In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from IPython import get_ipython
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
movies_data = pd.read_csv("movies.csv")
```

In [3]:

```
movies_data.head()
```

Out[3]:

	Unnamed: 0	id	title	overview	release_date	popularity	vote_average	vote_cou
0	0	278	The Shawshank Redemption	Framed in the 1940s for the double murder of h...	23-09-1994	62.636	8.7	214
1	1	19404	Dilwale Dulhania Le Jayenge	Raj is a rich, carefree, happy-go-lucky second...	20-10-1995	19.097	8.7	36
2	2	238	The Godfather	Spanning the years 1945 to 1955, a chronicle o...	14-03-1972	57.656	8.7	159
3	3	424	Schindler's List	The true story of how businessman Oskar Schind...	30-11-1993	41.077	8.6	127
4	4	240	The Godfather: Part II	In the continuing saga of the Corleone crime f...	20-12-1974	46.655	8.6	96

In [4]:

```
movies_data.tail()
```

Out[4]:

	Unnamed: 0	id	title	overview	release_date	popularity	vote_average	vote_cou
8555	8555	8457	Drillbit Taylor	Three kids hire a low-budget bodyguard to prot...	04-02-2008	9.382	5.7	82
8556	8556	445583	It's All About Karma	Giacomo befriends a con man, believing that he...	09-03-2017	5.406	5.7	25
8557	8557	411873	The Little Hours	Garfagnana, Italy, 1347. The handsome servant ...	30-06-2017	23.265	5.7	41
8558	8558	227783	The Nut Job	Surly, a curmudgeon, independent squirrel is b...	17-01-2014	17.392	5.7	79
8559	8559	446170	Black Tide	When a teenager suddenly disappears without a ...	18-07-2018	6.485	5.7	22

In [5]:

```
movies_data.shape
```

Out[5]:

```
(8560, 8)
```

In [6]:

```
movies_data.columns
```

Out[6]:

```
Index(['Unnamed: 0', 'id', 'title', 'overview', 'release_date', 'popularit
y',
       'vote_average', 'vote_count'],
      dtype='object')
```

In [7]:

```
movies_data = movies_data.drop(['Unnamed: 0', 'id'], axis = 1)
```

In [8]:

```
movies_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8560 entries, 0 to 8559
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   title         8560 non-null   object 
 1   overview      8556 non-null   object 
 2   release_date  8560 non-null   object 
 3   popularity    8560 non-null   float64
 4   vote_average  8560 non-null   float64
 5   vote_count    8560 non-null   int64  
dtypes: float64(2), int64(1), object(3)
memory usage: 401.4+ KB
```

In [9]:

```
movies_data.describe()
```

Out[9]:

	popularity	vote_average	vote_count
count	8560.000000	8560.000000	8560.000000
mean	34.483893	6.803832	1663.763201
std	259.280939	0.632387	2777.837511
min	0.600000	5.700000	198.000000
25%	8.350000	6.300000	327.000000
50%	11.703000	6.700000	625.000000
75%	21.335250	7.300000	1619.250000
max	11288.261000	8.700000	31575.000000

In [10]:

```
movies_data.isnull().sum()
```

Out[10]:

```
title           0
overview        4
release_date    0
popularity      0
vote_average    0
vote_count      0
dtype: int64
```

In [11]:

```
movies_data = movies_data.drop(['overview'], axis = 1)
```

In [12]:

```
movies_data.isnull().sum()
```

Out[12]:

```
title           0
release_date    0
popularity      0
vote_average    0
vote_count      0
dtype: int64
```

In [13]:

```
movies_data.duplicated().sum()
```

Out[13]:

```
0
```

In [14]:

```
new = movies_data["release_date"].str.split("-", n = 2, expand = True)
movies_data["day"]= new[0]
movies_data["month"]= new[1]
movies_data["year"]= new[2]
```

In [15]:

```
movies_data.head()
```

Out[15]:

	title	release_date	popularity	vote_average	vote_count	day	month	year
0	The Shawshank Redemption	23-09-1994	62.636	8.7	21456	23	09	1994
1	Dilwale Dulhania Le Jayenge	20-10-1995	19.097	8.7	3652	20	10	1995
2	The Godfather	14-03-1972	57.656	8.7	15990	14	03	1972
3	Schindler's List	30-11-1993	41.077	8.6	12778	30	11	1993
4	The Godfather: Part II	20-12-1974	46.655	8.6	9640	20	12	1974

In [16]:

```
movies_data.nunique()
```

Out[16]:

```
title           8307
release_date    5601
popularity      7134
vote_average      31
vote_count      3075
day               31
month             12
year             109
dtype: int64
```

In [17]:

```
movies_data['vote_average'].unique()
```

Out[17]:

```
array([8.7, 8.6, 8.5, 8.4, 8.3, 8.2, 8.1, 8. , 7.9, 7.8, 7.7, 7.6, 7.5,
       7.4, 7.3, 7.2, 7.1, 7. , 6.9, 6.8, 6.7, 6.6, 6.5, 6.4, 6.3, 6.2,
       6.1, 6. , 5.9, 5.8, 5.7])
```

In [18]:

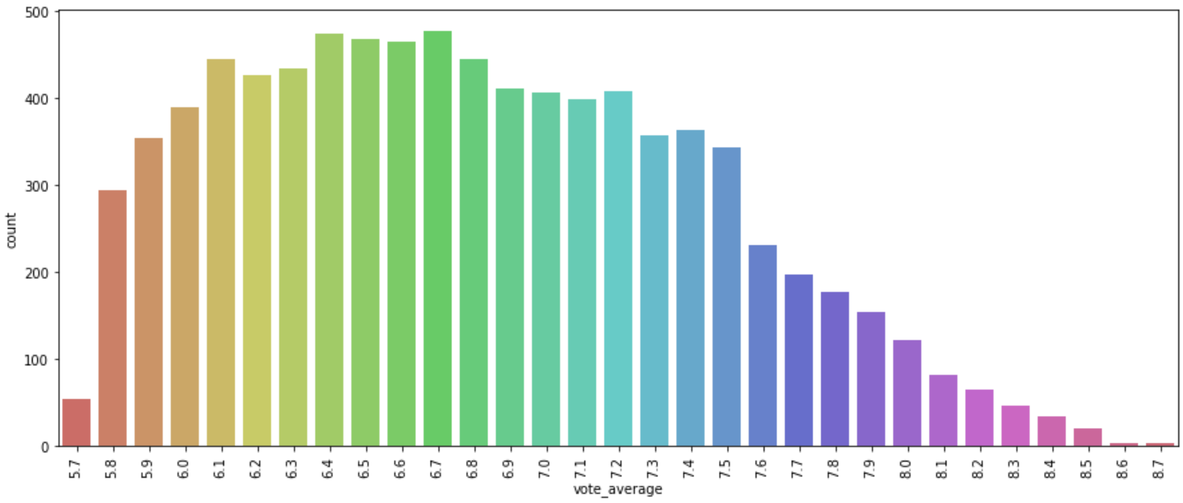
```
movies_data['vote_average'].value_counts()
```

Out[18]:

```
6.7    478
6.4    475
6.5    469
6.6    465
6.8    445
6.1    445
6.3    434
6.2    427
6.9    411
7.2    408
7.0    407
7.1    399
6.0    390
7.4    364
7.3    357
5.9    354
7.5    344
5.8    294
7.6    231
7.7    198
7.8    177
7.9    154
8.0    122
8.1     82
8.2     66
5.7     54
8.3     47
8.4     35
8.5     21
8.6      4
8.7      3
Name: vote_average, dtype: int64
```

In [19]:

```
plt.figure(figsize=(15,6))
sns.countplot('vote_average', data = movies_data,
                  palette='hls')
plt.xticks(rotation = 90)
plt.show()
```

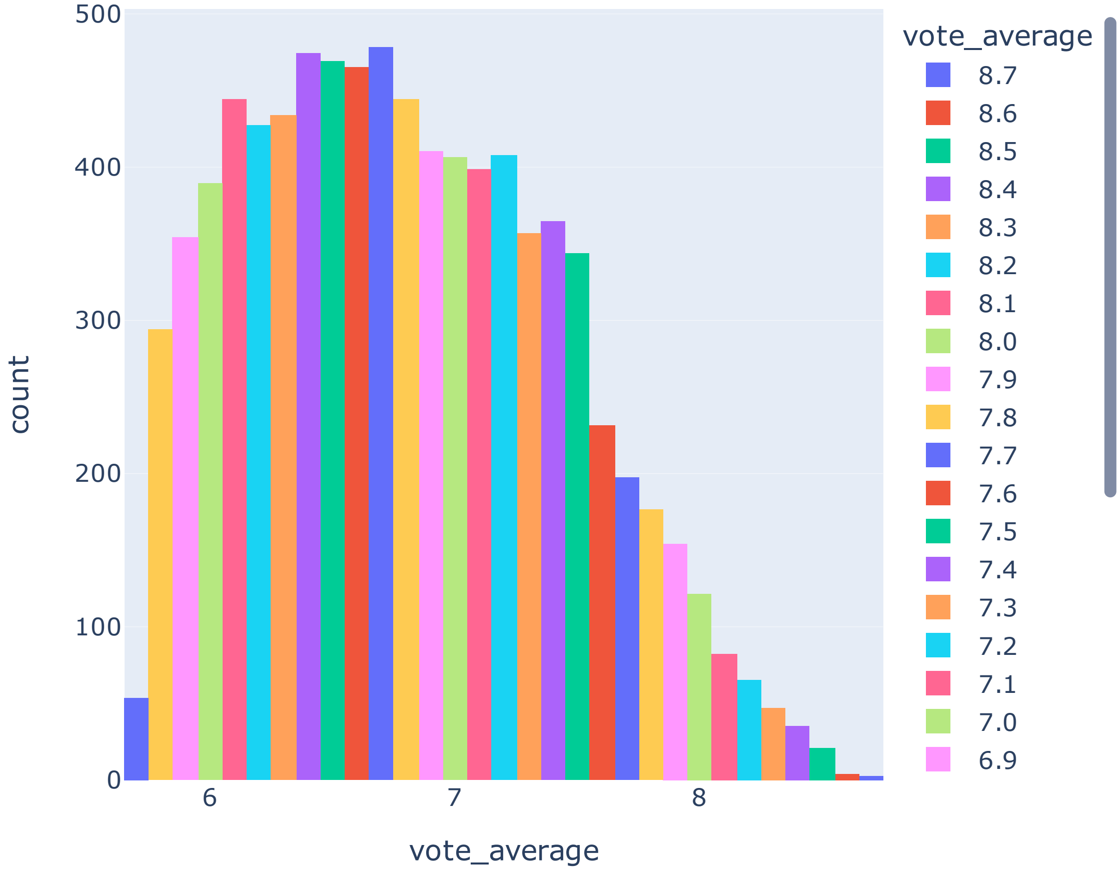


In [20]:

```
import plotly.express as px
```

In [21]:

```
fig1 = px.histogram(movies_data, x = 'vote_average', color = 'vote_average')
fig1.show()
```



In [22]:

```
movies_popularity = movies_data.copy()
```

In [23]:

```
movies_popularity = movies_popularity.sort_values(by = 'popularity',
                                                   ascending = False)
```

In [24]:

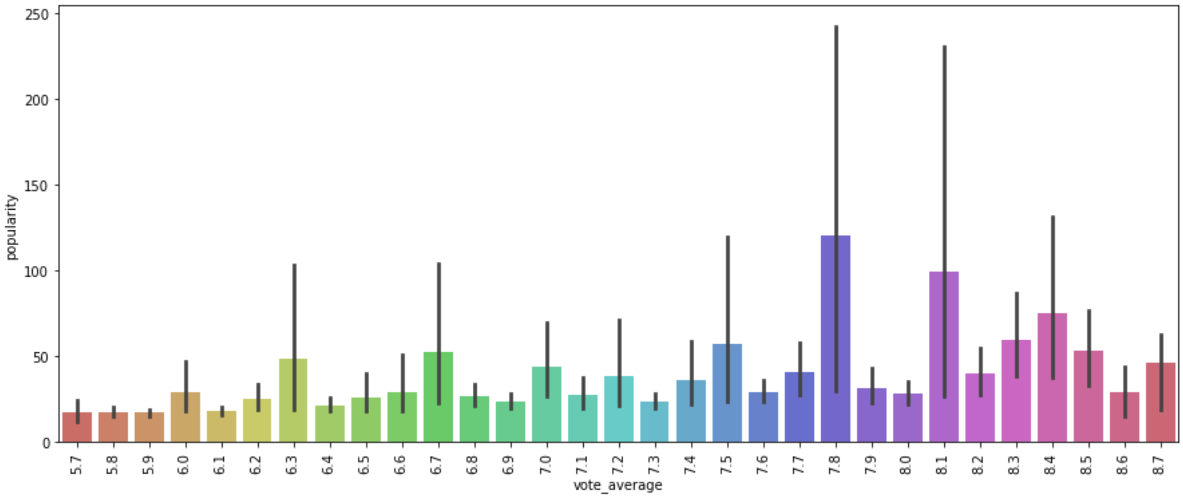
```
movies_popularity.head()
```

Out[24]:

	title	release_date	popularity	vote_average	vote_count	day	month	year
4238	The Lost City	24-03-2022	11288.261	6.7	768	24	03	2022
6238	Morbius	30-03-2022	11236.546	6.3	1125	30	03	2022
1400	The Northman	07-04-2022	7895.411	7.5	1095	07	04	2022
703	Sonic the Hedgehog 2	30-03-2022	7088.307	7.8	1622	30	03	2022
638	The Batman	01-03-2022	6372.913	7.8	4767	01	03	2022

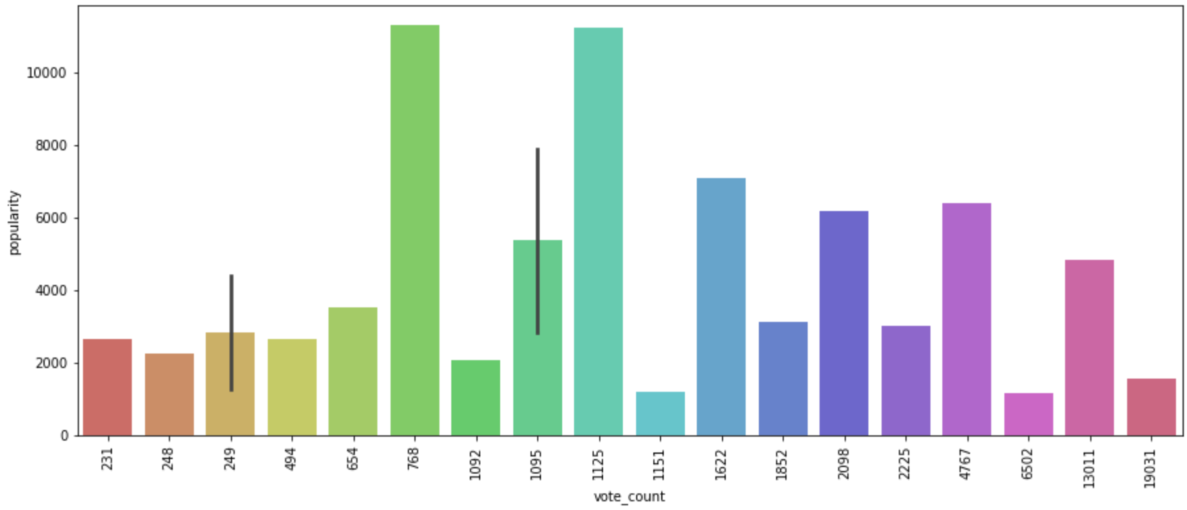
In [26]:

```
plt.figure(figsize=(15,6))
sns.barplot(x = 'vote_average', y = 'popularity',data = movies_popularity,
                  palette='hls')
plt.xticks(rotation = 90)
plt.show()
```



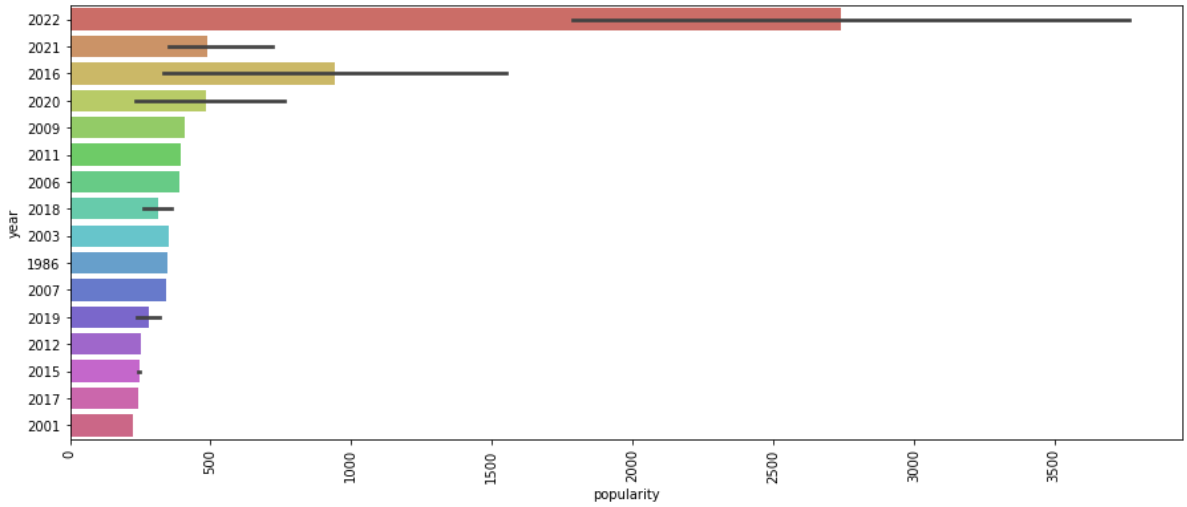
In [27]:

```
plt.figure(figsize=(15,6))
sns.barplot(x = 'vote_count', y = 'popularity',data = movies_popularity.head(20),
                  palette='hls')
plt.xticks(rotation = 90)
plt.show()
```



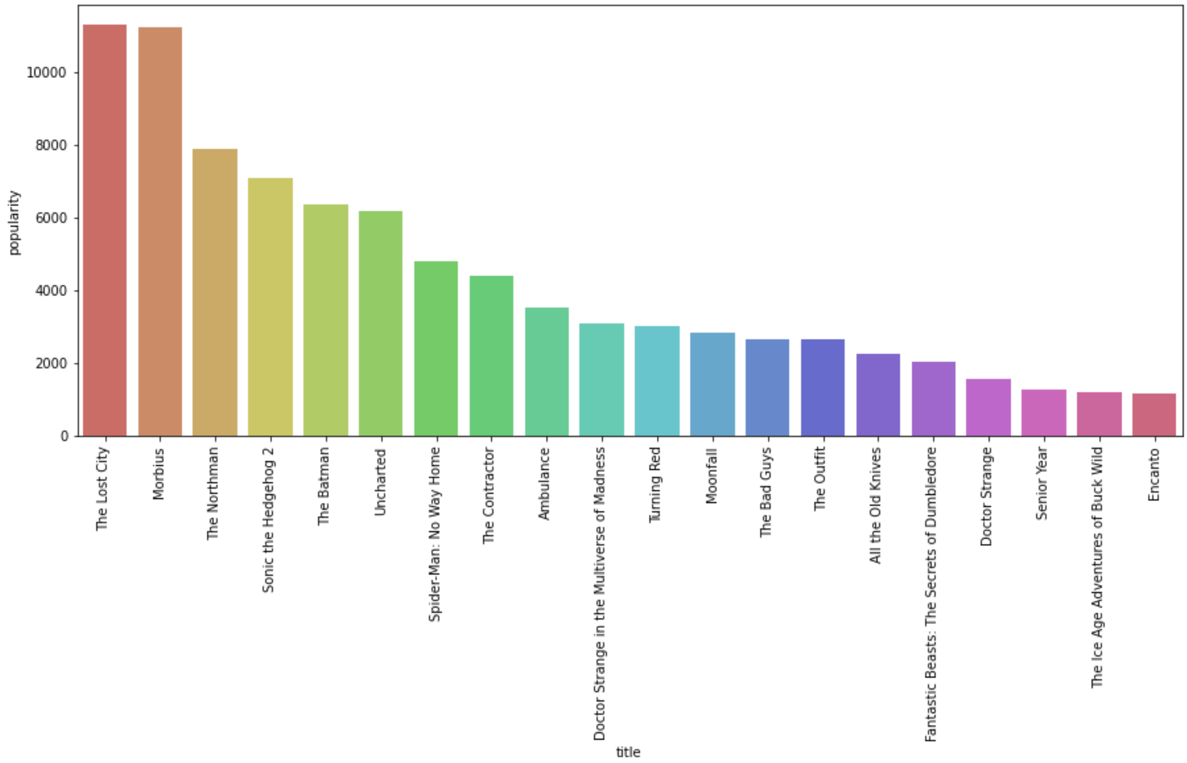
In [40]:

```
plt.figure(figsize=(15,6))
sns.barplot(y = 'year', x = 'popularity',data = movies_popularity.head(100),
                  palette='hls')
plt.xticks(rotation = 90)
plt.show()
```



In [28]:

```
plt.figure(figsize=(15,6))
sns.barplot(x = 'title', y = 'popularity',data = movies_popularity.head(20),
                  palette='hls')
plt.xticks(rotation = 90)
plt.show()
```



In [29]:

```
movies_vote_average = movies_data.copy()
```

In [30]:

```
movies_vote_average = movies_vote_average.sort_values(by = 'vote_average',
                                                      ascending = False)
```

In [31]:

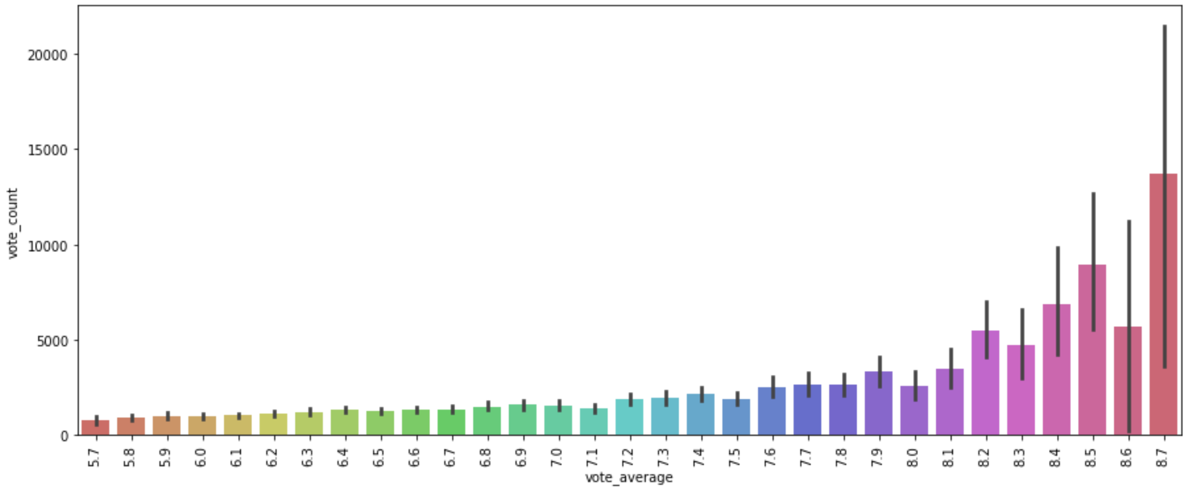
```
movies_vote_average.head()
```

Out[31]:

	title	release_date	popularity	vote_average	vote_count	day	month	year
0	The Shawshank Redemption	23-09-1994	62.636	8.7	21456	23	09	1994
2	The Godfather	14-03-1972	57.656	8.7	15990	14	03	1972
1	Dilwale Dulhania Le Jayenge	20-10-1995	19.097	8.7	3652	20	10	1995
3	Schindler's List	30-11-1993	41.077	8.6	12778	30	11	1993
4	The Godfather: Part II	20-12-1974	46.655	8.6	9640	20	12	1974

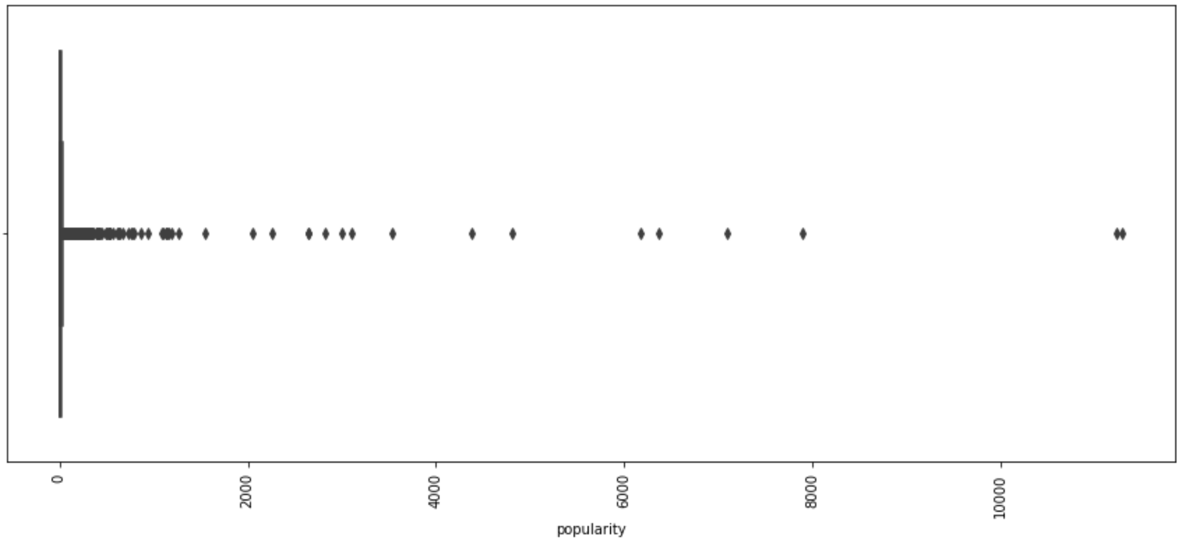
In [32]:

```
plt.figure(figsize=(15,6))
sns.barplot(x = 'vote_average', y = 'vote_count',data = movies_vote_average,
                  palette='hls')
plt.xticks(rotation = 90)
plt.show()
```



In [35]:

```
plt.figure(figsize=(15,6))
sns.boxplot(movies_data['popularity'])
plt.xticks(rotation = 90)
plt.show()
```



In [36]:

```
movies_popularity= movies_data['popularity']
Q3 = movies_popularity.quantile(0.75)
Q1 = movies_popularity.quantile(0.25)
IQR = Q3-Q1
lower_limit = Q1 -(1.5*IQR)
upper_limit = Q3 +(1.5*IQR)
popularity_outliers = movies_popularity[(movies_popularity <lower_limit) | (movies_popul
popularity_outliers
```

Out[36]:

```
0        62.636
2        57.656
3        41.077
4        46.655
7        69.900
          ...
8506     77.874
8519    155.600
8525     56.071
8526     57.453
8536     48.739
Name: popularity, Length: 1028, dtype: float64
```

In [42]:

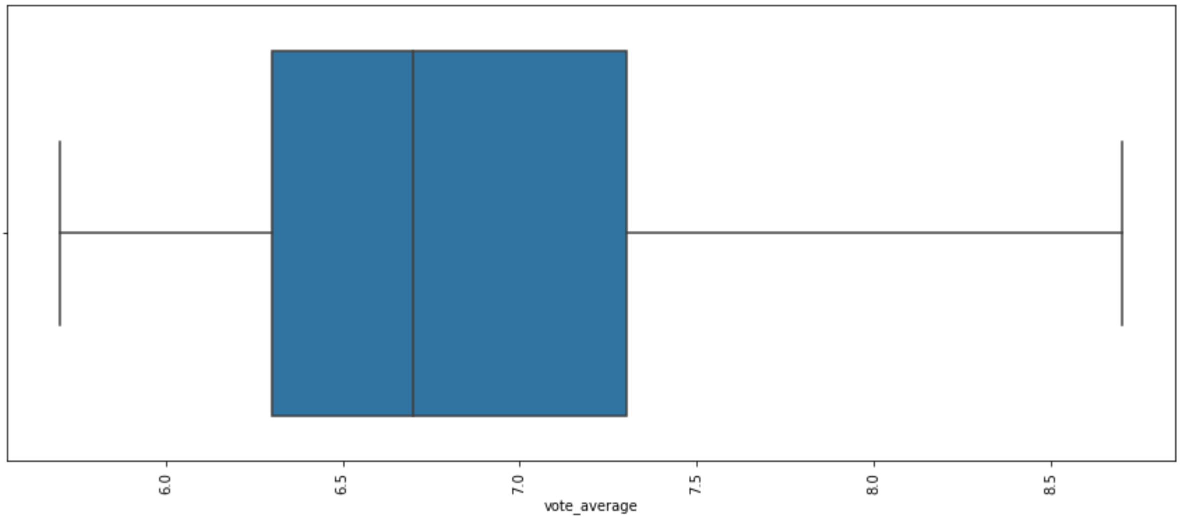
```
popularity_filtered = movies_popularity[(movies_popularity >lower_limit) & (movies_popu
popularity_filtered
```

Out[42]:

```
0       62.636
1       19.097
2       57.656
3       41.077
4       46.655
         ...  
8555     9.382
8556     5.406
8557    23.265
8558    17.392
8559     6.485
Name: popularity, Length: 8552, dtype: float64
```

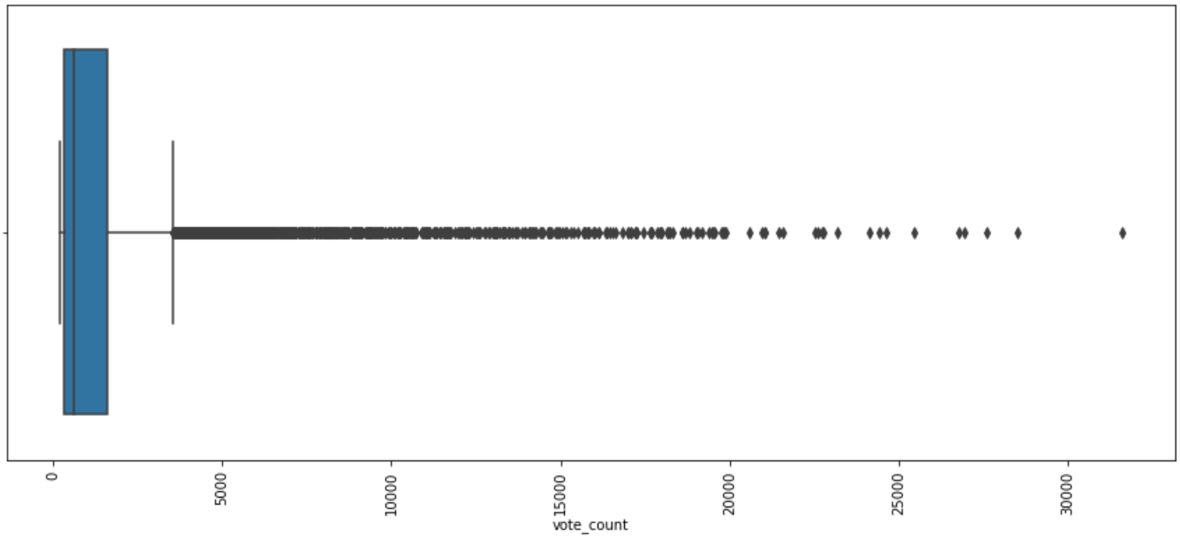
In [37]:

```
plt.figure(figsize=(15,6))
sns.boxplot(movies_data['vote_average'])
plt.xticks(rotation = 90)
plt.show()
```



In [39]:

```
plt.figure(figsize=(15,6))
sns.boxplot(movies_data['vote_count'])
plt.xticks(rotation = 90)
plt.show()
```



In [40]:

```
movies_vote_count = movies_data['vote_count']
Q3 = movies_vote_count.quantile(0.75)
Q1 = movies_vote_count.quantile(0.25)
IQR = Q3-Q1
lower_limit = Q1 -(1.5*IQR)
upper_limit = Q3 +(1.5*IQR)
vote_count_outliers = movies_vote_count[(movies_vote_count <lower_limit) | (movies_vote_
vote_count_outliers
```

Out[40]:

```
0       21456
1        3652
2       15990
3       12778
4        9640
        ...  
8460     4656
8463     3722
8477     3929
8479     3846
8550     3619
Name: vote_count, Length: 1022, dtype: int64
```

In [44]:

```
vote_count_filters = movies_vote_count[(movies_vote_count > lower_limit) & (movies_vote_
vote_count_filters
```

Out[44]:

```
5        237
6        230
9       2245
10      1411
11       353
        ... 
8555     827
8556     250
8557     419
8558     794
8559     225
Name: vote_count, Length: 7538, dtype: int64
```

In [45]:

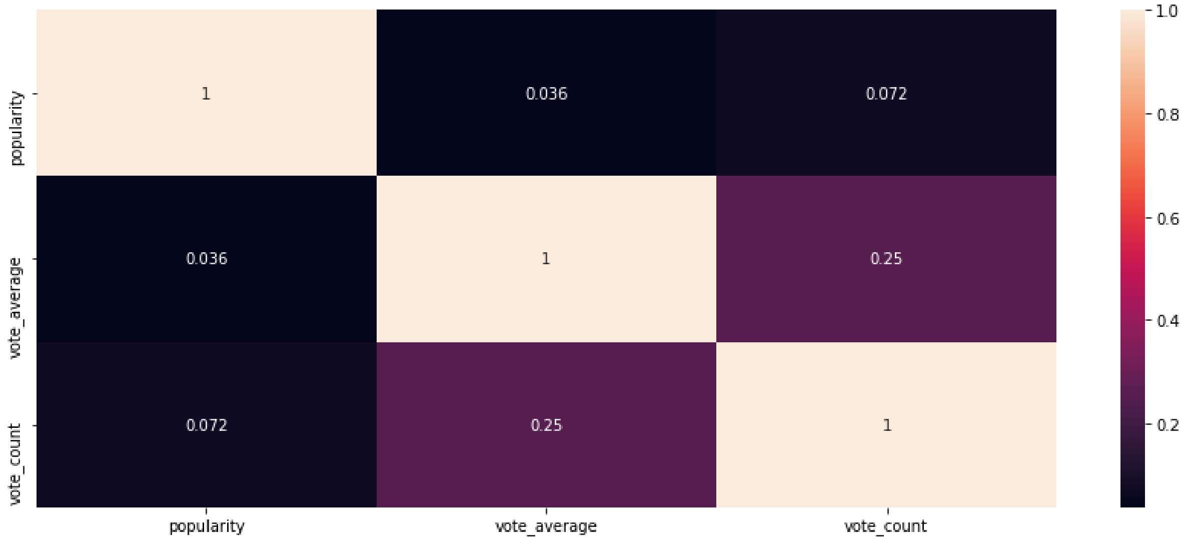
```
movies_data.corr()
```

Out[45]:

	popularity	vote_average	vote_count
popularity	1.000000	0.036491	0.071668
vote_average	0.036491	1.000000	0.253971
vote_count	0.071668	0.253971	1.000000

In [46]:

```
plt.figure(figsize=(15,6))
sns.heatmap(movies_data.corr(), annot = True)
plt.show()
```


popularity
vote_average
vote_count
popularity
vote_average
vote_count