In [3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

In [4]:
```
movie=pd.read_csv(r"D:\Full Stack Data Science\17 Aug\17th\MOVIE RATINGS _ ADVANCE VISUALIZATION
movie
```

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]:
```
movie.shape
```

Out[5]:
```
(559, 6)
```

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

Out[7]:
```
Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
       'Budget (million $)', 'Year of release'],
      dtype='object')
```

In [8]:

```
 # Rename the columns name
movie.columns=['Film','Genre','CriticRating','AudienceRatings','BudgetMillions','Year']
```

In [9]:

```
movie.columns
```

Out[9]:

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

In [10]:

```
movie.head()
```

Out[10]:

	Film	Genre	CriticRating	AudienceRatings	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 [11]:

```
movie.isnull().sum()
```

Out[11]:

```
Film               0
Genre              0
CriticRating       0
AudienceRatings    0
BudgetMillions     0
Year               0
dtype: int64
```

In [12]:

```
movie.describe() #Descriptive Statistics
```

Out[12]:

	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

In [13]:

```
movie.Film # To read column from data
```

Out[13]:

```
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 [14]:

```python
movie.Film=movie.Film.astype('category')
movie.Genre=movie.Genre.astype('category')
movie.Year=movie.Year.astype('category')
```

In [15]:

```python
movie.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
```

In [16]:

```python
movie.describe()
```

Out[16]:

	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

In [17]:

```python
movie.Film.cat.categories
```

Out[17]:

```
Index(['(500) Days of Summer ', '10,000 B.C.', '12 Rounds ', '127 Hours',
       '17 Again ', '2012', '27 Dresses', '30 Days of Night',
       '30 Minutes or Less', '50/50',
       ...
       'Yes Man', 'Yogi Bear', 'You Again', 'You Don't Mess with the Zohan',
       'You Will Meet a Tall Dark Stranger', 'Your Highness',
       'Youth in Revolt', 'Zodiac', 'Zombieland ', 'Zookeeper'],
      dtype='object', length=559)
```

In [18]:

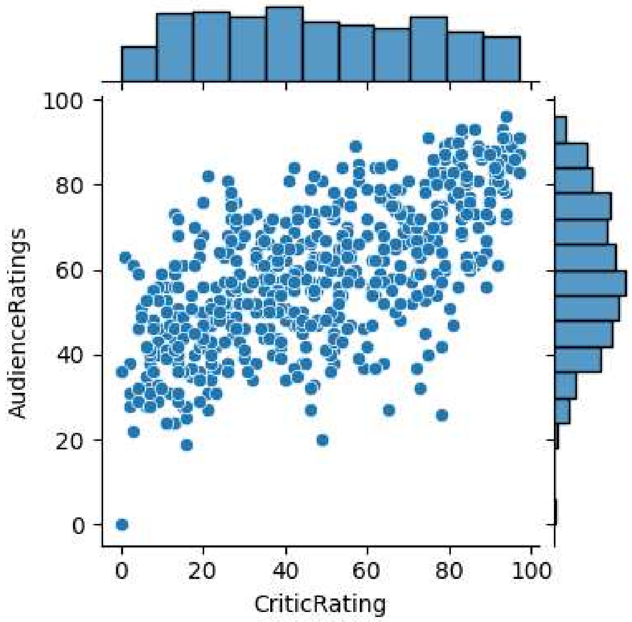
```python
movie.Genre.cat.categories
```

Out[18]:

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

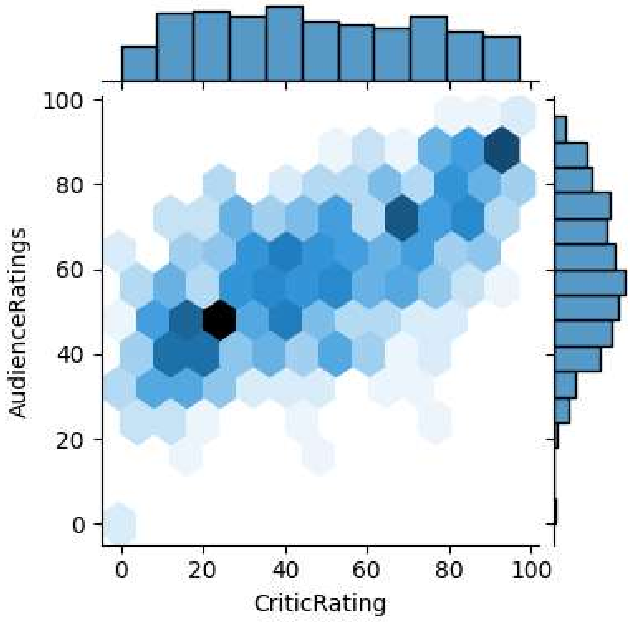
In [19]:

```python
plot1=sns.jointplot(data=movie,x='CriticRating', y='AudienceRatings', height=4) # height- change
```



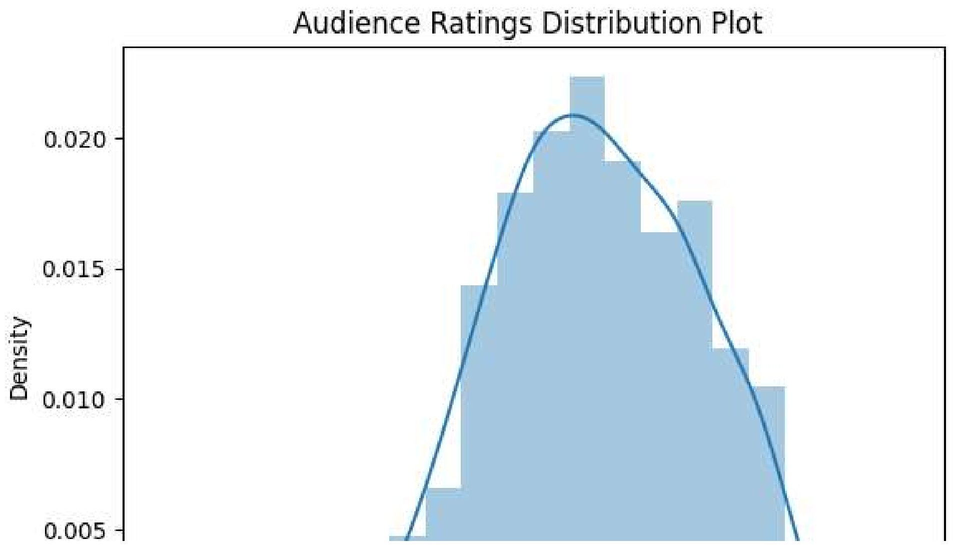
In [20]:

```python
plot1=sns.jointplot(data=movie,x='CriticRating', y='AudienceRatings',kind='hex',height=4)
```



Distribution Plot

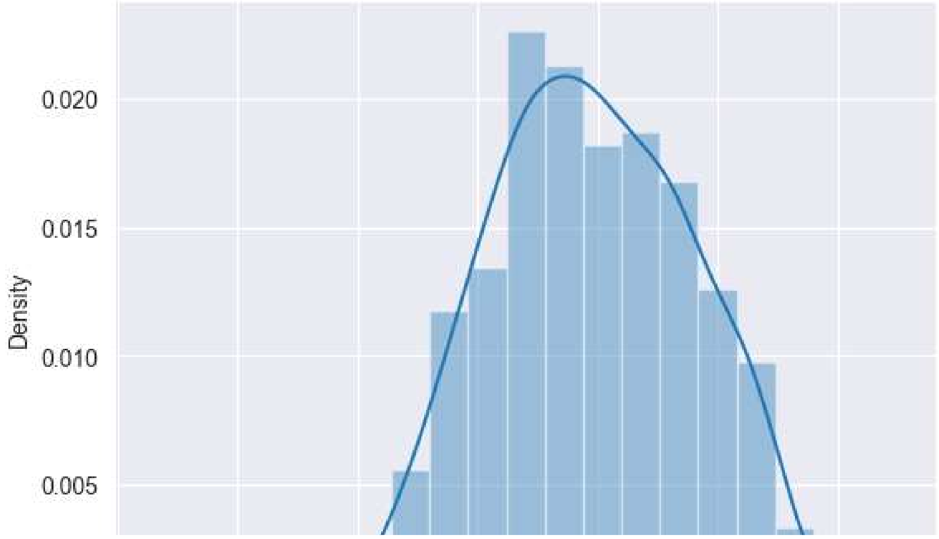
In [21]:
```python
vis2=sns.distplot(movie.AudienceRatings)
vis2.set_title('Audience Ratings Distribution Plot ');
```



In [22]:
```python
## To set style of Graph
sns.set_style('darkgrid')
```

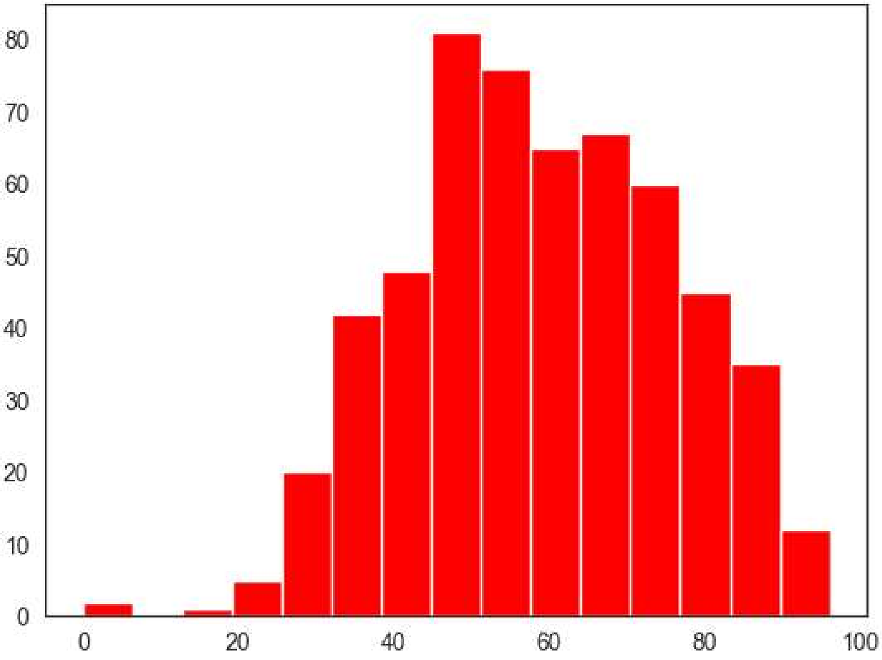
Normal Distribution/Gaussian Distribution/Bell curve

In [23]:
```python
vis2=sns.distplot(movie.AudienceRatings,bins=15)   # bins- number of bars
```



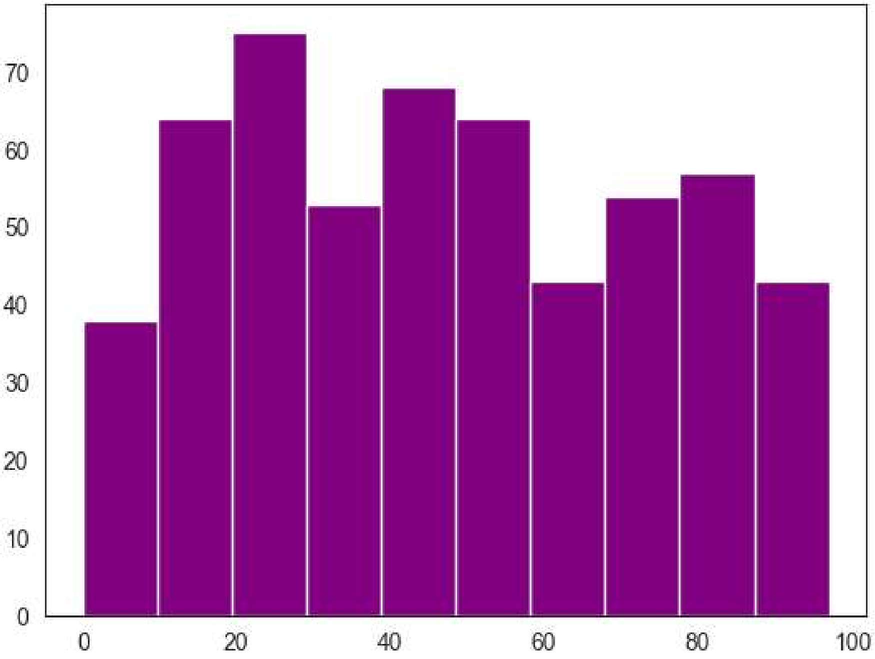
In [24]:

```python
sns.set_style('white')
vis2=plt.hist(movie.AudienceRatings,bins=15,color='r')
```



In [25]:

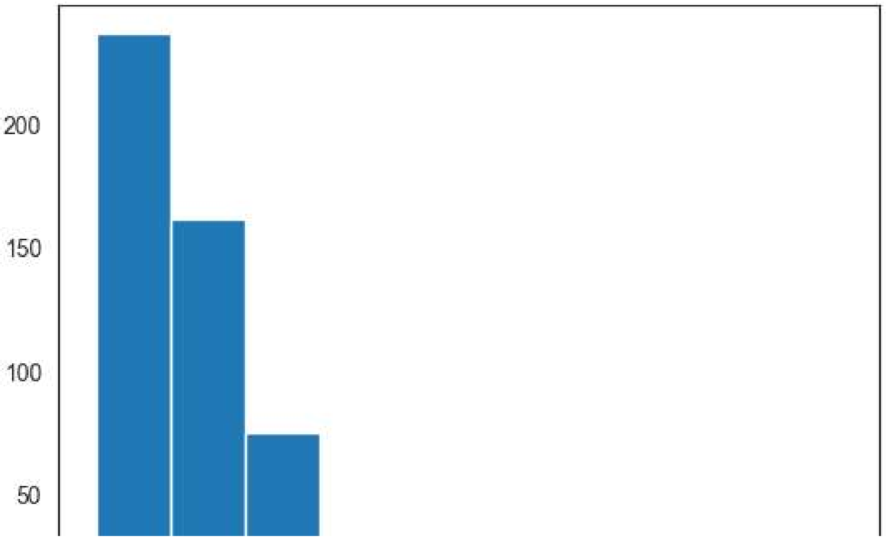
```python
vis3=plt.hist(movie.CriticRating,bins=10,color='purple')  # Uniform Distribution
```



Histogram of budget

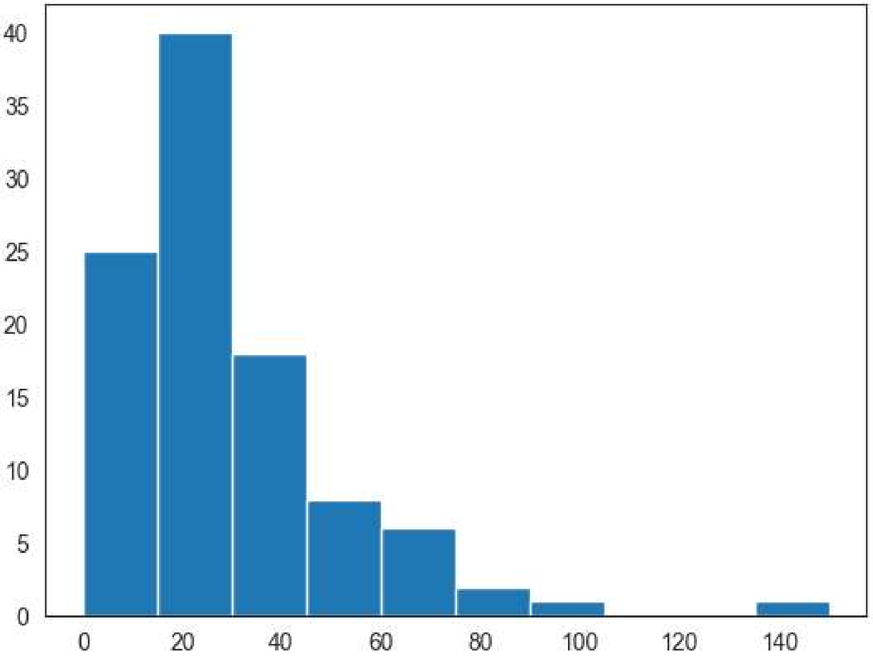
In [26]:

```
vis4=plt.hist(movie.BudgetMillions)
```



In [27]:

```
# From movie dataset we take Genre column and
# from that we take drama genre with corresponding budget to plot histogram.

plt.hist(movie[movie.Genre=='Drama'].BudgetMillions);
```



In [28]:

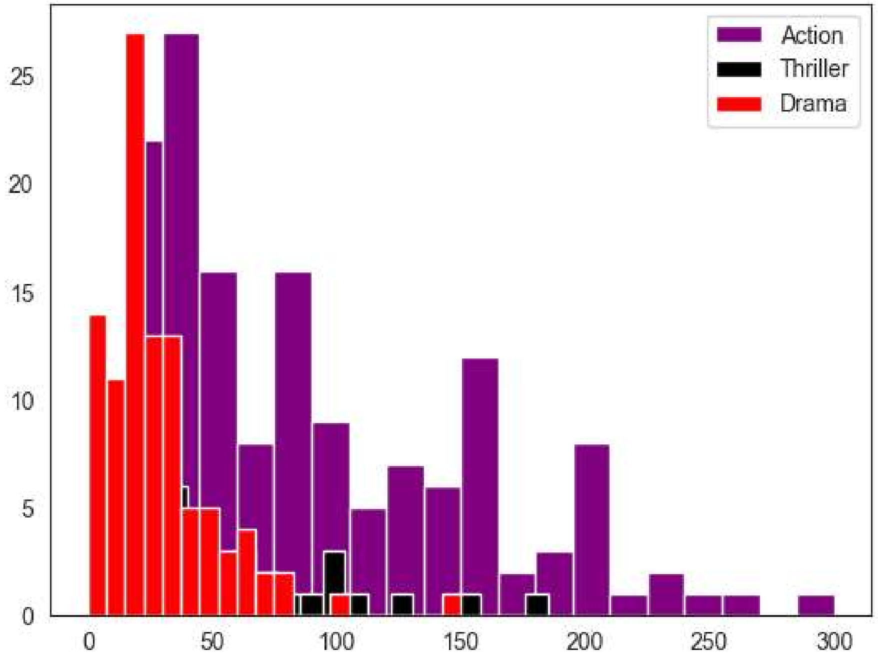
```
movie.Genre.unique()  # This gives unique elements from Genre column
```

Out[28]:

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

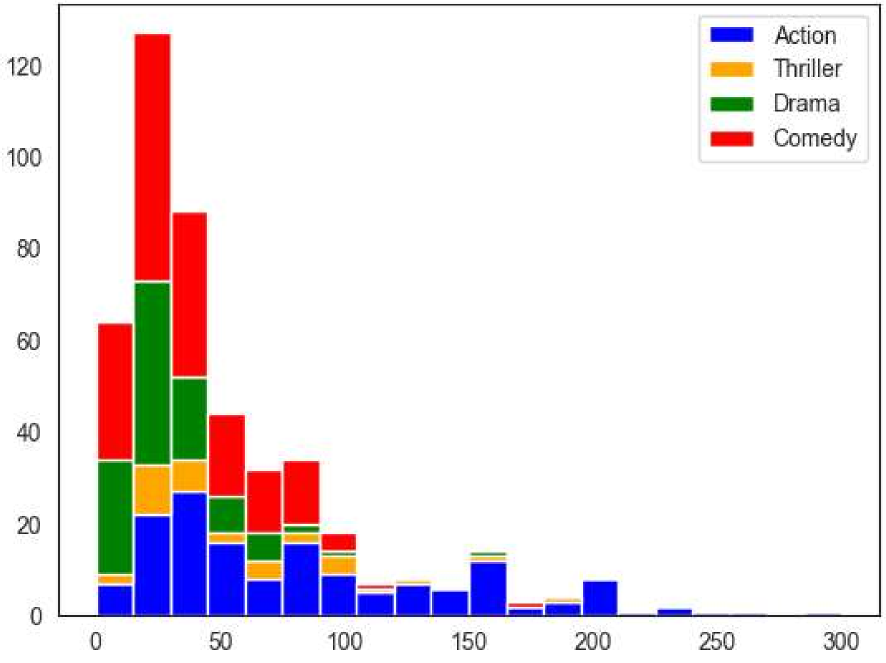
In [29]:

```python
# Below plots are stacked histogram because overlaped
plt.hist(movie[movie.Genre=='Action'].BudgetMillions,bins=20,color='purple',label='Action')
plt.hist(movie[movie.Genre=='Thriller'].BudgetMillions,bins=20,color='k',label='Thriller')
plt.hist(movie[movie.Genre=='Drama'].BudgetMillions,bins=20,color='r',label='Drama')
plt.legend()
plt.show()
```



In [30]:

```python
# Below plots are stacked histogram because overlaped
plt.hist([movie[movie.Genre=='Action'].BudgetMillions,\
          movie[movie.Genre=='Thriller'].BudgetMillions,\
          movie[movie.Genre=='Drama'].BudgetMillions,\
          movie[movie.Genre=='Comedy'].BudgetMillions],bins=20,stacked=True,label=('Action','Thr
plt.legend()
plt.show()
```



In [31]:

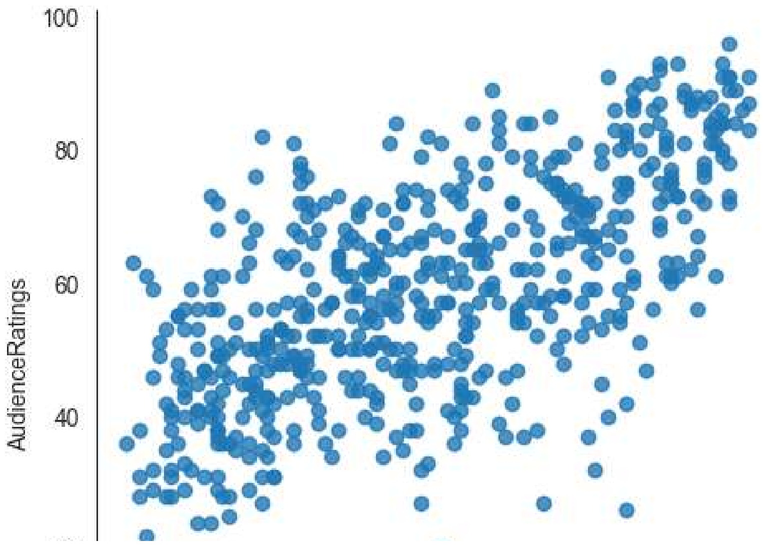
```
### if we have 100 categories we cannot copy & paste all the things

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

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

Linear Model Plot

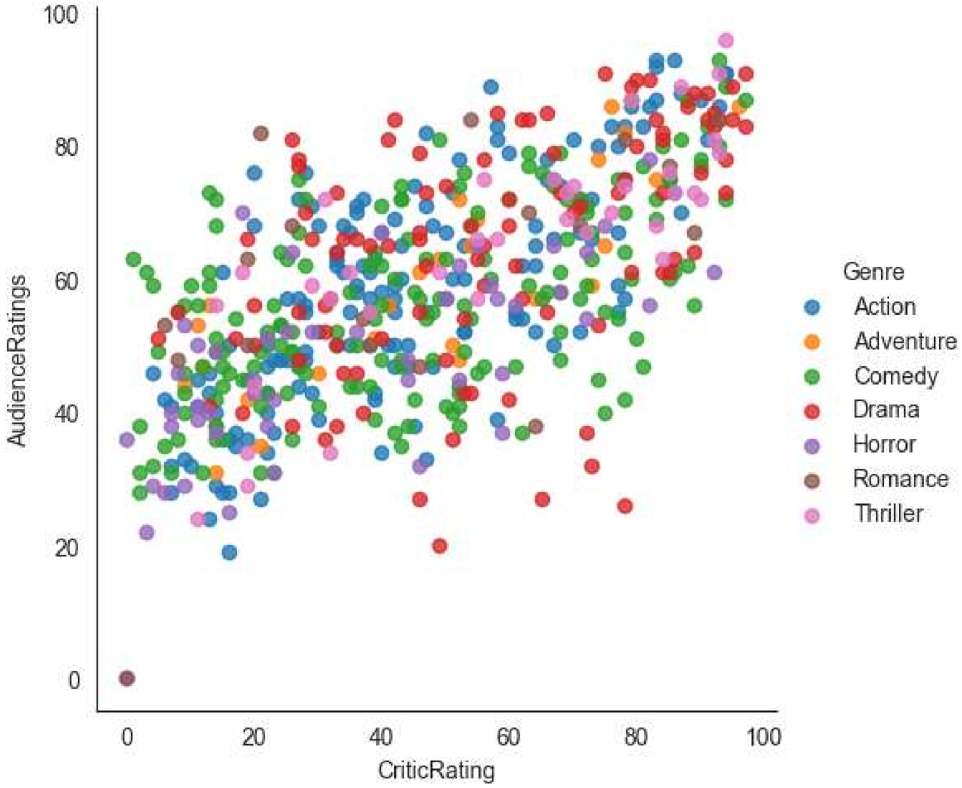
In [32]:

```
vis5=sns.lmplot(data=movie,x='CriticRating',y='AudienceRatings',fit_reg=False)
```



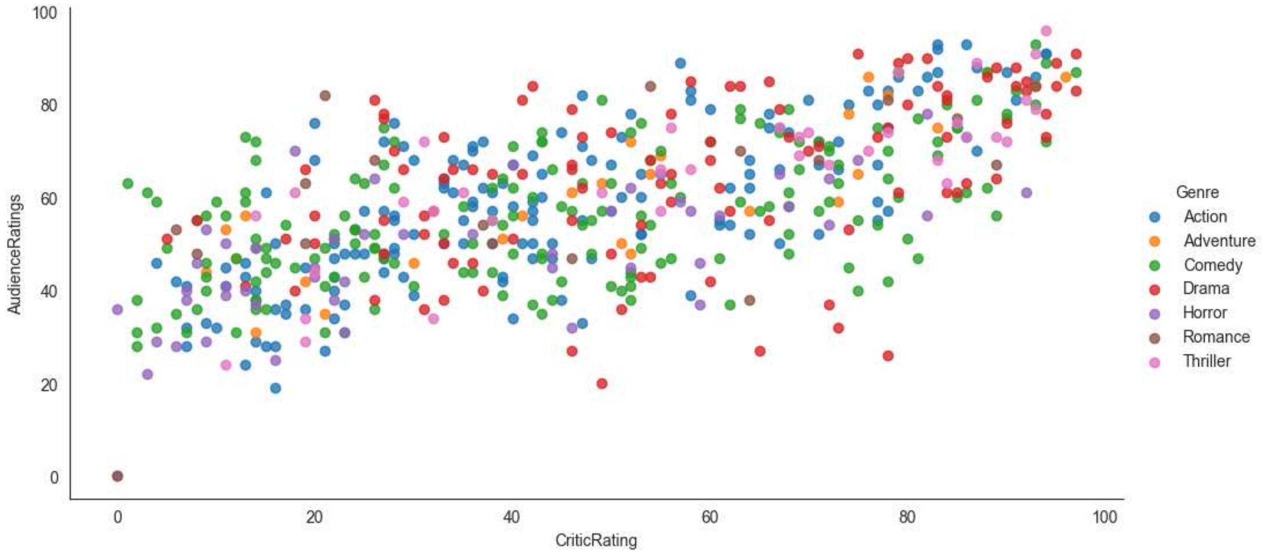
In [33]:

```
vis5=sns.lmplot(data=movie,x='CriticRating',y='AudienceRatings',fit_reg=False,hue='Genre')
```



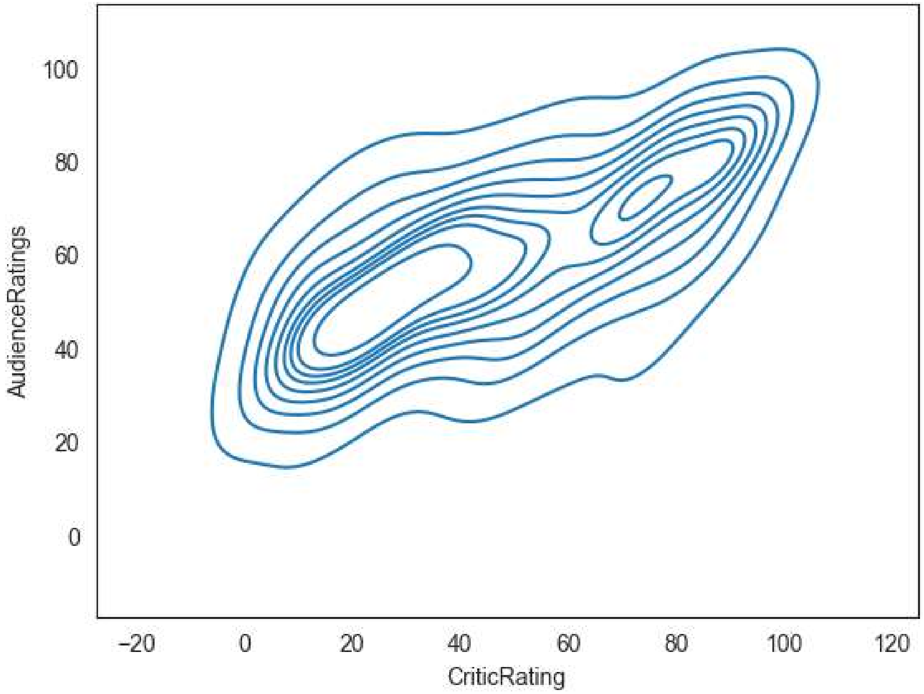
In [34]:

```
vis5=sns.lmplot(data=movie,x='CriticRating',y='AudienceRatings',fit_reg=False,hue='Genre',aspect
```

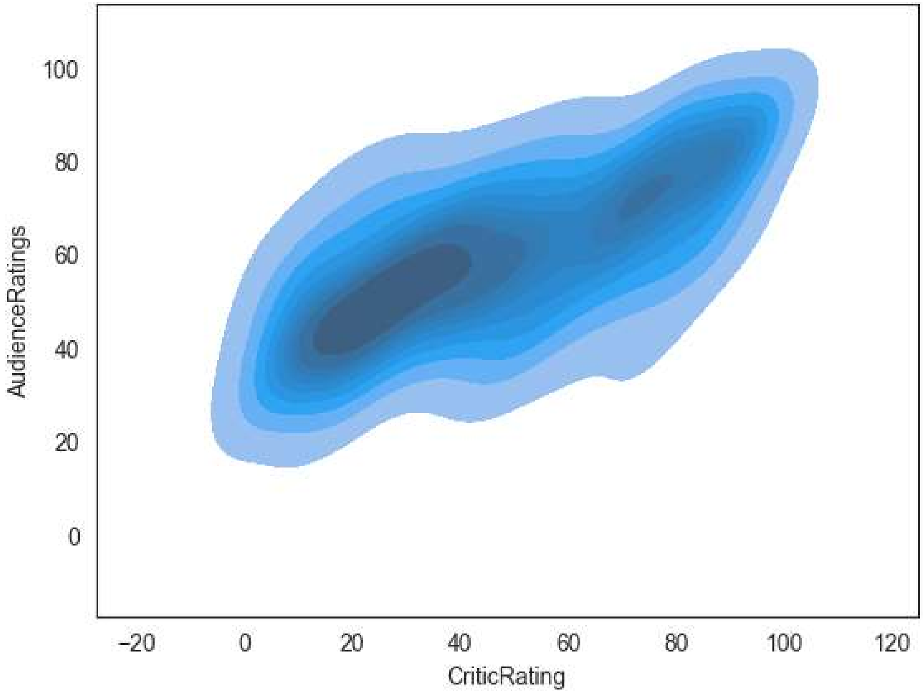


Kernal Density Estimate Plot(KDE Plot)

```
In [35]:  vis6=sns.kdeplot(data=movie,x='CriticRating',y='AudienceRatings')
```

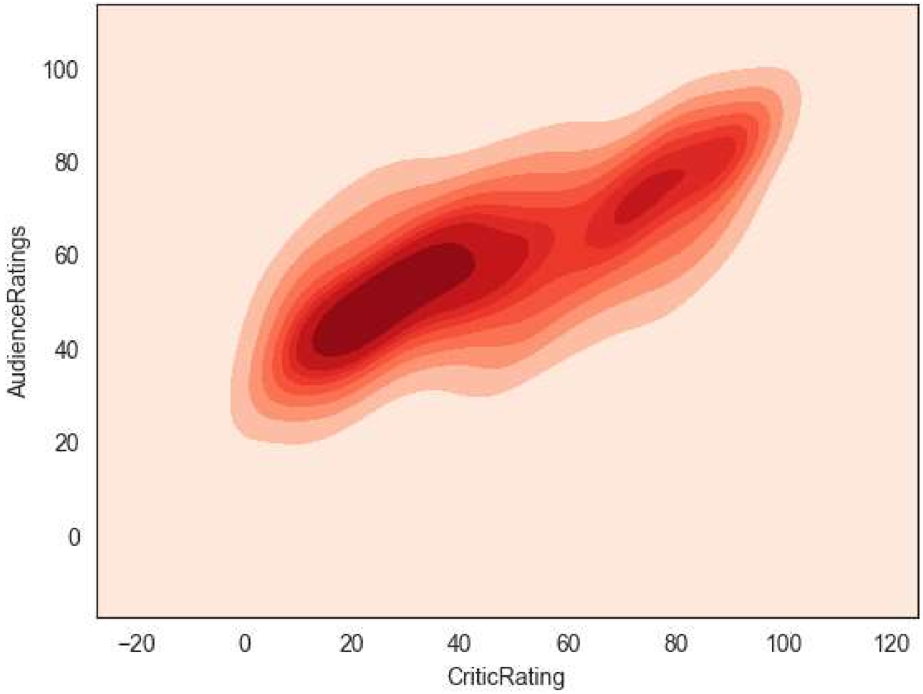


```
In [36]:  vis6=sns.kdeplot(data=movie,x='CriticRating',y='AudienceRatings',shade=True)
```



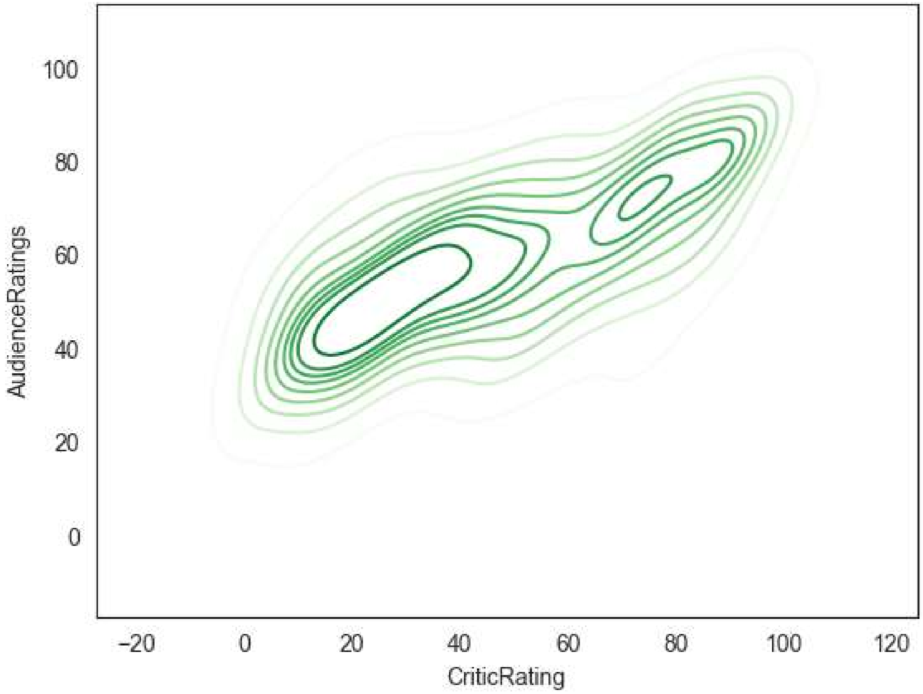
In [37]:

```
vis6=sns.kdeplot(data=movie,x='CriticRating',y='AudienceRatings',shade=True,shade_lowest=True,cma
```



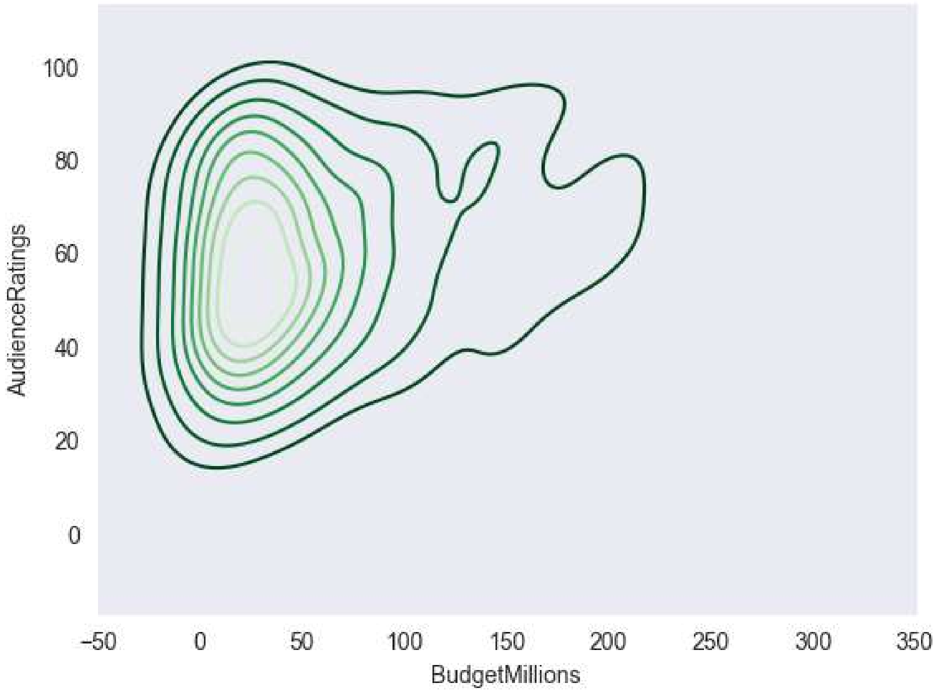
In [38]:

```
# Change the color of graph
vis6=sns.kdeplot(data=movie,x='CriticRating',y='AudienceRatings',shade_lowest=False,cmap='Greens
```



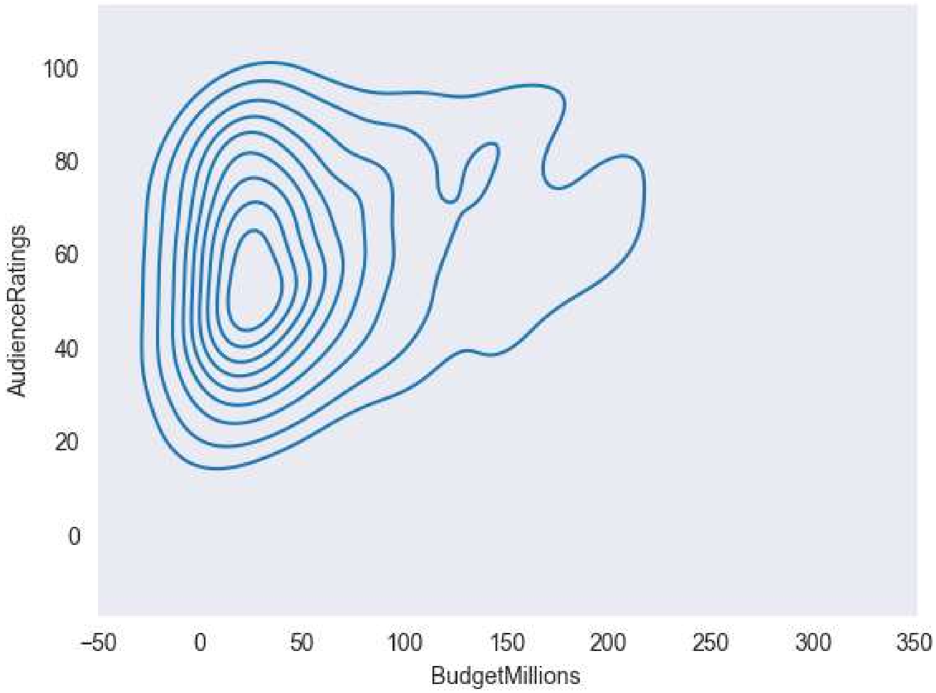
In [39]:

```
sns.set_style('dark')
k1=sns.kdeplot(data=movie,x='BudgetMillions',y='AudienceRatings',shade_lowest=False,cmap='Greens_
```



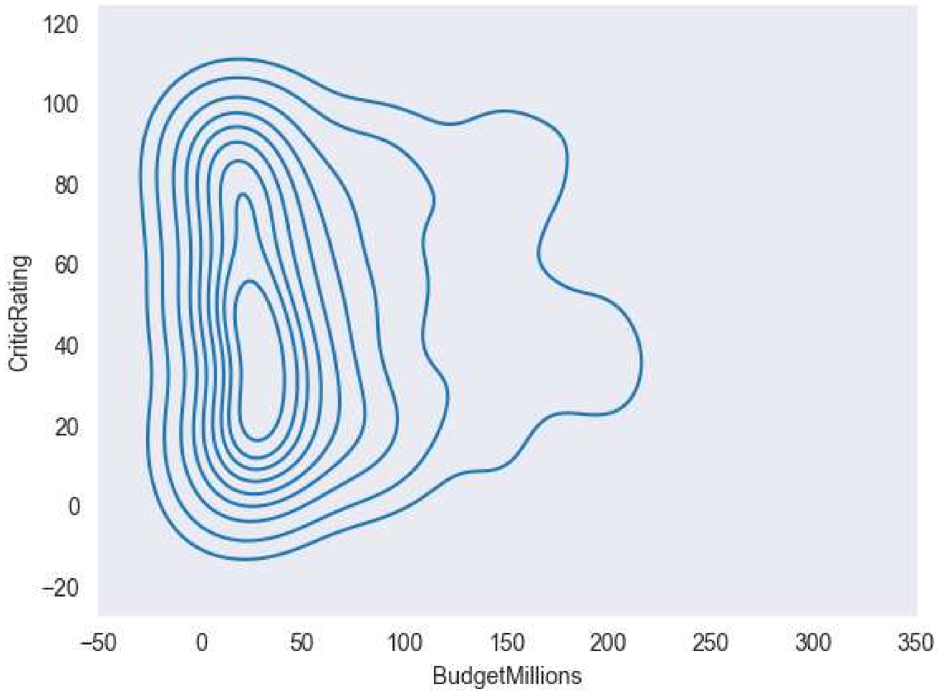
In [40]:

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



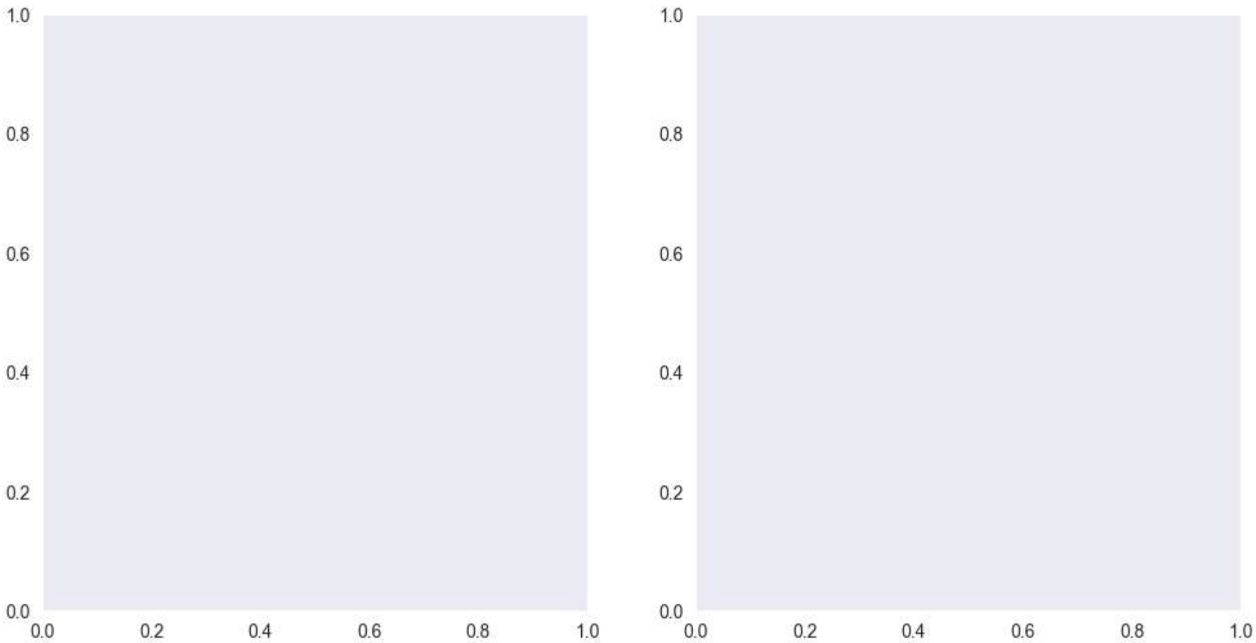
In [41]:

```
k2=sns.kdeplot(data=movie,x='BudgetMillions',y='CriticRating')
```



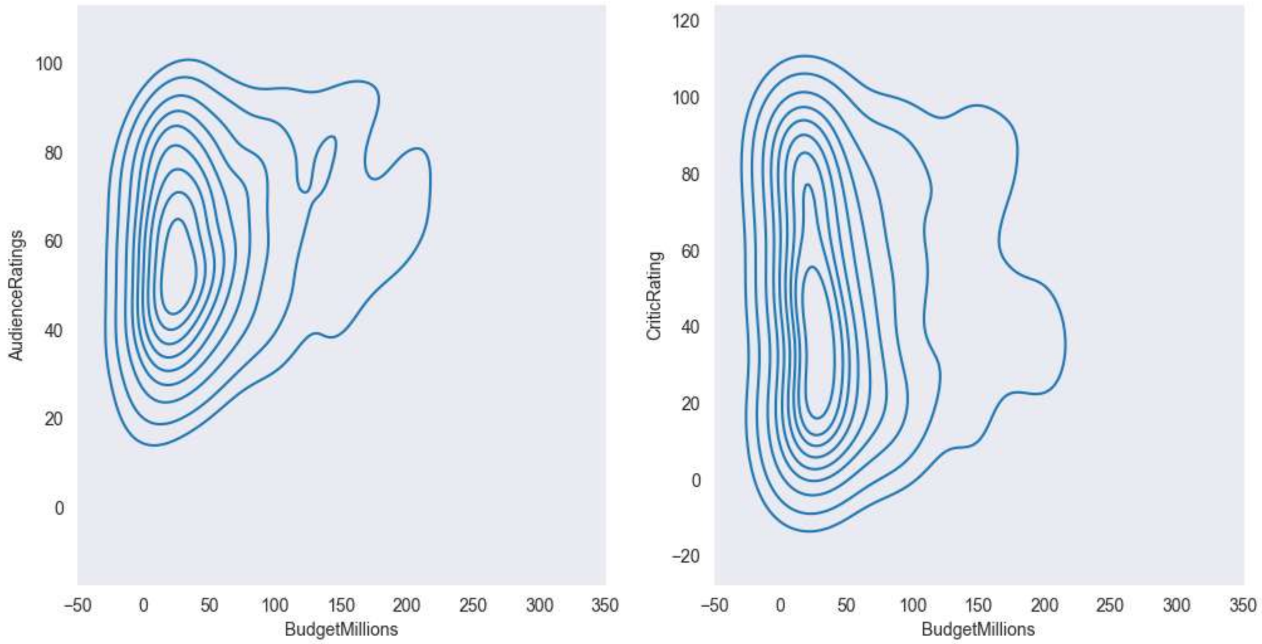
In [42]:

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



In [43]:

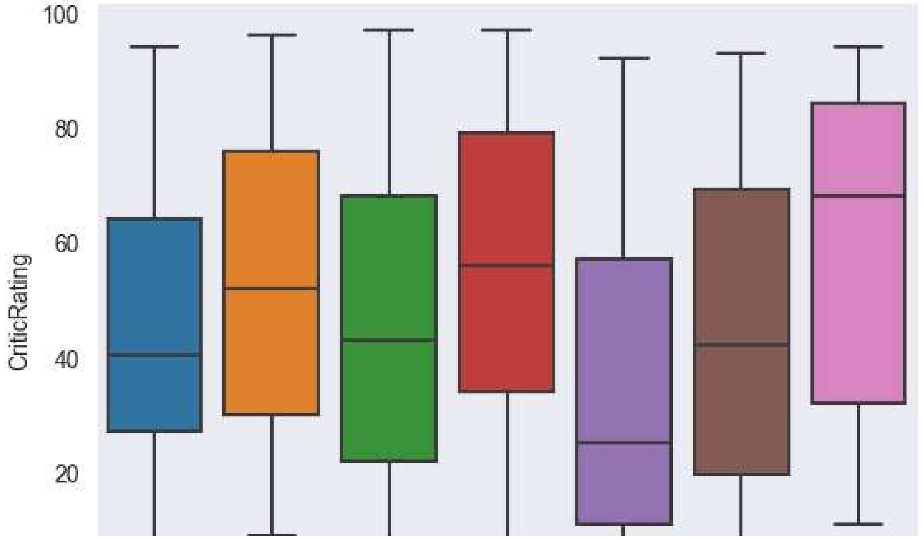
```python
f,axes=plt.subplots(1,2,figsize=(12,6))

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



Box plots- To find outliers from data

In [44]:

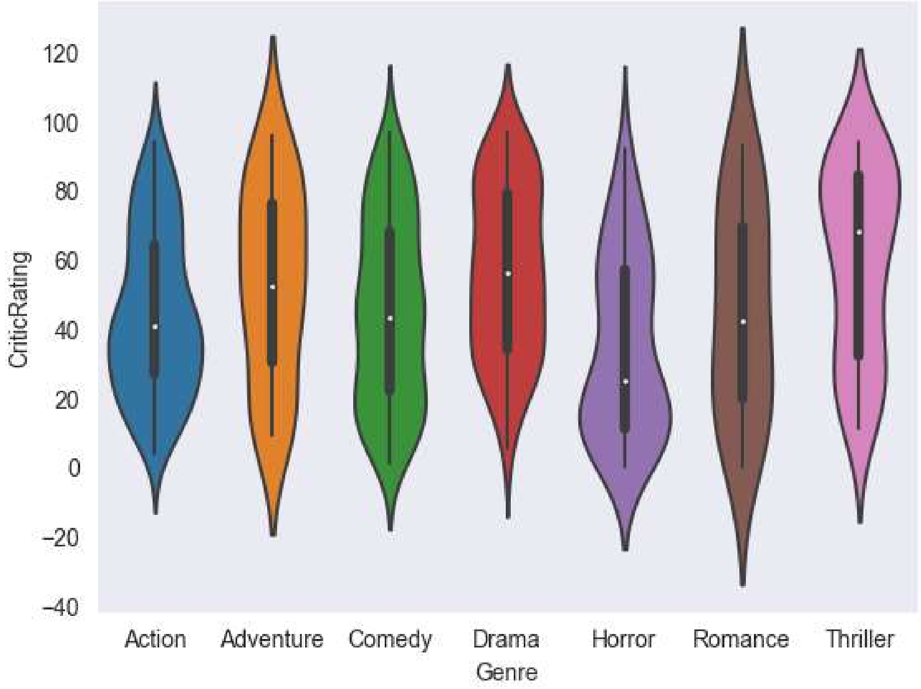
```python
vis6=sns.boxplot(data=movie,x='Genre',y='CriticRating')
```



Violin plot

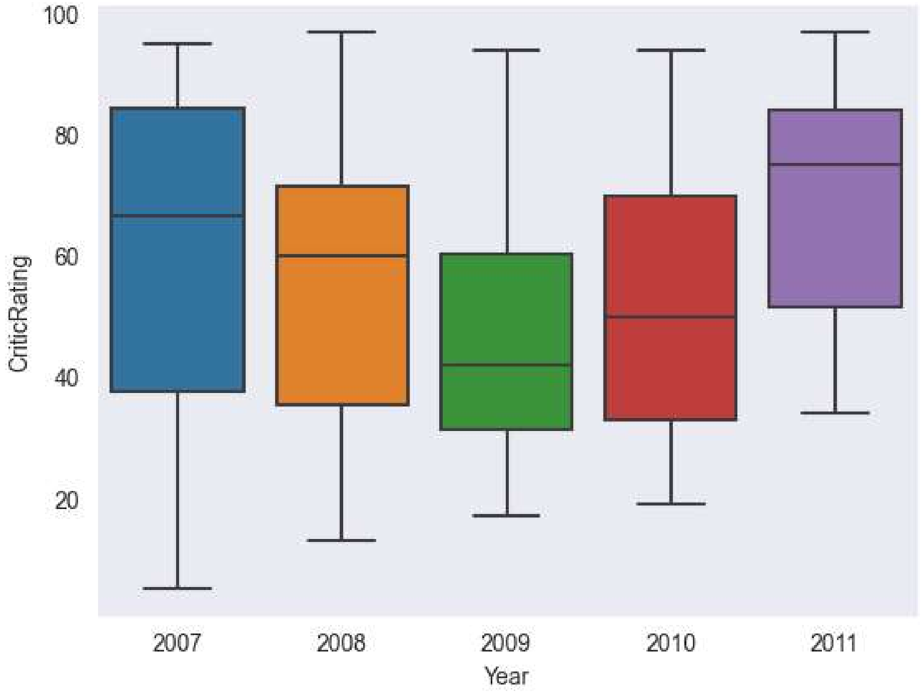
In [45]:

```
vis7=sns.violinplot(data=movie,x='Genre',y='CriticRating')
```



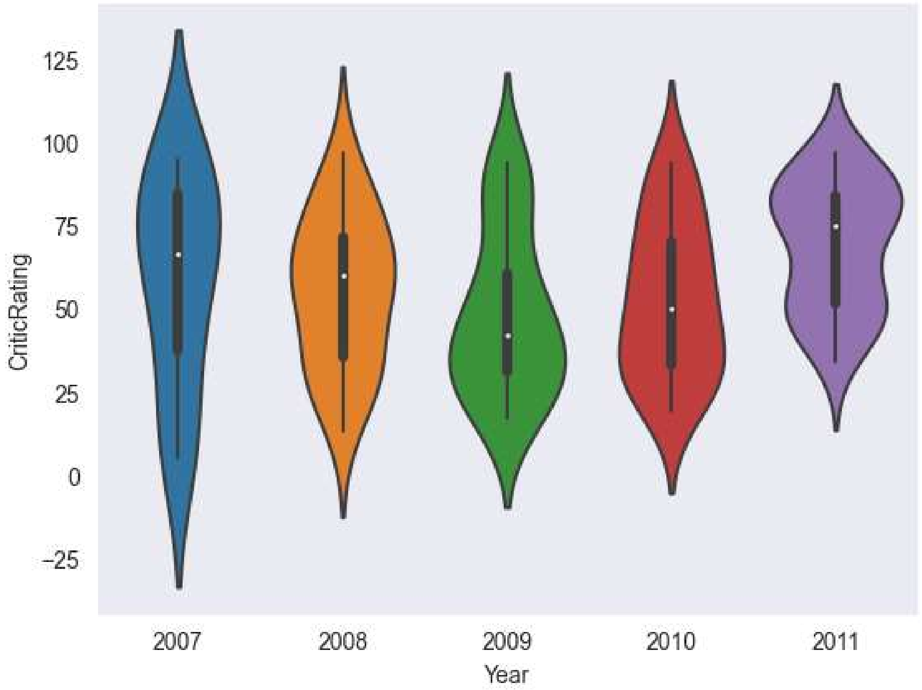
In [46]:

```
vis6_1=sns.boxplot(data=movie[movie.Genre=='Drama'],x='Year',y='CriticRating')
```



In [47]:

```
vis7_1=sns.violinplot(data=movie[movie.Genre=='Drama'],x='Year',y='CriticRating')
```



Creating a Facet Grid

In [48]:

```
movie.Genre.cat.categories
```

Out[48]:

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

In [49]:

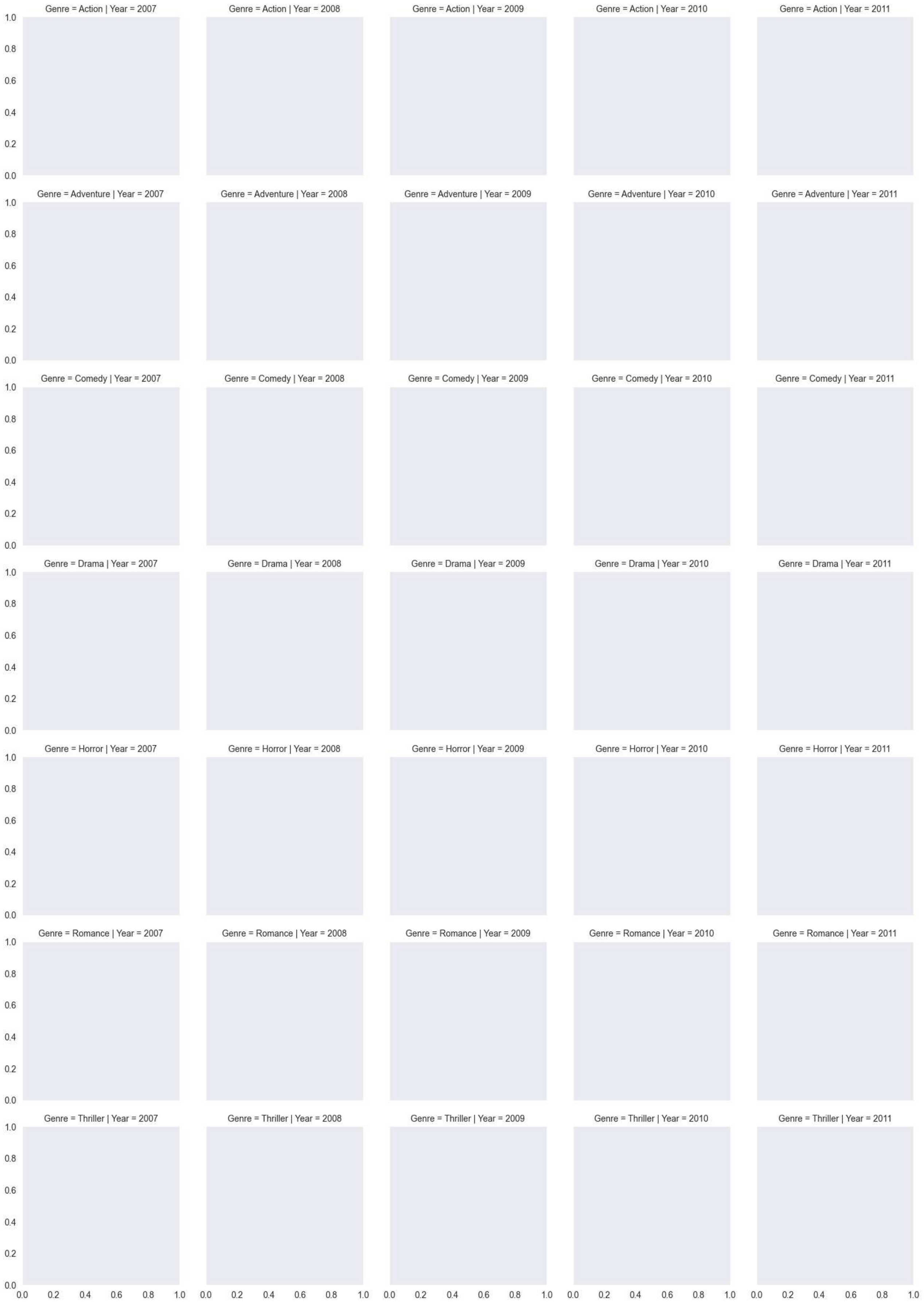
```
movie.Year.cat.categories
```

Out[49]:

```
Index([2007, 2008, 2009, 2010, 2011], dtype='int64')
```

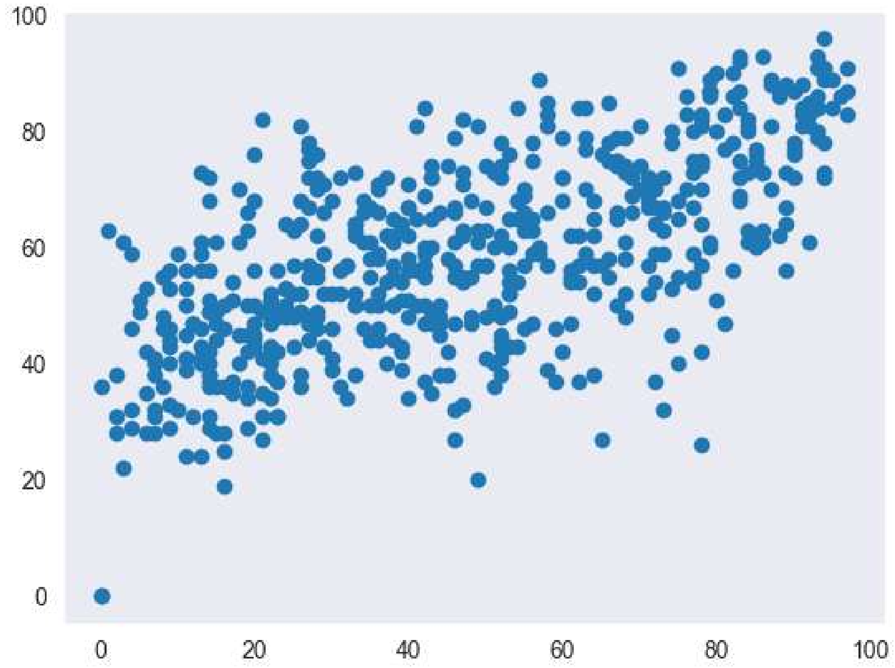
In [50]:

```
vis8=sns.FacetGrid(movie,row='Genre',col='Year',hue='Genre')
```



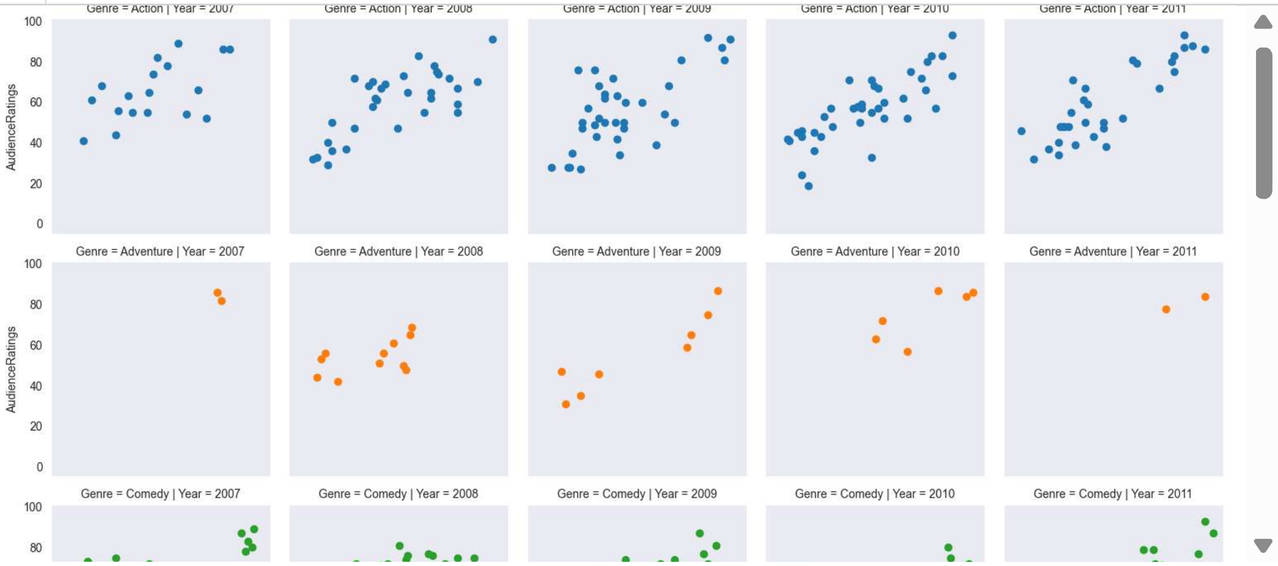
In [51]:
```
plt.scatter(movie.CriticRating,movie.AudienceRatings)
```

Out[51]:
```
<matplotlib.collections.PathCollection at 0x1d5a95e8850>
```



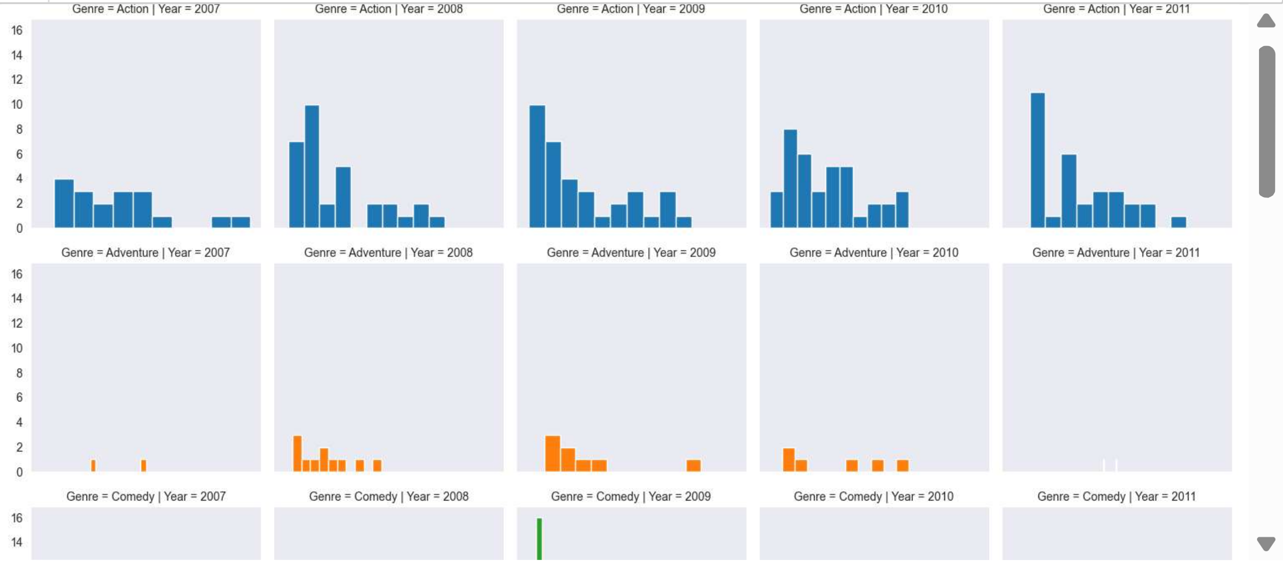
Scatter plots are mapped in FacetGrid

In [52]:
```
vis8=sns.FacetGrid(movie,row='Genre',col='Year',hue='Genre')
vis8=vis8.map(plt.scatter,'CriticRating','AudienceRatings')
```

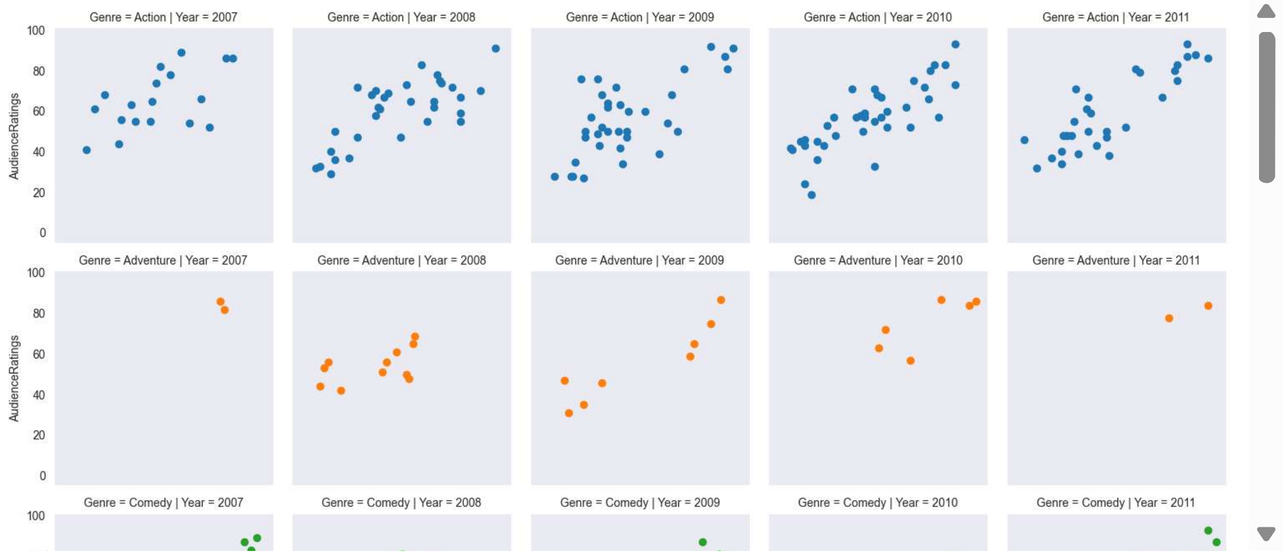


Histogram is mapped in FacetGrid

In [53]:
```
vis9=sns.FacetGrid(movie,row='Genre',col='Year',hue='Genre')
vis9=vis9.map(plt.hist,'BudgetMillions')
```



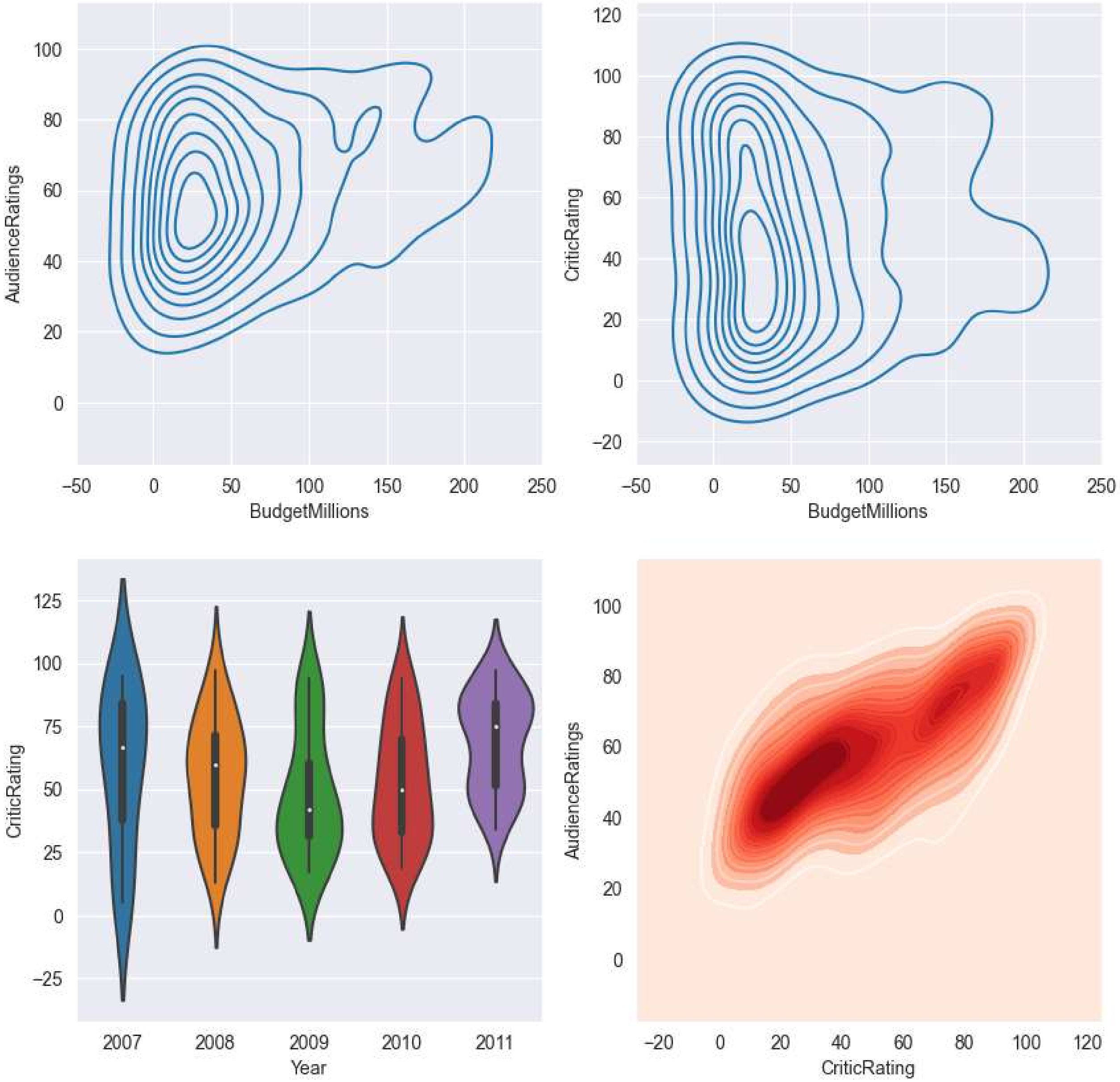
In [54]:
```
#scatterplots are mapped in facetgrid
vis8=sns.FacetGrid(movie,row='Genre',col='Year',hue='Genre')
kws=dict(s=50,linewidth=0.5,edgecolor='Black')
vis8=vis8.map(plt.scatter,'CriticRating','AudienceRatings')
```



Building a Dashboard (Dashboard-Combination of plots)

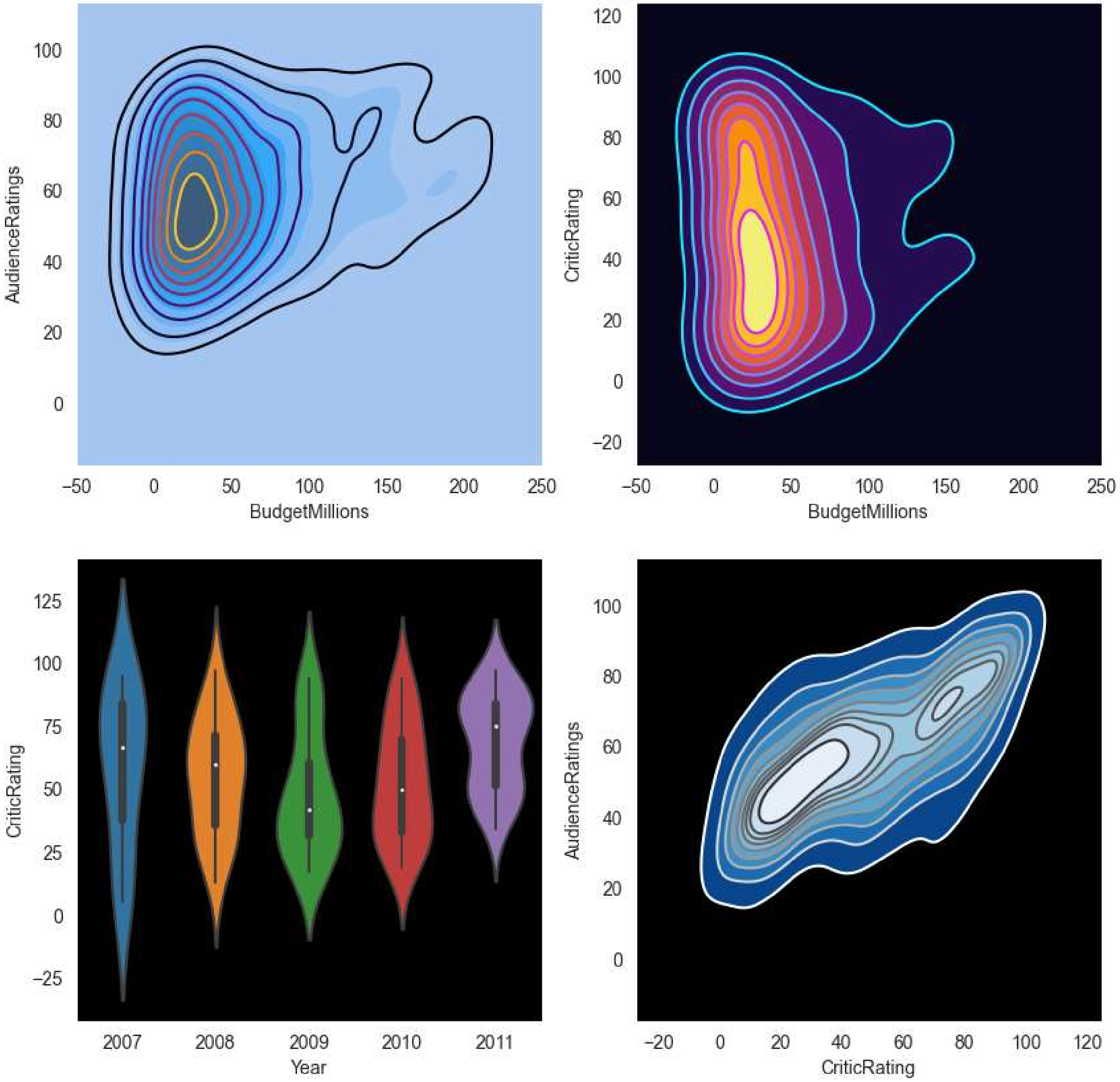
In [55]:

```
sns.set_style('darkgrid')
f,axes=plt.subplots(2,2,figsize=(10,10))


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

k1.set(xlim=(-50,250))
k2.set(xlim=(-50,250))

vis7_1=sns.violinplot(data=movie[movie.Genre=='Drama'],x='Year',y='CriticRating',ax=axes[1,0])

vis6=sns.kdeplot(data=movie,x='CriticRating',y='AudienceRatings',shade=True,shade_lowest=True,cma
vis6=sns.kdeplot(data=movie,x='CriticRating',y='AudienceRatings',cmap='Reds',axes=[1,1])
plt.show()
```



In [56]:

```
# Building a Dashboard (Dashboard-Combination of plots)

sns.set_style('dark',{'axes.facecolor':'black'})
f,axes=plt.subplots(2,2,figsize=(10,10))


k1=sns.kdeplot(data=movie,x='BudgetMillions',y='AudienceRatings',cmap='inferno',ax=axes[0,0])
k1b=sns.kdeplot(data=movie,x='BudgetMillions',y='AudienceRatings',shade=True,shade_lowest=True,a

k2=sns.kdeplot(data=movie,x='BudgetMillions',y='CriticRating',shade=True,shade_lowest=True,cmap=
k2b=sns.kdeplot(data=movie,x='BudgetMillions',y='CriticRating',shade_lowest=True,cmap='cool',ax=


k1.set(xlim=(-50,250))
k2.set(xlim=(-50,250))

vis7_1=sns.violinplot(data=movie[movie.Genre=='Drama'],x='Year',y='CriticRating',ax=axes[1,0])

k3=sns.kdeplot(data=movie,x='CriticRating',y='AudienceRatings',shade=True,shade_lowest=False,cma
k4=sns.kdeplot(data=movie,x='CriticRating',y='AudienceRatings',cmap='gist_gray_r',axes=[1,1])
plt.show()
```



In [ ]: