25-11-2025 ADVANCED VISUALIZATION MOVIE RATINGS

In [1]:

```
import pandas as pd
```

In [2]:

```
movies = pd.read_csv(r"C:\Users\karthik reddy\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]:

```
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
```

Out[14]:

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

In [15]:
```
movies.columns = ['Film', 'Genre', 'CriticRating', 'AudienceRating', 'BudgetMill
```

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

Out[16]:
```
Index(['Film', 'Genre', 'CriticRating', 'AudienceRating', 'BudgetMillions',
       'Year'],
      dtype='object')
```

In [17]:
```
movies.head(1)
```

Out[17]:

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

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

Out[18]:
```
(559, 6)
```

In [20]:
```
movies.describe() # descriptive statistics
```

Out[20]:

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

Out[21]:

	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
AudienceRating	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
Year	559.0	2009.152057	1.362632	2007.0	2008.0	2009.0	2010.0	2011.0

In [22]:
```
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 [24]:
```
movies.Film = movies.Film.astype('category')
```

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

Out[25]:

	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 [26]:
```
movies.Film
```

Out[26]:
```
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 [27]:
```
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   AudienceRating  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 [28]:

```
movies.Genre = movies.Genre.astype('category')
movies.Year = movies.Year.astype('category')
```

In [29]:

```
movies.Genre
movies.Year
```

Out[29]:

```
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 [30]:

```
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 [31]:

```
movies.describe()
```

Out[31]:

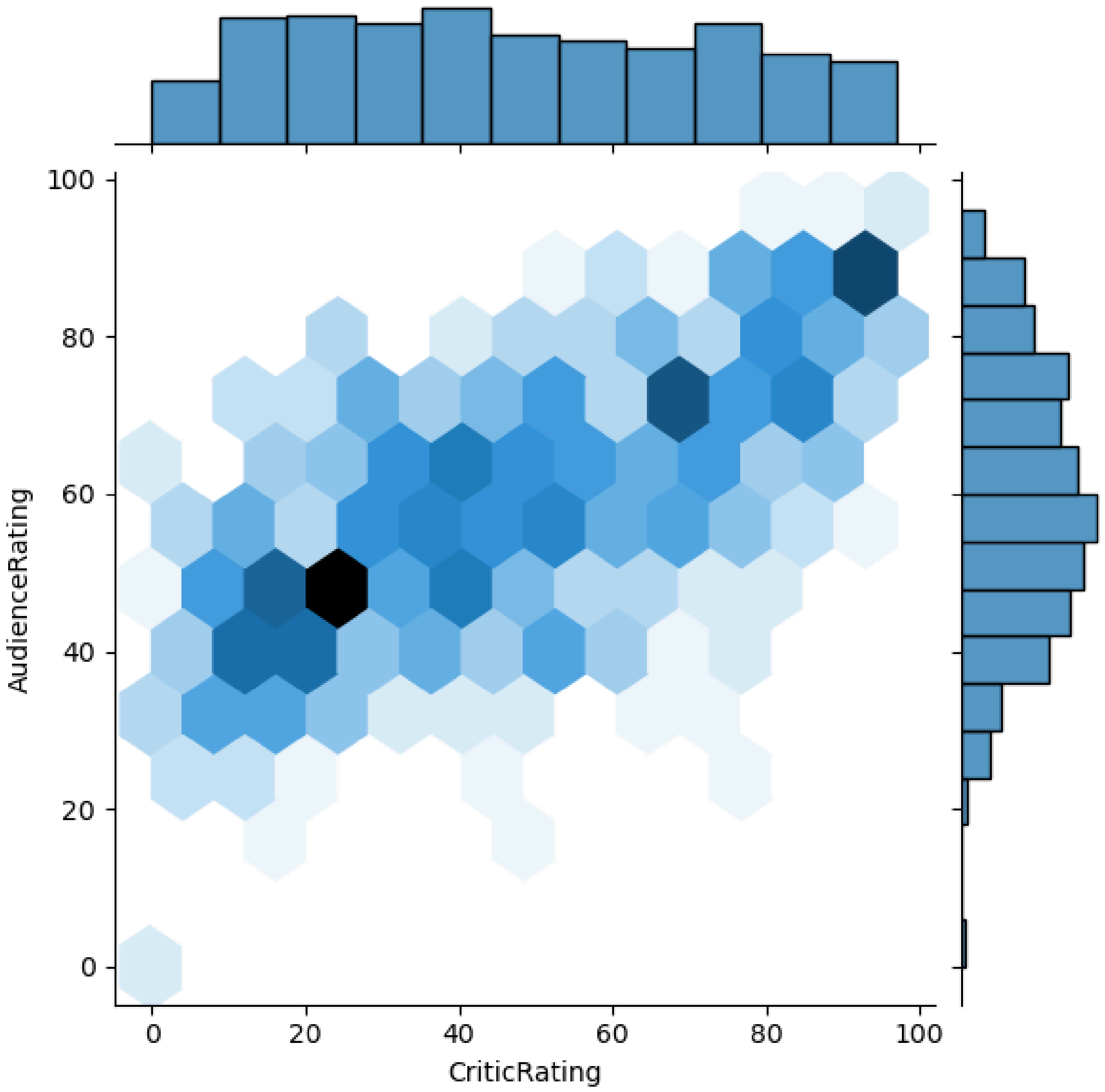
	CriticRating	AudienceRating	BudgetMillions
count	559.000000	559.000000	559.000000
mean	47.309481	58.744186	50.236136
std	26.413091	16.826887	48.731817
min	0.000000	0.000000	0.000000
25%	25.000000	47.000000	20.000000
50%	46.000000	58.000000	35.000000
75%	70.000000	72.000000	65.000000
max	97.000000	96.000000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

In [32]:

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

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

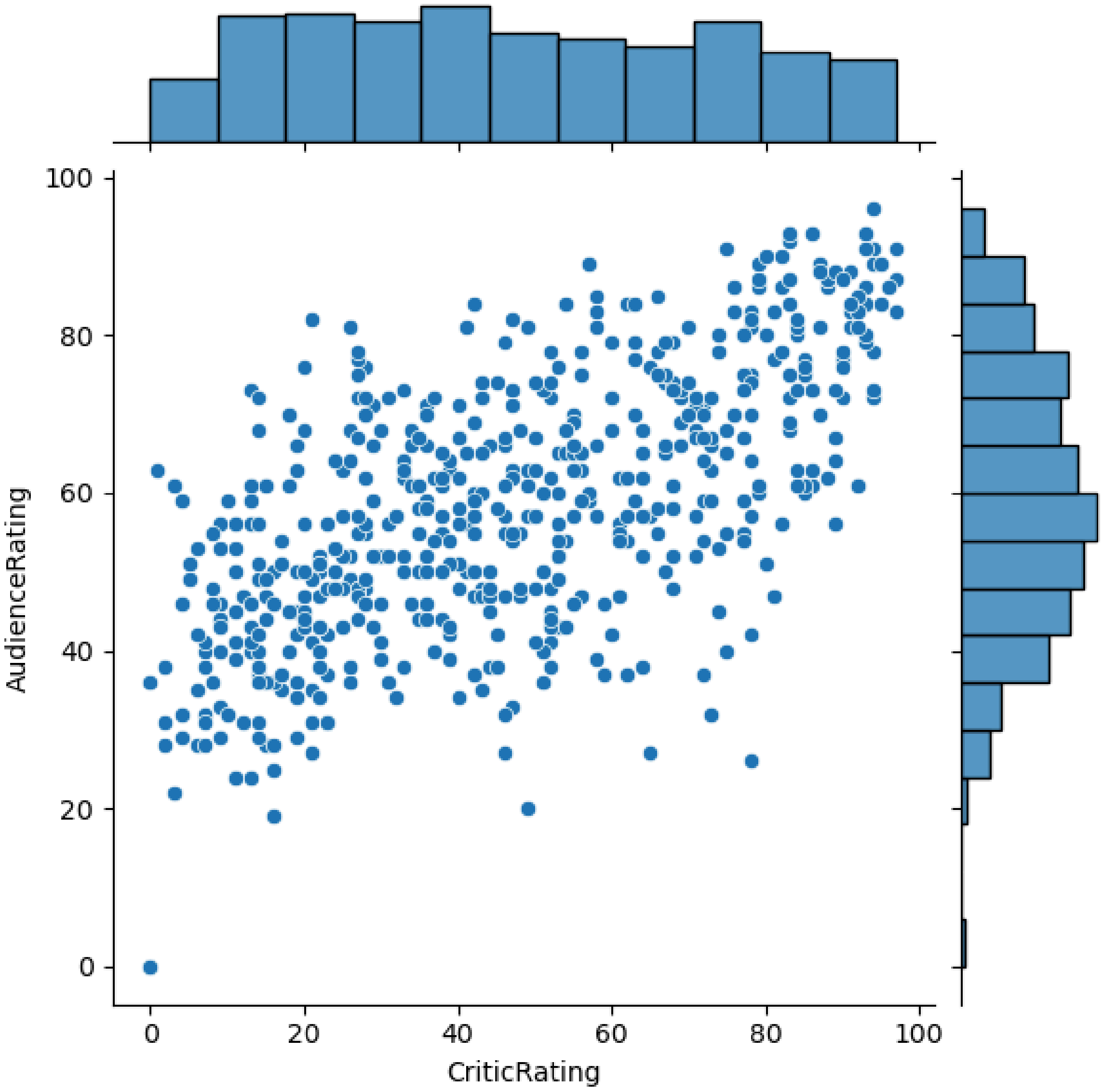
In [34]:

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

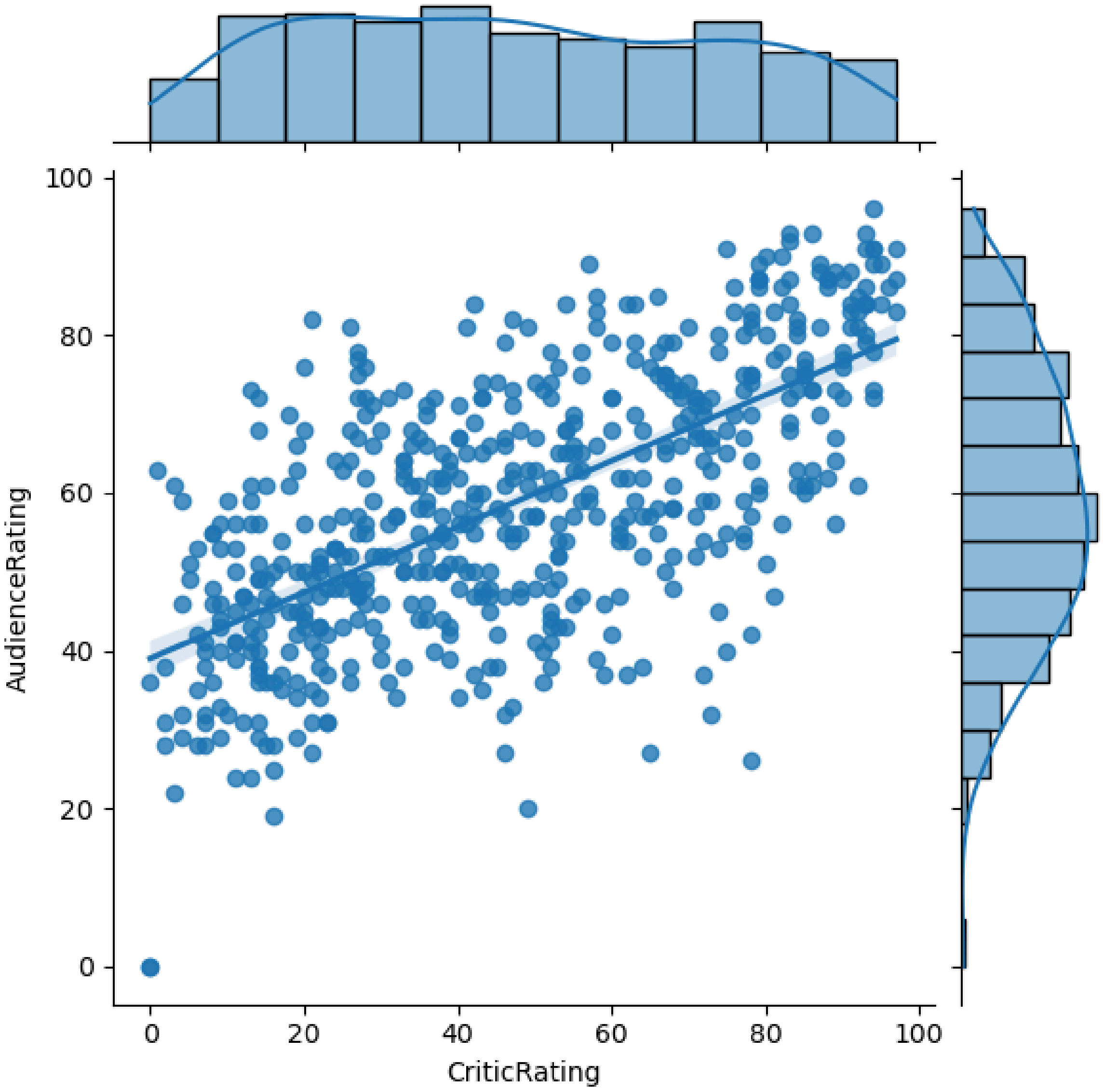


In [35]:

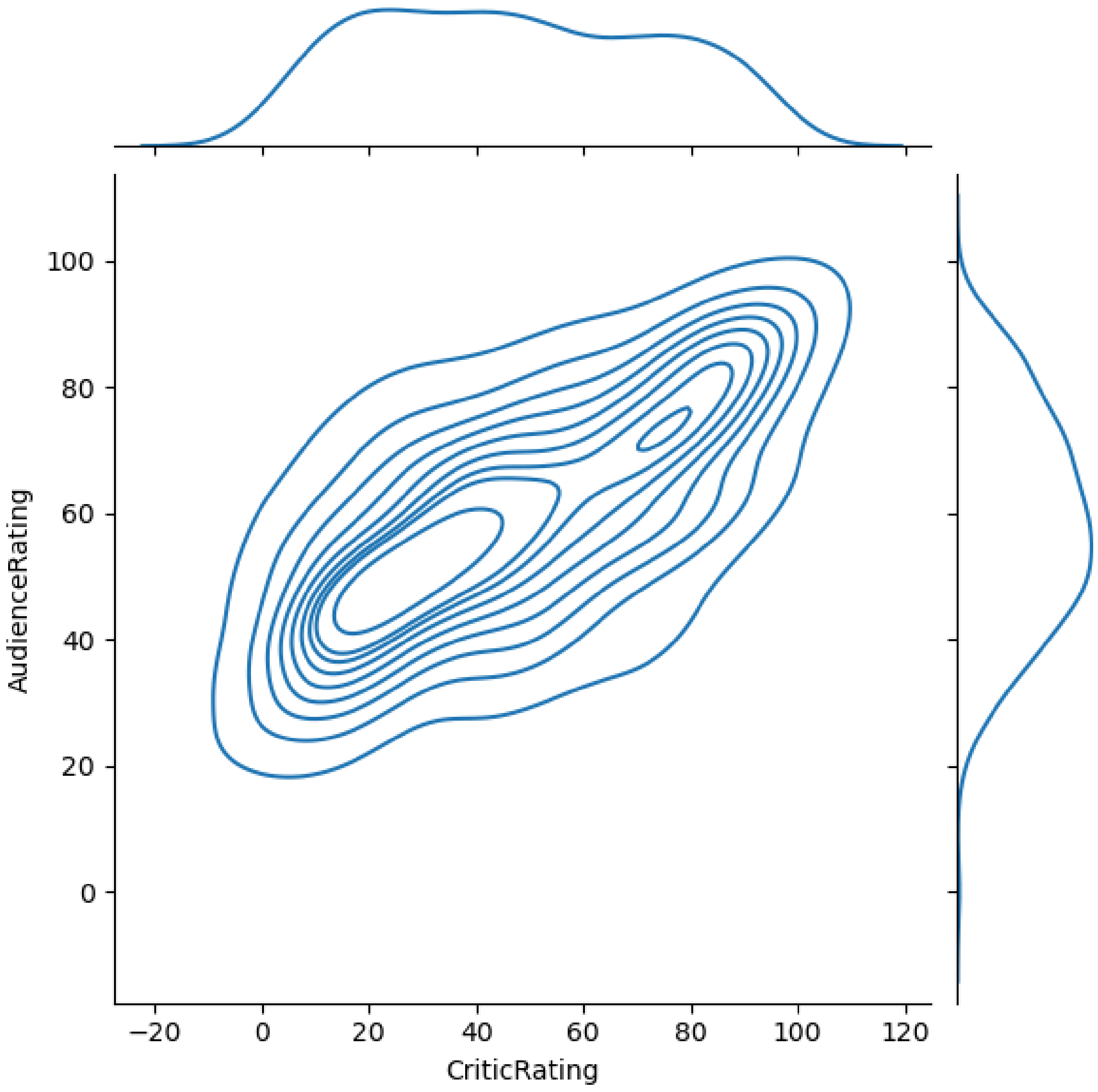
```
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating', kind=
```



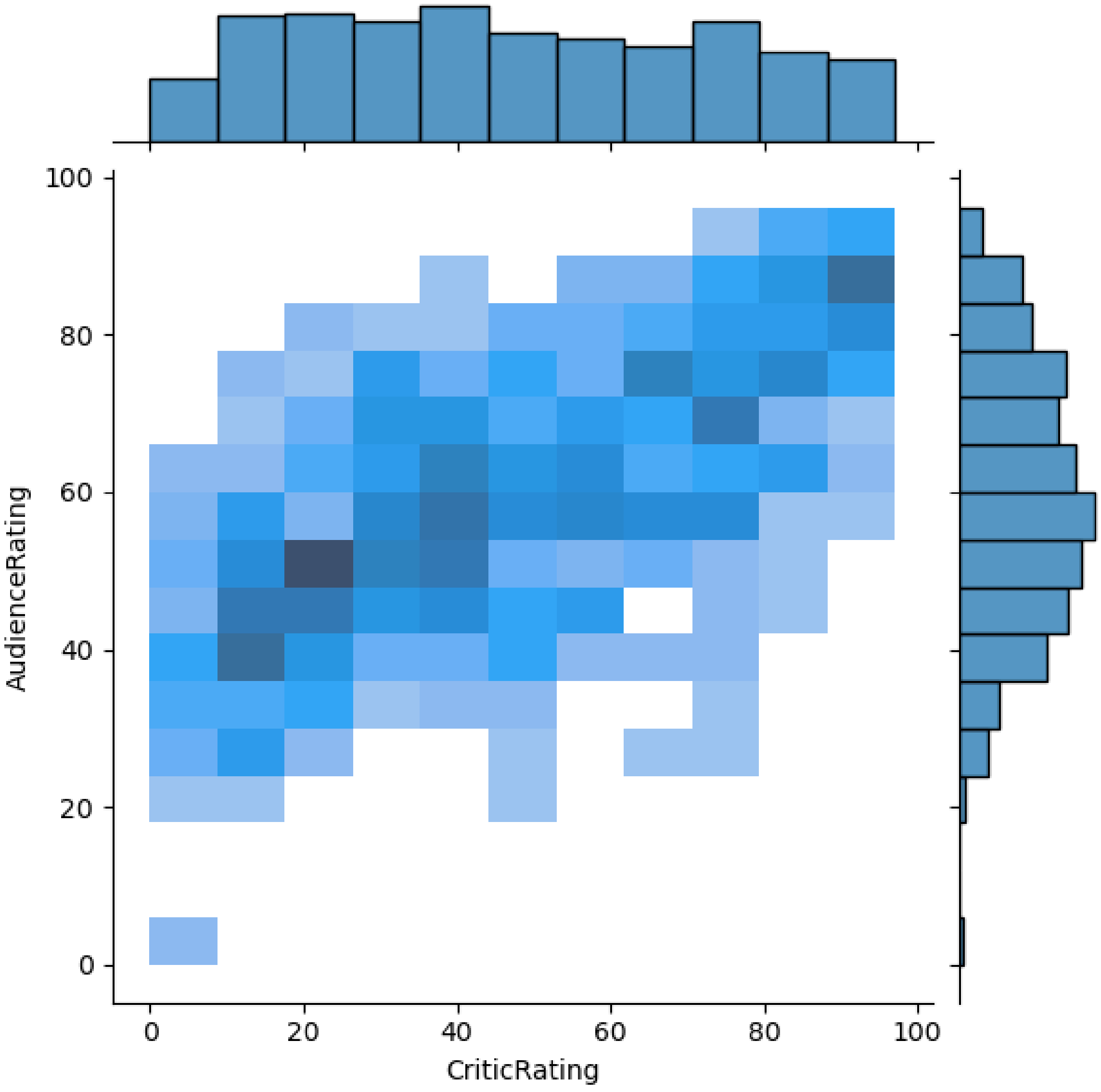
```
In [36]: j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating', kind=
```



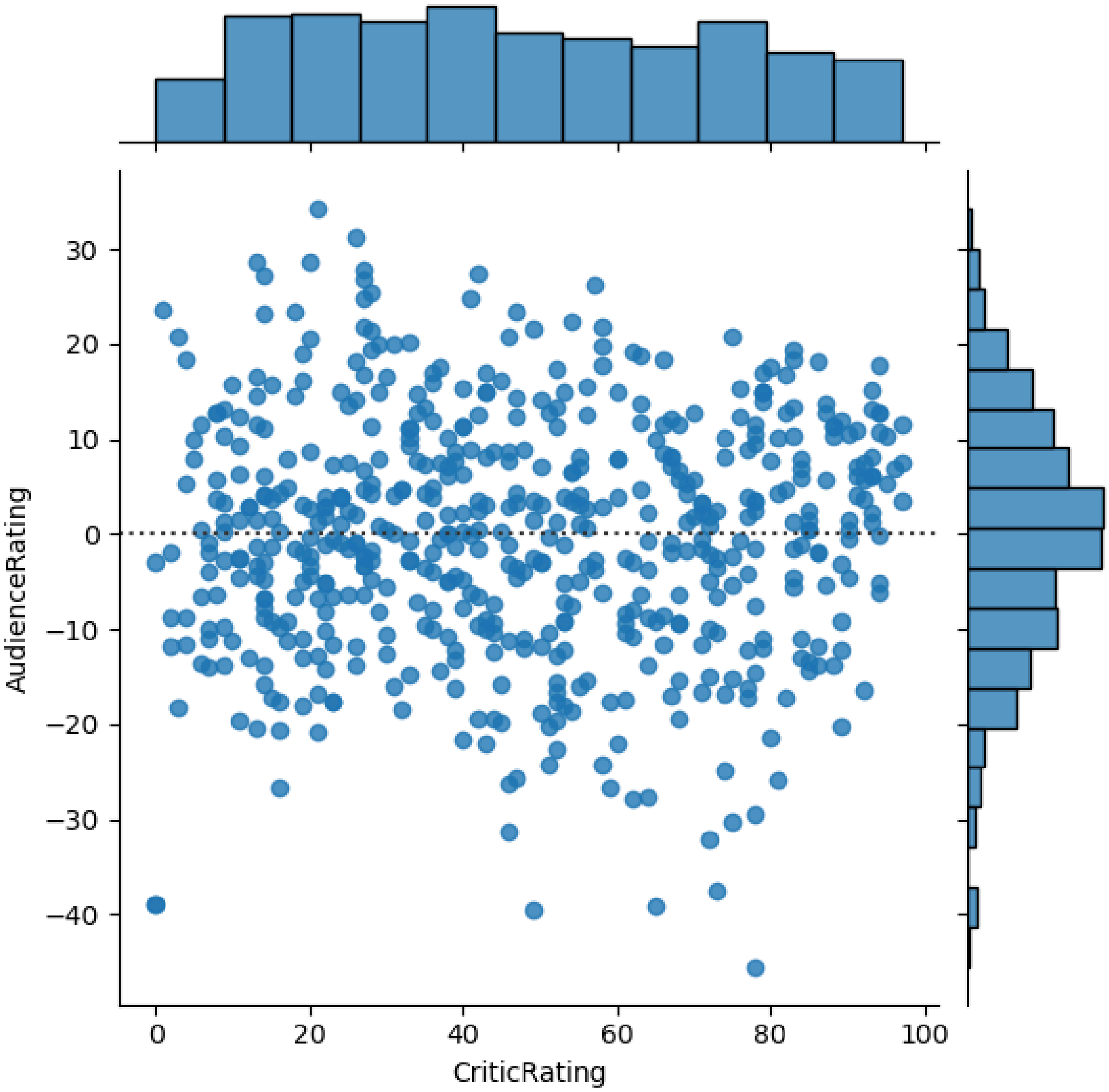
```
In [37]: j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating', kind=
```



```
In [38]: j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating', kind=
```

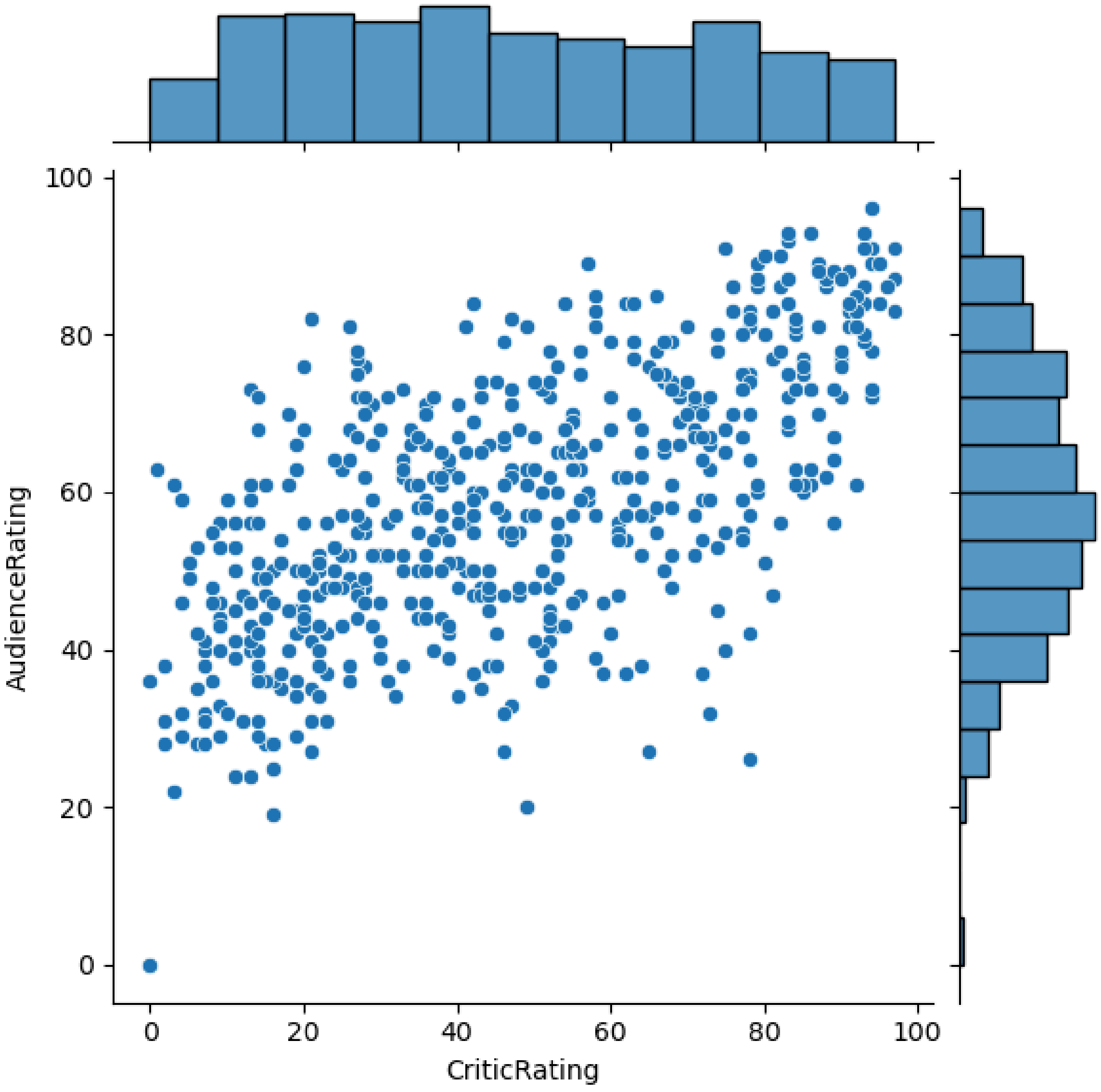


```
In [42]: j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating', kind=
```

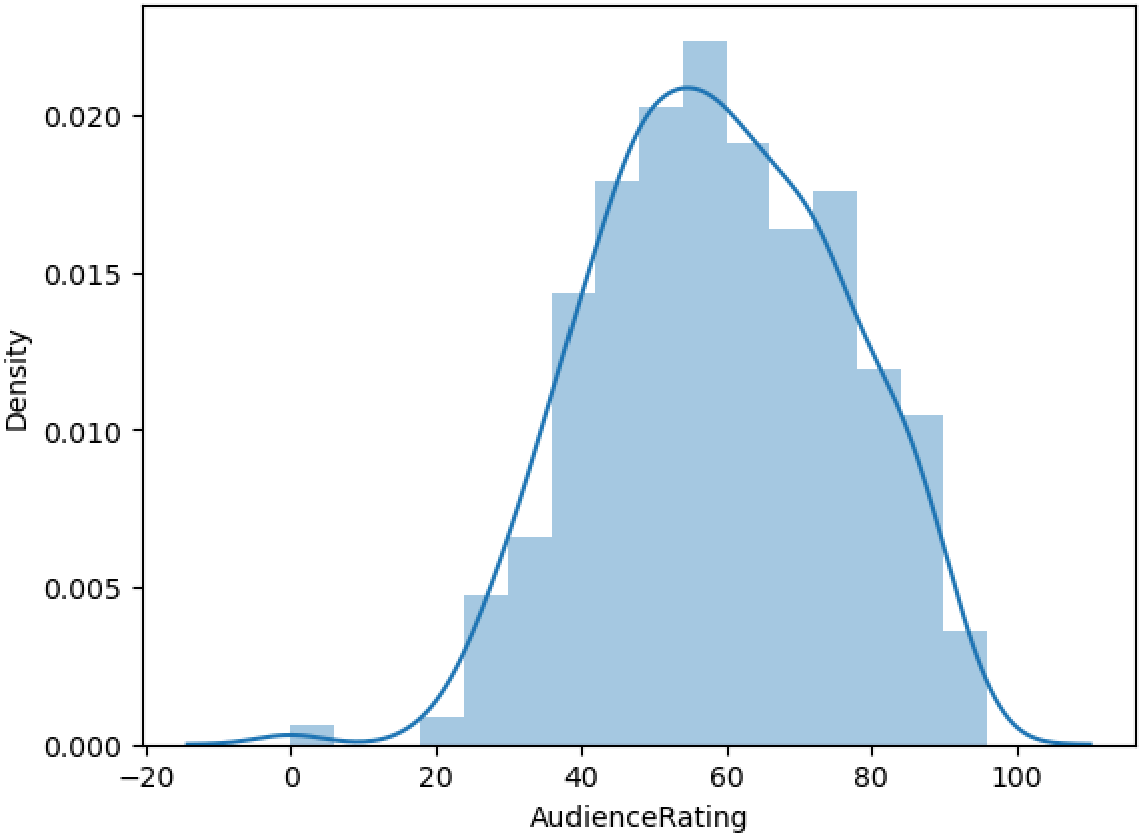


Continuation

```
In [43]: j = sns.jointplot( data = movies, x = 'CriticRating', y = 'AudienceRating')
```

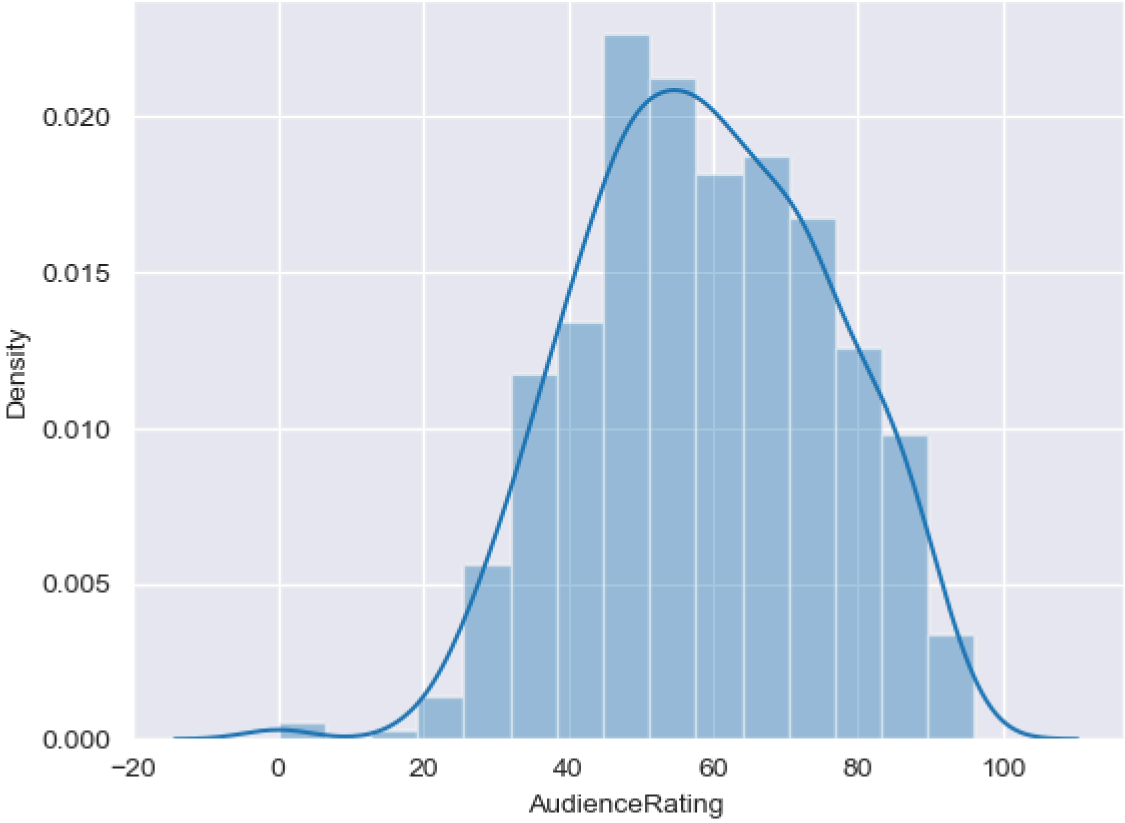


```
In [44]: m1 = sns.distplot(movies.AudienceRating)
```

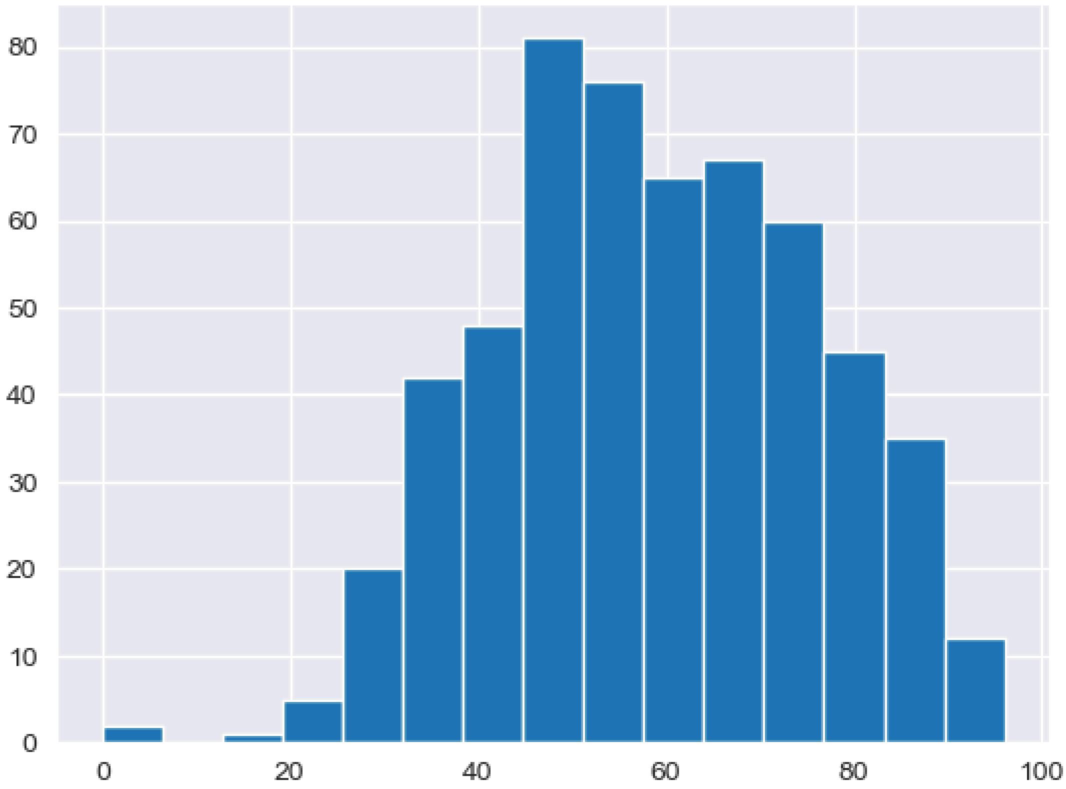


In [45]:
```
sns.set_style('darkgrid')
```

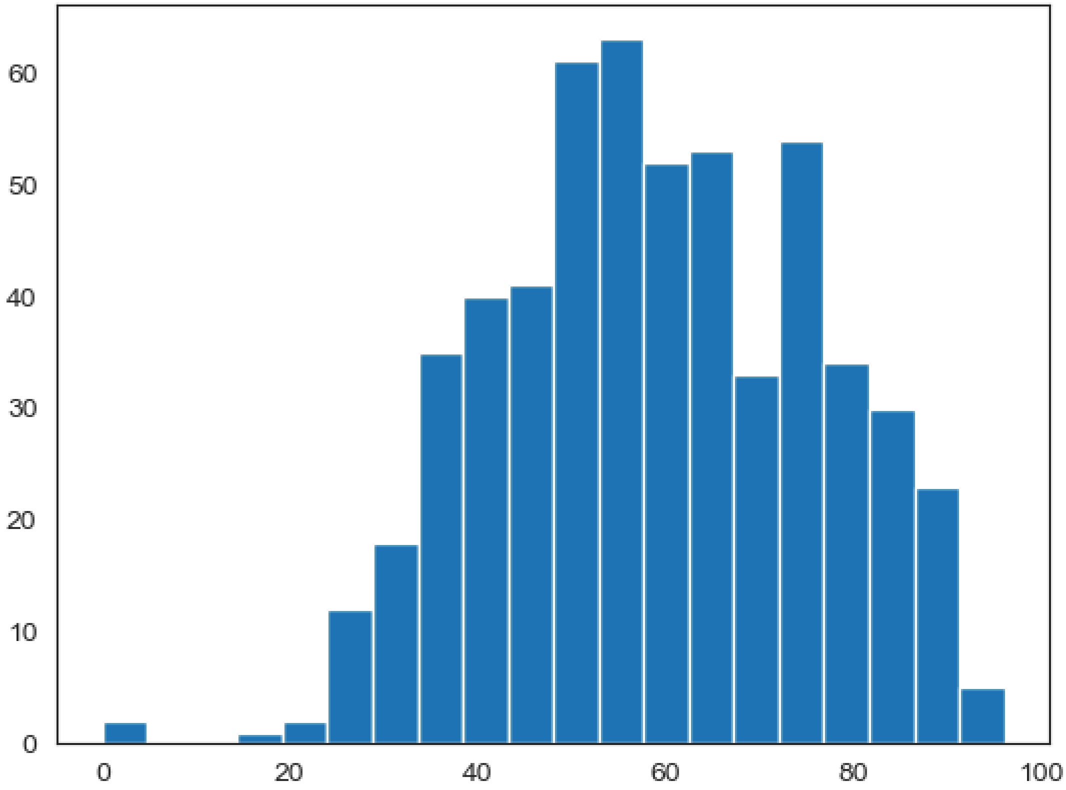
In [46]:
```
m2 = sns.distplot(movies.AudienceRating, bins = 15)
```



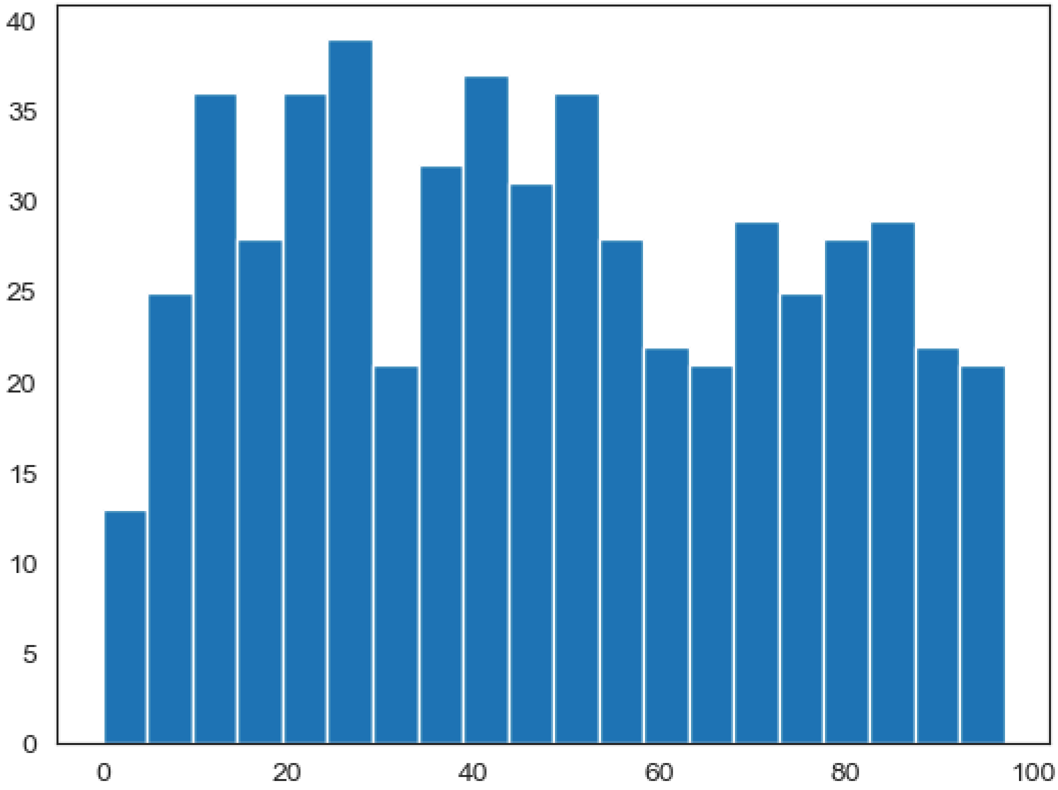
In [47]:
```
n1 = plt.hist(movies.AudienceRating, bins=15)
```



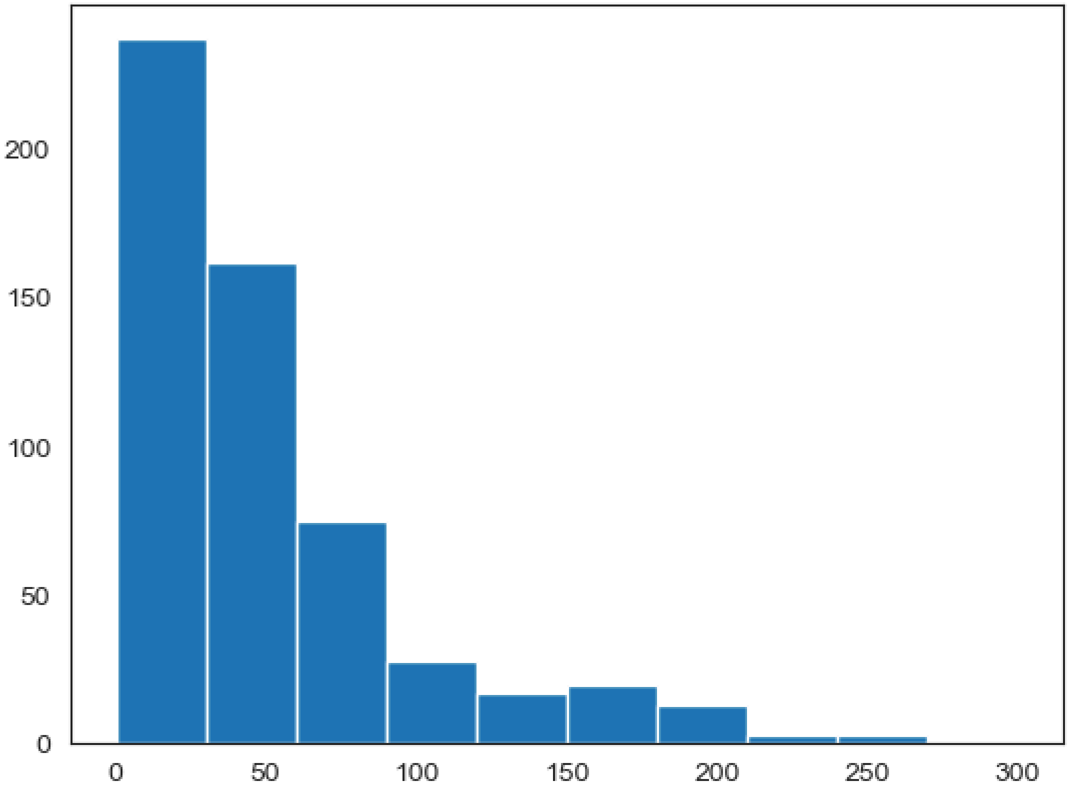
In [48]:
```
sns.set_style('white')
n1 = plt.hist(movies.AudienceRating, bins=20)
```



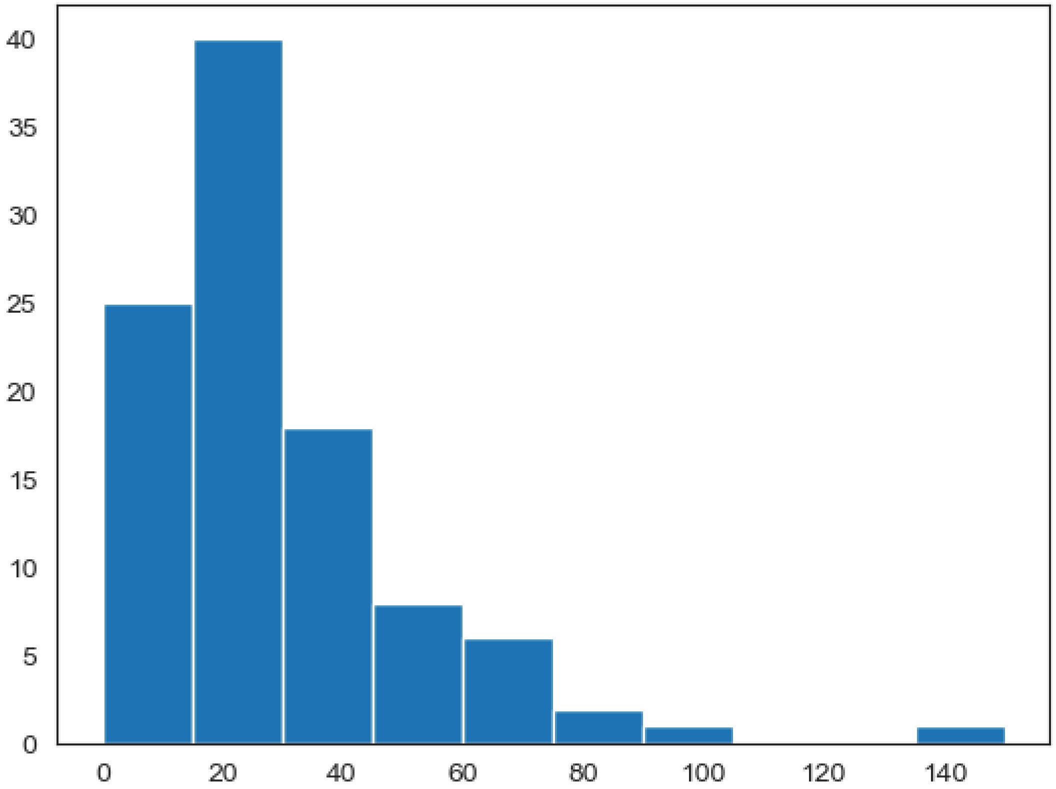
In [49]:
```
n1 = plt.hist(movies.CriticRating, bins=20)
```



In [50]:
```
plt.hist(movies.BudgetMillions)
plt.show()
```



In [51]:
```
plt.hist(movies[movies.Genre == 'Drama'].BudgetMillions)
plt.show()
```



In [52]:

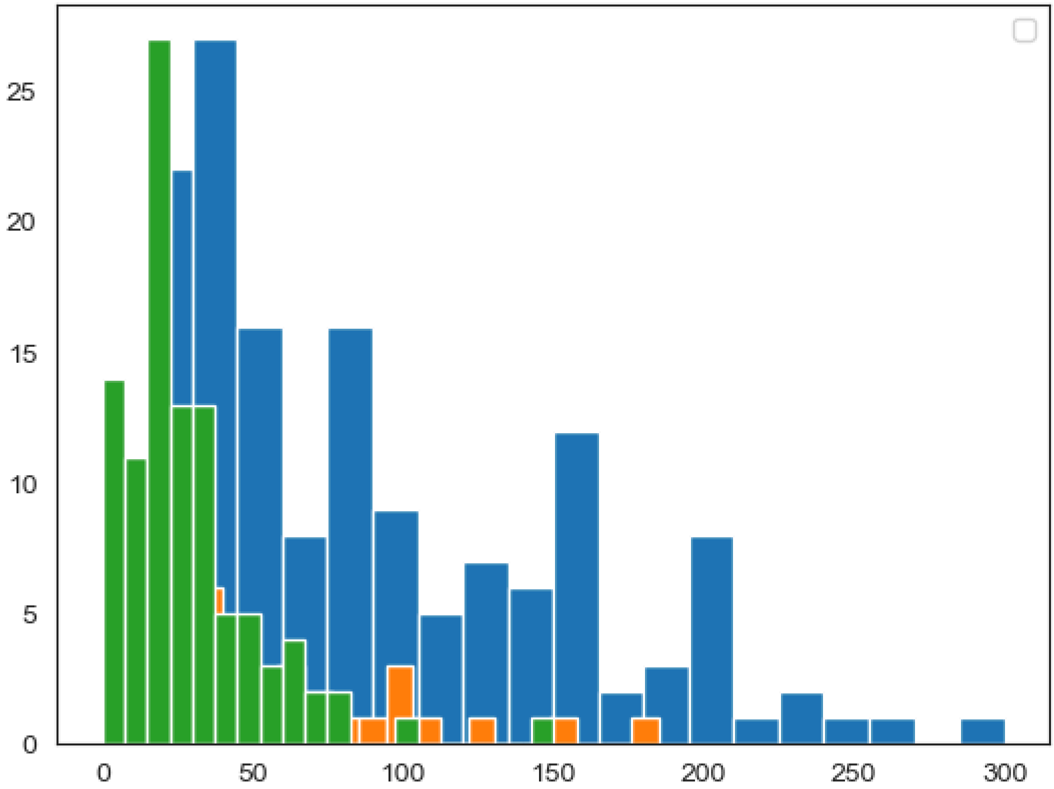
```
movies.head()
```

Out[52]:

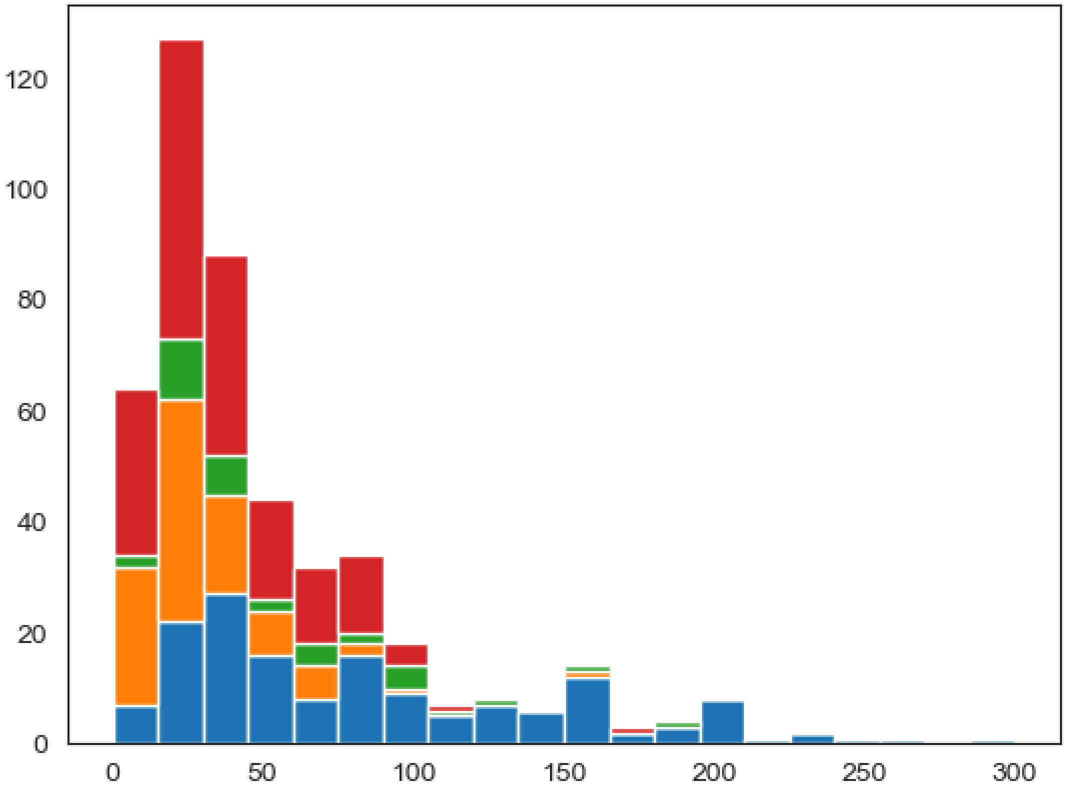
	Film	Genre	CriticRating	AudienceRating	BudgetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009

In [53]:

```
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()
```



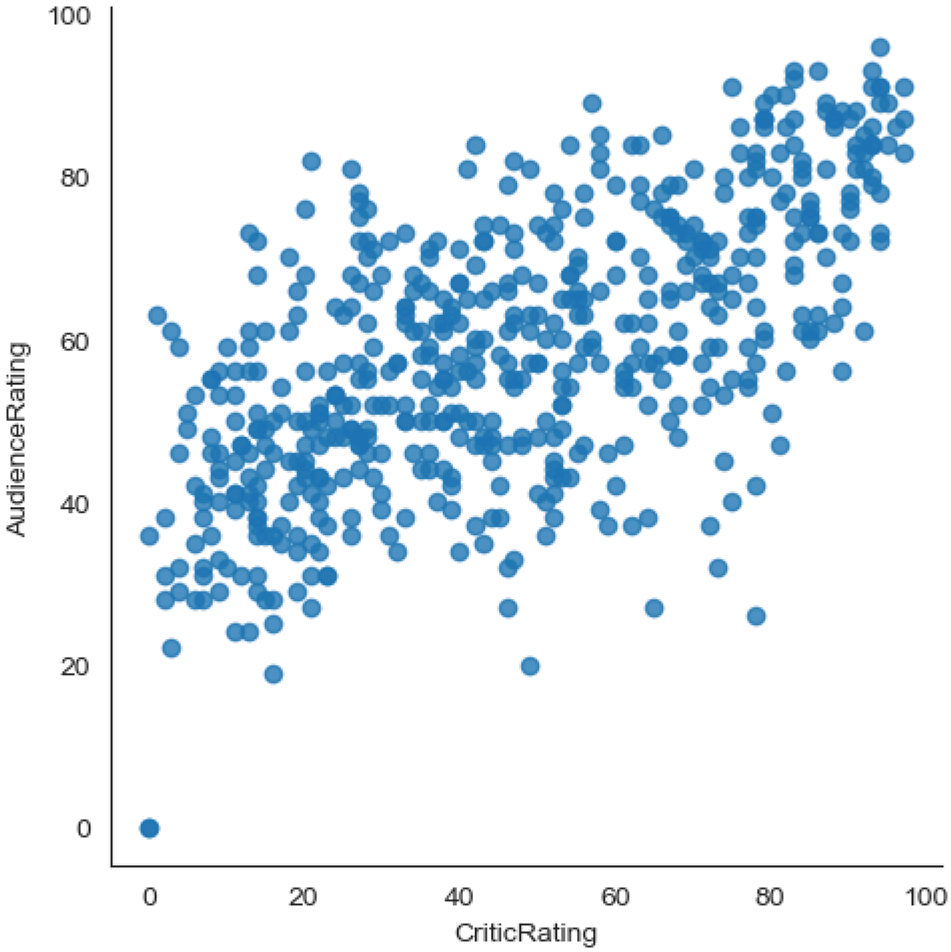
In [54]:
```
plt.hist([movies[movies.Genre == 'Action'].BudgetMillions,\
          movies[movies.Genre == 'Drama'].BudgetMillions, \
          movies[movies.Genre == 'Thriller'].BudgetMillions, \
          movies[movies.Genre == 'Comedy'].BudgetMillions],
         bins = 20, stacked = True)
plt.show()
```



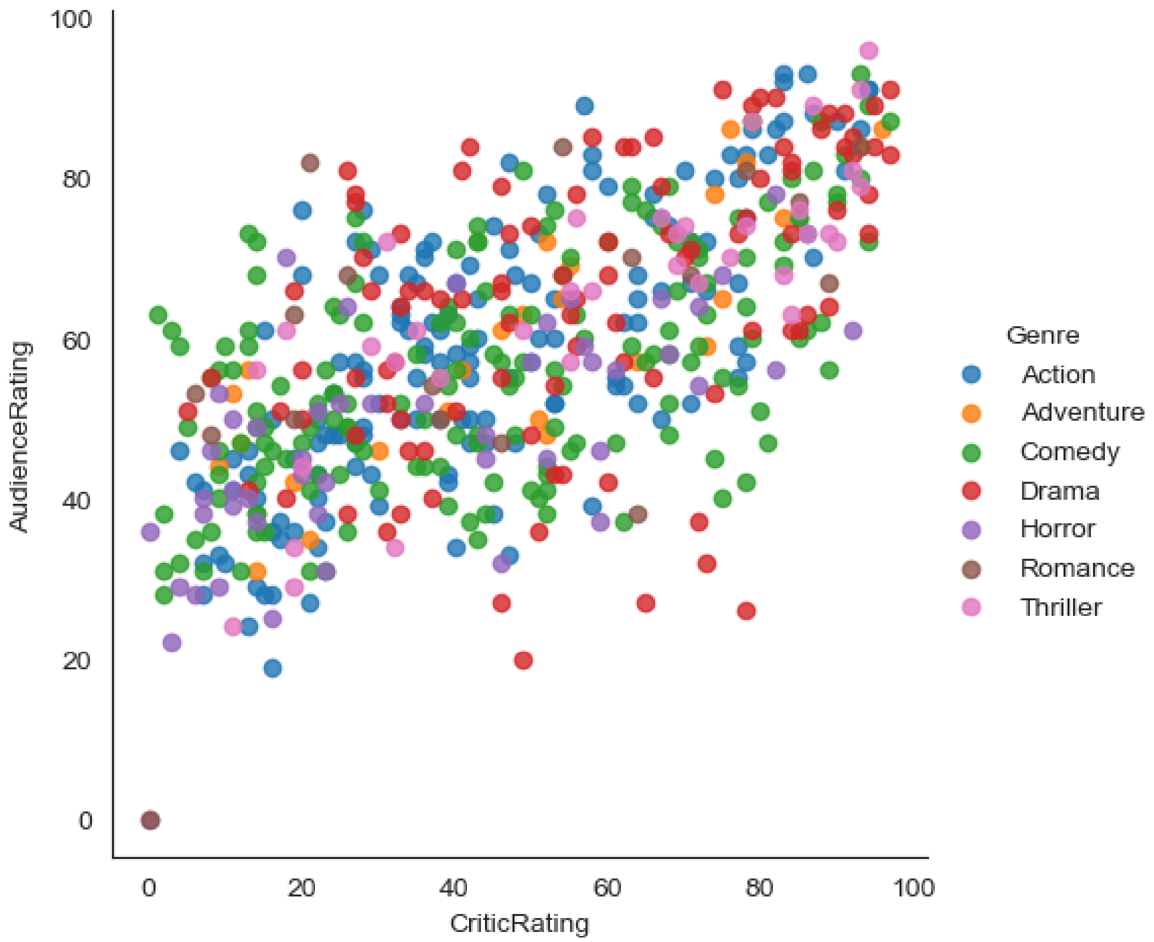
In [55]:
```
for gen in movies.Genre.cat.categories:
    print(gen)
```

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

In [56]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',\
                  fit_reg=False)
```

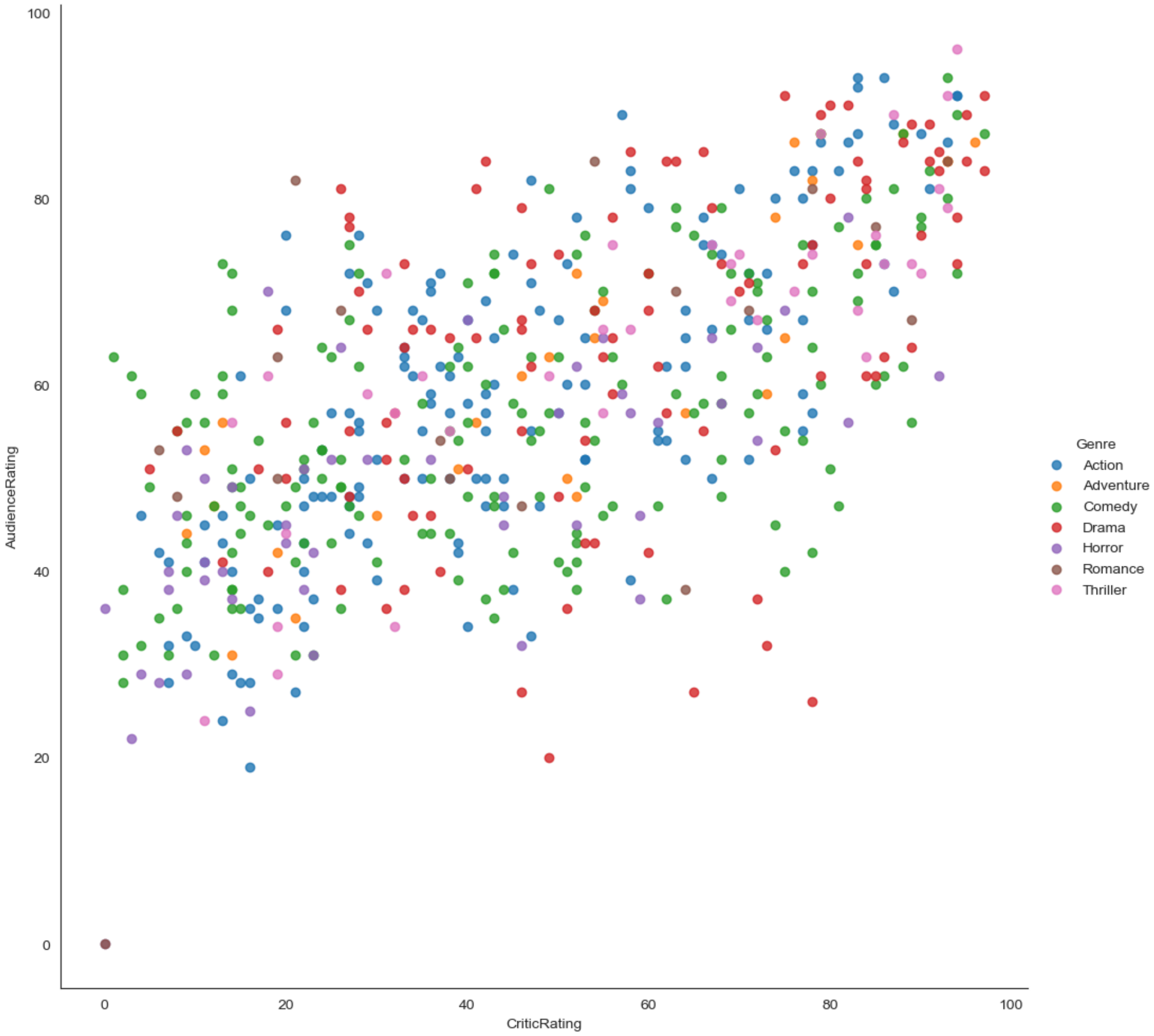


In [57]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',\
                  fit_reg=False, hue = 'Genre')
```

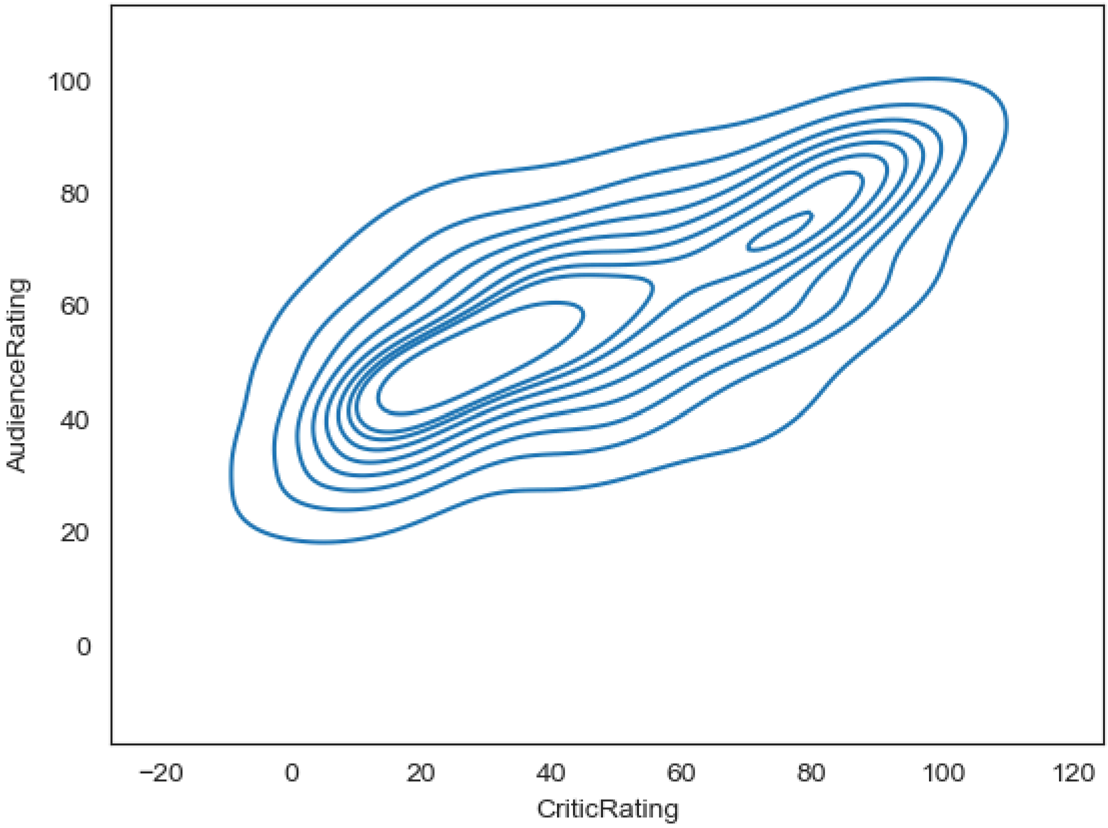


In [60]:

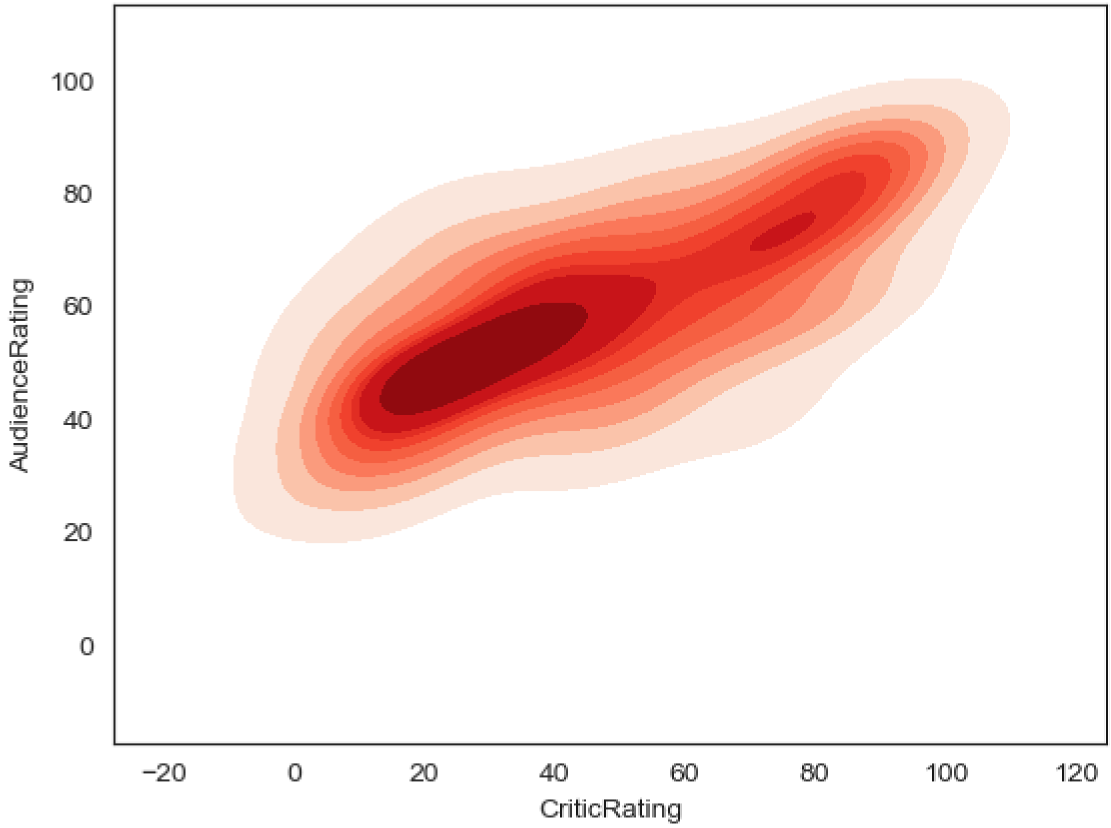
```
vis1 = sns.lmplot(
    data=movies,
    x='CriticRating',
    y='AudienceRating',
    fit_reg=False,
    hue='Genre',
    height=10,
    aspect=1
)
```



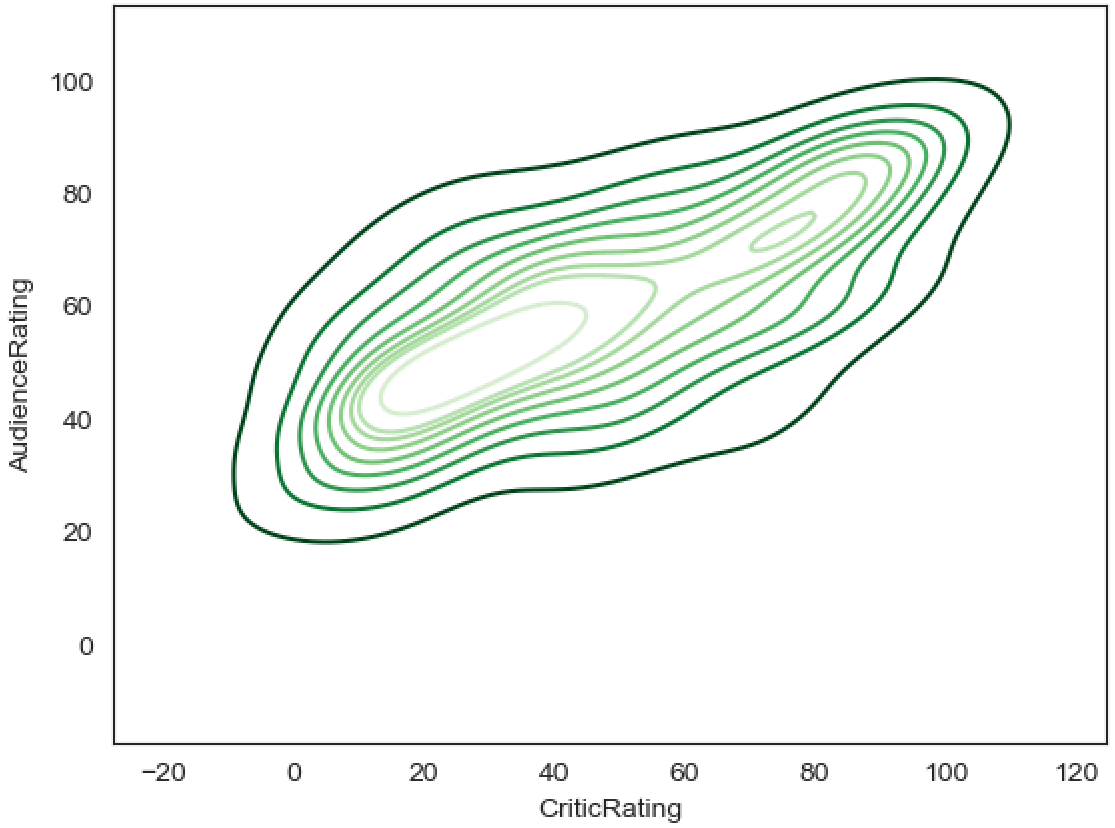
```
In [62]: k1 = sns.kdeplot(x=movies.CriticRating, y=movies.AudienceRating)
```



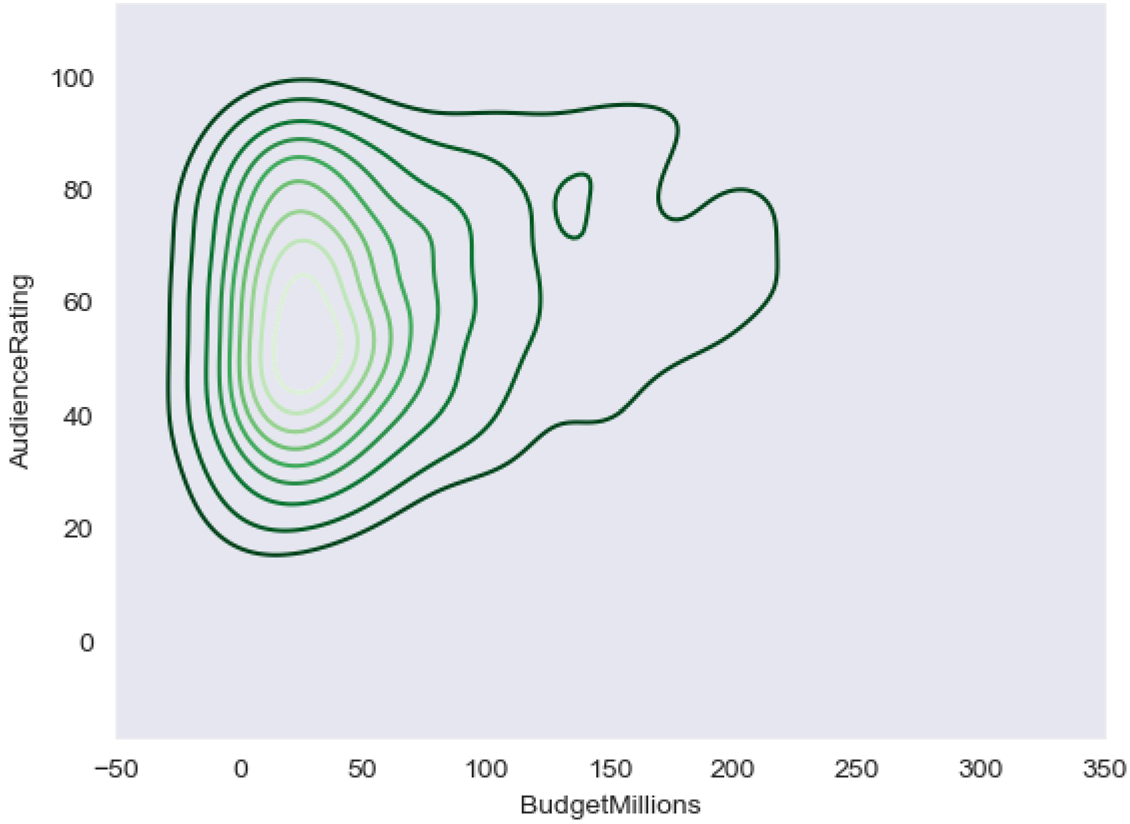
```
In [65]: k1 = sns.kdeplot(
             x=movies.CriticRating,
             y=movies.AudienceRating,
             shade=True,
             shade_lowest=False,
             cmap='Reds'
         )
```



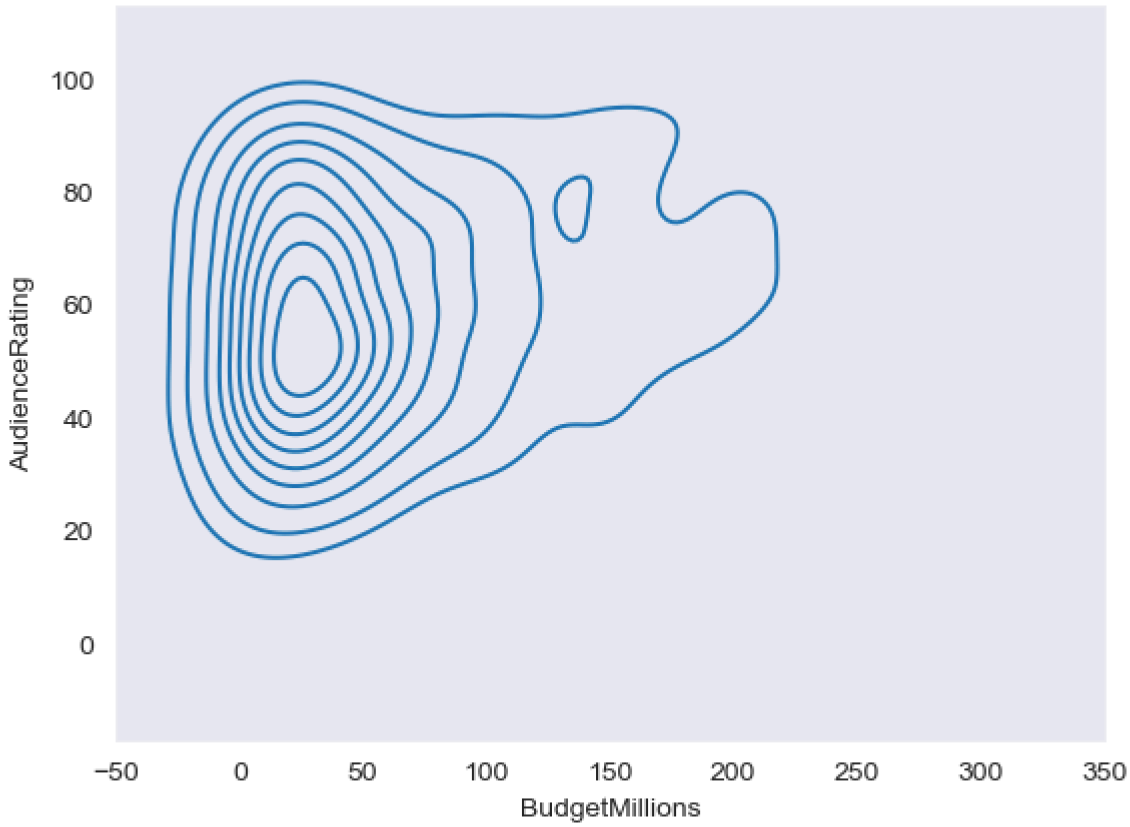
```
In [66]: k2 = sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,shade_lowest=Fals
```



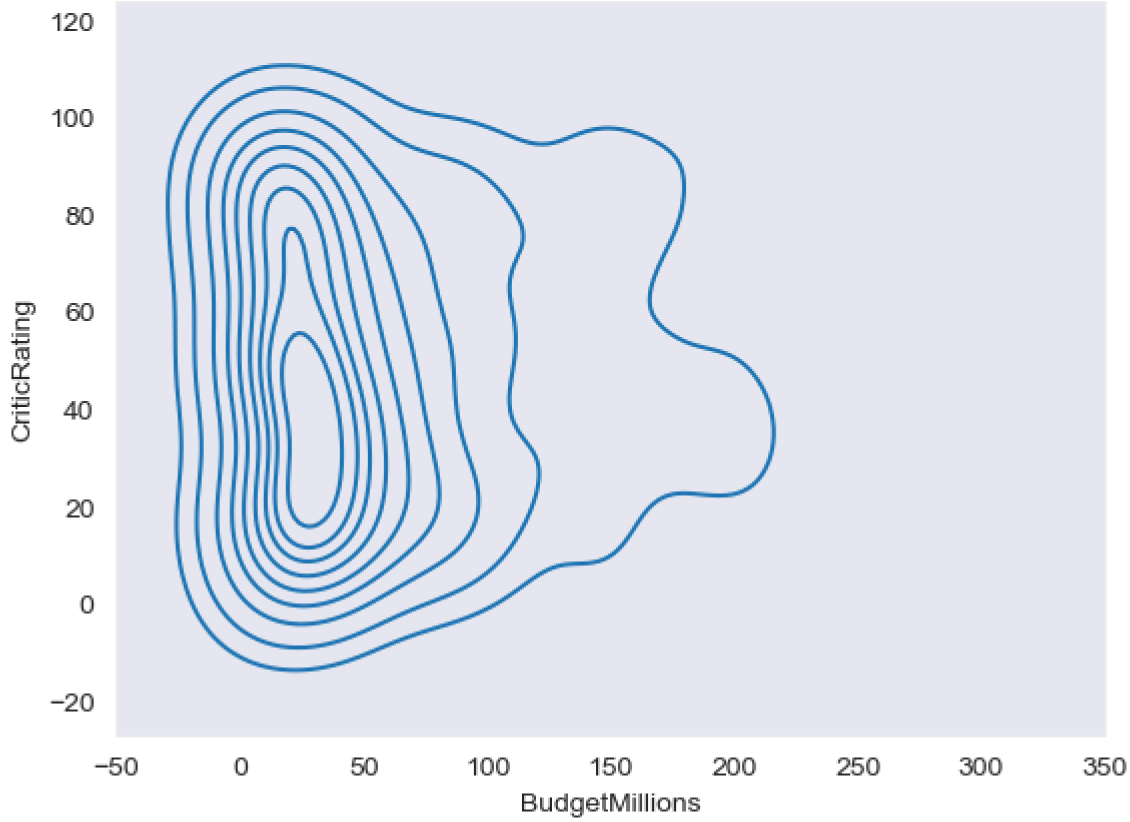
In [68]:
```
sns.set_style('dark')
k1 = sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,shade_lowest=Fa
```



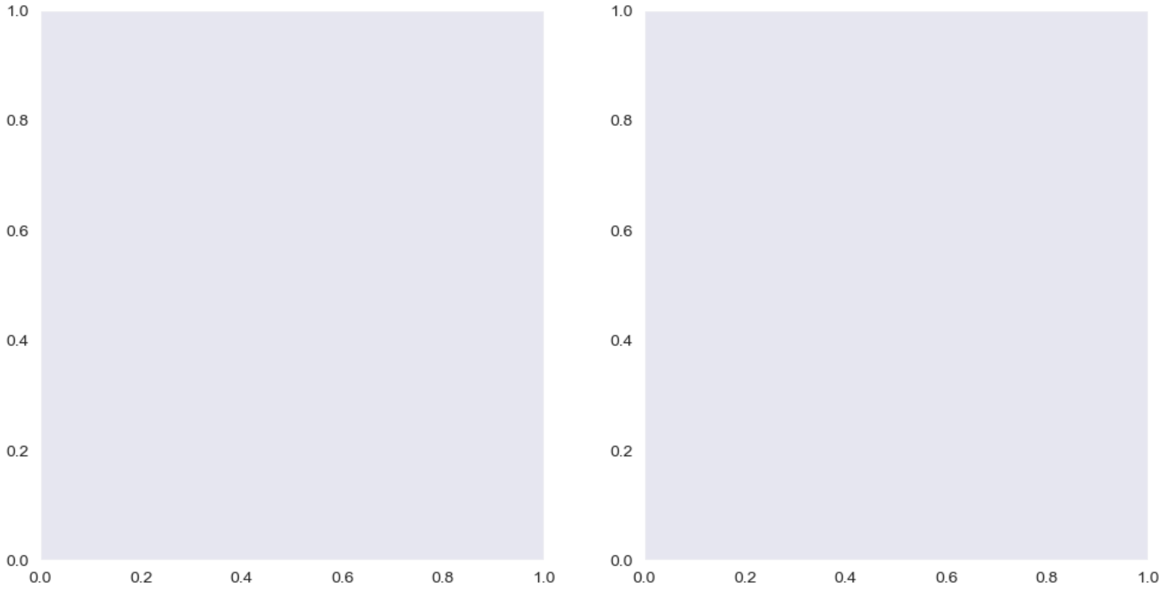
In [70]:
```
sns.set_style('dark')
k1 = sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating)
```



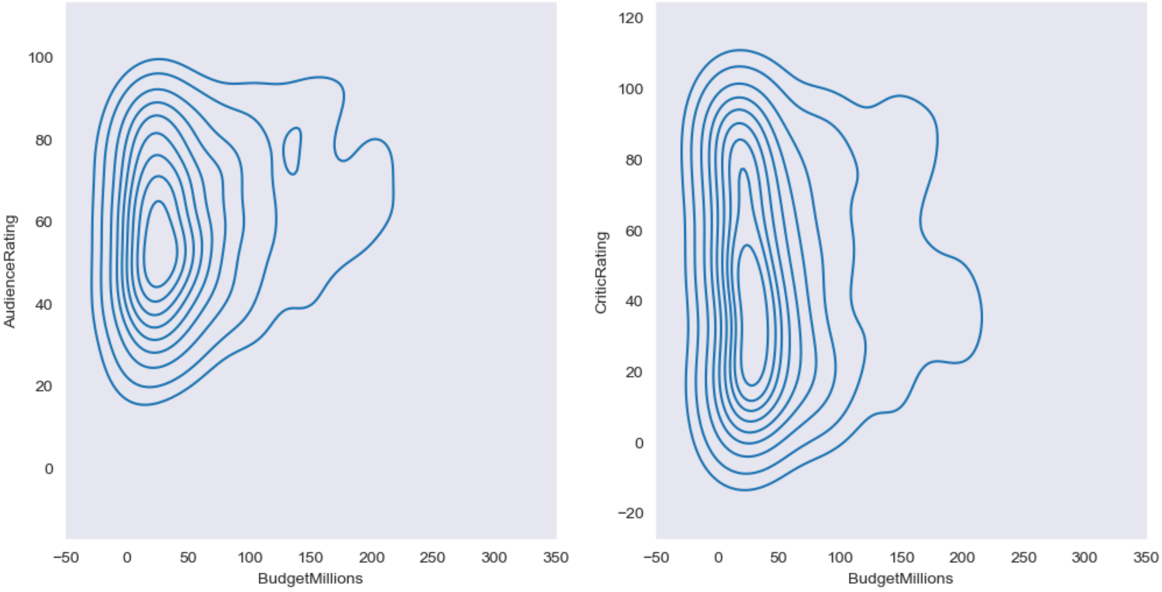
In [72]:
```
k2 = sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating)
```



In [73]:
```
f, ax = plt.subplots(1,2, figsize =(12,6))
```



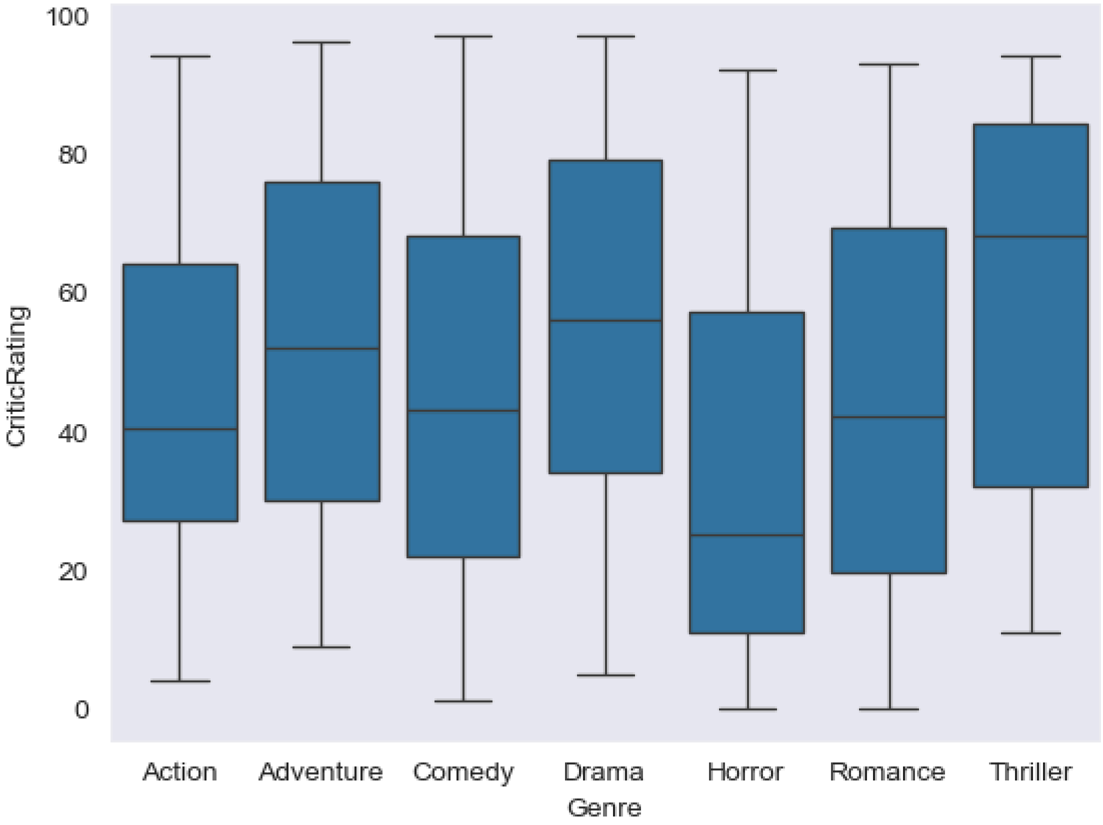
```
In [75]: f, axes = plt.subplots(1,2, figsize =(12,6))

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



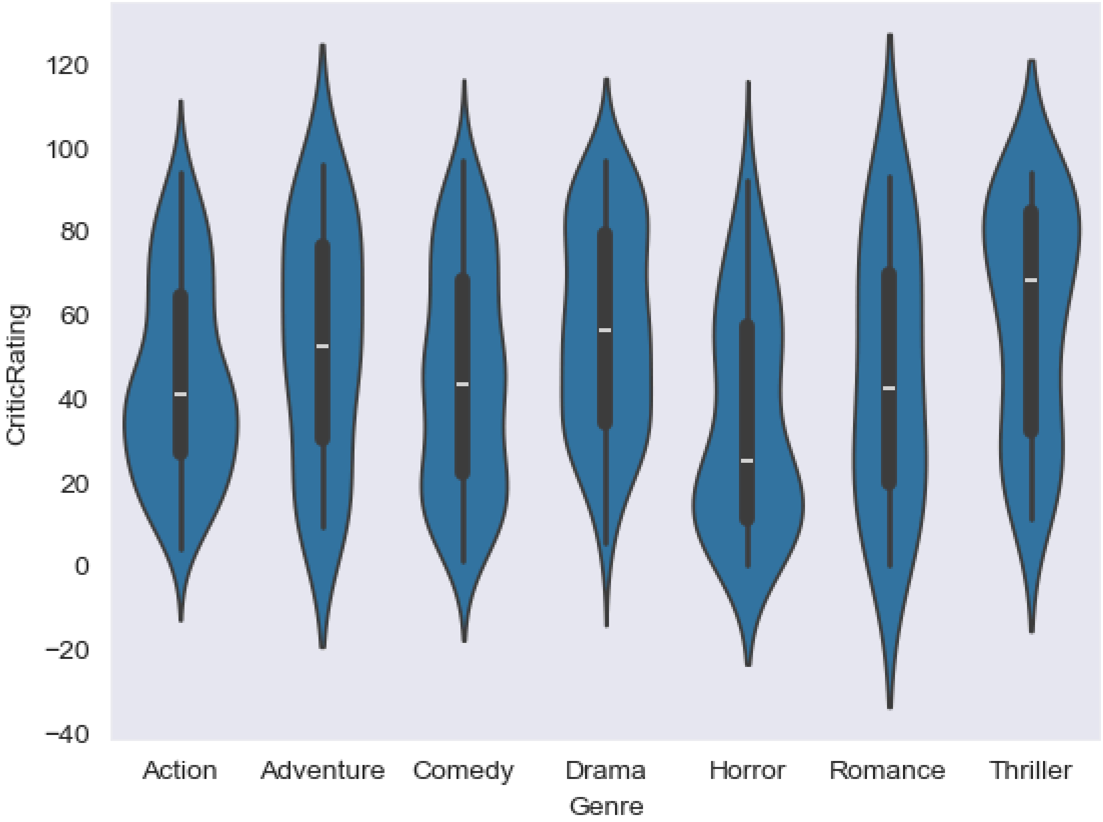
```
In [76]: axes
```

```
Out[76]: array([<Axes: xlabel='BudgetMillions', ylabel='AudienceRating'>,
                <Axes: xlabel='BudgetMillions', ylabel='CriticRating'>],
               dtype=object)
```

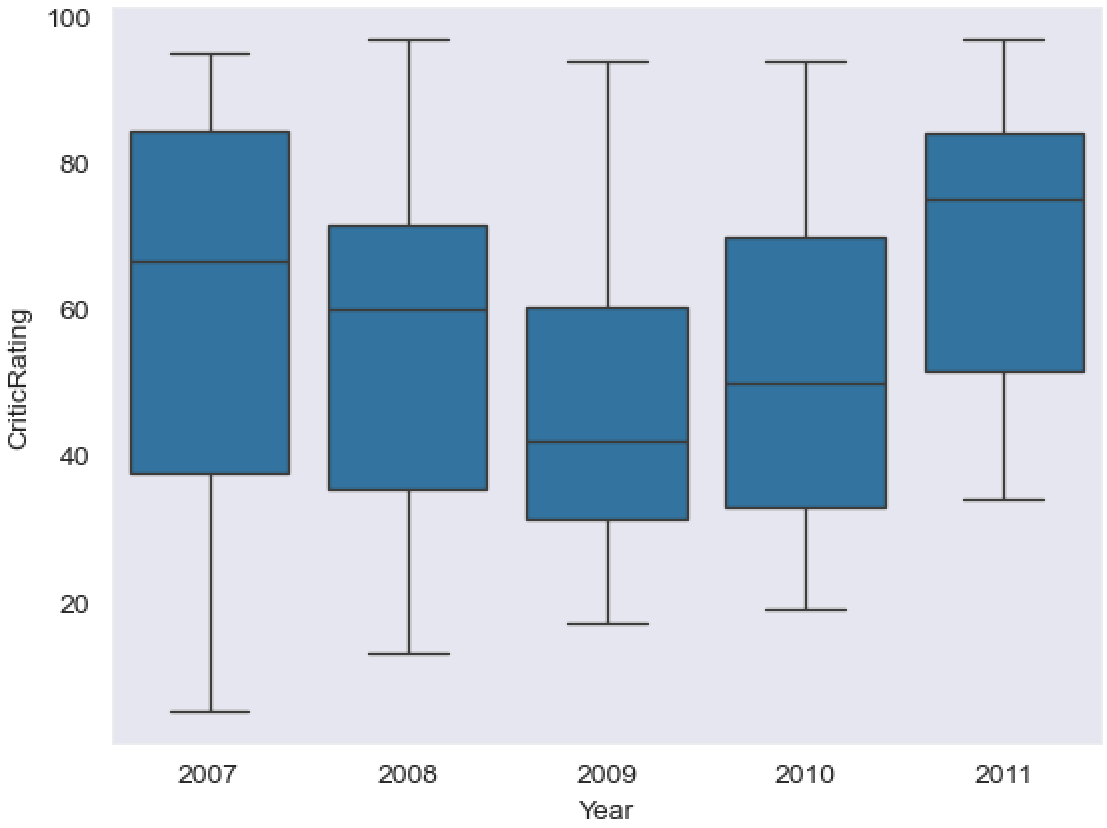
```
In [77]: w = sns.boxplot(data=movies, x='Genre', y = 'CriticRating')
```



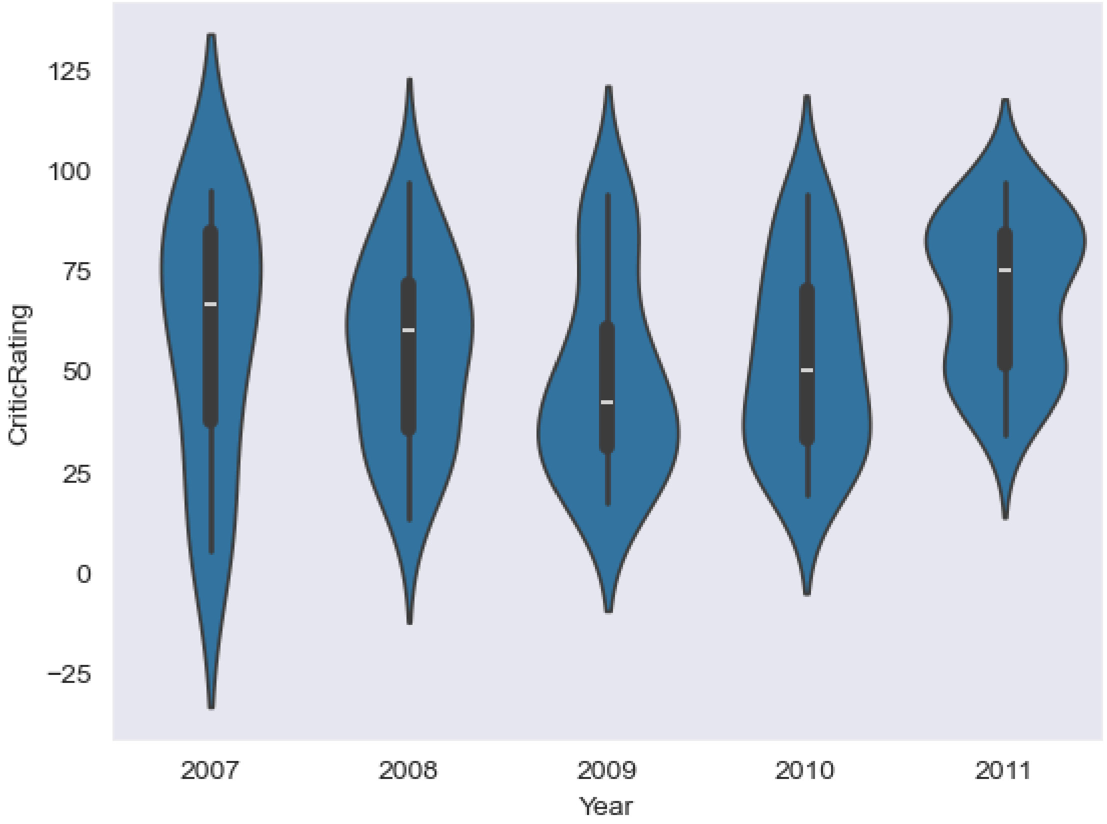
In [78]:
```
z = sns.violinplot(data=movies, x='Genre', y = 'CriticRating')
```



In [79]:
```
w1 = sns.boxplot(data=movies[movies.Genre == 'Drama'], x='Year', y = 'CriticRati
```



In [80]:
```
z = sns.violinplot(data=movies[movies.Genre == 'Drama'], x='Year', y = 'CriticRa
```



In [81]:
```
g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre')
```

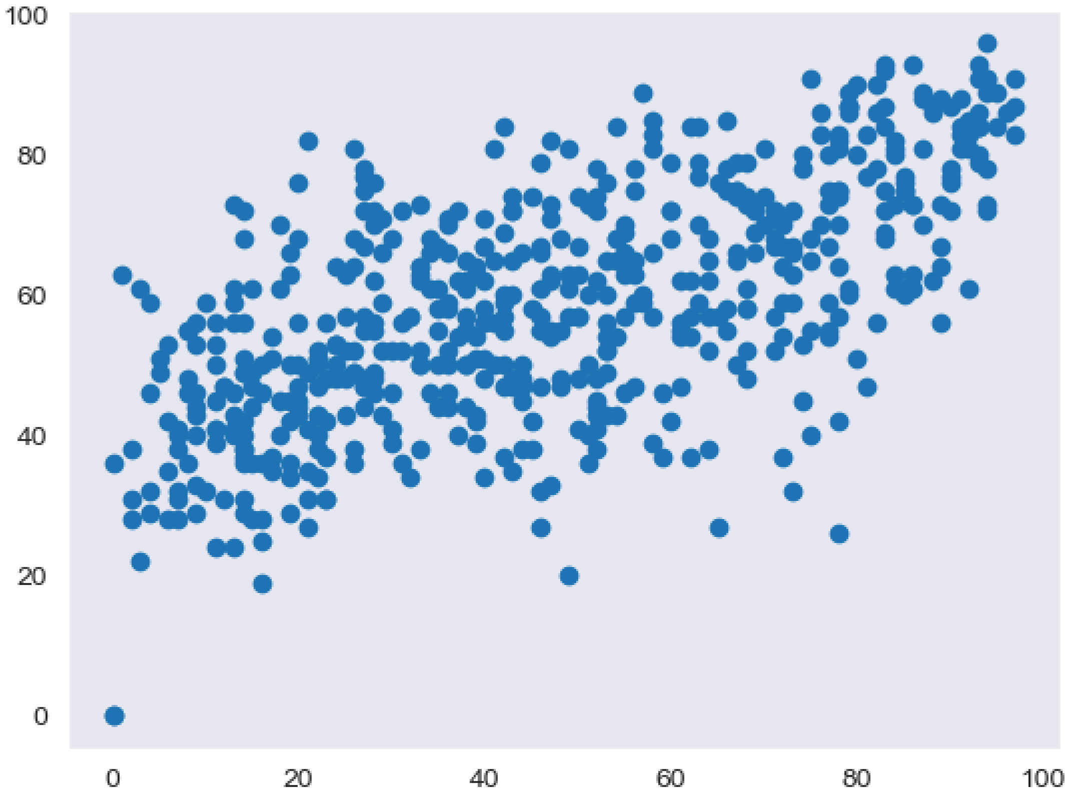
Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
Genre = Comedy | Year = 2007
Genre = Comedy | Year = 2008
Genre = Comedy | Year = 2009
Genre = Comedy | Year = 2010
Genre = Comedy | Year = 2011
Genre = Drama | Year = 2007
Genre = Drama | Year = 2008
Genre = Drama | Year = 2009
Genre = Drama | Year = 2010
Genre = Drama | Year = 2011
Genre = Horror | Year = 2007
Genre = Horror | Year = 2008
Genre = Horror | Year = 2009
Genre = Horror | Year = 2010
Genre = Horror | Year = 2011
Genre = Romance | Year = 2007
Genre = Romance | Year = 2008
Genre = Romance | Year = 2009
Genre = Romance | Year = 2010
Genre = Romance | Year = 2011
Genre = Thriller | Year = 2007
Genre = Thriller | Year = 2008
Genre = Thriller | Year = 2009
Genre = Thriller | Year = 2010
Genre = Thriller | Year = 2011
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

In [82]:

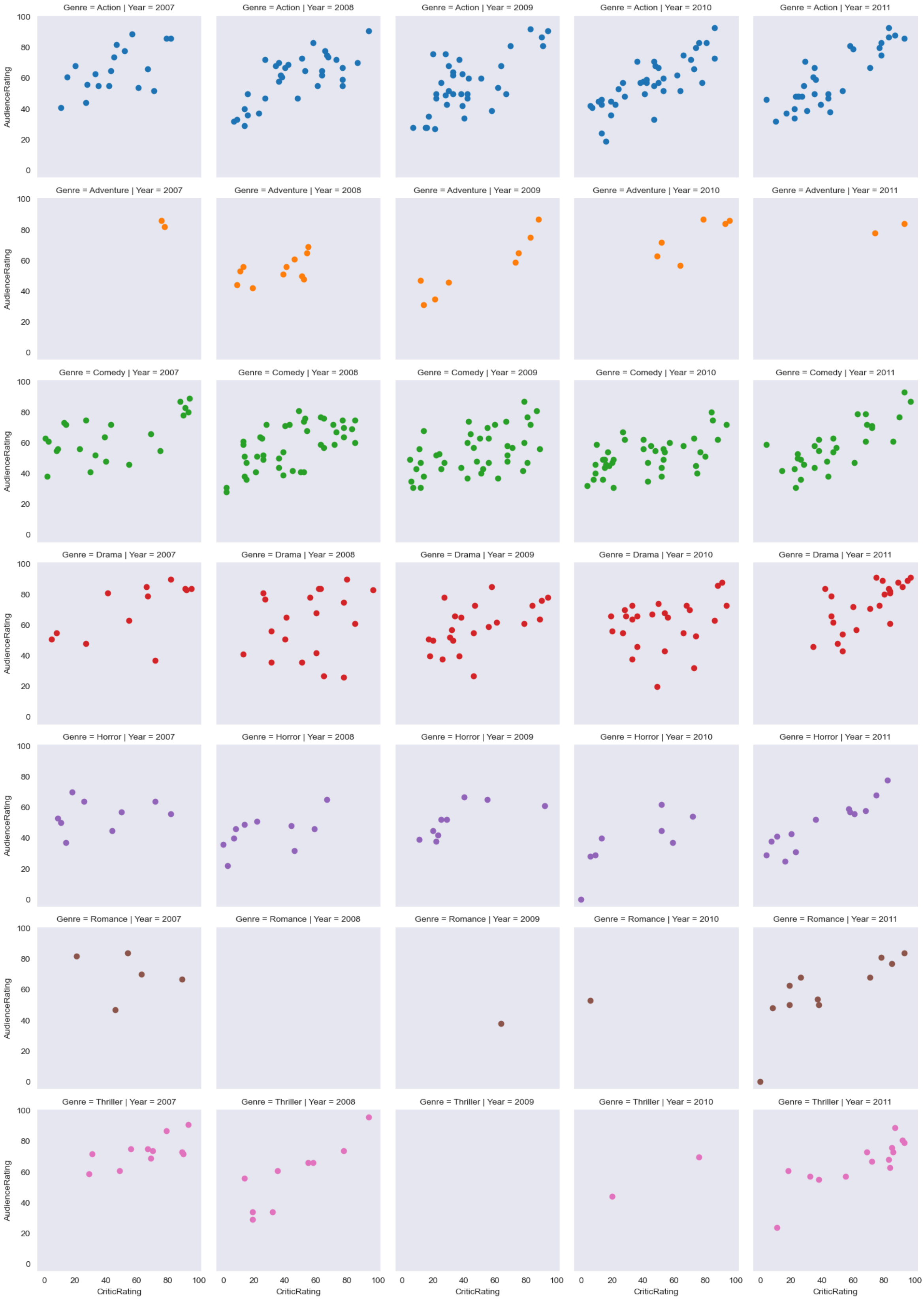
```
plt.scatter(movies.CriticRating,movies.AudienceRating)
```

Out[82]:

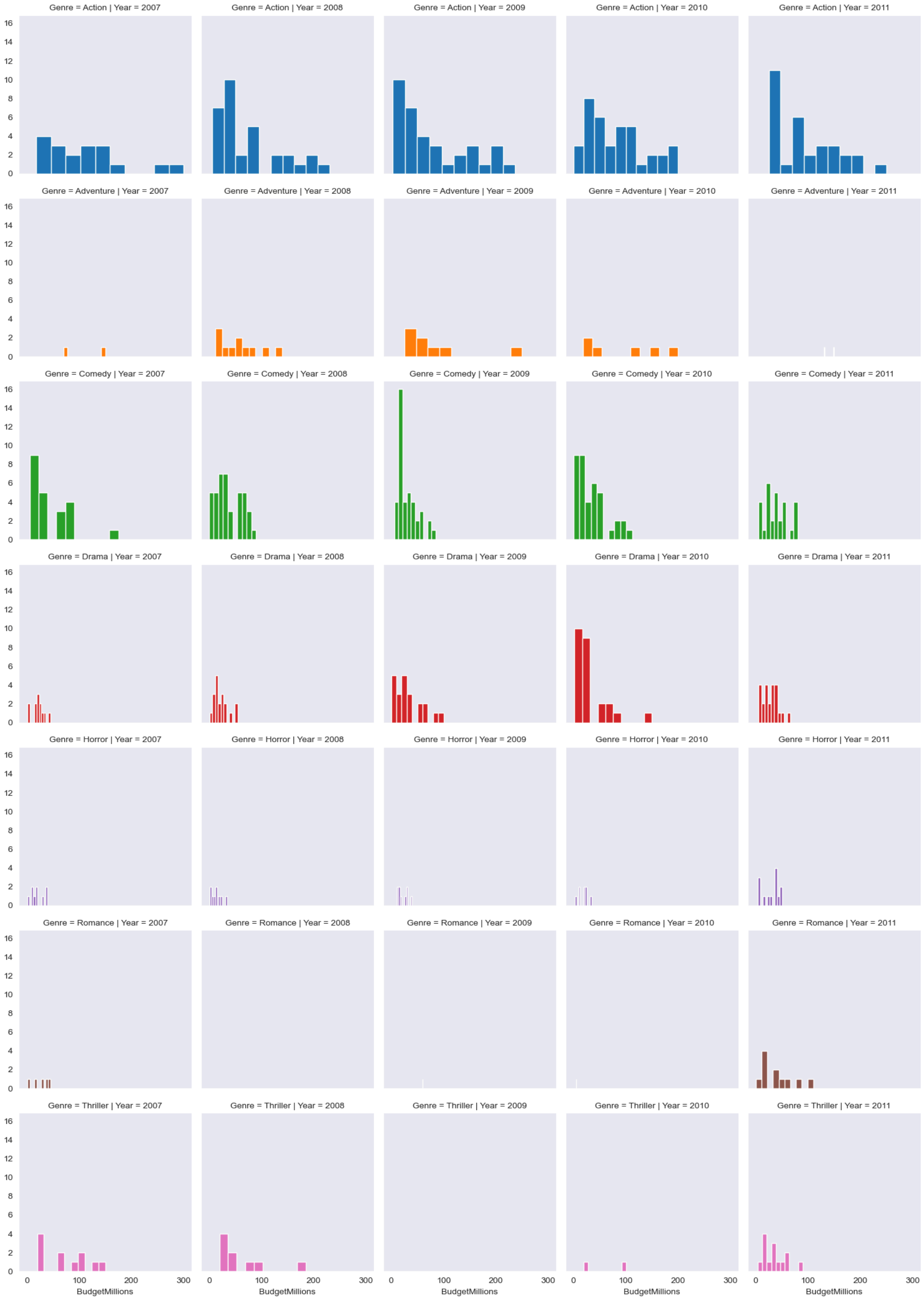
```
<matplotlib.collections.PathCollection at 0x2dc7063d450>
```



In [83]:
```
g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre')
g = g.map(plt.scatter, 'CriticRating', 'AudienceRating' )
```

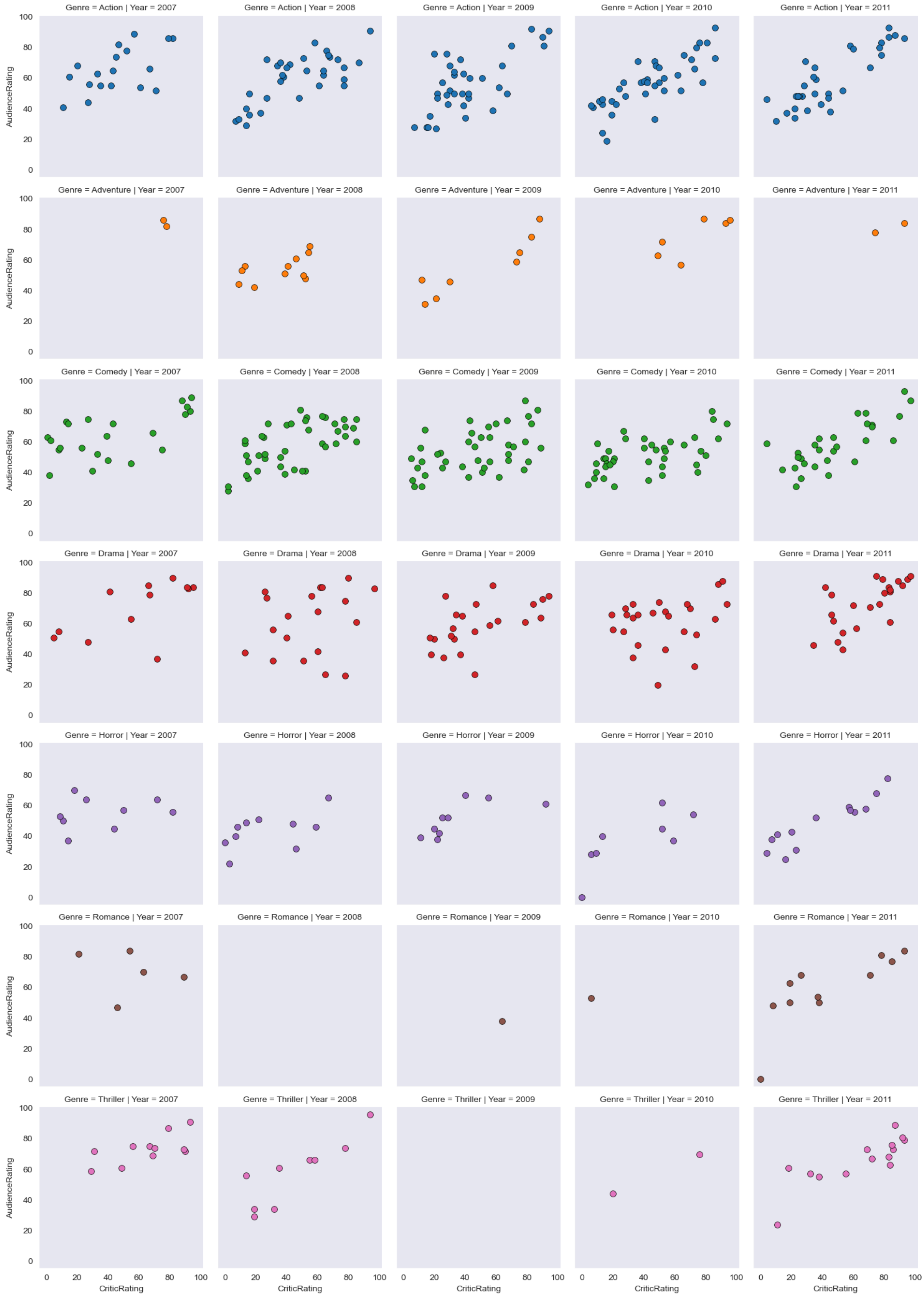


In [84]:
```
g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre')
g = g.map(plt.hist, 'BudgetMillions')
```



In [85]:

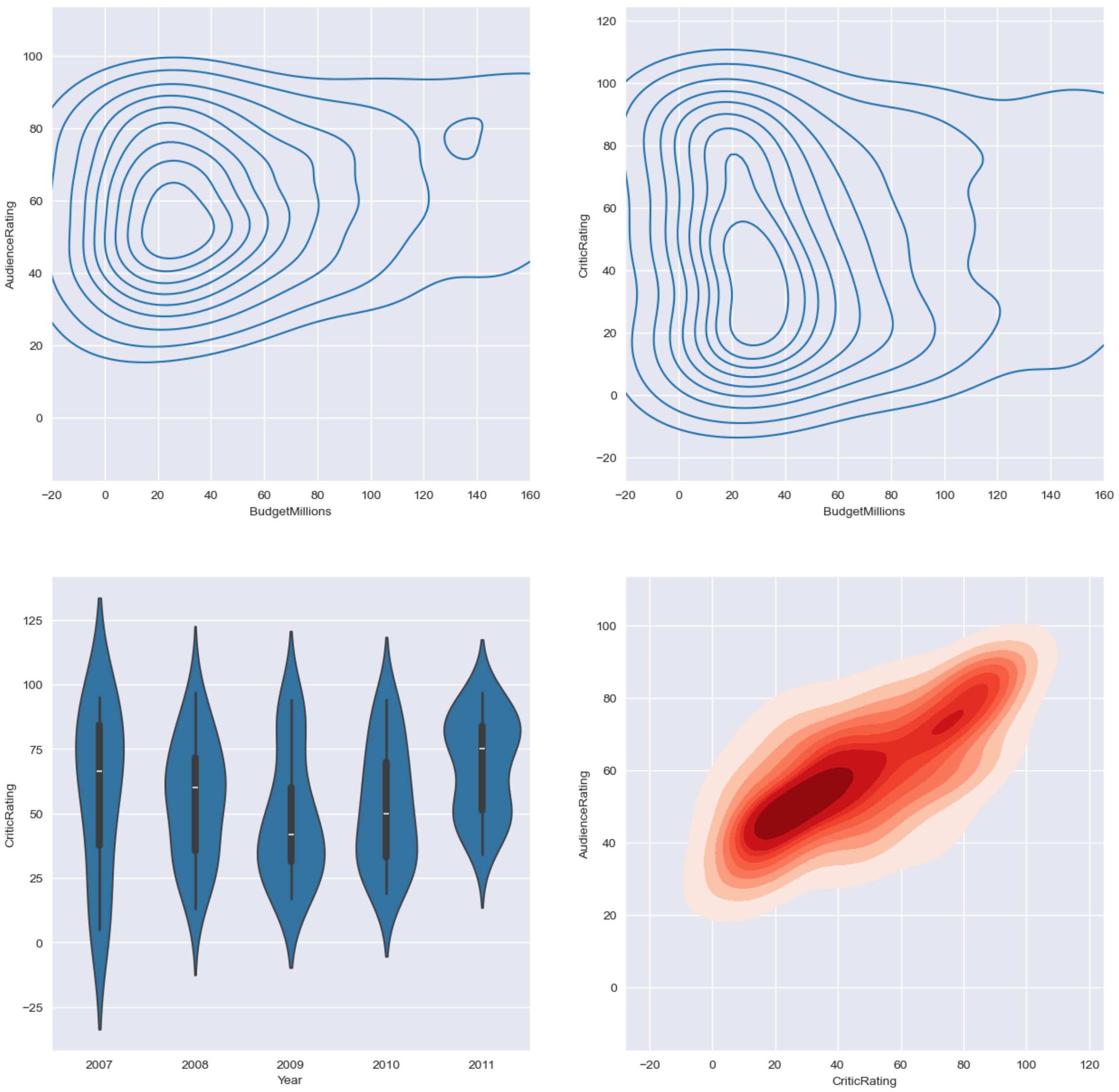
```
g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre')
kws = dict(s=50, linewidth=0.5,edgecolor='black')
g = g.map(plt.scatter, 'CriticRating', 'AudienceRating',**kws )
```



In [92]:

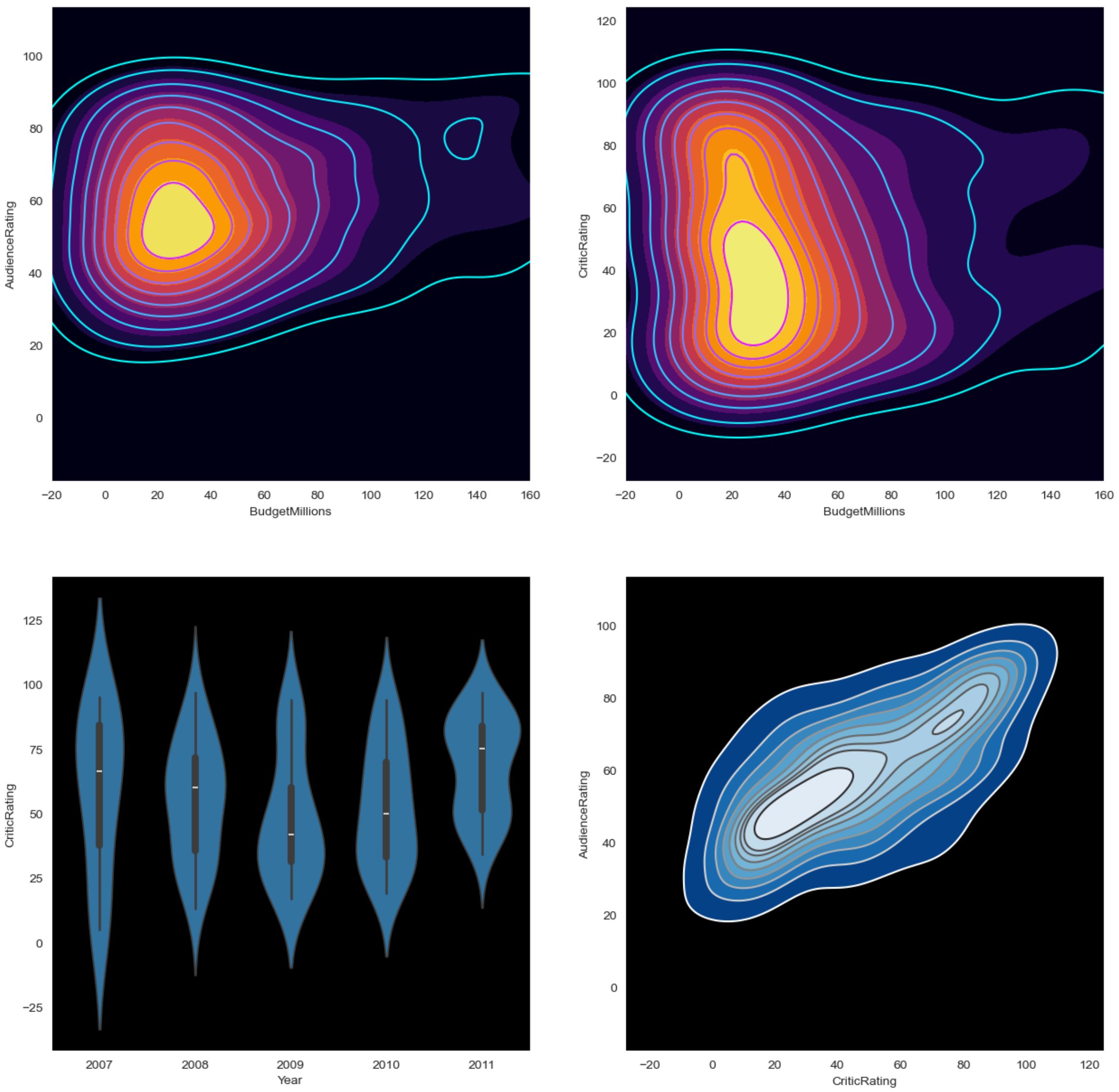
```
sns.set_style('darkgrid')
f, axes = plt.subplots (2,2, figsize = (15,15))

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


k4b = sns.kdeplot(movies.CriticRating, movies.AudienceRating,cmap='Reds',ax = ax

plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[92], line 22
     10 z = sns.violinplot(data=movies[movies.Genre=='Drama'], x='Year', y = 'Cri
ticRating', ax=axes[1,0])
     12 k4 = sns.kdeplot(
     13     x=movies.CriticRating,
     14     y=movies.AudienceRating,
   (...)
     18     ax=axes[1,1]
     19 )
---> 22 k4b = sns.kdeplot(movies.CriticRating, movies.AudienceRating,cmap='Reds',
ax = axes[1,1])
     24 plt.show()

TypeError: kdeplot() takes from 0 to 1 positional arguments but 2 positional argu
ments (and 1 keyword-only argument) were given
```



In [96]:

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

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()
```


AudienceRating
100
80
60
40
20
0
−20
0
20
40
60
80
100
120
140
160
BudgetMillions
CriticRating
120
100
80
60
40
20
0
−20
−20
0
20
40
60
80
100
120
140
160
BudgetMillions
CriticRating
125
100
75
50
25
0
−25
2007
2008
2009
2010
2011
Year
AudienceRating
100
80
60
40
20
0
−20
0
20
40
60
80
100
120
CriticRating

In [ ]: