

```
In [1]: import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
In [2]: df=pd.read_csv('Movie-Rating.csv')
```

```
In [3]: df
```

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]: df.columns
```

```
Out[4]: Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
   'Budget (million $)', 'Year of release'],
   dtype='object')
```

```
In [5]: # Removed spaces & % removed noise characters from columns.  
df.columns=['Film','Genre','CriticRating','AudienceRating','BudgetMillions','Year']
```

```
In [6]: df.columns
```

```
Out[6]: Index(['Film', 'Genre', 'CriticRating', 'AudienceRating', 'BudgetMillions',  
              'Year'],  
             dtype='object')
```

```
In [7]: df.head()
```

```
Out[7]:
```

	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 [8]: df.tail()
```

```
Out[8]:
```

	Film	Genre	CriticRating	AudienceRating	BudgetMillions	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 [9]: df.describe()
```

Out[9]:

	CriticRating	AudienceRating	BudgetMillions	Year
<b>count</b>	559.000000	559.000000	559.000000	559.000000
<b>mean</b>	47.309481	58.744186	50.236136	2009.152057
<b>std</b>	26.413091	16.826887	48.731817	1.362632
<b>min</b>	0.000000	0.000000	0.000000	2007.000000
<b>25%</b>	25.000000	47.000000	20.000000	2008.000000
<b>50%</b>	46.000000	58.000000	35.000000	2009.000000
<b>75%</b>	70.000000	72.000000	65.000000	2010.000000
<b>max</b>	97.000000	96.000000	300.000000	2011.000000

In [10]: `df.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]: `# if you look at the year the data type is int but when you look at the mean value it showing 2009 which is meaningless  
# we have to change to category type  
# also from object datatype we will convert to category datatypes.  
df.Film=df.Film.astype('category')  
df.Genre=df.Genre.astype('category')  
df.Year=df.Year.astype('category')`In [12]: `df.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 [13]: `df.describe()`

Out[13]:

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

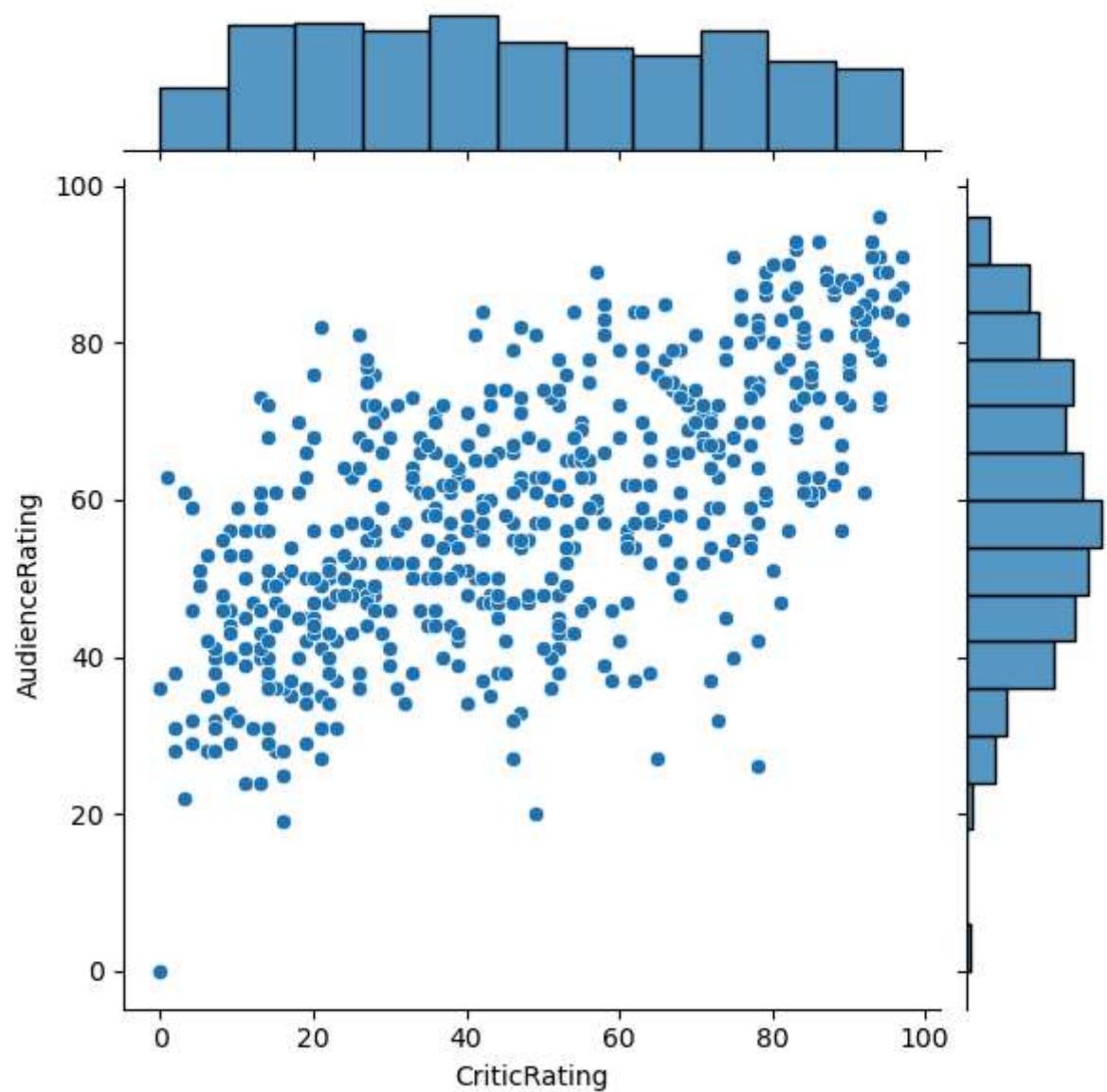
In [14]: `%matplotlib inline`

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

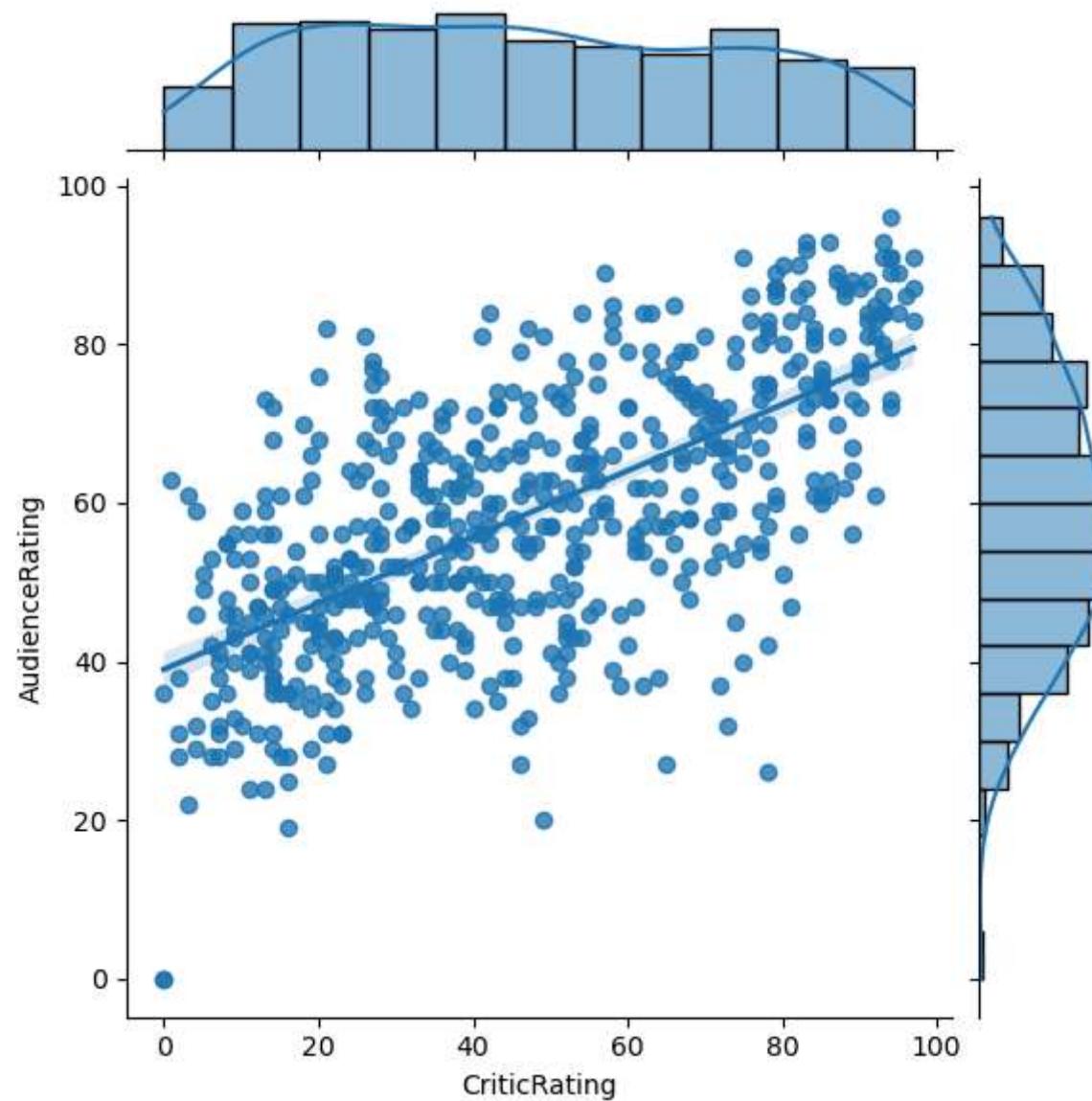
In [15]: `#joint plot`

```
# Audience rating is more dominant than critics rating
# Based on this we find out as most people are most likelihood to watch audience rating & less likely to watch critics
```

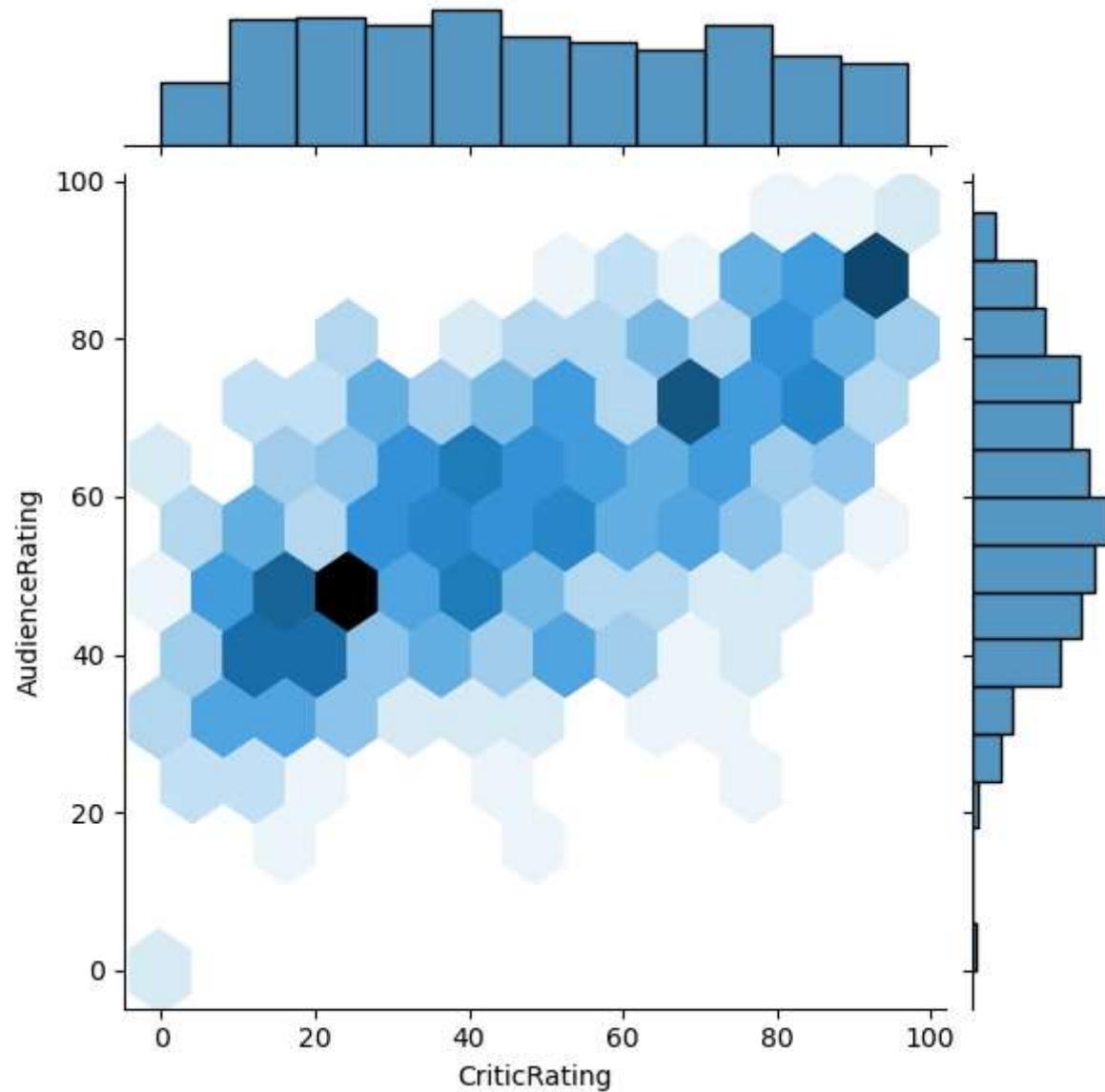
```
# Let me explain the excel - if you filter audience rating & critic rating. critic rating has very low values compare  
j=sns.jointplot(data=df,x='CriticRating',y='AudienceRating')
```



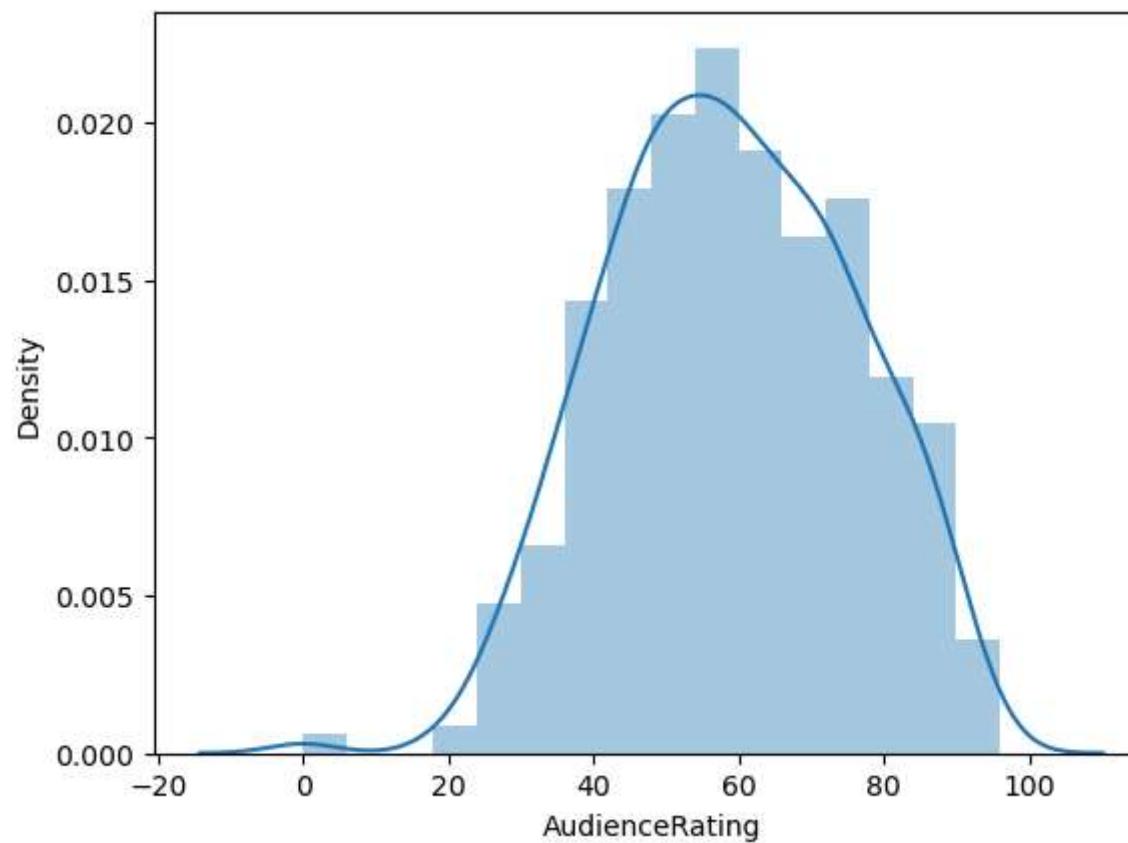
```
In [16]: #joint plot  
j=sns.jointplot(data=df,x='CriticRating',y='AudienceRating',kind='reg')
```



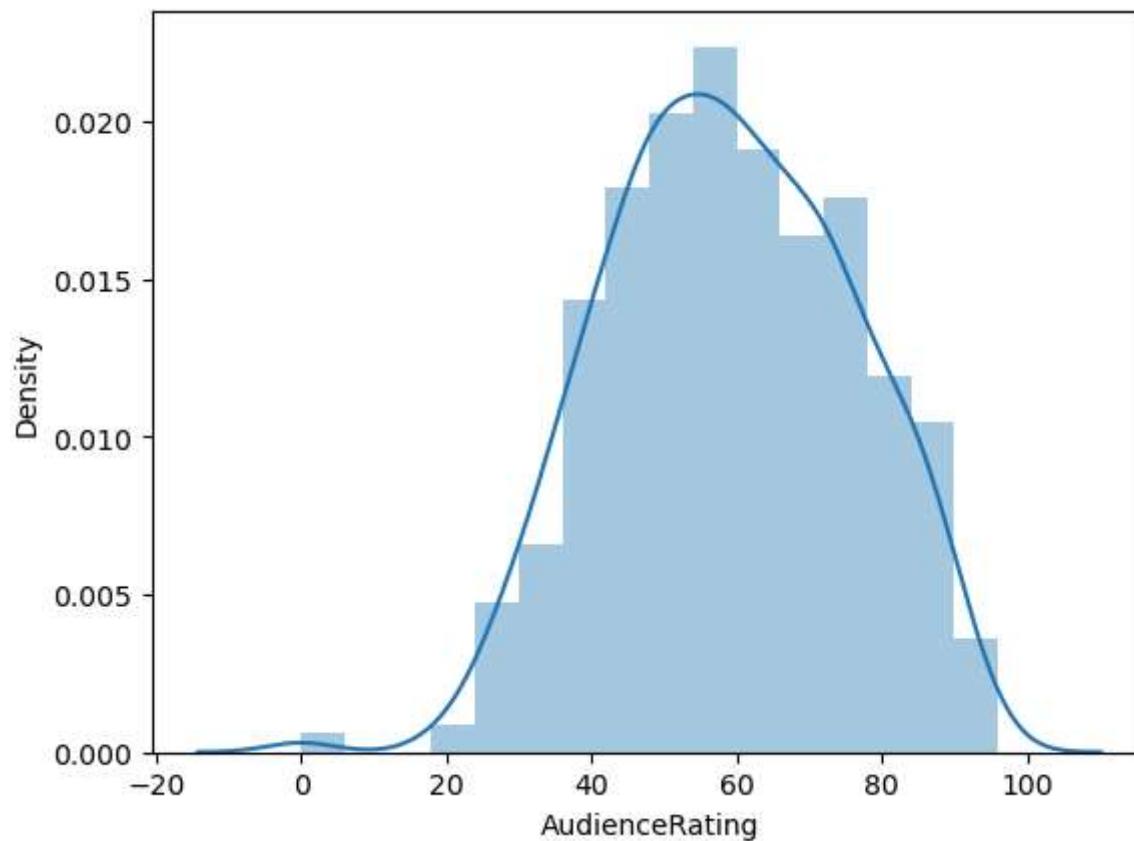
```
In [17]: j=sns.jointplot(data=df,x='CriticRating',y='AudienceRating',kind='hex')
```



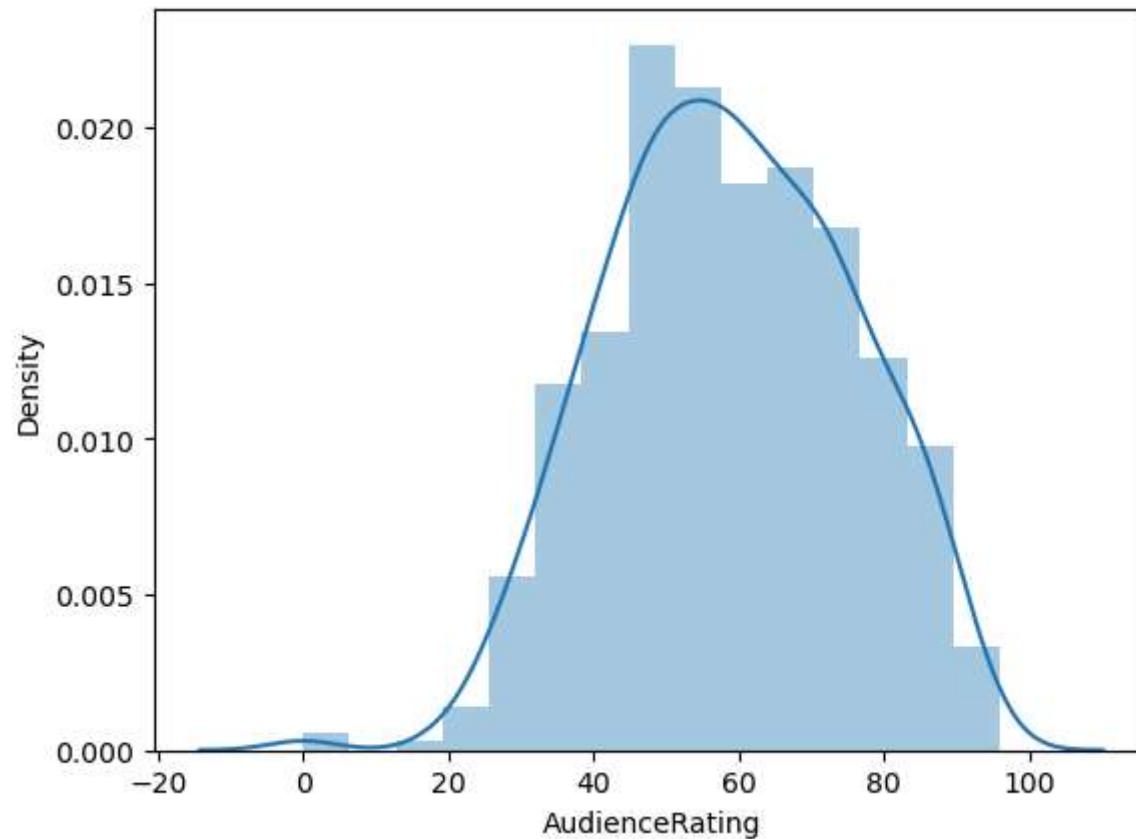
```
In [18]: m1=sns.distplot(df.AudienceRating)
```



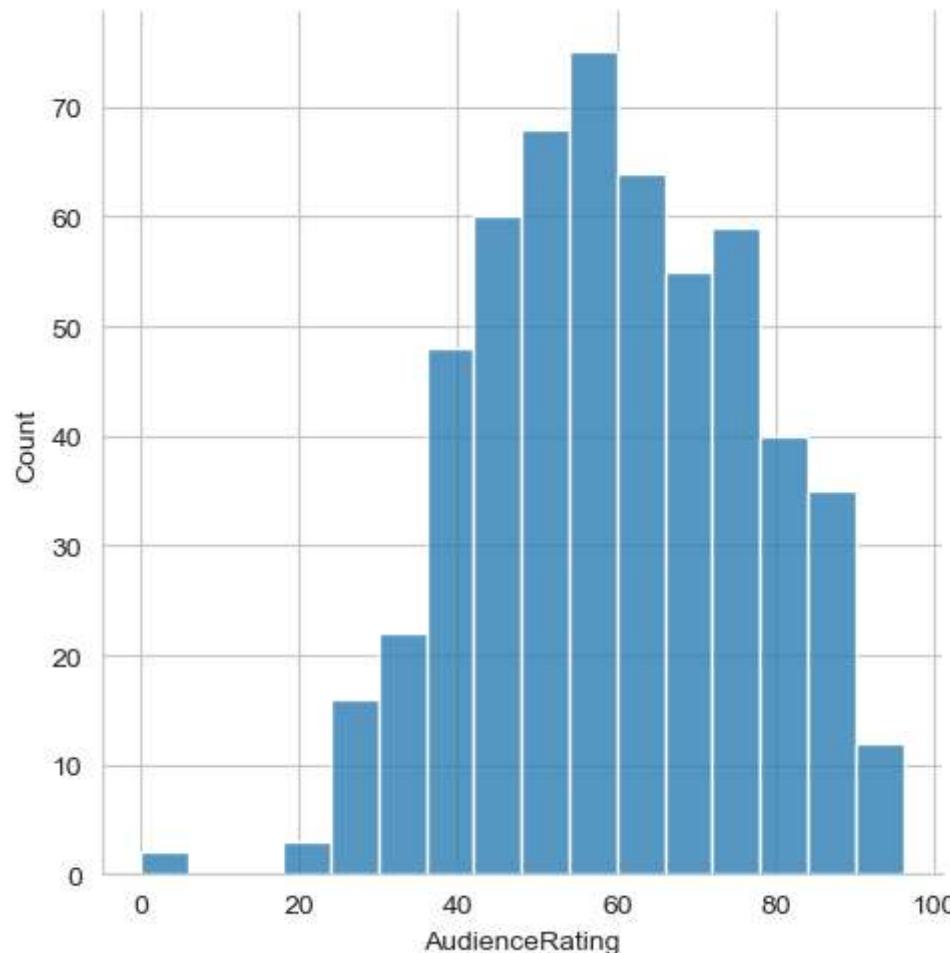
```
In [19]: m2=sns.distplot(df.AudienceRating)
```



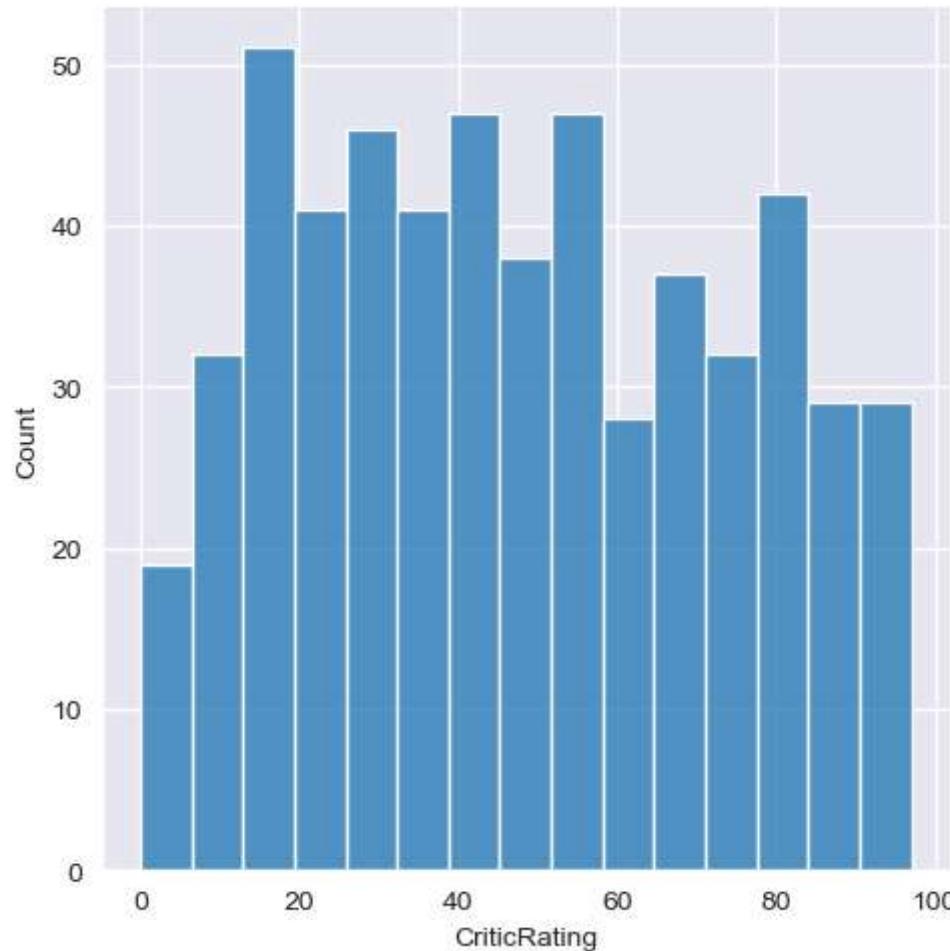
```
In [20]: #Normal Distribution & called bell curve.  
m3=sns.distplot(df.AudienceRating,bins=15)
```



```
In [21]: sns.set_style('whitegrid') #darkgrid, whitegrid, dark, white, ticks  
m2=sns.displot(df.AudienceRating)
```

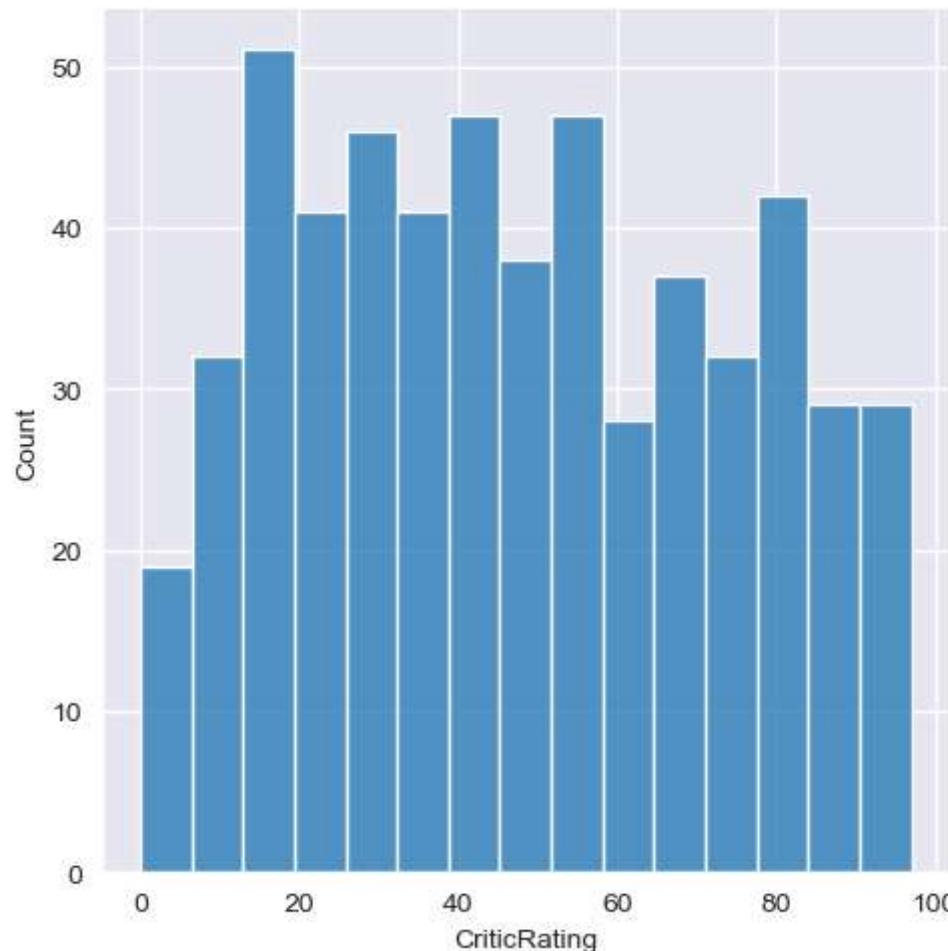


```
In [22]: sns.set_style('darkgrid')
m2=sns.displot(df.CriticRating,bins=15)
```

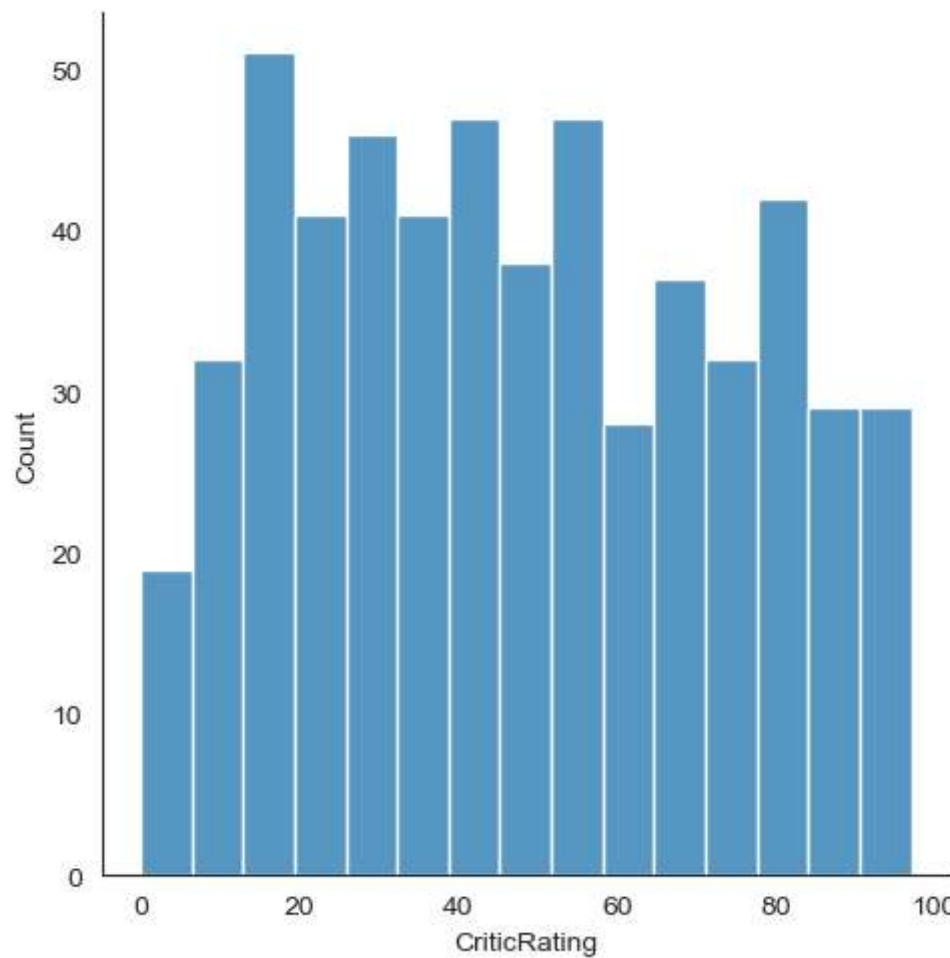


In [23]:

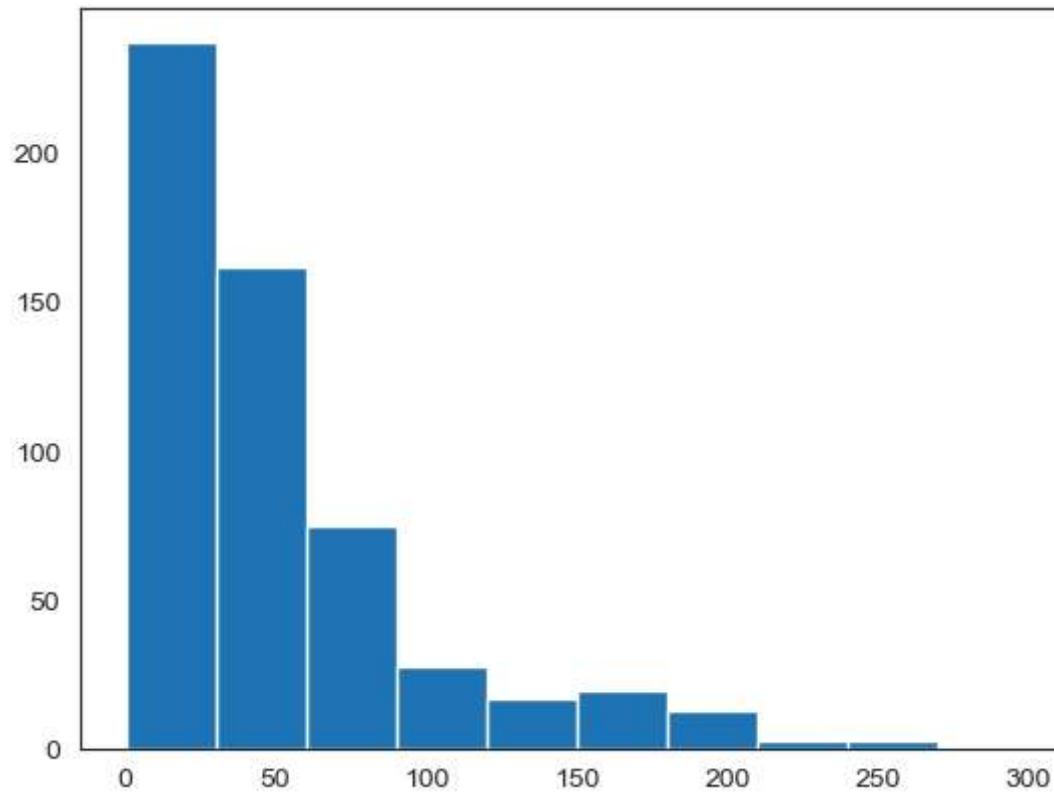
```
#uniform distribution.  
sns.set_style('darkgrid')  
m2=sns.displot(df.CriticRating,bins=15)
```



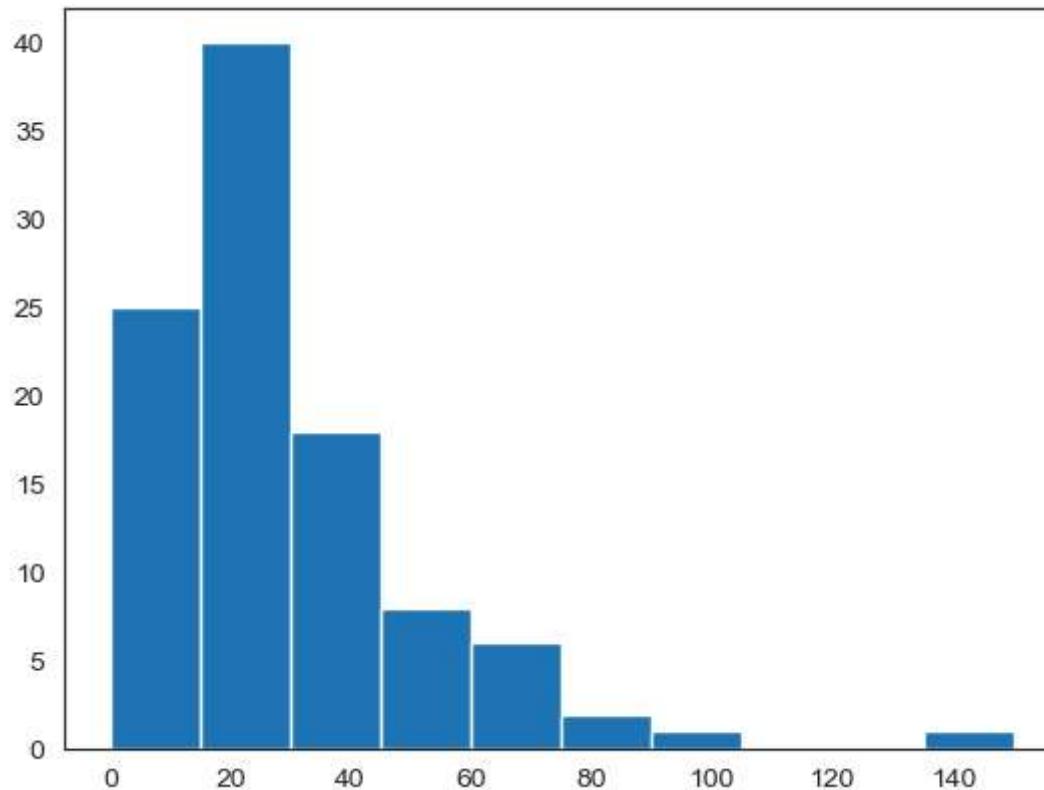
```
In [24]: sns.set_style('white')
m2=sns.displot(df.CriticRating,bins=15)
```



```
In [25]: # Creating stacked histograms & this is bit tough to understand.  
plt.hist(df.BudgetMillions)  
plt.show()
```



```
In [26]: plt.hist(df[df.Genre == 'Drama'].BudgetMillions)
plt.show()
```



```
In [27]: df.head()
```

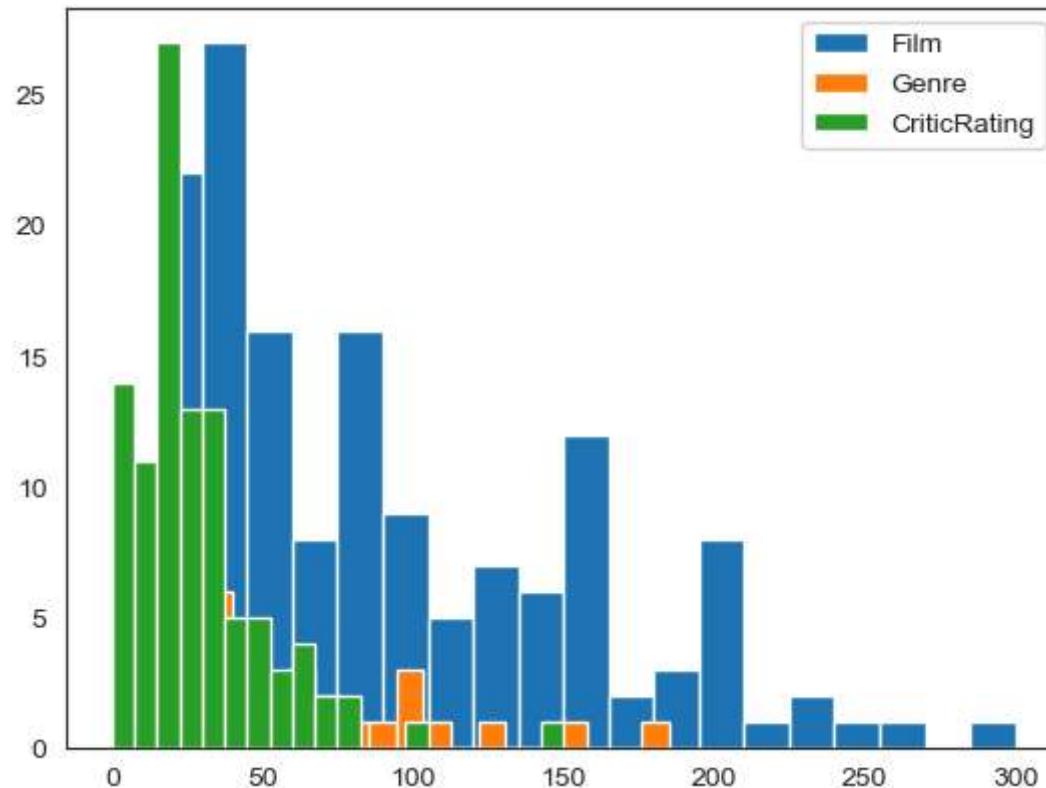
```
Out[27]:
```

	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 [28]: # Below plots are stacked histogram because overlaped
```

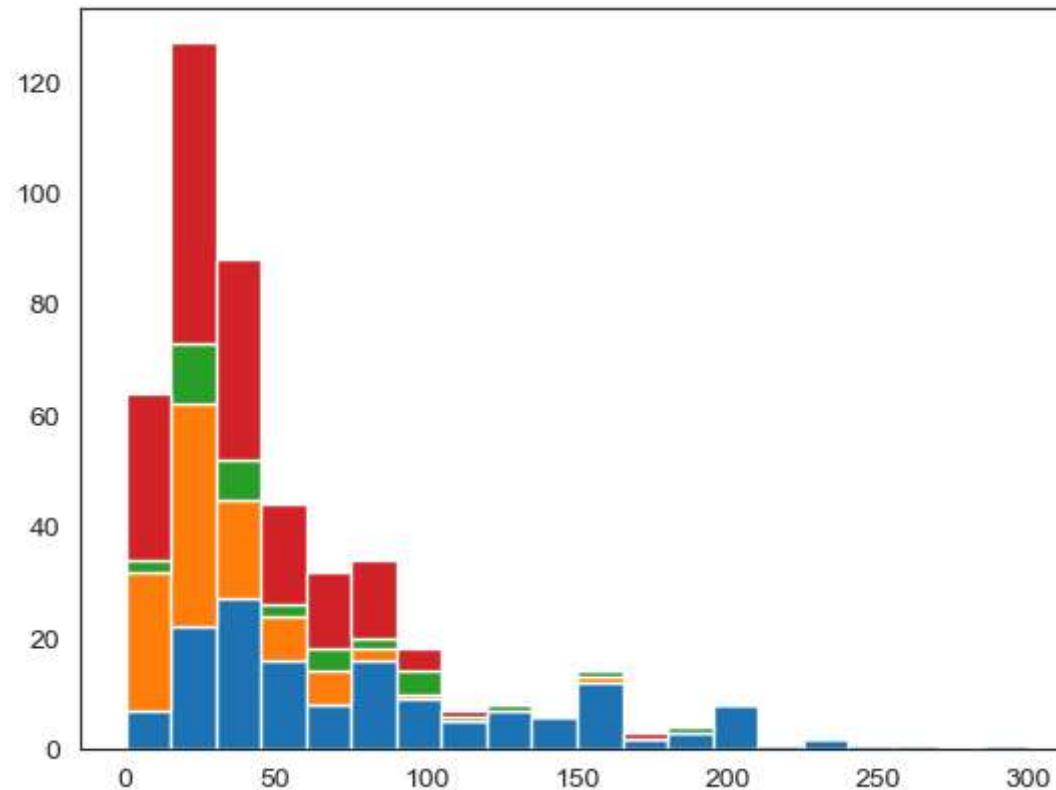
```
plt.hist(df[df.Genre == 'Action'].BudgetMillions, bins = 20)
```

```
plt.hist(df[df.Genre == 'Thriller'].BudgetMillions, bins = 20)
plt.hist(df[df.Genre == 'Drama'].BudgetMillions, bins = 20)
plt.legend(df)
plt.show()
```



```
In [29]: plt.hist([df[df.Genre == 'Action'],
              df[df.Genre == 'Drama'],
              df[df.Genre == 'Thriller'],
              df[df.Genre == 'Comedy']],
              bins = 20, stacked = True)

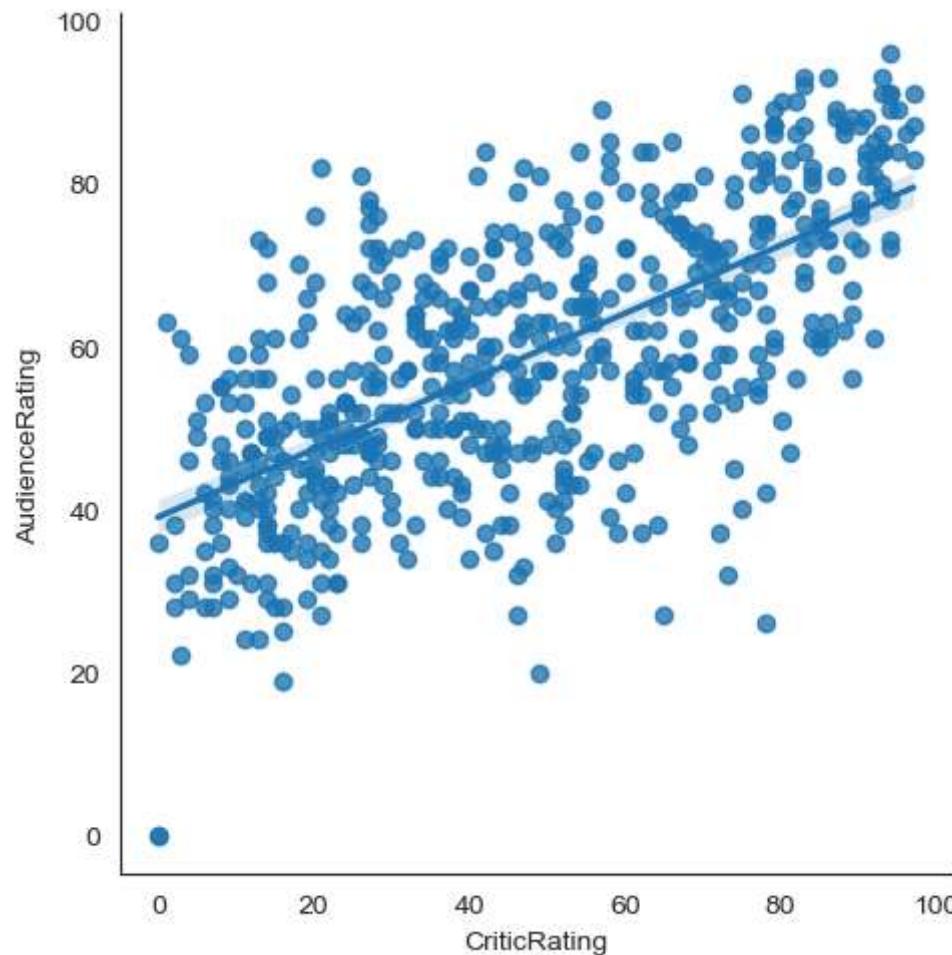
plt.show()
```



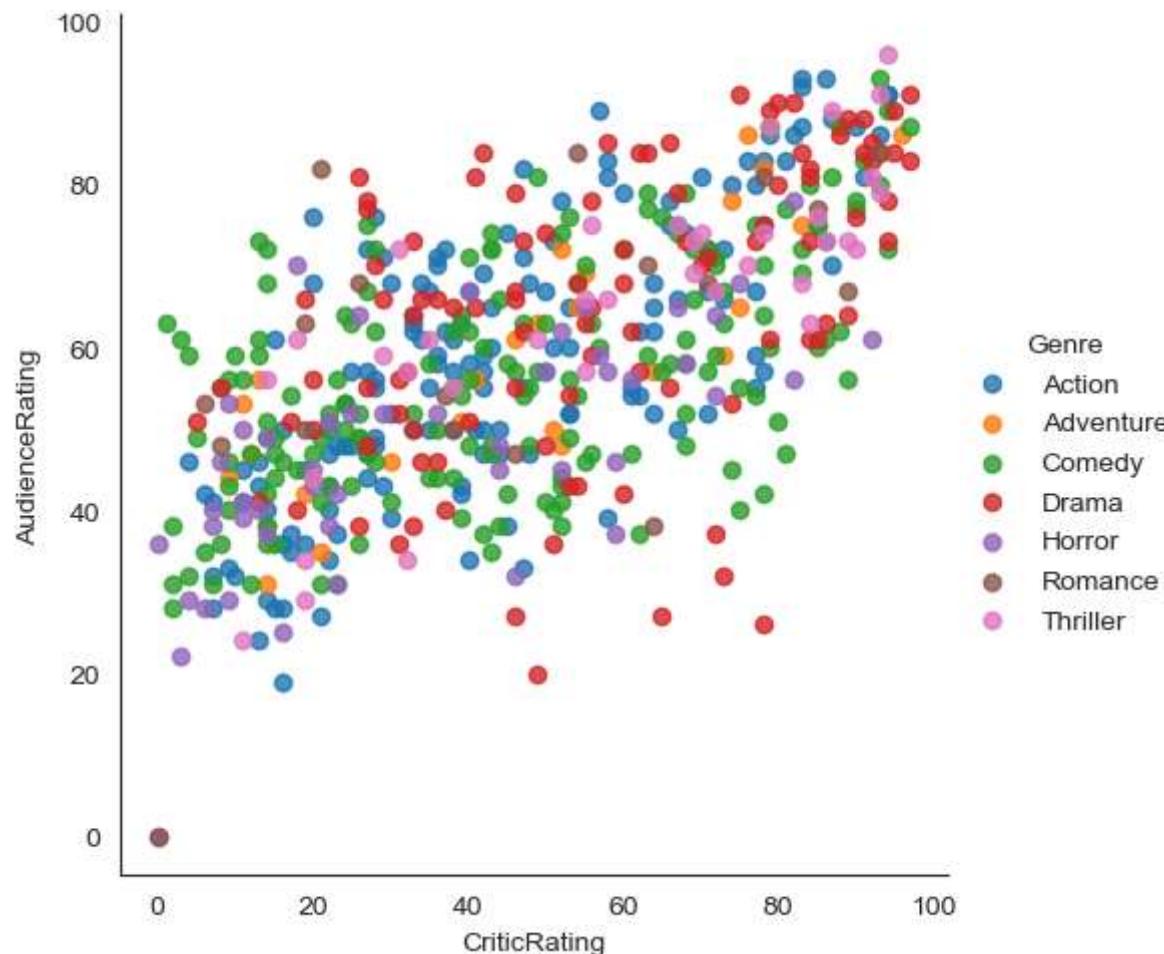
```
In [30]: # if you have 100 categories you cannot copy & paste all the things  
  
for gen in df.Genre.cat.categories:  
    print(gen)
```

Action  
Adventure  
Comedy  
Drama  
Horror  
Romance  
Thriller

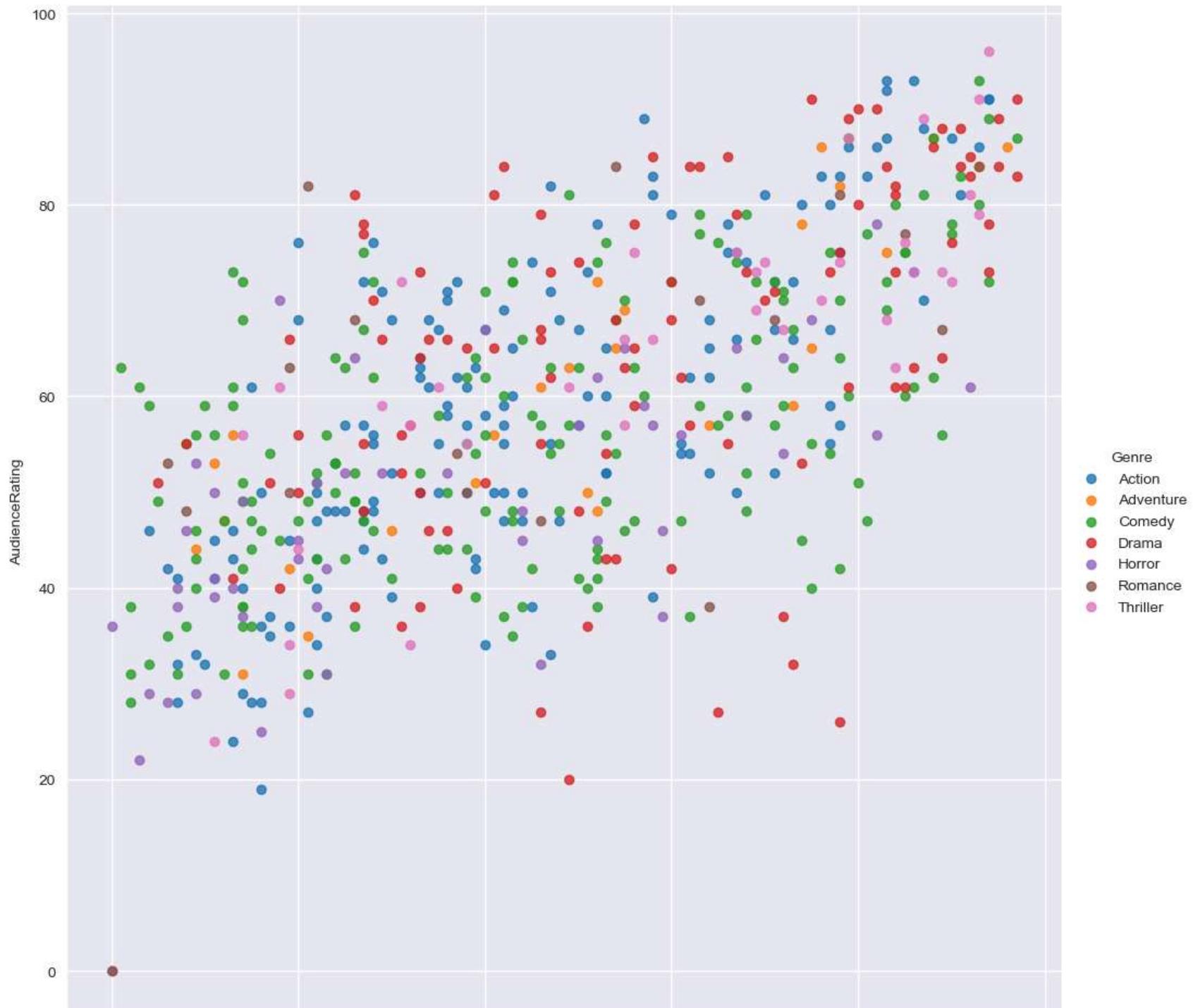
```
In [31]: j=sns.lmplot(data=df,x='CriticRating',y='AudienceRating',fit_reg=True)
```



```
In [32]: j1=sns.lmplot(data=df,x='CriticRating',y='AudienceRating',fit_reg=False,hue='Genre')
```



```
In [56]: sns.set_style('darkgrid')
j3=sns.lmplot(data=df,x='CriticRating',y='AudienceRating',\
    fit_reg=False,hue='Genre',height=10,aspect=1)
```

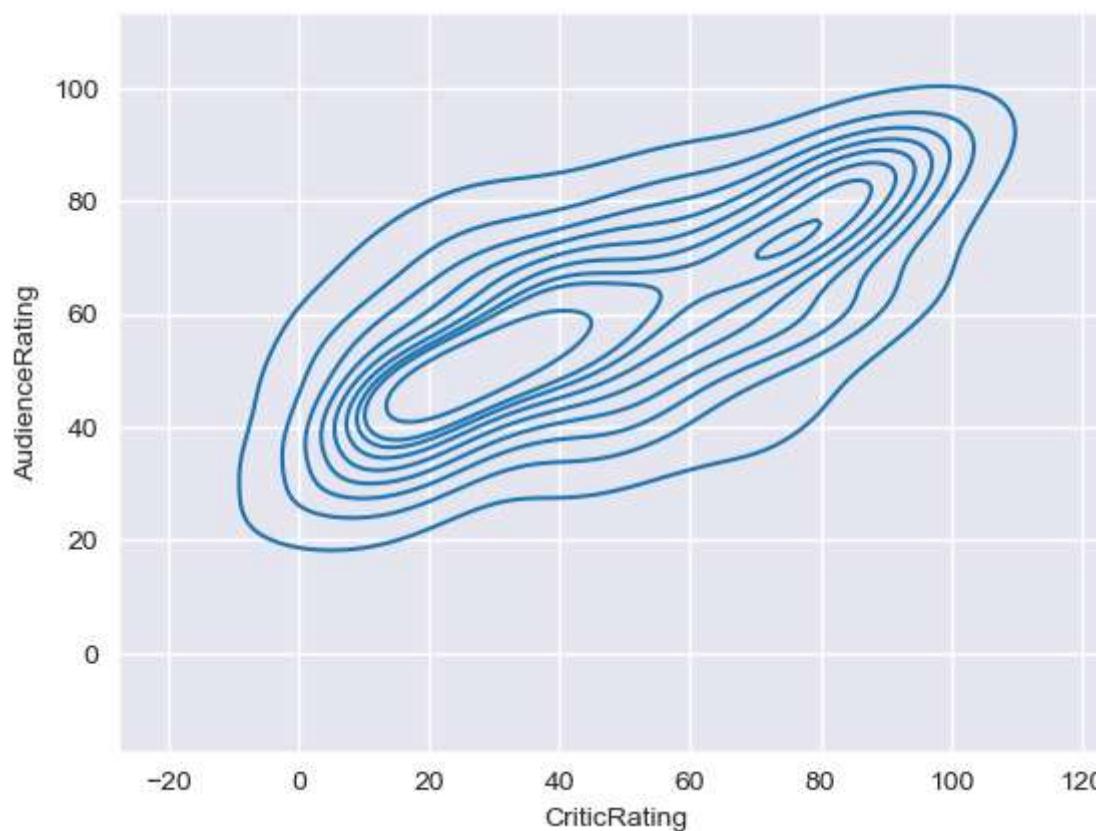




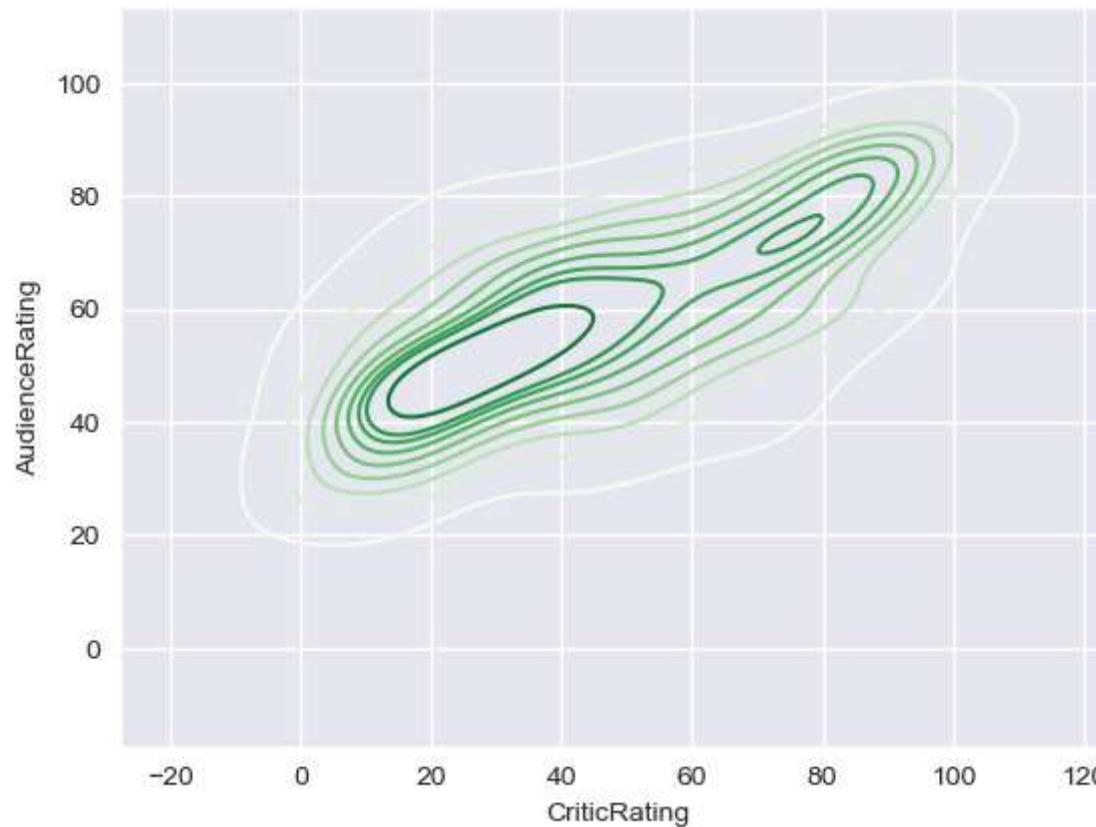
```
In [57]: # Kernal Density Estimate plot ( KDE PLOT )
# how can i visulize audience rating & critics rating . using scatterplot.
```

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

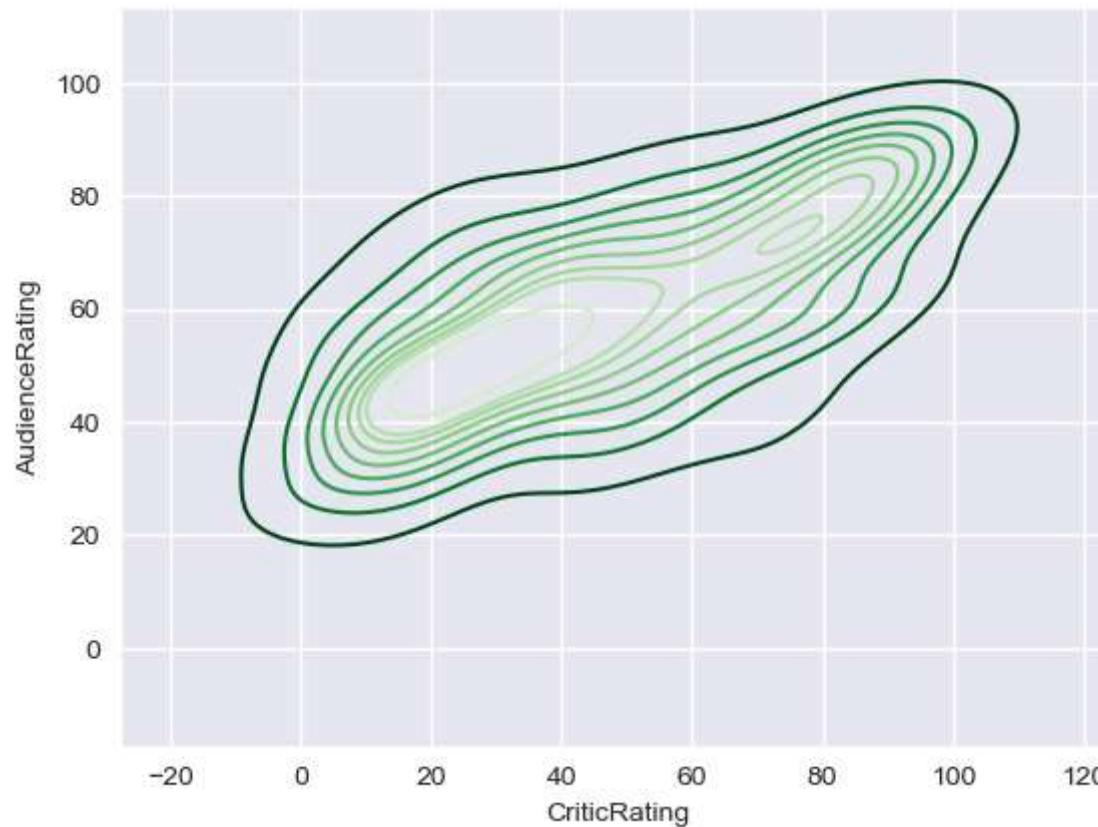
# where do u find more density and how density is distibuted across from the the chat
# center point is kernel this is calld KDE & insteaode of dots it visualize like this
# we can able to clearly see the spread at the audience ratings.
```



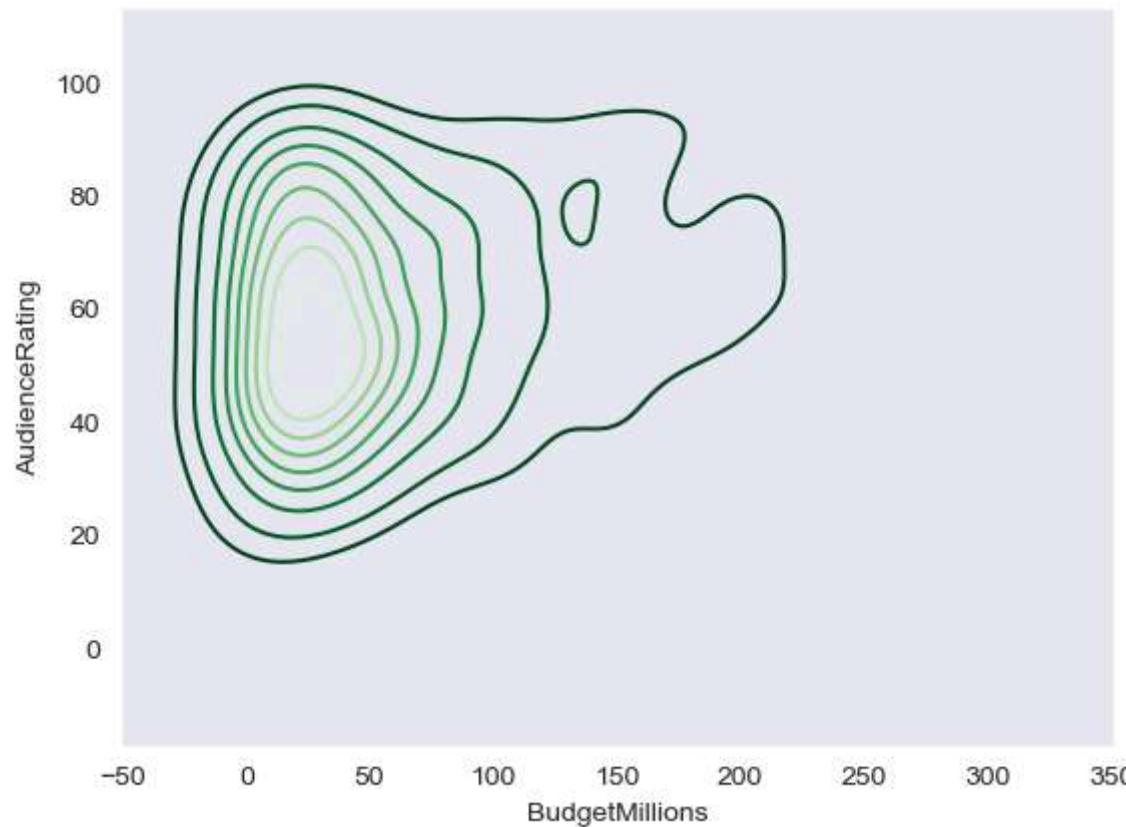
```
In [59]: k1 = sns.kdeplot(x=df.CriticRating,y=df.AudienceRating,
                      shade = False,shade_lowest=False,cmap='Greens')
```



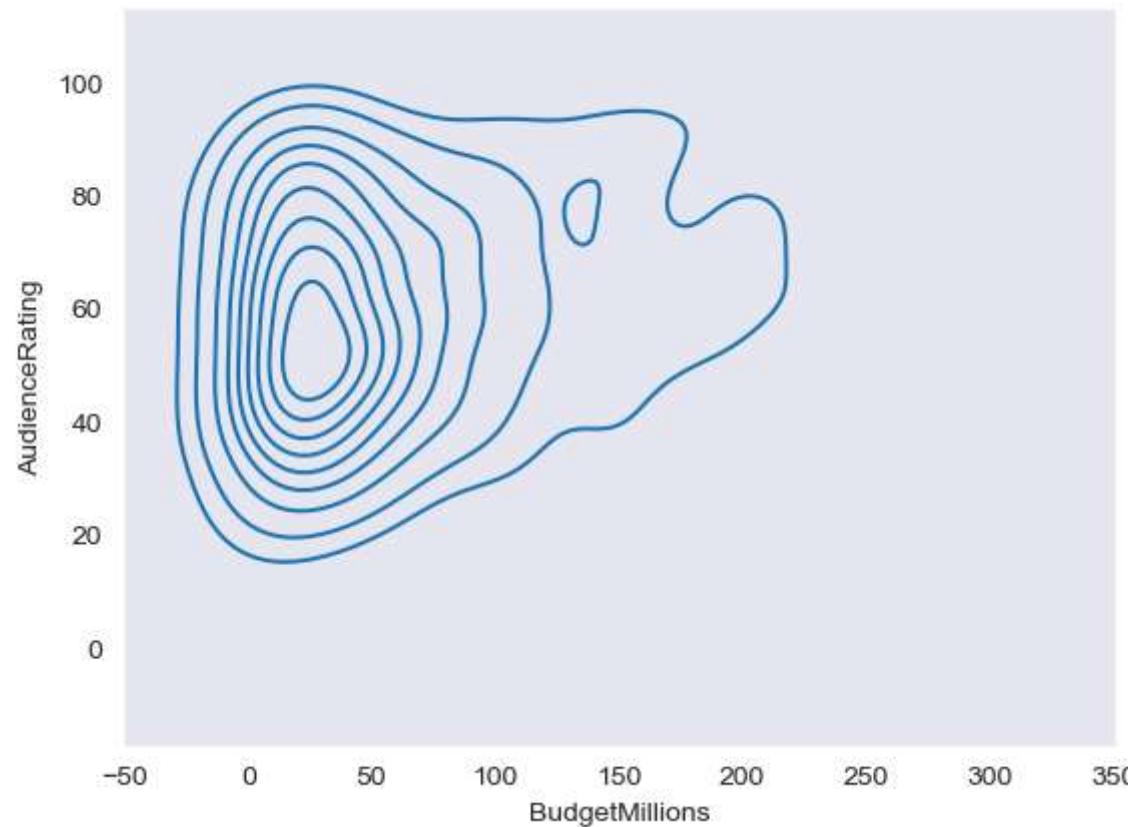
```
In [60]: k1 = sns.kdeplot(x=df.CriticRating,y=df.AudienceRating,\n                      shade = False,shade_lowest=False,cmap='Greens_r')
```



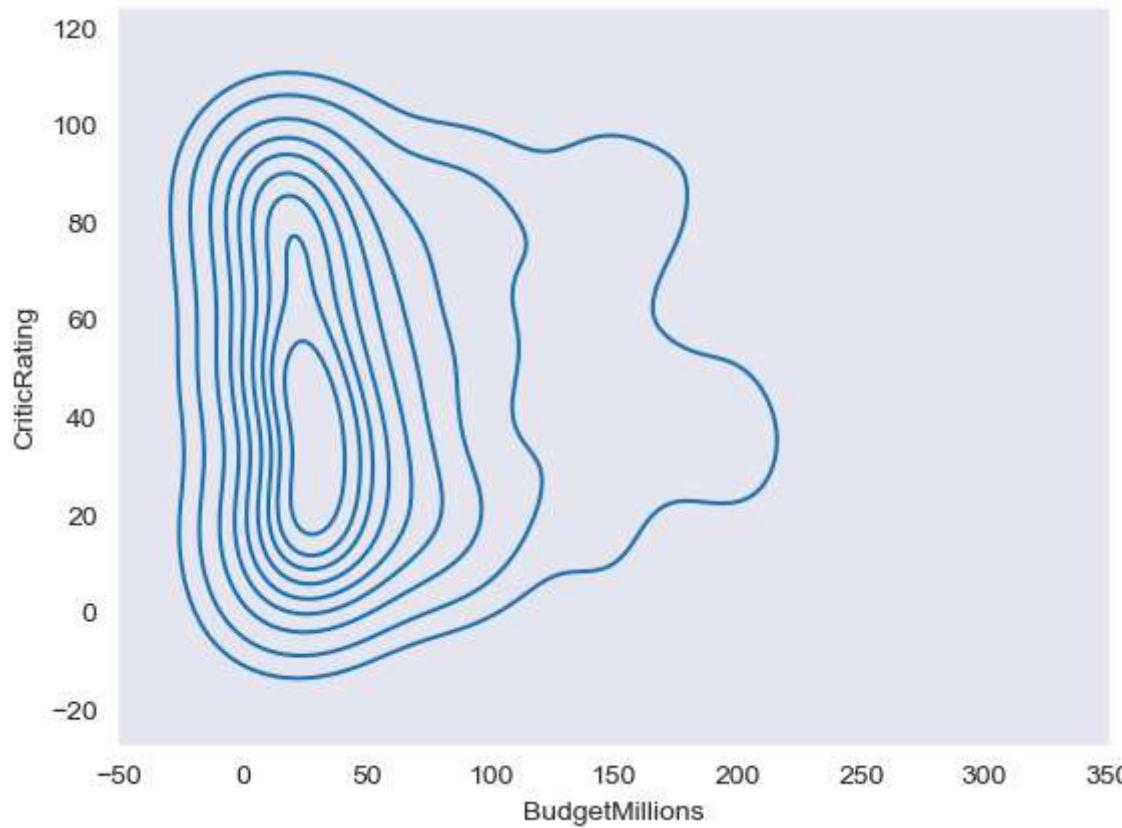
```
In [61]: sns.set_style('dark')
k1 = sns.kdeplot(x=df.BudgetMillions,y=df.AudienceRating,shade_lowest=False,cmap='Greens_r')
```



```
In [62]: sns.set_style('dark')
k1 = sns.kdeplot(x=df.BudgetMillions,y=df.AudienceRating)
```



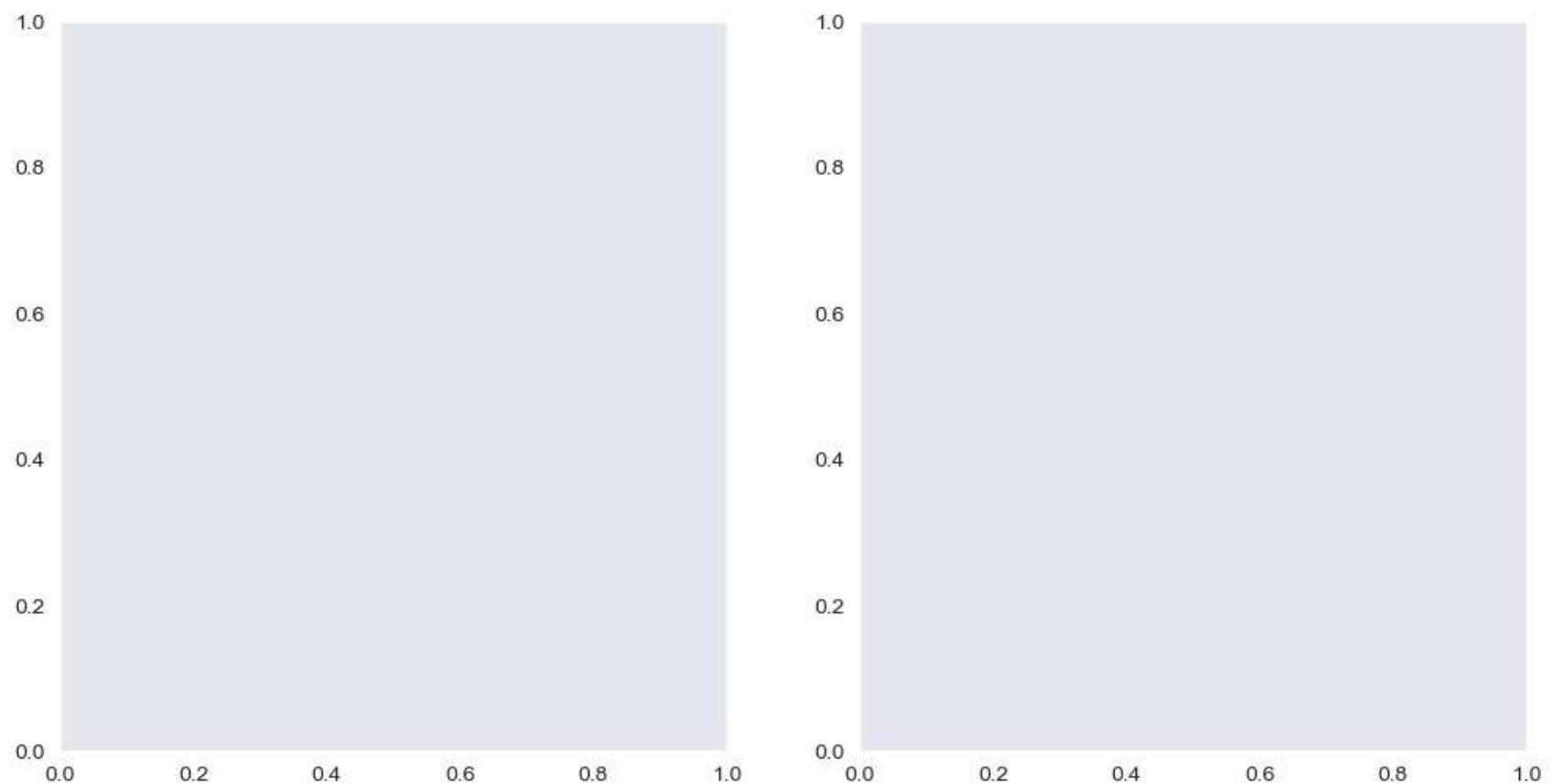
```
In [63]: k1 = sns.kdeplot(x=df.BudgetMillions,y=df.CriticRating)
```



## subplots

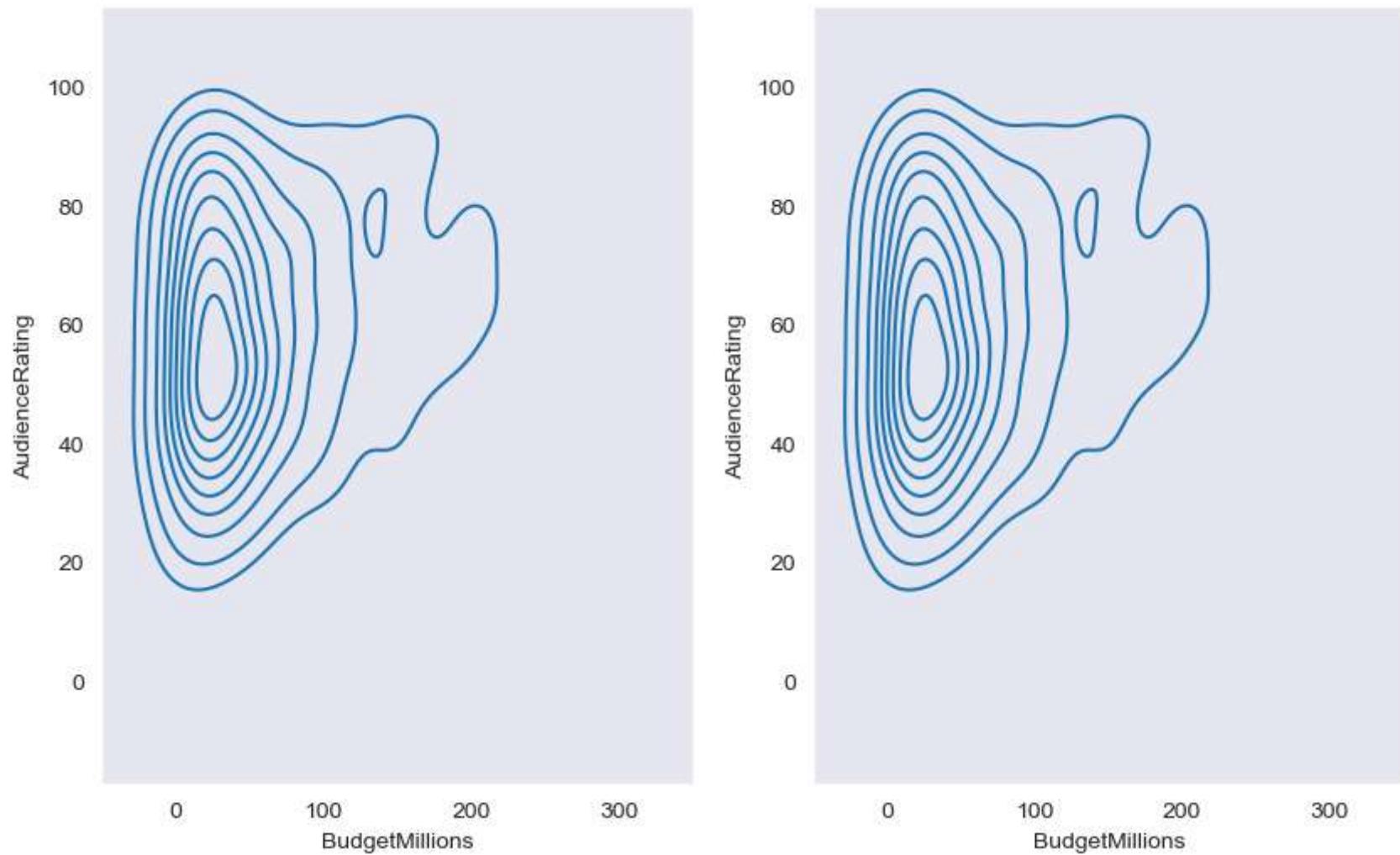
```
In [64]: #subPlots
```

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



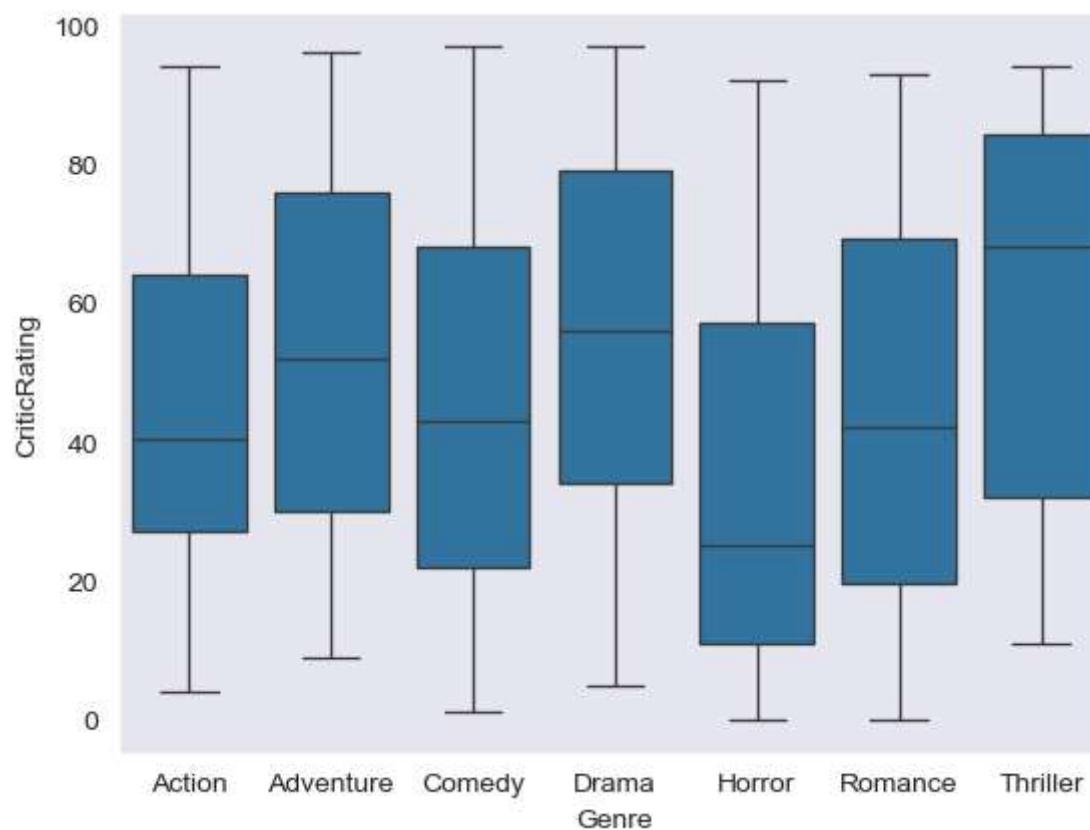
```
In [65]: f, axes = plt.subplots(1,2, figsize =(10,6))

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

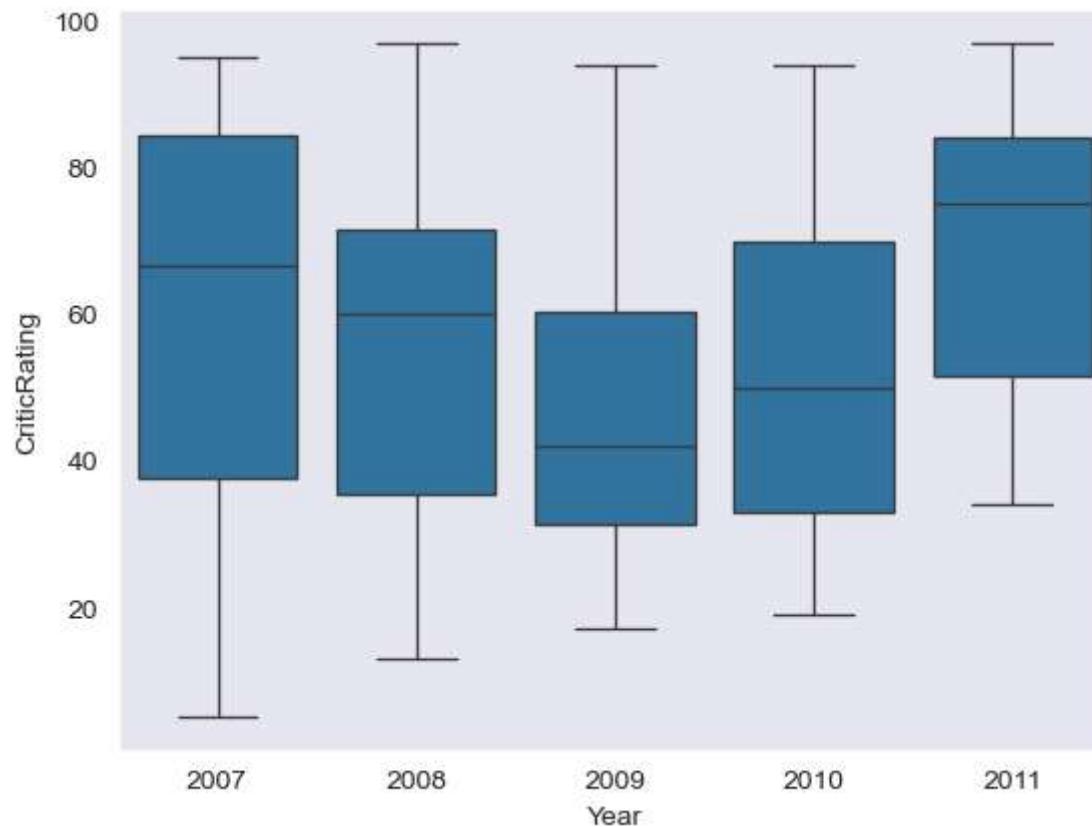


## Box plots

```
In [66]: #Box plots -  
b = sns.boxplot(data=df, x='Genre', y = 'CriticRating')
```

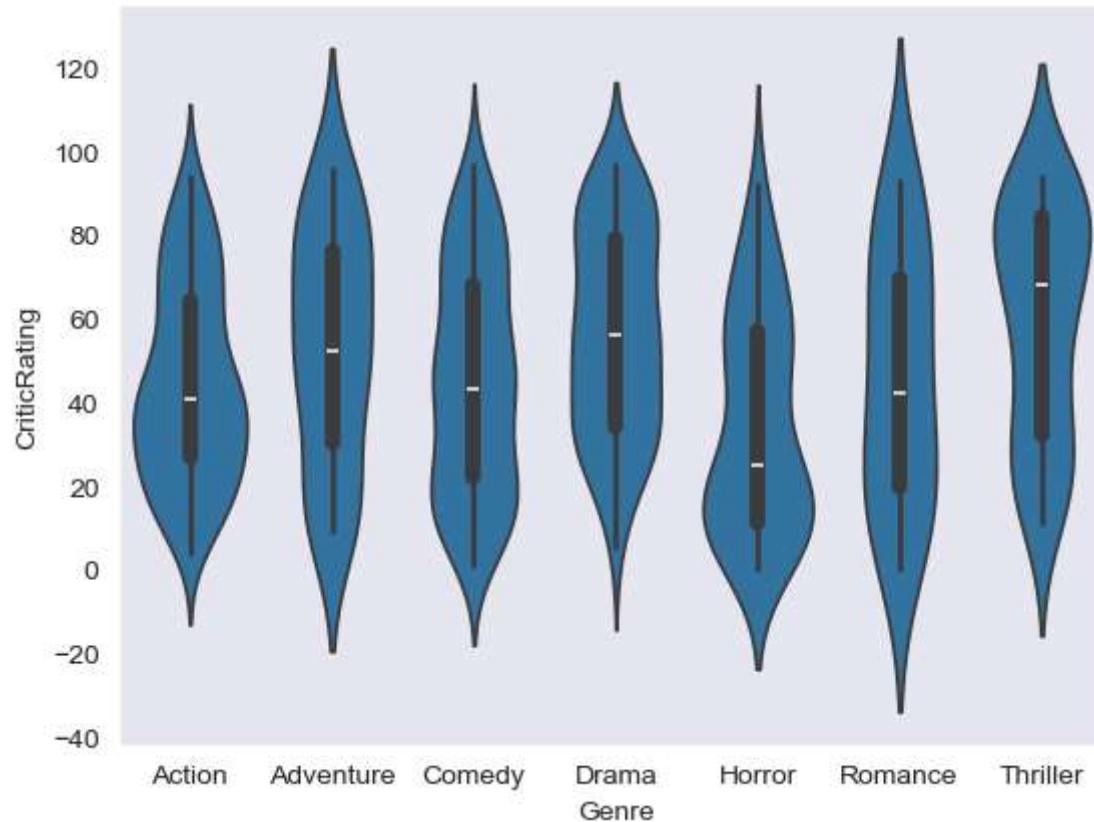


```
In [67]: b1 = sns.boxplot(data=df[df.Genre == 'Drama'], x='Year', y = 'CriticRating')
```

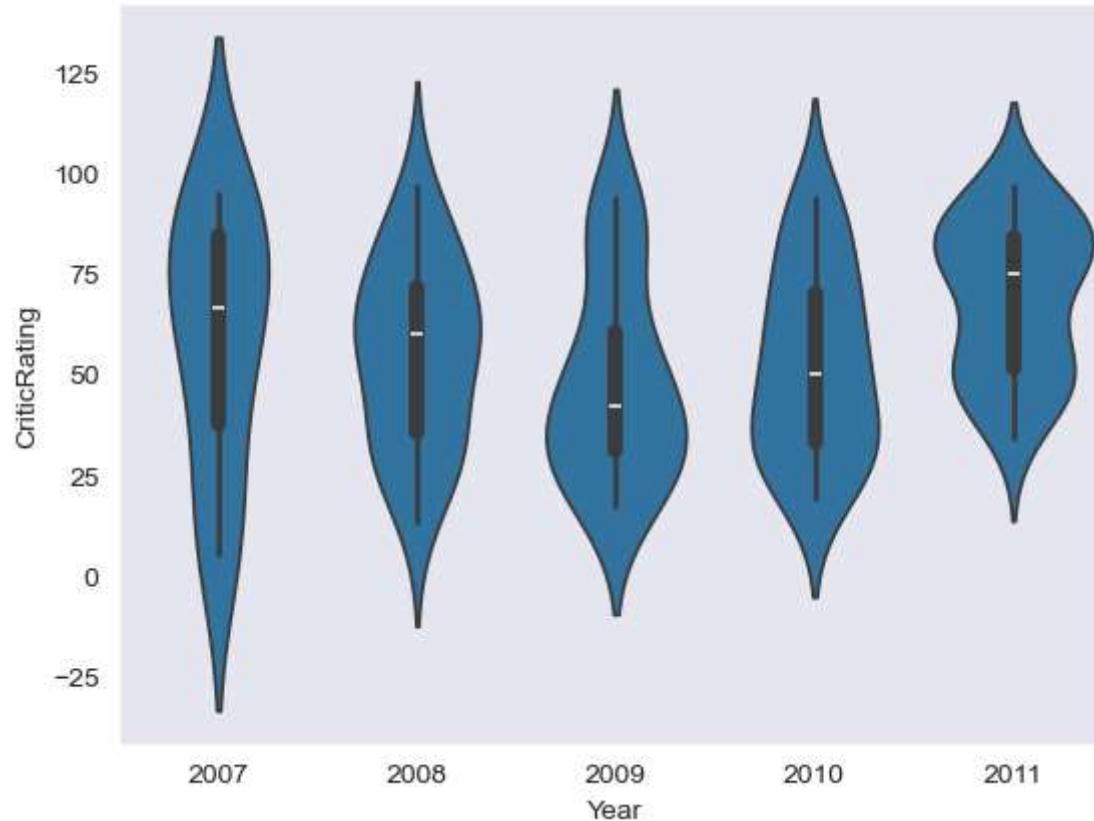


## violin plot

```
In [68]: #violin plot  
v = sns.violinplot(data=df, x='Genre', y = 'CriticRating')
```



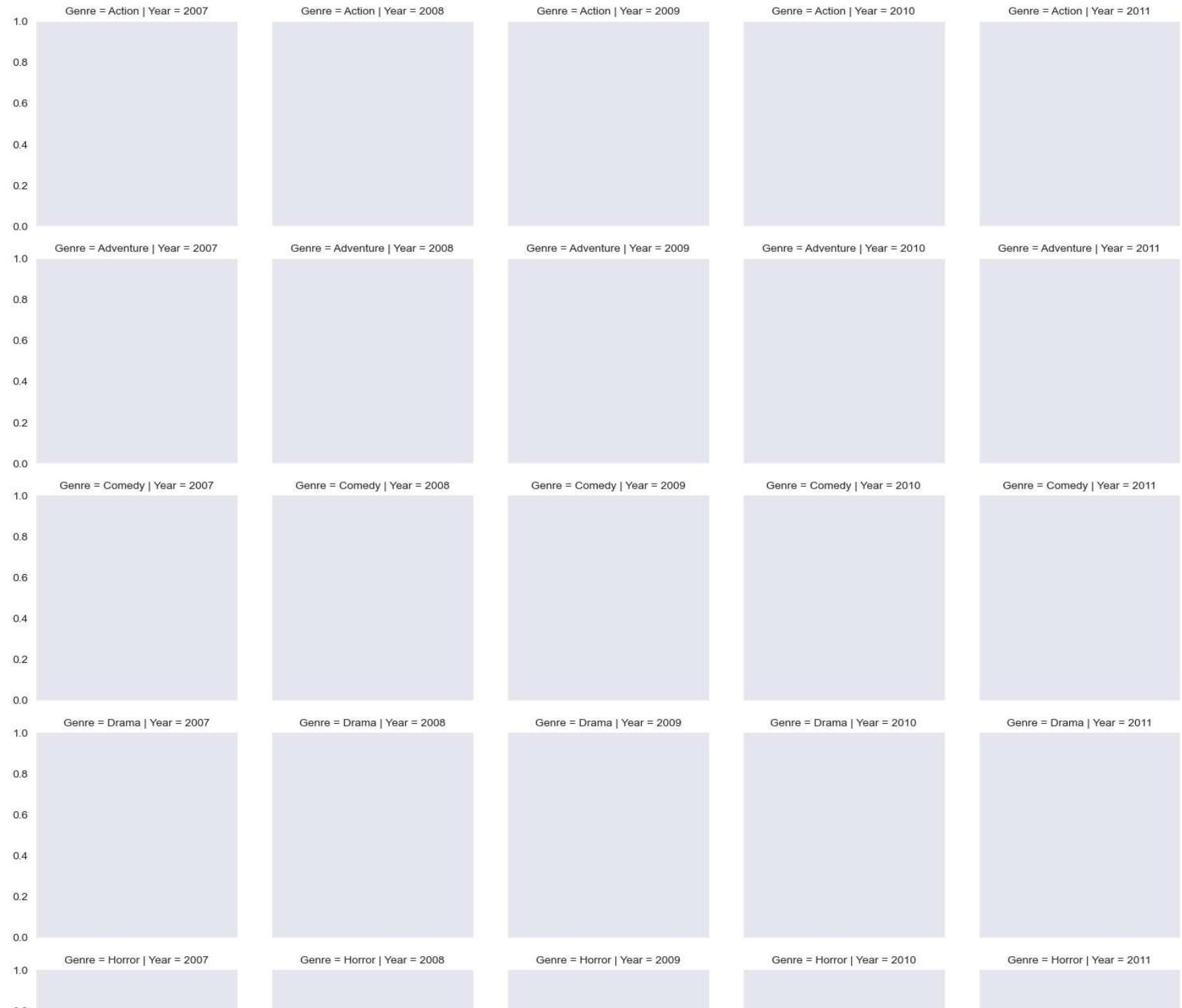
```
In [69]: v1= sns.violinplot(data=df[df.Genre == 'Drama'], x='Year', y = 'CriticRating')
```



## Createing a Facet grid

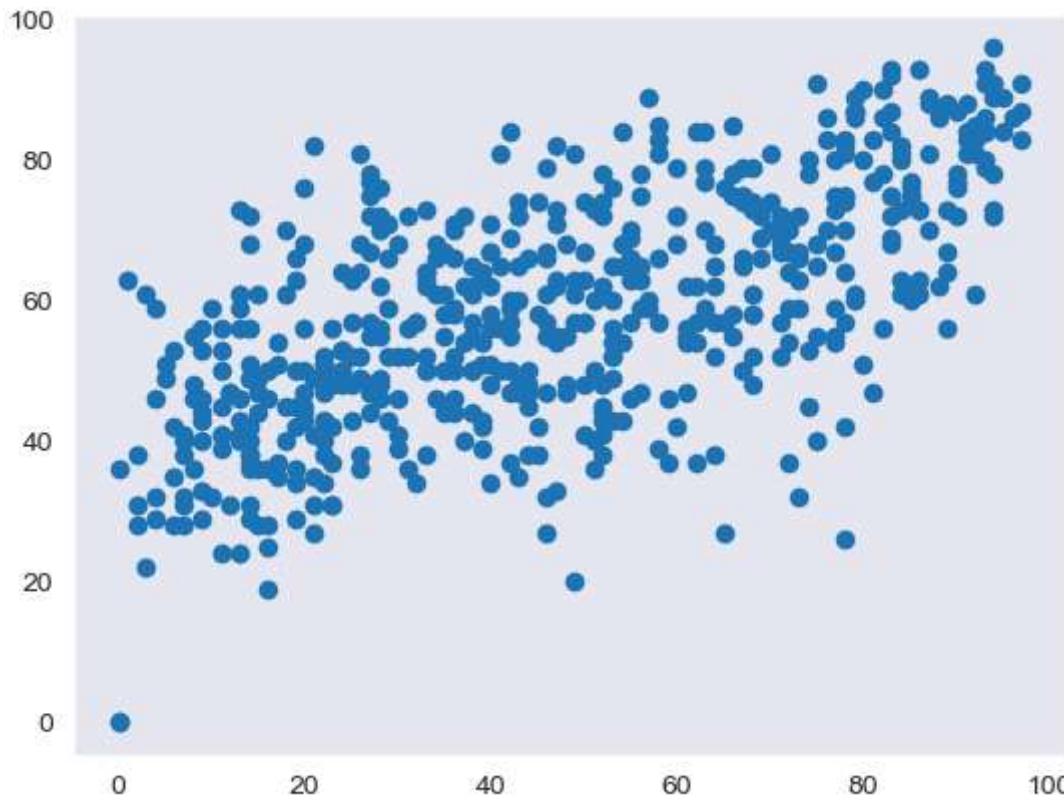
```
In [70]: f =sns.FacetGrid (df, row = 'Genre', col = 'Year', hue = 'Genre') #kind of subplots
```

## IMDB(Kaggle Dataset) Movie Ratings Advanced Analysis with Seaborn



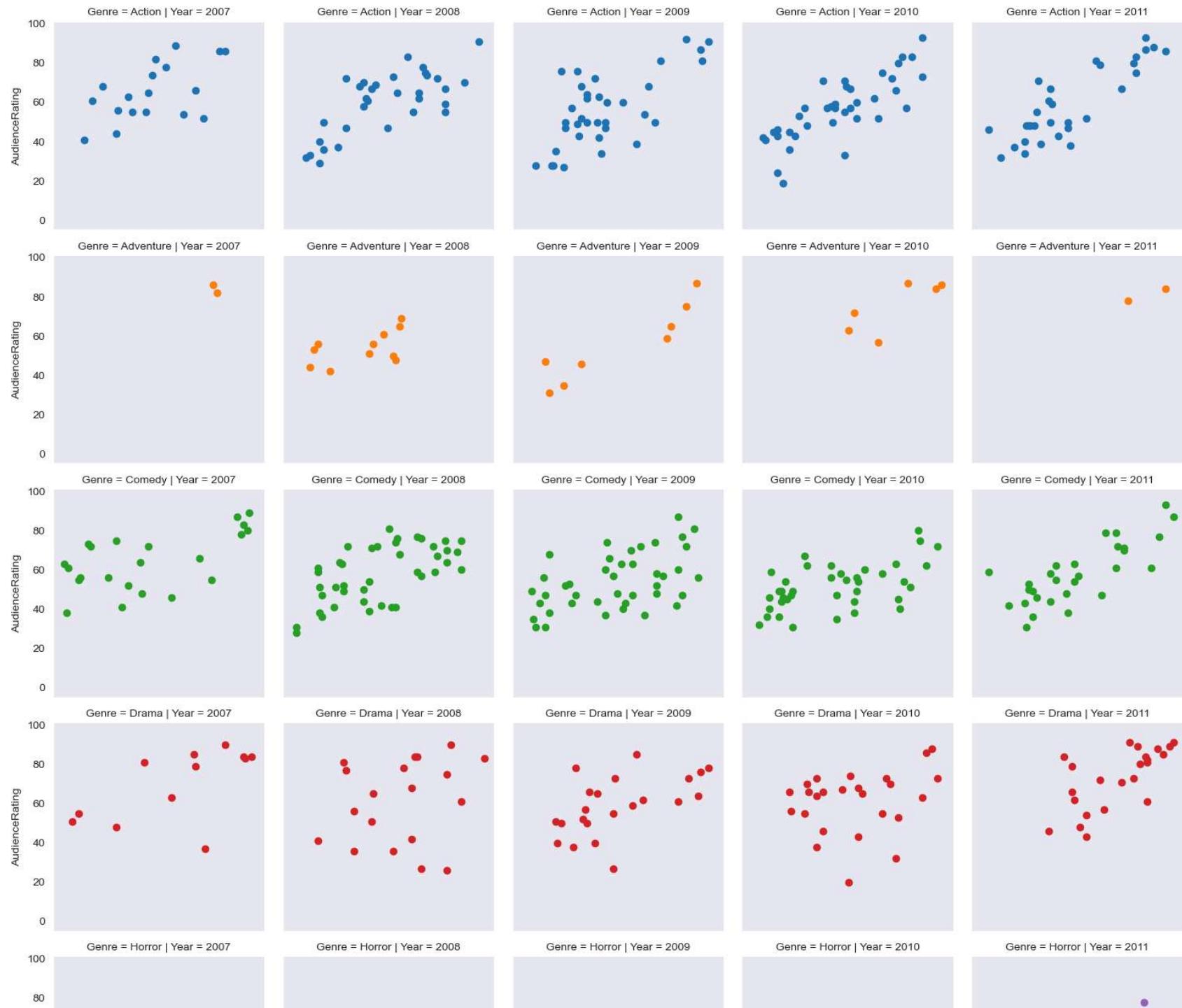


```
In [71]: plt.scatter(df.CriticRating,df.AudienceRating)
plt.show()
```

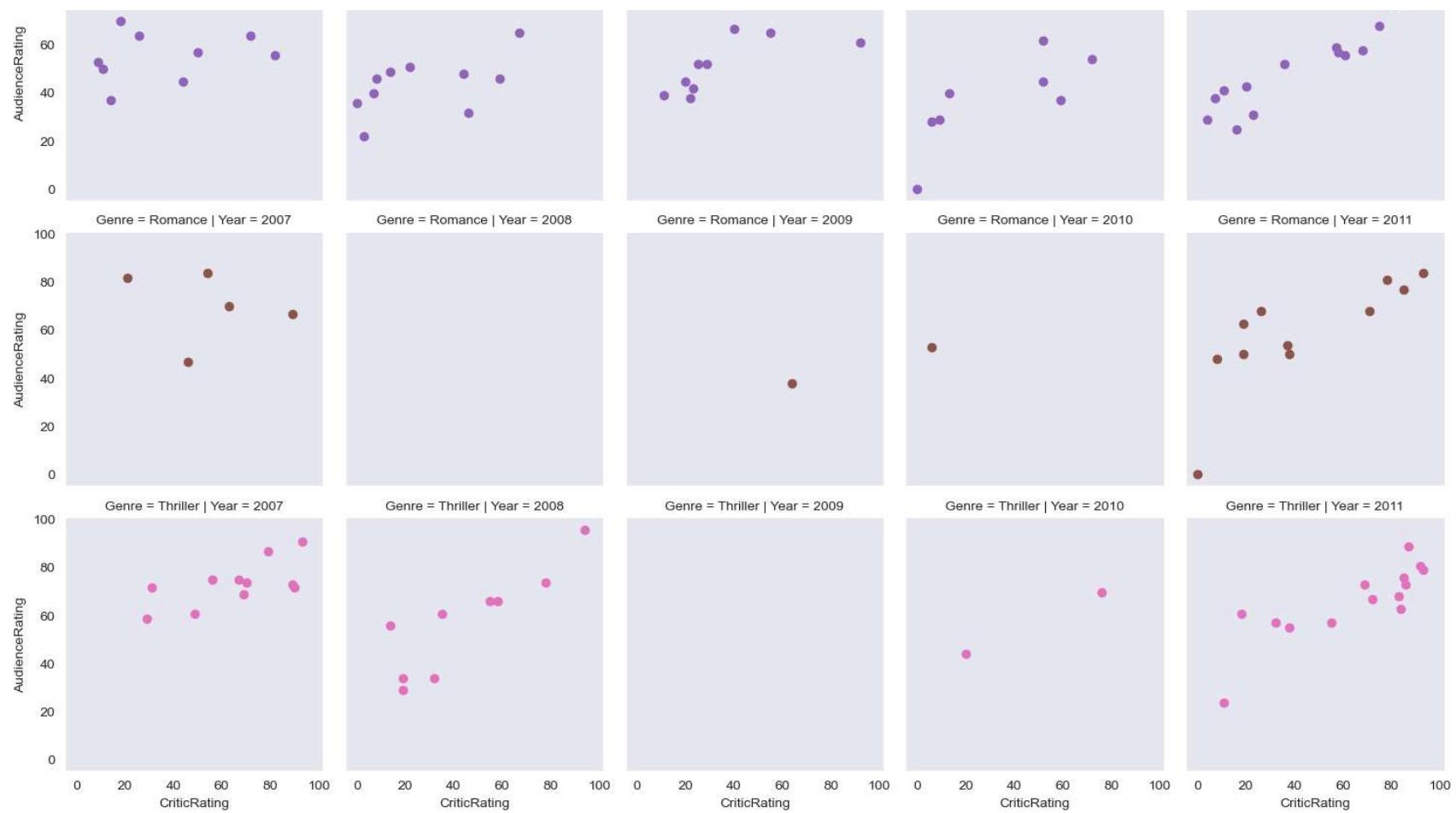


```
In [72]: #scatterplots are mapped in facetgrid
f=sns.FacetGrid (df, row = 'Genre', col = 'Year', hue = 'Genre')
f=f.map(plt.scatter, 'CriticRating', 'AudienceRating' )
```

## IMDB(Kaggle Dataset) Movie Ratings Advanced Analysis with Seaborn



## IMDB(Kaggle Dataset) Movie Ratings Advanced Analysis with Seaborn

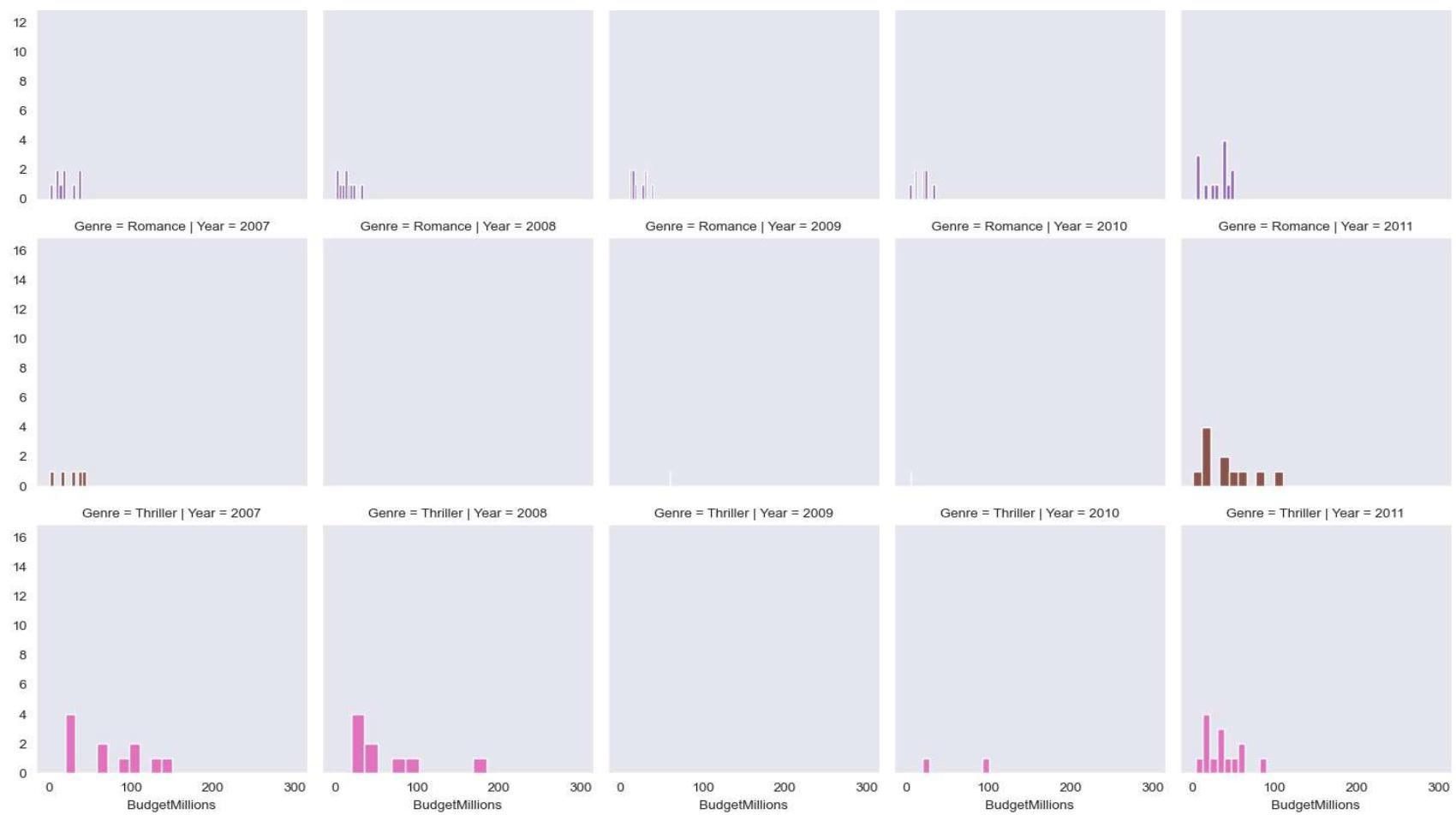


```
In [73]: # you can populated any type of chat.
```

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

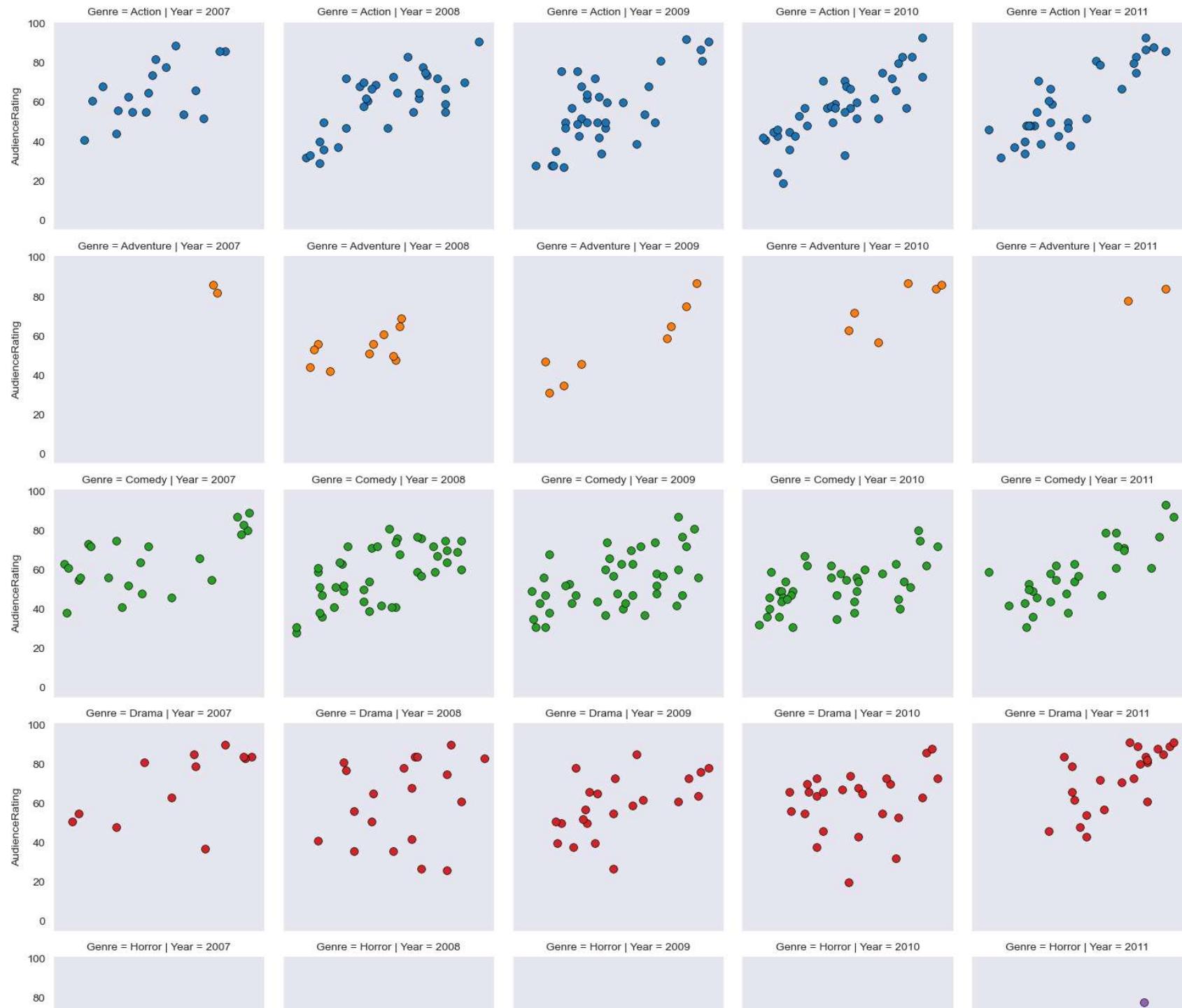
## IMDB(Kaggle Dataset) Movie Ratings Advanced Analysis with Seaborn



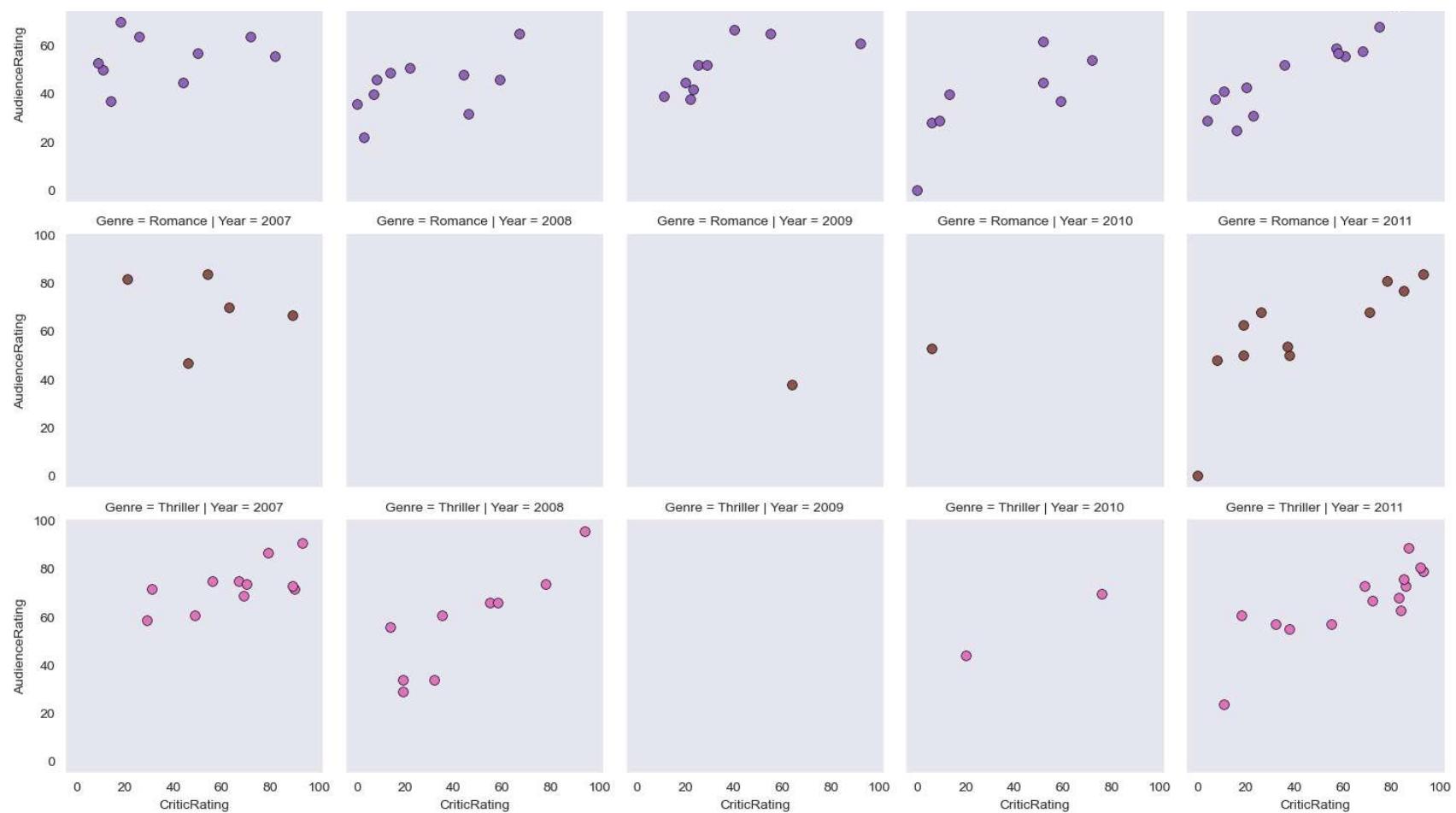


```
In [74]: g =sns.FacetGrid (df, row = 'Genre', col = 'Year', hue = 'Genre')
kws = dict(s=50, linewidth=0.5,edgecolor='black')
g = g.map(plt.scatter, 'CriticRating', 'AudienceRating',**kws ) #scatterplots are mapped in facetgrid
```

## IMDB(Kaggle Dataset) Movie Ratings Advanced Analysis with Seaborn



## IMDB(Kaggle Dataset) Movie Ratings Advanced Analysis with Seaborn



```
In [75]: # python is not vectorize programming Language
# Building dashboards (dashboard - combination of chats)

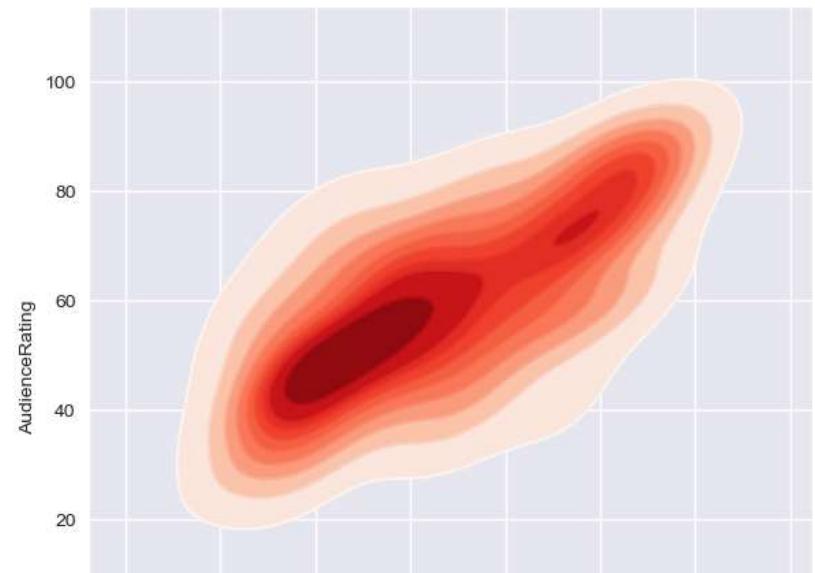
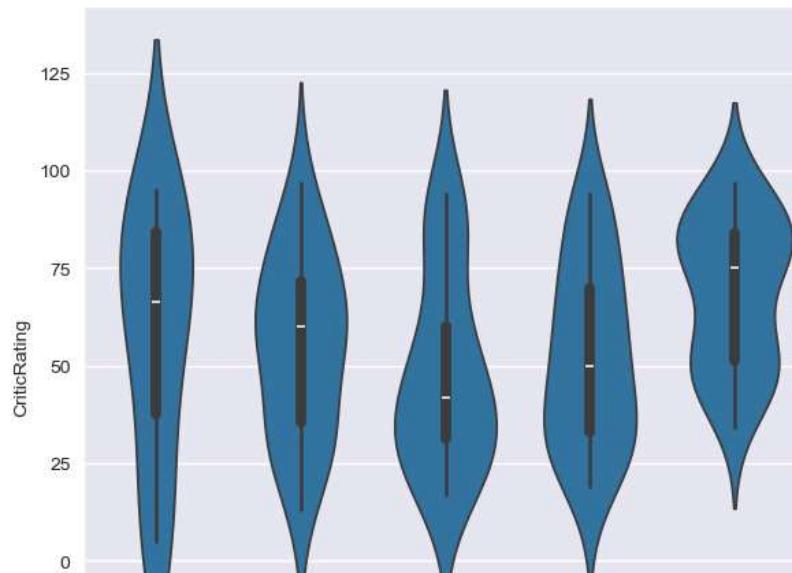
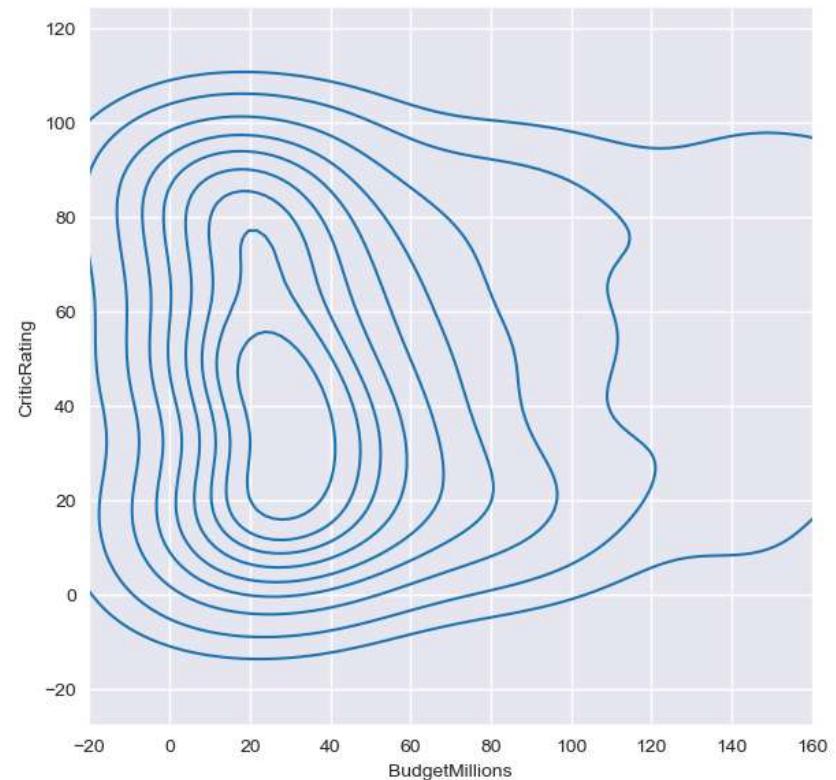
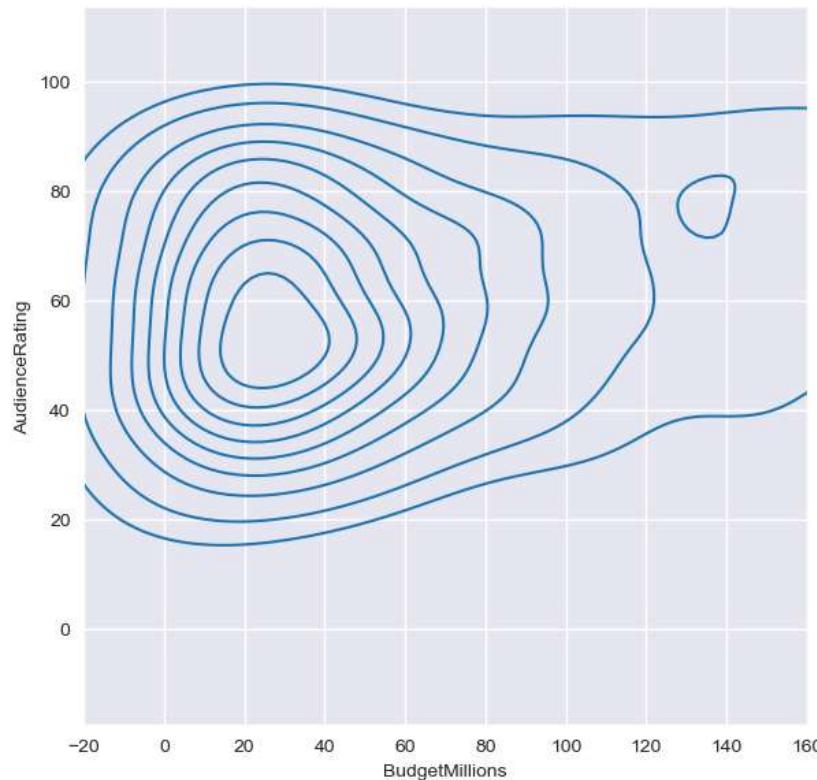
sns.set_style('darkgrid')
f, axes = plt.subplots(2,2, figsize = (15,15))

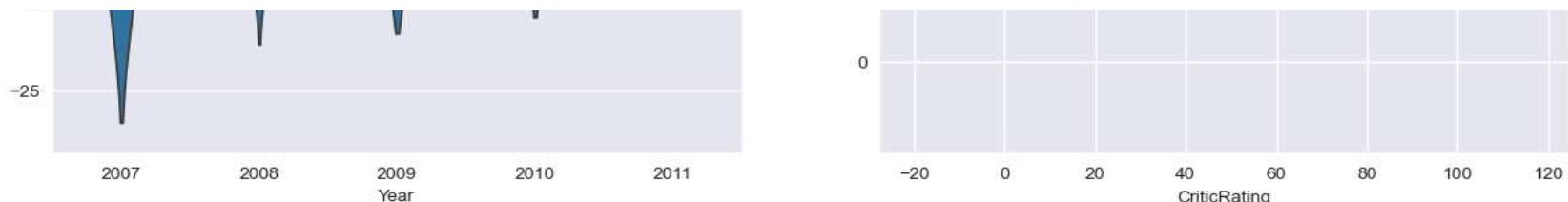
k1 = sns.kdeplot(x=df.BudgetMillions,y=df.AudienceRating,ax=axes[0,0])
k2 = sns.kdeplot(x=df.BudgetMillions,y=df.CriticRating,ax = axes[0,1])

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

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

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





```
In [77]: # 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))

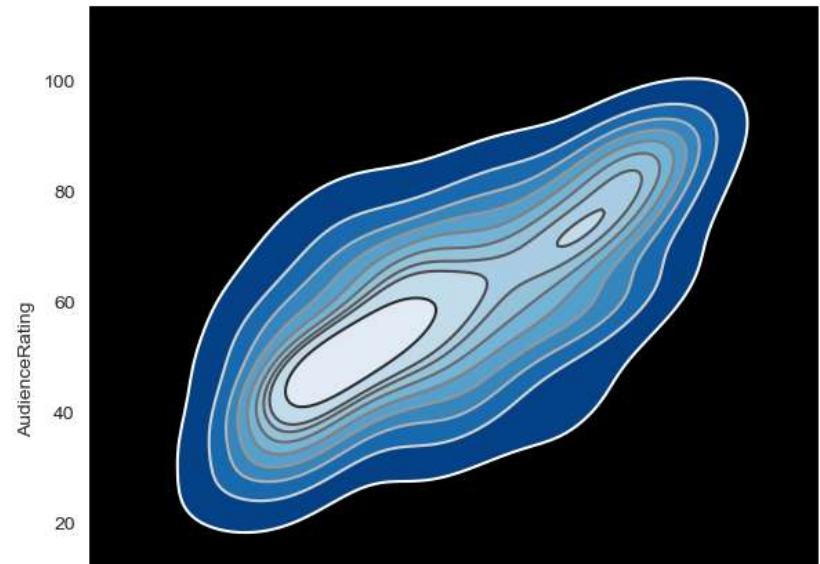
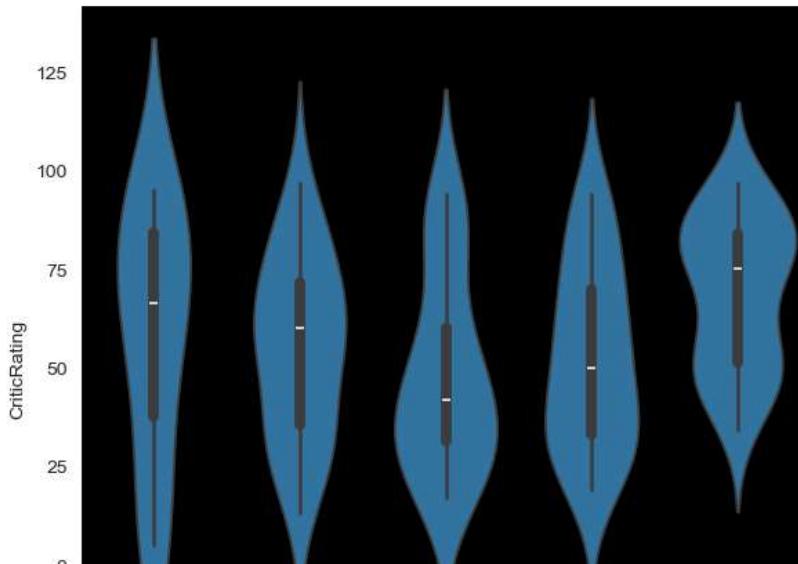
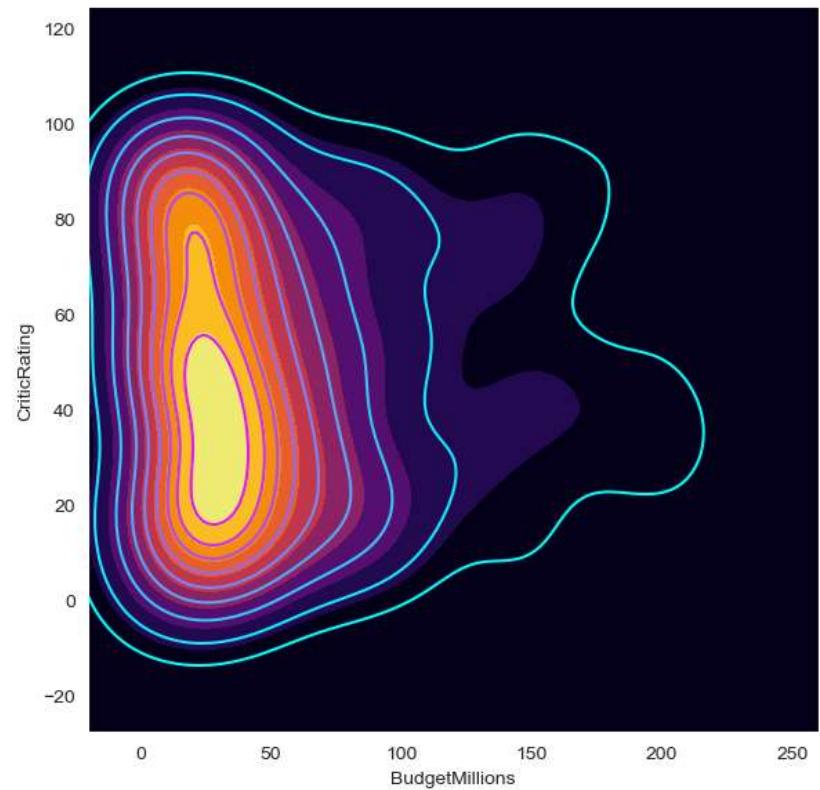
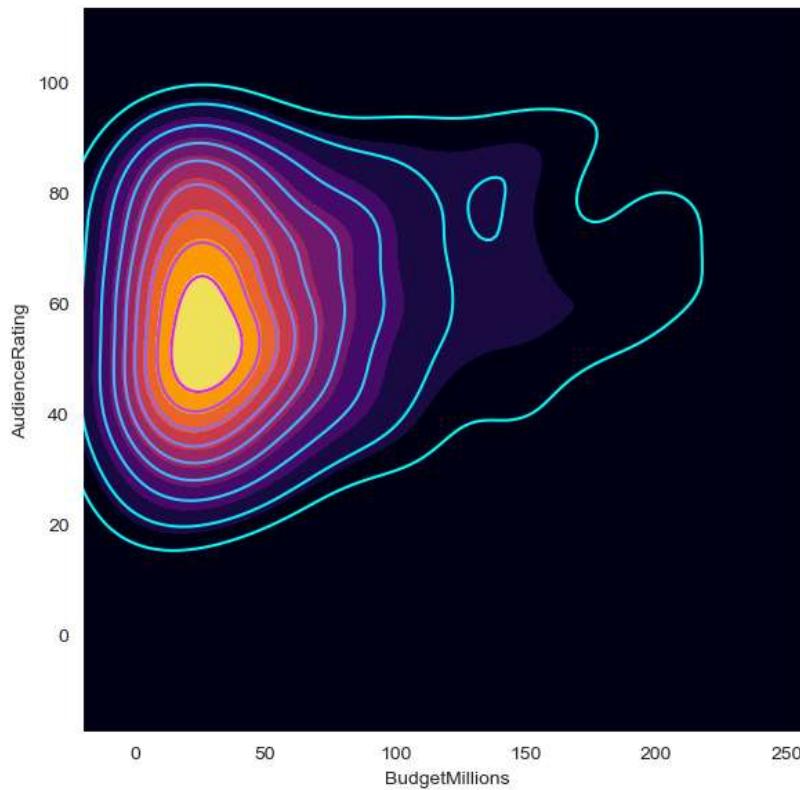
#plot [0,0]
k1 = sns.kdeplot(x=df.BudgetMillions,y=df.AudienceRating, \
                   shade = True, shade_lowest=True,cmap = 'inferno', \
                   ax = axes[0,0])
k1b = sns.kdeplot(x=df.BudgetMillions,y=df.AudienceRating, \
                   cmap = 'cool',ax = axes[0,0])

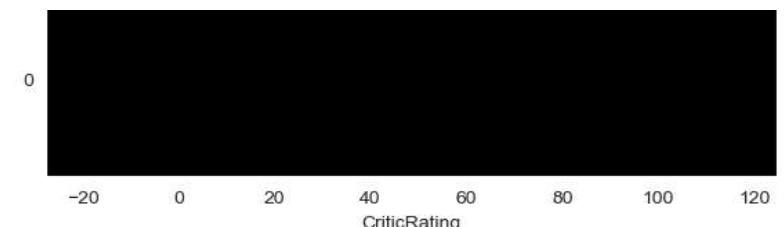
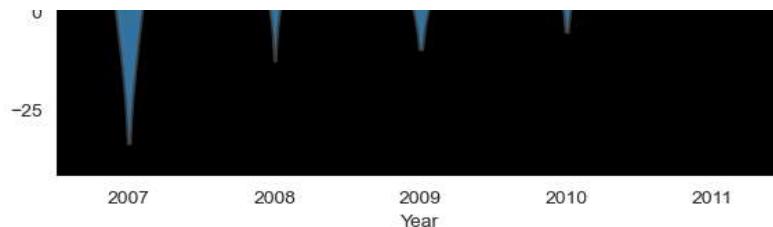
#plot [0,1]
k2 = sns.kdeplot(x=df.BudgetMillions,y=df.CriticRating, \
                   shade=True, shade_lowest=True, cmap='inferno', \
                   ax = axes[0,1])
k2b = sns.kdeplot(x=df.BudgetMillions,y=df.CriticRating, \
                   cmap = 'cool', ax = axes[0,1])

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

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

```
k1.set(xlim=(-20,260))  
k2.set(xlim=(-20,260))  
  
plt.show()
```





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> Facet grid 9> Building dashboards.

## EDA is completed.

In [ ]:

In [ ]: