

In [4]:

```
import pandas as pd
```

In [5]:

```
df=pd.read_csv(r'C:\Users\Dell\Downloads\10th,11th\10th,11th\MOVIE RATINGS _ ADVANCE VIS
```

In [6]:

```
df
```

Out[6]:

	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 [7]:

```
df.shape
```

Out[7]:

(559, 6)

In [8]:

df.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
```

In [ ]:

```
df.Film=df.Film.astype('category')
df.Genre=df.Genre.astype('category')
df['CriticalRating']=df['CriticalRating'].astype(int)
df['AudienceRatings']=df['AudienceRatings'].astype(int)
df['Budget']=df['Budget'].astype(int)
df['Year of release']=df['Year of release'].astype(int)
```

In [11]:

df.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   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: category(2), int64(4)
memory usage: 40.1 KB
```

In [12]:

```
df.head()
```

Out[12]:

	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 [13]:

```
df.head(10)
```

Out[13]:

	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
5	2012	Action	39	63	200	2009
6	27 Dresses	Comedy	40	71	30	2008
7	30 Days of Night	Horror	50	57	32	2007
8	30 Minutes or Less	Comedy	43	48	28	2011
9	50/50	Comedy	93	93	8	2011

In [14]:

```
df.tail()
```

Out[14]:

	Film	Genre	Rotten Tomatoes Ratings %	Audience Ratings %	Budget (million \$)	Year of release
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

In [ ]:

```
df.columns
```

In [28]:

```
df.columns=['Film', 'Genre', 'CriticalRating', 'AudienceRatings', 'Budget', 'Year of release']
```

In [29]:

```
df.columns
```

Out[29]:

```
Index(['Film', 'Genre', 'CriticalRating', 'AudienceRatings', 'Budget',  
      'Year of release'],  
      dtype='object')
```

In [30]:

```
df
```

Out[30]:

	Film	Genre	CriticalRating	AudienceRatings	Budget	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 [31]:

```
df[10:20]
```

Out[31]:

	Film	Genre	CriticalRating	AudienceRatings	Budget	Year of release
10	88 Minutes	Drama	5	51	30	2007
11	A Dangerous Method	Drama	79	89	20	2011
12	A Nightmare on Elm Street	Horror	13	40	35	2010
13	A Serious Man	Drama	89	64	7	2009
14	A Very Harold and Kumar Christmas	Comedy	72	71	19	2011
15	Abduction	Action	4	46	35	2011
16	Across the Universe	Romance	54	84	45	2007
17	Adventureland	Comedy	89	56	10	2009
18	Albert Nobbs	Drama	53	43	8	2011
19	Alice in Wonderland	Adventure	52	72	200	2010

In [17]:

```
len(df)
```

Out[17]:

559

In [32]:

```
categorical_data=df[['Film', 'Genre']]
```

In [33]:

```
categorical_data
```

Out[33]:

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

559 rows × 2 columns

In [35]:

```
numerical_data=df[['CriticalRating','AudienceRatings','Budget','Year of release']]
numerical_data
```

Out[35]:

	CriticalRating	AudienceRatings	Budget	Year of release
0	87	81	8	2009
1	9	44	105	2008
2	30	52	20	2009
3	93	84	18	2010
4	55	70	20	2009
...	...	...	...	...
554	26	36	50	2011
555	68	52	18	2009
556	89	73	65	2007
557	90	87	24	2009
558	14	42	80	2011

559 rows × 4 columns

In [36]:

```
df
```

Out[36]:

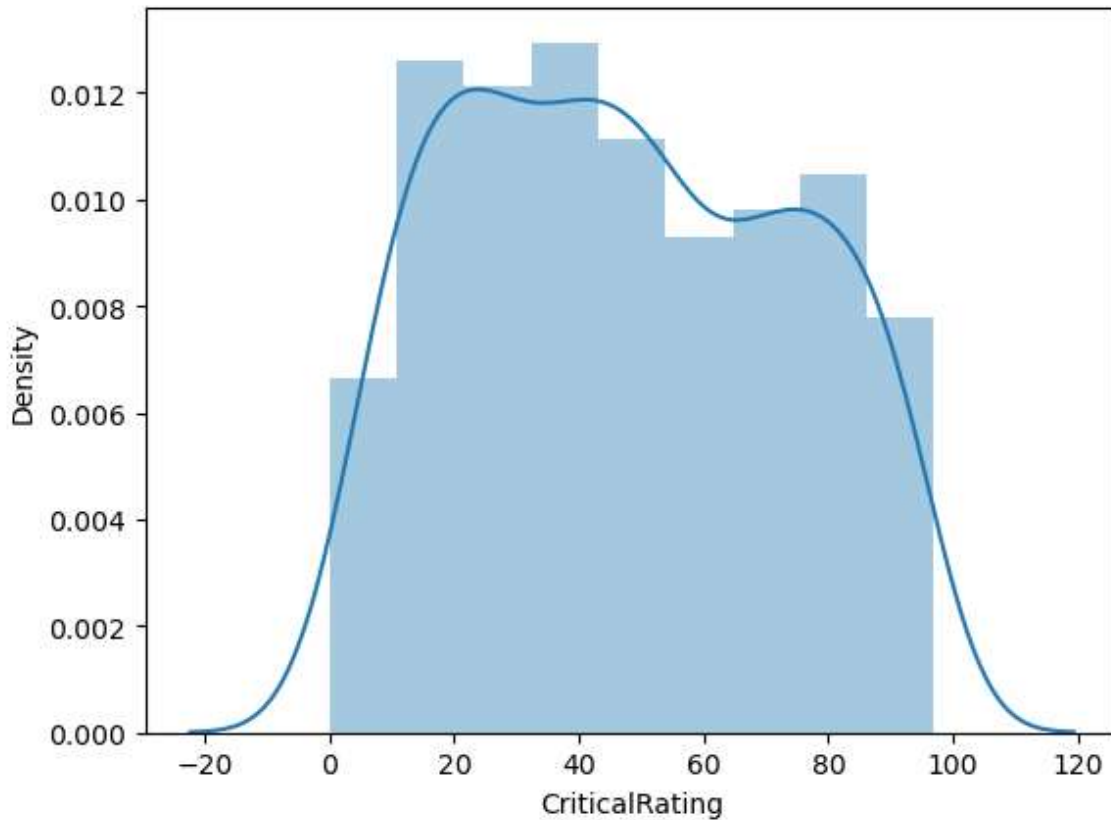
	Film	Genre	CriticalRating	AudienceRatings	Budget	Year of release
0	(500) Days of Summer	Comedy	87	81	8	2009
1	10,000 B.C.	Adventure	9	44	105	2008
2	12 Rounds	Action	30	52	20	2009
3	127 Hours	Adventure	93	84	18	2010
4	17 Again	Comedy	55	70	20	2009
...	...	...	...	...	...	...
554	Your Highness	Comedy	26	36	50	2011
555	Youth in Revolt	Comedy	68	52	18	2009
556	Zodiac	Thriller	89	73	65	2007

In [37]:

```
import os
os.getcwd()
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [38]:

```
vis1=sns.distplot(df['CriticalRating'])
```



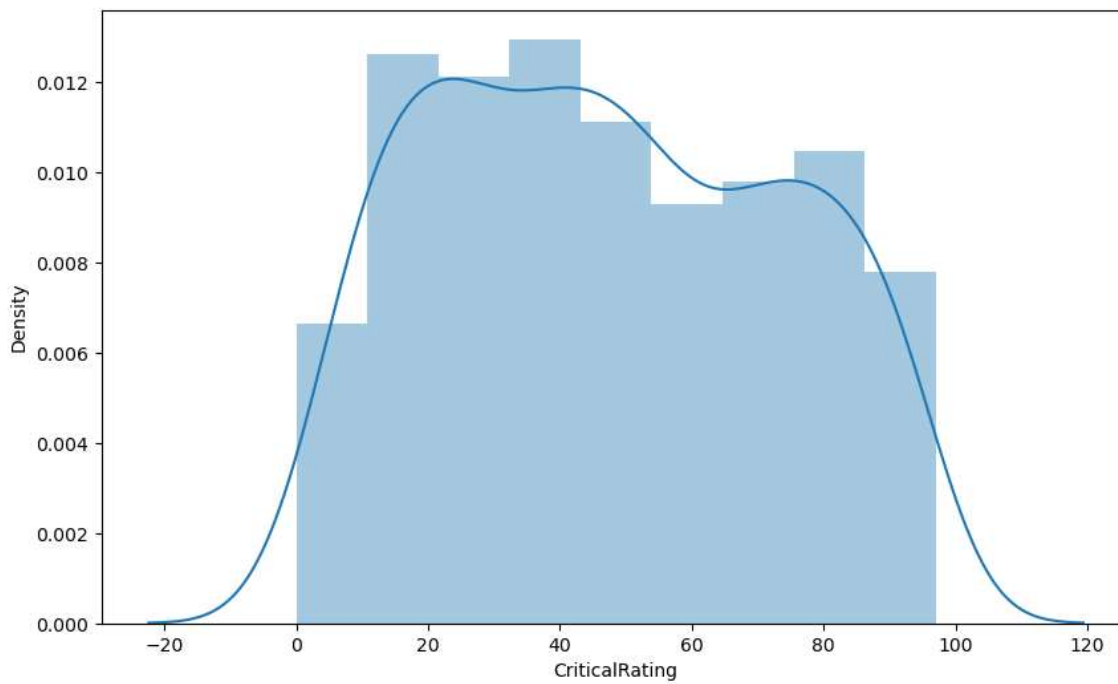
In [39]:

```
plt.rcParams['figure.figsize']=10,6
```



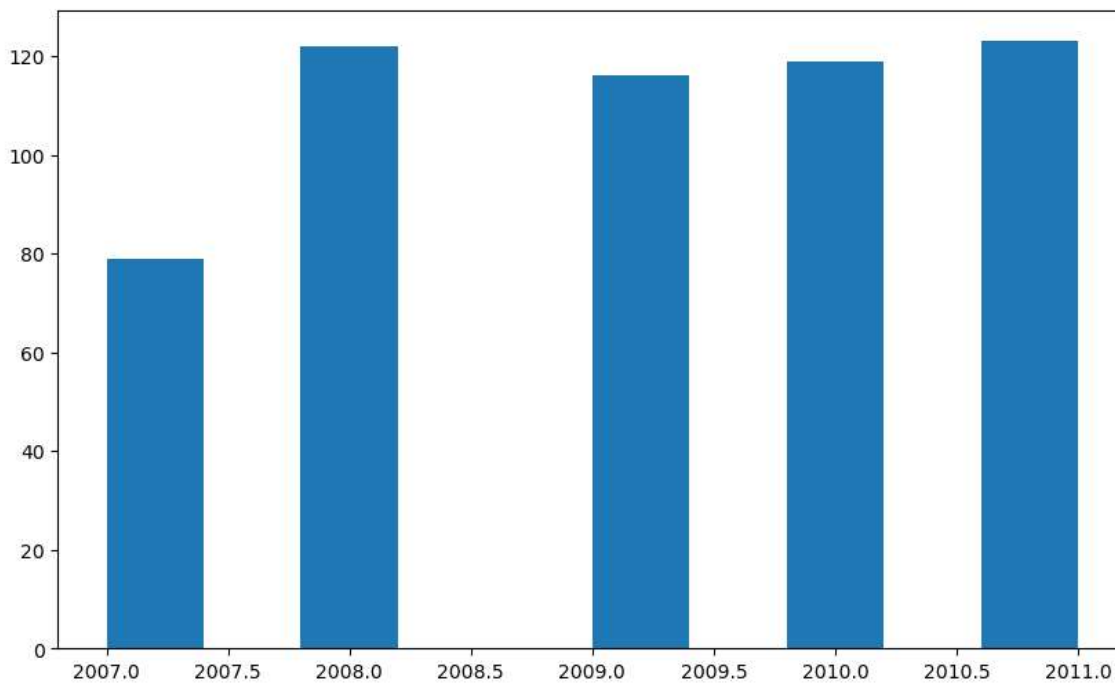
In [40]:

```
vis1=sns.distplot(df['CriticalRating'])
```



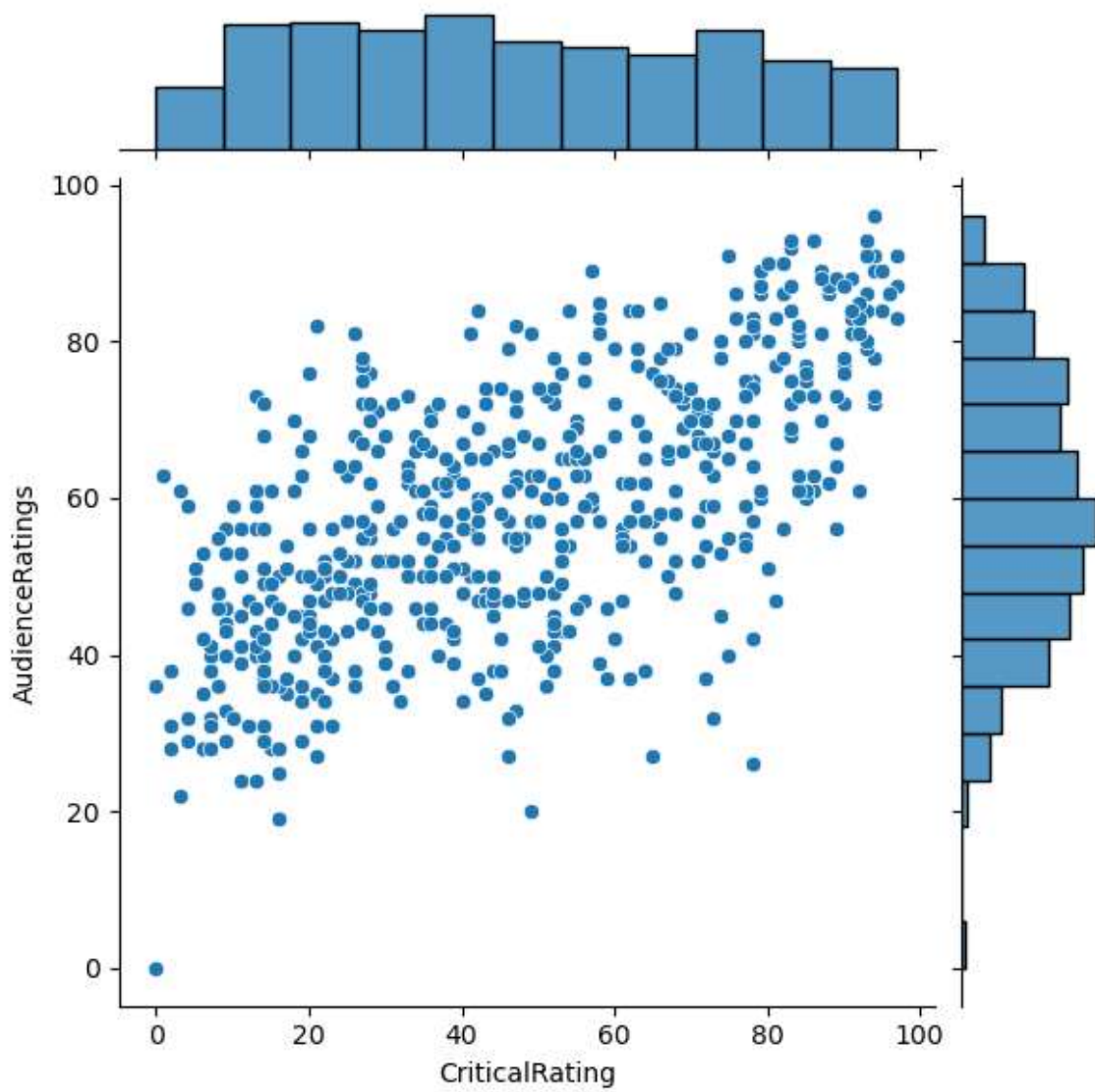
In [41]:

```
vis2=plt.hist(df['Year of release'])
```



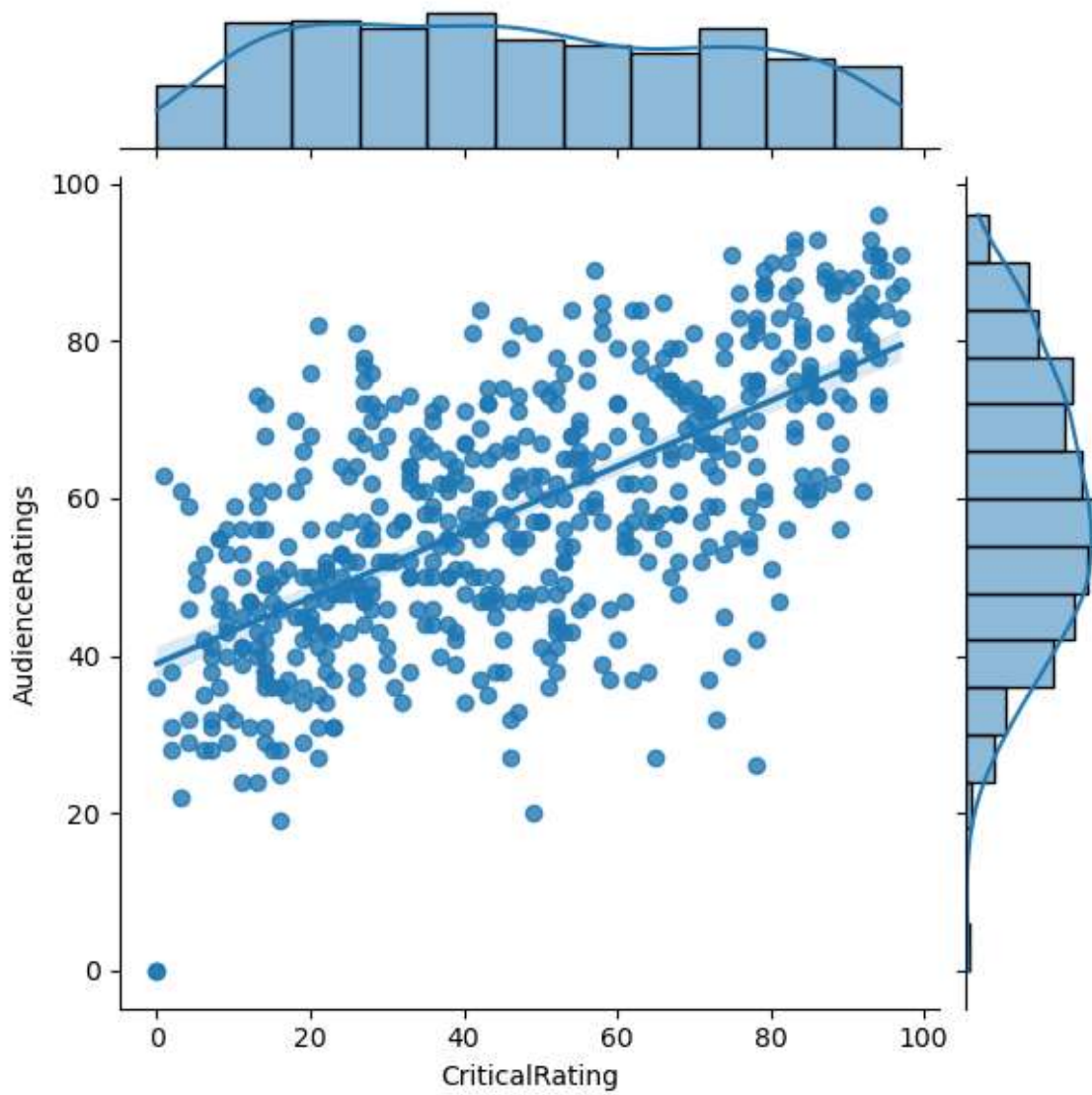
In [42]:

```
j=sns.jointplot(data=df, x='CriticalRating',y='AudienceRatings')
```



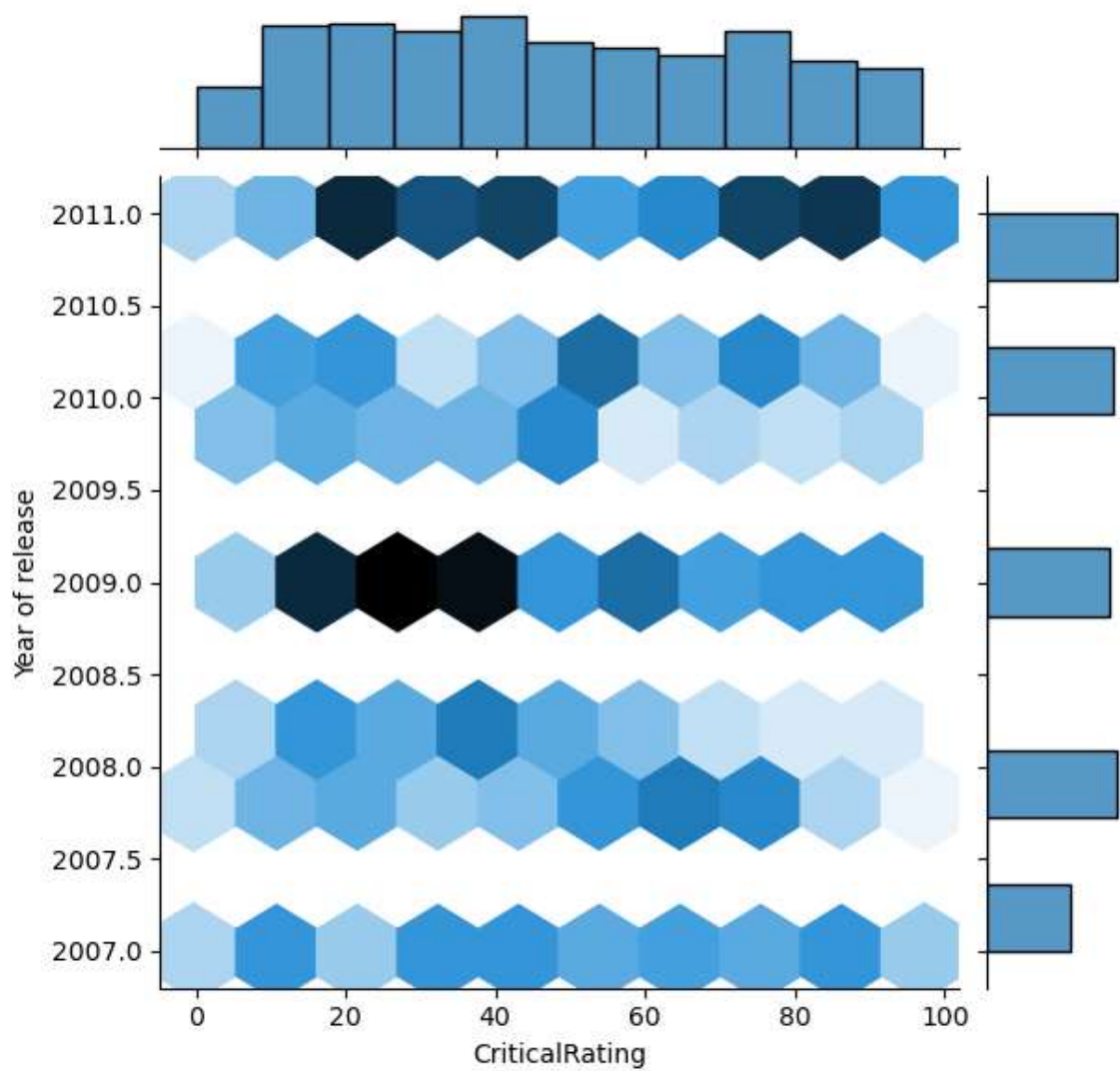
In [43]:

```
j=sns.jointplot(data=df, x='CriticalRating',y='AudienceRatings',kind='reg')
```



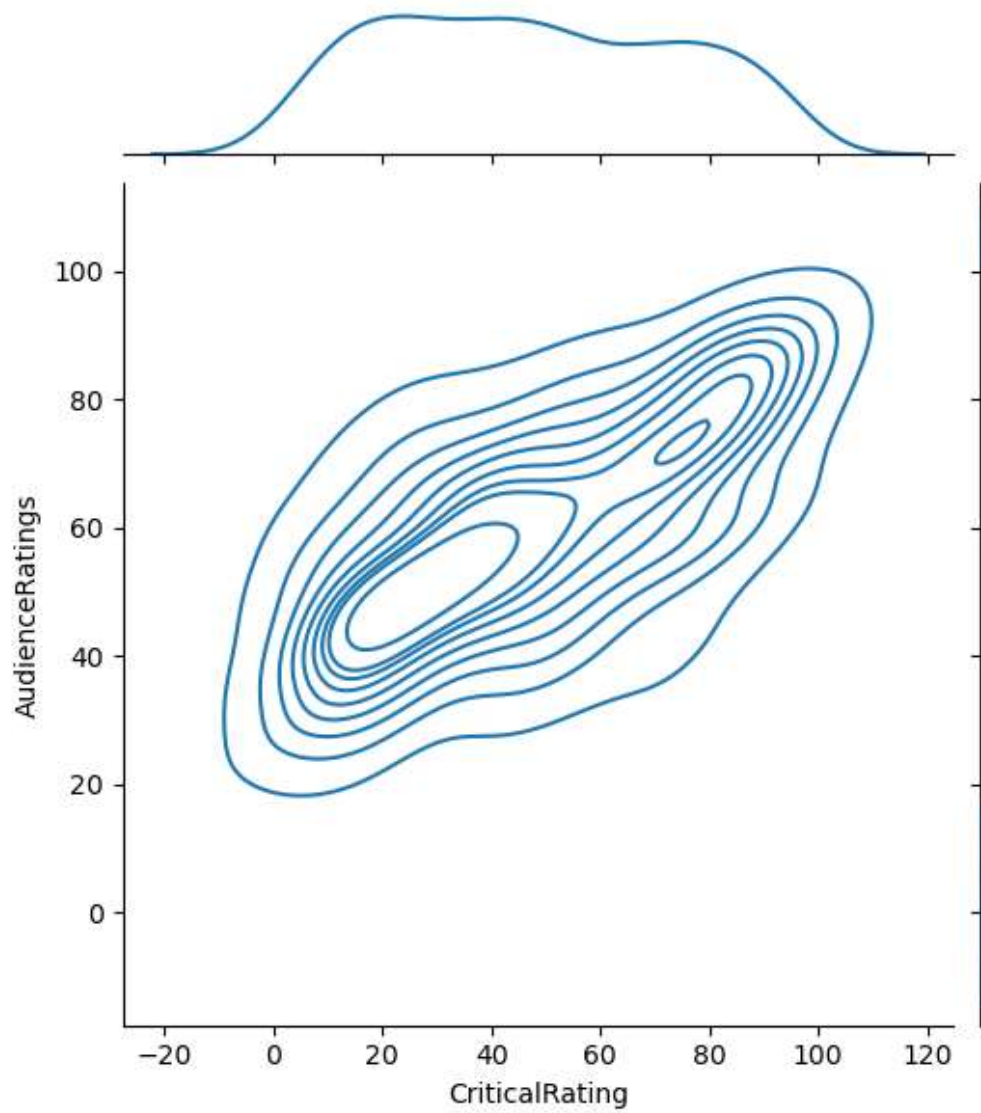
In [44]:

```
j=sns.jointplot(data=df, x='CriticalRating',y='Year of release',kind='hex')
```



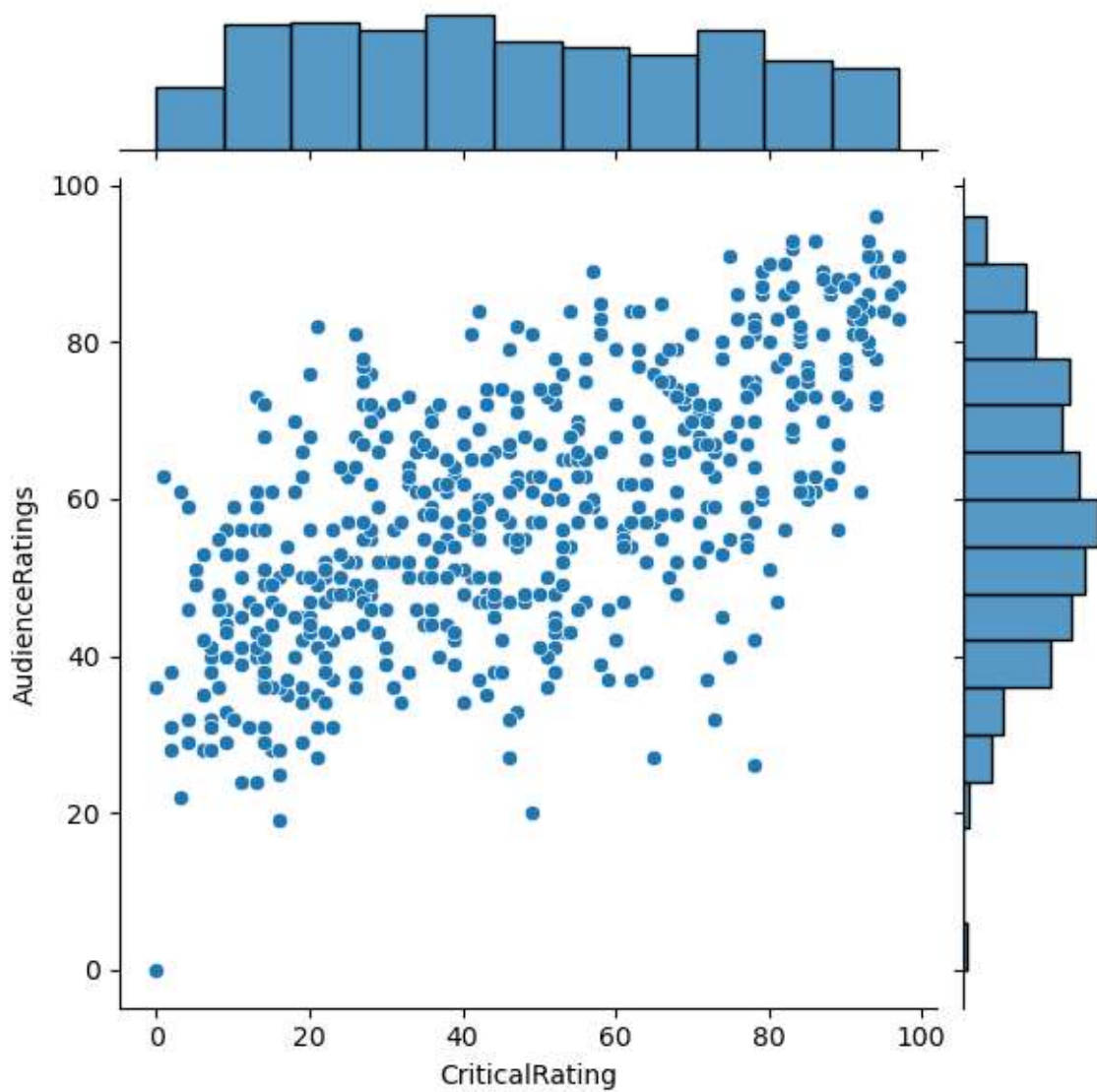
In [45]:

```
j=sns.jointplot(data=df, x='CriticalRating',y='AudienceRatings',kind='kde')
```



In [46]:

