Movies_Rating_2007 to 2011

July 5, 2023

```
[124]: import matplotlib.pylab as plt
       import pandas as pd
       import os
       %matplotlib inline
```

```
[125]: print(os.getcwd())
```

```
C:\Users\ddaya\OneDrive\Documents\Python programming
```

```
[126]: os.chdir("C:\\Users\ddaya\OneDrive\Documents\Python programming")
```

```
[127]: print(os.getcwd())
```

```
C:\Users\ddaya\OneDrive\Documents\Python programming
```

```
[128]: movies=pd.read_csv('Movie-Ratings.csv')
```

```
[129]: movies
```

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

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

```
554                  36                  50             2011
555                  52                  18             2009
556                  73                  65             2007
557                  87                  24             2009
558                  42                  80             2011

[559 rows x 6 columns]
```

[130]:

```
len(movies)
```

[130]:

```
559
```

[131]:

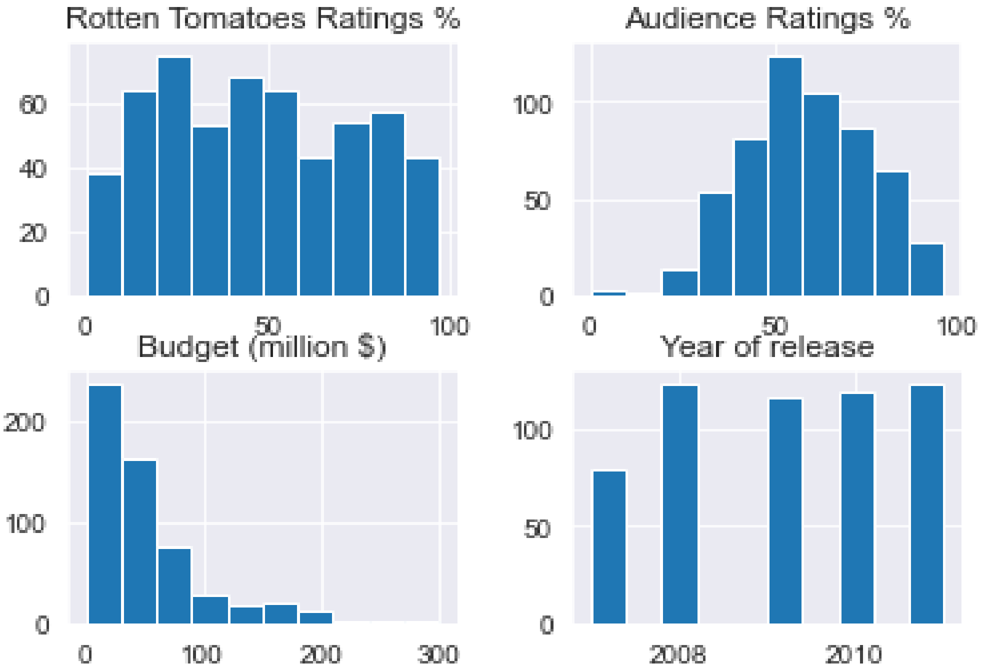
```
movies.head()
```

[131]:

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

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

[132]:

```
movies.hist()
plt.show()
plt.rcParams['figure.figsize']=30,5
```



[133]:
```
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
```

[134]:
```
movies.describe()
```

[134]:
```
       Rotten Tomatoes Ratings %  Audience Ratings %  Budget (million $)  \
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

       Year of release
count       559.000000
mean       2009.152057
std           1.362632
min        2007.000000
25%        2008.000000
50%        2009.000000
75%        2010.000000
max        2011.000000
```

[135]:
```
movies.describe().transpose()
```

[135]:
```
                           count         mean        std     min     25%  \
Rotten Tomatoes Ratings %  559.0    47.309481  26.413091     0.0    25.0
Audience Ratings %         559.0    58.744186  16.826887     0.0    47.0
Budget (million $)         559.0    50.236136  48.731817     0.0    20.0
Year of release            559.0  2009.152057   1.362632  2007.0  2008.0

                              50%     75%     max
Rotten Tomatoes Ratings %    46.0    70.0    97.0
Audience Ratings %           58.0    72.0    96.0
Budget (million $)           35.0    65.0   300.0
Year of release            2009.0  2010.0  2011.0
```

[136]:
```
movies.Film.astype('category').head()
```

[136]:
```
0    (500) Days of Summer
1             10,000 B.C.
2               12 Rounds
3               127 Hours
4                17 Again
Name: Film, dtype: category
Categories (559, object): ['(500) Days of Summer', '10,000 B.C.', '12 Rounds',
'127 Hours', …, 'Youth in Revolt', 'Zodiac', 'Zombieland', 'Zookeeper']
```

[137]:
```
movies
```

[137]:
```
                     Film      Genre  Rotten Tomatoes Ratings %  \
0    (500) Days of Summer     Comedy                         87
1             10,000 B.C.  Adventure                          9
2               12 Rounds     Action                         30
3               127 Hours  Adventure                         93
4                17 Again     Comedy                         55
..                    …         …                        …
```

```
554         Your Highness     Comedy                         26
555       Youth in Revolt     Comedy                         68
556                Zodiac   Thriller                         89
557            Zombieland     Action                         90
558             Zookeeper     Comedy                         14

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

[559 rows x 6 columns]
```

[138]:

```
movies.columns
```

[138]:

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

[139]:

```
movies.columns=['Film', 'Genre', 'Rotten_Tomatoes_Ratings', 'Audience_Ratings',
       'Budget', 'Year']
```

[140]:

```
movies
```

[140]:

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

     Audience_Ratings  Budget  Year
0                  81       8  2009
```

```
1                    44     105   2008
2                    52      20   2009
3                    84      18   2010
4                    70      20   2009
..                  …       …    …
554                  36      50   2011
555                  52      18   2009
556                  73      65   2007
557                  87      24   2009
558                  42      80   2011

[559 rows x 6 columns]
```

[141]:
```
movies.Genre=movies.Genre.astype('category')
movies.Year=movies.Year.astype('category')
```

[142]:
```
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    category
 2   Rotten_Tomatoes_Ratings 559 non-null    int64
 3   Audience_Ratings        559 non-null    int64
 4   Budget                  559 non-null    int64
 5   Year                    559 non-null    category
dtypes: category(2), int64(3), object(1)
memory usage: 19.2+ KB
```

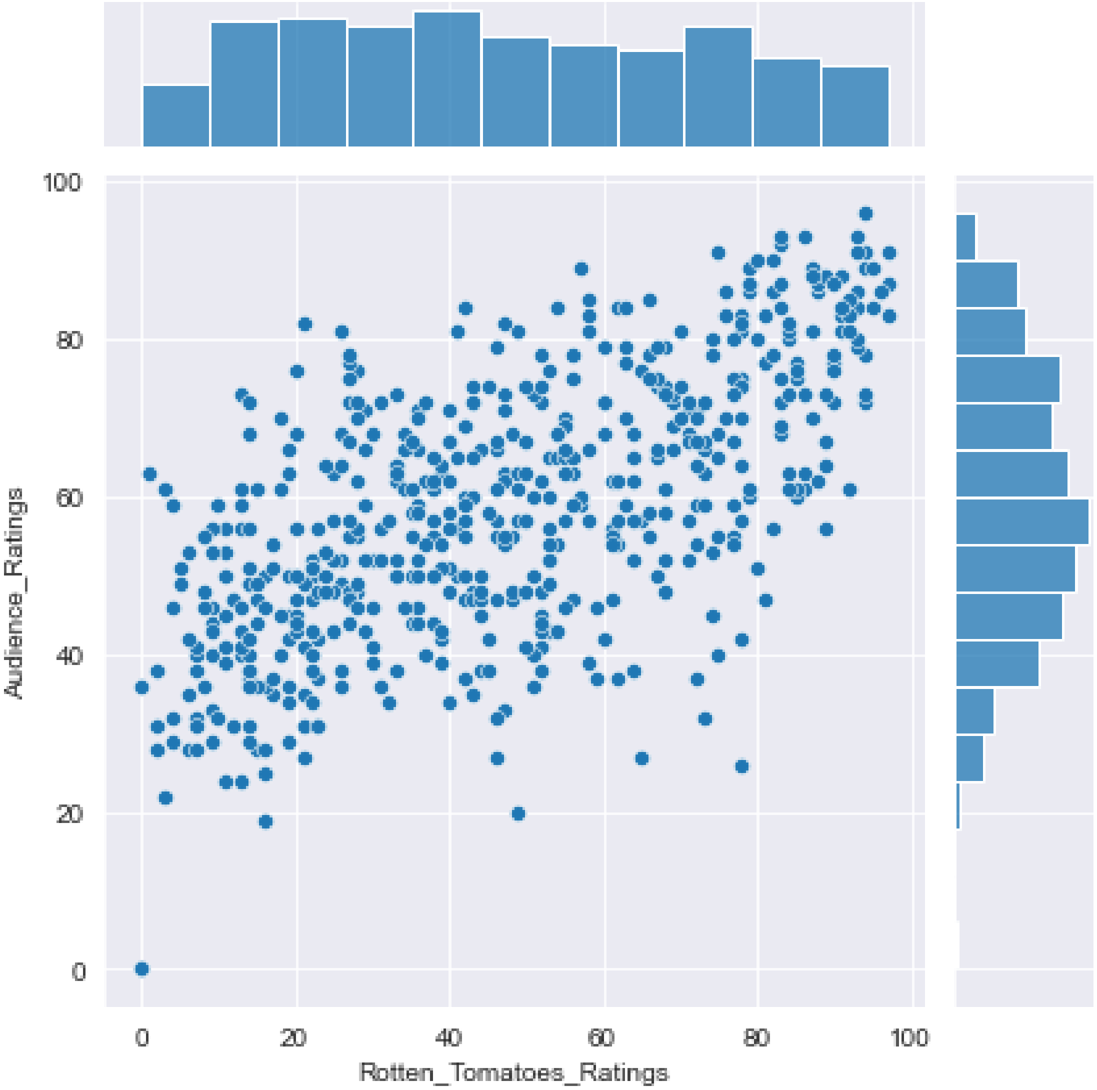
[143]:
```
movies.describe()
```

[143]:
```
       Rotten_Tomatoes_Ratings  Audience_Ratings      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
```

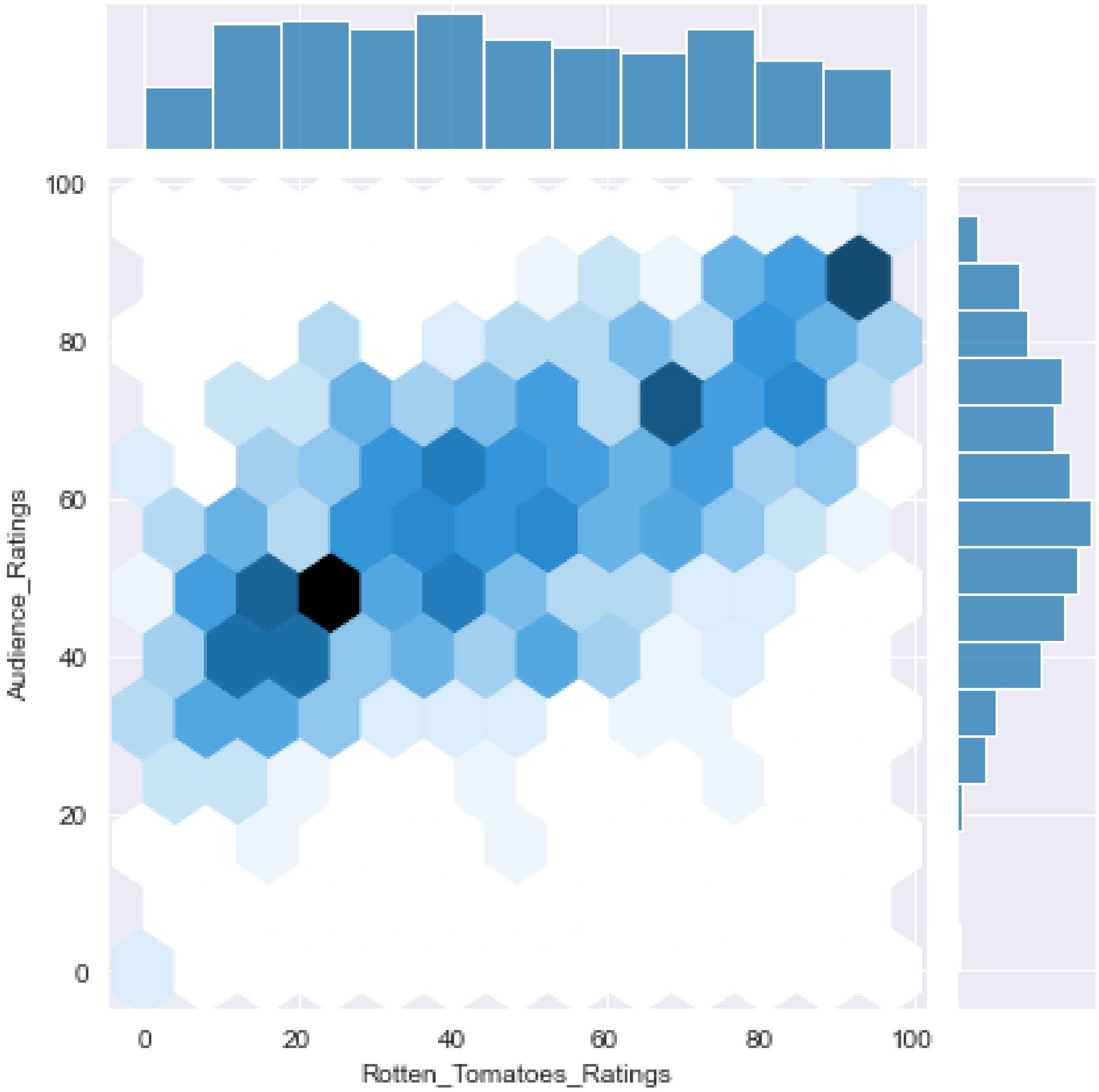
[144]:
```
import matplotlib as plt
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
import warnings
warnings.filterwarnings('ignore')
plt.rcParams['figure.figsize']=10,8
```

```
[145]: j=sns.jointplot(data=movies,x='Rotten_Tomatoes_Ratings',y='Audience_Ratings')
       plt.show()
```

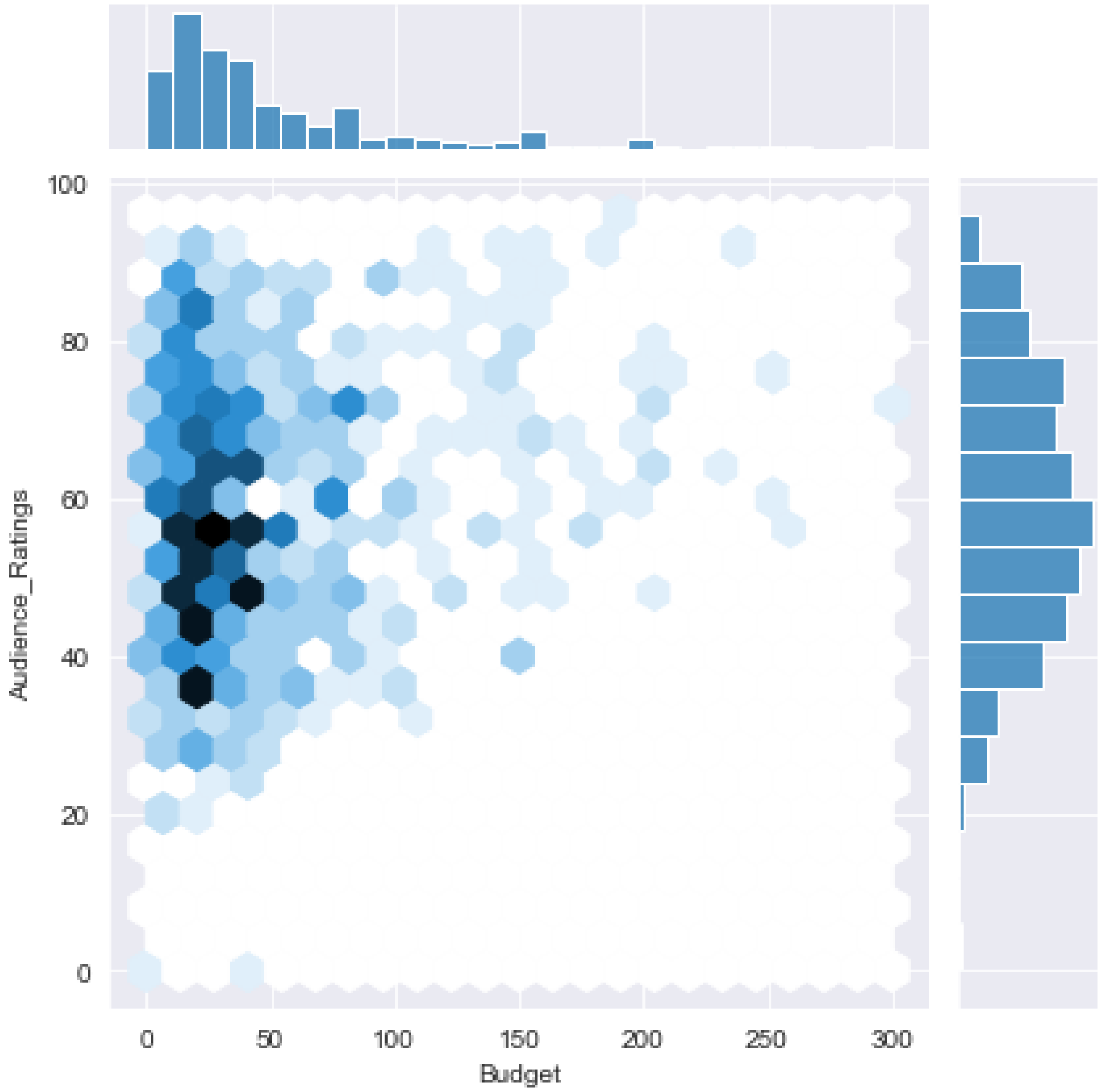


```
[146]: sns.
        ↪jointplot(data=movies,x='Rotten_Tomatoes_Ratings',y='Audience_Ratings',kind='hex')
       plt.show()
```



[147]:

```
sns.jointplot(data=movies,x="Budget",y="Audience_Ratings",kind="hex")
plt.show()
```



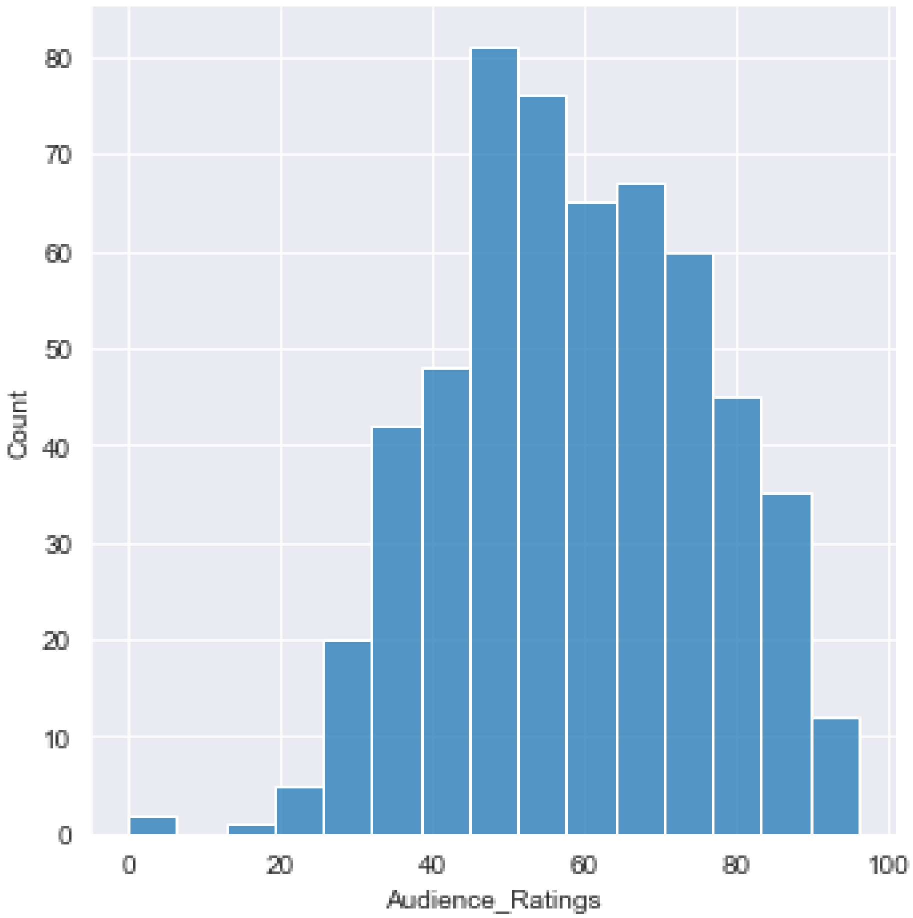
```
[148]: movies
```

```
[148]:                     Film      Genre  Rotten_Tomatoes_Ratings  \
       0    (500) Days of Summer     Comedy                       87
       1              10,000 B.C.  Adventure                        9
       2               12 Rounds     Action                       30
       3               127 Hours  Adventure                       93
       4                17 Again     Comedy                       55
       ..                    …          …                        …
       554          Your Highness     Comedy                       26
       555        Youth in Revolt     Comedy                       68
       556                 Zodiac   Thriller                       89
       557             Zombieland     Action                       90
```

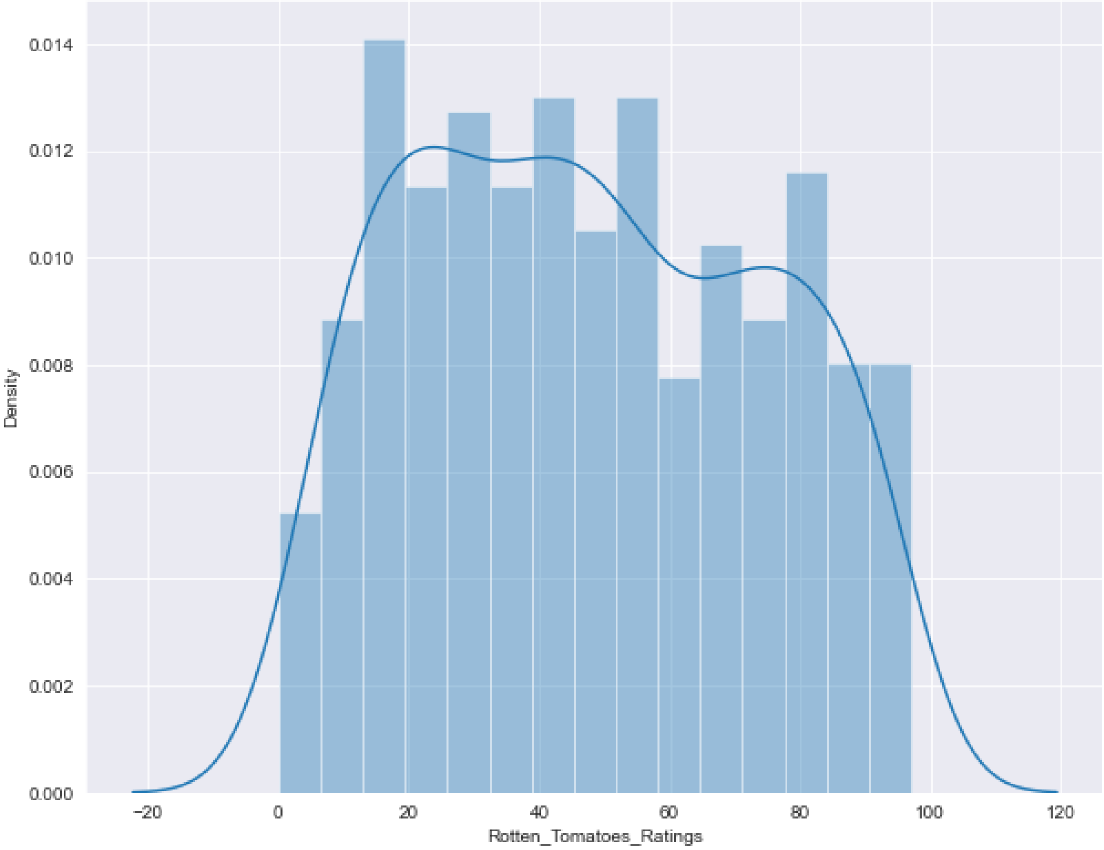
```
558               Zookeeper     Comedy                          14

     Audience_Ratings  Budget  Year
0                  81       8  2009
1                  44     105  2008
2                  52      20  2009
3                  84      18  2010
4                  70      20  2009
..                ...     ...   ...
554                36      50  2011
555                52      18  2009
556                73      65  2007
557                87      24  2009
558                42      80  2011

[559 rows x 6 columns]
```

[149]:

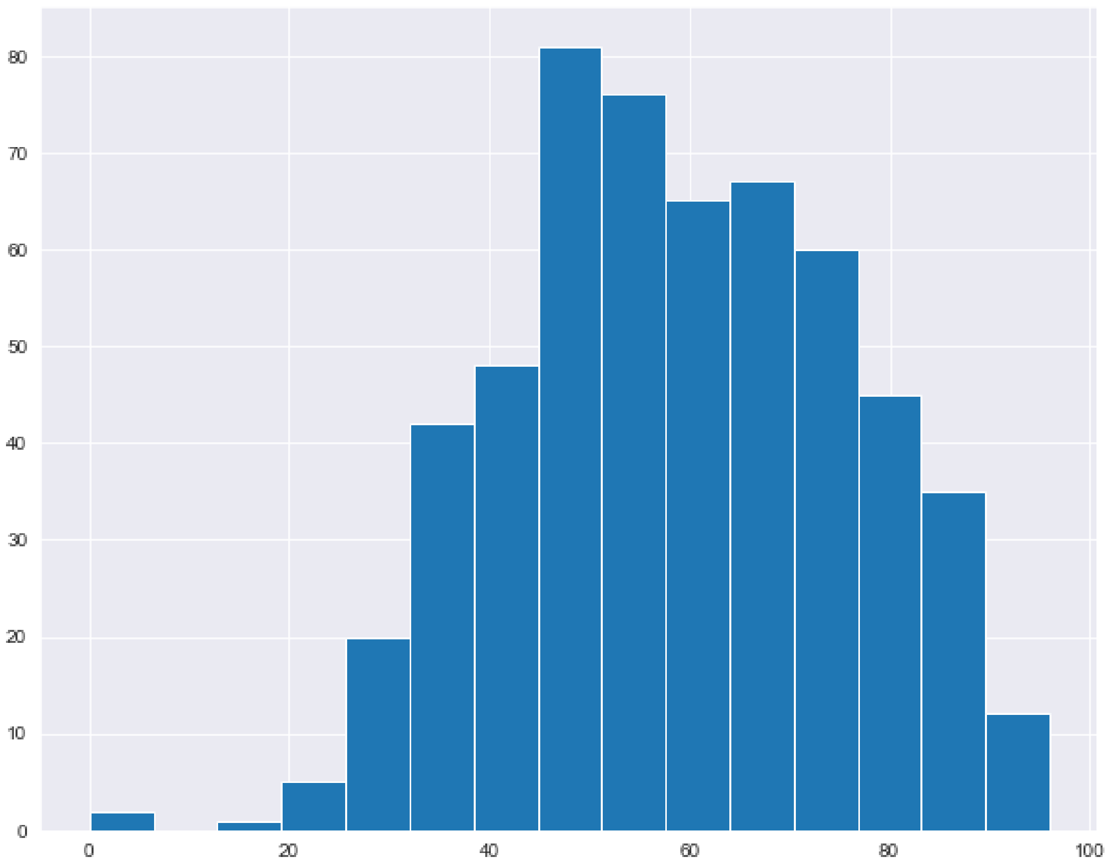
```
sns.displot(movies.Audience_Ratings,bins=15)
plt.show()
```



[150]:
```
m2=sns.distplot(movies.Rotten_Tomatoes_Ratings,bins=15)
plt.show()
```



[153]:
```
plt.hist(movies.Audience_Ratings,bins=15)
plt.show()
```



1 Stacked Histograms

```
[154]: movies.columns
```

```
[154]: Index(['Film', 'Genre', 'Rotten_Tomatoes_Ratings', 'Audience_Ratings',
              'Budget', 'Year'],
             dtype='object')
```

```
[155]: movies.columns=['Film', 'Genre', 'CriticRatings', 'AudienceRatings',
              'BudgetMillion', 'Year']
```

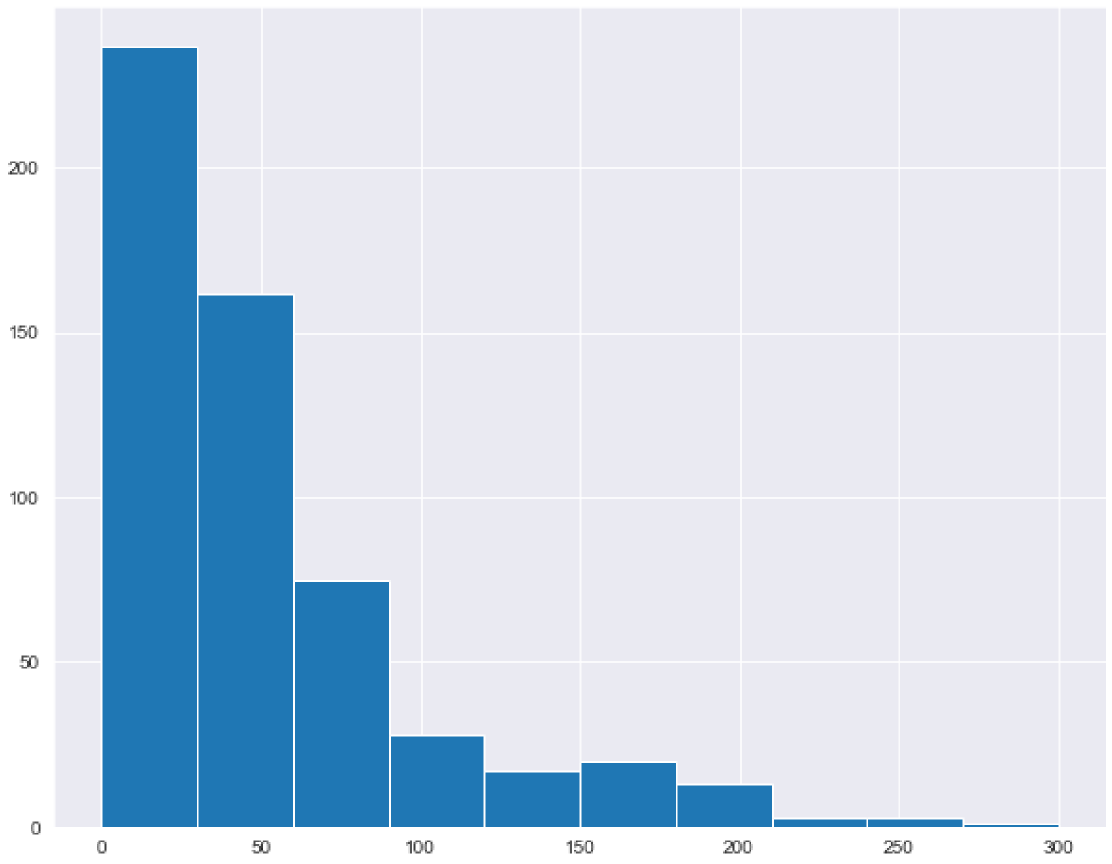
```
[156]: movies
```

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

```
..                    …        …            …              …
554         Your Highness     Comedy             26               36
555       Youth in Revolt     Comedy             68               52
556                Zodiac   Thriller             89               73
557            Zombieland     Action             90               87
558             Zookeeper     Comedy             14               42

     BudgetMillion  Year
0                8  2009
1              105  2008
2               20  2009
3               18  2010
4               20  2009
..             …   …
554             50  2011
555             18  2009
556             65  2007
557             24  2009
558             80  2011

[559 rows x 6 columns]
```

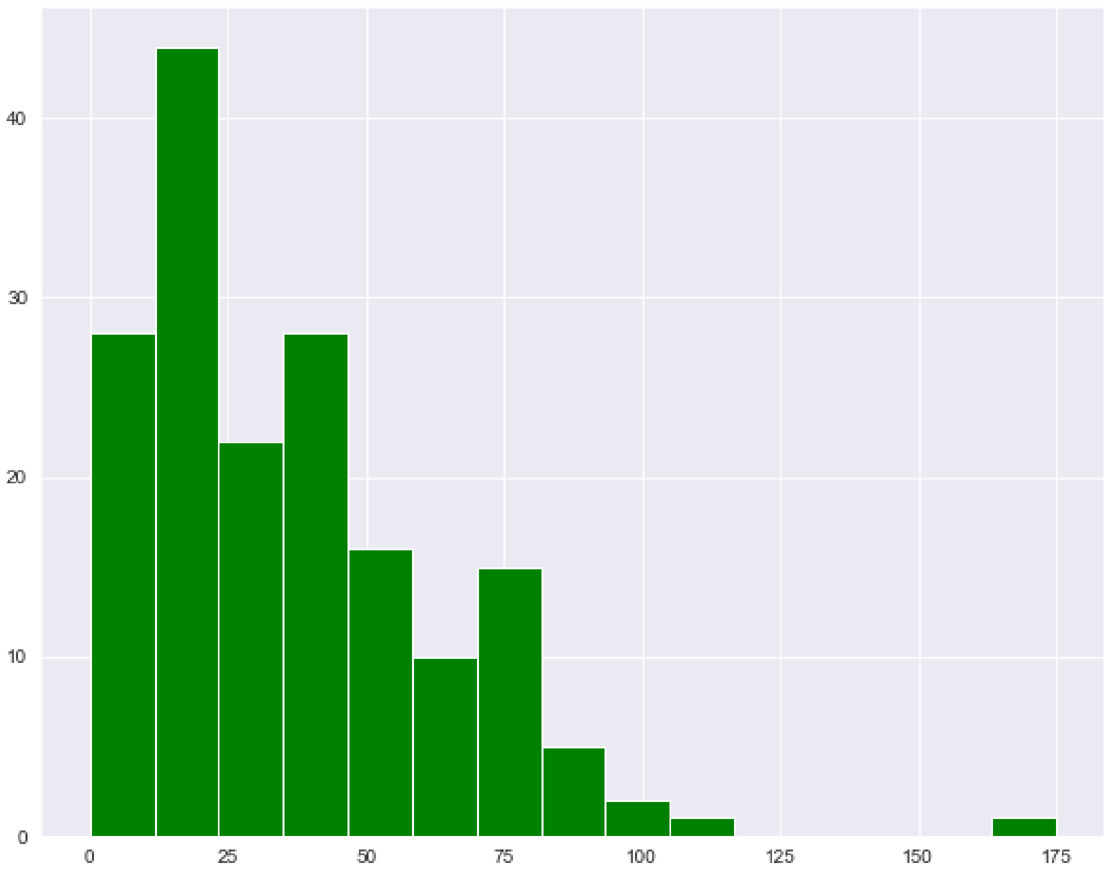
[157]:

```
movies[movies.Genre=='Comedy']
plt.hist(movies.BudgetMillion)
sns.set_style("darkgrid")
plt.show()
```

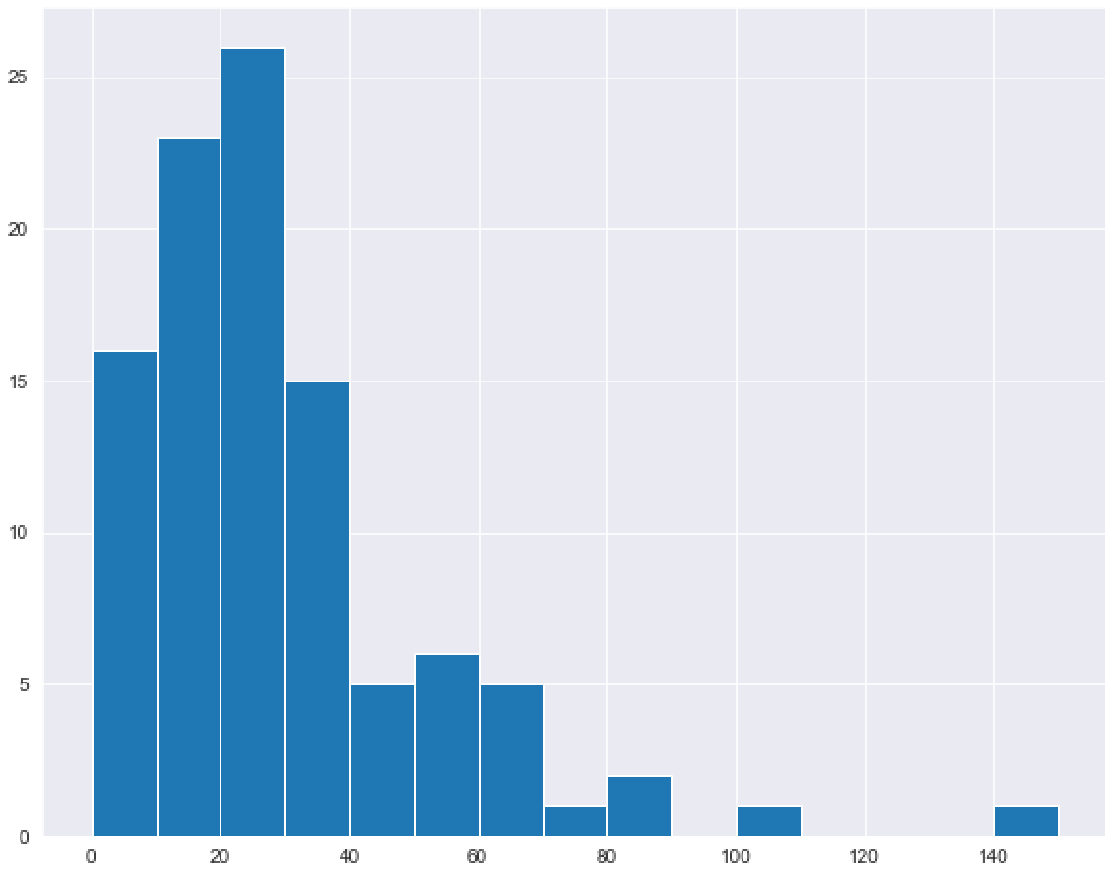


[158]:

```
plt.hist(movies[movies.Genre=="Comedy"].BudgetMillion,bins=15,color="Green")

plt.show()
```

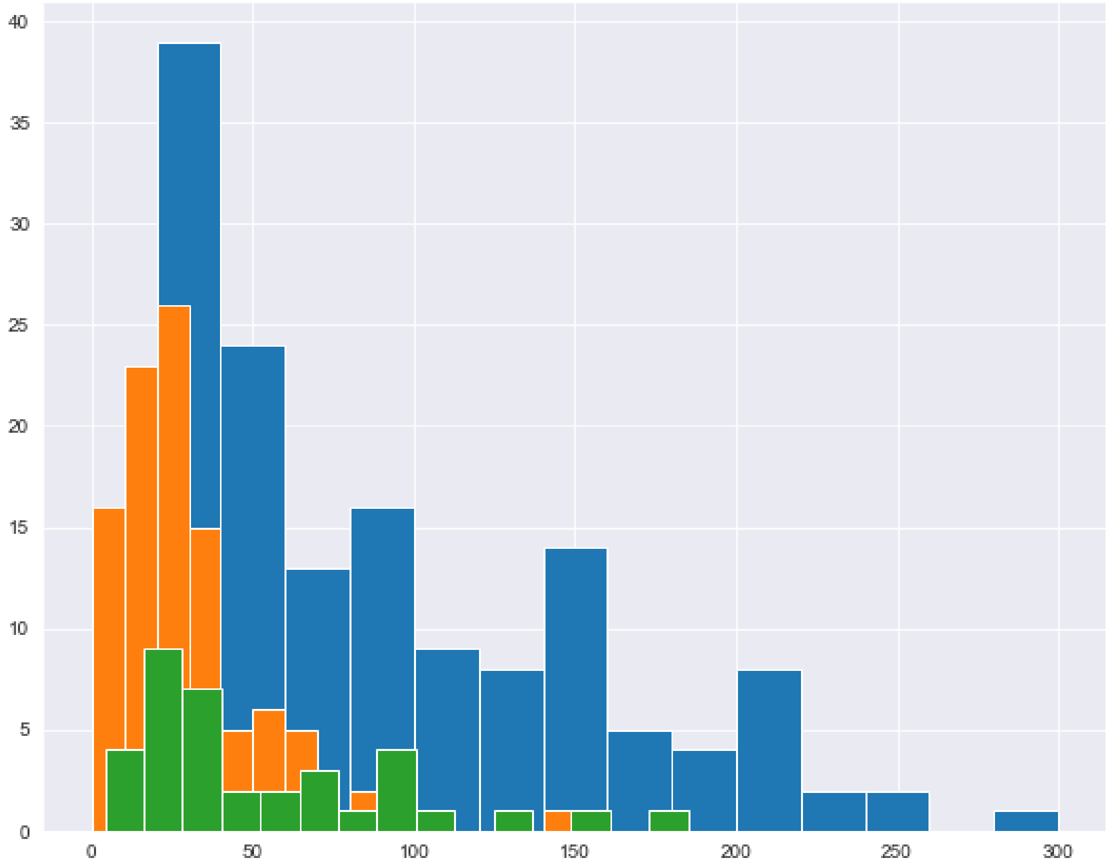


```
[159]: plt.hist(movies[movies.Genre=="Drama"].BudgetMillion,bins=15)
       plt.show()
```

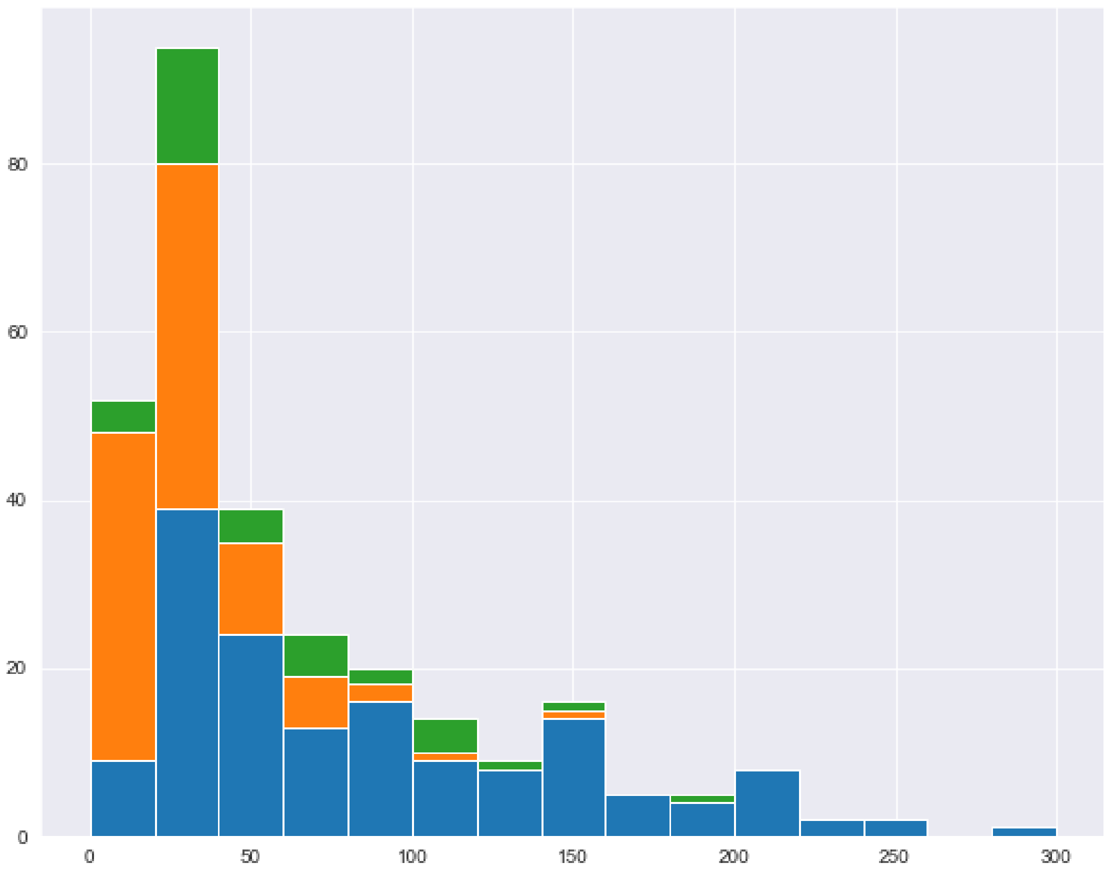


[160]:

```
plt.hist(movies[movies.Genre=="Action"].BudgetMillion,bins=15)
plt.hist(movies[movies.Genre=="Drama"].BudgetMillion,bins=15)
plt.hist(movies[movies.Genre=="Thriller"].BudgetMillion,bins=15)
plt.show()
```



```
[161]: plt.hist([movies[movies.Genre=="Action"].BudgetMillion,movies[movies.
        ↪Genre=="Drama"].BudgetMillion,
                movies[movies.Genre=="Thriller"].BudgetMillion],bins=15,stacked=True)
       plt.show()
```



[162]:

```
print(movies.Genre.cat.categories)
```

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

[163]:

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

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

[164]:

```
list1=[]
for gen in movies.Genre.cat.categories:
    list1.append(movies[movies.Genre==gen].BudgetMillion)
```

```
print(list1)
```

```
[2       20
5      200
15      35
29      20
30      20
      …
531    130
542     35
546    150
547    160
557     24
Name: BudgetMillion, Length: 154, dtype: int64, 1      105
3       18
19     200
21      45
24      40
32      78
46      20
65      38
68     140
130     73
165     12
166    125
167    250
168    150
176     36
178    150
192     70
193     60
241     60
272     37
341     19
363     70
386    130
401    155
459     59
463     25
506     38
540    100
548     60
Name: BudgetMillion, dtype: int64, 0       8
4      20
6      30
8      28
9       8
       ..
```

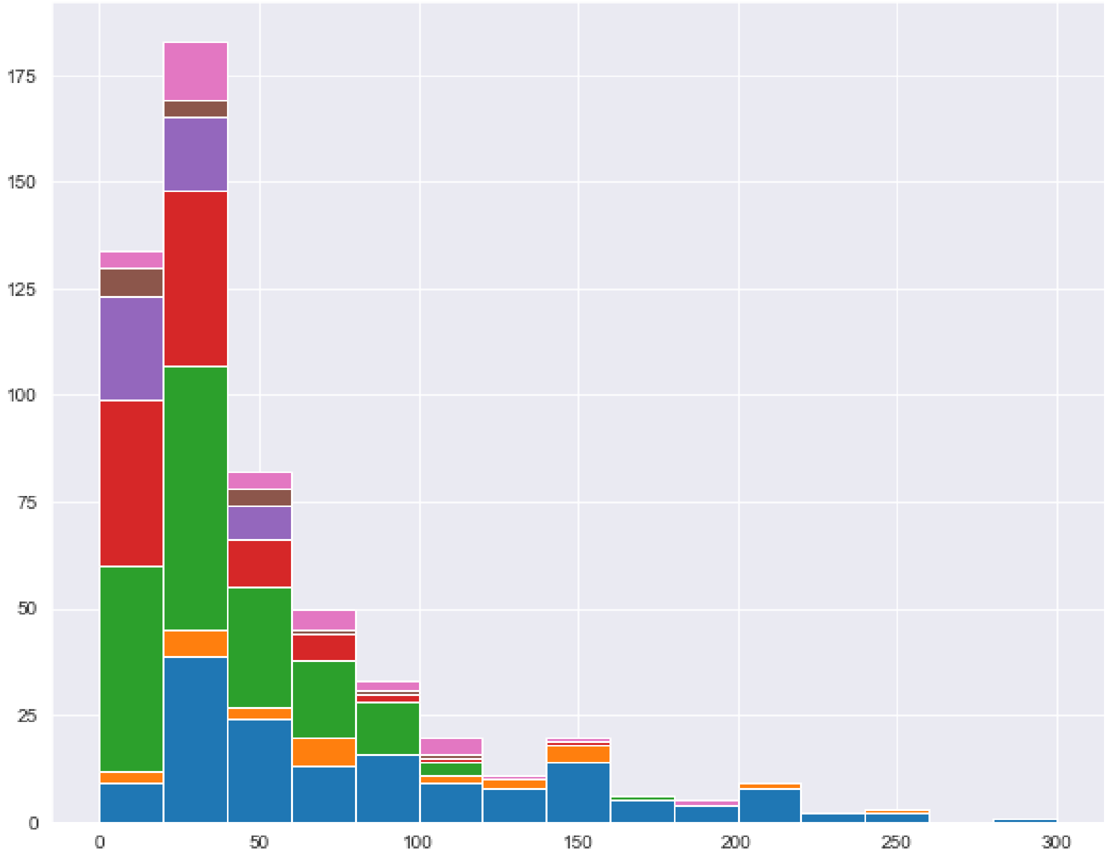
```
552    80
553    22
554    50
555    18
558    80
Name: BudgetMillion, Length: 172, dtype: int64, 10     30
11     20
13      7
18      8
23     20
       ..
529    66
532    38
534    21
541    15
545     2
Name: BudgetMillion, Length: 101, dtype: int64, 7      32
12     35
20     40
28      5
59     26
88     10
97     25
100    30
103    50
109    20
126    40
135    19
137    30
160    20
161    15
175    10
194     2
246    35
259    25
285    20
286    30
292     1
293     3
294     5
311    18
315    12
321    42
322     4
332    10
333    11
335    40
343    25
```

```
349     8
355    13
373    50
404    20
414    12
416    40
426     5
429    15
453    18
461    40
462    37
464    16
465    25
475     9
478    38
486    16
521    10
Name: BudgetMillion, dtype: int64, 16      45
42      17
78      50
108     60
136     35
201      0
208     80
244     17
250     20
255     40
266     56
284     15
290     30
354     35
507    110
510     15
524      5
525      2
Name: BudgetMillion, dtype: int64, 25     100
72      60
95      20
105     15
179    150
180     60
189     40
225     27
237      4
243     25
253     20
261     20
263    130
```

```
267     70
282     85
358     32
385     51
389     20
394    110
406    185
407    100
408     20
419     90
424     48
432     13
471     15
481    100
491     35
494     21
498     22
503     35
513     30
515     35
519     75
522     40
556     65
Name: BudgetMillion, dtype: int64]
```

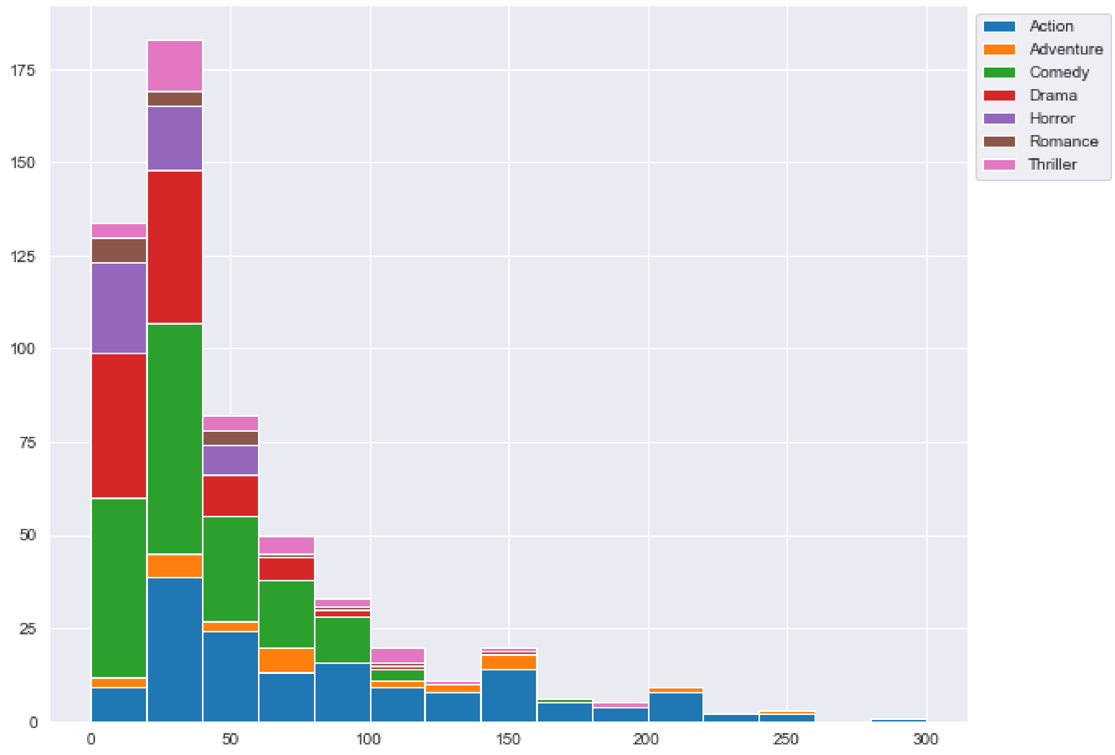
[165]:

```python
list1=[]
for gen in movies.Genre.cat.categories:
    list1.append(movies[movies.Genre==gen].BudgetMillion)
h=plt.hist(list1,stacked=True,bins=15,rwidth=1)
```



[166]:

```
list1=[]
mylable=[]
for gen in movies.Genre.cat.categories:
    list1.append(movies[movies.Genre==gen].BudgetMillion)
    mylable.append(gen)
h=plt.hist(list1,bins=15,stacked=True,rwidth=1,label=mylable)
plt.legend()
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.show()
```

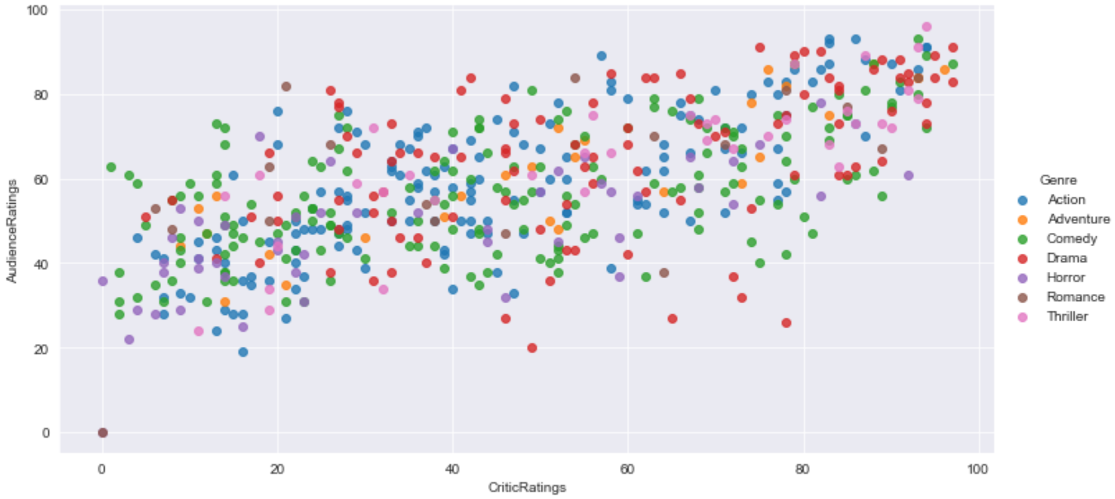


2 KDE Plot

[167]:
```
movies.head()
```

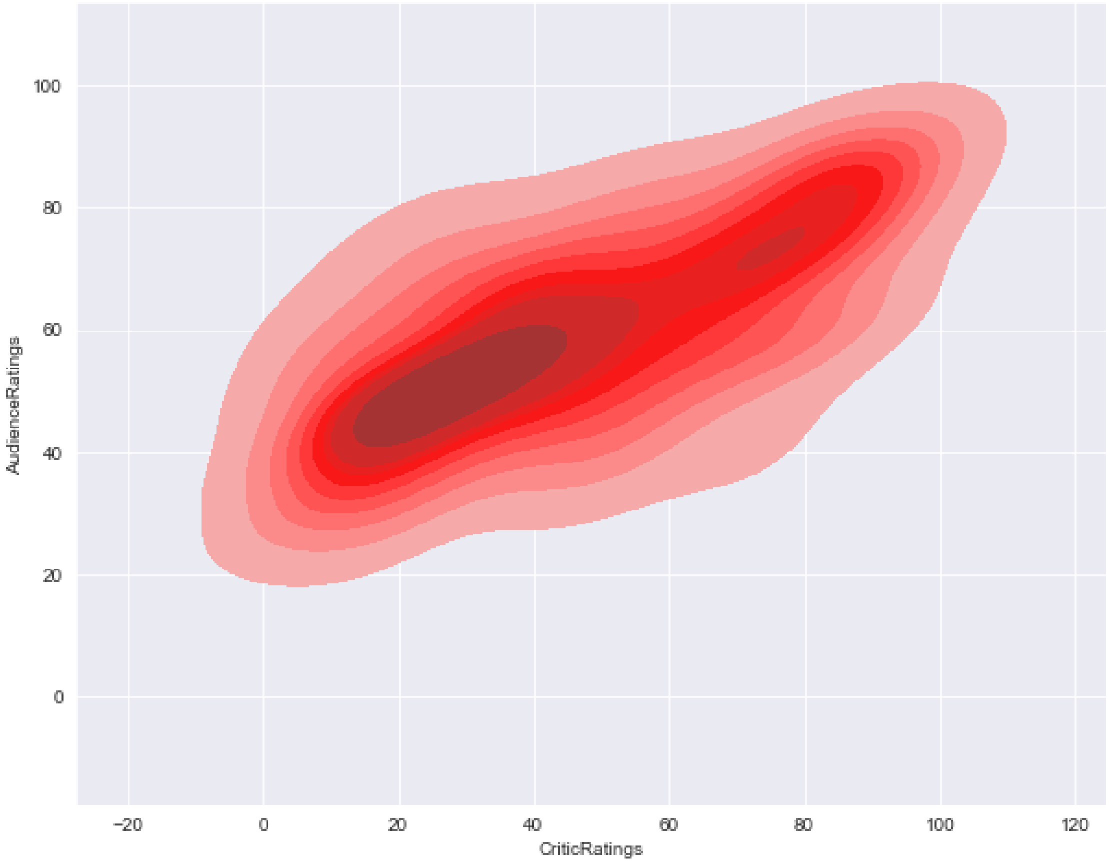
[167]:
```
                   Film      Genre  CriticRatings  AudienceRatings  \
0  (500) Days of Summer     Comedy             87               81
1          10,000 B.C.   Adventure              9               44
2            12 Rounds      Action             30               52
3            127 Hours   Adventure             93               84
4             17 Again      Comedy             55               70

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

[168]:
```
sns.
 ↪lmplot(data=movies,x="CriticRatings",y="AudienceRatings",fit_reg=False,hue="Genre",aspect=2
plt.show()
```



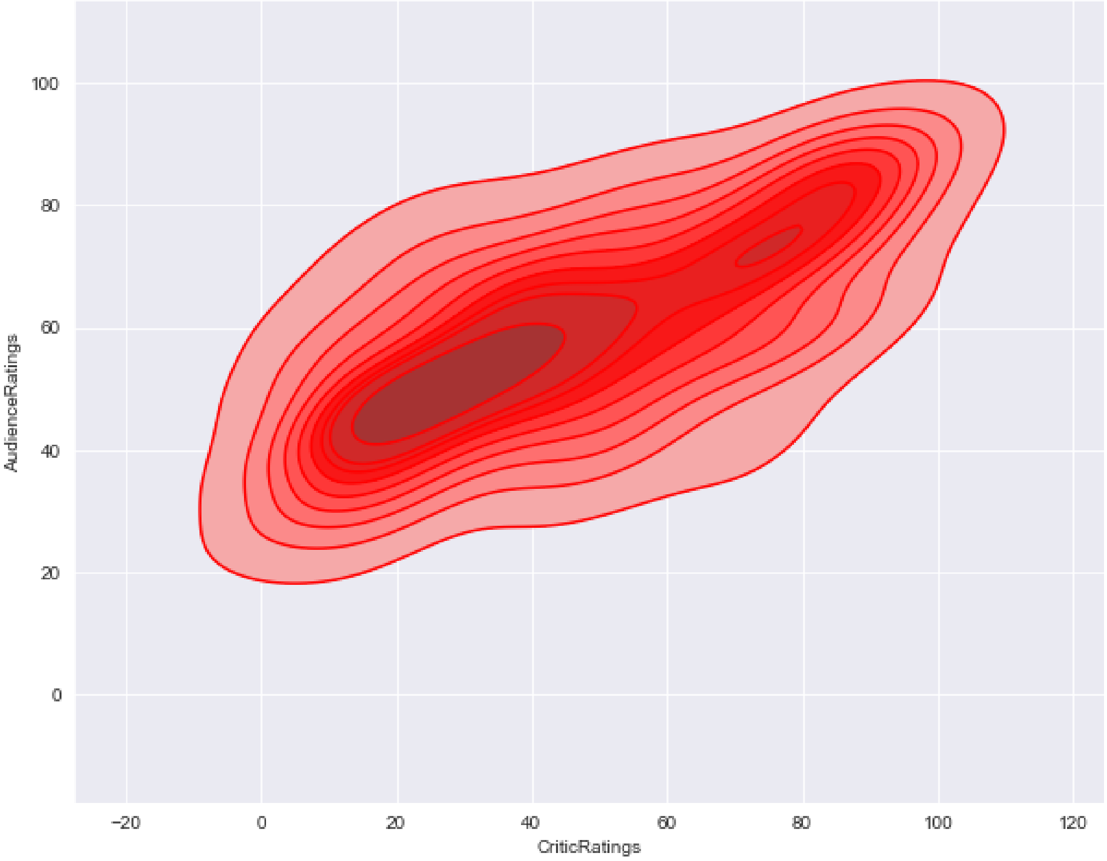
[169]:

```
k1=sns.kdeplot(movies.CriticRatings,movies.
↪AudienceRatings,shade=True,shade_lowest=False,color="Red")
```



[170]:

```
k1=sns.kdeplot(movies.CriticRatings,movies.
 ↪AudienceRatings,shade=True,shade_lowest=False,color='Red')
k1=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings,color="Red")
```



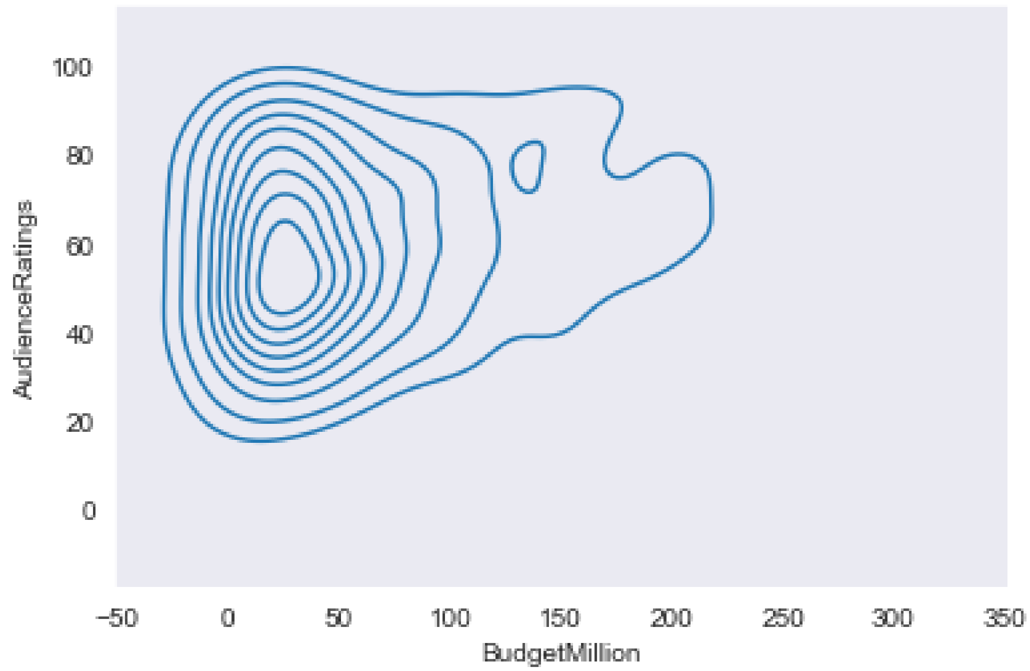
3 Subplots

[171]:

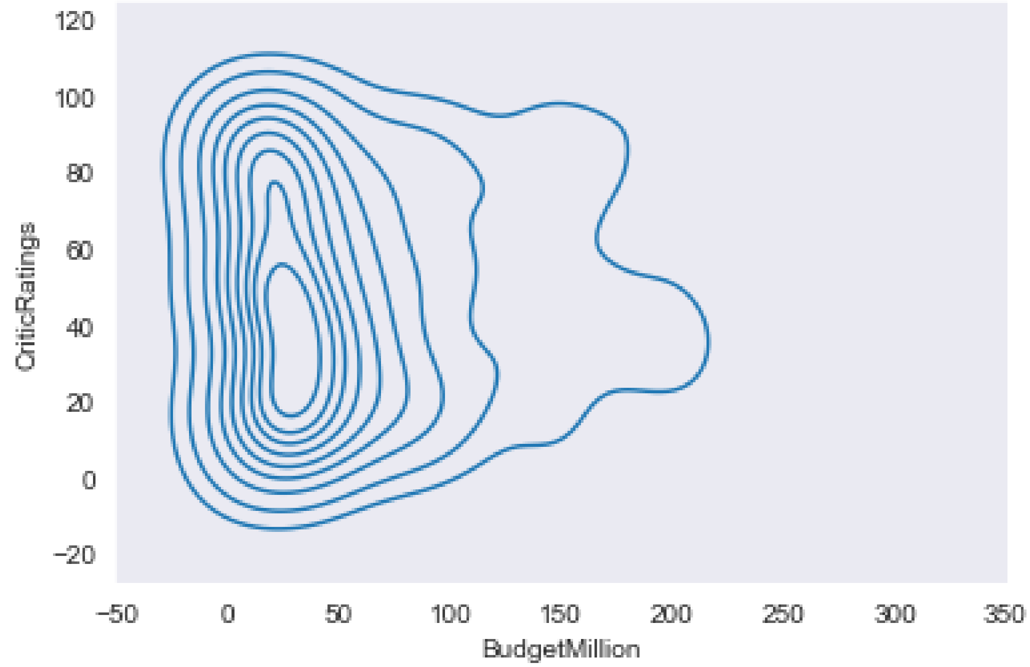
```
import matplotlib
from matplotlib import pyplot as plt
%matplotlib inline
```

[172]:

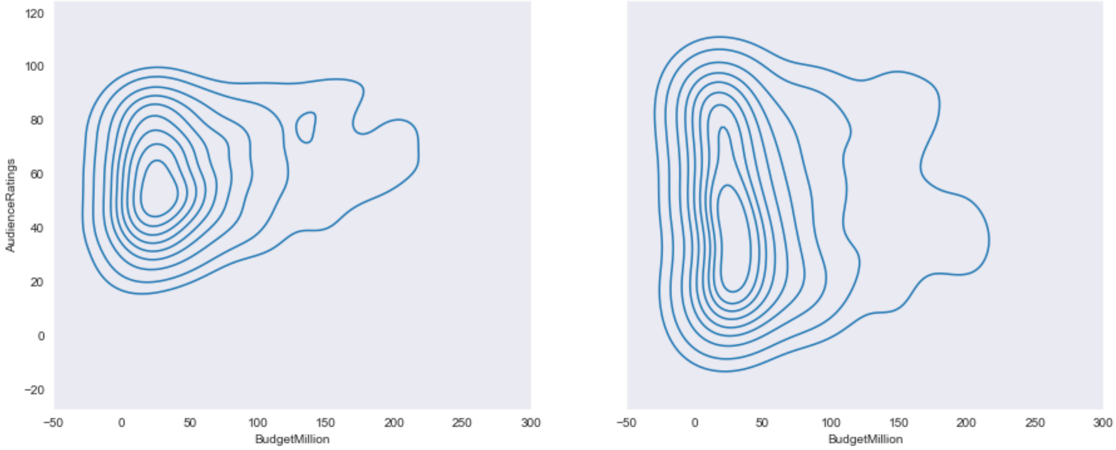
```
sns.set_style('dark')
k1=sns.kdeplot(movies.BudgetMillion,movies.AudienceRatings)
```



```
[173]: k2=sns.kdeplot(movies.BudgetMillion,movies.CriticRatings)
```

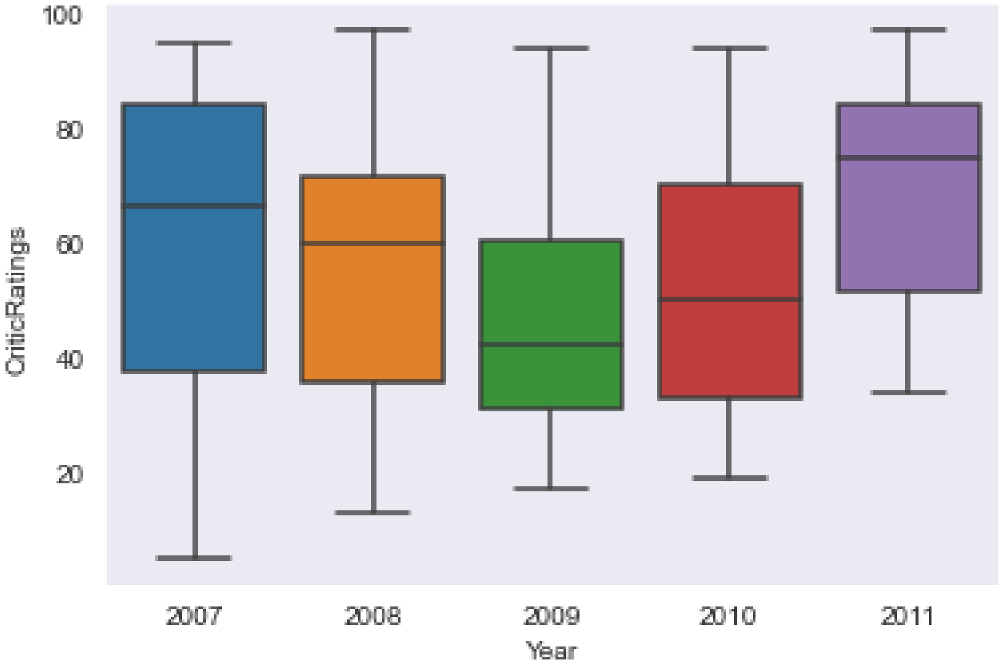


```
[174]: f, axes=plt.subplots(1,2, figsize=(15,6),sharex=True,sharey=True)
       k1=sns.kdeplot(movies.BudgetMillion,movies.AudienceRatings,ax=axes[0])
       k2=sns.kdeplot(movies.BudgetMillion,movies.CriticRatings,ax=axes[1])
       k1.set(xlim=(-50,300))
       plt.show()
```

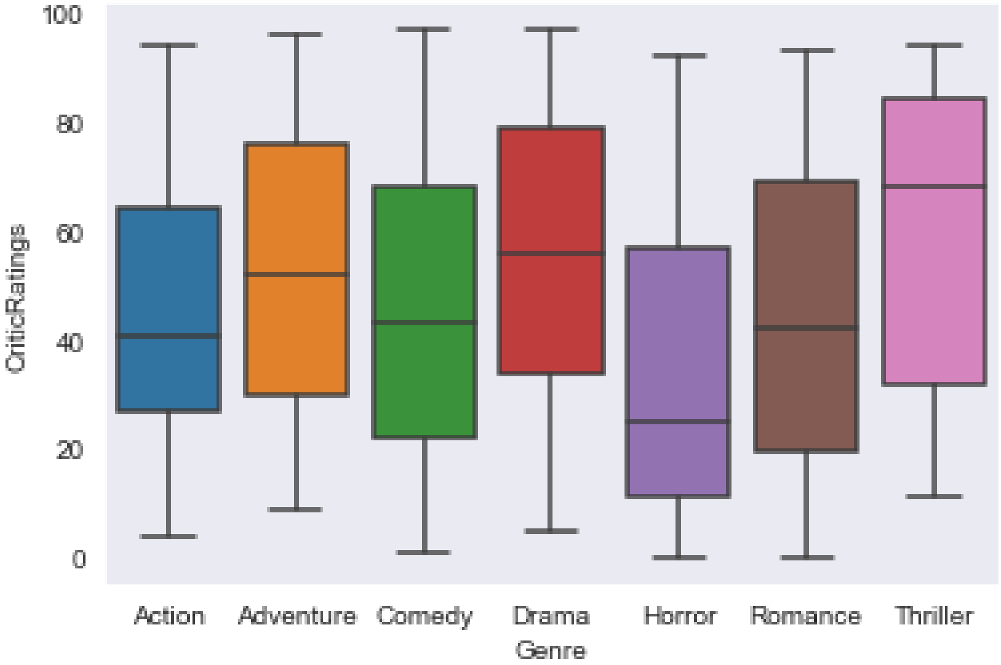


4 Violineplots Vs Boxplots

```
[175]: w=sns.boxplot(data=movies[movies.Genre=='Drama'],x='Year',y='CriticRatings')
```

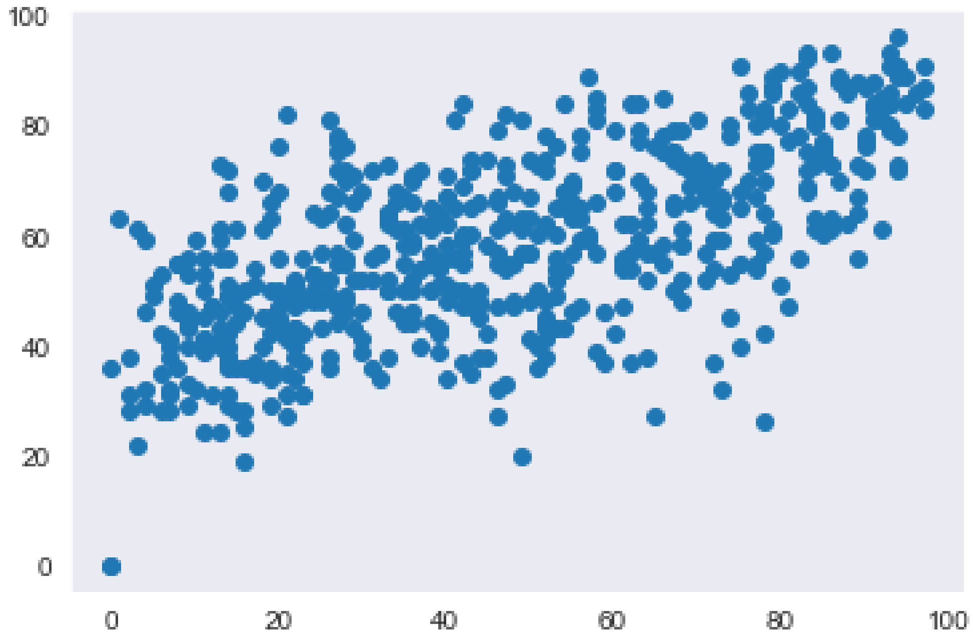


[176]:
```
w=sns.boxplot(data=movies,x='Genre',y='CriticRatings')
```

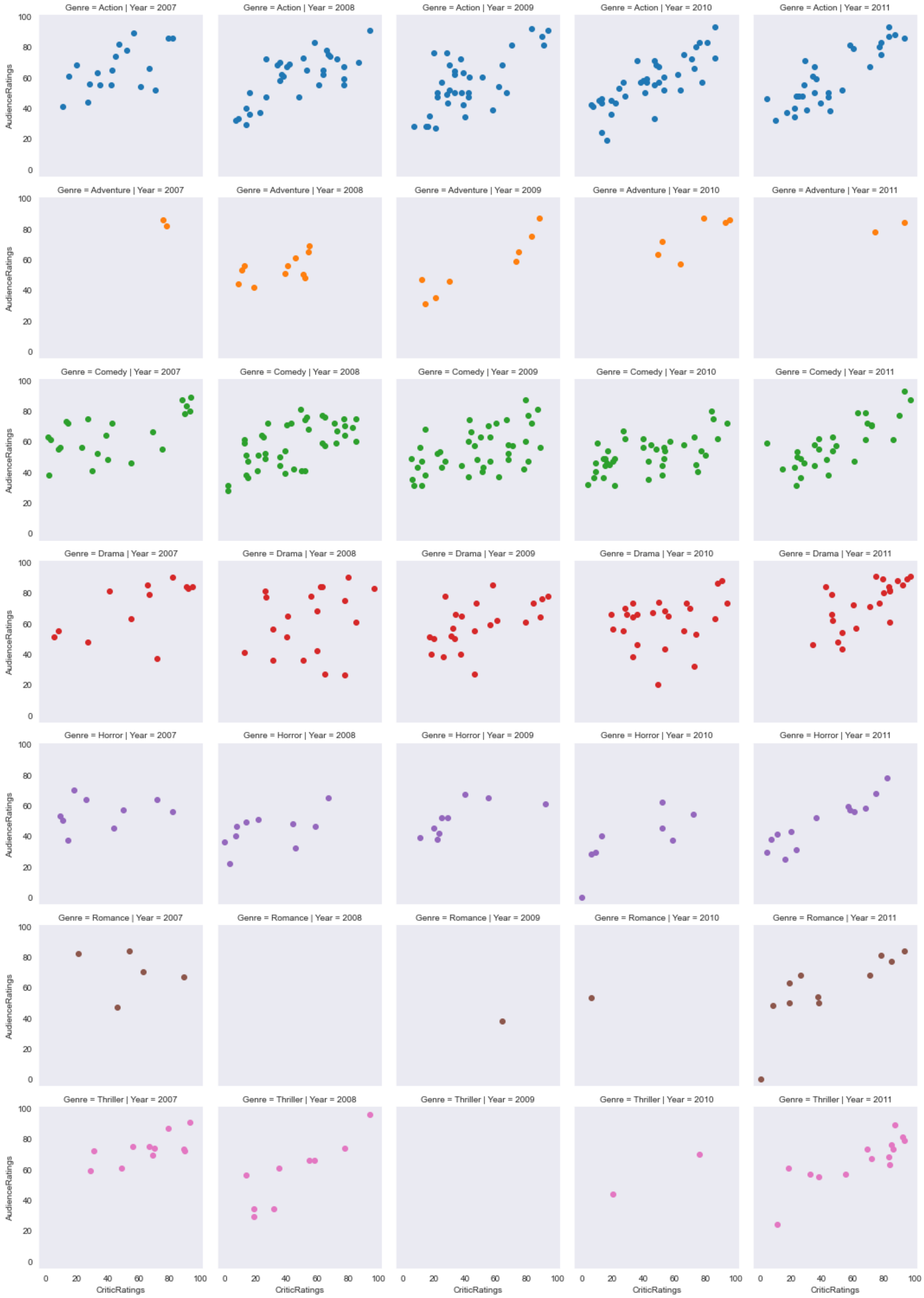


5 Creating a Facet Grid

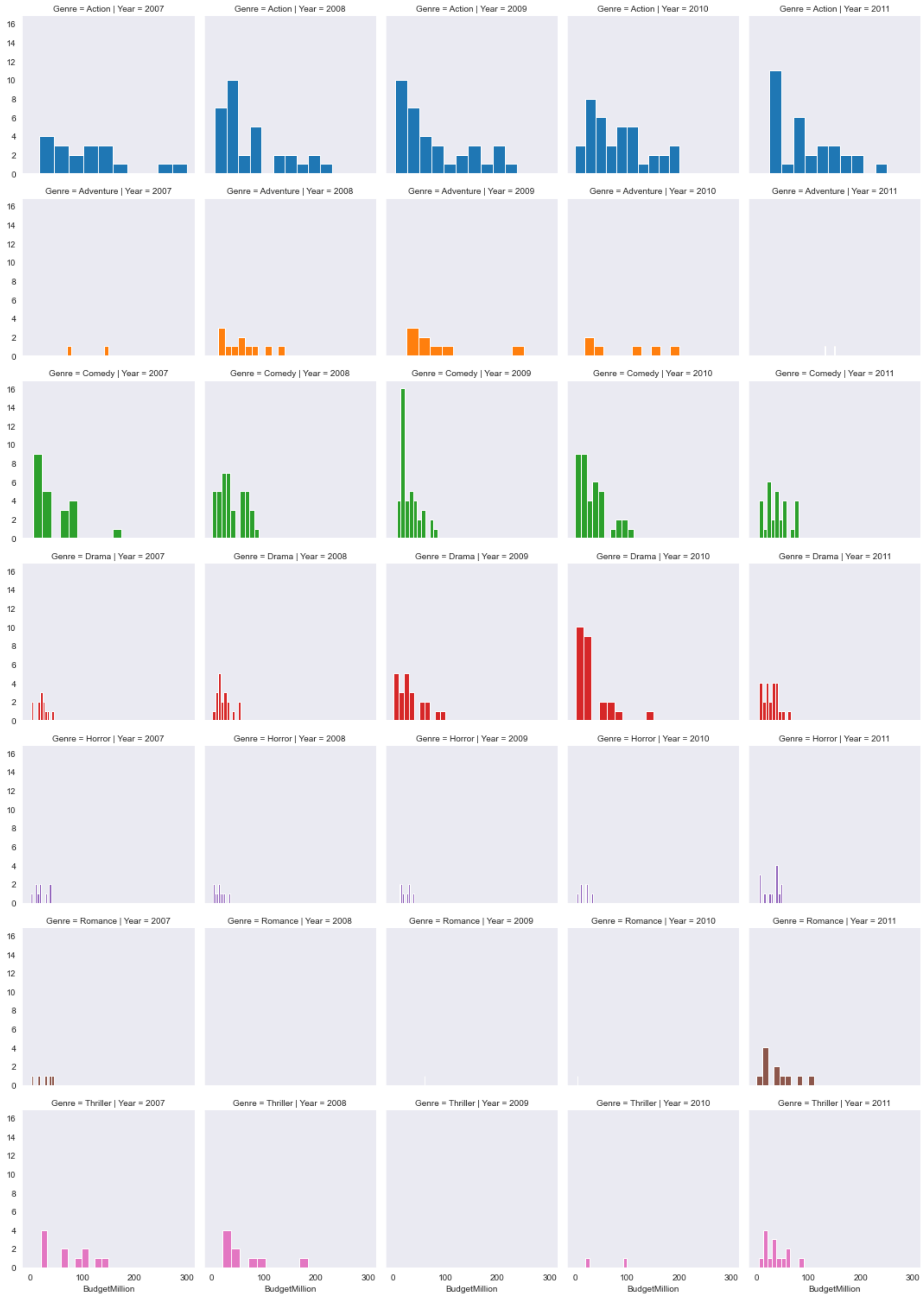
[179]:
```
plt.scatter(movies.CriticRatings,movies.AudienceRatings)
plt.show()
```



```
[180]: g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
       g=g.map(plt.scatter,'CriticRatings','AudienceRatings')
```

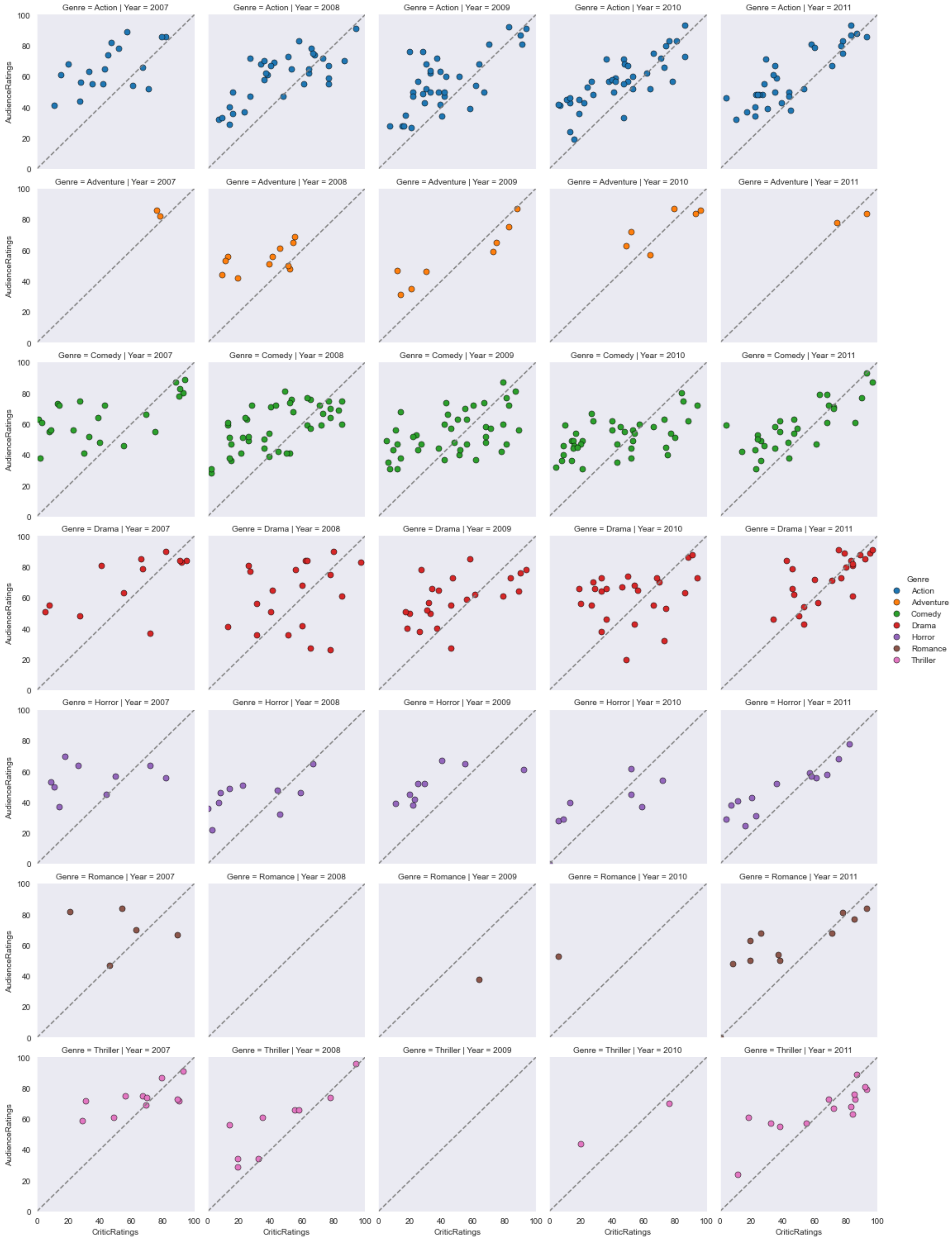

Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
Genre = Comedy | Year = 2007
Genre = Comedy | Year = 2008
Genre = Comedy | Year = 2009
Genre = Comedy | Year = 2010
Genre = Comedy | Year = 2011
Genre = Drama | Year = 2007
Genre = Drama | Year = 2008
Genre = Drama | Year = 2009
Genre = Drama | Year = 2010
Genre = Drama | Year = 2011
Genre = Horror | Year = 2007
Genre = Horror | Year = 2008
Genre = Horror | Year = 2009
Genre = Horror | Year = 2010
Genre = Horror | Year = 2011
Genre = Romance | Year = 2007
Genre = Romance | Year = 2008
Genre = Romance | Year = 2009
Genre = Romance | Year = 2010
Genre = Romance | Year = 2011
Genre = Thriller | Year = 2007
Genre = Thriller | Year = 2008
Genre = Thriller | Year = 2009
Genre = Thriller | Year = 2010
Genre = Thriller | Year = 2011
AudienceRatings
0
20
40
60
80
100
CriticRatings
0
20
40
60
80
100

[181]:
```
g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
g=g.map(plt.hist,'BudgetMillion')
```



[182]:

```
g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
kws=dict(s=50,linewidth=0.5,edgecolor='black')
g=g.map(plt.scatter,'CriticRatings','AudienceRatings',**kws)
g.set(xlim=(0,100),ylim=(0,100))
for ax in g.axes.flat:
    ax.plot((0,100),(0,100),c='gray',ls='--')
g.add_legend()
plt.show()
```


Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
Genre = Comedy | Year = 2007
Genre = Comedy | Year = 2008
Genre = Comedy | Year = 2009
Genre = Comedy | Year = 2010
Genre = Comedy | Year = 2011
Genre = Drama | Year = 2007
Genre = Drama | Year = 2008
Genre = Drama | Year = 2009
Genre = Drama | Year = 2010
Genre = Drama | Year = 2011
Genre = Horror | Year = 2007
Genre = Horror | Year = 2008
Genre = Horror | Year = 2009
Genre = Horror | Year = 2010
Genre = Horror | Year = 2011
Genre = Romance | Year = 2007
Genre = Romance | Year = 2008
Genre = Romance | Year = 2009
Genre = Romance | Year = 2010
Genre = Romance | Year = 2011
Genre = Thriller | Year = 2007
Genre = Thriller | Year = 2008
Genre = Thriller | Year = 2009
Genre = Thriller | Year = 2010
Genre = Thriller | Year = 2011
AudienceRatings
0
20
40
60
80
100
CriticRatings
0
20
40
60
80
100
Genre
Action
Adventure
Comedy
Drama
Horror
Romance
Thriller

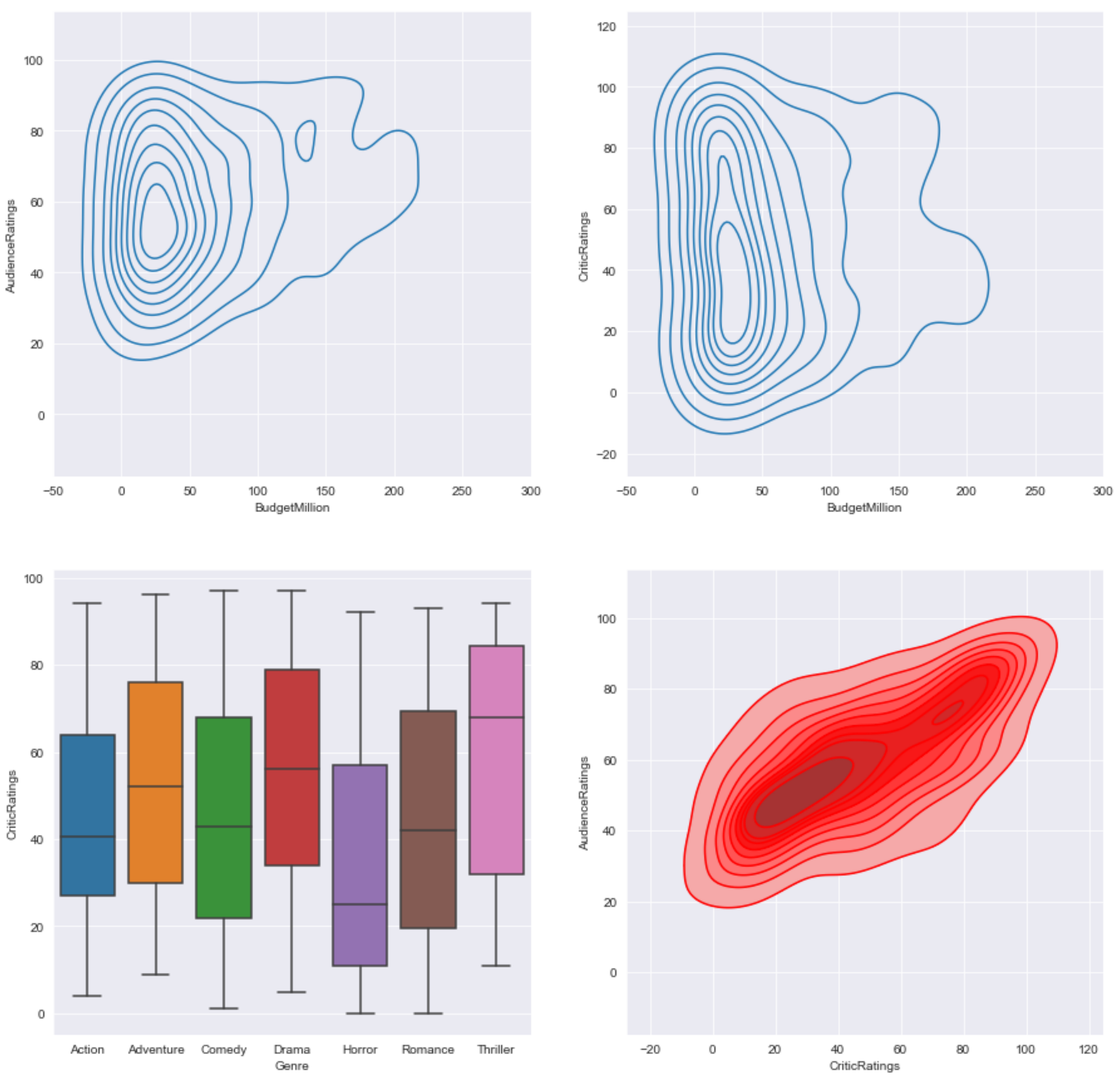
6 Dashboard

[183]:

```
import matplotlib
from matplotlib import pyplot as plt
%matplotlib inline
```

[184]:

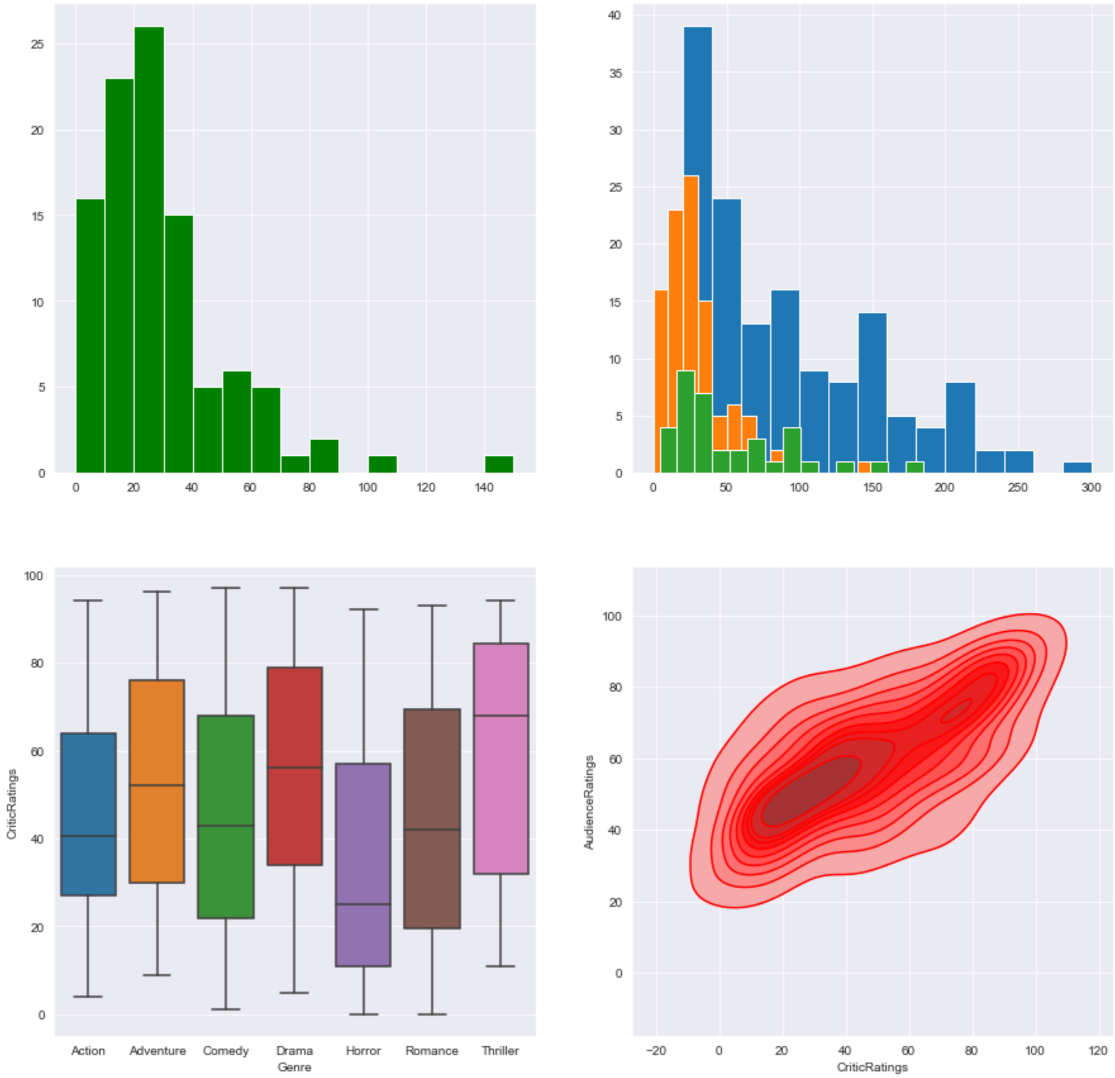
```
sns.set_style("darkgrid")
f, axes=plt.subplots(2,2,figsize=(15,15))
k1=sns.kdeplot(movies.BudgetMillion,movies.AudienceRatings,ax=axes[0,0])
k2=sns.kdeplot(movies.BudgetMillion,movies.CriticRatings,ax=axes[0,1])
k3=sns.boxplot(data=movies,x='Genre',y='CriticRatings',ax=axes[1,0])
k4=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings,\
              shade=True,shade_lowest=False,color='Red',ax=axes[1,1])
k4=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings,\
              color='Red',ax=axes[1,1])
k1.set(xlim=(-50,300))
k2.set(xlim=(-50,300))
plt.show()
```



[194]:

```
sns.set_style("darkgrid")
f, axes=plt.subplots(2,2,figsize=(15,15))
ax=axes[0,0].hist(movies[movies.Genre=='Drama'].
 ↪BudgetMillion,bins=15,color='Green')

ax=axes[0,1].hist(movies[movies.Genre=='Action'].BudgetMillion,bins=15)
ax=axes[0,1].hist(movies[movies.Genre=='Drama'].BudgetMillion,bins=15)
ax=axes[0,1].hist(movies[movies.Genre=='Thriller'].BudgetMillion,bins=15)

w=sns.boxplot(data=movies,x='Genre',y='CriticRatings',ax=axes[1,0])

k1=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings, \
              shade=True,shade_lowest=False,color='Red',ax=axes[1,1])
```

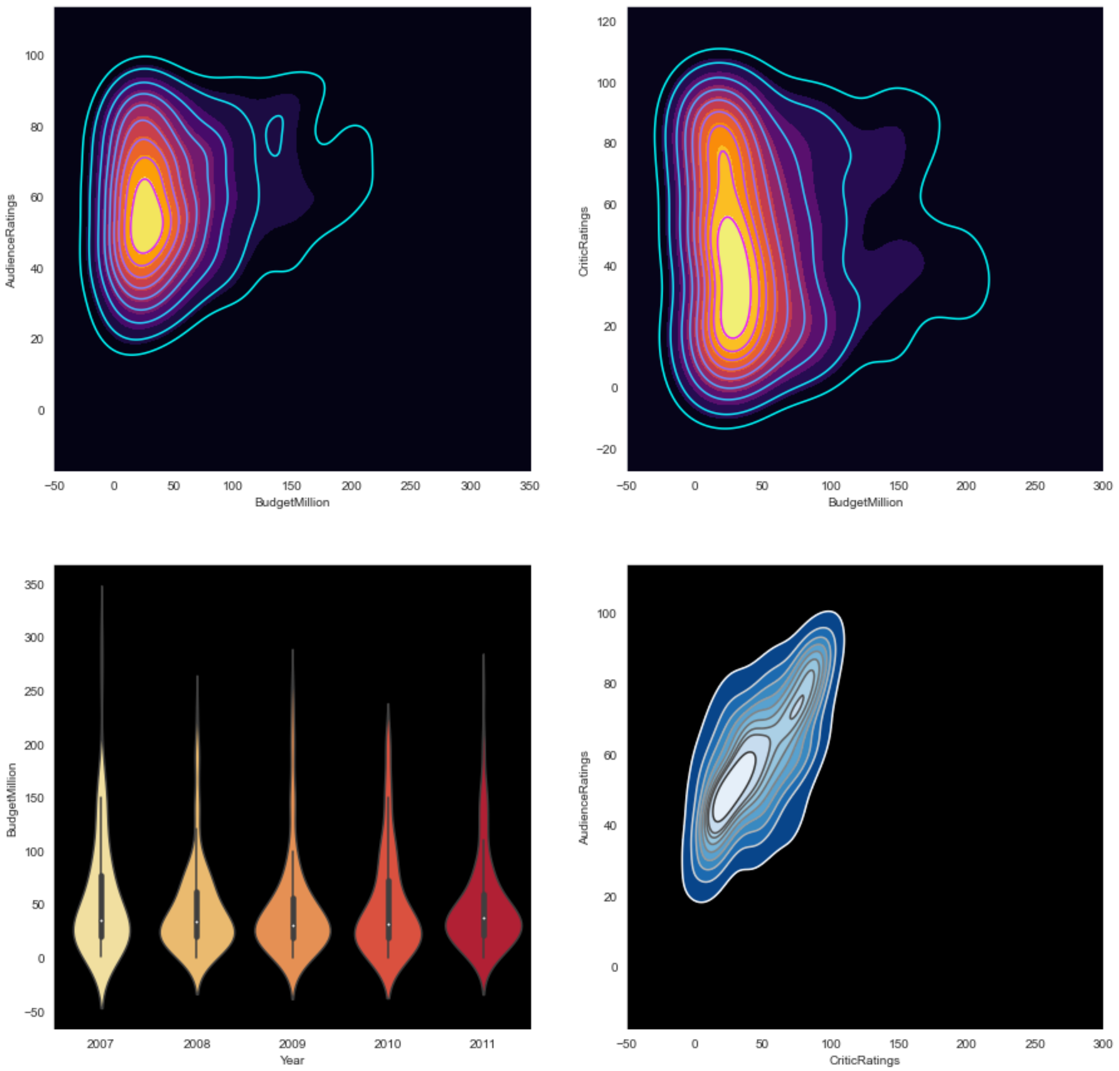
```
k1=sns.kdeplot(movies.CriticRatings,movies.
 ↪AudienceRatings,color='Red',ax=axes[1,1])
```



[196]:
```
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
```

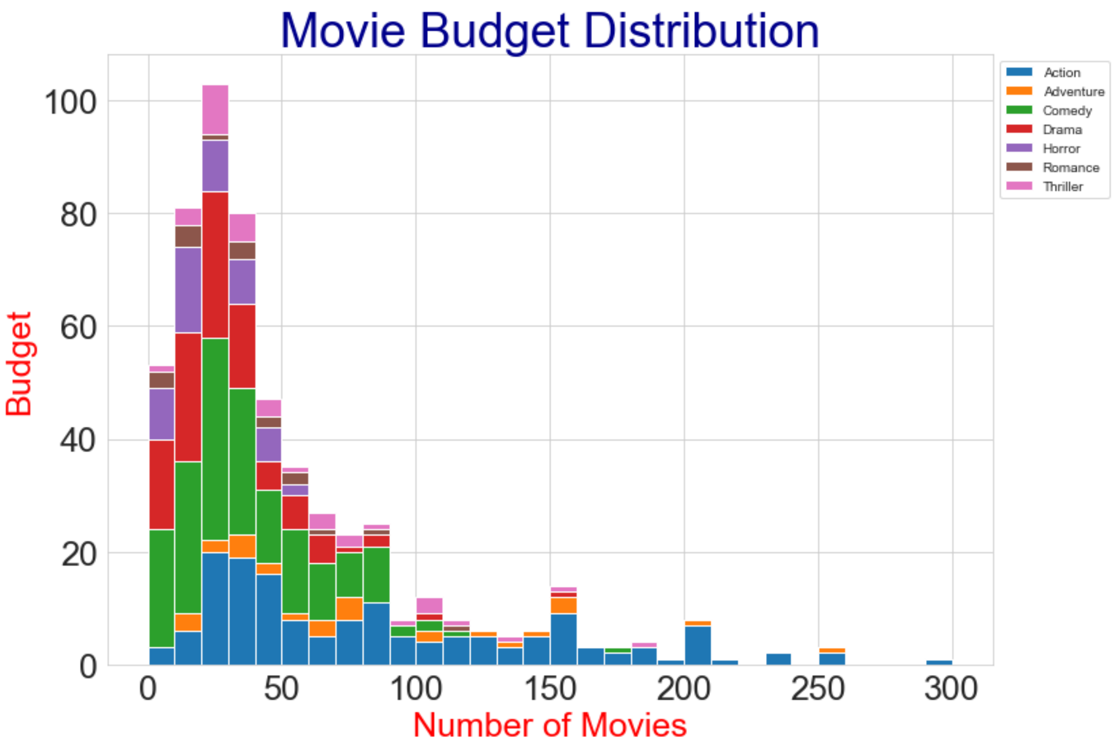
[213]:
```
sns.set_style("dark",{"axes.facecolor":"black"})
f,axes=plt.subplots(2,2,figsize=(15,15))

k1=sns.kdeplot(movies.BudgetMillion,movies.AudienceRatings,\
              shade=True,shade_lowest=True,cmap='inferno',ax=axes[0,0])
```

```
k1=sns.kdeplot(movies.BudgetMillion,movies.AudienceRatings,cmap='cool',␣
 ↪ax=axes[0,0])

k2=sns.kdeplot(movies.BudgetMillion,movies.CriticRatings,\
              shade=True,shade_lowest=True,cmap='inferno',ax=axes[0,1])

k2=sns.kdeplot(movies.BudgetMillion,movies.CriticRatings,\
              cmap='cool',ax=axes[0,1])

w=sns.violinplot(data=movies,x='Year',y='BudgetMillion',\
                palette='YlOrRd',ax=axes[1,0])

k1=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings,\
              shade=True,shade_lowest=False,cmap='Blues_r',ax=axes[1,1])
k1=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings,\
              cmap='gist_gray_r',color='Red',ax=axes[1,1])

k1.set(xlim=(-50,300))
k2.set(xlim=(-50,300))
plt.show()
```



[227]:

```
list1=list([])
mylable=list([])
for gen in movies.Genre.cat.categories:
    list1.append(movies[movies.Genre==gen].BudgetMillion)
    mylable.append(gen)

sns.set_style("whitegrid")
fig, ax=plt.subplots()
fig.set_size_inches(11.7,8.27)
h=plt.hist(list1,bins=30,stacked=True,rwidth=1,label=mylable)
plt.title("Movie Budget Distribution",fontsize=35,color='darkblue')
plt.xlabel("Number of Movies",fontsize=25,color='Red')
plt.ylabel("Budget",fontsize=25,color='Red')
plt.yticks(fontsize=25)
plt.xticks(fontsize=25)
plt.legend()
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.show()
```



[ ]: