In [2]:
```
!pip install pandas
```

```
Requirement already satisfied: pandas in e:\anaconda\lib\site-packages (1.4.4)
Requirement already satisfied: numpy>=1.18.5 in e:\anaconda\lib\site-packages (fro
m pandas) (1.21.5)
Requirement already satisfied: pytz>=2020.1 in e:\anaconda\lib\site-packages (from
pandas) (2022.1)
Requirement already satisfied: python-dateutil>=2.8.1 in e:\anaconda\lib\site-pack
ages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in e:\anaconda\lib\site-packages (from pyt
hon-dateutil>=2.8.1->pandas) (1.16.0)
```

In [3]:
```
!pip install numpy
```

```
Requirement already satisfied: numpy in e:\anaconda\lib\site-packages (1.21.5)
```

In [4]:
```
!pip install seaborn
```

```
Requirement already satisfied: seaborn in e:\anaconda\lib\site-packages (0.11.2)
Requirement already satisfied: matplotlib>=2.2 in e:\anaconda\lib\site-packages (f
rom seaborn) (3.5.2)
Requirement already satisfied: scipy>=1.0 in e:\anaconda\lib\site-packages (from s
eaborn) (1.9.1)
Requirement already satisfied: numpy>=1.15 in e:\anaconda\lib\site-packages (from
seaborn) (1.21.5)
Requirement already satisfied: pandas>=0.23 in e:\anaconda\lib\site-packages (from
seaborn) (1.4.4)
Requirement already satisfied: cycler>=0.10 in e:\anaconda\lib\site-packages (from
matplotlib>=2.2->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in e:\anaconda\lib\site-packages
(from matplotlib>=2.2->seaborn) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in e:\anaconda\lib\site-packages
(from matplotlib>=2.2->seaborn) (1.4.2)
Requirement already satisfied: packaging>=20.0 in e:\anaconda\lib\site-packages (f
rom matplotlib>=2.2->seaborn) (21.3)
Requirement already satisfied: python-dateutil>=2.7 in e:\anaconda\lib\site-packag
es (from matplotlib>=2.2->seaborn) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in e:\anaconda\lib\site-packages
(from matplotlib>=2.2->seaborn) (3.0.9)
Requirement already satisfied: pillow>=6.2.0 in e:\anaconda\lib\site-packages (fro
m matplotlib>=2.2->seaborn) (9.2.0)
Requirement already satisfied: pytz>=2020.1 in e:\anaconda\lib\site-packages (from
pandas>=0.23->seaborn) (2022.1)
Requirement already satisfied: six>=1.5 in e:\anaconda\lib\site-packages (from pyt
hon-dateutil>=2.7->matplotlib>=2.2->seaborn) (1.16.0)
```

In [5]:
```
!pip install matplotlib
```

```
Requirement already satisfied: matplotlib in e:\anaconda\lib\site-packages (3.5.2)
Requirement already satisfied: python-dateutil>=2.7 in e:\anaconda\lib\site-packag
es (from matplotlib) (2.8.2)
Requirement already satisfied: fonttools>=4.22.0 in e:\anaconda\lib\site-packages
(from matplotlib) (4.25.0)
Requirement already satisfied: pyparsing>=2.2.1 in e:\anaconda\lib\site-packages
(from matplotlib) (3.0.9)
Requirement already satisfied: packaging>=20.0 in e:\anaconda\lib\site-packages (f
rom matplotlib) (21.3)
Requirement already satisfied: kiwisolver>=1.0.1 in e:\anaconda\lib\site-packages
(from matplotlib) (1.4.2)
Requirement already satisfied: cycler>=0.10 in e:\anaconda\lib\site-packages (from
matplotlib) (0.11.0)
Requirement already satisfied: numpy>=1.17 in e:\anaconda\lib\site-packages (from
matplotlib) (1.21.5)
Requirement already satisfied: pillow>=6.2.0 in e:\anaconda\lib\site-packages (fro
m matplotlib) (9.2.0)
Requirement already satisfied: six>=1.5 in e:\anaconda\lib\site-packages (from pyt
hon-dateutil>=2.7->matplotlib) (1.16.0)
```

In [7]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [9]:
```
!pip install plotly
```

```
Requirement already satisfied: plotly in e:\anaconda\lib\site-packages (5.9.0)
Requirement already satisfied: tenacity>=6.2.0 in e:\anaconda\lib\site-packages (f
rom plotly) (8.0.1)
```

In [11]:
```
import plotly.offline as pyo
pyo.init_notebook_mode()
```

In [12]:
```
import plotly.express as px
```

In [13]:
```
matplotlib inline
```

In [19]:
```
df = pd.read_csv("Downloads/moviestreams.csv")
df.head()
```

Out[19]:

	Unnamed: 0	ID	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	Type
0	0	1	Inception	2010	13+	8.8	87%	1	0	0	0	0
1	1	2	The Matrix	1999	18+	8.7	87%	1	0	0	0	0
2	2	3	Avengers: Infinity War	2018	13+	8.5	84%	1	0	0	0	0
3	3	4	Back to the Future	1985	7+	8.5	96%	1	0	0	0	0
4	4	5	The Good, the Bad and the Ugly	1966	18+	8.8	97%	1	0	1	0	0

In [24]:
```
df.shape
```

Out[24]:
```
(16744, 17)
```

In [25]:
```
cols = df.columns.tolist()
cols
```

Out[25]:
```
['Unnamed: 0',
 'ID',
 'Title',
 'Year',
 'Age',
 'IMDb',
 'Rotten Tomatoes',
 'Netflix',
 'Hulu',
 'Prime Video',
 'Disney+',
 'Type',
 'Directors',
 'Genres',
 'Country',
 'Language',
 'Runtime']
```

In [26]:
```
df.drop(['Unnamed: 0' , 'ID'] , axis = 1, inplace = True)
cols = df.columns.tolist()
cols
```

Out[26]:
```
['Title',
 'Year',
 'Age',
 'IMDb',
 'Rotten Tomatoes',
 'Netflix',
 'Hulu',
 'Prime Video',
 'Disney+',
 'Type',
 'Directors',
 'Genres',
 'Country',
 'Language',
 'Runtime']
```

In [ ]:
```
// python recognise missing values as NaN
```

In [27]:
```
df.isna()
```

Out[27]:

	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	Type	Directors	Gen
0	False	False	False	False	False	False	False	False	False	False	False	Fa
1	False	False	False	False	False	False	False	False	False	False	False	Fa
2	False	False	False	False	False	False	False	False	False	False	False	Fa
3	False	False	False	False	False	False	False	False	False	False	False	Fa
4	False	False	False	False	False	False	False	False	False	False	False	Fa
...	...	...	...	...	...	...	...	...	...	...	...	
16739	False	False	True	False	True	False	False	False	False	False	False	Fa
16740	False	False	False	False	True	False	False	False	False	False	False	Fa
16741	False	False	True	False	True	False	False	False	False	False	False	Fa
16742	False	False	True	False	True	False	False	False	False	False	False	Fa
16743	False	False	True	True	True	False	False	False	False	False	False	Fa

16744 rows × 15 columns

In [28]:
```
df.isna().sum()
```

Out[28]:
```
Title                0
Year                 0
Age               9390
IMDb               571
Rotten Tomatoes  11586
Netflix              0
Hulu                 0
Prime Video          0
Disney+              0
Type                 0
Directors          726
Genres             275
Country            435
Language           599
Runtime            592
dtype: int64
```

In [29]:
```
df.dtypes
```

Out[29]:
```
Title              object
Year                int64
Age                object
IMDb              float64
Rotten Tomatoes    object
Netflix             int64
Hulu                int64
Prime Video         int64
Disney+             int64
Type                int64
Directors          object
Genres             object
Country            object
Language           object
Runtime           float64
dtype: object
```

In [30]:
```
df['Age']
```

Out[30]:
```
0        13+
1        18+
2        13+
3         7+
4        18+
        ...
16739    NaN
16740     7+
16741    NaN
16742    NaN
16743    NaN
Name: Age, Length: 16744, dtype: object
```

In [31]:
```
age_map = { '18+' : 18, '7+' : 7, '13+' : 13 , 'All': 0, '16': 16}
df['AgeCopy'] = df['Age'].map(age_map)
df['AgeCopy']
```

Out[31]:
```
0        13.0
1        18.0
2        13.0
3         7.0
4        18.0
         ...
16739     NaN
16740     7.0
16741     NaN
16742     NaN
16743     NaN
Name: AgeCopy, Length: 16744, dtype: float64
```

In [35]:
```
df['New_Rotten_tomatoes'] = df['Rotten Tomatoes'].str.replace('$', '')
for i in df['New_Rotten_tomatoes']:
    if i == str:
        i.astype(int)
```

```
C:\Users\Hp\AppData\Local\Temp\ipykernel_3996\443973476.py:1: FutureWarning:

The default value of regex will change from True to False in a future version. In addition, single character regular expressions will *not* be treated as literal strings when regex=True.
```

In [36]:
```
df['New_Rotten_tomatoes']
```

Out[36]:
```
0        87%
1        87%
2        84%
3        96%
4        97%
        ...
16739    NaN
16740    NaN
16741    NaN
16742    NaN
16743    NaN
Name: New_Rotten_tomatoes, Length: 16744, dtype: object
```

In [ ]:
```
//what is the number of movies for each age group
```

In [37]:
```
df['Age'].value_counts()
```

Out[37]:
```
18+    3474
7+     1462
13+    1255
all     843
16+     320
Name: Age, dtype: int64
```

In [ ]:
```
//top 10 languages in streaming services
```

In [40]:
```
df.Language.value_counts()
```

Out[40]:
```
English
10955
Hindi
503
English,Spanish
276
Spanish
267
English,French
174

...
English,German,Hungarian,Romanian
1
English,Spanish,Chinese,Latin
1
English,Danish,Malay,Dutch,Indonesian,Finnish,Luxembourgish,French Sign Language
1
Dutch,French
1
English,Algonquin
1
Name: Language, Length: 1102, dtype: int64
```

In [44]:
```
language = df.Language.value_counts().head(10)
language.index
```

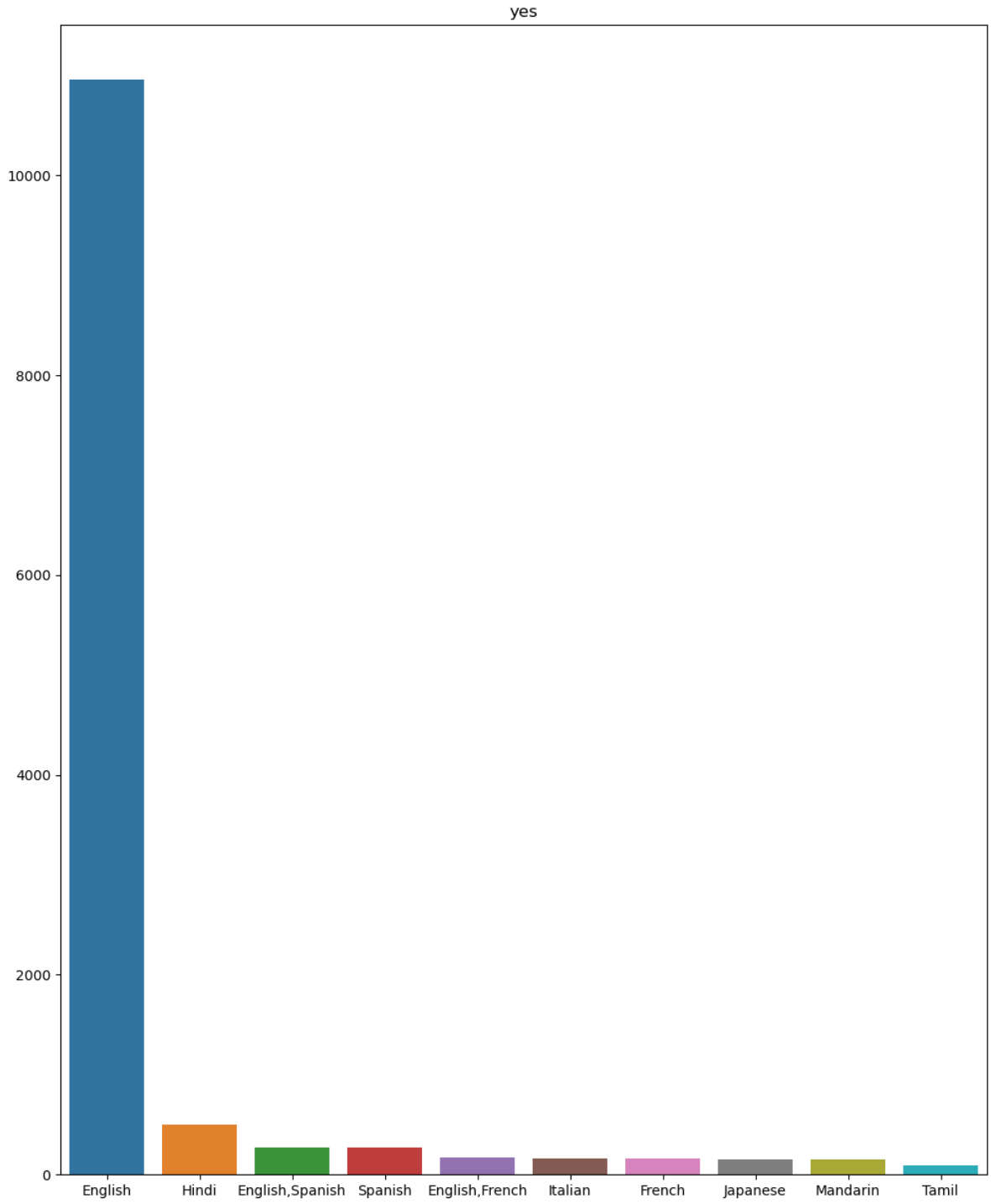
Out[44]:
```
Index(['English', 'Hindi', 'English,Spanish', 'Spanish', 'English,French',
       'Italian', 'French', 'Japanese', 'Mandarin', 'Tamil'],
      dtype='object')
```

In [45]:
```
language.values
```

Out[45]:
```
array([10955,   503,   276,   267,   174,   166,   163,   155,   151,
          93], dtype=int64)
```

In [47]:
```
language = df.Language.value_counts().head(10)
plt.figure(figsize=(12,15))
plt.title('yes')
sns.barplot(x = language.index, y = language.values)
```

Out[47]:
```
<AxesSubplot:title={'center':'yes'}>
```



In [48]:
```
from IPython.display import HTML
```

In [51]:
```
fig = px.pie(df,
             values = language.values,
             names = language.index,
             title = 'yes' , height = 600)
HTML(fig.to_html())
```

Out[51]:

In [ ]:
```
// number of movies in specific age group in all services
```

In [55]:
```
from IPython.display import HTML
fig = px.bar(df,
             x = df['Age'].value_counts().index,
             y = df['Age'].value_counts(),
             title = 'yes' , text = df['Age'].value_counts(),
             height = 600)
HTML(fig.to_html())
```

Out[55]:

In [ ]:
```
// number of movies in specific age group in all netflix
```

In [56]:
```
from IPython.display import HTML
netflix_df= df[df['Netflix']==1]
fig = px.bar(netflix_df,
             x = netflix_df['Age'].value_counts().index,
             y = netflix_df['Age'].value_counts(),
             title = 'yes' , text = netflix_df['Age'].value_counts(),
             height = 600)
HTML(fig.to_html())
```

Out[56]:

In [ ]:
```
// rotten tomato rating for each services
```

In [60]:
```
from IPython.display import HTML

fig = px.bar(df,
             x = df['Rotten Tomatoes'].value_counts().index,
             y = df['Rotten Tomatoes'].value_counts(),
             title = 'yes' , text = df['Rotten Tomatoes'].value_counts(),
             height = 600)
HTML(fig.to_html())
```

Out[60]:

In [ ]:
```
netflix_df= df[df['Netflix']==1]
prime_df= df[df['Prime Video']==1]
Hulu_df= df[df['Hulu']==1]
Disney+_df= df[df['Disney+']==1]
```

In [ ]:
```
// IMDB Rating
```

In [78]:
```
from IPython.display import HTML

fig = px.bar(df,
             x = df['IMDb'].value_counts().index,
             y = df['IMDb'].value_counts(),
             title = 'yes' , text = df['IMDb'].value_counts(),
             height = 600)
HTML(fig.to_html())
```

Out[78]:

In [80]:
```
RuntimeCount=pd.DataFrame(dict(df['Runtime'].value_counts().sort_values(ascending=
                  columns=['Runtime' , 'Count'])
RuntimeCount
```

Out[80]:

	Runtime	Count
0	90.0	971
1	95.0	489
2	92.0	434
3	93.0	422
4	85.0	408
...	...	...
152	19.0	8
153	32.0	8
154	9.0	8
155	7.0	8
156	10.0	8

157 rows × 2 columns

In [ ]: