Movie_Rating

November 25, 2025

[95]:
```
import pandas as pd
```

[96]:
```
movies = pd.read_csv(r'/Users/mahidharreddy/Downloads/Movie-Rating.csv')
```

[97]:
```
movies
```

[97]:
```
                       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]
```

[98]:
```
type(movies)
```

[98]:
```
pandas.core.frame.DataFrame
```

[99]:
```
movies.columns
```

[99]:
```
Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
       'Budget (million $)', 'Year of release'],
      dtype='object')
```

[100]:
```
len(movies)
```

[100]:
```
559
```

[101]:
```
import numpy
print(numpy.__version__)
```

```
2.3.5
```

[102]:
```
import pandas as pd
print(pd.__version__)
```

```
2.3.3
```

[103]:
```
movies.columns
```

[103]:
```
Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
       'Budget (million $)', 'Year of release'],
      dtype='object')
```

[104]:
```
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
```

[105]:
```
movies.shape #dimensions which mean rows * columns
```

[105]:
```
(559, 6)
```

[106]:
```
movies.head()
```

[106]:
```
                    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
```

[107]:
```
movies.tail()
```

[107]:
```
                Film     Genre  Rotten Tomatoes Ratings %  Audience Ratings %  \
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

     Budget (million $)  Year of release
554                  50             2011
555                  18             2009
556                  65             2007
557                  24             2009
558                  80             2011
```

[108]:
```
movies.columns
```

[108]:
```
Index(['Film', 'Genre', 'Rotten Tomatoes Ratings %', 'Audience Ratings %',
       'Budget (million $)', 'Year of release'],
      dtype='object')
```

[109]:
```
movies.columns = ['Film', 'Genre', 'CriticRating', 'AudienceRatings',
       'BudgetMillions', 'Year']
```

[110]:
```
movies.columns
```

[110]:
```
Index(['Film', 'Genre', 'CriticRating', 'AudienceRatings', 'BudgetMillions',
       'Year'],
      dtype='object')
```

[111]:
```
movies.head(1)
```

[111]:
```
                    Film   Genre  CriticRating  AudienceRatings  \
0  (500) Days of Summer   Comedy            87               81

   BudgetMillions  Year
0               8  2009
```

[112]:
```
movies.describe()
```

[112]:

	CriticRating	AudienceRatings	BudgetMillions	Year
count	559.000000	559.000000	559.000000	559.000000
mean	47.309481	58.744186	50.236136	2009.152057
std	26.413091	16.826887	48.731817	1.362632
min	0.000000	0.000000	0.000000	2007.000000
25%	25.000000	47.000000	20.000000	2008.000000
50%	46.000000	58.000000	35.000000	2009.000000
75%	70.000000	72.000000	65.000000	2010.000000
max	97.000000	96.000000	300.000000	2011.000000

[113]:
```
movies.describe().transpose()
```

[113]:

	count	mean	std	min	25%	50%	75%	max
CriticRating	559.0	47.309481	26.413091	0.0	25.0	46.0	70.0	97.0
AudienceRatings	559.0	58.744186	16.826887	0.0	47.0	58.0	72.0	96.0
BudgetMillions	559.0	50.236136	48.731817	0.0	20.0	35.0	65.0	300.0
Year	559.0	2009.152057	1.362632	2007.0	2008.0	2009.0	2010.0	2011.0

[114]:
```
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   CriticRating     559 non-null    int64
 3   AudienceRatings  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
```

[115]:
```
movies.Film  = movies.Film.astype('category')
```

[116]:
```
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   CriticRating     559 non-null    int64
 3   AudienceRatings  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
```

[117]:
```
movies.Genre = movies.Genre.astype('category')
```

[118]:
```
movies.Year = movies.Year.astype('category')
```

[119]:
```
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   CriticRating     559 non-null    int64
 3   AudienceRatings  559 non-null    int64
 4   BudgetMillions   559 non-null    int64
 5   Year             559 non-null    category
dtypes: category(3), int64(3)
memory usage: 36.5 KB
```

[120]:
```
movies.describe()
```

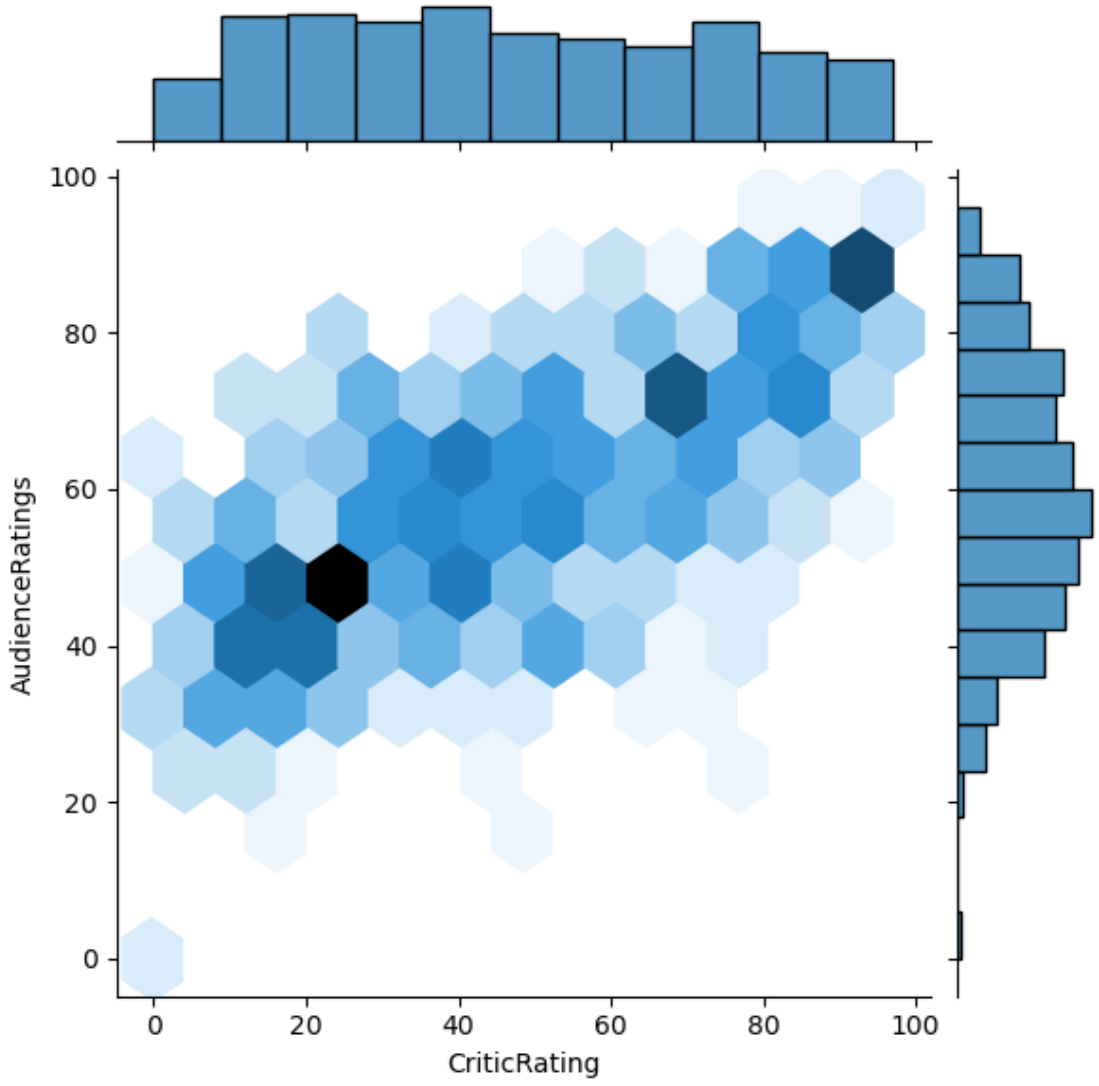
[120]:

	CriticRating	AudienceRatings	BudgetMillions
count	559.000000	559.000000	559.000000
mean	47.309481	58.744186	50.236136
std	26.413091	16.826887	48.731817
min	0.000000	0.000000	0.000000
25%	25.000000	47.000000	20.000000
50%	46.000000	58.000000	35.000000
75%	70.000000	72.000000	65.000000

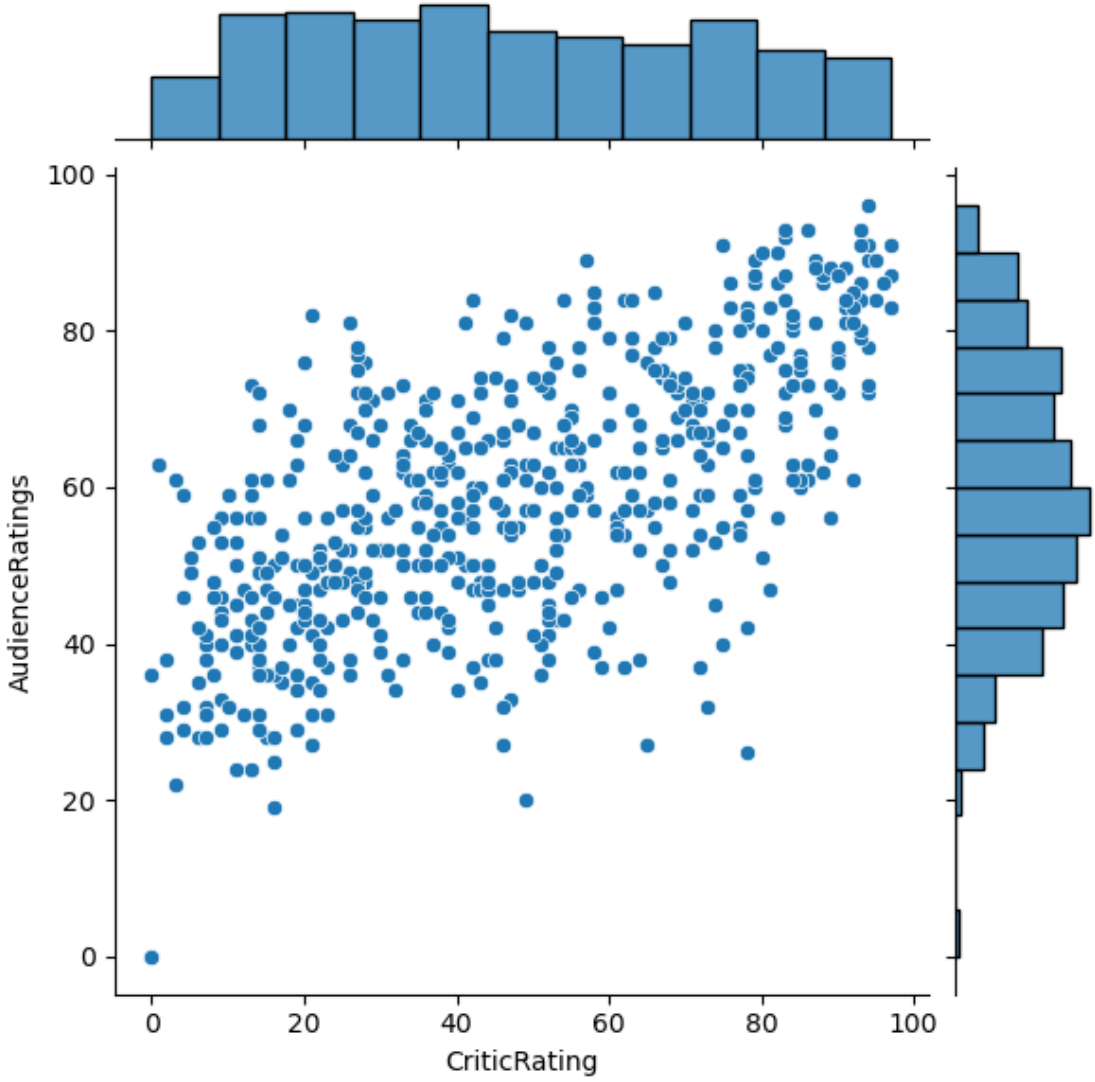
```
max        97.000000          96.000000        300.000000
```

[121]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

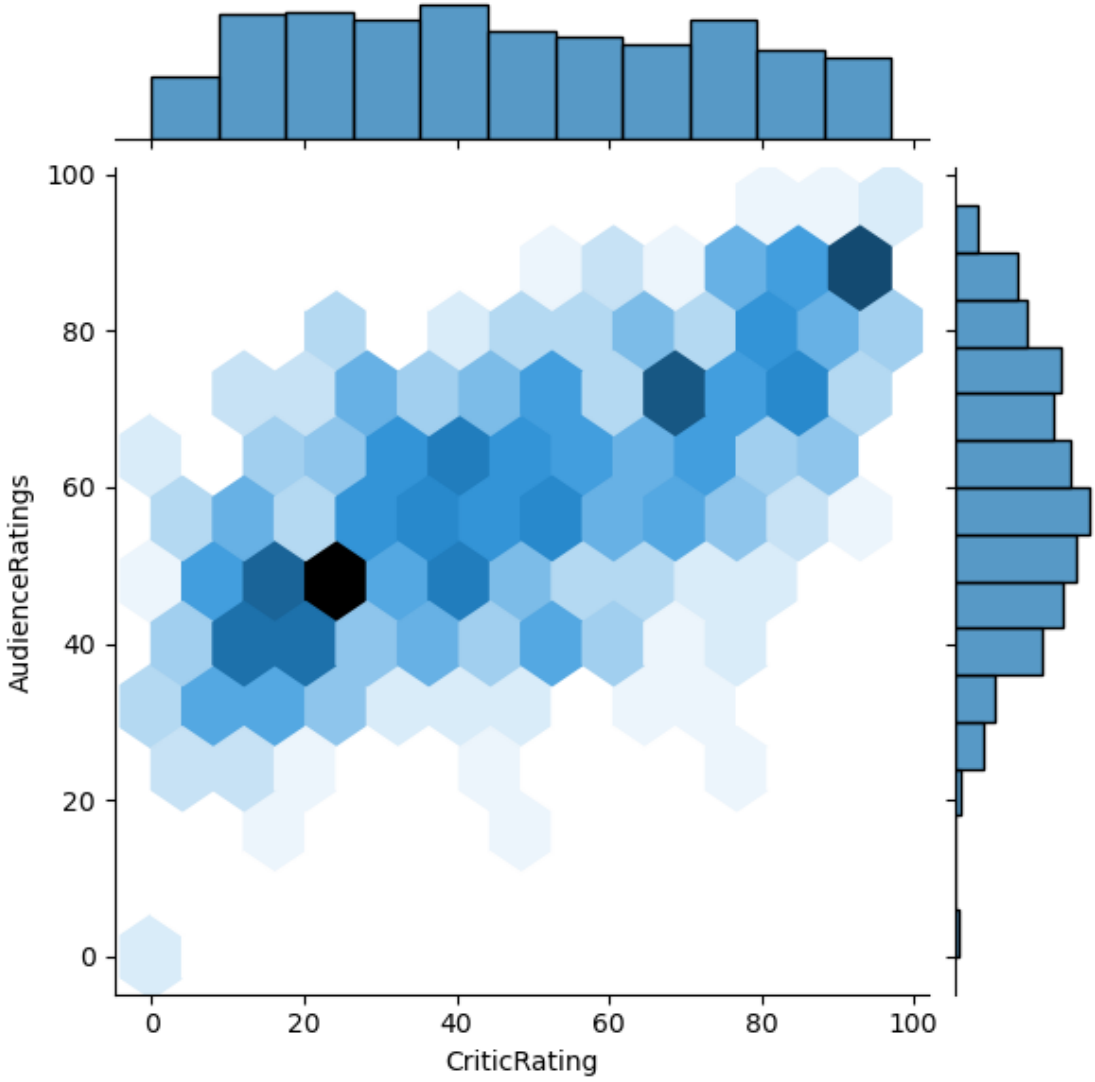
[123]:
```
j = sns.jointplot(data=movies, x='CriticRating', y='AudienceRatings',␣
↪kind='hex')
```



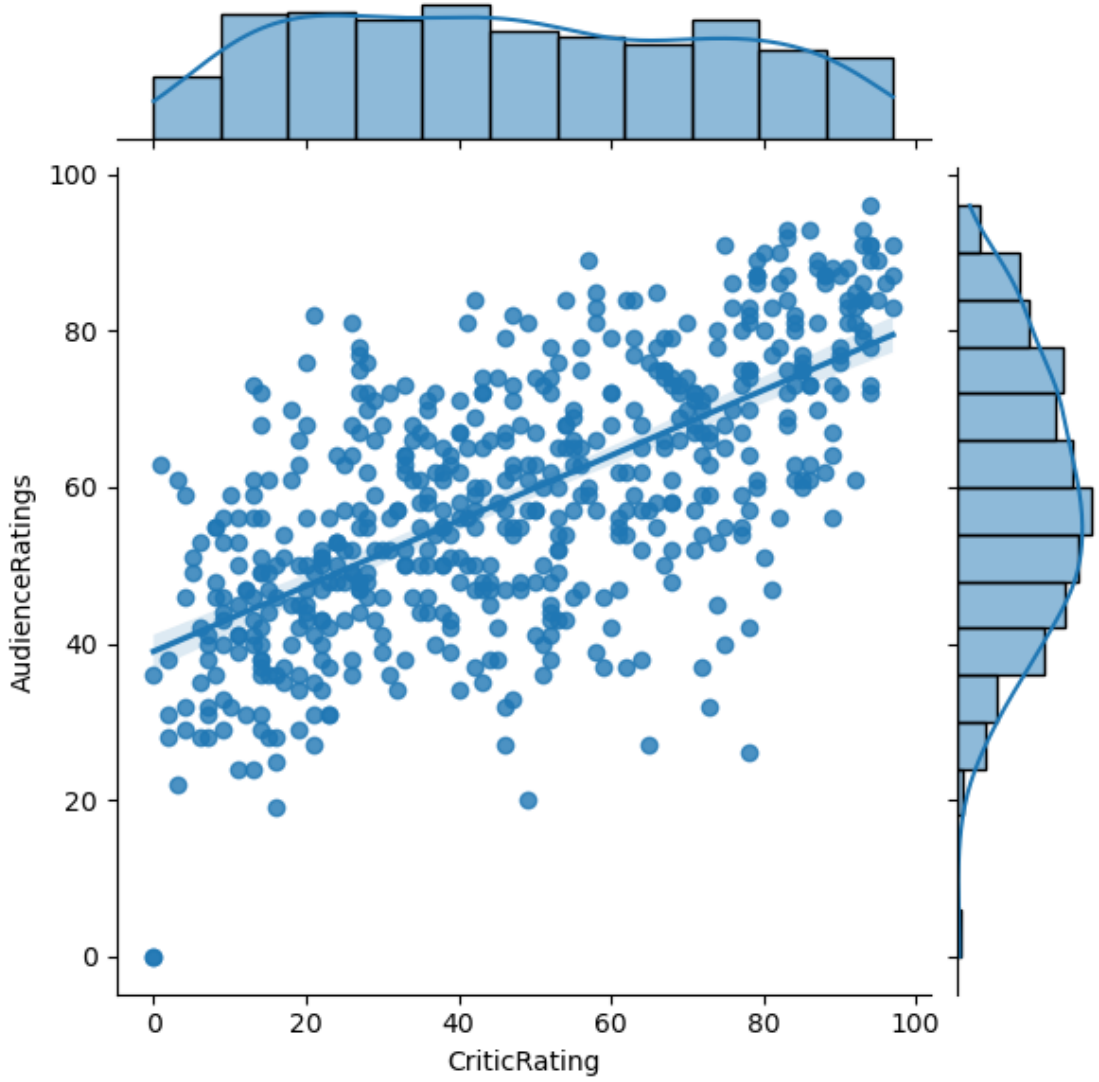
[124]:
```
j = sns.jointplot(data=movies, x='CriticRating', y='AudienceRatings',␣
↪kind='scatter')
```



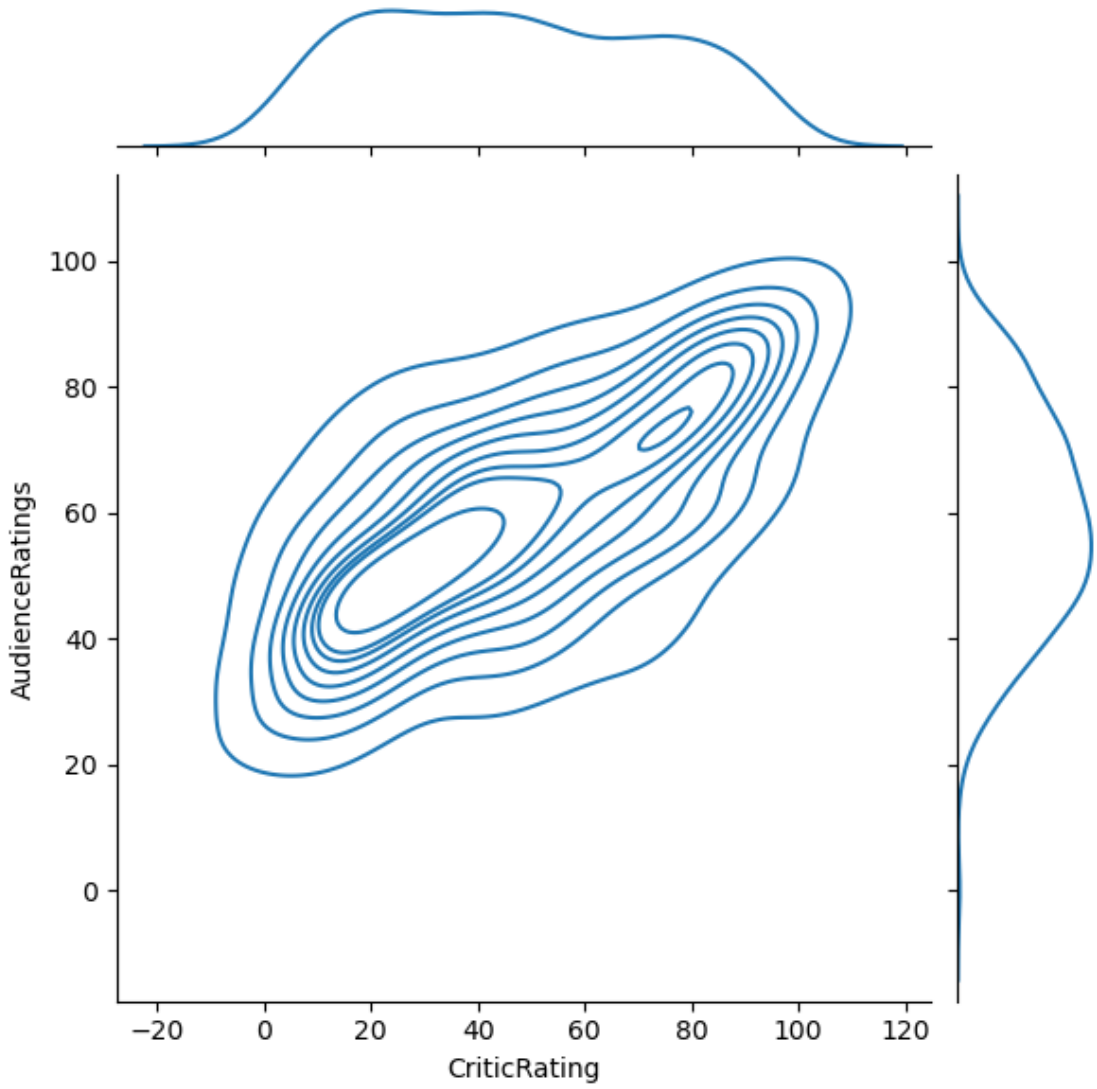
[125]:
```
j = sns.jointplot(data=movies, x='CriticRating', y='AudienceRatings',␣
↪kind='hex')
```



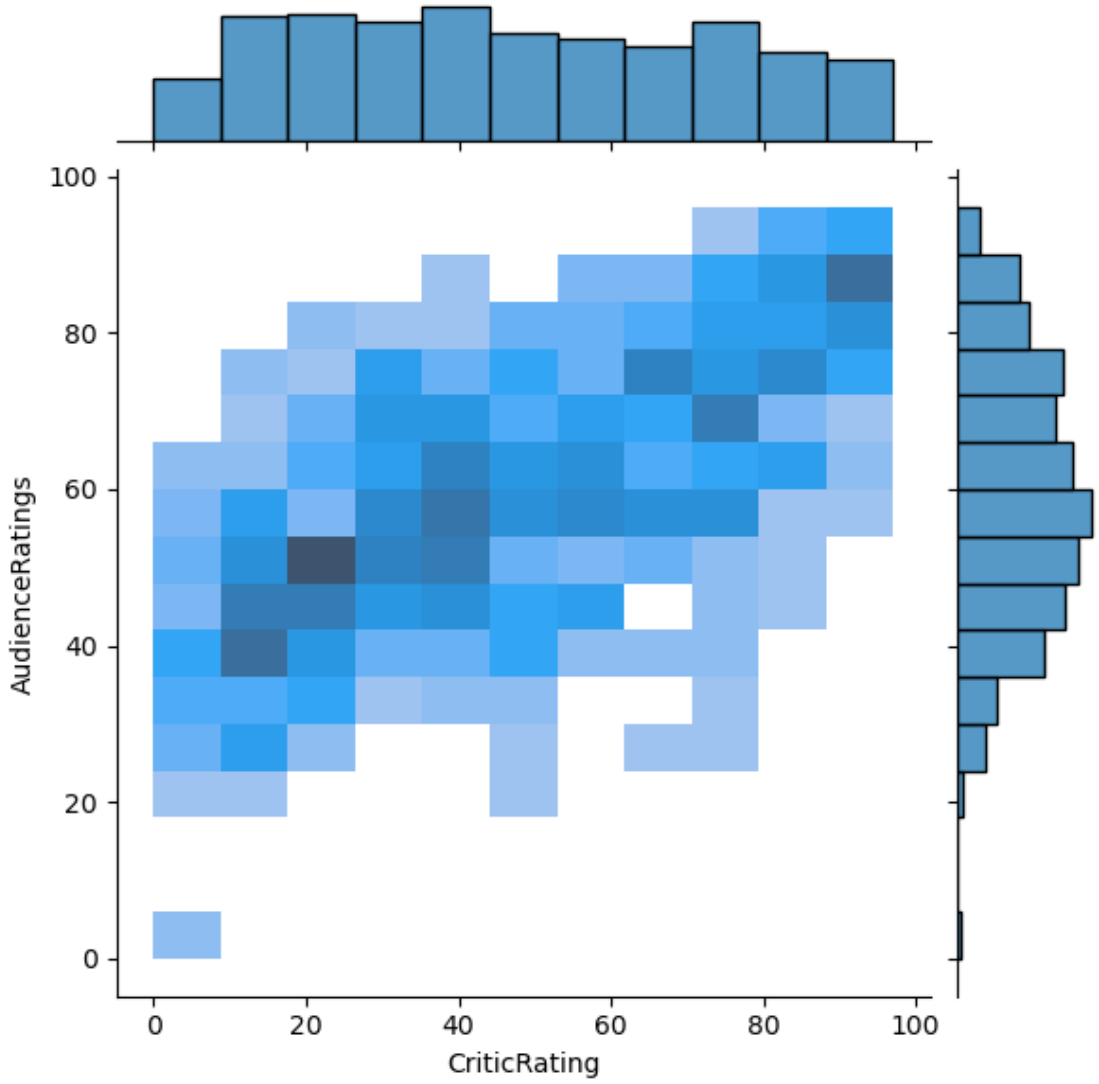
[126]:
```
j = sns.jointplot(data=movies, x='CriticRating', y='AudienceRatings',␣
↪kind='reg')
```



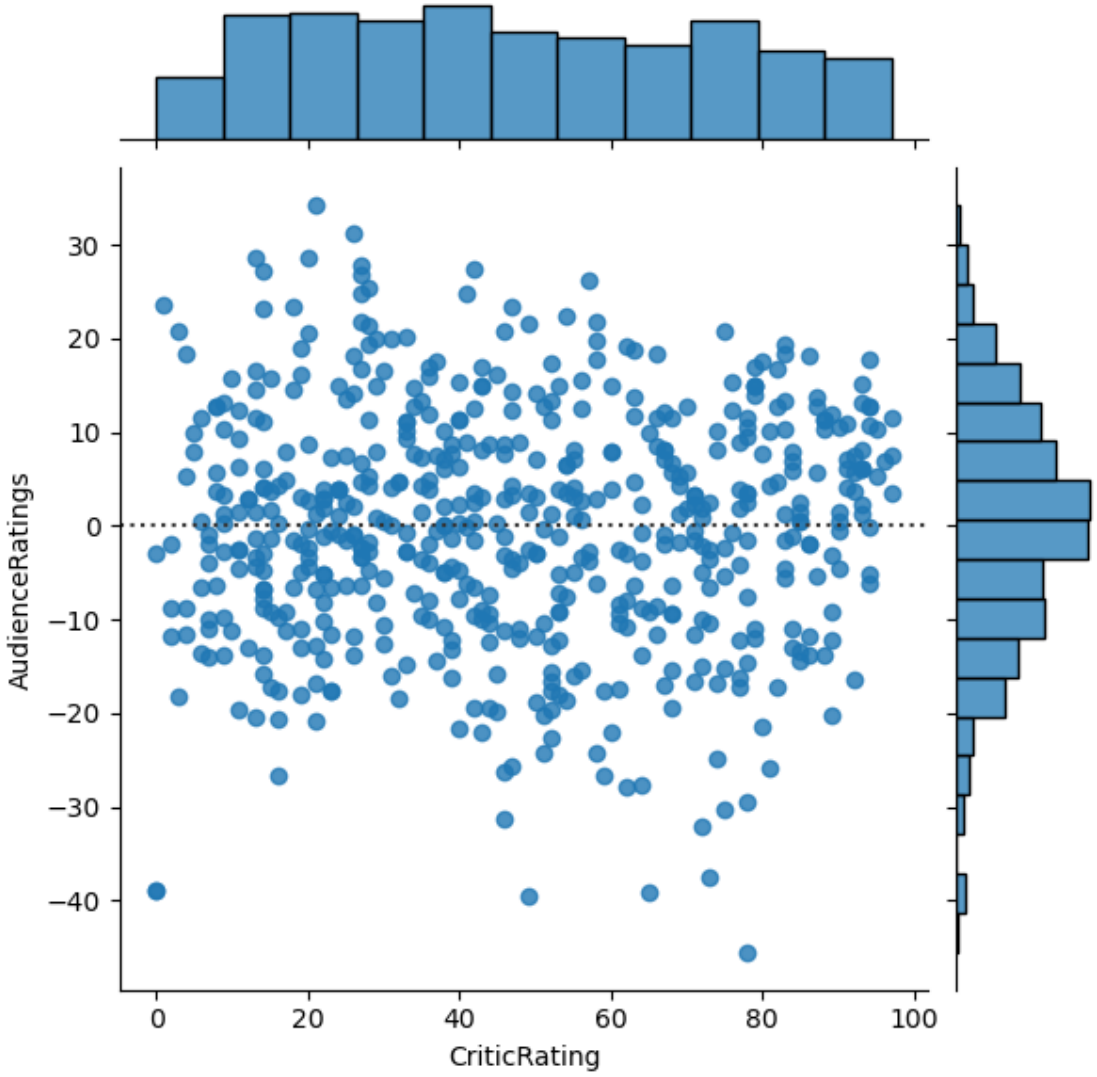
[127]:
```
j = sns.jointplot(data=movies, x='CriticRating', y='AudienceRatings',␣
↪kind='kde')
```



[128]:
```
j = sns.jointplot(data=movies, x='CriticRating', y='AudienceRatings',␣
↪kind='hist')
```



[129]:
```
j = sns.jointplot(data=movies, x='CriticRating', y='AudienceRatings',␣
↪kind='resid')
```

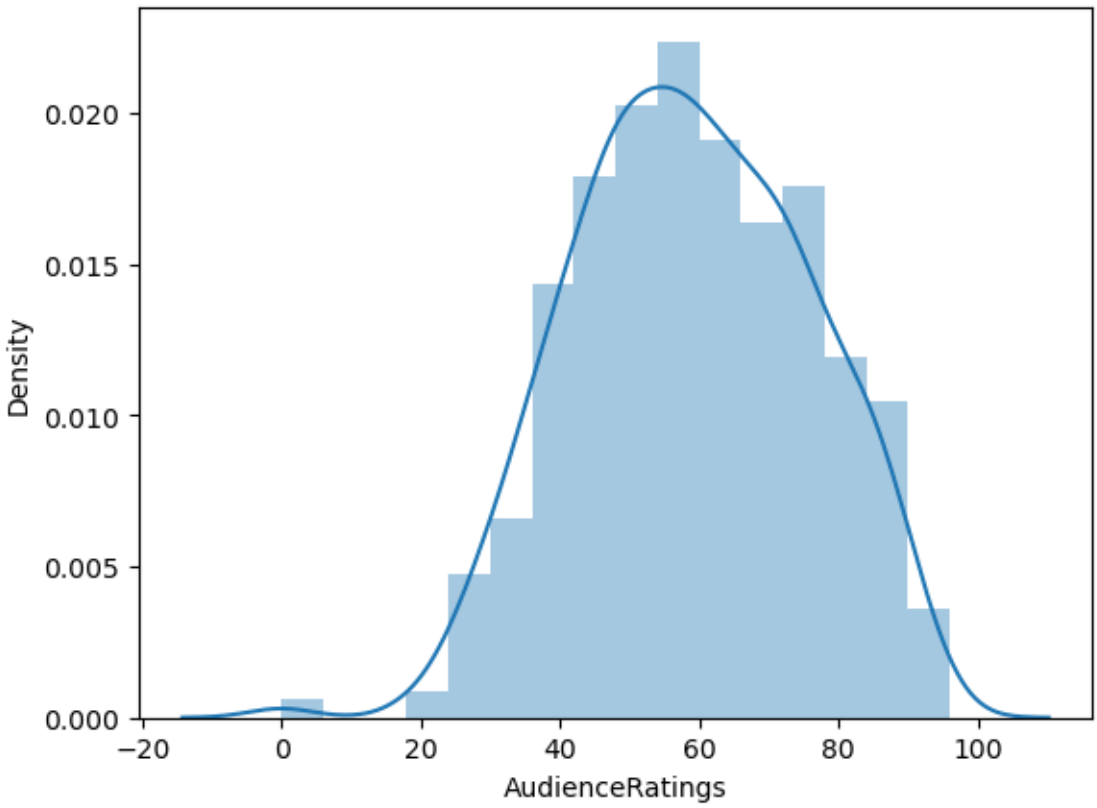


[ ]:
```
movies.columns
```

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

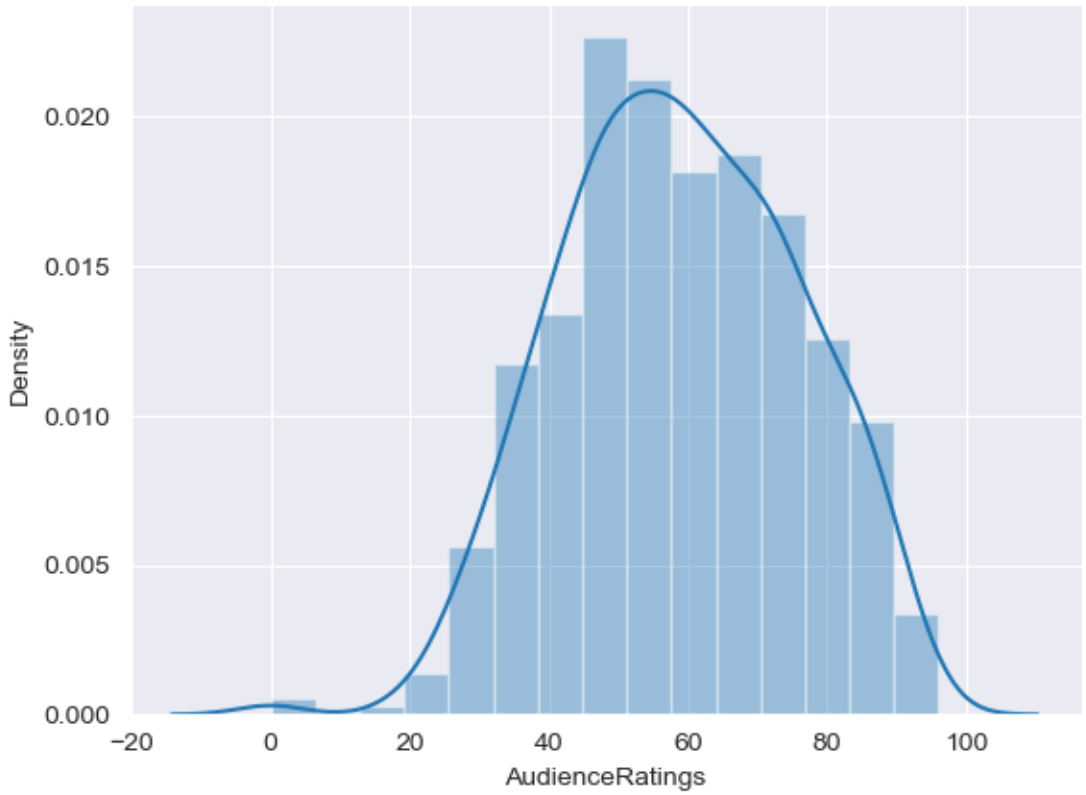
[ ]:
```
import seaborn as sns
```

[130]:
```
m = sns.distplot(movies.AudienceRatings)
```

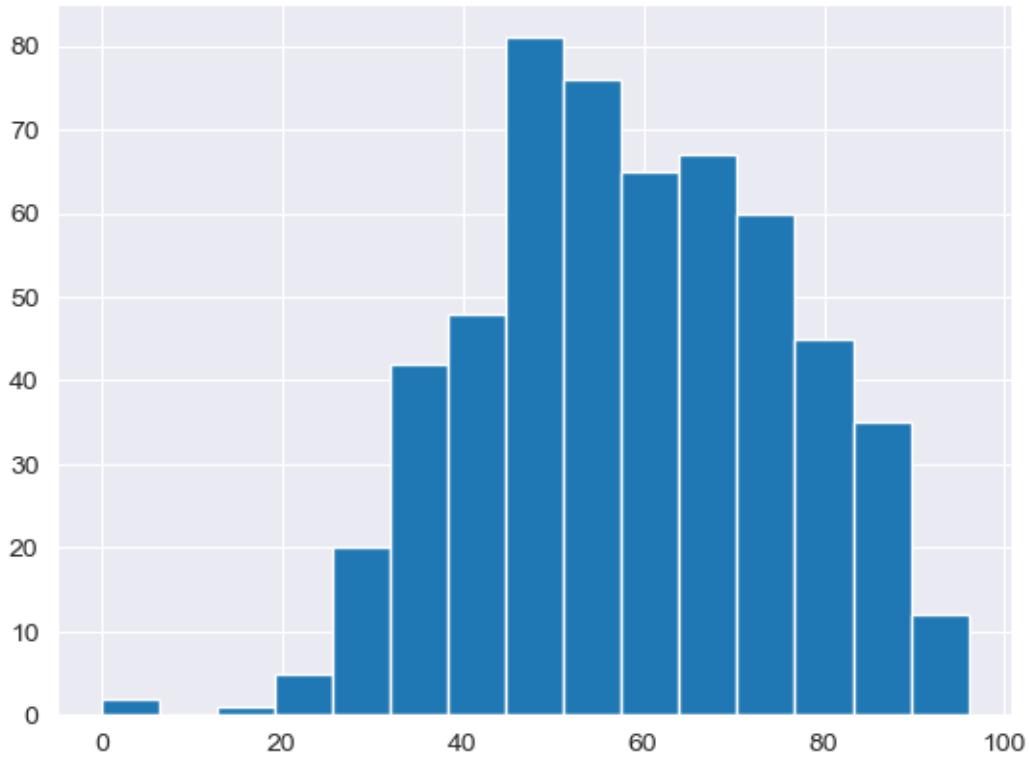


[131]:
```
sns.set_style('darkgrid')
```

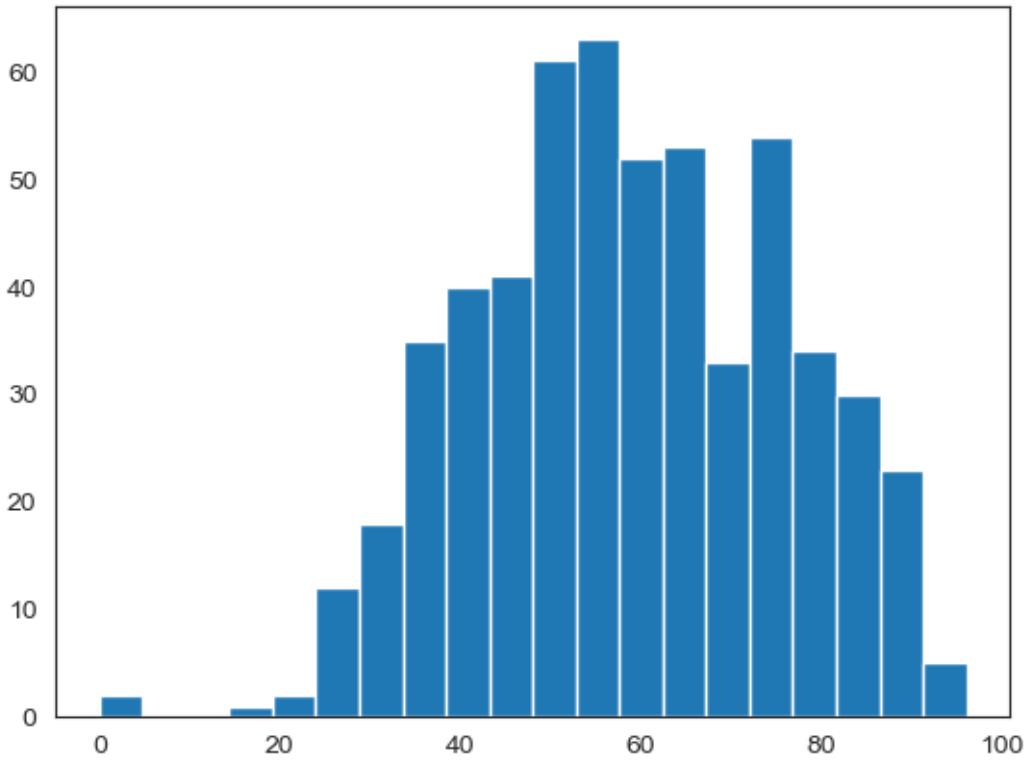
[133]:
```
m1 = sns.distplot(movies.AudienceRatings, bins=15)
```



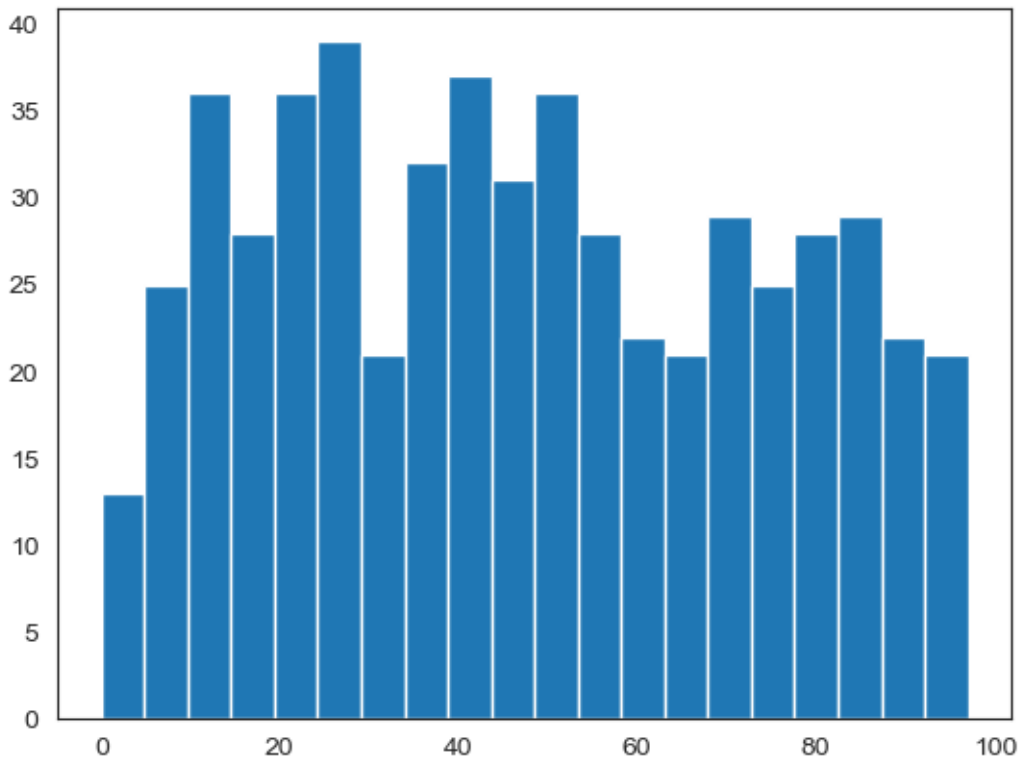
[135]:
```
m2 = plt.hist(movies.AudienceRatings, bins=15)
```



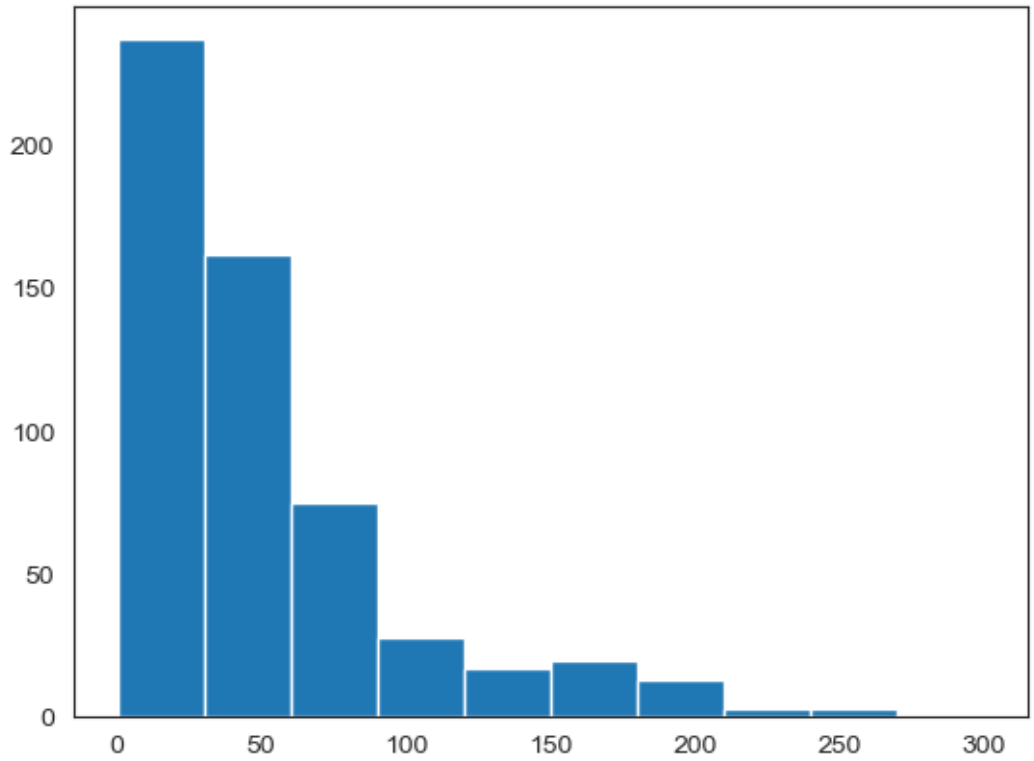
[139]:
```
sns.set_style('white')
m3 = plt.hist(movies.AudienceRatings, bins=20)
```



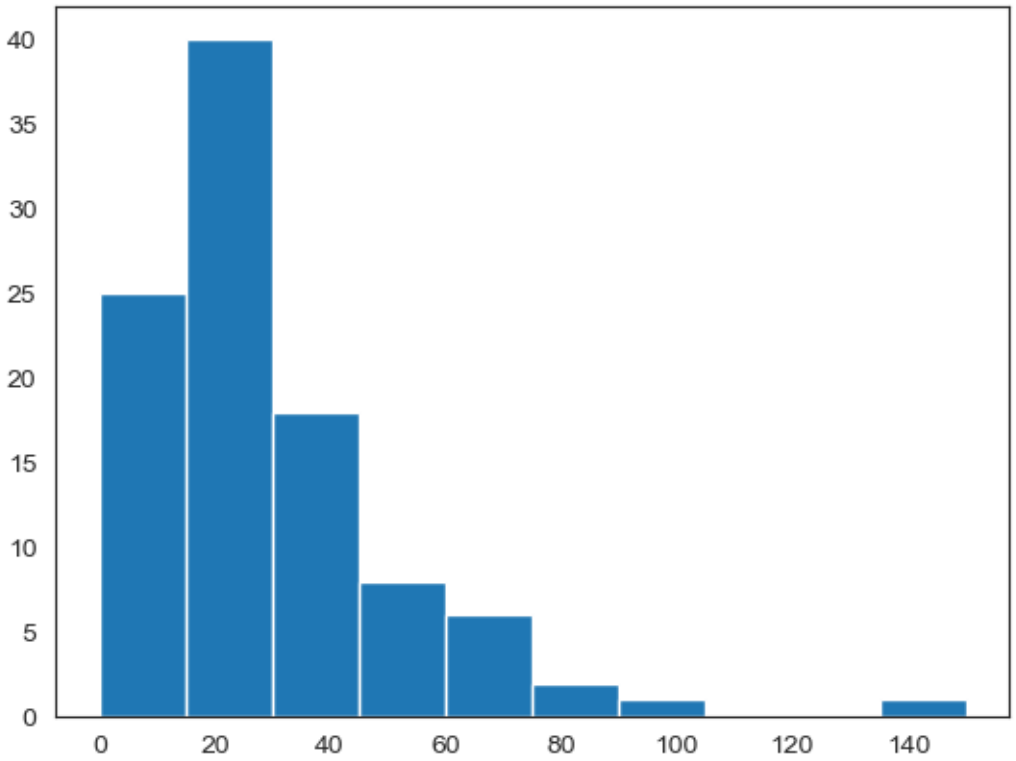
[140]:
```
m4 = plt.hist(movies.CriticRating, bins=20)
```



[ ]:
```
plt.hist(movies.BudgetMillions) #this graphs are bit hard to understand
plt.show()
```



[142]:
```
plt.hist(movies[movies.Genre == 'Drama'].BudgetMillions)
plt.show()
```



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

[143]:
```
                    Film      Genre  CriticRating  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

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

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: