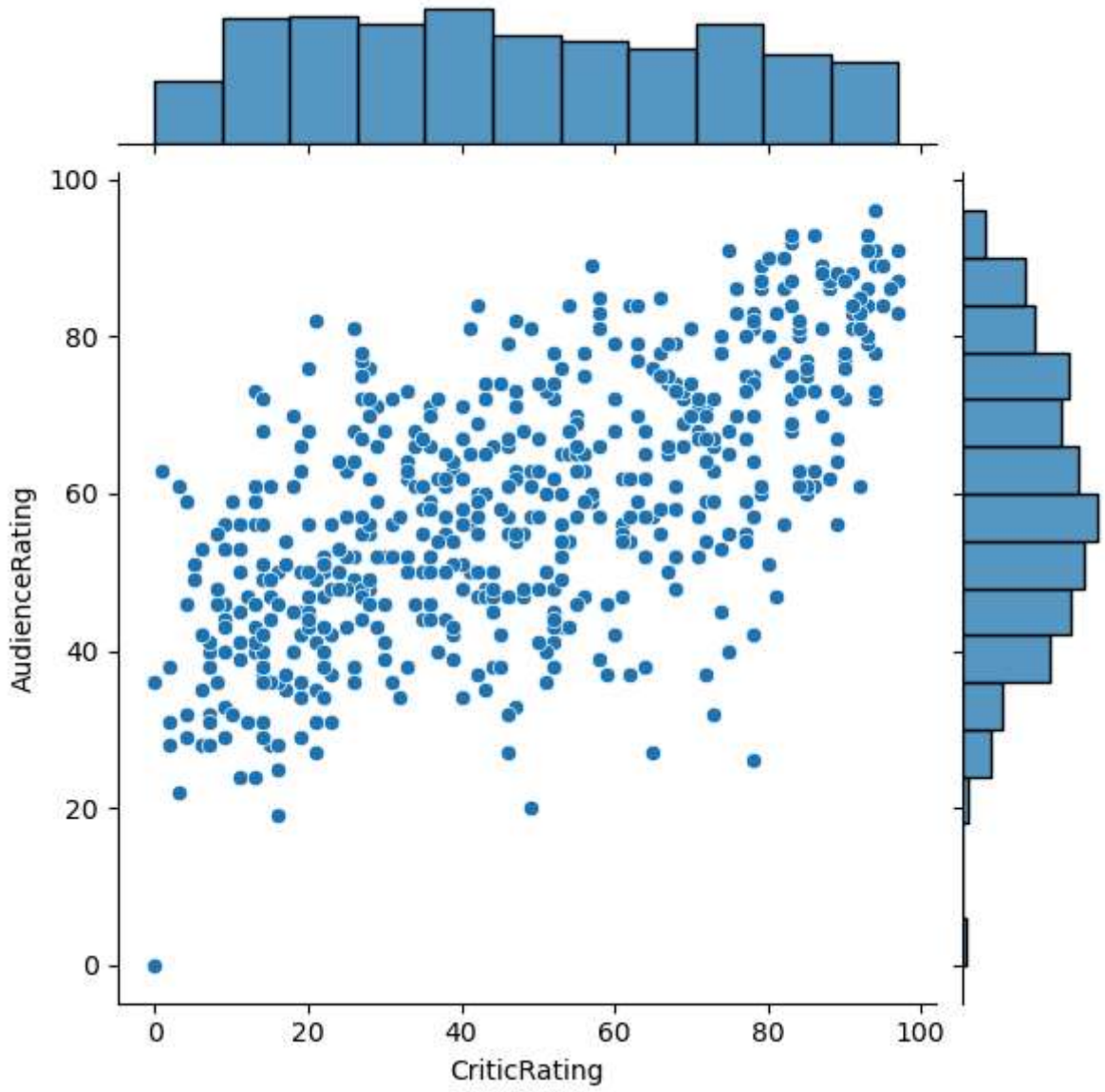
Out[24]:

	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

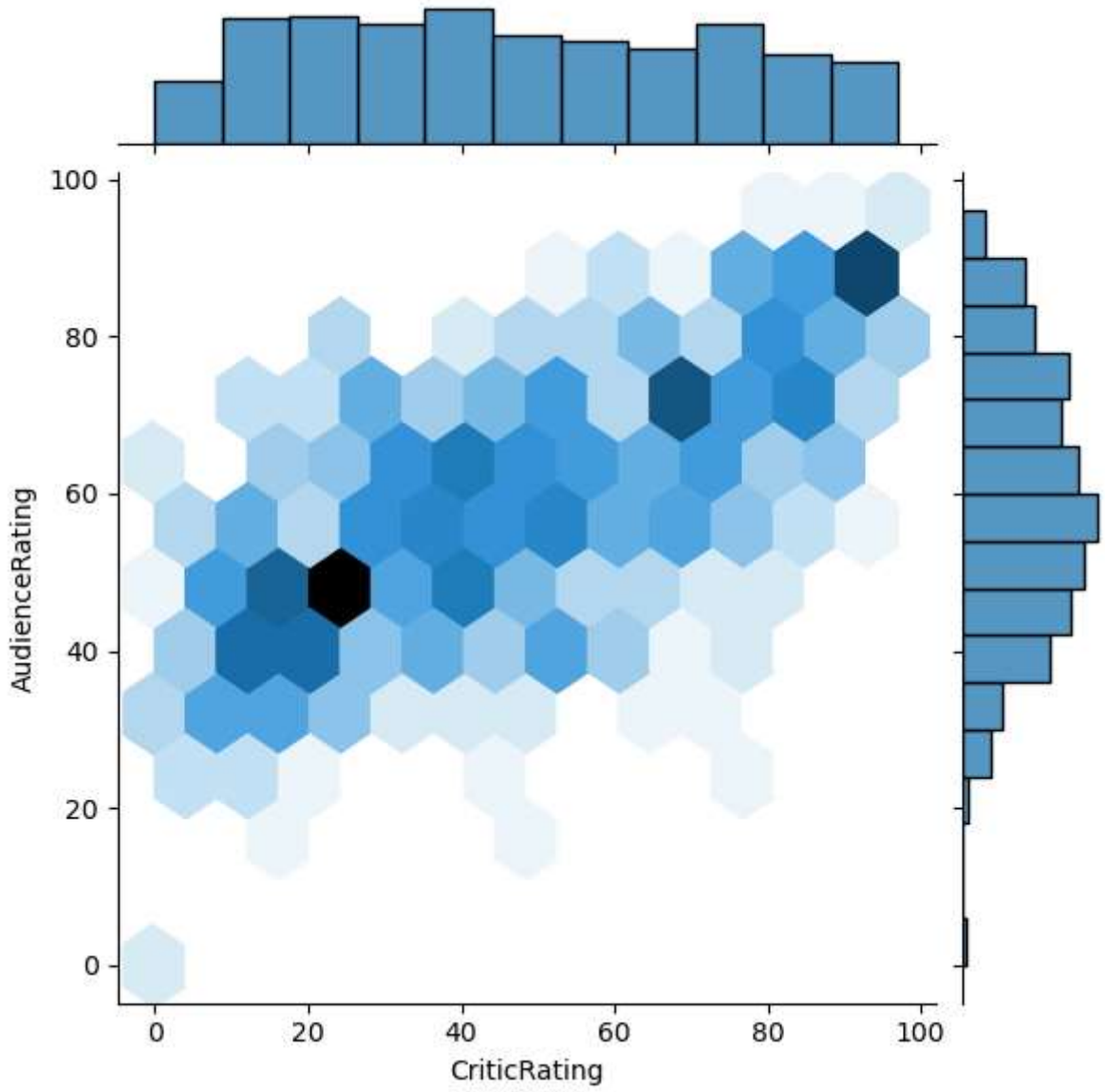
In [25]:
```
# How to working with joint plots
from matplotlib import pyplot as plt # visualization
import seaborn as sns # advanced visualization

#%matplotlib inline #All THE PLOT SHOULD INSIDE THE LINE

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

- basically joint plot is a scatter plot & it find the relation b/w audience & critics
- also if you look up you can find the uniform distribution (critics) and normal distribution (audience)

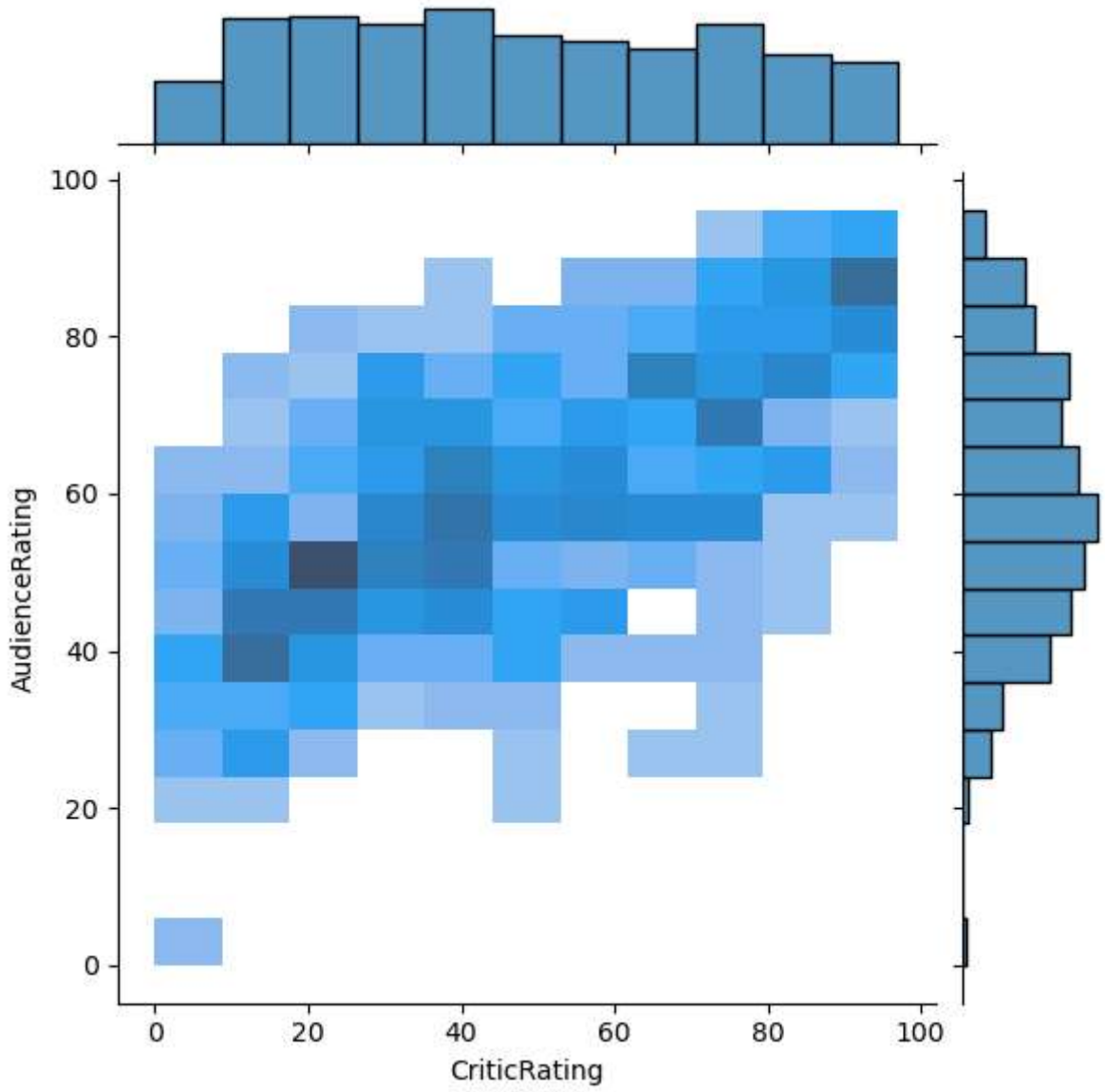
In [26]:
```
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating') # 24th MAY 2025

# Audience rating is more dominant then critics rating
# Based on this we find out as most people are most liklihood to watch audience rating & less likely to watch
# there is positive coreleation between 2 attributes
# Let me explain the excel - if you filter audience rating & critic rating. critic rating has very low value.
```



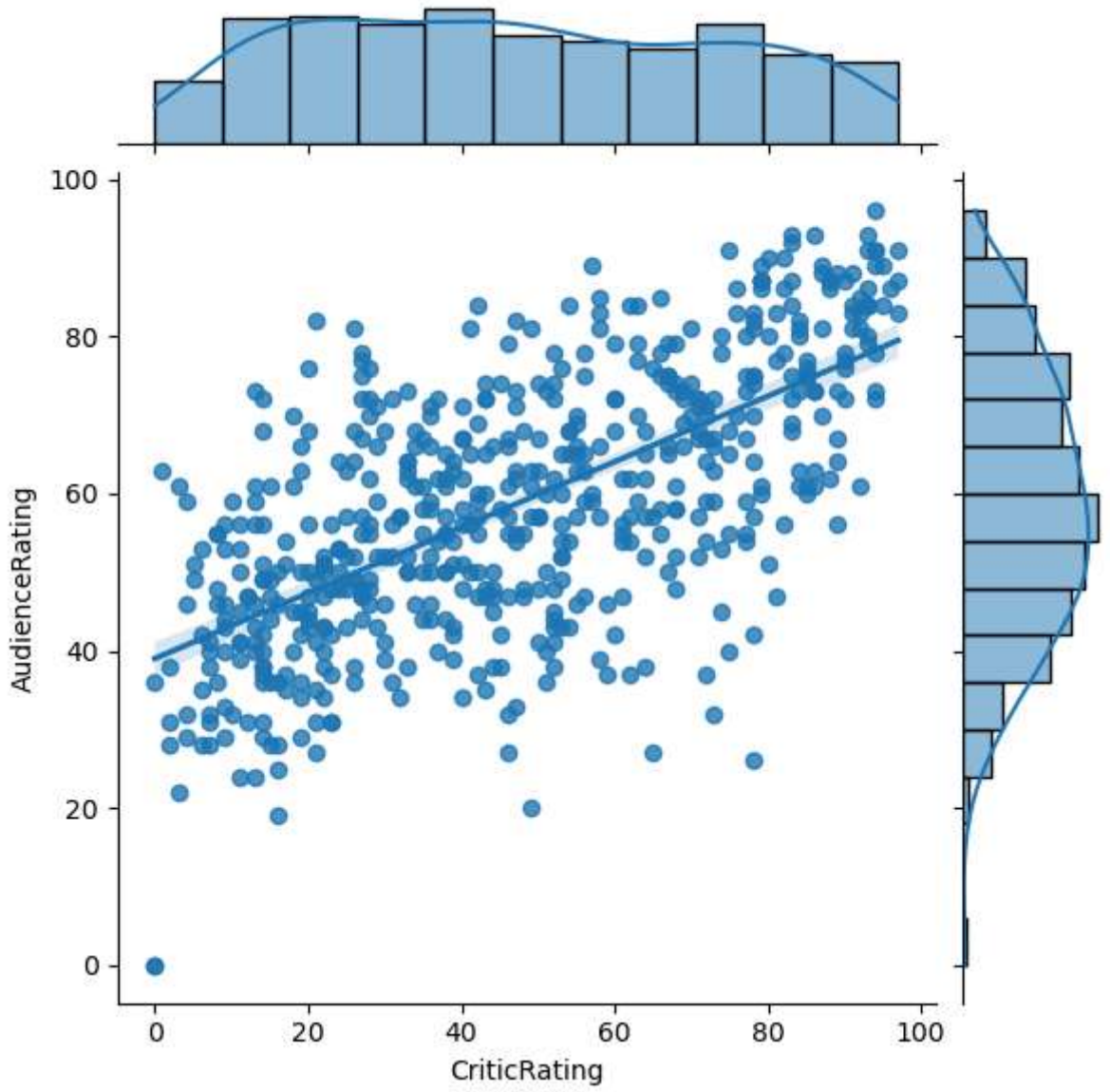
In [27]:
```
#j = sns.jointplot( data = movies, x = 'CriticRating', y = 'AudienceRating', kind='hex')
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating',kind='hex')
```



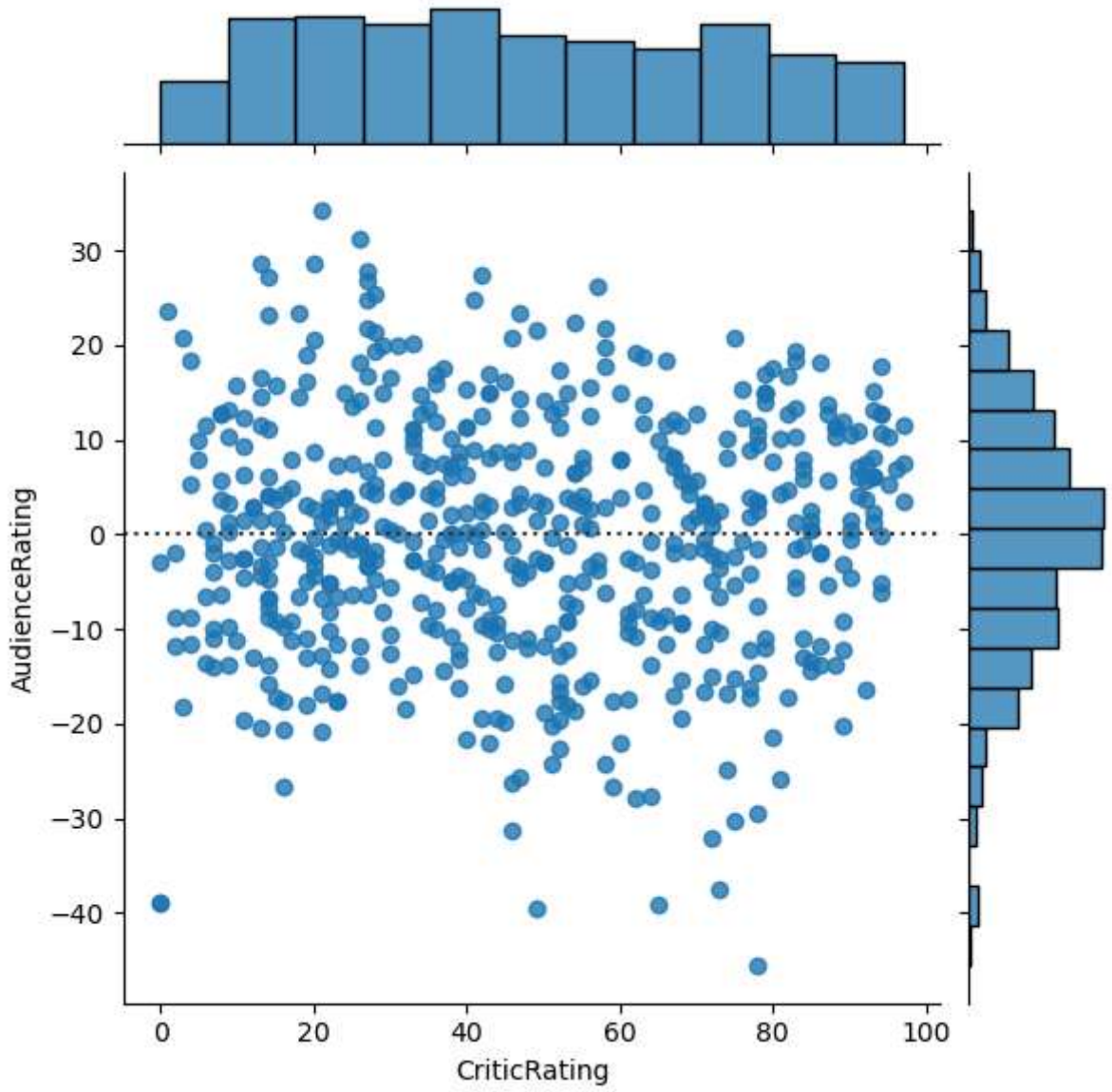
In [28]:
```
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating',kind='hist')
```



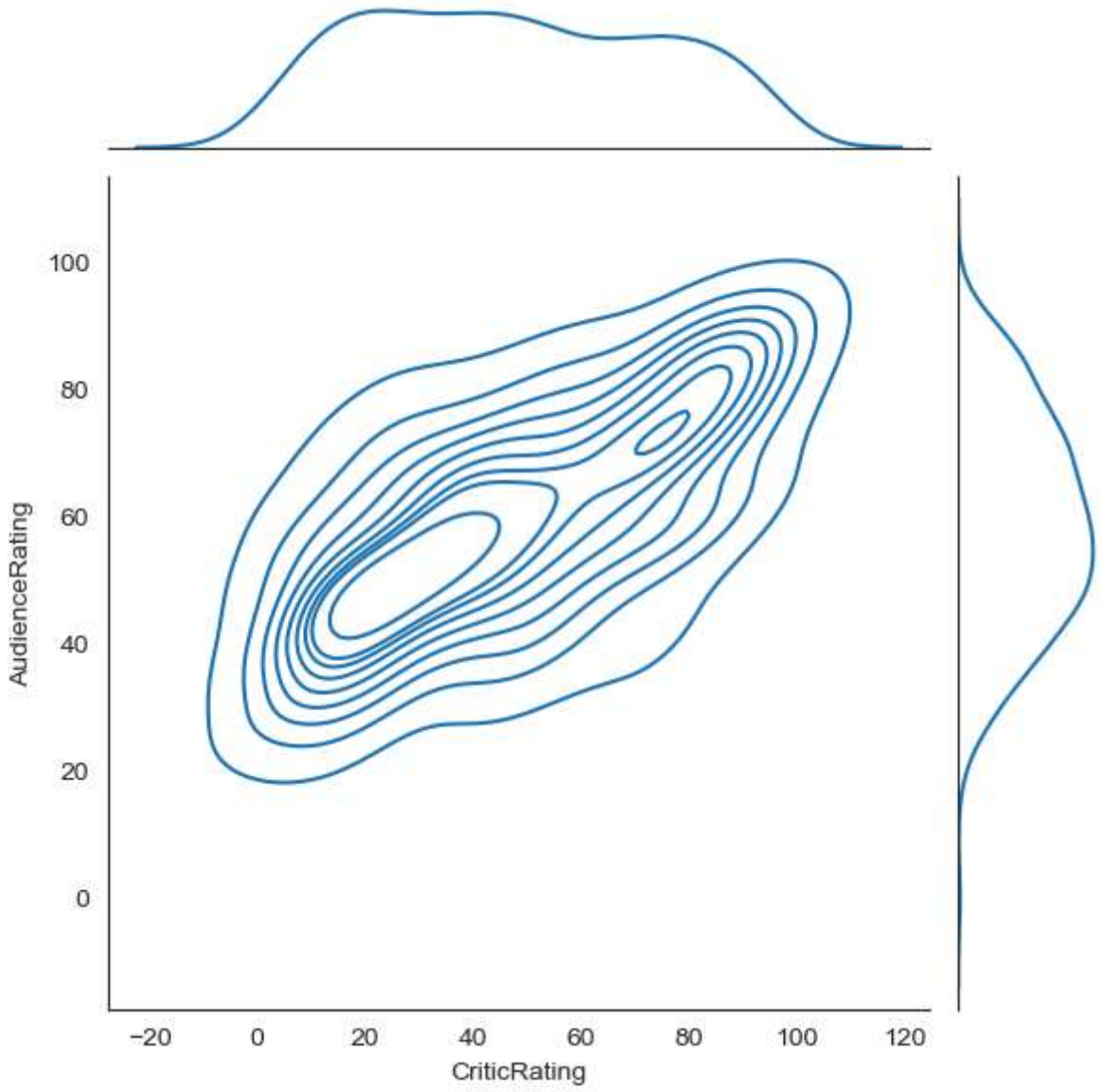
In [29]:
```
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating',kind='reg')
```



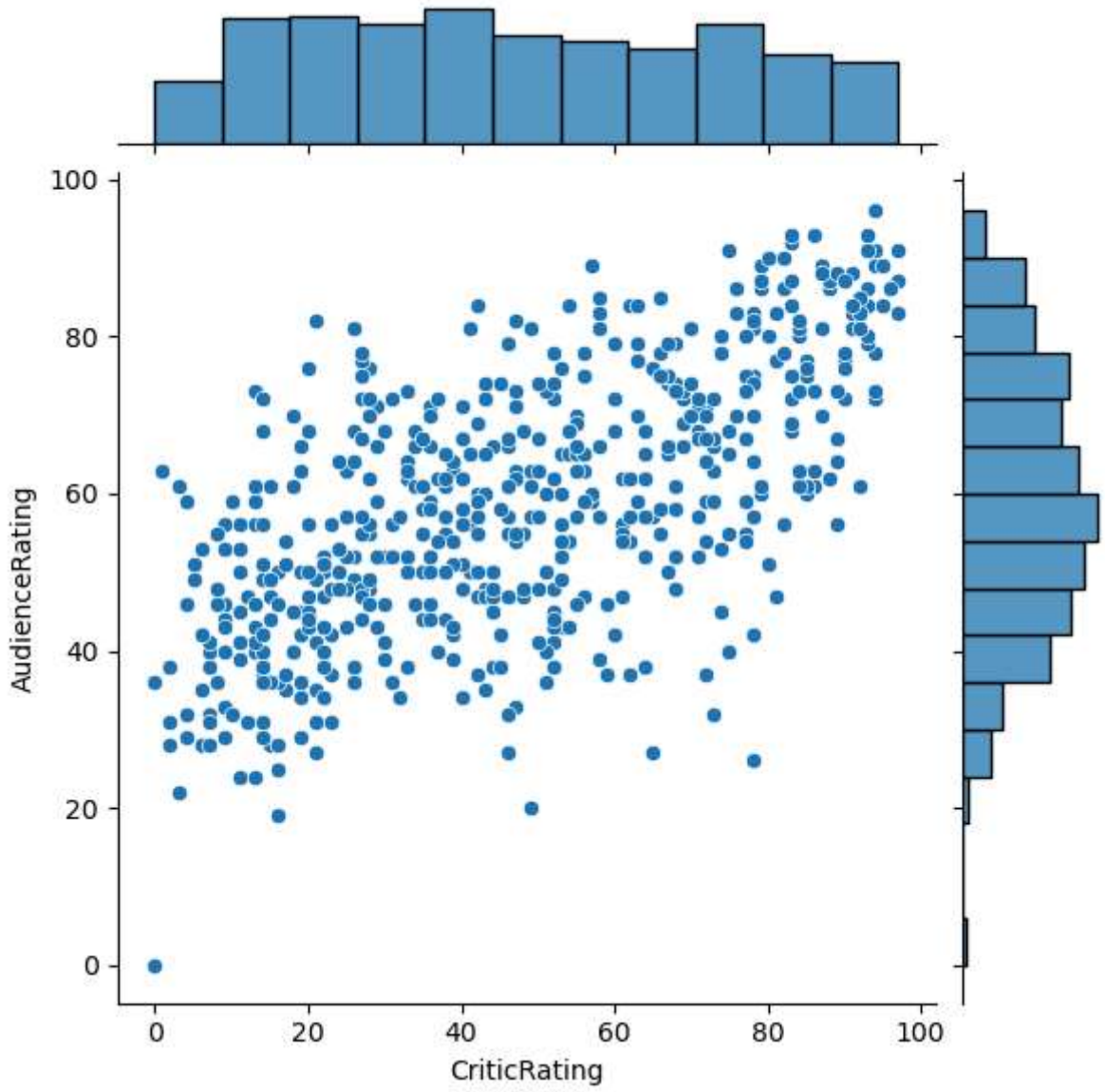
In [30]:
```
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating',kind='resid')
```



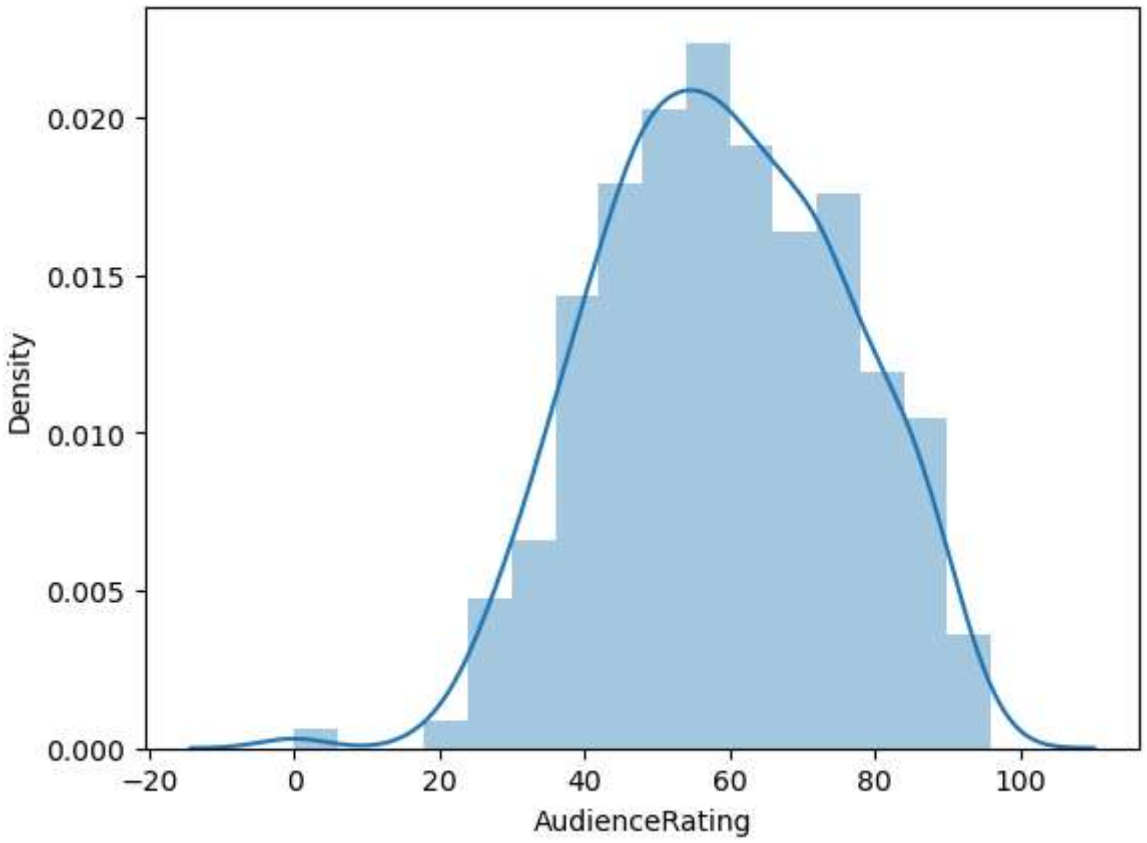
In [54]:
```
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating',kind='kde')
```



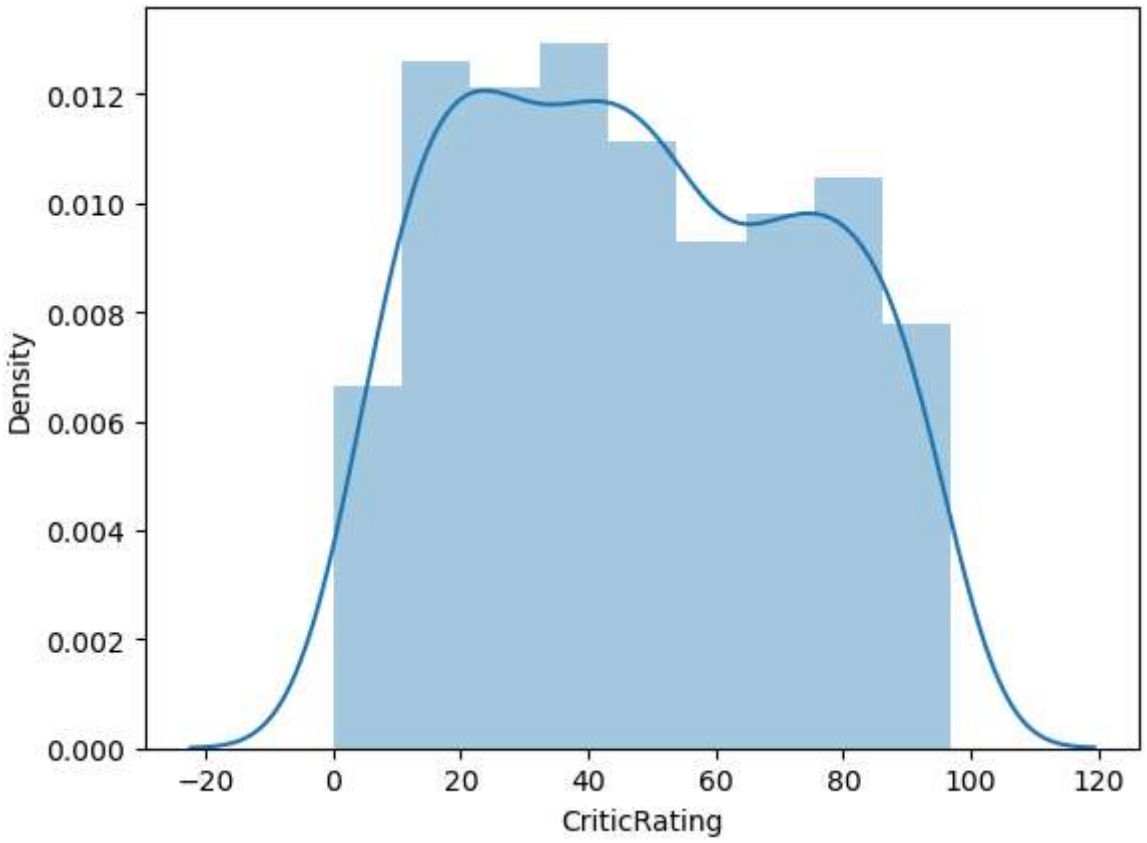
In [31]:
```
j = sns.jointplot(data = movies, x = 'CriticRating', y = 'AudienceRating',kind='scatter')
```



In [32]:
```
# Histrograms

# <<< chat1
# distribution plot
m1 = sns.distplot(movies.AudienceRating)
#y - axis generated by seaborn automatically that is the powerfull of seaborn gallery
```

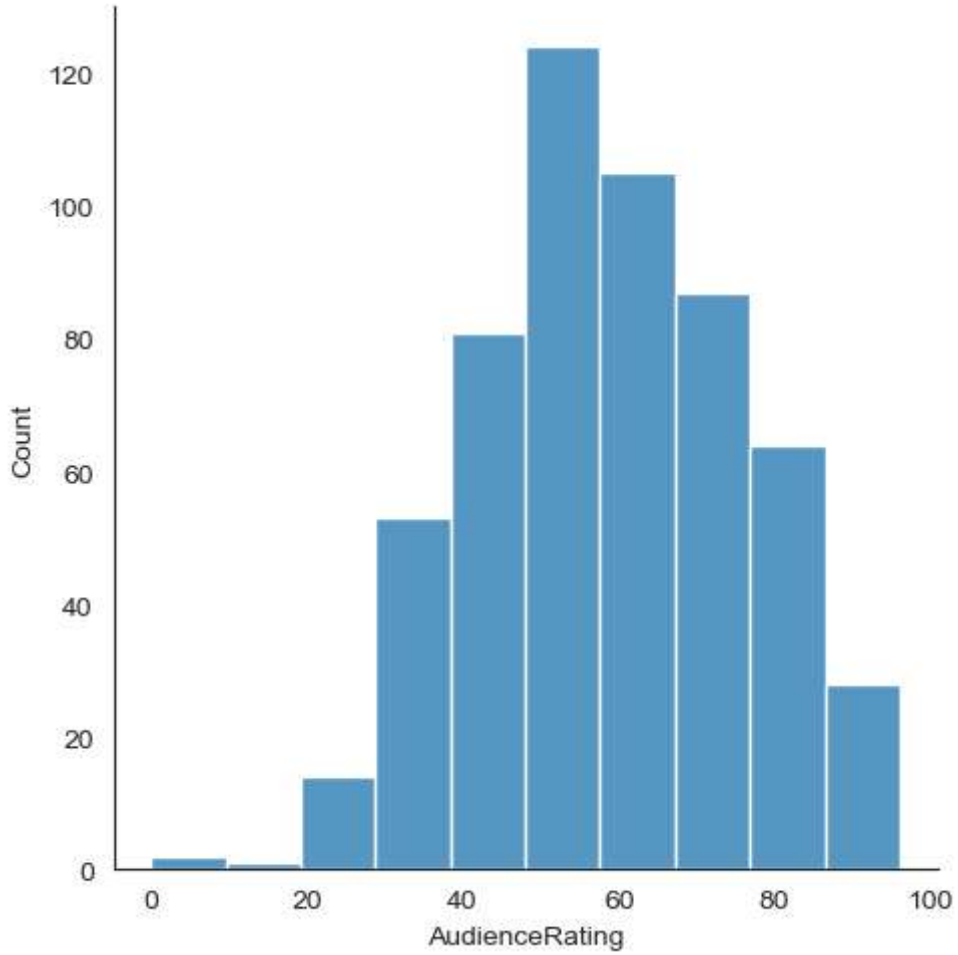


In [33]:
```
m1 = sns.distplot(movies.CriticRating)
```

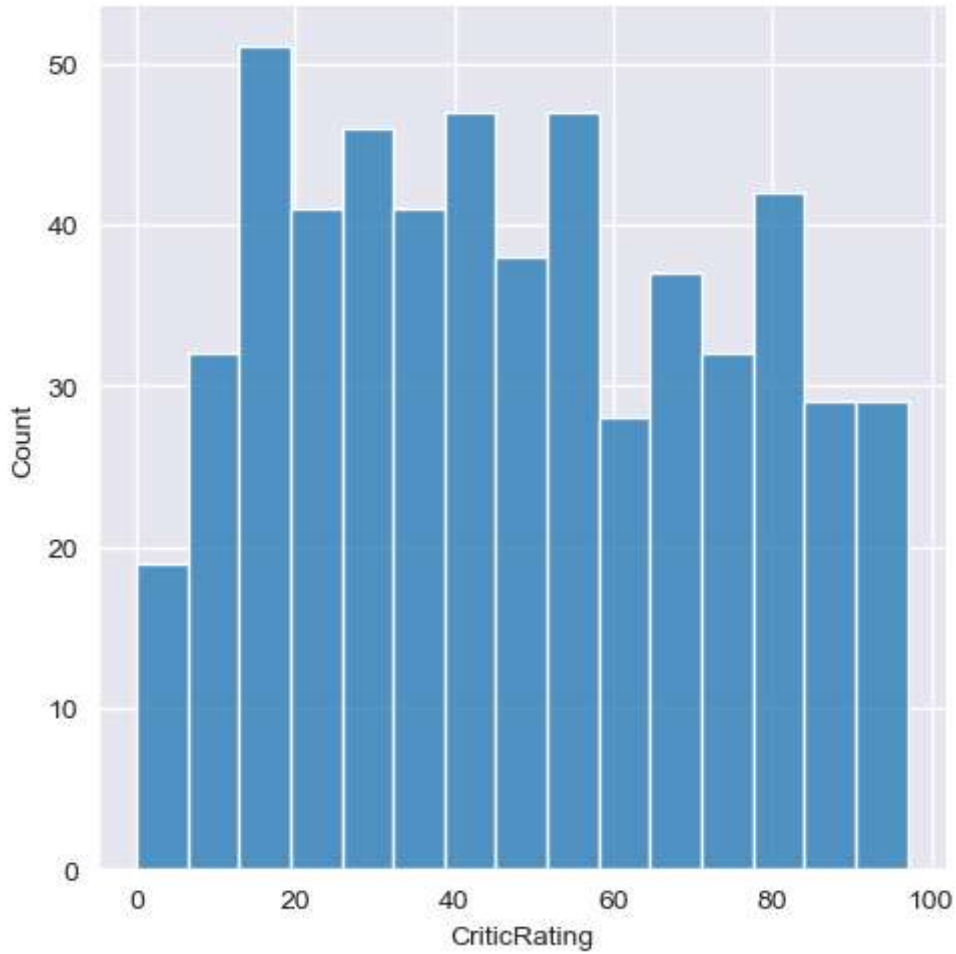


In [34]:
```
sns.set_style('darkgrid')
```

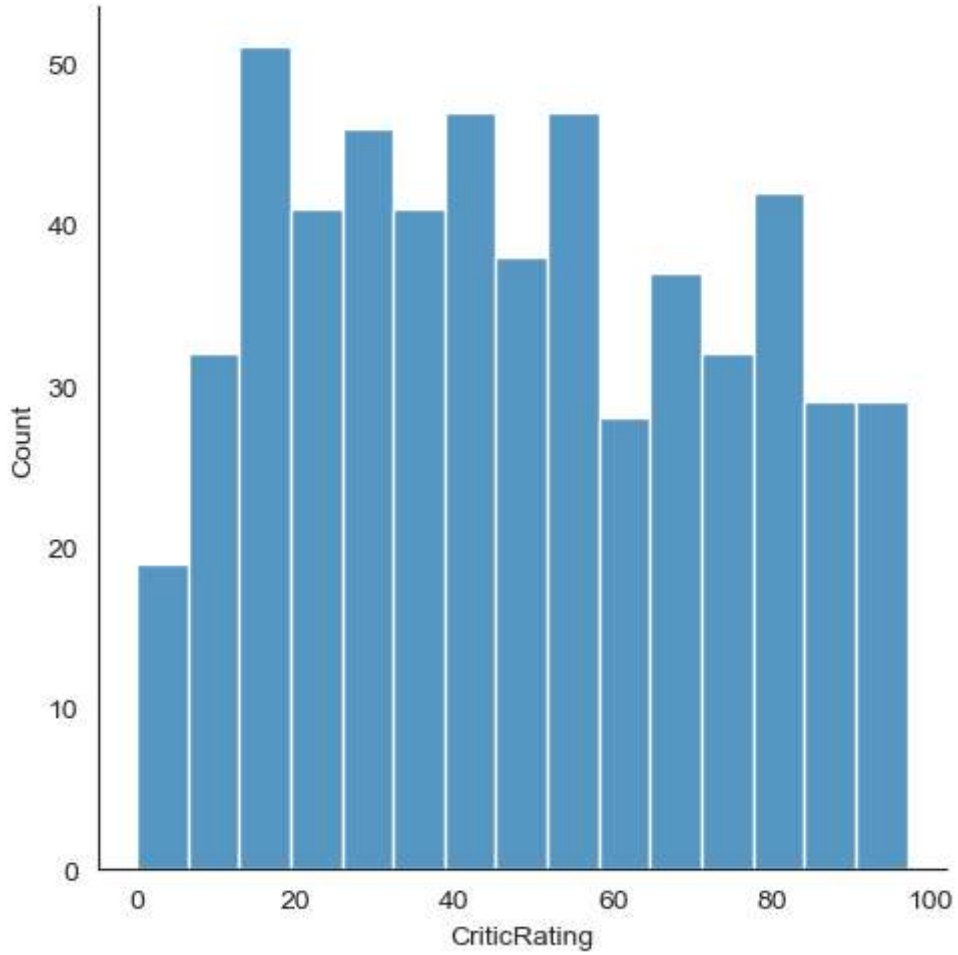
In [55]:
```
m2 = sns.displot(movies.AudienceRating, bins = 10)
```



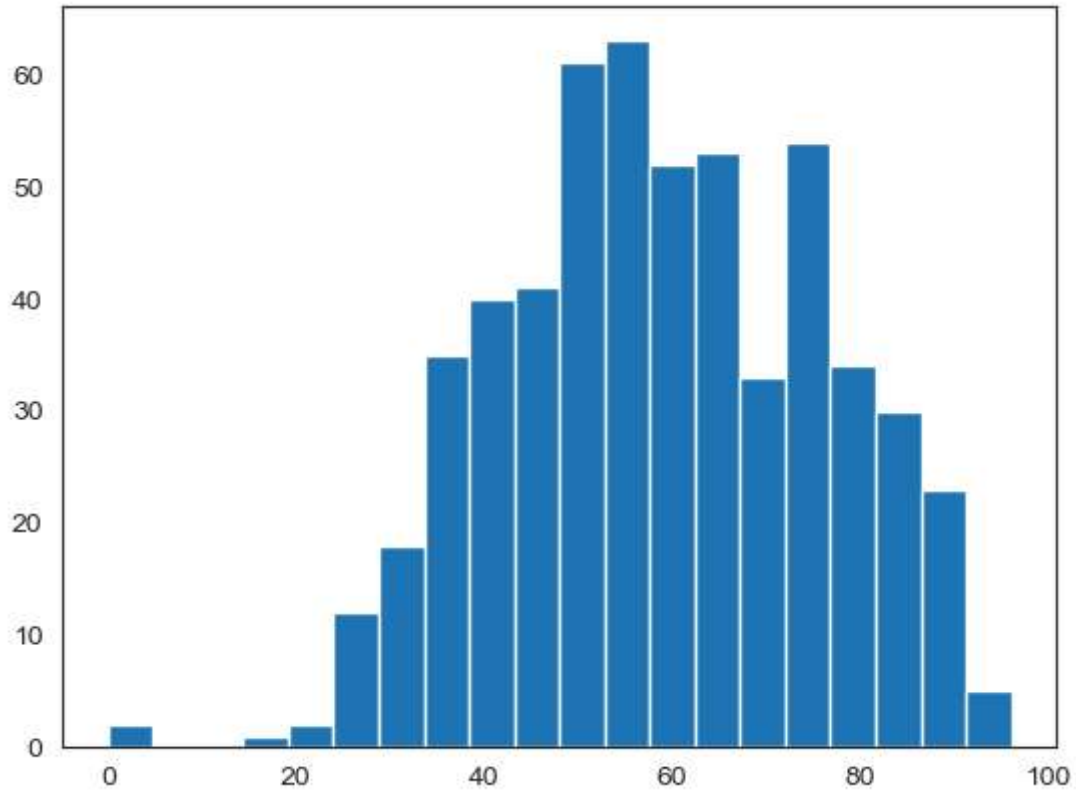
In [35]:
```
m2 = sns.displot(movies.CriticRating, bins=15)
```



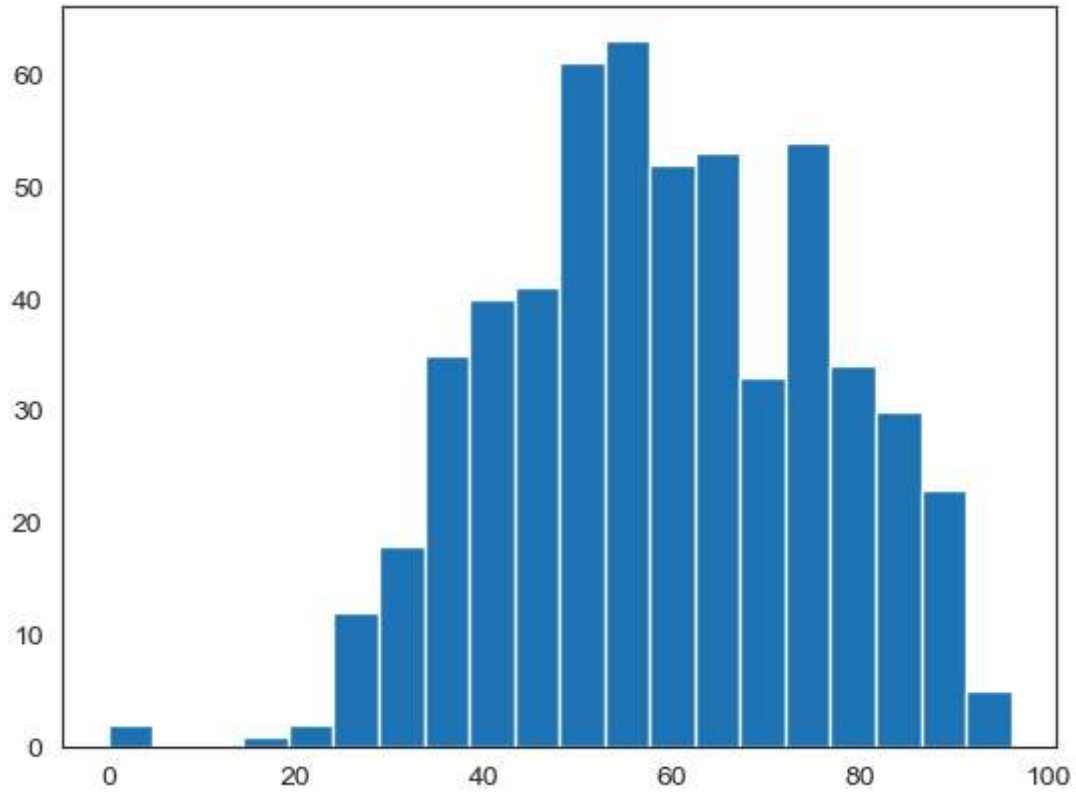
In [36]:
```
sns.set_style('white')
m2 = sns.displot(movies.CriticRating, bins=15)
```



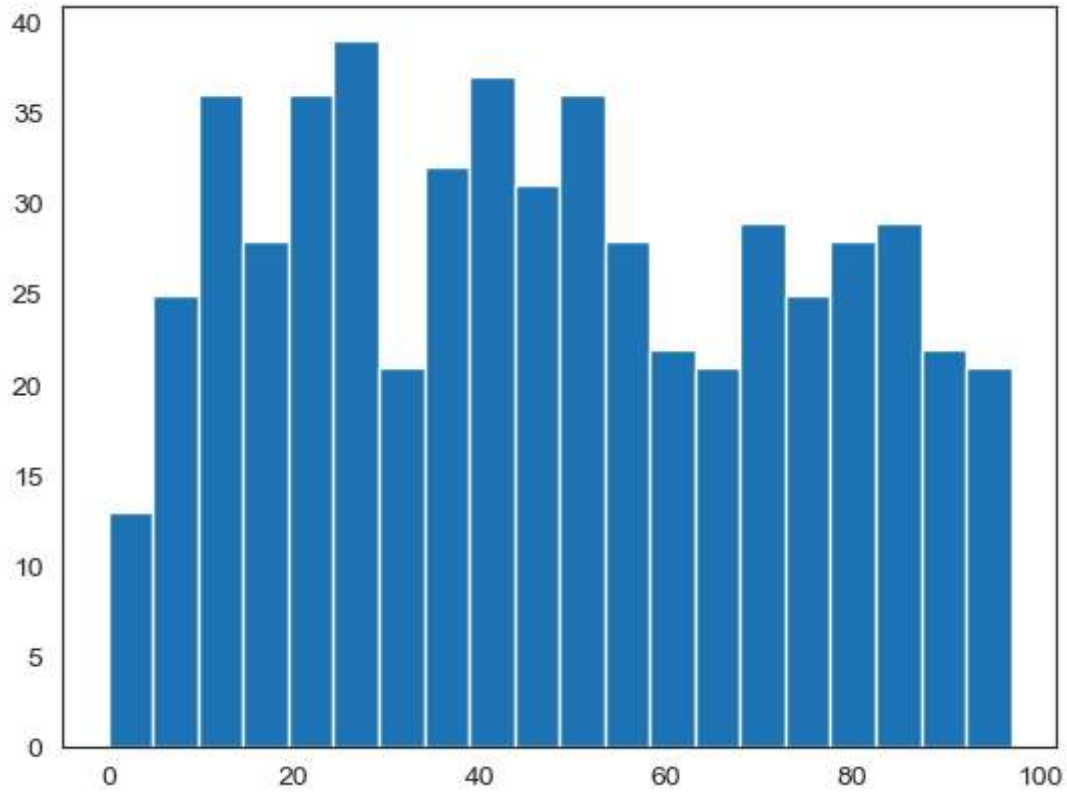
In [56]:
```
#sns.set_style('darkgrid')
n1 = plt.hist(movies.AudienceRating, bins=20)
```



In [57]:
```
sns.set_style('white') #normal distribution & called as bell curve
n1 = plt.hist(movies.AudienceRating, bins=20)
```

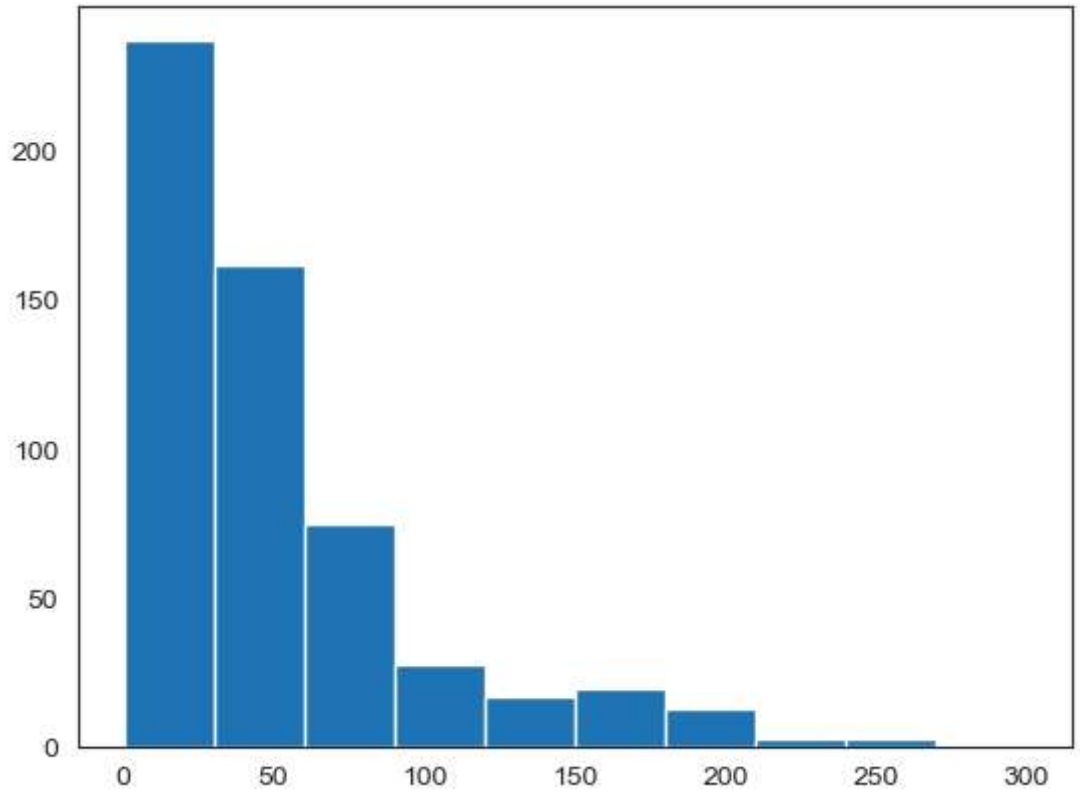


In [58]:
```
n1 = plt.hist(movies.CriticRating, bins=20) #uniform distribution
```

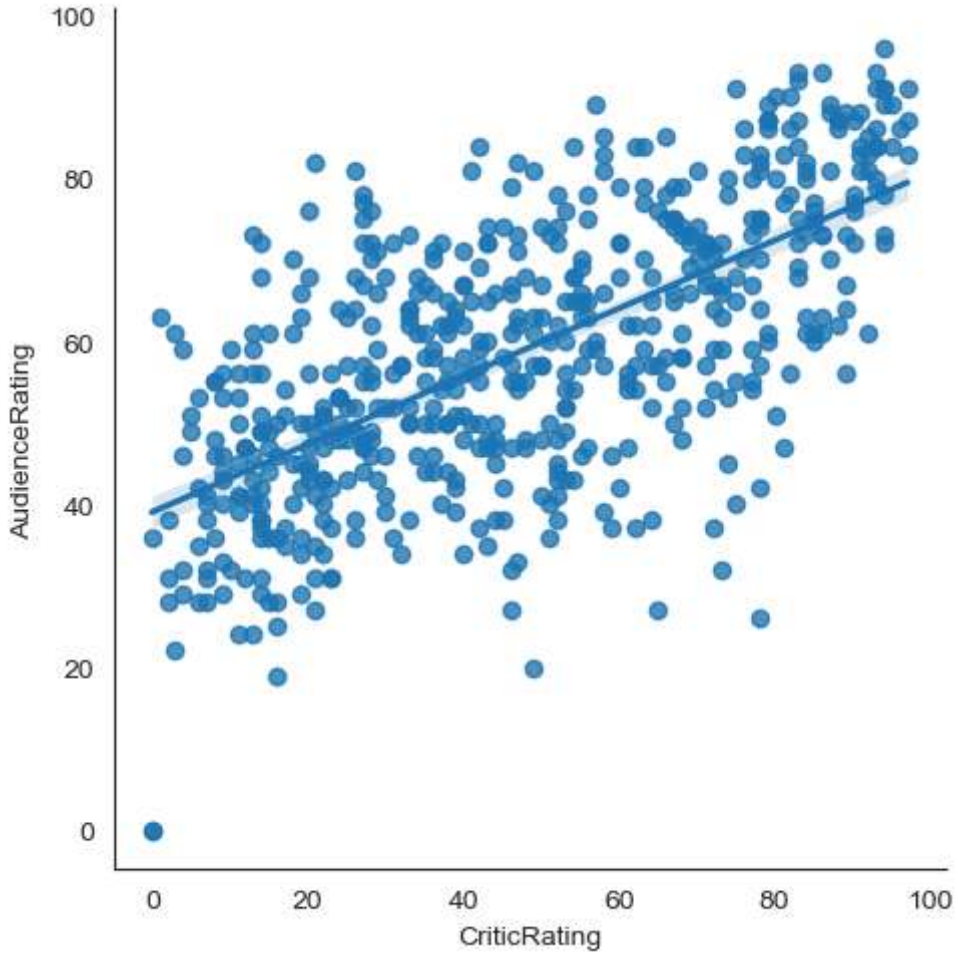


In [59]:
```
# <<< chat - 2

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

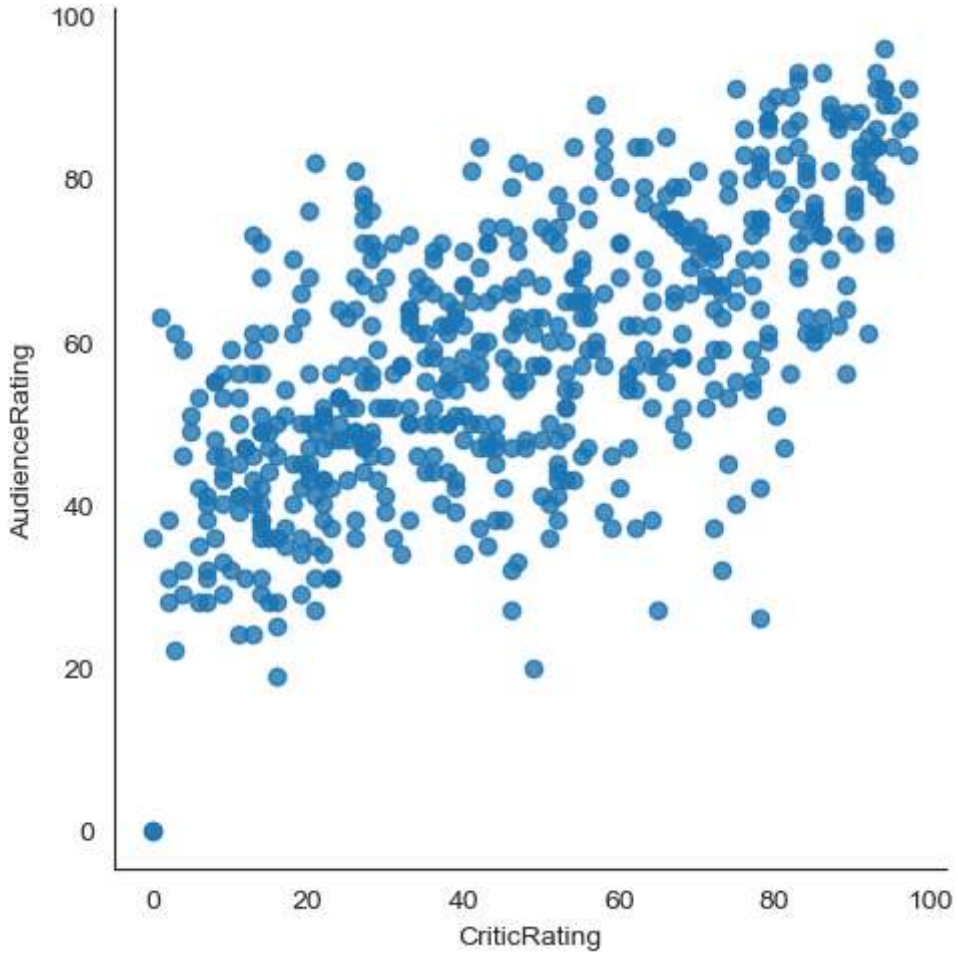
In [37]:
```
#h1 = plt.hist(movies.BudgetMillions)
plt.hist(movies.BudgetMillions)
plt.show()
```



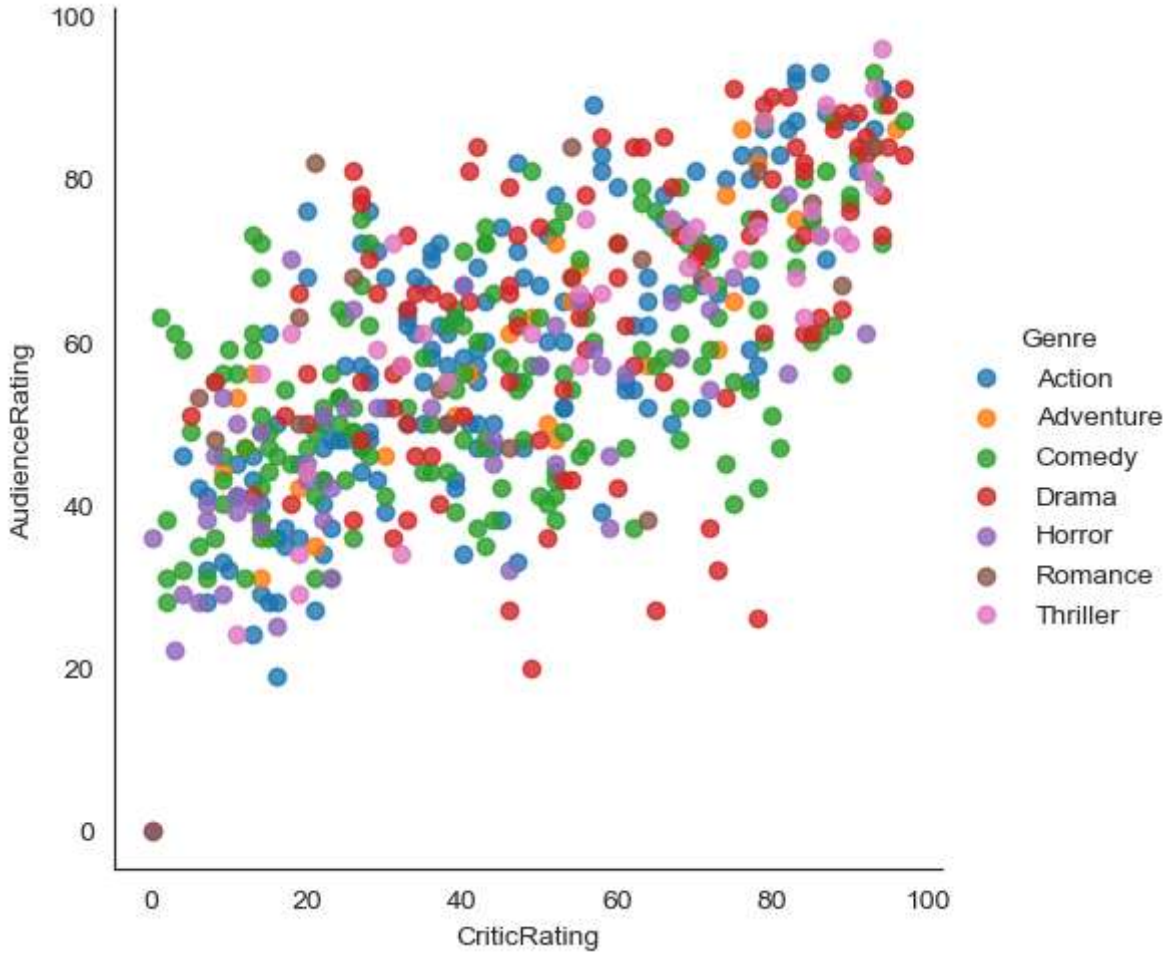
In [38]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating', fit_reg=True)
```



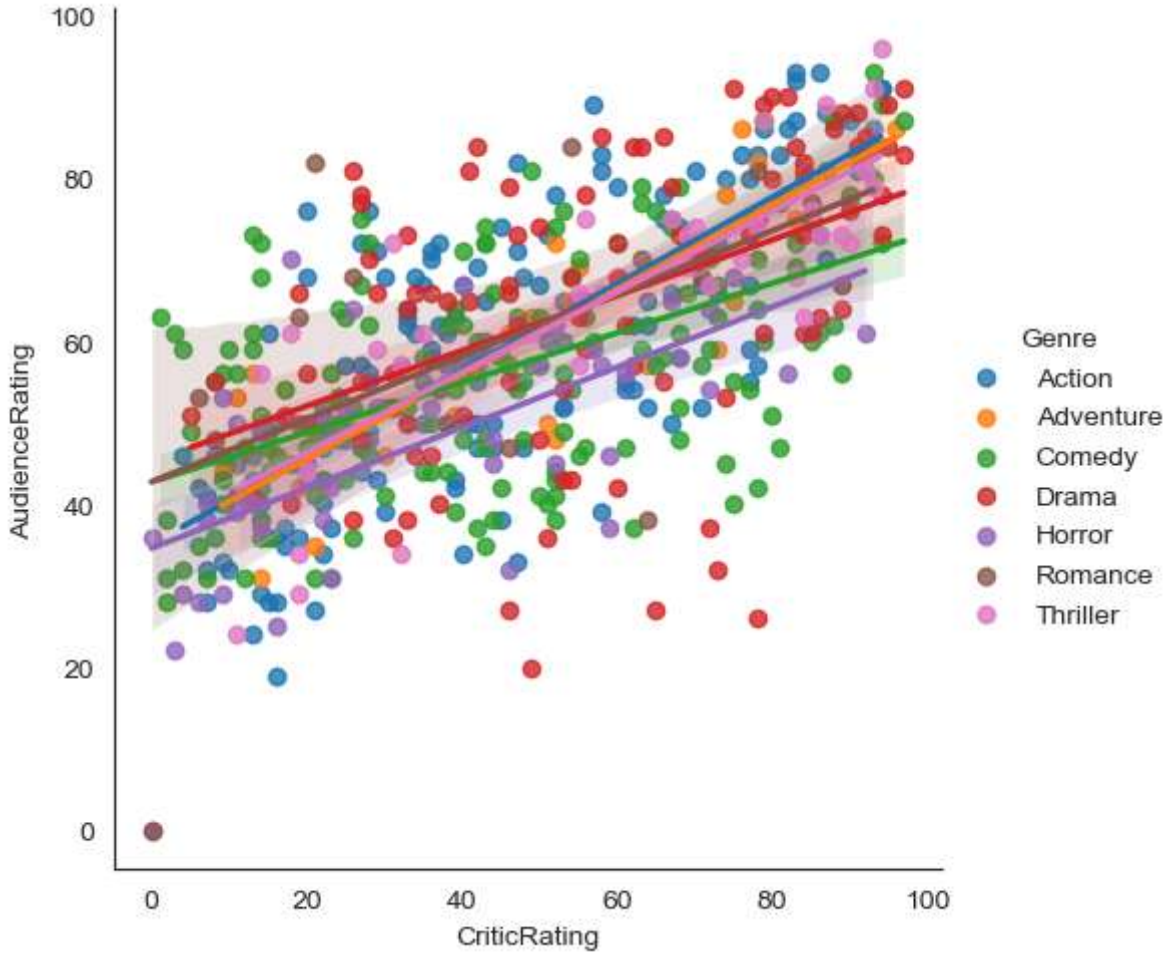
In [39]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating', fit_reg=False)
```



In [40]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating', \
                  fit_reg=False, hue = 'Genre')
```



In [41]:
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRating', \
                  fit_reg=True, hue = 'Genre')
```



In [ ]:
```
# Kernal Density Estimate plot (KDE PLOT)
# how can i visualize audiene rating & critics rating - using scatterplot
```

In [ ]:
```
#k1 = sns.kdeplot(data = movies,movies.CriticRating,movies.AudienceRating)
# where do u find more density and how density is distributed across from the chat
# center point is kernal this is called KDE & insteade of dots it visualize like this
# we can able to clearly see the spread at the audience ratings
```

In [ ]:
```
#k1 = sns.kdeplot(movies.CriticRating,movies.AudienceRating,shade = True,shade_Lowest=False,cmap='Reds')
```

In [ ]:
```
#k1 = sns.kdeplot(movies.CriticRating,movies.AudienceRating,shade = False,shade_Lowest=True,cmap='Reds')
```

In [ ]:
```
#k1 = sns.kdeplot(movies.CriticRating,movies.AudienceRating,shade_Lowest=True,cmap='Greens')
```

In [60]:
```
movies.head()
```

Out[60]:

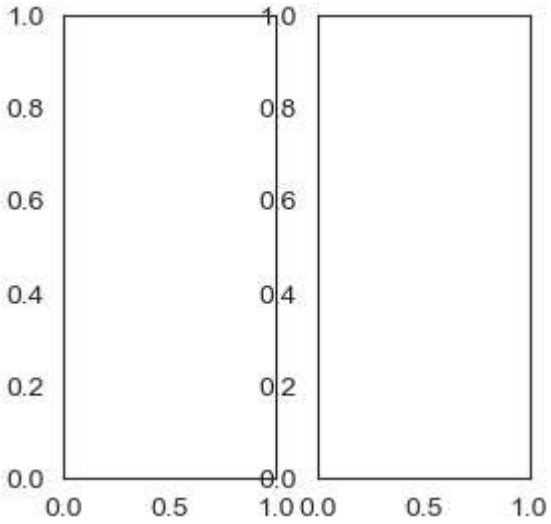
	Film	Genre	CriticRating	AudienceRating	BudgetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009

In [ ]:
```
#sns.set_style('dark')
#k1 = sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,shade_Lowest=True,cmap='Green_r')
```

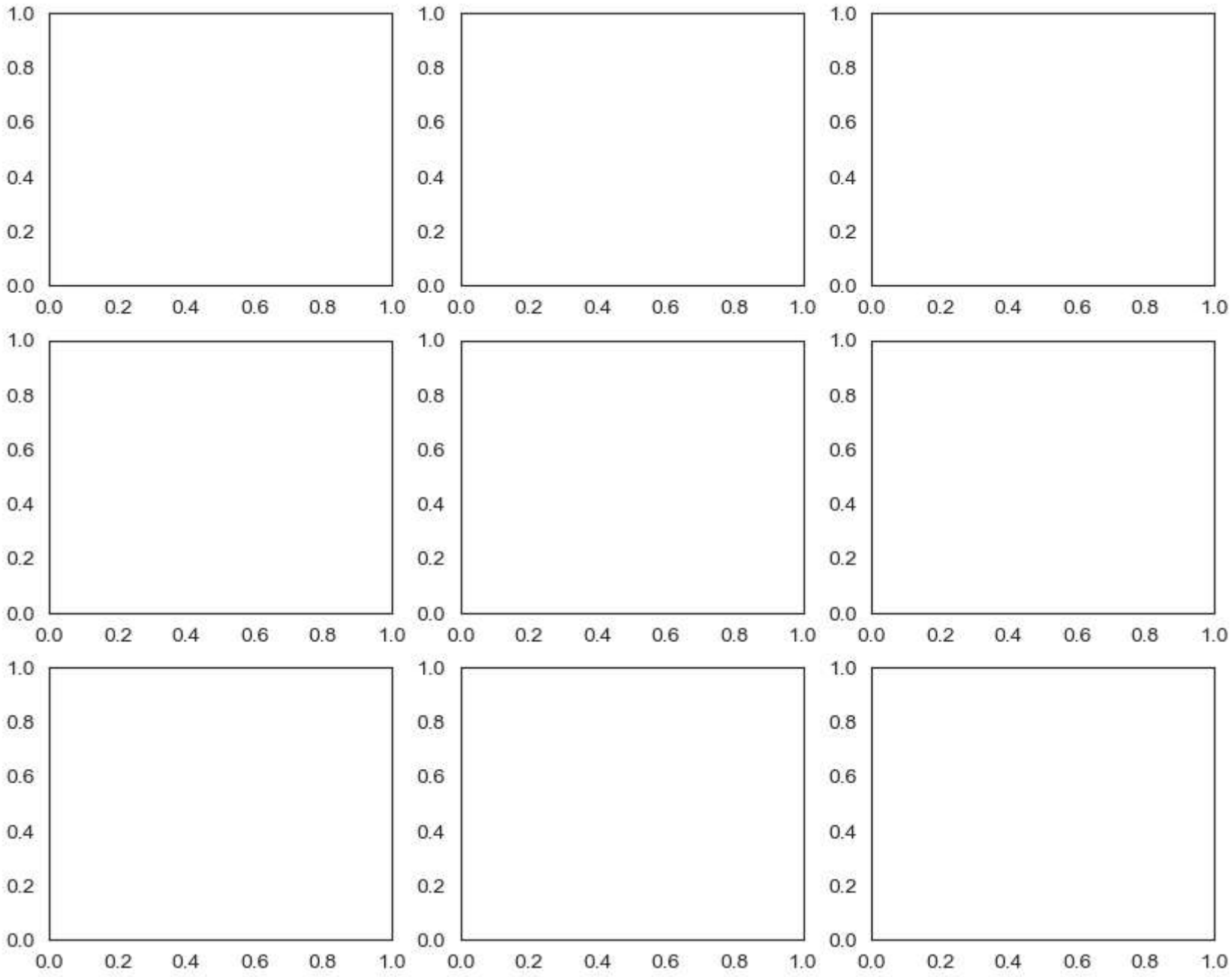
In [ ]:
```
#sns.set_style('dark')
#k1 = sns.kdeplot(movies.BudgeMillions,movies.AudienceRating)
```

In [ ]:
```
#k2 = sns.kdeplot(movies.BudgeMillions,movies.CriticRating)
```

In [65]:
```
#subplots
ax = plt.subplots(1,2, figsize = (3,3))
#ax = plt.subplots(3,3, figsize =(10,8))
```



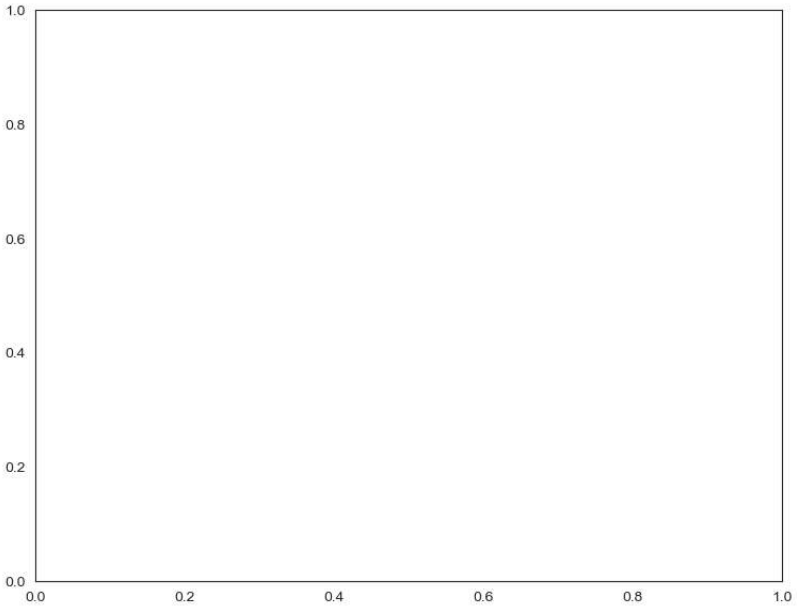
In [66]:
```
#subplots
#ax = plt.subplots(1,2, figsize = (3,3))
ax = plt.subplots(3,3, figsize =(10,8))
```

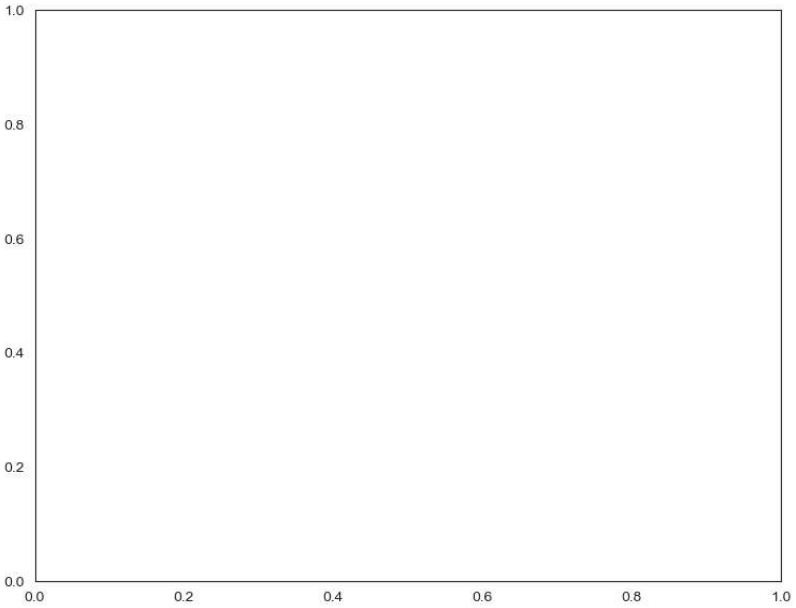


In [64]:
```
f, axes = plt.subplots(1,2, figsize =(20,7))
```

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

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[64], line 3
      1 f, axes = plt.subplots(1,2, figsize =(20,7))
----> 3 k1 = sns.kdeplot(movies.BudgetMillions,movies.AudienceRating,ax=axes[0])
      4 k2 = sns.kdeplot(movies.BudgetMillions,movies.CriticRating,ax = axes[1])

TypeError: kdeplot() takes from 0 to 1 positional arguments but 2 positional arguments (and 1 keyword-only argument)
were given
```





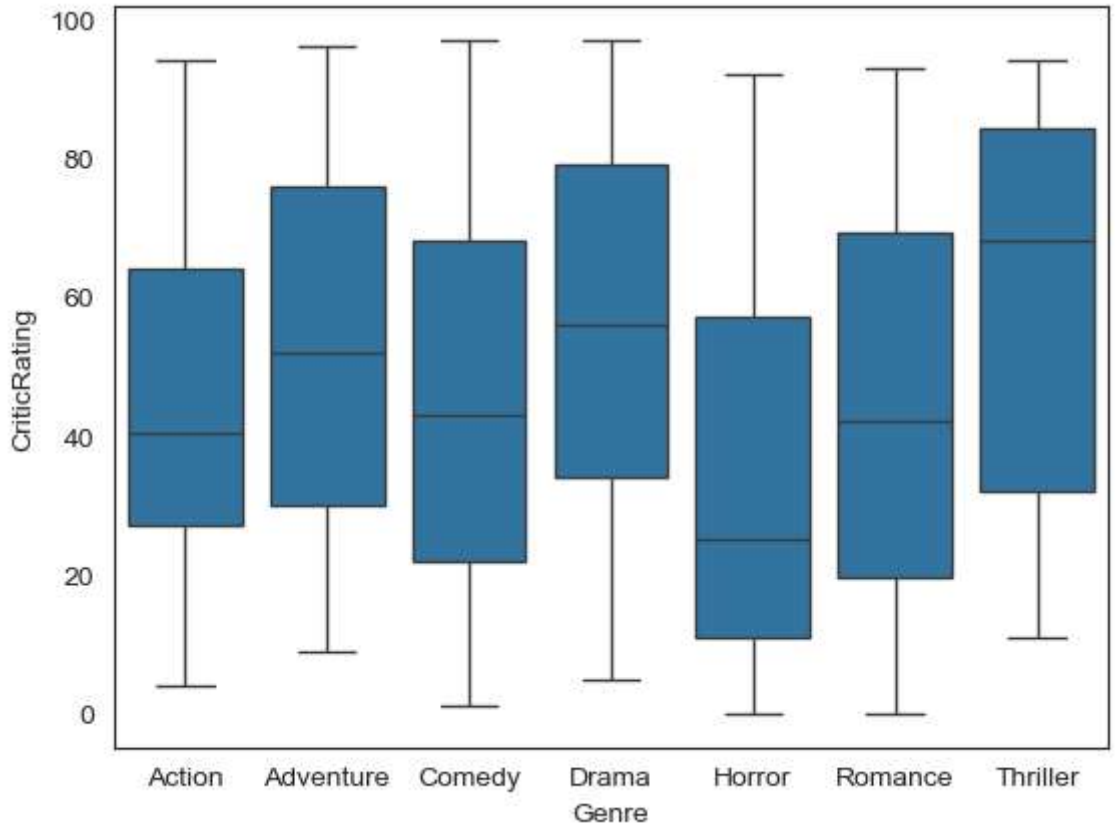
In [67]:
```
movies
```

Out[67]:

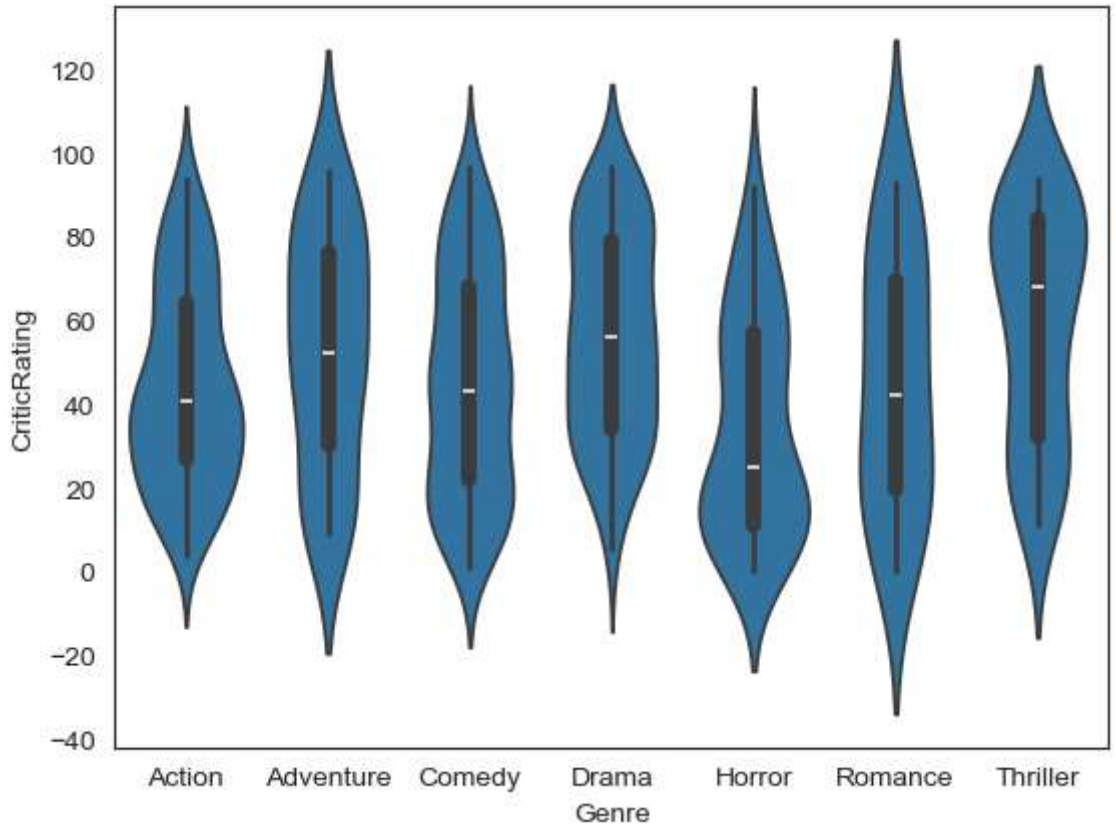
	Film	Genre	CriticRating	AudienceRating	BudgetMillions	Year
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009
...	...	...	...	...	...	...
554	Your Highness	Comedy	26	36	50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

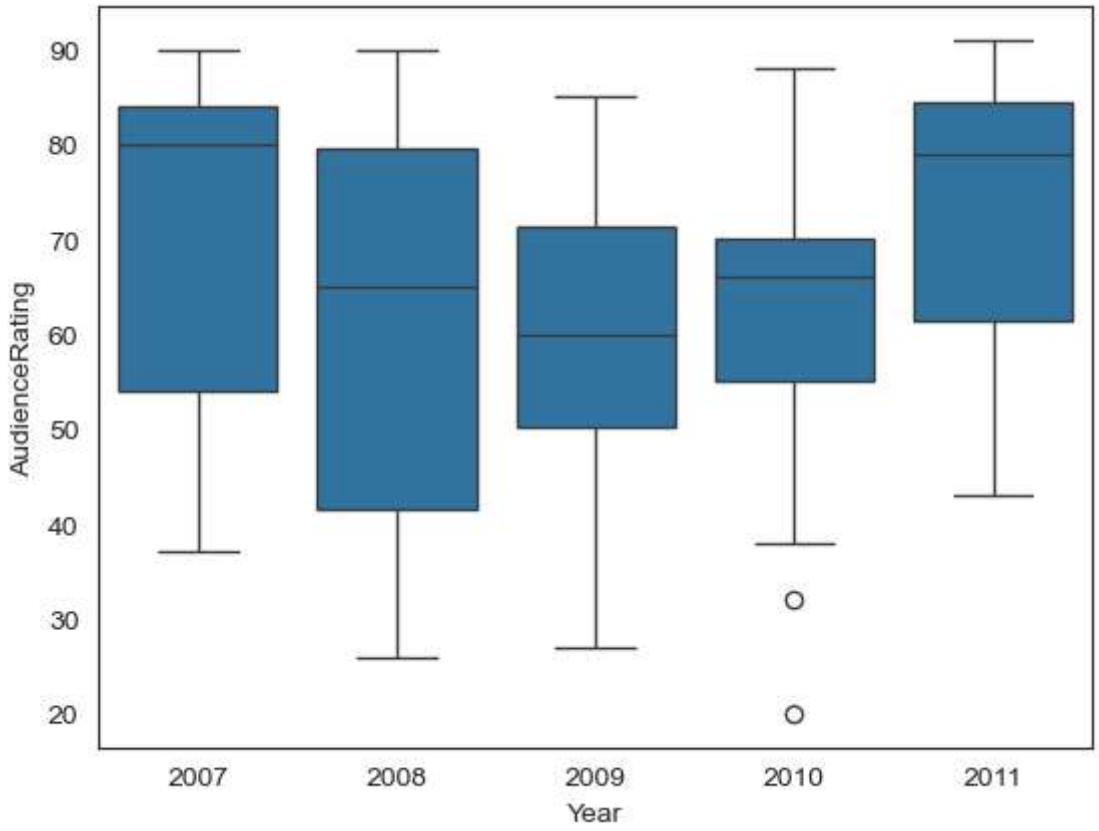
In [68]:
```
# box plots
w = sns.boxplot(data=movies, x='Genre', y = 'CriticRating')
```



In [46]:
```
#violin plot
z = sns.violinplot(data=movies, x='Genre', y = 'CriticRating')
```



In [47]:
```
w1 = sns.boxplot(data=movies[movies.Genre == 'Drama'], x='Year', y = 'AudienceRating')
```



In [48]:
```
movies.head()
```

Out[48]:

	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

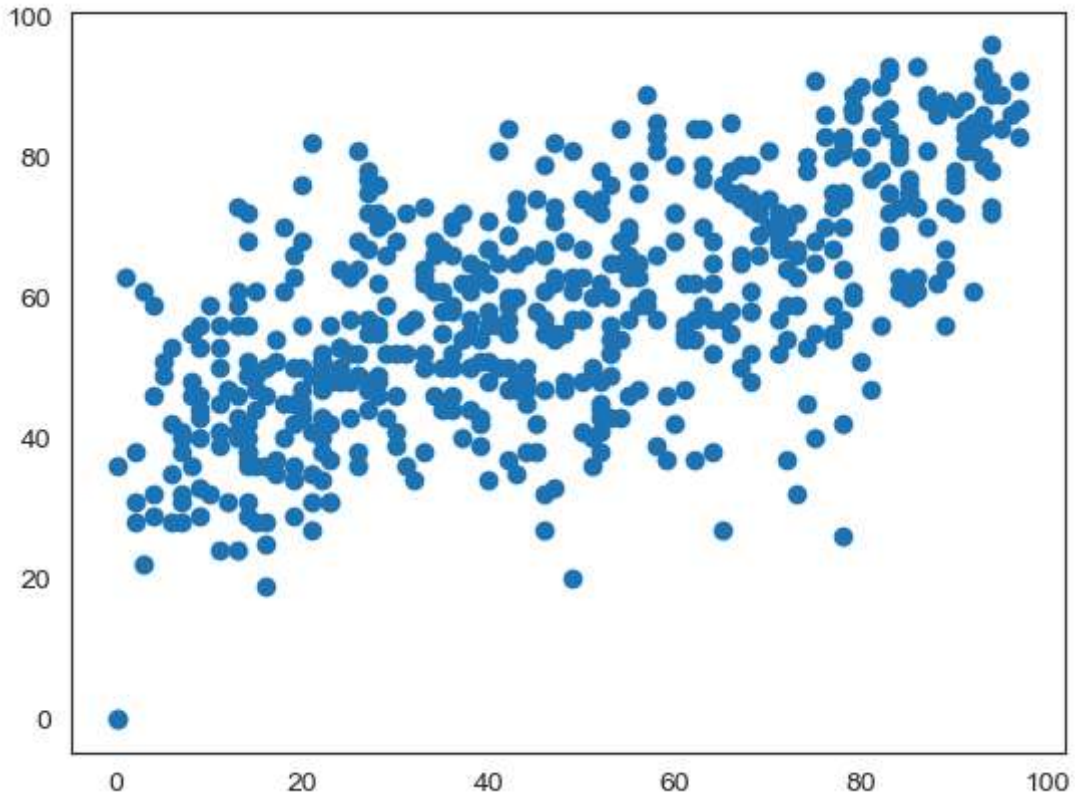
In [49]:
```
# Creating a Facet grid
```

In [50]:
```
g = sns.FacetGrid(movies, row = 'Genre', col = 'Year', hue = 'Genre') #kind of subplots
```

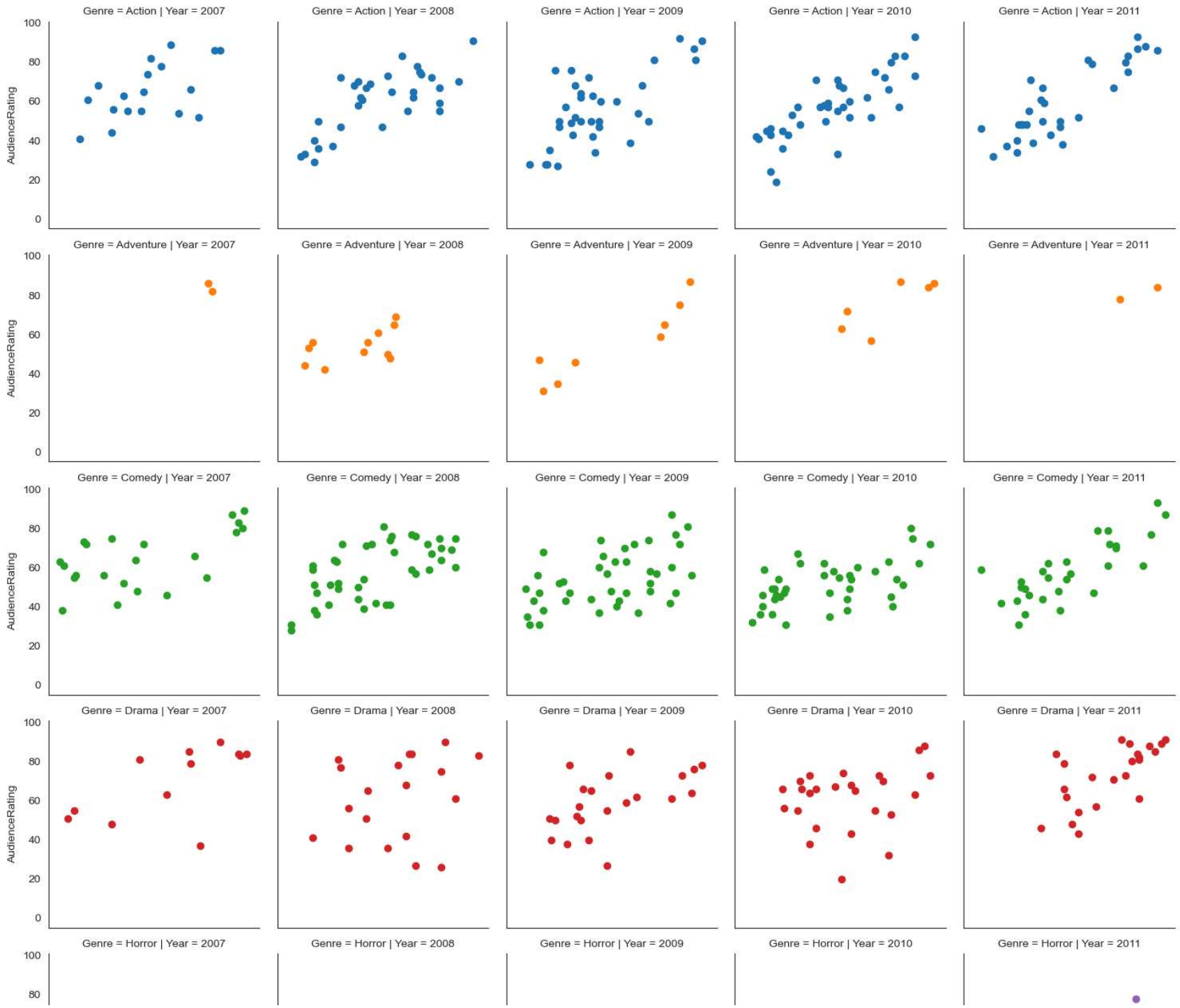

Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
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

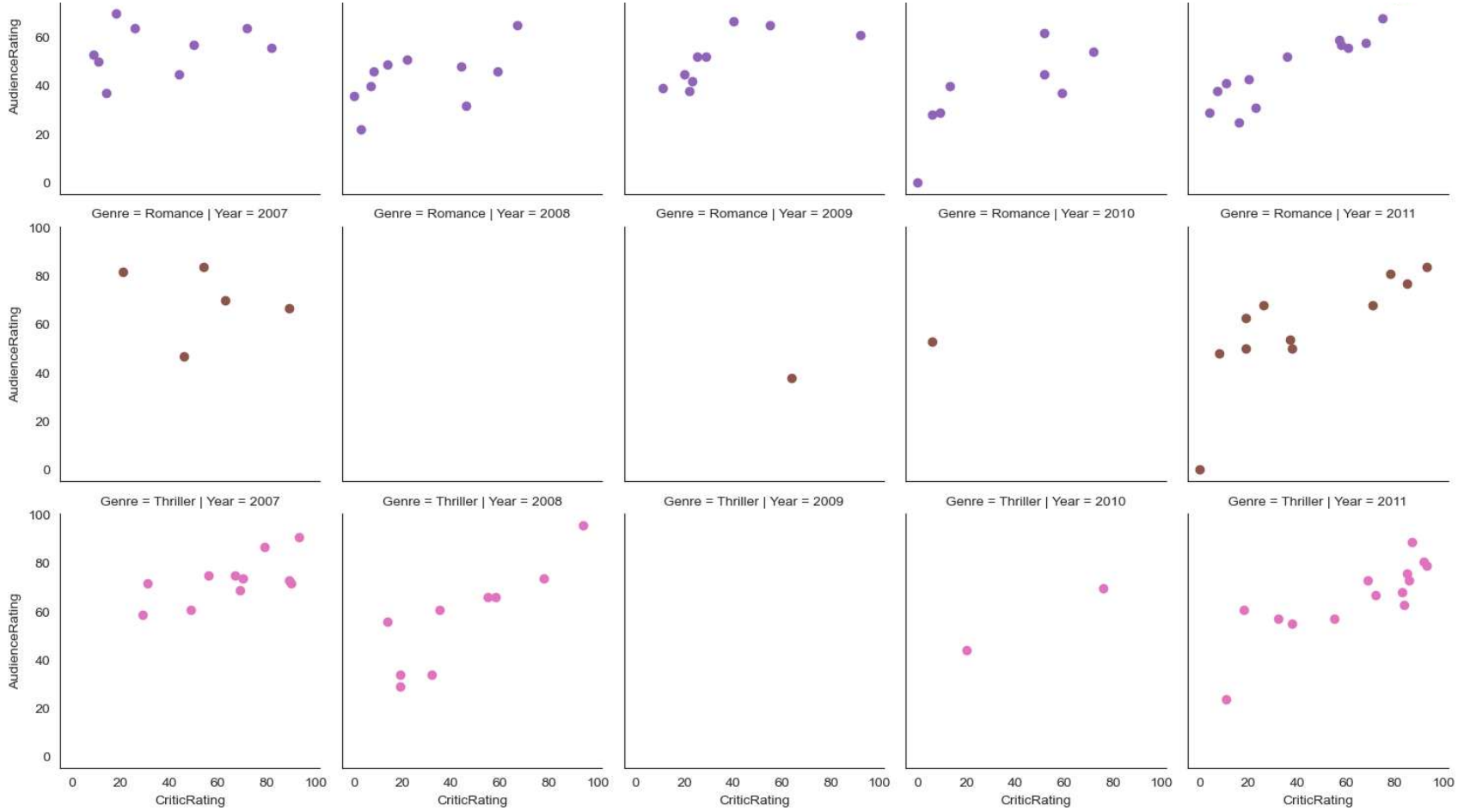
In [69]:
```
plt.scatter(movies.CriticRating,movies.AudienceRating)
```

Out[69]:
```
<matplotlib.collections.PathCollection at 0x1e3d4cbef90>
```



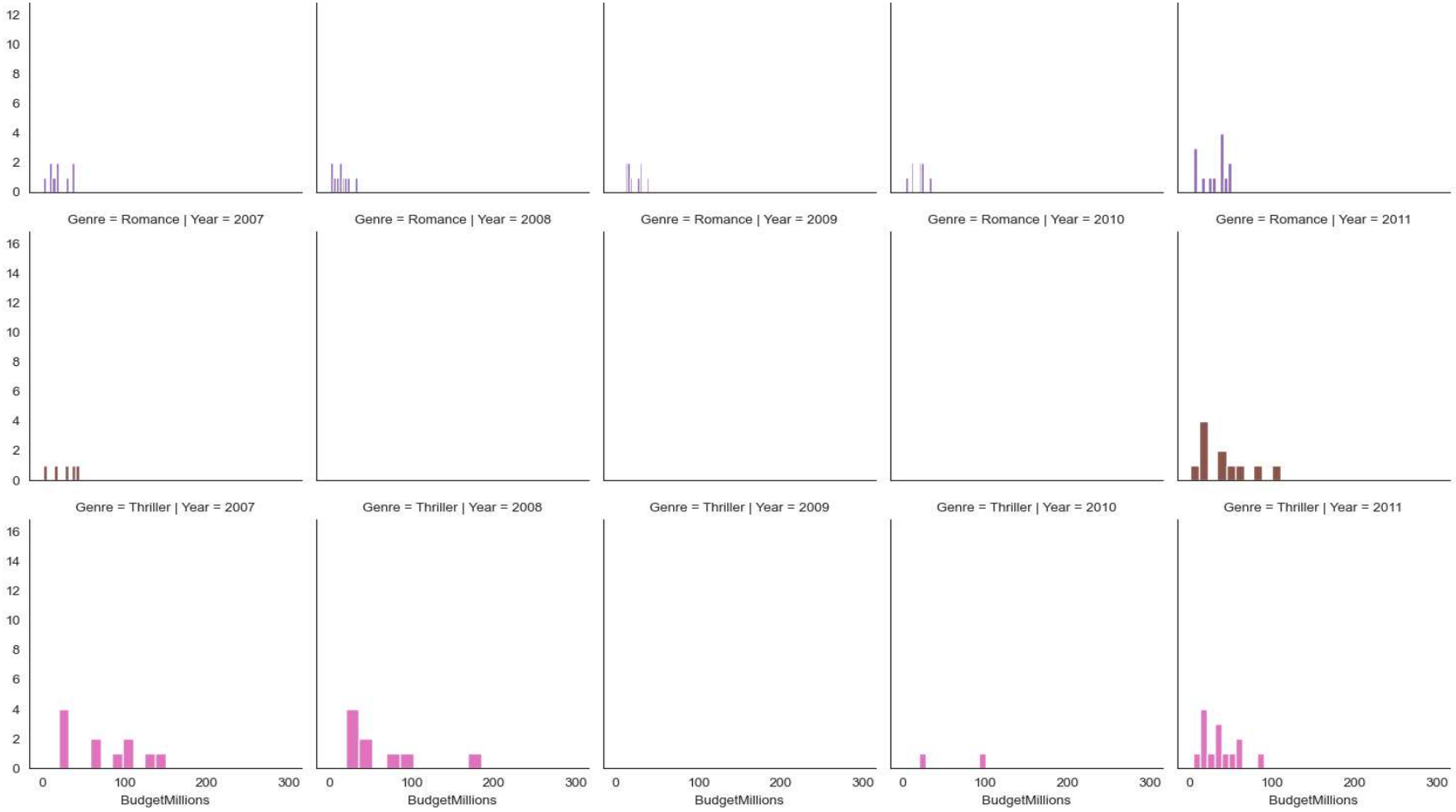
In [51]:
```
g = sns.FacetGrid(movies, row = 'Genre', col = 'Year', hue = 'Genre')
g = g.map(plt.scatter, 'CriticRating', 'AudienceRating') # scatterplots are mapped in facetgrid
```


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 [53]:
```
# you can populated any type of chat.
g = sns.FacetGrid(movies, row = 'Genre', col = 'Year', hue = 'Genre')
g = g.map(plt.hist, 'BudgetMillions') #histogram 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


Genre = Romance | Year = 2007
Genre = Romance | Year = 2008
Genre = Romance | Year = 2009
Genre = Romance | Year = 2010
Genre = Romance | Year = 2011
Genre = Thriller | Year = 2007
Genre = Thriller | Year = 2008
Genre = Thriller | Year = 2009
Genre = Thriller | Year = 2010
Genre = Thriller | Year = 2011
BudgetMillions

In [ ]: