Seaborn Project for Data visualization

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

In [2]:
```
movies = pd.read_csv(r"C:\Users\ankus\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]:
```
type(movies)
```

Out[3]:
```
pandas.core.frame.DataFrame
```

In [4]:
```
movies
```

Out[4]:

	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 [5]:
```
len(movies)
```

Out[5]:
```
559
```

In [6]:
```
import numpy
print(numpy.__version__)
```

```
2.1.3
```

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

```
2.2.3
```

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

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

In [9]:
```
movies.info()   #information of the columns
```

```
<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 [10]:
```
movies.shape
```

Out[10]:
```
(559, 6)
```

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

Out[11]:

	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 [12]:
```
movies.tail()
```

Out[12]:

	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 [13]:
```
movies.columns
```

Out[13]:
```
Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
       'Budget (million $)', 'Year of release'],
      dtype='object')
```

In [14]:
```
movies.columns=['Film', 'Genre', 'CriticRatings', 'AudienceRating', 'BudgetMilli
```

In [15]:
```
movies.head(1)          #Remove spaces and % removed noise characters
```

Out[15]:

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

In [16]:
```
movies.shape
```

Out[16]:
```
(559, 6)
```

In [17]:
```
movies.describe()    #discriptive statistics   gives always numerical data
```

Out[17]:

	CriticRatings	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 [18]:
```
movies.describe().transpose()
```

Out[18]:

	count	mean	std	min	25%	50%	75%	max
CriticRatings	559.0	47.309481	26.413091	0.0	25.0	46.0	70.0	97.0
AudienceRating	559.0	58.744186	16.826887	0.0	47.0	58.0	72.0	96.0
BudgetMillions	559.0	50.236136	48.731817	0.0	20.0	35.0	65.0	300.0
Year	559.0	2009.152057	1.362632	2007.0	2008.0	2009.0	2010.0	2011.0

In [19]:
```
# I don't want year
```

In [20]:
```
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   CriticRatings   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 [21]:
```
movies.Film=movies.Film.astype('category')    # changing data to category
```

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

Out[22]:

	CriticRatings	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 [23]:
```
movies.Genre = movies.Genre.astype('category')
movies.Year = movies.Year.astype('category')   # object to category
```

In [24]:
```
movies.describe()
```

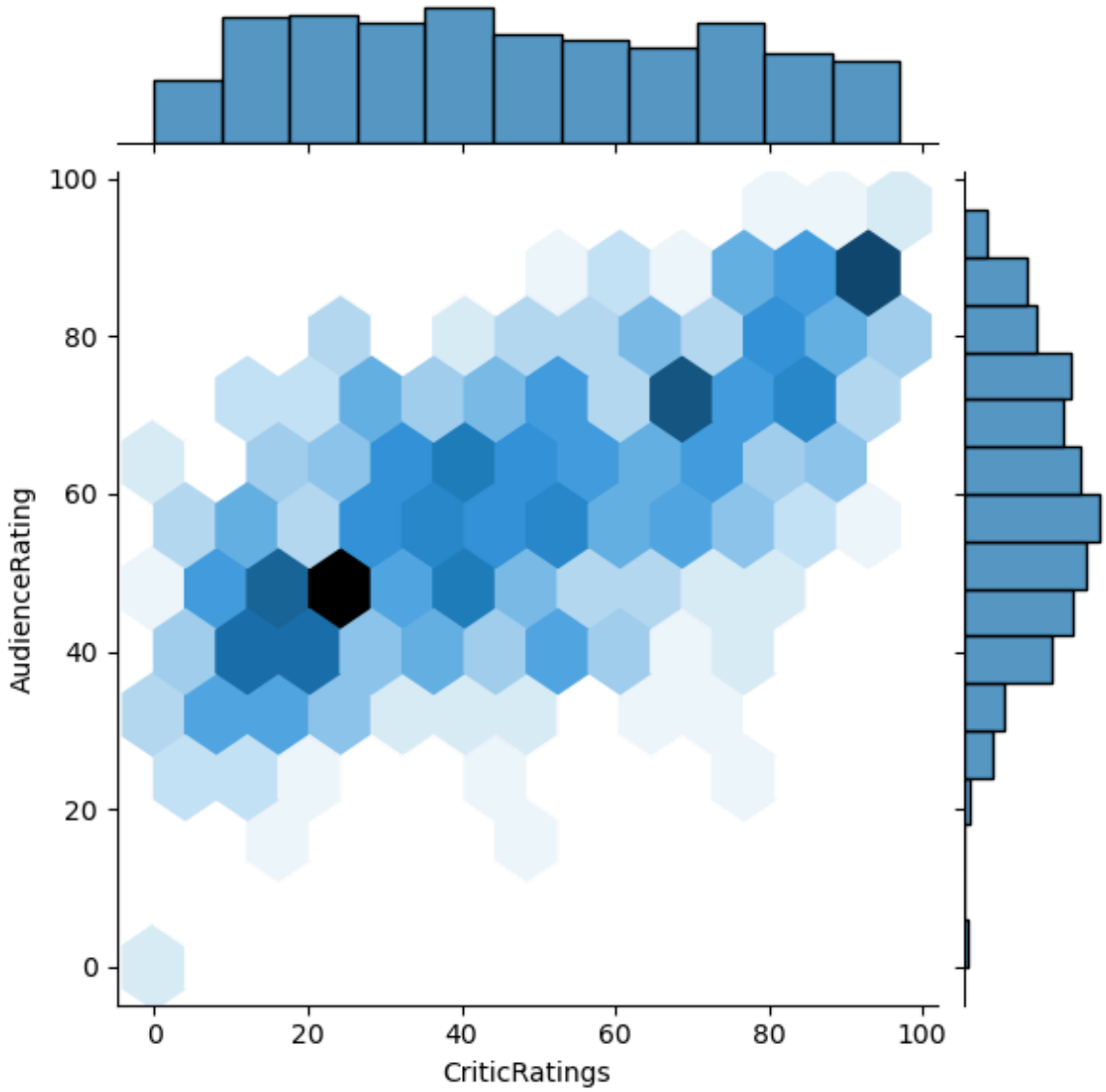
Out[24]:

	CriticRatings	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

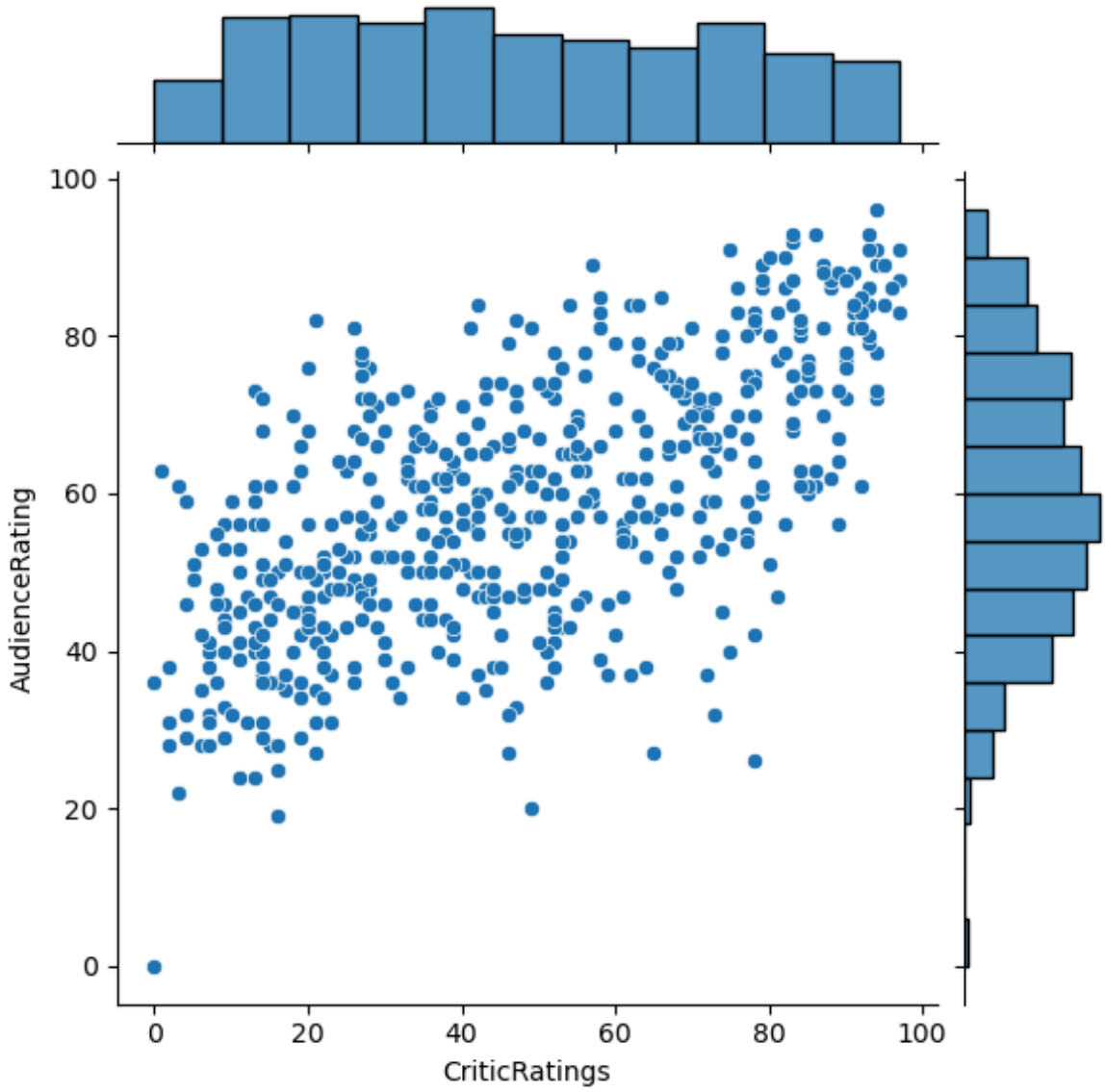
In [25]:
```
from matplotlib import pyplot as plt
import seaborn as sns
```

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

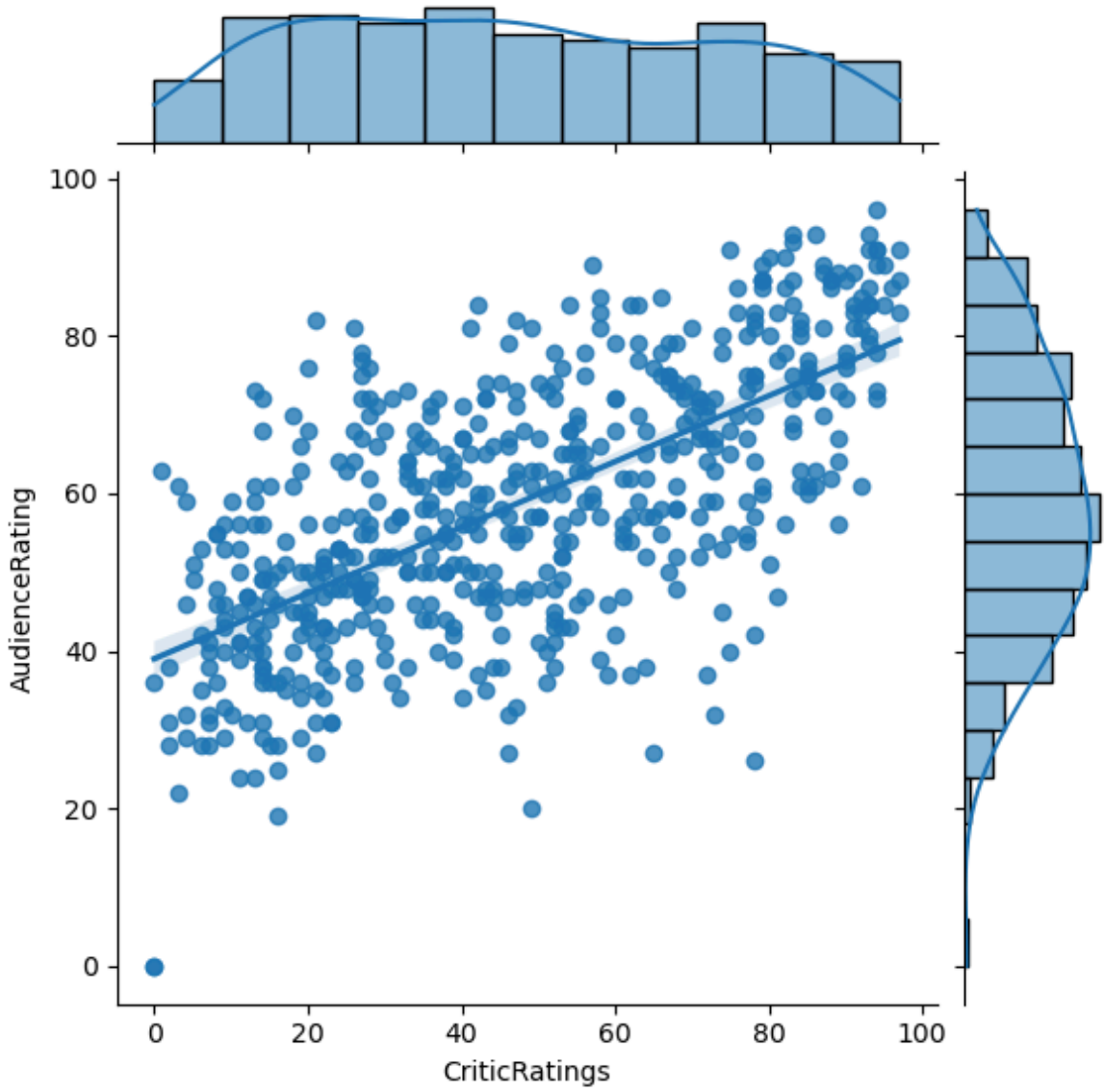
In [26]:
```
j =sns.jointplot(data=movies,x='CriticRatings',y='AudienceRating',kind='hex')  #
```



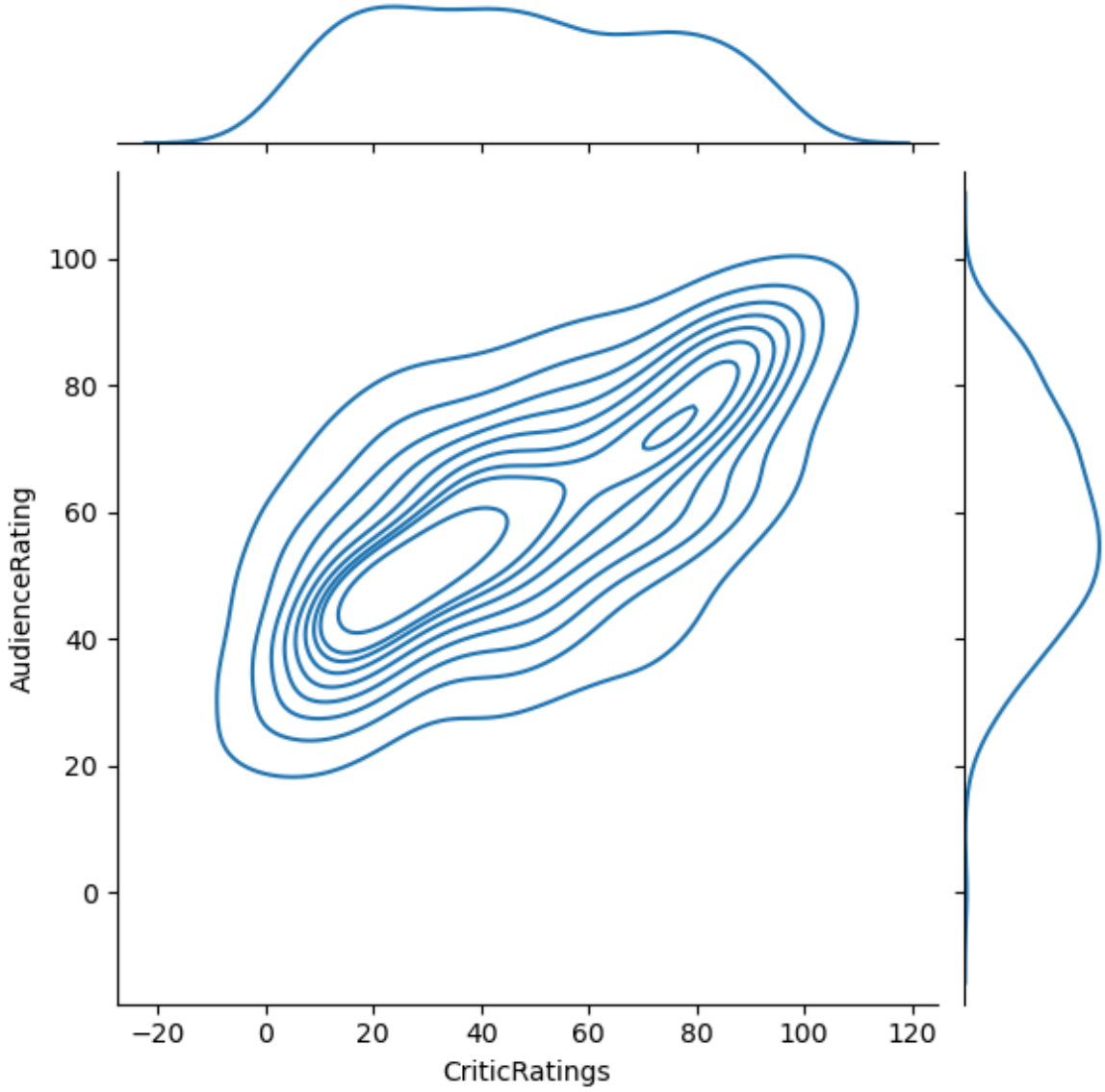
In [27]:
```
j =sns.jointplot(data=movies,x='CriticRatings',y='AudienceRating',kind='scatter'
```



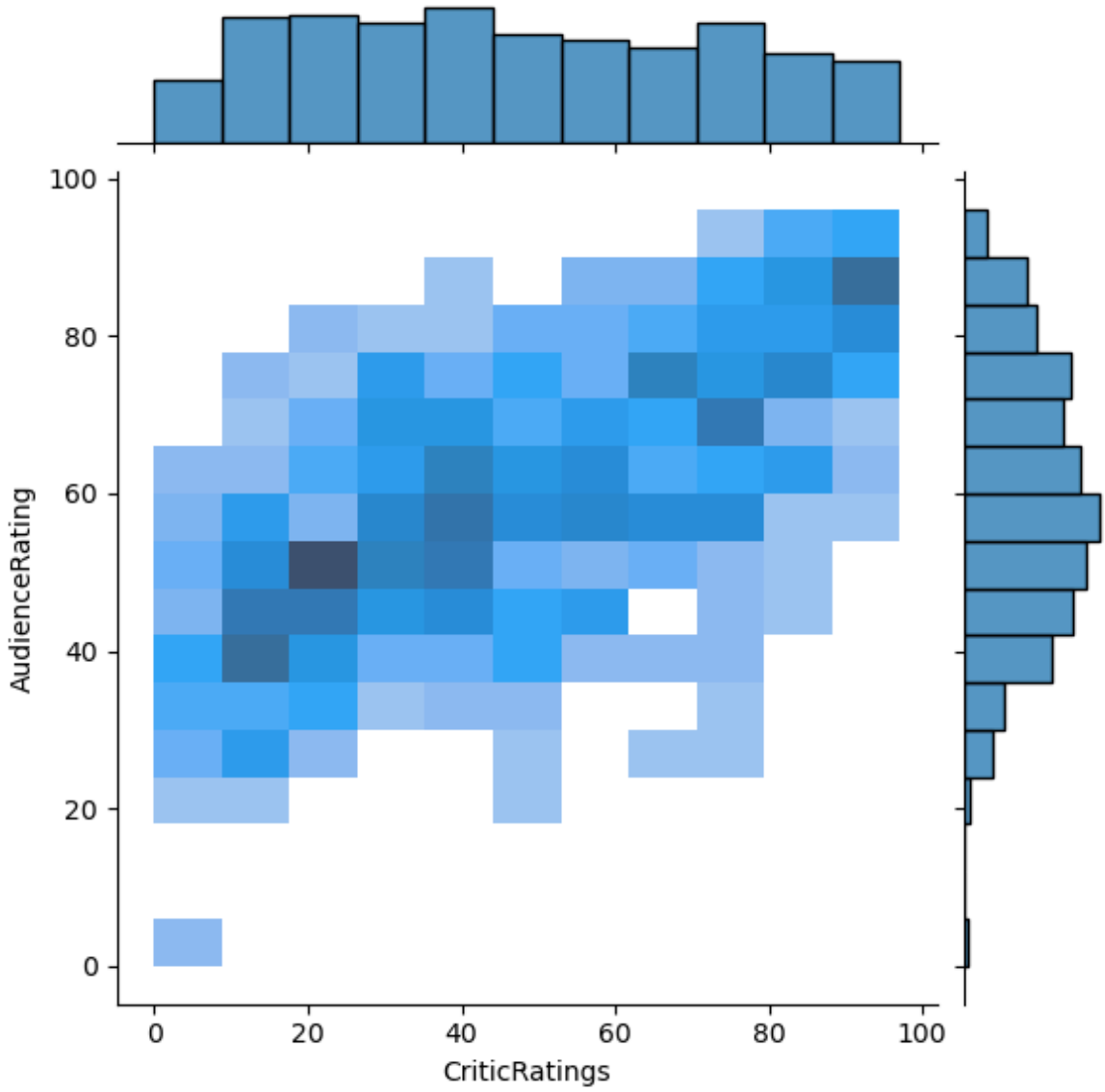
In [28]:
```
j =sns.jointplot(data=movies,x='CriticRatings',y='AudienceRating',kind='reg')
```



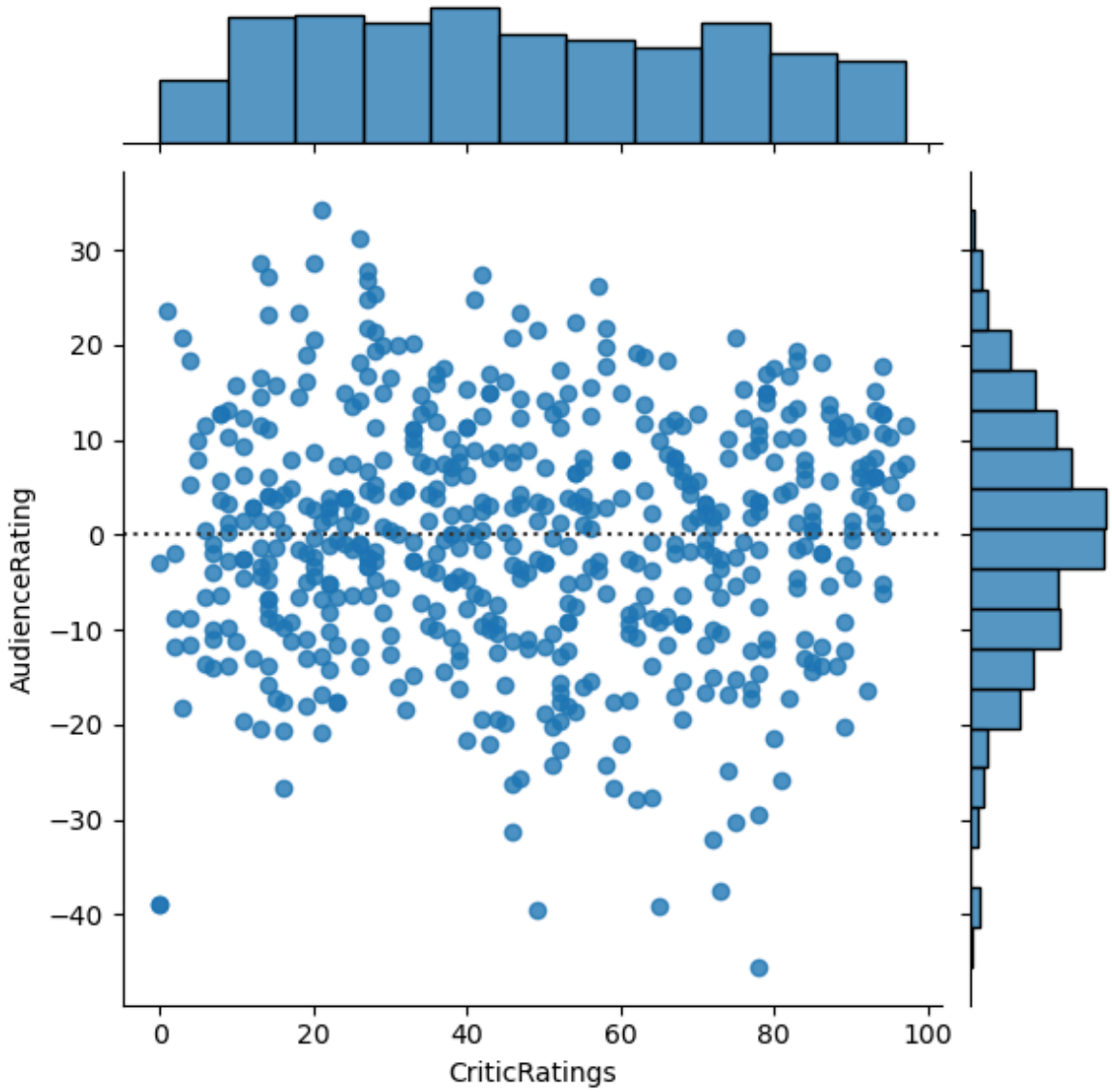
In [29]:
```
j =sns.jointplot(data=movies,x='CriticRatings',y='AudienceRating',kind='kde')
```



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

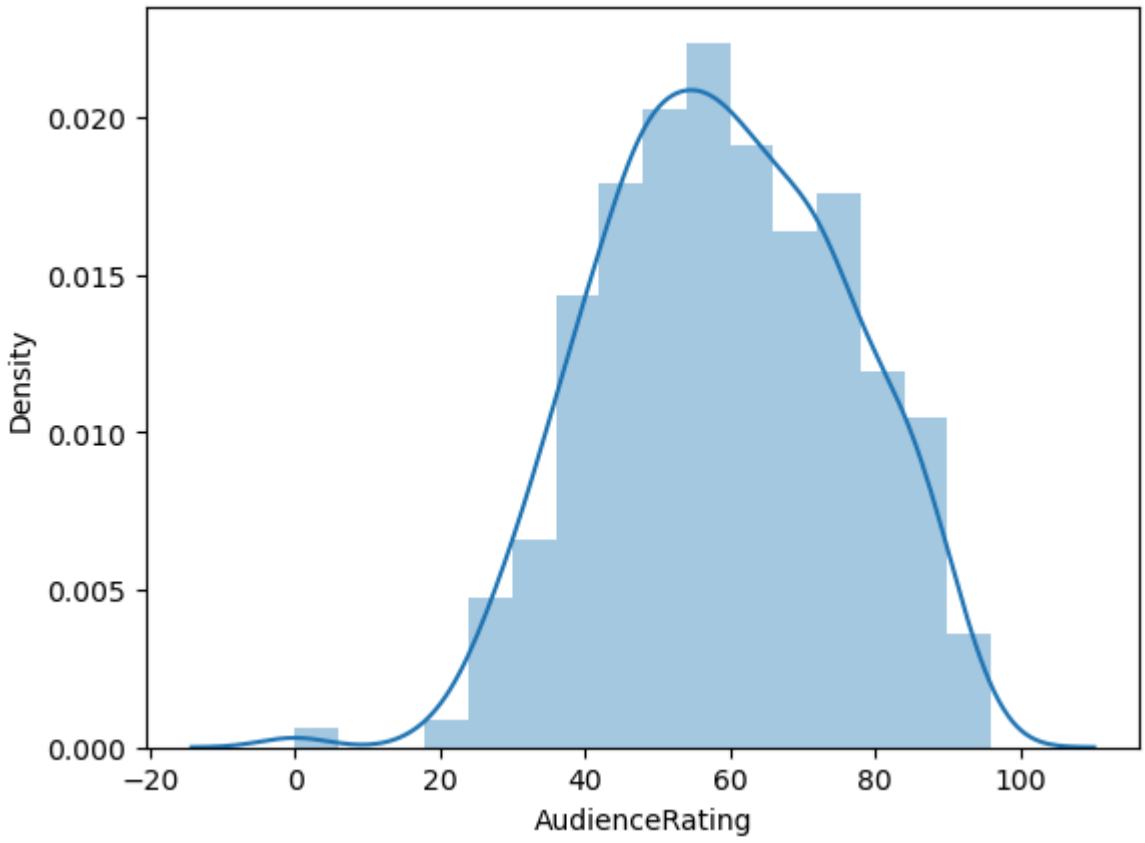


In [31]:
```
j =sns.jointplot(data=movies,x='CriticRatings',y='AudienceRating',kind='resid')
```



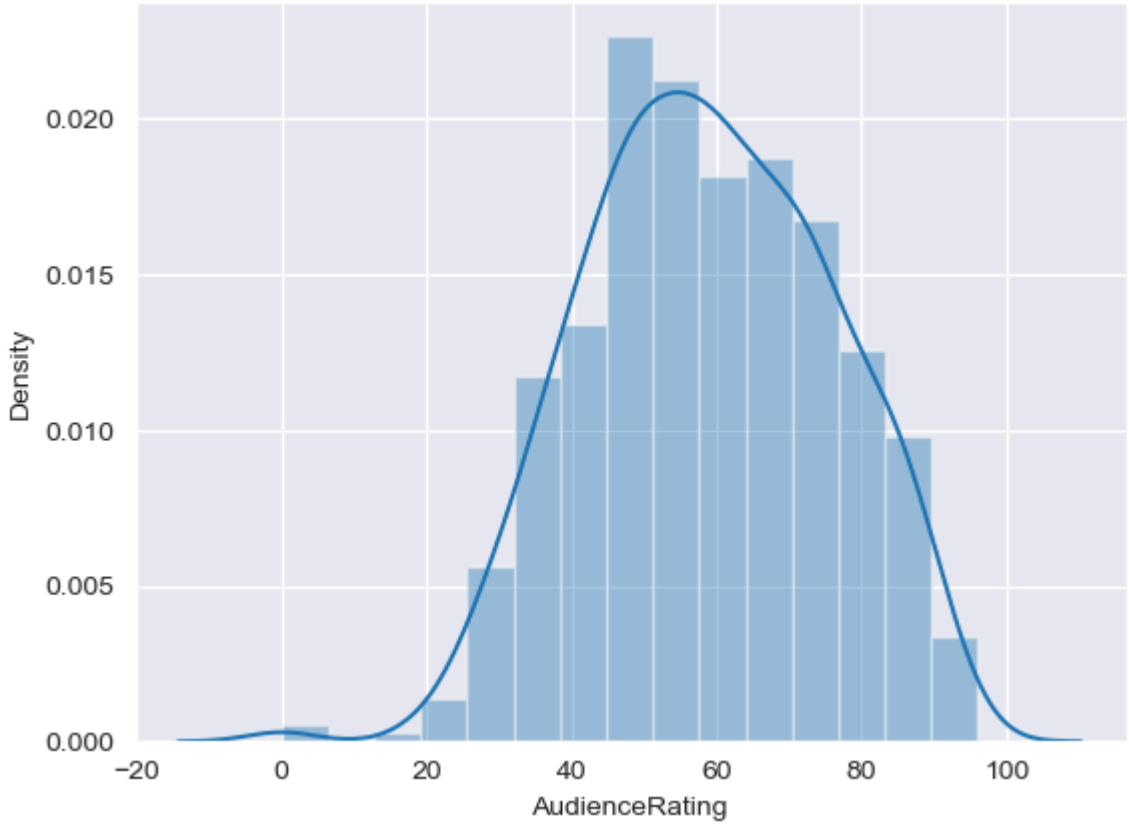
In [32]:
```
#Historams
```

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

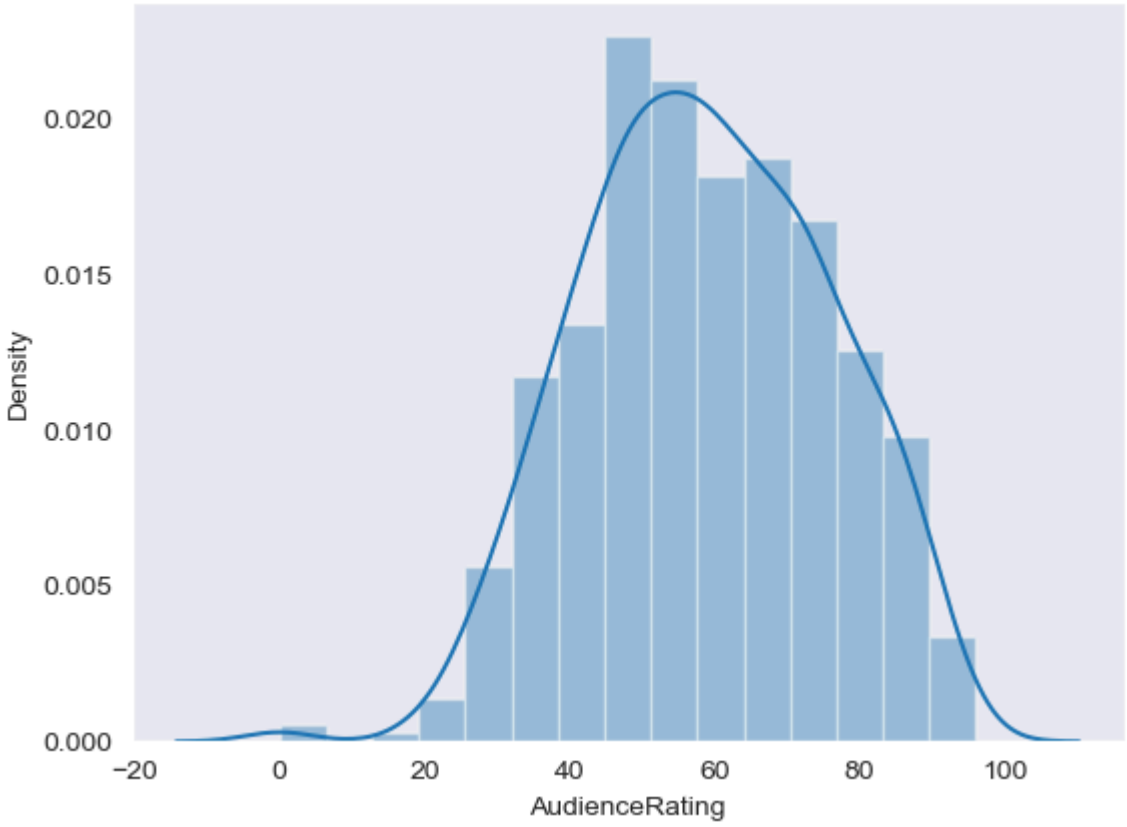


In [34]:
```
sns.set_style('darkgrid')
```

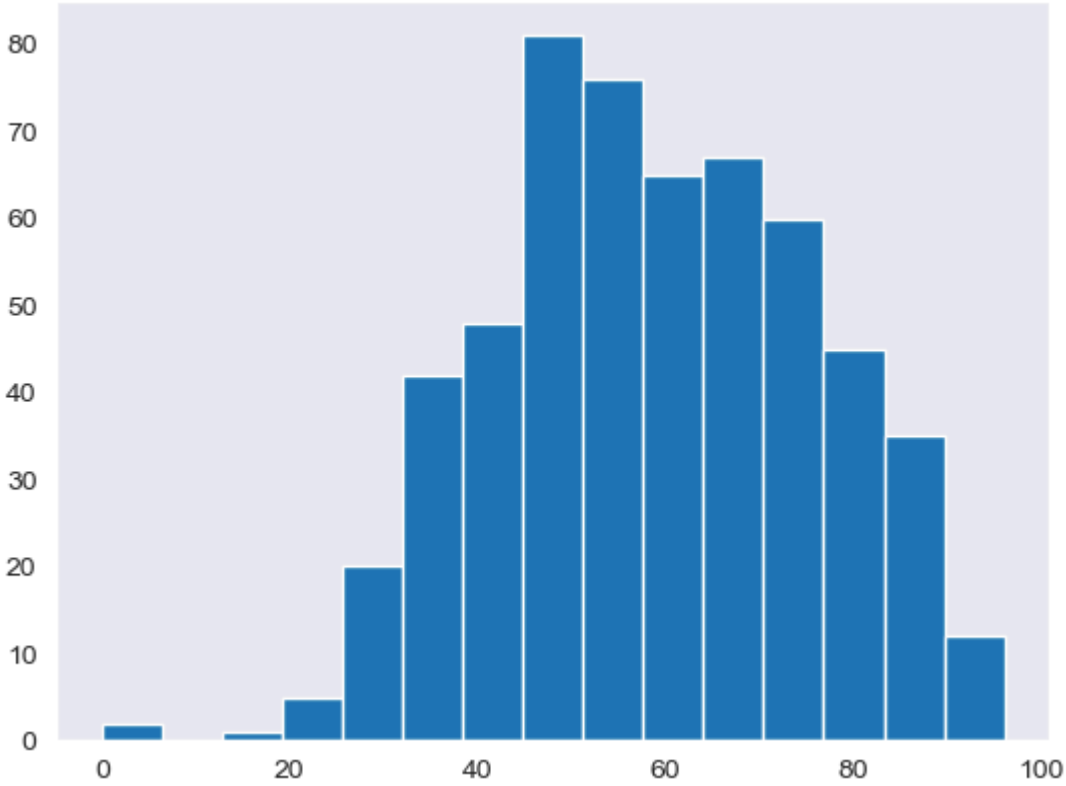
In [35]:
```
k=sns.distplot(movies.AudienceRating,bins=15)
```



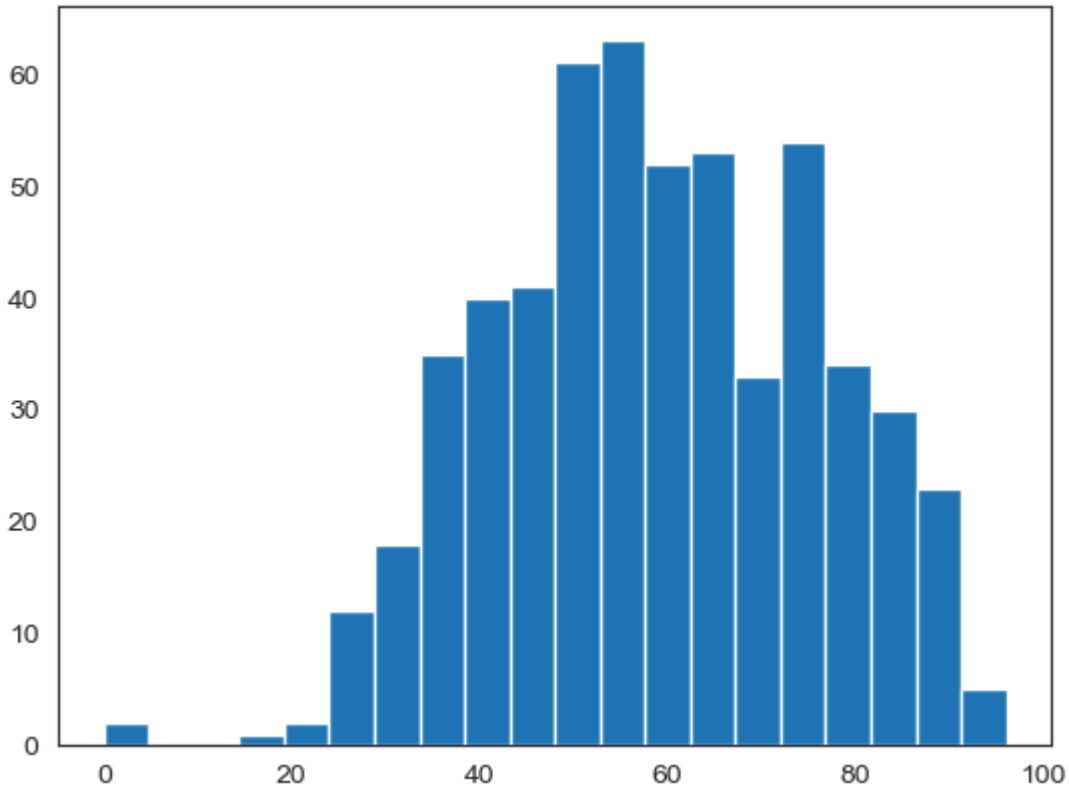
In [36]:
```
sns.set_style('dark')
k=sns.distplot(movies.AudienceRating,bins=15)
```



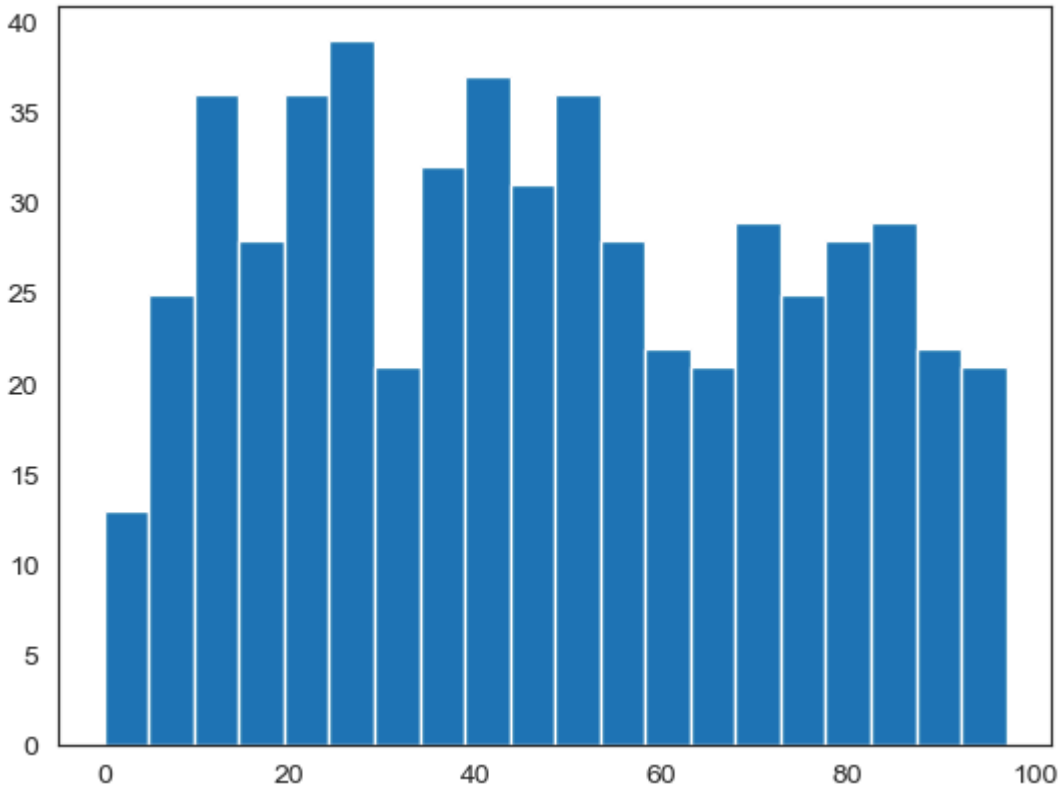
In [37]:
```
m = plt.hist(movies.AudienceRating,bins=15)
```



In [38]:
```
sns.set_style('white')
m=plt.hist(movies.AudienceRating,bins=20)
```

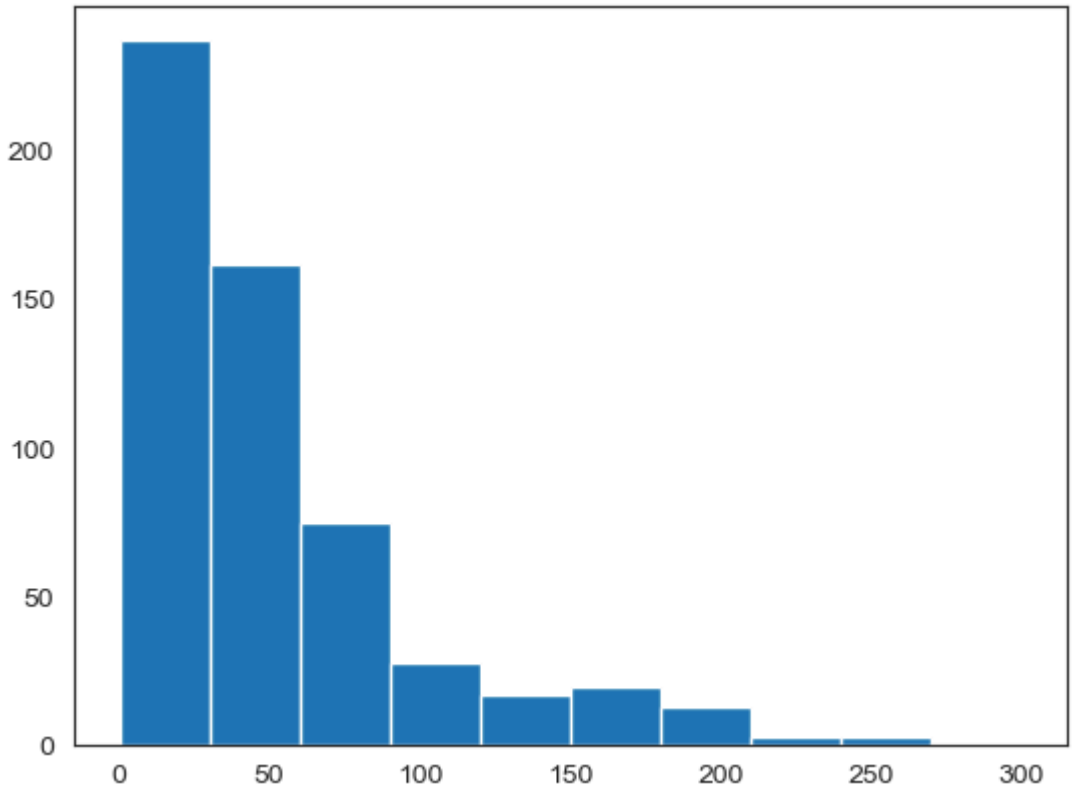


In [39]:
```
m= plt.hist(movies.CriticRatings, bins=20)   #uniform distribution
```



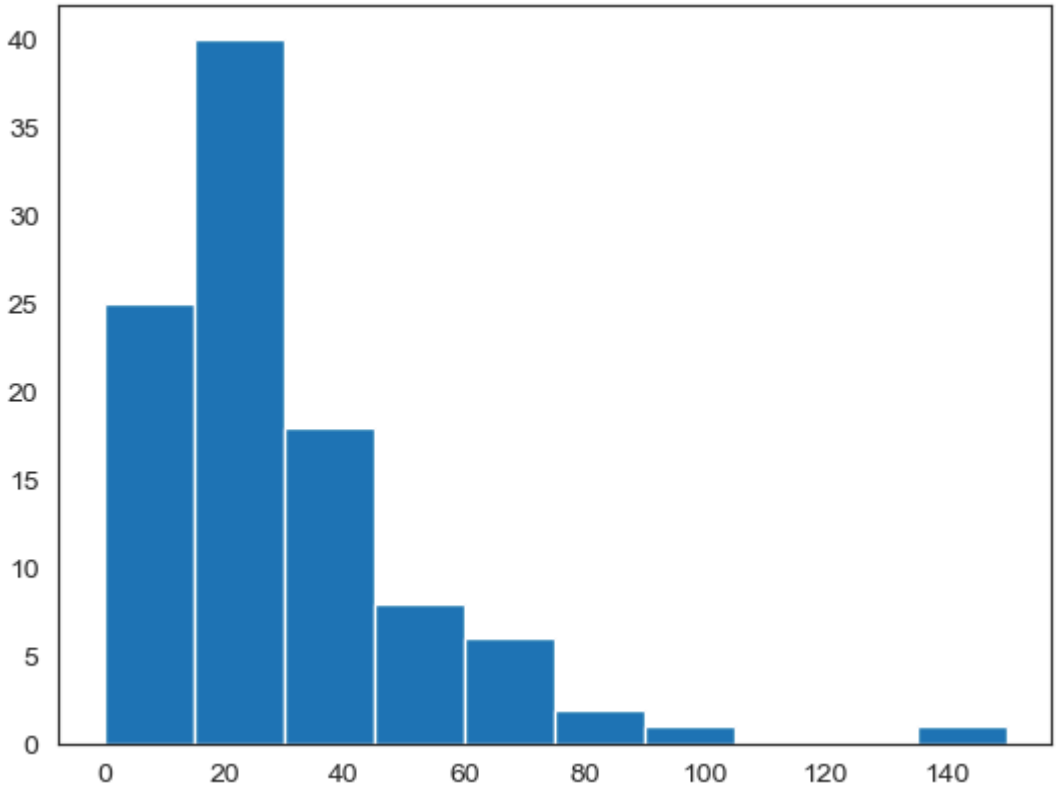
In [40]:
```
plt.hist(movies.BudgetMillions)
```

Out[40]:
```
(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 [41]:
```
plt.hist(movies[movies.Genre == 'Drama'].BudgetMillions)
```

Out[41]:
```
(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>)
```

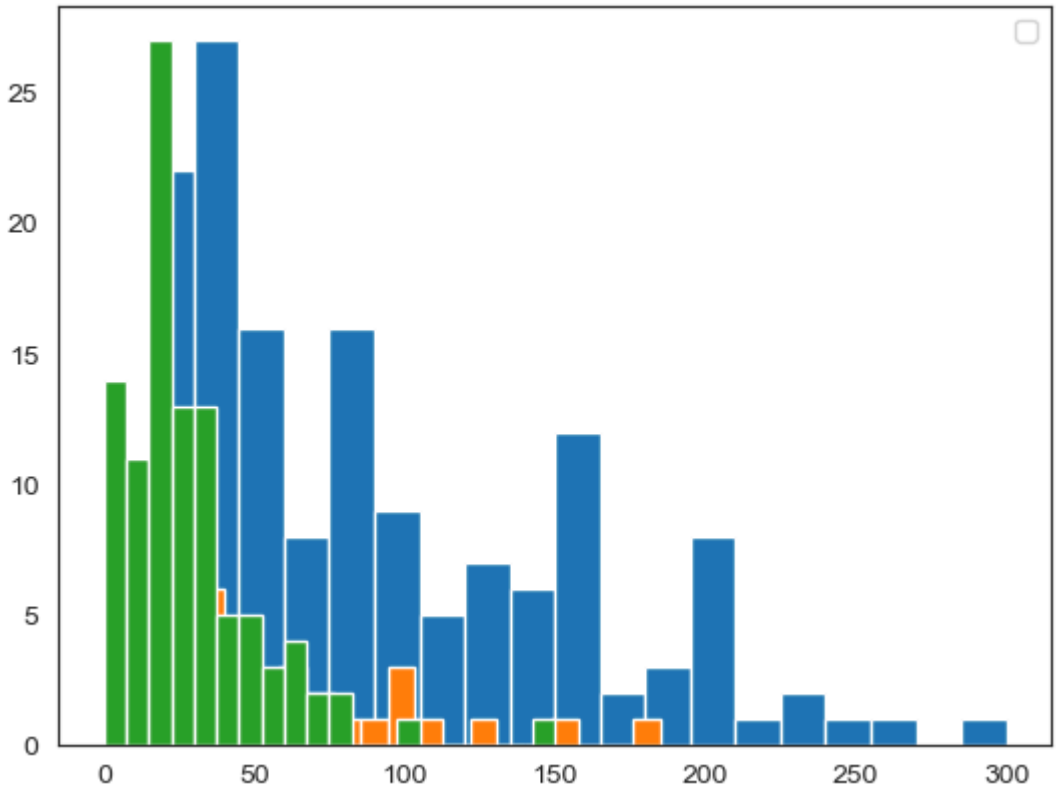


In [42]:
```
movies.head()
```

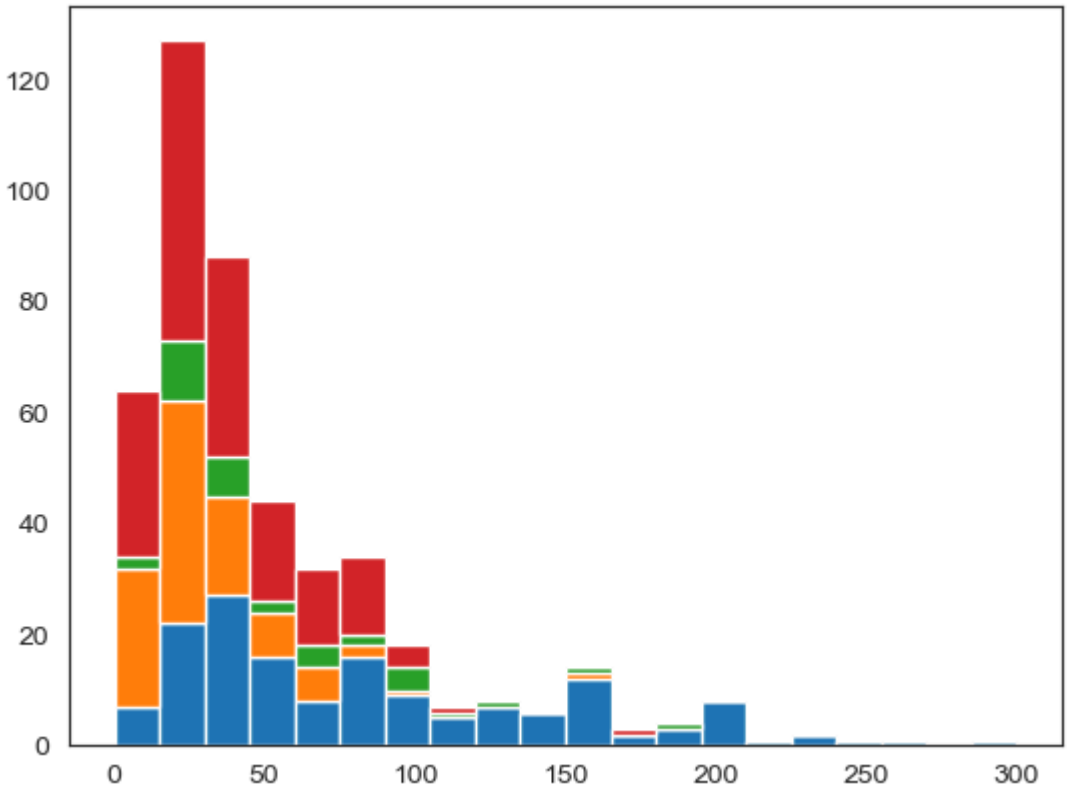
Out[42]:

	Film	Genre	CriticRatings	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]:
```
#Below plots are stacked histogram because overlaped

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()
plt.show()
```



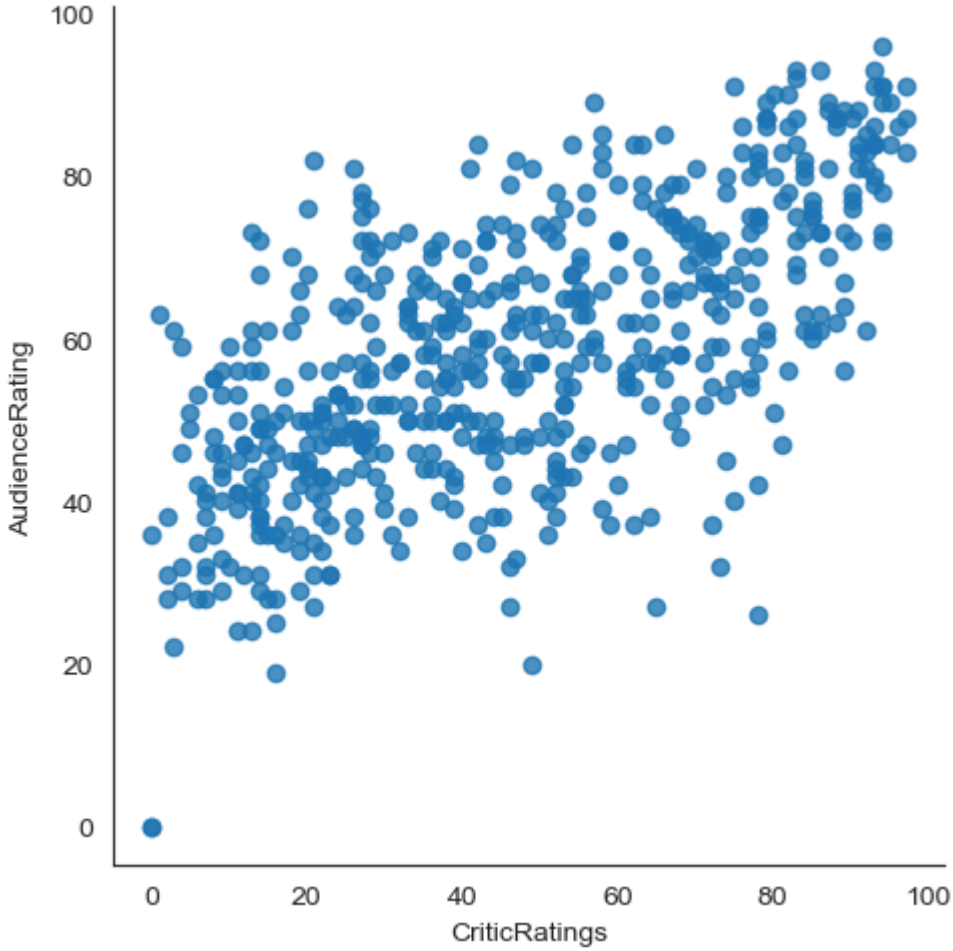
In [44]:
```
plt.hist([movies[movies.Genre == 'Action'].BudgetMillions,\
          movies[movies.Genre == 'Drama'].BudgetMillions,\
           movies[movies.Genre == 'Thriller'].BudgetMillions,\
           movies[movies.Genre == 'Comedy'].BudgetMillions],
            bins = 20,stacked =True)
plt.show()
```



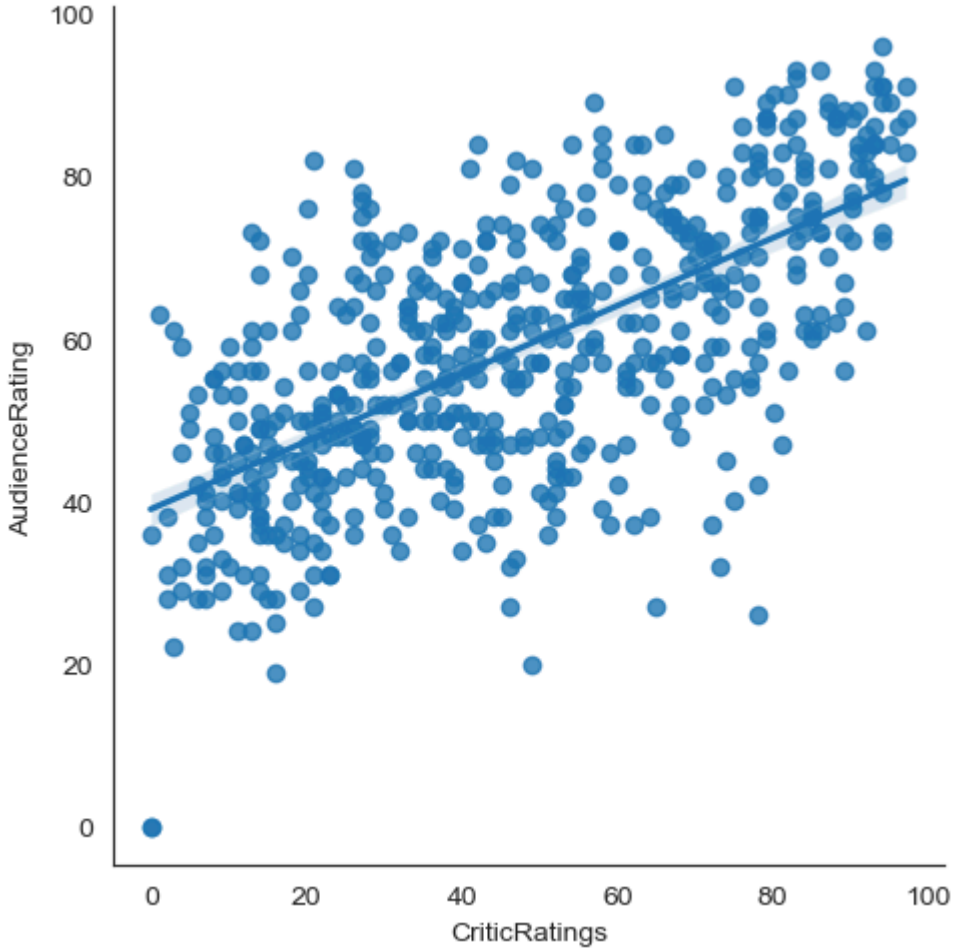
In [45]:
```
# if you have 100 categories you cannot copy and paste all the things
for gen in movies.Genre.cat.categories:
    print(gen)
```

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

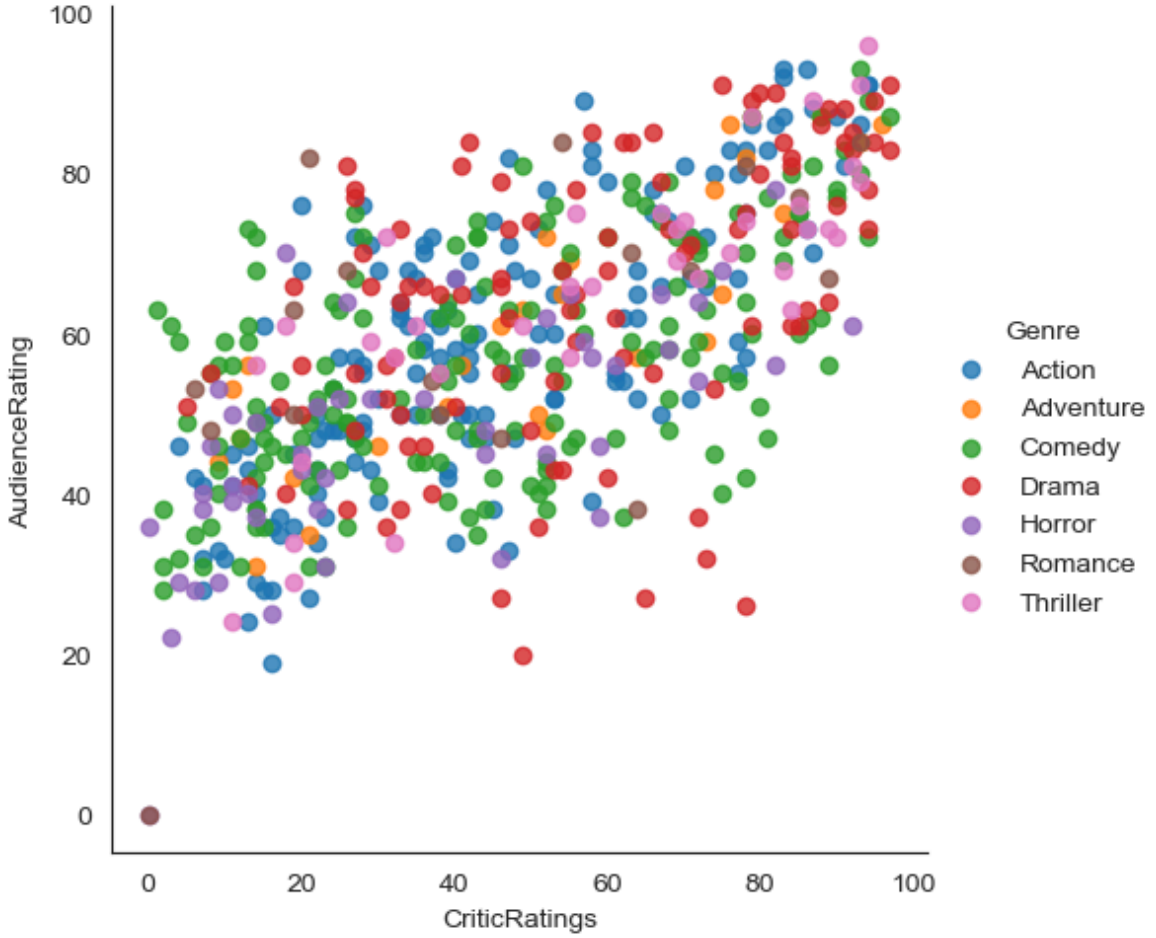
In [46]:
```
visl = sns.lmplot(data = movies,x ='CriticRatings',y='AudienceRating',fit_reg=Fa
```



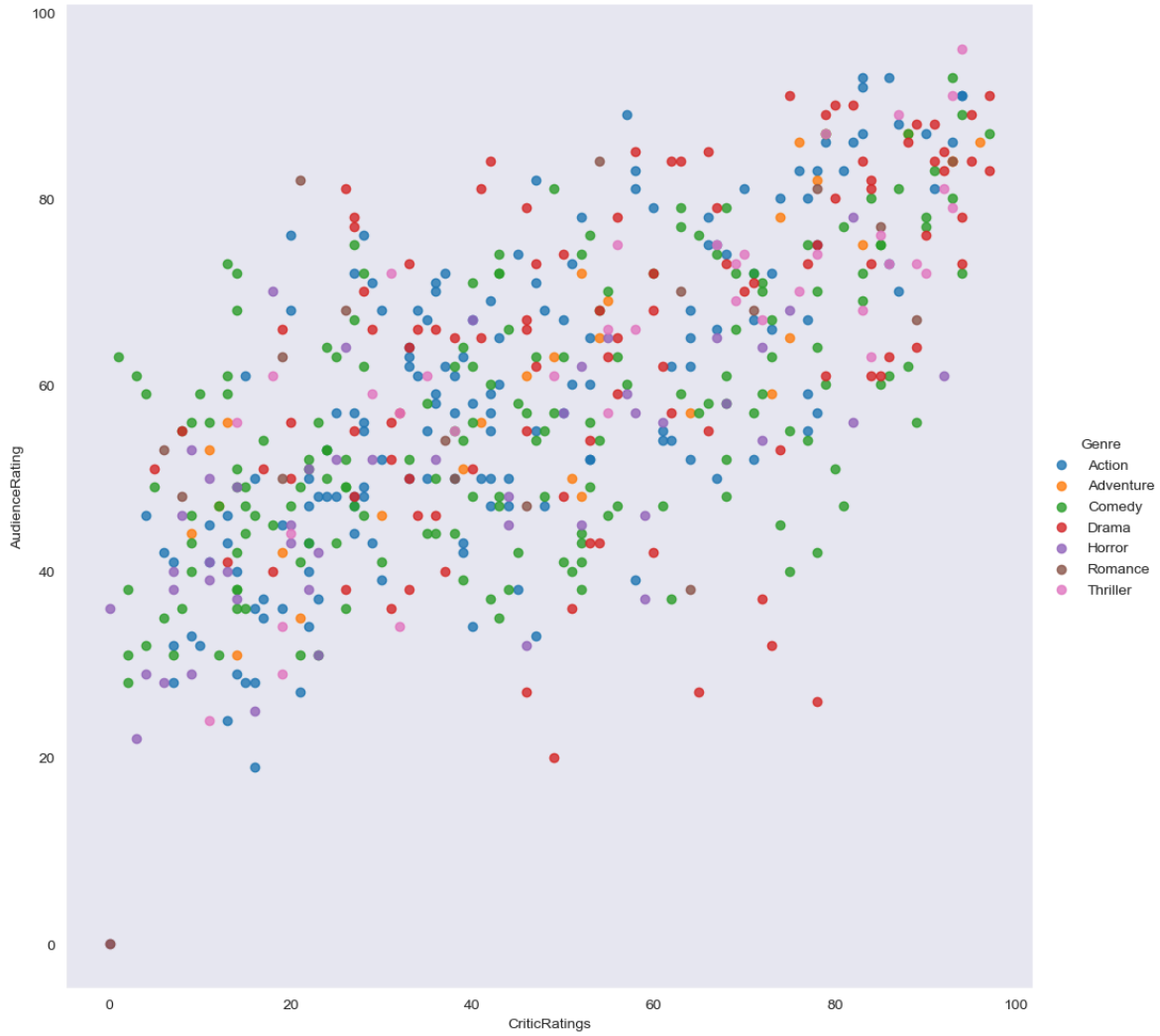
In [47]:
```
visl = sns.lmplot(data = movies,x ='CriticRatings',y='AudienceRating',fit_reg=Tr
```



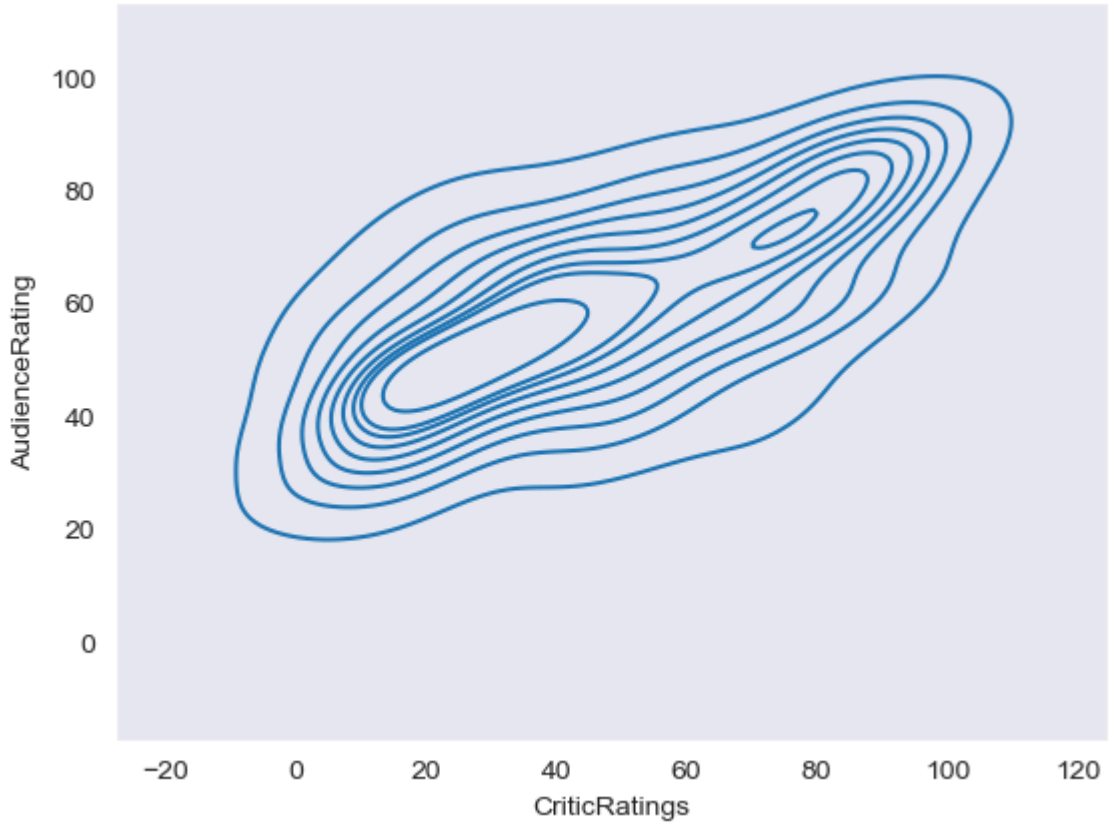
In [48]:
```
visl = sns.lmplot(data = movies,x ='CriticRatings',y='AudienceRating',fit_reg=Fa
```



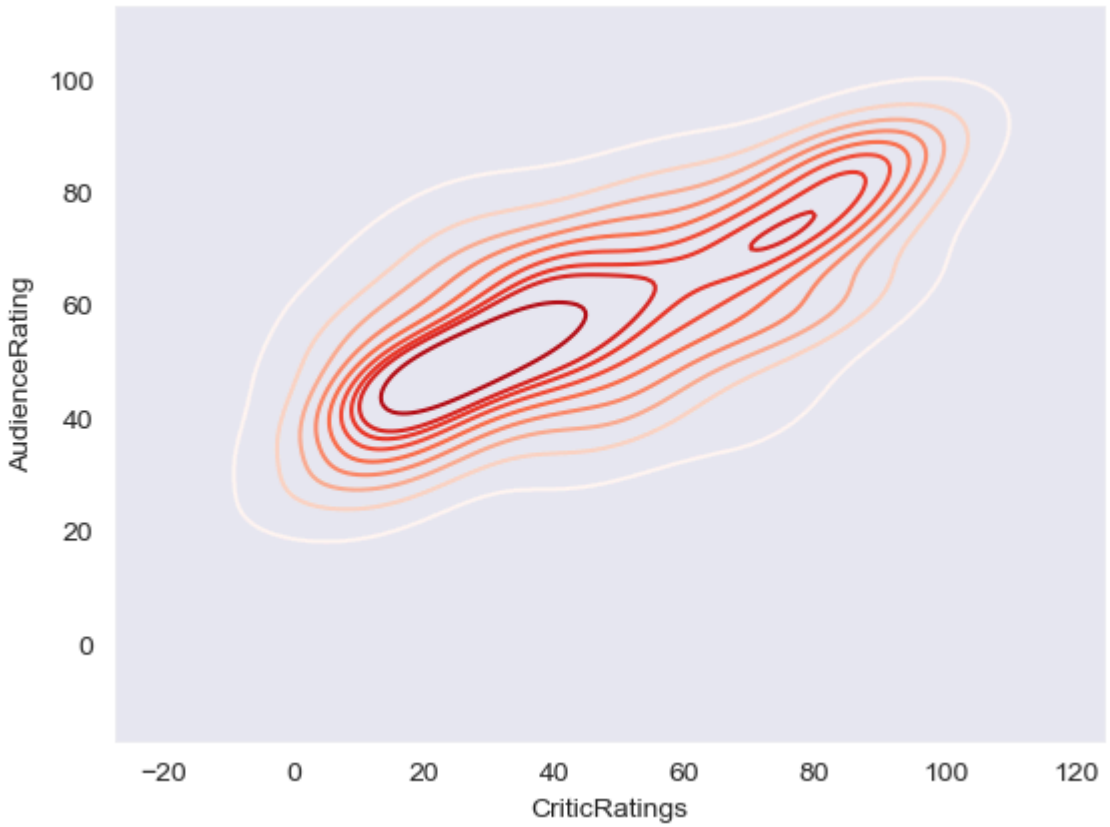
In [49]:
```
sns.set_style('dark')
visl = sns.lmplot(data=movies, x='CriticRatings',y='AudienceRating',fit_reg=Fals
```



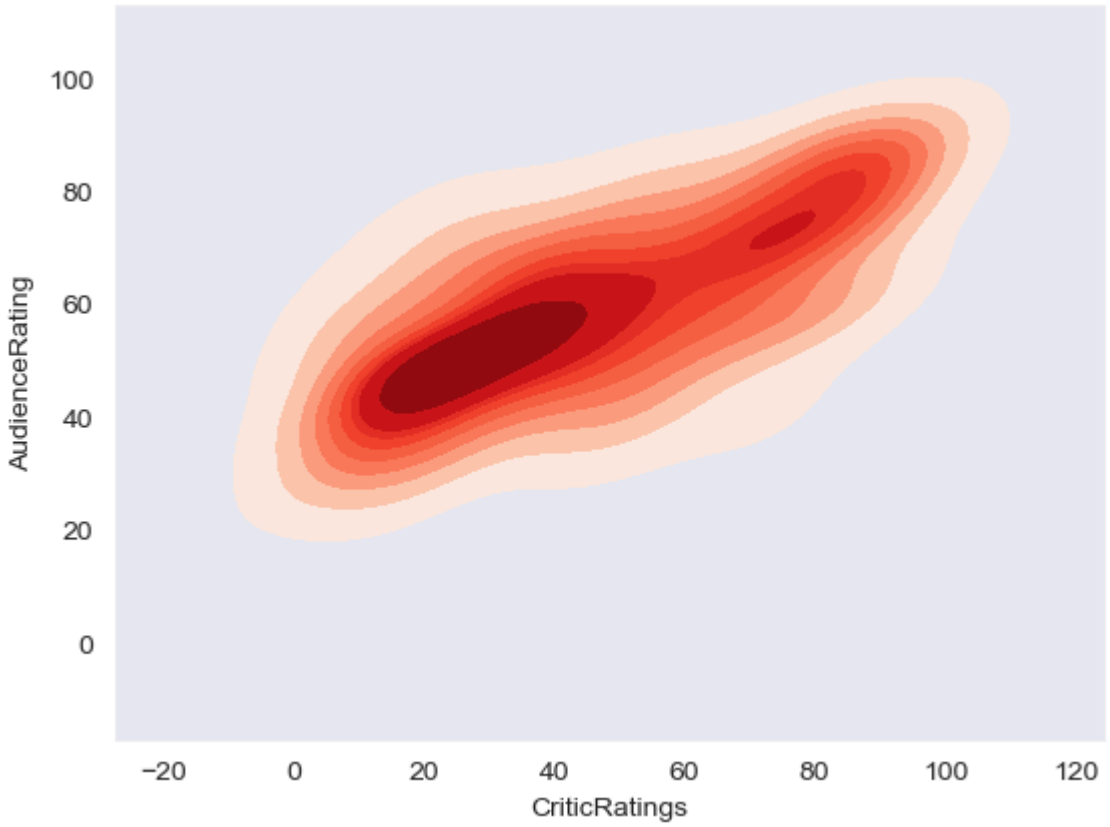
In [50]:
```
k1 = sns.kdeplot(x=movies.CriticRatings,y=movies.AudienceRating)
```



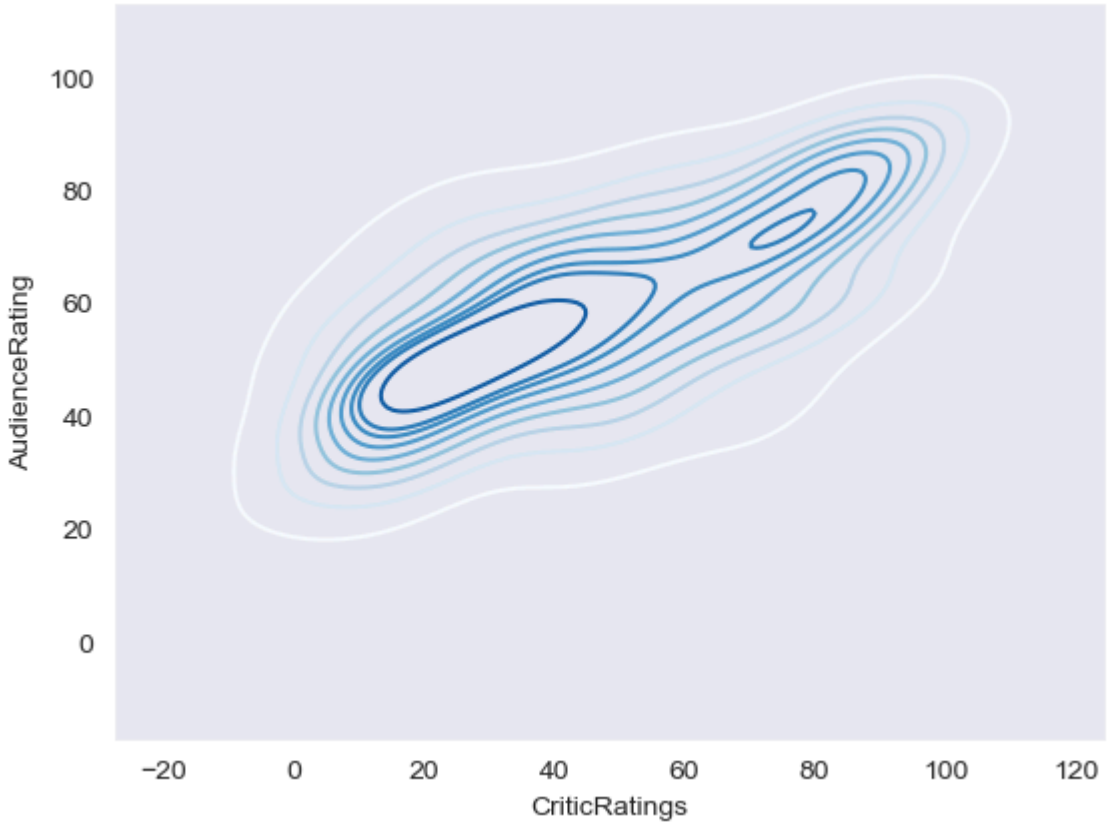
In [51]:
```
k1 = sns.kdeplot(x=movies.CriticRatings,y=movies.AudienceRating,shade =False,sha
```



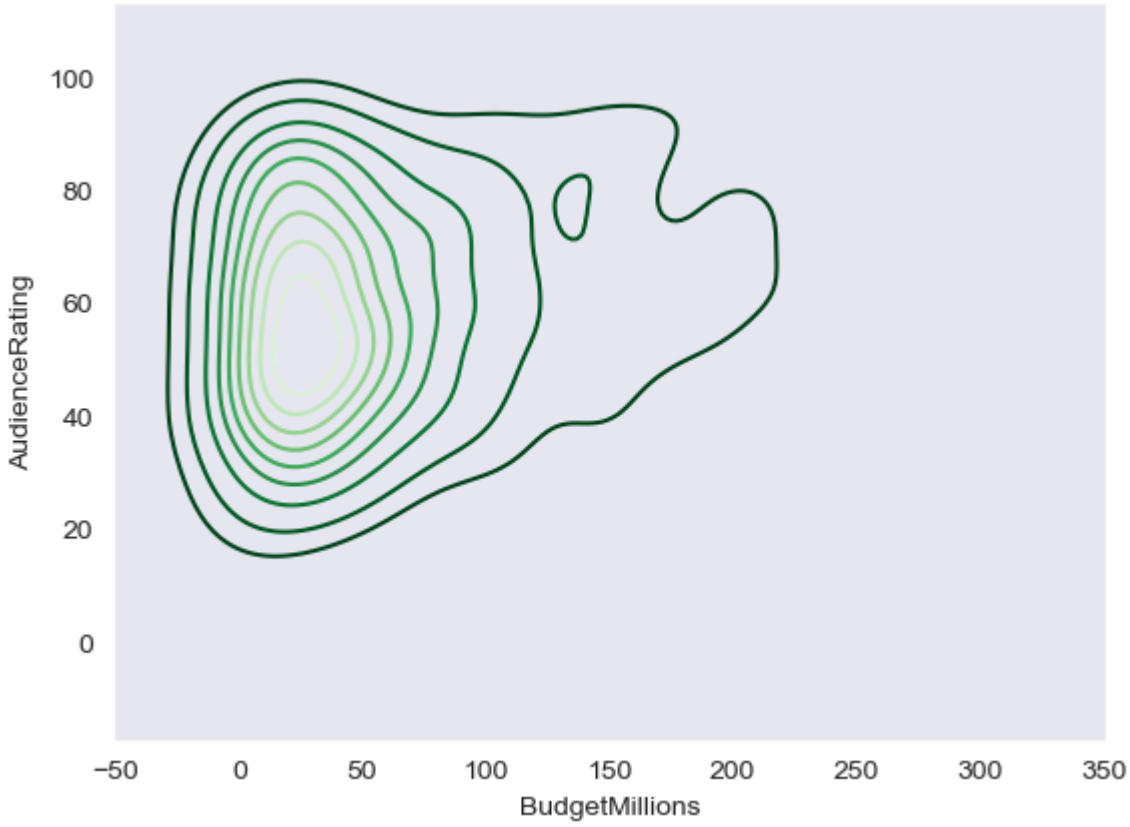
In [52]:
```
k1 = sns.kdeplot(x=movies.CriticRatings,y=movies.AudienceRating,shade =True,shad
```



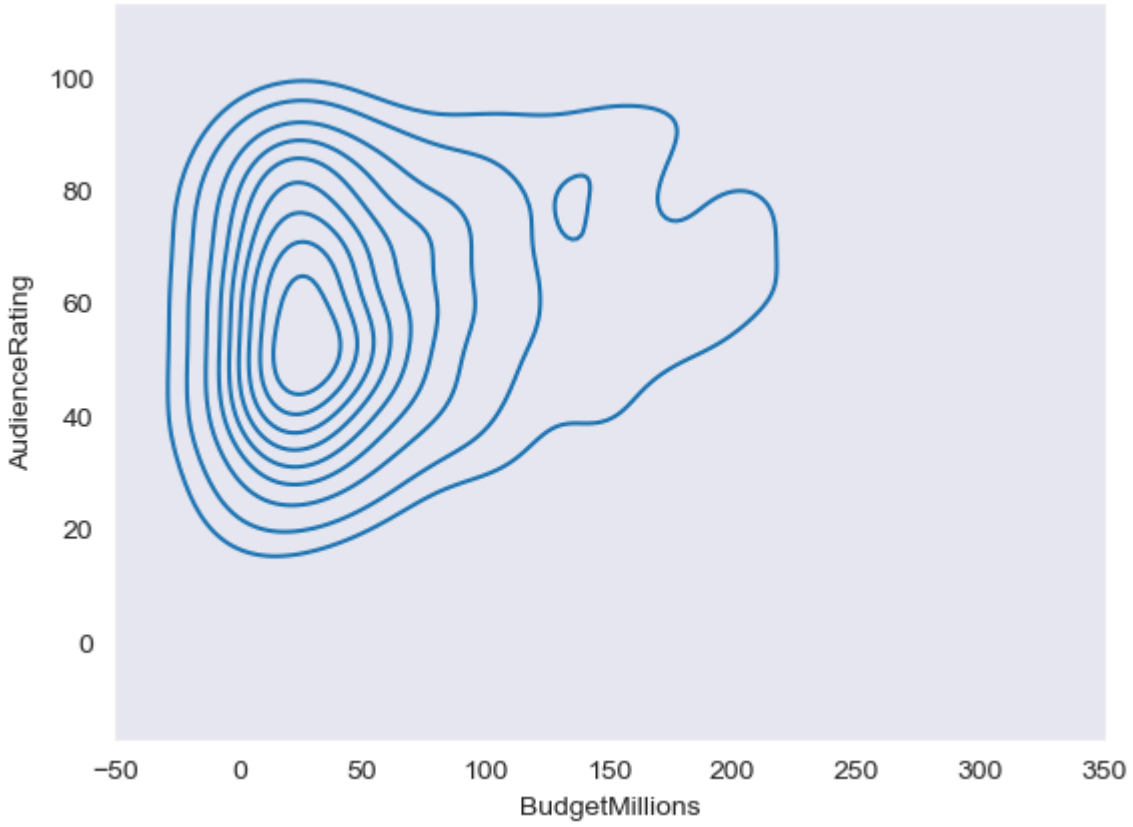
In [53]:
```
k1 = sns.kdeplot(x=movies.CriticRatings,y=movies.AudienceRating,shade_lowest=Fal
```



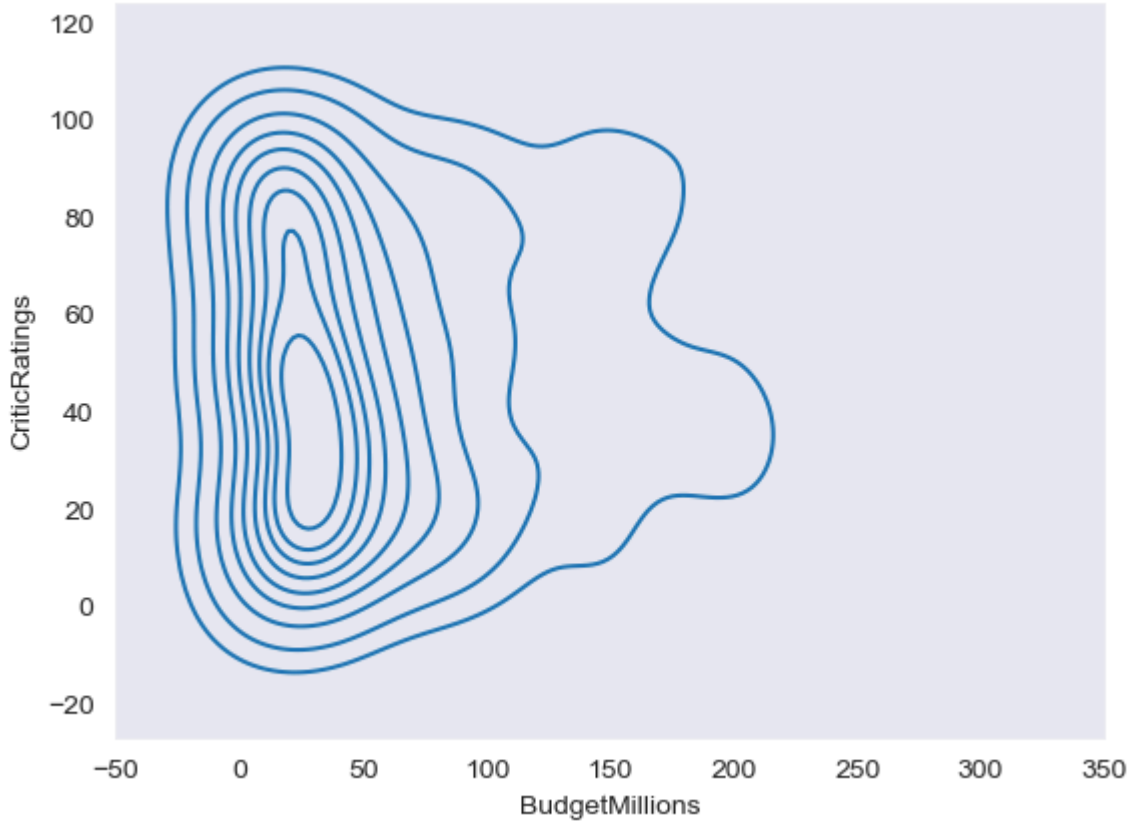
In [54]:
```
sns.set_style('dark')
k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,shade_lowest=Fals
```



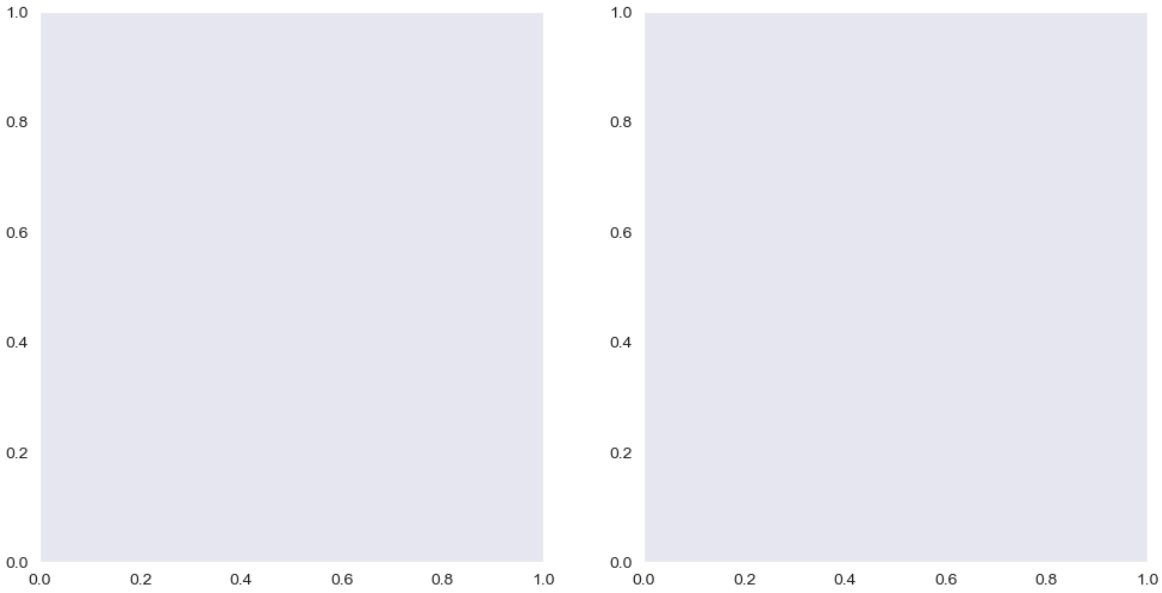
In [55]:
```
sns.set_style('dark')
k1=sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating)
```



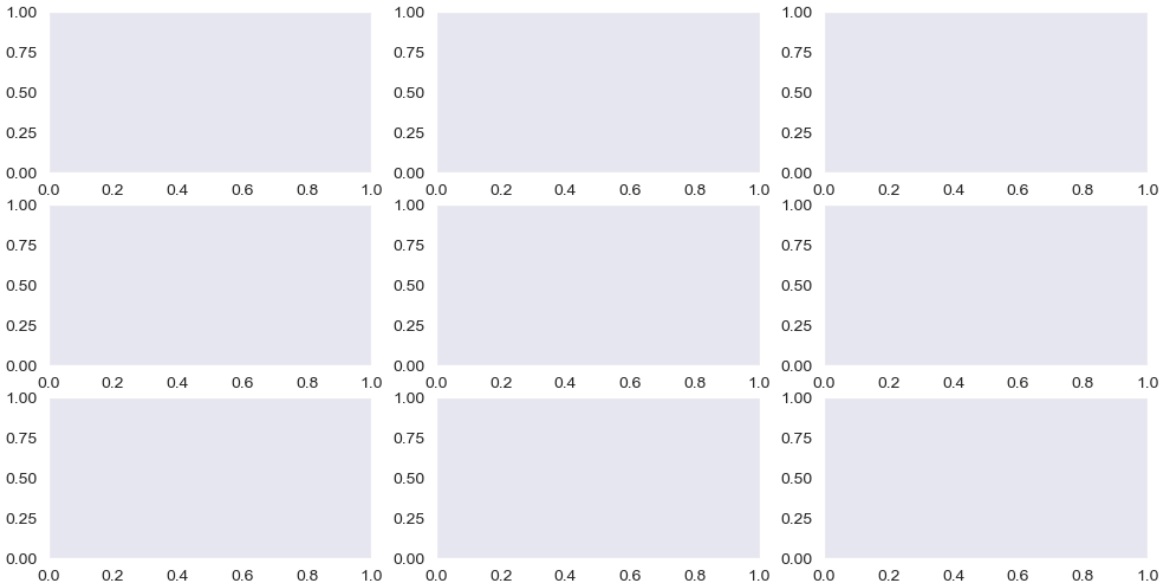
In [56]:
```
k2=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRatings)
```



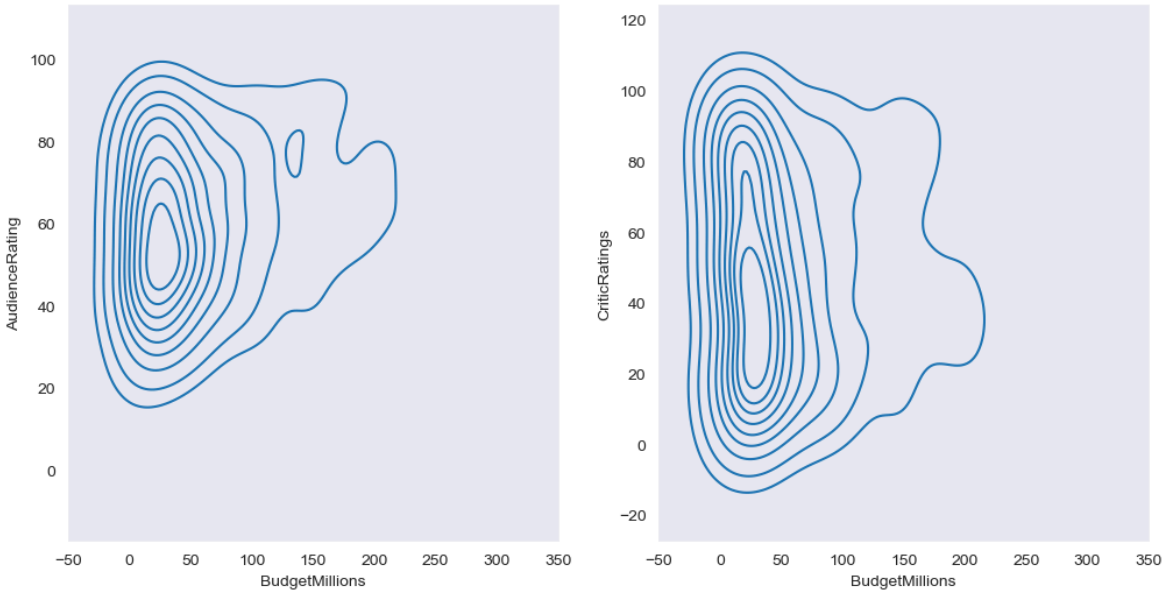
In [57]:
```
#subplot
f,ax=plt.subplots(1,2,figsize=(12,6))
```



In [58]:
```
f,ax=plt.subplots(3,3,figsize=(12,6))
```



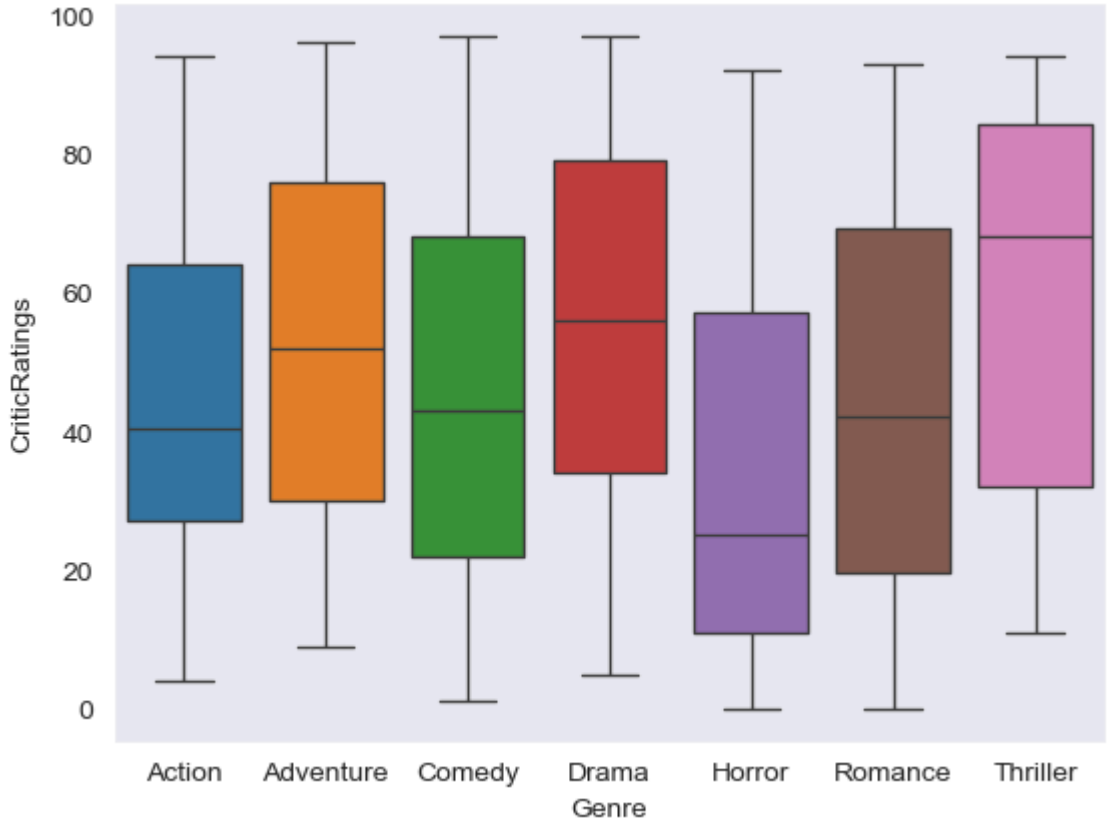
In [59]:
```
f,axes = plt.subplots(1,2,figsize =(12,6))

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



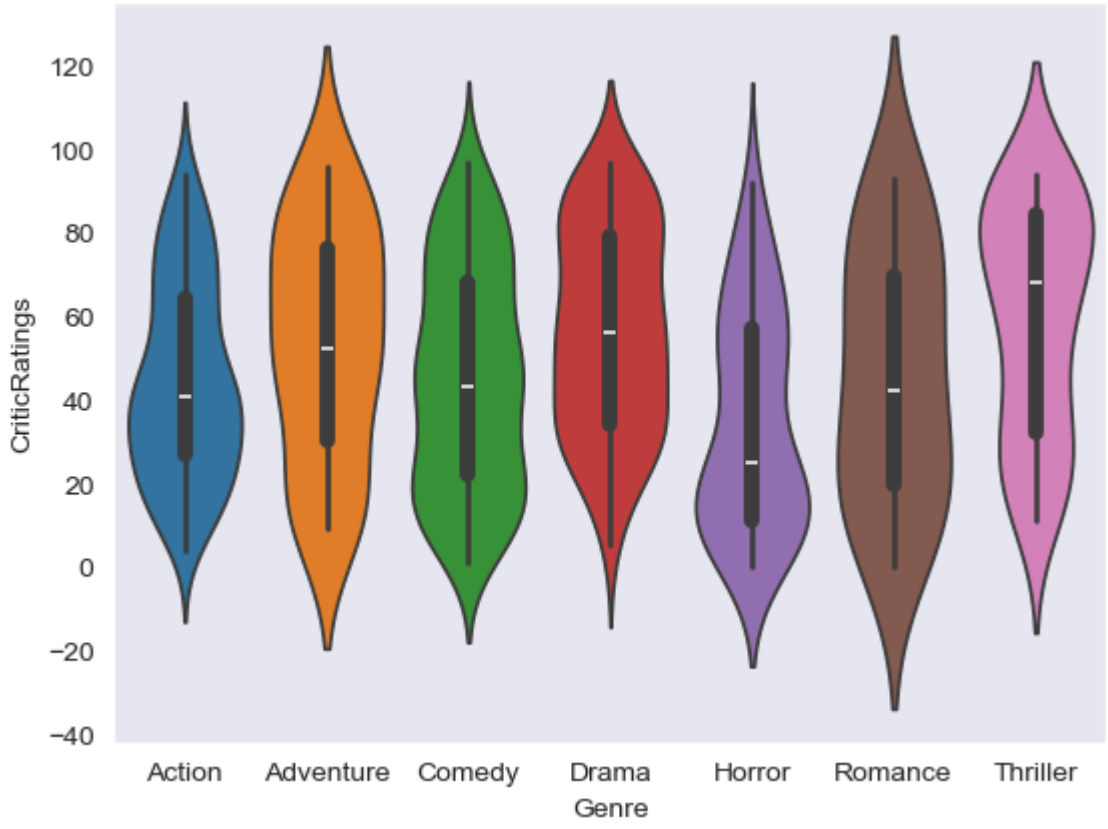
In [60]:
```
axes
```

Out[60]:
```
array([<Axes: xlabel='BudgetMillions', ylabel='AudienceRating'>,
       <Axes: xlabel='BudgetMillions', ylabel='CriticRatings'>],
      dtype=object)
```

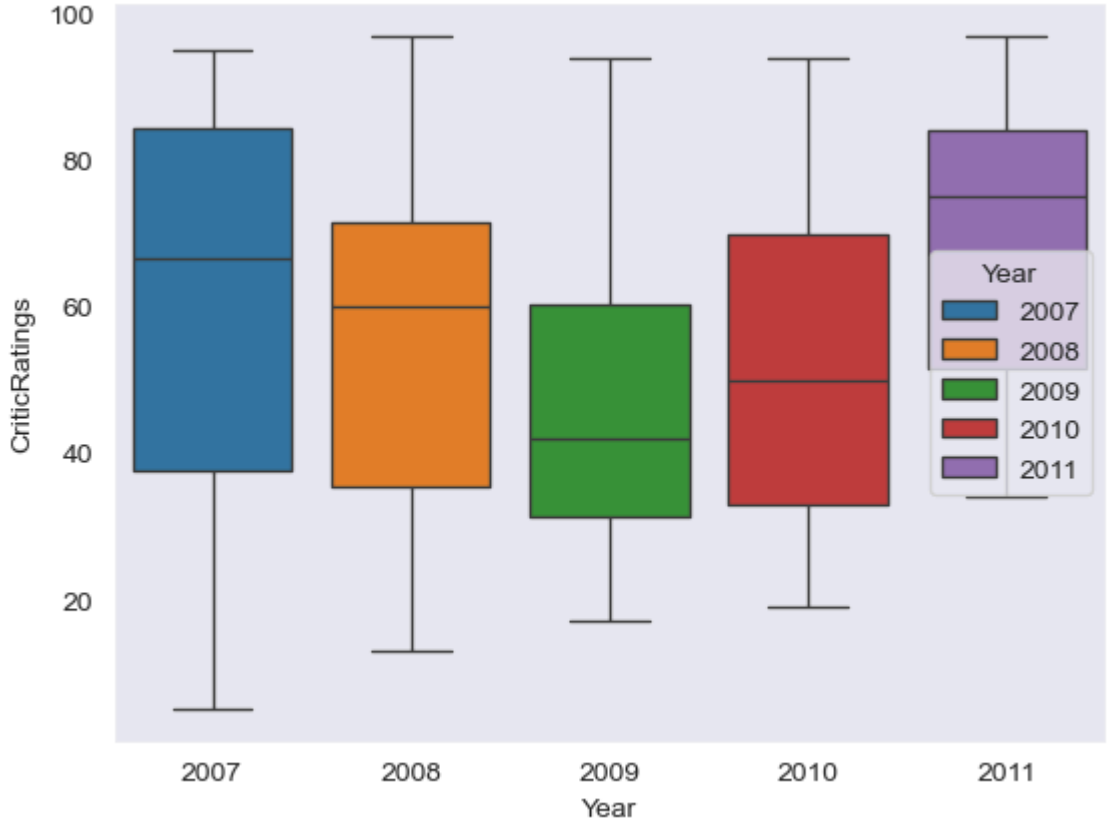
In [61]:
```
#Box plots

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



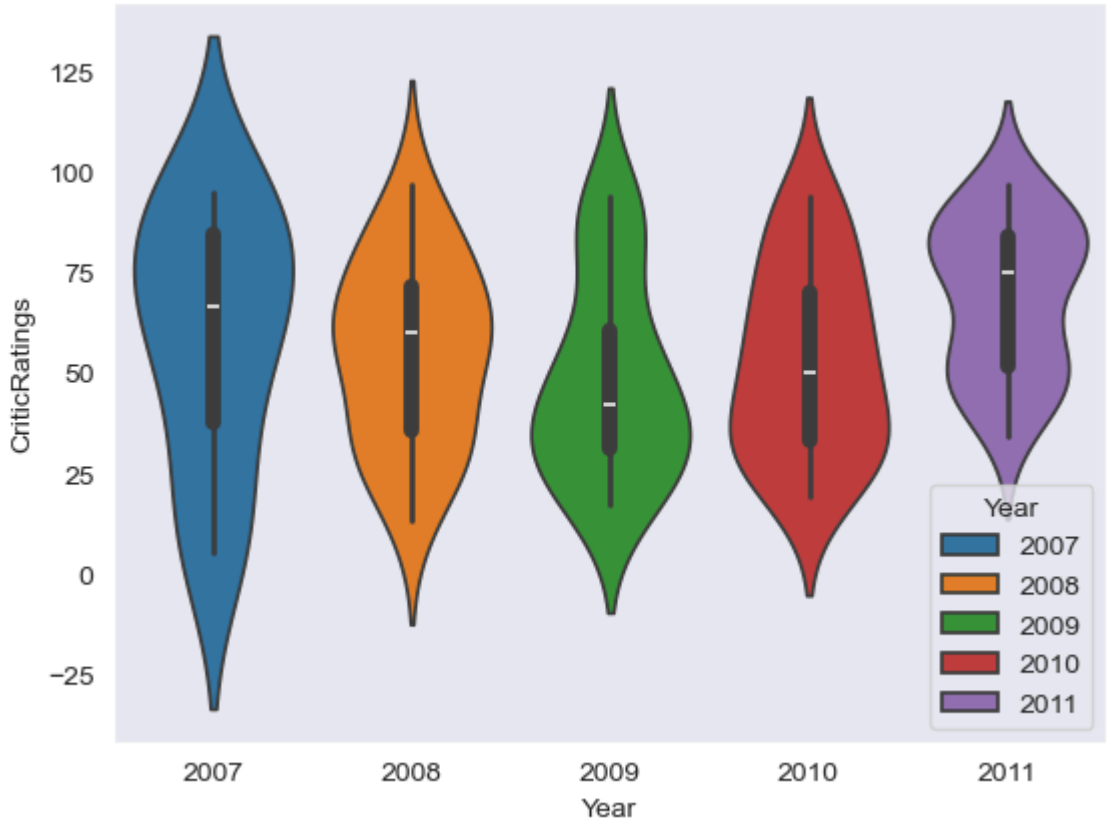
In [62]:
```
#violin plot
z= sns.violinplot(data =movies,x='Genre',y='CriticRatings',hue='Genre')
```



In [63]:
```
w1 = sns.boxplot(data = movies[movies.Genre=='Drama'],x='Year',y='CriticRatings'
```



In [64]:
```
z= sns.violinplot(data=movies[movies.Genre=='Drama'],x='Year',y='CriticRatings',
```

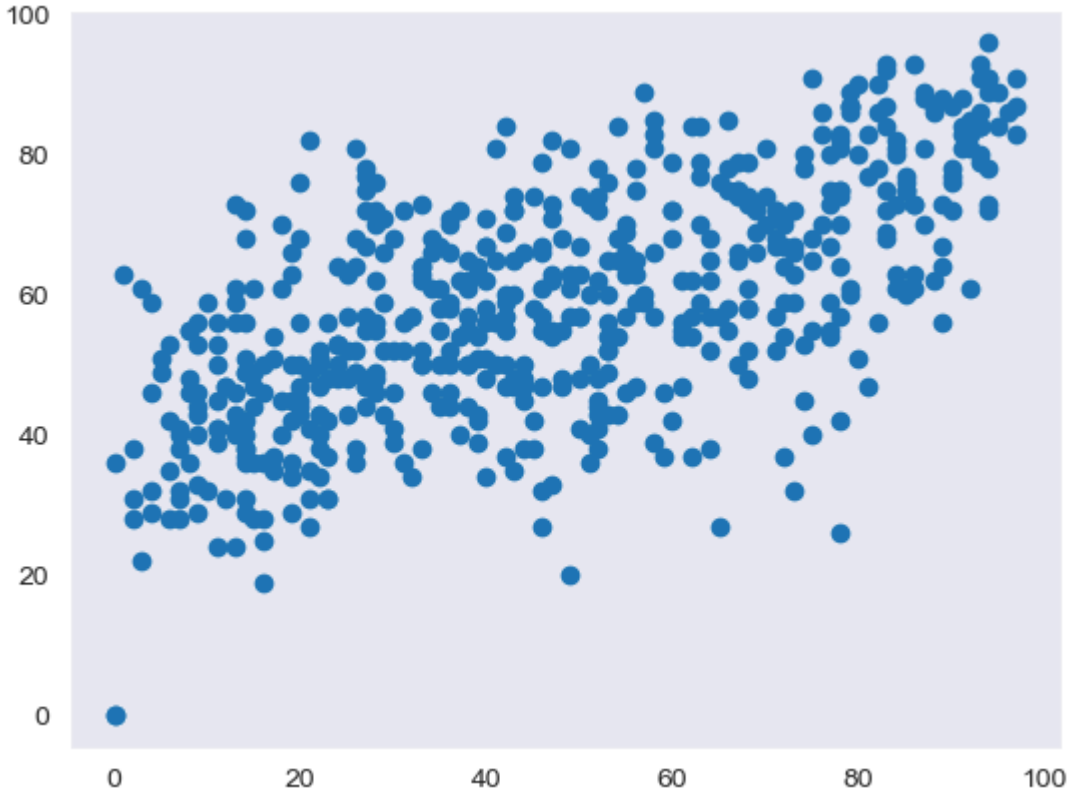


In [65]:
```
# Creating a facet grid
```

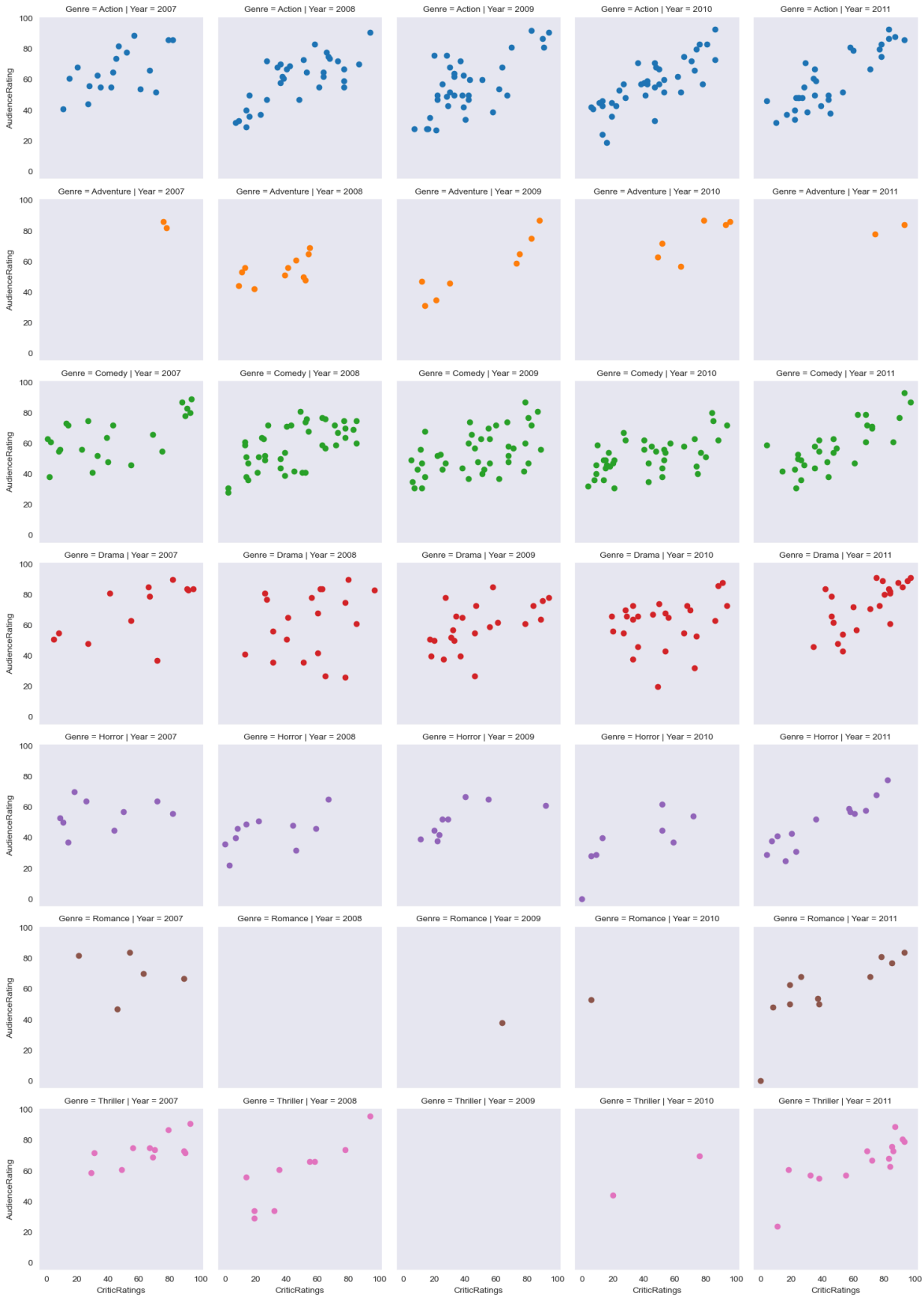
In [66]:
```
g = sns.FacetGrid(movies,row ='Genre',col = 'Year',hue = 'Genre')    #kind of su
```

In [68]:
```
plt.scatter(x=movies.CriticRatings,y=movies.AudienceRating)
```

Out[68]:
```
<matplotlib.collections.PathCollection at 0x262cba15e50>
```



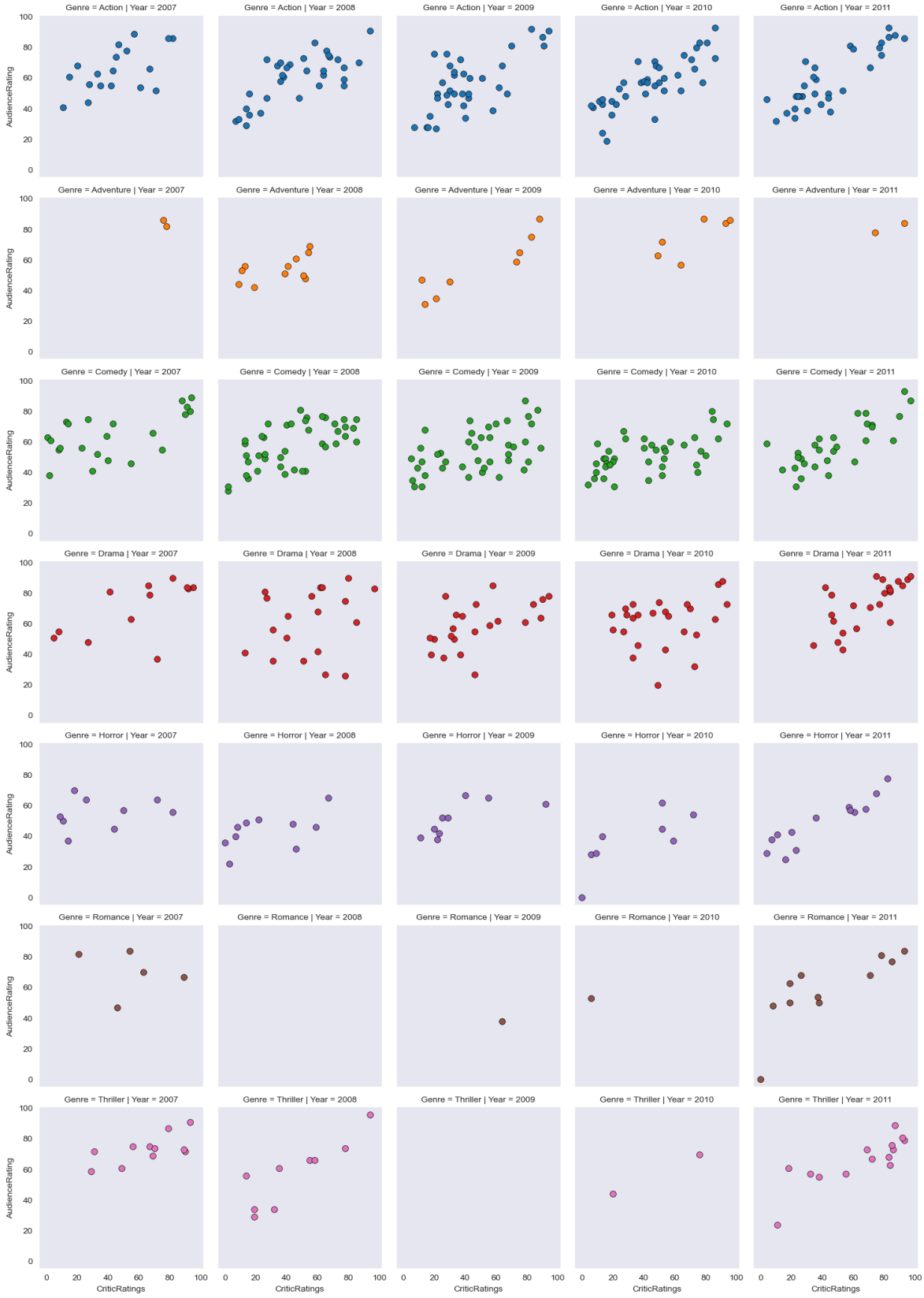
In [69]:
```
g = sns.FacetGrid(movies,row ='Genre',col = 'Year',hue = 'Genre')
g= g.map(plt.scatter,'CriticRatings','AudienceRating')     #scatterplots are map
```



In [70]:

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

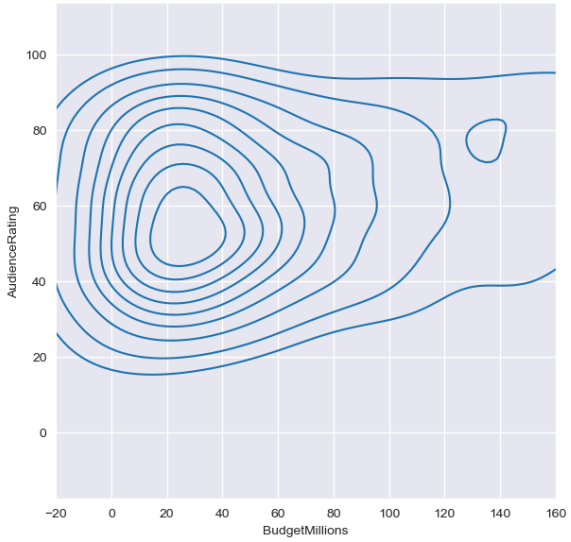
In [71]:
```
g = sns.FacetGrid(movies,row ='Genre',col = 'Year',hue = 'Genre')
kws=dict(s=50,linewidth=0.5,edgecolor='black')
g = g.map(plt.scatter,'CriticRatings','AudienceRating',**kws)
```

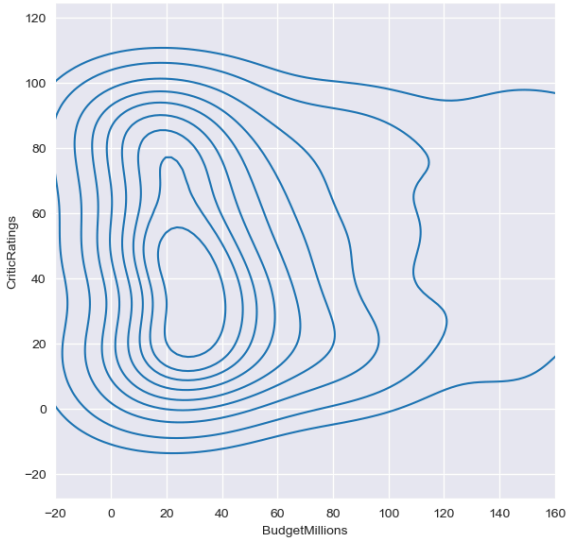


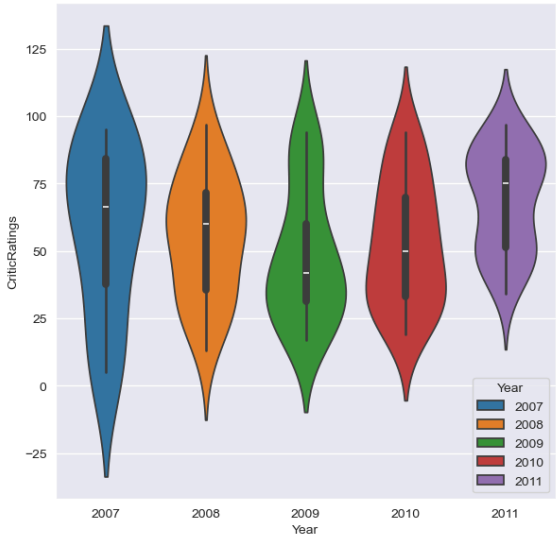
In [76]:

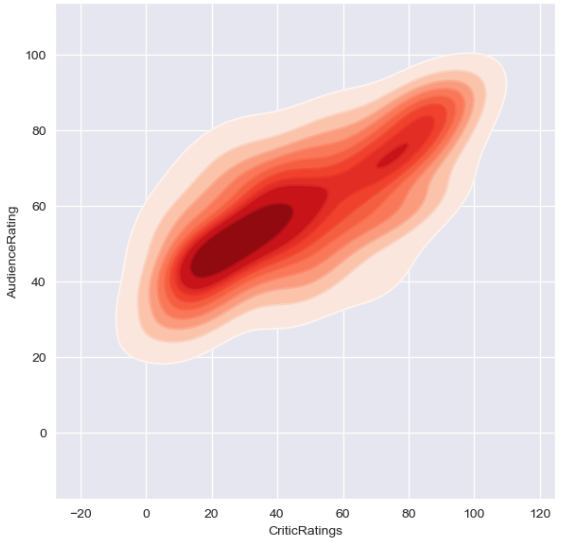
```
sns.set_style('darkgrid')
f,axes = plt.subplots(2,2,figsize =(15,15))

k1 = sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,ax=axes[0,0])
k2 = sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRatings,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='CriticRatings
```

```
k4=sns.kdeplot(x=movies.CriticRatings,y=movies.AudienceRating,shade = True,shade

k4b = sns.kdeplot(x=movies.CriticRatings,y=movies.AudienceRating,cmap = 'Reds',a
plt.show()
```



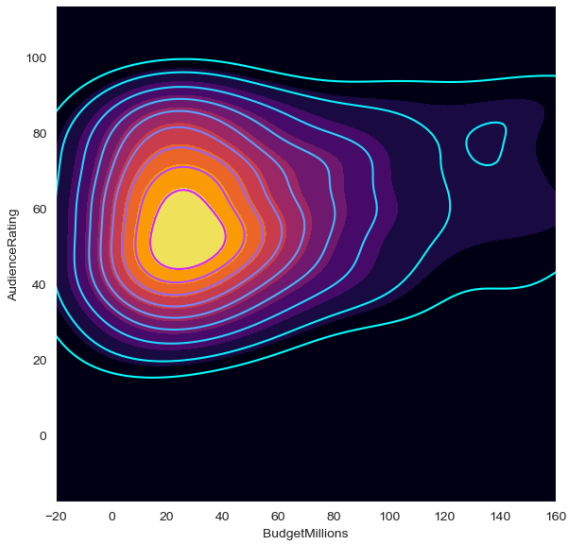


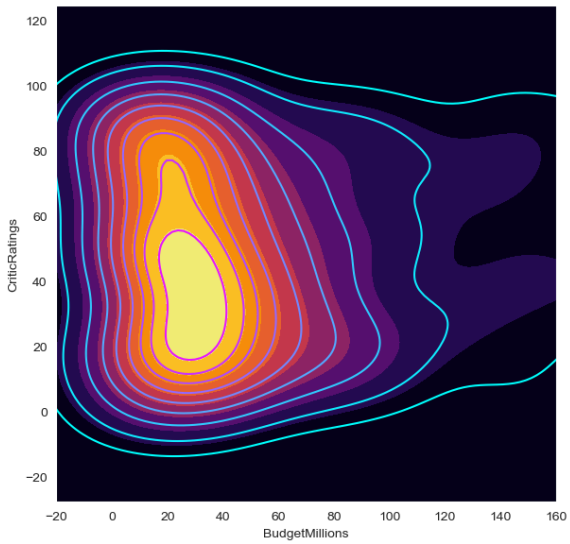


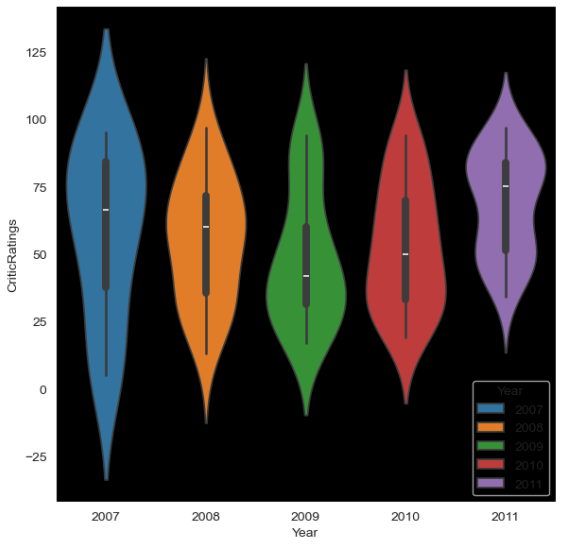


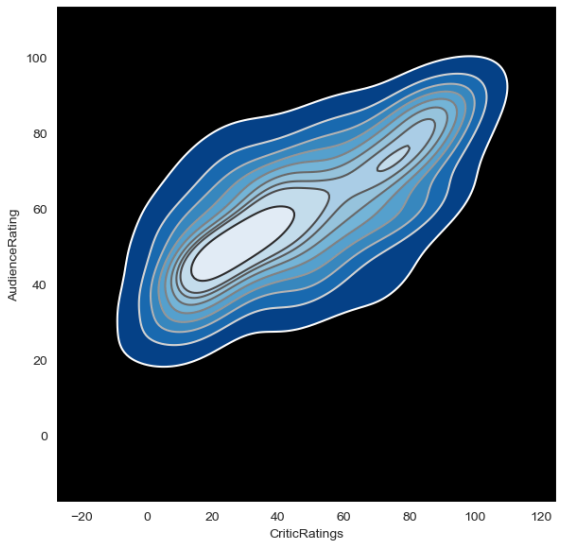
In [92]:
```
#style your dashboard using different color map

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,sh

k1b = sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRating,cmap='cool',ax
#plot[0,1]
k2 = sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRatings,shade=True,shade

k2b =sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRatings,cmap='cool',ax =

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

#plot[1,1]
k4 = sns.kdeplot(x=movies.CriticRatings,y=movies.AudienceRating,shade=True,shade
```

```
k4b = sns.kdeplot(x=movies.CriticRatings,y=movies.AudienceRating,cmap='gist_gray

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

plt.show()
```









In [ ]: