In [2]:
```
!pip install pandas
```

```
Requirement already satisfied: pandas in /opt/anaconda3/lib/python3.12/site-
packages (2.2.3)
Requirement already satisfied: numpy>=1.26.0 in /opt/anaconda3/lib/python3.1
2/site-packages (from pandas) (2.2.6)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/anaconda3/lib/
python3.12/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/anaconda3/lib/python3.1
2/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /opt/anaconda3/lib/python3.
12/site-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/lib/python3.12/sit
e-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

In [3]:
```
import numpy as np
import pandas as pd
```

In [4]:
```
movie_data = pd.read_csv("/Users/yaswanthkumarvejandla/Downloads/DATA SCIENC
```

In [5]:
```
movie_data
```

Out[5]:

| | 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 [6]:
```
movie_data.head(6)
```

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 |
| **5** | 2012 | Action | 39 | 63 | 200 | 2009 |

In [7]:
```
type(movie_data)
```
Out[7]:
```
pandas.core.frame.DataFrame
```
In [8]:
```
len(movie_data)
```
Out[8]:
```
559
```
In [9]:
```
movie_data.columns
```
Out[9]:
```
Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
       'Budget (million $)', 'Year of release'],
      dtype='object')
```
In [10]:
```
movie_data.shape
```
Out[10]:
```
(559, 6)
```
In [11]:
```
movie_data.describe()
```
Out[11]:

| | Rotten Tomatoes Ratings % | Audience Ratings % | Budget (million $) | Year of release |
|---|---|---|---|---|
| **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 [12]:
```
movie_data.columns = ['Film','Genre','CriticRating','AudienceRating','Budget
```

In [13]:
```
movie_data.head()
```

Out[13]:

| | Film | Genre | CriticRating | AudienceRating | BudgetMillions | Year |
|---|---|---|---|---|---|---|
| **0** | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |
| **1** | 10,000 B.C. | Adventure | 9 | 44 | 105 | 2008 |
| **2** | 12 Rounds | Action | 30 | 52 | 20 | 2009 |
| **3** | 127 Hours | Adventure | 93 | 84 | 18 | 2010 |
| **4** | 17 Again | Comedy | 55 | 70 | 20 | 2009 |

In [14]:
```
movie_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Film            559 non-null    object
 1   Genre           559 non-null    object
 2   CriticRating    559 non-null    int64 
 3   AudienceRating  559 non-null    int64 
 4   BudgetMillions  559 non-null    int64 
 5   Year            559 non-null    int64 
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

In [15]:
```
movie_data["Film"] = movie_data["Film"].astype("category")
```

In [16]:
```
movie_data.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   AudienceRating  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 [23]:
```
movie_data["Genre"] = movie_data["Genre"].astype("category")
```

In [24]:
```
movie_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype   
---  ------          --------------  -----   
 0   Film            559 non-null    category
 1   Genre           559 non-null    category
 2   CriticRating    559 non-null    int64   
 3   AudienceRating  559 non-null    int64   
 4   BudgetMillions  559 non-null    int64   
 5   Year            559 non-null    int64   
dtypes: category(2), int64(4)
memory usage: 40.1 KB
```

In [25]:
```
movie_data
```

Out[25]:

| | Film | Genre | CriticRating | AudienceRating | BudgetMillions | Year |
|---|---|---|---|---|---|---|
| **0** | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |
| **1** | 10,000 B.C. | Adventure | 9 | 44 | 105 | 2008 |
| **2** | 12 Rounds | Action | 30 | 52 | 20 | 2009 |
| **3** | 127 Hours | Adventure | 93 | 84 | 18 | 2010 |
| **4** | 17 Again | Comedy | 55 | 70 | 20 | 2009 |
| **...** | ... | ... | ... | ... | ... | ... |
| **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 [26]:
```
!pip install matplotlib
!pip install seaborn
```

```
Requirement already satisfied: matplotlib in /opt/anaconda3/lib/python3.12/s
ite-packages (3.10.3)
Requirement already satisfied: contourpy>=1.0.1 in /opt/anaconda3/lib/python
3.12/site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /opt/anaconda3/lib/python3.1
2/site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /opt/anaconda3/lib/pytho
n3.12/site-packages (from matplotlib) (4.58.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /opt/anaconda3/lib/pytho
n3.12/site-packages (from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in /opt/anaconda3/lib/python3.12/
site-packages (from matplotlib) (2.2.6)
Requirement already satisfied: packaging>=20.0 in /opt/anaconda3/lib/python
3.12/site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in /opt/anaconda3/lib/python3.12/si
te-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/anaconda3/lib/python
3.12/site-packages (from matplotlib) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /opt/anaconda3/lib/py
thon3.12/site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/lib/python3.12/sit
e-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: seaborn in /opt/anaconda3/lib/python3.12/site
-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /opt/anaconda3/lib/py
thon3.12/site-packages (from seaborn) (2.2.6)
Requirement already satisfied: pandas>=1.2 in /opt/anaconda3/lib/python3.12/
site-packages (from seaborn) (2.2.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /opt/anaconda3/li
b/python3.12/site-packages (from seaborn) (3.10.3)
Requirement already satisfied: contourpy>=1.0.1 in /opt/anaconda3/lib/python
3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /opt/anaconda3/lib/python3.1
2/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /opt/anaconda3/lib/pytho
n3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.58.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /opt/anaconda3/lib/pytho
n3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /opt/anaconda3/lib/python
3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.1)
Requirement already satisfied: pillow>=8 in /opt/anaconda3/lib/python3.12/si
te-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/anaconda3/lib/python
3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /opt/anaconda3/lib/py
thon3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/anaconda3/lib/python3.1
2/site-packages (from pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /opt/anaconda3/lib/python3.
12/site-packages (from pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/lib/python3.12/sit
e-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.
16.0)
```

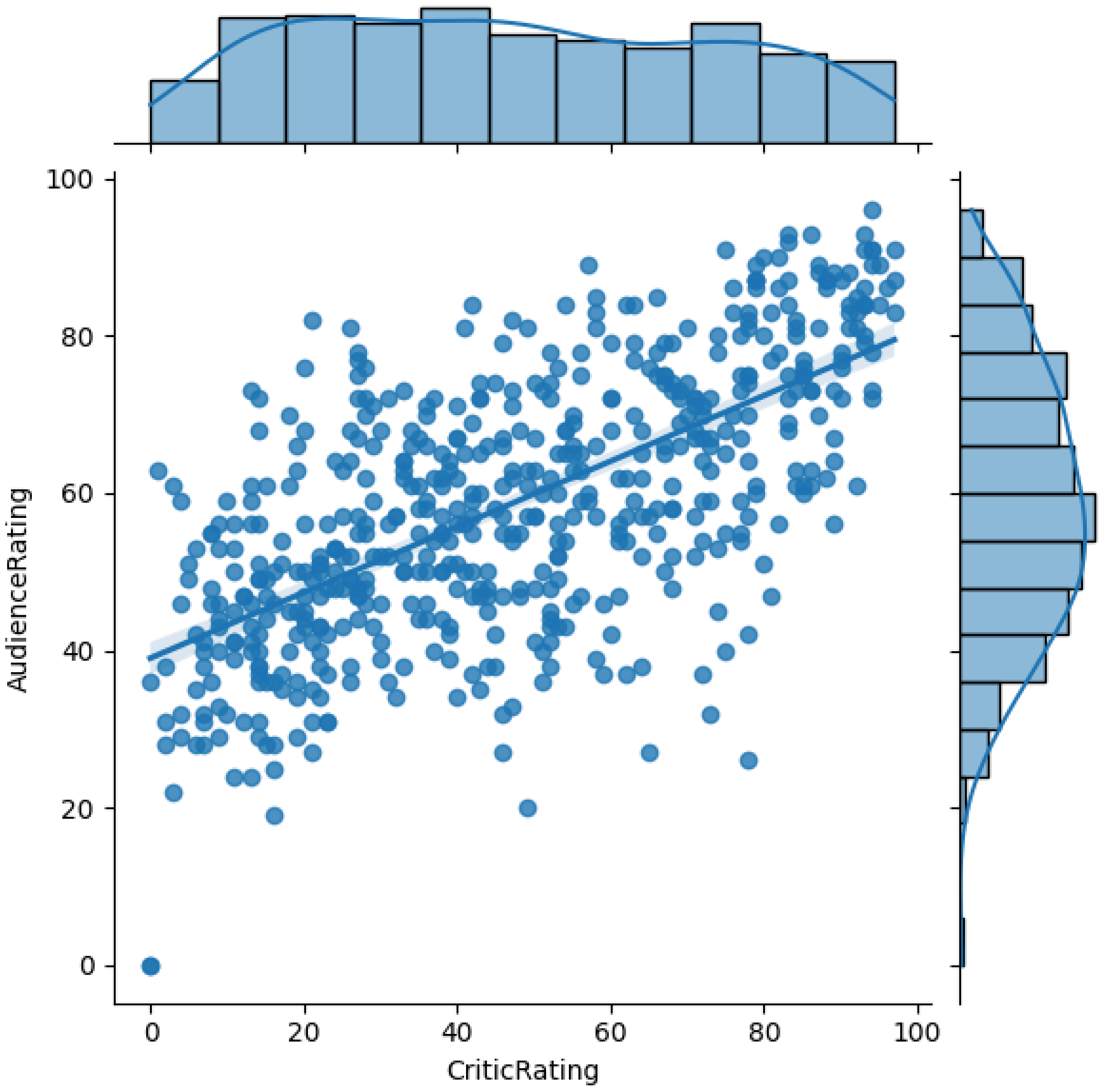
In [27]:
```python
import matplotlib.pyplot as plt
```

In [28]:
```
import seaborn as sns
```

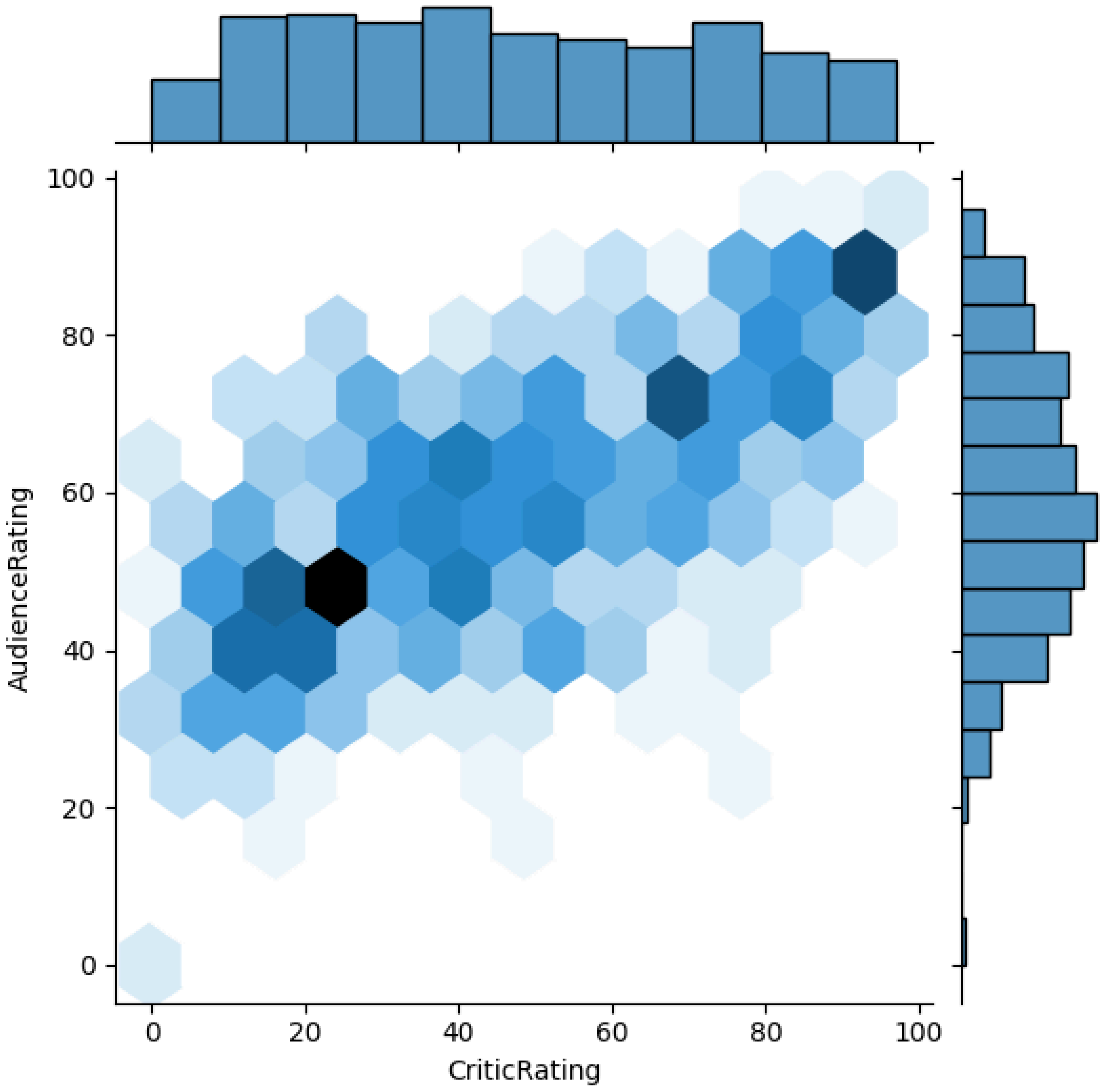
In [29]:
```
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

## working with JOINTS

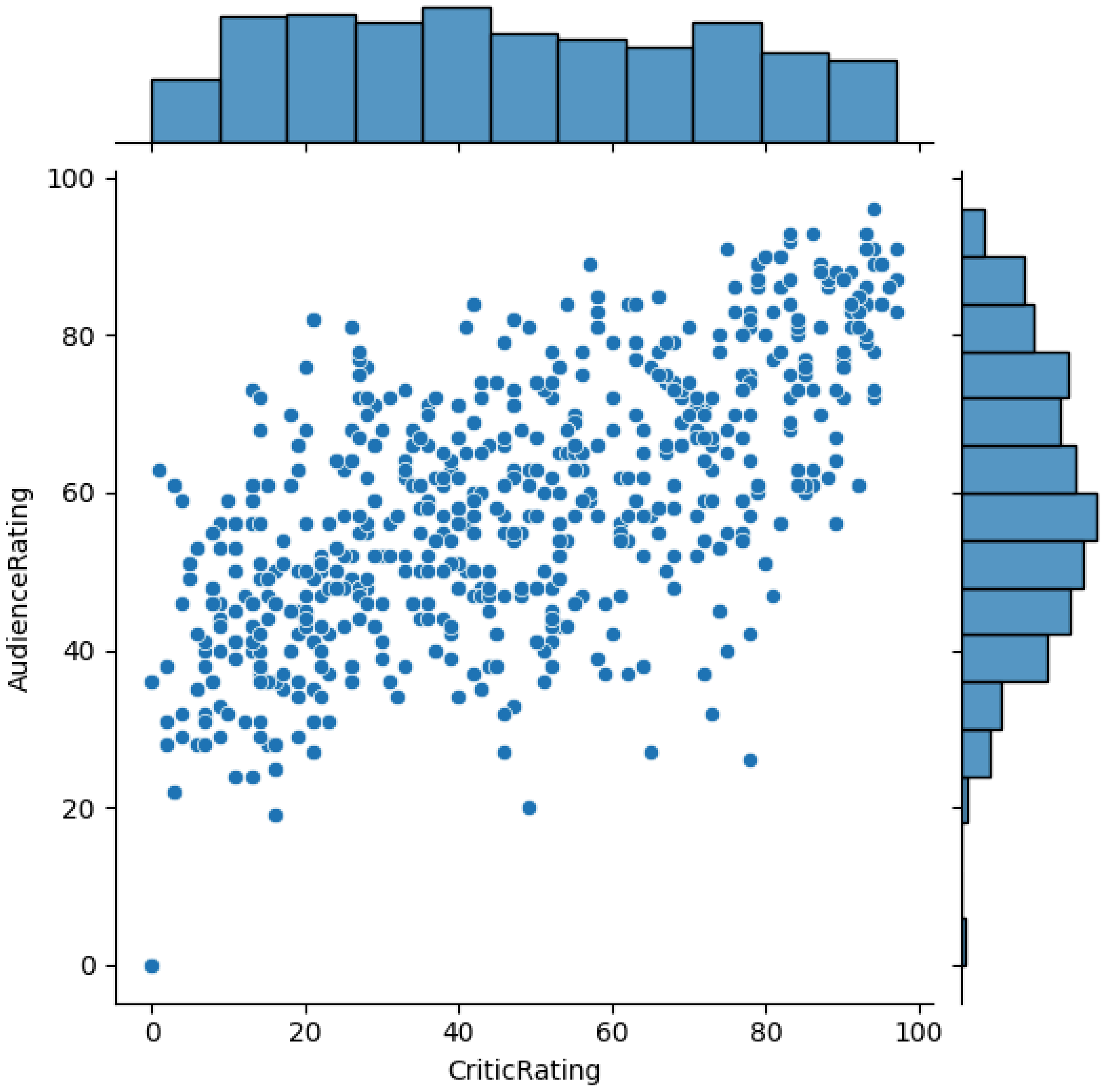
In [30]:
```
sns.jointplot(data = movie_data , x ='CriticRating' , y = 'AudienceRating',
plt.show()
```



In [31]:
```
sns.jointplot(data = movie_data , x ='CriticRating' , y = 'AudienceRating',
plt.show()
```

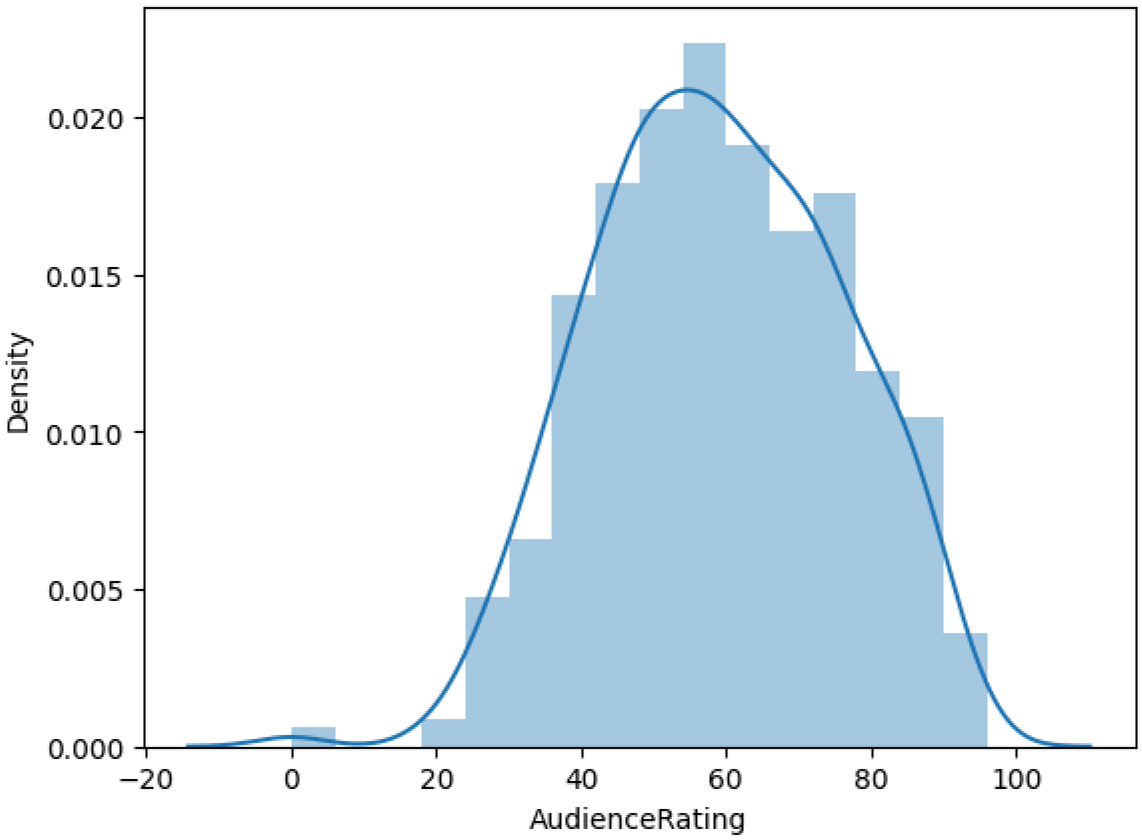


In [32]:
```
sns.jointplot(data = movie_data , x ='CriticRating' , y = 'AudienceRating',
plt.show()
```

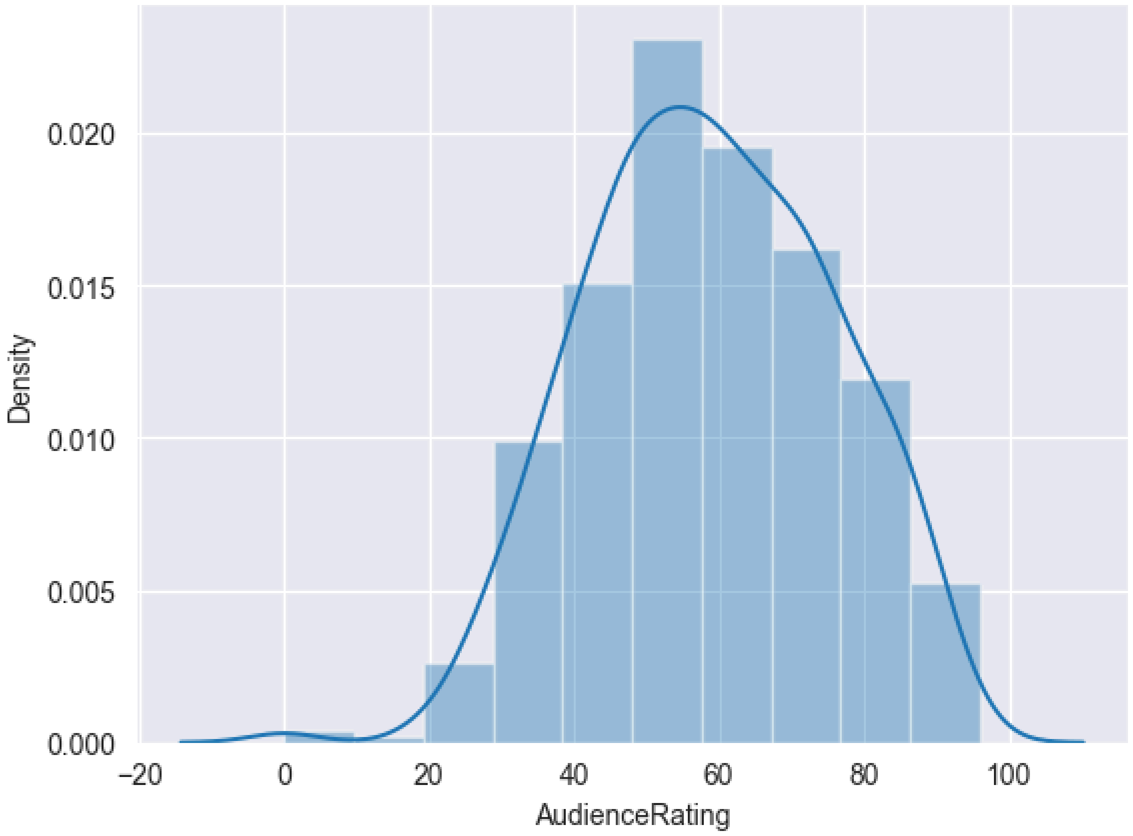


## Histograms

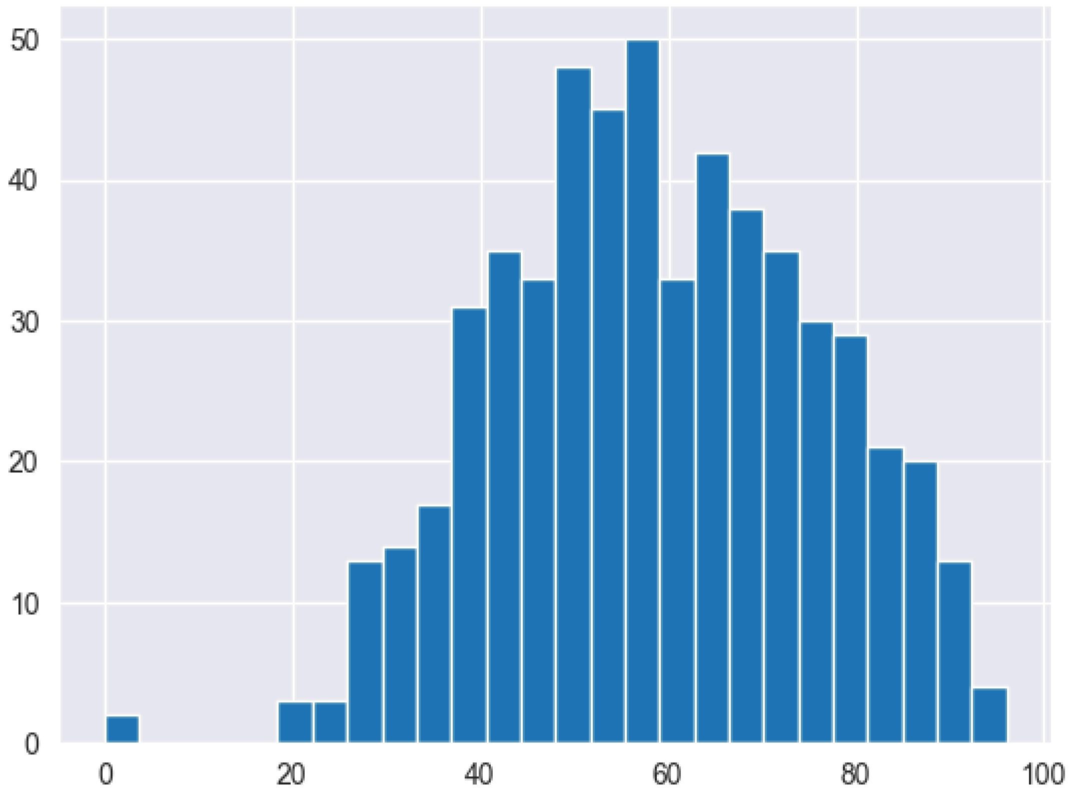
In [33]:
```
sns.distplot(movie_data['AudienceRating'])
plt.show()
```



In [34]:
```
sns.set_style('darkgrid')
sns.distplot(movie_data.AudienceRating, bins =10)
plt.show()
```



```
In [35]: plt.hist(movie_data.AudienceRating , bins=26)
         plt.show()
```



```
In [36]: movie_data.head(26)
```

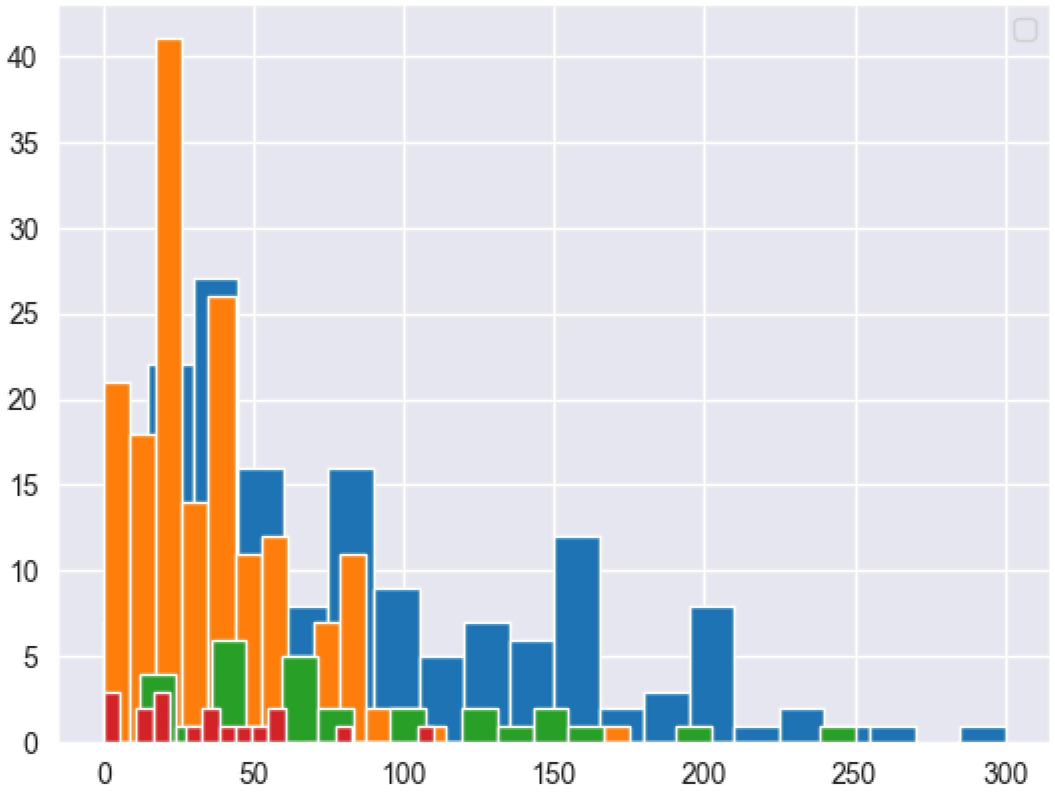
Out[36]:

| | Film | Genre | CriticRating | AudienceRating | BudgetMillions | Year |
|---|---|---|---|---|---|---|
| 0 | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |
| 1 | 10,000 B.C. | Adventure | 9 | 44 | 105 | 2008 |
| 2 | 12 Rounds | Action | 30 | 52 | 20 | 2009 |
| 3 | 127 Hours | Adventure | 93 | 84 | 18 | 2010 |
| 4 | 17 Again | Comedy | 55 | 70 | 20 | 2009 |
| 5 | 2012 | Action | 39 | 63 | 200 | 2009 |
| 6 | 27 Dresses | Comedy | 40 | 71 | 30 | 2008 |
| 7 | 30 Days of Night | Horror | 50 | 57 | 32 | 2007 |
| 8 | 30 Minutes or Less | Comedy | 43 | 48 | 28 | 2011 |
| 9 | 50/50 | Comedy | 93 | 93 | 8 | 2011 |
| 10 | 88 Minutes | Drama | 5 | 51 | 30 | 2007 |
| 11 | A Dangerous Method | Drama | 79 | 89 | 20 | 2011 |
| 12 | A Nightmare on Elm Street | Horror | 13 | 40 | 35 | 2010 |
| 13 | A Serious Man | Drama | 89 | 64 | 7 | 2009 |
| 14 | A Very Harold and Kumar Christmas | Comedy | 72 | 71 | 19 | 2011 |
| 15 | Abduction | Action | 4 | 46 | 35 | 2011 |
| 16 | Across the Universe | Romance | 54 | 84 | 45 | 2007 |
| 17 | Adventureland | Comedy | 89 | 56 | 10 | 2009 |
| 18 | Albert Nobbs | Drama | 53 | 43 | 8 | 2011 |
| 19 | Alice in Wonderland | Adventure | 52 | 72 | 200 | 2010 |
| 20 | Alien vs. Predator -- Requiem | Horror | 14 | 37 | 40 | 2007 |
| 21 | Aliens in the Attic | Adventure | 30 | 46 | 45 | 2009 |
| 22 | All About Steve | Comedy | 6 | 35 | 15 | 2009 |
| 23 | All Good Things | Drama | 33 | 64 | 20 | 2010 |
| 24 | Amelia | Adventure | 21 | 35 | 40 | 2009 |

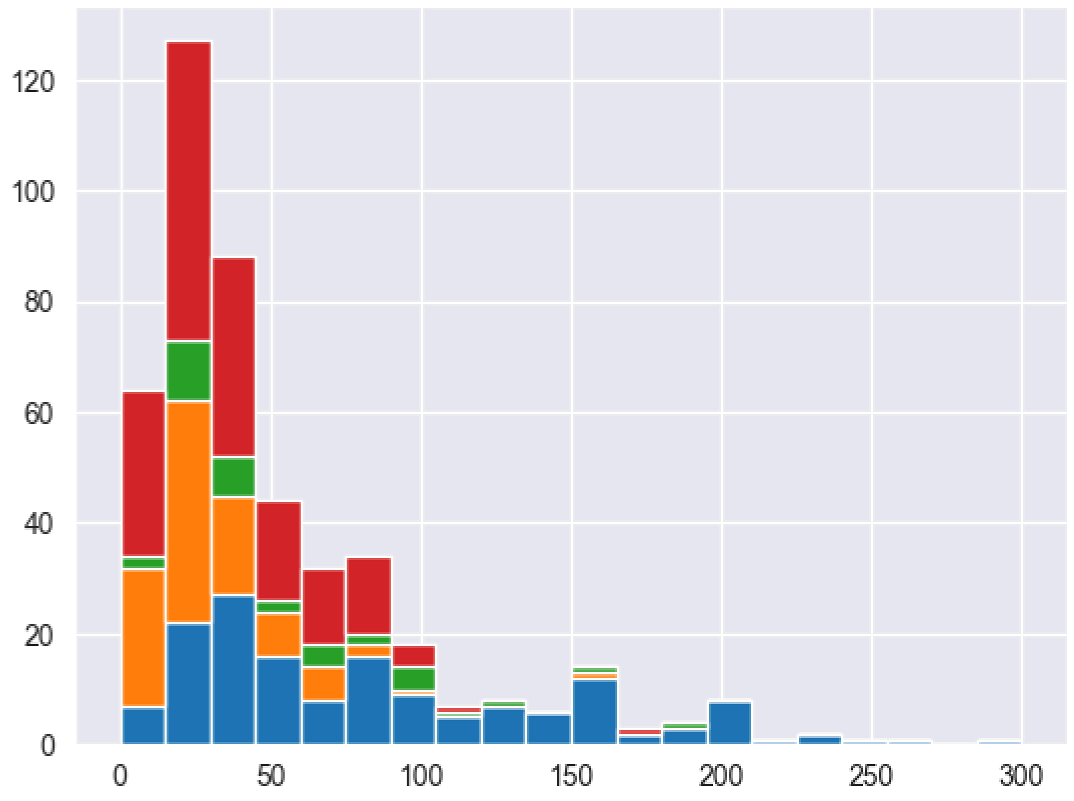
| | Film | Genre | CriticRating | AudienceRating | BudgetMillions | Year |
|---|---|---|---|---|---|---|
| **25** | American Gangster | Thriller | 79 | 87 | 100 | 2007 |

## working with filter

In [37]:
```
plt.hist(movie_data[movie_data.Genre=='Action'].BudgetMillions, bins=20)
plt.hist(movie_data[movie_data.Genre=='Comedy'].BudgetMillions, bins=20)
plt.hist(movie_data[movie_data.Genre=='Adventure'].BudgetMillions, bins=20)
plt.hist(movie_data[movie_data.Genre=='Romance'].BudgetMillions, bins=20)
plt.legend()
plt.show()
```



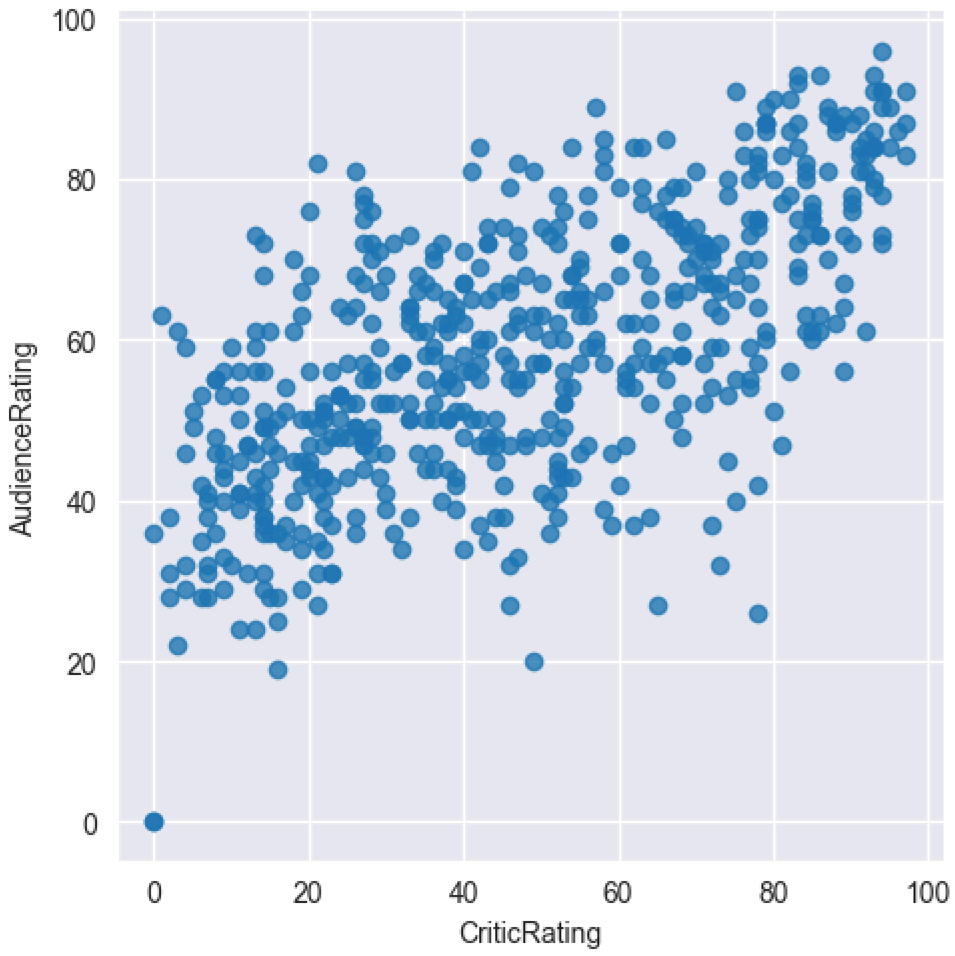
In [38]:
```
plt.hist([movie_data[movie_data.Genre == 'Action'].BudgetMillions,\
          movie_data[movie_data.Genre == 'Drama'].BudgetMillions, \
          movie_data[movie_data.Genre == 'Thriller'].BudgetMillions, \
          movie_data[movie_data.Genre == 'Comedy'].BudgetMillions],
         bins = 20, stacked = True)
plt.show()
```



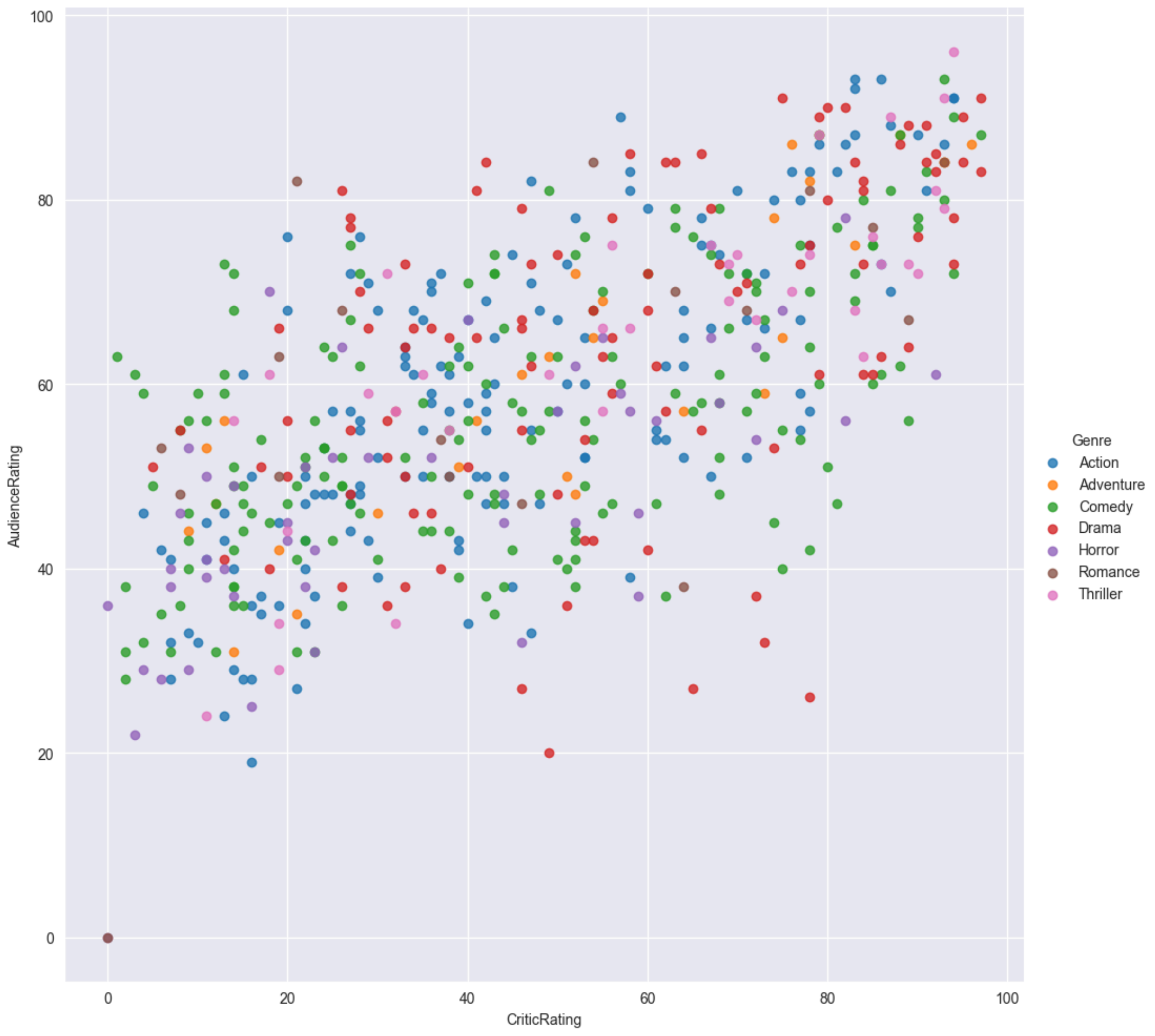
```
In [39]: for gen in movie_data.Genre.cat.categories:
             print(gen)
```

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

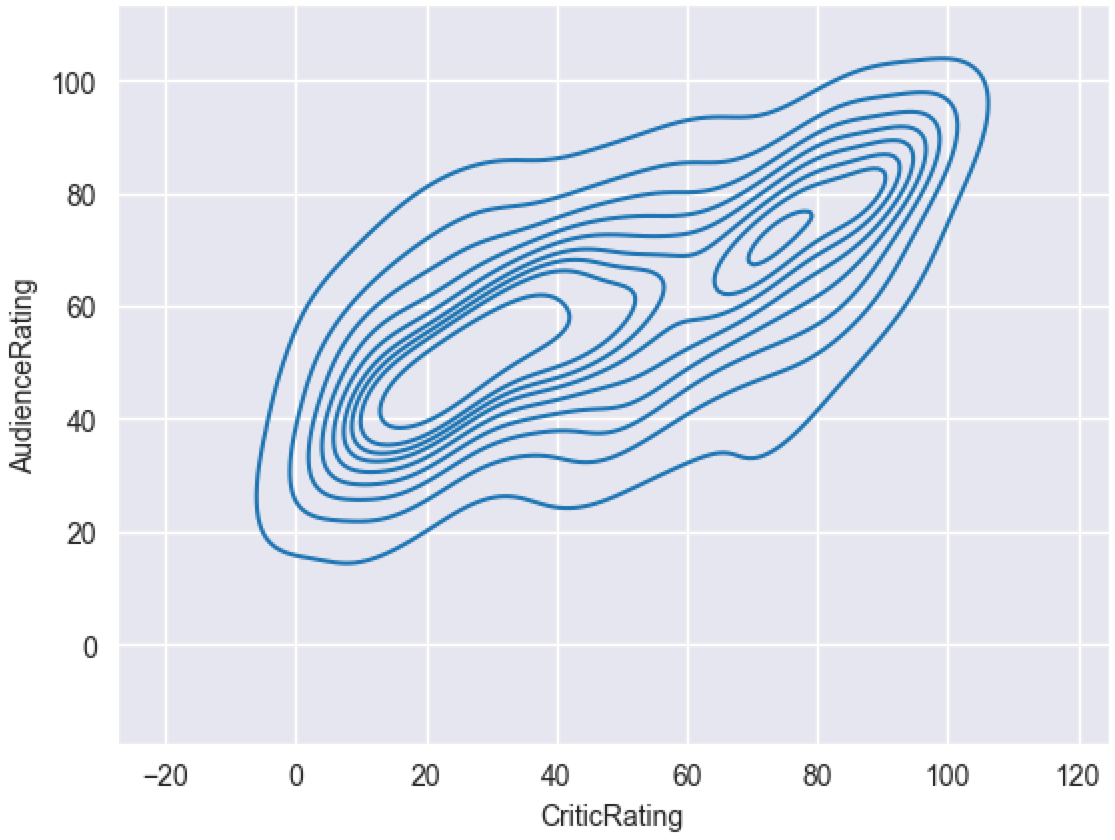
```
In [40]: sns.lmplot(data=movie_data , x='CriticRating',y='AudienceRating', fit_reg=Fa
         plt.show()
```



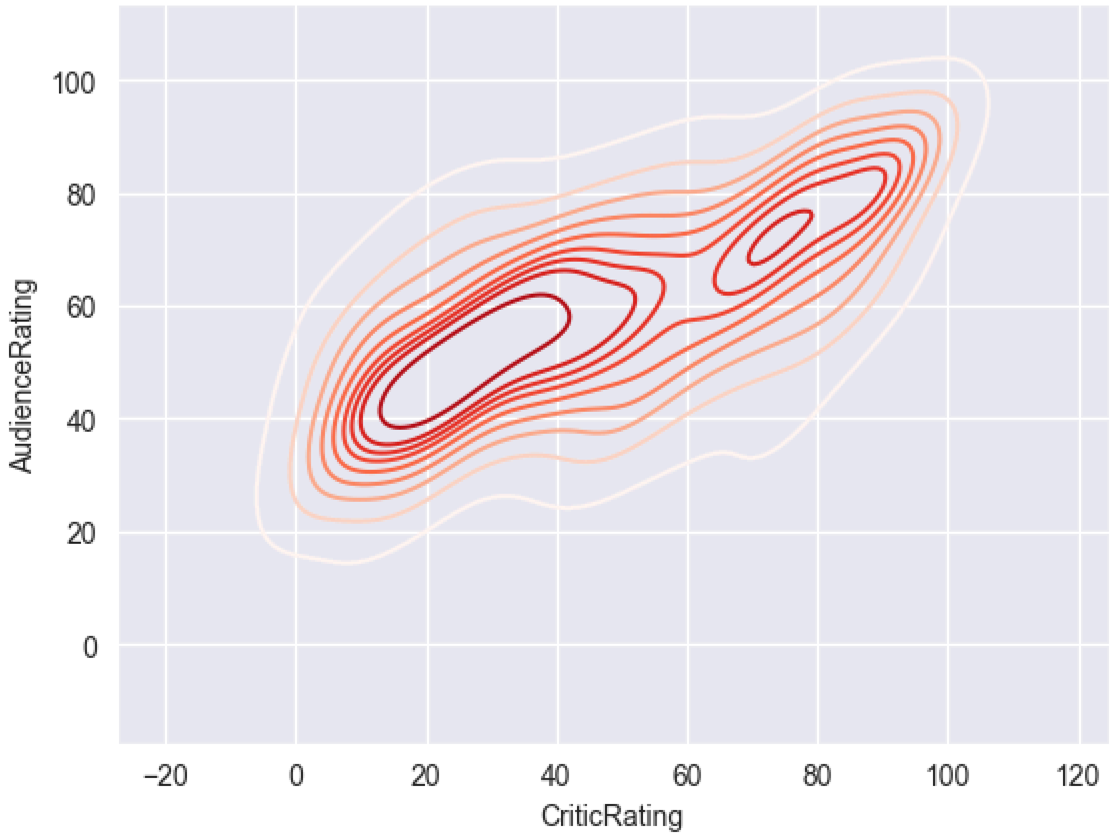
```
In [41]: sns.lmplot(data=movie_data , x='CriticRating',y='AudienceRating', fit_reg=Fa
         plt.show()
```



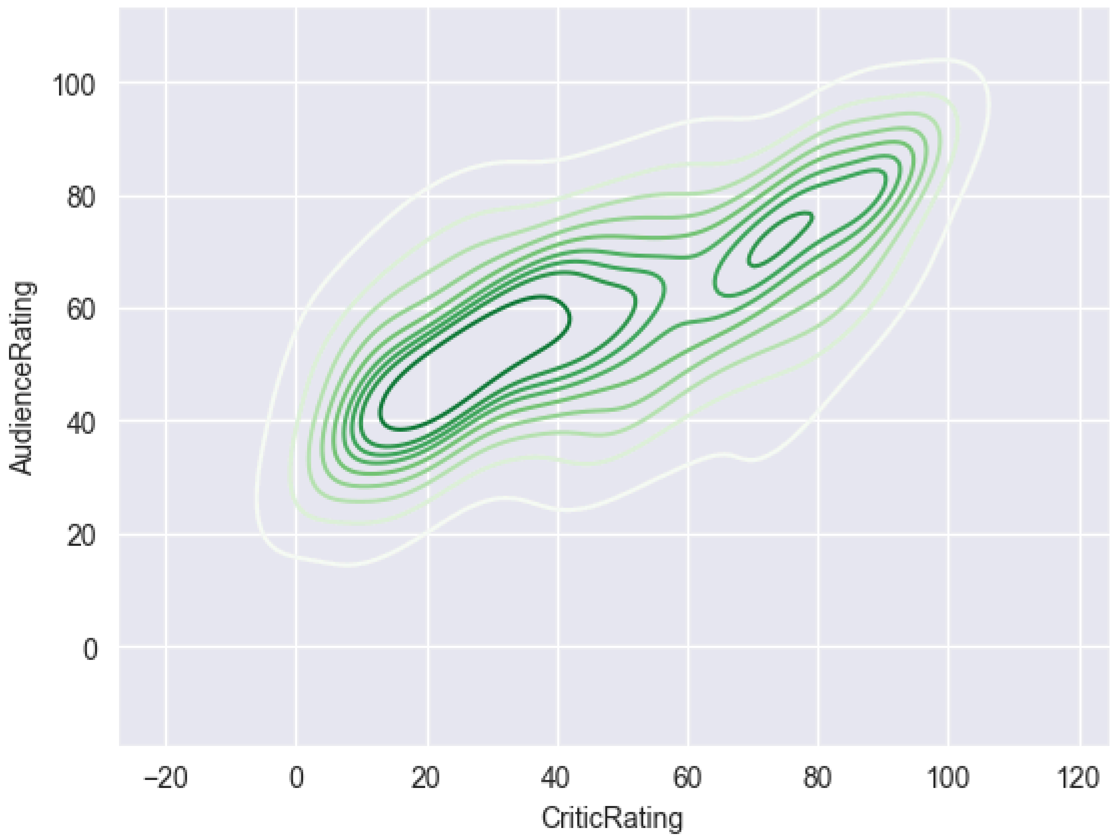
In [42]:
```
sns.kdeplot(x= movie_data['CriticRating'],y=movie_data['AudienceRating'])
plt.show()
```



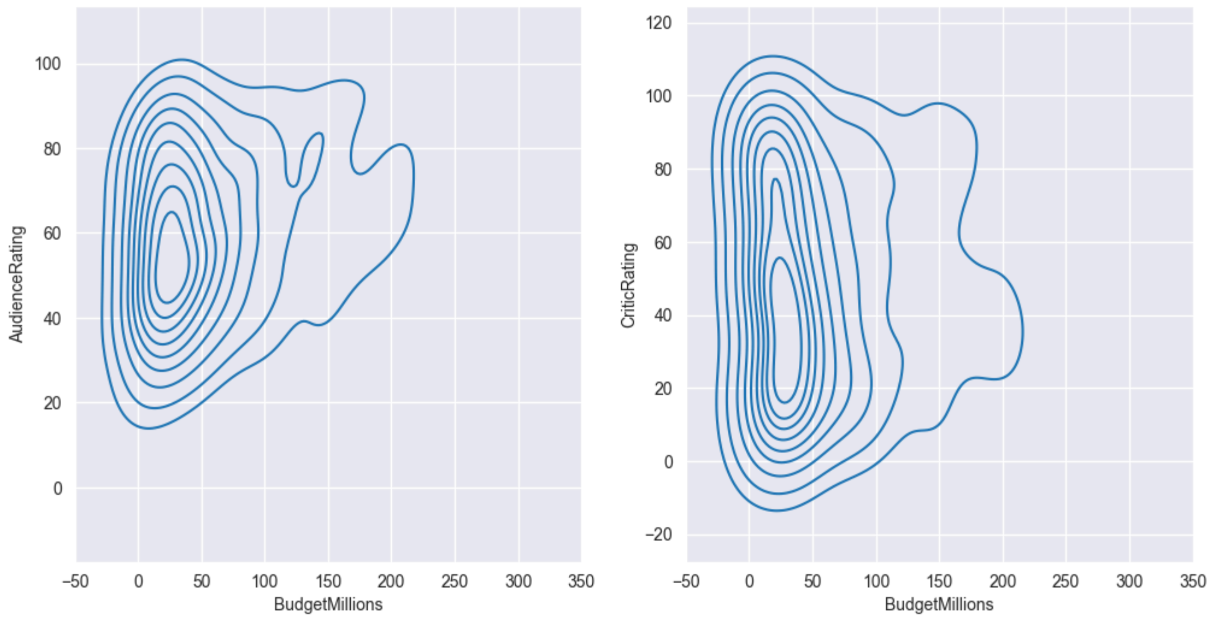
In [43]:
```
sns.kdeplot(x=movie_data['CriticRating'],y=movie_data['AudienceRating'],  sh
plt.show()
```



In [44]:
```
sns.kdeplot(x=movie_data['CriticRating'],y=movie_data['AudienceRating'],  sh
plt.show()
```

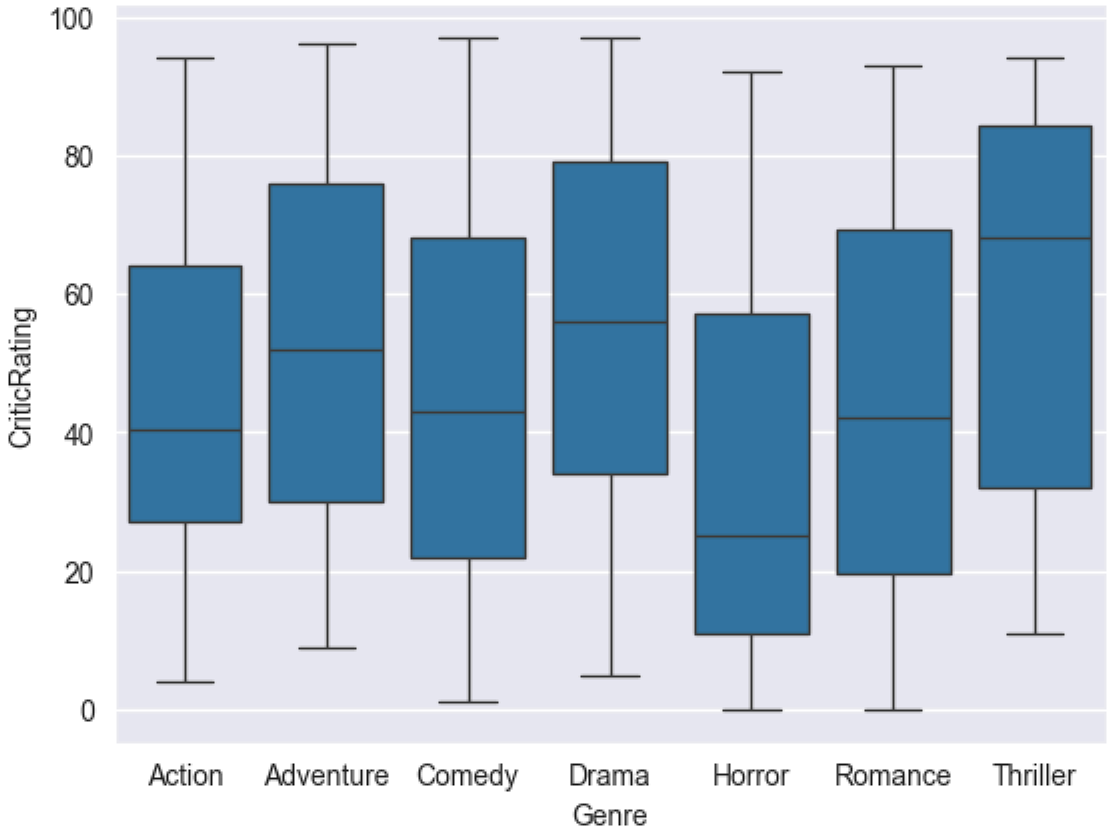


In [45]:
```
f, axes = plt.subplots(1, 2, figsize=(12, 6))

sns.kdeplot(x=movie_data['BudgetMillions'], y=movie_data['AudienceRating'],
sns.kdeplot(x=movie_data['BudgetMillions'], y=movie_data['CriticRating'], ax

plt.show()
```

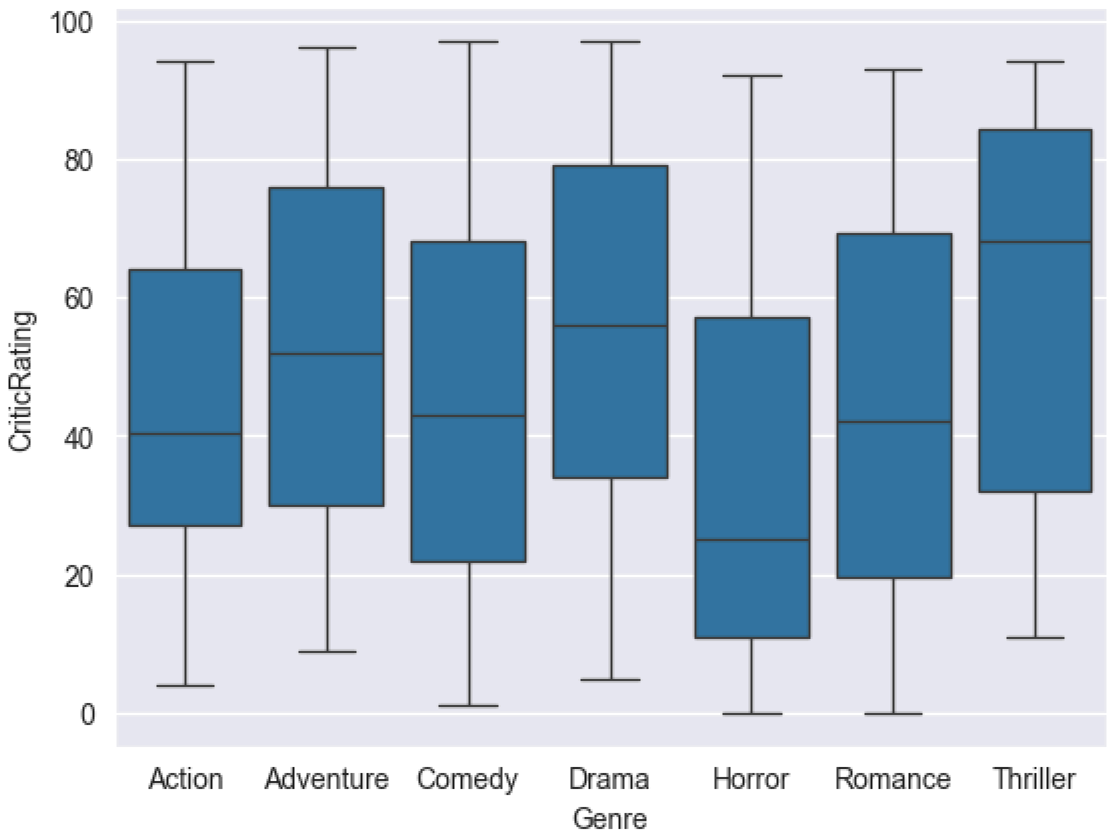


## Box plot

In [46]:
```
sns.boxplot(x=movie_data['Genre'], y=movie_data['CriticRating'])
plt.show()
```

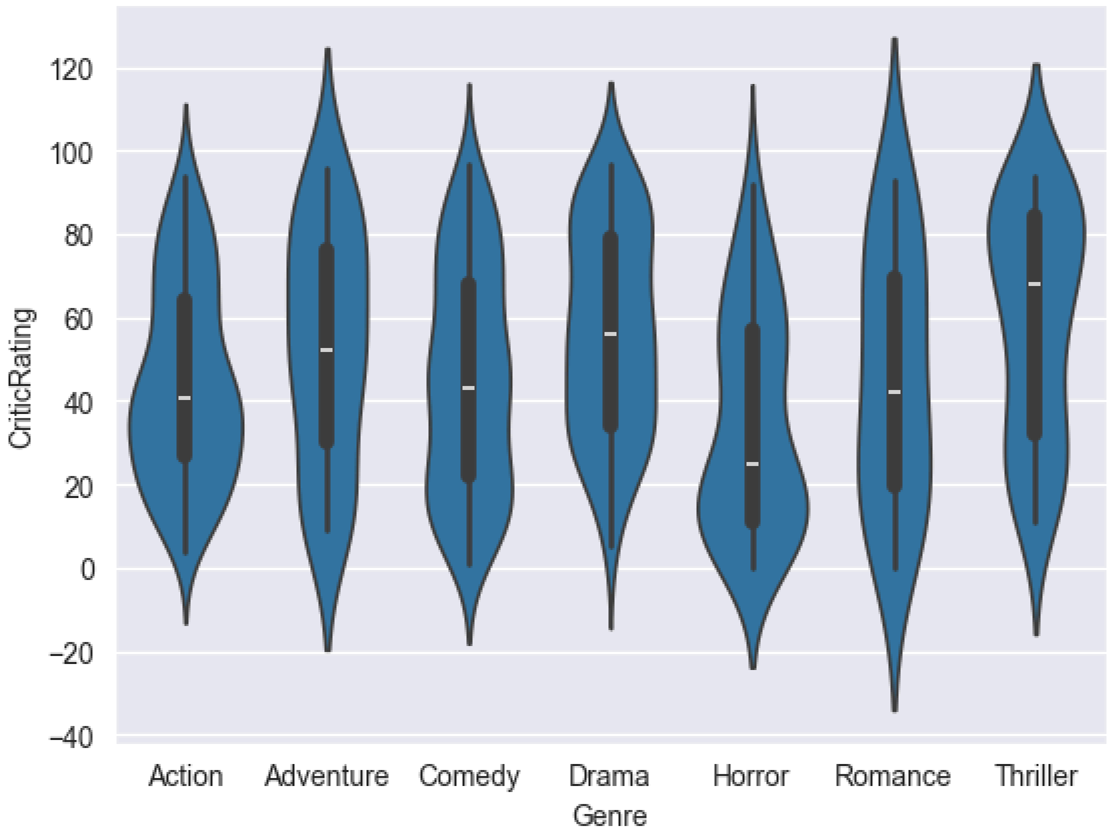


In [47]:
```
sns.boxplot(data=movie_data , x='Genre' , y ="CriticRating")
plt.show()
```



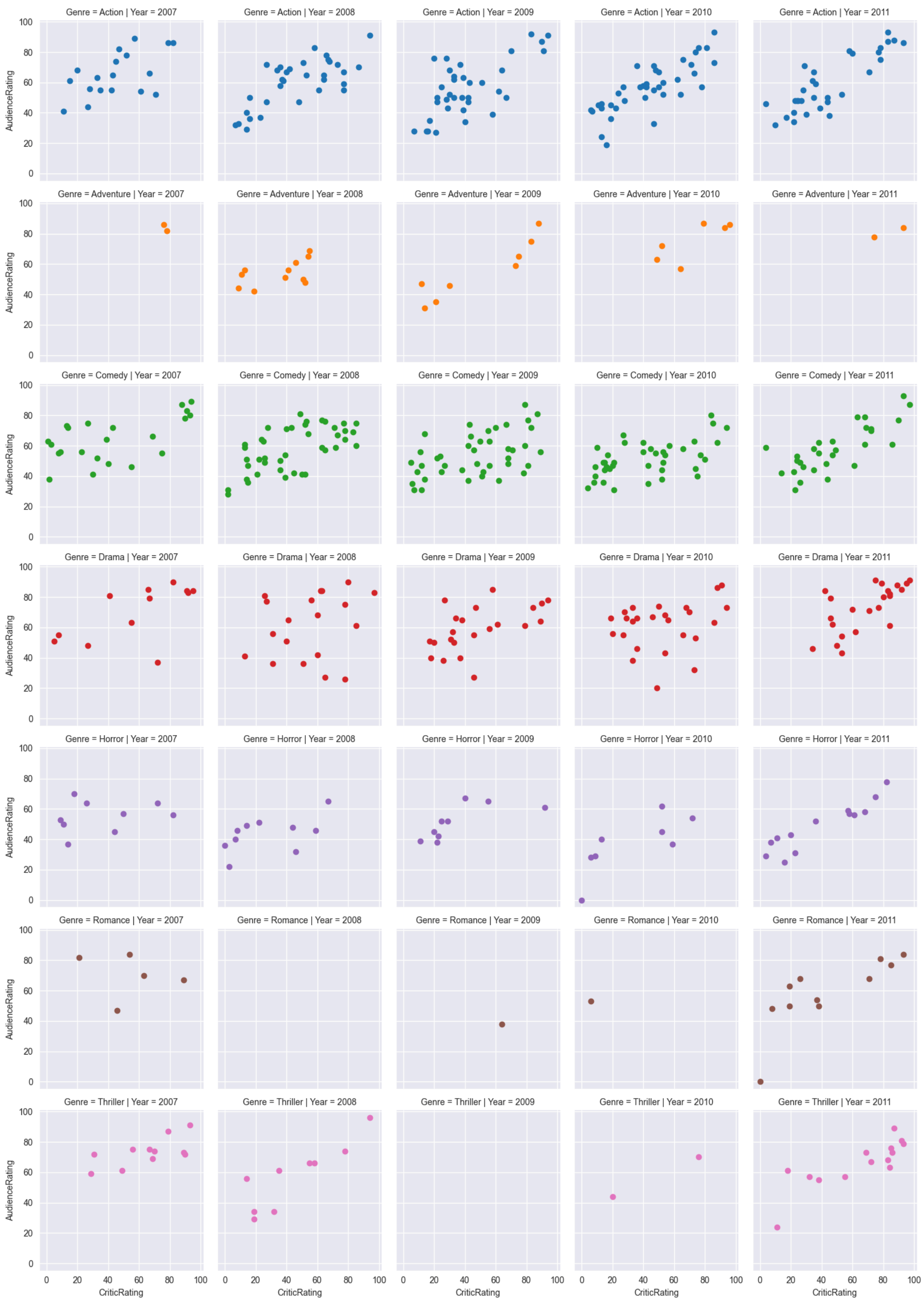
## Violin plot

In [48]:
```
sns.violinplot(data=movie_data , x='Genre' , y ="CriticRating")
plt.show()
```

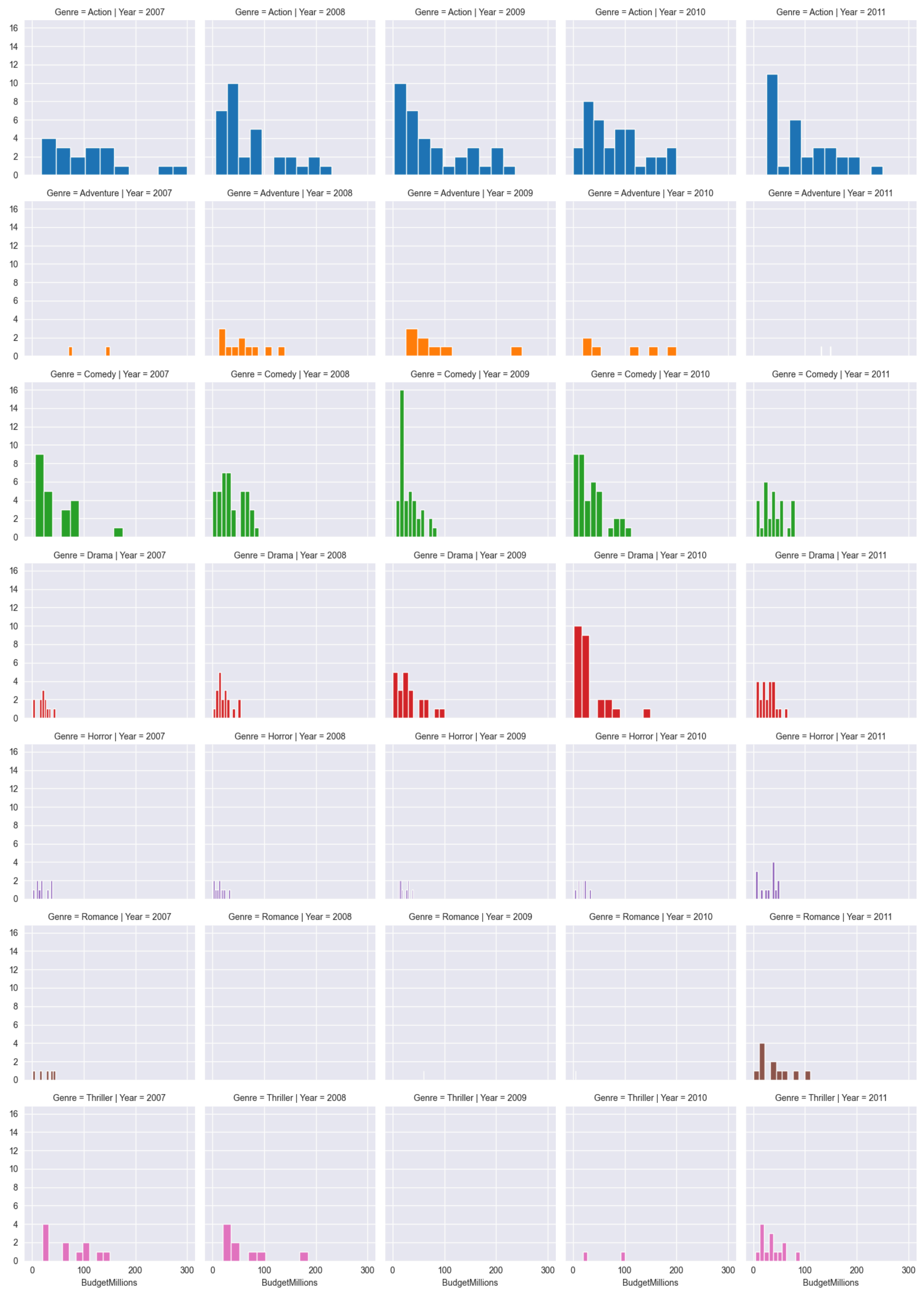


## CREATING FACET GRID

```
In [51]: k=sns.FacetGrid(movie_data,row = 'Genre',col='Year',hue='Genre')
         k=k.map(plt.scatter,'CriticRating','AudienceRating')
         plt.show()
```



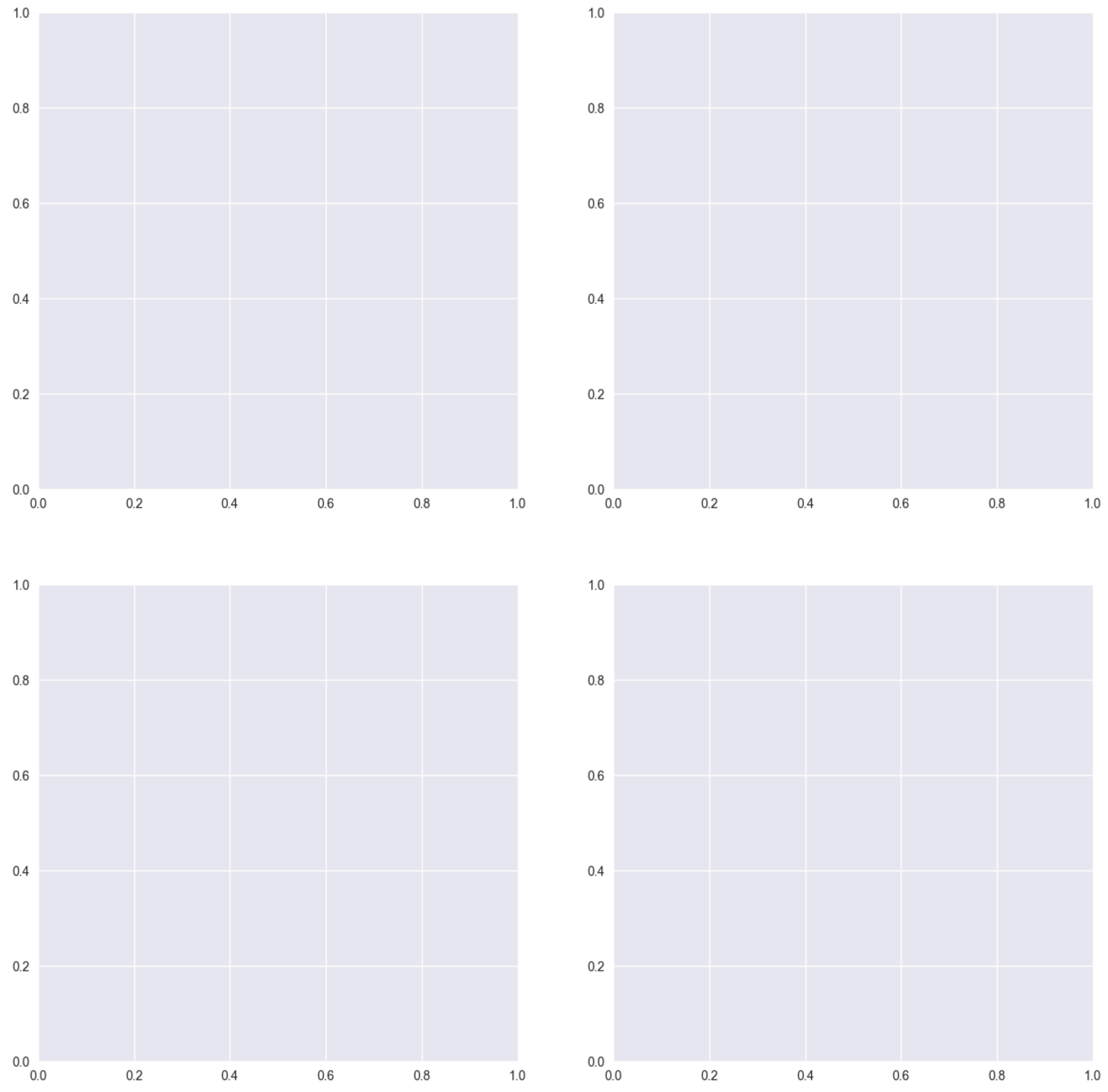
```
In [56]: g =sns.FacetGrid (movie_data, row = 'Genre', col = 'Year', hue = 'Genre')
         g = g.map(plt.hist, 'BudgetMillions') #scatterplots are mapped in facetgrid
         plt.show()
```

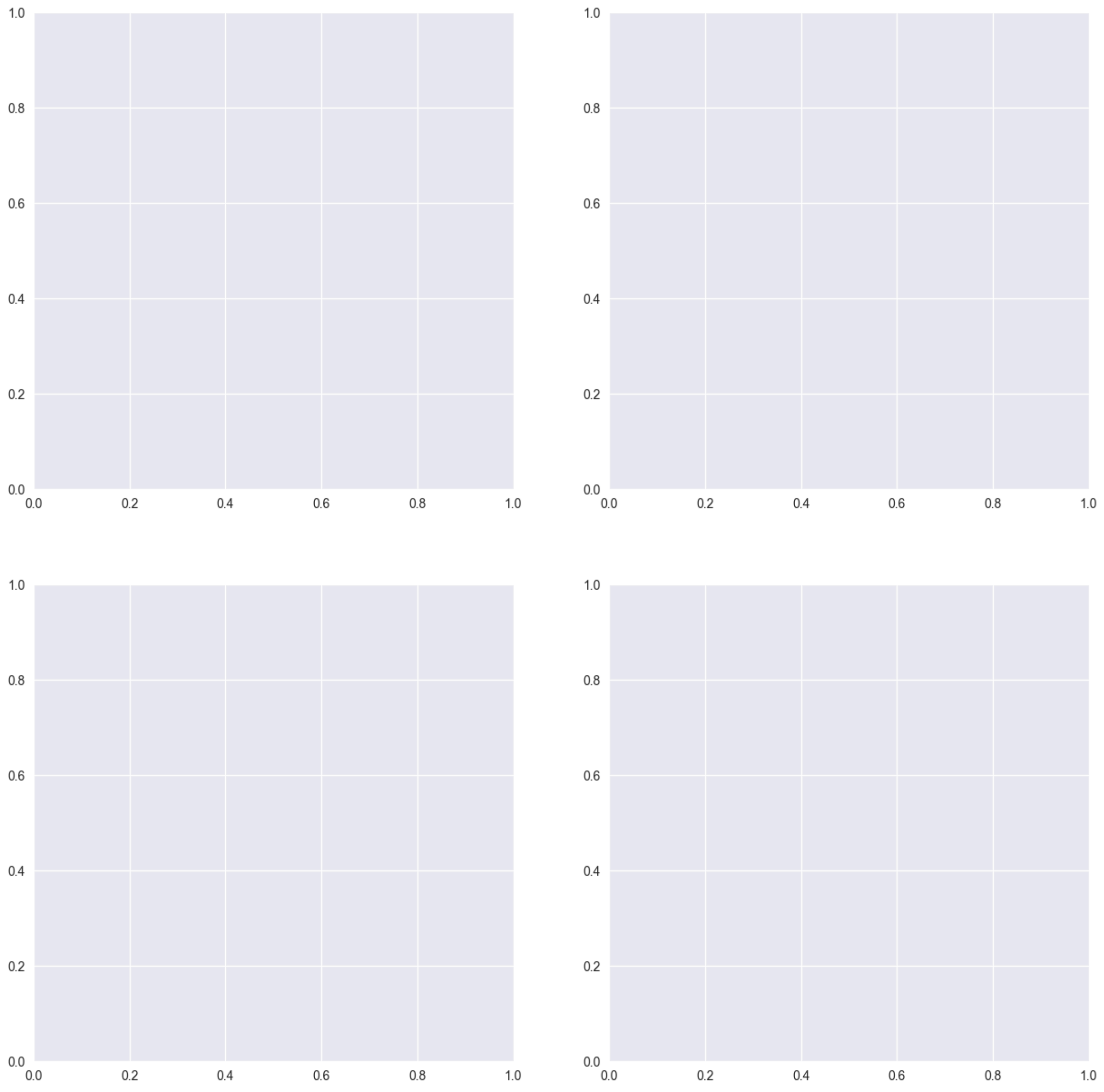


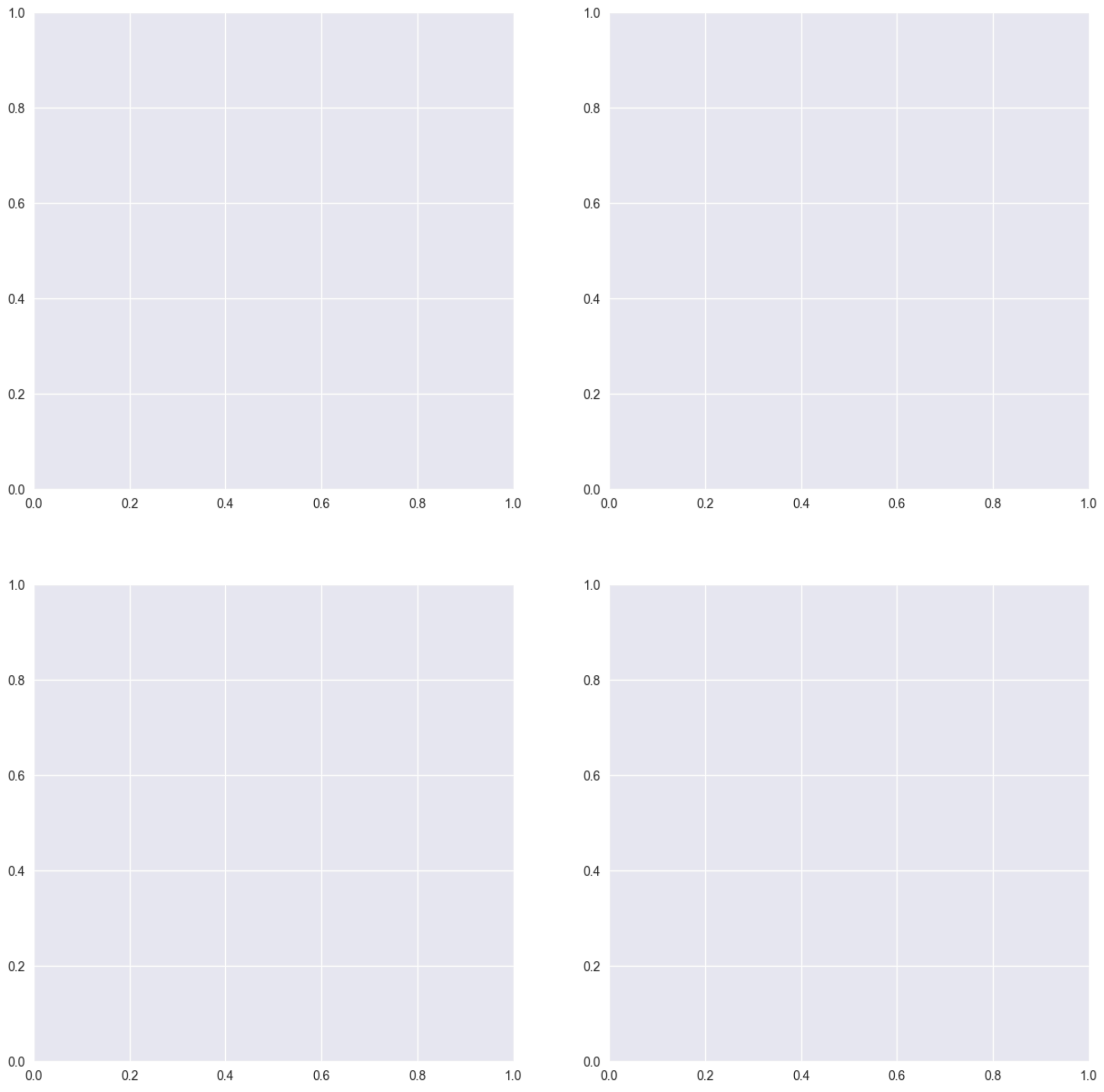
In [60]:
```
sns.set_style('darkgrid')
f, axes = plt.subplots (2,2, figsize = (15,15))

k1 = sns.kdeplot(movie_data['BudgetMillions'],movie_data['AudienceRating'],a
k2 = sns.kdeplot(movie_data['BudgetMillions'],movie_data['AudienceRating'],a

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

z = sns.violinplot(data=movie_data[movie_data.Genre=='Drama'], x='Year', y =

k4 = sns.kdeplot(movie_data.CriticRating,movie_data.AudienceRating,shade = T

k4b = sns.kdeplot(movie_data.CriticRating, movie_data.AudienceRating,cmap='R

plt.show()
```

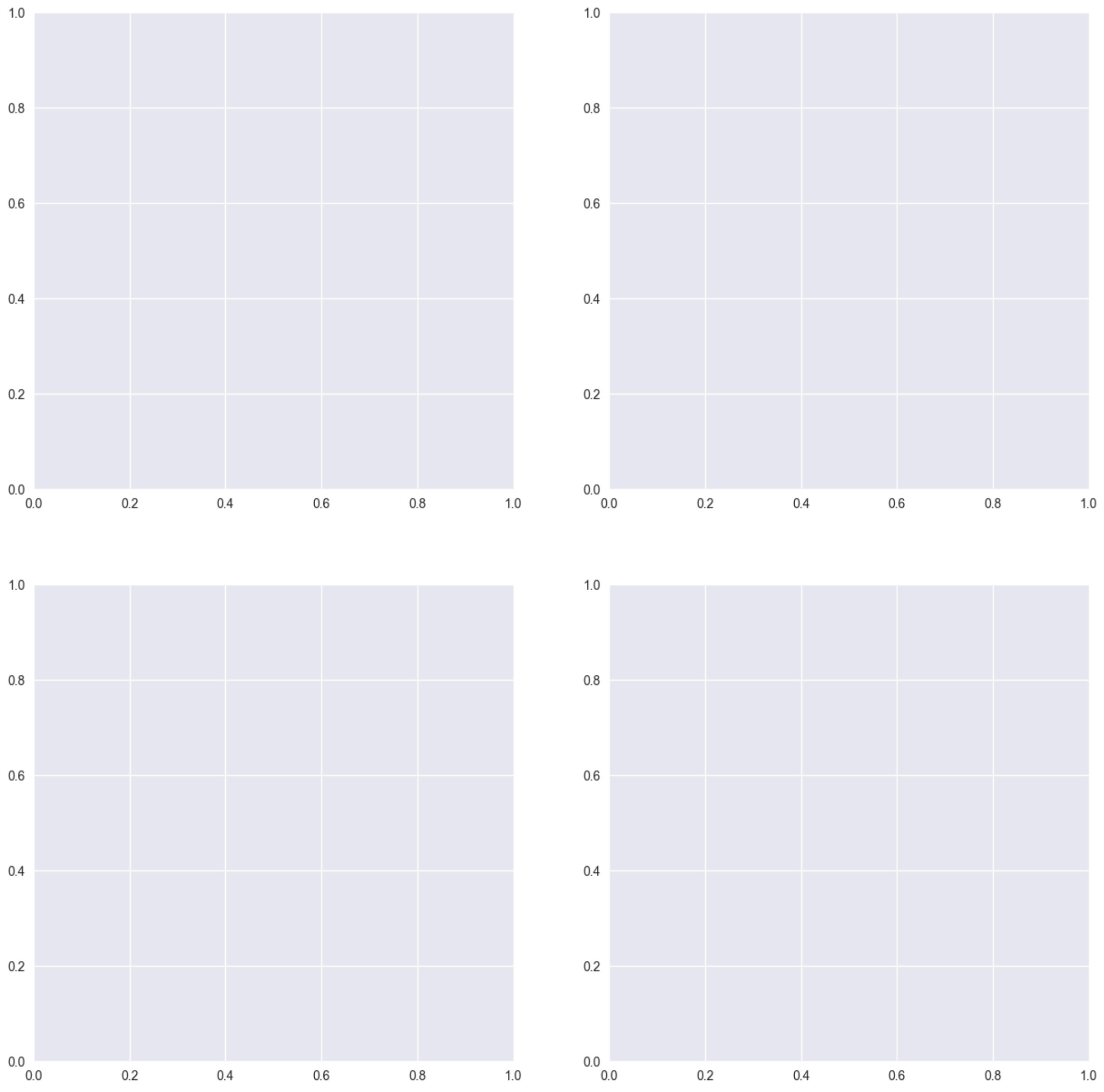
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[60], line 4
      1 sns.set_style('darkgrid')
      2 f, axes = plt.subplots (2,2, figsize = (15,15))
----> 4 k1 = sns.kdeplot(movie_data['BudgetMillions'],movie_data['AudienceRa
ting'],ax=axes[0,0])
      5 k2 = sns.kdeplot(movie_data['BudgetMillions'],movie_data['AudienceRa
ting'],ax = axes[0,1])
      7 k1.set(xlim=(-20,160))

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

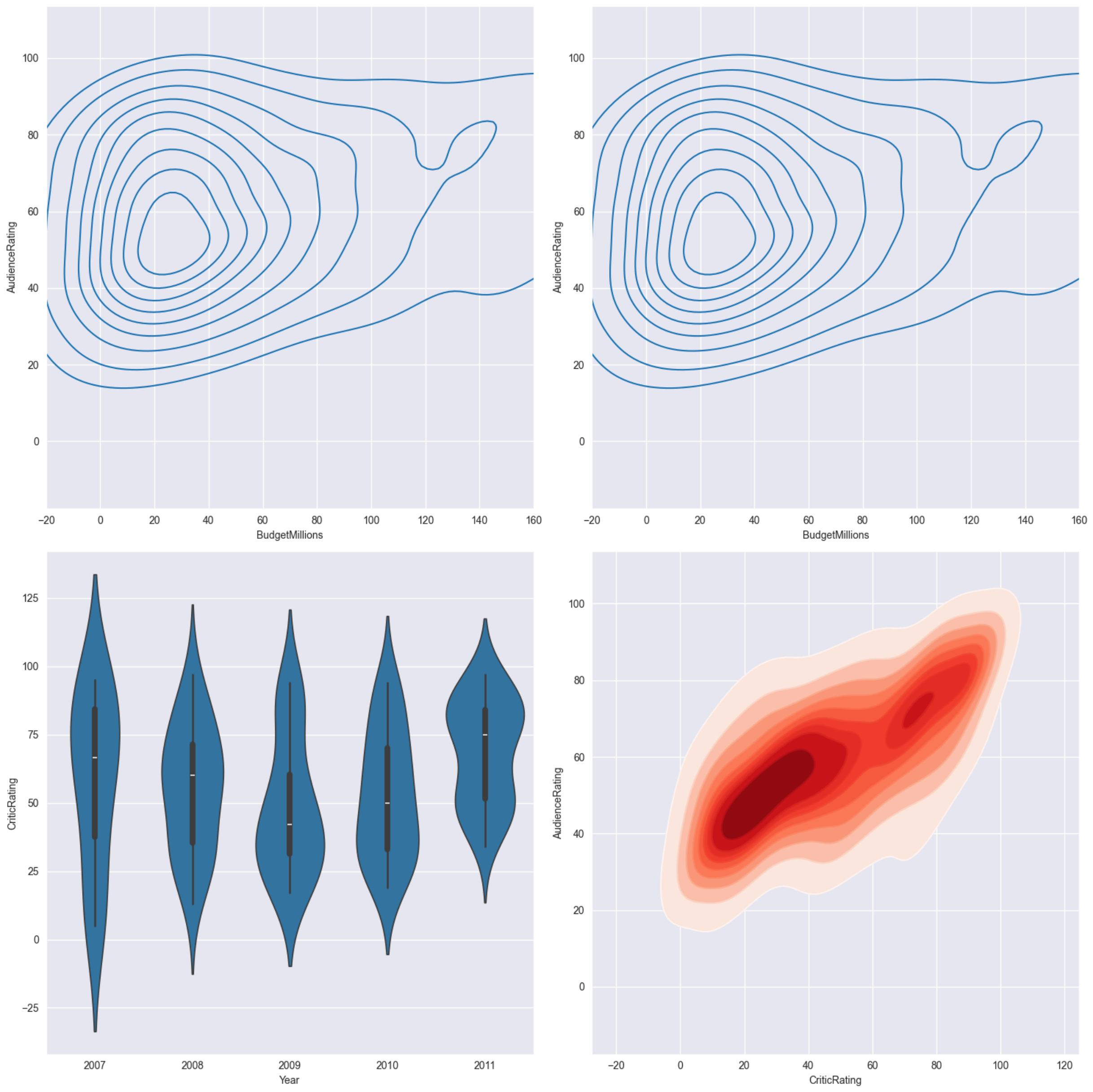
In [61]:
```
sns.set_style('darkgrid')
f, axes = plt.subplots(2, 2, figsize=(15, 15))

# KDE Plots with keyword arguments
k1 = sns.kdeplot(x=movie_data['BudgetMillions'], y=movie_data['AudienceRatir
k2 = sns.kdeplot(x=movie_data['BudgetMillions'], y=movie_data['AudienceRatir

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

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

# KDE Plot with shading and updated keyword arguments
k4 = sns.kdeplot(x=movie_data['CriticRating'], y=movie_data['AudienceRating'
                 fill=True, cmap='Reds', ax=axes[1, 1])
# Overlay another KDE line plot
k4b = sns.kdeplot(x=movie_data['CriticRating'], y=movie_data['AudienceRating
                  cmap='Reds', ax=axes[1, 1])

plt.tight_layout()
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.2 0.4 0.6 0.8 1.0


0.0
0.2
0.4
0.6
0.8
1.0

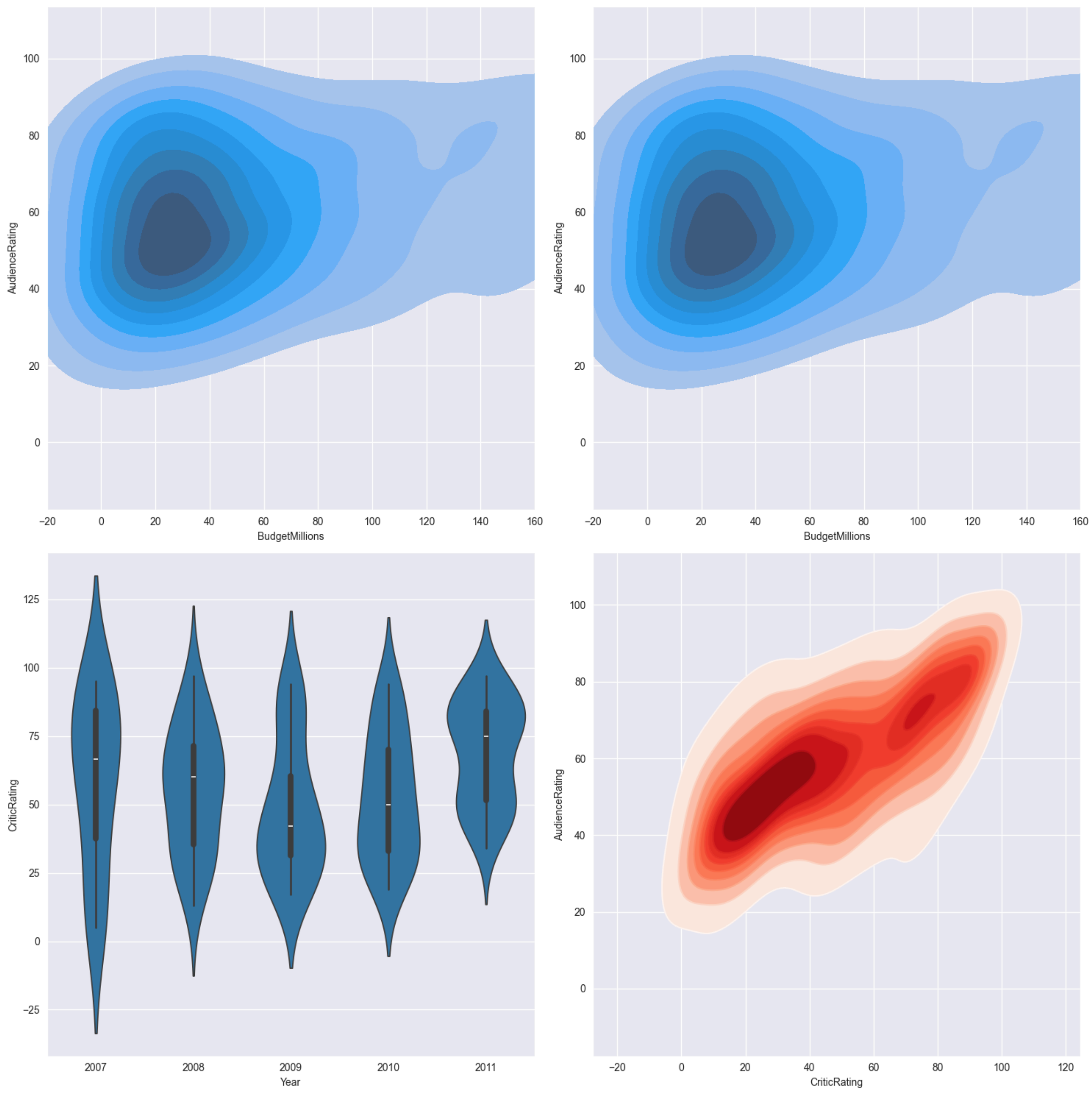

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0



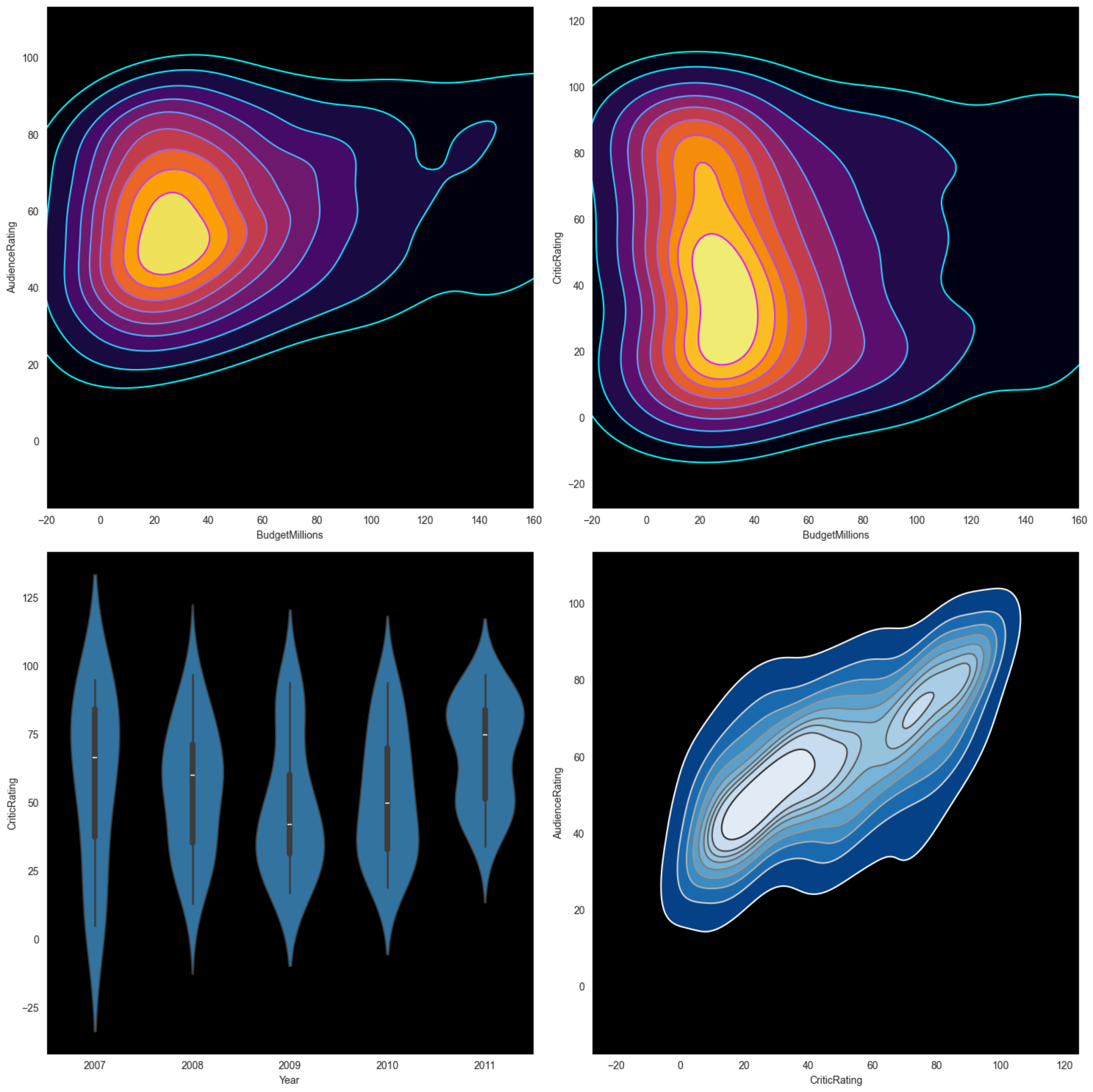


```
In [62]: import seaborn as sns
         import matplotlib.pyplot as plt

         # Drop rows with missing values in relevant columns
         movie_data_clean = movie_data.dropna(subset=['BudgetMillions', 'AudienceRati

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

         # First KDE plot
         k1 = sns.kdeplot(
             data=movie_data_clean, x='BudgetMillions', y='AudienceRating', ax=axes[0
         )
         k1.set(xlim=(-20, 160))

         # Second KDE plot (duplicate of first, maybe change?)
         k2 = sns.kdeplot(
             data=movie_data_clean, x='BudgetMillions', y='AudienceRating', ax=axes[0
         )
```

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

# Violin plot (ensure 'Drama' exists)
drama_data = movie_data_clean[movie_data_clean.Genre == 'Drama']
if not drama_data.empty:
    sns.violinplot(data=drama_data, x='Year', y='CriticRating', ax=axes[1, 0
else:
    axes[1, 0].text(0.5, 0.5, "No Drama data", ha='center')

# Final KDE plot: Critic vs Audience Rating
k4 = sns.kdeplot(
    data=movie_data_clean, x='CriticRating', y='AudienceRating', ax=axes[1,
)
sns.kdeplot(
    data=movie_data_clean, x='CriticRating', y='AudienceRating', ax=axes[1,
)

plt.tight_layout()
plt.show()
```



In [64]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Cleaned dataset to avoid NaNs
movies_clean = movie_data.dropna(subset=['BudgetMillions', 'AudienceRating',

# Set dark theme with custom background
sns.set_style('dark', {'axes.facecolor': 'black'})

f, axes = plt.subplots(2, 2, figsize=(15, 15))

# Plot [0, 0]: Budget vs AudienceRating with 'inferno' and 'cool'
sns.kdeplot(
    data=movies_clean, x='BudgetMillions', y='AudienceRating',
    fill=True, cmap='inferno', ax=axes[0, 0]
)
sns.kdeplot(
    data=movies_clean, x='BudgetMillions', y='AudienceRating',
    cmap='cool', ax=axes[0, 0]
)
axes[0, 0].set_xlim(-20, 160)

# Plot [0, 1]: Budget vs CriticRating with 'inferno' and 'cool'
sns.kdeplot(
    data=movies_clean, x='BudgetMillions', y='CriticRating',
    fill=True, cmap='inferno', ax=axes[0, 1]
)
sns.kdeplot(
    data=movies_clean, x='BudgetMillions', y='CriticRating',
    cmap='cool', ax=axes[0, 1]
)
axes[0, 1].set_xlim(-20, 160)

# Plot [1, 0]: Violin plot for Drama genre
drama_data = movies_clean[movies_clean.Genre == 'Drama']
if not drama_data.empty:
    sns.violinplot(data=drama_data, x='Year', y='CriticRating', ax=axes[1, 0
else:
    axes[1, 0].text(0.5, 0.5, 'No Drama data', ha='center', va='center')

# Plot [1, 1]: CriticRating vs AudienceRating with Blues_r and gist_gray_r
sns.kdeplot(
    data=movies_clean, x='CriticRating', y='AudienceRating',
    fill=True, cmap='Blues_r', ax=axes[1, 1]
)
sns.kdeplot(
    data=movies_clean, x='CriticRating', y='AudienceRating',
    cmap='gist_gray_r', ax=axes[1, 1]
)

plt.tight_layout()
plt.show()
```


AudienceRating
BudgetMillions
CriticRating
BudgetMillions
CriticRating
Year
2007
2008
2009
2010
2011
AudienceRating
CriticRating

```
In [ ]:
```