

```
#Import libraries
```

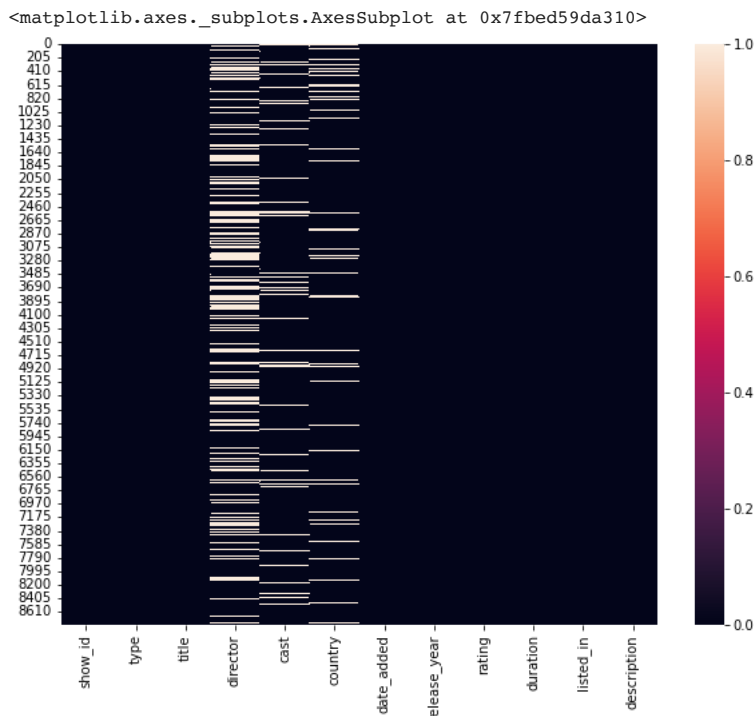
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
data = pd.DataFrame(pd.read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv'))
data.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	D
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	Internation T
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crin Internation
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseri
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K	India	September 24, 2021	2021	TV-MA	2 Seasons	Internation Romantic T

▼ Find out the missing or NaN values in the dataset

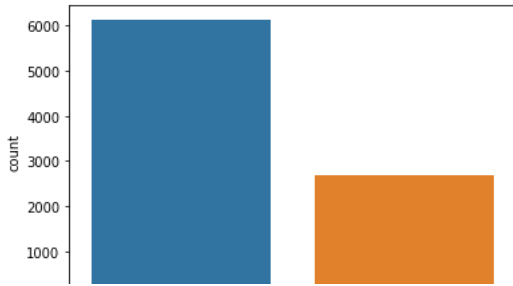
```
plt.figure(figsize=(10,8))
sns.heatmap(data.isna())
```



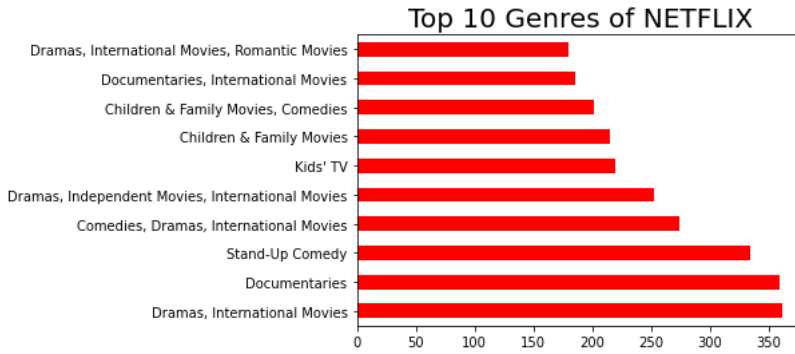
1. As we can see, a lot of records in the data doesn't have values for **Director, Cast and Country** fields
2. Since we are interested in increasing the revenue of Netflix, our main objective is to figure out which all shows and movies performed the best.
3. We will firstly look at the distribution of how many tv shows and movies are present in netflix. Later on, we will specifically start analysing more details about the ones that fetched the best reviews

```
sns.countplot(x= 'type', data=data)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fbed58456a0>



```
# plt.figure(figsize=(22,10))
data["listed_in"].value_counts()[:10].plot(kind="barh", color="red")
plt.title("Top 10 Genres of NETFLIX",size=20);
```



1. We have a lot of movies compared to TV shows, hence we will split our data so that our analysis is not skewed
2. Before that, we need to clean/preprocess our data to get better insights

▼ Movies

```
df = data.copy()
df = df[df['type'] == 'Movie']

df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	
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	C
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	Movi
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	Co
12	s13	Movie	Le Suie Karl	Christian	Luna Wedler, Jannis Niewöhner	Germany,	September	2021	TV-MA	127 min	

1. Since we don't have much values in the **Director** and **Cast** columns, we will drop them
2. However we cannot drop the countries as that is one of our primary attribute for this case study

```
df.drop(['director', 'cast'], axis=1, inplace=True)
df.head()
```

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	United States	September 25, 2021	2020	PG-13	90 min	Documentaries, International Movies
6	s7	Movie	My Little Pony: A New Generation	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies
7	s8	Movie	Sankofa	United States, Ghana, Burkina Faso, United Kingdom	September 24, 2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies
9	s10	Movie	The Starlina	United States	September 24, 2021	2021	PG-13	104 min	Comedies, Documentaries

df[df['country'].isnull()]

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in
6	s7	Movie	My Little Pony: A New Generation	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies
13	s14	Movie	Confessions of an Invisible Girl	NaN	September 22, 2021	2021	TV-PG	91 min	Children & Family Movies, Comedies
16	s17	Movie	Europe's Most Dangerous Man: Otto Skorzeny in ...	NaN	September 22, 2021	2020	TV-MA	67 min	Documentaries, International Movies
18	s19	Movie	Intrusion	NaN	September 22, 2021	2021	TV-14	94 min	Thrillers
22	s23	Movie	Avvai Shanmughi	NaN	September 21, 2021	1996	TV-PG	161 min	Comedies, International Movies
...
8585	s8586	Movie	Three-Quarters Decent	NaN	June 20, 2019	2010	TV-14	96 min	Comedies, Dramas, International Movies
8602	s8603	Movie	Tom and Jerry: The Magic Ring	NaN	December 15, 2019	2001	TV-Y7	60 min	Children & Family Movies, Comedies
8622	s8623	Movie	Tremors 2: Aftershocks	NaN	January 1, 2020	1995	PG-13	100 min	Comedies, Horror Movies, Sci-Fi & Fantasy
8718	s8719	Movie	Westside vs. the World	NaN	August 9, 2019	2019	TV-MA	96 min	Documentaries, Sports Movies
8759	s8760	Movie	World's Weirdest Homes	NaN	February 1, 2019	2015	TV-PG	49 min	Movies

440 rows x 10 columns

1. Assuming the data might have been collected with some sequence, we will probably start imputing with forward fill method instead of using mode or mean values to impute the data.

Note: We can train a model just to impute the values, but that will consume a lot of our time hence parked for later improvements.

```
df['country'] = df['country'].ffill(axis=0)
df[df['country'].isnull()]
```

	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
	7527	s7528	Movie	My Hero Academia: Two Heroes	Japan	March 1, 2017	2015	NaN	115 min	Dramas

Since there are only 2 of them, we can manually fill there values from internet

```
ratings = ['TV-PG', 'TV-MA']

for id, rating in zip(df[df['rating'].isnull()].index, ratings):
    df['rating'].loc[id] = rating

/usr/local/lib/python3.8/dist-packages/pandas/core/indexing.py:1732: 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
self._setitem_single_block(indexer, value, name)
```

```
df[df['rating'].isnull()]
```

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

For records, where date was null, we will drop them since they are quite scarce and won't affect our analysis much.

We will look into the same later on by imputing the values manually as they are also not much in count

Let us verify again whether there is a null data or not

```
df = df[df['date_added'].notna()]
df.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
```

As we can see, the country column holds multiple values of countries where the content was displayed. Assuming the first country will be the major source of the content we are going to use those for our further analysis

```
df['main_country'] = df['country'].apply(lambda x: x.split(',')[0])
df
```

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	
0	s1	Movie	Dick Johnson Is Dead	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As I en
6	s7	Movie	My Little Pony: A New Generation	United States	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equ a br
7	s8	Movie	Sankofa	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	i C
9	s10	Movie	The Starling	United States	September 24, 2021	2021	PG-13	104 min	Comedies, Dramas	A
12	s13	Movie	Je Suis Karl	Germany, Czech Republic	September 23, 2021	2021	TV-MA	127 min	Dramas, International Movies	Afte is i
...	
8801	s8802	Movie	Zinzana	United Arab Emirates, Jordan	March 9, 2016	2015	TV-MA	96 min	Dramas, International Movies, Thrillers	R Tala
8802	s8803	Movie	Zodiac	United States	November 20, 2019	2007	R	158 min	Cult Movies, Dramas, Thrillers	A p crit
8804	s8805	Movie	Zombieland	United States	November 1, 2019	2009	R	88 min	Comedies, Horror Movies	Loc world
8805	s8806	Movie	Zoom	United States	January 11, 2020	2006	PG	88 min	Children & Family Movies, Comedies	D
8806	s8807	Movie	Zubaan	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A sc w

Let us start our analysis for problem statement.

We will divide the data based on the top 10 main countries we found and group the data based on their genres and ratings.

Since we will have insights into what kind of content fetched most good reviews in which country based on top genres, we can promote the development of similar content in future

For ease of analysis, let's pick top 10 countries with maximum content produced

```
top_countries = df.groupby('main_country').count().sort_values('type', ascending=False)[:10]
top_countries
```

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
main_country										
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

For ease of analysis, let's pick top 10 genres with maximum content produced

```
top_genres = df.groupby('listed_in').count().sort_values('type', ascending=False)[:10]
top_genres
```

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

Filtering the data for top 10 genres

```
df = df[df['listed_in'].isin(list(top_genres.index))]
```

Filtering the data for top 10 countries

```
df = df[df['main_country'].isin(list(top_countries.index))]
```

df

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	
0	s1	Movie	Dick Johnson Is Dead	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As e
6	s7	Movie	My Little Pony: A New Generation	United States	September 24, 2021	2021	PG	91 min	Children & Family Movies	Eq a t
7	s8	Movie	Sankofa	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	
30	s31	Movie	Ankahi Kahaniya	United States, India, France	September 17, 2021	2021	TV-14	111 min	Dramas, Independent Movies, International Movies	/ arot
45	s46	Movie	My Heroes Were Cowboys	United States	September 16, 2021	2021	PG	23 min	Documentaries	Rol chi
...	

```

ratings=[]
for rate in df['rating'].unique():
    ratings.append(rate)

countries = df['main_country'].unique()

listing = df['listed_in'].unique()

ratings, countries, listing

(['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, Comedies',
       'Comedies, Dramas, International Movies',
       'Documentaries, International Movies',
       'Dramas, International Movies, Romantic Movies',
       'Comedies, International Movies', 'Stand-Up Comedy'], dtype=object))

```

Top content based on genres and the top ratings within each in different countries is displayed below

```

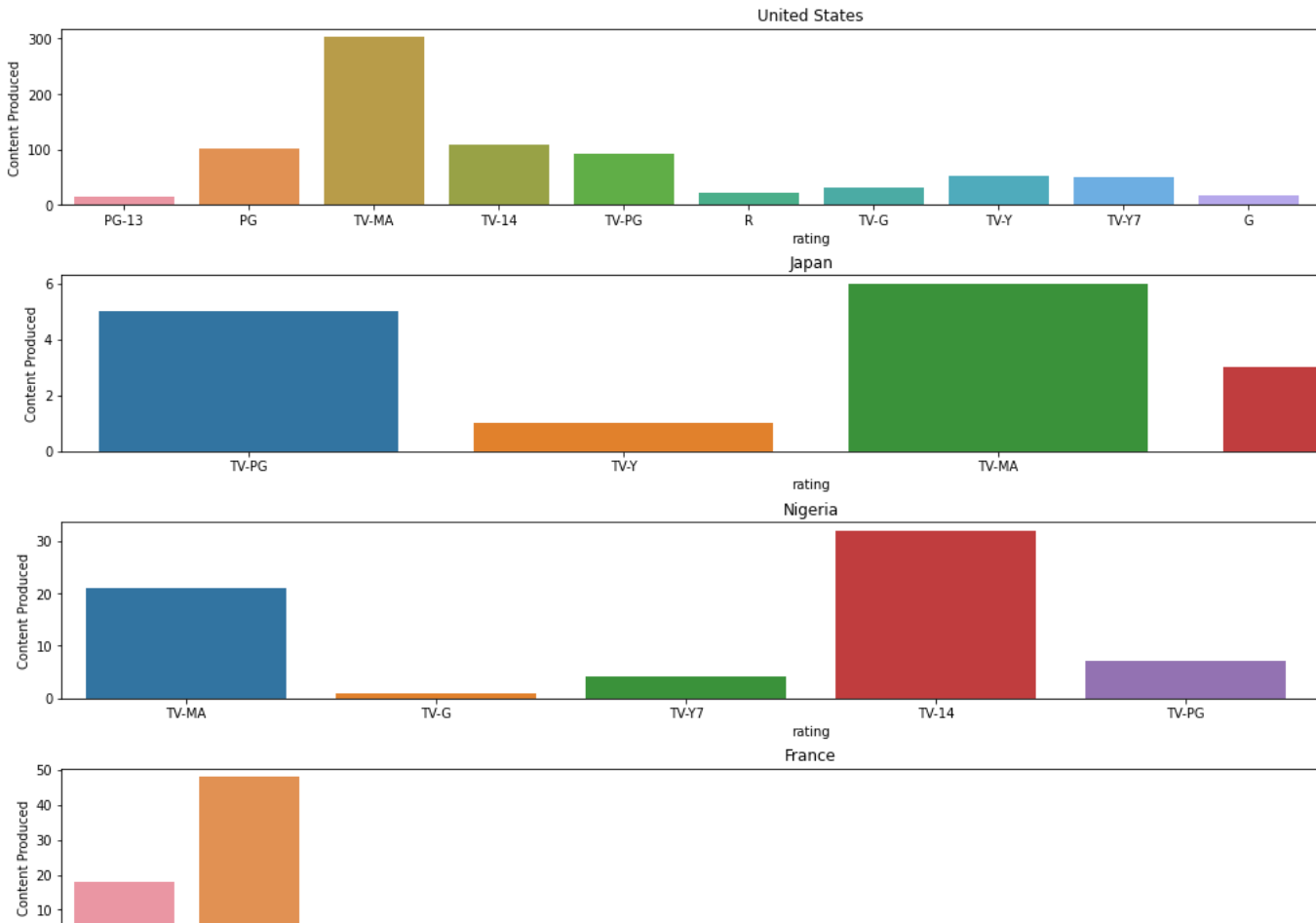
fig = plt.figure(
    figsize=(30,40)
)

for i, name in enumerate(countries):
    frame = df[df['main_country'] == str(name)]
    ax = fig.add_subplot(len(countries),1,i+1)
    topic = name
    sns.countplot(x='listed_in', data= frame[frame['listed_in'].isin(listing)], hue='rating')
    ax.set_title(topic)
    plt.subplots_adjust(left=0.1,
                        bottom=0.1,
                        right=0.9,
                        top=1.5,
                        wspace=0.5,
                        hspace=2.0)
    plt.xlabel('Genre')
    plt.xticks(rotation = 50)
    ax.set(ylabel='Content Produced')

```



```
fig = plt.figure(  
    figsize=(20,32)  
)  
  
for i, name in enumerate(countries):  
    frame = df[df['main_country'] == str(name)]  
    ax = fig.add_subplot(len(countries),1,i+1)  
    topic = name  
    sns.countplot(x='rating', data= frame[frame['rating'].isin(ratings)])  
    ax.set_title(topic)  
    plt.subplots_adjust(left=0.1,  
                        bottom=0.1,  
                        right=0.9,  
                        top=0.9,  
                        wspace=0.4,  
                        hspace=0.4)  
    ax.set(ylabel='Content Produced')
```

From the above graph, we can clearly visualize what type of content and ratings are attracting the audience in top 10 countries.

Note: Top 10 genres were considered for displaying the above data

All the steps and analysis done for Movies is repeated for TV Shows as well and conclusions are drawn from them on a similar basis

TV Show

```
df = data.copy()
df = df[df['type'] == 'TV Show']

df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	Inti Shows
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crim Inti Shc
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuserie
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	Inti Shows,
5	s6	TV	Midnight	Mike	Kate Siegel, Zach Gilford, Hamish	NaN	September 24,	2021	TV-MA	1 Season	TV Drama

```
df.drop(['director', 'cast'], axis=1, inplace=True)
df.head()
```

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in
1	s2	TV Show	Blood & Water	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries
2	s3	TV Show	Ganglands	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...

```
df[df['country'].isnull()]
```

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in
2	s3	TV Show	Ganglands	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act
3	s4	TV Show	Jailbirds New Orleans	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV
5	s6	TV Show	Midnight Mass	NaN	September 24, 2021	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries
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 Shows
11	s12	TV Show	Bangkok Breaking	NaN	September 23, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act
...
8679	s8680	TV Show	ViR: The Robot Boy	NaN	March 31, 2018	2013	TV-Y7	2 Seasons	Kids' TV Shows
8690	s8691	TV Show	Wake Up	NaN	March 31, 2018	2017	TV-14	2 Seasons	International TV Shows, TV Dramas
8783	s8784	TV Show	Yoko	NaN	June 23, 2018	2016	TV-Y	1 Season	Kids' TV Shows

```
df['country'] = df['country'].ffill(axis=0)
df[df['country'].isnull()]
```

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

```
df[df['rating'].isnull()]
```

	show_id	type	title	country	date_added	release_year	rating	duration
6827	s6828	TV	Gargantia on the ...	Japan	December 1, ...	2013	NaN	1 Season

```
ratings = ['TV-14', 'TV-MA']

for id, rating in zip(df[df['rating'].isnull()].index, ratings):
    df['rating'].loc[id] = rating

/usr/local/lib/python3.8/dist-packages/pandas/core/indexing.py:1732: 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-self-view
self._setitem_single_block(indexer, value, name)
```

```
df[df['rating'].isna()]
```

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

```
df = df[df['date_added'].notna()]
df.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

df['main_country'] = df['country'].apply(lambda x: x.split(',')[0])
df

<ipython-input-30-3c9495461b4a>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vs-returning-a-copy
df['main_country'] = df['country'].apply(lambda x: x.split(',')[0])
```

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	
1	s2	TV Show	Blood & Water	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After the rain
2	s3	TV Show	Ganglands	South Africa	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Action	To protect and serve
3	s4	TV Show	Jailbirds New Orleans	South Africa	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feud: The Bitchy Women
4	s5	TV Show	Kota Factory	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV Dramas	century
5	s6	TV Show	Midnight Mass	India	September 24, 2021	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	chance
...
8795	s8796	TV Show	Yu-Gi-Oh! Arc-V	Japan, Canada	May 1, 2018	2015	TV-Y7	2 Seasons	Anime Series, Kids' TV	Now
8796	s8797	TV Show	Yunus Emre	Turkey	January 17, 2017	2016	TV-PG	2 Seasons	International TV Shows, TV Dramas	in
8797	s8798	TV Show	Zak Storm	United States, France, South Korea, Indonesia	September 13, 2018	2016	TV-Y7	3 Seasons	Kids' TV	Teen
8800	s8801	TV Show	Zindagi Gulzar Hai	Pakistan	December 15, 2016	2012	TV-PG	1 Season	International TV Shows, Romantic TV Shows, TV Dramas	Strong

```
# For ease of analysis, let's pick top 10 countries with maximum content produced

top_countries = df.groupby('main_country').count().sort_values('type', ascending=False)[:10]
top_countries
```

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description
main_country										
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

```
# For ease of analysis, let's pick top 10 genres with maximum content produced

top_genres = df.groupby('listed_in').count().sort_values('type', ascending=False)[:10]
top_genres
```

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

```
# Filtering the data for top 10 genres
```

```
df = df[df['listed_in'].isin(list(top_genres.index))]
```

```
# Filtering the data for top 10 countries
```

```
df = df[df['main_country'].isin(list(top_countries.index))]
```

```
ratings=[]
```

```
for rate in df['rating'].unique():
    ratings.append(rate)
```

```
countries = df['main_country'].unique()
```

```
listing = df['listed_in'].unique()
```

```
ratings, countries, listing
```

```
(['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))
```

```
fig = plt.figure(
    figsize=(30,40)
)
```

```
for i, name in enumerate(countries):
    frame = df[df['main_country'] == str(name)]
    ax = fig.add_subplot(len(countries),1,i+1)
    topic = name
    sns.countplot(x='listed_in', data= frame[frame['listed_in'].isin(listing)], hue='rating')
    ax.set_title(topic)
    plt.subplots_adjust(left=0.1,
                        bottom=0.1,
                        right=0.9,
                        top=1.5,
                        wspace=0.5,
                        hspace=2.0)
    plt.xlabel('Genre')
    plt.xticks(rotation = 50)
    ax.set(ylabel='Content Produced')
```

