

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

```
In [2]: movies=pd.read_csv(r'C:\Users\arsha_4tjdyqj\Downloads\Movie-Rating.csv')
movies
```

```
Out[2]:
```

	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 [3]: movies.columns
```

```
Out[3]: Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
              'Budget (million $)', 'Year of release'],
              dtype='object')
```

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

```
Out[4]: pandas.core.frame.DataFrame
```

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

1.26.4

```
In [6]: len(movies)
```

```
Out[6]: 559
```

```
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.describe()
```

```
Out[10]:
```

	Rotten Tomatoes Ratings %	Audience Ratings %	Budget (million \$)	Year of release
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 [11]: movies.columns=['Film', 'Genre', 'CriticRating', 'AudienceRatings',
                          'BudgetMillions', 'Year']
```

```
In [12]: movies.columns
```

```
Out[12]: Index(['Film', 'Genre', 'CriticRating', 'AudienceRatings', 'BudgetMillions',
                'Year'],
                dtype='object')
```

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

Out[13]:

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

In [14]:

```
movies.describe()
```

Out[14]:

	CriticRating	AudienceRatings	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.Film=movies.Film.astype('category')
movies.Genre=movies.Genre.astype('category')
movies.Year=movies.Year.astype('category')
```

In [16]:

```
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   AudienceRatings 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 [17]:

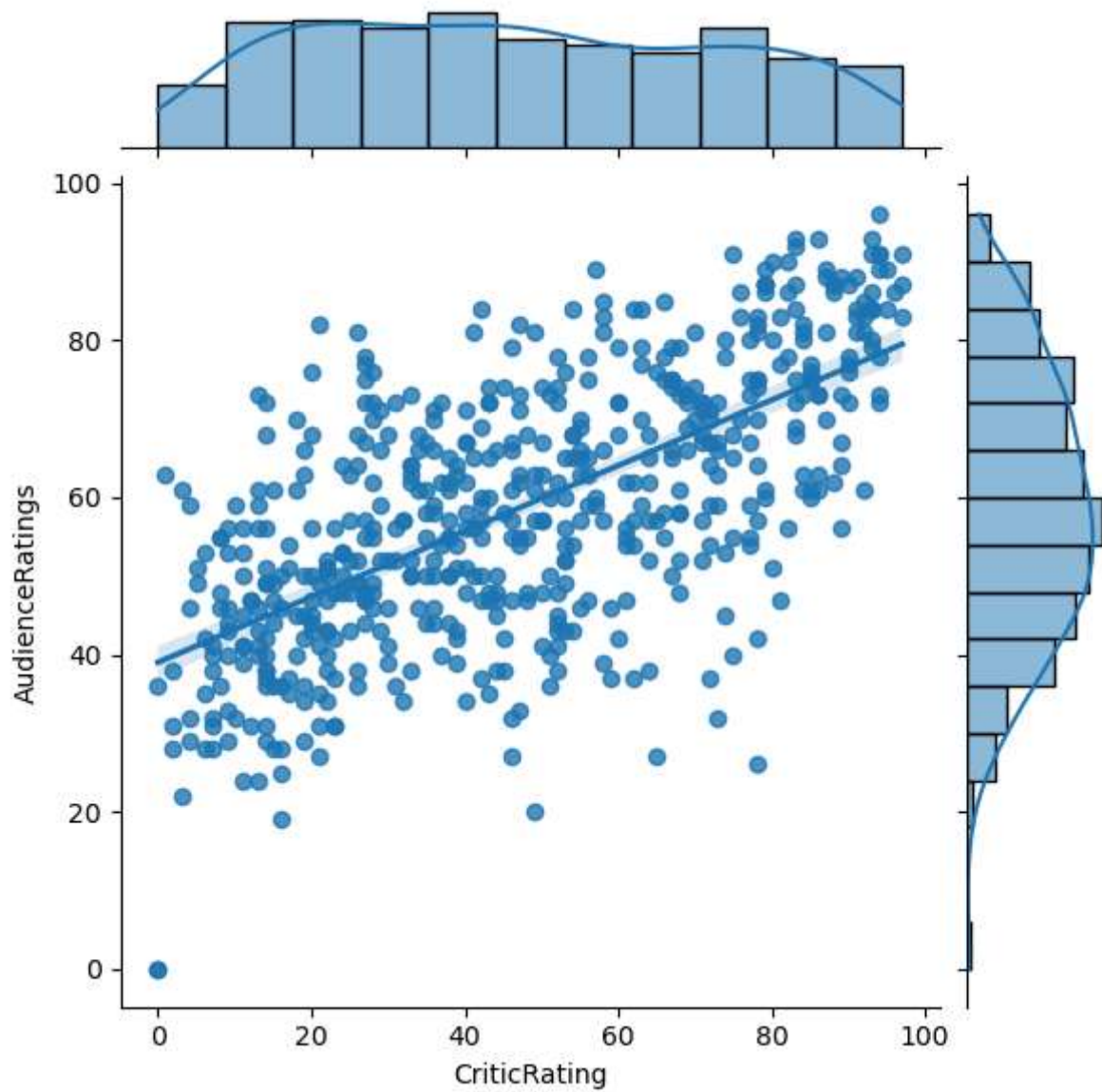
```
movies.describe()
```

Out[17]:

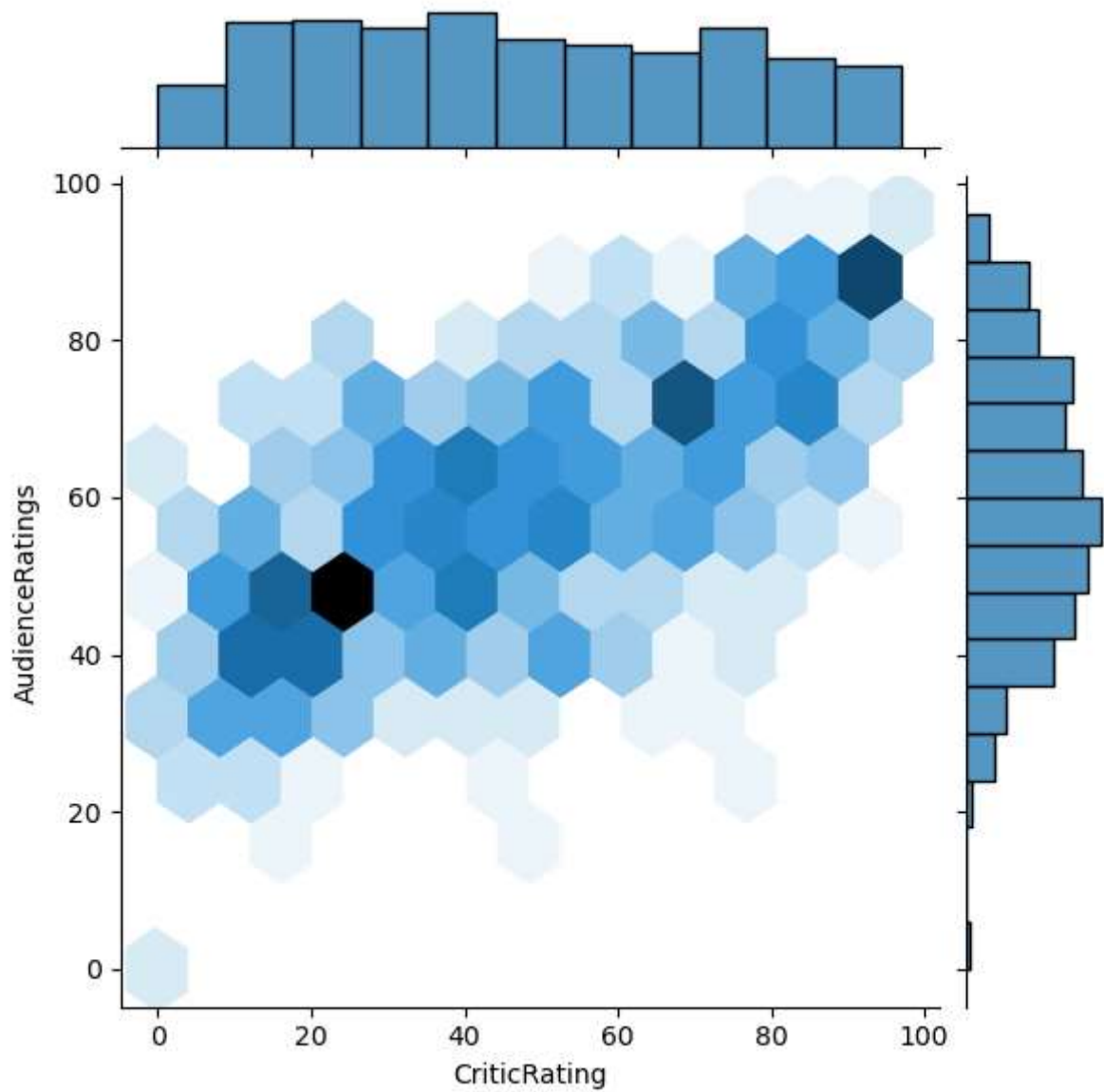
	CriticRating	AudienceRatings	BudgetMillions
count	559.000000	559.000000	559.000000
mean	47.309481	58.744186	50.236136
std	26.413091	16.826887	48.731817
min	0.000000	0.000000	0.000000
25%	25.000000	47.000000	20.000000
50%	46.000000	58.000000	35.000000
75%	70.000000	72.000000	65.000000
max	97.000000	96.000000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

```
In [18]: from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

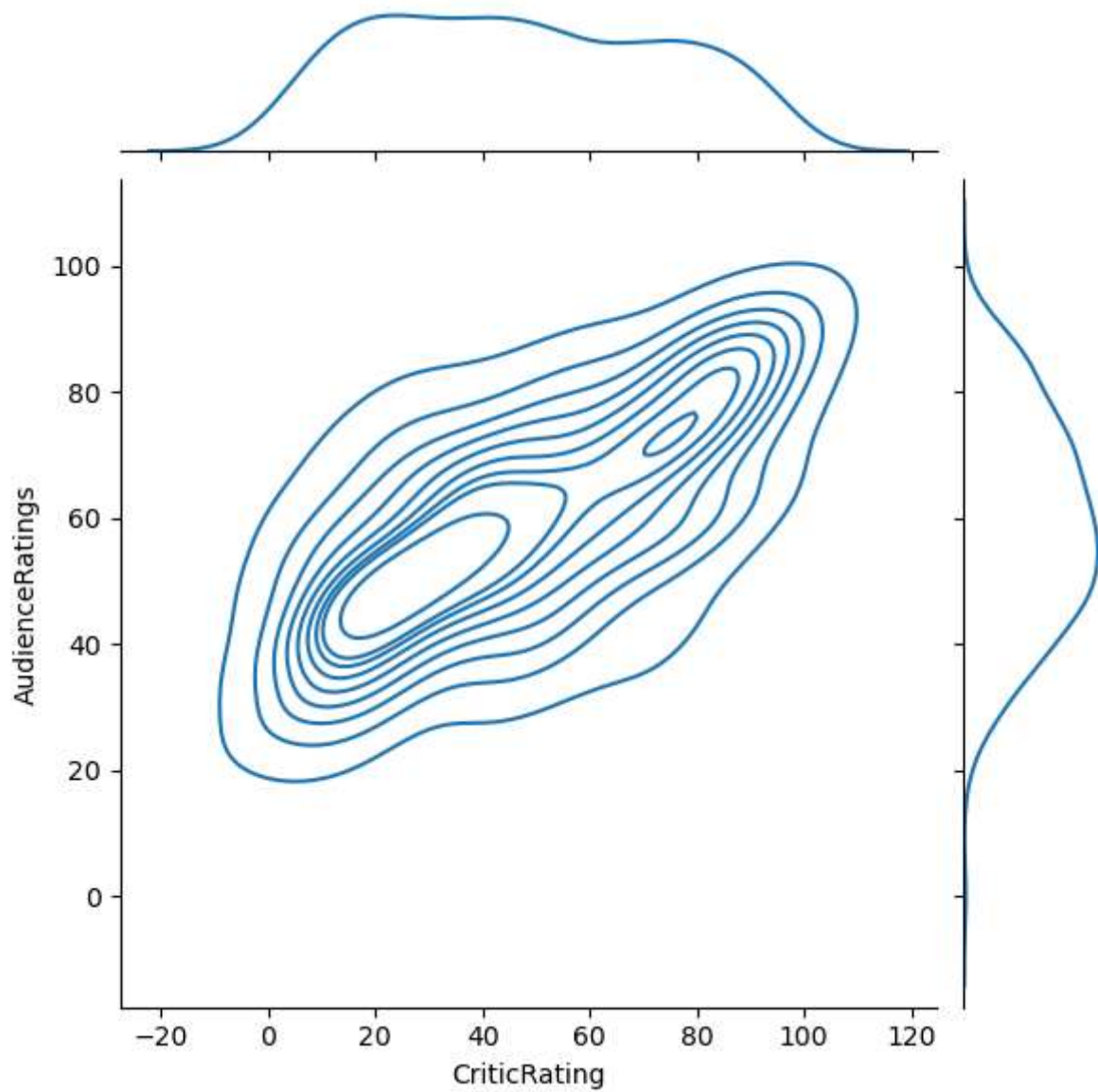
```
In [19]: j=sns.jointplot(data=movies, x='CriticRating',y= 'AudienceRatings', kind='reg')
```



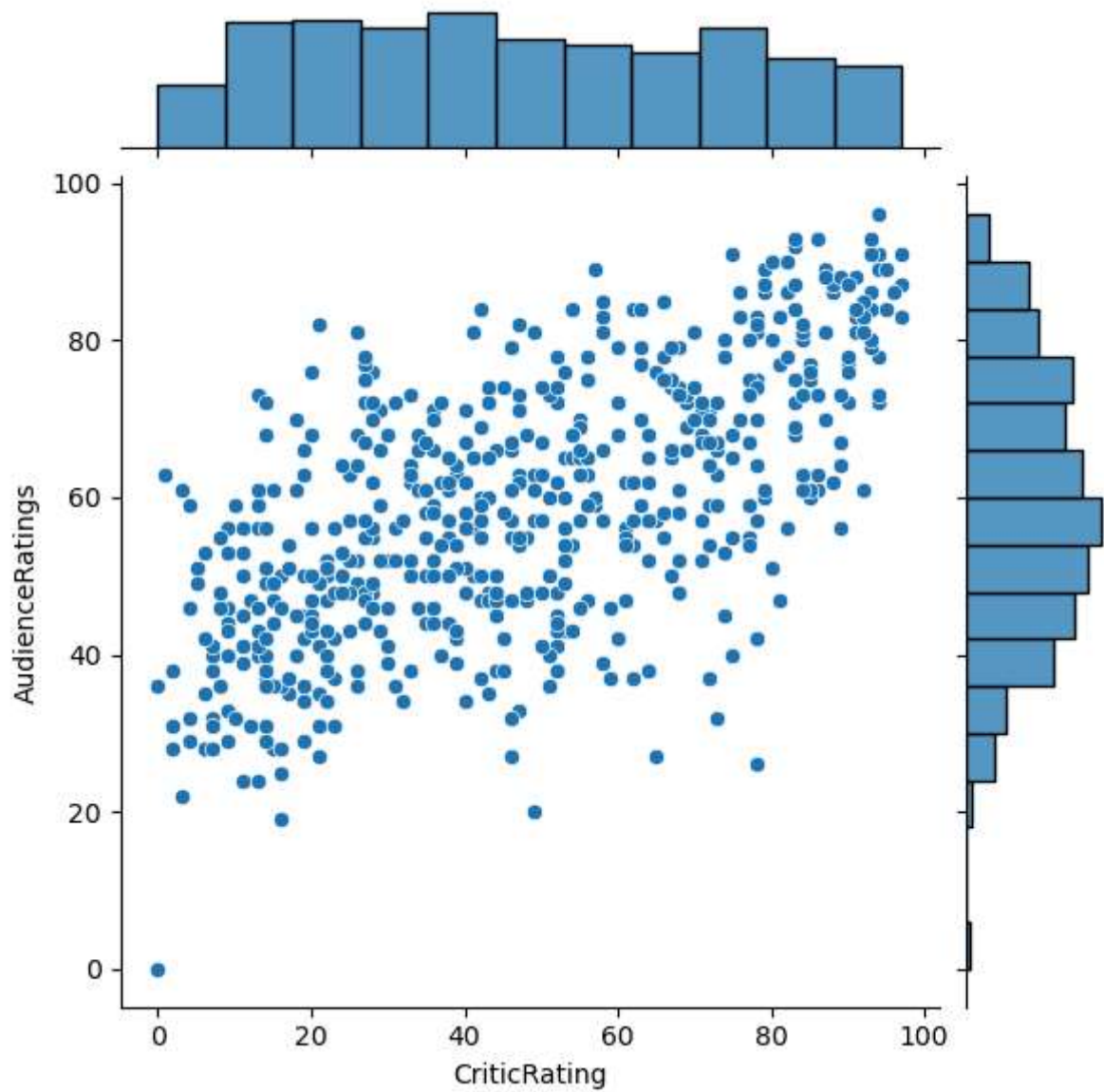
```
In [20]: j=sns.jointplot(data=movies, x='CriticRating',y= 'AudienceRatings', kind='hex')
```



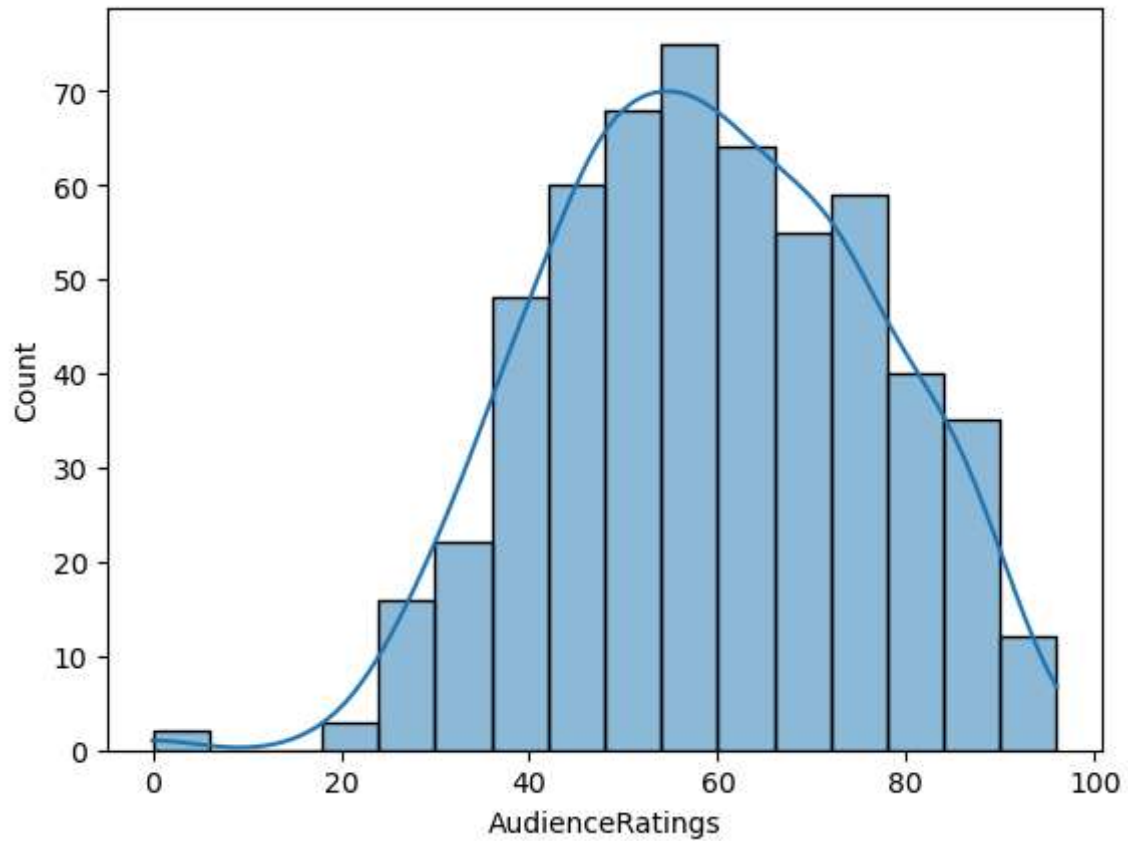
```
In [21]: j=sns.jointplot(data=movies, x='CriticRating',y= 'AudienceRatings', kind='kde')
```



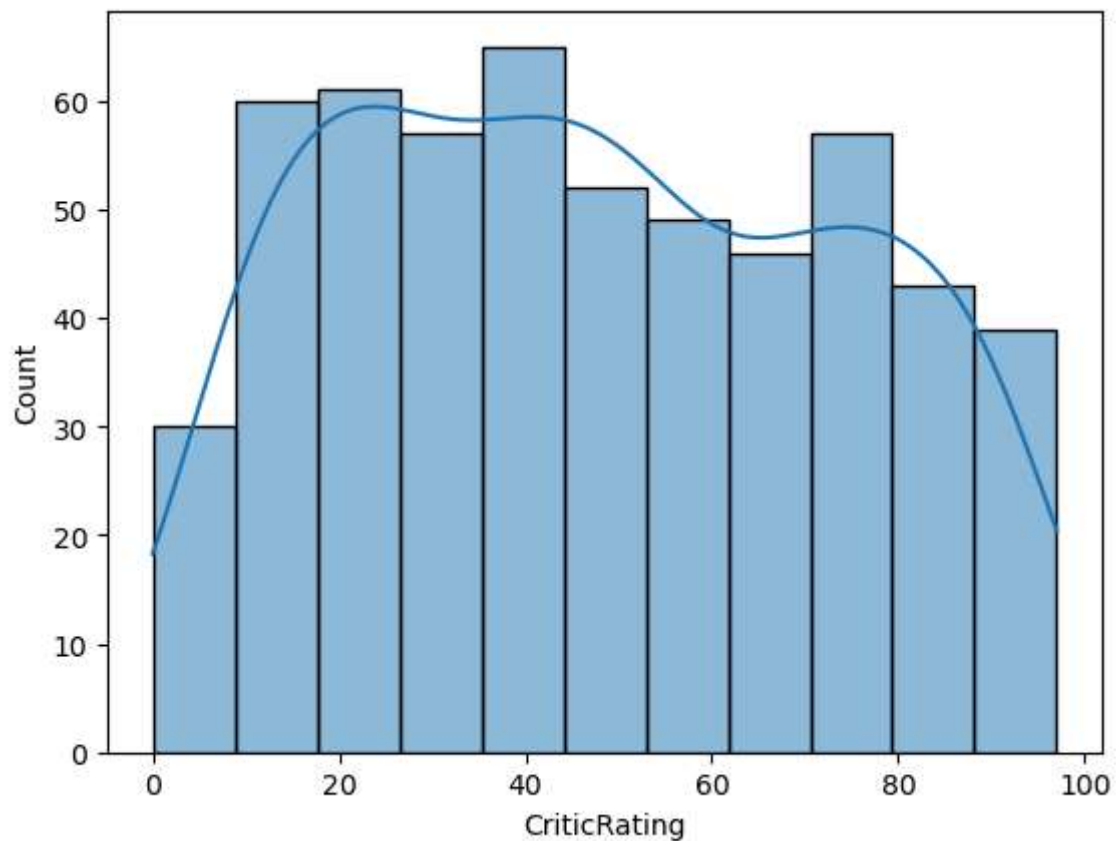
```
In [22]: j=sns.jointplot(data=movies, x='CriticRating',y= 'AudienceRatings', kind='scatter')
```



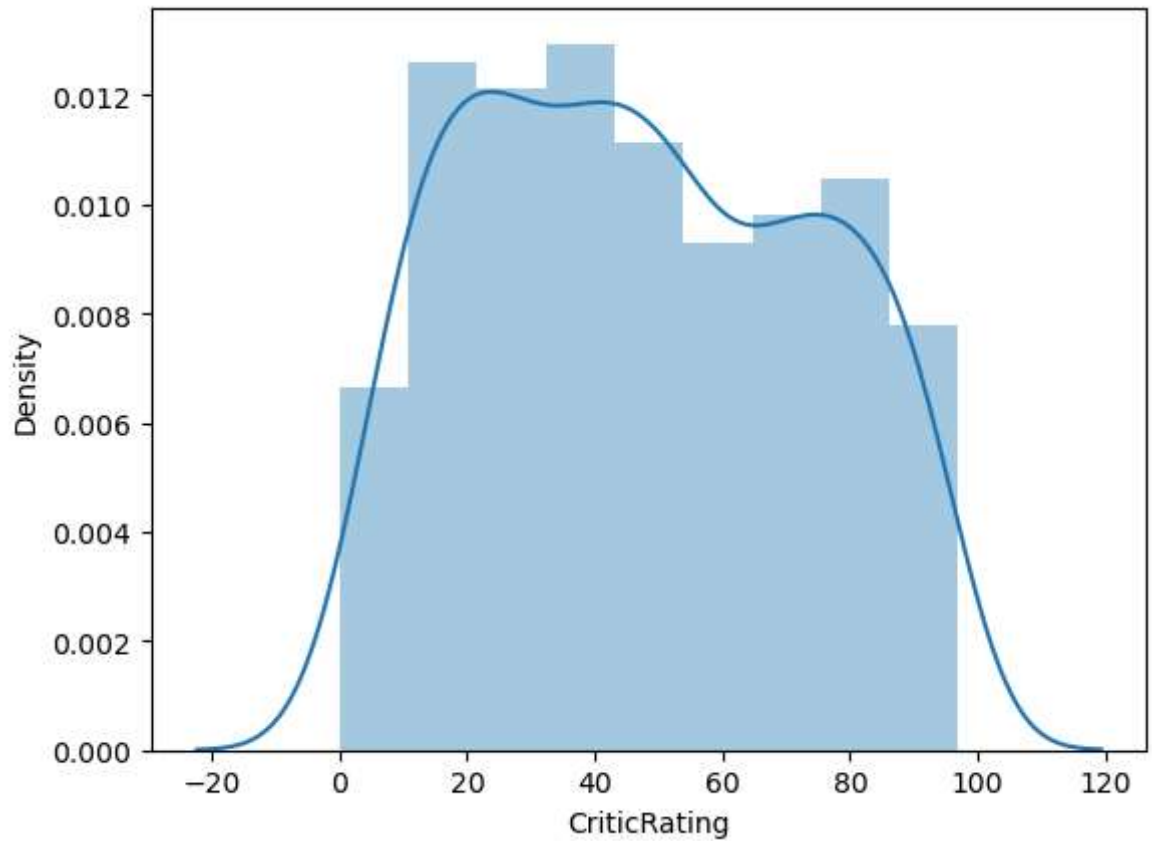
```
In [23]: m1=sns.histplot(movies.AudienceRatings,kde=True)
```

```
In [24]: m1=sns.histplot(movies.CriticRating,kde=True)
```

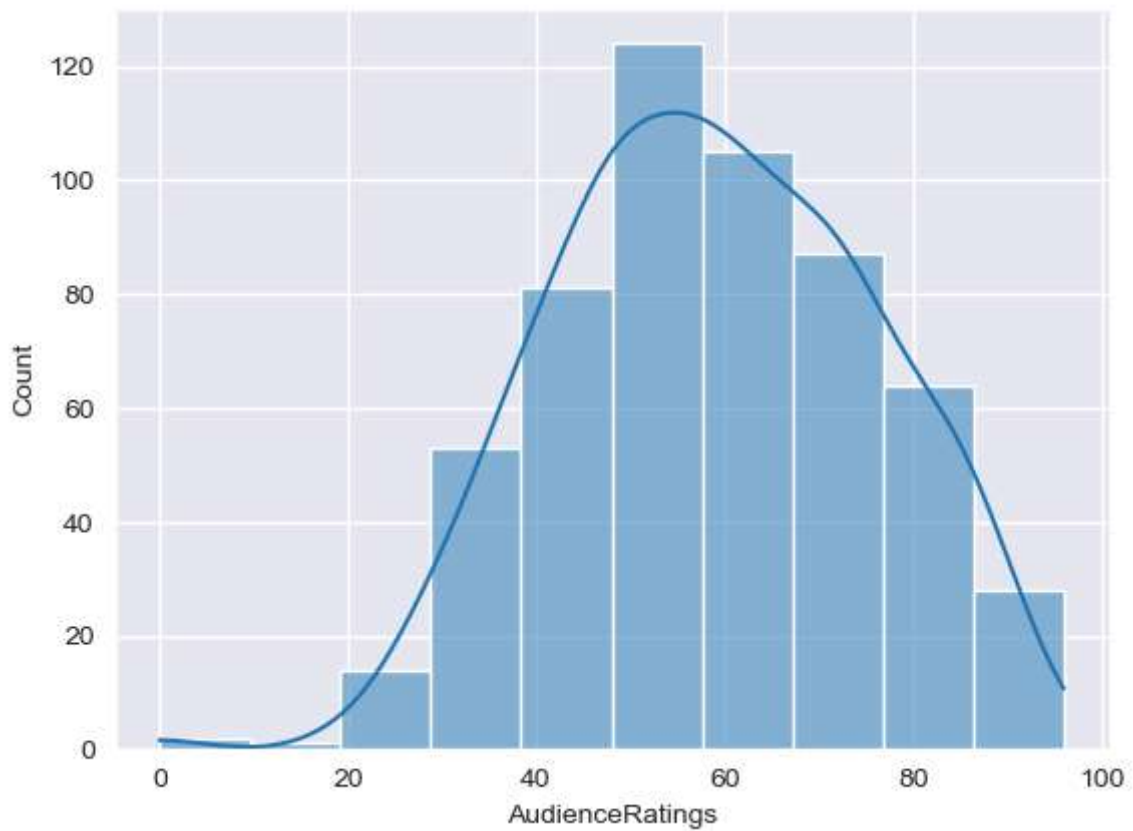


```
In [25]: m1=sns.distplot(movies.CriticRating)
```



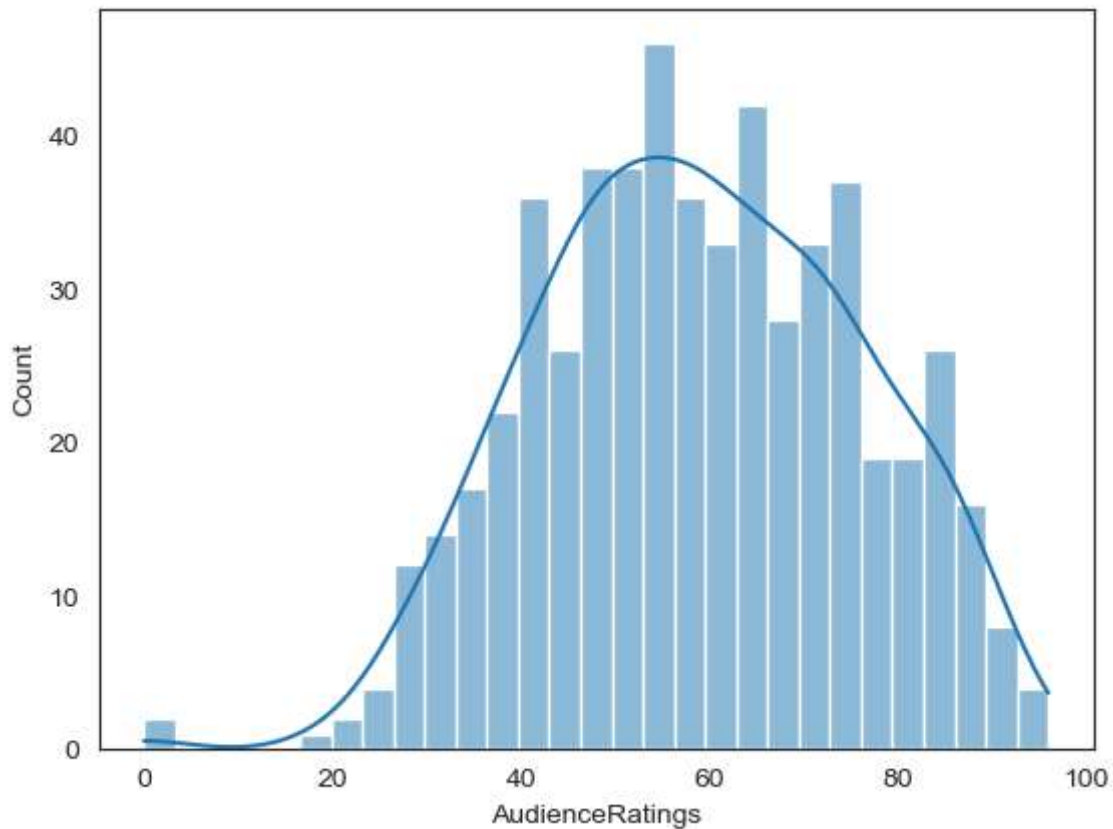
```
In [26]: sns.set_style('darkgrid')
```

```
In [27]: m2=sns.histplot(movies.AudienceRatings, bins=10,kde=True)
```



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

```
In [29]: m3=sns.histplot(movies.AudienceRatings, bins=29,kde=True)
```



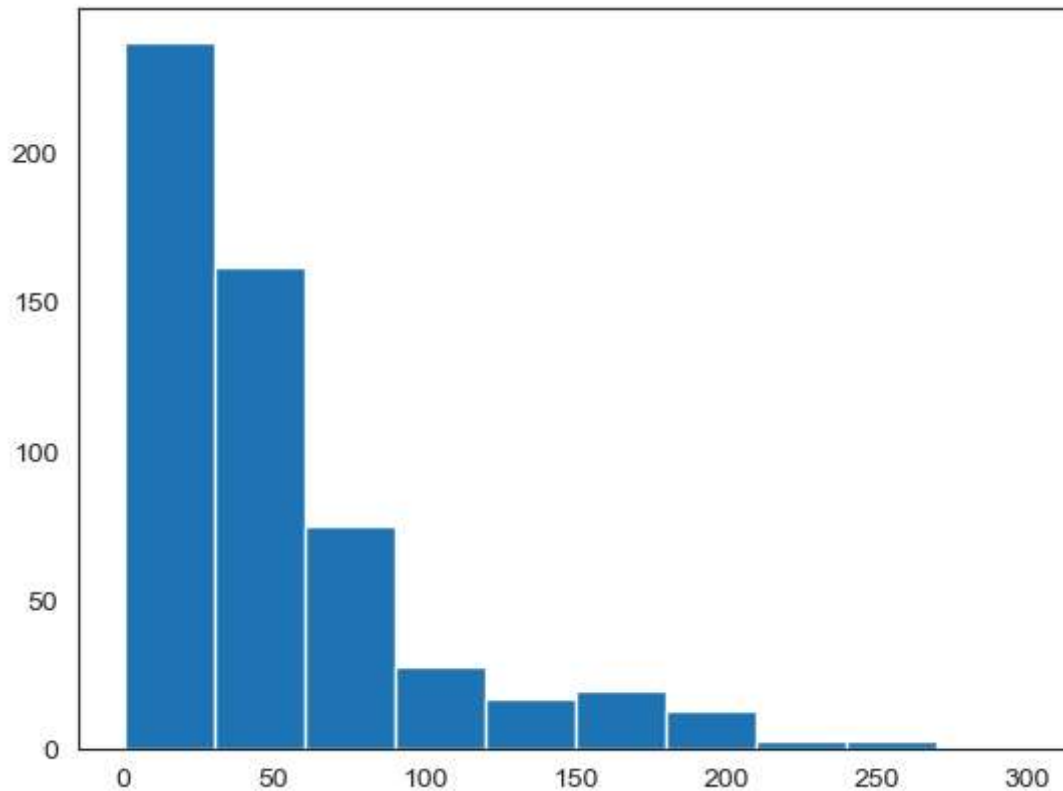
```
In [30]: movies.head()
```

```
Out[30]:
```

	Film	Genre	CriticRating	AudienceRatings	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 [31]: plt.hist(data=movies,x='BudgetMillions')
```

```
Out[31]: (array([237., 162., 75., 28., 17., 20., 13., 3., 3., 1.]),
array([ 0., 30., 60., 90., 120., 150., 180., 210., 240., 270., 300.]),
<BarContainer object of 10 artists>)
```



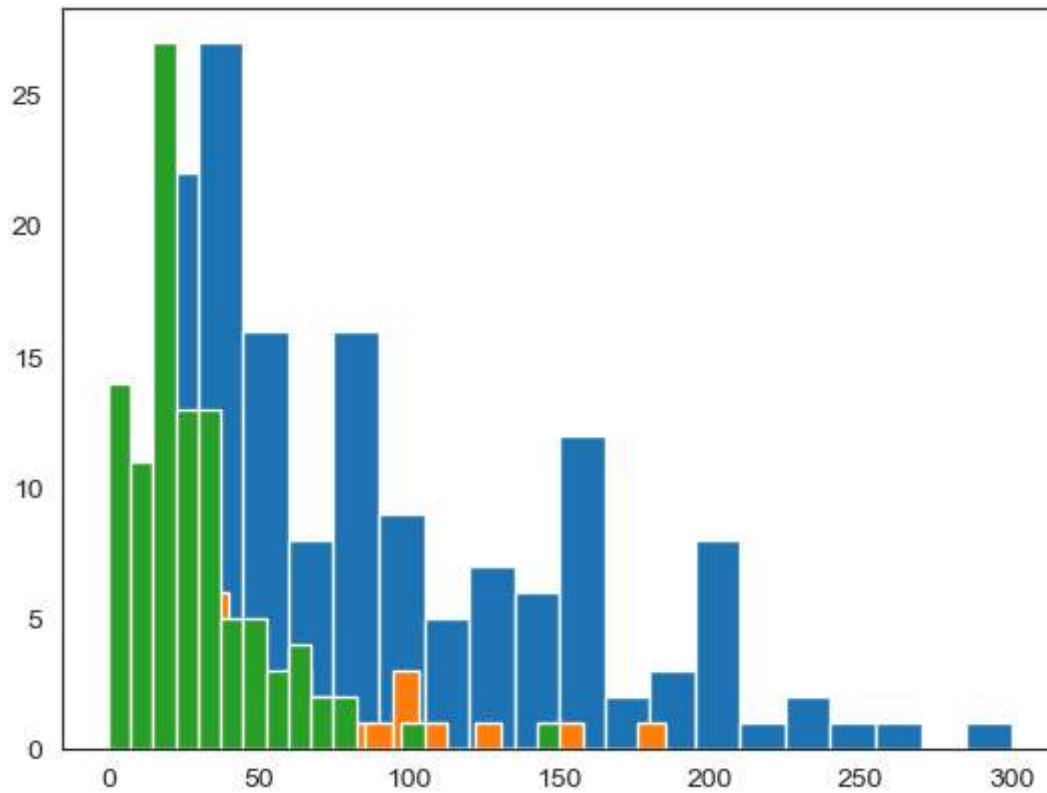
```
In [32]: if 'Genre' in movies.columns:
          print("Genre is present in movies DataFrame.")
        else:
          print("Genre is not present in movies DataFrame.")
```

Genre is present in movies DataFrame.

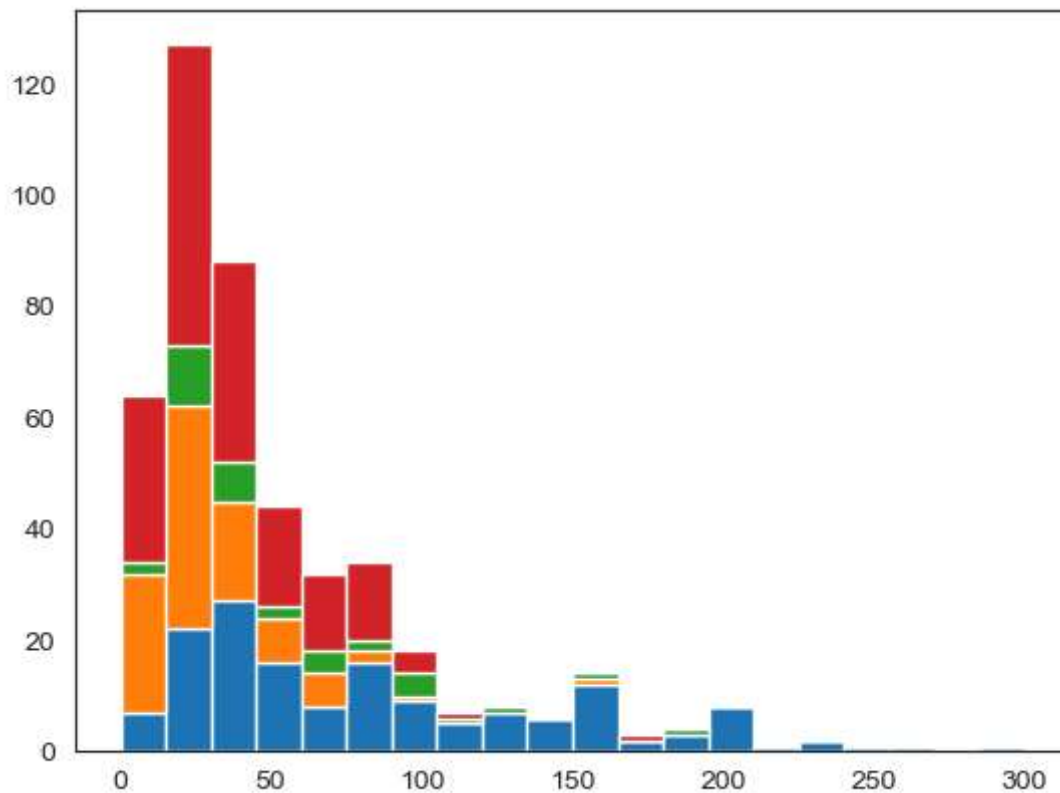
```
In [33]: movies.columns
```

```
Out[33]: Index(['Film', 'Genre', 'CriticRating', 'AudienceRatings', 'BudgetMillions',
               'Year'],
              dtype='object')
```

```
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.show()
```



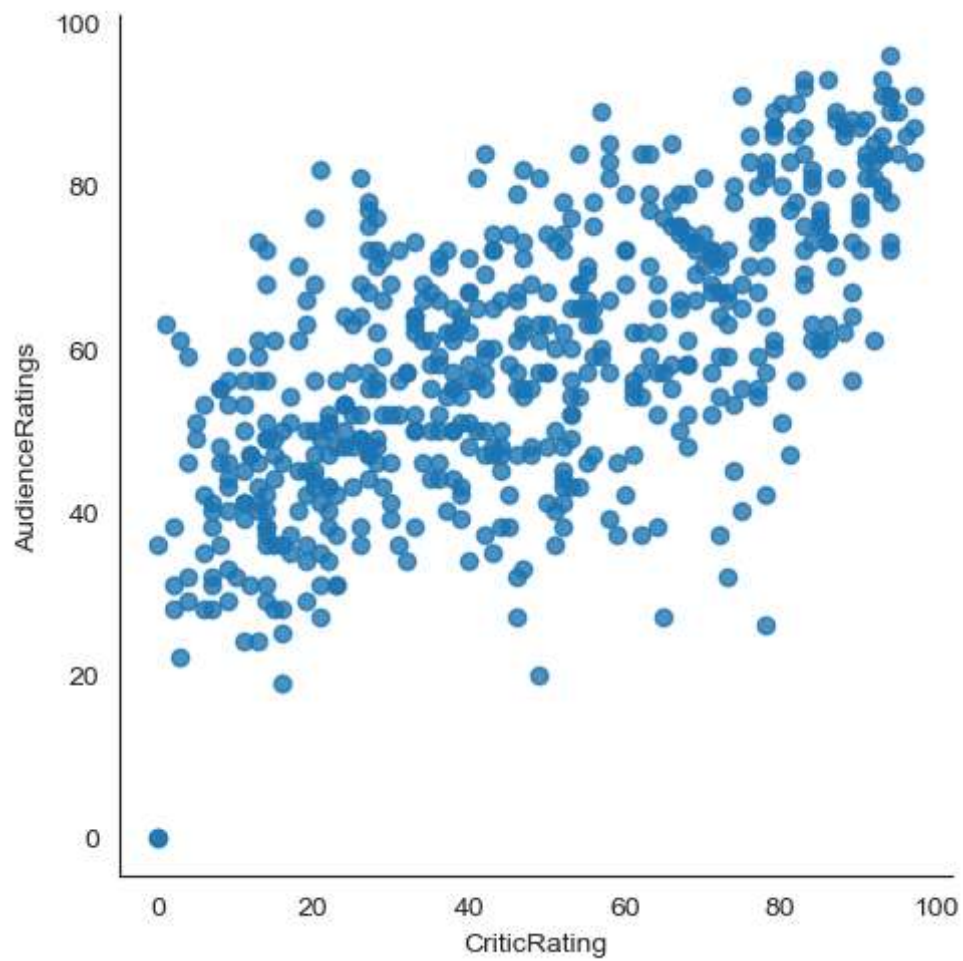
```
In [35]: plt.hist([movies[movies.Genre == 'Action'].BudgetMillions,\n                 movies[movies.Genre == 'Drama'].BudgetMillions, \n                 movies[movies.Genre == 'Thriller'].BudgetMillions, \n                 movies[movies.Genre == 'Comedy'].BudgetMillions],\n                 bins =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



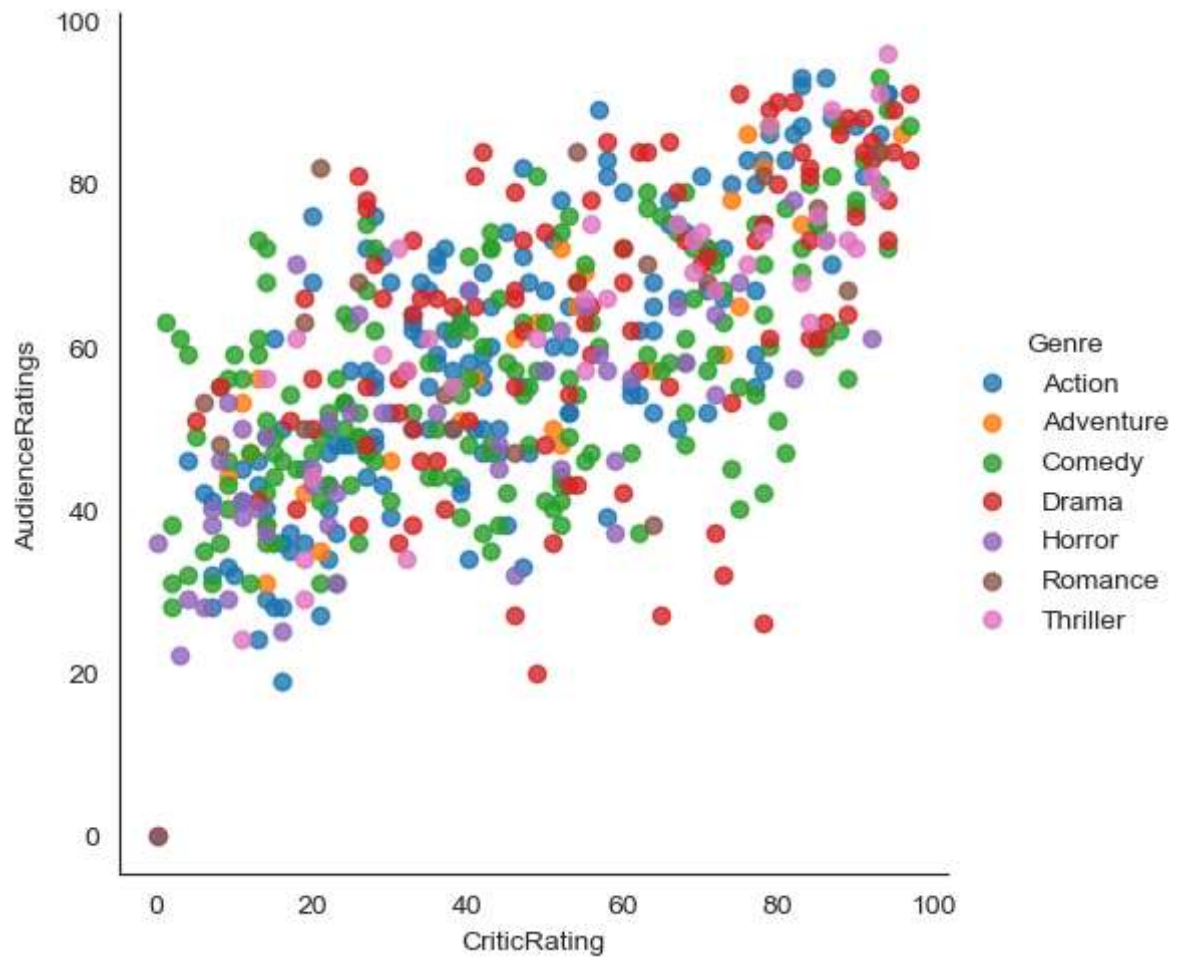
```
In [36]: for gen in movies.Genre.cat.categories:  
         print(gen)
```

Action
Adventure
Comedy
Drama
Horror
Romance
Thriller

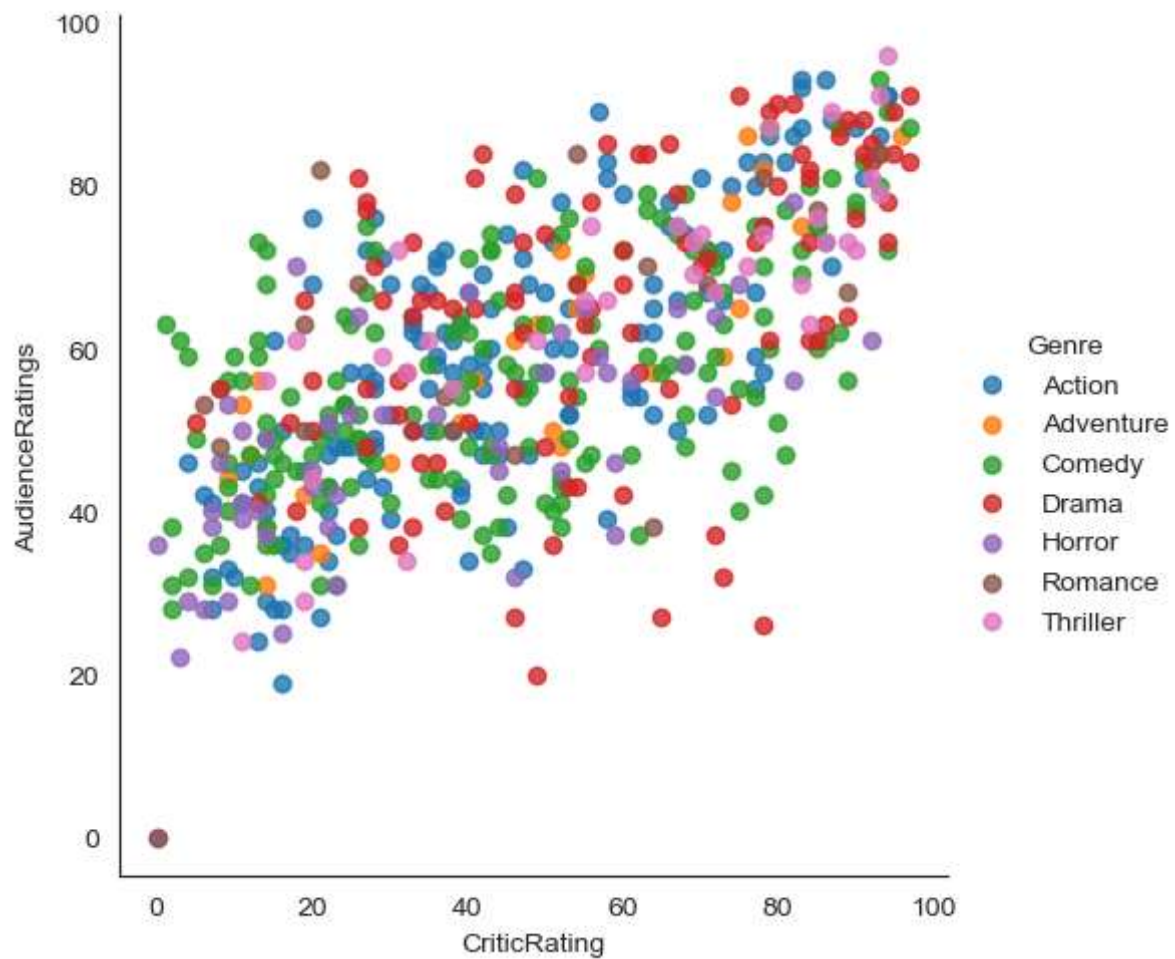
```
In [37]: vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRatings',\  
                        fit_reg=False)
```



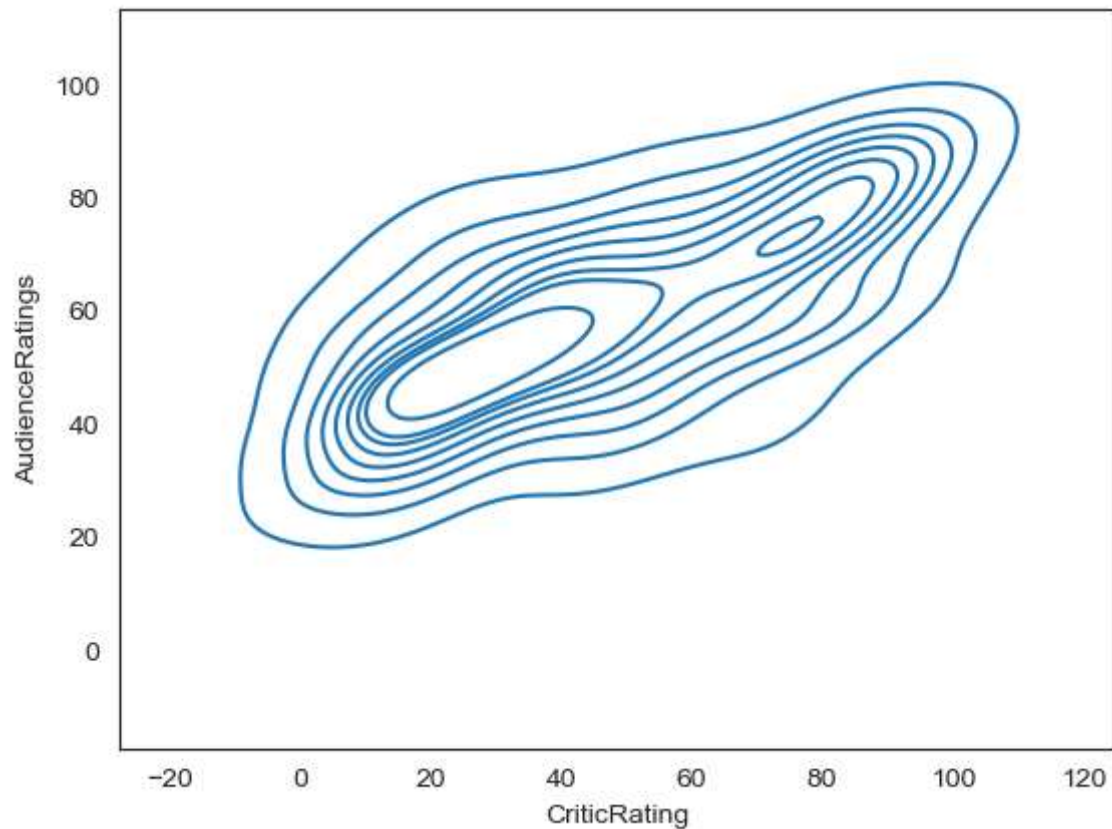
```
In [38]: vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRatings',\
                        fit_reg=False , hue='Genre')
```



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

```
In [47]: k1=sns.kdeplot(data=movies,x='CriticRating',y='AudienceRatings')
```



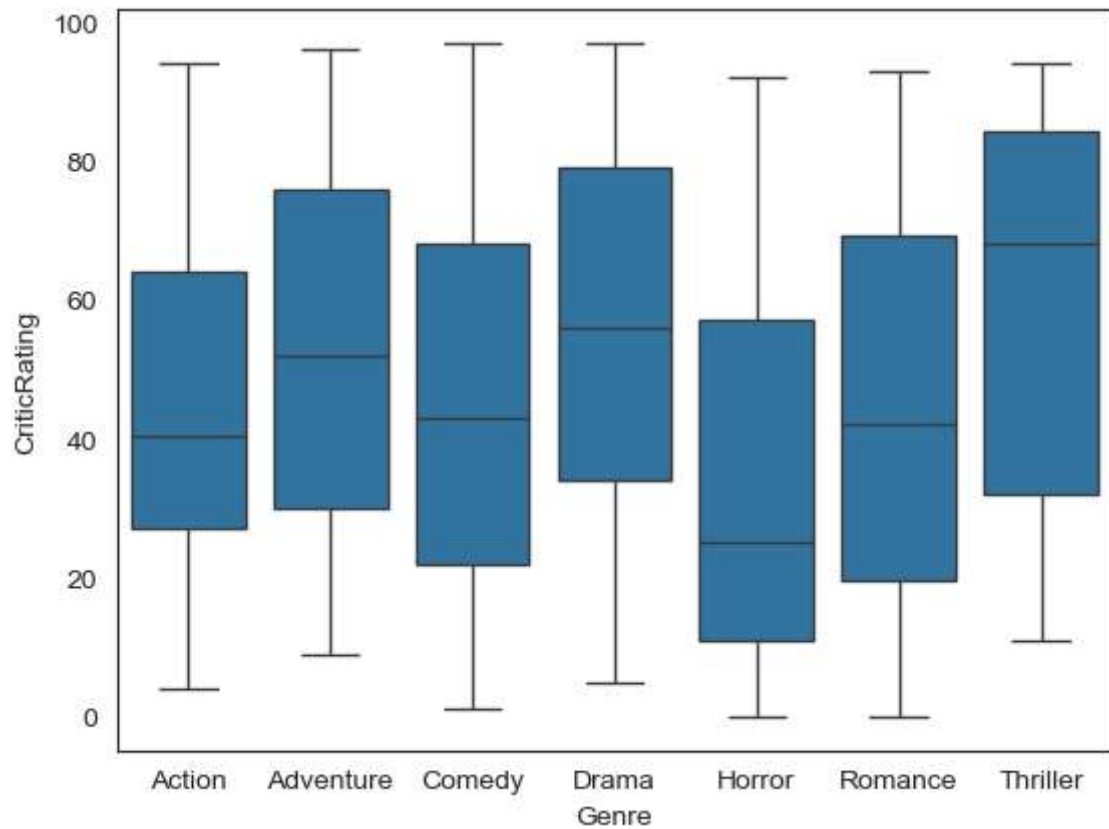
```
In [43]: movies.columns
```

```
Out[43]: Index(['Film', 'Genre', 'CriticRating', 'AudienceRatings', 'BudgetMillions',
               'Year'],
              dtype='object')
```

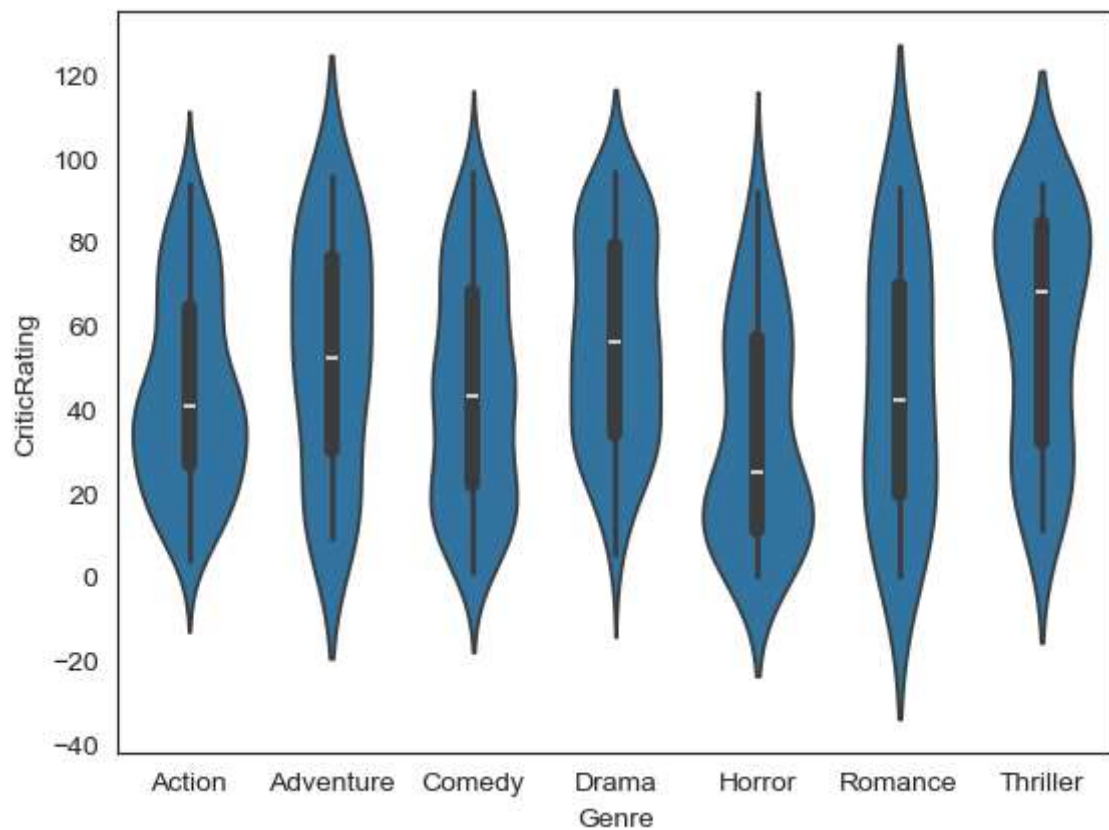
```
In [49]: movies.info()
```

```
<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   AudienceRatings 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 [51]: w=sns.boxplot(data=movies, x='Genre',y='CriticRating')
```



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



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

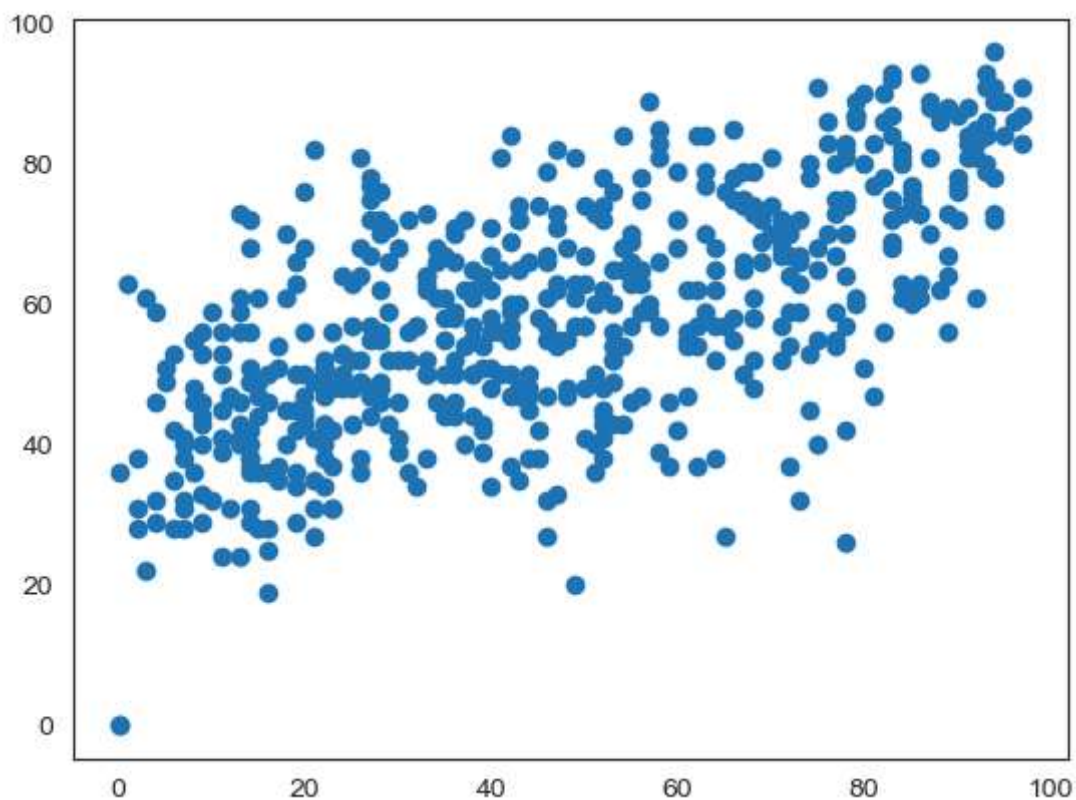


```
In [61]: movies.info()
```

```
<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   AudienceRatings       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 [69]: plt.scatter(movies.CriticRating,movies.AudienceRatings)
```

```
Out[69]: <matplotlib.collections.PathCollection at 0x181e2b2d640>
```



```
In [75]: 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   AudienceRatings       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 []:

In []: