

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

```
In [2]: movies=pd.read_csv(r"C:\Users\Twinkle\OneDrive\Desktop\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]: type(movies)
```

```
Out[4]: pandas.core.frame.DataFrame
```

```
In [5]: movies
```

Out[5]:

| | 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 [6]: `len(movies)`

Out[6]: 559

In [7]: `import numpy
print(numpy.__version__)`

2.1.3

In [8]: `import pandas
print(pandas.__version__)`

2.2.3

In [9]: `movies.columns`

Out[9]: `Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
 'Budget (million $)', 'Year of release'],
 dtype='object')`

In [10]: `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 [11]: movies.shape
```

```
Out[11]: (559, 6)
```

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

```
Out[12]:
```

| | 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 [13]: movies.tail()
```

```
Out[13]:
```

| | 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 [14]: movies.columns=['Film', 'Genre', 'CriticRating','AudianceRating', 'BudgetMillions',
```

```
In [15]: movies.head(1)
```

```
Out[15]:
```

| | Film | Genre | CriticRating | AudianceRating | BudgetMillions | Years |
|---|----------------------|--------|--------------|----------------|----------------|-------|
| 0 | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |

```
In [16]: movies.shape
```

```
Out[16]: (559, 6)
```

```
In [17]: movies.describe()
```

| | CriticRating | AudianceRating | BudgetMillions | Years |
|-------|--------------|----------------|----------------|-------------|
| 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 [18]: movies.describe().transpose()
```

| | count | mean | std | min | 25% | 50% | 75% | max |
|----------------|-------|-------------|-----------|--------|--------|--------|--------|--------|
| CriticRating | 559.0 | 47.309481 | 26.413091 | 0.0 | 25.0 | 46.0 | 70.0 | 97.0 |
| AudianceRating | 559.0 | 58.744186 | 16.826887 | 0.0 | 47.0 | 58.0 | 72.0 | 96.0 |
| BudgetMillions | 559.0 | 50.236136 | 48.731817 | 0.0 | 20.0 | 35.0 | 65.0 | 300.0 |
| Years | 559.0 | 2009.152057 | 1.362632 | 2007.0 | 2008.0 | 2009.0 | 2010.0 | 2011.0 |

```
In [19]: 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   AudianceRating    559 non-null    int64  
 4   BudgetMillions    559 non-null    int64  
 5   Years             559 non-null    int64  
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

```
In [20]: movies.Film=movies.Film.astype('category')
```

```
In [21]: movies.describe()
```

```
Out[21]:
```

| | CriticRating | AudianceRating | BudgetMillions | Years |
|--------------|--------------|----------------|----------------|-------------|
| 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 [22]: movies.Genre=movies.Genre.astype('category')
```

```
In [23]: movies.describe()
```

```
Out[23]:
```

| | CriticRating | AudianceRating | BudgetMillions | Years |
|--------------|--------------|----------------|----------------|-------------|
| 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 [24]: movies.Years=movies.Years.astype('category')
```

```
In [25]: movies.describe()
```

Out[25]:

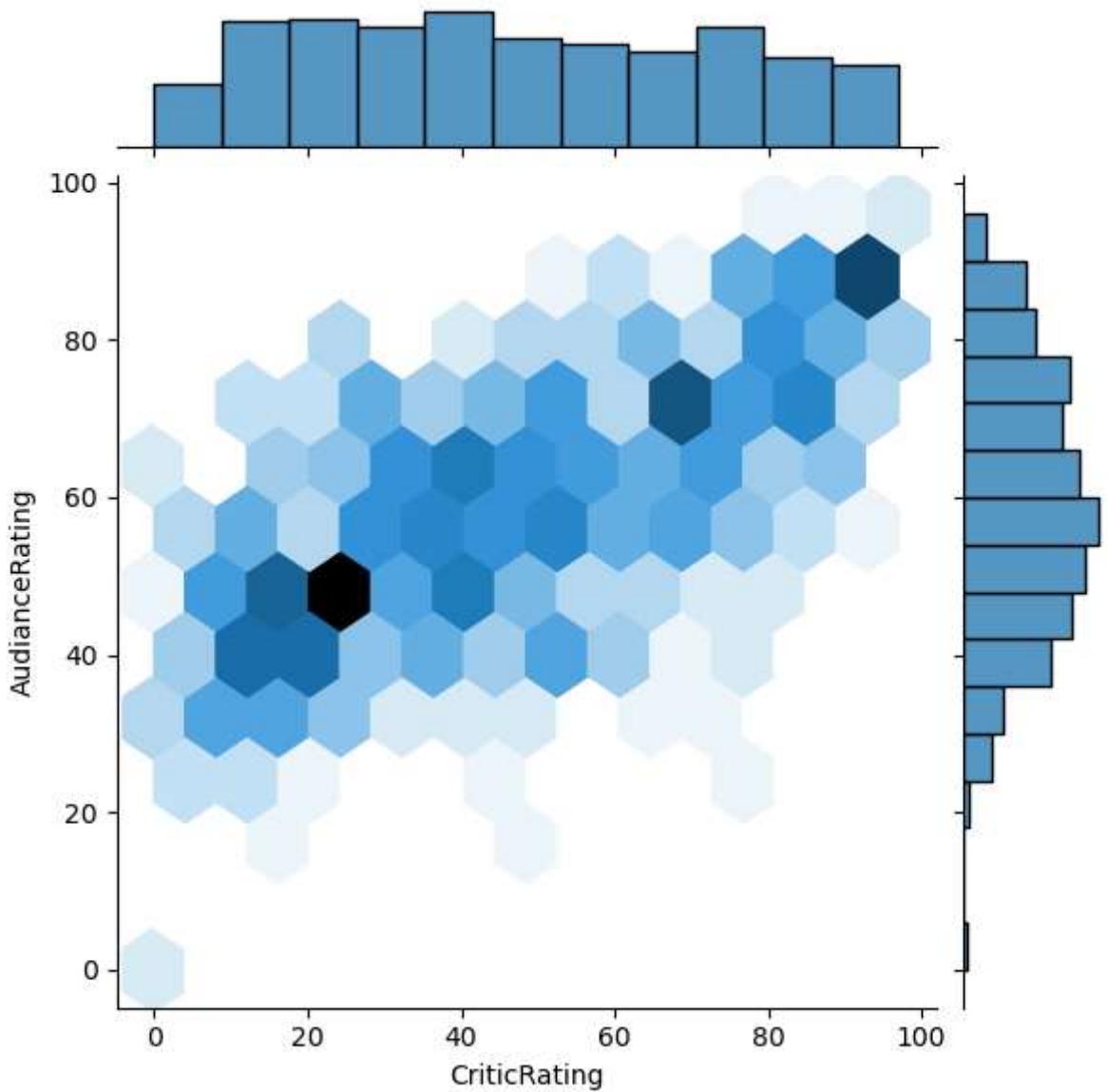
| | CriticRating | AudianceRating | BudgetMillions |
|--------------|--------------|----------------|----------------|
| count | 559.000000 | 559.000000 | 559.000000 |
| mean | 47.309481 | 58.744186 | 50.236136 |
| std | 26.413091 | 16.826887 | 48.731817 |
| min | 0.000000 | 0.000000 | 0.000000 |
| 25% | 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 [26]:

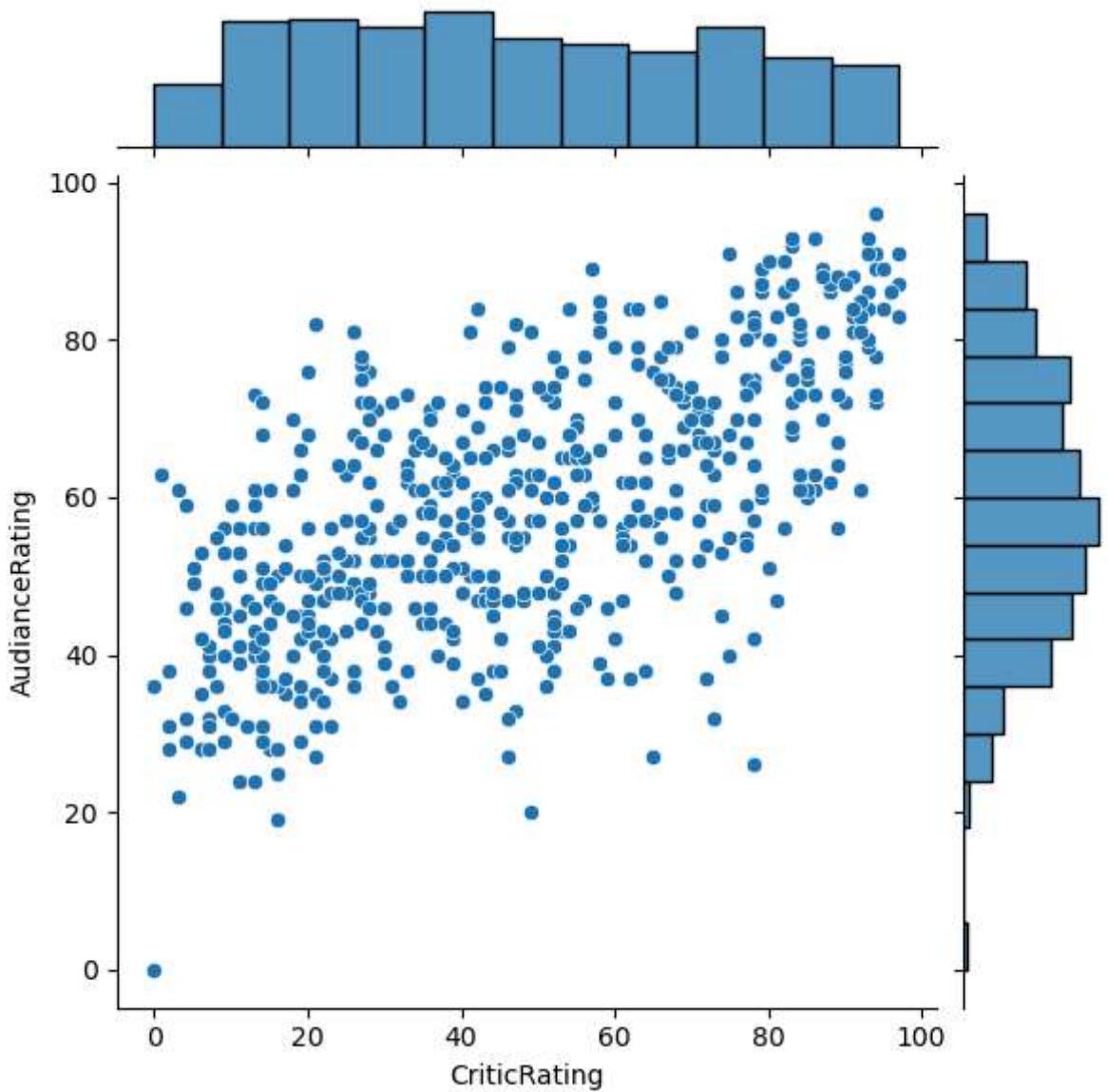
```
import matplotlib as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [27]:

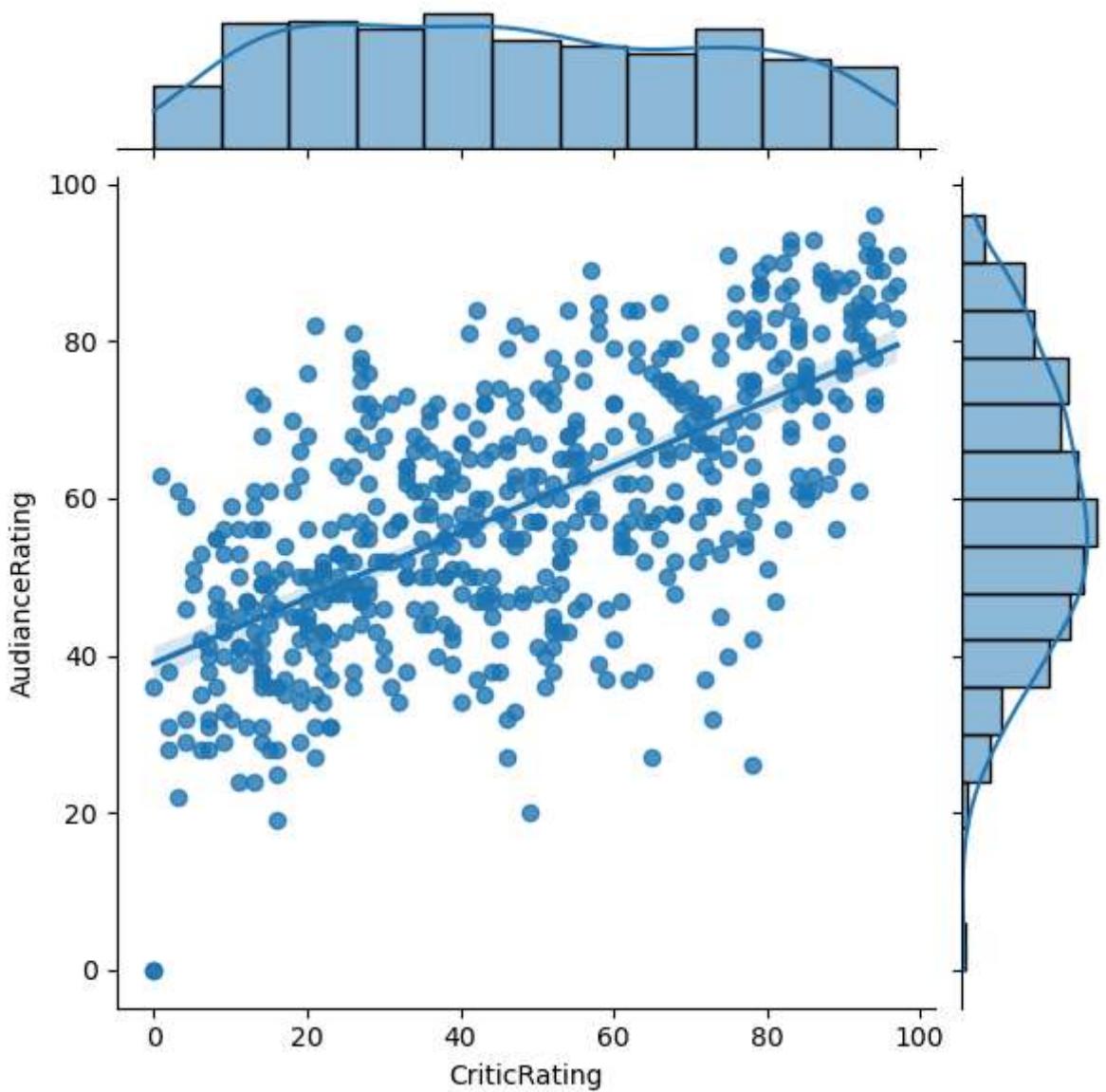
```
j= sns.jointplot(data=movies, x='CriticRating',y='AudianceRating',kind='hex')
```



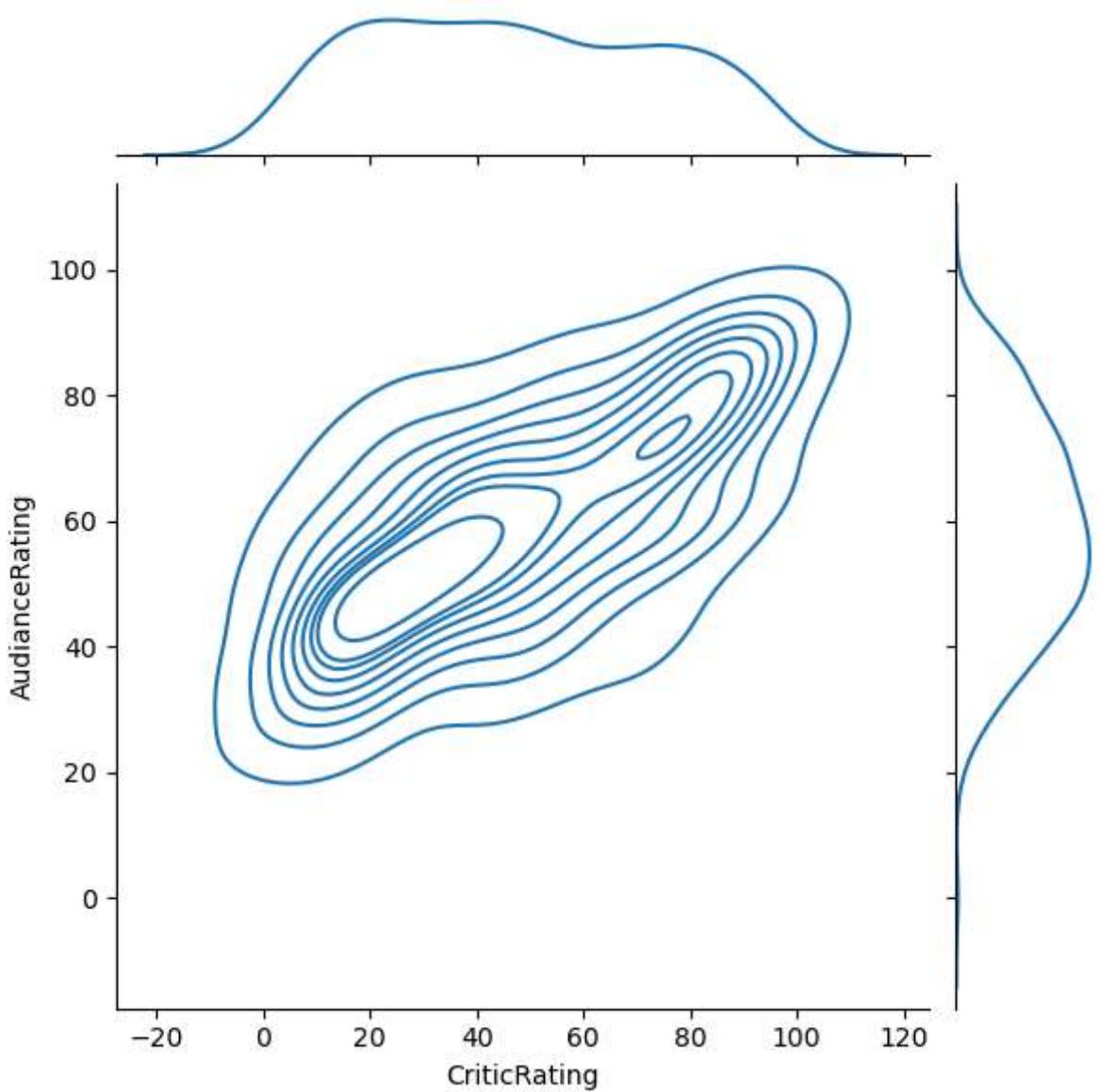
```
In [28]: j= sns.jointplot(data=movies, x='CriticRating',y='AudianceRating',kind='scatter')
```



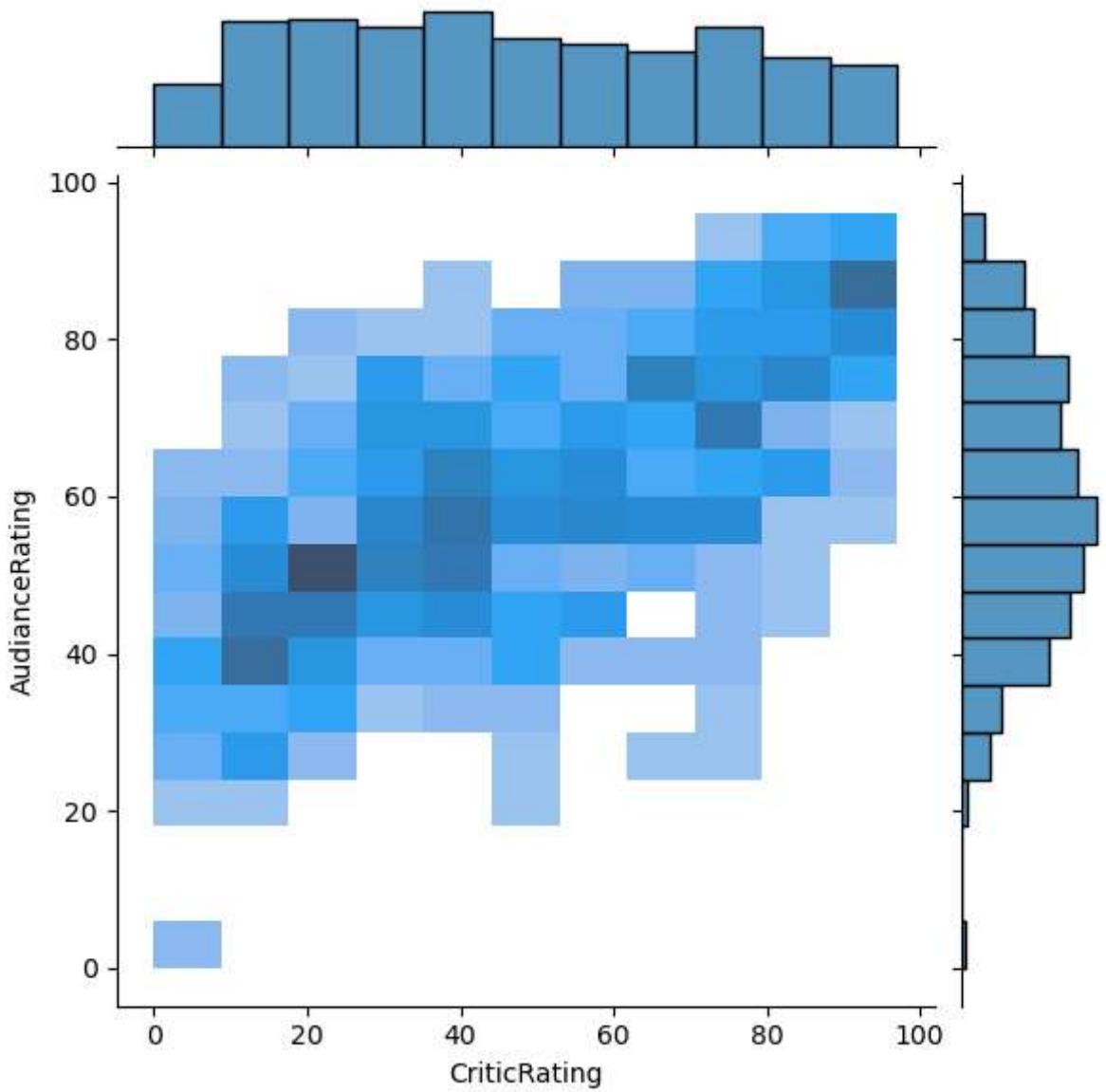
```
In [29]: j= sns.jointplot(data=movies, x='CriticRating',y='AudianceRating',kind='reg')
```



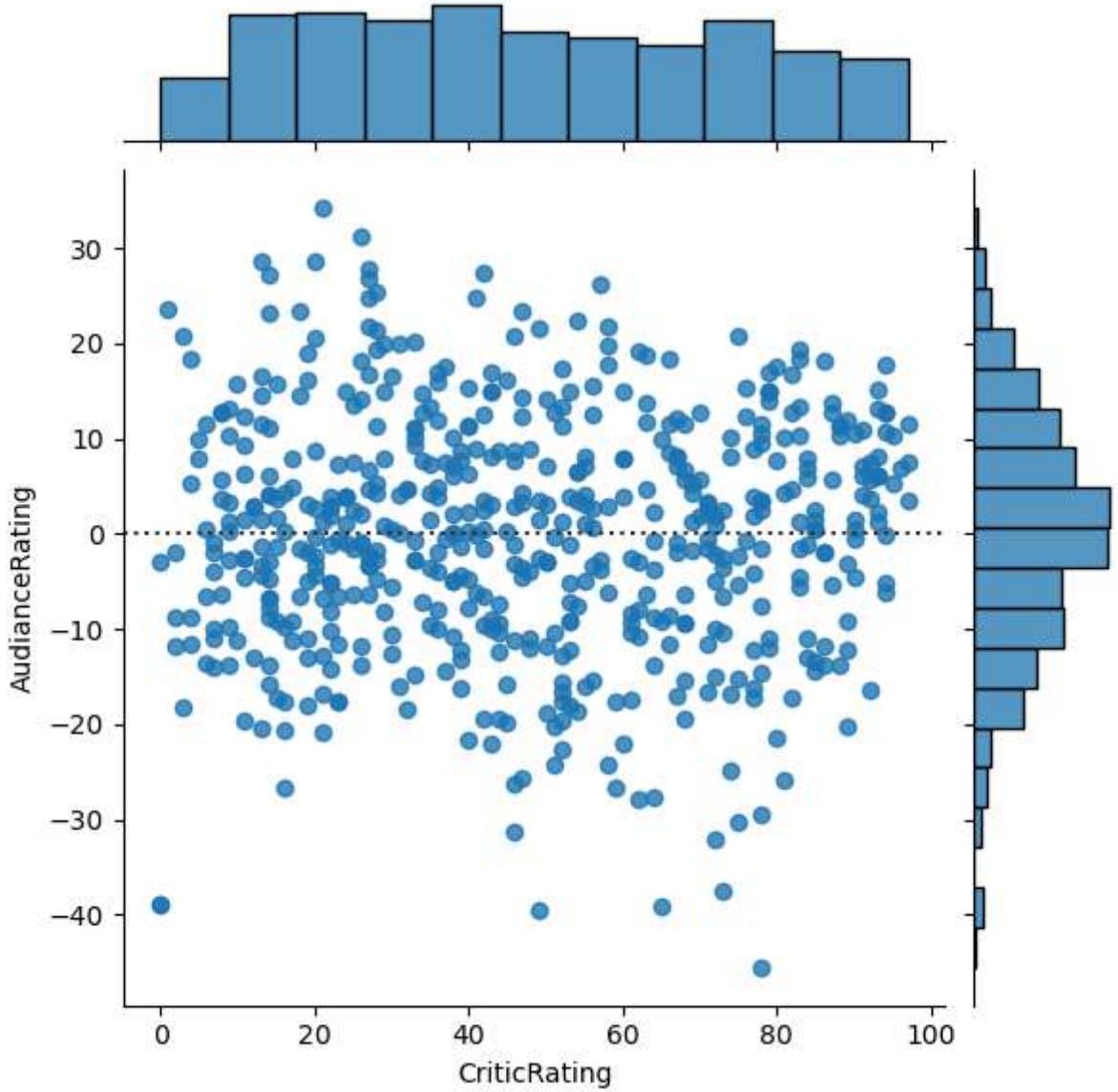
```
In [30]: j= sns.jointplot(data=movies, x='CriticRating',y='AudianceRating',kind='kde')
```



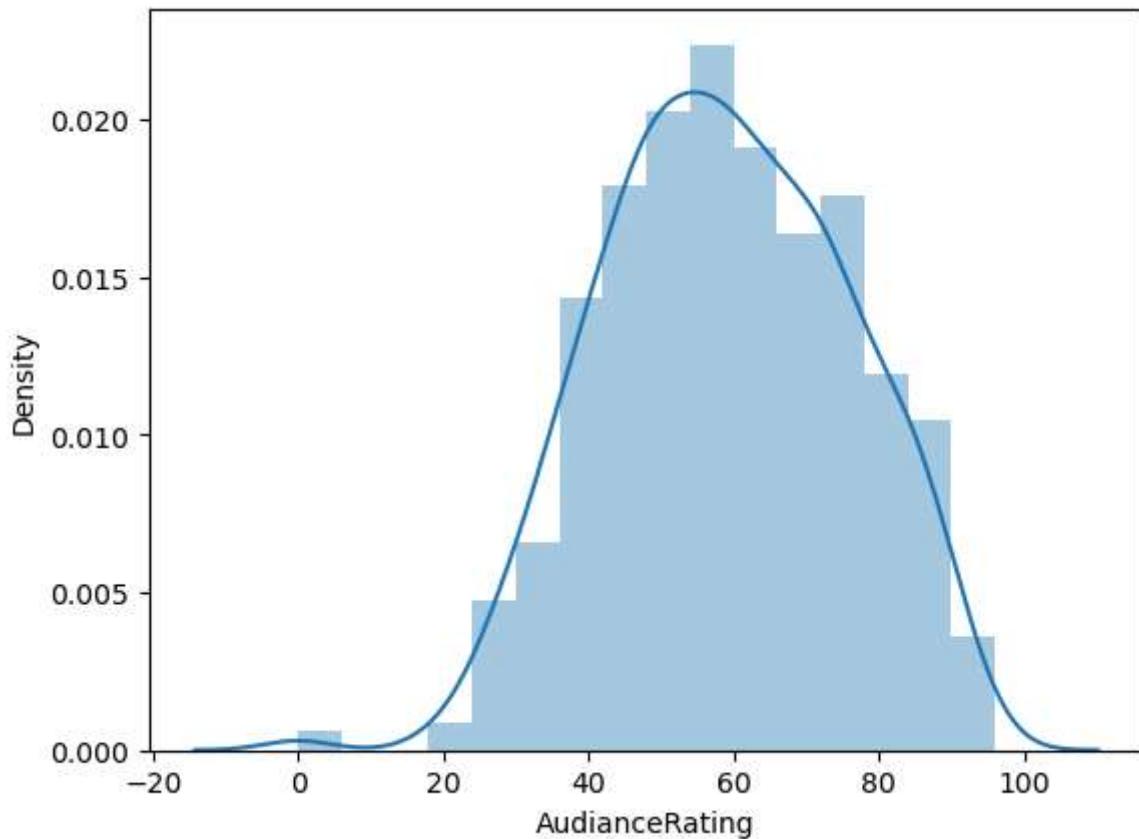
```
In [31]: j= sns.jointplot(data=movies, x='CriticRating',y='AudianceRating',kind='hist')
```



```
In [32]: j= sns.jointplot(data=movies, x='CriticRating',y='AudianceRating',kind='resid')
```

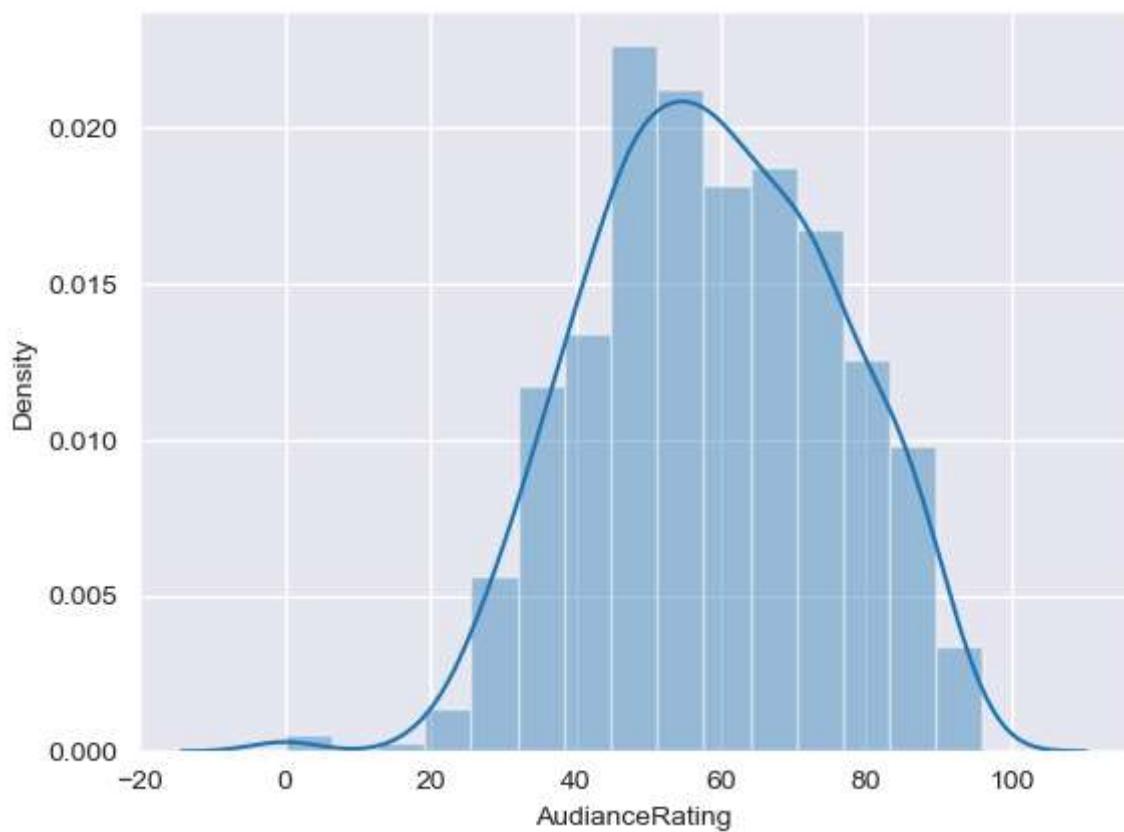


```
In [33]: m1=sns.distplot(movies.AudianceRating)
```



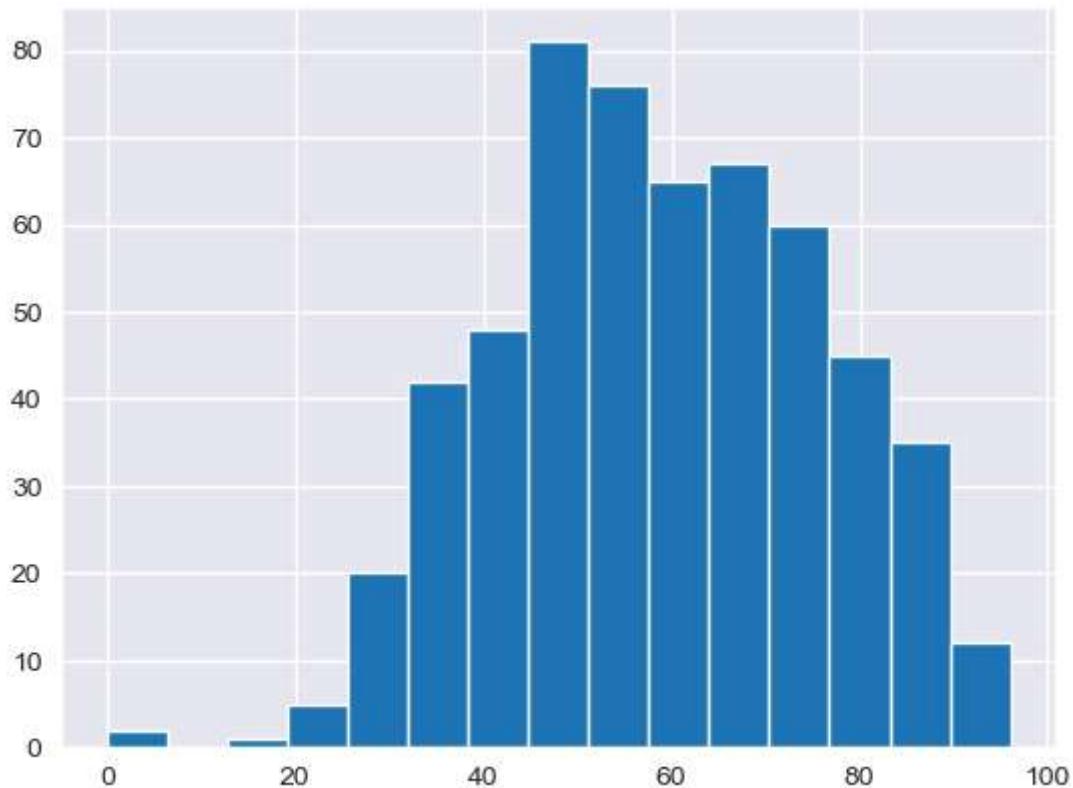
```
In [34]: sns.set_style('darkgrid')
```

```
In [35]: m2 = sns.distplot(movies.AudianceRating, bins = 15)
```



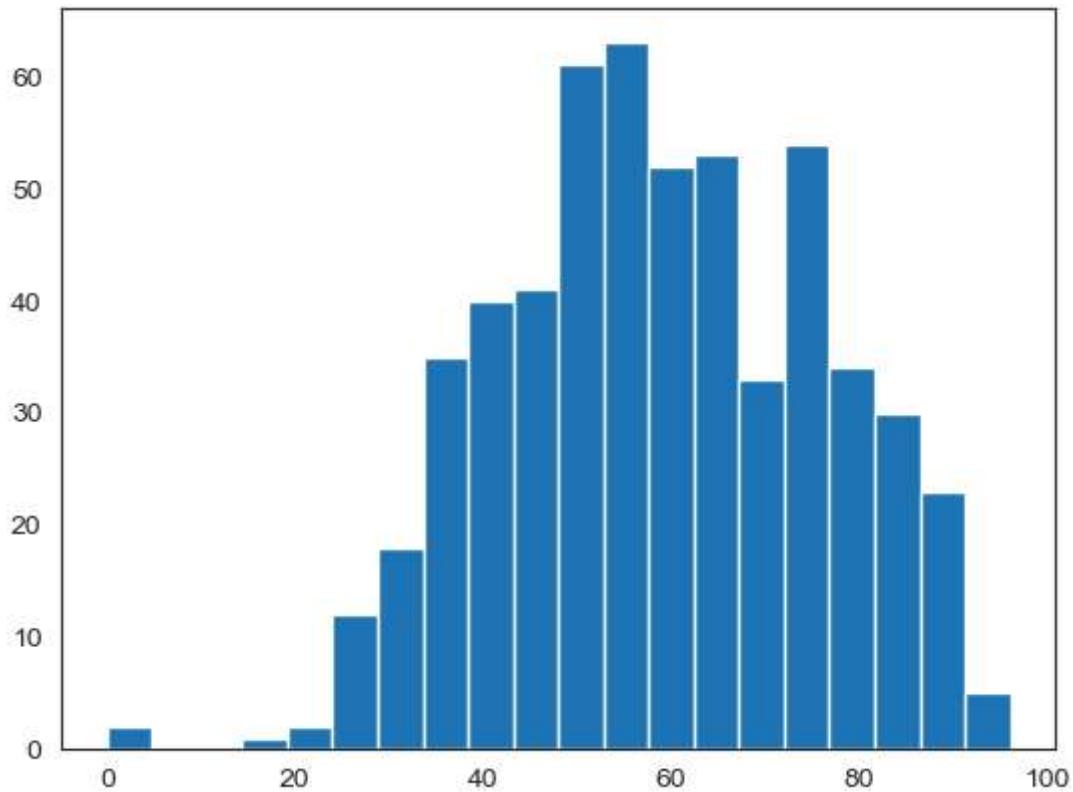
```
In [36]: import seaborn as sns  
import matplotlib.pyplot as plt
```

```
In [37]: #sns.set_style('darkgrid')  
n1 = plt.hist(movies.AudianceRating, bins=15)
```

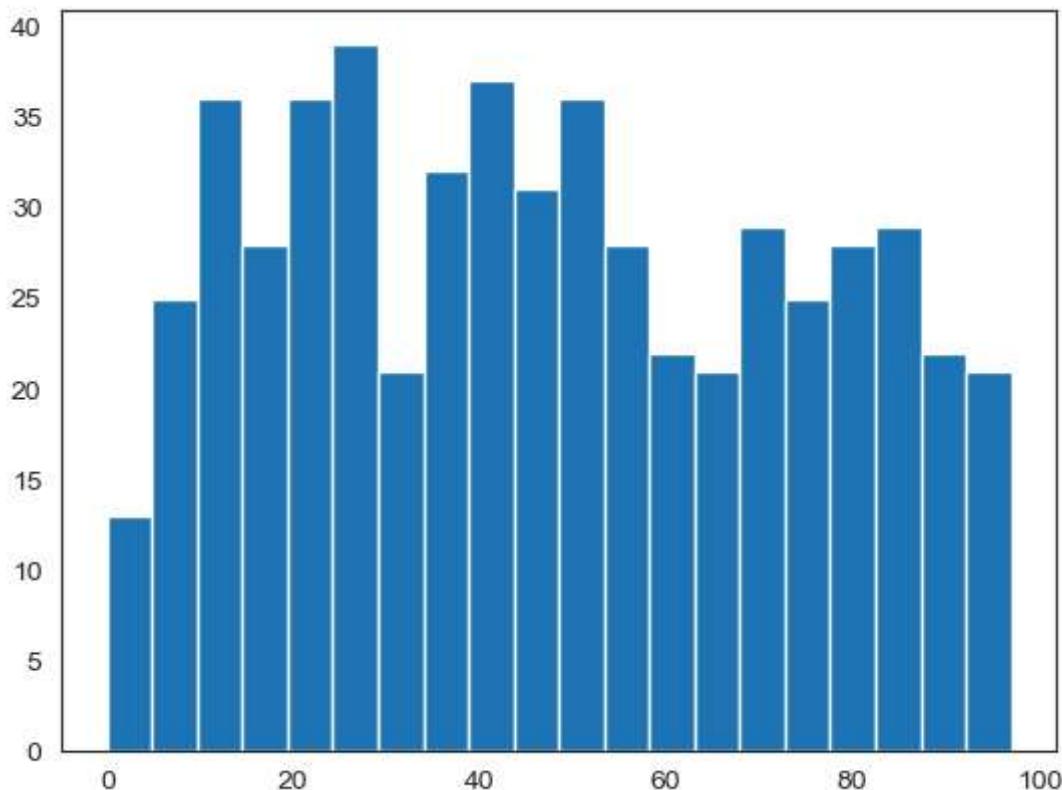


```
In [38]: import seaborn as sns  
import matplotlib.pyplot as plt
```

```
In [39]: sns.set_style('white') #normal distribution & called as bell curve  
n1 = plt.hist(movies.AudianceRating, bins=20)
```

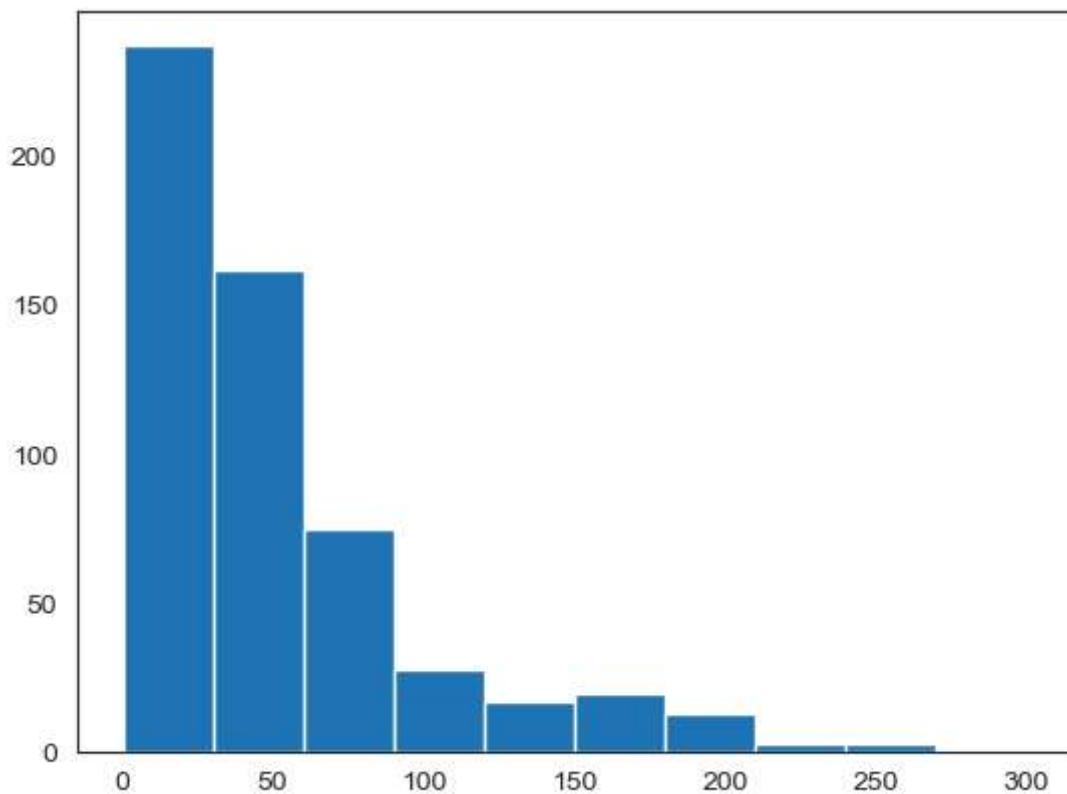


```
In [40]: n1= plt.hist(movies.CriticRating, bins=20)
```

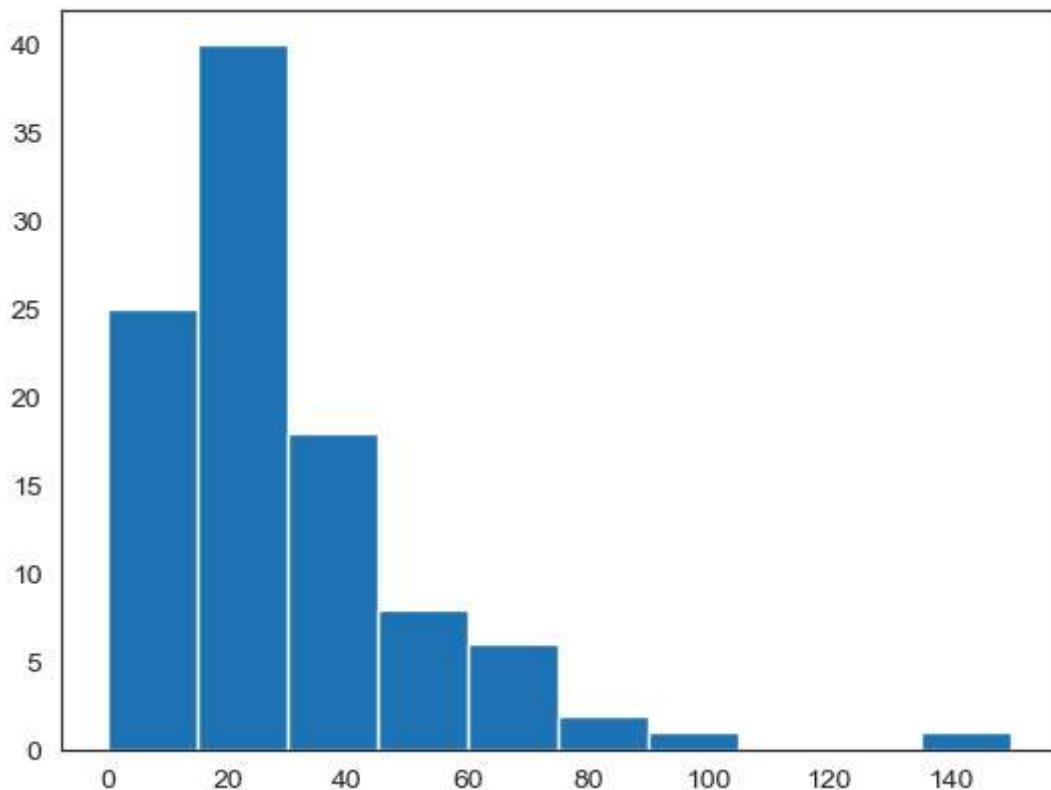


```
In [41]: plt.hist(movies.BudgetMillions)  
plt.show
```

```
Out[41]: <function matplotlib.pyplot.show(close=None, block=None)>
```



```
In [42]: plt.hist(movies[movies.Genre=='Drama'].BudgetMillions)
plt.show()
```



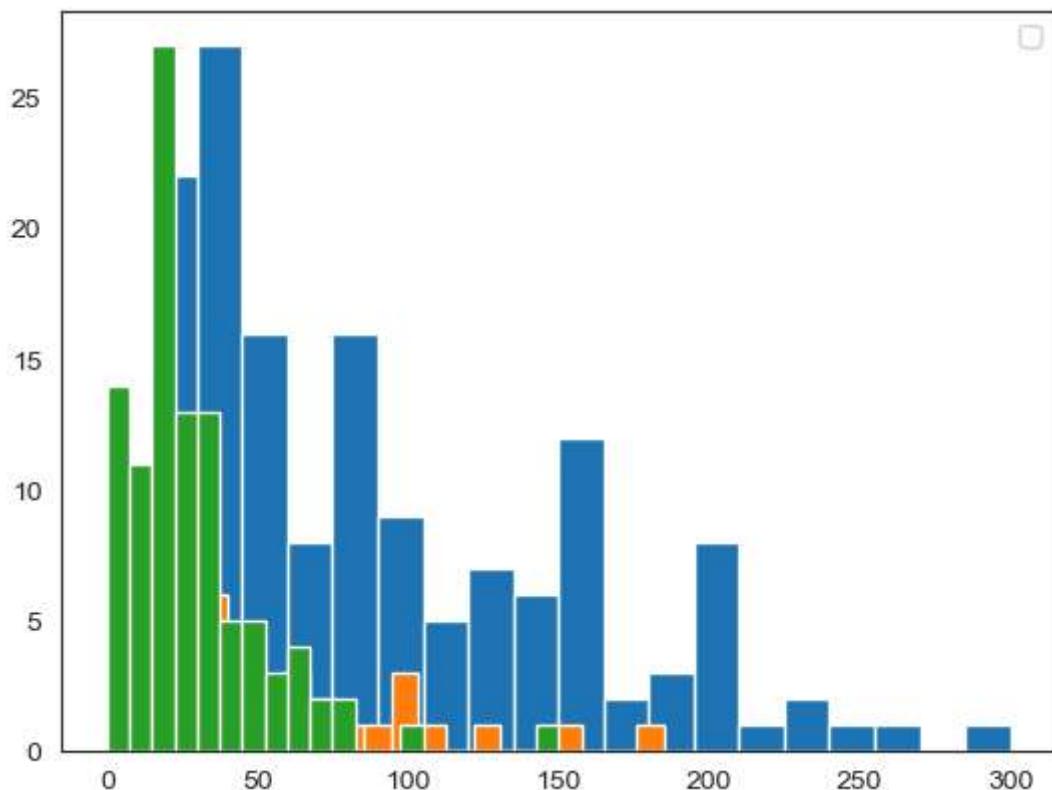
```
In [43]: movies.head()
```

Out[43]:

| | Film | Genre | CriticRating | AudianceRating | BudgetMillions | Years |
|---|----------------------|-----------|--------------|----------------|----------------|-------|
| 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 |

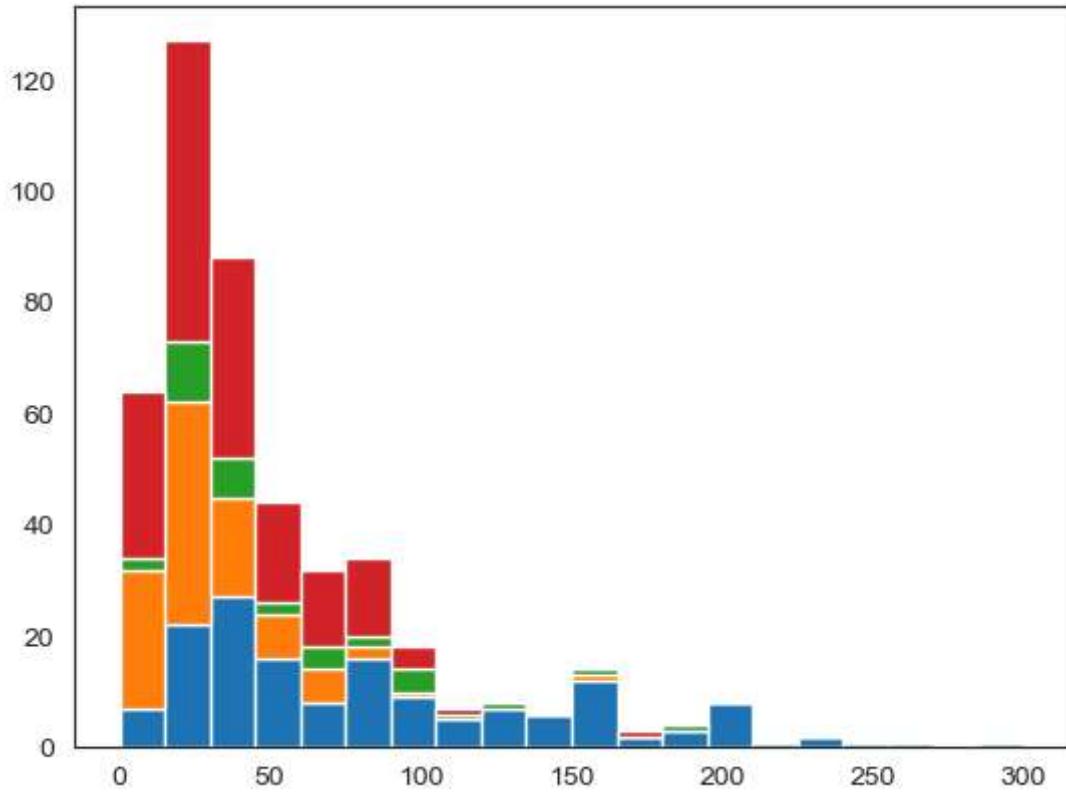
In [44]:

```
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 [45]:

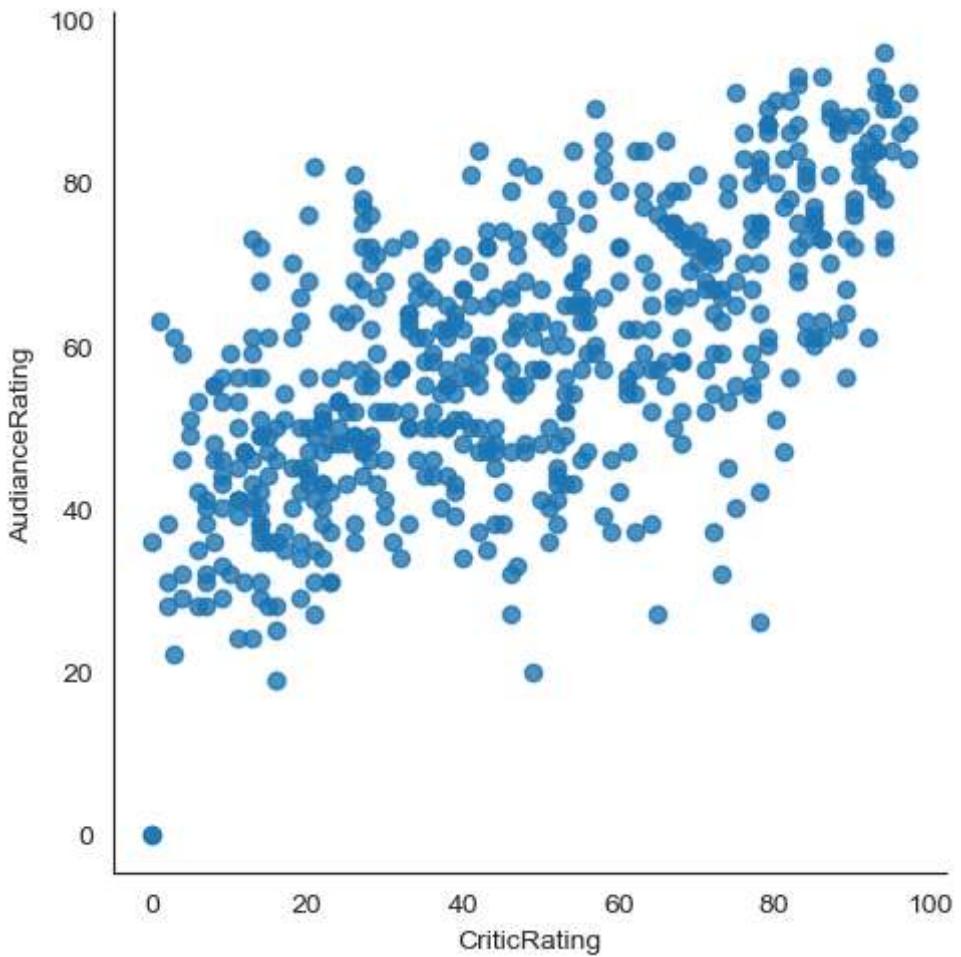
```
plt.hist([movies[movies.Genre=='Action'].BudgetMillions,\n         movies[movies.Genre=='Drama'].BudgetMillions,\n         movies[movies.Genre=='Thriller'].BudgetMillions,\n         movies[movies.Genre=='Comedy'].BudgetMillions],bins=20,stacked= True)\nplt.show()
```



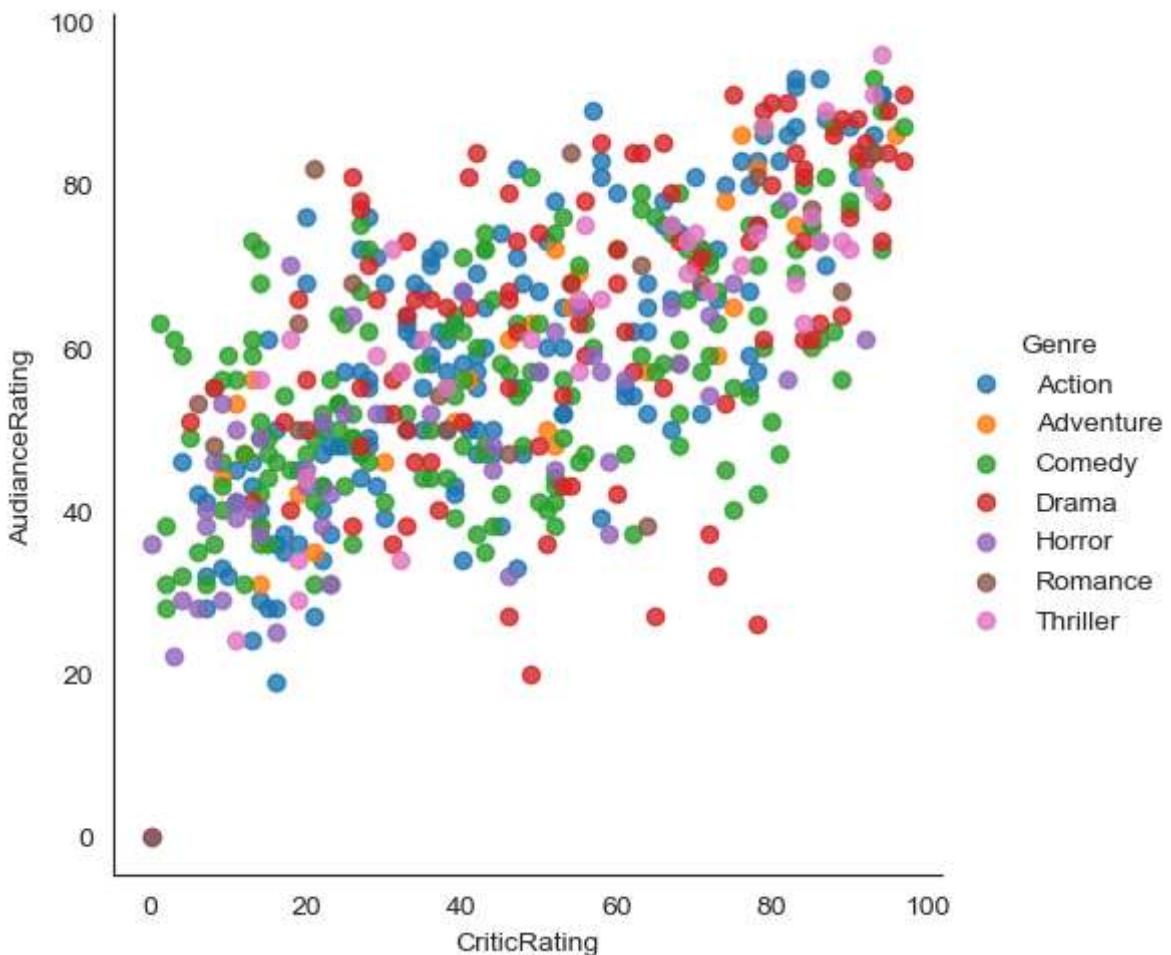
```
In [46]: for gen in movies.Genre.cat.categories:  
    print(gen)
```

Action
Adventure
Comedy
Drama
Horror
Romance
Thriller

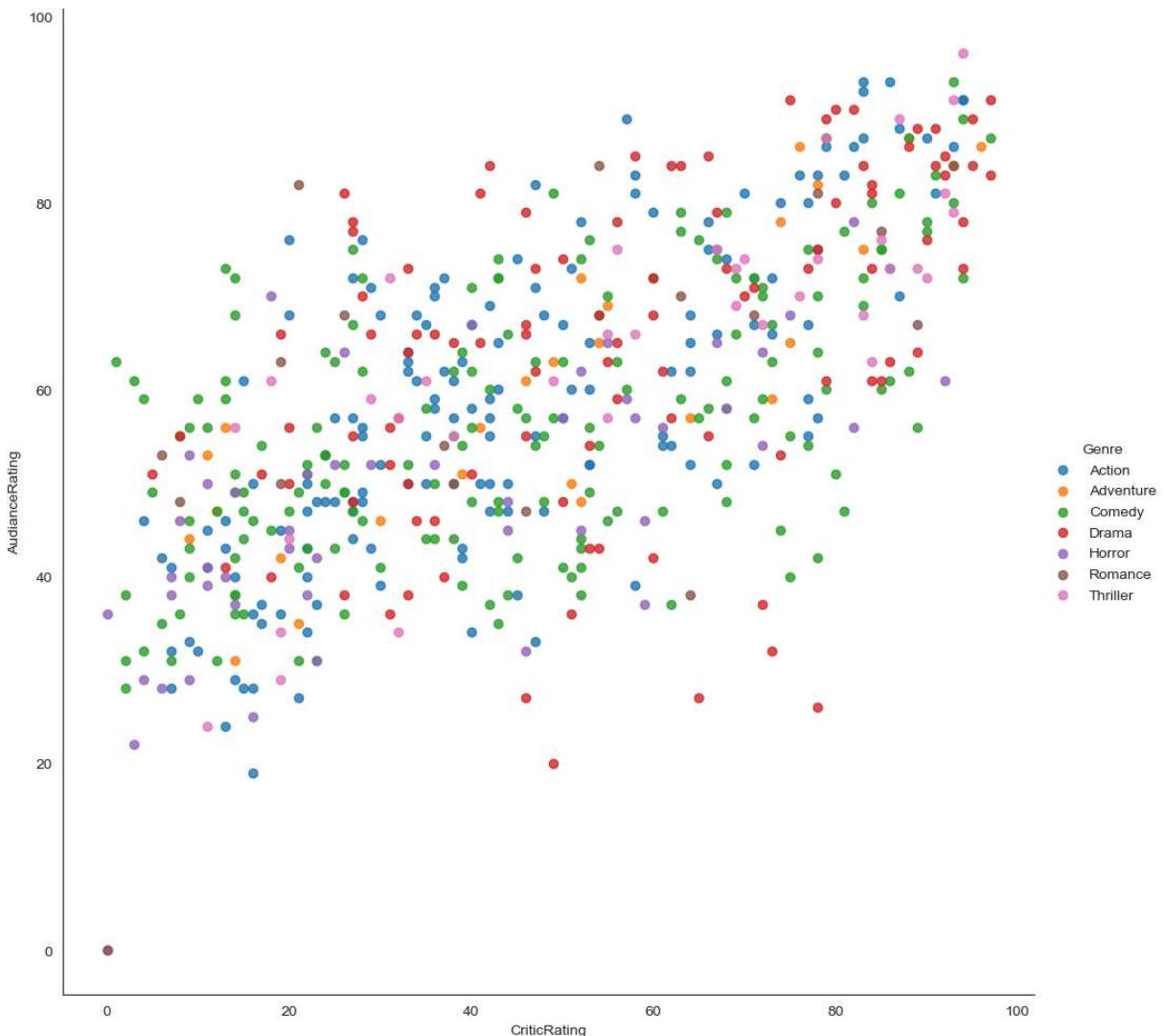
```
In [47]: vis1= sns.lmplot(data=movies, x='CriticRating',y='AudianceRating',  
                      fit_reg= False)
```



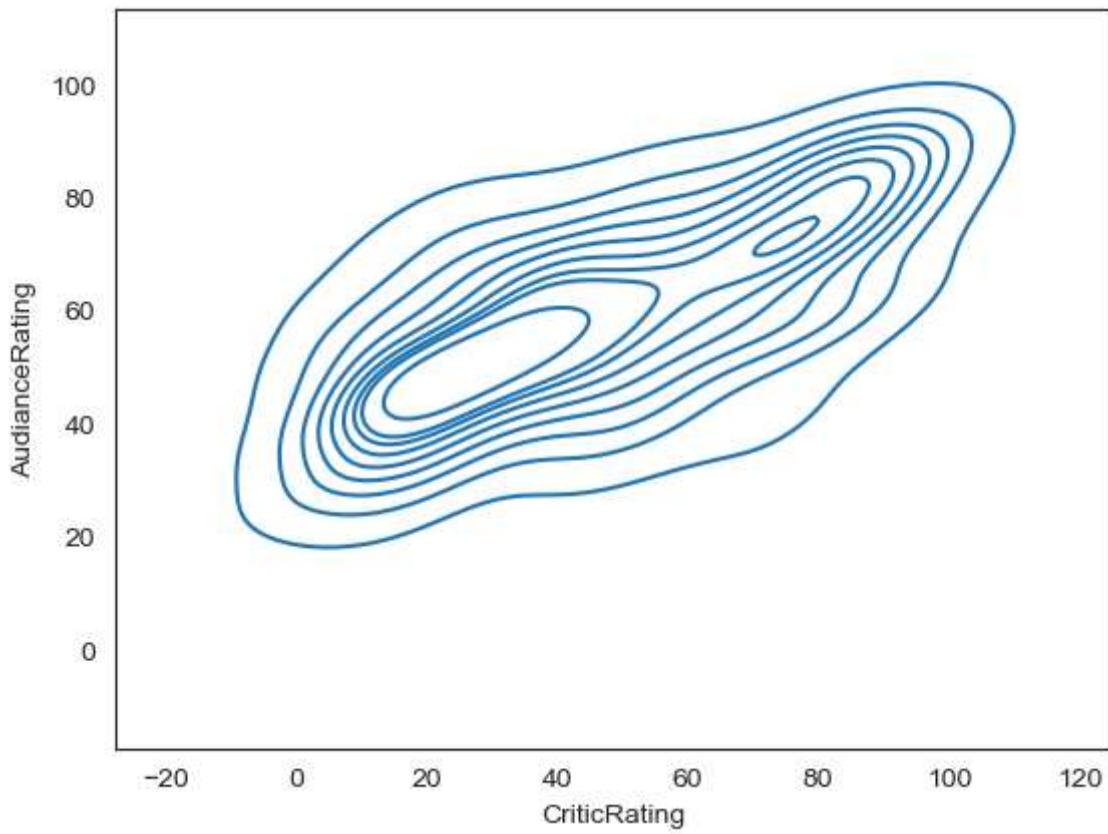
```
In [48]: vis1= sns.lmplot(data=movies, x='CriticRating',y='AudianceRating',\n                     fit_reg= False, hue= 'Genre')
```



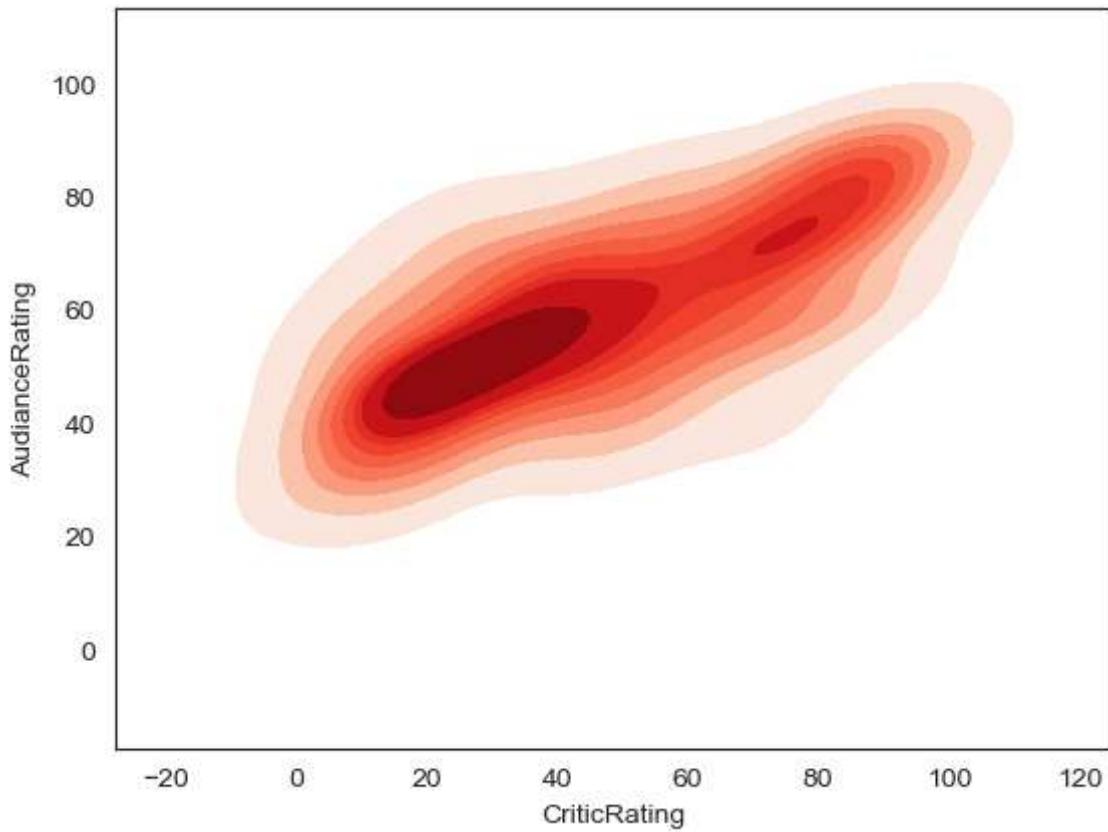
```
In [49]: vis1= sns.lmplot(data=movies, x='CriticRating',y='AudianceRating',\n                     fit_reg= False, hue= 'Genre',height=10, aspect=1)
```



```
In [50]: k1 = sns.kdeplot(x=movies.CriticRating,y=movies.AudianceRating)
```

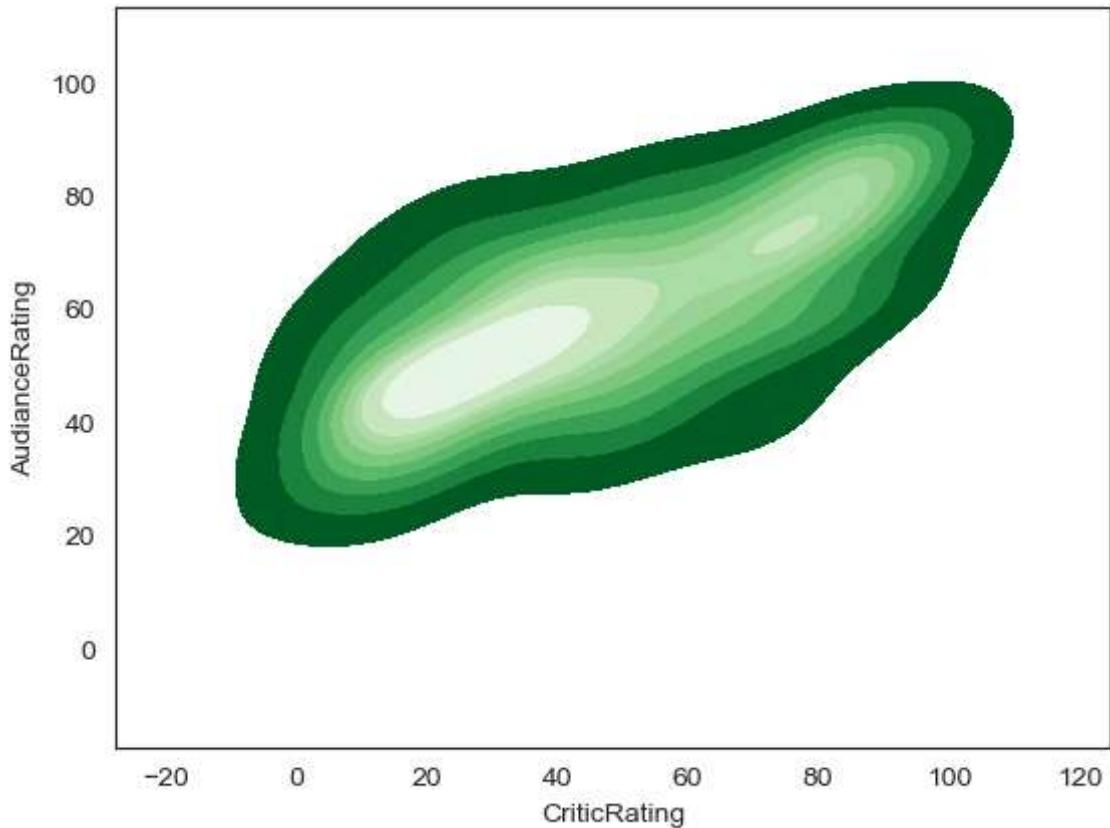


```
In [51]: k1 = sns.kdeplot(x=movies.CriticRating,y=movies.AudianceRating,shade= True, shade_lowest=False)
```



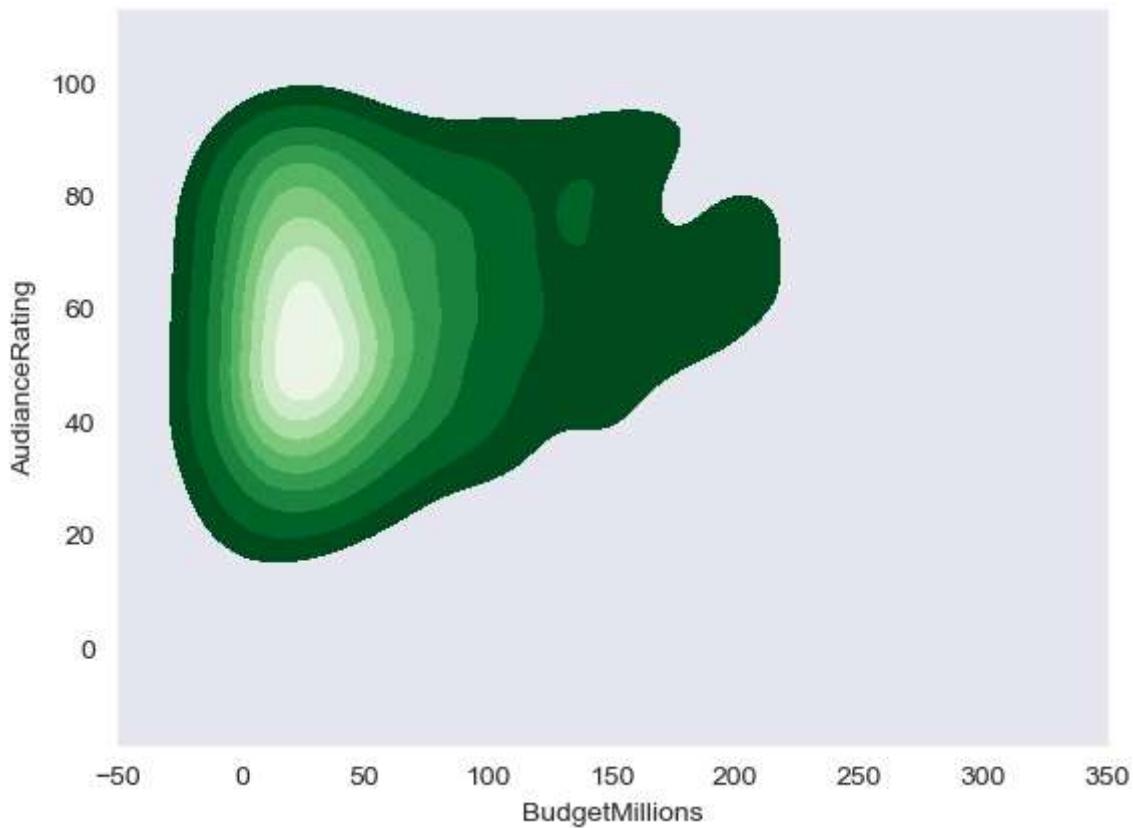
```
In [52]: sns.kdeplot(x=movies.CriticRating,y=movies.AudianceRating,shade= True, shade_lowest=False)
```

```
Out[52]: <Axes: xlabel='CriticRating', ylabel='AudianceRating'>
```

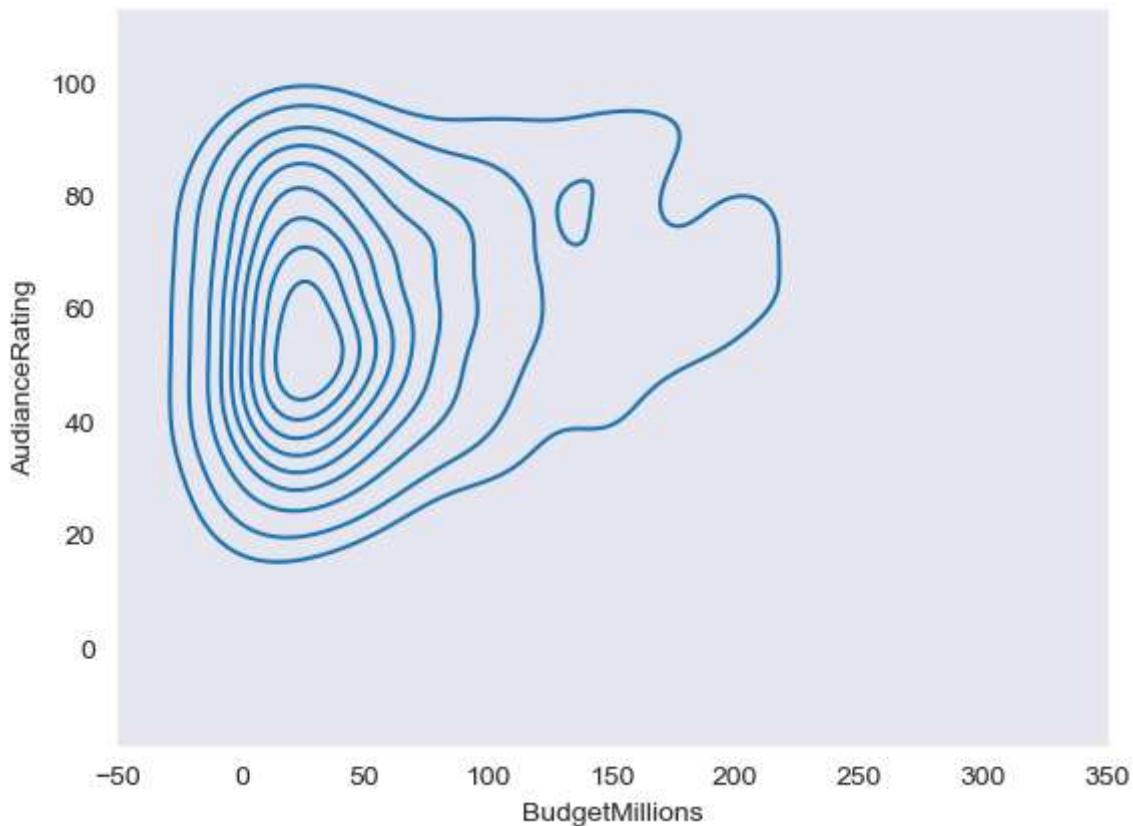


```
In [53]: sns.set_style('dark')
sns.kdeplot(x=movies.BudgetMillions,y=movies.AudianceRating,shade= True,shade_lowes
```

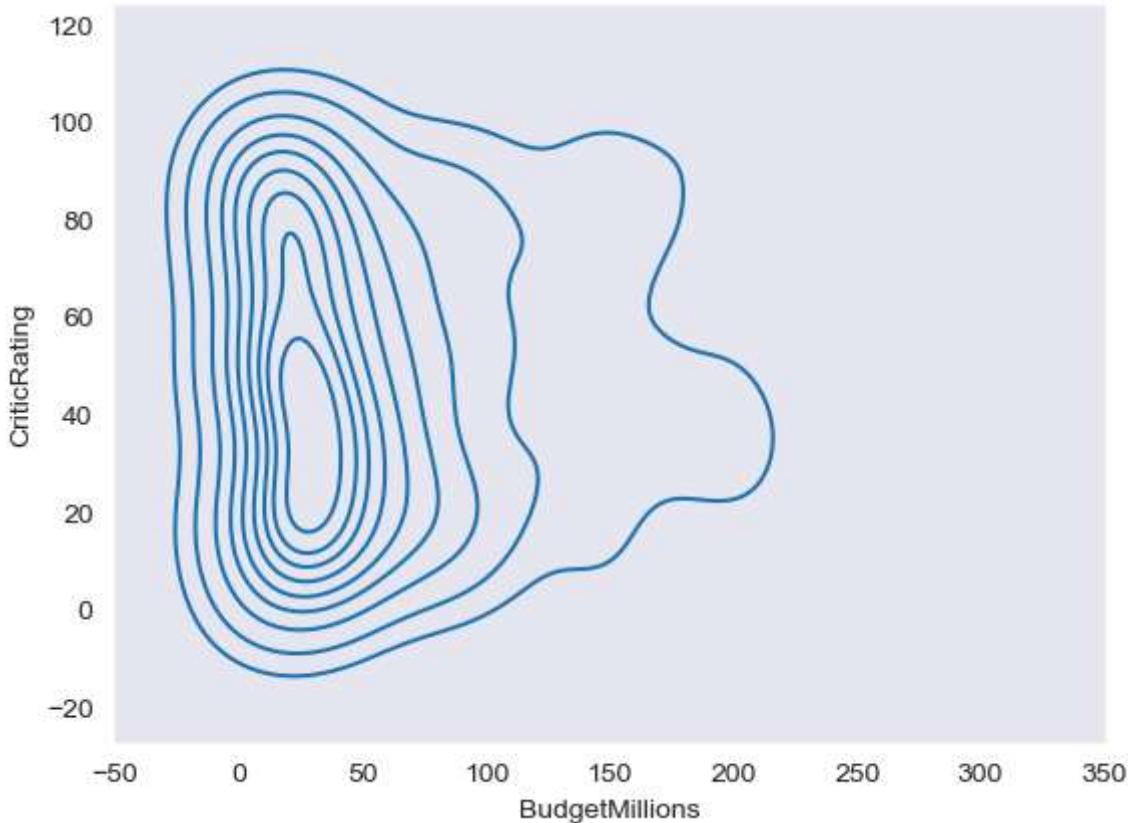
```
Out[53]: <Axes: xlabel='BudgetMillions', ylabel='AudianceRating'>
```



```
In [54]: sns.set_style('dark')
k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudianceRating)
```

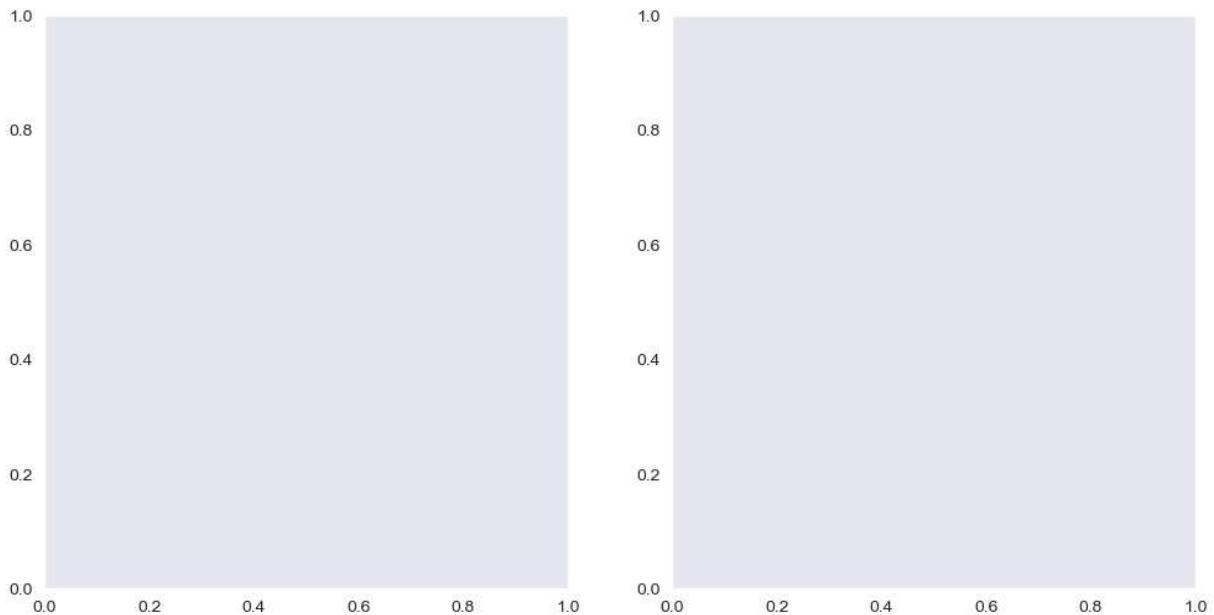


```
In [55]: k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating)
```



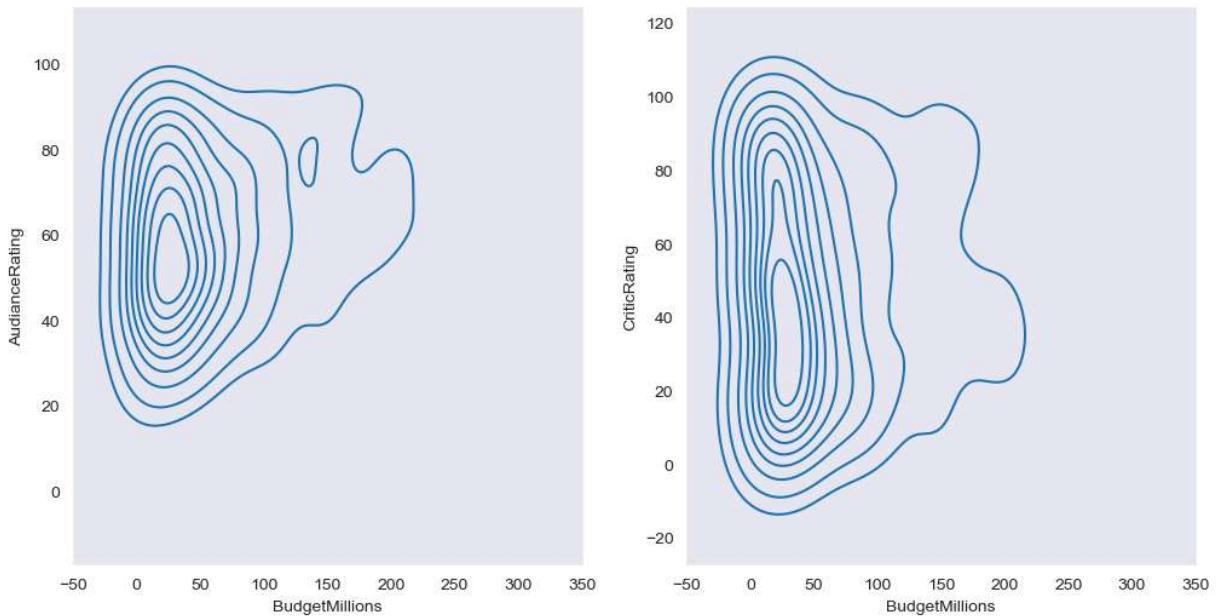
```
In [56]: #subplots
```

```
f, ax = plt.subplots(1,2, figsize=(12,6))  
#f, ax = plt.subplots(3,3, figsize=(12,6))
```



```
In [57]: f, axes = plt.subplots(1,2, figsize=(12,6))
```

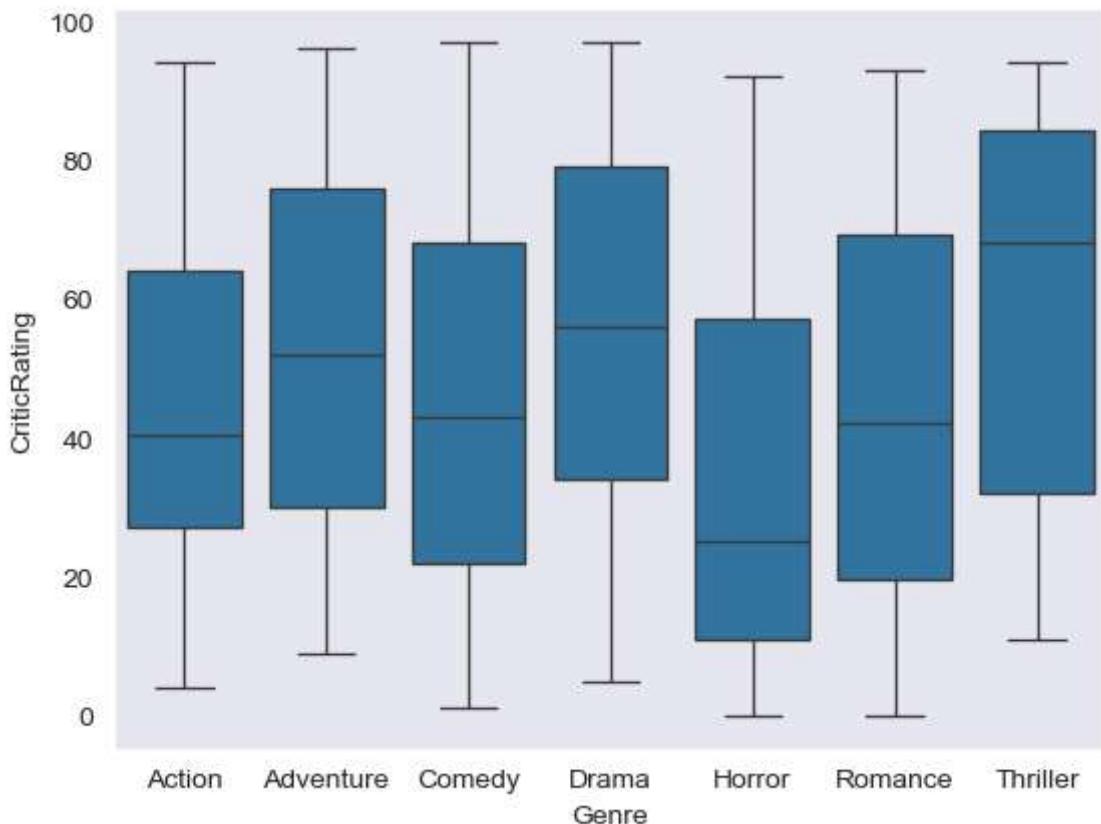
```
k1 = sns.kdeplot(x=movies.BudgetMillions,y=movies.AudianceRating,ax=axes[0])
k2 = sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating,ax = axes[1])
```



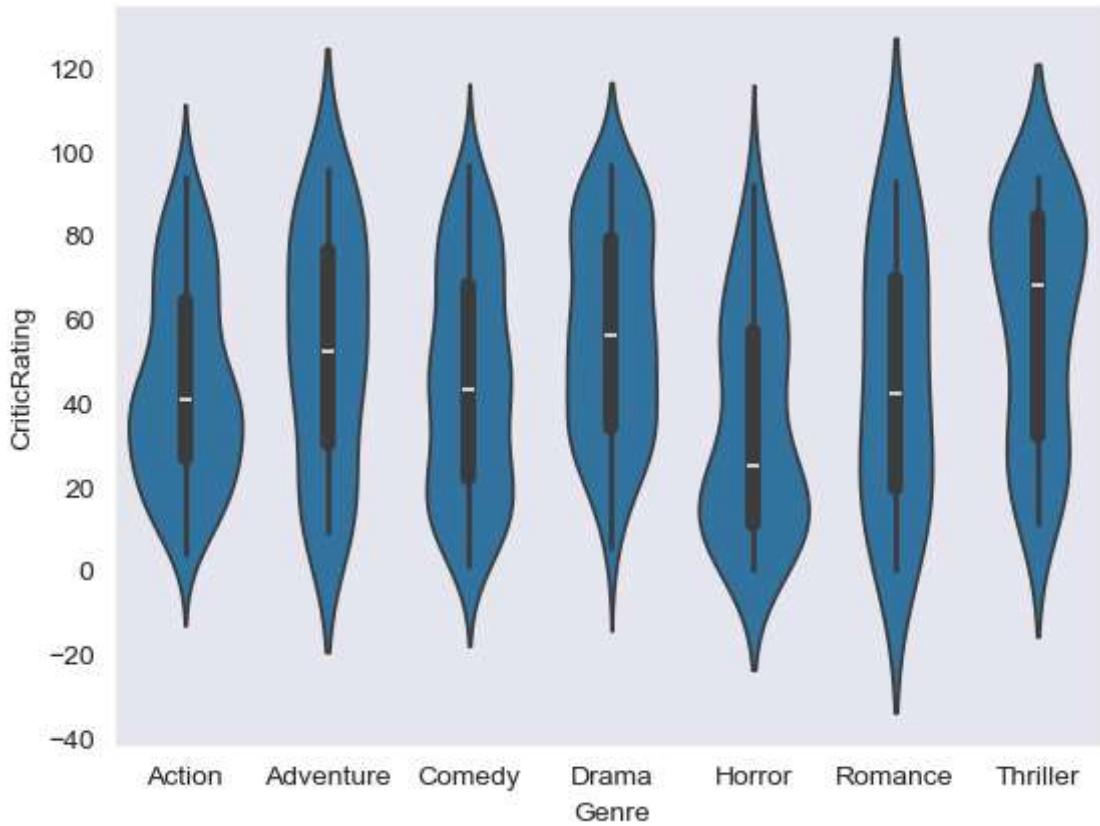
In [58]: `axes`

Out[58]: `array([<Axes: xlabel='BudgetMillions', ylabel='AudianceRating'>,
 <Axes: xlabel='BudgetMillions', ylabel='CriticRating'>],
 dtype=object)`

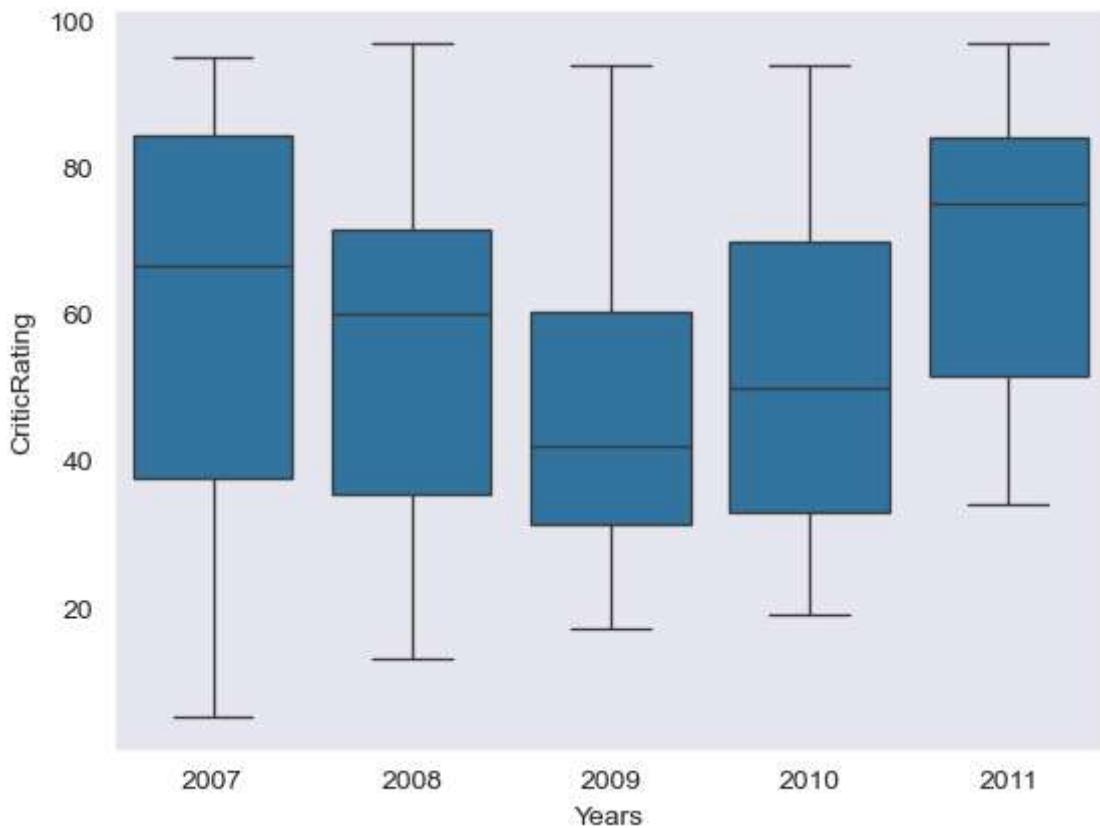
In [59]: `w= sns.boxplot(data=movies, x='Genre', y= 'CriticRating')`



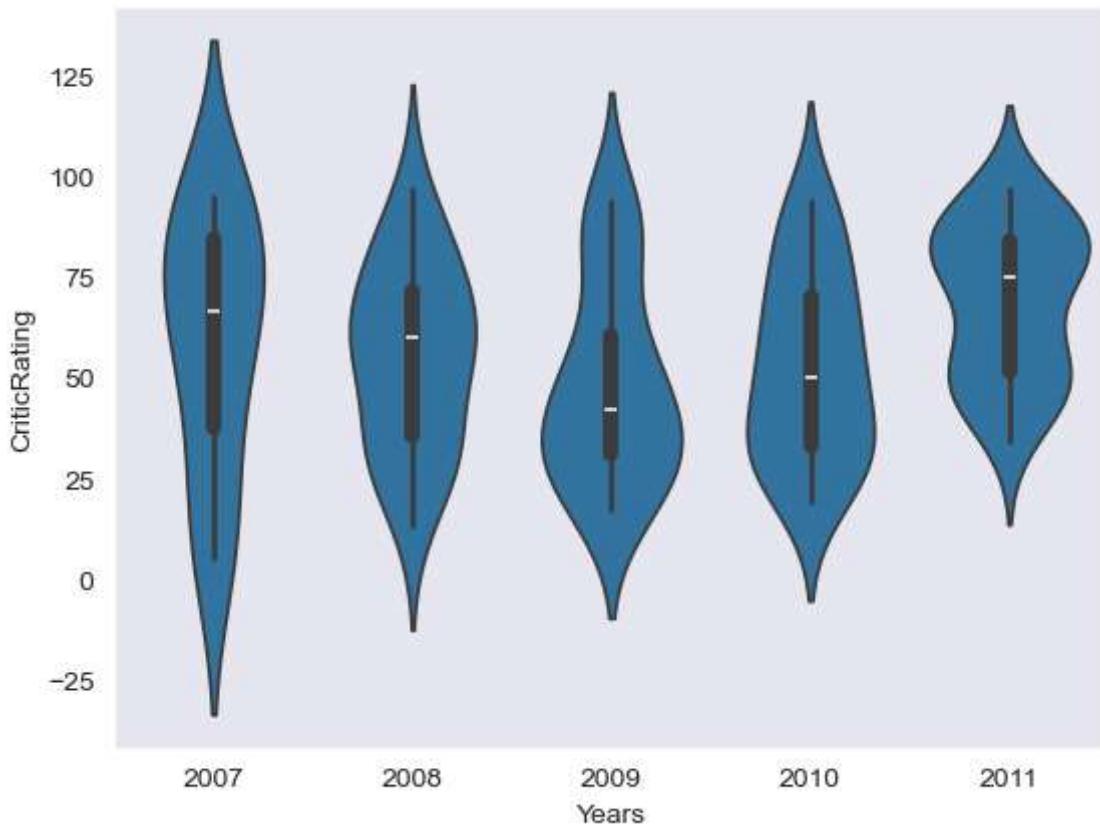
```
In [60]: z=sns.violinplot(data=movies, x='Genre', y= 'CriticRating')
```



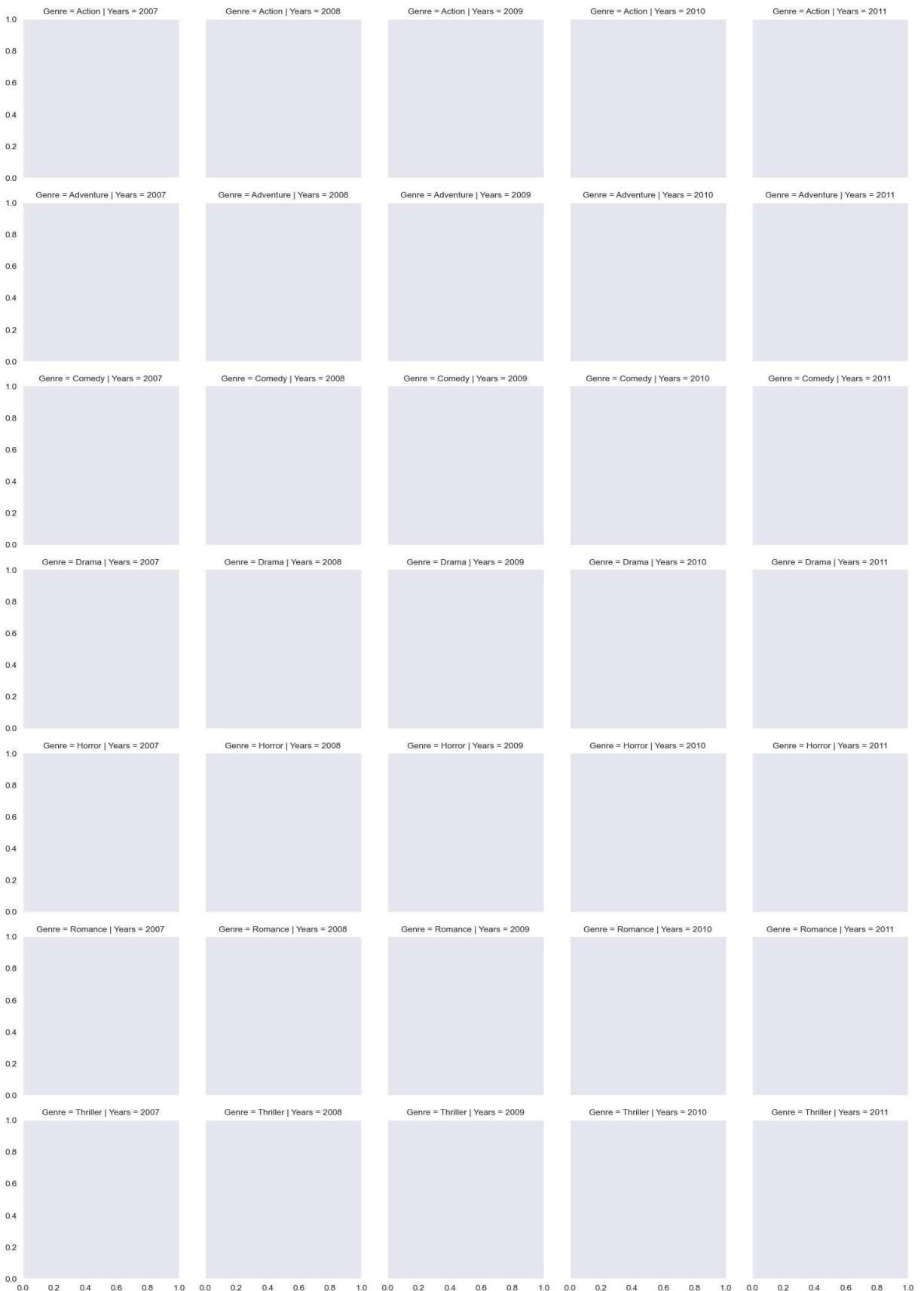
```
In [61]: w1=sns.boxplot(data=movies[movies.Genre == 'Drama'], x= 'Years', y= 'CriticRating')
```



```
In [62]: z = sns.violinplot(data=movies[movies.Genre == 'Drama'], x='Years', y = 'CriticRating')
```

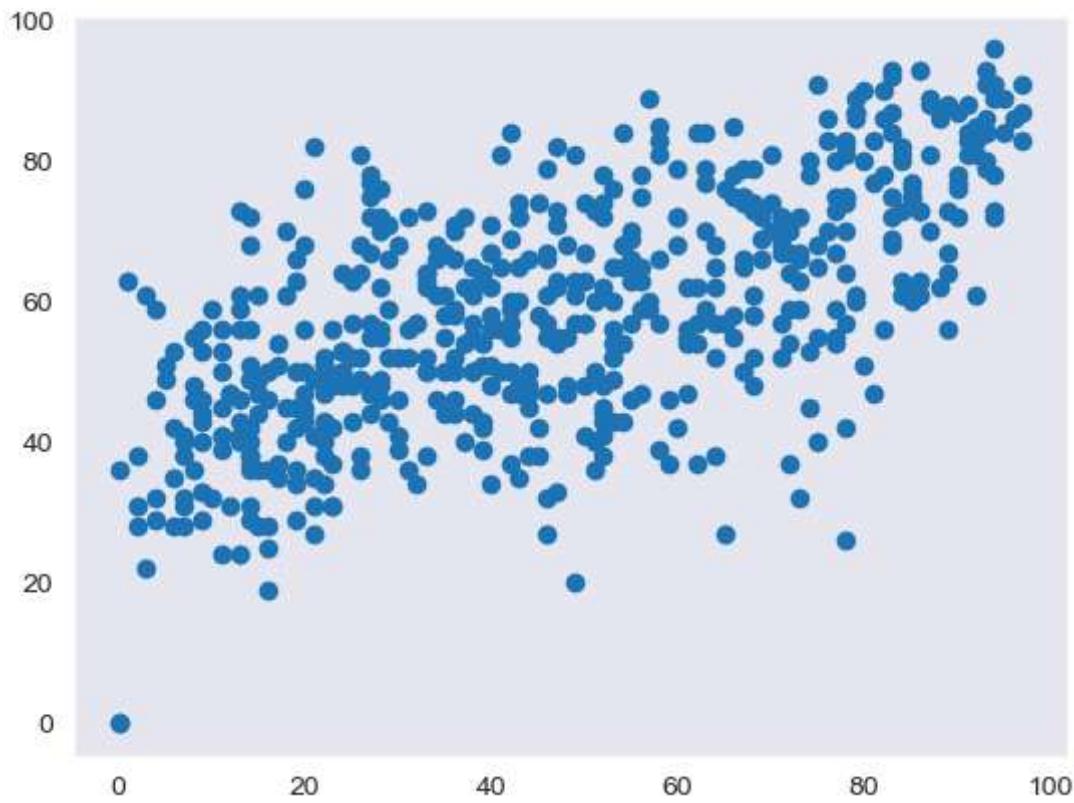


```
In [63]: g=sns.FacetGrid (movies, row = 'Genre', col = 'Years', hue = 'Genre') #kind of sub
```

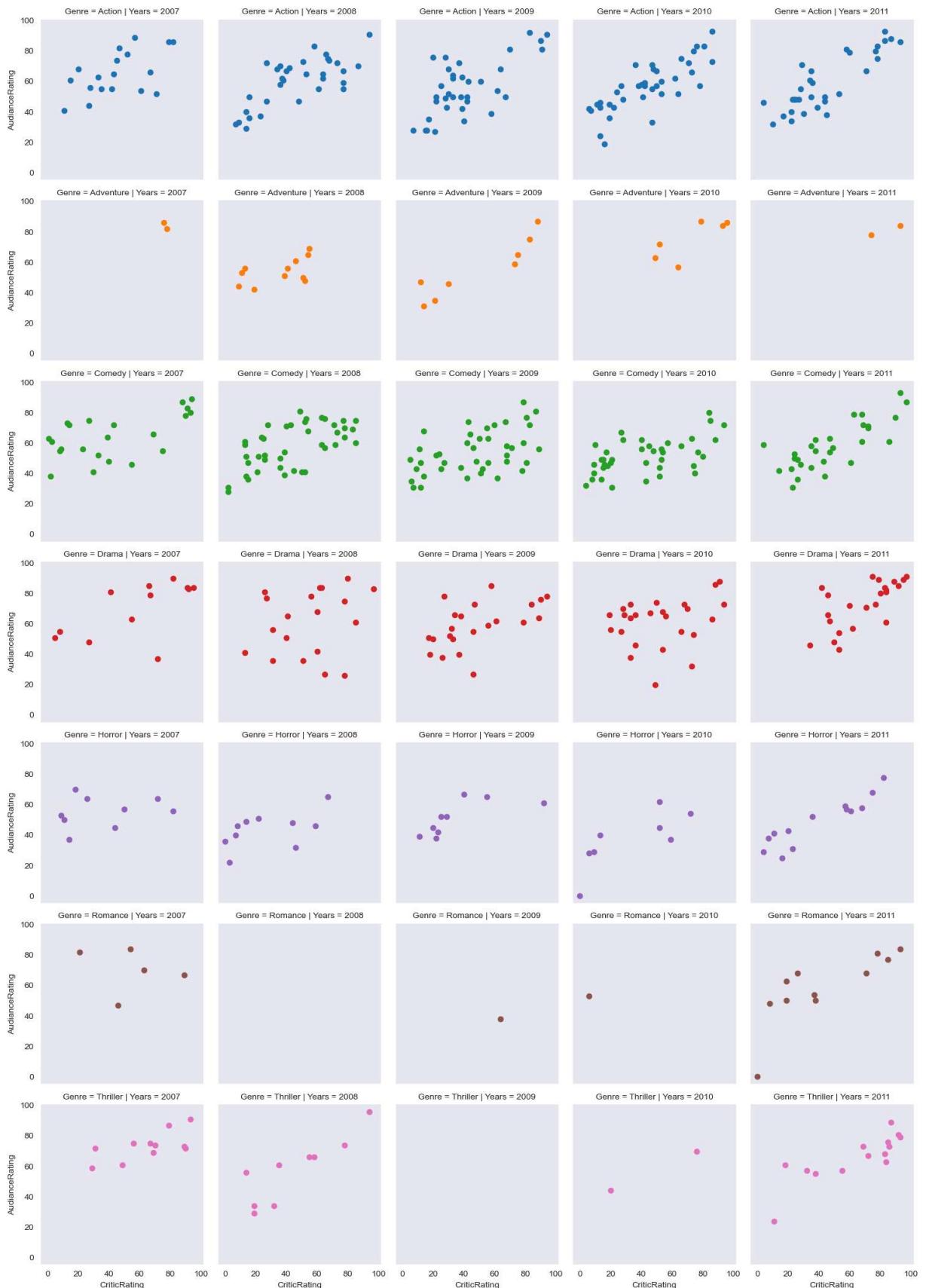


```
In [64]: plt.scatter(movies.CriticRating,movies.AudianceRating)
```

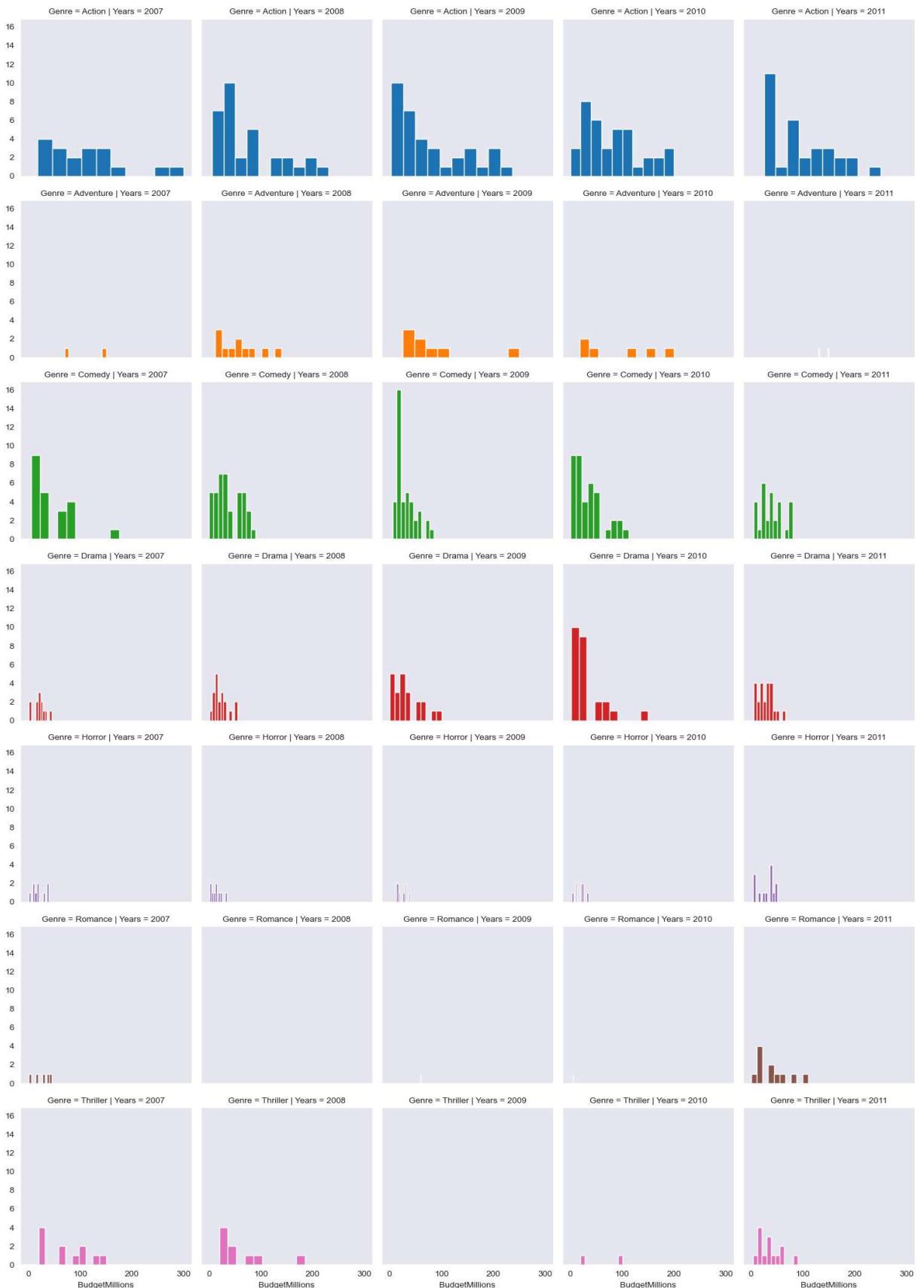
```
Out[64]: <matplotlib.collections.PathCollection at 0x1f73dc82c10>
```



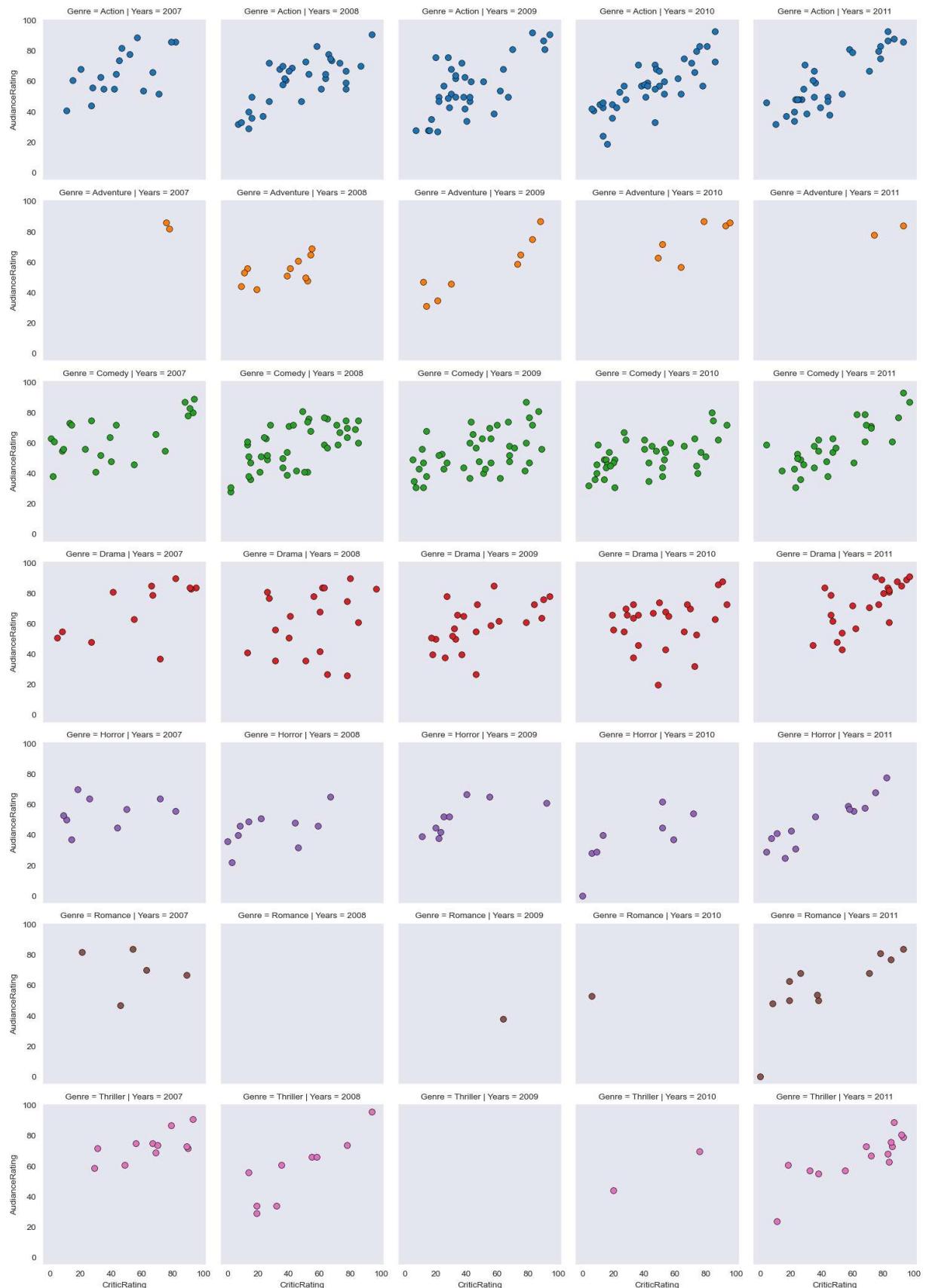
```
In [65]: g= sns.FacetGrid(movies, row= 'Genre', col= 'Years',hue='Genre')
g= g.map(plt.scatter, 'CriticRating', 'AudianceRating')
```



```
In [66]: g = sns.FacetGrid(movies, row= 'Genre', col= 'Years',hue='Genre')
g = g.map(plt.hist, 'BudgetMillions')
```



```
In [67]: g = sns.FacetGrid(movies, row= 'Genre', col= 'Years',hue='Genre')
kws= dict(s=50, linewidth= 0.5, edgecolor= 'black')
g= g.map(plt.scatter, 'CriticRating', 'AudianceRating', **kws)
```



```
In [68]: # python is not vectorize programming language
# Building dashboards (dashboard - combination of chats)

sns.set_style('darkgrid')
```

```

f, axes = plt.subplots (2,2, figsize = (15,15))

k1 = sns.kdeplot(x=movies.BudgetMillions, y=movies.AudianceRating, ax = axes[0,0])
k2 = sns.kdeplot(x=movies.BudgetMillions, y=movies.CriticRating, ax = axes[0,1])

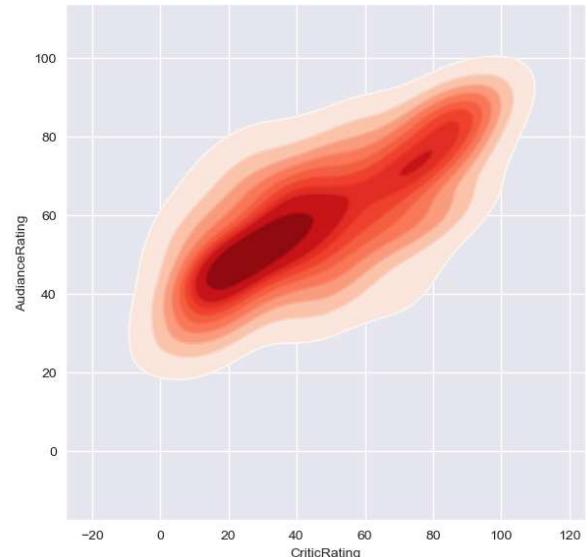
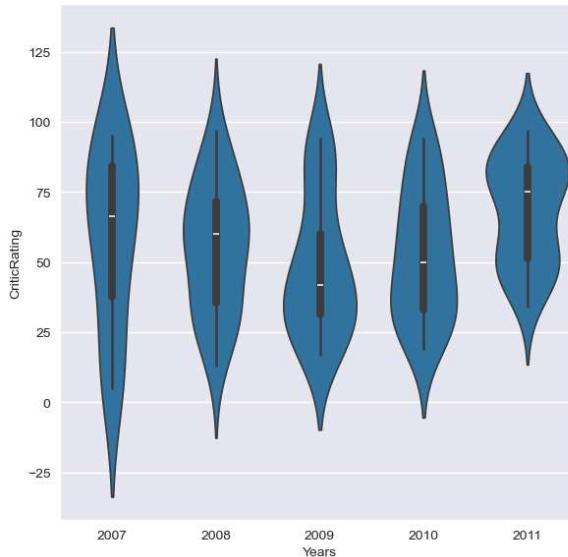
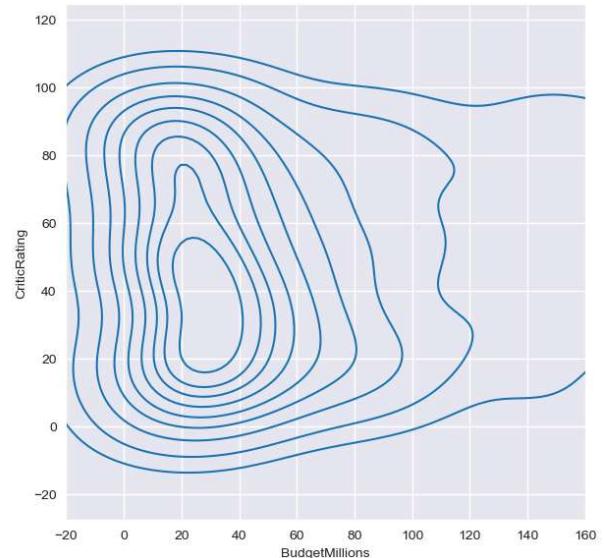
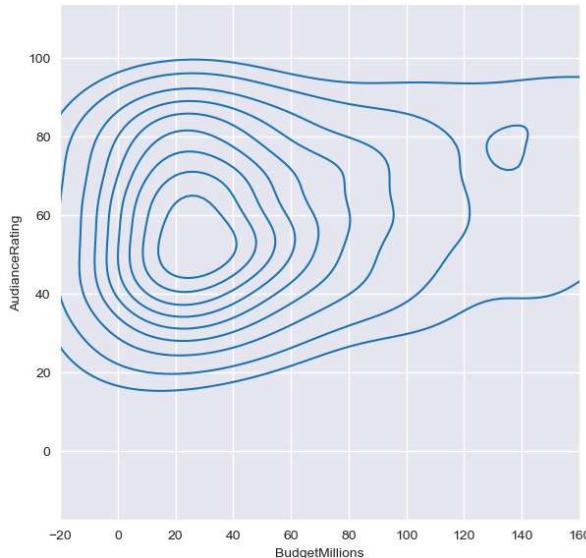
k1.set(xlim=(-20,160))
k2.set(xlim=(-20,160))

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

k4 = sns.kdeplot(x=movies.CriticRating, y=movies.AudianceRating, shade = True, shade_lowest=False)
k4b = sns.kdeplot(x=movies.CriticRating, y= movies.AudianceRating,cmap='Reds',ax = axes[1,1])

plt.show()

```



In []: `sns.set_style('dark', {'axes.facecolor':`