In [109…

```
import pandas as pd
```

In [110…

```
import os
```

In [111…

```
os.getcwd()
```

Out[111…

```
'C:\\Users\\DELL\\FSDS'
```

In [112…

```
movies=pd.read_csv(r"D:\Data Science with AI\Data Science With AI\24th, 25th-jul
```

In [113…

```
movies
```

Out[113…

	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 [114…

```
len(movies)
```

Out[114…

```
559
```

In [115…

```
movies.head()
```

Out[115…

	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 [116…

```
movies.tail()
```

Out[116…

	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 [117…

```
movies.columns
```

Out[117…

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

In [118…

```
movies.columns=['Film','Genre','CriticRating','AudienceRating','BudgetMillions',
```

In [119…

```
movies.head()
```

Out[119…

	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 [120…

```
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 [121…

```
movies.describe()
```

Out[121…

	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 [122…

```
movies['Film']
```

Out[122…

```
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 [123…

```
movies.Film
```

Out[123…

```
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 [124…

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

In [125…

```
movies.Film
```

Out[125…

```
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 [126…

```
movies.head()
```

Out[126…

	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 [127…

```
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 [128…

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

In [129…

```
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 [130…

```
movies.Genre
```

Out[130…

```
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 [131…

```
movies.Year
```

Out[131…

```
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 [132…

```
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 [133…

```
movies.Genre.cat.categories
```

Out[133…

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

In [134…

```
movies.Year.cat.categories
```

Out[134…

```
Index([2007, 2008, 2009, 2010, 2011], dtype='int64')
```

In [135…

```
movies.describe()
```

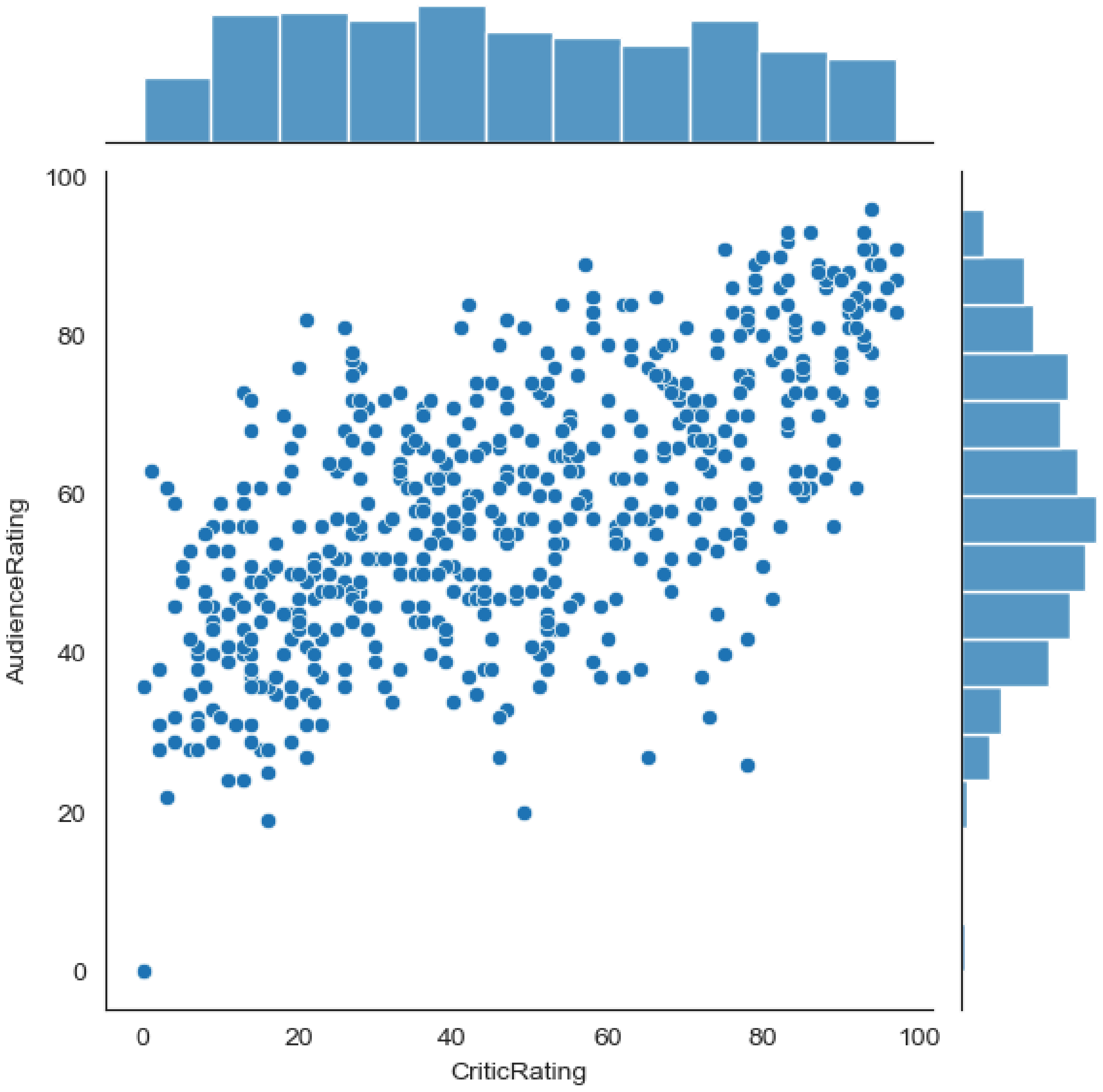
Out[135…

	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 [136…
```
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

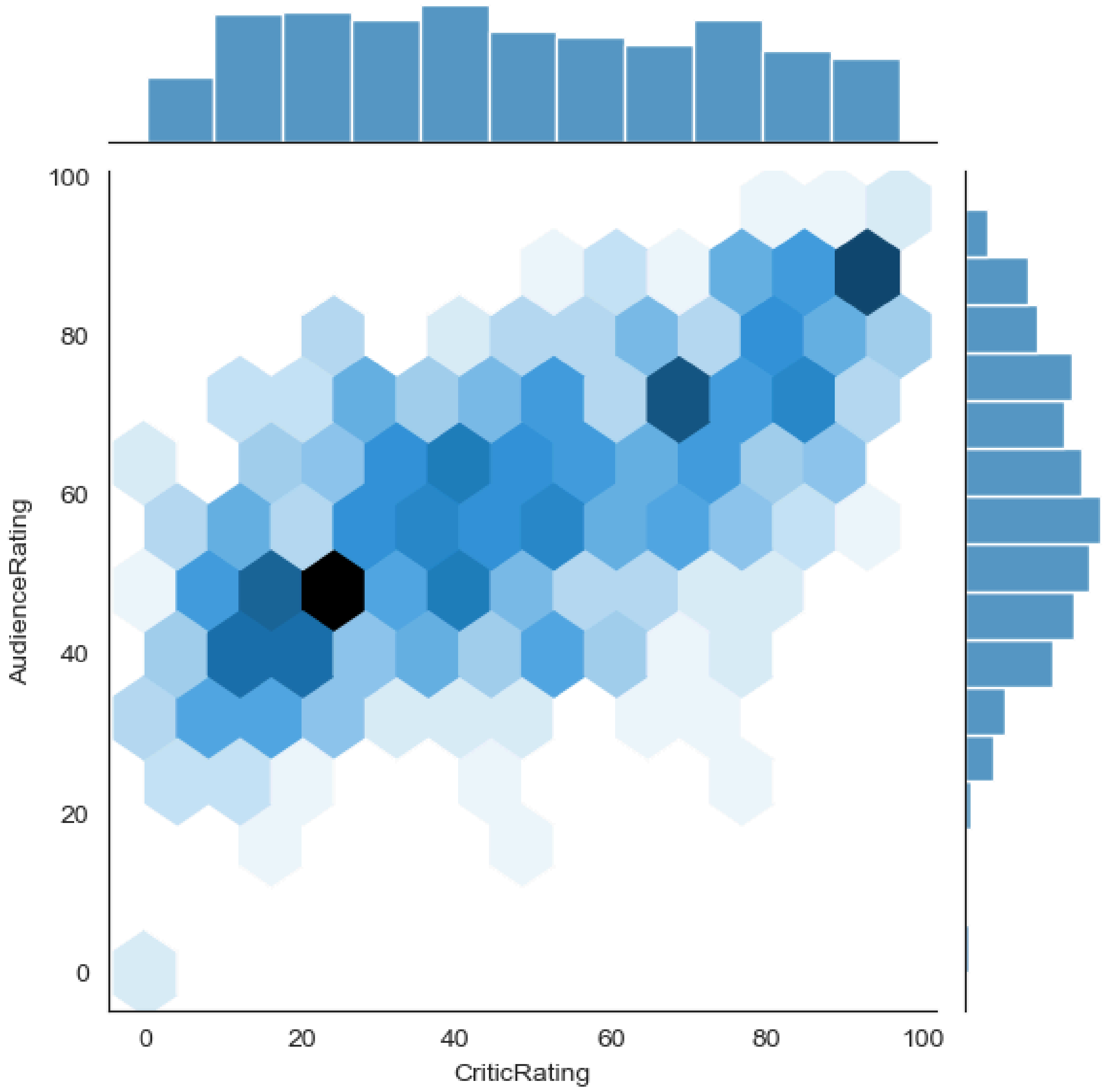
In [137…
```
j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating')
```

In [138…
```
plt.show()
```



In [139…
```
j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='hex')
```

In [140…
```
plt.show()
```

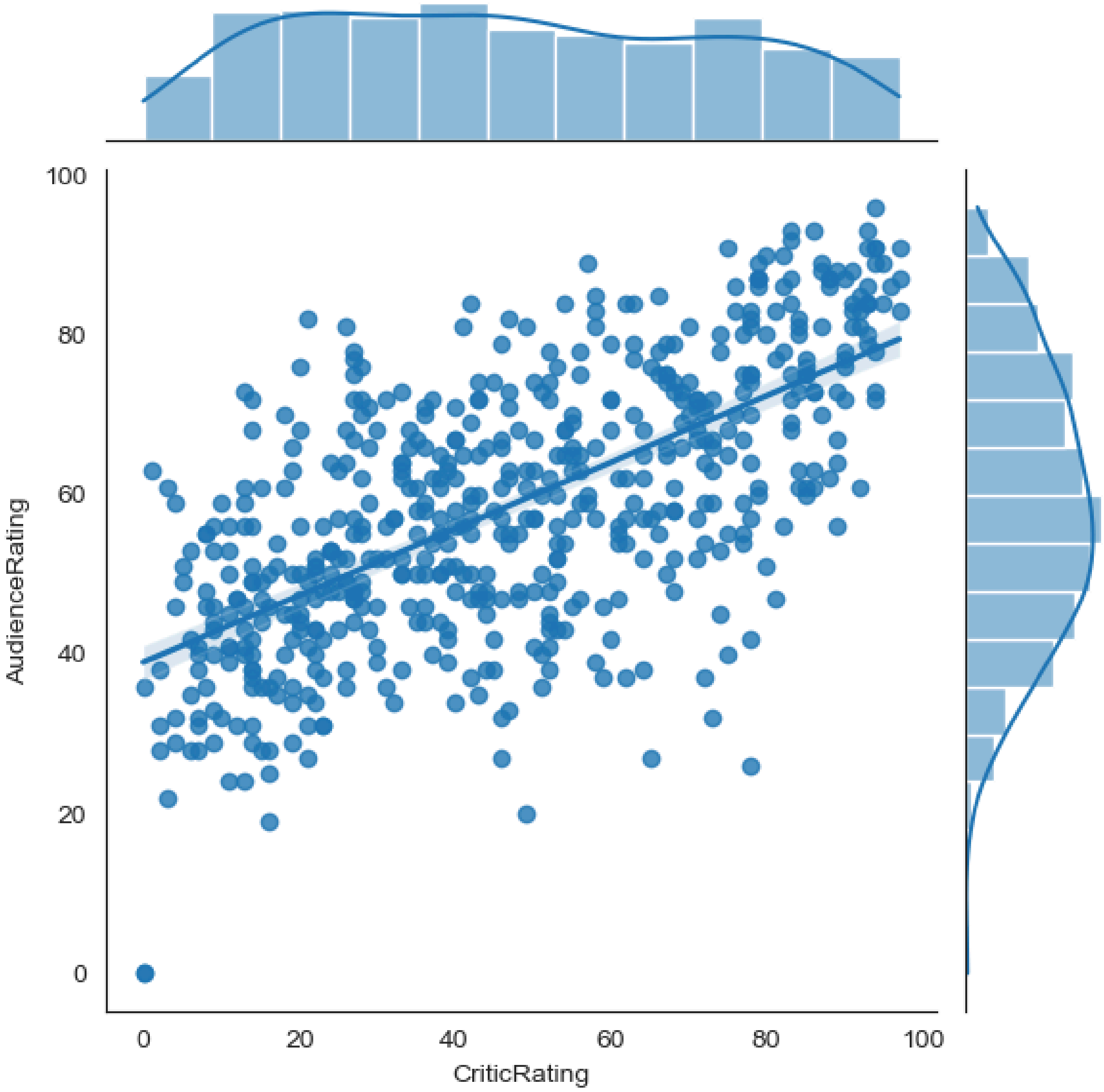


In [141…

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

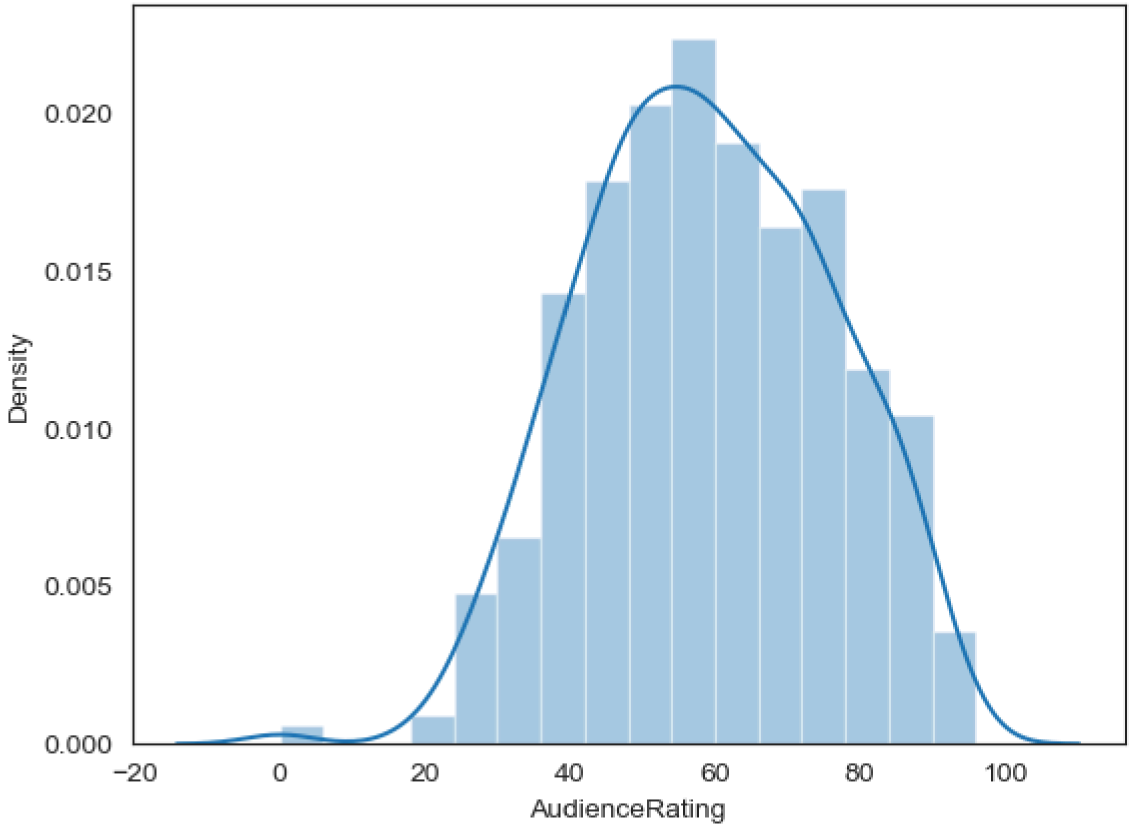
In [142…

```
plt.show()
```



```
In [143…  m1=sns.distplot(movies['AudienceRating'])
```

```
In [144…  plt.show()
```



In [145…

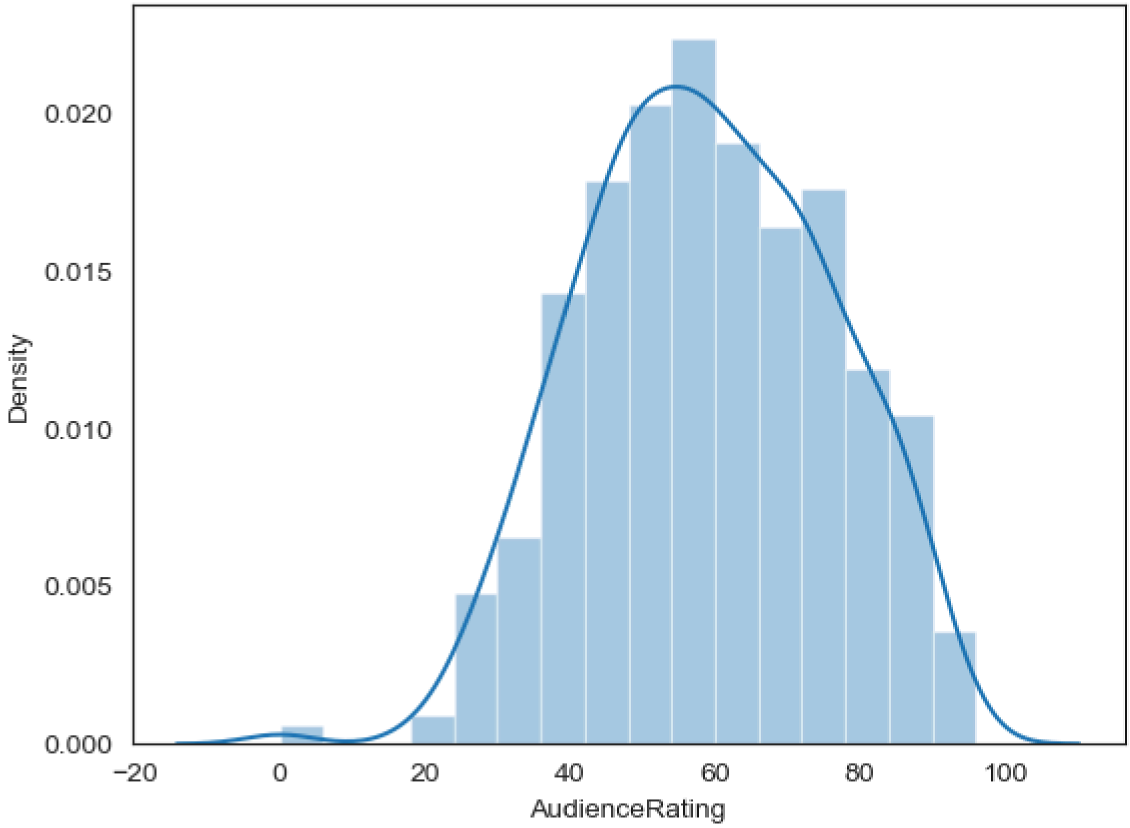
```
sns.distplot(movies.AudienceRating)
```

Out[145…

```
<Axes: xlabel='AudienceRating', ylabel='Density'>
```

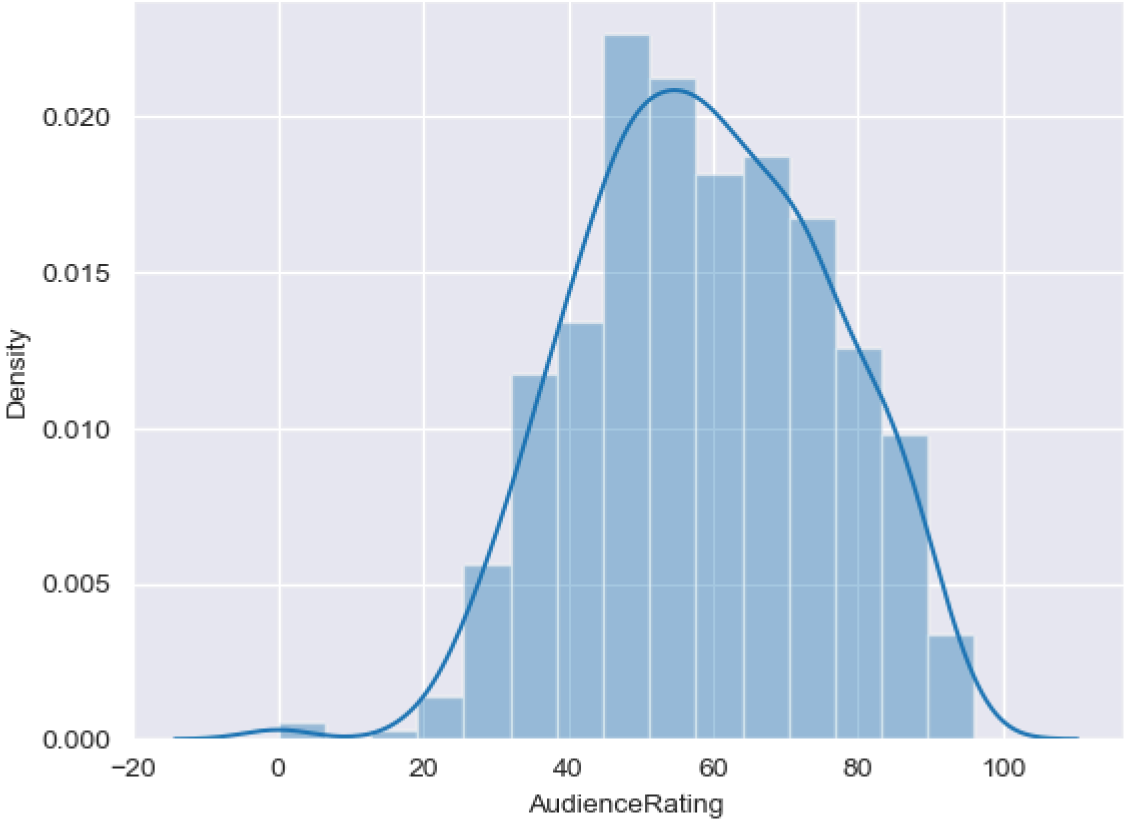
In [146…

```
plt.show()
```



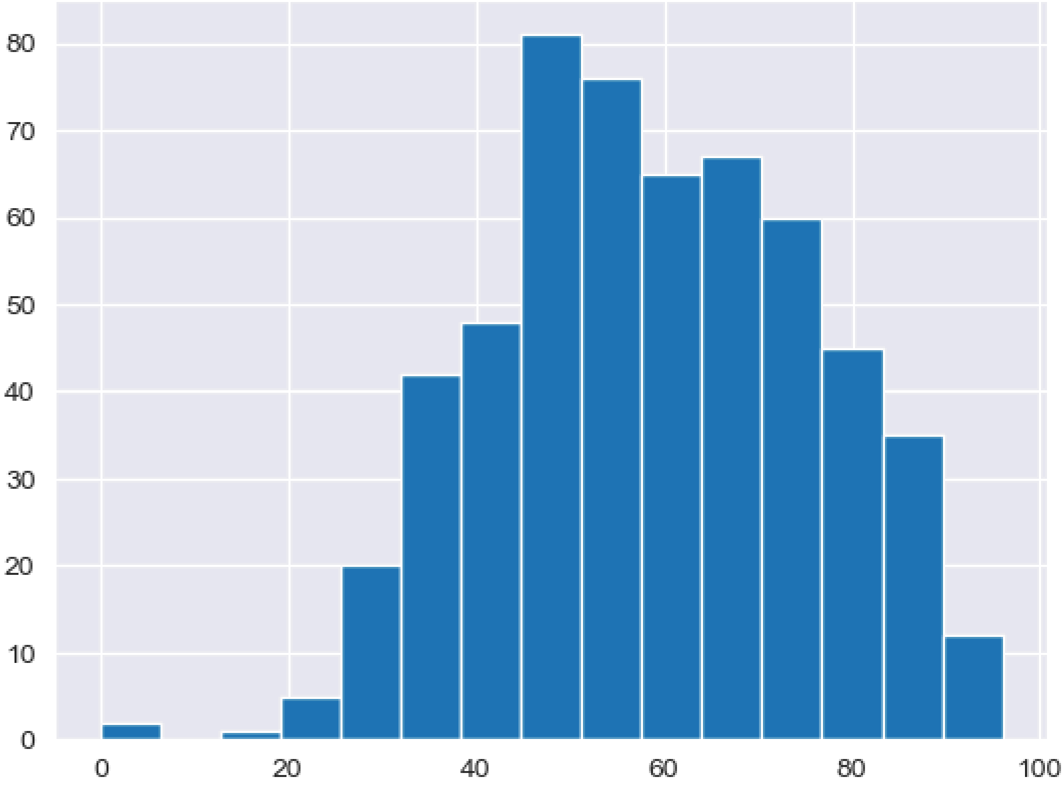
In [147…

```
sns.set_style('darkgrid')
m2=sns.distplot(movies.AudienceRating,bins=15)
plt.show()
```



In [148…

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

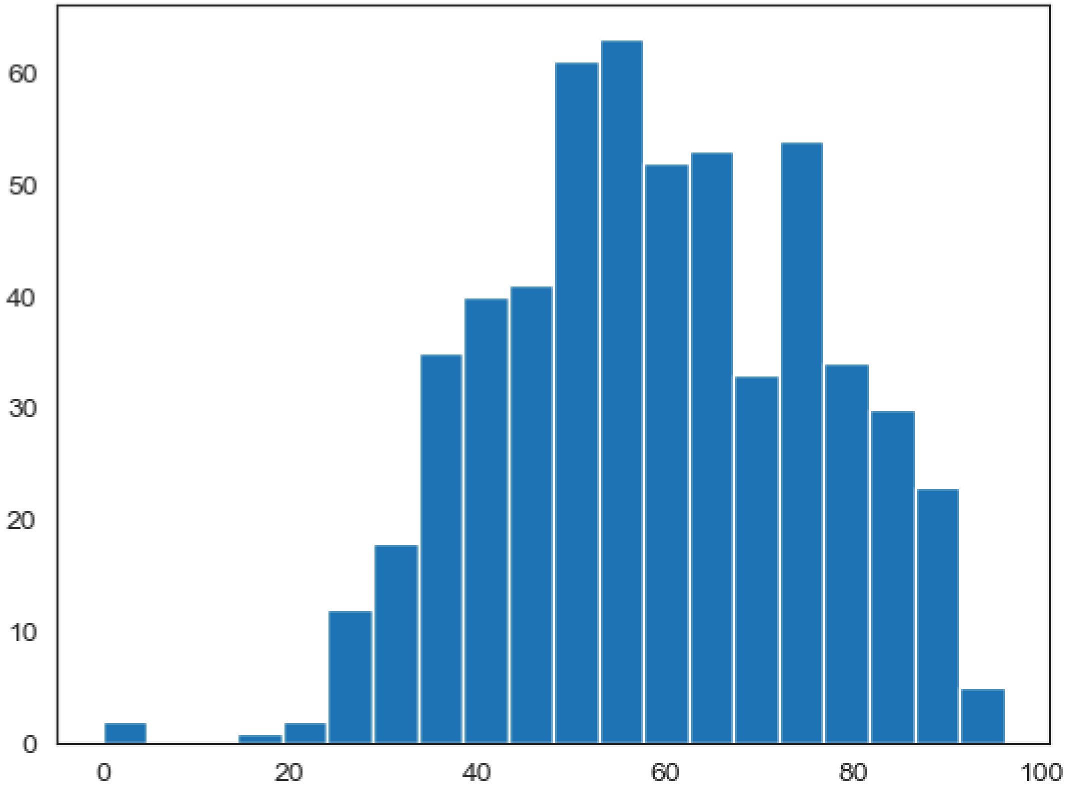


In [149…

```
sns.set_style('white')
n1=plt.hist(movies.AudienceRating,bins=20)
```

In [150…

```
plt.show()
```

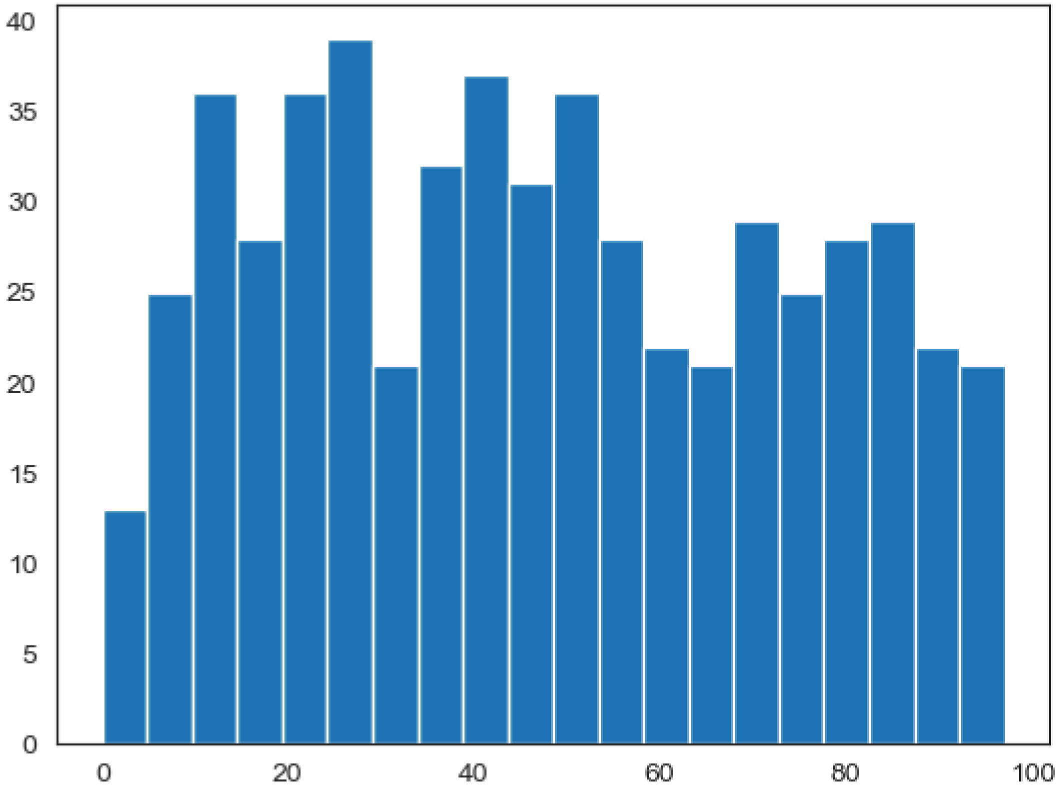


In [151…

```
n1=plt.hist(movies.CriticRating,bins=20)
```

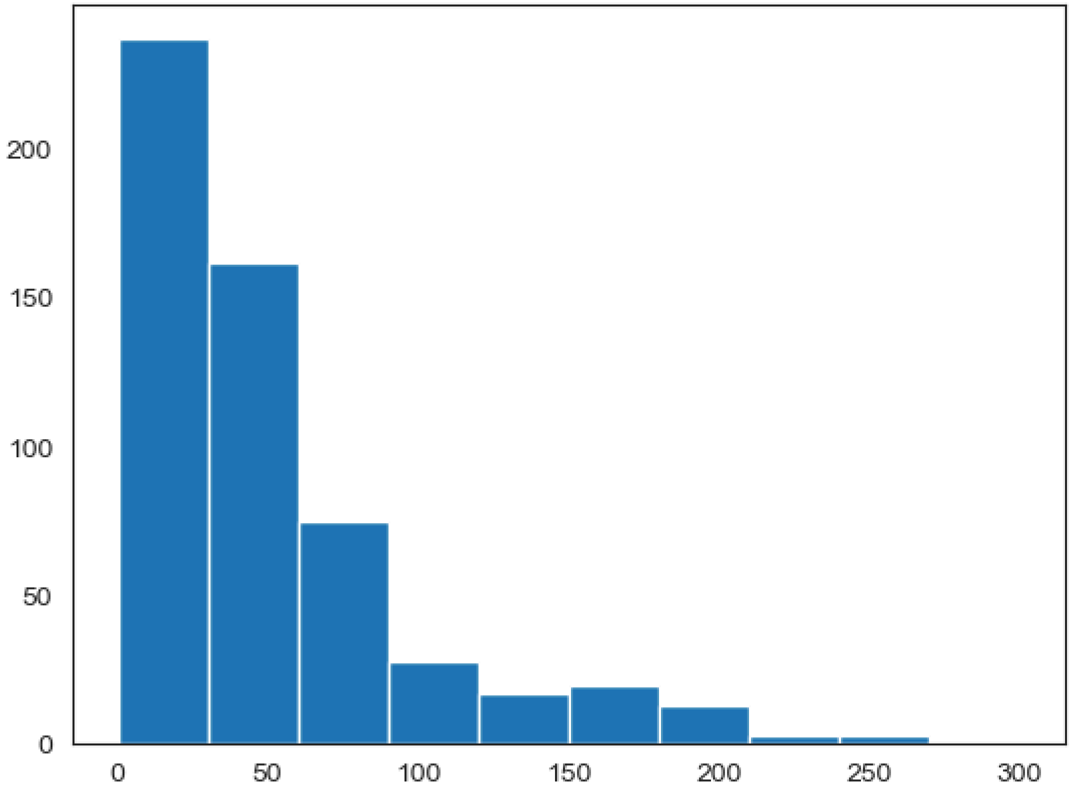
In [152…

```
plt.show()
```



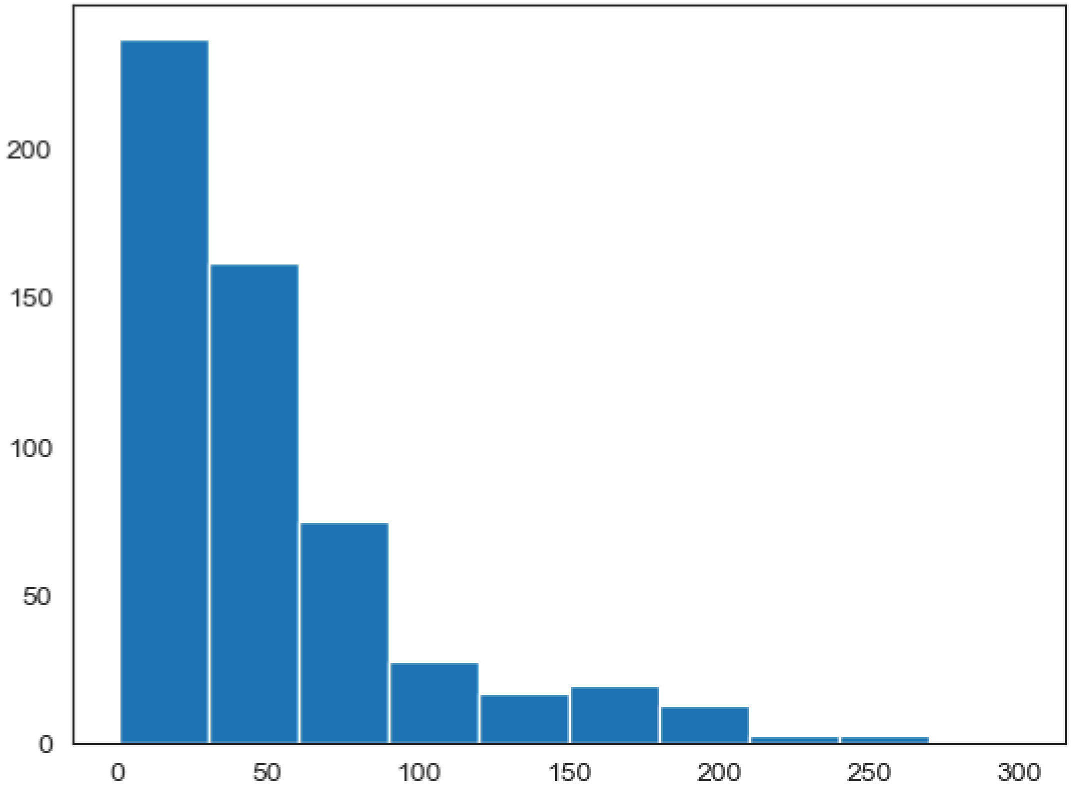
In [153…

```
h1=plt.hist(movies.BudgetMillions)
plt.show()
```



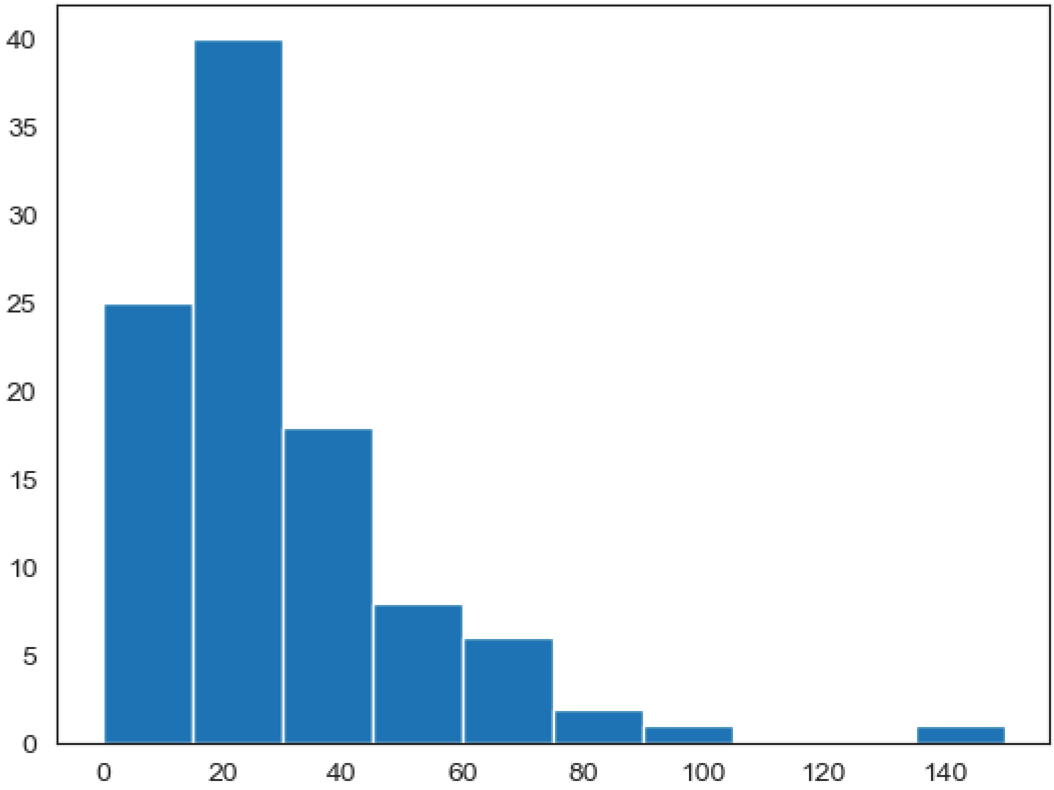
In [154…

```
plt.hist(movies.BudgetMillions)
plt.show()
```



In [155…

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



In [156…

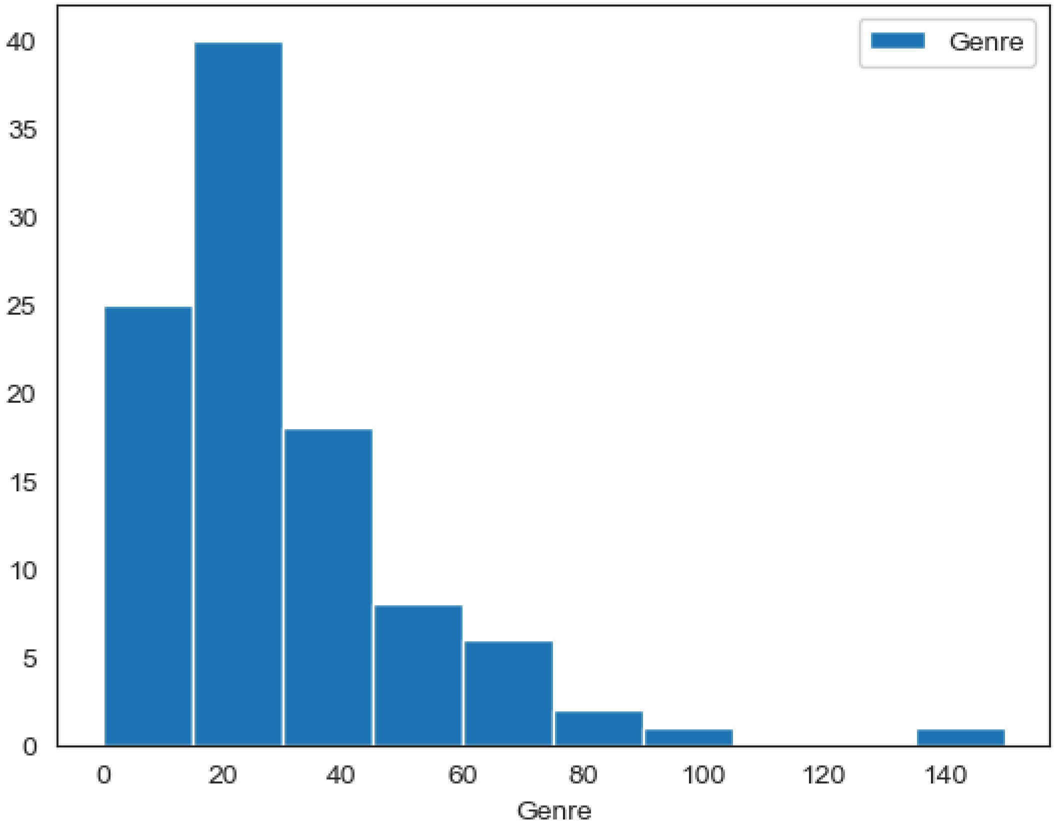
```
plt.hist(movies(movies.Genre=='Drama').BudgetMillions,label='Genre')
plt.xlabel('year')
plt.legend()
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[156], line 1
----> 1 plt.hist(movies(movies.Genre=='Drama').BudgetMillions,label='Genre')
      2 plt.xlabel('year')
      3 plt.legend()

TypeError: 'DataFrame' object is not callable
```

In [157…

```
plt.hist(movies[movies.Genre=='Drama'].BudgetMillions,label='Genre')
plt.xlabel('Genre')
plt.legend()
plt.show()
```



In [158…

```
plt.hist(movies.BudgetMillions,label='Genre','year')
plt.legend()
plt.show()
```

```
  Cell In[158], line 1
    plt.hist(movies.BudgetMillions,label='Genre','year')
                                                       ^
SyntaxError: positional argument follows keyword argument
```

In [159…

```
movies
```

Out[159…

	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
...	...	...	...	...	...	...
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 [160…

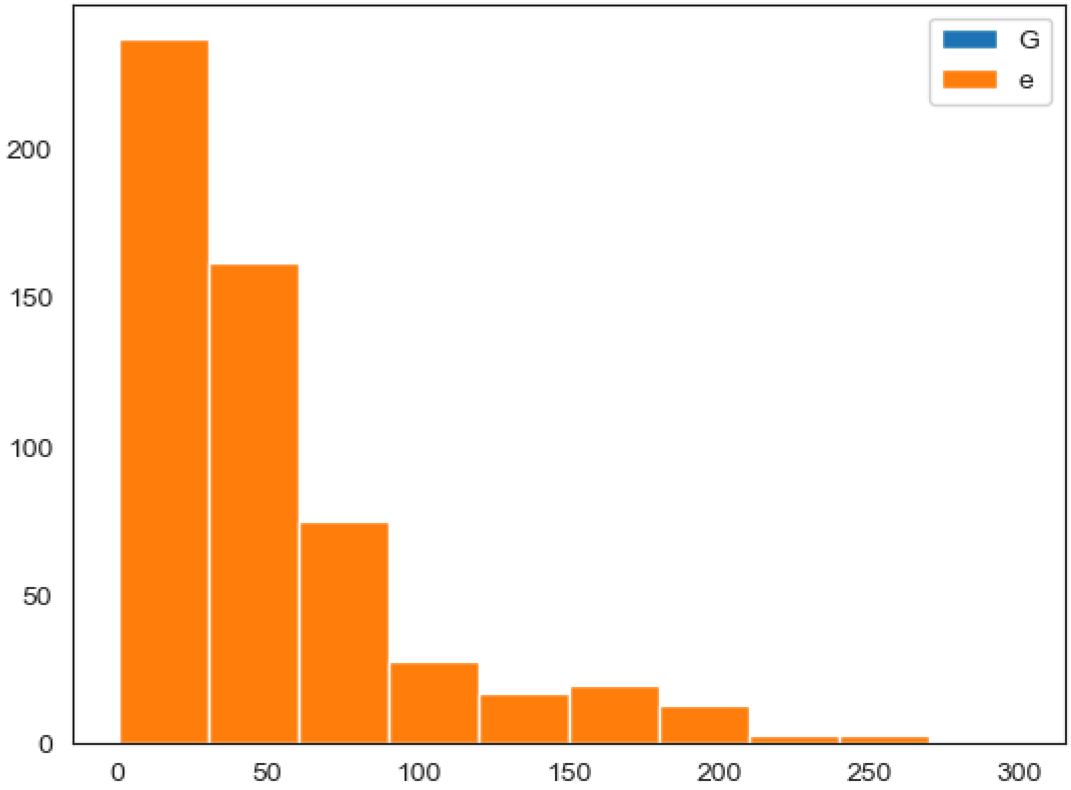
```
plt.hist(movies.BudgetMillions)
plt.legend(Genre)
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[160], line 2
      1 plt.hist(movies.BudgetMillions)
----> 2 plt.legend(Genre)
      3 plt.show()

NameError: name 'Genre' is not defined
```

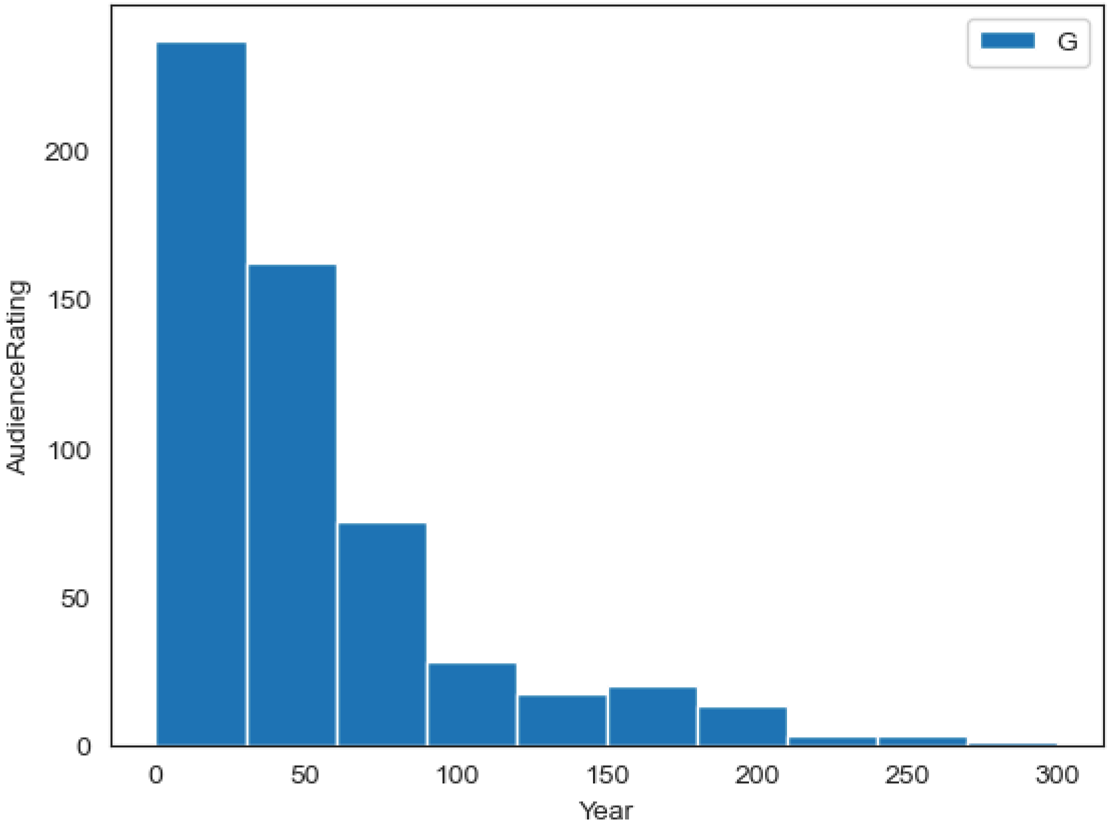
In [161…

```
plt.hist(movies.BudgetMillions)
plt.legend('Genre')
plt.show()
```



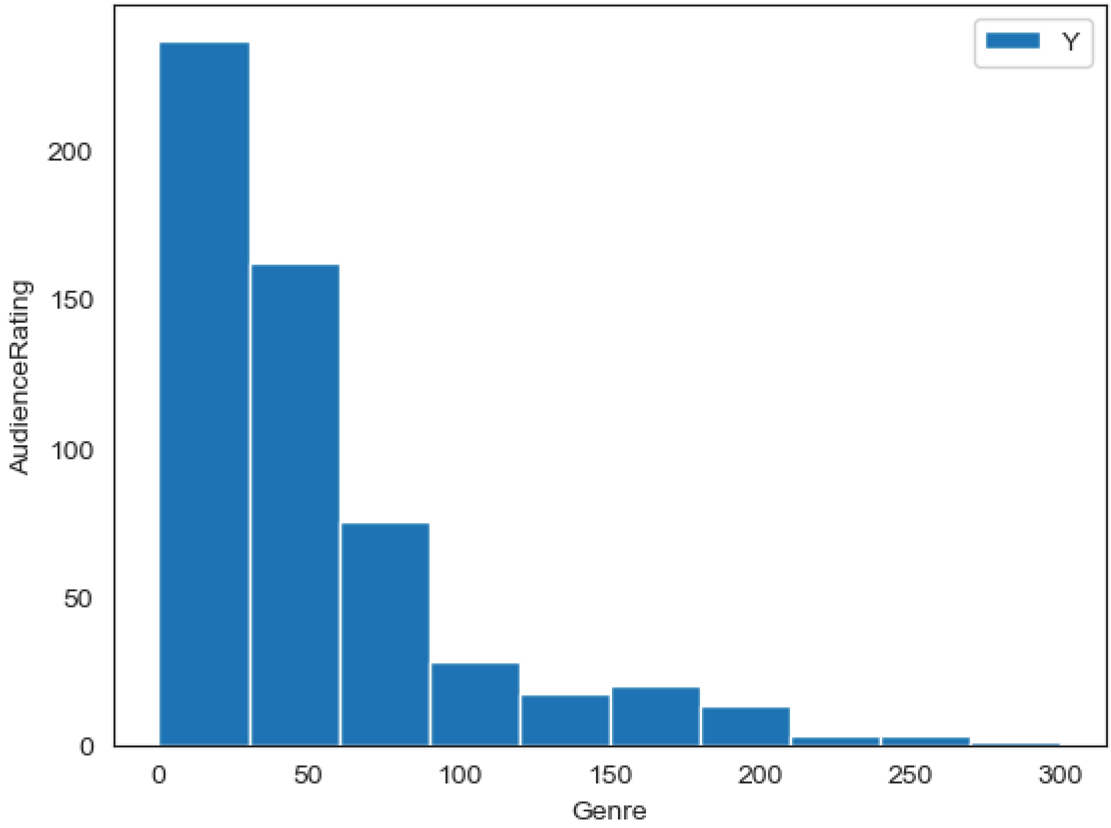
In [162…

```
plt.hist(movies.BudgetMillions)
plt.xlabel('Year')
plt.ylabel('AudienceRating')
plt.legend('Genre')
plt.show()
```



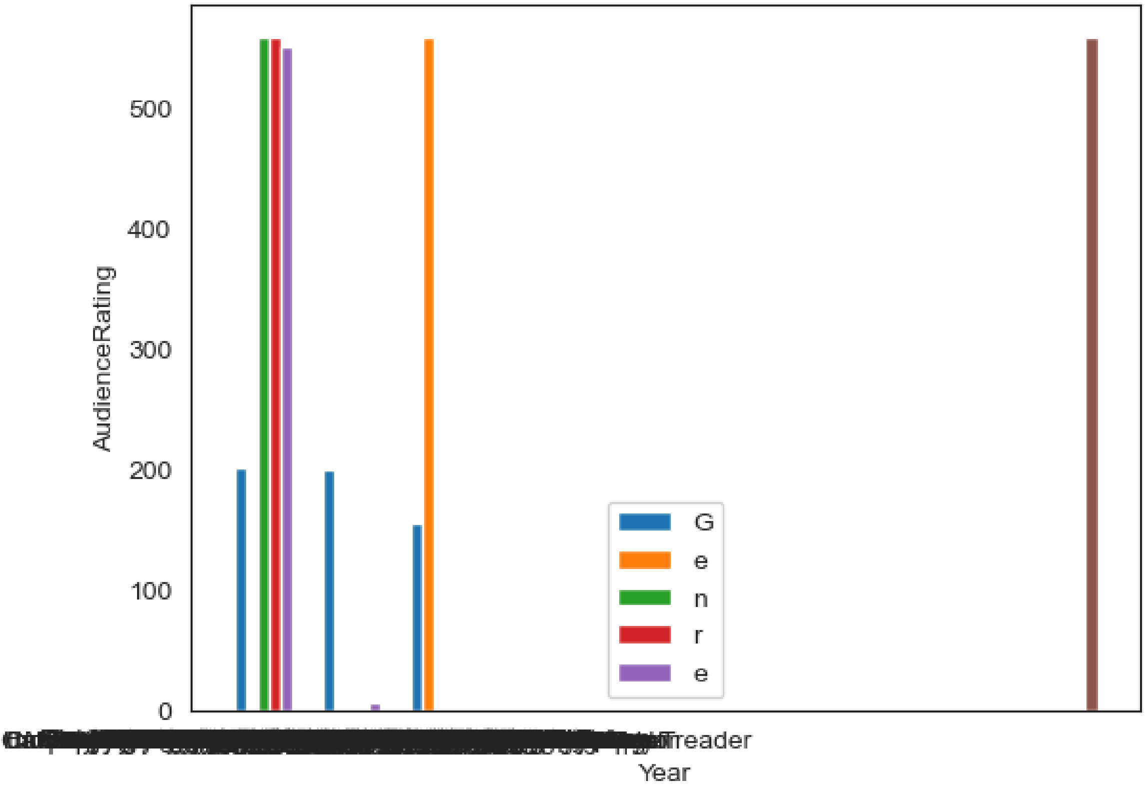
In [163…

```
plt.hist(movies.BudgetMillions)
plt.xlabel('Genre')
plt.ylabel('AudienceRating')
plt.legend('Year')
plt.show()
```



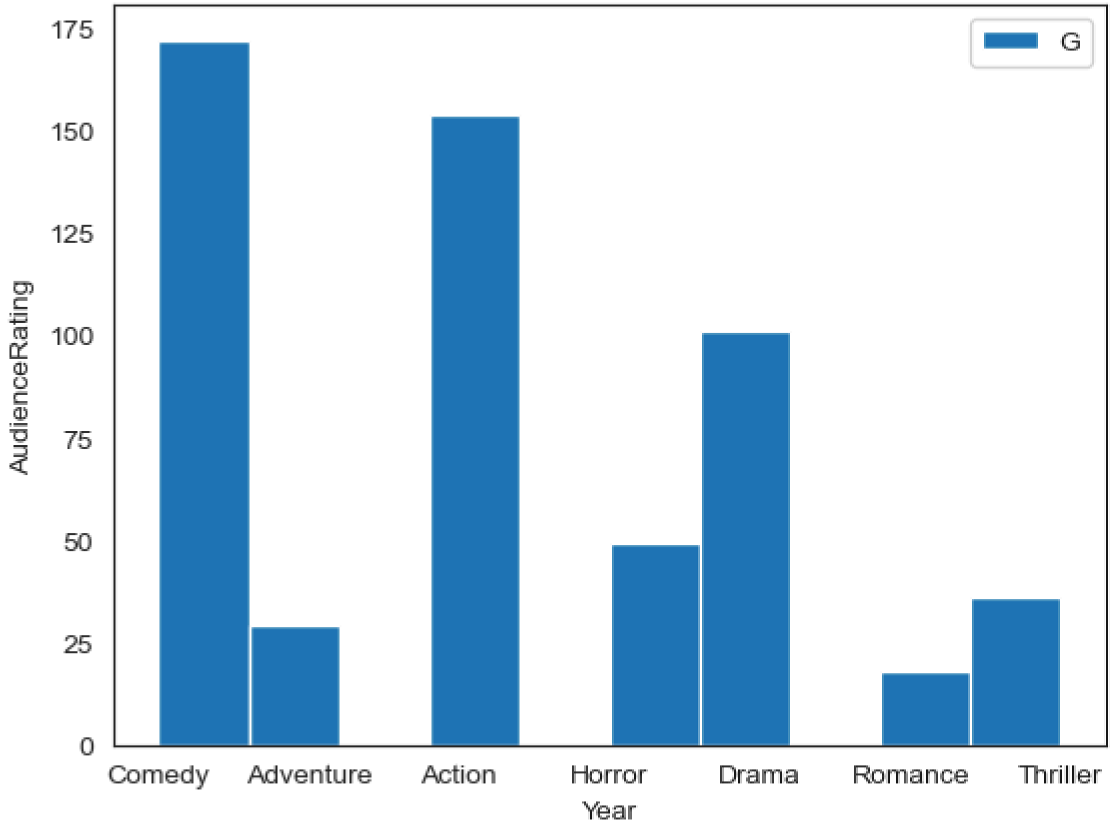
In [164…

```
plt.hist(movies)
plt.xlabel('Year')
plt.ylabel('AudienceRating')
plt.legend('Genre')
plt.show()
```



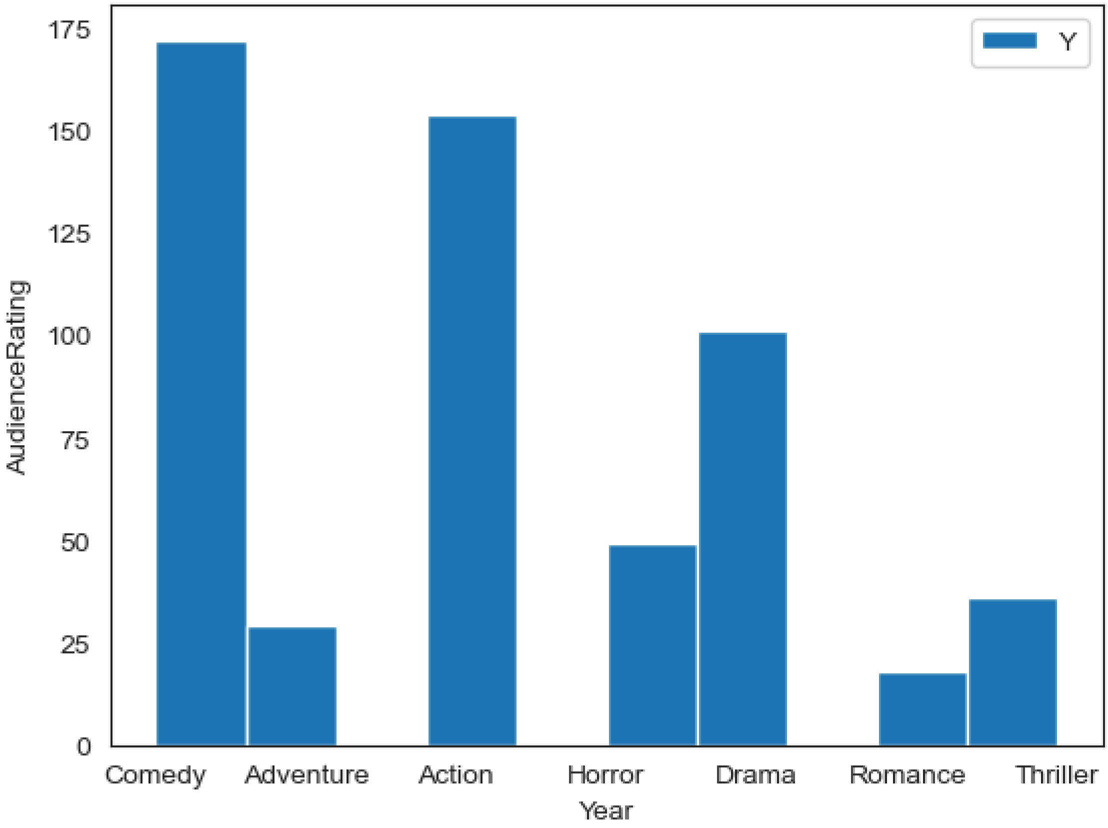
In [165…

```
plt.hist(movies.Genre)
plt.xlabel('Year')
plt.ylabel('AudienceRating')
plt.legend('Genre')
plt.show()
```



In [166…

```
plt.hist(movies.Genre)
plt.xlabel('Year')
plt.ylabel('AudienceRating')
plt.legend('Year')
plt.show()
```



In [167…

```
plt.hist(movies.[movies.Genre].Year)
plt.xlabel('Year')
plt.ylabel('AudienceRating')
plt.legend('Year')
plt.show()
```

```
  Cell In[167], line 1
    plt.hist(movies.[movies.Genre].Year)
                    ^
SyntaxError: invalid syntax
```

In [ ]:

```
plt.hist(movies[movies.Genre].Year)
plt.xlabel('Year')
plt.ylabel('AudienceRating')
plt.legend('Year')
plt.show()
```

In [ ]:

```
movies.head()
```

In [ ]:

```
movies.Genre.unique()
```

In [ ]:

```
movies.Genre.nunique()
```

In [ ]:

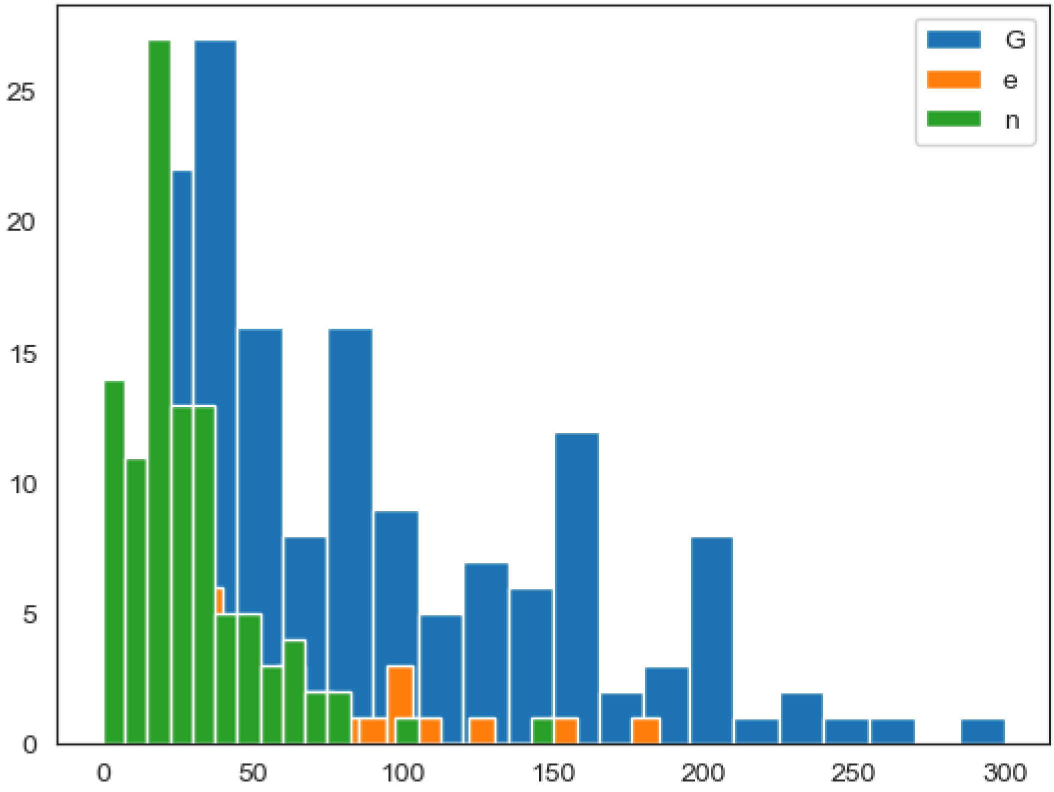
```
plt.hist(movies[movies.Genre=='Action'].BudgetMillions,bins=20)
plt.hist(movies[movies.Genre=='Thriller'].BudgetMillions,bins=20)
plt.hist(movies[movies.Genre=='Drama'].BudgetMillions,bins=20)
plt.legend()
plt.show()
```

In [ ]:

```
movies.Genre[movies.BudgetMillions]
```

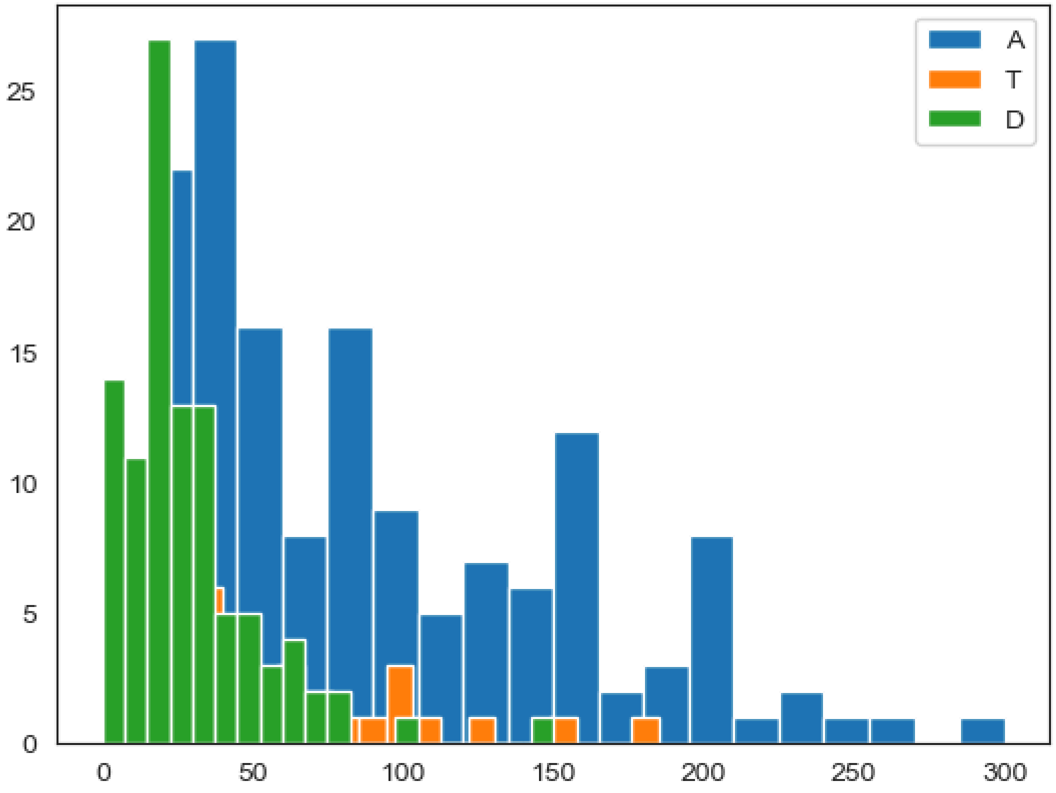
In [168…

```
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('Genre')
plt.show()
```



In [169…

```
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('ATD')
plt.show()
```



In [170…

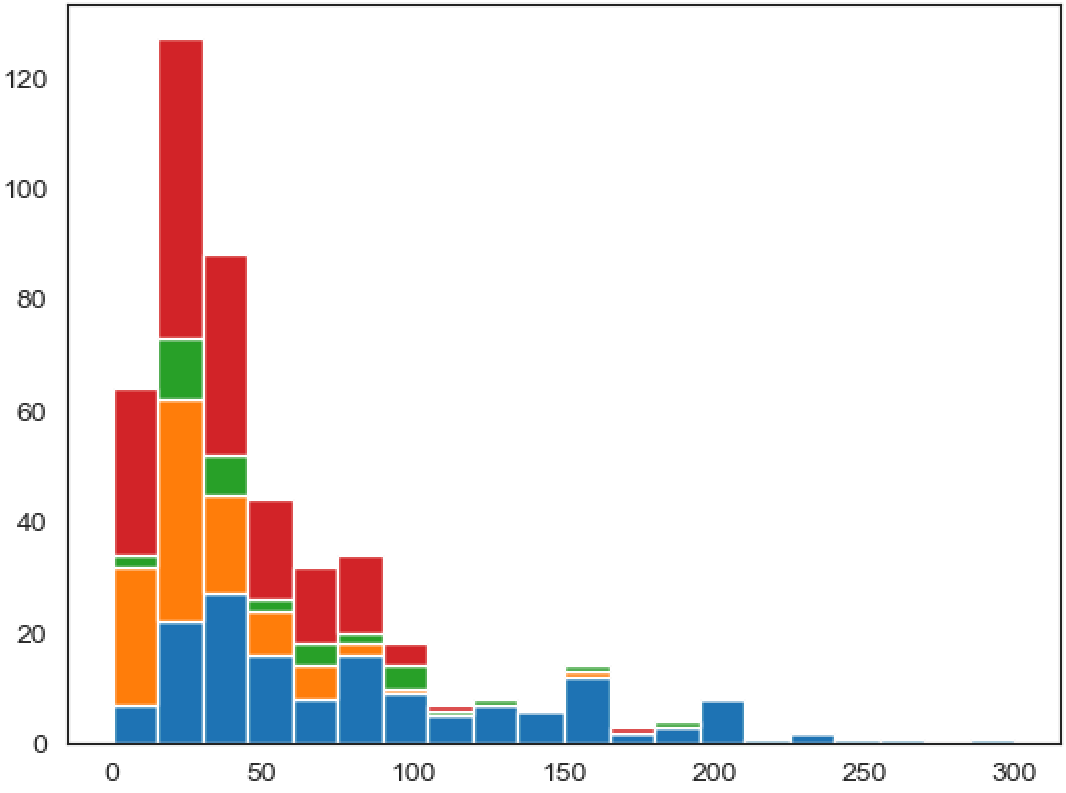
```
movies.Genre[movies.BudgetMillions]
```

Out[170…

```
8          Comedy
105      Thriller
20         Horror
18          Drama
20         Horror
          ...
50         Comedy
18          Drama
65      Adventure
24      Adventure
80         Comedy
Name: Genre, Length: 559, dtype: category
Categories (7, object): ['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'R
omance', 'Thriller']
```

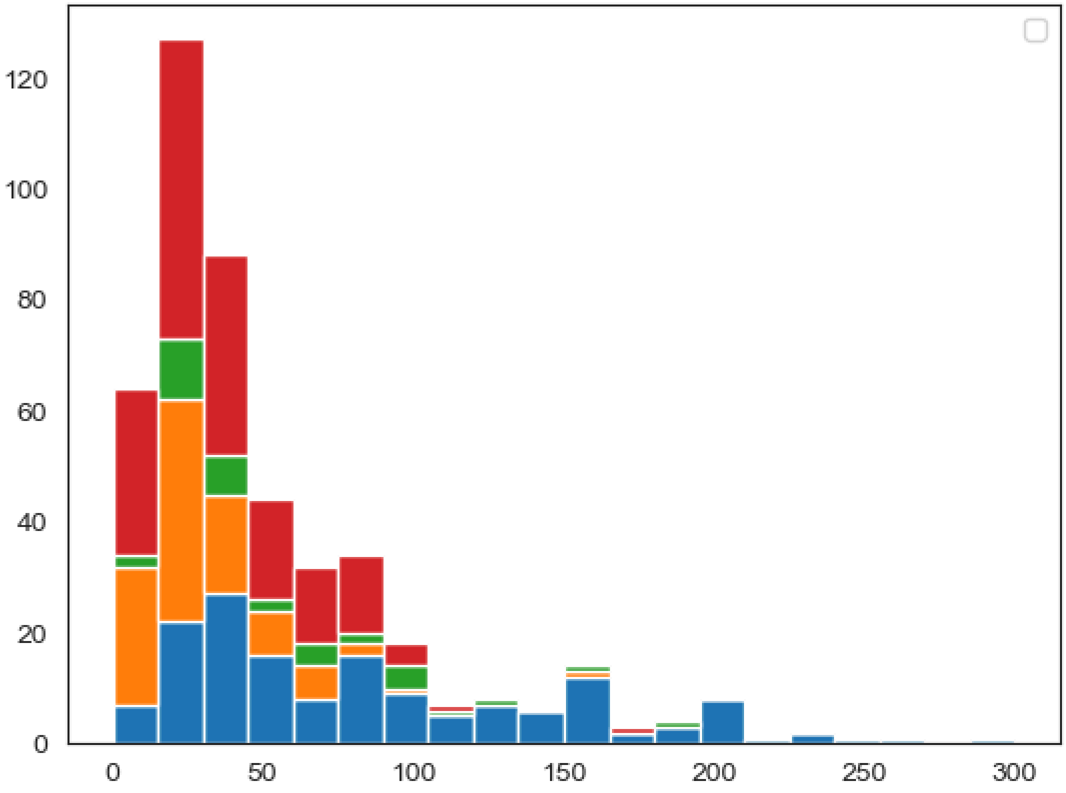
In [171…

```
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 [172…

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



In [173…

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

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

In [174…

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

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[174], line 1
----> 1 for gen in Genre.cat.categories:
      2     print(gen)

NameError: name 'Genre' is not defined
```

In [175…

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

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

In [176…

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

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

In [177…

```
movies.Genre.cat.categories
```

Out[177…

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

In [178…

```
movies.Genre.cat.category
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[178], line 1
----> 1 movies.Genre.cat.category

AttributeError: 'CategoricalAccessor' object has no attribute 'category'
```

In [179…

```
movies.Genre
```

Out[179…

```
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 [180…

```python
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 [181…

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

In [182…

```python
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 [183…

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

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

In [184…

```
movies.Genre
```

Out[184…

```
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 [185…

```
movies.Genre.cat.categories
```

Out[185…

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

In [186…

```
movies.Year.cat.categories
```

Out[186…

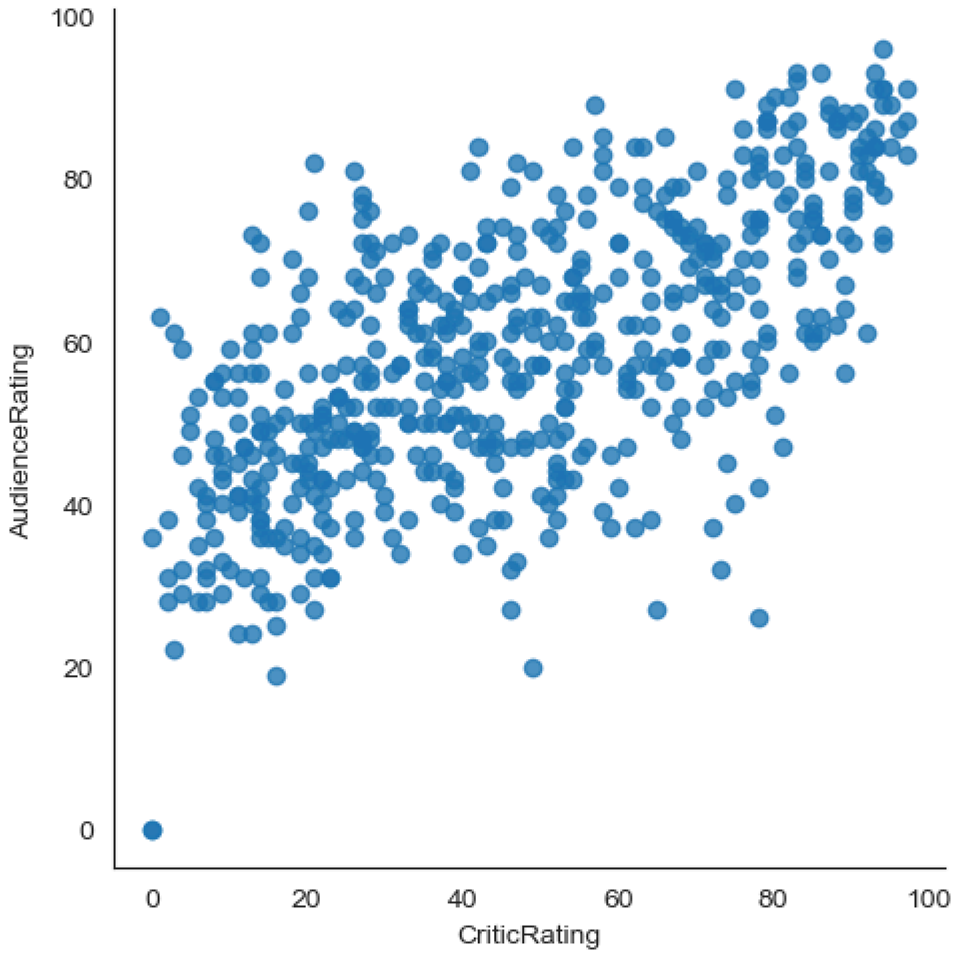
```
Index([2007, 2008, 2009, 2010, 2011], dtype='int64')
```

In [187…

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

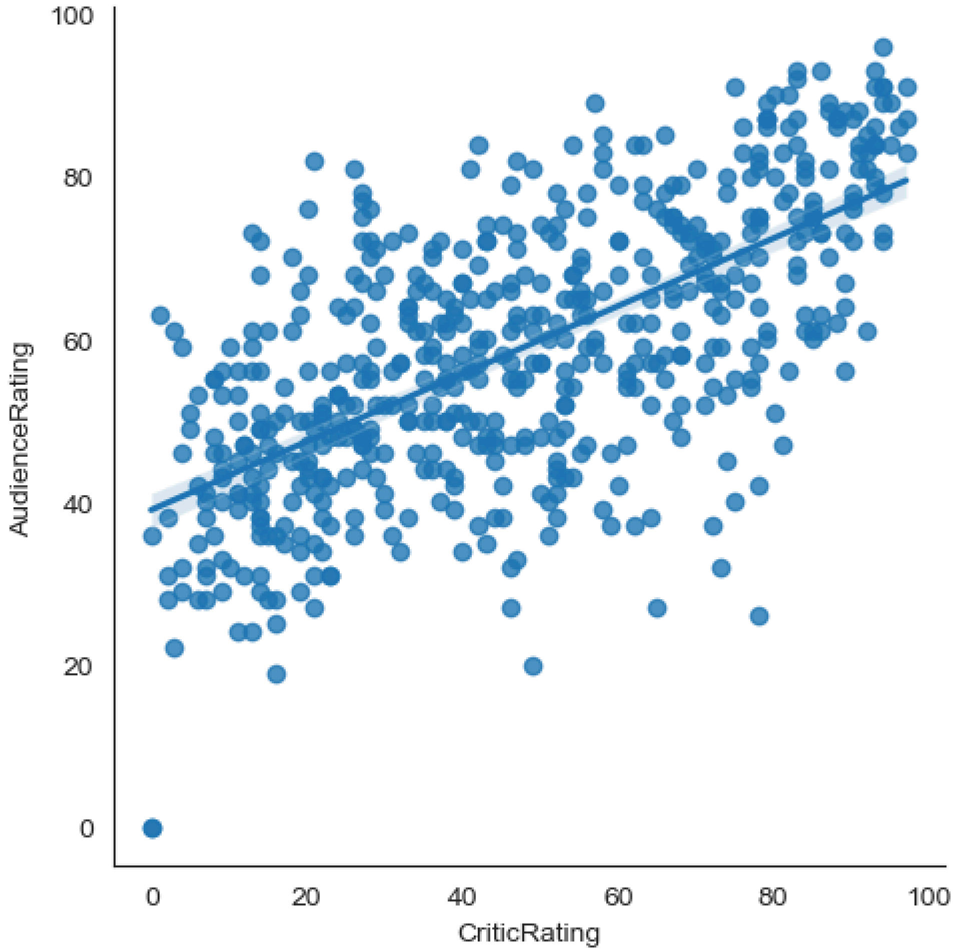
In [188…

```
plt.show()
```



In [189…

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

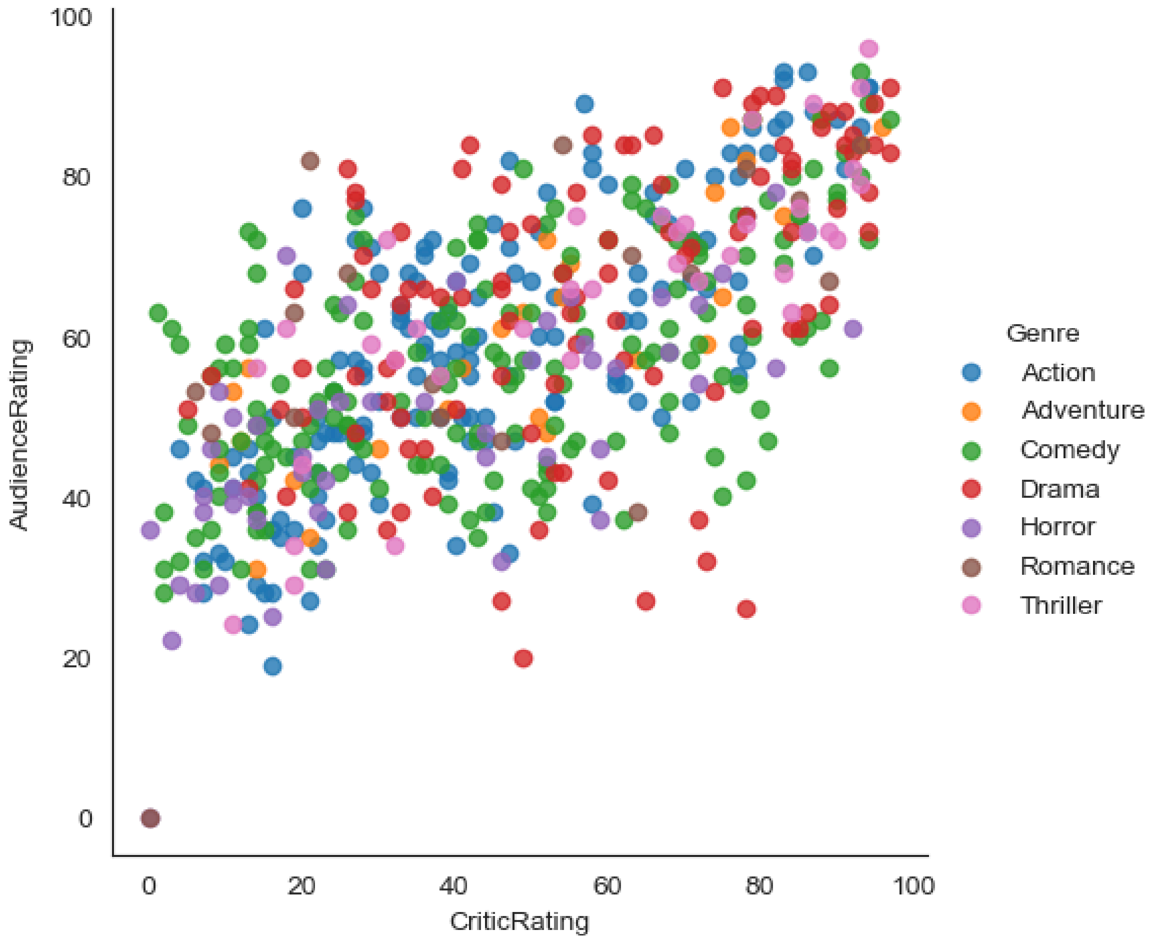


In [190…

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

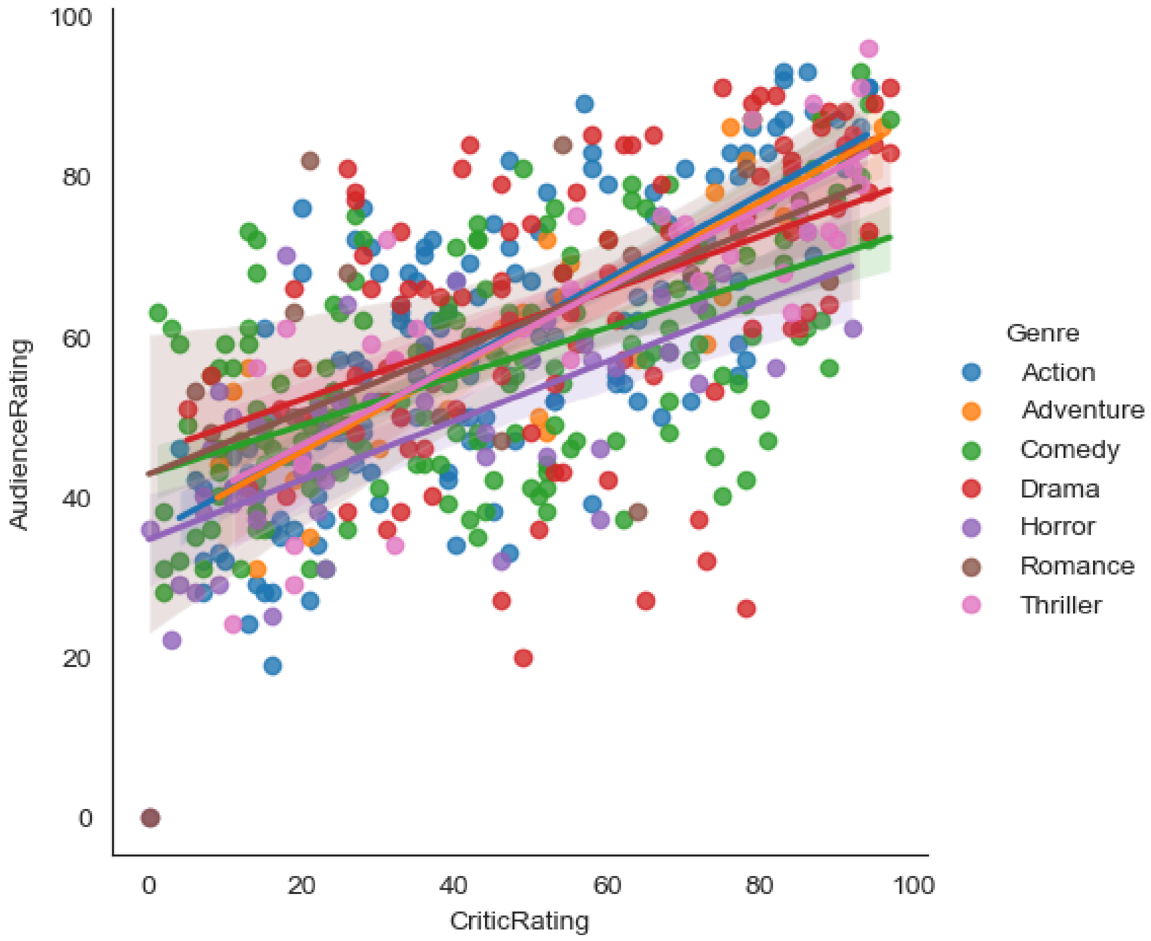
In [191…

```
plt.show()
```



In [192…

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



In [197…

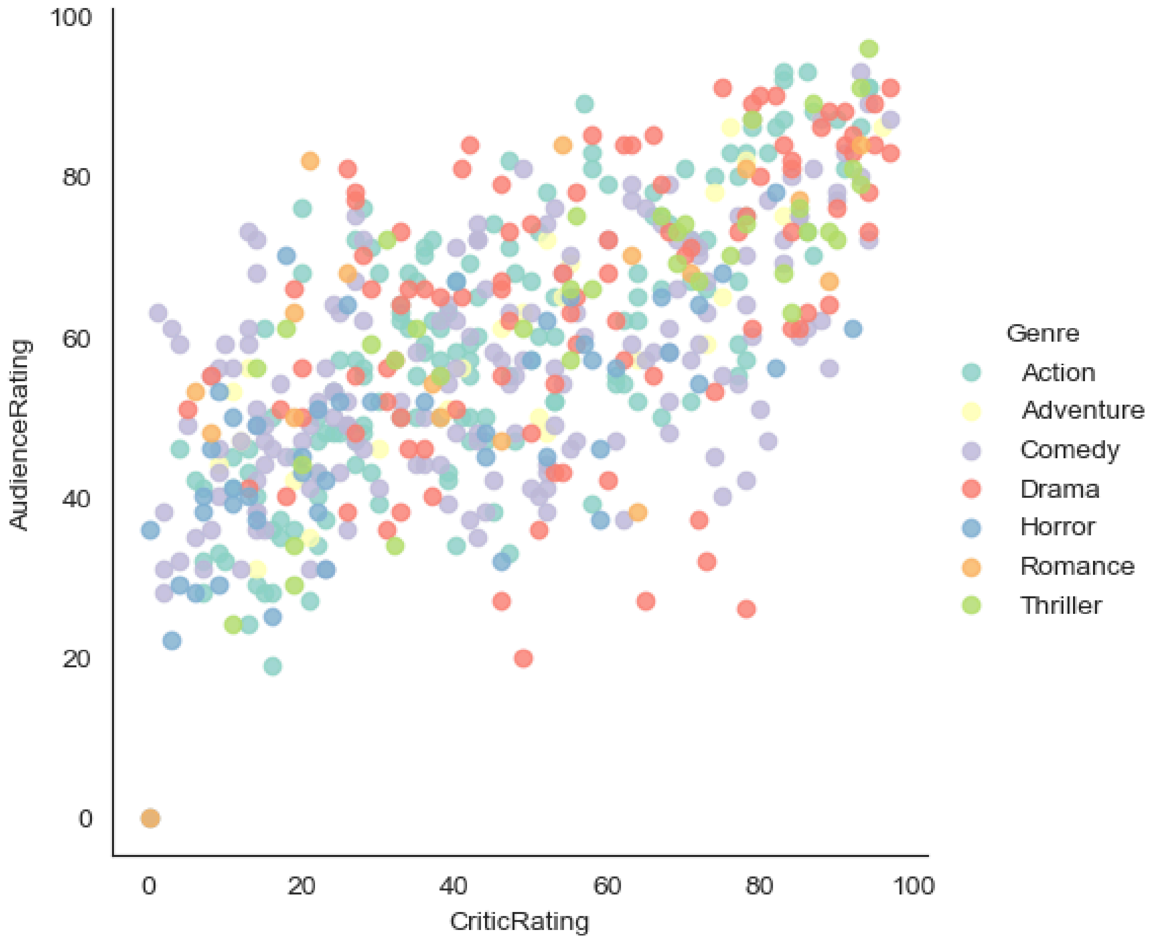
```
vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',\
fit_reg=False,hue='Genre',aspect=1,size=10)
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[197], line 1
----> 1 vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',\
      2 fit_reg=False,hue='Genre',aspect=1,size=10)
      3 plt.show()

TypeError: lmplot() got an unexpected keyword argument 'size'
```

In [199…

```
vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',\
fit_reg=False,hue='Genre',aspect=1,palette='Set3')
plt.show()
```



In [200…

```
sns.kdeplot(movies.CriticRating,movies.AudienceRating)
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[200], line 1
----> 1 sns.kdeplot(movies.CriticRating,movies.AudienceRating)
      2 plt.show()

TypeError: kdeplot() takes from 0 to 1 positional arguments but 2 were given
```

In [201…

```
k1=sns.kdeplot(movies.CriticRating,movies.AudienceRating)
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[201], line 1
----> 1 k1=sns.kdeplot(movies.CriticRating,movies.AudienceRating)
      2 plt.show()

TypeError: kdeplot() takes from 0 to 1 positional arguments but 2 were given
```

In [202…

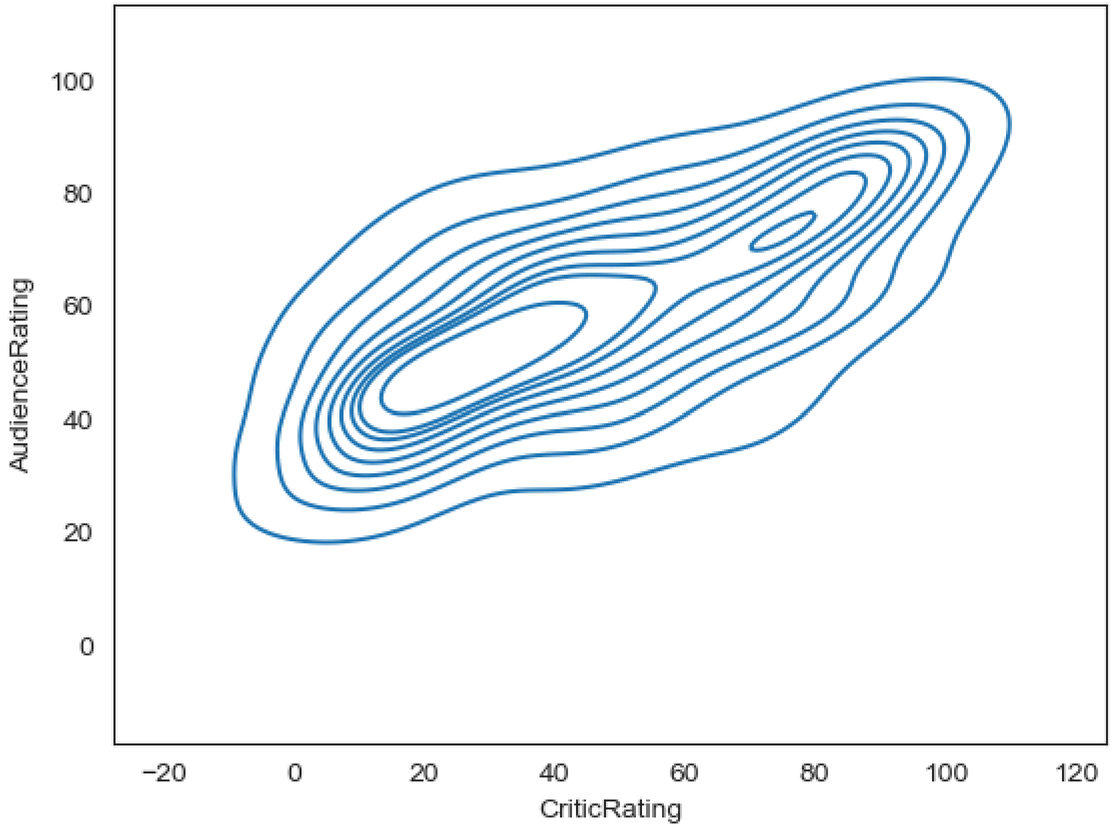
```
sns.kdeplot(data=movies,movies.CriticRating,movies.AudienceRating)
```

```
  Cell In[202], line 1
    sns.kdeplot(data=movies,movies.CriticRating,movies.AudienceRating)
                                                                     ^
SyntaxError: positional argument follows keyword argument
```

In [203…

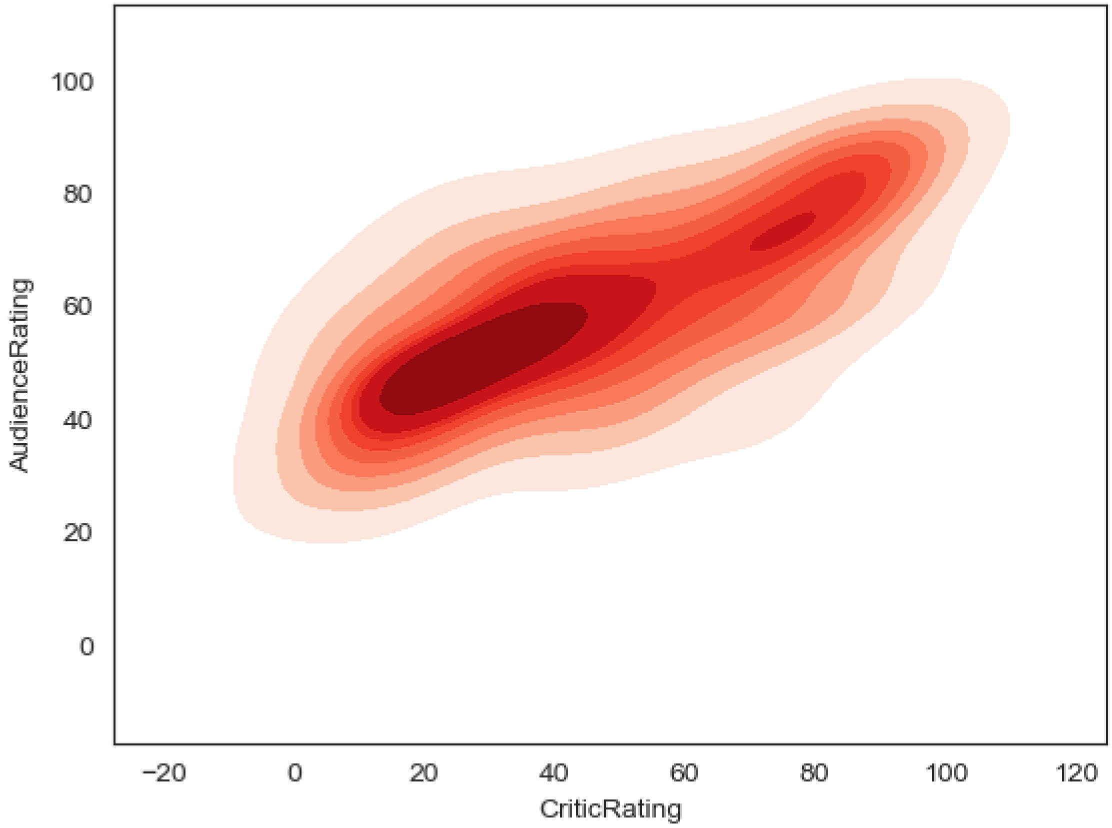
```
sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating')
```

```
plt.show()
```



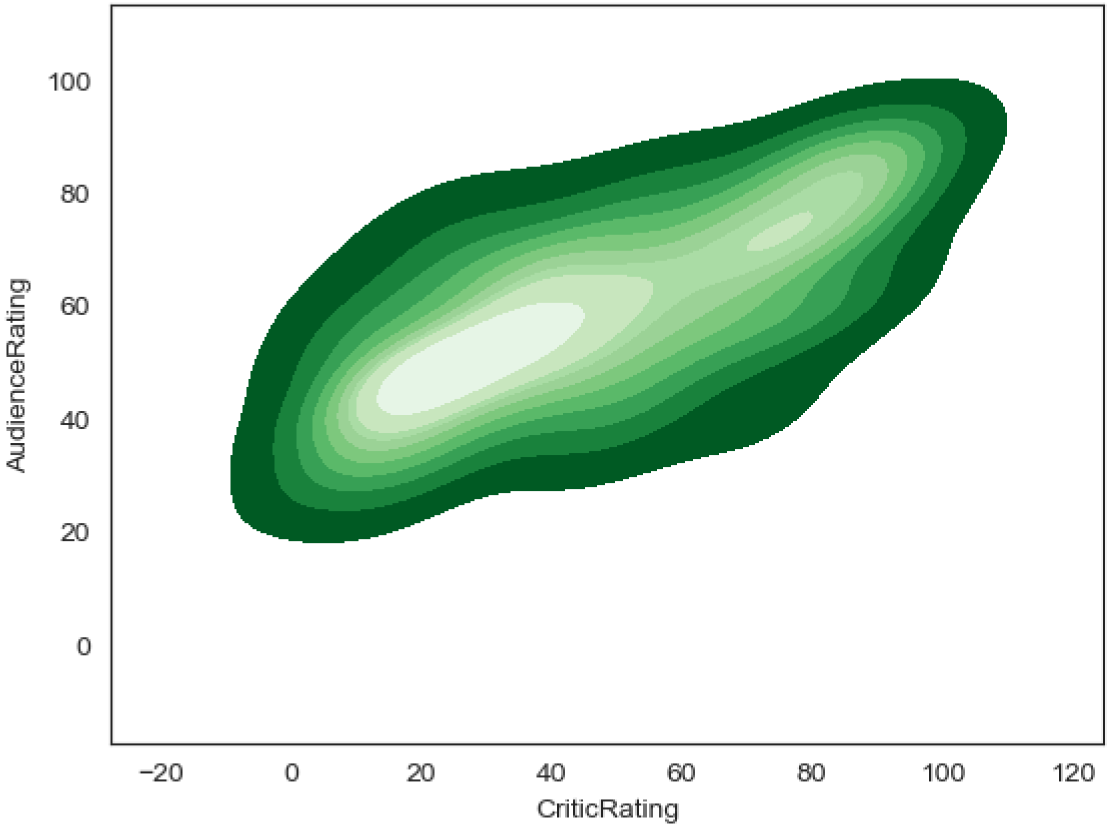
In [206…

```
sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating',shade=True,shade_low
plt.show()
```



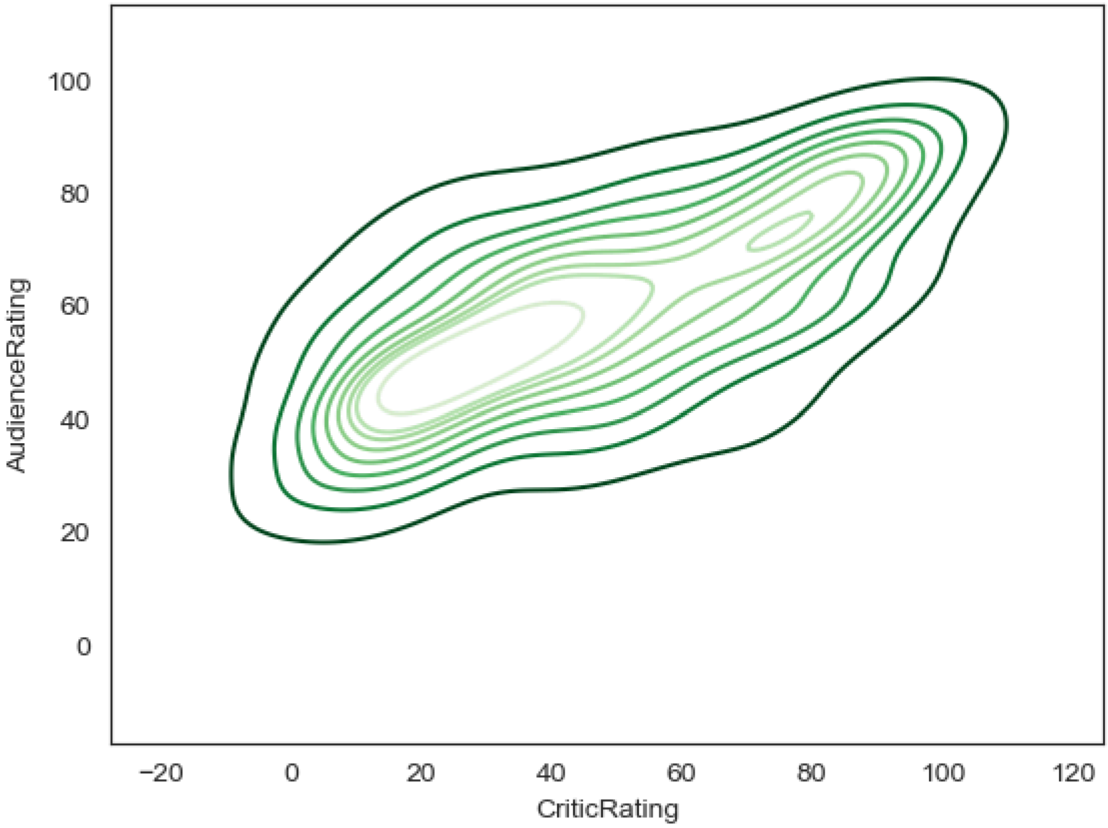
In [208…

```
k2=sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating',shade=True,shade_
plt.show()
```



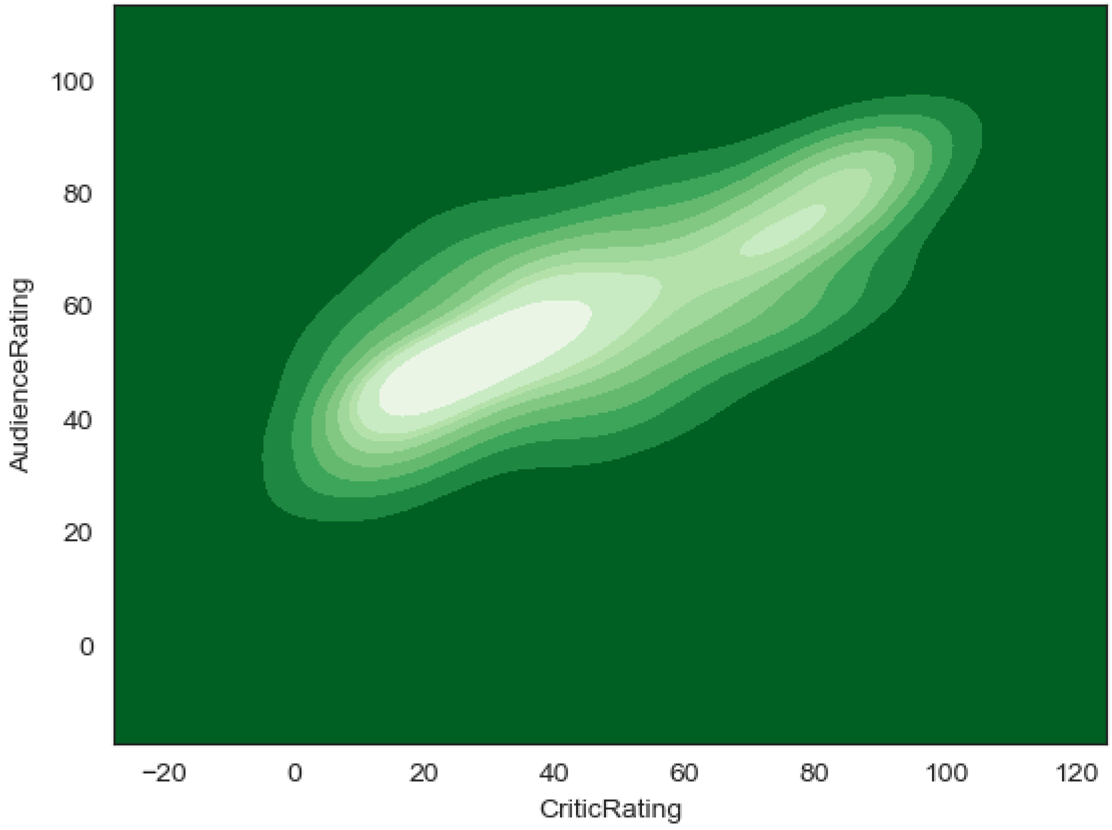
In [209…

```
k2=sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating',shade=False,shade
plt.show()
```



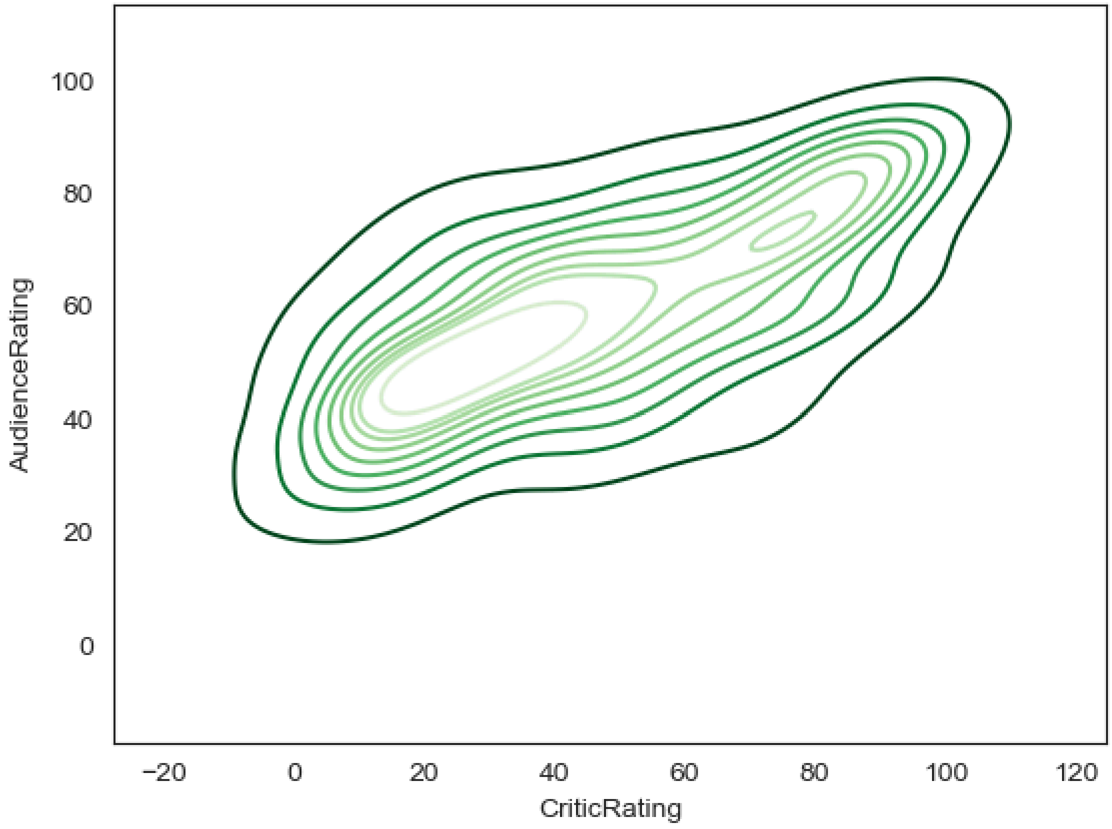
In [210…

```
k2=sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating',shade=True,shade_
plt.show()
```



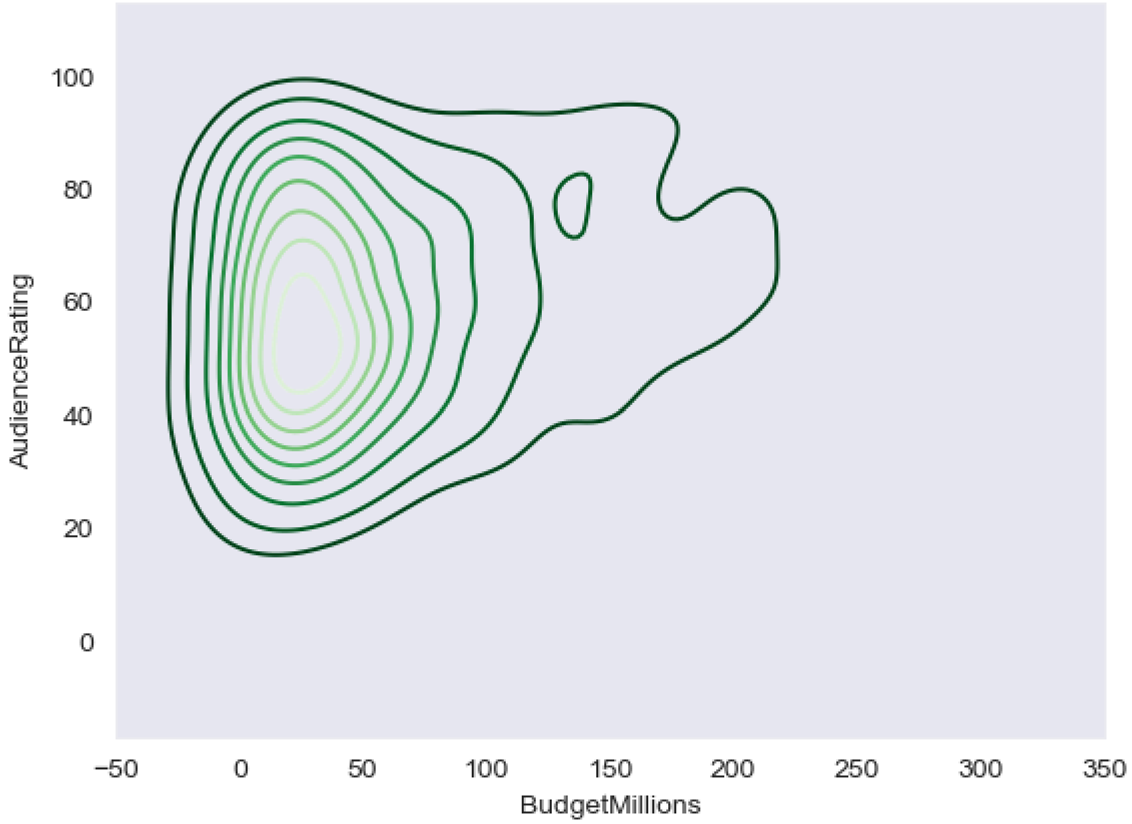
In [212…

```
k2=sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating',shade_lowest=Fals
plt.show()
```



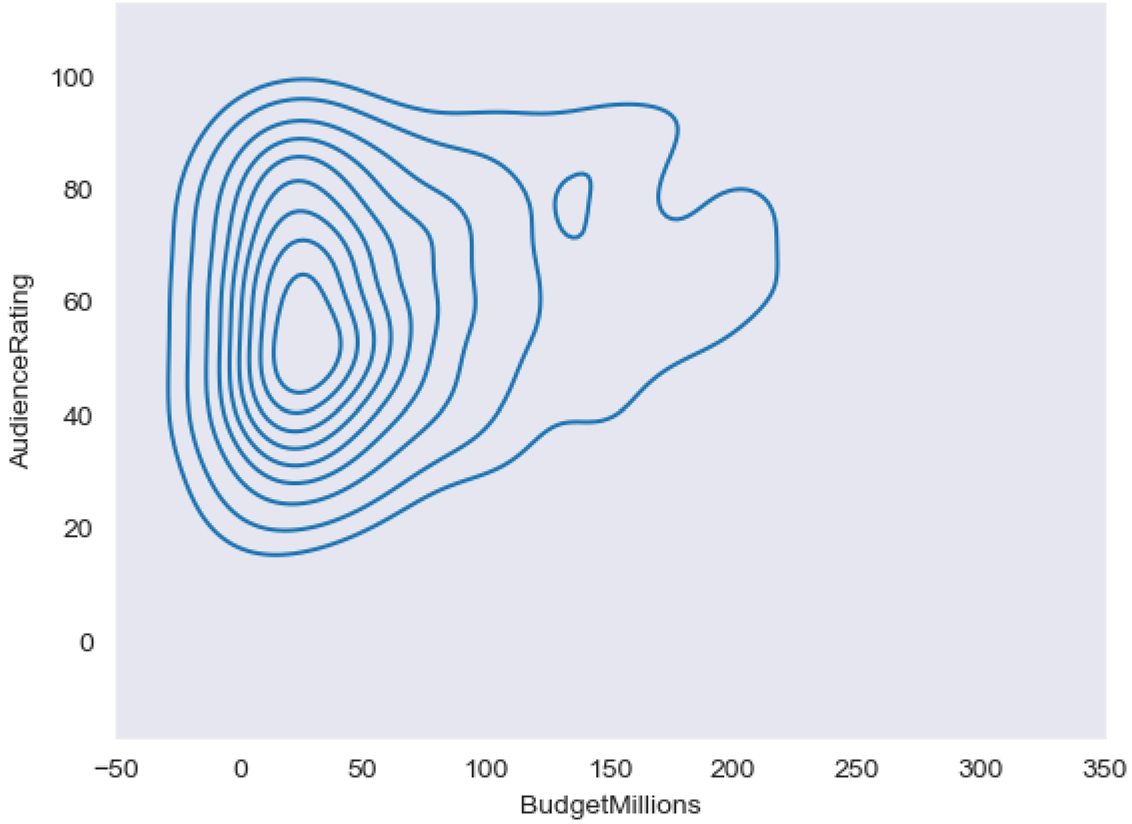
In [214…

```
sns.set_style('dark')
k1=sns.kdeplot(data=movies,x='BudgetMillions',y='AudienceRating',shade_lowest=Fa
plt.show()
```



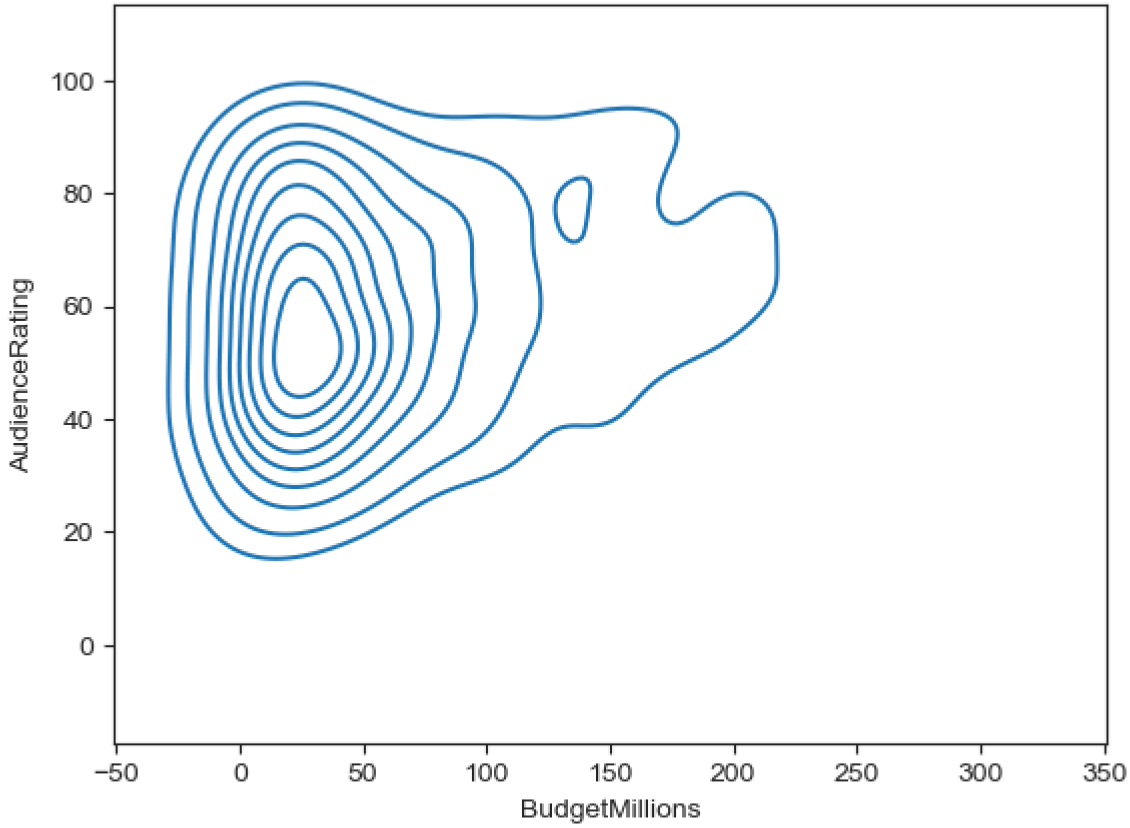
In [215…

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



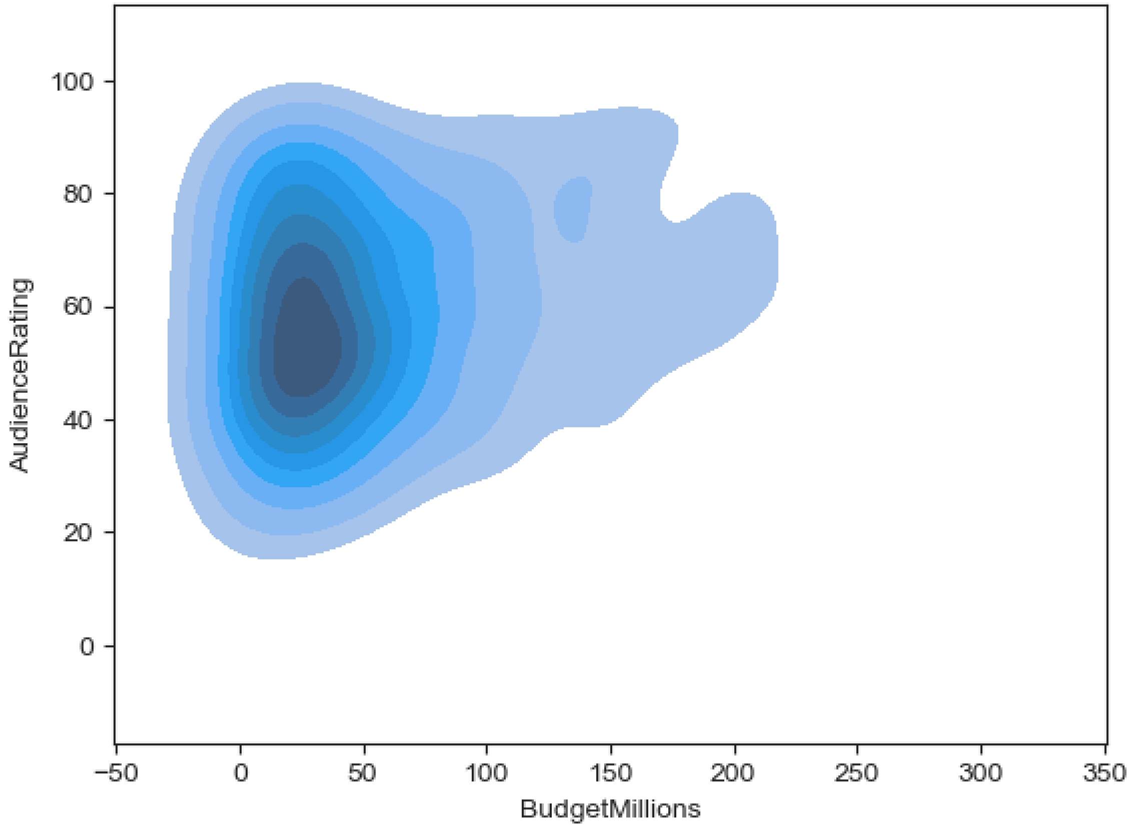
In [216…

```
sns.set_style('ticks')
sns.kdeplot(data=movies,x='BudgetMillions',y='AudienceRating')
plt.show()
```



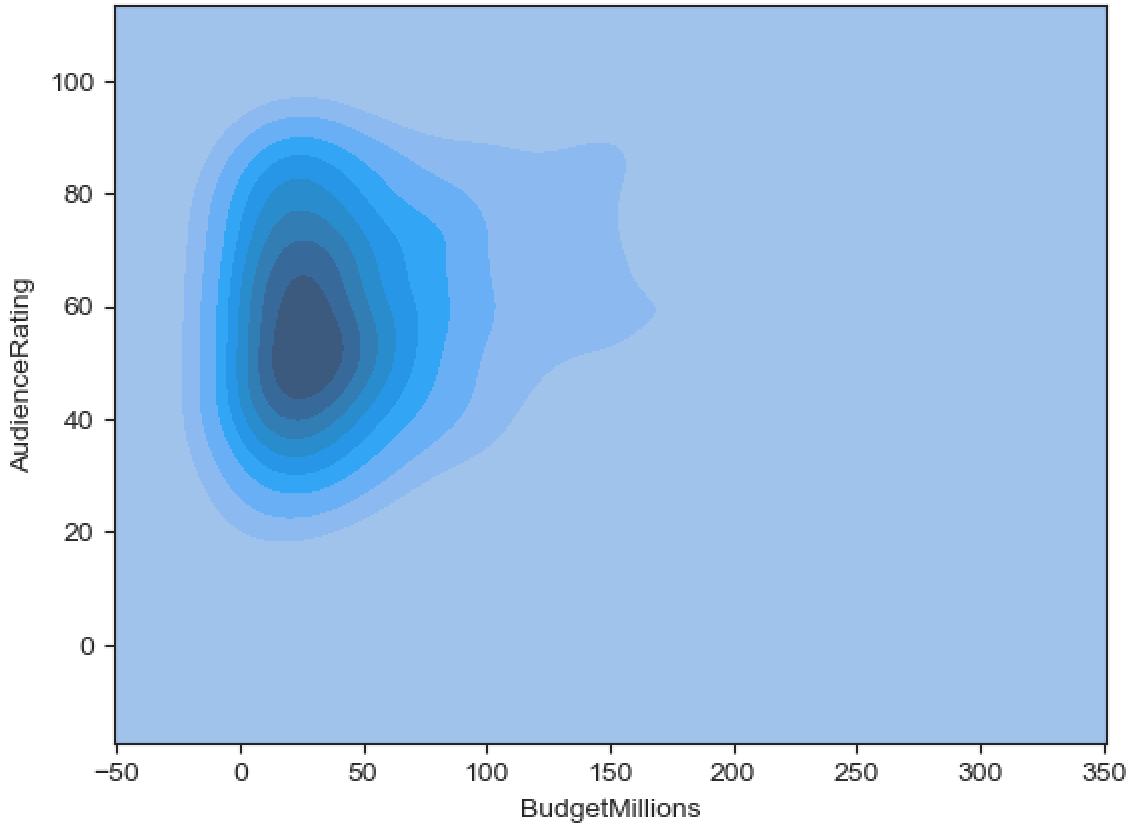
In [217…

```
sns.set_style('ticks')
sns.kdeplot(data=movies,x='BudgetMillions',y='AudienceRating',shade=True,shade_l
plt.show()
```



In [218…

```
sns.set_style('ticks')
sns.kdeplot(data=movies,x='BudgetMillions',y='AudienceRating',shade=True,shade_l
plt.show()
```



In [219…

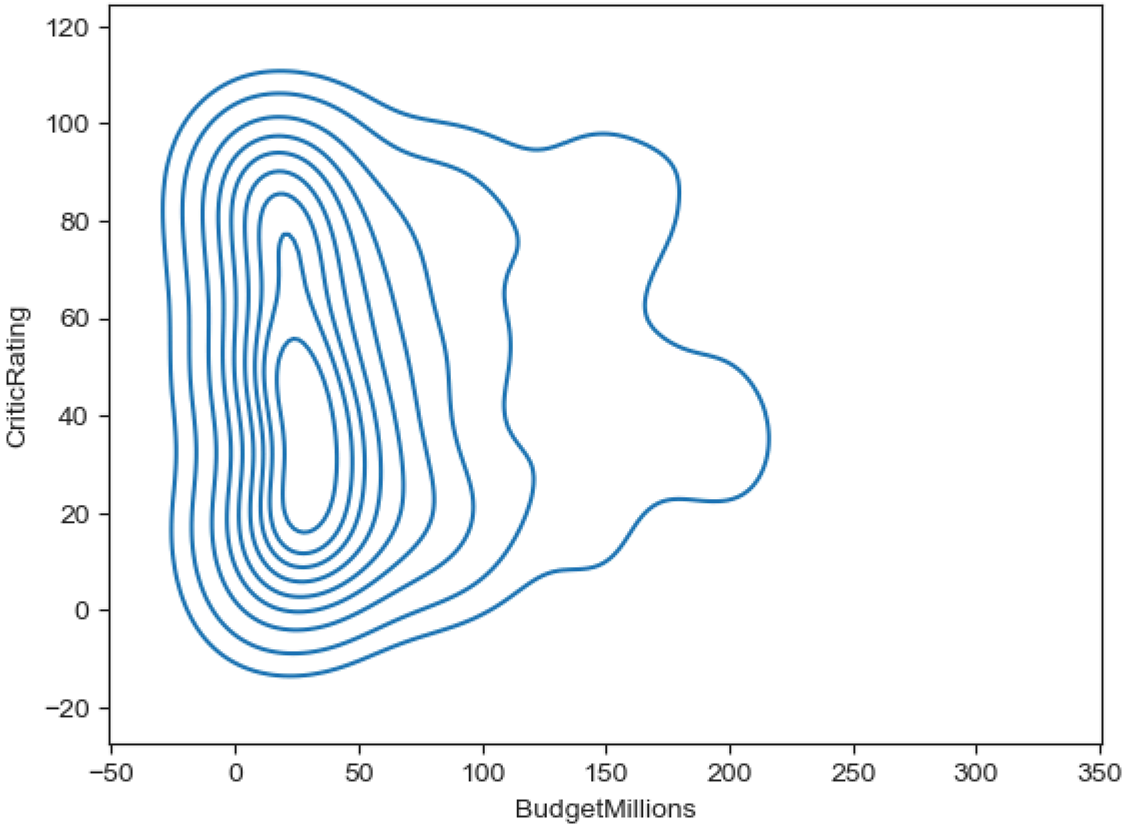
```
k2=sns.kdeplot(movies.BudgetMillions,movies.CriticRating)
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[219], line 1
----> 1 k2=sns.kdeplot(movies.BudgetMillions,movies.CriticRating)
      2 plt.show()

TypeError: kdeplot() takes from 0 to 1 positional arguments but 2 were given
```

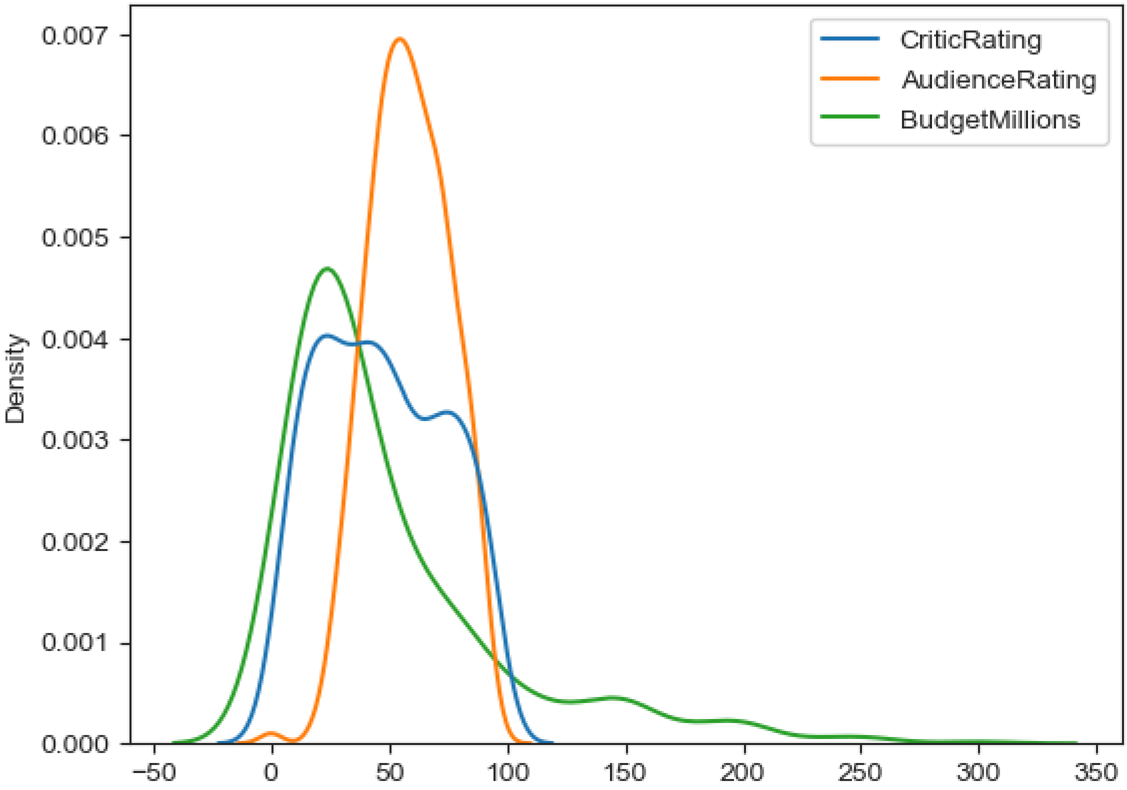
In [227…

```
k2=sns.kdeplot(data=movies,x='BudgetMillions',y='CriticRating')
plt.show()
```



In [221…

```
sns.kdeplot(data=movies)
plt.show()
```



In [222…

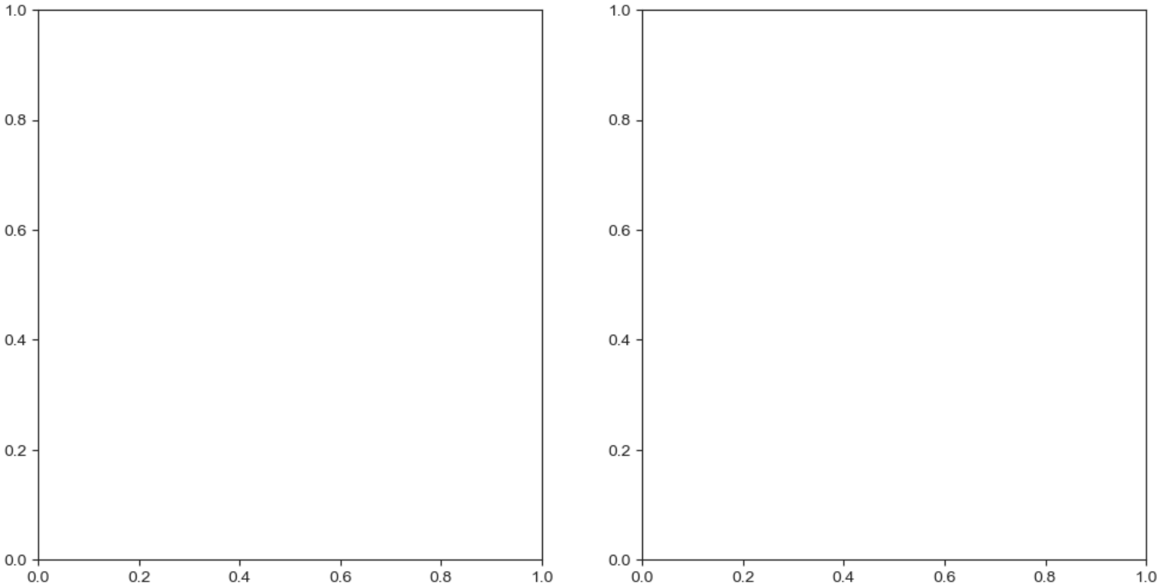
```
movies.describe()
```

Out[222…

	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

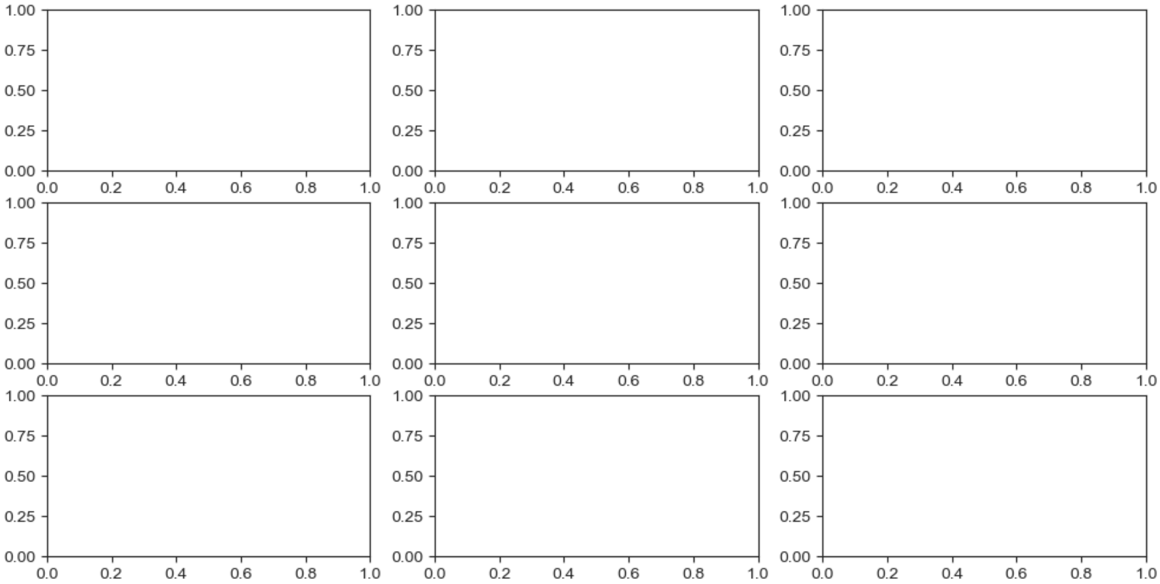
In [225…

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



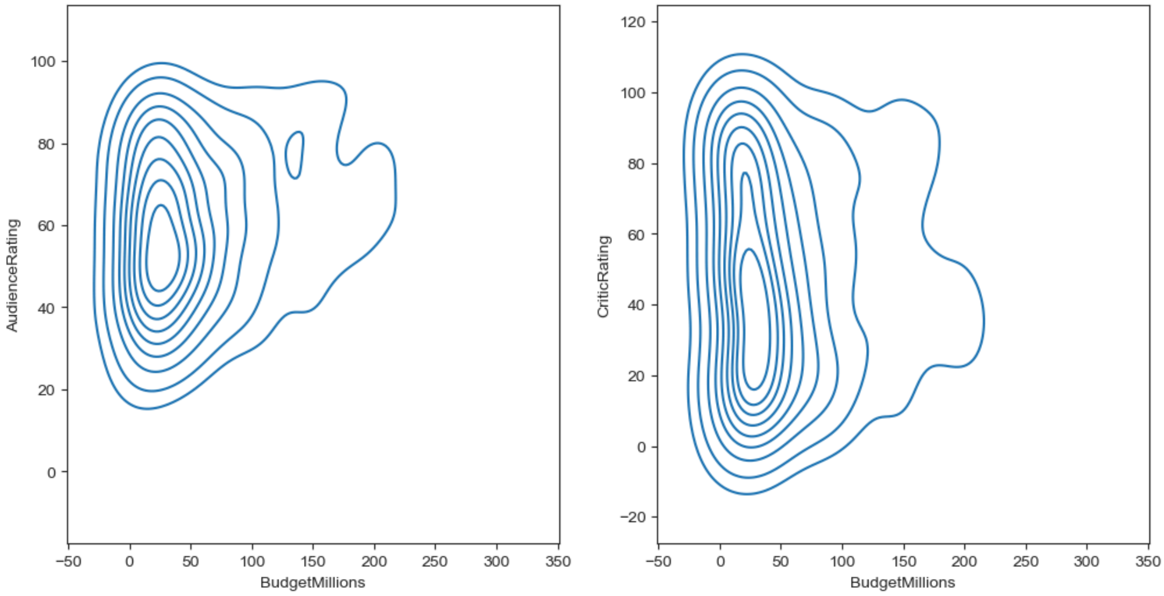
In [229…

```
f=plt.subplots(3,3 ,figsize=(12,6))
plt.show()
```



In [230…

```
f,axes=plt.subplots(1,2,figsize=(12,6))
k1=sns.kdeplot(data=movies,x='BudgetMillions',y='AudienceRating',ax=axes[0])
k2=sns.kdeplot(data=movies,x='BudgetMillions',y='CriticRating',ax=axes[1])
plt.show()
```



In [231…

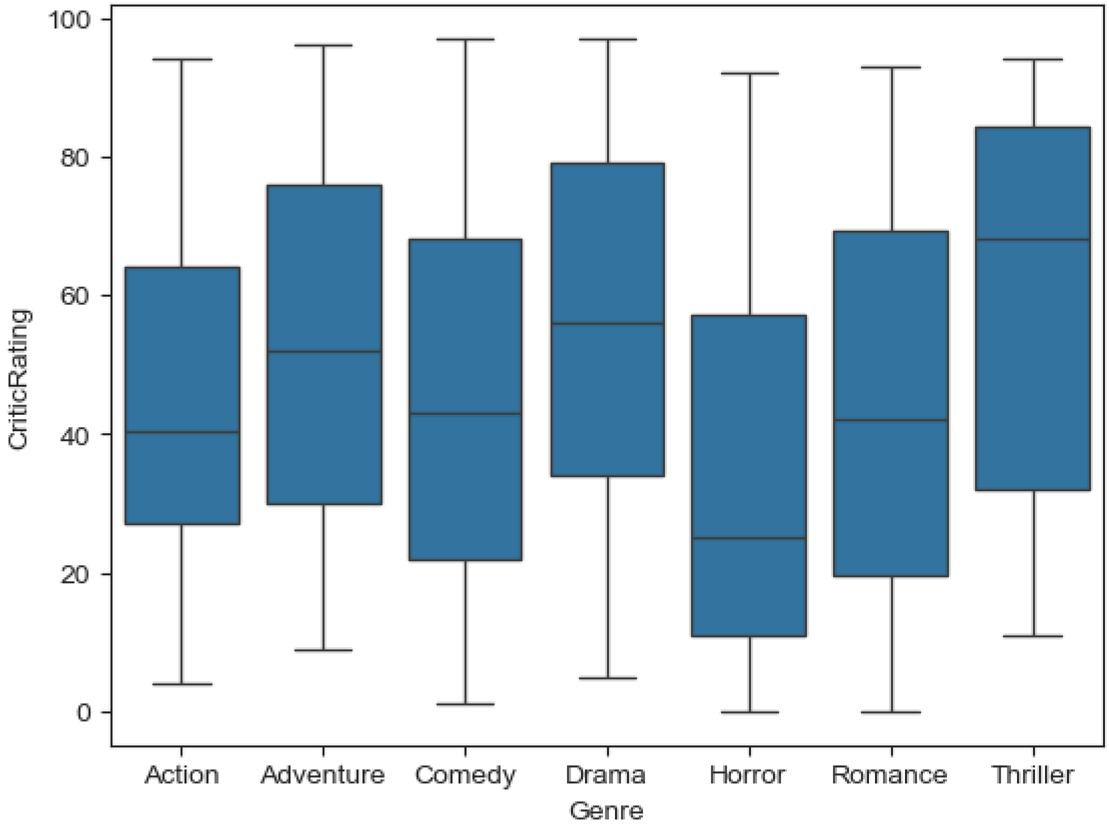
```
axes
```

Out[231…

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

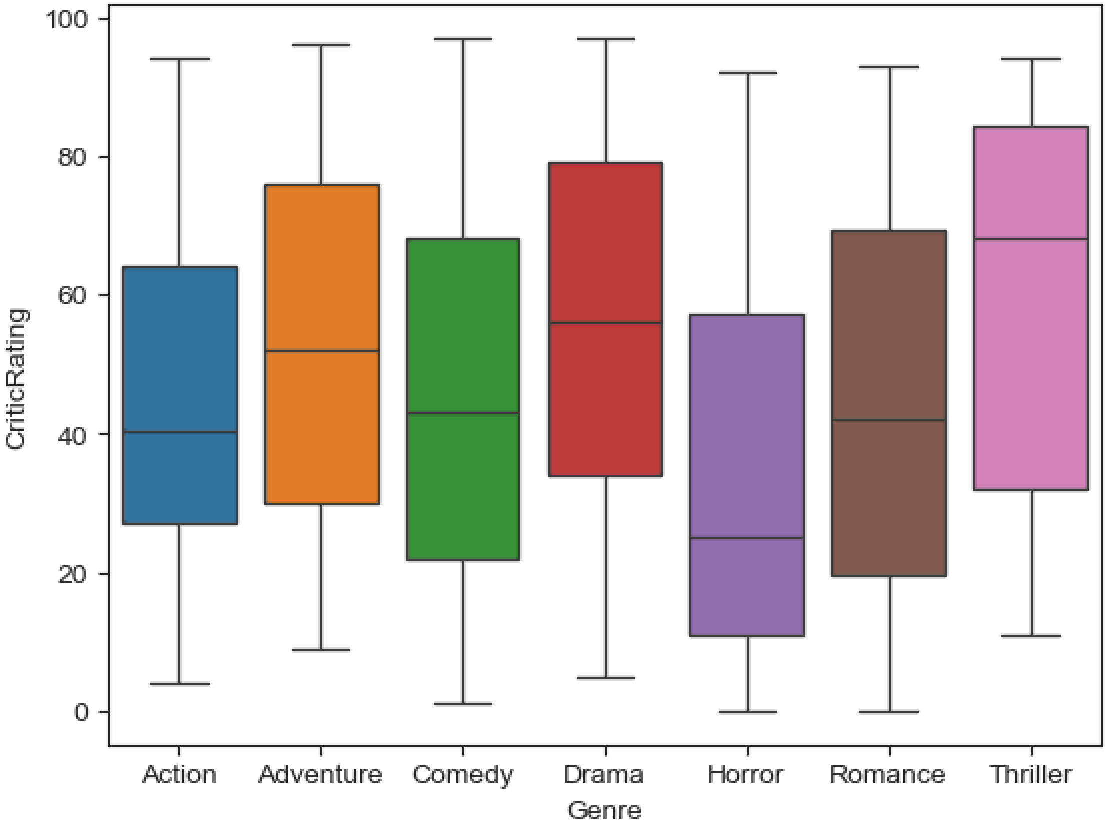
In [232…

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



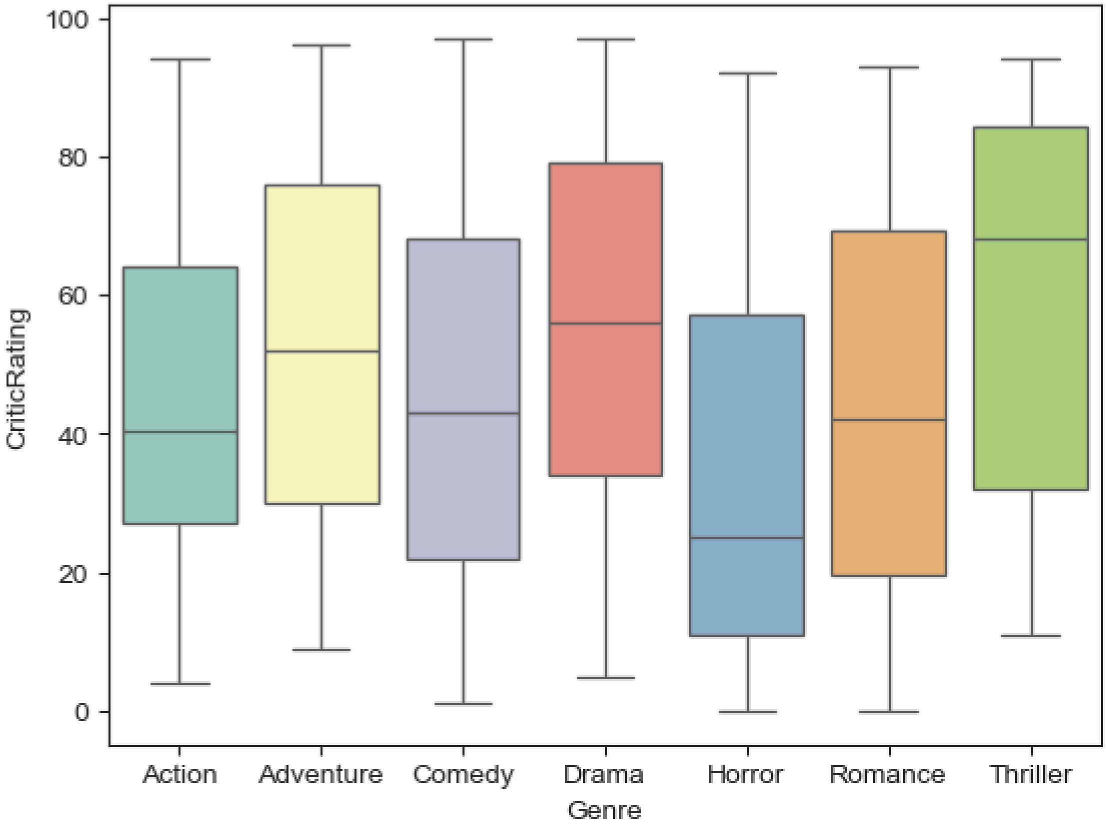
In [233…

```
w=sns.boxplot(data=movies,x='Genre',y='CriticRating',palette='tab10')
plt.show()
```



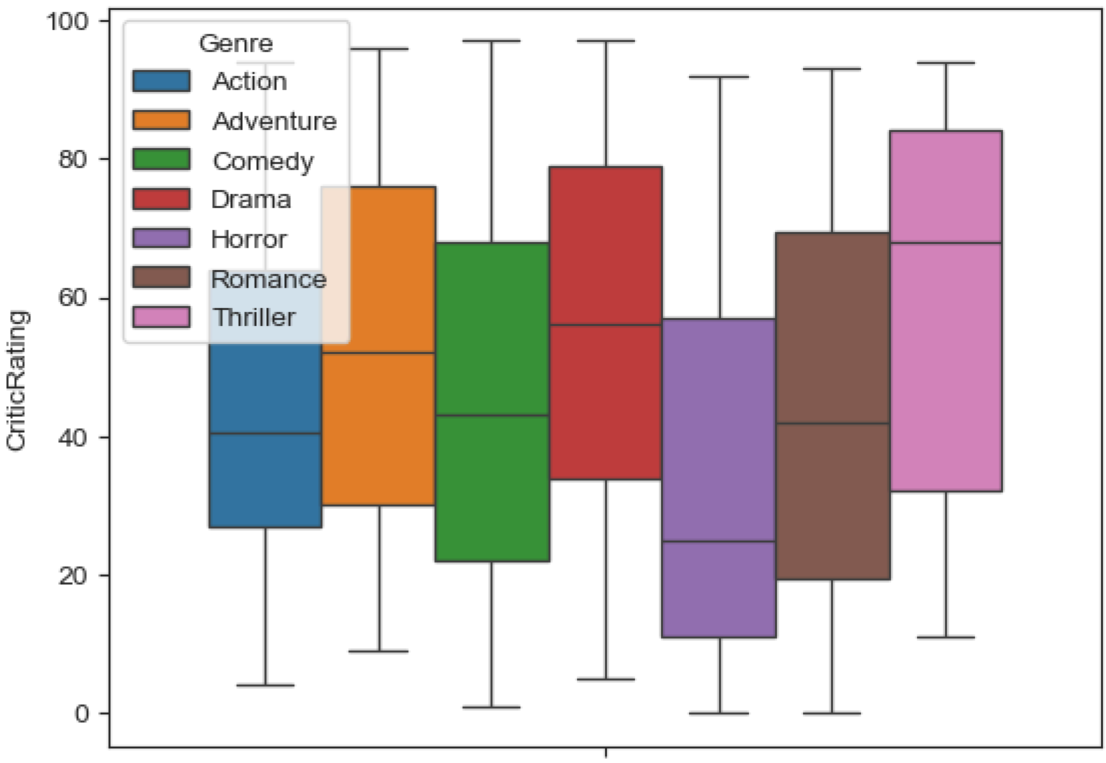
In [234…

```
w=sns.boxplot(data=movies,x='Genre',y='CriticRating',palette='Set3')
plt.show()
```



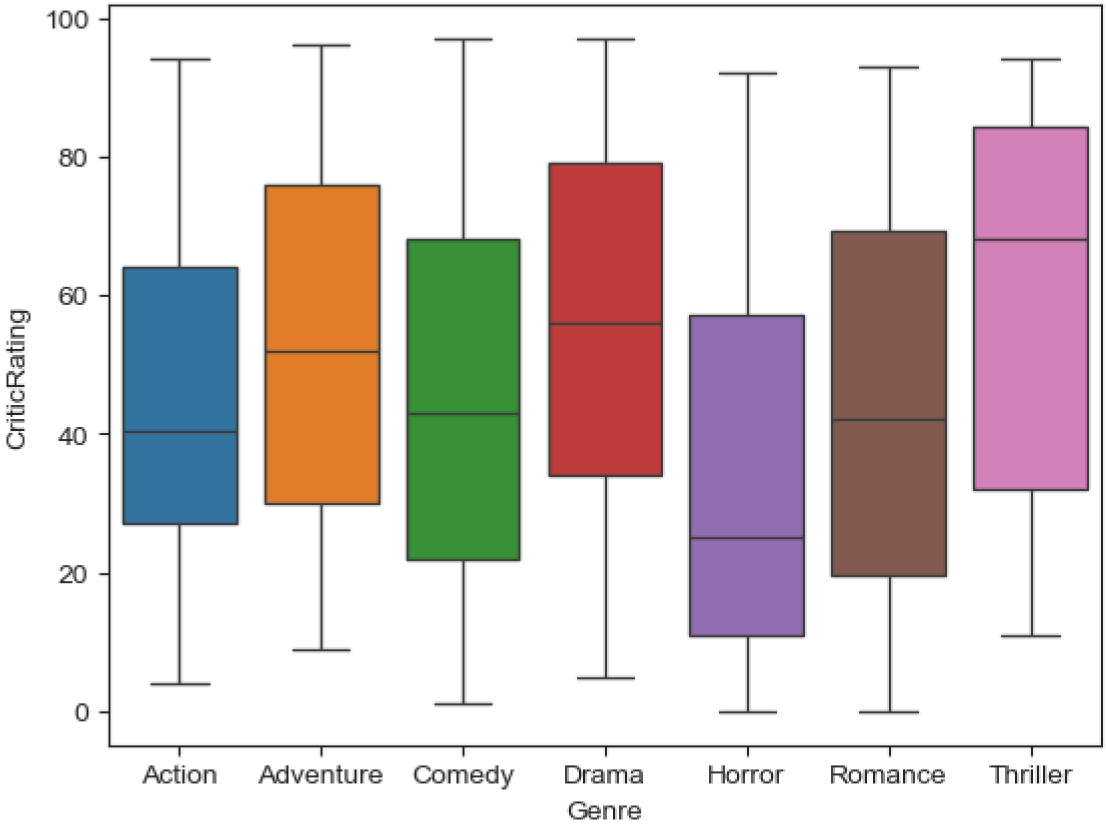
In [236…

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



In [240…

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



In [241…

```
movies.Genre
```

Out[241…

```
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 [242…

```
movies['Genre'].Action.value_counts()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_31676\1380449636.py in ?()
----> 1 movies['Genre'].Action.value_counts()

D:\New folder\Lib\site-packages\pandas\core\generic.py in ?(self, name)
   6295             and name not in self._accessors
   6296             and self._info_axis._can_hold_identifiers_and_holds_name(nam
e)
   6297         ):
   6298             return self[name]
-> 6299         return object.__getattribute__(self, name)

AttributeError: 'Series' object has no attribute 'Action'
```

In [243…

```
movies.Genre.Action.value_counts()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_31676\3621683978.py in ?()
----> 1 movies.Genre.Action.value_counts()

D:\New folder\Lib\site-packages\pandas\core\generic.py in ?(self, name)
   6295             and name not in self._accessors
   6296             and self._info_axis._can_hold_identifiers_and_holds_name(nam
e)
   6297         ):
   6298             return self[name]
-> 6299         return object.__getattribute__(self, name)

AttributeError: 'Series' object has no attribute 'Action'
```

In [244…

```
movies.Action.info()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_31676\2748353175.py in ?()
----> 1 movies.Action.info()

D:\New folder\Lib\site-packages\pandas\core\generic.py in ?(self, name)
   6295             and name not in self._accessors
   6296             and self._info_axis._can_hold_identifiers_and_holds_name(nam
e)
   6297         ):
   6298             return self[name]
-> 6299         return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'Action'
```

In [245…

```
Action.info()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[245], line 1
----> 1 Action.info()

NameError: name 'Action' is not defined
```

In [246…

```
movies.Genre.Action.describe()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_31676\4103437533.py in ?()
----> 1 movies.Genre.Action.describe()

D:\New folder\Lib\site-packages\pandas\core\generic.py in ?(self, name)
   6295             and name not in self._accessors
   6296             and self._info_axis._can_hold_identifiers_and_holds_name(nam
e)
   6297         ):
   6298             return self[name]
-> 6299         return object.__getattribute__(self, name)

AttributeError: 'Series' object has no attribute 'Action'
```

In [247…

```
movies.Genre.describe()
```

Out[247…

```
count        559
unique         7
top       Comedy
freq         172
Name: Genre, dtype: object
```

In [248…

```
movies['Genre'].describe()
```

Out[248…

```
count        559
unique         7
top       Comedy
freq         172
Name: Genre, dtype: object
```

In [249…

```
movies.describe()
```

Out[249…

	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 [252…

```
movies['Genre']['Action'].describe()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[252], line 1
----> 1 movies['Genre']['Action'].describe()

File D:\New folder\Lib\site-packages\pandas\core\series.py:1121, in Series.__geti
tem__(self, key)
   1118     return self._values[key]
   1120 elif key_is_scalar:
-> 1121     return self._get_value(key)
   1123 # Convert generator to list before going through hashable part
   1124 # (We will iterate through the generator there to check for slices)
   1125 if is_iterator(key):

File D:\New folder\Lib\site-packages\pandas\core\series.py:1237, in Series._get_v
alue(self, label, takeable)
   1234     return self._values[label]
   1236 # Similar to Index.get_value, but we do not fall back to positional
-> 1237 loc = self.index.get_loc(label)
   1239 if is_integer(loc):
   1240     return self._values[loc]

File D:\New folder\Lib\site-packages\pandas\core\indexes\range.py:417, in RangeIn
dex.get_loc(self, key)
    415         raise KeyError(key) from err
    416 if isinstance(key, Hashable):
--> 417     raise KeyError(key)
    418 self._check_indexing_error(key)
    419 raise KeyError(key)

KeyError: 'Action'
```

In [253…

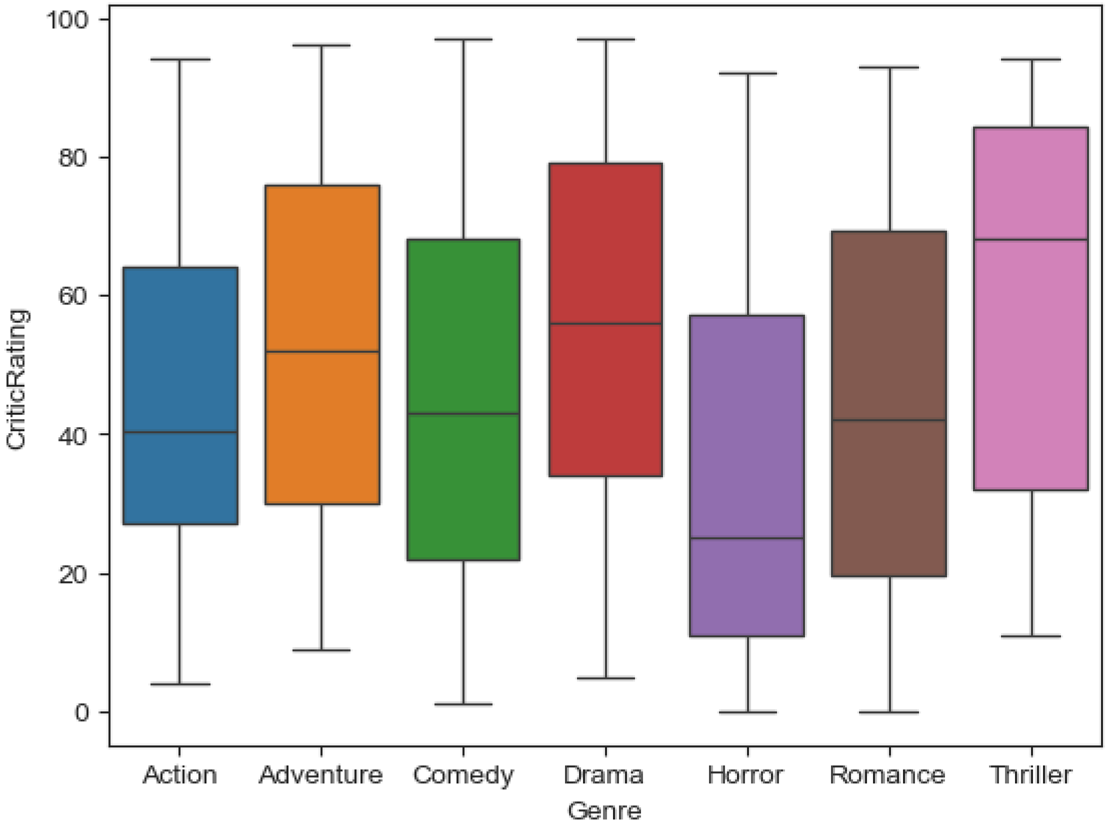
```
movies.Genre
```

Out[253…

```
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']
```

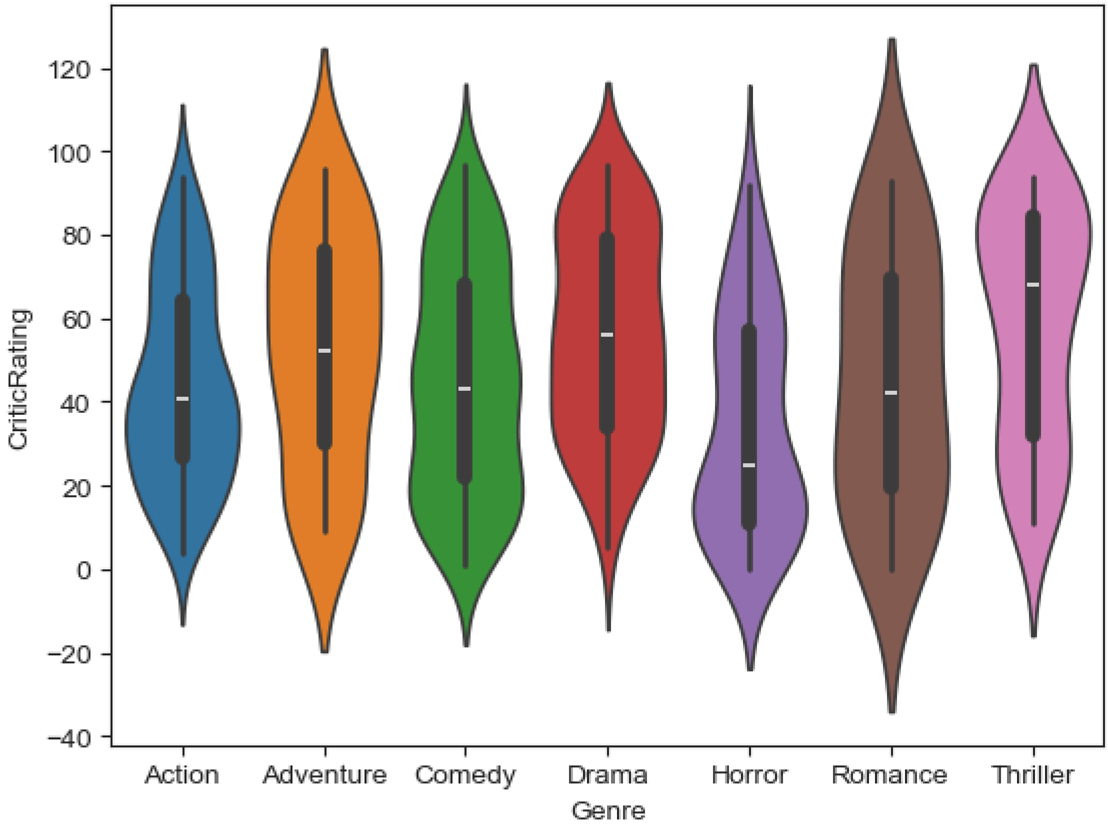
In [254…

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



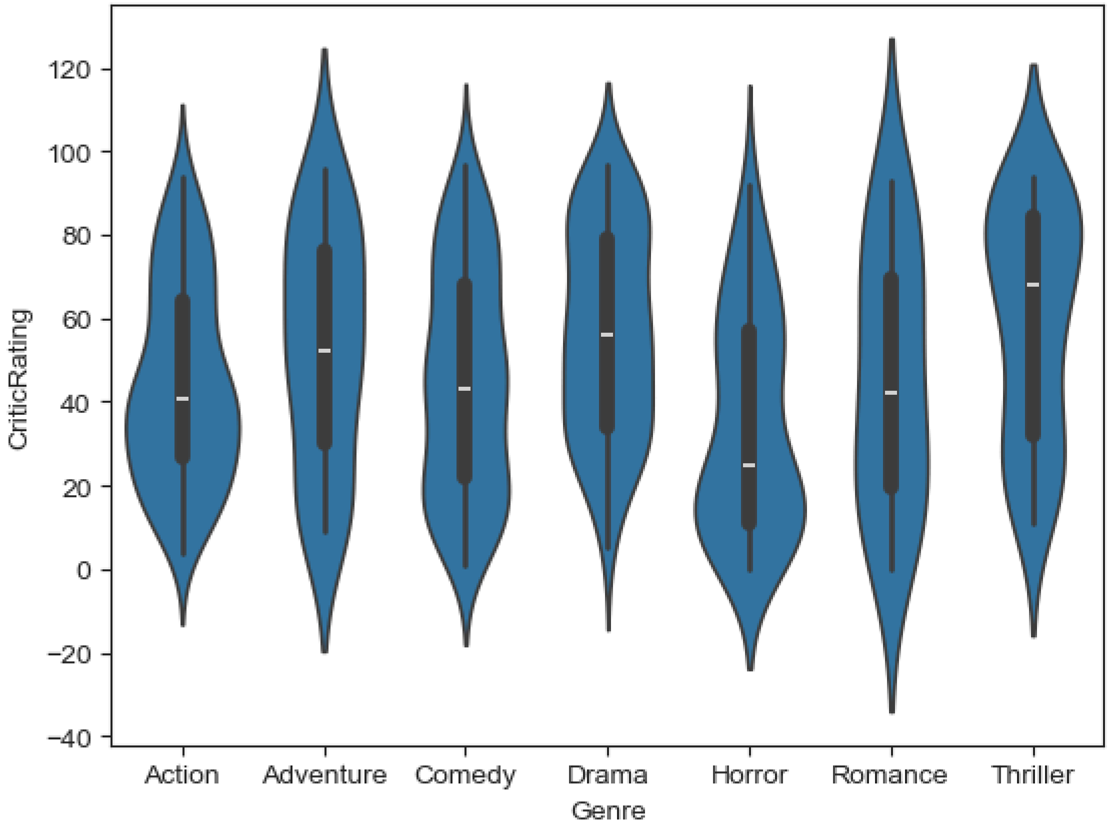
In [255…

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



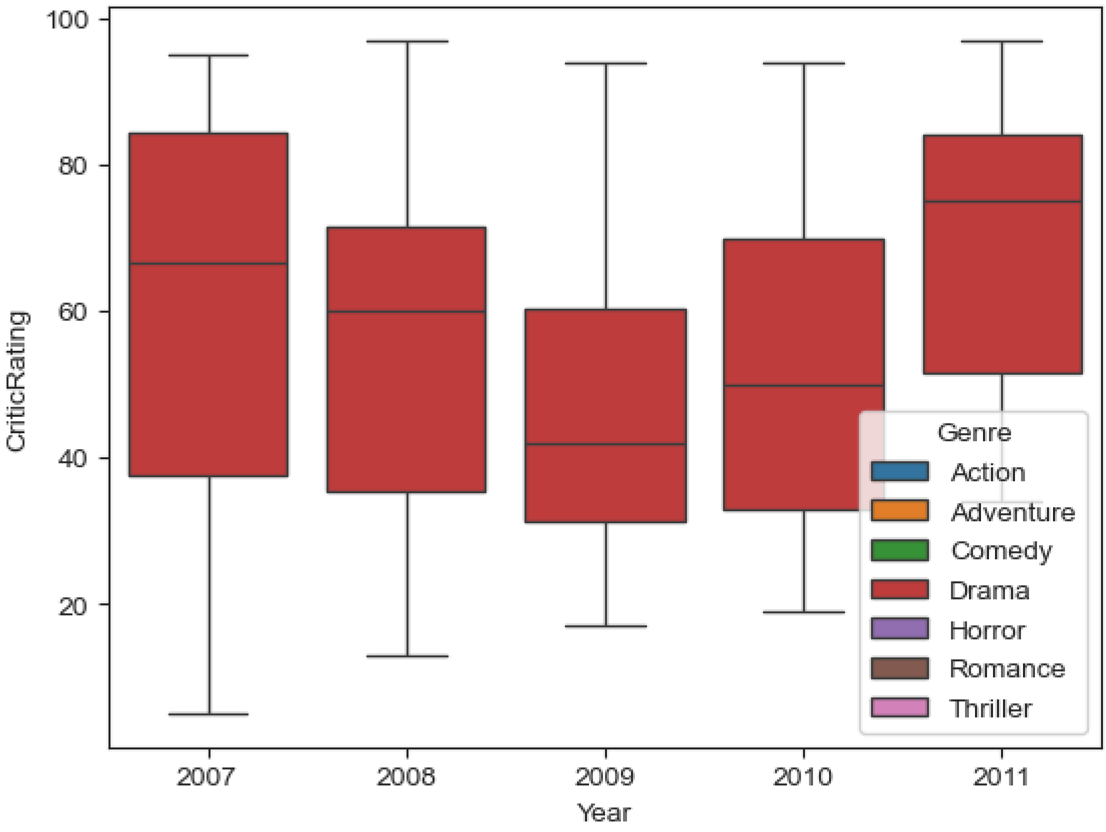
In [256…

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



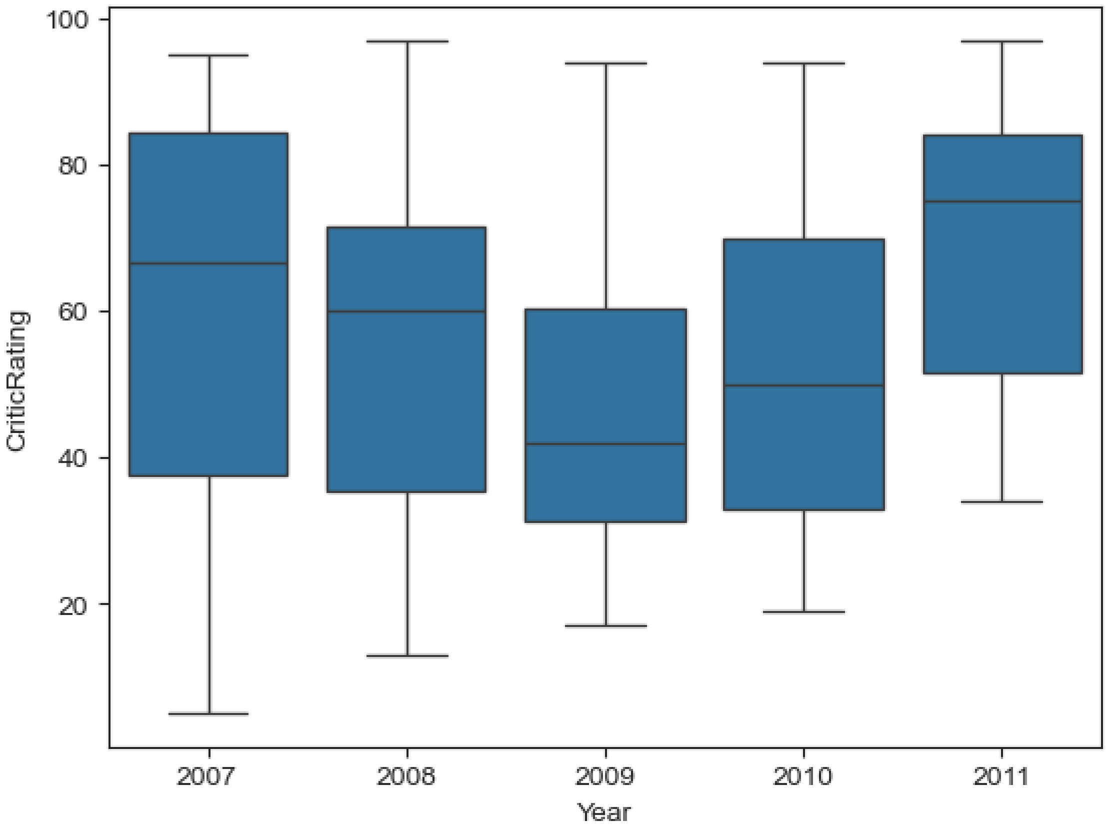
In [257…

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



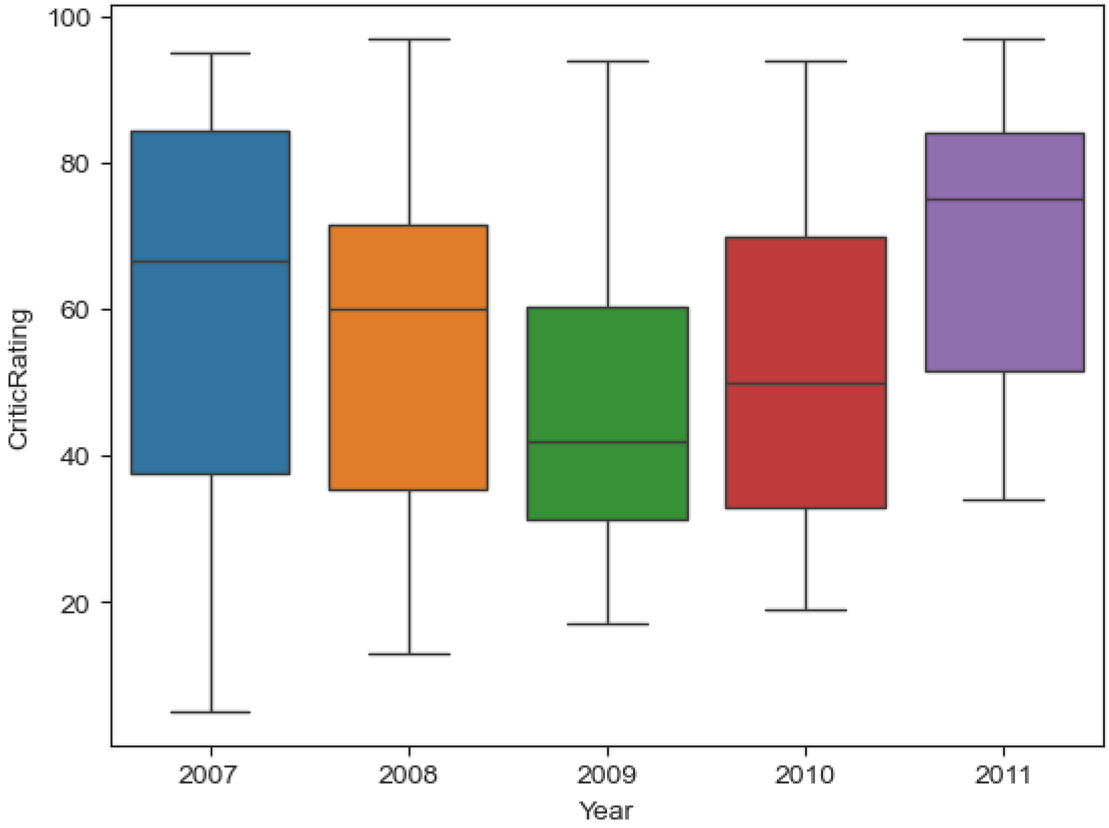
In [258…

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



In [261…

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



In [263…

```
a=[movies.Genre=='Drama'].describe()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[263], line 1
----> 1 a=[movies.Genre=='Drama'].describe()

AttributeError: 'list' object has no attribute 'describe'
```

In [264…

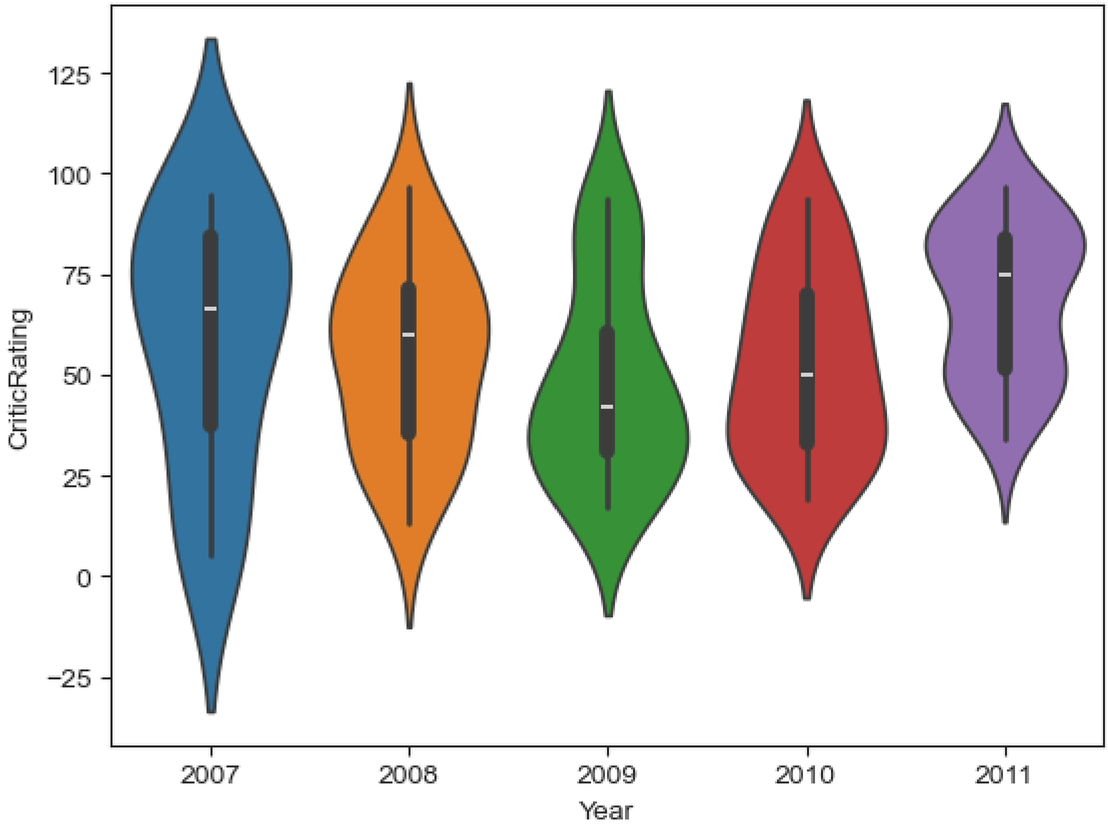
```
movies.Genre
```

Out[264…

```
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 [265…

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



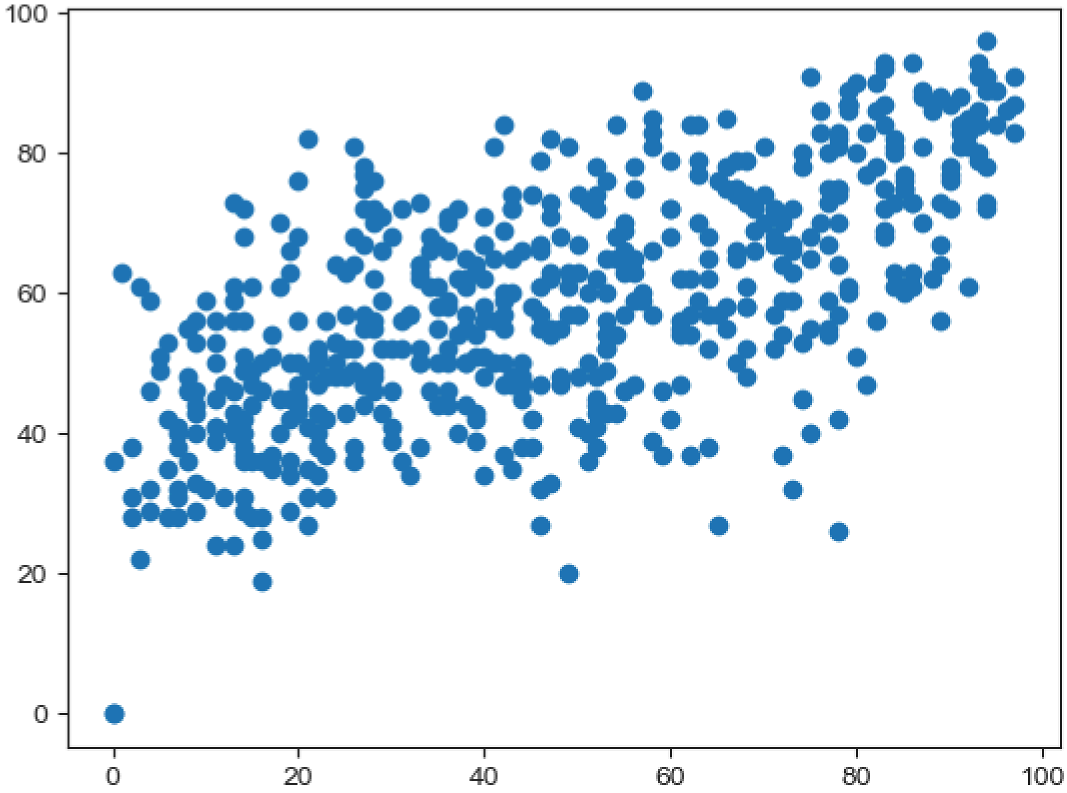
In [267…

```
g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
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
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
0.0
0.2
0.4
0.6
0.8
1.0

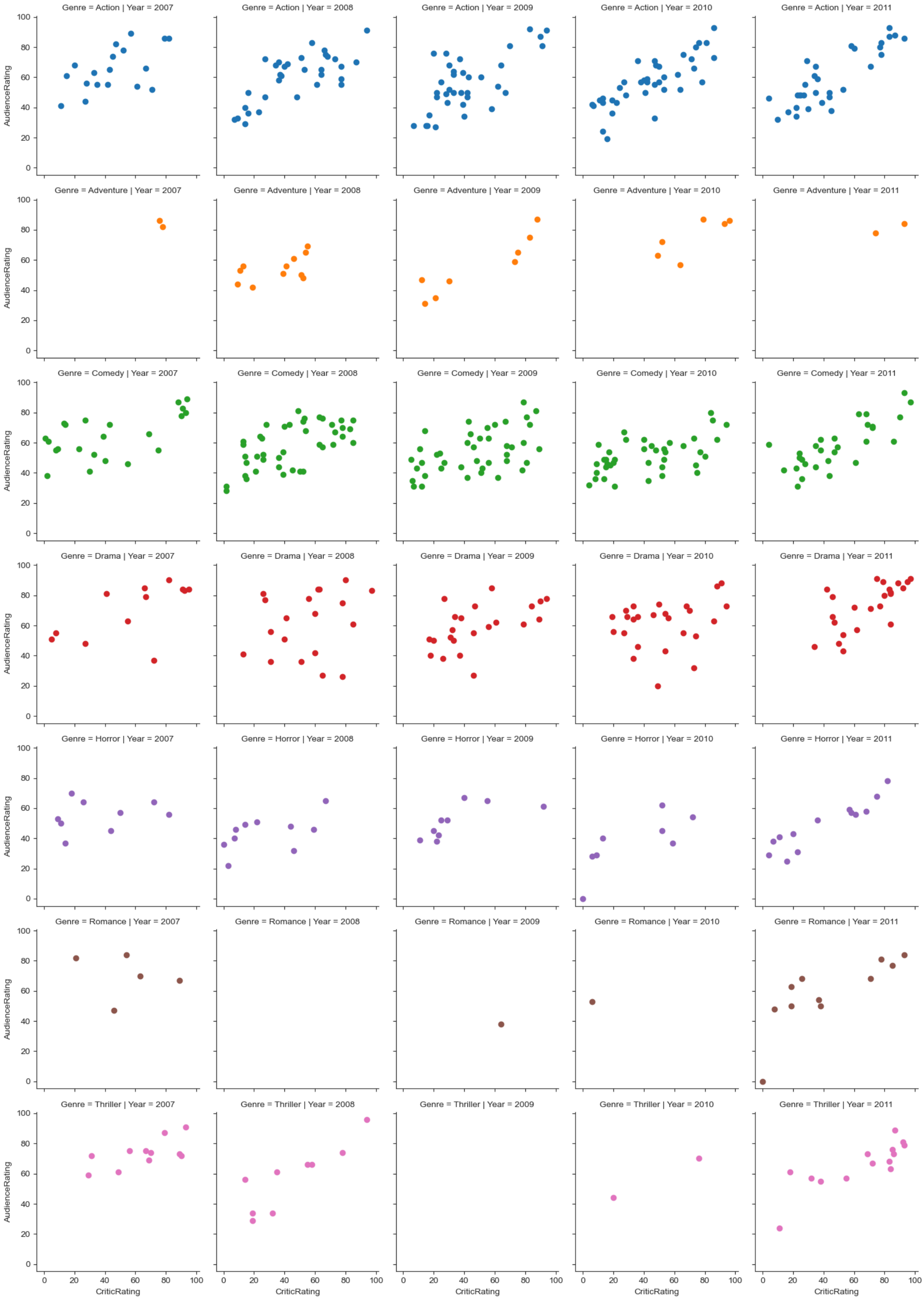
In [268…

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



In [269…

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



In [272…

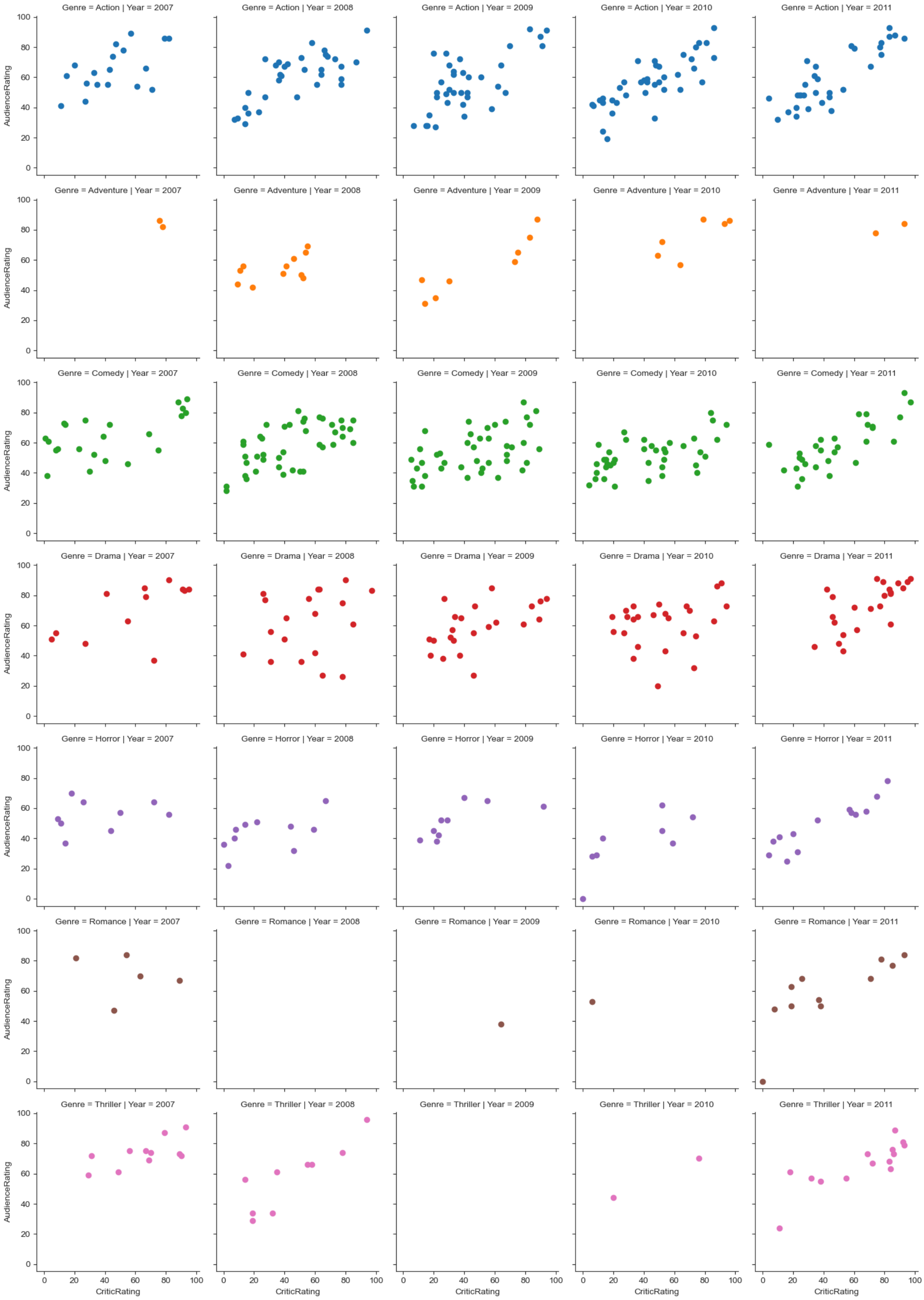
```
g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre',reg=True)
g=g.map(plt.scatter,'CriticRating','AudienceRating')
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[272], line 1
----> 1 g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre',reg=True)
      2 g=g.map(plt.scatter,'CriticRating','AudienceRating')
      3 plt.show()

TypeError: FacetGrid.__init__() got an unexpected keyword argument 'reg'
```

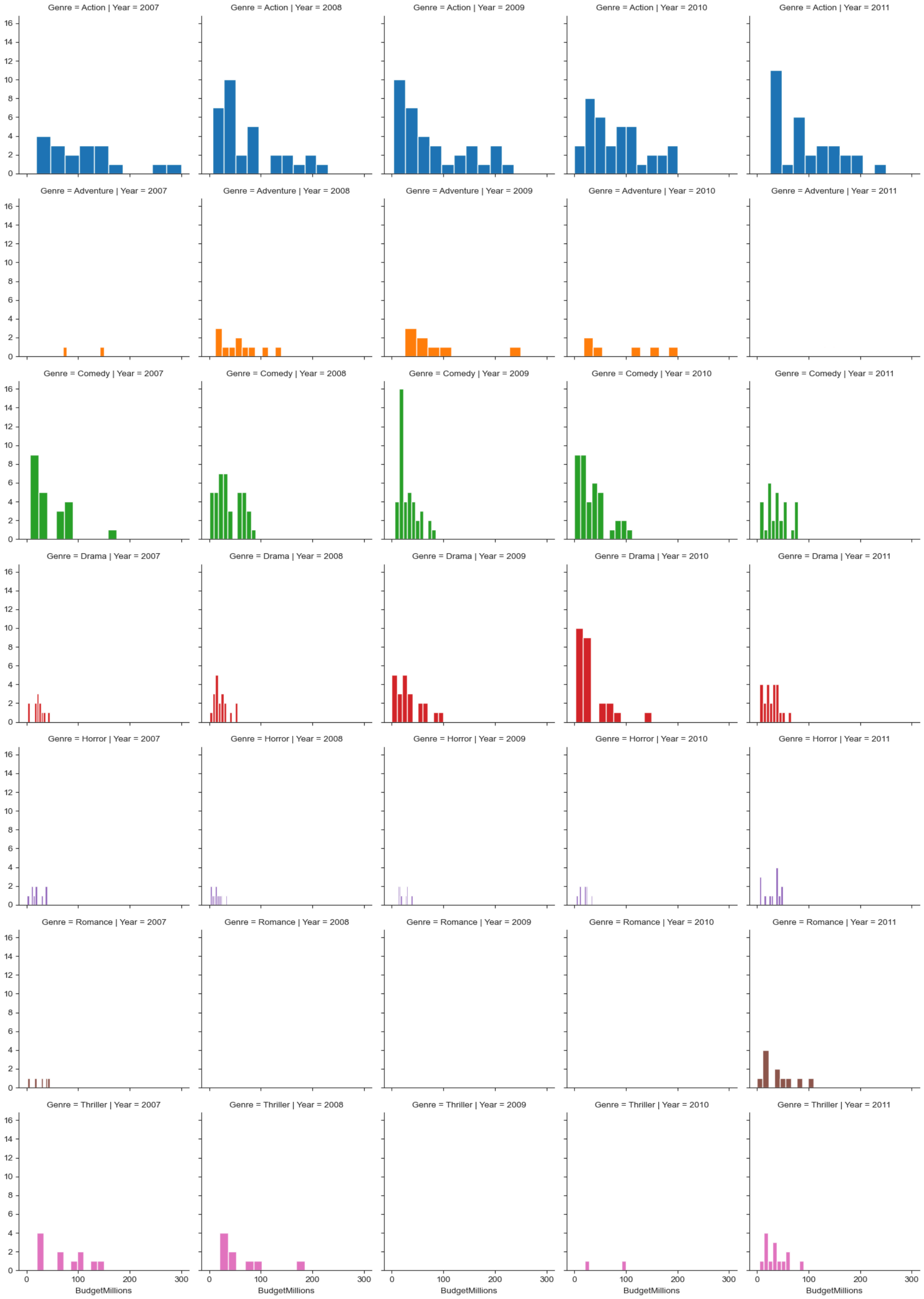
In [273…

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



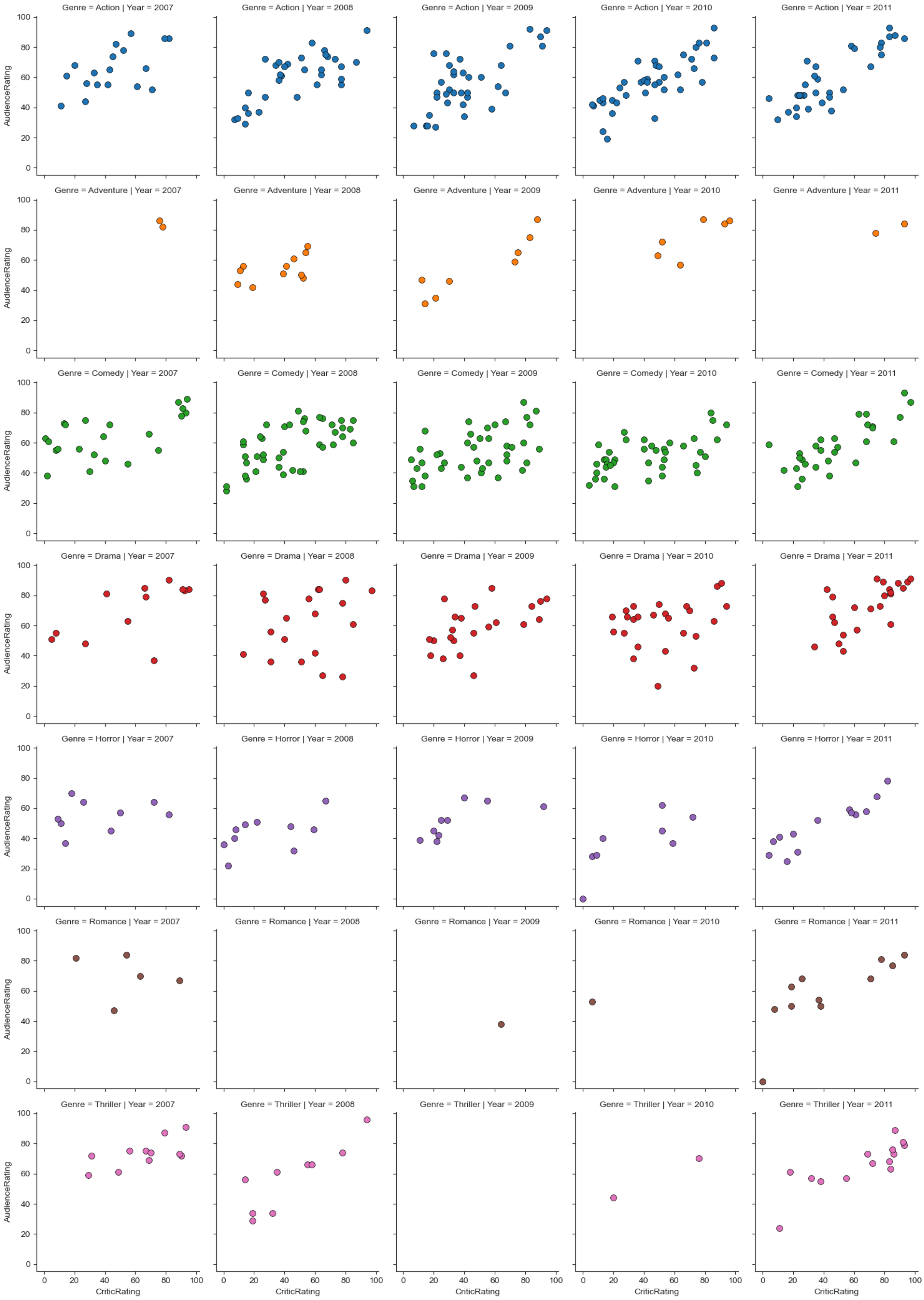
In [274…

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



In [277…

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



In [278…

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

k1=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,ax=axes[0,0])
k2=sns.kdeplot(movies.BudgetMillions,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

k4=sns.kdeplot(movies.CriticRating,movies.AudienceRating,shade=True,shade_lowest


k4b=sns.kdeplot(movies.CriticRating,movies.AudienceRating,cmap='Reds',ax=axes[1,

plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[278], line 4
      1 sns.set_style('darkgrid')
      2 f,axes=plt.subplots(2,2,figsize=(15,15))
----> 4 k1=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,ax=axes[0,0])
      5 k2=sns.kdeplot(movies.BudgetMillions,movies.CriticRating,ax=axes[0,1])
      8 k1.set(xlim=(-20,160))

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

In [280…

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

k1=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,ax=axes[0,0])
k2=sns.kdeplot(movies.BudgetMillions,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

k4=sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating',shade=True,shade_


k4b=sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating',cmap='Reds',ax=a

plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[280], line 4
      1 sns.set_style('darkgrid')
      2 f,axes=plt.subplots(2,2,figsize=(15,15))
----> 4 k1=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,ax=axes[0,0])
      5 k2=sns.kdeplot(movies.BudgetMillions,movies.CriticRating,ax=axes[0,1])
      8 k1.set(xlim=(-20,160))

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

In [281…

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

k1 = sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,ax=axes[0,0])
k2 = sns.kdeplot(movies.BudgetMillions,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(movies.CriticRating,movies.AudienceRating,shade = True,shade_lo

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

plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[281], line 4
      1 sns.set_style('darkgrid')
      2 f, axes = plt.subplots (2,2, figsize = (15,15))
----> 4 k1 = sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,ax=axes[0,
0])
      5 k2 = sns.kdeplot(movies.BudgetMillions,movies.CriticRating,ax = axes[0,
1])
      7 k1.set(xlim=(-20,160))

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

In [282…

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

k1=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,ax=axes[0,0])
k2=sns.kdeplot(movies.BudgetMillions,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

k4=sns.kdeplot(movies.CriticRating,movies.AudienceRating,shade=True,shade_lowest



plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[282], line 4
      1 sns.set_style('darkgrid')
      2 f,axes=plt.subplots(2,2,figsize=(15,15))
----> 4 k1=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,ax=axes[0,0])
      5 k2=sns.kdeplot(movies.BudgetMillions,movies.CriticRating,ax=axes[0,1])
      8 k1.set(xlim=(-20,160))

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

In [284…

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

k1=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,ax=axes[0,0])
k2=sns.kdeplot(movies.BudgetMillions,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

k4=sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating',shade=True,cmap='



plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[284], line 4
      1 sns.set_style('darkgrid')
      2 f,axes=plt.subplots(2,2,figsize=(15,15))
----> 4 k1=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,ax=axes[0,0])
      5 k2=sns.kdeplot(movies.BudgetMillions,movies.CriticRating,ax=axes[0,1])
      8 k1.set(xlim=(-20,160))

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

In [285…

```
sns.set_style('dark',{'axes.facecolor':'black'})
f,axes=plt.subplots(2,2,figsize=(15,15))
k1=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating, \
               shade=True,shade_lowest=True,cmp='inferno', \
               ax=axes[0,0])
k1b=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating, \
                cmap='cool',ax=axes[0,0])

k2=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating, \
               shade=True,shade_lowest=True,cmap='inferno', \
               ax=axes[0,1])

k2b=sns.kdeplot(movies.BudgetMillions,movies.CriticRating, \
                cmap='cool',ax=axes[0,1])
z=sns.violinplot(data=movies[movies.Genre=='Drama'], \
                 x='Year', y='CriticRating',ax=axes[1,0])

k4=sns.kdeplot(movies.CriticRating,movies.AudienceRating, \
               cmap='gist_gray_r',ax=axes[1,1])
k1.set(xlim=(-20,160))
k2.set(xlim=(-20,160))

plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[285], line 3
      1 sns.set_style('dark',{'axes.facecolor':'black'})
      2 f,axes=plt.subplots(2,2,figsize=(15,15))
----> 3 k1=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating, \
      4                shade=True,shade_lowest=True,cmp='inferno', \
      5                ax=axes[0,0])
      6 k1b=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating, \
      7                 cmap='cool',ax=axes[0,0])
      9 k2=sns.kdeplot(movies.BudgetMillions,movies.AudienceRating, \
     10                shade=True,shade_lowest=True,cmap='inferno', \
     11                ax=axes[0,1])

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

```
In [ ]:
```