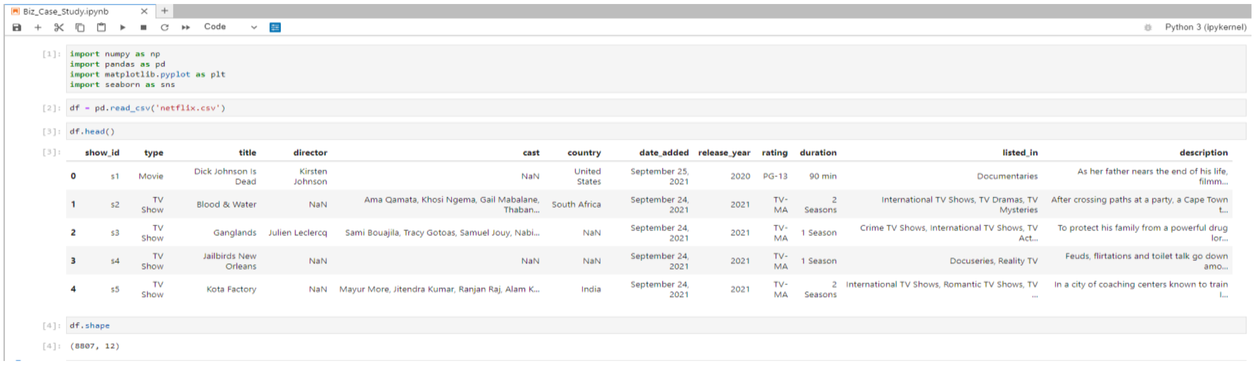
Business Case: Netflix - Data Exploration and Visualization

After loading the file, we can see as below,



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

[2]: df = pd.read_csv('netflix.csv')

[3]: df.head()
```

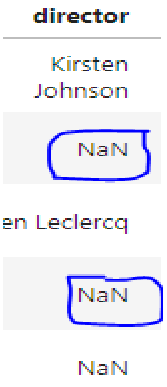
[3]:

	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...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

```
[4]: df.shape

[4]: (8807, 12)
```

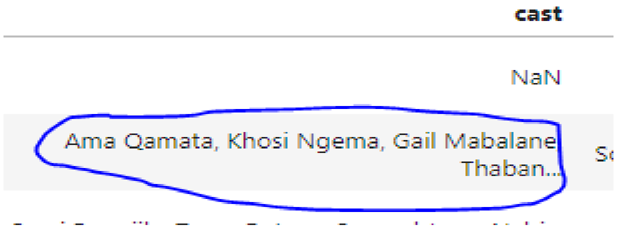
From the file 'netflix.csv' we can see it have some issue regarding the data,

1. 'Nan' values are present in the director, cast, country etc. columns.



director
Kirsten Johnson
NaN
en Leclercq
NaN
NaN

2. Some columns like cast, listed in, country, director etc. have nested data like,

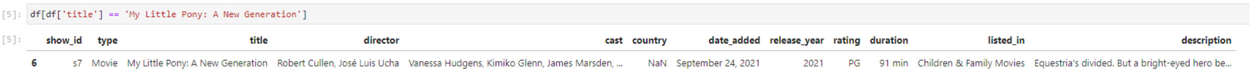


cast	
NaN	
Ama Qamata, Khosi Ngema, Gail Mabalane Thaban...	Sc

To analyze the data, we need to clean the raw data first. We will do the below operations,

Operation 01: Un-nesting the columns,

i. Director:

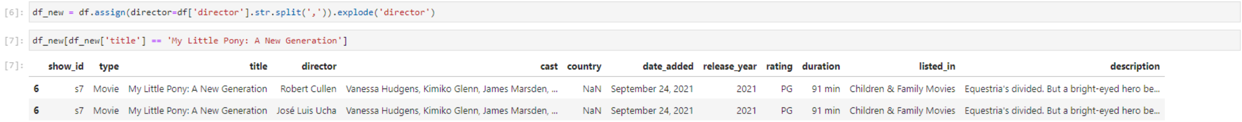


```
[5]: df[df['title'] == 'My Little Pony: A New Generation']
```

[5]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...

Here we can see for this movie two director present, after un-nesting the column we are getting as below,

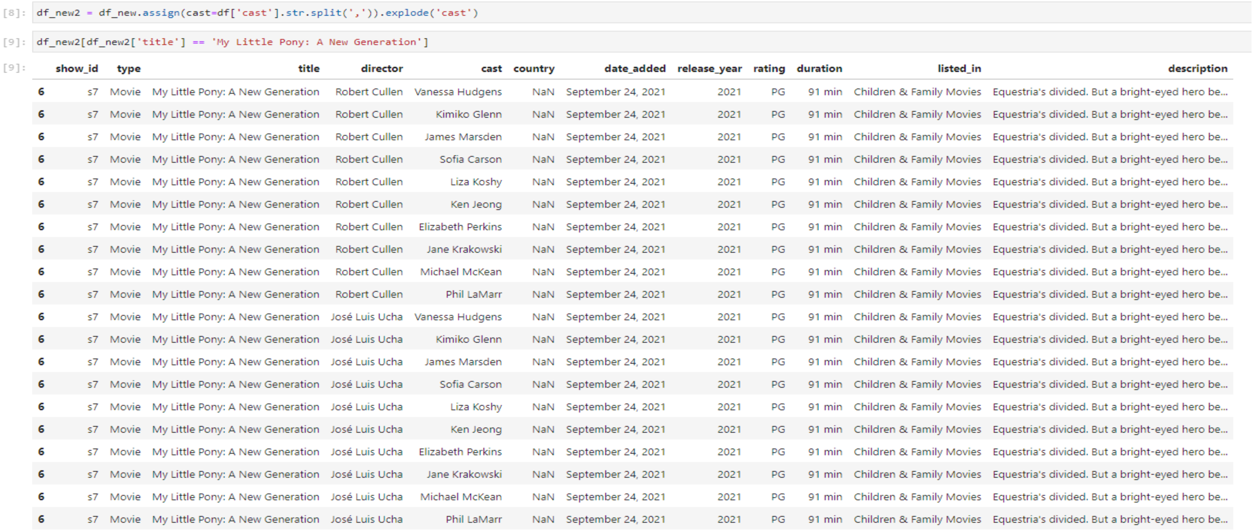


```
[6]: df_new = df.assign(director=df['director'].str.split(',')).explode('director')

[7]: df_new[df_new['title'] == 'My Little Pony: A New Generation']
```

[7]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...

ii. Cast:

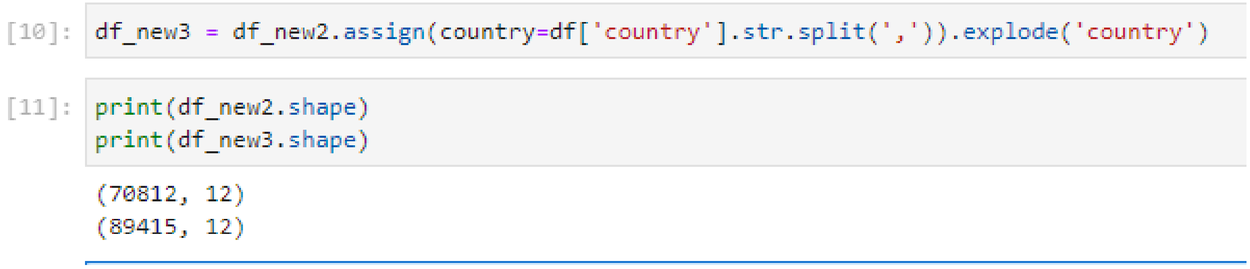


```
[8]: df_new2 = df_new.assign(cast=df['cast'].str.split(',')).explode('cast')

[9]: df_new2[df_new2['title'] == 'My Little Pony: A New Generation']
```

[9]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Vanessa Hudgens	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Kimiko Glenn	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen	James Marsden	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Sofia Carson	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Liza Koshy	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Ken Jeong	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Elizabeth Perkins	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Jane Krakowski	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Michael McKean	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen	Phil LaMarr	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	José Luis Ucha	Vanessa Hudgens	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	José Luis Ucha	Kimiko Glenn	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	José Luis Ucha	James Marsden	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	José Luis Ucha	Sofia Carson	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	José Luis Ucha	Liza Koshy	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	José Luis Ucha	Ken Jeong	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	José Luis Ucha	Elizabeth Perkins	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	José Luis Ucha	Jane Krakowski	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	José Luis Ucha	Michael McKean	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
6	s7	Movie	My Little Pony: A New Generation	José Luis Ucha	Phil LaMarr	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...

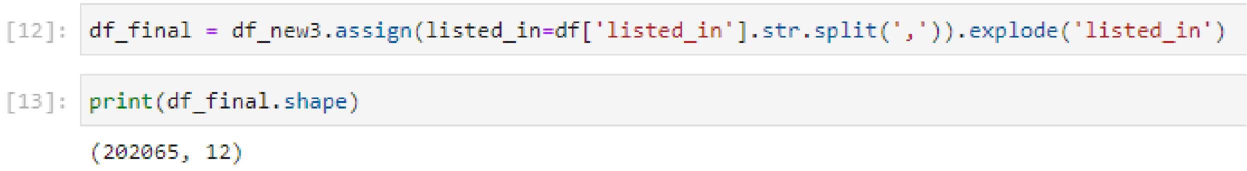
iii. Country:



```
[10]: df_new3 = df_new2.assign(country=df['country'].str.split(',')).explode('country')

[11]: print(df_new2.shape)
      print(df_new3.shape)

      (70812, 12)
      (89415, 12)
```

Same operation we have done for country also and we can see the row count increases.

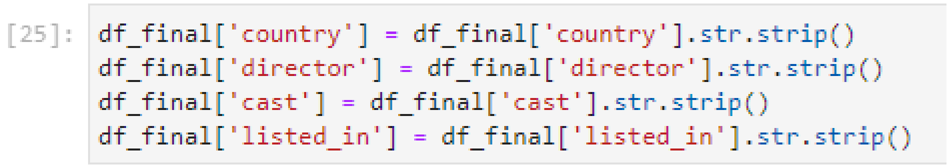
iv. Listed in:



```
[12]: df_final = df_new3.assign(listed_in=df['listed_in'].str.split(',')).explode('listed_in')

[13]: print(df_final.shape)

      (202065, 12)
```

My final data frame has now 202065 number of rows.

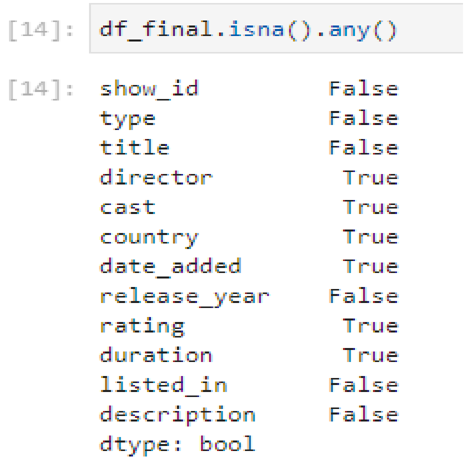
Let’s trim the whitespaces,



```
[25]: df_final['country'] = df_final['country'].str.strip()
      df_final['director'] = df_final['director'].str.strip()
      df_final['cast'] = df_final['cast'].str.strip()
      df_final['listed_in'] = df_final['listed_in'].str.strip()
```

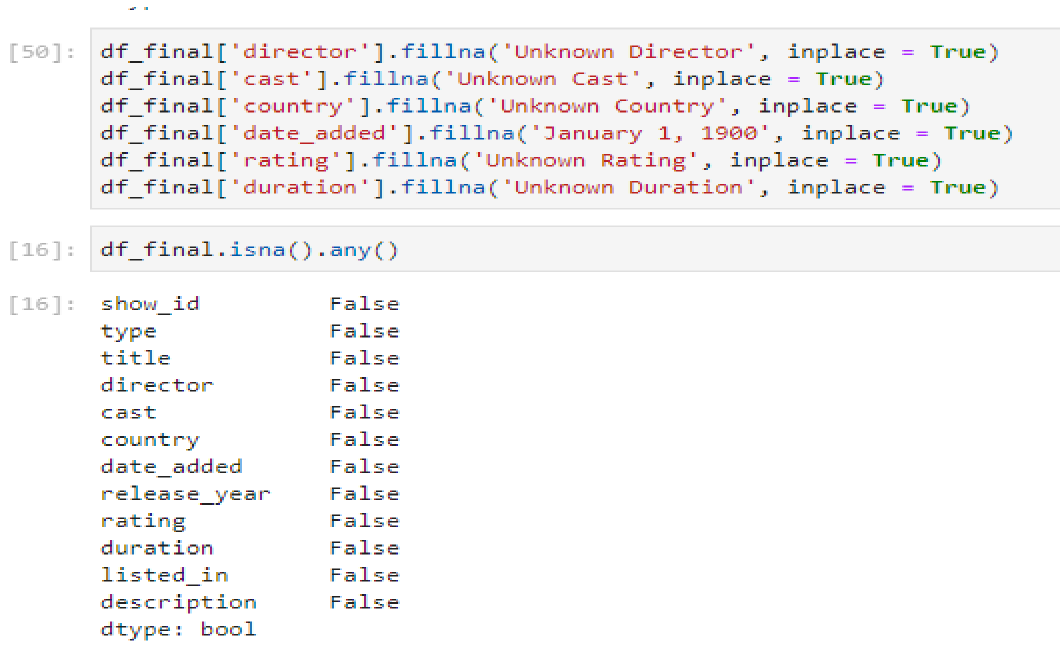
Operation 02:

Now we need to remove the ‘NaN’ values from the data frame.



```
[14]: df_final.isna().any()

[14]: show_id         False
      type            False
      title           False
      director         True
      cast             True
      country          True
      date_added       True
      release_year    False
      rating           True
      duration         True
      listed_in       False
      description     False
      dtype: bool
```

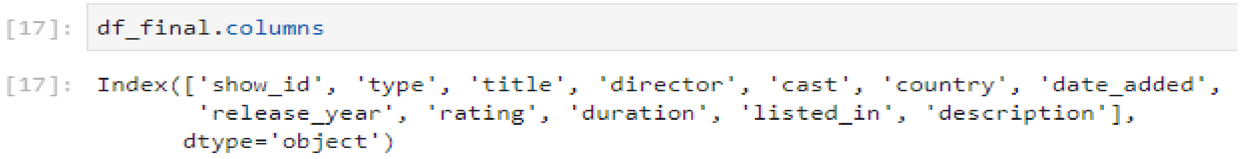
From above we can see some columns have Nan values, we need to handle those. As all the columns are categorical columns hence filled with ‘Unknown Column Name’,



```
[50]: df_final['director'].fillna('Unknown Director', inplace = True)
      df_final['cast'].fillna('Unknown Cast', inplace = True)
      df_final['country'].fillna('Unknown Country', inplace = True)
      df_final['date_added'].fillna('January 1, 1900', inplace = True)
      df_final['rating'].fillna('Unknown Rating', inplace = True)
      df_final['duration'].fillna('Unknown Duration', inplace = True)

[16]: df_final.isna().any()

[16]: show_id         False
      type            False
      title           False
      director        False
      cast            False
      country         False
      date_added      False
      release_year    False
      rating          False
      duration        False
      listed_in       False
      description     False
      dtype: bool
```

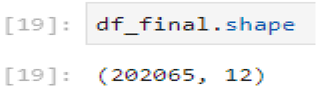
As we prepared the data, now let’s analyze the data.

1. Find the counts of each categorical variable both using graphical and non-graphical analysis



```
[17]: df_final.columns
```

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

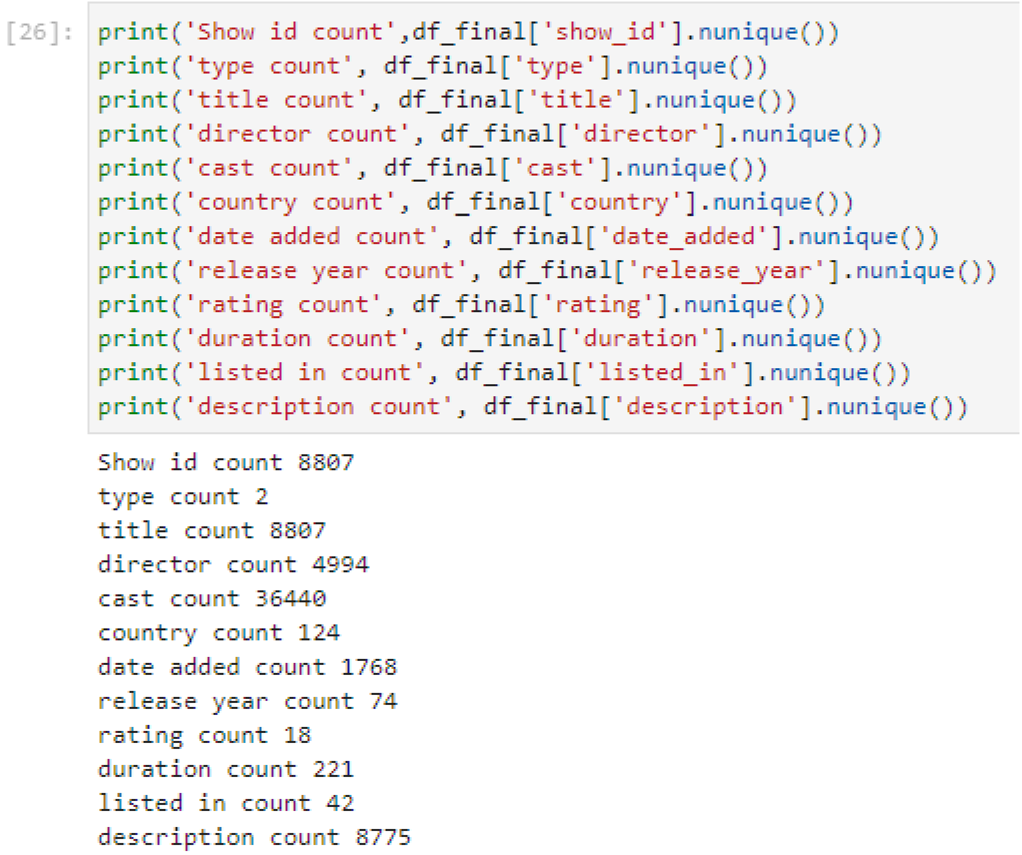


```
[19]: df_final.shape
```

```
[19]: (202065, 12)
```

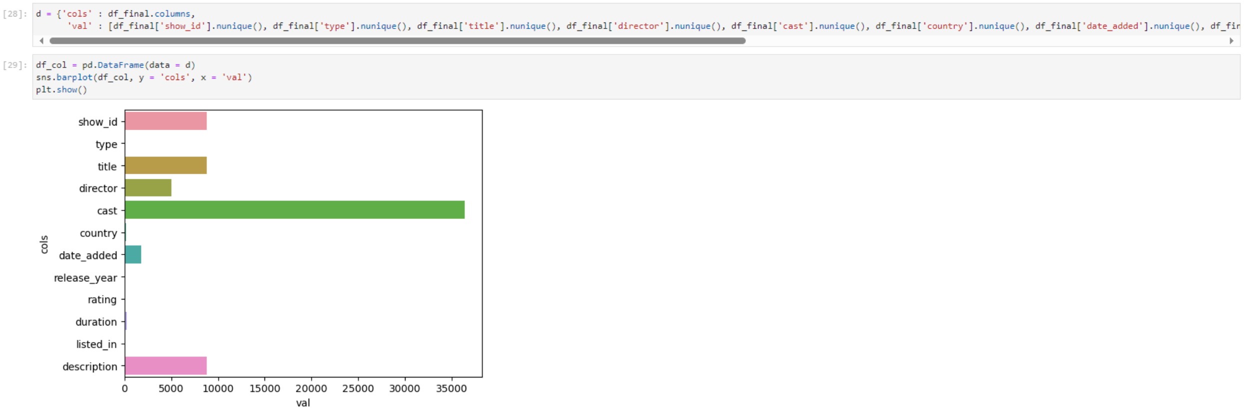
We can see we have 12 columns and 202065 rows.

Below analysis shows the unique count present for each columns:



```
[26]: print('Show id count',df_final['show_id'].nunique())
      print('type count', df_final['type'].nunique())
      print('title count', df_final['title'].nunique())
      print('director count', df_final['director'].nunique())
      print('cast count', df_final['cast'].nunique())
      print('country count', df_final['country'].nunique())
      print('date added count', df_final['date_added'].nunique())
      print('release year count', df_final['release_year'].nunique())
      print('rating count', df_final['rating'].nunique())
      print('duration count', df_final['duration'].nunique())
      print('listed in count', df_final['listed_in'].nunique())
      print('description count', df_final['description'].nunique())
```

```
Show id count 8807
type count 2
title count 8807
director count 4994
cast count 36440
country count 124
date added count 1768
release year count 74
rating count 18
duration count 221
listed in count 42
description count 8775
```



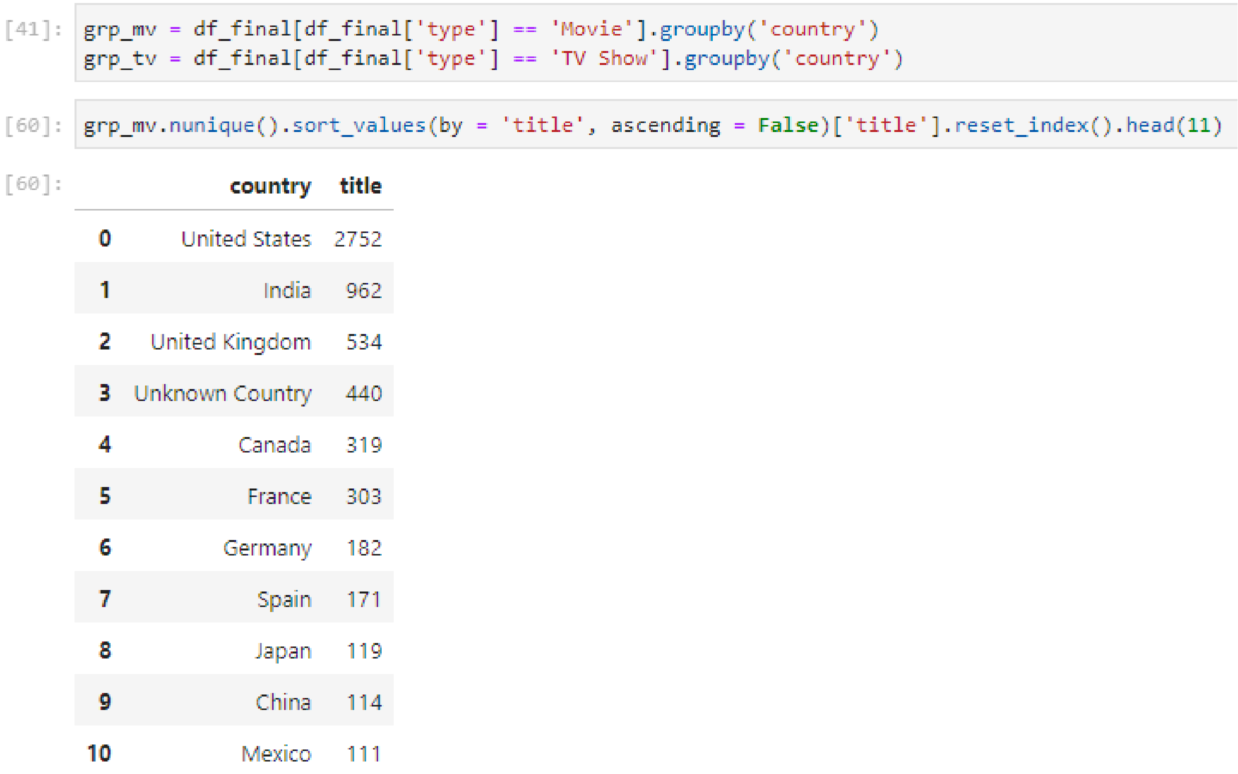
```
[28]: d = {'cols' : df_final.columns,
           'val' : [df_final['show_id'].nunique(), df_final['type'].nunique(), df_final['title'].nunique(), df_final['director'].nunique(), df_final['cast'].nunique(), df_final['country'].nunique(), df_final['date_added'].nunique(), df_fir
```

```
[29]: df_col = pd.DataFrame(data = d)
      sns.barplot(df_col, y = 'cols', x = 'val')
      plt.show()
```

Insights: From the above analysis we can see Netflix has rich amount of movies and tv shows. It have various shows from 1925 to 2018, among 124 countries.

2. Comparison of tv shows vs. movies.

a. Find the number of movies produced in each country and pick the top 10 countries.



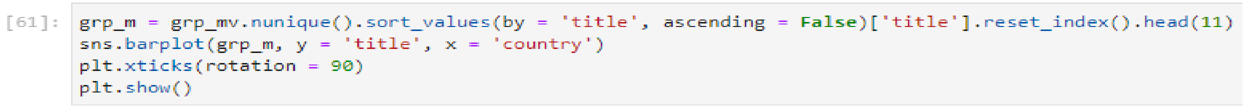
```
[41]: grp_mv = df_final[df_final['type'] == 'Movie'].groupby('country')
      grp_tv = df_final[df_final['type'] == 'TV Show'].groupby('country')
```

```
[60]: grp_mv.nunique().sort_values(by = 'title', ascending = False)['title'].reset_index().head(11)
```

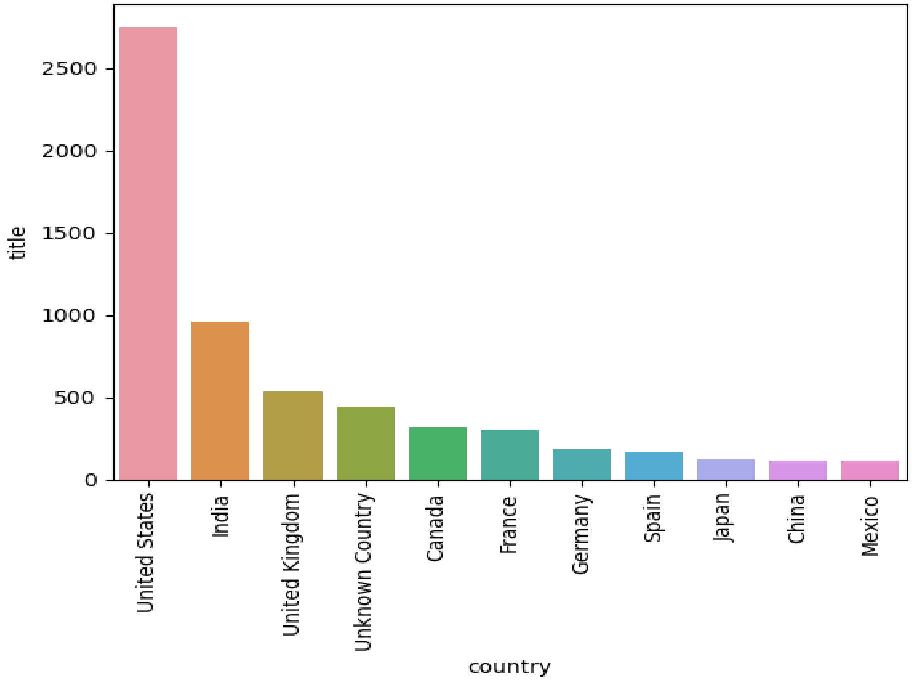
[60]:

	country	title
0	United States	2752
1	India	962
2	United Kingdom	534
3	Unknown Country	440
4	Canada	319
5	France	303
6	Germany	182
7	Spain	171
8	Japan	119
9	China	114
10	Mexico	111

From above we can see top 10 countries if we ignore unknown country which produce movies.

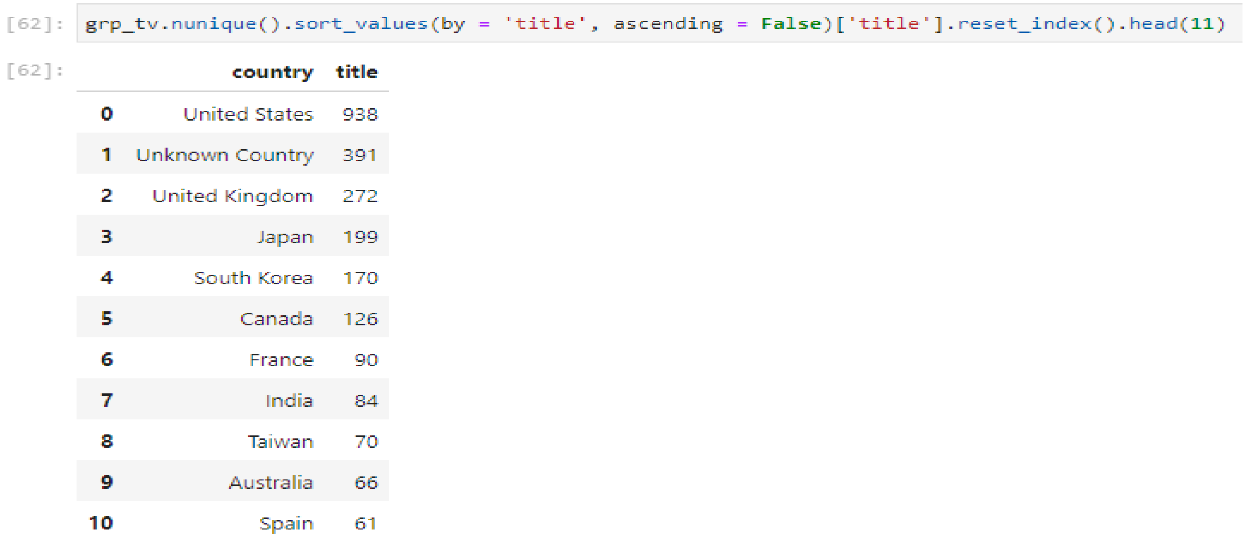


```
[61]: grp_m = grp_mv.nunique().sort_values(by = 'title', ascending = False)['title'].reset_index().head(11)
      sns.barplot(grp_m, y = 'title', x = 'country')
      plt.xticks(rotation = 90)
      plt.show()
```



Insights: From the graph United States produce almost double movie than India which is in 2nd place of top 10. The gap after 2nd position is nominal.

b. Find the number of Tv-Shows produced in each country and pick the top 10 countries.

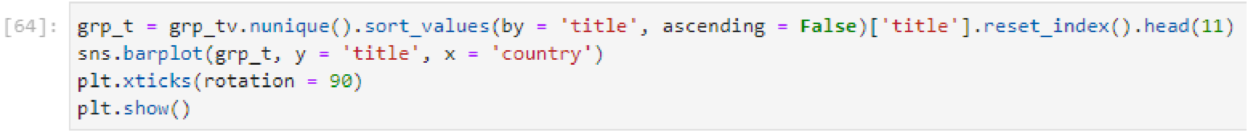


```
[62]: grp_tv.nunique().sort_values(by = 'title', ascending = False)['title'].reset_index().head(11)
```

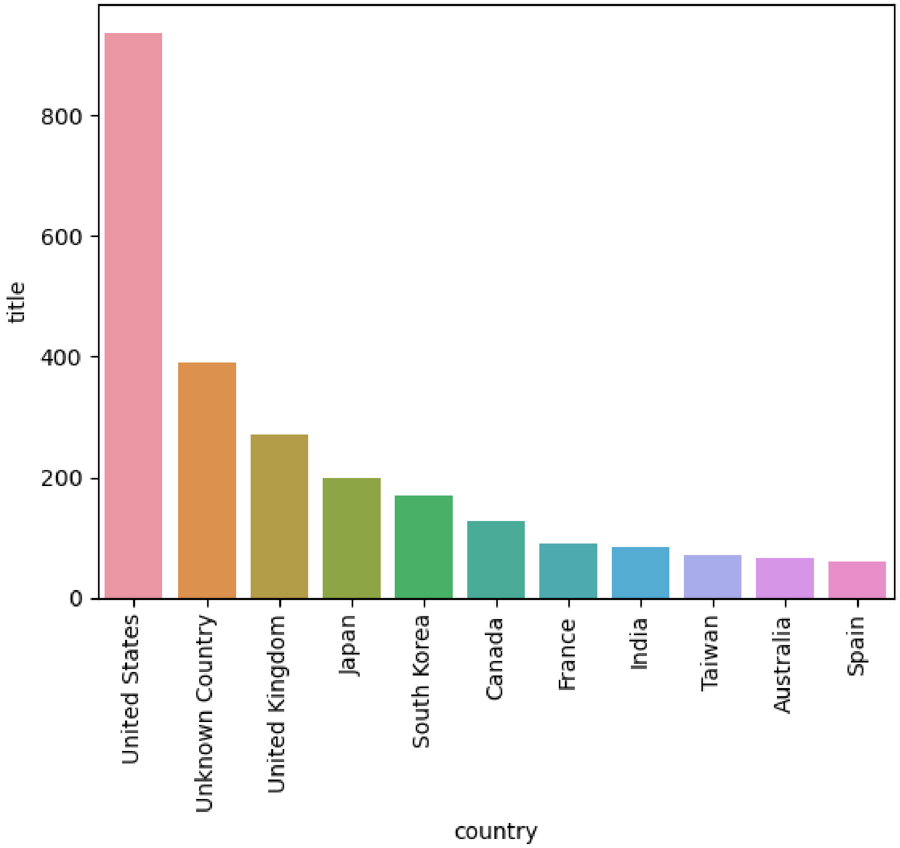
[62]:

	country	title
0	United States	938
1	Unknown Country	391
2	United Kingdom	272
3	Japan	199
4	South Korea	170
5	Canada	126
6	France	90
7	India	84
8	Taiwan	70
9	Australia	66
10	Spain	61

From above we can see top 10 countries if we ignore unknown country which produce Tv Shows.



```
[64]: grp_t = grp_tv.nunique().sort_values(by = 'title', ascending = False)['title'].reset_index().head(11)
      sns.barplot(grp_t, y = 'title', x = 'country')
      plt.xticks(rotation = 90)
      plt.show()
```



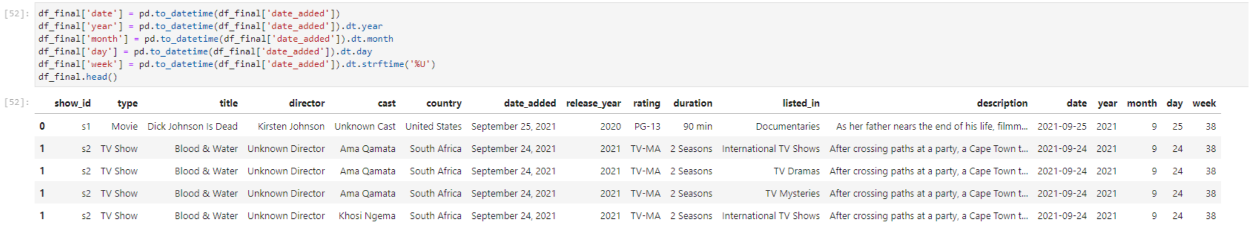
Insights: From the graph United States produce almost thrice TV Shows than. The gap after 1st position is nominal.

Full Analysis Insight:

From both the cases we have seen United States produces most Movies and TV Shows in the world. We can say that United States have great focus on entertainment.

3. What is the best time to launch a TV show and Movies?

Before start this analysis lets create some required columns in the data frame,



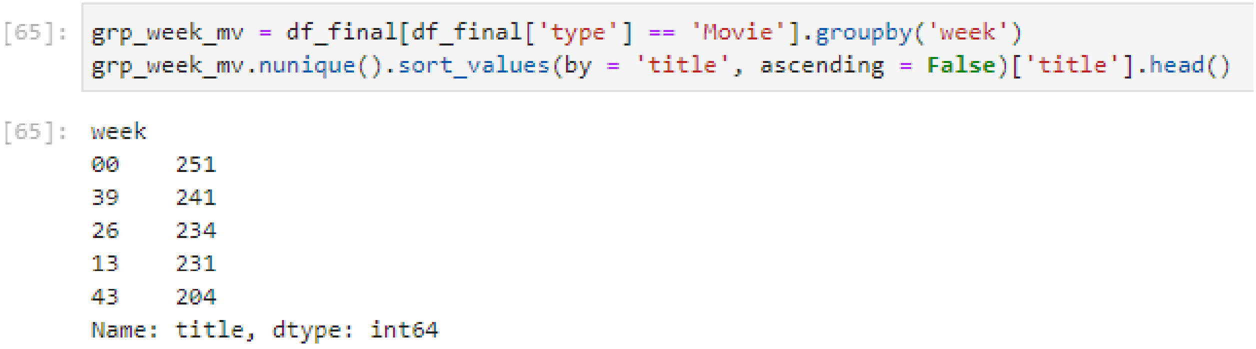
```
[52]: df_final['date'] = pd.to_datetime(df_final['date_added'])
      df_final['year'] = pd.to_datetime(df_final['date_added']).dt.year
      df_final['month'] = pd.to_datetime(df_final['date_added']).dt.month
      df_final['day'] = pd.to_datetime(df_final['date_added']).dt.day
      df_final['week'] = pd.to_datetime(df_final['date_added']).dt.strftime('%U')
      df_final.head()
```

[52]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	date	year	month	day	week
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown Cast	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	2021-09-25	2021	9	25	38
1	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...	2021-09-24	2021	9	24	38
1	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Dramas	After crossing paths at a party, a Cape Town t...	2021-09-24	2021	9	24	38
1	s2	TV Show	Blood & Water	Unknown Director	Ama Qamata	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	TV Mysteries	After crossing paths at a party, a Cape Town t...	2021-09-24	2021	9	24	38
1	s2	TV Show	Blood & Water	Unknown Director	Khosi Ngema	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows	After crossing paths at a party, a Cape Town t...	2021-09-24	2021	9	24	38

a. Find which is the best week to release the TV-show or the movie. Let’s do the analysis separately for TV-shows and Movies.

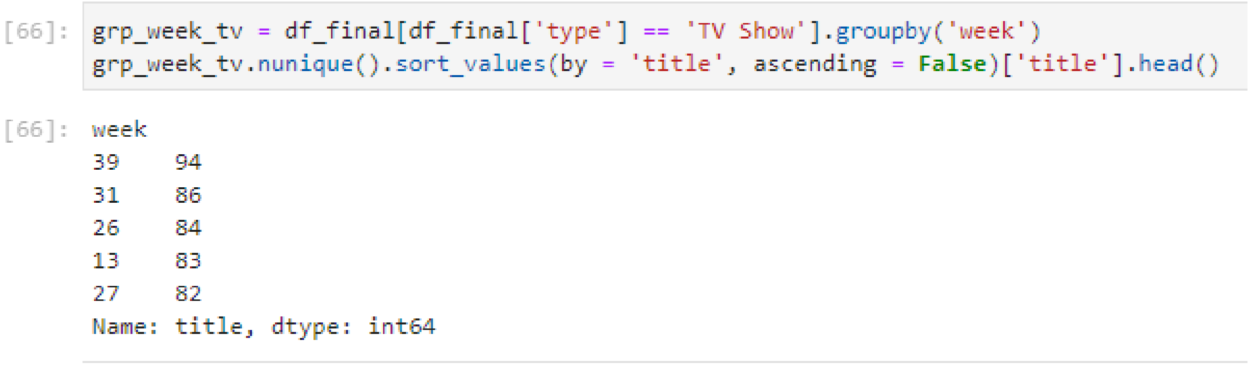
Movies:



```
[65]: grp_week_mv = df_final[df_final['type'] == 'Movie'].groupby('week')
      grp_week_mv.nunique().sort_values(by = 'title', ascending = False)['title'].head()

[65]: week
      00    251
      39    241
      26    234
      13    231
      43    204
      Name: title, dtype: int64
```

First week, 39th and 26th week of every year is best time to release the Movies.

TV Shows:



```
[66]: grp_week_tv = df_final[df_final['type'] == 'TV Show'].groupby('week')
      grp_week_tv.nunique().sort_values(by = 'title', ascending = False)['title'].head()

[66]: week
      39    94
      31    86
      26    84
      13    83
      27    82
      Name: title, dtype: int64
```

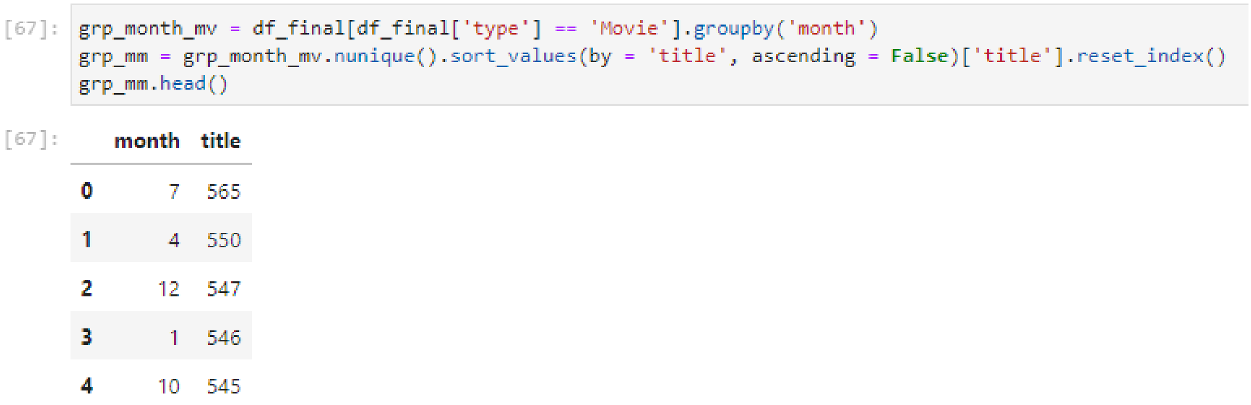
For TV Show we can see 39th, 31st, 26th Week of the year is the best time.

Full Analysis Insight:

From the above analysis we can say that 39th week and 26th week of every year is best for Movie and TV Show release.

b. Find which is the best month to release the TV-show or the movie. Let’s do the analysis separately for TV-shows and Movies.

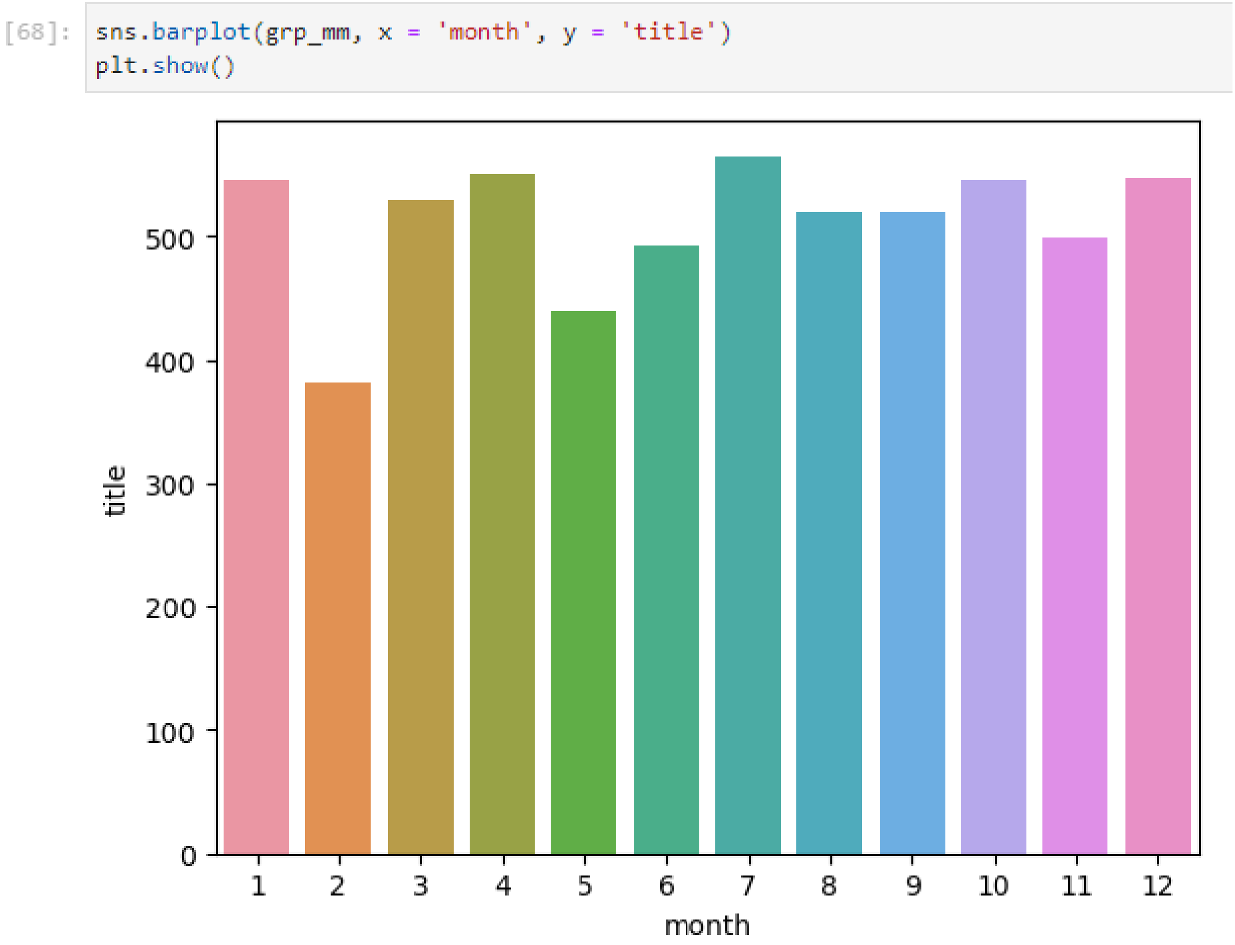
Movies:



```
[67]: grp_month_mv = df_final[df_final['type'] == 'Movie'].groupby('month')
      grp_mm = grp_month_mv.nunique().sort_values(by = 'title', ascending = False)['title'].reset_index()
      grp_mm.head()
```

[67]:

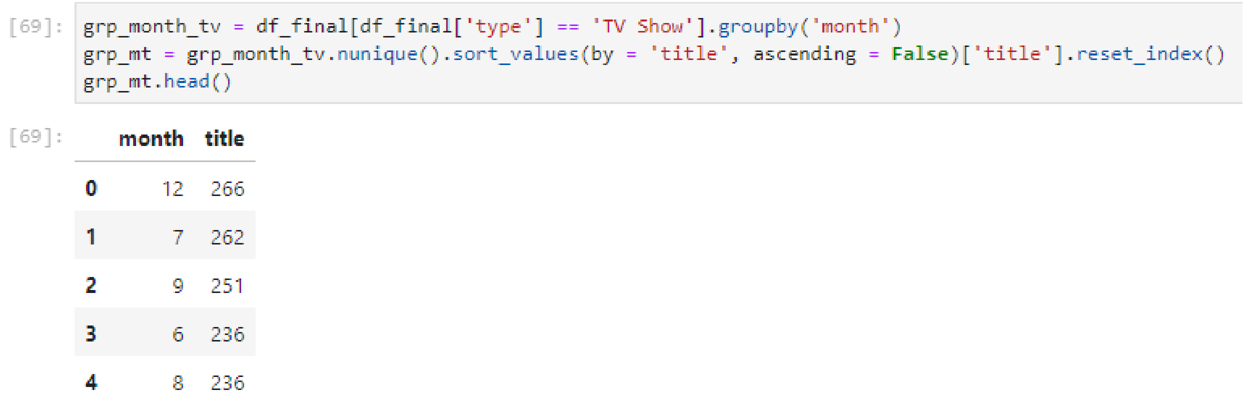
	month	title
0	7	565
1	4	550
2	12	547
3	1	546
4	10	545



```
[68]: sns.barplot(grp_mm, x = 'month', y = 'title')
      plt.show()
```

Here we can see the best month is July for Movies.

TV Show:

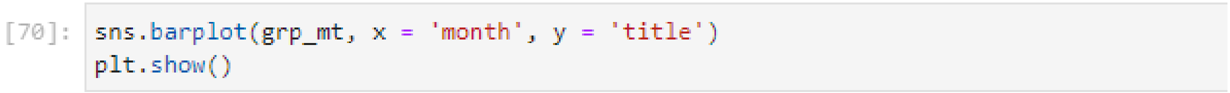


[69]:

```
grp_month_tv = df_final[df_final['type'] == 'TV Show'].groupby('month')
grp_mt = grp_month_tv.nunique().sort_values(by = 'title', ascending = False)['title'].reset_index()
grp_mt.head()
```

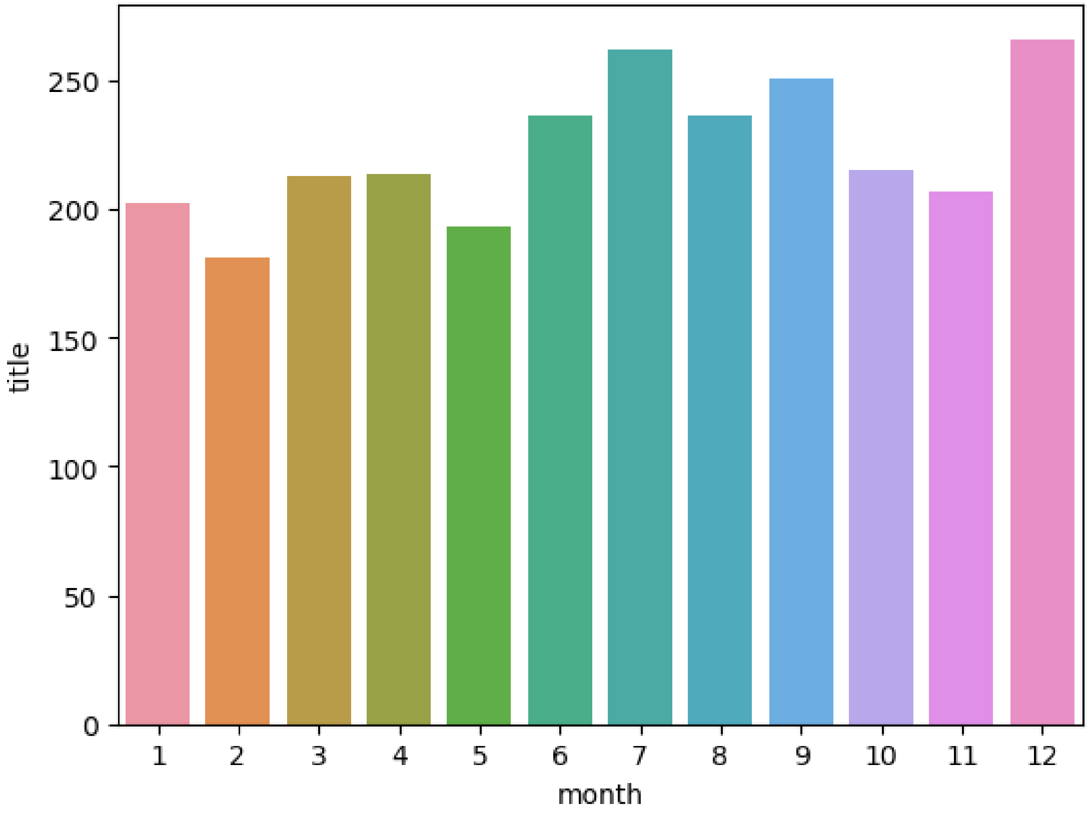
[69]:

	month	title
0	12	266
1	7	262
2	9	251
3	6	236
4	8	236



[70]:

```
sns.barplot(grp_mt, x = 'month', y = 'title')
plt.show()
```



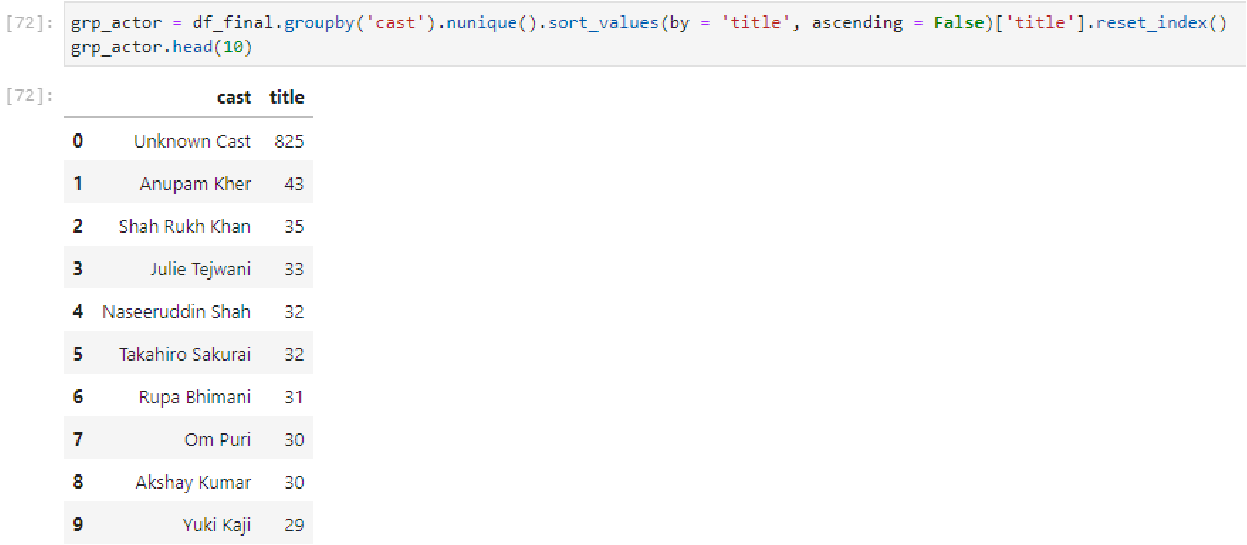
December, July are the best months for TV Shows.

Full Analysis Insight:

From the month and week analysis we can see that data are matching for both the week and month. Jan, July are the best time for both Movies and TV shows.

4. Analysis of actors/directors of different types of shows/movies.

a. Identify the top 10 actors who have appeared in most movies or TV shows.



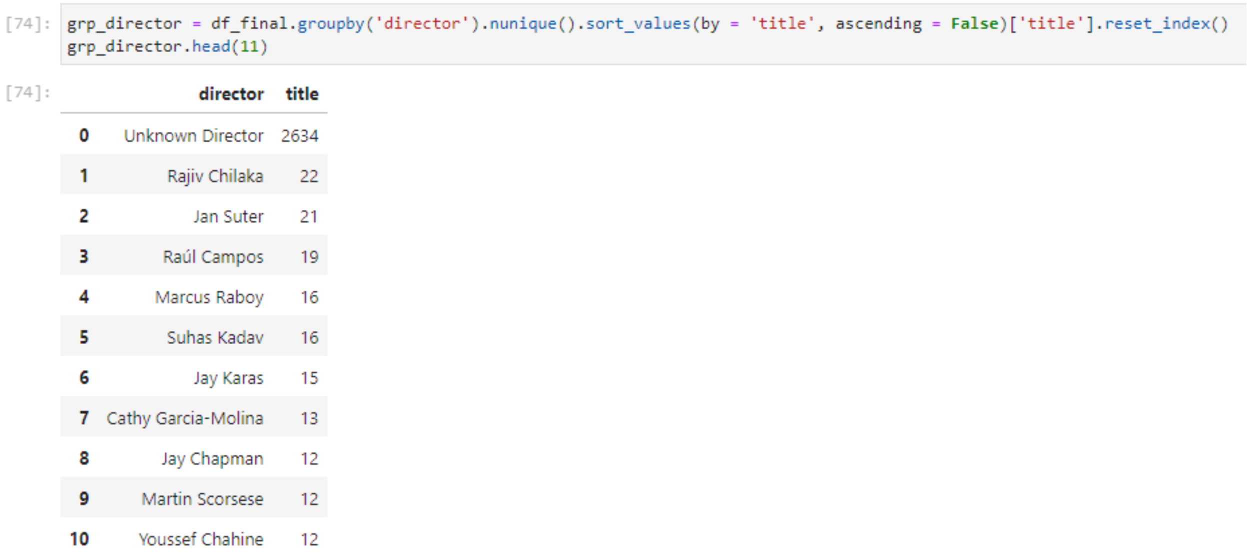
```
[72]: grp_actor = df_final.groupby('cast').nunique().sort_values(by = 'title', ascending = False)['title'].reset_index()
      grp_actor.head(10)
```

[72]:

	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

Here the above table represent the top 10 actors who are worked on most TV shows and Movies. Though we have 825 unknown actors are there as it is not provided in the data file. I have tried with Mode as it is a categorical column but it leads to wrong value.

b. Identify the top 10 directors who have appeared in most movies or TV shows.



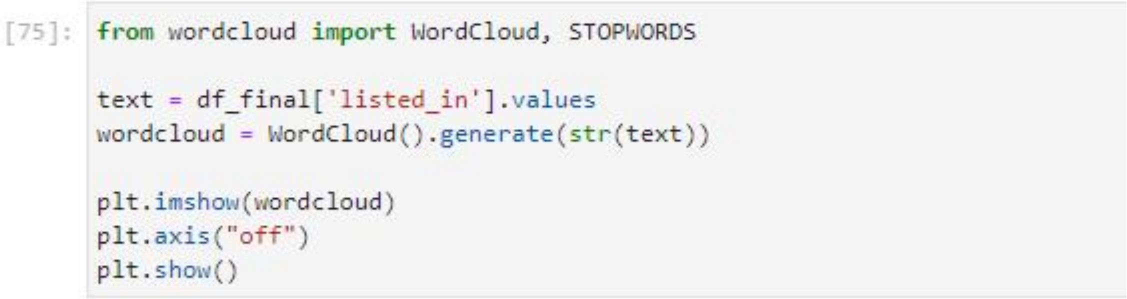
```
[74]: grp_director = df_final.groupby('director').nunique().sort_values(by = 'title', ascending = False)['title'].reset_index()
      grp_director.head(11)
```

[74]:

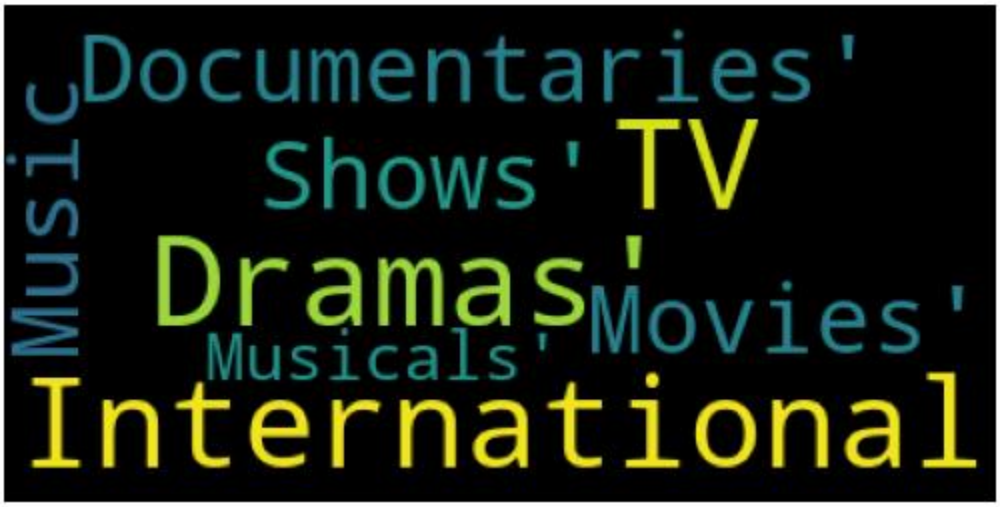
	director	title
0	Unknown Director	2634
1	Rajiv Chilaka	22
2	Jan Suter	21
3	Raúl Campos	19
4	Marcus Raboy	16
5	Suhas Kadav	16
6	Jay Karas	15
7	Cathy Garcia-Molina	13
8	Jay Chapman	12
9	Martin Scorsese	12
10	Youssef Chahine	12

Here the above table represent the top 10 actors who are worked on most TV shows and Movies. Though we have 2634 unknown directors are there as it is not provided in the data file. I have tried with Mode as it is a categorical column but it leads to wrong value.

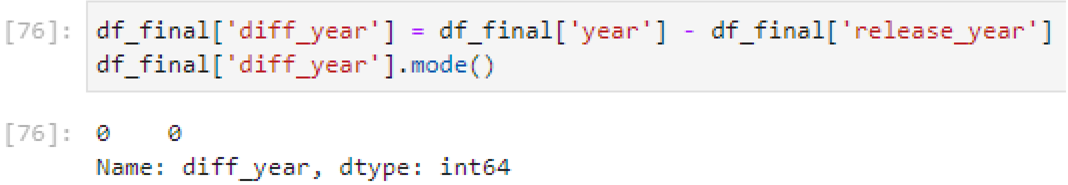
5. Which genre movies are more popular or produced more



```
[75]: from wordcloud import WordCloud, STOPWORDS

      text = df_final['listed_in'].values
      wordcloud = WordCloud().generate(str(text))

      plt.imshow(wordcloud)
      plt.axis("off")
      plt.show()
```



6. Find After how many days the movie will be added to Netflix after the release of the movie



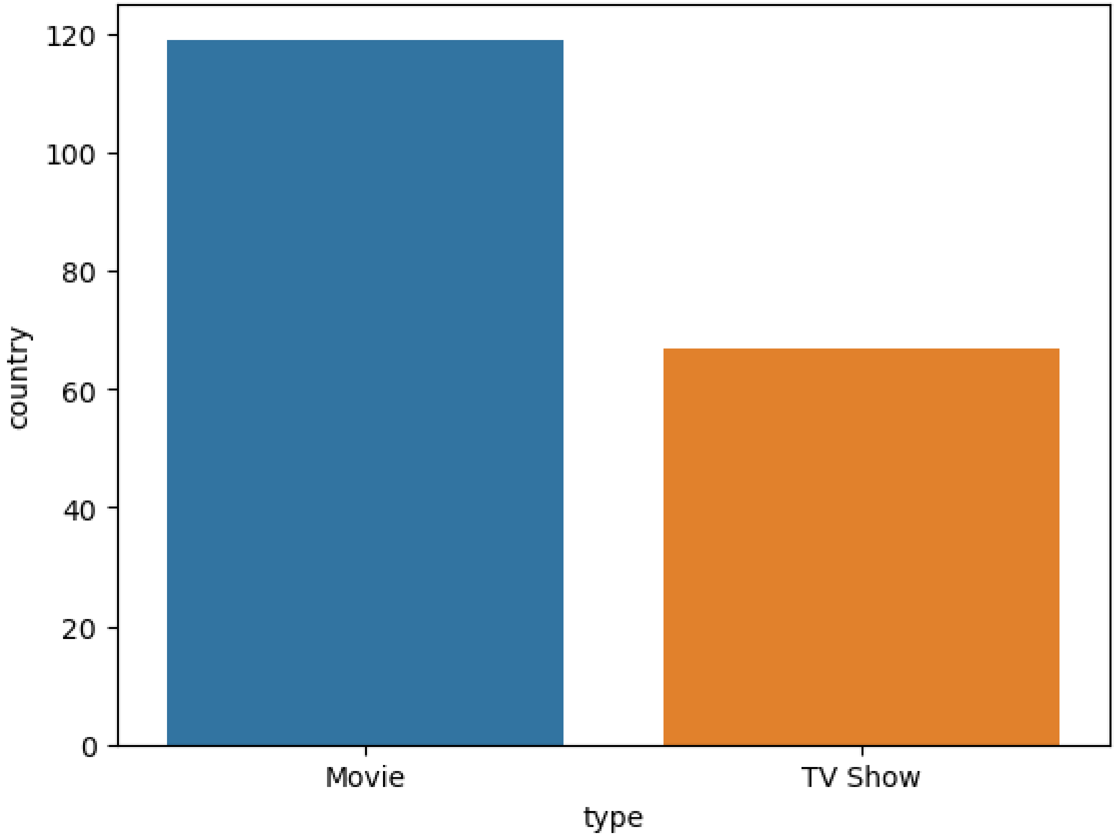
```
[76]: df_final['diff_year'] = df_final['year'] - df_final['release_year']
      df_final['diff_year'].mode()

[76]: 0    0
      Name: diff_year, dtype: int64
```

From the past data we can observed that maximum movies are added in same year of release in the Netflix.

7. Type of contents in the countries:

From the below graph we can say that most of the countries preferred Movies over TV shows.

```
[77]: grp_t = df_final.groupby('type')
      grp_tp = grp_t['country'].nunique().reset_index()
      sns.barplot(grp_tp, x = 'type', y = 'country', )
      plt.show()
```

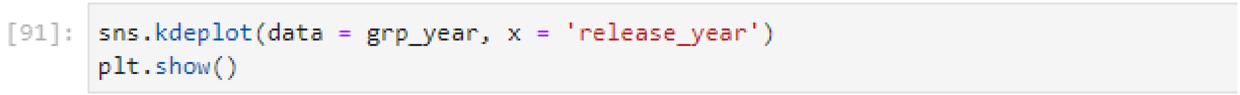


8. How has the number of movies released per year changed over the last 20-30 years?

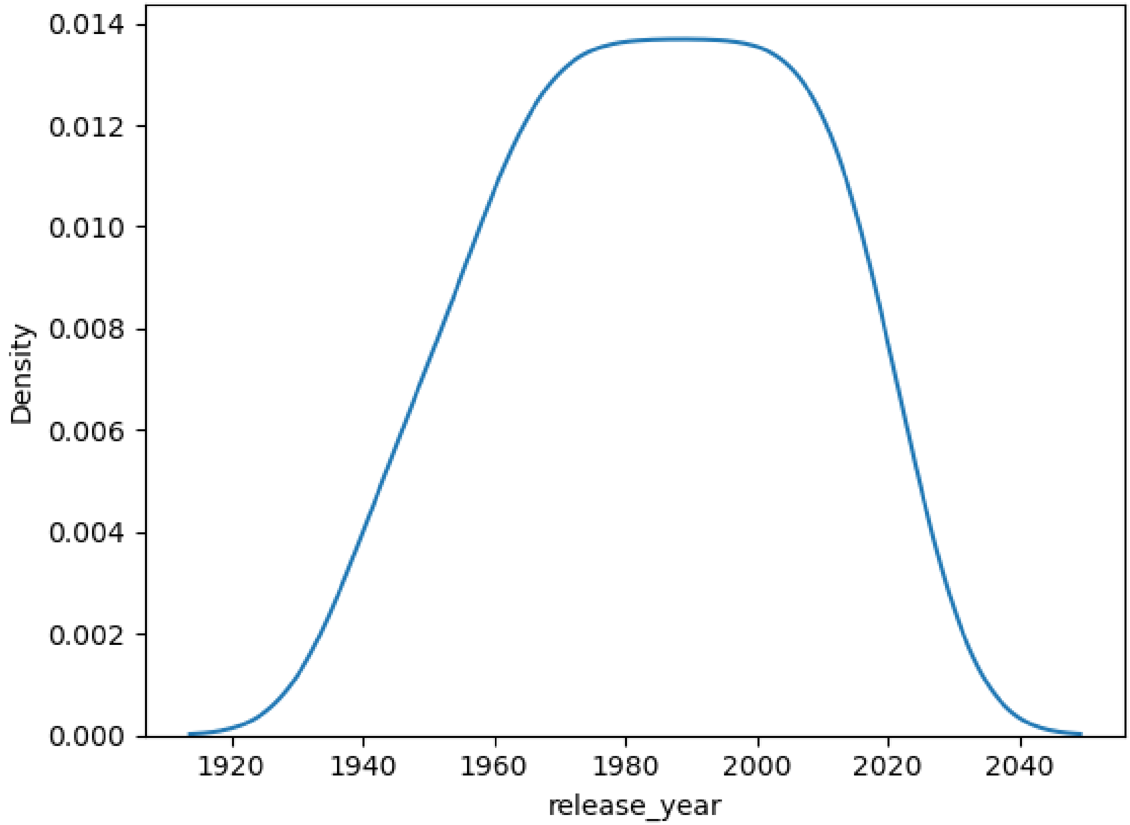
```
[90]: grp_y = df_final[df_final['type'] == 'Movie'].groupby('release_year')
      grp_year = grp_y['title'].nunique().sort_index(ascending = False).reset_index()
      grp_year.head()
```

[90]:

	release_year	title
0	2021	277
1	2020	517
2	2019	633
3	2018	767
4	2017	767



```
[91]: sns.kdeplot(data = grp_year, x = 'release_year')
      plt.show()
```



So the above KDE Plot is slightly right skewed, which depict that last 20 to 30 years had more movie release.

Final Insights:

1. Netflix has more movies than TV shows.

2. Netflix added movies and TV shows mostly within a year of their release date.

3. Netflix increases their content over last 20 to 30 years.

4. July month of every year is best time to release Movies and TV shows over the platform.

5. Maximum content is from United States.

Recommendations:

1. Netflix should add more TV shows.

2. Netflix should add content from more countries not only maximum from United States, other countries also have rich amount of contents, which lead Netflix to achieve more users.

3. Netflix should release more content on festival months depends on the countries.