

```
In [1]: import pandas as pd  
import os  
os.getcwd()
```

```
Out[1]: 'C:\\\\Users\\\\Jan Saida'
```

```
In [2]: movies = pd.read_csv(r"C:\\\\Users\\\\Jan Saida\\\\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]: movies.head()
```

```
Out[5]:
```

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

```
Out[6]:
```

	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 [7]: movies.columns
```

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

```
In [8]: movies.columns=['Film', 'Genre', 'CriticRating', 'AudienceRating', 'BudgetMillions', 'Year']
```

```
In [9]: movies.head()
```

	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 [10]: `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 [11]: `movies.describe()`

	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 [12]: `movies.transpose()`

	0	1	2	3	4	5	6	7	8	9	...	549	550
Film	(500) Days of Summer	10,000 B.C.	12 Rounds	127 Hours	17 Again	2012	27 Dresses	30 Days of Night	30 Minutes or Less	50/50	...	Yes Man	Yogi Bear
Genre	Comedy	Adventure	Action	Adventure	Comedy	Action	Comedy	Horror	Comedy	Comedy	...	Comedy	Comedy
CriticRating	87	9	30	93	55	39	40	50	43	93	...	43	14
AudienceRating	81	44	52	84	70	63	71	57	48	93	...	72	36
BudgetMillions	8	105	20	18	20	200	30	32	28	8	...	70	80
Year	2009	2008	2009	2010	2009	2009	2008	2007	2011	2011	...	2008	2010

6 rows × 559 columns

In [13]: `movies['Film']`

```
0      (500) Days of Summer
1                  10,000 B.C.
2                      12 Rounds
3                     127 Hours
4                     17 Again
...
554             Your Highness
555       Youth in Revolt
556              Zodiac
557          Zombieland
558        Zookeeper
Name: Film, Length: 559, dtype: object
```

In [14]: `movies['AudienceRating']`

```
Out[14]: 0      81
1      44
2      52
3      84
4      70
...
554     36
555     52
556     73
557     87
558     42
Name: AudienceRating, Length: 559, dtype: int64
```

```
In [15]: movies.Film
```

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

```
In [17]: movies.Film
```

```
Out[17]: 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 [18]: movies.head()
```

```
Out[18]:   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 [19]: 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 [20]: movies.Genre=movies.Genre.astype('category')
movies.Year=movies.Year.astype('category')
```

```
In [21]: movies.Genre
```

```
Out[21]: 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 [22]: movies.Year
```

```
Out[22]: 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 [23]: 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 [24]: movies.Genre.cat.categories
```

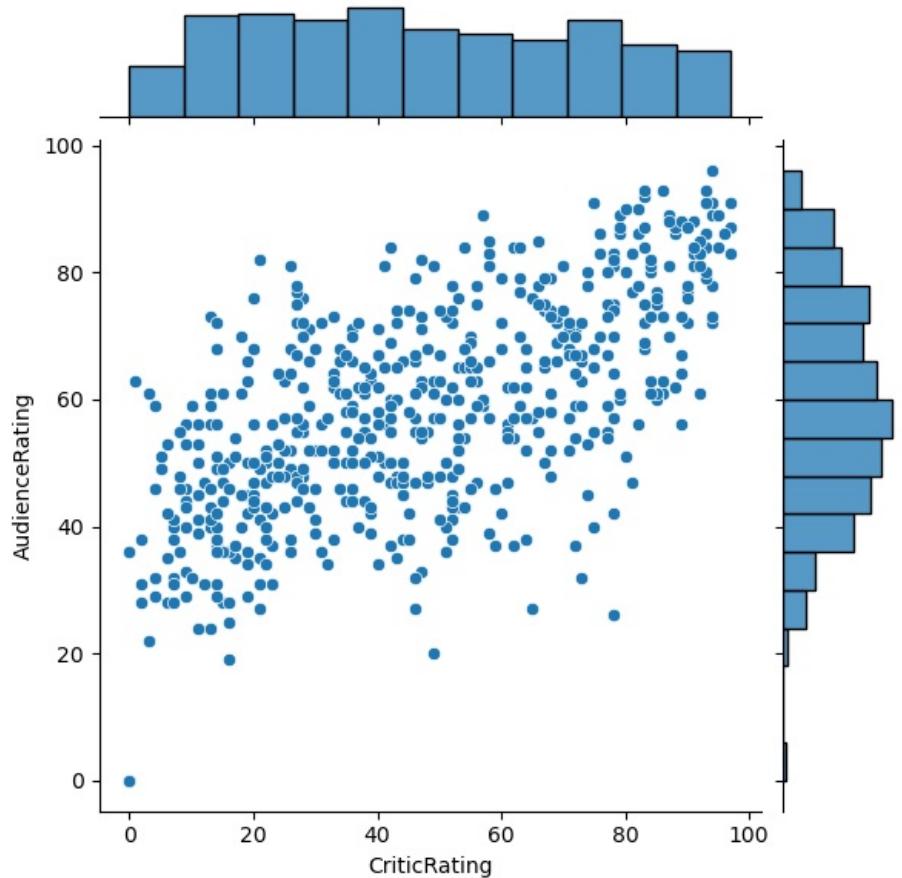
```
Out[24]: Index(['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance',
 'Thriller'],
 dtype='object')
```

```
In [25]: movies.describe()
```

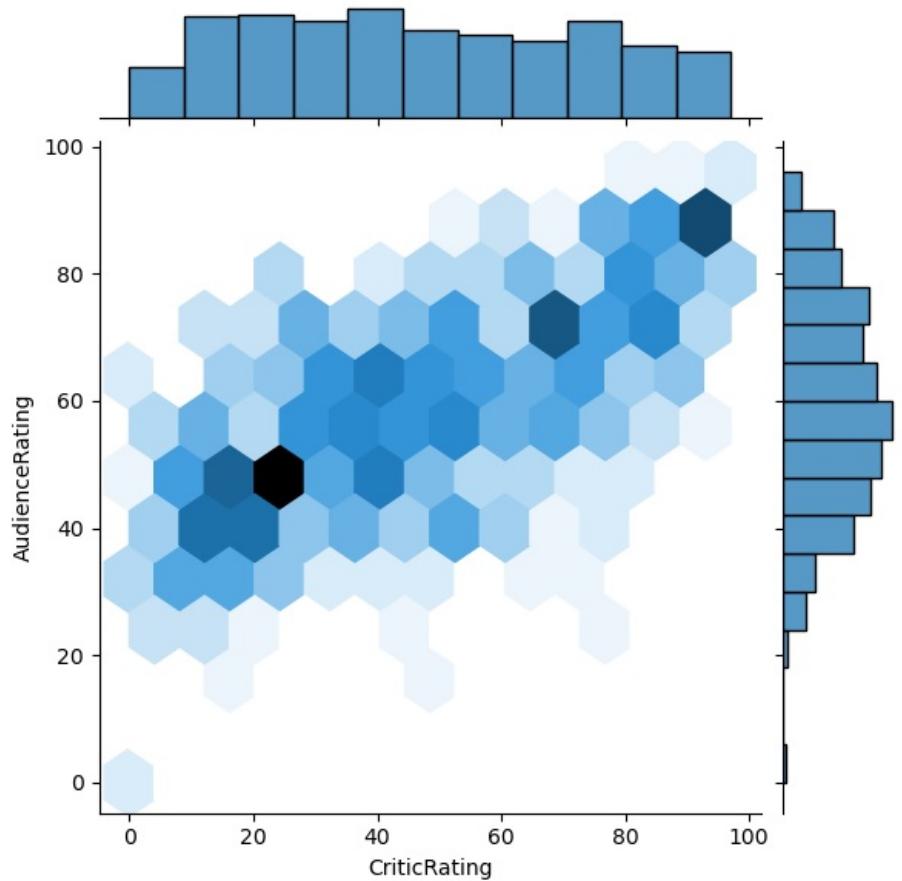
```
Out[25]:   CriticRating  AudienceRating  BudgetMillions
count      559.000000      559.000000      559.000000
mean       47.309481      58.744186      50.236136
std        26.413091      16.826887      48.731817
min        0.000000      0.000000      0.000000
25%       25.000000      47.000000  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

```
In [26]: from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

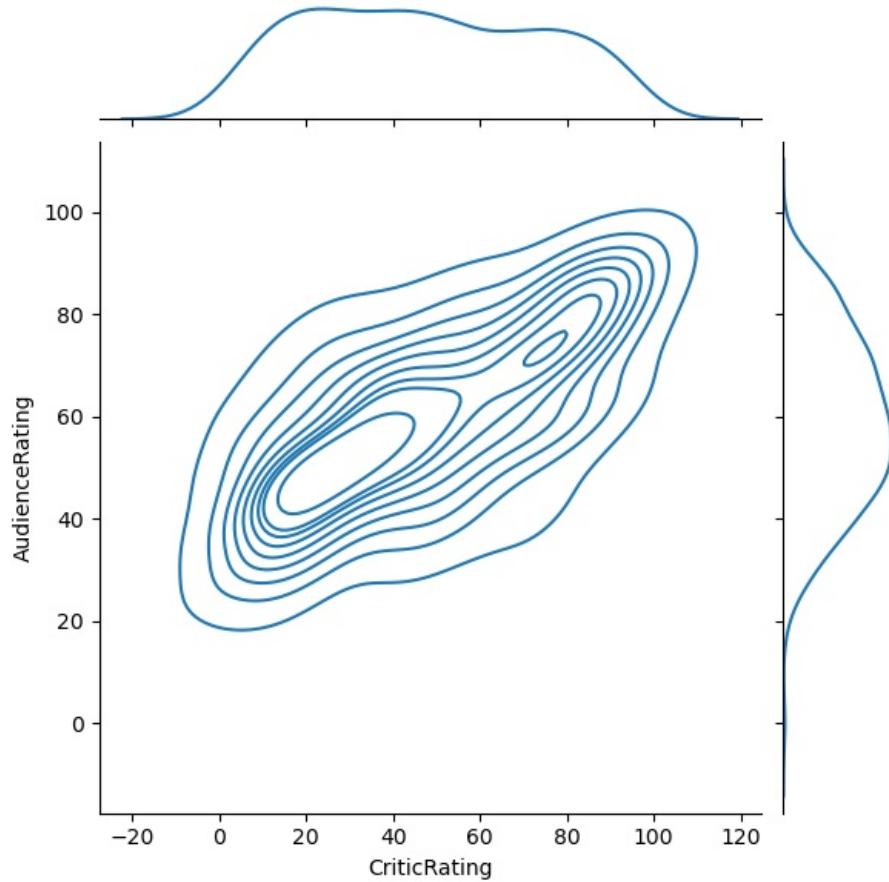
```
In [27]: j=sns.jointplot(data=movies, x='CriticRating', y='AudienceRating')
```



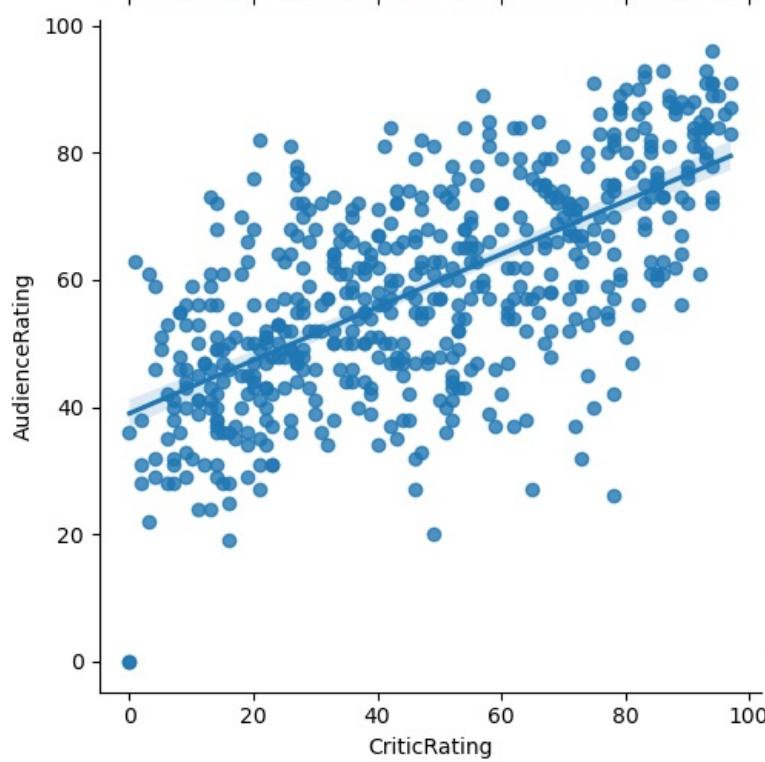
```
In [28]: j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating', kind='hex')
```



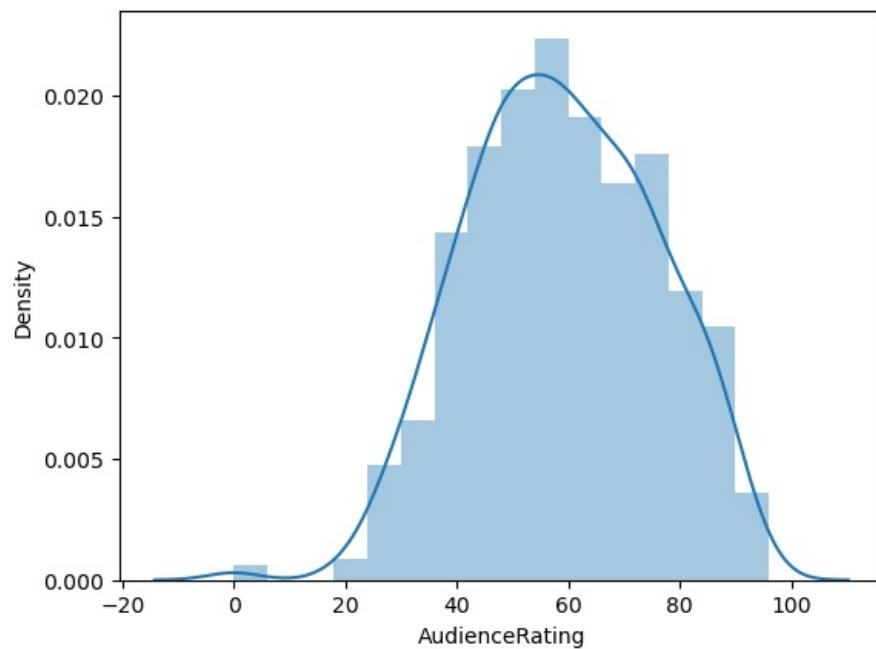
```
In [29]: j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating', kind='kde')
```



```
In [30]: j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating', kind='reg')
```

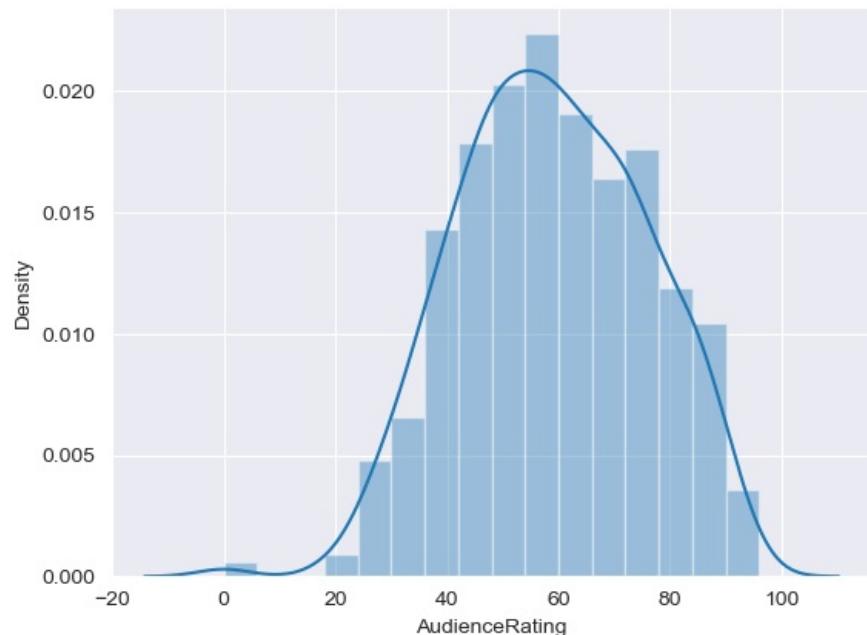


```
In [31]: m1=sns.distplot(movies.AudienceRating)
```

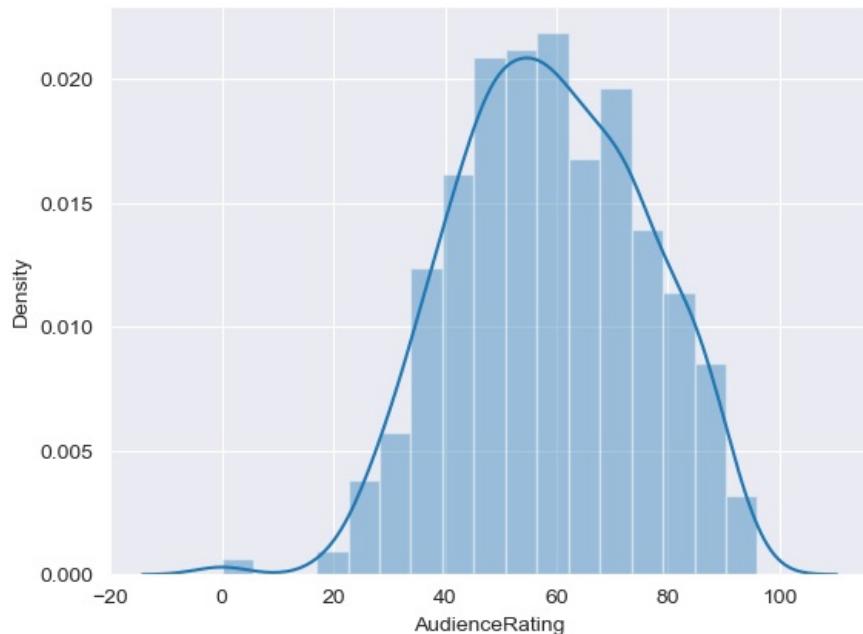


```
In [32]: sns.set_style('darkgrid')
```

```
In [33]: m2=sns.distplot(movies.AudienceRating)
```

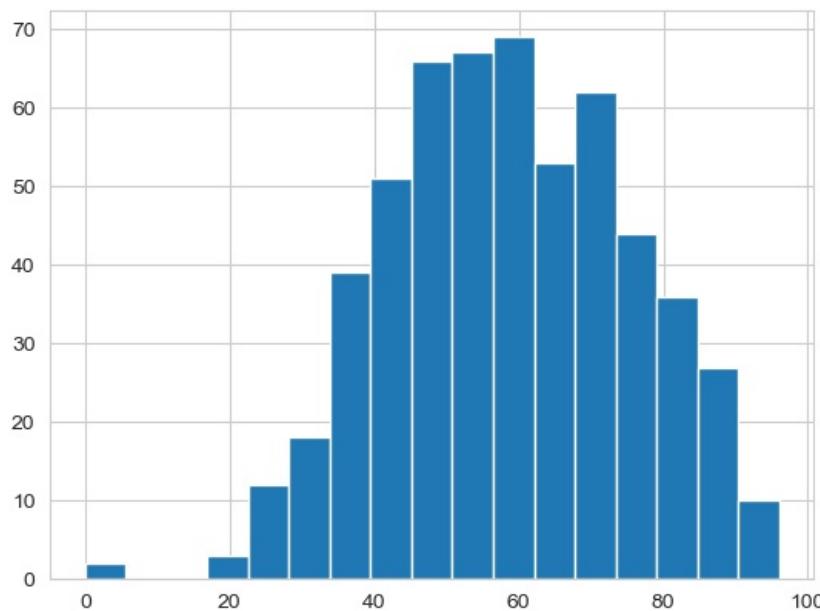


```
In [34]: m2=sns.distplot(movies.AudienceRating,bins=17)
```

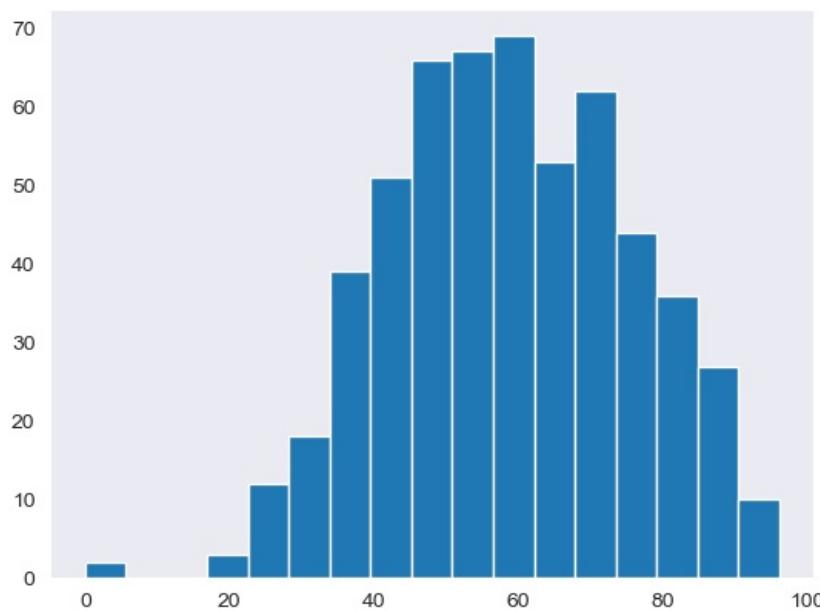


```
In [35]: sns.set_style('whitegrid')
```

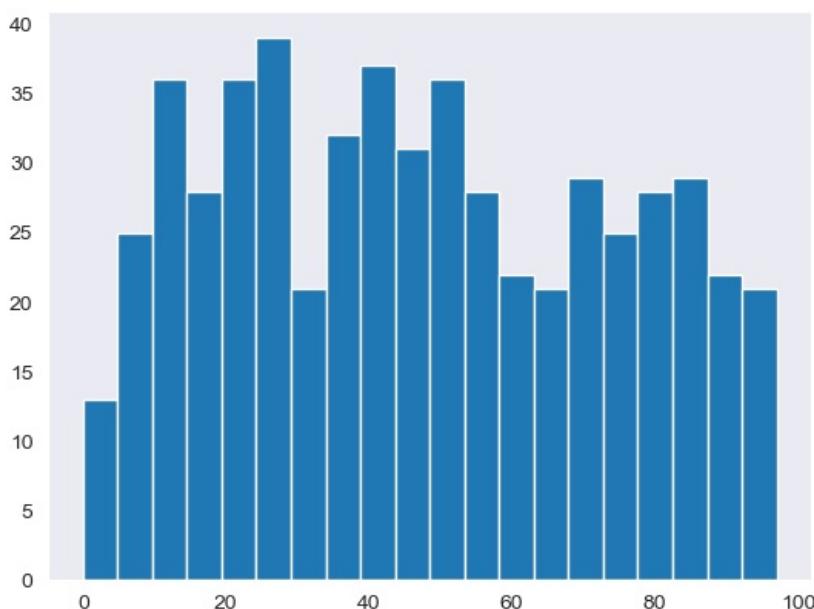
```
In [36]: m2=plt.hist(movies.AudienceRating,bins=17)
```



```
In [37]: sns.set_style('dark')
n1=plt.hist(movies.AudienceRating,bins=17)
```

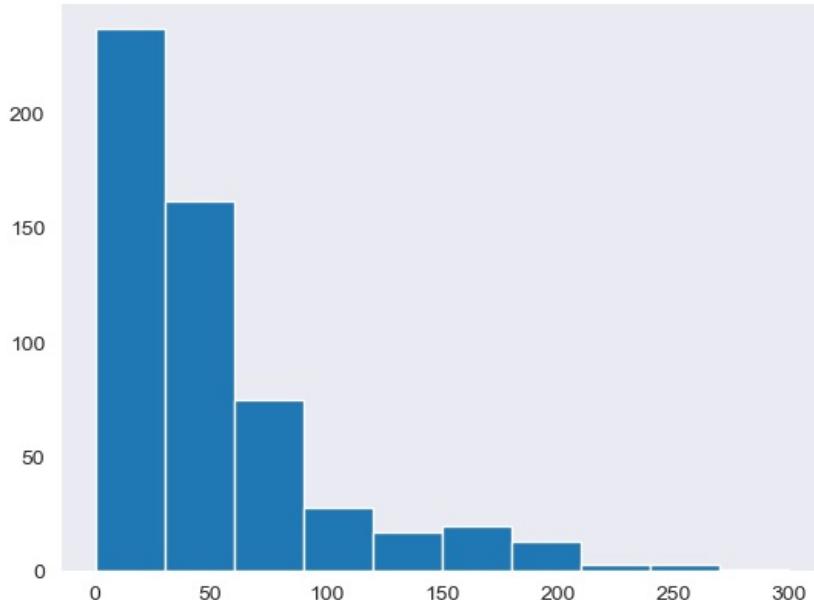


```
In [38]: n1=plt.hist(movies.CriticRating,bins=20)
```



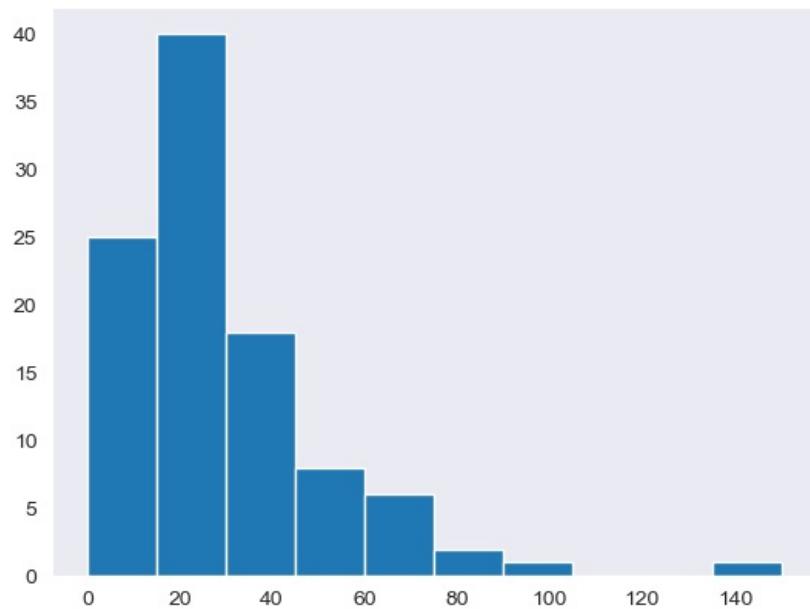
```
In [39]: plt.hist(movies.BudgetMillions)
```

```
Out[39]: (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 [40]: plt.hist(movies[movies.Genre=='Drama'].BudgetMillions)
```

```
Out[40]: (array([25., 40., 18., 8., 6., 2., 1., 0., 0., 1.]),
 array([ 0., 15., 30., 45., 60., 75., 90., 105., 120., 135., 150.]),
 <BarContainer object of 10 artists>)
```



In [41]: movies

	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 [42]: movies.head()

	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

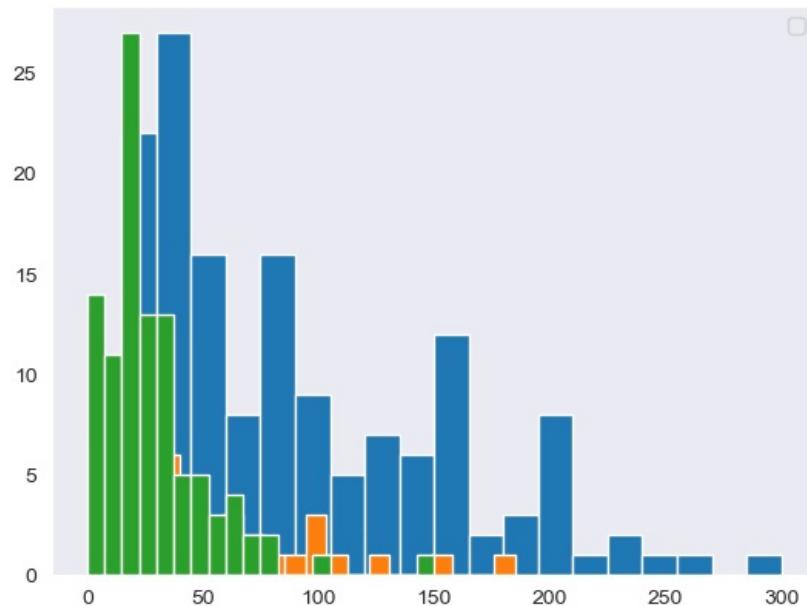
In [43]: `movies.Genre.unique()`

Out[43]: `['Comedy', 'Adventure', 'Action', 'Horror', 'Drama', 'Romance', 'Thriller']`  
`Categories (7, object): ['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance', 'Thriller']`

In [44]: `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()`

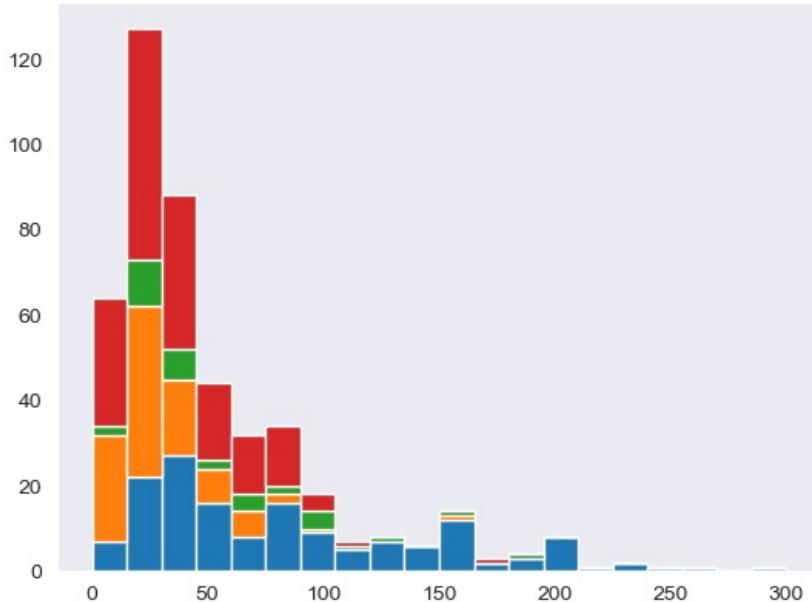
No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.

Out[44]: `<matplotlib.legend.Legend at 0x25027fd10a0>`



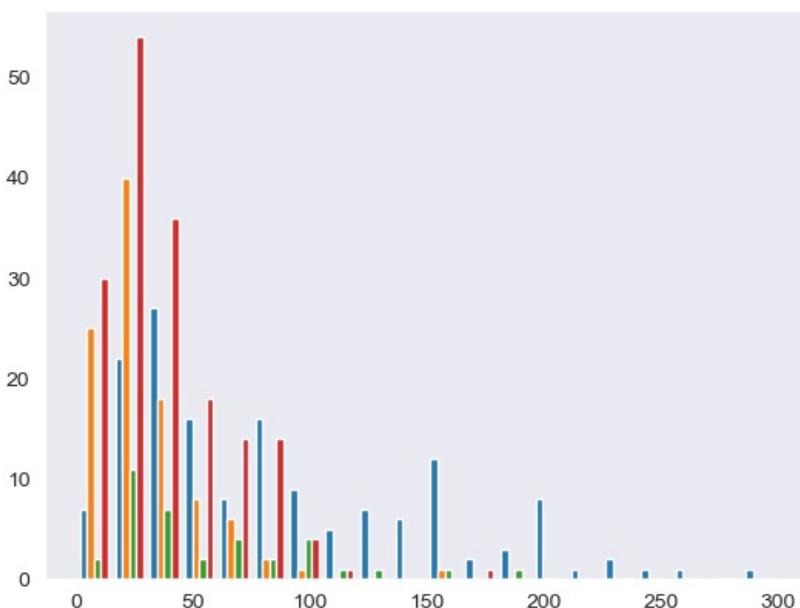
In [45]: `plt.hist([movies[movies.Genre=='Action'].BudgetMillions,\n movies[movies.Genre=='Drama'].BudgetMillions,\n movies[movies.Genre=='Thriller'].BudgetMillions,\n movies[movies.Genre=='Comedy'].BudgetMillions],\n bins=20, stacked=True)`

Out[45]: `(array([[ 7., 22., 27., 16., 8., 16., 9., 5., 7., 6., 12.,\n 2., 3., 8., 1., 2., 1., 1., 0., 1.],\n [ 32., 62., 45., 24., 14., 18., 10., 5., 7., 6., 13.,\n 2., 3., 8., 1., 2., 1., 1., 0., 1.],\n [ 34., 73., 52., 26., 18., 20., 14., 6., 8., 6., 14.,\n 2., 4., 8., 1., 2., 1., 1., 0., 1.],\n [ 64., 127., 88., 44., 32., 34., 18., 7., 8., 6., 14.,\n 3., 4., 8., 1., 2., 1., 1., 0., 1.]]),\n array([ 0., 15., 30., 45., 60., 75., 90., 105., 120., 135., 150.,\n 165., 180., 195., 210., 225., 240., 255., 270., 285., 300.]),\n <a list of 4 BarContainer objects>)`



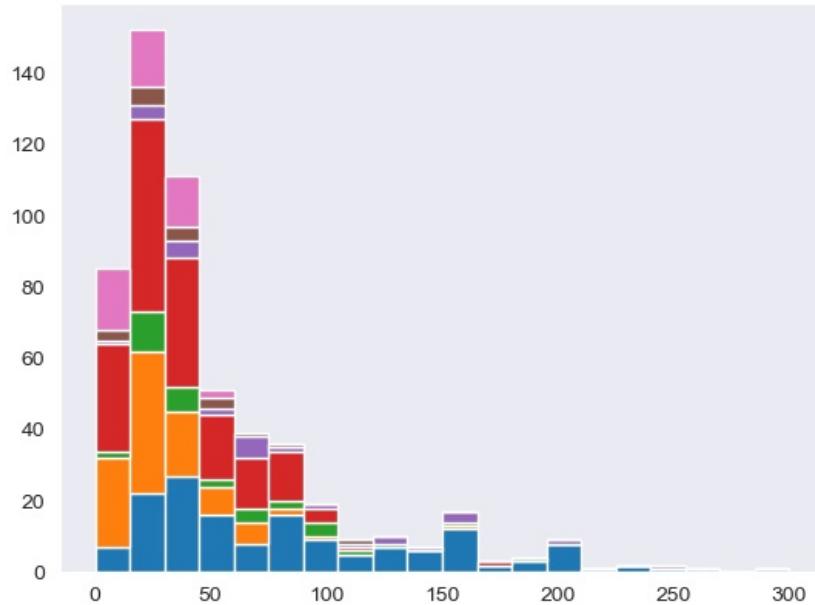
```
In [46]: plt.hist([movies[movies.Genre=='Action'].BudgetMillions,
               movies[movies.Genre=='Drama'].BudgetMillions,
               movies[movies.Genre=='Thriller'].BudgetMillions,
               movies[movies.Genre=='Comedy'].BudgetMillions],
               bins=20, stacked=False)
```

```
Out[46]: (array([[ 7.,  22.,  27.,  16.,   8.,  16.,   9.,   5.,
                   7.,   6.,  12.,   2.,   3.,
                   8.,   1.,   2.,   1.,   1.,   0.,   1.],
                  [25.,  40.,  18.,   8.,   6.,   2.,   1.,   0.,
                   0.,   0.,   0.,   0.,   0.,   0.,   0.,
                   0.,   0.,   0.,   0.,   0.,   0.,   0.],
                  [ 2.,  11.,   7.,   2.,   4.,   2.,   4.,   1.,
                   1.,   0.,   1.,   0.,   1.,   0.,   1.,
                   0.,   0.,   0.,   0.,   0.,   0.,   0.],
                  [30.,  54.,  36.,  18.,  14.,  14.,   4.,   1.,
                   0.,   0.,   0.,   0.,   0.,   0.,   0.,
                   0.,   0.,   0.,   0.,   0.,   0.,   0.]]),
 array([ 0.,  15.,  30.,  45.,  60.,  75.,  90., 105.,
        120., 135., 150.,
        165., 180., 195., 210., 225., 240., 255., 270.,
        285., 300.]),
 <a list of 4 BarContainer objects>)
```



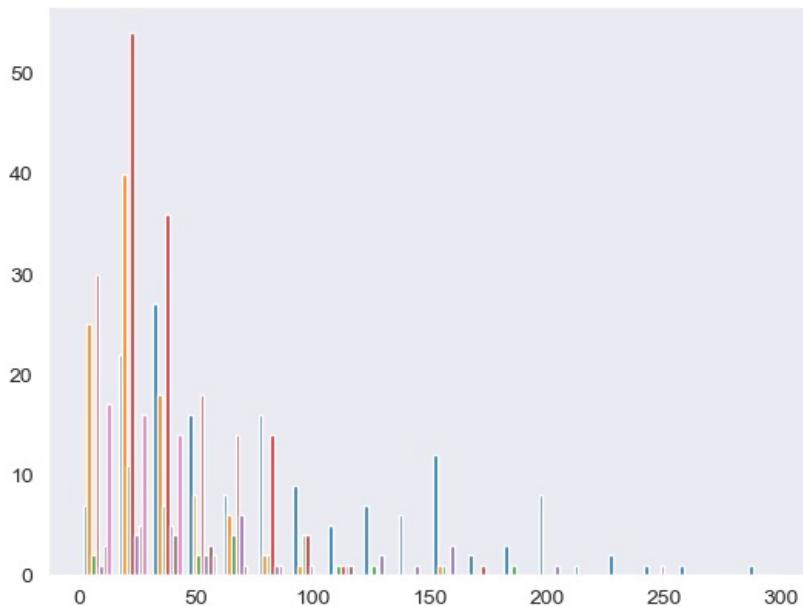
```
In [47]: plt.hist([movies[movies.Genre=='Action'].BudgetMillions,\n    movies[movies.Genre=='Drama'].BudgetMillions,\n    movies[movies.Genre=='Thriller'].BudgetMillions,\n    movies[movies.Genre=='Comedy'].BudgetMillions,\n    movies[movies.Genre=='Adventure'].BudgetMillions,\n    movies[movies.Genre=='Romance'].BudgetMillions,\n    movies[movies.Genre=='Horror'].BudgetMillions],\nbins=20,stacked=True)
```

```
Out[47]: (array([[ 7.,  22.,  27.,  16.,  8.,  16.,  9.,  5.,  7.,  6.,  12.,\n        2.,  3.,  8.,  1.,  2.,  1.,  1.,  0.,  1.],\n       [ 32.,  62.,  45.,  24.,  14.,  18.,  10.,  5.,  7.,  6.,  13.,\n        2.,  3.,  8.,  1.,  2.,  1.,  1.,  0.,  1.],\n       [ 34.,  73.,  52.,  26.,  18.,  20.,  14.,  6.,  8.,  6.,  14.,\n        2.,  4.,  8.,  1.,  2.,  1.,  1.,  0.,  1.],\n       [ 64.,  127.,  88.,  44.,  32.,  34.,  18.,  7.,  8.,  6.,  14.,\n        3.,  4.,  8.,  1.,  2.,  1.,  1.,  0.,  1.],\n       [ 65.,  131.,  93.,  46.,  38.,  35.,  19.,  8.,  10.,  7.,  17.,\n        3.,  4.,  9.,  1.,  2.,  2.,  1.,  0.,  1.],\n       [ 68.,  136.,  97.,  49.,  39.,  36.,  19.,  9.,  10.,  7.,  17.,\n        3.,  4.,  9.,  1.,  2.,  2.,  1.,  0.,  1.],\n       [ 85.,  152.,  111.,  51.,  39.,  36.,  19.,  9.,  10.,  7.,  17.,\n        3.,  4.,  9.,  1.,  2.,  2.,  1.,  0.,  1.]]),\n      array([ 0.,  15.,  30.,  45.,  60.,  75.,  90.,  105.,  120.,  135.,  150.,\n             165.,  180.,  195.,  210.,  225.,  240.,  255.,  270.,  285.,  300.]),\n      <a list of 7 BarContainer objects>)
```



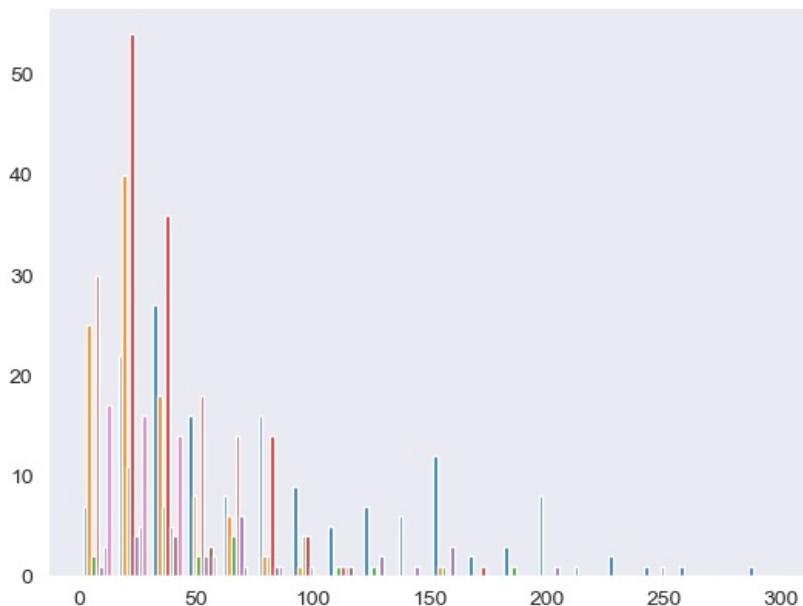
```
In [48]: plt.hist([movies[movies.Genre=='Action'].BudgetMillions,\n    movies[movies.Genre=='Drama'].BudgetMillions,\n    movies[movies.Genre=='Thriller'].BudgetMillions,\n    movies[movies.Genre=='Comedy'].BudgetMillions,\n    movies[movies.Genre=='Adventure'].BudgetMillions,\n    movies[movies.Genre=='Romance'].BudgetMillions,\n    movies[movies.Genre=='Horror'].BudgetMillions],\nbins=20,stacked=False)
```

```
Out[48]: (array([[ 7.,  22.,  27.,  16.,  8.,  16.,  9.,  5.,  7.,  6.,  12.,  2.,  3.,\n        8.,  1.,  2.,  1.,  1.,  0.,  1.],\n       [ 25.,  40.,  18.,  8.,  6.,  2.,  1.,  0.,  0.,  0.,  1.,  0.,  0.,\n        0.,  0.,  0.,  0.,  0.,  0.],\n       [ 2.,  11.,  7.,  2.,  4.,  2.,  4.,  1.,  1.,  0.,  1.,  0.,  1.,\n        0.,  0.,  0.,  0.,  0.,  0.],\n       [ 30.,  54.,  36.,  18.,  14.,  14.,  4.,  1.,  0.,  0.,  0.,  1.,  0.,\n        0.,  0.,  0.,  0.,  0.,  0.],\n       [ 1.,  4.,  5.,  2.,  6.,  1.,  1.,  1.,  2.,  1.,  3.,  0.,  0.,\n        1.,  0.,  0.,  1.,  0.,  0.],\n       [ 3.,  5.,  4.,  3.,  1.,  1.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,\n        0.,  0.,  0.,  0.,  0.,  0.],\n       [ 17.,  16.,  14.,  2.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,\n        0.,  0.,  0.,  0.,  0.,  0.]]),\n      array([ 0.,  15.,  30.,  45.,  60.,  75.,  90.,  105.,  120.,  135.,  150.,\n             165.,  180.,  195.,  210.,  225.,  240.,  255.,  270.,  285.,  300.]),\n      <a list of 7 BarContainer objects>)
```



```
In [49]: 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             movies[movies.Genre=='Adventure'].BudgetMillions,\n             movies[movies.Genre=='Romance'].BudgetMillions,\n             movies[movies.Genre=='Horror'].BudgetMillions],\nbins=20)
```

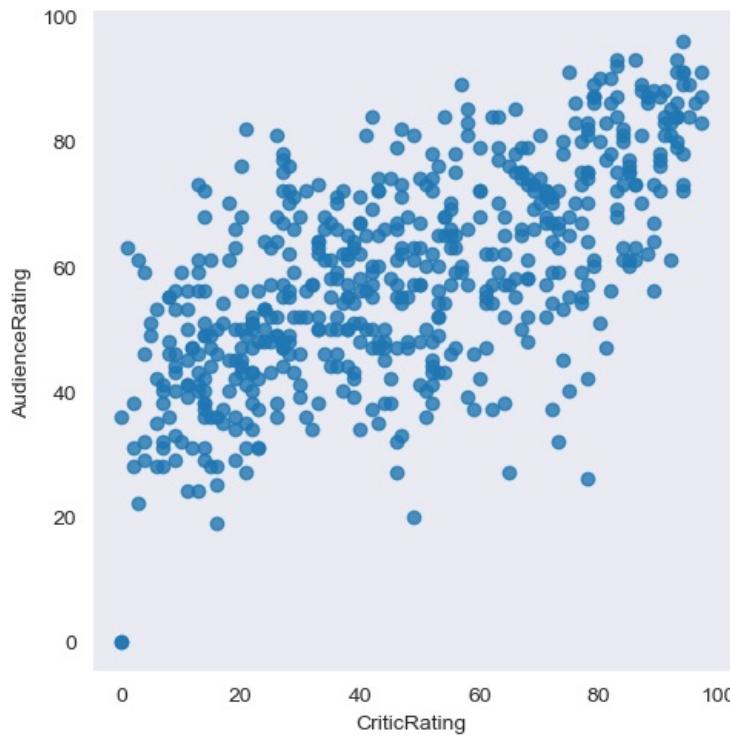
```
Out[49]: (array([[ 7., 22., 27., 16., 8., 16., 9., 5., 7., 6., 12., 2., 3.,\n                 8., 1., 2., 1., 1., 0., 1.],\n                [25., 40., 18., 8., 6., 2., 1., 0., 0., 0., 1., 0., 0.,\n                 0., 0., 0., 0., 0., 0.],\n                [ 2., 11., 7., 2., 4., 2., 4., 1., 1., 0., 1., 0., 1.,\n                 0., 0., 0., 0., 0., 0.],\n                [30., 54., 36., 18., 14., 14., 4., 1., 0., 0., 0., 1., 0.,\n                 0., 0., 0., 0., 0., 0.],\n                [ 1., 4., 5., 2., 6., 1., 1., 1., 2., 1., 3., 0., 0.,\n                 1., 0., 0., 1., 0., 0.],\n                [ 3., 5., 4., 3., 1., 1., 0., 1., 0., 0., 0., 0., 0.,\n                 0., 0., 0., 0., 0., 0.],\n                [17., 16., 14., 2., 0., 0., 0., 0., 0., 0., 0., 0., 0.,\n                 0., 0., 0., 0., 0., 0.]]),\narray([ 0., 15., 30., 45., 60., 75., 90., 105., 120., 135., 150.,\n       165., 180., 195., 210., 225., 240., 255., 270., 285., 300.]),\n<a list of 7 BarContainer objects>)
```



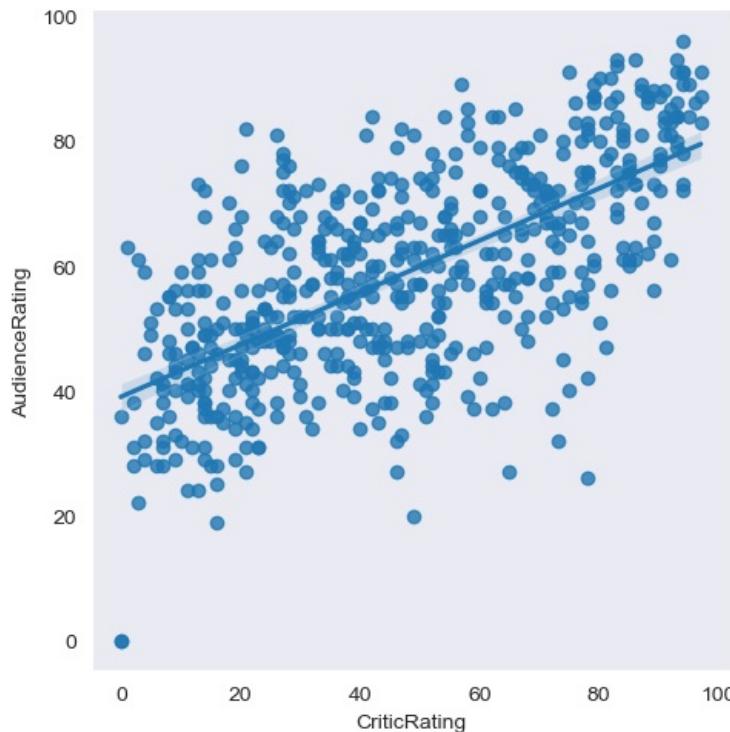
```
In [50]: for gen in movies.Genre.cat.categories:  
    print(gen)
```

Action  
Adventure  
Comedy  
Drama  
Horror  
Romance  
Thriller

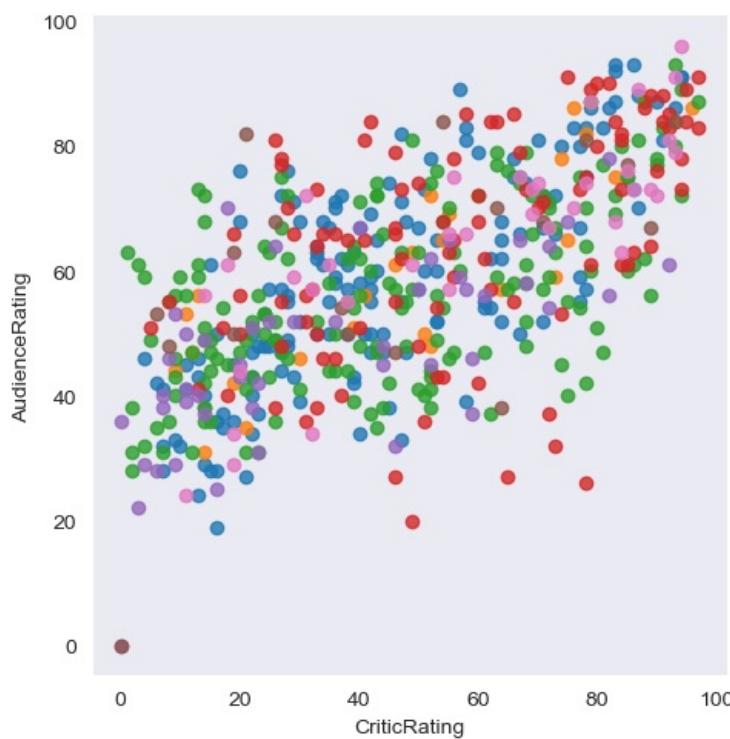
```
In [51]: vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',  
                     fit_reg=False)
```



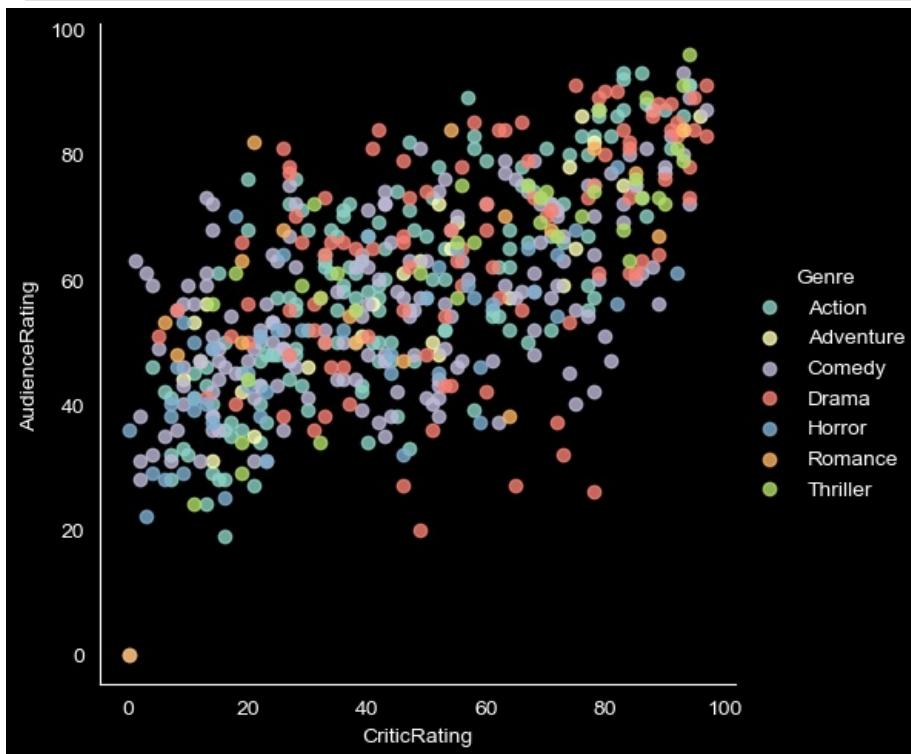
```
In [52]: vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',fit_reg=True)
```



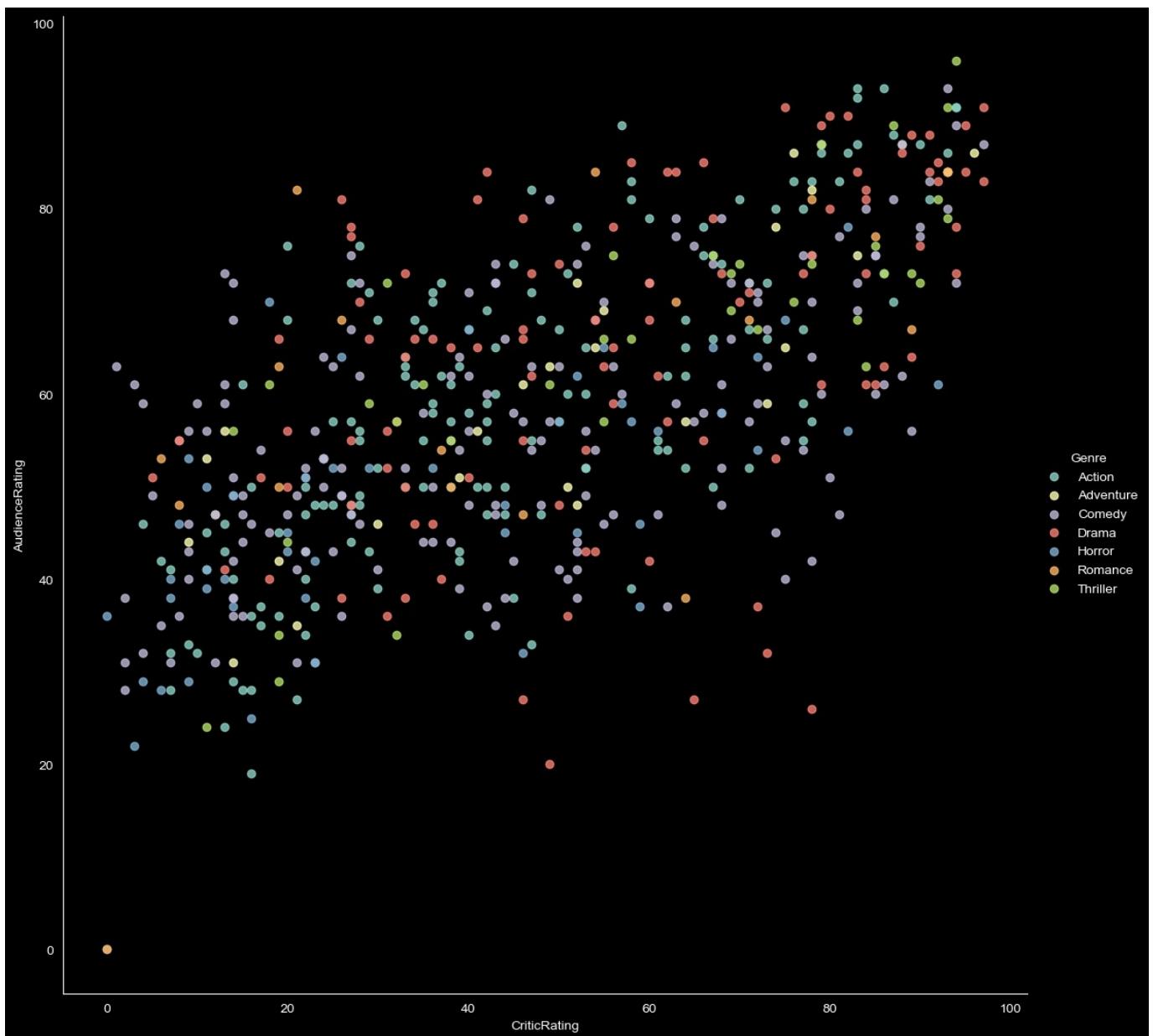
```
In [53]: vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',fit_reg=False,hue='Genre')
```



```
In [54]: plt.style.use('dark_background')
vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',fit_reg=False,hue='Genre')
```



```
In [55]: plt.style.use('dark_background')
vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',fit_reg=False,hue='Genre',height=11,aspect=1)
```

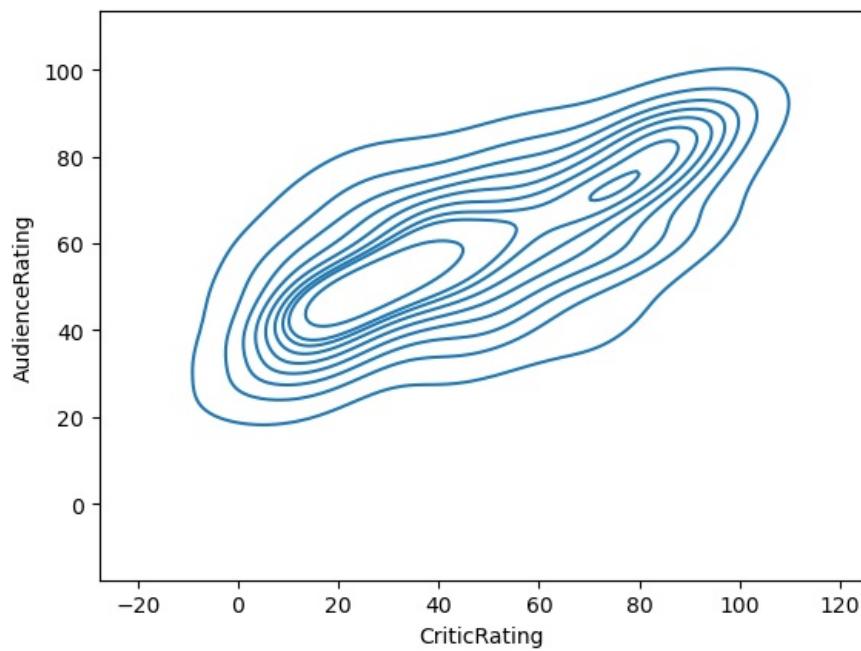


KDE PLOT---

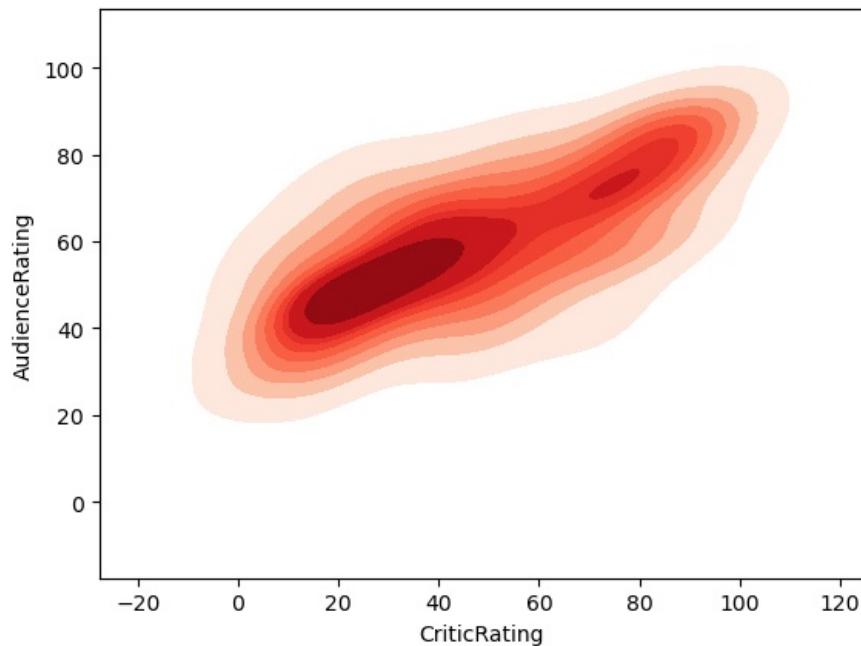
KDE=KERNEL DENSITY ESTIMATE PLOT

```
In [57]: plt.style.use('default')
```

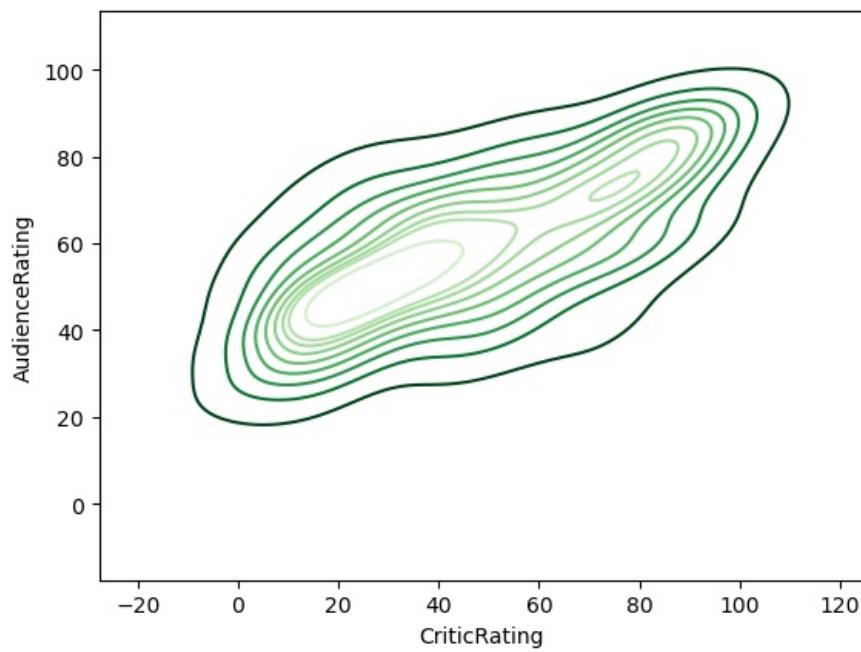
```
In [58]: k1=sns.kdeplot(x='CriticRating',y='AudienceRating',data=movies)
```



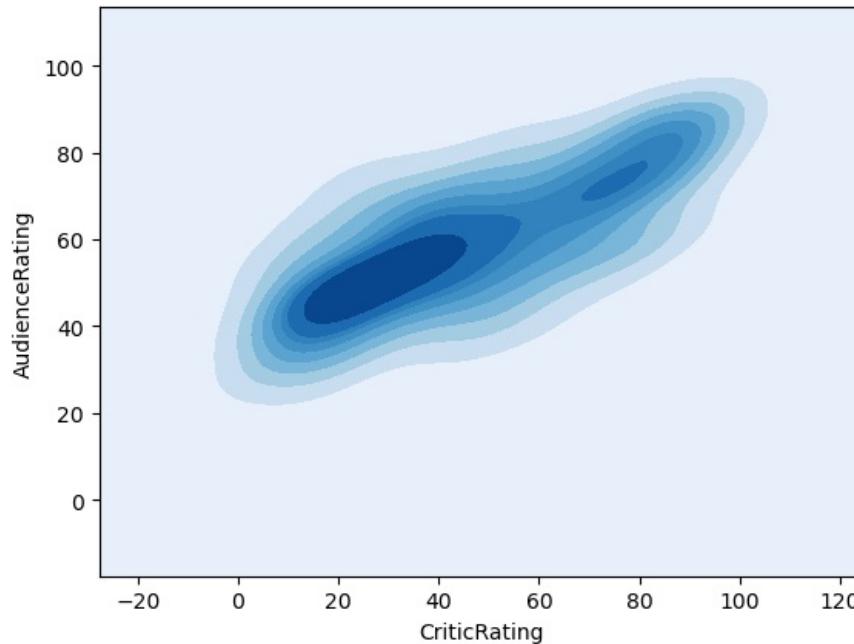
```
In [59]: k1=sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating',shade=True,shade_lowest=False,cmap='Reds')
```



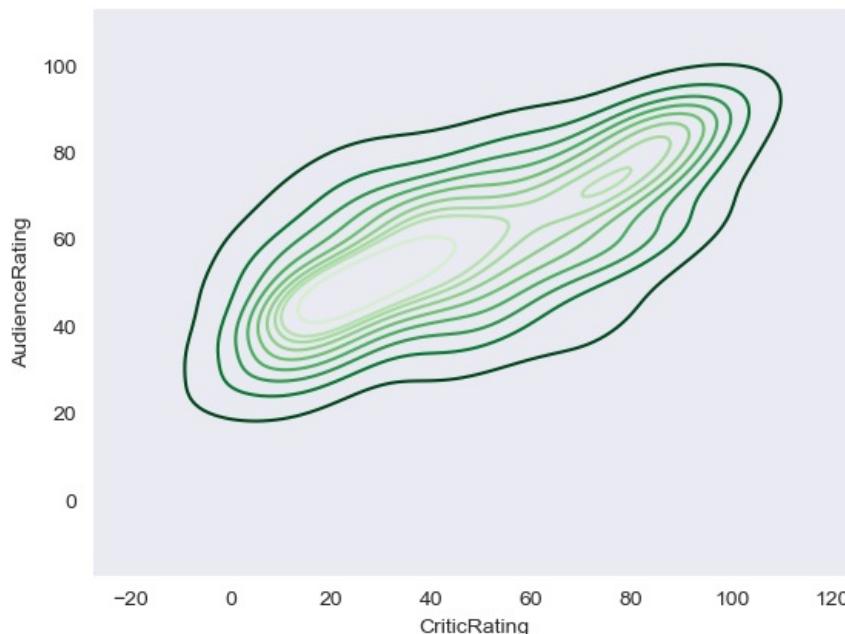
```
In [60]: k2 = sns.kdeplot(data = movies, x= 'CriticRating', y= 'AudienceRating',shade_lowest=False,cmap='Greens_r')
```



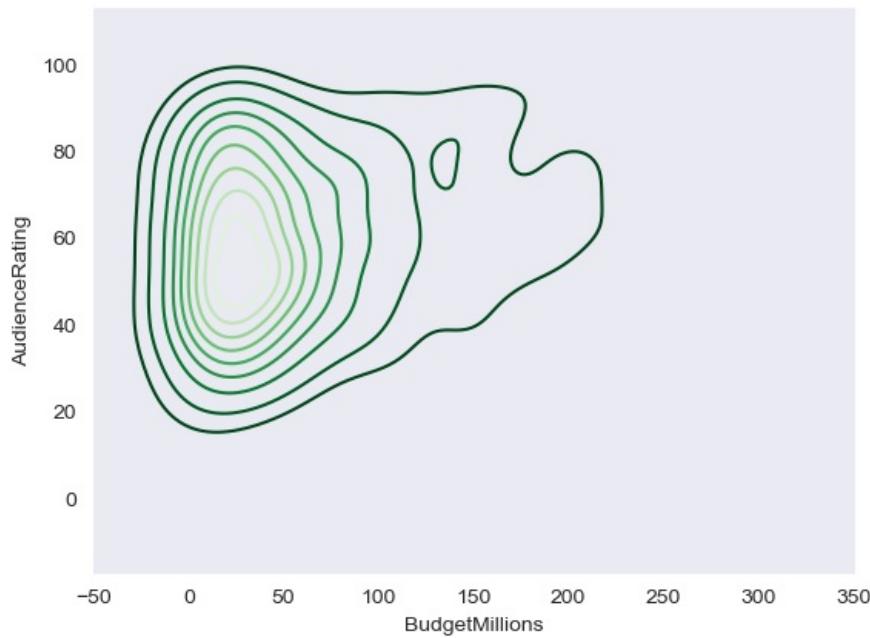
```
In [61]: k1=sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating',shade=True,shade_lowest=True,cmap='Blues')
```



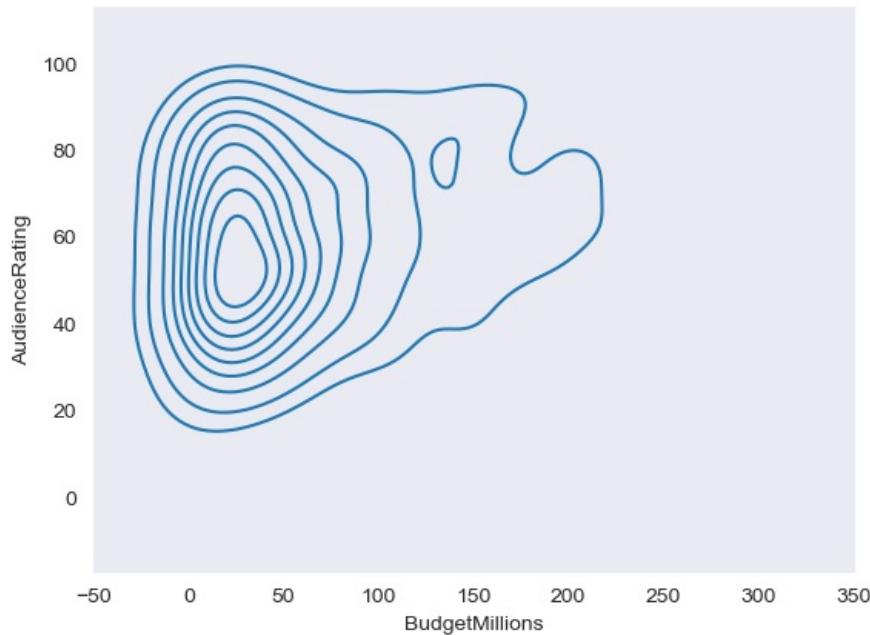
```
In [62]: sns.set_style('dark')  
k1=sns.kdeplot(data=movies,x='CriticRating',y='AudienceRating',shade_lowest=False,cmap='Greens_r')
```



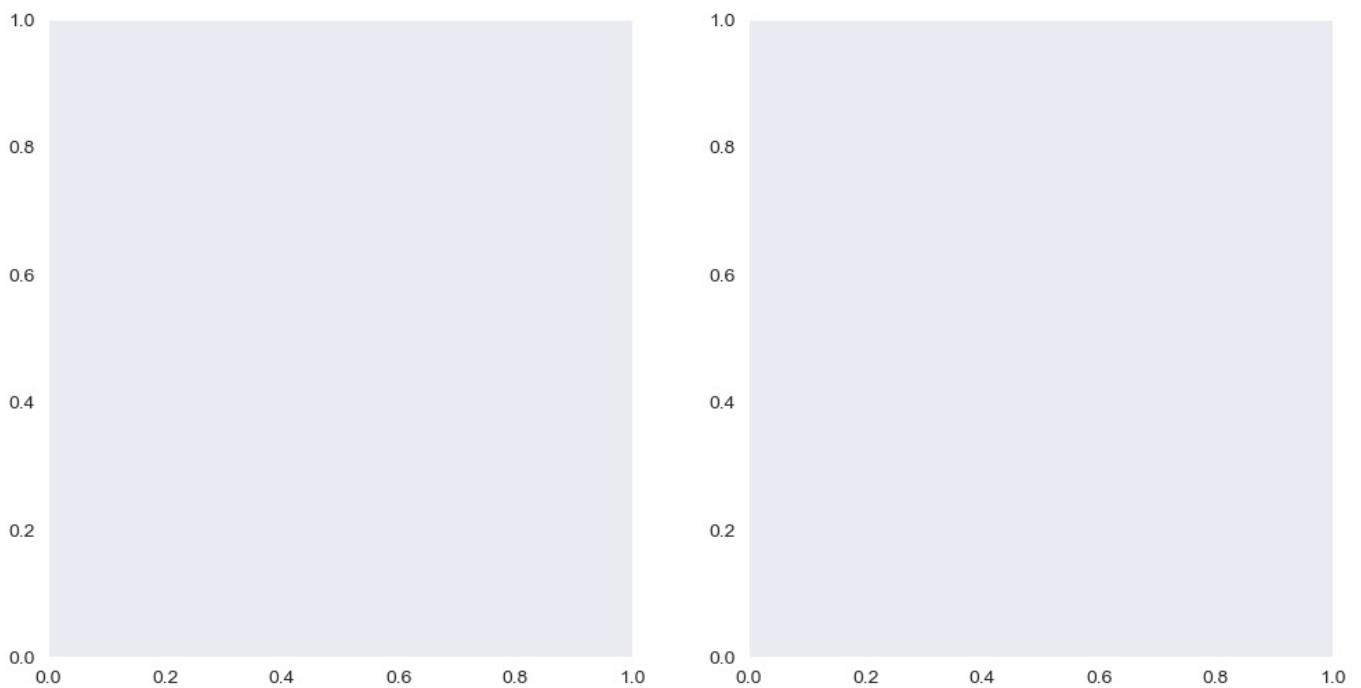
```
In [63]: sns.set_style('dark')  
k1 = sns.kdeplot(data = movies, x = 'BudgetMillions',y ='AudienceRating',shade_lowest=False,cmap='Greens_r')
```



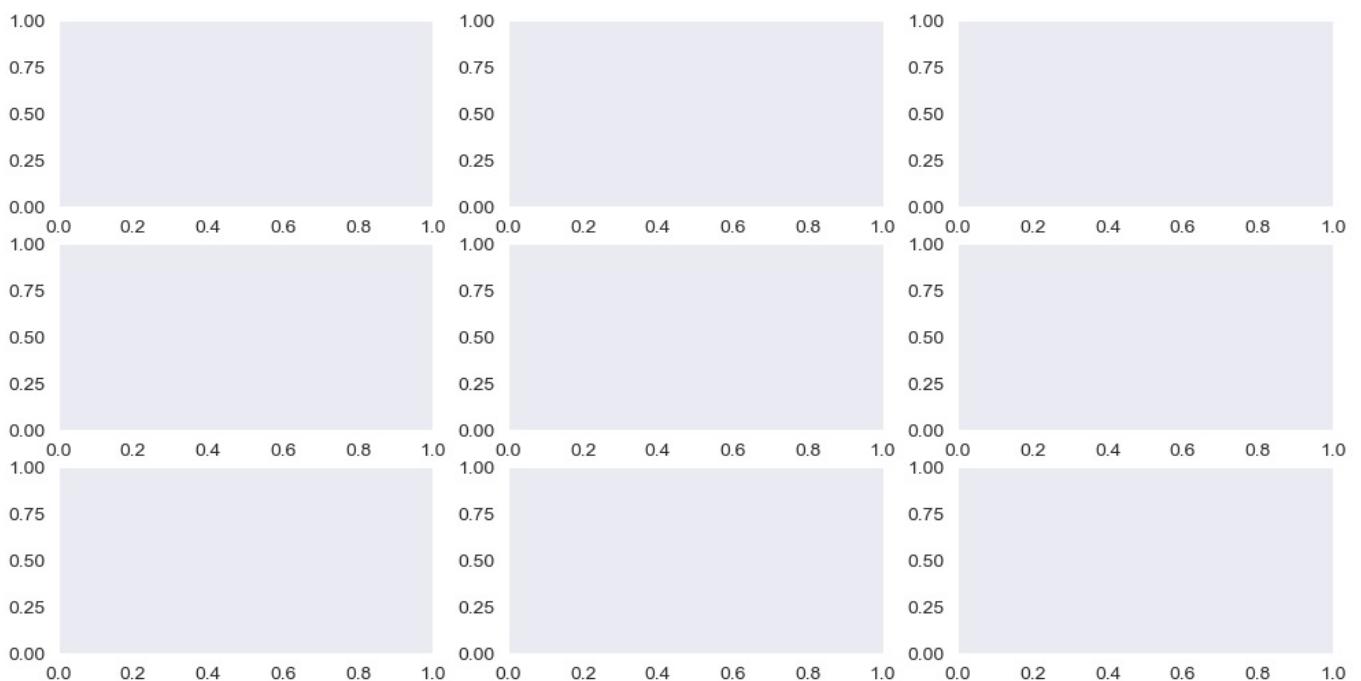
```
In [64]: sns.set_style('dark')
k1 = sns.kdeplot(data = movies, x = 'BudgetMillions',y ='AudienceRating')
```



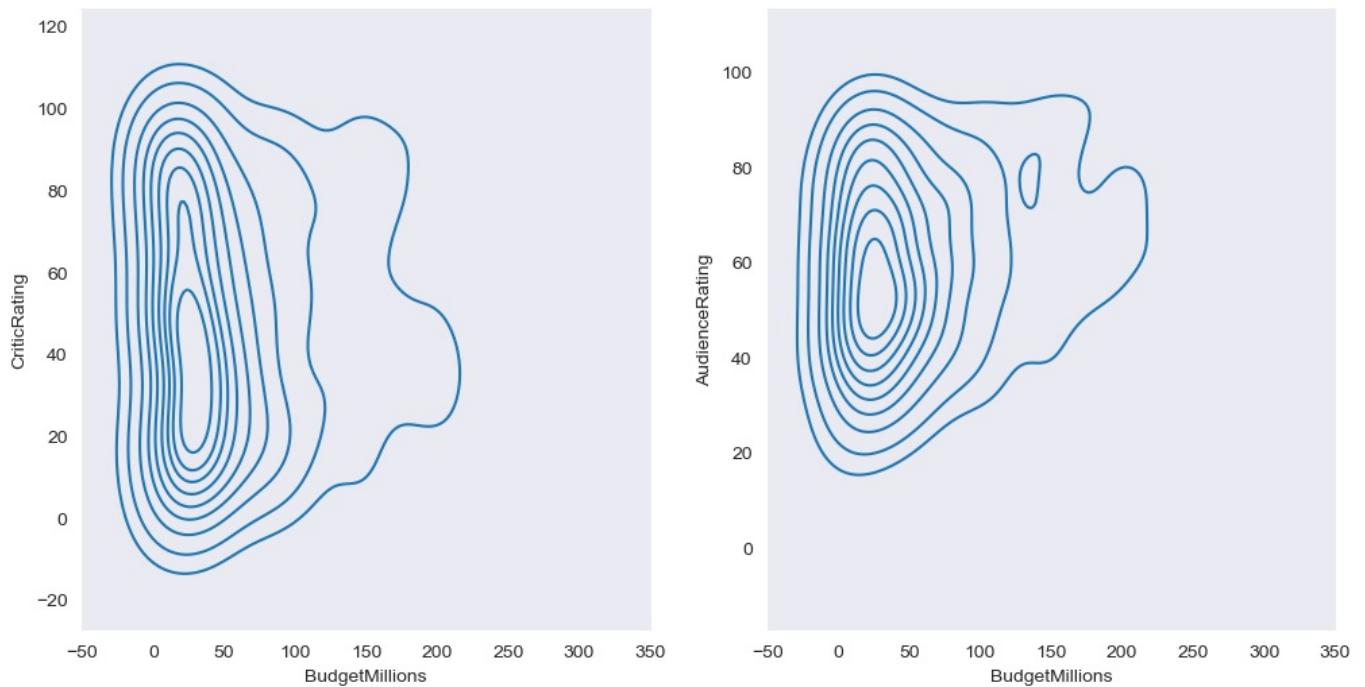
```
In [65]: f, ax=plt.subplots(1,2,figsize=(12,6))
```



```
In [66]: f,ax=plt.subplots(3,3,figsize=(12,6))
```



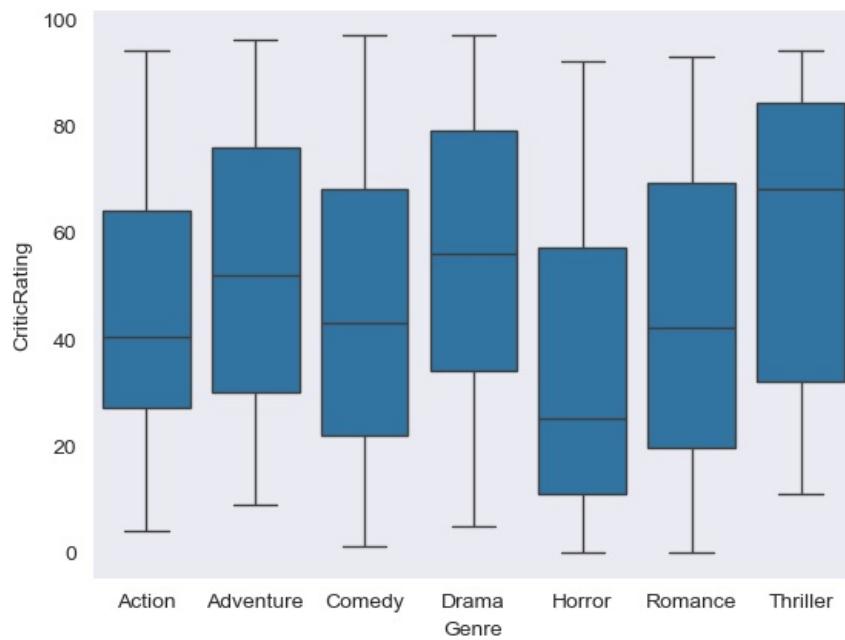
```
In [67]: f,axes=plt.subplots(1,2,figsize=(12,6))
k1=sns.kdeplot(data=movies,x='BudgetMillions',y='CriticRating',ax=axes[0])
k2=sns.kdeplot(data=movies,x='BudgetMillions',y='AudienceRating',ax=axes[1])
```



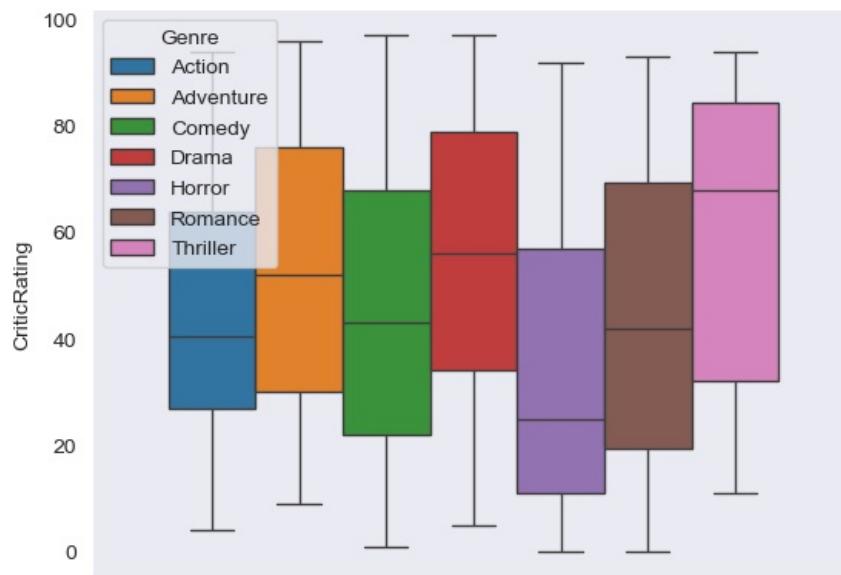
```
In [68]: axes
```

```
Out[68]: array([<Axes: xlabel='BudgetMillions', ylabel='CriticRating'>,
   <Axes: xlabel='BudgetMillions', ylabel='AudienceRating'>],
  dtype=object)
```

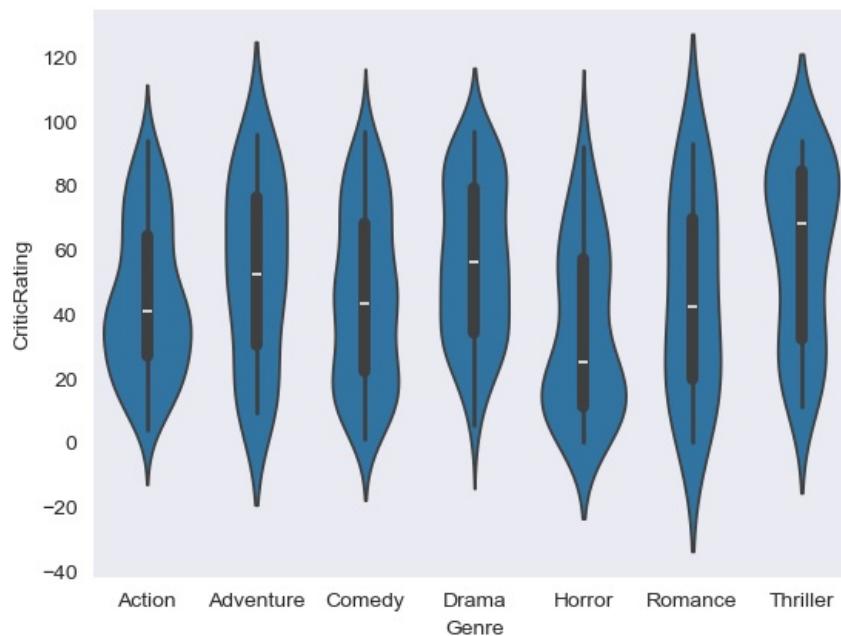
```
In [69]: #BOX PLOTS
w=sns.boxplot(data=movies,x='Genre',y='CriticRating')
```



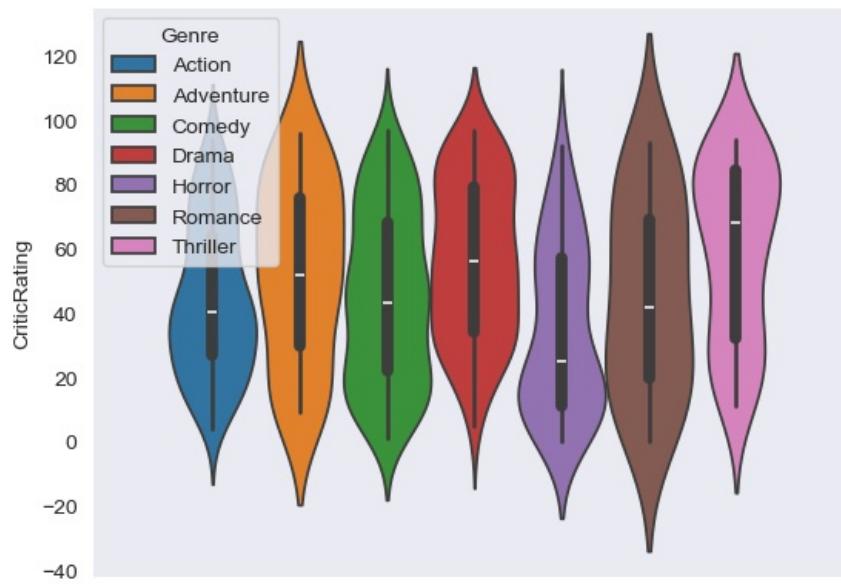
```
In [70]: w = sns.boxplot(data=movies, hue='Genre', y = 'CriticRating')
```



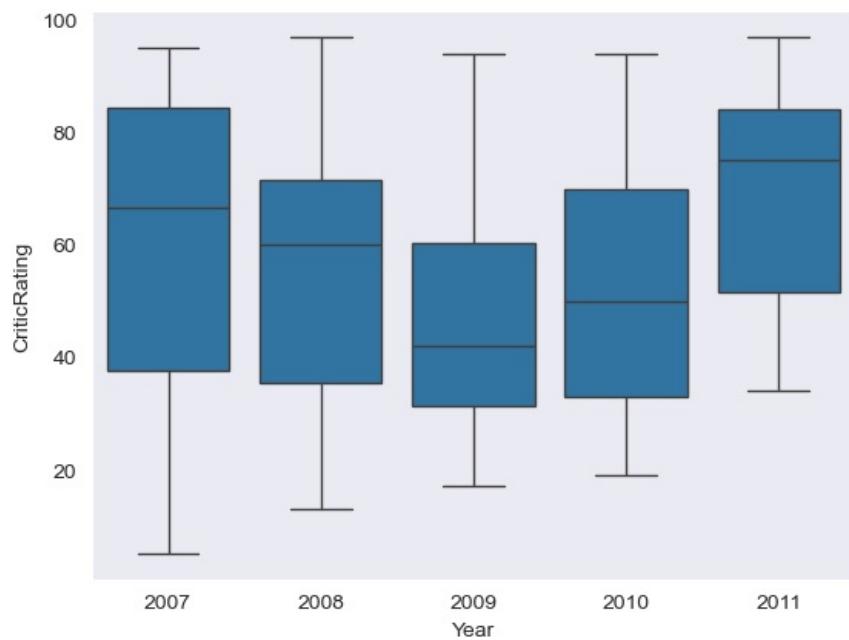
```
In [71]: #VIOLIN PLOT
z = sns.violinplot(data=movies, x='Genre', y = 'CriticRating')
```



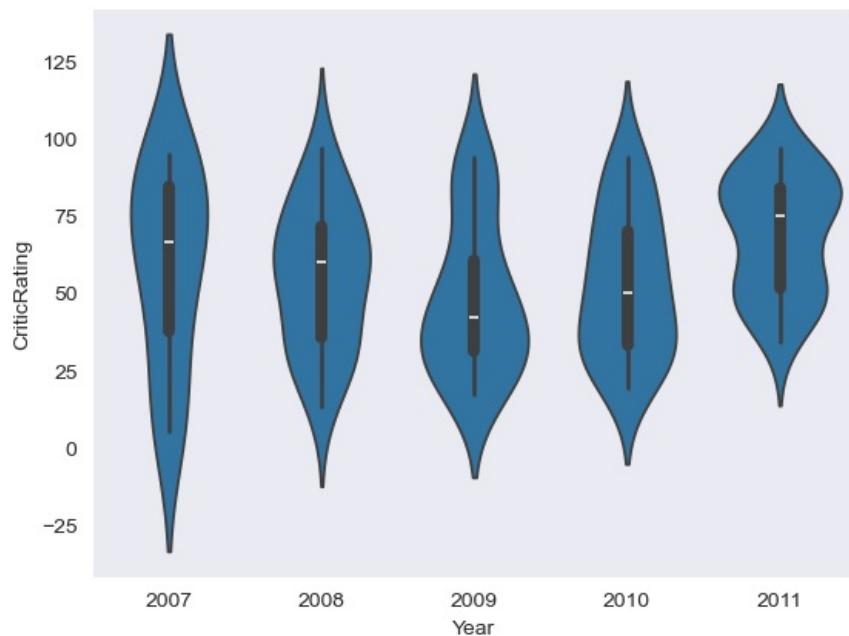
```
In [72]: z = sns.violinplot(data=movies, hue='Genre', y = 'CriticRating')
```



```
In [73]: w1 = sns.boxplot(data=movies[movies.Genre == 'Drama'], x='Year', y = 'CriticRating')
```

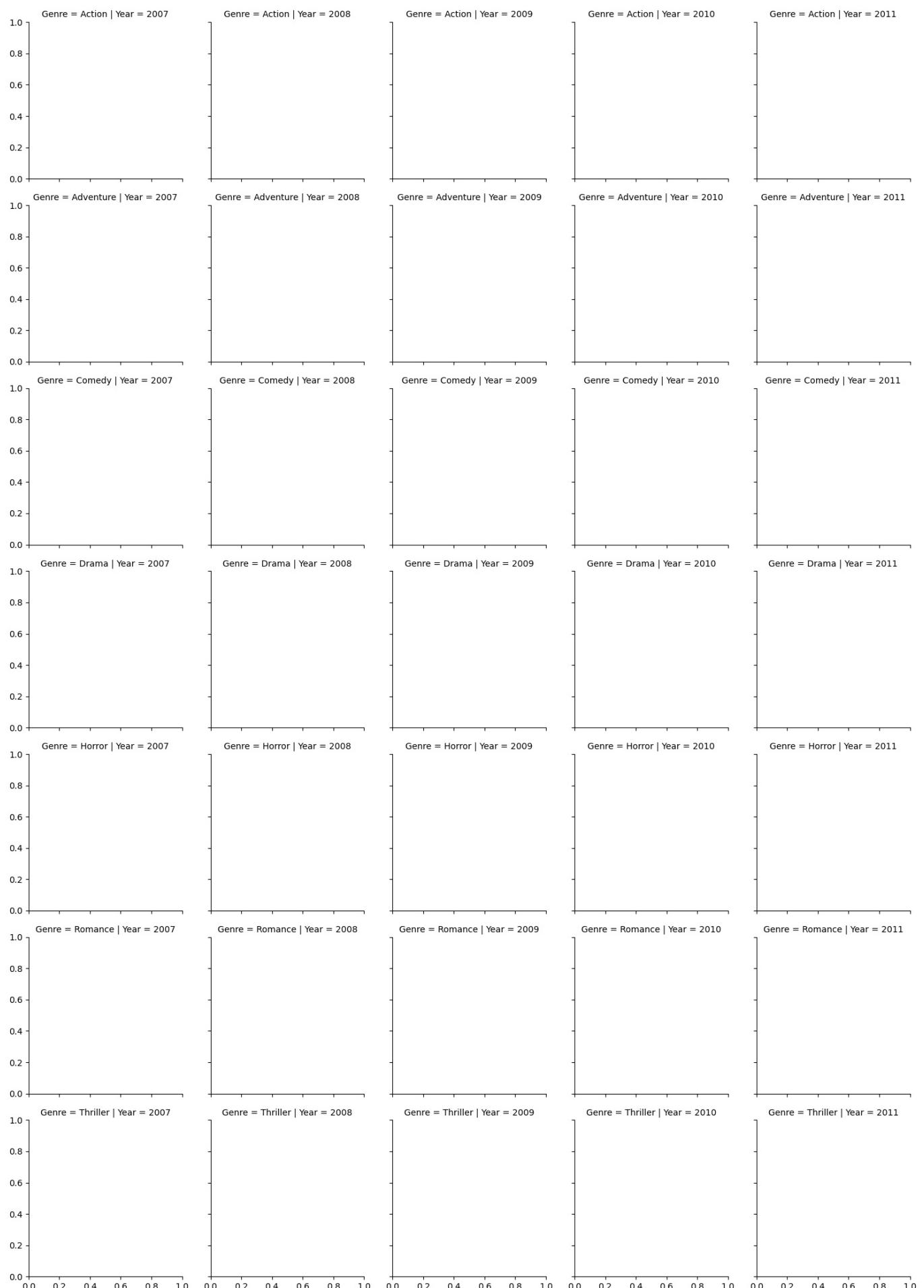


```
In [74]: w1 = sns.violinplot(data=movies[movies.Genre == 'Drama'], x='Year', y = 'CriticRating')
```



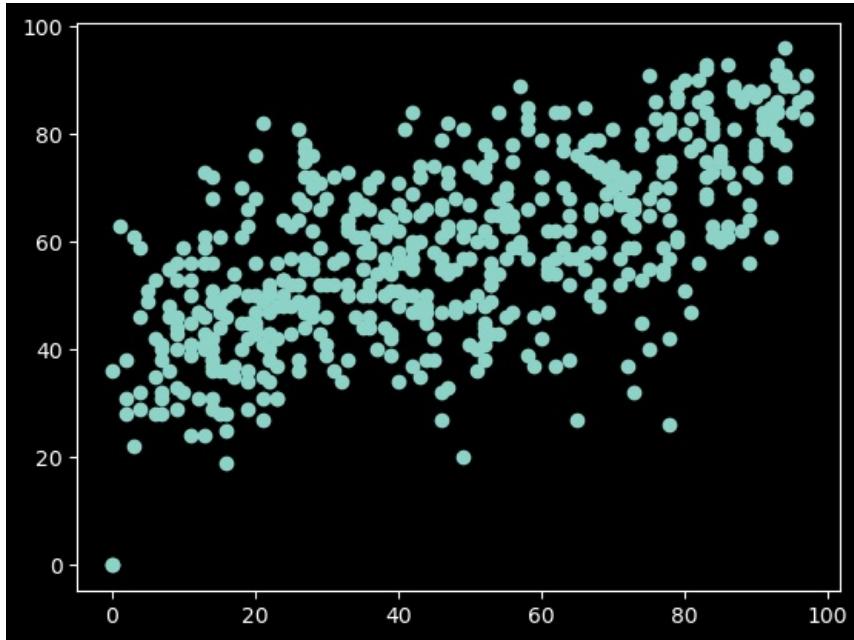
```
In [75]: plt.style.use('default')
```

```
In [76]: g=sns.FacetGrid(movies, row='Genre', col='Year', hue='Genre')
```

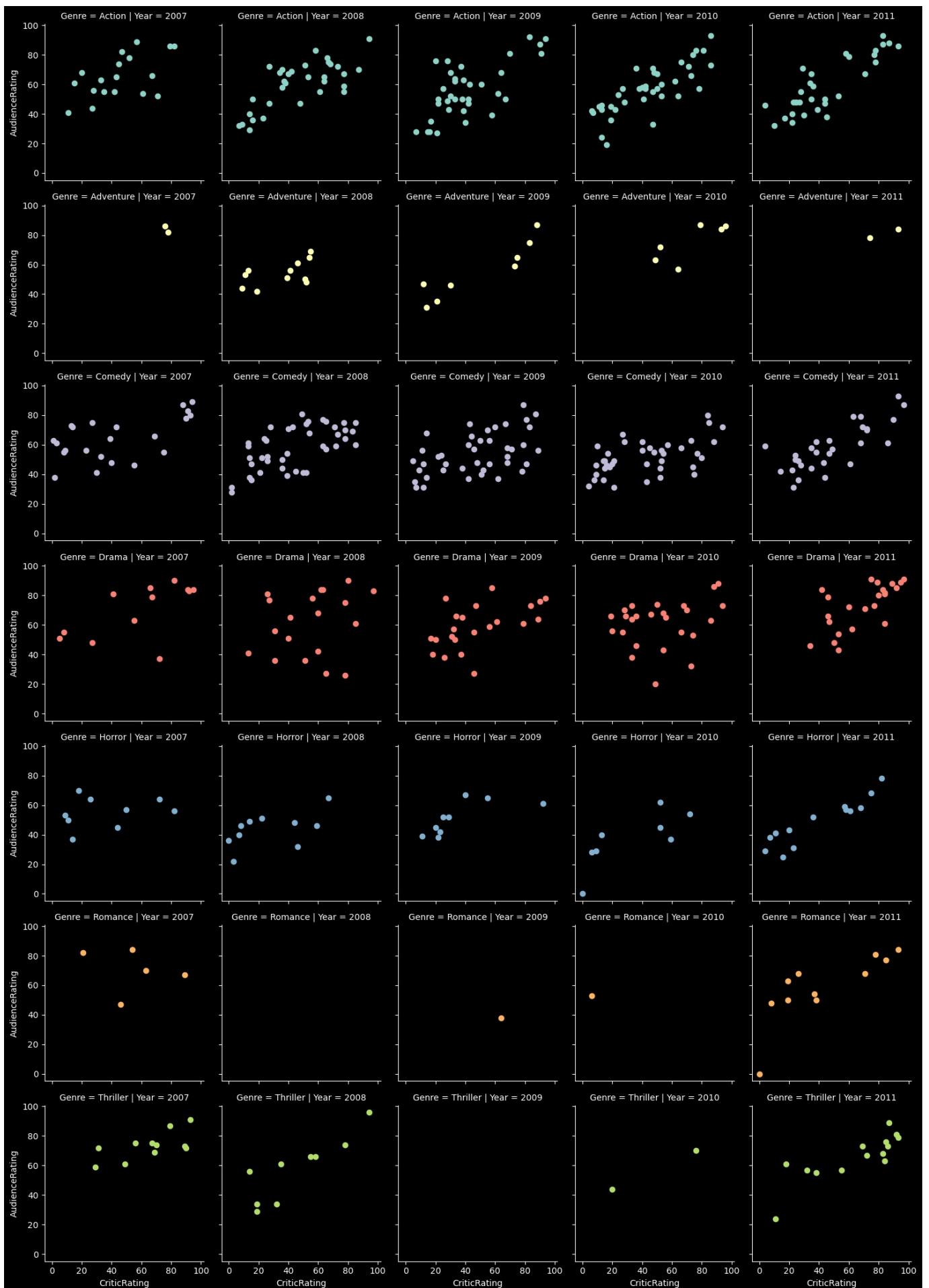


```
In [77]: plt.style.use('dark_background')
plt.scatter(movies.CriticRating,movies.AudienceRating)
```

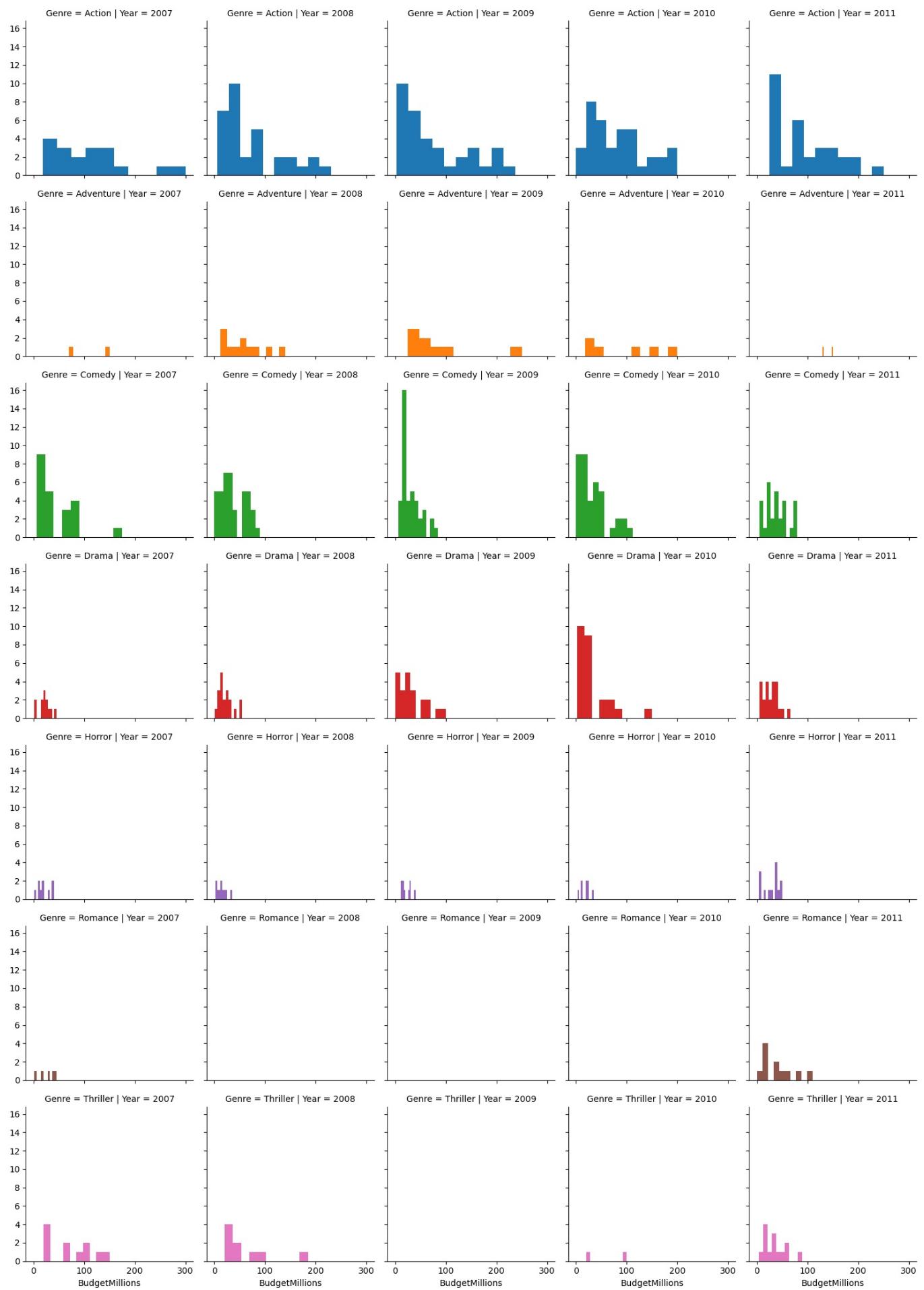
```
Out[77]: <matplotlib.collections.PathCollection at 0x2502d8f5af0>
```



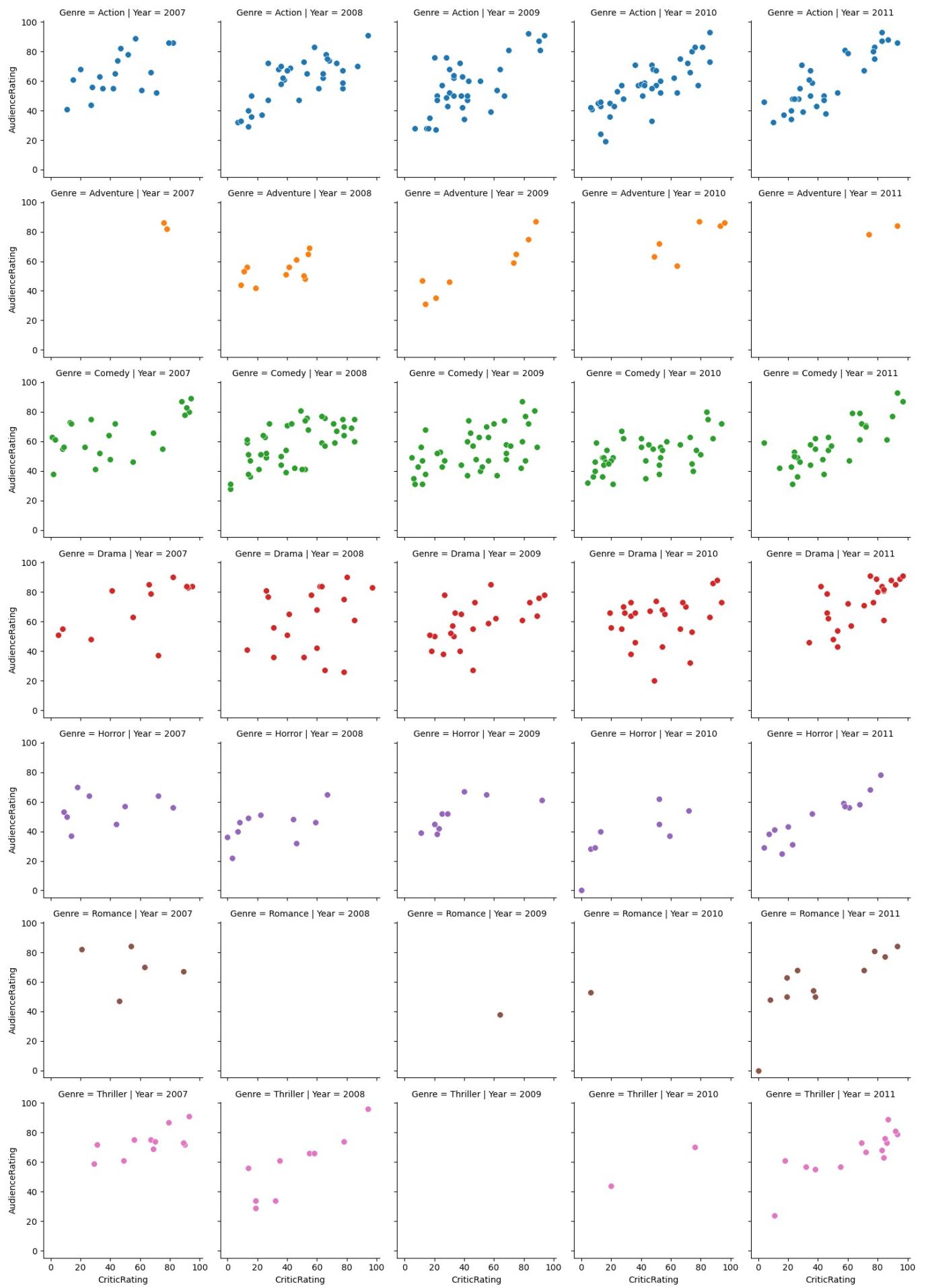
```
In [78]: plt.style.use('dark_background')
g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
g=g.map(plt.scatter,'CriticRating','AudienceRating')
```



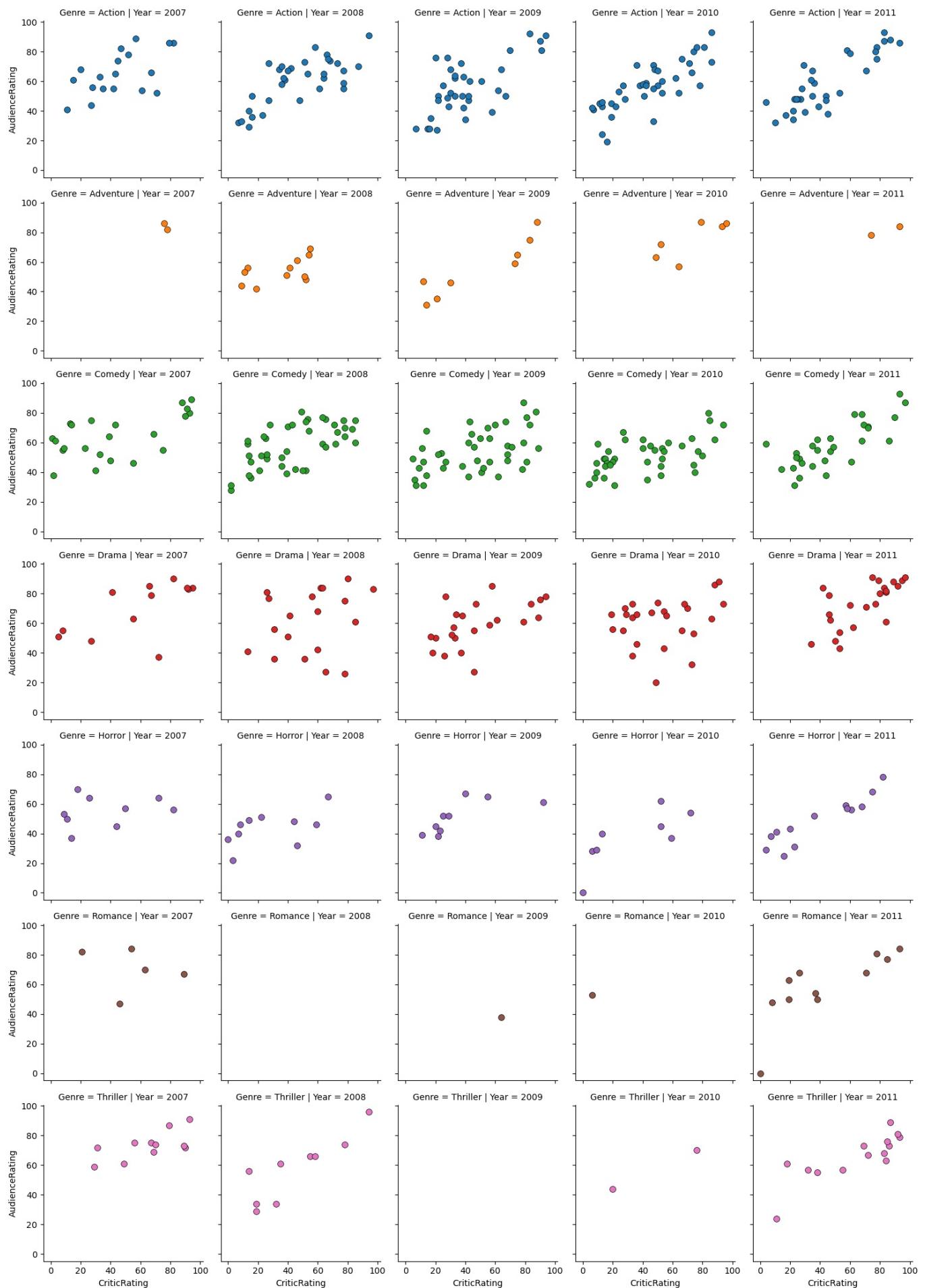
```
In [79]: plt.style.use('default')
g=sns.FacetGrid(movies, row='Genre', col='Year', hue='Genre')
g=g.map(plt.hist, 'BudgetMillions')
```



```
In [80]: g=sns.FacetGrid(movies, row='Genre', col='Year', hue='Genre')
kws=dict(s=50, linewidth=0.5, edgecolor='white')
g=g.map(plt.scatter, 'CriticRating', 'AudienceRating', **kws)
```



```
In [81]: 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 )
```



```
In [82]: sns.set_style('darkgrid')
f, axes = plt.subplots(2,2, figsize = (8,8))

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])

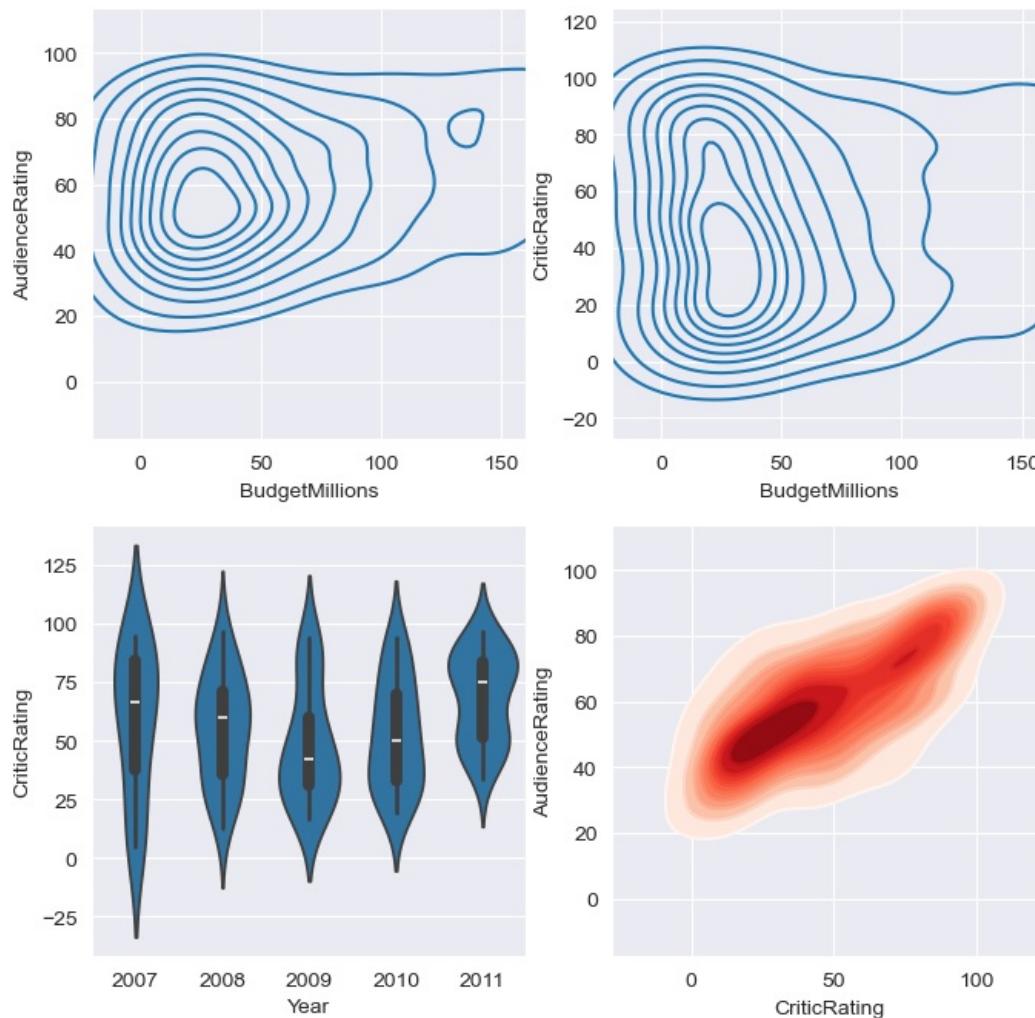
k1.set(xlim=(-20,160))
k2.set(xlim=(-20,160))
```

```

z = sns.violinplot(data=movies[movies.Genre=='Drama'], x='Year', y = 'CriticRating', ax=axes[1,0])

k4 = sns.kdeplot(x = movies.CriticRating,y=movies.AudienceRating,shade = True,shade_lowest=False,cmap='Reds',ax=k4)
k4b = sns.kdeplot(x = movies.CriticRating,y = movies.AudienceRating,cmap='Reds',ax = axes[1,1])

```



## STYLING DASHBOARD USING DIFFERENT COLOURS

```

In [84]: sns.set_style('dark', {'axes.facecolor': 'black'})
f,axes=plt.subplots(2,2,figsize=(15,15))

#plot[0,0]
k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,shade=True,shade_lowest=True,cmap='inferno',ax=axes[0,0])
k1b=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,cmap='cool',ax=axes[0,0])

#plot[0,1]
k2=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating,shade=True,shade_lowest=True,cmap='inferno',ax=axes[0,1])
k2b=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating,cmap='cool',ax=axes[0,1])

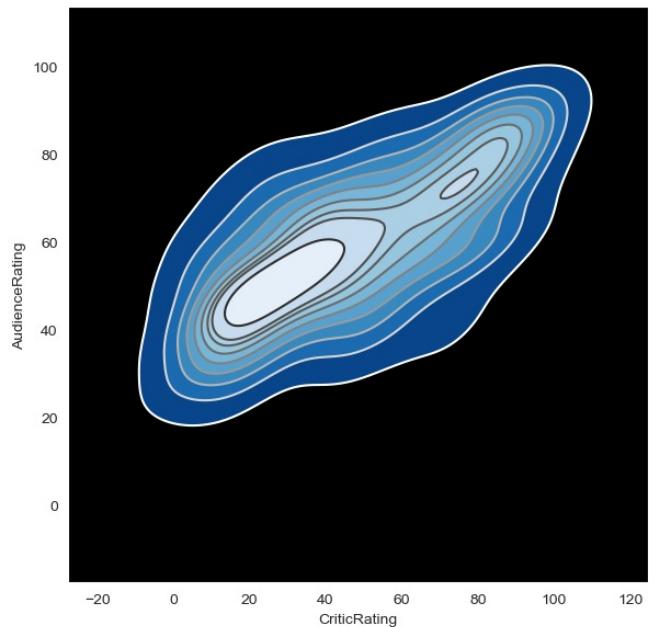
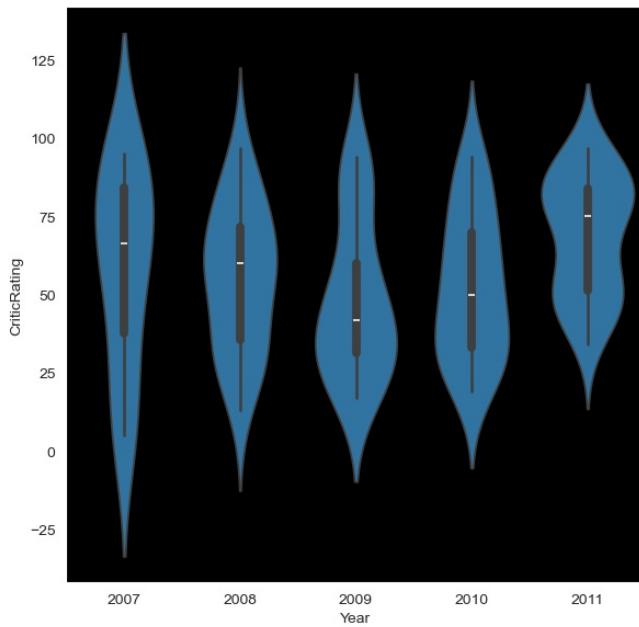
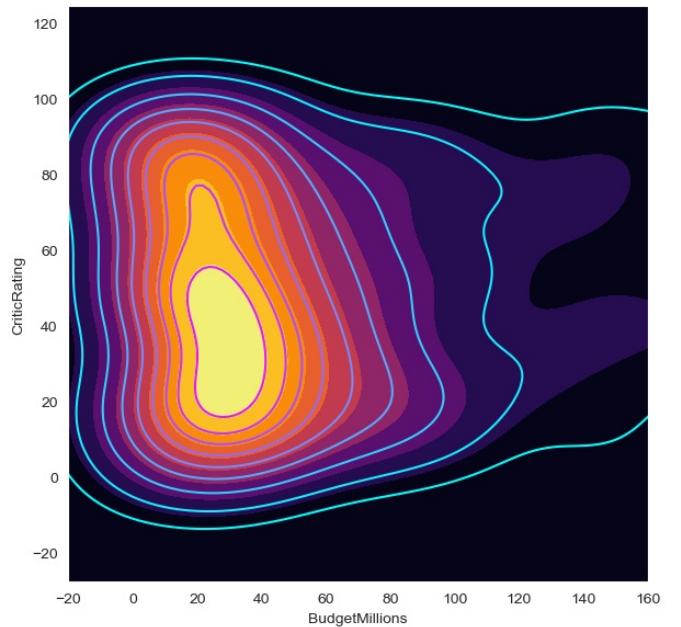
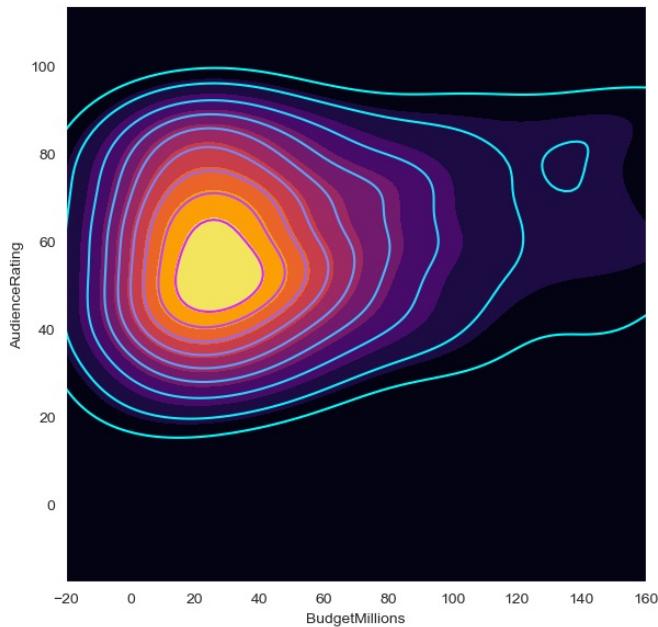
#plot[1,0]
z=sns.violinplot(data=movies[movies.Genre=='Drama'],x='Year',y='CriticRating',ax=axes[1,0])

#plot[1,1]
k4=sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,shade=True,shade_lowest=False,cmap='Blues_r',ax=axes[1,1])
k4b=sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRating,cmap='gist_gray_r',ax=axes[1,1])

k1.set(xlim=(-20,160))
k2.set(xlim=(-20,160))

```

```
Out[84]: [(-20.0, 160.0)]
```



#HERE,I USED 1- CATEGORY DATATYPE IN PYTHON 2- JOINTPLOTS 3- HIST PLOT 4- STACKED HISTOGRAMS 5- KDE PLOTS 6- SUBPLOTS  
7- VIOLIN PLOTS 8= FACET GRIDS 9- BUILDING DASHBOARDS

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js