

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

```
In [2]: os.getcwd()
```

```
Out[2]: 'C:\\Users\\soham'
```

```
In [3]: moive = pd.read_csv(r'C:\Users\soham\OneDrive\Desktop\Movie-Rating.csv')
```

```
In [4]: moive
```

```
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(moive)
```

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

```
In [6]: moive.head()
```

Out[6]:

	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 [7]: `moive.tail()`

Out[7]:

	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]: `moive.columns`

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

In [17]: `moive.head()`

Out[17]:

	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 [19]: `moive.columns=['Film','Genre','CriticRatinig','AudioRating','BudgetMillions','Ye`In [21]: `moive`

Out[21]:

	Film	Genre	CriticRatinig	AudioRating	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 [23]: `moive.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   CriticRatinig   559 non-null   int64
3   AudioRating     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 [25]: `moive.describe()`

```

# if you look at the year the data type is int but when you look at the mean val
# we have to change to category type
# also from object datatype we will convert to category datatypes

```

Out[25]:

	CriticRatinig	AudioRating	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 [27]:

moive['Film']

Out[27]:

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

In [29]:

moive.Film

Out[29]:

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

In [31]:

moive.Film=moive.Film.astype('category')

In [33]:

moive.Film

```
Out[33]: 0      (500) Days of Summer
         1      10,000 B.C.
         2      12 Rounds
         3      127 Hours
         4      17 Again
         ...
         554     Your Highness
         555     Youth in Revolt
         556     Zodiac
         557     Zombieland
         558     Zookeeper
Name: Film, Length: 559, dtype: category
Categories (559, object): ['(500) Days of Summer ', '10,000 B.C.', '12 Rounds ', '127 Hours', ..., 'Youth in Revolt', 'Zodiac', 'Zombieland ', 'Zookeeper']
```

```
In [35]: moive.head()
```

```
Out[35]:
```

	Film	Genre	CriticRatinig	AudioRating	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 [37]: moive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Film            559 non-null   category
1   Genre           559 non-null   object
2   CriticRatinig   559 non-null   int64
3   AudioRating     559 non-null   int64
4   BudgetMillions  559 non-null   int64
5   Year            559 non-null   int64
dtypes: category(1), int64(4), object(1)
memory usage: 43.6+ KB
```

```
In [39]: moive.Genre = moive.Genre.astype('category')
         moive.Year = moive.Year.astype('category')
```

```
In [41]: moive.Genre
```

```
Out[41]: 0      Comedy
         1      Adventure
         2      Action
         3      Adventure
         4      Comedy
         ...
        554     Comedy
        555     Comedy
        556     Thriller
        557     Action
        558     Comedy
        Name: Genre, Length: 559, dtype: category
        Categories (7, object): ['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance', 'Thriller']
```

```
In [43]: moive.Year
```

```
Out[43]: 0      2009
         1      2008
         2      2009
         3      2010
         4      2009
         ...
        554     2011
        555     2009
        556     2007
        557     2009
        558     2011
        Name: Year, Length: 559, dtype: category
        Categories (5, int64): [2007, 2008, 2009, 2010, 2011]
```

```
In [45]: moive.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   CriticRatinig        559 non-null   int64
 3   AudioRating          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 [47]: moive.Genre.cat.categories
```

```
Out[47]: Index(['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance',
               'Thriller'],
              dtype='object')
```

```
In [49]: moive.describe()
```

Out[49]:

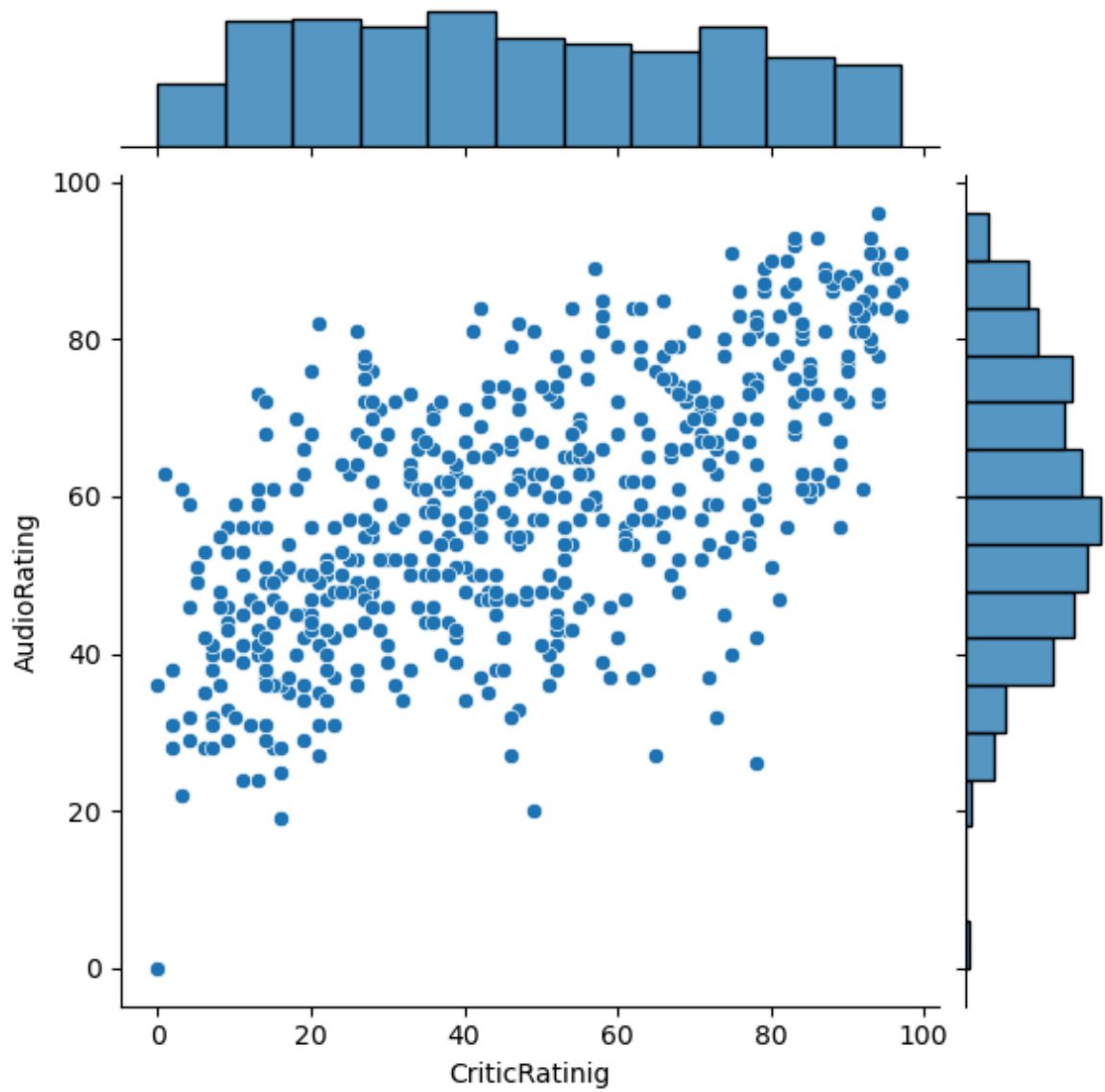
	CriticRatinig	AudioRating	BudgetMillions
count	559.000000	559.000000	559.000000
mean	47.309481	58.744186	50.236136
std	26.413091	16.826887	48.731817
min	0.000000	0.000000	0.000000
25%	25.000000	47.000000	20.000000
50%	46.000000	58.000000	35.000000
75%	70.000000	72.000000	65.000000
max	97.000000	96.000000	300.000000

```
In [51]: from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

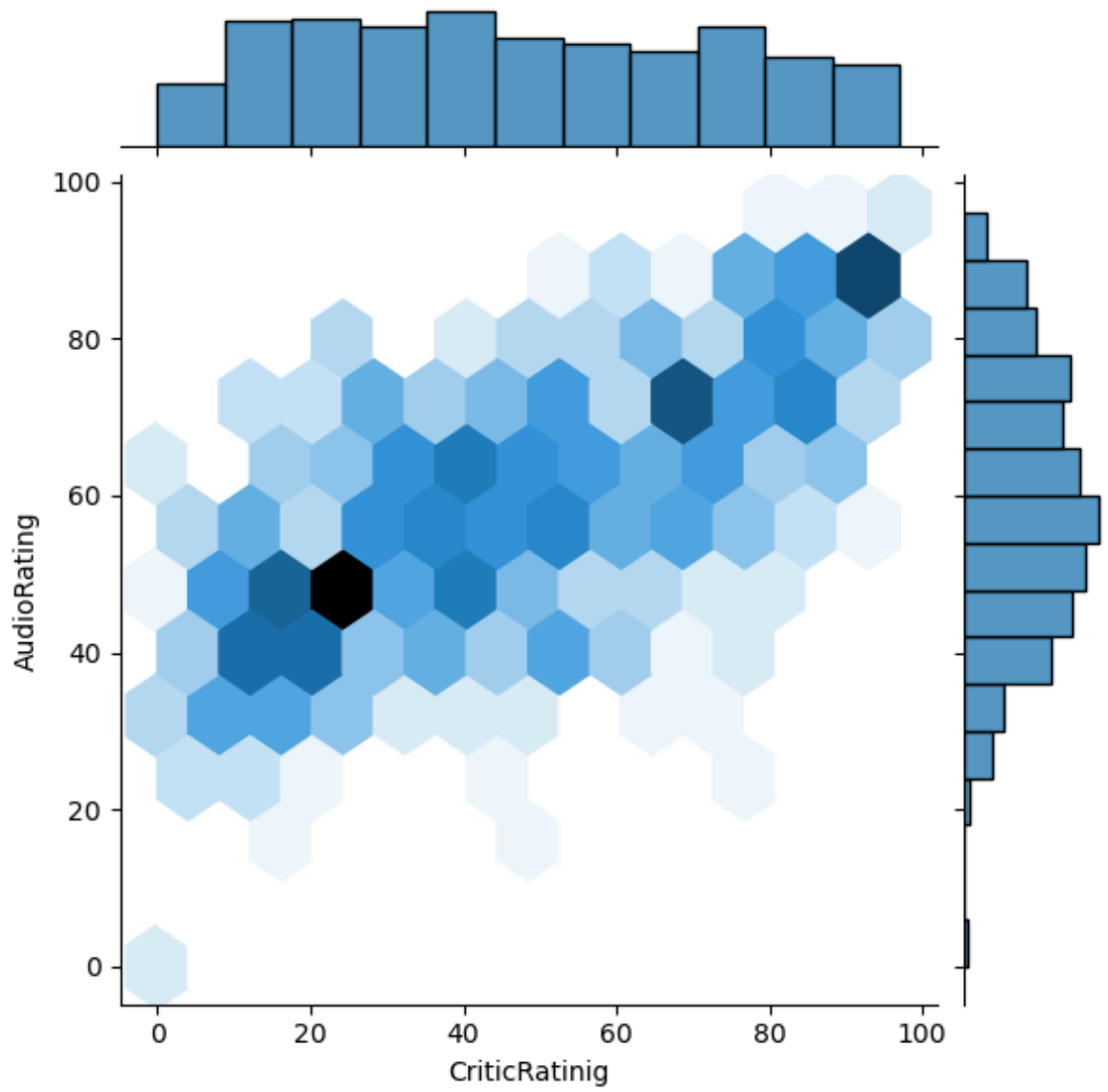
- basically joint plot is a scatter plot & it find the relation b/w audiene & critics
- also if you look up you can find the uniform distribution (critics)and normal distriution (audience)

```
In [53]: vis = sns.jointplot( data = moive, x = 'CriticRatinig', y = 'AudioRating')

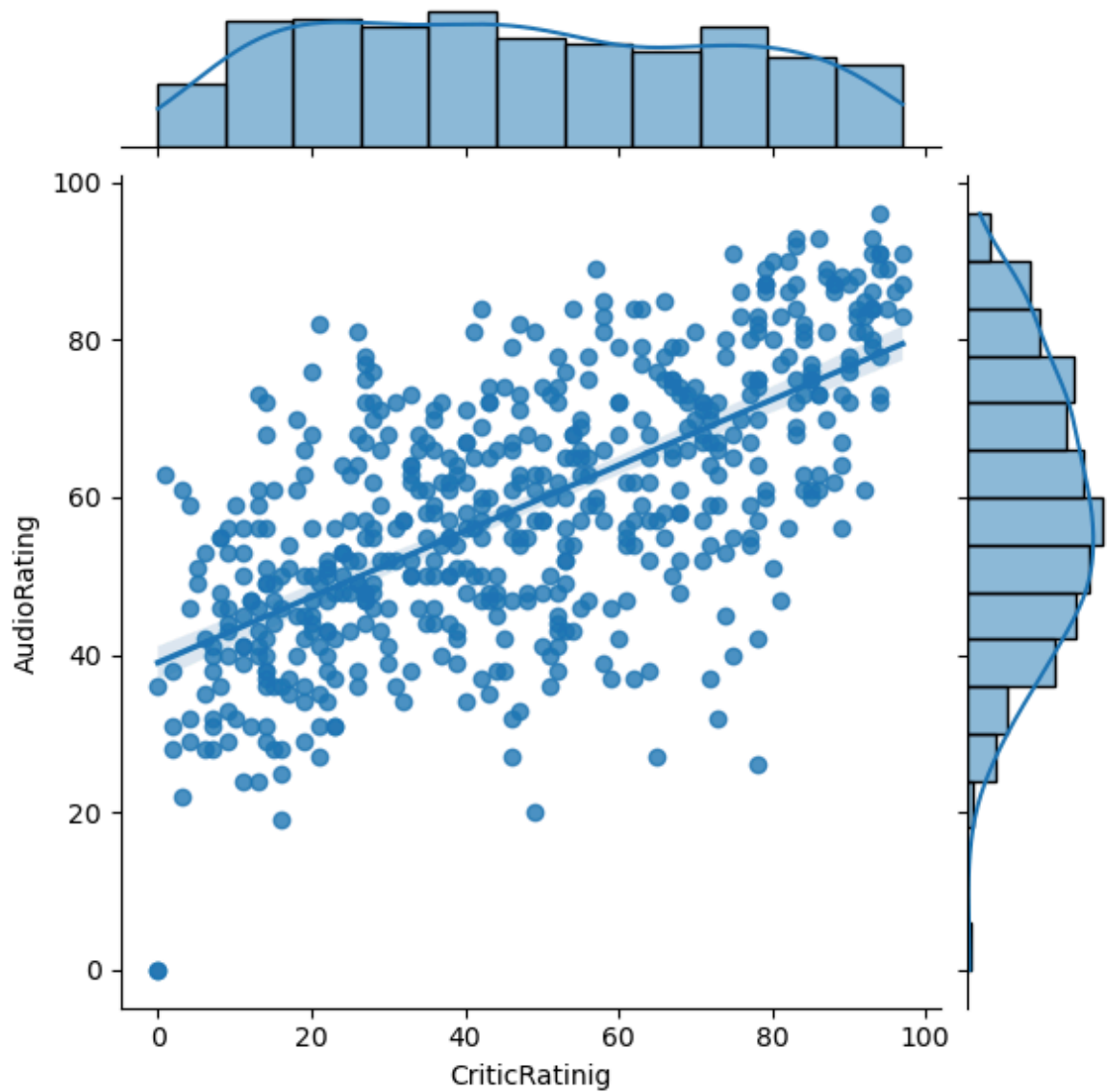
# Audience rating is more dominant then critics rating
# Based on this we find out as most people are most liklihood to watch audience
# Let me explain the excel - if you filter audience rating & critic rating. crit
```



```
In [54]: vis = sns.jointplot(data = moive , x='CriticRatinig', y='AudioRating',kind='hex')
```

```
In [55]: vis = sns.jointplot(data = moive , x='CriticRatinig', y='AudioRating',kind='reg')
```



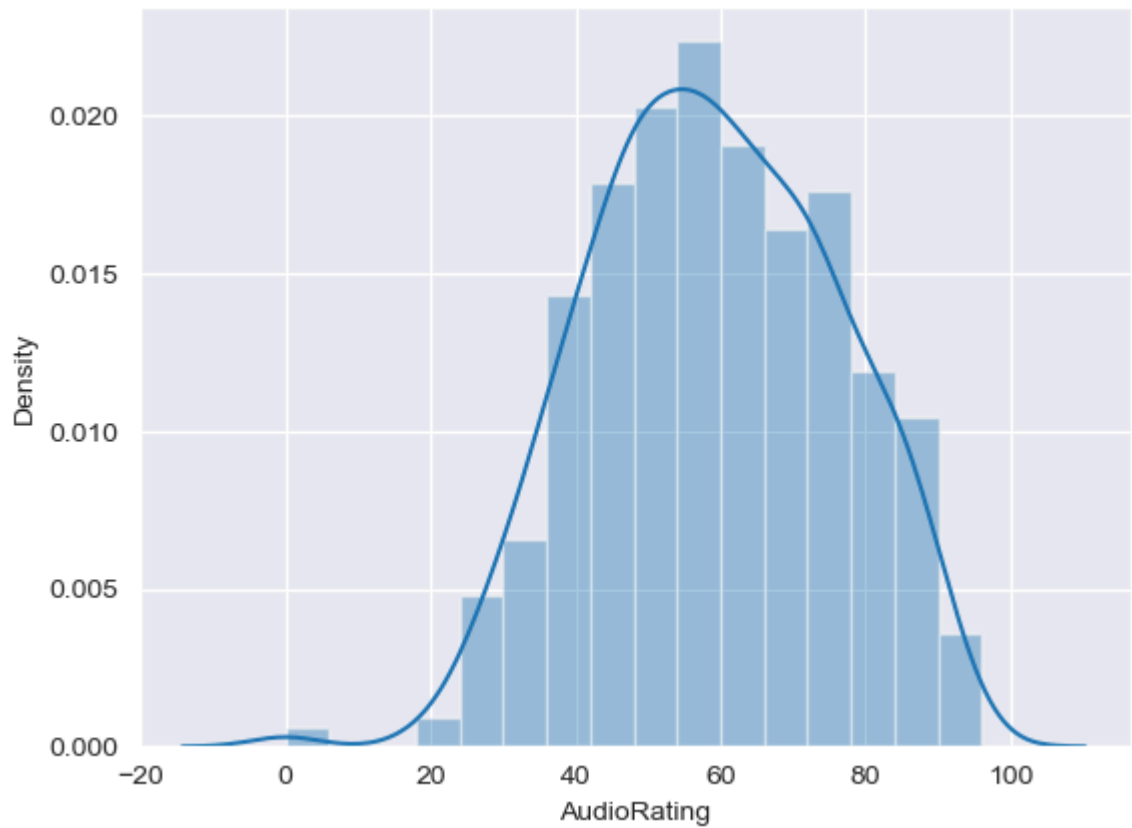
```
In [56]: sns.set_style('darkgrid')
```

```
In [57]: #Histograms

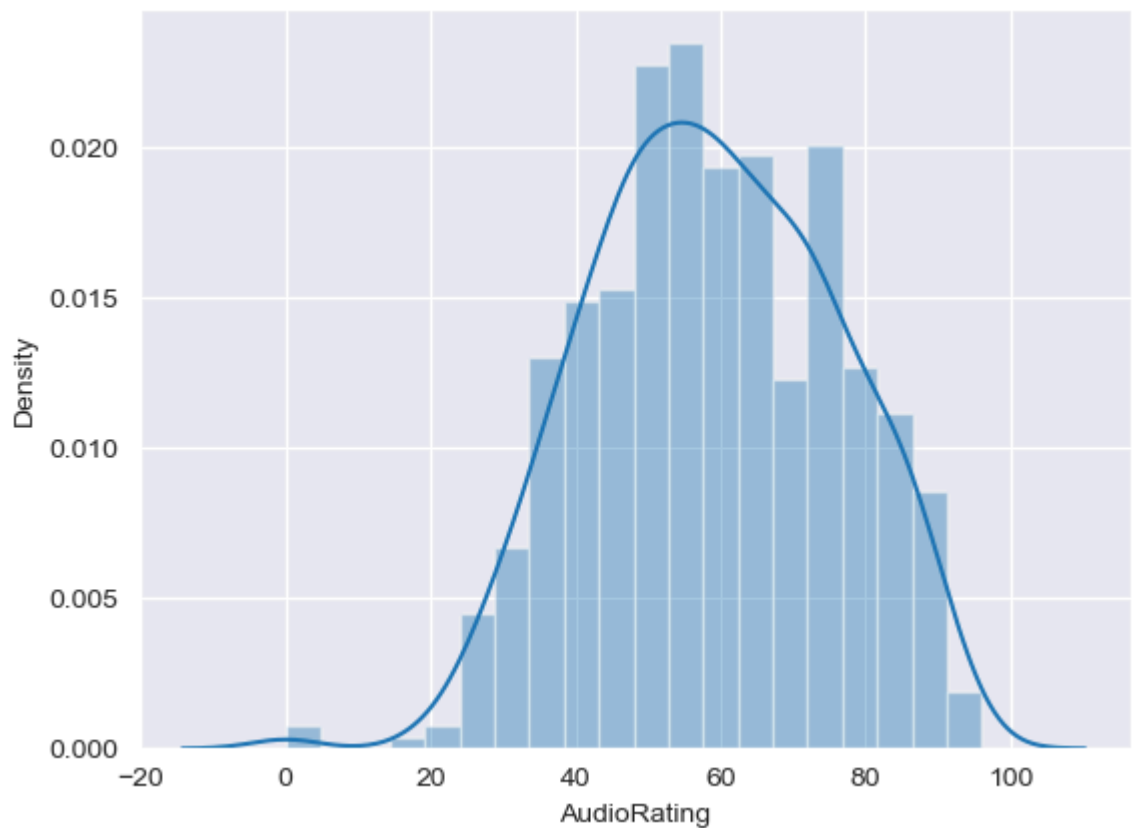
# <<< chat1

vis1 = sns.distplot(moive.AudioRating)

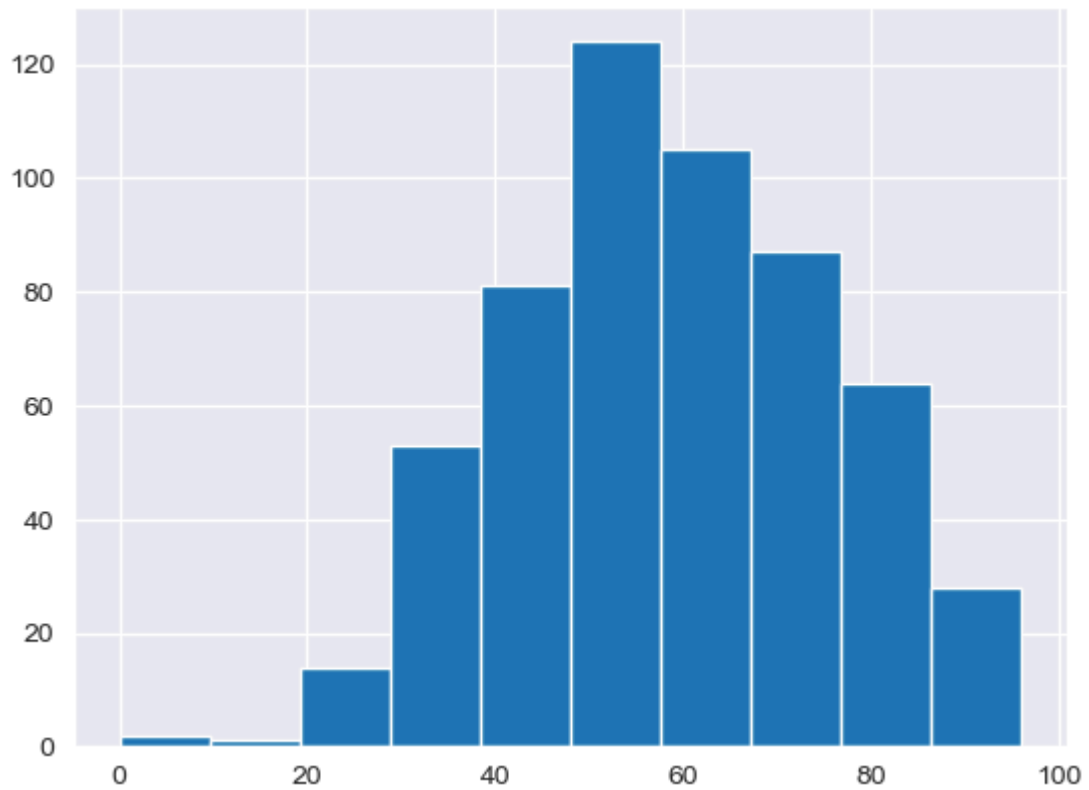
#y - axis generated by seaborn automatically that is the powefull of seaborn gal
```



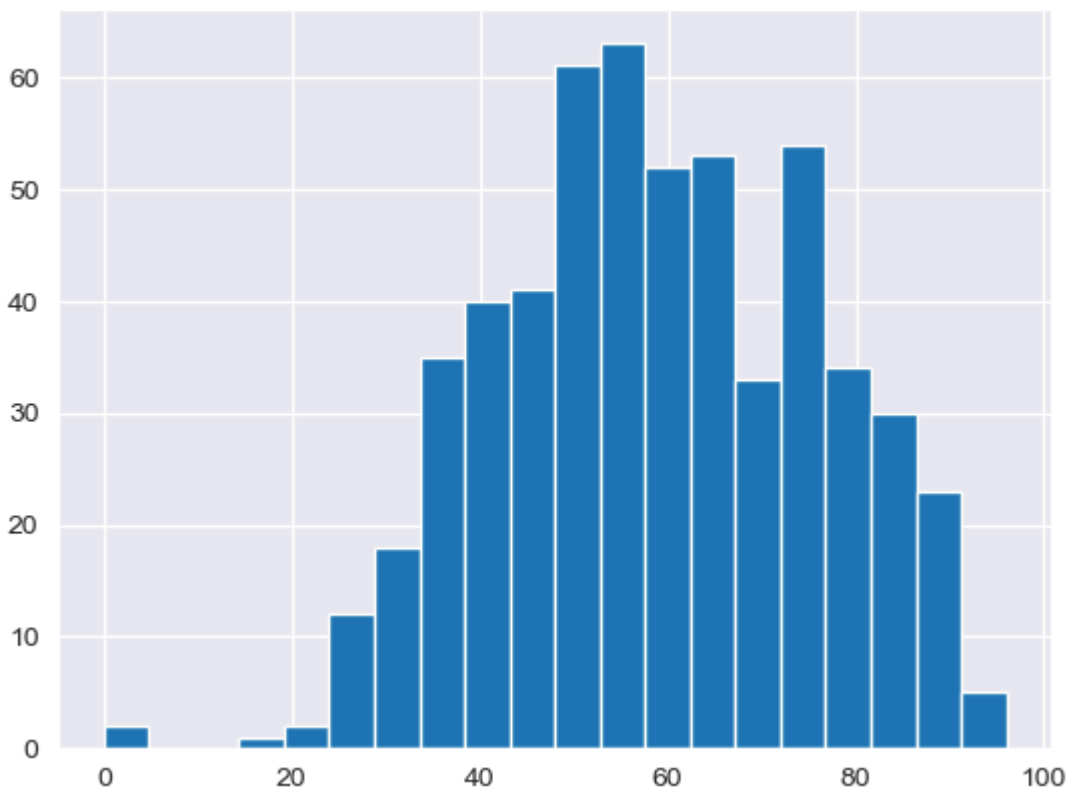
```
In [58]: vis1 = sns.distplot(moive.AudioRating, bins=20)
```



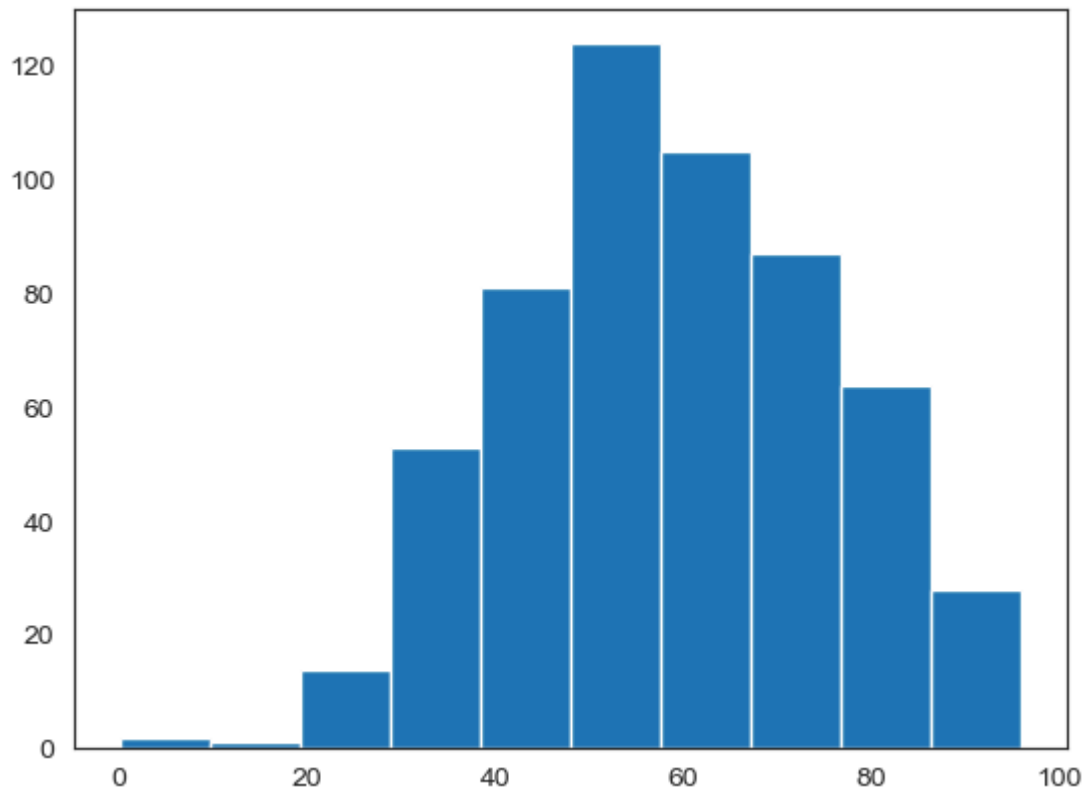
```
In [59]: vis1 = plt.hist(moive.AudioRating)
```



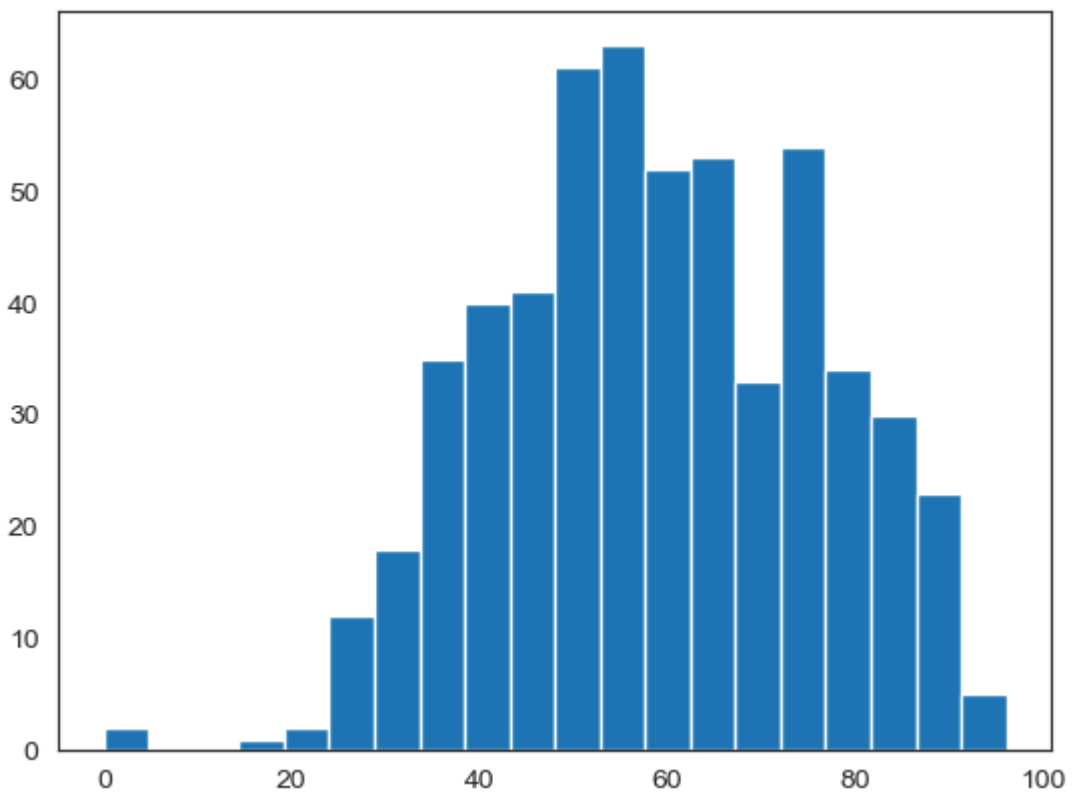
```
In [61]: vis1 = plt.hist(moive.AudioRating,bins=20)
```



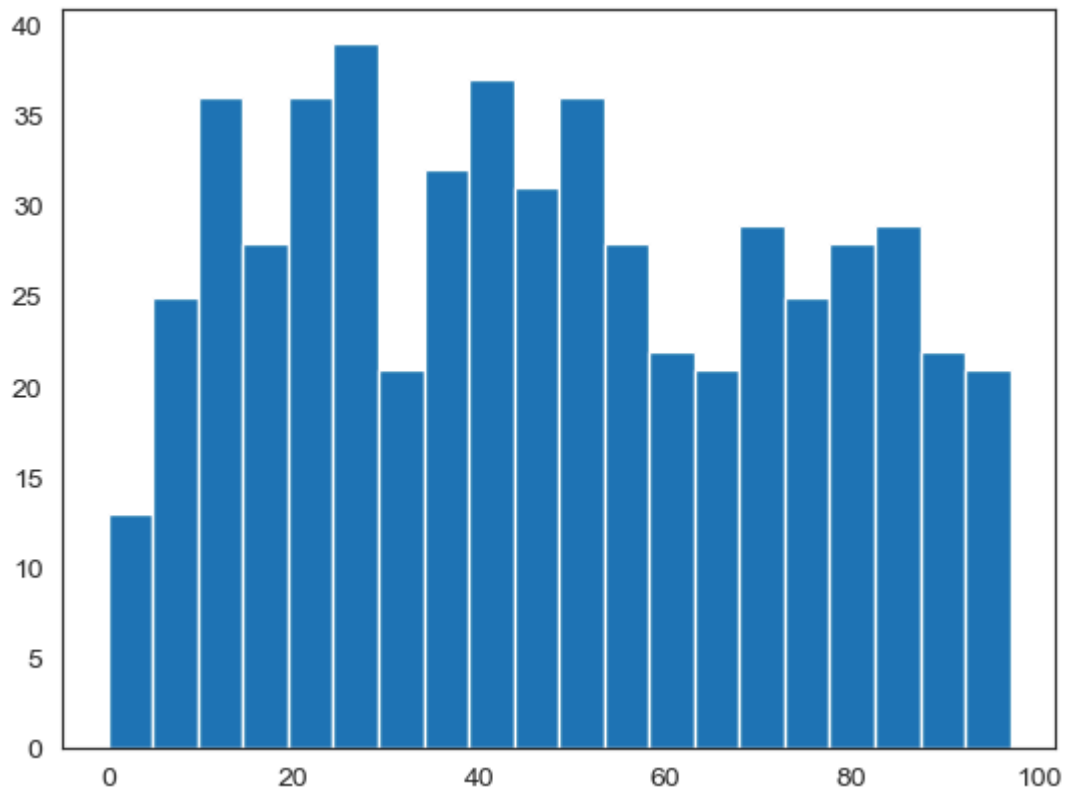
```
In [62]: sns.set_style('white')  
vis1 = plt.hist(moive.AudioRating)
```



```
In [64]: vis1 = plt.hist(moive.AudioRating,bins=20)
```

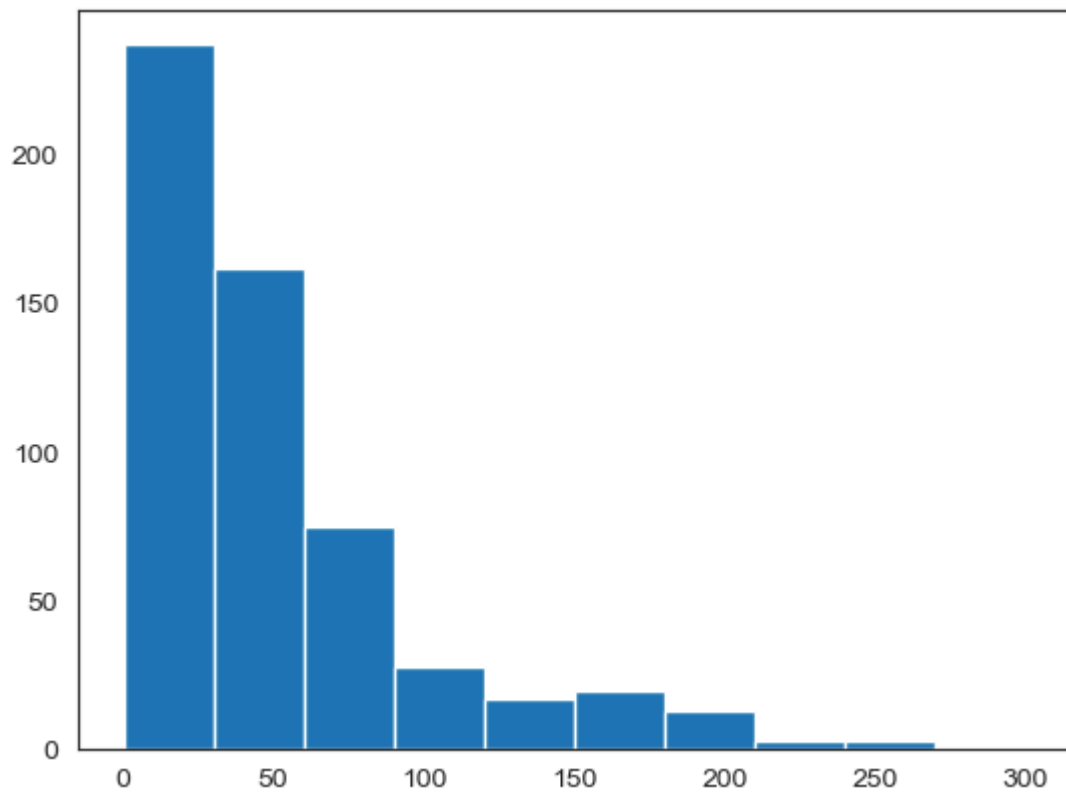


```
In [66]: vis1 = plt.hist(moive.CriticRatinig,bins=20)
```



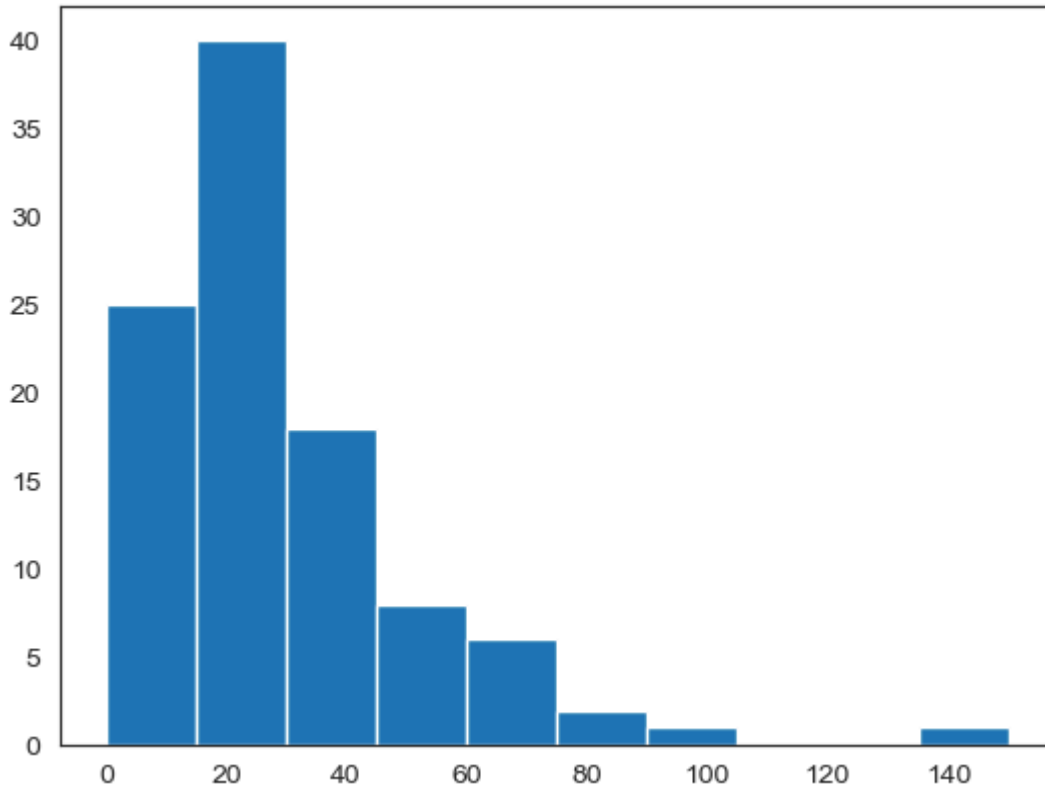
```
In [67]: # Creating stacked histograms & this is bit tough to understand  
#chat-2
```

```
In [68]: plt.hist(moive.BudgetMillions)  
plt.show()
```



```
In [71]: plt.hist(moive[moive.Genre == 'Drama'].BudgetMillions)
```

```
Out[71]: (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 [73]: moive.head()
```

```
Out[73]:
```

	Film	Genre	CriticRatinig	AudioRating	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

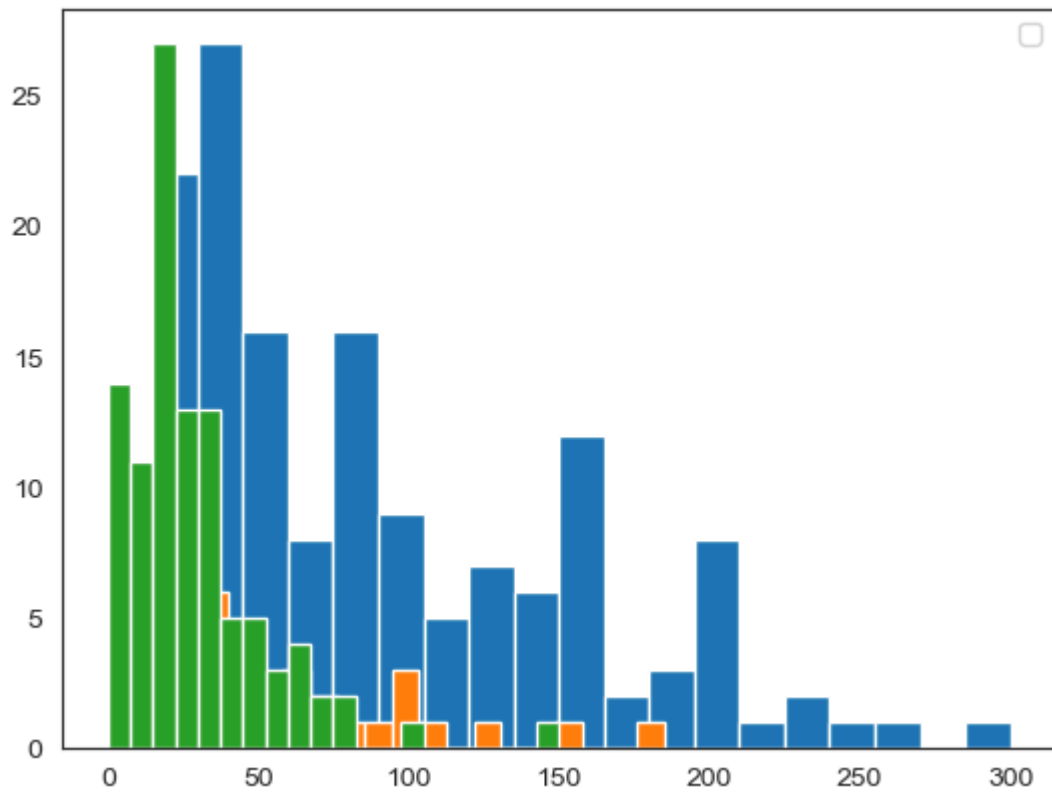
```
In [75]: moive.Genre.unique()
```

```
Out[75]: ['Comedy', 'Adventure', 'Action', 'Horror', 'Drama', 'Romance', 'Thriller']
Categories (7, object): ['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance', 'Thriller']
```

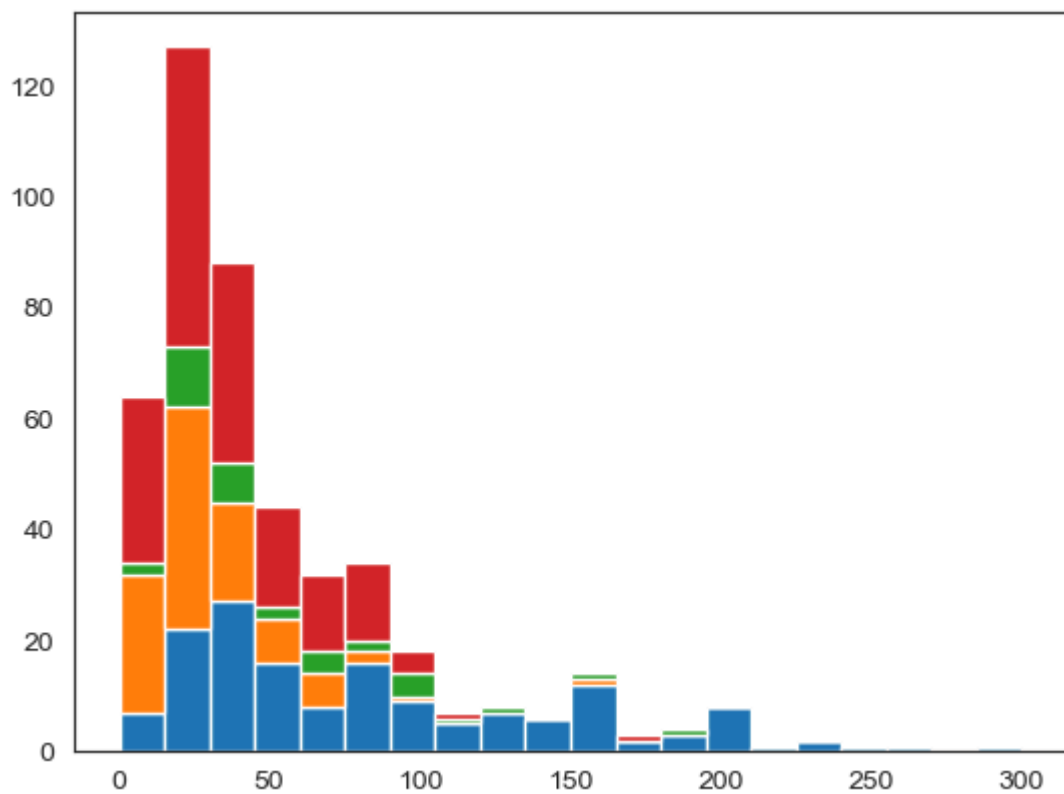
```
In [77]: # Below plot are stacked histogram becuae overlaped
```

```
plt.hist(moive[moive.Genre == 'Action'].BudgetMillions,bins=20)
plt.hist(moive[moive.Genre == 'Thriller'].BudgetMillions,bins=20)
plt.hist(moive[moive.Genre == 'Drama'].BudgetMillions,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 when legend() is called with no argument.



```
In [79]: plt.hist([moive[moive.Genre == 'Action'].BudgetMillions,\
moive[moive.Genre == 'Drama'].BudgetMillions,\
moive[moive.Genre == 'Thriller'].BudgetMillions,\
moive[moive.Genre == 'Comedy'].BudgetMillions],\
bins = 20,stacked = True)\
plt.show()
```



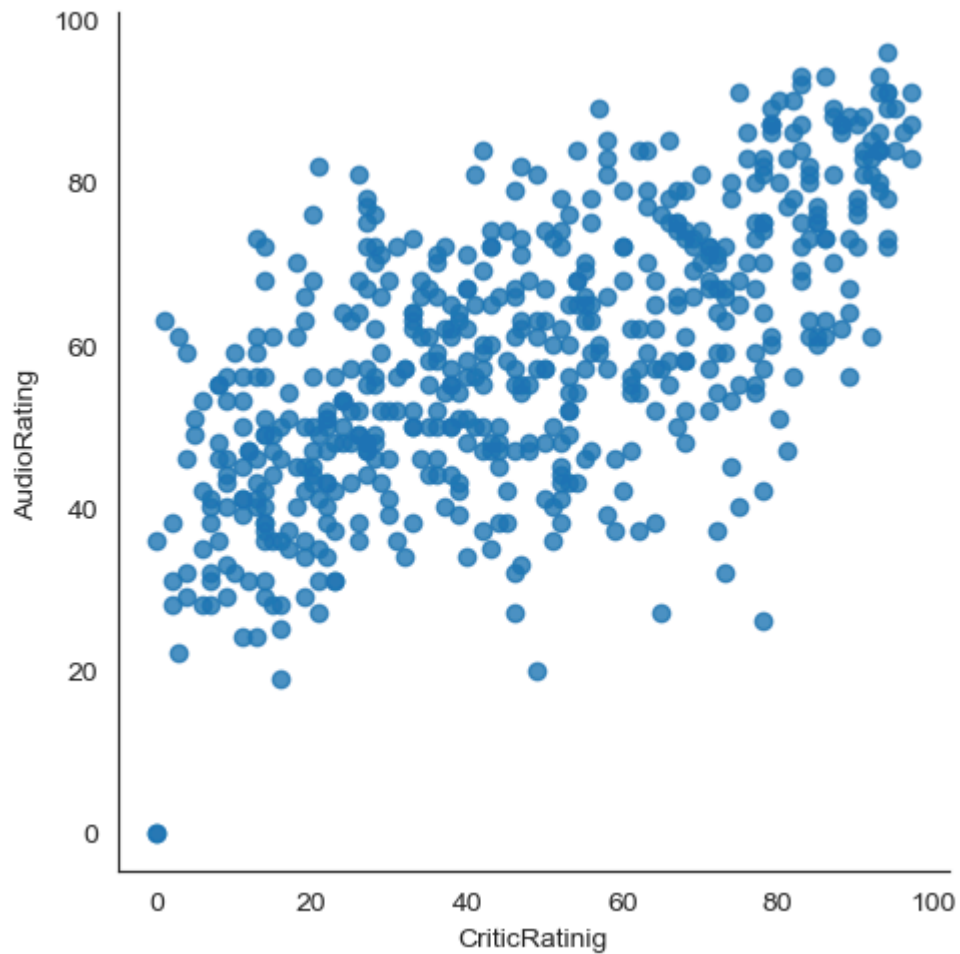
```
In [81]: # if you have 100 categories you cannot copy & paste all the things
```



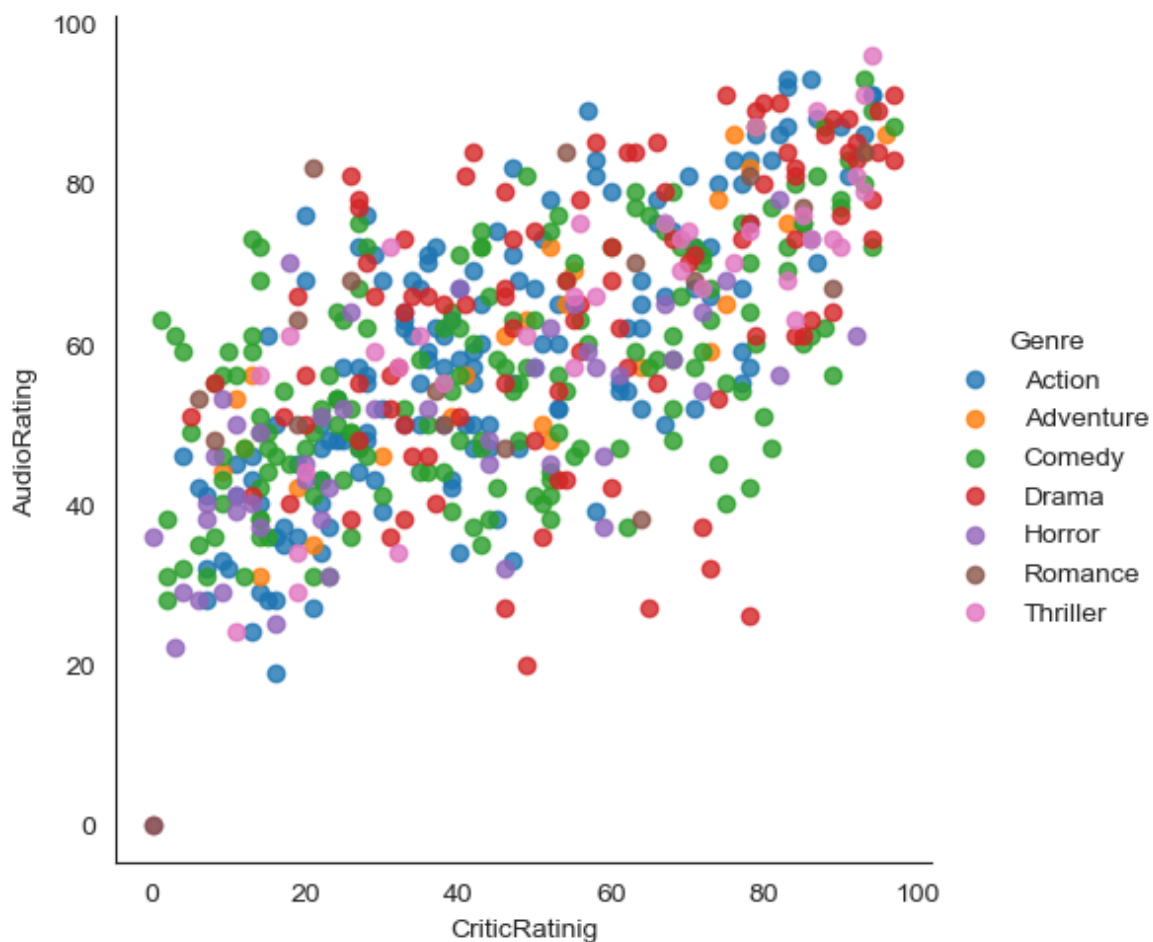
```
for gen in moive.Genre.cat.categories:  
    print(gen)
```

Action
Adventure
Comedy
Drama
Horror
Romance
Thriller

```
In [84]: vis1 = sns.lmplot(data=moive,x='CriticRatinig',y='AudioRating',\  
                           fit_reg=False)
```



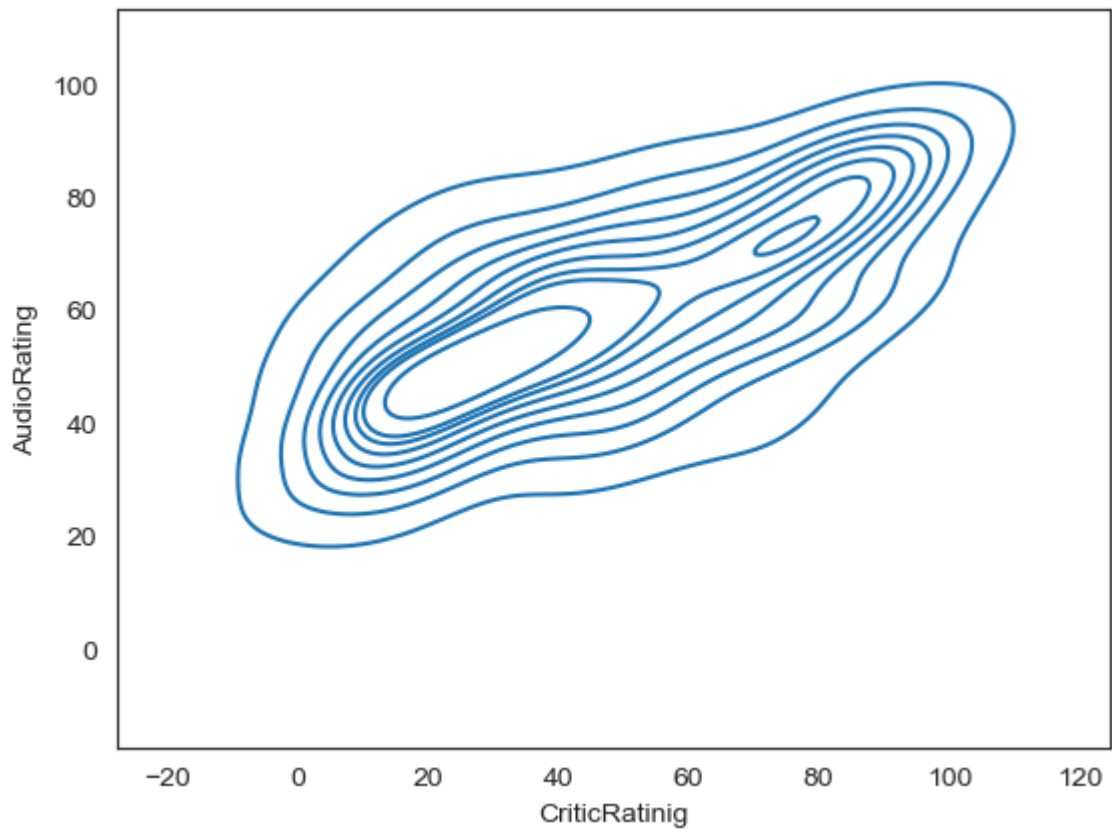
```
In [87]: # hue  
vis1 = sns.lmplot(data=moive,x='CriticRatinig',y='AudioRating',\  
                  fit_reg=False,hue='Genre',aspect=1)
```



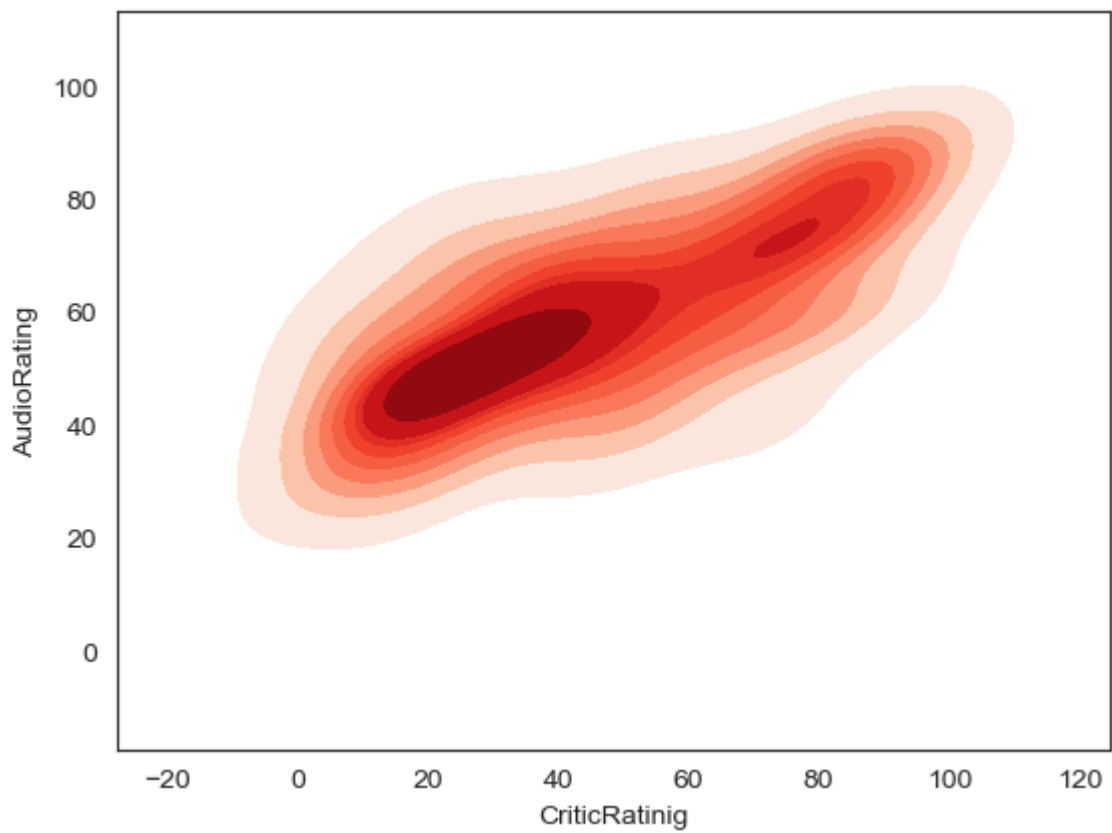
```
In [88]: # Kernel Density Estimate plot (KDE PLOT)
# How can i visualize audience rating & critics rating. using scatterplot
```

```
In [92]: k1 = sns.kdeplot(data=moive,x=moive.CriticRatinig,y=moive.AudioRating)

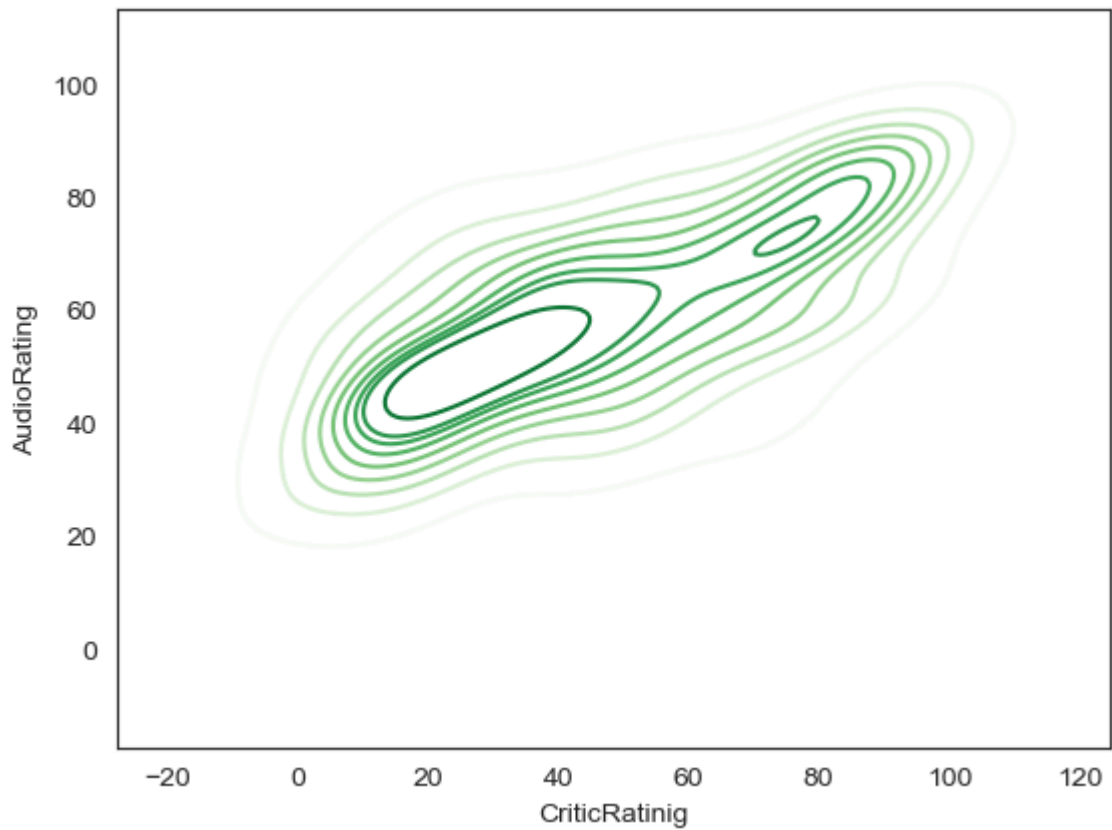
# where do u find more density and how density is distributed across from the the
# center point is kernal this is calld KDE & insteade of dots it visualize like
# we can able to clearly see the spread at the audience ratings
```



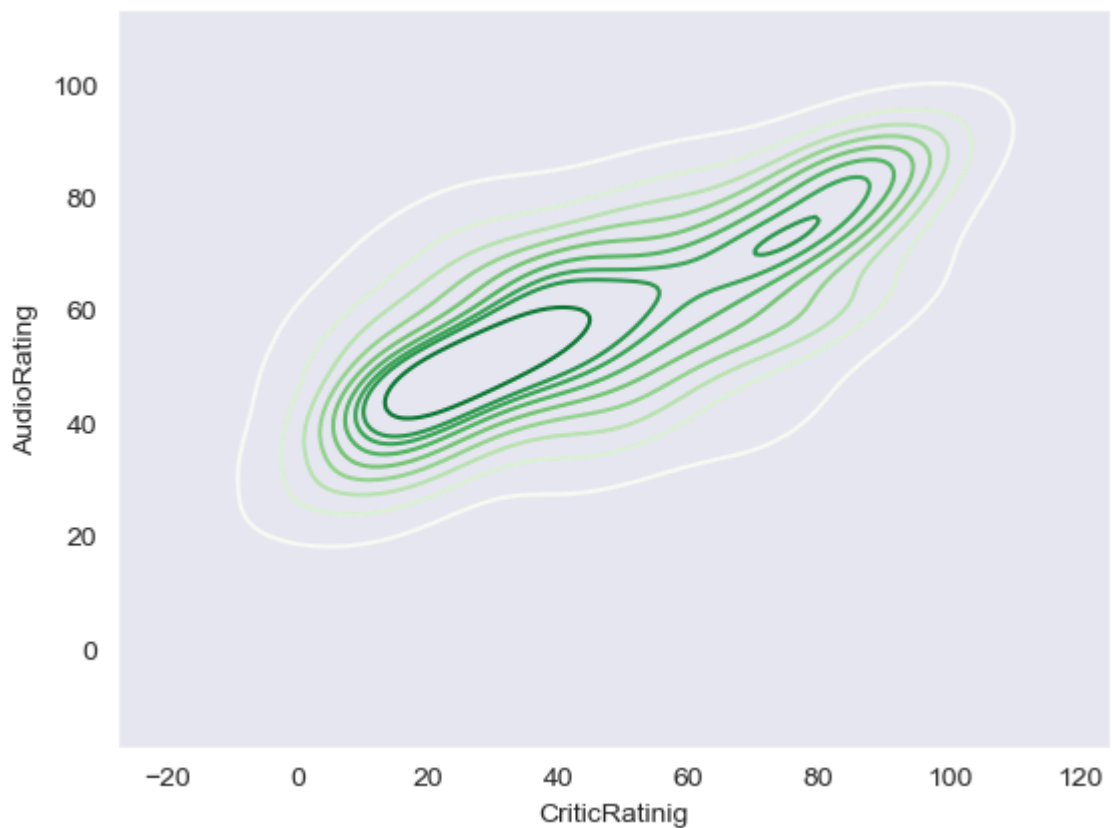
```
In [94]: k1 = sns.kdeplot(data=moive,x=moive.CriticRatinig,y=moive.AudioRating,shade = Tr
```



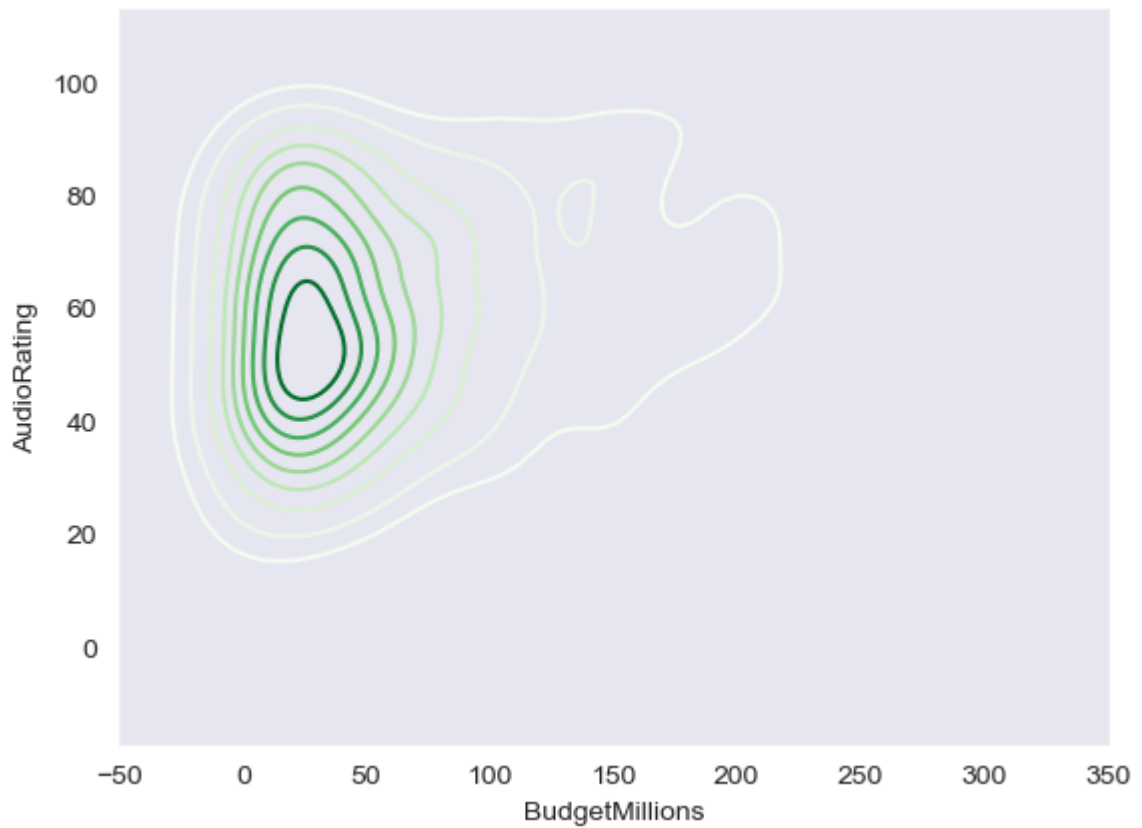
```
In [95]: k1 = sns.kdeplot(data=moive,x=moive.CriticRatinig,y=moive.AudioRating,shade = Fa
```



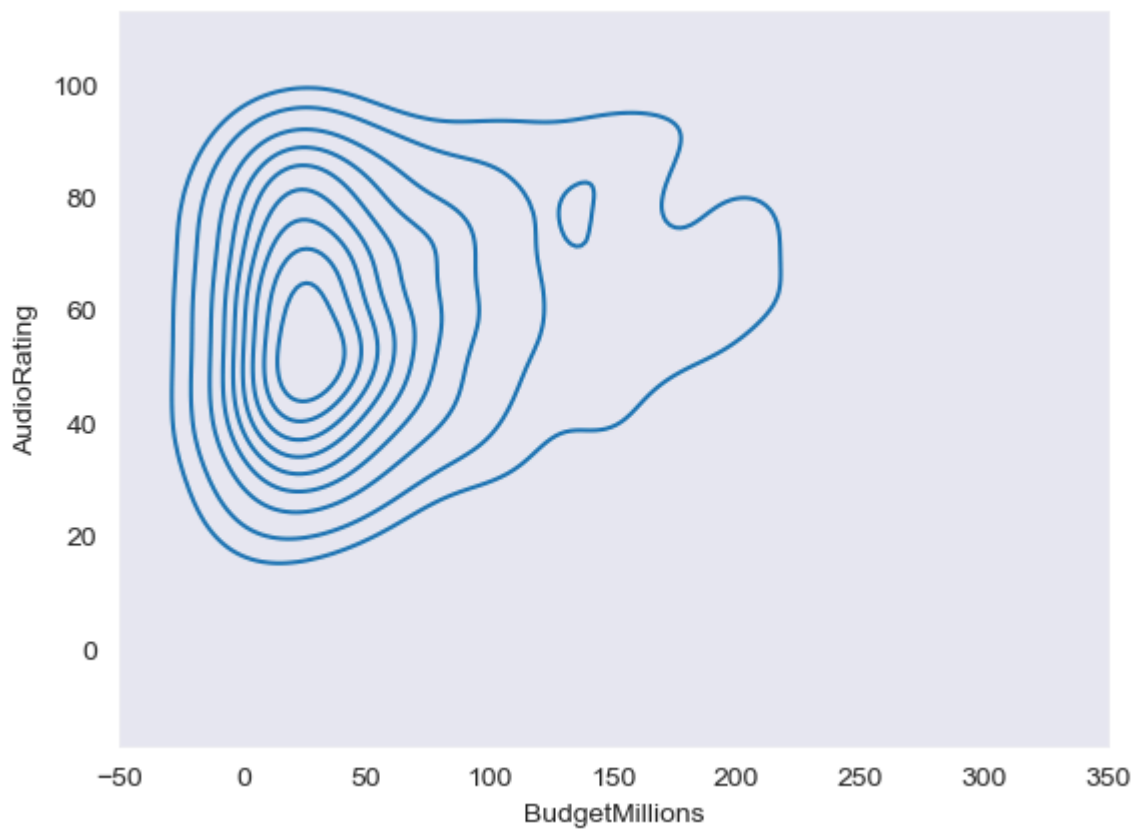
```
In [97]: sns.set_style('dark')
k1 = sns.kdeplot(data=moive,x=moive.CriticRatinig,y=moive.AudioRating,shade = Fa
```



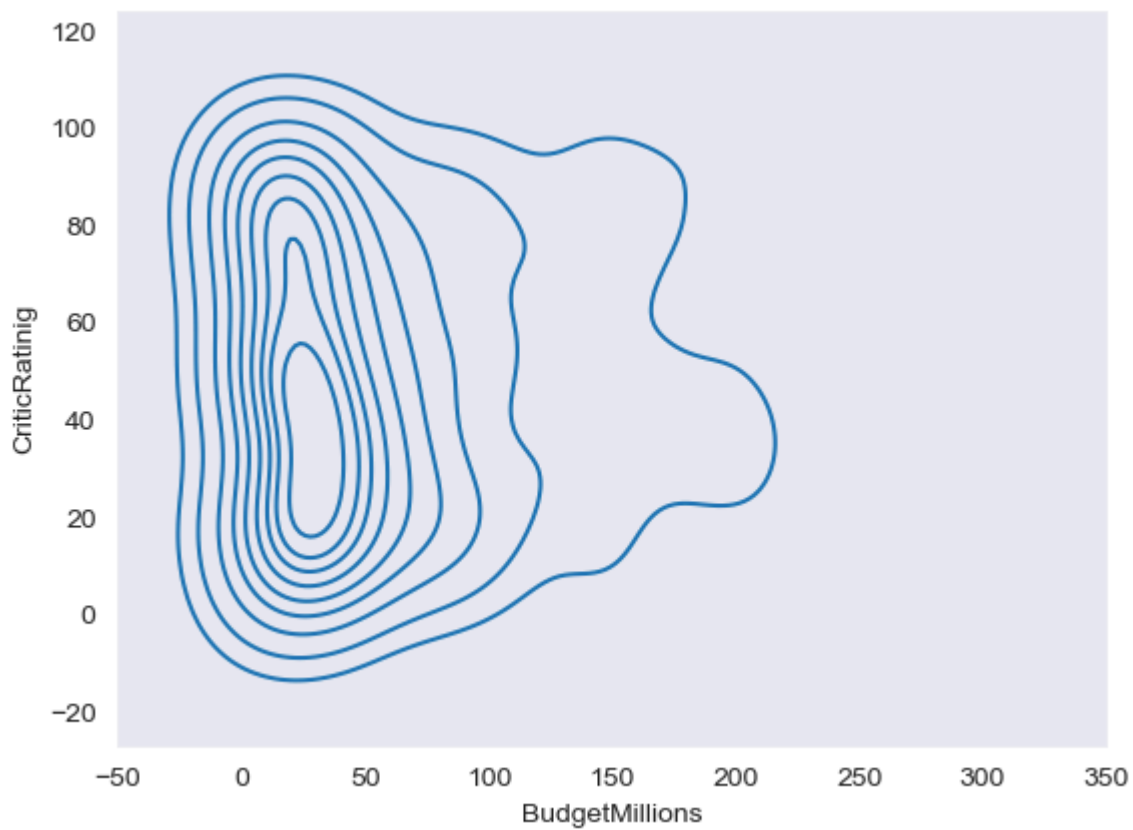
```
In [99]: sns.set_style('dark')
k1 = sns.kdeplot(data=moive,x=moive.BudgetMillions,y=moive.AudioRating,shade = F
```



```
In [101... sns.set_style('dark')  
k1 = sns.kdeplot(data=moive,x=moive.BudgetMillions,y=moive.AudioRating)
```



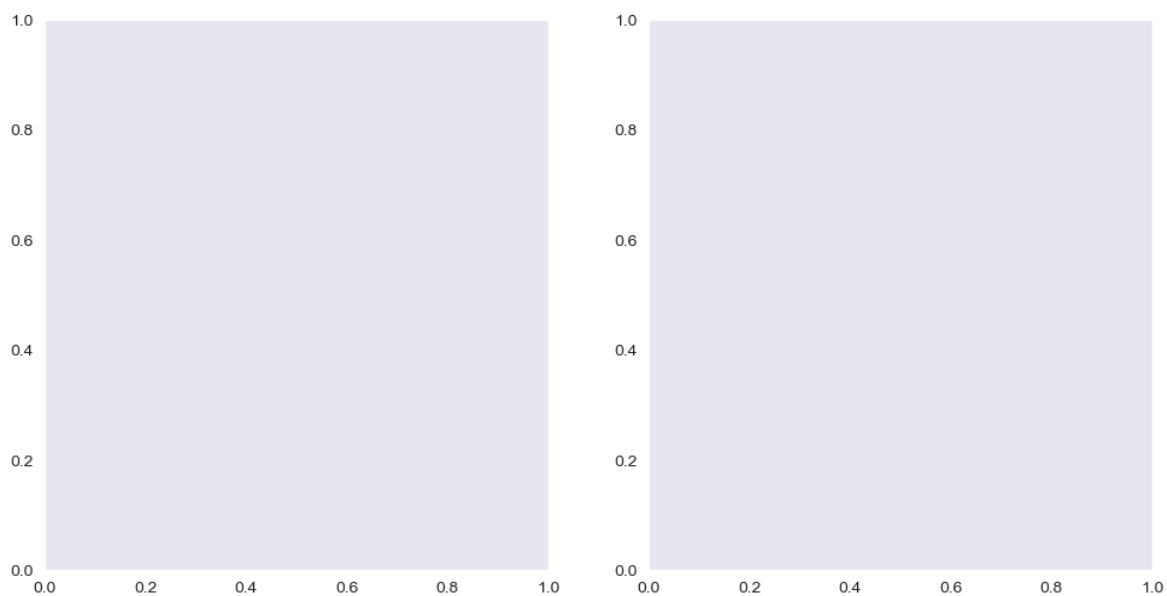
```
In [102... k2 = sns.kdeplot(x=moive.BudgetMillions,y=moive.CriticRatinig)
```



In [103...

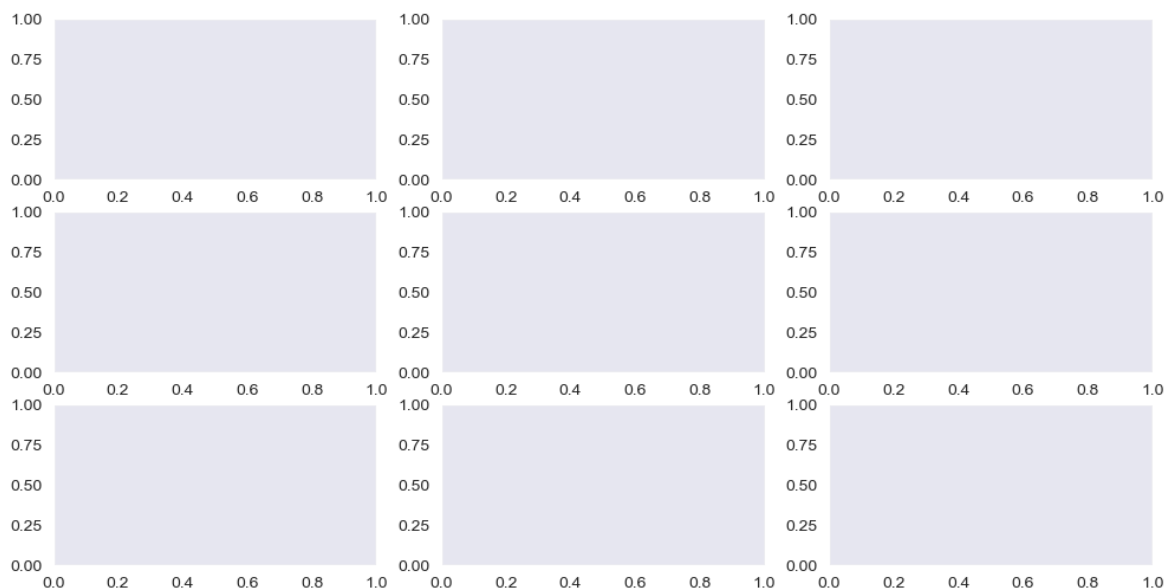
#SUBPOLS

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



In [106...

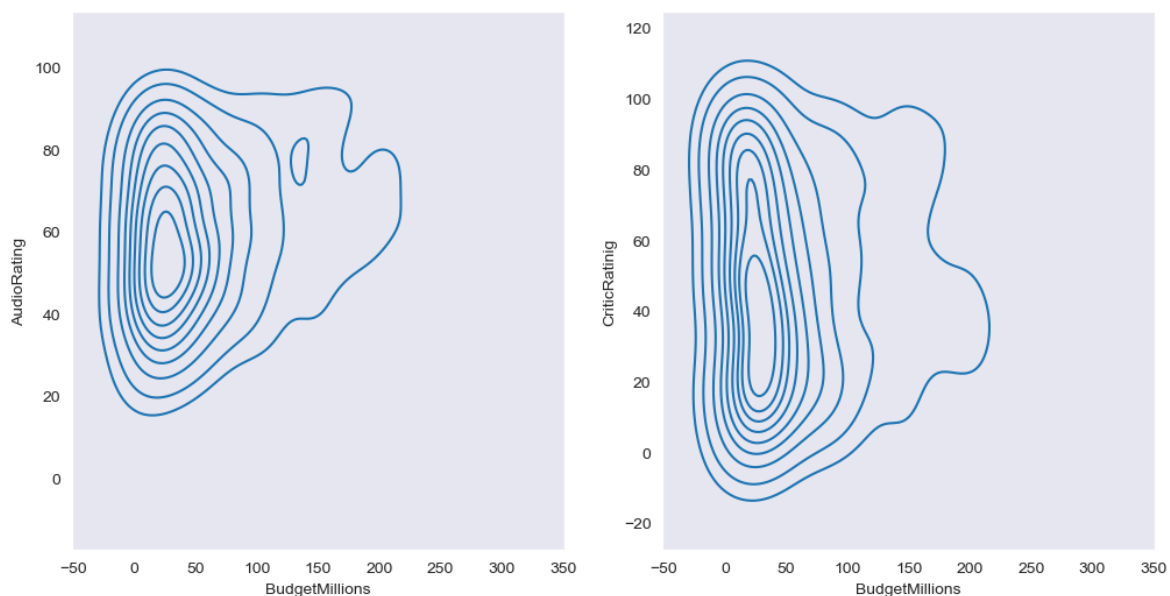
```
f, ax = plt.subplots(3,3, figsize=(12,6))
```



In [107...

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

k1 = sns.kdeplot(data=moive,x=moive.BudgetMillions,y=moive.AudioRating,ax=axes[0])
k2 = sns.kdeplot(x=moive.BudgetMillions,y=moive.CriticRating,ax=axes[1])
```



In [108...

axes

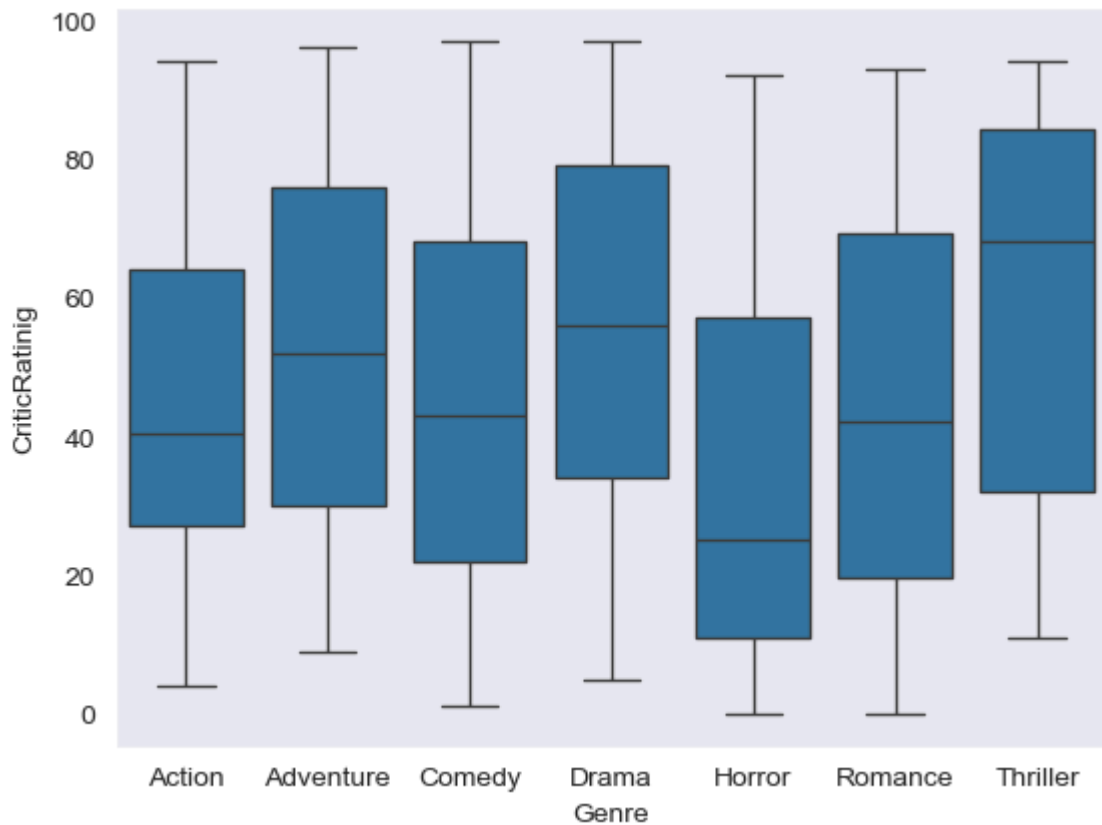
Out[108...

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

In [110...

```
#Box plots

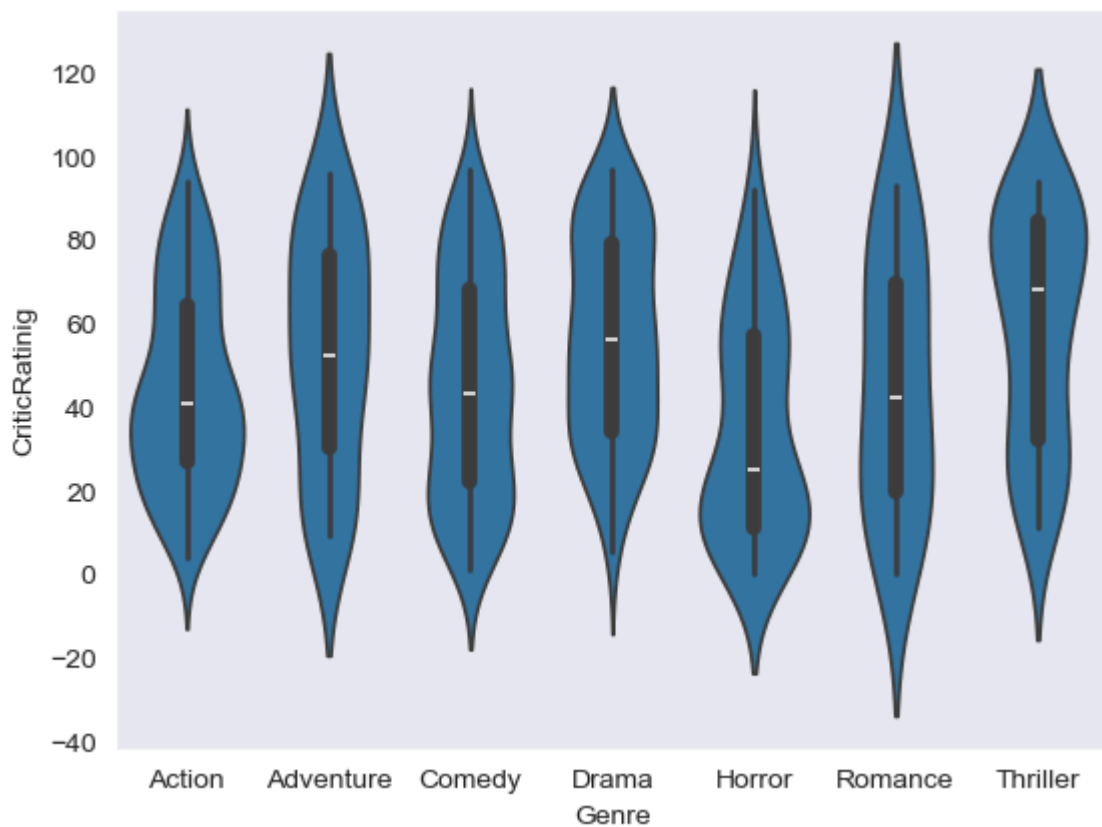
w=sns.boxplot(data=moive,x='Genre',y = 'CriticRating')
```



In [111...

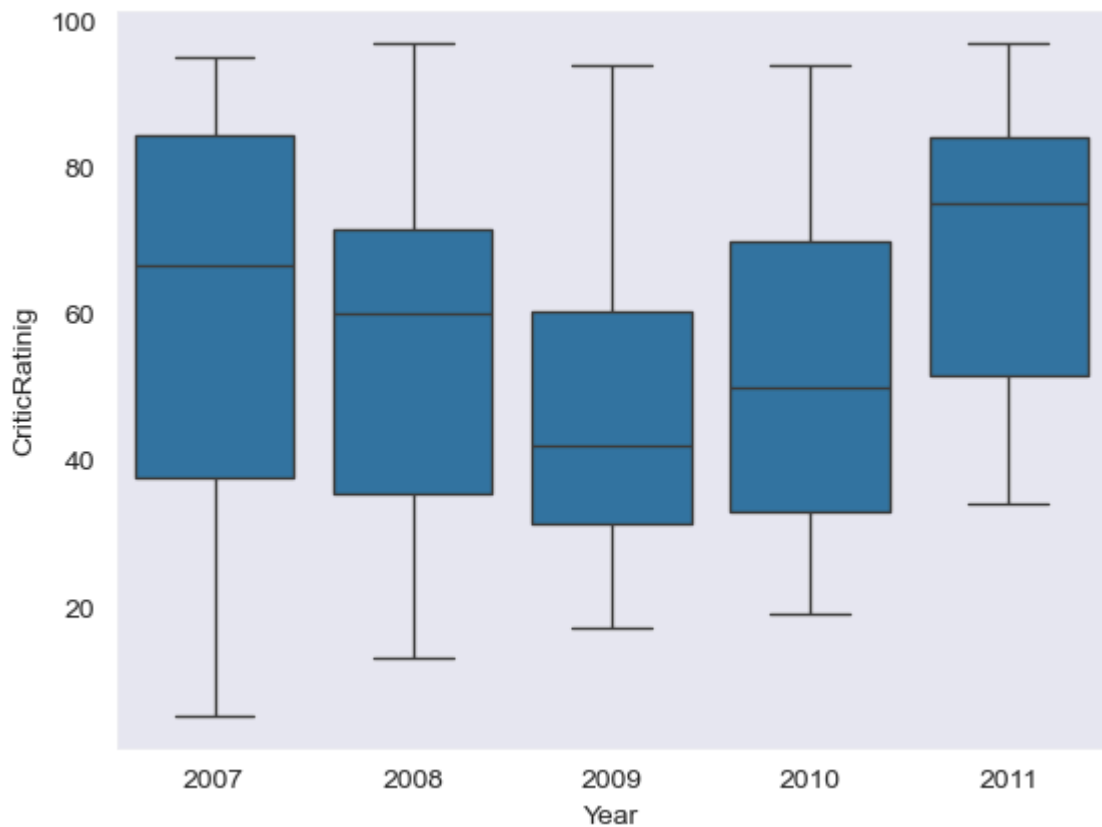
#violin plot

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

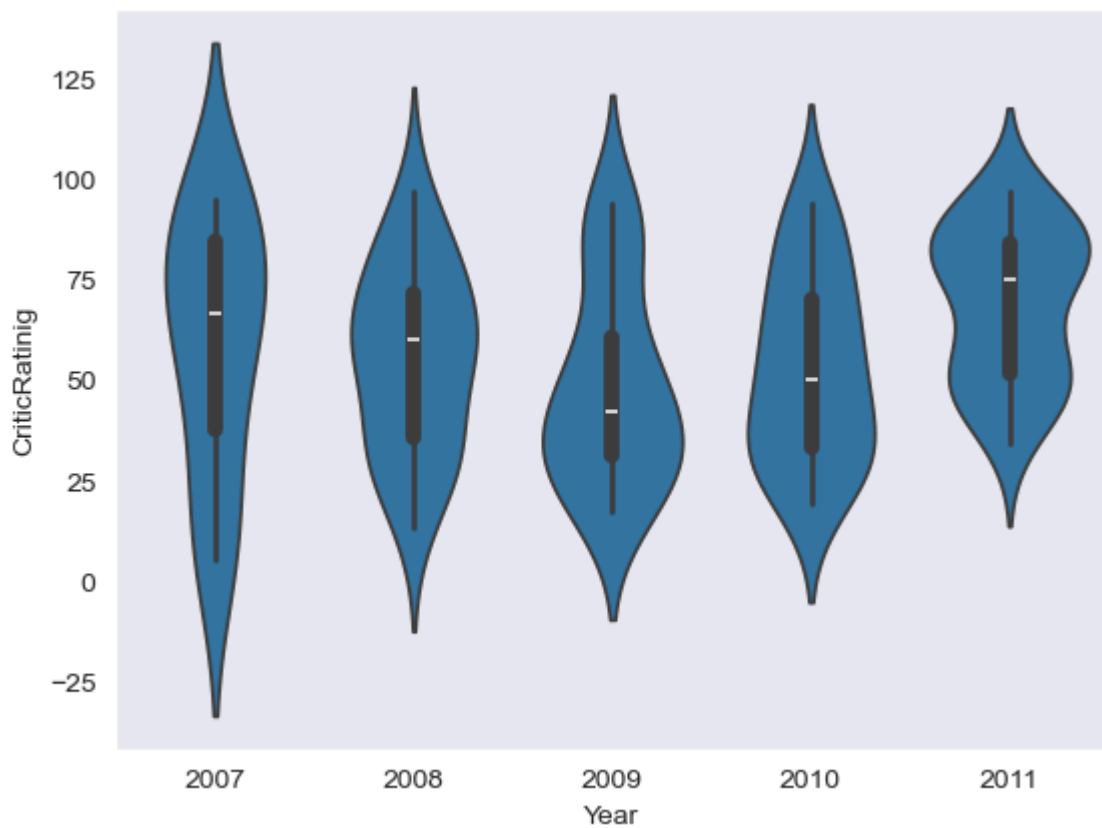


In [113...

```
w1 = sns.boxplot(data=moive[moive.Genre == 'Drama'],x='Year',y='CriticRating')
```

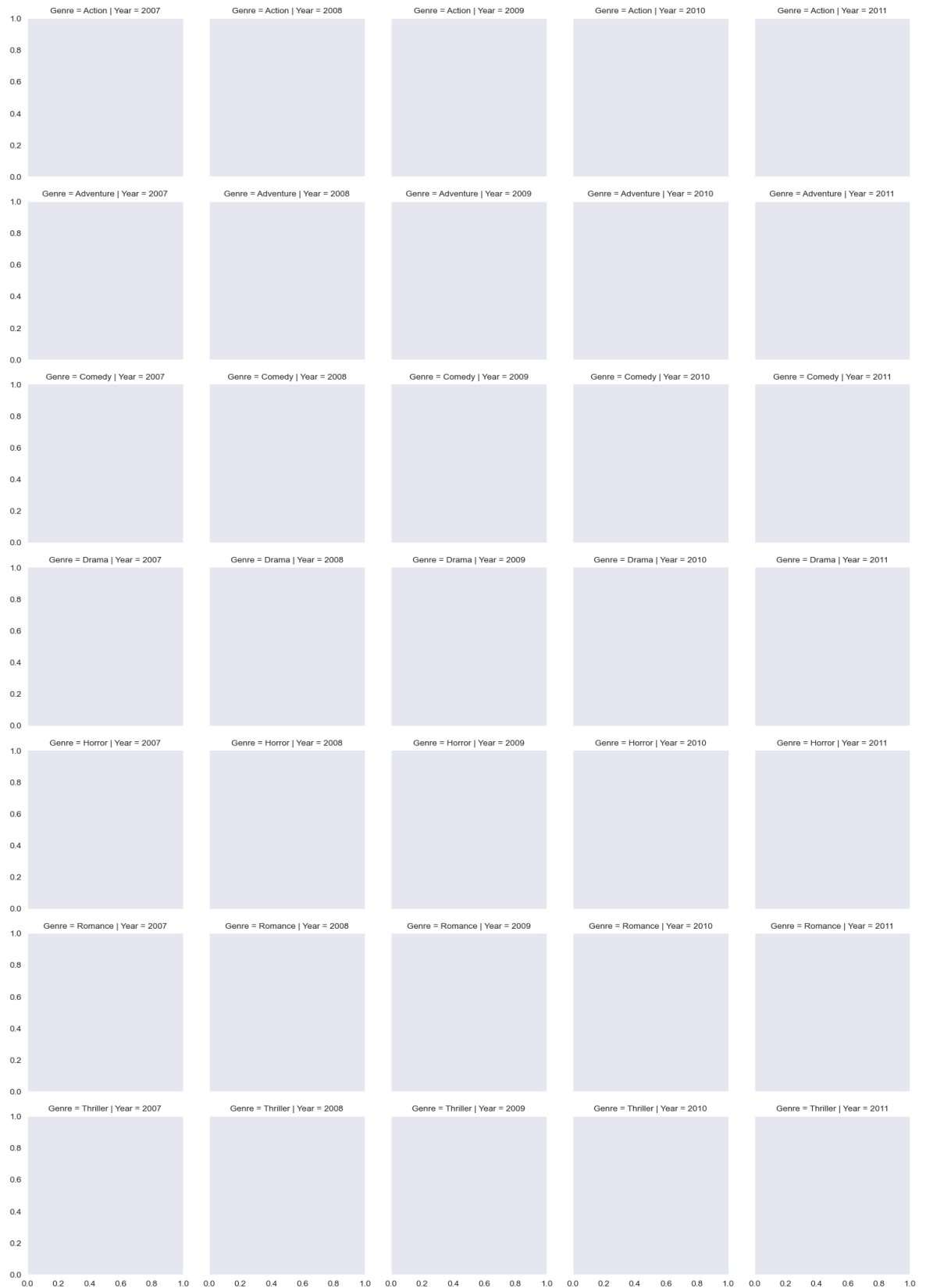



In [114... `z = sns.violinplot(data=moive[moive.Genre == 'Drama'], x='Year', y='CriticRating')`



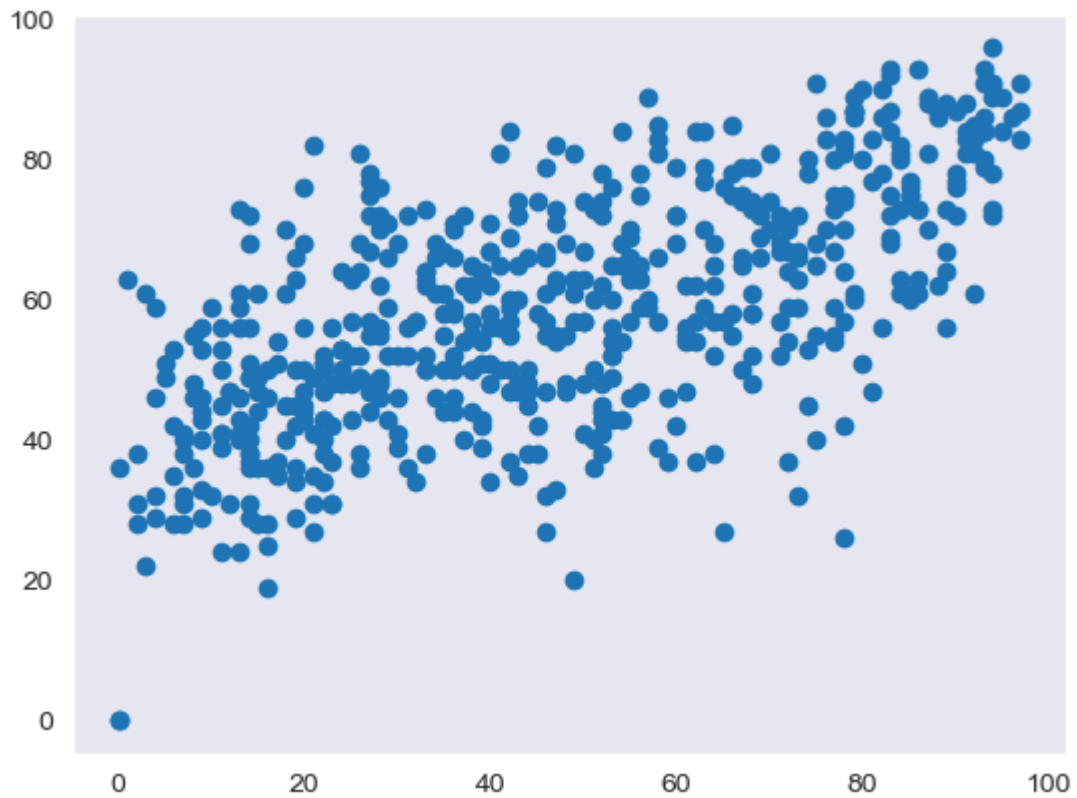
In [116... `#createing a Facet Grid`

In [117... `g = sns.FacetGrid (moive,row = 'Genre', col = 'Year', hue='Genre')`



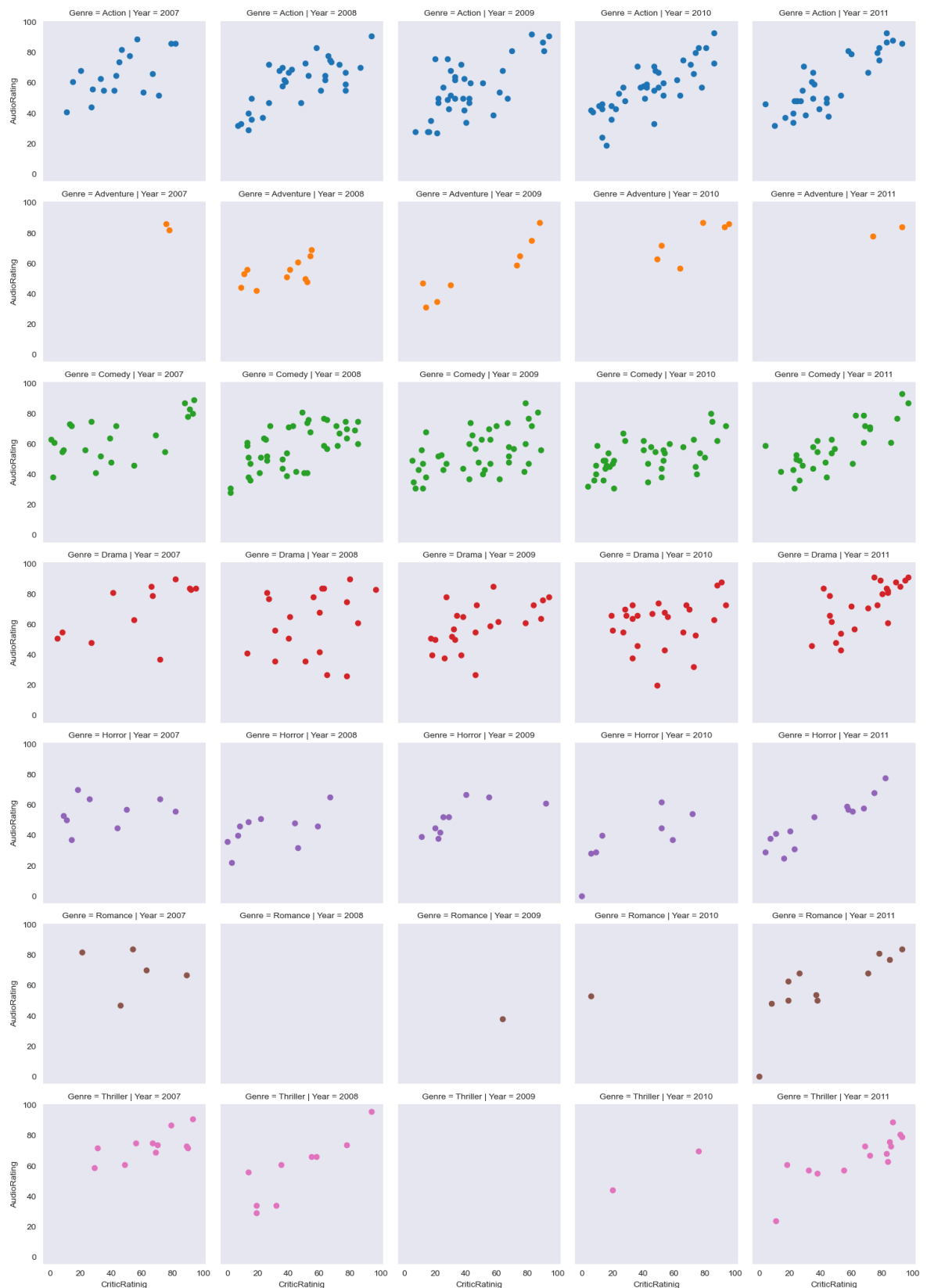
```
In [118... plt.scatter(moive.CriticRatinig,moive.AudioRating)
```

```
Out[118... <matplotlib.collections.PathCollection at 0x1a00472a120>
```



In [120...

```
g = sns.FacetGrid (moive,row = 'Genre', col = 'Year', hue='Genre')  
g = g.map(plt.scatter,'CriticRatinig','AudioRating')  
#scatterplots are mapped in facetgrid
```



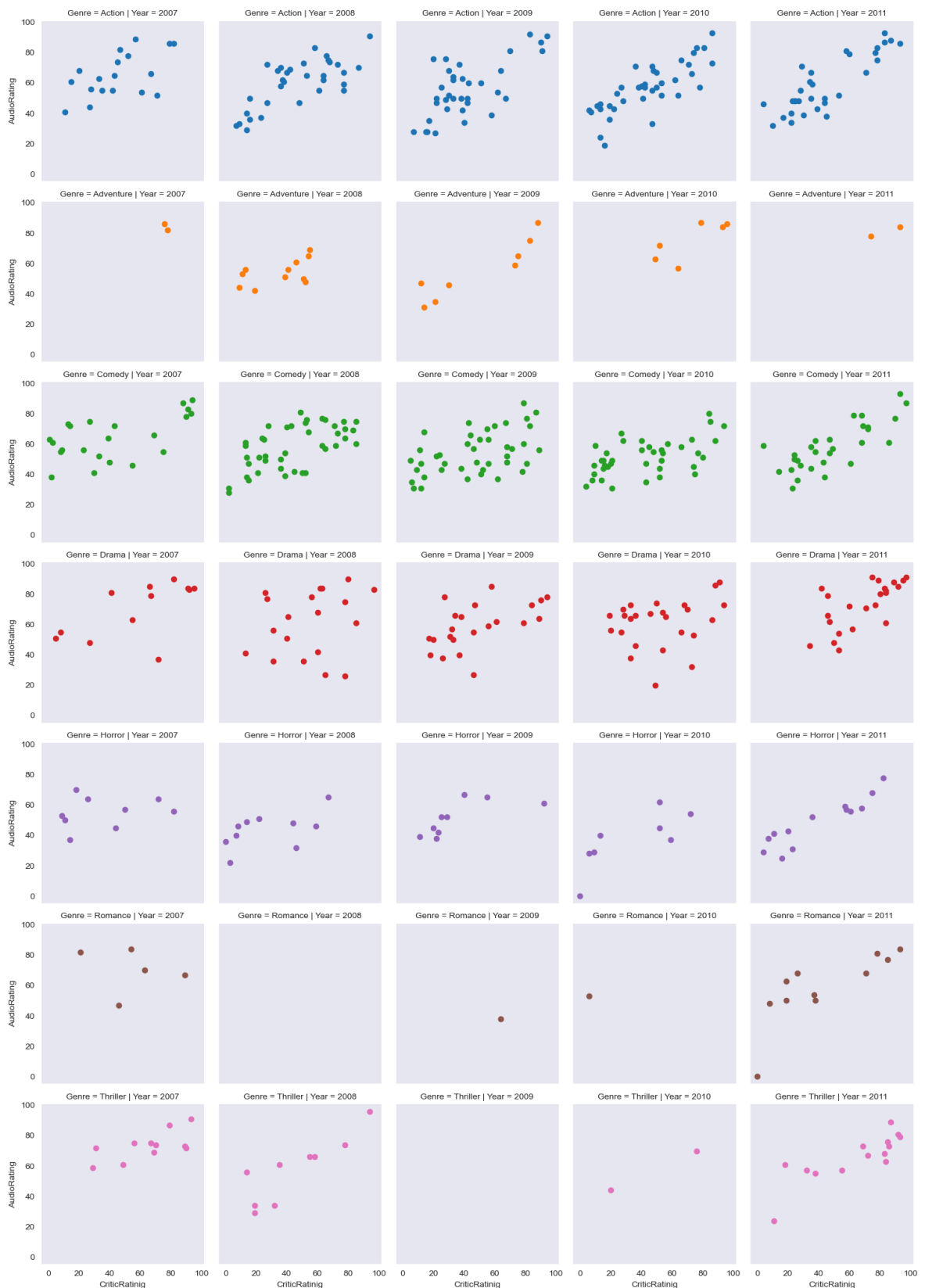
In [121... *# you can populated any type of chat.*

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



In [140...

```
g = sns.FacetGrid(moive,row = 'Genre',col = 'Year',hue = 'Genre')
kws = dict(s=50,linewidth=0.5,edgecolor='black')
g = g.map(plt.scatter,'CriticRatinig','AudioRating')
#scatterplots are mapped in facetgrid
```



In [142...

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

In [154...

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

#k1 = sns.kdeplot(moive.BudgetMillions,moive.AudioRating,ax=axes[0,0])
#k2 = sns.kdeplot(moive.BudgetMillions,moive.CriticRating,ax = axes[0,1])

k1 = sns.kdeplot(data=moive,x=moive.BudgetMillions,y=moive.AudioRating,ax=axes[0,0])
```

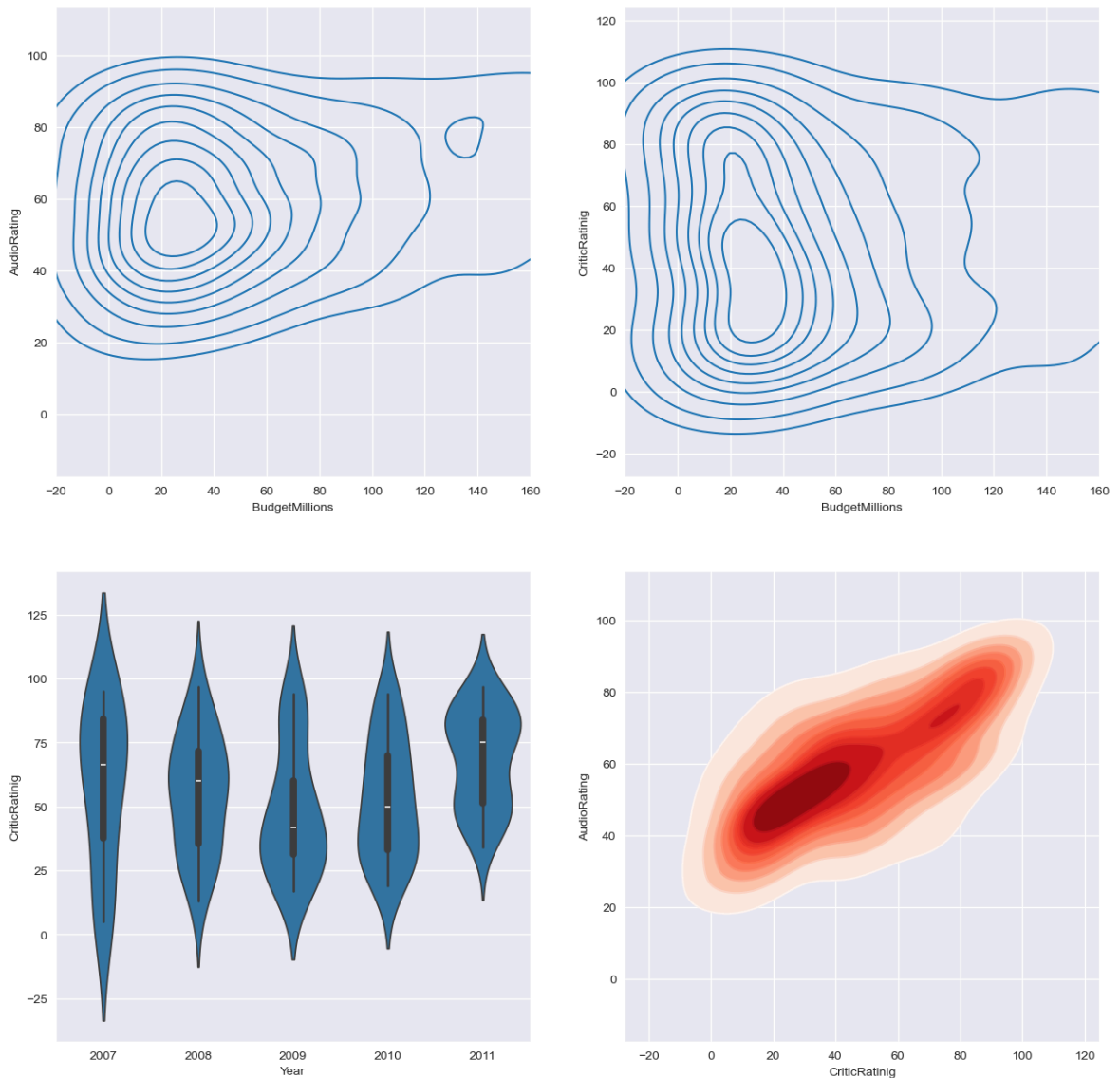
```

k2 = sns.kdeplot(x=moive.BudgetMillions,y=moive.CriticRatinig,ax=axes[0,1])

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

z = sns.violinplot(data=moive[moive.Genre=='Drama'], x='Year', y = 'CriticRatini
k4 = sns.kdeplot(x=moive.CriticRatinig,y=moive.AudioRating,shade = True,shade_lo
k4b = sns.kdeplot(x=moive.CriticRatinig, y=moive.AudioRating,cmap='Reds',ax = ax
plt.show()

```



In [156...

```
# 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(moive.BudgetMillions,moive.AudioRating, \
```

```

        shade = True, shade_lowest=True, cmp = 'inferno', \
        ax = axes[0,0])
k1b = sns.kdeplot(moive.BudgetMillions, moive.AudioRating, \
        cmap = 'cool', ax = axes[0,0])

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

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

#plot[1,1]
k4 = sns.kdeplot(moive.CriticRatinig, moive.AudioRating, \
        shade = True, shade_lowest=False, cmap='Blues_r', \
        ax=axes[1,1])

k4b = sns.kdeplot(moive.CriticRatinig, moive.AudioRating, \
        cmap='gist_gray_r', ax = axes[1,1])

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

plt.show()

```

TypeError

Traceback (most recent call last)

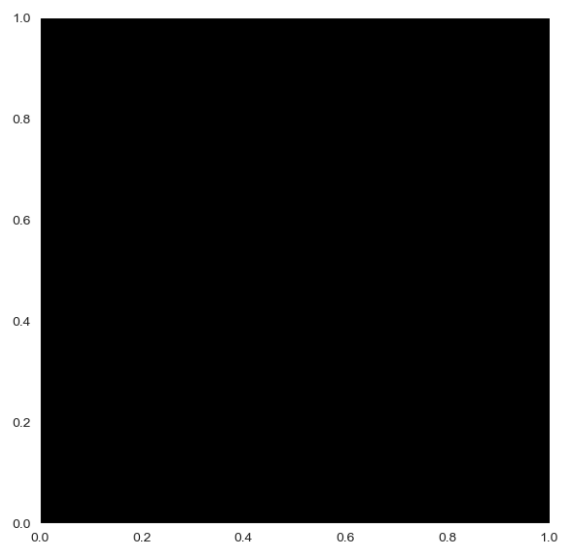
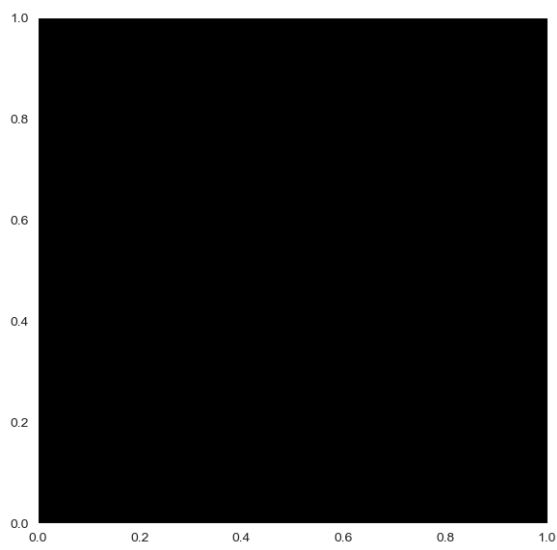
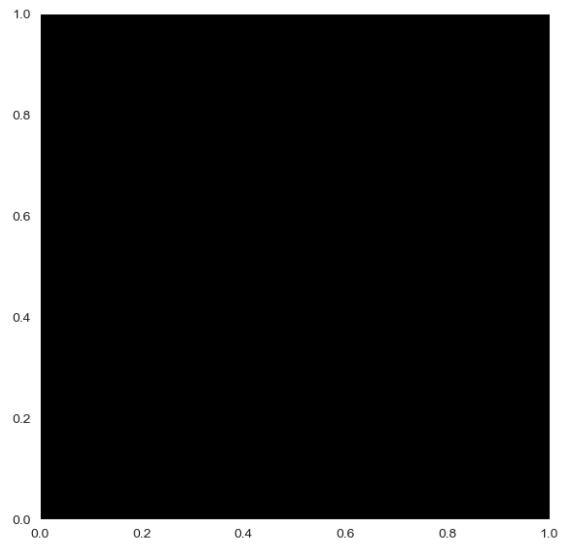
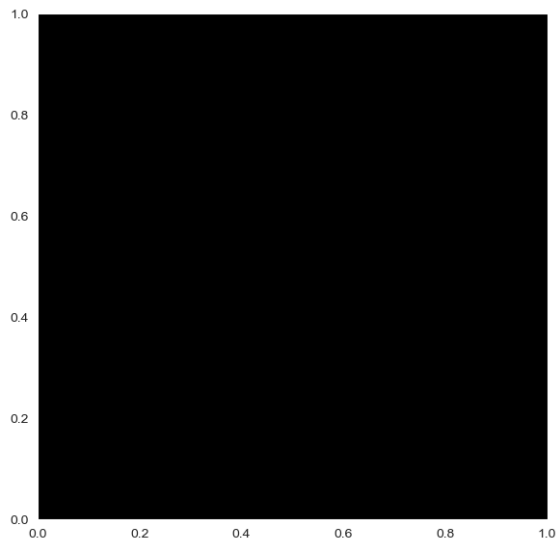
Cell In[156], line 10

```

7 f, axes = plt.subplots (2,2, figsize = (15,15))
9 #plot [0,0]
--> 10 k1 = sns.kdeplot(moive.BudgetMillions, moive.AudioRating, \
11                     shade = True, shade_lowest=True, cmp = 'inferno', \
12                     ax = axes[0,0])
13 k1b = sns.kdeplot(moive.BudgetMillions, moive.AudioRating, \
14                   cmap = 'cool', ax = axes[0,0])
16 #plot [0,1]

```

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



In []: