MOVIE RATING ANALYTIC

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

In [2]:
```
os.getcwd() #if you want to change the working directory
```

Out[2]:
```
'c:\\Users\\AR ANSARI\\vscode\\Python\\EDA'
```

In [3]:
```
movies = pandas.read_csv(r"C:\Users\AR ANSARI\Desktop\Movie-Rating.csv")
```

In [4]:
```
movies
```

Out[4]:

	Film	Genre	Rotten Tomatoes Ratings %	Audience Ratings %	Budget (million $)
0	(500) Days of Summer	Comedy	87	81	8
1	10,000 B.C.	Adventure	9	44	105
2	12 Rounds	Action	30	52	20
3	127 Hours	Adventure	93	84	18
4	17 Again	Comedy	55	70	20
...	...	...	...	...	...
554	Your Highness	Comedy	26	36	50
555	Youth in Revolt	Comedy	68	52	18
556	Zodiac	Thriller	89	73	65
557	Zombieland	Action	90	87	24
558	Zookeeper	Comedy	14	42	80

559 rows × 6 columns

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

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

In [6]:
```
movies.head()   #first 5 rows
```

Out[6]:

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

In [7]:
```
movies.tail() #last 5 rows
```

Out[7]:

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

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

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

In [9]:
```
movies.columns = ['Film', 'Genre', 'CriticRating', 'AudienceRating','BudgetMilli
```

In [10]:
```
movies.head() #removed spaces and removed noise characters
```

Out[10]:

	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 [11]:
```
movies.info() #getting info about the dataset
```

```
<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 [12]:
```
movies.describe() #getting statistical summary of the dataset
```

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['Film'] #accessing a particular column
#movies['AudienceRating %]
```

Out[13]:
```
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 [14]:
```
movies.Film #another way of accessing a particular column
```

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') #changing datatype of a column
```

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 ', '1
Hours', ..., 'Youth in Revolt', 'Zodiac', 'Zombieland ', 'Zookeeper']
```

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

Out[17]:

	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 [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')#changing datatype of a column
movies.Year = movies.Year.astype('category') #changing datatype of a column
```

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', 'Roman
'Thriller']
```

In [21]:
```
movies.Year #is it real no you can take  average ,min,max but out come have n me
```

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.Genre.cat.categories #to see the categories
```

Out[22]:
```
Index(['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance',
       'Thriller'],
      dtype='object')
```

In [23]:
```
movies.describe() #statistical summary of numerical columns
```

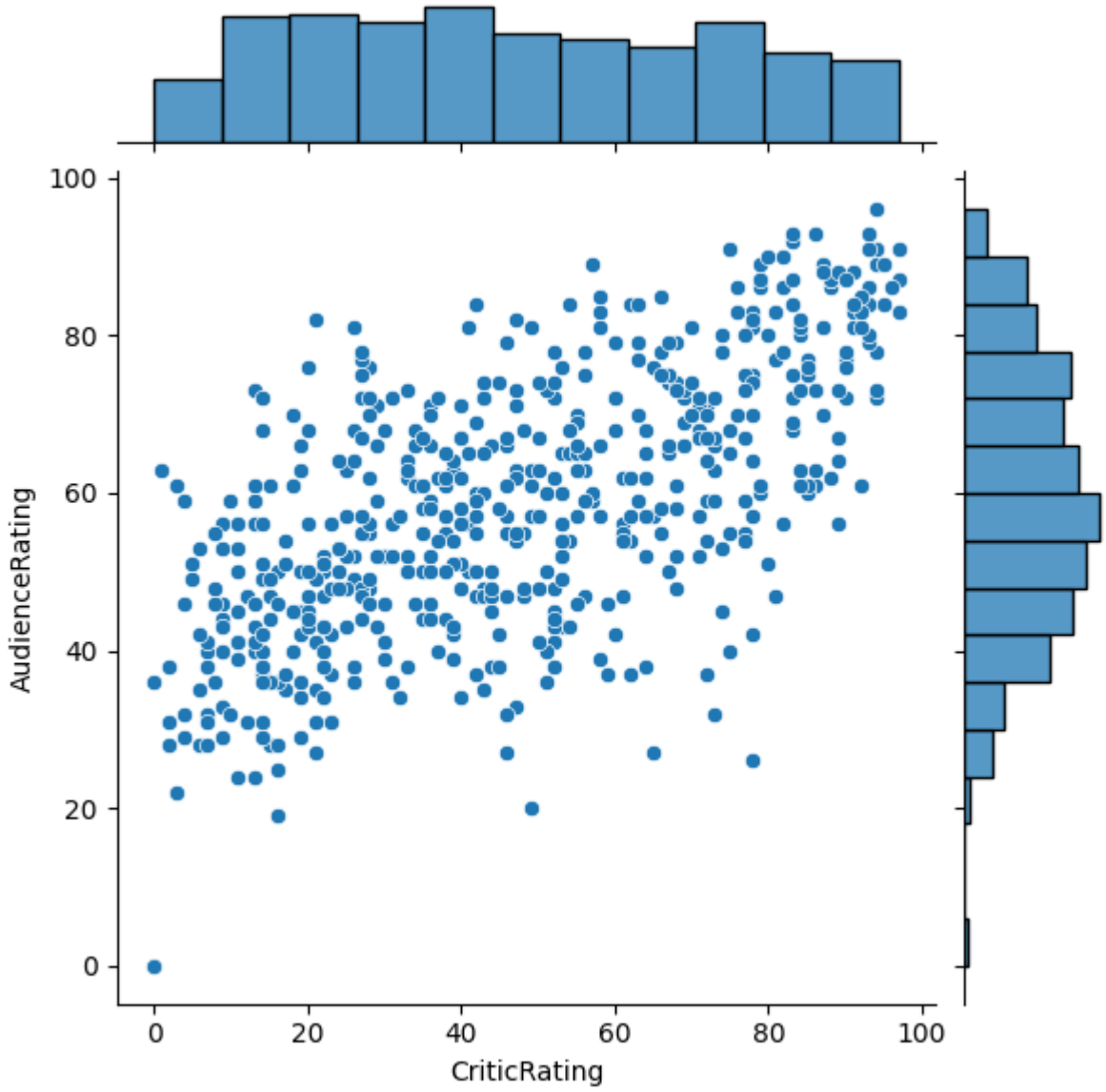
Out[23]:

	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

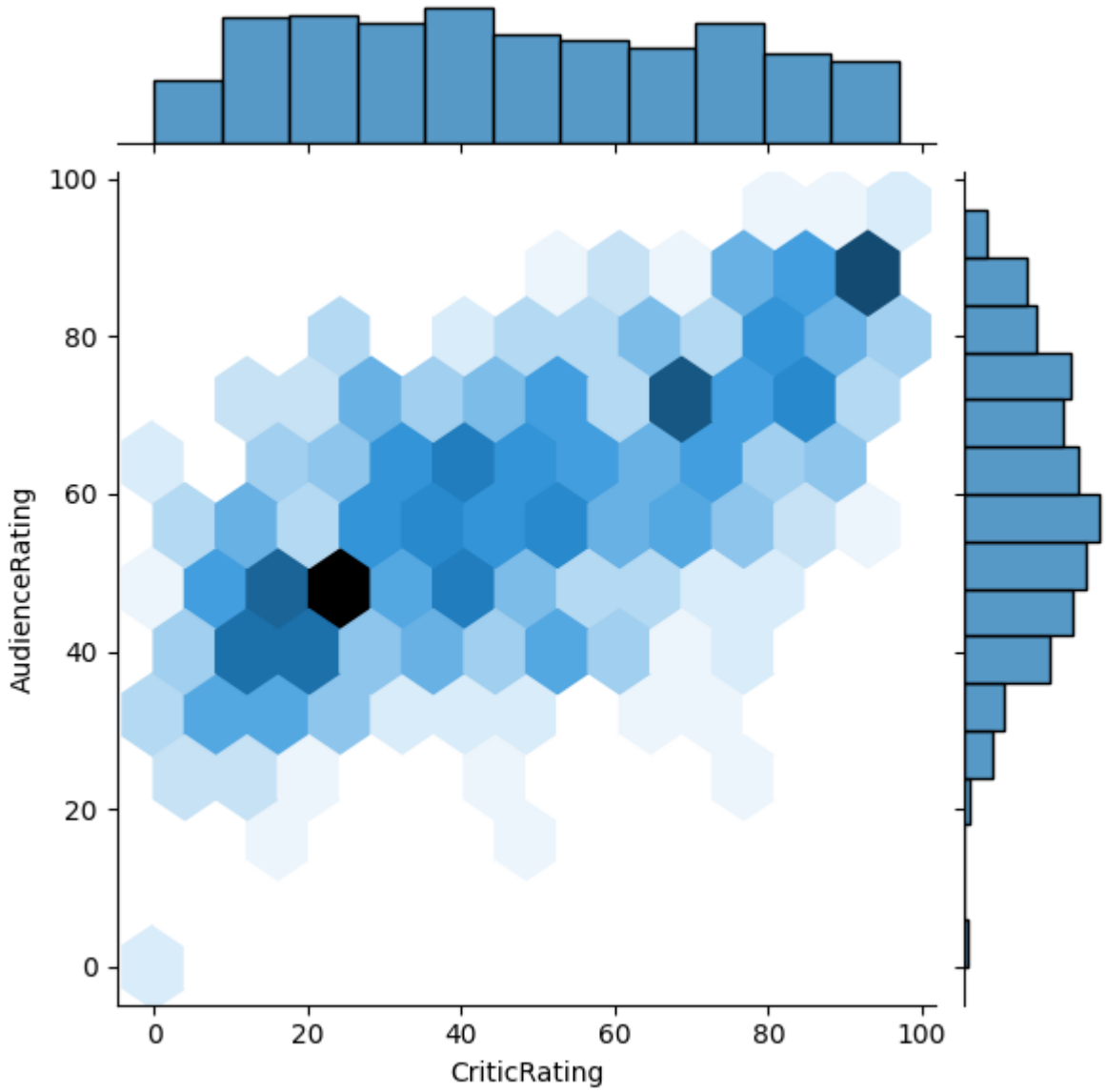
In [24]:
```
#How to working with joint plots

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

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

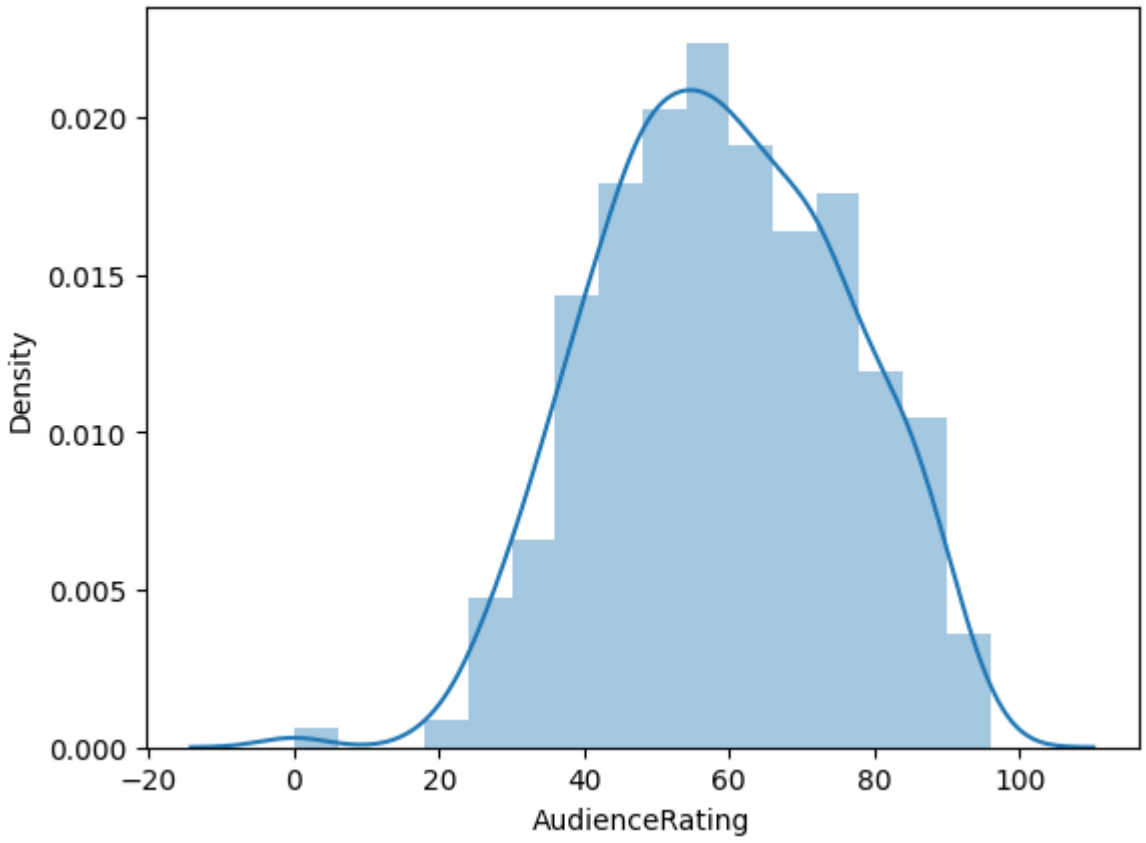
In [25]:
```
j = sns.jointplot(data=movies, x='CriticRating', y='AudienceRating')
```



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

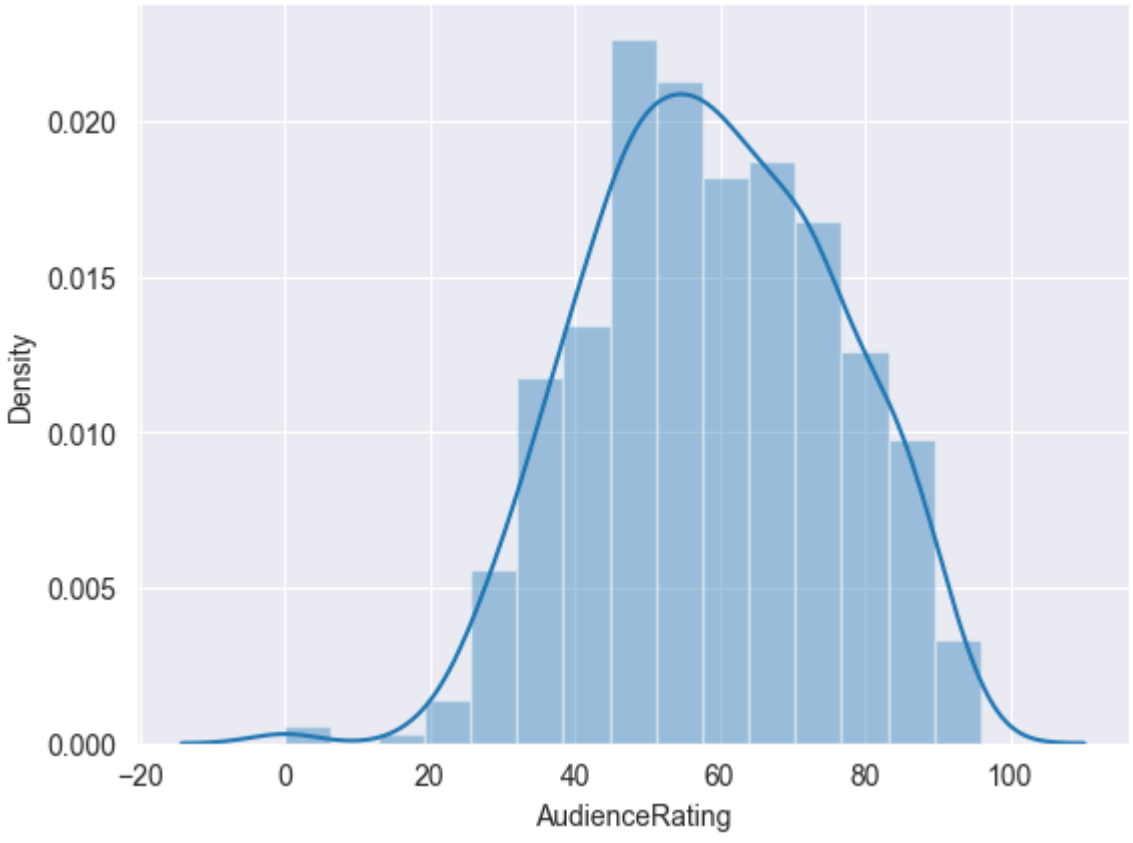


In [27]:
```
#Histogram
# <<< chat1
m1 = sns.distplot(movies.AudienceRating )
```

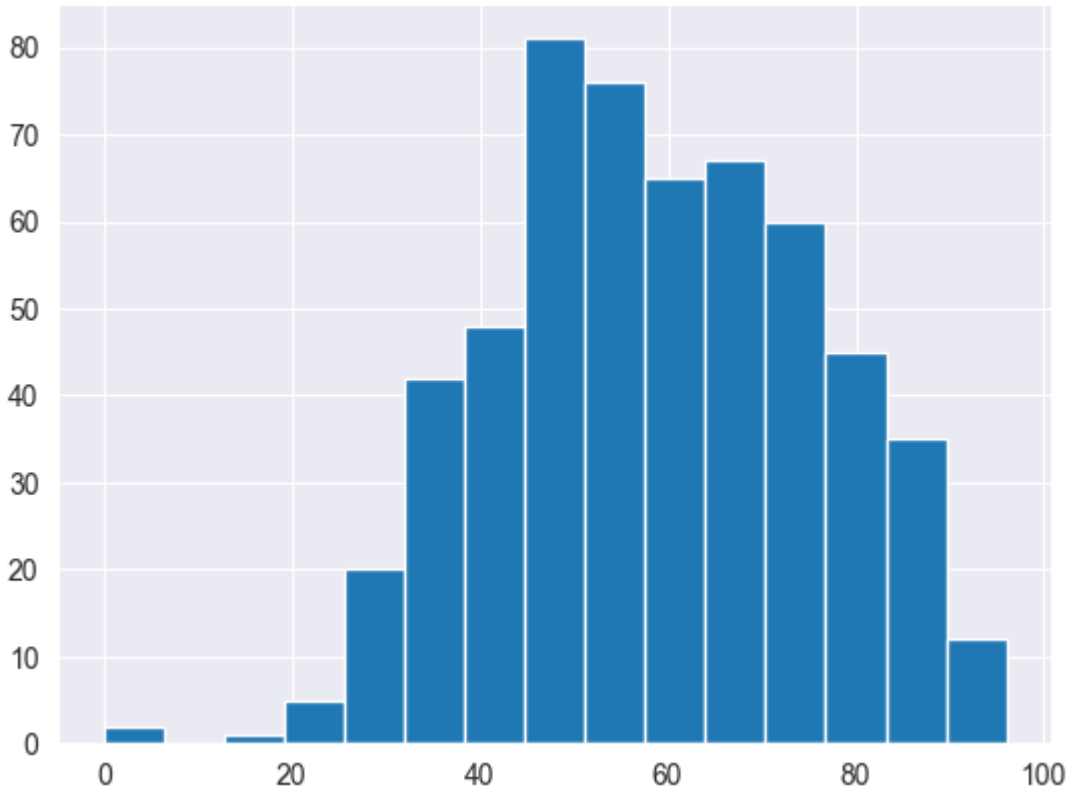


In [28]:
```
sns.set_style('darkgrid')
```

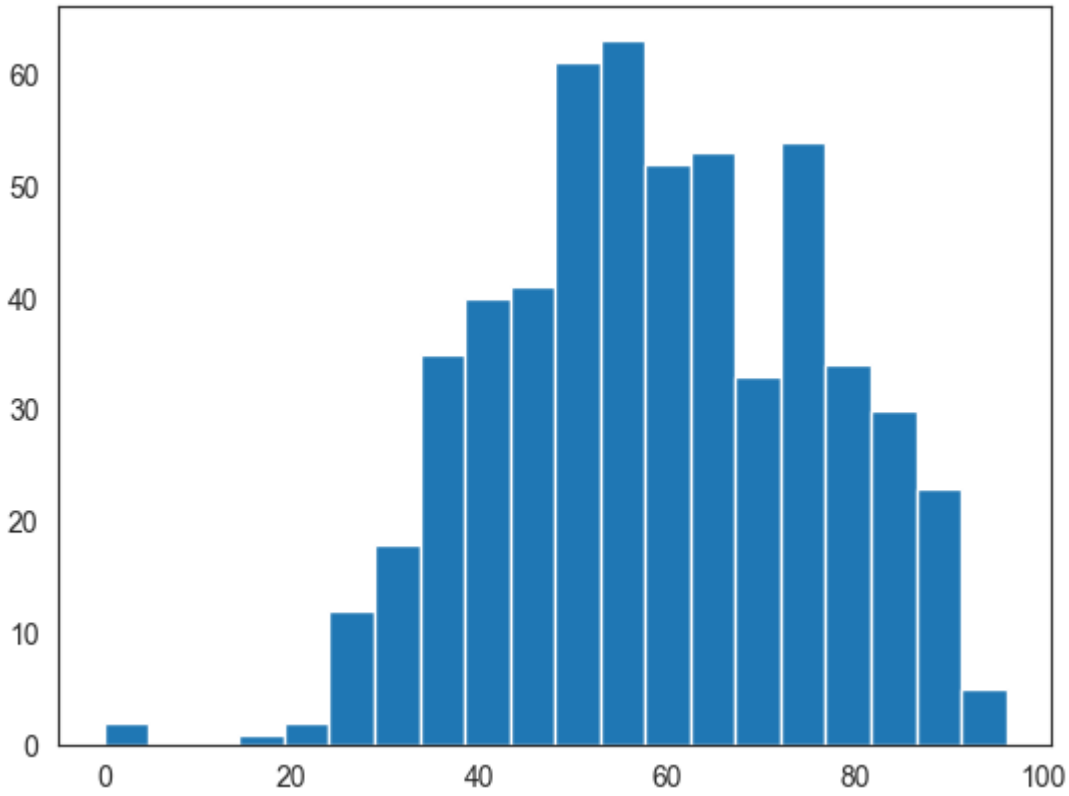
In [29]:
```
m2 = sns.distplot(movies.AudienceRating , bins=15) #bins is used to increase the
```



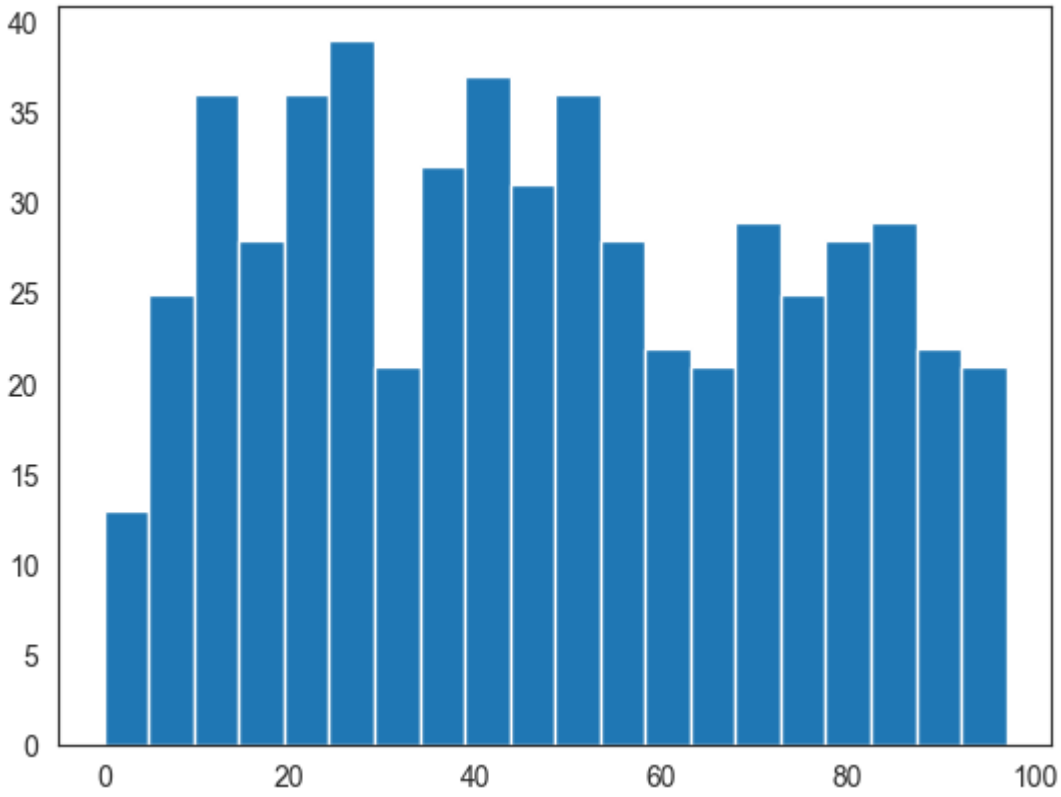
In [30]:
```
#sns.set_style('darkgrid')
n1 = plt.hist(movies.AudienceRating , bins=15) #bins is used to increase the num
```



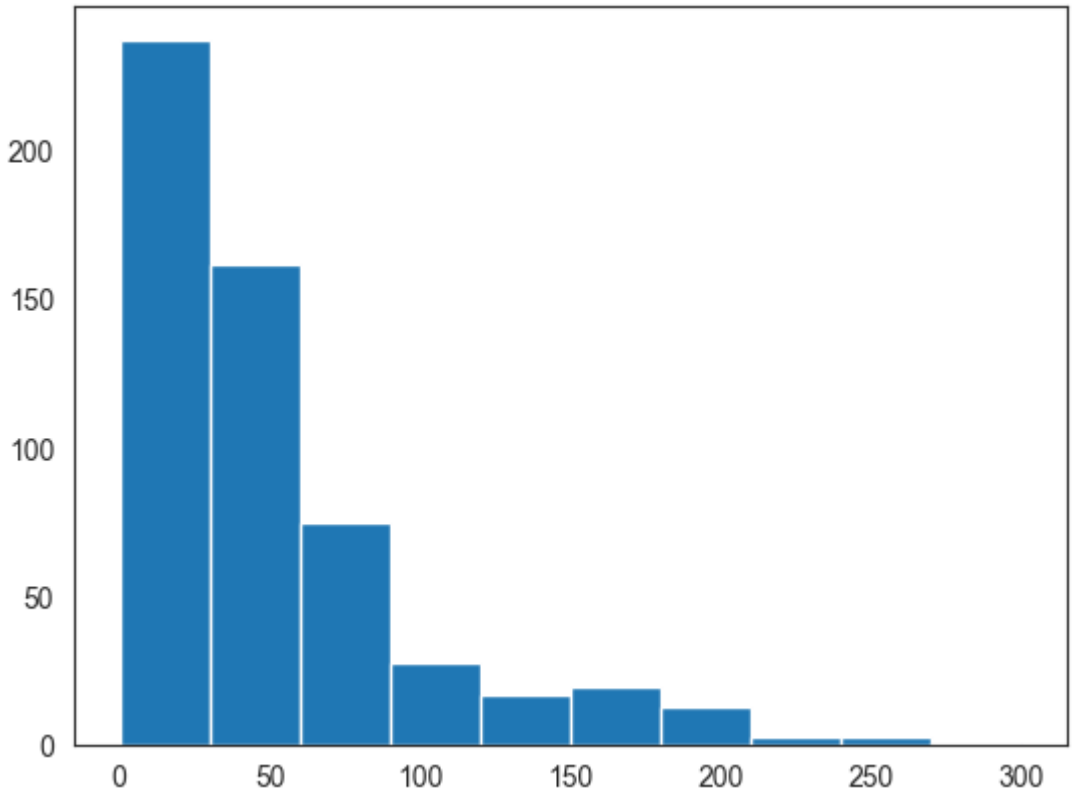
In [31]:
```
sns.set_style('white')#normal distribution & called as bell curve
n1 = plt.hist(movies.AudienceRating , bins=20) #bins is used to increase the num
```



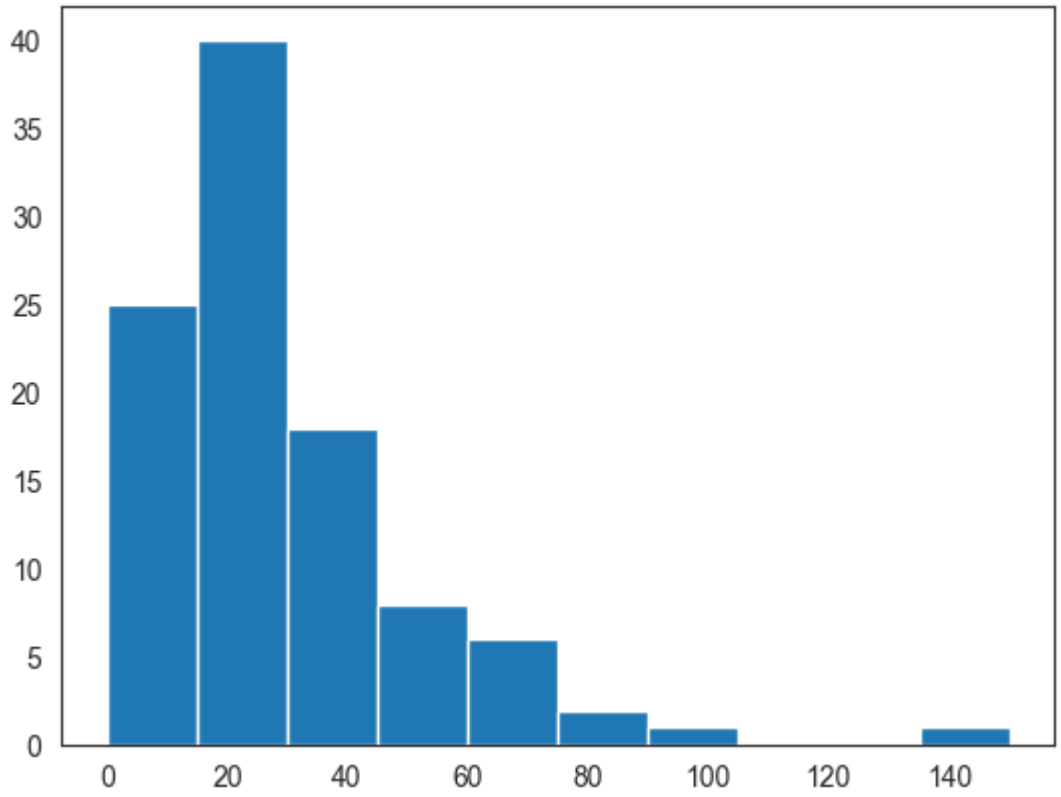
In [32]:
```
n1 = plt.hist(movies.CriticRating , bins=20) #bins is used to increase the numbe
```



In [33]:
```
plt.hist(movies.BudgetMillions)
plt.show()
```



In [34]:
```
plt.hist(movies[movies.Genre == 'Drama'].BudgetMillions)
plt.show()
```

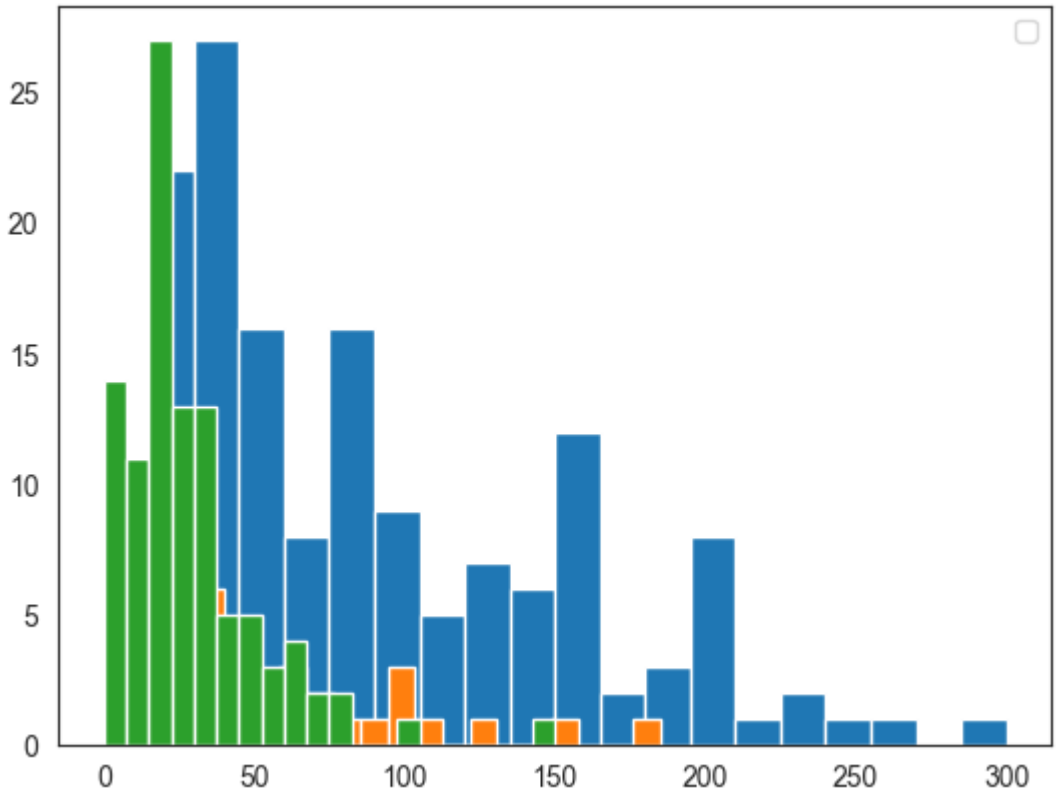


In [35]:
```
movies.head()
```

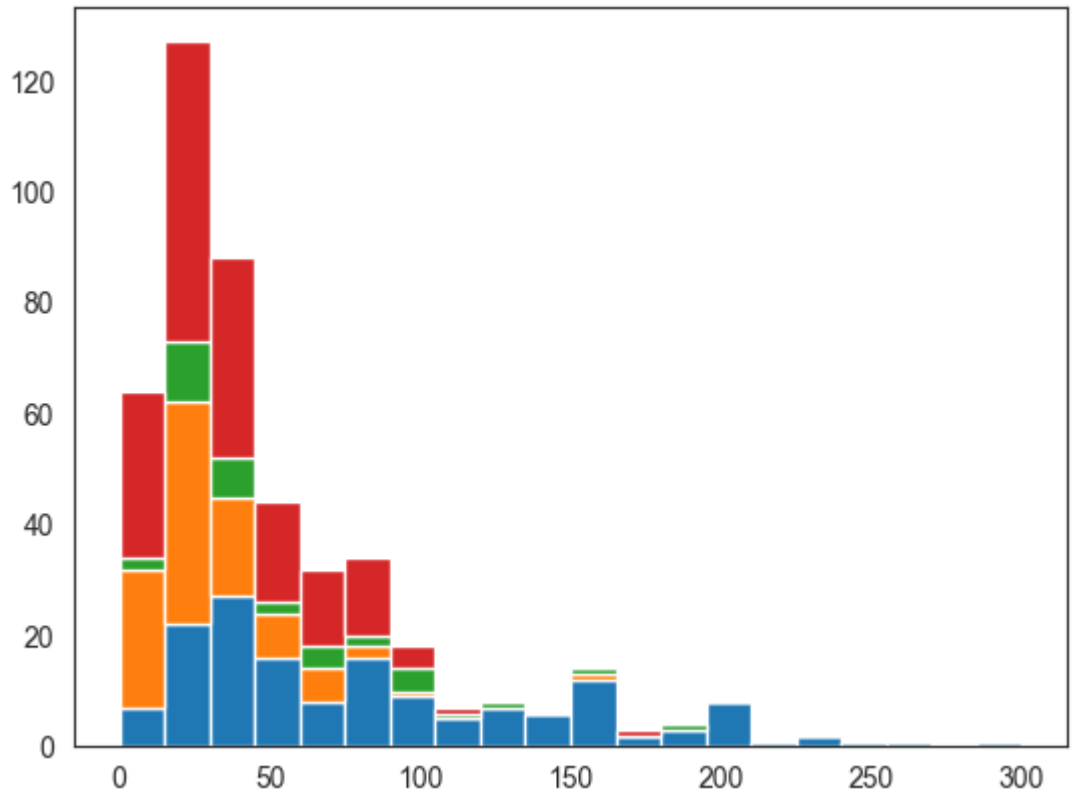
Out[35]:

	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 [36]:
```
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()
```



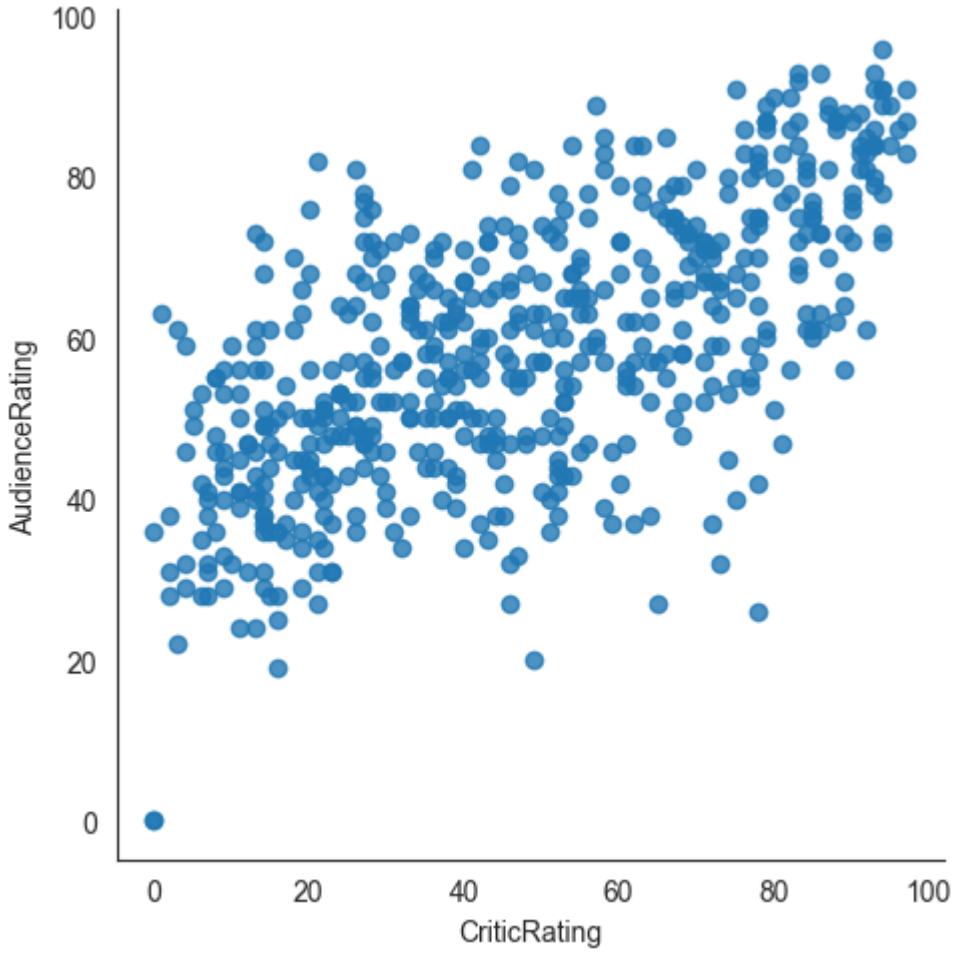
In [37]:
```
plt.hist([movies[movies.Genre == 'Action'].BudgetMillions,\
          movies[movies.Genre == 'Drama'].BudgetMillions,\
          movies[movies.Genre == 'Thriller'].BudgetMillions,\
          movies[movies.Genre == 'Comedy'].BudgetMillions],
          bins = 20, stacked= True)
plt.show()
```



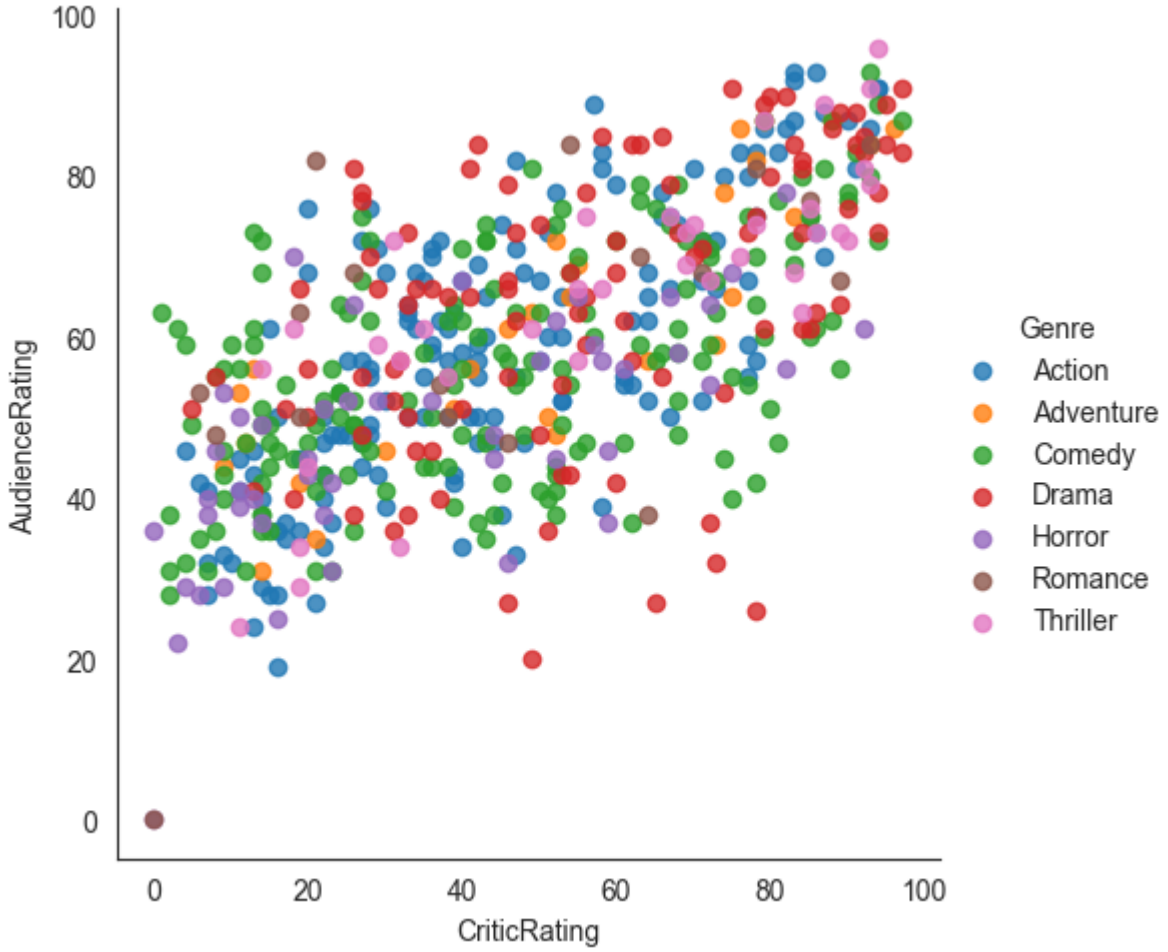
In [38]:
```
for gen in movies.Genre.cat.categories:
    print(gen)
```

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

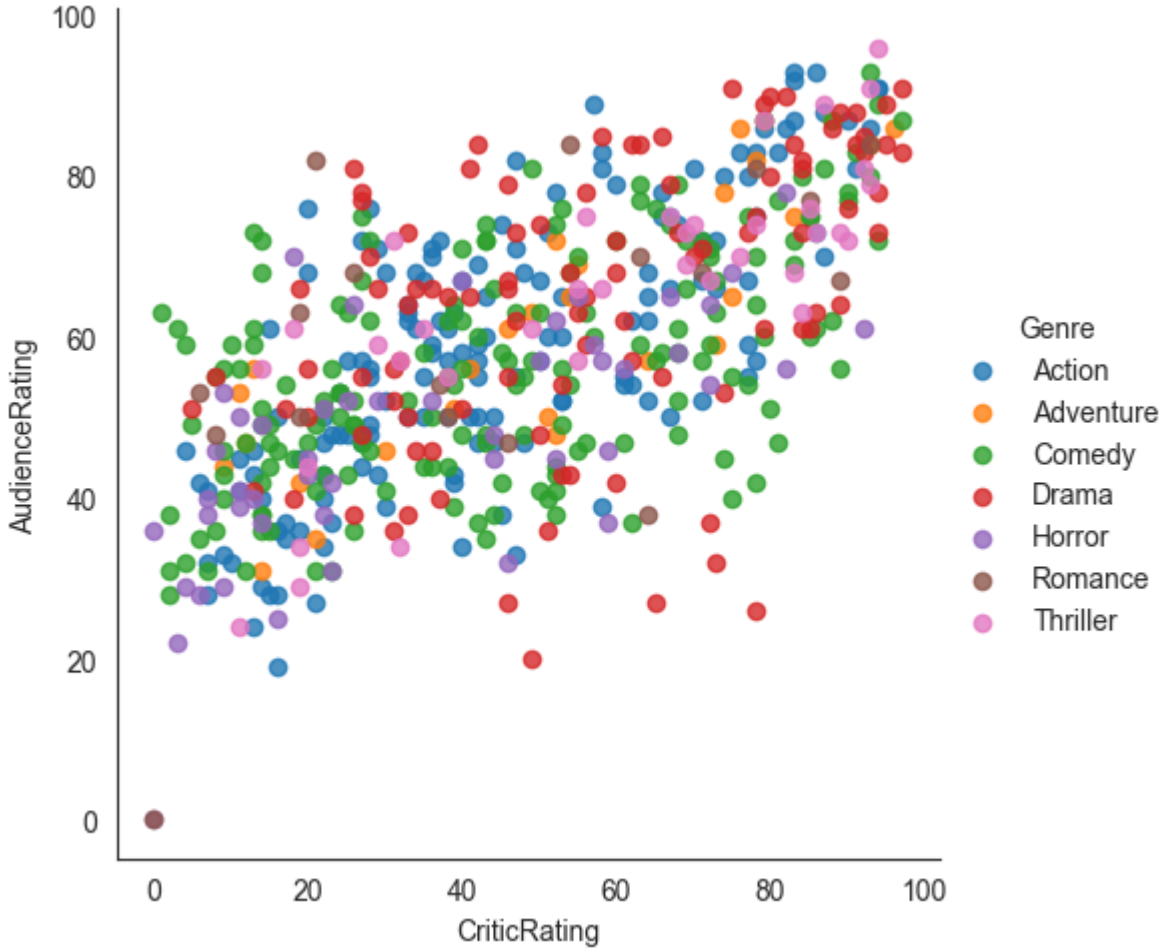
In [39]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',fit_reg=Fals
```



In [40]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',fit_reg=Fals
```

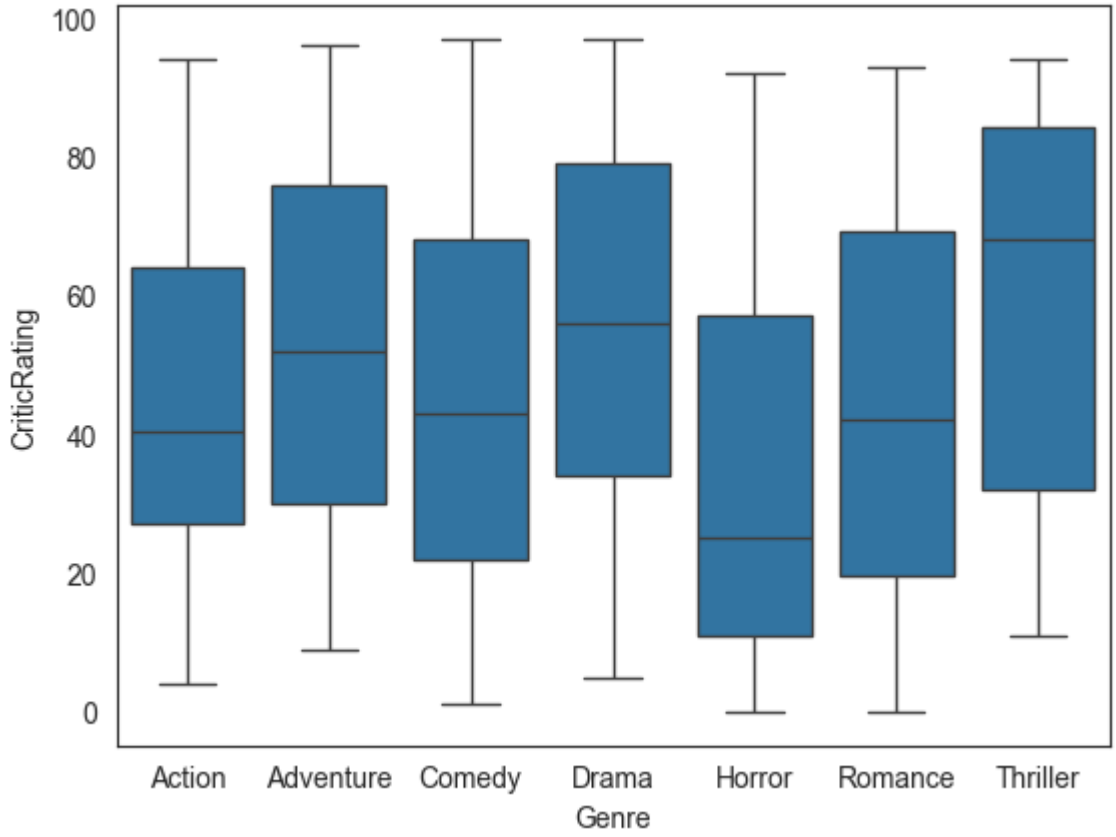


In [45]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',fit_reg=Fals
```

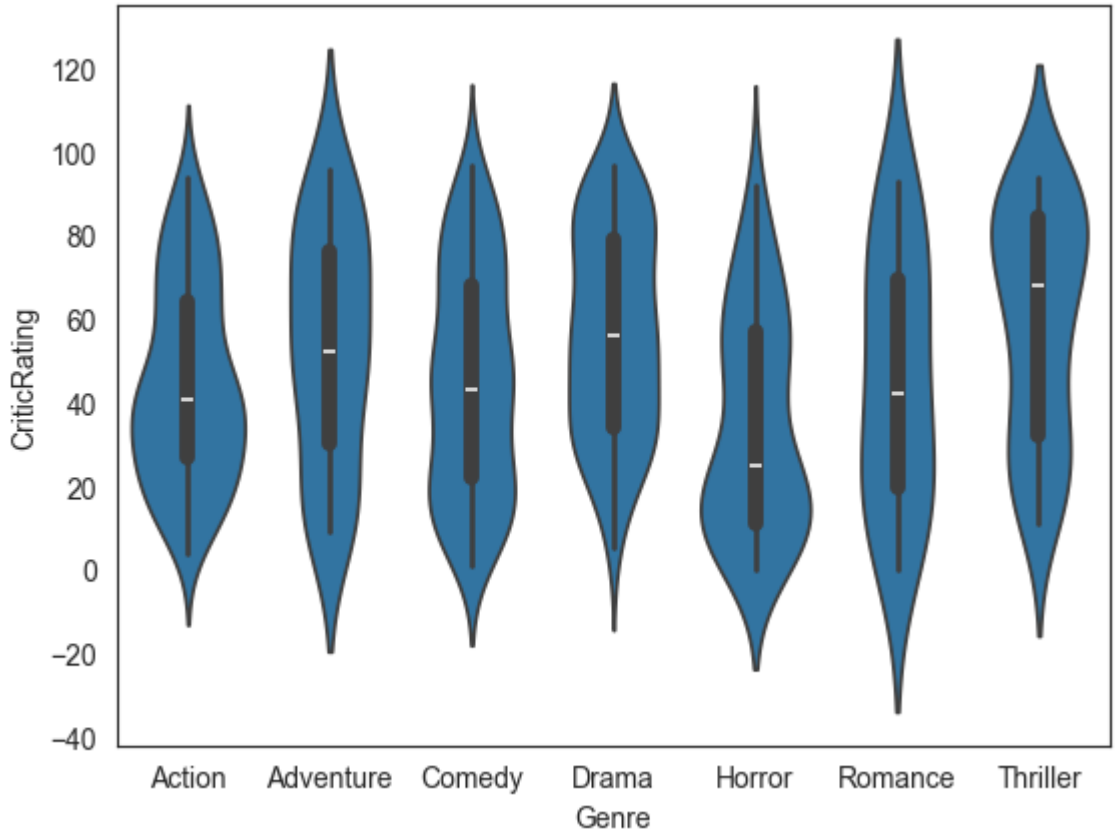


In [46]:
```
g = sns.FacetGrid(movies, row='Genre', col='Year', hue='Genre')
```

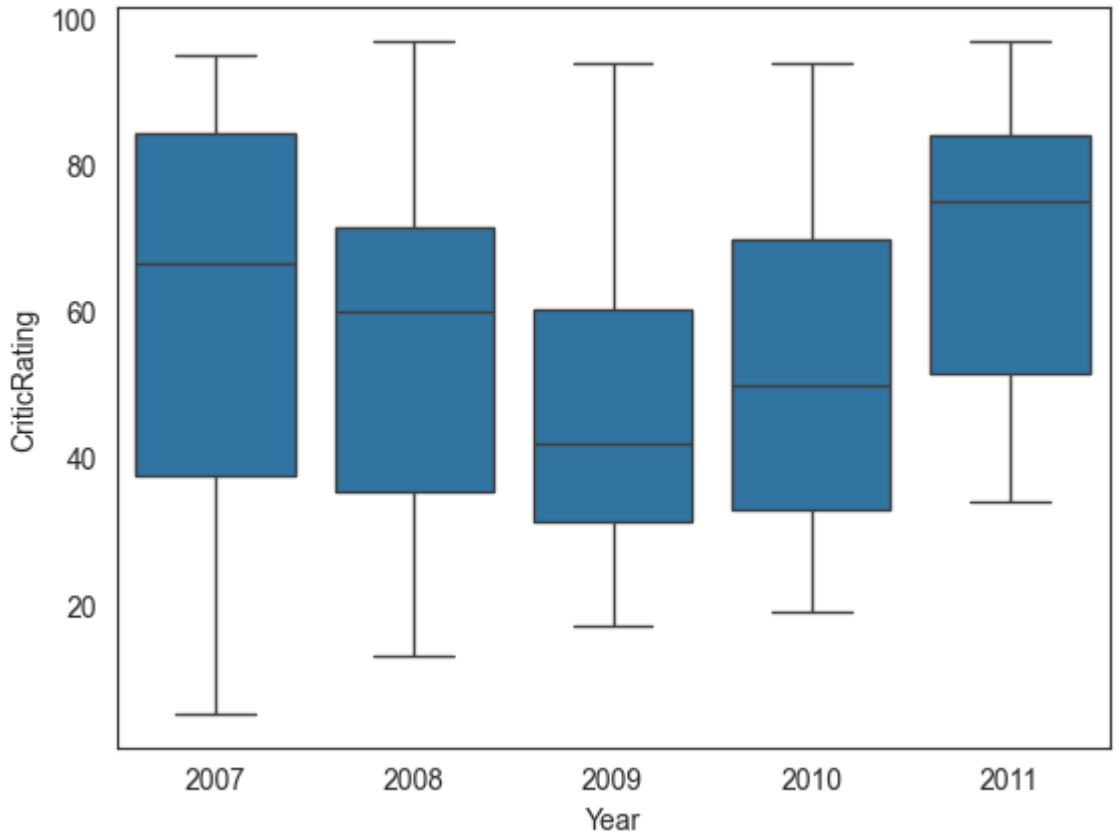
In [ ]:
```
w = sns.boxplot(data=movies, x='Genre', y='CriticRating')
```



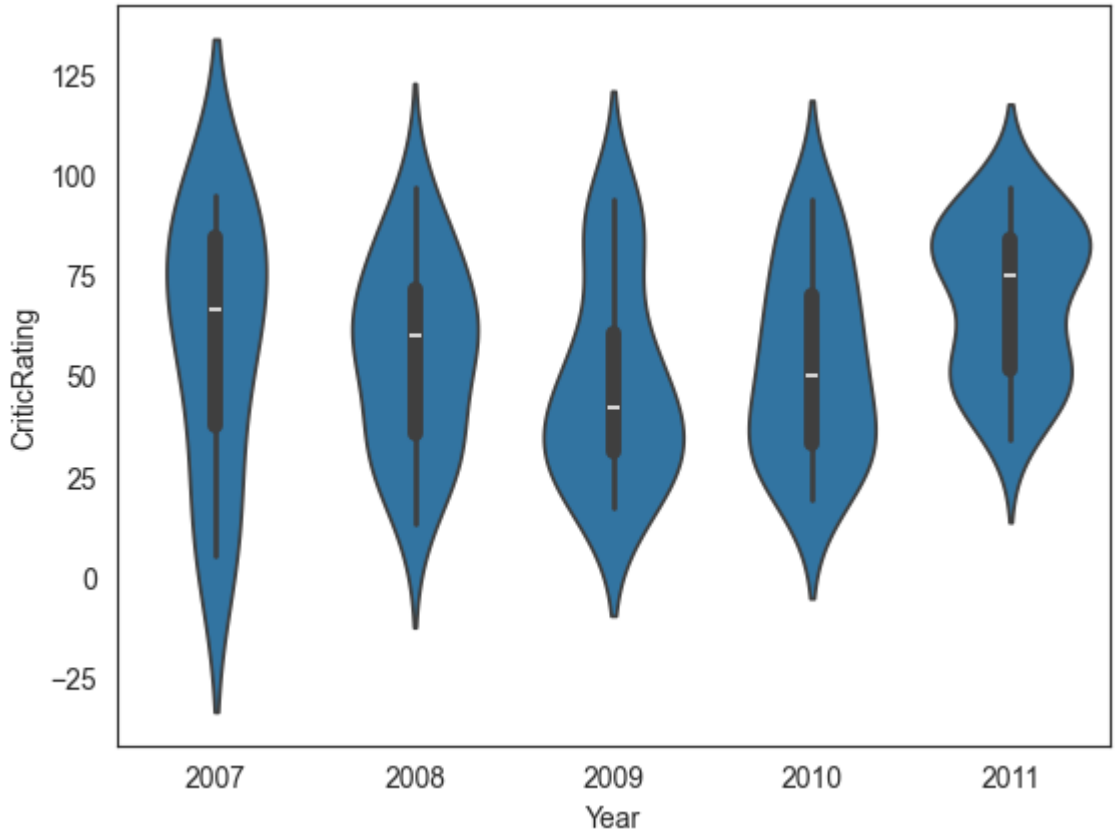
In [49]:
```
w = sns.violinplot(data=movies, x='Genre', y='CriticRating')
```



In [51]:
```
w = sns.boxplot(data=movies[movies.Genre == 'Drama'], x='Year', y='CriticRating'
```

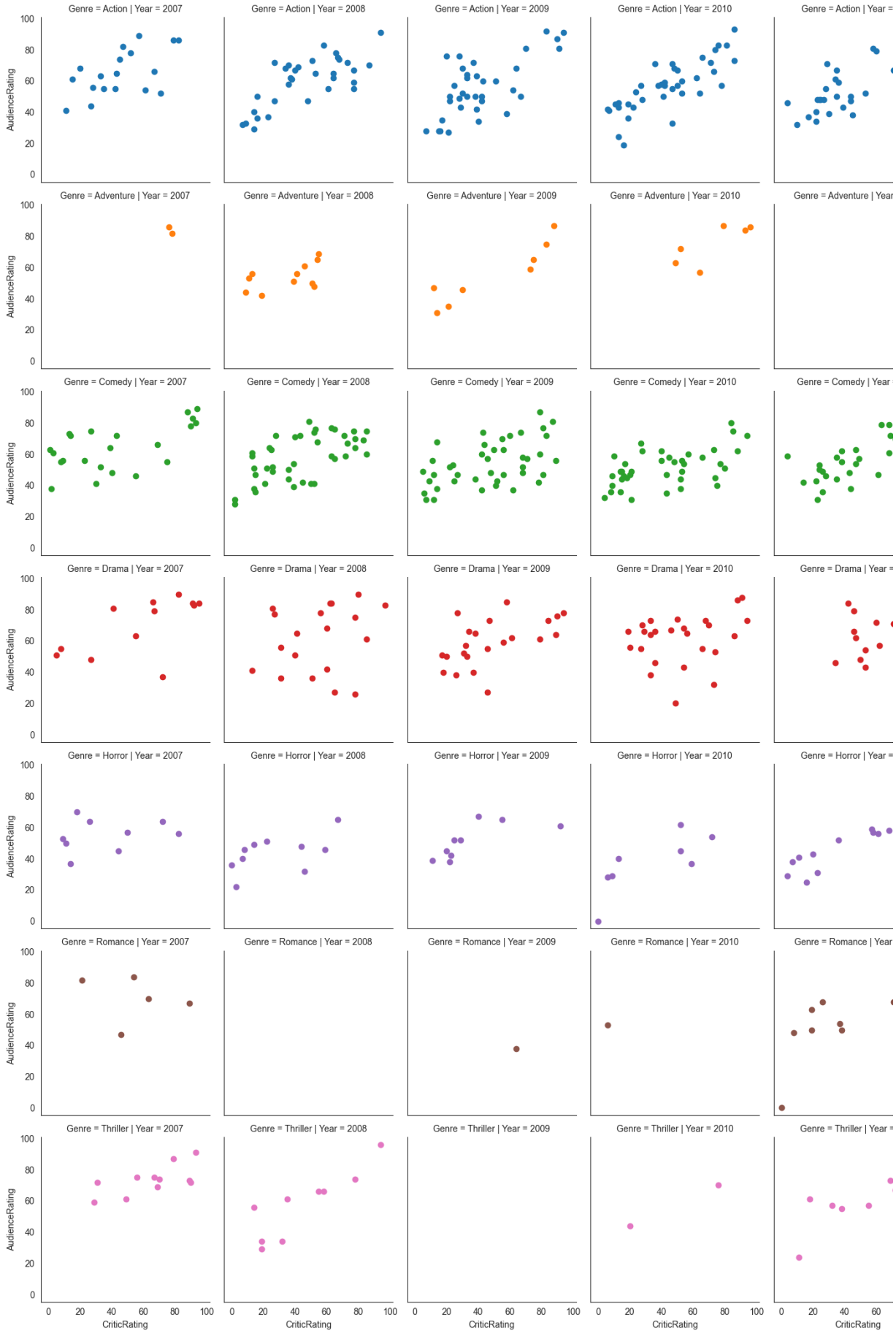


In [52]:
```
w = sns.violinplot(data=movies[movies.Genre == 'Drama'], x='Year', y='CriticRati
```

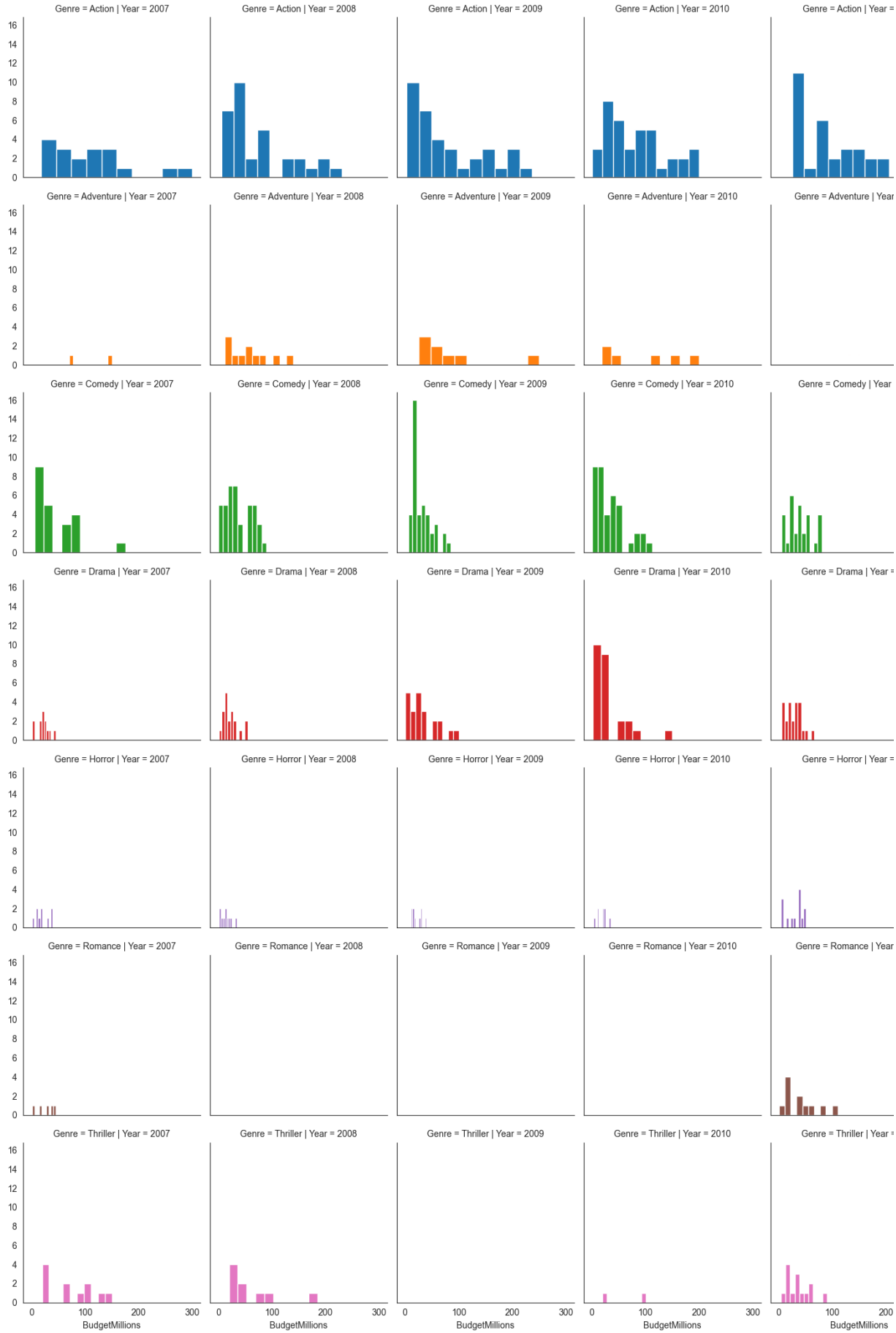


In [54]:
```
g = sns.FacetGrid(movies, row='Genre', col='Year', hue='Genre')
```

In [55]:
```
g = sns.FacetGrid(movies, row='Genre', col='Year', hue='Genre')
g = g.map(plt.scatter, 'CriticRating', 'AudienceRating')
```



In [57]:
```
g = sns.FacetGrid(movies, row ='Genre', col ='Year', hue ='Genre')
g = g.map(plt.hist, 'BudgetMillions')
```


Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year =
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year
Genre = Comedy | Year = 2007
Genre = Comedy | Year = 2008
Genre = Comedy | Year = 2009
Genre = Comedy | Year = 2010
Genre = Comedy | Year
Genre = Drama | Year = 2007
Genre = Drama | Year = 2008
Genre = Drama | Year = 2009
Genre = Drama | Year = 2010
Genre = Drama | Year =
Genre = Horror | Year = 2007
Genre = Horror | Year = 2008
Genre = Horror | Year = 2009
Genre = Horror | Year = 2010
Genre = Horror | Year =
Genre = Romance | Year = 2007
Genre = Romance | Year = 2008
Genre = Romance | Year = 2009
Genre = Romance | Year = 2010
Genre = Romance | Year
Genre = Thriller | Year = 2007
Genre = Thriller | Year = 2008
Genre = Thriller | Year = 2009
Genre = Thriller | Year = 2010
Genre = Thriller | Year =
0 2 4 6 8 10 12 14 16
0 100 200 300
BudgetMillions