

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

```
In [2]: 1 csv_file_path = "C:/Users/ankin/Downloads/netflix.csv"
```

```
In [4]: 1 df = pd.read_csv(csv_file_path)
```

```
In [5]: 1 df  
2
```

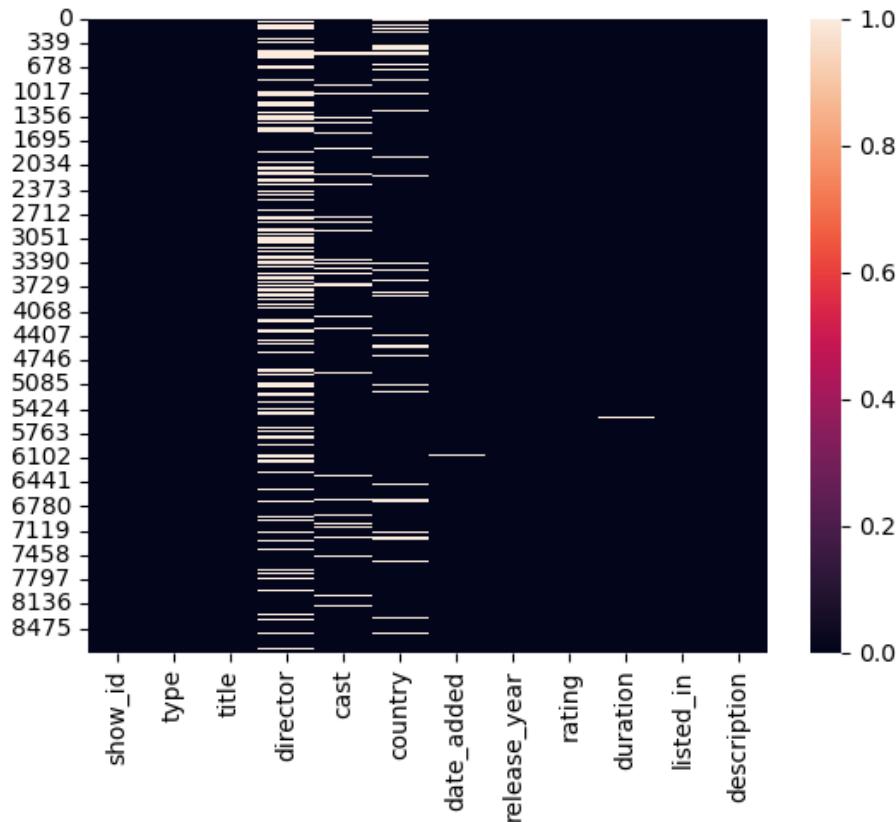
Out[5]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	list
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Nan	United States	September 25, 2021	2020	PG-13	90 min	Documentary
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 Show Drama Mystery
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel	Nan	September 24, 2021	2021	TV-MA	1 Season	Critical TV Show

- Let's see the heatmap to find any missing values.

```
In [7]: 1 sns.heatmap(df.isna())
```

```
Out[7]: <Axes: >
```

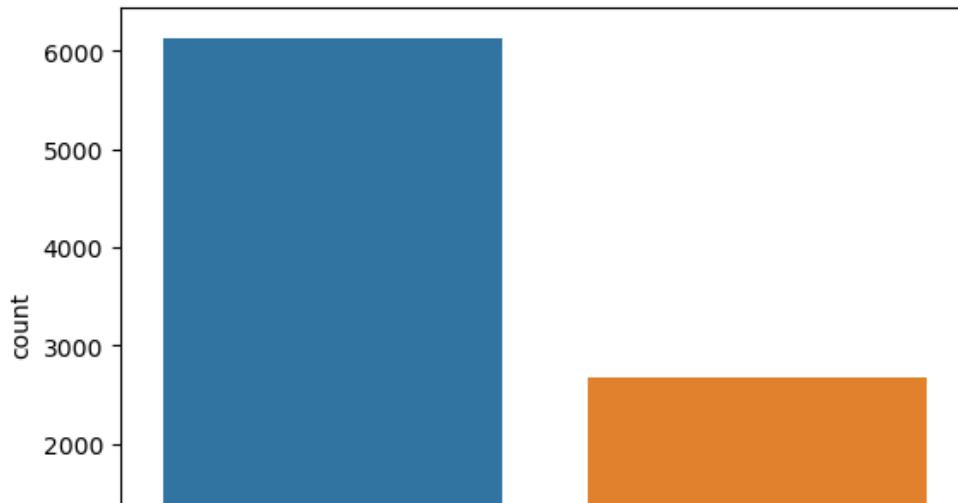


Well a lot of values are missing mainly in director, cast, country. Very few missing in date_added and duration.

-
- Let's now see the count of TV Shows vs Films

```
In [8]: 1 sns.countplot(x= 'type', data=df)
2
```

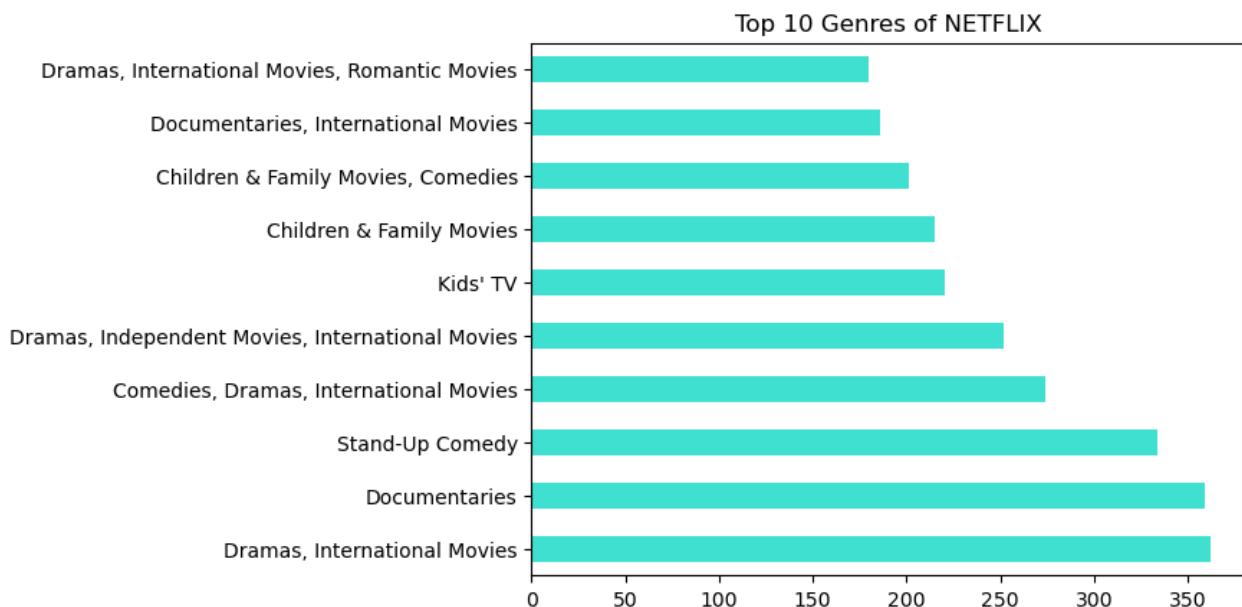
Out[8]: <Axes: xlabel='type', ylabel='count'>



Movies are a lot more in number as compared to TV Show

- Let's have a look at the Top 10 genres popular on netflix from the column listed_in

```
In [12]: 1 df["listed_in"].value_counts()[:10].plot(kind="barh", color="turquoise")
2 plt.title("Top 10 Genres of NETFLIX");
```



- Let's take copy of our dataframe df to do some cleaning with content type as Movie

```
In [14]: 1 df1 = df.copy()
2 df1 = df1[df1['type'] == 'Movie']
3 df1
```

						Marsden, ...					
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125 min	
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	September 24, 2021	2021	PG-13	104 min	
12	s13	Movie	Je Suis Karl	Christian Schwochow	Luna Wedler, Jannis Niewöhner, Milan Peschel	Germany, Czech Republic	September 23, 2021	2021	TV-MA	127 min	

- We are dropping director and cast as there are many missing values which we found in the heatmap we created earlier but not dropping country as that is an important column for us.

```
In [15]: 1 df1.drop(['director', 'cast'], axis=1, inplace=True)
2 df1.head()
```

Out[15]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
6	s7	Movie	My Little Pony: A New Generation	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
7	s8	Movie	Sankofa	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...

- Let us now filter and retrieve the rows which have country as null.

```
In [16]: 1 df1[df1['country'].isnull()]
2
```

Out[16]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
6	s7	Movie	My Little Pony: A New Generation	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
13	s14	Movie	Confessions of an Invisible Girl	NaN	September 22, 2021	2021	TV-PG	91 min	Children & Family Movies, Comedies	When the clever but socially-awkward Tetê join...
16	s17	Movie	Europe's Most Dangerous Man: Otto Skorzeny in ...	NaN	September 22, 2021	2020	TV-MA	67 min	Documentaries, International Movies	Declassified documents reveal the post-WWII li...
18	s19	Movie	Intrusion	NaN	September 22, 2021	2021	TV-14	94 min	Thrillers	After a deadly home invasion at a couple's new...
22	s23	Movie	Avvai Shanmughi	NaN	September 21, 2021	1996	TV-PG	161 min	Comedies, International Movies	Newly divorced and denied visitation rights wi...
...
8585	s8586	Movie	Three-Quarters Decent	NaN	June 20, 2019	2010	TV-14	96 min	Comedies, Dramas, International Movies	Determined to fight corruption in his country,...
8602	s8603	Movie	Tom and Jerry: The Magic Ring	NaN	December 15, 2019	2001	TV-Y7	60 min	Children & Family Movies, Comedies	When a young wizard leaves Tom to guard his pr...
8622	s8623	Movie	Tremors 2: Aftershocks	NaN	January 1, 2020	1995	PG-13	100 min	Comedies, Horror Movies, Sci-Fi & Fantasy	A rag-tag team of survivalists and scientists ...
8718	s8719	Movie	Westside vs. the World	NaN	August 9, 2019	2019	TV-MA	96 min	Documentaries, Sports Movies	A look into the journey of influential strengt...
8759	s8760	Movie	World's Weirdest Homes	NaN	February 1, 2019	2015	TV-PG	49 min	Movies	From a bubble-shaped palace to an island built...

440 rows × 10 columns

- I am using the forward fill method to fill the missing values assuming that the data frame was created in a sequential manner.

```
In [19]: 1 df1['country'] = df1['country'].ffill(axis=0)
```

```
In [20]: 1 df1[df1['country'].isnull()]
```

Out[20]:

show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
---------	------	-------	---------	------------	--------------	--------	----------	-----------	-------------

- Let's see if there are null values in rating

```
In [21]: 1 df1[df1['rating'].isnull()]
```

Out[21]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
5989	s5990	Movie	13TH: A Conversation with Oprah Winfrey & Ava ...	United States	January 26, 2017	2017	NaN	37 min	Movies	Oprah Winfrey sits down with director Ava DuVe...
7537	s7538	Movie	My Honor Was Loyalty	Italy	March 1, 2017	2015	NaN	115 min	Dramas	Amid the chaos and horror of World War II, a c...

- Since there are only two null values in rating, let's update them manually using iteration with zip function.

```
In [26]: 1 ratings = ['TV-PG', 'TV-MA']
2
3 for id, rating in zip(df1[df1['rating'].isnull()].index, ratings):
4     df1['rating'].loc[id] = rating
```

C:\Users\ankin\AppData\Local\Temp\ipykernel_12524\1409508373.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
df1['rating'].loc[id] = rating

```
In [27]: 1 df1[df1['rating'].isnull()]
```

Out[27]:

show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
---------	------	-------	---------	------------	--------------	--------	----------	-----------	-------------

- The date_added column may also have some null values but we can drop those rows as it will not affect our analysis. Let's update the same dataframe without the rows with null values

```
In [28]: 1 df1 = df1[df1['date_added'].notna()]
2 df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6131 entries, 0 to 8806
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   show_id     6131 non-null   object  
 1   type        6131 non-null   object  
 2   title       6131 non-null   object  
 3   country     6131 non-null   object  
 4   date_added  6131 non-null   object  
 5   release_year 6131 non-null   int64  
 6   rating      6131 non-null   object  
 7   duration    6128 non-null   object  
 8   listed_in   6131 non-null   object  
 9   description  6131 non-null   object  
dtypes: int64(1), object(9)
memory usage: 526.9+ KB
```

- Let us now separate the multiple countries listed in one cell by using function to split based on delimiter comma

```
In [30]: 1 df1['main_country'] = df1['country'].apply(lambda x: x.split(',')[0])
2 df1
```

Out[30]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmmaker...
6	s7	Movie	My Little Pony: A New Generation	United States	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...
7	s8	Movie	Sankofa	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...

- Let's start with our analysis now.

We will fetch the top 10 countries and group them based on their genres and ratings. This will give us an idea that what kind of content is popular in which country based on genres and we can then try to get similar kind of content in those countries in the future.

```
In [31]: 1 top_countries = df1.groupby('main_country').count().sort_values('type', ascending=False)[:10]
          2 top_countries
```

Out[31]:

main_country	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
United States	2541	2541	2541	2541	2541	2541	2541	2538	2541	2541
India	998	998	998	998	998	998	998	998	998	998
United Kingdom	410	410	410	410	410	410	410	410	410	410
Canada	196	196	196	196	196	196	196	196	196	196
France	159	159	159	159	159	159	159	159	159	159
Spain	133	133	133	133	133	133	133	133	133	133
Nigeria	111	111	111	111	111	111	111	111	111	111
Egypt	106	106	106	106	106	106	106	106	106	106
Mexico	95	95	95	95	95	95	95	95	95	95
Japan	94	94	94	94	94	94	94	94	94	94

Same way, let's fetch top 10 genres with maximum content produced.

```
In [32]: 1 top_genres = df1.groupby('listed_in').count().sort_values('type', ascending = False)[:10]
2 top_genres
```

Out[32]:

listed_in	show_id	type	title	country	date_added	release_year	rating	duration	description	main_country
Dramas, International Movies	362	362	362	362	362	362	362	362	362	362
Documentaries	359	359	359	359	359	359	359	359	359	359
Stand-Up Comedy	334	334	334	334	334	334	334	334	334	334
Comedies, Dramas, International Movies	274	274	274	274	274	274	274	274	274	274
Dramas, Independent Movies, International Movies	252	252	252	252	252	252	252	252	252	252
Children & Family Movies	215	215	215	215	215	215	215	215	215	215
Children & Family Movies, Comedies	201	201	201	201	201	201	201	201	201	201
Documentaries, International Movies	186	186	186	186	186	186	186	186	186	186
Dramas, International Movies, Romantic Movies	180	180	180	180	180	180	180	180	180	180
Comedies, International Movies	176	176	176	176	176	176	176	176	176	176

Let's now filter the data for top 10 countries and genres

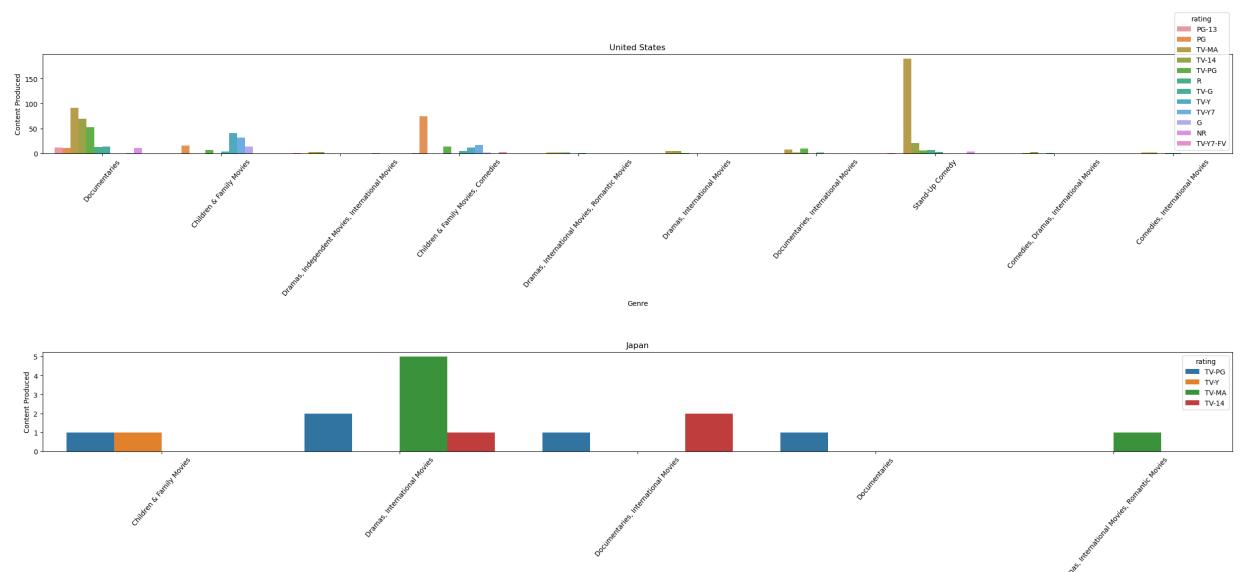
```
In [38]: 1 df1 = df1[df1['main_country'].isin(list(top_countries.index))]
2 df1 = df1[df1['listed_in'].isin(list(top_genres.index))]
```

Let's get a list now for ratings with all the unique ratings, countries with unique countries from main_country and listing from unique listing.

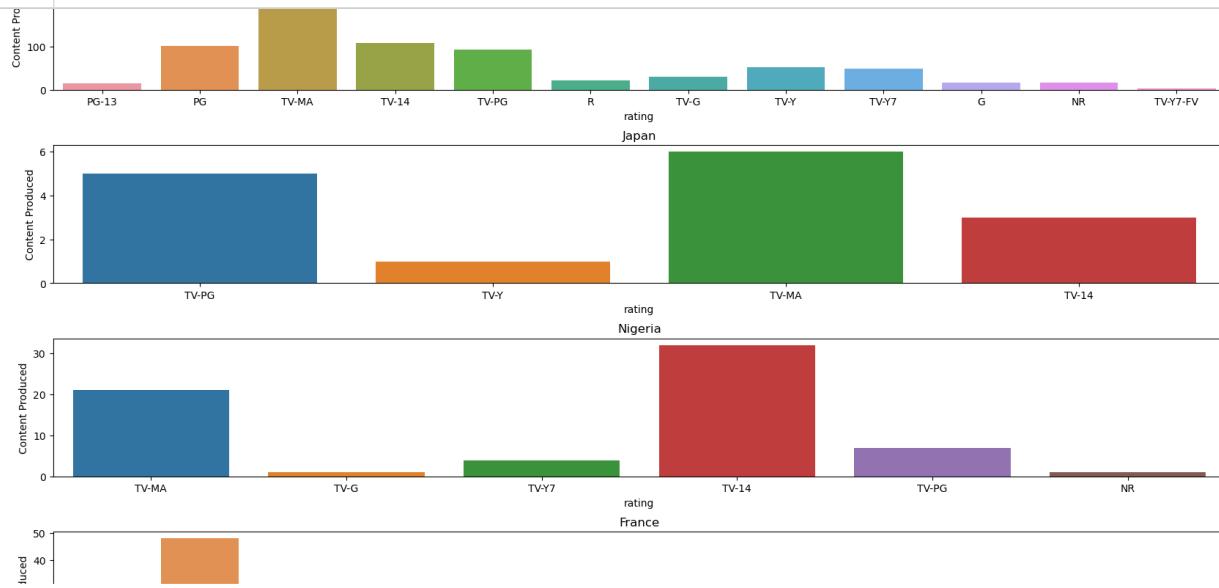
```
In [39]: 1 ratings=[]
2 for rate in df1['rating'].unique():
3     ratings.append(rate)
4
5 countries = df1['main_country'].unique()
6
7 listing = df1['listed_in'].unique()
8
9 ratings, countries, listing
```

```
Out[39]: ([ 'PG-13',
  'PG',
  'TV-MA',
  'TV-14',
  'TV-PG',
  'TV-Y',
  'TV-G',
  'TV-Y7',
  'R',
  'G',
  'NC-17',
  'NR',
  'TV-Y7-FV',
  'UR'],
array(['United States', 'Japan', 'Nigeria', 'France', 'United Kingdom',
       'India', 'Mexico', 'Egypt', 'Canada', 'Spain'], dtype=object),
array(['Documentaries', 'Children & Family Movies',
       'Dramas, Independent Movies, International Movies',
       'Dramas, International Movies',
       'Children & Family Movies',
       'Dramas, Independent Movies, International Movies',
       'Dramas, International Movies'],
       dtype=object))
```

```
In [47]: 1 fig = plt.figure()
2     figsize=(30,40)
3 )
4
5 for i, name in enumerate(countries):
6     portion = df1[df1['main_country'] == str(name)]
7     ax = fig.add_subplot(len(countries),1,i+1)
8     header = name
9     sns.countplot(x='listed_in', data=portion[portion['listed_in'].isin(listing)], hue='rating',
10     ax.set_title(header)
11     plt.subplots_adjust(left=0.1,
12                         bottom=0.1,
13                         right=0.9,
14                         top=1.5,
15                         wspace=0.5,
16                         hspace=2.0)
17     plt.xlabel('Genre')
18     plt.xticks(rotation = 50)
19     ax.set(ylabel='Content Produced')
```



```
In [48]: 1 fig = plt.figure(figsize=(20,32))
2
3 for i, name in enumerate(countries):
4     portion = df1[df1['main_country'] == str(name)]
5     ax = fig.add_subplot(len(countries),1,i+1)
6     header = name
7     sns.countplot(x='rating', data=portion[portion['rating'].isin(ratings)])
8     ax.set_title(header)
9     plt.subplots_adjust(left=0.1,
10                         bottom=0.1,
11                         right=0.9,
12                         top=0.9,
13                         wspace=0.4,
14                         hspace=0.4)
15     ax.set(ylabel='Content Produced')
```



```
1 Let's dive in TV Shows now
```

```
In [53]: 1 df2 = df.copy()
2 df2 = df2[df2['type'] == 'TV Show']
3 df2.head()
```

Out[53]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
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
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoa, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...
3	s4	TV	Jailbirds New Orleans	NaN	NaN	NaN	September	2021	TV-	1	Docuseries, ...

```
In [54]: 1 df2.drop(['director', 'cast'], axis=1, inplace=True)
2 df2.head()
```

Out[54]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
1	s2	TV Show	Blood & Water	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	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 lord...
3	s4	TV Show	Jailbirds New Orleans	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	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...
5	s6	TV Show	Midnight Mass	NaN	September 24, 2021	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brin...

```
In [55]: 1 df2[df2['country'].isnull()]  
2
```

Out[55]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
2	s3	TV Show	Ganglands	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 lord...
3	s4	TV Show	Jailbirds New Orleans	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
5	s6	TV Show	Midnight Mass	NaN	September 24, 2021	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brin...
10	s11	TV Show	Vendetta: Truth, Lies and The Mafia	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, Docuseries, International TV S...	Sicily boasts a bold "Anti-Mafia" coalition. B...

```
In [56]: 1 df2['country'] = df2['country'].ffill(axis=0)  
2 df2[df2['country'].isnull()]
```

Out[56]:

show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
---------	------	-------	---------	------------	--------------	--------	----------	-----------	-------------

```
In [57]: 1 df2[df2['rating'].isnull()]
```

Out[57]:

show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
---------	------	-------	---------	------------	--------------	--------	----------	-----------	-------------

```
In [58]: 1 df2[df2['rating'].isna()]
```

Out[58]:

show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
---------	------	-------	---------	------------	--------------	--------	----------	-----------	-------------

```
In [59]: 1 df2 = df2[df2['date_added'].notna()]  
2 df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 2666 entries, 1 to 8803  
Data columns (total 10 columns):  
 #   Column      Non-Null Count  Dtype     
 ---  --          --          --  
 0   show_id     2666 non-null    object    
 1   type        2666 non-null    object    
 2   title       2666 non-null    object    
 3   country     2666 non-null    object    
 4   date_added  2666 non-null    object    
 5   release_year 2666 non-null    int64    
 6   rating      2666 non-null    object    
 7   duration    2666 non-null    object    
 8   listed_in   2666 non-null    object    
 9   description  2666 non-null    object    
dtypes: int64(1), object(9)  
memory usage: 229.1+ KB
```

```
In [60]: 1 df2['main_country'] = df2['country'].apply(lambda x: x.split(',')[0])
2 df2
```

Out[60]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description	ma
1	s2	TV Show	Blood & Water	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	South Africa	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	South Africa	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...	\$

```
In [61]: 1 top_countries = df2.groupby('main_country').count().sort_values('type', ascending=False)[:10]
2 top_countries
```

Out[61]:

main_country	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
United States	990	990	990	990	990	990	990	990	990	990
United Kingdom	281	281	281	281	281	281	281	281	281	281
South Korea	194	194	194	194	194	194	194	194	194	194
Japan	194	194	194	194	194	194	194	194	194	194
India	101	101	101	101	101	101	101	101	101	101
Canada	99	99	99	99	99	99	99	99	99	99
Taiwan	80	80	80	80	80	80	80	80	80	80
France	79	79	79	79	79	79	79	79	79	79
Australia	65	65	65	65	65	65	65	65	65	65
Spain	60	60	60	60	60	60	60	60	60	60

```
In [62]: 1 top_genres = df2.groupby('listed_in').count().sort_values('type', ascending=False)[:10]
2 top_genres
```

Out[62]:

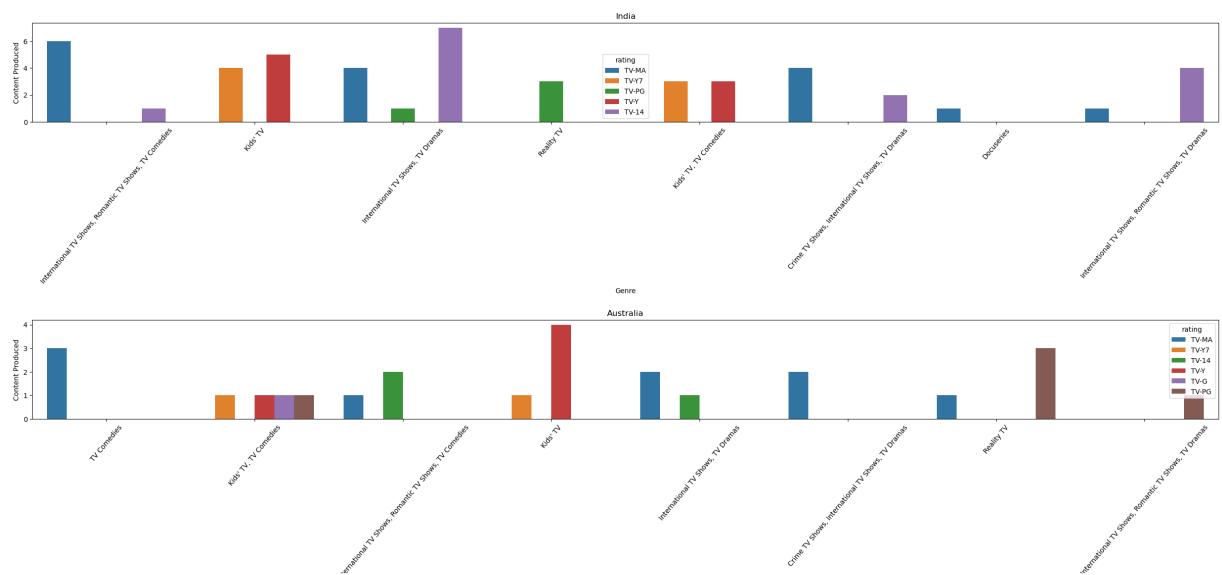
listed_in	show_id	type	title	country	date_added	release_year	rating	duration	description	main_country
Kids' TV	219	219	219	219	219	219	219	219	219	219
International TV Shows, TV Dramas	121	121	121	121	121	121	121	121	121	121
Crime TV Shows, International TV Shows, TV Dramas	110	110	110	110	110	110	110	110	110	110
Kids' TV, TV Comedies	98	98	98	98	98	98	98	98	98	98
Reality TV	95	95	95	95	95	95	95	95	95	95
International TV Shows, Romantic TV Shows, TV Comedies	94	94	94	94	94	94	94	94	94	94
International TV Shows, Romantic TV Shows, TV Dramas	90	90	90	90	90	90	90	90	90	90
Anime Series, International TV Shows	88	88	88	88	88	88	88	88	88	88
Docuseries	84	84	84	84	84	84	84	84	84	84
TV Comedies	68	68	68	68	68	68	68	68	68	68

```
In [63]: 1 df2 = df2[df2['main_country'].isin(list(top_countries.index))]
2 df2 = df2[df2['listed_in'].isin(list(top_genres.index))]
3
```

```
In [64]: 1 ratings=[]
2 for rate in df2['rating'].unique():
3     ratings.append(rate)
4
5 countries = df2['main_country'].unique()
6
7 listing = df2['listed_in'].unique()
8
9 ratings, countries, listing
```

```
Out[64]: ([['TV-MA', 'TV-Y7', 'TV-PG', 'TV-Y', 'TV-14', 'TV-G'],
array(['India', 'Australia', 'United Kingdom', 'United States', 'Japan',
       'France', 'South Korea', 'Taiwan', 'Canada', 'Spain'], dtype=object),
array(['International TV Shows, Romantic TV Shows, TV Comedies',
       'TV Comedies', "Kids' TV", 'Reality TV', "Kids' TV, TV Comedies",
       'International TV Shows, TV Dramas',
       'Anime Series, International TV Shows', 'Docuseries',
       'Crime TV Shows, International TV Shows, TV Dramas',
       'International TV Shows, Romantic TV Shows, TV Dramas'], dtype=object))
```

```
In [65]: 1 fig = plt.figure()
2     figsize=(30,40)
3 )
4
5 for i, name in enumerate(countries):
6     frame = df2[df2['main_country'] == str(name)]
7     ax = fig.add_subplot(len(countries),1,i+1)
8     topic = name
9     sns.countplot(x='listed_in', data= frame[frame['listed_in'].isin(listing)], hue='rating')
10    ax.set_title(topic)
11    plt.subplots_adjust(left=0.1,
12                         bottom=0.1,
13                         right=0.9,
14                         top=1.5,
15                         wspace=0.5,
16                         hspace=2.0)
17    plt.xlabel('Genre')
18    plt.xticks(rotation = 50)
19    ax.set(ylabel='Content Produced')
```



```
In [66]:  
1 fig = plt.figure(  
2     figsize=(20,32)  
3     )  
4  
5 for i, name in enumerate(countries):  
6     frame = df2[df2['main_country'] == str(name)]  
7     ax = fig.add_subplot(len(countries),1,i+1)  
8     topic = name  
9     sns.countplot(x='rating', data= frame[frame['rating'].isin(ratings)])  
10    ax.set_title(topic)  
11    plt.subplots_adjust(left=0.1,  
12                         bottom=0.1,  
13                         right=0.9,  
14                         top=0.9,  
15                         wspace=0.4,  
16                         hspace=0.4)  
17    plt.xlabel('Rating')  
18    ax.set(ylabel='Content Produced')
```