

```
In [1]: import pandas as pd
```

```
In [2]: movies=pd.read_csv(r"C:\Users\Sonu\Downloads\Movie-Rating.csv")
```

```
In [3]: movies
```

Out[3]:

	Film	Genre	Rotten Tomatoes Ratings %	Audience Ratings %	Budget (million \$)	Year of release
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009
...
554	Your Highness	Comedy	26	36	50	2011
555	Youth in Revolt	Comedy	68	52	18	2009
556	Zodiac	Thriller	89	73	65	2007
557	Zombieland	Action	90	87	24	2009
558	Zookeeper	Comedy	14	42	80	2011

559 rows × 6 columns

```
In [4]: movies.columns
```

```
Out[4]: Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
       'Budget (million $)', 'Year of release'],
       dtype='object')
```

```
In [5]: movies.shape
```

```
Out[5]: (559, 6)
```

```
In [6]: movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Film              559 non-null    object  
 1   Genre             559 non-null    object  
 2   Rotten Tomatoes Ratings %  559 non-null    int64  
 3   Audience Ratings %  559 non-null    int64  
 4   Budget (million $) 559 non-null    int64  
 5   Year of release   559 non-null    int64  
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

In [7]: `movies.head()`

	Film	Genre	Rotten Tomatoes Ratings %	Audience Ratings %	Budget (million \$)	Year of release
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009

In [8]: `movies.tail()`

	Film	Genre	Rotten Tomatoes Ratings %	Audience Ratings %	Budget (million \$)	Year of release
554	Your Highness	Comedy	26	36	50	2011
555	Youth in Revolt	Comedy	68	52	18	2009
556	Zodiac	Thriller	89	73	65	2007
557	Zombieland	Action	90	87	24	2009
558	Zookeeper	Comedy	14	42	80	2011

In [9]: `movies.columns=['Film', 'Genre', 'CriticRating', 'AudienceRating', 'BudgetMillions', 'Ye`

In [10]: `movies.columns`

```
Out[10]: Index(['Film', 'Genre', 'CriticRating', 'AudienceRating', 'BudgetMillions',
       'Year'],
      dtype='object')
```

In [11]: `movies.head(1)`

Out[11]:

	Film	Genre	CriticRating	AudienceRating	BudgetMillions	Year
0	(500) Days of Summer	Comedy		87	81	8 2009

In [12]: `movies.describe()`

Out[12]:

	CriticRating	AudienceRating	BudgetMillions	Year
count	559.000000	559.000000	559.000000	559.000000
mean	47.309481	58.744186	50.236136	2009.152057
std	26.413091	16.826887	48.731817	1.362632
min	0.000000	0.000000	0.000000	2007.000000
25%	25.000000	47.000000	20.000000	2008.000000
50%	46.000000	58.000000	35.000000	2009.000000
75%	70.000000	72.000000	65.000000	2010.000000
max	97.000000	96.000000	300.000000	2011.000000

In [13]: `movies.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Film              559 non-null    object  
 1   Genre             559 non-null    object  
 2   CriticRating      559 non-null    int64  
 3   AudienceRating    559 non-null    int64  
 4   BudgetMillions   559 non-null    int64  
 5   Year              559 non-null    int64  
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

In [14]: `movies.Film=movies.Film.astype('category')`

In [15]: `movies.Film`

```
Out[15]: 0      (500) Days of Summer
          1          10,000 B.C.
          2          12 Rounds
          3         127 Hours
          4          17 Again
          ...
          554      Your Highness
          555      Youth in Revolt
          556          Zodiac
          557      Zombieland
          558      Zookeeper
Name: Film, Length: 559, dtype: category
Categories (559, object): ['(500) Days of Summer', '10,000 B.C.', '12 Rounds',
 '127 Hours', ..., 'Youth in Revolt', 'Zodiac', 'Zombieland', 'Zookeeper']
```

```
In [16]: movies.Genre=movies.Genre.astype('category')
movies.Year=movies.Year.astype('category')
```

```
In [17]: movies.Genre
```

```
Out[17]: 0      Comedy
          1      Adventure
          2      Action
          3      Adventure
          4      Comedy
          ...
          554      Comedy
          555      Comedy
          556      Thriller
          557      Action
          558      Comedy
Name: Genre, Length: 559, dtype: category
Categories (7, object): ['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance', 'Thriller']
```

```
In [18]: movies.Year
```

```
Out[18]: 0      2009
          1      2008
          2      2009
          3      2010
          4      2009
          ...
          554      2011
          555      2009
          556      2007
          557      2009
          558      2011
Name: Year, Length: 559, dtype: category
Categories (5, int64): [2007, 2008, 2009, 2010, 2011]
```

```
In [19]: movies.describe()
```

Out[19]:

	CriticRating	AudienceRating	BudgetMillions
count	559.000000	559.000000	559.000000
mean	47.309481	58.744186	50.236136
std	26.413091	16.826887	48.731817
min	0.000000	0.000000	0.000000
25%	25.000000	47.000000	20.000000
50%	46.000000	58.000000	35.000000
75%	70.000000	72.000000	65.000000
max	97.000000	96.000000	300.000000

In [20]:

`movies.info()`

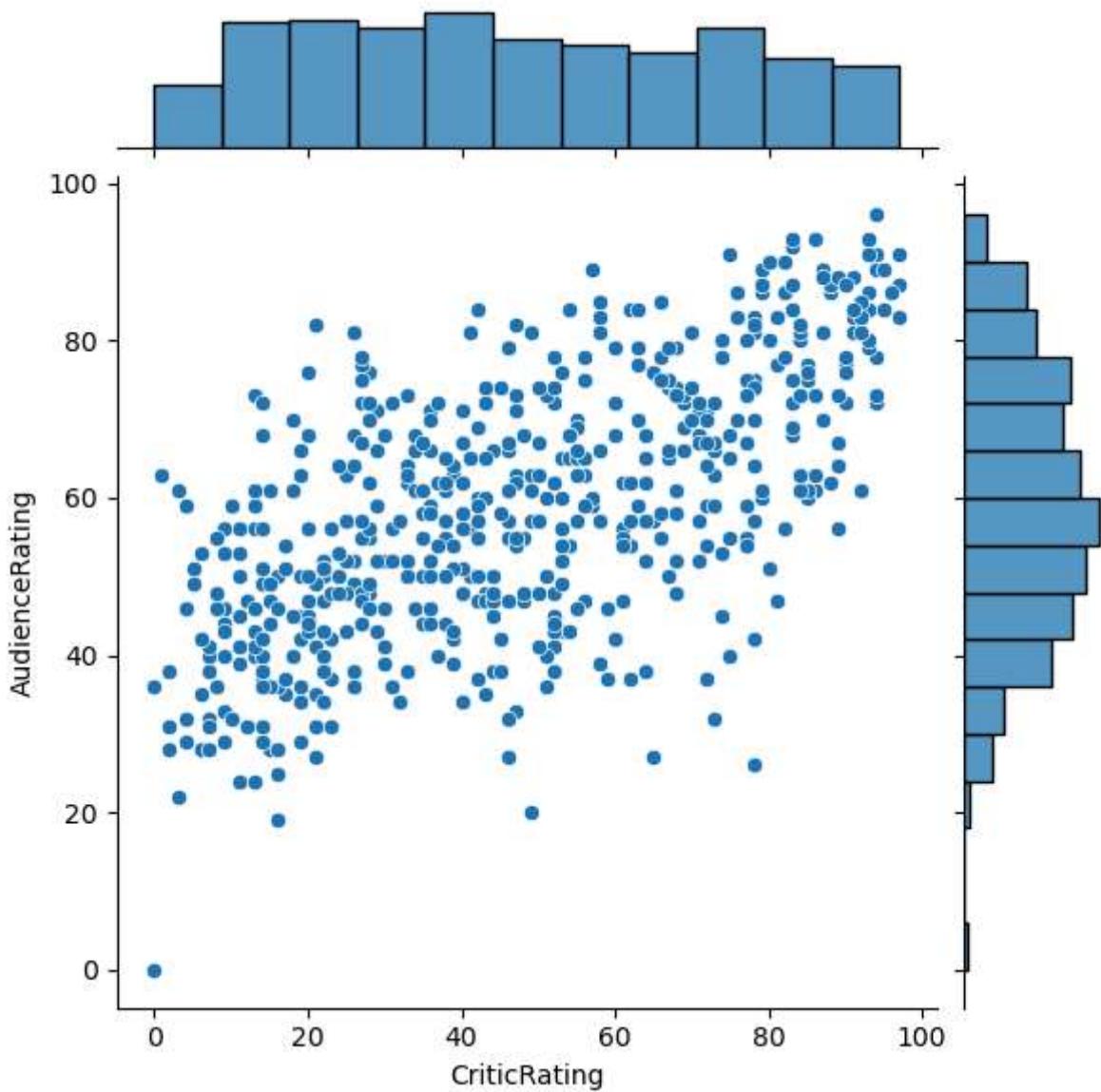
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Film              559 non-null    category
 1   Genre             559 non-null    category
 2   CriticRating      559 non-null    int64  
 3   AudienceRating    559 non-null    int64  
 4   BudgetMillions   559 non-null    int64  
 5   Year              559 non-null    category
dtypes: category(3), int64(3)
memory usage: 36.5 KB
```

In [21]:

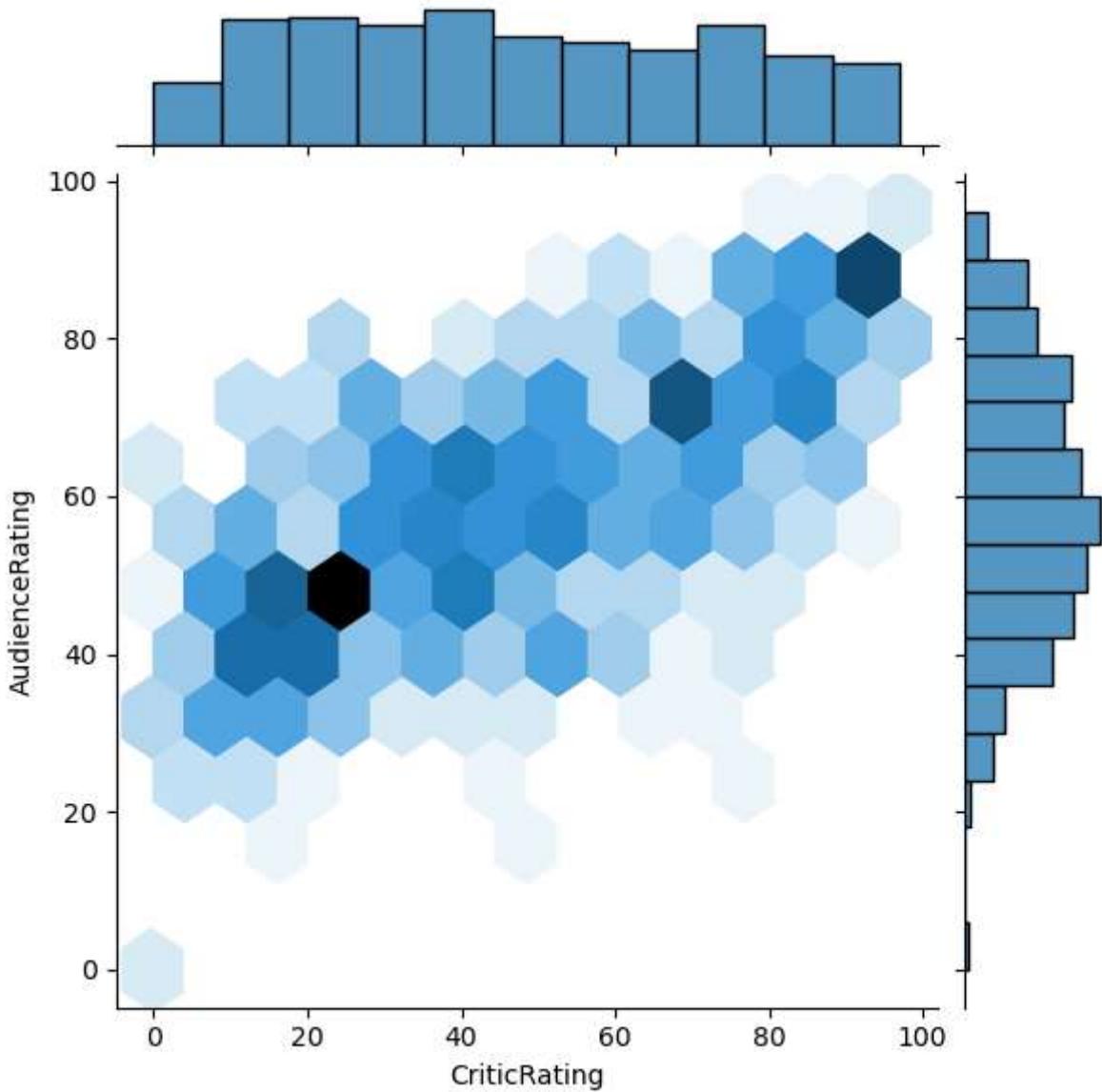
```
from matplotlib import pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

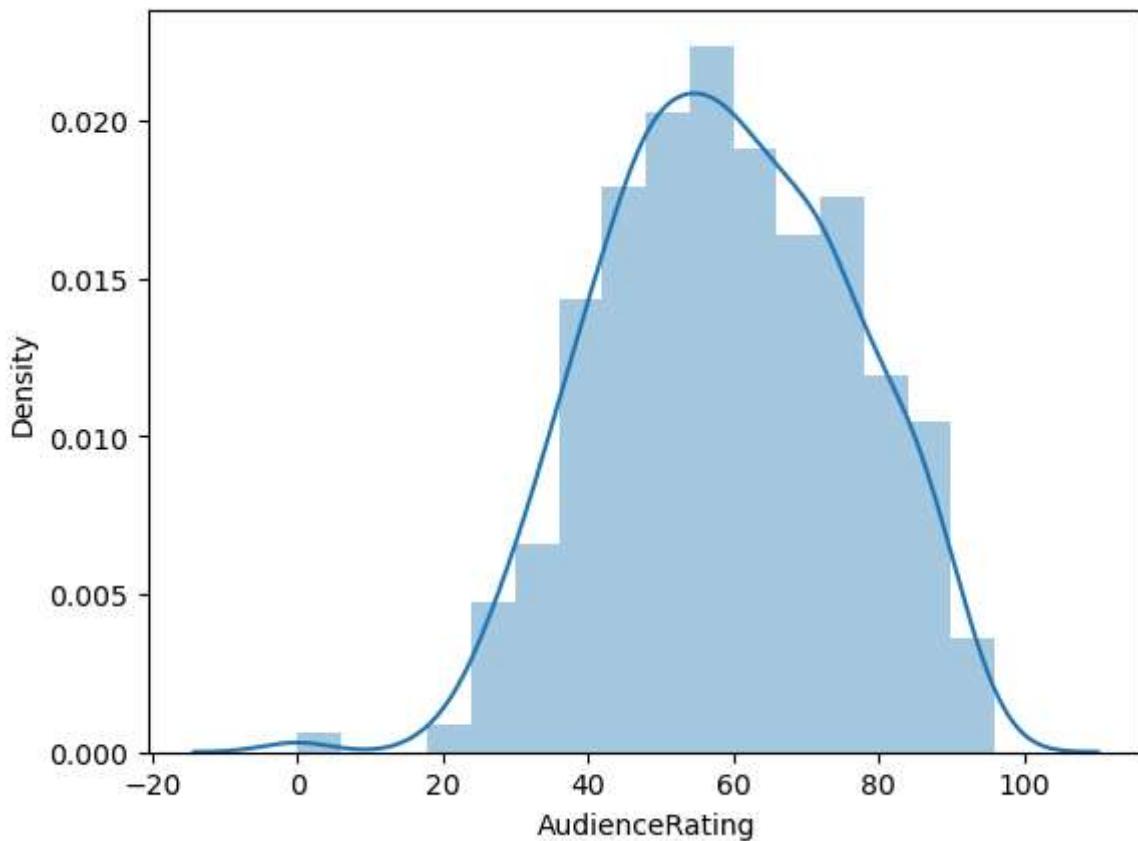
In [22]:

```
j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='scatter')
```



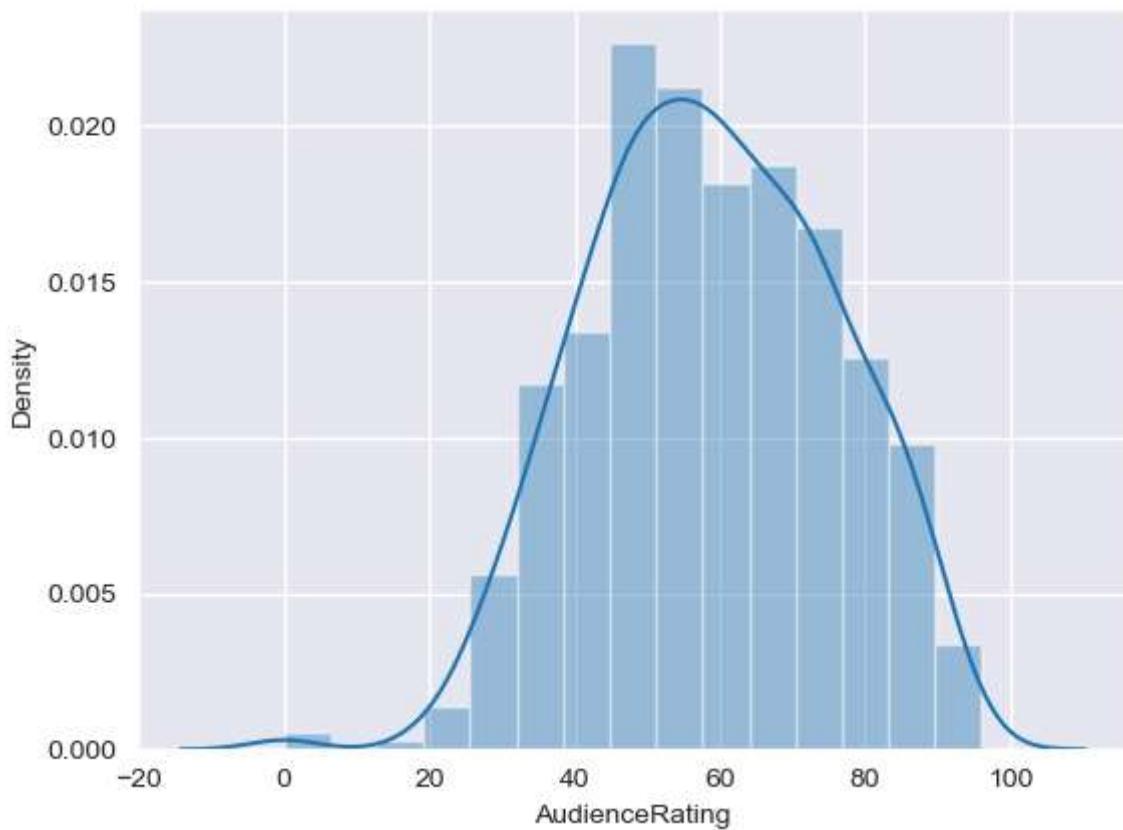
```
In [23]: j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='hex')
```



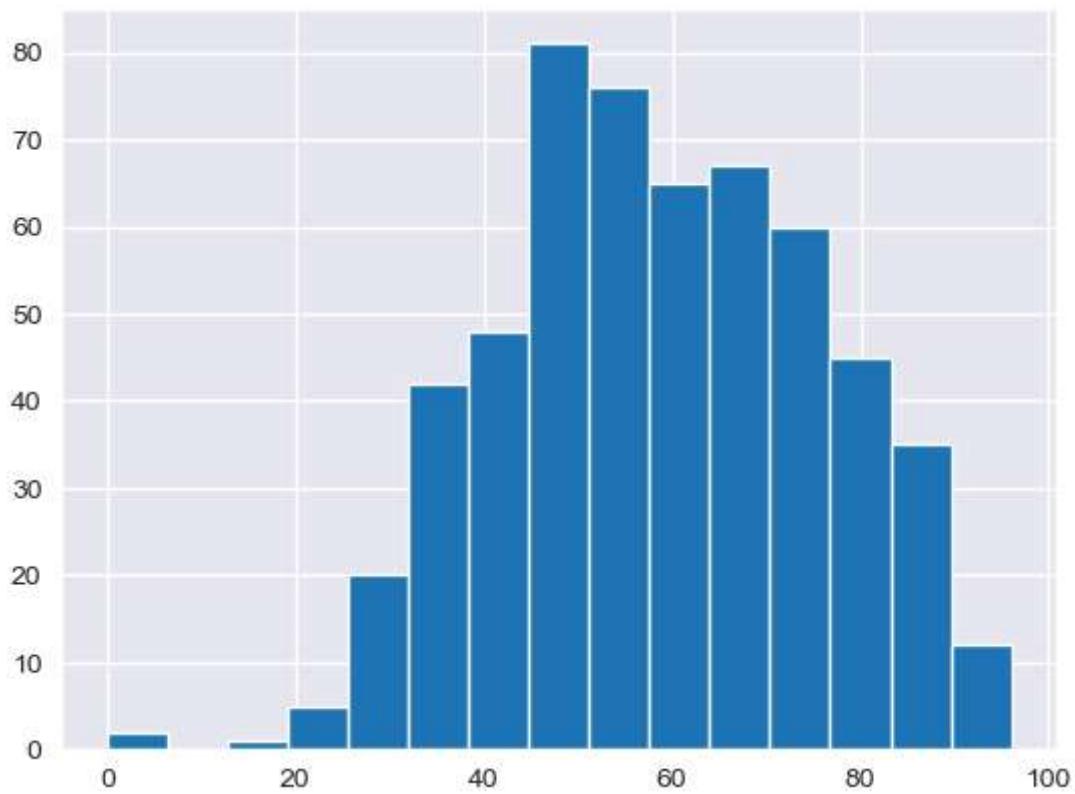


```
In [25]: sns.set_style('darkgrid')
```

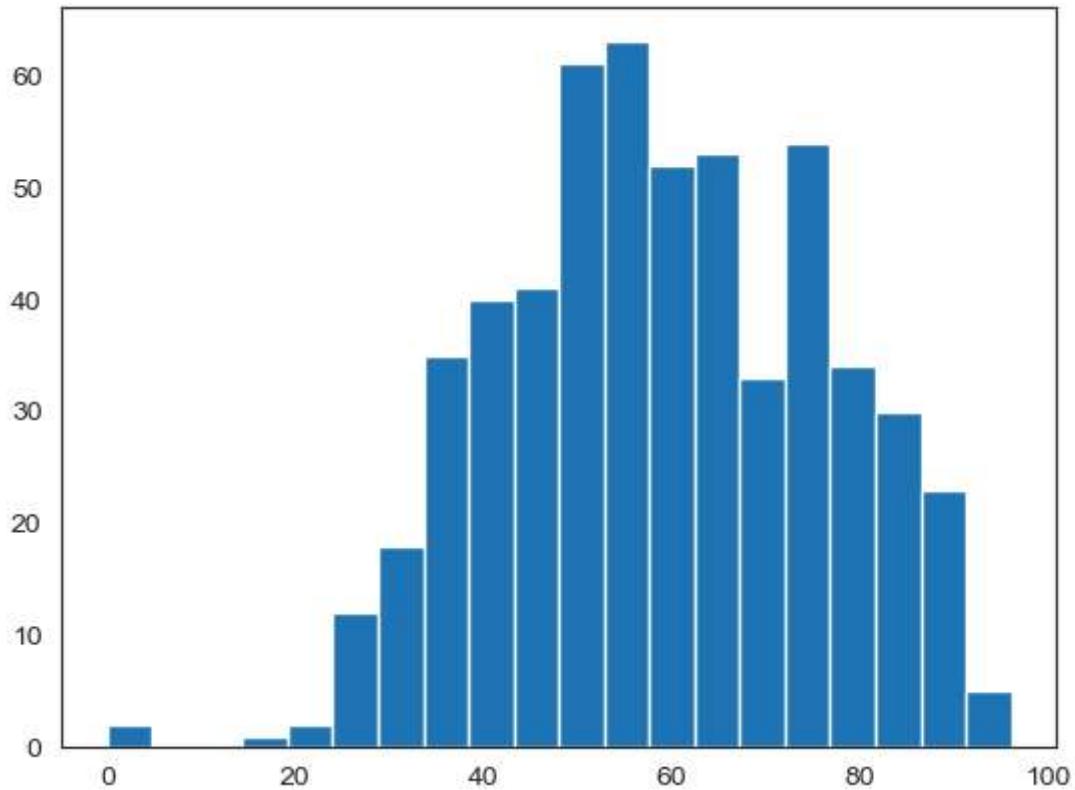
```
In [26]: m2=sns.distplot(movies.AudienceRating,bins=15)
```



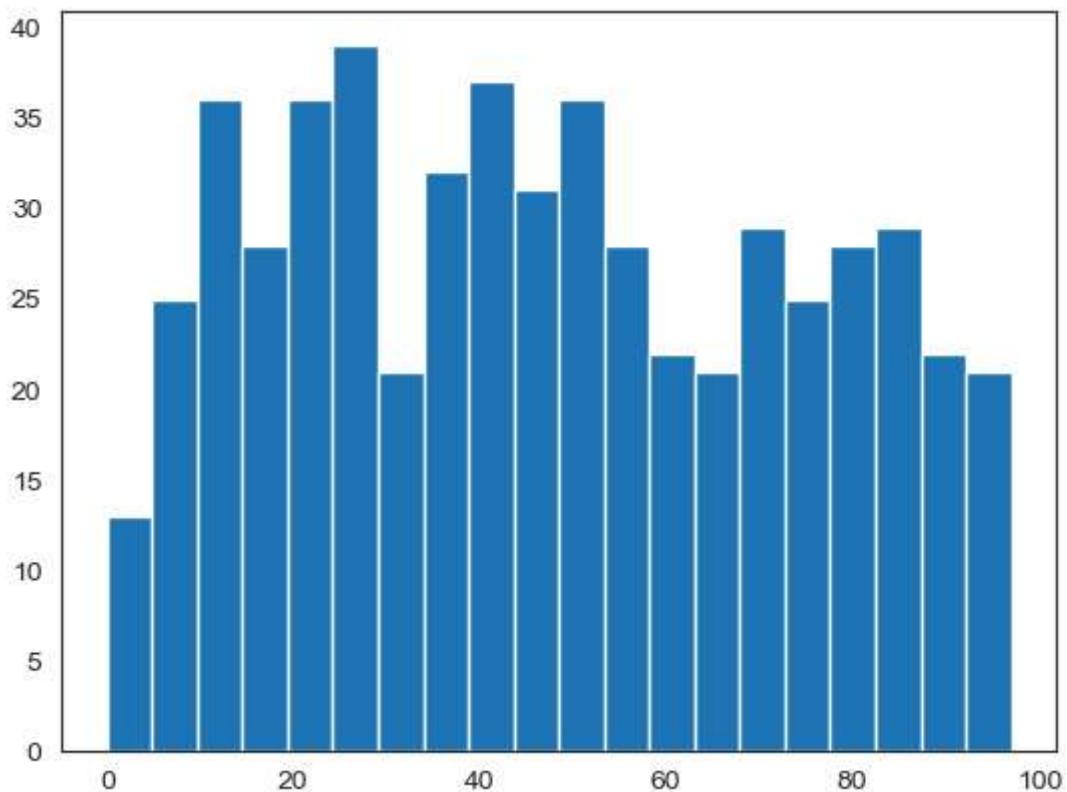
```
In [27]: n1=plt.hist(movies.AudienceRating,bins=15)
```



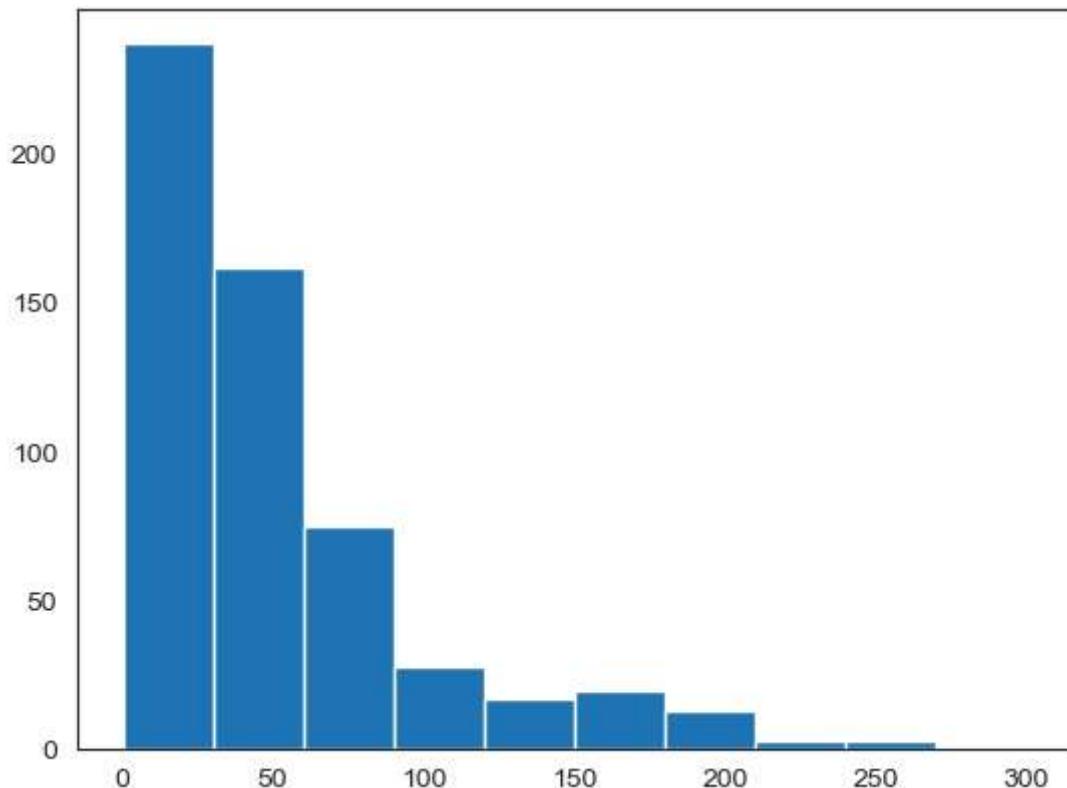
```
In [28]: sns.set_style('white')  
n1=plt.hist(movies.AudienceRating,bins=20)
```



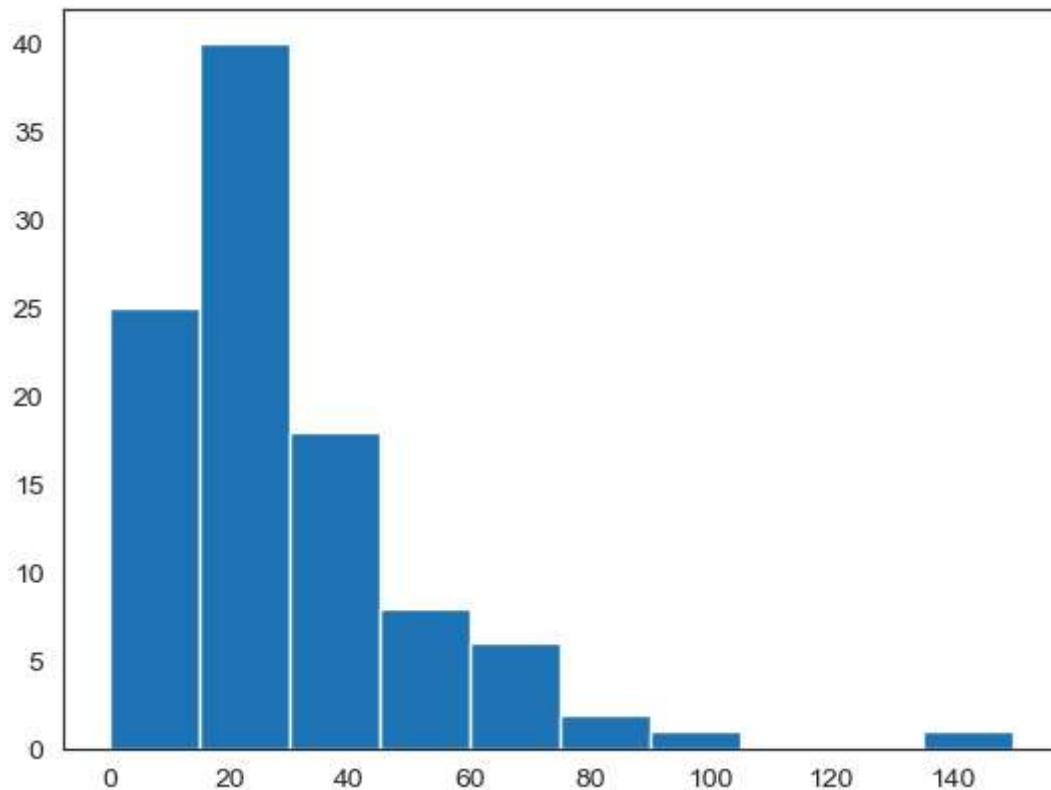
```
In [29]: n1=plt.hist(movies.CriticRating,bins=20)
```



```
In [30]: plt.hist(movies.BudgetMillions)
plt.show()
```



```
In [31]: plt.hist(movies[movies.Genre=='Drama'].BudgetMillions)
plt.show()
```

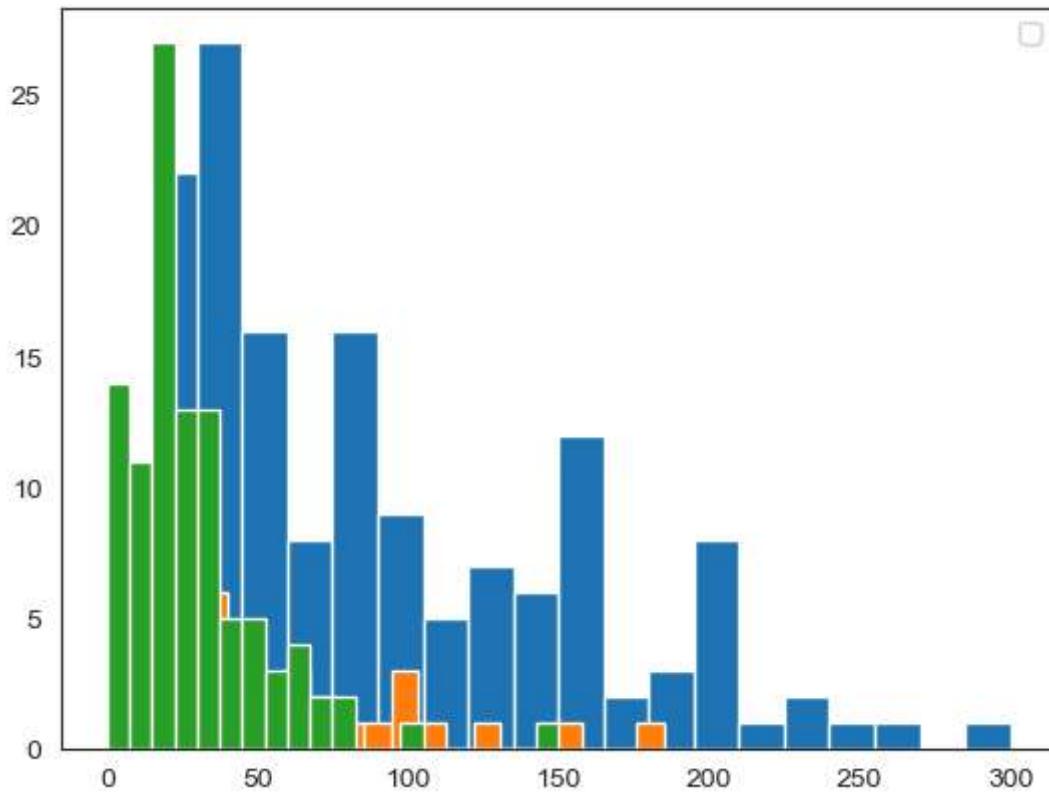


```
In [32]: movies.head()
```

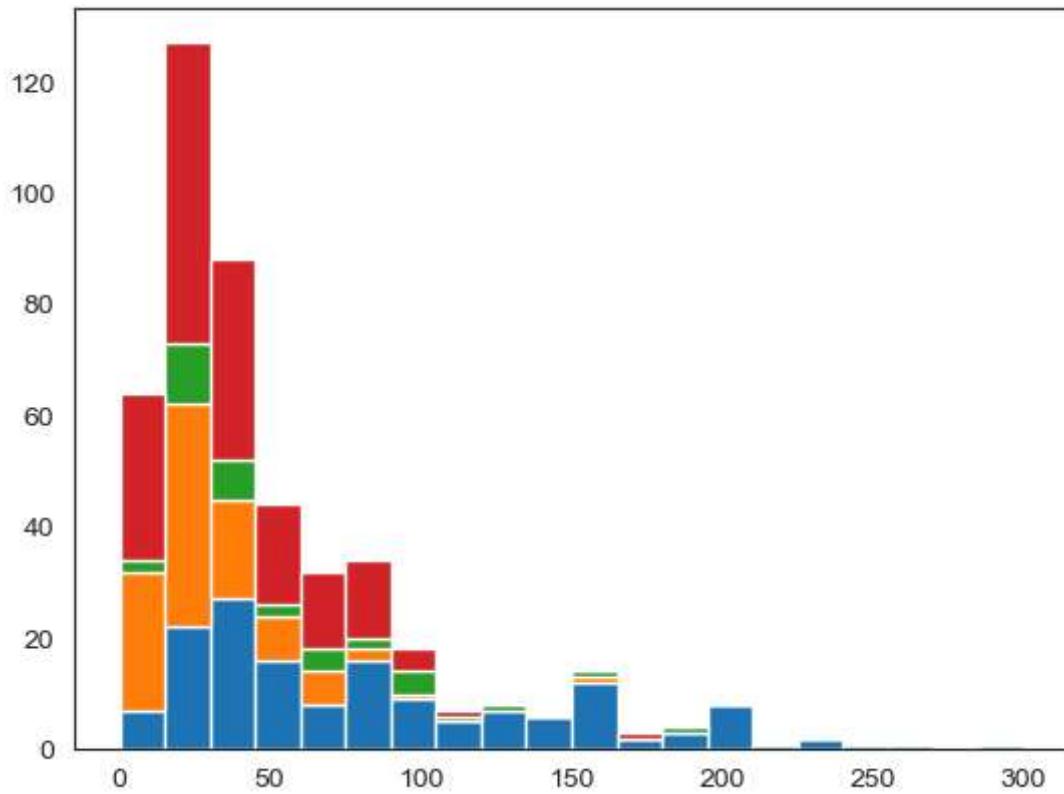
```
Out[32]:
```

	Film	Genre	CriticRating	AudienceRating	BudgetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009

```
In [33]: plt.hist(movies[movies.Genre=='Action'].BudgetMillions,bins=20)
plt.hist(movies[movies.Genre=='Thriller'].BudgetMillions,bins=20)
plt.hist(movies[movies.Genre=='Drama'].BudgetMillions,bins=20)
plt.legend()
plt.show()
```



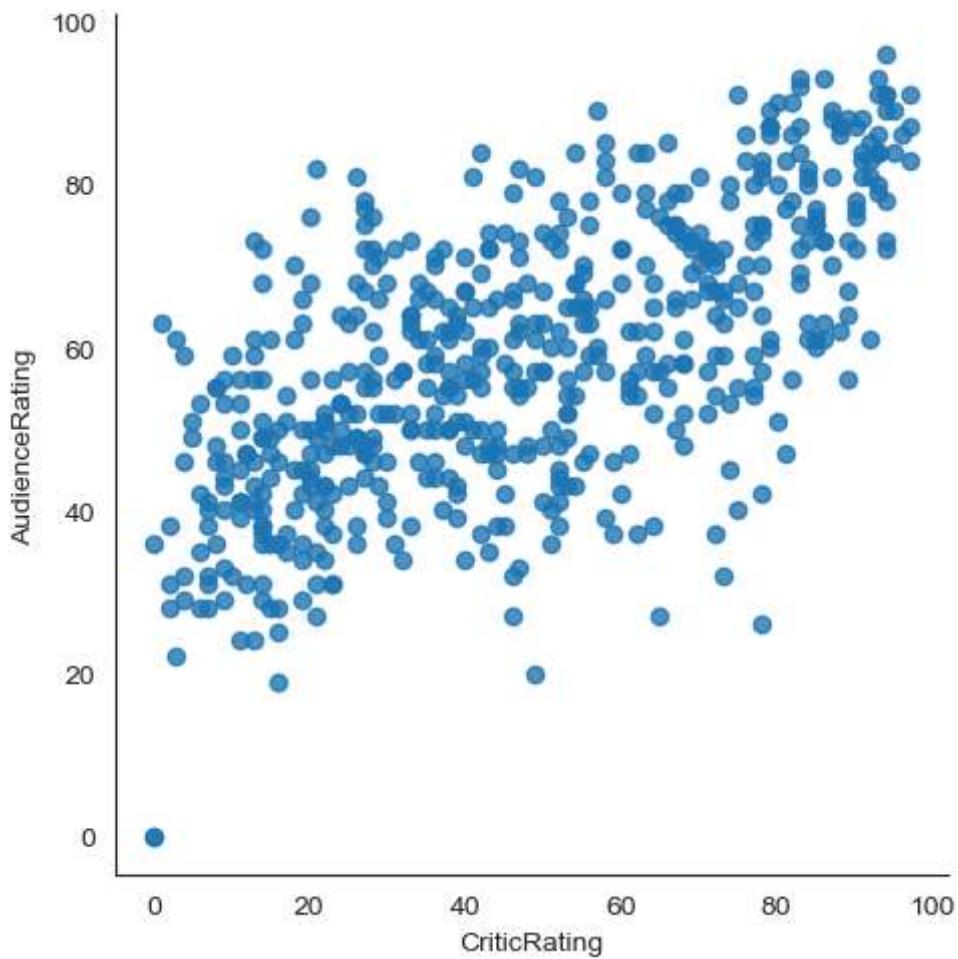
```
In [34]: plt.hist([movies[movies.Genre=='Action'].BudgetMillions,\n               movies[movies.Genre=='Drama'].BudgetMillions,\n               movies[movies.Genre=='Thriller'].BudgetMillions,\n               movies[movies.Genre=='Comedy'].BudgetMillions],\n               bins=20,stacked=True)\nplt.show()
```



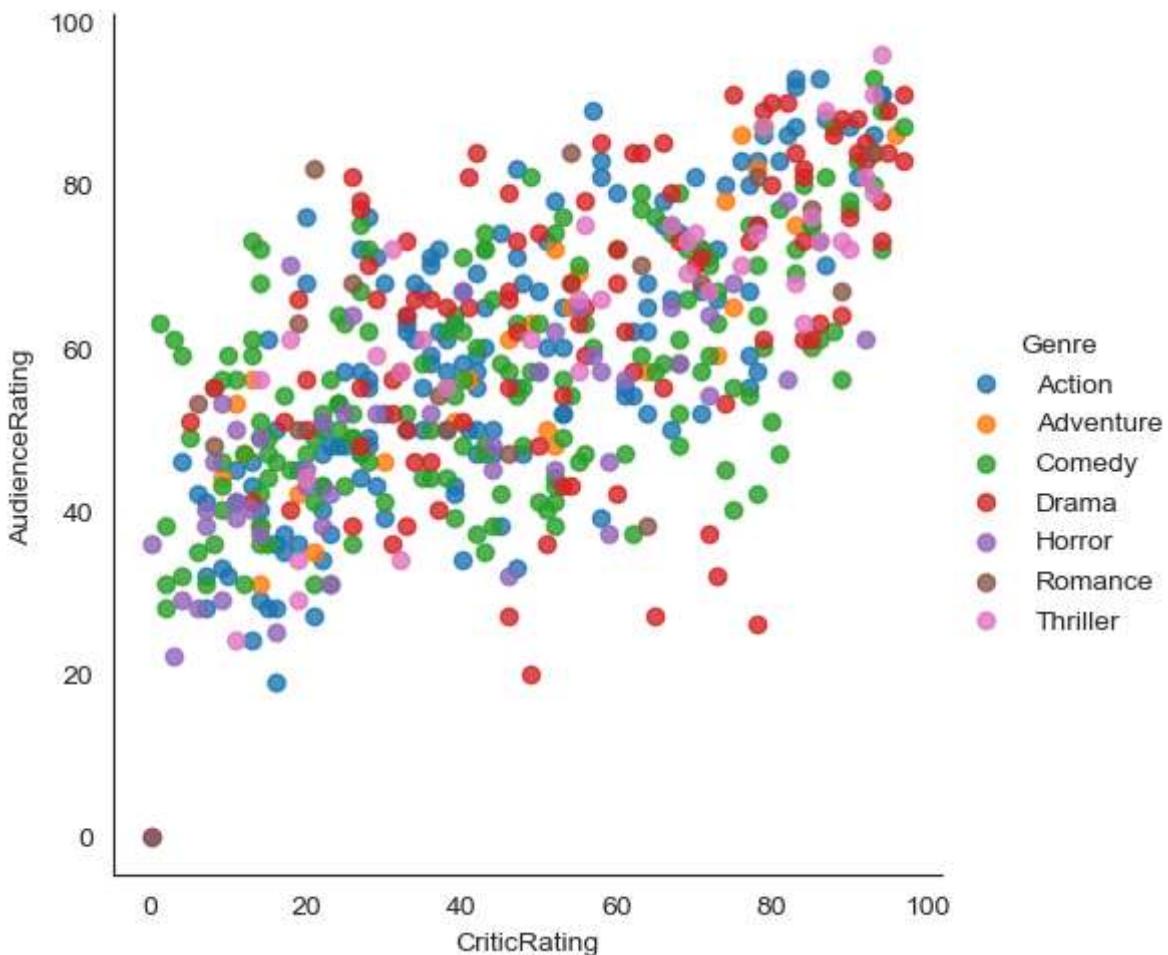
```
In [35]: for gen in movies.Genre.cat.categories:  
    print(gen)
```

Action
Adventure
Comedy
Drama
Horror
Romance
Thriller

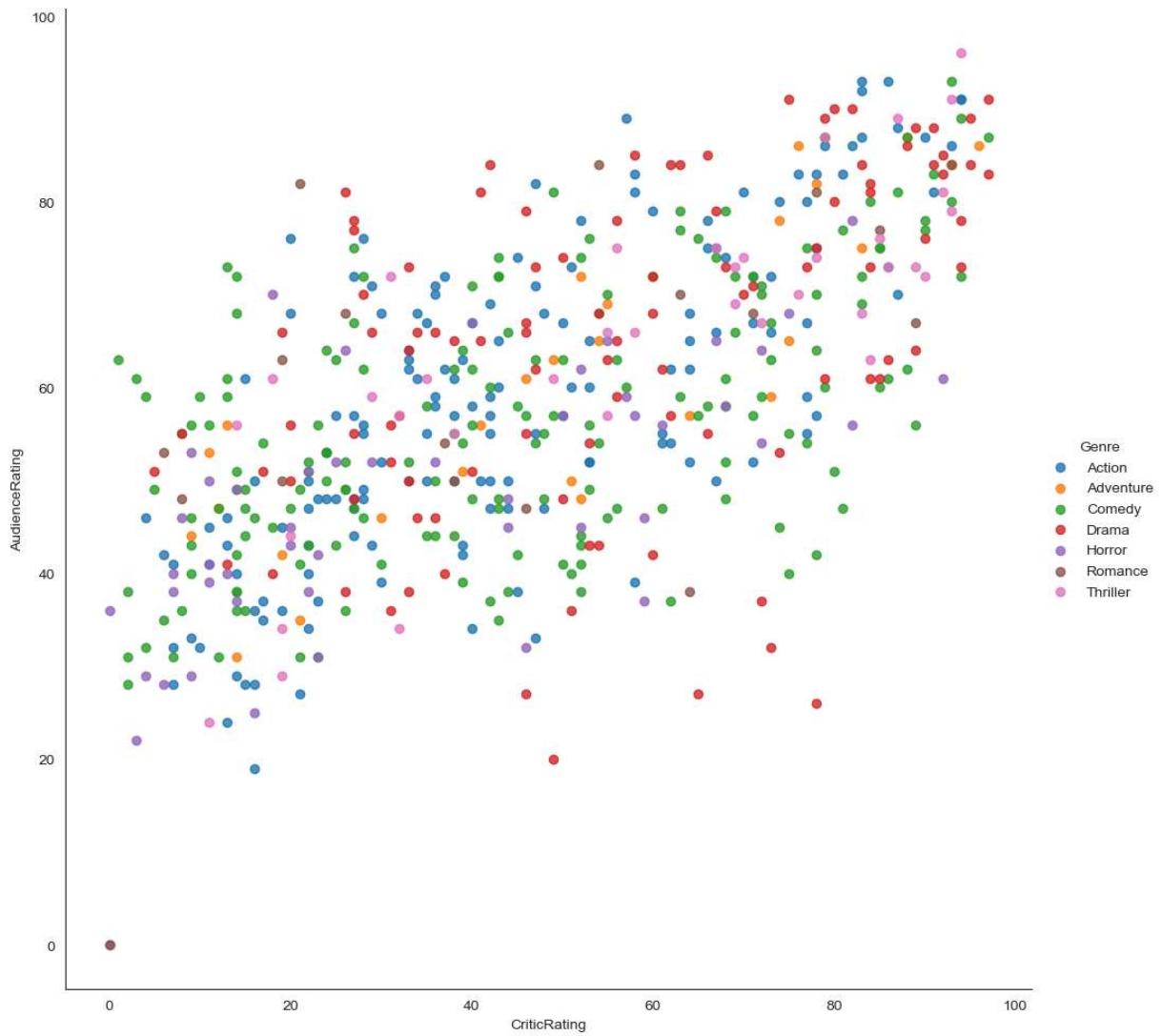
```
In [36]: vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',  
                     fit_reg=False)
```



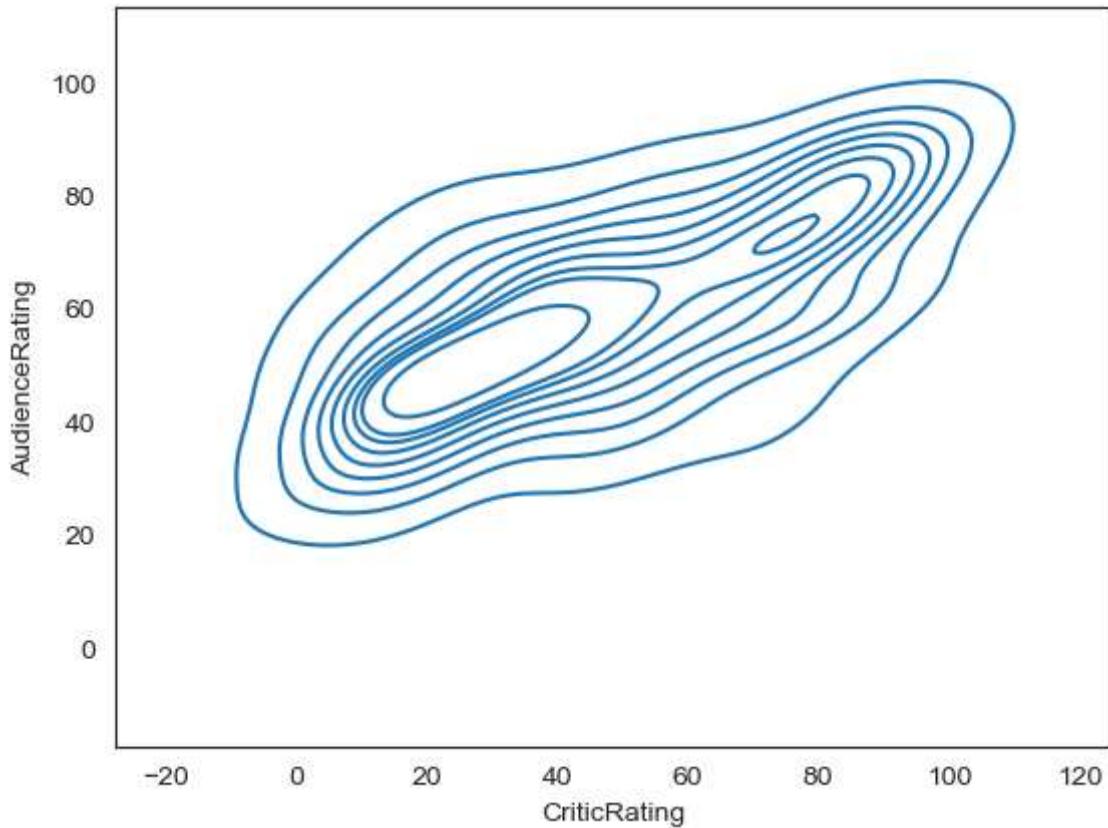
```
In [37]: vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',\n                      fit_reg=False,hue='Genre')
```



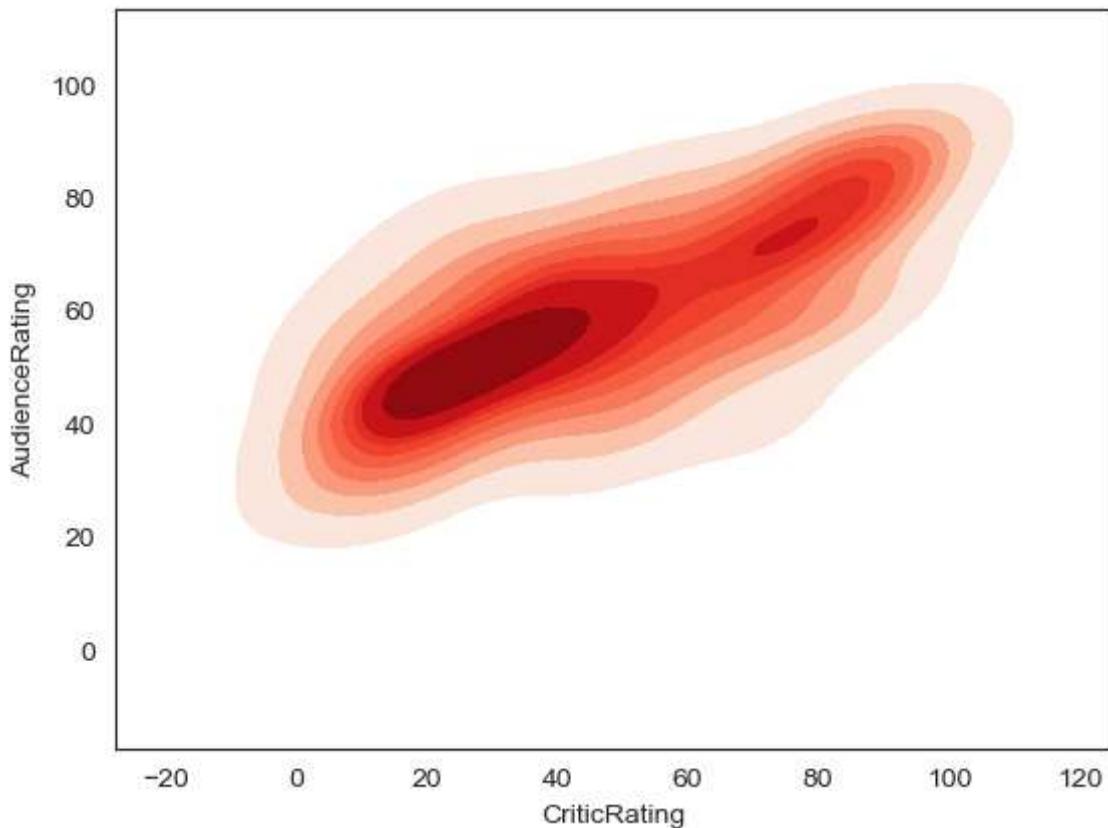
```
In [38]: vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',\n                      fit_reg=False,hue='Genre',height=10,aspect=1)
```



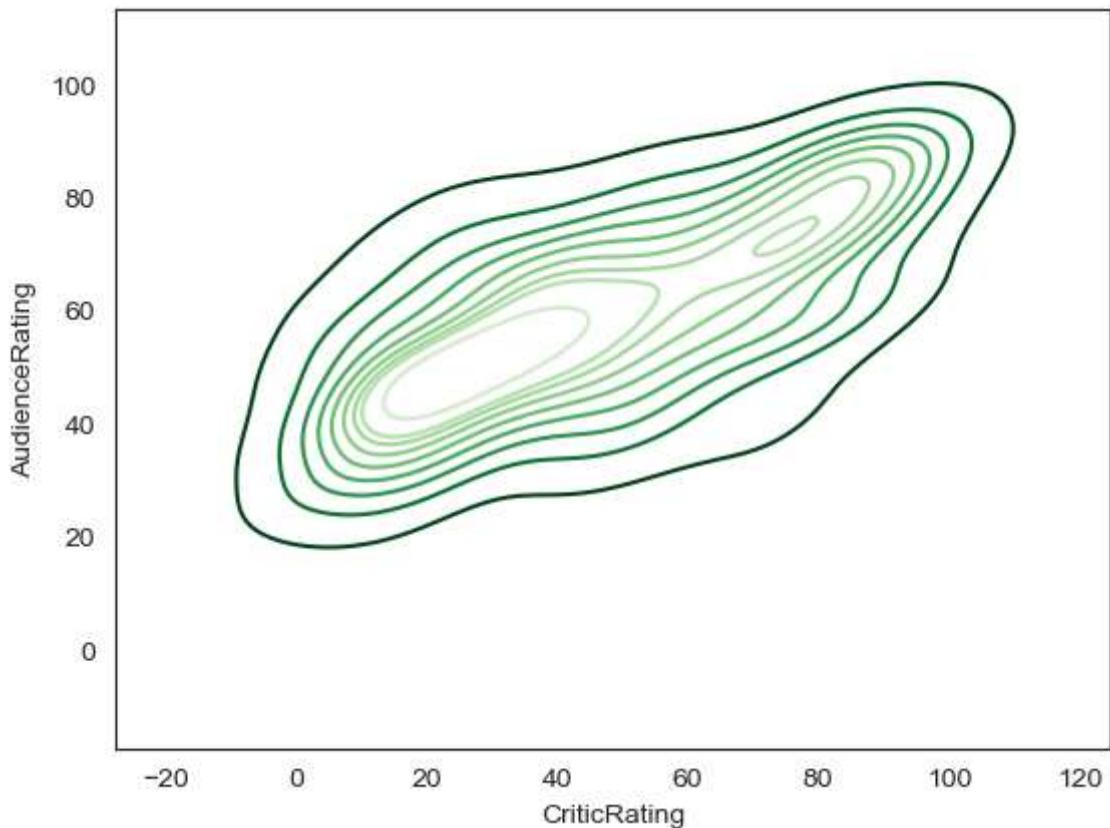
```
In [39]: k1=sns.kdeplot(x='CriticRating',y='AudienceRating',data=movies)
```



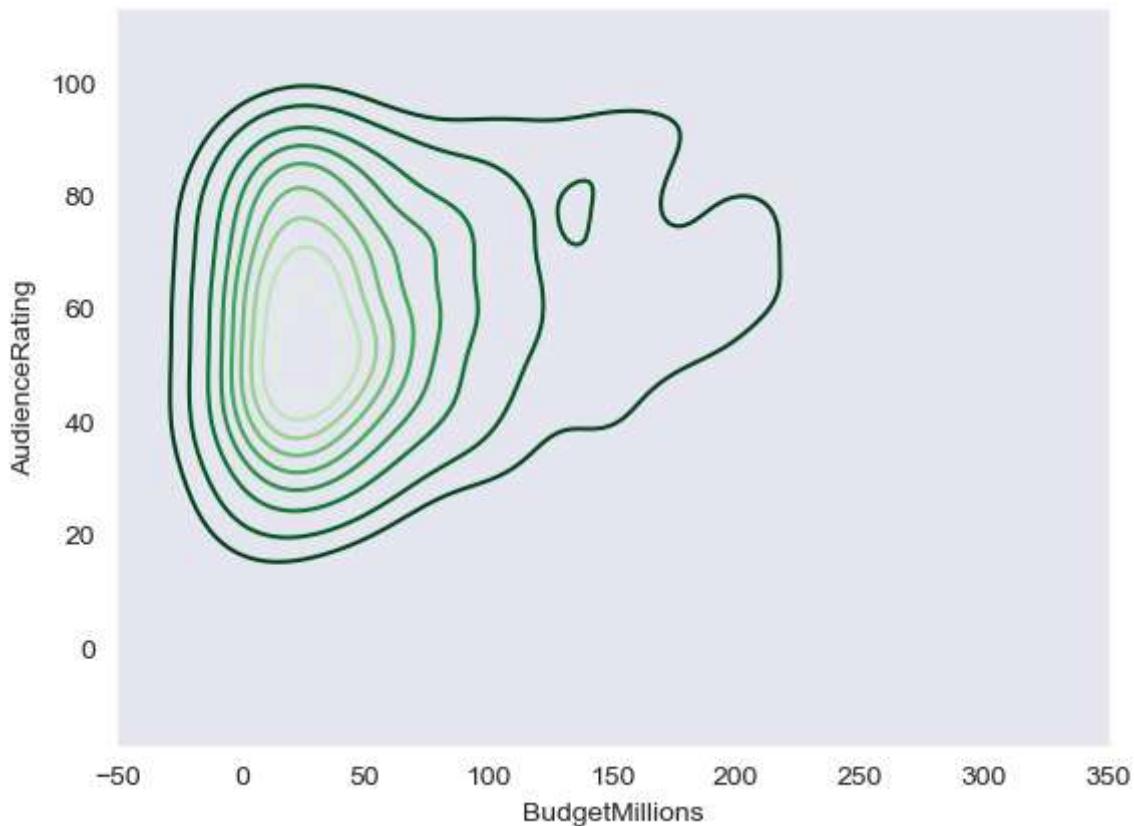
```
In [40]: k1=sns.kdeplot(x = 'CriticRating',y = 'AudienceRating',shade=True,shade_lowest=False)
```



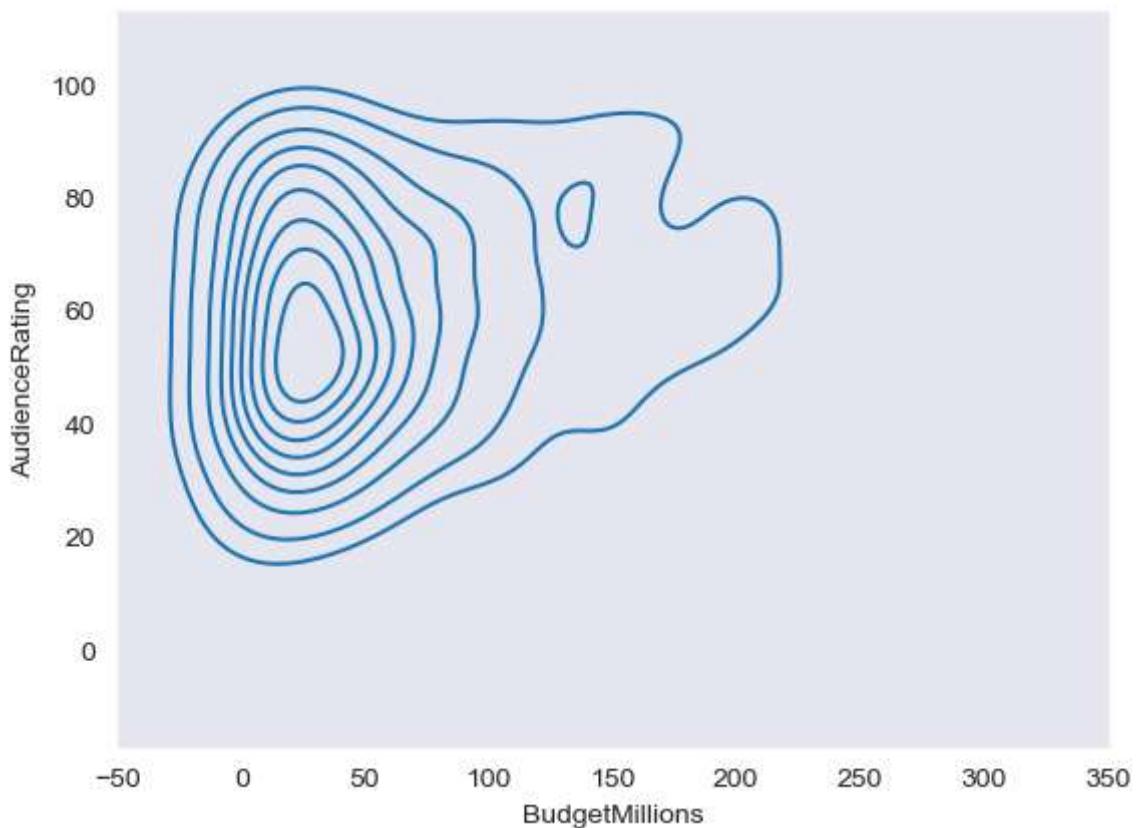
```
In [41]: k2=sns.kdeplot(x='CriticRating',y='AudienceRating',shade_lowest=False,cmap='Greens_
```



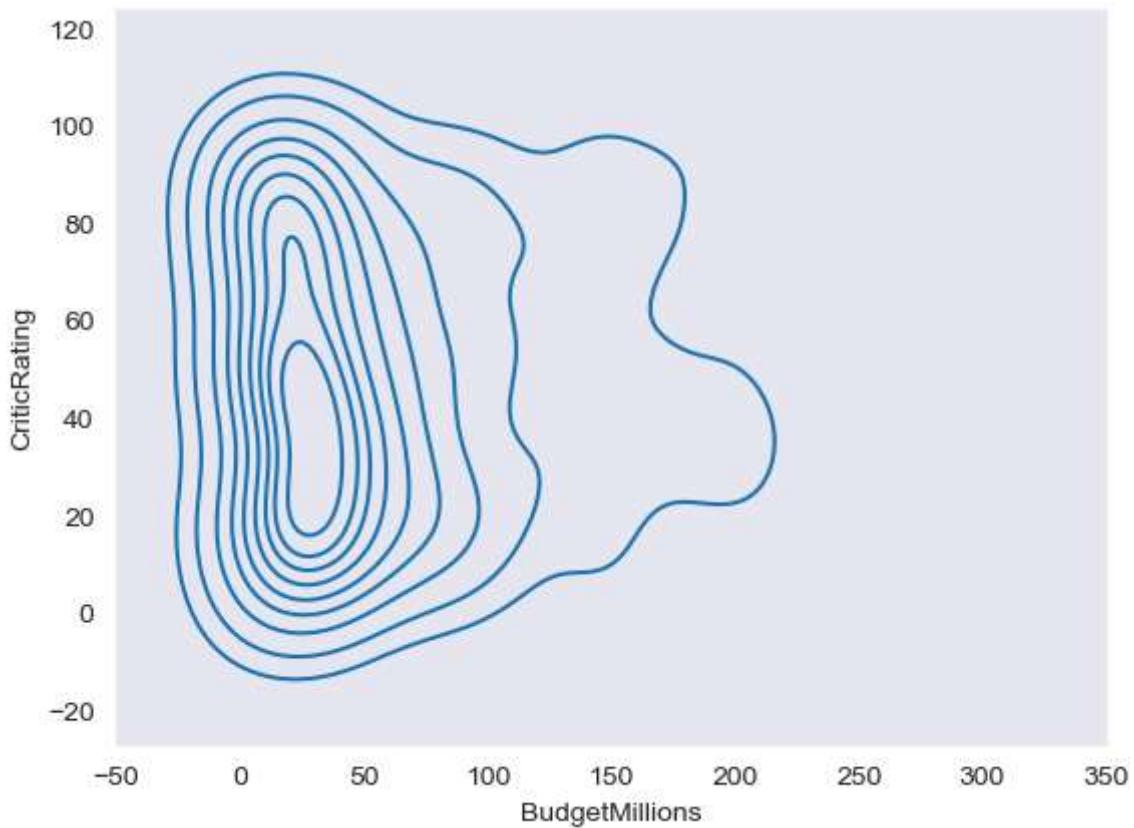
```
In [42]: sns.set_style('dark')
k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,cmap='Greens_r',data
```



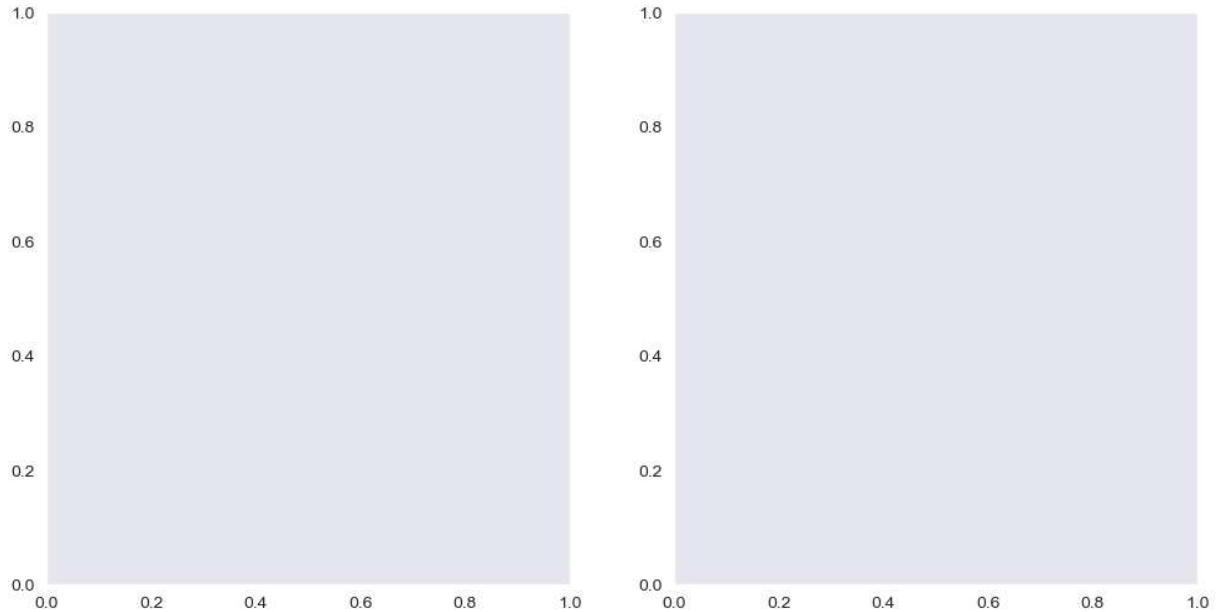
```
In [58]: sns.set_style('dark')
k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating)
```



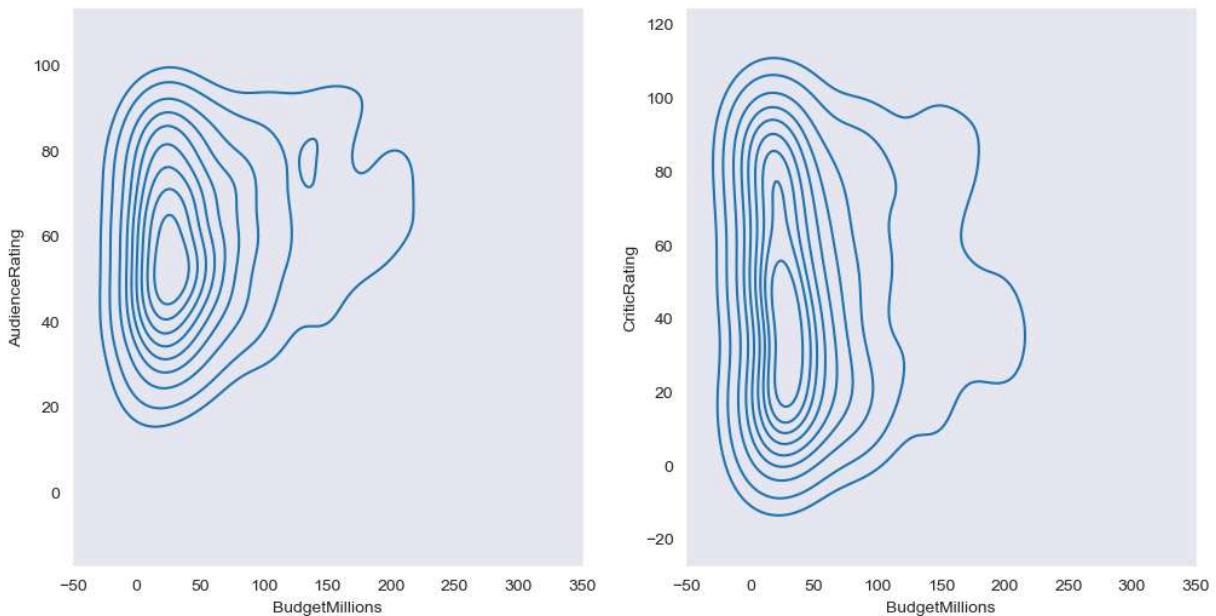
```
In [59]: k2=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating)
```



```
In [60]: f,ax=plt.subplots(1,2,figsize=(12,6))
```



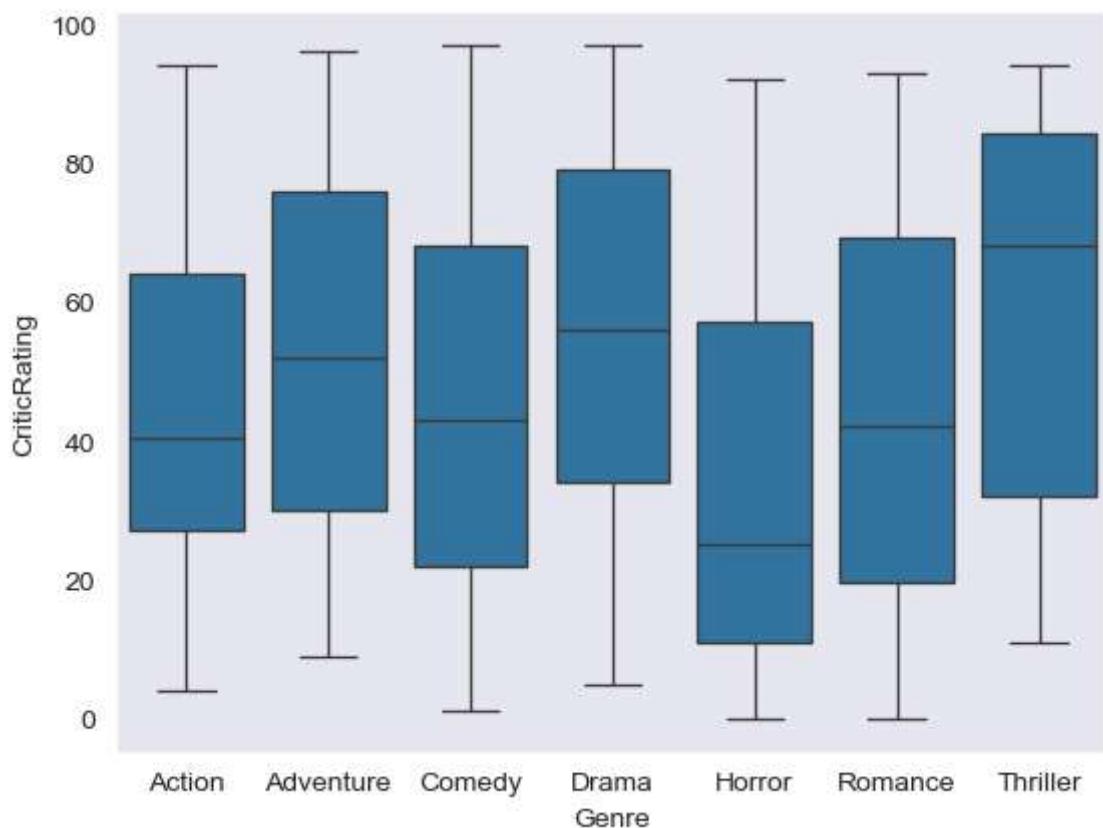
```
In [61]: f,axes=plt.subplots(1,2,figsize=(12,6))
k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,ax=axes[0])
k2=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating,ax=axes[1])
```



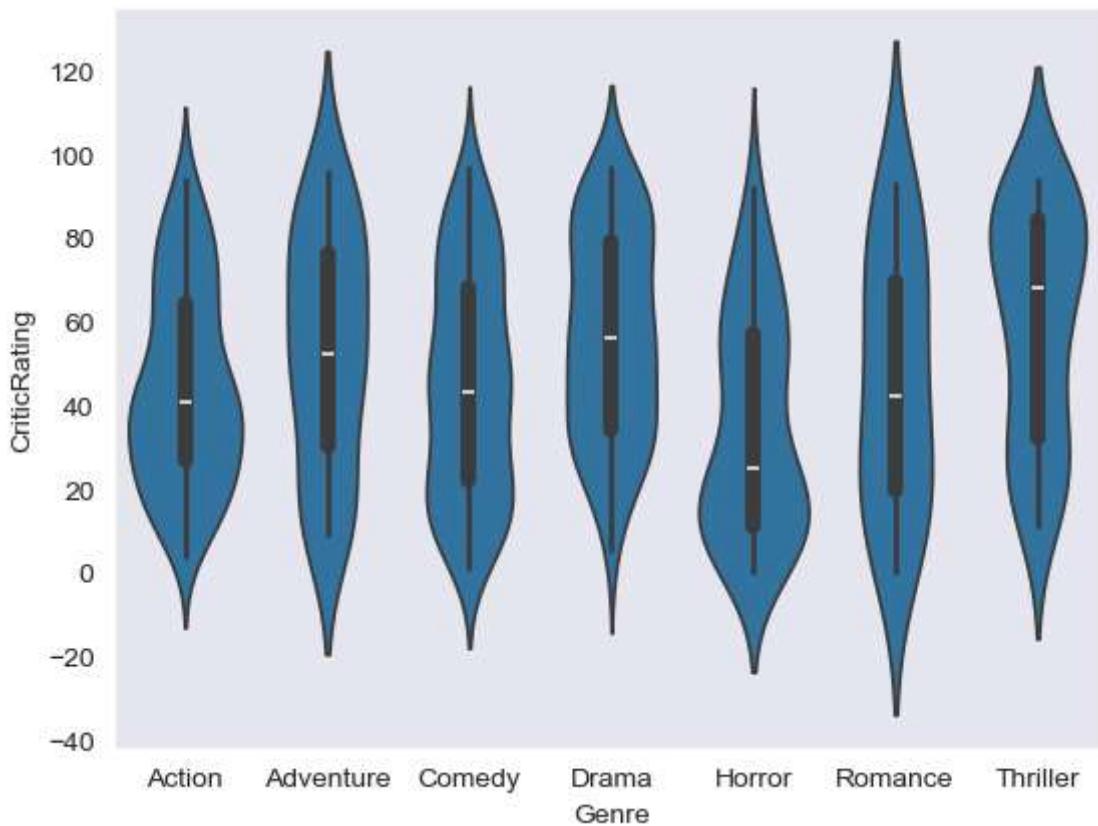
```
In [62]: axes
```

```
Out[62]: array([<Axes: xlabel='BudgetMillions', ylabel='AudienceRating'>,
   <Axes: xlabel='BudgetMillions', ylabel='CriticRating'>],
  dtype=object)
```

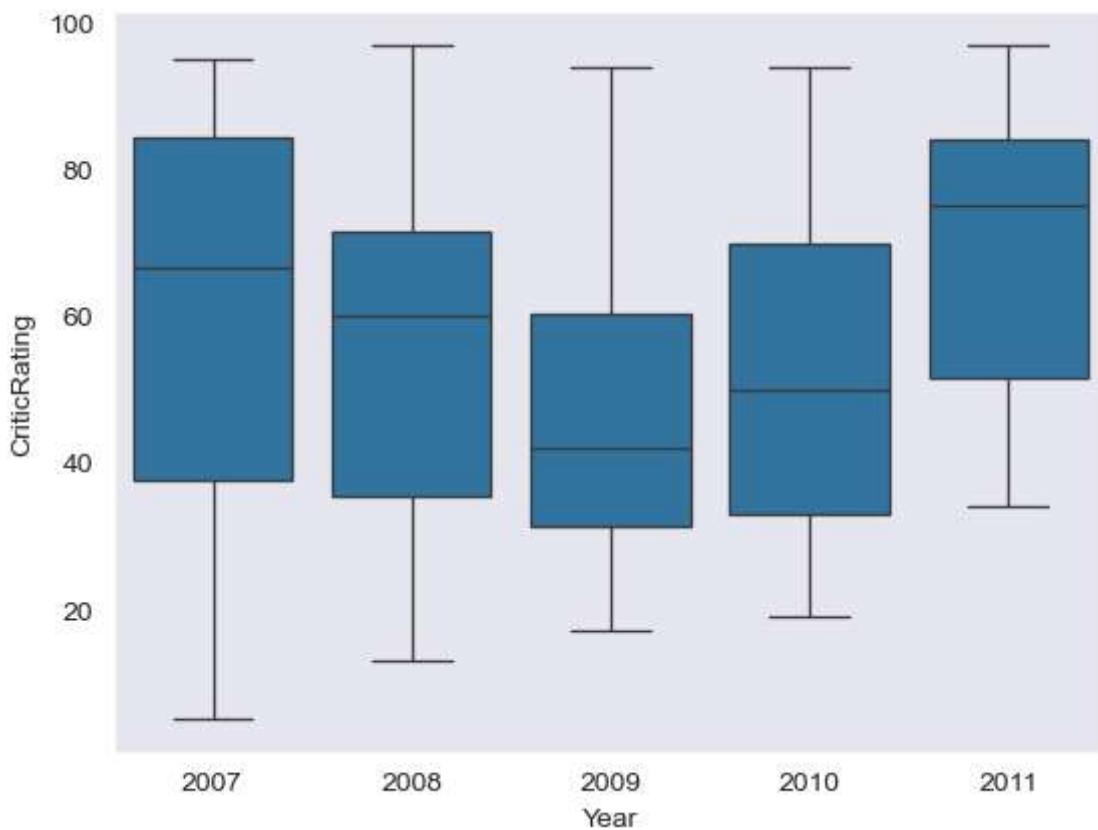
```
In [63]: w=sns.boxplot(data=movies,x='Genre',y='CriticRating')
```



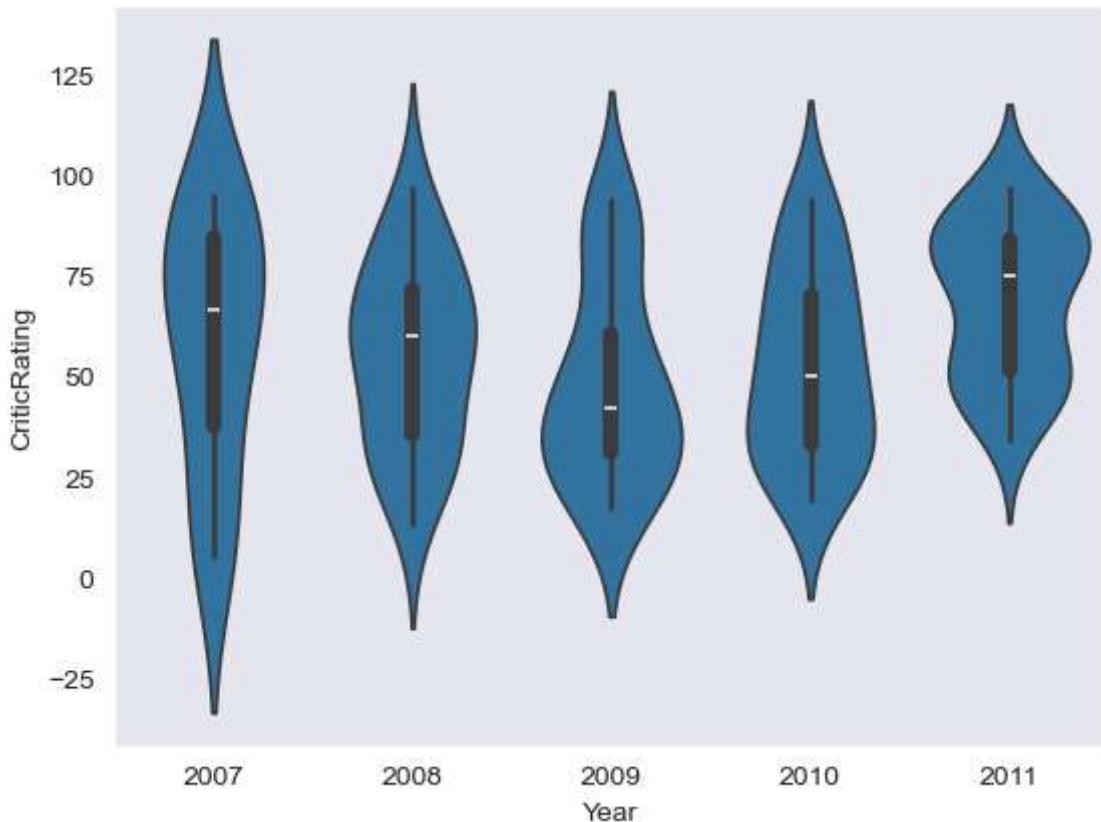
```
In [64]: z=sns.violinplot(data=movies,x='Genre',y='CriticRating')
```



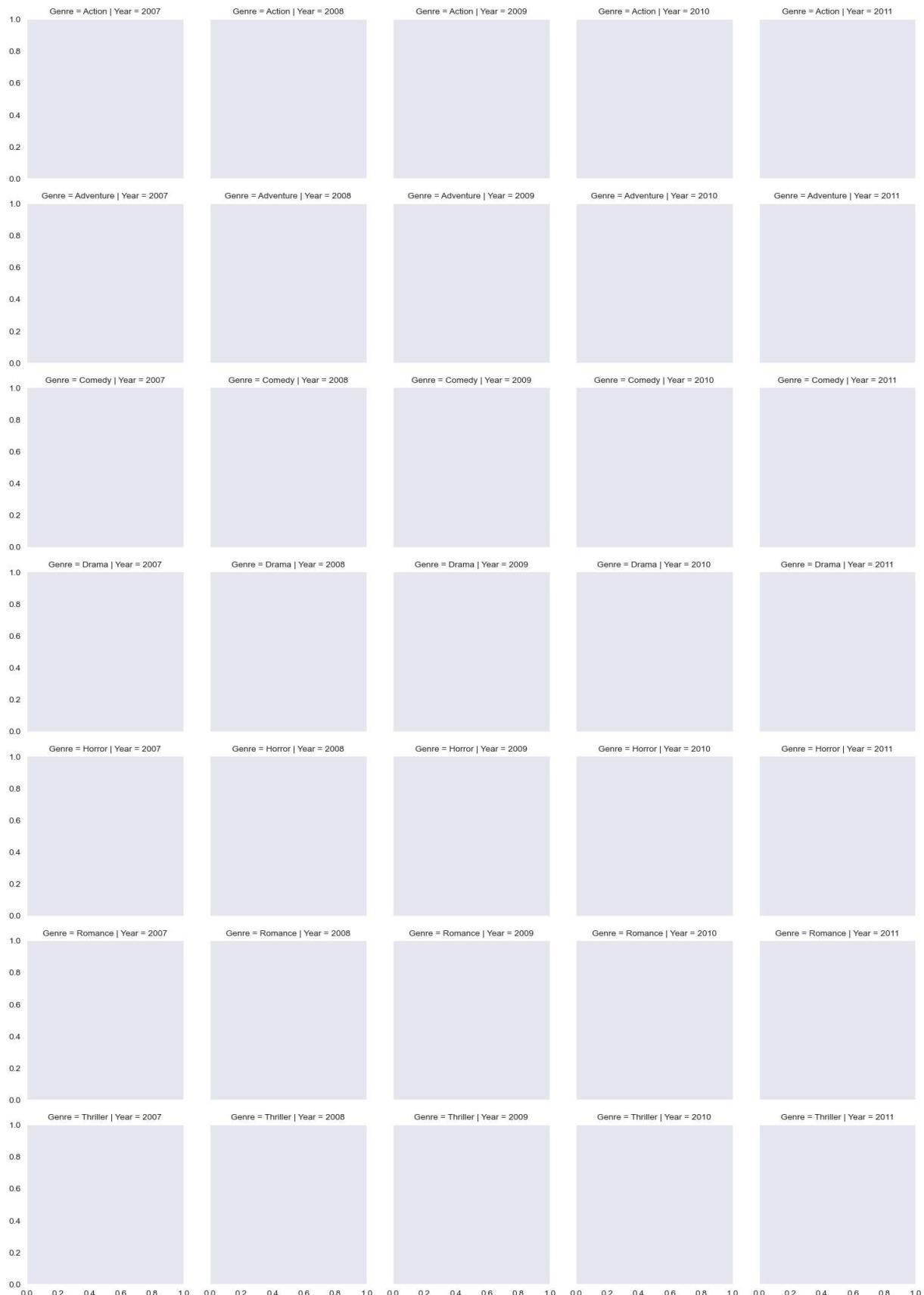
```
In [65]: w1=sns.boxplot(data=movies[movies.Genre=='Drama'],x='Year',y='CriticRating')
```



```
In [66]: z=sns.violinplot(data=movies[movies.Genre=='Drama'],x='Year',y='CriticRating')
```

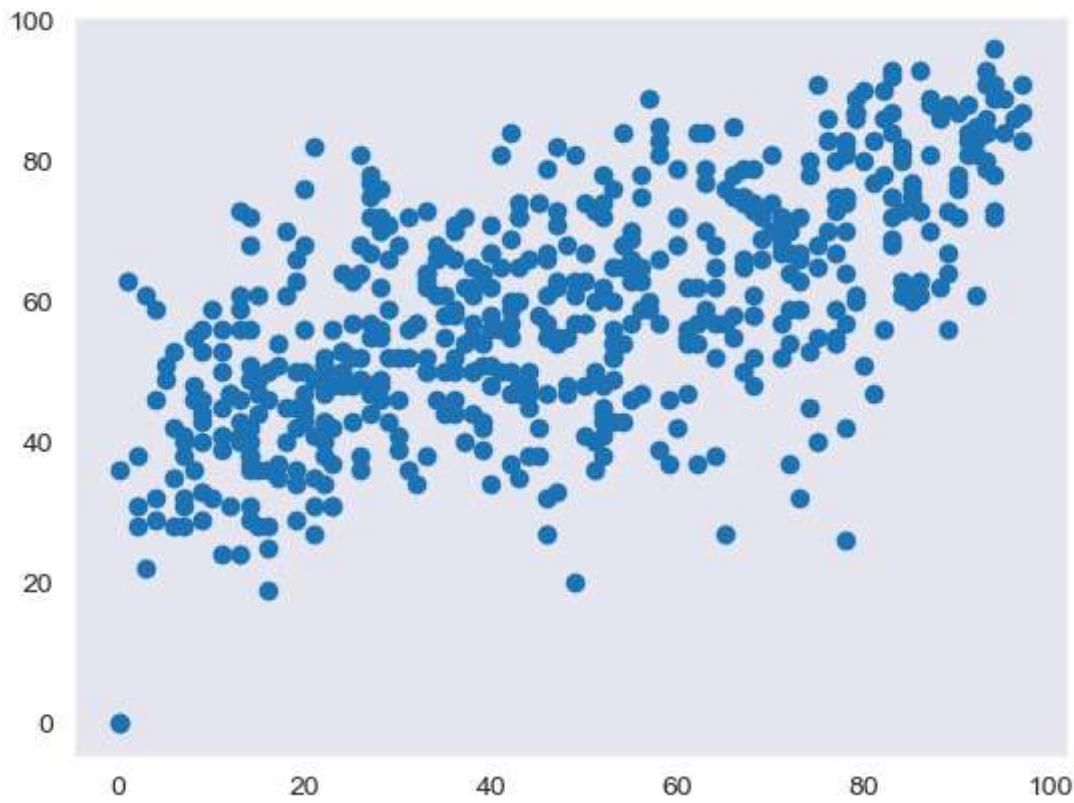


```
In [67]: g=sns.FacetGrid(movies, row='Genre', col='Year', hue='Genre')
```

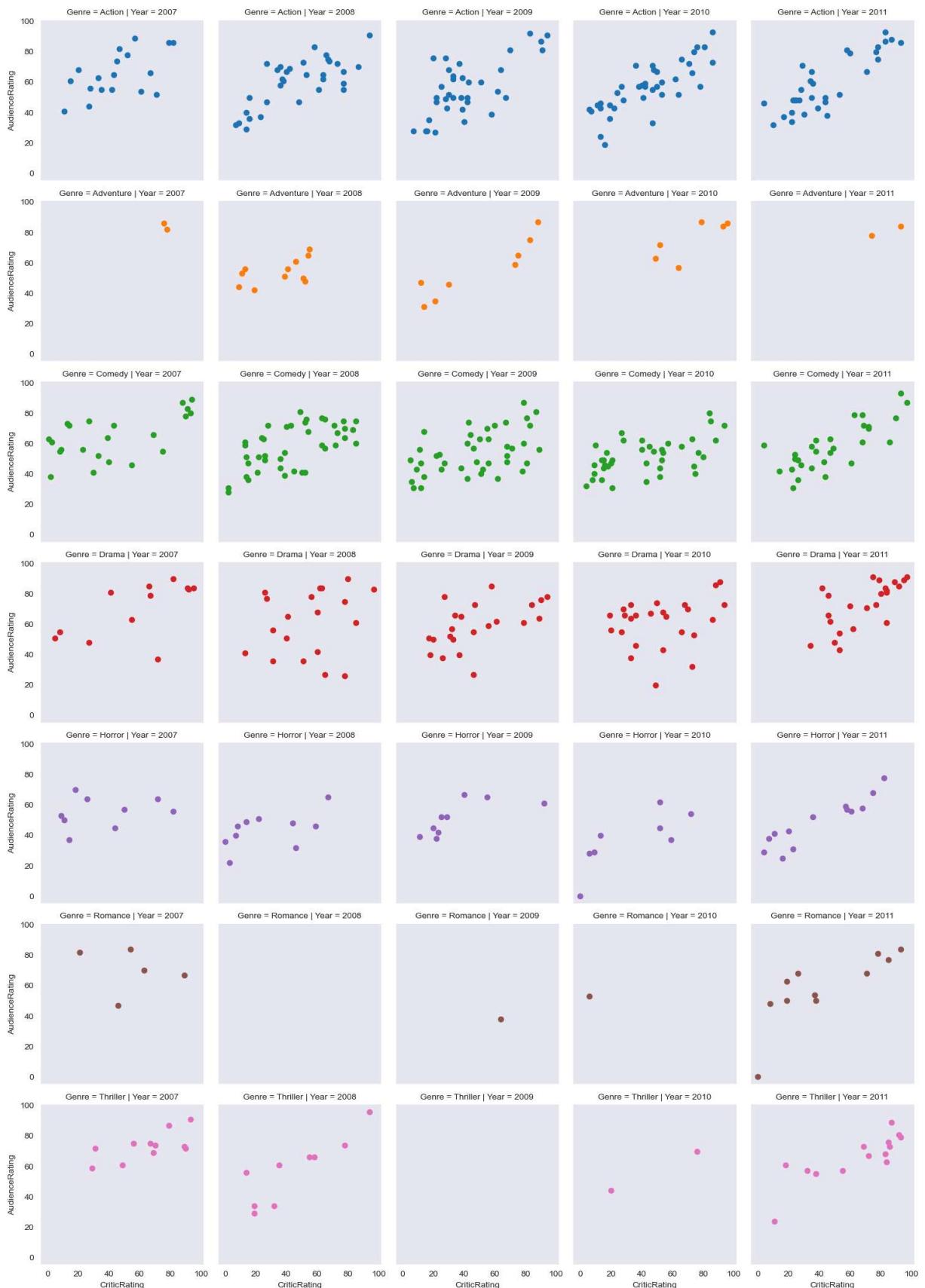


```
In [68]: plt.scatter(movies.CriticRating,movies.AudienceRating)
```

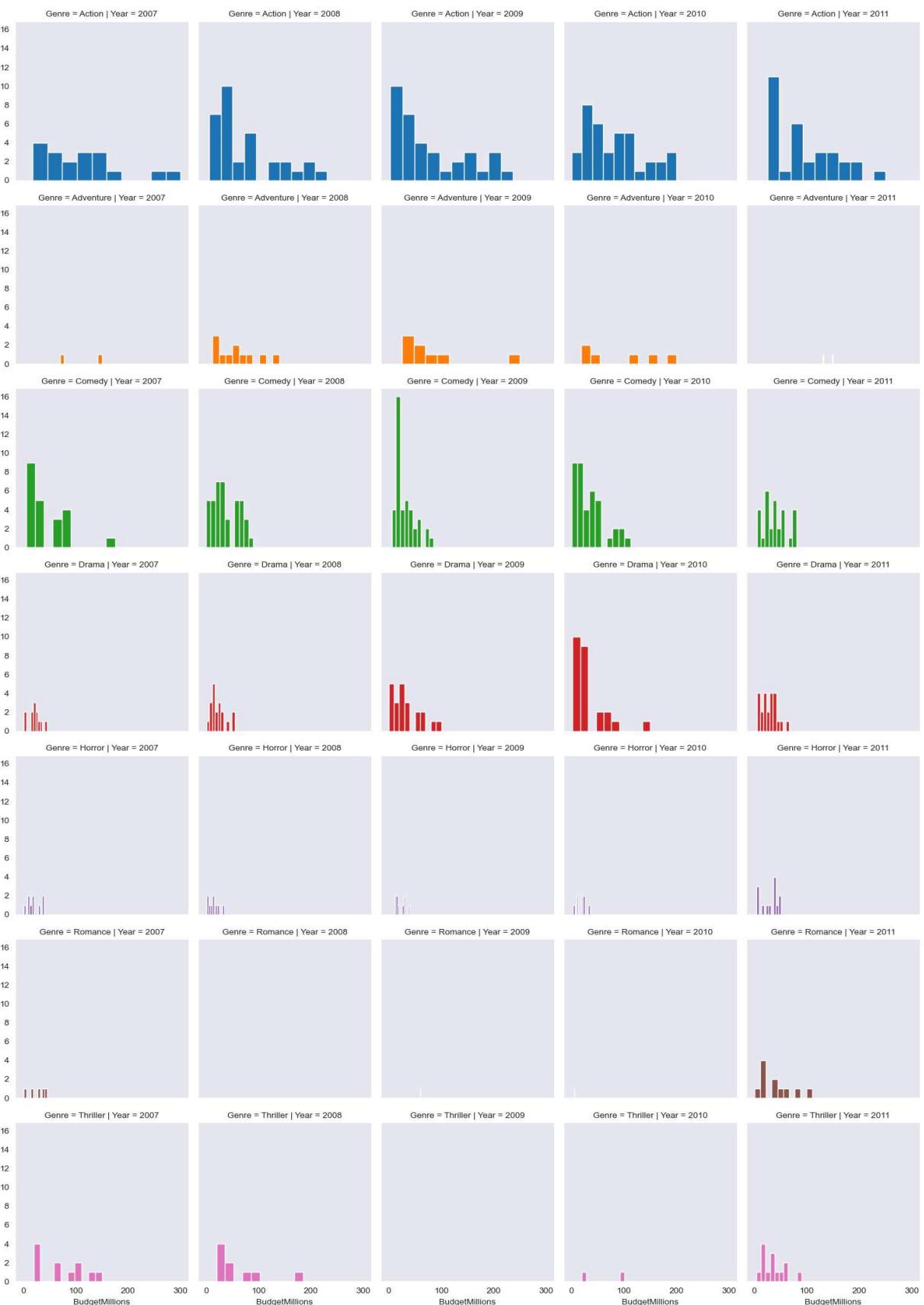
```
Out[68]: <matplotlib.collections.PathCollection at 0x1b5f10b41d0>
```



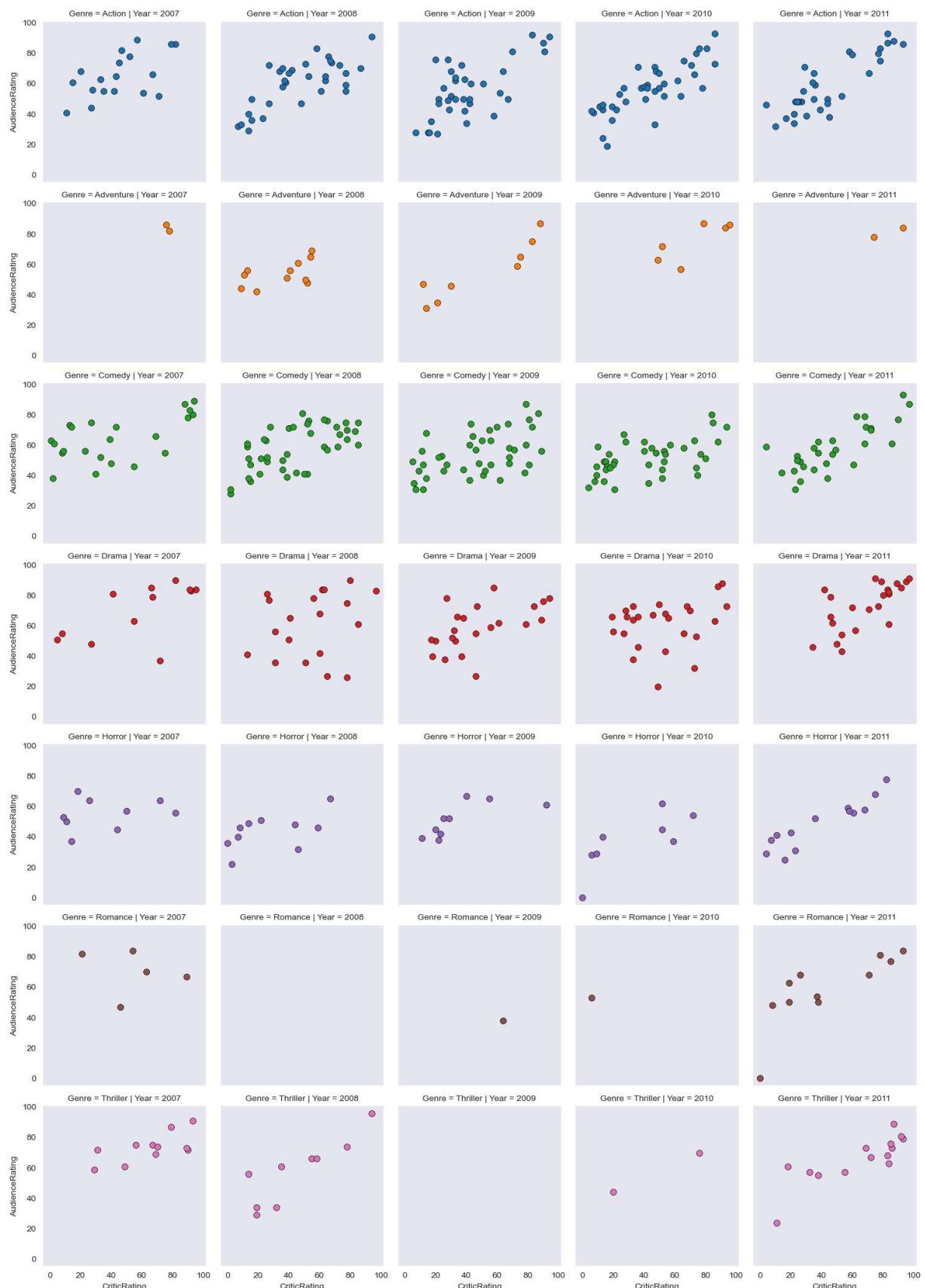
```
In [69]: g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
g=g.map(plt.scatter,'CriticRating','AudienceRating')
```



```
In [56]: g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
g=g.map(plt.hist,'BudgetMillions')
```



```
In [57]: g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
kws=dict(s=50,linewidth=0.5,edgecolor='black')
g=g.map(plt.scatter,'CriticRating','AudienceRating',**kws)
```



```
In [70]: sns.set_style('darkgrid')
f, axes = plt.subplots(2,2, figsize = (15,15))

k1 = sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,ax=axes[0,0])
```

```

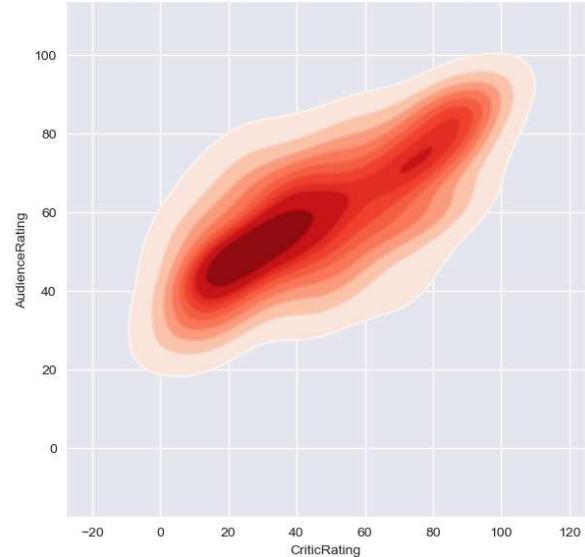
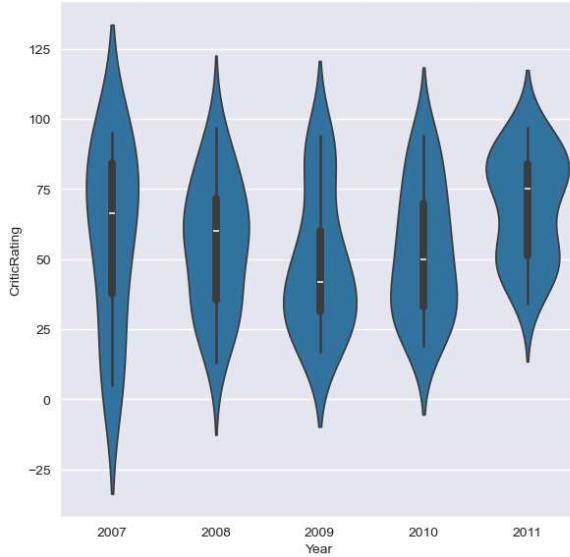
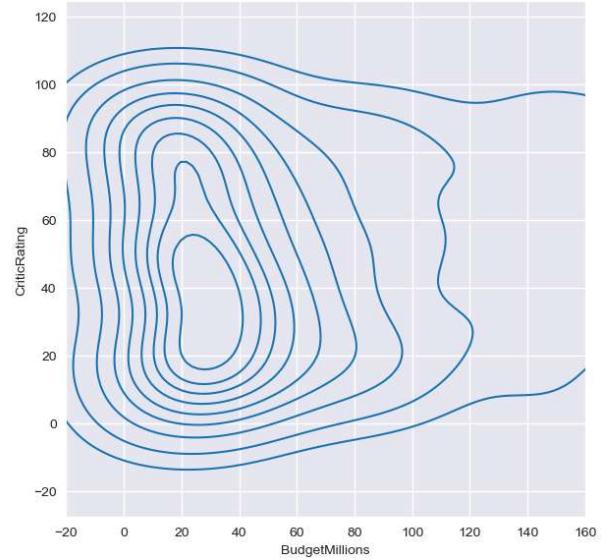
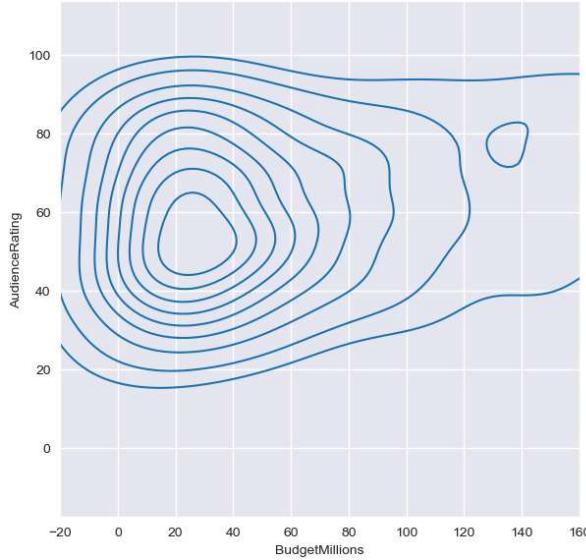
k2 = sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating,ax = axes[0,1])

k1.set(xlim=(-20,160))
k2.set(xlim=(-20,160))

z = sns.violinplot(data=movies[movies.Genre=='Drama'], x='Year', y = 'CriticRating')

k4 = sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,shade = True,shade_l
k4b = sns.kdeplot(x=movies.CriticRating,y= movies.AudienceRating,cmap='Reds',ax = a
plt.show()

```



In [72]: # How can you style your dashboard using different color map

```

# python is not vectorize programming language
# Building dashboards (dashboard - combination of chats)

sns.set_style('dark',{'axes.facecolor':'black'})
f, axes = plt.subplots(2,2, figsize = (15,15))

```

```
#plot [0,0]
k1 = sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating, \
                   shade = True, shade_lowest=True,cmap = 'inferno', \
                   ax = axes[0,0])
k1b = sns.kdeplot(x=movies.BudgetMillions, y=movies.AudienceRating, \
                   cmap = 'cool',ax = axes[0,0])

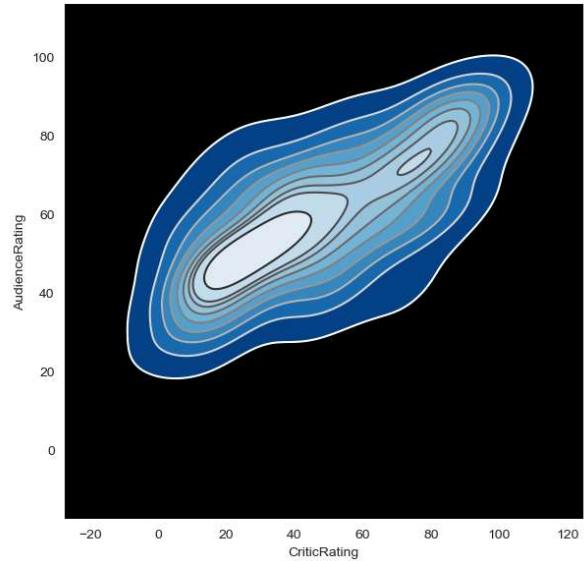
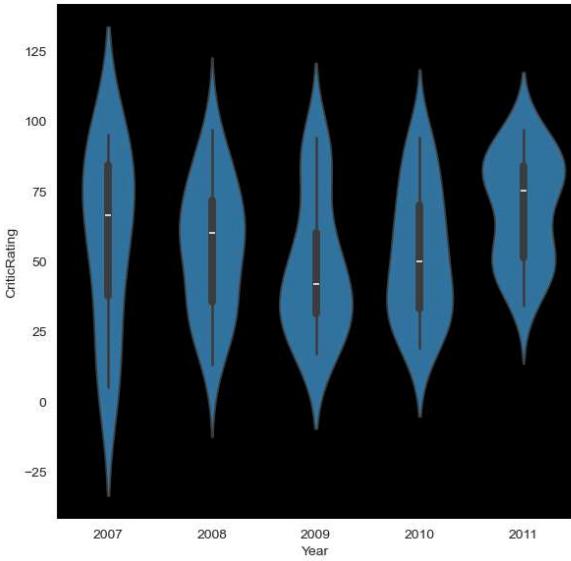
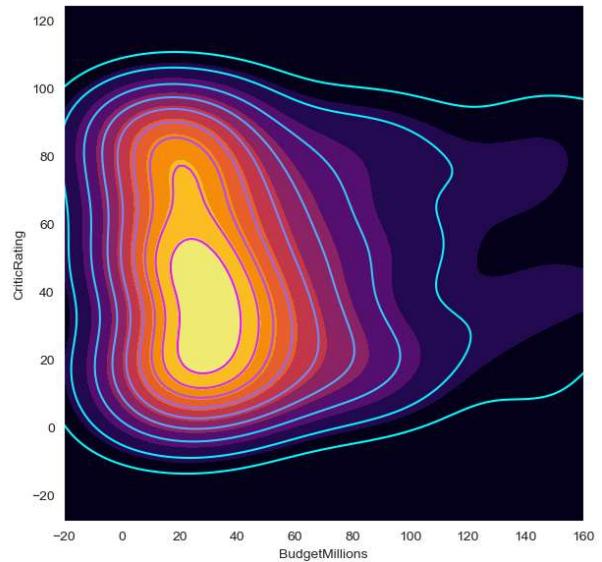
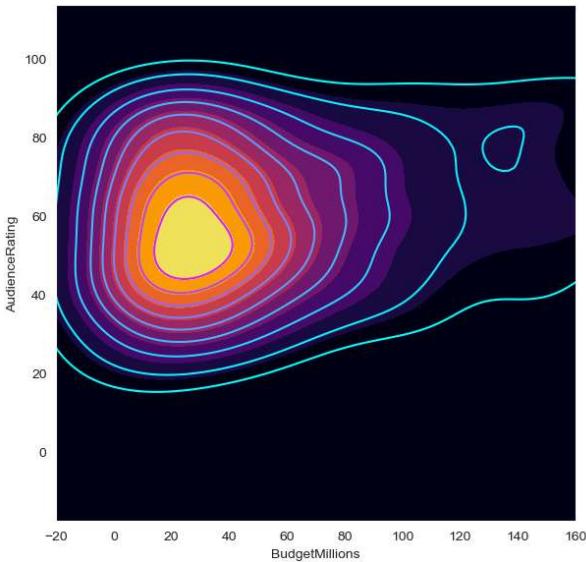
#plot [0,1]
k2 = sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating, \
                   shade=True, shade_lowest=True, cmap='inferno', \
                   ax = axes[0,1])
k2b = sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating, \
                   cmap = 'cool', ax = axes[0,1])

#plot[1,0]
z = sns.violinplot(data=movies[movies.Genre=='Drama'], \
                     x='Year', y = 'CriticRating', ax=axes[1,0])

#plot[1,1]
k4 = sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating, \
                   shade = True,shade_lowest=False,cmap='Blues_r', \
                   ax=axes[1,1])
k4b = sns.kdeplot(x=movies.CriticRating, y=movies.AudienceRating, \
                   cmap='gist_gray_r',ax = axes[1,1])

k1.set(xlim=(-20,160))
k2.set(xlim=(-20,160))

plt.show()
```



In []: