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

In [2]:
```
pd.__version__
```

Out[2]:
```
'2.2.2'
```

In [3]:
```
import os
os.getcwd()
```

Out[3]:
```
'C:\\Users\\jayes'
```

In [4]:
```
movies = pd.read_csv(r"C:\Users\jayes\OneDrive\Desktop\NareshIT\28_mar\28th - Iris, movie analytics Project\28th -Sea
```

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]:
```
movies.head()
```

Out[7]:

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

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

Out[8]:

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

In [9]:
```
movies.columns
```

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

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

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

Out[11]:

	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 [12]:
```
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 [13]:
```
movies.describe()
```

Out[13]:

	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 [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
```

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

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

In [17]:
```
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 [36]:
```
movies.Genre = movies.Genre.astype('category')
movies.Year = movies.Year.astype('category')                # mean, median of year don't make sense
```

In [38]:
```
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 [40]:
```
movies.Genre.cat.categories
```

Out[40]:
```
Index(['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance',
       'Thriller'],
      dtype='object')
```

In [42]:
```
movies.describe()
```

Out[42]:

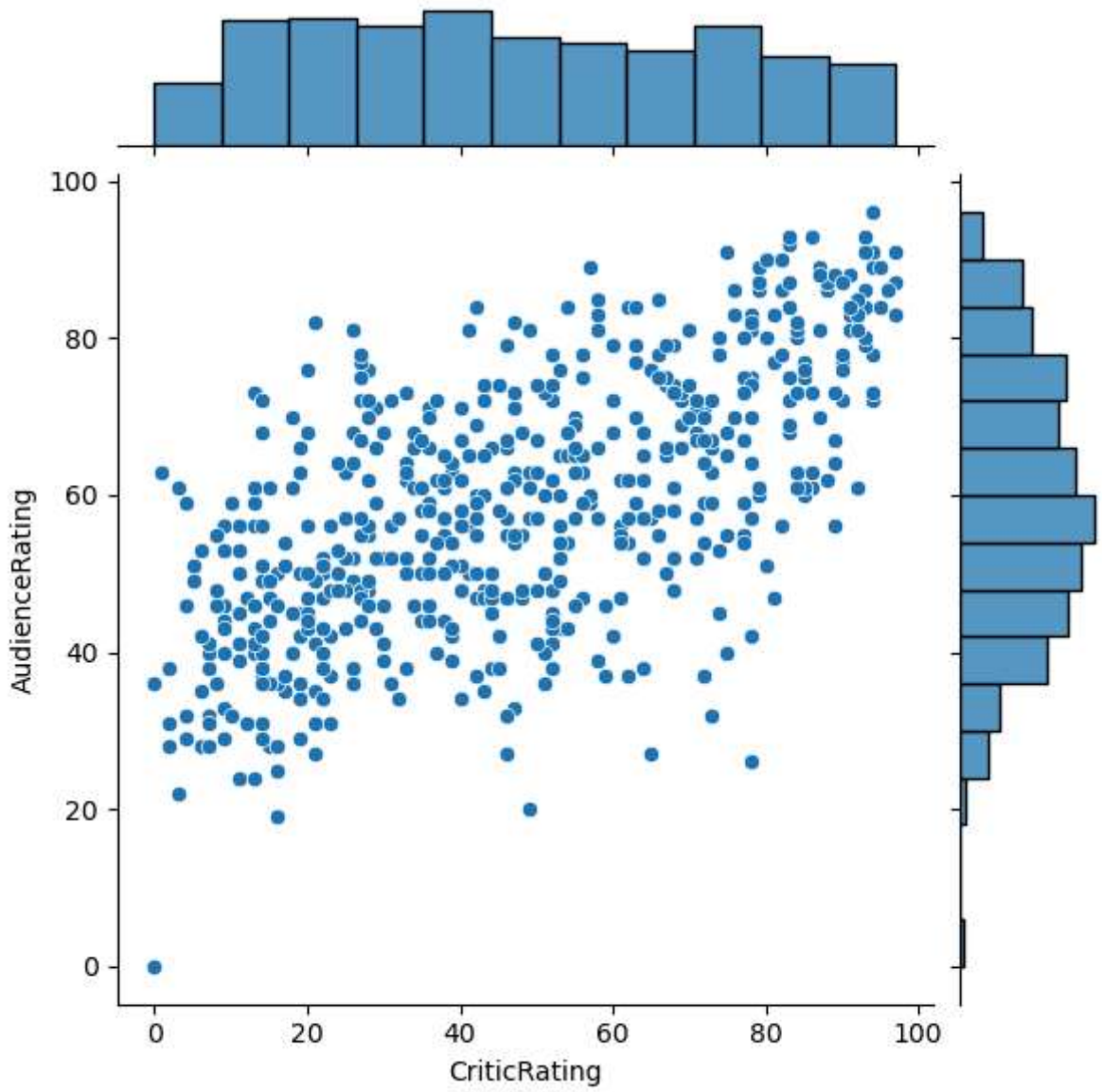
	CriticRating	AudienceRating	BudgetMillions
count	559.000000	559.000000	559.000000
mean	47.309481	58.744186	50.236136
std	26.413091	16.826887	48.731817
min	0.000000	0.000000	0.000000
25%	25.000000	47.000000	20.000000
50%	46.000000	58.000000	35.000000
75%	70.000000	72.000000	65.000000
max	97.000000	96.000000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

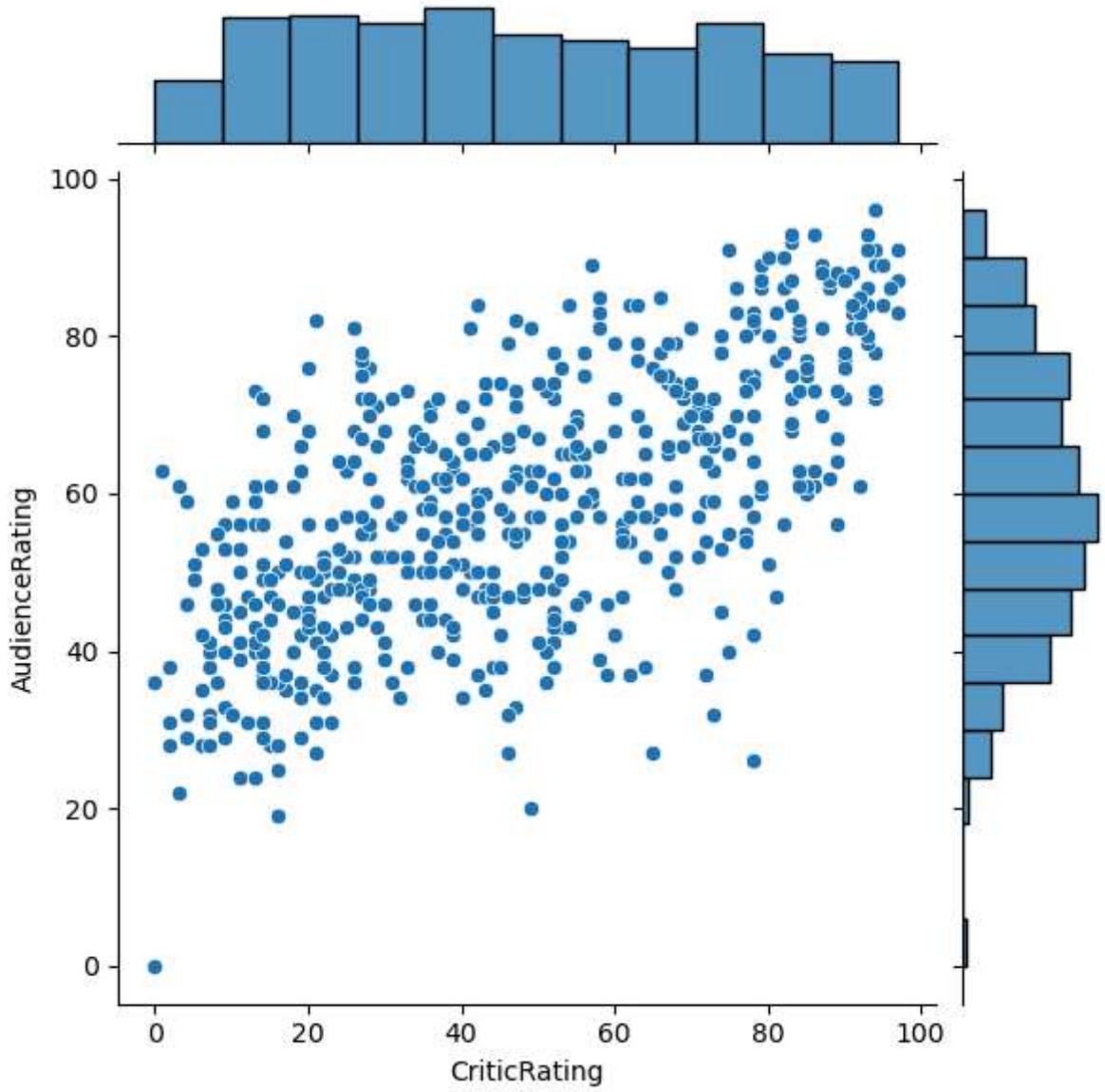
In [44]:
```
# How to working with joint plots

from matplotlib import pyplot as plt
```

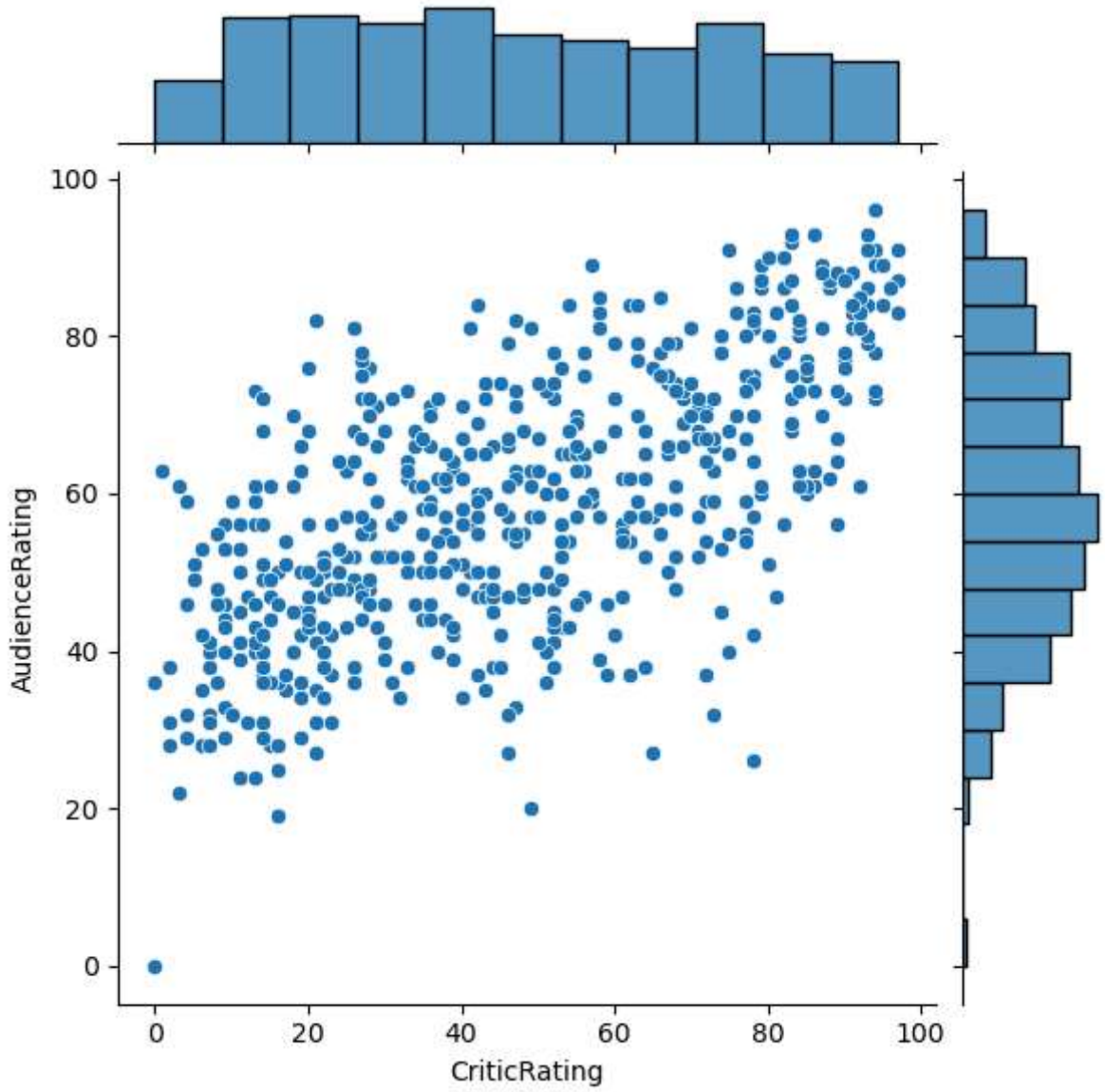
```
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

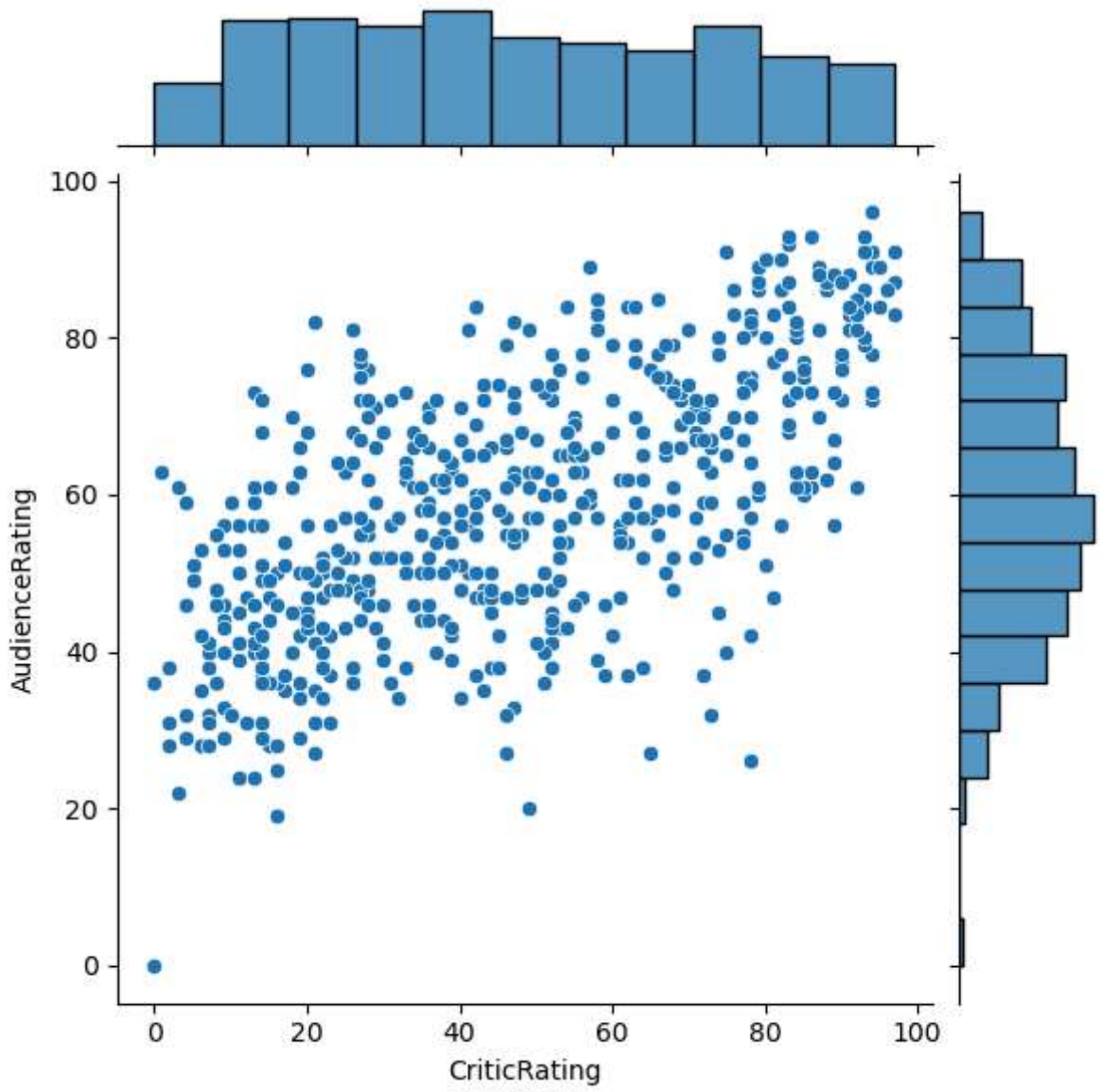
- basically joint plot is a scatter plot & it find the relation b/w audiene & critics
- also if you look up you can find the uniform distribution (critics)and normal distriution (audience)

In [55]:
```
j = sns.jointplot( data = movies, x = 'CriticRating', y = 'AudienceRating')
plt.show()
```

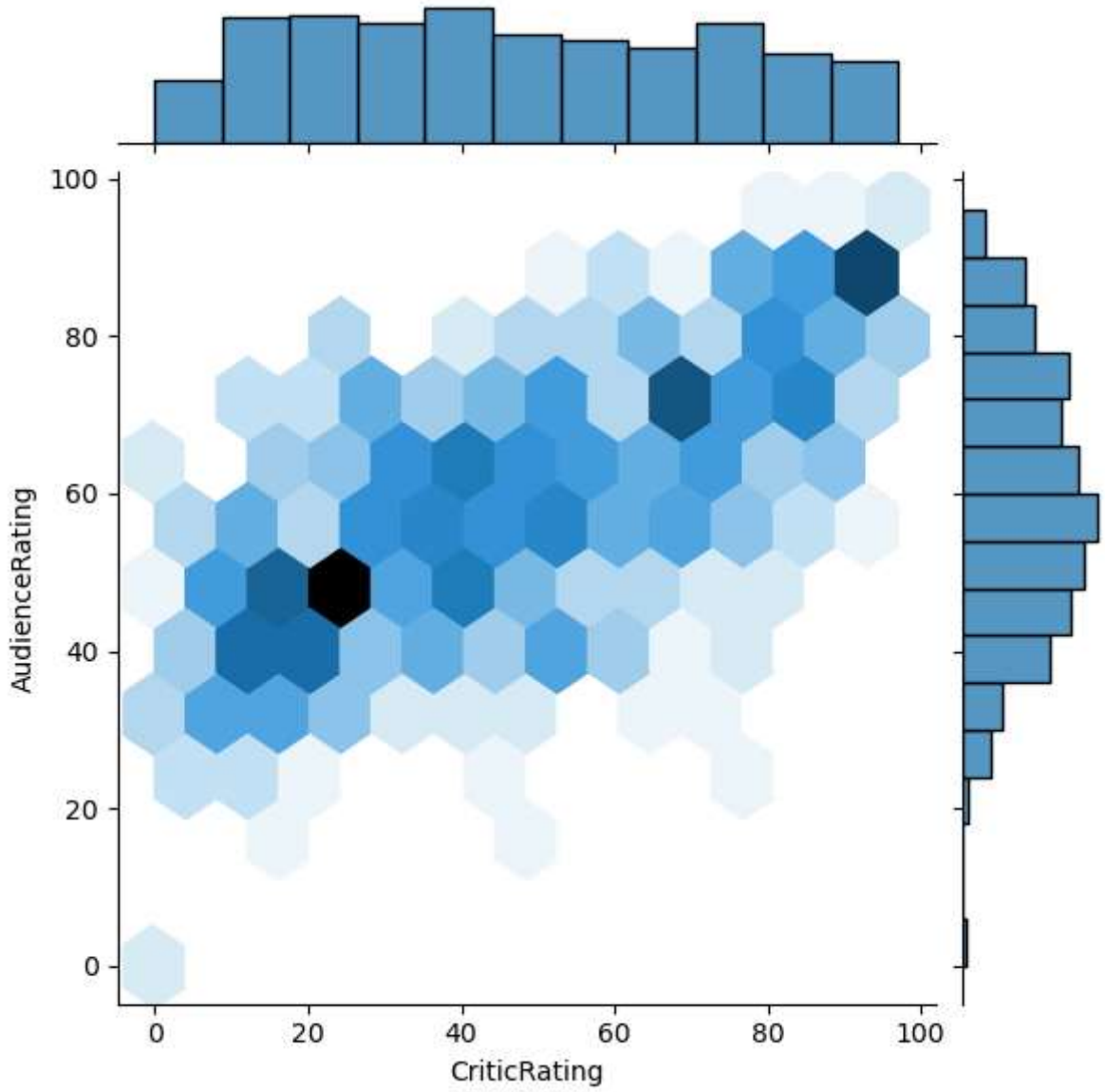

AudienceRating
CriticRating
0
20
40
60
80
100

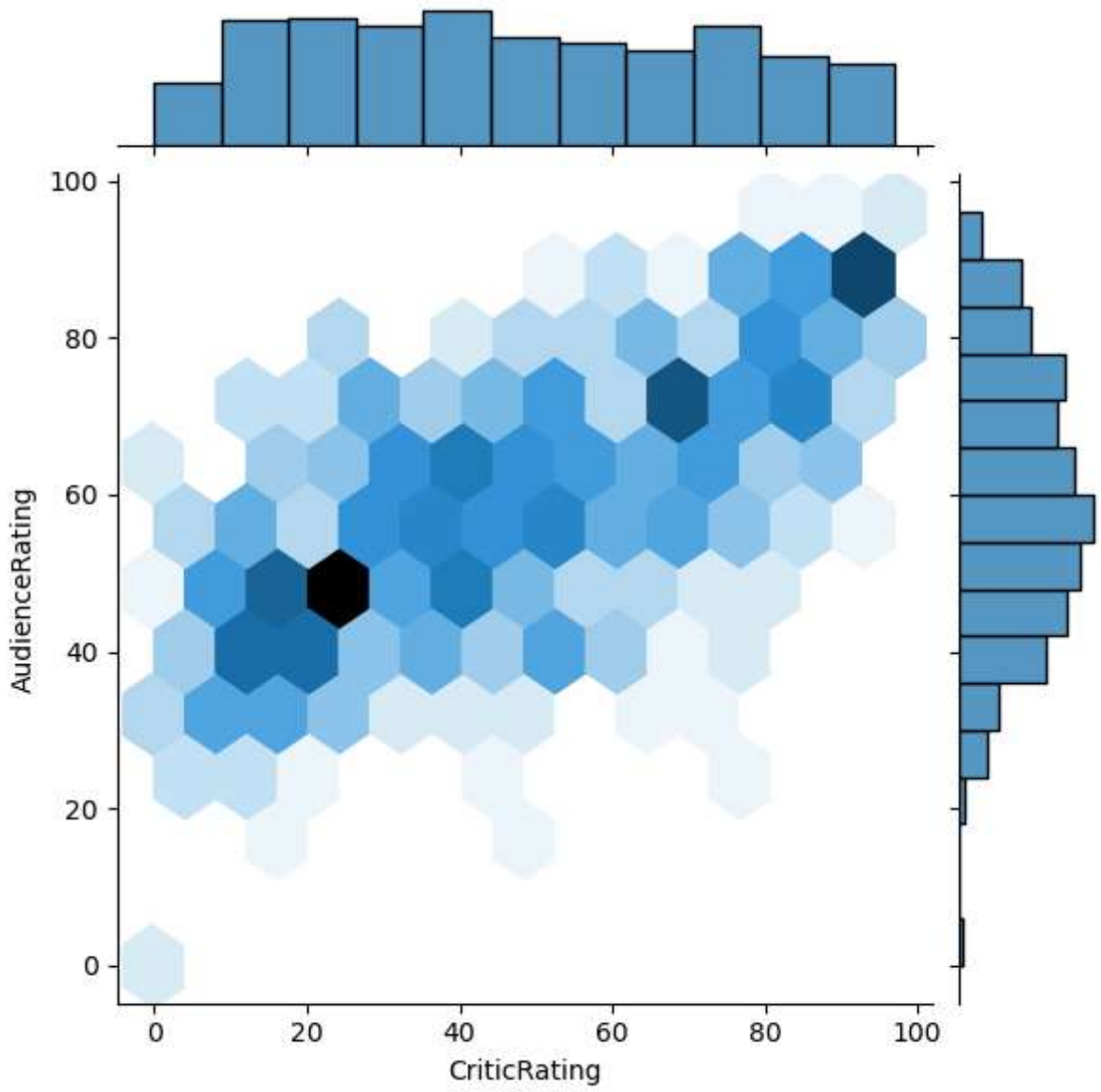

100
80
60
40
20
0
AudienceRating
0
20
40
60
80
100
CriticRating


AudienceRating
CriticRating
0
20
40
60
80
100



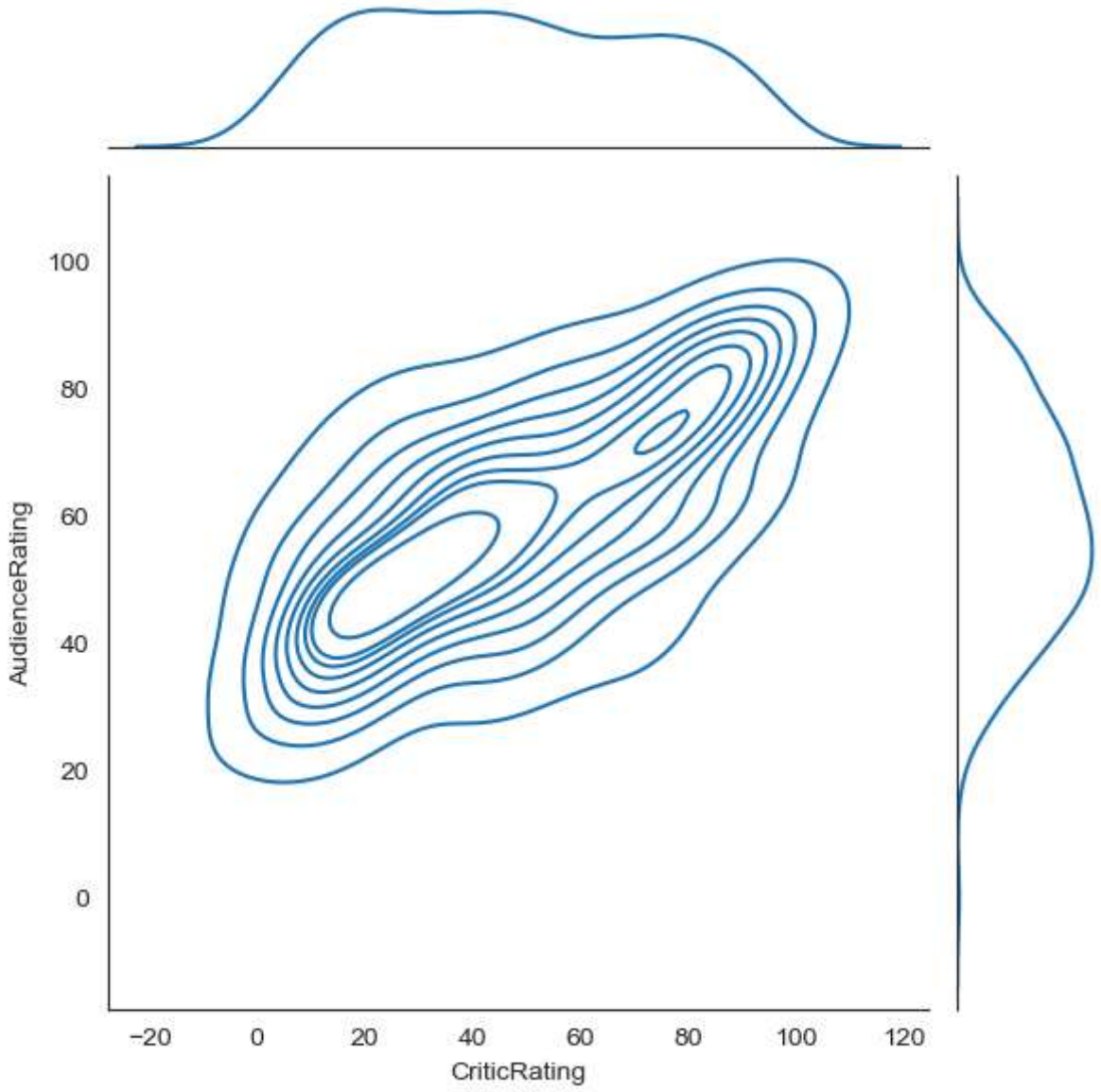
In [59]:
```
j = sns.jointplot( data = movies, x = 'CriticRating', y = 'AudienceRating', kind='hex')
plt.show()
```


100
80
60
40
20
0
AudienceRating
0
20
40
60
80
100
CriticRating

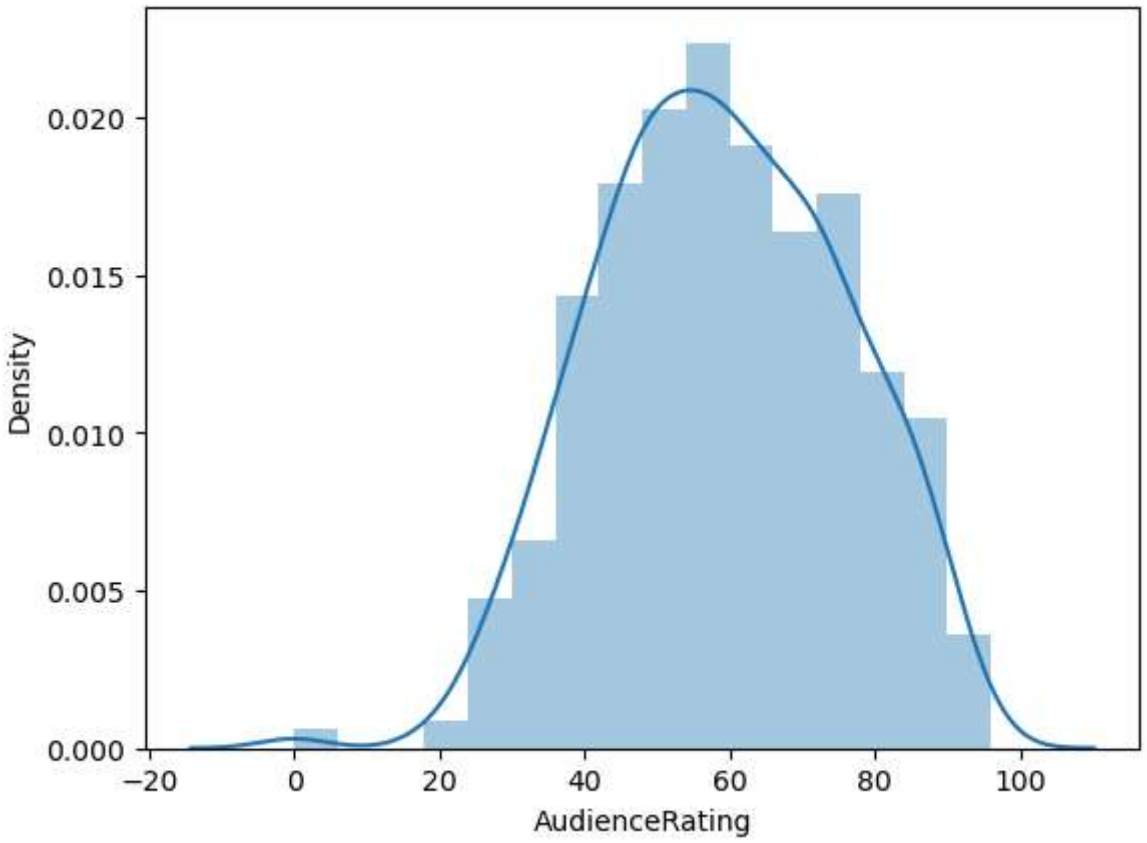


In [113…

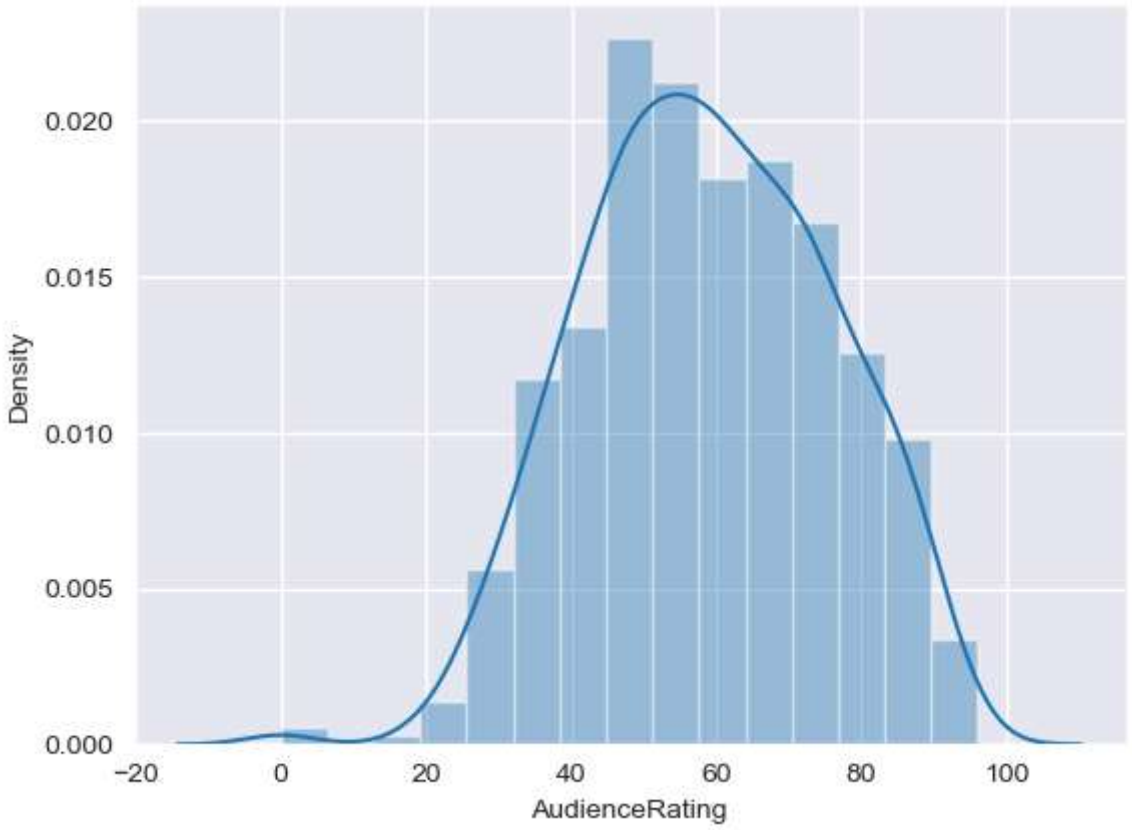
```
j = sns.jointplot( data = movies, x = 'CriticRating', y = 'AudienceRating', kind='kde')
plt.show()
```



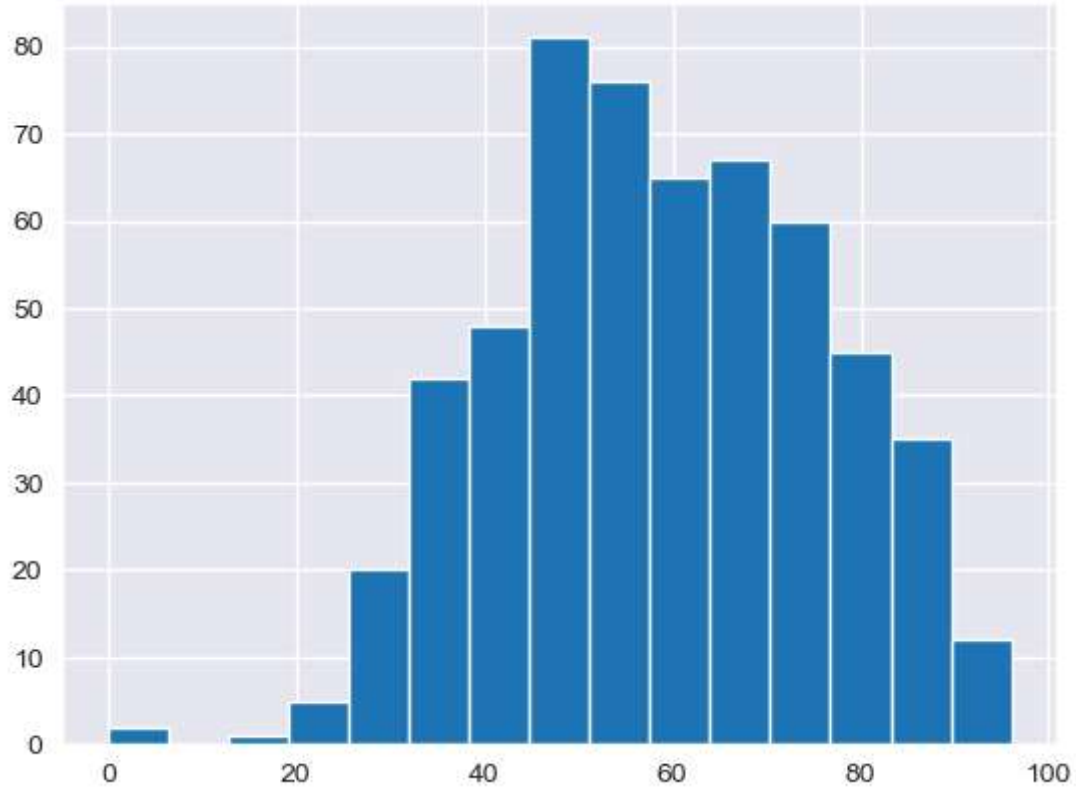
In [65]:
```
m1 = sns.distplot(movies.AudienceRating)
plt.show()

# y - axis generated by seaborn automatically that is the powefull of seaborn gallery
```



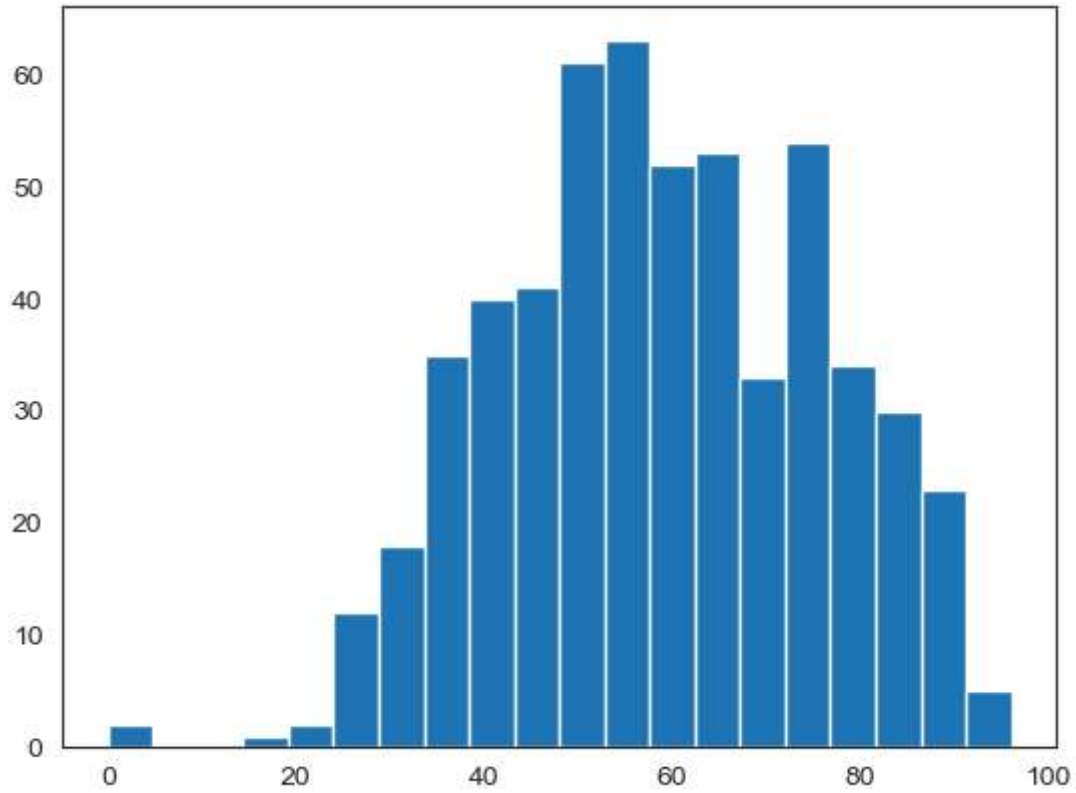
In [67]:
```
sns.set_style('darkgrid')
m2 = sns.distplot(movies.AudienceRating, bins = 15)
plt.show()
```



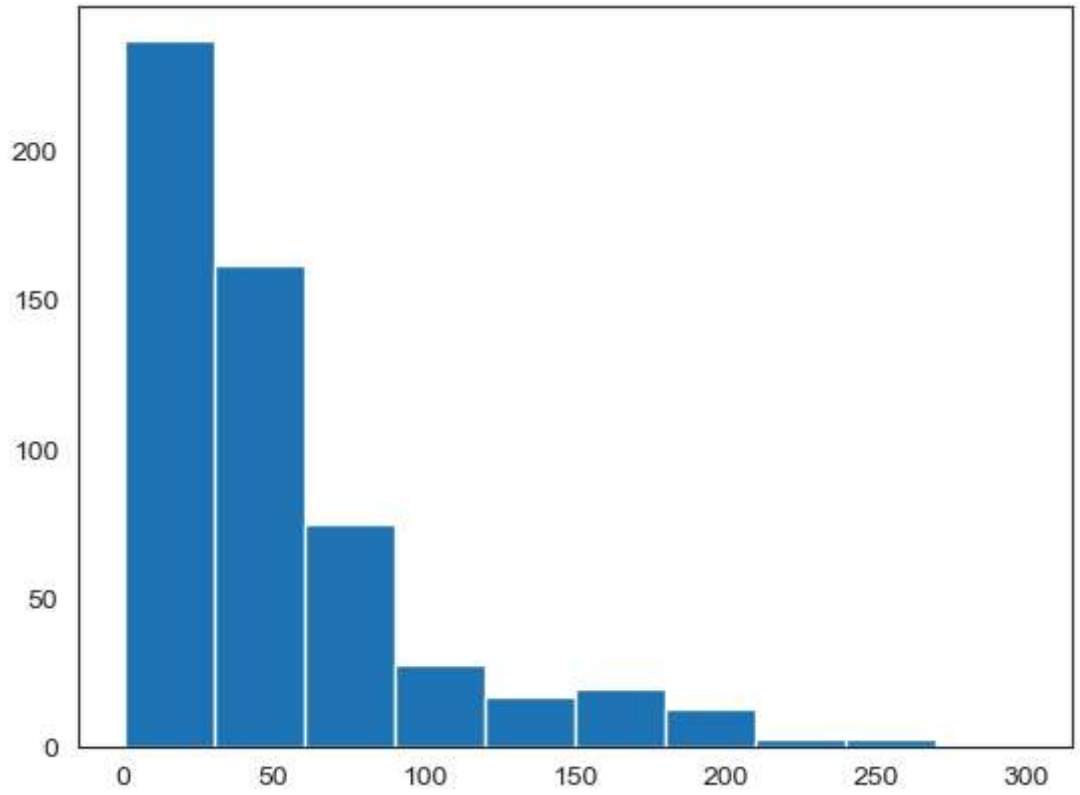
In [69]:
```
n1 = plt.hist(movies.AudienceRating, bins=15)
plt.show()
```



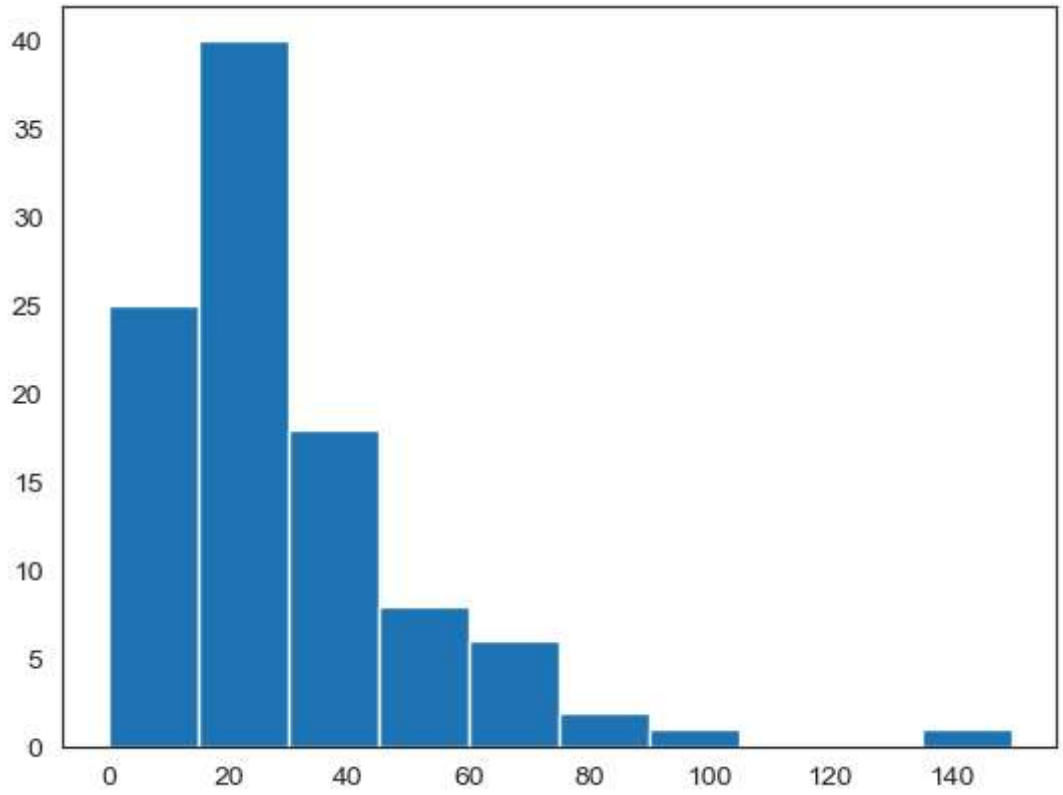
In [71]:
```
sns.set_style('white') #normal distribution & called as bell curve
n1 = plt.hist(movies.AudienceRating, bins=20)
plt.show()
```



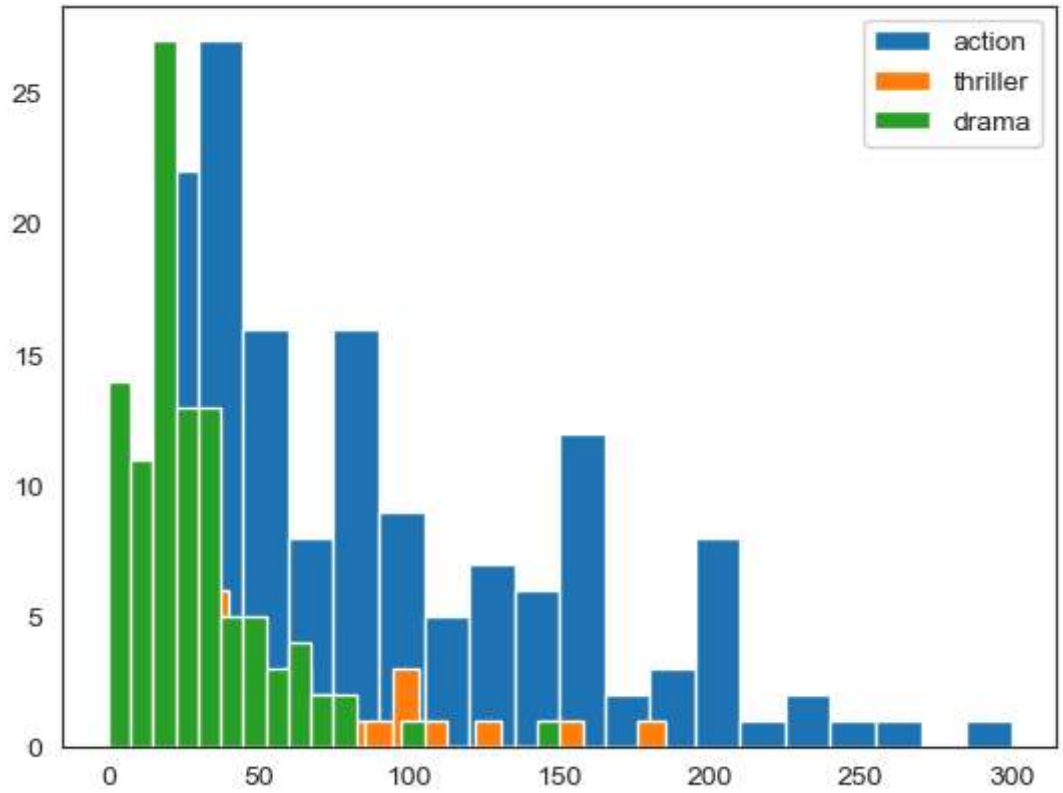
In [73]:
```
plt.hist(movies.BudgetMillions)
plt.show()
```



In [75]:
```
plt.hist(movies[movies.Genre == 'Drama'].BudgetMillions)
plt.show()
```

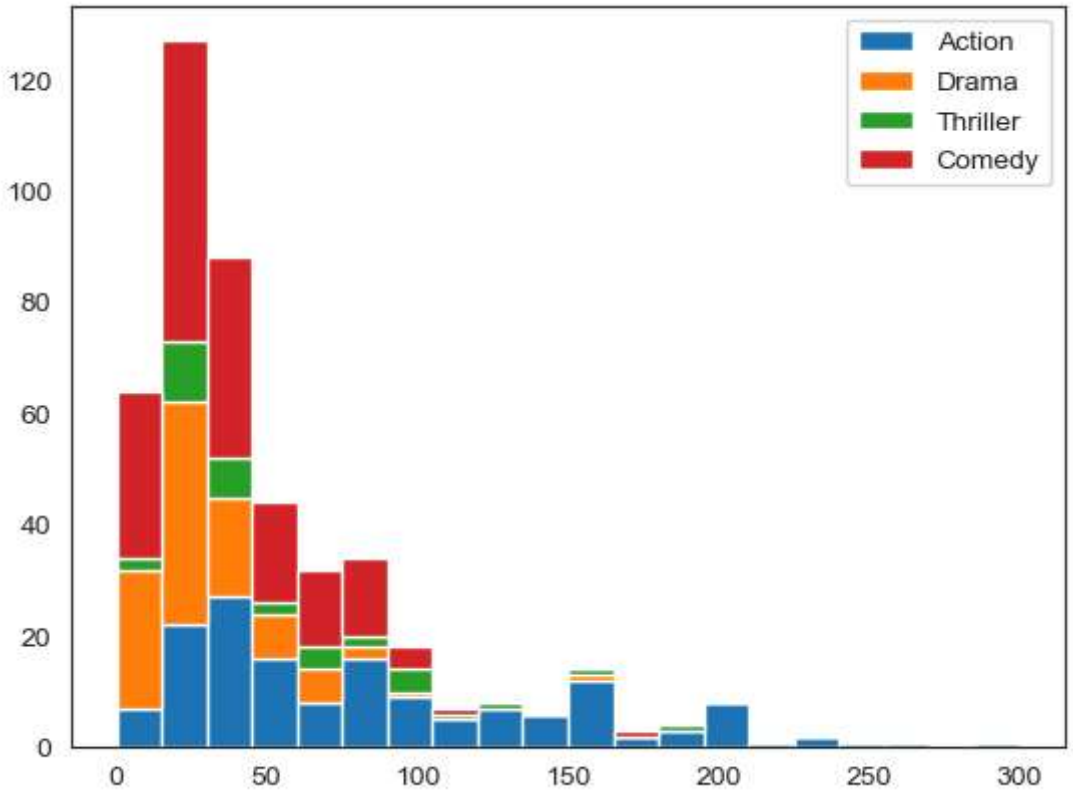


In [79]:
```
plt.hist(movies[movies.Genre == 'Action'].BudgetMillions, bins = 20, label='action')
plt.hist(movies[movies.Genre == 'Thriller'].BudgetMillions, bins = 20, label='thriller')
plt.hist(movies[movies.Genre == 'Drama'].BudgetMillions, bins = 20, label='drama')
plt.legend()
plt.show()
```



In [101…

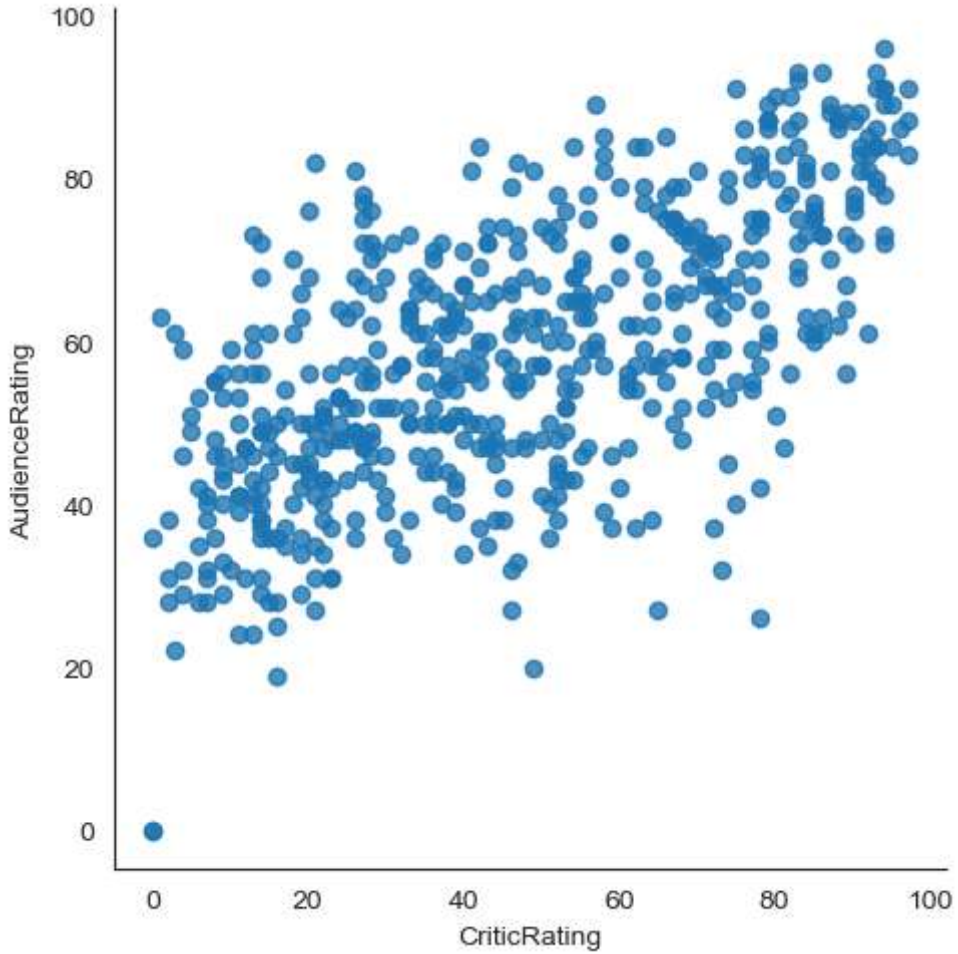
```
plt.hist([movies[movies.Genre == 'Action'].BudgetMillions,\
          movies[movies.Genre == 'Drama'].BudgetMillions,\
          movies[movies.Genre == 'Thriller'].BudgetMillions,\
          movies[movies.Genre == 'Comedy'].BudgetMillions],
         bins = 20, stacked = True)
plt.legend(['Action',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

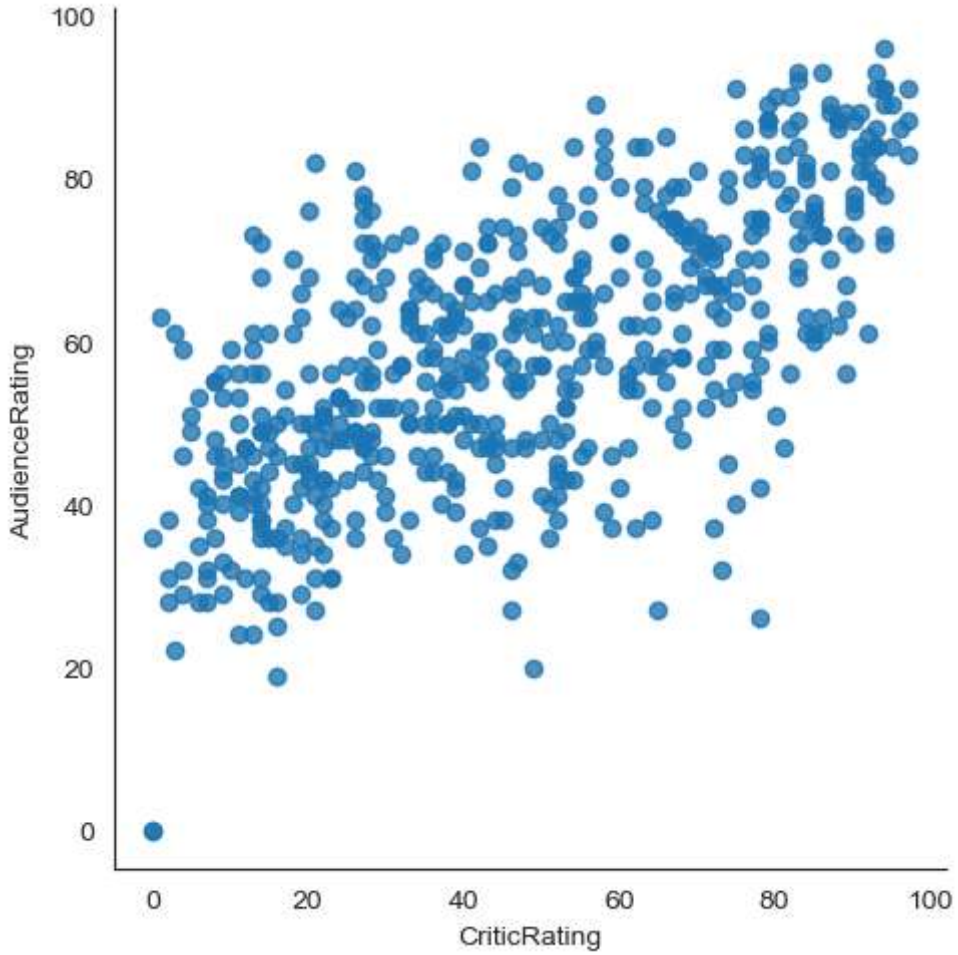


In [103…
```
for gen in movies.Genre.cat.categories:
    print(gen)
```

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

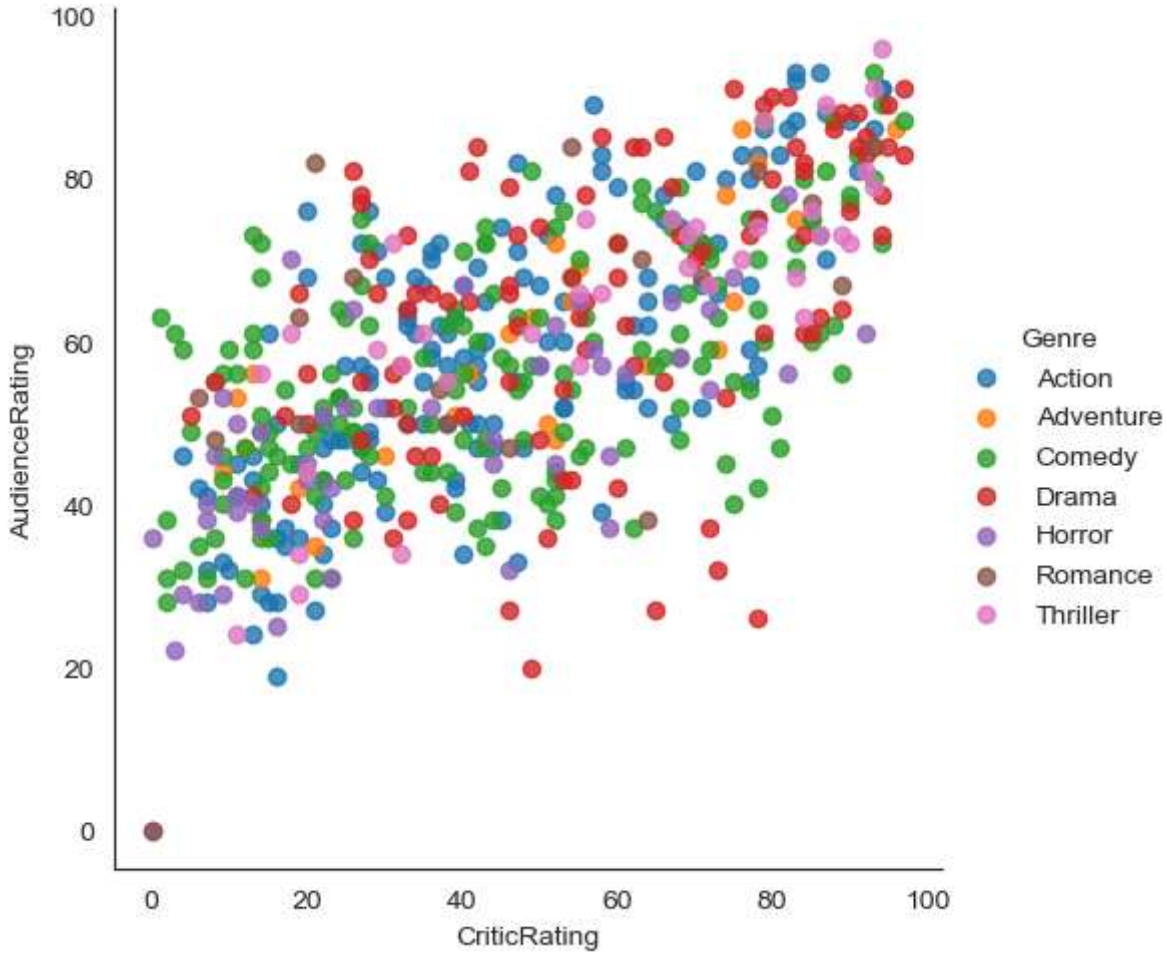
In [107…
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',\
                  fit_reg=False)
plt.show()
```


100
80
60
40
20
0
AudienceRating
0
20
40
60
80
100
CriticRating



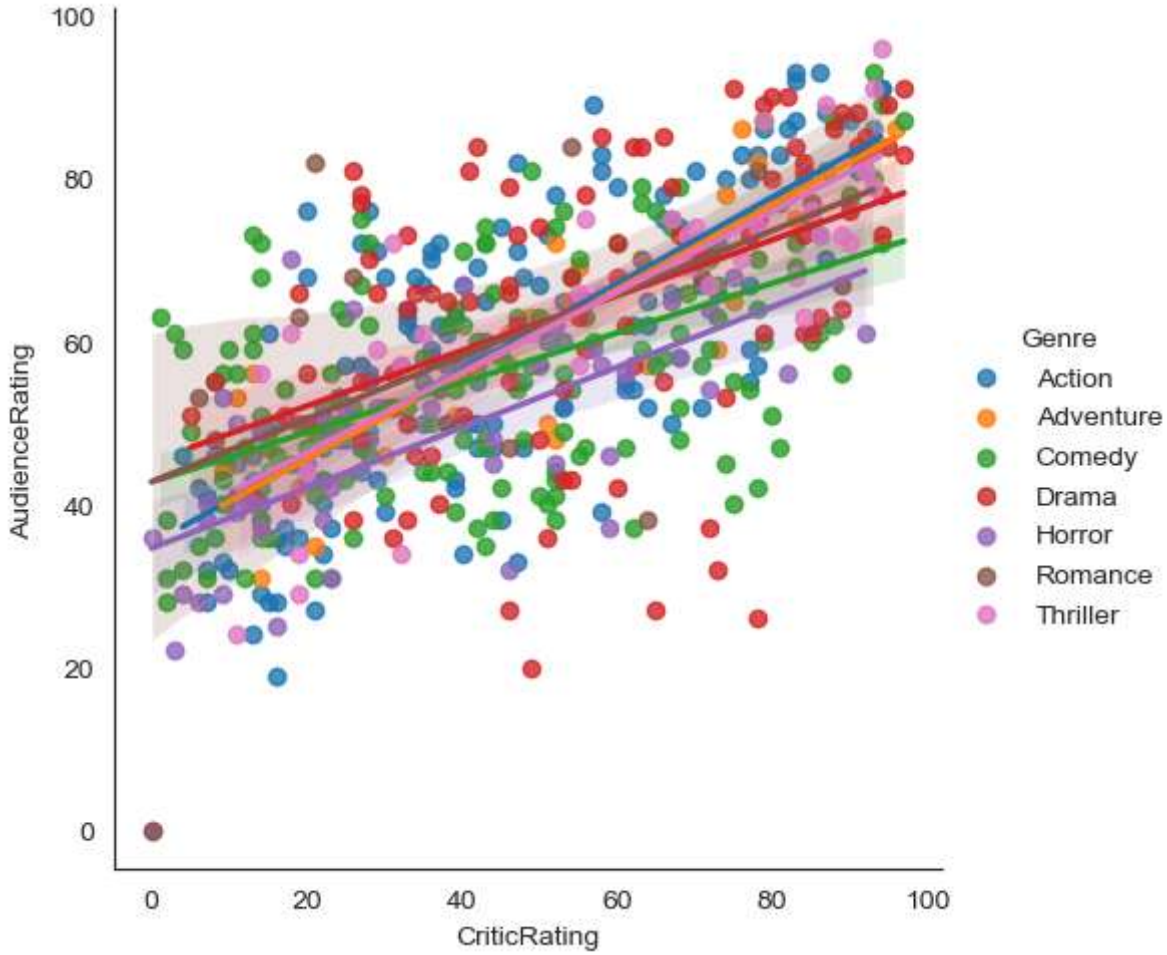
In [109…

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



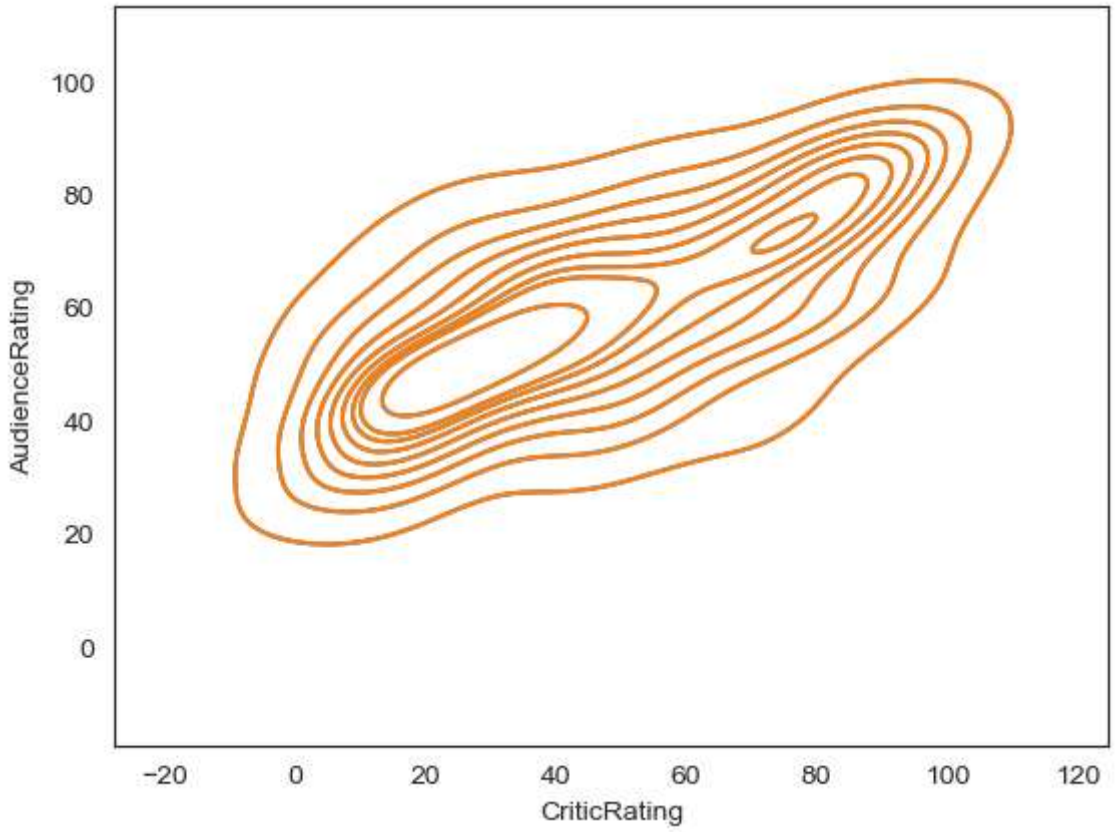
In [115…

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



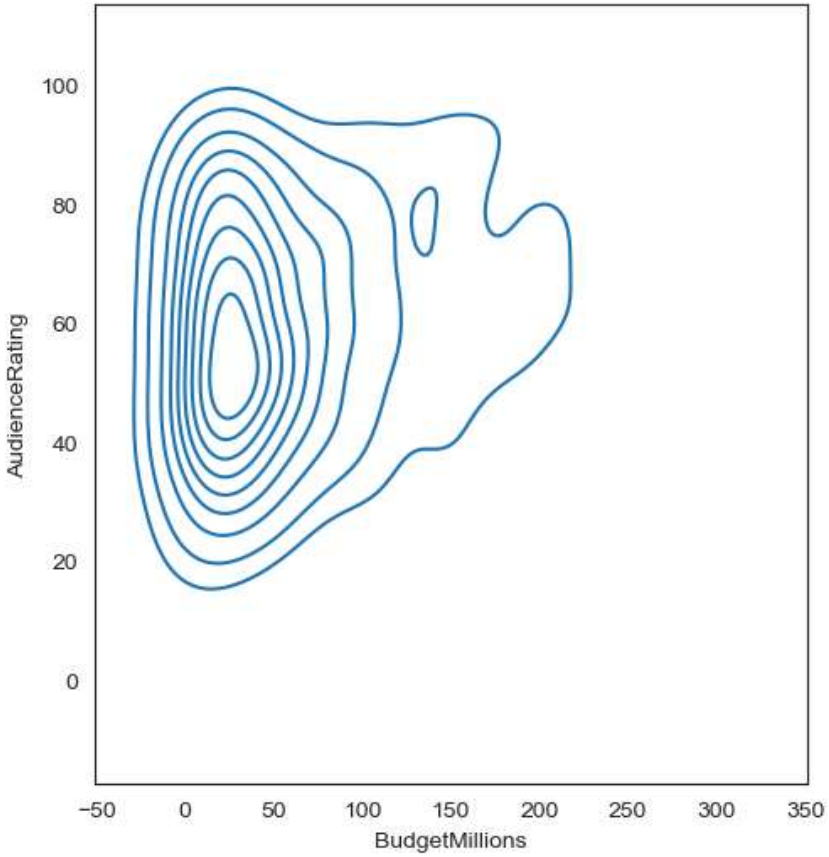
In [129…

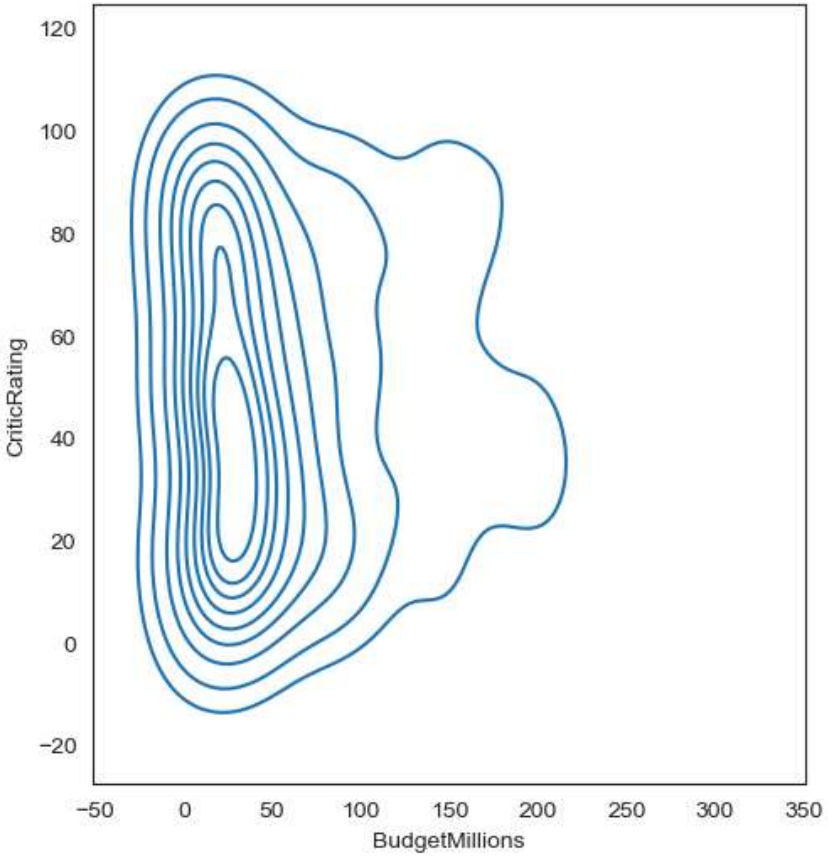
```
k1 = sns.kdeplot(x=movies['CriticRating'],y=movies['AudienceRating'])
plt.show()
```



In [131…

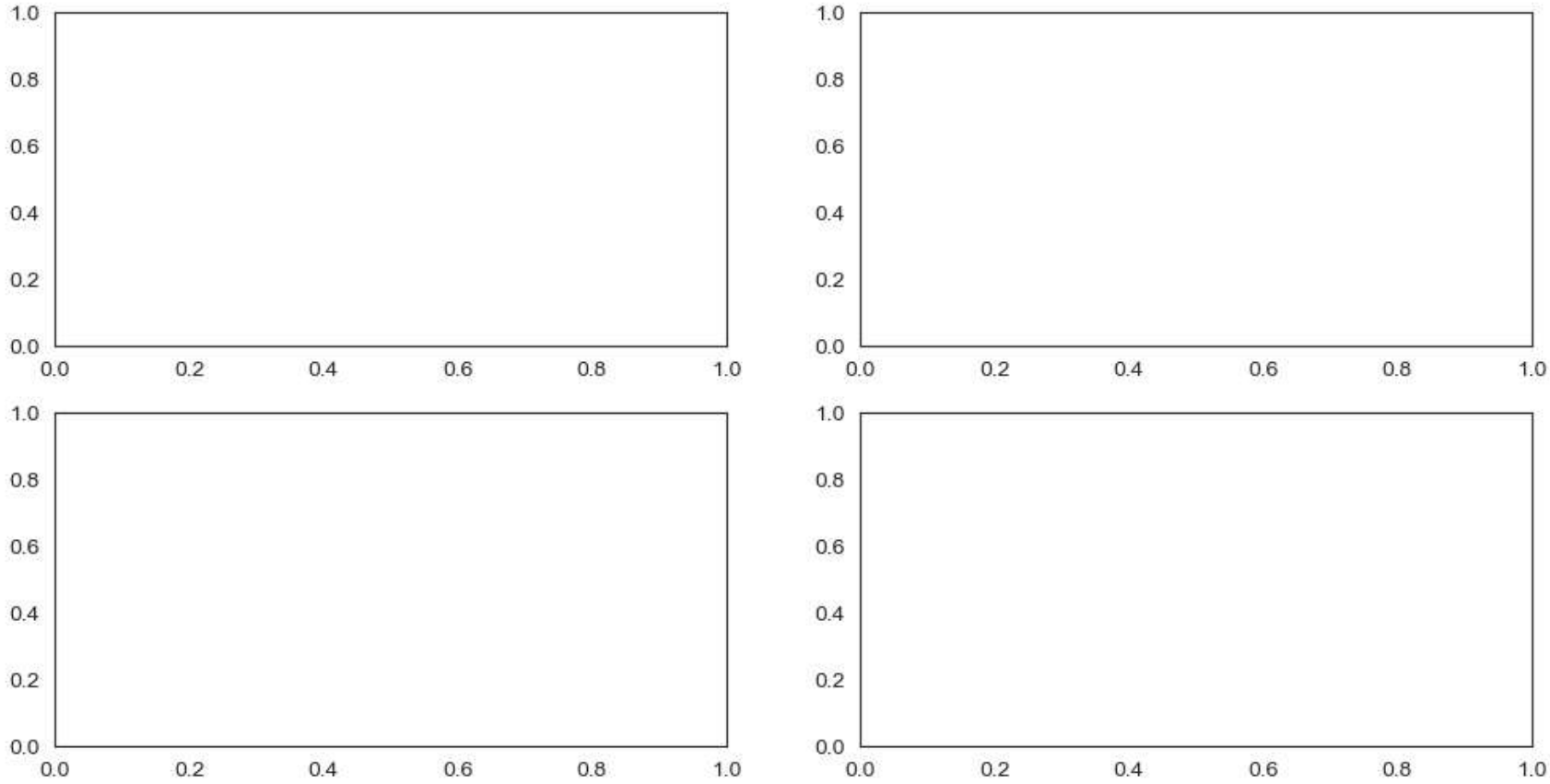
```
#subplots
f, axes = plt.subplots(1,2, figsize =(12,6))

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

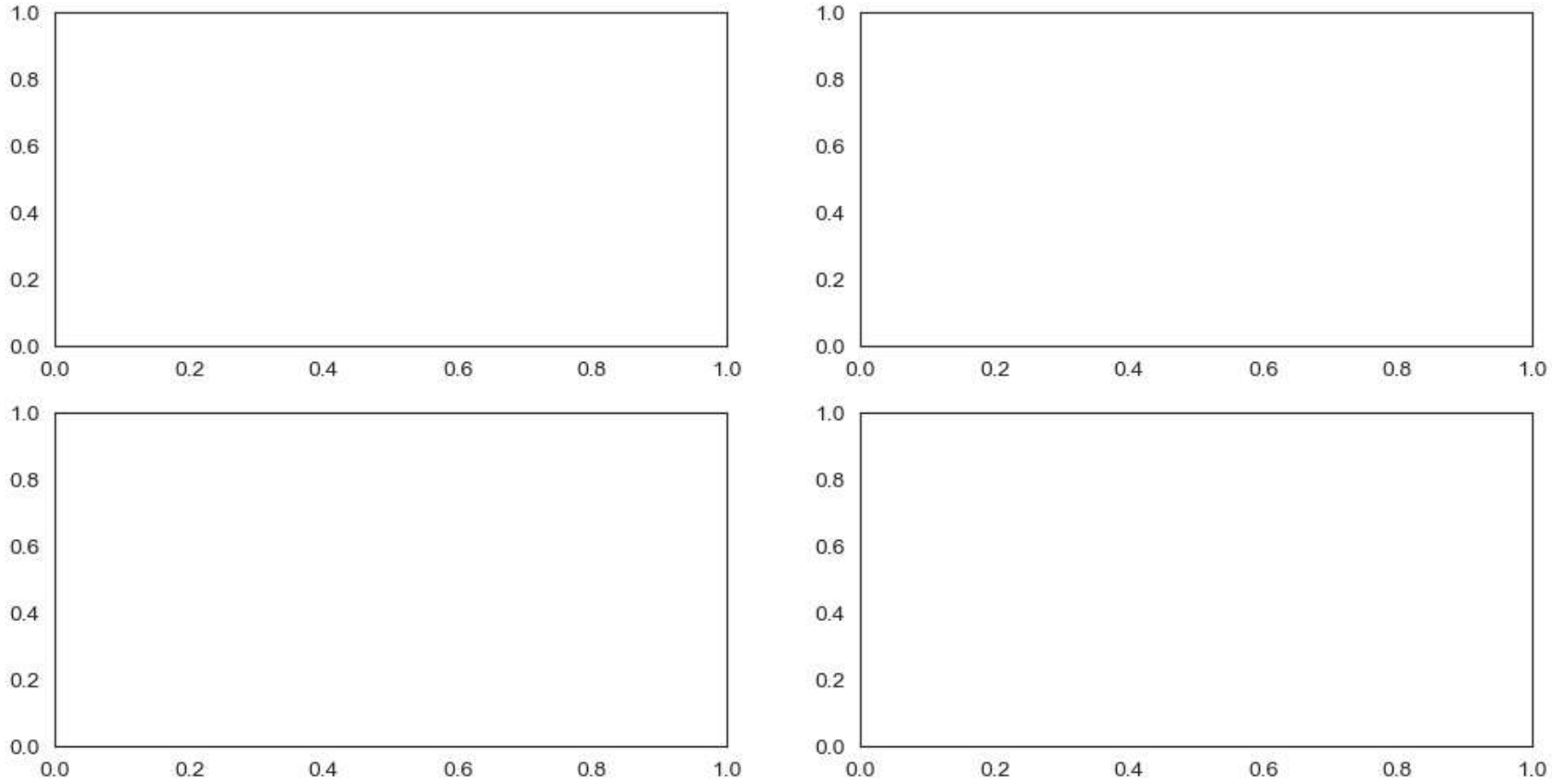




In [135…

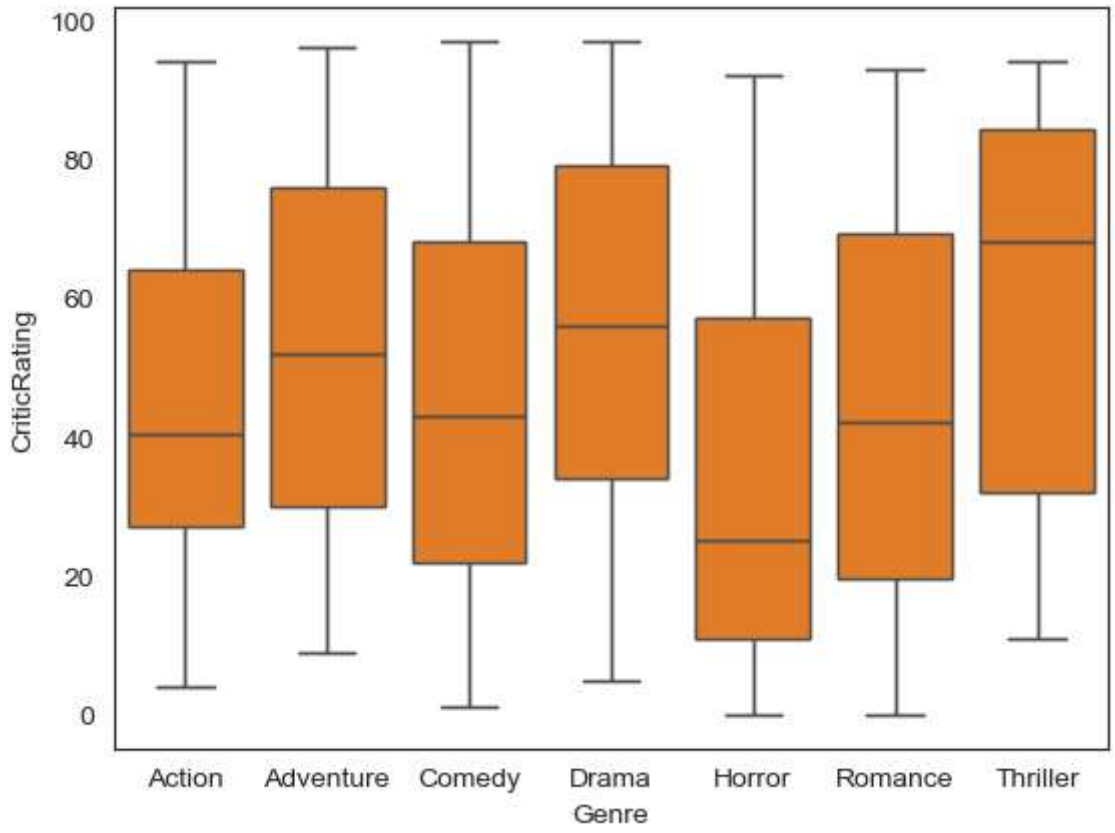
```
#subplots
f, axes = plt.subplots(2,2, figsize =(12,6))
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0



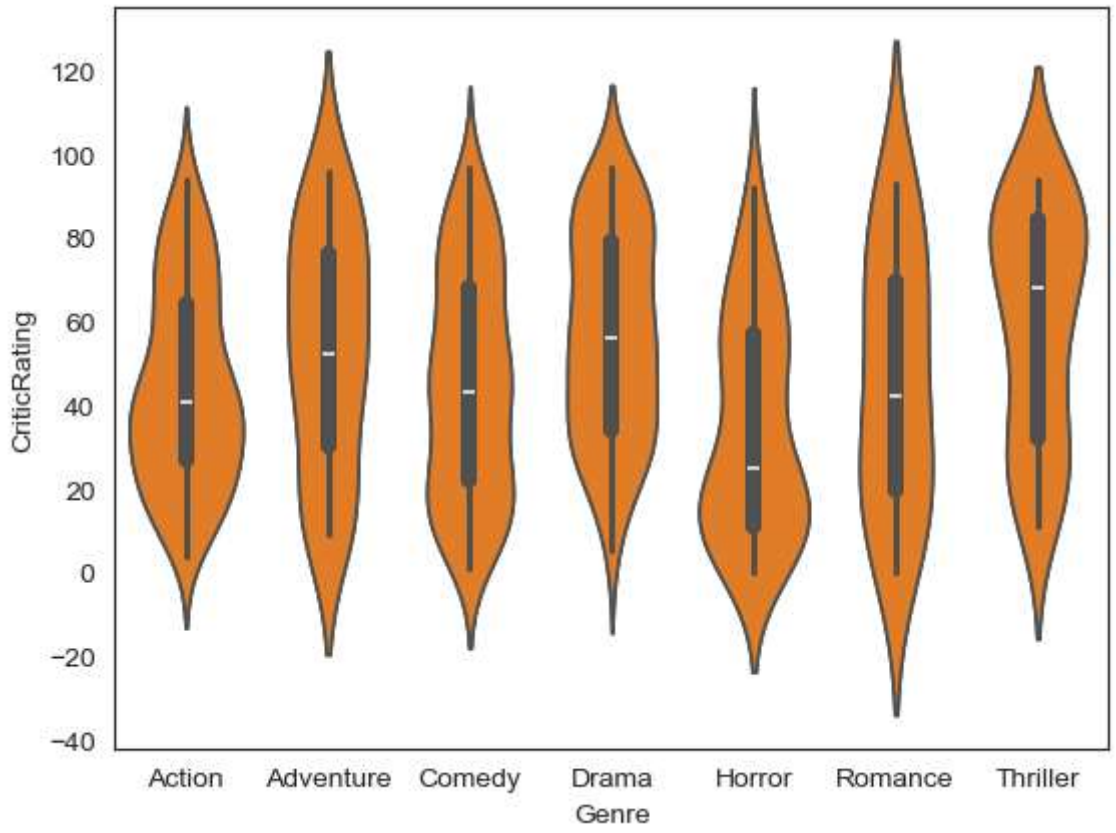
In [139…

```
#Box plots -

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



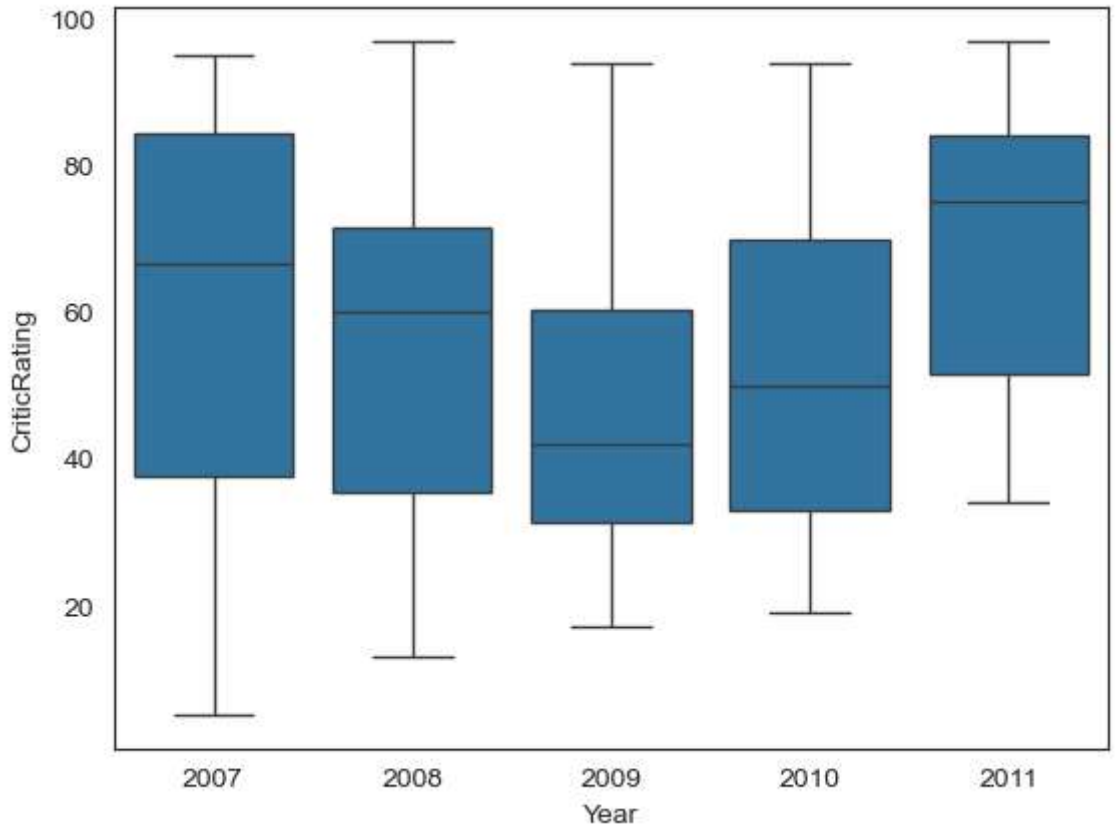
In [143…

```
#violin plot

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



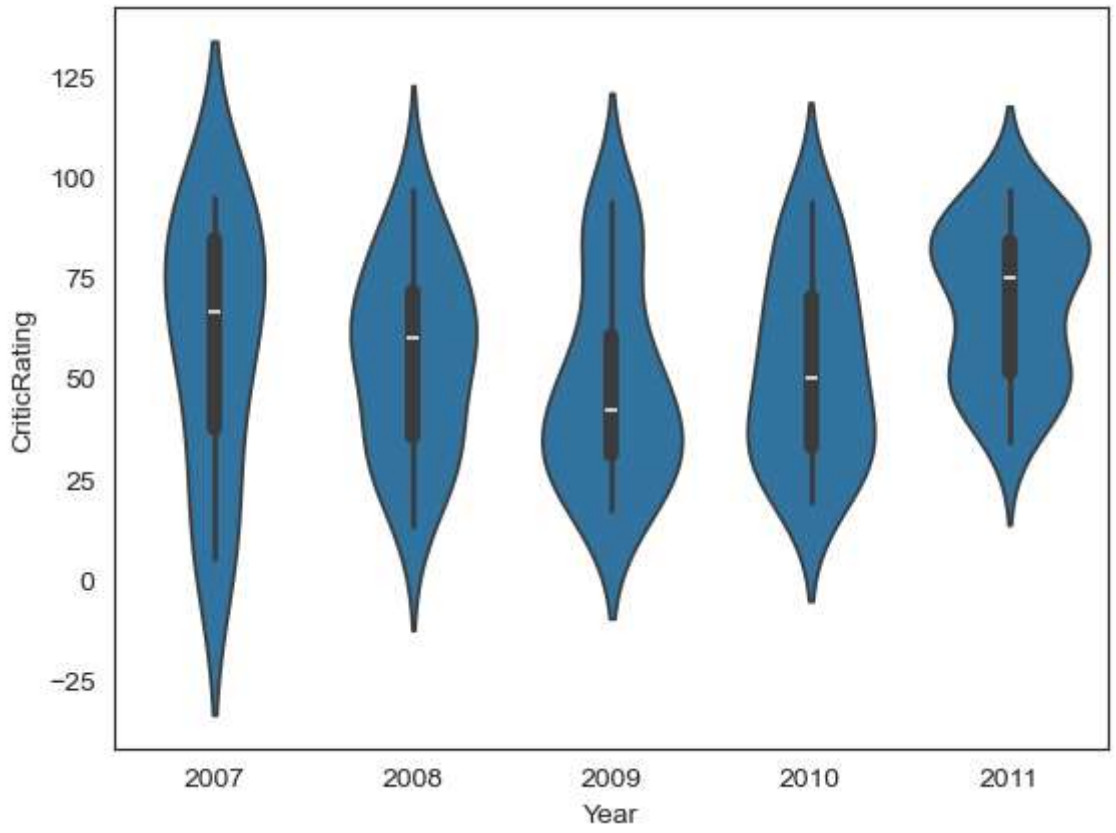
In [145…

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



In [147…

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



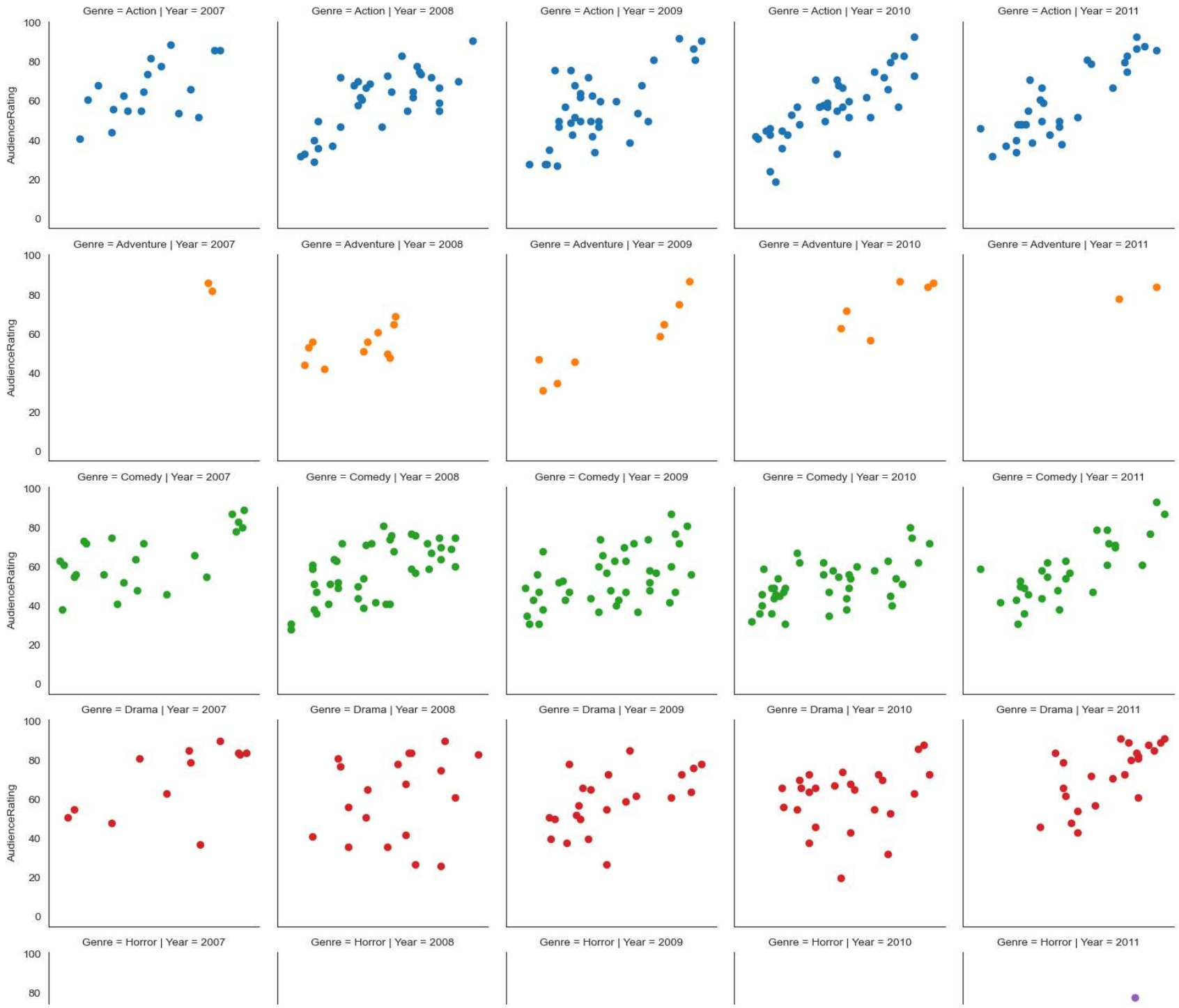
In [149…

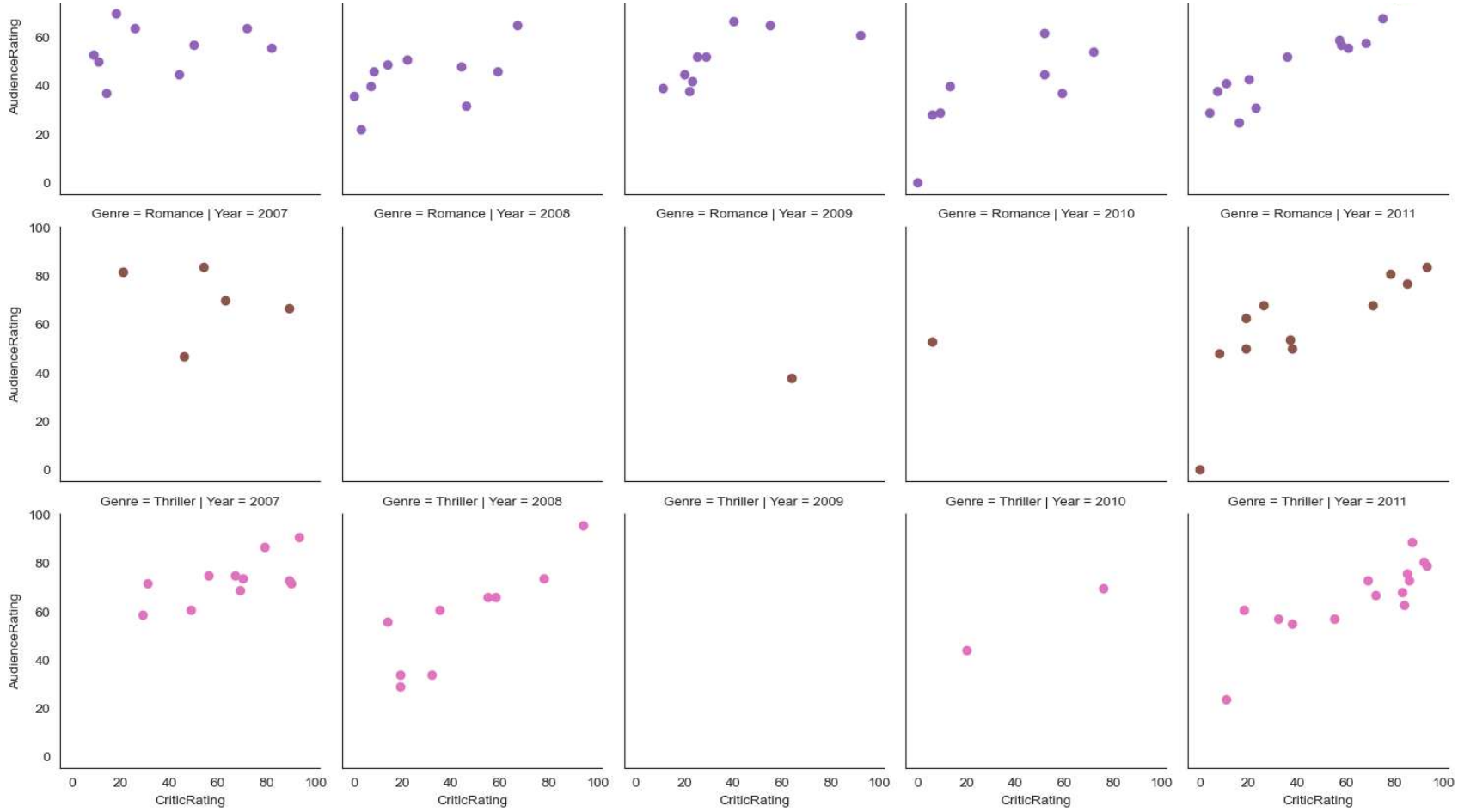
```
g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre') #kind of subplots
plt.show()
```


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
1.0
0.8
0.6
0.4
0.2
0.0

In [151…

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

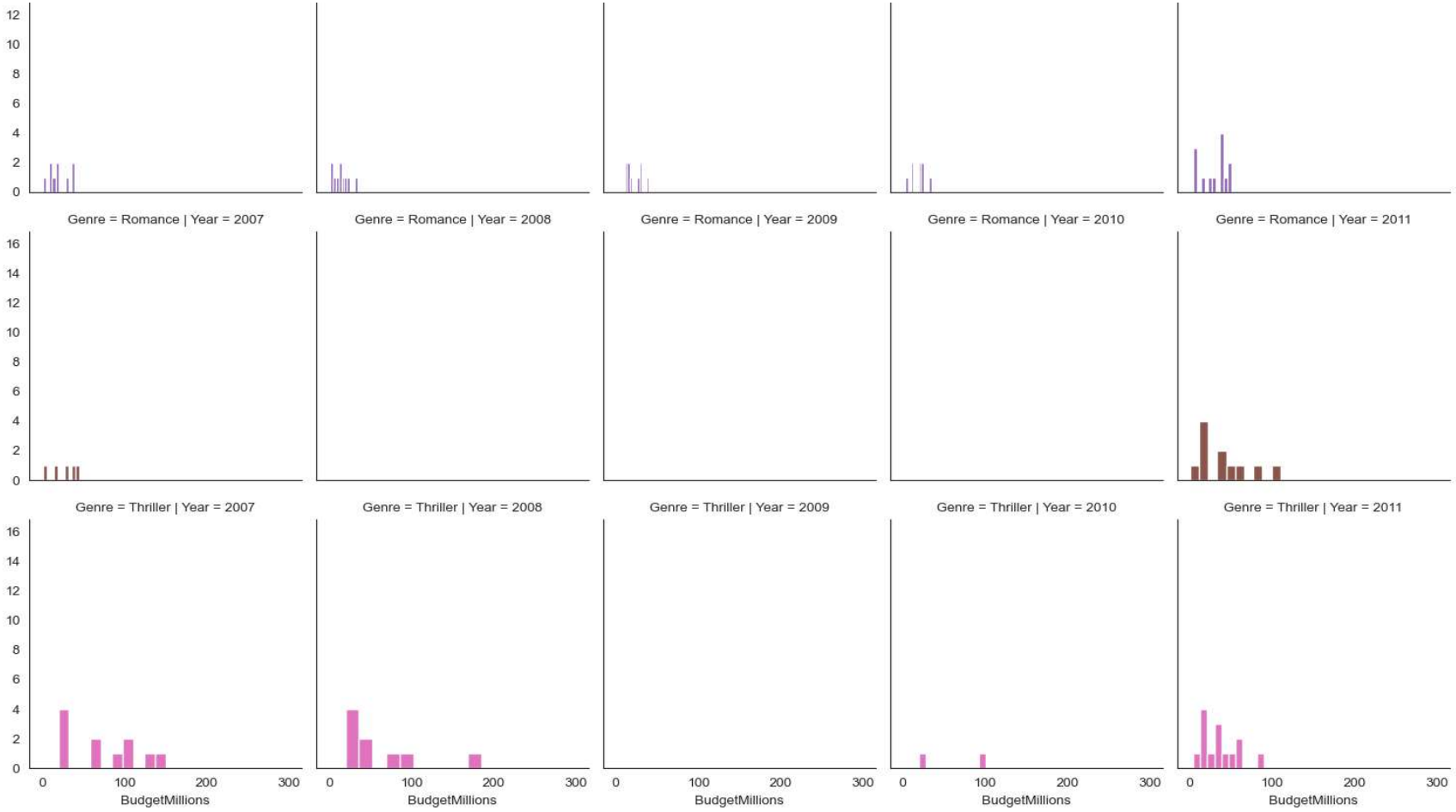

Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
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
AudienceRating
0
20
40
60
80
100



In [153…

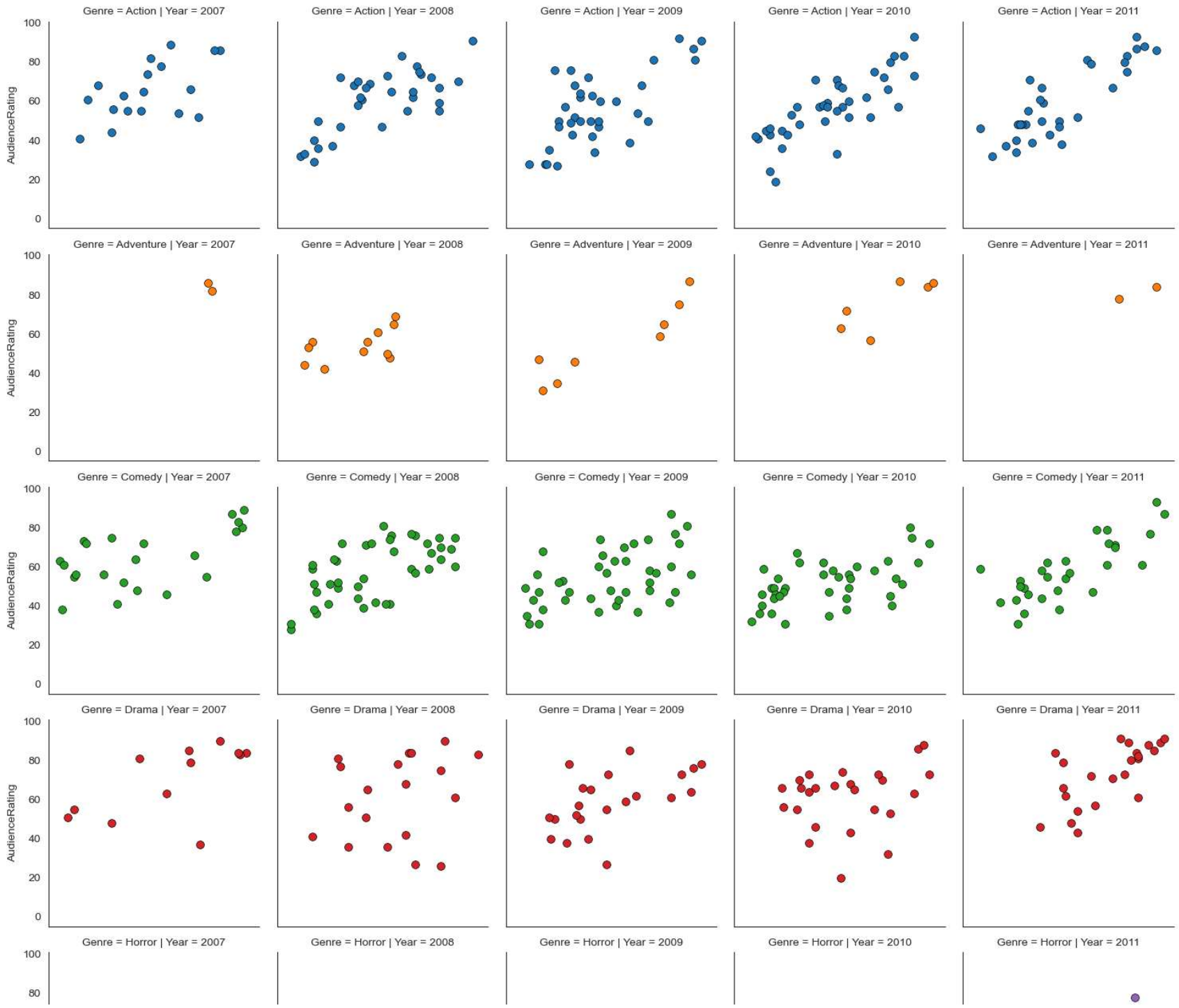
```
g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre')
g = g.map(plt.hist, 'BudgetMillions') #histogramplots are mapped in facetgrid
plt.show()
```

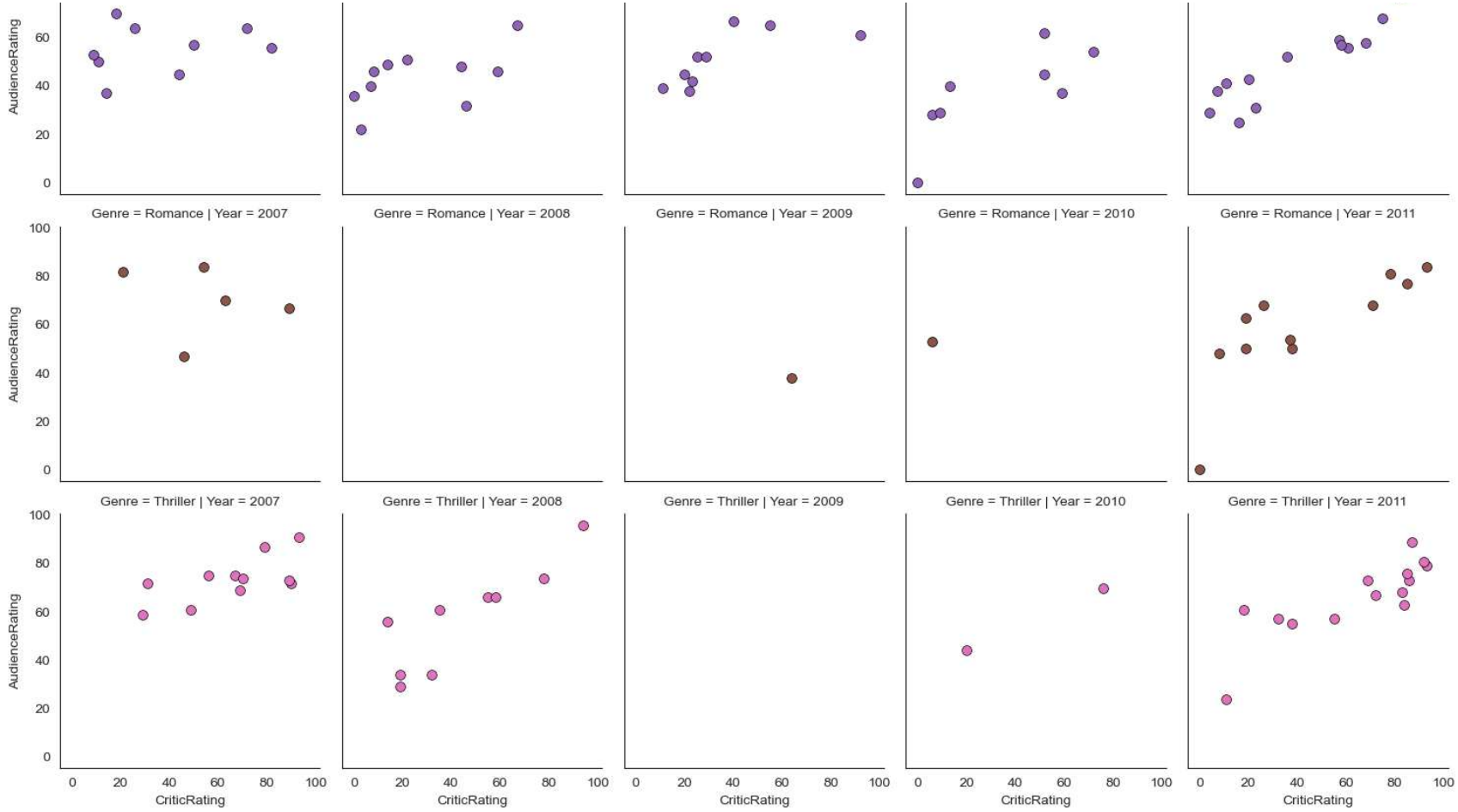

Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
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
0
2
4
6
8
10
12
14
16



In [155…

```
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 ) #scatterplots are mapped in facetgrid
plt.show()
```

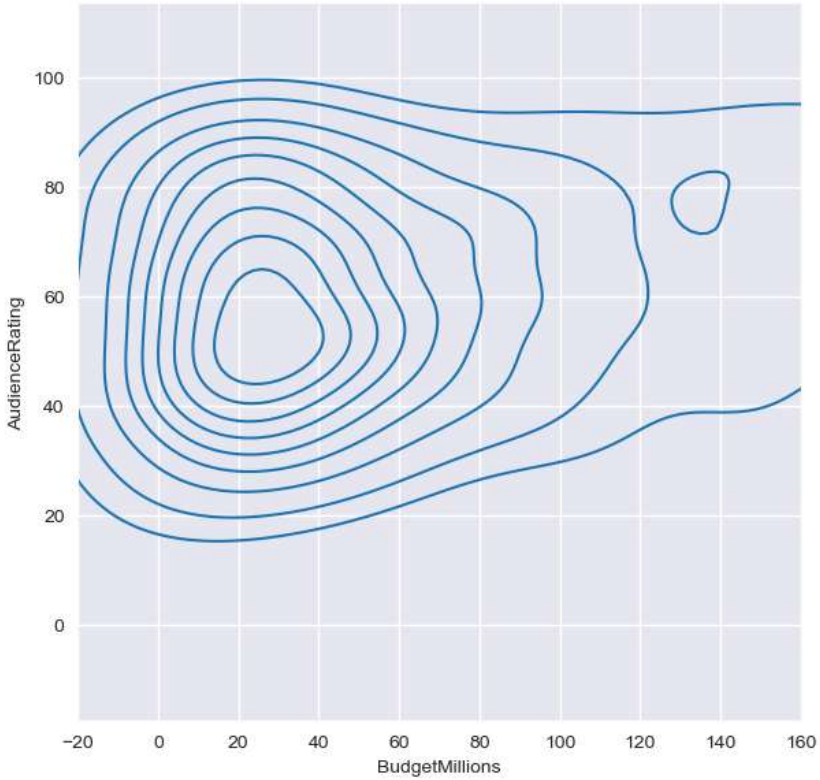

Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
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
AudienceRating
0
20
40
60
80
100

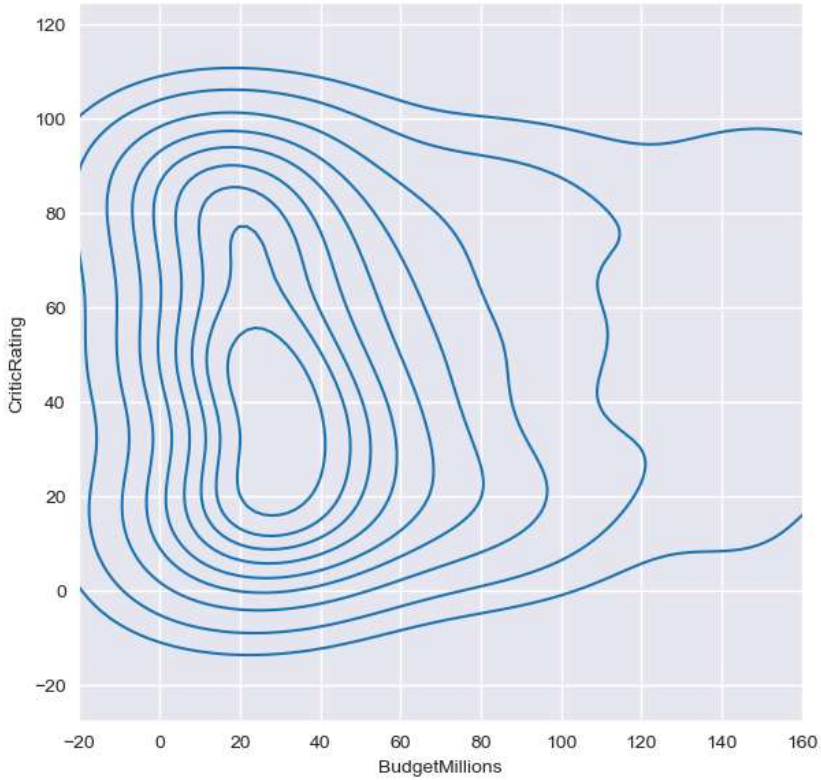


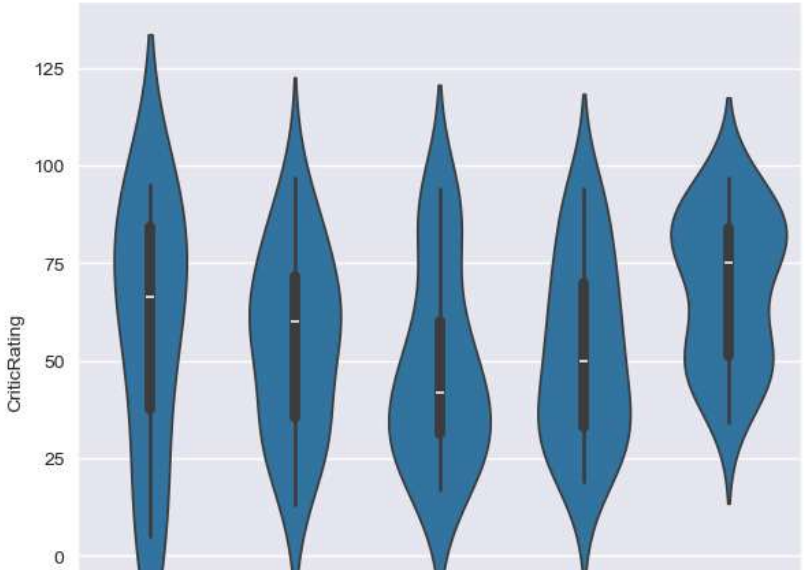
In [169…

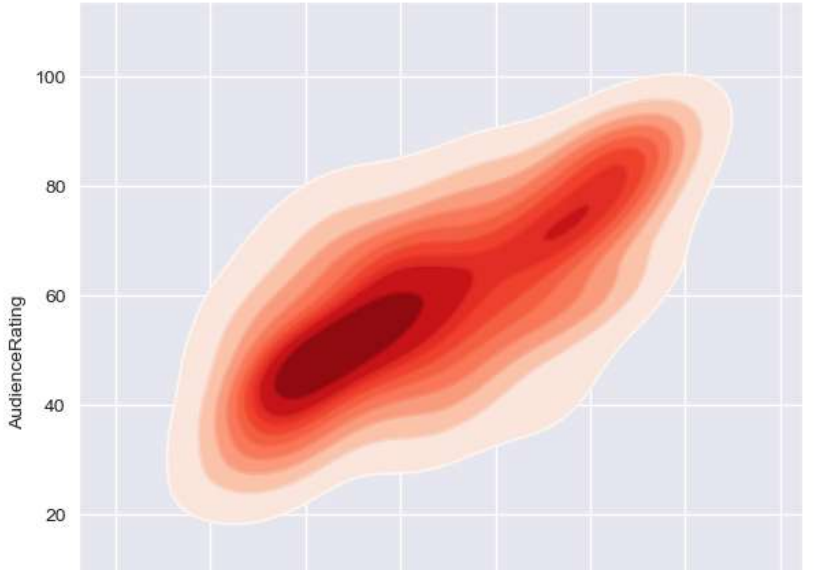
```
sns.set_style('darkgrid')
f, axes = plt.subplots (2,2, figsize = (15,15))

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

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

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

k4 = sns.kdeplot(x=movies['CriticRating'],y=movies['AudienceRating'],shade = True,shade_lowest=False,cmap='Reds',ax=a
```

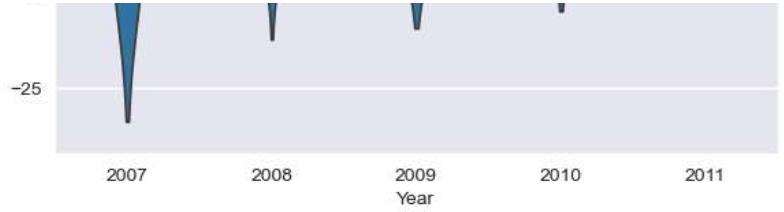
```
k4b = sns.kdeplot(x=movies['CriticRating'],y=movies['AudienceRating'],cmap='Reds',ax = axes[1,1])

plt.show()
```

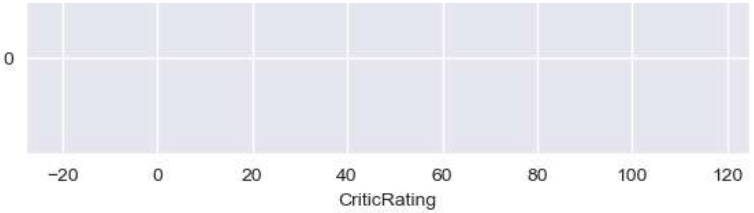

AudienceRating
BudgetMillions
100
80
60
40
20
0
−20
0
20
40
60
80
100
120
140
160


CriticRating
BudgetMillions
120
100
80
60
40
20
0
−20
−20
0
20
40
60
80
100
120
140
160


CriticRating
125
100
75
50
25
0

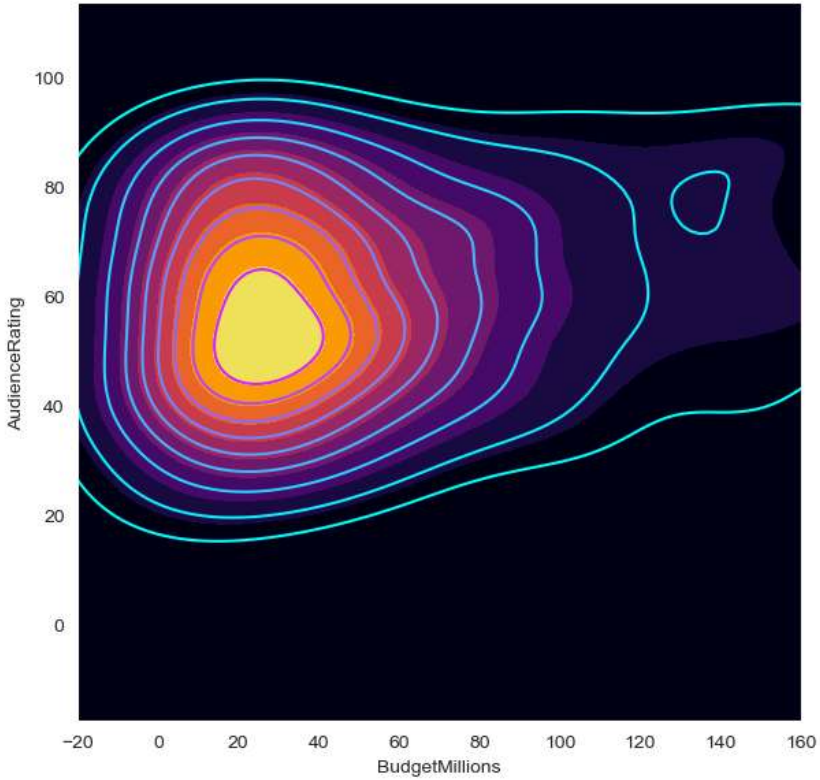

AudienceRating
100
80
60
40
20

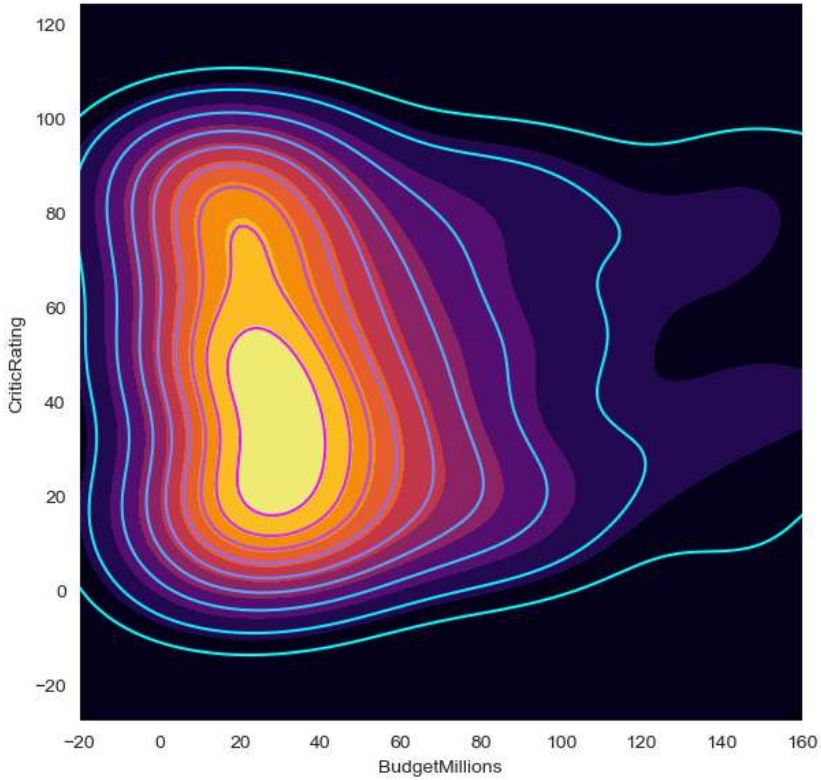


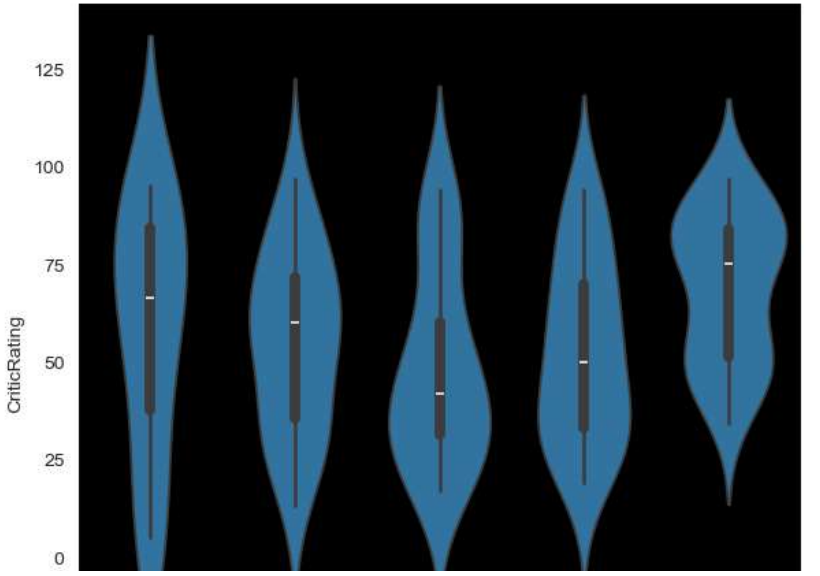


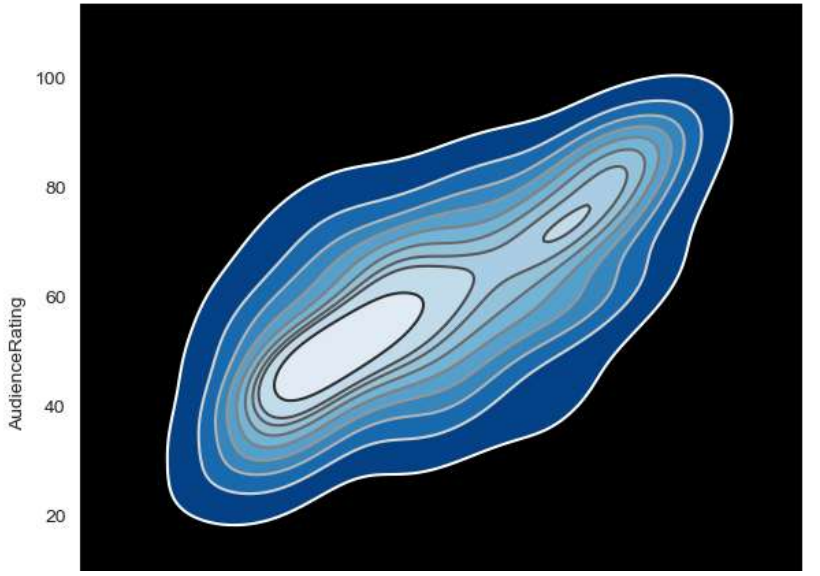
In [181…

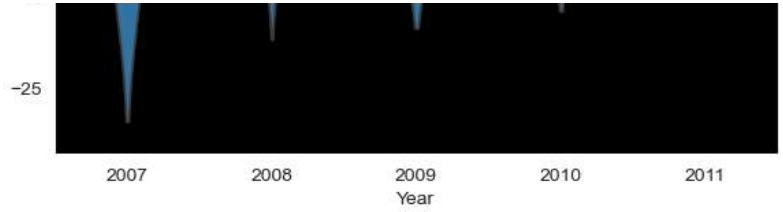
```
sns.set_style('dark',{'axes.facecolor':'black'})
f, axes = plt.subplots (2,2, figsize = (15,15))

#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
BudgetMillions
100
80
60
40
20
0
−20
0
20
40
60
80
100
120
140
160


CriticRating
BudgetMillions
120
100
80
60
40
20
0
−20
−20
0
20
40
60
80
100
120
140
160


CriticRating
125
100
75
50
25
0


AudienceRating
100
80
60
40
20



In [ ]:
```
Final discussion what we learn so far -
-1> category datatype in python
-2> jointplots
-3> histogram
-4> stacked histograms
-5> Kde plot
-6> subplot
-7> violin plots
-8> Factet grid
-9> Building dashboards
```

eda completed