In [107…
```
import numpy as np  # matrix
import pandas as pd  # tables
import matplotlib.pyplot as plt
import seaborn as sns
```

In [108…
```
file=pd.read_csv(r"C:\Users\WELCOME\Documents\Data Science\July 2025\24th july-M
file
```

Out[108…

| | 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 [109…
```
file.head()
```

Out[109…

| | 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 [110…
```
file.columns
```

Out[110…
```
Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
       'Budget (million $)', 'Year of release'],
      dtype='object')
```

In [111…
```
file.columns=['Film','Genre','CriticRating','AudienceRating','BudgetMillions','Y
```

In [112…
```
file.head(1)   # remove spaces and % removed noisy characters
```

Out[112…

| | Film | Genre | CriticRating | AudienceRating | BudgetMillions | Year |
|---|---|---|---|---|---|---|
| **0** | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |

In [113…
```
file.shape
```

Out[113… (559, 6)

In [114…
```
file.describe()   # descriptive statistics
```

Out[114…

| | CriticRating | AudienceRating | 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 [115…
```
file.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 [116…
```
# change film type 'object' to type 'category'
file.Film = file.Film.astype('category')
```

In [117…
```
file.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 [118…
```
# now we will change genre to category and year to category
file.Genre = file.Genre.astype('category')
file.Year = file.Year.astype('category')
```

In [119…
```
file.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    category
dtypes: category(3), int64(3)
memory usage: 36.5 KB
```

In [120…
```
file.Year # is it real no. year you can take average,min,max but out come have n
```

Out[120…
```
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 [121…
```
file.Genre.cat.categories
```

Out[121…
```
Index(['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance',
       'Thriller'],
      dtype='object')
```

In [122…
```
# now when we see the describe() you will get only integer value mean, sd ....so
file.describe()
```

Out[122...

| | CriticRating | AudienceRating | 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 |

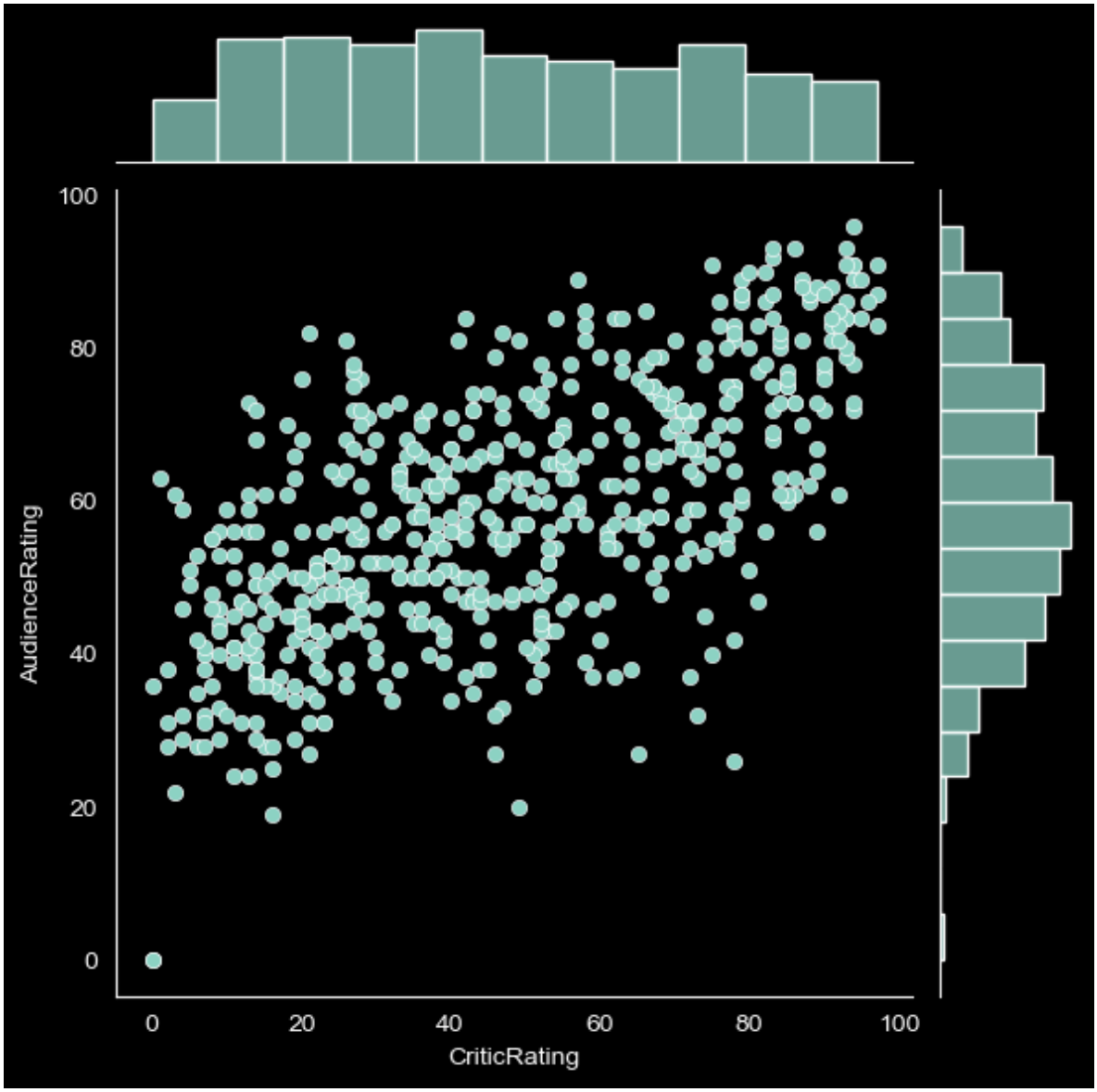
In [123...

```
# how to working with joint plots
from matplotlib import pyplot as plt   # visualization
import seaborn as sns  # advance visualization

#%matplotlib inline means all the plot the plot should inside the line
import warnings
warnings.filterwarnings('ignore') # ignore os error
```

- **basically joint plot is scatter plot & it find the relation between audience and critics**
- **also if you look up you can find the uniform distribution (critics)and normal disctribution(audience)**

In [124...

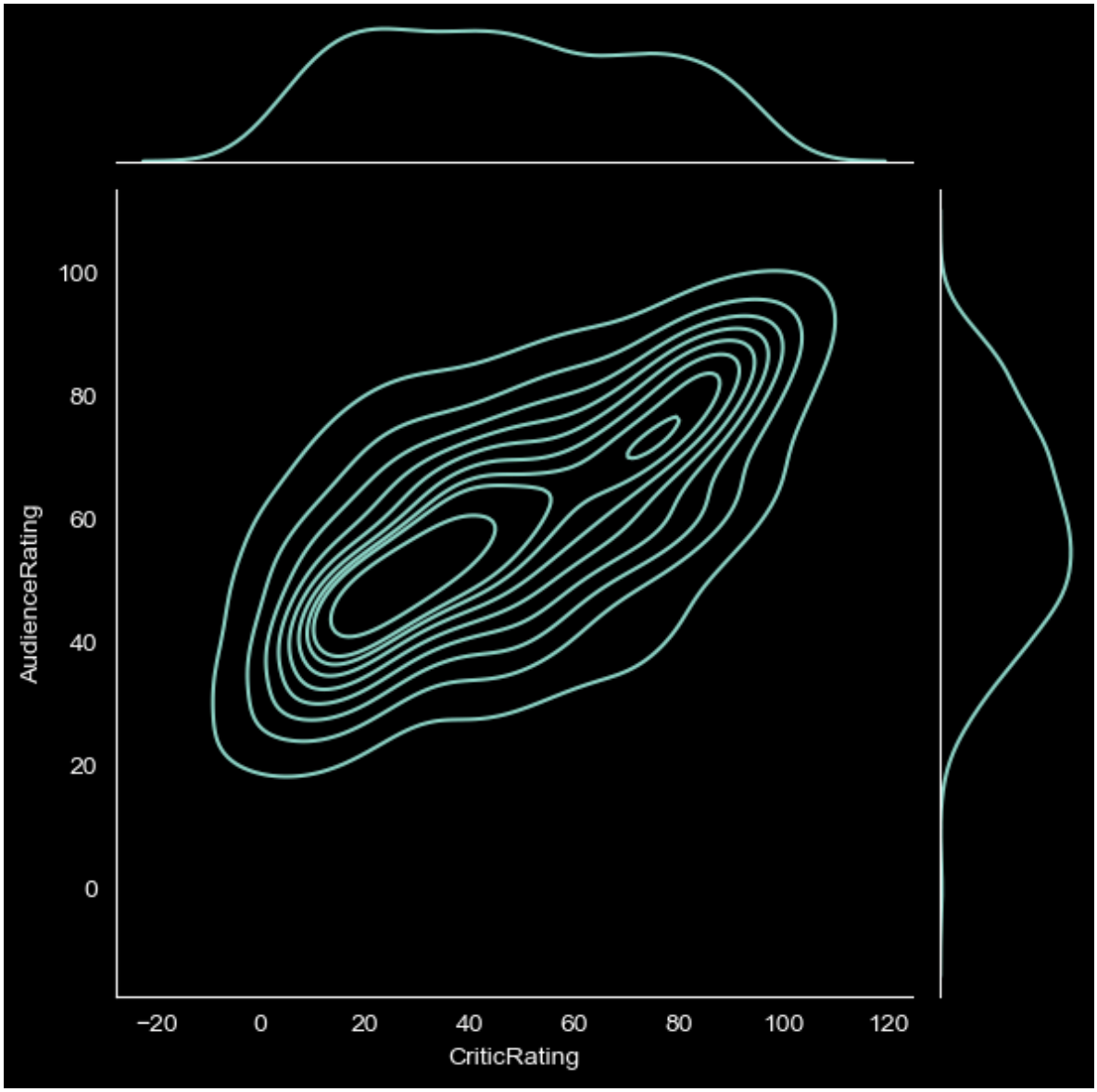
```
j=sns.jointplot(data = file, x='CriticRating', y='AudienceRating', kind='scatter

# plt has no attribute joinplot so use sns
# Audience rating is more dominant than critics rating
# Based on this we find out as most people are most liklihood to watch audience
# let me explain the excel - if you filter audience rating & critic rating. crit
```



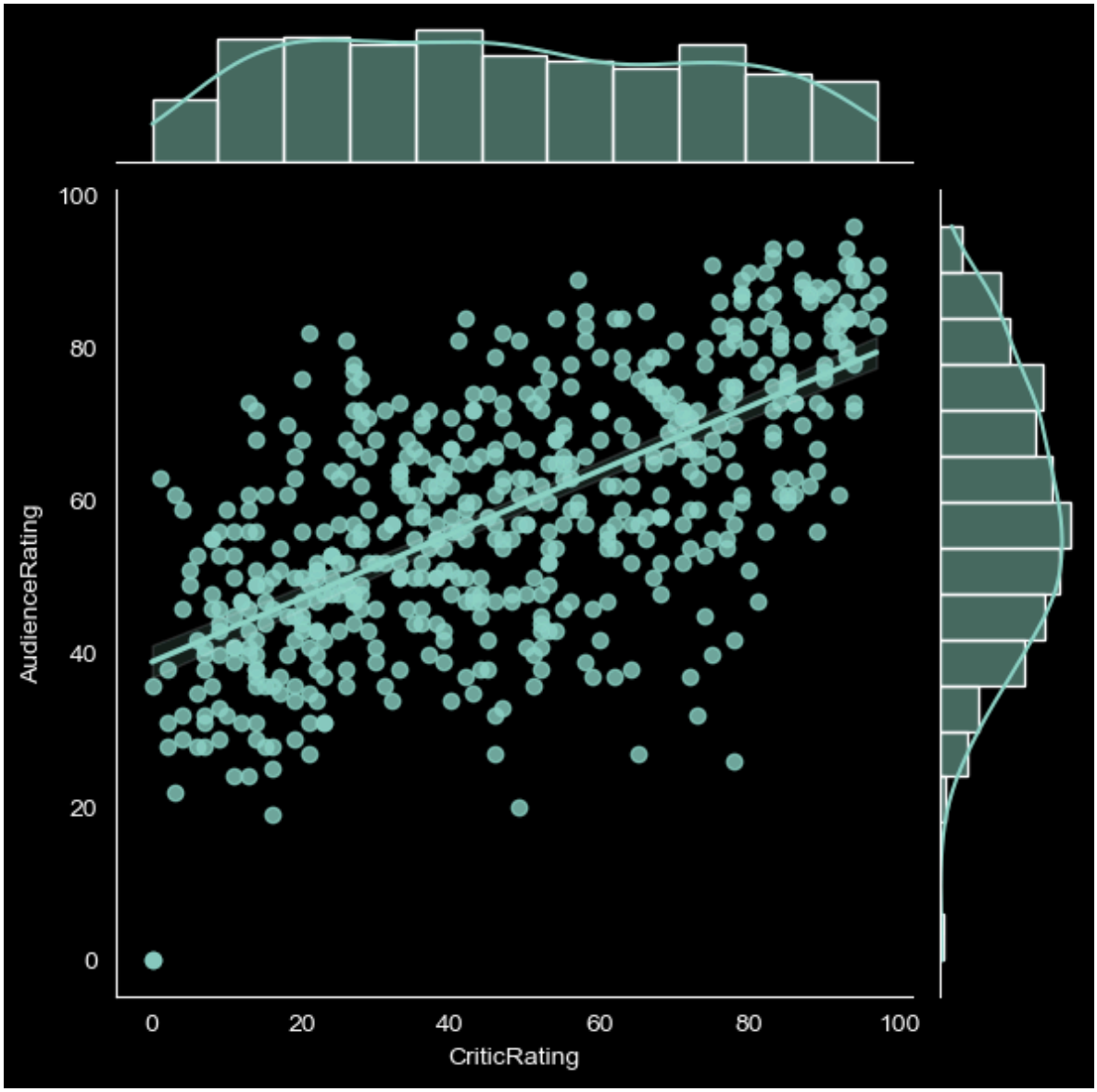

## kde: kernel density plot :if we merge all the data point , we get new shape

In [125…

```
j=sns.jointplot(data = file, x='CriticRating', y='AudienceRating', kind='kde')
```

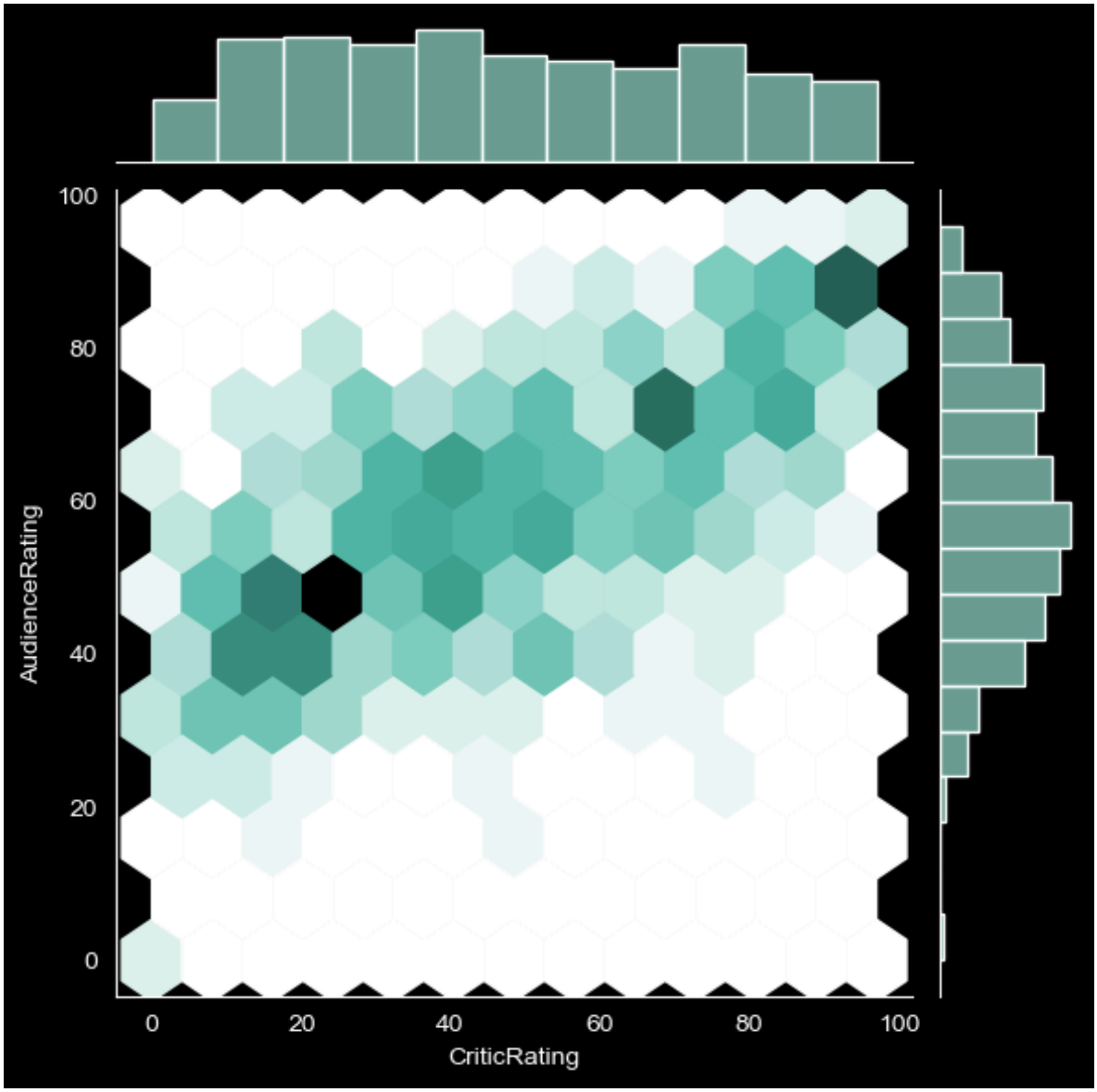



In [126…
```
j = sns.jointplot( data = file, x = 'CriticRating', y = 'AudienceRating', kind='
```



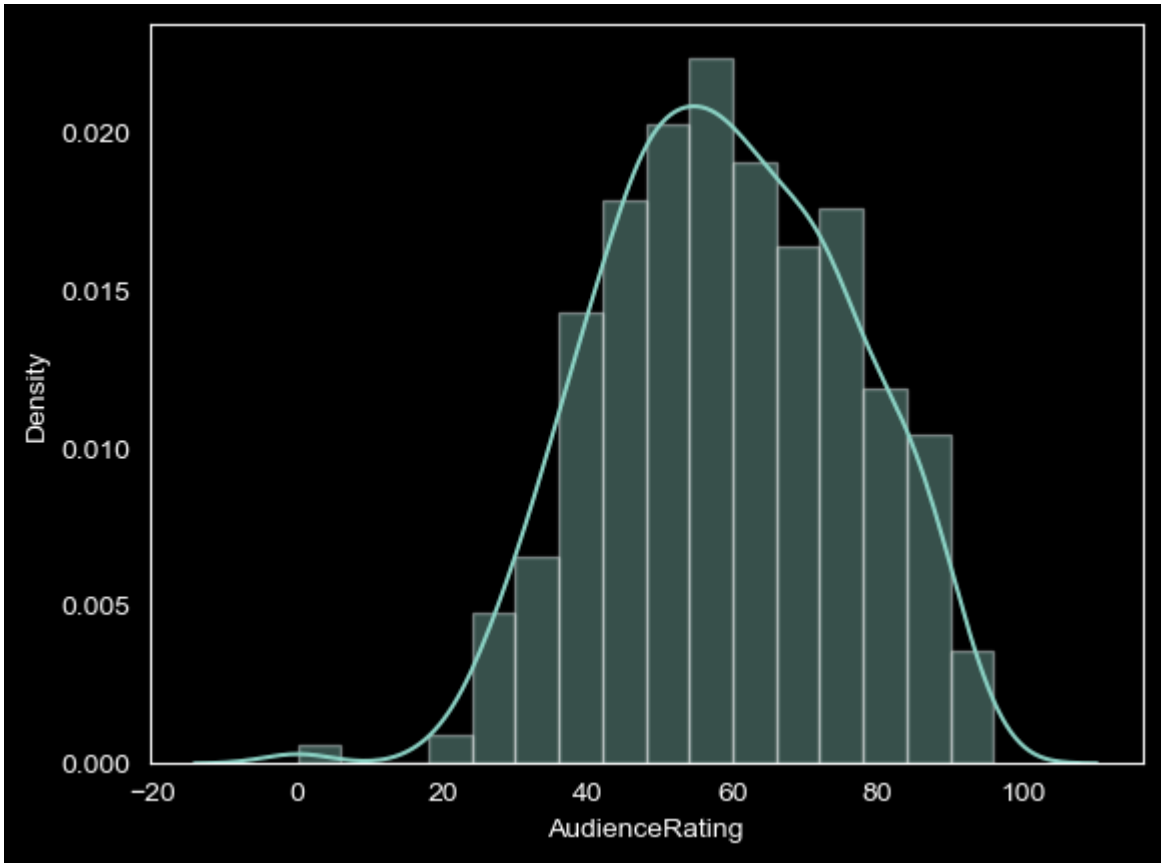

In [127…

```
j = sns.jointplot( data = file, x = 'CriticRating', y = 'AudienceRating', kind='

#j = sns.jointplot( data = movies, x = 'CriticRating', y = 'AudienceRating', kin
```

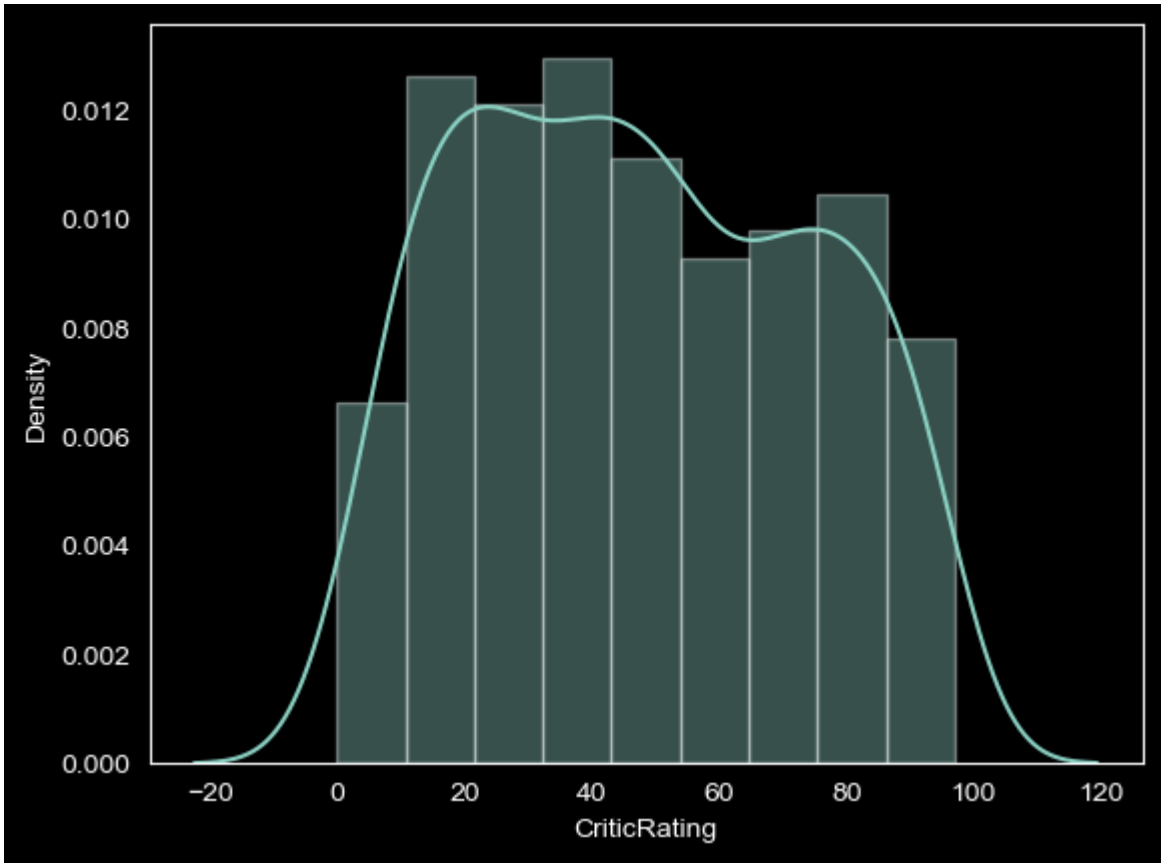



In [128…

```
#Histograms
m1 = sns.distplot(file.AudienceRating)  # there is normal distribution or bell c

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

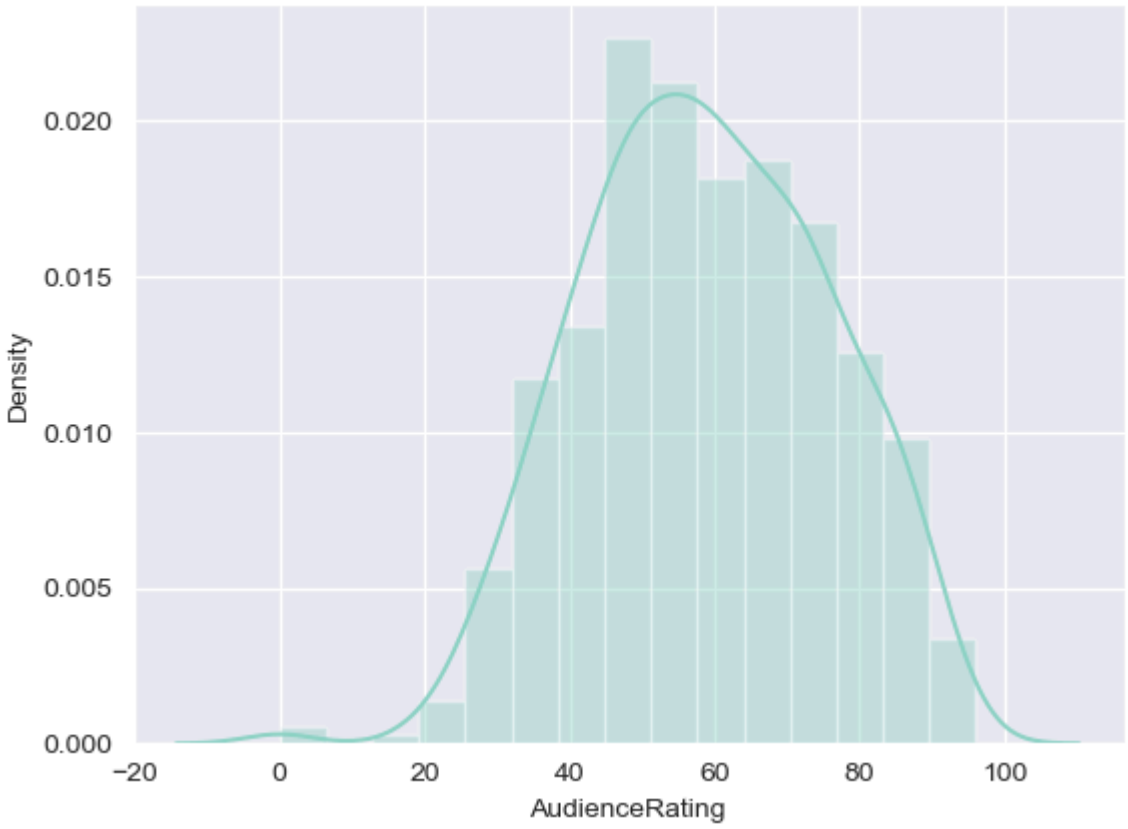



In [129…
```
m1 = sns.distplot(file.CriticRating)
```

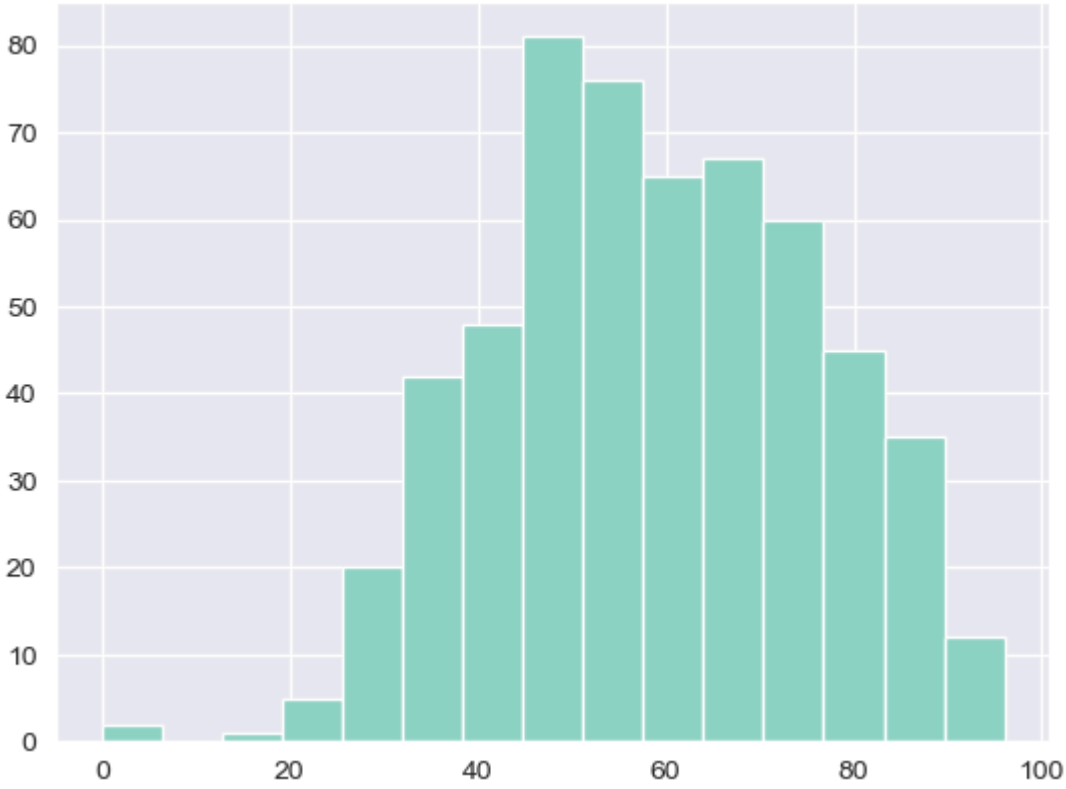



In [130…
```
sns.set_style('darkgrid')
```

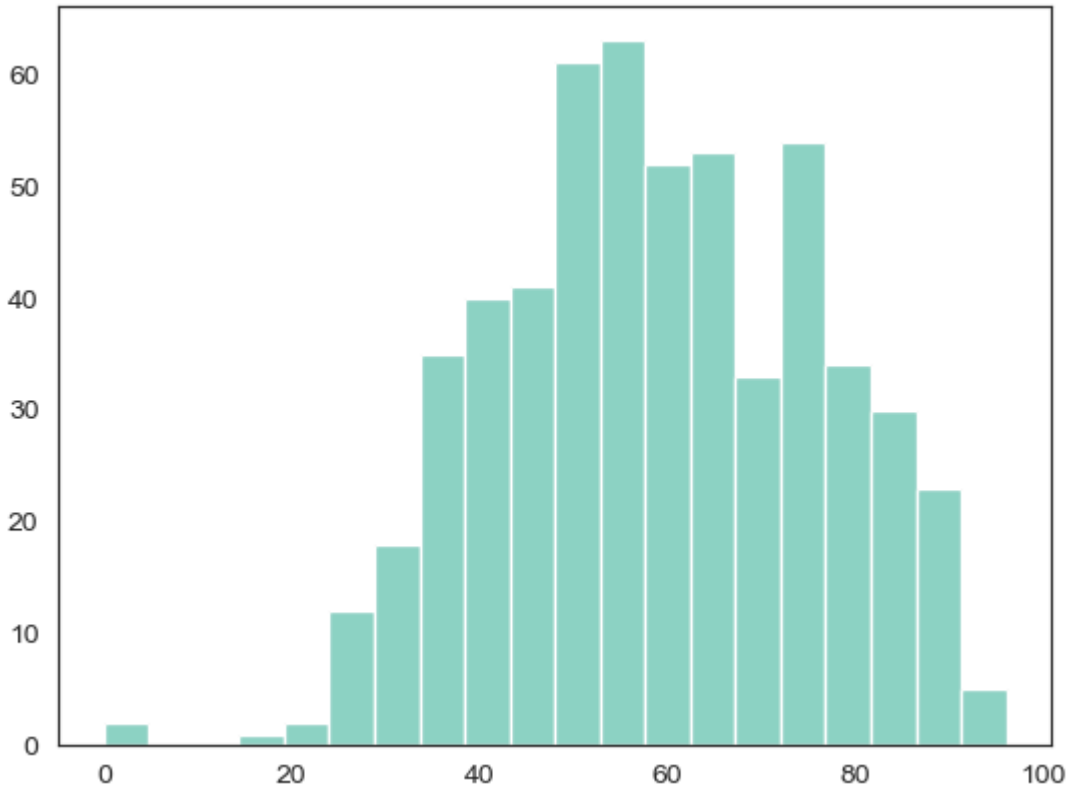
In [131…
```
m2 = sns.distplot(file.AudienceRating, bins = 15)
```



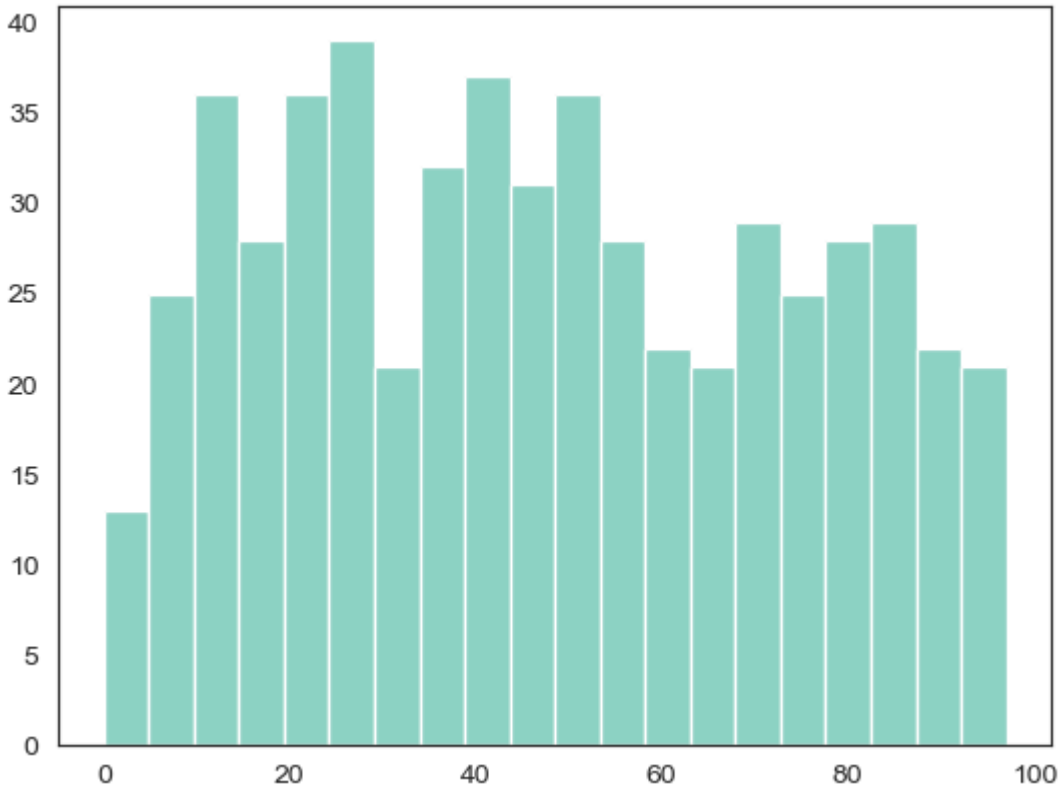

In [132…
```
#sns.set_style('darkgrid')
n1 = plt.hist(file.AudienceRating, bins=15)
```




In [133…
```
sns.set_style('white') #normal distribution & called as bell curve
n1 = plt.hist(file.AudienceRating, bins=20)
```



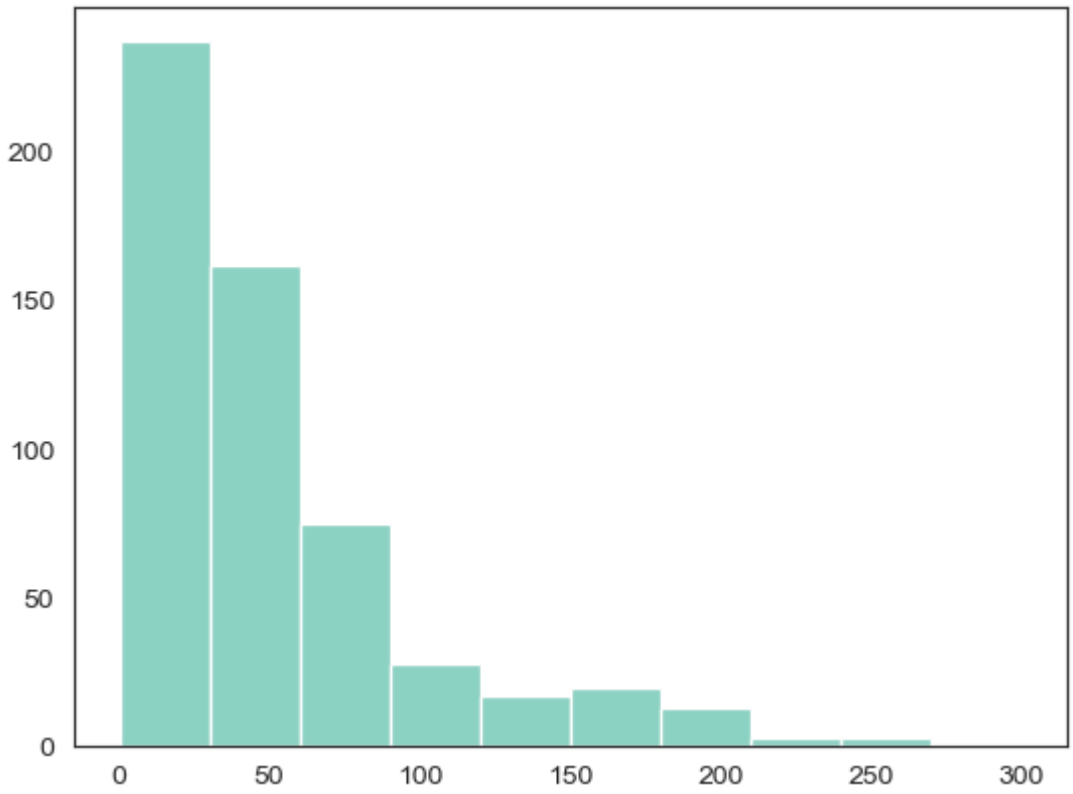

In [134…
```
n1 = plt.hist(file.CriticRating, bins=20) #uniform distribution
```




# Creating stacked histograms & this is bit tough to understand

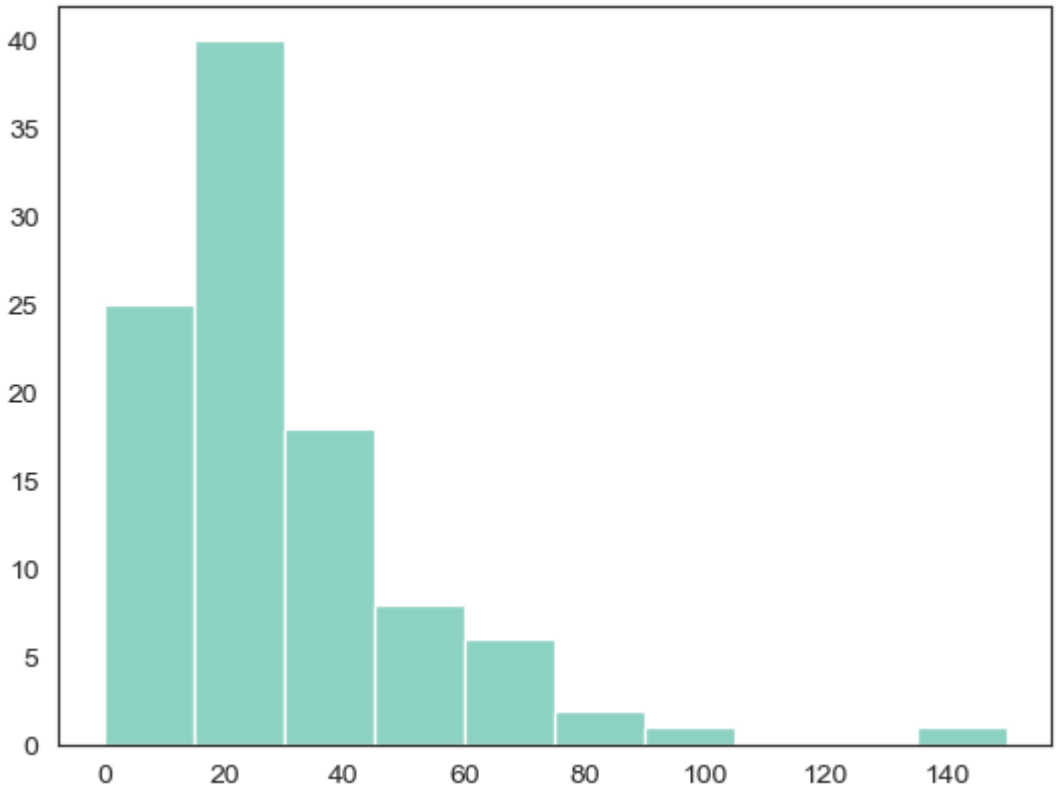
In [135…
```
#h1 = plt.hist(file.BudgetMillions)
```

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




In [136…

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




In [137…

```
file.head()
```

Out[137…

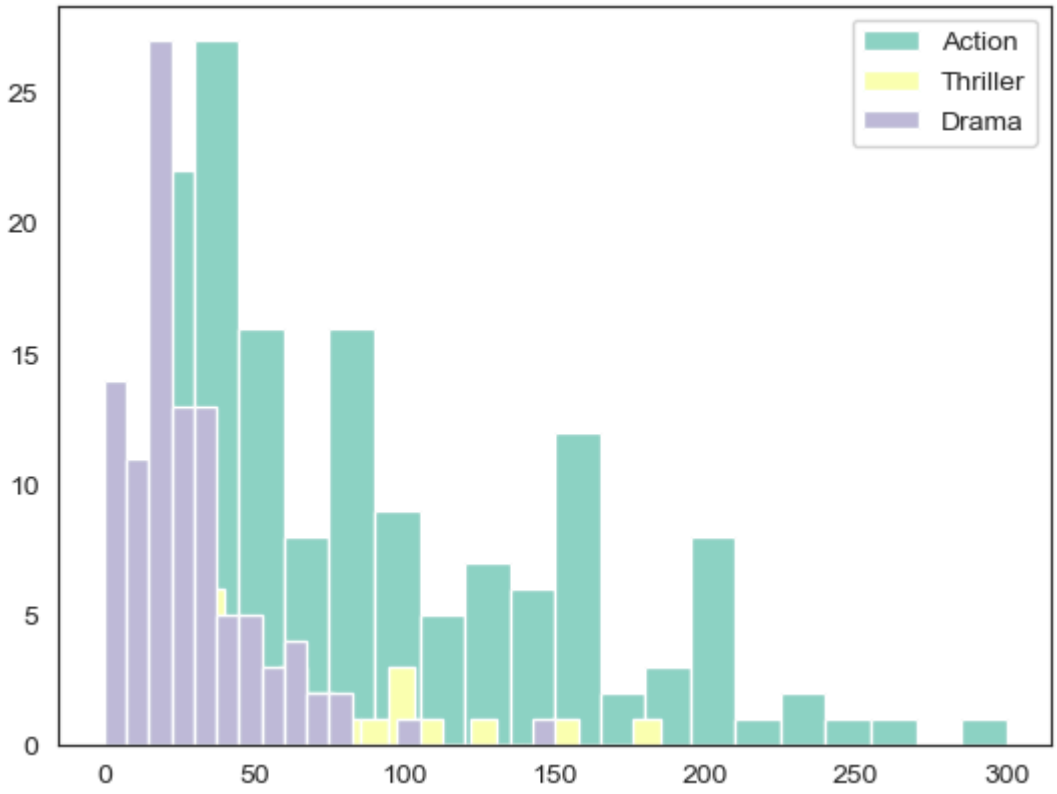
| | Film | Genre | CriticRating | AudienceRating | BudgetMillions | Year |
|---|---|---|---|---|---|---|
| **0** | (500) Days of Summer | Comedy | 87 | 81 | 8 | 2009 |
| **1** | 10,000 B.C. | Adventure | 9 | 44 | 105 | 2008 |
| **2** | 12 Rounds | Action | 30 | 52 | 20 | 2009 |
| **3** | 127 Hours |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

In [138…

```
#movies.Genre.unique()
```

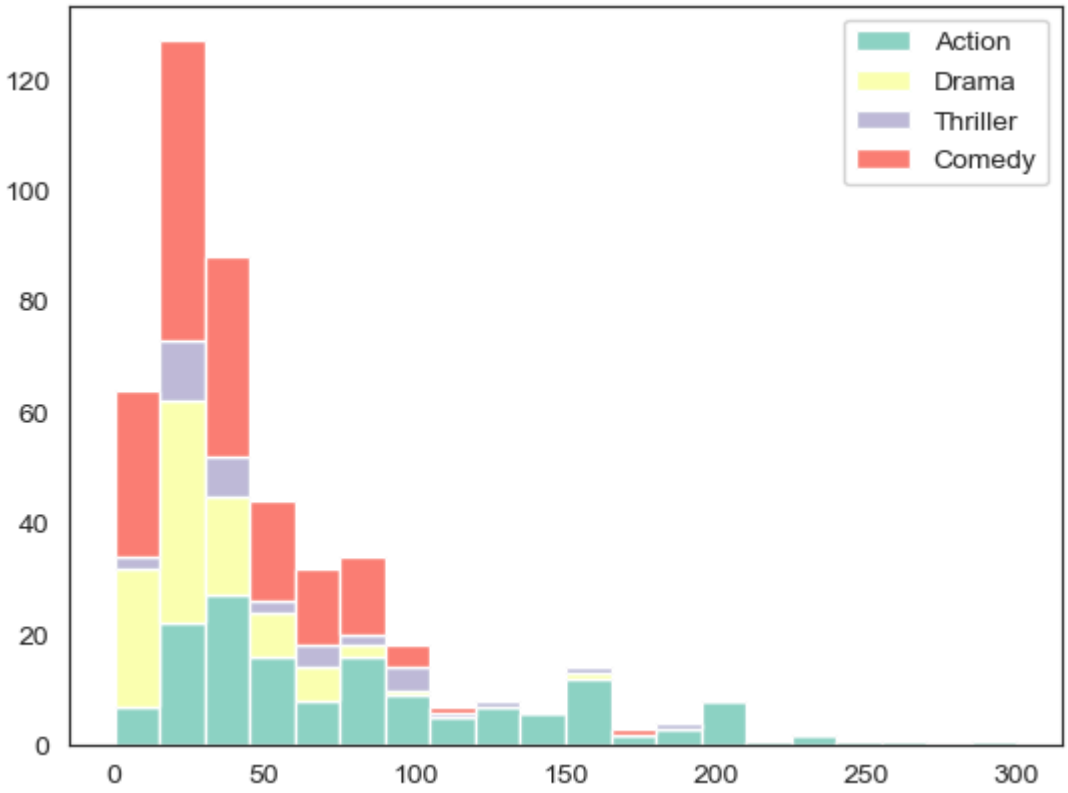
In [139…

```
# Below plots are stacked histogram becuase overlaped

plt.hist(file[file.Genre == 'Action'].BudgetMillions, bins = 20,label='Action')
plt.hist(file[file.Genre == 'Thriller'].BudgetMillions, bins = 20,label='Thrille
plt.hist(file[file.Genre == 'Drama'].BudgetMillions, bins = 20,label='Drama')
plt.legend()
plt.show()
```




In [140…

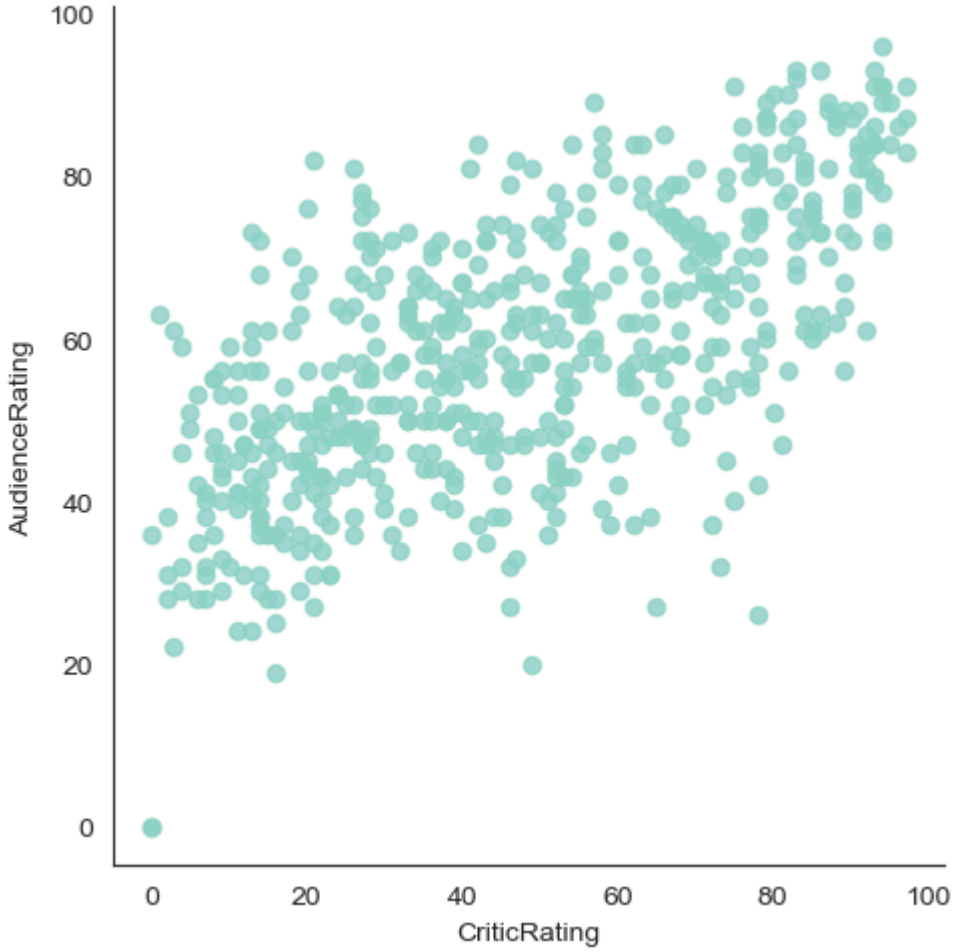
```
plt.hist([file[file.Genre == 'Action'].BudgetMillions,\
          file[file.Genre == 'Drama'].BudgetMillions, \
          file[file.Genre == 'Thriller'].BudgetMillions, \
          file[file.Genre == 'Comedy'].BudgetMillions],
         bins = 20, stacked = True,label=('Action','Drama','Thriller','Comedy'))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




In [141…
```
# if you have 100 categories you cannot copy & paste all the things

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

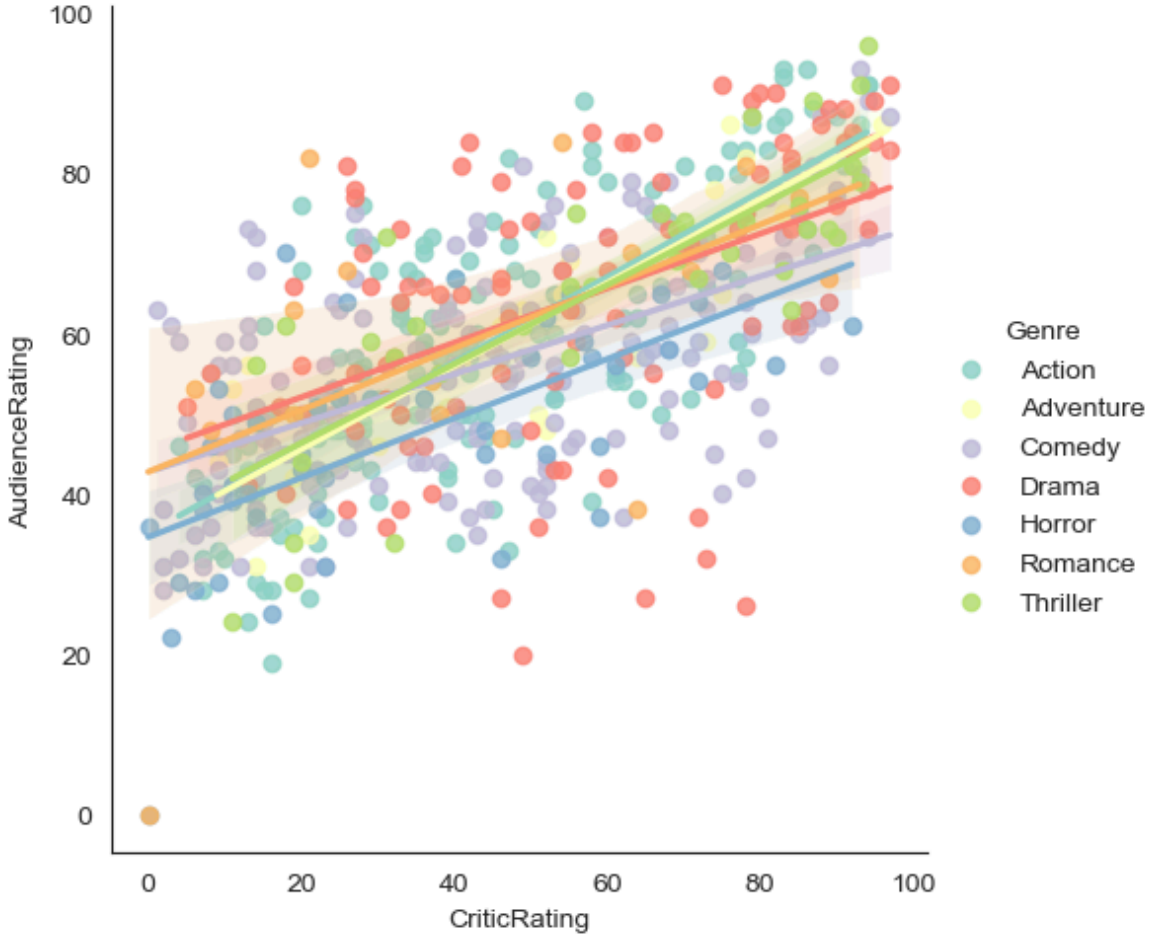
```
Action
Adventure
Comedy
Drama
Horror
Romance
Thriller
```

In [142…
```
vis1 = sns.lmplot(data=file, x='CriticRating', y='AudienceRating',\
                  fit_reg=False)
```



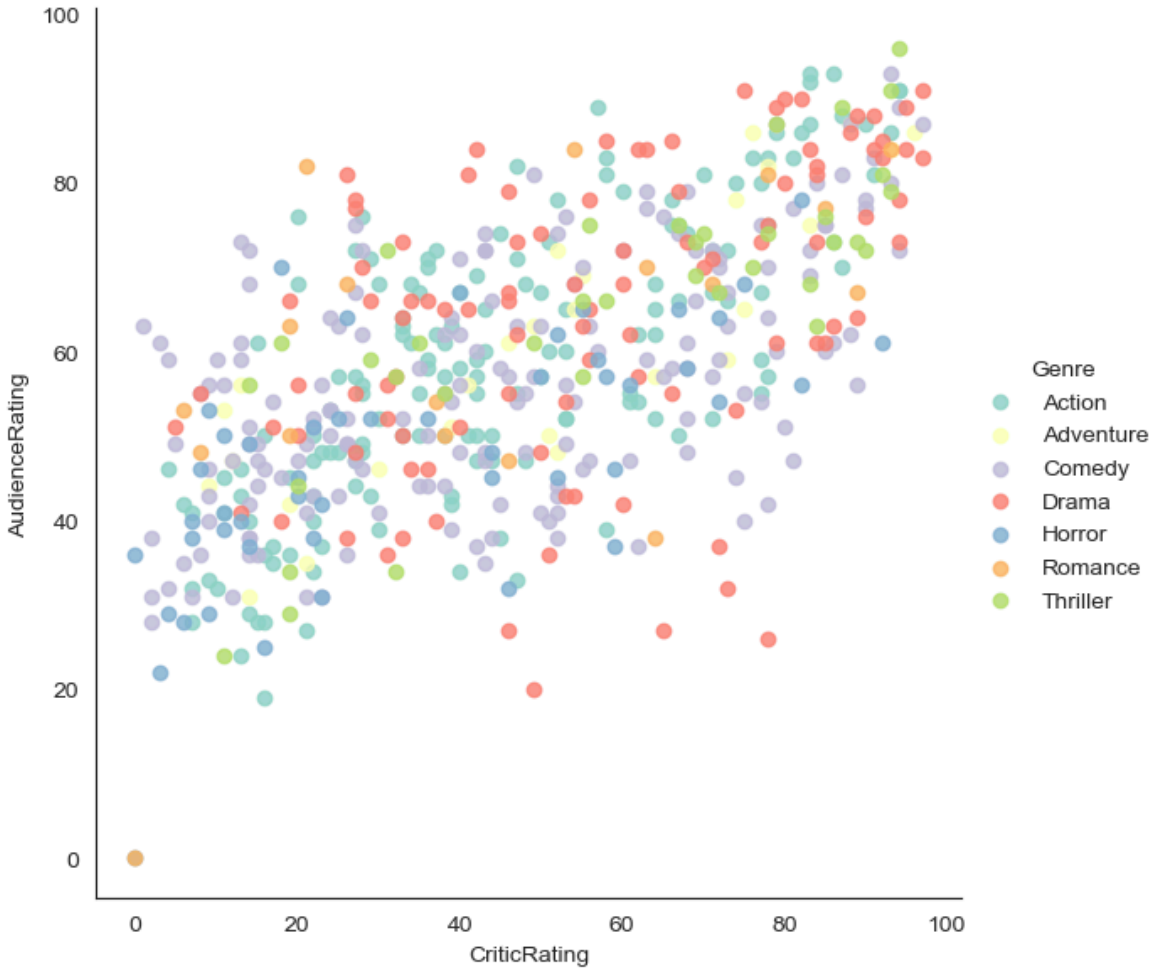

In [143…

```
vis1 = sns.lmplot(data=file, x='CriticRating', y='AudienceRating',
                  fit_reg=True, hue = 'Genre')           # hue: refers to the us
```



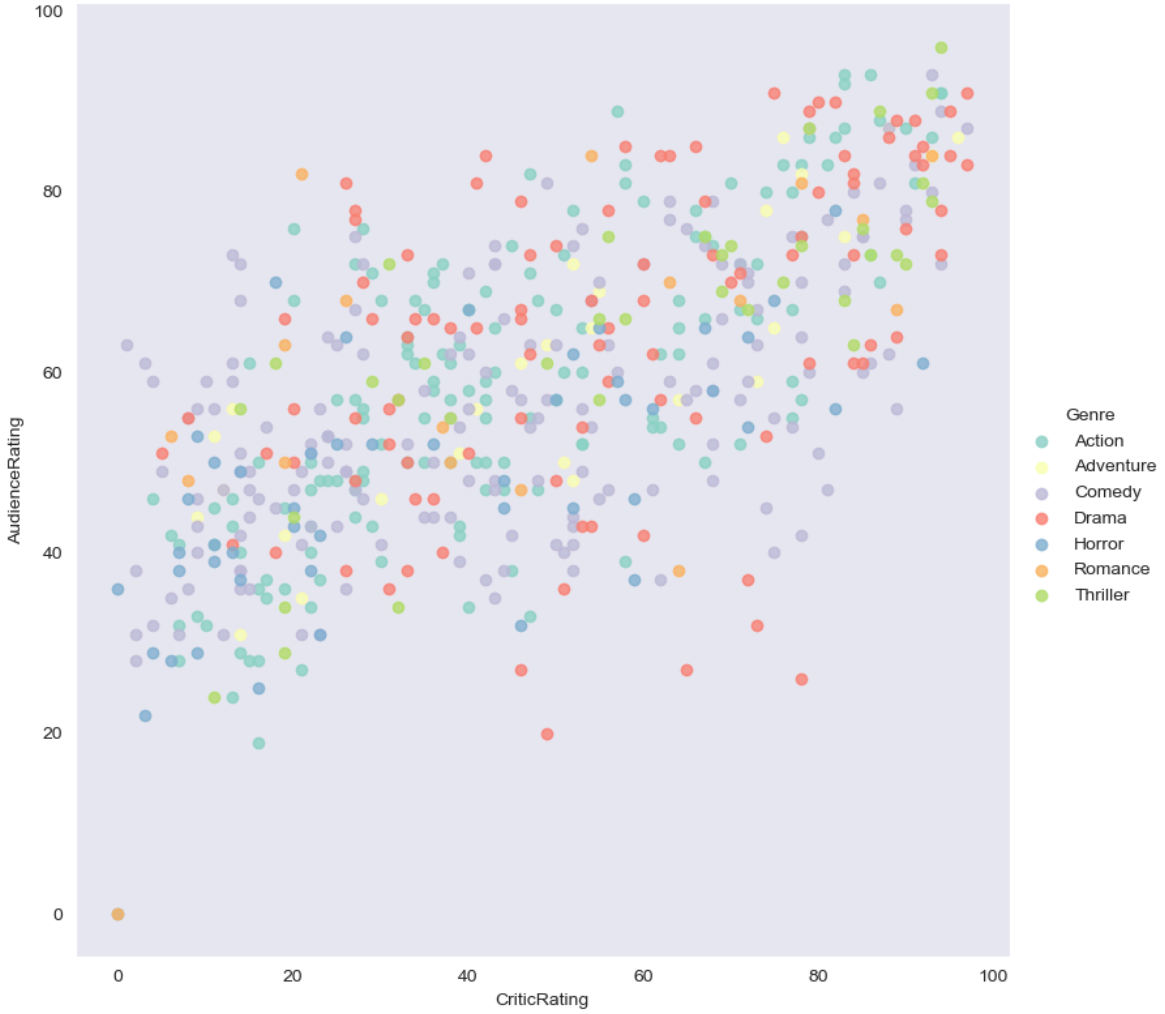

In [144…

```
vis1 = sns.lmplot(data=file, x='CriticRating', y='AudienceRating',
                  fit_reg=False, hue = 'Genre', height = 6,aspect=1)
```



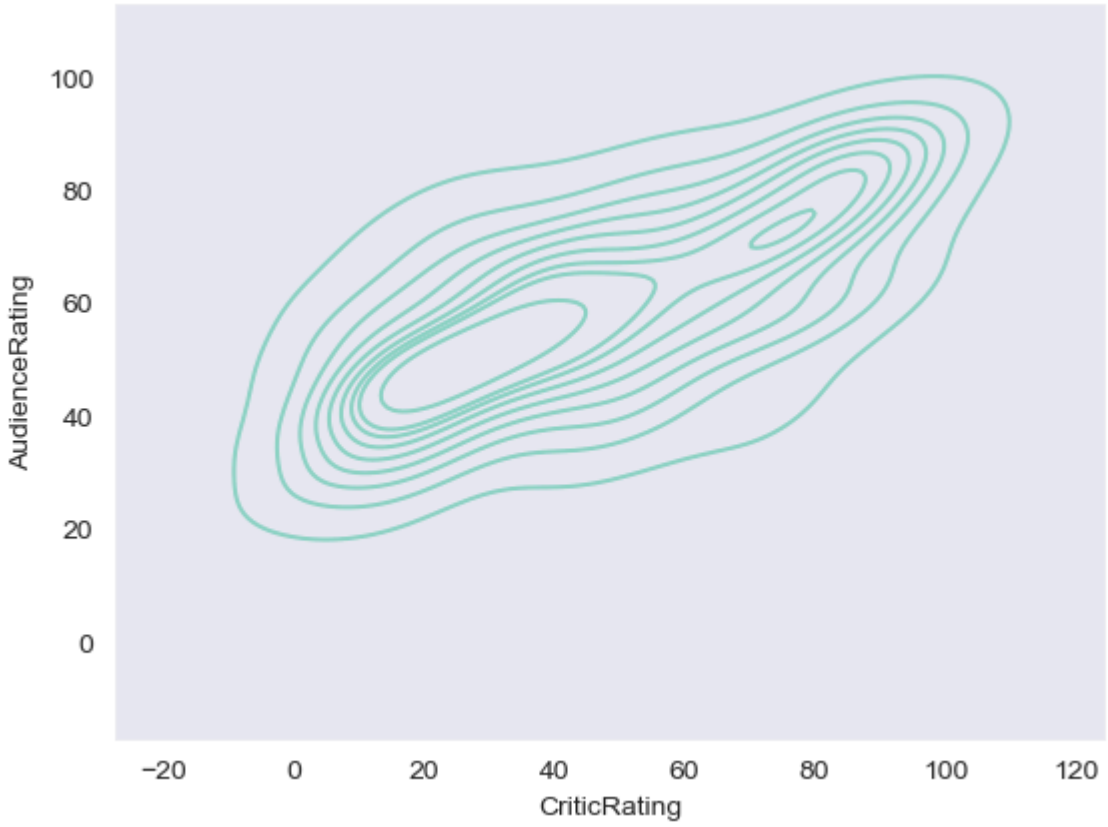

In [145…

```
sns.set_style('dark')
vis1 = sns.lmplot(data=file, x='CriticRating', y ='AudienceRating',
           fit_reg=False, hue = 'Genre', aspect=1, height= 8)
```

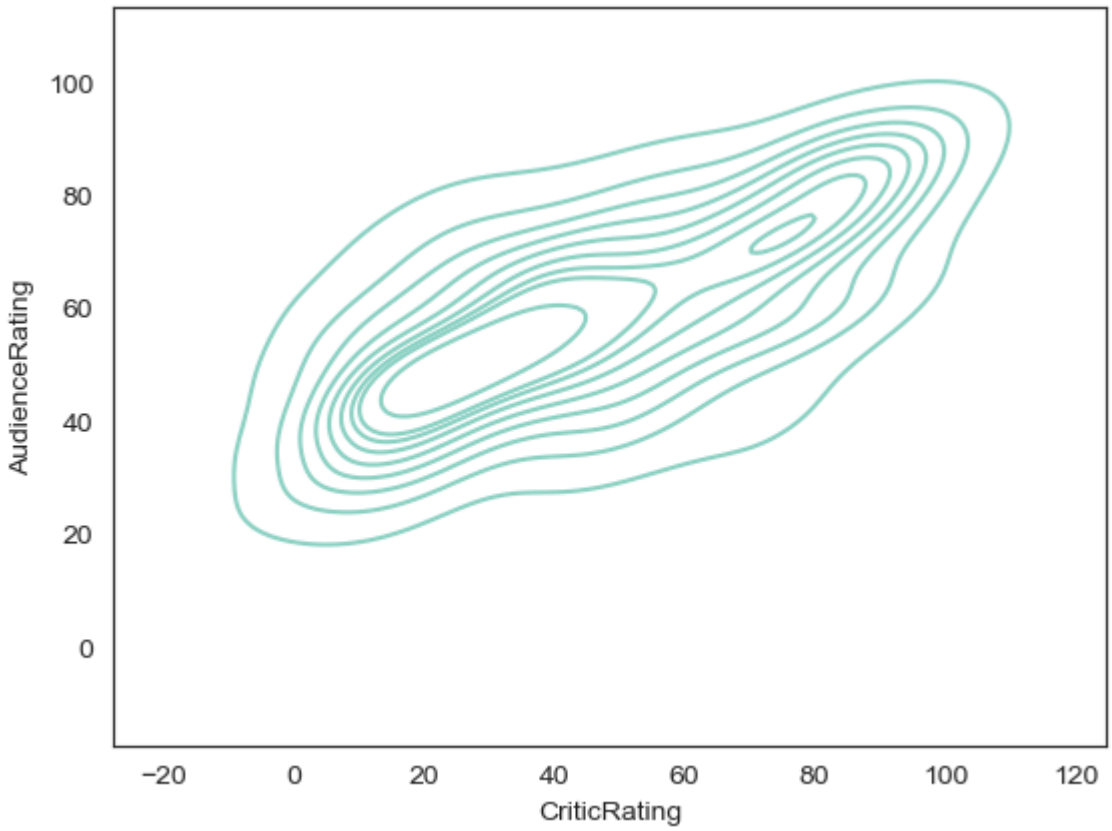



In [146…
```
# Kernal Density Estimate plot ( KDE PLOT)
# how can i visulize audience rating & critics rating . using scatterplot
```

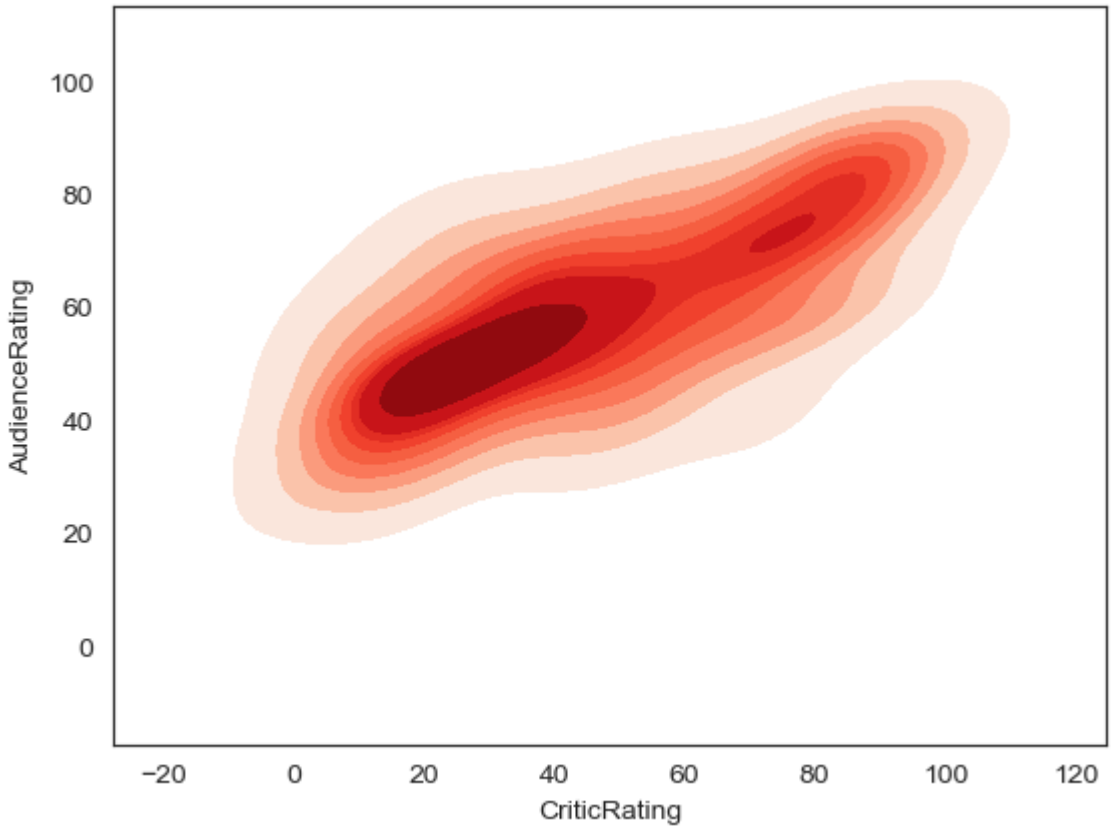
In [147…
```
k1 = sns.kdeplot(x=file['CriticRating'],y=file['AudienceRating'])
plt.style.use('dark_background')
sns.set_style('white')
plt.show()
```



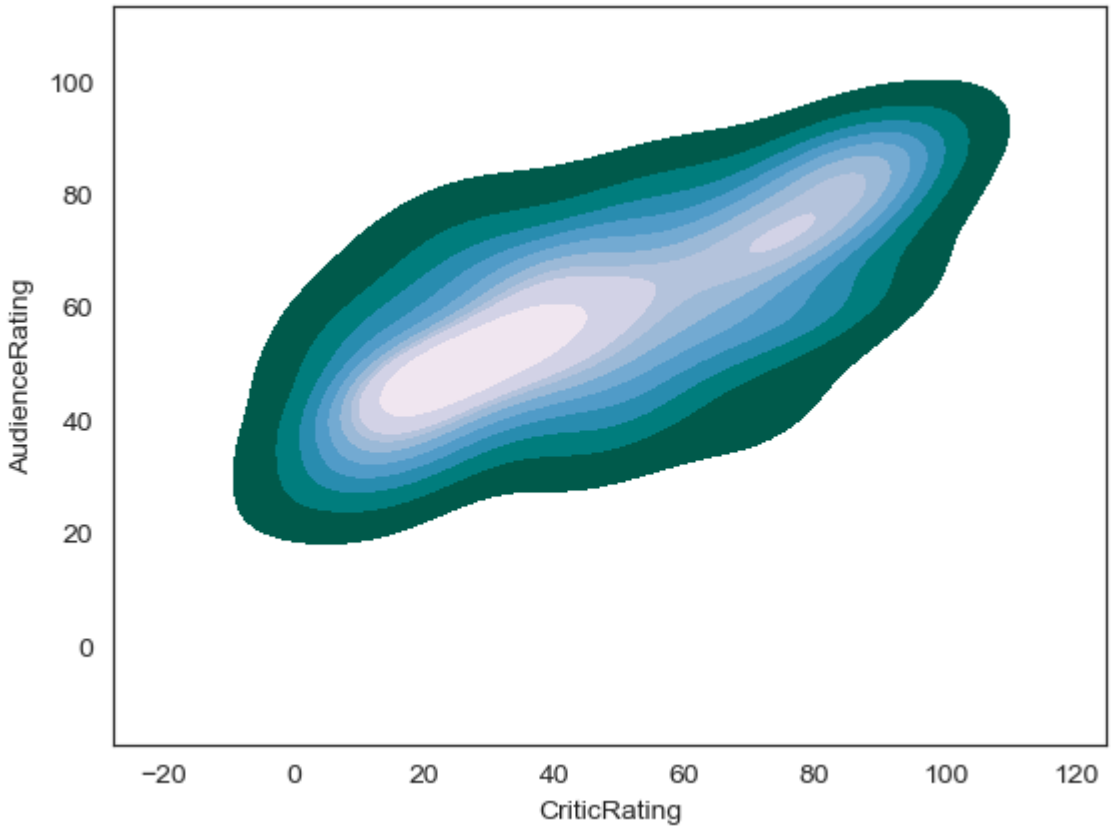

In [148…
```
k1 = sns.kdeplot(x=file['CriticRating'],y=file['AudienceRating'])

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



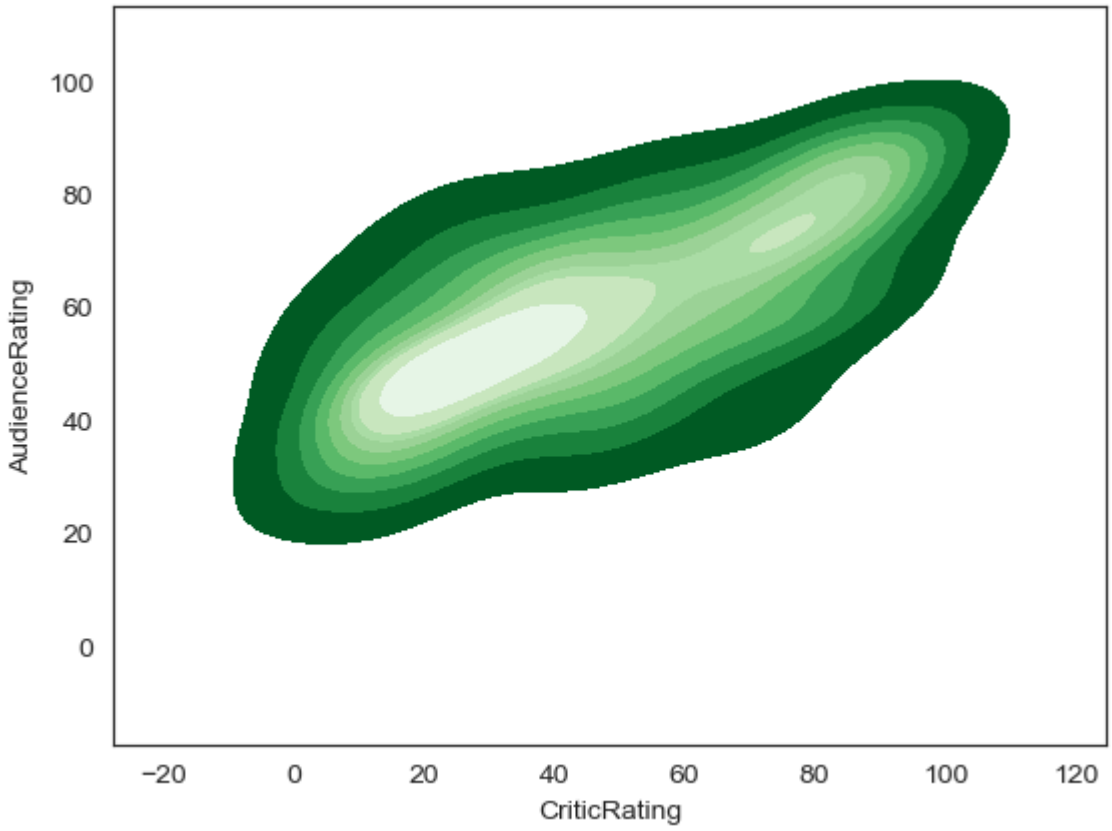

In [149…
```
k1 = sns.kdeplot(x=file.CriticRating,y=file.AudienceRating,shade = True,shade_lo
```



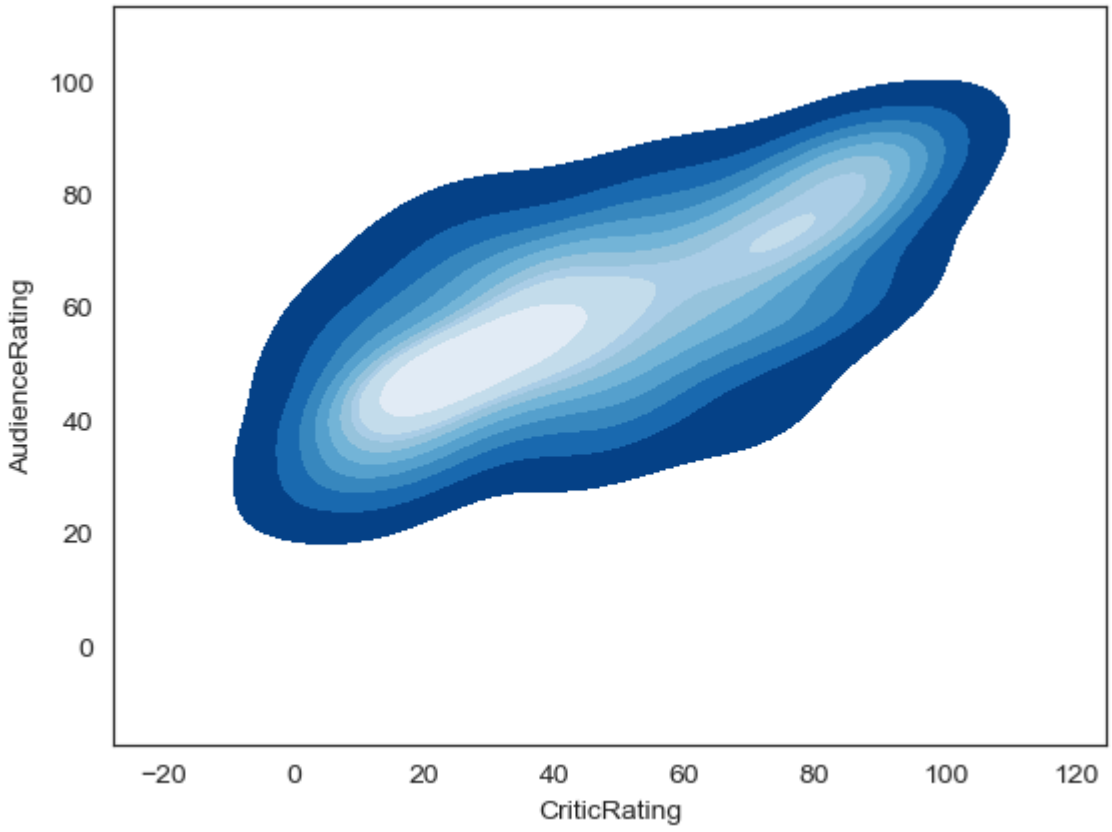

In [150…
```
k2 = sns.kdeplot(x= file.CriticRating,y= file.AudienceRating,shade=True,shade_lo
```



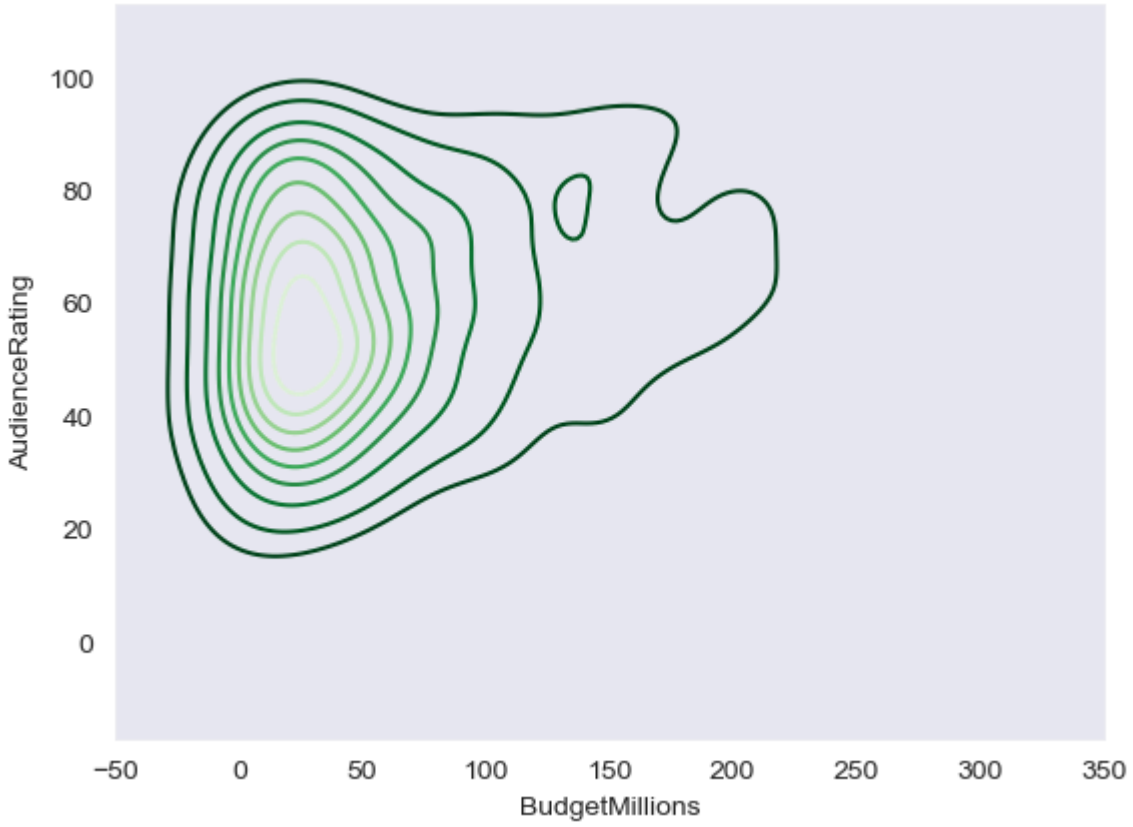

In [151…
```
k2 = sns.kdeplot(x= file.CriticRating,y= file.AudienceRating,shade=True,shade_lo
```




In [152…
```
k2 = sns.kdeplot(x= file.CriticRating,y= file.AudienceRating,shade=True,shade_lo
```

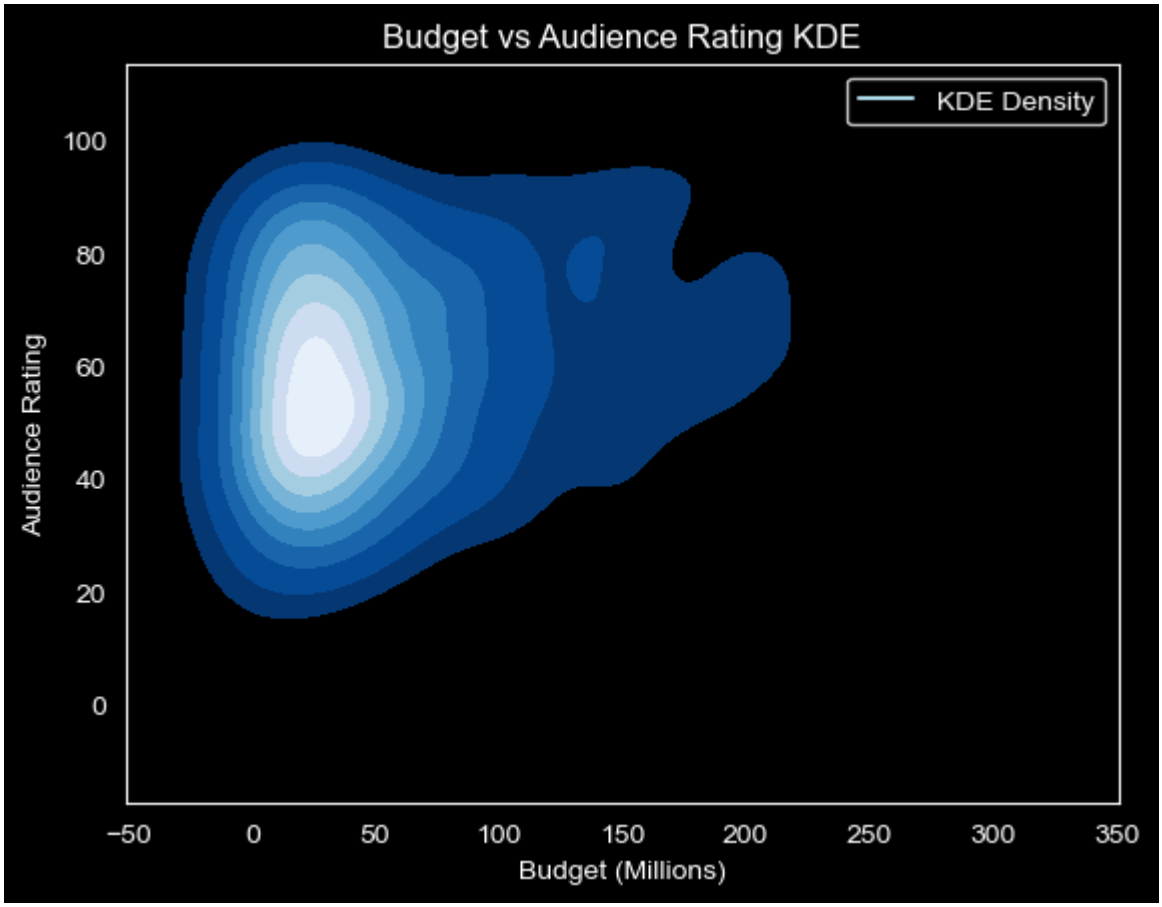



In [153…
```
sns.set_style('dark')
k1 = sns.kdeplot(x= file.BudgetMillions,y= file.AudienceRating,shade_lowest=Fals
```



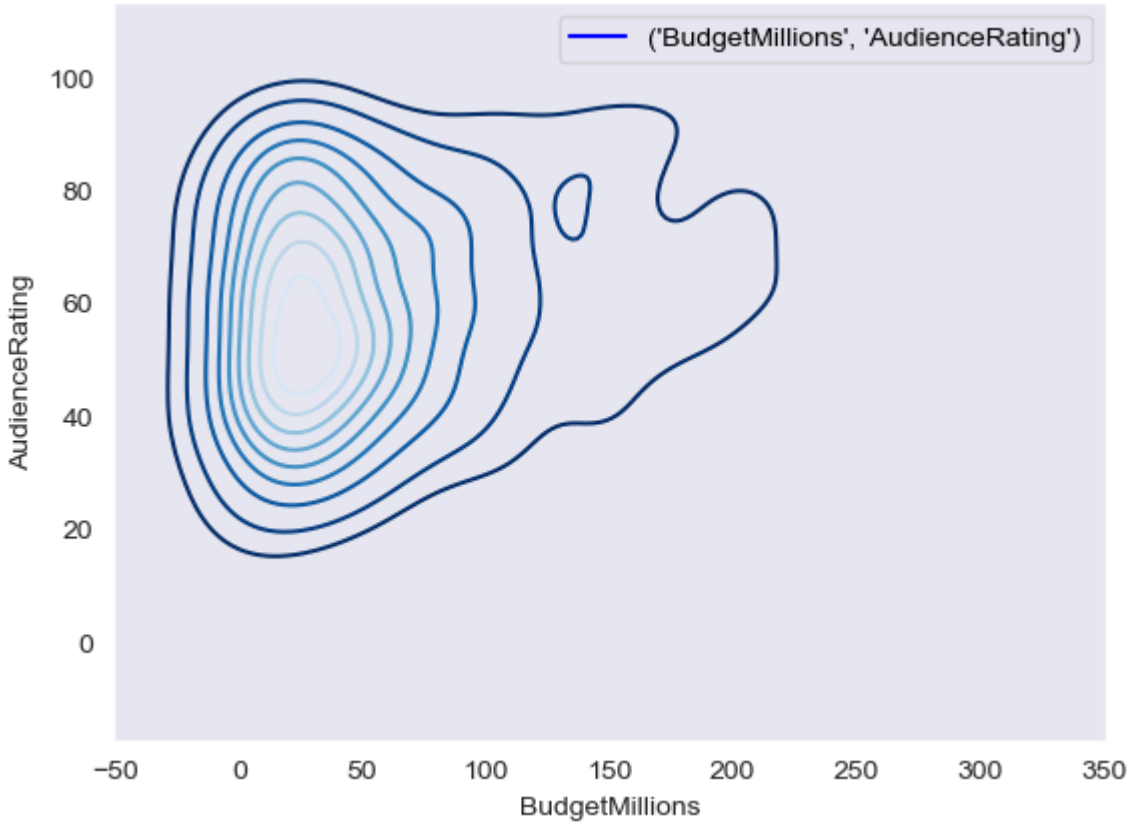

In [154…

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('dark')
plt.style.use('dark_background')
# KDE plot (2D)
sns.kdeplot(
x=file.BudgetMillions,
y=file.AudienceRating,
fill=True,
cmap='Blues_r',
thresh=0.05
)
# Add fake line just for the legend
plt.plot([], [], color='lightblue', label='KDE Density')
# Add title and labels
plt.title("Budget vs Audience Rating KDE", color='white')
plt.xlabel("Budget (Millions)", color='white')
plt.ylabel("Audience Rating", color='white')
# Add legend
plt.legend(facecolor='black', edgecolor='white', labelcolor='white')
plt.show()
```



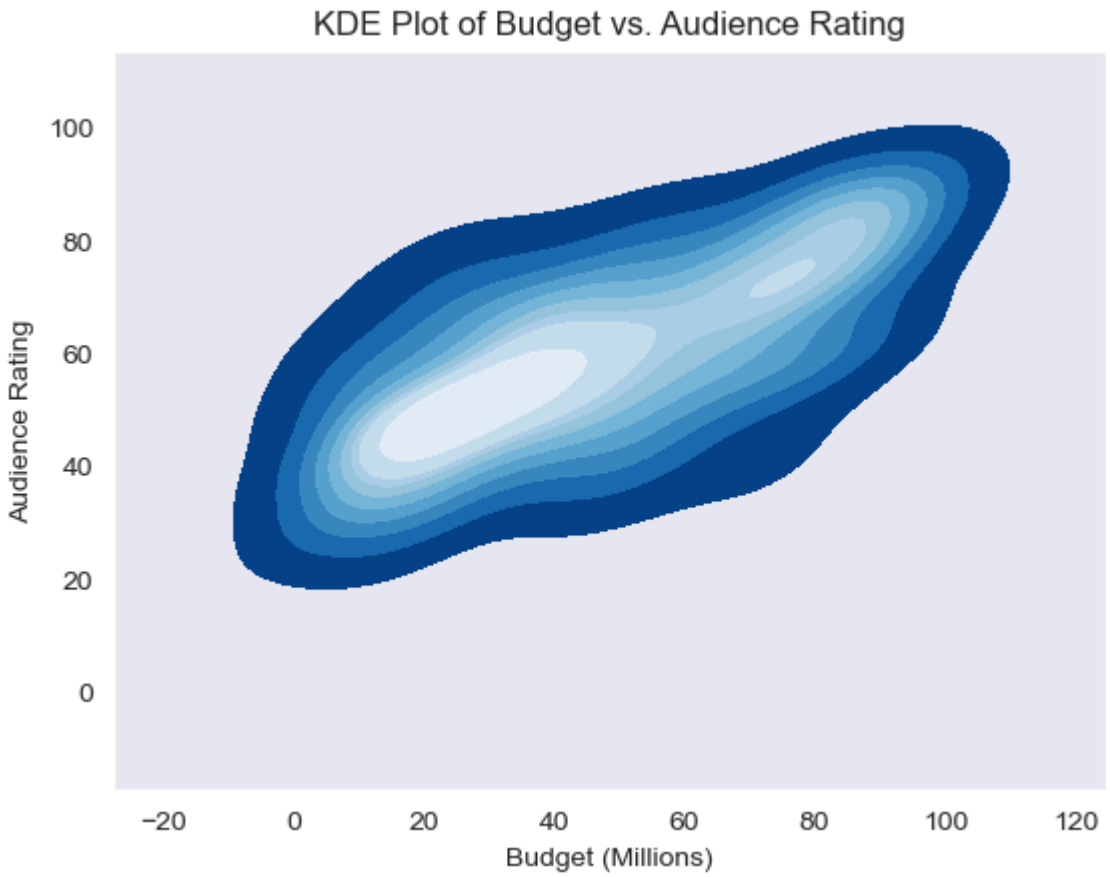

In [155…

```
sns.set_style('dark')
k1 = sns.kdeplot(
x=file.BudgetMillions,
y=file.AudienceRating,
shade_lowest=False,
cmap='Blues_r'
)
plt.plot( [], [], color='blue',label=('BudgetMillions','AudienceRating'))
plt.legend( labelcolor='black')
plt.show()
```



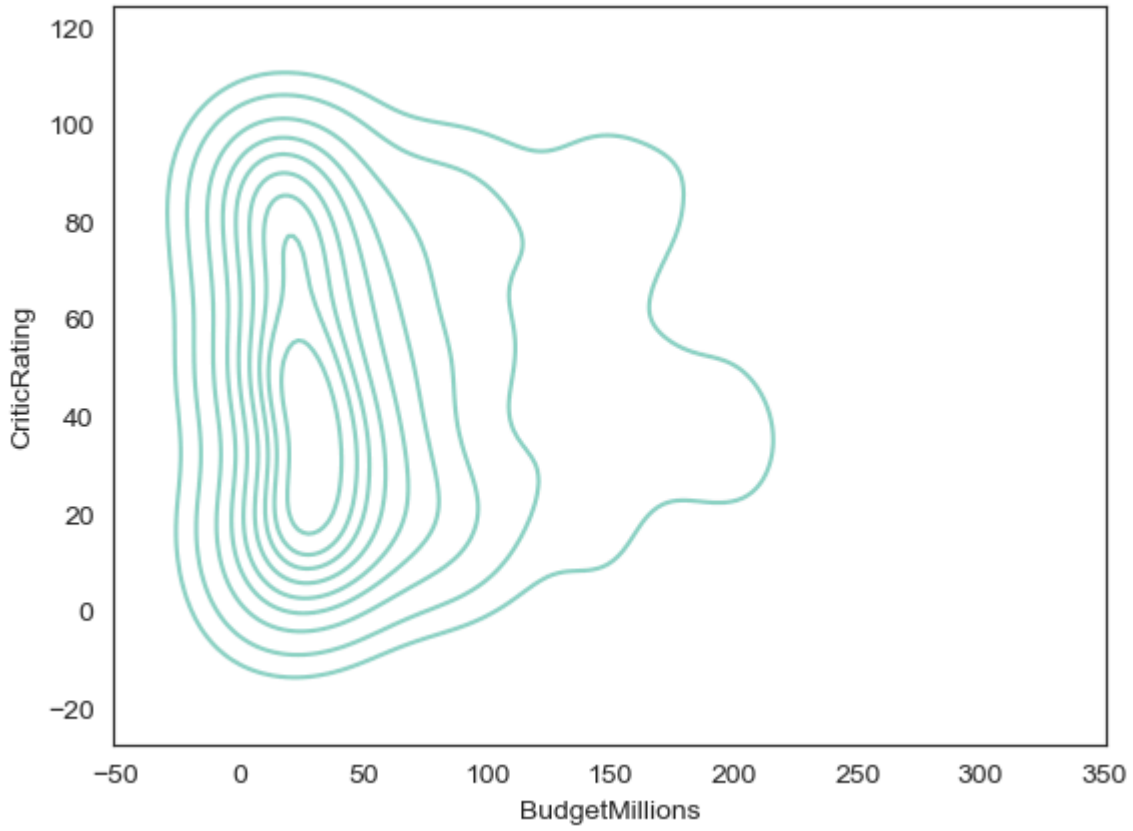

In [156…

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.kdeplot(
    x=file.CriticRating,
    y=file.AudienceRating,
    fill=True,
    cmap='Blues_r',
    thresh=0.05
)

plt.title("KDE Plot of Budget vs. Audience Rating")
plt.xlabel("Budget (Millions)")
plt.ylabel("Audience Rating")
plt.show()
```



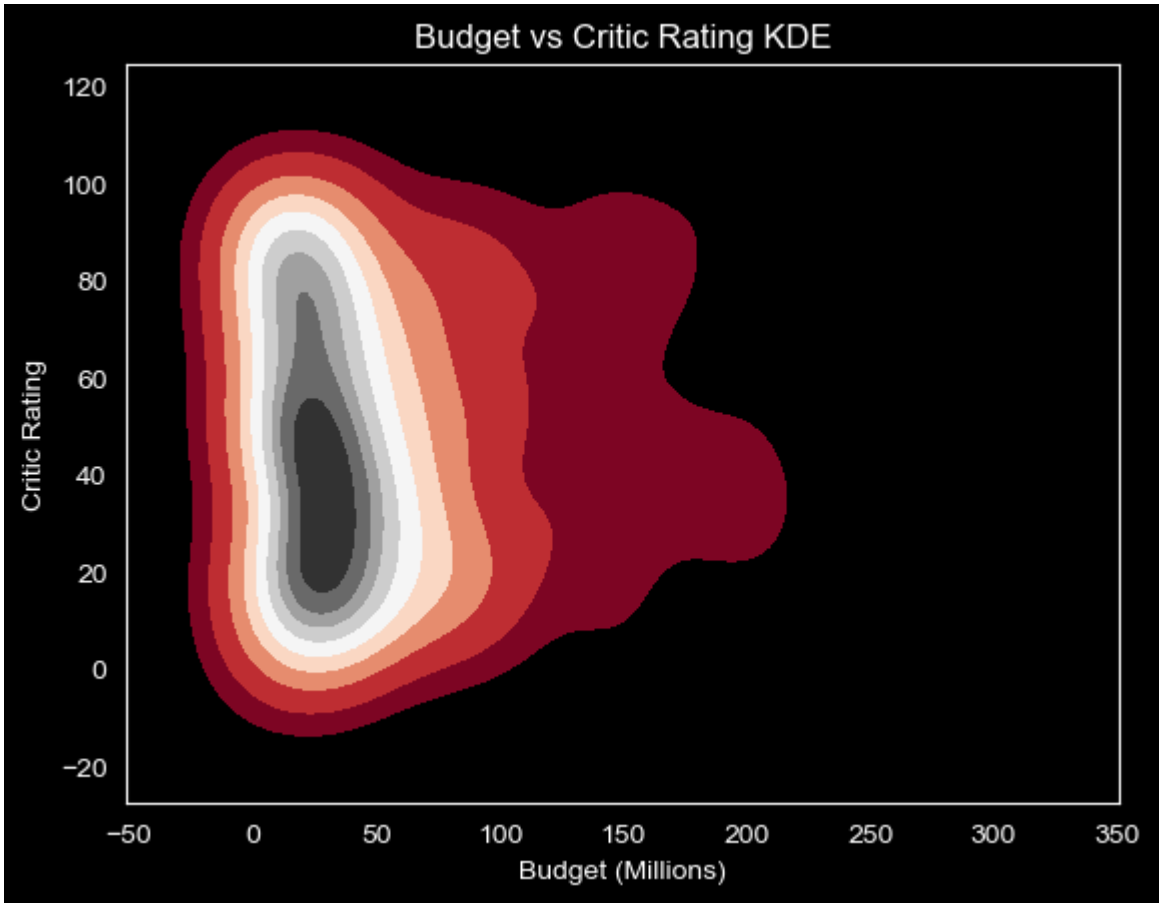

In [157…

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('white')
k2=sns.kdeplot(
    x=file.BudgetMillions,
    y=file.CriticRating
)
plt.show()
```



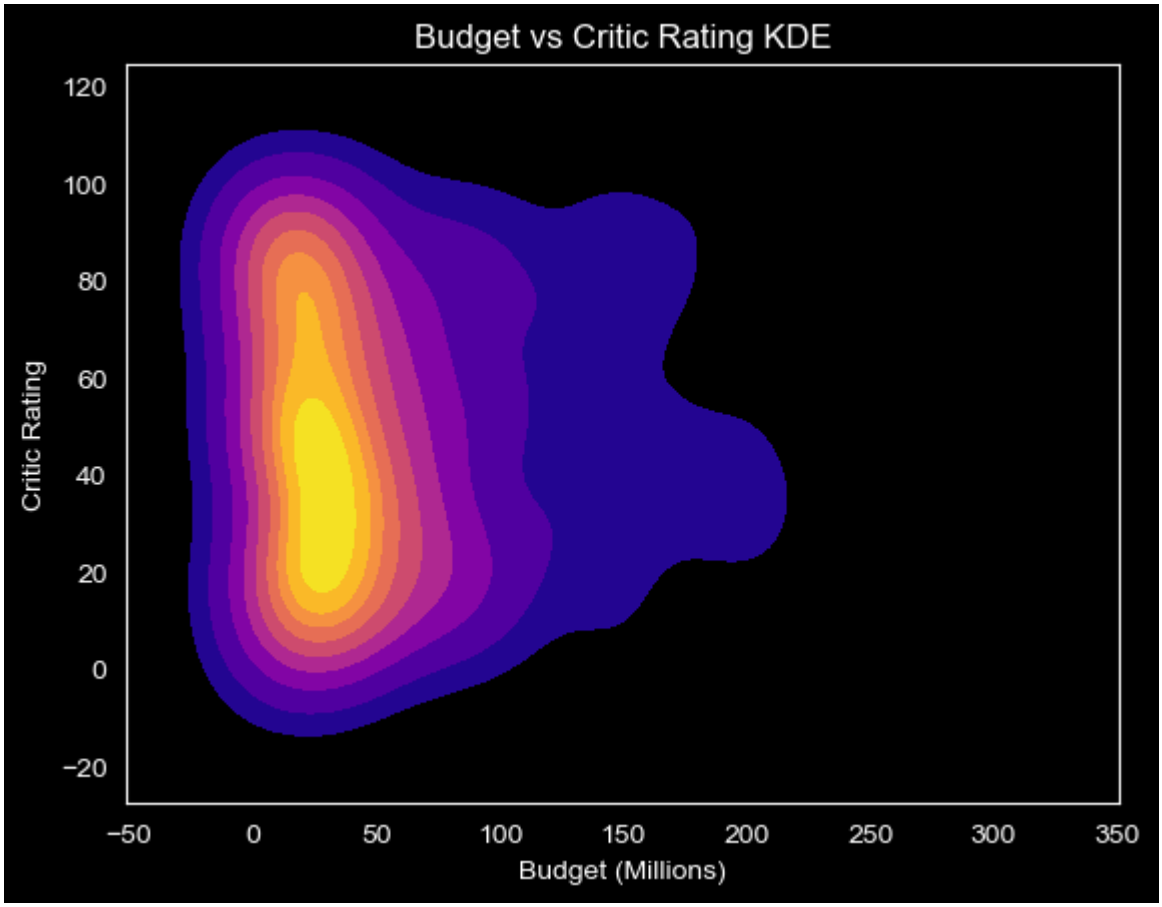

In [160…

```
sns.set_style('dark')
plt.style.use('dark_background')
# KDE plot with color
k2 = sns.kdeplot(
x=file.BudgetMillions,
y=file.CriticRating,
fill=True,
cmap=('RdGy'),
thresh=0.05
)
plt.title("Budget vs Critic Rating KDE", color='white')
plt.xlabel("Budget (Millions)", color='white')
plt.ylabel("Critic Rating", color='white')
plt.show()
```



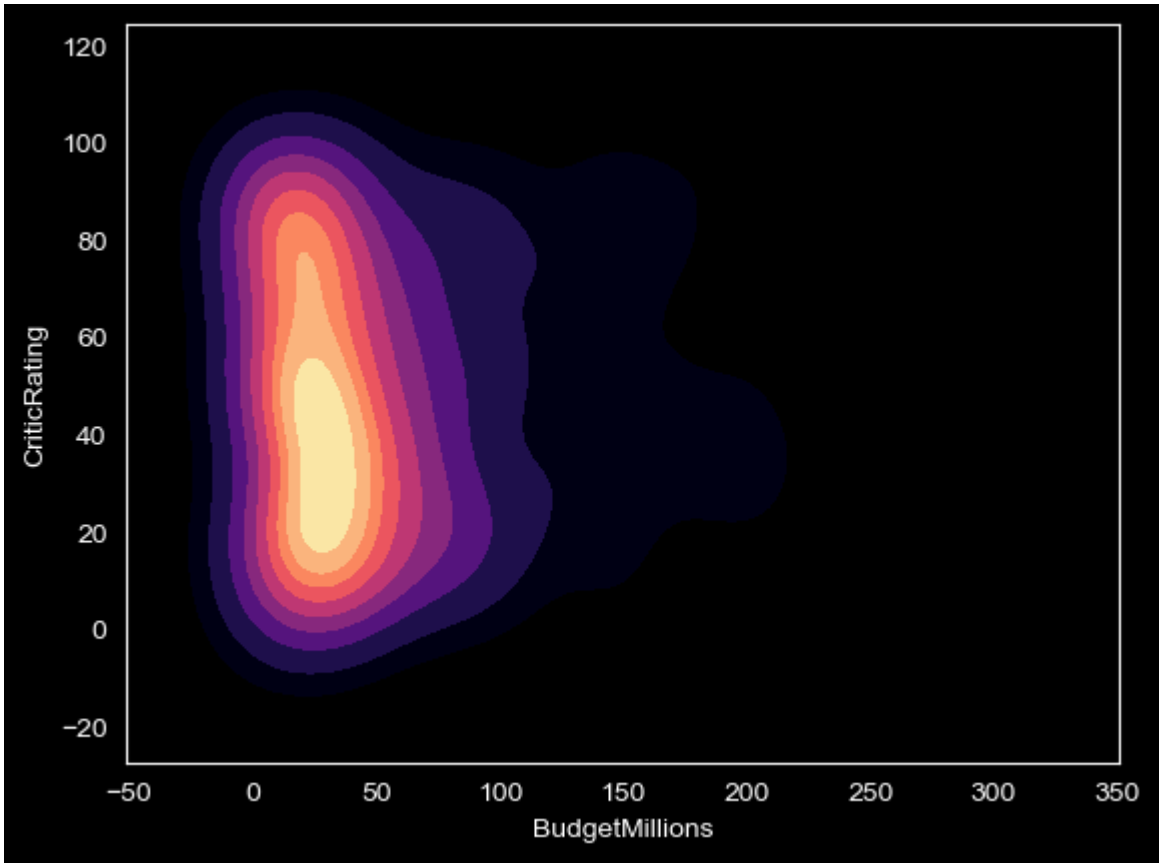

In [162…

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('dark')
plt.style.use('dark_background')
k2 = sns.kdeplot(
x=file.BudgetMillions,
y=file.CriticRating,
fill=True,
# Fill the area
cmap='plasma',
thresh=0.05,
levels=10,
linewidths=1.2,
color='white'
)
# Fill color (gradient)
# Number of contour levels (controls detail)
# Thickness of border lines
#
#✅Border/contour line color
plt.title("Budget vs Critic Rating KDE", color='white')
plt.xlabel("Budget (Millions)", color='white')
plt.ylabel("Critic Rating", color='white')
plt.show()
```



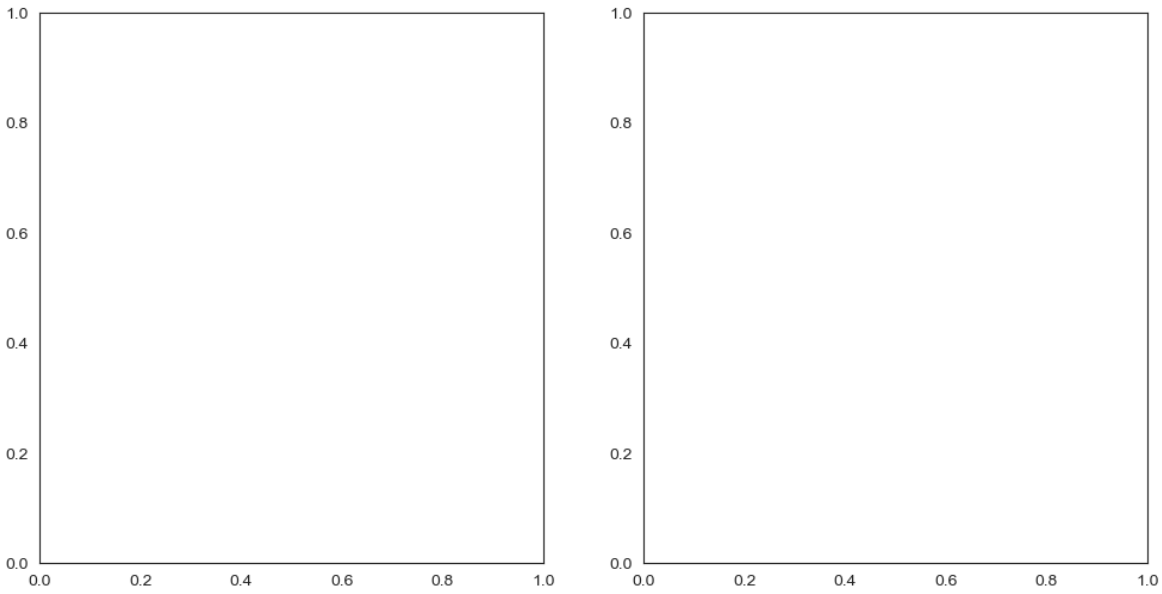

In [163…
```
sns.kdeplot(
x=file.BudgetMillions,
y=file.CriticRating,
fill=True,
cmap='magma',
thresh=0.05
)
```

Out[163… <Axes: xlabel='BudgetMillions', ylabel='CriticRating'>



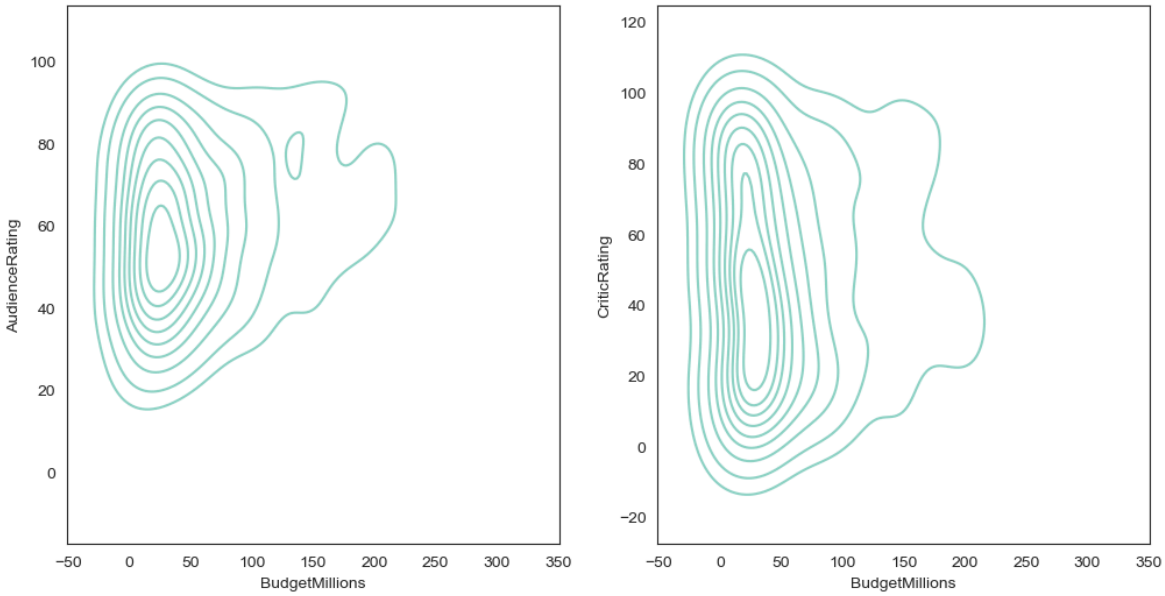

In [165…

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('white')
f, ax = plt.subplots(1,2, figsize =(12,6))
```



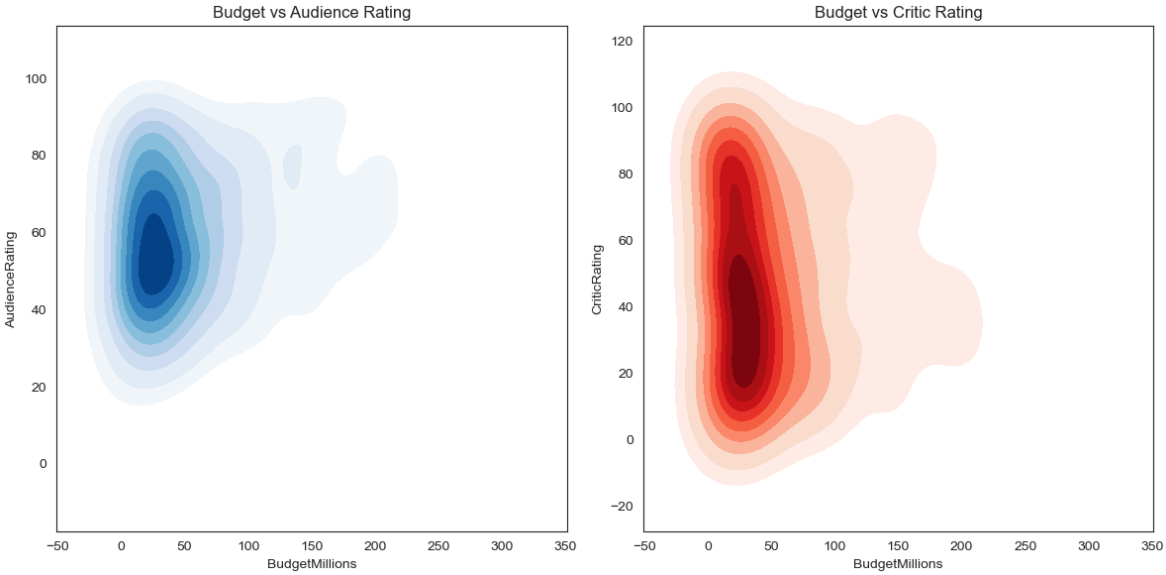

In [166…

```
f, axes = plt.subplots(1, 2, figsize=(12, 6))
k1 = sns.kdeplot(x=file.BudgetMillions,y=file.AudienceRating,ax = axes[0])
k2 = sns.kdeplot(x=file.BudgetMillions,y=file.CriticRating,ax = axes[1])
```




In [167…

```
import matplotlib.pyplot as plt
import seaborn as sns
f, axes = plt.subplots(1, 2, figsize=(12, 6))
k1 = sns.kdeplot(
x=file.BudgetMillions,
y=file.AudienceRating,
fill=True,
cmap='Blues',
ax=axes[0]
)
axes[0].set_title("Budget vs Audience Rating")
k2 = sns.kdeplot(
x=file.BudgetMillions,
y=file.CriticRating,
fill=True,
cmap='Reds',
ax=axes[1]
)
axes[1].set_title("Budget vs Critic Rating")
plt.tight_layout()
plt.show()
```

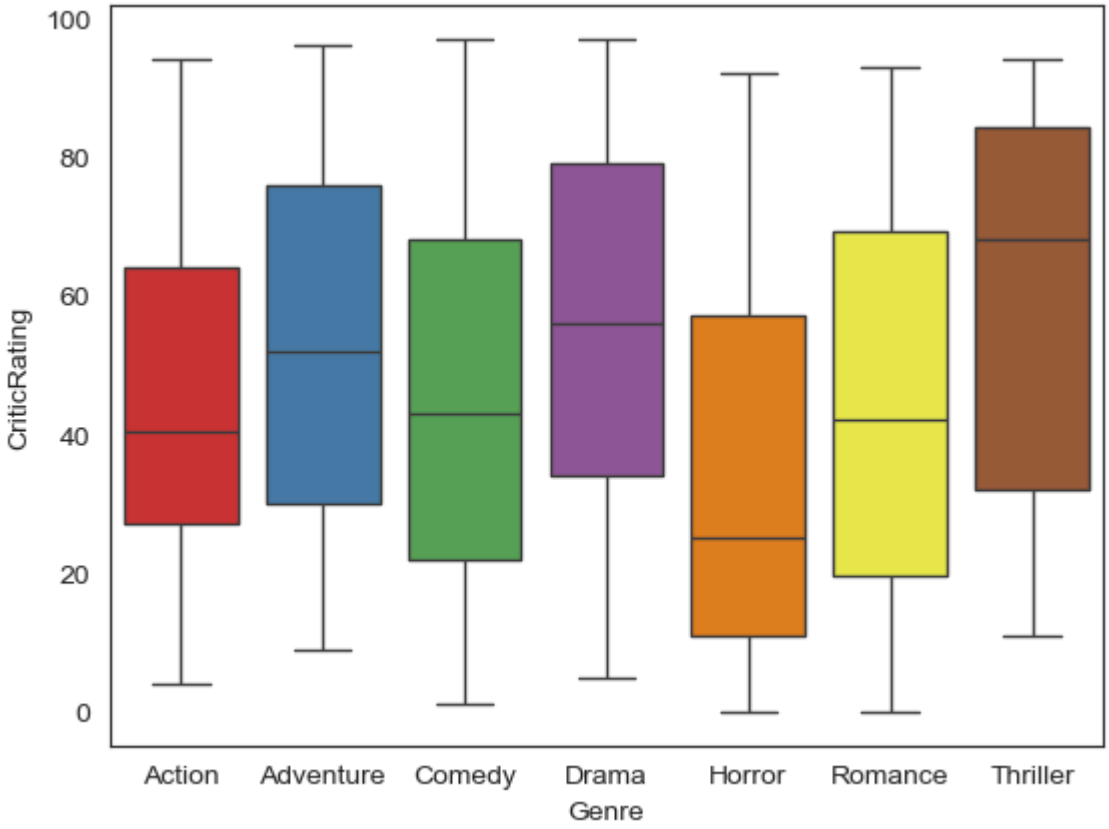



In [168…

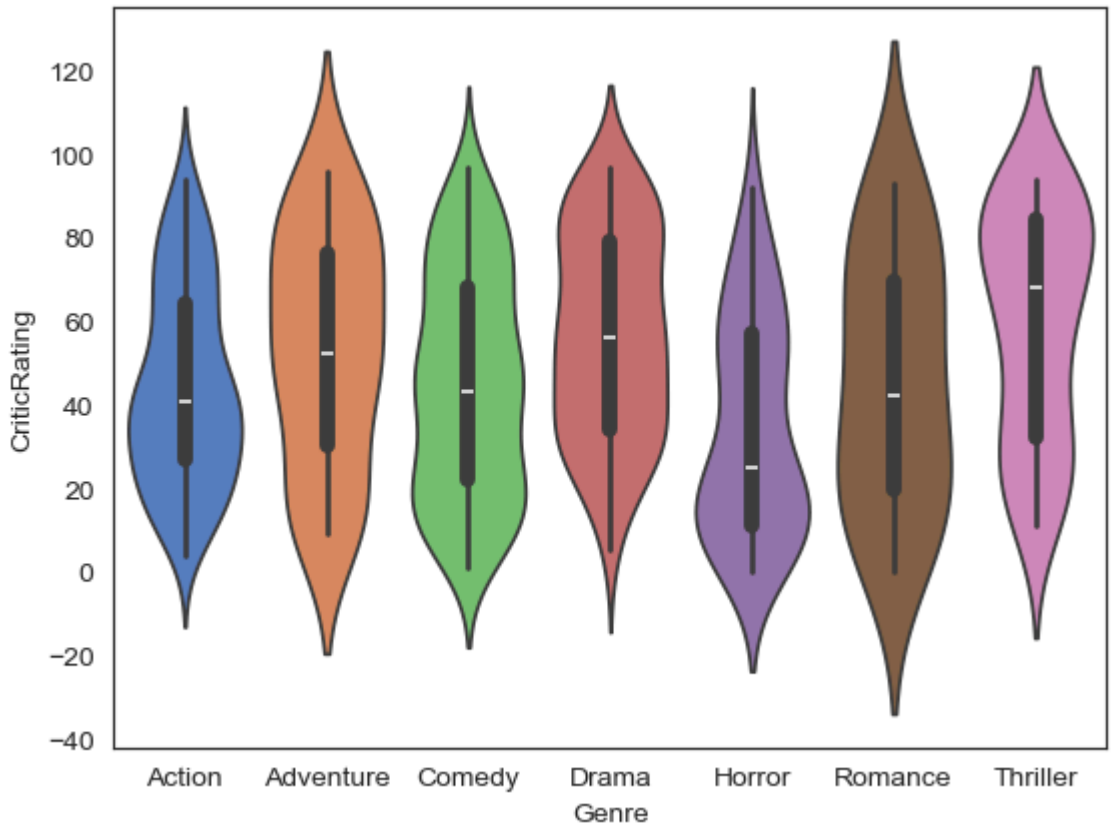
```
axes
```

Out[168…
```
array([<Axes: title={'center': 'Budget vs Audience Rating'}, xlabel='BudgetMill
ions', ylabel='AudienceRating'>,
       <Axes: title={'center': 'Budget vs Critic Rating'}, xlabel='BudgetMillio
ns', ylabel='CriticRating'>],
      dtype=object)
```

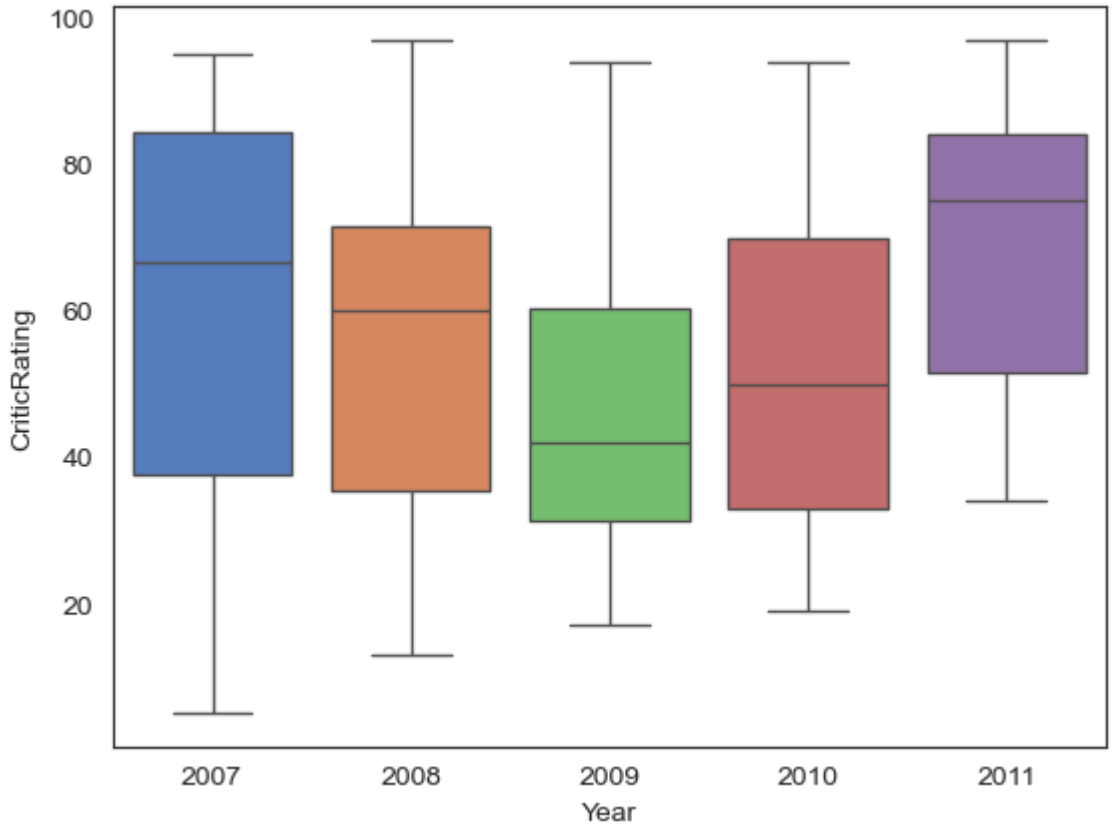
In [169…
```
w = sns.boxplot(data=file, x='Genre', y = 'CriticRating',palette='Set1')
```



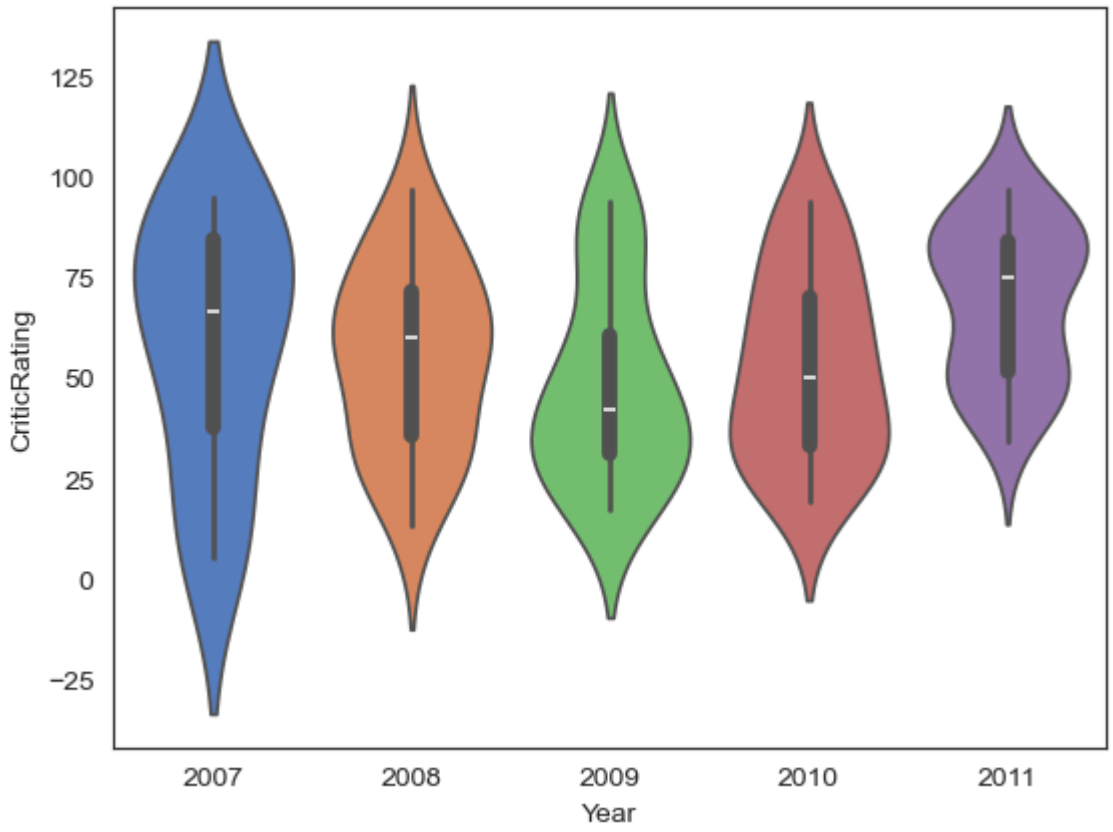

In [170…
```
z = sns.violinplot(data=file, x='Genre', y = 'CriticRating',palette='muted')
```




In [172…
```
w1 = sns.boxplot(data=file[file.Genre == 'Drama'], x='Year', y = 'CriticRating',
```

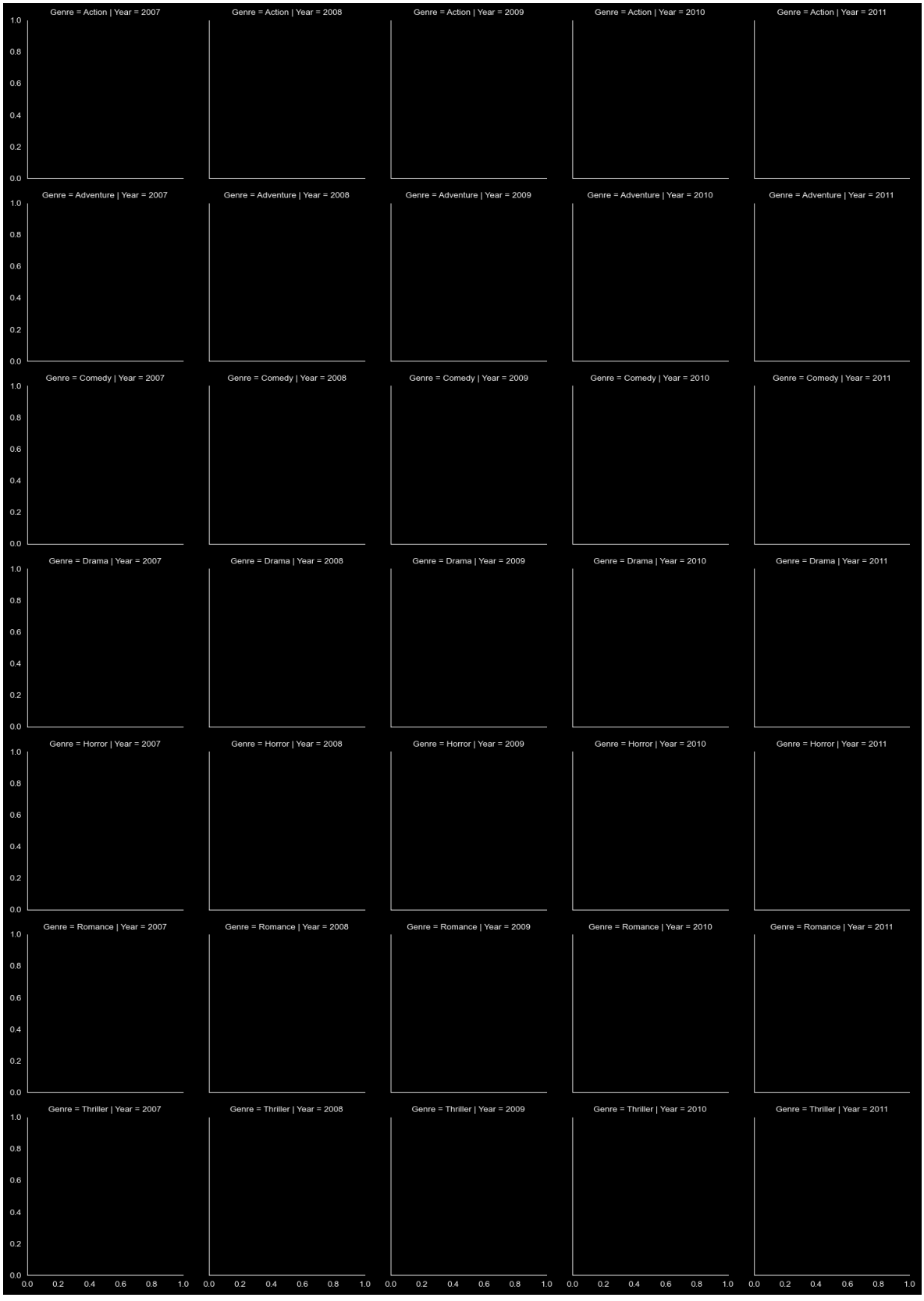



In [173…
```
z = sns.violinplot(data=file[file.Genre == 'Drama'], x='Year', y='CriticRating',
```




In [175…

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('dark')
plt.style.use('dark_background')
g =sns.FacetGrid (file, row = 'Genre', col = 'Year', hue = 'Genre')
```

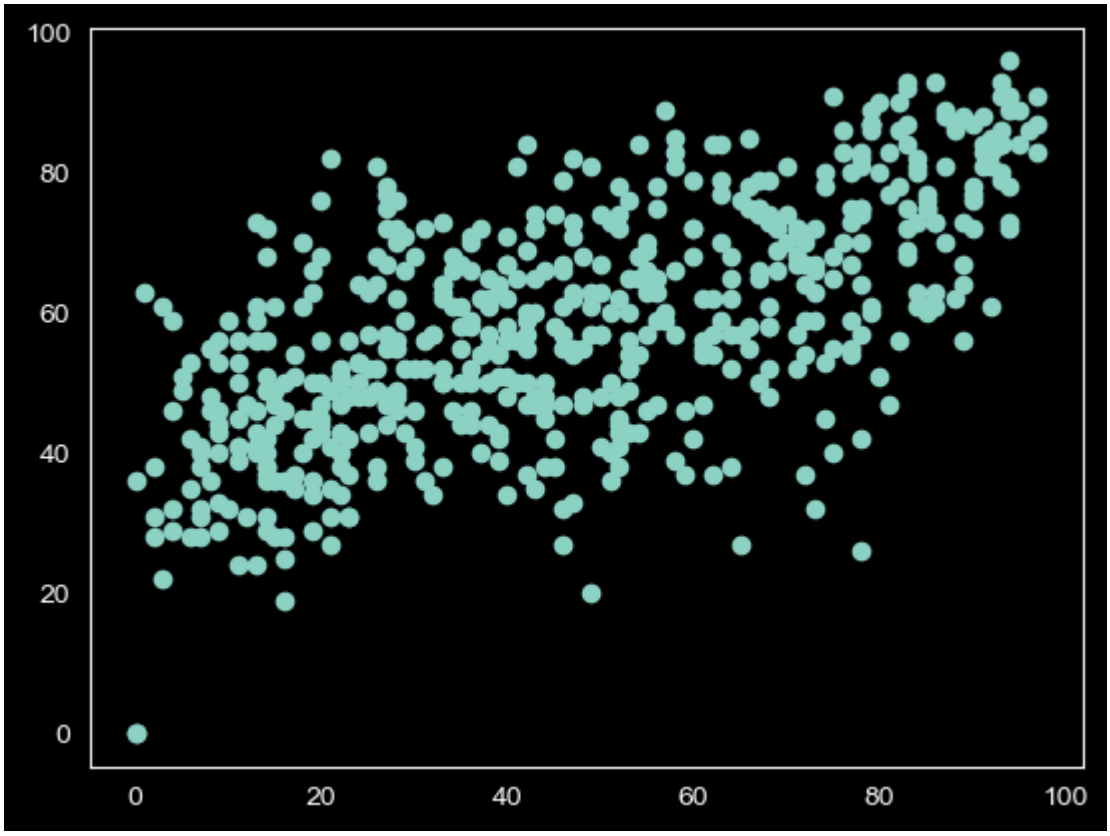



In [176…

```
sns.set_style('dark')
plt.style.use('dark_background')
plt.scatter(file.CriticRating,file.AudienceRating)
```

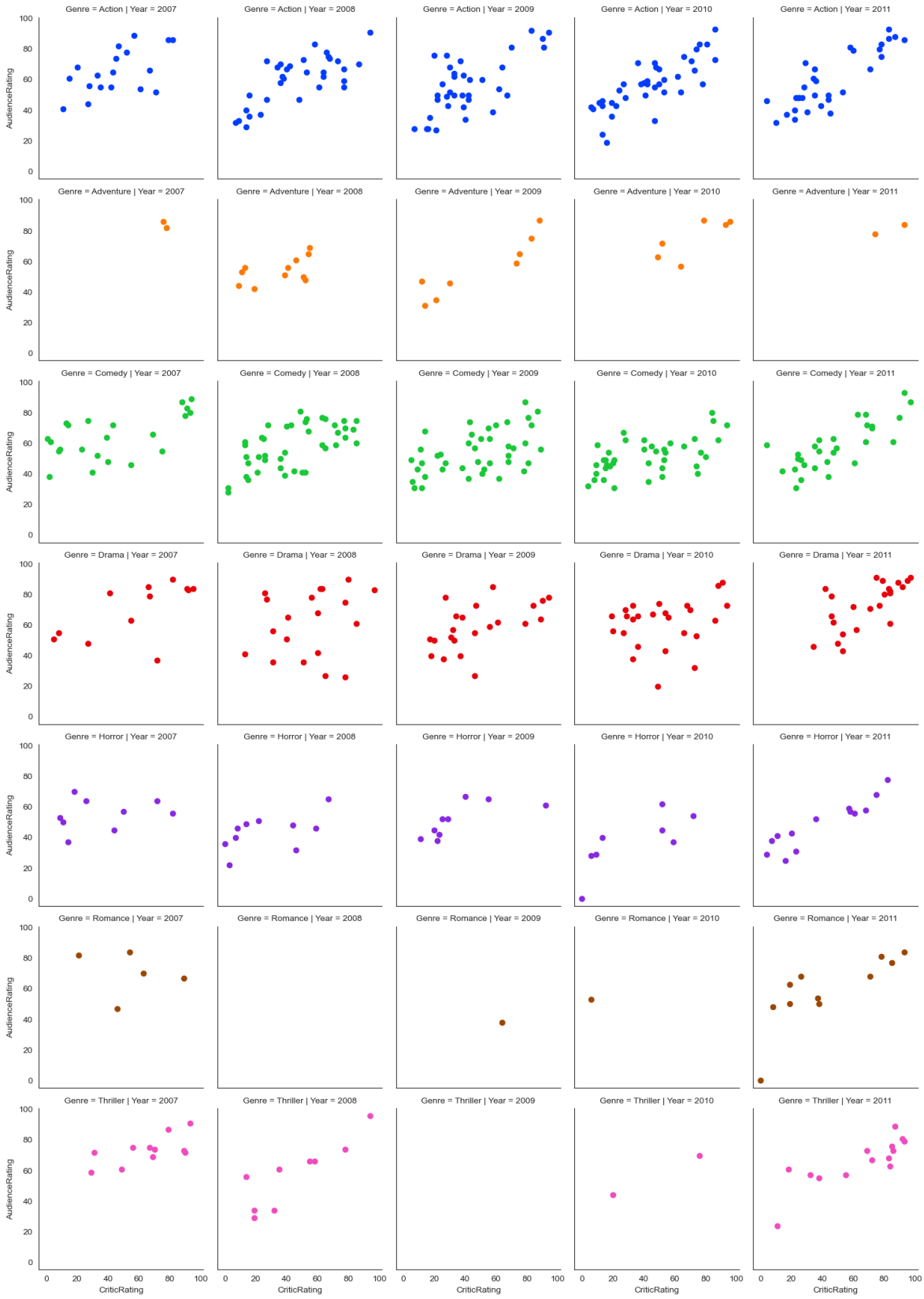
Out[176…

```
<matplotlib.collections.PathCollection at 0x205b7131f40>
```



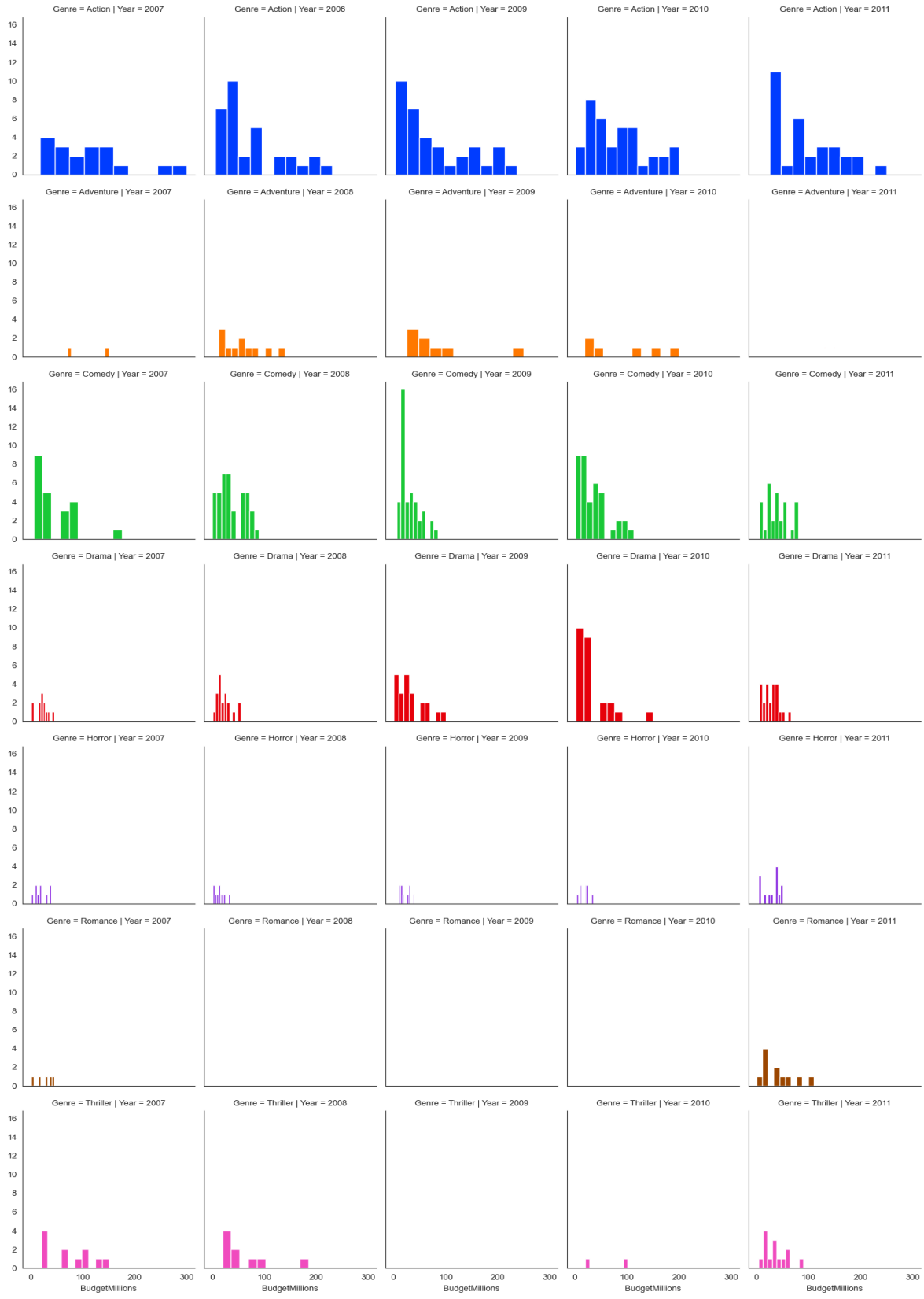

In [179…

```
sns.set_style('white')
g =sns.FacetGrid (file, row = 'Genre', col = 'Year', hue = 'Genre',palette='brig
g = g.map(plt.scatter, 'CriticRating', 'AudienceRating')
```



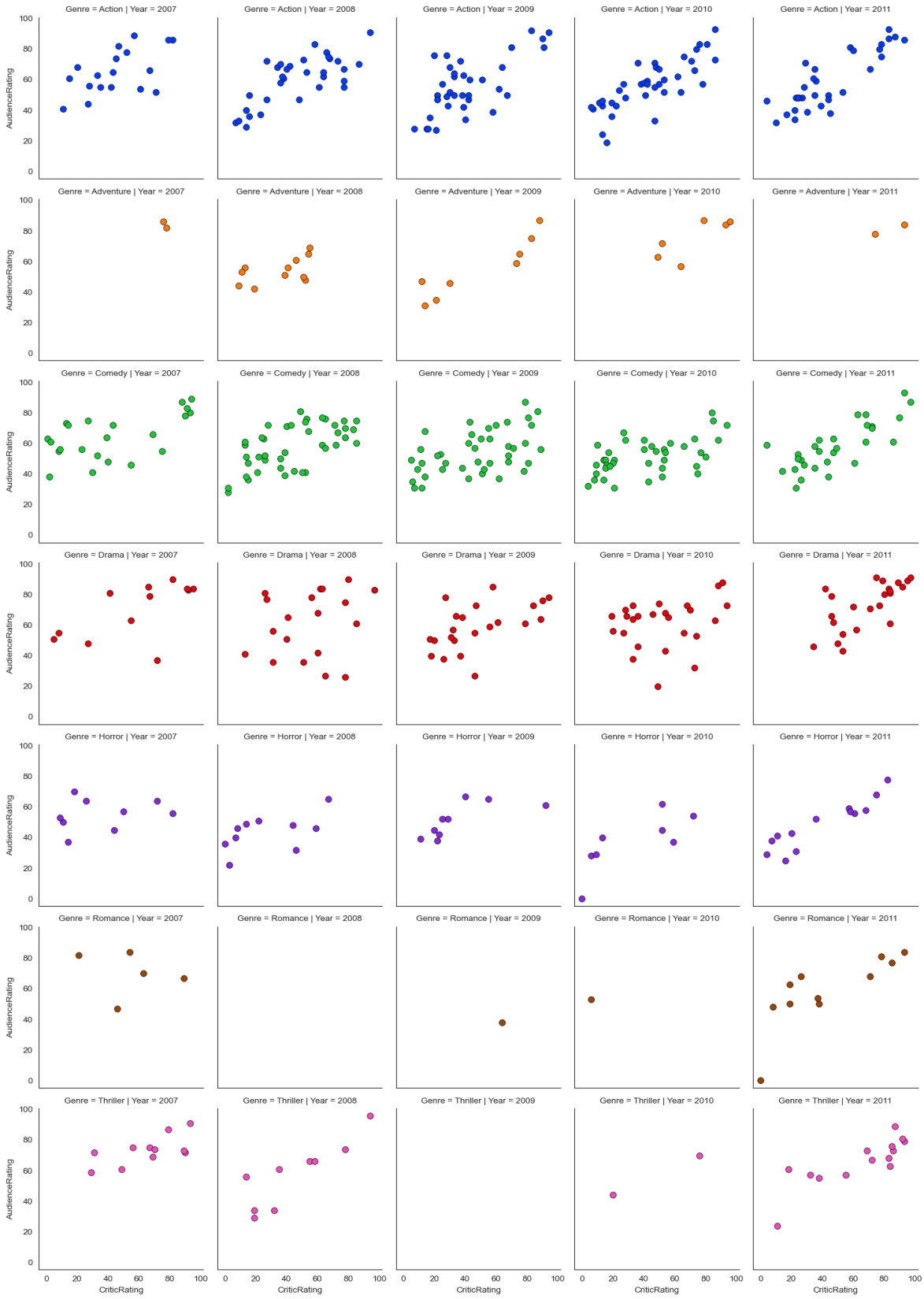

In [181…

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



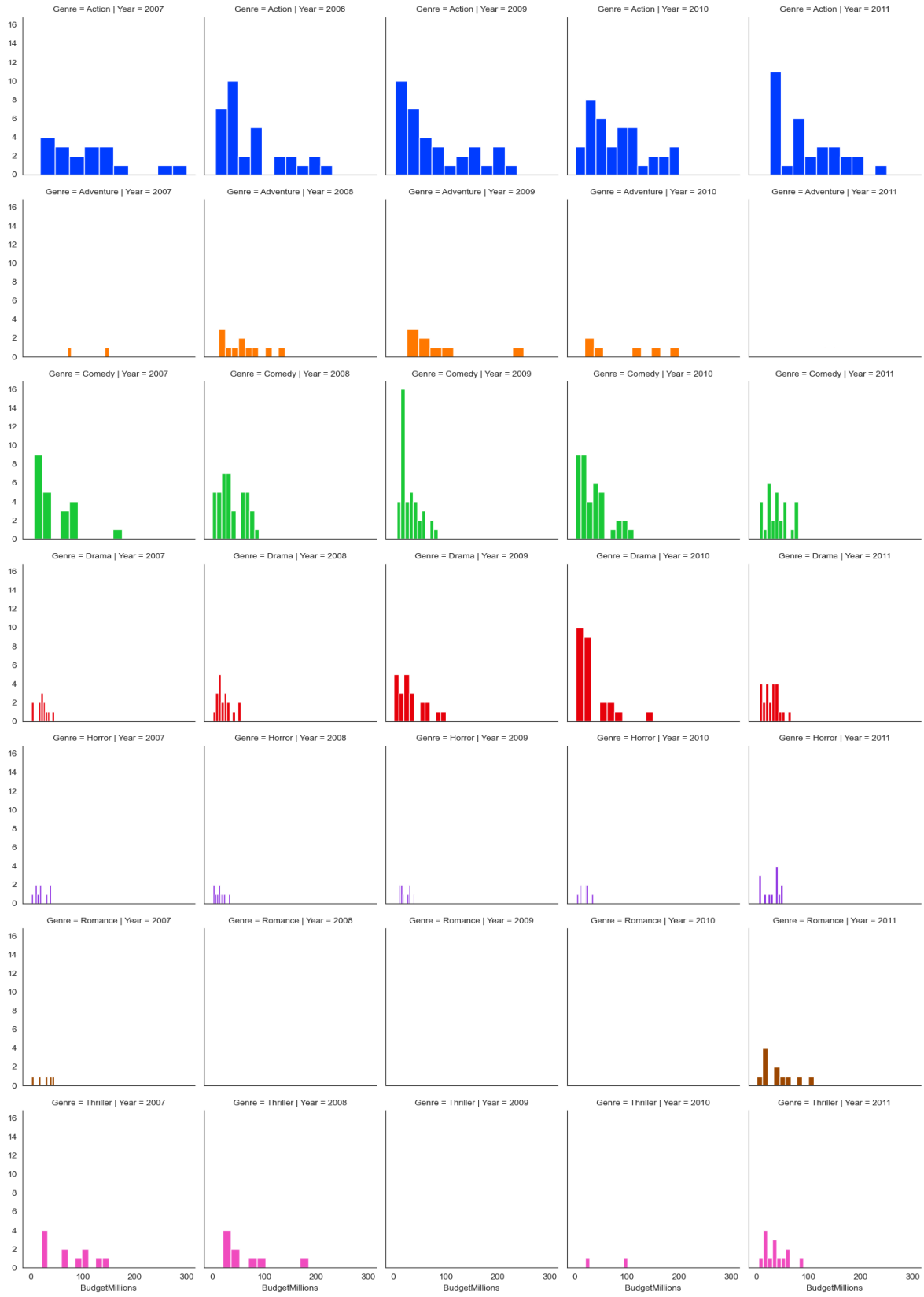

In [183…

```
sns.set_style('white')
g =sns.FacetGrid (file, row = 'Genre', col = 'Year', hue = 'Genre',palette='brig
kws = dict(s=50, linewidth=0.5,edgecolor='black')
g = g.map(plt.scatter, 'CriticRating', 'AudienceRating',**kws )
```



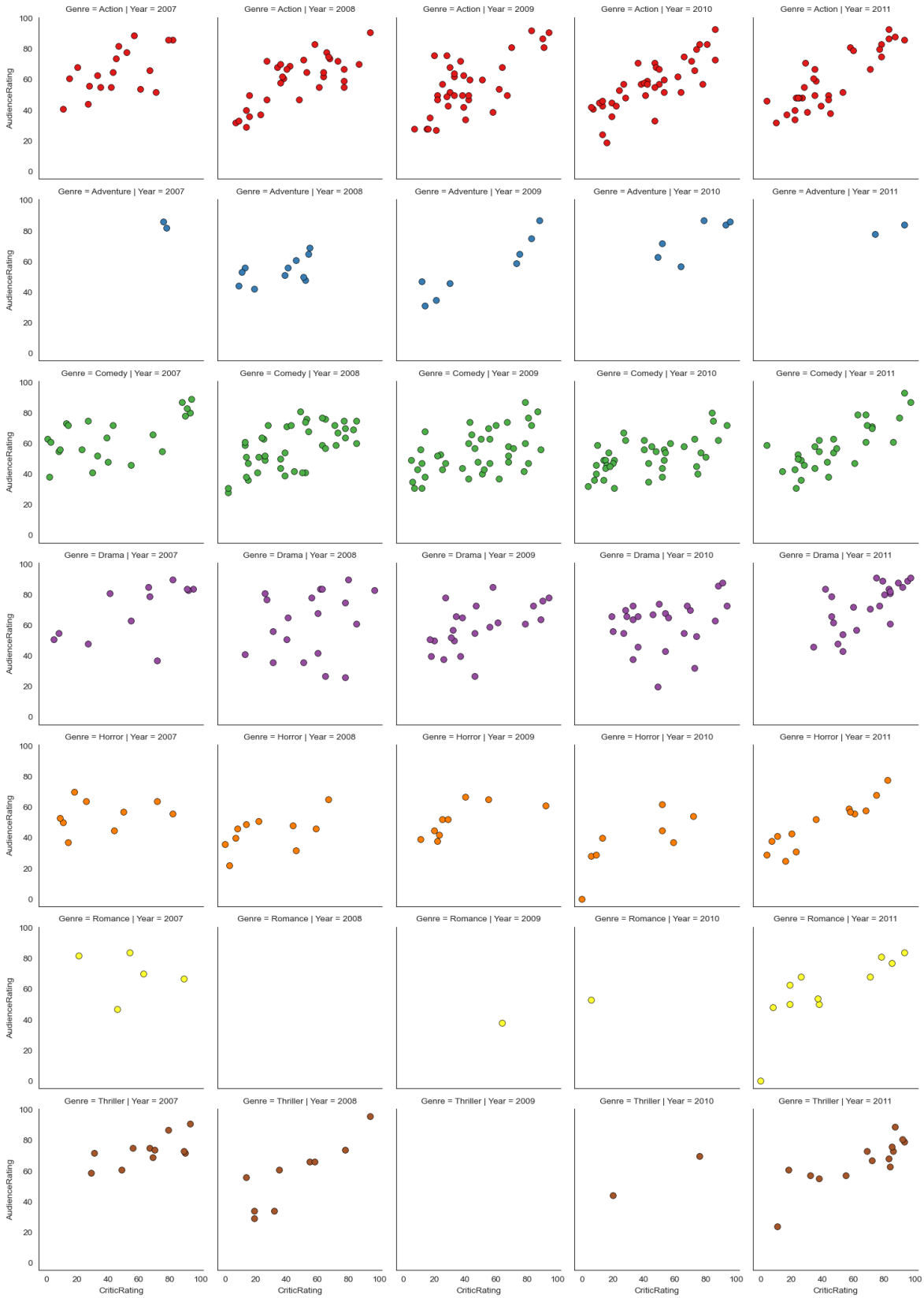

In [184…

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




In [185…

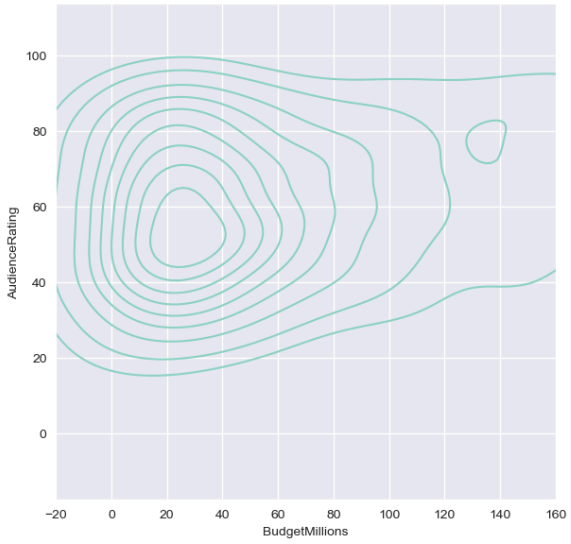
```
g =sns.FacetGrid (file, row = 'Genre', col = 'Year', hue = 'Genre',palette='Set1
kws = dict(s=50, linewidth=0.5,edgecolor='black')
g = g.map(plt.scatter, 'CriticRating', 'AudienceRating',**kws)
```

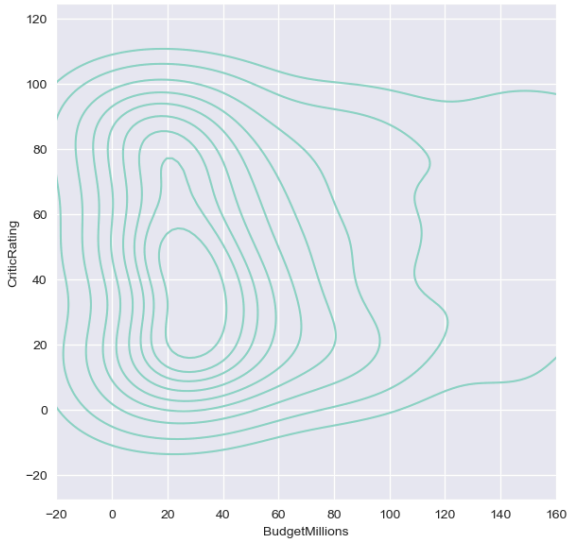


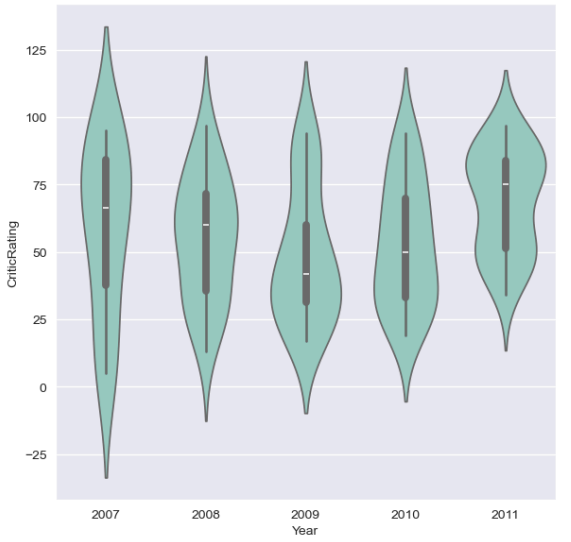

In [186…

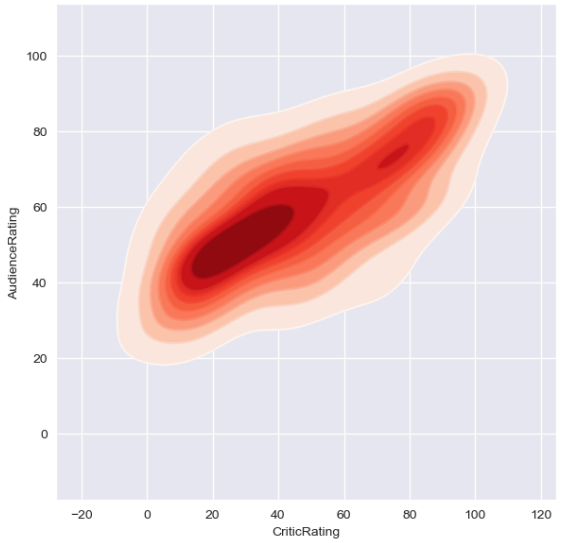
```
sns.set_style('darkgrid')
f, axes = plt.subplots(2, 2, figsize=(15, 15))

# 2D KDE plots
k1 = sns.kdeplot(x=file.BudgetMillions, y=file.AudienceRating, ax=axes[0, 0])
k2 = sns.kdeplot(x=file.BudgetMillions, y=file.CriticRating, ax=axes[0, 1])

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

```
# Violin plot
z = sns.violinplot(data=file[file.Genre == 'Drama'], x='Year', y='CriticRating',

# 2D KDE with fill (replaces deprecated `shade=True`)
k4 = sns.kdeplot(
    x=file.CriticRating, y=file.AudienceRating,
    fill=True, cmap='Reds', ax=axes[1, 1]
)

# Overlaid KDE without fill
k4b = sns.kdeplot(
    x=file.CriticRating, y=file.AudienceRating,
    fill=False, cmap='Reds', ax=axes[1, 1]
)

plt.show()
```



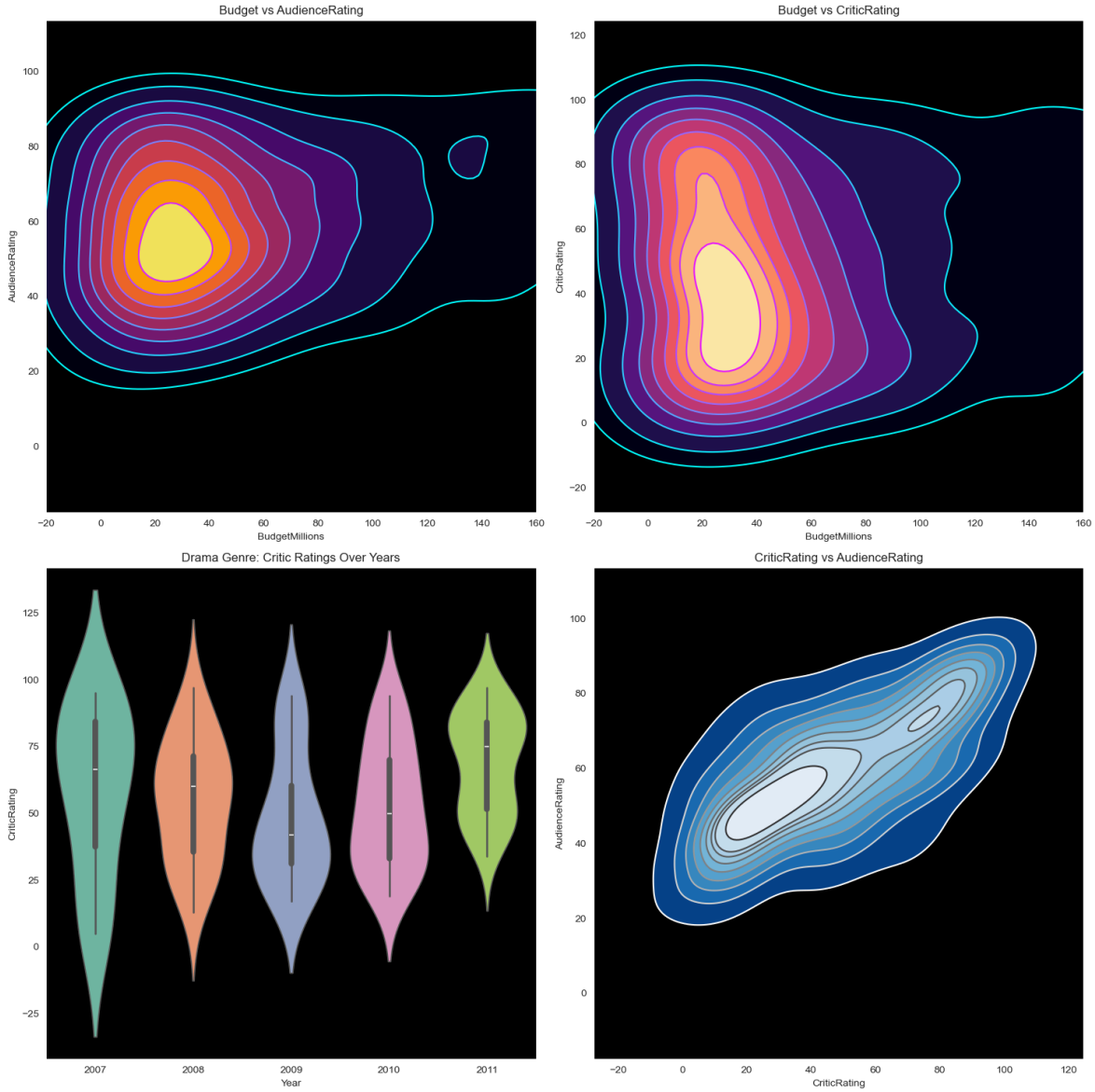




In [187…

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

# Create subplot layout (2x2)
f, axes = plt.subplots(2, 2, figsize=(15, 15))

# KDE plot 1: Budget vs AudienceRating
sns.kdeplot(data=file, x='BudgetMillions', y='AudienceRating',
```

```
            fill=True, cmap='inferno', ax=axes[0, 0], thresh=0.05)
sns.kdeplot(data=file, x='BudgetMillions', y='AudienceRating',
            cmap='cool', ax=axes[0, 0])

# KDE plot 2: Budget vs CriticRating
sns.kdeplot(data=file, x='BudgetMillions', y='CriticRating',
            fill=True, cmap='magma', ax=axes[0, 1], thresh=0.05)
sns.kdeplot(data=file, x='BudgetMillions', y='CriticRating',
            cmap='cool', ax=axes[0, 1])

# Violin plot: Drama Genre Ratings over Years
sns.violinplot(data=file[file.Genre == 'Drama'],
               x='Year', y='CriticRating', palette='Set2', ax=axes[1, 0])

# KDE plot 3: CriticRating vs AudienceRating
sns.kdeplot(data=file, x='CriticRating', y='AudienceRating',
            fill=True, cmap='Blues_r', ax=axes[1, 1], thresh=0.05)
sns.kdeplot(data=file, x='CriticRating', y='AudienceRating',
            cmap='gist_gray_r', ax=axes[1, 1])

# Set common x-limits for top plots
axes[0, 0].set_xlim(-20, 160)
axes[0, 1].set_xlim(-20, 160)

# Optional: add titles to each subplot
axes[0, 0].set_title("Budget vs AudienceRating")
axes[0, 1].set_title("Budget vs CriticRating")
axes[1, 0].set_title("Drama Genre: Critic Ratings Over Years")
axes[1, 1].set_title("CriticRating vs AudienceRating")

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


Budget vs AudienceRating
AudienceRating
BudgetMillions
Budget vs CriticRating
CriticRating
BudgetMillions
Drama Genre: Critic Ratings Over Years
CriticRating
Year
CriticRating vs AudienceRating
AudienceRating
CriticRating

In [ ]: