In [353…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [76]:
```
plt.figure(facecolor='black', figsize=(18, 10))
plt.text(0.5, 0.75, 'NETFLIX',fontsize=120, fontweight='bold',ha='center', va='cent
plt.text(0.5, 0.45, 'DATA ANALYSIS PROJECT',fontsize=45,ha='center', va='center',co
plt.text(0.5, 0.25, 'by AKHILESH JAIN',fontsize=28,ha='center',color='white',fontfa
plt.axis('off')
plt.tight_layout()
plt.show()
```

DATA COLLECTION AND EDA OF THE NETFLIX DATA

In [355…
```
data=pd.read_csv(r"C:\Users\vanak\OneDrive\ドキュメント\DATASETS\netflix_titles.csv"
```

In [72]:
```
data.head()
```

Out[72]:

	show_id	type	title	director	cast	country	date_added	release_year	ra
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	unknown	United States	September 25, 2021	2020	P(
1	s2	TV Show	Blood & Water	unknown	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	unknown	September 24, 2021	2021	
3	s4	TV Show	Jailbirds New Orleans	unknown	unknown	unknown	September 24, 2021	2021	
4	s5	TV Show	Kota Factory	unknown	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	

In [70]:
```
data.shape
```

Out[70]:
```
(8807, 12)
```

In [74]:
```
data.dtypes
```

Out[74]:
```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

In [13]:
```
data.info()
```

```
<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
```

In [17]:
```
data.isnull().sum()
```

Out[17]:
```
show_id           0
type              0
title             0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

In [38]:
```
data['director']=data['director'].fillna("unknown")
data['cast']=data['cast'].fillna('unknown')
data['country']=data['country'].fillna('unknown')
data['date_added']=data['date_added'].fillna('unknown')
```

```
data['rating']=data['rating'].fillna('unknown')
data['duration']=data['duration'].fillna('unknown')
```

In [40]:
```
data.isnull().sum()
```

Out[40]:
```
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
```

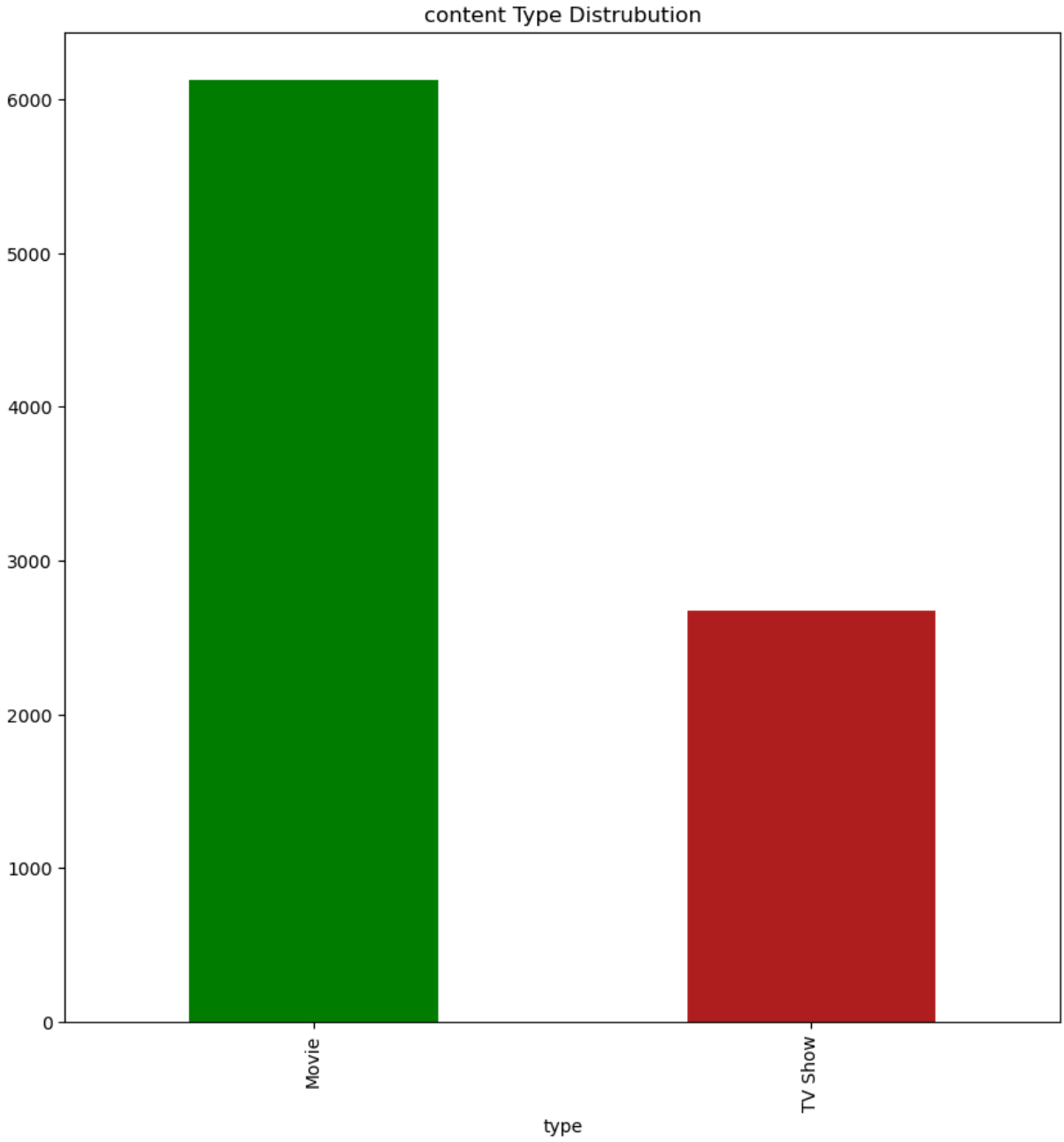
Content Type Distribution

In [92]:
```
types=data['type'].value_counts()
```

In [94]:
```
print(types)
```

```
type
Movie      6131
TV Show    2676
Name: count, dtype: int64
```

In [114…
```
plt.figure(figsize=(10,10))
types.plot(kind='bar',color=['green','firebrick'])
plt.title("content Type Distrubution")
plt.show()
```



Year with highest reales

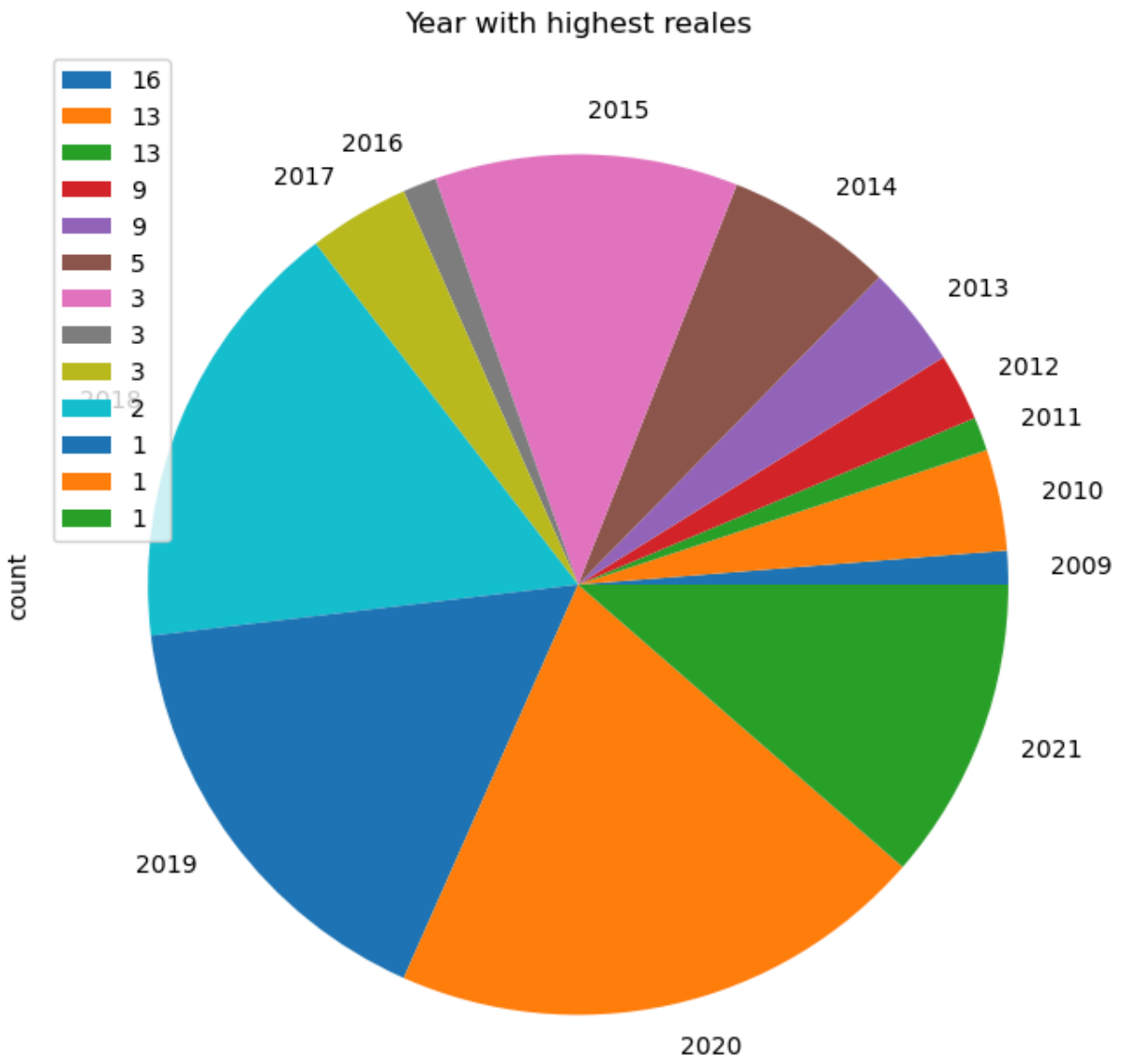
In [158…
```
year=data['release_year'].value_counts().head(10)
```

In [160…
```
counts=year.head(10)
print(counts)
```

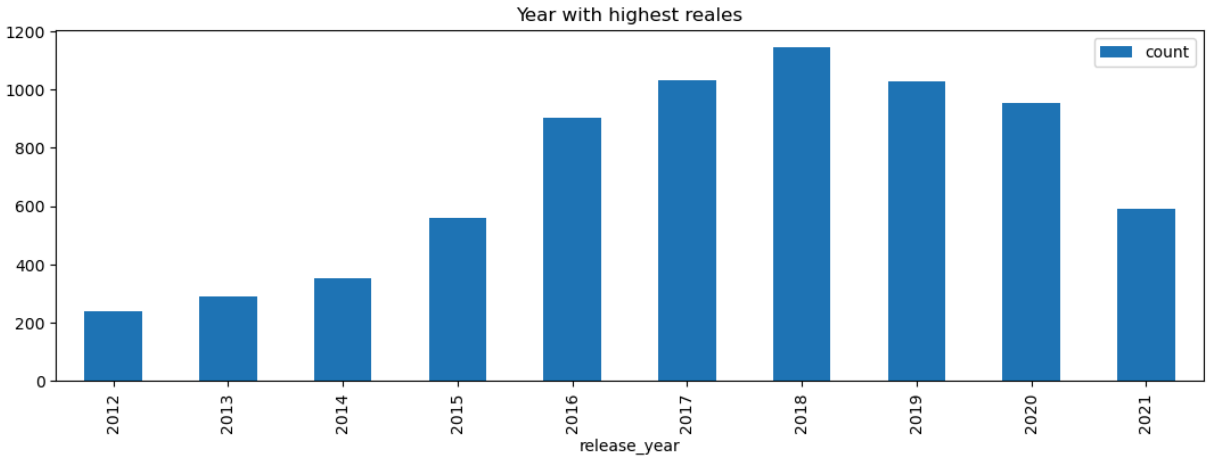
```
release_year
2018    1147
2017    1032
2019    1030
2020     953
2016     902
2021     592
2015     560
2014     352
2013     288
2012     237
Name: count, dtype: int64
```

In [383…

```
plt.figure(figsize=(10,8))
year.sort_index().plot(kind='pie',color=['green','firebrick','Aqua','MediumSlateBlu
plt.title("Year with highest reales")
plt.legend(year)
plt.show()
```



In [196…
```
plt.figure(figsize=(13,4))
year.sort_index().plot(kind='bar',)
plt.title("Year with highest reales")
plt.legend()
plt.show()
```



Indian TV Shows by Release Year

In [236…
```
india=data[(data['country'].str.lower()=='india')& (data['type'] == 'TV Show')]
x=india[['title', 'release_year']]
```

In [238…
```
print(x)
```

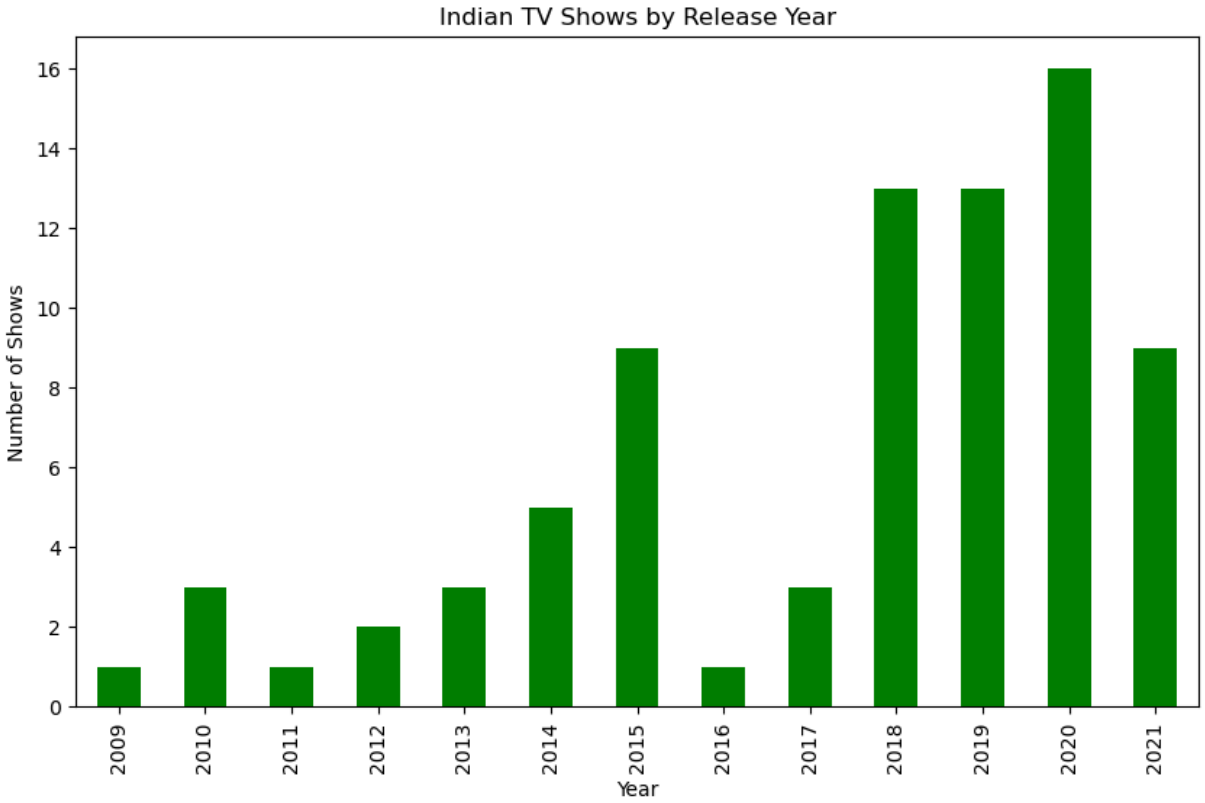
```
                                  title  release_year
4                          Kota Factory          2021
39                         Chhota Bheem          2021
50                        Dharmakshetra          2014
66         Raja Rasoi Aur Anya Kahaniyan          2014
69        Stories by Rabindranath Tagore          2015
...                                  ...           ...
8173                         Thackeray          2019
8235                       The Calling          2018
8321  The Golden Years with Javed Akhtar          2016
8349              The House That Made Me          2015
8775                    Yeh Meri Family          2018

[79 rows x 2 columns]
```

In [256…
```
year=india['release_year'].value_counts()
```

In [258…
```
year.head(10)
```

Out[258…
```
release_year
2020    16
2019    13
2018    13
2021     9
2015     9
2014     5
2017     3
2013     3
2010     3
2012     2
Name: count, dtype: int64
```

In [246…
```
plt.figure(figsize=(10, 6))
year.sort_index().plot(kind='bar', color='green')
plt.title('Indian TV Shows by Release Year')
plt.xlabel('Year')
plt.ylabel('Number of Shows')
plt.show()
```



MONTH WISE TV_SHOW OR MOVIES TOTAL

In [271…
```
data['date_added'] = data['date_added'].replace('unknown', pd.NA)
data['date_added'] = pd.to_datetime(data['date_added'])
data['month_added'] = data['date_added'].dt.month_name()
```

In [273…
```
print(data['month_added'])
```

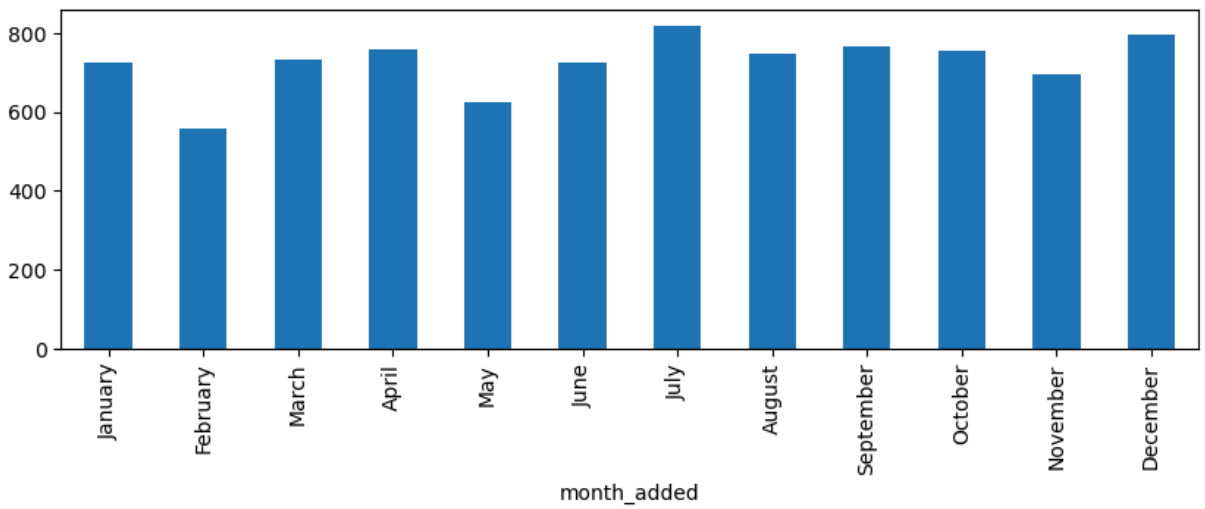
```
0       September
1       September
2       September
3       September
4       September
          ...
8802     November
8803         July
8804     November
8805      January
8806        March
Name: month_added, Length: 8807, dtype: object
```

In [289…
```
months=data['month_added'].value_counts()
print("Months vise counting of the total:",months)
```

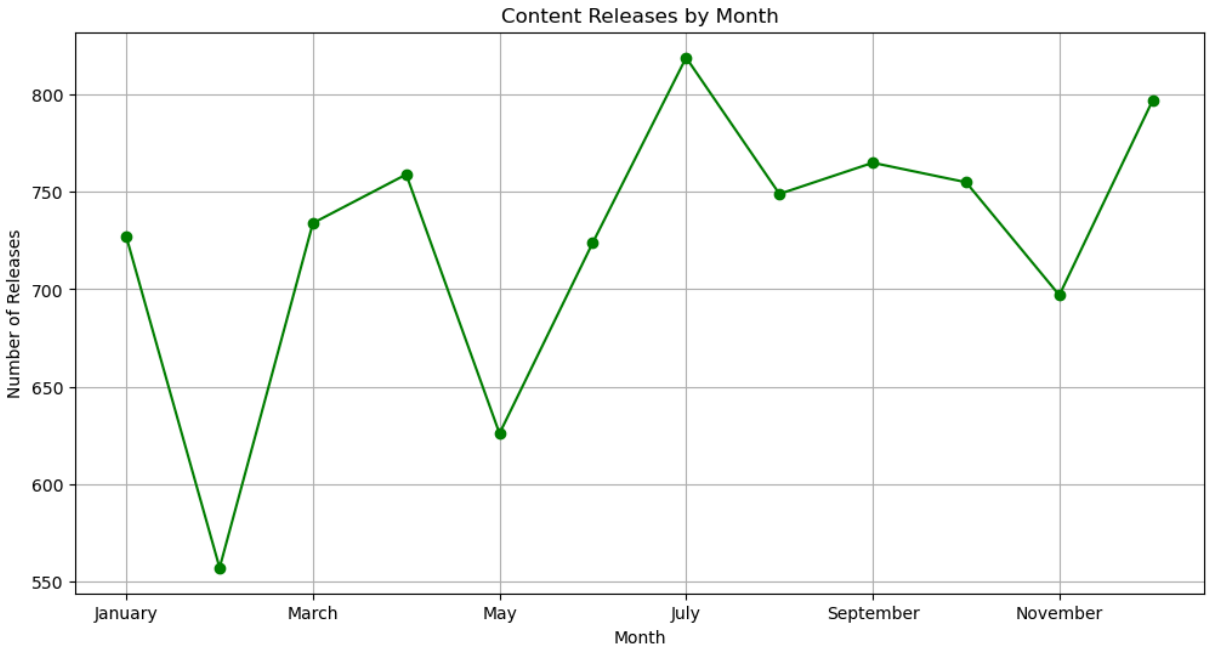
```
Months vise counting of the total: month_added
July          819
December      797
September     765
April         759
October       755
August        749
March         734
January       727
June          724
November      697
May           626
February      557
Name: count, dtype: int64
```

In [298…
```
months_order = ['January', 'February', 'March', 'April', 'May', 'June','July', 'Aug
re_month=months.reindex(months_order)
```

In [300…
```
plt.figure(figsize=(10,3))
re_month.plot(kind='bar')
plt.show()
```



In [308…
```
plt.figure(figsize=(12, 6))
re_month.plot(kind='line', marker='o', color='green')
plt.title('Content Releases by Month')
plt.xlabel('Month')
plt.ylabel('Number of Releases')
plt.grid(True)
plt.show()
```



Content By Country

In [ ]:
```
```

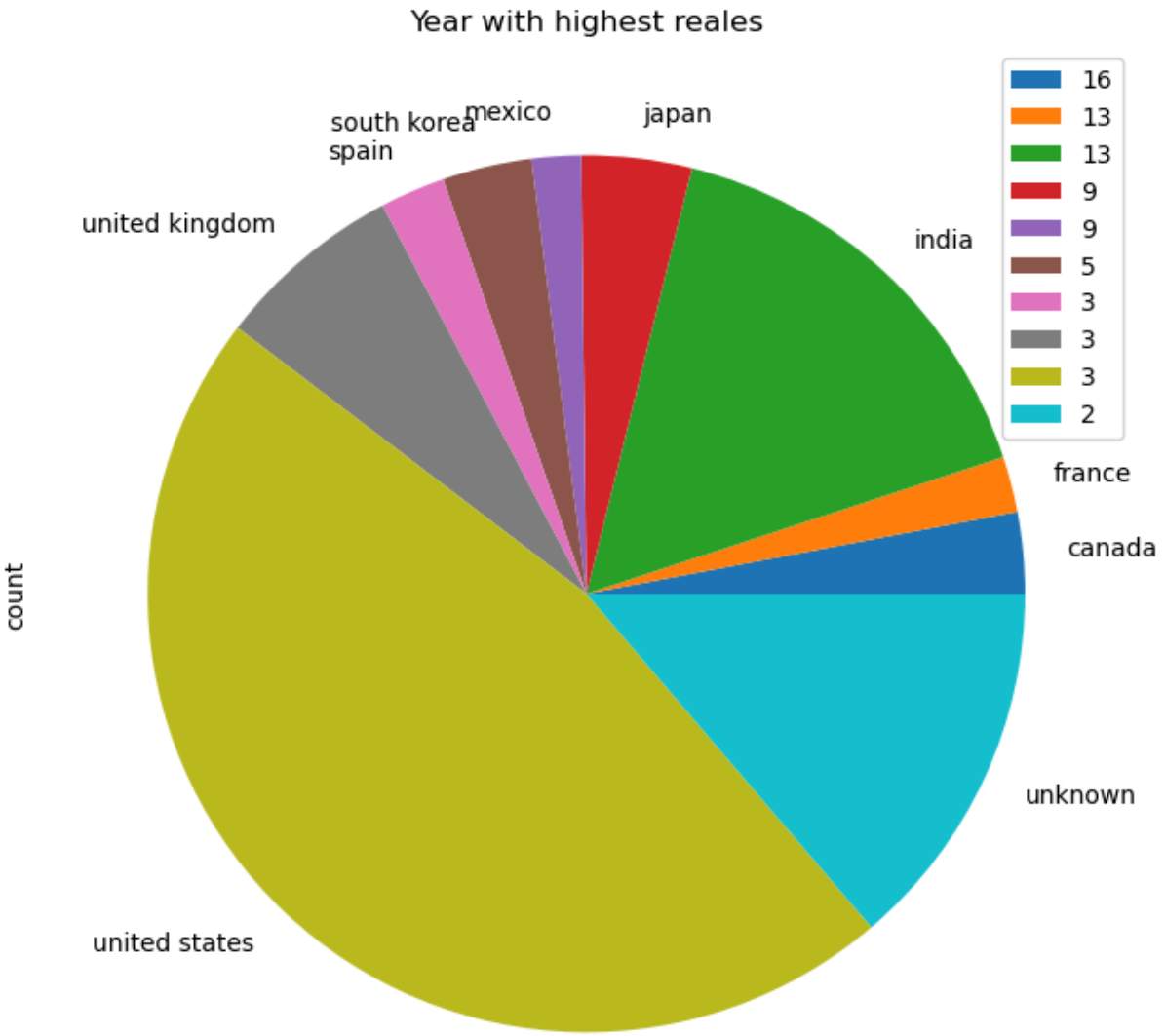
In [315…
```
content_country=data['country'].value_counts()
```

Top 10 countries

In [324…
```
top_country=content_country.head(10)
print(top_country)
```

```
country
united states     2818
india              972
unknown            831
united kingdom     419
japan              245
south korea        199
canada             181
spain              145
france             124
mexico             110
Name: count, dtype: int64
```

In [322…
```
plt.figure(figsize=(15,8))
top_country.sort_index().plot(kind='pie',color=['green','firebrick','Aqua','MediumS
plt.title("Year with highest reales")
plt.legend(year)
plt.show()
```

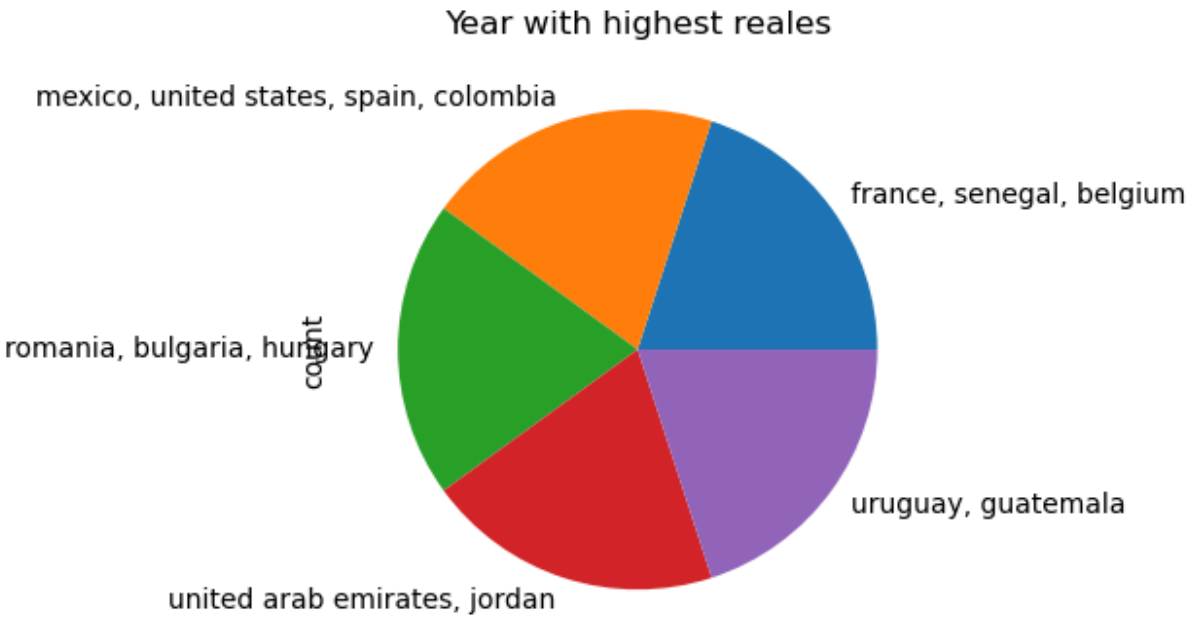


last 5 country are

In [327…
```
last_country=content_country.tail()
print(last_country)
```

```
country
romania, bulgaria, hungary                1
uruguay, guatemala                        1
france, senegal, belgium                  1
mexico, united states, spain, colombia    1
united arab emirates, jordan              1
Name: count, dtype: int64
```

In [339…
```
plt.figure(figsize=(10,4))
last_country.sort_index().plot(kind='pie',color=['green','firebrick','Aqua','Medium
plt.title("Year with highest reales")
plt.show()
```



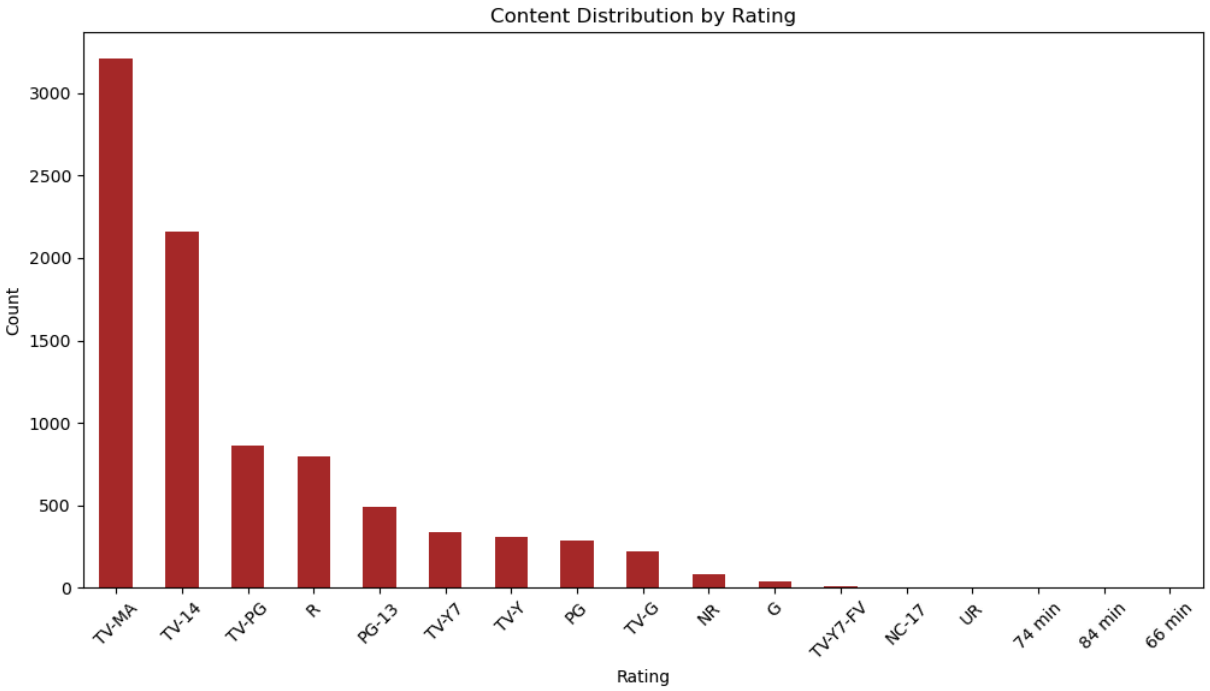
Rating wise distrubution

In [361…
```
print("\nContent Ratings:")
ratings=data['rating'].value_counts()
ratings
```

```
Content Ratings:
```

Out[361…
```
rating
TV-MA       3207
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
NC-17          3
UR             3
74 min         1
84 min         1
66 min         1
Name: count, dtype: int64
```

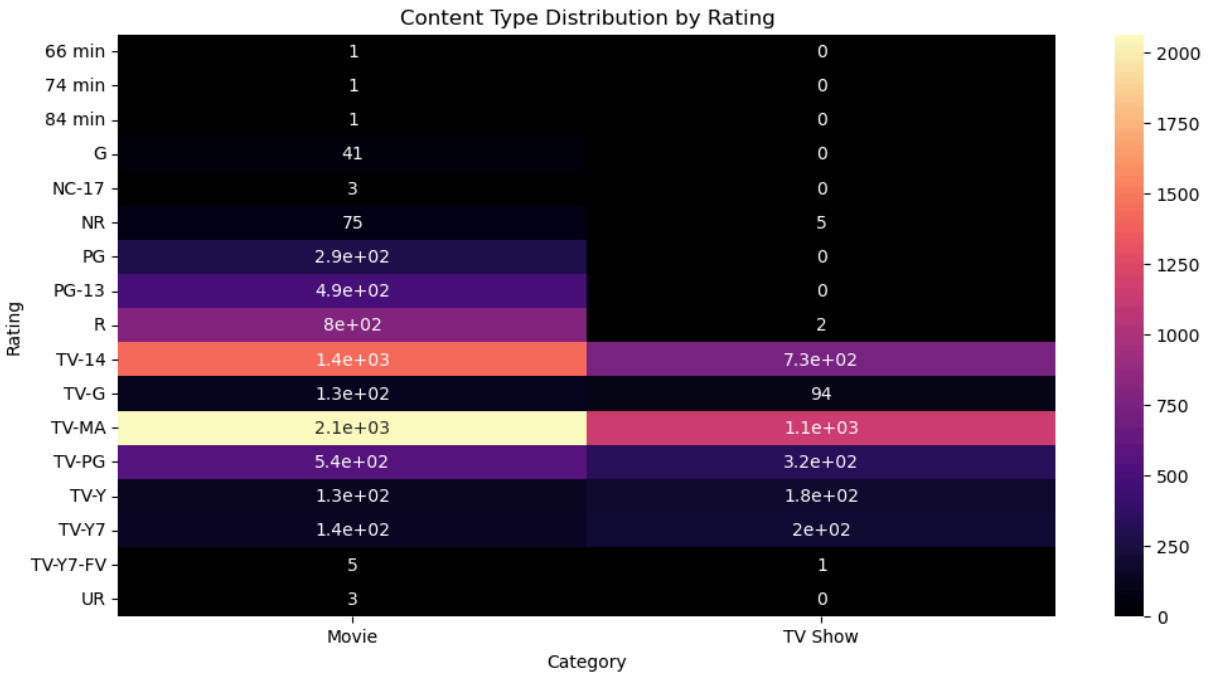
In [365…
```
plt.figure(figsize=(12, 6))
ratings.plot(kind='bar', color='brown')
plt.title('Content Distribution by Rating')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



Content Type Distribution by Rating

In [ ]:
```
rating_by_type = pd.crosstab(data['rating'], data['type'])
```

In [380…
```
plt.figure(figsize=(12, 6))
sns.heatmap(rating_by_type, annot=True, cmap='magma')
plt.title('Content Type Distribution by Rating')
plt.xlabel('Category')
plt.ylabel('Rating')
plt.show()
```


Content Type Distribution by Rating
66 min
74 min
84 min
G
NC-17
NR
PG
PG-13
R
TV-14
TV-G
TV-MA
TV-PG
TV-Y
TV-Y7
TV-Y7-FV
UR
Rating
1
0
1
0
1
0
41
0
3
0
75
5
2.9e+02
0
4.9e+02
0
8e+02
2
1.4e+03
7.3e+02
1.3e+02
94
2.1e+03
1.1e+03
5.4e+02
3.2e+02
1.3e+02
1.8e+02
1.4e+02
2e+02
5
1
3
0
Movie
TV Show
Category
2000
1750
1500
1250
1000
750
500
250
0

In [ ]: