

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

```
In [2]: movies=pd.read_csv(r"E:\fsds_course\Movie-Rating.xlsx - Movie-Rating.csv")
```

```
In [3]: movies
```

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 x 6 columns

```
In [4]: type(movies)
```

Out[4]: pandas.core.frame.DataFrame

```
In [5]: movies.columns
```

Out[5]: Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %', 'Budget (million \$)', 'Year of release'], dtype='object')

```
In [6]: 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
```

```
In [7]: len(movies)
```

Out[7]: 559

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

Out[8]: (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 [9]:

movies.tail()

Out[9]:

	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 [10]:

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

In [11]:

movies.columns

Out[11]:

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

In [12]:

movies

Out[12]:

	Film	Genre	CriticRating	AudienceRatings	Budget	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 [13]:

movies.describe()

Out[13]:

	CriticRating	AudienceRatings	Budget	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 [14]: 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   CriticRating    559 non-null   int64
 3   AudienceRatings 559 non-null   int64
 4   Budget          559 non-null   int64
 5   Year           559 non-null   int64
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

```
In [16]: movies['Film']
```

```
Out[16]: 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 [20]: movies.Film
```

```
Out[20]: 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 [25]: movies.Film = movies.Film.astype('category')
```

```
In [27]: movies.Film
```

```
Out[27]: 0      2009
1      2008
2      2009
3      2010
4      2009
...
554    2011
555    2009
556    2007
557    2009
558    2011
Name: Film, Length: 559, dtype: category
Categories (5, int64): [2007, 2008, 2009, 2010, 2011]
```

In [30]: `movies.head()`

Out[30]:

	Film	Genre	CriticRating	AudienceRatings	Budget	Year
0	2009	Comedy	87	81	8	2009
1	2008	Adventure	9	44	105	2008
2	2009	Action	30	52	20	2009
3	2010	Adventure	93	84	18	2010
4	2009	Comedy	55	70	20	2009

In [31]: `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   object
2   CriticRating    559 non-null   int64
3   AudienceRatings 559 non-null   int64
4   Budget          559 non-null   int64
5   Year            559 non-null   int64
dtypes: category(1), int64(4), object(1)
memory usage: 22.7+ KB
```

In [36]: `movies.Genre = movies.Genre.astype('category')`  
`movies.Year = movies.Year.astype('category')`

In [37]: `movies.Genre`

Out[37]:

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 [38]: `movies.Year`

Out[38]:

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 [39]: `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   Budget          559 non-null   int64  
 5   Year            559 non-null   category
dtypes: category(3), int64(3)
memory usage: 15.6 KB
```

In [42]: `movies.Genre.cat.categories`

Out[42]: Index(['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance',  
                  'Thriller'],  
              dtype='object')

In [43]: `movies.describe()`

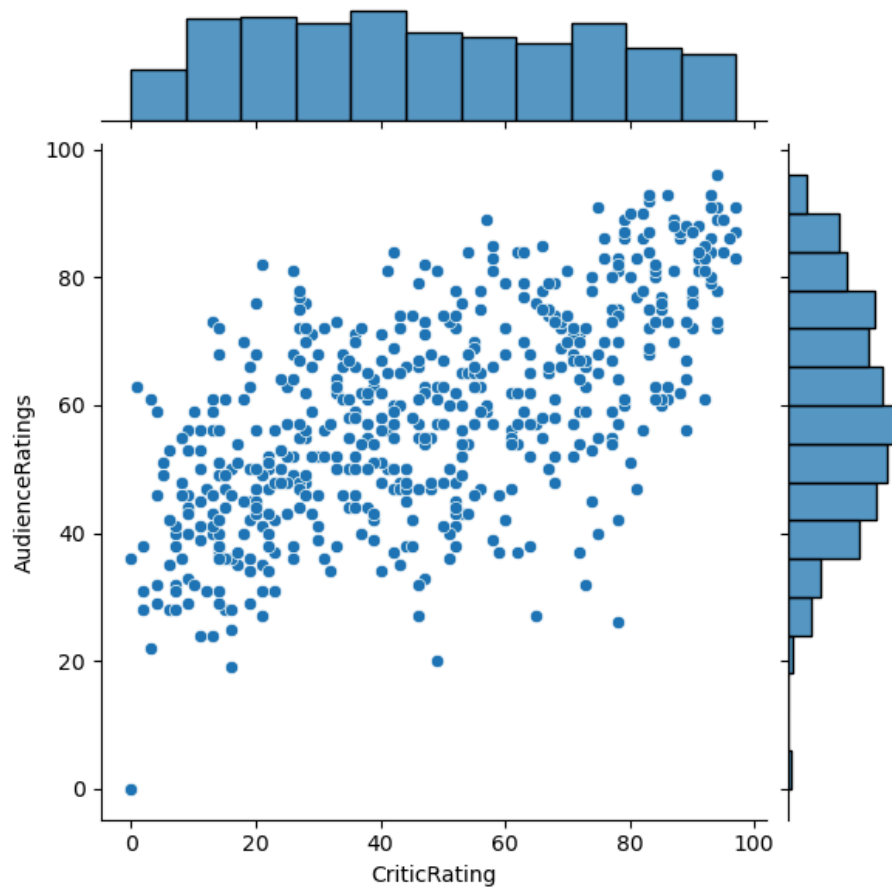
Out[43]:

	CriticRating	AudienceRatings	Budget
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

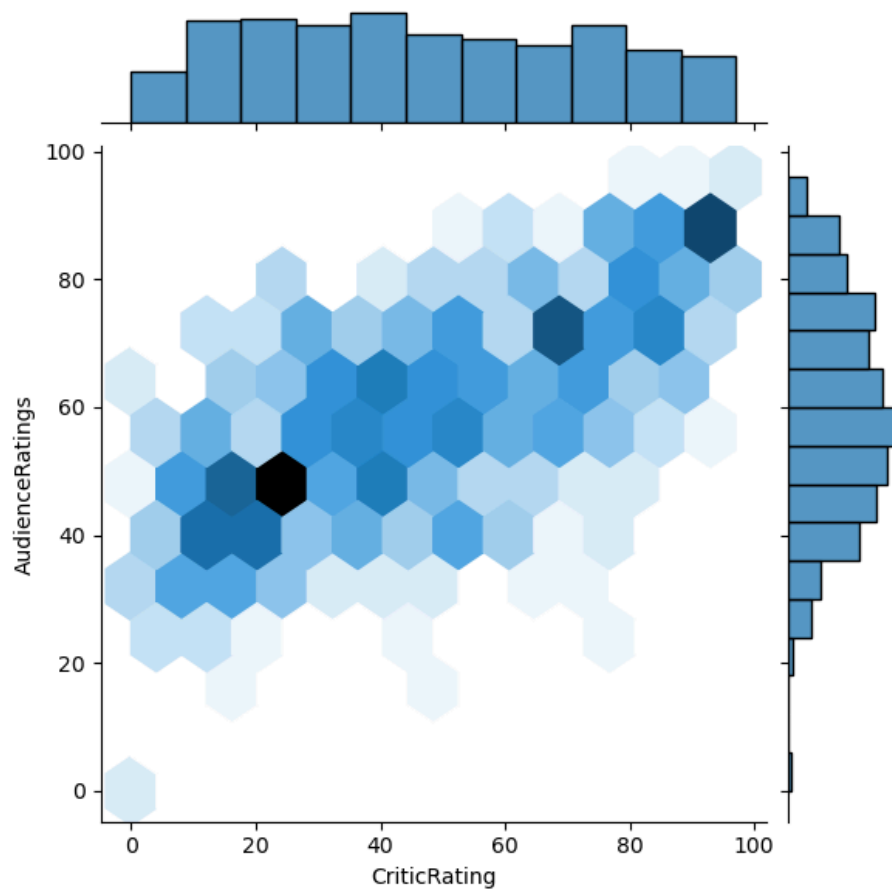
In [46]: `#how to working with joint plot`

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

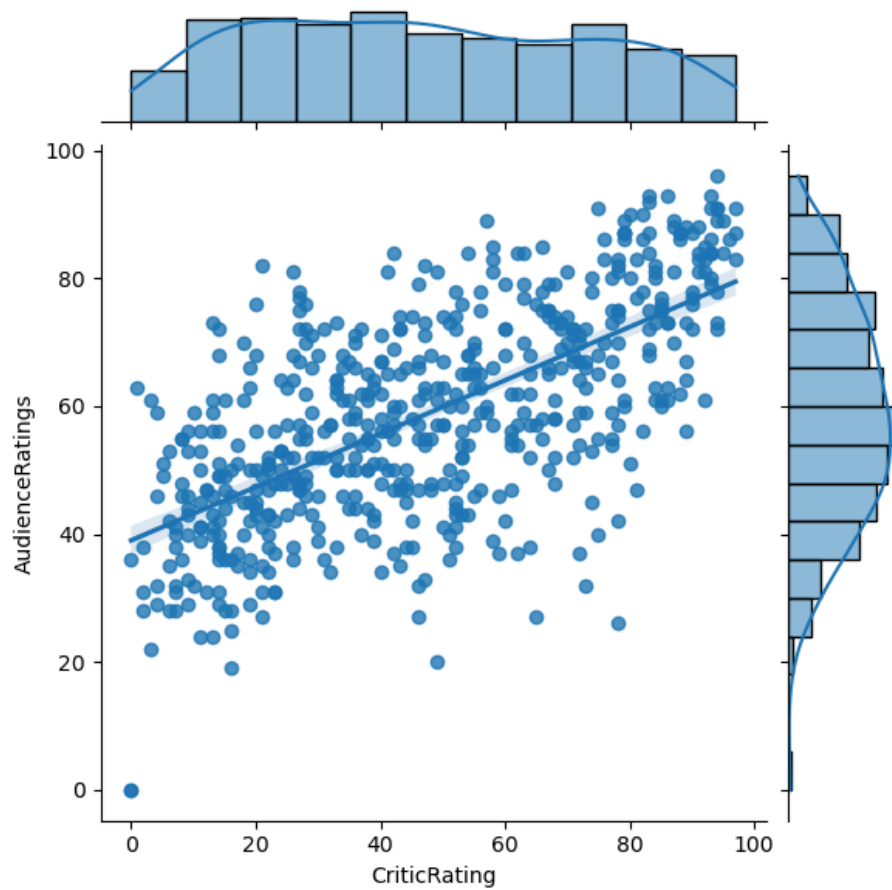
```
In [51]: j = sns.jointplot( data = movies, x = 'CriticRating', y = 'AudienceRatings')
```



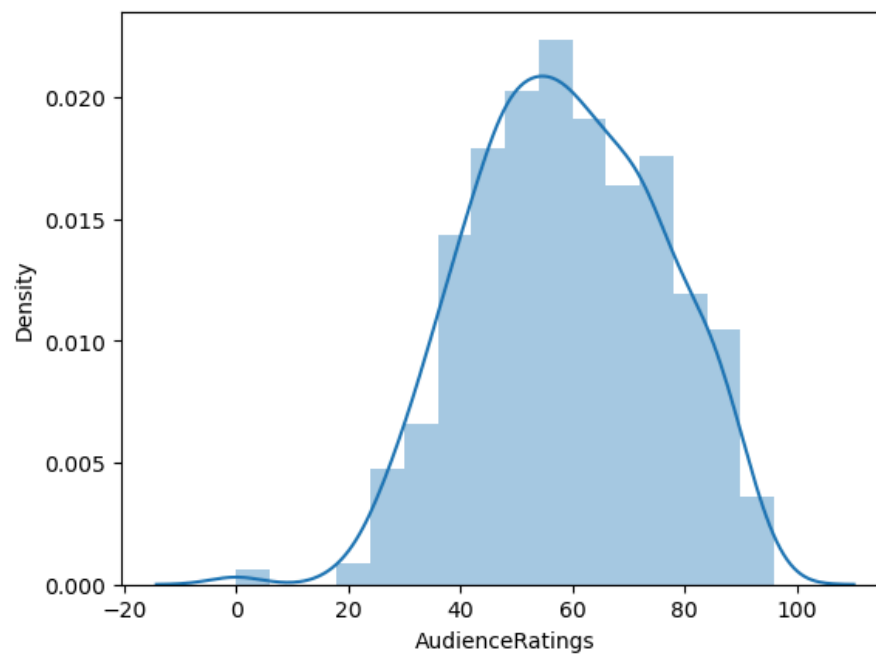
```
In [53]: j = sns.jointplot( data = movies, x = 'CriticRating', y = 'AudienceRatings', kind='hex')
```



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



```
In [55]: m1= sns.distplot(movies.AudienceRatings)
```



```
In [57]: sns.set_style('darkgrid')
```

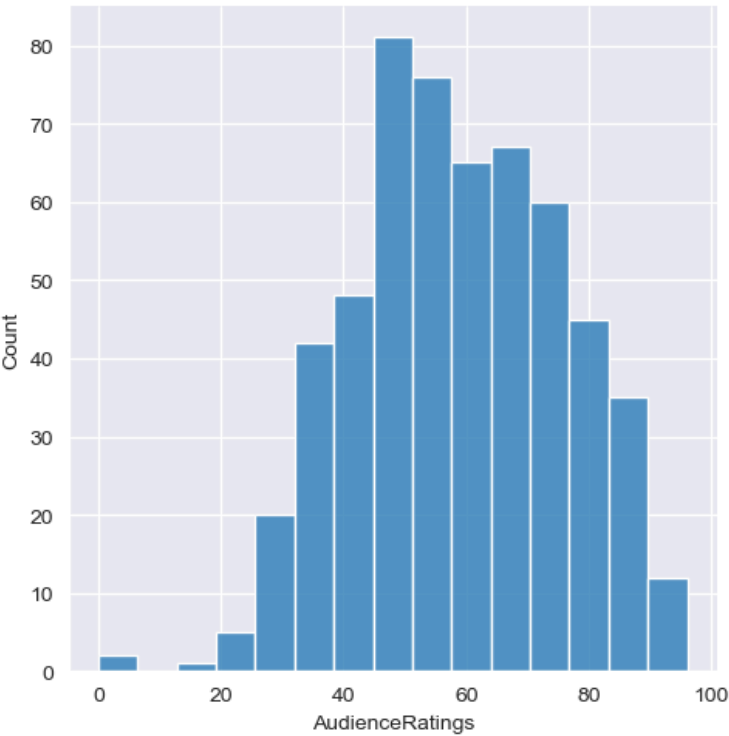
```
In [56]: movies
```

Out[56]:

	Film	Genre	CriticRating	AudienceRatings	Budget	Year
0	2009	Comedy	87	81	8	2009
1	2008	Adventure	9	44	105	2008
2	2009	Action	30	52	20	2009
3	2010	Adventure	93	84	18	2010
4	2009	Comedy	55	70	20	2009
...	...	...	...	...	...	...
554	2011	Comedy	26	36	50	2011
555	2009	Comedy	68	52	18	2009
556	2007	Thriller	89	73	65	2007
557	2009	Action	90	87	24	2009
558	2011	Comedy	14	42	80	2011

559 rows × 6 columns

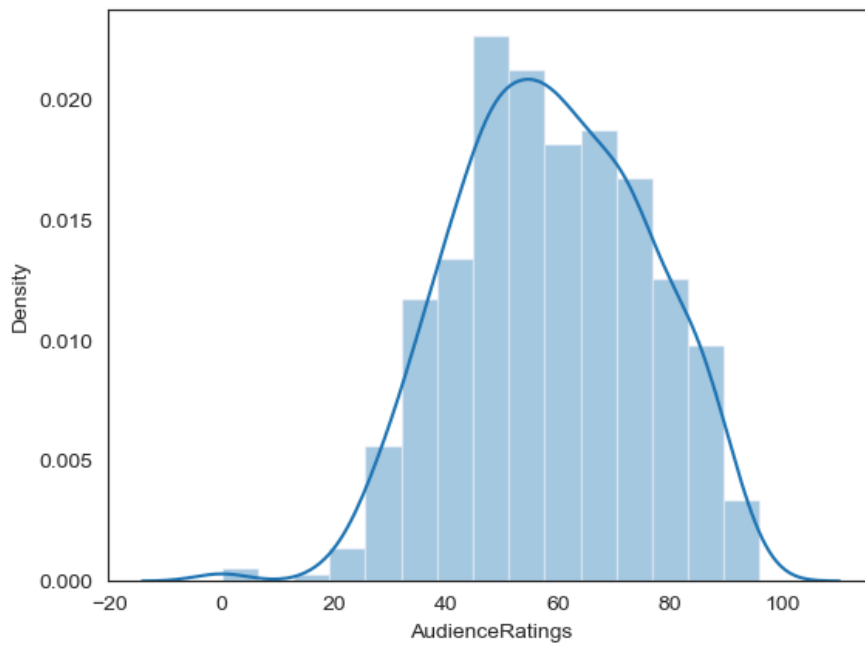
```
In [58]: m2= sns.displot(movies.AudienceRatings,bins=15)
```



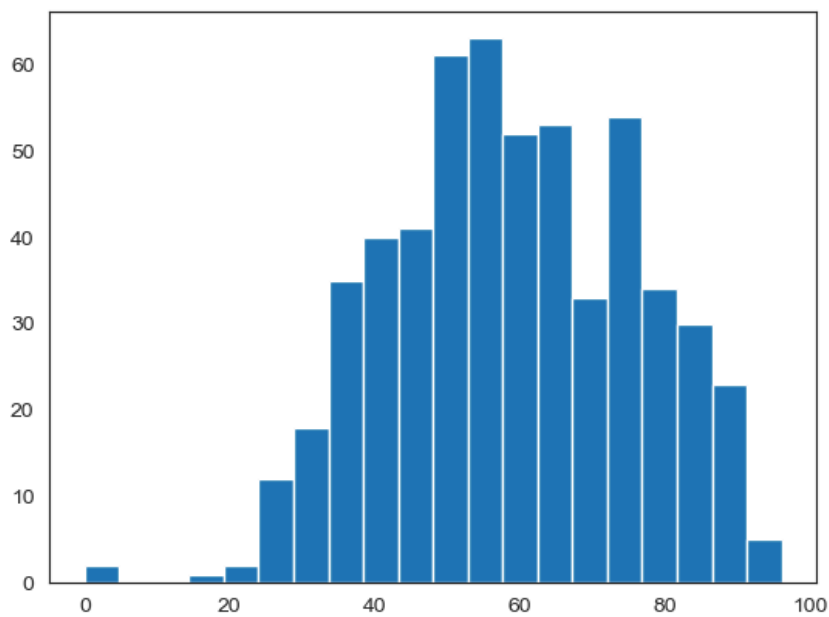
```
In [59]: sns.set_style('white')
```



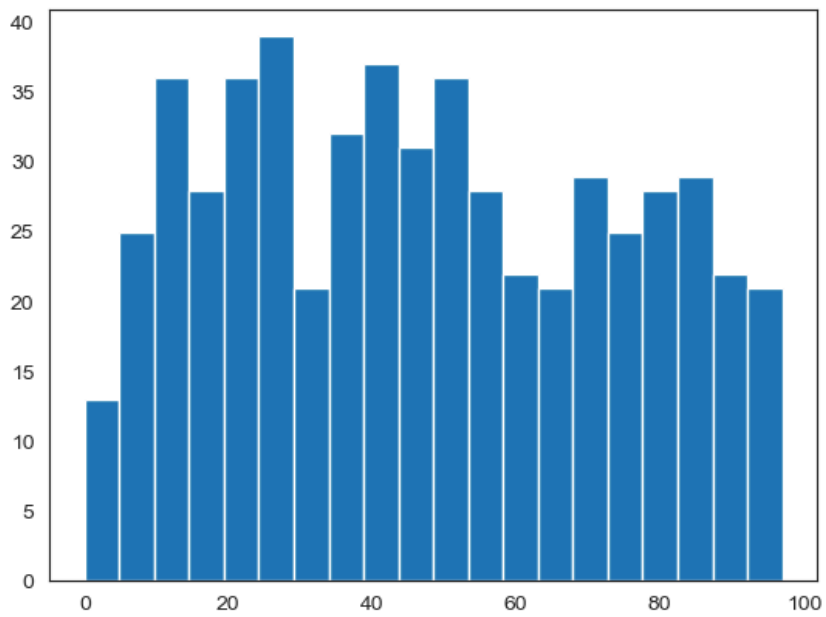
```
In [60]: m2= sns.distplot(movies.AudienceRatings,bins = 15)
```



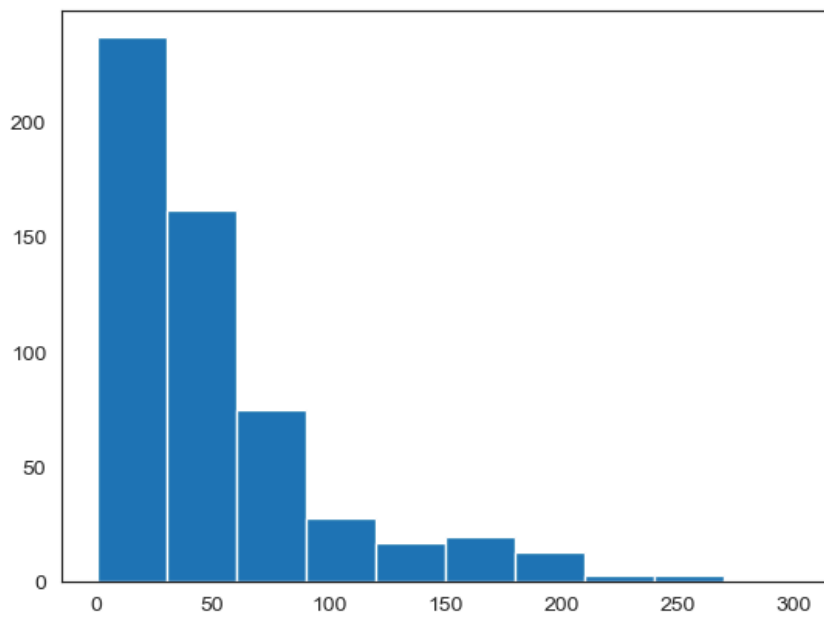
```
In [61]: sns.set_style('white')  
n1=plt.hist(movies.AudienceRatings,bins=20) # normal distribution
```



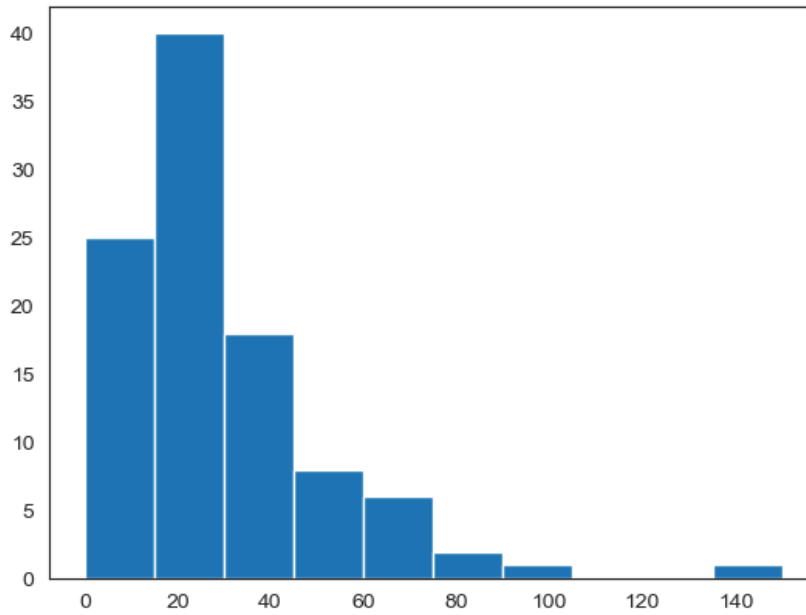
```
In [62]: n1 = plt.hist(movies.CriticRating, bins= 20) # uniform distribution
```



```
In [63]: plt.hist(movies.Budget)  
plt.show()
```



```
In [66]: plt.hist(movies[movies.Genre=='Drama'].Budget)
plt.show()
```



```
In [65]: movies.head()
```

```
Out[65]:
```

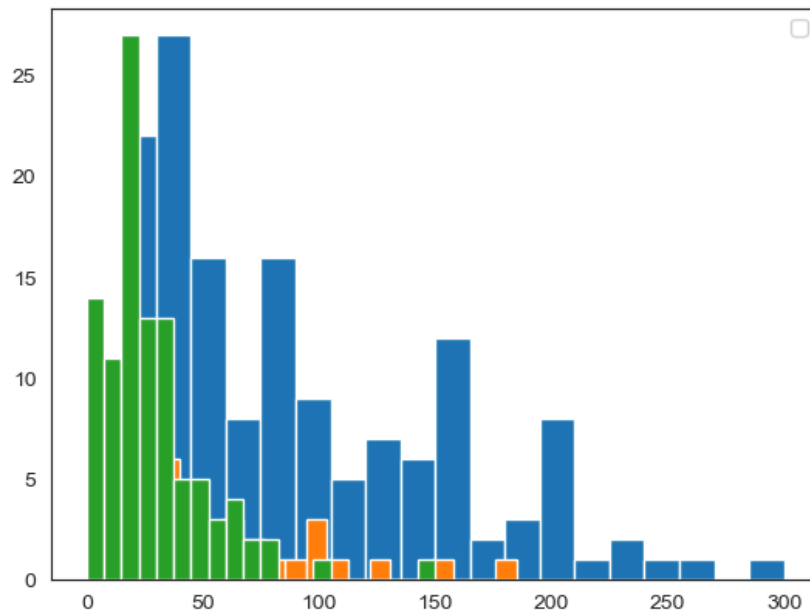
	Film	Genre	CriticRating	AudienceRatings	Budget	Year
0	2009	Comedy	87	81	8	2009
1	2008	Adventure	9	44	105	2008
2	2009	Action	30	52	20	2009
3	2010	Adventure	93	84	18	2010
4	2009	Comedy	55	70	20	2009

```
In [67]: movies.Genre.unique()
```

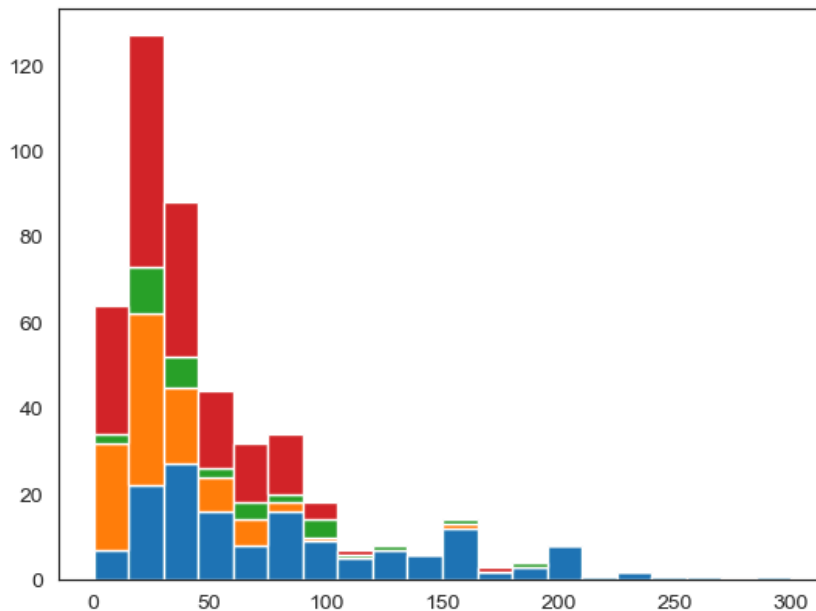
```
Out[67]: ['Comedy', 'Adventure', 'Action', 'Horror', 'Drama', 'Romance', 'Thriller']
Categories (7, object): ['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance', 'Thriller']
```

```
In [68]: plt.hist(movies[movies.Genre=='Action'].Budget,bins=20)
plt.hist(movies[movies.Genre=='Thriller'].Budget , bins=20)
plt.hist(movies[movies.Genre=='Drama'].Budget, 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.



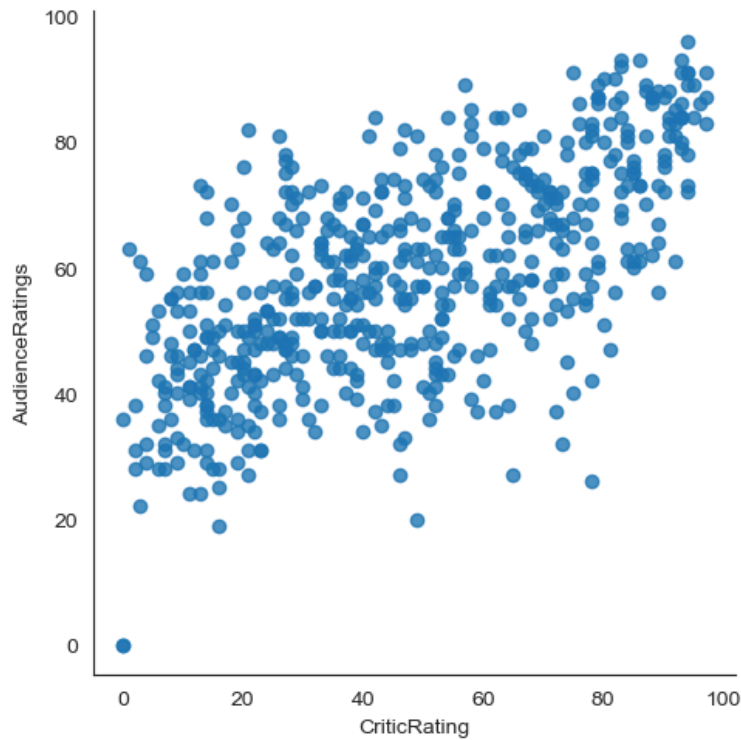
```
In [71]: plt.hist([movies[movies.Genre=='Action'].Budget,\
    movies[movies.Genre=='Drama'].Budget,\
    movies[movies.Genre=='Thriller'].Budget,\
    movies[movies.Genre=='Comedy'].Budget],bins=20, stacked=True)
plt.show()
```



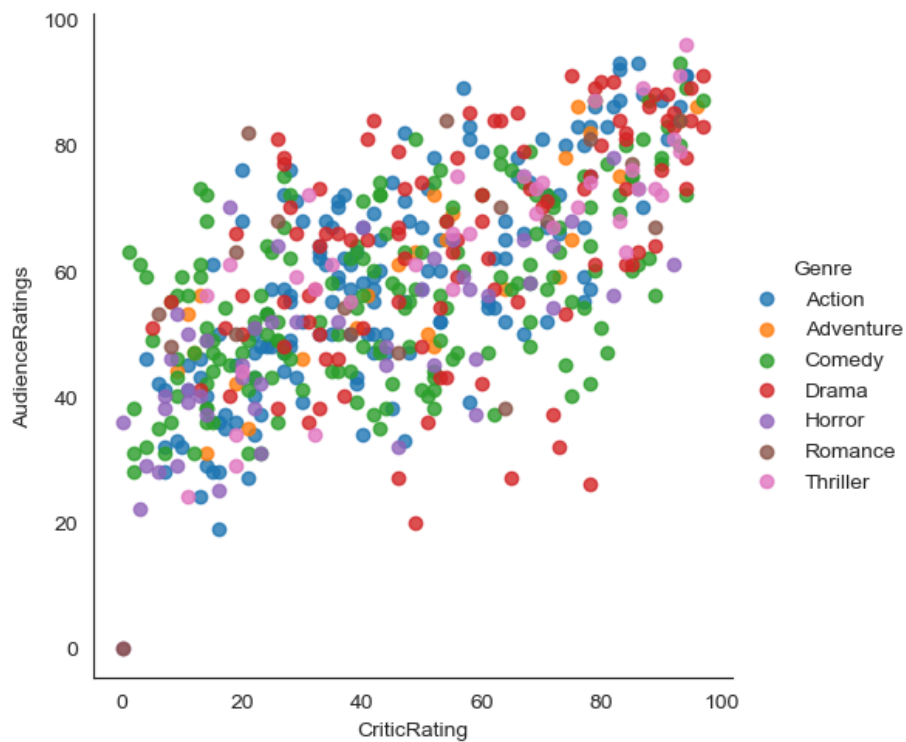
```
In [73]: for gen in movies.Genre.cat.categories:
    print(gen)
```

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

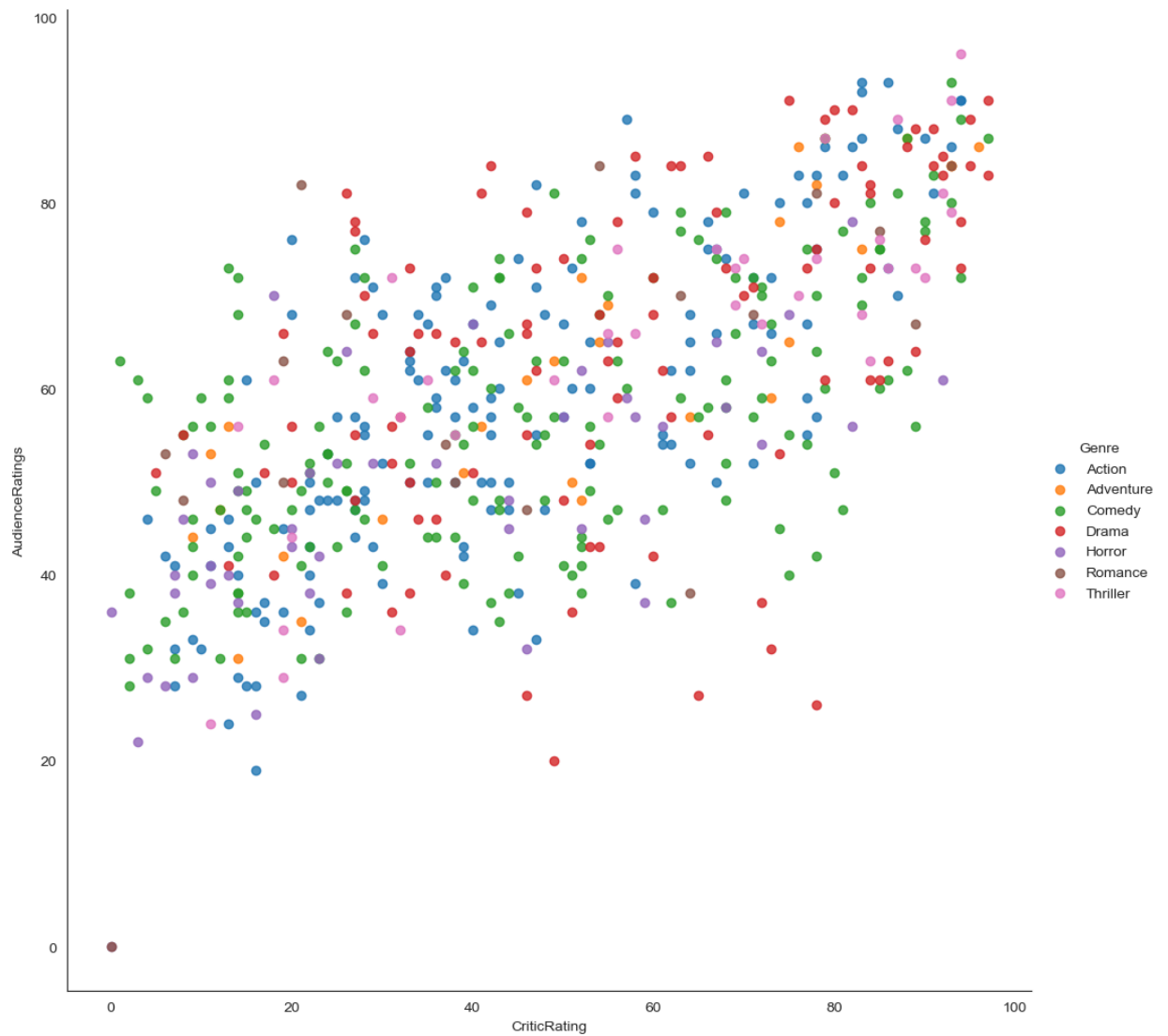
```
In [77]: vis1 = sns.lmplot(data = movies, x = 'CriticRating', y = 'AudienceRatings',  
                           fit_reg=False)
```



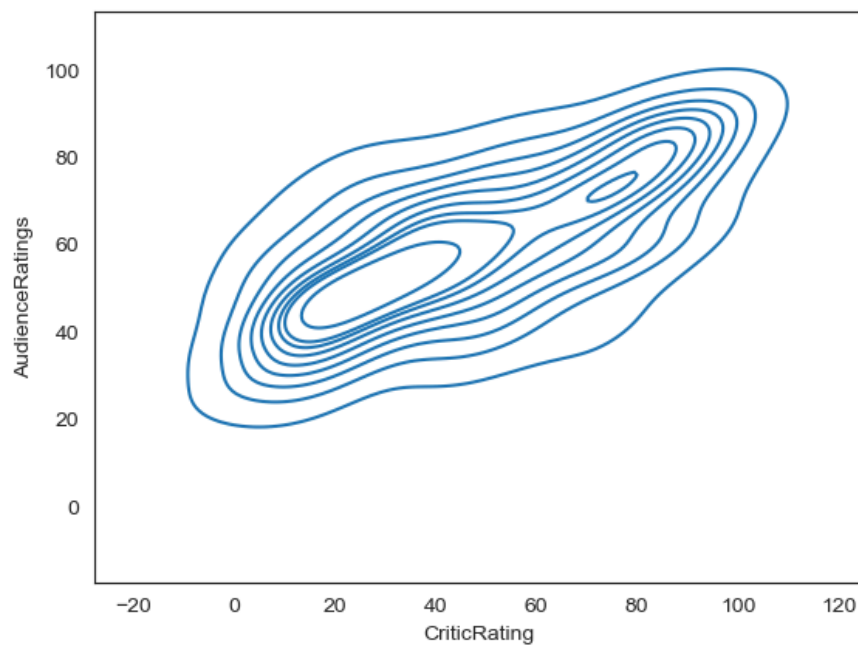
```
In [78]: vis1 = sns.lmplot(data = movies, x = 'CriticRating', y = 'AudienceRatings',  
                           fit_reg=False, hue='Genre')
```



```
In [84]: vis1 = sns.lmplot(data = movies, x = 'CriticRating', y = 'AudienceRatings',  
                           fit_reg=False, hue='Genre', height = 10, aspect=1)
```



```
In [97]: k1 = sns.kdeplot(data = movies, x= 'CriticRating' , y= 'AudienceRatings')
```



In [96]:

movies

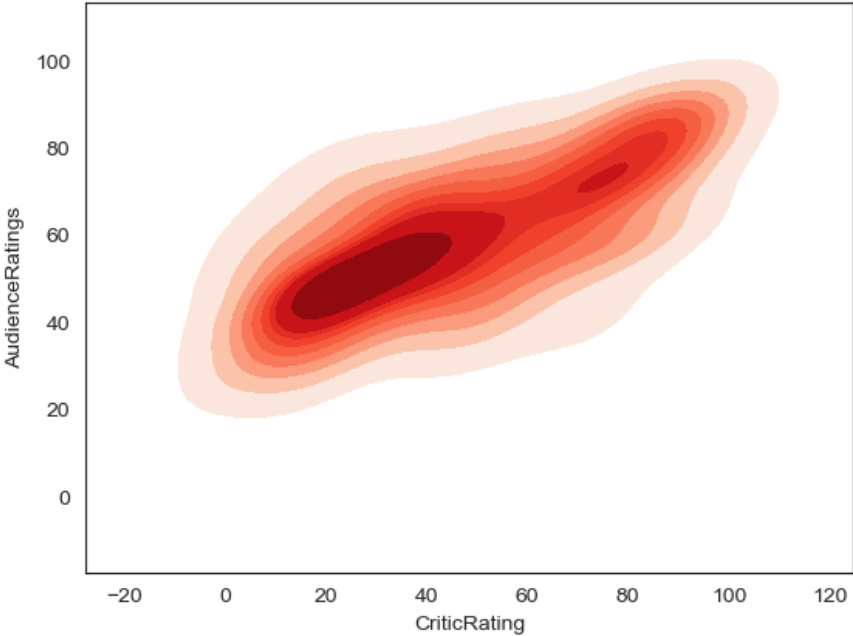
Out[96]:

	Film	Genre	CriticRating	AudienceRatings	Budget	Year
0	2009	Comedy	87	81	8	2009
1	2008	Adventure	9	44	105	2008
2	2009	Action	30	52	20	2009
3	2010	Adventure	93	84	18	2010
4	2009	Comedy	55	70	20	2009
...	...	...	...	...	...	...
554	2011	Comedy	26	36	50	2011
555	2009	Comedy	68	52	18	2009
556	2007	Thriller	89	73	65	2007
557	2009	Action	90	87	24	2009
558	2011	Comedy	14	42	80	2011

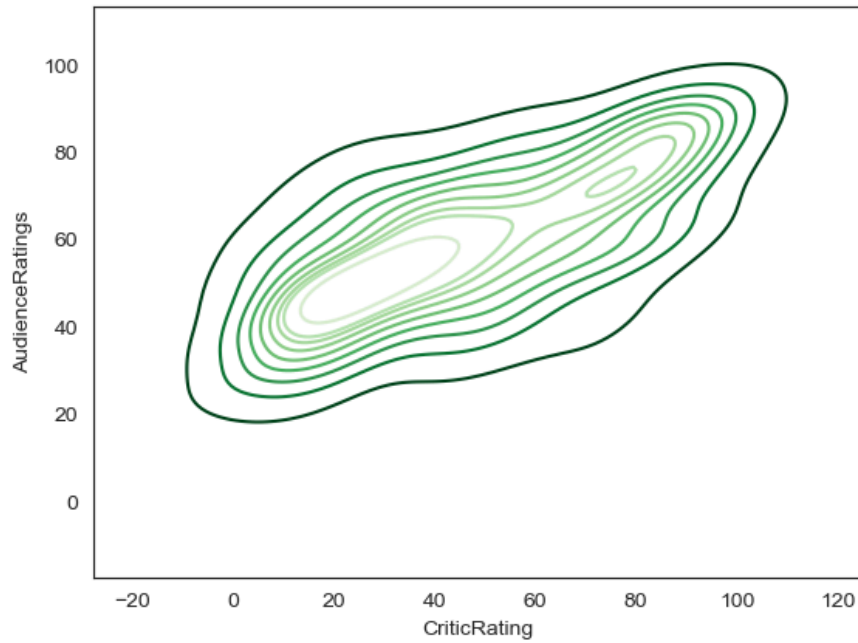
559 rows × 6 columns

In [98]:

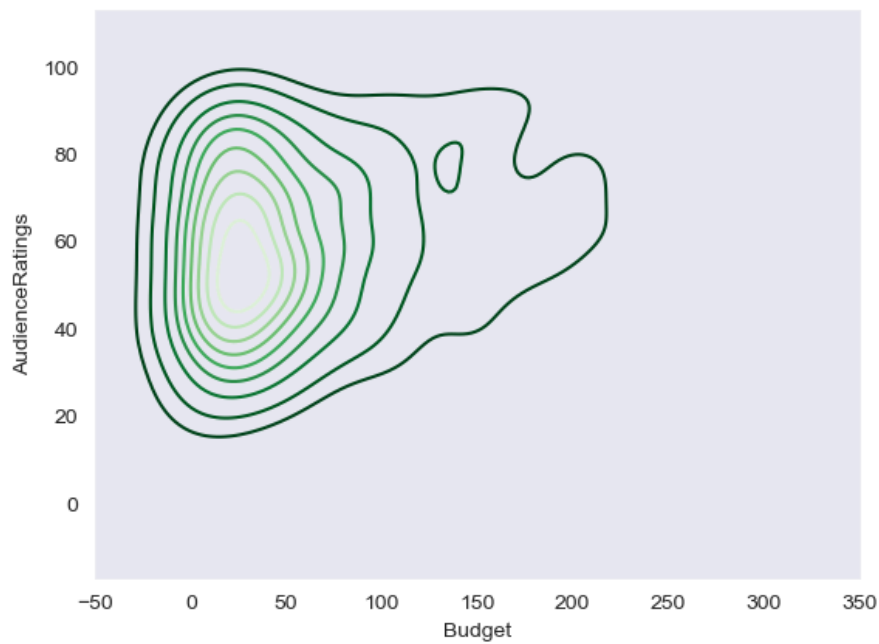
k1 = sns.kdeplot(data = movies, x= 'CriticRating' , y= 'AudienceRatings',shade=True,shade\_lowest=False,cm



```
In [99]: sns.kdeplot(data = movies, x= 'CriticRating' , y= 'AudienceRatings',shade_lowest=False,cmap='Greens_r')
```

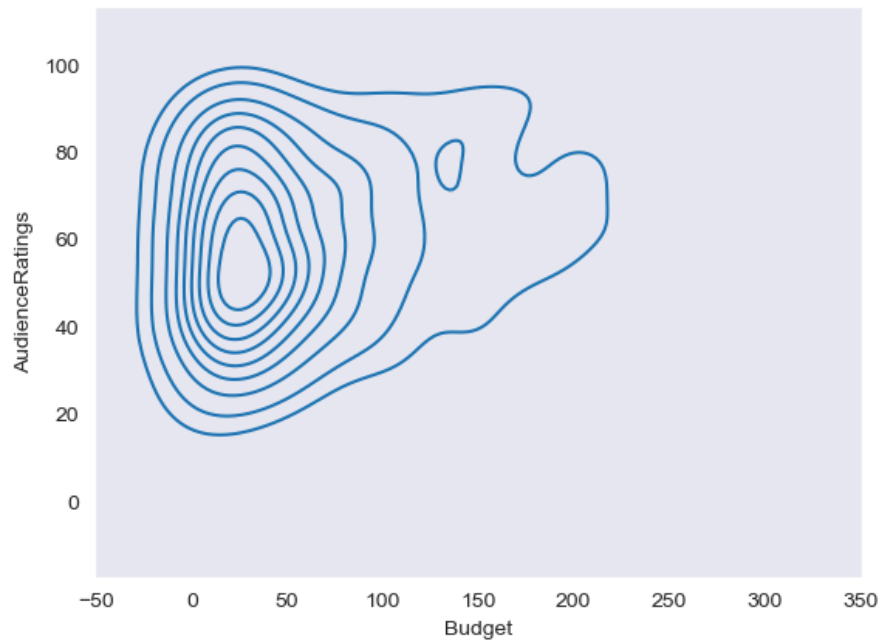


```
In [101]: sns.set_style('dark')
k1 = sns.kdeplot(data = movies, x= 'Budget' , y= 'AudienceRatings',shade_lowest=False,cmap='Greens_r')
```

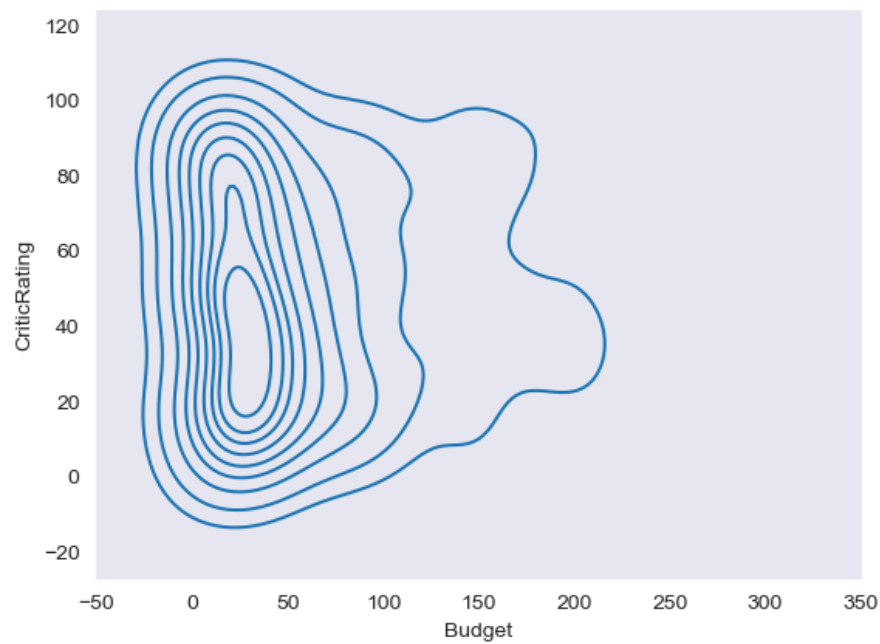




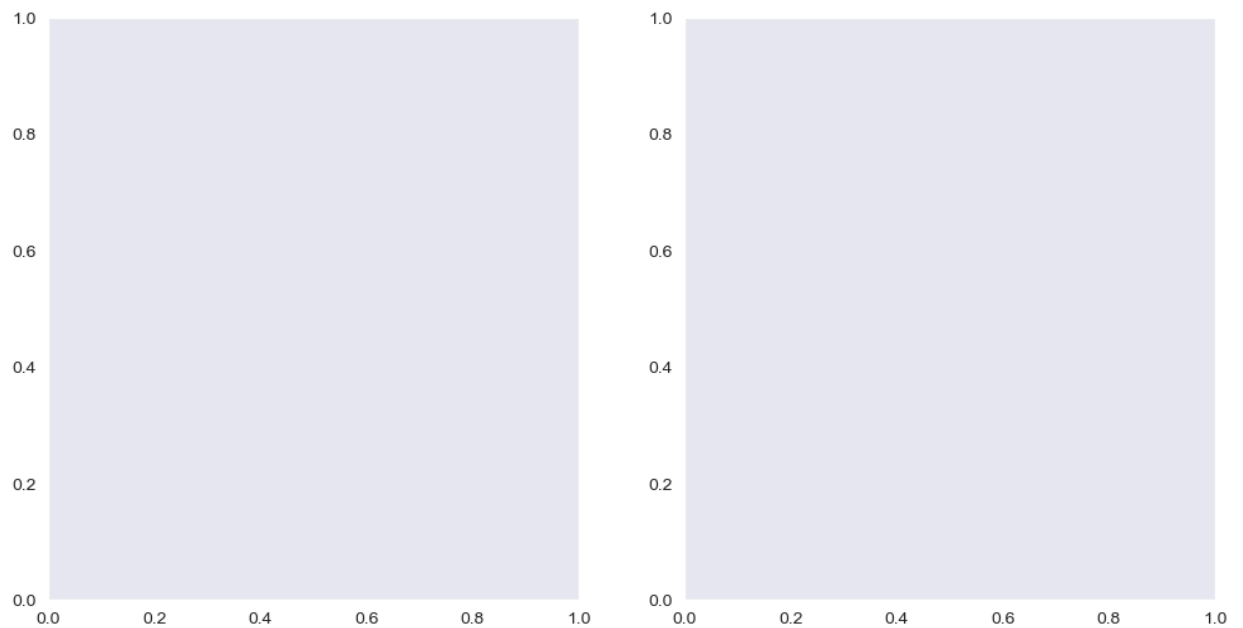
```
In [103]: sns.set_style('dark')  
k1 = sns.kdeplot(data = movies, x= 'Budget' , y= 'AudienceRatings')
```



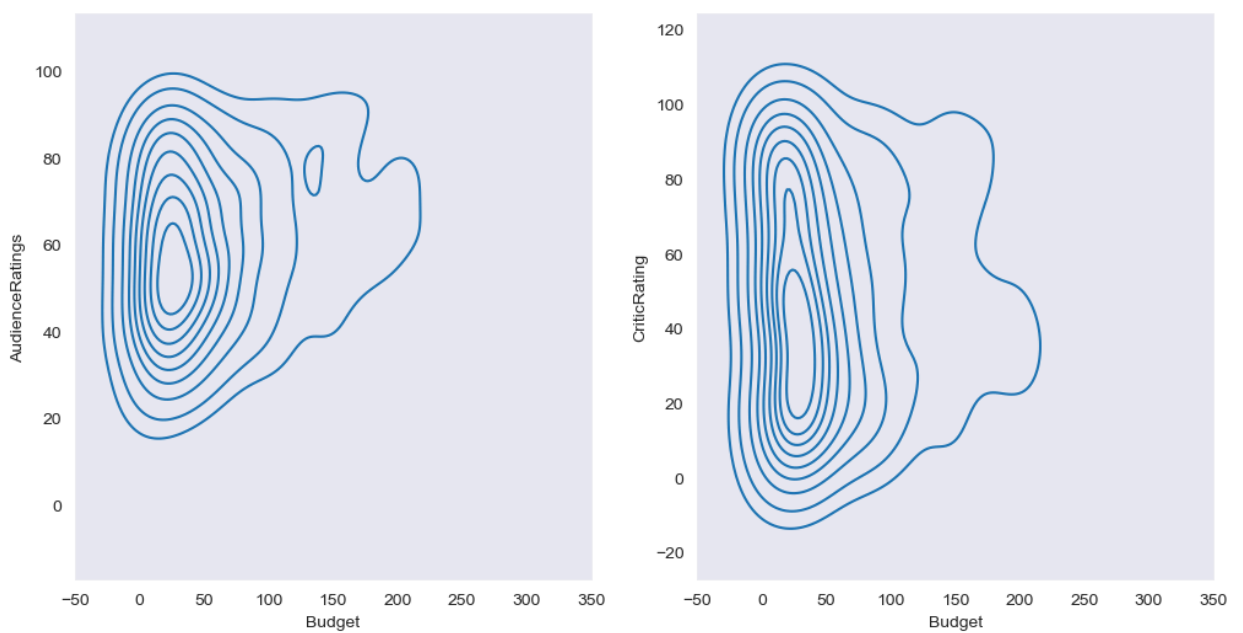
```
In [106]: k2 = sns.kdeplot(data= movies,x='Budget',y= 'CriticRating')
```



```
In [107]: #subplots
f,ax = plt.subplots(1,2,figsize = (12,6))
```



```
In [108]: f, axes = plt.subplots(1,2, figsize = (12,6))
k1= sns.kdeplot(data=movies, x='Budget',y='AudienceRatings',ax= axes[0])
k1= sns.kdeplot(data=movies, x='Budget',y='CriticRating',ax= axes[1])
```

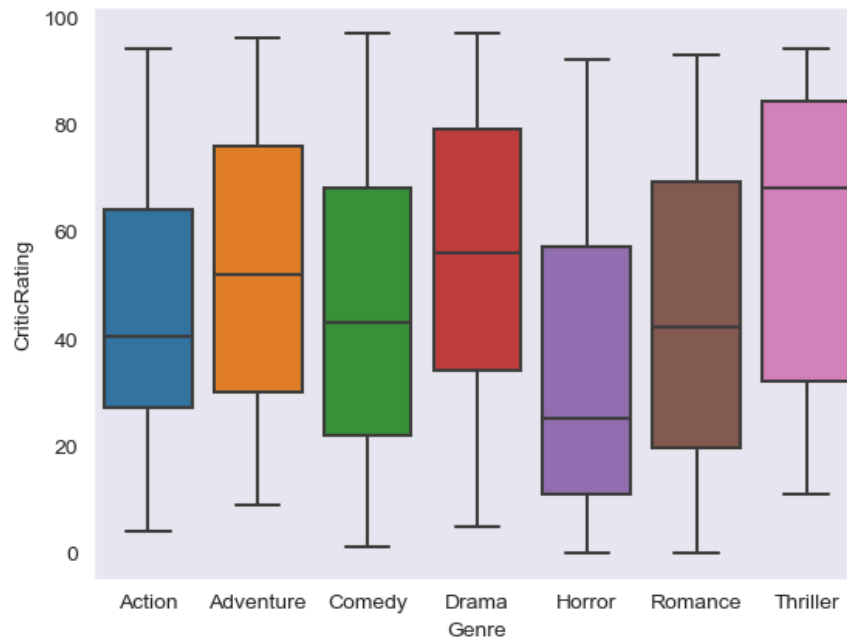


```
In [109]: axes
```

```
Out[109]: array([<Axes: xlabel='Budget', ylabel='AudienceRatings'>,
                  <Axes: xlabel='Budget', ylabel='CriticRating'>], dtype=object)
```

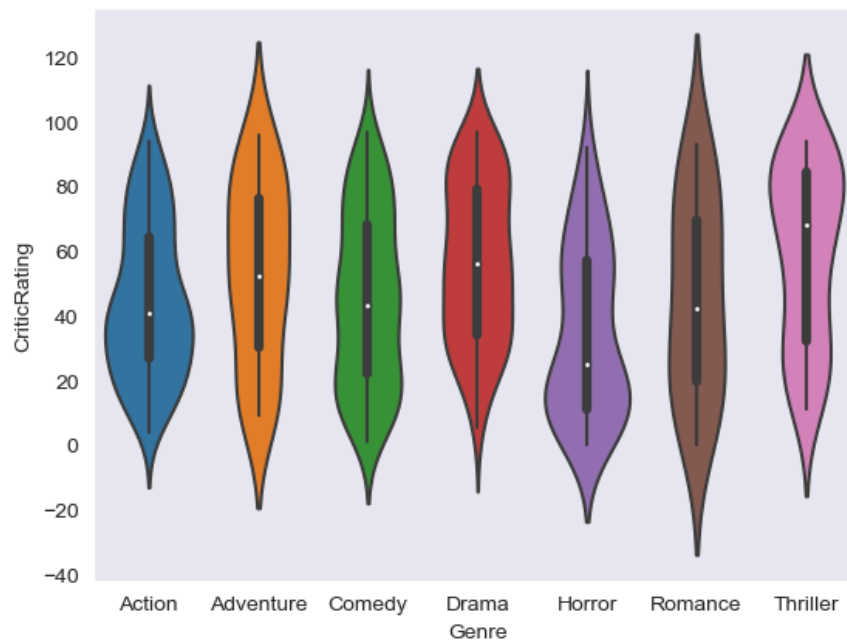
```
In [110]: #Boxplot
```

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

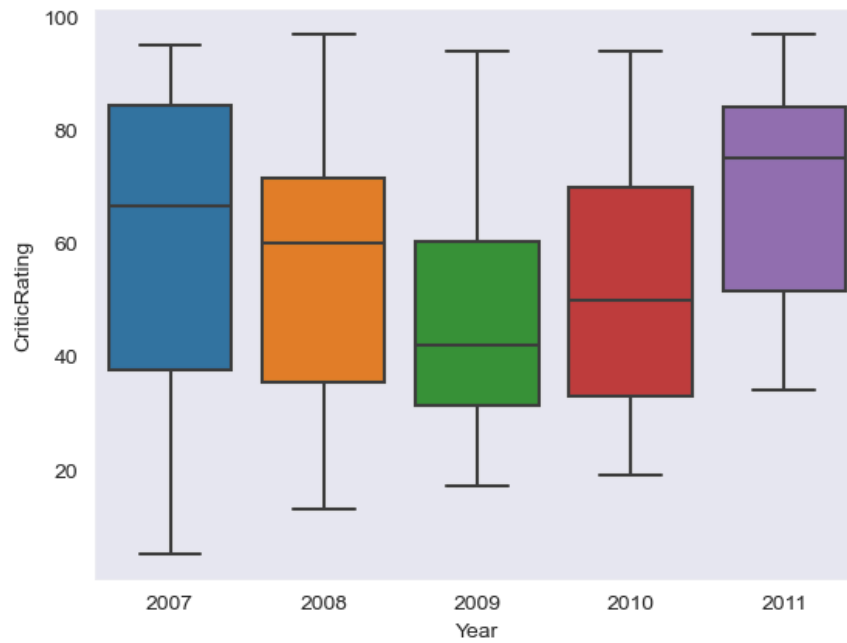


```
In [111]: #violin plot
```

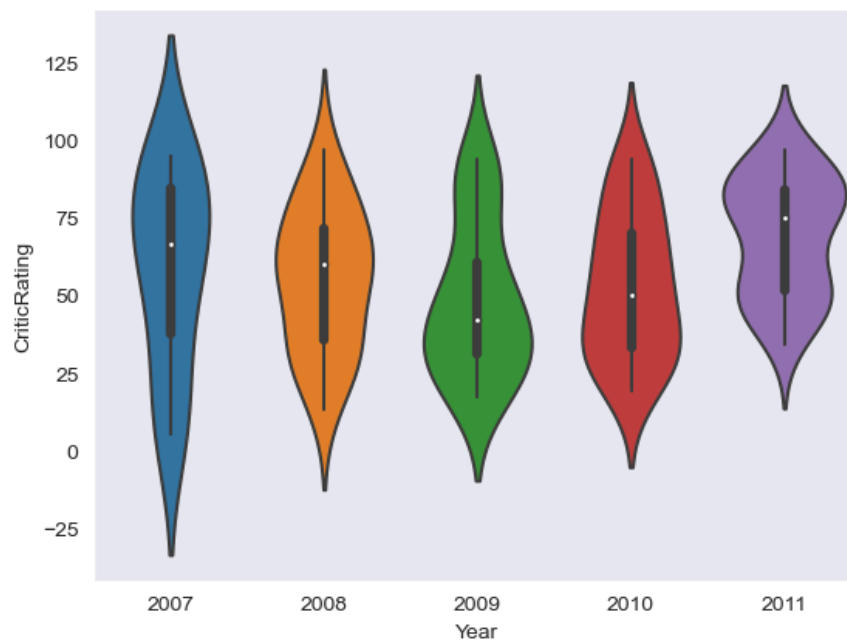
```
z=sns.violinplot(data=movies, x='Genre', y = 'CriticRating')
```



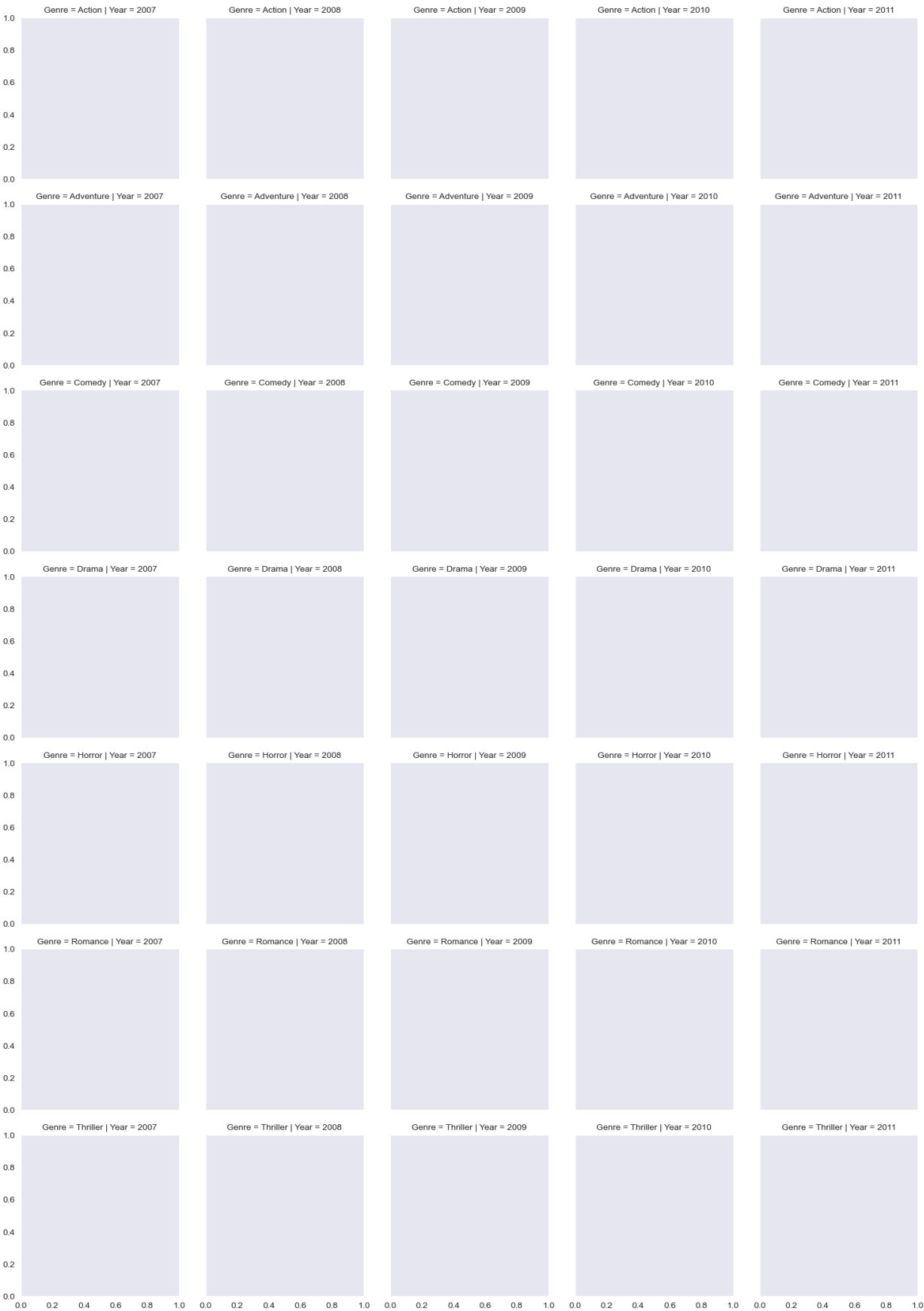
```
In [113]: w1= sns.boxplot(data=movies[movies.Genre=='Drama'], x='Year',y = 'CriticRating')
```



```
In [115]: z = sns.violinplot(data=movies[movies.Genre == 'Drama'], x='Year', y = 'CriticRating')
```

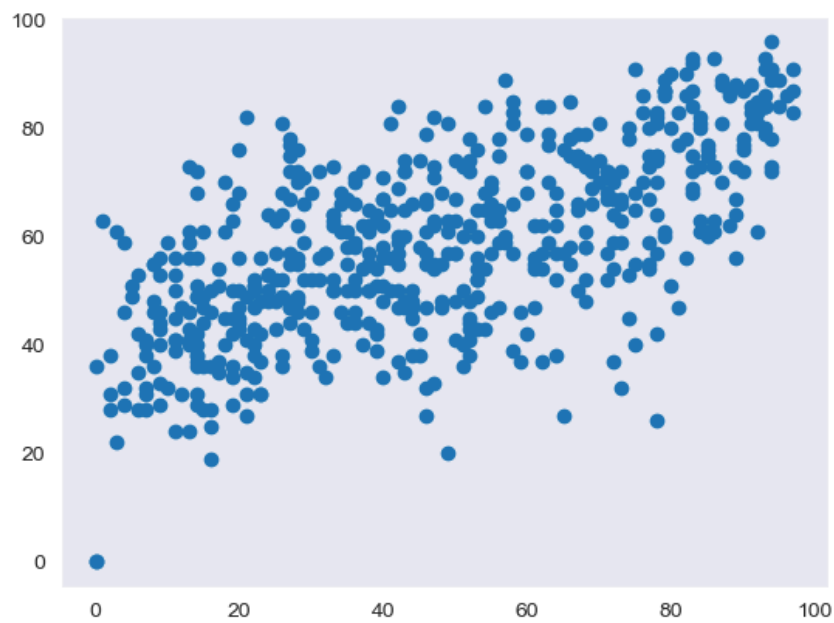


```
In [119]: g=sns.FacetGrid(movies, row='Genre', col='Year',hue='Genre')
```

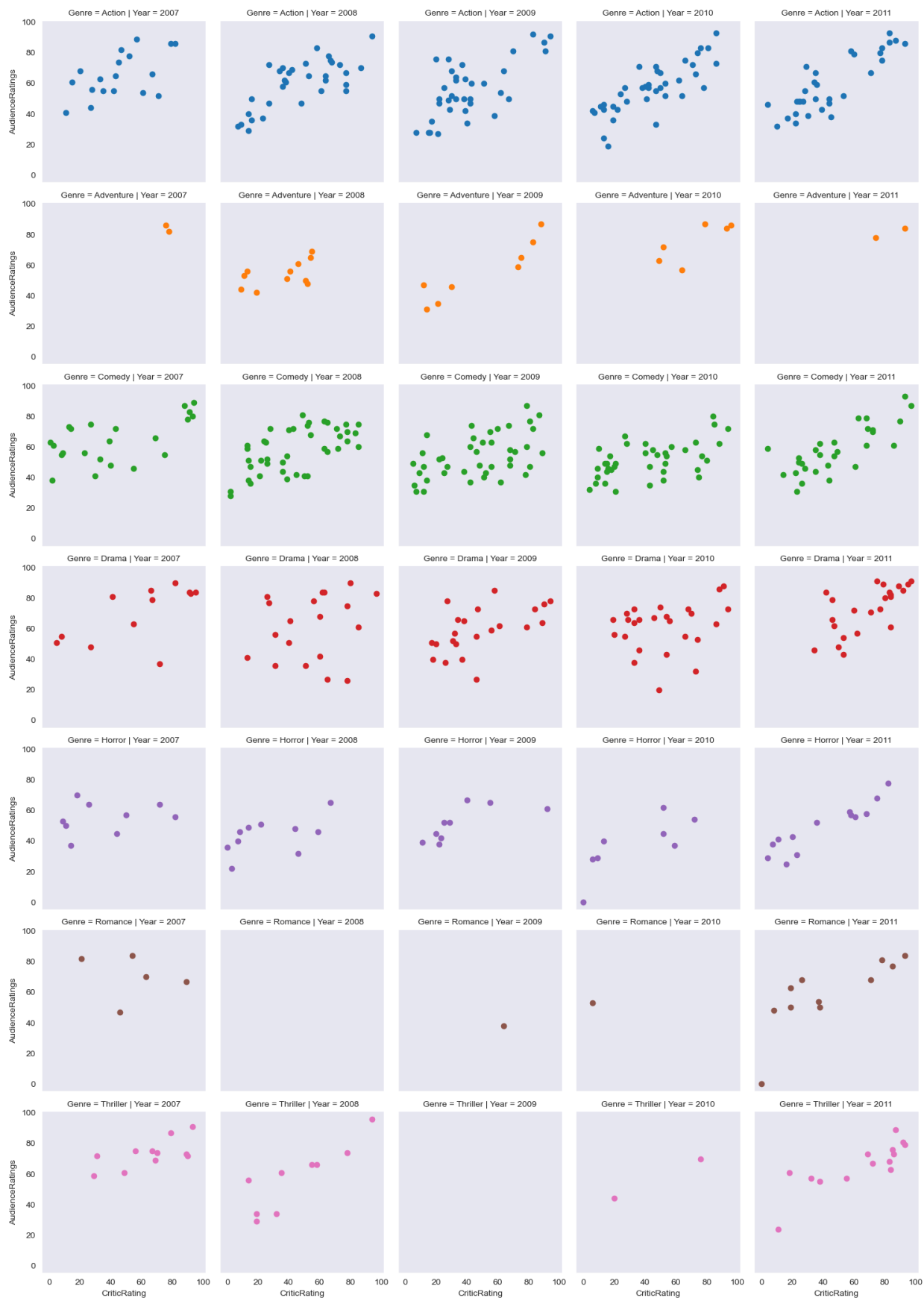


```
In [120]: plt.scatter(movies.CriticRating,movies.AudienceRatings)
```

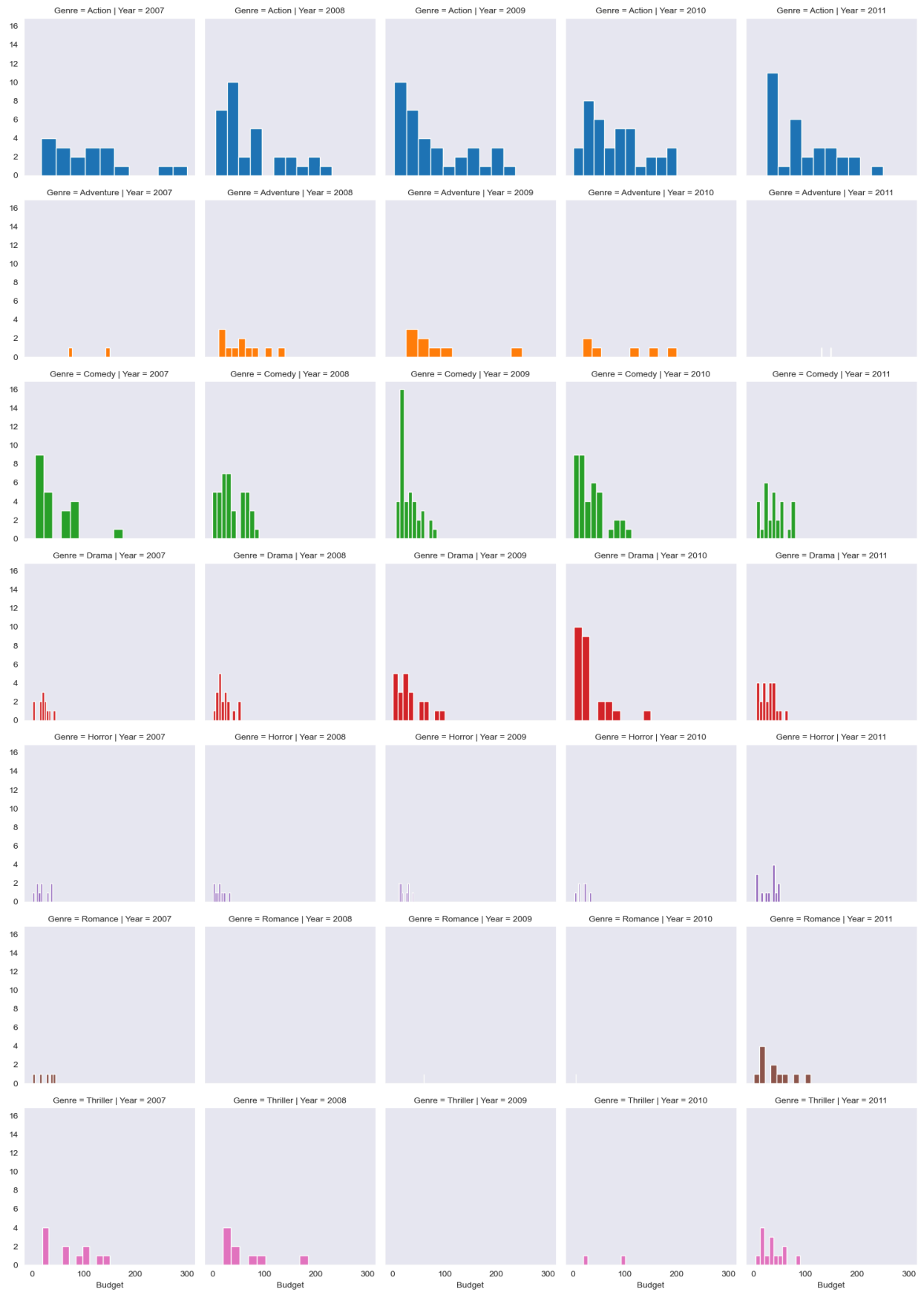
```
Out[120]: <matplotlib.collections.PathCollection at 0x22e1164f610>
```



```
In [122]: g = sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre')  
g = g.map(plt.scatter, 'CriticRating', 'AudienceRatings' ) #scatterplots are mapped in facetgrid
```

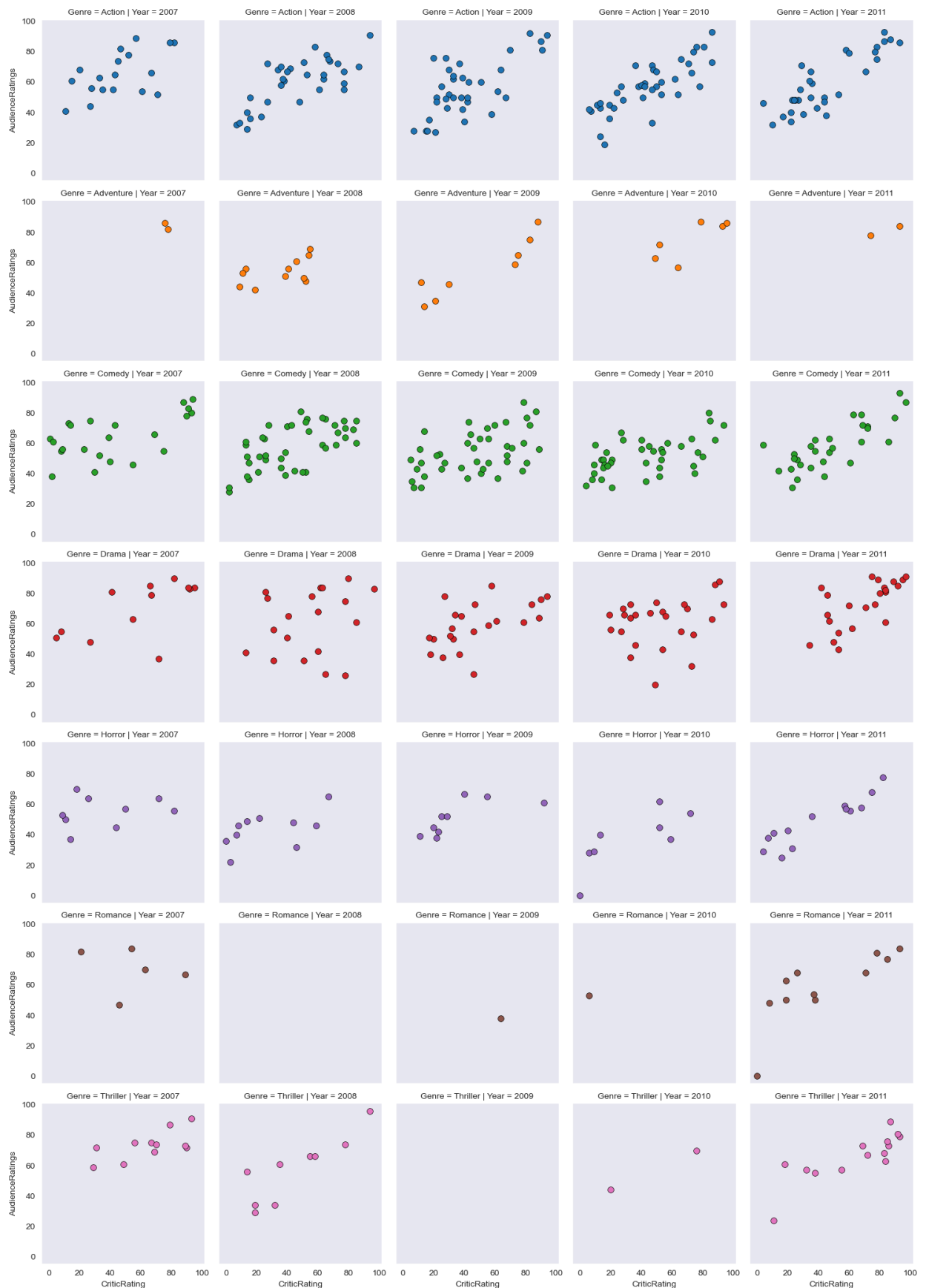


```
In [123]: g=sns.FacetGrid(movies,row='Genre',col='Year', hue='Genre')
g = g.map(plt.hist,'Budget')
```





```
In [124]: g = sns.FacetGrid (movies, row = 'Genre', col = 'Year', hue = 'Genre')
kws = dict(s=50, linewidth=0.5, edgecolor='black')
g = g.map(plt.scatter, 'CriticRating', 'AudienceRatings', **kws ) #scatterplots are mapped in facetgrid
```



```
In [131]: sns.set_style('darkgrid')
f, axes = plt.subplots(2,2, figsize=(15,15))

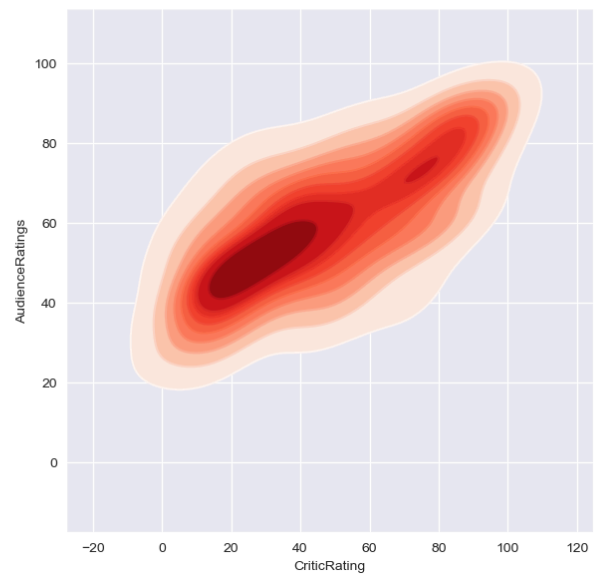
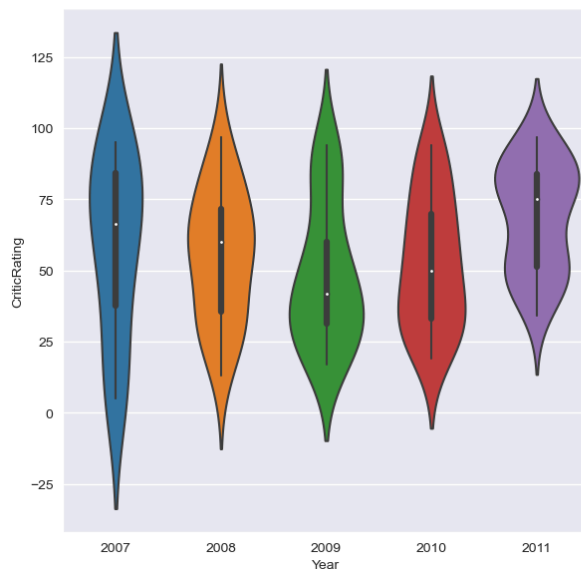
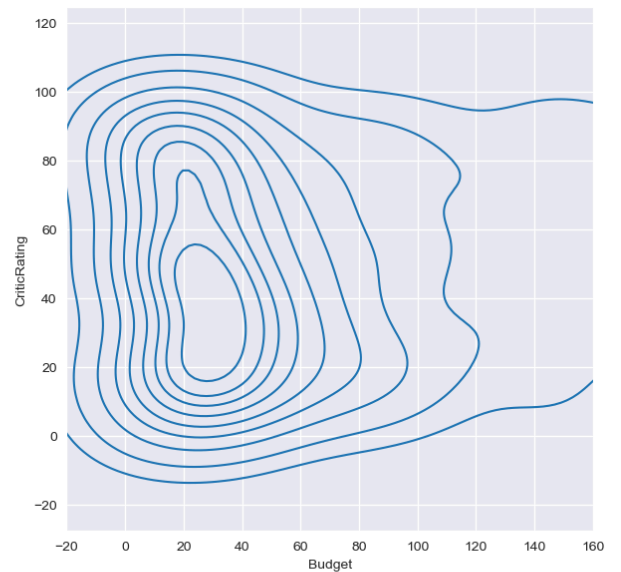
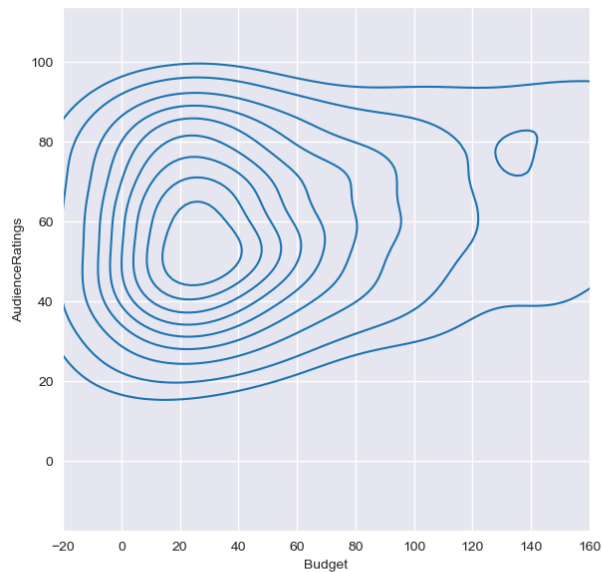
k1 = sns.kdeplot(data=movies, x='Budget', y='AudienceRatings', ax=axes[0,0])
k2 = sns.kdeplot(data=movies, x='Budget', y='CriticRating', ax=axes[0,1])

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

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

k4 = sns.kdeplot(data=movies, x='CriticRating', y='AudienceRatings', shade=True, shade_lowest=False, cmap='R')
k4b = sns.kdeplot(data=movies, x='CriticRating', y='AudienceRatings', cmap='Reds', ax=axes[1,1])

plt.show()
```



```
In [149]: sns.set_style('dark',{ 'axes.facecolor':'black'})
f, axes = plt.subplots (2,2, figsize = (15,15))

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

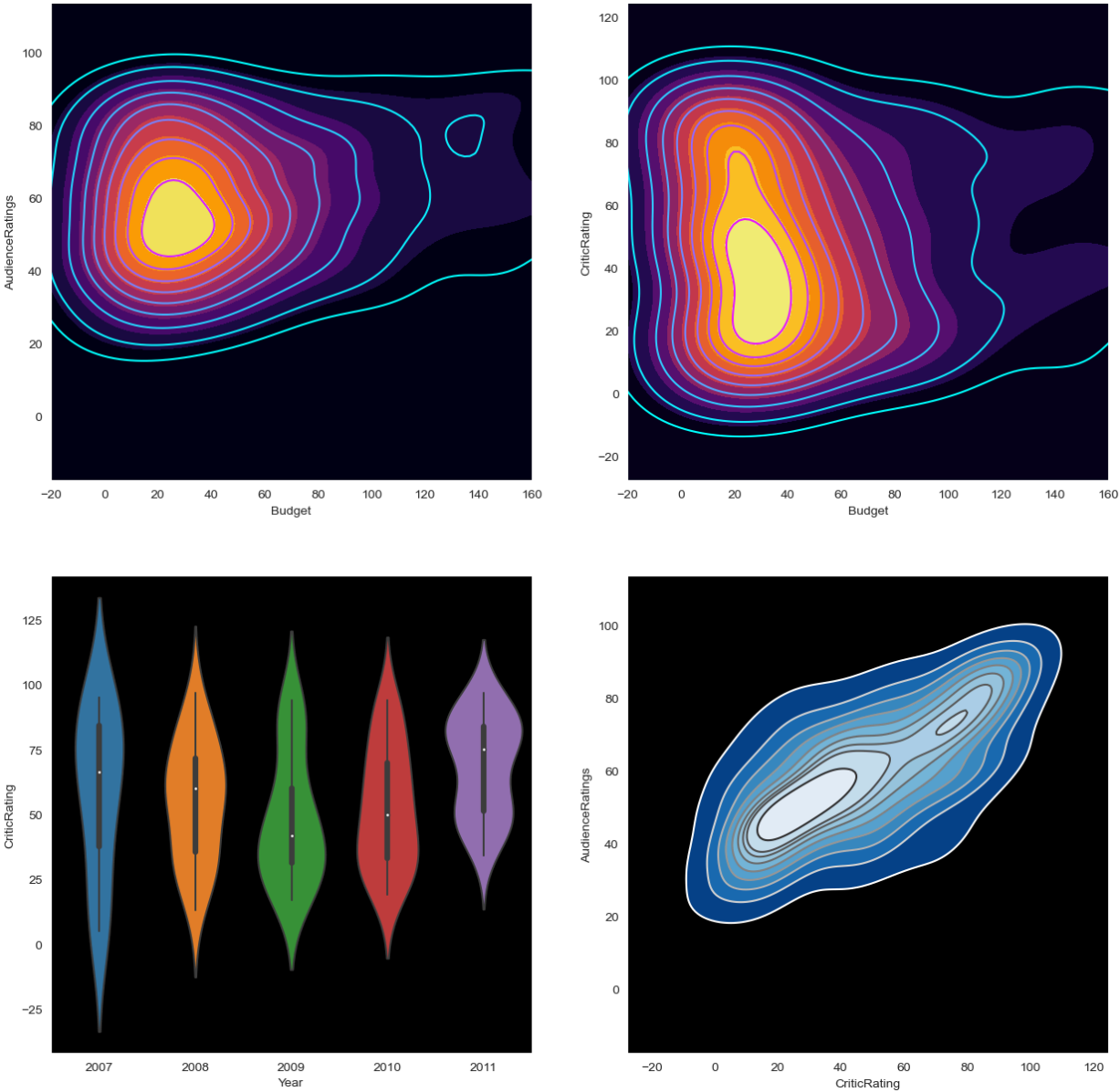
#plot [0,1]
k2 = sns.kdeplot(data=movies,x='Budget',y='CriticRating',\
                 shade=True, shade_lowest=True, cmap='inferno',\
                 ax = axes[0,1])
k2b = sns.kdeplot(data=movies,x='Budget',y='CriticRating',\
                 cmap = 'cool', ax = axes[0,1])

#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(data=movies,x='CriticRating',y='AudienceRatings', \
                 shade = True,shade_lowest=False,cmap='Blues_r', \
                 ax=axes[1,1])
k4b = sns.kdeplot( data=movies,x='CriticRating',y='AudienceRatings',\
                 cmap='gist_gray_r',ax = axes[1,1])

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

plt.show()
```



In [150]: #EDA is completed.....

In [ ]: