

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

```
In [2]: df1=pd.read_csv(r'C:\Users\sneha\Downloads\Movie+Assignment+Data.csv')
```

```
In [3]: df1.head()
```

```
Out[3]:
```

	Title	title_year	budget	Gross	actor_1_name	actor_2_name	actor_3_name	actor_
0	La La Land	2016	30000000	151101803	Ryan Gosling	Emma Stone	Amiée Conn	
1	Zootopia	2016	150000000	341268248	Ginnifer Goodwin	Jason Bateman	Idris Elba	
2	Lion	2016	12000000	51738905	Dev Patel	Nicole Kidman	Rooney Mara	
3	Arrival	2016	47000000	100546139	Amy Adams	Jeremy Renner	Forest Whitaker	
4	Manchester by the Sea	2016	9000000	47695371	Casey Affleck	Michelle Williams	Kyle Chandler	

5 rows × 62 columns

```
In [4]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 62 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Title                                100 non-null    object
1   title_year                          100 non-null    int64
2   budget                              100 non-null    int64
3   Gross                               100 non-null    int64
4   actor_1_name                        100 non-null    object
5   actor_2_name                        100 non-null    object
6   actor_3_name                        100 non-null    object
7   actor_1_facebook_likes              100 non-null    int64
8   actor_2_facebook_likes              99 non-null     float64
9   actor_3_facebook_likes              98 non-null     float64
10  IMDb_rating                         100 non-null    float64
11  genre_1                             100 non-null    object
12  genre_2                             97 non-null     object
13  genre_3                             74 non-null     object
14  MetaCritic                         95 non-null     float64
15  Runtime                             100 non-null    int64
16  CVotes10                            100 non-null    int64
17  CVotes09                            100 non-null    int64
18  CVotes08                            100 non-null    int64
19  CVotes07                            100 non-null    int64
20  CVotes06                            100 non-null    int64
21  CVotes05                            100 non-null    int64
22  CVotes04                            100 non-null    int64
23  CVotes03                            100 non-null    int64
24  CVotes02                            100 non-null    int64
25  CVotes01                            100 non-null    int64
26  CVotesMale                          100 non-null    int64
27  CVotesFemale                        100 non-null    int64
28  CVotesU18                           100 non-null    int64
```

```

29  CVotesU18M          100 non-null    int64
30  CVotesU18F          100 non-null    int64
31  CVotes1829          100 non-null    int64
32  CVotes1829M         100 non-null    int64
33  CVotes1829F         100 non-null    int64
34  CVotes3044          100 non-null    int64
35  CVotes3044M         100 non-null    int64
36  CVotes3044F         100 non-null    int64
37  CVotes45A           100 non-null    int64
38  CVotes45AM          100 non-null    int64
39  CVotes45AF          100 non-null    int64
40  CVotes1000          100 non-null    int64
41  CVotesUS            100 non-null    int64
42  CVotesnUS           100 non-null    int64
43  VotesM              100 non-null    float64
44  VotesF              100 non-null    float64
45  VotesU18            100 non-null    float64
46  VotesU18M           100 non-null    float64
47  VotesU18F           100 non-null    float64
48  Votes1829           100 non-null    float64
49  Votes1829M          100 non-null    float64
50  Votes1829F          100 non-null    float64
51  Votes3044           100 non-null    float64
52  Votes3044M          100 non-null    float64
53  Votes3044F          100 non-null    float64
54  Votes45A            100 non-null    float64
55  Votes45AM           100 non-null    float64
56  Votes45AF           100 non-null    float64
57  Votes1000           100 non-null    float64
58  VotesUS             100 non-null    float64
59  VotesnUS            100 non-null    float64
60  content_rating       100 non-null    object
61  Country              100 non-null    object
dtypes: float64(21), int64(32), object(9)
memory usage: 48.6+ KB

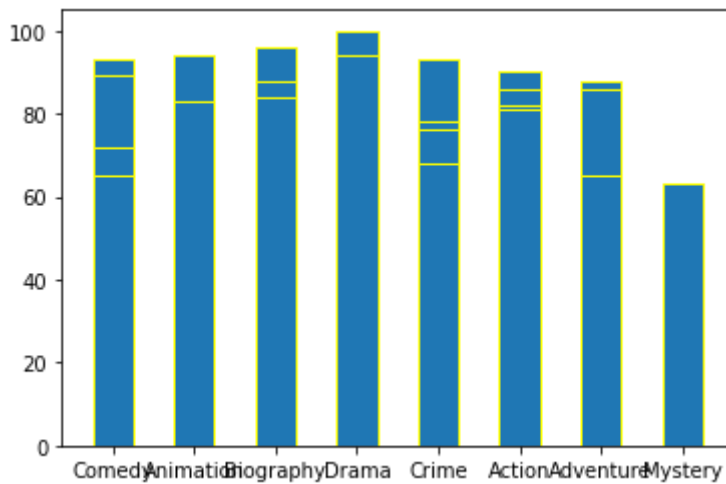
```

## Bar Plot

```
In [5]: df2=df1.groupby(by=['genre_1'])['CVotes10'].sum()
df2
```

```
Out[5]: genre_1
Action      2928407
Adventure   1058779
Animation    681562
Biography    666831
Comedy       371217
Crime        383290
Drama       1080725
Mystery     150405
Name: CVotes10, dtype: int64
```

```
In [6]: # using matplotlib
plt.bar(df1['genre_1'],df1['MetaCritic'],width=0.5,edgecolor="yellow")
plt.show()
```



```
In [7]: df1[['genre_1', 'MetaCritic']].head(15).value_counts()
```

```
Out[7]: genre_1    MetaCritic
Drama          96.0         1
           81.0         1
Crime          88.0         1
Comedy         93.0         1
Biography      69.0         1
Animation      92.0         1
           78.0         1
           71.0         1
Adventure      66.0         1
Action         81.0         1
           78.0         1
           75.0         1
           74.0         1
           72.0         1
           69.0         1
dtype: int64
```

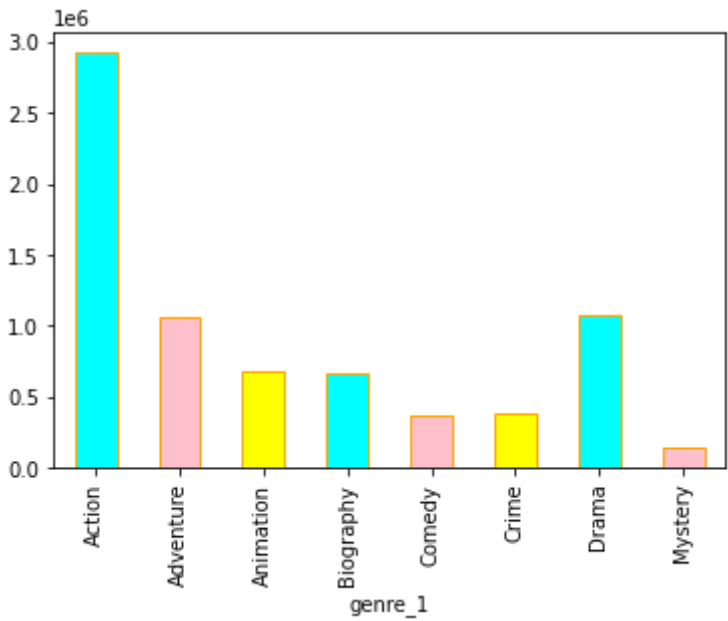
```
In [8]: df1.head(15).groupby(by=['genre_1'])['MetaCritic'].sum()
```

```
Out[8]: genre_1
Action          449.0
Adventure        66.0
Animation        241.0
Biography         69.0
Comedy           93.0
Crime            88.0
Drama           177.0
Name: MetaCritic, dtype: float64
```

```
In [9]: plt.figure(figsize=[10,10])
plt.bar(data=df1.head(15),x='genre_1',height='MetaCritic',edgecolor='blue',color=['c
plt.title("Title here \n",fontdict={'fontsize':20,'fontweight':5,'color':'green'})
ticks=np.arange(0,100,5)
labels=[i for i in ticks]
plt.yticks(ticks,labels)
plt.xlabel('xlabel')
plt.ylabel('ylabel')
plt.grid(axis='y')
plt.show()
```



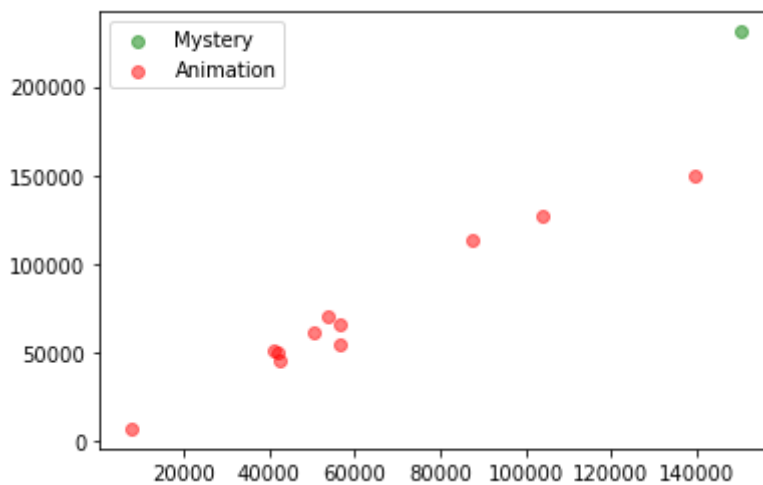
```
In [10]: df1.groupby(by=['genre_1'])['CVotes10'].sum().plot.bar(edgecolor='orange',color=['cy', 'pink', 'yellow', 'cyan', 'pink', 'yellow', 'pink'])
plt.show()
```



## Scatter Plot

```
In [11]: plt.scatter(data=df1[df1['genre_1']=="Mystery"],x='CVotes10',y='CVotes09',alpha=0.5,
plt.scatter(data=df1[df1['genre_1']=="Animation"],x='CVotes10',y='CVotes09',alpha=0.5,
plt.legend()
```

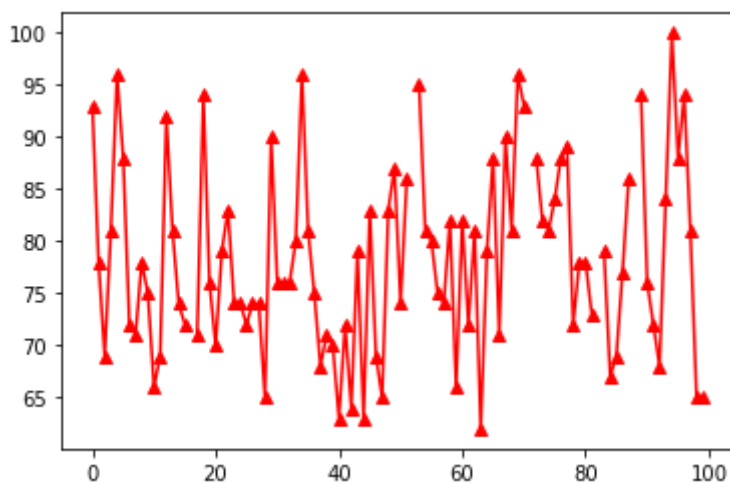
Out[11]: <matplotlib.legend.Legend at 0x240dd6bbe80>



## Line Graps and Histograms

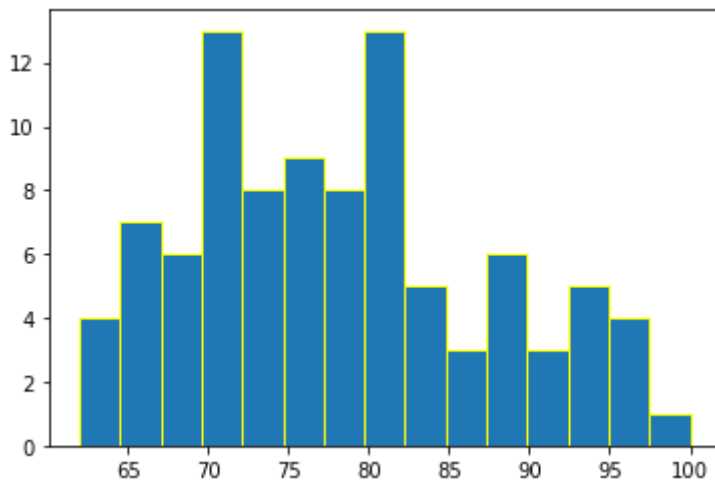
```
In [12]: plt.plot(df1['MetaCritic'],'red',marker='^')
```

Out[12]: [<matplotlib.lines.Line2D at 0x240dd6a2070>]



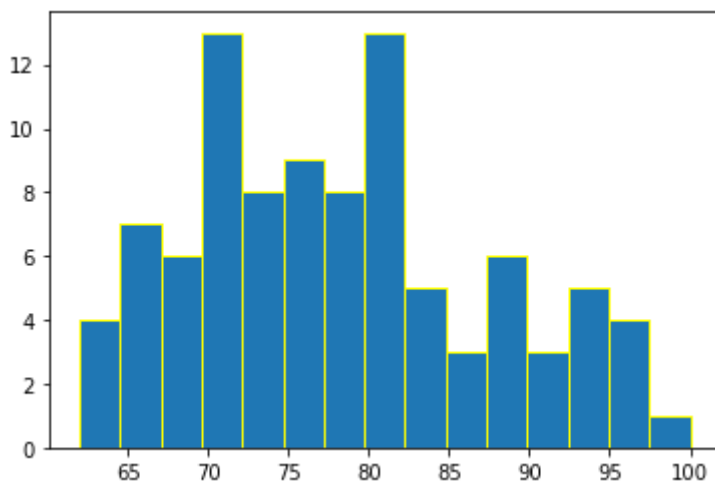
```
In [13]: plt.hist(df1['MetaCritic'],bins=15,edgecolor='yellow')
```

```
Out[13]: (array([ 4.,  7.,  6., 13.,  8.,  9.,  8., 13.,  5.,  3.,  6.,  3.,  5.,
  4.,  1.]),
array([ 62.         ,  64.53333333,  67.06666667,  69.6        ,
  72.13333333,  74.66666667,  77.2        ,  79.73333333,
  82.26666667,  84.8        ,  87.33333333,  89.86666667,
  92.4        ,  94.93333333,  97.46666667, 100.        ]),
<BarContainer object of 15 artists>)
```



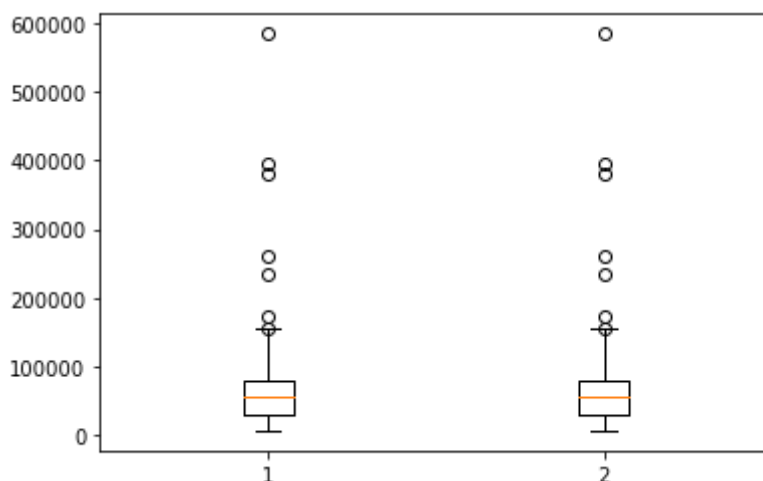
```
In [14]: plt.hist(df1['MetaCritic'],bins=15,edgecolor='yellow')
```

```
Out[14]: (array([ 4.,  7.,  6., 13.,  8.,  9.,  8., 13.,  5.,  3.,  6.,  3.,  5.,
        4.,  1.]),
  array([ 62.         ,  64.53333333,  67.06666667,  69.6         ,
        72.13333333,  74.66666667,  77.2         ,  79.73333333,
        82.26666667,  84.8         ,  87.33333333,  89.86666667,
        92.4         ,  94.93333333,  97.46666667, 100.         ]),
  <BarContainer object of 15 artists>)
```



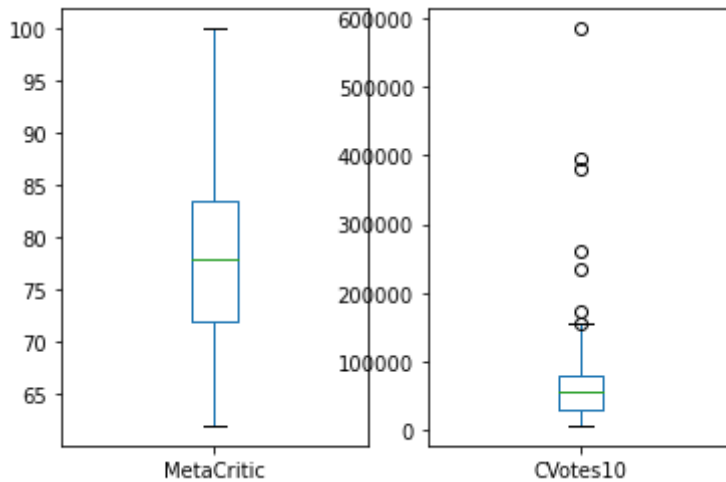
## BoxPlots

```
In [15]: #via matplotlib
plt.boxplot(df1[['CVotes10','CVotes10']])
plt.show()
```



```
In [16]: #via pandas
plt.figure(figsize=[10,7])
df1[['MetaCritic','CVotes10']].plot(kind='box',subplots=True)
```

```
Out[16]: MetaCritic      AxesSubplot(0.125,0.125;0.352273x0.755)
CVotes10      AxesSubplot(0.547727,0.125;0.352273x0.755)
dtype: object
<Figure size 720x504 with 0 Axes>
```



```
In [17]: df1['MetaCritic'].quantile(0.5)
```

```
Out[17]: 78.0
```

```
In [18]: df1.describe()
```

```
Out[18]:
```

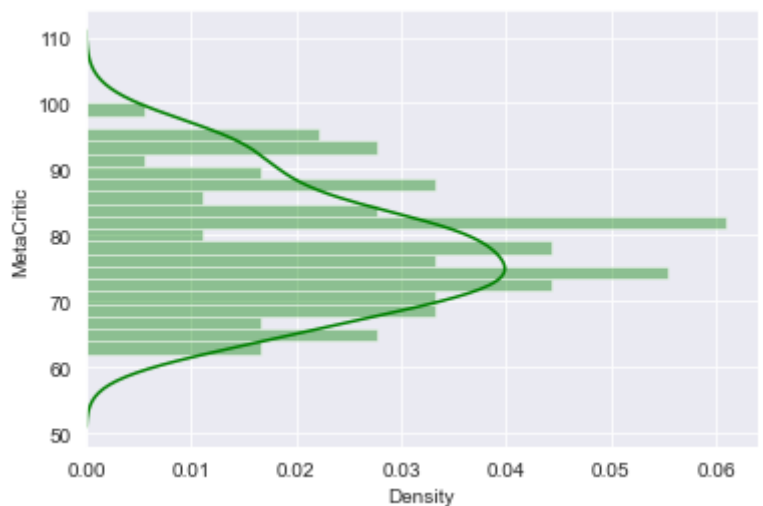
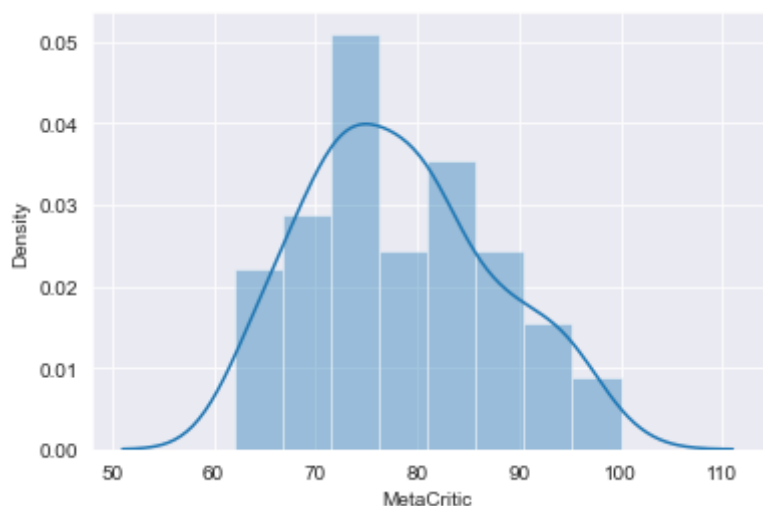
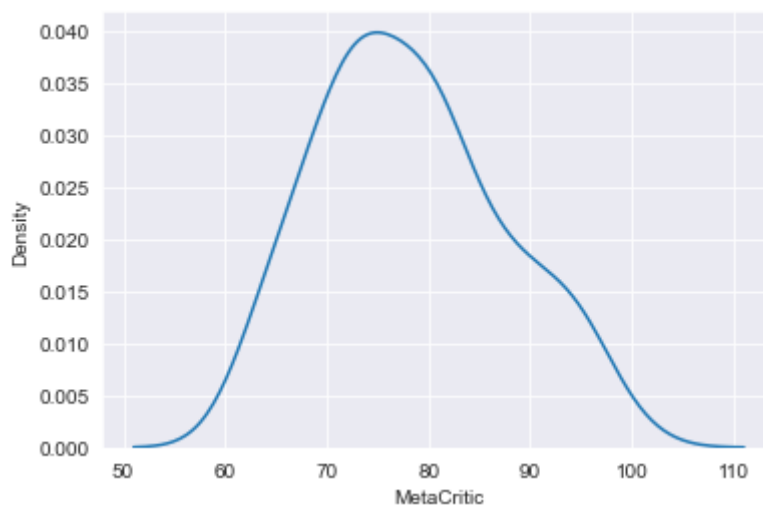
	title_year	budget	Gross	actor_1_facebook_likes	actor_2_facebook_likes	actc
<b>count</b>	100.000000	1.000000e+02	1.000000e+02	100.000000	99.000000	
<b>mean</b>	2012.820000	7.838400e+07	1.468679e+08	13407.270000	7377.303030	
<b>std</b>	1.919491	7.445295e+07	1.454004e+08	10649.037862	13471.568216	
<b>min</b>	2010.000000	3.000000e+06	2.238380e+05	39.000000	12.000000	
<b>25%</b>	2011.000000	1.575000e+07	4.199752e+07	1000.000000	580.000000	
<b>50%</b>	2013.000000	4.225000e+07	1.070266e+08	13000.000000	1000.000000	
<b>75%</b>	2014.000000	1.500000e+08	2.107548e+08	20000.000000	11000.000000	
<b>max</b>	2016.000000	2.600000e+08	9.366622e+08	35000.000000	96000.000000	

8 rows × 53 columns

## Seaborn

```
In [19]: import seaborn as sns
```

```
In [20]: sns.set_style("darkgrid")
sns.distplot(df1['MetaCritic'],kde=True,hist=False)
plt.show()
sns.distplot(df1['MetaCritic'],kde=True,hist=True)
plt.show()
sns.distplot(df1['MetaCritic'],kde=True,hist=True,vertical=True,bins=20,color='g')
plt.show()
```



In [21]: `plt.style.available`

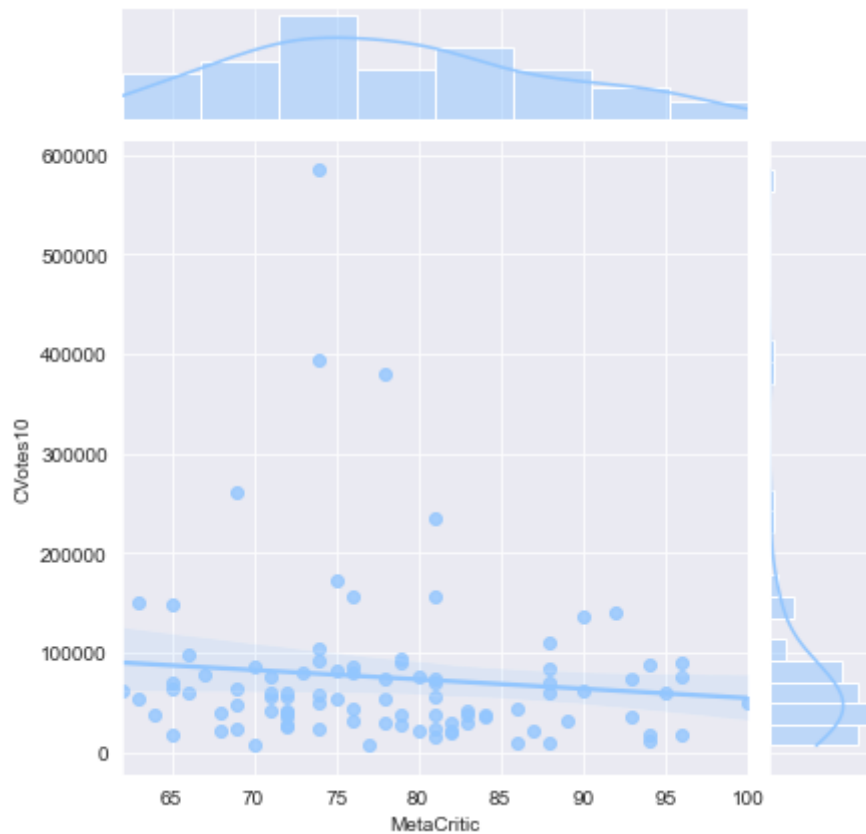
Out[21]: ['Solarize\_Light2',  
'\_classic\_test\_patch',  
'bmh',  
'classic',  
'dark\_background',  
'fast',  
'fivethirtyeight',  
'ggplot',  
'grayscale',  
'seaborn',  
'seaborn-bright',  
'seaborn-colorblind',  
'seaborn-dark',



```
'seaborn-dark-palette',
'seaborn-darkgrid',
'seaborn-deep',
'seaborn-muted',
'seaborn-notebook',
'seaborn-paper',
'seaborn-pastel',
'seaborn-poster',
'seaborn-talk',
'seaborn-ticks',
'seaborn-white',
'seaborn-whitegrid',
'tableau-colorblind10']
```

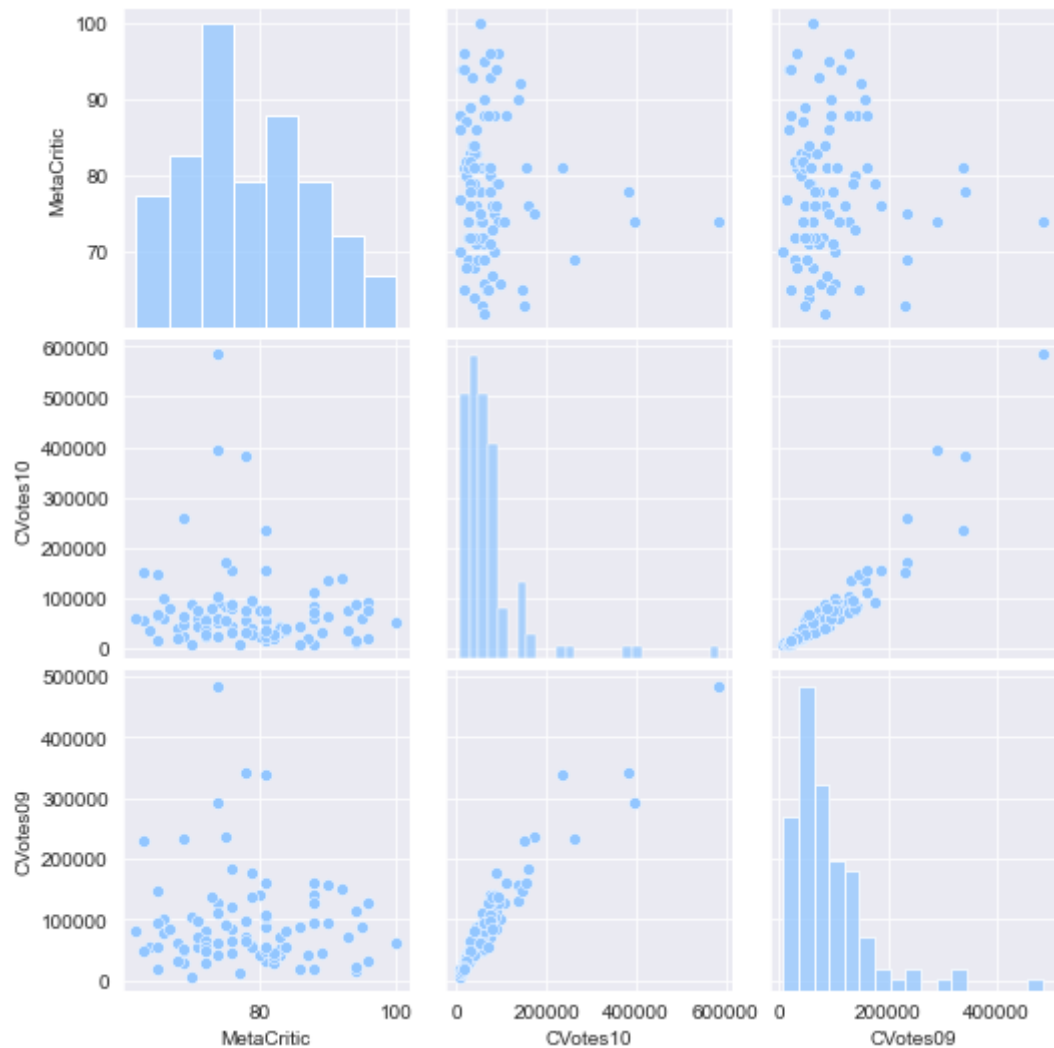
```
In [22]: plt.style.use('seaborn-pastel')
sns.jointplot(df1['MetaCritic'],df1['CVotes10'],kind='reg')
```

```
Out[22]: <seaborn.axisgrid.JointGrid at 0x240dd6b1fa0>
```



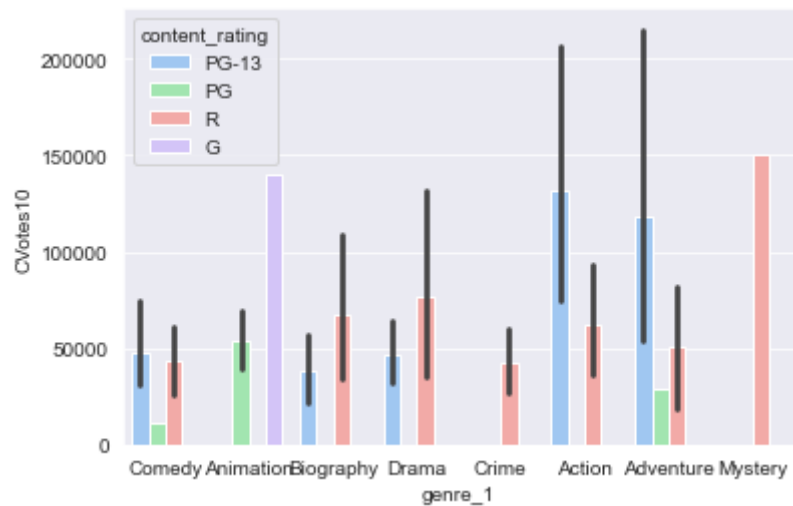
```
In [23]: sns.pairplot(df1[['MetaCritic','CVotes10','CVotes09']])
```

```
Out[23]: <seaborn.axisgrid.PairGrid at 0x240df8d2280>
```



```
In [24]: sns.barplot(data=df1,x='genre_1',y='CVotes10',hue='content_rating',estimator=np.mean)

Out[24]: <AxesSubplot:xlabel='genre_1', ylabel='CVotes10'>
```



```
In [39]: pil=pd.pivot_table(data=df1,index='genre_1',columns='genre_2',values='CVotes10',aggf
pil

Out[39]:
```

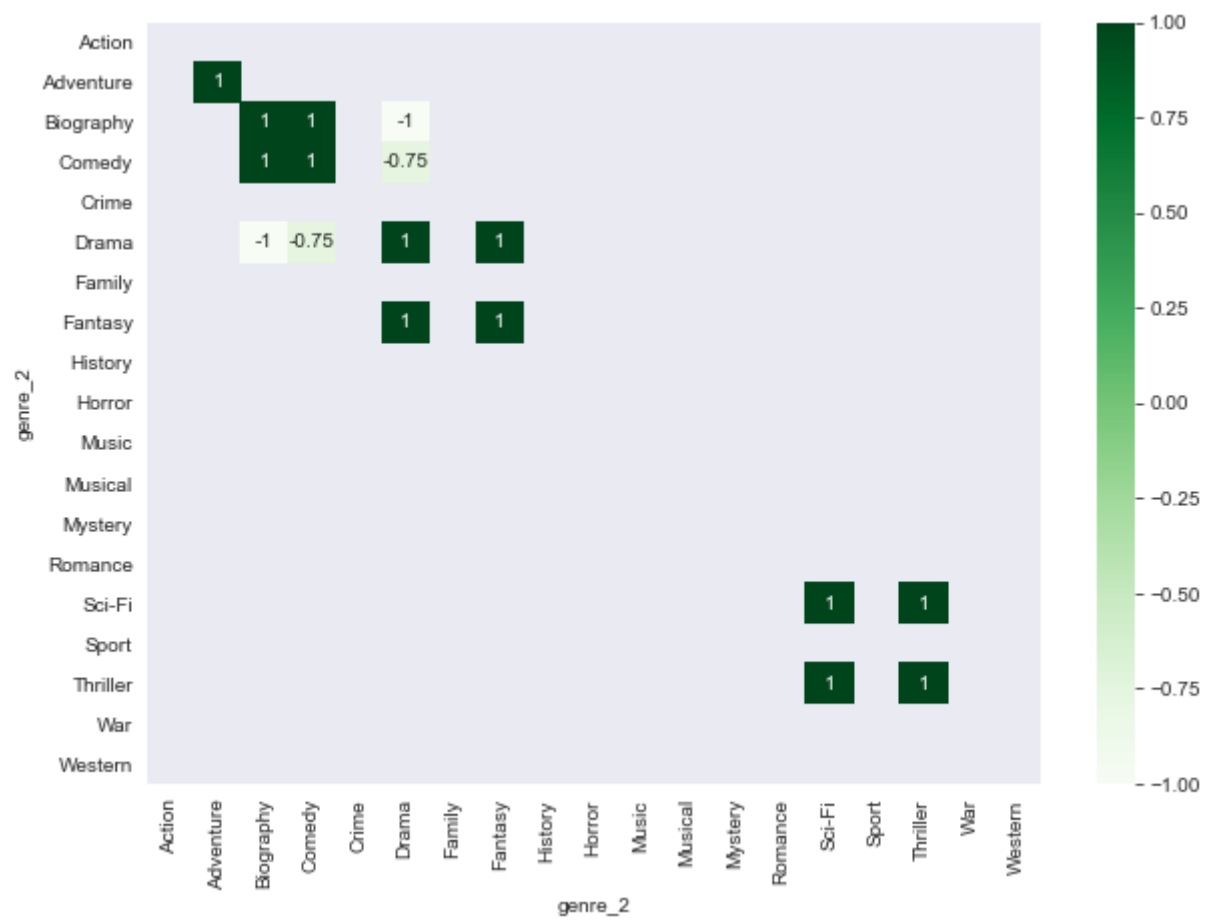
	genre_2	Action	Adventure	Biography	Comedy	Crime	Drama	Family	Fantasy	History
genre_1	Action	NaN	84943.0	37365.5	53516.0	19576.0	38666.0	NaN	NaN	NaN

genre_2	Action	Adventure	Biography	Comedy	Crime	Drama	Family	Fantasy	History
genre_1									
Adventure	NaN	NaN	28939.0	34789.0	NaN	75560.0	68937.0	116807.0	NaN
Animation	46473.5	56474.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Biography	NaN	NaN	NaN	97269.5	NaN	37461.0	NaN	NaN	NaN
Comedy	NaN	NaN	NaN	NaN	NaN	38556.0	NaN	37002.0	NaN
Crime	NaN	NaN	NaN	NaN	NaN	35421.0	NaN	NaN	NaN
Drama	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	15757.0
Mystery	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [40]:

```
plt.figure(figsize=[10,7])
sns.heatmap(pi1.corr(),cmap='Greens',annot=True)
```

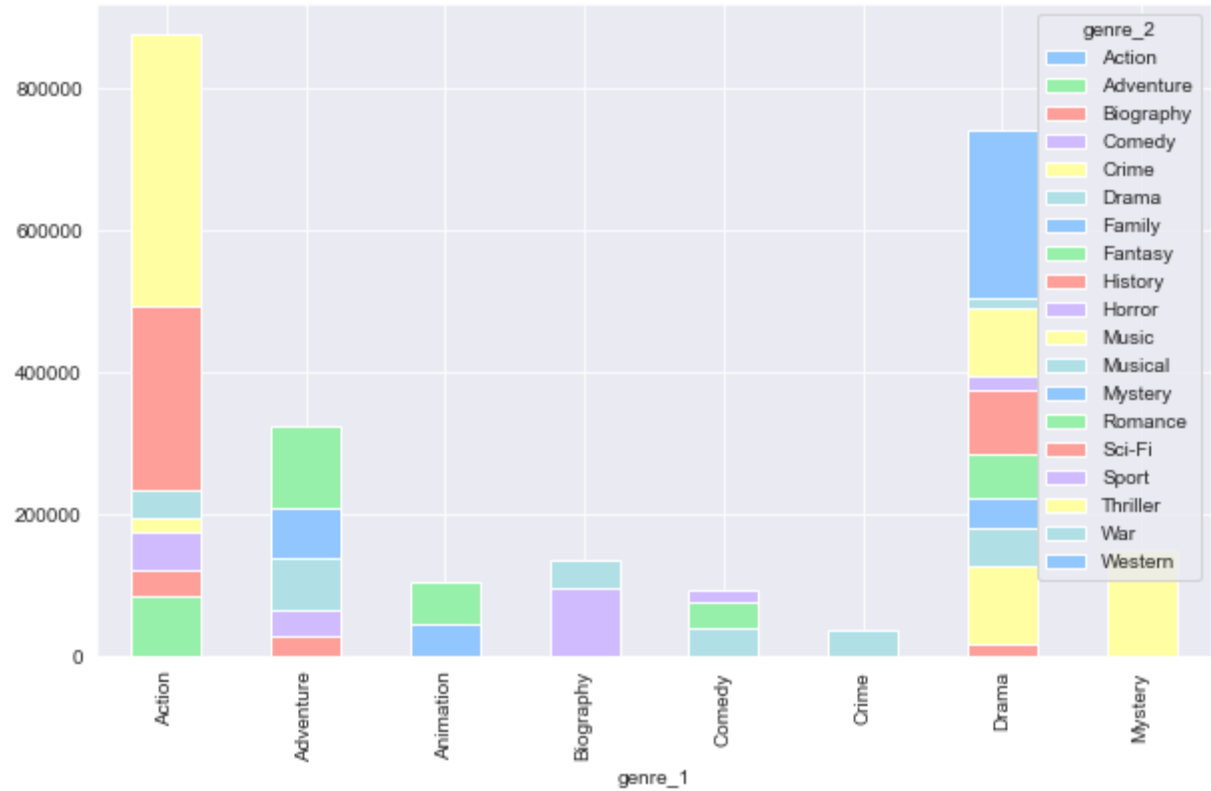
Out[40]: <AxesSubplot:xlabel='genre\_2', ylabel='genre\_2'>



In [41]:

```
pi1.plot(kind='bar',stacked=True,figsize=[10,6])
```

Out[41]: <AxesSubplot:xlabel='genre\_1'>



```
In [ ]:
```