In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

In [85]:
```
df = pd.read_csv(r'D:\data science\Data analyst\Project 2\netflix.csv')
df.head(9)
```

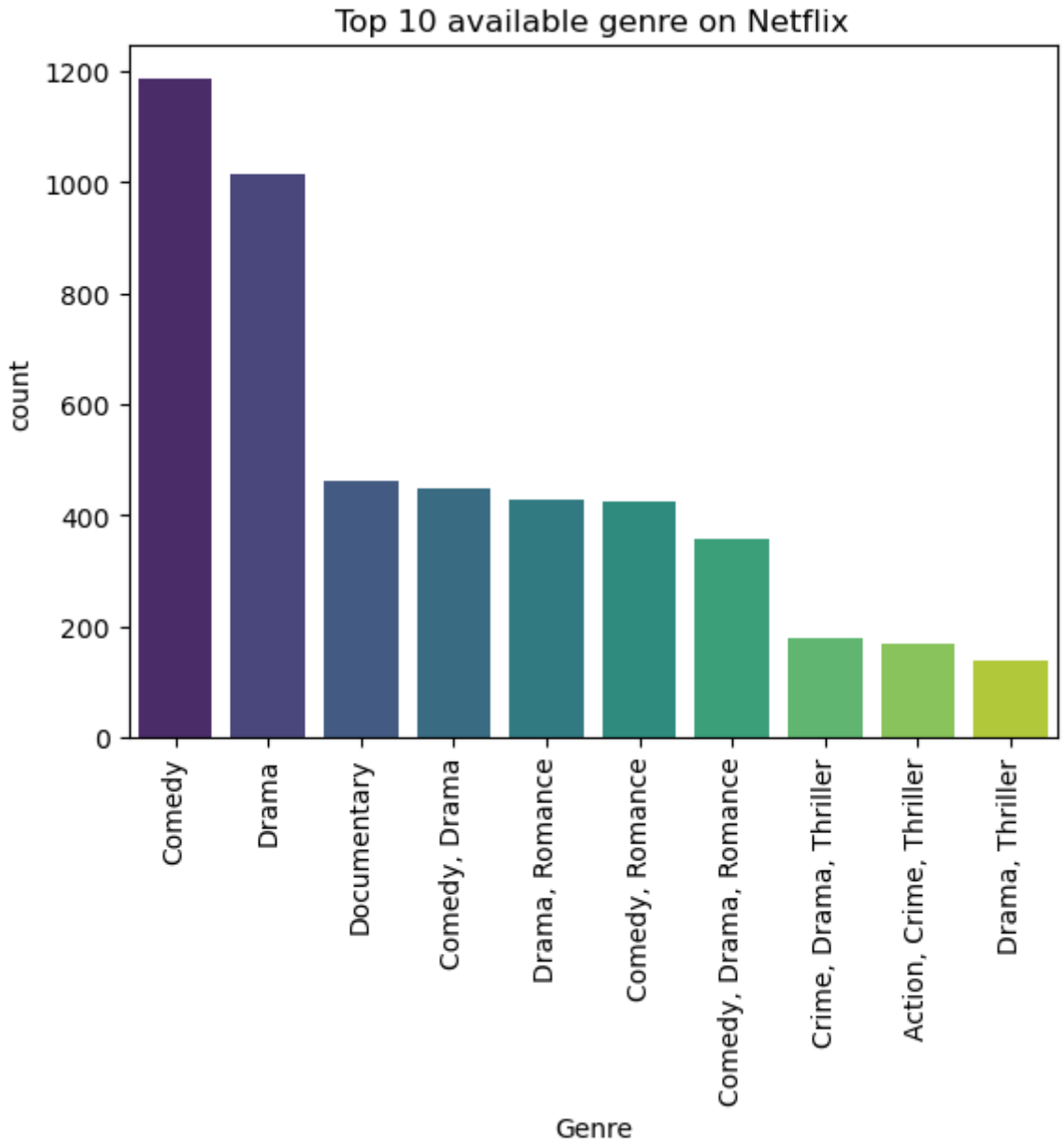
Out[85]:

	Title	Genre	Tags	Languages	Series or Movie	Hidden Gem Score	
0	Lets Fight Ghost	Crime, Drama, Fantasy, Horror, Romance	Comedy Programmes,Romantic TV Comedies,Horror ...	Swedish, Spanish	Series	4.3	
1	HOW TO BUILD A GIRL	Comedy	Dramas,Comedies,Films Based on Books,British	English	Movie	7.0	
2	Centigrade	Drama, Thriller	Thrillers	English	Movie	6.4	
3	ANNE+	Drama	TV Dramas,Romantic TV Dramas,Dutch TV Shows	Turkish	Series	7.7	
4	Moxie	Animation, Short, Drama	Social Issue Dramas,Teen Movies,Dramas,Comedie...	English	Movie	8.1	Lithuania,Poland
5	The Con-Heartist	Comedy, Romance	Romantic Comedies,Comedies,Romantic Films,Thai...	Thai	Movie	8.6	
6	Gleboka woda	Drama	TV Dramas,Polish TV Shows,Social Issue TV Dramas	Polish	Series	8.7	
7	Instynkt	Crime	TV Dramas,Crime TV Dramas,Polish TV Shows	Polish	Series	6.9	
8	Only a Mother	Drama	Social Issue Dramas,Dramas,Movies Based on Boo...	Swedish	Movie	8.3	Lithuania,Poland,

9 rows × 29 columns

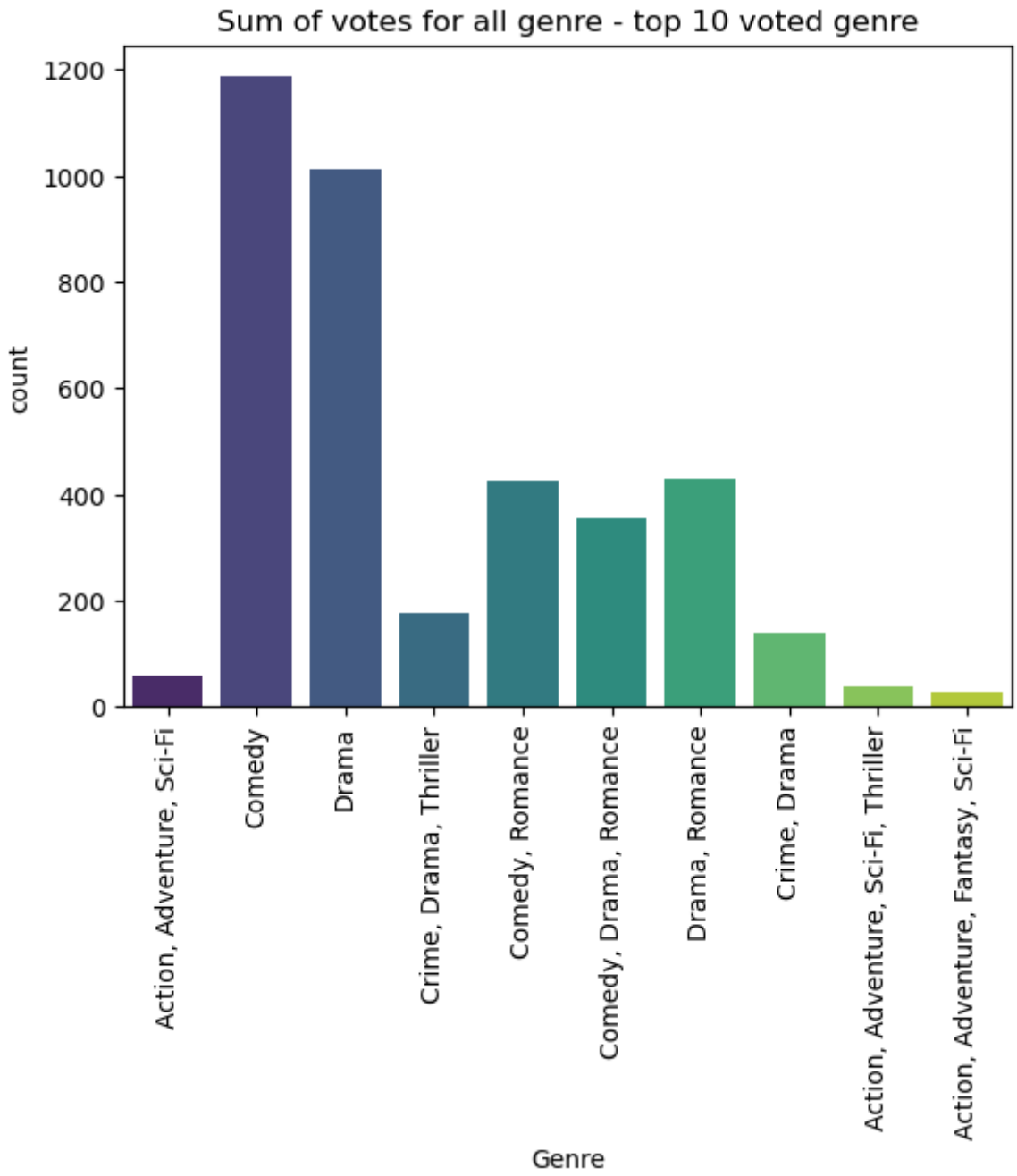
In [40]:
```
# Top 10 available genre
top = df['Genre'].value_counts().head(10)
sns.countplot(data = df,x = 'Genre',order = top.index,palette = 'viridis')
plt.xticks(rotation=90)
plt.title("Top 10 available genre on Netflix")
```

Out[40]:
```
Text(0.5, 1.0, 'Top 10 available genre on Netflix')
```



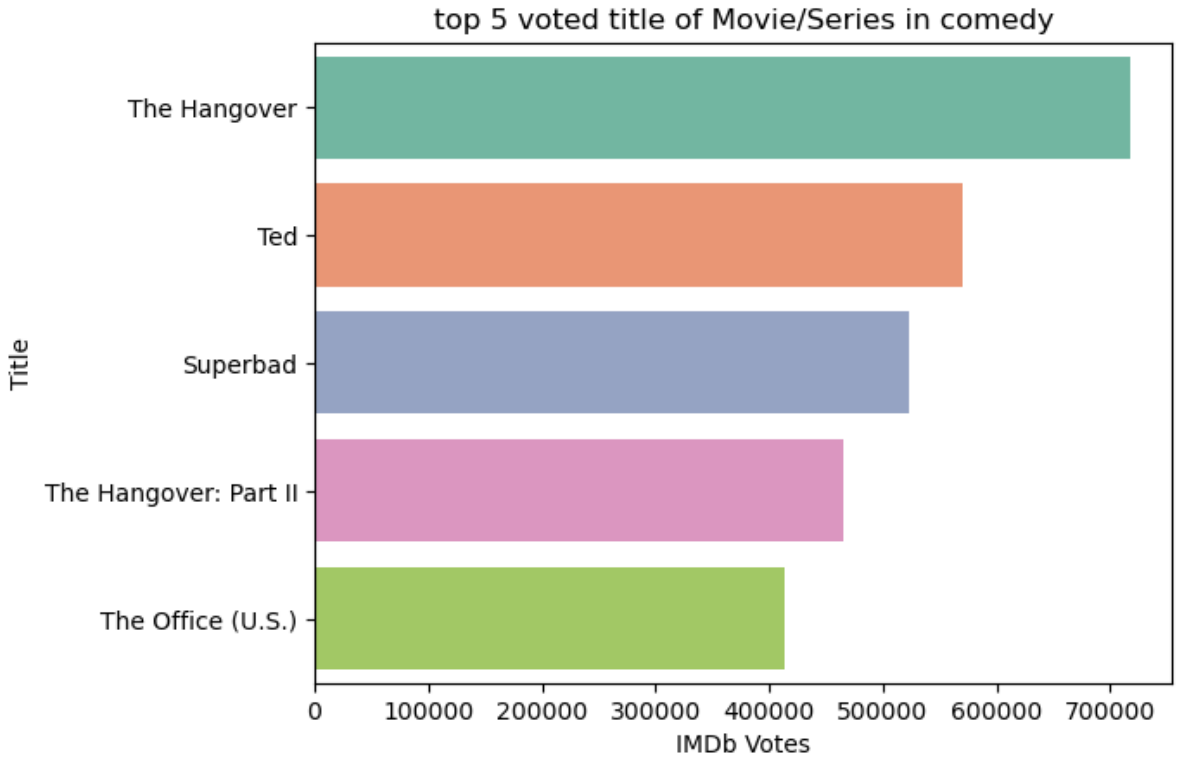
In [42]:
```
#sum of votes for all genre - top 10 voted genre

topgen = df.groupby('Genre')['IMDb Votes'].sum().nlargest(10).index

topgen2 = df[df['Genre'].isin(topgen)]

sns.countplot(data = topgen2,x = 'Genre',order = topgen,palette='viridis')
plt.xticks(rotation = 90)

plt.title("Sum of votes for all genre - top 10 voted genre")
```

Out[42]:
```
Text(0.5, 1.0, 'Sum of votes for all genre - top 10 voted genre')
```



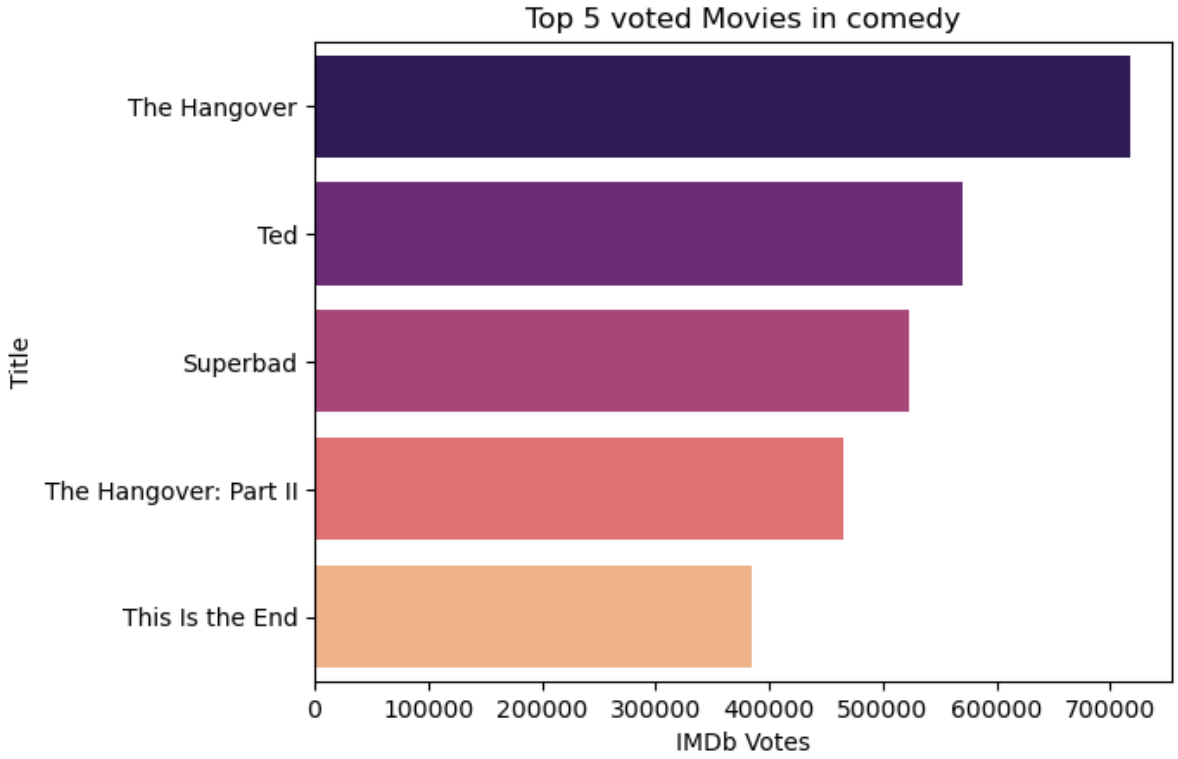
In [78]:
```
#top 5 voted title of Movie/Series

comedydf = df[df['Genre'] == 'Comedy']

comedysort = comedydf.sort_values(by='IMDb Votes',ascending = False)

top5 = comedysort.head(5)


sns.barplot(data = top5,x = 'IMDb Votes',y = 'Title',palette='Set2')

plt.title("top 5 voted title of Movie/Series in comedy")
```

Out[78]:
```
Text(0.5, 1.0, 'top 5 voted title of Movie/Series in comedy')
```



In [60]:
```
cdf = df[df['Genre'] == 'Comedy']
movdf = cdf[cdf['Series or Movie'] == 'Movie']
sorteddf = movdf.sort_values(by = 'IMDb Votes',ascending = False)
top5 = sorteddf.head(5)
sns.barplot(data = top5,x = 'IMDb Votes',y = 'Title',palette='magma')

plt.title("Top 5 voted Movies in comedy")
```

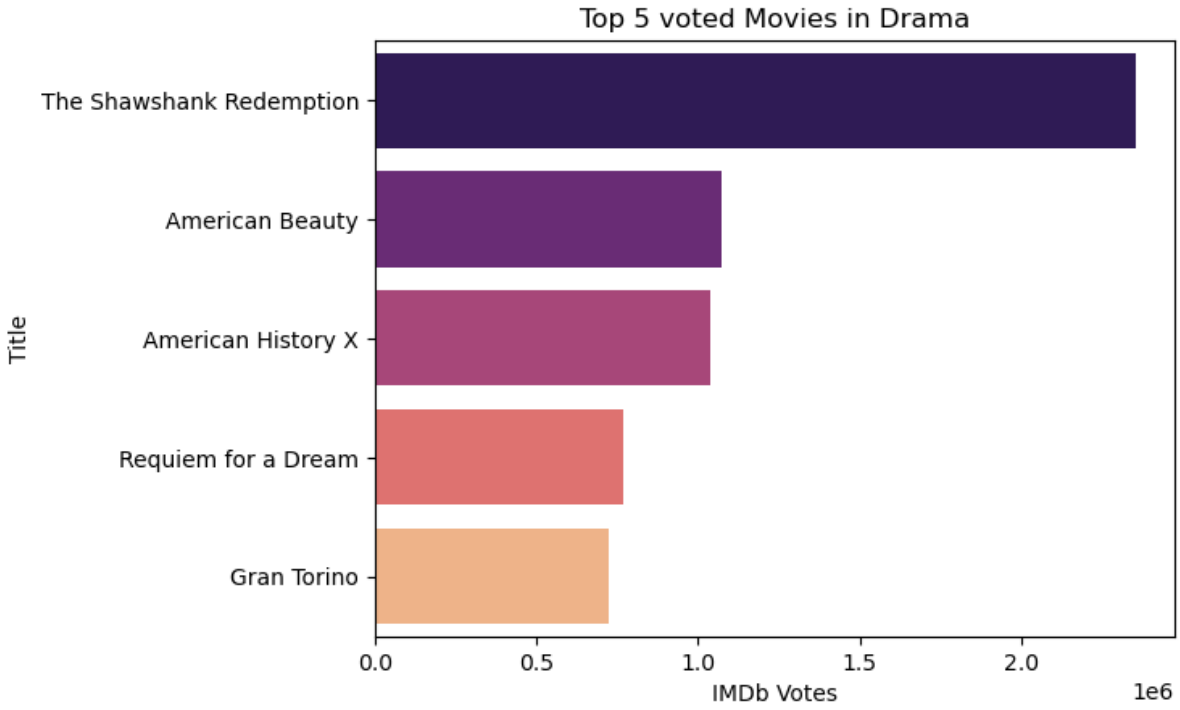
Out[60]:
```
Text(0.5, 1.0, 'Top 5 voted Movies in comedy')
```



In [77]:
```
cdf = df[df['Genre'] == 'Drama']
movdf = cdf[cdf['Series or Movie'] == 'Movie']
sorteddf = movdf.sort_values(by = 'IMDb Votes',ascending = False)
top5 = sorteddf.head(5)
```

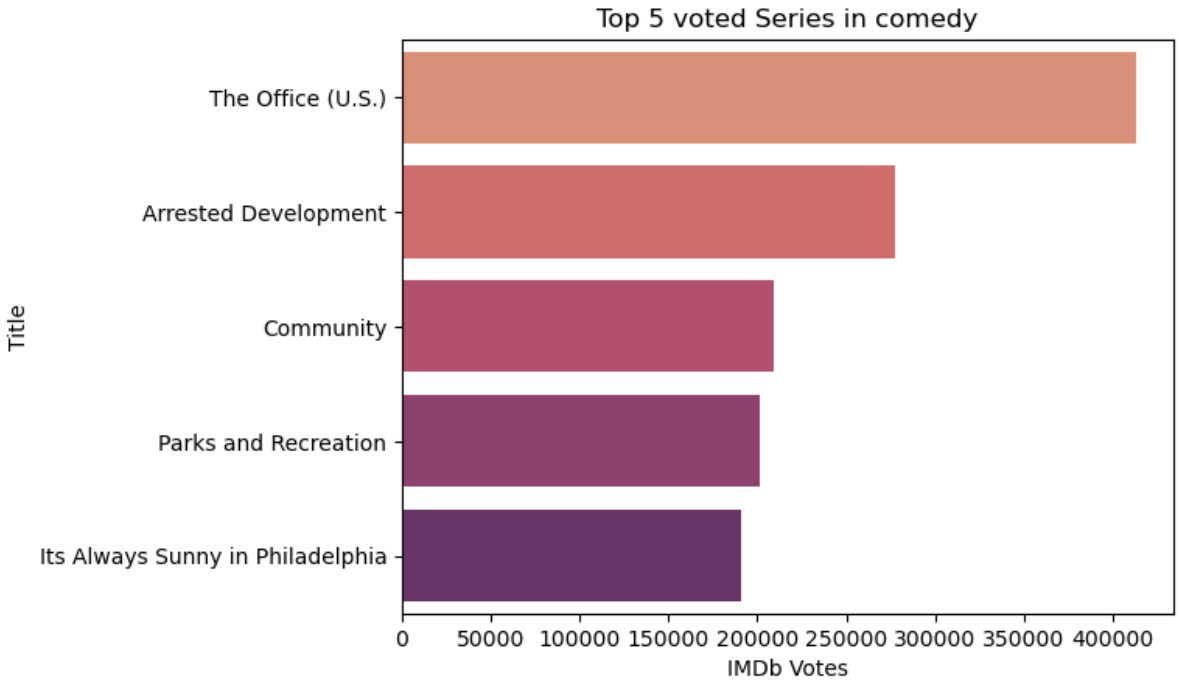
```
sns.barplot(data = top5,x = 'IMDb Votes',y = 'Title',palette='magma')

plt.title("Top 5 voted Movies in Drama")
```

Out[77]:
```
Text(0.5, 1.0, 'Top 5 voted Movies in Drama')
```



In [75]:
```
cdf = df[df['Genre'] == 'Comedy']
movdf = cdf[cdf['Series or Movie'] == 'Series']
sorteddf = movdf.sort_values(by = 'IMDb Votes',ascending = False)
top5 = sorteddf.head(5)
sns.barplot(data = top5,x = 'IMDb Votes',y = 'Title',palette='flare')

plt.title("Top 5 voted Series in comedy")
```

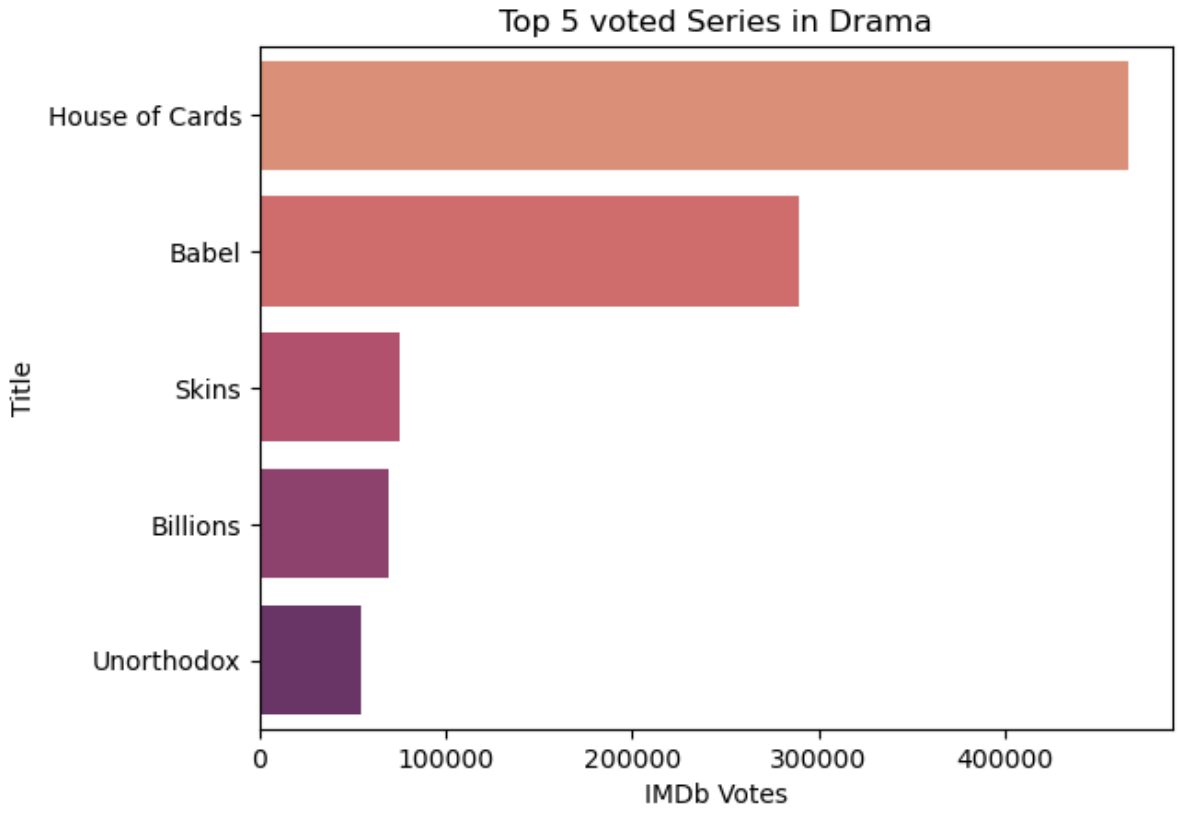
Out[75]:
```
Text(0.5, 1.0, 'Top 5 voted Series in comedy')
```



In [79]:
```
cdf = df[df['Genre'] == 'Drama']
movdf = cdf[cdf['Series or Movie'] == 'Series']
```

```
sorteddf = movdf.sort_values(by = 'IMDb Votes',ascending = False)
top5 = sorteddf.head(5)
sns.barplot(data = top5,x = 'IMDb Votes',y = 'Title',palette='flare')

plt.title("Top 5 voted Series in Drama")
```

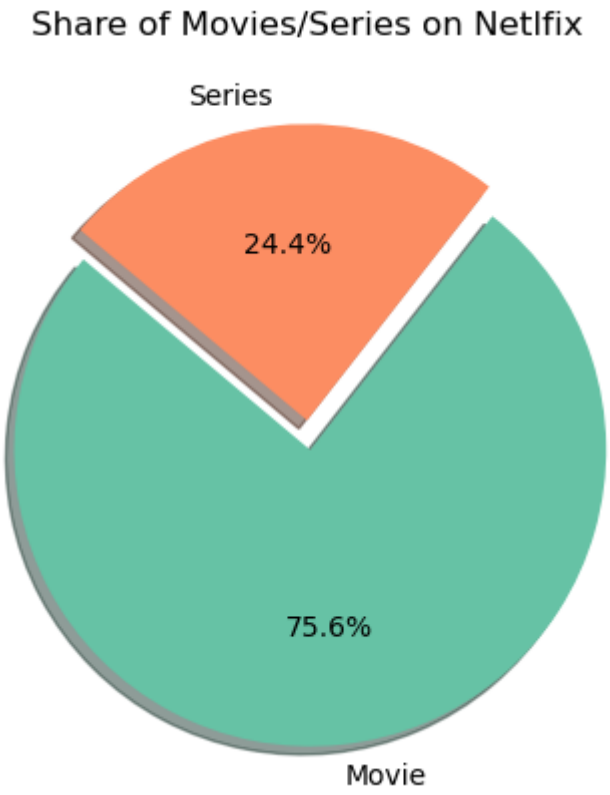
Out[79]:
```
Text(0.5, 1.0, 'Top 5 voted Series in Drama')
```



In [74]:
```
typecount = df['Series or Movie'].value_counts()
explode = (0.1, 0)
colors = ['#66c2a5', '#fc8d62']
plt.pie(typecount,labels = typecount.index, autopct = "%1.1f%%",startangle=140, exp
print(typecount)
plt.title("Share of Movies/Series on Netlfix")
```

```
Series or Movie
Movie     11697
Series     3783
Name: count, dtype: int64
```

Out[74]:
```
Text(0.5, 1.0, 'Share of Movies/Series on Netlfix')
```



In [ ]: