```
import pandas as pd
import os

os.getcwd()   # if u want to change the working directory

'C:\\Users\\Admin'

movies=pd.read_csv(r"C:\Users\Admin\Downloads\Movie-Rating.csv")

movies

                      Film      Genre  Rotten Tomatoes Ratings %  \
0    (500) Days of Summer      Comedy                         87
1              10,000 B.C.  Adventure                          9
2               12 Rounds      Action                         30
3               127 Hours  Adventure                         93
4                17 Again      Comedy                         55
..                     ...        ...                        ...
554           Your Highness      Comedy                         26
555         Youth in Revolt      Comedy                         68
556                  Zodiac   Thriller                         89
557             Zombieland      Action                         90
558               Zookeeper      Comedy                         14

     Audience Ratings %  Budget (million $)  Year of release
0                    81                   8             2009
1                    44                 105             2008
2                    52                  20             2009
3                    84                  18             2010
4                    70                  20             2009
..                  ...                 ...              ...
554                  36                  50             2011
555                  52                  18             2009
556                  73                  65             2007
557                  87                  24             2009
558                  42                  80             2011

[559 rows x 6 columns]

len(movies)

559

movies.columns

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

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   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

movies.head()   # Removed spaces & % removed noise characters

                    Film      Genre  Rotten Tomatoes Ratings %  \
0  (500) Days of Summer      Comedy                         87   
1            10,000 B.C.  Adventure                          9   
2             12 Rounds      Action                         30   
3              127 Hours  Adventure                         93   
4              17 Again      Comedy                         55   

   Audience Ratings %  Budget (million $)  Year of release  
0                  81                   8             2009  
1                  44                 105             2008  
2                  52                  20             2009  
3                  84                  18             2010  
4                  70                  20             2009  

movies.tail()

                Film     Genre  Rotten Tomatoes Ratings %  Audience 
Ratings %  \
554    Your Highness    Comedy                         26  
36   
555  Youth in Revolt    Comedy                         68  
52   
556           Zodiac  Thriller                         89  
73   
557       Zombieland    Action                         90  
87   
558        Zookeeper    Comedy                         14  
42   

     Budget (million $)  Year of release  
554                  50             2011  
555                  18             2009  
556                  65             2007  
```

```
557                     24             2009
558                     80             2011

movies.columns=['Film', 'Genre', 'CriticRating',
'AudienceRatings','BudgetMillions','Year']

movies.columns

Index(['Film', 'Genre', 'CriticRating', 'AudienceRatings',
'BudgetMillions',
       'Year'],
      dtype='object')

movies.head()
# Removed spaces & % removed noise characters

                    Film      Genre  CriticRating  AudienceRatings  \
0  (500) Days of Summer     Comedy            87               81
1           10,000 B.C.  Adventure             9               44
2            12 Rounds     Action            30               52
3             127 Hours  Adventure            93               84
4             17 Again     Comedy            55               70

   BudgetMillions  Year
0               8  2009
1             105  2008
2              20  2009
3              18  2010
4              20  2009

movies.describe()
# 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 categroy type
# also from object datatype we will convert to category datatypes

       CriticRating  AudienceRatings  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

movies['Film']
#movies['Audience Ratings %']

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
```

```
movies.Film=movies.Film.astype('category')
```

```
movies.Film
```

```
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']
```

```
movies.head()
```

```
                    Film      Genre  CriticRating  AudienceRatings  \
0  (500) Days of Summer      Comedy            87               81
1            10,000 B.C.  Adventure             9               44
2             12 Rounds      Action            30               52
3              127 Hours  Adventure            93               84
4              17 Again      Comedy            55               70

   BudgetMillions  Year
0               8  2009
1             105  2008
2              20  2009
3              18  2010
4              20  2009
```

```
movies.info()
# now the same thing we will change genra to category & year to
category
```

```
<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   CriticRating      559 non-null    int64   
 3   AudienceRatings   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
```

```
movies.Genre=movies.Genre.astype('category')
movies.Year=movies.Year.astype('category')
```

```
movies.Genre
```

```
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']
```

```
movies.Year   # is it real no. year you can take average,min,max but
out come have no meaning
```

```
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]
```

```
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    category
 1   Genre             559 non-null    category
 2   CriticRating      559 non-null    int64   
 3   AudienceRatings   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
```

```
movies.describe()
# now the same thing we will change genra to category & year to
category
```

```
       CriticRating  AudienceRatings  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
```

```
movies.Genre.cat.categories
```

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

```
print(movies.Genre)
print(movies.Year)
```

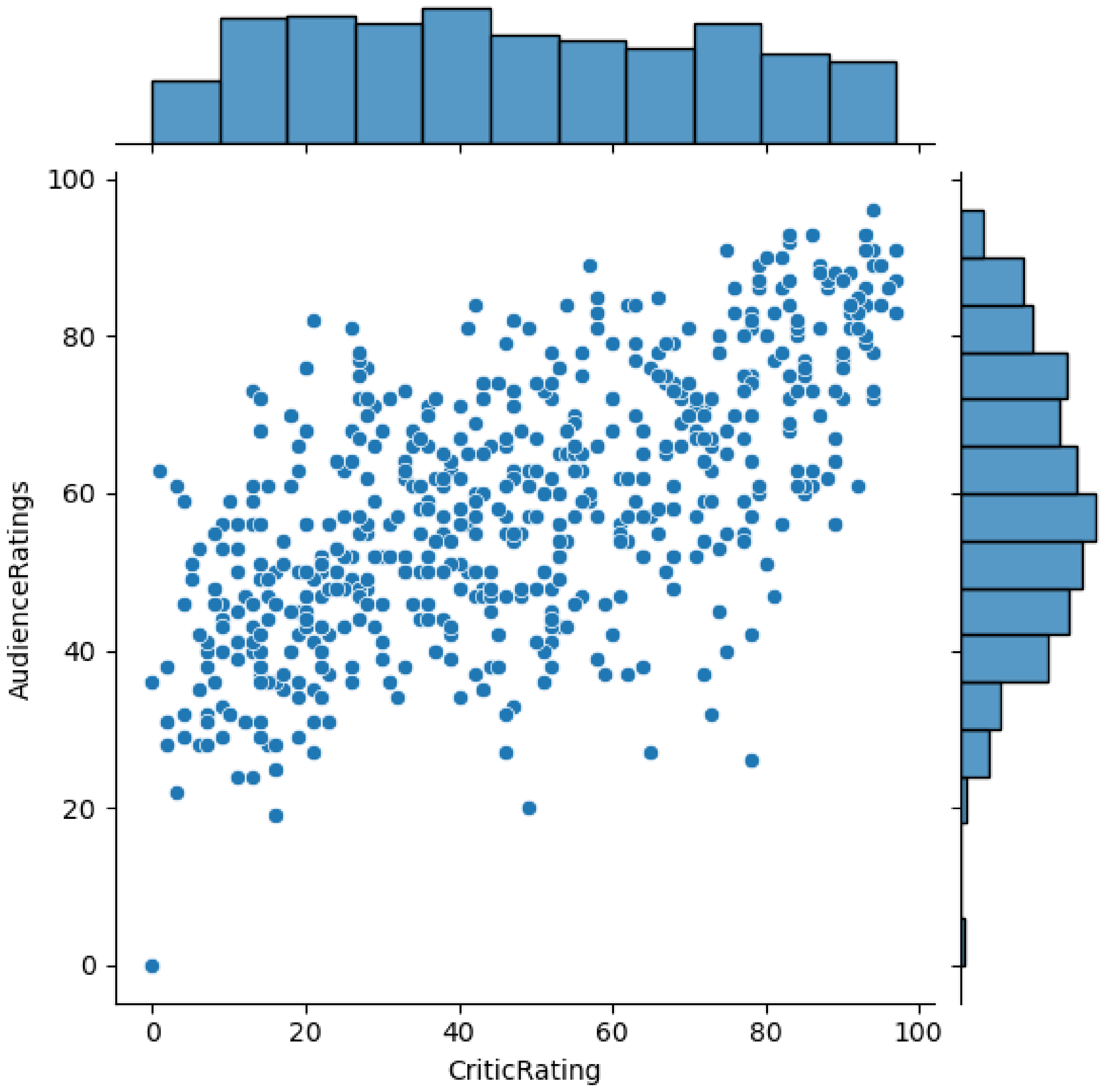
```
0         Comedy
1      Adventure
2         Action
3      Adventure
4         Comedy
         ...    
554       Comedy
555       Comedy
556     Thriller
557       Action
558       Comedy
Name: Genre, Length: 559, dtype: category
```

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

# How to working with joint plots

from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

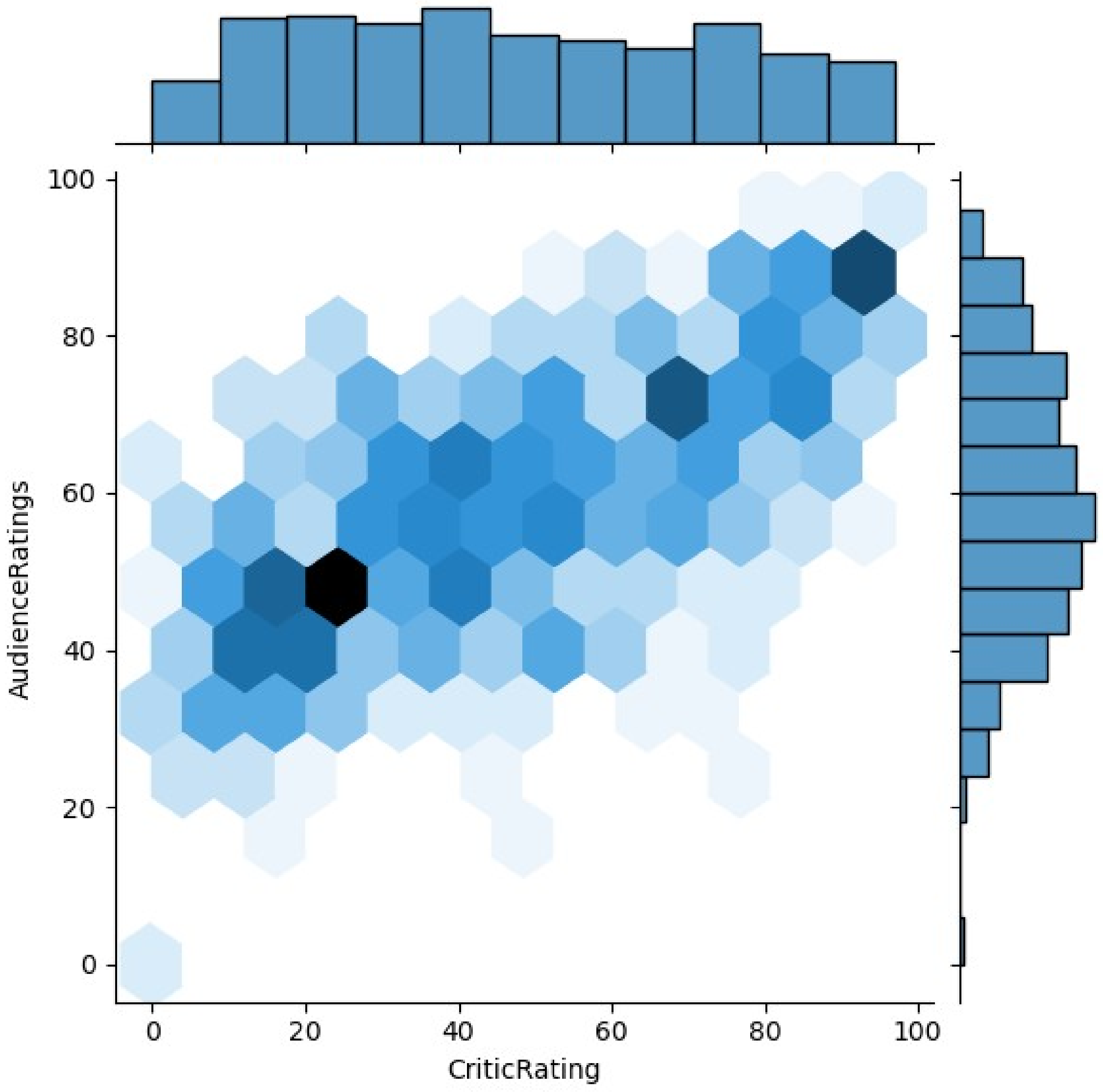
#basically joint plot is a scatter plot & it find the relation b/w audiene & critics

#also if you look up you can find the uniform distribution (critics)and normal distriution (audience)

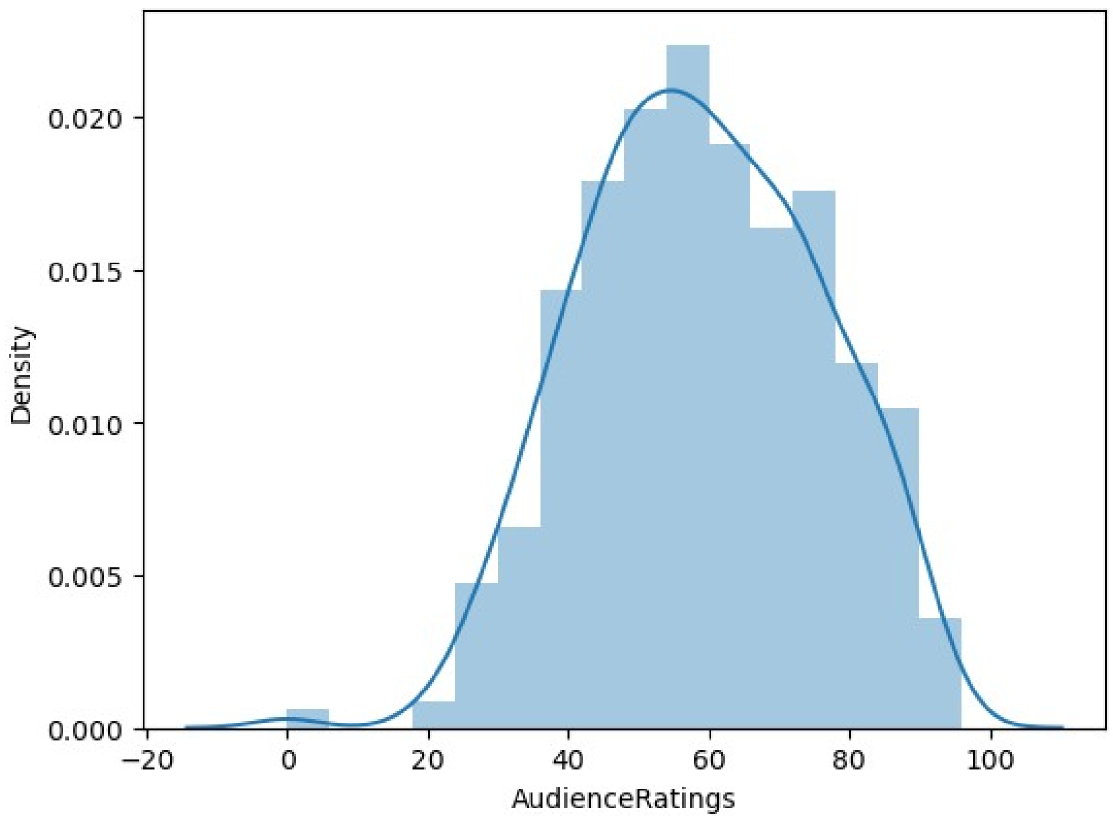
```
movies.describe()

       CriticRating  AudienceRatings  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

j = sns.jointplot( data = movies, x = 'CriticRating', y =
'AudienceRatings')
# 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 wathc critics rating
# let me explain the excel - if you filter audience rating & critic
rating. critic rating has very low values compare to audience rating

plt.show()
```



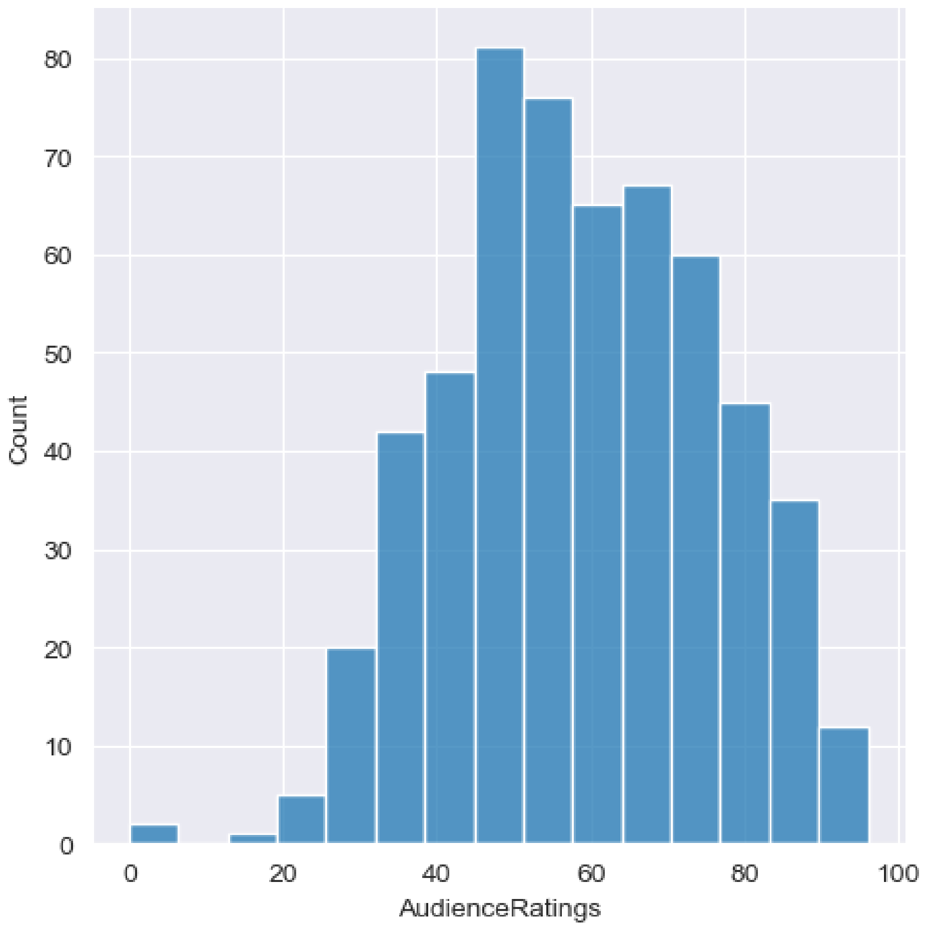
```
j = sns.jointplot( data = movies, x = 'CriticRating', y =
'AudienceRatings', kind='hex')
plt.show()
#j = sns.jointplot( data = movies, x = 'CriticRating', y =
'AudienceRating', kind='reg')
```



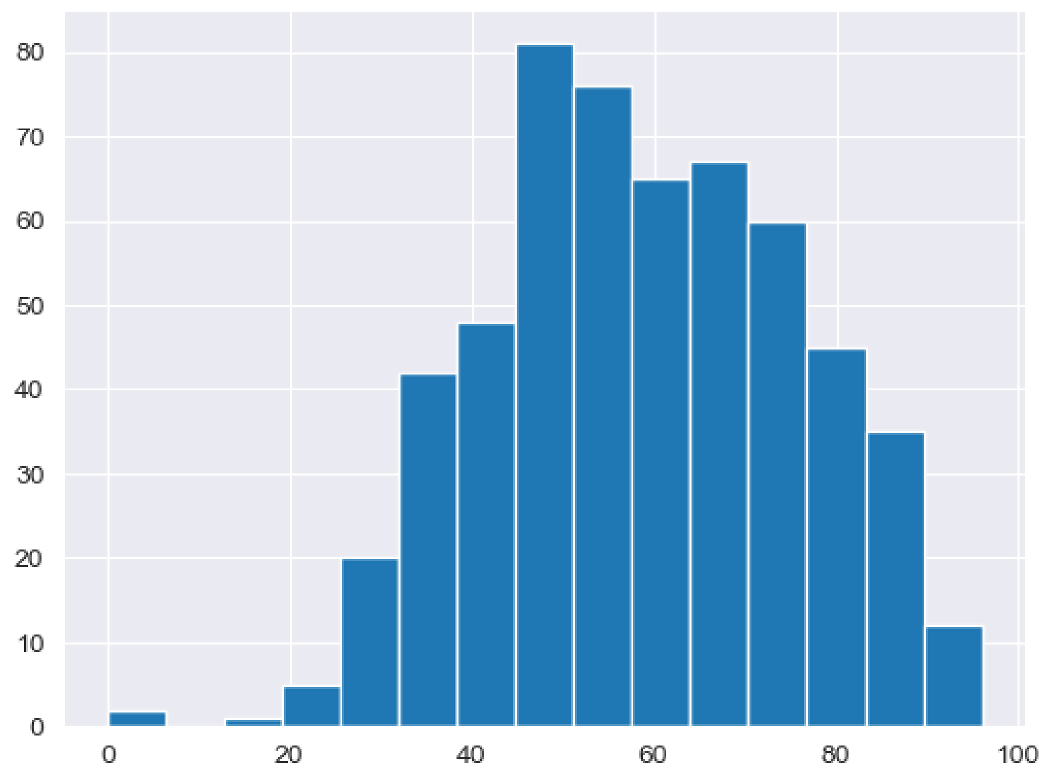
```
#Histograms

# <<< chat1

m1 = sns.distplot(movies.AudienceRatings)

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

plt.show()
```



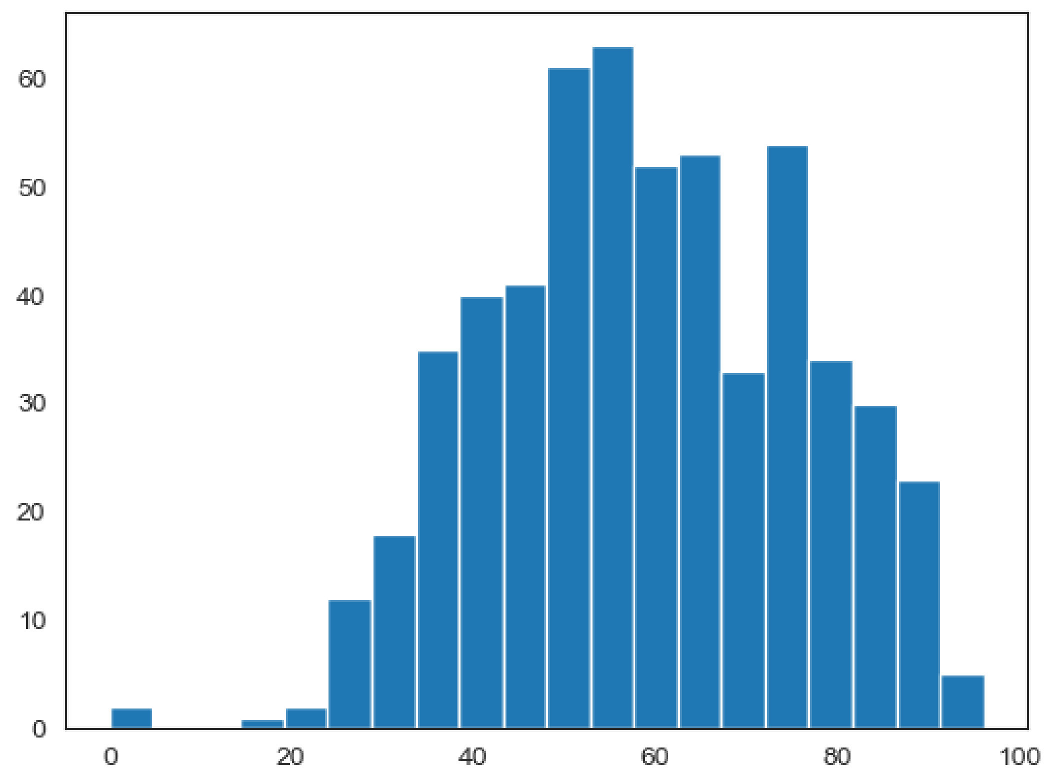
```
sns.set_style('darkgrid')

m2=sns.displot(movies.AudienceRatings, bins=15)
plt.show()
```



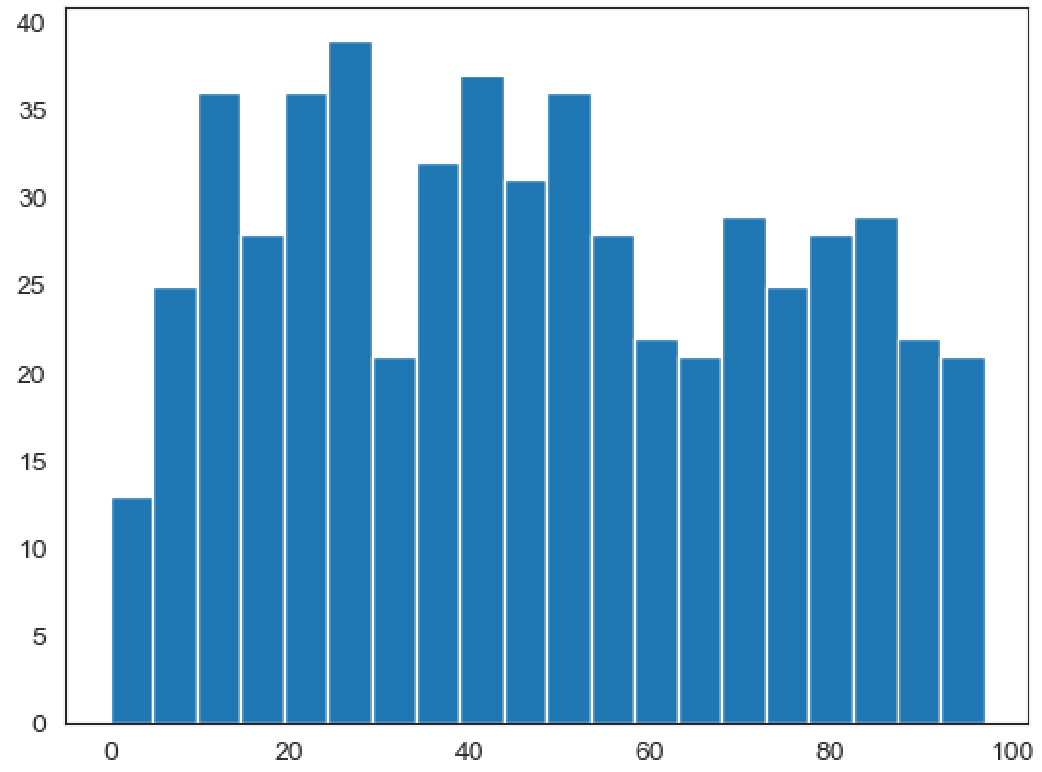
```python
#sns.set_style('darkgrid')
n1=plt.hist(movies.AudienceRatings, bins=15)
plt.show()
```



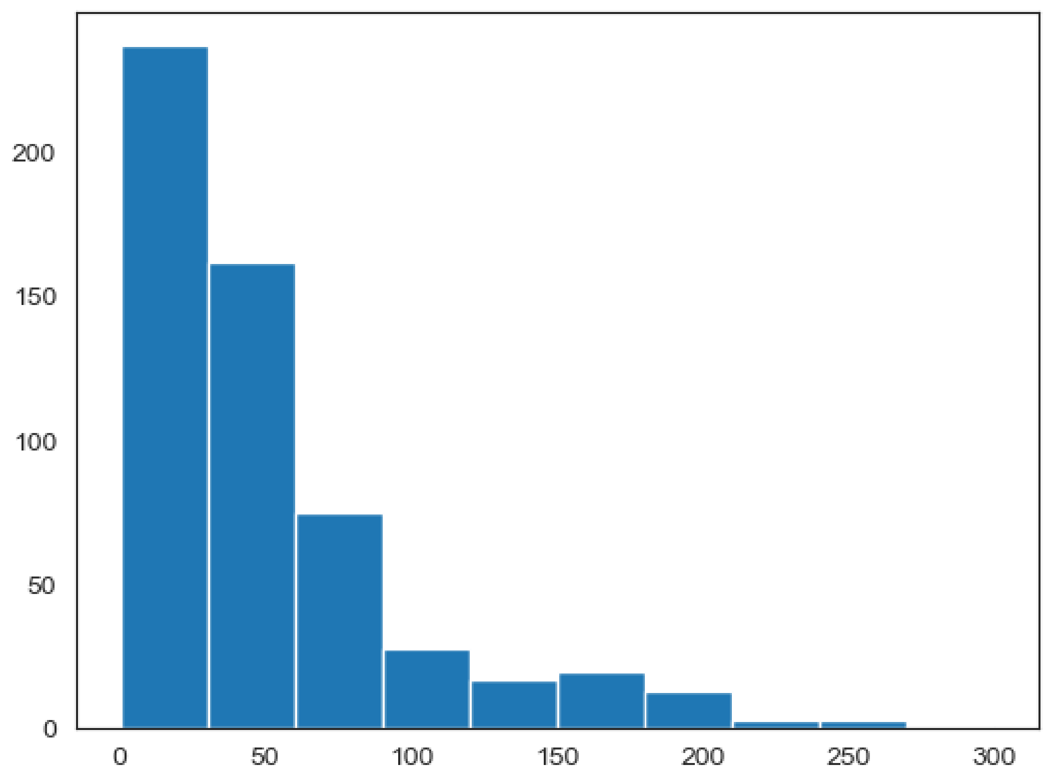
```
sns.set_style('white')  #normal distribution & called as bell curve
n1=plt.hist(movies.AudienceRatings, bins=20)

plt.show()
```



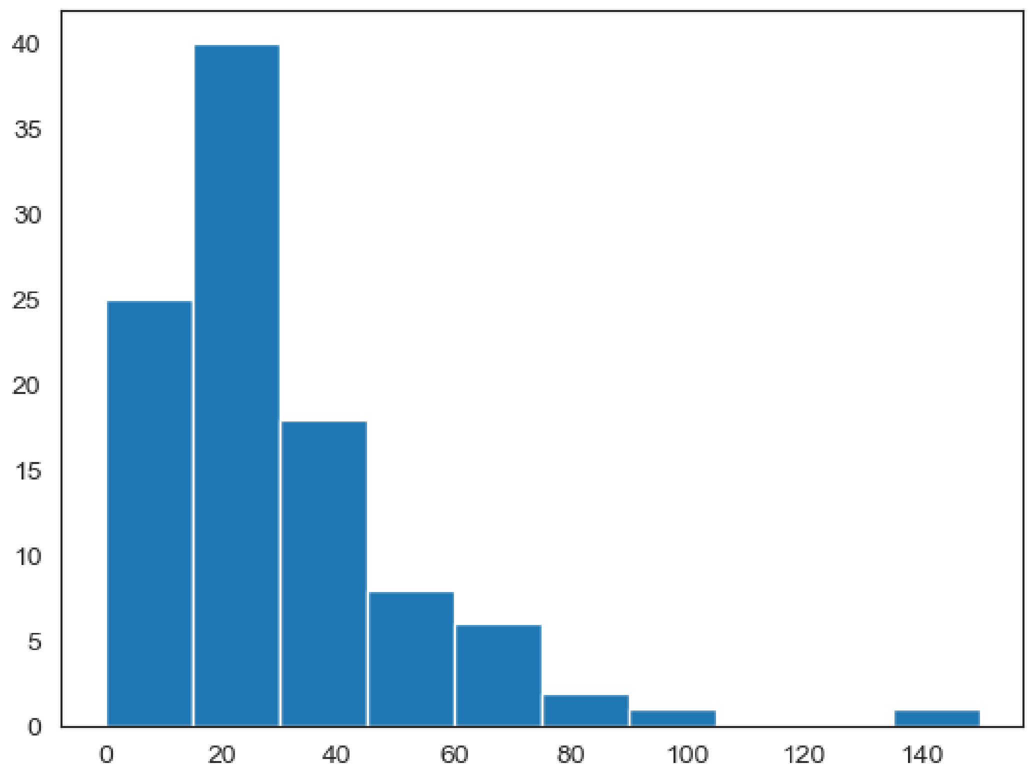
```
n1=plt.hist(movies.CriticRating, bins=20)  ##uniform distribution
plt.show()
```



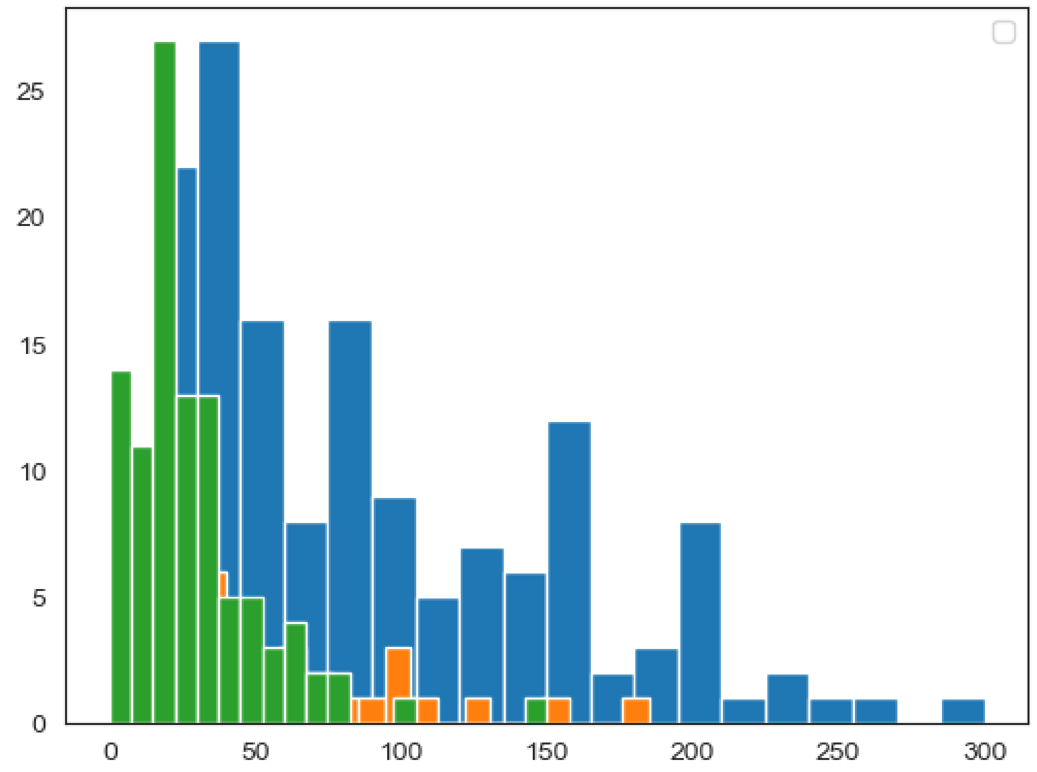
```
#h1 = plt.hist(movies.BudgetMillions)

plt.hist(movies.BudgetMillions)
plt.show()
```



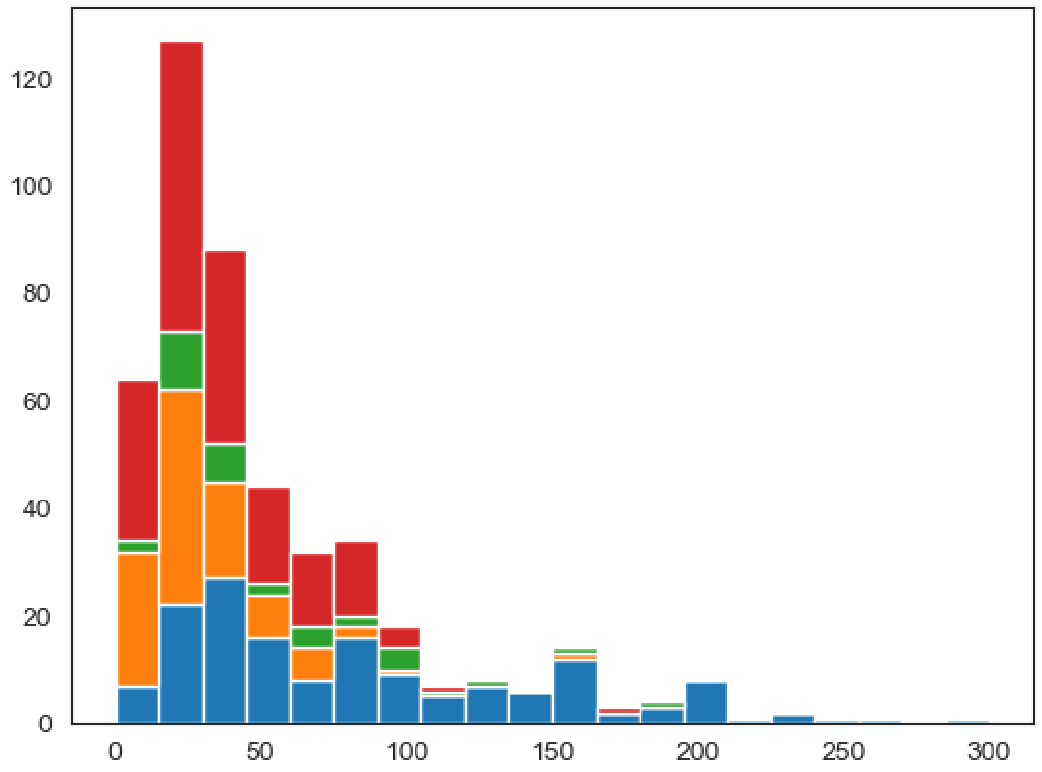
```
plt.hist(movies[movies.Genre == 'Drama'].BudgetMillions)
plt.show()
```



```
movies.head()

                   Film      Genre  CriticRating  AudienceRatings  \
0  (500) Days of Summer     Comedy            87               81   
1           10,000 B.C.  Adventure             9               44   
2            12 Rounds      Action            30               52   
3             127 Hours  Adventure            93               84   
4             17 Again      Comedy            55               70   

   BudgetMillions  Year  
0               8  2009  
1             105  2008  
2              20  2009  
3              18  2010  
4              20  2009  

#movies.Genre.unique()

# Below plots are stacked histogram becuase 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()
```



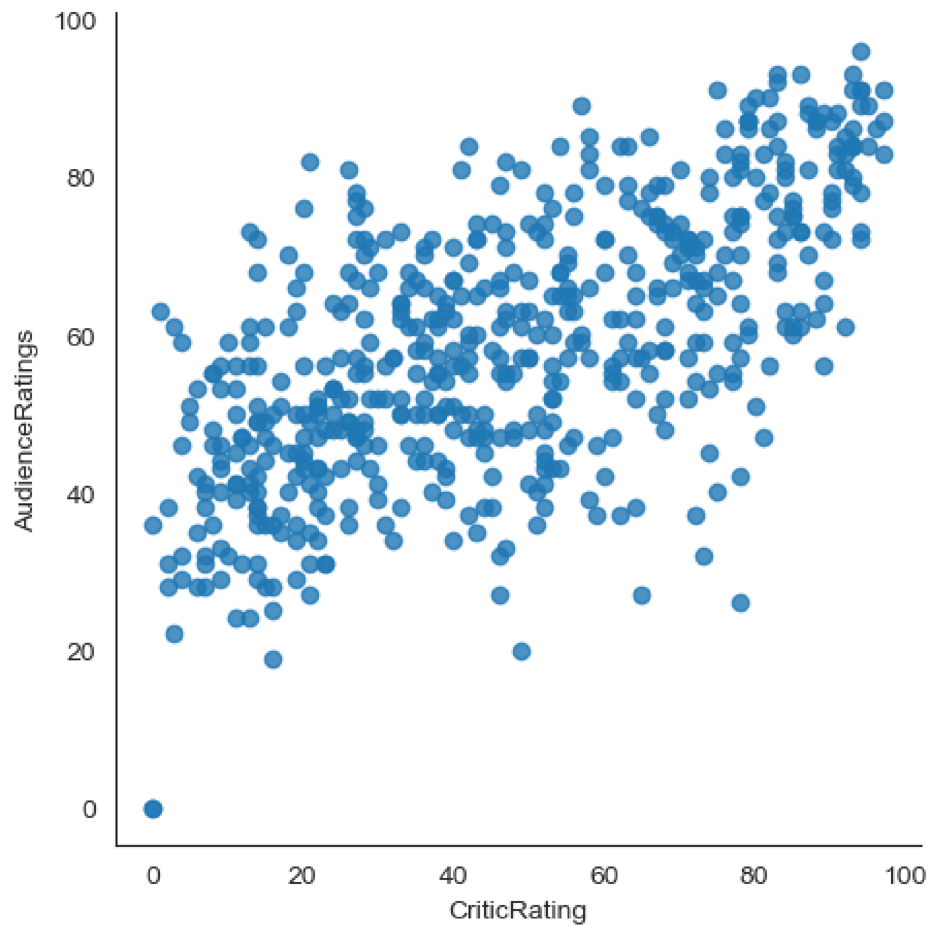
```
plt.hist([movies[movies.Genre == 'Action'].BudgetMillions,\
          movies[movies.Genre == 'Drama'].BudgetMillions, \
          movies[movies.Genre == 'Thriller'].BudgetMillions, \
          movies[movies.Genre == 'Comedy'].BudgetMillions],
         bins = 20, stacked = True)
plt.show()
```



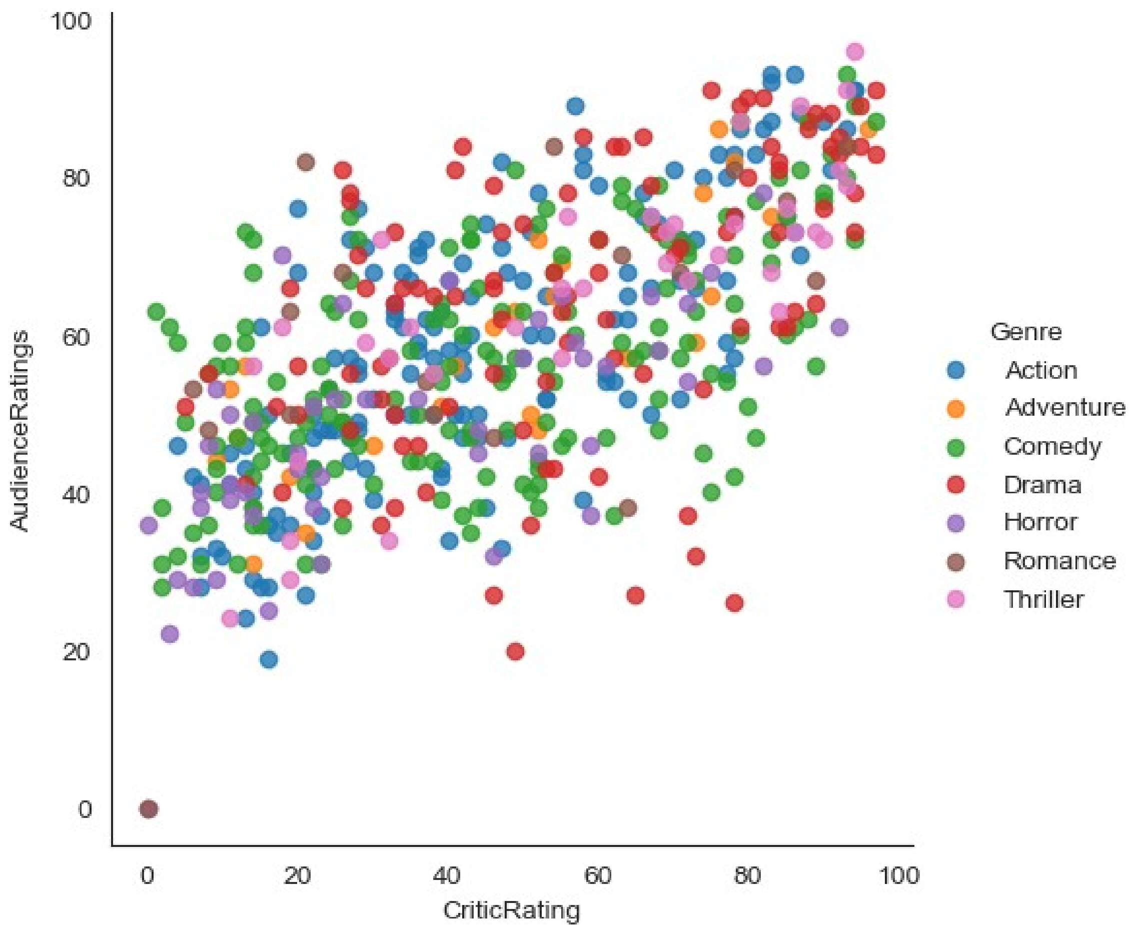
```python
# if you have 100 categories you cannot copy & paste all the things

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

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

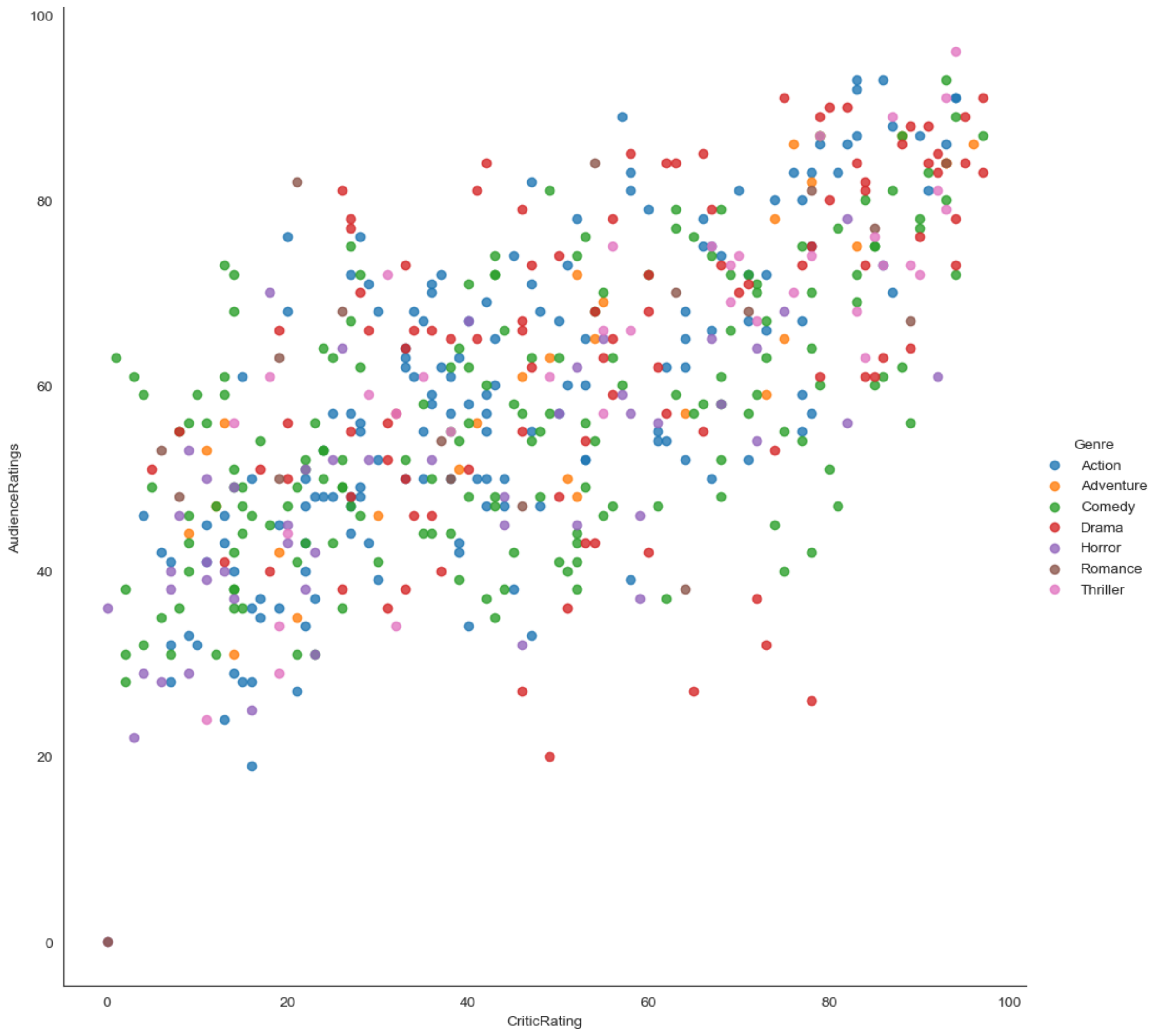
```python
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRatings',\
                  fit_reg=False)
plt.show(vis1)
```



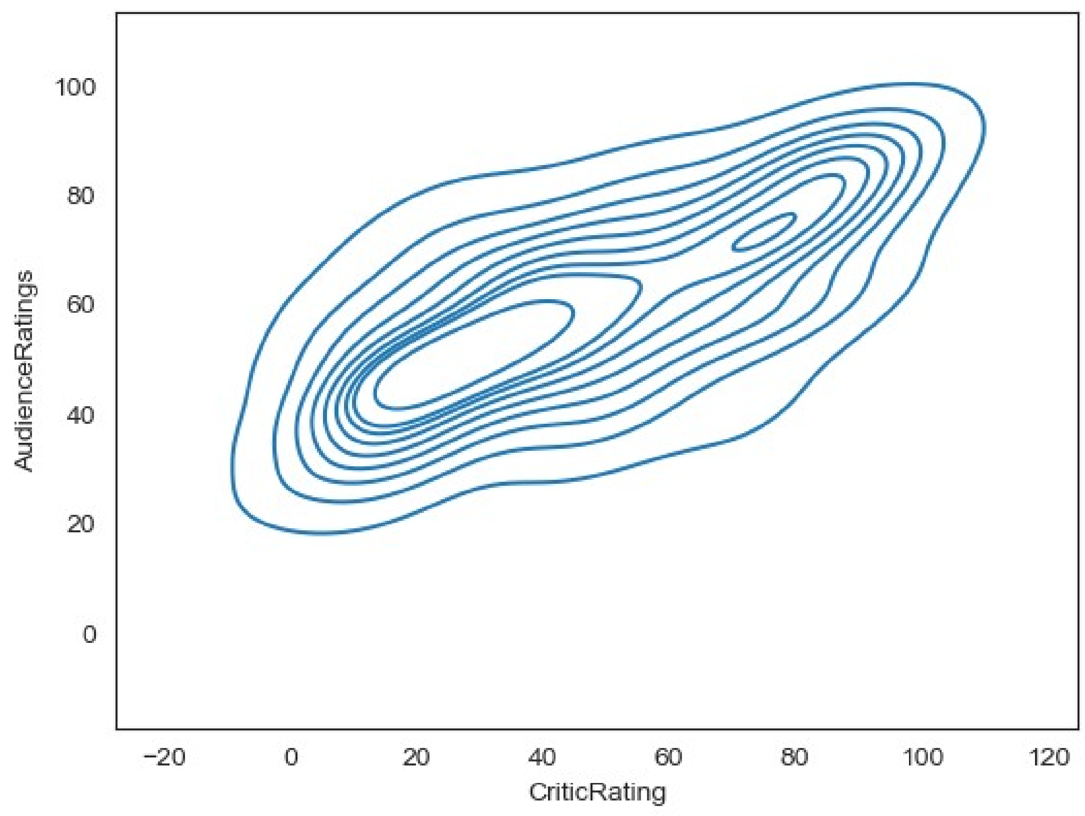
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRatings',\
                  fit_reg=False, hue = 'Genre')
plt.show(vis1)
```



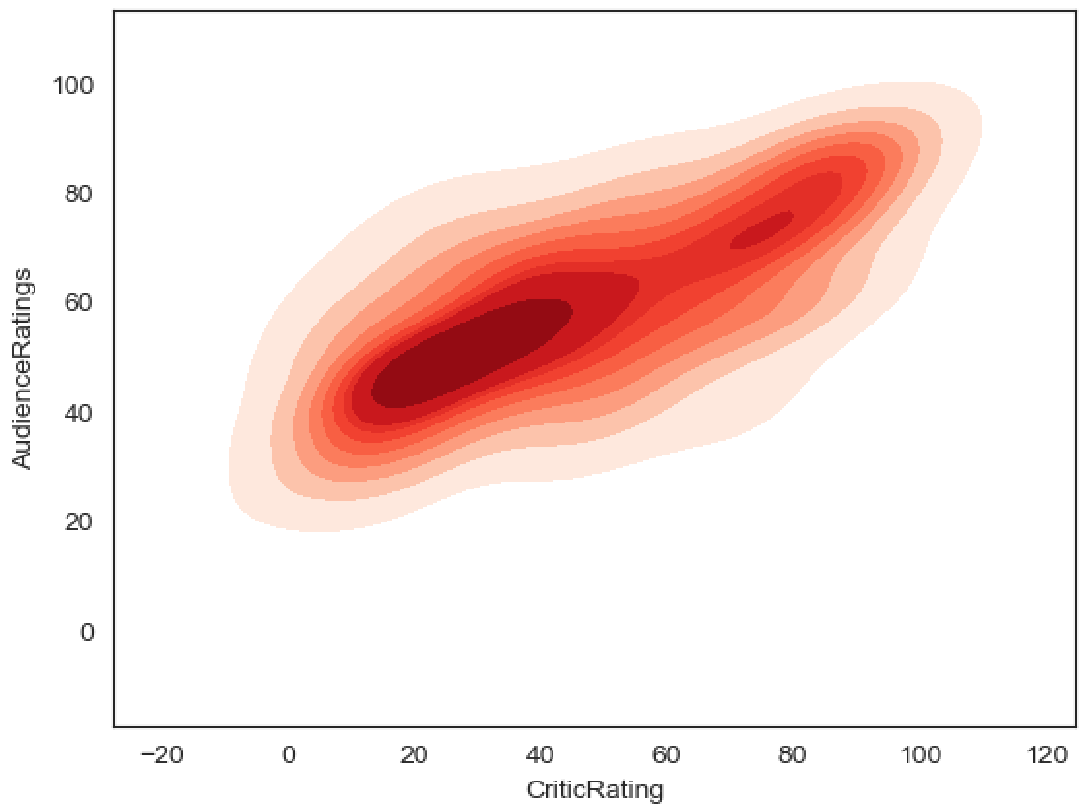
```
vis1 = sns.lmplot(data=movies, x='CriticRating', y='AudienceRatings',\
                    fit_reg=False, hue = 'Genre', height = 10,aspect=1)
plt.show(vis1)
```



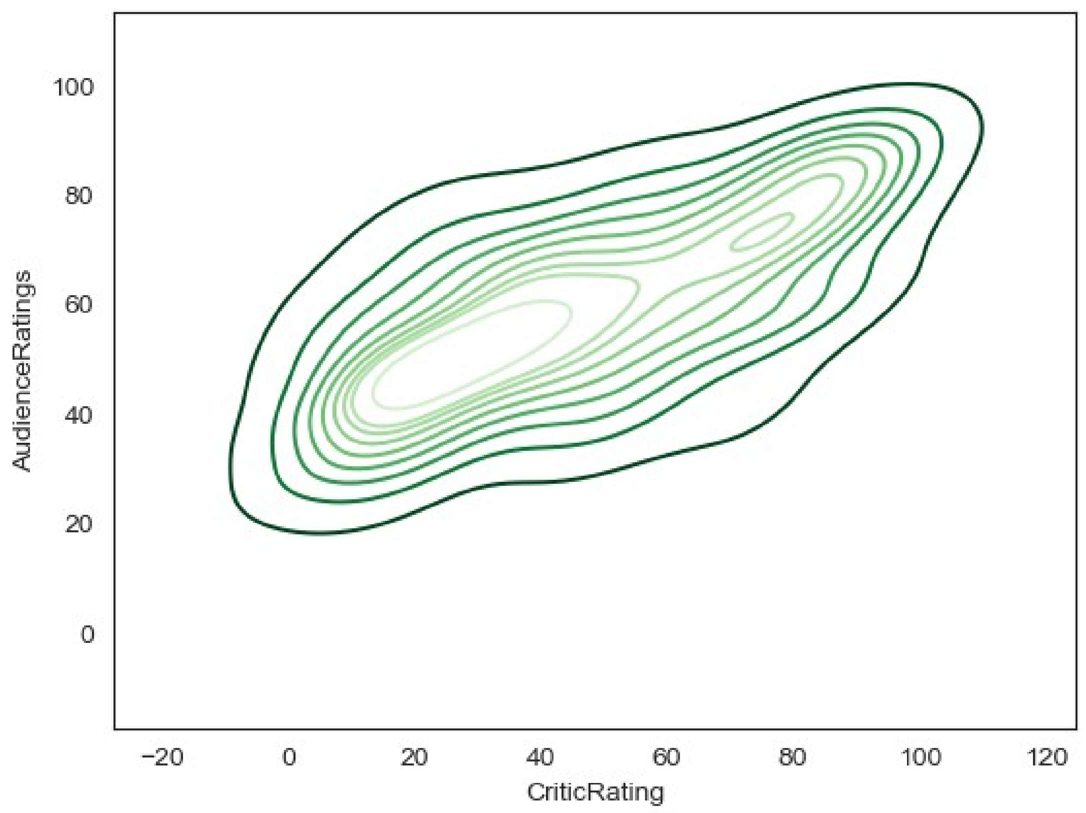
```
k1 = sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRatings)

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

plt.show()
```



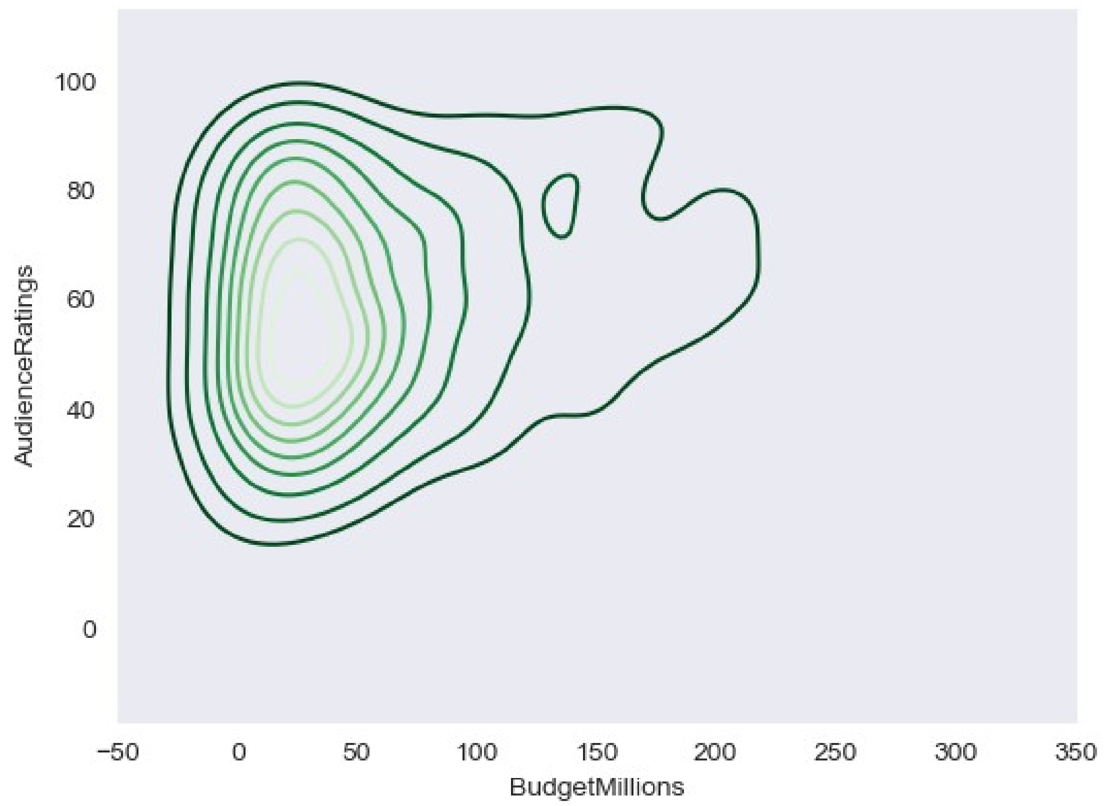
```
k1= sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRatings,shade =
True,shade_lowest=False,cmap='Reds')

plt.show()
```



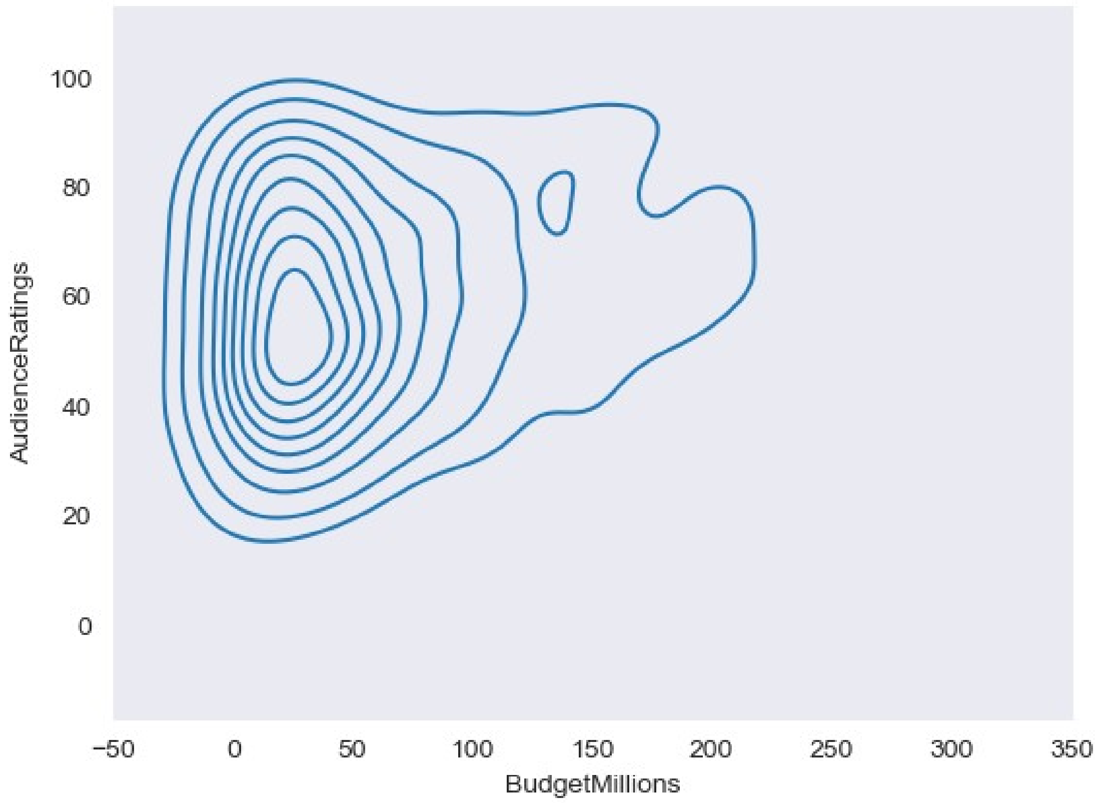
```
k2 =
sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRatings,shade_lowes
t=False,cmap='Greens_r')

plt.show()
```



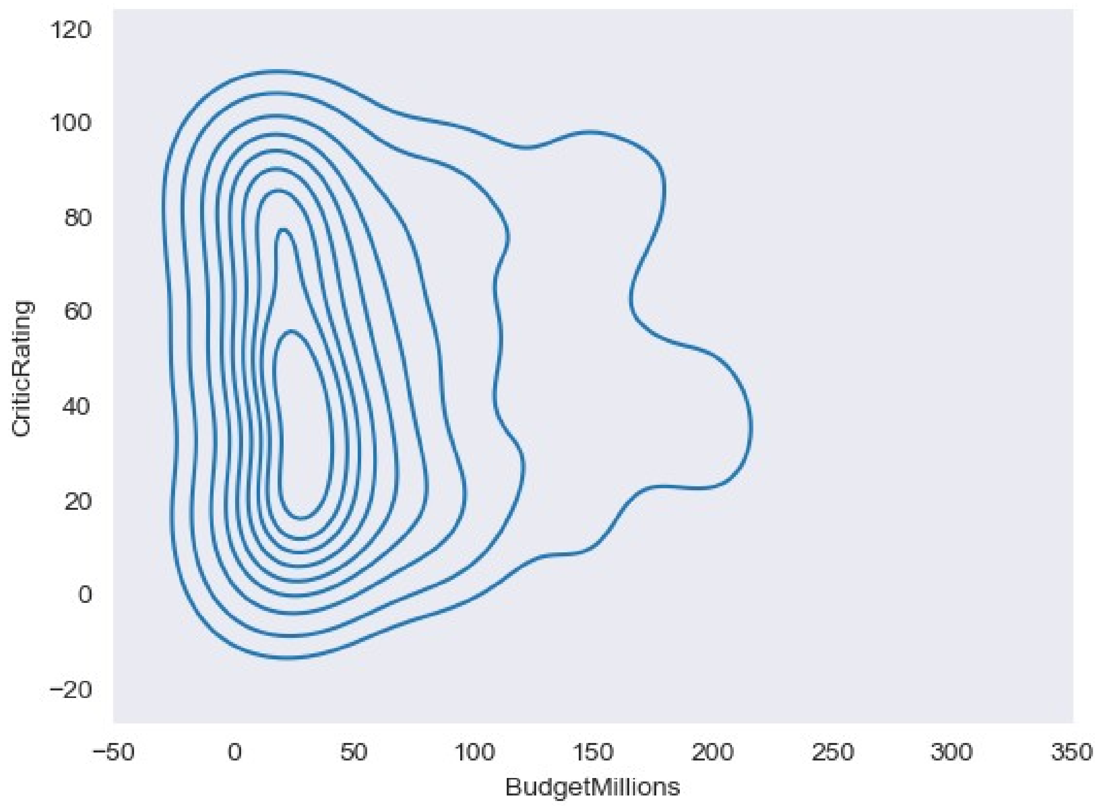
```
sns.set_style('dark')
k1 =
sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRatings,shade_low
est=False,cmap='Greens_r')

plt.show()
```



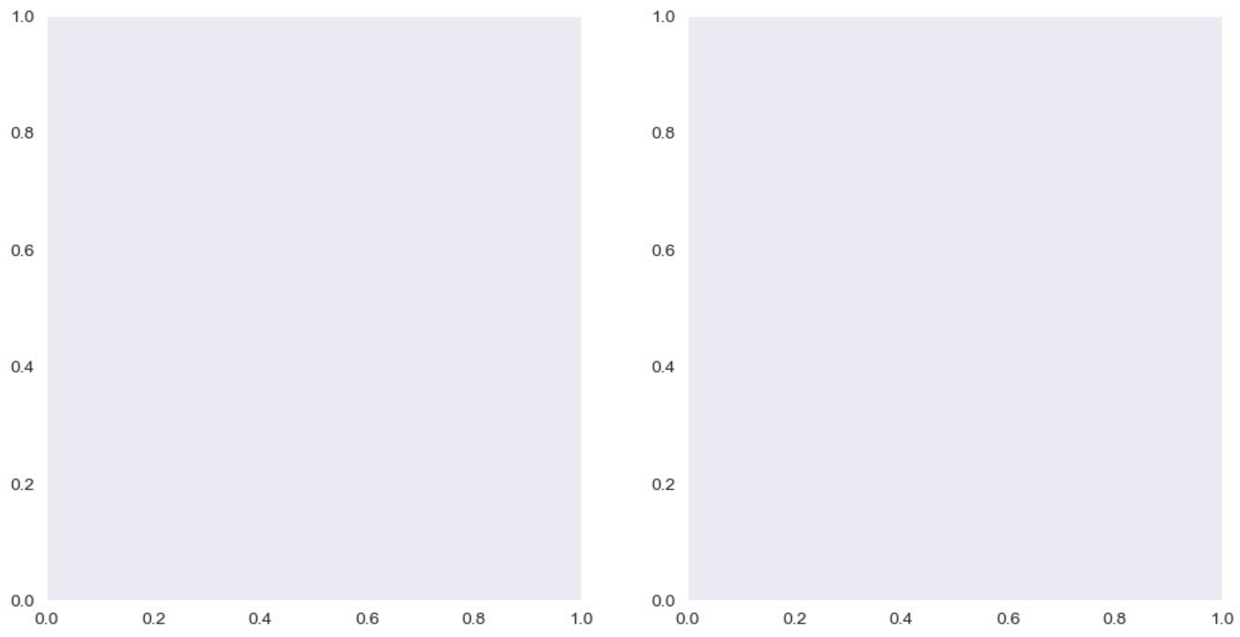
```
sns.set_style('dark')
k1 = sns.kdeplot(x=movies.BudgetMillions,y=movies.AudienceRatings)

plt.show()
```



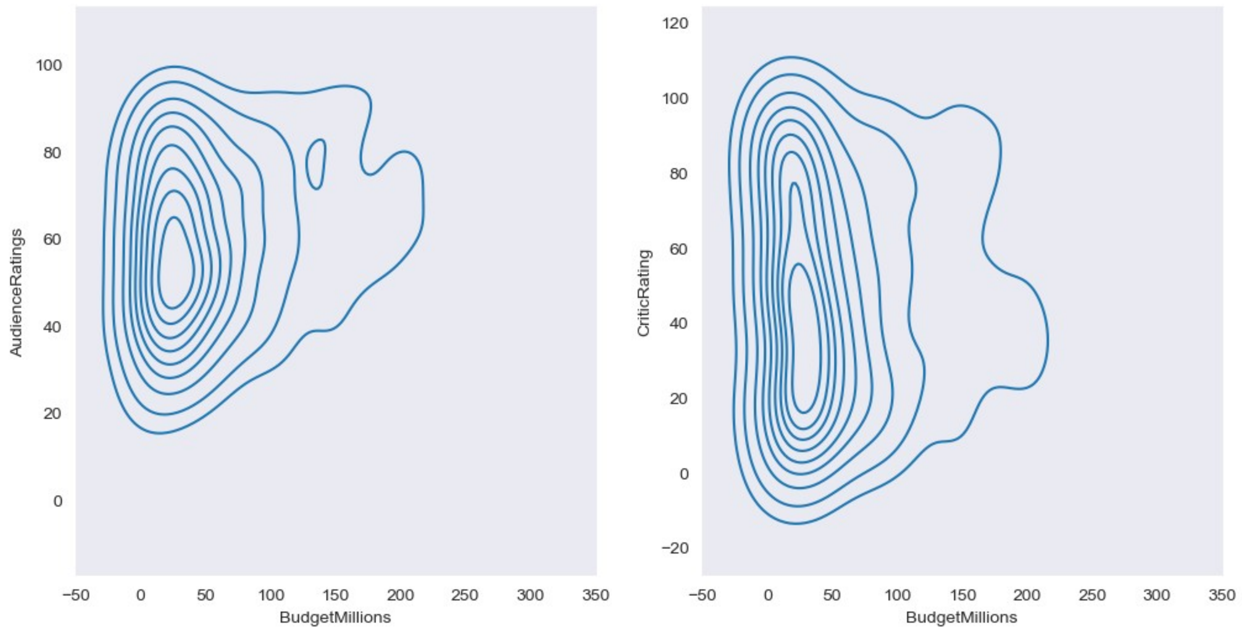
```
k2=sns.kdeplot(x=movies.BudgetMillions,y=movies.CriticRating)

plt.show()
```



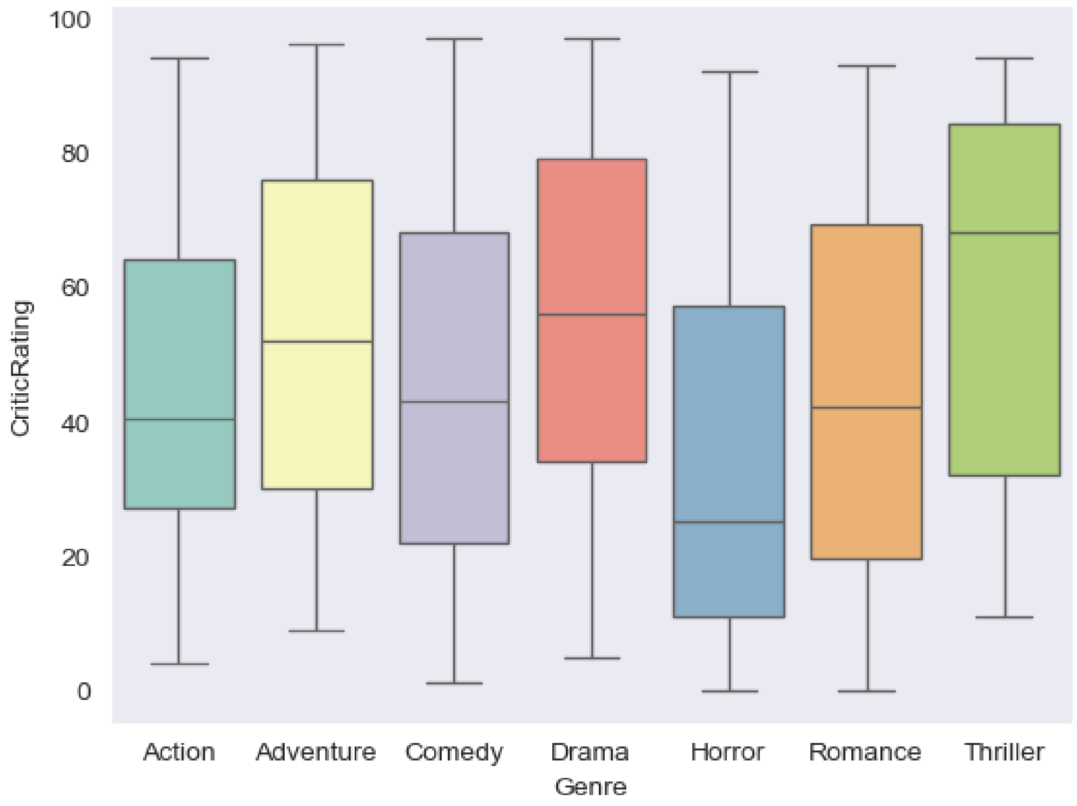
```
## subplots
f,ax=plt.subplots(1,2,figsize=(12,6))
#f,ax=plt.subplots(3,3,figsize=(12,6))

plt.show()
```



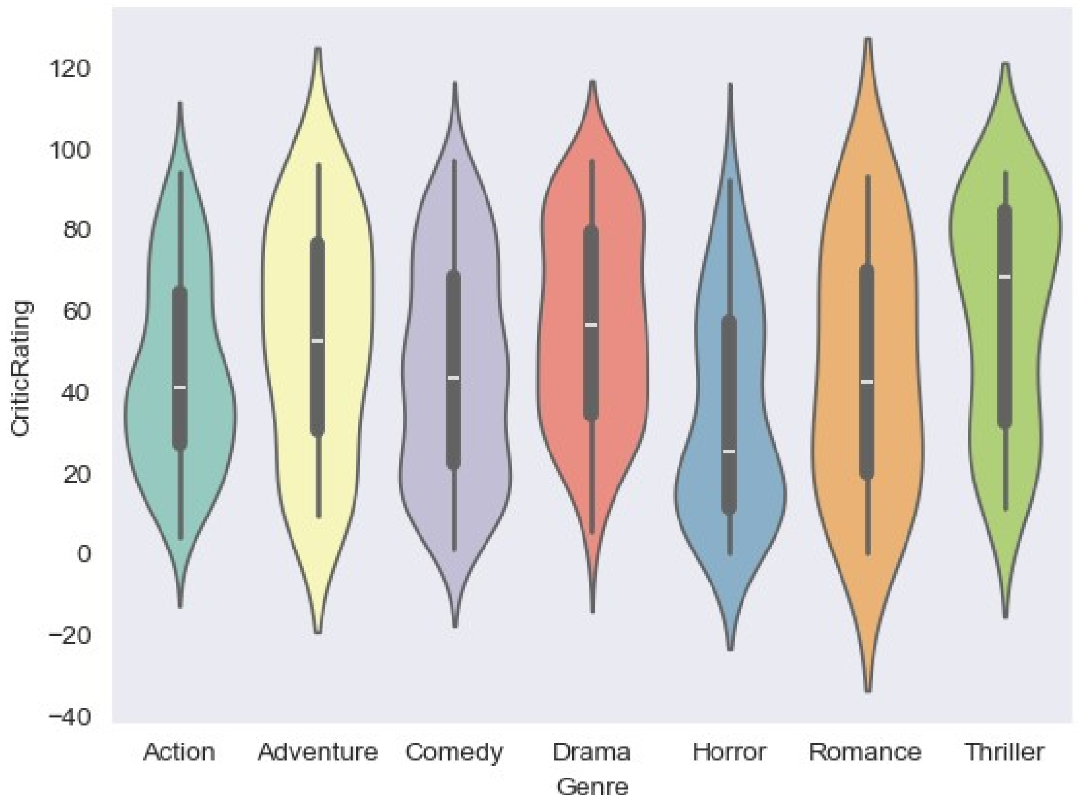
```
f, axes = plt.subplots(1,2, figsize =(12,6))

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

plt.show()
```



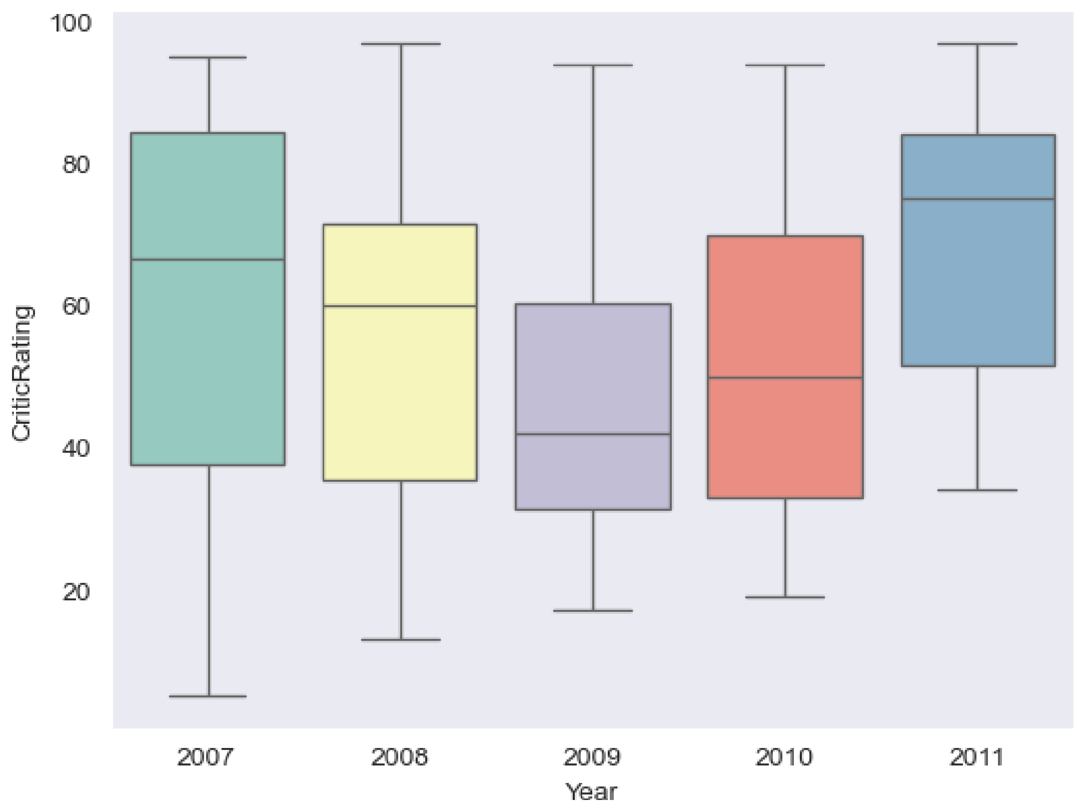
```
axes

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

#Box plots -

w = sns.boxplot(data=movies, x='Genre', y =
'CriticRating',palette='Set3')

plt.show()
```



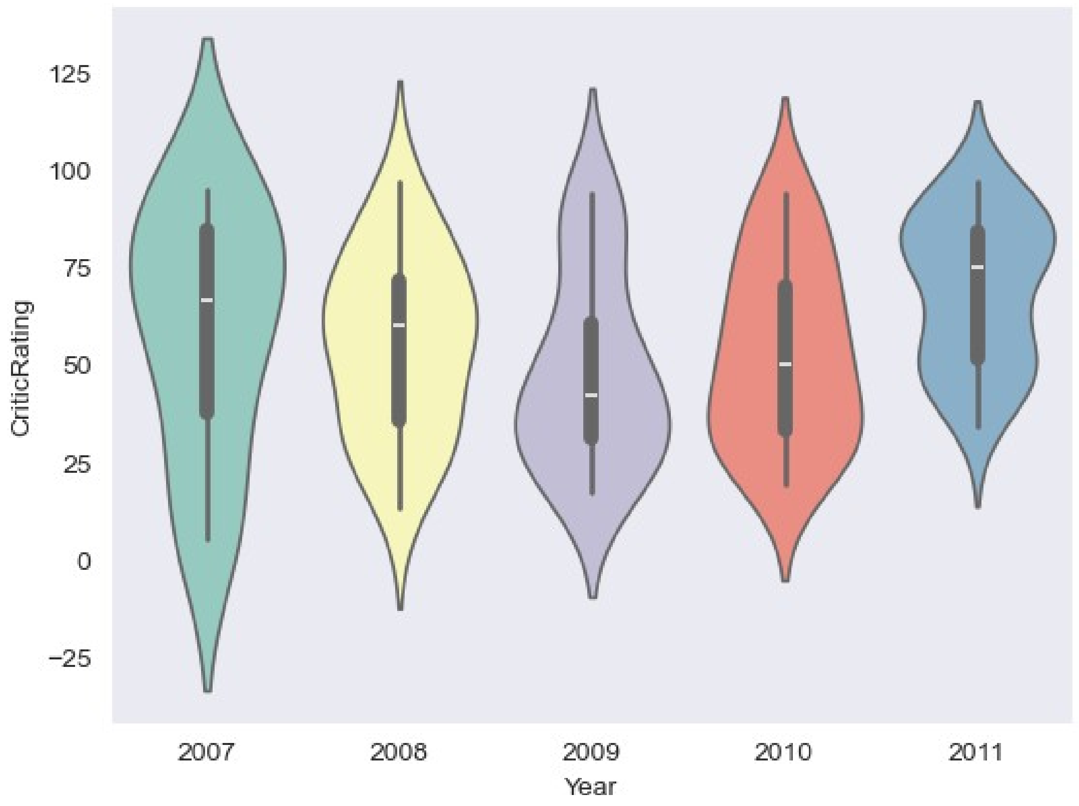
```
#violin plot

z = sns.violinplot(data=movies, x='Genre', y =
'CriticRating',palette='Set3')

plt.show()
```



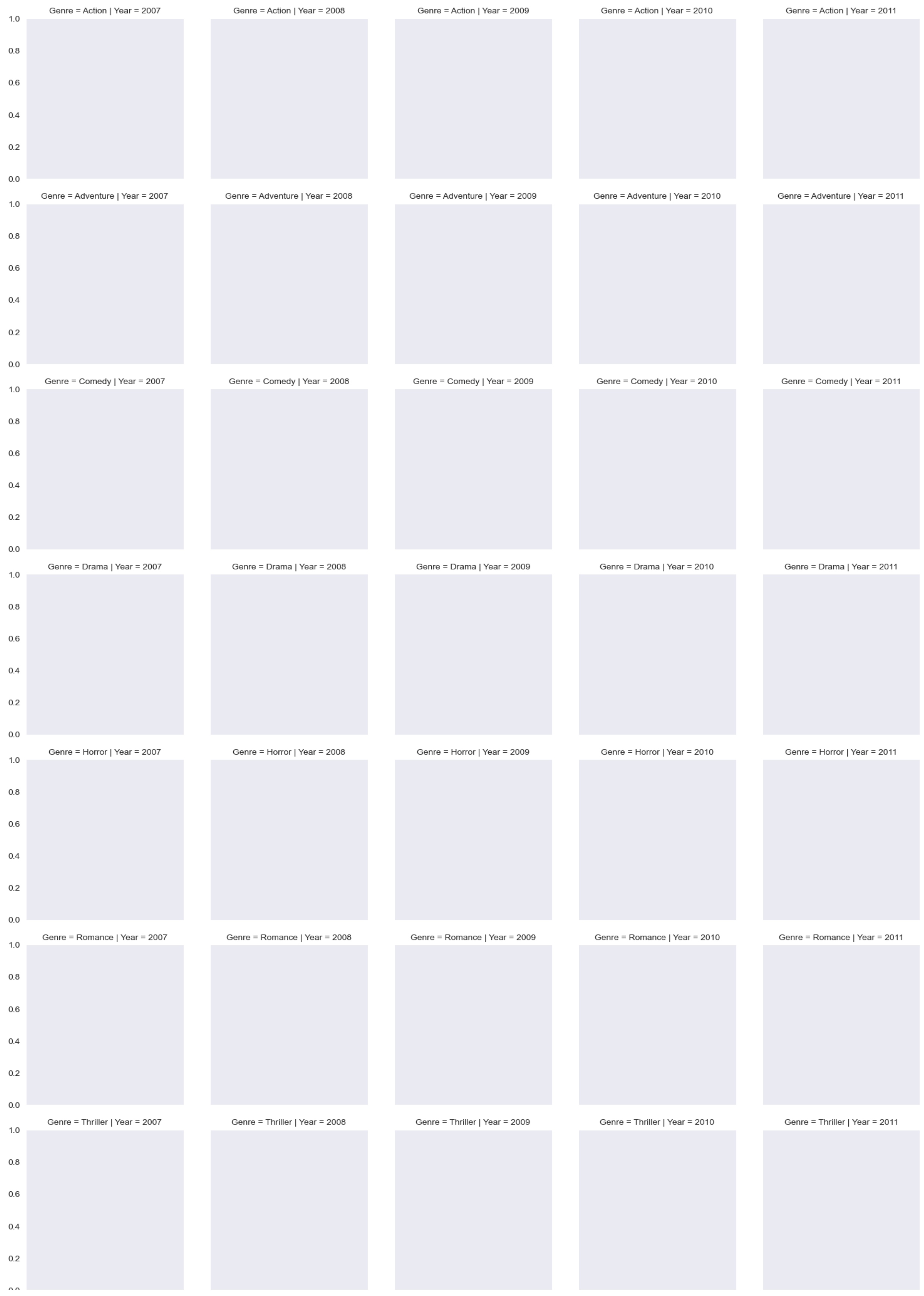
```
w1 = sns.boxplot(data=movies[movies.Genre == 'Drama'], x='Year', y =
'CriticRating',palette='Set3')

plt.show()
```



```
z = sns.violinplot(data=movies[movies.Genre == 'Drama'], x='Year', y =
'CriticRating',palette='Set3')

plt.show()
```



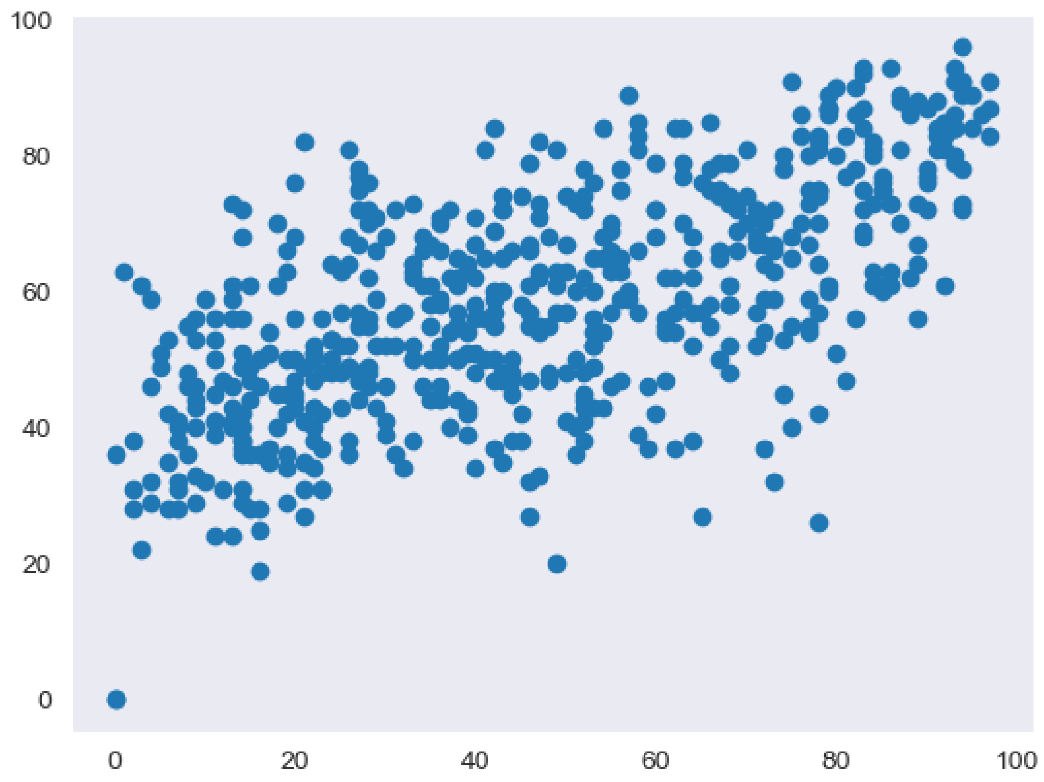
```
#Creating a Facet grid

g=sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue =
'Genre',palette='Set3') #kind of subplots

plt.show()
```


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
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
1.0
0.8
0.6
0.4
0.2
0.0

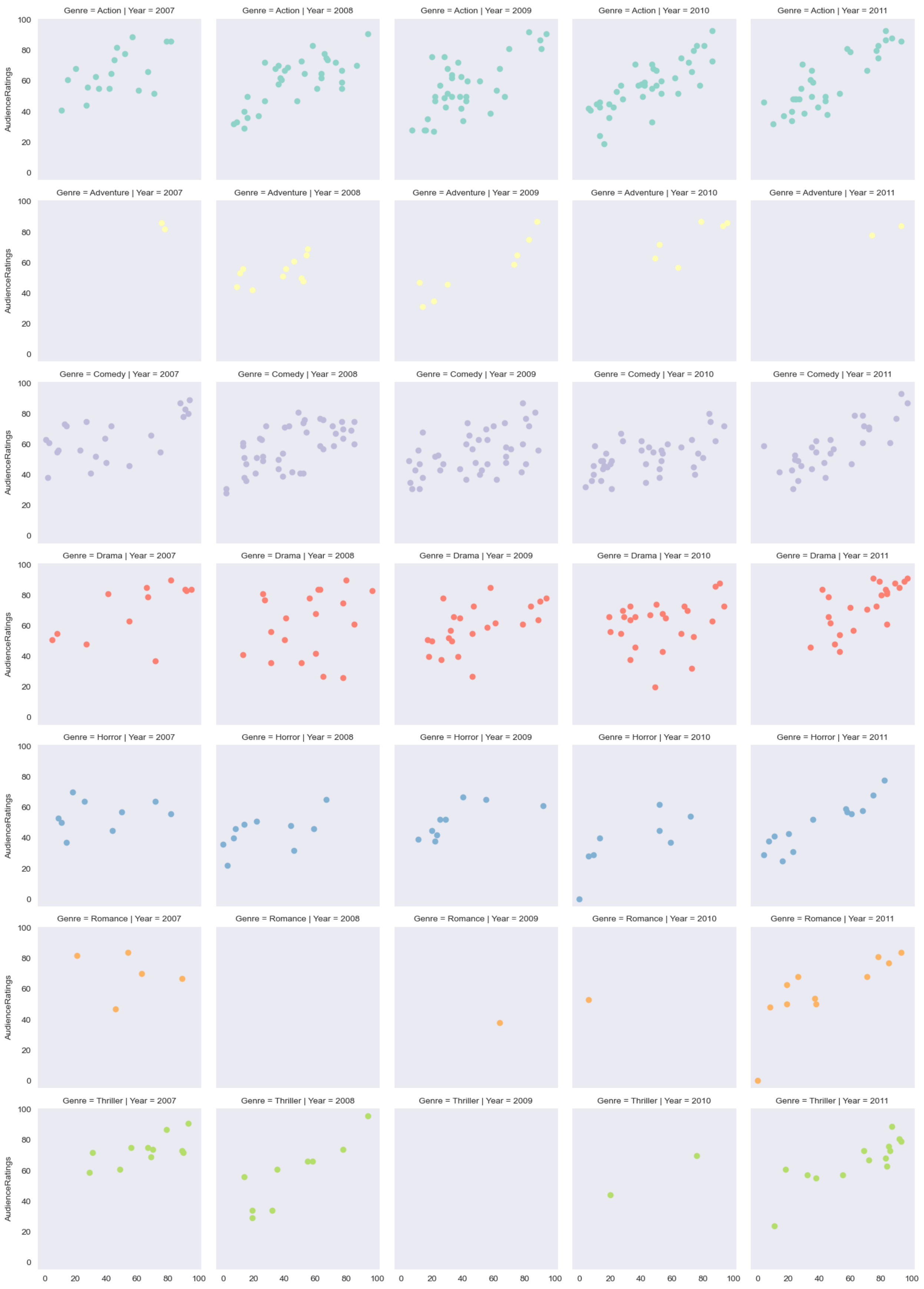
```
plt.scatter(x=movies.CriticRating,y=movies.AudienceRatings)
```

```
<matplotlib.collections.PathCollection at 0x1dd280152b0>
```

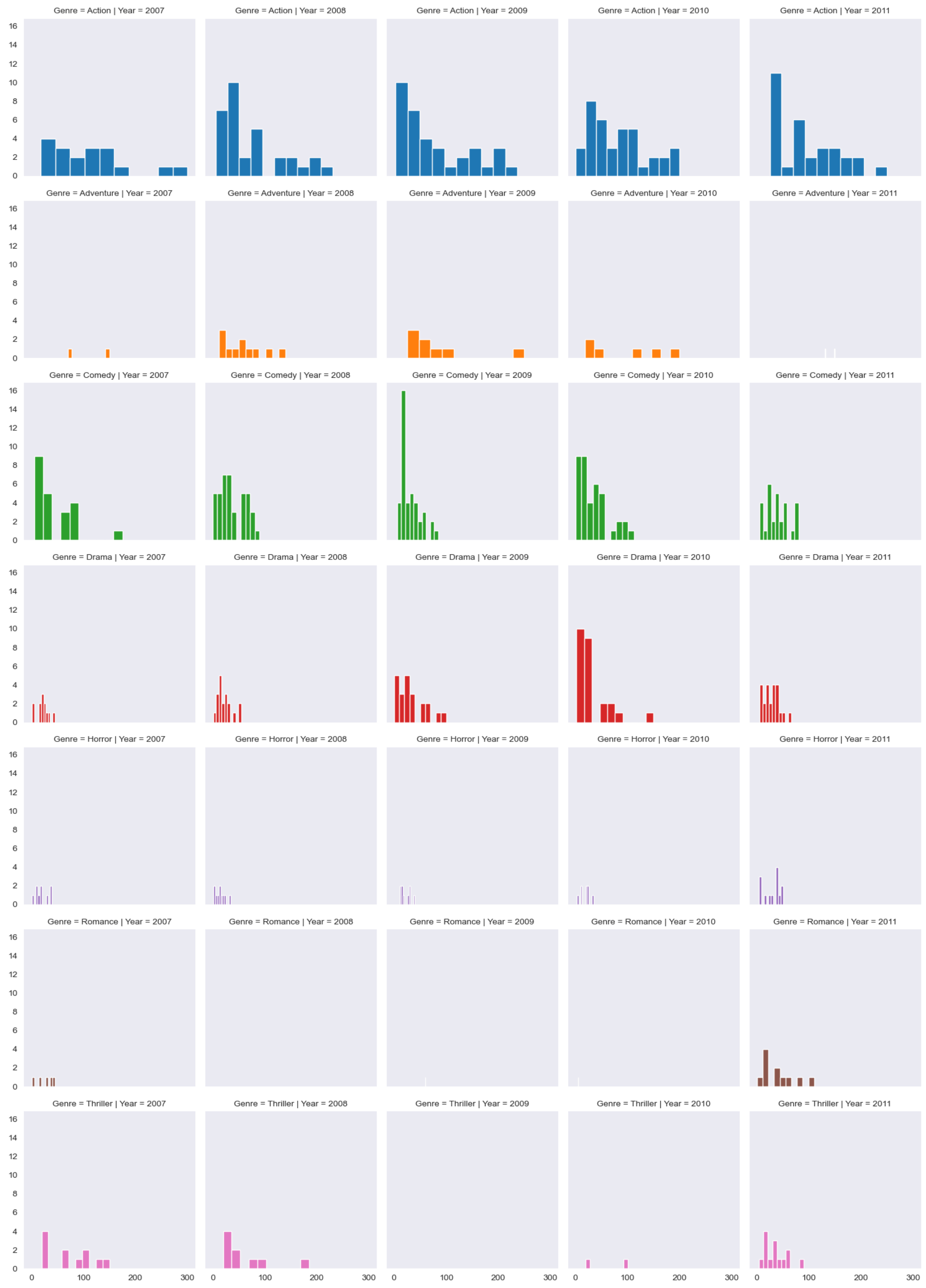
```
plt.show()
```



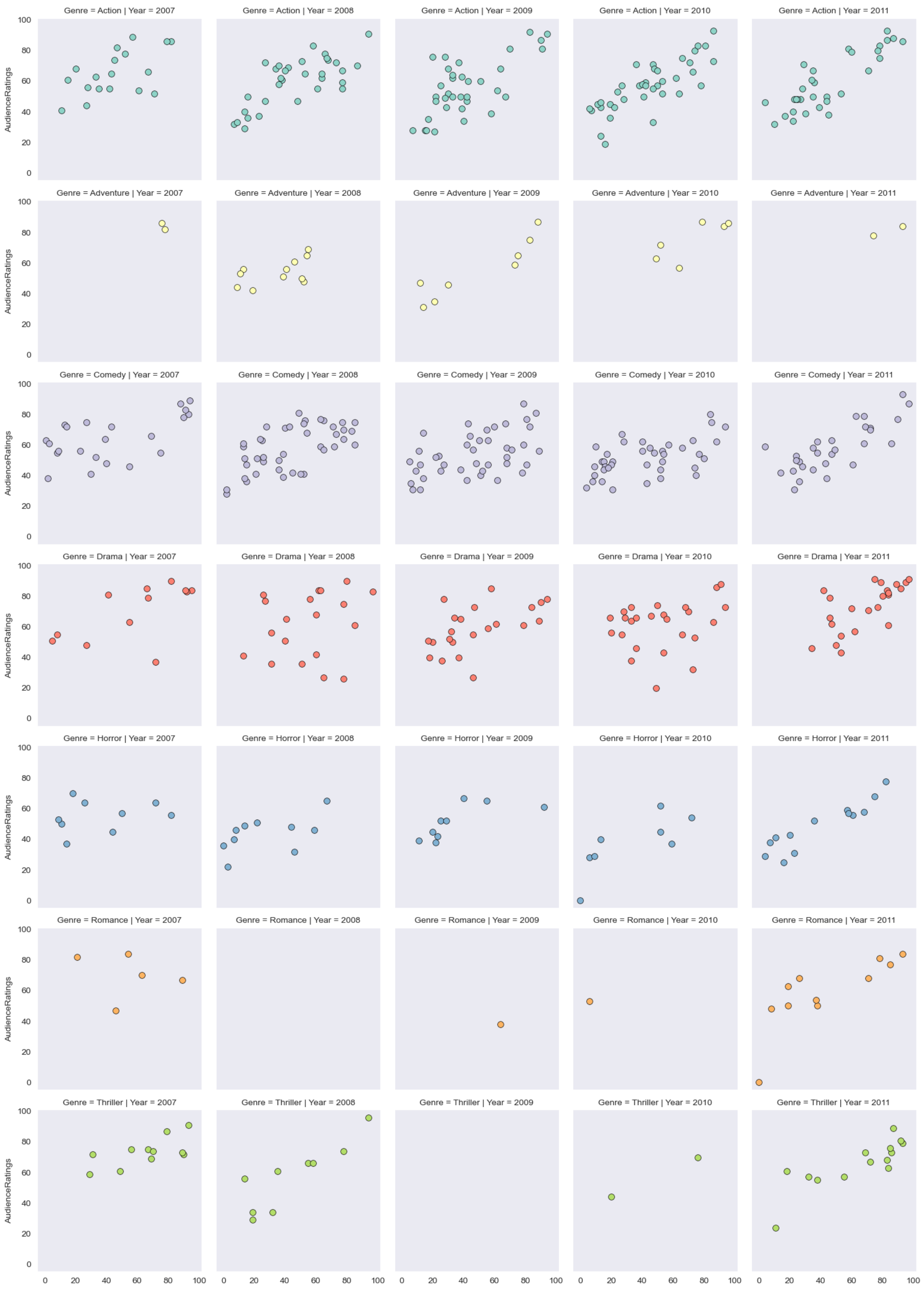
```
g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue =
'Genre',palette='Set3')
g = g.map(plt.scatter, 'CriticRating', 'AudienceRatings' )
#scatterplots are mapped in facetgrid

plt.show()
```


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
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
AudienceRatings
0
20
40
60
80
100

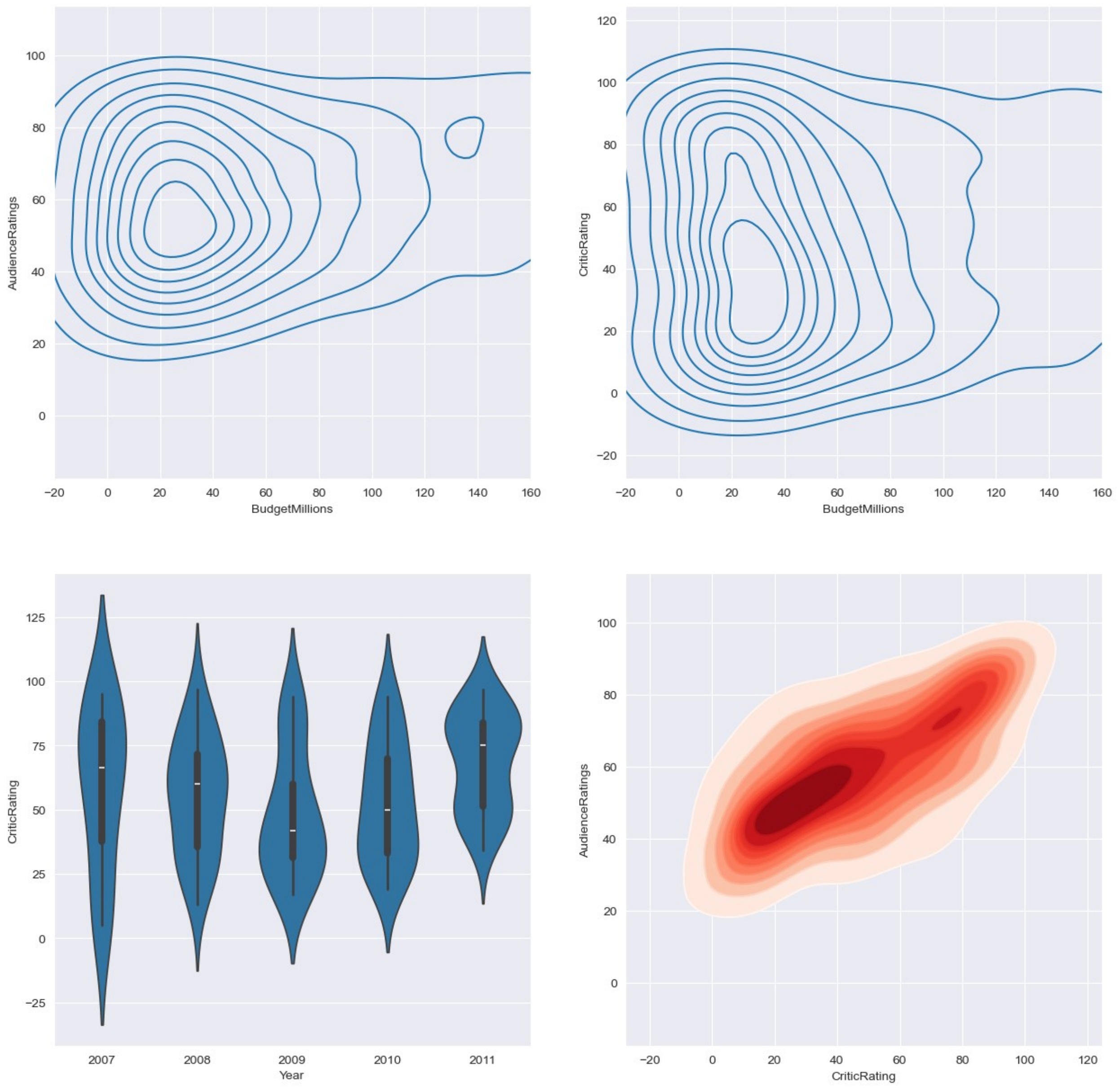
```
# you can populated any type of chat.

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

plt.show()
```


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
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
0
2
4
6
8
10
12
14
16
0
100
200
300

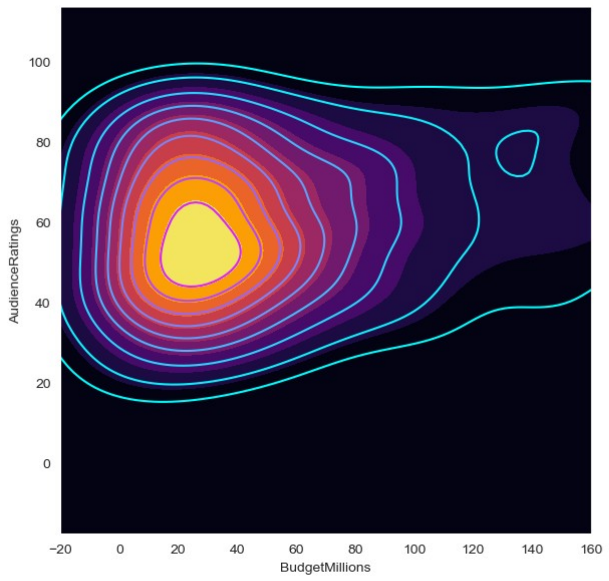
```
#
g =sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue =
'Genre',palette='Set3')
kws = dict(s=50, linewidth=0.5,edgecolor='black')
g = g.map(plt.scatter, 'CriticRating', 'AudienceRatings',**kws )
#scatterplots are mapped in facetgrid

plt.show()
```

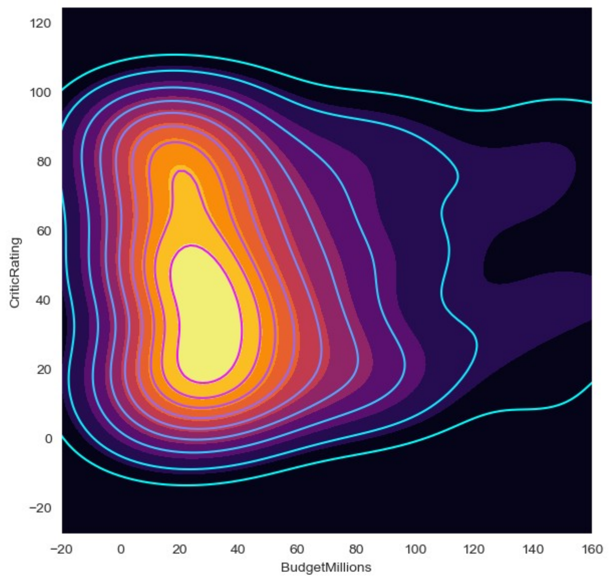

Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011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

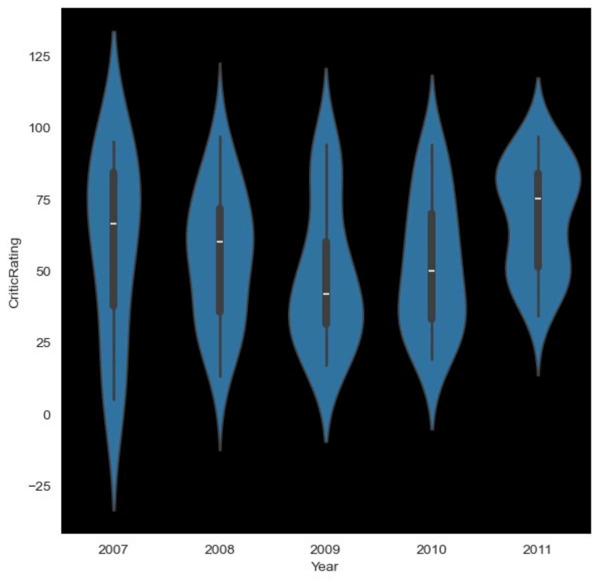
```python
# python is not vectorize programming language
# Building dashboards (dashboard - combination of chats)

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

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

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

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

k4 = sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRatings,shade
= True,shade_lowest=False,cmap='Reds',ax=axes[1,1])

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

plt.show()
```

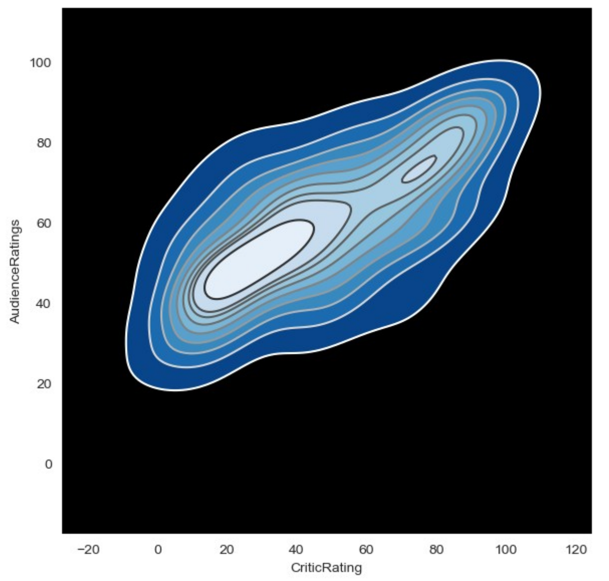


```
# How can you style your dashboard  using different color map

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

sns.set_style('dark',{'axes.facecolor':'black'})
f, axes = plt.subplots (2,2, figsize = (15,15))

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

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

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

#plot[1,1]
k4 = sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRatings, \
                 shade = True,shade_lowest=False,cmap='Blues_r', \
                 ax=axes[1,1])

k4b = sns.kdeplot(x=movies.CriticRating,y=movies.AudienceRatings, \
                  cmap='gist_gray_r',ax = axes[1,1])


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

plt.show()
```


AudienceRatings
BudgetMillions


CriticRating
BudgetMillions


CriticRating
Year
2007
2008
2009
2010
2011


AudienceRatings
CriticRating