Python初級數據分析員證書

（六）數據分析 及可視化專案

13. 數據分析專案

Demo 7 – Netflix

Review

- Statistics
- Hypothesis testing
- Algebra
- Linear regression
- Propositional logic
- Python
- R
- SQL

- Pandas, NumPy, SciPy
- Data Visualization, Matplotlib, Seaborn, Plotly
- Dashboard Visualization, Business Intelligence
- Storytelling

Chapter Summary

- Scenario
- Data Import
- Movie statistics
- TV Series statistics
- Hong Kong statistics

Scenario

Netflix Inc. is an American media company based in Los Gatos, California, founded in 1997. Its streaming service offers a wide variety of award-winning TV shows, movies, anime, and documentaries.

You boss would like to acquire part of its share business in Asia, and ask you to analyse its media products in general.

Data Import

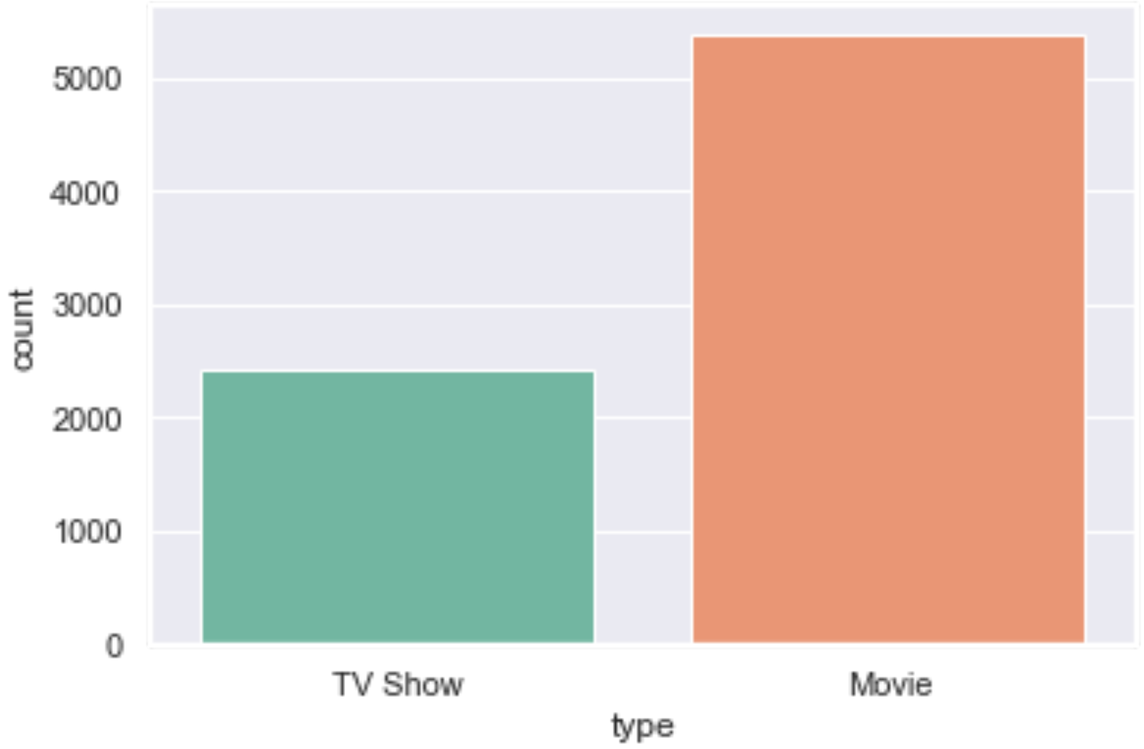
```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 import matplotlib.pyplot as plt
6
```

```
1 netflix_overall = pd.read_csv("netflix_titles.csv")
2 netflix_overall.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	TV Show	3%	NaN	João Miguel, Bianca Comparato, Michel Gomes, R...	Brazil	August 14, 2020	2020	TV-MA	4 Seasons	International TV Shows, TV Dramas, TV Sci-Fi &...	In a future where the elite inhabit an island ...
1	s2	Movie	7:19	Jorge Michel Grau	Demián Bichir, Héctor Bonilla, Oscar Serrano, ...	Mexico	December 23, 2016	2016	TV-MA	93 min	Dramas, International Movies	After a devastating earthquake hits Mexico Cit...
2	s3	Movie	23:59	Gilbert Chan	Tedd Chan, Stella Chung, Henley Hii, Lawrence ...	Singapore	December 20, 2018	2011	R	78 min	Horror Movies, International Movies	When an army recruit is found dead, his fellow...
3	s4	Movie	9	Shane Acker	Elijah Wood, John C. Reilly, Jennifer Connelly...	United States	November 16, 2017	2009	PG-13	80 min	Action & Adventure, Independent Movies, Sci-Fi...	In a postapocalyptic world, rag-doll robots hi...
4	s5	Movie	21	Robert Luketic	Jim Sturgess, Kevin Spacey, Kate Bosworth, Aar...	United States	January 1, 2020	2008	PG-13	123 min	Dramas	A brilliant group of students become card-coun...

TV Show vs Movie

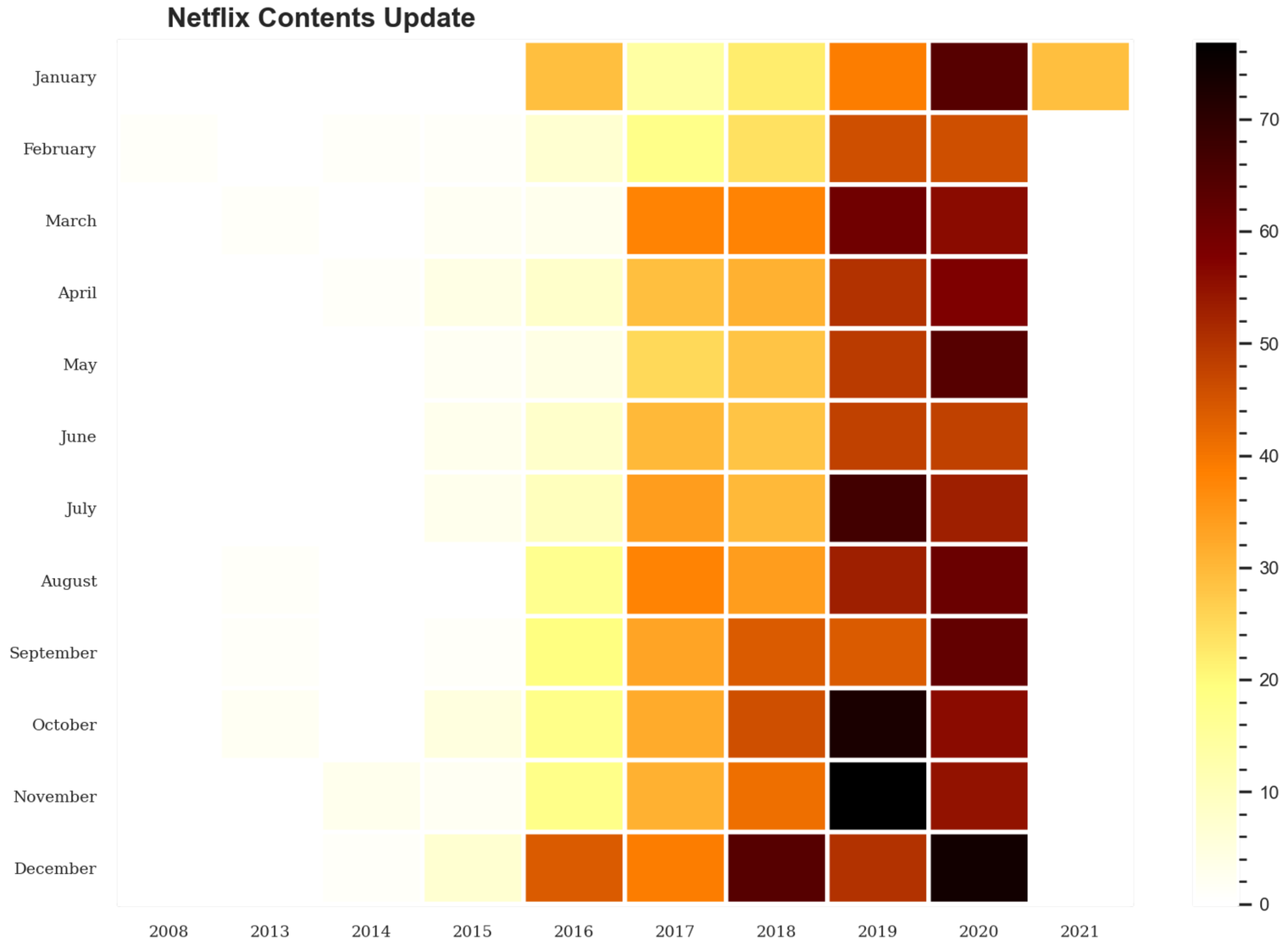
```
1 netflix_shows=netflix_overall[netflix_overall['type']=='TV Show']
2 netflix_movies=netflix_overall[netflix_overall['type']=='Movie']
3
4 sns.set(style="darkgrid")
5 ax = sns.countplot(x="type", data=netflix_overall, palette="Set2")
```



Monthly Content Update Number

```
netflix_date = netflix_shows[['date_added']].dropna()
netflix_date['year'] = netflix_date['date_added'].apply(lambda x : x.split(', ')[-1])
netflix_date['month'] = netflix_date['date_added'].apply(lambda x : x.lstrip().split(' ')[0])

month_order = ['January', 'February', 'March', 'April', 'May', 'June',
               'July', 'August', 'September', 'October', 'November', 'December'][::-1]
df = netflix_date.groupby('year')['month'].value_counts().unstack().fillna(0)[month_order].T
plt.figure(figsize=(10, 7), dpi=200)
plt.pcolor(df, cmap='afmhot_r', edgecolors='white', linewidths=2) # heatmap
plt.xticks(np.arange(0.5, len(df.columns), 1), df.columns, fontsize=7, fontfamily='serif')
plt.yticks(np.arange(0.5, len(df.index), 1), df.index, fontsize=7, fontfamily='serif')

plt.title('Netflix Contents Update', fontsize=12, fontweight='bold', position=(0.20, 1.0+0.02))
cbar = plt.colorbar()

cbar.ax.tick_params(labelsize=8)
cbar.ax.minorticks_on()
plt.show()
```

Monthly Content Update Number



TV Rating (Parental Guideline)

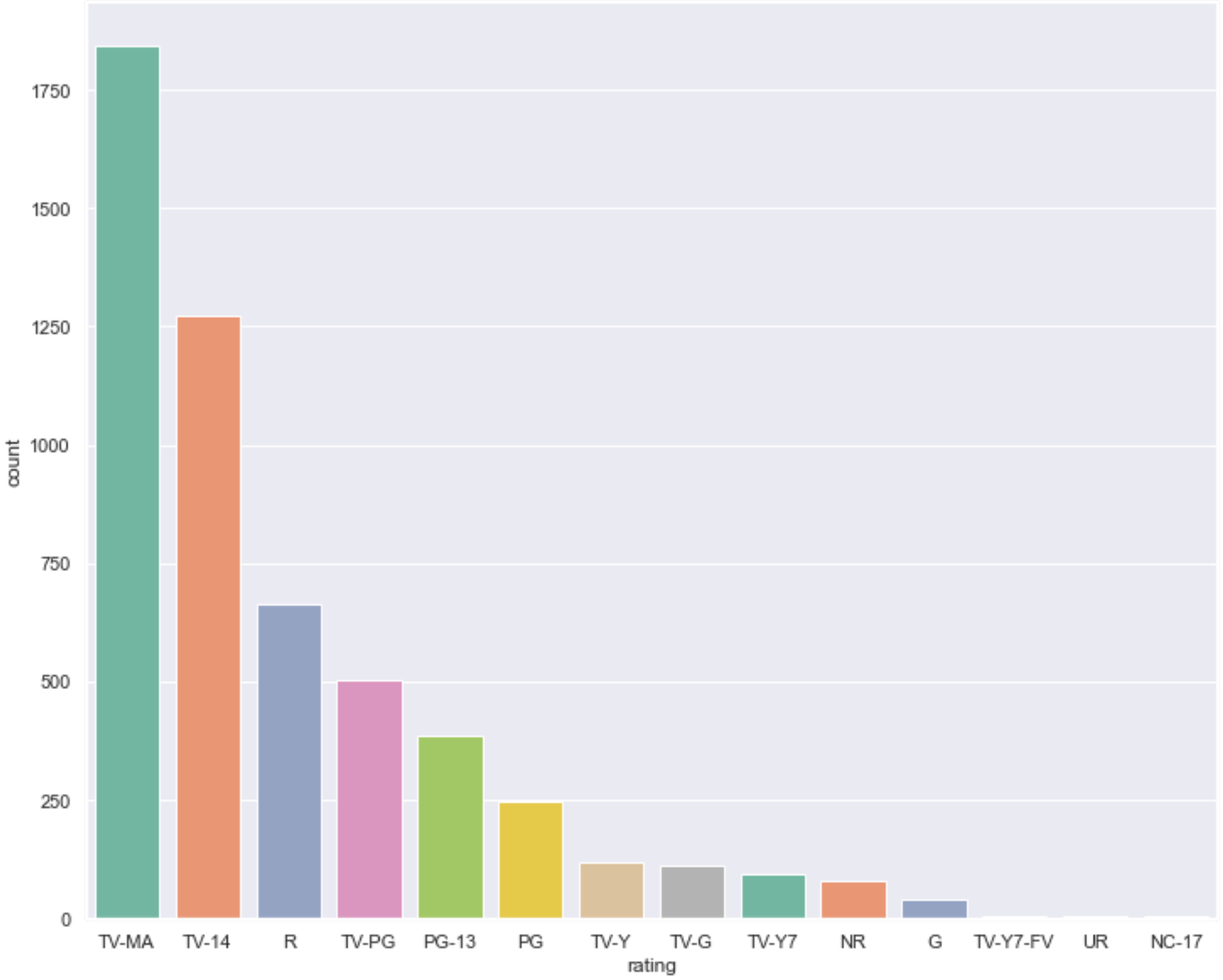
The largest count of movies are made with the 'TV-MA' rating. "TV-MA" is a rating assigned by the TV Parental Guidelines to a television program that was designed for mature audiences only.

```
1  netflix_movies['rating'].unique()
```

```
array(['TV-MA', 'R', 'PG-13', 'TV-14', 'TV-PG', 'NR', 'TV-G', 'TV-Y', nan,
       'PG', 'G', 'TV-Y7', 'NC-17', 'TV-Y7-FV', 'UR'], dtype=object)
```

```
1  plt.figure(figsize=(12,10))
2  sns.set(style="darkgrid")
3  ax = sns.countplot(x="rating", data=netflix_movies, palette="Set2",
4                     order=netflix_movies['rating'].value_counts().index[0:15])
```

TV Rating (Parental Guideline)



Merge IMDb rating and movie data

There are 2 IMDb CSV file. We merge it on title in Netflix.

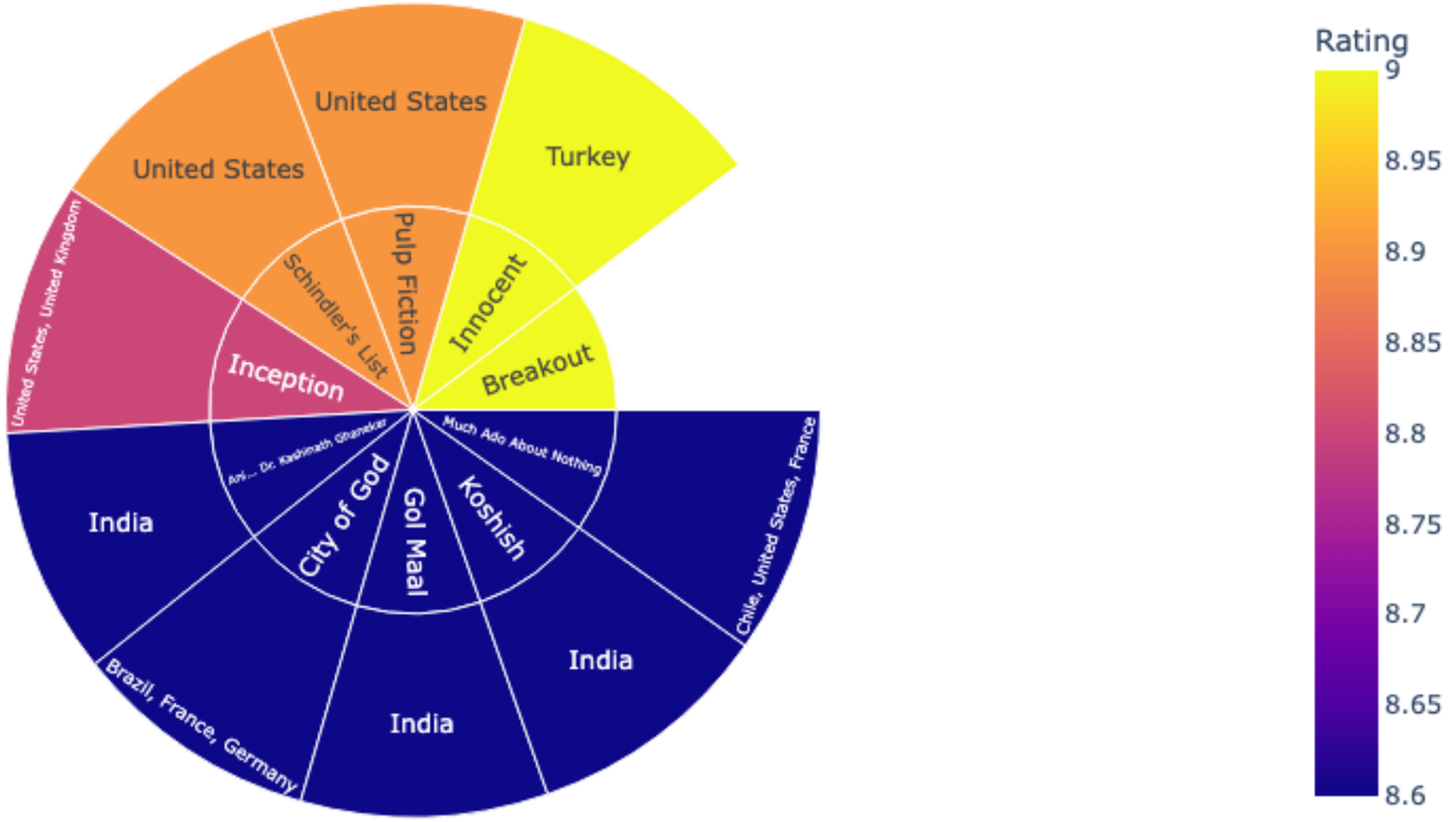
```
imdb_ratings = pd.read_csv('IMDb_ratings.csv',usecols=['weighted_average_vote'])
imdb_titles = pd.read_csv('IMDb_movies.csv', usecols=['title','year','genre'],low_memory=False)
ratings = pd.DataFrame({'Title':imdb_titles.title,
                        'Release Year':imdb_titles.year,
                        'Rating': imdb_ratings.weighted_average_vote,
                        'Genre':imdb_titles.genre})
ratings.drop_duplicates(subset=['Title','Release Year','Rating'], inplace=True)
ratings.shape
```

```
(85852, 4)
```

```
ratings.dropna()
joint_data = ratings.merge(netflix_overall,left_on='Title',right_on='title',how='inner')
joint_data = joint_data.sort_values(by='Rating', ascending=False)
```

Top rated 10 movies on Netflix

```
1  top_rated=joint_data[0:10]
2  fig =px.sunburst( top_rated, path=['title','country'],
3      values='Rating', color='Rating')
4  fig.show()
```



Countries with highest rated content

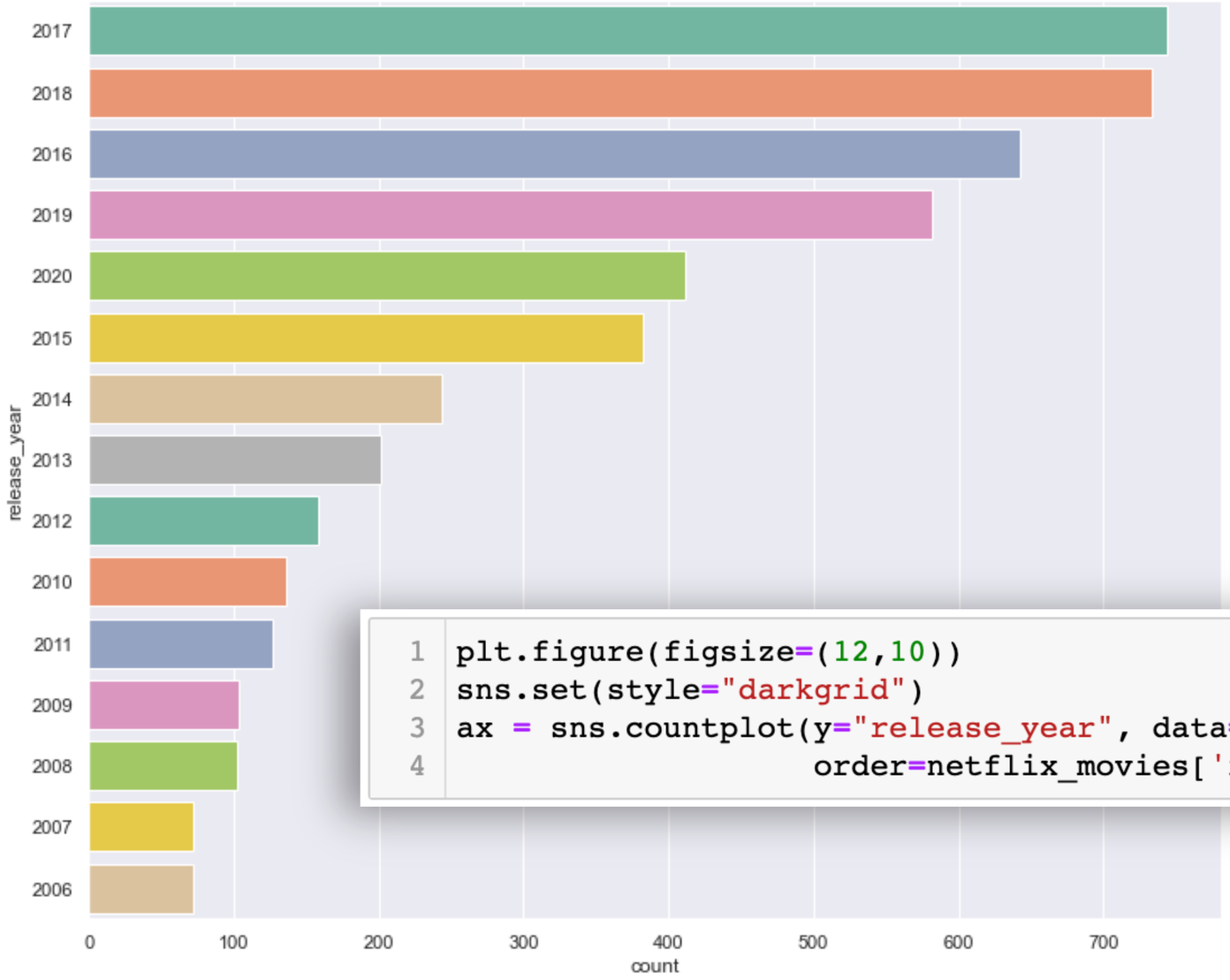
```
country_count=joint_data['country'].value_counts().sort_values(ascending=False)
country_count=pd.DataFrame(country_count)
topcountries=country_count[0:11]
topcountries
```

	country
United States	799
India	701
United Kingdom	107
Canada	56
Philippines	50
Spain	40
South Korea	36
Indonesia	35
France	33
United Kingdom, United States	31
Australia	30

```
# Hong Kong people don't know IMDb much
country_count.loc[country_count.index=='Hong Kong']
```

	country
Hong Kong	8

Movies released per year




```
plt.figure(figsize=(12,10))
sns.set(style="darkgrid")
ax = sns.countplot(y="release_year", data=netflix_movies, palette="Set2",
                   order=netflix_movies['release_year'].value_counts().index[0:15])
```

Production countries

We like to count which countries produce the most movie.

Yet some movie was produced in a few countries.

```
1 # some movie was filmed and produced in a few countries
2 netflix_movies['country'].sample(5)
```

```
5626                                          India
2997                          Germany, United States
6408    United Kingdom, United States, Morocco
1614                                         Mexico
3299                                   South Africa
Name: country, dtype: object
```

Write a short code to separate the country name

```
countries={}
netflix_movies.loc[netflix_movies.index, ['country']] = netflix_movies['country'].fillna('Unknown')
cou=list(netflix_movies['country'])
for i in cou:
    i=list(i.split(','))
    if len(i)==1:
        if i in list(countries.keys()):
            countries[i]+=1
        else:
            countries[i[0]]=1
    else:
        for j in i:
            if j in list(countries.keys()):
                countries[j]+=1
            else:
                countries[j]=1
```

```
countries
```

```
{'Mexico': 1,
 'Singapore': 3,
 'United States': 1,
 'Egypt': 1,
 'India': 1,
 'Thailand': 1,
 'Nigeria': 1,
 'Norway': 2,
 ' Iceland': 4,
 ' United States': 331,
 'United Kingdom': 2,
```

Sum up and sort the dict by its value

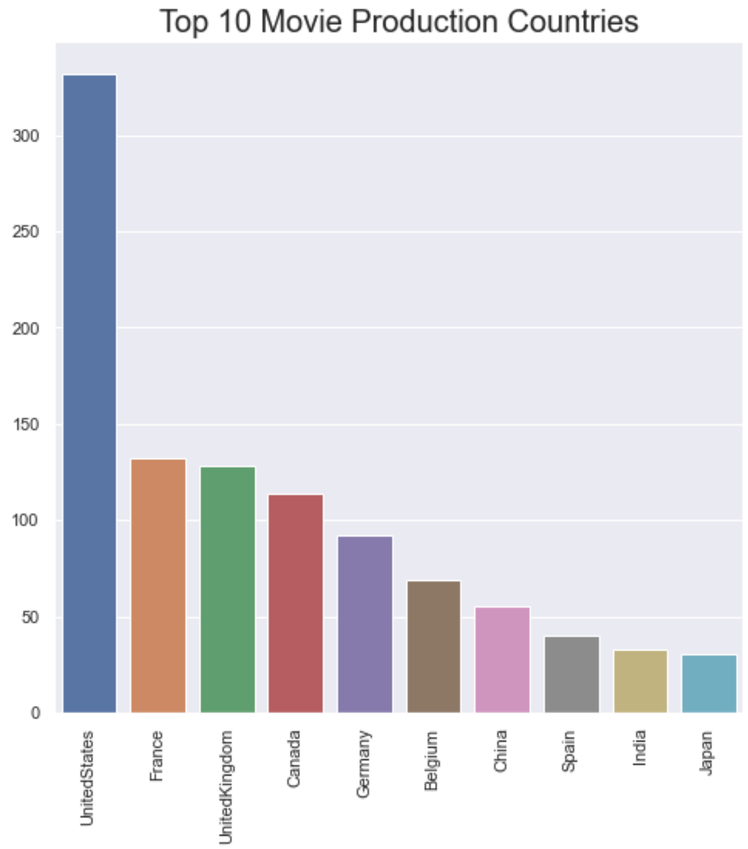
```
countries_fin = {}
for country,no in countries.items():
    country=country.replace(' ','')
    if country in list(countries_fin.keys()):
        countries_fin[country]+=no
    else:
        countries_fin[country]=no

countries_fin={k: v for k, v in sorted(countries_fin.items(), key=lambda item: item[1], reverse= True)}
```

```
countries_fin
```

```
{'UnitedStates': 332,
 'France': 132,
 'UnitedKingdom': 128,
 'Canada': 114,
 'Germany': 92,
 'Belgium': 69,
 'China': 55,
 'Spain': 40,
 'India': 33,
 'Japan': 30,
 'Australia': 29,
```

Top 10 Movie Production Countries

```
1 plt.figure(figsize=(8,8))
2 ax = sns.barplot(x=list(countries_fin.keys())[0:10], y=list(countries_fin.values())[0:10])
3 ax.set_xticklabels(list(countries_fin.keys())[0:10], rotation = 90)
4 ax.set_title('Top 10 Movie Production Countries', fontsize=20)
```



Duration of Movie

We like to analyse the duration of movies, however the data is in string format.

```
1  netflix_movies['duration']
```

```
1          93 min
2          78 min
3          80 min
4         123 min
6          95 min
           ...
7781       88 min
7782       99 min
7783      111 min
7784       44 min
7786       90 min
Name: duration, Length: 5377, dtype: object
```

Duration of Movie

Split the ‘min’ ad convert the string to int.

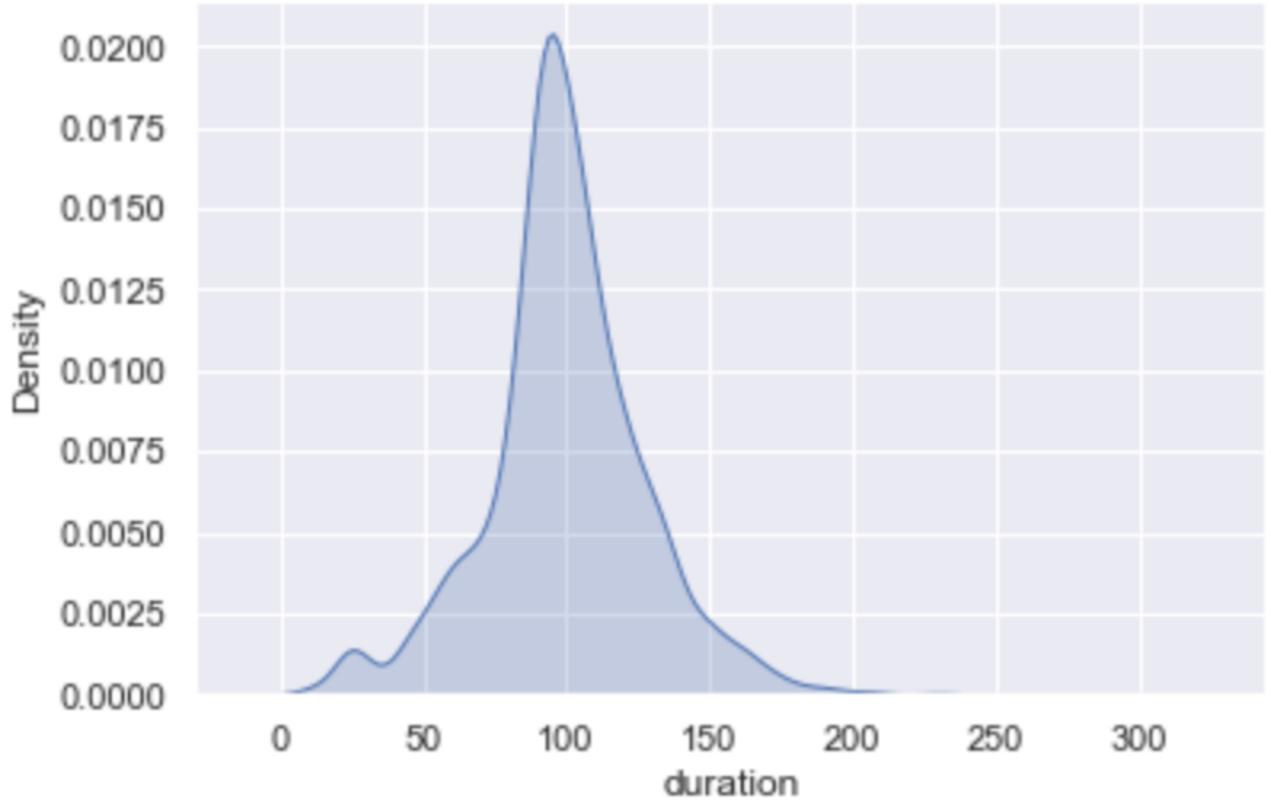
```
1 netflix_movies.loc[netflix_movies.index,['duration']] = netflix_movies['duration'].map(
2                                                         lambda x: int(x.split("min")[0]))
3 netflix_movies['duration']
```

```
1          93
2          78
3          80
4         123
6          95
         ...
7781       88
7782       99
7783      111
7784       44
7786       90
Name: duration, Length: 5377, dtype: object
```

Duration of Movie

```
1  sns.set(style="darkgrid")
2  sns.kdeplot(data=netflix_movies['duration'], fill=True)
```

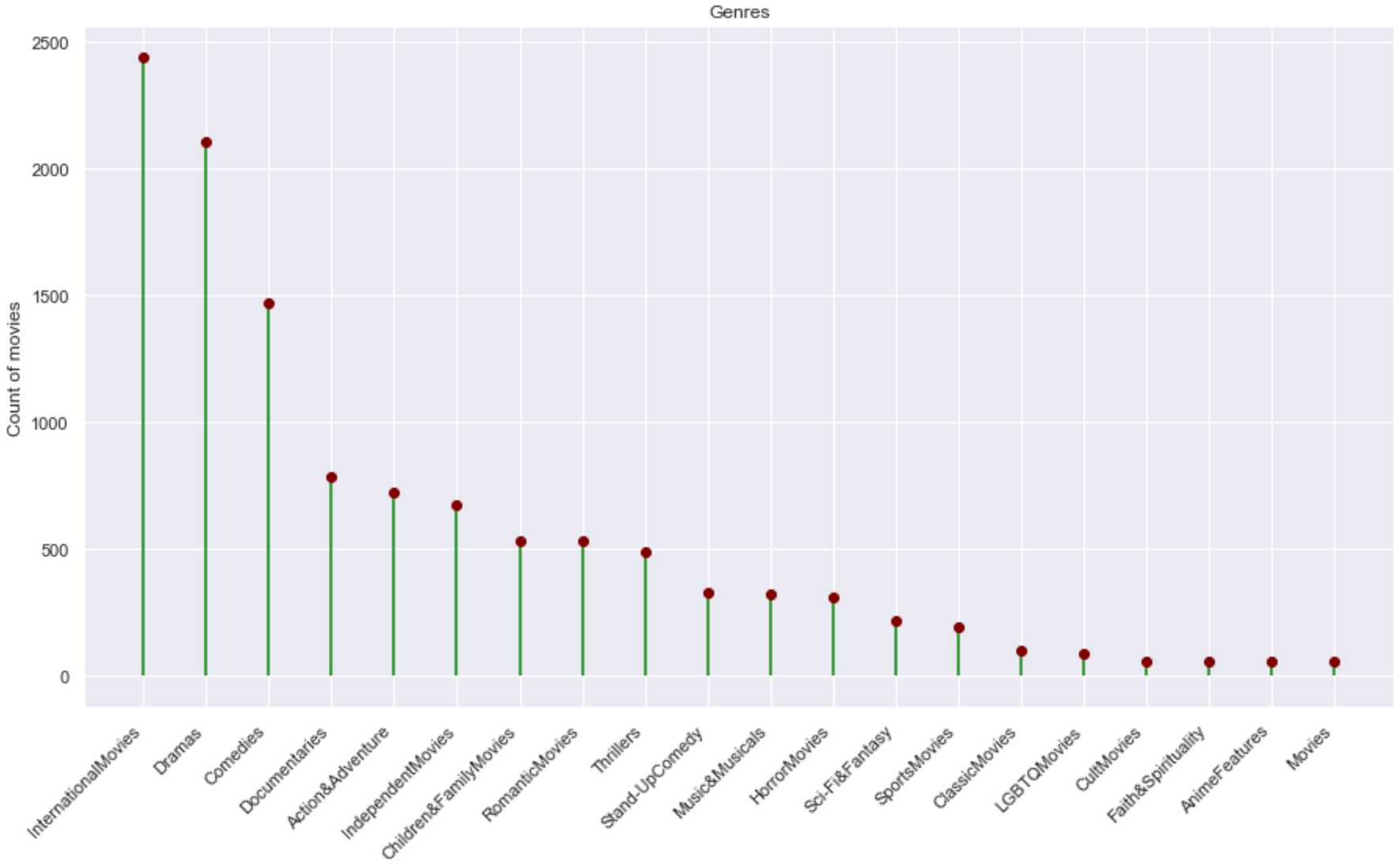
```
<AxesSubplot:xlabel='duration', ylabel='Density'>
```



Genres and their count

```
from collections import Counter

genres=list(netflix_movies['listed_in'])
gen=[]

for i in genres:
    i=list(i.split(','))
    for j in i:
        gen.append(j.replace(' ',""))
g=Counter(gen)
g={k: v for k, v in sorted(g.items(), key=lambda item: item[1], reverse= True)}

fig, ax = plt.subplots(figsize=(15,8))

fig = plt.figure(figsize = (10, 10))
x=list(g.keys())
y=list(g.values())
ax.vlines(x, ymin=0, ymax=y, color='green')
ax.plot(x,y, "o", color='maroon')
ax.set_xticklabels(x, rotation=45, ha='right')
ax.set_ylabel("Count of movies")
# set a title
ax.set_title("Genres")
```

Genres and their count



TV Series on Netflix

Again we like to know which countries produced the most number of of TV series

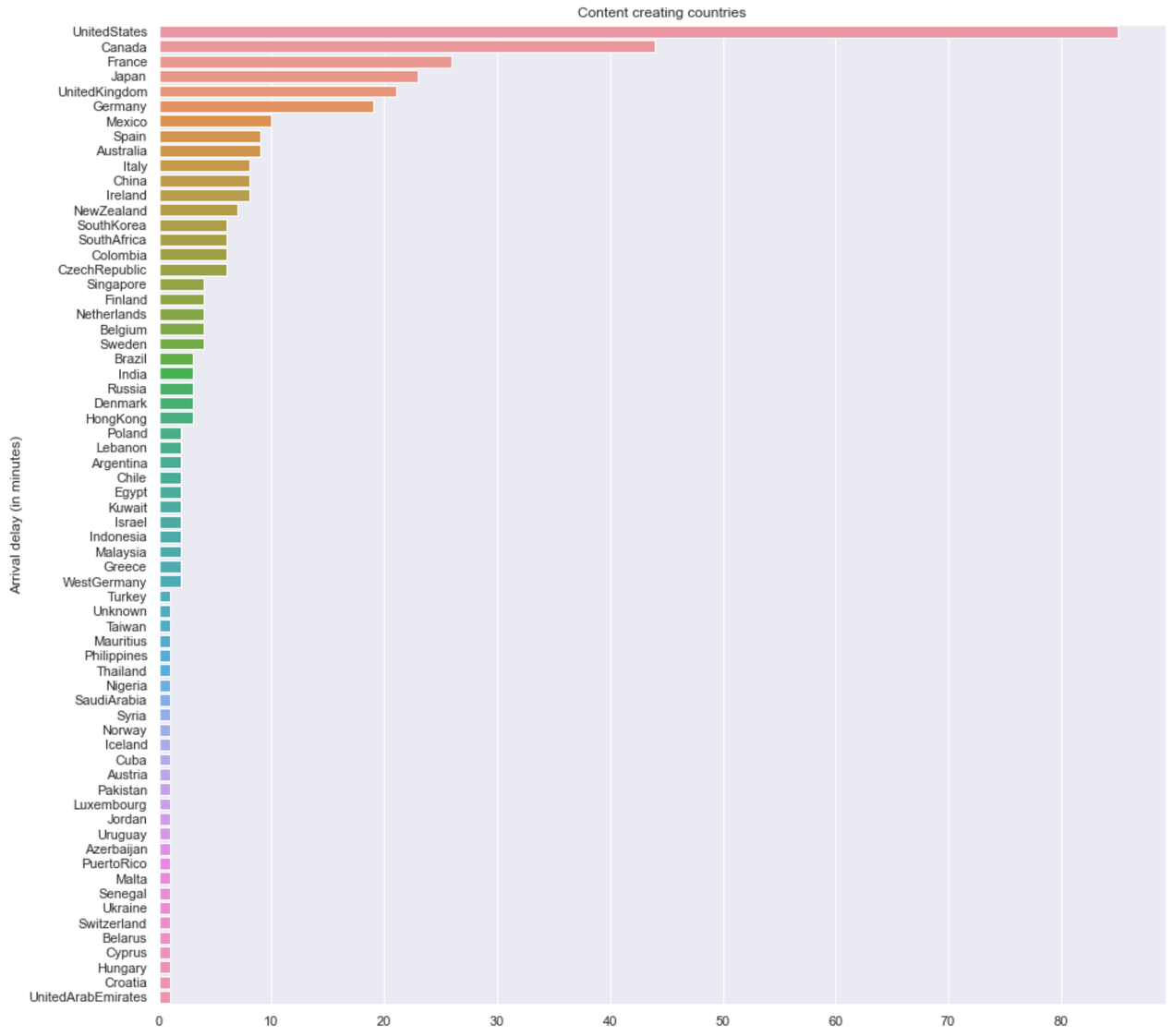
```
countries1={}
netflix_shows.loc[netflix_shows.index,['country']]=netflix_shows['country'].fillna('Unknown')
cou1=list(netflix_shows['country'])
for i in cou1:
    #print(i)
    i=list(i.split(','))
    if len(i)==1:
        if i in list(countries1.keys()):
            countries1[i]+=1
        else:
            countries1[i[0]]=1
    else:
        for j in i:
            if j in list(countries1.keys()):
                countries1[j]+=1
            else:
                countries1[j]=1
```

TV Series on Netflix

```
countries_fin1={}
for country,no in countries1.items():
    country=country.replace(' ','')
    if country in list(countries_fin1.keys()):
        countries_fin1[country]+=no
    else:
        countries_fin1[country]=no

countries_fin1={k: v for k, v in sorted(countries_fin1.items(), key=lambda item: item[1], reverse= True)}
```

```
plt.figure(figsize=(15,15))
plt.title("Content creating countries")
sns.barplot(y=list(countries_fin1.keys()), x=list(countries_fin1.values()))
plt.ylabel("Arrival delay (in minutes)")
```

TV Series on Netflix

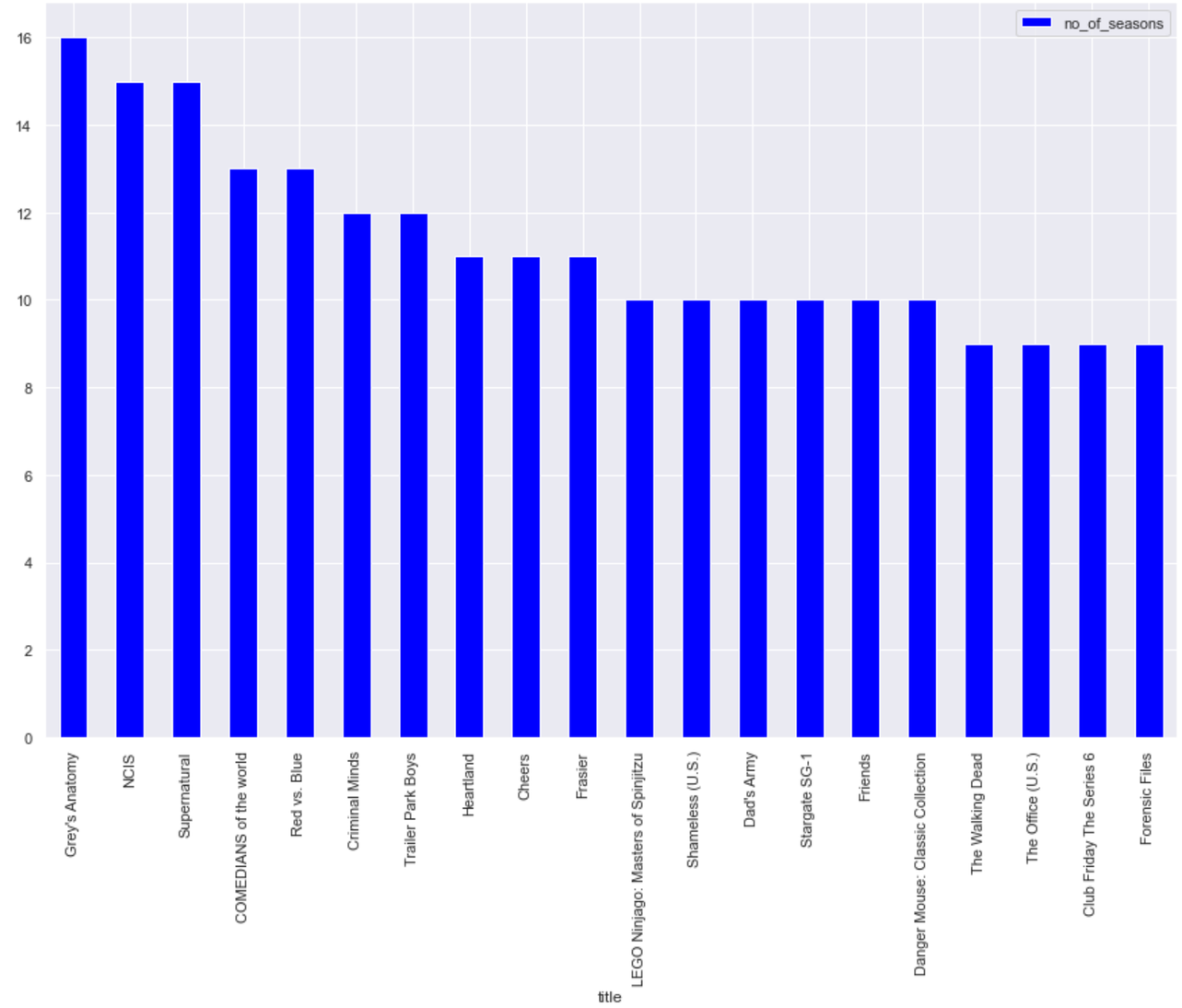


TV shows with largest number of seasons

```
features=['title','duration']
durations = netflix_shows[features]
durations.loc[durations.index,['no_of_seasons']]=durations['duration'].str.replace(' Season','')
durations.loc[durations.index,['no_of_seasons']]=durations['no_of_seasons'].str.replace('s','')
durations.loc[durations.index,['no_of_seasons']]=durations['no_of_seasons'].astype(str).astype(int)
```

```
t=['title','no_of_seasons']
top=durations[t]
top=top.sort_values(by='no_of_seasons', ascending=False)
top20=top[0:20]
top20.plot(kind='bar',x='title',y='no_of_seasons', color='blue',figsize=(15,10))
```

TV shows with largest number of seasons



TV shows with smallest number of seasons

```
1 bottom=top.sort_values(by='no_of_seasons')
2 bottom=bottom[20:50]
```

```
1 bottom
```

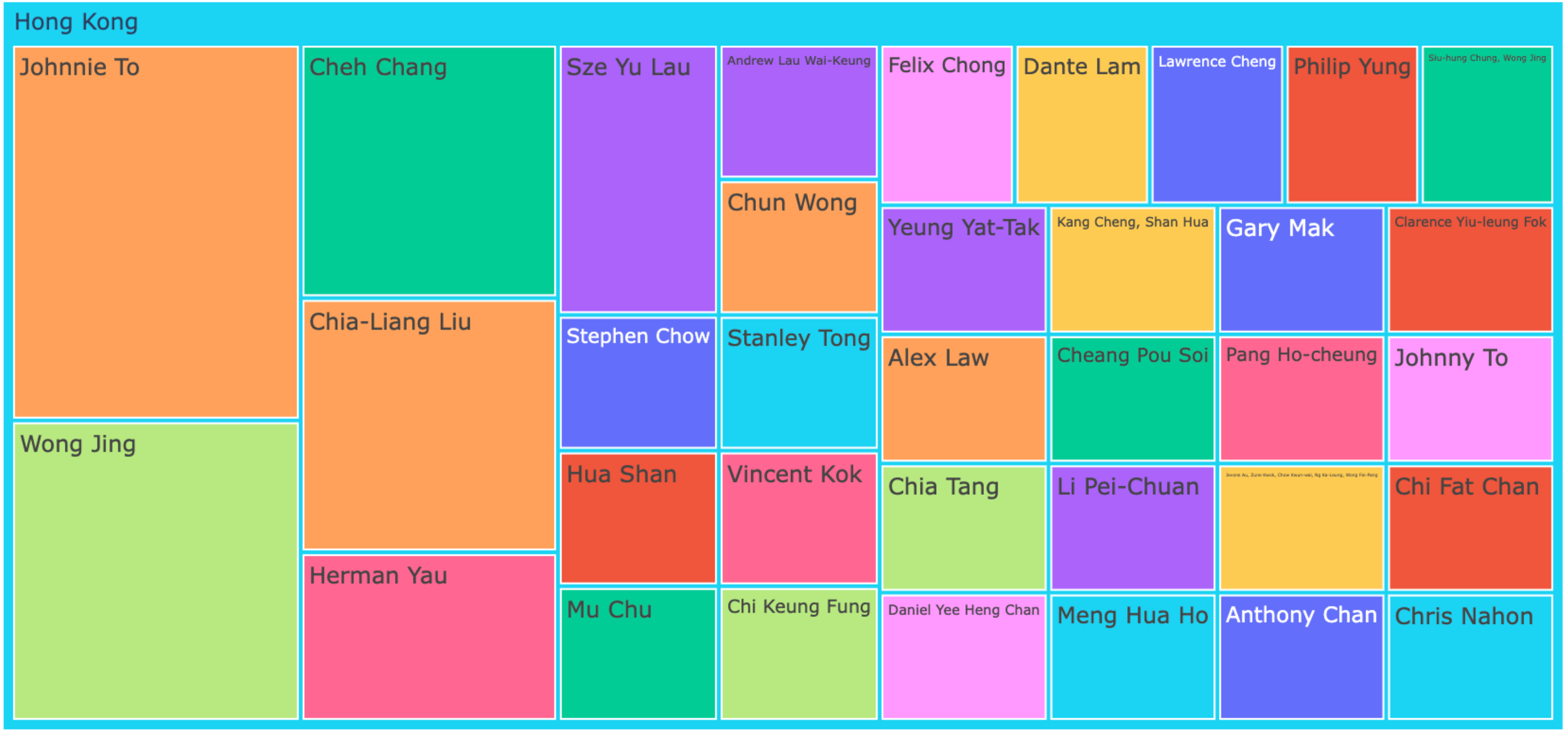
	title	no_of_seasons
5101	Rapture	1
5186	Revolting Rhymes	1
3340	Kevin Hart: Don't F**k This Up	1
3386	Killer Inside: The Mind of Aaron Hernandez	1
3379	Kill la Kill	1
1882	Drug Squad: Costa del Sol	1
3370	Kid-E-Cats	1
3368	Kicko & Super Speedo	1
1883	Drugs, Inc.	1

Content from Hong Kong

View Hong Kong data by directors.

```
netflix_hk = netflix_overall[netflix_overall['country']=='Hong Kong']
nfhk = netflix_hk.dropna()

fig = px.treemap(nfhk, path=['country','director'],
                 color='director', hover_data=['director','title'],
                color_continuous_scale='Purples')
fig.show()
```

Content from Hong Kong



View Hong Kong latest TV show and movies

```
newest_HK_series = netflix_hk.sort_values(by='release_year', ascending=False)[0:20]
newest_HK_series
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
187	s188	Movie	A Home with A View	Herman Yau	Francis Chun-Yu Ng, Louis Koo, Anita Yuen, Tat...	Hong Kong	April 30, 2019	2019	TV-MA	92 min	Comedies, International Movies	When a neighbor blocks their view of the city ...
5528	s5529	TV Show	Sexy Central	NaN	Jeana Ho Pui-yu, Joyce Cheng, Ava Liu, Shiga L...	Hong Kong	July 20, 2019	2019	TV-MA	1 Season	International TV Shows, Romantic TV Shows, TV ...	In the bustling center of Hong Kong, five youn...
7497	s7498	Movie	We Are Legends	Daniel Yee Heng Chan	Lam Yiu-sing, Ma Chi Wai, Wiyona Yeung, Eric K...	Hong Kong	June 1, 2019	2019	TV-14	109 min	Action & Adventure, International Movies, Spor...	Raised in a boxing gym, two orphaned brothers ...
2797	s2798	TV Show	Hong Kong West Side Stories	NaN	Louis Cheung, Myolie Wu, Justin Cheung, Brian ...	Hong Kong	April 30, 2019	2018	TV-MA	1 Season	International TV Shows, TV Comedies, TV Dramas	The intimate lives of young men and women from...
4566	s4567	TV Show	OCTB	NaN	Jordan Chan, Justin Cheung, Kwok-Kwan Chan, Sa...	Hong Kong	February 5, 2018	2017	TV-14	1 Season	Crime TV Shows, International TV Shows, TV Dramas	An undercover detective crosses paths with fam...

Latest released

```
fig = go.Figure(data=[go.Table(header=dict(values=['Title', 'Release Year']),
                 cells=dict(values=[newest_HK_series['title'],newest_HK_series['release_year']]))
                     ])
fig.show()
```

Title	Release Year
A Home with A View	2019
Sexy Central	2019
We Are Legends	2019
Hong Kong West Side Stories	2018
OCTB	2017
Lady Bloodfight	2016
Mad World	2016
Drink Drank Drunk	2016
Weeds on Fire	2016
Ten Years	2015
Break Up 100	2014
May We Chat	2014
The Midas Touch	2013
SDU: Sex Duties Unit	2013
Don't Go Breaking My Heart	2011
Love In A Puff	2010

Chapter Wrap Up

- Data cleaning is not just fillna or dropna, sometimes we need to change its attributes to fit for numerical analysis.
- Data analysis could be fun and practical. Everything around you must be data back-ended.
- You may have some subjective hypothesis, but should conclude with factual statistics afterwards.

Reference & Resources

Official Website:

https://plotly.com/python/

Plotly Graph Objects:
https://plotly.com/python/graph-objects/

Kaggle dataset:
https://www.kaggle.com/datasets

WorldBank API:

- https://blogs.worldbank.org/opendata/introducing-wbgapi-new-python-package-accessing-world-bank-data
- https://nbviewer.org/github/tgherzog/wbgapi/blob/master/examples/wbgapi-cookbook.ipynb
- https://pypi.org/project/wbgapi/

GitHub Open Source Code:
https://github.com/plotly/plotly.py