```
#load all necessary libraries and data
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')  #to ignore warnings
```

```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
To: /content/netflix.csv
100% 3.40M/3.40M [00:00<00:00, 17.7MB/s]
```

```
data=pd.read_csv('netflix.csv')   #read csv file
data.head()   # first 5 rows to check or inspect data
```

	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...

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
data.info()  #tells about data type and non-null values in each column/field
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

```
data.describe()  #tells basic stats of about numeric columns
```

	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

```
data.shape  #tells no of rows and columns
```

```
(8807, 12)
```

```
data.isnull().sum()/len(data) *100#check percentage of missing values in each attribute
```

	0
show_id	0.000000
type	0.000000
title	0.000000
director	29.908028
cast	9.367549
country	9.435676
date_added	0.113546
release_year	0.000000
rating	0.045418
duration	0.034064
listed_in	0.000000
description	0.000000

dtype: float64

As per above data it shows there are 29% of missing/null values are in director column

```
data.head()
```

	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...

Next steps: Generate code with data | View recommended plots | New interactive sheet

Data already has default serial no so no use of show_id

```
data.drop('show_id',axis=1,inplace=True) #remove show_id column from data
```

```
data.head()
```

	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	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	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	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...

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
data.nunique()  #check unique values in each attribute/column
```

	0
type	2
title	8807
director	4528
cast	7692
country	748
date_added	1767
release_year	74
rating	17
duration	220
listed_in	514
description	8775

dtype: int64

```
data.duplicated().sum() #duplicate values in data
```

```
0
```

No duplicate value in data

Preprocessing of Data

```
data['date_added']=pd.to_datetime(data['date_added'],format='mixed')  #convert to datetime dtype
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   type          8807 non-null   object
 1   title         8807 non-null   object
 2   director      6173 non-null   object
 3   cast          7982 non-null   object
 4   country       7976 non-null   object
 5   date_added    8797 non-null   datetime64[ns]
 6   release_year  8807 non-null   int64
 7   rating        8803 non-null   object
 8   duration      8804 non-null   object
 9   listed_in     8807 non-null   object
 10  description   8807 non-null   object
dtypes: datetime64[ns](1), int64(1), object(9)
memory usage: 757.0+ KB
```

Handling missing values

```
data[['director','cast','country']]=data[['director','cast','country']].fillna('Unknown')
data['duration']=data['duration'].fillna(0)
rating_mode=data['rating'].mode()[0]
date_added_mode=data['date_added'].mode()[0]
data['rating'].fillna(rating_mode,inplace=True)
data['date_added'].fillna(date_added_mode,inplace=True)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   type          8807 non-null   object
 1   title         8807 non-null   object
 2   director      8807 non-null   object
 3   cast          8807 non-null   object
 4   country       8807 non-null   object
 5   date_added    8807 non-null   datetime64[ns]
 6   release_year  8807 non-null   int64
 7   rating        8807 non-null   object
 8   duration      8807 non-null   object
 9   listed_in     8807 non-null   object
 10  description   8807 non-null   object
dtypes: datetime64[ns](1), int64(1), object(9)
memory usage: 757.0+ KB
```

```
data.isnull().sum()
```

	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

there is no missing value in data after this

```
data['rating'].unique()
```

```
array(['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', 'TV-Y', 'TV-Y7', 'R',
       'TV-G', 'G', 'NC-17', '74 min', '84 min', '66 min', 'NR',
       'TV-Y7-FV', 'UR'], dtype=object)
```

In rating column ['74 min','84 min','66 min'] these are wrong ratings it seems like duration

```
wrong_ratings=['74 min','84 min','66 min']
data.loc[data['rating'].isin(wrong_ratings)]
```

	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
5541	Movie	Louis C.K. 2017	Louis C.K.	Louis C.K.	United States	2017-04-04	2017	74 min	0	Movies	Louis C.K. muses on religion, eternal love, gi...
5794	Movie	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	United States	2016-09-16	2010	84 min	0	Movies	Emmy-winning comedy writer Louis C.K. brings h...
		Louis C.K.:									The comic puts his

duration has null values which has wrong ratings

```
mask=data['rating'].isin(wrong_ratings)
data.loc[mask,'duration']=data.loc[mask,'rating']   #replace values of duration which has wrong ratings
data.loc[data['rating'].isin(wrong_ratings),'rating']=data['rating'].mode()[0]  #replace wrong ratings with NR(No Rating)
data['rating'].unique()
```

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

```
#Replace ratings with actual meaning of these ratings

new_rating = {'TV-MA':'Adults',
              'R':'Adults',
              'PG-13':'Teens',
              'TV-14':'Young Adults',
              'TV-PG':'Older Kids',
              'NR':'Adults',
              'TV-G':'Kids',
              'TV-Y':'Kids',
              'TV-Y7':'Older Kids',
              'PG':'Older Kids',
              'G':'Kids',
              'NC-17':'Adults',
              'TV-Y7-FV':'Older Kids',
              'UR':'Adults'}

data['rating'].replace(new_rating, inplace = True)
data['rating'].unique()
```

```
array(['Teens', 'Adults', 'Older Kids', 'Young Adults', 'Kids'],
      dtype=object)
```

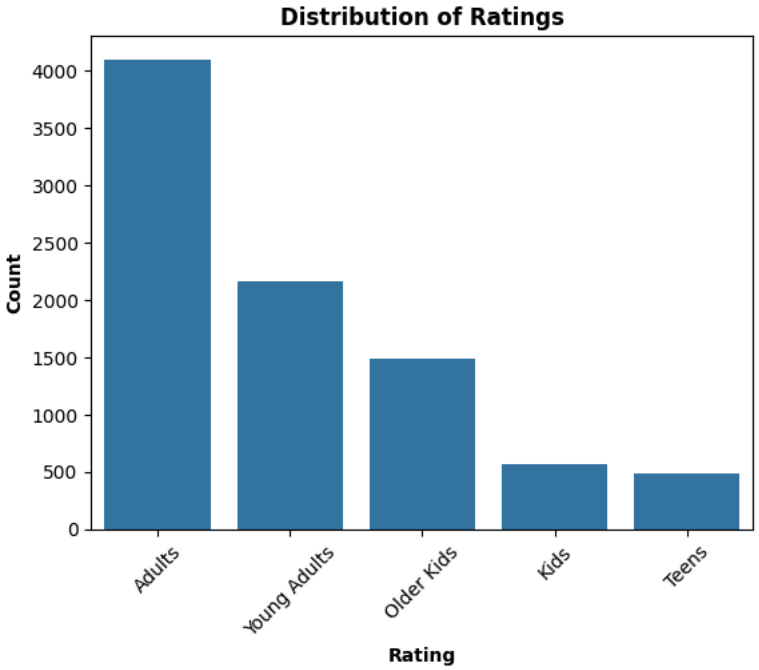
Which type of rating content has netflix most?

```
data['rating'].value_counts()
```

rating	count
Adults	4099
Young Adults	2160
Older Kids	1490
Kids	568
Teens	490

dtype: int64

```
sns.barplot(x='rating',y='count',data=data['rating'].value_counts().reset_index())
plt.xticks(rotation=45)
plt.xlabel('Rating',fontweight='bold')
plt.ylabel('Count',fontweight='bold')
plt.title('Distribution of Ratings', fontweight='bold')
plt.show()
```

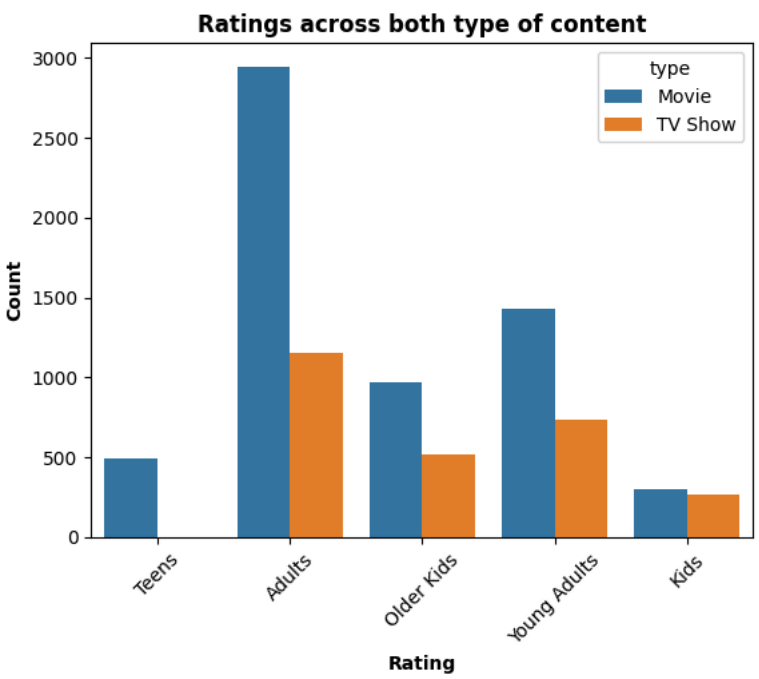


Netflix has more Adult content followed by Young Adult content

Netflix has least content for Teens

Compare Ratings for both type of Content

```
sns.countplot(x='rating',hue='type',data=data)
plt.xticks(rotation=45)
plt.xlabel('Rating',fontweight='bold')
plt.ylabel('Count',fontweight='bold')
plt.title('Ratings across both type of content',fontweight='bold')
plt.show()
```



For Teenagers Netflix has least no of content

Netflix has more content for adult audience

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

```
data['year_added']=data['date_added'].dt.year  #create new column in which year content added to netflix
data['year_added']
```

	year_added
0	2021
1	2021
2	2021
3	2021
4	2021
...	...
8802	2019
8803	2019
8804	2019
8805	2020
8806	2019

8807 rows × 1 columns

dtype: int32

```
movies=data[data['type']=='Movie']
tv_shows=data[data['type']=='TV Show']
```

```
movies_per_year=movies['release_year'].value_counts().reset_index()
tv_shows_per_year=tv_shows['release_year'].value_counts().reset_index()
movies_per_year
```

	release_year	count
0	2017	767
1	2018	767
2	2016	658
3	2019	633
4	2020	517
...	...	...
68	1966	1
69	1961	1
70	1946	1
71	1963	1
72	1947	1

73 rows × 2 columns

Next steps: Generate code with movies_per_year | View recommended plots | New interactive sheet

In 2017 most movies are released

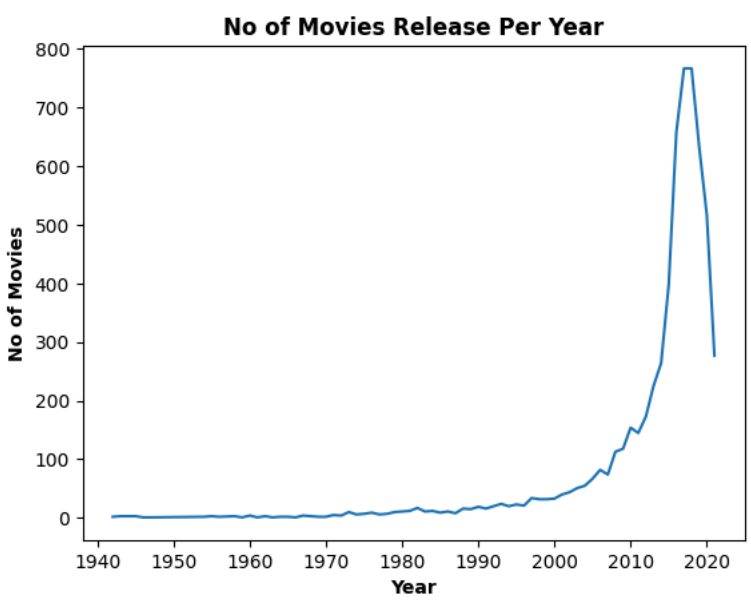
```
tv_shows_per_year
```

	release_year	count
0	2020	436
1	2019	397
2	2018	380
3	2021	315
4	2017	265
5	2016	244
6	2015	162
7	2014	88
8	2012	64
9	2013	63
10	2010	40
11	2011	40
12	2009	34
13	2008	23
14	2006	14
15	2007	14
16	2005	13
17	2003	10
18	2004	9
19	1999	7
20	2002	7
21	2001	5
22	1993	4
23	2000	4
24	1997	4
25	1998	4
26	1990	3
27	1996	3
28	1992	3
29	1995	2
30	1994	2
31	1988	2
32	1986	2
33	1989	1
34	1967	1
35	1985	1
36	1946	1
37	1981	1
38	1972	1
39	1979	1
40	1977	1
41	1991	1
42	1974	1
43	1925	1
44	1945	1
45	1963	1

Next steps: Generate code with tv_shows_per_year | View recommended plots | New interactive sheet

In 2020 most TV Shows are released

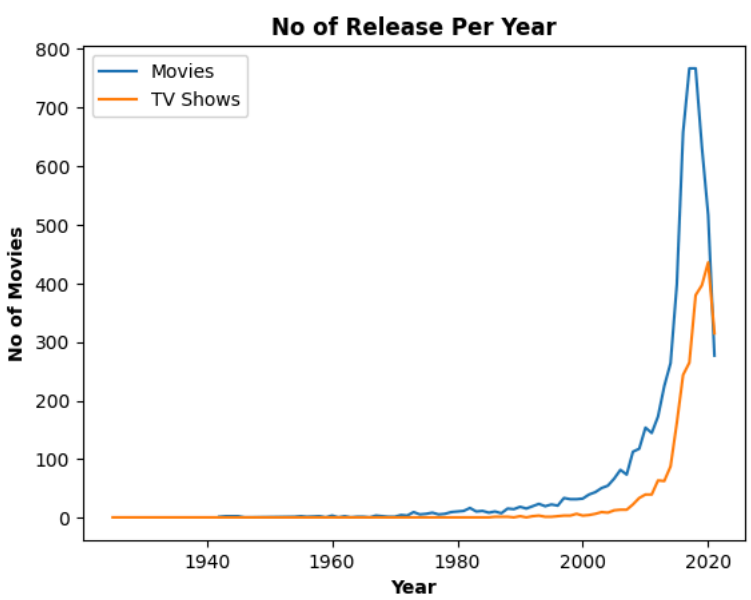
```
sns.lineplot(x='release_year',y='count',data=movies_per_year)
plt.xlabel('Year',fontweight='bold')
plt.ylabel('No of Movies',fontweight='bold')
plt.title('No of Movies Release Per Year',fontweight='bold')
plt.show()
```



From year 2000 there is gradually increase in movie release and in year 2017 and 2018 most no of movies released followed by 2016 and 2019 after that there is decline in movie release

Comparison of Movies and TV Shows Release Year

```
movies_per_year['release_year']=pd.to_numeric(movies_per_year['release_year'],errors='coerce')
tv_shows_per_year['release_year']=pd.to_numeric(tv_shows_per_year['release_year'],errors='coerce')
sns.lineplot(x='release_year',y='count',data=movies_per_year,label='Movies')
sns.lineplot(x='release_year',y='count',data=tv_shows_per_year,label='TV Shows')
plt.xlabel('Year',fontweight='bold')
plt.ylabel('No of Movies',fontweight='bold')
plt.title('No of Release Per Year',fontweight='bold')
plt.legend()
plt.show()
```



Netflix has most no of movies than tv shows

For TV Shows:

- Gradually there is increase in relase from 2006 onwards
- In year 2020 most no of TV Shows are released after that there is some decline same in movies also.

```
data.head()
```

	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	year_added
0	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unknown	United States	2021-09-25	2020	Teens	90 min	Documentaries	As her father nears the end of his life, filmm...	2021
1	TV Show	Blood & Water	Unknown	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	Adults	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	2021
2	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Unknown	2021-09-24	2021	Adults	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	2021
											Feuds,	

Next steps: Generate code with data | View recommended plots | New interactive sheet

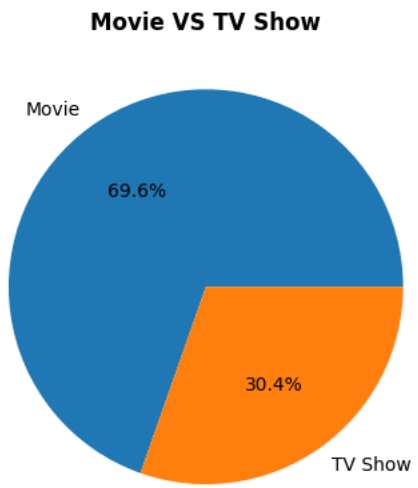
Comparison of Movies VS TV Shows

```
data['type'].value_counts()
```

type	count
Movie	6131
TV Show	2676

dtype: int64

```
plt.pie(data['type'].value_counts(),labels=data['type'].value_counts().index,autopct='%1.1f%%')
plt.title('Movie VS TV Show',fontweight='bold')
plt.show()
```



Netflix has more movies content than tv shows

Top 10 Countries

```
data['country'].value_counts()
```

country	count
United States	2818
India	972
Unknown	831
United Kingdom	419
Japan	245
...	...
Romania, Bulgaria, Hungary	1
Uruguay, Guatemala	1
France, Senegal, Belgium	1
Mexico, United States, Spain, Colombia	1
United Arab Emirates, Jordan	1

749 rows × 1 columns

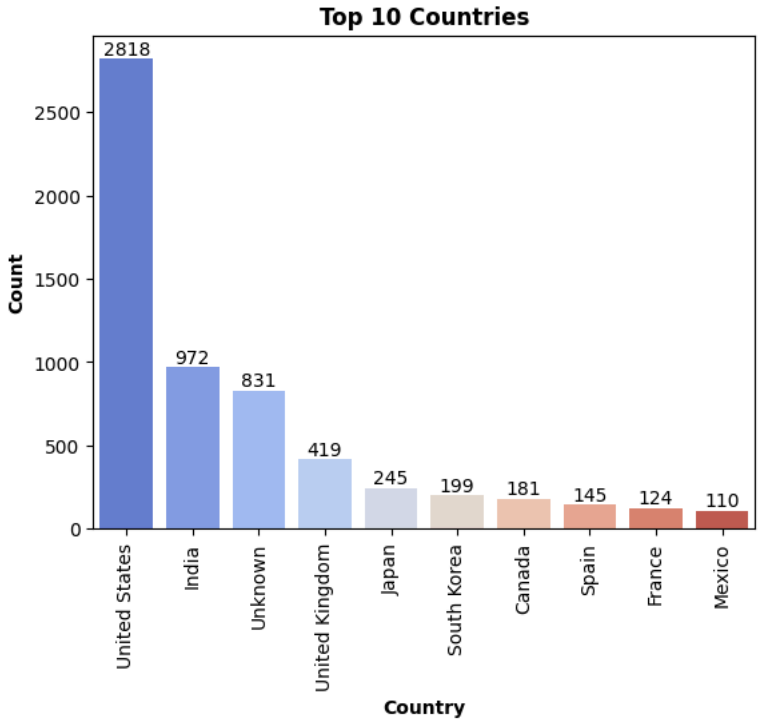
dtype: int64

```
top10_country=data['country'].value_counts().head(10)
top10_country
```

country	count
United States	2818
India	972
Unknown	831
United Kingdom	419
Japan	245
South Korea	199
Canada	181
Spain	145
France	124
Mexico	110

dtype: int64

```
ax=sns.barplot(x=top10_country.index,y=top10_country.values,palette='coolwarm')
plt.xlabel('Country',fontweight='bold')
plt.ylabel('Count',fontweight='bold')
plt.xticks(rotation=90)
plt.title('Top 10 Countries',fontweight='bold')
for container in ax.containers:
  ax.bar_label(container)
plt.show()
```



Netflix has more United States Content followed by India

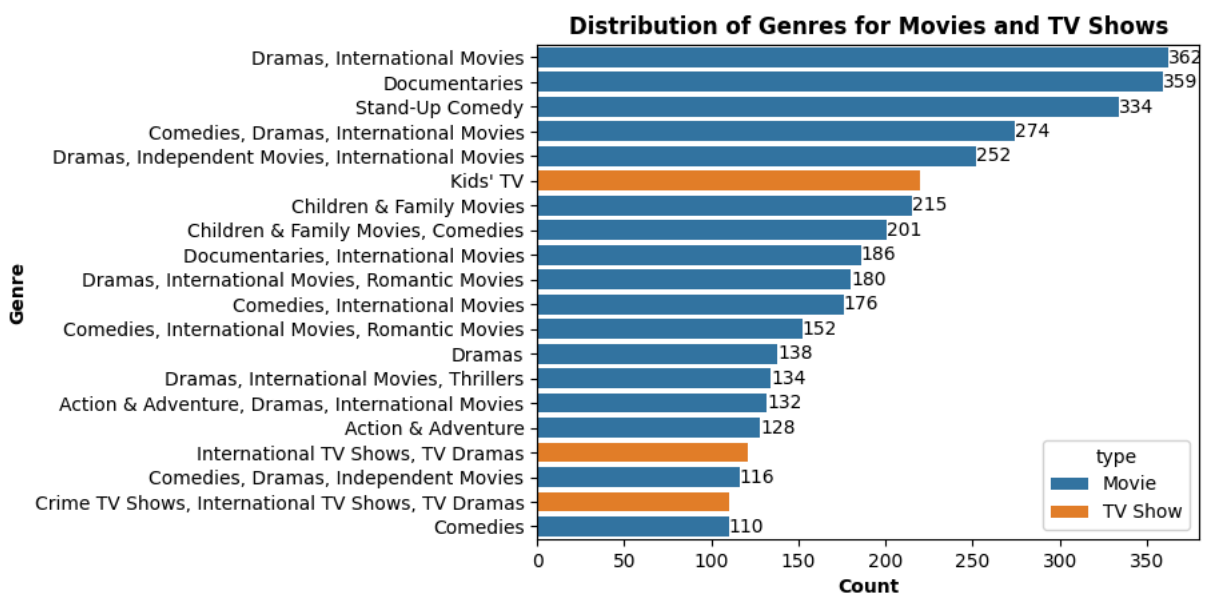
Top 10 Genre

```
top10_genre=data['listed_in'].value_counts().head(10)
top10_genre
```

listed_in	count
Dramas, International Movies	362
Documentaries	359
Stand-Up Comedy	334
Comedies, Dramas, International Movies	274
Dramas, Independent Movies, International Movies	252
Kids' TV	220
Children & Family Movies	215
Children & Family Movies, Comedies	201
Documentaries, International Movies	186
Dramas, International Movies, Romantic Movies	180

dtype: int64

```
ax=sns.countplot(y='listed_in',data=data,order=data['listed_in'].value_counts().index[:20],hue='type')
plt.title('Distribution of Genres for Movies and TV Shows',fontweight='bold')
plt.xlabel('Count',fontweight='bold')
plt.ylabel('Genre',fontweight='bold')
ax.bar_label(ax.containers[0])
plt.show()
```



As per above graph it shows Netflix has more focus on movies

- In top 10 Genres there is only one genre of tv shows
- In top 20 Genres there is only three genre of tv shows

Top 10 Directors

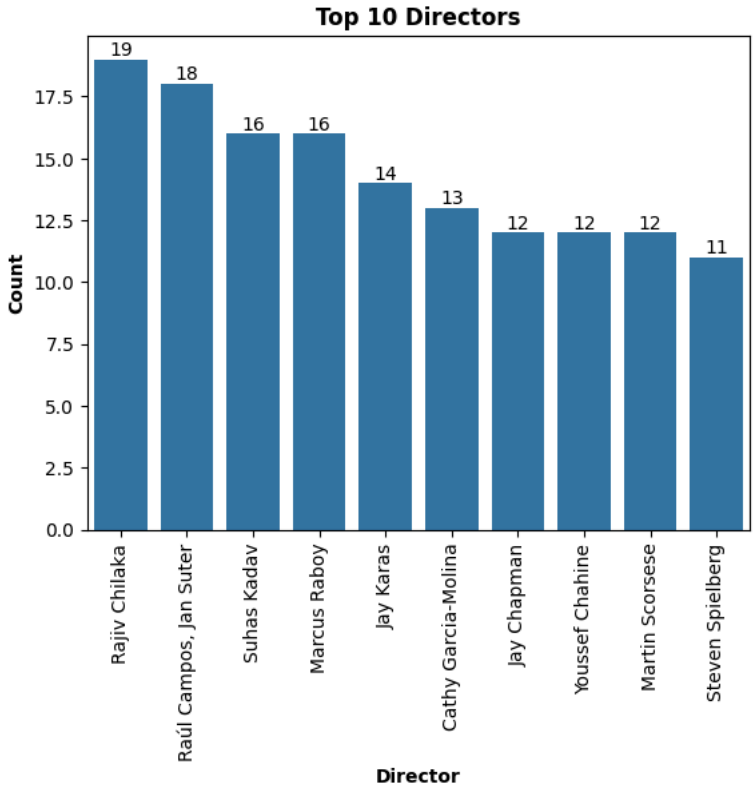
```
dir=data['director'].explode().value_counts()
dir=dir[~(dir.index=='Unknown')]
dir
```

director	count
Rajiv Chilaka	19
Raúl Campos, Jan Suter	18
Suhas Kadav	16
Marcus Raboy	16
Jay Karas	14
...	...
Raymie Muzquiz, Stu Livingston	1
Joe Menendez	1
Eric Bross	1
Will Eisenberg	1
Mozez Singh	1

4528 rows × 1 columns

dtype: int64

```
ax=sns.barplot(x=dir.index[:10],y=dir.values[:10])
plt.xticks(rotation=90)
plt.xlabel('Director',fontweight='bold')
plt.ylabel('Count',fontweight='bold')
plt.title('Top 10 Directors',fontweight='bold')
ax.bar_label(ax.containers[0])
plt.show()
```



Most Content(**19**) has been directed by Rajiv Chilaka followed by Raul Campus and Jan Suter

```
tvshow_dir=data[~(data['director']=='Unknown') & (data['type']=='TV Show')].director.value_counts()
tvshow_dir
```

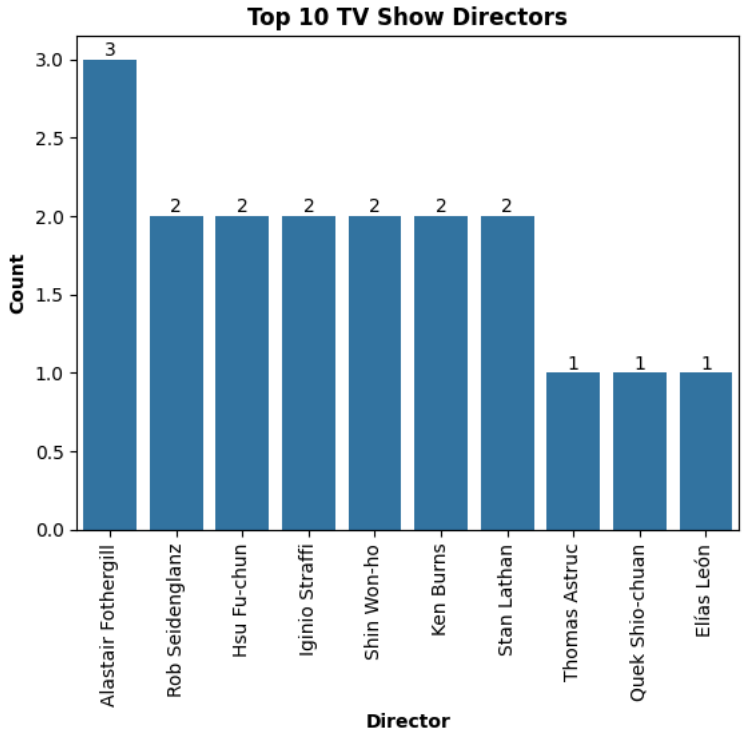
director	count
Alastair Fothergill	3
Rob Seidenglanz	2
Hsu Fu-chun	2
Iginio Straffi	2
Shin Won-ho	2
...	...
Juliana Vicente	1
Chang Chin-jung, Chen Rong-hui	1
Thierry Demaizière, Alban Teurlai	1
Manolo Caro	1
Michael Cumming	1

222 rows × 1 columns

dtype: int64

Top 10 TV Show Director

```
ax=sns.barplot(x=tvshow_dir.index[:10],y=tvshow_dir.values[:10])
plt.xticks(rotation=90)
ax.bar_label(ax.containers[0])
plt.xlabel('Director',fontweight='bold')
plt.ylabel('Count',fontweight='bold')
plt.title('Top 10 TV Show Directors',fontweight='bold')
plt.show()
```



Alastair Fothergill has directed most tv shows

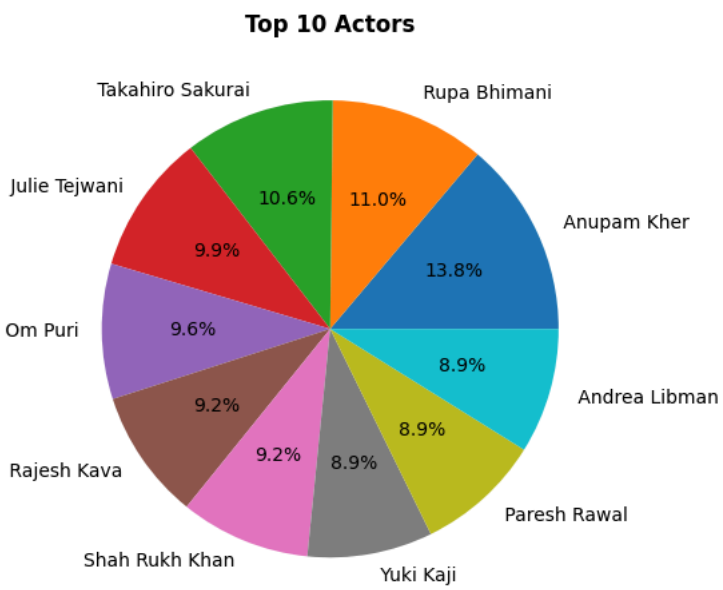
```
cast_act=data['cast'].apply(lambda x : x.split(','))
cast_act = cast_act[cast_act.apply(lambda x: x != ['Unknown'])]
cast_act=cast_act.explode()
cast_act=cast_act.value_counts()
cast_act
```

cast	count
Anupam Kher	39
Rupa Bhimani	31
Takahiro Sakurai	30
Julie Tejwani	28
Om Puri	27
...	...
Vedika	1
Tedros Teclebrhan	1
Maryam Zaree	1
Melanie Straub	1
Chittaranjan Tripathy	1

39296 rows × 1 columns

dtype: int64

```
top10_actors=cast_act[:10]
plt.pie(top10_actors,labels=cast_act.index[:10],autopct='%1.1f%%')
plt.title('Top 10 Actors',fontweight='bold')
plt.tight_layout()
plt.show()
```



Anupam Kher has appeared as actor in most no of content followed by Rupa Bhamani and Takahiro Sakurai

What duration has highest no of movies

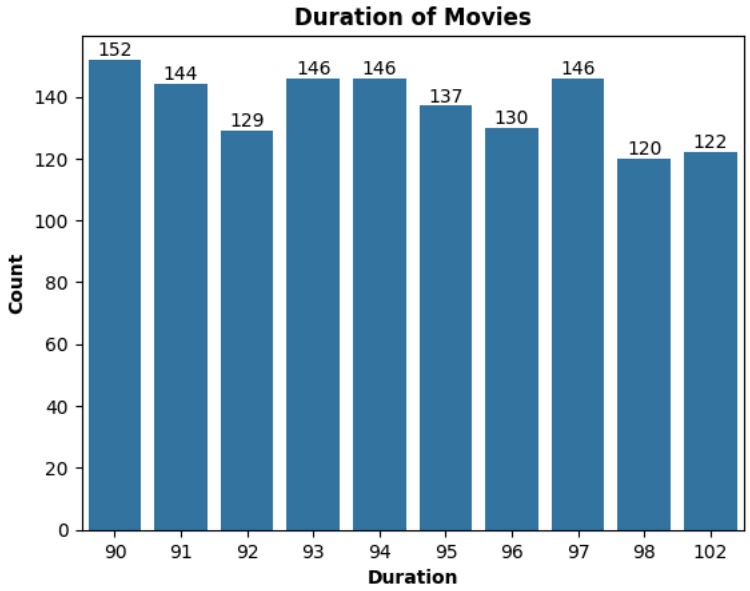
```
dur_movies=data[data['type'] == 'Movie'].duration.str.replace(' min', '').astype(int).value_counts()
dur_movies
```

duration	count
90	152
94	146
97	146
93	146
91	144
...	...
208	1
5	1
16	1
186	1
191	1

205 rows × 1 columns

dtype: int64

```
ax=sns.barplot(x=dur_movies.index[:10],y=dur_movies.values[:10])
plt.xlabel('Duration',fontweight='bold')
plt.ylabel('Count',fontweight='bold')
plt.title('Duration of Movies',fontweight='bold')
ax.bar_label(ax.containers[0])
plt.show()
```



Most no of movies has duration 90 min followed by 93,94 and 97 min

```
data[data['type'] == 'Movie'].duration.str.replace(' min', '').astype(int).mean()
```

```
99.56499755341706
```

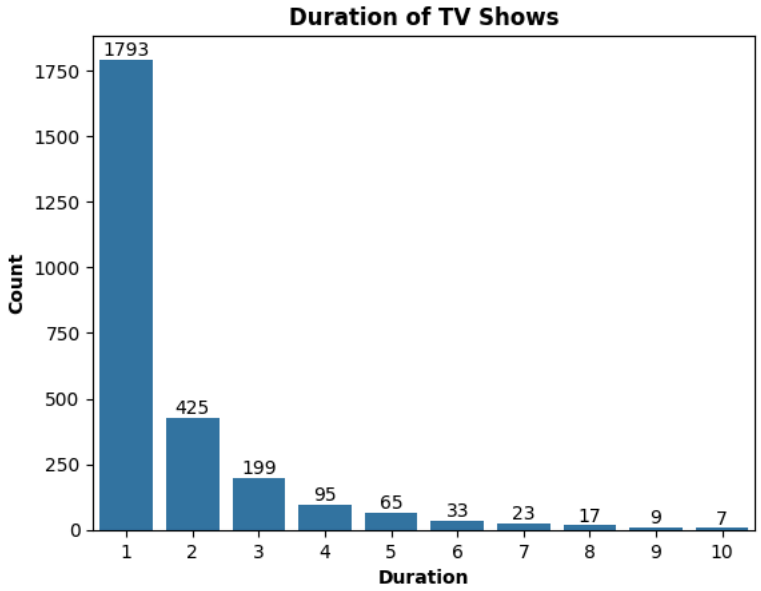
Average duration of movies is 99 min

```
dur_tvshow = data[data['type'] == 'TV Show'].duration.str.extract('(\d+)').astype(int).value_counts()
dur_tvshow
```

	count
0	
1	1793
2	425
3	199
4	95
5	65
6	33
7	23
8	17
9	9
10	7
13	3
11	2
12	2
15	2
17	1

dtype: int64

```
dur_tvshow=dur_tvshow.reset_index()
dur_tvshow.columns=['duration','count']
ax=sns.barplot(x='duration',y='count',data=dur_tvshow[:10])
plt.xlabel('Duration',fontweight='bold')
plt.ylabel('Count',fontweight='bold')
plt.title('Duration of TV Shows',fontweight='bold')
ax.bar_label(ax.containers[0])
plt.show()
```




```
data[data['type'] == 'TV Show'].duration.str.extract('(\d+)').astype(int).mean()
```

	0
0	1.764948

dtype: float64

Average duration of TV Show is 1.76 ~ 2 seasons

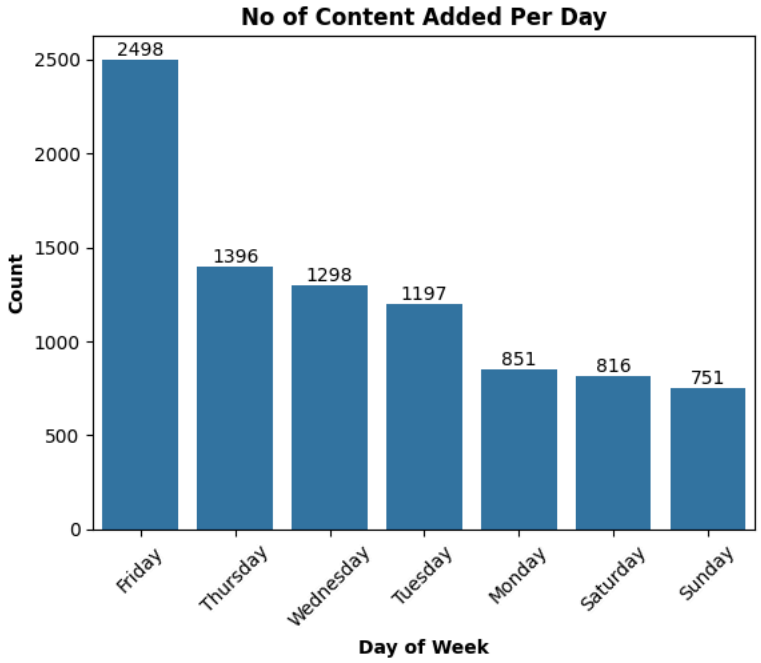
In which day highest no of content added

```
data['day_of_week']=data['date_added'].dt.day_name()
data['day_of_week'].value_counts()
```

day_of_week	count
Friday	2498
Thursday	1396
Wednesday	1298
Tuesday	1197
Monday	851
Saturday	816
Sunday	751

dtype: int64

```
ax=sns.barplot(x=data['day_of_week'].value_counts().index,y=data['day_of_week'].value_counts().values)
plt.xlabel('Day of Week',fontweight='bold')
plt.ylabel('Count',fontweight='bold')
plt.xticks(rotation=45)
ax.bar_label(ax.containers[0])
plt.title('No of Content Added Per Day',fontweight='bold')
plt.show()
```



Most content has been added on Friday

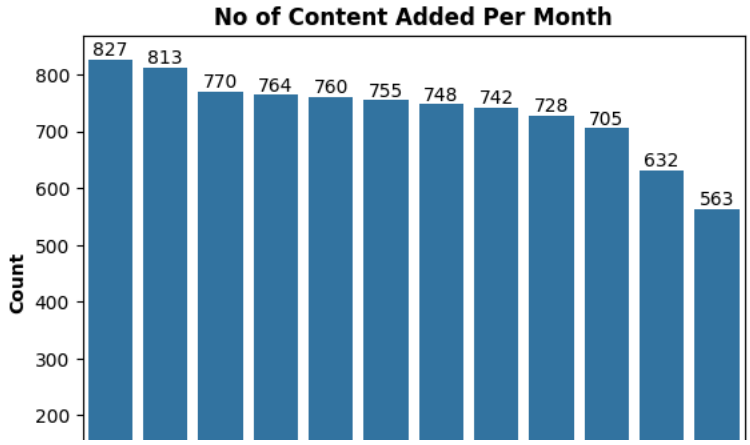
In which month most content added on Netflix

```
data['month_added']=data['date_added'].dt.month_name()
data['month_added'].value_counts()
```

month_added	count
July	827
December	813
September	770
April	764
October	760
August	755
January	748
March	742
June	728
November	705
May	632
February	563

dtype: int64

```
ax=sns.barplot(x=data['month_added'].value_counts().index,y=data['month_added'].value_counts().values)
plt.xlabel('Month',fontweight='bold')
plt.ylabel('Count',fontweight='bold')
plt.xticks(rotation=45)
ax.bar_label(ax.containers[0])
plt.title('No of Content Added Per Month',fontweight='bold')
plt.show()
```

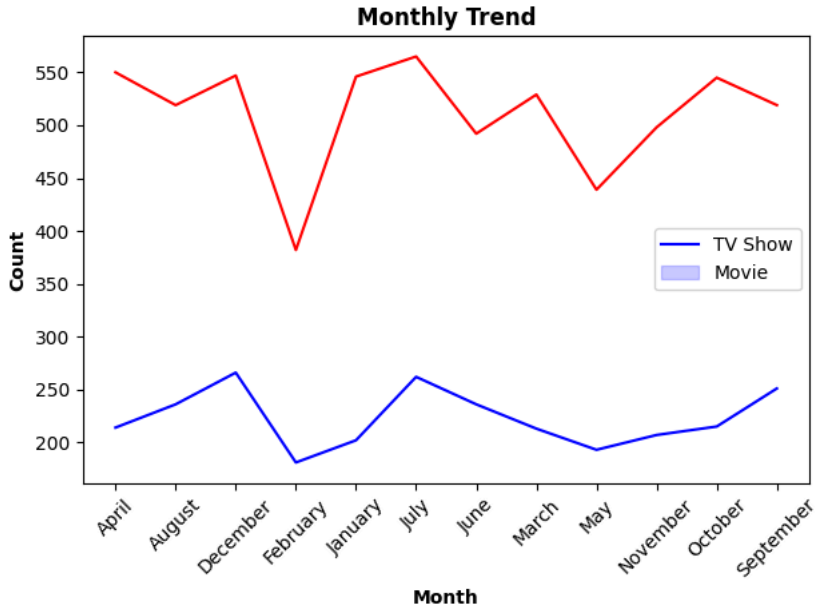


Most content added on Netflix in the month of July followed by December

Monthly Trend

```
month_movie_count=data[data['type']=='Movie'].groupby('month_added').size()
month_tvshow_count=data[data['type']=='TV Show'].groupby('month_added').size()


sns.lineplot(x=month_tvshow_count.index,y=month_tvshow_count.values,color='blue')
sns.lineplot(x=month_movie_count.index,y=month_movie_count.values,color='red')
plt.xlabel('Month',fontweight='bold')
plt.ylabel('Count',fontweight='bold')
plt.title('Monthly Trend',fontweight='bold')
plt.xticks(rotation=45)
plt.legend(['TV Show','Movie'])
plt.tight_layout()
plt.show()
```



Correlation between release year nad year_added

```
corr_data=data[['release_year','year_added']].corr()
corr_data
```

	release_year	year_added
release_year	1.000000	0.111206
year_added	0.111206	1.000000

Next steps: Generate code with corr_data View recommended plots New interactive sheet