In [80]:
```python
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns


import plotly.graph_objects as go

from termcolor import colored
#from plotly.offline import init_notebook_mode, iplo

import warnings
warnings.filterwarnings("ignore")
```

In [81]:
```python
# read dataset in jupyter notebook
dataset1 = pd.read_csv("dataset1.csv")
```

In [82]:
```python
dataset1["Netflix Release Date"]= pd.to_datetime(dataset1["Netflix Release Date"])
```

In [83]:
```python
#
dataset1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2217 entries, 0 to 2216
Data columns (total 37 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Title                  2217 non-null   object
 1   Genre                  2217 non-null   object
 2   Languages              2217 non-null   object
 3   Series or Movie        2217 non-null   object
 4   Country Availability   2217 non-null   object
 5   Runtime                2217 non-null   object
 6   Director               2217 non-null   object
 7   Writer                 2217 non-null   object
 8   Actors                 2217 non-null   object
 9   View Rating            2217 non-null   object
 10  IMDb Votes             2217 non-null   int64
 11  IMDb Score             2217 non-null   float64
 12  Rotten Tomatoes Score  2217 non-null   int64
 13  Metacritic Score       2217 non-null   int64
 14  Awards Received        2217 non-null   int64
 15  Awards Nominated For   2217 non-null   int64
 16  Boxoffice              2217 non-null   object
 17  Release Date           2217 non-null   object
 18  Netflix Release Date   2217 non-null   datetime64[ns]
 19  Production House       2217 non-null   object
 20  Netflix Release Year   2217 non-null   int64
 21  Comedy                 746 non-null    object
 22  Crime                  406 non-null    object
 23  Drama                  1225 non-null   object
 24  Fantasy                341 non-null    object
 25  Horror                 233 non-null    object
 26  Romance                453 non-null    object
 27  Thriller               683 non-null    object
 28  Action                 645 non-null    object
 29  Adventure              511 non-null    object
 30  Sci-Fi                 321 non-null    object
 31  Mystery                265 non-null    object
 32  Biography              226 non-null    object
 33  History                116 non-null    object
 34  Movie                  2170 non-null   object
 35  Series                 0 non-null      float64
 36  Director_name          2217 non-null   object
dtypes: datetime64[ns](1), float64(2), int64(6), object(28)
memory usage: 641.0+ KB
```

In [84]:
```python
dataset1["Netflix Release Date"].dt.year
```

Out[84]:
```
0       2021
1       2021
2       2021
3       2021
4       2021
        ...
2212    2015
2213    2015
2214    2015
2215    2015
2216    2015
Name: Netflix Release Date, Length: 2217, dtype: int64
```

dataset1["Netflix Release Year"]=dataset["Netflix Release Date"].dt.year

In [85]:
```python
dataset1.isnull().sum()
```

Out[85]:
```
Title                      0
Genre                      0
Languages                  0
Series or Movie            0
Country Availability       0
Runtime                    0
Director                   0
Writer                     0
Actors                     0
View Rating                0
IMDb Votes                 0
IMDb Score                 0
Rotten Tomatoes Score      0
Metacritic Score           0
Awards Received            0
Awards Nominated For       0
Boxoffice                  0
Release Date               0
Netflix Release Date       0
Production House           0
Netflix Release Year       0
Comedy                  1471
Crime                   1811
Drama                    992
Fantasy                 1876
Horror                  1984
Romance                 1764
Thriller                1534
Action                  1572
Adventure               1706
Sci-Fi                  1896
Mystery                 1952
Biography               1991
History                 2101
Movie                     47
Series                  2217
Director_name              0
dtype: int64
```

dataset1.shape

In [86]:
```python
dataset1
```

Out[86]:

	Title	Genre	Languages	Series or Movie	Country Availability	Runtime	Director	Writer	Actors	View Rating	...	Thriller	Action	Adventure	Sci-Fi	Mystery	Biography	History	Movie	Series
0	Lets Fight Ghost	Crime, Drama, Fantasy, Horror, Romance	Swedish, Spanish	Series	Thailand	< 30 minutes	Tomas Alfredson	John Ajvide Lindqvist	Kåre Hedebrant, Per Ragnar, Lina Leandersson, ...	R	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Joker	Crime, Drama, Thriller	English	Movie	Lithuania,Poland,France,Italy,Spain,Greece,Bel...	1-2 hour	Todd Phillips	Scott Silver, Jerry Robinson, Todd Phillips, B...	Joaquin Phoenix, Zazie Beetz, Robert De Niro, ...	R	...	Thriller	NaN	NaN	NaN	NaN	NaN	NaN	Movie	NaN
2	I	Action, Adventure, Fantasy, Sci-Fi	English, Sanskrit	Movie	Lithuania,Poland,France,Italy,Spain,Greece,Cze...	1-2 hour	George Lucas	George Lucas	Liam Neeson, Ewan McGregor, Natalie Portman, J...	PG	...	NaN	Action	Adventure	Sci-Fi	NaN	NaN	NaN	Movie	NaN
3	Harrys Daughters	Adventure, Drama, Fantasy, Mystery	English	Movie	Lithuania,Poland,France,Italy,Spain,Greece,Cze...	1-2 hour	David Yates	Steve Kloves, J.K. Rowling	Ralph Fiennes, Alan Rickman, Michael Gambon, D...	PG-13	...	NaN	NaN	Adventure	NaN	Mystery	NaN	NaN	Movie	NaN
4	The Closet	Comedy	French	Movie	South Korea	1-2 hour	Francis Veber	Francis Veber	Gérard Depardieu, Michèle Laroque, Daniel Aute...	R	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Movie	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2212	50 First Dates	Comedy, Drama, Romance	English, Hawaiian, Mandarin	Movie	Hong Kong,South Korea,Japan,United Kingdom,Tha...	1-2 hour	Peter Segal	George Wing	Sean Astin, Adam Sandler, Rob Schneider, Drew ...	PG-13	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Movie	NaN
2213	21	Crime, Drama, History, Thriller	English	Movie	South Korea,Russia,Hong Kong,Portugal,Belgium,...	> 2 hrs	Robert Luketic	Allan Loeb, Ben Mezrich, Peter Steinfeld	Kevin Spacey, Aaron Yoo, Kate Bosworth, Jim St...	PG-13	...	Thriller	NaN	NaN	NaN	NaN	NaN	History	Movie	NaN
2214	One Chance	Biography, Comedy, Drama, Music	English, Italian	Movie	Japan,Malaysia	1-2 hour	David Frankel	Justin Zackham	James Corden, Colm Meaney, Alexandra Roach, Ju...	PG-13	...	NaN	NaN	NaN	NaN	NaN	Biography	NaN	Movie	NaN
2215	The Twilight Saga: Breaking Dawn: Part 1	Adventure, Drama, Fantasy, Romance, Thriller	English, Portuguese	Movie	Canada,Romania,Japan,Switzerland,United Kingdo...	> 2 hrs	Bill Condon	Stephenie Meyer, Melissa Rosenberg	Billy Burke, Taylor Lautner, Sarah Clarke, Gil...	PG-13	...	Thriller	NaN	Adventure	NaN	NaN	NaN	NaN	Movie	NaN
2216	One for the Money	Action, Comedy, Crime, Thriller	English	Movie	Canada	1-2 hour	Julie Anne Robinson	Liz Brixius, Karen Ray, Janet Evanovich, Stacy...	John Leguizamo, Jason O'Mara, Katherine Heigl,...	PG-13	...	Thriller	Action	NaN	NaN	NaN	NaN	NaN	Movie	NaN

2217 rows × 37 columns

In [87]:
```python
dataset1.duplicated().sum()
```

Out[87]: 0

In [88]:
```python
# to count how many type of shows
dataset1['Series or Movie'].value_counts()
```

Out[88]:
```
Movie     2170
Series      47
Name: Series or Movie, dtype: int64
```

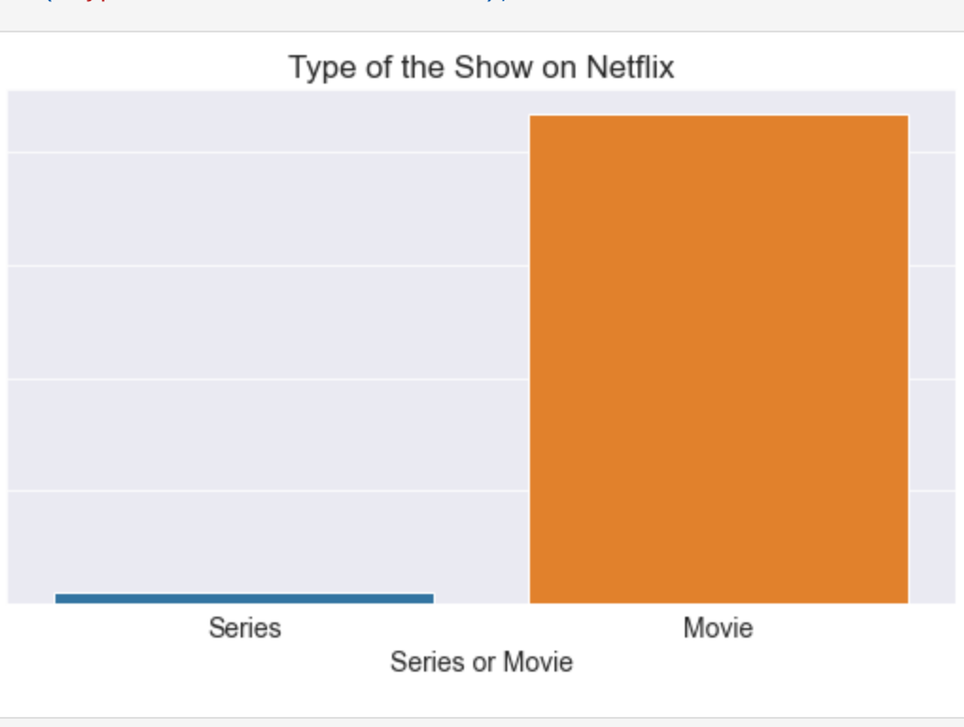
In [89]:
```python
#count listed_in shows
dataset1['Genre'].value_counts()
```

Out[89]:
```
Drama                                                       99
Comedy, Drama, Romance                                      83
Drama, Romance                                              78
Comedy                                                      65
Comedy, Romance                                             63
                                                            ..
Adventure, Drama, Western                                    1
Action, Adventure, Biography, Drama, History                 1
Animation, Adventure, Comedy, Family, Music, Musical, Romance 1
Adventure, Family, Fantasy, Musical                          1
Biography, Crime, Drama, Thriller, War                       1
Name: Genre, Length: 563, dtype: int64
```

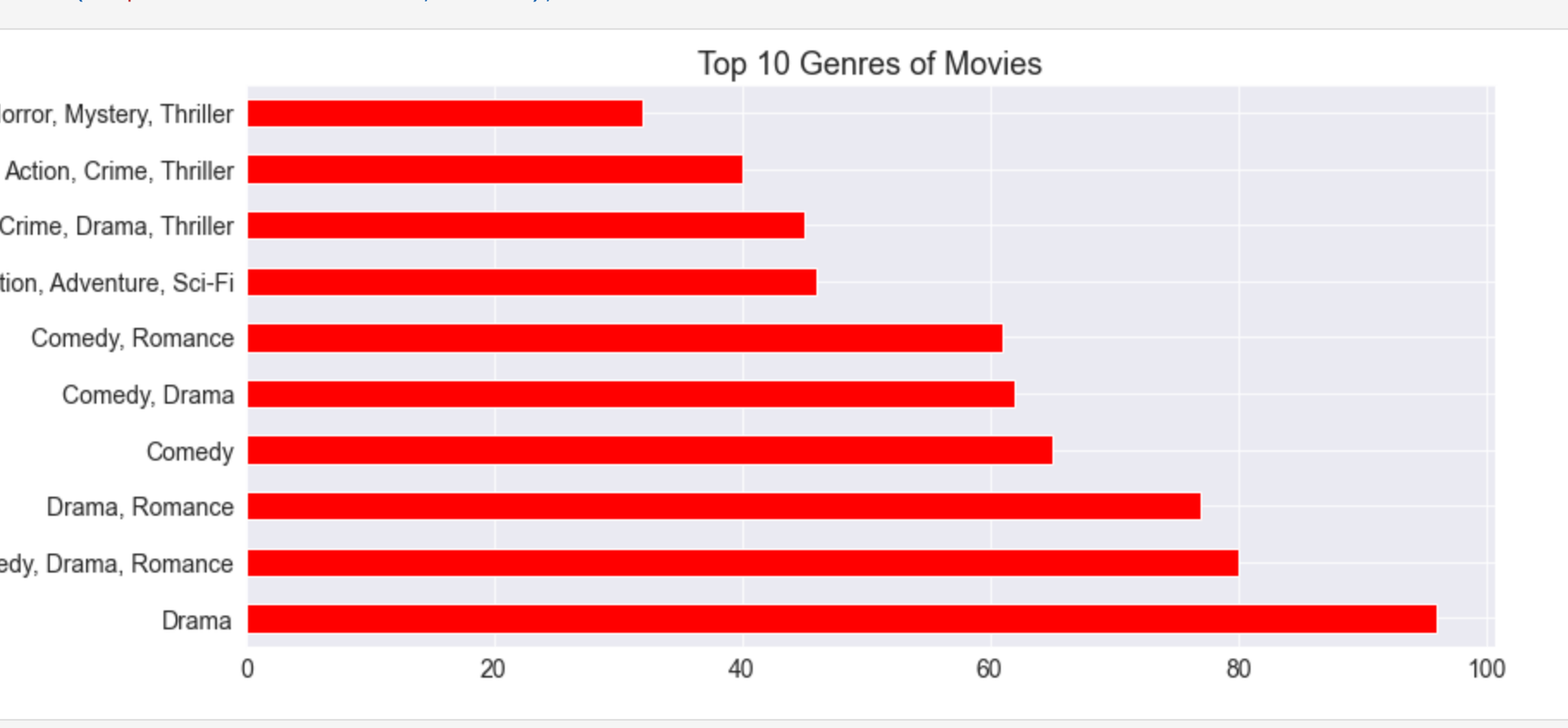
In [90]:
```python
#import libs for  data visulaisation
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9, 5)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

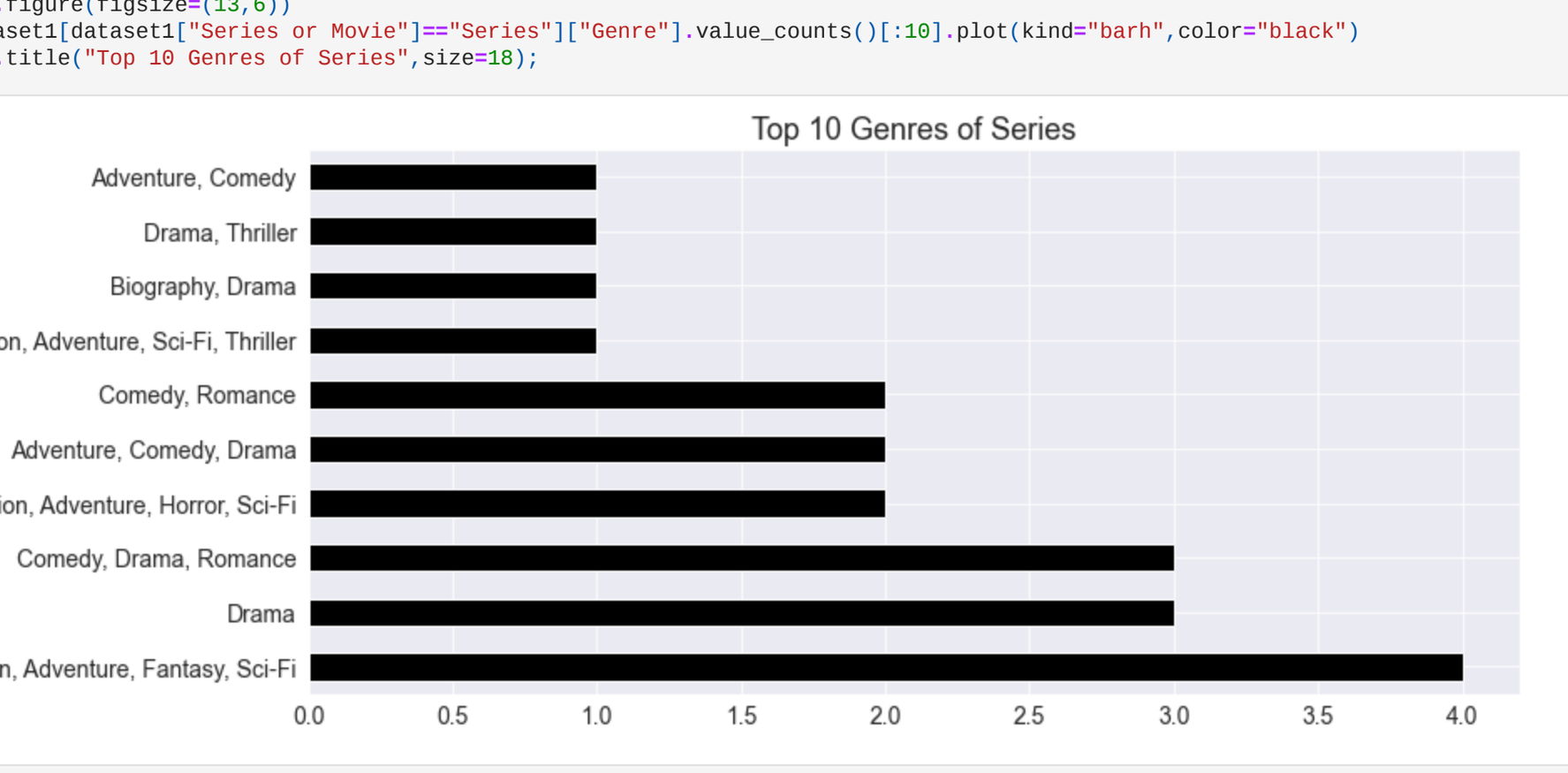
In [91]:
```python
#Firstly the difference in types of shows available was analysed
sns.countplot(x='Series or Movie', data=dataset1)
plt.title("Type of the Show on Netflix");
```



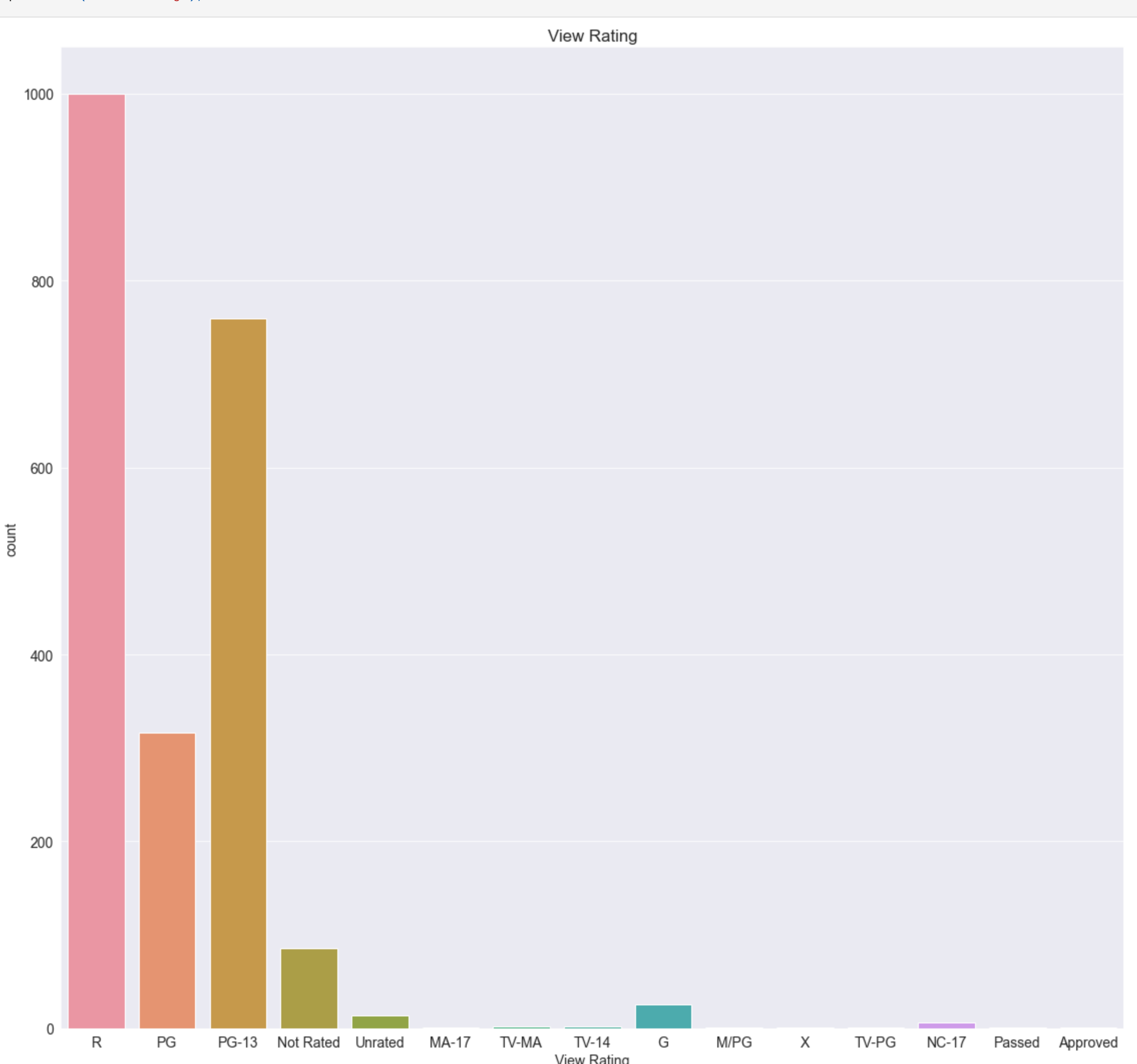
In [92]:
```python
#We will now look at what Genres of movie account for the most on Netflix using a bar graph.
plt.figure(figsize=(13,6))
dataset1[dataset1["Series or Movie"]=="Movie"]["Genre"].value_counts()[:10].plot(kind="barh",color="red")
plt.title("Top 10 Genres of Movies",size=18);
```



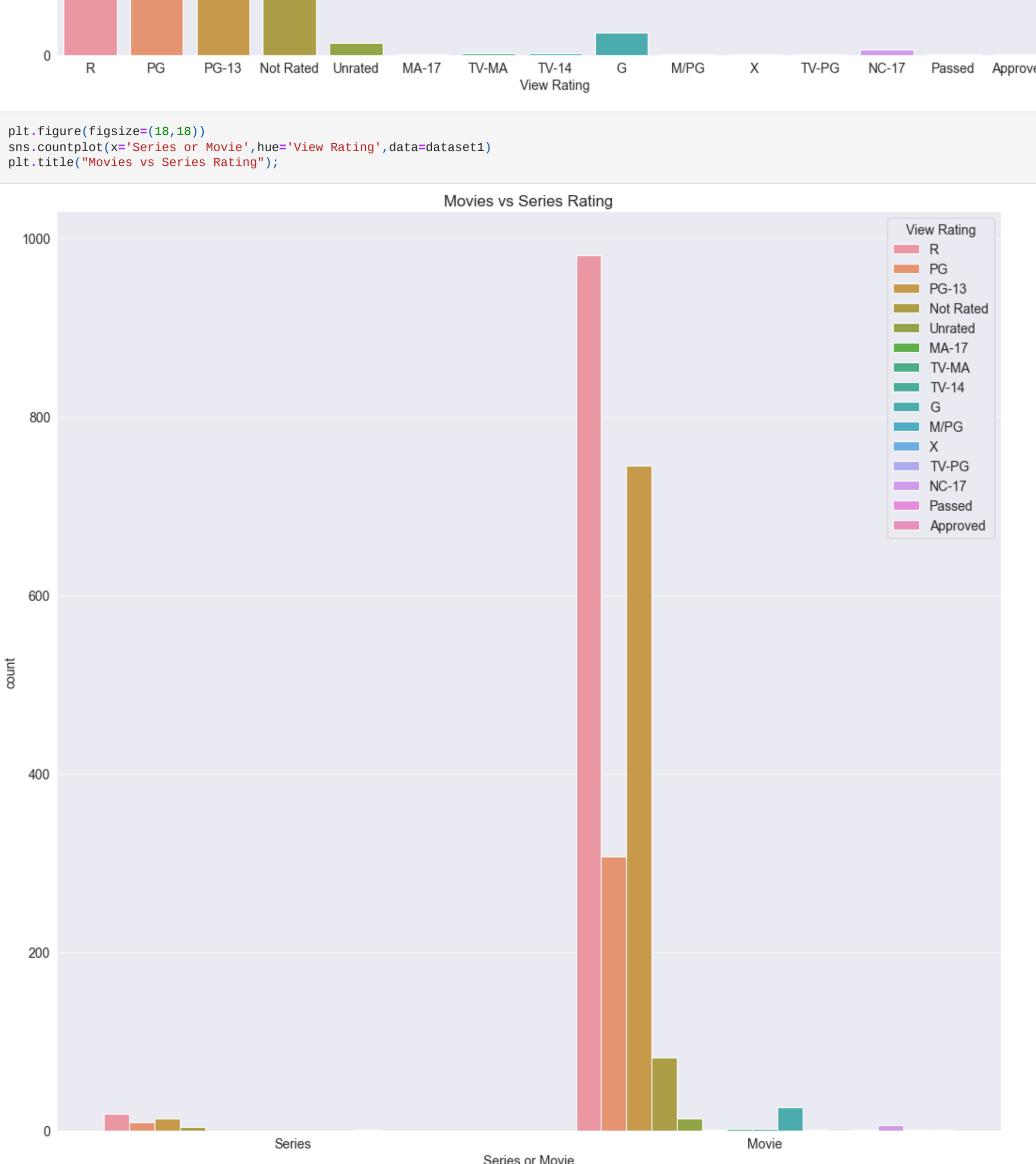
In [93]:
```python
#We will now look at what Genres of movie account for the most on Netflix using a bar graph.
plt.figure(figsize=(13,6))
dataset1[dataset1["Series or Movie"]=="Series"]["Genre"].value_counts()[:10].plot(kind="barh",color="black")
plt.title("Top 10 Genres of Series",size=18);
```



In [94]:
```python
#Types of shows by ratings
plt.figure(figsize=(19,18))
sns.countplot(x='View Rating',data=dataset1)
plt.title("View Rating");
```

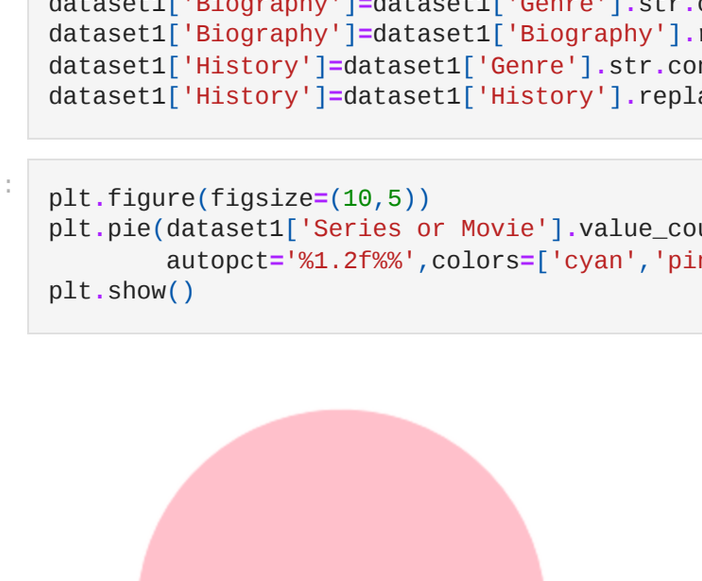


In [95]:
```python
plt.figure(figsize=(18,18))
sns.countplot(x='Series or Movie',hue='View Rating',data=dataset1)
plt.title("Movies vs Series Rating");
```



In [96]:
```python
dataset1['Comedy']=dataset1['Genre'].str.contains('Comedy')
dataset1['Comedy']=dataset1['Comedy'].replace({True:'Comedy',False:'NaN',})
dataset1['Crime']=dataset1['Genre'].str.contains('Crime')
dataset1['Crime']=dataset1['Crime'].replace({True:'Crime',False:'NaN',})
dataset1['Drama']=dataset1['Genre'].str.contains('Drama')
dataset1['Drama']=dataset1['Drama'].replace({True:'Drama',False:'NaN',})
dataset1['Fantasy']=dataset1['Genre'].str.contains('Drama')
dataset1['Fantasy']=dataset1['Fantasy'].replace({True:'Fantasy',False:'NaN',})
dataset1['Horror']=dataset1['Genre'].str.contains('Horror')
dataset1['Horror']=dataset1['Horror'].replace({True:'Horror',False:'NaN',})
dataset1['Romance']=dataset1['Genre'].str.contains('Romance')
dataset1['Romance']=dataset1['Romance'].replace({True:'Romance',False:'NaN',})
dataset1['Thriller']=dataset1['Genre'].str.contains('Thriller')
dataset1['Thriller']=dataset1['Thriller'].replace({True:'Thriller',False:'NaN',})
dataset1['Action']=dataset1['Genre'].str.contains('Action')
dataset1['Action']=dataset1['Action'].replace({True:'Action',False:'NaN',})
dataset1['Adventure']=dataset1['Genre'].str.contains('Adventure')
dataset1['Adventure']=dataset1['Adventure'].replace({True:'Adventure',False:'NaN',})
dataset1['Fantasy']=dataset1['Genre'].str.contains('Fantasy')
dataset1['Fantasy']=dataset1['Fantasy'].replace({True:'Fantasy',False:'NaN',})
dataset1['Sci-Fi']=dataset1['Genre'].str.contains('Sci-Fi')
dataset1['Sci-Fi']=dataset1['Sci-Fi'].replace({True:'Sci-Fi',False:'NaN',})
dataset1['Mystery']=dataset1['Genre'].str.contains('Mystery')
dataset1['Mystery']=dataset1['Mystery'].replace({True:'Mystery',False:'NaN',})
dataset1['Biography']=dataset1['Genre'].str.contains('Biography')
dataset1['Biography']=dataset1['Biography'].replace({True:'Biography',False:'NaN',})
dataset1['History']=dataset1['Genre'].str.contains('History')
dataset1['History']=dataset1['History'].replace({True:'History',False:'NaN',})
```

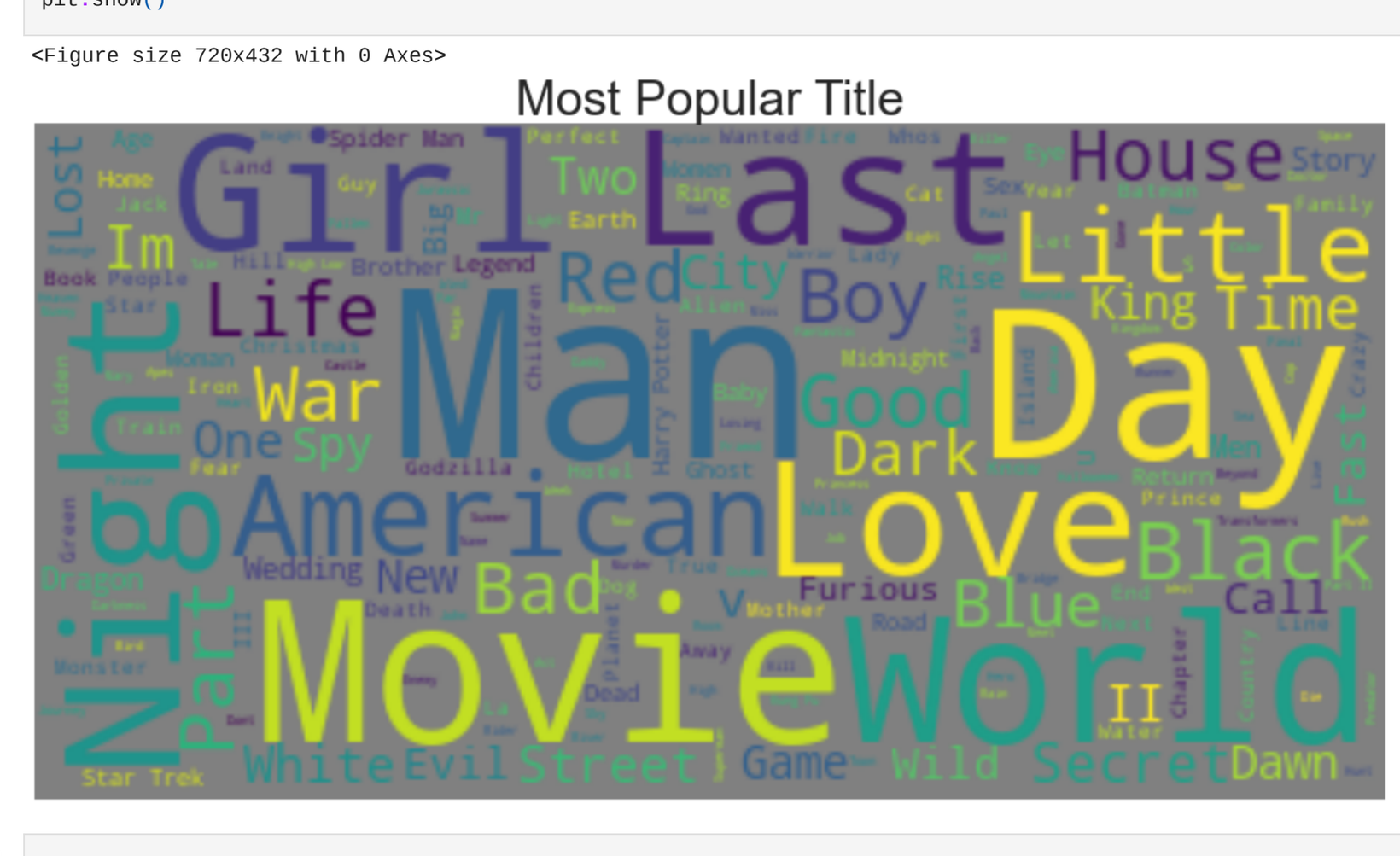
In [97]:
```python
plt.figure(figsize=(10,5))
plt.pie(dataset1['Series or Movie'].value_counts().sort_values(),labels=dataset1['Series or Movie'].value_counts().index,explode=[0.05,0],
        autopct='%1.2f%%',colors=['cyan','pink'])
plt.show()
```



In [98]:
```python
from wordcloud import WordCloud, ImageColorGenerator
```

In [99]:
```python
text = " ".join(str(each) for each in dataset1.Title)
# Create and generate a word cloud image:
wordcloud = WordCloud(max_words=200, background_color="gray").generate(text)
plt.figure(figsize=(10,6))
plt.figure(figsize=(15,10))
# Display the generated image:
plt.imshow(wordcloud, interpolation='Bilinear')
plt.title('Most Popular Title',fontsize = 30)
plt.axis("off")
plt.show()
```

<Figure size 720x432 with 0 Axes>



In [100]:
```python
order =  ['G','PG','PG-13','Not Rated','Unrated','PG-13','PG-13','TV-14','R','NC-17','TV-MA']
plt.figure(figsize=(15,7))
g = sns.countplot(dataset1["View Rating"], hue= dataset1["Series or Movie"] , order=order, palette="pastel");
plt.title("Ratings for Movies & Series")
plt.xlabel("Rating")
plt.ylabel("Total Count")
plt.show()
```

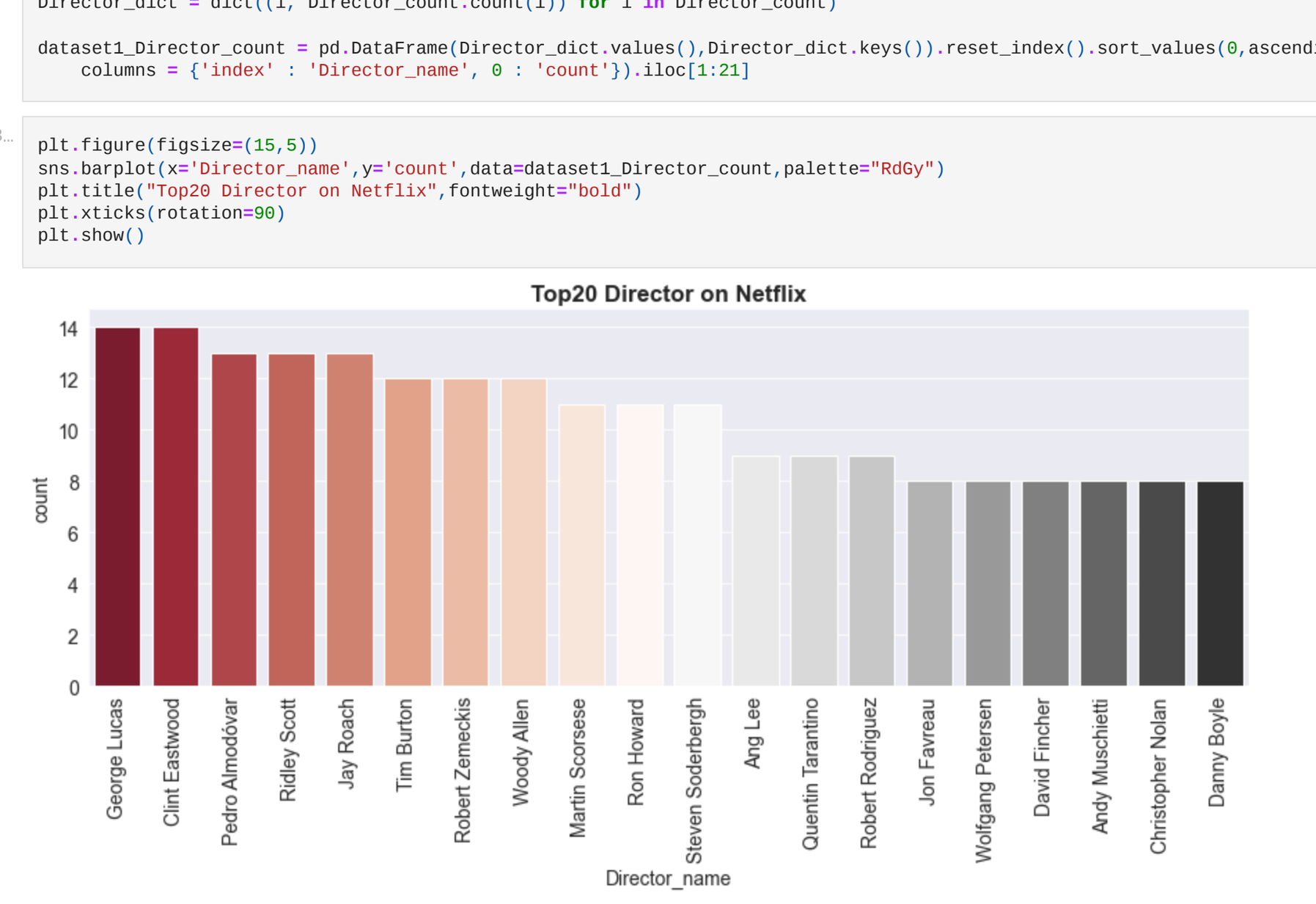
In [101]:
```python
fig, ax = plt.subplots(1,2, figsize=(19, 5))
g1 = sns.countplot(dataset1["View Rating"], order=order,palette="Set2", ax=ax[0]);
g1.set_title("Ratings for Movies")
g1.set_xlabel("Rating")
g1.set_ylabel("Total Count")
g2 = sns.countplot(dataset1["View Rating"], order=order,palette="Set2", ax=ax[1]);
g2.set(yticks=np.arange(0,1600,200))
g2.set_title("Ratings for Series")
g2.set_xlabel("Rating")
g2.set_ylabel("Total Count")
fig.show()
```

In [102]:
```python
dataset1['Director_name'] = dataset1['Director'].apply(lambda x :  x.replace(' ,',',').replace(', ',',').split(','))
Director_count = []
for i in dataset1['Director_name']: Director_count += i

Director_dict = dict((i, Director_count.count(i)) for i in Director_count)

dataset1_Director_count = pd.DataFrame(Director_dict.values(),Director_dict.keys()).reset_index().sort_values(0,ascending=False).rename(
    columns = {'index' : 'Director_name', 0 : 'count'}).iloc[1:21]
```

In [103]:
```python
plt.figure(figsize=(15,5))
sns.barplot(x='Director_name',y='count',data=dataset1_Director_count,palette="RdGy")
plt.title("Top20 Director on Netflix",fontweight="bold")
plt.xticks(rotation=90)
plt.show()
```



In [104]:
```python
dataset1_series1 = dataset1[dataset1["Series or Movie"] == "Series"]
dataset1_movies1 = dataset1[dataset1["Series or Movie"] == "Movie"]
```

In [105]:
```python
dataset1_content = dataset1['Netflix Release Year'].value_counts().reset_index().rename(columns = {
    'Netflix Release Year' : 'count', 'index' : 'Netflix Release Year'}).sort_values('Netflix Release Year')
dataset1_content['percent'] = dataset1_content['count'].apply(lambda x : 100*x/sum(dataset1_content['count']))


dataset1_series1 = dataset1_series1['Netflix Release Year'].value_counts().reset_index().rename(columns = {
    'Netflix Release Year' : 'count', 'index' : 'Netflix Release Year'}).sort_values('Netflix Release Year')
dataset1_series1['percent'] = dataset1_series1['count'].apply(lambda x : 100*x/sum(dataset1_series1['count']))


