Movie-Ratings Analysis

August 17, 2021

[2]:
```
import pandas as pd
import os
```

[3]:
```
os.getcwd()
```

[3]:
```
'C:\\Users\\ddaya\\Documents\\Python Programs'
```

[4]:
```
os.chdir('C:\\Users\\ddaya\\OneDrive\\Documents\\Python programming')
```

[5]:
```
movies=pd.read_csv('Movie-Ratings.csv')
```

[6]:
```
movies
```

[6]:
```
                     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]
```

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

```
[7]: 559
```

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

```
[8]:                  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
```

```
[9]: movies.columns
```

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

```
[10]: movies.columns=['Film', 'Genre', 'CriticRatings', 'AudienceRatings',
             'Budget(million $)', 'Year']
```

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

```
[11]:                  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

        Budget(million $)  Year
     0                  8  2009
     1                105  2008
     2                 20  2009
     3                 18  2010
     4                 20  2009
```

[12]:
```
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   CriticRatings      559 non-null    int64
 3   AudienceRatings    559 non-null    int64
 4   Budget(million $)  559 non-null    int64
 5   Year               559 non-null    int64
dtypes: int64(4), object(2)
memory usage: 26.3+ KB
```

[13]:
```
movies.describe()  # it's worng as year also calculate
```

[13]:
```
       CriticRatings  AudienceRatings  Budget(million $)         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
```

[14]:
```
movies.Film=movies.Film.astype('category')
```

[15]:
```
movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Film               559 non-null    category
 1   Genre              559 non-null    object
 2   CriticRatings      559 non-null    int64
 3   AudienceRatings    559 non-null    int64
 4   Budget(million $)  559 non-null    int64
 5   Year               559 non-null    int64
dtypes: category(1), int64(4), object(1)
memory usage: 43.6+ KB
```

[16]:
```
movies.Genre=movies.Genre.astype('category')
movies.Year=movies.Year.astype('category')
```

[17]:
```
movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 559 entries, 0 to 558
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Film                559 non-null    category
 1   Genre               559 non-null    category
 2   CriticRatings       559 non-null    int64
 3   AudienceRatings     559 non-null    int64
 4   Budget(million $)   559 non-null    int64
 5   Year                559 non-null    category
dtypes: category(3), int64(3)
memory usage: 36.5 KB
```

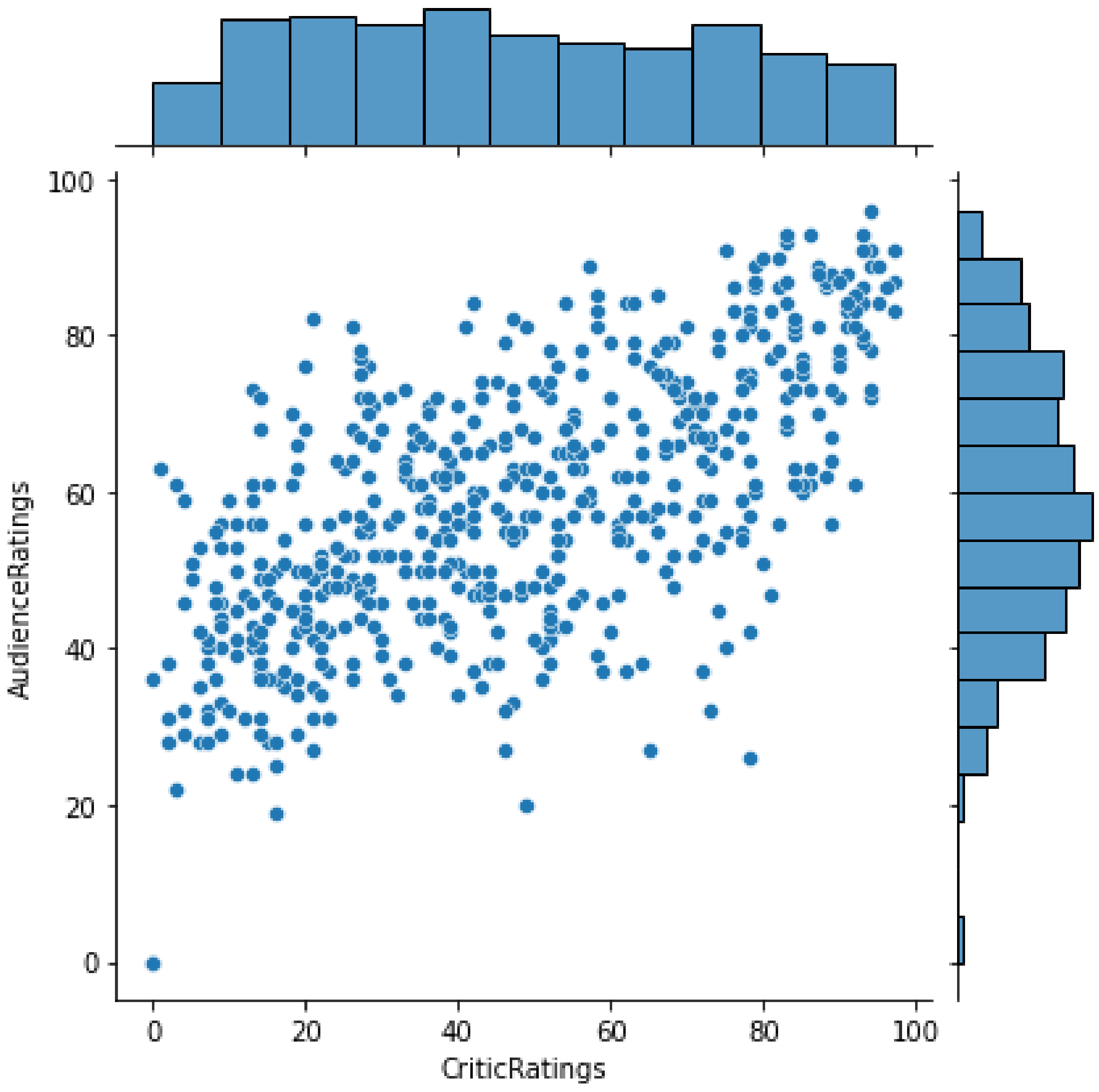
[18]:
```
movies.describe()
```

[18]:
```
       CriticRatings  AudienceRatings  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
```

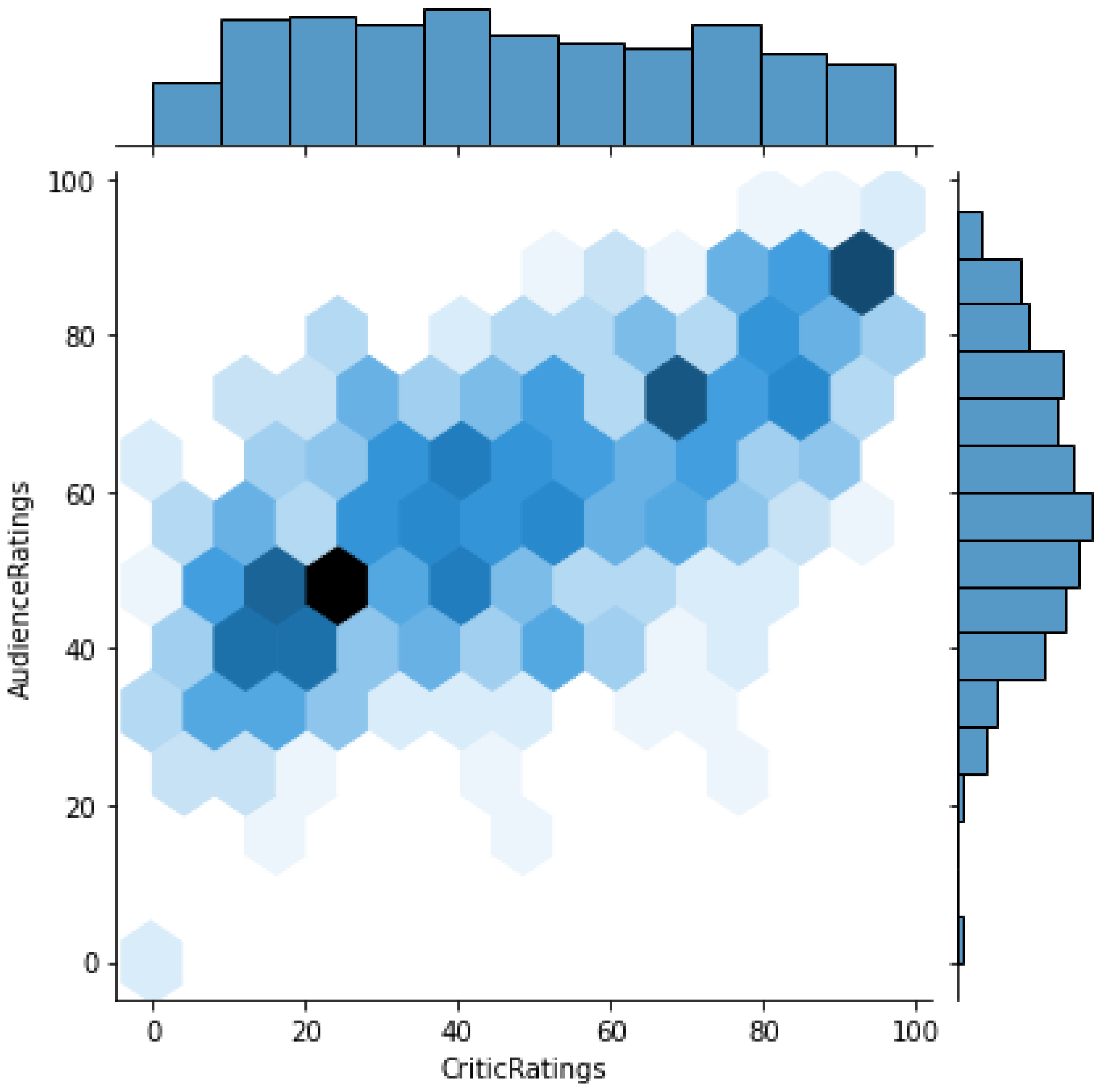
0.1 # Jointplots

[19]:
```
import matplotlib as plt
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

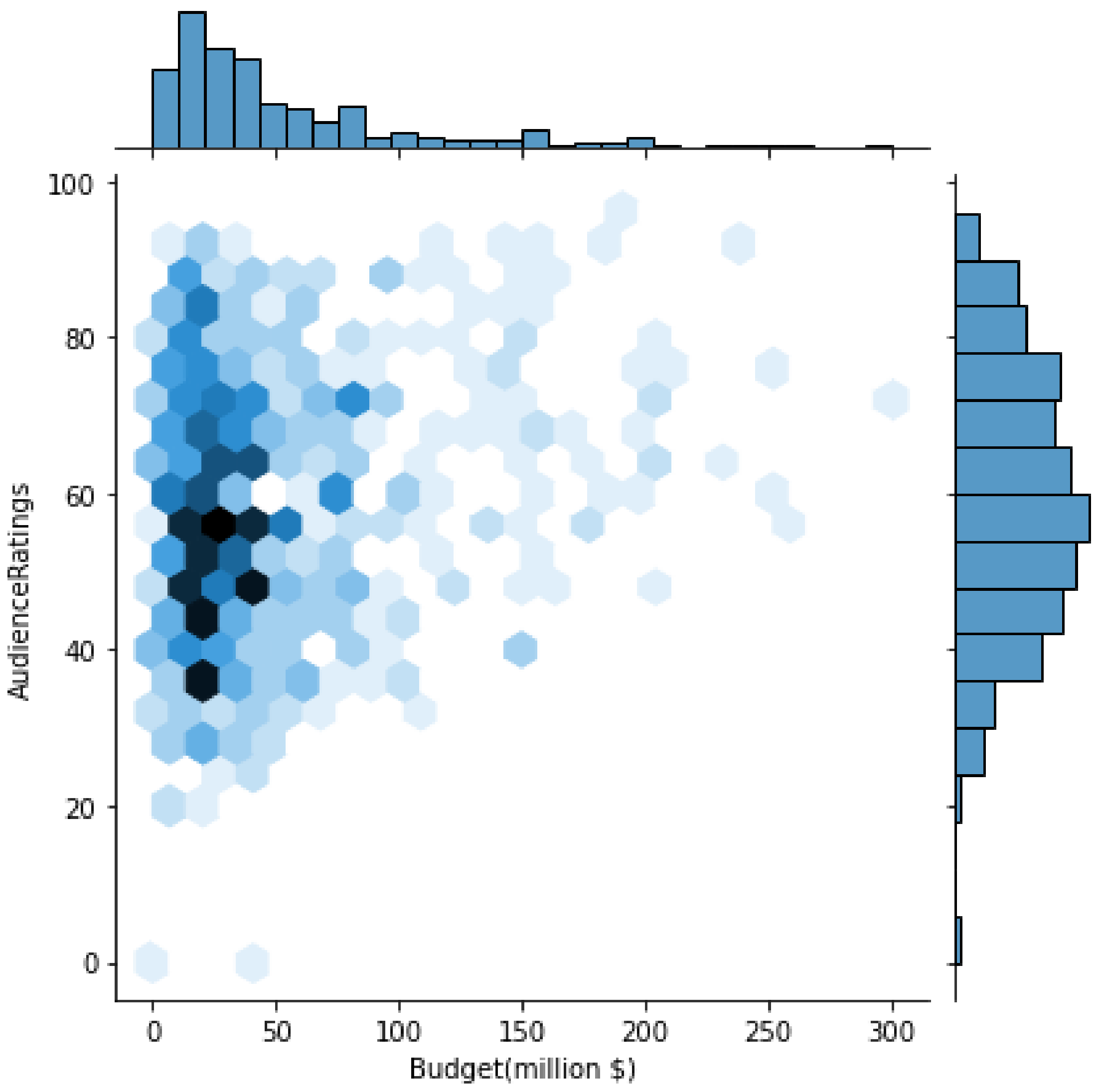
[20]:
```
j=sns.jointplot(data=movies, x='CriticRatings',y='AudienceRatings')
```



[21]:
```
j=sns.jointplot(data=movies, x='CriticRatings',y='AudienceRatings',kind='hex')
```



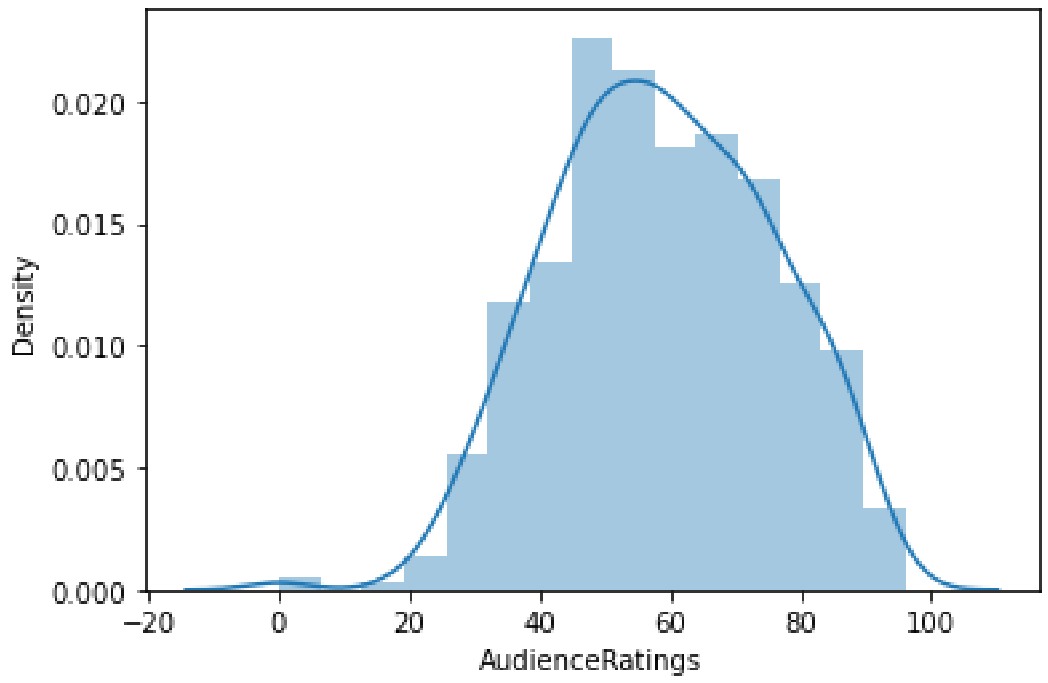
```
[22]: j=sns.jointplot(data=movies, x='Budget(million␣
       ↪$)',y='AudienceRatings',kind='hex')
```



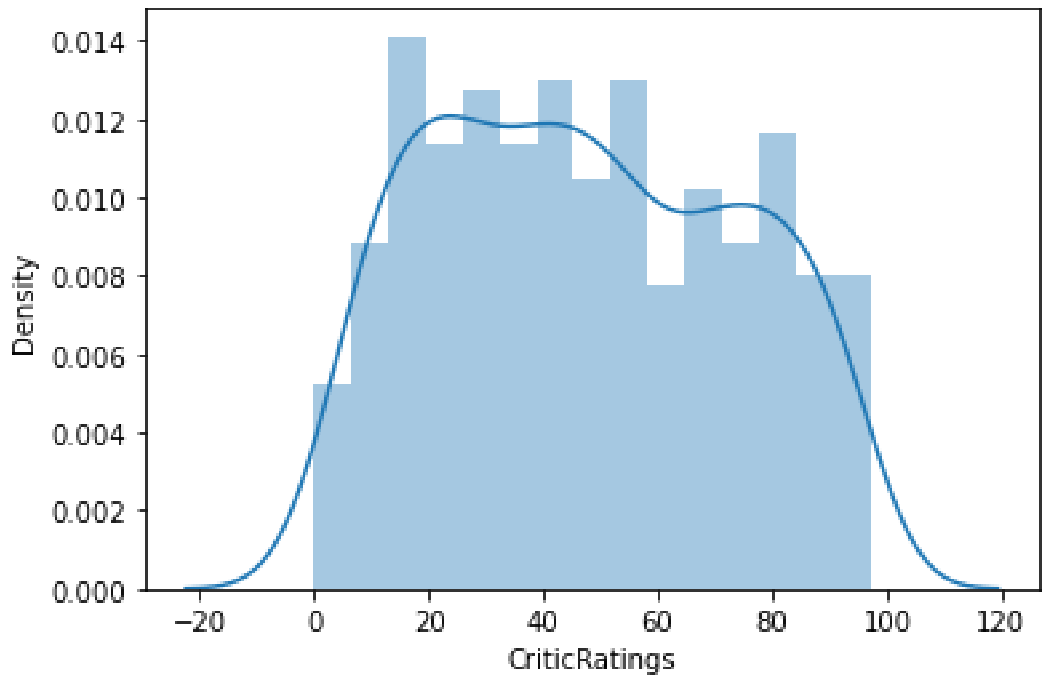
[23]:
```
# Chart 1
```

1 Histograms

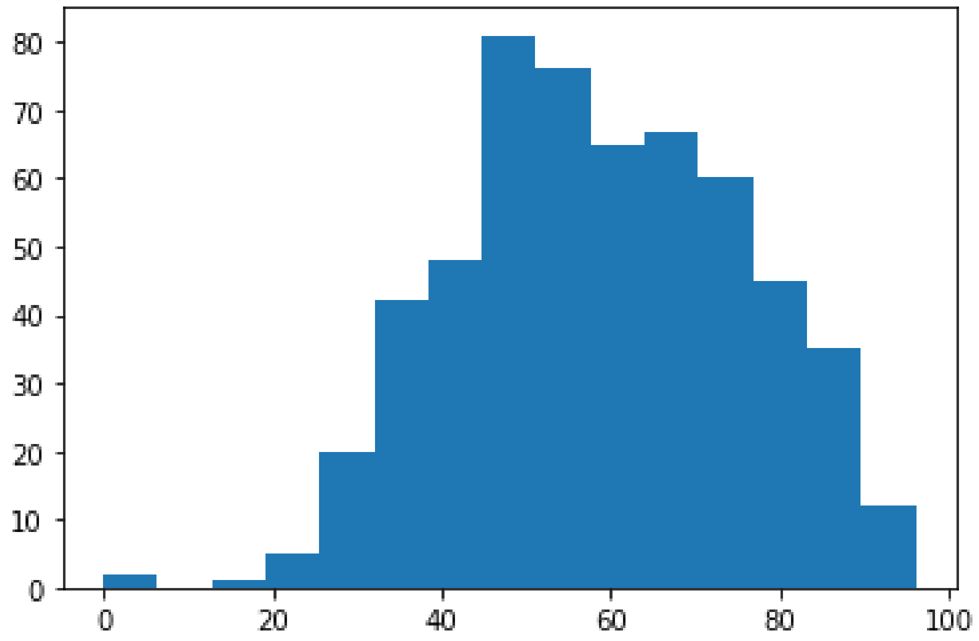
[24]:
```
m1=sns.distplot(movies.AudienceRatings,bins=15)
```



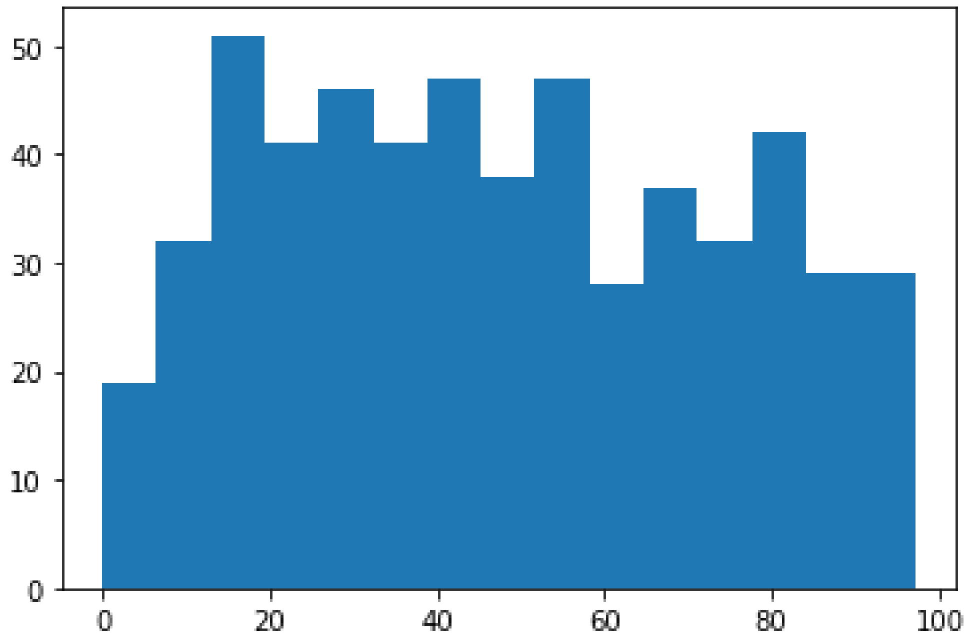
[25]:
```
m2=sns.distplot(movies.CriticRatings,bins=15)
```



[26]:
```
# chart 2
n1 = plt.hist(movies.AudienceRatings,bins=15)
```



[27]:
```
n2 = plt.hist(movies.CriticRatings,bins=15)
```



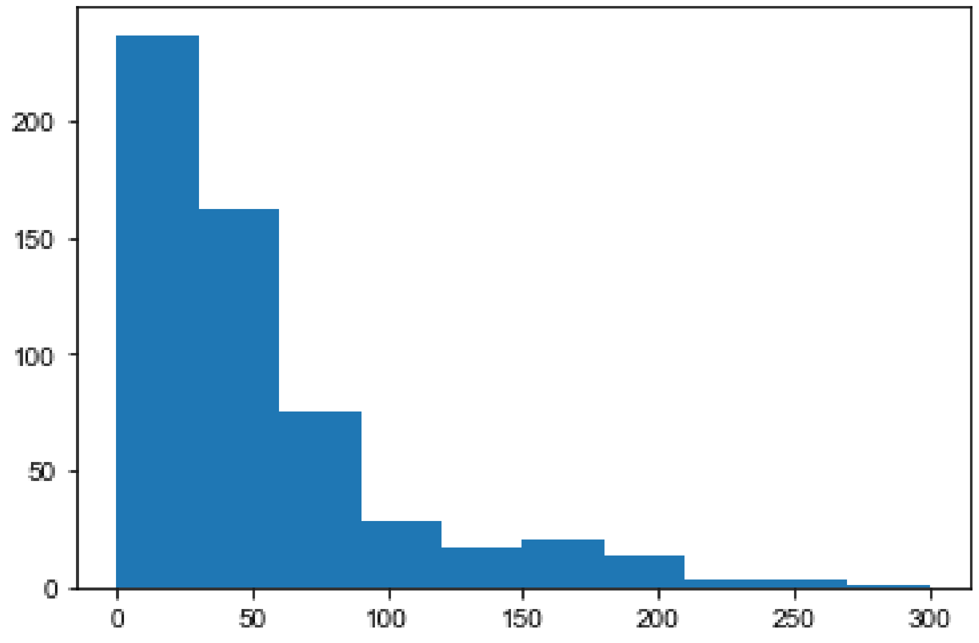
2 Stacked Histograms

[28]:
```
movies.columns=['Film', 'Genre', 'CriticRatings', 'AudienceRatings',
       'BudgetMillion', 'Year']
```

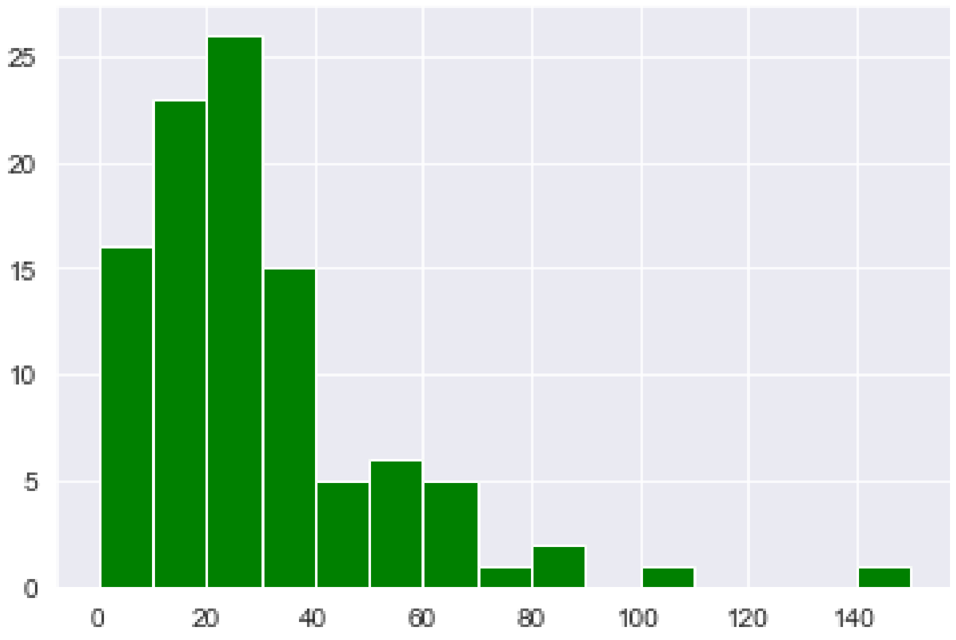
[29]:
```
movies.head()
```

[29]:
```
                   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
```

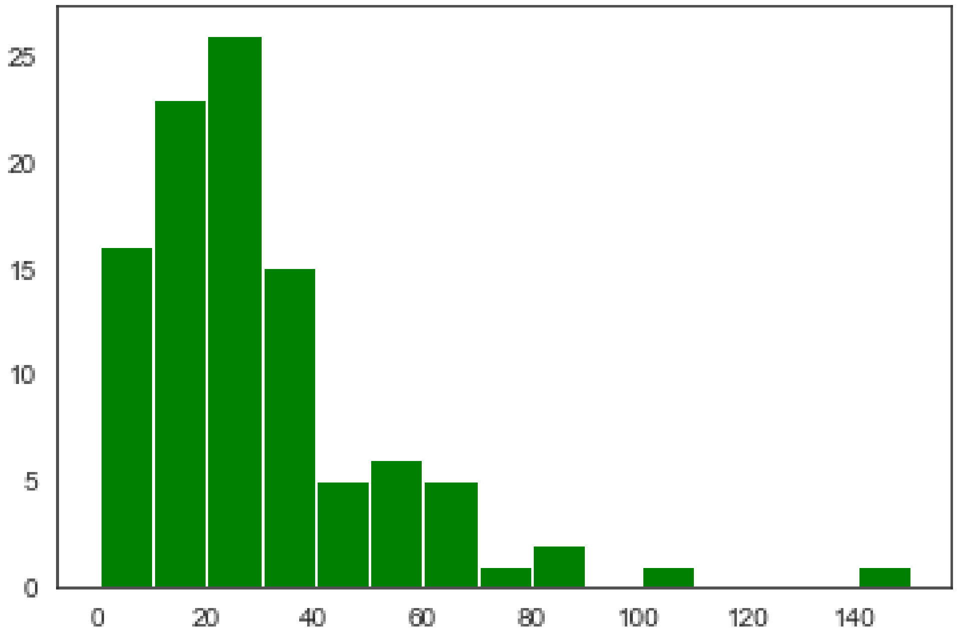
[30]:
```
movies[movies.Genre=='Comedy'] # Filter
plt.hist(movies.BudgetMillion)
sns.set_style("darkgrid")
plt.show()
```



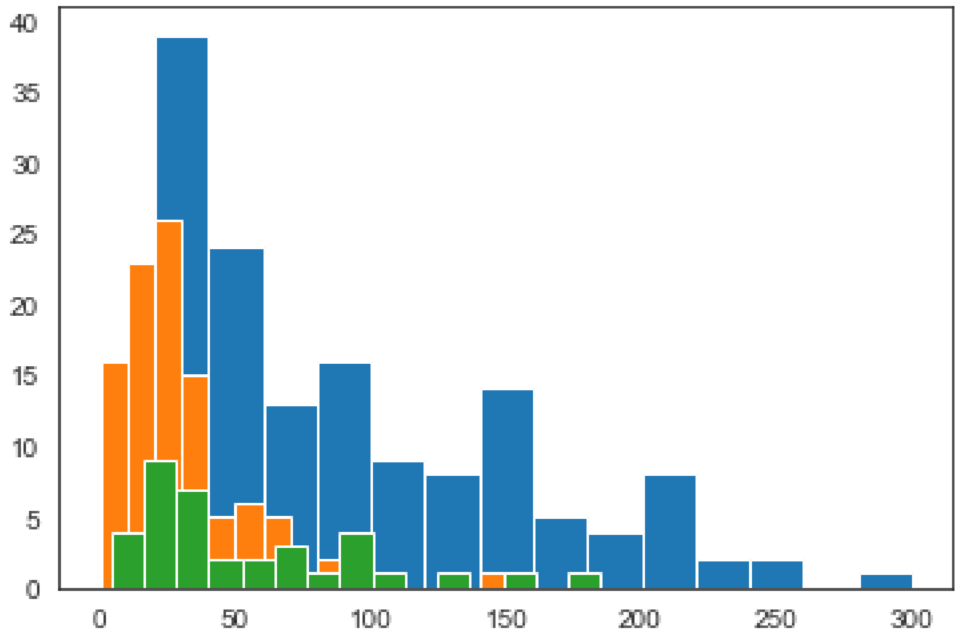
[31]:
```
plt.hist(movies[movies.Genre=='Drama'].BudgetMillion,bins=15,color='Green')
plt.show()
```



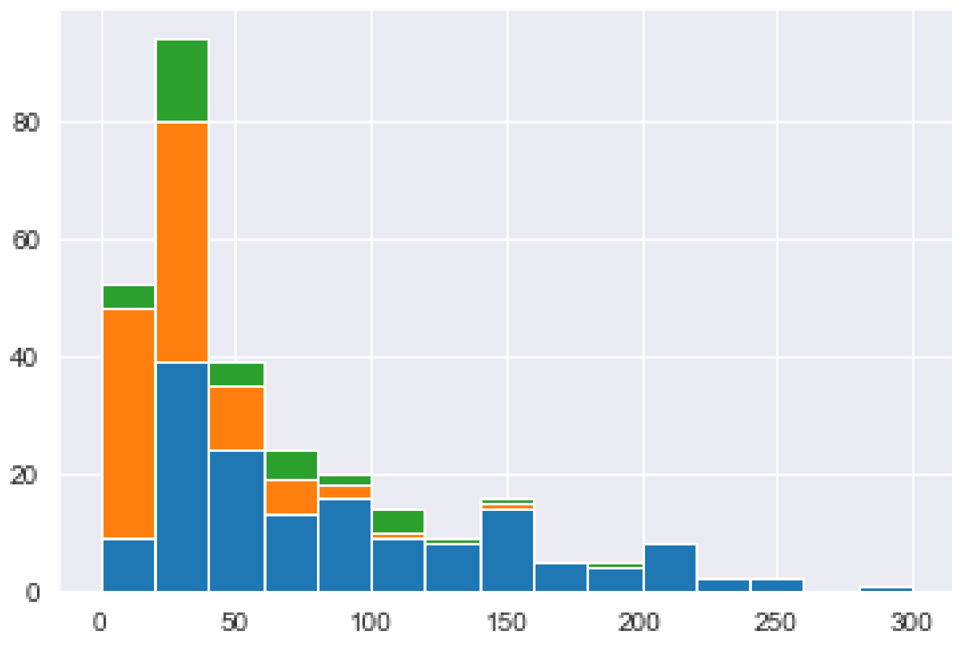
[32]:
```
sns.set_style("white")
plt.hist(movies[movies.Genre=='Drama'].BudgetMillion,bins=15,color='Green')
plt.show()
```



[33]:
```
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)
sns.set_style("darkgrid")
plt.show()
```



[34]:
```
# OR

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



[35]:
```
# OR

movies.Genre.cat.categories
```

[35]:
```
Index(['Action', 'Adventure', 'Comedy', 'Drama', 'Horror', 'Romance',
       'Thriller'],
      dtype='object')
```

[36]:
```
for gen in movies.Genre.cat.categories:
    print(gen)
```

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

[37]:
```
list1=list([])
for gen in movies.Genre.cat.categories:
    list1.append(movies[movies.Genre==gen].BudgetMillion)
print(list1)
```

```
[2       20
5      200
15      35
```

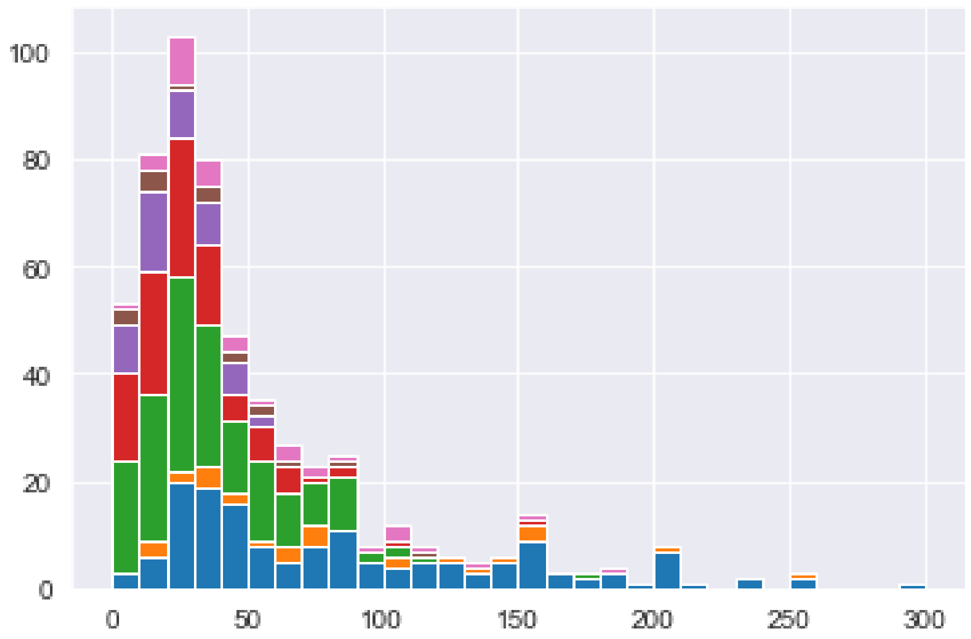
```
29      20
30      20
      …
531    130
542     35
546    150
547    160
557     24
Name: BudgetMillion, Length: 154, dtype: int64, 1      105
3       18
19     200
21      45
24      40
32      78
46      20
65      38
68     140
130     73
165     12
166    125
167    250
168    150
176     36
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
```

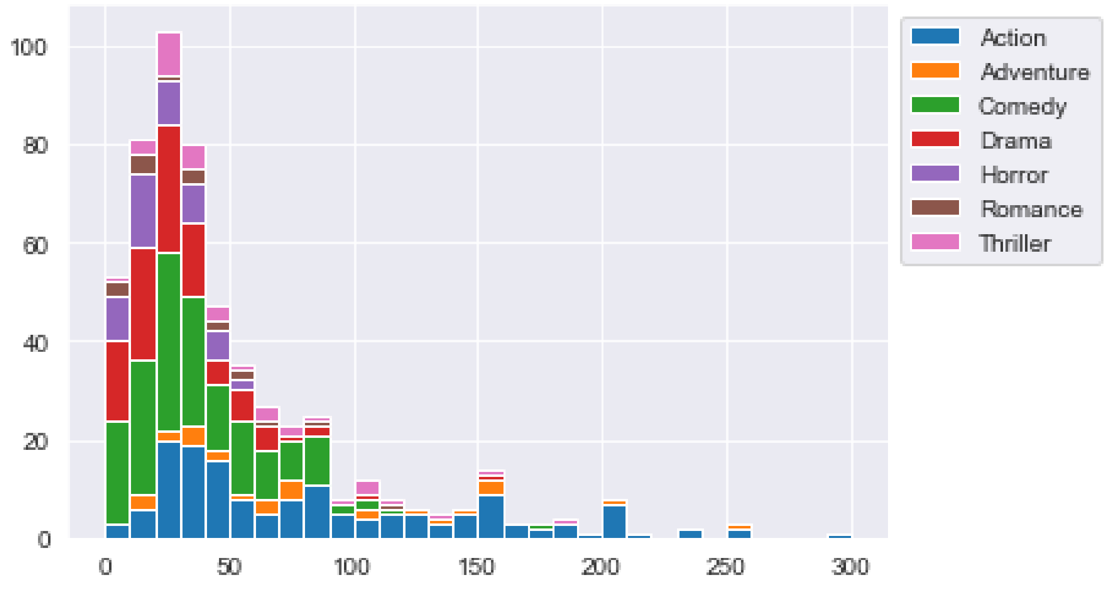
```
416    40
426     5
429    15
453    18
461    40
462    37
464    16
465    25
475     9
478    38
486    16
521    10
Name: BudgetMillion, dtype: int64, 16       45
42      17
78      50
108     60
136     35
201      0
208     80
244     17
250     20
255     40
266     56
284     15
290     30
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]
```

[38]:
```
list1=list([])
mylabel=list([])
for gen in movies.Genre.cat.categories:
    list1.append(movies[movies.Genre==gen].BudgetMillion)
h=plt.hist(list1, bins=30,stacked=True,rwidth=1)
```



[39]:
```
list1=list([])
mylabel=list([])
for gen in movies.Genre.cat.categories:
    list1.append(movies[movies.Genre==gen].BudgetMillion)
    mylabel.append(gen)
h=plt.hist(list1, bins=30,stacked=True,rwidth=1,label=mylabel)
plt.legend()
plt.legend(loc='upper left',bbox_to_anchor=(1,1))
plt.show()
```



[40]:
```
# <<< chart 4
```

---

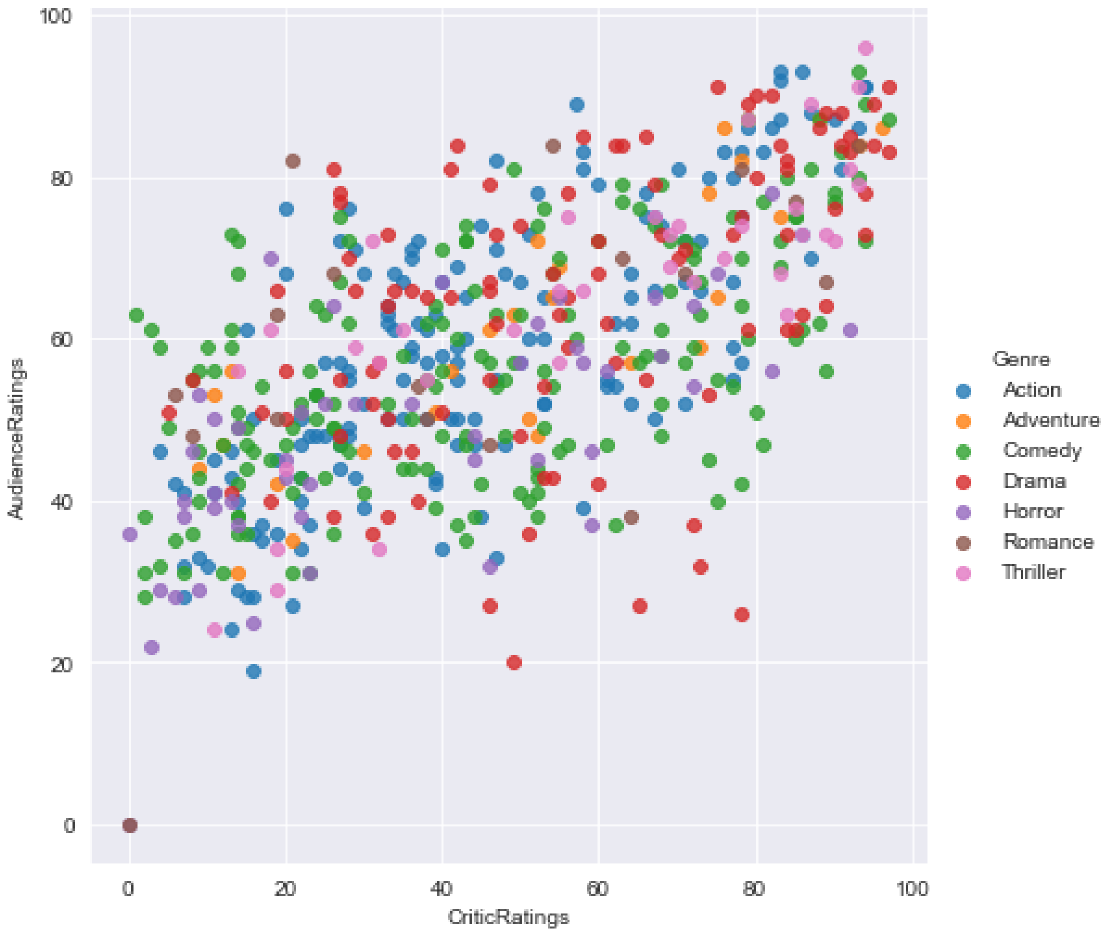
3 KDE Plot

[41]:
```
movies.head()
```

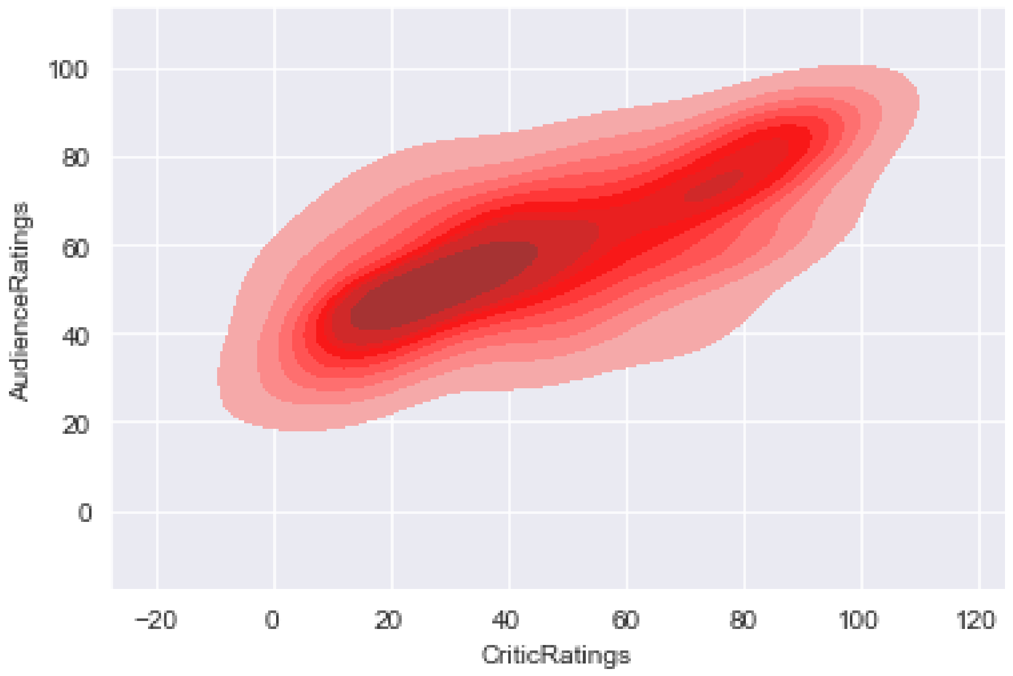
[41]:
```
                   Film      Genre  CriticRatings  AudienceRatings  \
0  (500) Days of Summer     Comedy             87               81
1            10,000 B.C.  Adventure              9               44
2             12 Rounds     Action             30               52
3            127 Hours  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
```

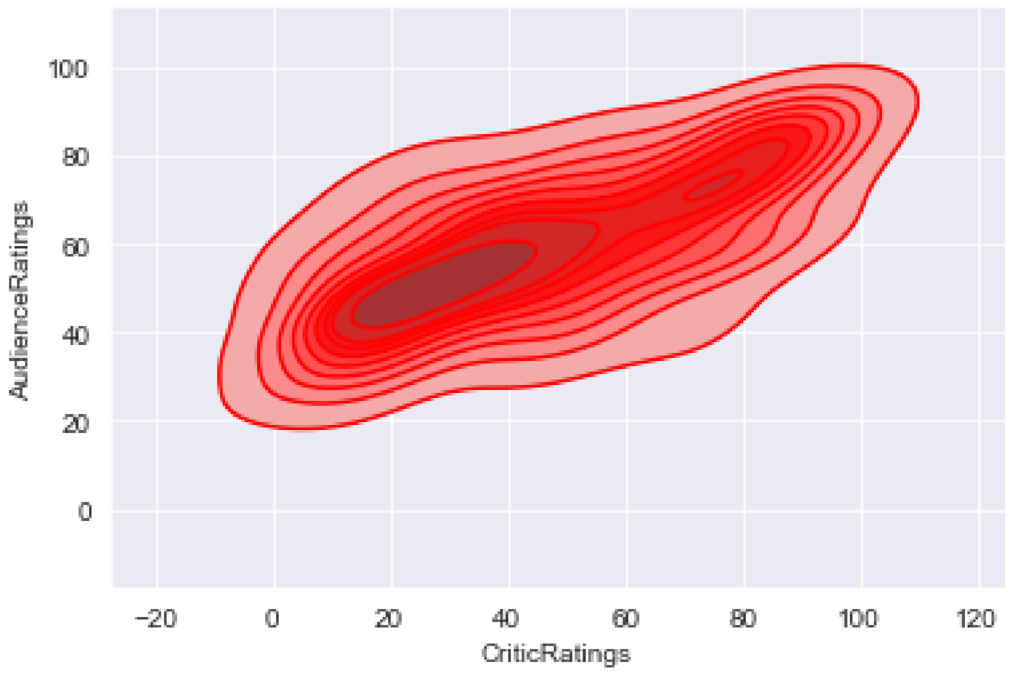
[42]:
```
sns.
 ↪lmplot(data=movies,x='CriticRatings',y='AudienceRatings',fit_reg=False,hue='Genre',size=6,a
plt.show()
```



[43]:
```
k1=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings, \
               shade=True,shade_lowest=False,color='Red')
```



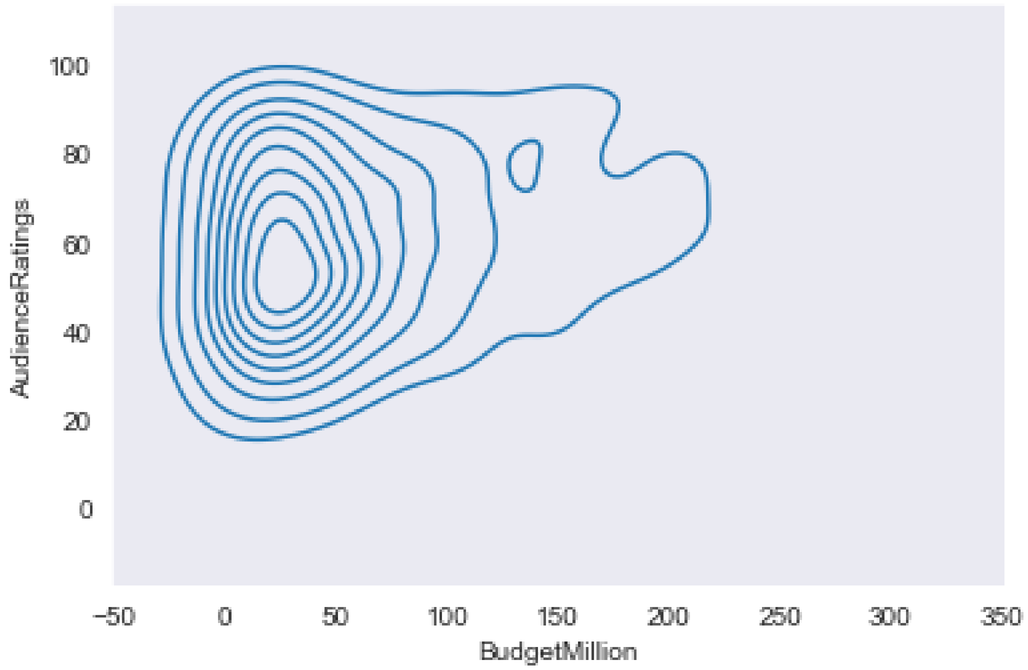
```
[44]: k1=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings, \
                     shade=True,shade_lowest=False,color='Red')

      k1=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings, \
                     color='Red')
```



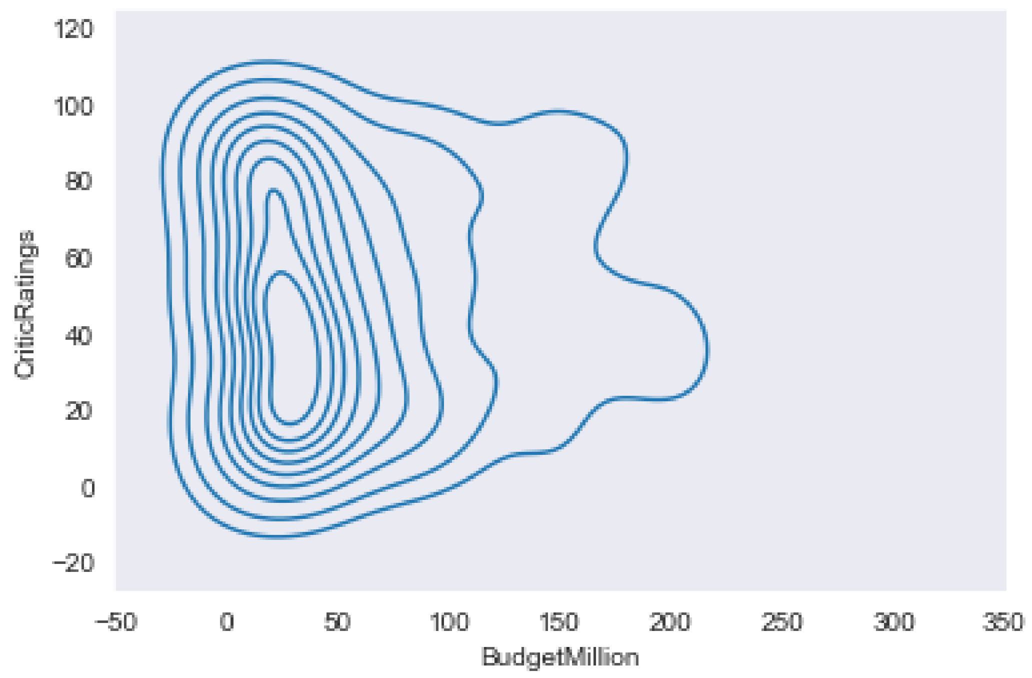
4 working with Subplots()

[45]:
```
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

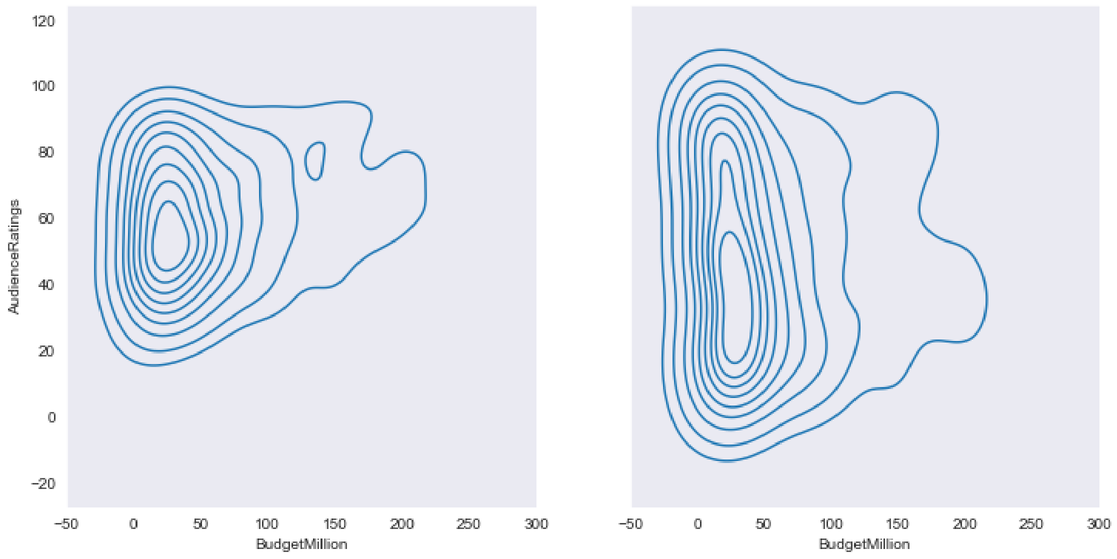
[46]:
```
sns.set_style('dark')
k1=sns.kdeplot(movies.BudgetMillion,movies.AudienceRatings)
plt.show()
```



[47]:
```
k2=sns.kdeplot(movies.BudgetMillion,movies.CriticRatings)
plt.show()
```

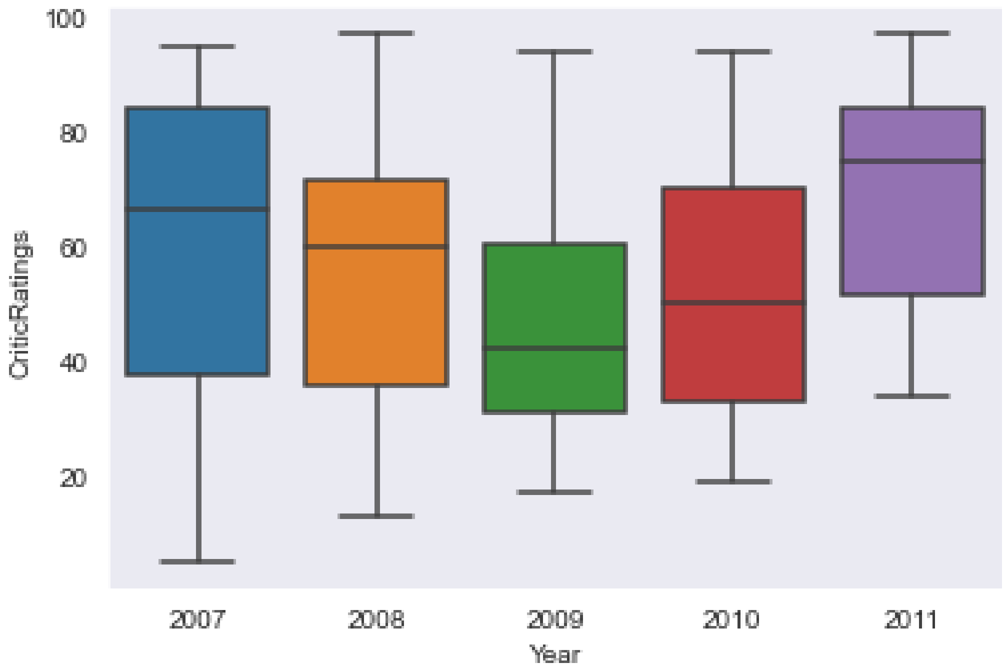


[48]:
```
f, axes=plt.subplots(1,2, figsize=(12,6),sharex=True,sharey=True)
k1=sns.kdeplot(movies.BudgetMillion,movies.AudienceRatings,ax=axes[0])
k2=sns.kdeplot(movies.BudgetMillion,movies.CriticRatings,ax=axes[1])
k1.set(xlim=(-50,300))
plt.show()
```

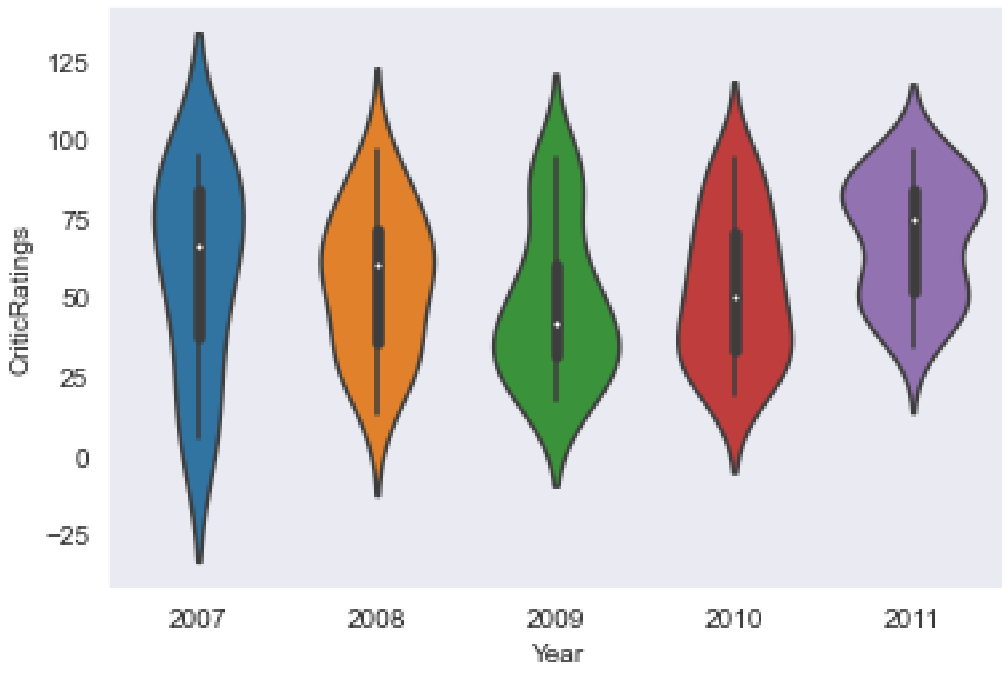


5 Violinplots Vs Boxplots

[49]:
```
w=sns.boxplot(data=movies[movies.Genre=='Drama'],x='Year',y='CriticRatings')
# w=sns.boxplot(data=movies,x='Genre',y='CriticRatings')
```

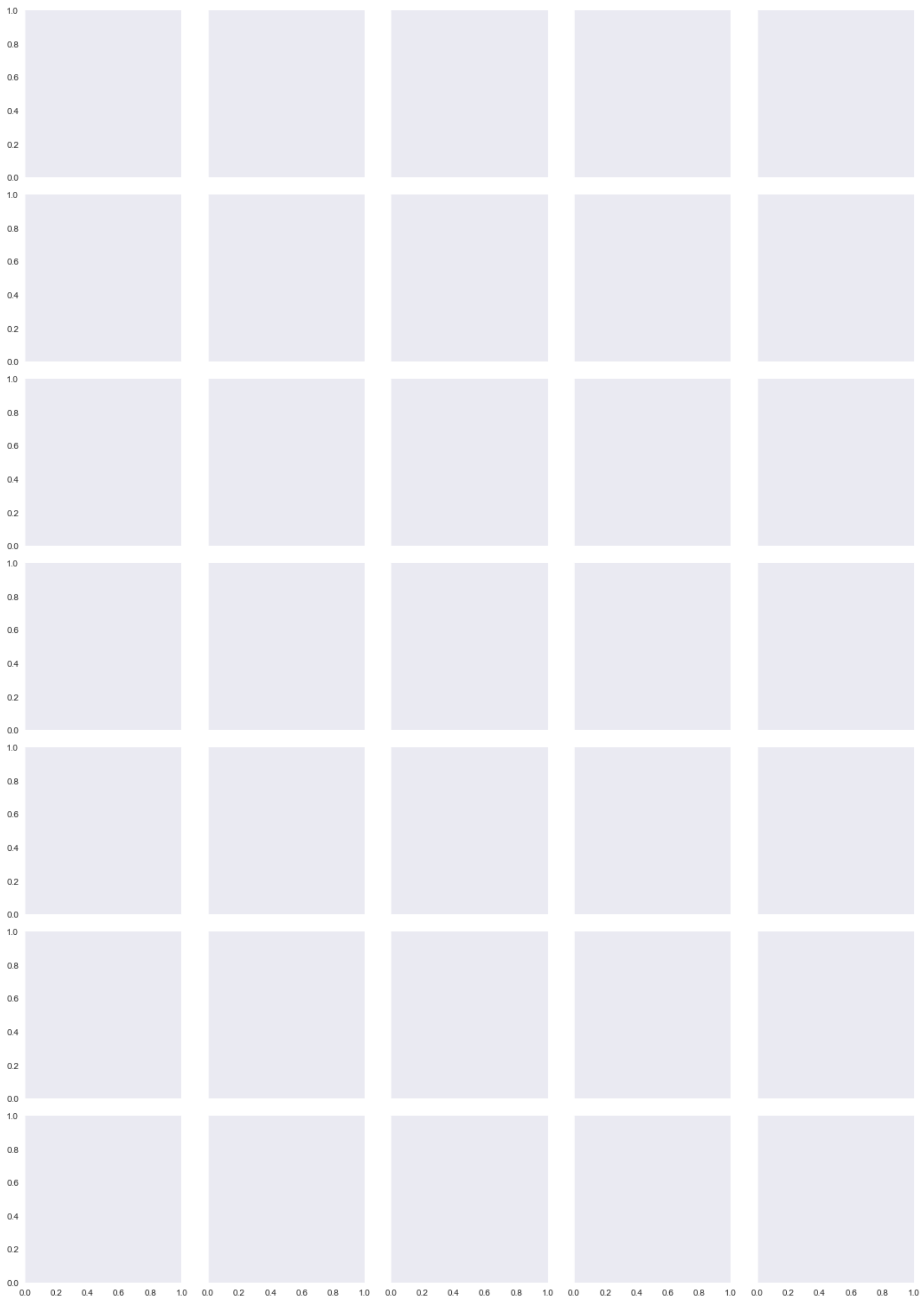


[50]:
```
z=sns.violinplot(data=movies[movies.Genre=='Drama'],x='Year',y='CriticRatings')
# z=sns.violinplot(data=movies,x='Genre',y='CriticRatings')
```

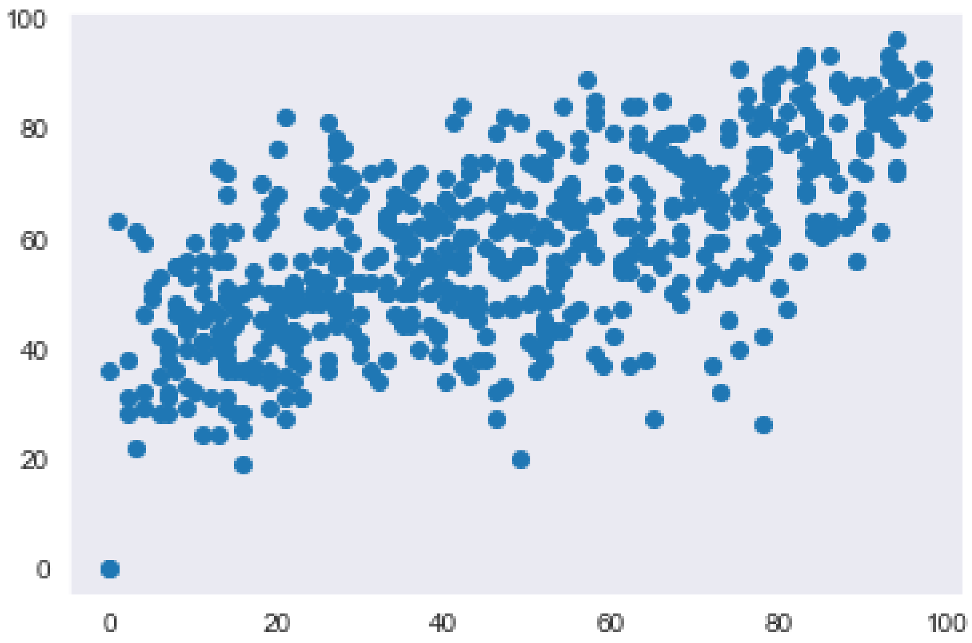


6 Creating a Facet grid

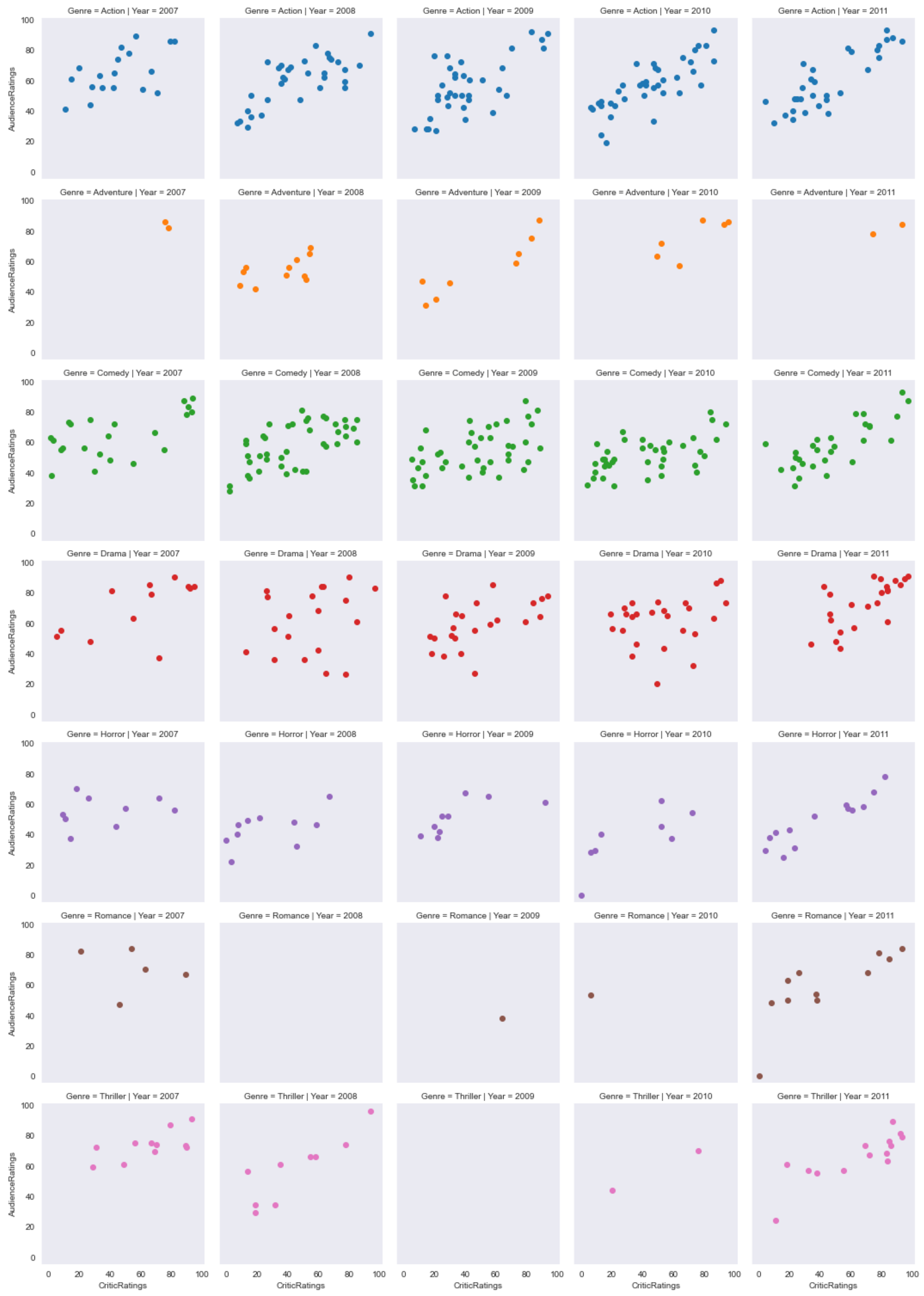
```
[51]: # g=sns.FacetGrid(movies,row='Genre',hue='Genre')
      g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
```



[52]:
```
# g=g.map()
plt.scatter(movies.CriticRatings,movies.AudienceRatings)
plt.show()
```

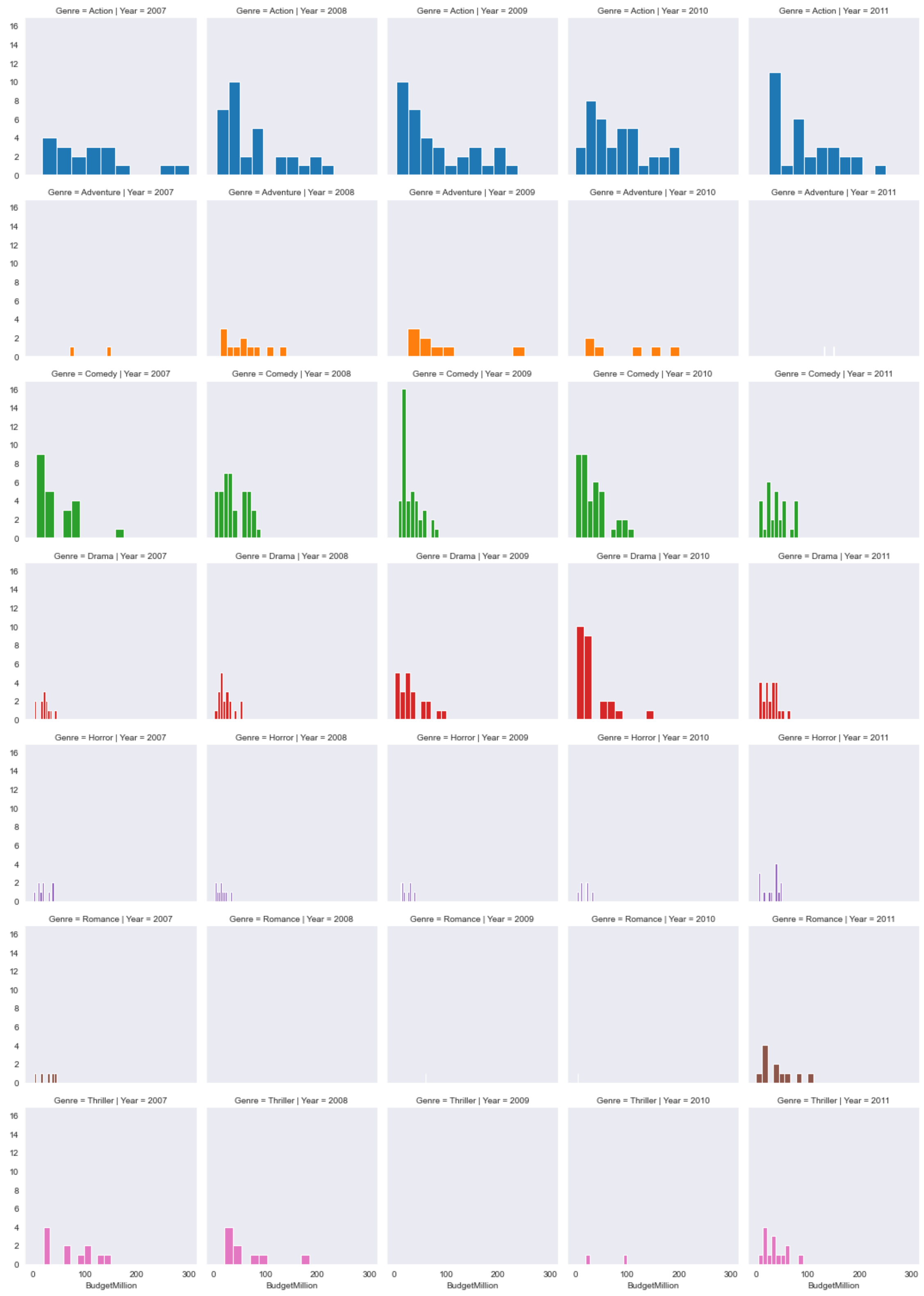


[53]:
```
g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
g=g.map(plt.scatter,'CriticRatings','AudienceRatings')
plt.show()
```

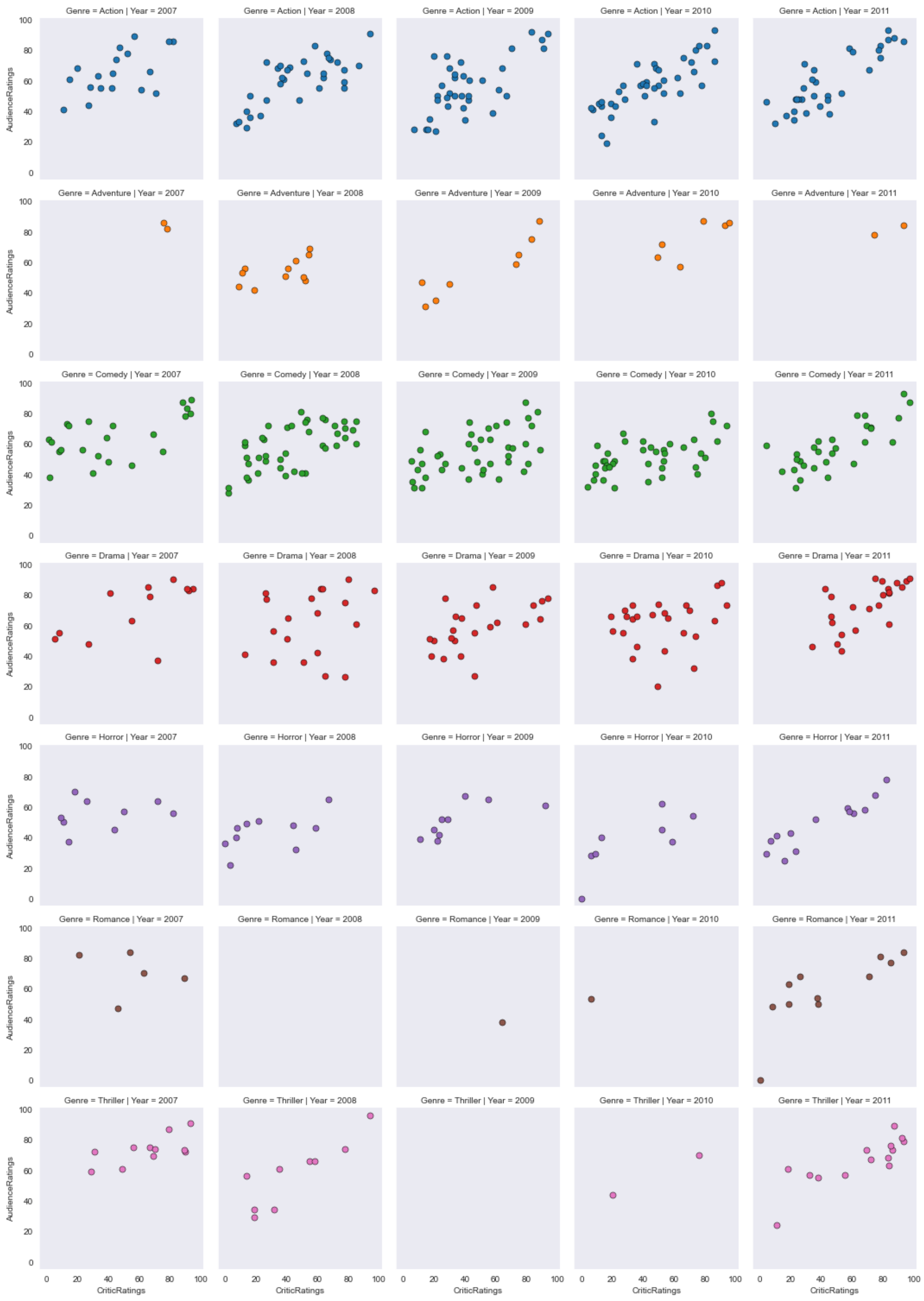

Genre = Action | Year = 2007
Genre = Action | Year = 2008
Genre = Action | Year = 2009
Genre = Action | Year = 2010
Genre = Action | Year = 2011
Genre = Adventure | Year = 2007
Genre = Adventure | Year = 2008
Genre = Adventure | Year = 2009
Genre = Adventure | Year = 2010
Genre = Adventure | Year = 2011
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
CriticRatings
0
20
40
60
80
100

7 We can papulate with any kind of chart. (Ex. Histograms)

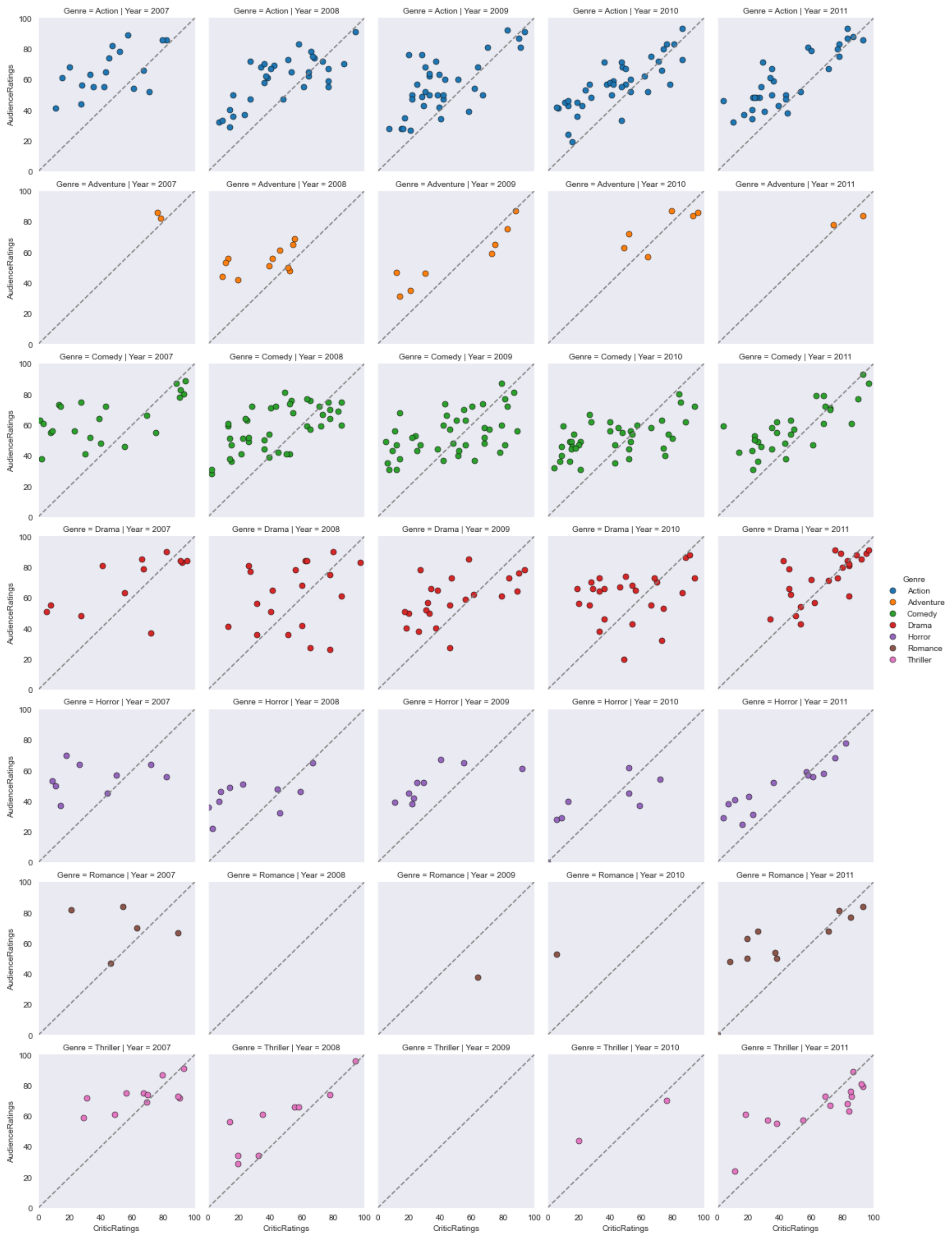
[54]:
```
g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
g=g.map(plt.hist,'BudgetMillion')
plt.show()
```


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
0
2
4
6
8
10
12
14
16
0
100
200
300
BudgetMillion

[55]:
```
# Back on
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)
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
CriticRatings
0
20
40
60
80
100

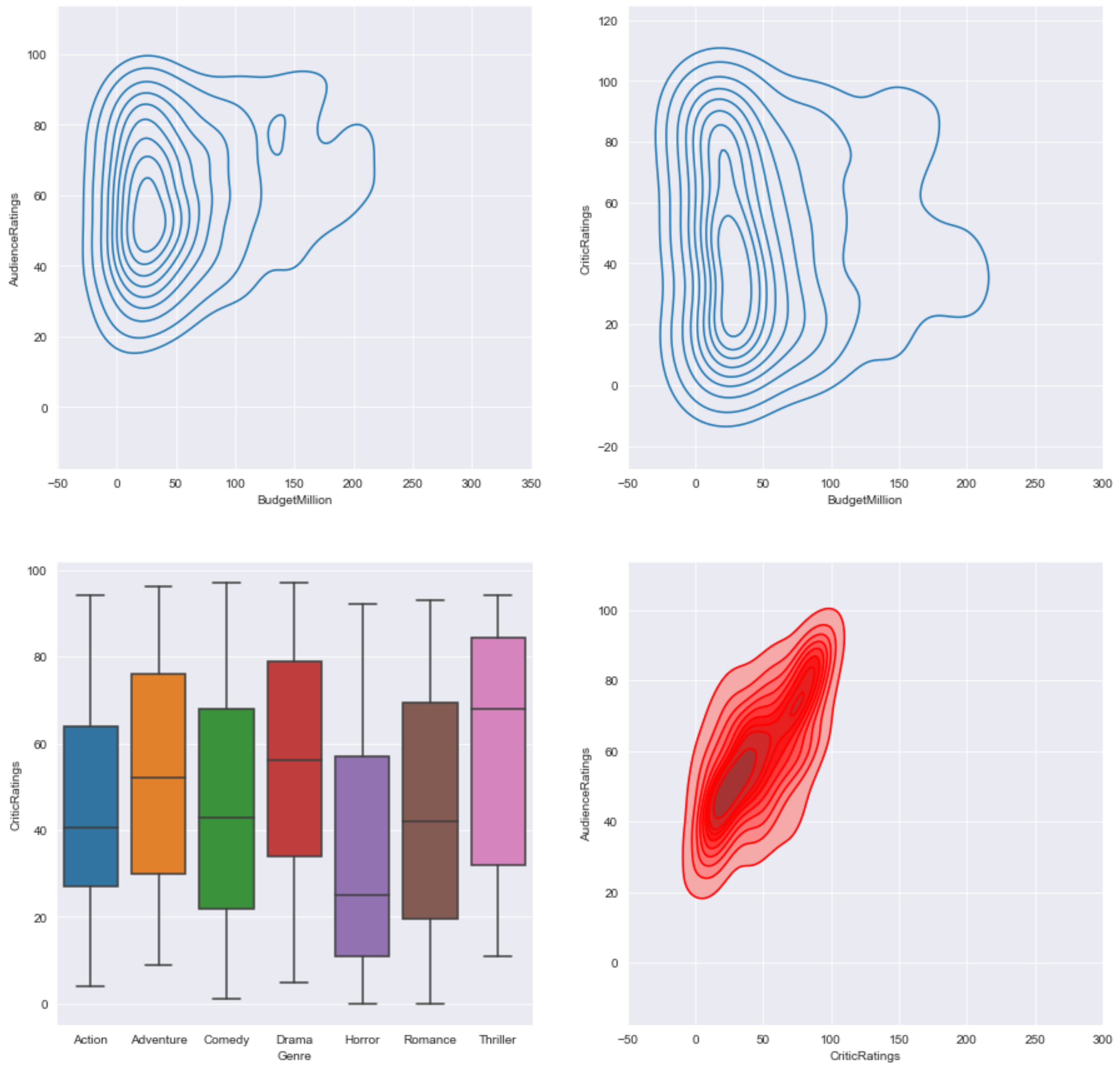
[56]:
```
g=sns.FacetGrid(movies,row='Genre',col='Year',hue='Genre')
kws=dict(s=50, linewidth=0.5,edgecolor='black')
g=g.map(plt.scatter,'CriticRatings','AudienceRatings',**kws)
g.set(xlim=(0,100),ylim=(0,100))
for ax in g.axes.flat:
    ax.plot((0,100),(0,100),c='gray',ls='--')
g.add_legend()
plt.show()
```


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
CriticRatings
0
20
40
60
80
100
Genre
Action
Adventure
Comedy
Drama
Horror
Romance
Thriller

8 Creating a Dashboard

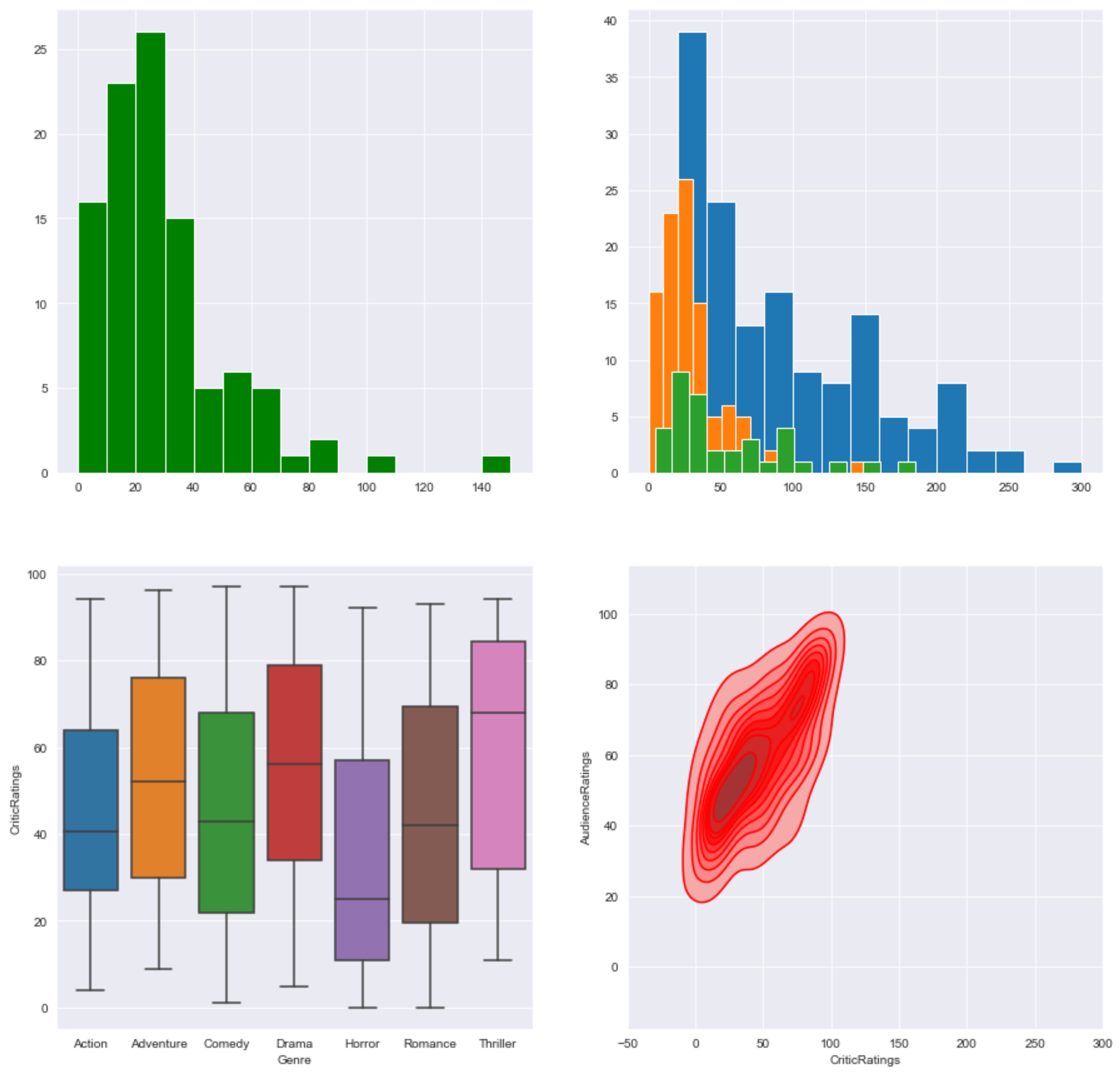
[57]:
```
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

[58]:
```
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])
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])
kw1=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings, \
              color='Red',ax=axes[1,1])
k1.set(xlim=(-50,300))
k2.set(xlim=(-50,300))
plt.show()
```



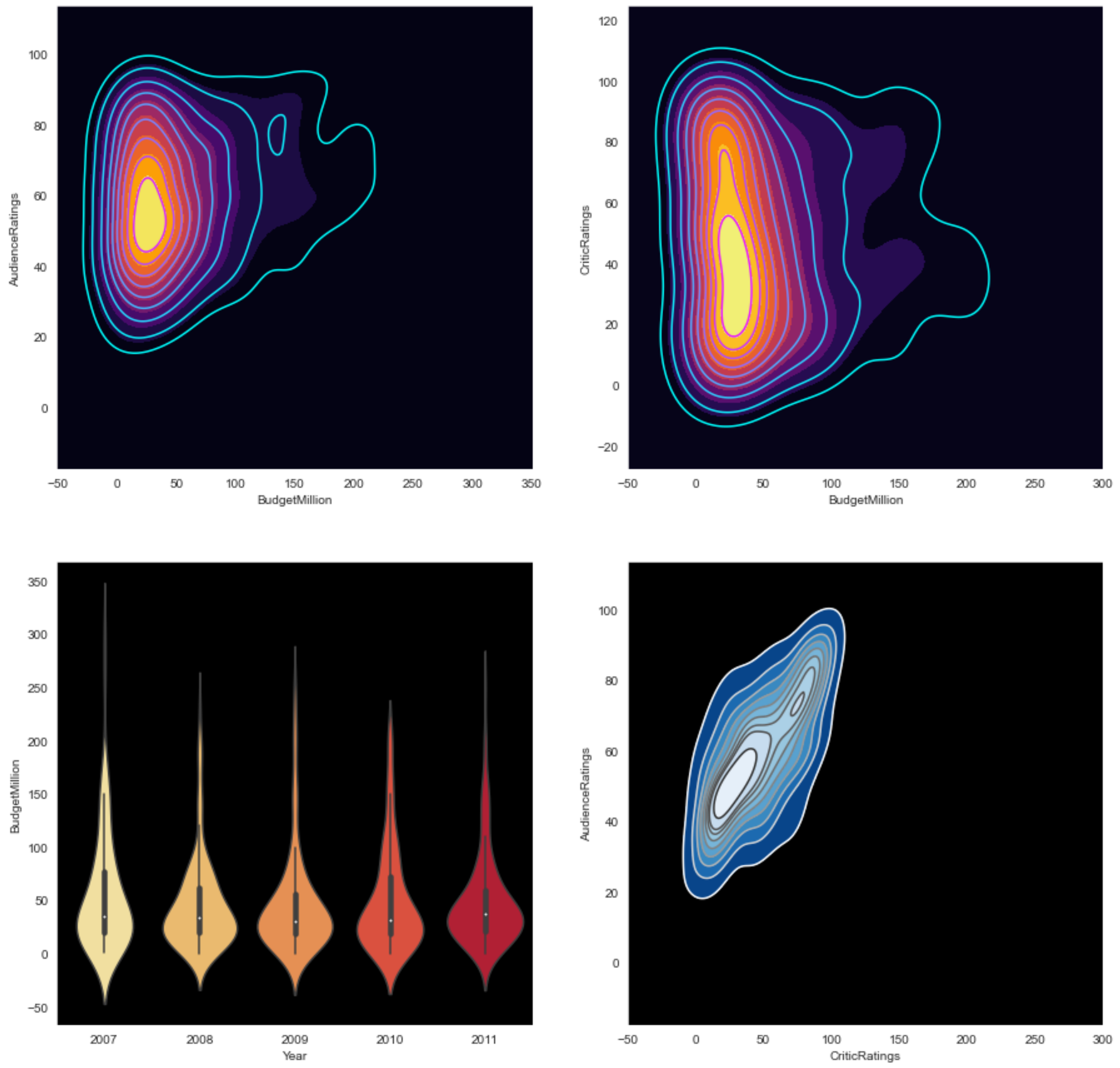
```
[59]: sns.set_style("darkgrid")
      f, axes=plt.subplots(2,2,figsize=(15,15))
      #k1=sns.kdeplot(movies.BudgetMillion,movies.AudienceRatings,ax=axes[0,0])
      #k2=sns.kdeplot(movies.BudgetMillion,movies.CriticRatings,ax=axes[0,1])
      axes[0,0].hist(movies[movies.Genre=='Drama'].
       ↪BudgetMillion,bins=15,color='Green') # as matplotlib

      axes[0,1].hist(movies[movies.Genre=='Action'].BudgetMillion,bins=15)
      axes[0,1].hist(movies[movies.Genre=='Drama'].BudgetMillion,bins=15)
      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])
kw1=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings, \
              color='Red',ax=axes[1,1])
k1.set(xlim=(-50,300))
k2.set(xlim=(-50,300))
plt.show()
```

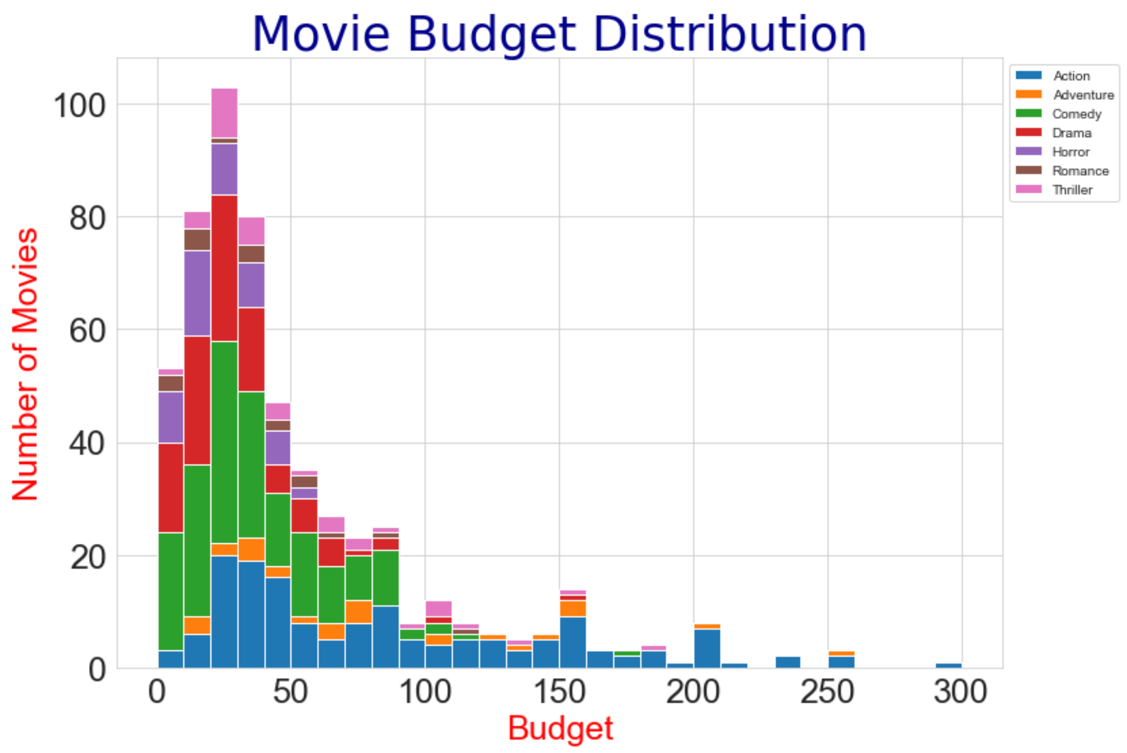


[60]:
```
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

[61]:
```
sns.set_style("dark",{"axes.facecolor":"black"}) # white, whitegrid, dark,␣
 ↪darkgrid, ticks
f, axes=plt.subplots(2,2,figsize=(15,15))
# plot[0,0]
k1=sns.kdeplot(movies.BudgetMillion,movies.AudienceRatings,\
              shade=True,shade_lowest=True,cmap='inferno',ax=axes[0,0])

k1b=sns.kdeplot(movies.BudgetMillion,movies.AudienceRatings,\
              cmap='cool',ax=axes[0,0])
# plot[0,1]
k2=sns.kdeplot(movies.BudgetMillion,movies.CriticRatings,\
               shade=True,shade_lowest=True,cmap='inferno',ax=axes[0,1])
k2b=sns.kdeplot(movies.BudgetMillion,movies.CriticRatings,ax=axes[0,1],
               cmap='cool')
# plot[1,0]
w=sns.violinplot(data=movies,x='Year',y='BudgetMillion',\
              palette='YlOrRd',ax=axes[1,0])
# plot[1,1]
k1=sns.kdeplot(movies.CriticRatings,movies.AudienceRatings, \
              shade=True,shade_lowest=False,cmap='Blues_r',ax=axes[1,1])
kw1=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()
```



[80]:
```
list1=list([])
mylabel=list([])
for gen in movies.Genre.cat.categories:
    list1.append(movies[movies.Genre==gen].BudgetMillion)
    mylabel.append(gen)

sns.set_style("whitegrid")
fig, ax=plt.subplots()
fig.set_size_inches(11.7,8.27) # A4 size paper
h=plt.hist(list1, bins=30,stacked=True,rwidth=1,label=mylabel)
plt.title("Movie Budget Distribution" ,fontsize=35,␣
 ↪color='darkblue',fontname='console')
plt.ylabel("Number of Movies",fontsize=25,color='Red')
plt.xlabel("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()
```



```
[ ]:
```

```
[ ]:
```

```
[ ]:
```