

```
In [82]: import pandas as pd
```

```
In [83]: movies = pd.read_csv(r'C:\Users\lenovo\Desktop\Movie-Rating\Movie-Rating.csv')
```

```
In [84]: movies
```

```
Out[84]:
```

	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 [85]: type(movies)
```

```
Out[85]: pandas.core.frame.DataFrame
```

```
In [86]: len(movies)
```

```
Out[86]: 559
```

```
In [87]: import numpy
print(numpy.__version__)
```

1.26.4

```
In [88]: import pandas
print(pandas.__version__)
```

2.2.2

```
In [89]: movies.columns
```

```
Out[89]: Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
               'Budget (million $)', 'Year of release'],
              dtype='object')
```

In [90]: `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 [91]: `movies.shape`

Out[91]: (559, 6)

In [92]: `movies.head()`

Out[92]:

	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 [93]: `movies.tail()`

Out[93]:

	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 [94]: `movies.columns`

```
Out[94]: Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
              'Budget (million $)', 'Year of release'],
              dtype='object')
```

```
In [95]: movies.columns = ['Film', 'Genre', 'CriticRating', 'AudienceRating', 'BudgetMill
```

```
In [96]: movies.head()
```

```
Out[96]:
```

	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 [97]: movies.describe()
```

```
Out[97]:
```

	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 [98]: 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 [99]: movies.Film = movies.Film.astype('category')
movies.Genre = movies.Genre.astype('category')
movies.Year = movies.Year.astype('category')
```

In [100...

movies.Film

Out[100...

```
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 [101...

movies.Genre

Out[101...

```
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 [102...

movies.Year

Out[102...

```
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 [103...

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 [104... `movies.describe()`

Out[104... 

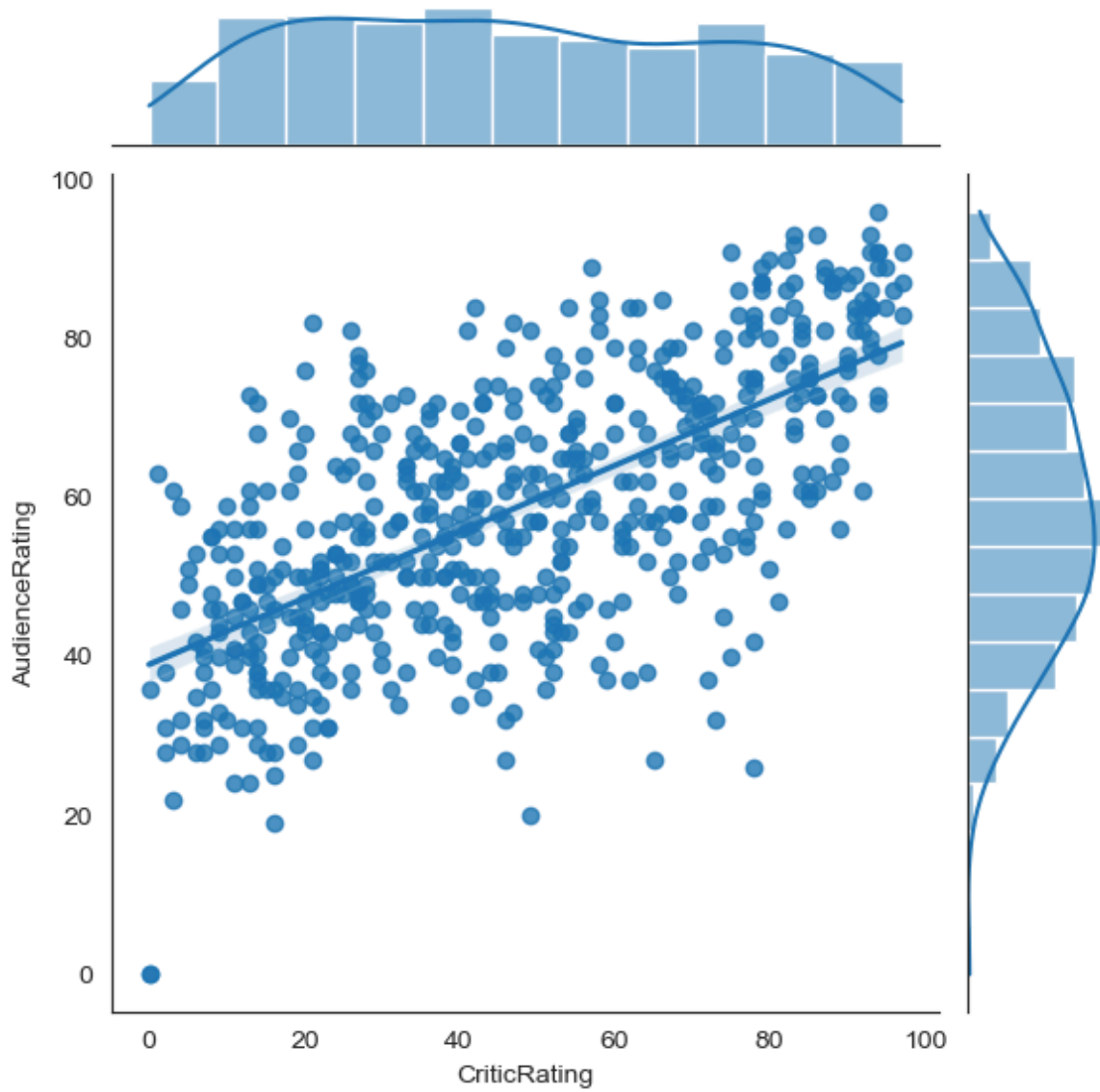
	CriticRating	AudienceRating	BudgetMillions
count	559.000000	559.000000	559.000000
mean	47.309481	58.744186	50.236136
std	26.413091	16.826887	48.731817
min	0.000000	0.000000	0.000000
25%	25.000000	47.000000	20.000000
50%	46.000000	58.000000	35.000000
75%	70.000000	72.000000	65.000000
max	97.000000	96.000000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

In [105... 

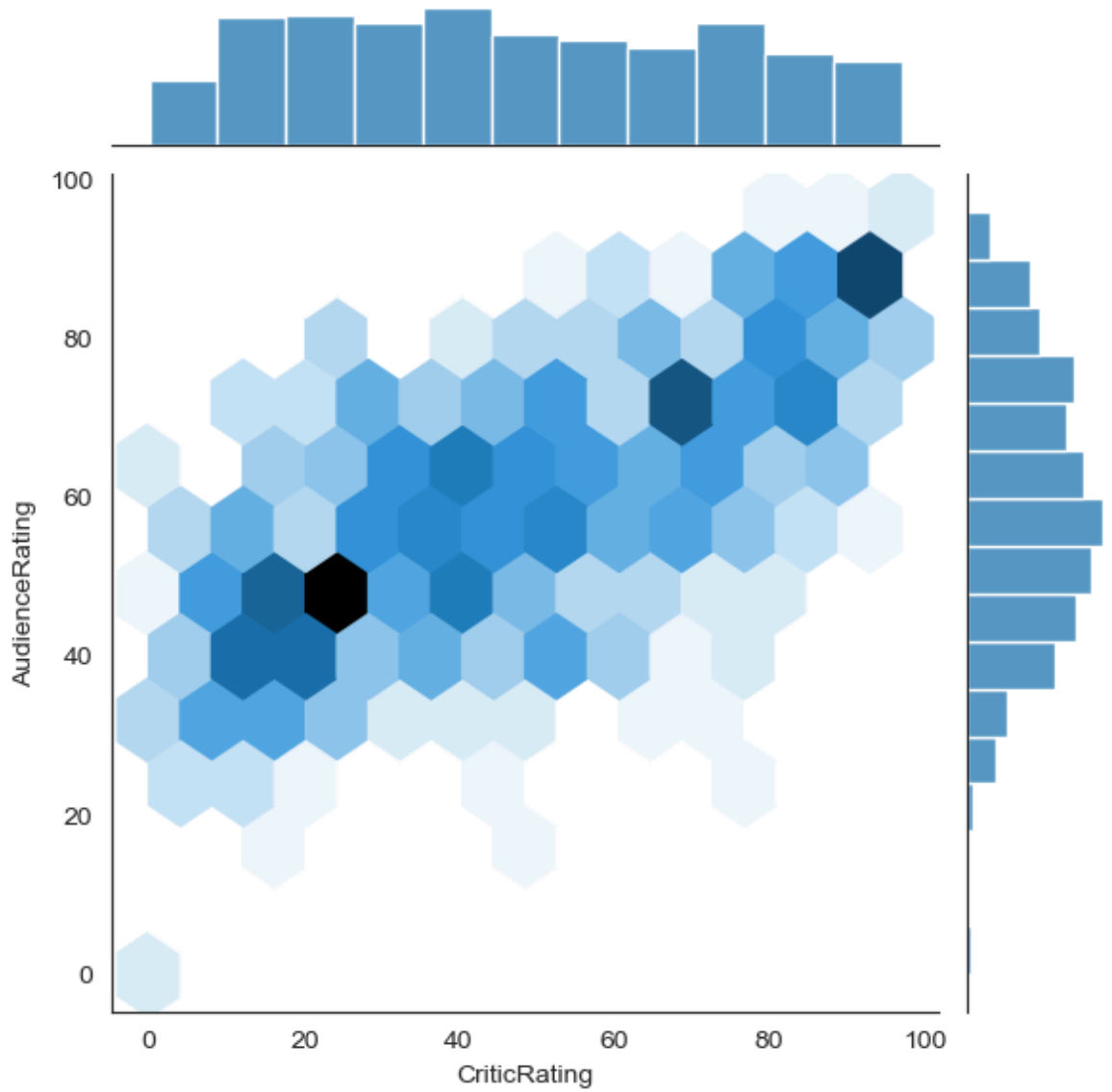
```
from matplotlib import pyplot as plt  
import seaborn as sns  
%matplotlib inline  
import warnings  
warnings.filterwarnings('ignore')
```

In [106... 

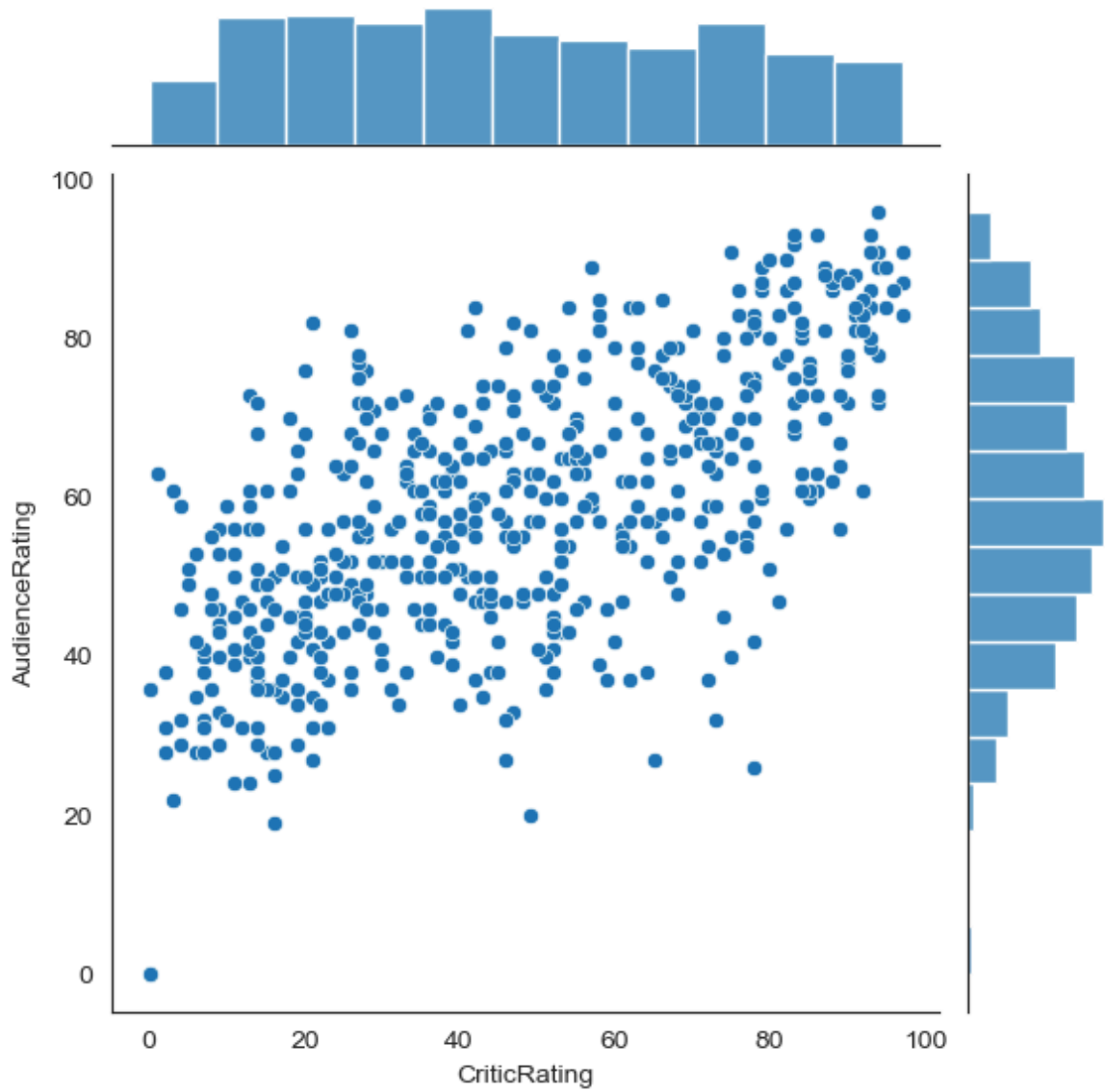
```
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating', kind
```



```
In [107... j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating', kind
```

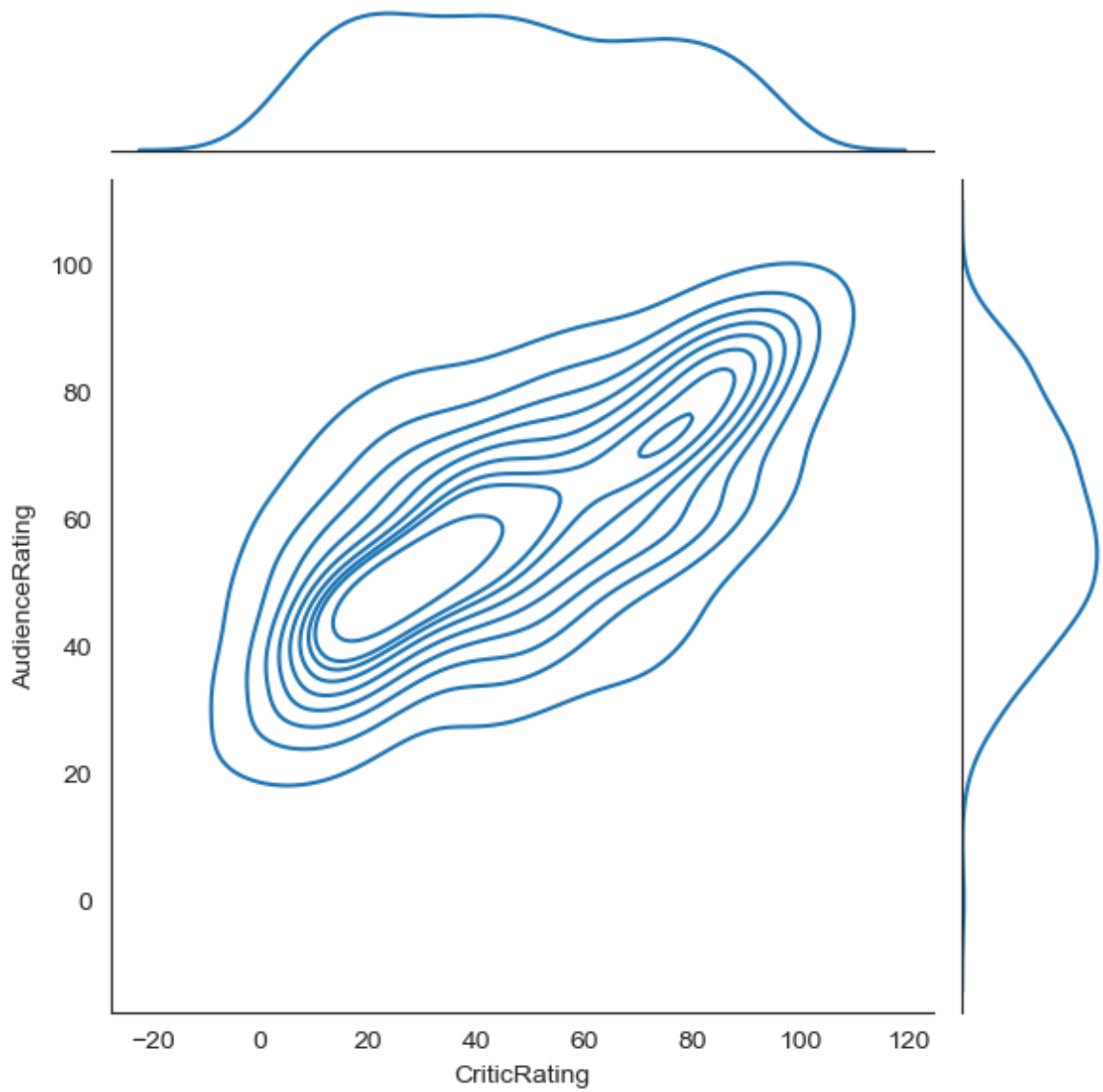


```
In [108... j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating', kind
```

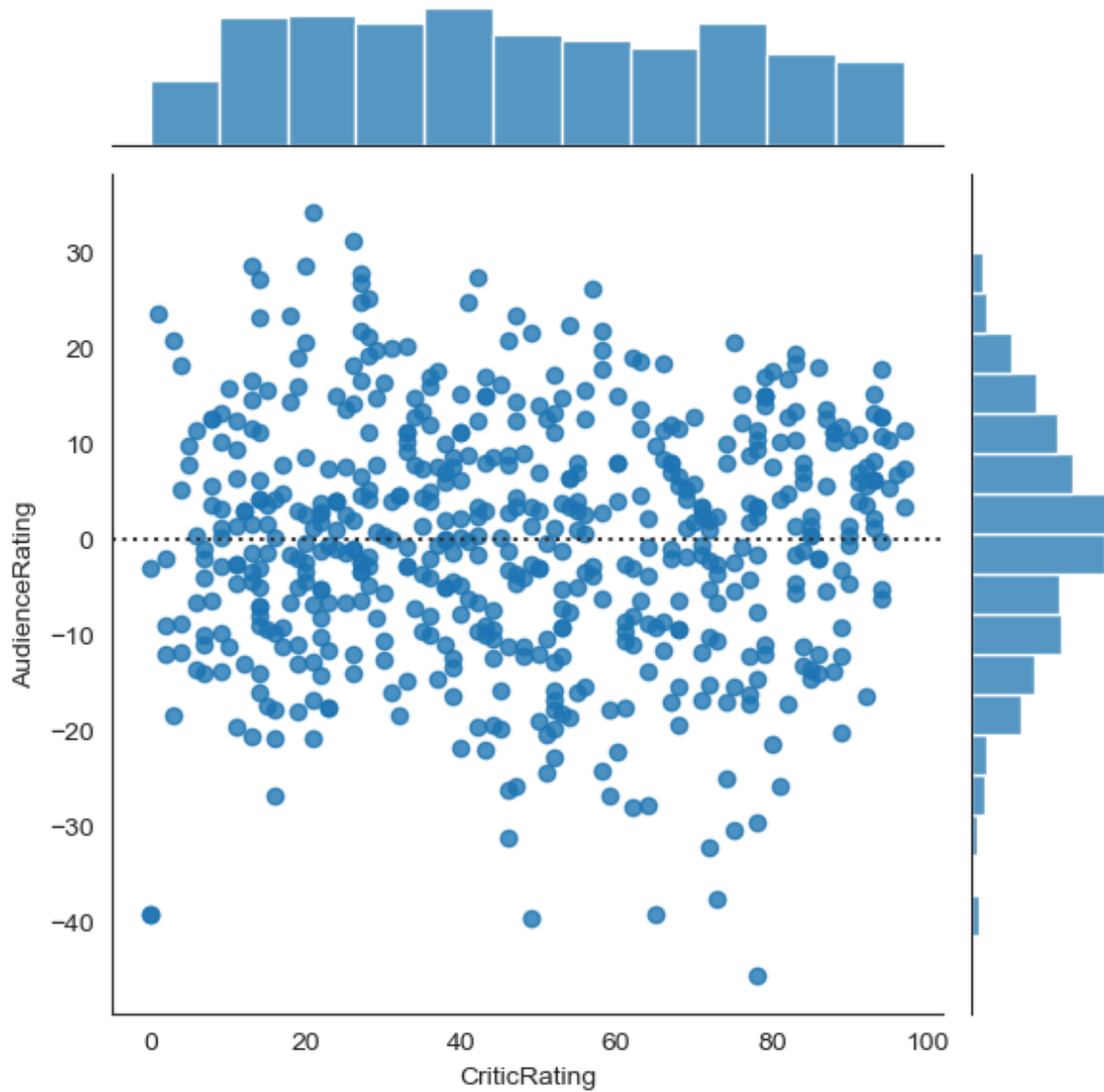


```
In [109... j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating', kind
```



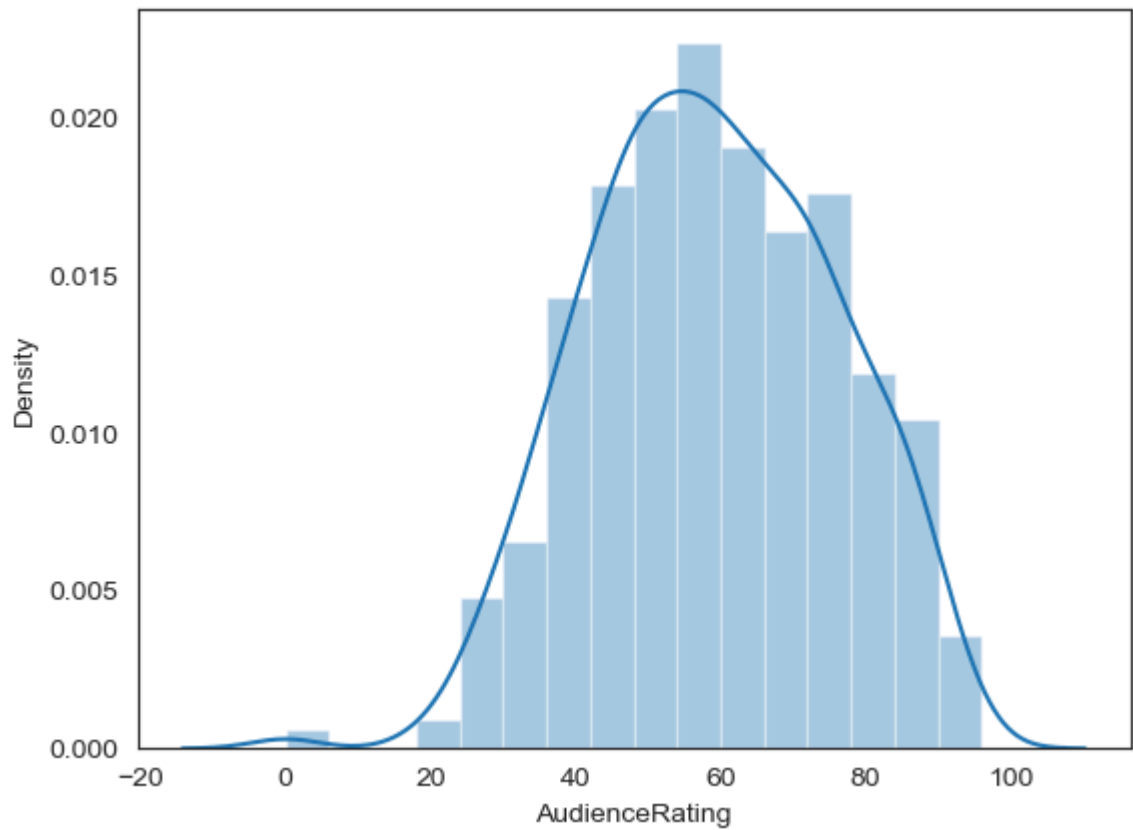


```
In [110... j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating', kind
```



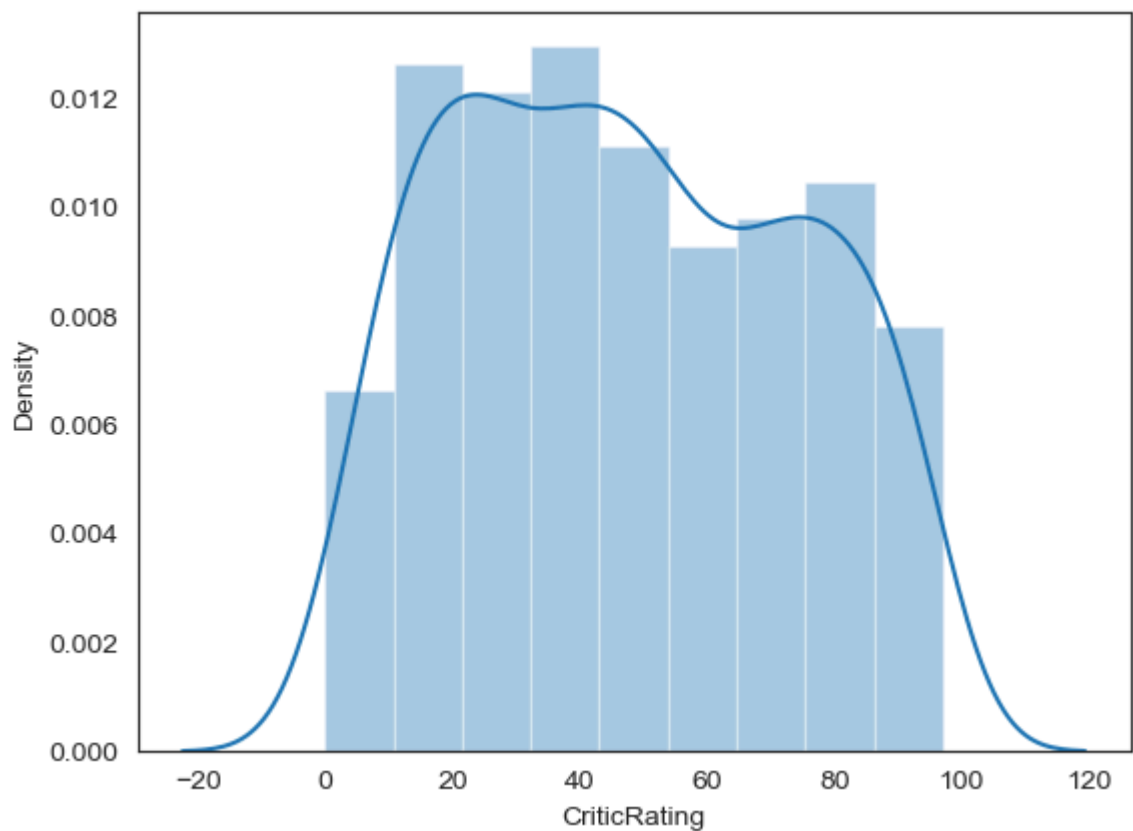
```
In [111... #Insights for the above graphs:  
#Audience rating is more dominant than critics rating  
#Based on these we found out that, most of the people are most likely to watch a  
#There is positive correlation between two attributes, that is critics rating an  
#If you see the Excel sheet, the audience rating and critics rating based on the
```

```
In [112... m1 = sns.distplot(movies.AudienceRating)
```



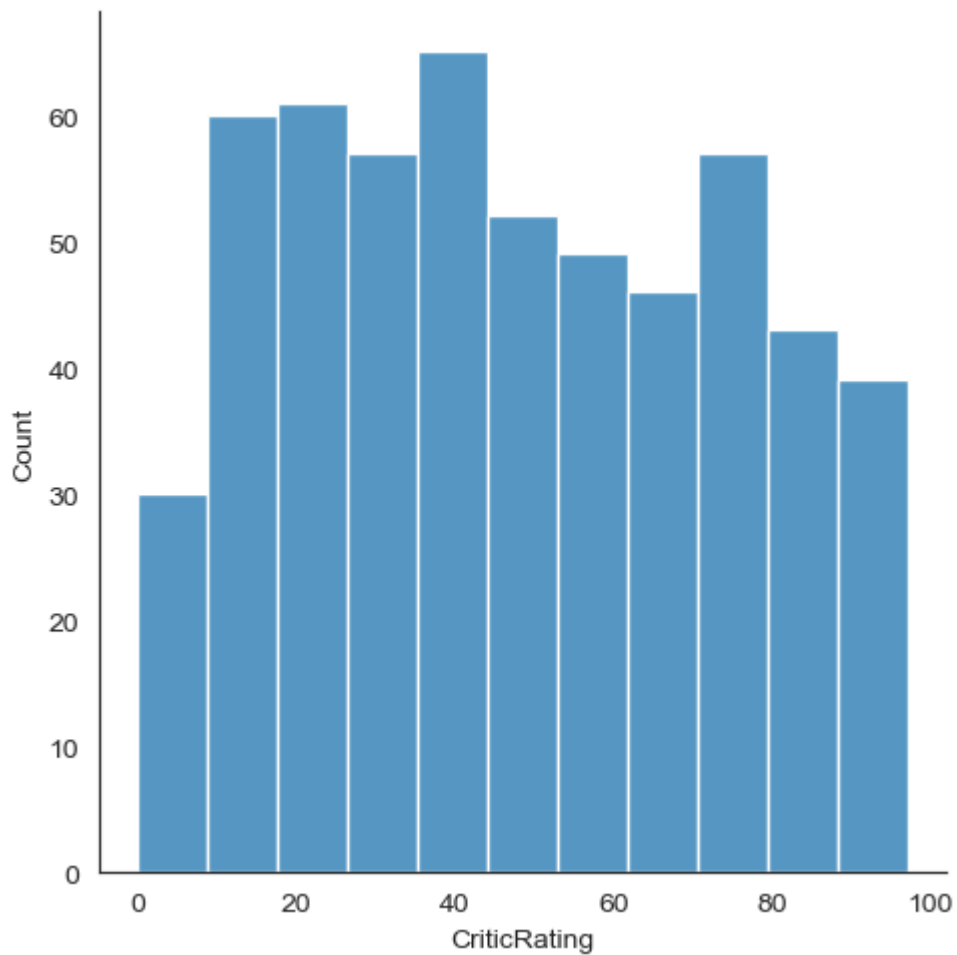
In [113...

```
m1 = sns.distplot(movies.CriticRating) #here we mention distplot  
#This graph is uniform distribution
```



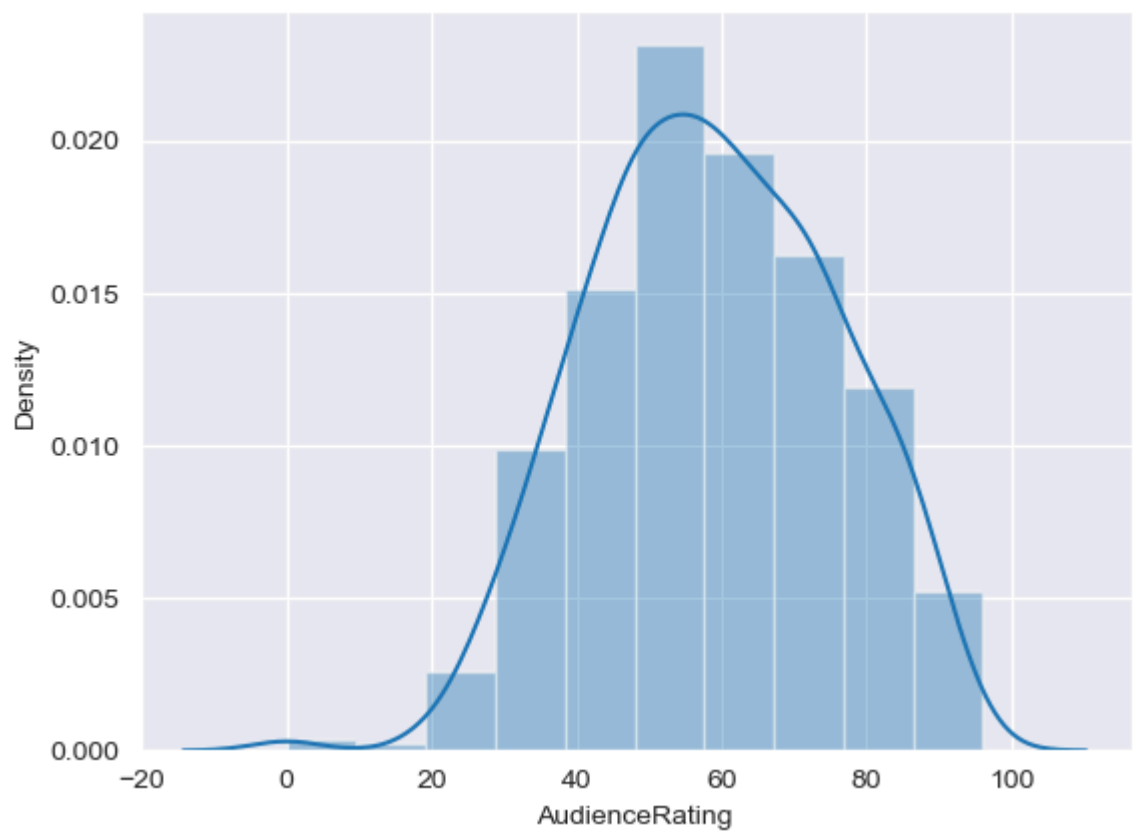
In [114...

```
m1 = sns.displot(movies.CriticRating) #here we mention displot #this is an histo
```

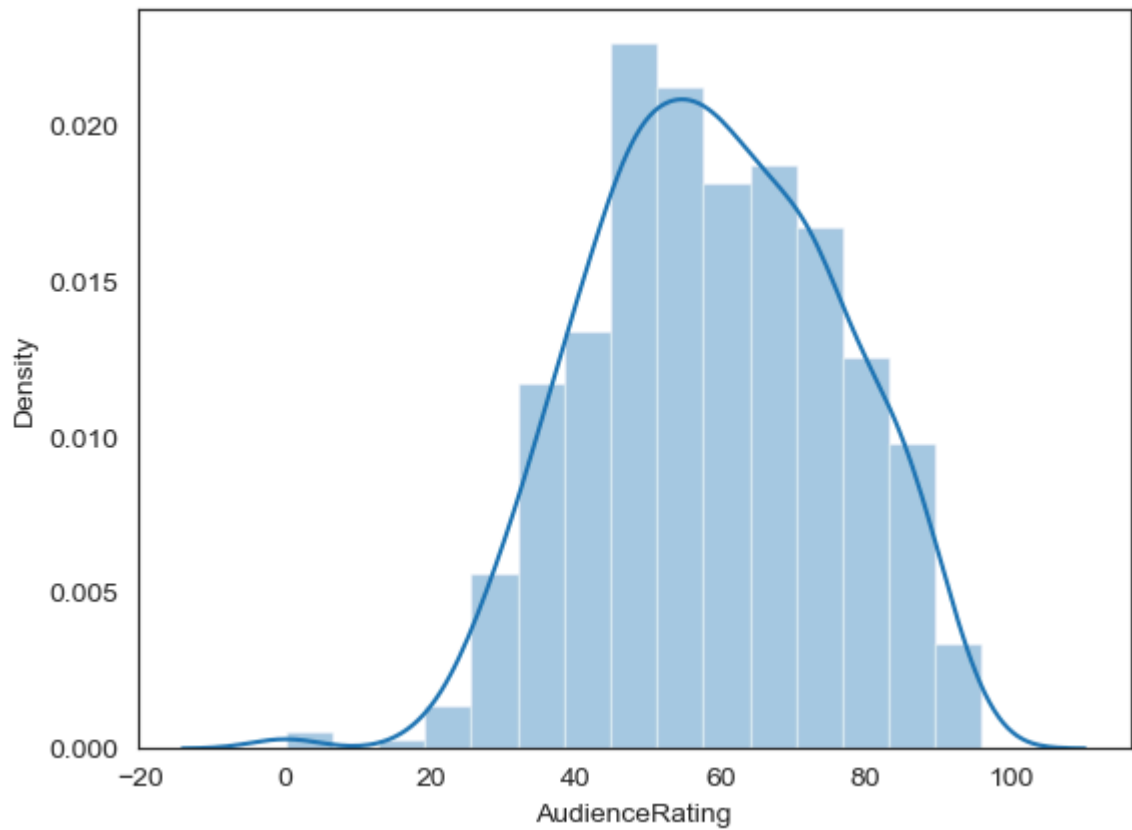


```
In [115...] sns.set_style('darkgrid')
```

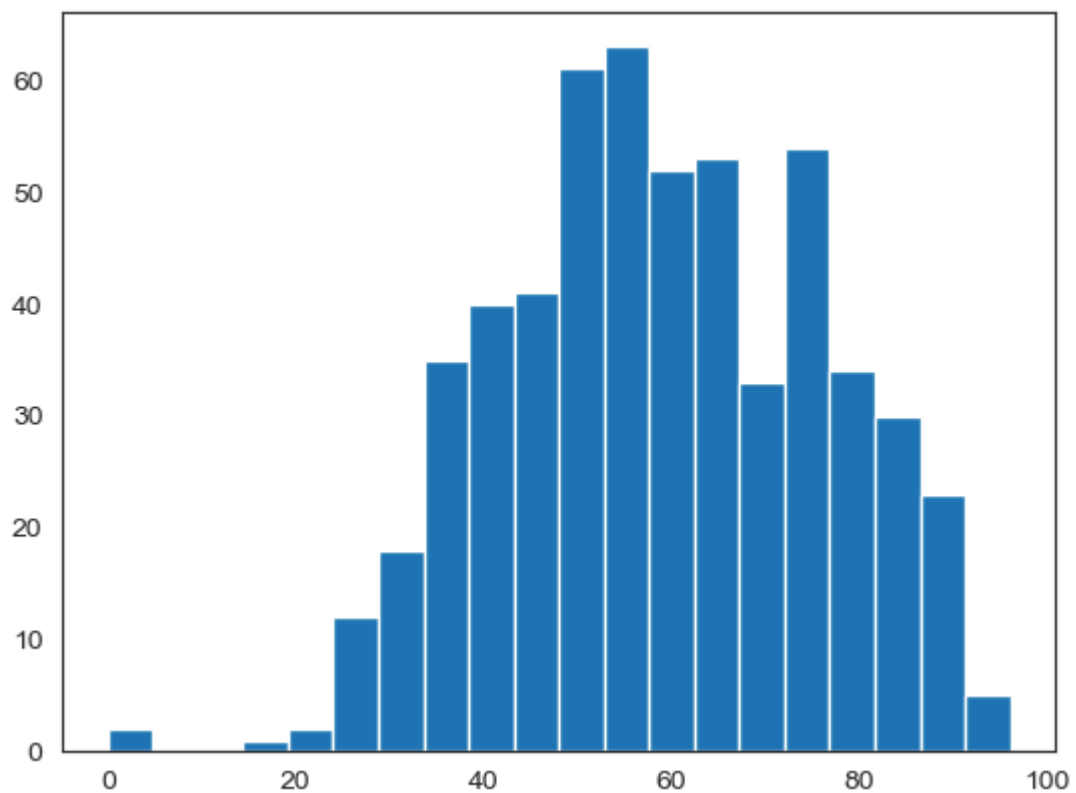
```
In [116...] m2 = sns.distplot(movies.AudienceRating, bins = 10)
```



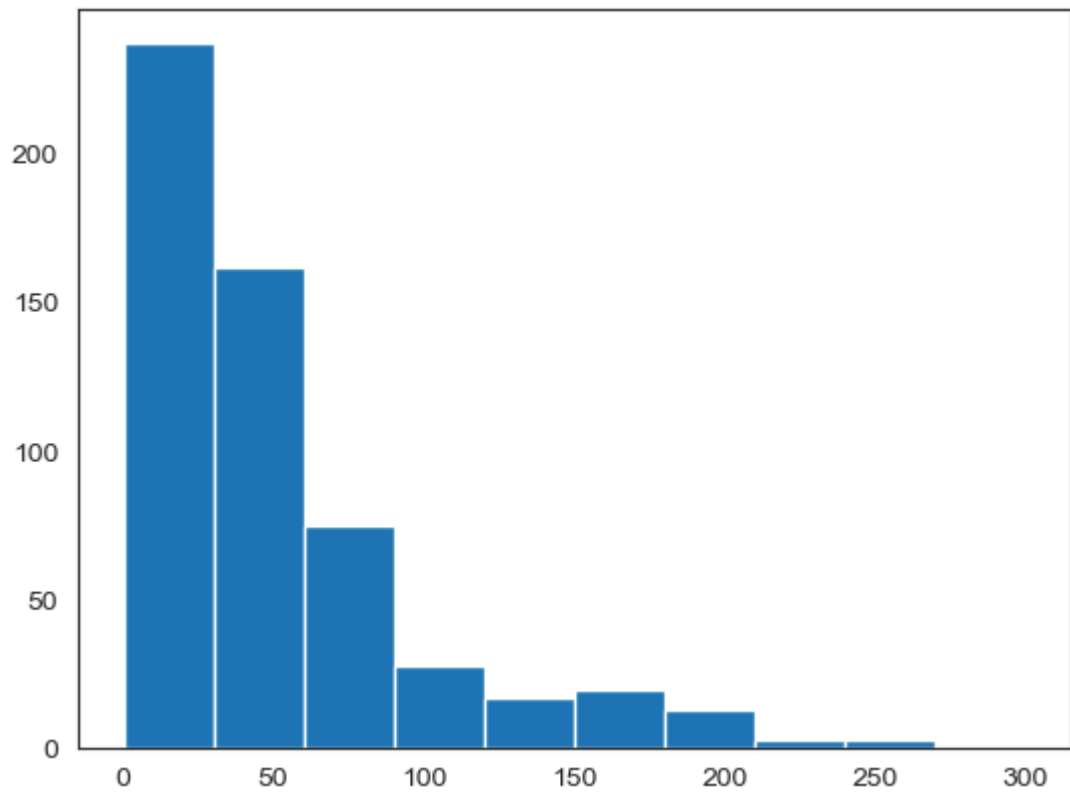
```
In [117... sns.set_style('white')  
m3 = sns.distplot(movies.AudienceRating, bins = 15)
```



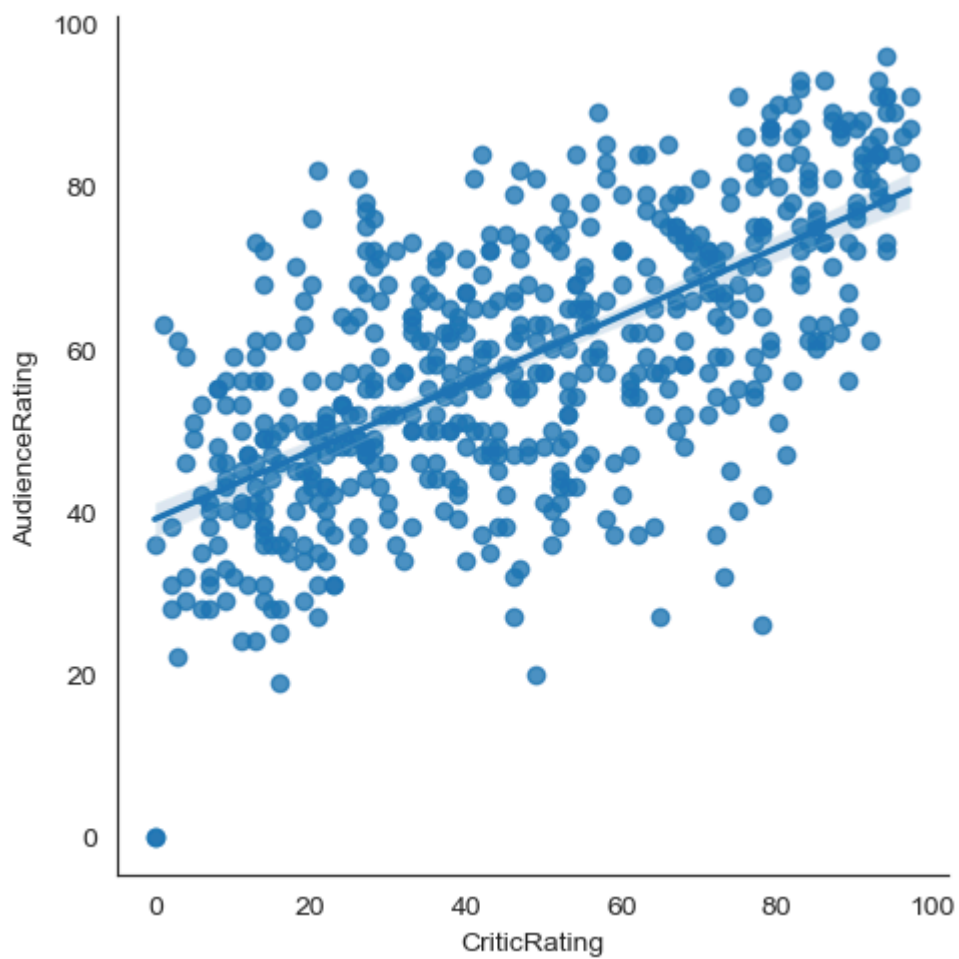
```
In [118... n1 = plt.hist(movies.AudienceRating, bins = 20)
```



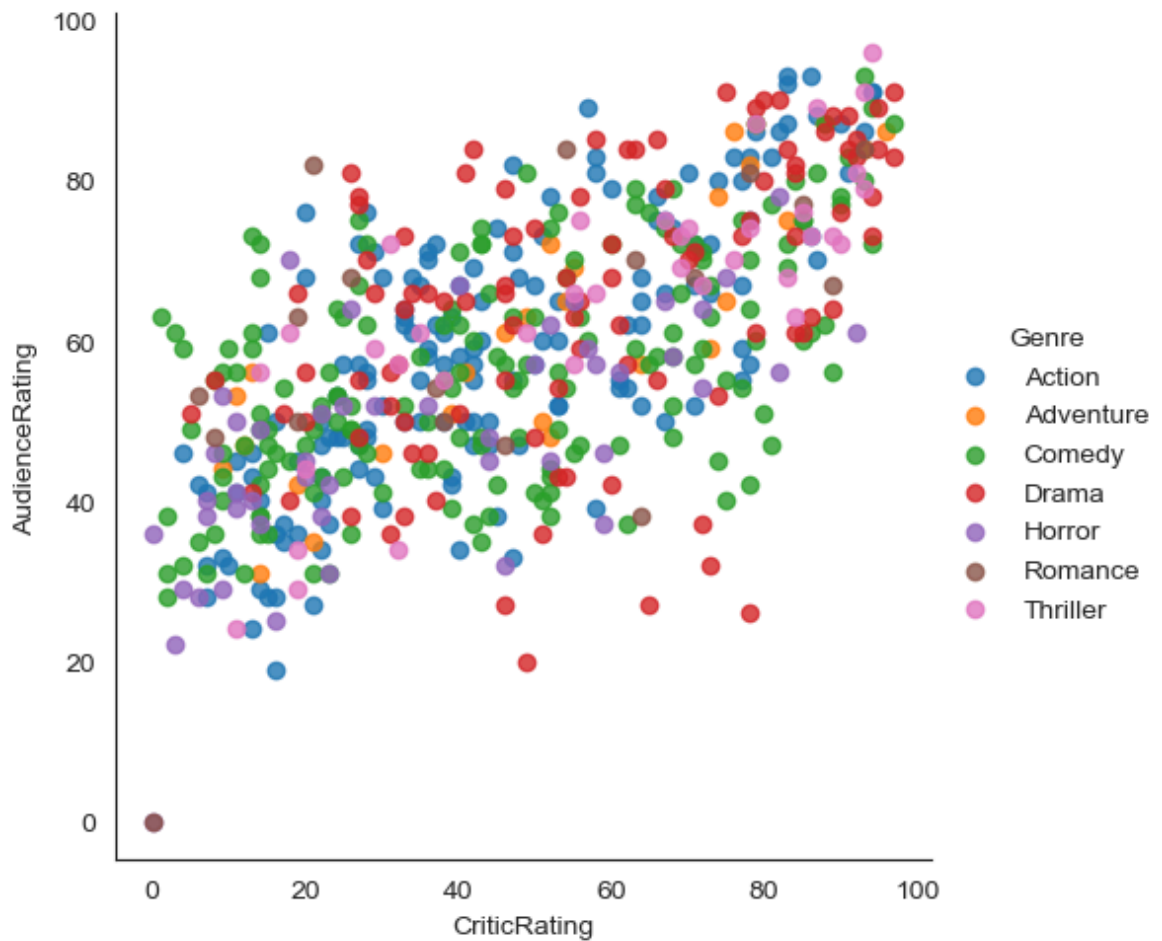
```
In [119... plt.hist(movies.BudgetMillions)  
plt.show()
```



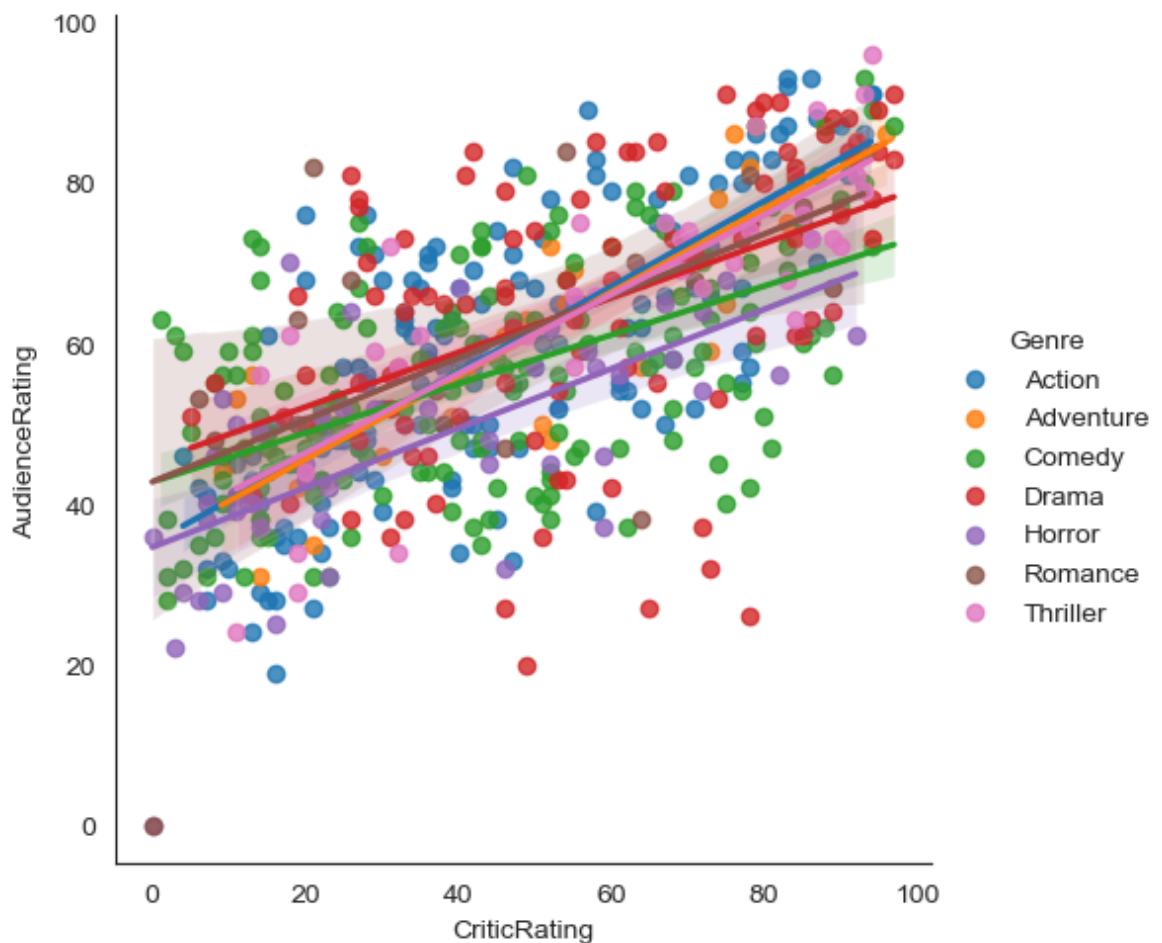
```
In [120...] vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating', fit_reg = T
```



```
In [121...] vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating', fit_reg = F
```

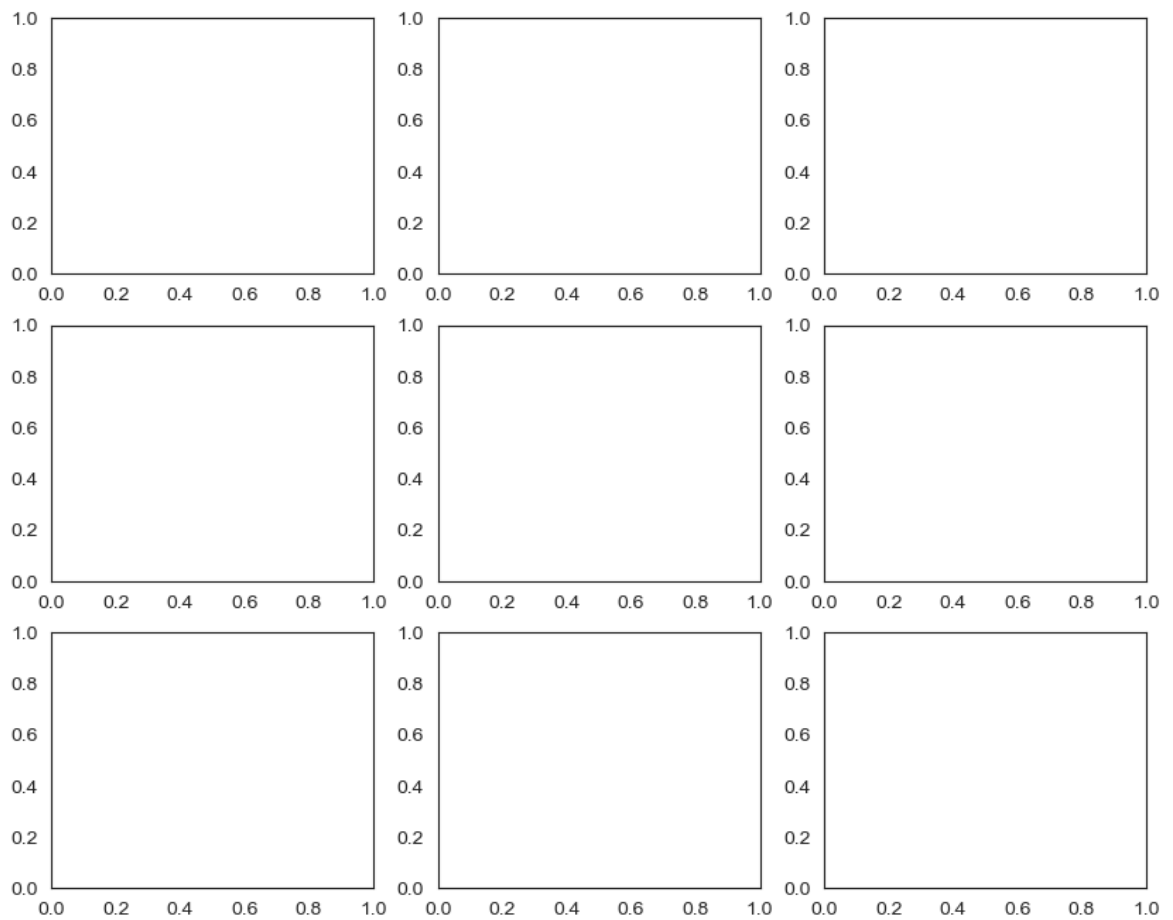


```
In [122... vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating', fit_reg = T
```



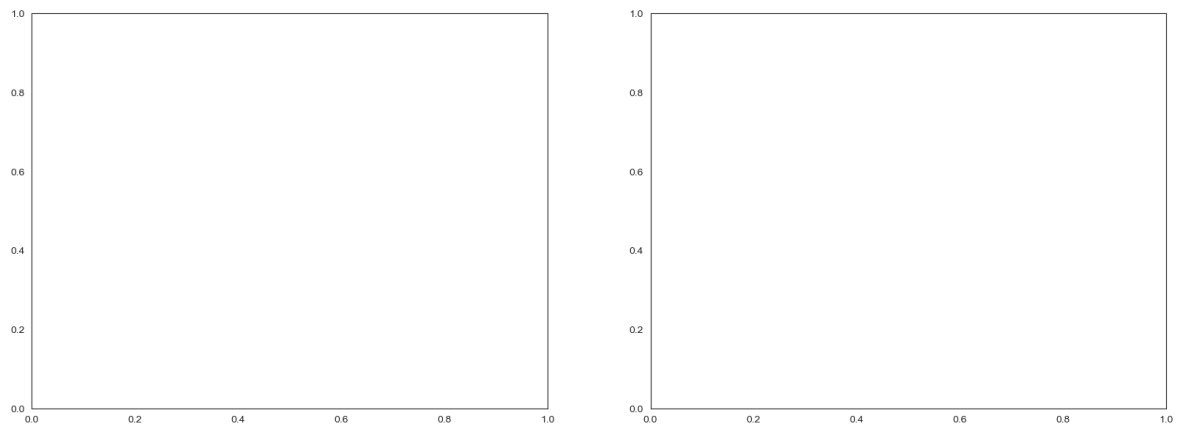
In [123...

```
ax = plt.subplots(3,3, figsize = (10,8))
```



In [124...

```
f, axes = plt.subplots(1,2, figsize = (20,7))
```



In [125...

```
movies
```



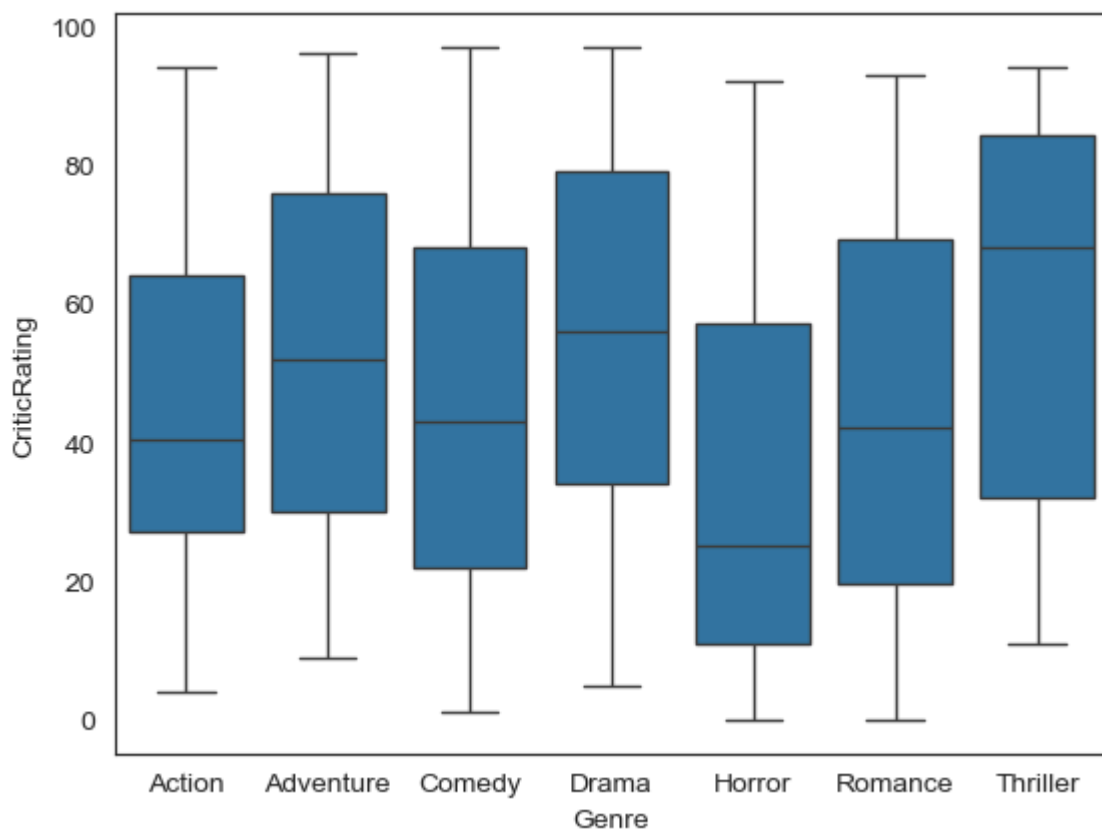
Out[125...

	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

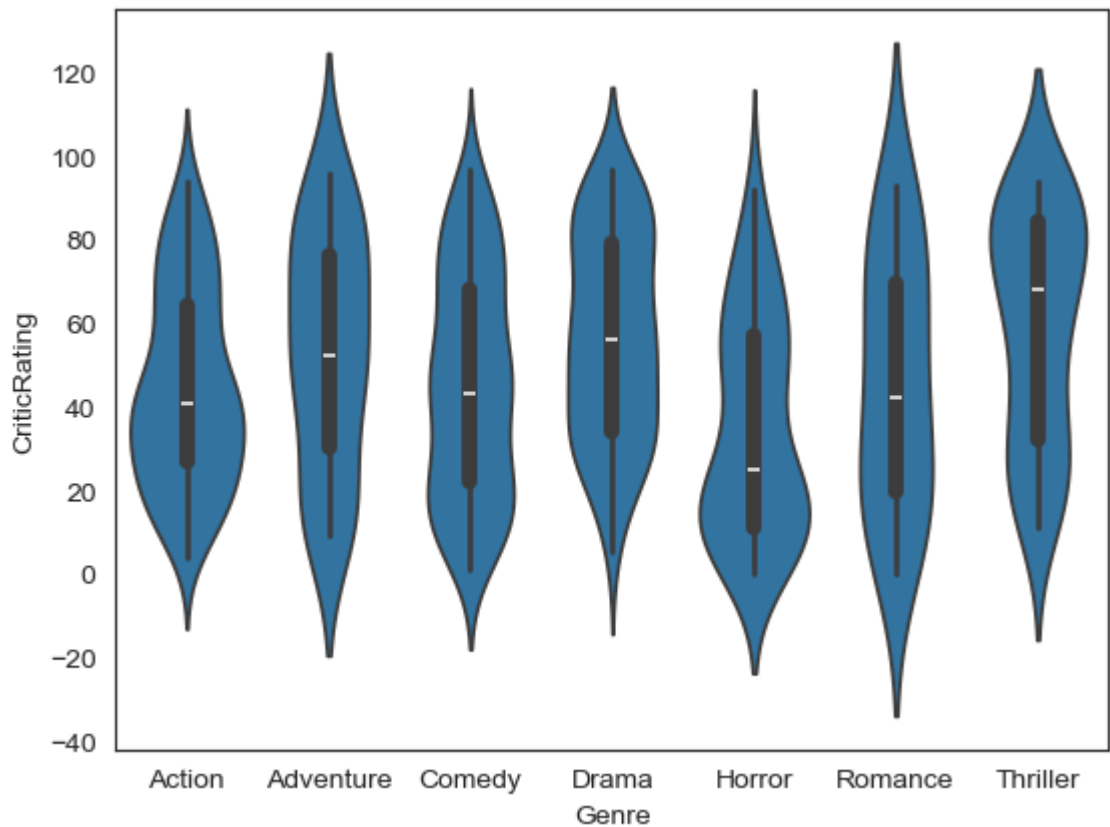
In [126...

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



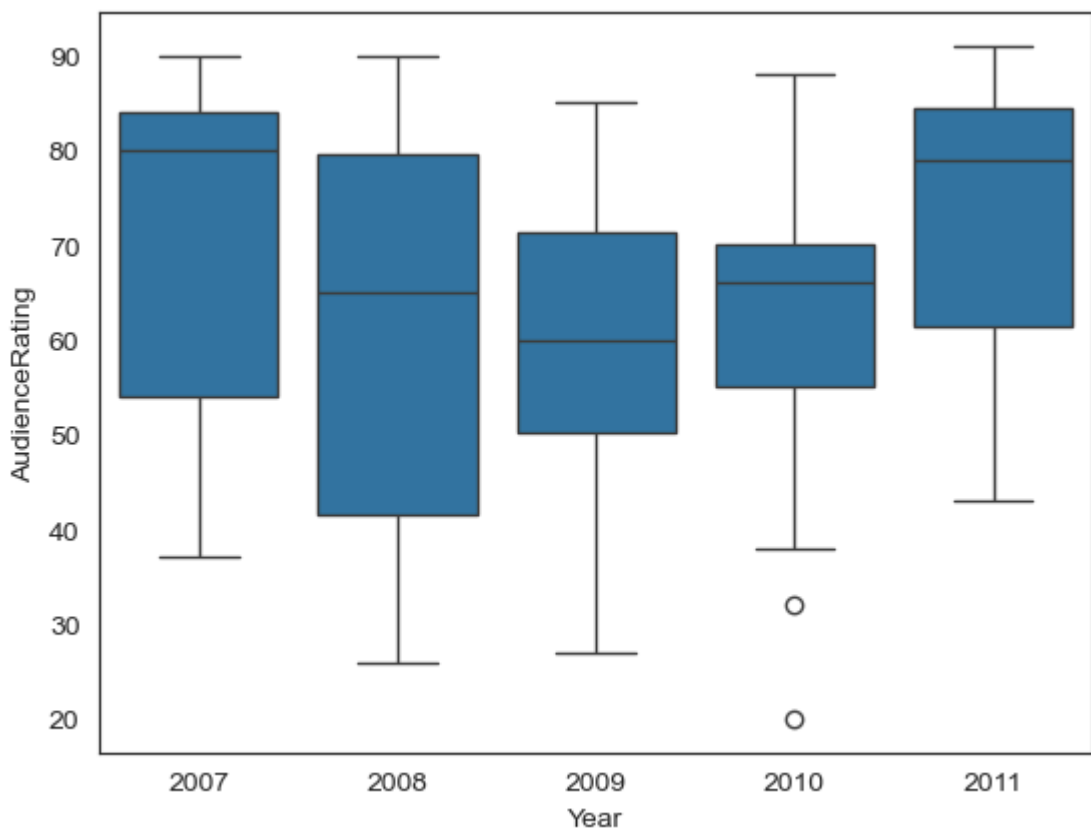
In [127...

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



In [128...

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



In [129...

```
movies.head()
```

Out[129...

	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

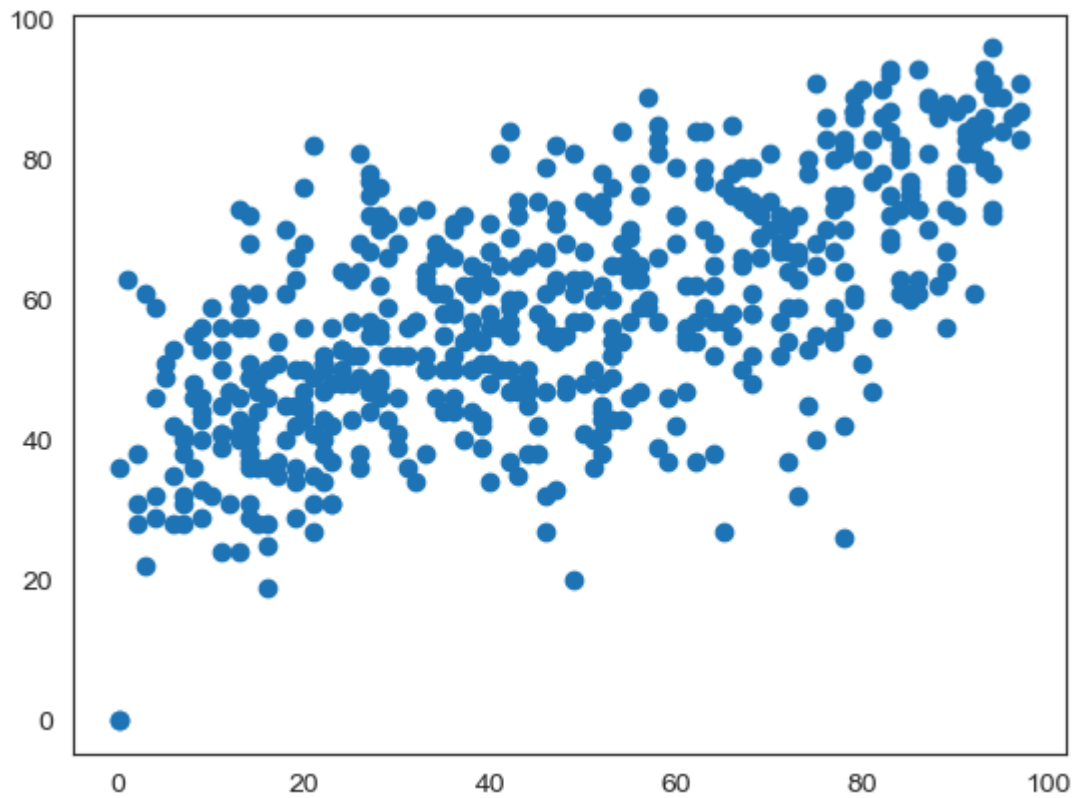
In [130...

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

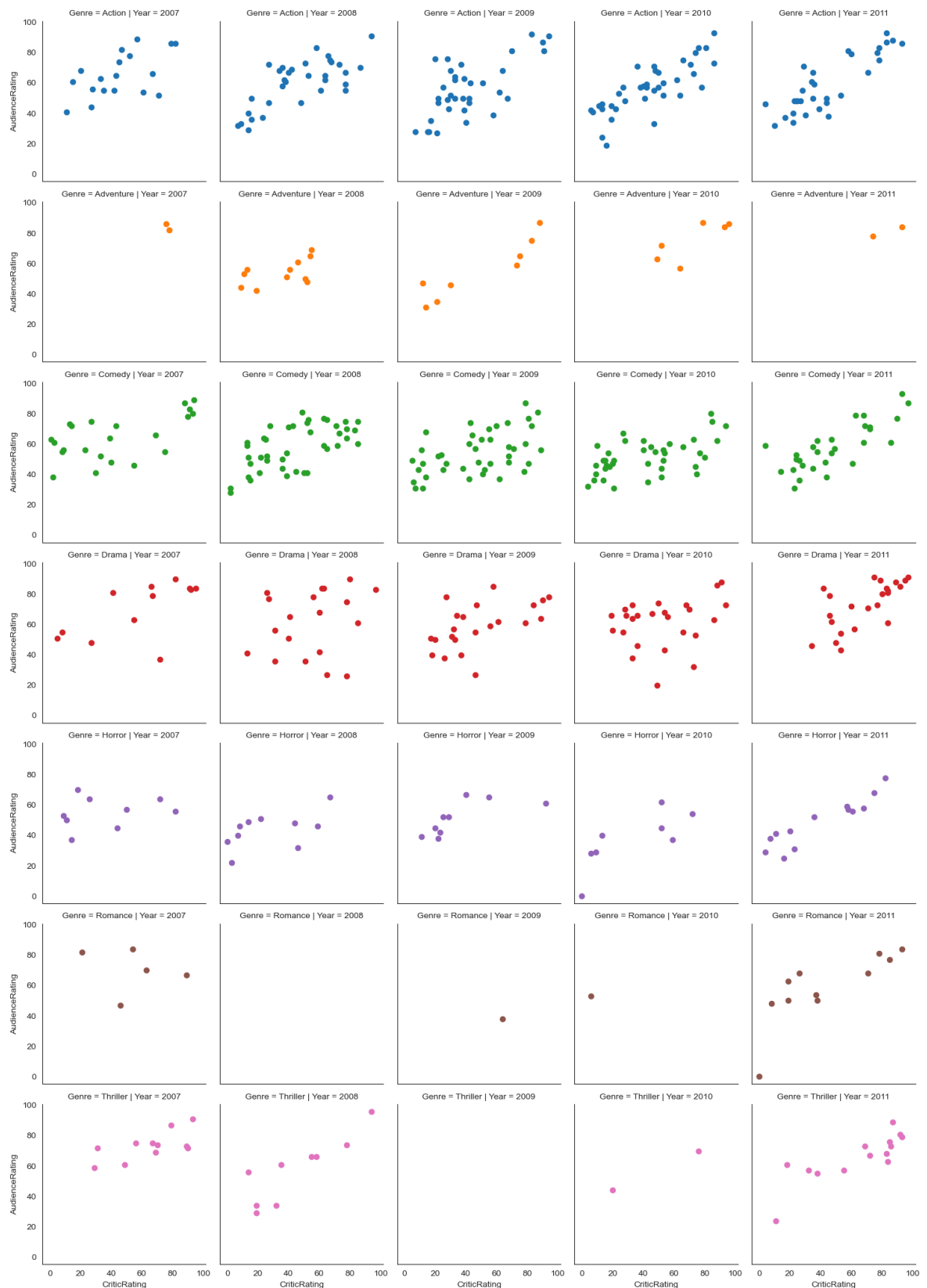


```
In [131... plt.scatter(movies.CriticRating,movies.AudienceRating)
```

```
Out[131... <matplotlib.collections.PathCollection at 0x2b6b165b440>
```

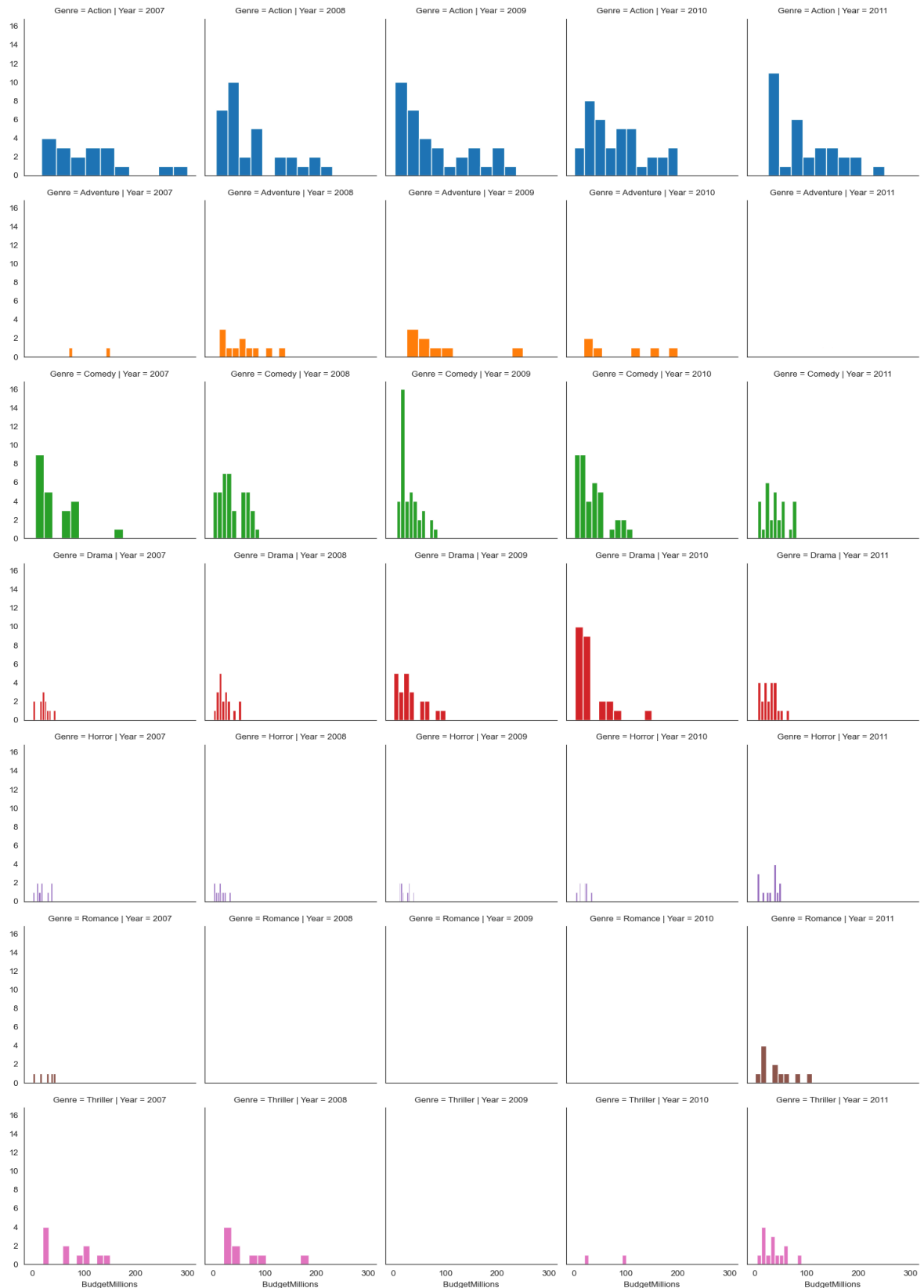


```
In [132... g = sns.FacetGrid(movies, row = 'Genre', col = 'Year', hue = 'Genre')  
g = g.map(plt.scatter, 'CriticRating', 'AudienceRating')
```



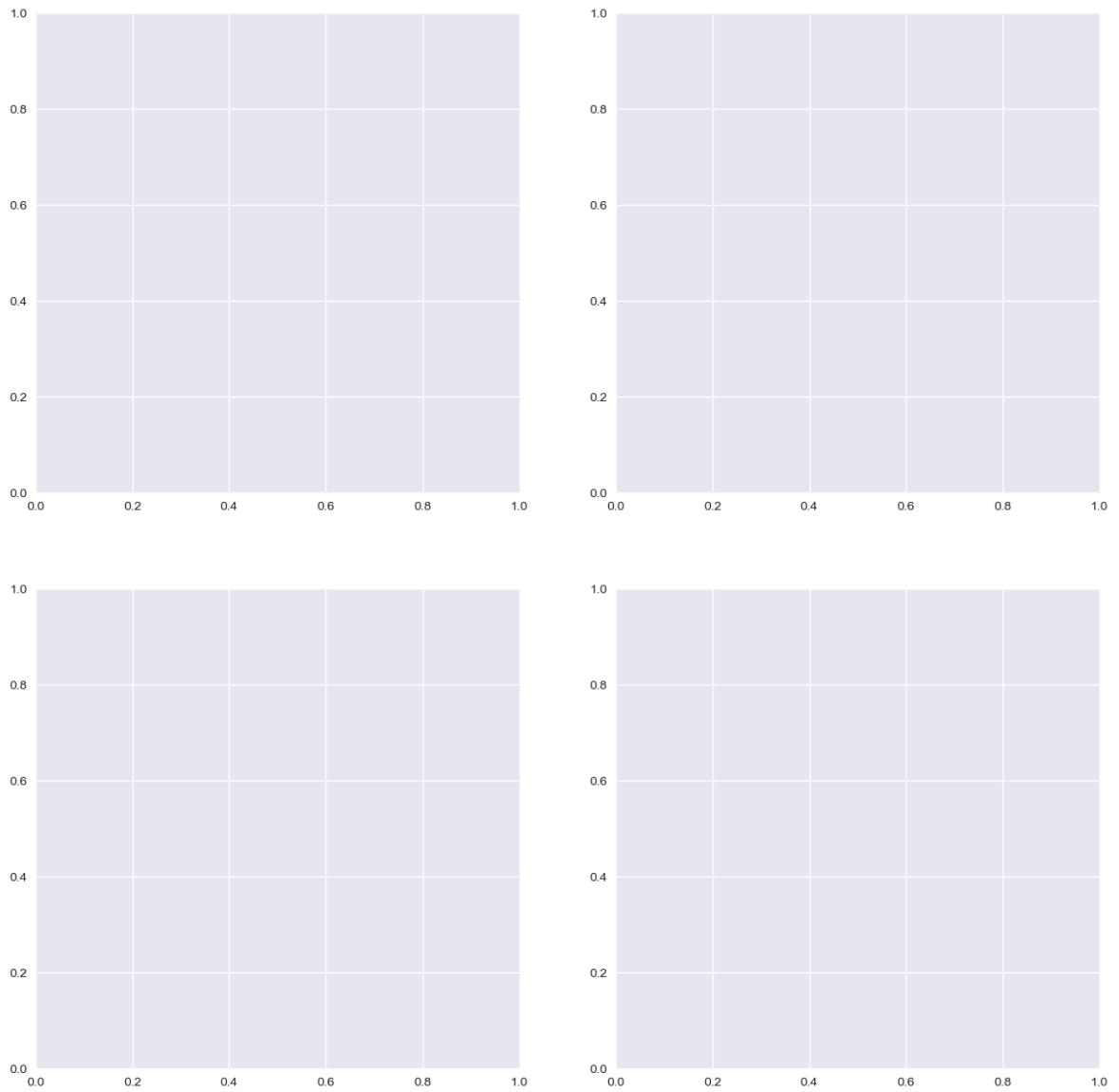
In [133...

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



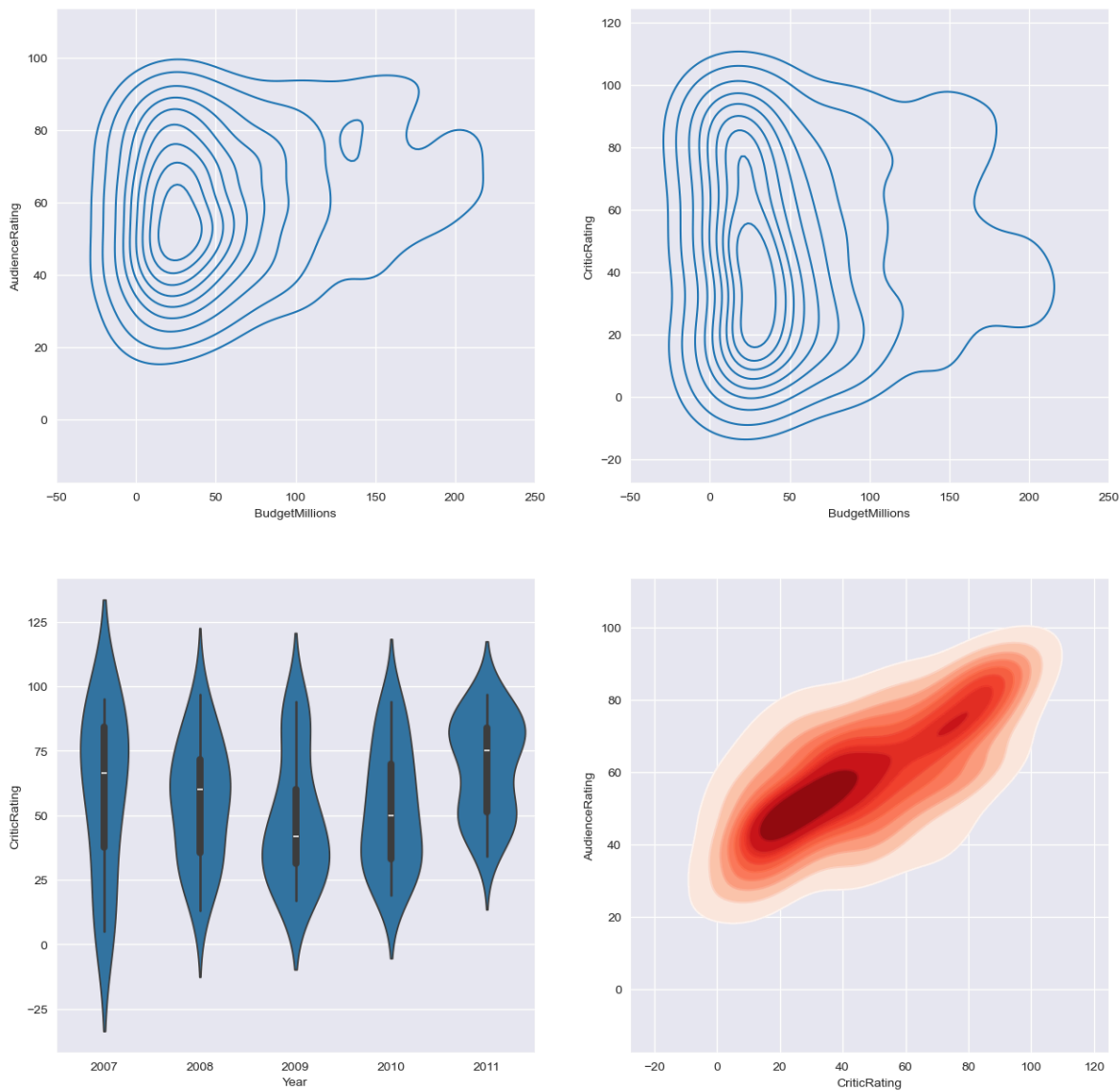
In [134...

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



```
In [135... 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=(-50,250))
k2.set(xlim=(-50,250))
z = sns.violinplot(data=movies[movies.Genre=='Drama'], x='Year', y='CriticRating')
k3 = sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,shade = True,shad
k3b = sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,cmap='Reds',ax =
plt.show()
```





In [137...

```
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_lowest=True,cmap = 'inferno', \
ax = axes[0,0])
k2 = sns.kdeplot(x=movies.BudgetMillions, y=movies.CriticRating,\
shade=True, shade_lowest=False, cmap='Spectral',\
ax = axes[0,1])

k3 = sns.kdeplot(x=movies.CriticRating, y=movies.AudienceRating, \
shade = False,shade_lowest=True,cmap='Blues_r', \
ax=axes[1,0])

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

k1.set(xlim=(-50,250))
k2.set(xlim=(-50,250))
plt.show()
```

