans | nan |
| 4 | Kota Factory | India |

```
#Repeat same for genre (listed_in) -
df_listed_in_r = pd.DataFrame(df['listed_in'].apply(lambda x: str(x).split(',')).tolist(), index =df['title'])
df_listed_in = df_listed_in_r.stack().reset_index()
df_listed_in.drop('level_1', axis = 1, inplace = True)
df_listed_in.rename(columns ={0:'listed_in'}, inplace = True)
```

```
df_listed_in.head()
```

| | title | listed_in |
|---|---|---|
| 0 | Dick Johnson Is Dead | Documentaries |
| 1 | Blood & Water | International TV Shows |
| 2 | Blood & Water | TV Dramas |
| 3 | Blood & Water | TV Mysteries |
| 4 | Ganglands | Crime TV Shows |

```
#Now let us join using merge.
df_new = df_director.merge(df_cast, how = "inner", on = "title")
df_new
```

| | title | director | cast |
|---|---|---|---|
| 0 | Dick Johnson Is Dead | Kirsten Johnson | nan |
| 1 | Blood & Water | nan | Ama Qamata |
| 2 | Blood & Water | nan | Khosi Ngema |
| 3 | Blood & Water | nan | Gail Mabalane |
| 4 | Blood & Water | nan | Thabang Molaba |
| ... | ... | ... | ... |
| 70807 | Zubaan | Mozez Singh | Manish Chaudhary |
| 70808 | Zubaan | Mozez Singh | Meghna Malik |
| 70809 | Zubaan | Mozez Singh | Malkeet Rauni |
| 70810 | Zubaan | Mozez Singh | Anita Shabdish |
| 70811 | Zubaan | Mozez Singh | Chittaranjan Tripathy |

70812 rows × 3 columns

```
#merge new and country
df_new = df_new.merge(df_country, how = "inner", on = "title")
df_new
```

| | title | director | cast | country |
|---|---|---|---|---|
| 0 | Dick Johnson Is Dead | Kirsten Johnson | nan | United States |
| 1 | Blood & Water | nan | Ama Qamata | South Africa |
| 2 | Blood & Water | nan | Khosi Ngema | South Africa |
| 3 | Blood & Water | nan | Gail Mabalane | South Africa |
| 4 | Blood & Water | nan | Thabang Molaba | South Africa |
| ... | ... | ... | ... | ... |
| 89410 | Zubaan | Mozez Singh | Manish Chaudhary | India |
| 89411 | Zubaan | Mozez Singh | Meghna Malik | India |
| 89412 | Zubaan | Mozez Singh | Malkeet Rauni | India |
| 89413 | Zubaan | Mozez Singh | Anita Shabdish | India |
| 89414 | Zubaan | Mozez Singh | Chittaranjan Tripathy | India |

89415 rows × 4 columns

```
#merge new_new and df_listed_in
df_new = df_new.merge(df_listed_in, how = "inner", on = "title")
df_new
```

| | title | director | cast | country | listed_in |
|---|---|---|---|---|---|
| 0 | Dick Johnson Is Dead | Kirsten Johnson | nan | United States | Documentaries |
| 1 | Blood & Water | nan | Ama Qamata | South Africa | International TV Shows |
| 2 | Blood & Water | nan | Ama Qamata | South Africa | TV Dramas |
| 3 | Blood & Water | nan | Ama Qamata | South Africa | TV Mysteries |
| 4 | Blood & Water | nan | Khosi Ngema | South Africa | International TV Shows |
| ... | ... | ... | ... | ... | ... |
| 202060 | Zubaan | Mozez Singh | Anita Shabdish | India | International Movies |
| 202061 | Zubaan | Mozez Singh | Anita Shabdish | India | Music & Musicals |
| 202062 | Zubaan | Mozez Singh | Chittaranjan Tripathy | India | Dramas |
| 202063 | Zubaan | Mozez Singh | Chittaranjan Tripathy | India | International Movies |

```
df.columns
```

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

```
df_final = df_new.merge(df[["show_id", "type", "title", "date_added", "release_year", "rating", "duration", "description"]], how = "inner", o
```

```
df_final
```

| | title | director | cast | country | listed_in | show_id | type | date_added |
|---|---|---|---|---|---|---|---|---|
| 0 | Dick Johnson Is Dead | Kirsten Johnson | nan | United States | Documentaries | s1 | Movie | September 25, 2021 |
| 1 | Blood & Water | nan | Ama Qamata | South Africa | International TV Shows | s2 | TV Show | September 24, 2021 |
| 2 | Blood & Water | nan | Ama Qamata | South Africa | TV Dramas | s2 | TV Show | September 24, 2021 |
| 3 | Blood & Water | nan | Ama Qamata | South Africa | TV Mysteries | s2 | TV Show | September 24, 2021 |

```
#During list conversion, all NaN values will be converted to string "NaN". This should be rectified.
df_final["cast"].replace(["nan"], ["Unknown Actor"], inplace = True)
df_final["director"].replace(["nan"], ["Unknown Director"], inplace = True)
df_final["country"].replace(["nan"], ["Unknown Country"], inplace = True)
df_final["listed_in"].replace(["nan"], ["Unknown listed_in"], inplace = True)
```

```
df_final.sample(10)
```

| | title | director | cast | country | listed_in | show_id | type | date_ad |
|---|---|---|---|---|---|---|---|---|
| **88663** | Bolívar | Andrés Beltrán | Shany Nadan | Colombia | Romantic TV Shows | s3720 | TV Show | June 2 |
| **119740** | Together | Unknown Director | Elvin Ng | Singapore | International TV Shows | s5196 | TV Show | Novem 1, 2 |
| **195287** | The Show | Giancarlo Esposito | Sarah Wayne Callies | United States | Dramas | s8506 | Movie | Jun 2 |

```
#we saw that duration column is null.
df_final[df_final["duration"].isnull()]
```

| | title | director | cast | country | listed_in | show_id | type | date_added | release |
|---|---|---|---|---|---|---|---|---|---|
| **126582** | Louis C.K. 2017 | Louis C.K. | Louis C.K. | United States | Movies | s5542 | Movie | April 4, 2017 | |

```
#Duration is NaN, but its located in rating, this needs to be updated - this is done by hard coding -
df_final["duration"].fillna(df_final[df_final["duration"].isnull()]["rating"], inplace = True)
```

```
df_final
```

| | title | director | cast | country | listed_in | show_id | type | date_added |
|---|---|---|---|---|---|---|---|---|
| **0** | Dick Johnson Is Dead | Kirsten Johnson | Unknown Actor | United States | Documentaries | s1 | Movie | September 25, 2021 |
| **1** | Blood & Water | Unknown Director | Ama Qamata | South Africa | International TV Shows | s2 | TV Show | September 24, 2021 |
| **2** | Blood & Water | Unknown Director | Ama Qamata | South Africa | TV Dramas | s2 | TV Show | September 24, 2021 |
| **3** | Blood & Water | Unknown Director | Ama Qamata | South Africa | TV Mysteries | s2 | TV Show | September 24, 2021 |

**Till here, data is sorted.** There are two main goals of this case study:

1. Provide data backed insights to grow the business.
2. Which type of movies/ shows should Netflix produce to meet the demand of the viewers. Should netflix focus more on Movies or on Shows.

Exploration: **Step 1:** Basic understanding of cleaned data -

```
#To get started, looking at the shape of the cleaned data.
df_final.shape
```

```
(202065, 12)
```

```
#converting to datetime format to perform some date related operations.
```

```
df_final["date_added"] = pd.to_datetime(df_final["date_added"], format="mixed")
```

```
# Looking at datatypes of different columns:
df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 202065 entries, 0 to 202064
Data columns (total 12 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   title         202065 non-null  object
 1   director      202065 non-null  object
 2   cast          202065 non-null  object
 3   country       202065 non-null  object
 4   listed_in     202065 non-null  object
 5   show_id       202065 non-null  object
 6   type          202065 non-null  object
 7   date_added    201907 non-null  datetime64[ns]
 8   release_year  202065 non-null  int64
 9   rating        201998 non-null  object
 10  duration      202065 non-null  object
 11  description   202065 non-null  object
dtypes: datetime64[ns](1), int64(1), object(10)
memory usage: 18.5+ MB
```

```
# Majority NaN values have been removed successfully.
df_final.isnull().sum()
```

```
title             0
director          0
cast              0
country           0
listed_in         0
show_id           0
type              0
date_added      158
release_year      0
rating           67
duration          0
description       0
dtype: int64
```

```
df_final.describe()
```

| | date_added | release_year |
|---|---|---|
| **count** | 201907 | 202065.000000 |
| **mean** | 2019-06-19 13:11:39.951958272 | 2013.448950 |
| **min** | 2008-01-01 00:00:00 | 1925.000000 |
| **25%** | 2018-06-25 00:00:00 | 2012.000000 |
| **50%** | 2019-09-01 00:00:00 | 2016.000000 |
| **75%** | 2020-09-10 00:00:00 | 2019.000000 |
| **max** | 2021-09-25 00:00:00 | 2021.000000 |
| **std** | NaN | 9.013616 |

```
df_final.nunique()
```

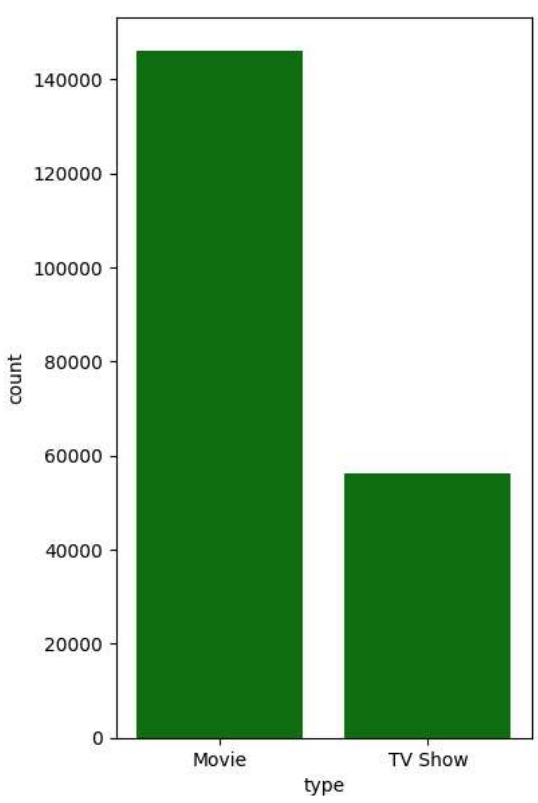
```
title            8807
director         5121
cast            39297
country           198
listed_in          73
show_id          8807
type                2
date_added       1714
release_year       74
rating             17
duration          220
description      8775
dtype: int64
```

```
#Q1: What is the total number of movies and shows on Netflix -
types = df_final["type"].value_counts()
```

```
types
```

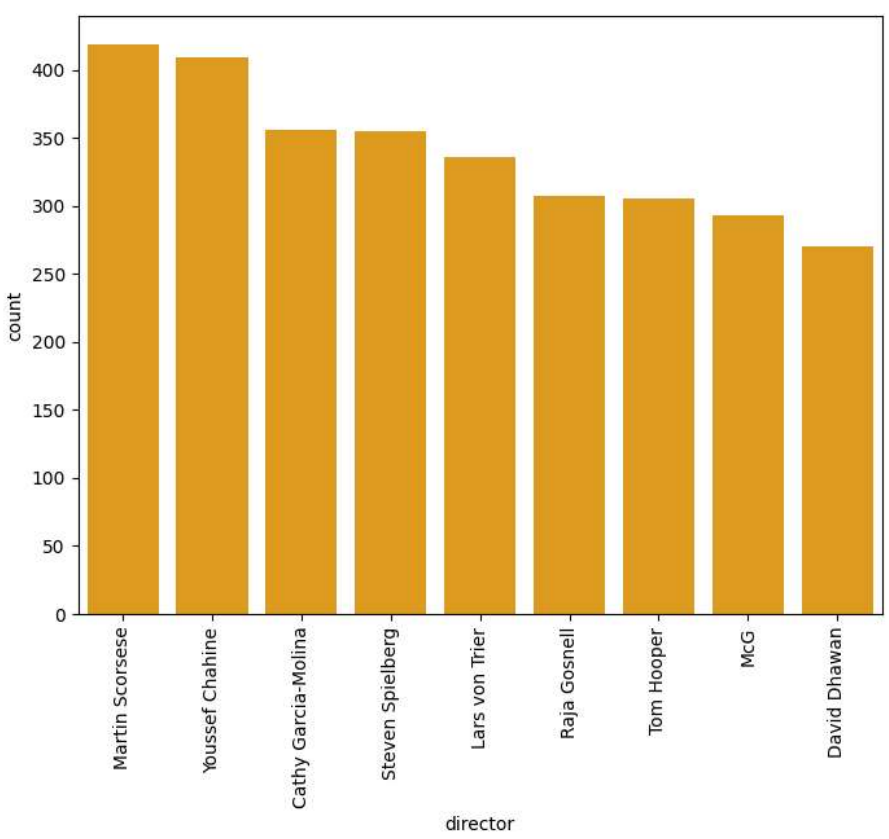
```
type
Movie      145917
TV Show     56148
Name: count, dtype: int64
```

```
plt.figure(figsize=(4, 7))
sns.barplot(data = types, color = "green")
plt.show()
```




```
#Q2: Most active director:
active_dir = df_final["director"].value_counts()[1:10]
```

```
plt.figure(figsize=(8, 6))
sns.barplot(data = active_dir, color = "orange")
plt.xticks(rotation = 90)
plt.show()
```




```
df_final[df_final["director"] == "Martin Scorsese"]["release_year"].min(), df_final[df_final["director"] == "Martin Scorsese"]["release_year
```

```
(1967, 2019)
```

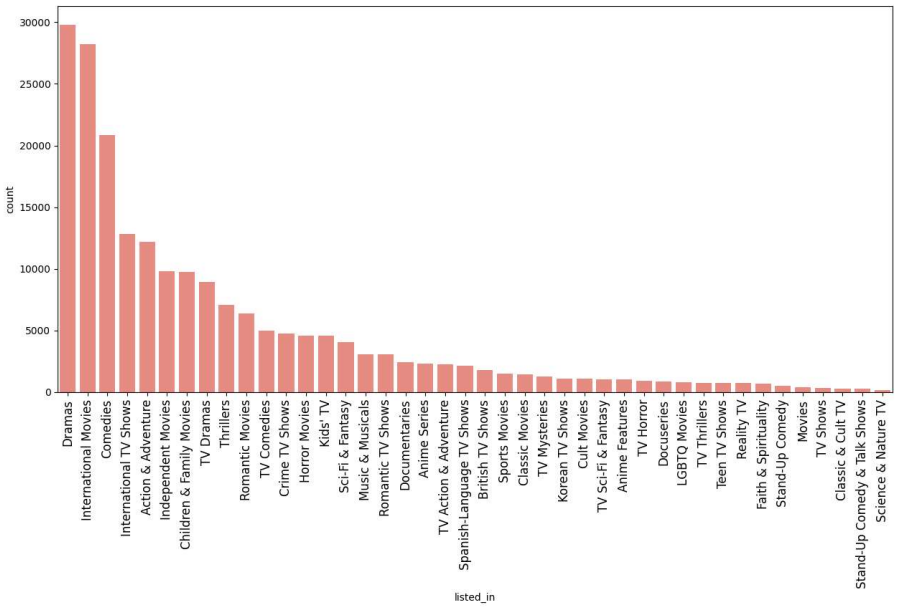
**Insights:** Martine Scorsese with 419 titles has the highest number especially given that he has been active from 1967 until 2019.

**Recommendations:** Since Martine Scorsese is a renowned popular director, Netflix should collaborate with him to produce a couple of movies.

```
#Let us dive a little deep into the data to get some insights by plotting Univariate graphs:
#Q3: Volume of content (Movies and shows) over the years:
genre_count = df_final['listed_in'].value_counts()
```

```
#Removing trailing and following whitespaces from genres.
df_final["listed_in"] = df_final["listed_in"].str.strip()
```

```
plt.figure(figsize = (15, 7))
plt.xticks(rotation = 90, fontsize = 12)
sns.countplot(x = "listed_in", data = df_final, order= df_final["listed_in"].value_counts().index, color = "salmon")
plt.show()
```

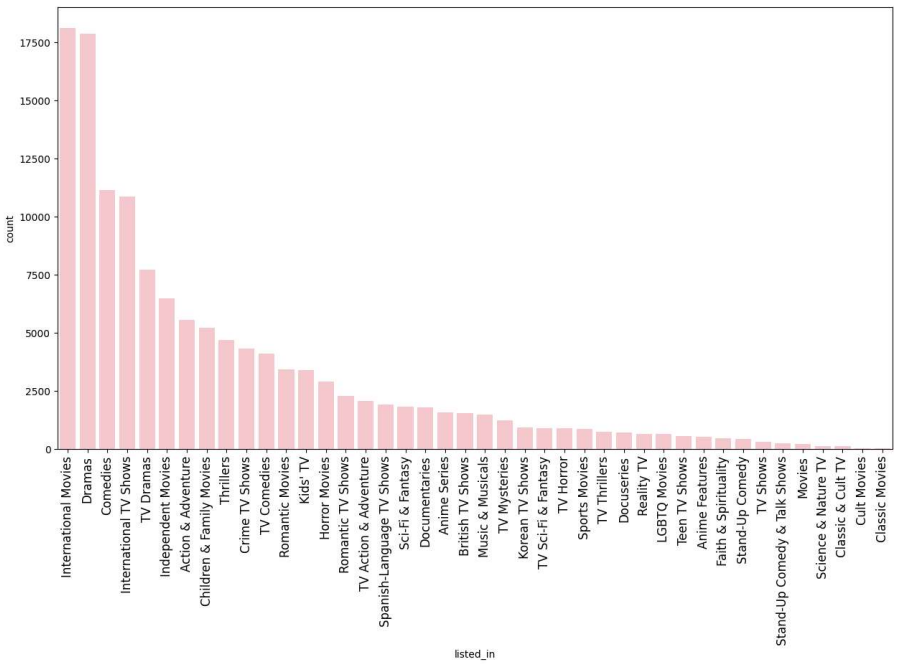



```
#Looking at the last 5-6 years data:
df_final_2015 = df_final[df_final["release_year"] >= 2015]
df_final_2015
```

| | title | director | cast | country | listed_in | show_id | type | date_added |
|---|---|---|---|---|---|---|---|---|
| **0** | Dick Johnson Is Dead | Kirsten Johnson | Unknown Actor | United States | Documentaries | s1 | Movie | 2021-09-25 |
| **1** | Blood & Water | Unknown Director | Ama Qamata | South Africa | International TV Shows | s2 | TV Show | 2021-09-24 |
| **2** | Blood & Water | Unknown Director | Ama Qamata | South Africa | TV Dramas | s2 | TV Show | 2021-09-24 |
| **3** | Blood & Water | Unknown Director | Ama Qamata | South Africa | TV Mysteries | s2 | TV Show | 2021-09-24 |

```
plt.figure(figsize = (15, 8))
plt.xticks(rotation = 90, fontsize = 12)
sns.countplot(x = "listed_in", data = df_final_2015, order= df_final_2015["listed_in"].value_counts().index, color = "pink")
plt.show()
```

```
plt.show()
```



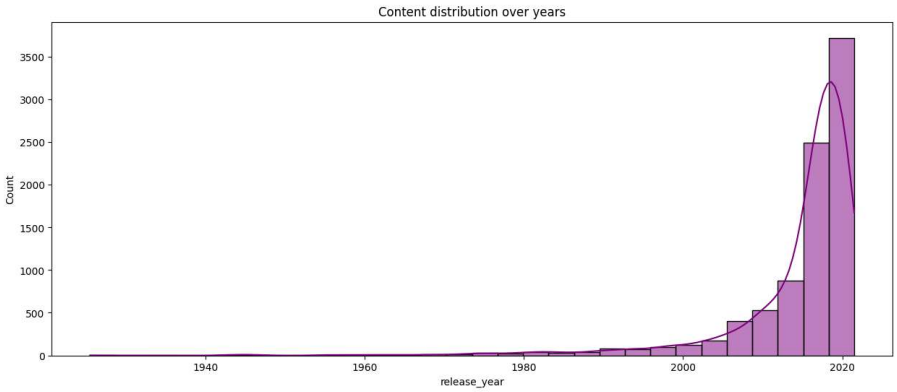

**Insights:**

1. International movies have the highest count of all time followed by Dramas, Comedies and Action & Adventure.
2. When we compare all time data and last 5 years data, there is a small shift: International movies have gained more popularity in the recent 6 years data.
3. When we compare with all time data, International TV shows, TV dramas, Crime tv shows have had a higher count in the recent past.

**Recommendations:**

1. Netflix should focus on collaborating more with foreign countries and not just USA since we see the rise that International movies have.
2. Netflix should focus more on producing International TV shows, TV dramas, Crime tv shows as they are rising in number in the recent years.
3. Also Teen tv shows and Reality shows are on the rise, Netflix should produce more of that content as well.

```
#Q: Looking at distribution of content over the years:
plt.figure(figsize = (15, 6))
sns.histplot(df["release_year"], bins = 30, kde = True, color = "purple")
plt.title("Content distribution over years")
plt.show()
```




```
df_final["country"] = df_final["country"].str.strip()
```

```
#Top 10 coutries for movies -
df_country = df_final.groupby("country")["type"].value_counts().sort_values(ascending = False).reset_index()
df_country
```

| | country | type | count |
|---|---|---|---|
| **0** | United States | Movie | 45817 |
| **1** | India | Movie | 21411 |
| **2** | United States | TV Show | 13533 |
| **3** | United Kingdom | Movie | 8580 |
| **4** | France | Movie | 6607 |
| **...** | ... | ... | ... |
| **181** | Ukraine | Movie | 2 |
| **182** | Palestine | Movie | 2 |
| **183** | Kazakhstan | Movie | 1 |
| **184** | Uganda | Movie | 1 |
| **185** | Nicaragua | Movie | 1 |

186 rows × 3 columns

```
df_country[df_country["type"] == "Movie"].reset_index().drop("index", axis =1).head(11)
```

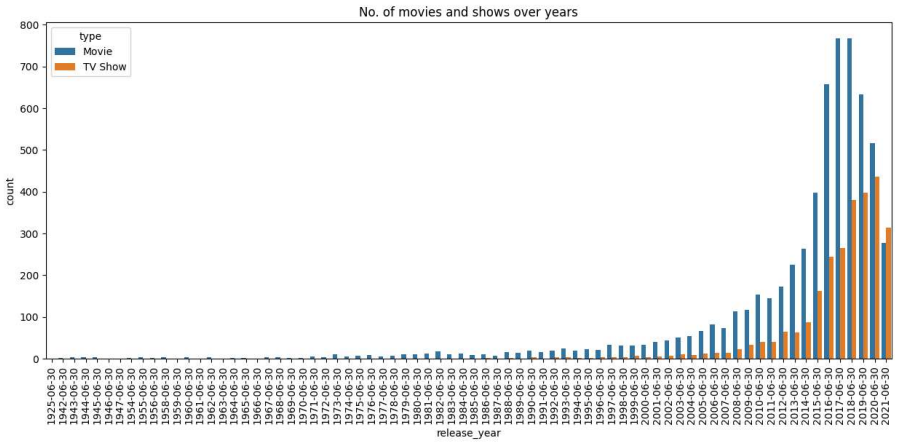
| | country | type | count |
|---|---|---|---|
| 0 | United States | Movie | 45817 |
| 1 | India | Movie | 21411 |
| 2 | United Kingdom | Movie | 8580 |
| 3 | France | Movie | 6607 |
| 4 | Unknown Country | Movie | 6199 |
| 5 | Canada | Movie | 5738 |
| 6 | Japan | Movie | 3525 |
| 7 | Spain | Movie | 3469 |
| 8 | Germany | Movie | 3427 |
| 9 | China | Movie | 2377 |
| 10 | Nigeria | Movie | 2236 |

```
#Converting to datetime format.
df["date_added"] = pd.to_datetime(df["date_added"], format='mixed')
```

```
#Similarly lets find for - TV Shows -
df_country[df_country["type"] == "TV Show"].reset_index().drop("index", axis =1).head(11)
```

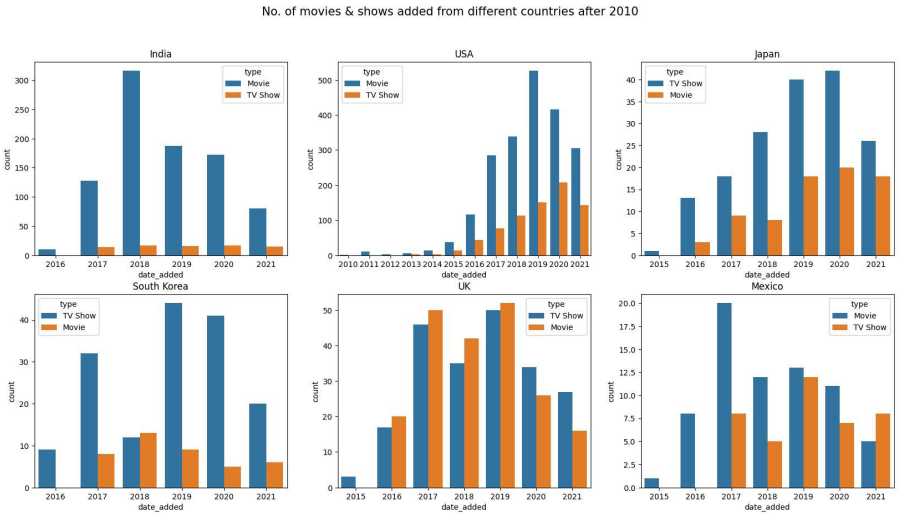
| | country | type | count |
|---|---|---|---|
| 0 | United States | TV Show | 13533 |
| 1 | Unknown Country | TV Show | 5698 |
| 2 | Japan | TV Show | 5154 |
| 3 | United Kingdom | TV Show | 4385 |
| 4 | South Korea | TV Show | 3754 |
| 5 | Canada | TV Show | 2177 |
| 6 | Mexico | TV Show | 2018 |
| 7 | Spain | TV Show | 1846 |
| 8 | Taiwan | TV Show | 1719 |
| 9 | France | TV Show | 1647 |
| 10 | India | TV Show | 1403 |

```
#Looking at the same graphically -
plt.figure(figsize = (15,6))
sns.countplot(data = df, x = "release_year", hue = "type", order = sorted(df["release_year"].unique()))
plt.title("No. of movies and shows over years")
plt.xticks(rotation = 90)
plt.show()
```




```
#Q: Perform a graphical comparison between movies and tv shows for the last 10 years for the top 6 countries:
df_india = df[((df["country"] == "India") & (df["date_added"] > "2010-01-01"))]
df_usa = df[((df["country"] == "United States") & (df["date_added"] > "2010-01-01"))]
df_japan = df[((df["country"] == "Japan") & (df["date_added"] > "2010-01-01"))]
df_sk = df[((df["country"] == "South Korea") & (df["date_added"] > "2010-01-01"))]
df_uk = df[((df["country"] == "United Kingdom") & (df["date_added"] > "2010-01-01"))]
df_mex = df[((df["country"] == "Mexico") & (df["date_added"] > "2010-01-01"))]
```

```
#let us visualise it in the form of dodged countplot for the 6 countries -
plt.figure(figsize = (20, 10)).suptitle("No. of movies & shows added from different countries after 2010", fontsize = 15)

plt.subplot(2, 3, 1)
plt.title("India")
sns.countplot(x = df_india["date_added"].dt.year, hue = "type", data = df_india)

plt.subplot(2, 3, 2)
plt.title("USA")
sns.countplot(x = df_usa["date_added"].dt.year, hue = "type", data = df_usa)

plt.subplot(2, 3, 3)
plt.title("Japan")
sns.countplot(x = df_japan["date_added"].dt.year, hue = "type", data = df_japan)

plt.subplot(2, 3, 4)
plt.title("South Korea")
sns.countplot(x = df_sk["date_added"].dt.year, hue = "type", data = df_sk)

plt.subplot(2, 3, 5)
plt.title("UK")
sns.countplot(x = df_uk["date_added"].dt.year, hue = "type", data = df_uk)

plt.subplot(2, 3, 6)
plt.title("Mexico")
sns.countplot(x = df_mex["date_added"].dt.year, hue = "type", data = df_mex)

plt.show()
```




**Insights:**

1. Netflix has added a good amount of content in the last 5 years from countries like India, UK, Mexico, South Korea and Japan.This shows that they have come out of US and recognized content from other countries.
2. In the US, there has been huge splurge of content post 2015. This can be attributed to the revolution of high internet speeds in the last 5 years and Netflix has made good use of it.
3. Only in US, there is an upward trend of TV shows, but same consistent increase is not seen in other countries (its mostly constant).
4. There seems to be a dip in count after 2020, this could be attributed to the pandemic, hence understable.

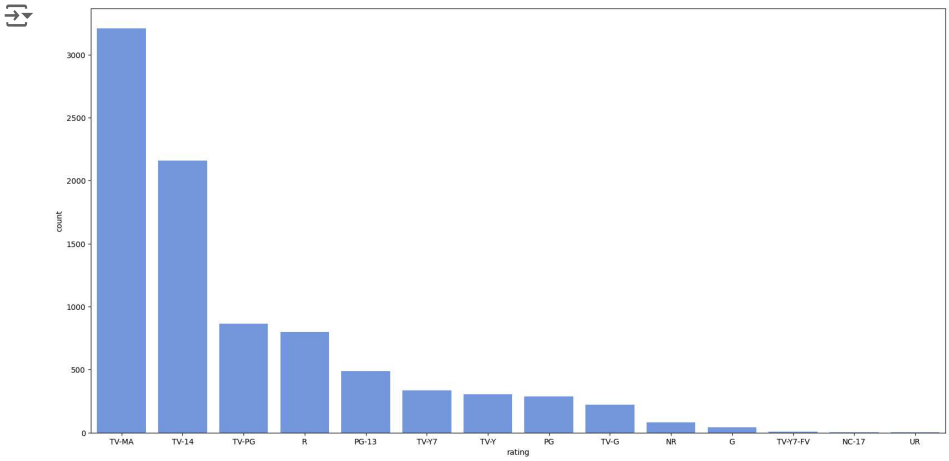
**Recommendations:**

1. Netflix should focus on foreign countries content, tie up with local content based on the popularity they already have and should produce more in these regions, particularly Asia, given that they constitute nearly 59% of total world population (Source: Google).
2. There is a stagnancy is the amount of tv shows being added/ released in countries other than US and Japan when compared between 2019 and 2021. This should be corrected and Netflix needs to redistribute/ pour more funds into countries like India and South Korea.
3. Netflix needs to create a strong subscriber base in Asian countries given their love for movies and shows and cheap internet. To do this, they need to produce quality content and provide subsriptions at a lower price (for handheld devices).

```
#Let us see the what kind of movies/ tv shows ratings are most made

df_rating = df["rating"].value_counts().reset_index().head(14)
df_rating
```

| | rating | count |
|---|---|---|
| 0 | TV-MA | 3207 |
| 1 | TV-14 | 2160 |
| 2 | TV-PG | 863 |
| 3 | R | 799 |
| 4 | PG-13 | 490 |
| 5 | TV-Y7 | 334 |
| 6 | TV-Y | 307 |
| 7 | PG | 287 |
| 8 | TV-G | 220 |
| 9 | NR | 80 |
| 10 | G | 41 |
| 11 | TV-Y7-FV | 6 |
| 12 | NC-17 | 3 |
| 13 | UR | 3 |

```
plt.figure(figsize = (20,10))
sns.barplot(x = "rating", y = "count", data = df_rating, color = "cornflowerblue")
plt.show()
```




**Insights:**

1. Highest count is of TV-MA, TV-14.
2. TV-PG, PG-13 and PG have really low content.

**Recommendations:**

1. Netflix seems to be more focussed on producing content for adults and not much for the teenage group. This needs to be addressed as we know how the youth is consuming content these days.
2. Netflix needs to be responsible with the kind of content they provide and ensure that the youth gets to view the right kind of content by enabling parental locks for adult content. This will increase parents' trust on Netflix and good will certainly leads to more business.

```
#q: Best week to launch a Movie/ tv show -
df["date_added"] = pd.to_datetime(df["date_added"], format='mixed')
df["week"] = df["date_added"].dt.isocalendar().week
```

```
week_mov_count = df[df["type"] == "Movie"].groupby("week")["title"].agg("count").sort_values(ascending = False).reset_index()
week_mov_count
```

| | week | title |
|---|---|---|
| **0** | 1 | 316 |
| **1** | 44 | 243 |
| **2** | 40 | 215 |
| **3** | 9 | 207 |
| **4** | 26 | 195 |
| **5** | 35 | 189 |
| **6** | 31 | 185 |
| **7** | 13 | 174 |
| **8** | 18 | 173 |
| **9** | 27 | 154 |
| **10** | 22 | 146 |
| **11** | 48 | 139 |
| **12** | 5 | 135 |
| **13** | 14 | 124 |
| **14** | 16 | 124 |
| **15** | 50 | 119 |
| **16** | 30 | 116 |
| **17** | 11 | 115 |
| **18** | 37 | 114 |
| **19** | 23 | 112 |
| **20** | 39 | 111 |
| **21** | 17 | 109 |
| **22** | 10 | 107 |
| **23** | 7 | 106 |
| **24** | 33 | 105 |
| **25** | 34 | 102 |
| **26** | 25 | 101 |
| **27** | 15 | 100 |
| **28** | 36 | 97 |
| **29** | 49 | 95 |
| **30** | 29 | 94 |
| **31** | 42 | 90 |
| **32** | 28 | 89 |
| **33** | 24 | 89 |
| **34** | 38 | 88 |
| **35** | 43 | 88 |
| **36** | 51 | 86 |
| **37** | 20 | 85 |
| **38** | 47 | 85 |
| **39** | 41 | 84 |
| **40** | 46 | 83 |
| **41** | 3 | 81 |
| **42** | 52 | 80 |
| **43** | 2 | 78 |
| **44** | 21 | 76 |
| **45** | 32 | 73 |

| | | |
|---|---|---|
| **46** | 19 | 73 |
| **47** | 8 | 72 |
| **48** | 12 | 67 |
| **49** | 6 | 64 |
| **50** | 45 | 61 |
| **51** | 53 | 61 |
| **52** | 4 | 56 |

**Insights:**

1. Based on data, the best time to launch a movie would be first week of Jan and the month of October .

**Recommendations:**

1. Netflix should release new movies in Jan 1st week. This is ideal time as people would have just come back home after vacations and would not want to step out and might be scrolling for new releases.
2. Oct (Week 44) is also a good time as this is around Halloween.
3. For a country like India, Netflix should release movies during festival times and during Independence Day or Republic day. This will certainly drive up their viewership.

```
week_ts_count = df[df["type"] == "TV Show"].groupby("week")["title"].agg("count").sort_values(ascending = False).reset_index()
week_ts_count
```

| | week | title |
|---|---|---|
| **0** | 27 | 86 |
| **1** | 31 | 83 |
| **2** | 13 | 76 |
| **3** | 44 | 75 |
| **4** | 24 | 75 |
| **5** | 35 | 74 |
| **6** | 5 | 73 |
| **7** | 26 | 73 |
| **8** | 40 | 72 |
| **9** | 50 | 70 |
| **10** | 37 | 69 |
| **11** | 18 | 61 |
| **12** | 48 | 60 |
| **13** | 22 | 60 |
| **14** | 1 | 56 |
| **15** | 39 | 55 |
| **16** | 15 | 52 |
| **17** | 52 | 52 |
| **18** | 51 | 51 |
| **19** | 38 | 51 |
| **20** | 46 | 51 |
| **21** | 32 | 49 |
| **22** | 14 | 49 |
| **23** | 33 | 48 |
| **24** | 11 | 48 |
| **25** | 9 | 47 |
| **26** | 29 | 46 |
| **27** | 20 | 46 |
| **28** | 17 | 45 |
| **29** | 49 | 45 |
| **30** | 36 | 45 |
| **31** | 42 | 45 |
| **32** | 30 | 44 |
| **33** | 53 | 43 |
| **34** | 19 | 43 |
| **35** | 25 | 42 |
| **36** | 12 | 42 |
| **37** | 34 | 41 |
| **38** | 28 | 41 |
| **39** | 7 | 41 |
| **40** | 21 | 41 |
| **41** | 23 | 39 |
| **42** | 8 | 38 |
| **43** | 45 | 37 |
| **44** | 16 | 36 |
| **45** | 47 | 35 |

| | | |
|---|---|---|
| **46** | 6 | 33 |
| **47** | 4 | 32 |
| **48** | 3 | 32 |
| **49** | 41 | 32 |
| **50** | 2 | 30 |
| **51** | 10 | 28 |
| **52** | 43 | 28 |

**Insights:**

1. Based on data, the best time to release a tv show would be July first week.

**Recommendations:**

1. Netflix should release new shows during July 1st week. This is ideal time as this is Independence Day (United States).
2. For a country like India, Netflix should release movies during festival times and during Independence Day or Republic day. This will certainly drive up their viewership.
3. Release shows/ movies during a particular nation's holiday season.

```
df["month"] = df["date_added"].dt.month_name()
```

```
month_mov_count = df[df["type"] == "Movie"].groupby("month")["title"].agg("count").sort_values(ascending = False).reset_index()
month_mov_count
```

| | month | title |
|---|---|---|
| **0** | July | 565 |
| **1** | April | 550 |
| **2** | December | 547 |
| **3** | January | 546 |
| **4** | October | 545 |
| **5** | March | 529 |
| **6** | August | 519 |
| **7** | September | 519 |
| **8** | November | 498 |
| **9** | June | 492 |
| **10** | May | 439 |
| **11** | February | 382 |

```
month_ts_count = df[df["type"] == "TV Show"].groupby("month")["title"].agg("count").sort_values(ascending = False).reset_index()
month_ts_count
```

| | month | title |
|---|---|---|
| 0 | December | 266 |
| 1 | July | 262 |
| 2 | September | 251 |
| 3 | August | 236 |
| 4 | June | 236 |
| 5 | October | 215 |
| 6 | April | 214 |
| 7 | March | 213 |
| 8 | November | 207 |
| 9 | May | 193 |
| 10 | January | 192 |
| 11 | February | 181 |

**Insights:**

1. Most movies or shows are getting released during the months of July, December, January and October.

**Recommendations:**

1. These are festive seasons, hence most movies are added in that season. Similarly for other countries, Netflix should release shows/ movies during a that nation's holiday season to attract viewers.

```
#Q: Top 10 actors appearing in most shows/ Movies
top_actors = df_final.groupby("cast")["title"].agg("count").sort_values(ascending = False).reset_index()[1:11]
top_actors
```

| | cast | title |
|---|---|---|
| 1 | Alfred Molina | 160 |
| 2 | Salma Hayek | 130 |
| 3 | Frank Langella | 128 |
| 4 | John Rhys-Davies | 125 |
| 5 | John Krasinski | 121 |
| 6 | Liam Neeson | 120 |
| 7 | Anupam Kher | 116 |
| 8 | David Attenborough | 103 |
| 9 | Quvenzhané Wallis | 100 |
| 10 | James Faulkner | 93 |

**Insights:**

1. Alfred Molina, Salma Hayek, Frank Langella, John Krasinski, Liam Neeson are actors that appear most in movies and shows.

Recommendations:

1. Clearly these are household names and Netflix should collaborate more with these actors so that Netflix would appear on the radar of more and more people.
2. Anupam Kher, an Indian actor is at No.7. He is very popular and hence Netflix should focus on casting more such legendary actors whose presence in the movies cast would certainly increase its fame, hence would attract more viewers.

```
#Q: Top 10 actors appearing directing most shows/ Movies
top_dir = df_final.groupby("director")["title"].agg("count").sort_values(ascending = False).reset_index()[1:11]
top_dir
```

| | director | title |
|---|---|---|
| 1 | Martin Scorsese | 419 |
| 2 | Youssef Chahine | 409 |
| 3 | Cathy Garcia-Molina | 356 |
| 4 | Steven Spielberg | 355 |
| 5 | Lars von Trier | 336 |
| 6 | Raja Gosnell | 308 |
| 7 | Tom Hooper | 306 |
| 8 | McG | 293 |
| 9 | David Dhawan | 270 |
| 10 | Wilson Yip | 260 |

**Insights:**

1. Its no surprise that veteran directors like Martin Scorsese (US), Youssef Chahine(Egypt), Steven Spielberg (US), Lars von Trier (Denmark) and David Dhawan (India) are in top 10 list.

**Recommendations:**

1. Netflix should look at these directors carefully, appoint local Managers for each region, because these directors might have been legends in their time, but they might not have the same level of creativity now, hence these have to be approached with caution.

```
popular_genre_mov = df_final[df_final["type"] == "Movie"]['listed_in'].value_counts().reset_index()
popular_genre_mov
```

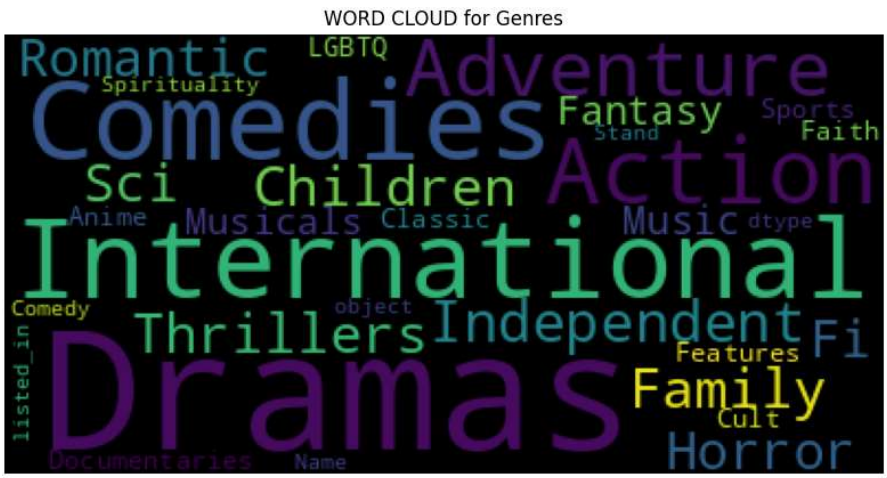
| | listed_in | count |
|---|---|---|
| 0 | Dramas | 29806 |
| 1 | International Movies | 28243 |
| 2 | Comedies | 20829 |
| 3 | Action & Adventure | 12216 |
| 4 | Independent Movies | 9834 |
| 5 | Children & Family Movies | 9771 |
| 6 | Thrillers | 7107 |
| 7 | Romantic Movies | 6412 |
| 8 | Horror Movies | 4571 |
| 9 | Sci-Fi & Fantasy | 4037 |
| 10 | Music & Musicals | 3077 |
| 11 | Documentaries | 2409 |
| 12 | Sports Movies | 1531 |
| 13 | Classic Movies | 1443 |
| 14 | Cult Movies | 1077 |
| 15 | Anime Features | 1045 |
| 16 | LGBTQ Movies | 838 |
| 17 | Faith & Spirituality | 719 |
| 18 | Stand-Up Comedy | 540 |
| 19 | Movies | 412 |

```
popular_genre_mov["listed_in"] = popular_genre_mov["listed_in"].str.replace("Movies", " ")
```

```
popular_genre_mov
```

| | listed_in | count |
|---|---|---|
| 0 | Dramas | 29806 |
| 1 | International | 28243 |
| 2 | Comedies | 20829 |
| 3 | Action & Adventure | 12216 |
| 4 | Independent | 9834 |
| 5 | Children & Family | 9771 |
| 6 | Thrillers | 7107 |
| 7 | Romantic | 6412 |
| 8 | Horror | 4571 |
| 9 | Sci-Fi & Fantasy | 4037 |
| 10 | Music & Musicals | 3077 |
| 11 | Documentaries | 2409 |
| 12 | Sports | 1531 |
| 13 | Classic | 1443 |
| 14 | Cult | 1077 |
| 15 | Anime Features | 1045 |
| 16 | LGBTQ | 838 |
| 17 | Faith & Spirituality | 719 |
| 18 | Stand-Up Comedy | 540 |
| 19 | | 412 |

```
from wordcloud import WordCloud
plt.figure(figsize = (20, 5))

wordcloud = WordCloud(
                      background_color = 'black',
                      max_font_size = 75,
                      random_state = 40
                     ).generate(str(popular_genre_mov["listed_in"]))
plt.imshow(wordcloud)
plt.title("WORD CLOUD for Genres", fontsize = 12)
plt.axis('off')
plt.show()
```




**Insights:**

1. From the word cloud, it is clear that some of the popular genres include Dramas,Comedies, Action and Adventure and Thriller.

**Recommendations:**

1. Netflix should identify which director has given most hits in a particular genre and then that director can be approached for making that movie/ show.

```
#Q: Finding how many days a movie/show will be added to Netflix after its release
df["additional"] = "-06-30"
df["release_year"] = df["release_year"].apply(str)
df["release_year"] = df["release_year"] + df["additional"]
```

```
df["release_year"] = pd.to_datetime(df["release_year"])
```

```
df["days_diff"] = df["date_added"] - df["release_year"]
```

```
df["days_diff"].sort_values(ascending = False)
```

```
4250   34151 days
1331   27618 days
7790   27303 days
8205   27303 days
8763   26938 days
          ...
7196          NaT
7254          NaT
7406          NaT
7847          NaT
8182          NaT
Name: days_diff, Length: 8807, dtype: timedelta64[ns]
```

```
recent_year = df[df["release_year"].dt.year > 2010] #considering only recent data.
```

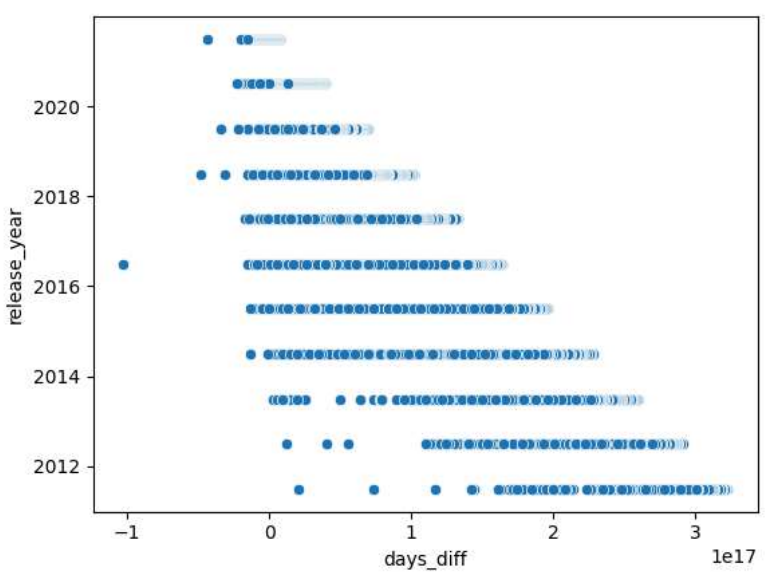
```
recent_year["days_diff"].mode()
```

```
0   154 days
Name: days_diff, dtype: timedelta64[ns]
```

```
#Where the days difference is very small -
recent_year[(recent_year["days_diff"] > "0 days") & (recent_year["days_diff"] < "15 days")]
```

| | show_id | type | title | director | cast | country | date_added | release_yea |
|---|---|---|---|---|---|---|---|---|
| **463** | s464 | Movie | A Classic Horror Story | Roberto De Feo, Paolo Strippoli | Matilda Lutz, Francesco Russo, Peppino Mazzott... | Italy | 2021-07-14 | 2021-06-3 |
| **464** | s465 | Movie | Gunpowder Milkshake | Navot Papushado | Karen Gillan, Lena Headey, Carla Gugino, Chloe... | NaN | 2021-07-14 | 2021-06-3 |
| **465** | s466 | TV Show | Heist | NaN | NaN | NaN | 2021-07-14 | 2021-06-3 |
| **466** | s467 | TV Show | My Unorthodox Life | NaN | NaN | NaN | 2021-07-14 | 2021-06-3 |
| **467** | s468 | Movie | Private Network: Who Killed Manuel Buendía? | Manuel Alcalá | Daniel Giménez Cacho | NaN | 2021-07-14 | 2021-06-3 |

```
sns.scatterplot(x = "days_diff", y = "release_year", data = recent_year)
plt.show()
```




**Insights:**

1. The highest days difference between release data and added date is in the range 30 to 40 years. But these are old movies release in early 1900s and Netflix as a company was established only in 2007.
2. As the years go by, the days difference between release date and added date is decreasing. The mode (highest frequency) is 154 days as determined above.

**Recommendations:**

1. The days difference is on the higher side. Netflix should add a movie within 1 or 2 months of the movie releasing so that the momentum of popularity of a movie would be carried over to Netflix as well. This would eventually attract more viewers as not all would have watched all movies in theaters and also there would be repeat viewers, Netflix would benefit from both.
2. The difference should not be too less, 10-15 days of the movie release (As seen above in the case of 130 movies and shows). This is because, if viewers get a feeling that they can watch any movie on Netflix, then they wouldnt want to go to theaters and this would eventually hurt Netflix as the popularity would not be that high.

## --------END OF BUSINESS CASE STUDY--------

The days difference is on the higher side. Netflix should add a movie within 1 or 2 months of the movie releasing so that the momentum of popularity of a movie would be carried over to Netflix as well. This would eventually attract more viewers as not all would have watched all movies in theaters and also there would be repeat viewers, Netflix would benefit from both.

The difference should not be too less, 10-15 days of the movie release (As seen above in the case of 130 movies and shows). This is because, if viewers get a feeling that they can watch any movie on Netflix, then they wouldnt want to go to theaters and this would eventually hurt Netflix as the popularity would not be that high. The days difference is on the higher side. Netflix should add a movie within 1 or 2 months of the movie releasing so that the momentum of popularity of a movie would be carried over to Netflix as well. This would eventually attract more viewers as not all would have watched all movies in theaters and also there would be repeat viewers, Netflix would benefit from both.

The difference should not be too less, 10-15 days of the movie release (As seen above in the case of 130 movies and shows). This is because, if viewers get a feeling that they can watch any movie on Netflix, then they wouldnt want to go to theaters and this would eventually hurt Netflix as the popularity would not be that high. The days difference is on the higher side. Netflix should add a movie within 1 or 2 months of the movie releasing so that the momentum of popularity of a movie would be carried over to Netflix as well. This would eventually attract more viewers as not all would have watched all movies in theaters and also there would be repeat viewers, Netflix would benefit from both.

The difference should not be too less, 10-15 days of the movie release (As seen above in the case of 130 movies and shows). This is because, if viewers get a feeling that they can watch any movie on Netflix, then they wouldnt want to go to theaters and this would eventually hurt Netflix as the popularity would not be that high.

The days difference is on the higher side. Netflix should add a movie within 1 or 2 months of the movie releasing so that the momentum of popularity of a movie would be carried over to Netflix as well. This would eventually attract more viewers as not all would have watched all