In [3]:

```
import numpy as np
import pandas as pd
import seaborn as sns
```

In [4]:

```
df= pd.read_csv("heart.csv")
```

In [5]:

```
df.head()
```

Out[5]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

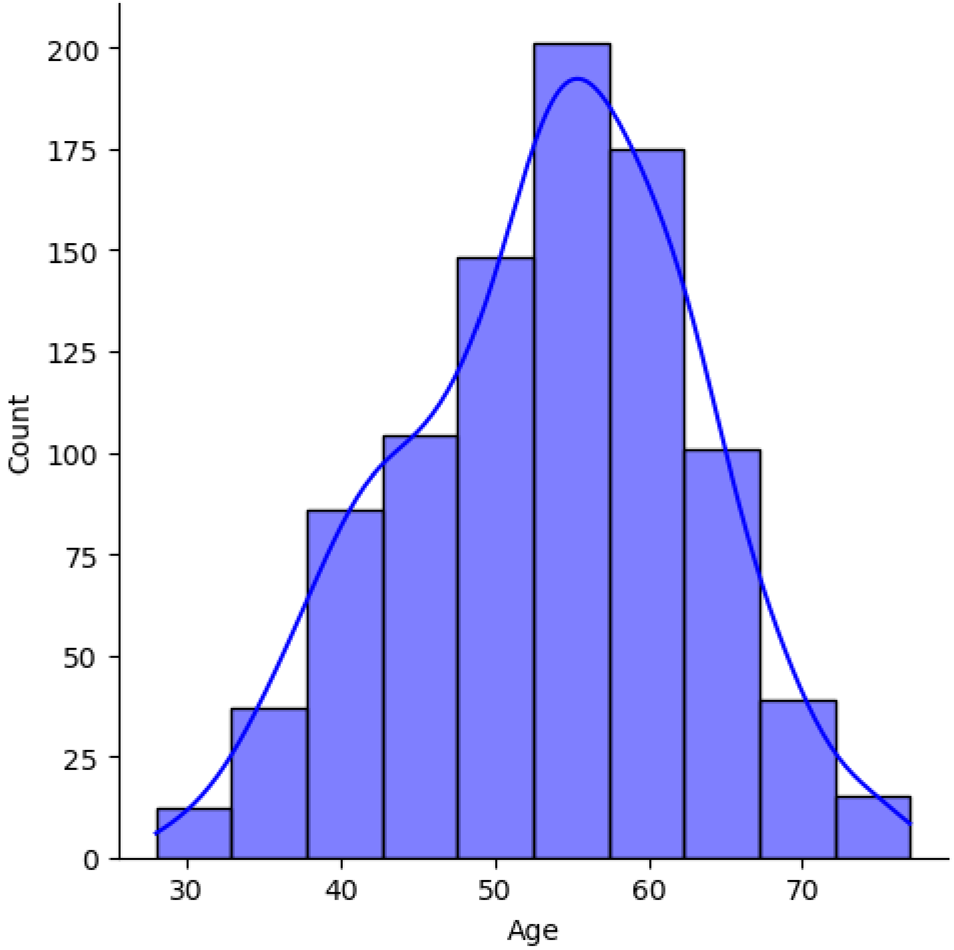
Distribution plot for numerical columns

In [7]:

```
sns.displot(df['Age'], kde= True, bins=10, color= 'blue')
```

Out[7]:

```
<seaborn.axisgrid.FacetGrid at 0x1f486b16550>
```

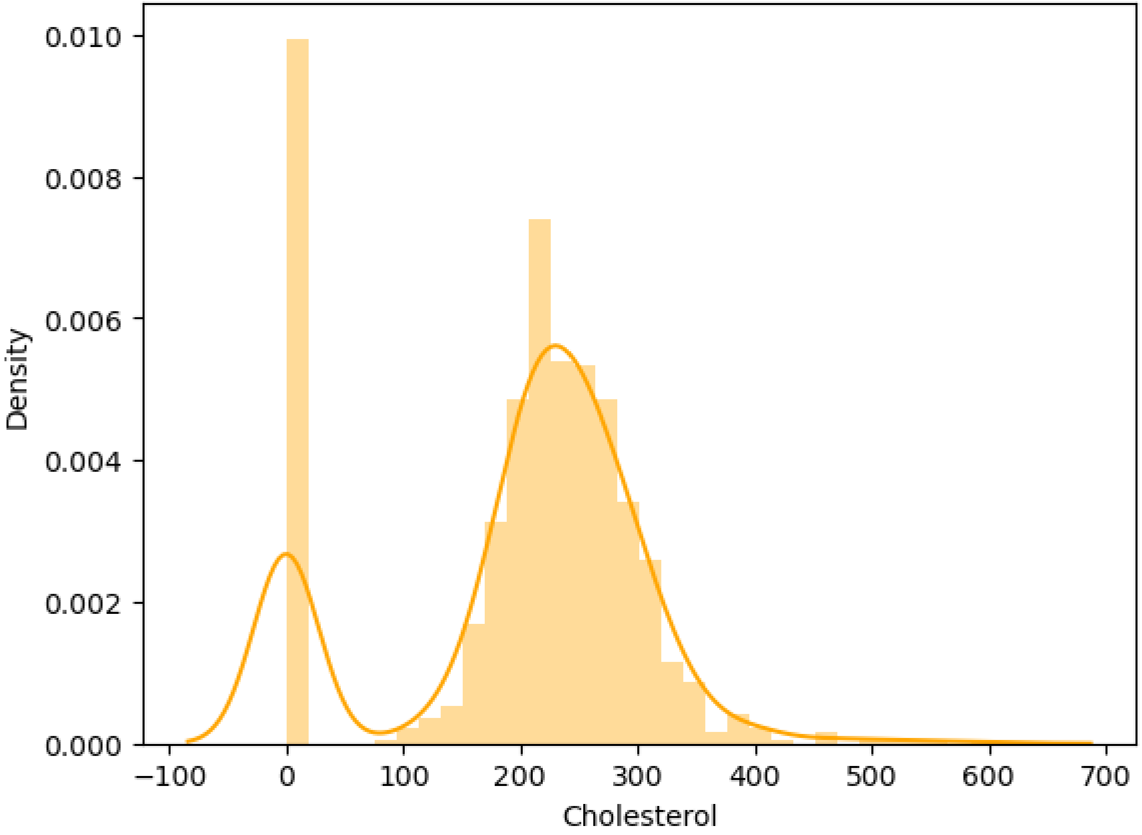


In [12]:
```
sns.distplot(df['Cholesterol'], kde= True, color= 'orange')
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17612\3171080371.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Cholesterol'], kde= True, color= 'orange')
```

Out[12]:

```
<Axes: xlabel='Cholesterol', ylabel='Density'>
```

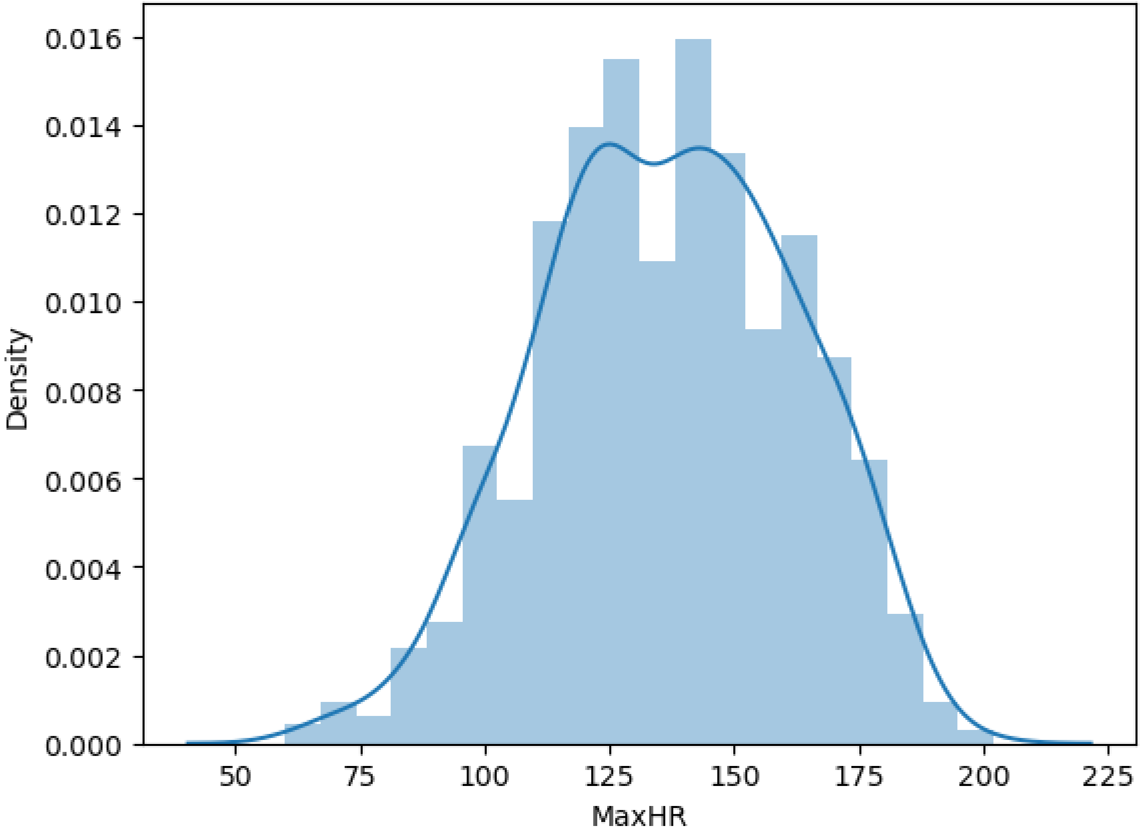


In [11]:

```
sns.distplot(df['MaxHR'], kde= True)
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17612\1529129802.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['MaxHR'], kde= True)
```

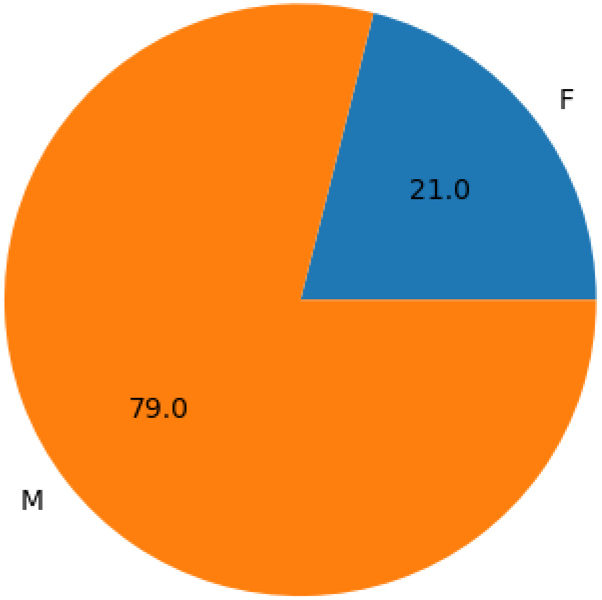
```
Out[11]: <Axes: xlabel='MaxHR', ylabel='Density'>
```



Categorical Columns

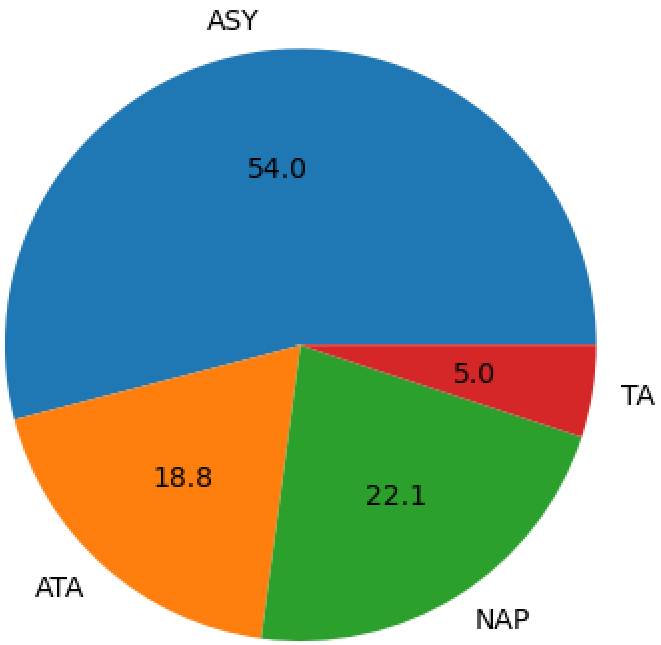
```
In [13]: df.groupby('Sex').size().plot(kind='pie', autopct='%0.1f')
```

```
Out[13]: <Axes: >
```



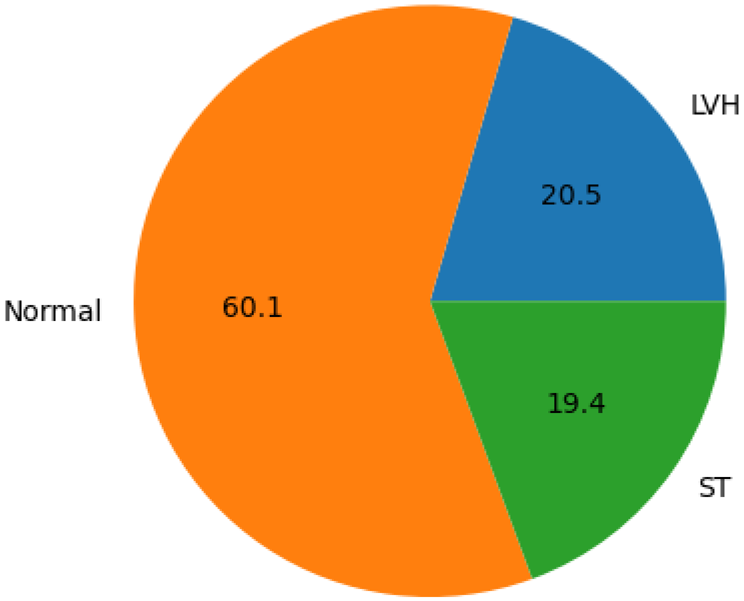
```
In [21]: df.groupby('ChestPainType').size().plot(kind='pie', autopct='%0.1f')
```

```
Out[21]: <Axes: >
```



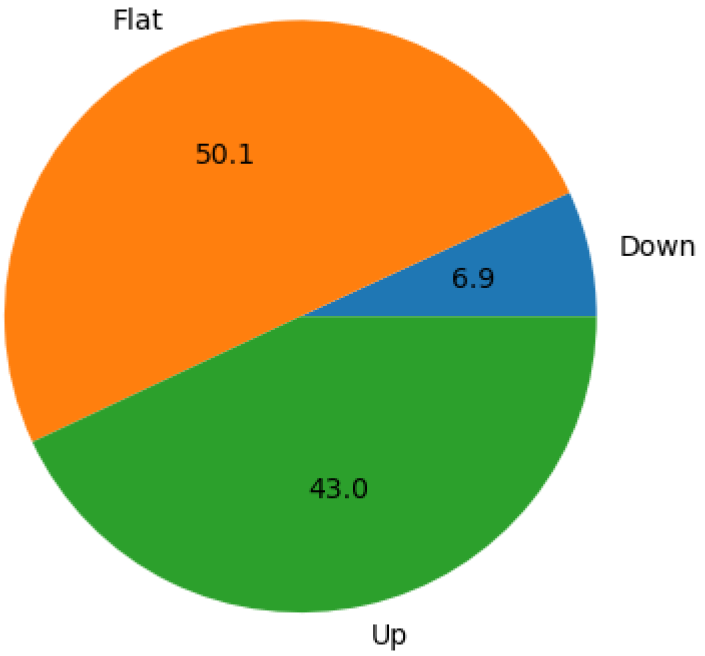
```
In [22]: df.groupby('RestingECG').size().plot(kind='pie', autopct='%0.1f')
```

```
Out[22]: <Axes: >
```



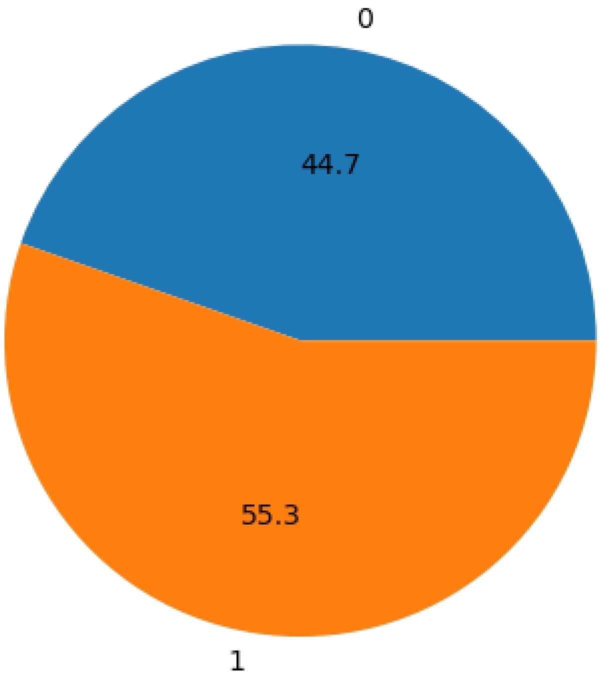
In [23]:
```
df.groupby('ST_Slope').size().plot(kind='pie', autopct='%0.1f')
```

Out[23]:
```
<Axes: >
```



```
In [24]: df.groupby('HeartDisease').size().plot(kind='pie', autopct='%0.1f')
```

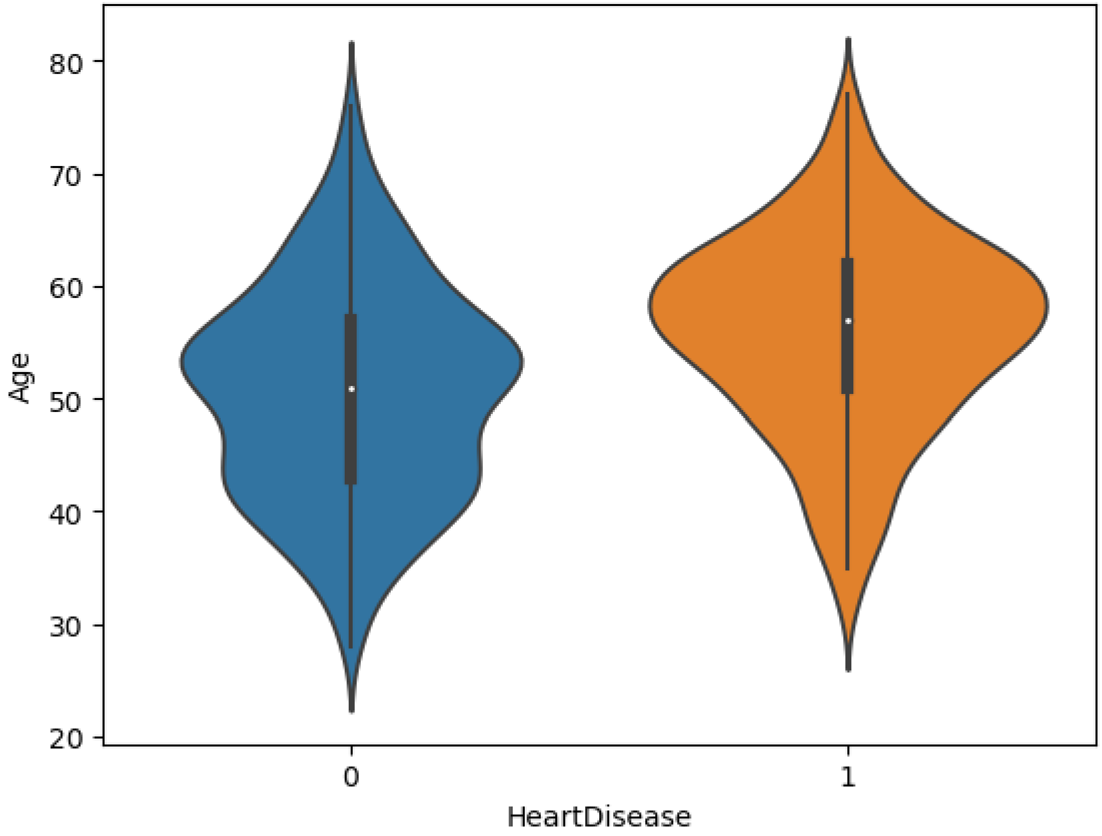
```
Out[24]: <Axes: >
```



Violin Plot

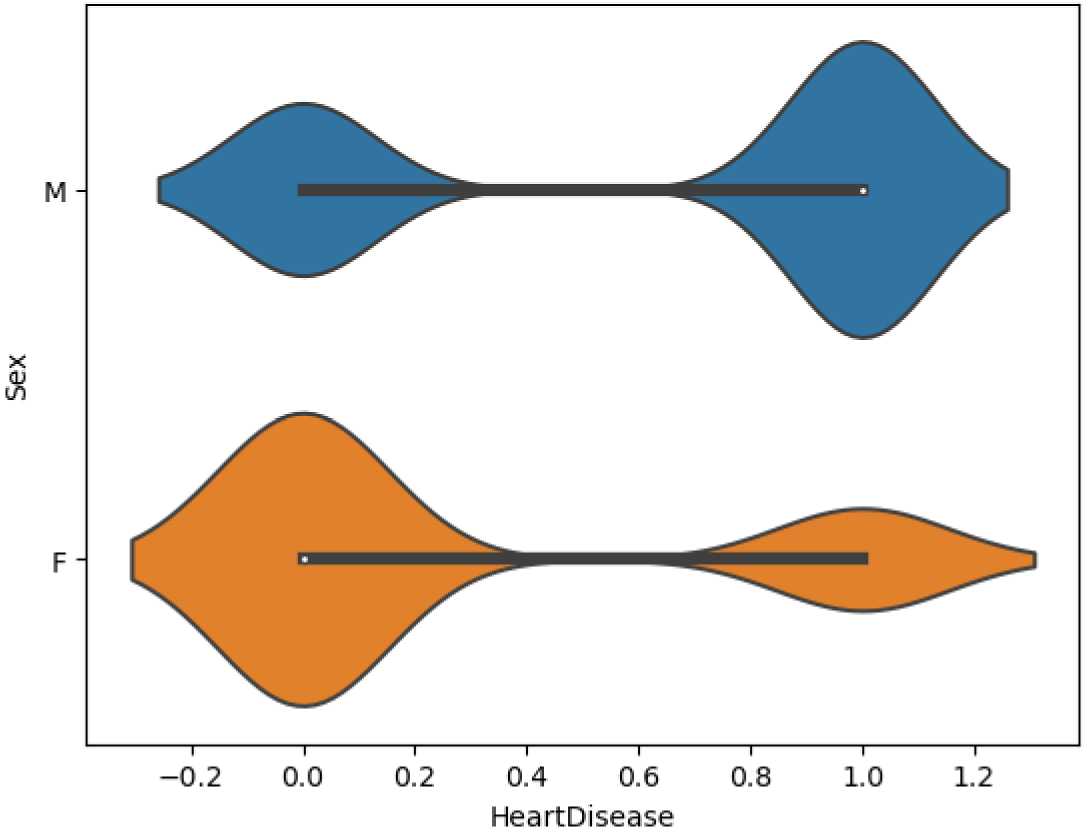
```
In [26]: sns.violinplot(y= df['Age'], x= df['HeartDisease'])
```

```
Out[26]: <Axes: xlabel='HeartDisease', ylabel='Age'>
```



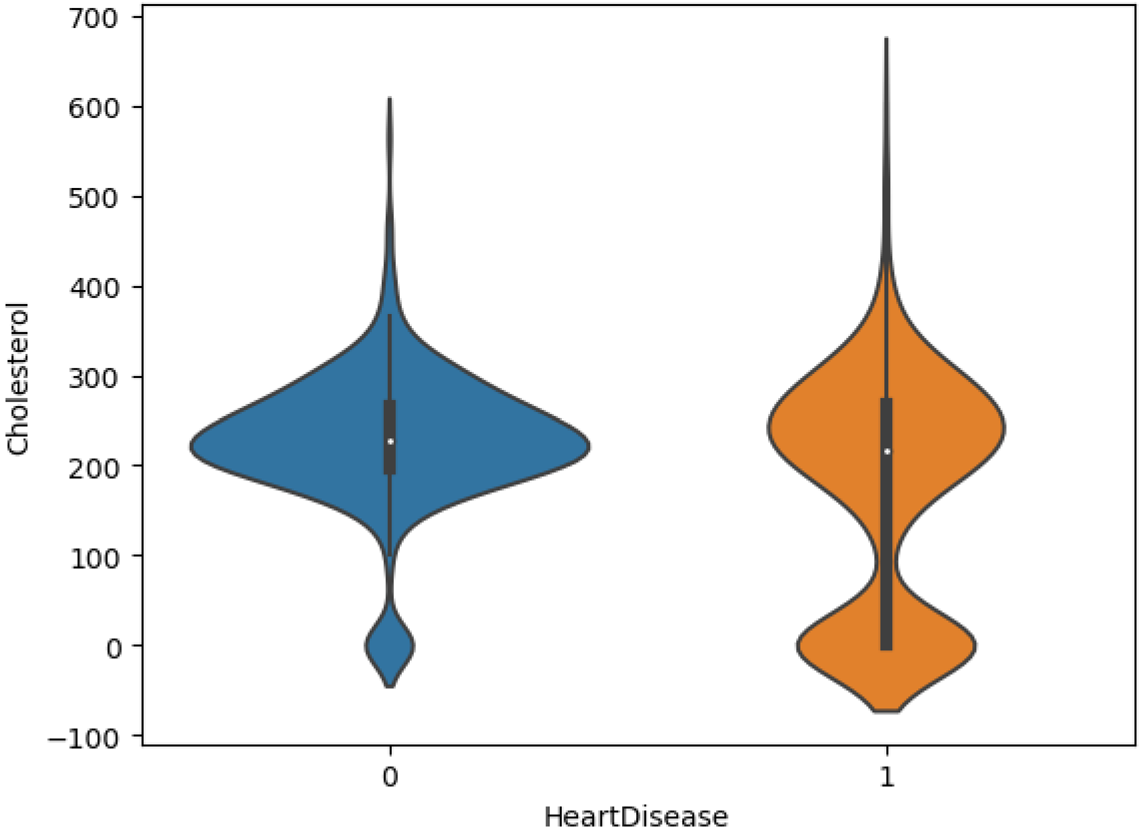
```
In [27]: sns.violinplot(y= df['Sex'], x= df['HeartDisease'])
```

```
Out[27]: <Axes: xlabel='HeartDisease', ylabel='Sex'>
```



In [28]:
```
sns.violinplot(y= df['Cholesterol'], x= df['HeartDisease'])
```

Out[28]:
```
<Axes: xlabel='HeartDisease', ylabel='Cholesterol'>
```



Correlation

In [29]:

```
df.corr()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17612\1134722465.py:1: FutureWarning: The default value of numeric_only in DataFrame.
corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_onl
y to silence this warning.
  df.corr()
```

Out[29]:

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
Age	1.000000	0.254399	-0.095282	0.198039	-0.382045	0.258612	0.282039
RestingBP	0.254399	1.000000	0.100893	0.070193	-0.112135	0.164803	0.107589
Cholesterol	-0.095282	0.100893	1.000000	-0.260974	0.235792	0.050148	-0.232741
FastingBS	0.198039	0.070193	-0.260974	1.000000	-0.131438	0.052698	0.267291
MaxHR	-0.382045	-0.112135	0.235792	-0.131438	1.000000	-0.160691	-0.400421
Oldpeak	0.258612	0.164803	0.050148	0.052698	-0.160691	1.000000	0.403951
HeartDisease	0.282039	0.107589	-0.232741	0.267291	-0.400421	0.403951	1.000000

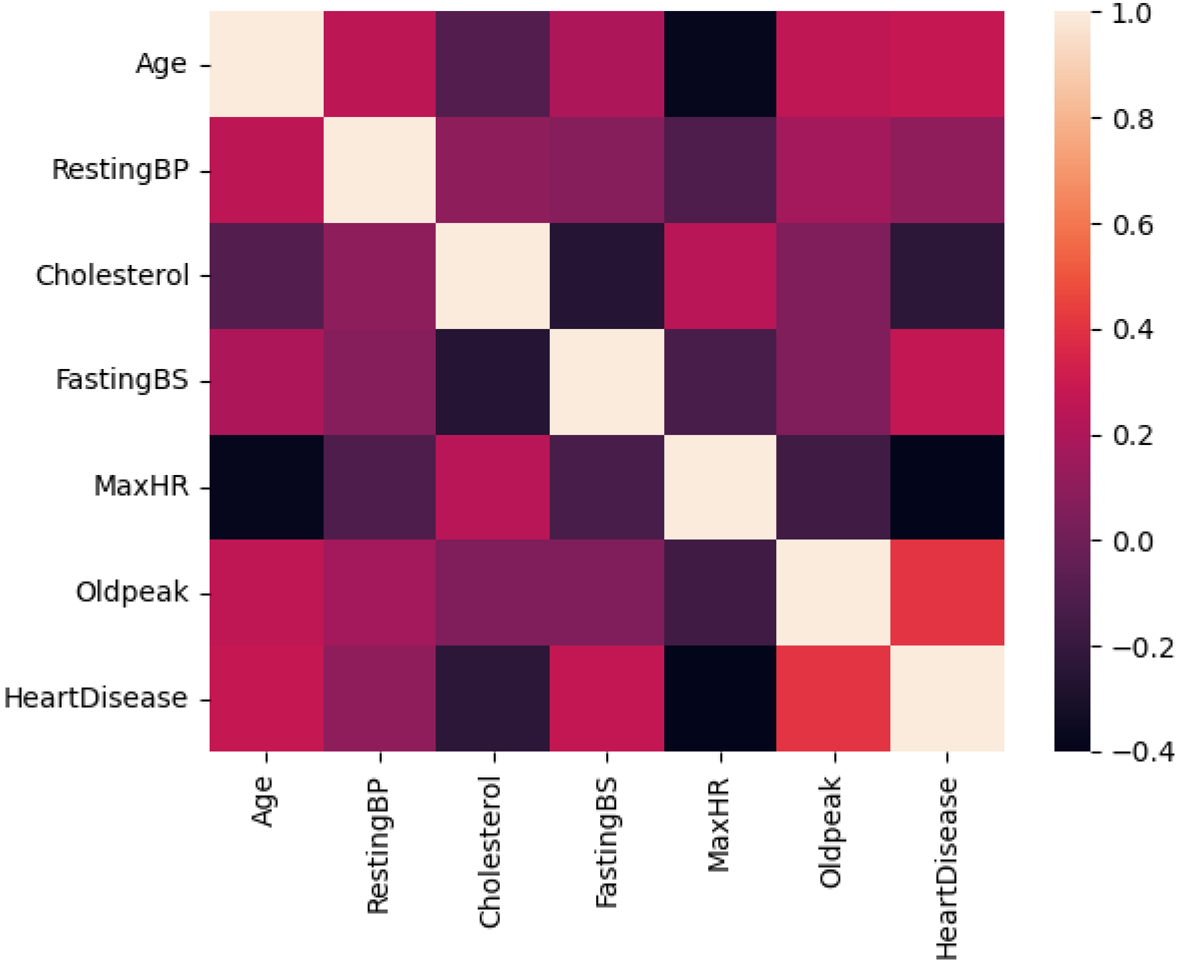
In [30]:

```
sns.heatmap(df.corr())
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_17612\58359773.py:1: FutureWarning: The default value of numeric_only in DataFrame.co
rr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(df.corr())
```

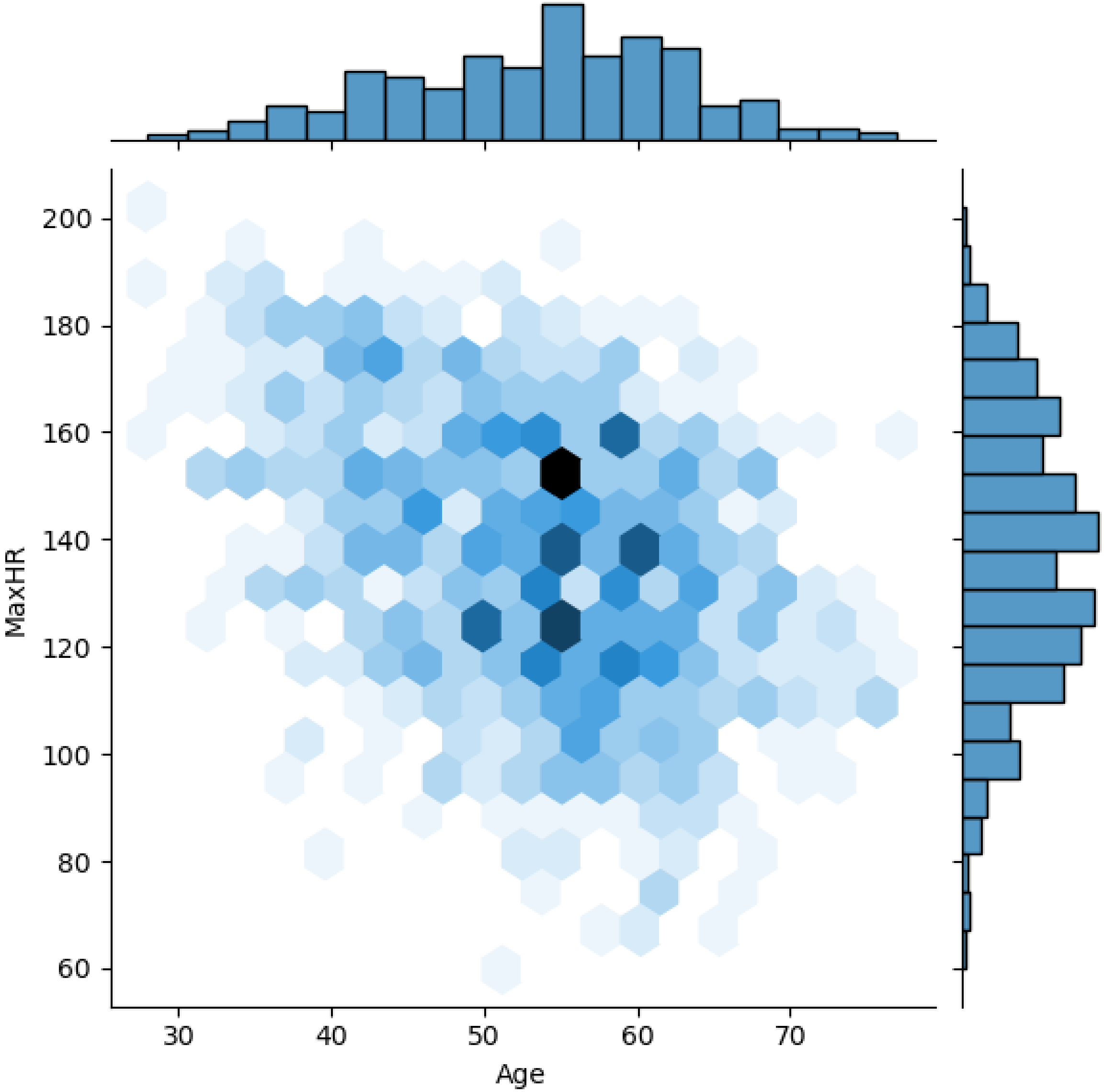
Out[30]:

```
<Axes: >
```

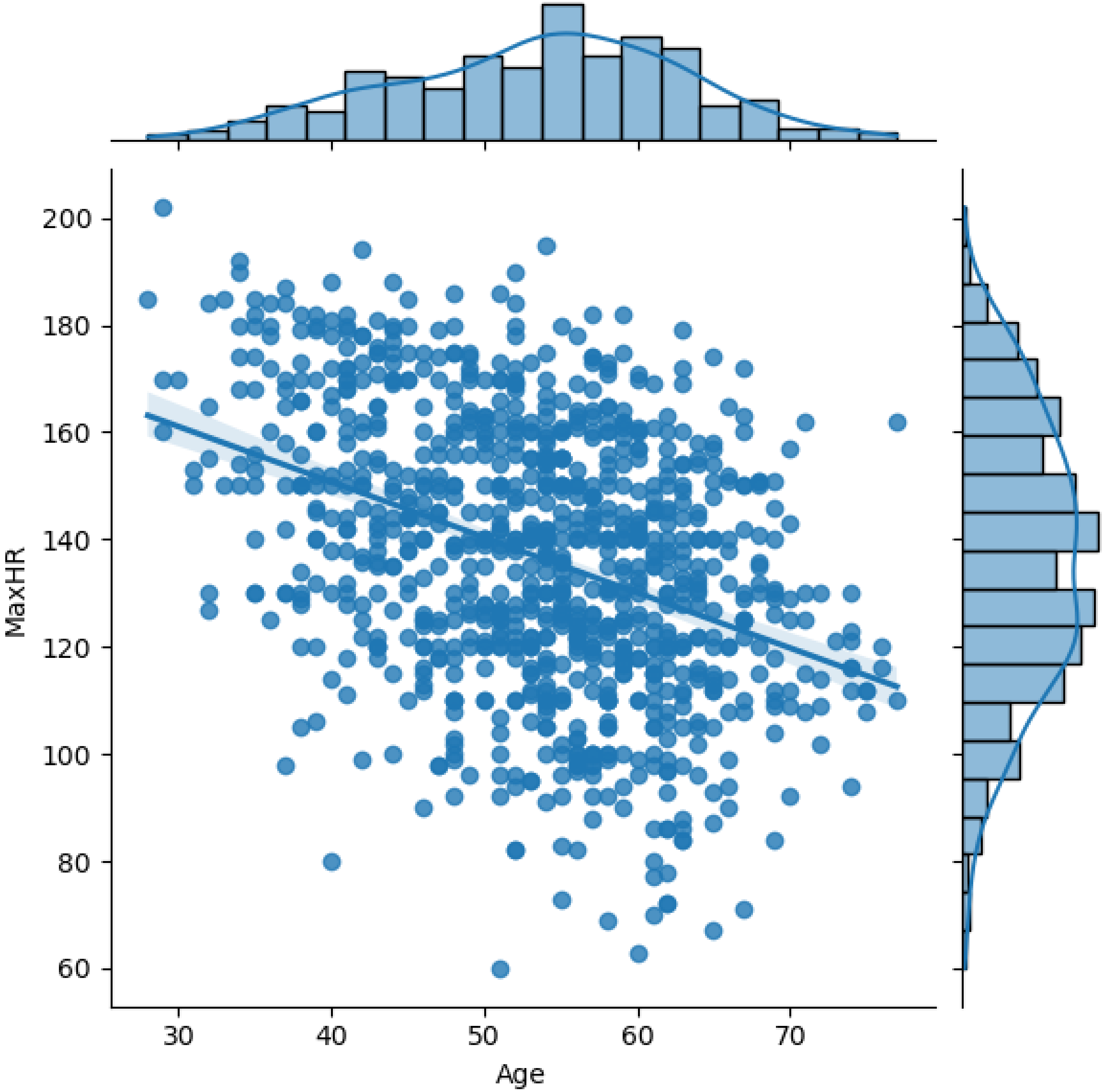


In [32]:
```
sns.jointplot(x='Age', y='MaxHR', data=df, kind='hex')
```

Out[32]:
```
<seaborn.axisgrid.JointGrid at 0x1f4904e6590>
```

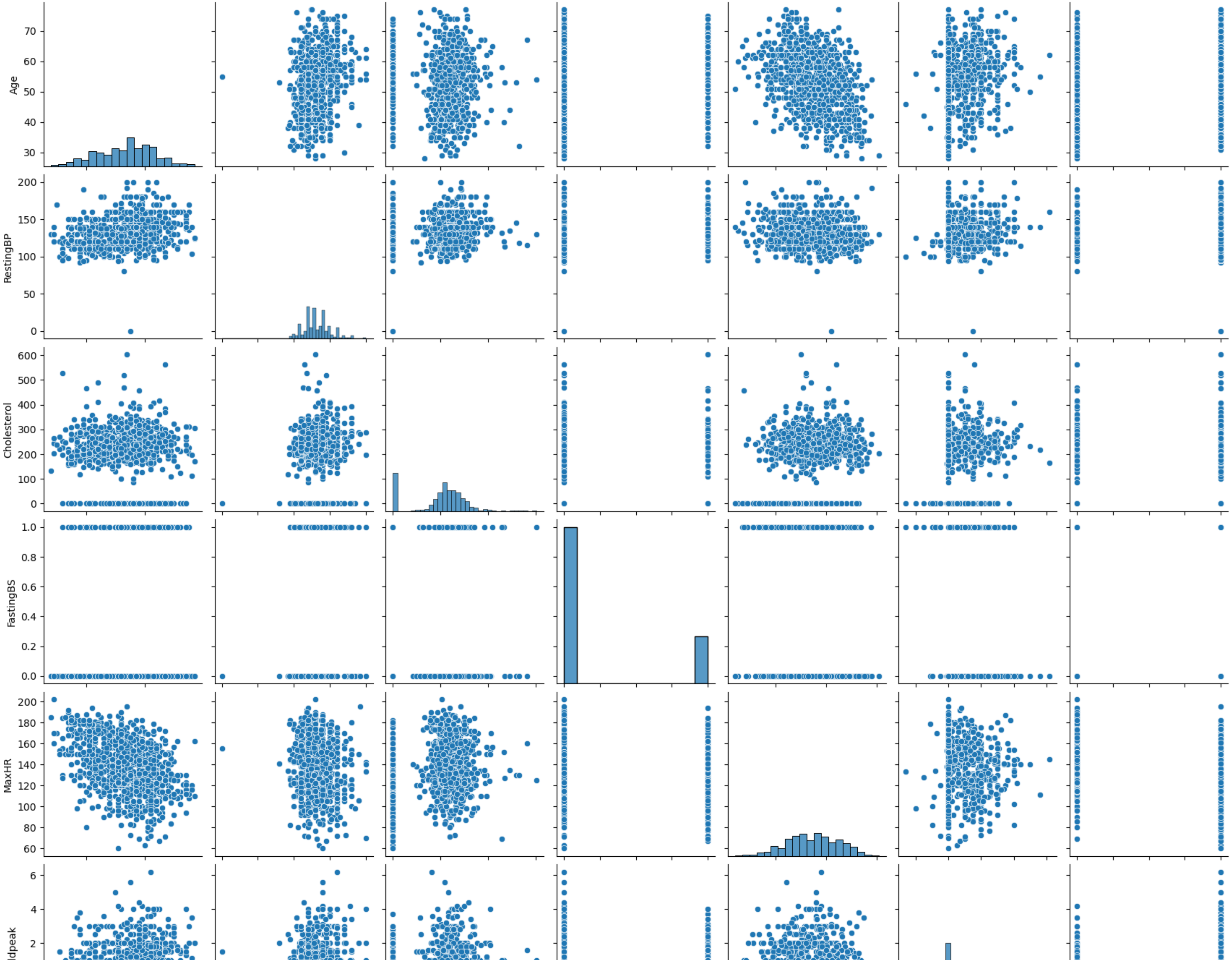


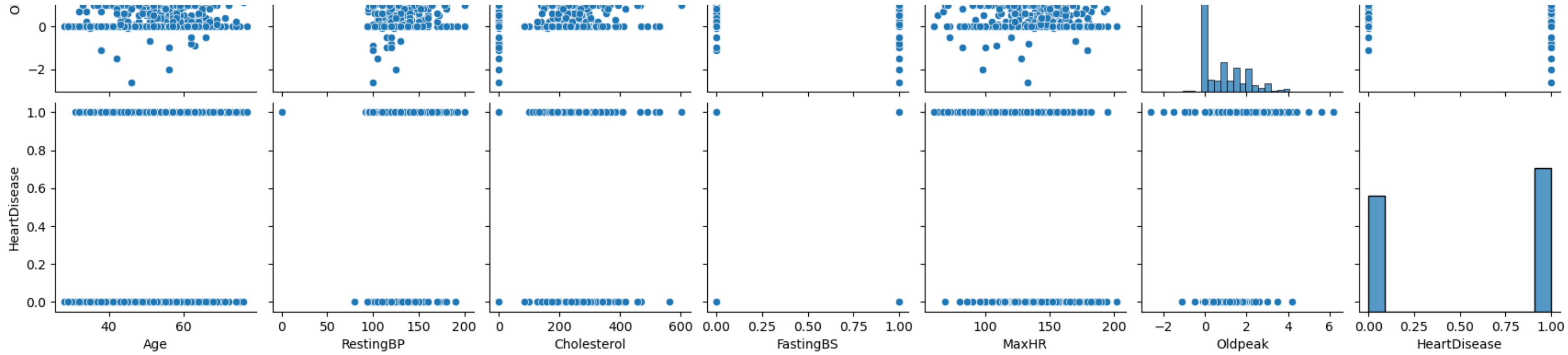
```
In [33]: sns.jointplot(x='Age', y='MaxHR', data=df, kind='reg')

Out[33]: <seaborn.axisgrid.JointGrid at 0x1f490ba5410>
```



In [37]:
```
sns.pairplot(df)
```

Out[37]:
```
<seaborn.axisgrid.PairGrid at 0x1f4942976d0>
```


Age
70
60
50
40
30
RestingBP
200
150
100
50
0
Cholesterol
600
500
400
300
200
100
0
FastingBS
1.0
0.8
0.6
0.4
0.2
0.0
MaxHR
200
180
160
140
120
100
80
60
ldpeak
6
4
2



Insights and conclusion

-In seaborn, a distplot is a function that plots a histogram, kernel density estimate (KDE) plot, and/or rug plot on a single figure. It is used to visualize the distribution of a single continuous variable.So after doing the visualization we can infer that minimum age is around 30 and maximum age is around 80. Maximum of the data falls within the range of 40 to 70.

-With the help of pie chart we can see that 79% of the people who are having heart disease are males and 21% are females in our dataset

-In seaborn, a violinplot is a graphical representation of a continuous distribution, showing the probability density of the data at different values. It is a combination of a box plot and a kernel density plot, with a rotated kernel density plot on each side.So from this we can see there are more males who are having the chances of having a heart disease compared to females.

-Heatmap-The values in the matrix are the Pearson correlation coefficients between the different variables, with 1 indicating a strong positive correlation, -1 indicating a strong negative correlation, and 0 indicating no correlation. The colors in the heatmap represent the strength of the correlation, with darker colors indicating a stronger correlation.

-Correlation pairplots are useful for visualizing the relationships between variables in a dataset and identifying potential correlations that may be of interest. They can also be used to identify variables that are highly correlated, which may be indicative of multicollinearity in a statistical model.

-Jointplots are useful for visualizing the relationship between two variables and the individual distributions of the variables. They can also be used to identify patterns in the data and to estimate the strength of the relationship between the variables.