In [412]:

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

In [413]:

```
#import csv file(data)
data = pd.read_csv('tv_shows.csv')
```

In [414]:

```
#informaion regarding the data
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5611 entries, 0 to 5610
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Unnamed: 0       5611 non-null   int64  
 1   Title            5611 non-null   object 
 2   Year             5611 non-null   int64  
 3   Age              3165 non-null   object 
 4   IMDb             4450 non-null   float64
 5   Rotten Tomatoes  1011 non-null   object 
 6   Netflix          5611 non-null   int64  
 7   Hulu             5611 non-null   int64  
 8   Prime Video      5611 non-null   int64  
 9   Disney+          5611 non-null   int64  
 10  type             5611 non-null   int64  
dtypes: float64(1), int64(7), object(3)
memory usage: 482.3+ KB
```

In [415]:

```
# type
type(data)
```

Out[415]:

```
pandas.core.frame.DataFrame
```

In [416]:

```
#shape of the Dataset
data.shape
```

Out[416]:

```
(5611, 11)
```

In [417]:

```
#first 5 rows of the dataset
data.head()
```

Out[417]:

	Unnamed: 0	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	type
0	0	Breaking Bad	2008	18+	9.5	96%	1	0	0	0	1
1	1	Stranger Things	2016	16+	8.8	93%	1	0	0	0	1
2	2	Money Heist	2017	18+	8.4	91%	1	0	0	0	1
3	3	Sherlock	2010	16+	9.1	78%	1	0	0	0	1
4	4	Better Call Saul	2015	18+	8.7	97%	1	0	0	0	1

In [418]:

```
#Last 5 rows of the dataset
data.tail()
```

Out[418]:

	Unnamed: 0	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	type
5606	5606	Tut's Treasures: Hidden Secrets	2018	NaN	NaN	NaN	0	0	0	1	1
5607	5607	Paradise Islands	2017	NaN	NaN	NaN	0	0	0	1	1
5608	5608	Wild Russia	2018	NaN	NaN	NaN	0	0	0	1	1
5609	5609	Love & Vets	2017	NaN	NaN	NaN	0	0	0	1	1
5610	5610	United States of Animals	2016	NaN	NaN	NaN	0	0	0	1	1

In [419]:

```
#Title of the first 5 datarows
data['Title'].head()
```

Out[419]:

```
0         Breaking Bad
1      Stranger Things
2          Money Heist
3             Sherlock
4     Better Call Saul
Name: Title, dtype: object
```

In [420]:

```
#dimension of the object
data.ndim
```

Out[420]:

```
2
```

In [421]:

```
#size of the object
data.size
```

Out[421]:

```
61721
```

In [422]:

```
#columns/ features of the dataset
data.axes
```

Out[422]:

```
[RangeIndex(start=0, stop=5611, step=1),
 Index(['Unnamed: 0', 'Title', 'Year', 'Age', 'IMDb', 'Rotten Tomatoes',
        'Netflix', 'Hulu', 'Prime Video', 'Disney+', 'type'],
       dtype='object')]
```

In [423]:

```
#columns/ features of the dataset
data.columns
```

Out[423]:

```
Index(['Unnamed: 0', 'Title', 'Year', 'Age', 'IMDb', 'Rotten Tomatoes',
       'Netflix', 'Hulu', 'Prime Video', 'Disney+', 'type'],
      dtype='object')
```

In [424]:

```
#Datatypes of all the columns
data.dtypes
```

Out[424]:

```
Unnamed: 0           int64
Title               object
Year                 int64
Age                 object
IMDb               float64
Rotten Tomatoes     object
Netflix              int64
Hulu                 int64
Prime Video          int64
Disney+              int64
type                 int64
dtype: object
```

In [425]:

```
#checking the emptyiness of the datset
data.empty
```

Out[425]:

```
False
```

In [426]:

```
#Technique to convert DataFrame to Numpy array
dt = data.values
dt[0]
```

Out[426]:

```
array([0, 'Breaking Bad', 2008, '18+', 9.5, '96%', 1, 0, 0, 0, 1],
      dtype=object)
```