```
j=sns.jointplot(data=df, x='CriticalRating',y='AudienceRatings',kind='scatter')
```

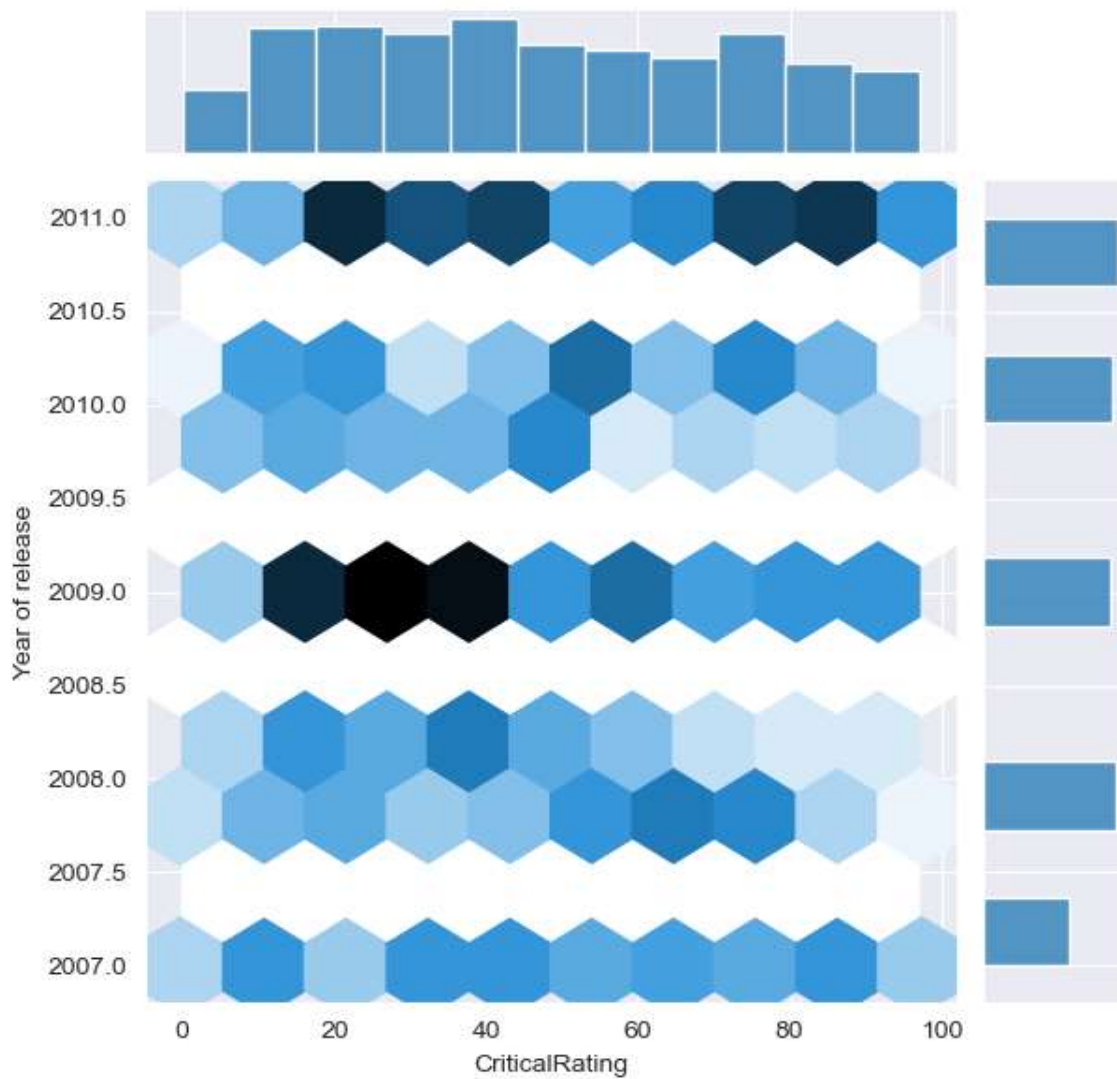


In [47]:

```
sns.set_style('darkgrid')
```

In [48]:

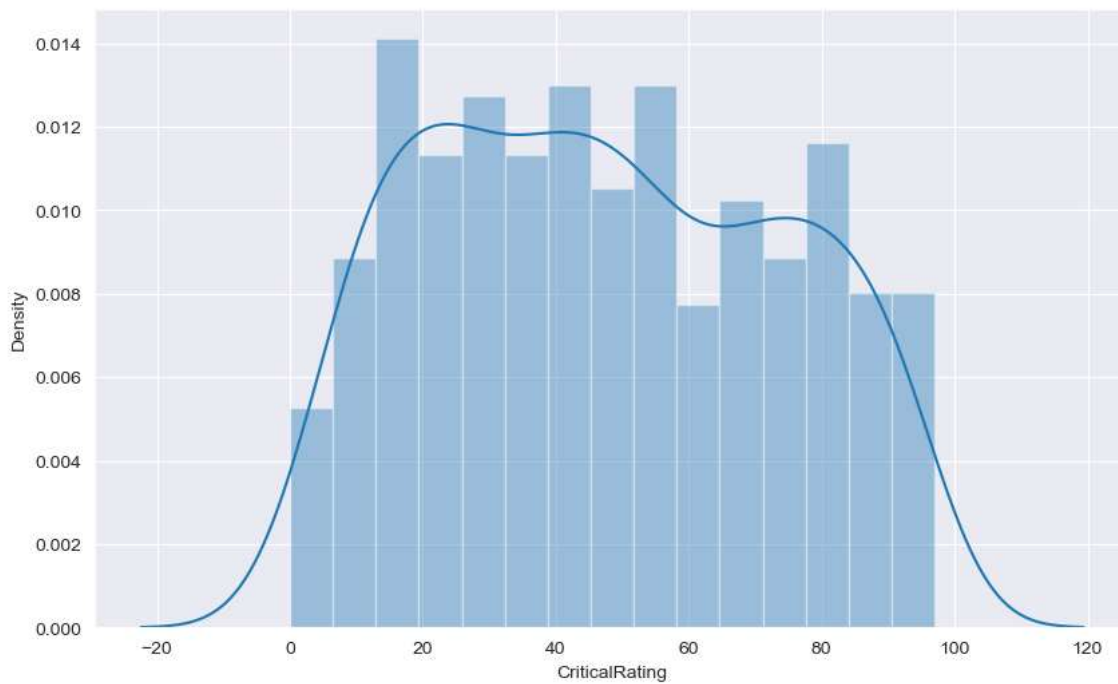
```
j=sns.jointplot(data=df, x='CriticalRating',y='Year of release',kind='hex')
```





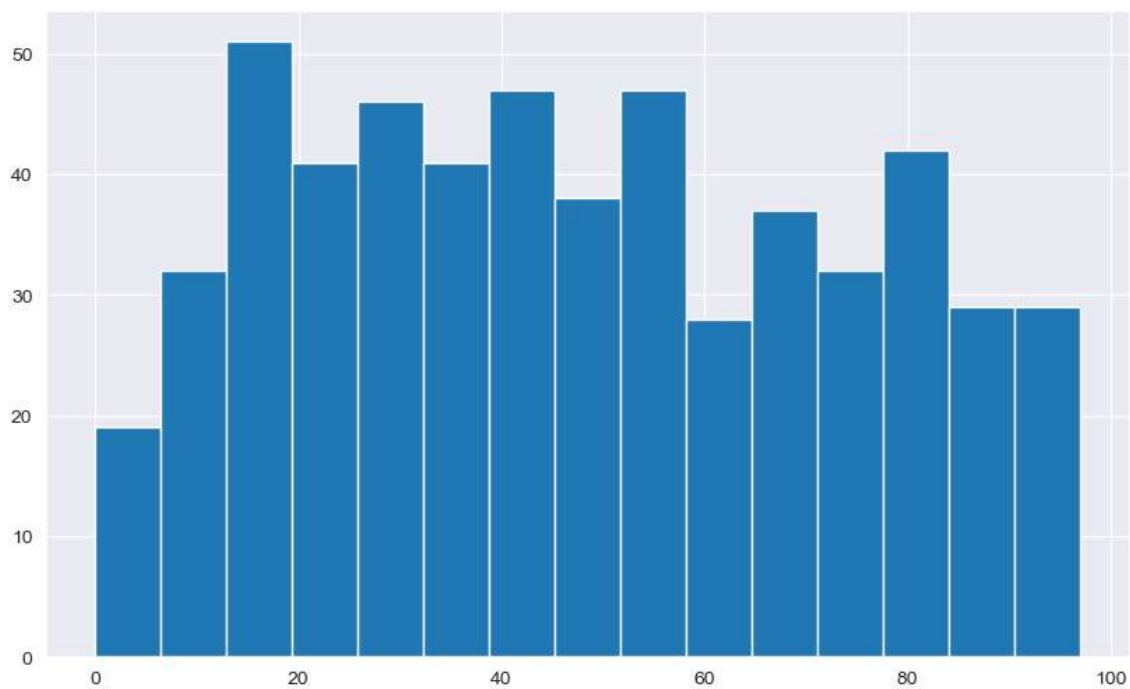
In [49]:

```
m2=sns.distplot(df.CriticalRating,bins=15)
```



In [50]:

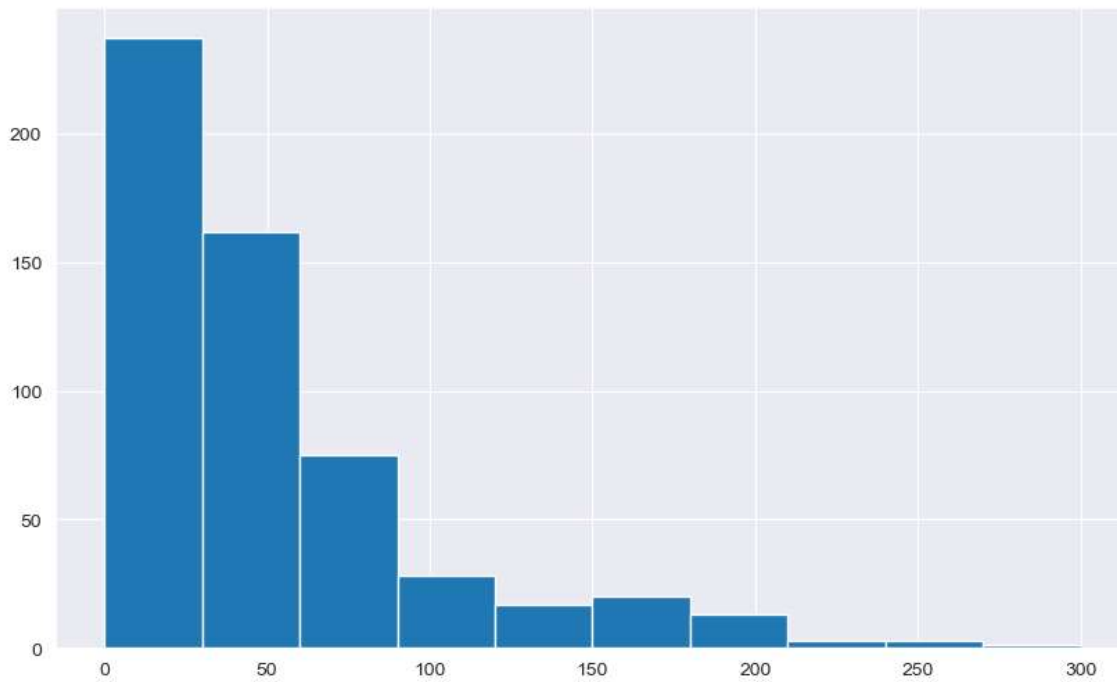
```
n1=plt.hist(df.CriticalRating,bins=15)
```





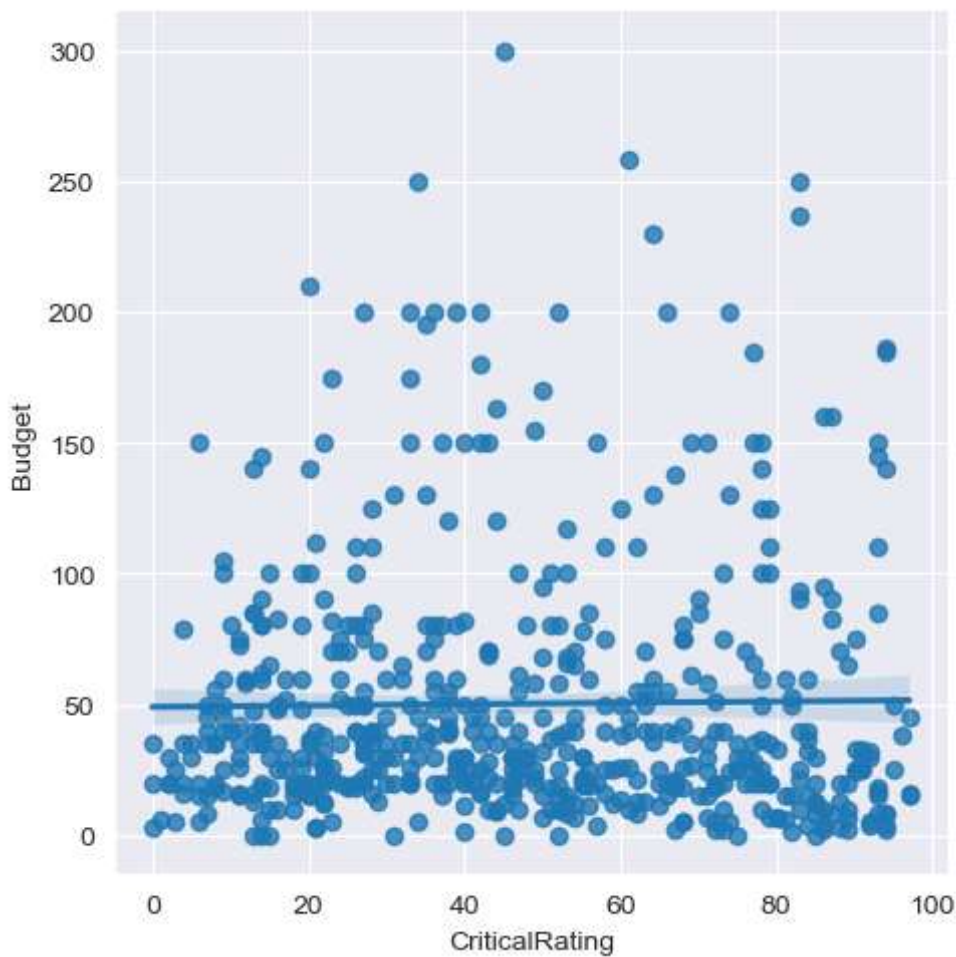
In [51]:

```
plt.hist(df.Budget)  
plt.show()
```



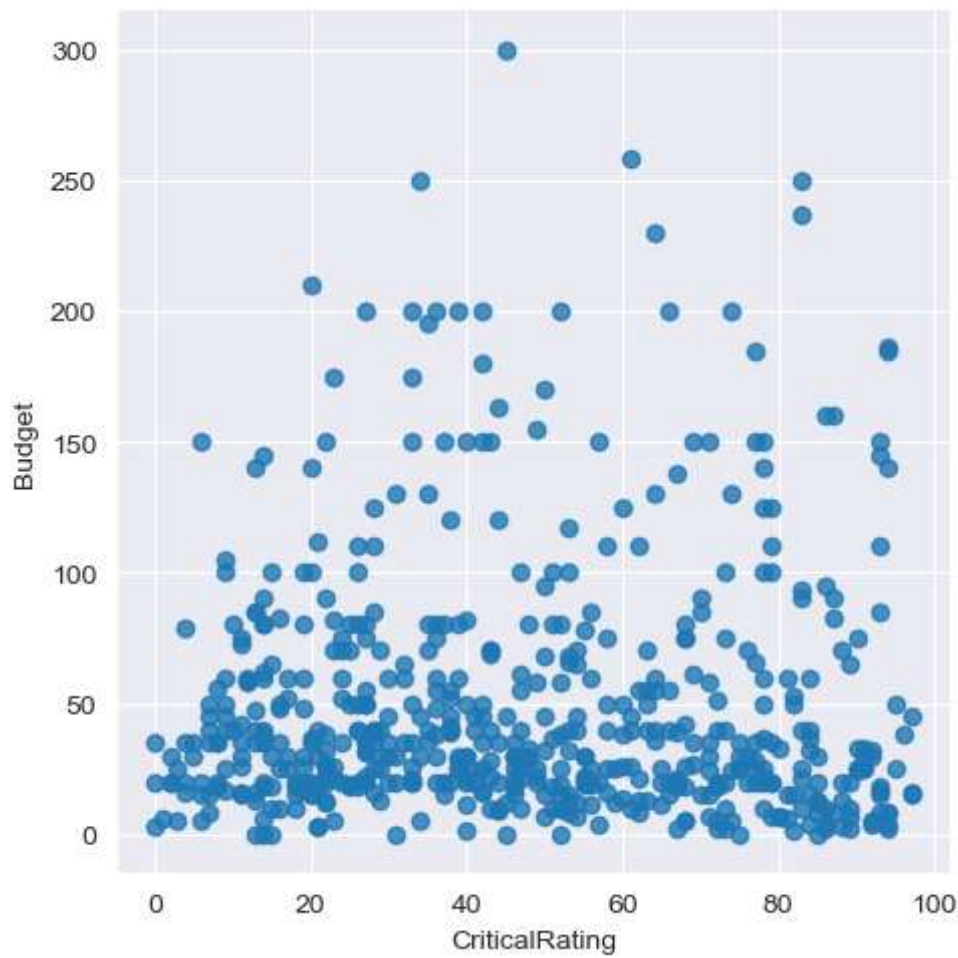
In [52]:

```
vis1=sns.lmplot(data=df,x='CriticalRating',y='Budget',fit_reg=True)
```



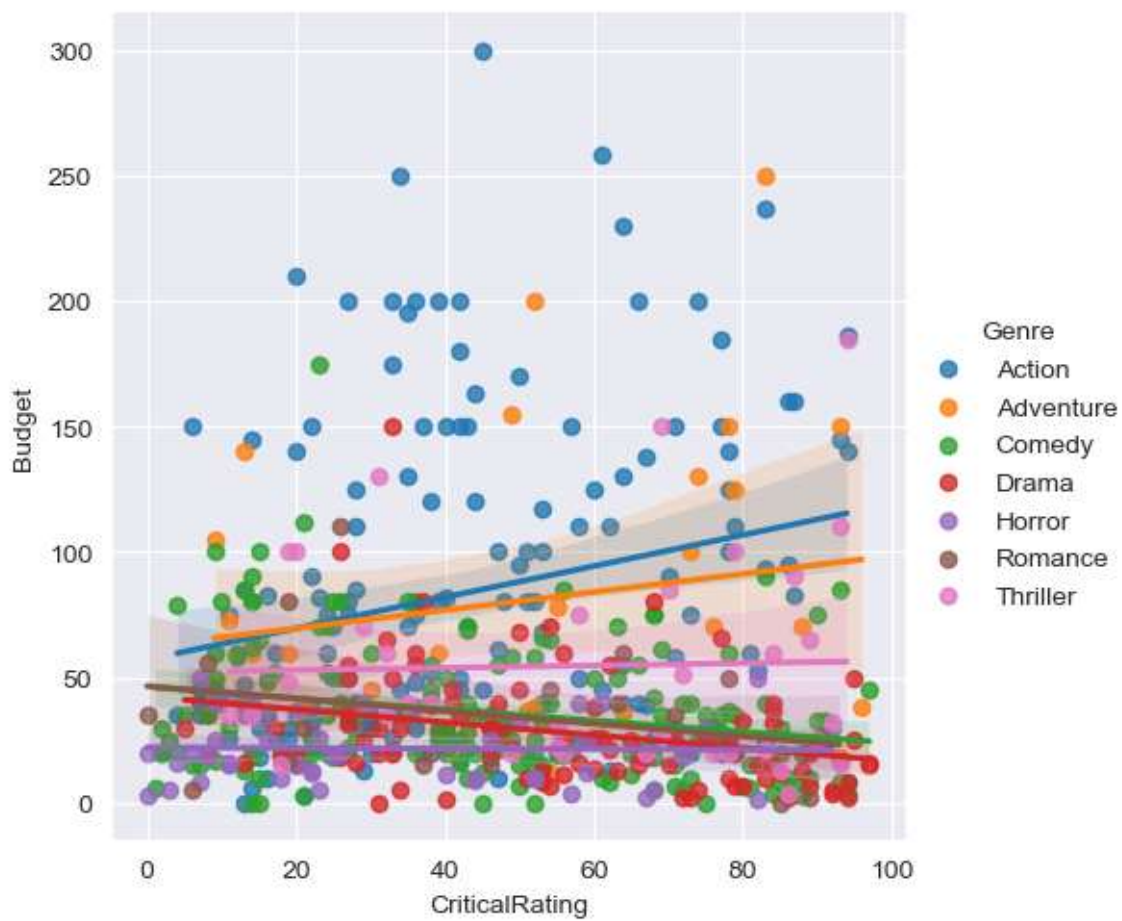
In [53]:

```
vis1=sns.lmplot(data=df,x='CriticalRating',y='Budget',fit_reg=False)
```



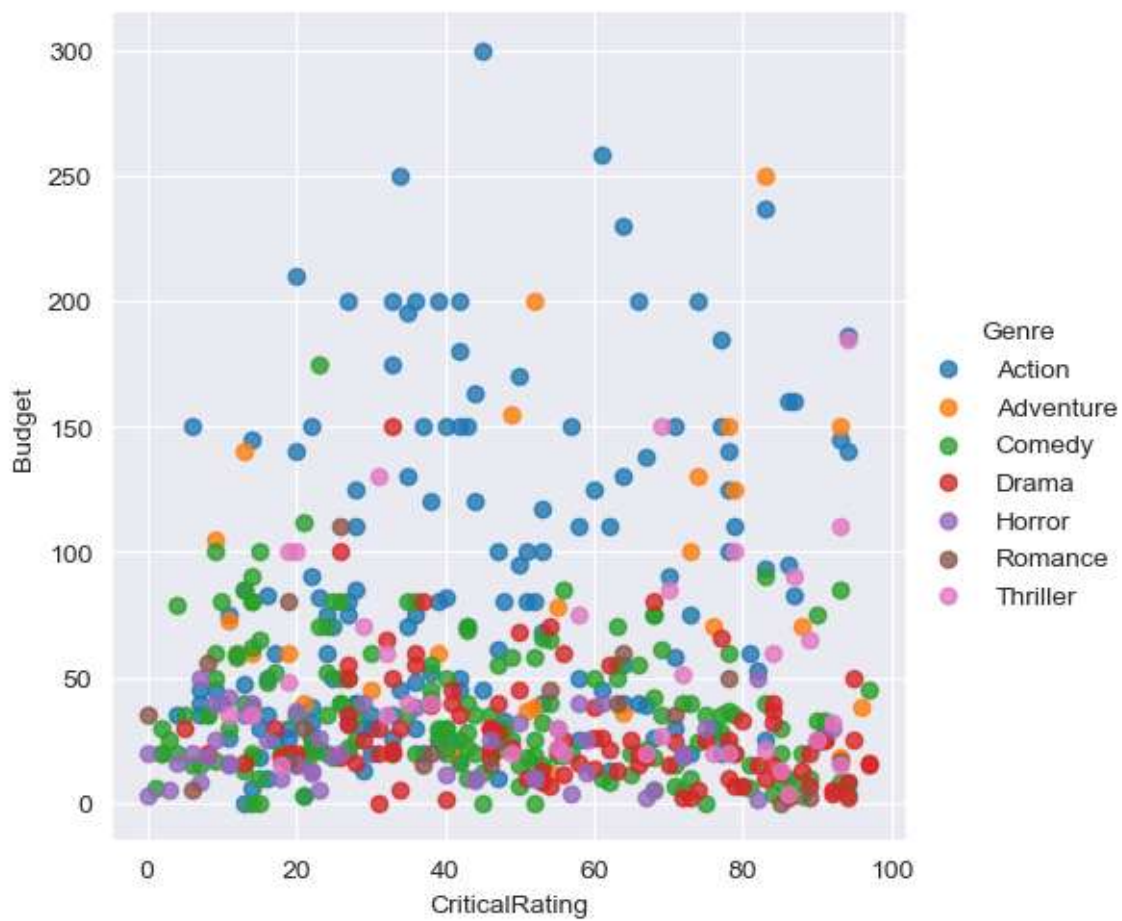
In [54]:

```
vis1=sns.lmplot(data=df,x='CriticalRating',y='Budget',fit_reg=True,hue='Genre')
```



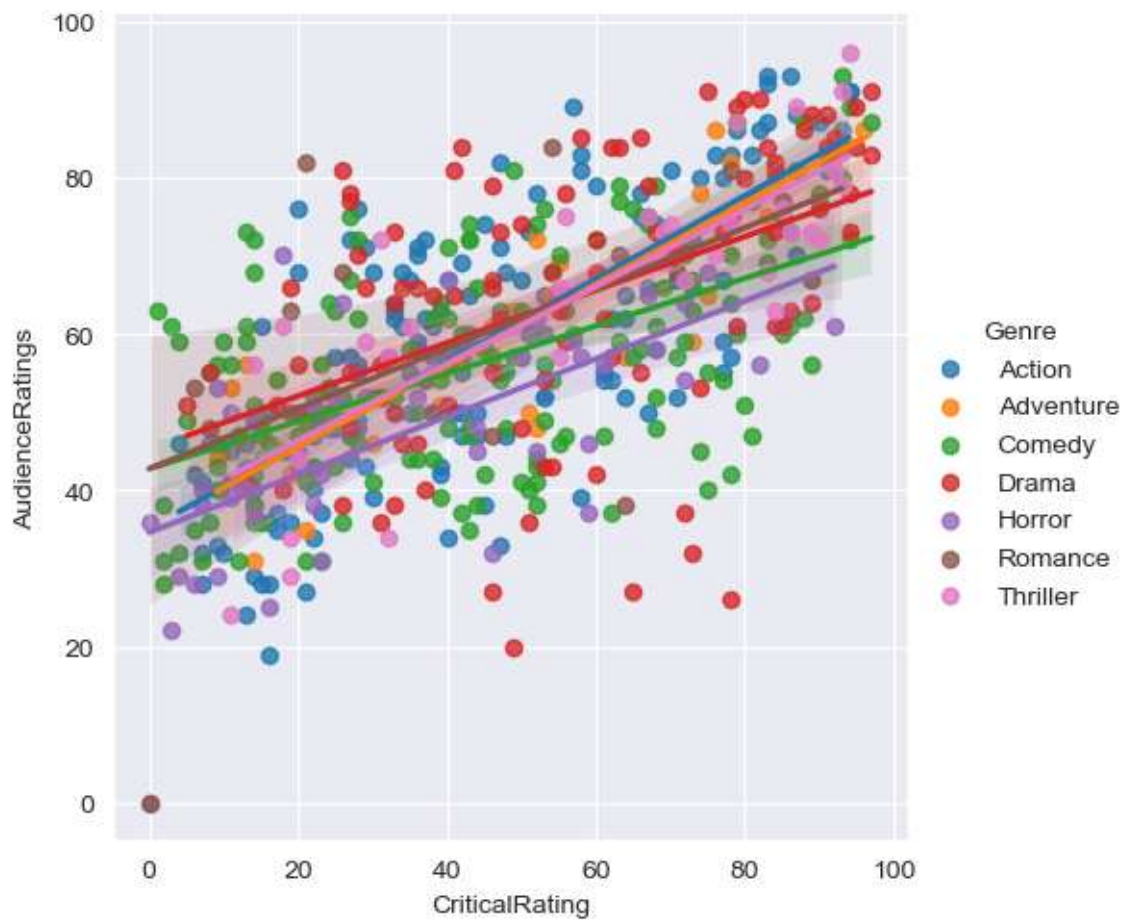
In [55]:

```
vis1=sns.lmplot(data=df,x='CriticalRating',y='Budget',fit_reg=False,hue='Genre')
```



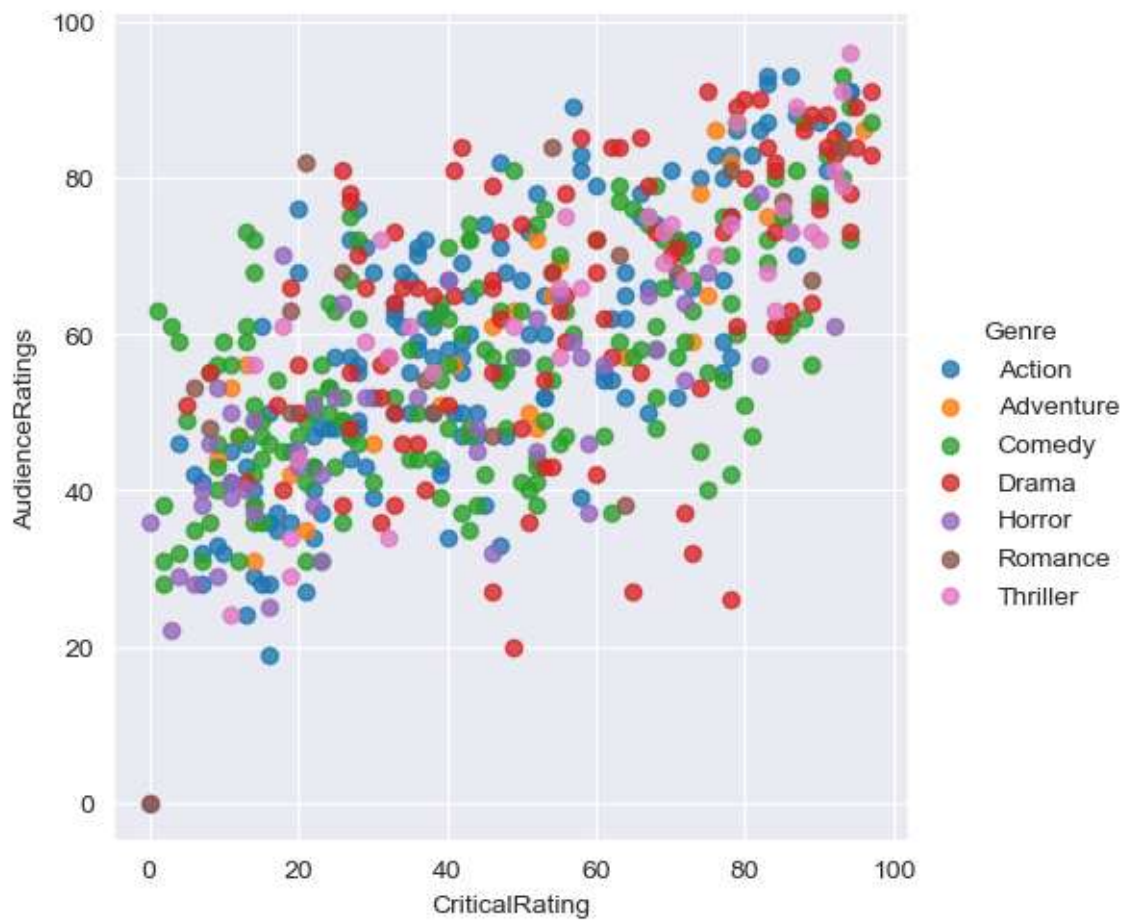
In [56]:

```
vis1=sns.lmplot(data=df,x='CriticalRating',y='AudienceRatings',fit_reg=True,hue='Genre')
```



In [57]:

```
vis1=sns.lmplot(data=df,x='CriticalRating',y='AudienceRatings',fit_reg=False,hue='Genre')
```

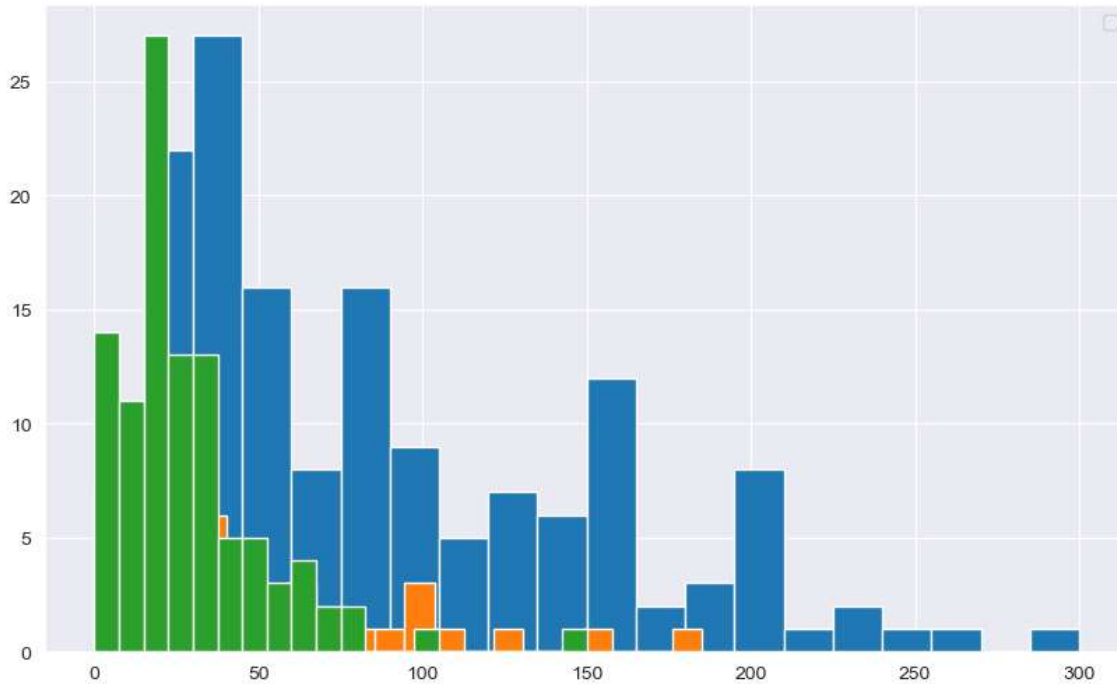




In [62]:

```
plt.hist(df[df.Genre == 'Action'].Budget, bins = 20)  
plt.hist(df[df.Genre == 'Thriller'].Budget, bins = 20)  
plt.hist(df[df.Genre == 'Drama'].Budget, bins = 20)  
plt.legend()  
plt.show()
```

No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.



In [68]:

```
sns.set_style('dark')
```

In [69]:

```
vis1 = sns.lmplot(data=df, x='CriticalRating', y='AudienceRatings',\n                  fit_reg=False, hue = 'Genre', aspect=1)
```