dataset1_movies1 = dataset1_movies1['Netflix Release Year'].value_counts().reset_index().rename(columns = {
    'Netflix Release Year' : 'count', 'index' : 'Netflix Release Year'}).sort_values('Netflix Release Year')
dataset1_movies1['percent'] = dataset1_movies1['count'].apply(lambda x : 100*x/sum(dataset1_movies1['count']))

t1 = go.Scatter(x=dataset1_movies1['Netflix Release Year'], y=dataset1_movies1["count"], name="Movies", marker=dict(color="#a678de"))
t2 = go.Scatter(x=dataset1_series1['Netflix Release Year'], y=dataset1_series1["count"], name="Series", marker=dict(color="#6ad49b"))
t3 = go.Scatter(x=dataset1_content['Netflix Release Year'], y=dataset1_content["count"], name="Total Contents", marker=dict(color="brown"))

data = [t1, t2, t3]

layout = go.Layout(title="Content added over the years", legend=dict(x=0.1, y=1.1, orientation="h"))
fig = go.Figure(data, layout=layout)
fig.show()
```

In [106]:
```python
import plotly.graph_objects as go
from plotly.offline import init_notebook_mode, iplot
```

In [107]:
```python
filtered_countries = dataset1.set_index('Title')["Country Availability"].str.split(', ', expand=True).stack().reset_index(level=1, drop=True);
filtered_countries = filtered_countries[filtered_countries != 'Country Unavailable']

iplot([go.Choropleth(
    locationmode='country names',
    locations=filtered_countries,
    z=filtered_countries.value_counts()
)])
```

In [108]:
```python
plt.figure(figsize=(15,5))
sns.barplot(x = dataset1["Runtime"].value_counts().head(10).index,
          y = dataset1["Runtime"].value_counts().head(10).values,palette="RdGy")
plt.xticks(rotation=80)
plt.title("Based on Runtime",fontweight="bold")
plt.show()
```