In [427]:

```
type(dt)
```

Out[427]:

```
numpy.ndarray
```

In [428]:
```
print('Tv shows on Netflix:',data['Netflix'].sum(),'/',data['Netflix'].count())
print('Tv shows on Hulu:',data['Hulu'].sum(),'/',data['Hulu'].count())
print('Tv shows on Prime Video:',data['Prime Video'].sum(),'/',data['Prime Video'
print('Tv shows on Disney+:',data['Disney+'].sum(),'/',data['Disney+'].count())
```

```
Tv shows on Netflix: 1931 / 5611
Tv shows on Hulu: 1754 / 5611
Tv shows on Prime Video: 2144 / 5611
Tv shows on Disney+: 180 / 5611
```

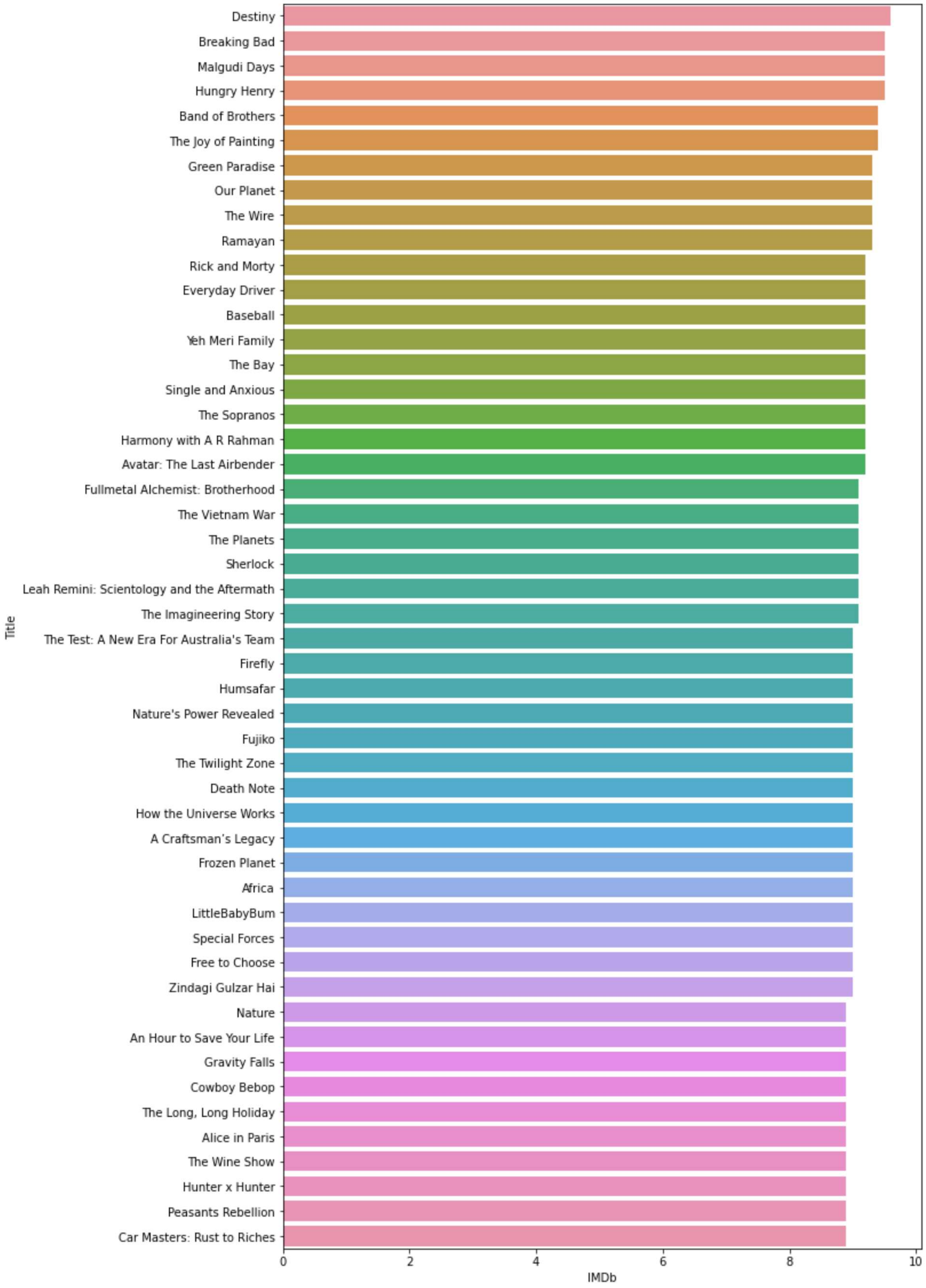
In [429]:
```
#describe the dataset with some basic functionality
data.describe(include='all')
```

Out[429]:

	Unnamed: 0	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	H
count	5611.000000	5611	5611.000000	3165	4450.000000	1011	5611.000000	5611.000
unique	NaN	5564	NaN	5	NaN	88	NaN	N
top	NaN	Undercover	NaN	16+	NaN	100%	NaN	N
freq	NaN	3	NaN	1018	NaN	109	NaN	N
mean	2805.000000	NaN	2011.021030	NaN	7.113258	NaN	0.344145	0.312
std	1619.900511	NaN	11.005116	NaN	1.132060	NaN	0.475131	0.463
min	0.000000	NaN	1901.000000	NaN	1.000000	NaN	0.000000	0.000
25%	1402.500000	NaN	2010.000000	NaN	6.600000	NaN	0.000000	0.000
50%	2805.000000	NaN	2015.000000	NaN	7.300000	NaN	0.000000	0.000
75%	4207.500000	NaN	2017.000000	NaN	7.900000	NaN	1.000000	1.000
max	5610.000000	NaN	2020.000000	NaN	9.600000	NaN	1.000000	1.000

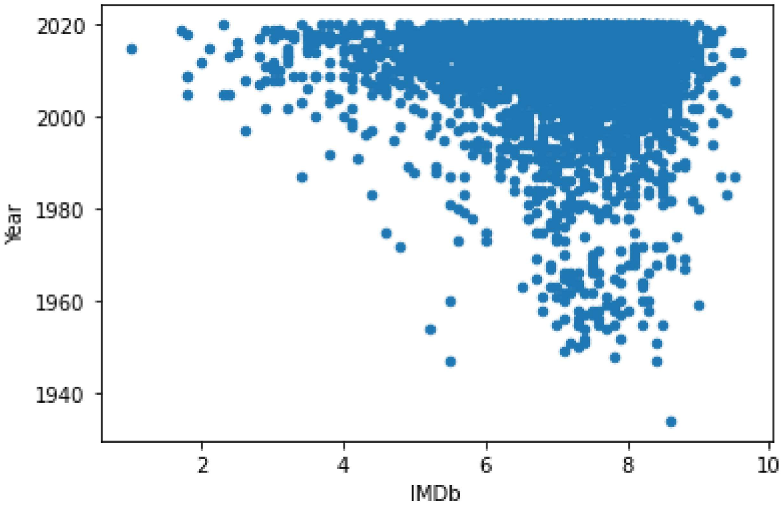
In [430]:
```
#top 50 IMDb rated Tv shows
plt.subplots(figsize=(10,20))
sns.barplot(x="IMDb", y="Title" , data= data.sort_values("IMDb",ascending=False).
```

Out[430]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x260c2cac5f8>
```



```
In [431]: data.plot.scatter(x='IMDb', y='Year')

Out[431]: <matplotlib.axes._subplots.AxesSubplot at 0x260c2e5c438>
```



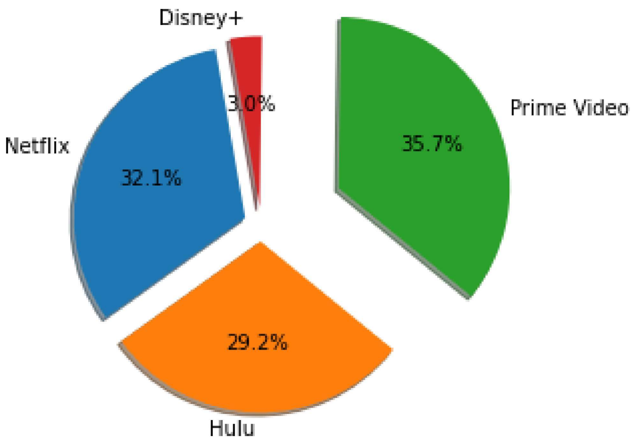
In [432]:

```
#quantity of shows on various platforms present in the dataset
labels = 'Netflix' , 'Hulu', 'Prime Video', 'Disney+'
sizes = [data['Netflix'].sum(),data['Hulu'].sum(),data['Prime Video'].sum(),data
explode = (0.1, 0.1, 0.5, 0.1 )

fig1 , ax1 = plt.subplots()

ax1.pie(sizes,
        explode = explode,
        labels = labels,
        autopct = '%1.1f%%',
        shadow = True,
        startangle = 100)

ax1.axis ('equal')
plt.show()
```



In [433]:

```
netflix_shows = data.loc[data['Netflix'] == 1]
hulu_shows = data.loc[data['Hulu'] == 1]
prime_video_shows = data.loc[data['Prime Video'] == 1]
disney_shows = data.loc[data['Disney+'] == 1]
```

In [434]:

```
#list of top shows on netflix
netflix_top_shows = netflix_shows.loc[netflix_shows['IMDb']>8.0]
hulu_top_shows = hulu_shows.loc[hulu_shows['IMDb']>8.0]
prime_video_top_shows = prime_video_shows.loc[prime_video_shows['IMDb']>8.0]
disney_top_shows = disney_shows.loc[disney_shows['IMDb']>8.0]
```

In [435]:

```
netflix_top_shows['IMDb'].sum()
```

Out[435]:

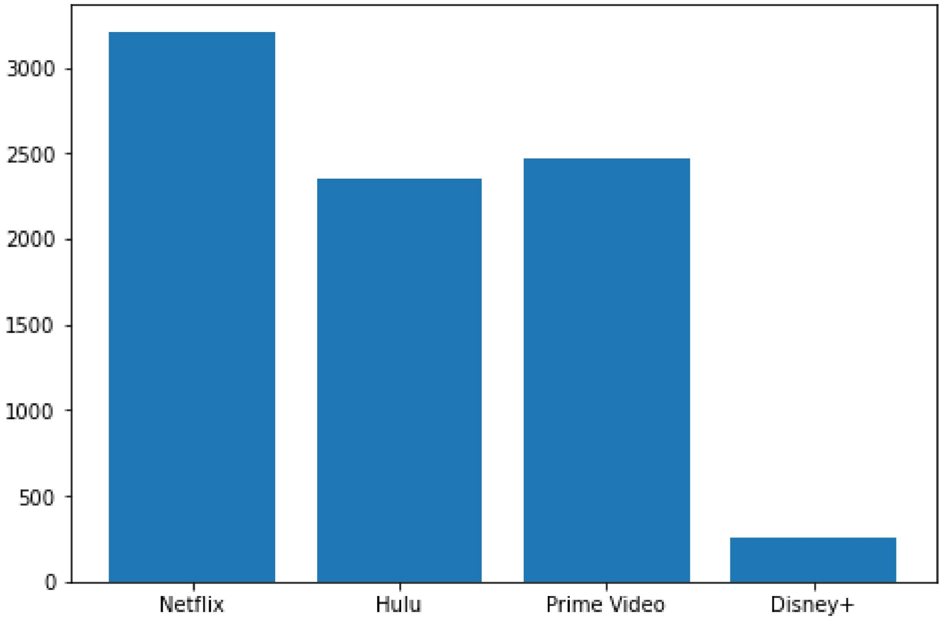
```
3213.2
```

In [436]:

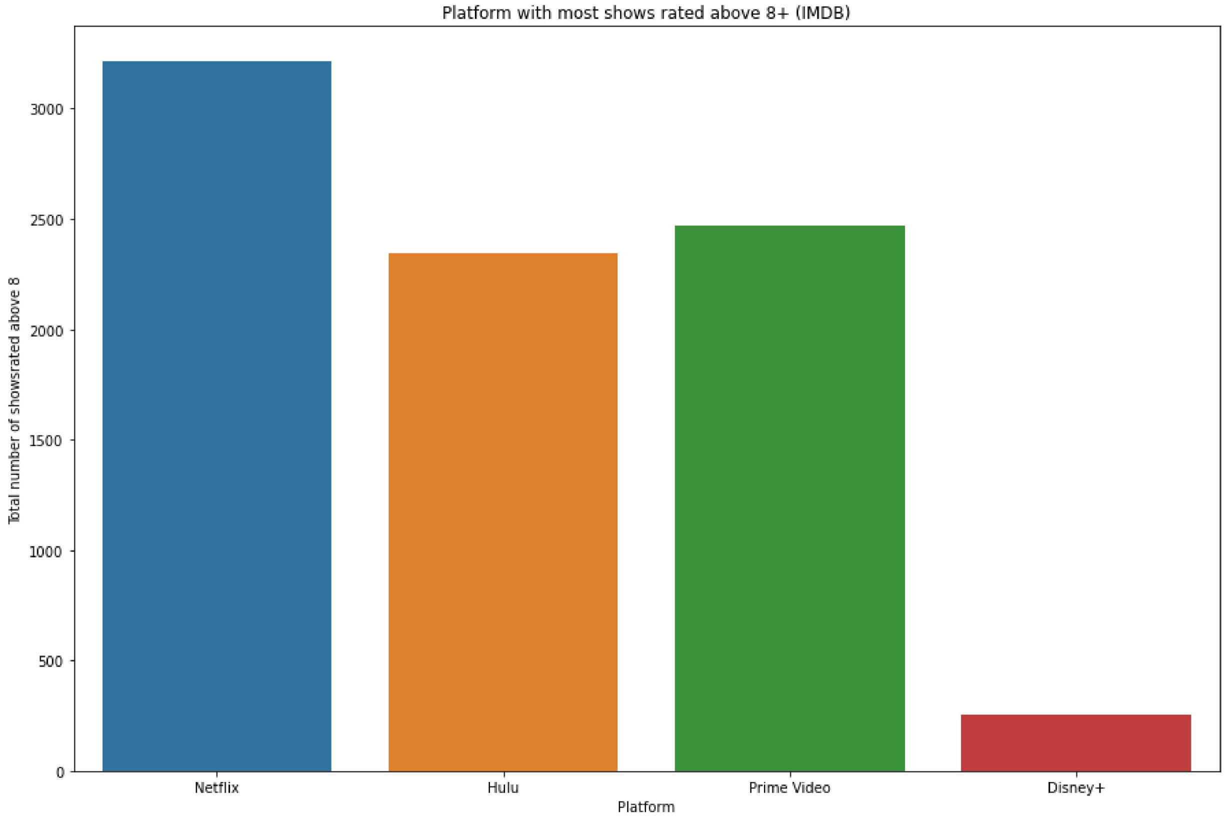
```
#lets plot a bar graph of platforms with highest IMDb shows
platform = ['Netflix', 'Hulu', 'Prime Video', 'Disney+']
count = [netflix_top_shows['IMDb'].sum(),hulu_top_shows['IMDb'].sum(),prime_videc
```

In [437]:

```
fig = plt.figure()
ax = fig.add_axes([1,1,1,1])
ax.bar(platform,count)
plt.show()
```



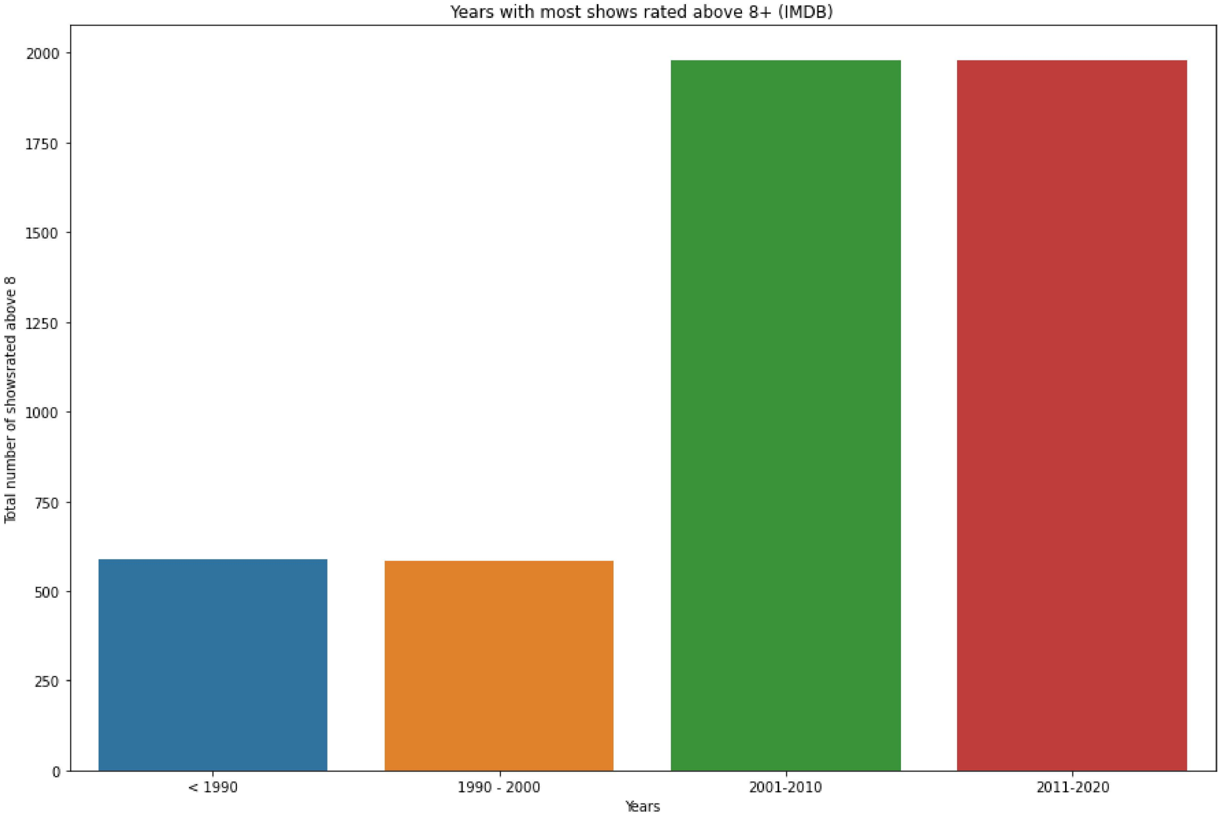
In [438]:
```
#Platform with most shows rated above 8+ (IMDB)
plt.figure(figsize = (15, 10))
sns.barplot(
           x = platform,
           y = count
)
plt.xlabel('Platform')
plt.ylabel('Total number of showsrated above 8')
plt.title('Platform with most shows rated above 8+ (IMDB)')
plt.show()
```



In [439]:

```
#lets plot a bar graph of years with highest IMDb shows
top1990_shows = data.loc[(data['IMDb'] >= 8.0) & (data['Year']<= 1990)]
top2000_shows = data.loc[(data['IMDb']>=8.0) & (data['Year']>1990) & (data['Year'
top2010_shows = data.loc[(data['IMDb']>=8.0) & (data['Year']>2000)&(data['Year']<
top2020_shows = data.loc[(data['IMDb']>=8.0) & (data['Year']>2010)&(data['Year']<
years = ['< 1990', '1990 - 2000', '2001-2010', '2011-2020']
counts = [top1990_shows['IMDb'].sum(),top2000_shows['IMDb'].sum(),top2010_shows['

plt.figure(figsize = (15, 10))
sns.barplot(
          x = years,
          y = counts
)
plt.xlabel('Years')
plt.ylabel('Total number of showsrated above 8')
plt.title('Years with most shows rated above 8+ (IMDB)')
plt.show()
```



```
In [ ]:
```

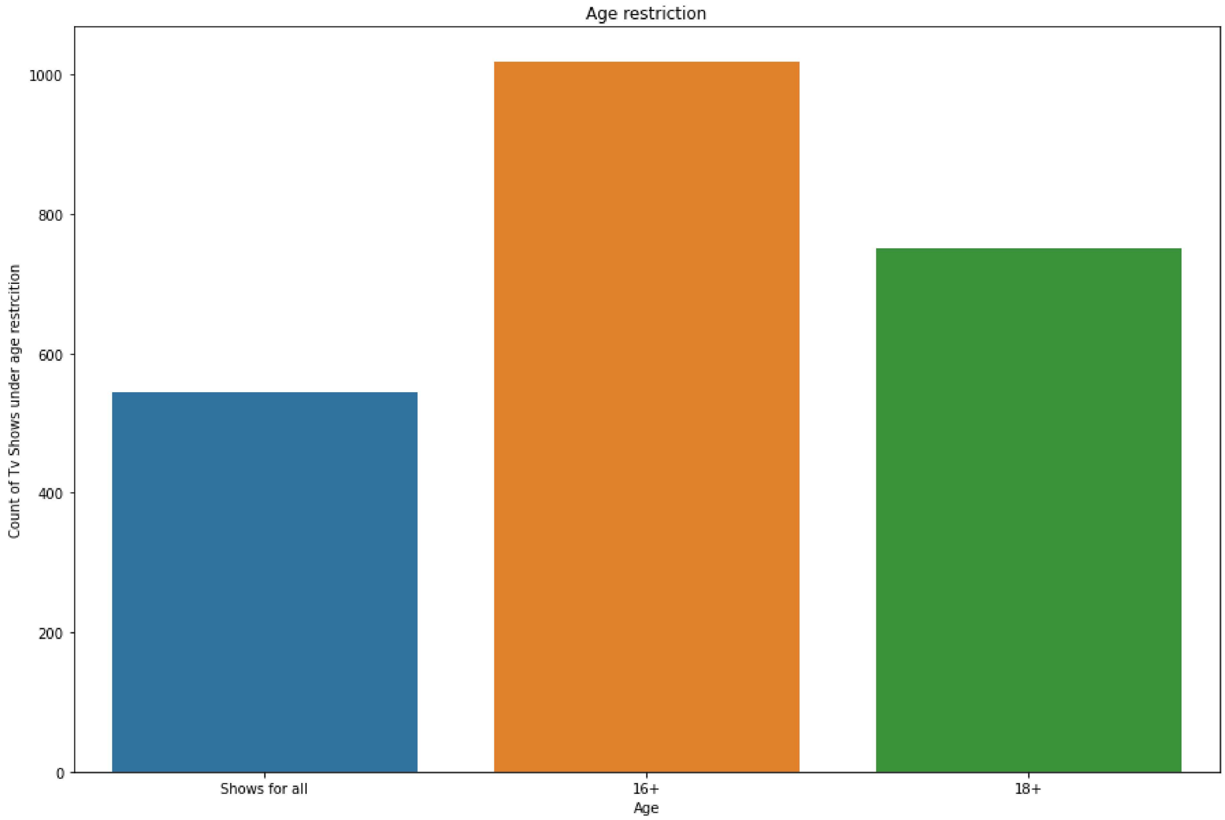
```
In [440]: all_rated = data.loc[data['Age']=='all']
          _16_rated = data.loc[data['Age']=='16+']
          _18_rated = data.loc[data['Age']=='18+']
```

```
In [441]: print(len(all_rated))
          print(len(_16_rated))
          print(len(_18_rated))
```

```
545
1018
750
```

In [442]:
```
age = ['Shows for all', '16+', '18+']
counts = [len(all_rated),len(_16_rated),len(_18_rated)]

plt.figure(figsize = (15, 10))
sns.barplot(
           x = age,
           y = counts
)
plt.xlabel('Age')
plt.ylabel('Count of Tv Shows under age restrcition')
plt.title('Age restriction')
plt.show()
```



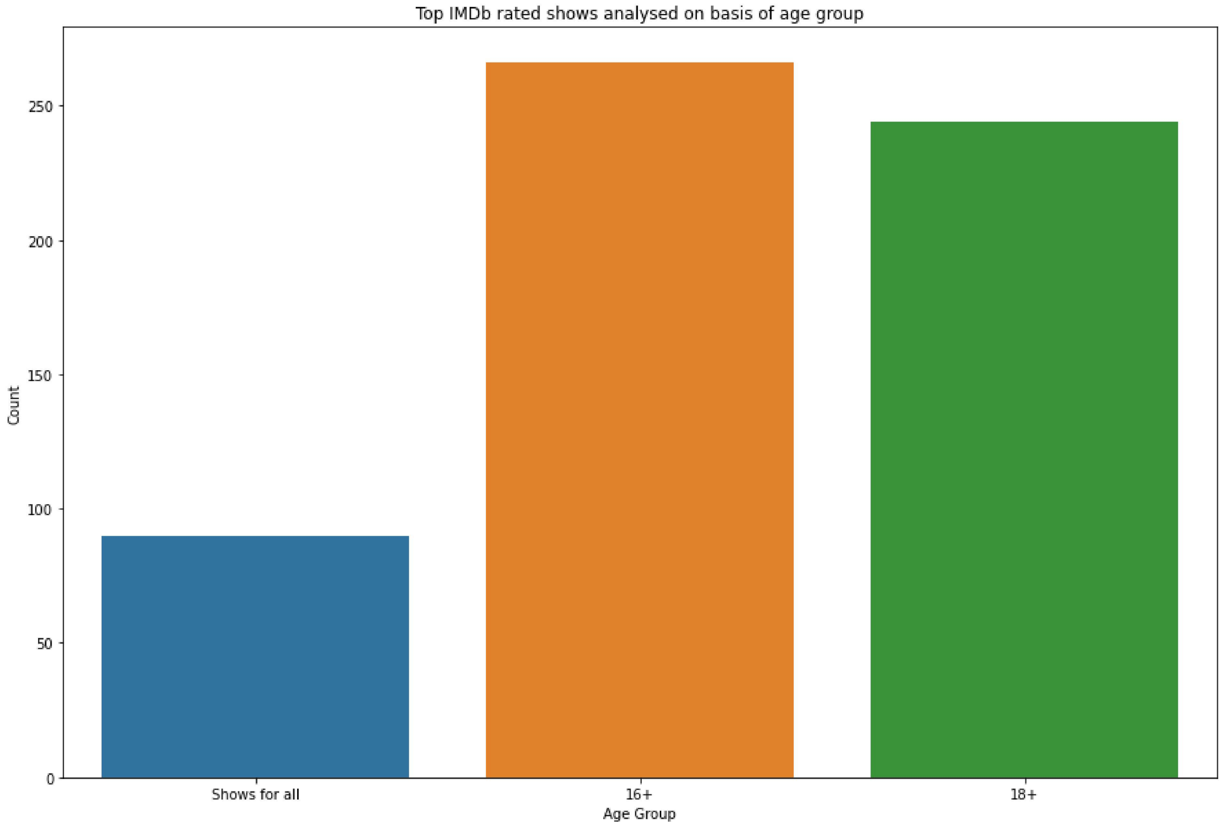
In [443]:
```
# movie with IMDb 8+ which are for all age groups
all_rated_high_rate = all_rated.loc[all_rated['IMDb']>=8.0]
rated_16_high_rate = _16_rated.loc[_16_rated['IMDb']>=8.0]
rated_18_high_rate = _18_rated.loc[_18_rated['IMDb']>=8.0]
```

In [444]:
```
all_rated_high_rate['Title'].count()
rated_16_high_rate['Title'].count()
rated_18_high_rate['Title'].count()
```

Out[444]:
```
244
```

In [445]:
```
#Top IMDb rated Shows analysed on basis of age group
age = ['Shows for all', '16+', '18+']
counts = [len(all_rated_high_rate),len(rated_16_high_rate),len(rated_18_high_rate

plt.figure(figsize = (15, 10))
sns.barplot(
           x = age,
           y = counts
)
plt.xlabel('Age Group')
plt.ylabel('Count')
plt.title('Top IMDb rated shows analysed on basis of age group')
plt.show()
```



In [446]:
```
#Must watch shows
must_watch = data.loc[data['IMDb']>9.0]
```

In [447]:

```
must_watch['Title']
```

Out[447]:

```
0                                 Breaking Bad
3                                     Sherlock
9                   Avatar: The Last Airbender
15            Fullmetal Alchemist: Brotherhood
91                                  Our Planet
97                             The Vietnam War
282                            Yeh Meri Family
325                                    Ramayan
1931                            Rick and Morty
2236    Leah Remini: Scientology and the Aftermath
2365                       The Joy of Painting
3023                                   Destiny
3177                              Hungry Henry
3566                                  The Wire
3567                          Band of Brothers
3568                              The Sopranos
3649                               The Planets
3701                                  Baseball
3747                              Malgudi Days
3798                                   The Bay
4029                   Harmony with A R Rahman
4041                           Everyday Driver
4128                            Green Paradise
4257                        Single and Anxious
5465                    The Imagineering Story
Name: Title, dtype: object
```

In [448]:

```
#Top 5 Must watch Shows
a = data.sort_values("IMDb",ascending=False).head(5)
a['Title']
```

Out[448]:

```
3023             Destiny
0           Breaking Bad
3747        Malgudi Days
3177        Hungry Henry
3567    Band of Brothers
Name: Title, dtype: object
```