In [1]:

```python
import pandas as pd
```

In [2]:

```python
movies = pd.read_csv('/Users/chandnisingh/Downloads/Movie-Rating.csv')
```

In [3]:

```python
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]:

```python
len(movies)
```

Out[4]:

```
559
```

In [5]:

```python
type(movies)
```

Out[5]:

```
pandas.core.frame.DataFrame
```

In [6]:

```python
import numpy
print(numpy.__version__)
```

```
1.26.4
```

In [7]:

```python
import pandas
print(pandas.__version__)
```

```
2.2.2
```

```
In [8]: movies.columns
```

```
Out[8]: Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings
        %',
               'Budget (million $)', 'Year of release'],
              dtype='object')
```

```
In [9]: 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 [10]: movies.shape
```

```
Out[10]: (559, 6)
```

```
In [11]: movies.tail()
```

Out[11]:

| | 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 [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.columns = ['Film', 'Genre', 'CriticRating', 'AudienceRatings',
       'BudgetMillions', 'Year']
```

In [14]:
```
movies.head()
```

Out[14]:

| | Film | Genre | CriticRating | AudienceRatings | 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 [15]:
```
movies.head(1)
```

Out[15]:

| | Film | Genre | CriticRating | AudienceRatings | BudgetMillions | Year |
|---|---|---|---|---|---|---|
| **0** | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |

In [16]:
```
movies.describe()
```

Out[16]:

| | CriticRating | AudienceRatings | 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 [17]:
```
movies.Film = movies.Film.astype('category')
```

In [18]:
```
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    object  
 2   CriticRating     559 non-null    int64   
 3   AudienceRatings  559 non-null    int64   
 4   BudgetMillions   559 non-null    int64   
 5   Year             559 non-null    int64   
dtypes: category(1), int64(4), object(1)
memory usage: 43.6+ KB
```

In [19]:
```
movies.Genre = movies.Genre.astype('category')
movies.Year = movies.Year.astype('category')
```

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   AudienceRatings  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]:

```
movies.describe()
```

Out[21]:

| | CriticRating | AudienceRatings | 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 [22]:

```
from matplotlib import pyplot as plt
import seaborn as sns
```

In [23]:

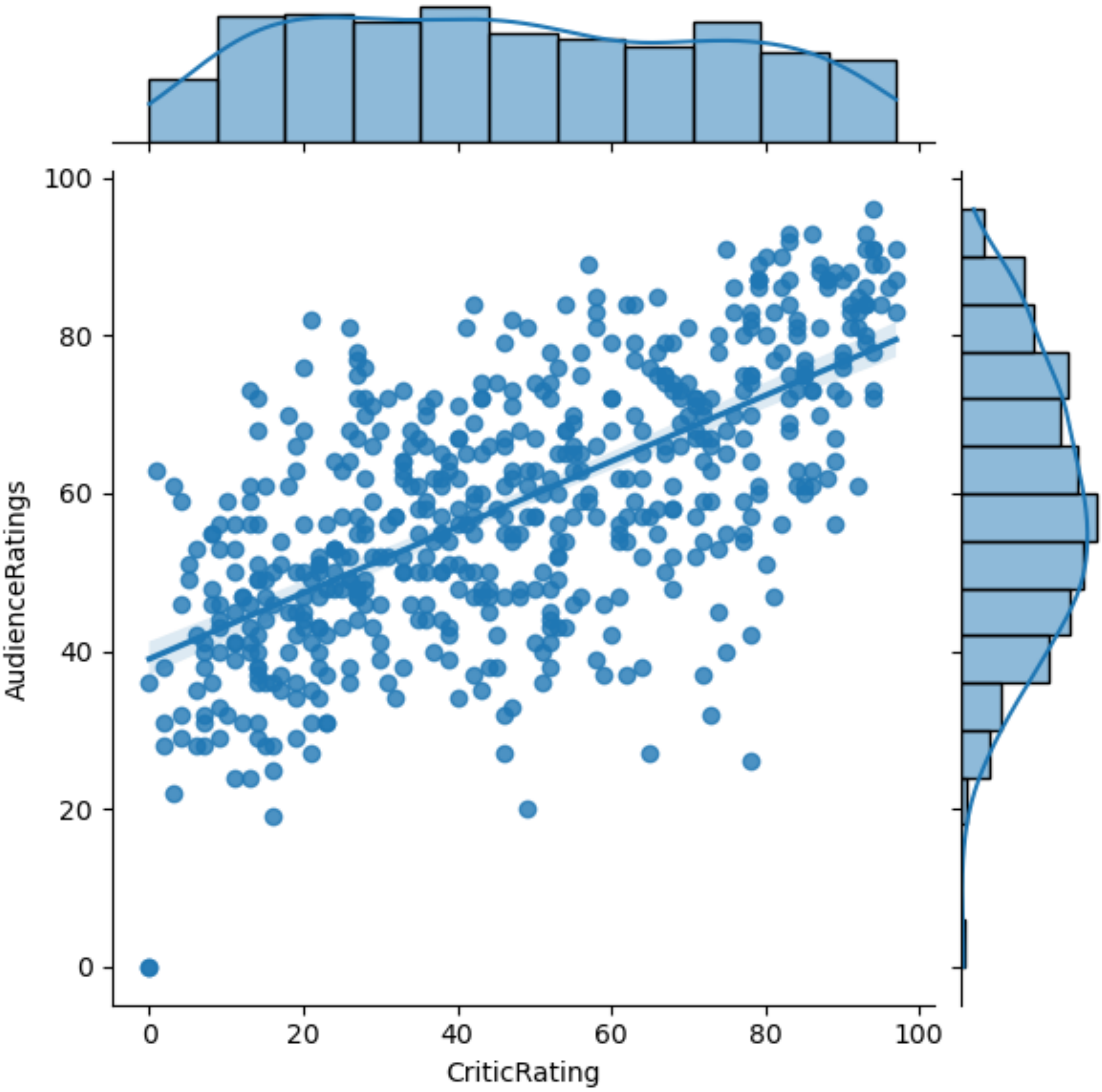
```
%matplotlib inline
```

In [24]:

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

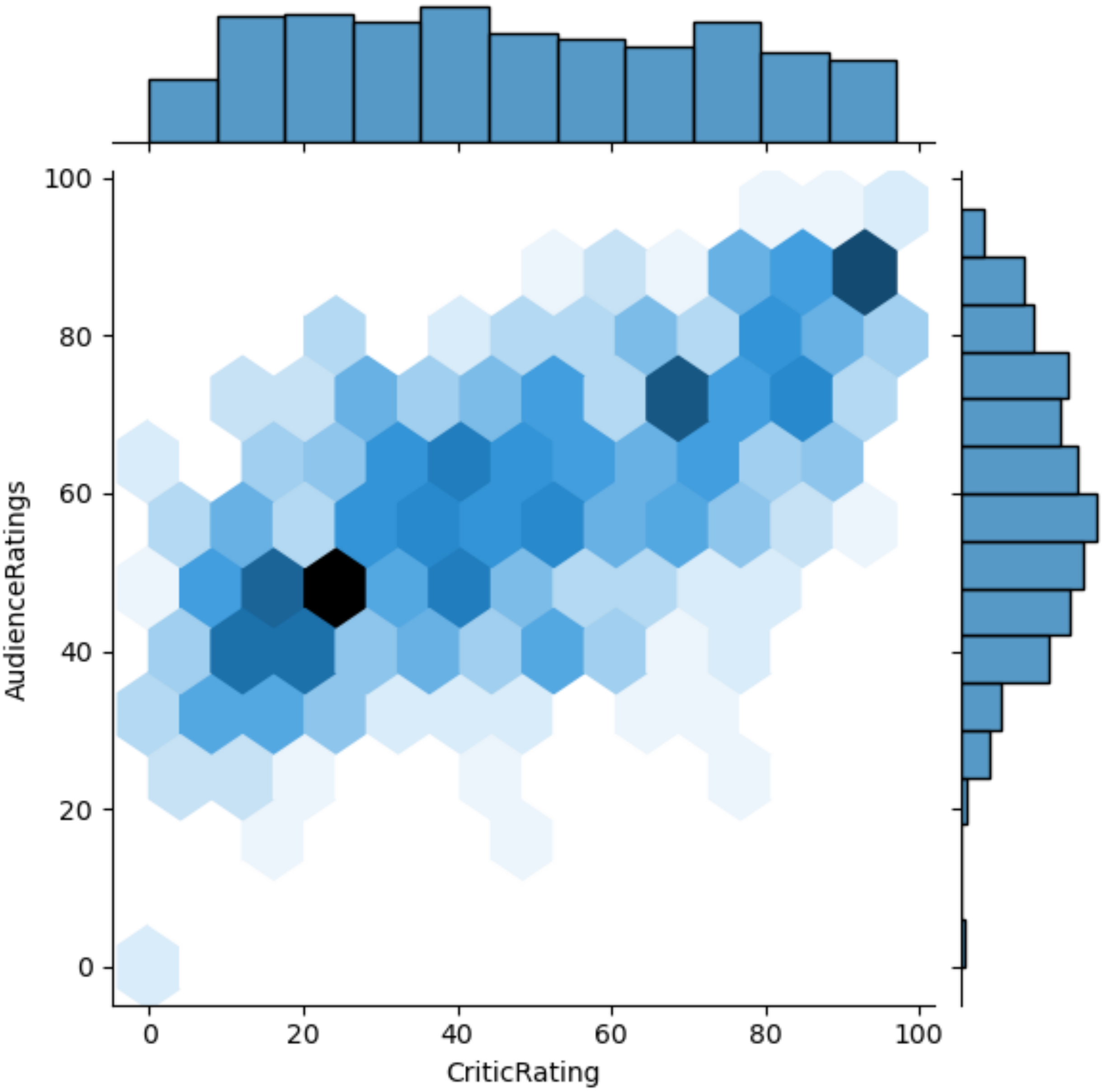
In [25]:

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

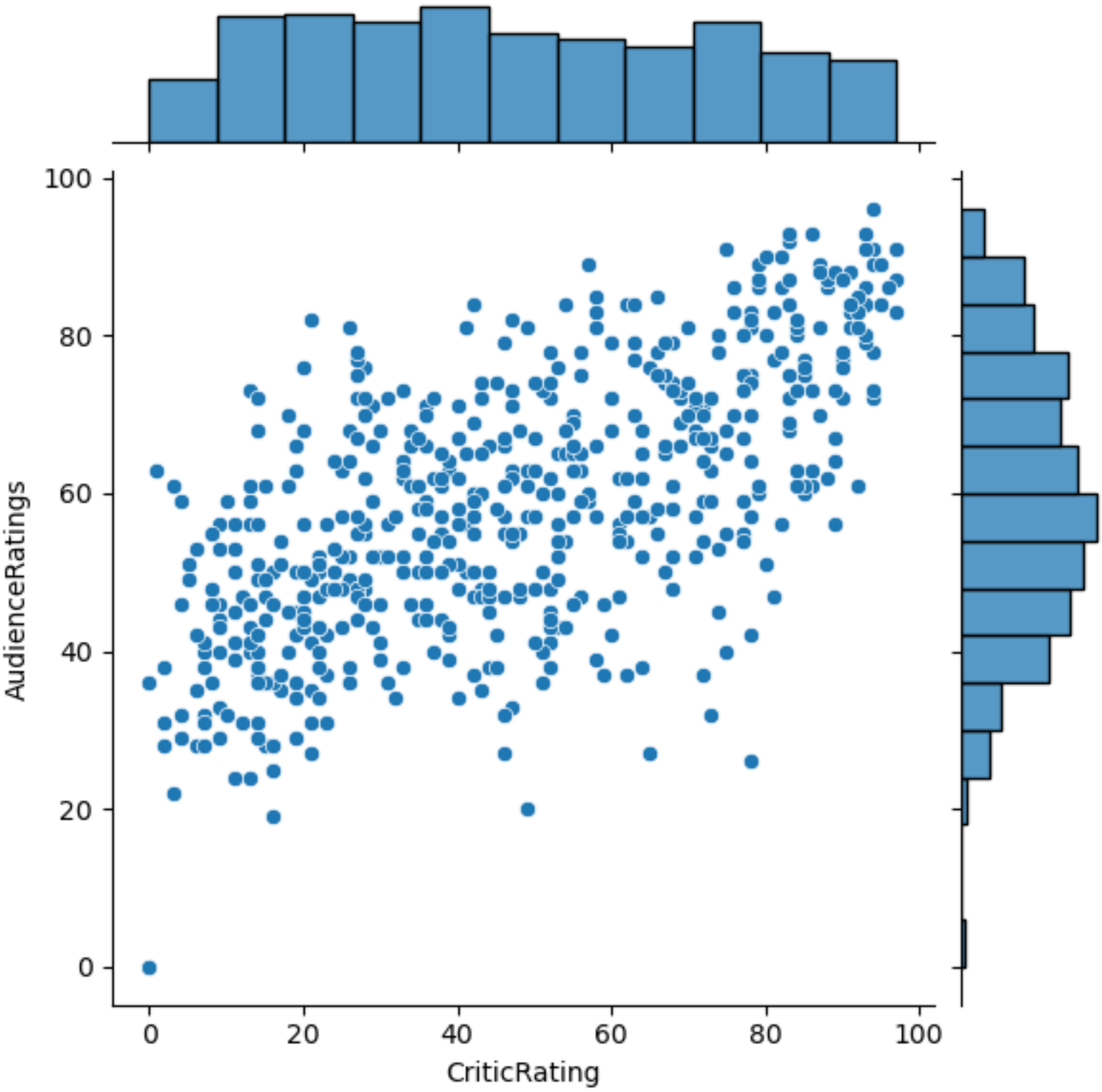


In [26]:

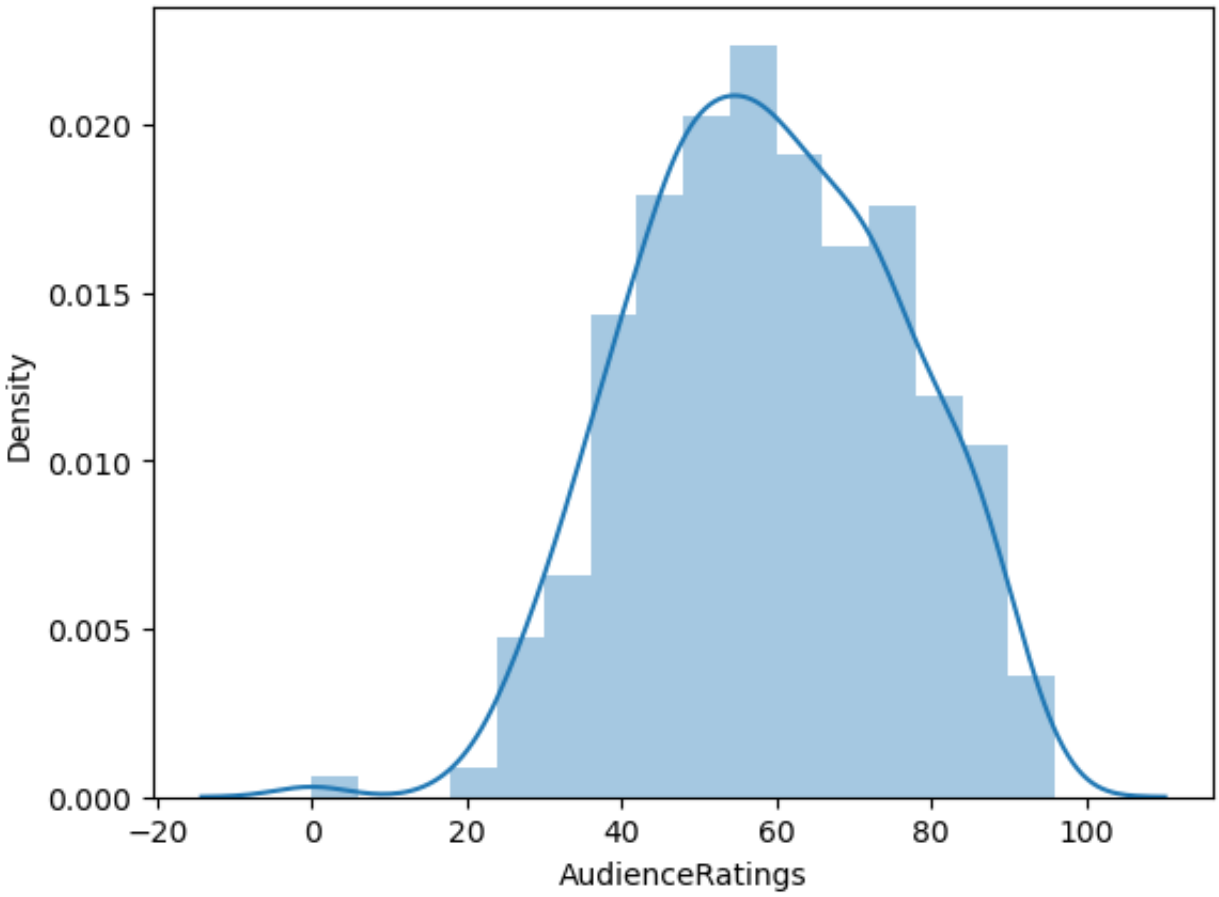
```
j = sns.jointplot( data = movies,x = 'CriticRating' , y = 'AudienceRating
```



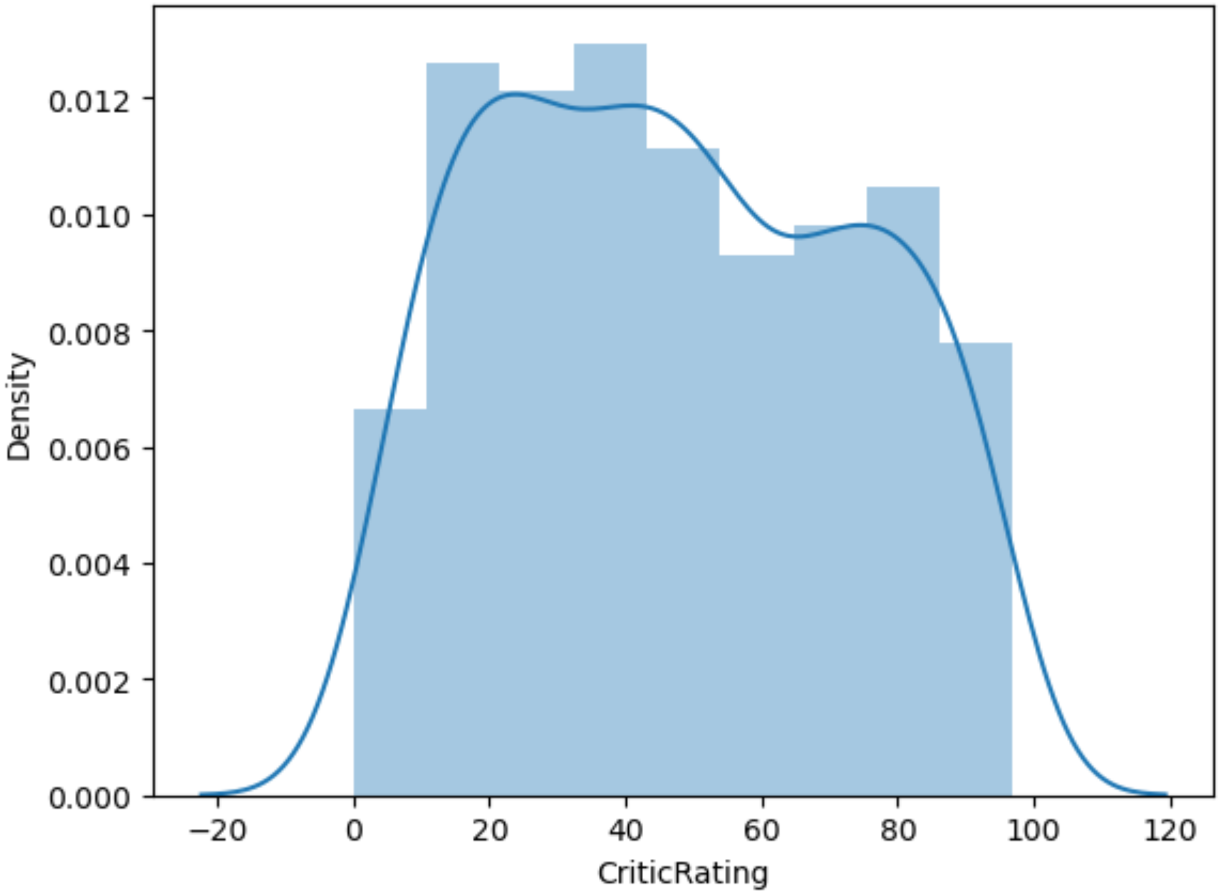
```
In [27]: j = sns.jointplot( data = movies,x = 'CriticRating' , y = 'AudienceRating
```



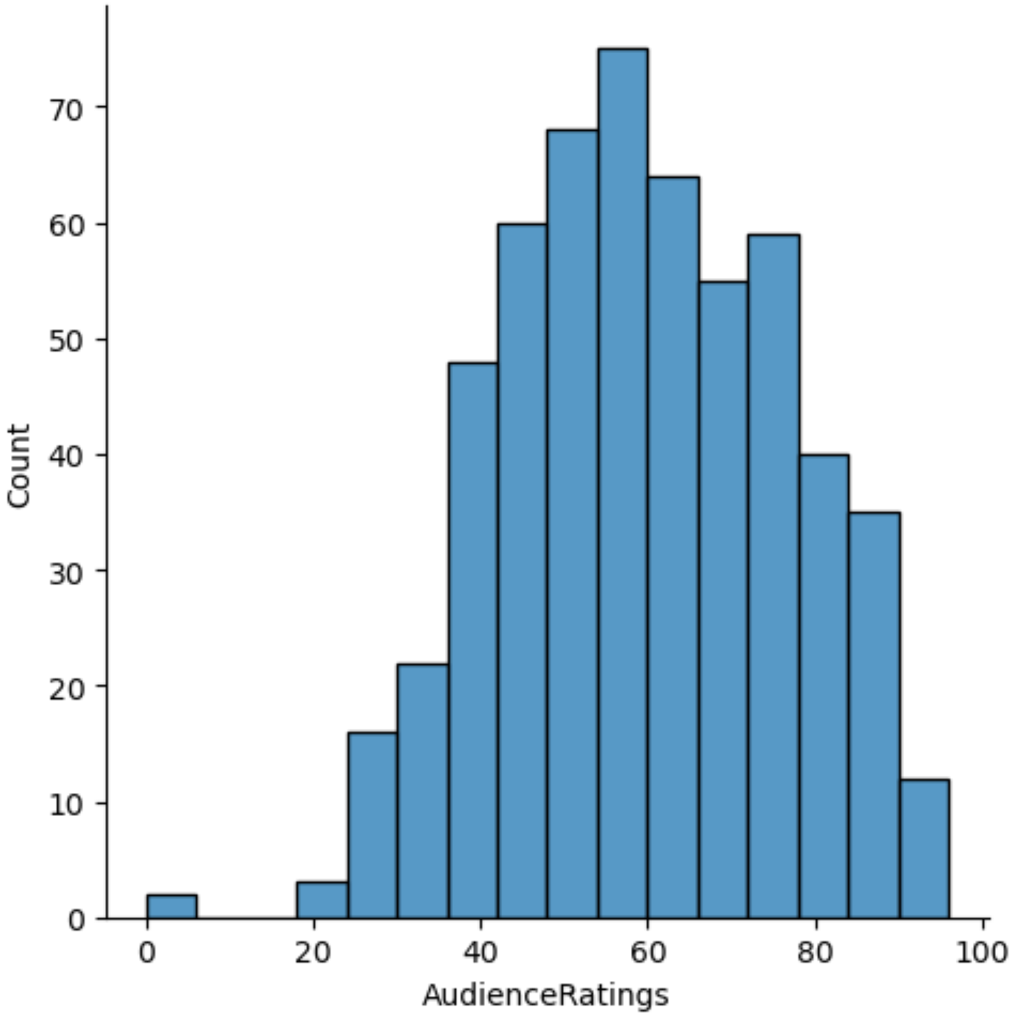
```
In [28]: m1 = sns.distplot(movies.AudienceRatings)
```



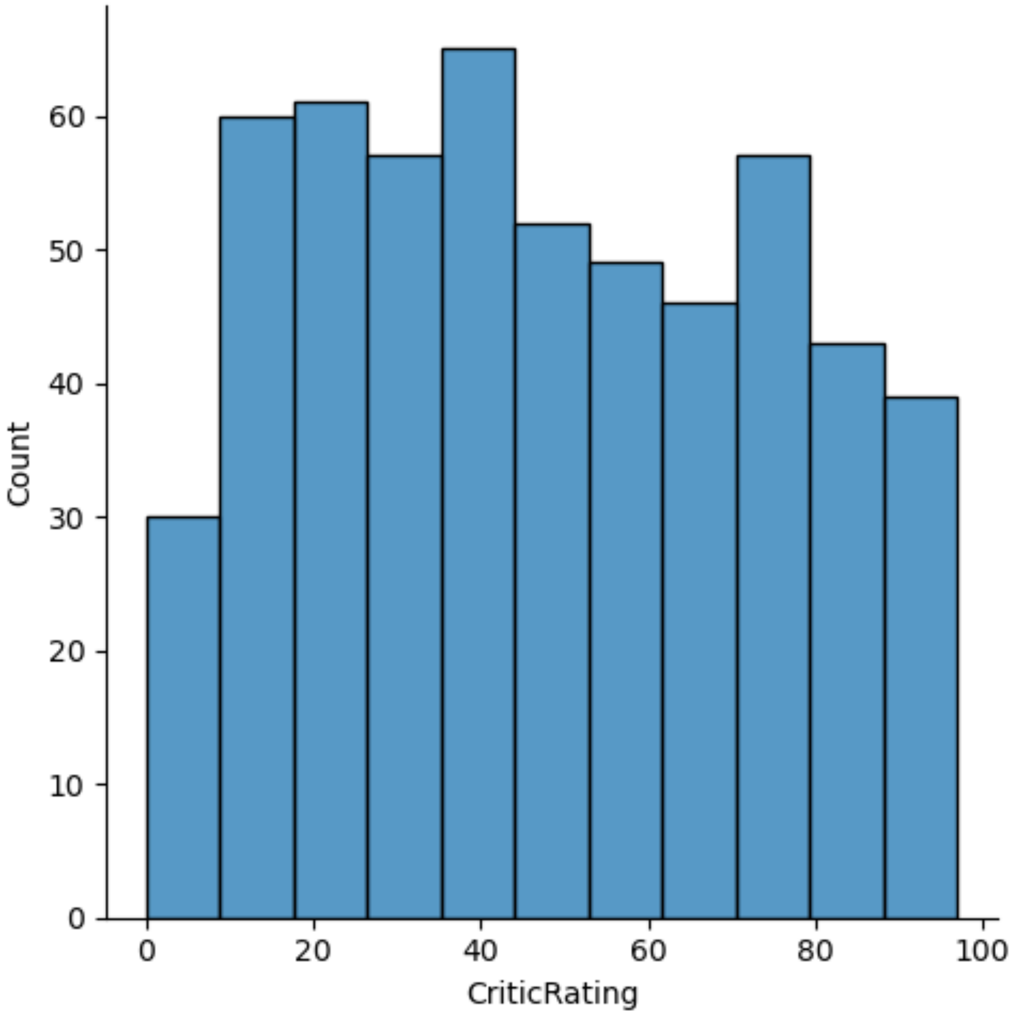
```
In [29]: m2 = sns.distplot(movies.CriticRating)
```



```
In [30]: m3 = sns.displot(movies.AudienceRatings)
```

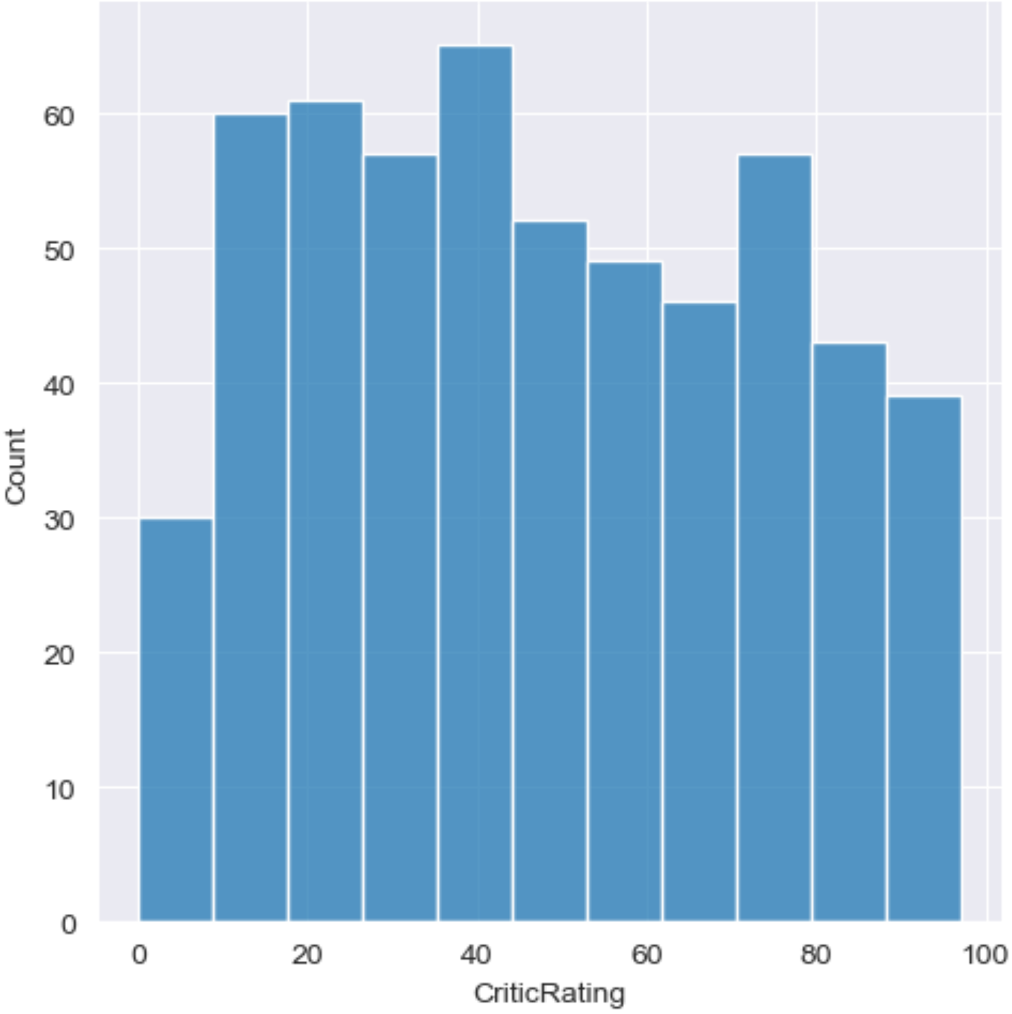


```
In [31]: m4 = sns.displot(movies.CriticRating)
```



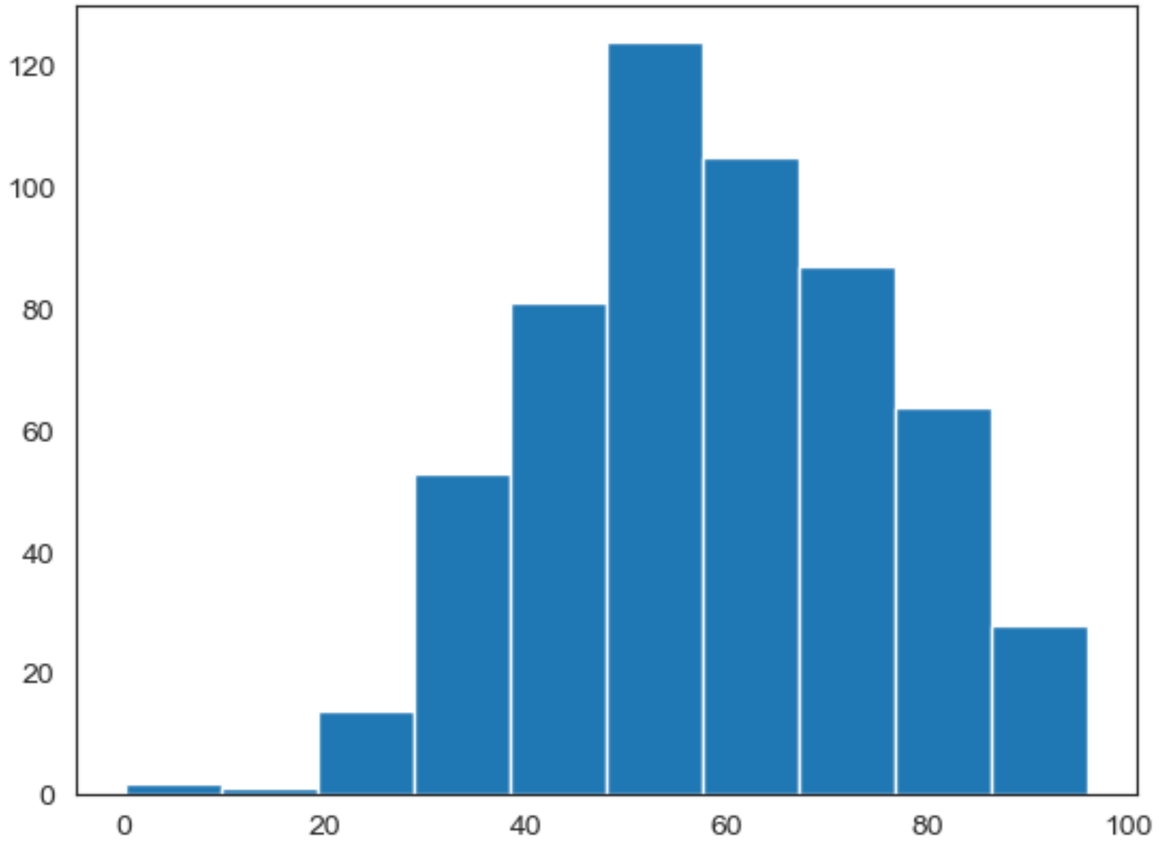
```
In [32]: sns.set_style('darkgrid')
```

```
In [33]: m4 = sns.displot(movies.CriticRating)
```

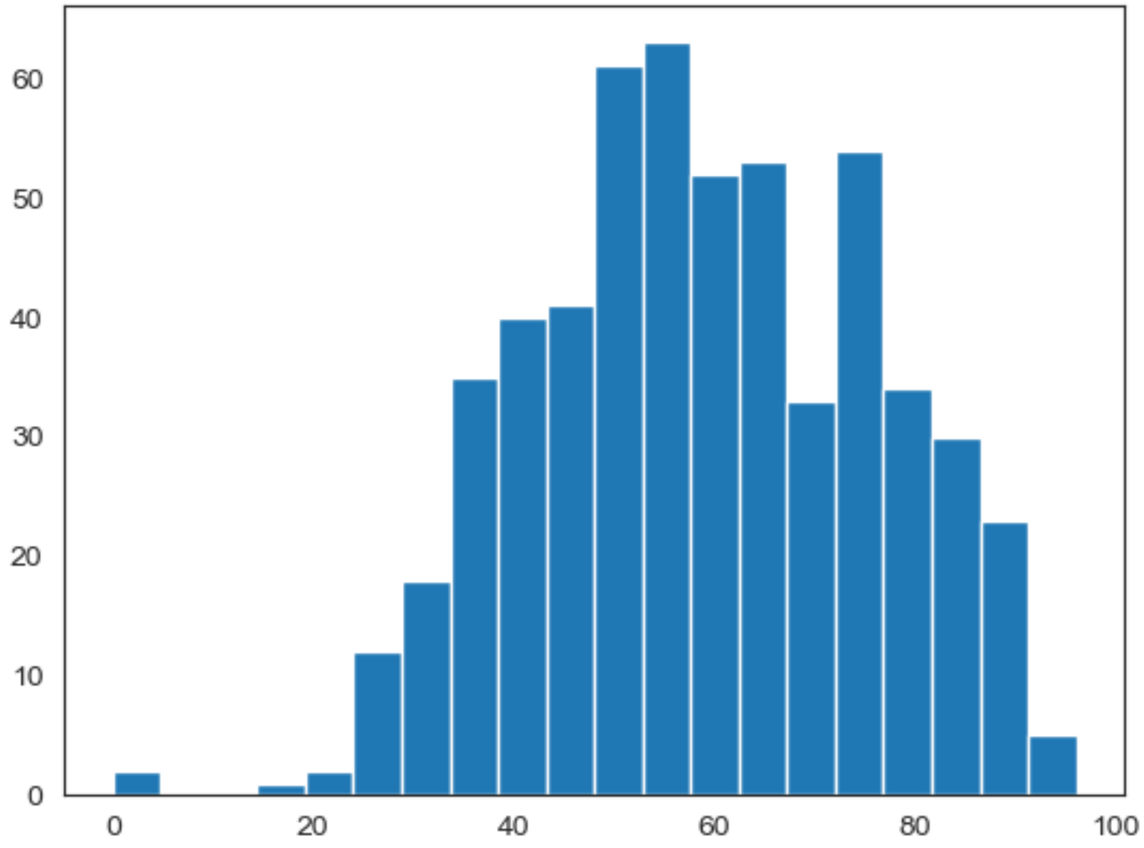


In [34]:
```
sns.set_style('white')
```

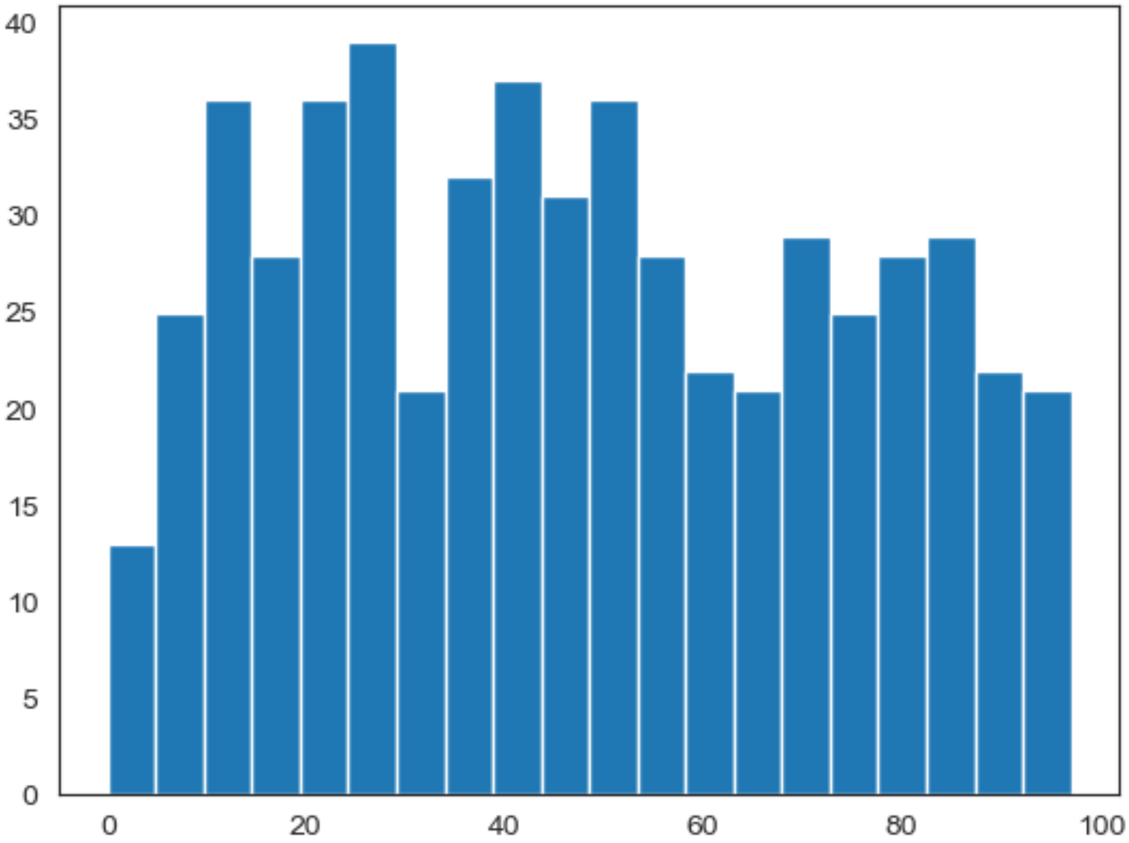
In [35]:
```
n1 = plt.hist(movies.AudienceRatings)
```



```
In [36]: n1 = plt.hist(movies.AudienceRatings,bins =20)
```

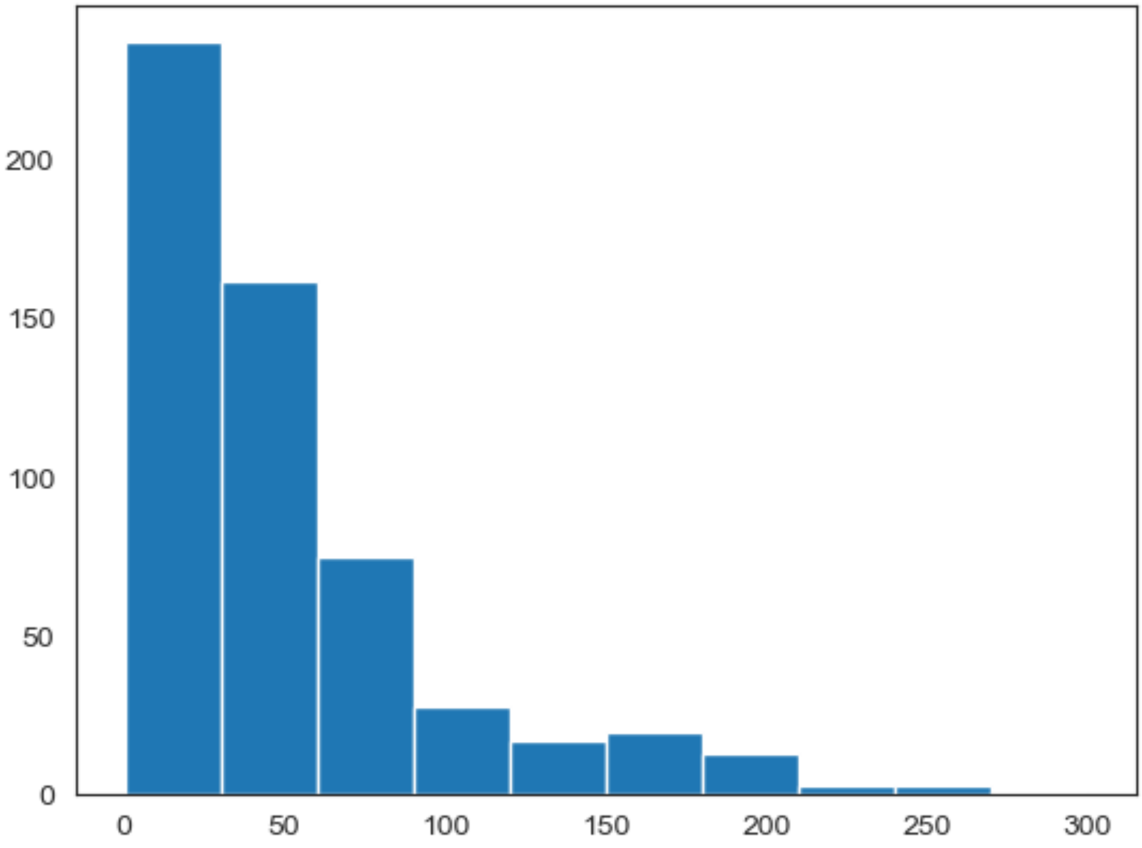


```
In [37]: n2 = plt.hist(movies.CriticRating,bins =20)
```

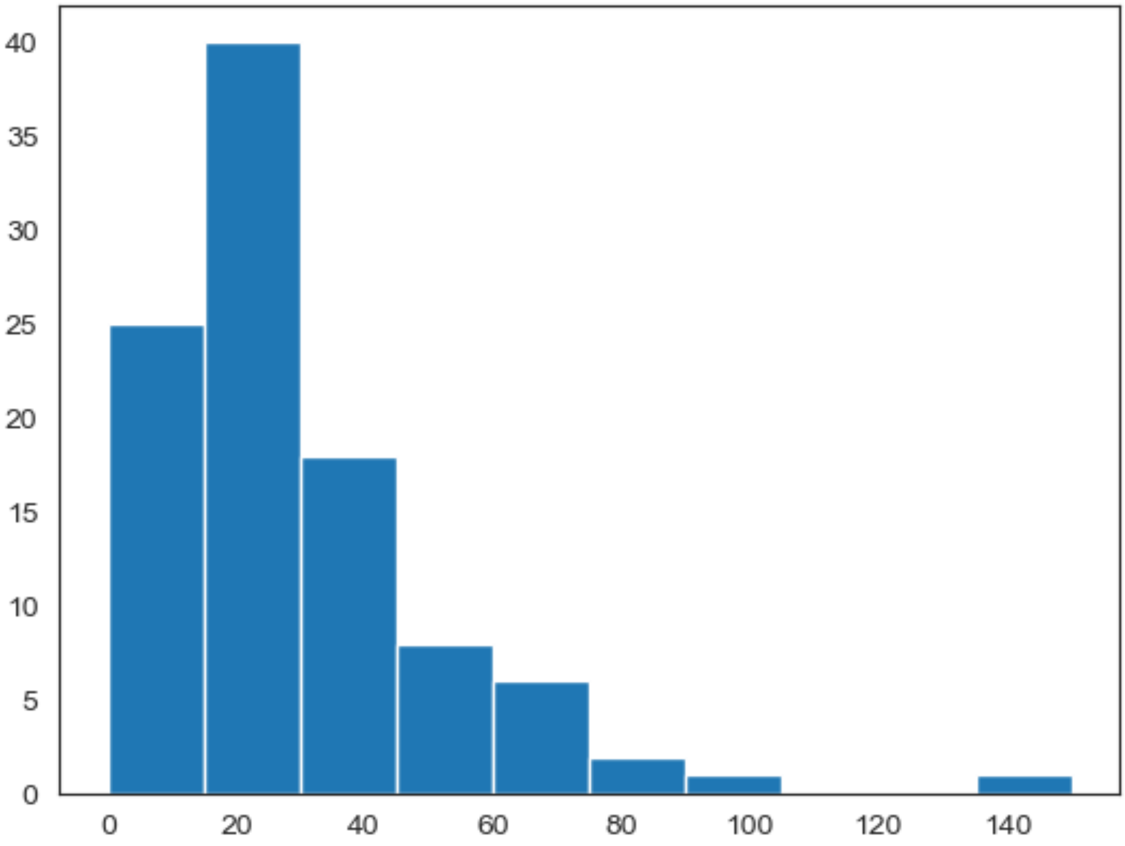


```
In [38]: plt.hist(movies.BudgetMillions)
```

```
Out[38]: (array([237., 162.,  75.,  28.,  17.,  20.,  13.,   3.,   3.,   1.]),
          array([  0.,  30.,  60.,  90., 120., 150., 180., 210., 240., 270., 30
         0.]),
          <BarContainer object of 10 artists>)
```



In [39]:
```
plt.hist(movies[movies.Genre == 'Drama'].BudgetMillions)
plt.show()
```



In [40]:

```
movies.head()
```

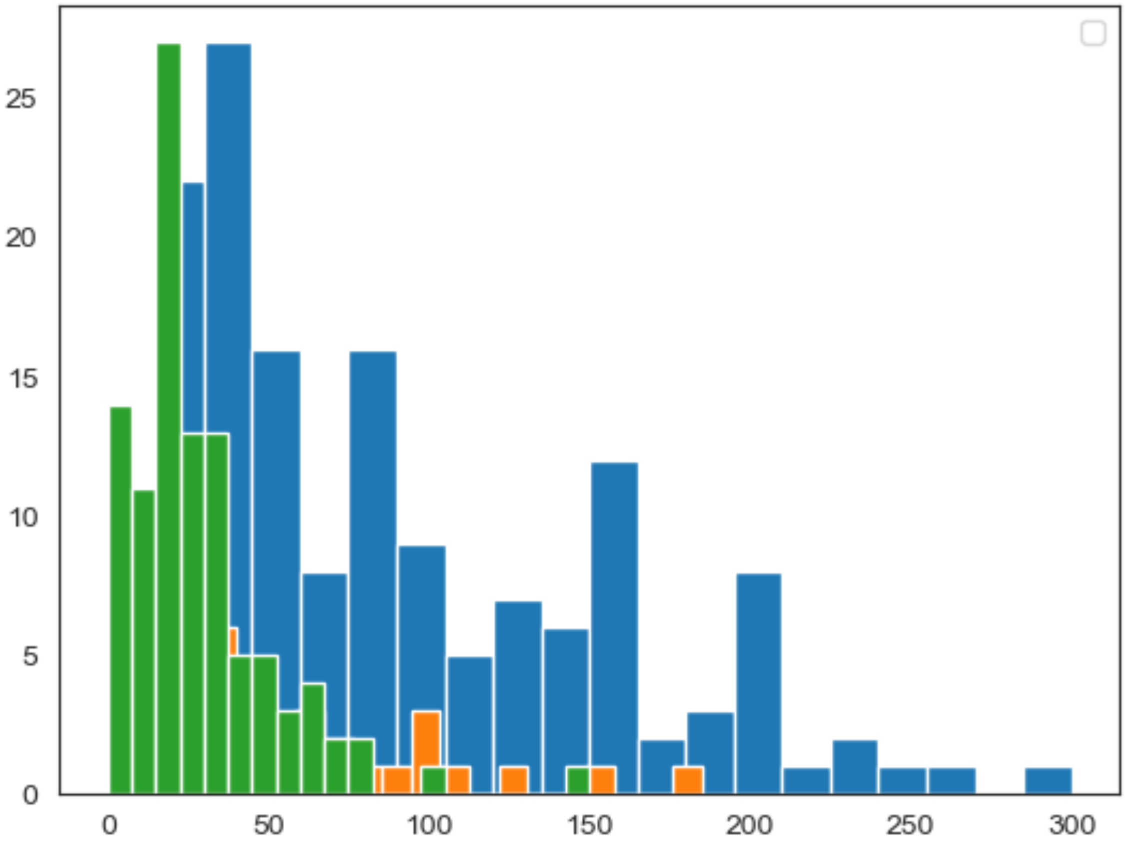
Out[40]:

| | Film | Genre | CriticRating | AudienceRatings | 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 [41]:

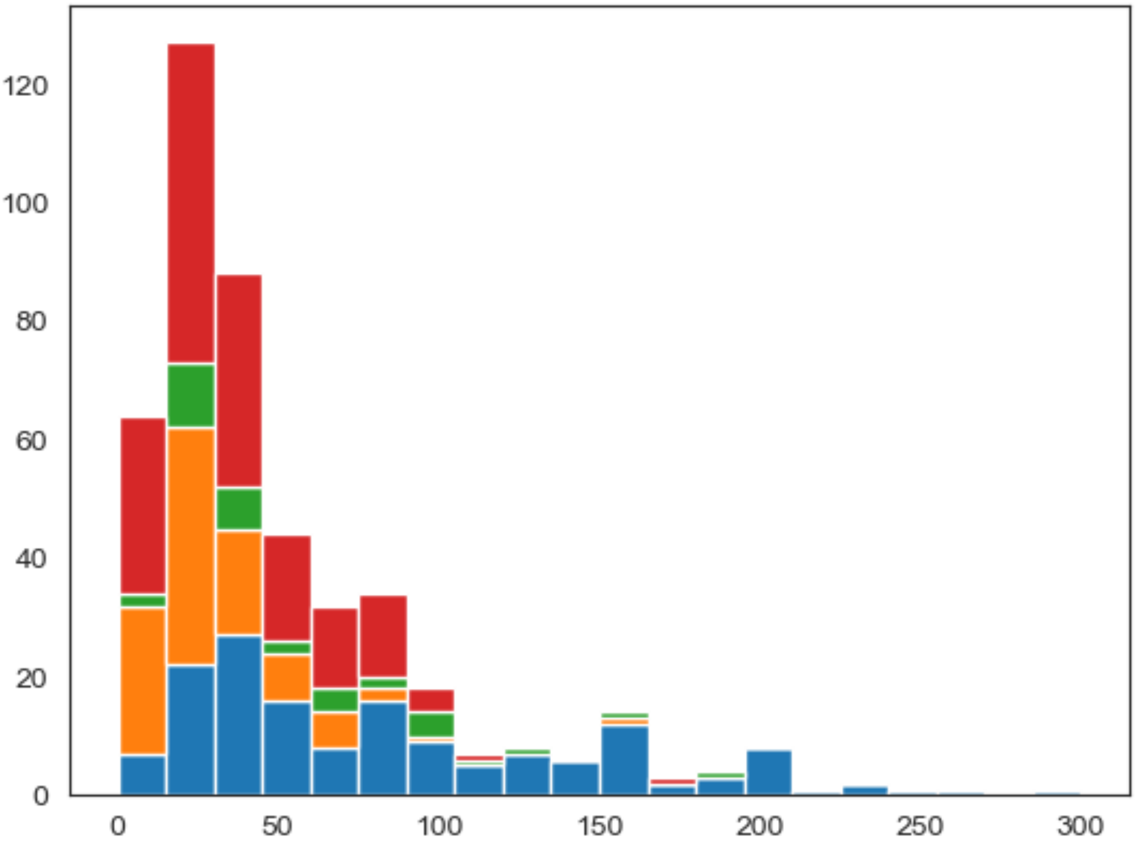
```
plt.hist(movies[movies.Genre == 'Action'].BudgetMillions, bins = 20)
plt.hist(movies[movies.Genre == 'Thriller'].BudgetMillions , bins =20)
plt.hist(movies[movies.Genre == 'Drama' ].BudgetMillions , bins = 20)
plt.legend()
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose la
bel start with an underscore are ignored when legend() is called with no a
rgument.
```



In [42]:

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



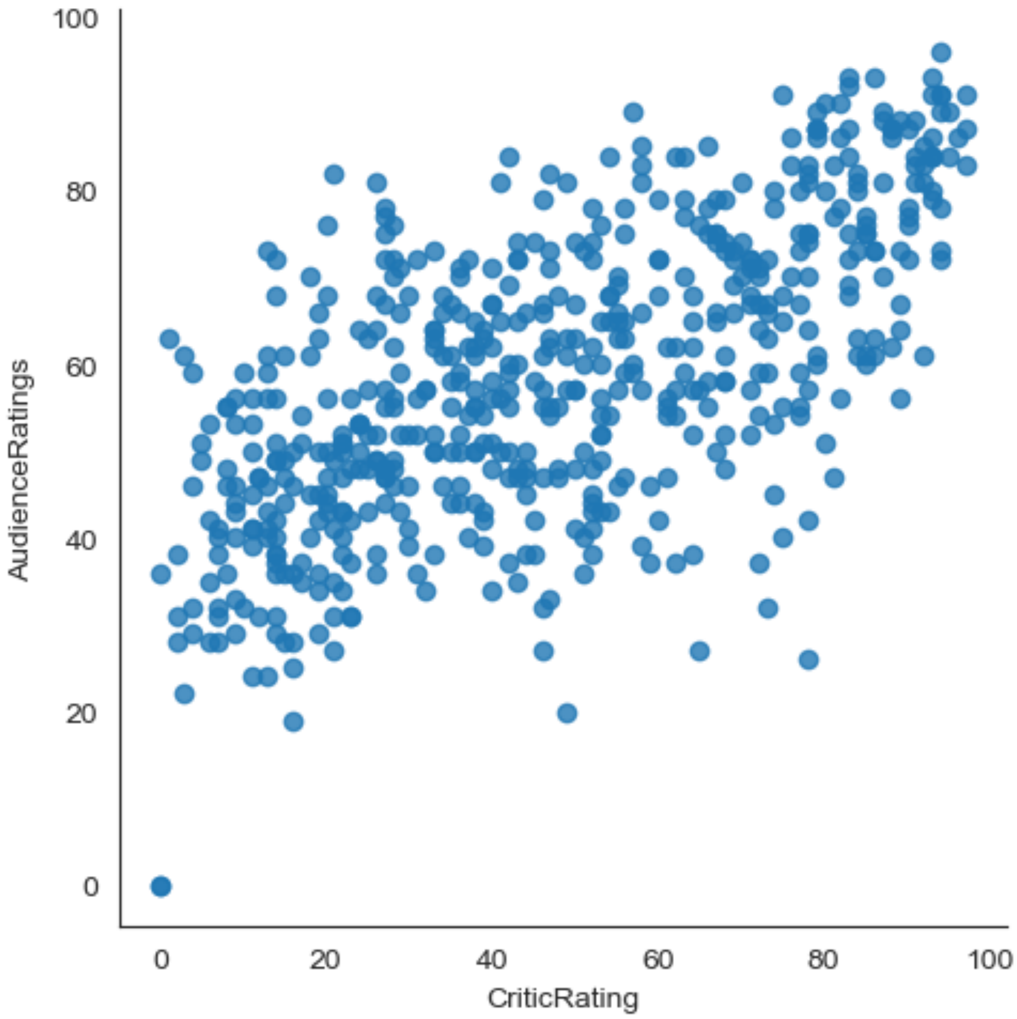
In [43]:

```
for gen in movies.Genre.cat.categories:
    print(gen)
```

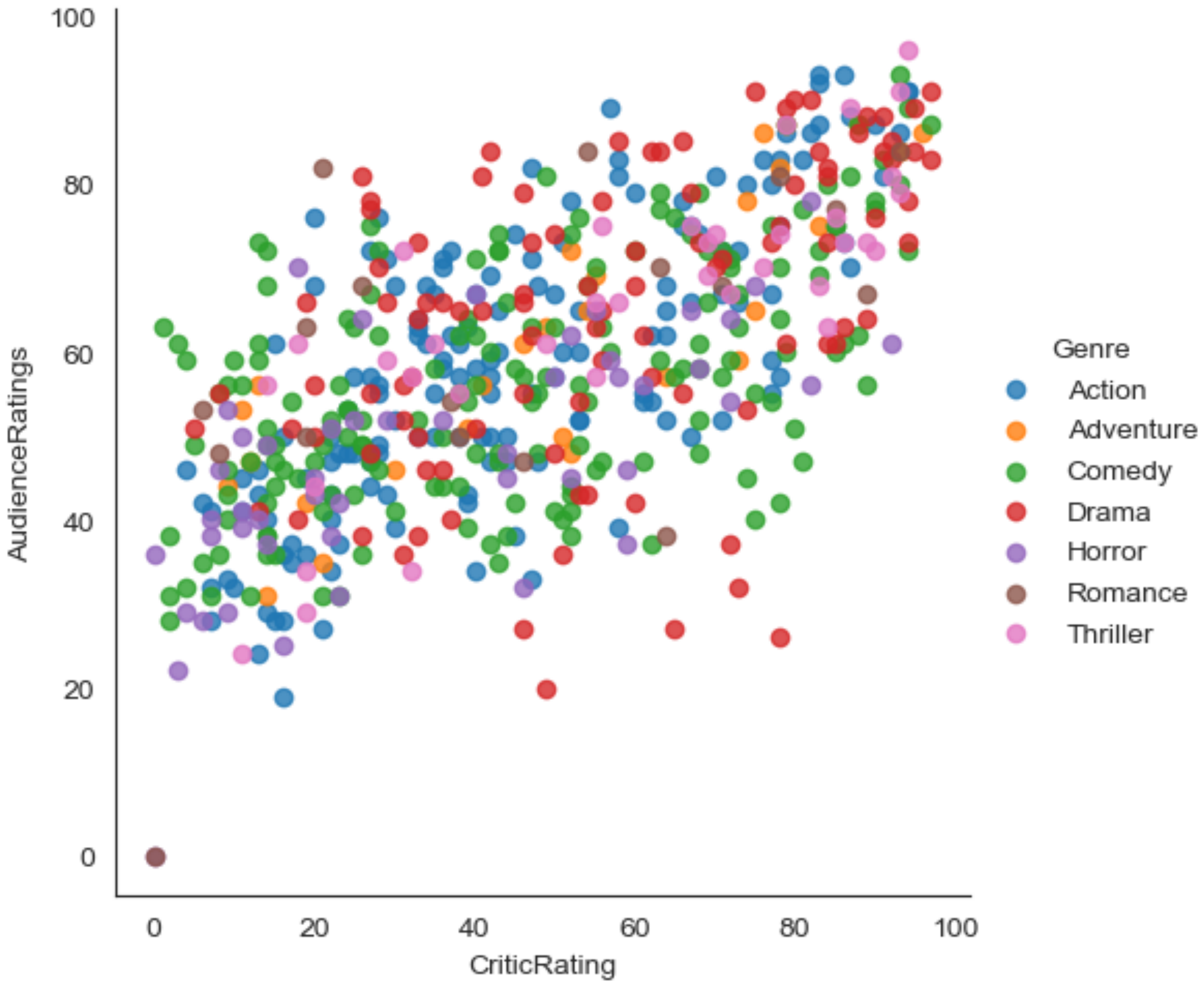
```
Action
Adventure
Comedy
Drama
Horror
Romance
Thriller
```

In [44]:

```
vis1 = sns.lmplot(data = movies ,  x = 'CriticRating' , y = 'AudienceRati
```

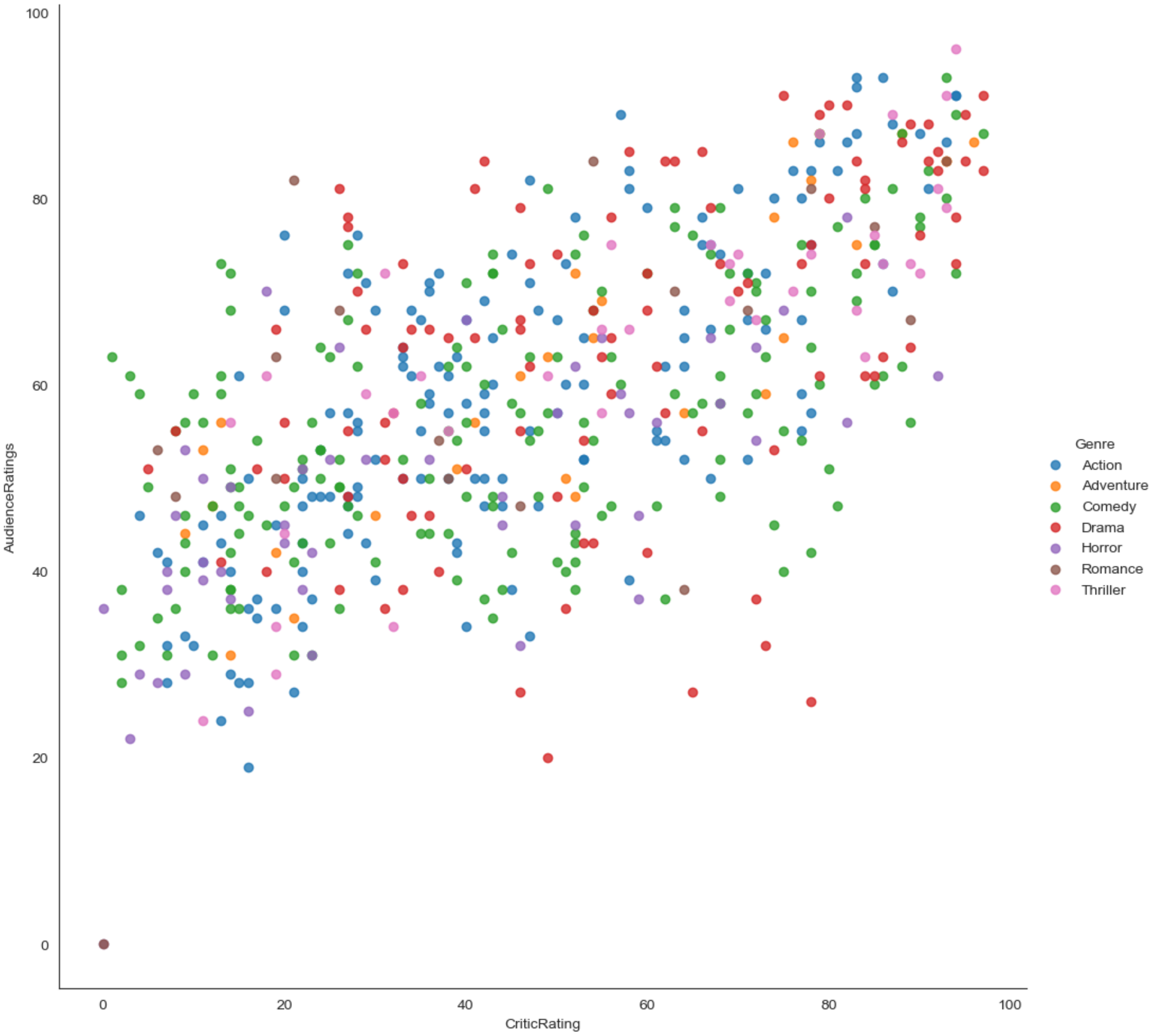


```
In [45]: vis1 = sns.lmplot(data = movies ,  x = 'CriticRating' , y = 'AudienceRati
```

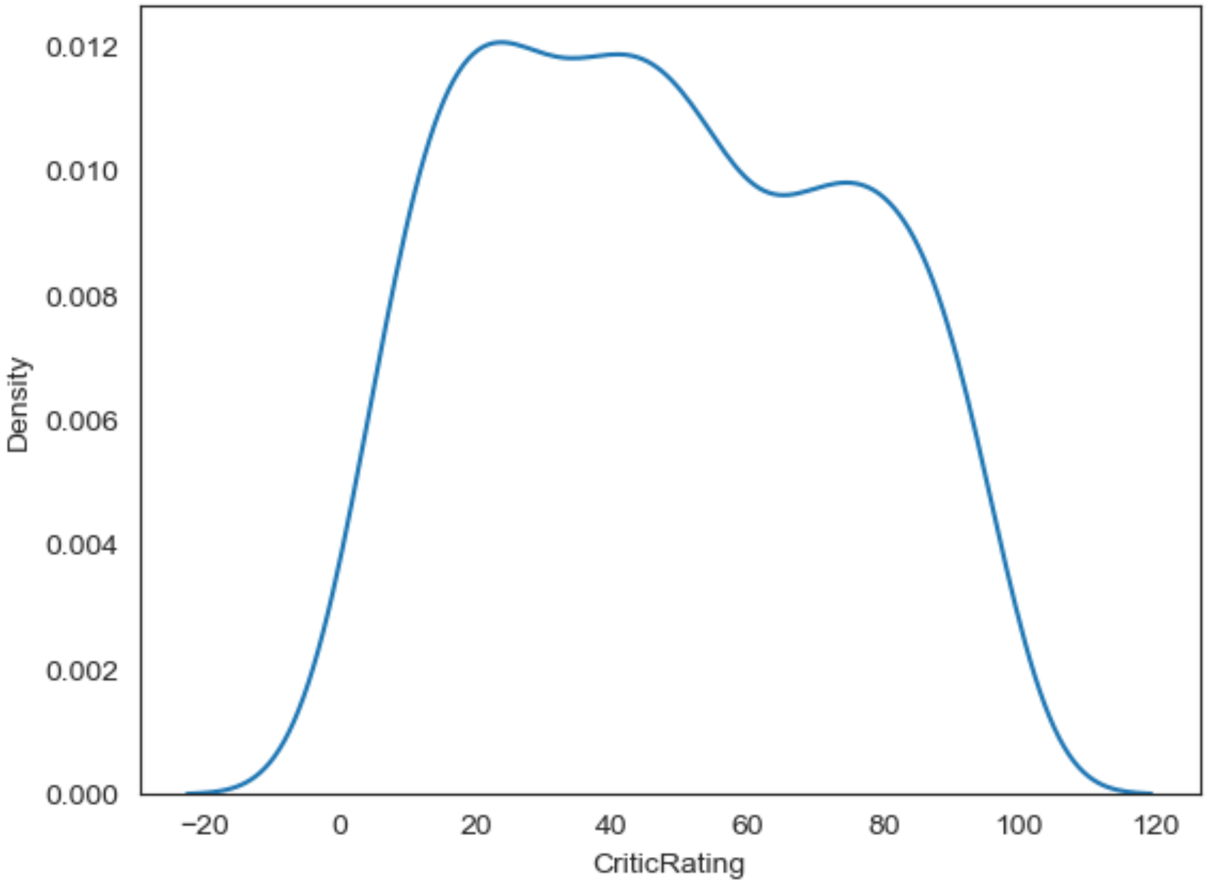


In [46]:

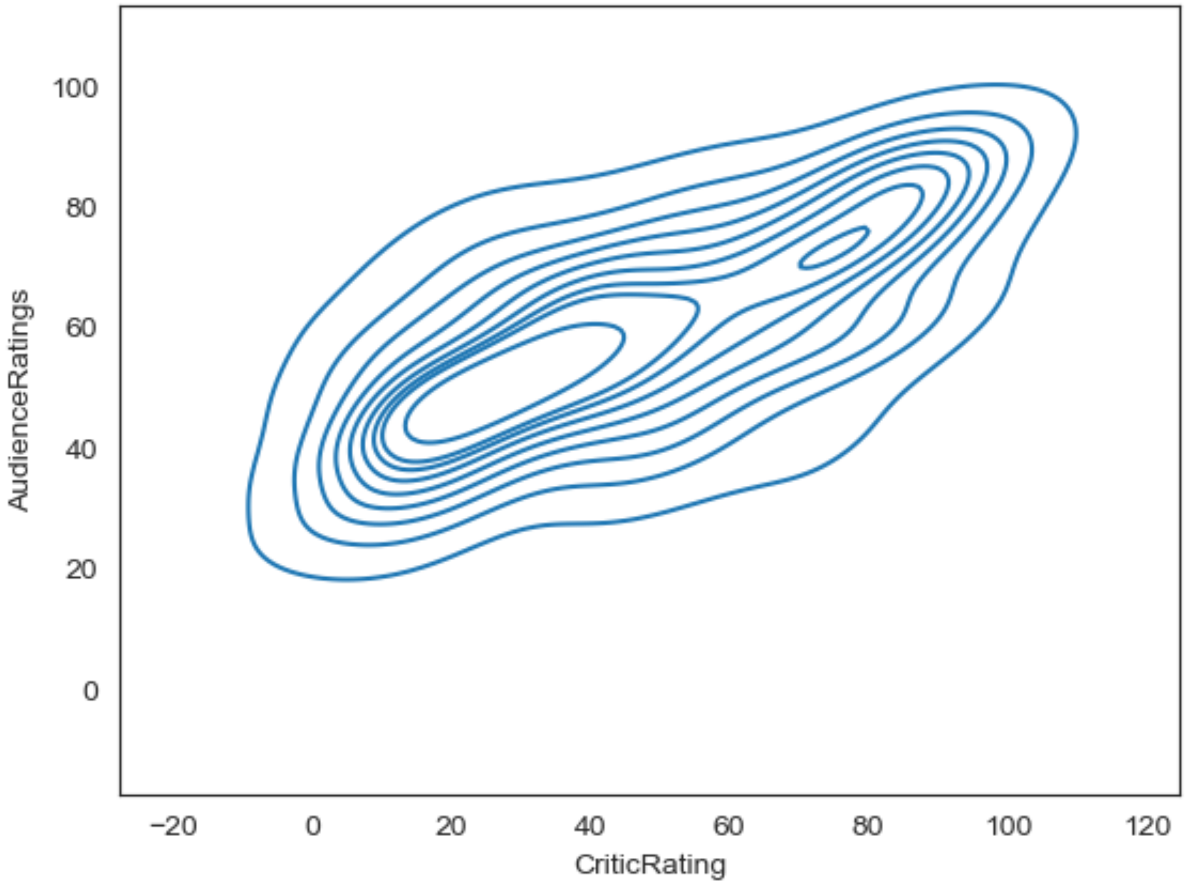
```
vis1 = sns.lmplot(data = movies ,  x = 'CriticRating' , y = 'AudienceRati
                  fit_reg = False, hue = 'Genre' , height = 10 , aspect =
```



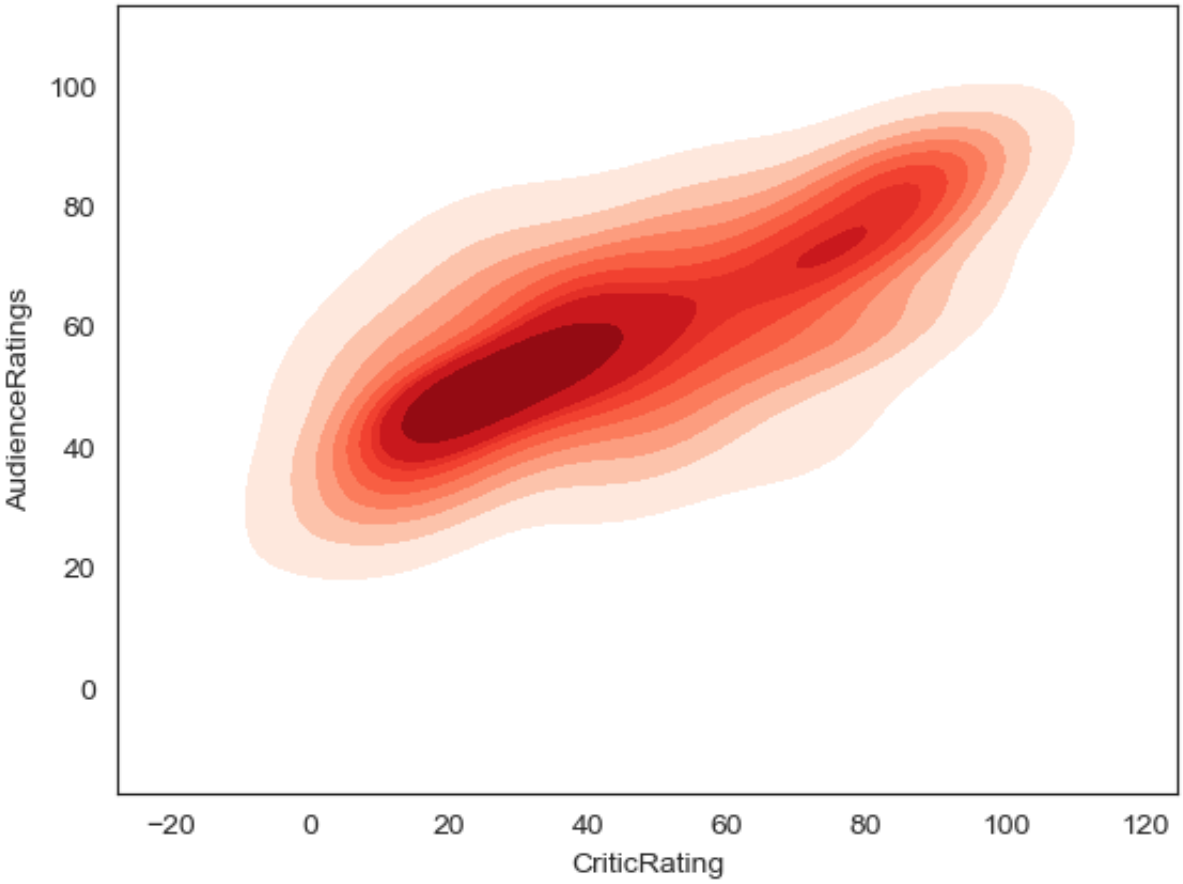
```
In [47]: k1 = sns.kdeplot(movies.CriticRating)
```



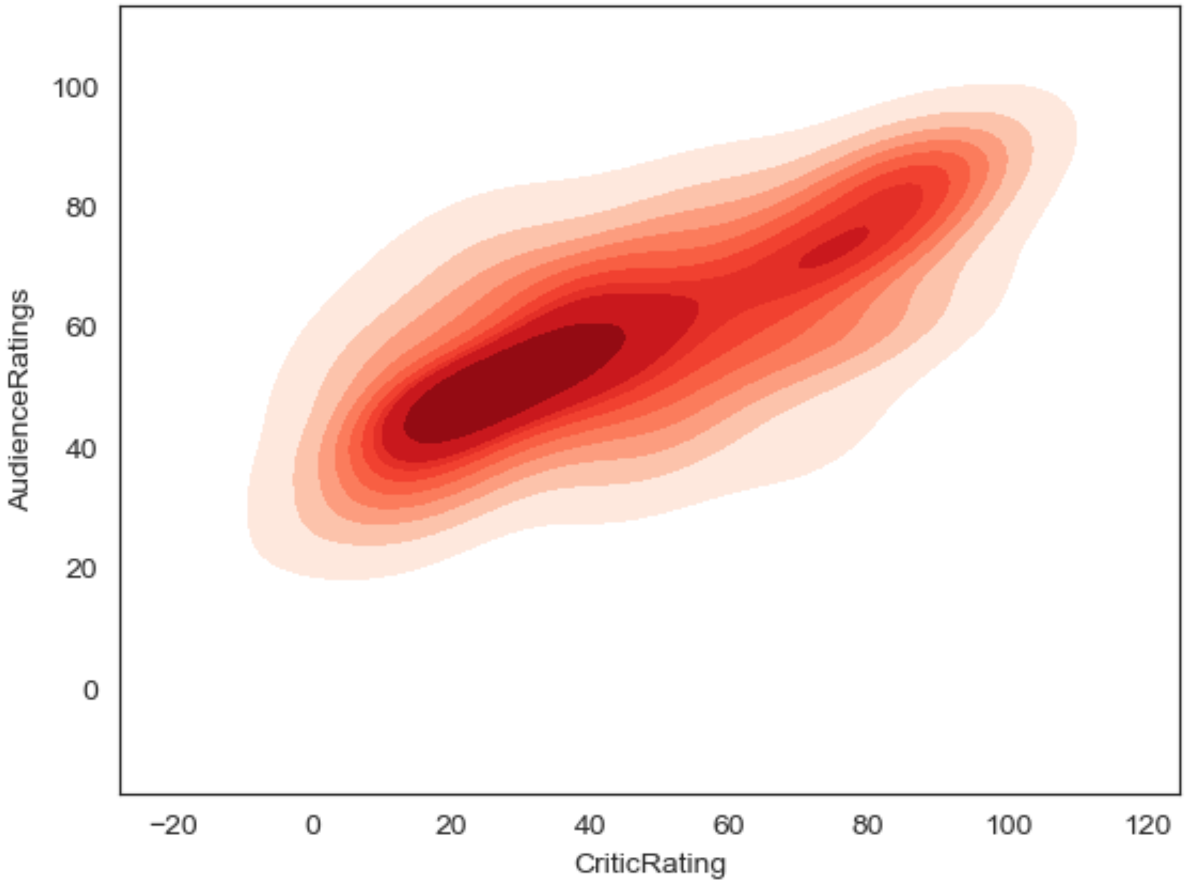
```
In [48]: k1 = sns.kdeplot(data = movies , x='CriticRating' ,y = 'AudienceRatings')
```



In [49]:
```
k1 = sns.kdeplot(data = movies , x='CriticRating' ,y = 'AudienceRatings'
                 shade_lowest = False , cmap = 'Reds')
```

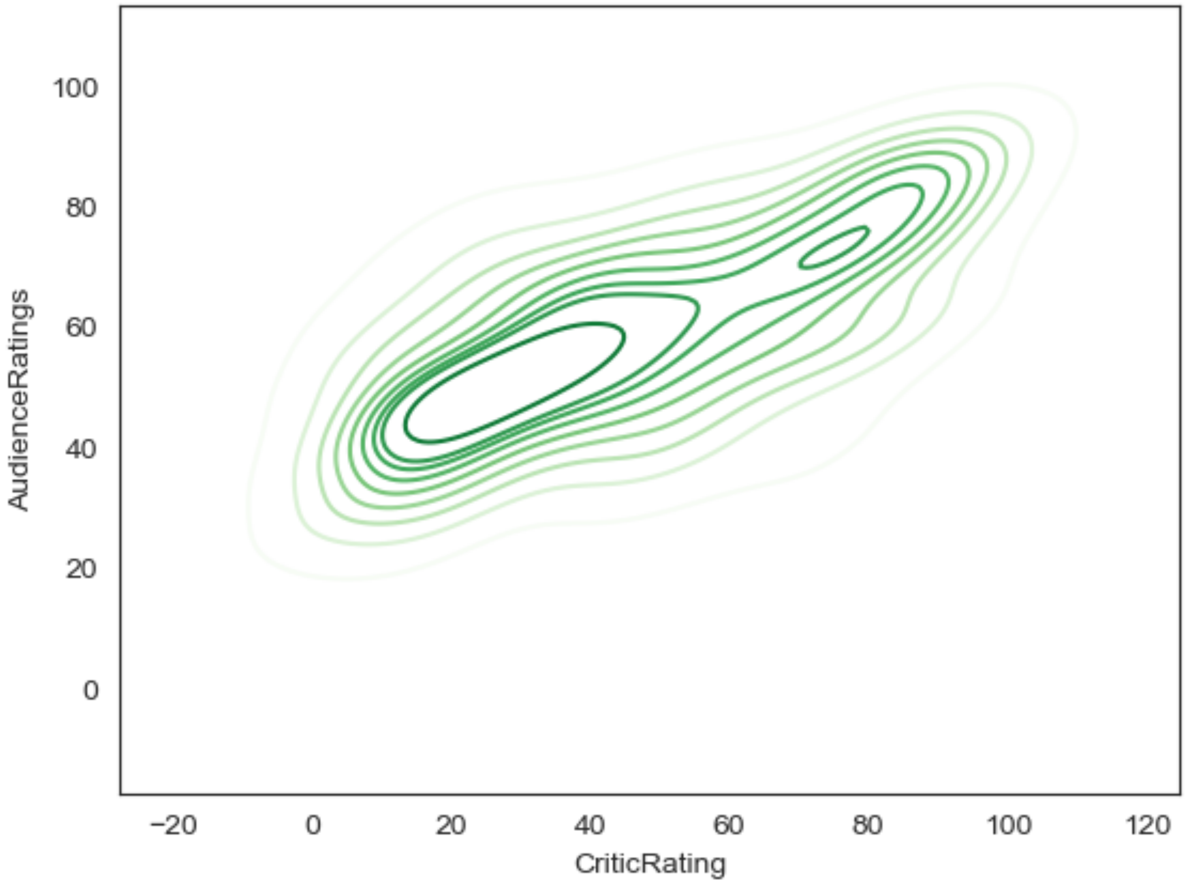


In [50]:
```
k1 = sns.kdeplot(data = movies , x='CriticRating' ,y = 'AudienceRatings'
                 cmap = 'Reds')
```

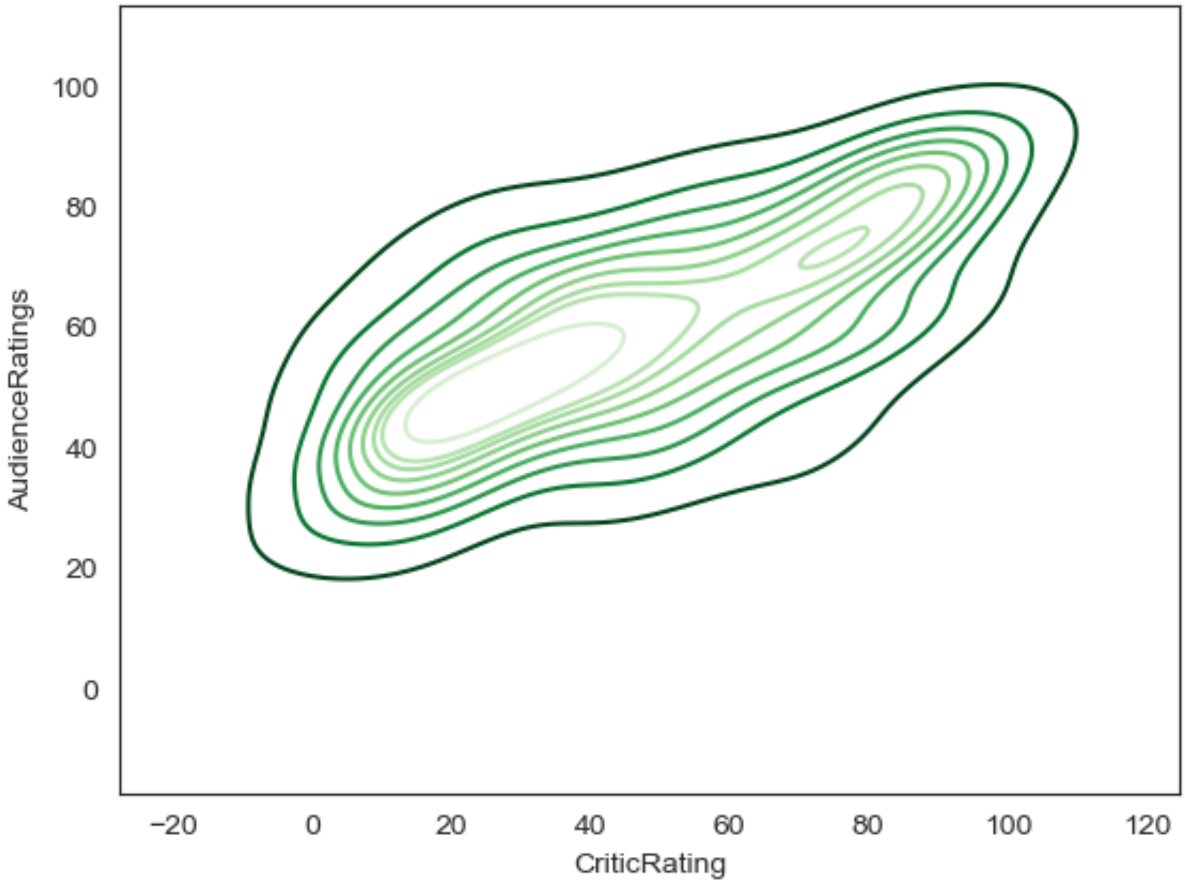


In [51]:

```
k2 = sns.kdeplot(data = movies , x='CriticRating' ,y = 'AudienceRatings'
                cmap = 'Greens')
```



```
In [52]: k2 = sns.kdeplot(data = movies , x='CriticRating' ,y = 'AudienceRatings'
                          cmap = 'Greens_r')
```

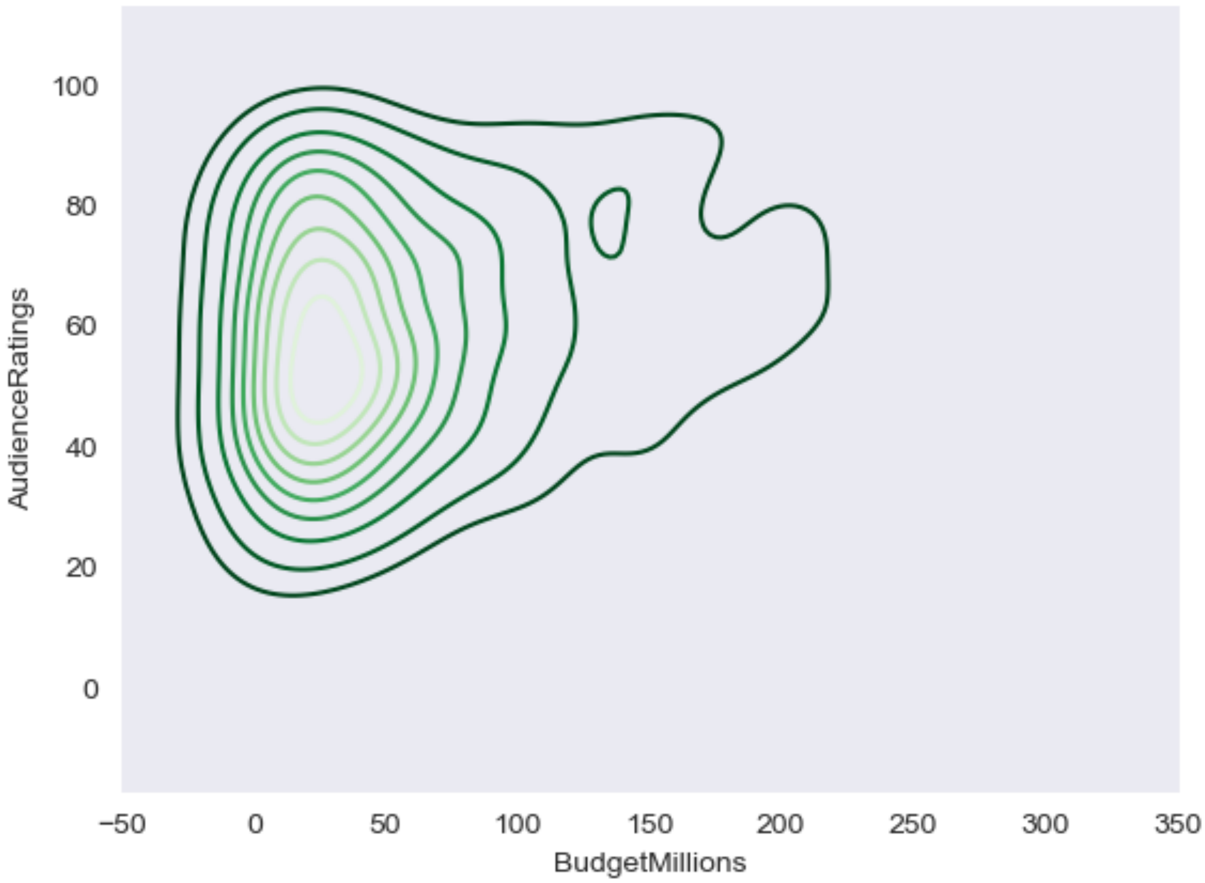


In [53]:

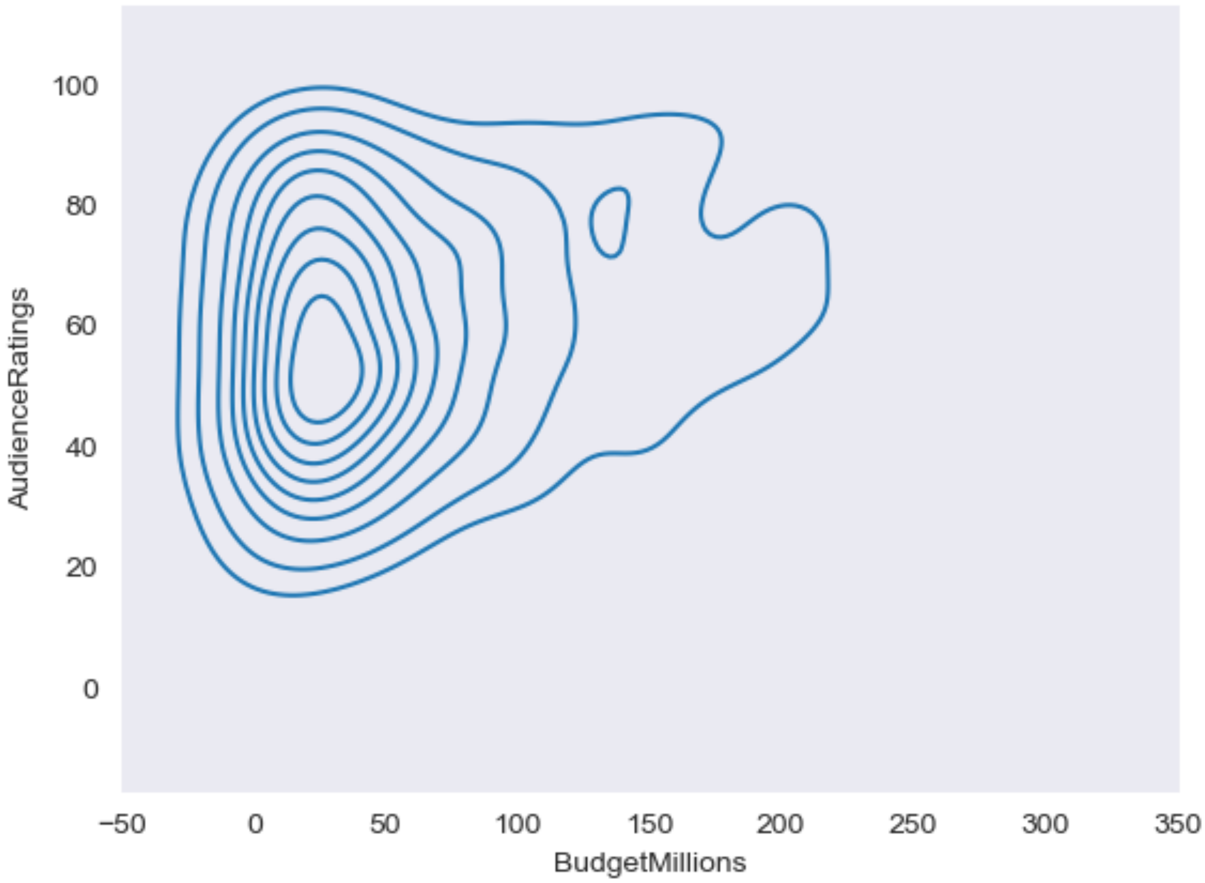
```
sns.set_style('dark')
```

In [54]:

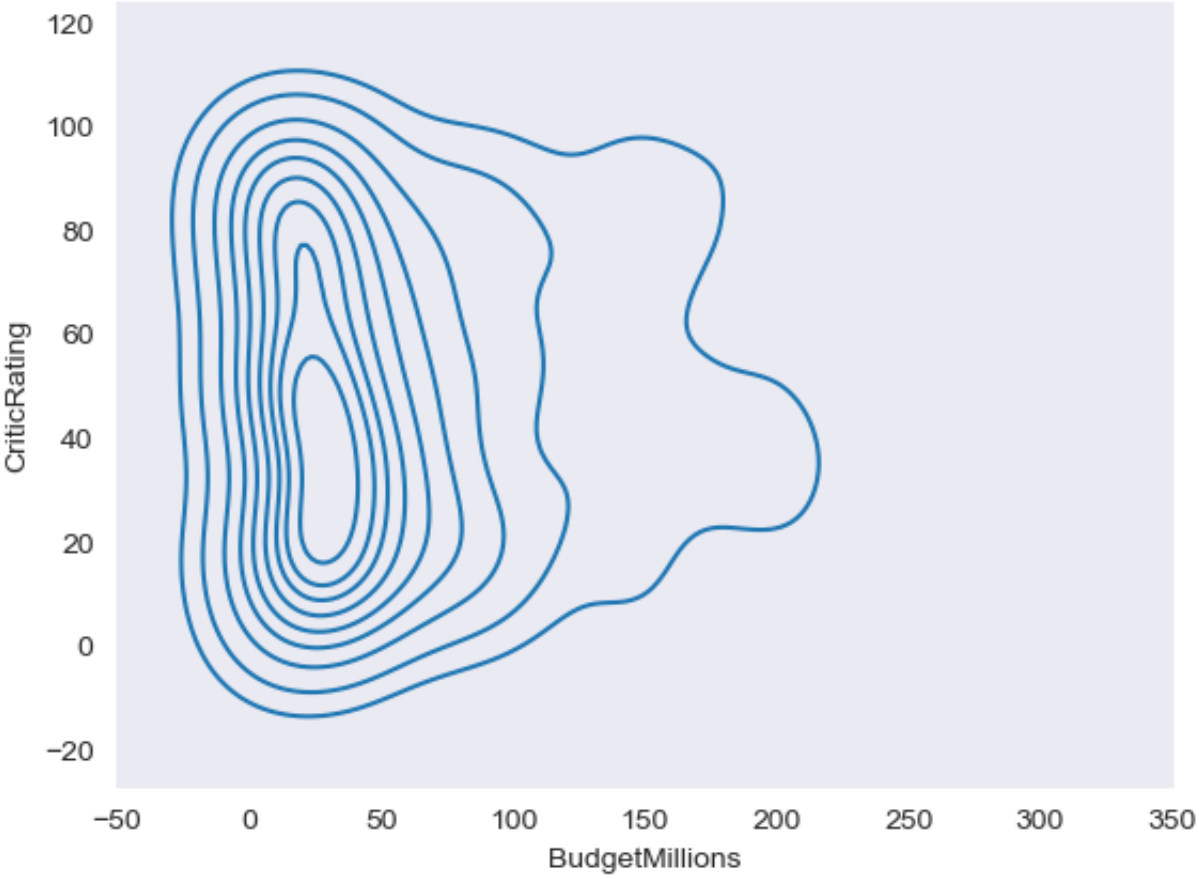
```
k1 = sns.kdeplot(data = movies , x = 'BudgetMillions', y = 'AudienceRatin
                 cmap = 'Greens_r')
```



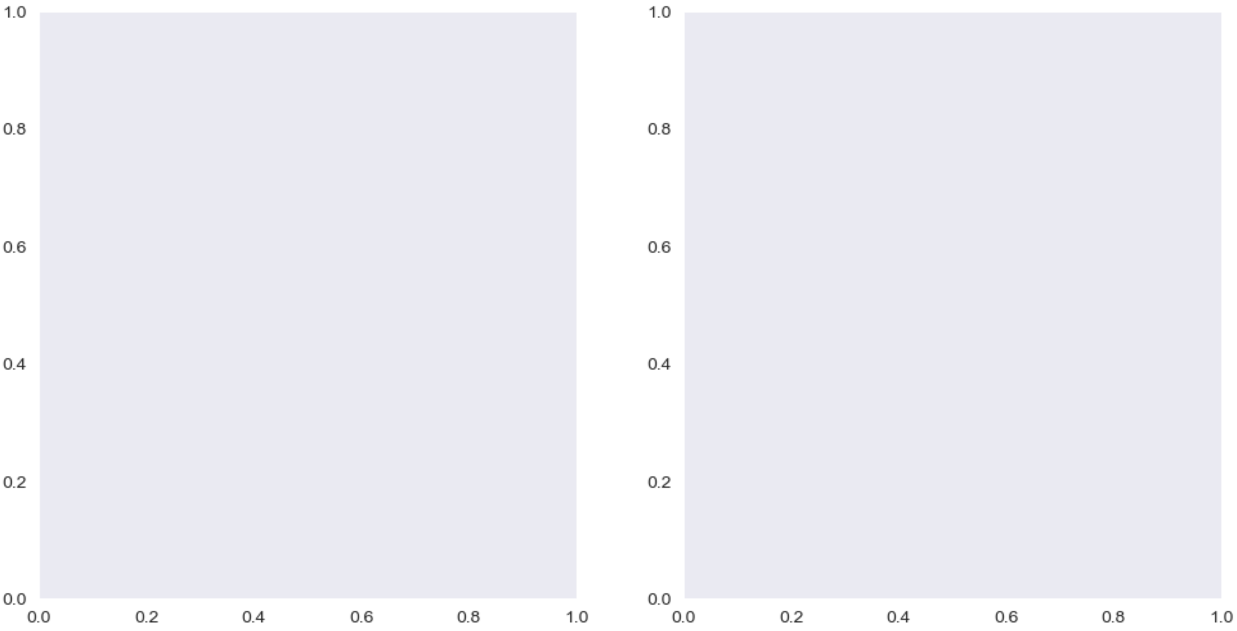
In [55]:
```
k3 = sns.kdeplot(data = movies , x = 'BudgetMillions', y = 'AudienceRatin
```



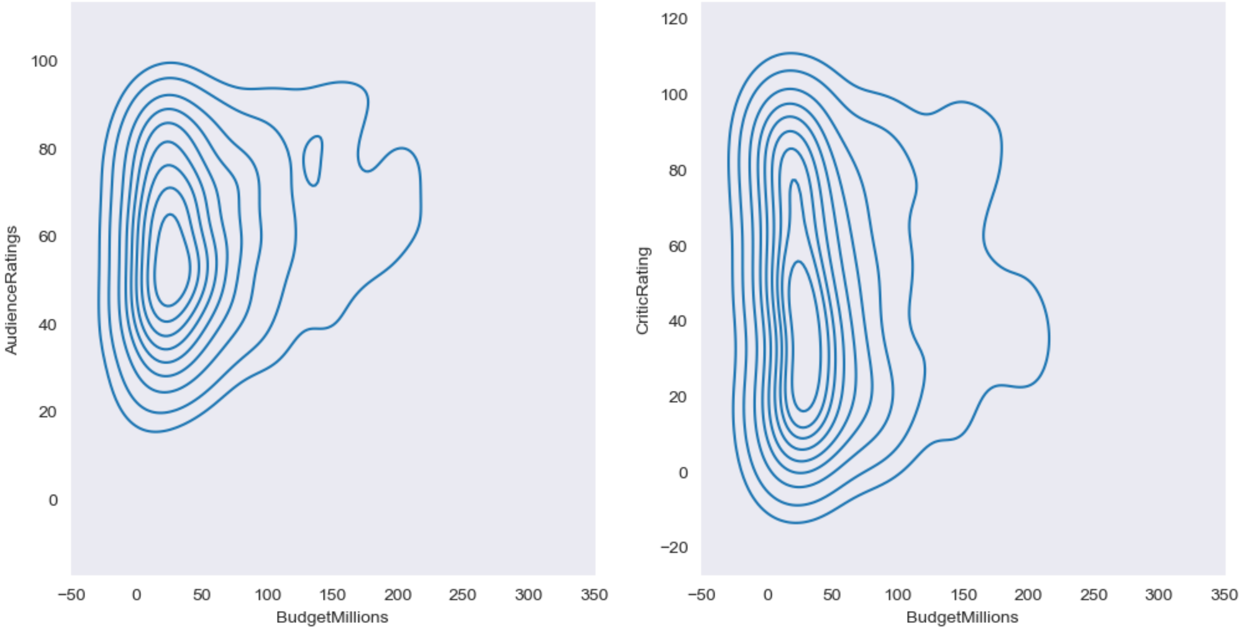
In [56]:
```
k4 = sns.kdeplot(data = movies , x = 'BudgetMillions', y = 'CriticRating'
```



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



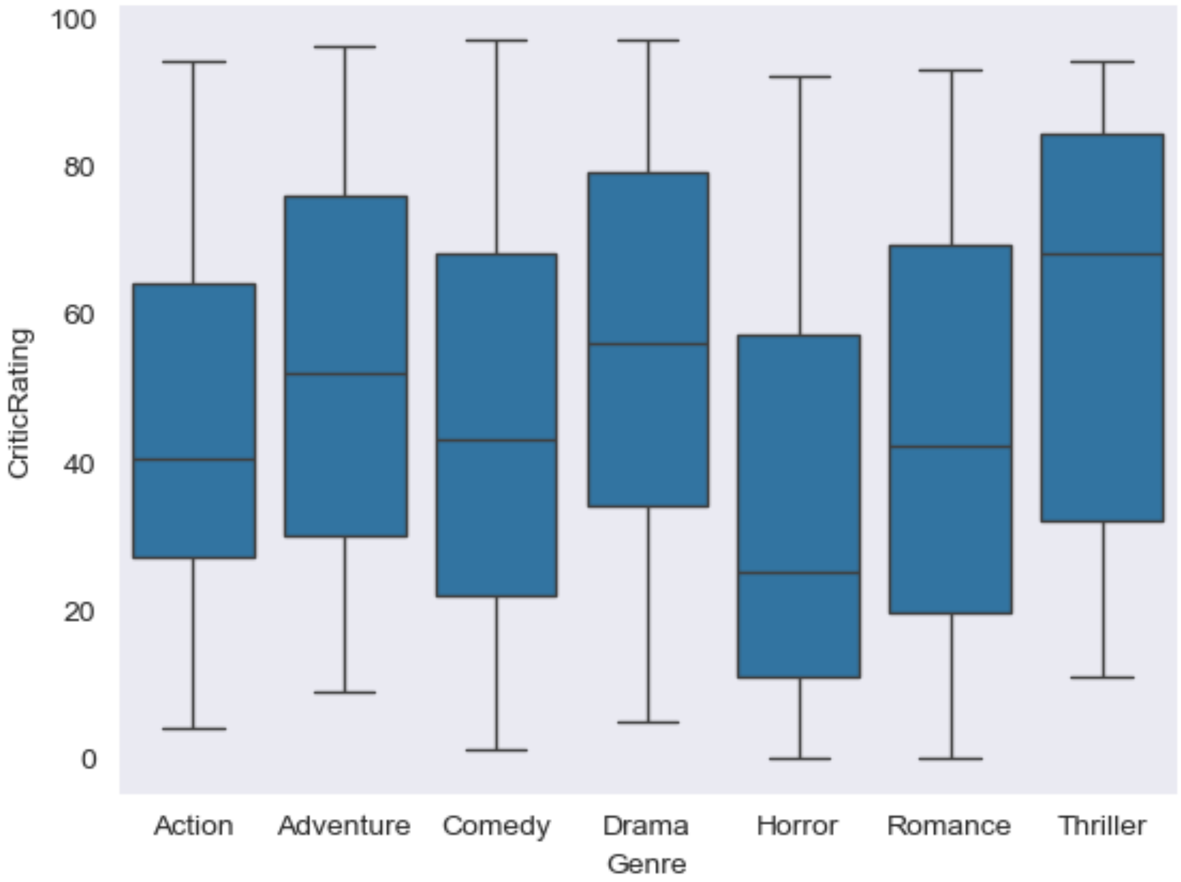
In [58]:
```
f ,  axes  = plt.subplots(1,2, figsize = (12,6))
k1 = sns.kdeplot(data = movies, x = 'BudgetMillions' , y = 'AudienceRatin
k2 = sns.kdeplot(data = movies, x = 'BudgetMillions' , y = 'CriticRating'
```



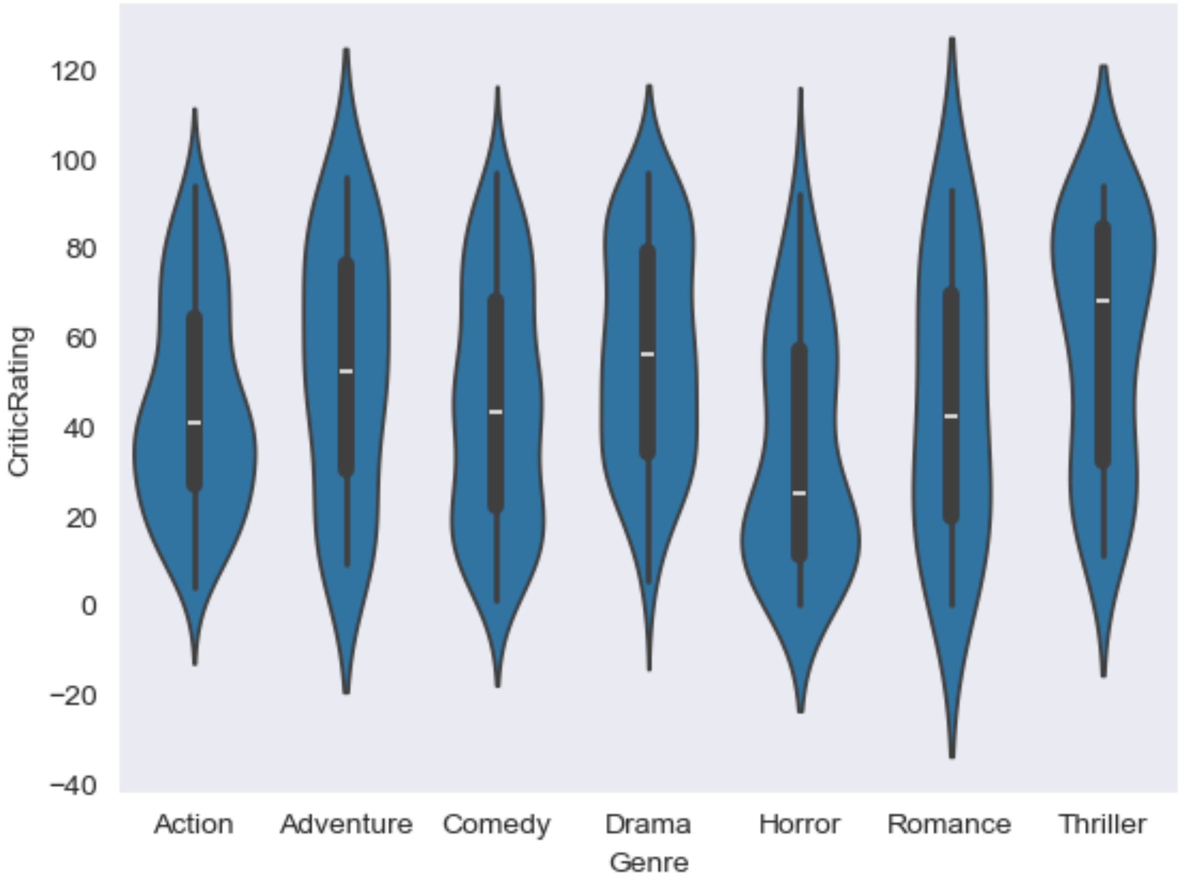
In [59]:
```
axes
```

Out[59]:
```
array([<Axes: xlabel='BudgetMillions', ylabel='AudienceRatings'>,
       <Axes: xlabel='BudgetMillions', ylabel='CriticRating'>],
      dtype=object)
```

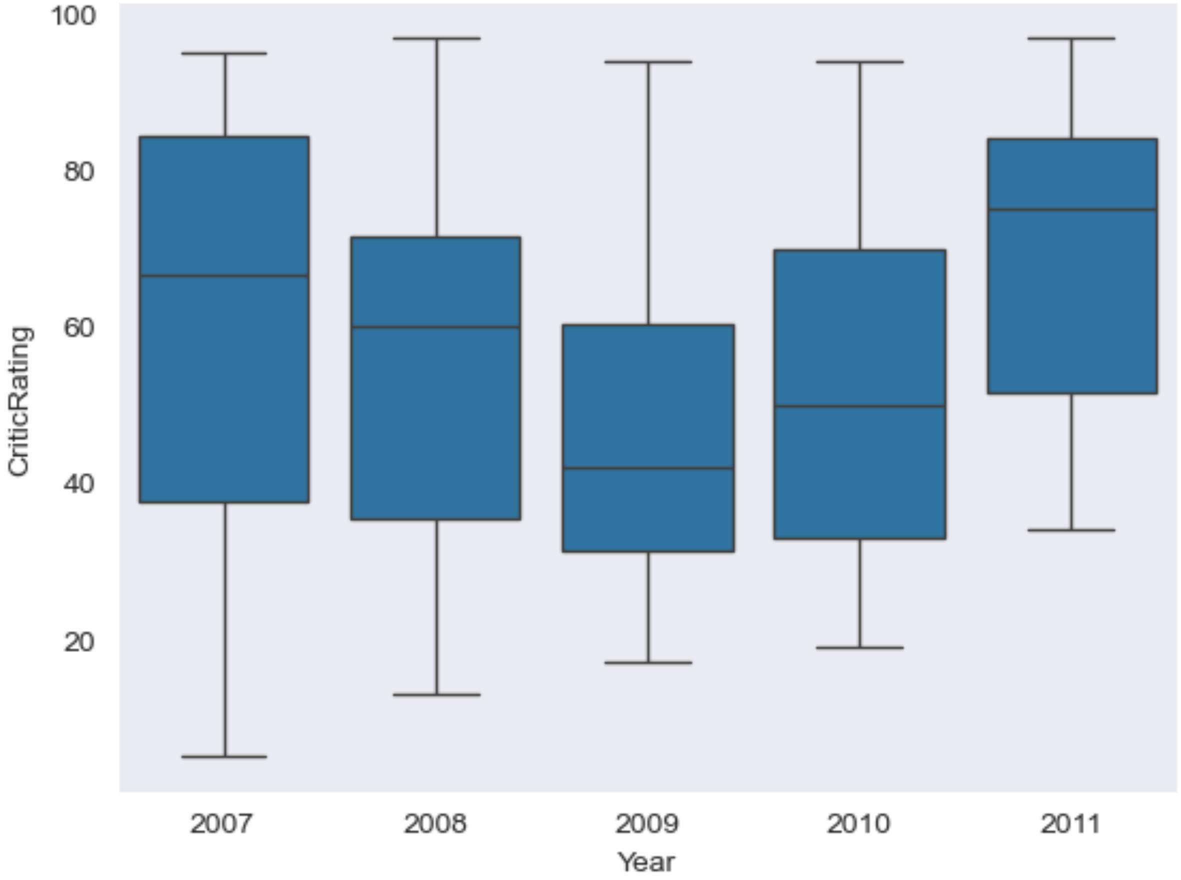
In [60]:
```
w = sns.boxplot(data = movies , x = 'Genre' , y = 'CriticRating')
```



In [61]:
```
z = sns.violinplot(data=movies, x='Genre', y = 'CriticRating')
```

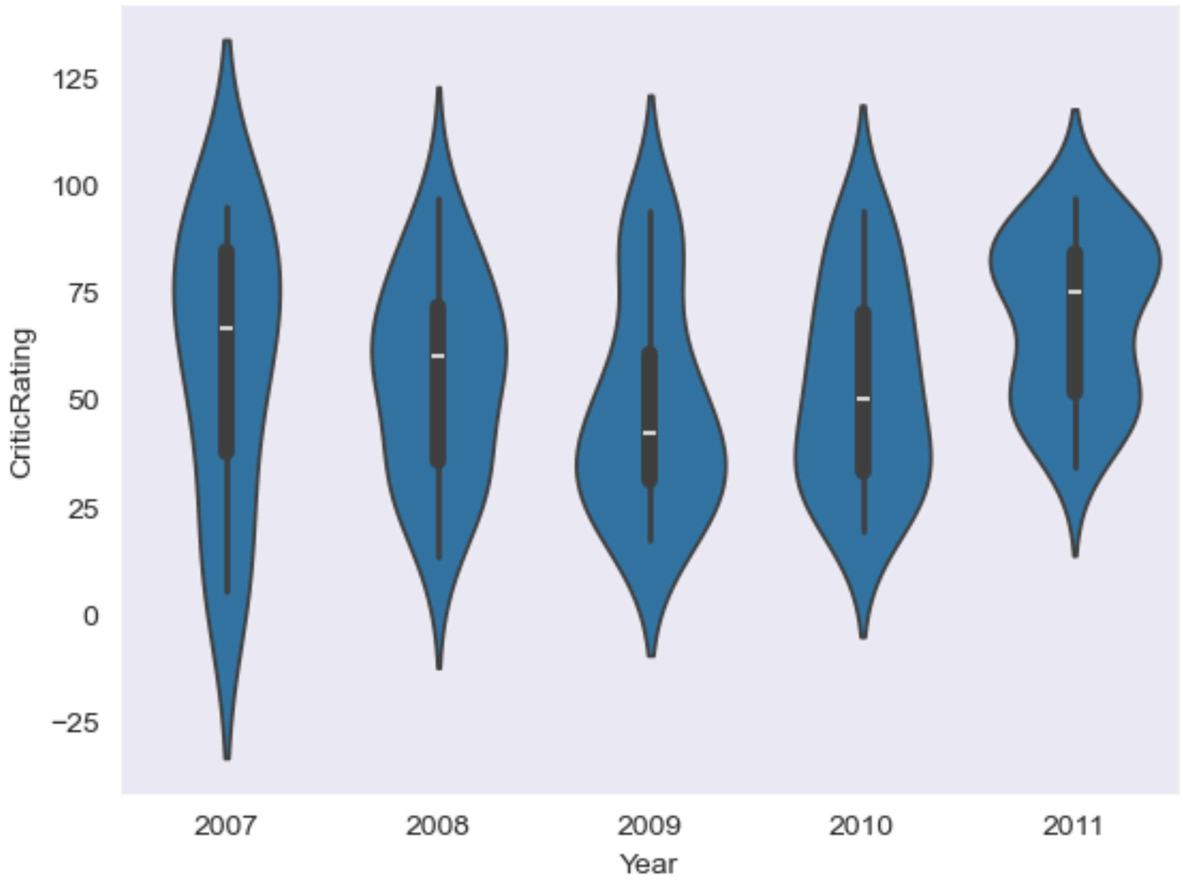


In [62]:
```
w1 = sns.boxplot(data = movies[movies.Genre == 'Drama'] , x = 'Year' , y
```



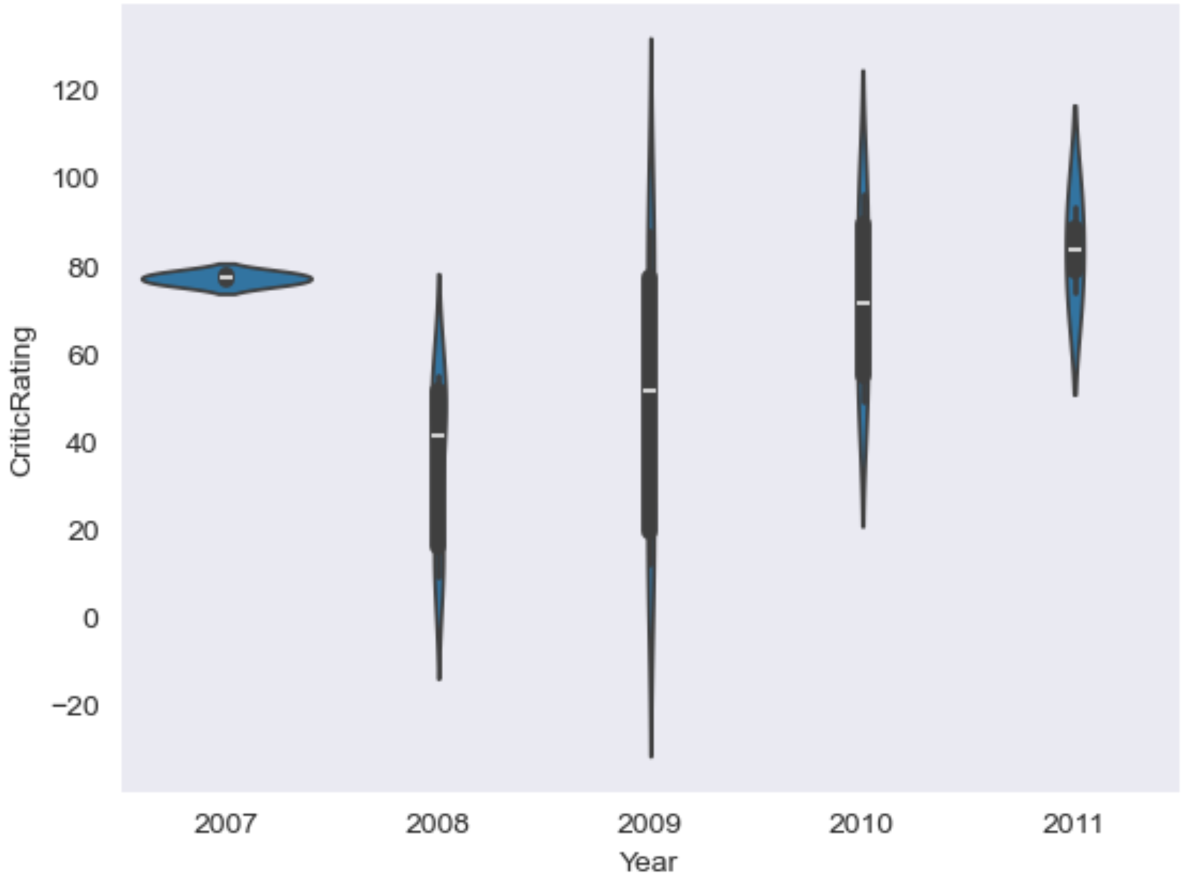
In [63]:

```
z1 = sns.violinplot(data = movies[movies.Genre == 'Drama'] , x = 'Year' ,
```

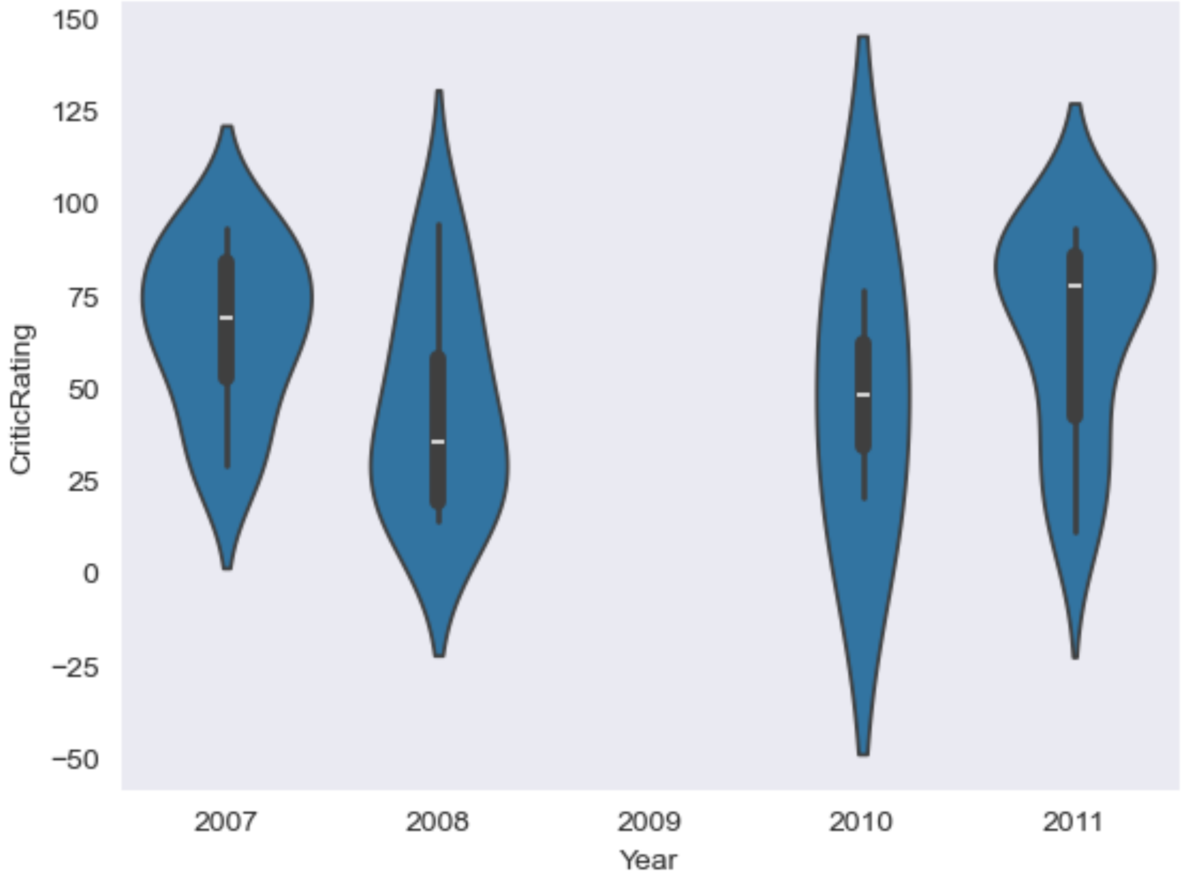


In [64]:

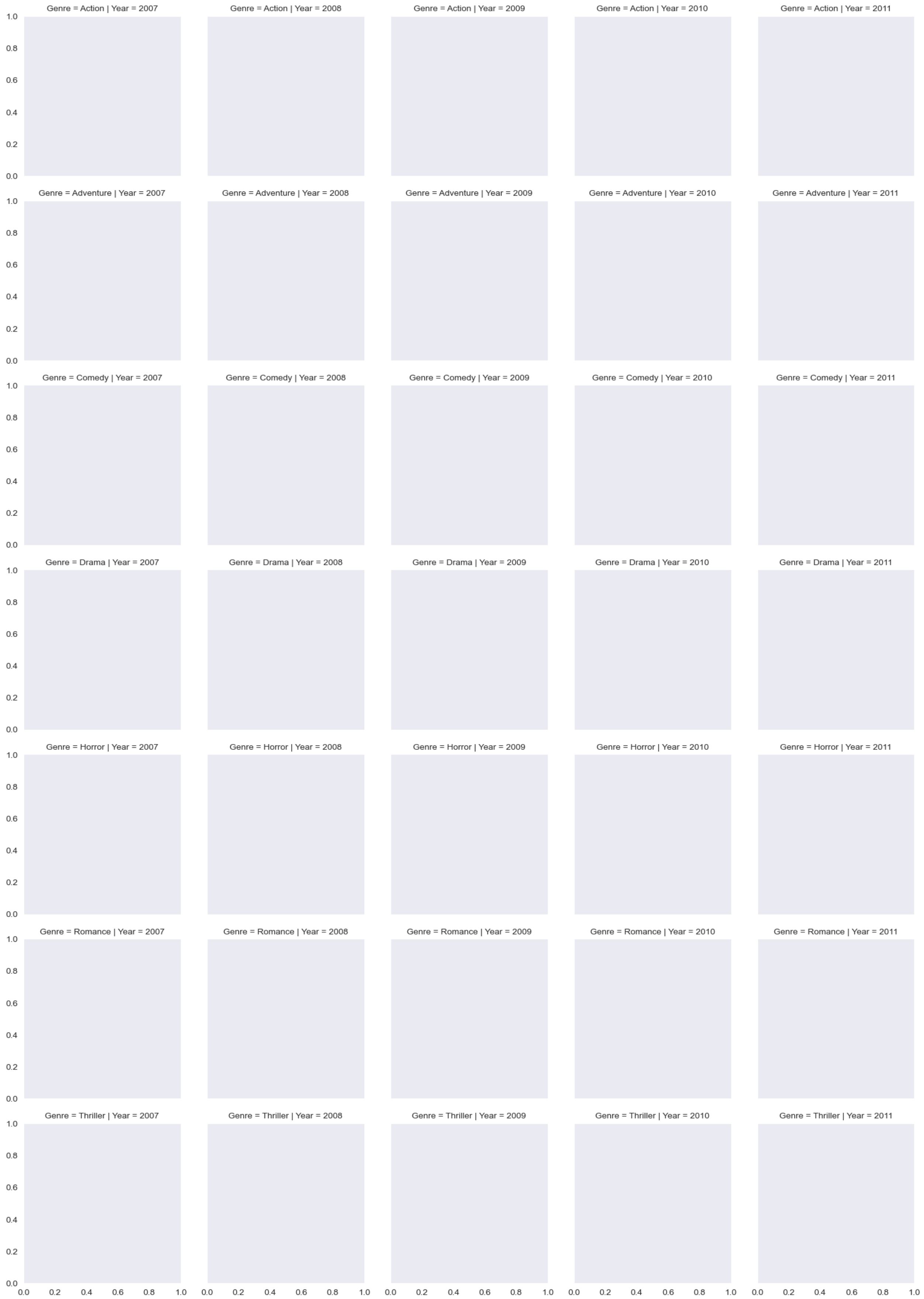
```
z2 = sns.violinplot(data = movies[movies.Genre == 'Adventure'] , x = 'Yea
```



In [65]:
```
z1 = sns.violinplot(data = movies[movies.Genre == 'Thriller'] , x = 'Year
```

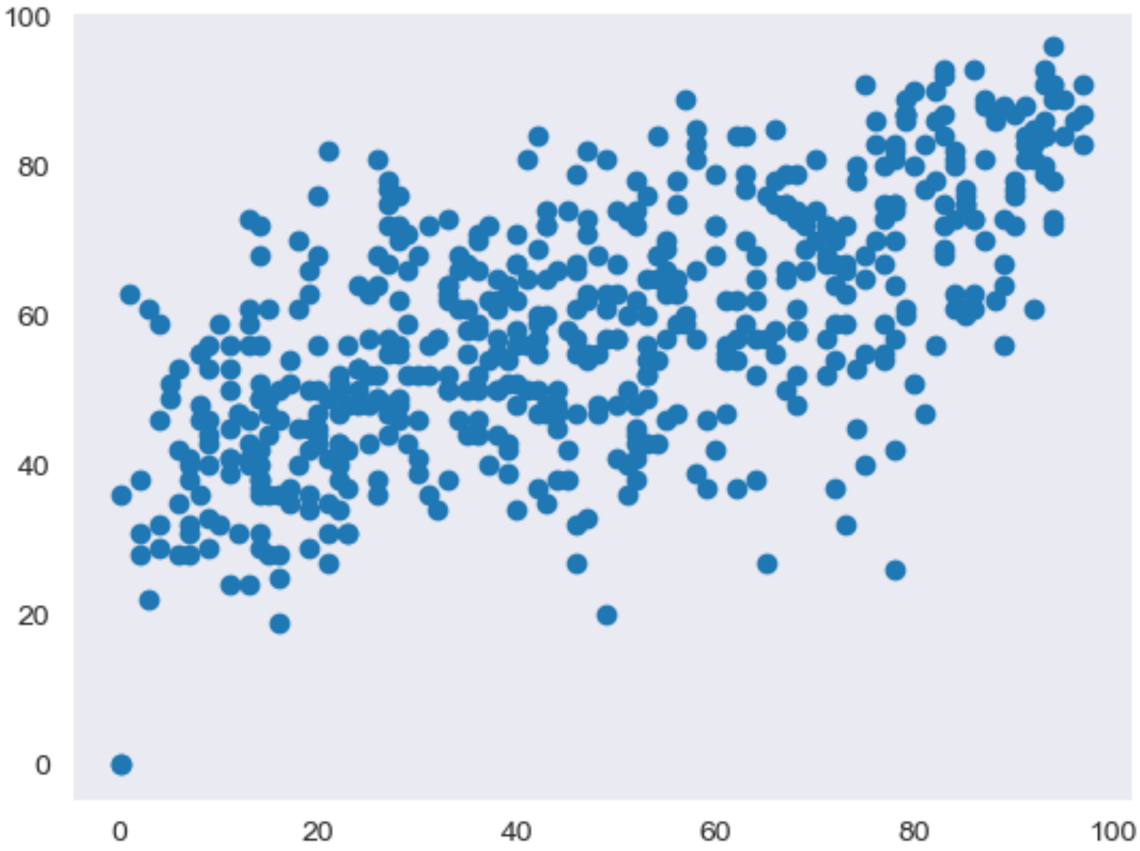


In [66]:
```
g = sns.FacetGrid  (movies ,  row = 'Genre' , col = 'Year' , hue = 'Genre
```

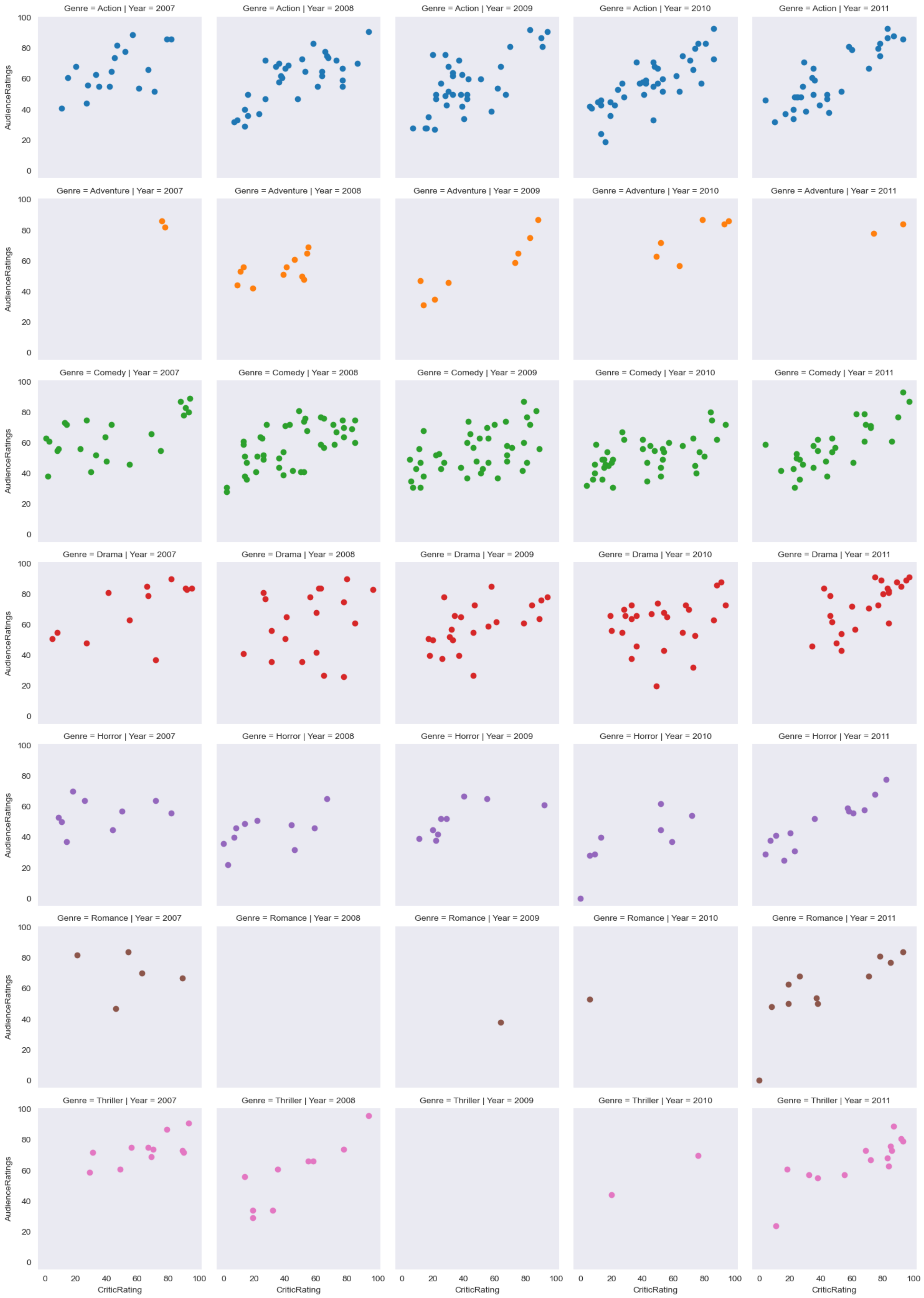


In [67]:
```
plt.scatter(movies.CriticRating,movies.AudienceRatings)
```

```
Out[67]: <matplotlib.collections.PathCollection at 0x131bd1f40>
```

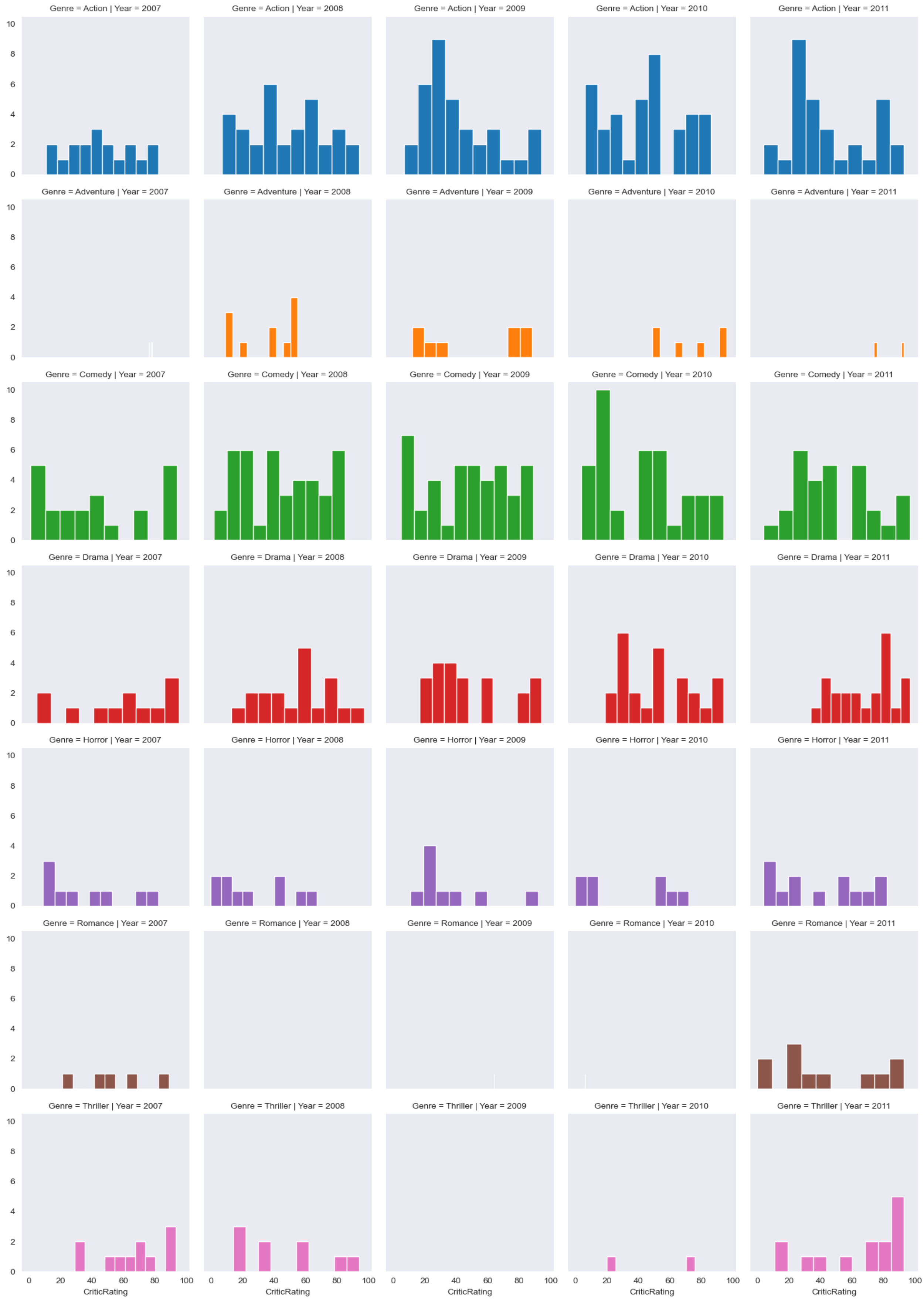


```
In [68]: g = sns.FacetGrid  (movies ,  row = 'Genre' , col = 'Year' , hue = 'Genre
         g = g.map(plt.scatter,'CriticRating' , 'AudienceRatings')
```



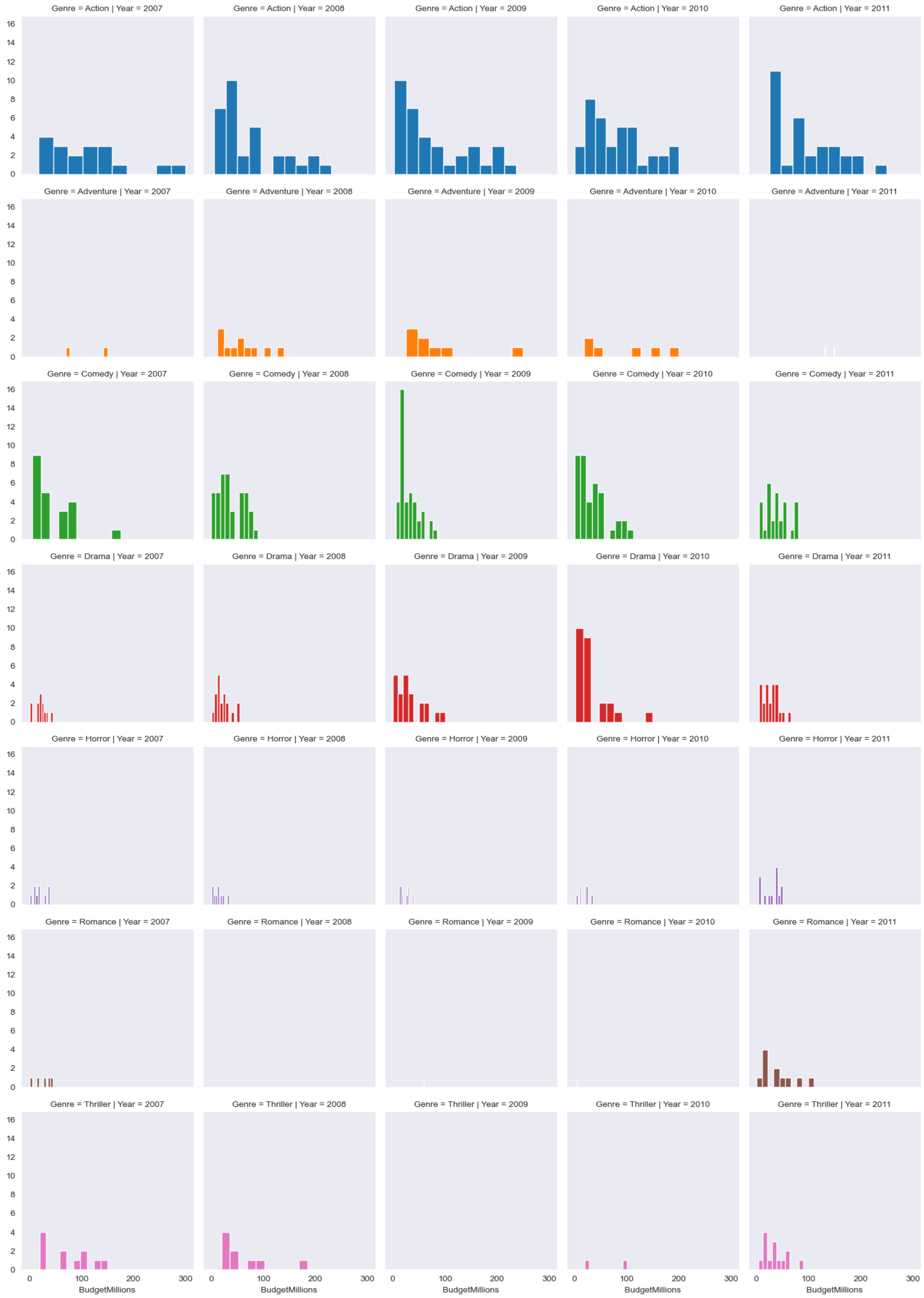
In [69]:

```
g = sns.FacetGrid  (movies ,  row = 'Genre' , col = 'Year' , hue = 'Genre
g = g.map(plt.hist,'CriticRating' )
```



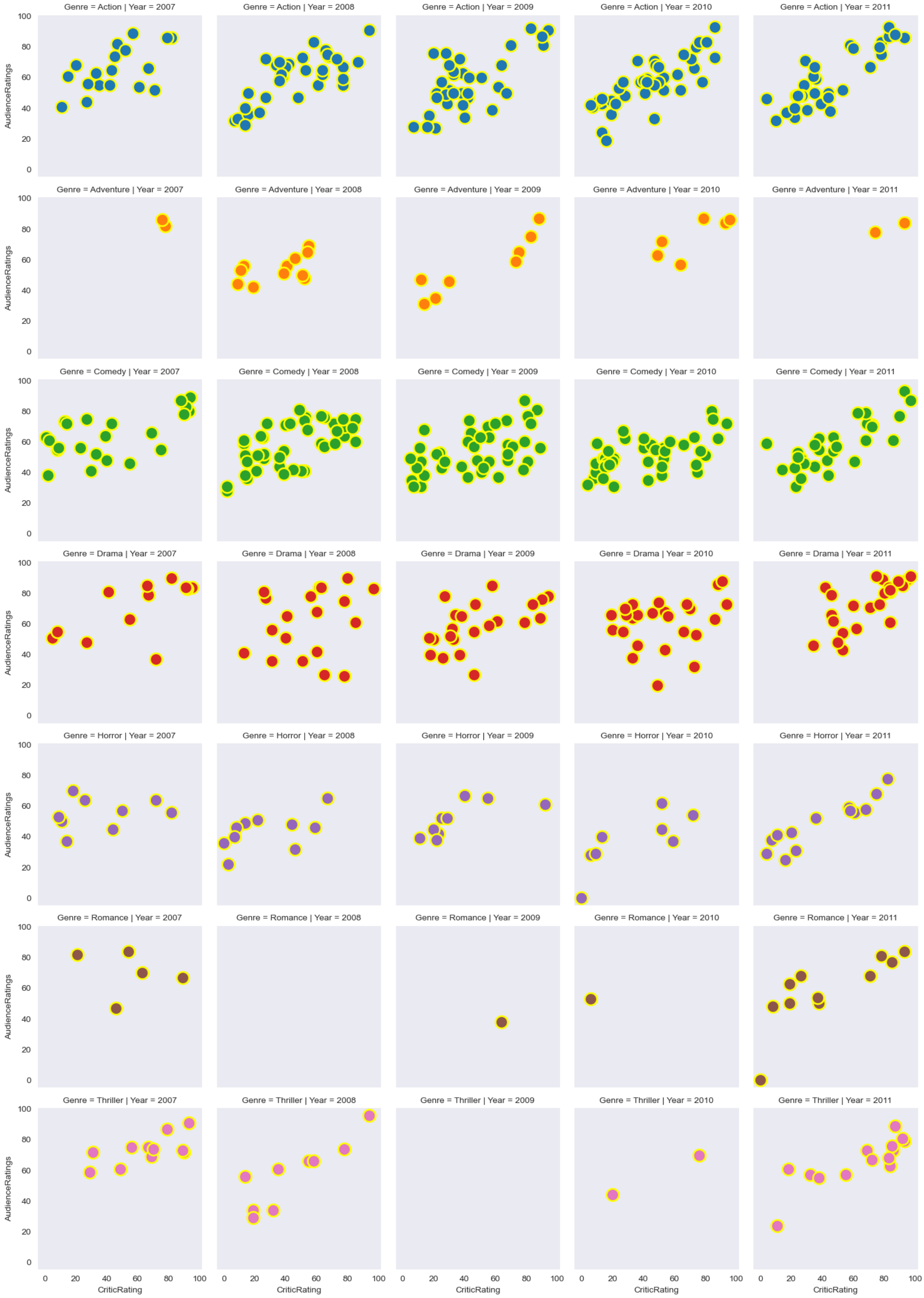
In [70]:

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



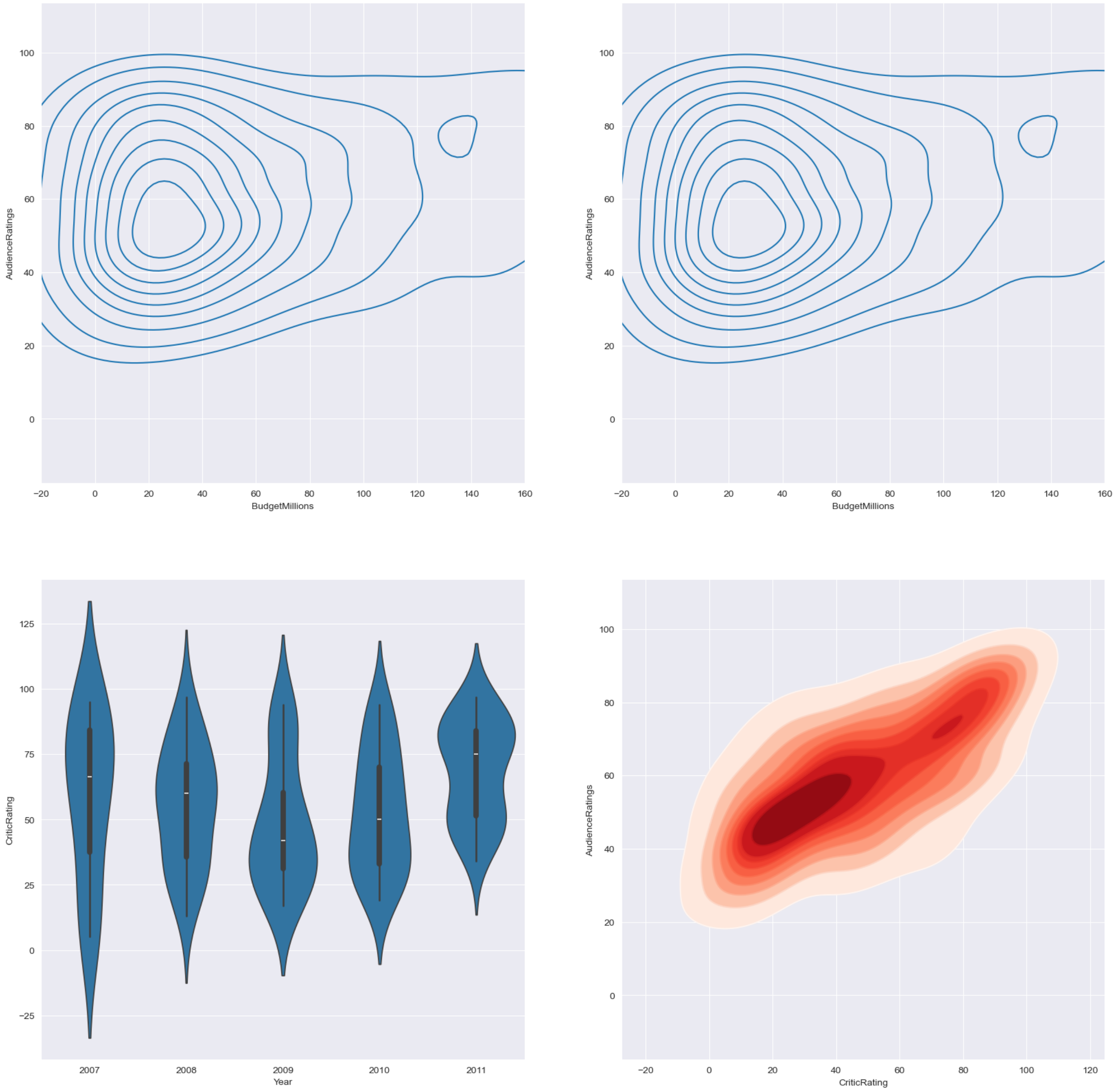
In [71]:

```
g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre')
kws = dict(s = 200,linewidth = 2 , edgecolor ='yellow')
g = g.map(plt.scatter, 'CriticRating' , 'AudienceRatings',**kws)
```



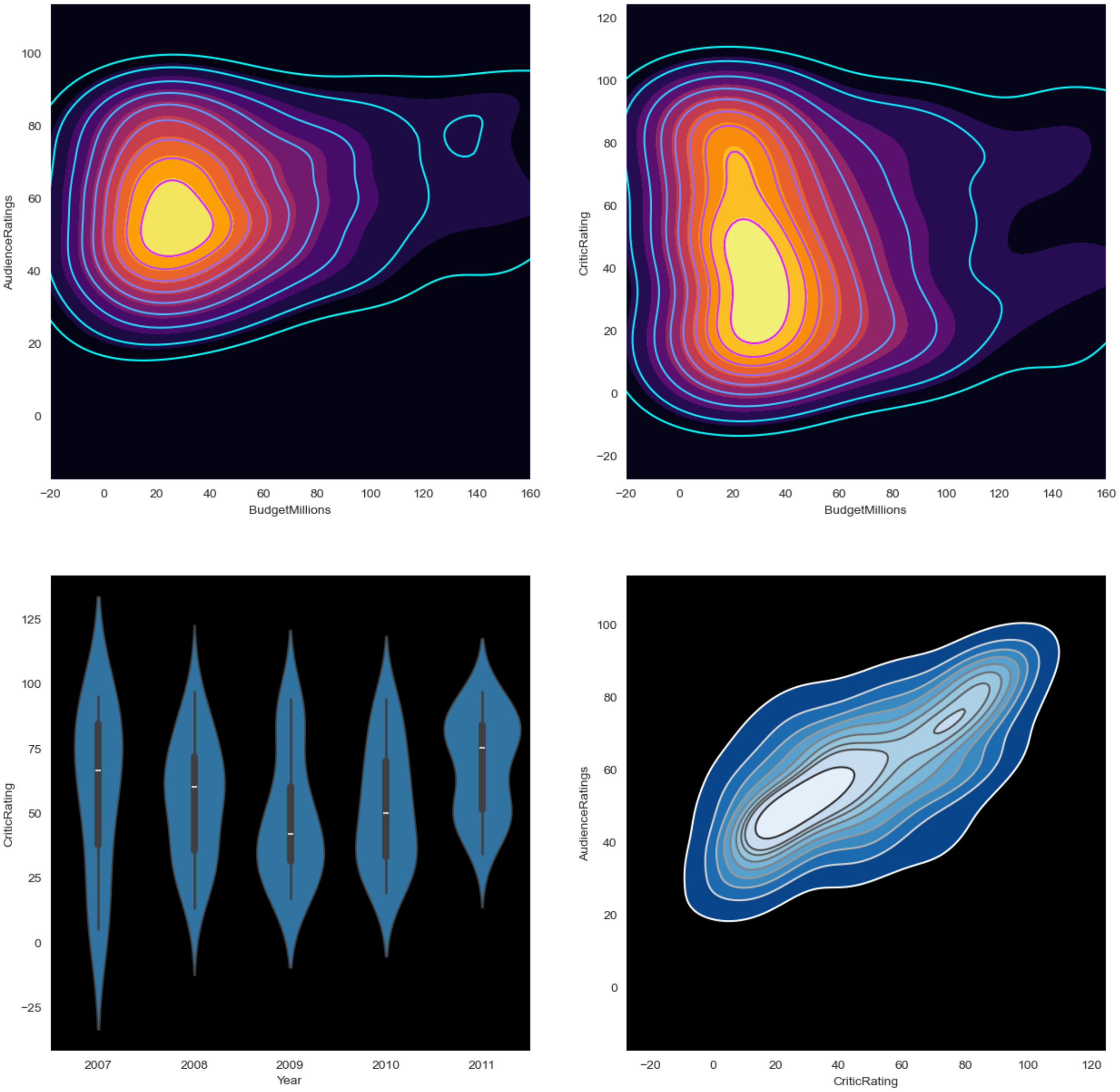
In [178…

```
sns.set_style('darkgrid')
f,axes = plt.subplots(2,2 , figsize = (20,20))
k1 = sns.kdeplot(data = movies, x ='BudgetMillions',y = 'AudienceRatings'
k2 = sns.kdeplot(data = movies, x ='BudgetMillions',y = 'AudienceRatings'
k1.set(xlim = (-20 , 160))
k2.set(xlim = (-20 , 160))
z = sns.violinplot(data=movies[movies.Genre=='Drama'], x='Year', y = 'Cri
k4 = sns.kdeplot(data = movies, x = 'CriticRating' , y ='AudienceRatings'
k4b = sns.kdeplot(data = movies, x ='CriticRating', y  = 'AudienceRatings
plt.show()
```



In [159…

```
sns.set_style('dark',{'axes.facecolor' : 'black'})
f, axes = plt.subplots (2,2, figsize = (15,15))
#plot[0,0]

k1 = sns.kdeplot(data = movies , x = 'BudgetMillions' , y = 'AudienceRati
                shade  = True , shade_lowest = True, cmap = 'inferno', \
                ax = axes[0,0])

k1b = sns.kdeplot(data = movies , x = 'BudgetMillions' , y = 'AudienceRat
                 cmap = 'cool', ax = axes[0,0])

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

z = sns.violinplot(data=movies[movies.Genre=='Drama'], \
                   x='Year', y = 'CriticRating', ax=axes[1,0])
k4 = sns.kdeplot(data = movies , x = 'CriticRating' , y = 'AudienceRating
                shade  = True , shade_lowest = False, cmap = 'Blues_r', \
                ax = axes[1,1])
k4b = sns.kdeplot(data = movies , x = 'CriticRating' , y = 'AudienceRatin
                cmap = 'gist_gray_r', ax = axes[1,1])

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



In [ ]: