

# Seaborn Advance

We use movies dataset for Visualization.

Steps: Firstly we have cleaned the Data. Secondly we Visualized using Seaborn

```
jointplot()
displot()
histplot()
kdeplot()
boxplot()
violinplot()
lmpplot()
FacetGrid()
set_theme()
set_style()
```

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

```
In [2]: movies=pd.read_csv(r"C:\Users\ymani\Dropbox\PC\Downloads\Movie-Rating.csv")
```

```
In [3]: movies
```

```
Out[3]:
```

	Film	Genre	Rotten Tomatoes Ratings %	Audience Ratings %	Budget (million \$)	Year of release
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009
...	...	...	...	...	...	...
554	Your Highness	Comedy	26	36	50	2011
555	Youth in Revolt	Comedy	68	52	18	2009
556	Zodiac	Thriller	89	73	65	2007
557	Zombieland	Action	90	87	24	2009
558	Zookeeper	Comedy	14	42	80	2011

559 rows × 6 columns

```
In [4]: len(movies)
```

Out[4]: 559

```
In [5]: import numpy as np
print(np.__version__)
```

1.26.4

```
In [6]: import pandas as pd
print(pd.__version__)
```

2.2.2

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

```
Out[7]: Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
              'Budget (million $)', 'Year of release'],
              dtype='object')
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Film                                  559 non-null    object
1   Genre                                559 non-null    object
2   Rotten Tomatoes Ratings %           559 non-null    int64
3   Audience Ratings %                  559 non-null    int64
4   Budget (million $)                  559 non-null    int64
5   Year of release                      559 non-null    int64
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

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

Out[9]: (559, 6)

```
In [10]: movies.head()
```

Out[10]:

	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 [11]: movies.tail()
```

Out[11]:

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

In [13]: `movies.head(1)`

Out[13]:

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

In [14]: `movies.describe()` *#descriptive Statistics*

Out[14]:

	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 [15]: `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 [16]: movies.Film=movies.Film.astype('category')
movies.Genre=movies.Genre.astype('category')
movies.Year=movies.Year.astype('category')
```

```
In [17]: print(movies.Genre)
print(movies.Film)
print(movies.Year)
```

```

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', 'Romance', 'Thriller']
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']
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 [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   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

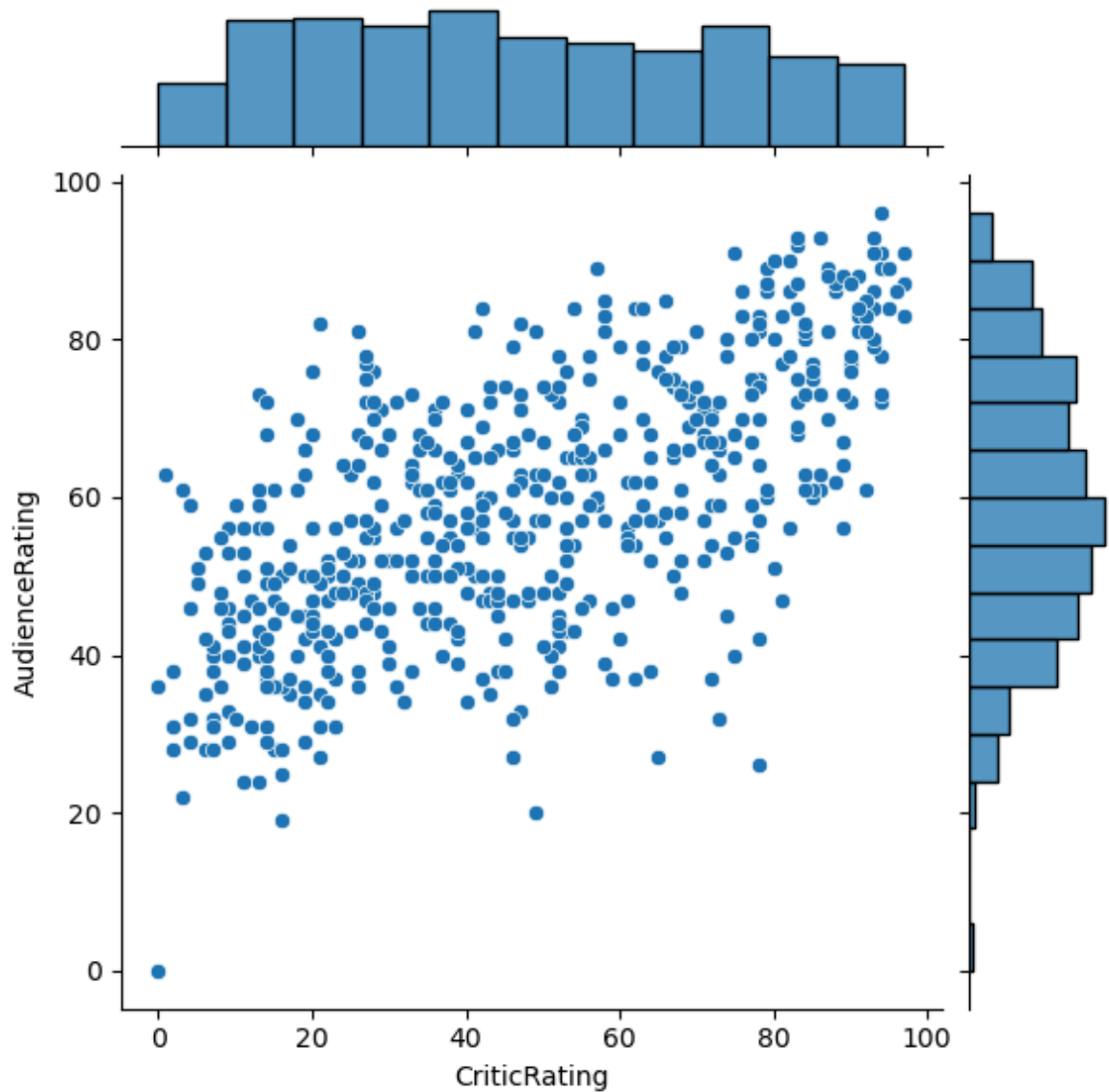
```

In [19]: `from matplotlib import pyplot as plt #Visualizations
import seaborn as sns # advance visualizations`

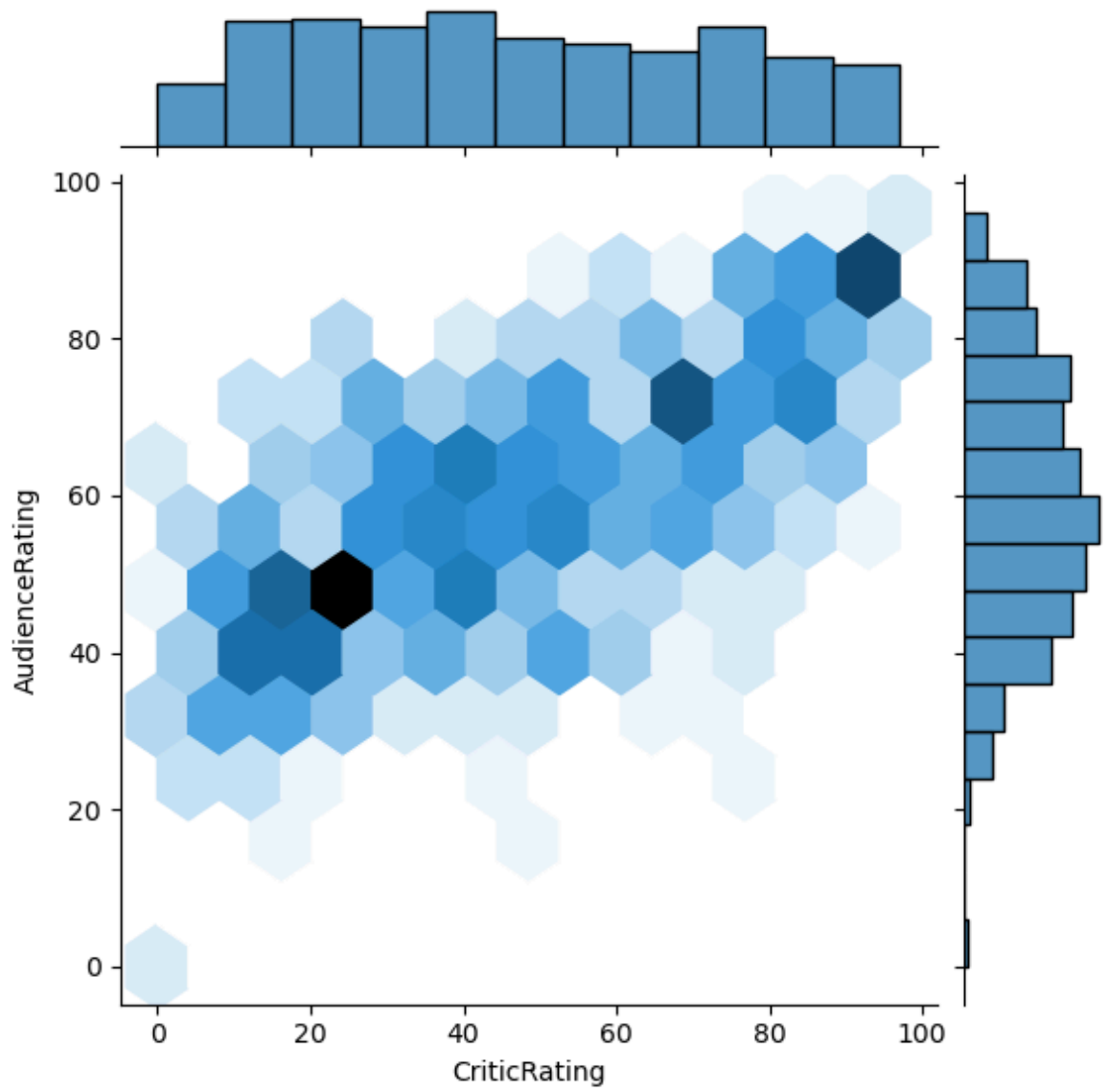
```
%matplotlib inline

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

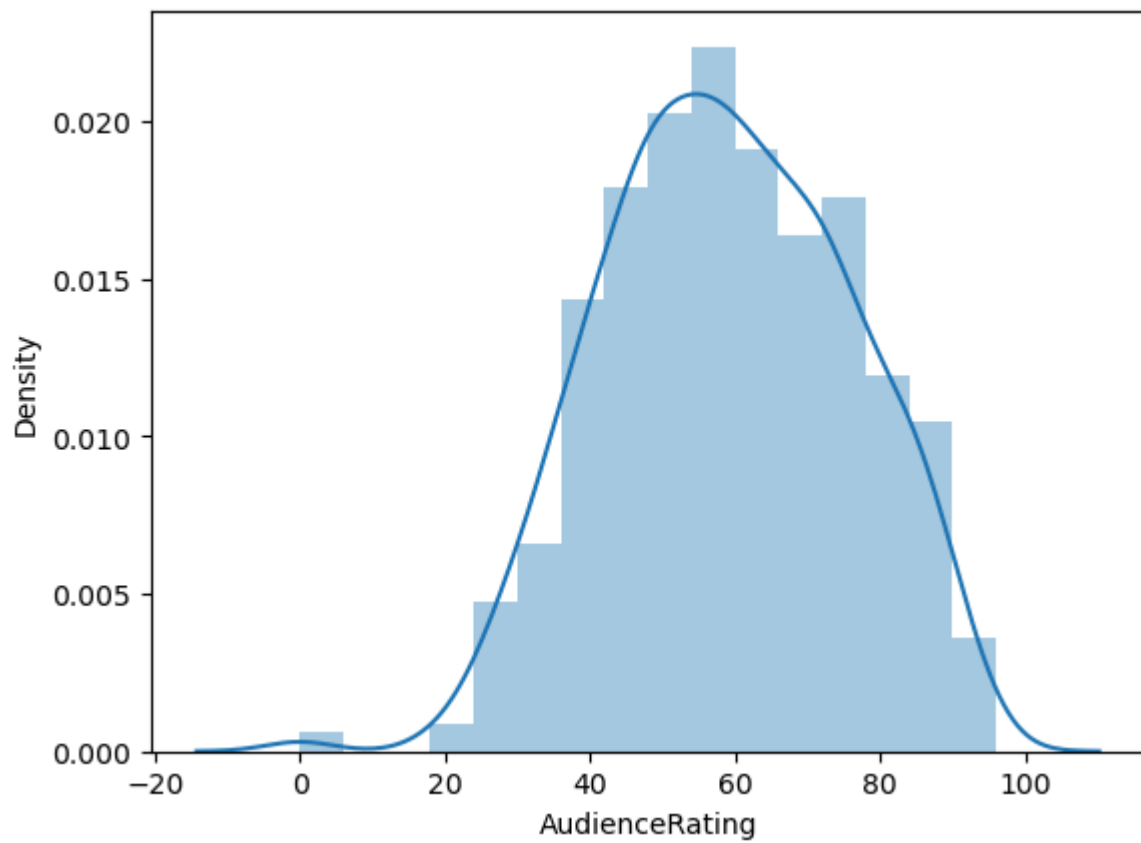
```
In [20]: j=sns.jointplot(data=movies, x='CriticRating',y='AudienceRating')
plt.show()
```



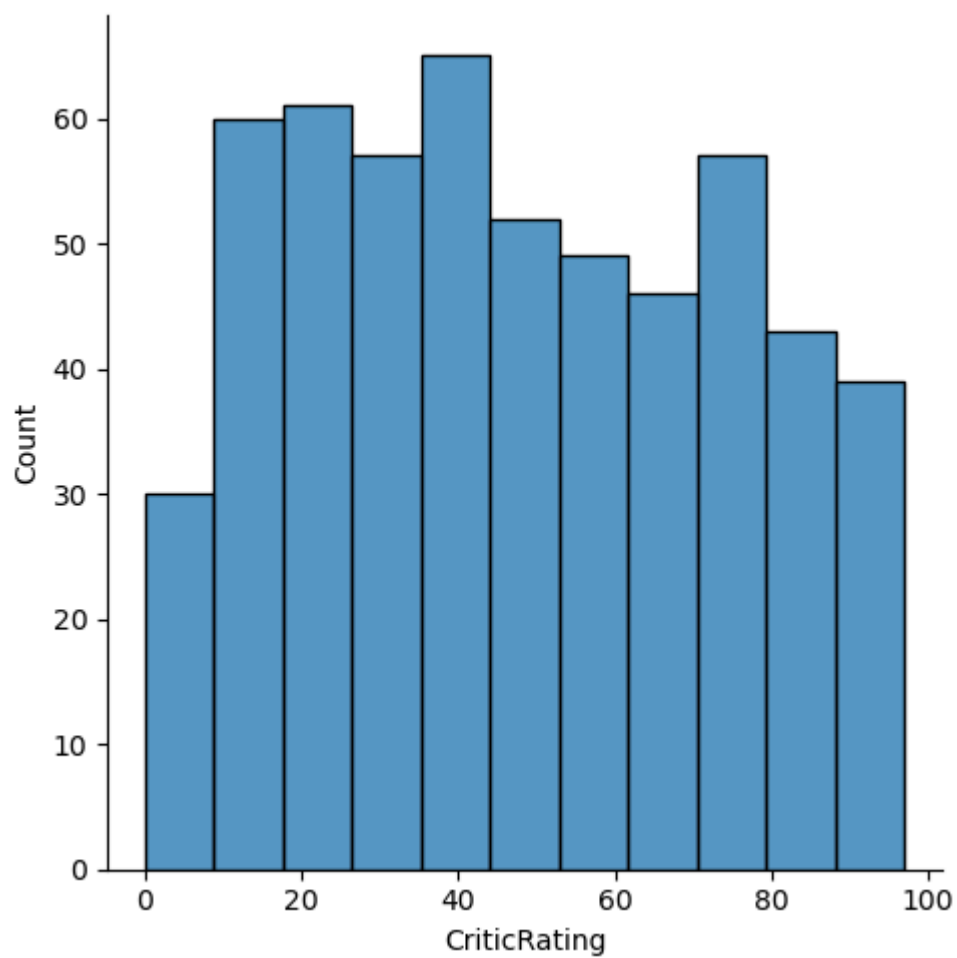
```
In [21]: j=sns.jointplot(data=movies, x='CriticRating',y='AudienceRating',kind='hex')
plt.show()
```



```
In [22]: m1=sns.distplot(movies.AudienceRating)
plt.show()
```

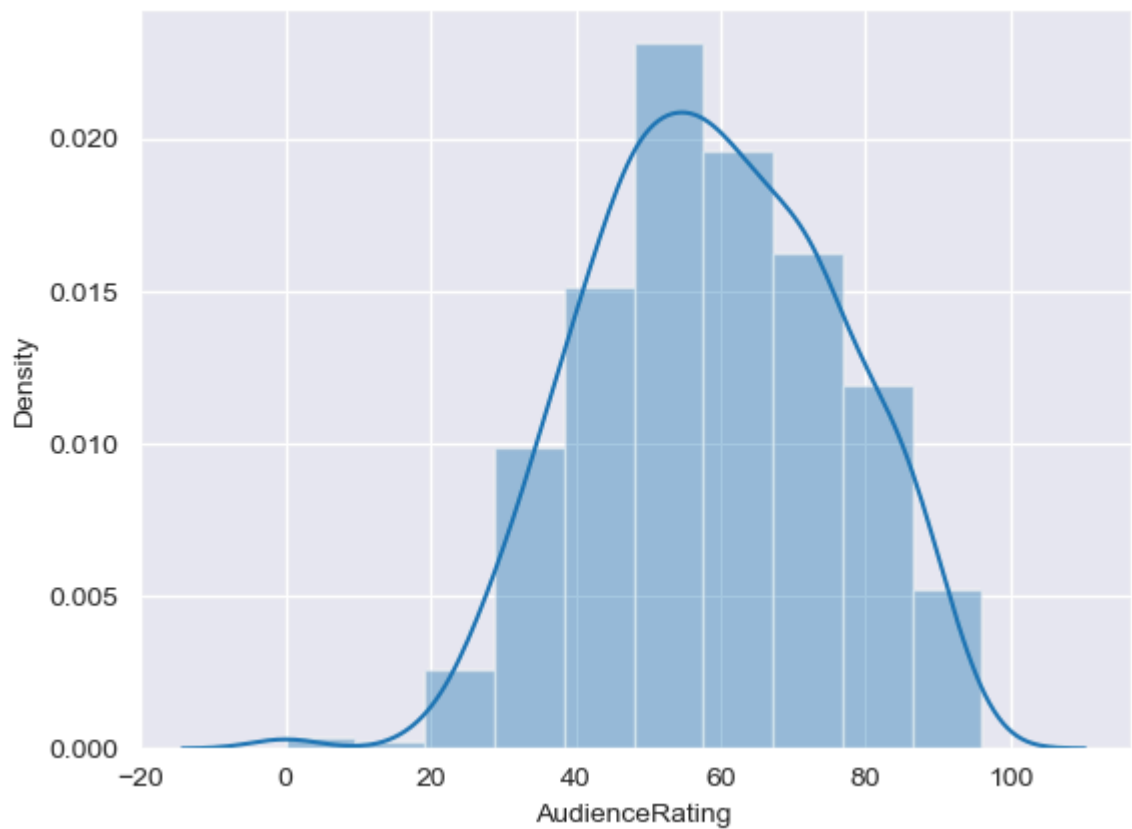


```
In [23]: m1=sns.displot(movies.CriticRating)
plt.show()
```

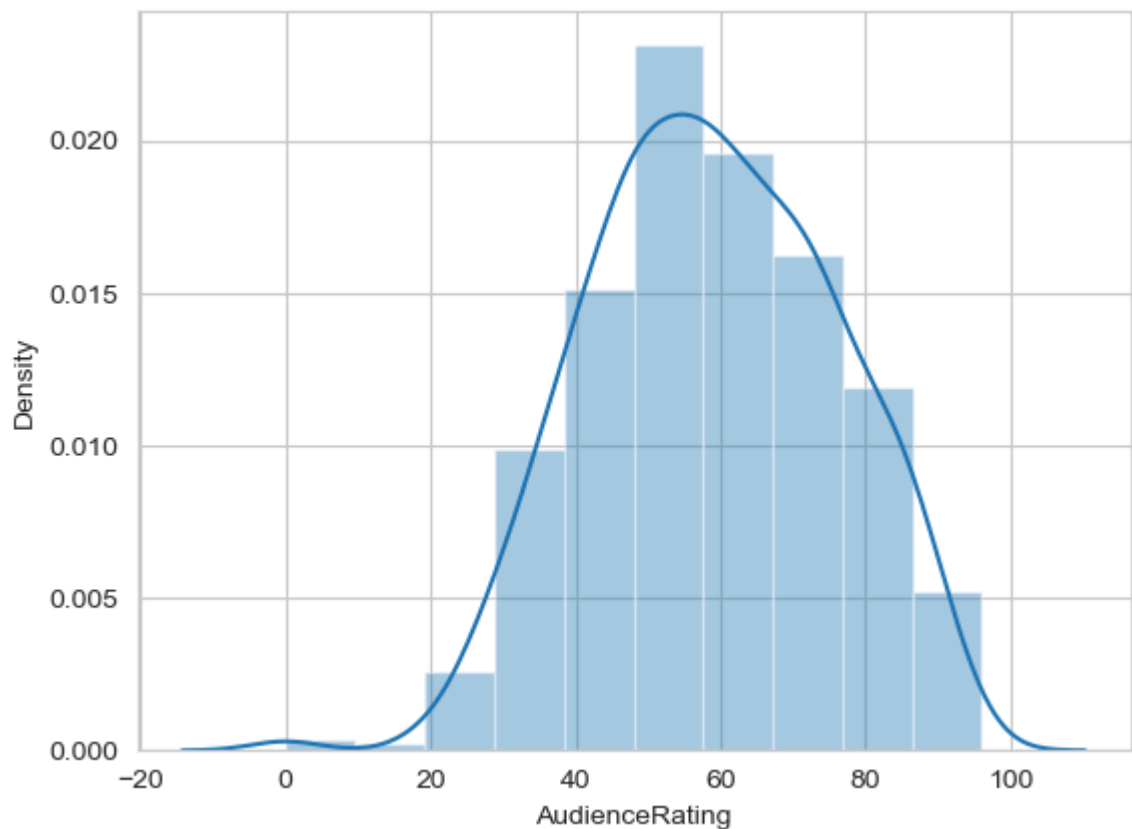




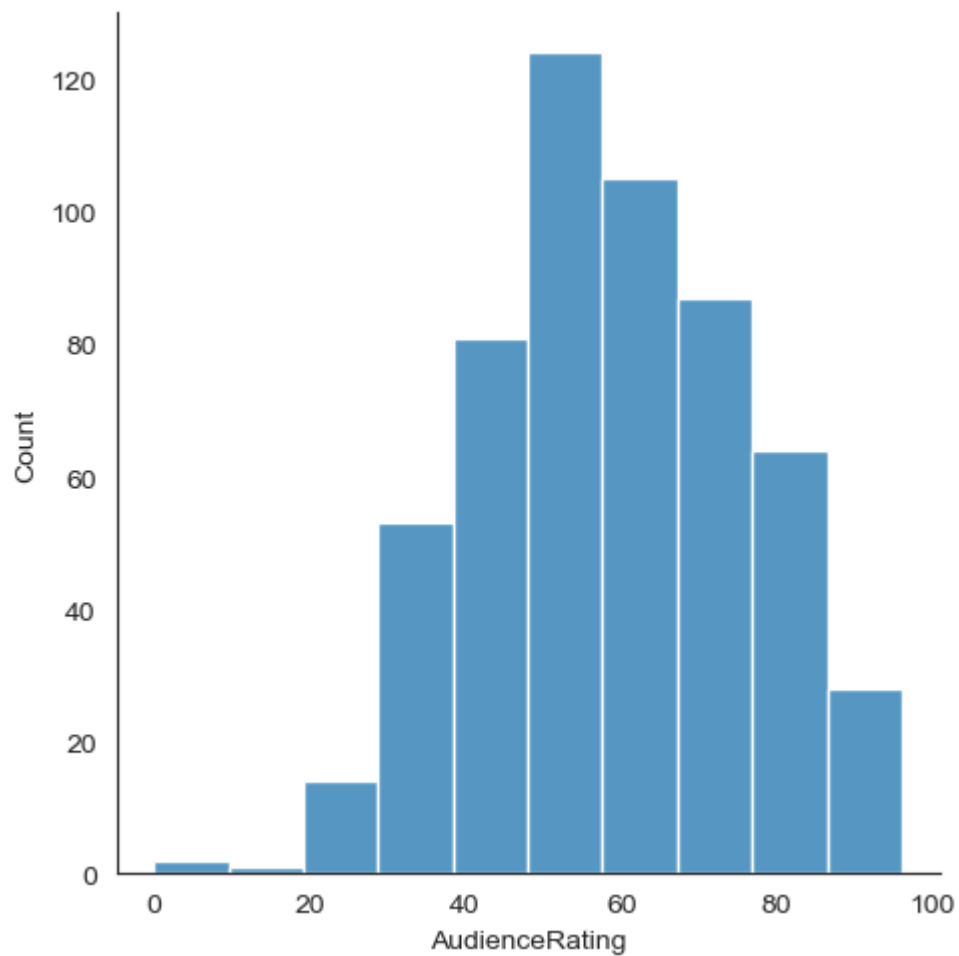
```
In [24]: sns.set_style('darkgrid') #darkgrid,whitegrid,dark,white,ticks
m3=sns.distplot(movies.AudienceRating,bins=10)
plt.show()
```



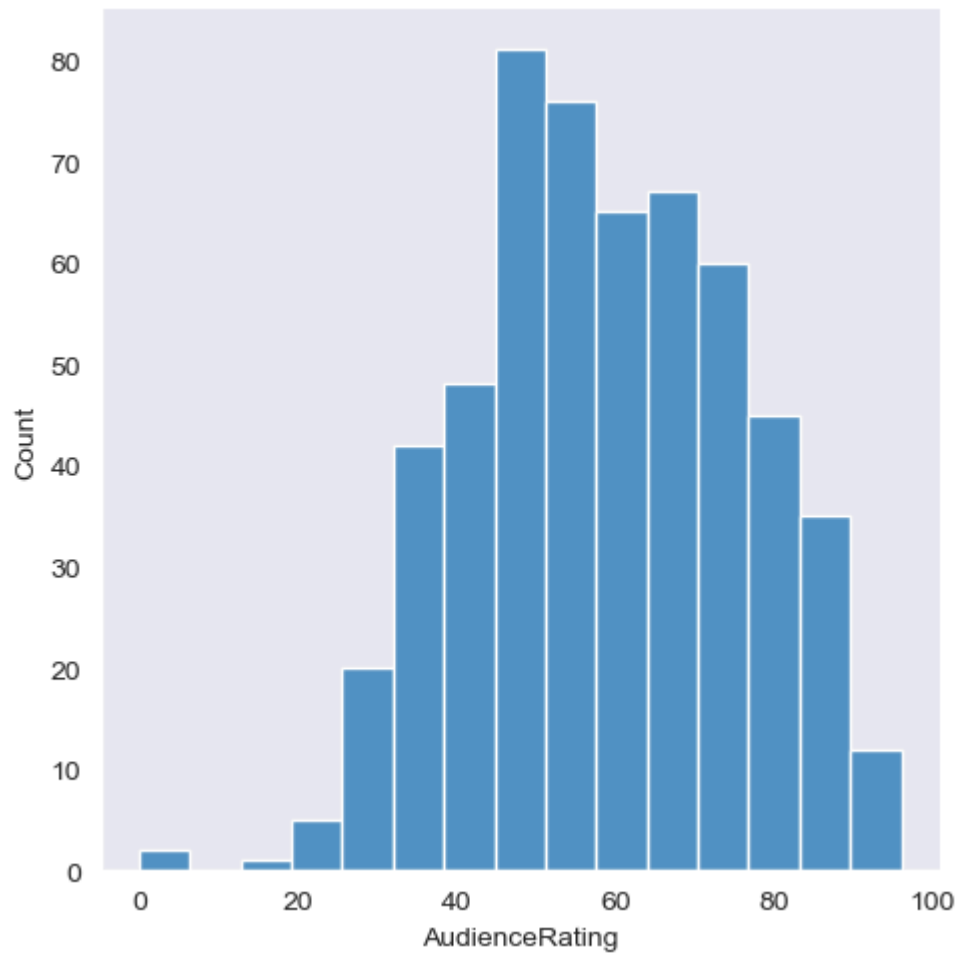
```
In [25]: sns.set_style('whitegrid') #darkgrid,whitegrid,dark,white,ticks
m3=sns.distplot(movies.AudienceRating,bins=10)
plt.show()
```



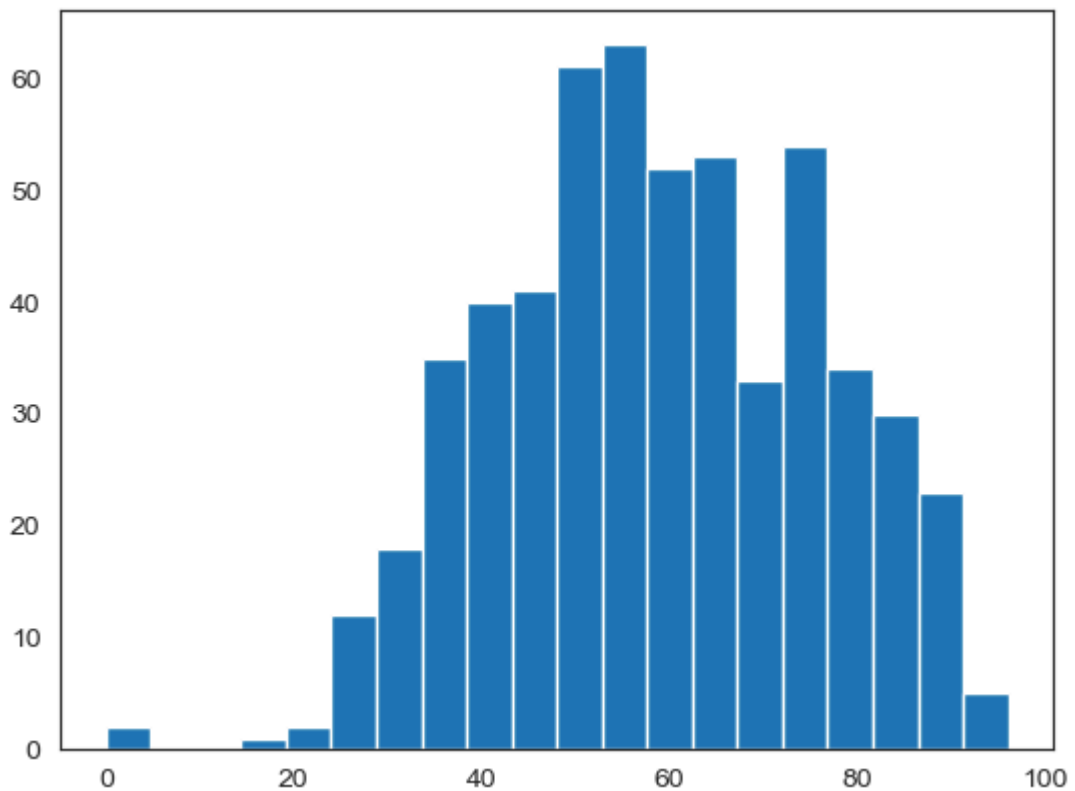
```
In [26]: sns.set_style('white') #darkgrid,whitegrid,dark,white,ticks
m3=sns.displot(movies.AudienceRating,bins=10)
plt.show()
```



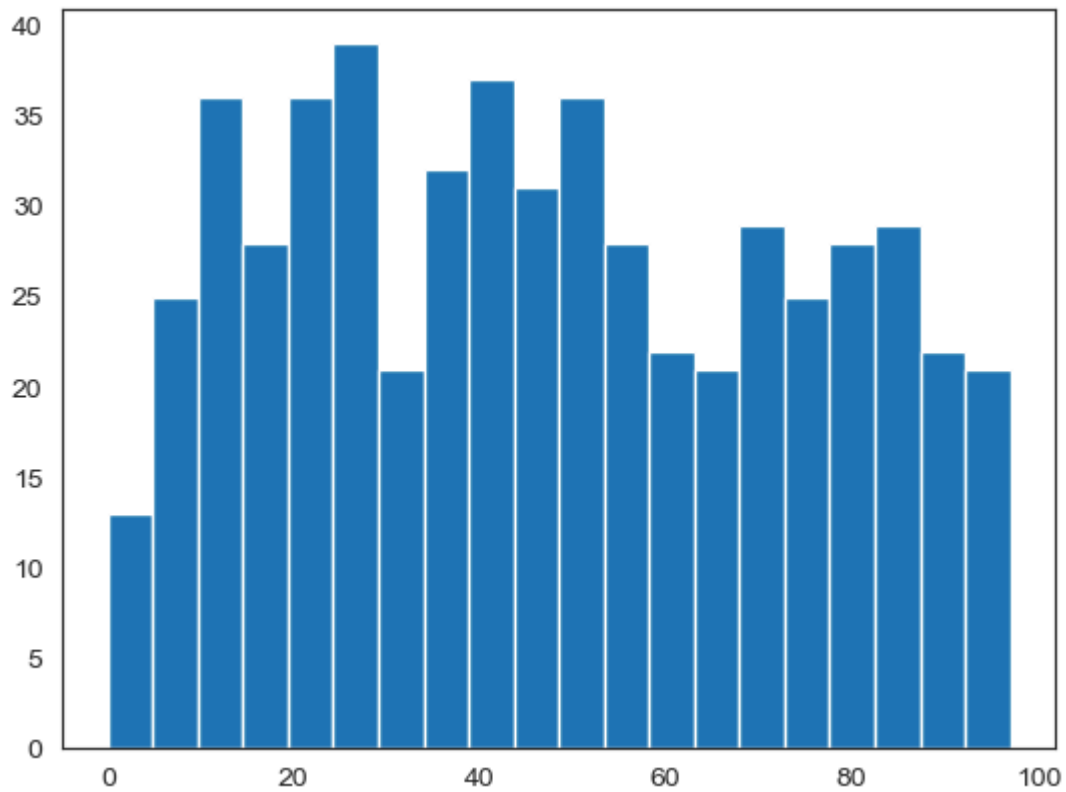
```
In [27]: sns.set_style('dark') #darkgrid,whitegrid,dark,white,ticks
m3=sns.displot(movies.AudienceRating,bins=15)
plt.show()
```



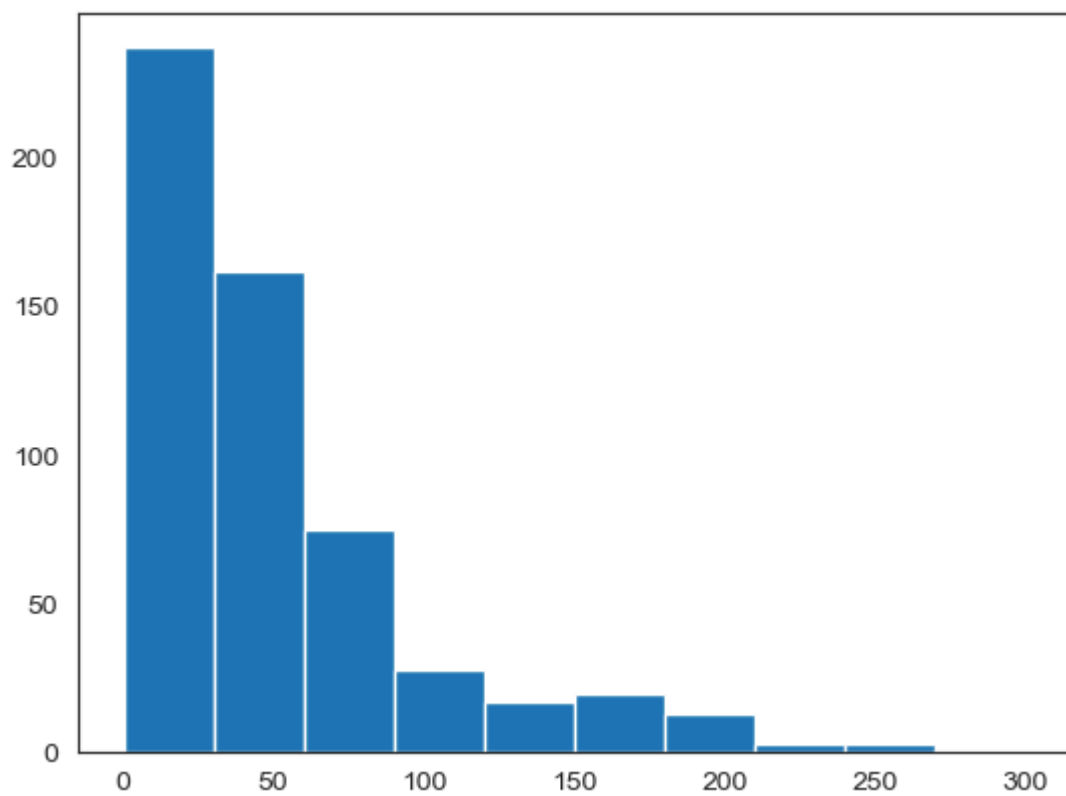
```
In [28]: sns.set_style('white') # normal distribution and called as bell curve
n1=plt.hist(movies.AudienceRating,bins=20)
plt.show()
```



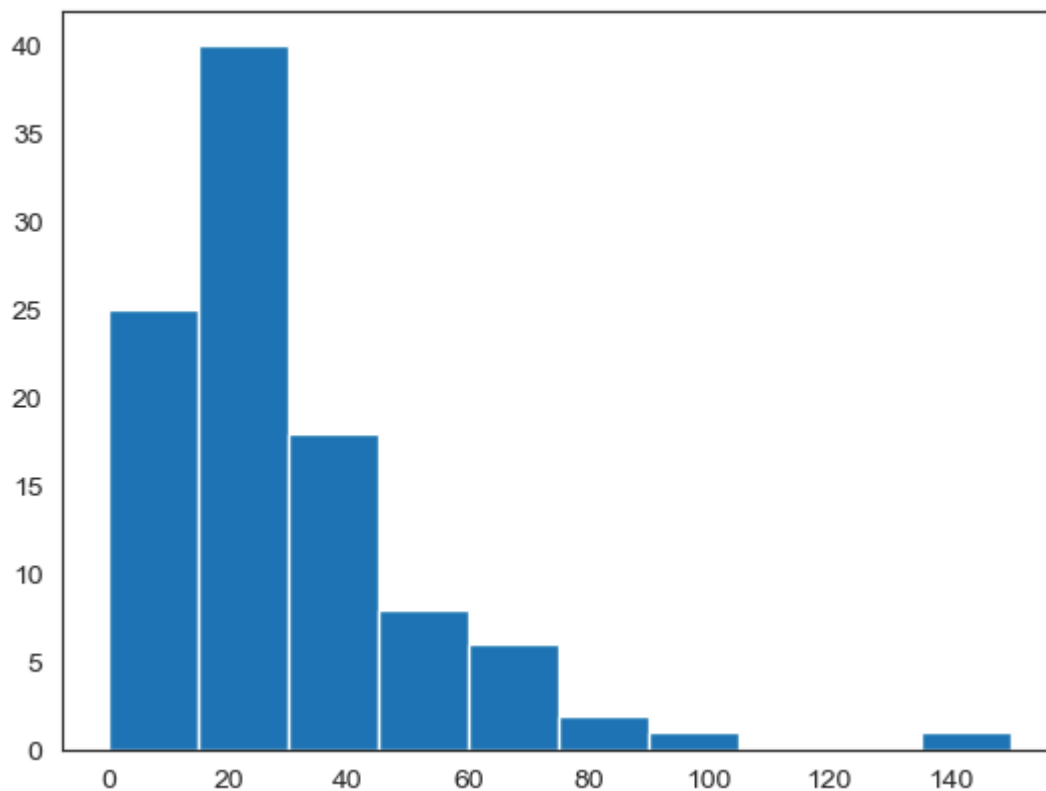
```
In [29]: n1=plt.hist(movies.CriticRating,bins=20)
plt.show()
```



```
In [30]: plt.hist(movies.BudgetMillions)
plt.show()
```



```
In [31]: plt.hist(movies[movies.Genre=='Drama'].BudgetMillions)
plt.show()
```



```
In [32]: movies.head()
```

```
Out[32]:
```

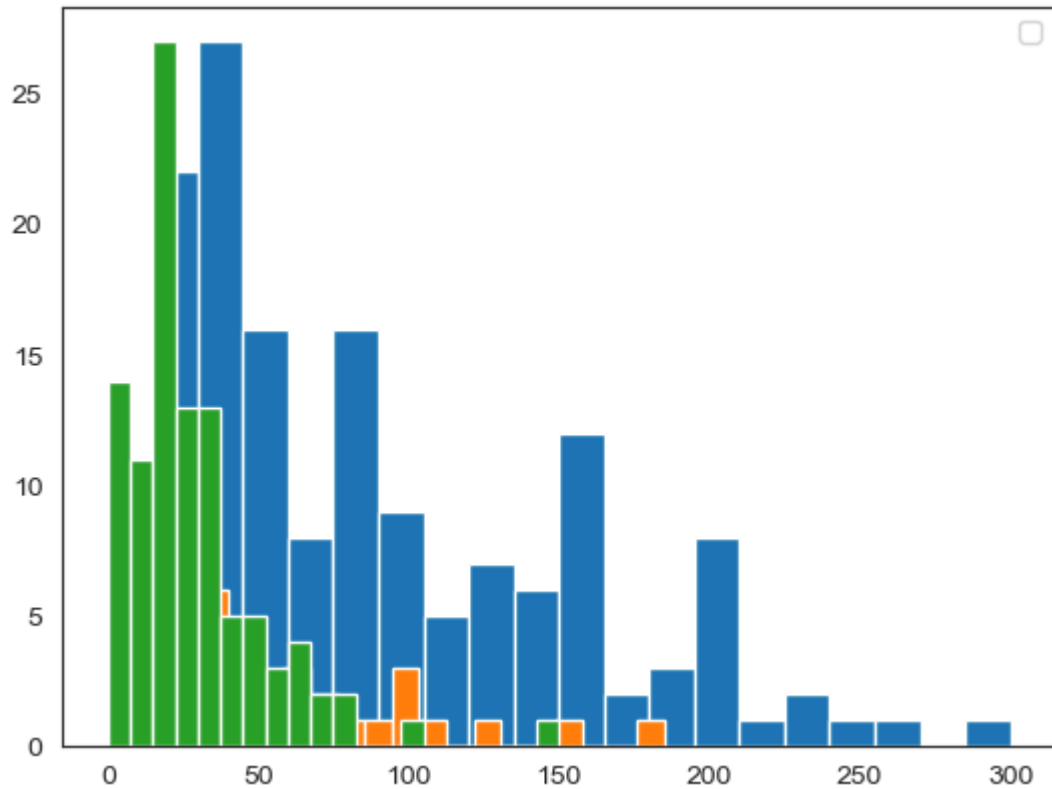
	Film	Genre	CriticRating	AudienceRating	BudgetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009

```
In [33]: movies.Genre.unique
```

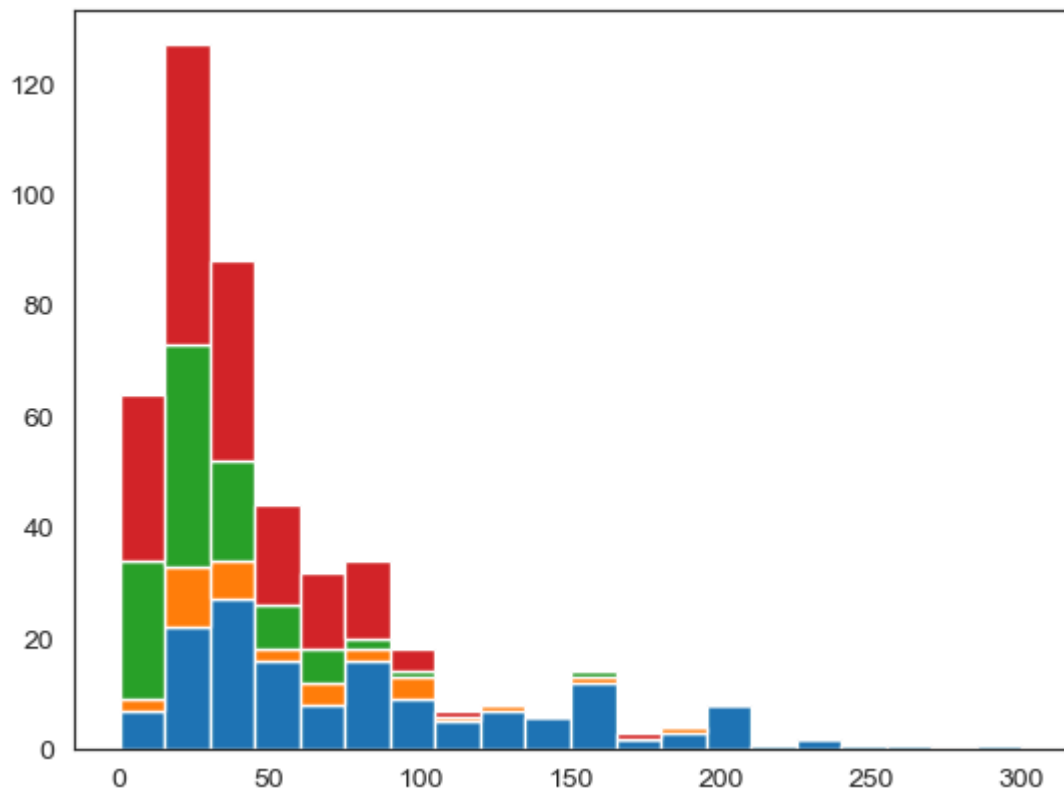
```
Out[33]: <bound method Series.unique of 0          Comedy
1          Adventure
2           Action
3          Adventure
4           Comedy
...
554        Comedy
555        Comedy
556       Thriller
557        Action
558        Comedy
Name: Genre, Length: 559, dtype: category
Categories (7, object): ['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance', 'Thriller']>
```

```
In [34]: 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()
```



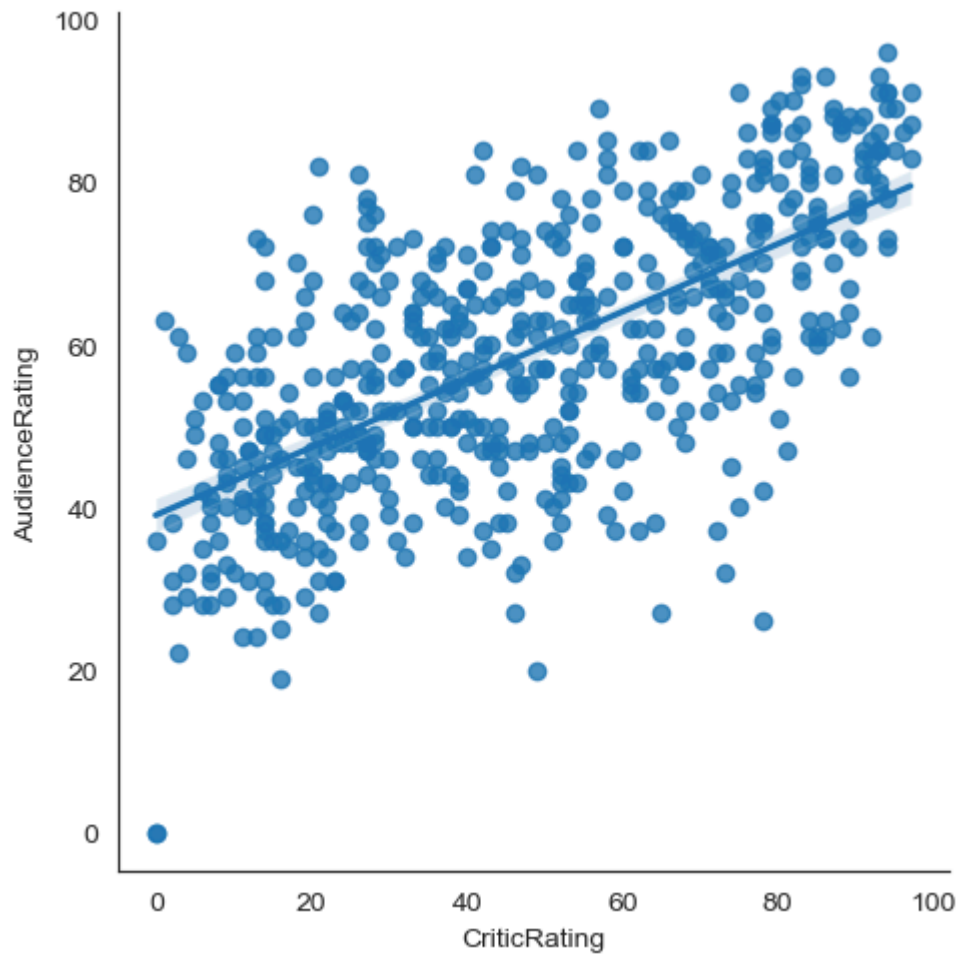
```
In [35]: plt.hist([movies[movies.Genre=='Action'].BudgetMillions,\n                  movies[movies.Genre=='Thriller'].BudgetMillions,\n                  movies[movies.Genre=='Drama'].BudgetMillions,\n                  movies[movies.Genre=='Comedy'].BudgetMillions],bins=20,stacked=True)\nplt.show()
```



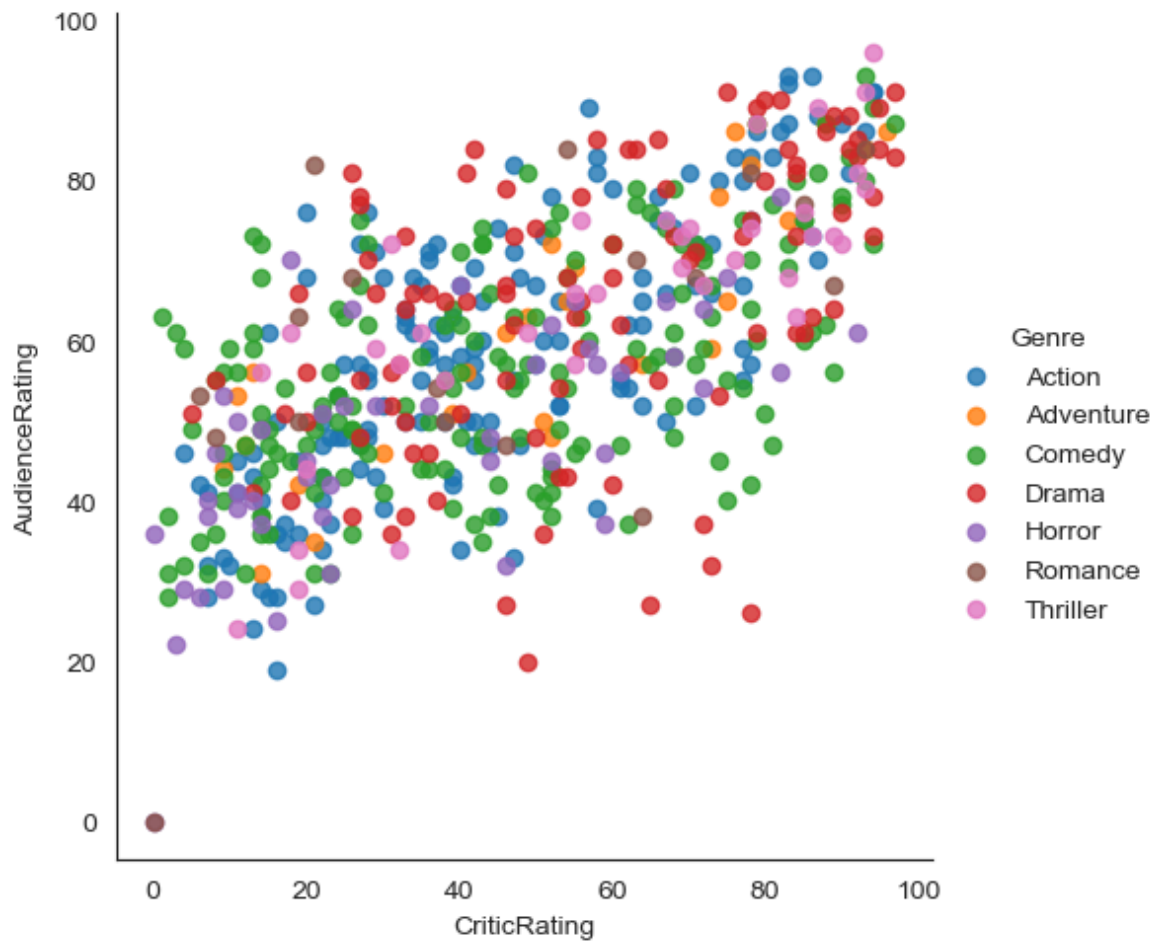
```
In [36]: for gen in movies.Genre.cat.categories:  
         print(gen)
```

Action  
Adventure  
Comedy  
Drama  
Horror  
Romance  
Thriller

```
In [37]: vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',fit_reg=True)  
         plt.show()
```

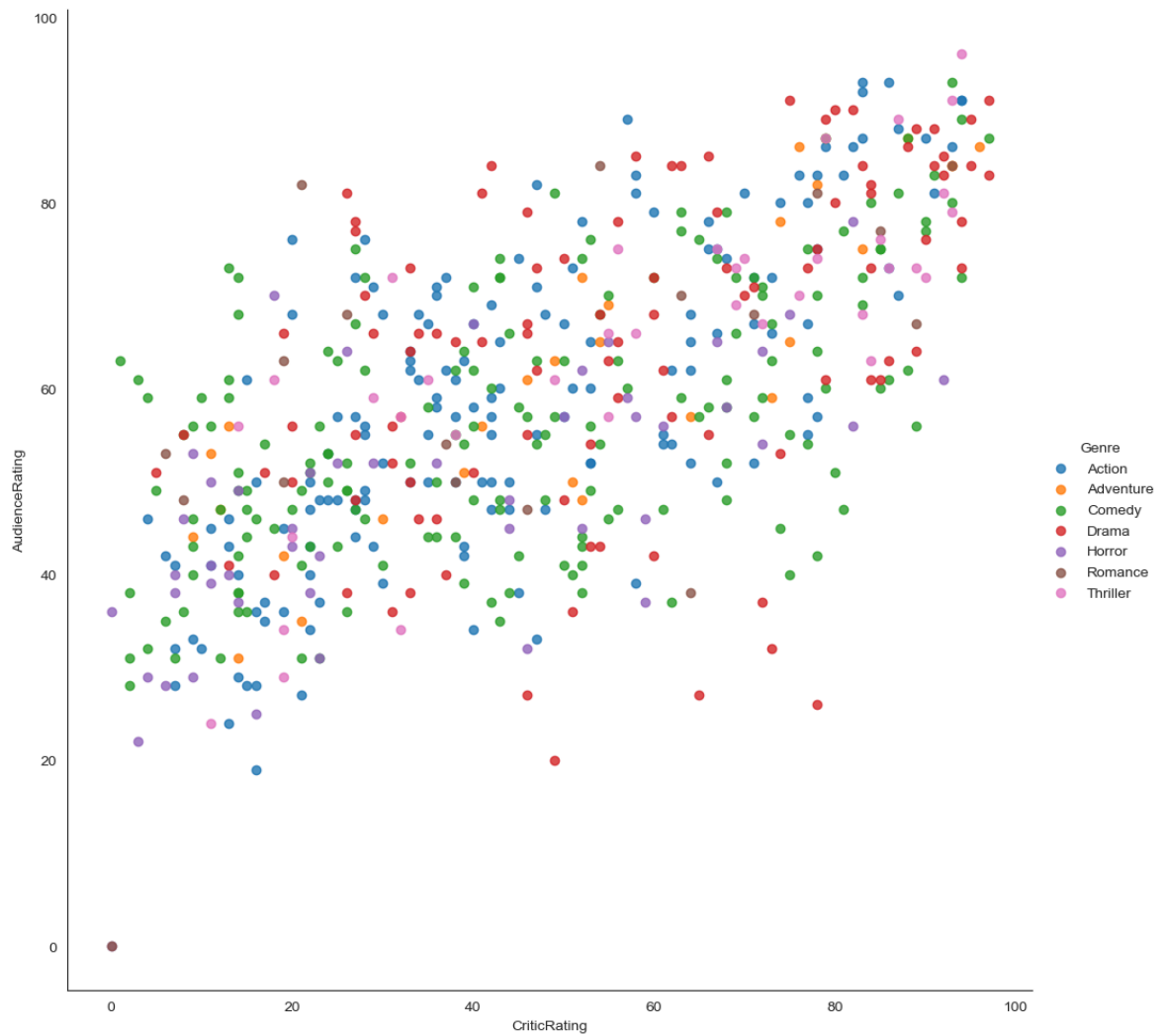


```
In [38]: vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',fit_reg=False,hu  
         plt.show()
```



```
In [39]: vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',\
                           fit_reg=False, hue = 'Genre',height=10,aspect=1)\
plt.show()
```

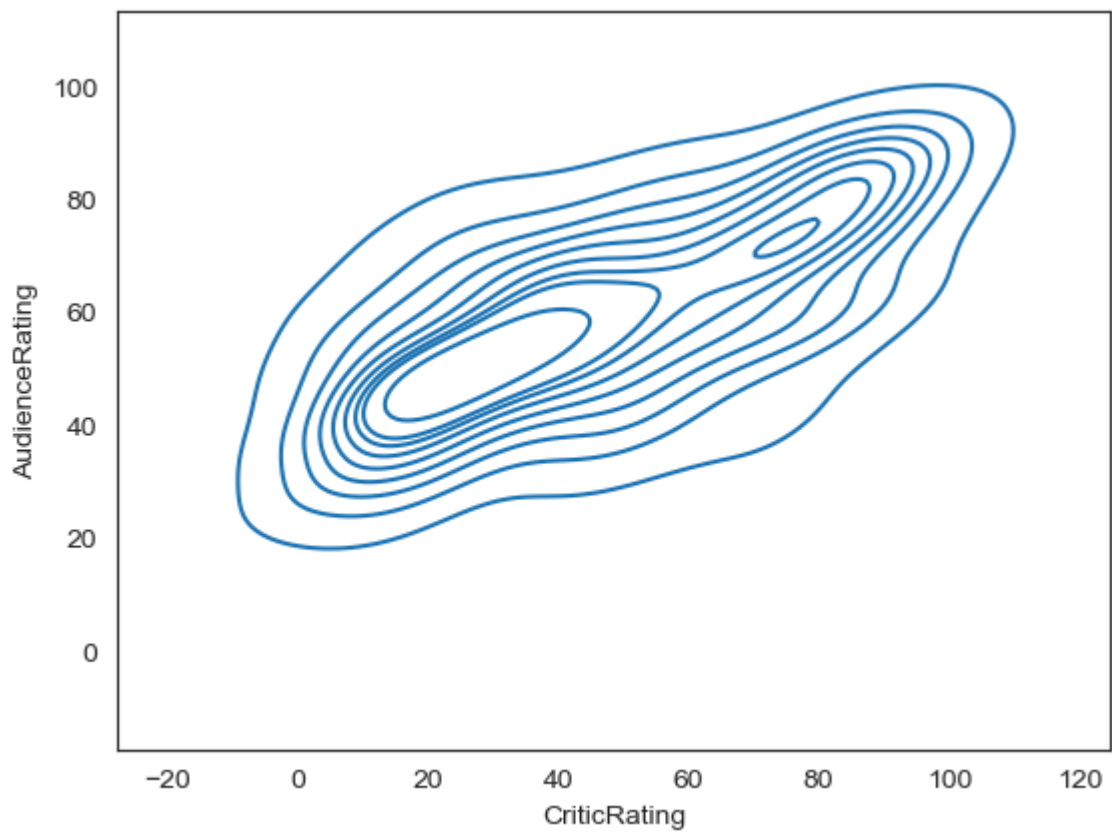




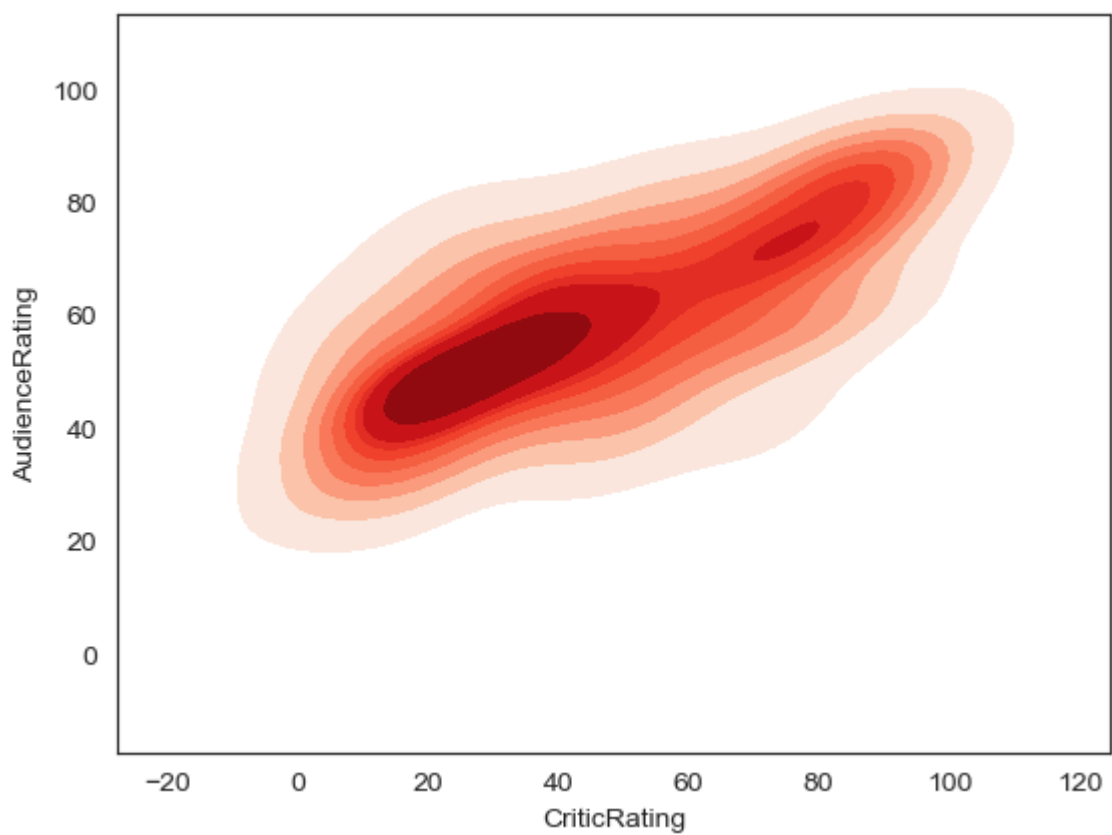
```
In [40]: #Kernel density estimate plot
movies.info()
```

```
<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
```

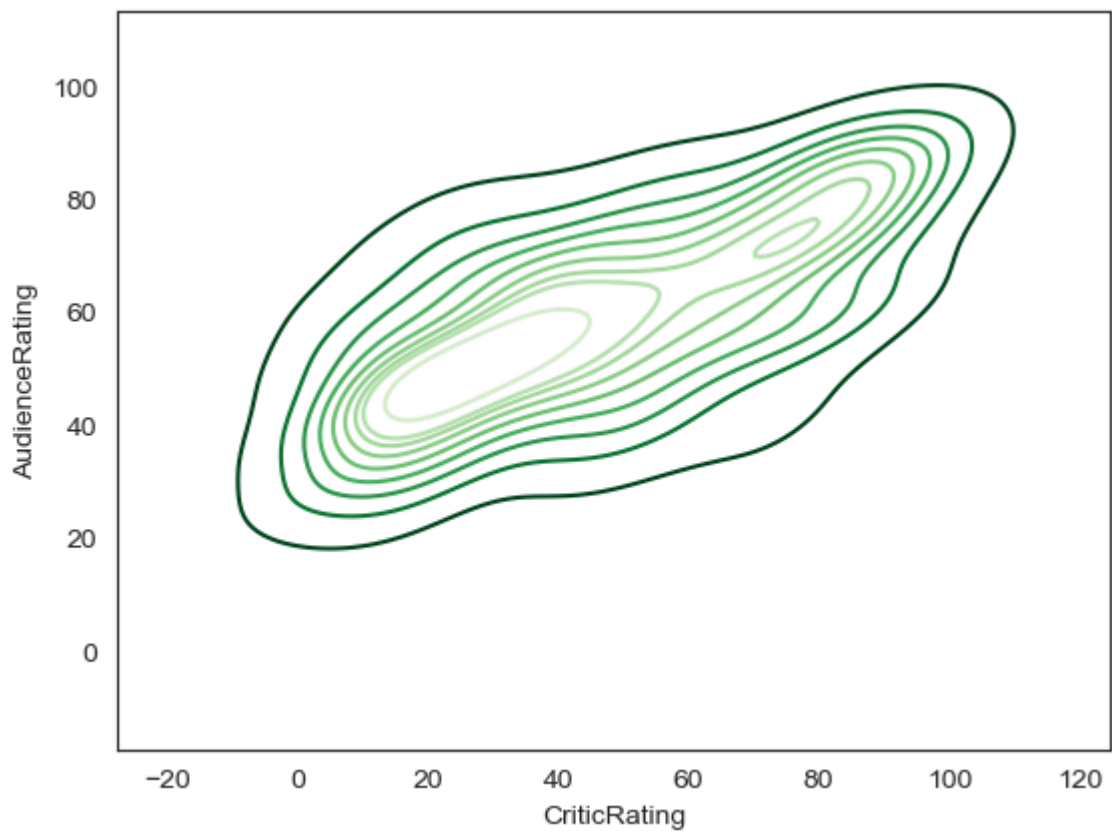
```
In [41]: k1=sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating)
plt.show()
```



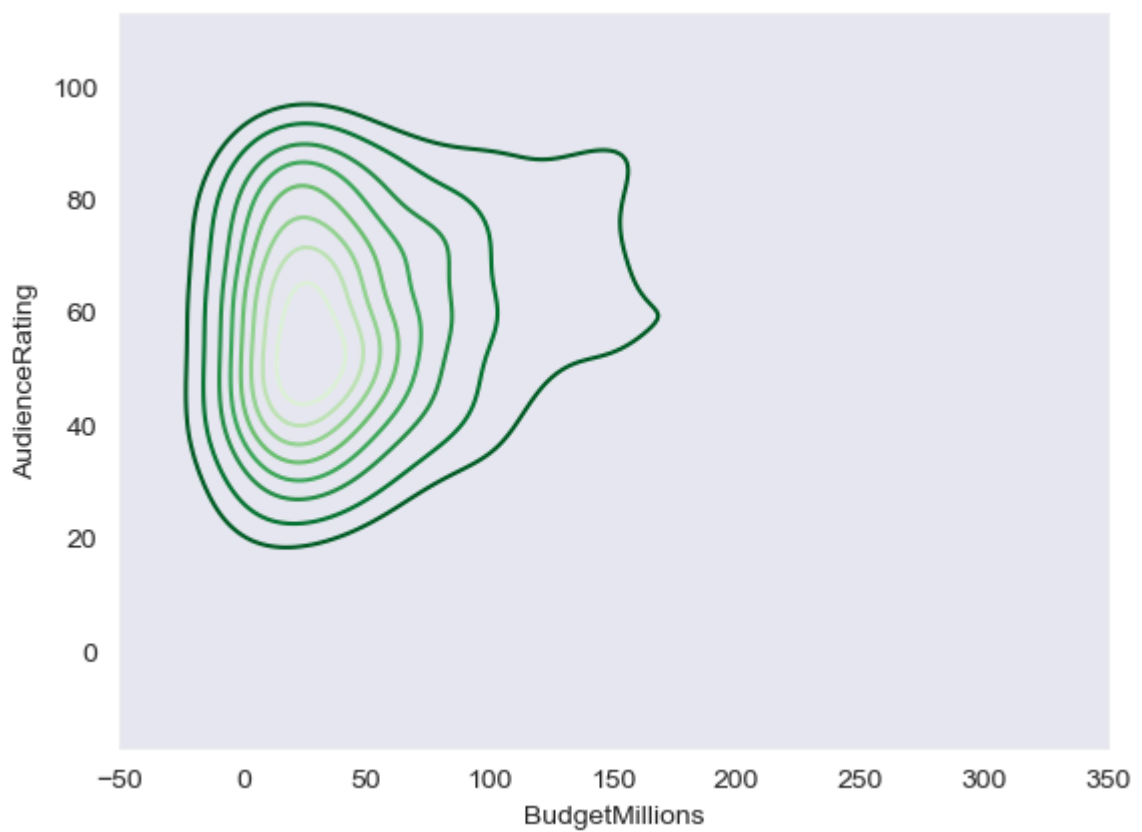
```
In [42]: k1=sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,shade=True,shade_lowest=False,plt.show())
```



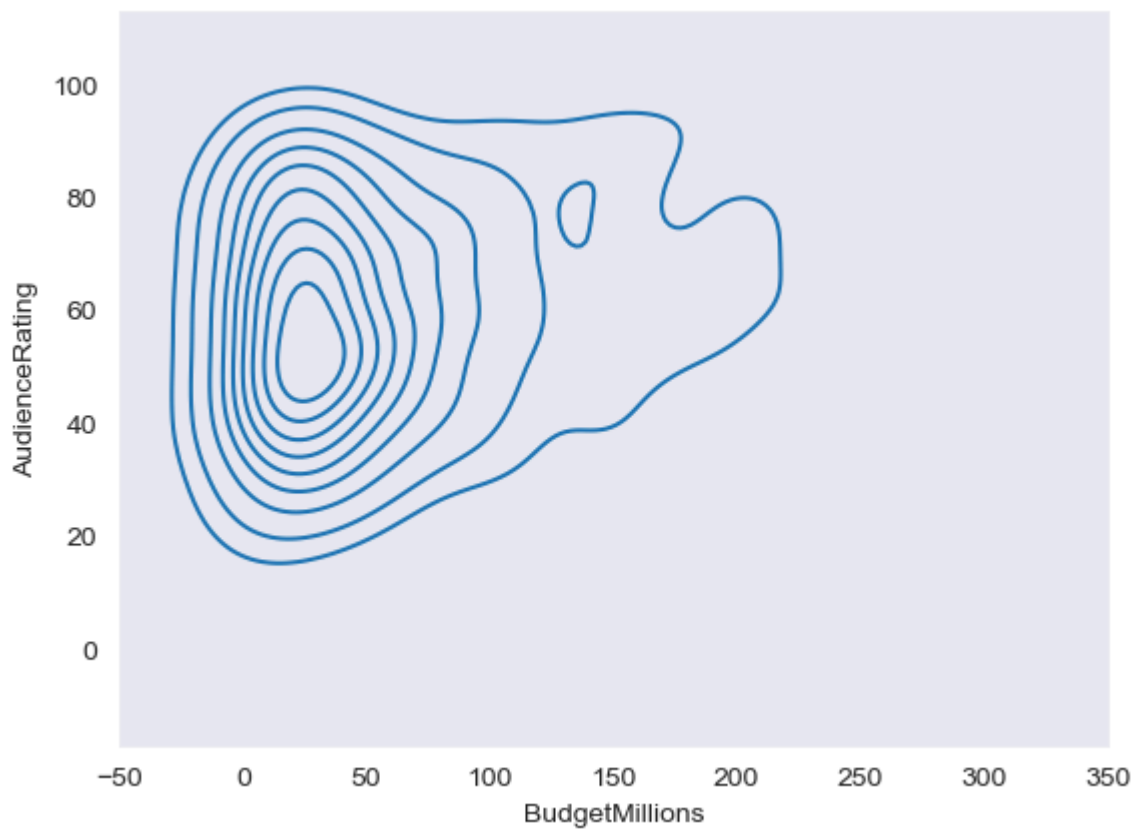
```
In [43]: k2=sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,shade_lowest=False,plt.show())
```



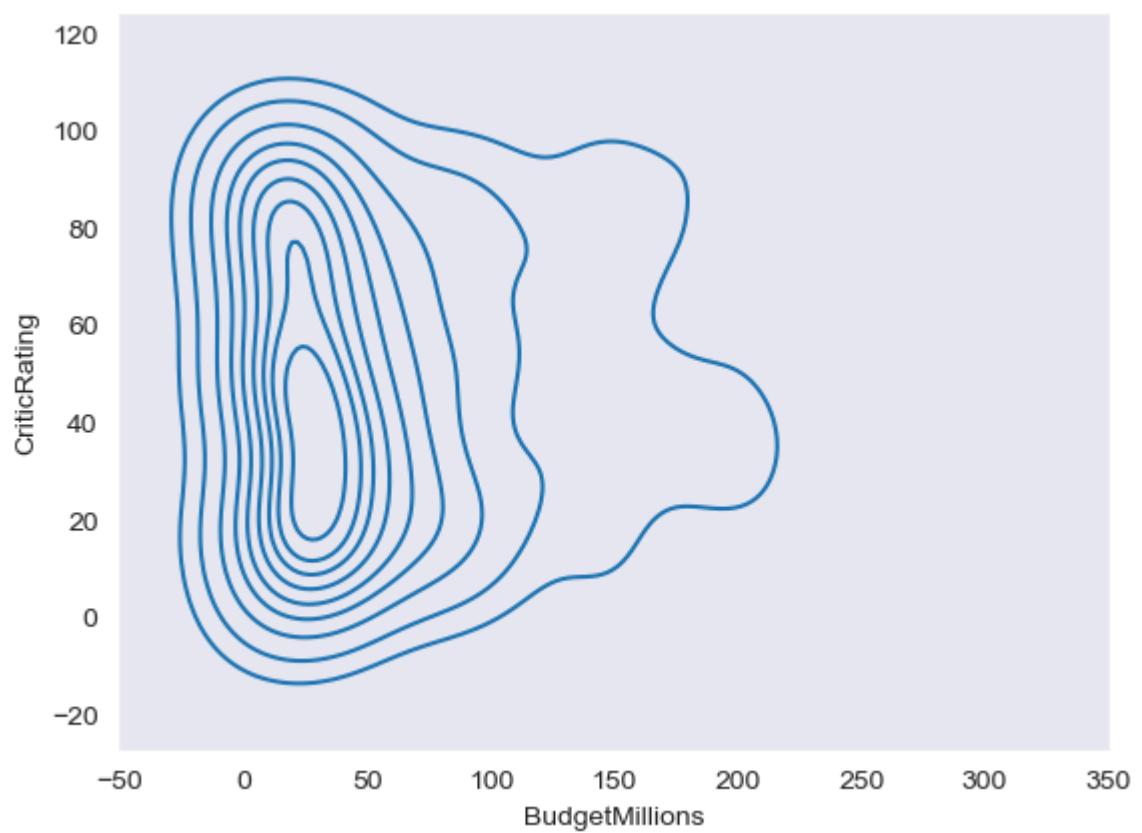
```
In [44]: sns.set_style('dark')
k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,shade_lowest=True)
plt.show()
```



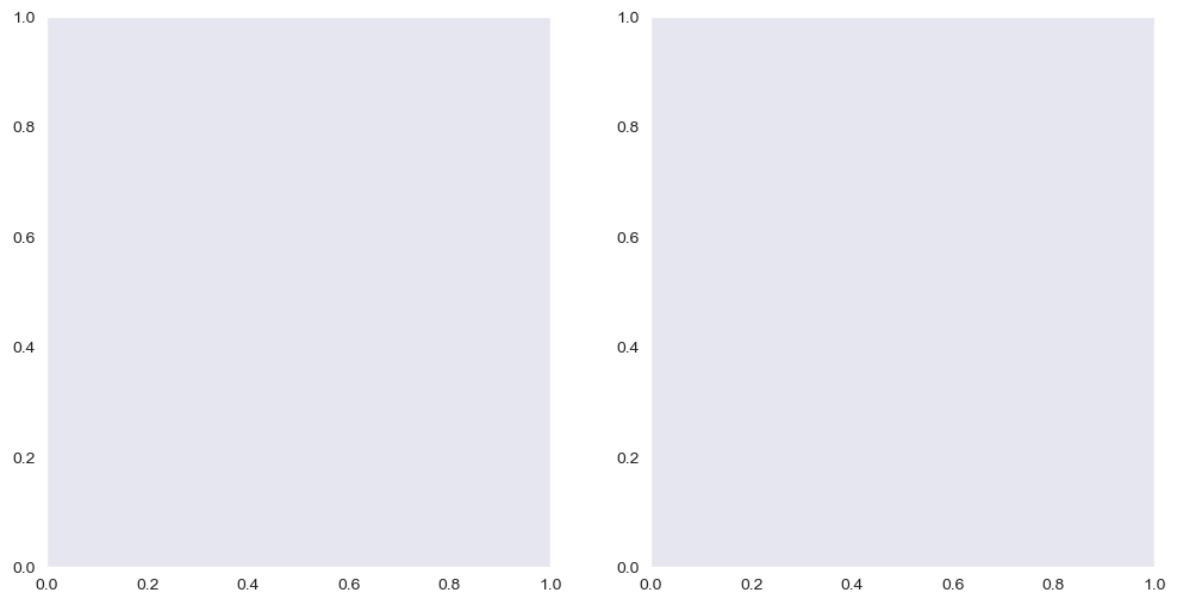
```
In [45]: sns.set_style('dark')
k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating)
plt.show()
```



```
In [46]: k2=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating)
plt.show()
```

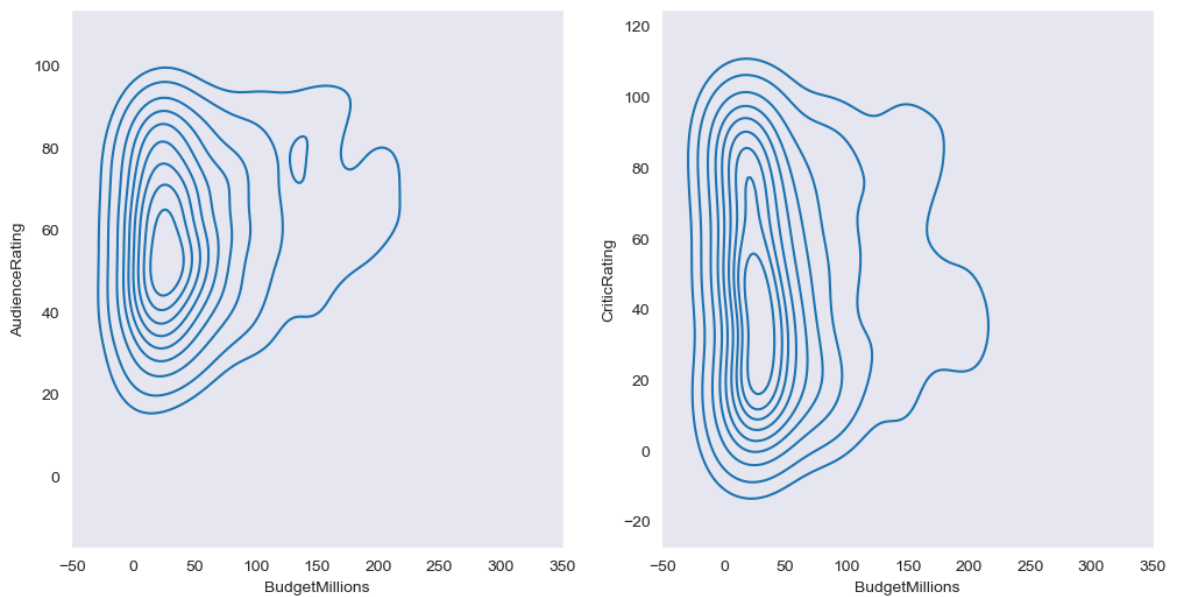


```
In [47]: f,ax=plt.subplots(1,2,figsize=(12,6))
plt.show()
```



```
In [48]: f, axes=plt.subplots(1,2,figsize=(12,6))

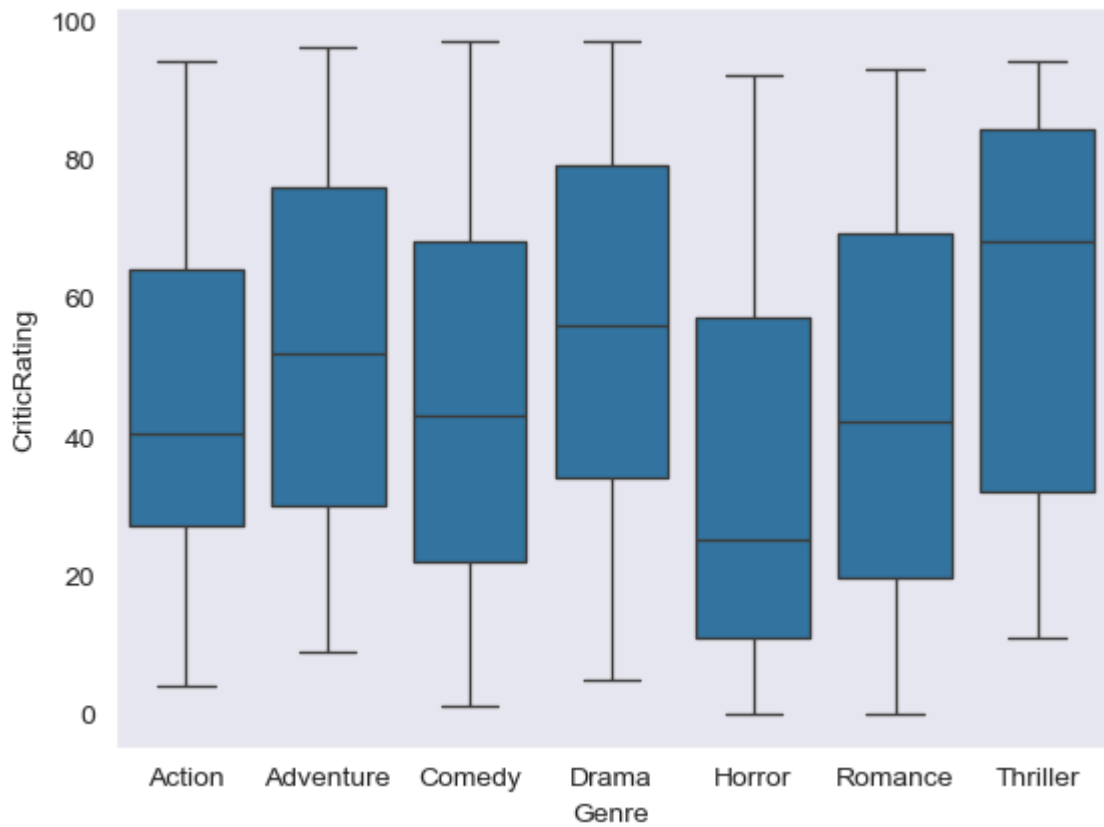
k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,ax=axes[0])
k2=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating,ax=axes[1])
plt.show()
```



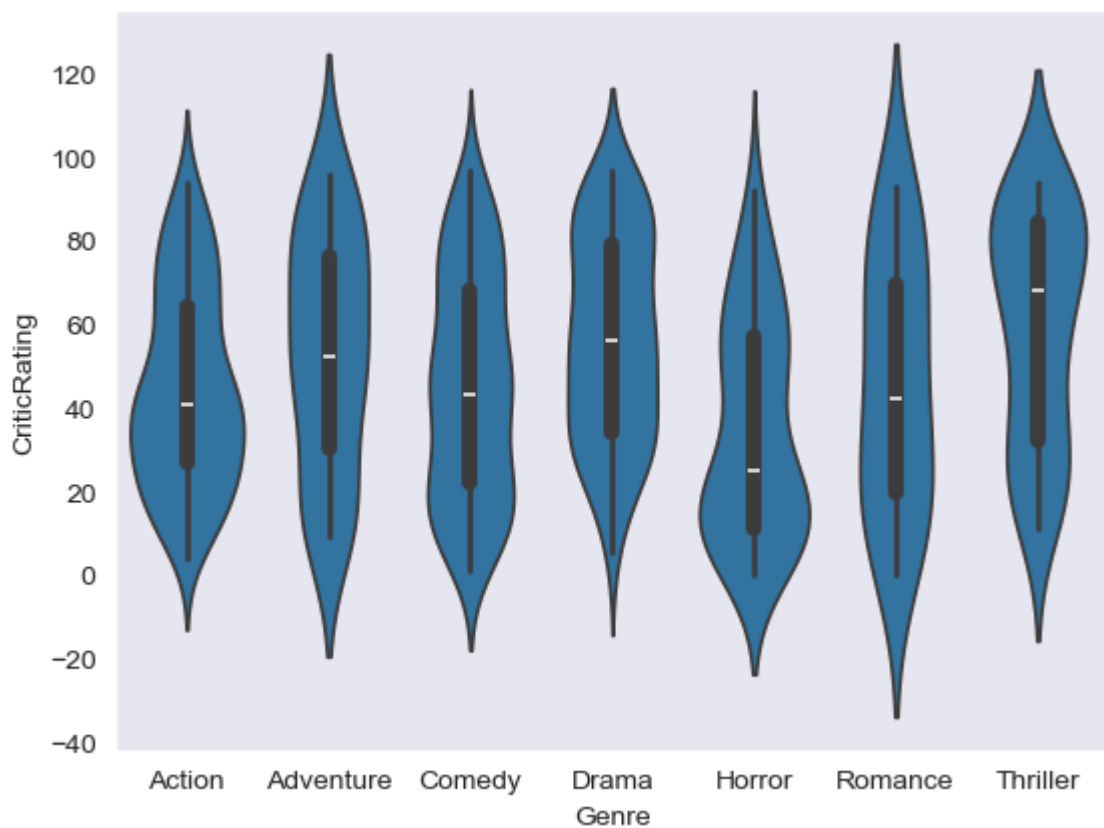
```
In [49]: axes
```

```
Out[49]: array([<Axes: xlabel='BudgetMillions', ylabel='AudienceRating'>,
                <Axes: xlabel='BudgetMillions', ylabel='CriticRating'>],
              dtype=object)
```

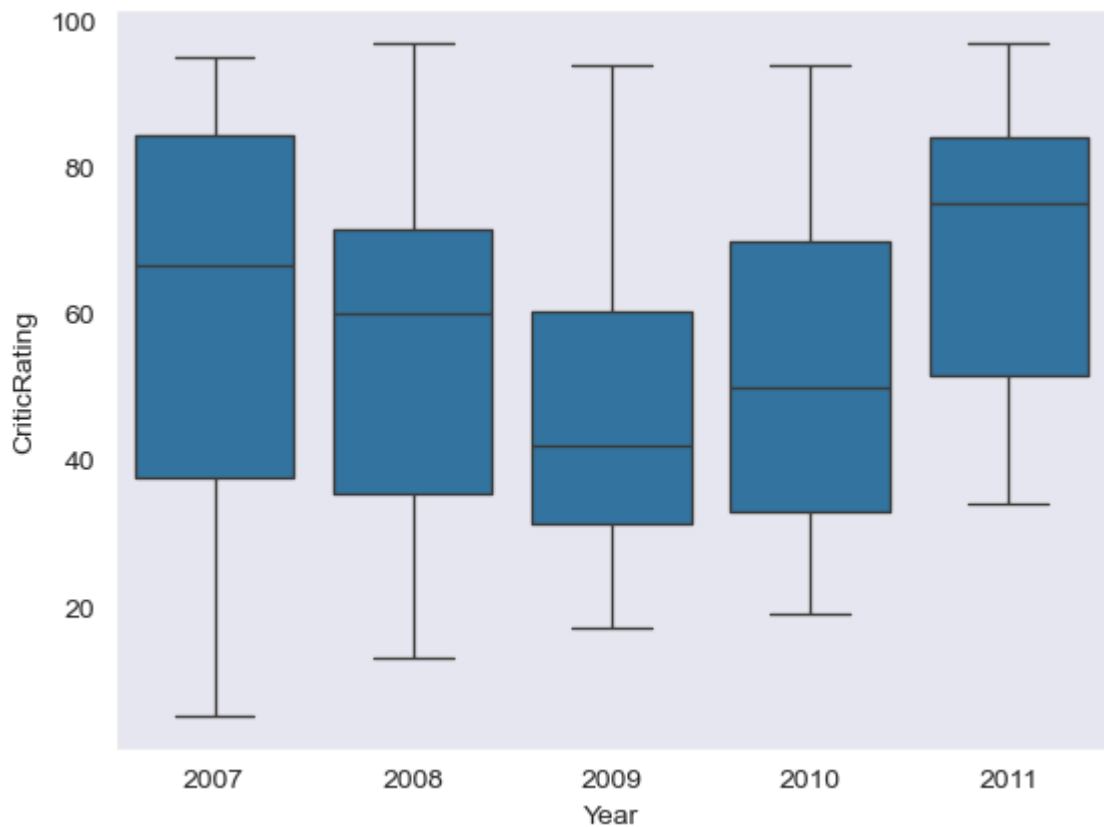
```
In [50]: w=sns.boxplot(data=movies,x='Genre',y='CriticRating')
plt.show()
```



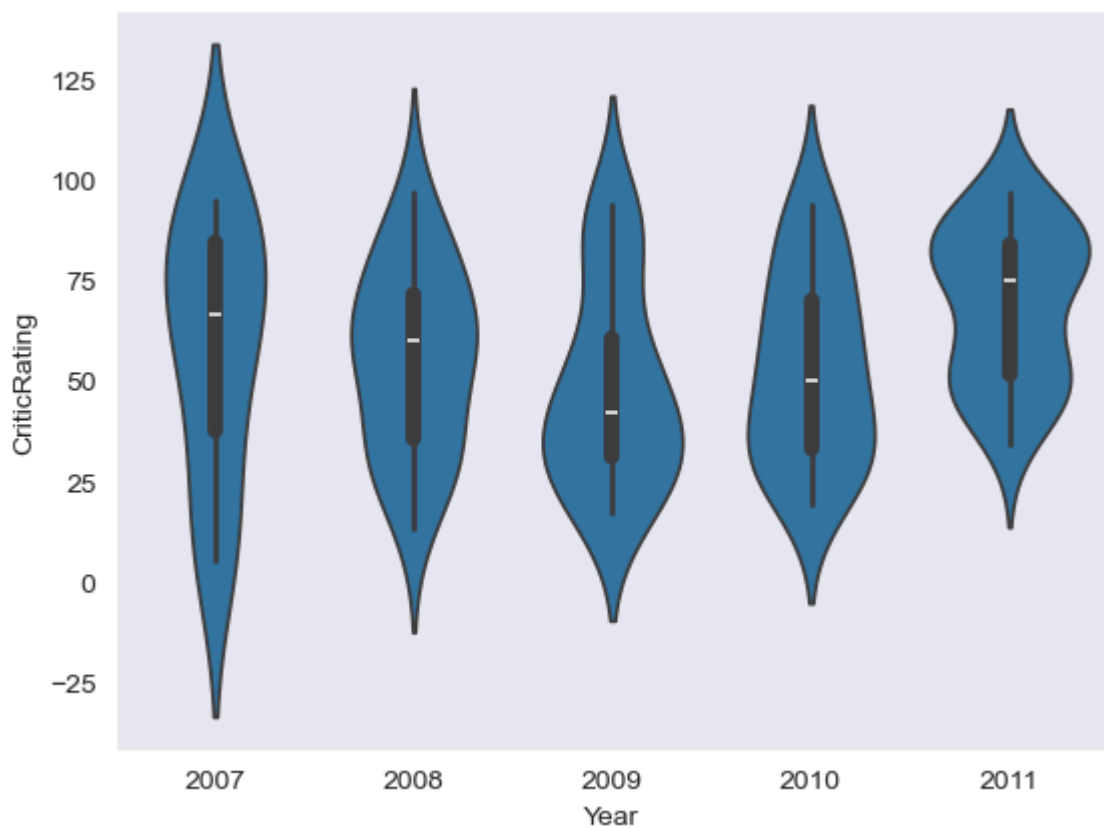
```
In [51]: z=sns.violinplot(data=movies,x='Genre',y='CriticRating')
plt.show()
```



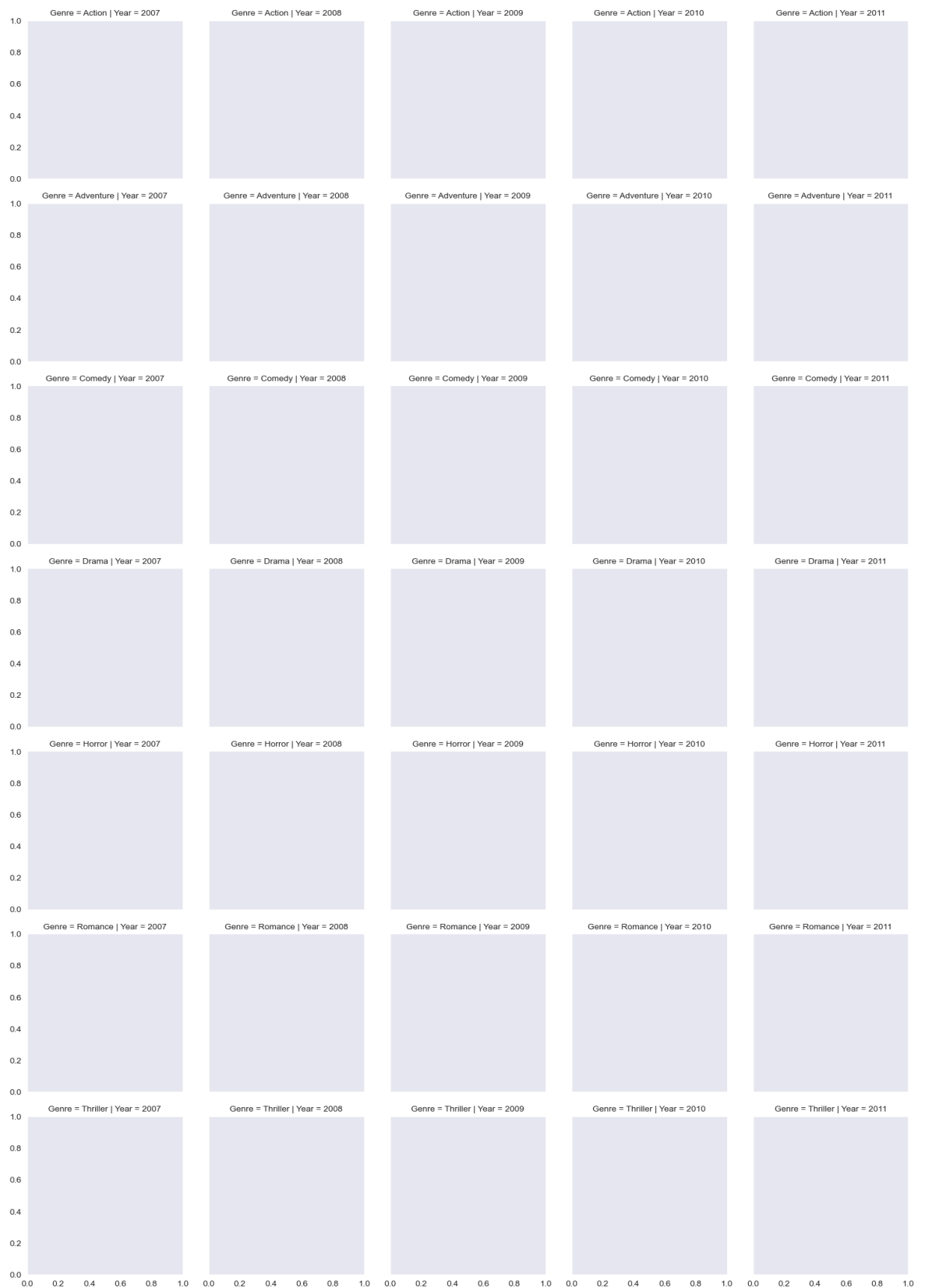
```
In [52]: w1=sns.boxplot(data=movies[movies.Genre=='Drama'],x='Year',y='CriticRating')
plt.show()
```



```
In [53]: z=sns.violinplot(data=movies[movies.Genre=='Drama'],x='Year',y='CriticRating')
plt.show()
```

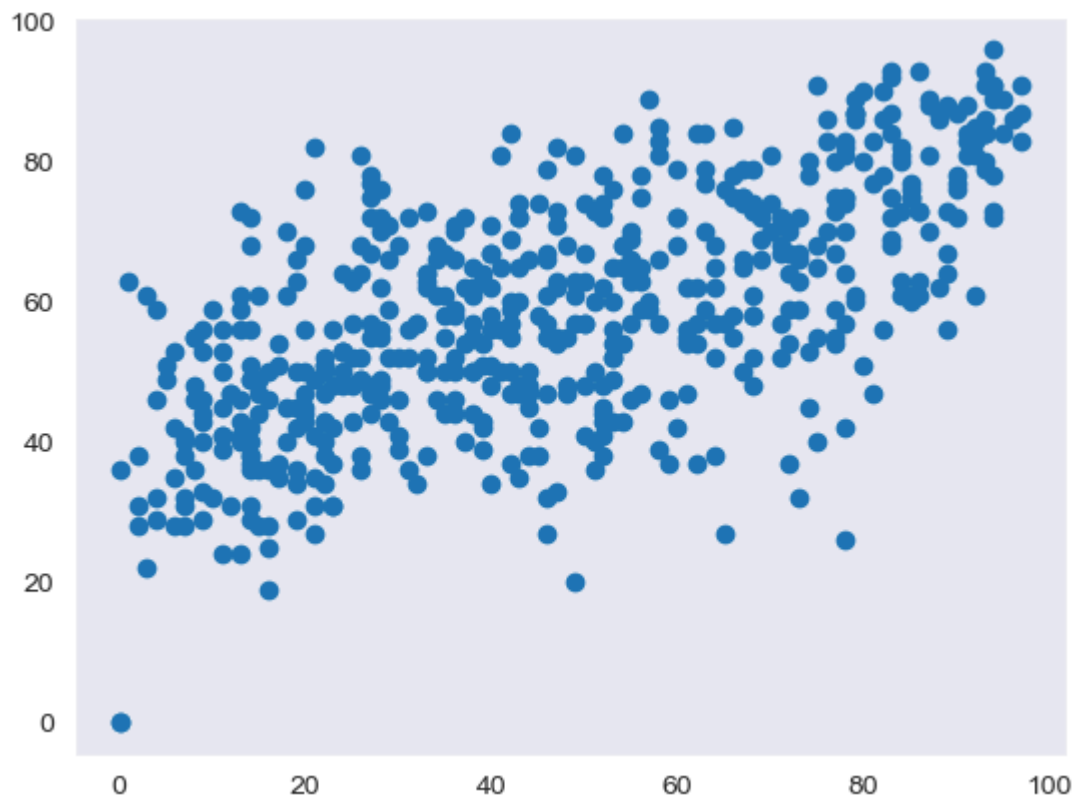


```
In [54]: # Creating a FacetGrid
g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
plt.show()
```

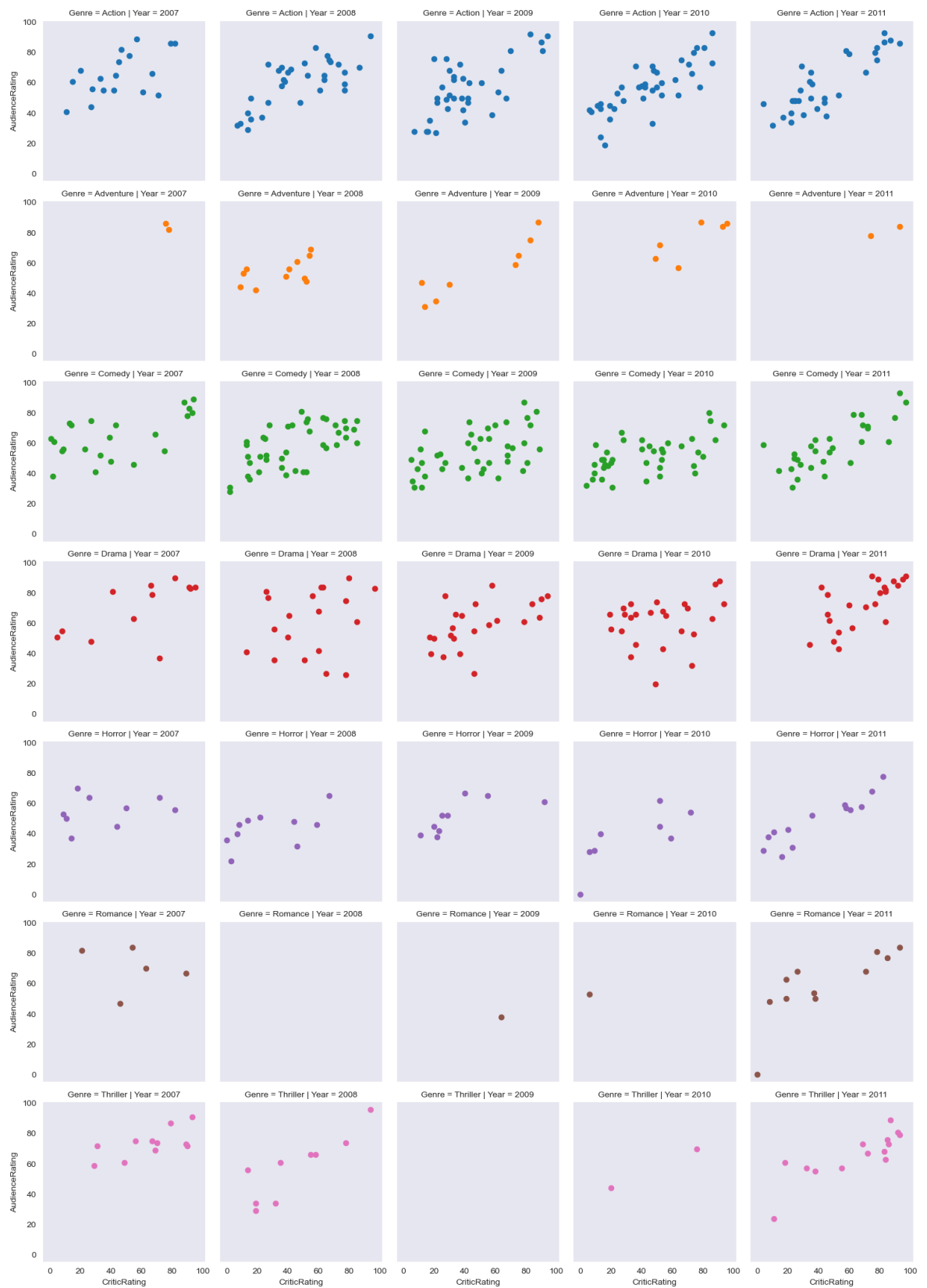


```
In [55]: plt.scatter(movies.CriticRating,movies.AudienceRating)
plt.show()
```





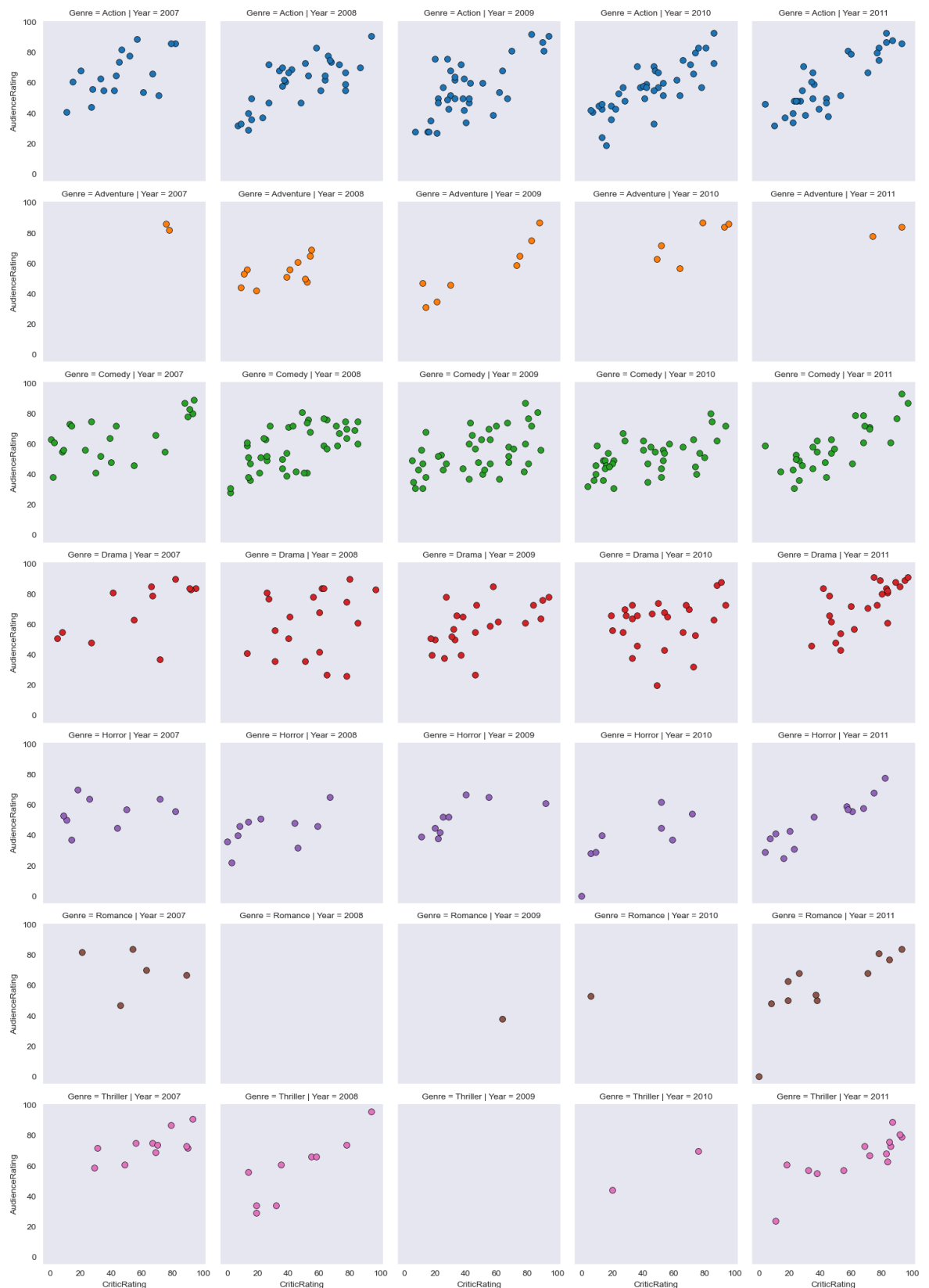
```
In [56]: g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
g=g.map(plt.scatter,'CriticRating','AudienceRating')
plt.show()
```



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



```
In [58]: 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()
```



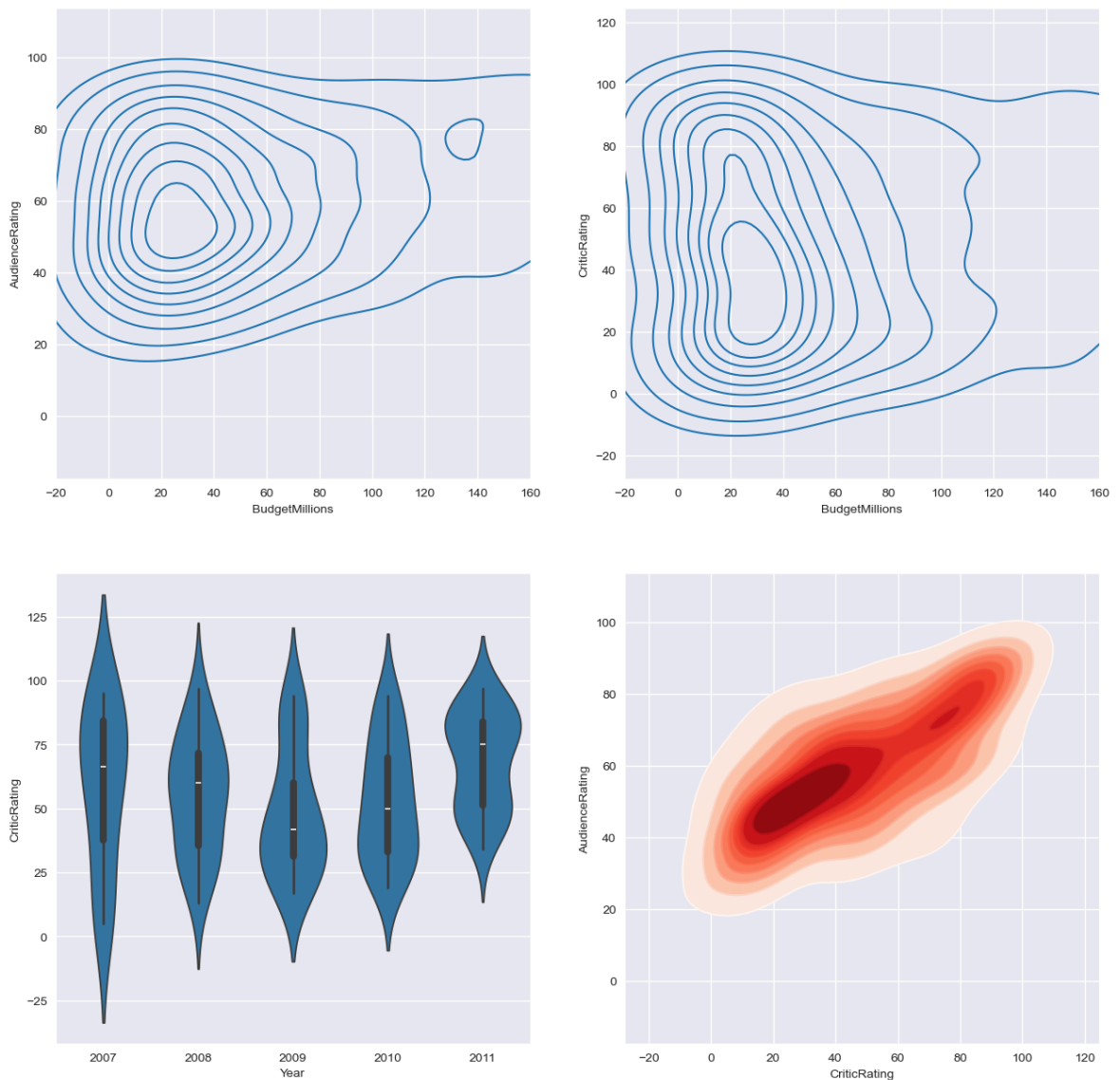
```
In [59]: sns.set_style('darkgrid')
f,axes=plt.subplots(2,2,figsize=(15,15))

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
```

```
k4=sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,shade=True,shade_lo
k4b=sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,cmap='Reds',ax=ax
plt.show()
```



```
In [60]: sns.set_style('dark',{'axes.facecolor':'black'})
f,axes=plt.subplots(2,2,figsize=(15,15))

k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,shade=True,shade_
k1b=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,cmap='cool',ax=a
k2=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating,shade=True,shade_lo
k2b=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating,cmap='cool',ax=ax
z=sns.violinplot(data=movies[movies.Genre=='Drama'],x='Year',y='CriticRating',ax
k4=sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,shade=True,shade_lo
k4b=sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,cmap='gist_gray_r'
k1.set(xlim=(-20,160))
```

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

```
plt.show()
```

