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

In [2]:
```
movies=pd.read_csv(r'C:\Users\Rachana Jena\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]:
```
type(movies)
```

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

In [5]:
```
movies
```

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

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

In [7]:
```
import numpy
print(numpy.__version__)
```

```
1.26.4
```

In [8]:
```
movies.columns
```

Out[8]:
```
Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
       'Budget (million $)', 'Year of release'],
      dtype='object')
```

In [9]:
```
movies.columns = ['Film','Genre','CriticRating','AudienceRating','BugetMillions','Year']
```

In [10]:
```
movies.head(1)
```

Out[10]:

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

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

Out[11]:

	Film	Genre	CriticRating	AudienceRating	BugetMillions	Year
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
```

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

In [13]:
```
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   BugetMillions   559 non-null    int64 
 5   Year            559 non-null    int64 
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

In [14]:
```
movies.shape
```

Out[14]:
```
(559, 6)
```

In [15]:
```
movies.describe()
```

Out[15]:

	CriticRating	AudienceRating	BugetMillions	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 [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    object
 1   Genre           559 non-null    object
 2   CriticRating    559 non-null    int64
 3   AudienceRating  559 non-null    int64
 4   BugetMillions   559 non-null    int64
 5   Year            559 non-null    int64
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

In [17]:
```
movies.Film = movies.Film.astype('category')
```

In [18]:
```
movies.Genre =  movies.Genre.astype('category')
movies.Year =  movies.Year.astype('category')
```

In [19]:
```
movies.info
```

Out[19]:
```
<bound method DataFrame.info of                     Film      Genre  CriticRating  AudienceRating  \
0    (500) Days of Summer      Comedy            87              81
1              10,000 B.C.  Adventure             9              44
2               12 Rounds      Action            30              52
3               127 Hours  Adventure            93              84
4               17 Again      Comedy            55              70
..                     ...         ...           ...             ...
554         Your Highness     Comedy            26              36
555        Youth in Revolt     Comedy            68              52
556                 Zodiac   Thriller            89              73
557           Zombieland      Action            90              87
558             Zookeeper     Comedy            14              42

     BugetMillions  Year
0                8  2009
1              105  2008
2               20  2009
3               18  2010
4               20  2009
..             ...   ...
554             50  2011
555             18  2009
556             65  2007
557             24  2009
558             80  2011

[559 rows x 6 columns]>
```

In [20]:
```
movies.describe()
```

Out[20]:

	CriticRating	AudienceRating	BugetMillions
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 [21]:
```
movies.Film = movies.Film.astype('category')
movies.Genre =  movies.Genre.astype('category')
movies.Year =  movies.Year.astype('category')
```

In [22]:
```
movies.describe()
```

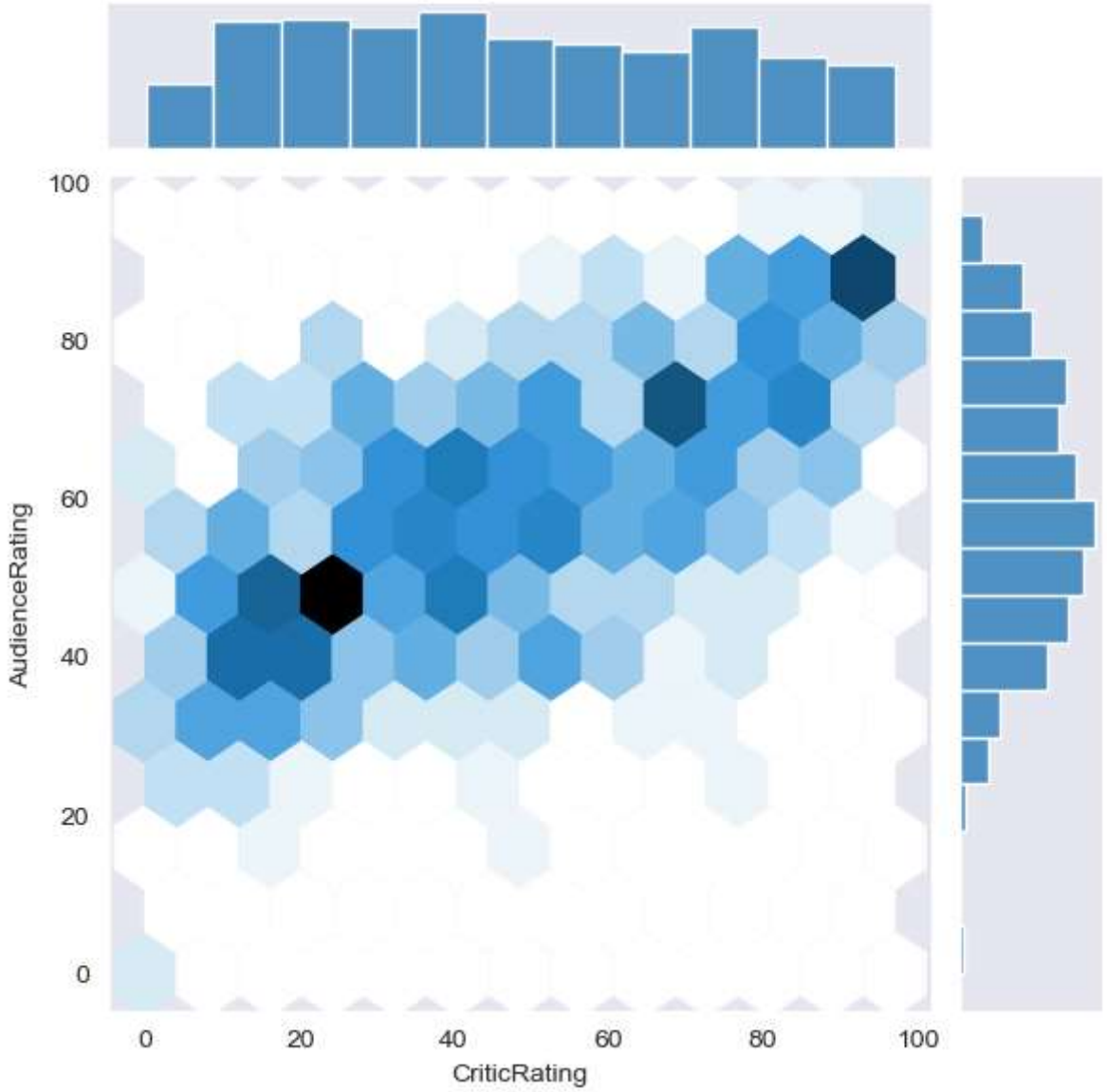
Out[22]:

	CriticRating	AudienceRating	BugetMillions
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

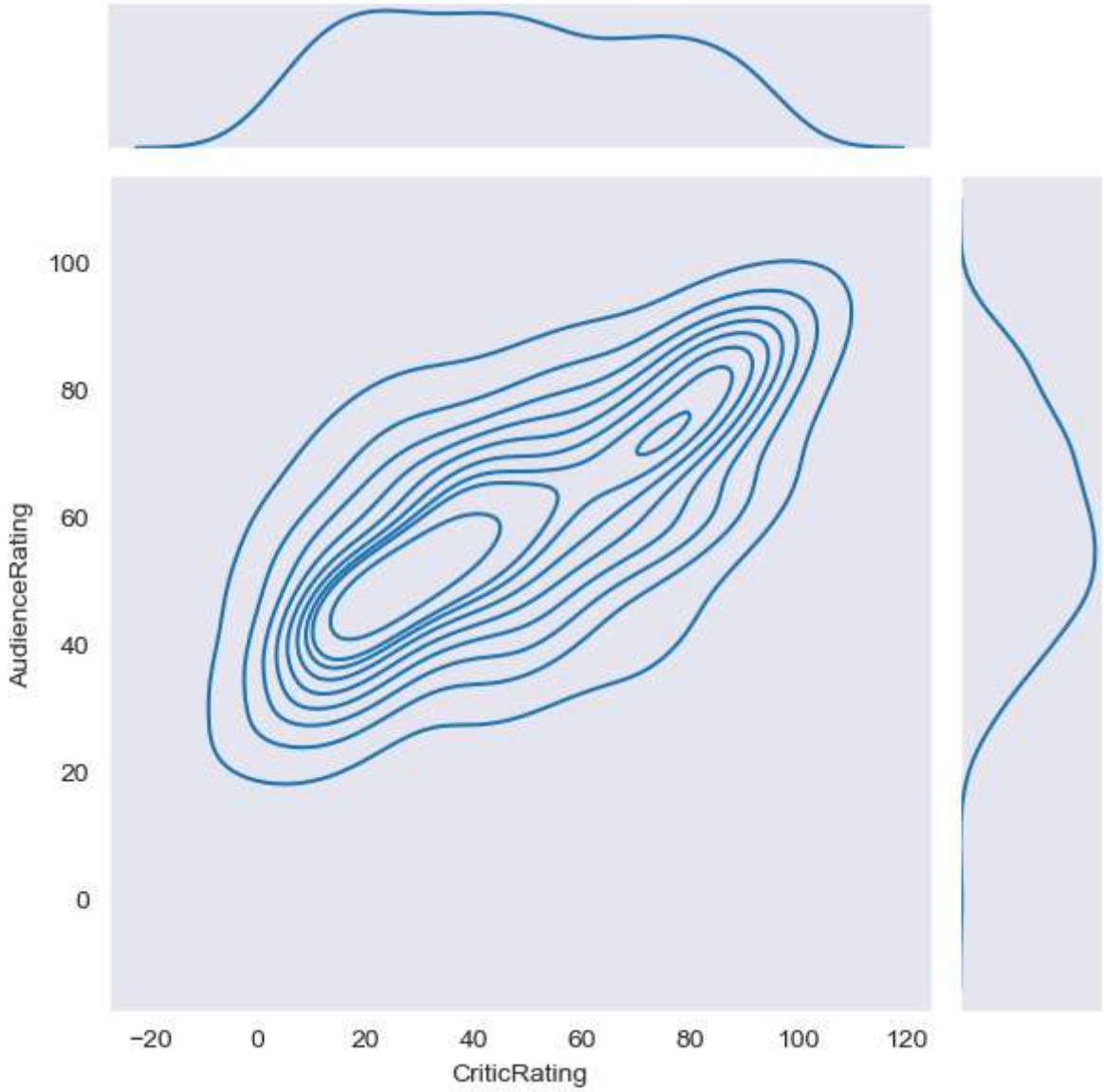
In [ ]:
```
from matplotlib  import pyplot as plt
import seaborn as sns
%matplotlib inline #All the  should inside the line

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

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

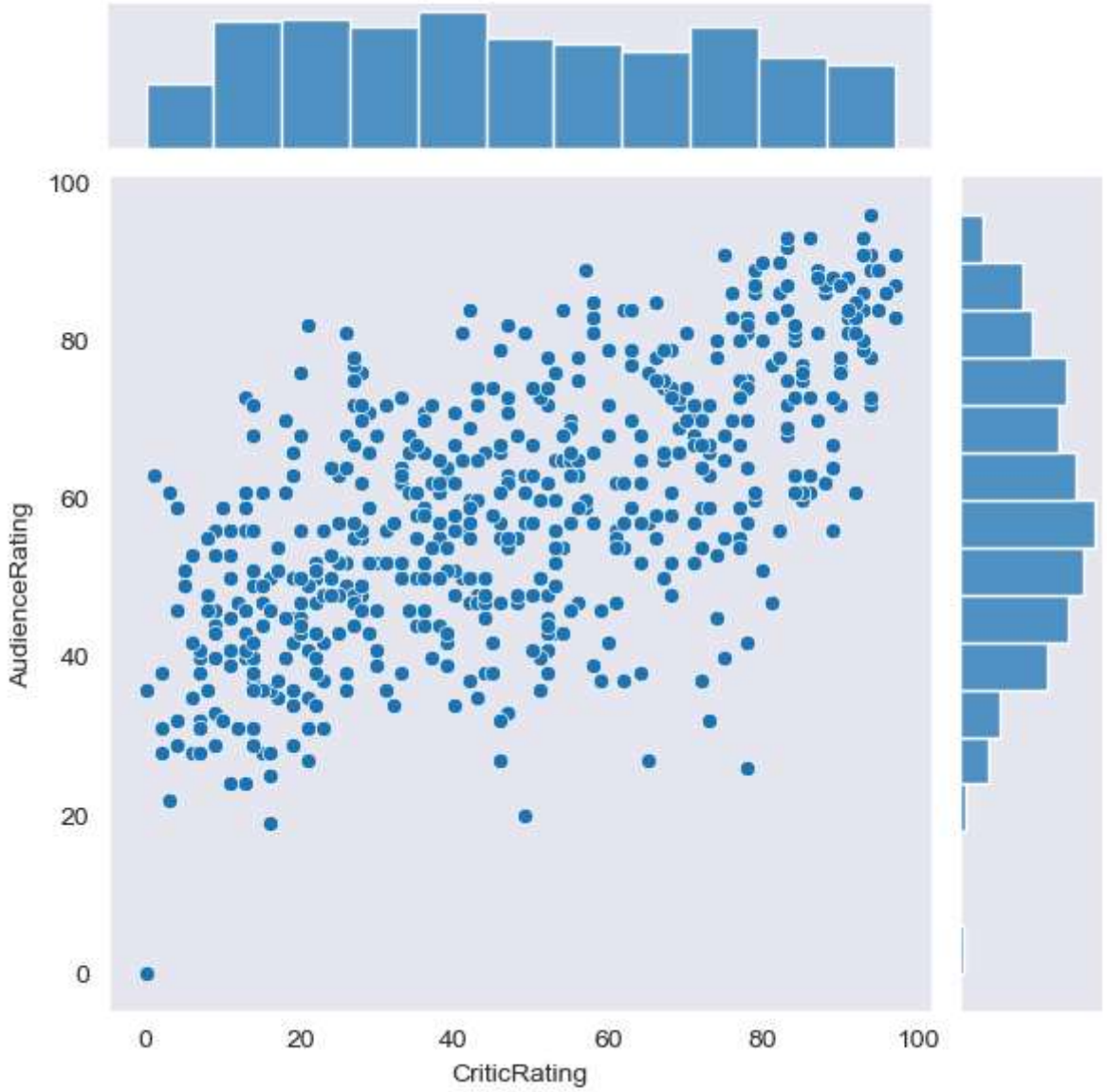
In [54]:
```
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating',kind='hex')
```



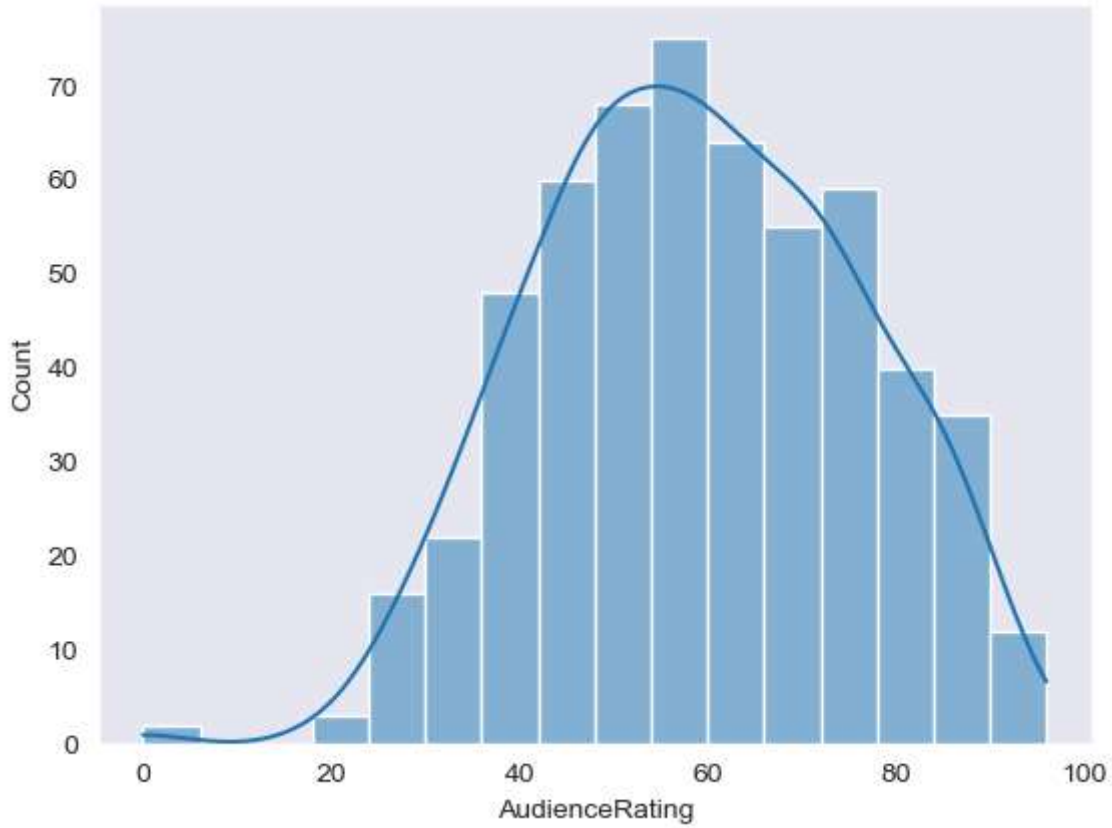
In [55]:
```
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating',kind='kde')
```



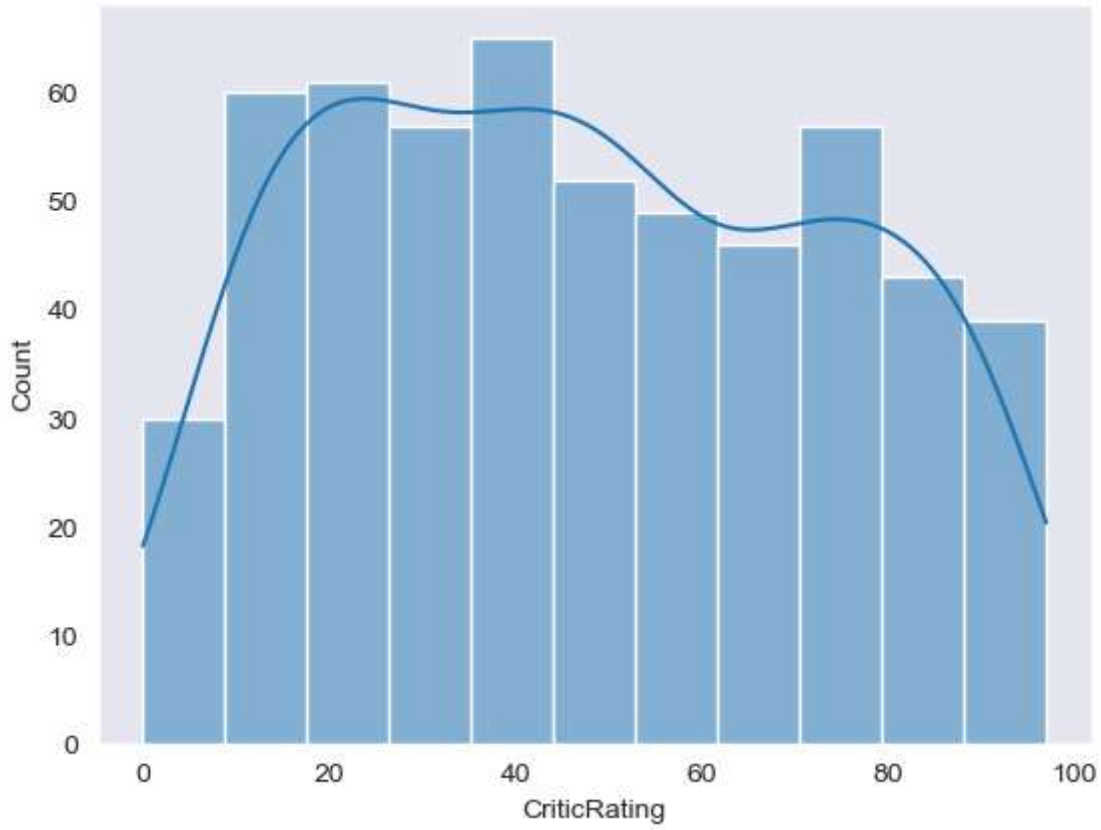
In [56]:
```
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating',kind='scatter')
```



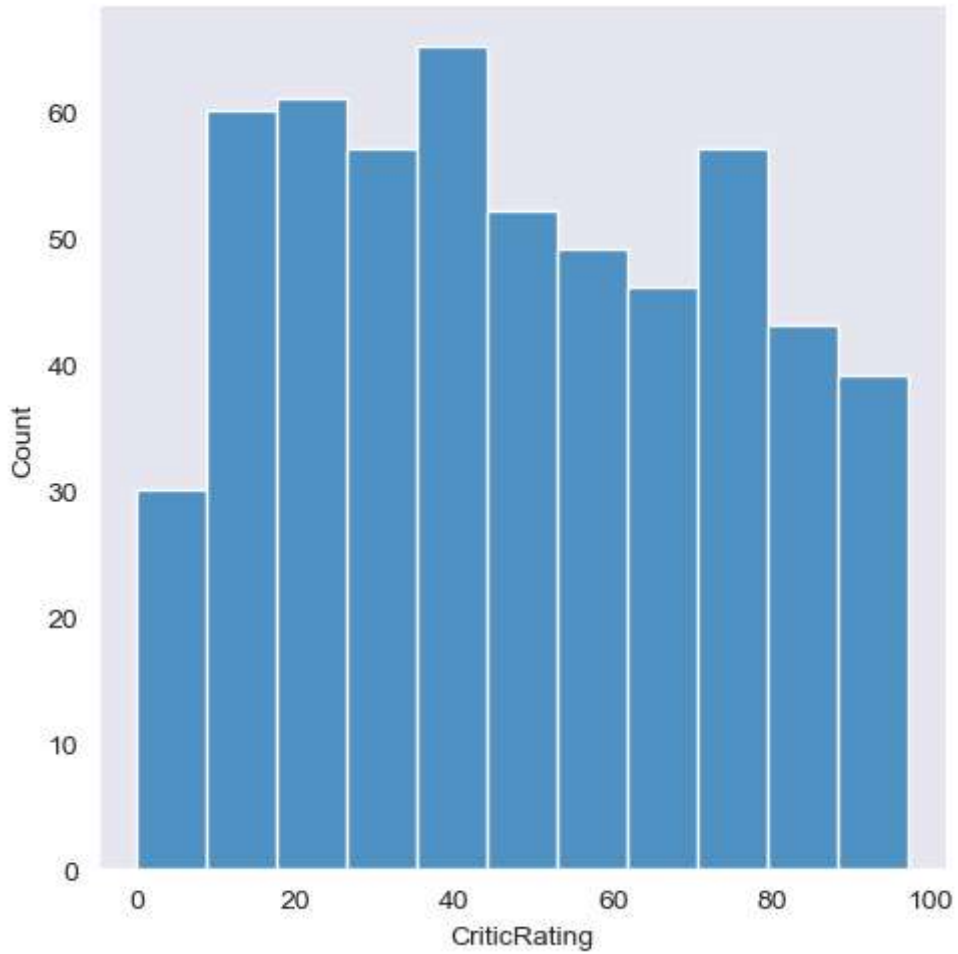
In [57]:
```
m1 = sns.histplot(movies.AudienceRating,kde=True)
```



In [58]:
```
m1 = sns.histplot(movies.CriticRating,kde=True)
```

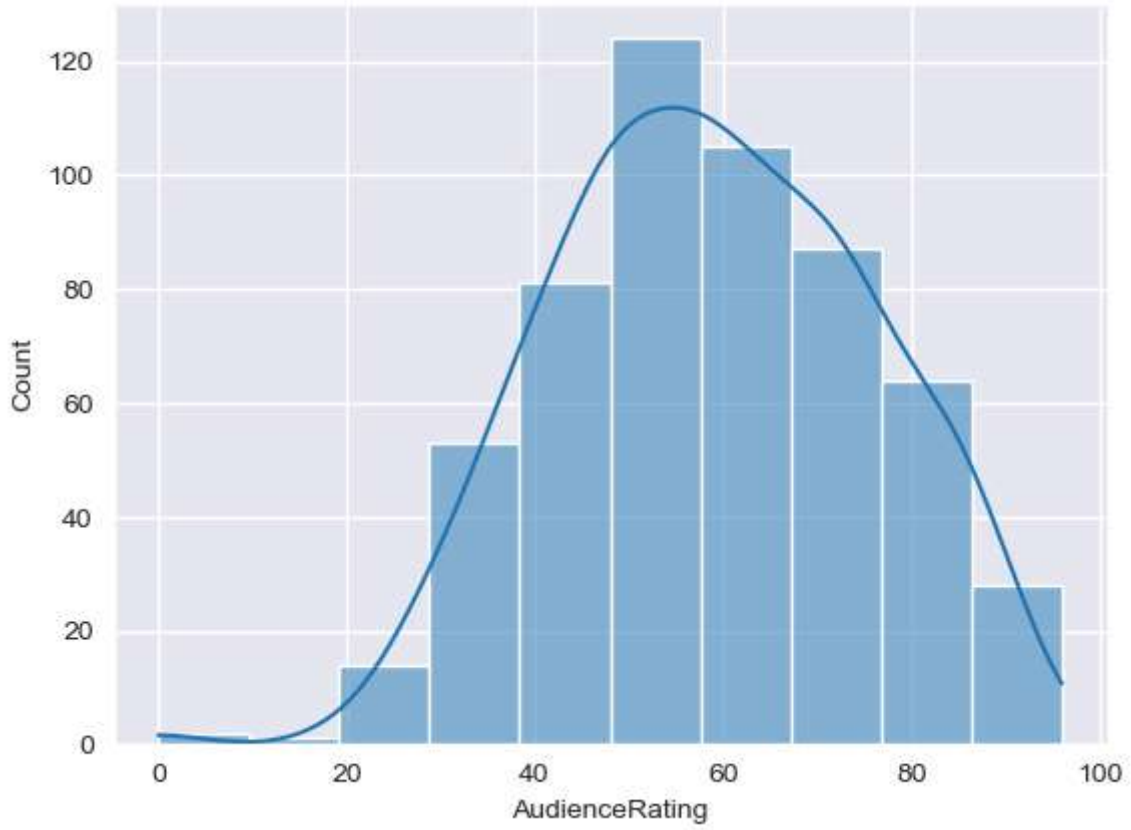


In [59]:
```
m1 = sns.displot(movies.CriticRating )
```



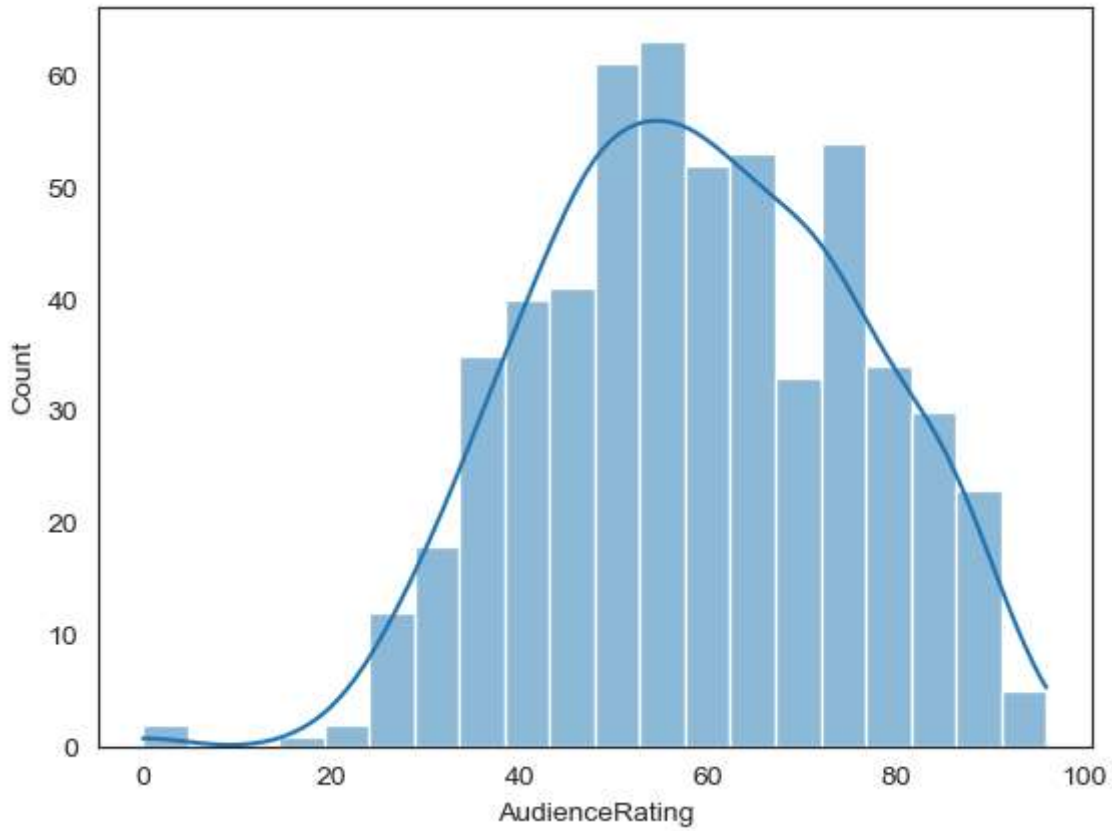
In [60]:
```
sns.set_style('darkgrid')
```

In [61]:
```
m2 = sns.histplot(movies.AudienceRating, bins=10,kde=True)
```

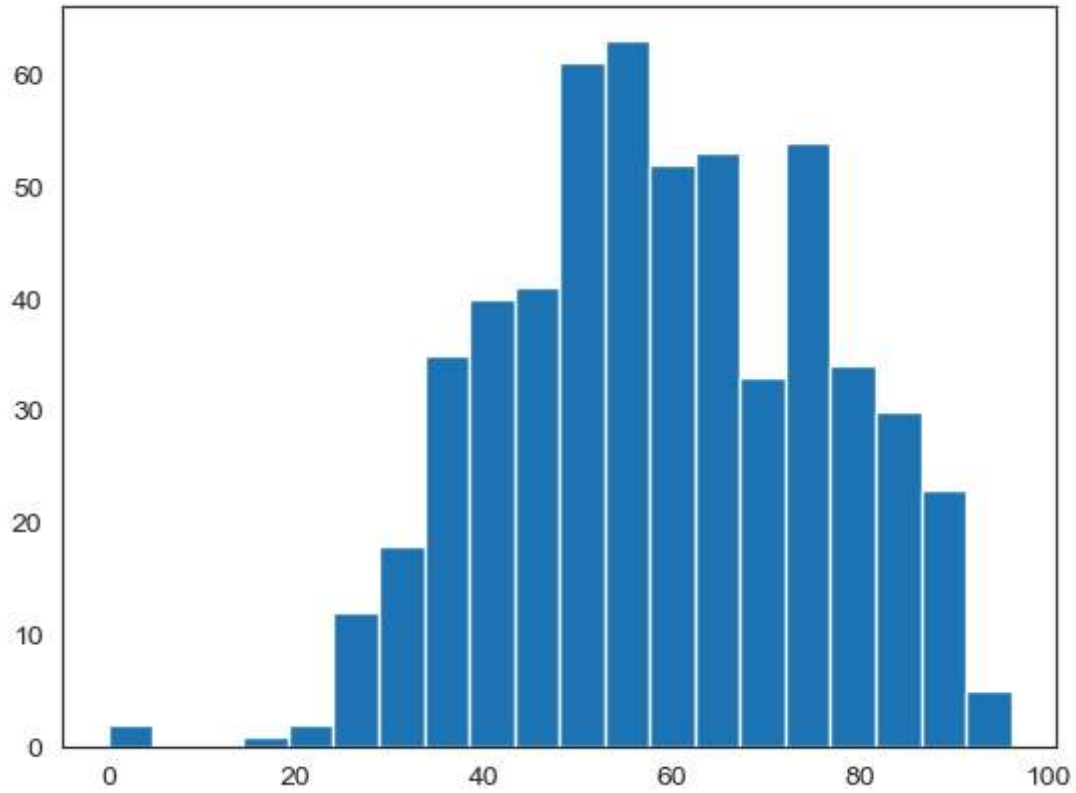


In [62]:
```
sns.set_style('white')
```

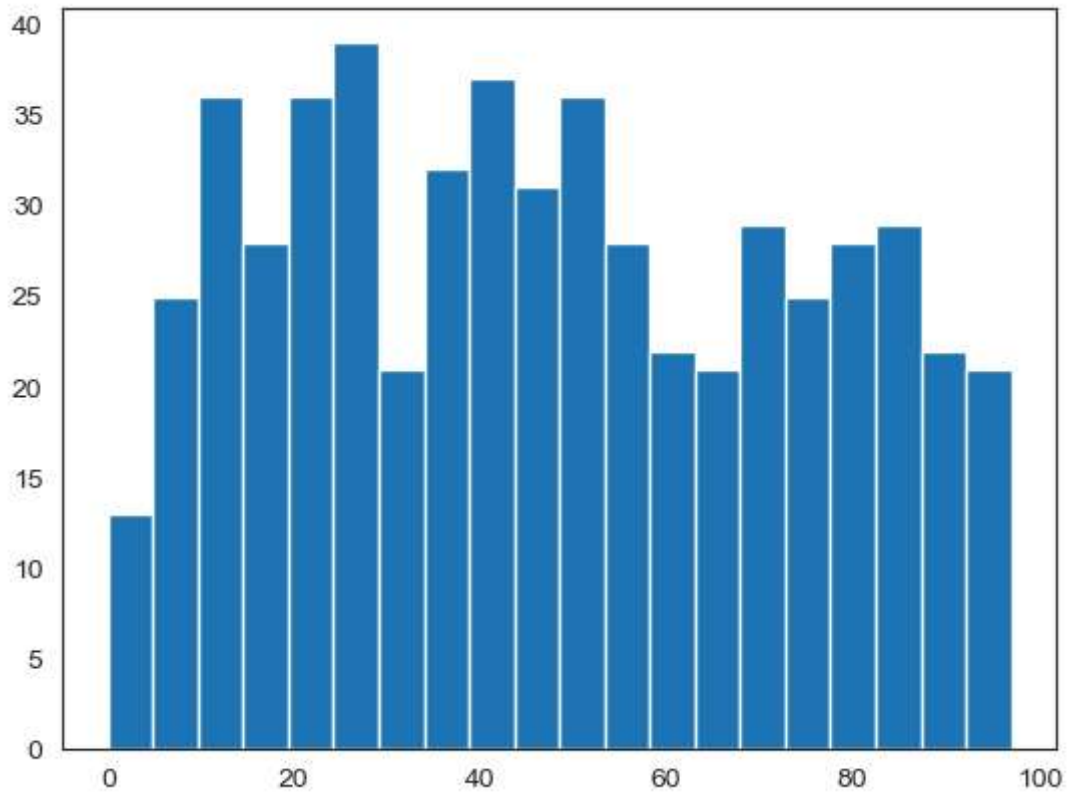
In [63]:
```
m3 = sns.histplot(movies.AudienceRating, bins=20,kde=True)
```



In [64]:
```
n1 = plt.hist(movies.AudienceRating, bins=20)
```



In [65]:
```
n1 = plt.hist(movies.CriticRating, bins=20)
```



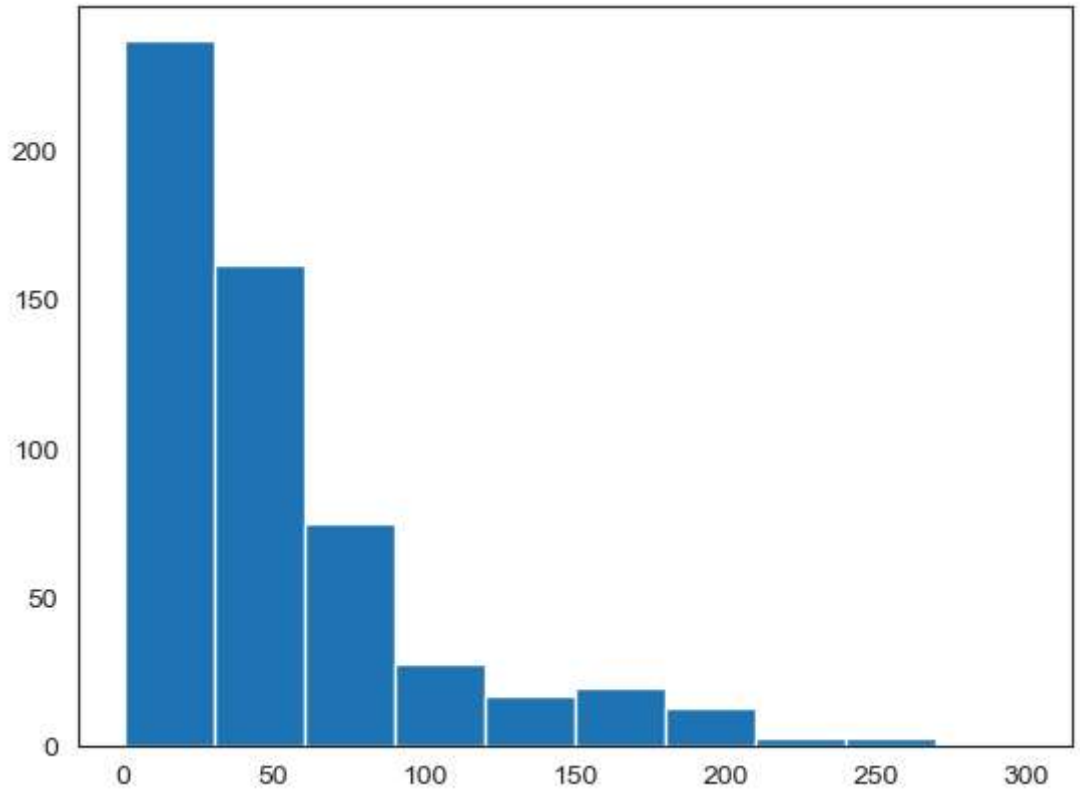
In [66]:
```
# <<< chat -  2

# Creating stacked histograms & this is bit tough to understand
```

In [67]:
```
movies.head()
```

Out[67]:

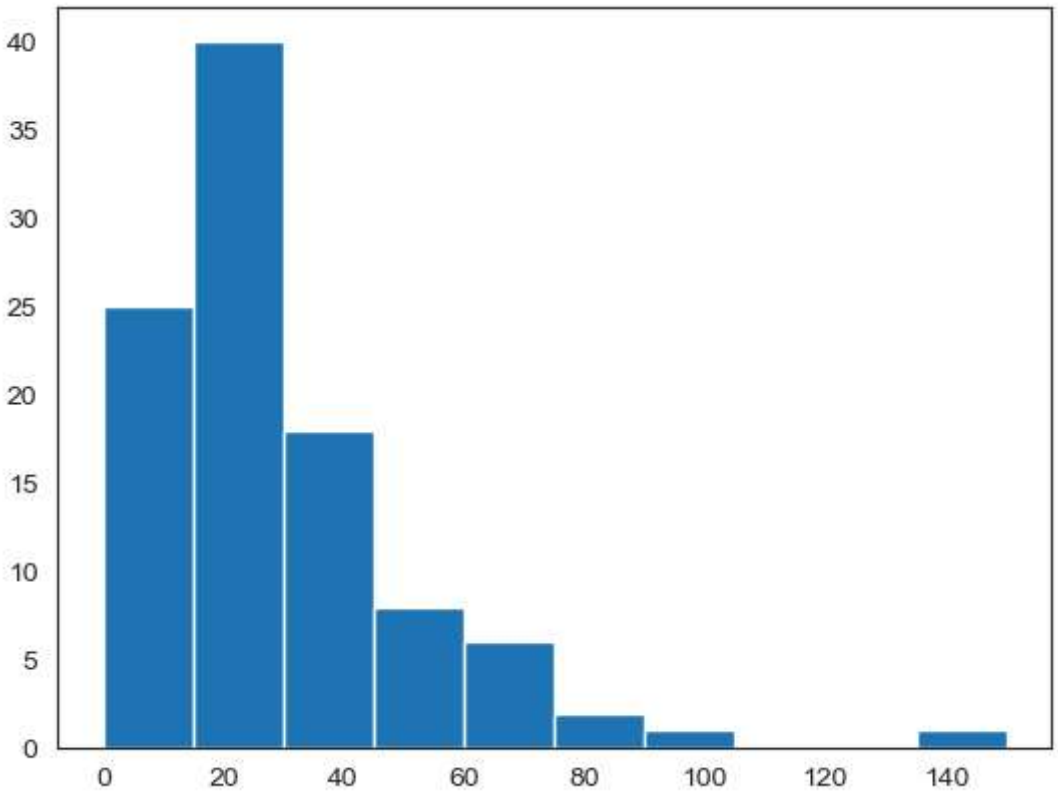
	Film	Genre	CriticRating	AudienceRating	BugetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009

In [68]:
```
plt.hist(data=movies,x='BugetMillions')
plt.show()
```



In [69]:
```
plt.hist(movies[movies.Genre == 'Drama'].       BugetMillions)
```

```
plt.show()
```



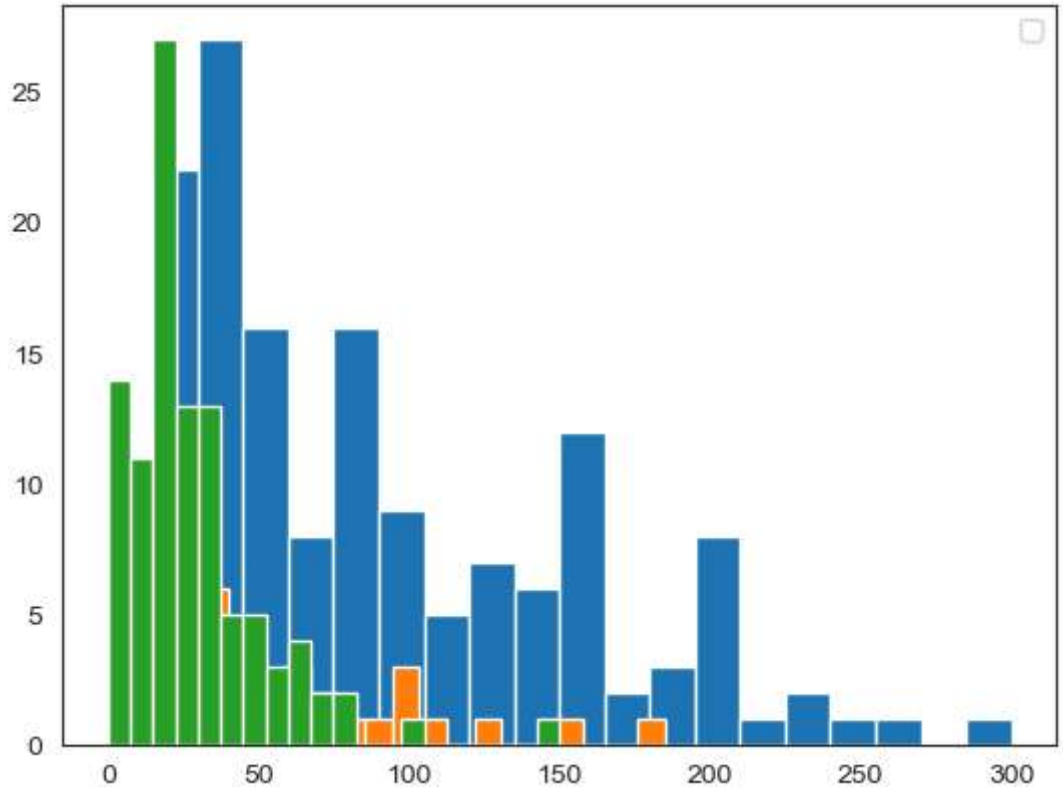
In [70]:
```
movies.head()
```

Out[70]:

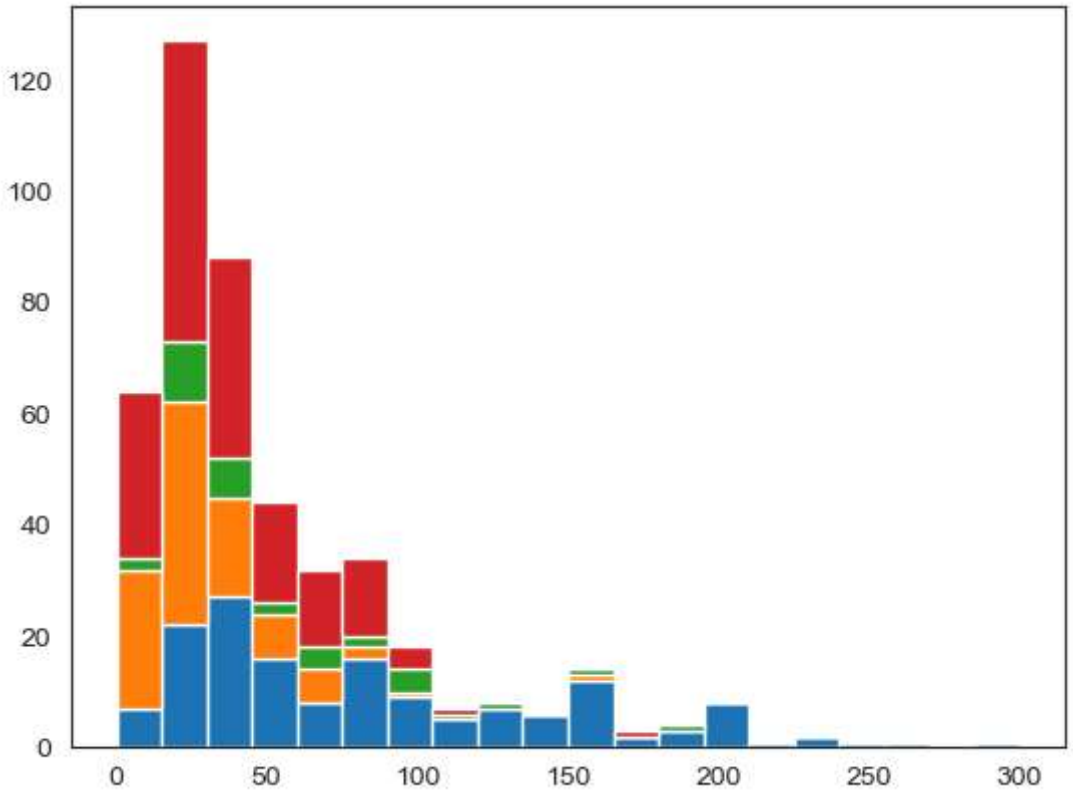
	Film	Genre	CriticRating	AudienceRating	BugetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009

In [71]:
```
# Below plots are stacked histogram becuase overlaped

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

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



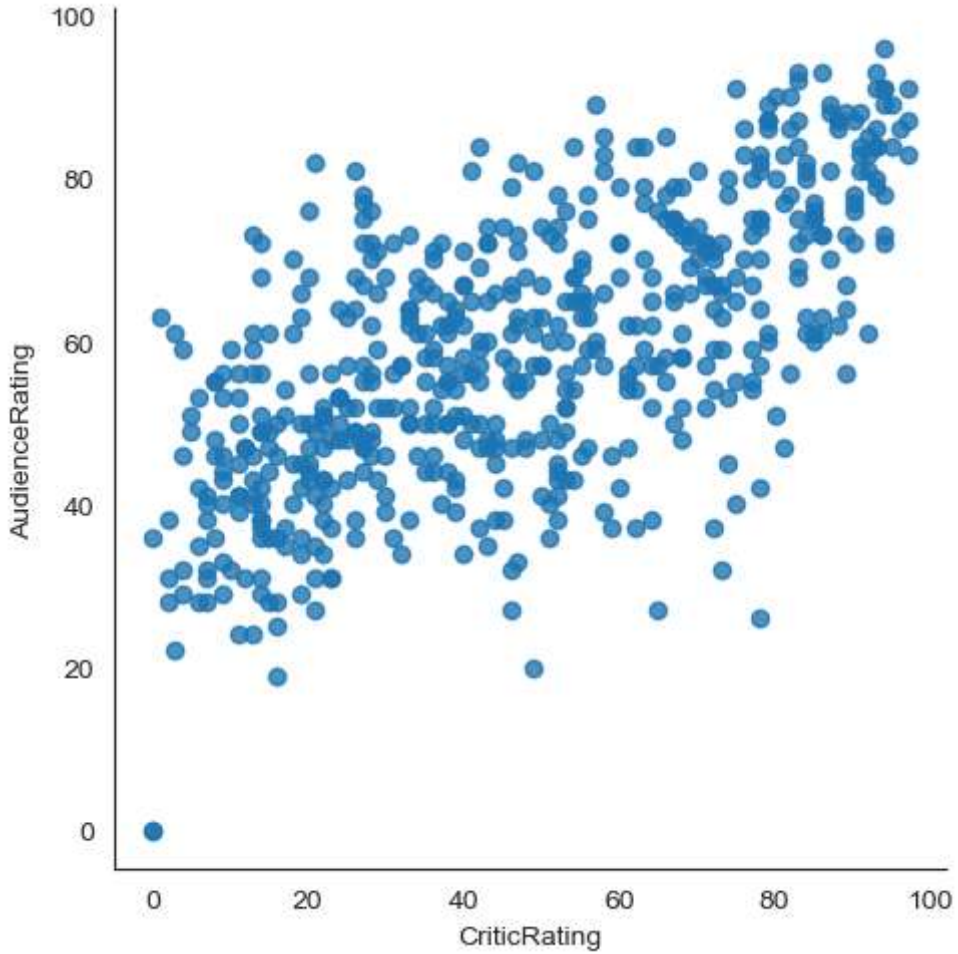
In [72]:
```
plt.hist([movies[movies.Genre == 'Action'].     BugetMillions,\
          movies[movies.Genre == 'Drama'].      BugetMillions, \
          movies[movies.Genre == 'Thriller'].   BugetMillions, \
          movies[movies.Genre == 'Comedy'].     BugetMillions],
         bins = 20, stacked = True)
plt.show()
```



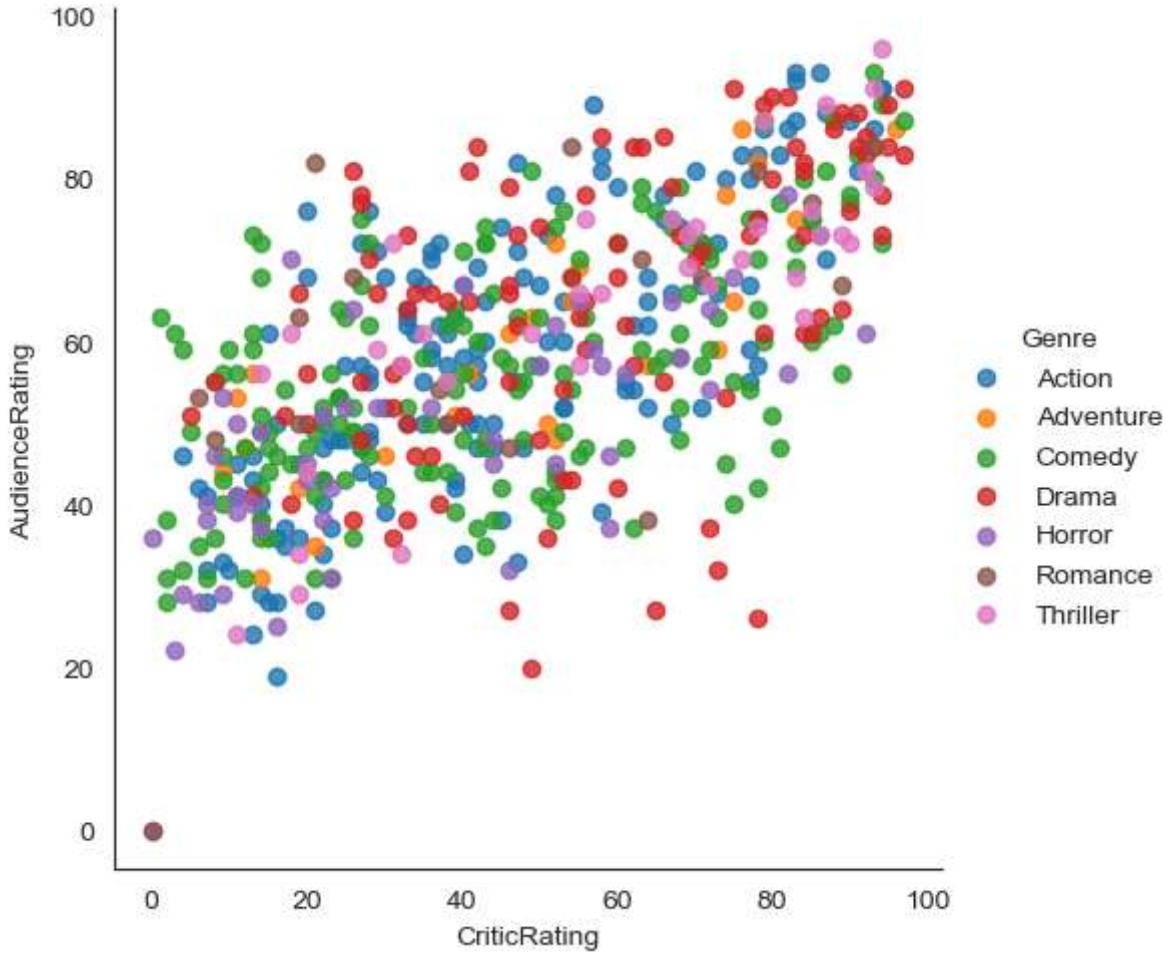
In [73]:
```
# if you have 100 categories you cannot copy & paste all the things

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

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

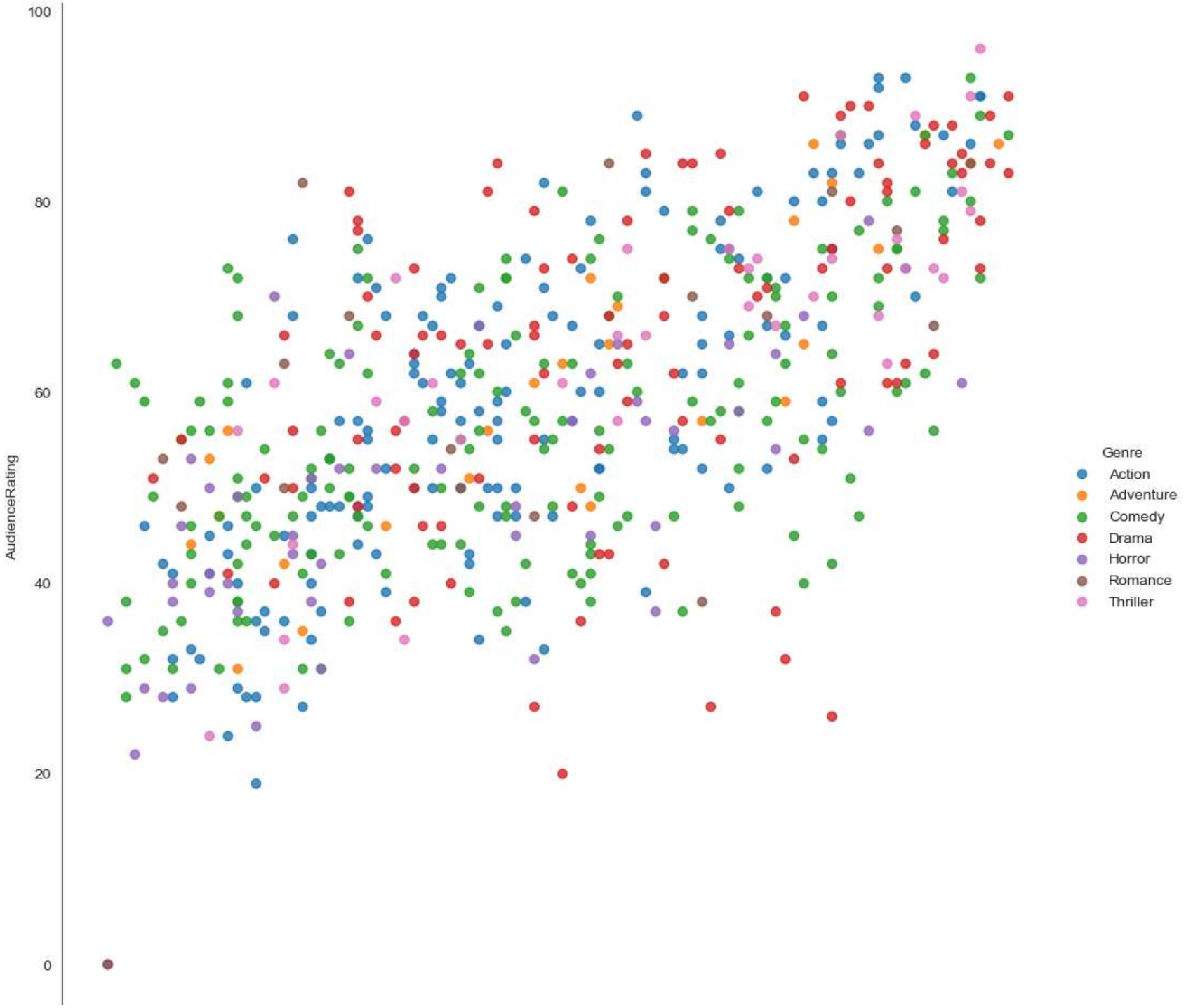
In [74]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',\
                  fit_reg=False)
```



In [75]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',\
                  fit_reg=False, hue = 'Genre')
```

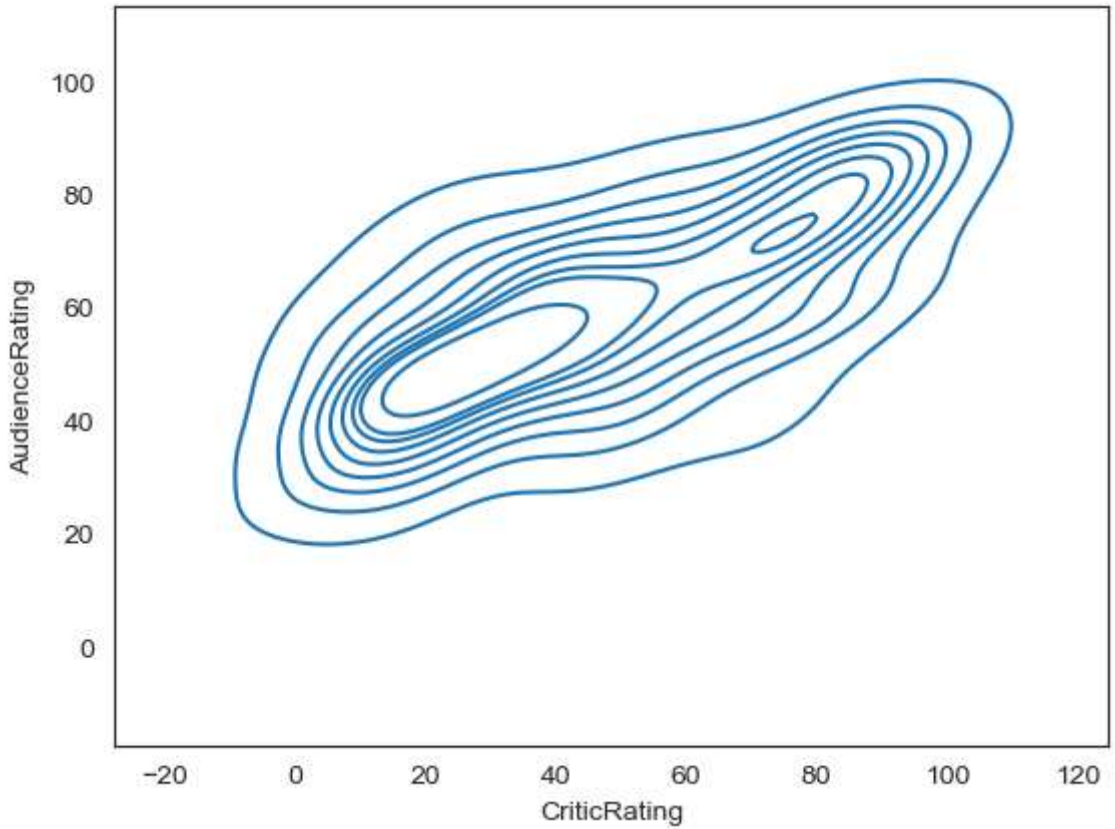


In [76]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating',\
                  fit_reg=False, hue = 'Genre', height = 10,aspect=1)
```


100
80
60
40
20
0
AudienceRating
Genre
Action
Adventure
Comedy
Drama
Horror
Romance
Thriller

In [77]:
```
# Kernal Density Estimate plot ( KDE PLOT)
# how can i visulize audience rating & critics rating . using scatterplot
```

In [78]:
```
k1 = sns.kdeplot(data = movies, x = 'CriticRating', y = 'AudienceRating')

# where do u find more density and how density is distibuted across from the the chat
# center point is kernal this is calld KDE & insteade of dots it visualize like this
# we can able to clearly see the spread at the audience ratings
```



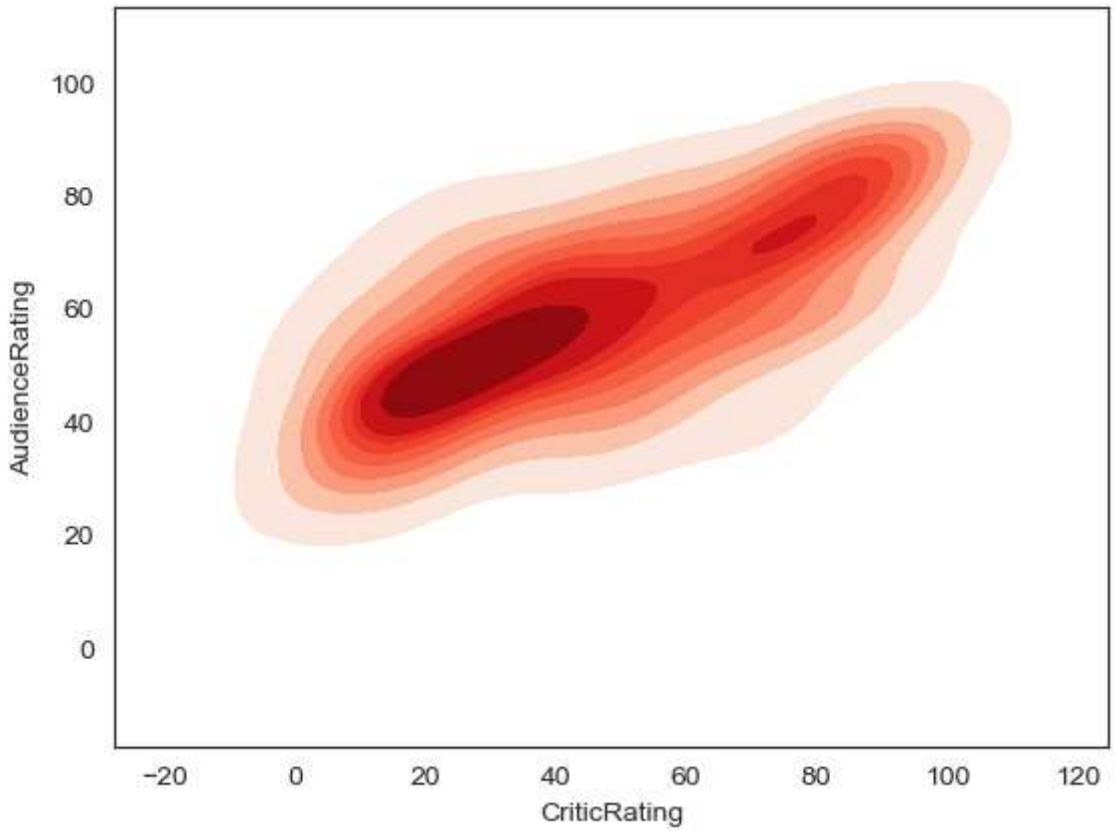
In [79]:
```
movies
```

Out[79]:

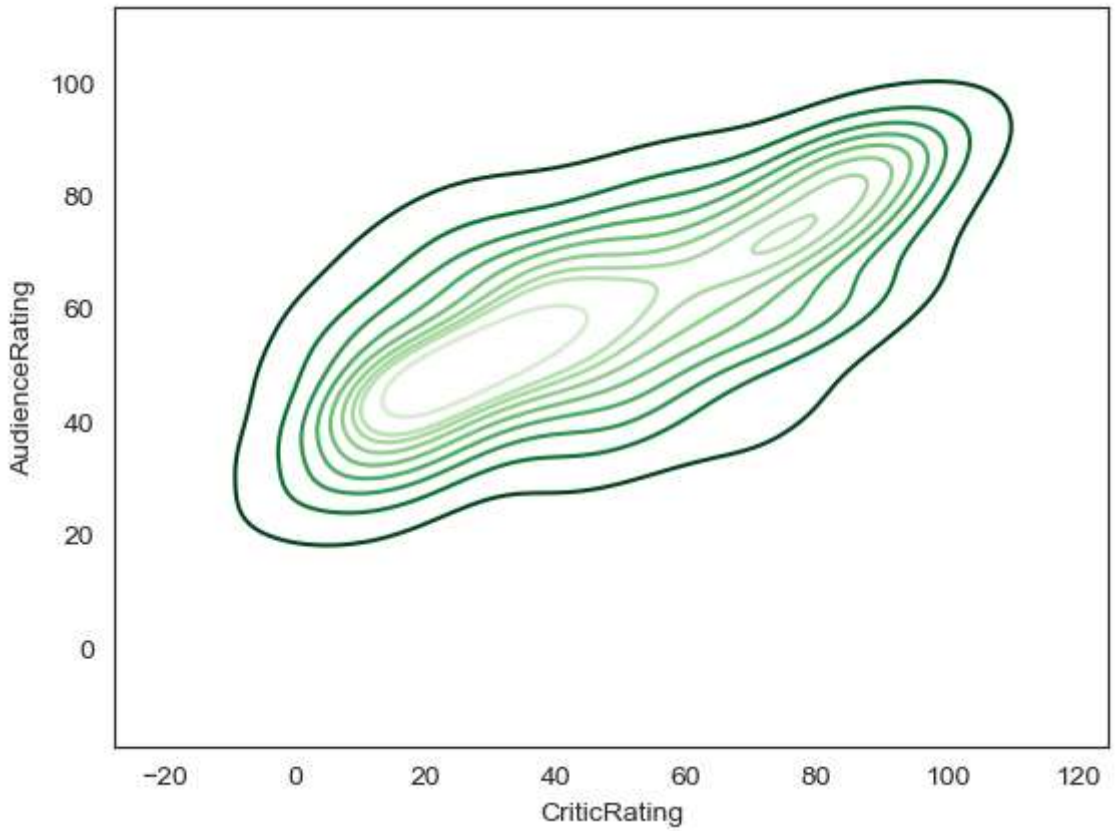
	Film	Genre	CriticRating	AudienceRating	BugetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009
...	...	...	...	...	...	...
554	Your Highness	Comedy	26	36	50	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

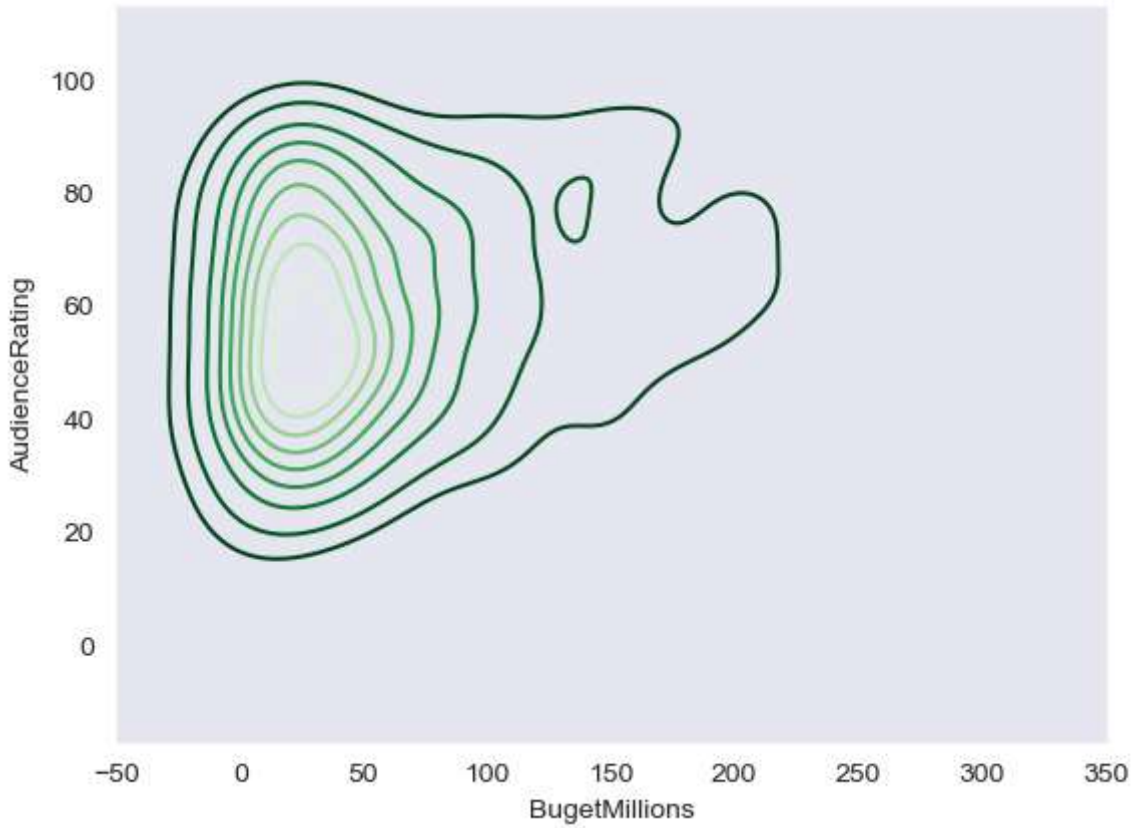
In [80]:
```
import warnings
warnings.filterwarnings('ignore')
k1 = sns.kdeplot(data = movies, x = 'CriticRating', y = 'AudienceRating',shade=True,shade_lowest=False,cmap='Reds')
```



In [81]:
```
k1 = sns.kdeplot(data = movies, x = 'CriticRating', y = 'AudienceRating',shade_lowest=False,cmap='Greens_r')
```



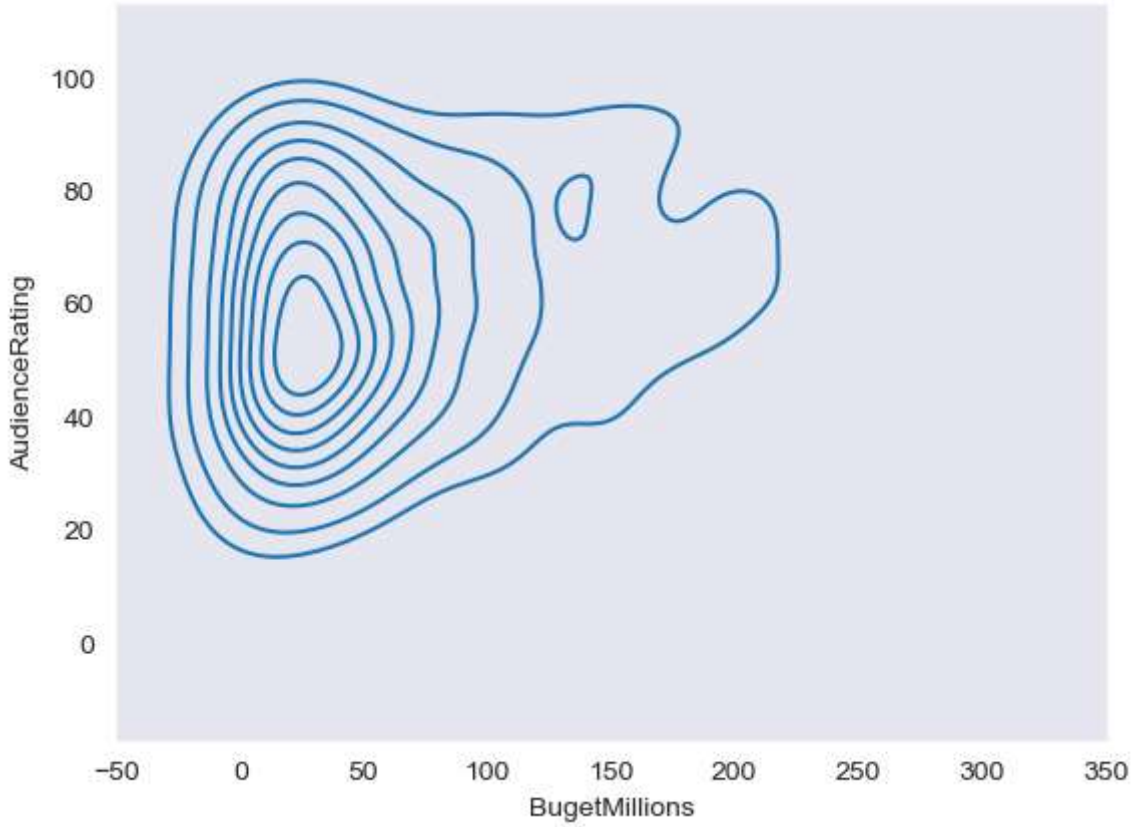
In [82]:
```
sns.set_style('dark')
k1 = sns.kdeplot(data = movies, x = 'BugetMillions', y = 'AudienceRating',shade_lowest=False,cmap='Greens_r')
```



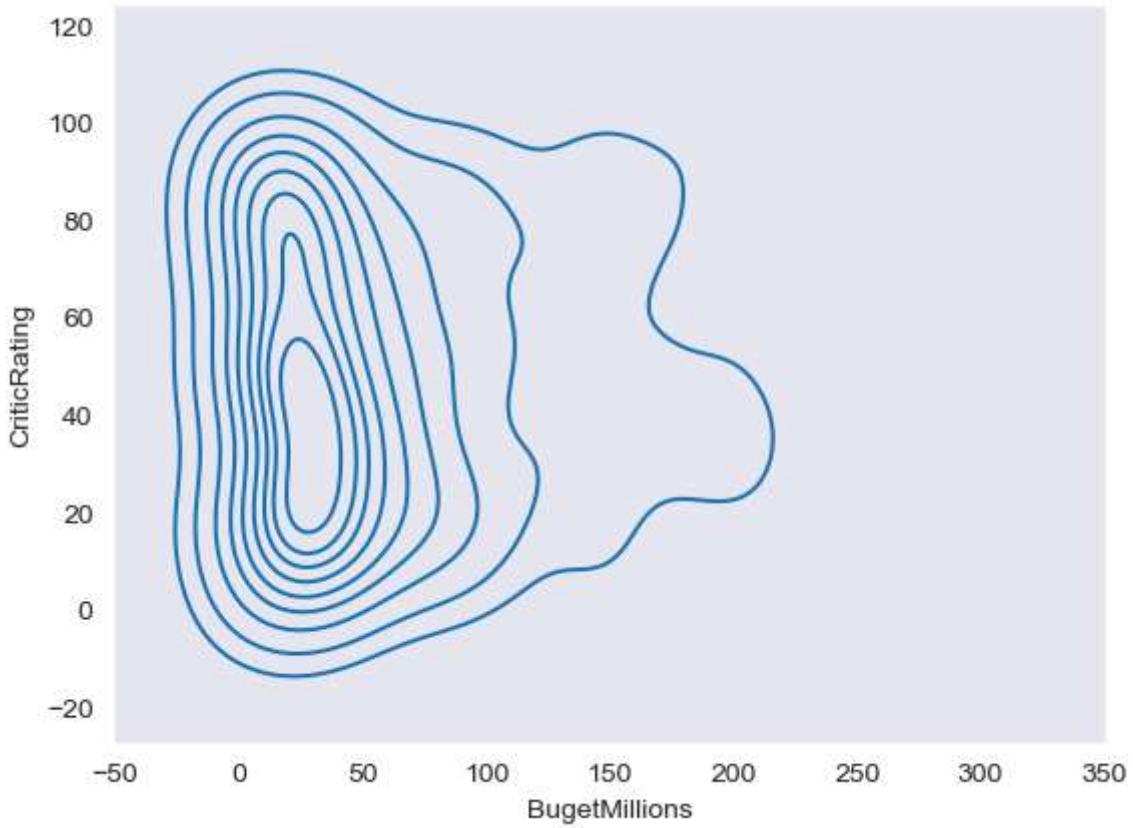
In [83]:
```
movies['BugetMillions']
```

Out[83]:
```
0        8
1      105
2       20
3       18
4       20
      ...
554     50
555     18
556     65
557     24
558     80
Name: BugetMillions, Length: 559, dtype: int64
```

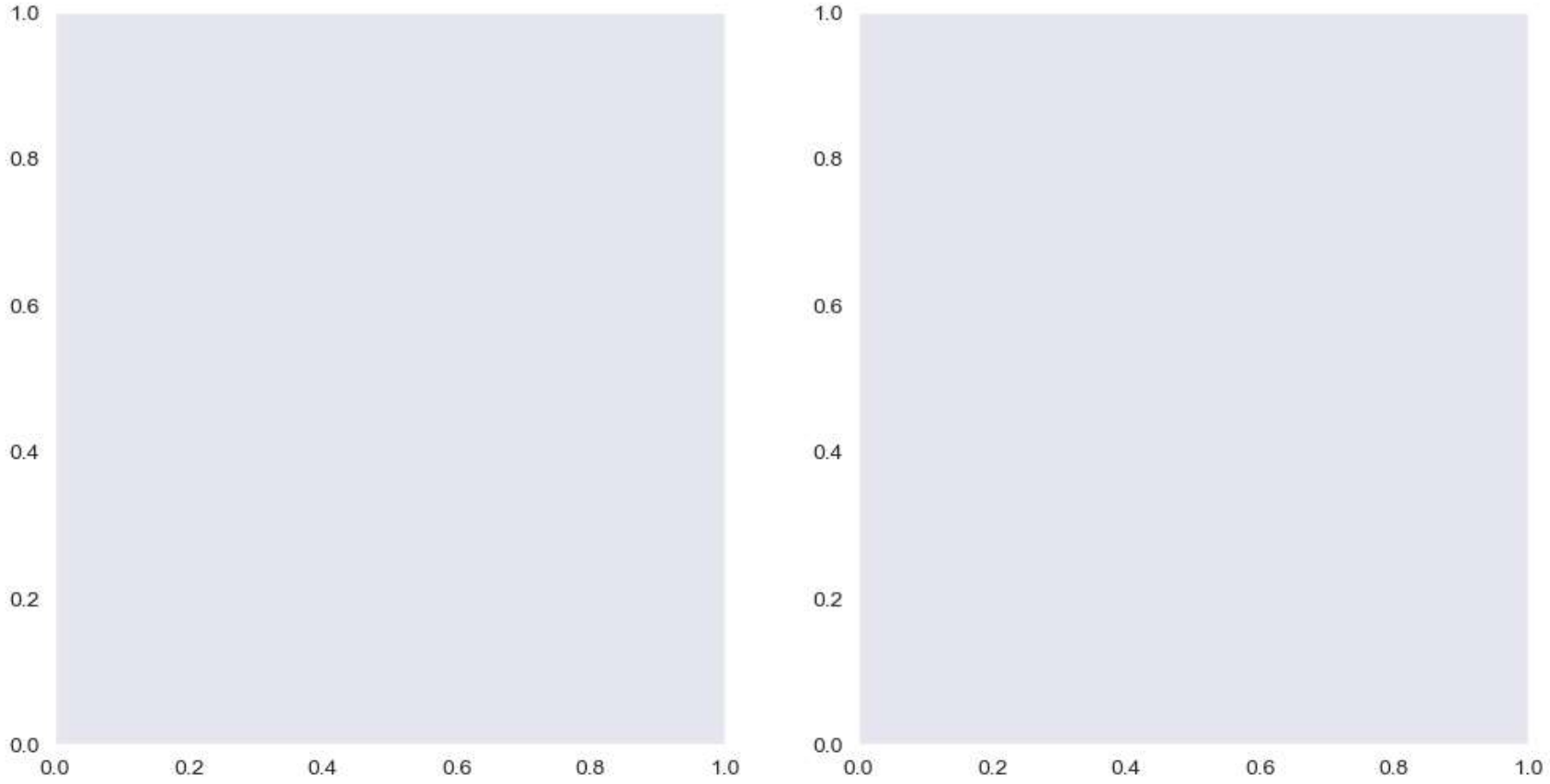
In [84]:
```
sns.set_style('dark')
k1 = sns.kdeplot(data = movies, x = 'BugetMillions', y = 'AudienceRating')
```



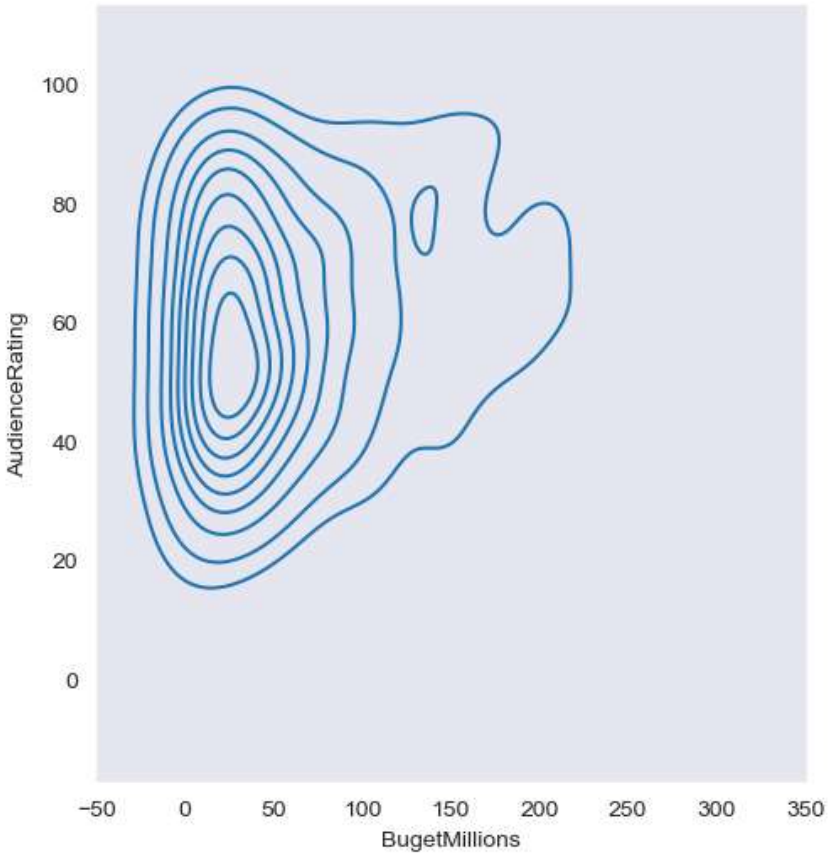
In [85]:
```
sns.set_style('dark')
k1 = sns.kdeplot(data = movies, x = 'BugetMillions', y = 'CriticRating')
```

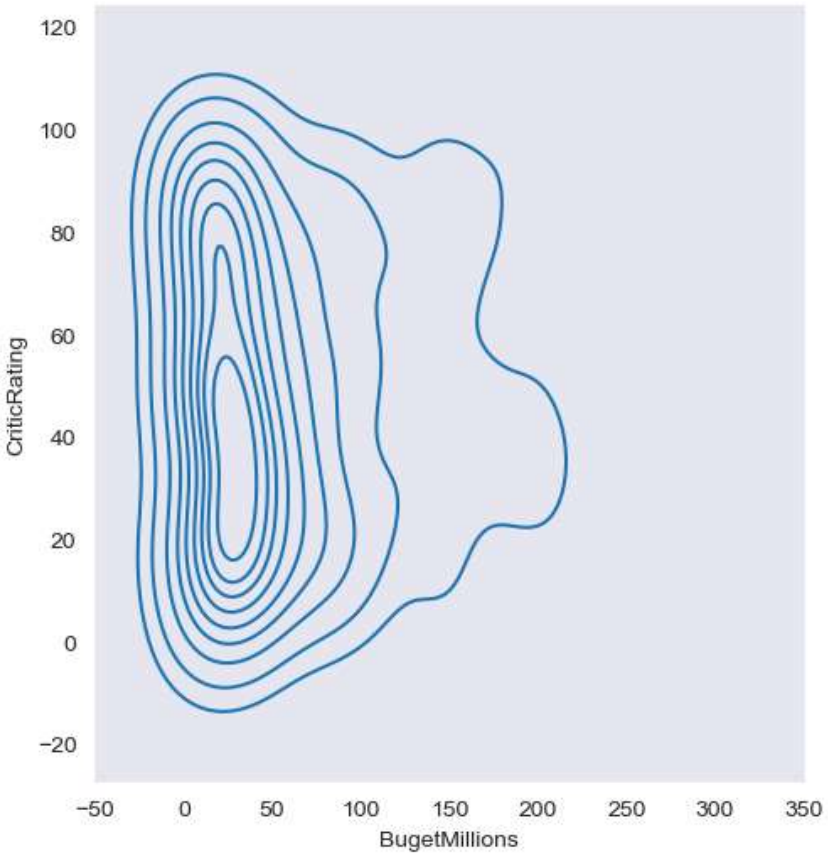


In [86]:
```
#subplots

f, ax = plt.subplots(1,2, figsize =(12,6))
#f, ax = plt.subplots(3,3, figsize =(12,6))
```



In [87]:
```
f, axes = plt.subplots(1,2, figsize =(12,6))

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

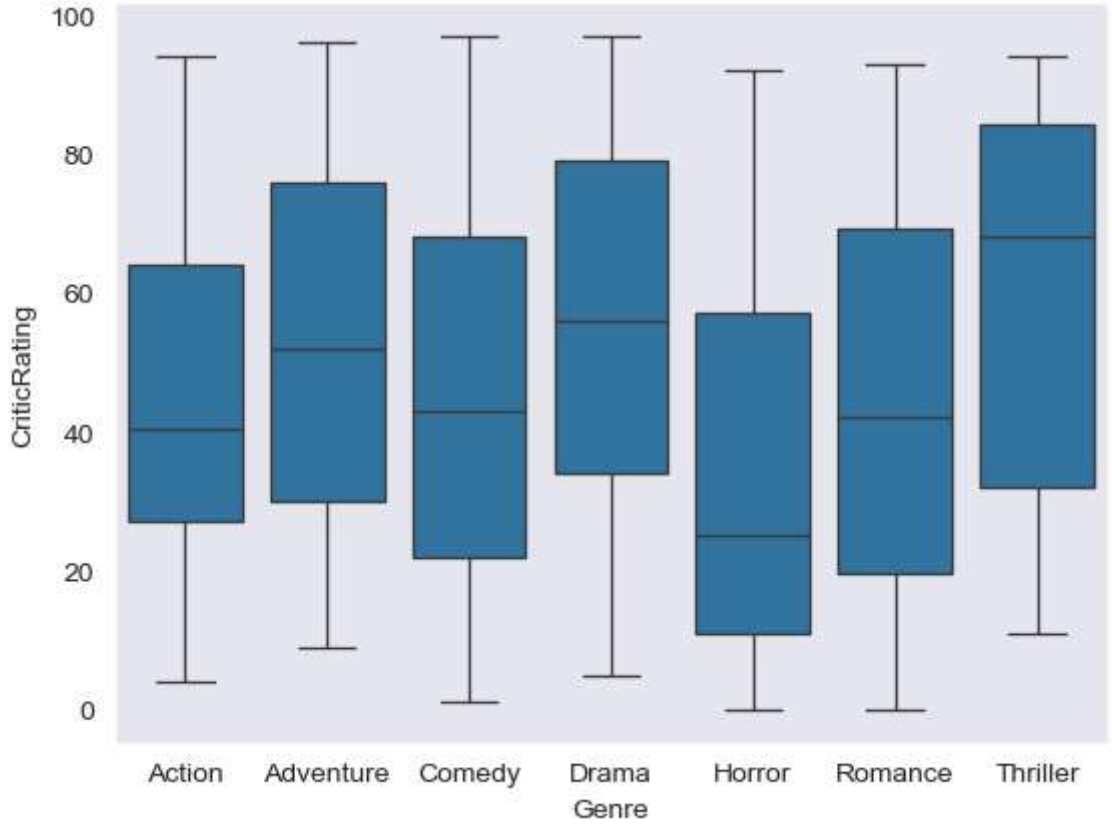




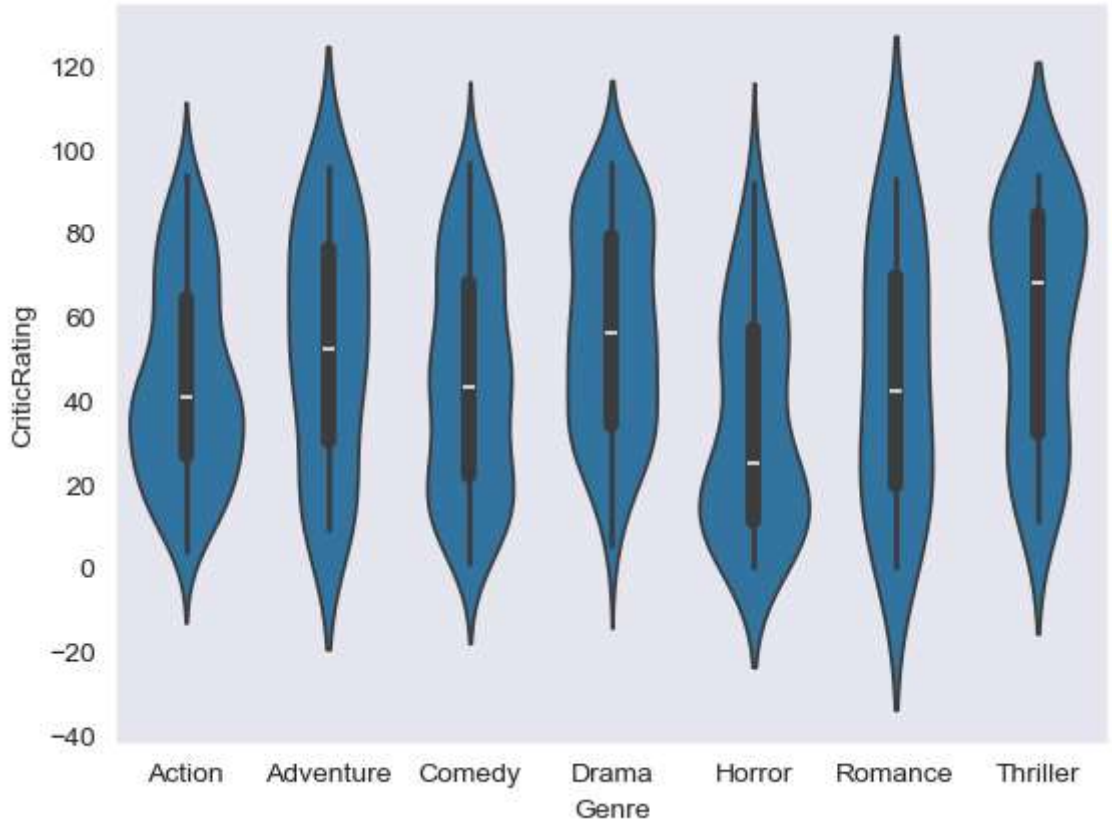
In [88]:
```
axes
```

Out[88]:
```
array([<Axes: xlabel='BugetMillions', ylabel='AudienceRating'>,
       <Axes: xlabel='BugetMillions', ylabel='CriticRating'>],
      dtype=object)
```

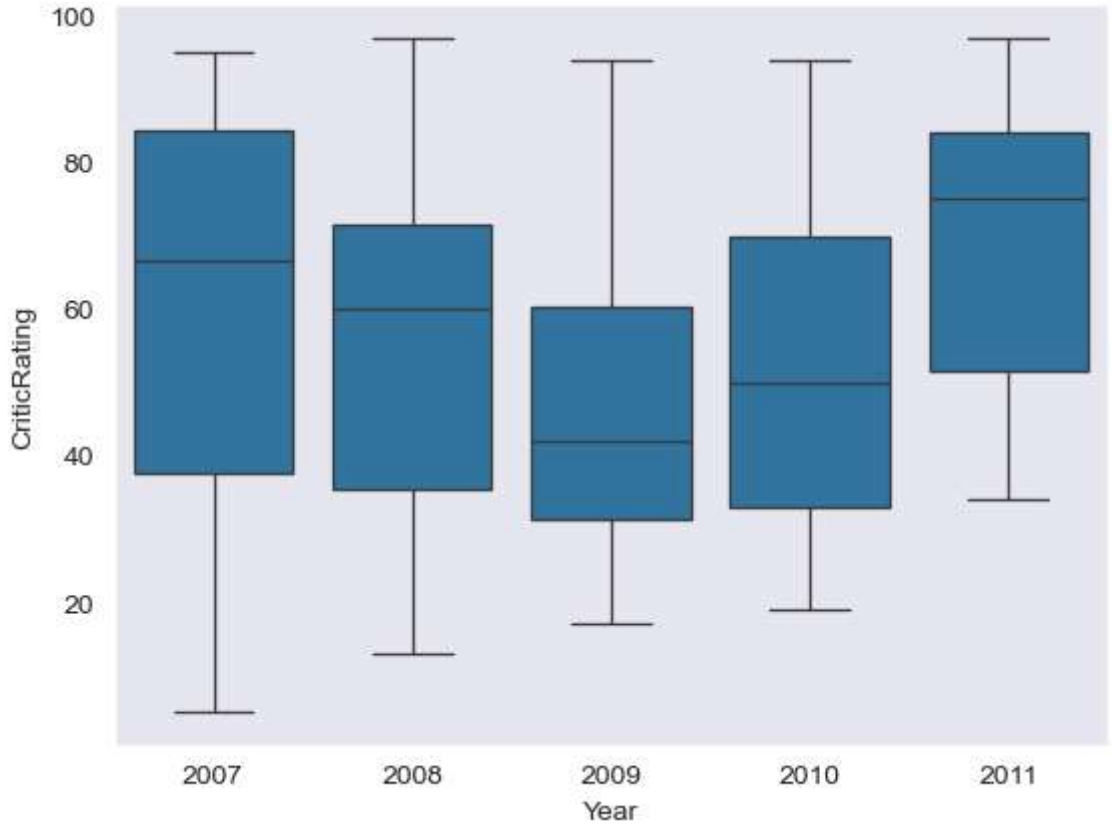
In [89]:
```
#Box plots -

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



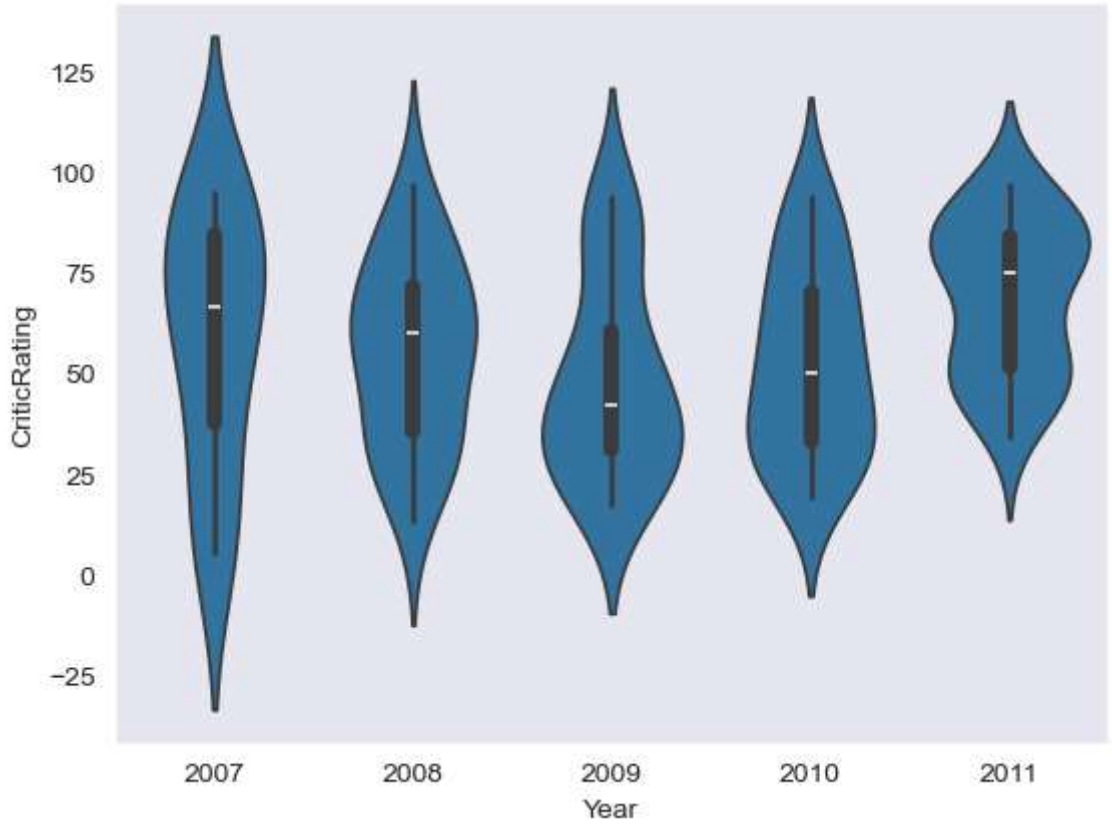
In [90]:
```
#violin plot

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



In [91]:
```
w1 = sns.boxplot(data=movies[movies.Genre == 'Drama'], x='Year', y = 'CriticRating')
```

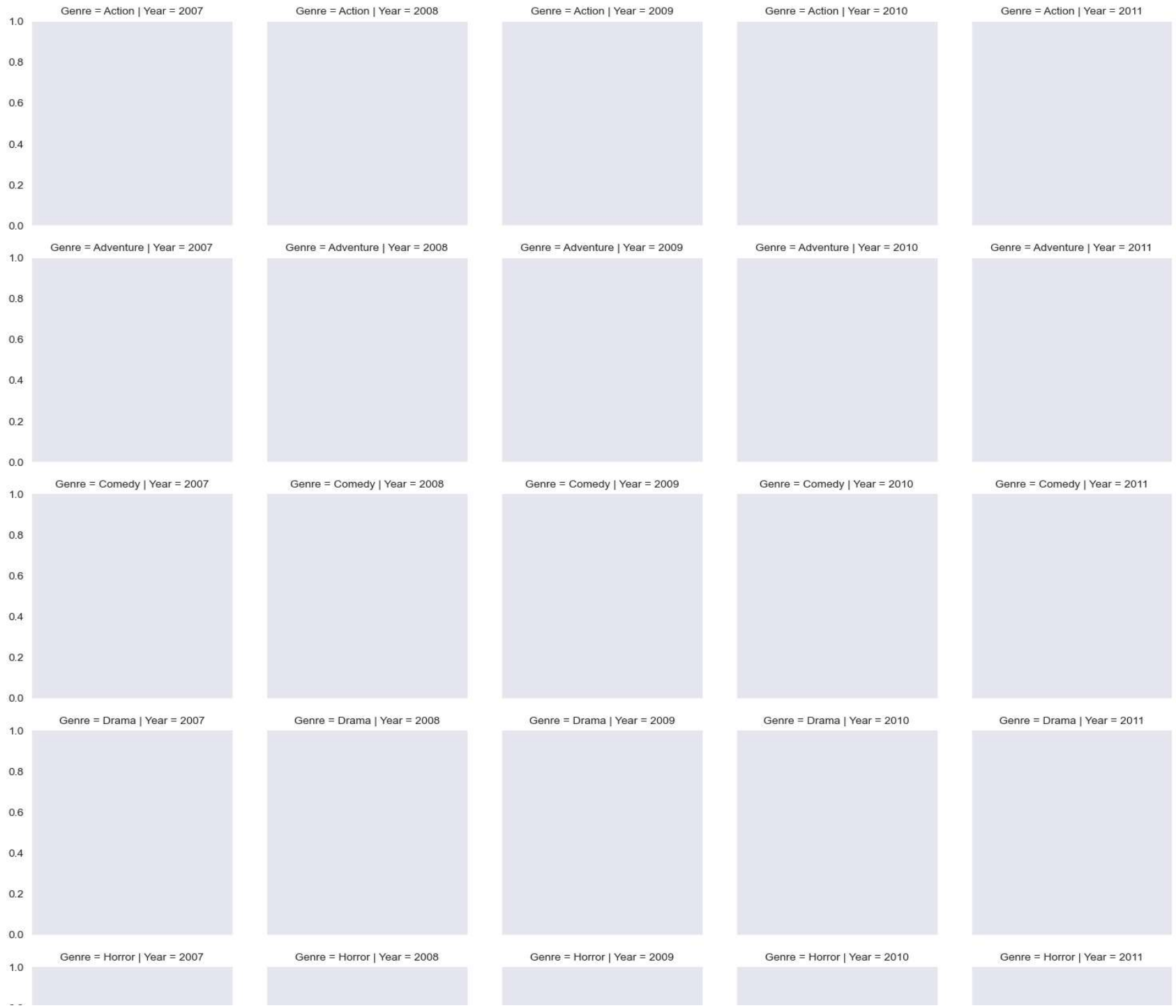


In [92]:
```
z = sns.violinplot(data=movies[movies.Genre == 'Drama'], x='Year', y = 'CriticRating')
```



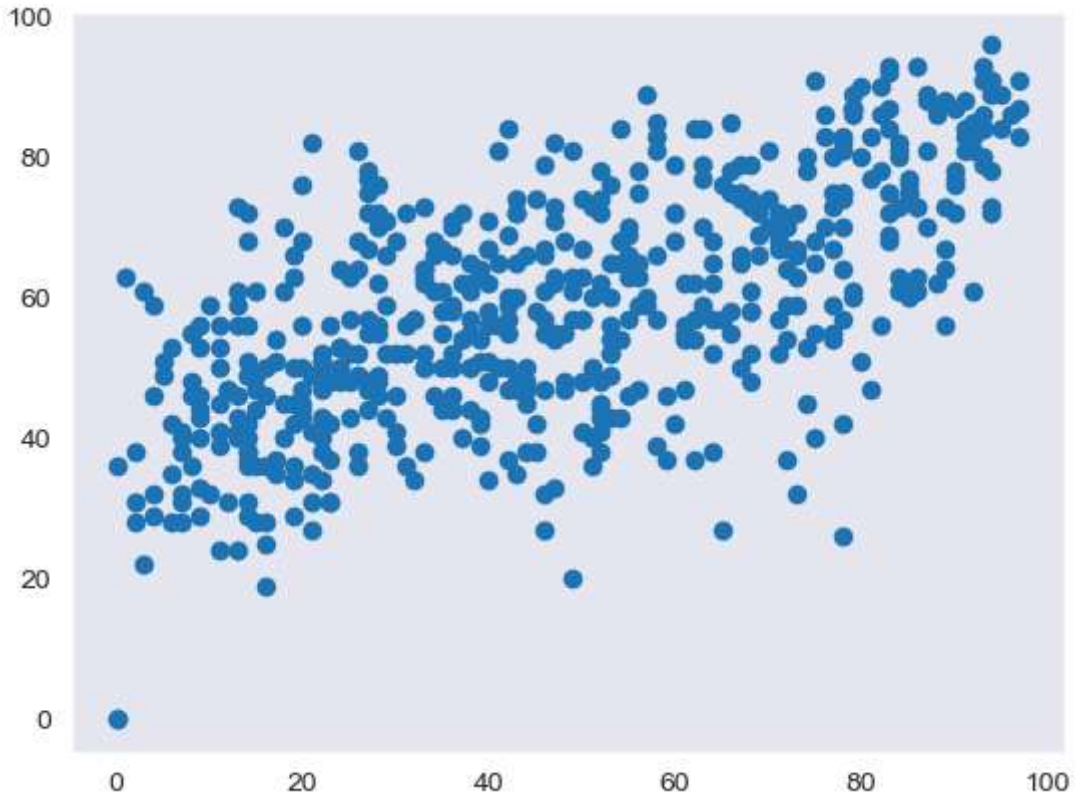
In [93]:
```
# Createing a Facet grid
```

In [94]:
```
g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre') #kind of subplots
```

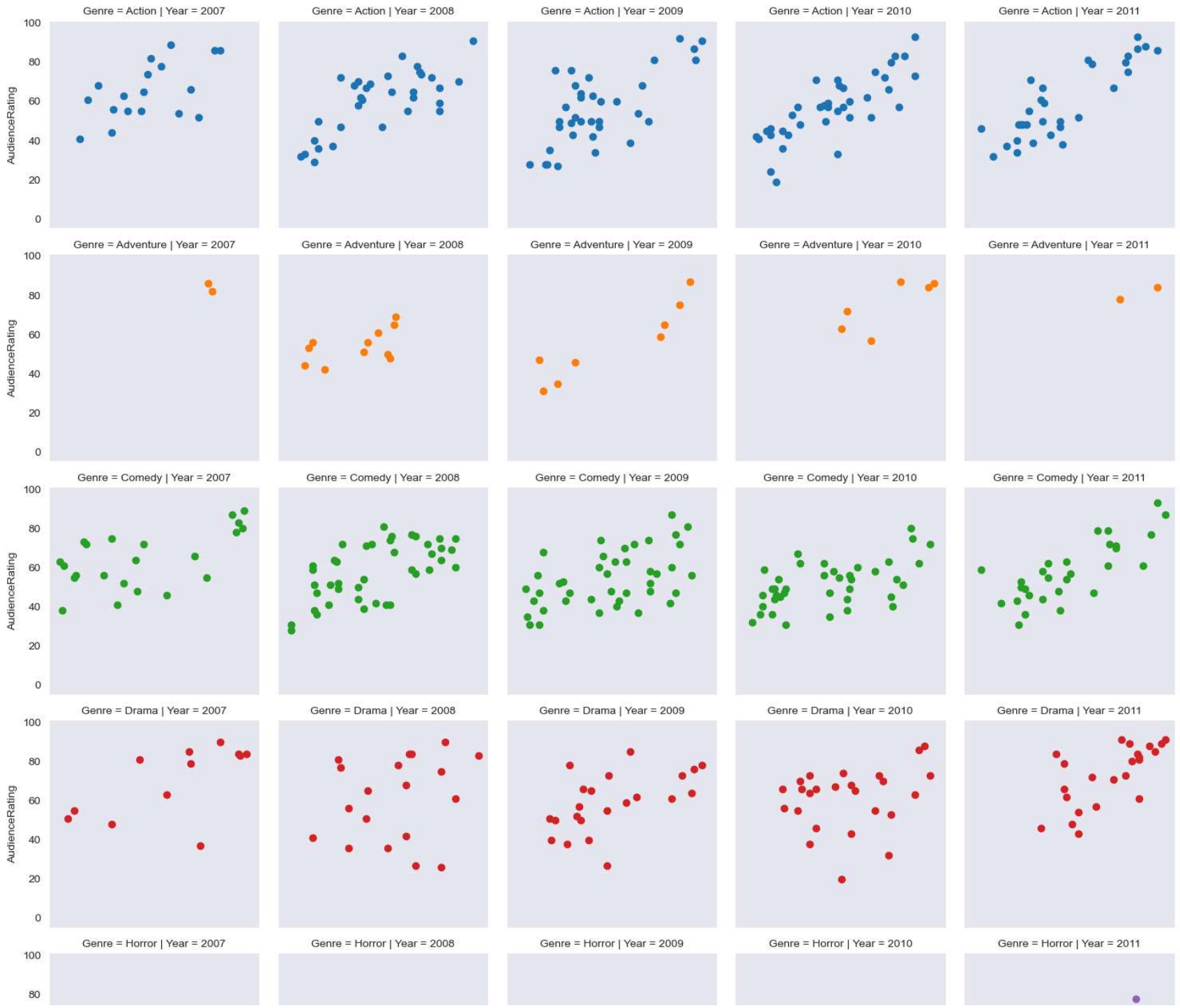

Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
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
1.0
0.8
0.6
0.4
0.2
0.0

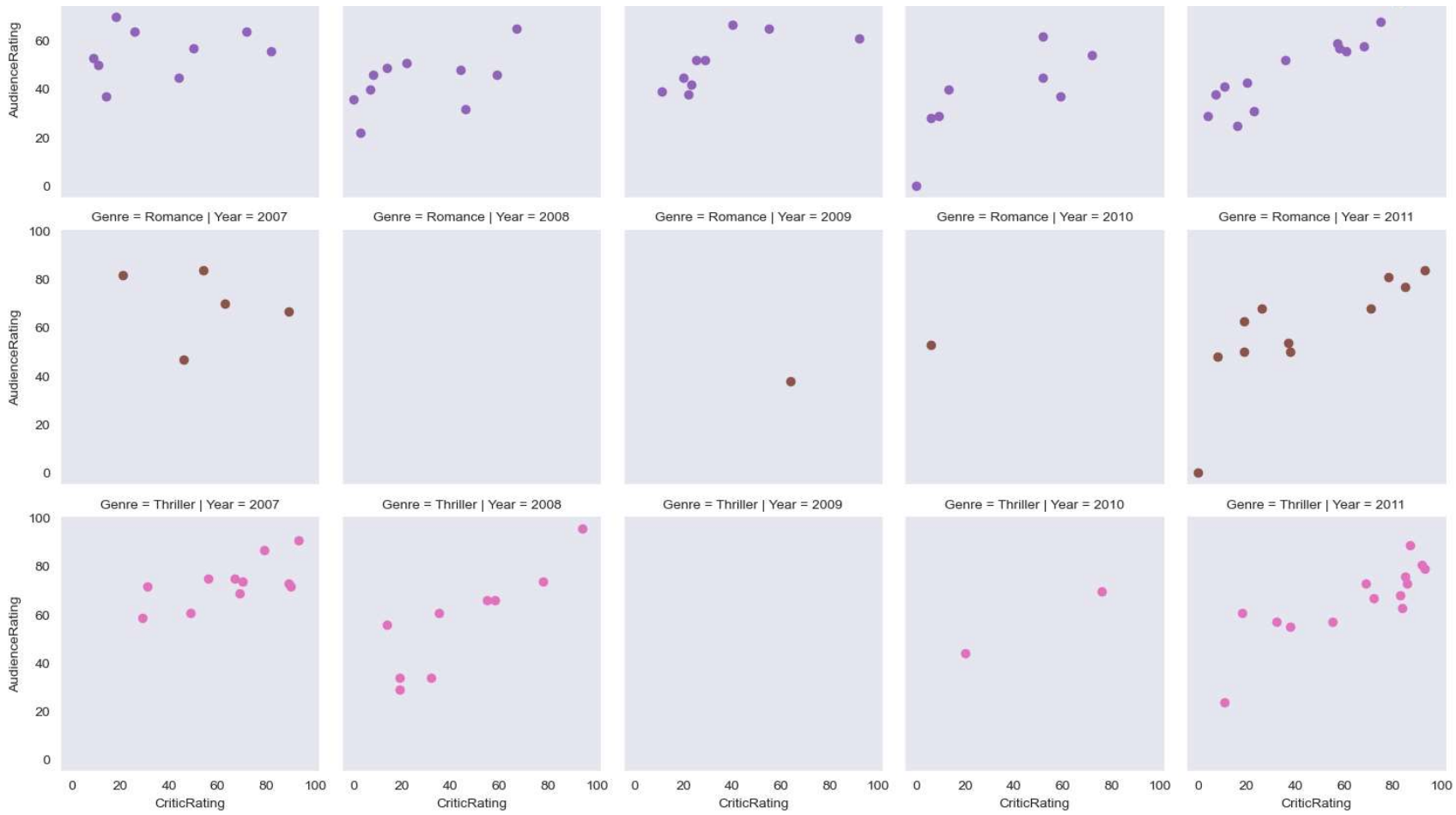
In [95]:
```
plt.scatter(movies.CriticRating,movies.AudienceRating)
```

Out[95]:
```
<matplotlib.collections.PathCollection at 0x11f779158e0>
```



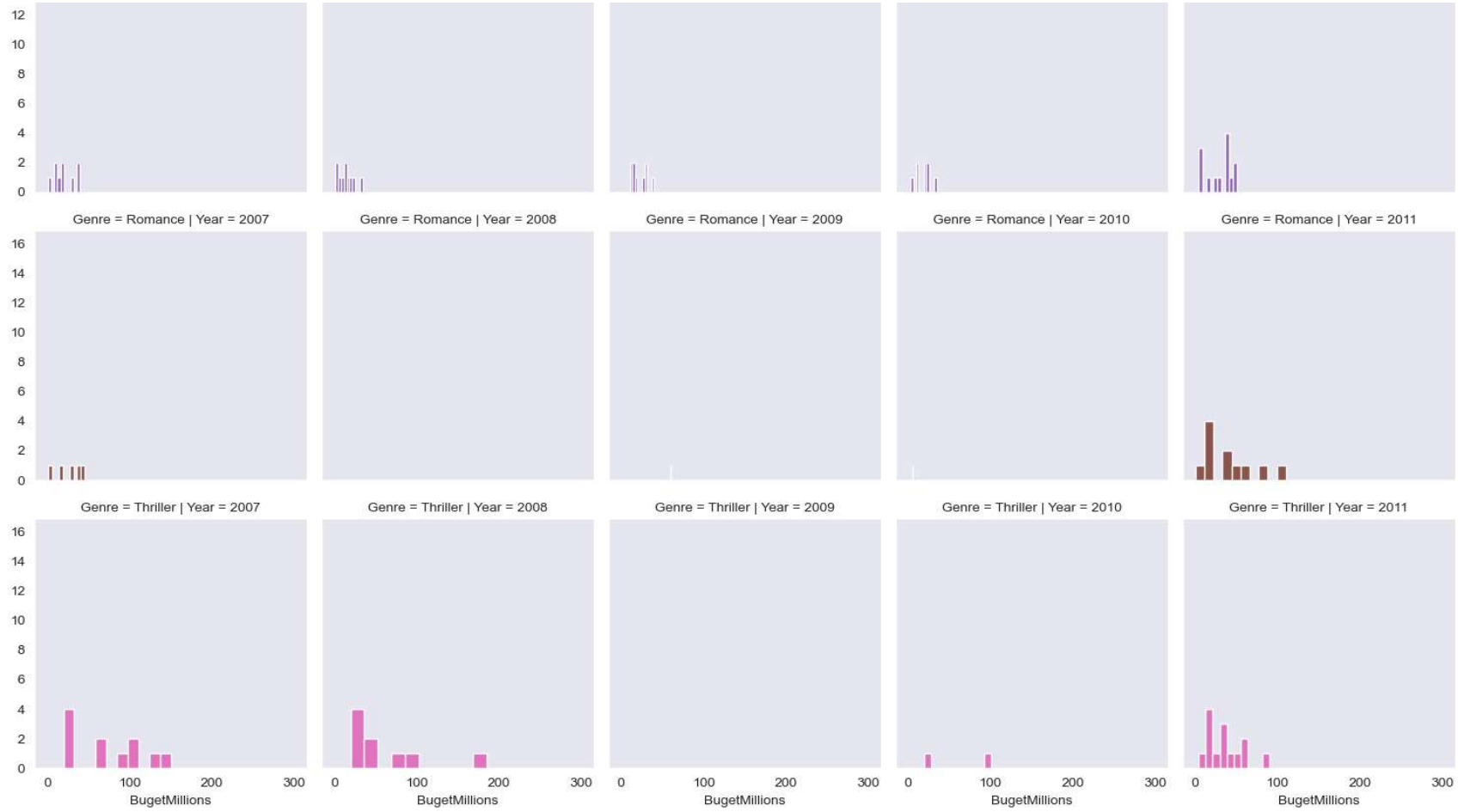
In [96]:
```
g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre')
g = g.map(plt.scatter, 'CriticRating', 'AudienceRating' ) #scatterplots are mapped in facetgrid
```


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
AudienceRating
0
20
40
60
80
100

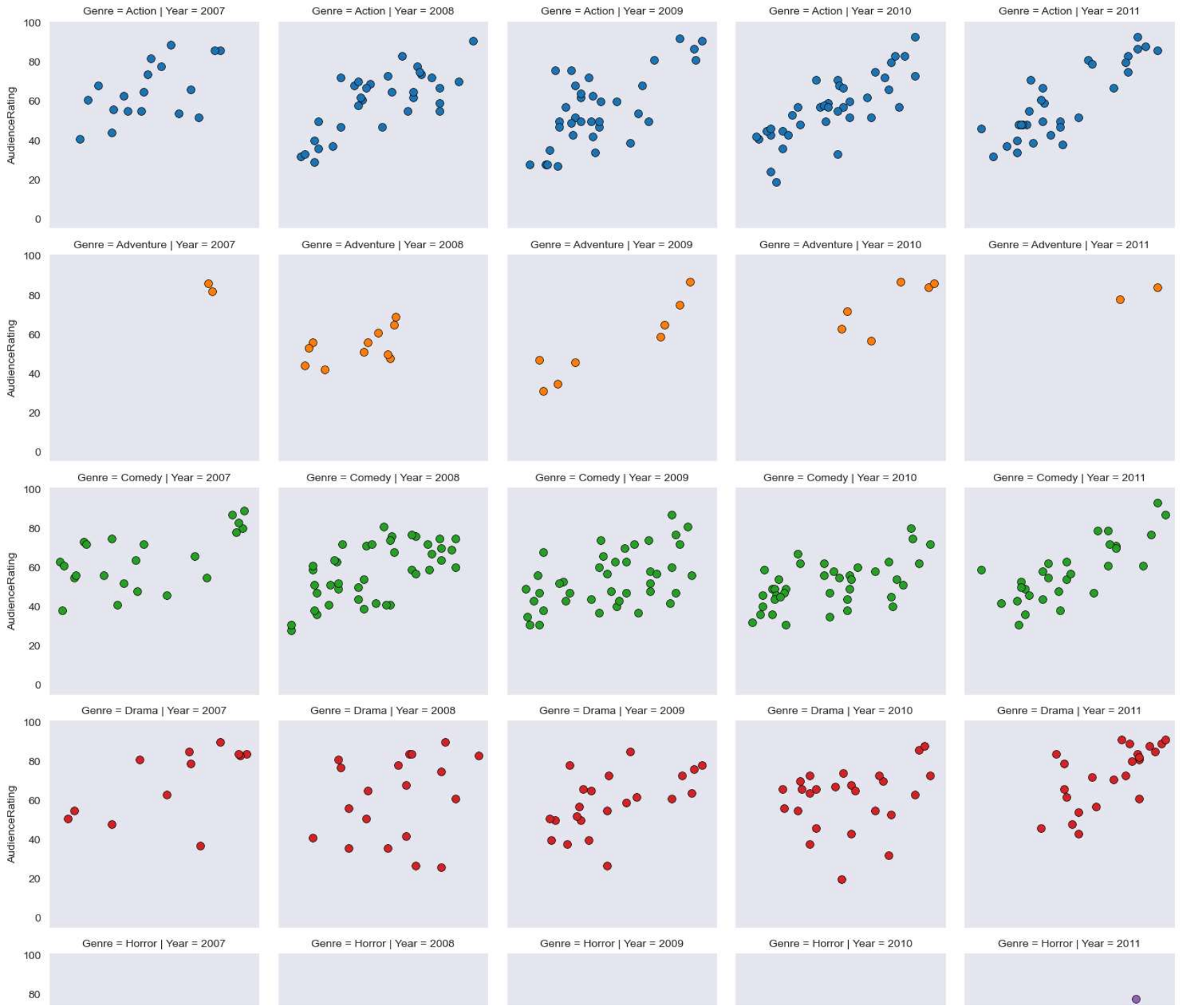


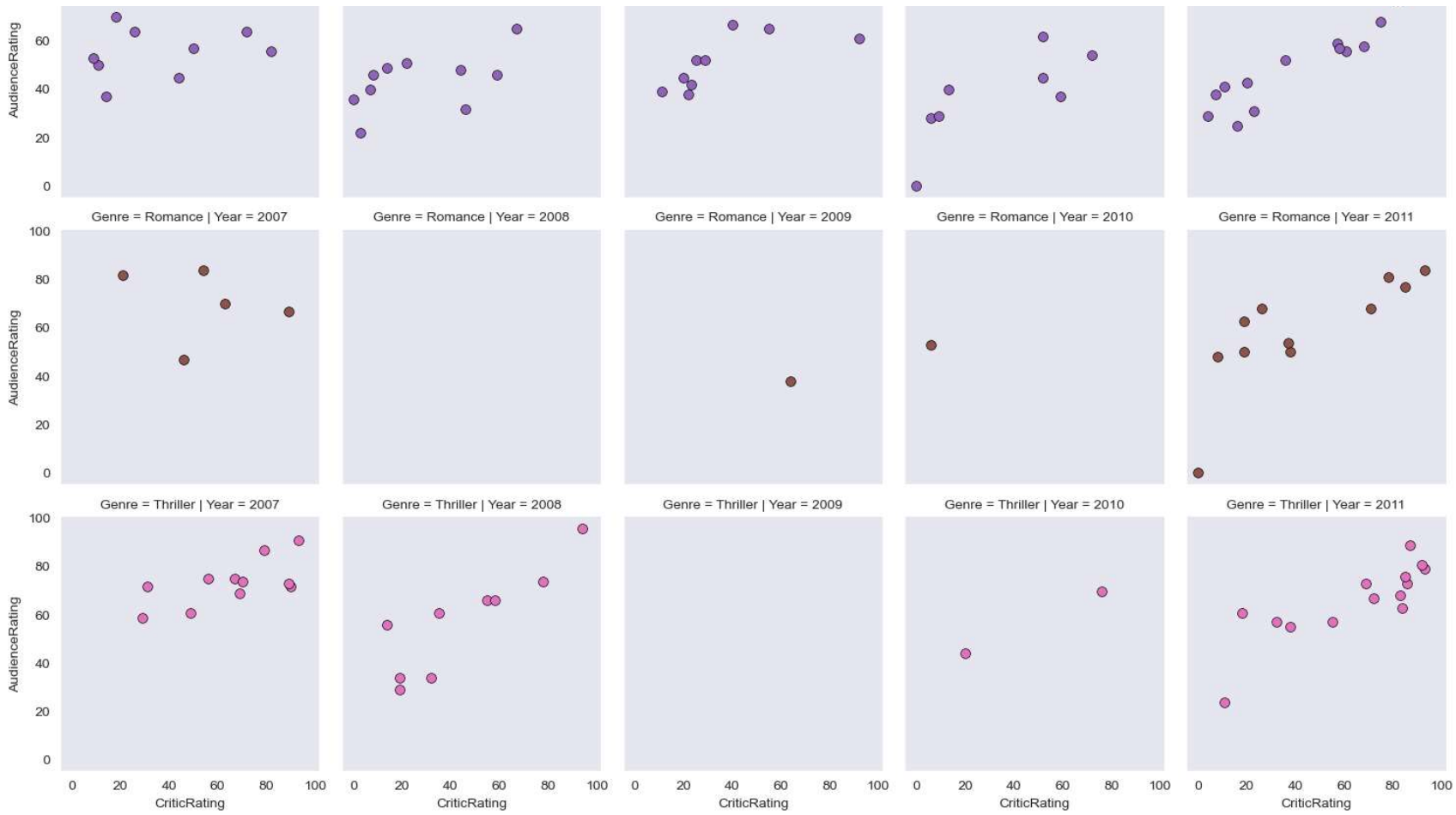
In [98]:
```
# you can populated any type of chat.

g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre')
g = g.map(plt.hist, 'BugetMillions') #scatterplots are mapped in facetgrid
```


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
0
2
4
6
8
10
12
14
16



In [99]:
```
#
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 ) #scatterplots are mapped in facetgrid
```

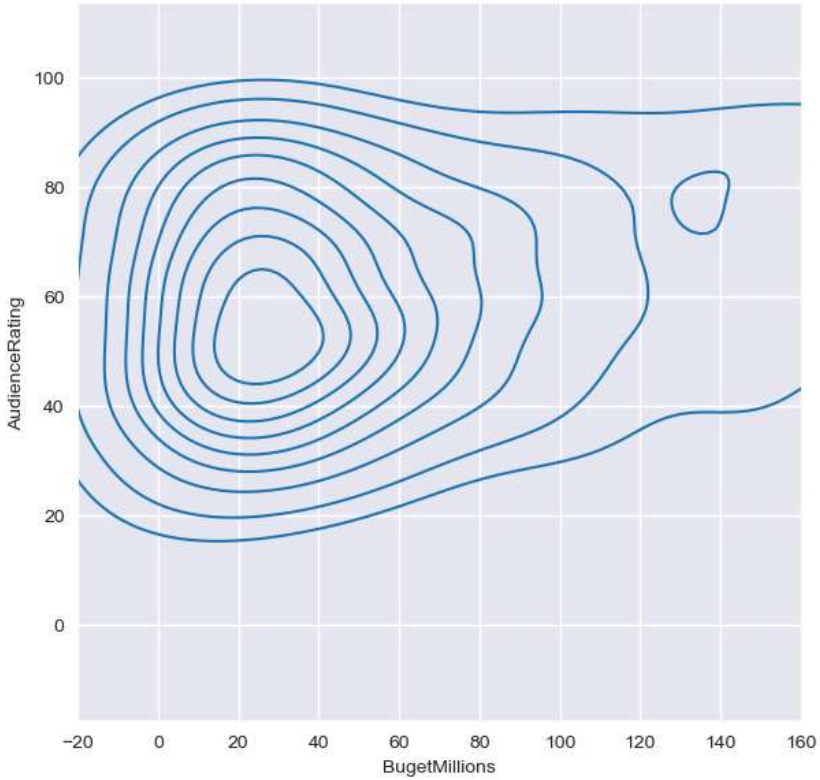

Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
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
AudienceRating
0
20
40
60
80
100

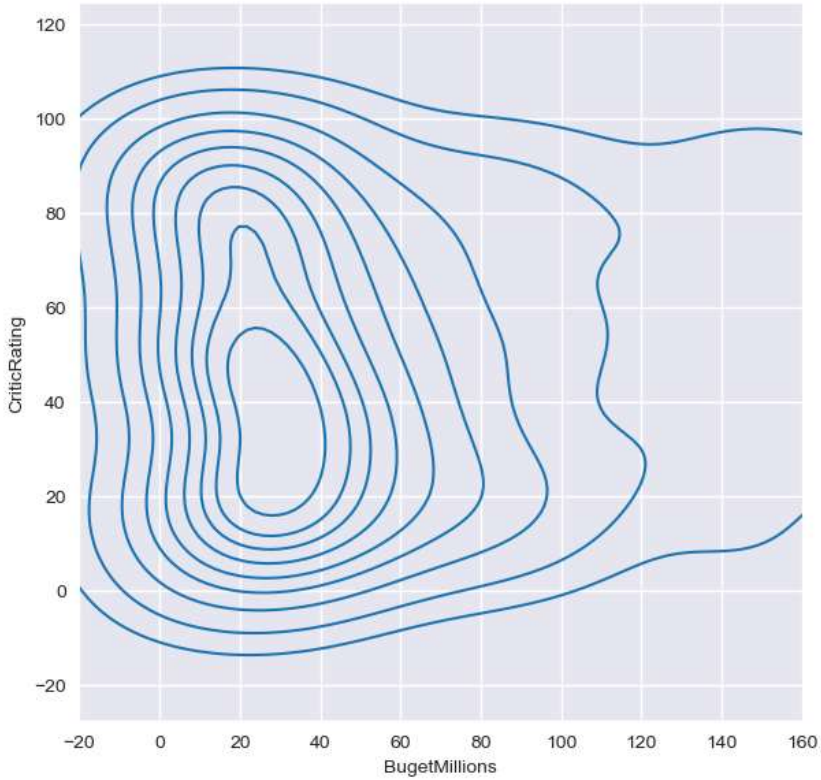


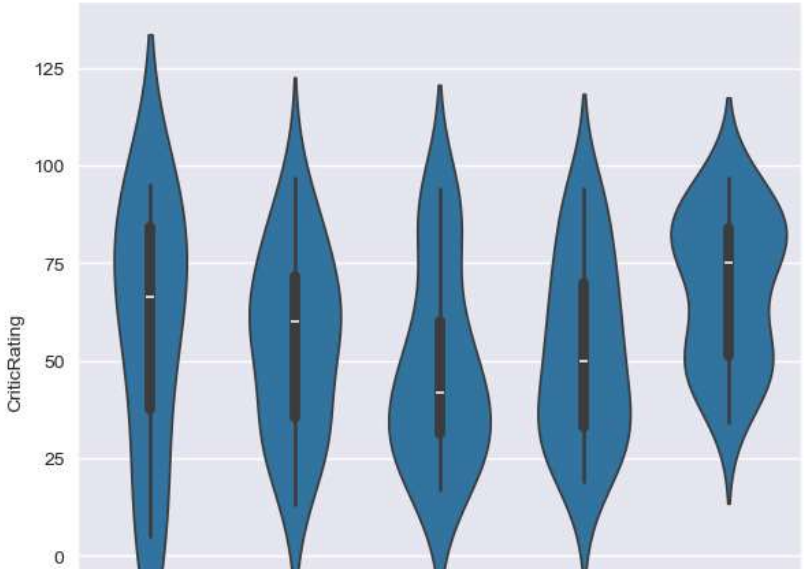
In [103…

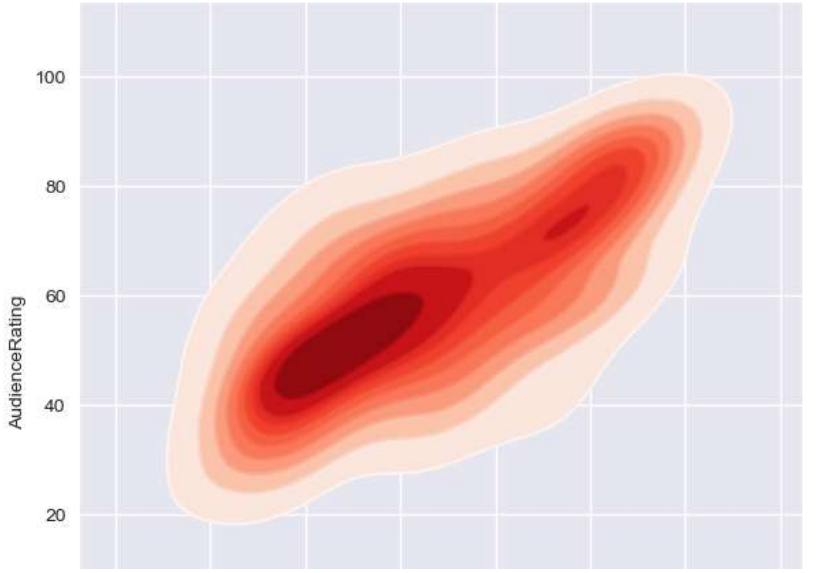
```
# python is not vectorize programming language
# Building dashboards (dashboard - combination of chats)

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

k1 = sns.kdeplot(data=movies,x=movies.BugetMillions,y=movies.AudienceRating,ax=axes[0,0])
k2 = sns.kdeplot(data=movies,x=movies.BugetMillions,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])
```

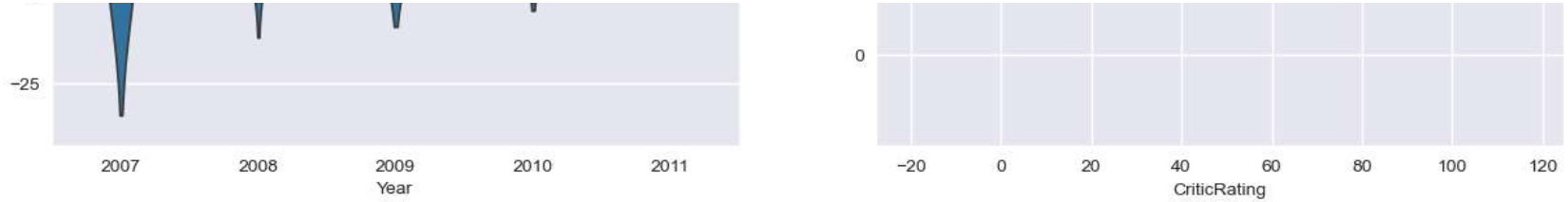
```
k4 = sns.kdeplot(data=movies,x=movies.CriticRating,y=movies.AudienceRating,shade = True,shade_lowest=False,cmap='Reds

k4b = sns.kdeplot(data=movies,x=movies.CriticRating,y=movies.AudienceRating,cmap='Reds',ax = axes[1,1])

plt.show()
```


AudienceRating
BugetMillions


CriticRating
BugetMillions

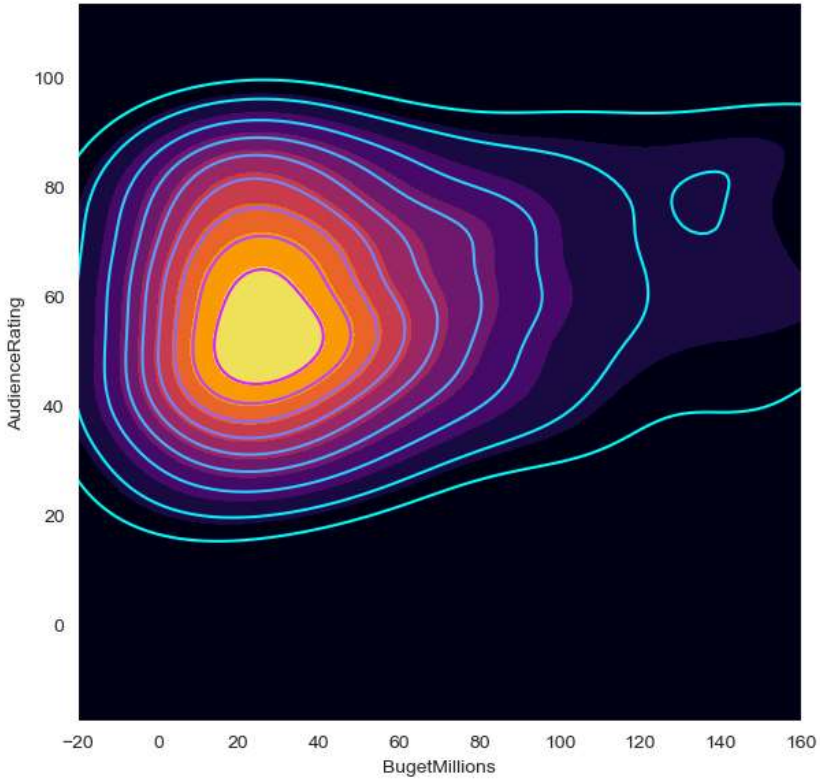

CriticRating

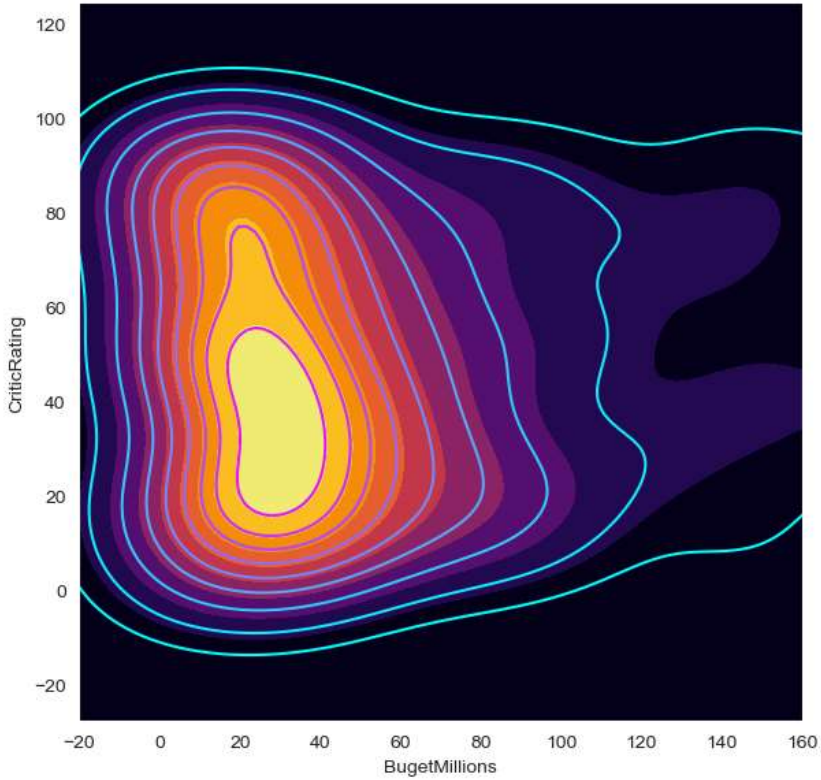

AudienceRating

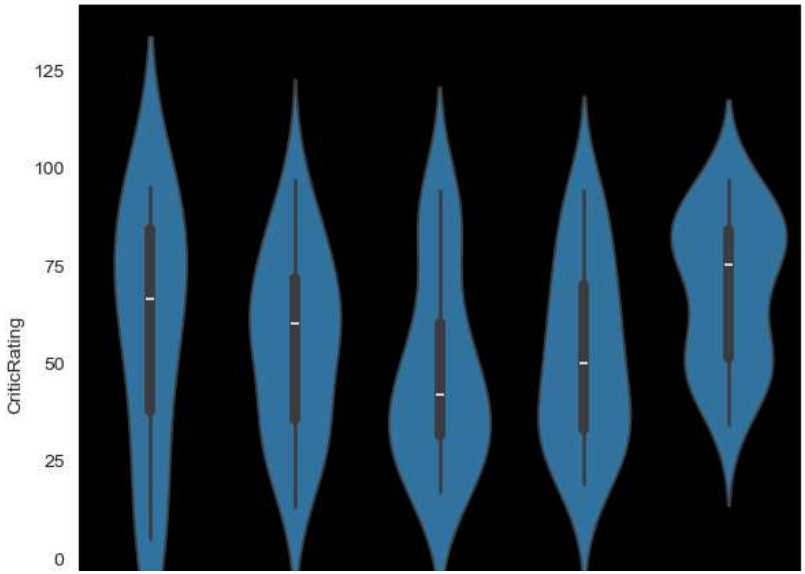


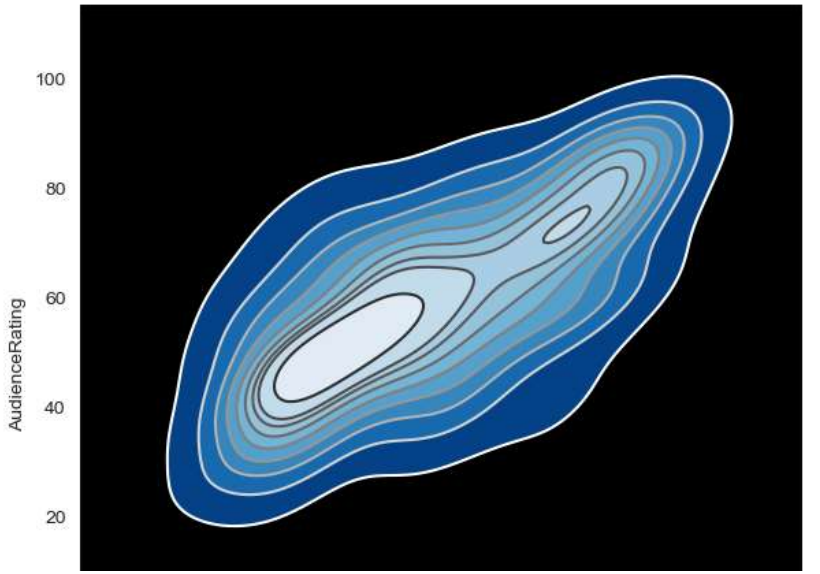
In [106…

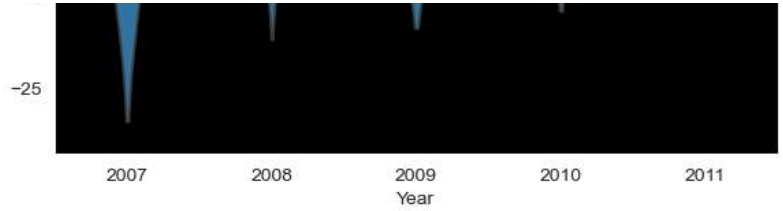
```
# How can you style your dashboard using different color map
# python is not vectorize programming language
# Building dashboards (dashboard - combination of chats)
sns.set_style('dark',{'axes.facecolor':'black'})
f, axes = plt.subplots (2,2, figsize = (15,15))
#plot [0,0]
k1 = sns.kdeplot(x=movies.BugetMillions,y=movies.AudienceRating, \
shade = True, shade_lowest=True,cmap = 'inferno', \
ax = axes[0,0])
k1b = sns.kdeplot(x=movies.BugetMillions,y=movies.AudienceRating, \
cmap = 'cool',ax = axes[0,0])
#plot [0,1]
k2 = sns.kdeplot(x=movies.BugetMillions,y=movies.CriticRating,\
shade=True, shade_lowest=True, cmap='inferno',\
ax = axes[0,1])
k2b = sns.kdeplot(x=movies.BugetMillions,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))
plt.show()
```


AudienceRating
BugetMillions
100
80
60
40
20
0
−20
0
20
40
60
80
100
120
140
160


CriticRating
BugetMillions
120
100
80
60
40
20
0
−20
−20
0
20
40
60
80
100
120
140
160


CriticRating
125
100
75
50
25
0


AudienceRating
100
80
60
40
20



Final discussion what we learn so far - 1> category datatype in python 2> jointplots 3> histogram 4> stacked histograms 5> Kde plot 6> subplot 7> violin plots 8> Factet grid 9> Building dashboards

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: