The aim of this notebook to:

Importing Libraies--Data preprocessing--Handling missing data--Data visualization

1) Type content is available . 2) Top Five Rating Category. 3) Top Five Directors. 4) Top Five Actors. 5) Trend of focus on TV Shows and movies in recent years. 6) Top Ten countries with most content

Importing Libraries

In [1]:
```
!pip install cutecharts
```

```
Collecting cutecharts
  Downloading cutecharts-1.2.0-py3-none-any.whl (17 kB)
Requirement already satisfied: jinja2 in c:\users\user\anaconda3\lib\site-packages (from cutecharts) (2.11.3)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\user\anaconda3\lib\site-packages (from jinja2->cutecharts) (2.0.
1)
Installing collected packages: cutecharts
Successfully installed cutecharts-1.2.0
```

In [25]:
```
import pandas as pd
import numpy as np
import cutecharts.charts as ctc
from cutecharts.charts import Line
from cutecharts.faker import Faker
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

This dataset contains information about Netflix Movies and TV Shows.

In [26]:
```
data = pd.read_csv("netflix_titles.csv")
```

In [27]:
```
print('-' * 50)
print('\nSize of Netflix data is {}\n'.format(data.shape))
print('-' * 50)
data.head()
```

```
--------------------------------------------------

Size of Netflix data is (8807, 12)

--------------------------------------------------
```

Out[27]:

	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...

Data Preprocessing

In [28]:
```
print('-' * 50)
print("\nStatstical information about the given Data\n")
print('-' * 50)
data.describe()
```

```
--------------------------------------------------

Statstical information about the given Data

--------------------------------------------------
```

Out[28]:

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

Handling Missing Data

1)replace missing with 'No Director'
2)replace missing cast with 'No Cast'
3)replace missing countries with 'Not Specify'

In [29]:
```
data['director'].replace(np.nan, 'No Director',inplace=True)
data['cast'].replace(np.nan, 'No Cast',inplace=True)
data['country'].replace(np.nan, 'Not Specify',inplace=True)
data.isnull().sum()
```

Out[29]:
```
show_id          0
type             0
title            0
director         0
cast             0
country          0
date_added      10
release_year     0
rating           4
duration         3
listed_in        0
description      0
dtype: int64
```

In [30]:
```
#drop null value
data = data.dropna()
data.isnull().sum()
```

Out[30]:
```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
dtype: int64
```

In [31]:
```
print('-' * 50)
print("Check Duplicates")
print('-' * 50)
print('Total Duplicates values: ',data.duplicated().sum())
print('-' * 50)
```

```
--------------------------------------------------
Check Duplicates
--------------------------------------------------
Total Duplicates values:  0
--------------------------------------------------
```

Data Visualization

1) Type of the content available

In [32]:
```
data['type'].value_counts()
```

Out[32]:
```
Movie      6126
TV Show    2664
Name: type, dtype: int64
```

Pie chart of content

In [35]:
```
t_labels = data['type'].unique()
t_labels
```

Out[35]:
```
array(['Movie', 'TV Show'], dtype=object)
```

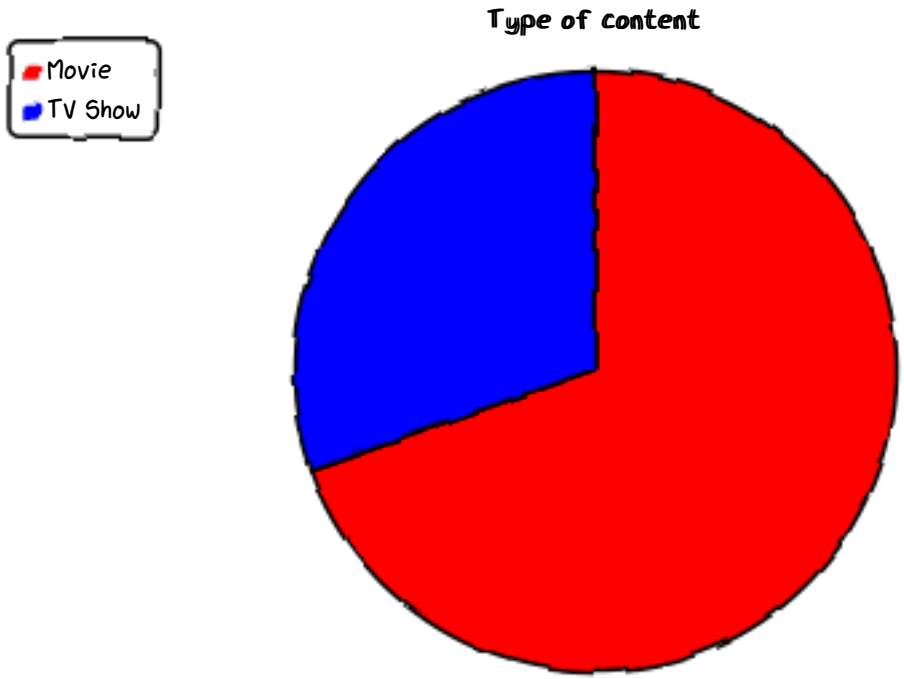
In [48]:
```
# pie chart
pie = ctc.Pie('Type of content', # title
              width='600px',height='300px')

# set the chart options
pie.set_options(labels=list(t_labels), # names as labels
                inner_radius=0,       # inner radius set to 0
                colors=['Red','blue'])

# label to be shown on graph
pie.add_series(list(t_values))

# display the charts
pie.render_notebook()
```

Out[48]:



2) Top Five Rating Category

In [77]:
```
newdata = data.groupby('rating').size().rename_axis('Rating').reset_index(name='Count')
nd = newdata.sort_values(by ='Count', ascending=True)
nd = nd.tail(5)
```

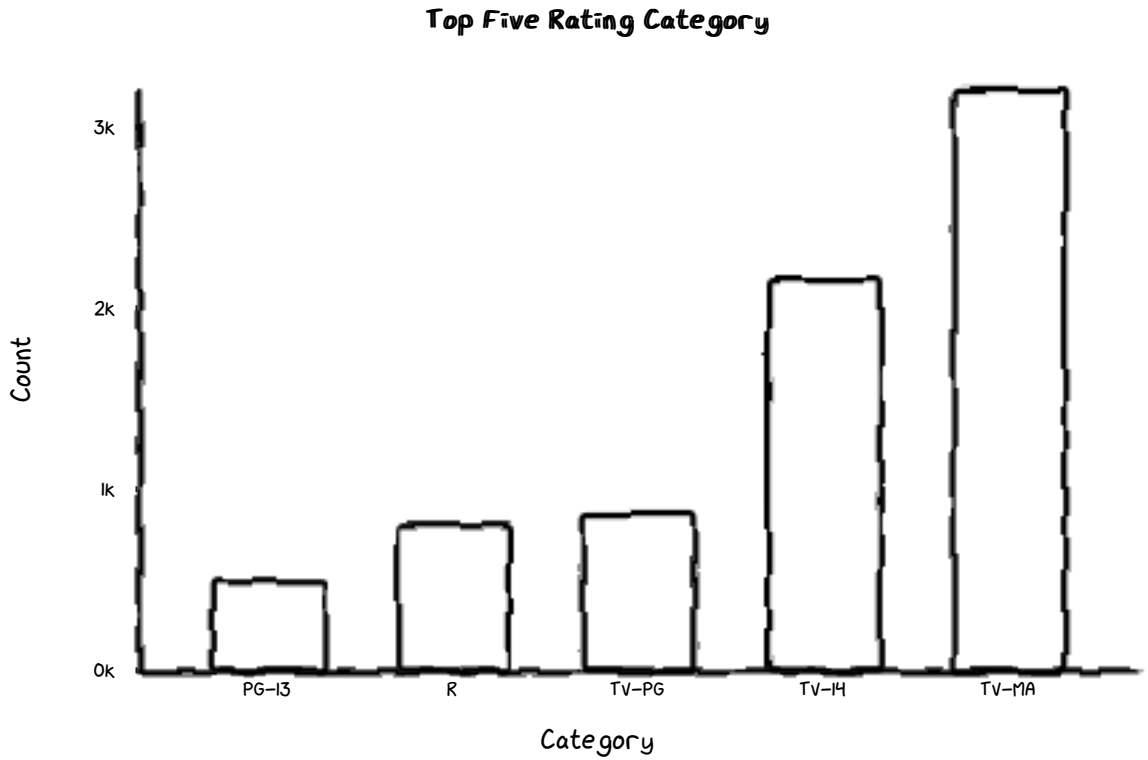
In [78]:
```
chart = ctc.Bar('Top Five Rating Category', width='600px', height='300px')

chart.set_options(labels=list(nd.Rating), x_label='Category', y_label='Count', colors=Faker.colors)

chart.add_series('Geners',list(nd['Count']))

chart.render_notebook()
```

Out[78]:



3) Top Five Directors

In [79]:
```
fil_directors = data['director'].str.split(',',expand=True).stack()
fil_directors= pd.DataFrame(fil_directors)
fil_directors.columns = ['director']
directors = fil_directors.groupby(['director']).size().reset_index(name='counts')
directors = directors.sort_values(by='counts',ascending=False)
directors = directors[directors['director'] != 'No Director']
directors = directors.head(5)
directors
```

Out[79]:

	director	counts
4019	Rajiv Chilaka	22
4066	Raúl Campos	18
261	Jan Suter	18
4650	Suhas Kadav	16
3233	Marcus Raboy	16

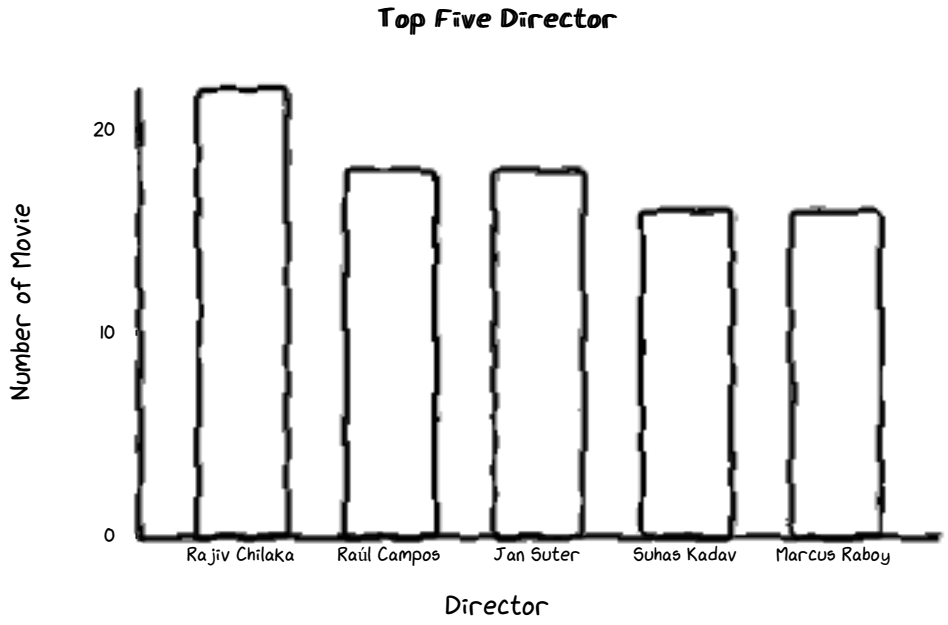
In [81]:
```
chart = ctc.Bar('Top Five Director', width='500px', height='100px')

chart.set_options(labels=list(directors.director),x_label='Director',y_label='Number of Movie', colors=Faker.colors)

chart.add_series('Geners',list(directors.counts))

chart.render_notebook()
```

Out[81]:



4) Top Five Actors

In [82]:
```
fil_actors = data['cast'].str.split(',',expand=True).stack()
fil_actors= pd.DataFrame(fil_actors)
fil_actors.columns = ['cast']
actors = fil_actors.groupby(['cast']).size().reset_index(name='counts')
actors = actors.sort_values(by='counts',ascending=False)
actors = actors[actors['cast'] != 'No Cast']
actors = actors.head(5)
actors
```

Out[82]:

	cast	counts
2605	Anupam Kher	39
26903	Rupa Bhimani	31
30263	Takahiro Sakurai	30
15518	Julie Tejwani	28
23591	Om Puri	27

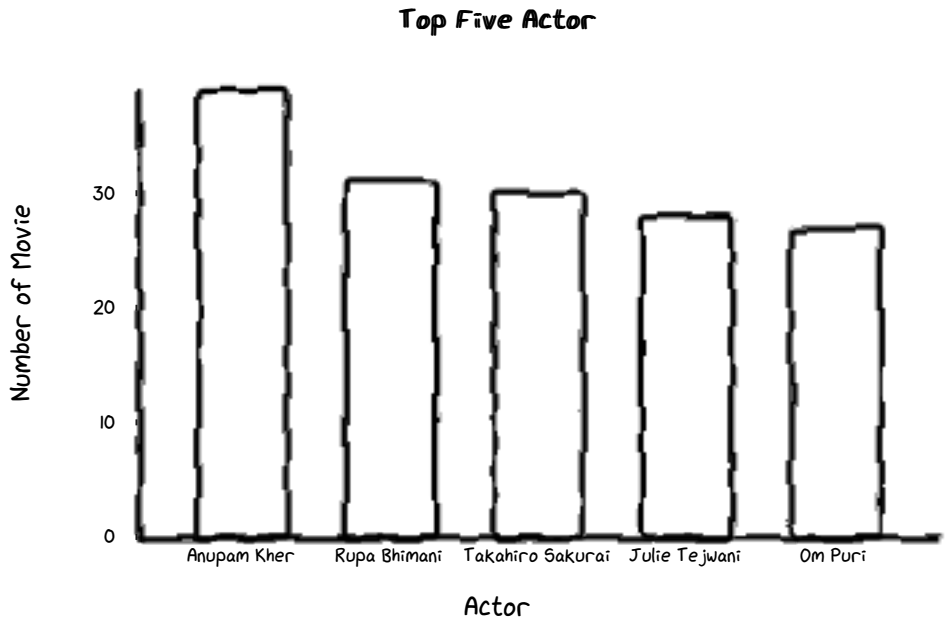
In [88]:
```
chart = ctc.Bar('Top Five Actor', width='500px', height='100px')

chart.set_options(labels=list(actors.cast),x_label='Actor',y_label='Number of Movie', colors=Faker.colors)

chart.add_series('Geners',list(actors.counts))

chart.render_notebook()
```

Out[88]:

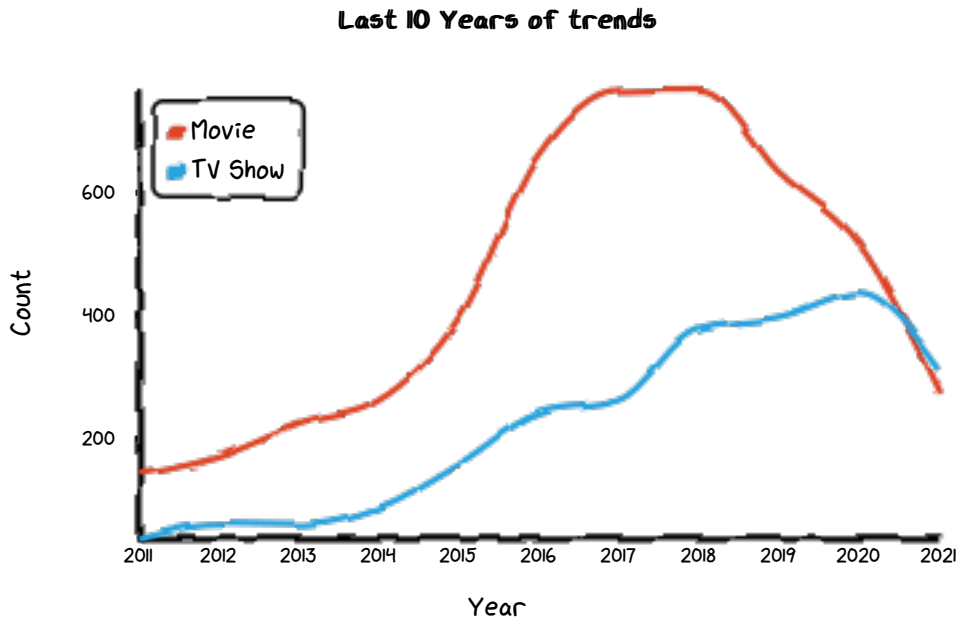


5) Trend of focus on TV Shows and movies in recent years.

In [89]:
```
dff = data[['type','release_year']]
dff = dff.rename(columns = {'release_year' : 'Release Year'})
dff2 = dff.groupby(['Release Year','type']).size().reset_index(name='Total Content')
dff2 = dff2[dff2['Release Year']>=2011]
dff3 = dff2[dff2['type']=='Movie']
dff4 = dff2[dff2['type']=='TV Show']
```

In [104…
```
chart = Line('Last 10 Years of trends', width='500px', height='100px')
chart.set_options(labels=list(dff3['Release Year']), x_label='Year', y_label='Count',)
chart.add_series('Movie', list(dff3['Total Content']))
chart.add_series('TV Show', list(dff4['Total Content']))
chart.render_notebook()
```

Out[104]:



6) Top Ten countries with most content

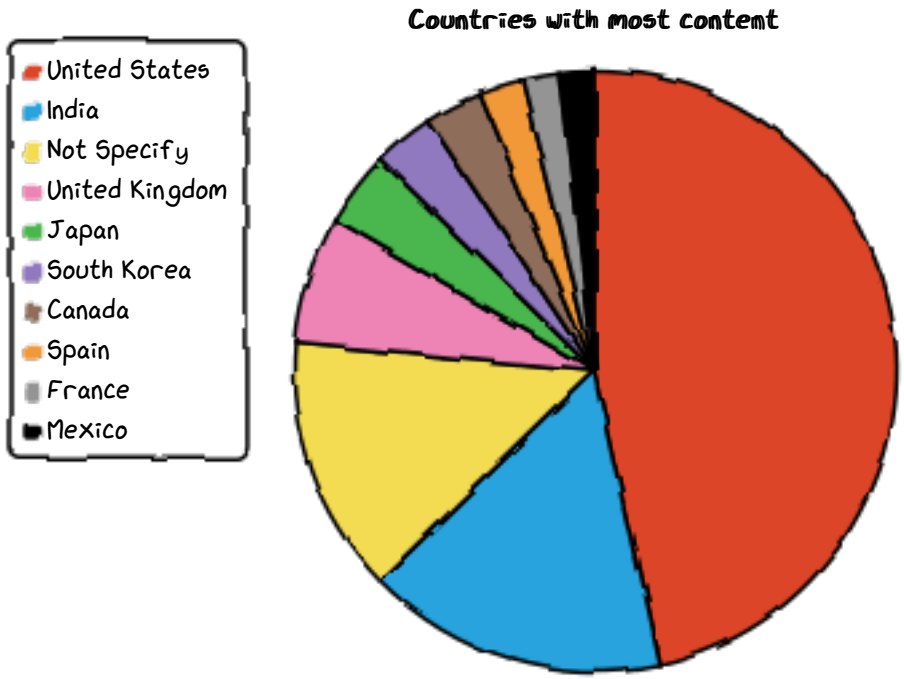
In [94]:
```
top_countries=data['country'].value_counts()[:10].to_frame(name='count')
top_countries
```

Out[94]:

	count
United States	2809
India	972
Not Specify	829
United Kingdom	418
Japan	243
South Korea	199
Canada	181
Spain	145
France	124
Mexico	110

In [96]:
```
pie =ctc.Pie('Countries with most content',width='600px',height='300px')
pie.set_options(labels=list(top_countries.index),inner_radius=0)
pie.add_series(list(top_countries['count']))
pie.render_notebook()
```

Out[96]:


Countries with most content
United States
India
Not Specify
United Kingdom
Japan
South Korea
Canada
Spain
France
Mexico

In [ ]: