

```
import pandas as pd
movies=pd.read_csv(r"/content/Movie-Rating (1).csv")
movies
```

	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

Next steps: [Generate code with movies](#) [View recommended plots](#) [New interactive sheet](#)

```
import pandas
print(pandas.__version__) #check pandas version
```

2.2.2

```
type(movies) #type of data structure in file movies
```

```
pandas.core.frame.DataFrame
def __init__(data=None, index: Axes | None=None, columns: Axes | None=None, dtype: Dtype | None=None, copy: bool | None=None) -> None

/usr/local/lib/python3.11/dist-packages/pandas/core/frame.py
Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns).
Arithmetic operations align on both row and column labels. Can be
thought of as a dict-like container for Series objects. The primary
```

```
len(movies)
```

559

```
movies.columns
```

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

```
movies.columns=['Film','Genre','CriticRating','AudienceRating','BudgetMillions','Year']
```

```
movies.head(1) # Removed spaces and % removed noise characters
```

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

Next steps: [Generate code with movies](#) [View recommended plots](#) [New interactive sheet](#)

```
movies.shape
```

(559, 6)

```
movies.describe() # descriptive statistic
```

	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

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 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

movies.Film=movies.Film.astype('category')

movies.Film

	Film
0	(500) Days of Summer
1	10,000 B.C.
2	12 Rounds
3	127 Hours
4	17 Again
...	...
554	Your Highness
555	Youth in Revolt
556	Zodiac
557	Zombieland
558	Zookeeper

559 rows × 1 columns

dtype: category

movies.head()

	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

Next steps:

Generate code with movies

☒ View recommended plots

New interactive sheet

movies.describe()



	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



```
movies.Genre=movies.Genre.astype('category')
```

```
movies.Genre
```



	Genre
0	Comedy
1	Adventure
2	Action
3	Adventure
4	Comedy
...	...
554	Comedy
555	Comedy
556	Thriller
557	Action
558	Comedy

559 rows × 1 columns

dtype: category



```
movies.Year=movies.Year.astype('category')
```

```
movies.Year
```




	Year
0	2009
1	2008
2	2009
3	2010
4	2009
...	...
554	2011
555	2009
556	2007
557	2009
558	2011

559 rows × 1 columns

dtype: category



```
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   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
```

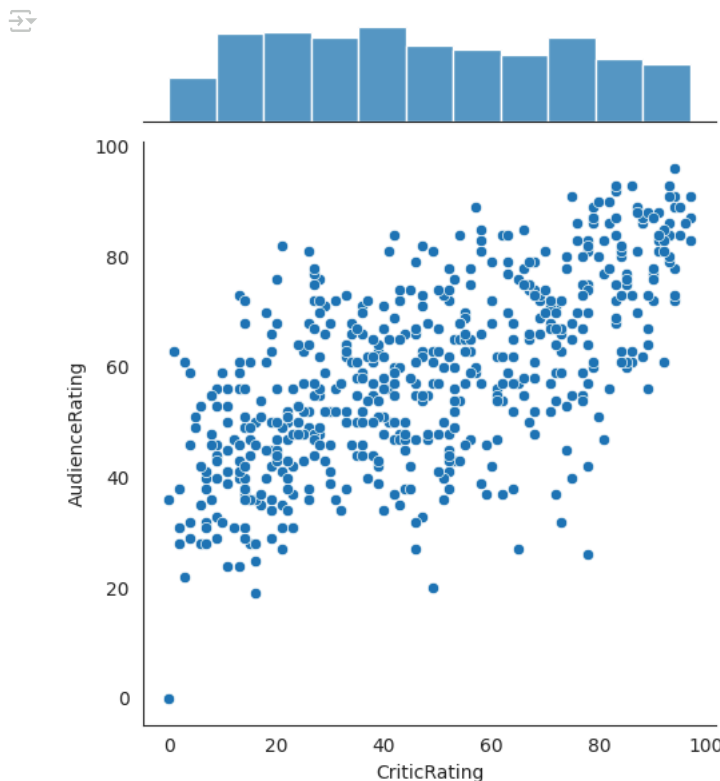
```
movies.describe()
```

	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

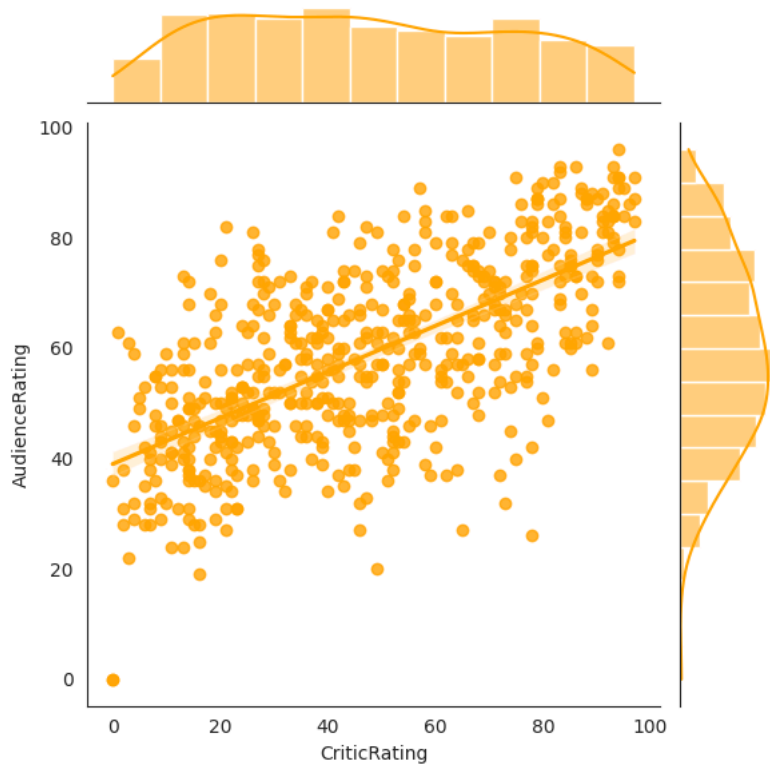
```
#how to working with joint plot
from matplotlib import pyplot as plt#visualization
import seaborn as sns
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')

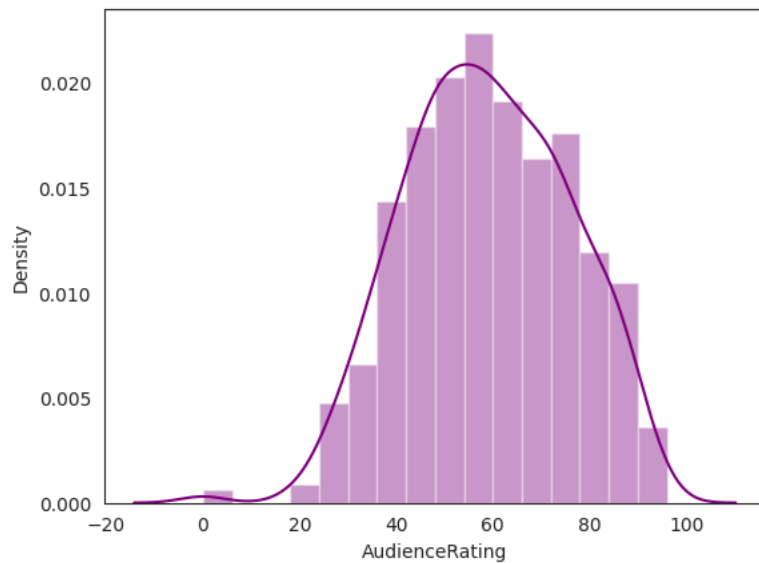
j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating')
#Audience rating is more dominant the critics rating
#Based on this we find out as most people are most liklihood to watch audience rating and less likely to watch critic rating
#There is positive correlation between 2 attributes
#let me explain the excel -if you filter audience rating and critic rating.critic rating has very low value
```



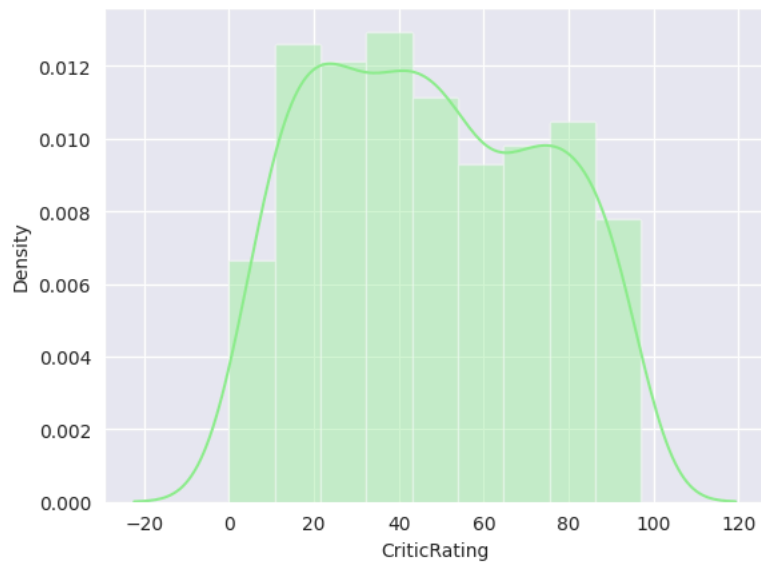
```
j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='reg',color='orange')
```



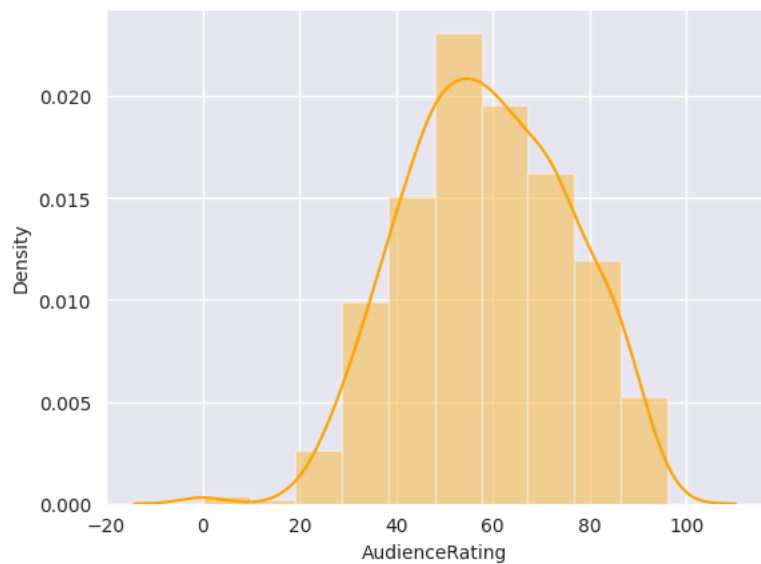
```
m1=sns.distplot(movies.AudienceRating,color='purple')
```



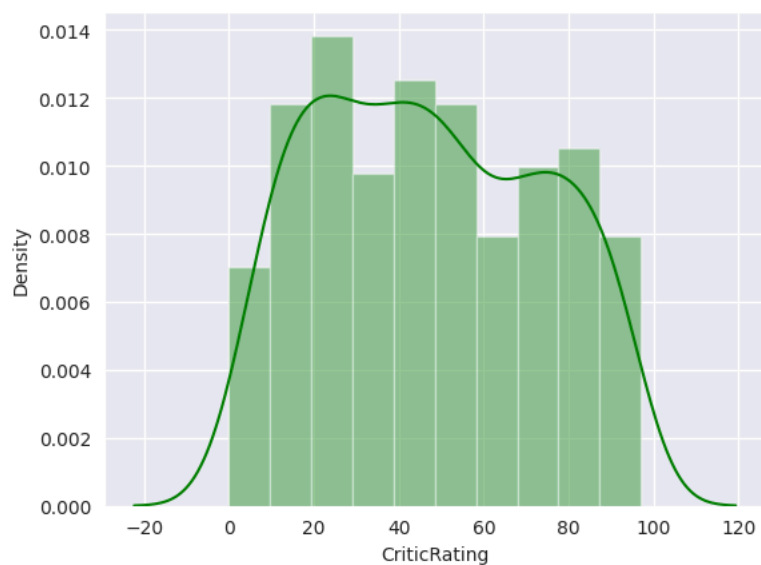
```
sns.set_style('darkgrid')  
m1=sns.distplot(movies.CriticRating,color='lightgreen')
```



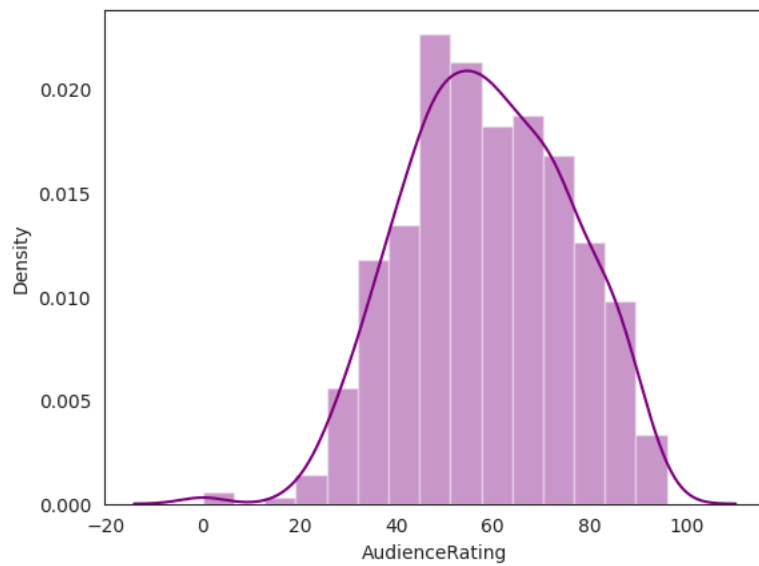
```
m2=sns.distplot(movies.AudienceRating,bins=10,color='orange')
sns.set_style('darkgrid')
```



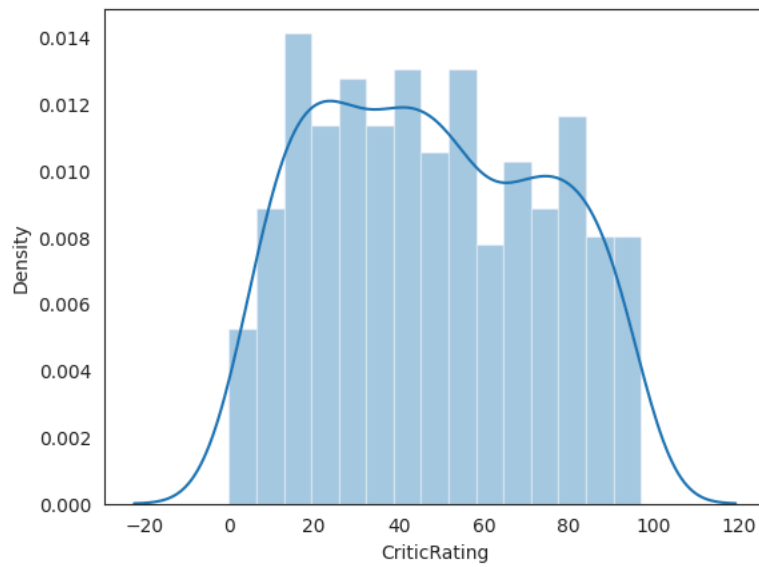
```
sns.set_style('darkgrid')
m2=sns.distplot(movies.CriticRating,bins=10,color='green')
```



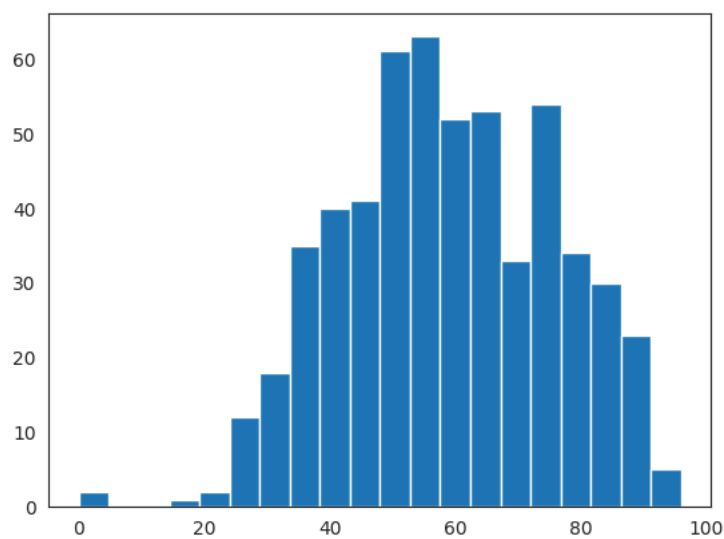
```
sns.set_style('white')
m3=sns.distplot(movies.AudienceRating,bins=15,color='purple')
```



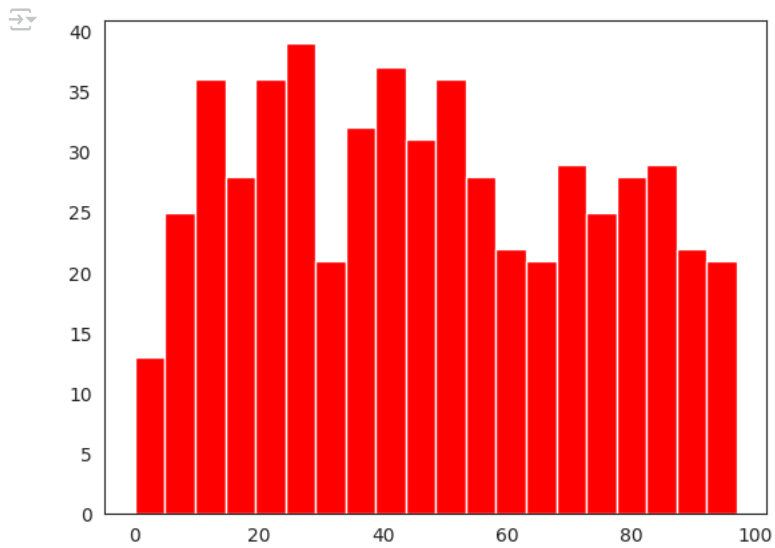
```
sns.set_style('white')  
m3=sns.distplot(movies.CriticRating,bins=15)
```



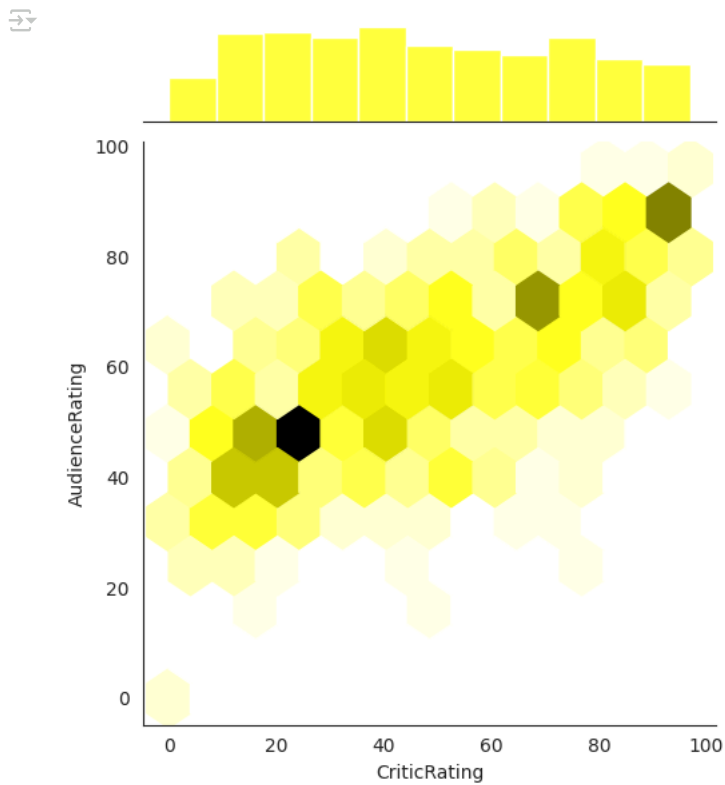
```
n1=plt.hist(movies.AudienceRating,bins=20)
```



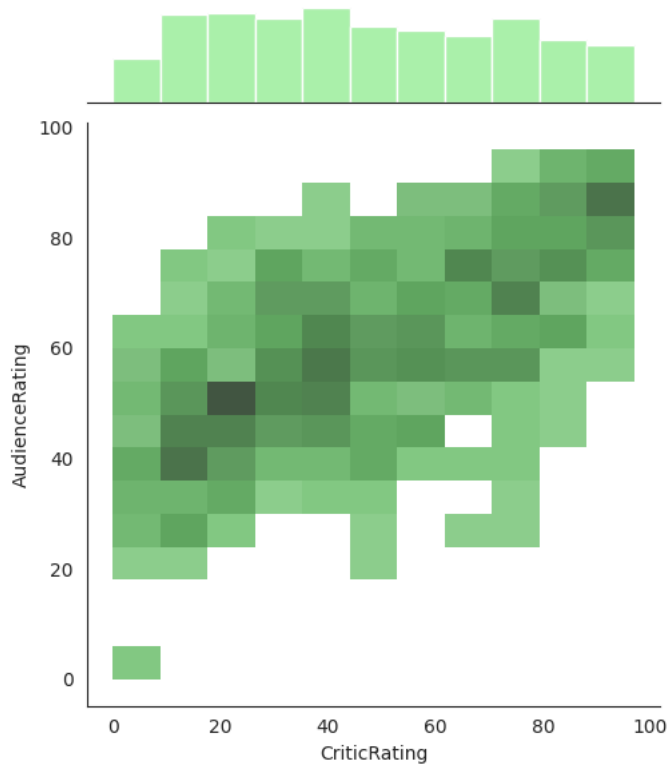
```
n1=plt.hist(movies.CriticRating,bins=20,color='red')
```



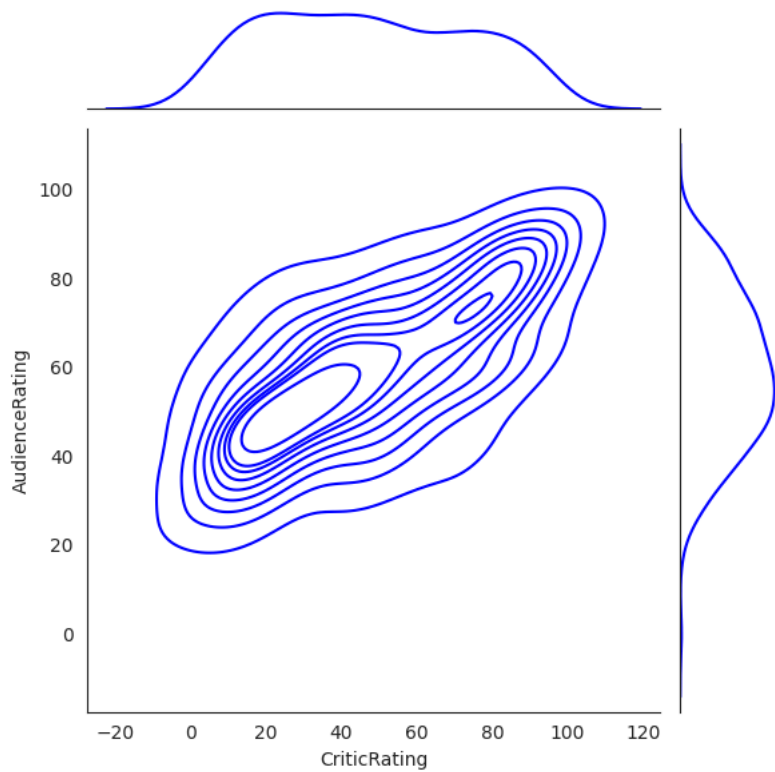
```
j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='hex',color='yellow')
```



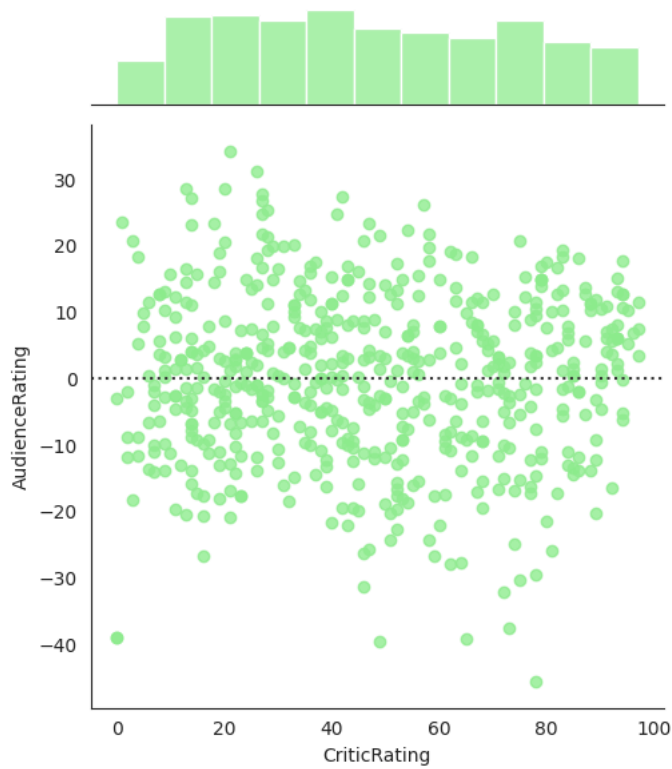
```
j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='hist',color='lightgreen')
```

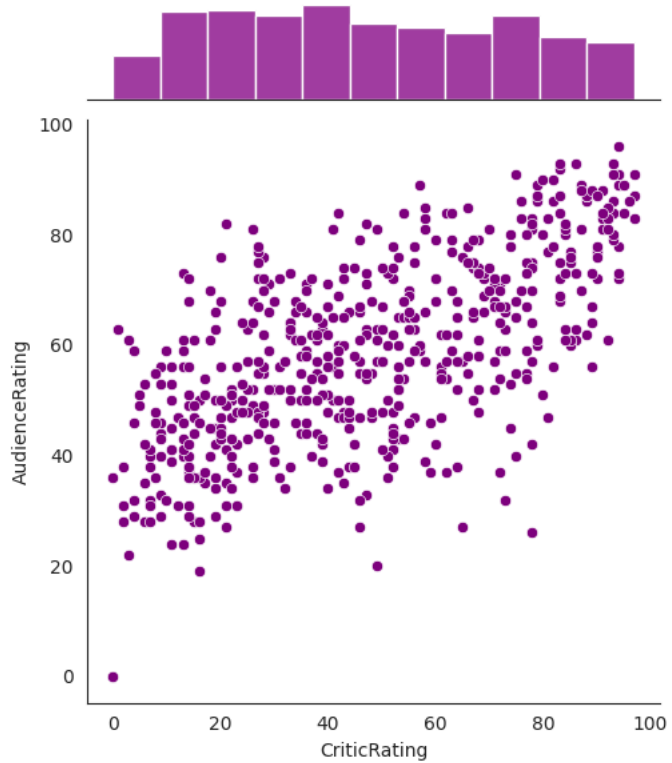
```
j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='kde',color='blue')
```



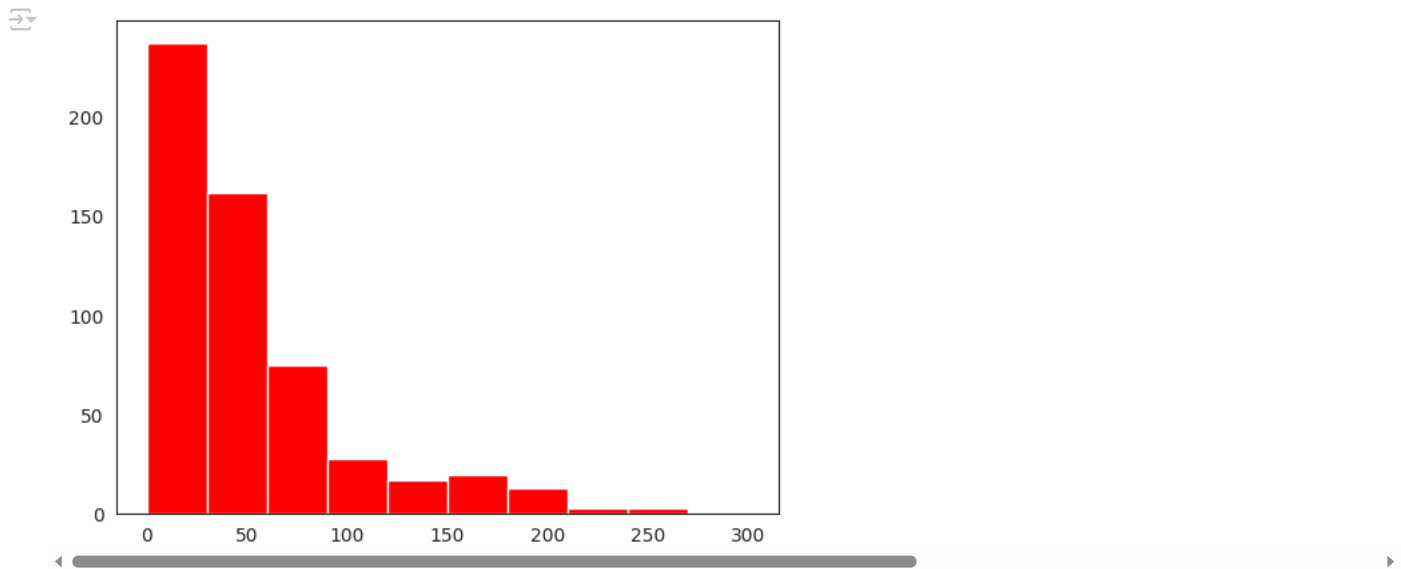
```
j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='resid',color='lightgreen')
```



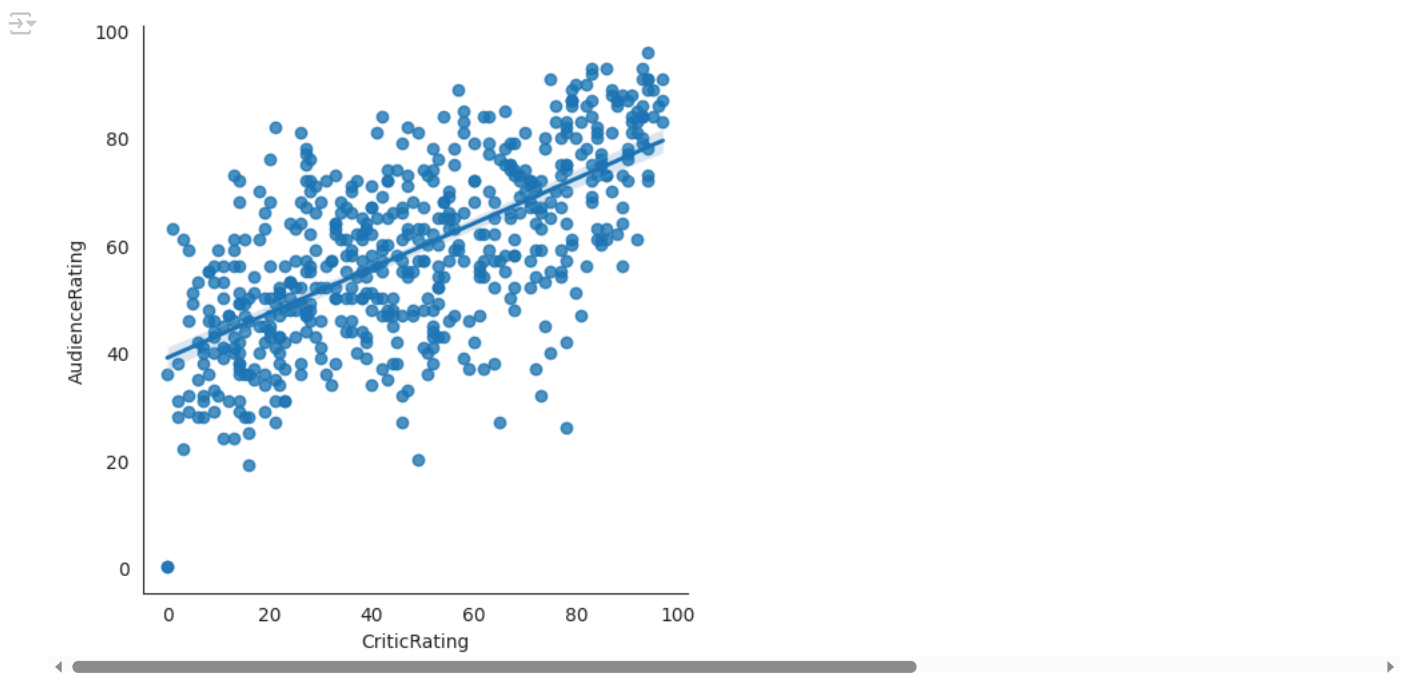
```
j=sns.jointplot(data=movies,x='CriticRating',y='AudienceRating',kind='scatter',color='purple')
```



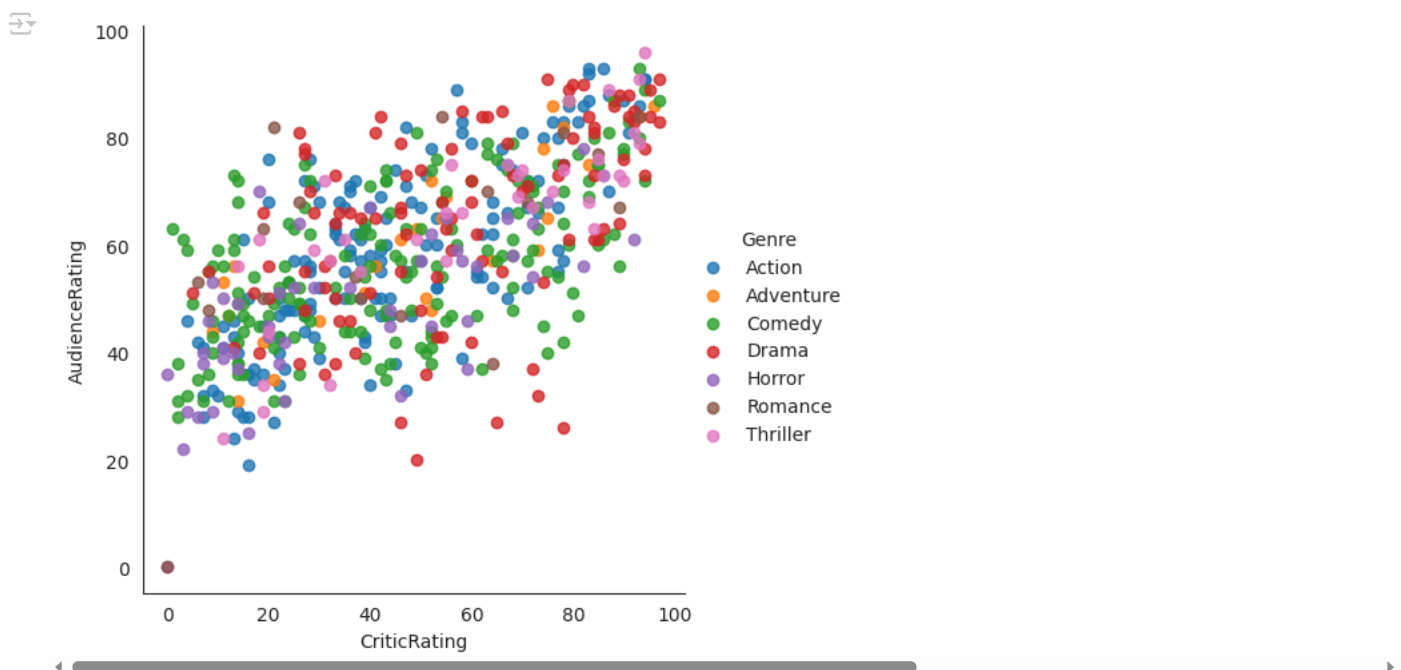
```
plt.hist(movies.BudgetMillions,color='red')
plt.show()
```



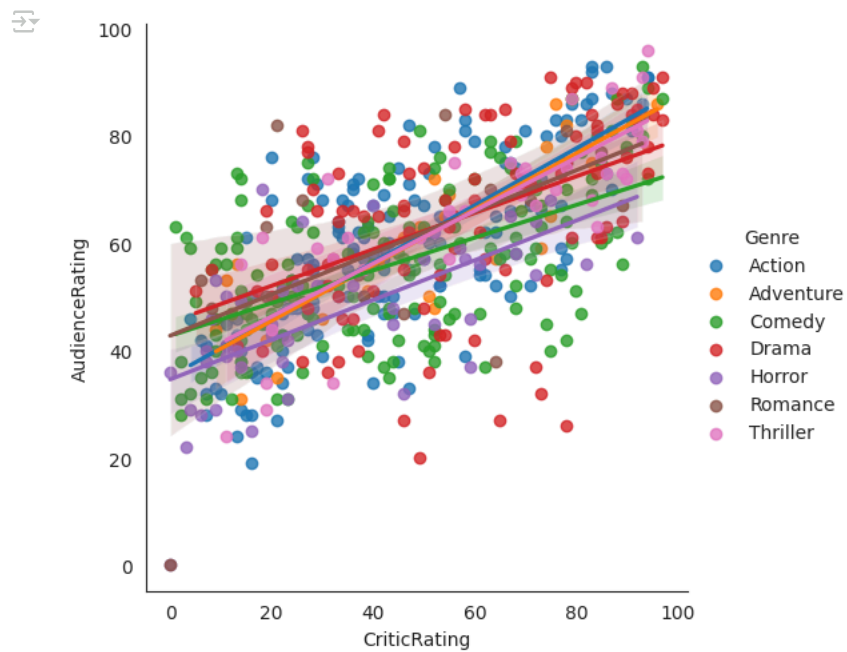
```
vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',fit_reg=True)
```



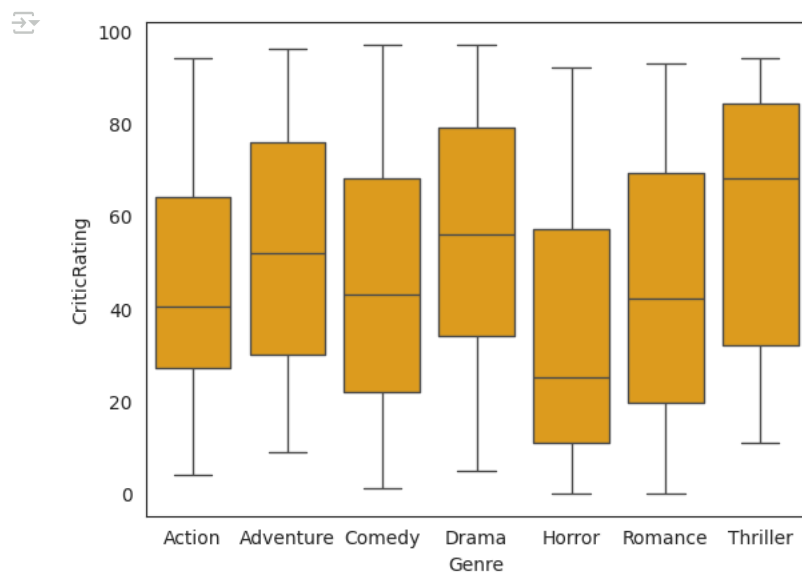
```
vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',fit_reg=False,hue='Genre')
```



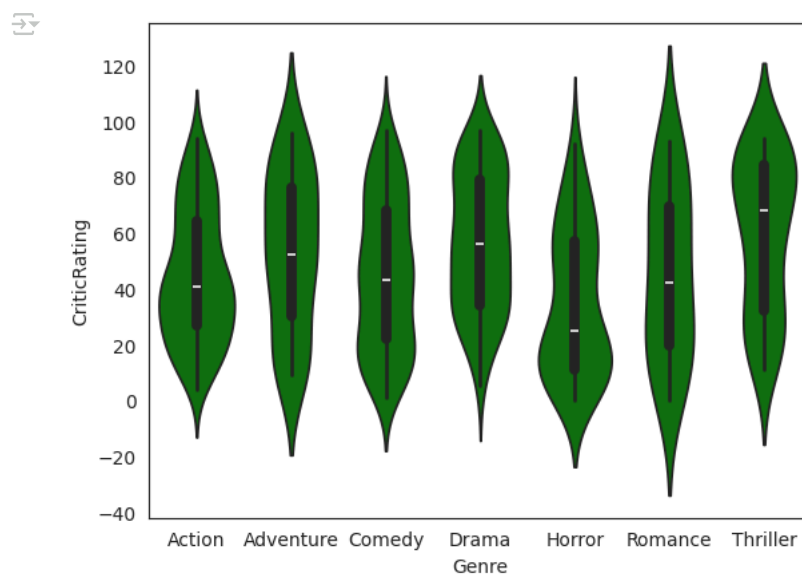
```
vis1=sns.lmplot(data=movies,x='CriticRating',y='AudienceRating',fit_reg=True,hue='Genre')
```



```
w=sns.boxplot(data=movies,x='Genre',y='CriticRating',color='orange')
```

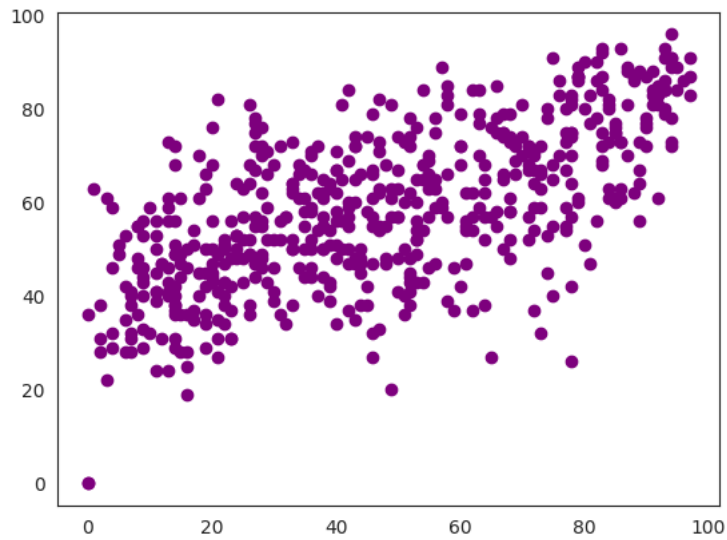


```
z=sns.violinplot(data=movies,x='Genre',y='CriticRating',color='green')
```



```
nlt.scatter(movies.CriticRating,movies.AudienceRating,color='purple')
```

```
<matplotlib.collections.PathCollection at 0x78f3d58c1d10>
```



```
w1=sns.boxplot(data=movies[movies.Genre=='Drama'],x='Year',y='AudienceRating',color='red')
```

