Movie Rating Analytics

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
movies = pd.read_csv("Movie-Rating.csv")
```

In [4]:
```
movies.head()
```

Out[4]:

	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 [5]:
```
len(movies)
```

Out[5]:
```
559
```

In [6]:
```
movies.shape
```

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

In [7]:
```
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 [8]:
```
movies.columns
```

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

In [9]:
```
movies.tail()
```

Out[9]:

	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 [10]:
```
movies.columns = ['Film', 'Genre', 'CriticRating', 'AudienceRating',
       'BudgetMillions', 'Year']
```

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']
```

Out[14]:

```
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: object
```

In [15]:

```
movies.Film  = movies.Film.astype('category')
```

In [16]:

```
movies.Film
```

Out[16]:

```
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 [17]:

```
movies.describe()
```

Out[17]:

	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 [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   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 [19]:

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

In [20]:

```
movies.Genre
```

Out[20]:

```
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', 'R
omance', 'Thriller']
```

In [21]:

```
movies.Year
```

Out[21]:
```
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 [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    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 [23]:
```
movies.Genre.cat.categories
```

Out[23]:
```
Index(['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance',
       'Thriller'],
      dtype='object')
```

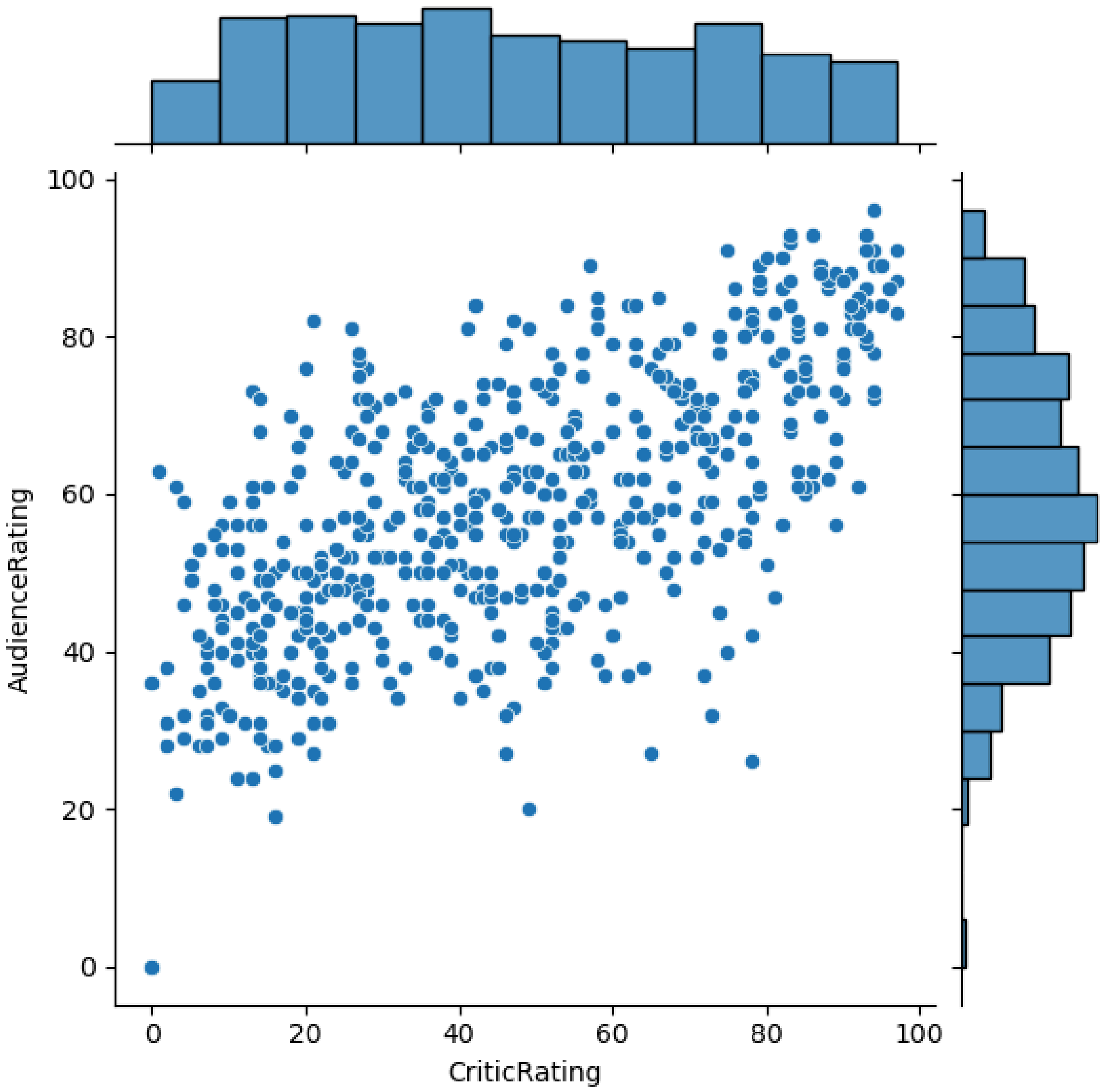
In [24]:
```
movies.describe()
```

Out[24]:

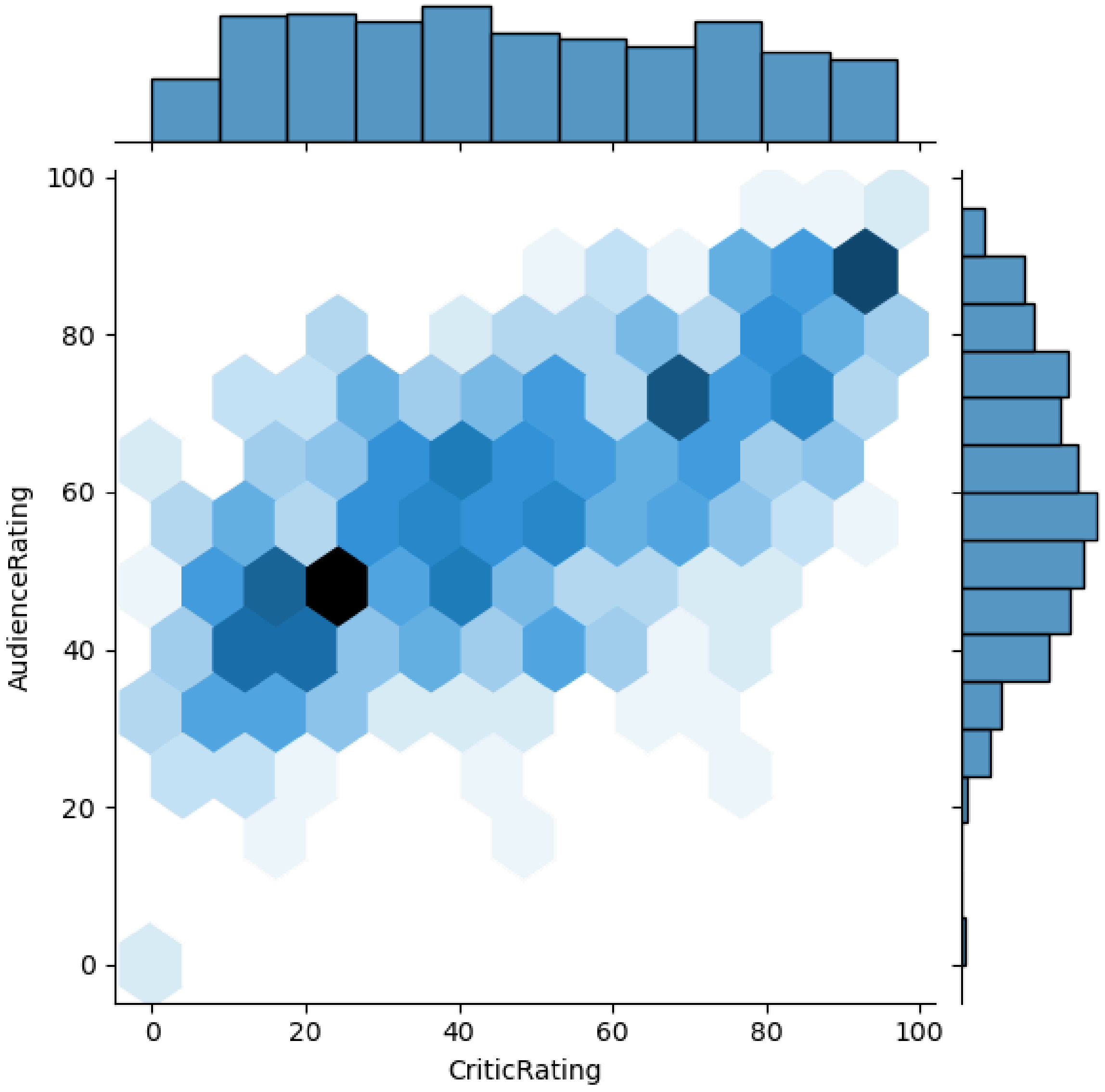
	CriticRating	AudienceRating	BudgetMillions
count	559.000000	559.000000	559.000000
mean	47.309481	58.744186	50.236136
std	26.413091	16.826887	48.731817
min	0.000000	0.000000	0.000000
25%	25.000000	47.000000	20.000000
50%	46.000000	58.000000	35.000000
75%	70.000000	72.000000	65.000000
max	97.000000	96.000000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

In [25]:
```
from matplotlib import pyplot as plt
import seaborn as sns
```

In [26]:
```
j = sns.jointplot(data = movies, x ='CriticRating', y = 'AudienceRating', kind =
```

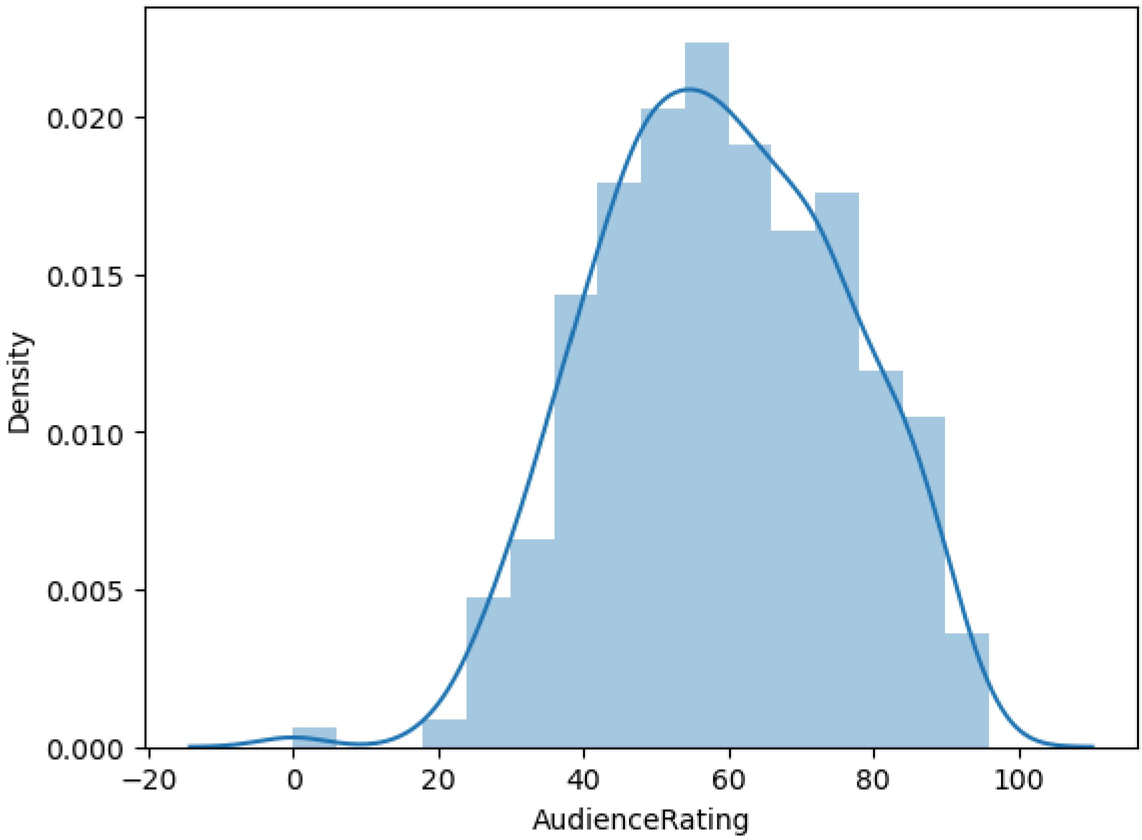


```
In [27]: j = sns.jointplot(data = movies, x='CriticRating', y='AudienceRating', kind ='he
```

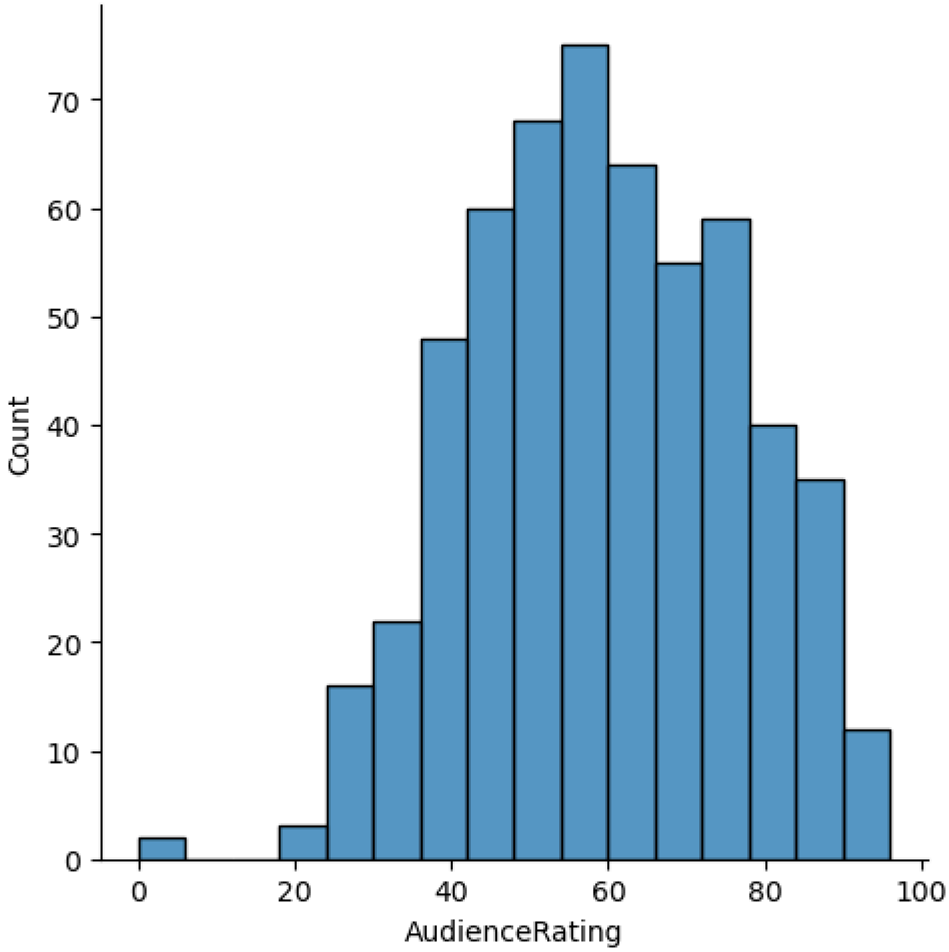


In [28]:

```
# Histograms

m1 = sns.distplot(movies.AudienceRating)                        # normal distribution
```

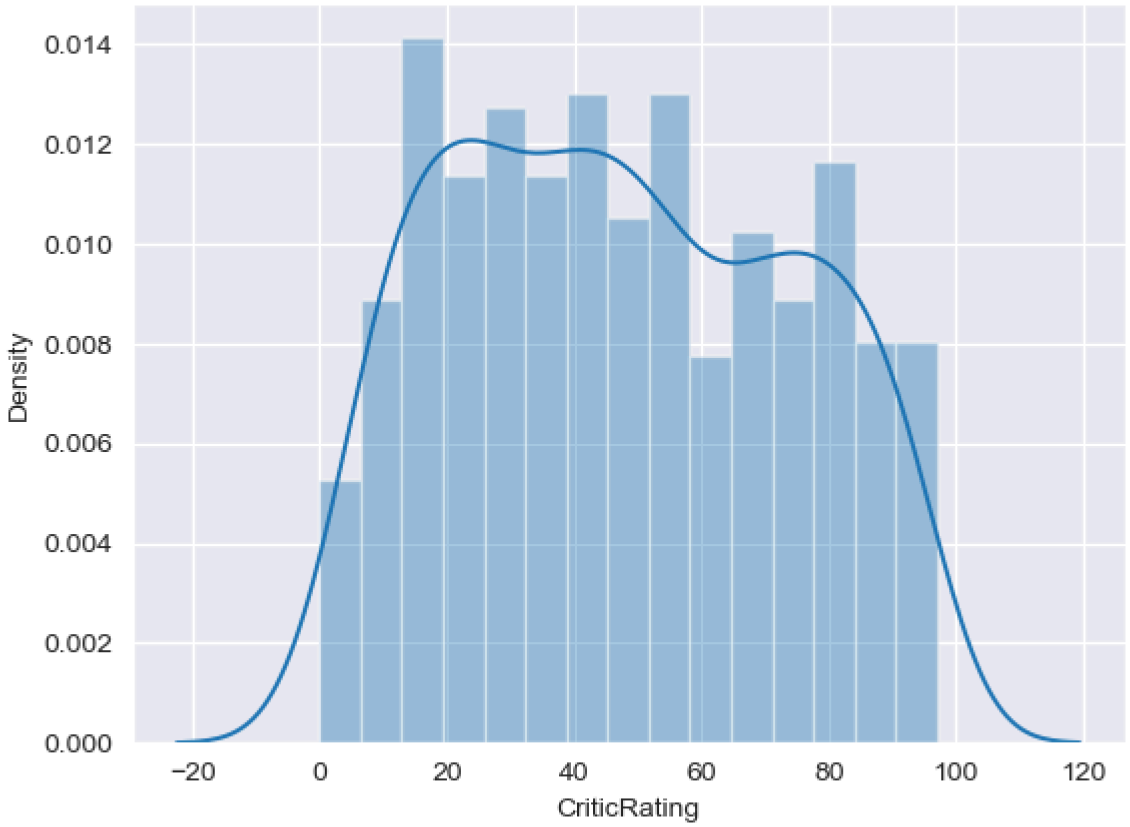


In [29]:
```
s1 = sns.displot(movies.AudienceRating)                    # normal distribution
```

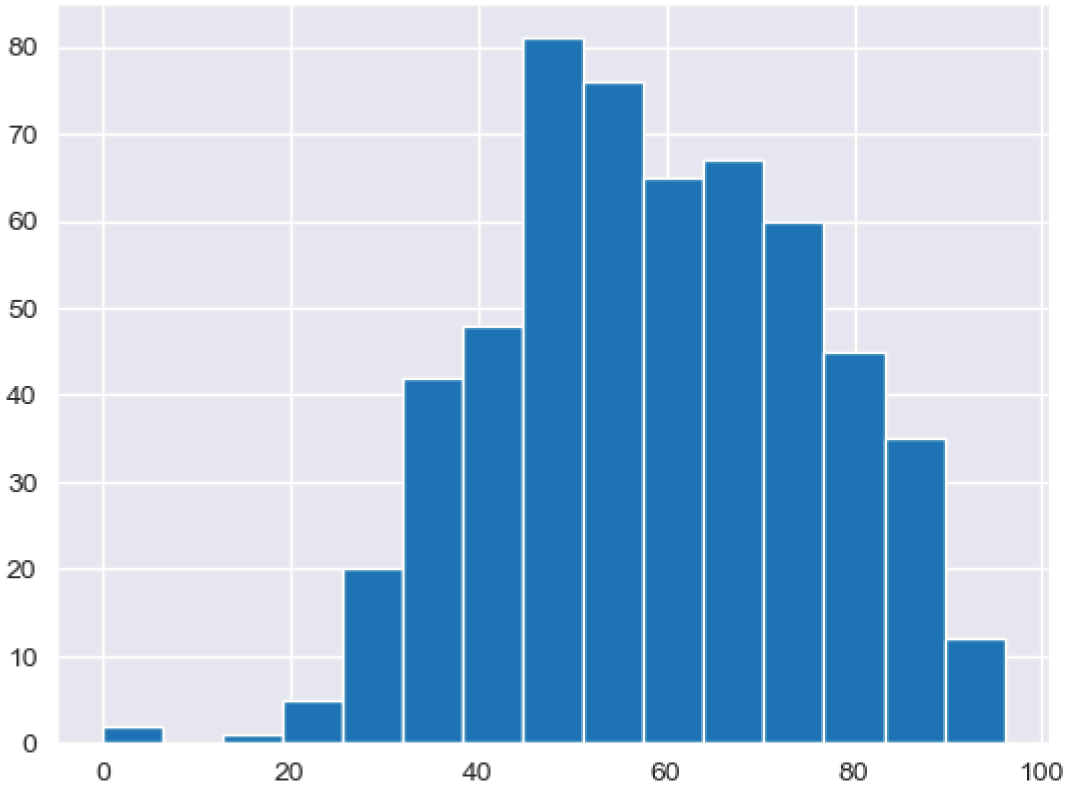


In [30]:
```
sns.set_style('darkgrid')
```

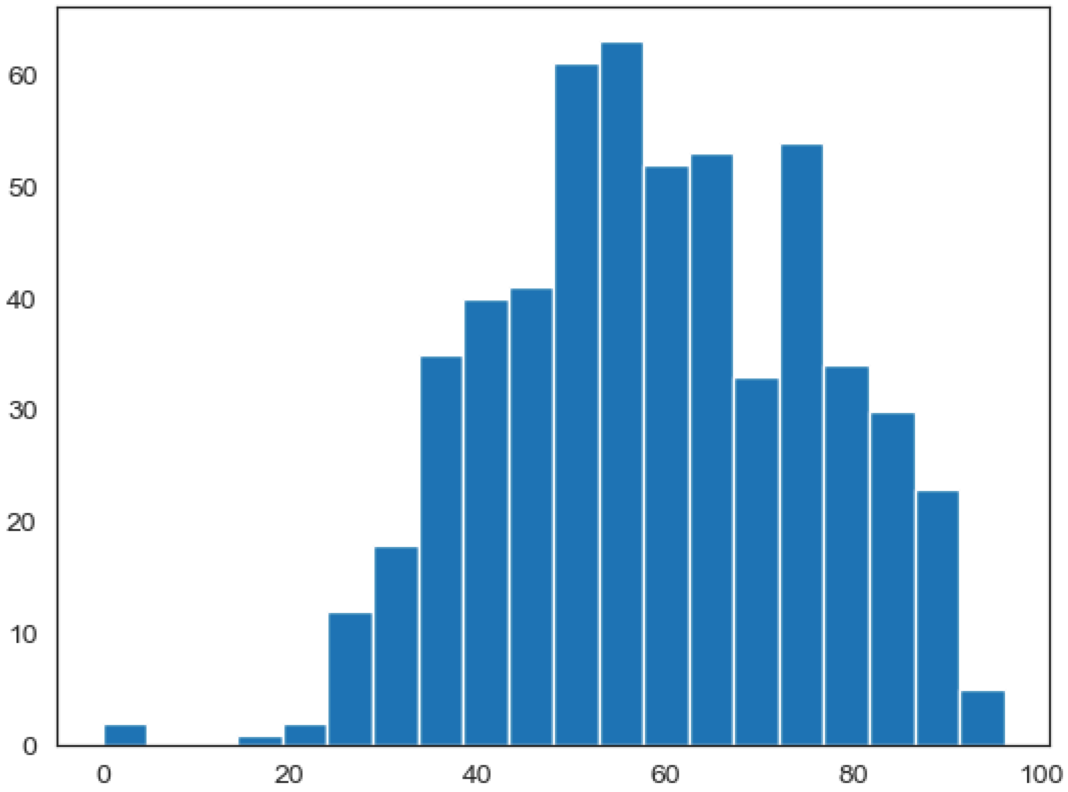
```
In [31]: m2 = sns. distplot(movies.CriticRating, bins = 15)
```



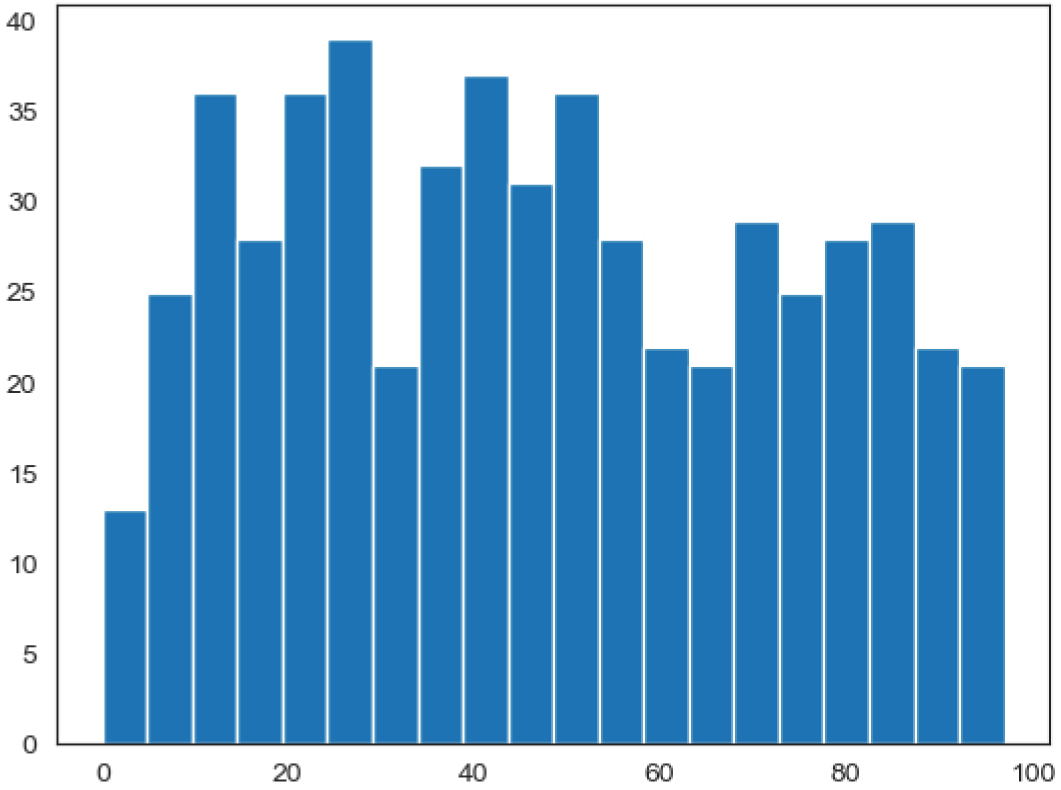
```
In [32]: # sns.set_style('darkgrid')

         n1 = plt.hist(movies.AudienceRating, bins=15)
```



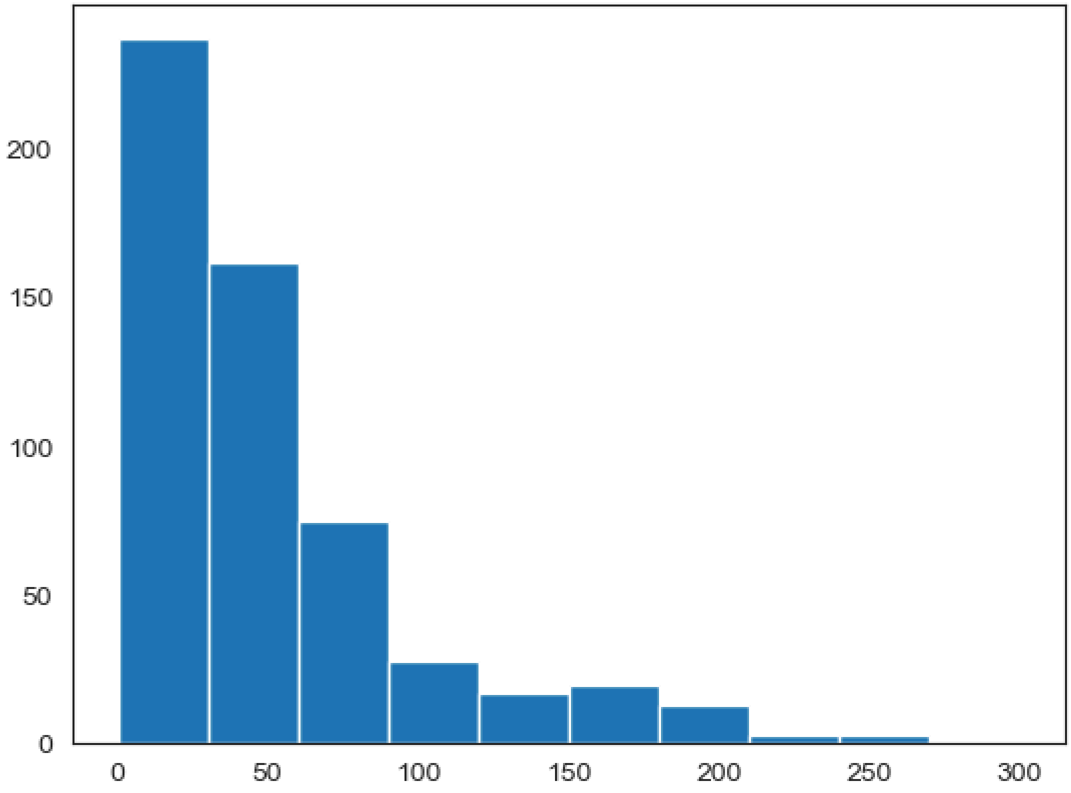
```
In [33]: sns.set_style('white')  #normal ditribution & called as bell curve
         n1 = plt.hist(movies.AudienceRating,bins=20)
```



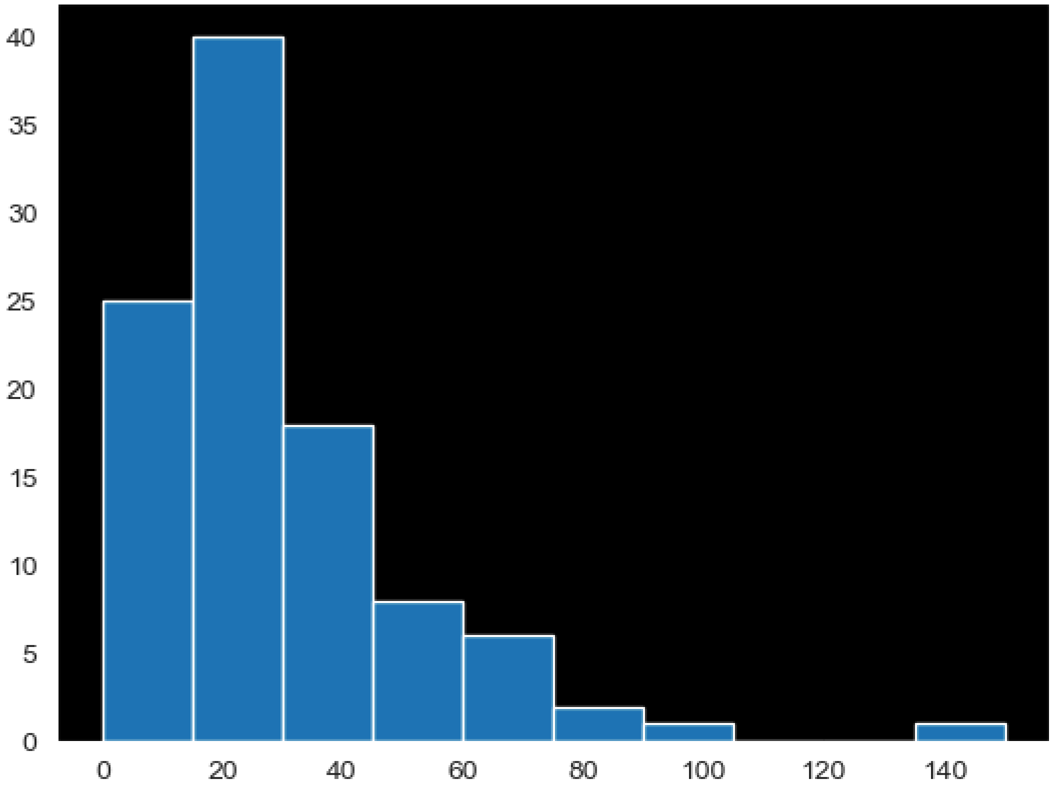
```
In [34]: n1 = plt.hist(movies.CriticRating,bins=20)
```



```
In [35]: plt.hist(movies.BudgetMillions)
         plt.show()
```



```
In [64]: plt.hist(movies[movies.Genre == 'Drama'].BudgetMillions)
         plt.show()
```



```
In [36]: movies.head()
```

Out[36]:

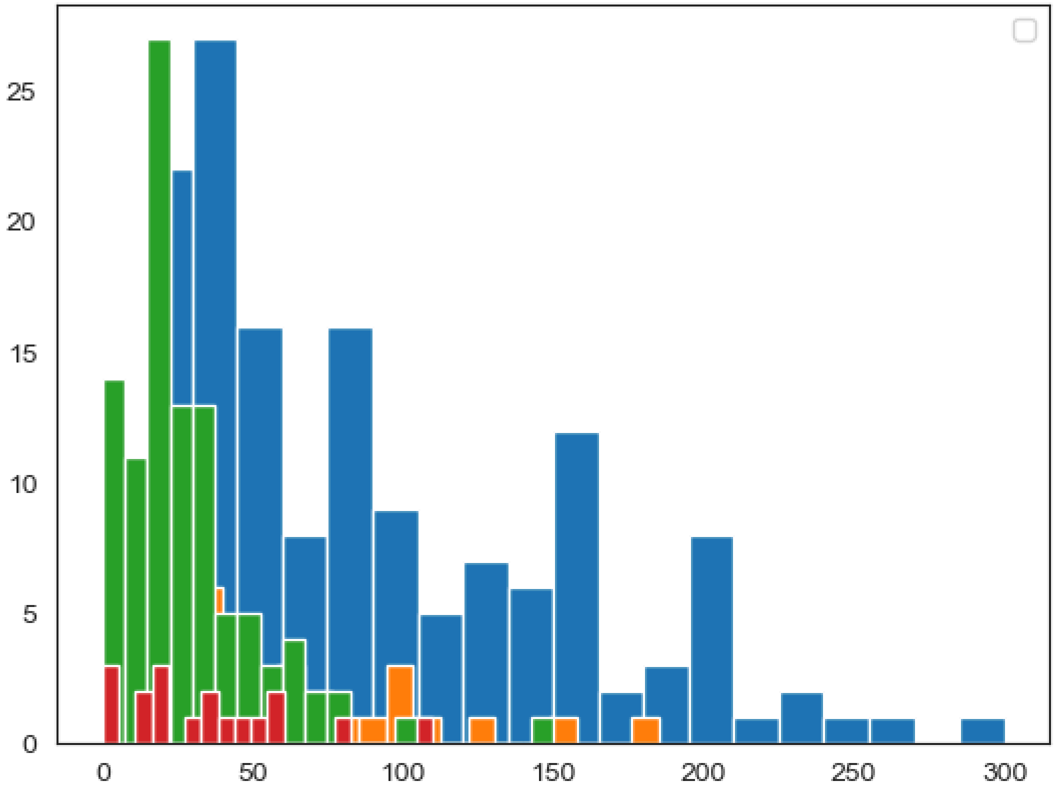
	Film	Genre	CriticRating	AudienceRating	BudgetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009

In [37]:
```
movies.Genre.unique()
```

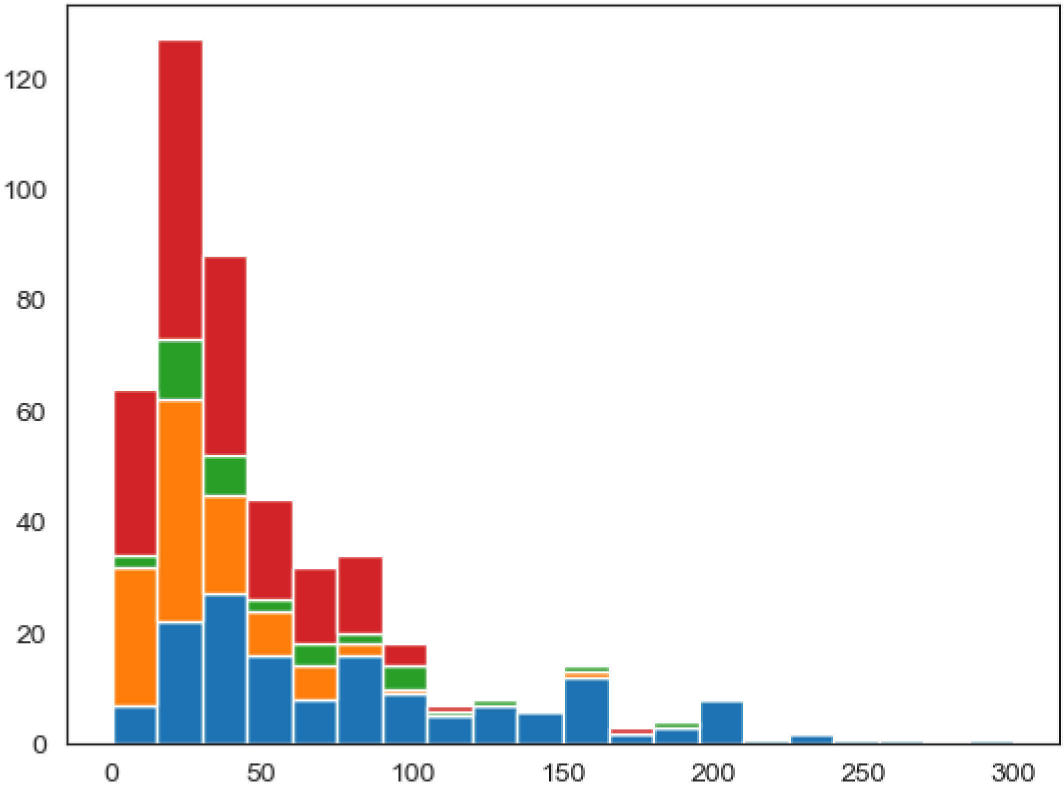
Out[37]:
```
['Comedy', 'Adventure', 'Action', 'Horror', 'Drama', 'Romance', 'Thriller']
Categories (7, object): ['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'R
omance', 'Thriller']
```

In [38]:
```
# Below plots are stacked histograms because overlapped
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.hist(movies[movies.Genre == 'Romance'].BudgetMillions, bins = 20)

plt.legend()
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose label sta
rt with an underscore are ignored when legend() is called with no argument.
```



In [39]:
```
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()
```



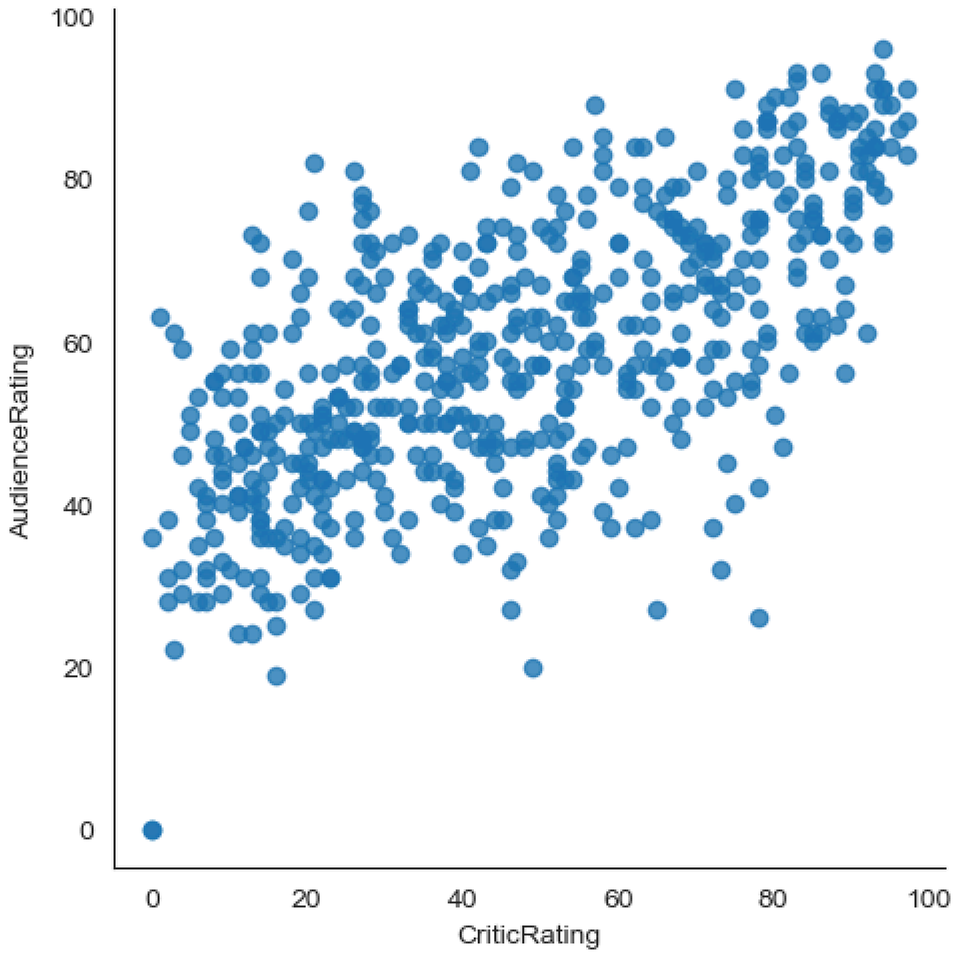
In [40]:

```
# if you have 100 cetgories you cannot copy & paste all the things

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

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

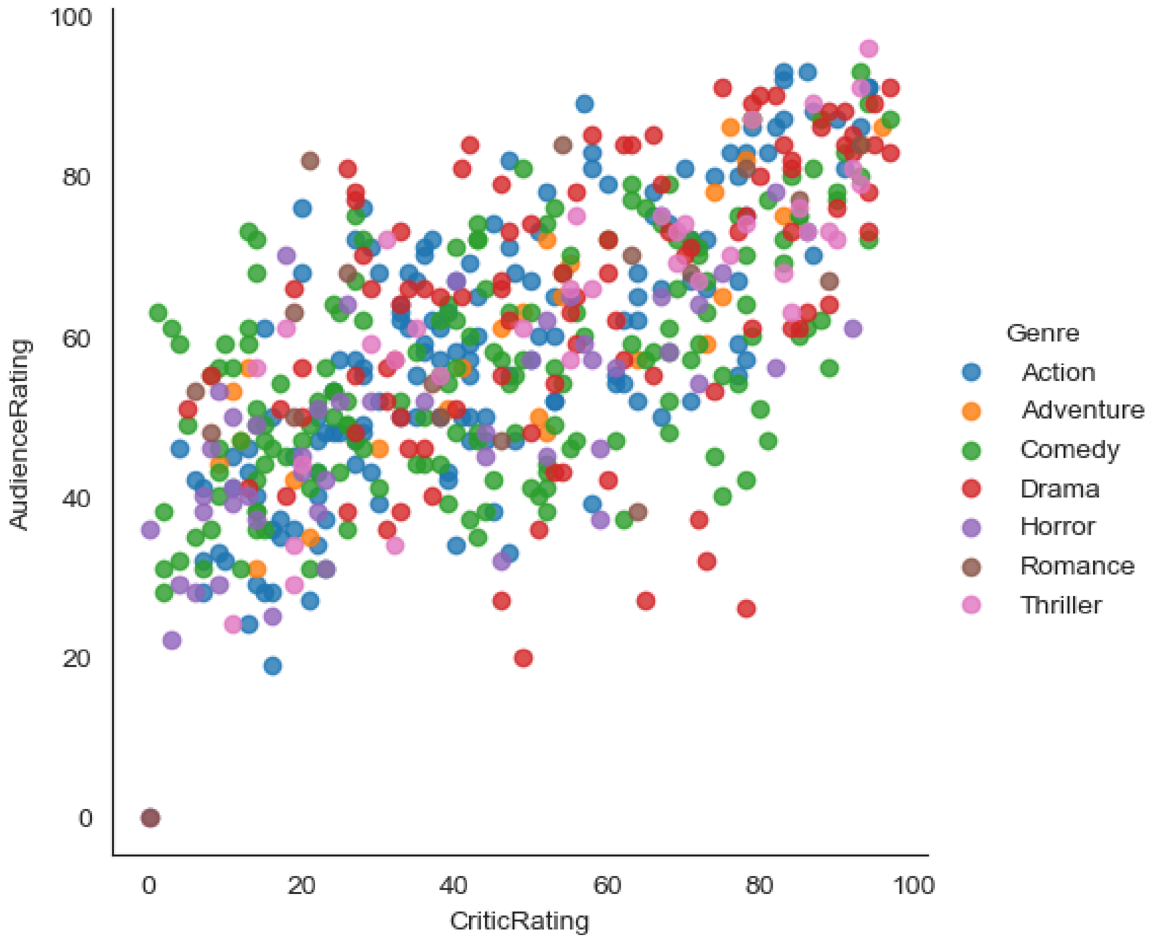
In [41]:

```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',\
                  fit_reg = False)
```

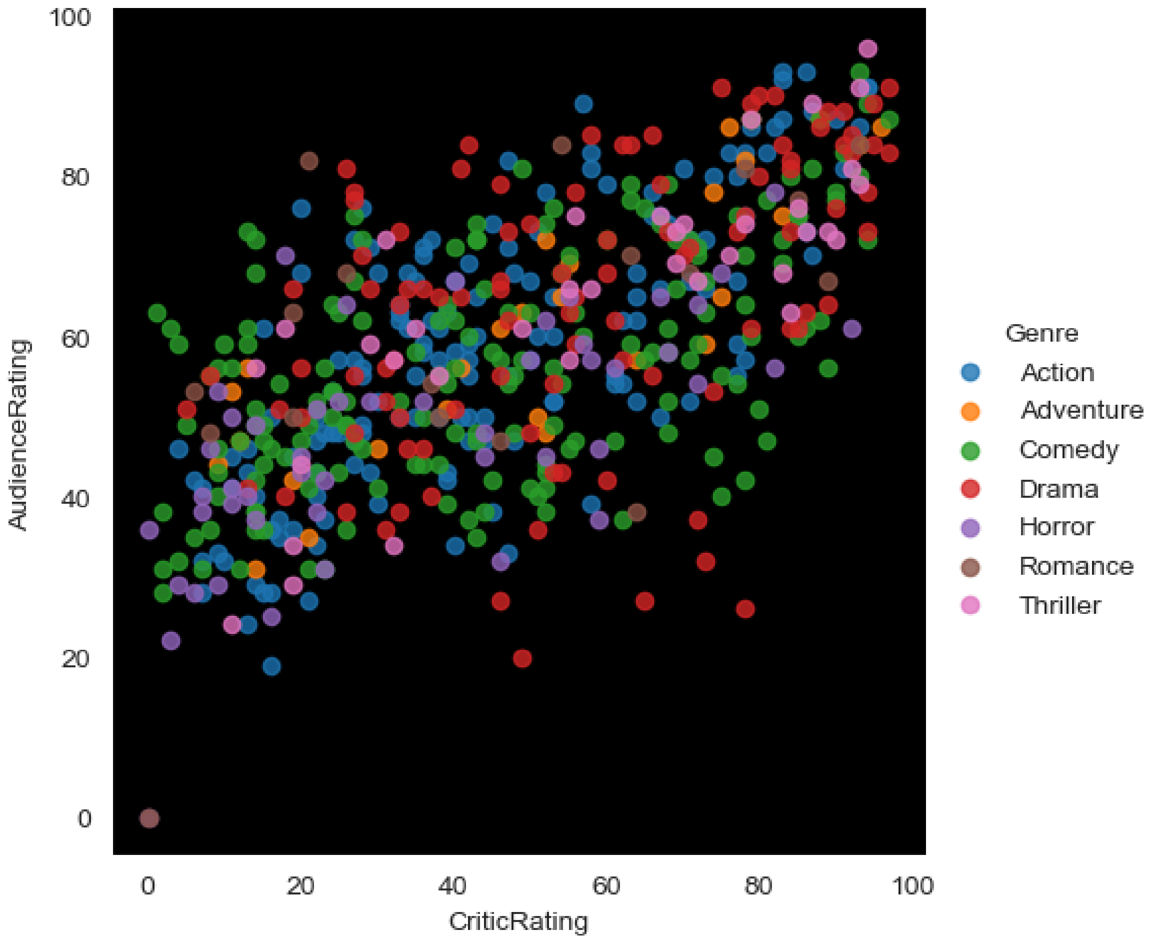


In [42]:

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

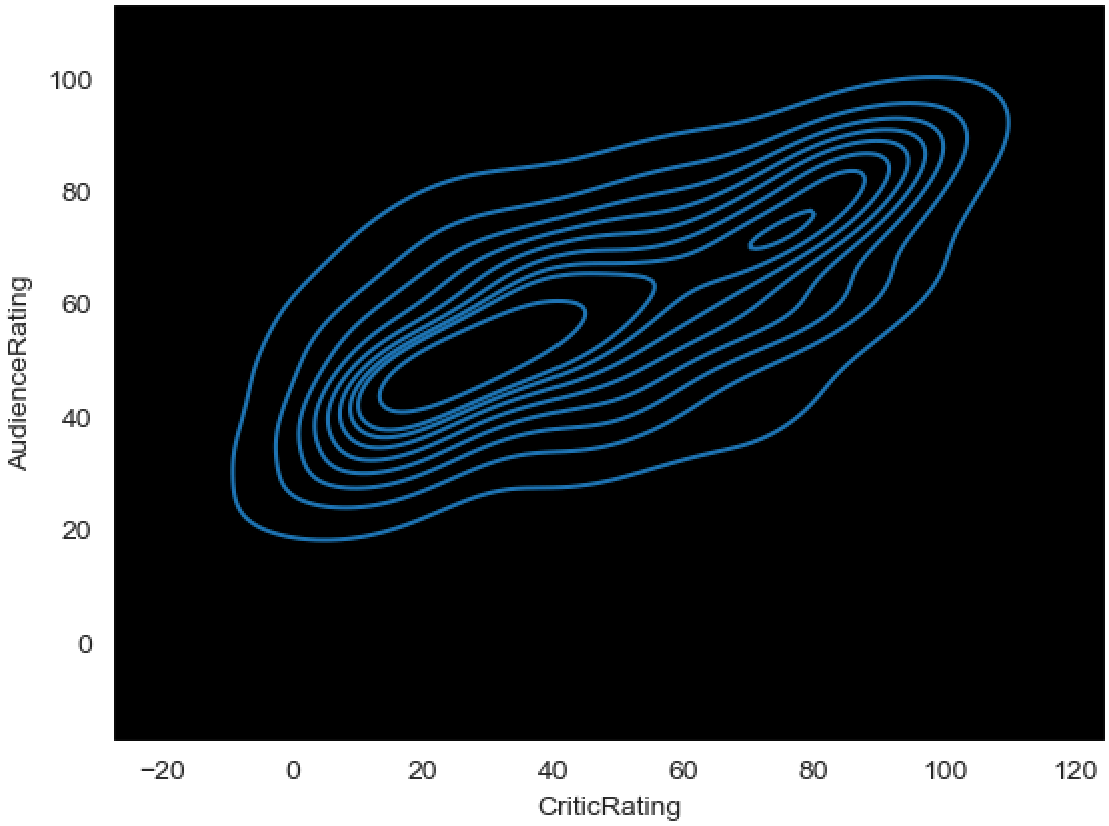


```
In [66]: vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',\
                          fit_reg = False, hue = 'Genre', height=5, aspect=1)
```



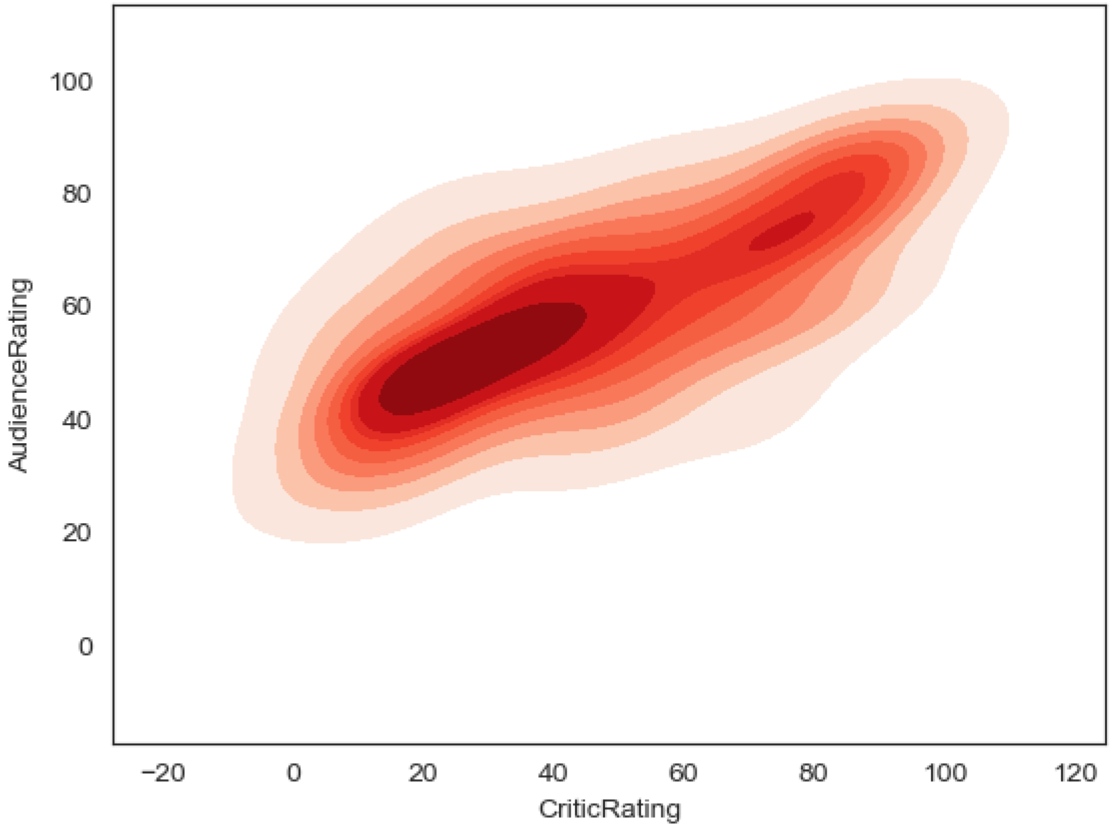
```
In [44]: # Kernal Density Estimate plot(KDE Plot)
         # how can i visualize audience rating & critics rating using scatterplot
```

```
In [72]: k1 = sns.kdeplot( data=movies,x=movies.CriticRating,y=movies.AudienceRating)
```



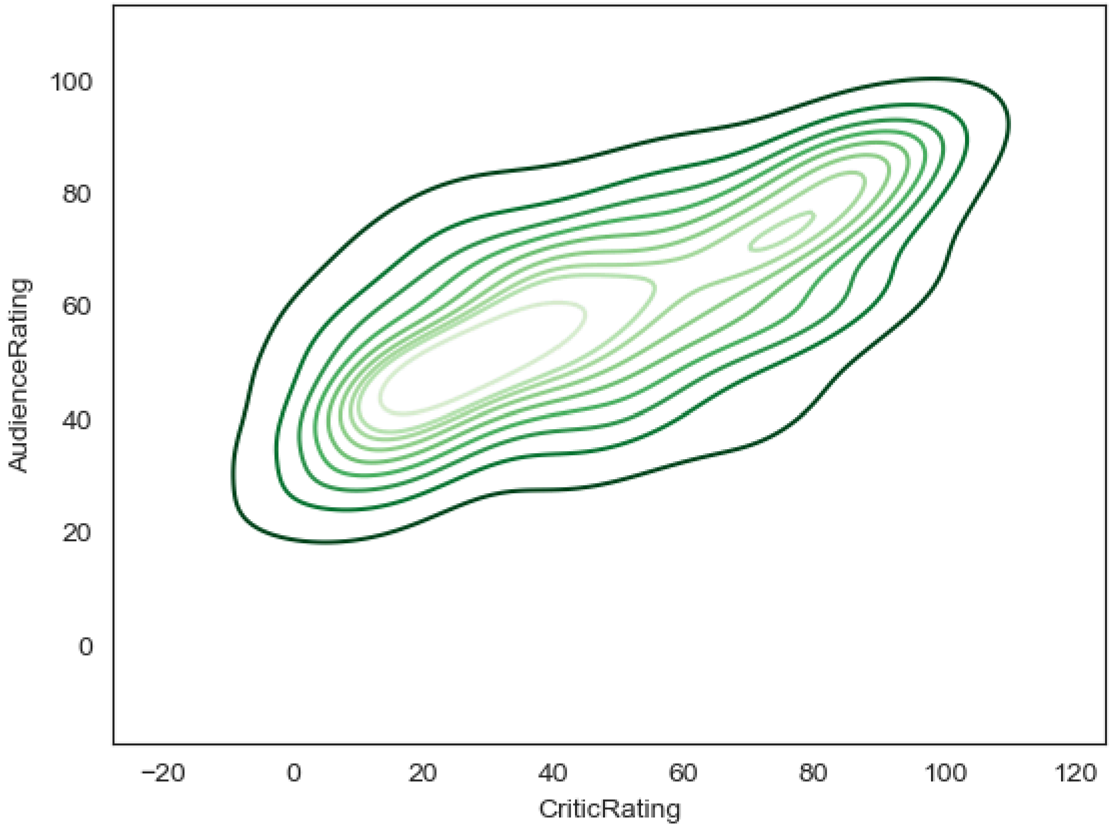
In [46]:

```
k1 = sns.kdeplot(data=movies,x=movies.CriticRating,y=movies.AudienceRating,shade
```



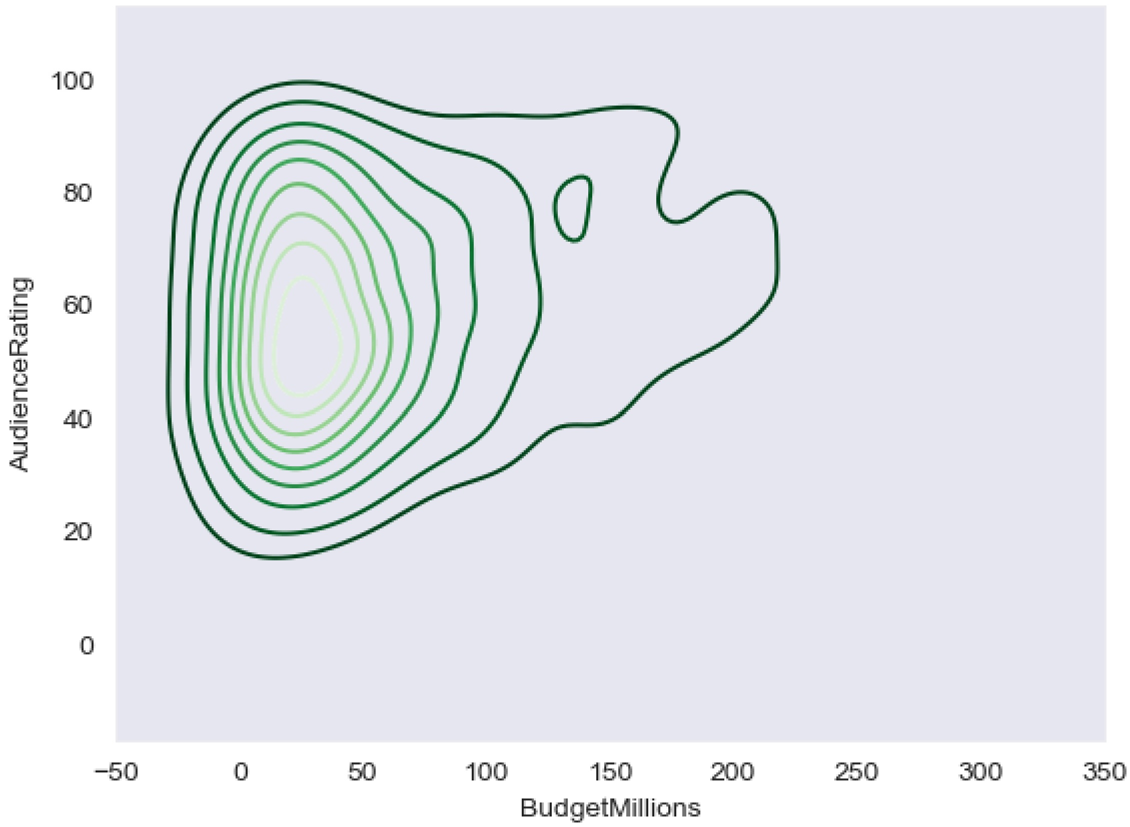
In [47]:

```
k2 = sns.kdeplot(data=movies,x=movies.CriticRating,y=movies.AudienceRating,shade
```

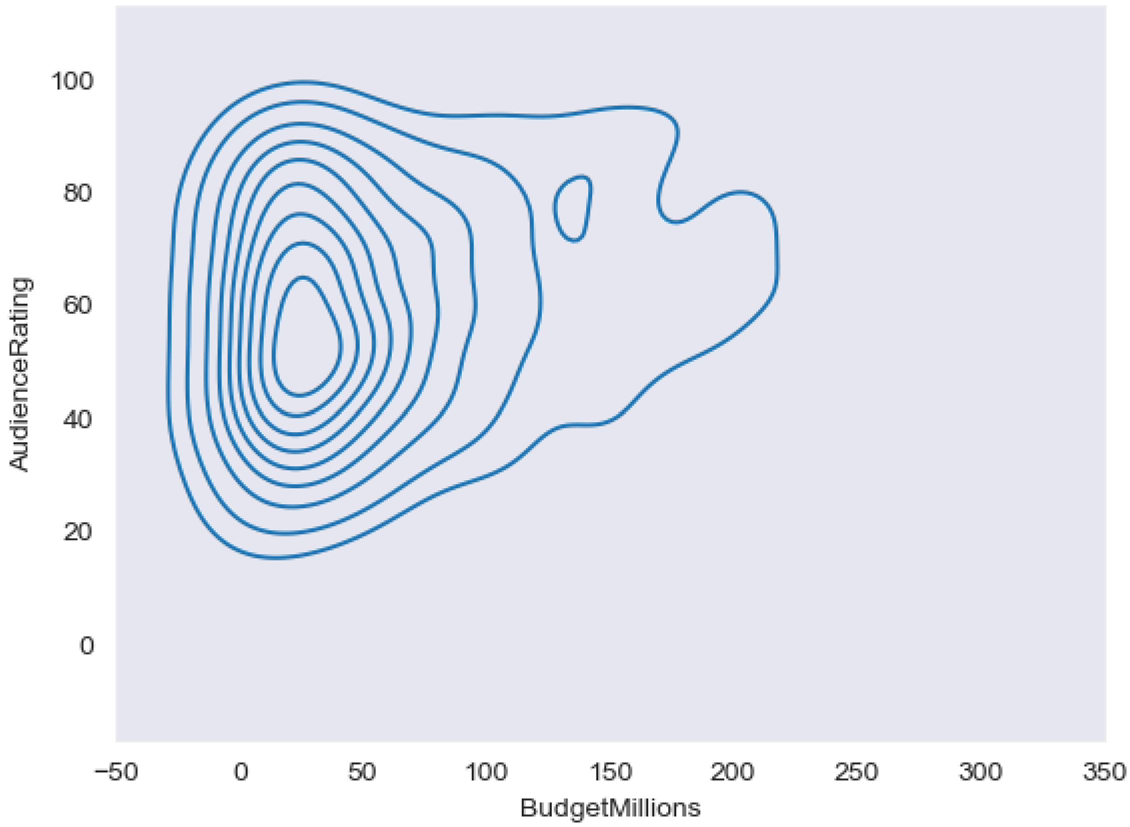


In [48]:

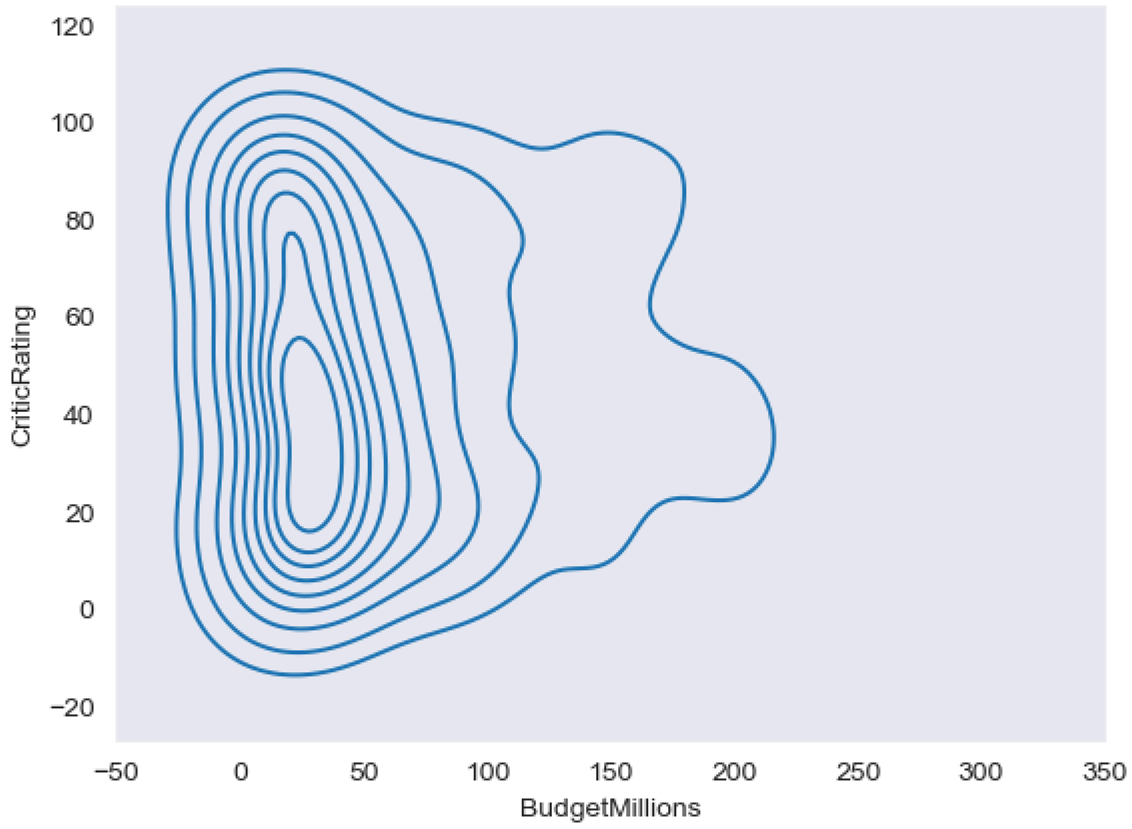
```
sns.set_style('dark')
k1 = sns.kdeplot(data=movies,x=movies.BudgetMillions,y=movies.AudienceRating,sha
```



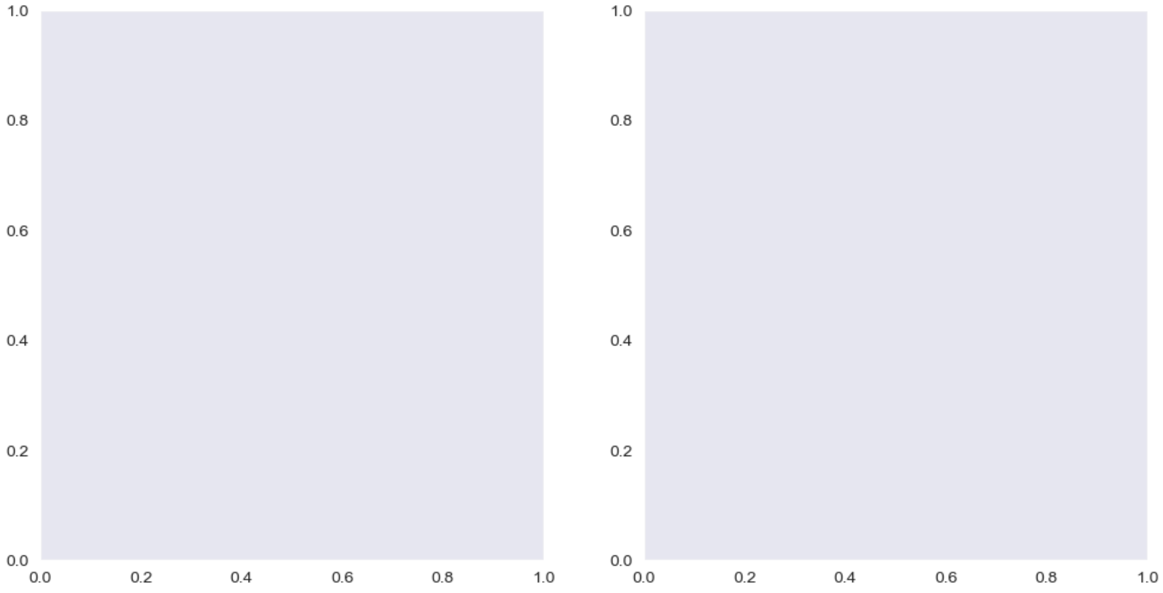
```
In [49]: sns.set_style('dark')
         k1 = sns.kdeplot(data=movies,x=movies.BudgetMillions, y=movies.AudienceRating)
```



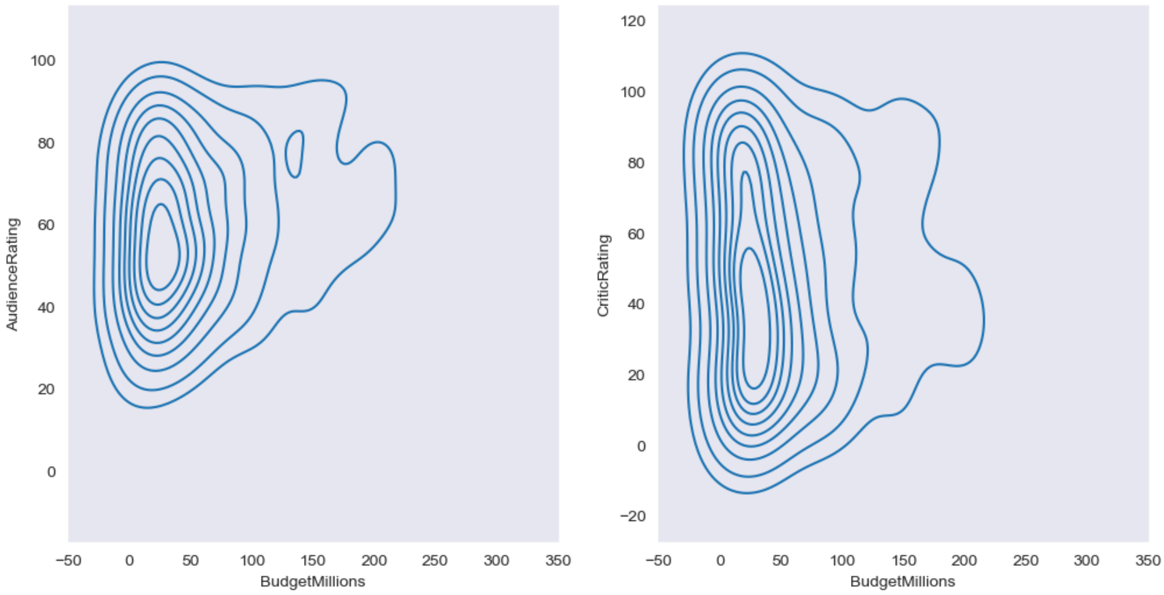
```
In [50]: k2=sns.kdeplot(data=movies, x='BudgetMillions', y='CriticRating', fill=False)
```



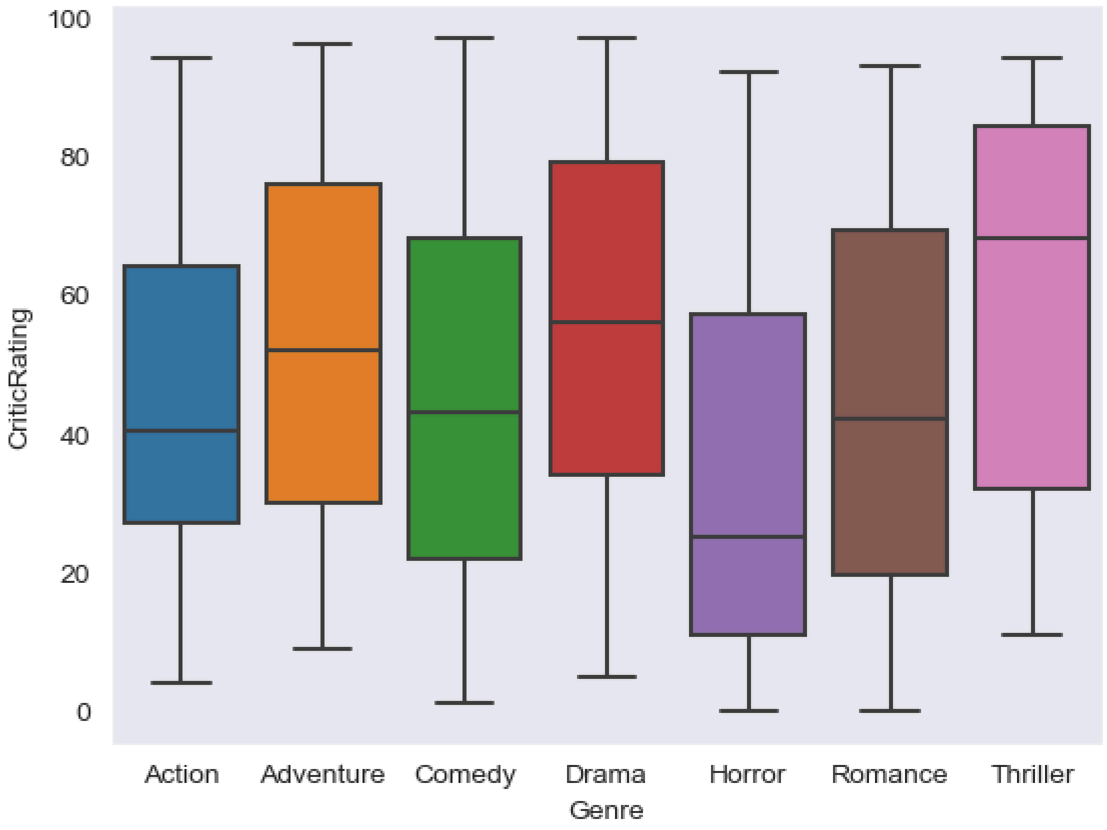
In [51]:
```python
# subplots

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



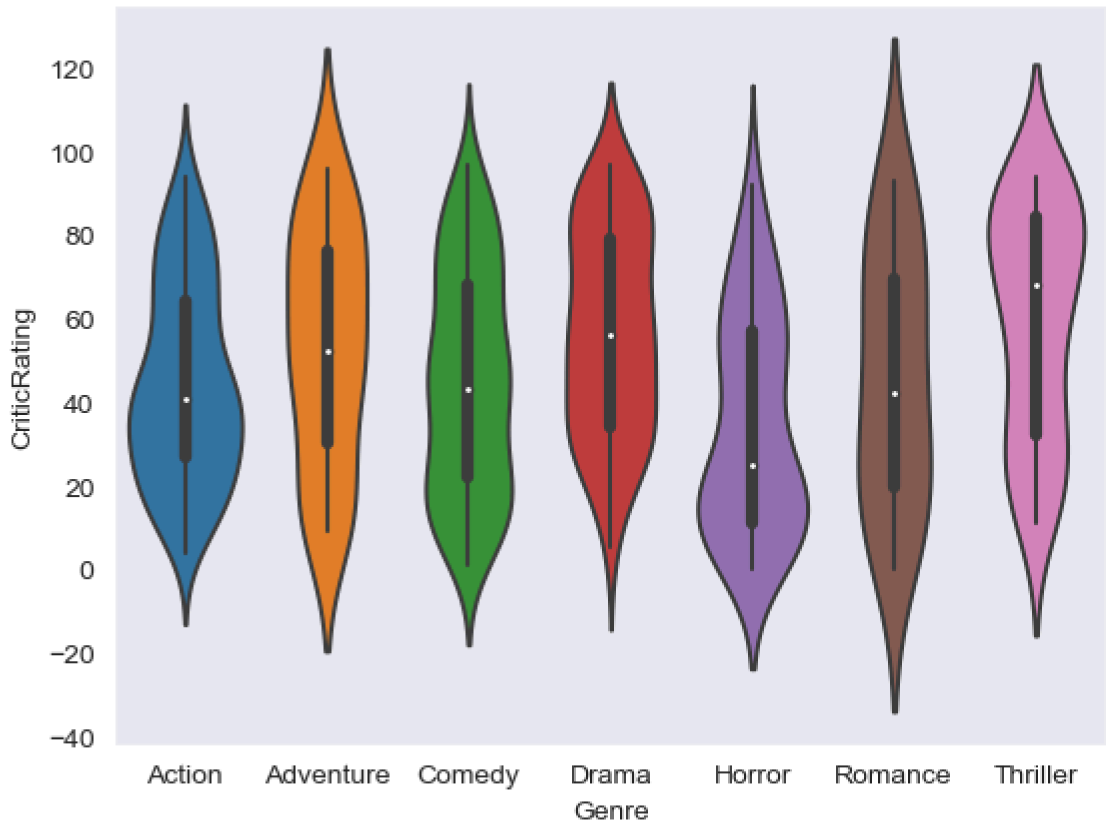
In [52]:
```python
f,ax = plt.subplots(1,2, figsize=(12,6))

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



In [53]:
```
# Box Plot

w = sns.boxplot(data=movies, x='Genre', y='CriticRating')
```

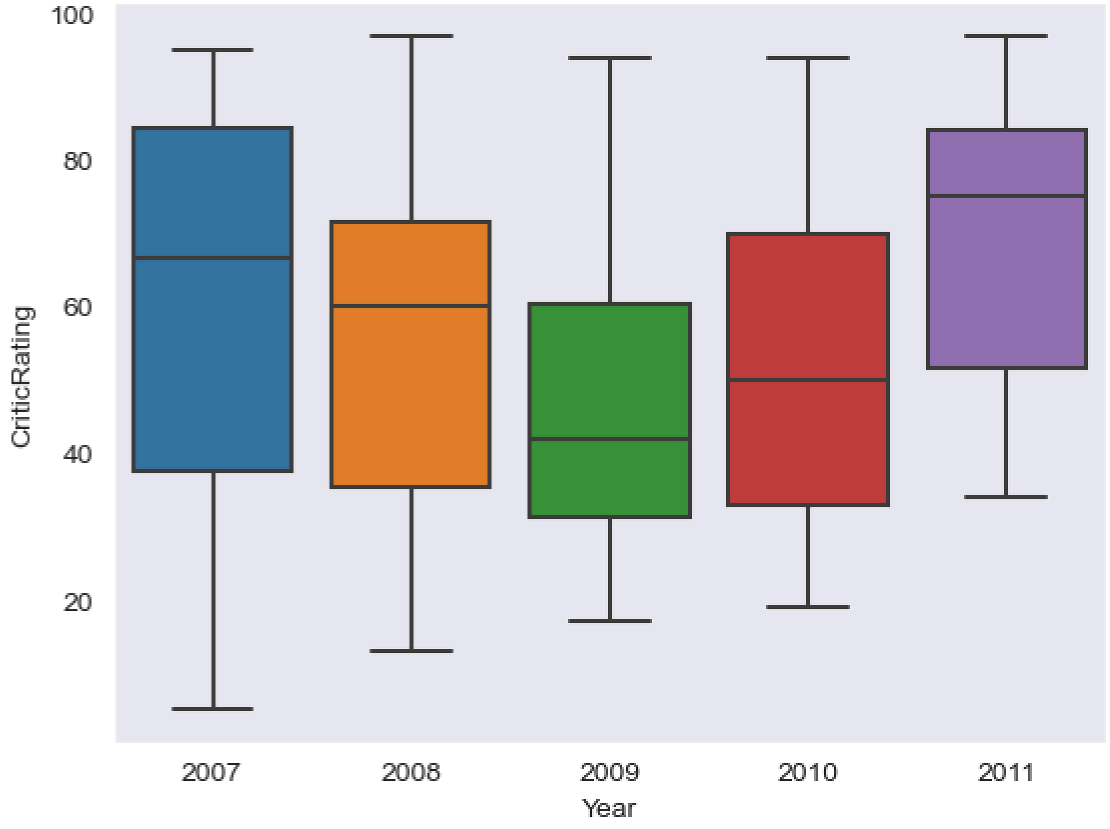


In [54]:
```
# violon plot

z= sns.violinplot(data=movies, x='Genre', y='CriticRating')
```



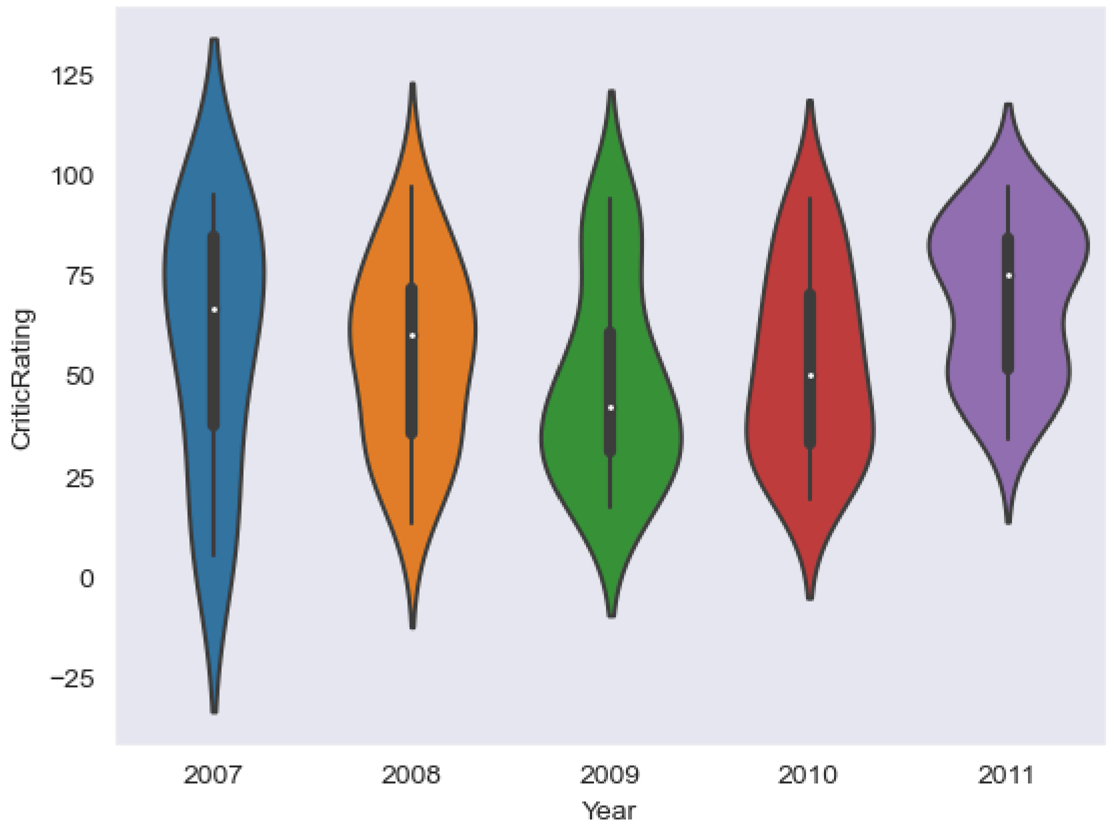
In [55]:

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



In [56]:

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

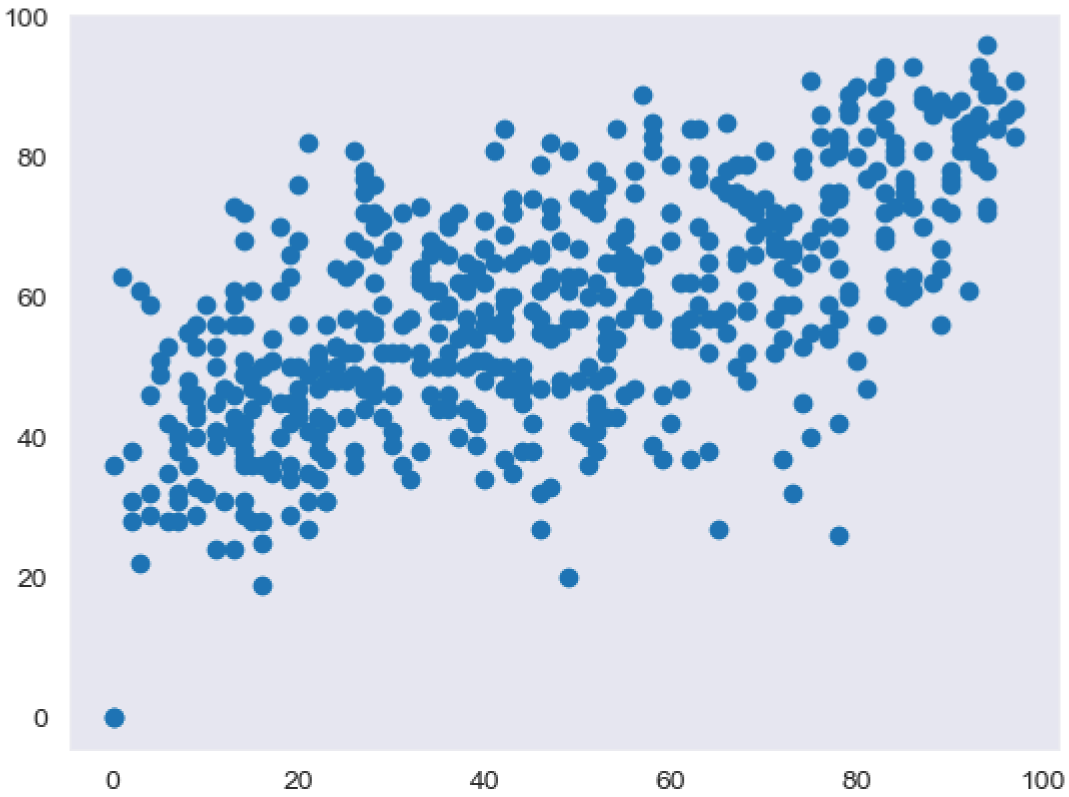


In [57]:
```
# creating a Facet grid

g = sns.FacetGrid(movies, row = 'Genre', col='Year',hue='Genre') # kind of subpl
```

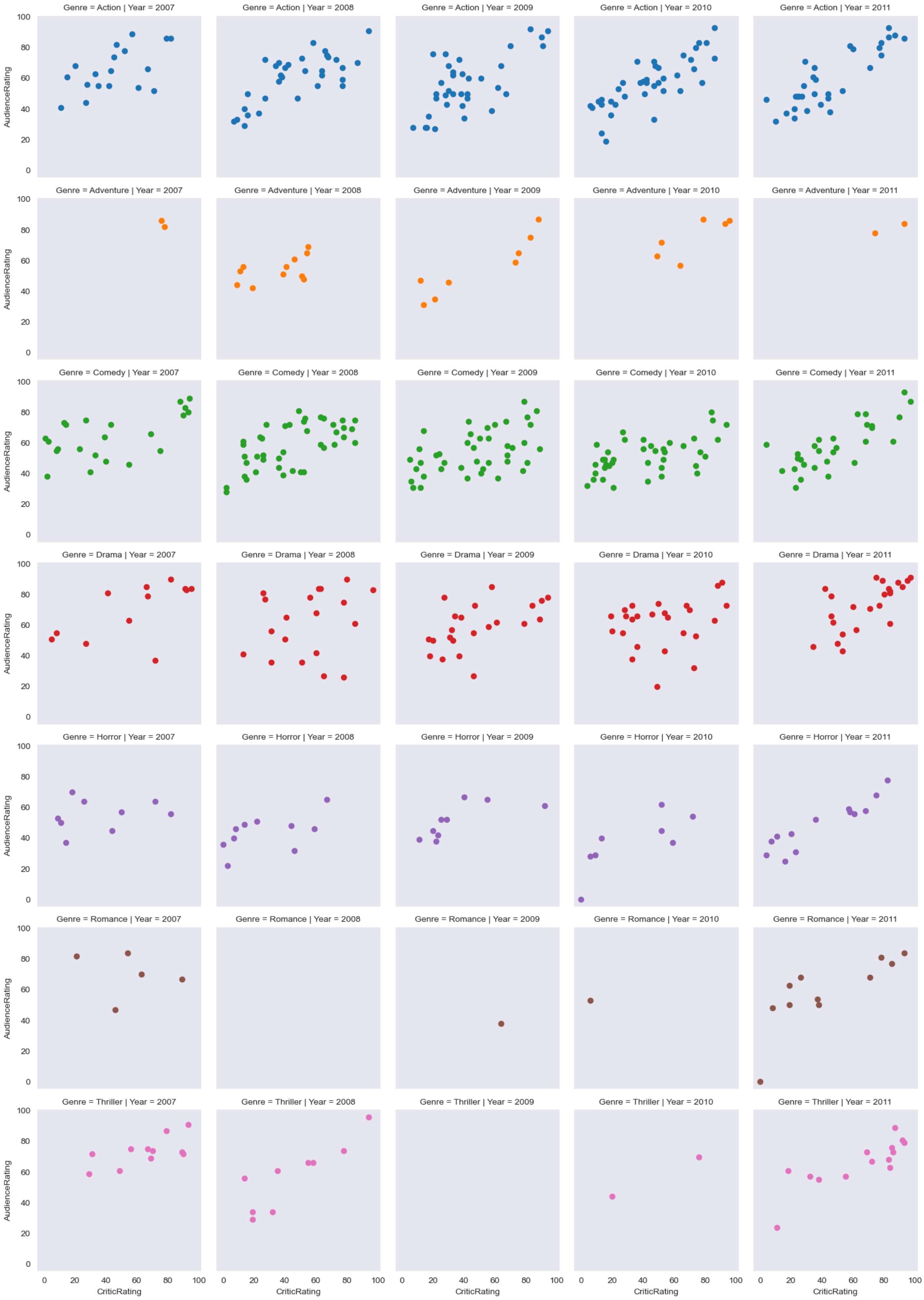
Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
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 [58]: plt.scatter(movies.CriticRating,movies.AudienceRating)
```

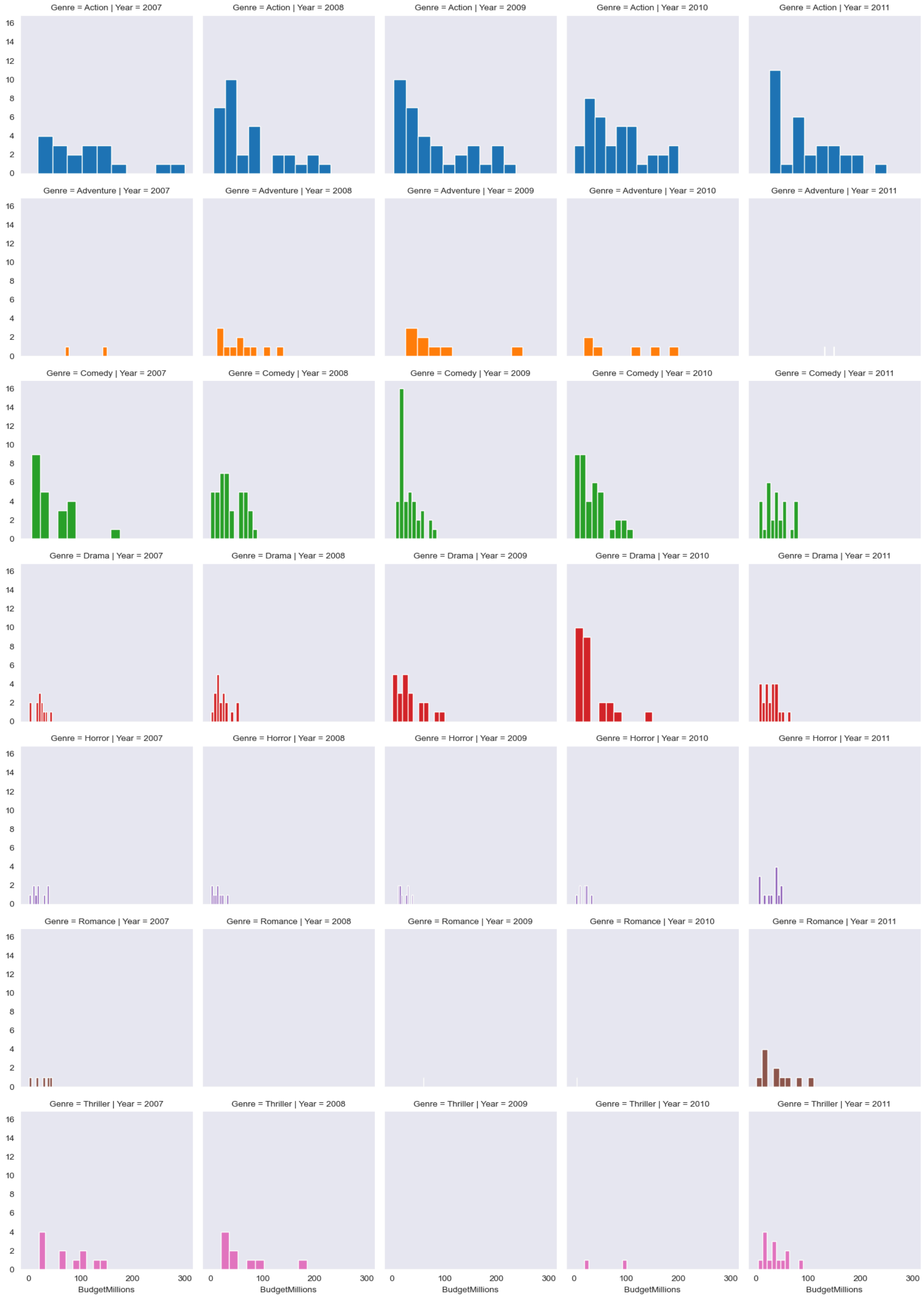
```
Out[58]: <matplotlib.collections.PathCollection at 0x26ce575ead0>
```



```
In [59]: g = sns.FacetGrid(movies, row = 'Genre', col='Year',hue='Genre')
         g = g.map(plt.scatter, 'CriticRating', 'AudienceRating')            # scatte
```

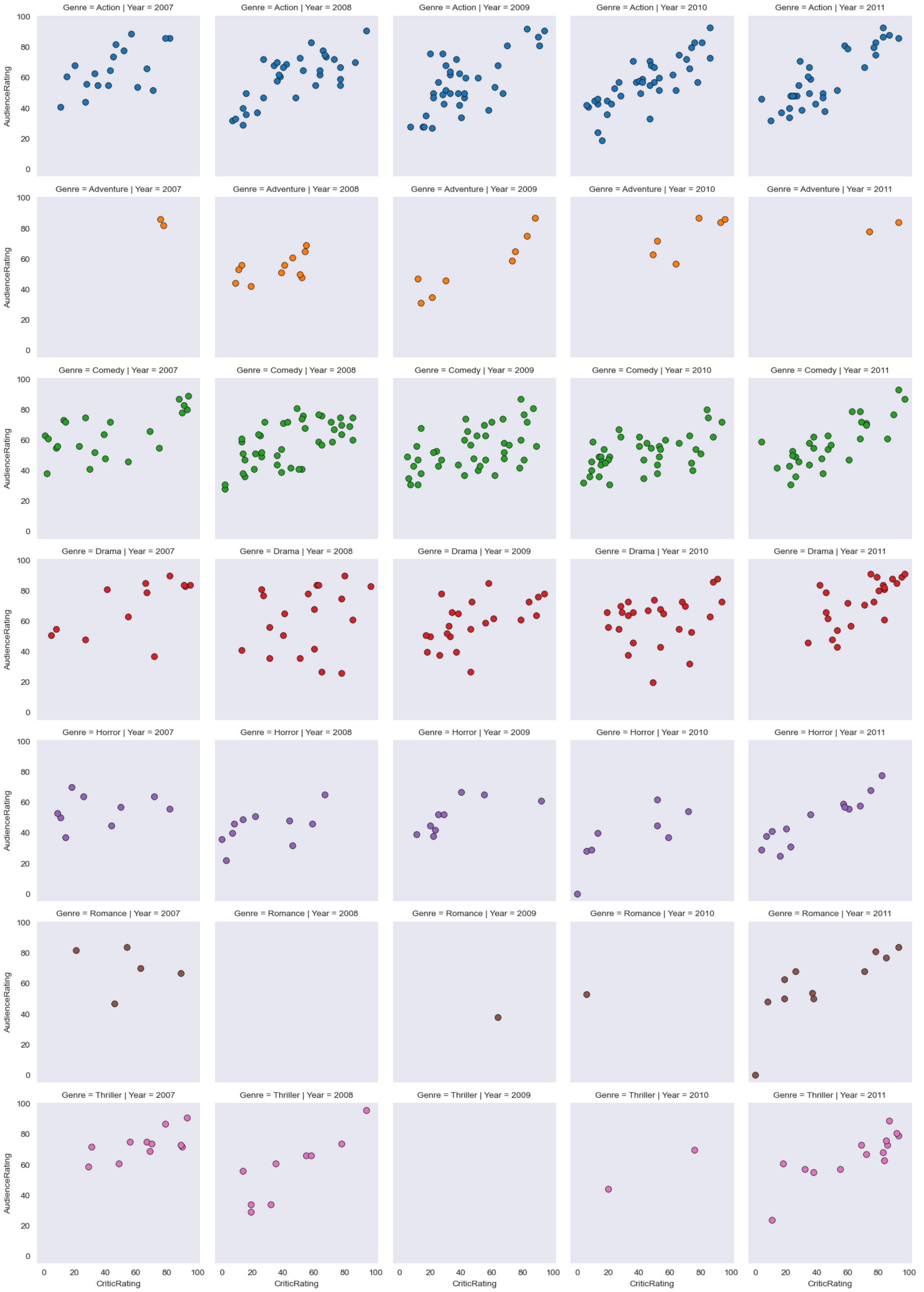


```
In [60]: # you can populated any type of chat

         g = sns.FacetGrid(movies, row = 'Genre', col='Year',hue='Genre')
         g = g.map(plt.hist, 'BudgetMillions')   # scatterplots are mapped in facetgrid
```



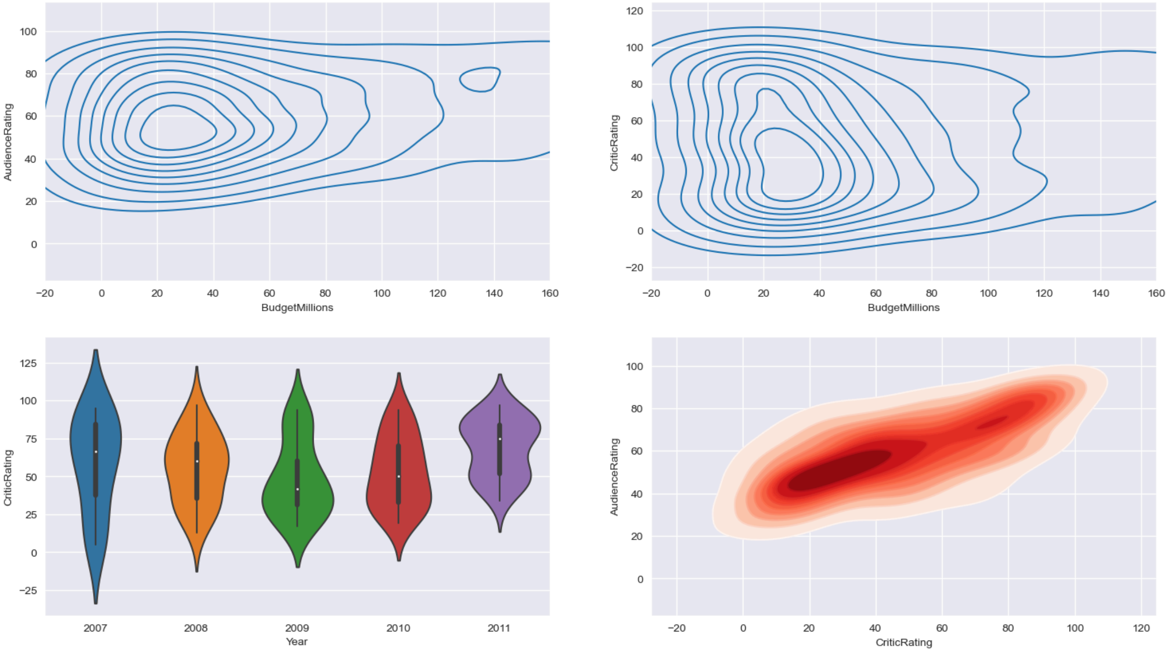
In [61]:

```
#
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)        # scatter
```



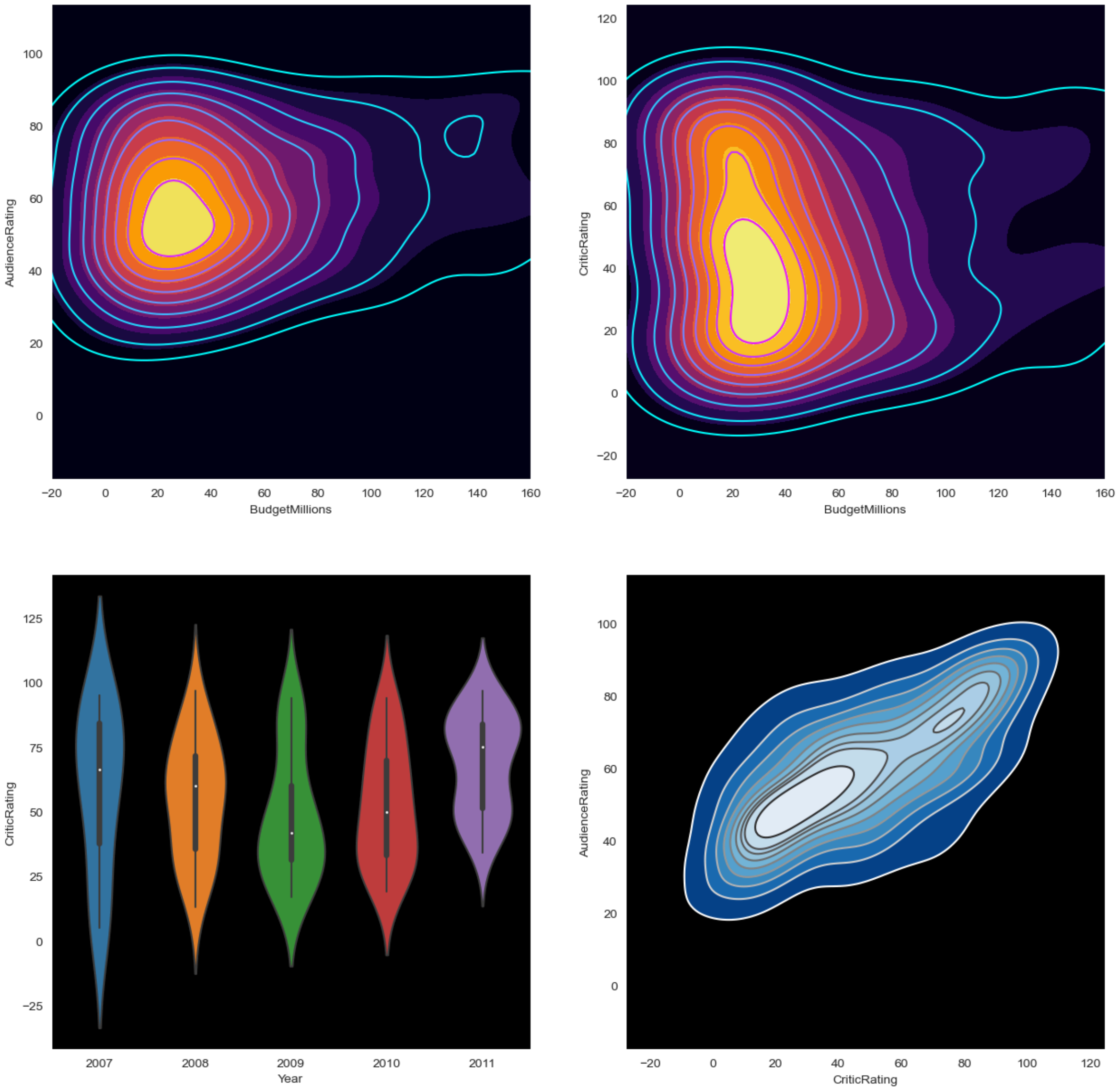
Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
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

```
In [62]: sns.set_style('darkgrid')
         f, axes = plt.subplots (2,2, figsize = (18,10))

         k1 = sns.kdeplot(data=movies,x=movies.BudgetMillions,y=movies.AudienceRating,ax=
         k2 = sns.kdeplot(data=movies,x=movies.BudgetMillions,y=movies.CriticRating,ax = 

         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(data=movies,x=movies.CriticRating,y=movies.AudienceRating,shade

         k4b = sns.kdeplot(data=movies,x=movies.CriticRating, y=movies.AudienceRating,cma

         plt.show()
```



In [63]:
```
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, \
                 fill=True,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 [ ]:
```