

# Python-Libraries Based Project

-Harsh Chaudhary

## 1 Project - Heart Disease Exploratory Data Analysis

```
[2]: import pandas as pd
import numpy as np
!pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\harsh\anaconda3\lib\site-
packages (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in
c:\users\harsh\anaconda3\lib\site-packages (from seaborn) (1.24.3)
Requirement already satisfied: pandas>=0.25 in
c:\users\harsh\anaconda3\lib\site-packages (from seaborn) (2.0.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in
c:\users\harsh\anaconda3\lib\site-packages (from seaborn) (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\harsh\anaconda3\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (1.0.5)
Requirement already satisfied: cycler>=0.10 in
c:\users\harsh\anaconda3\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\harsh\anaconda3\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\harsh\anaconda3\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
c:\users\harsh\anaconda3\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in
c:\users\harsh\anaconda3\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (9.4.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in
c:\users\harsh\anaconda3\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\harsh\anaconda3\lib\site-packages (from
matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
```

Requirement already satisfied: pytz>=2020.1 in  
 c:\users\harsh\anaconda3\lib\site-packages (from pandas>=0.25->seaborn)  
 (2023.3.post1)  
 Requirement already satisfied: tzdata>=2022.1 in  
 c:\users\harsh\anaconda3\lib\site-packages (from pandas>=0.25->seaborn) (2023.3)  
 Requirement already satisfied: six>=1.5 in c:\users\harsh\anaconda3\lib\site-  
 packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)

```
[3]: df=pd.read_csv('heart.csv')
df.head()
```

```
[3]:   Age  Sex ChestPainType  RestingBP  Cholesterol  FastingBS  RestingECG  MaxHR  \
0   40   M             ATA         140           289           0       Normal    172
1   49   F             NAP         160           180           0       Normal    156
2   37   M             ATA         130           283           0           ST     98
3   48   F             ASY         138           214           0       Normal    108
4   54   M             NAP         150           195           0       Normal    122

   ExerciseAngina  Oldpeak  ST_Slope  HeartDisease
0                N       0.0        Up             0
1                N       1.0        Flat            1
2                N       0.0        Up             0
3                Y       1.5        Flat            1
4                N       0.0        Up             0
```

```
[4]: # Number of rows and columns in the dataset
df.shape
```

```
[4]: (918, 12)
```

### 1.0.1 Importing seaborn library - Data Visualization Liabrary

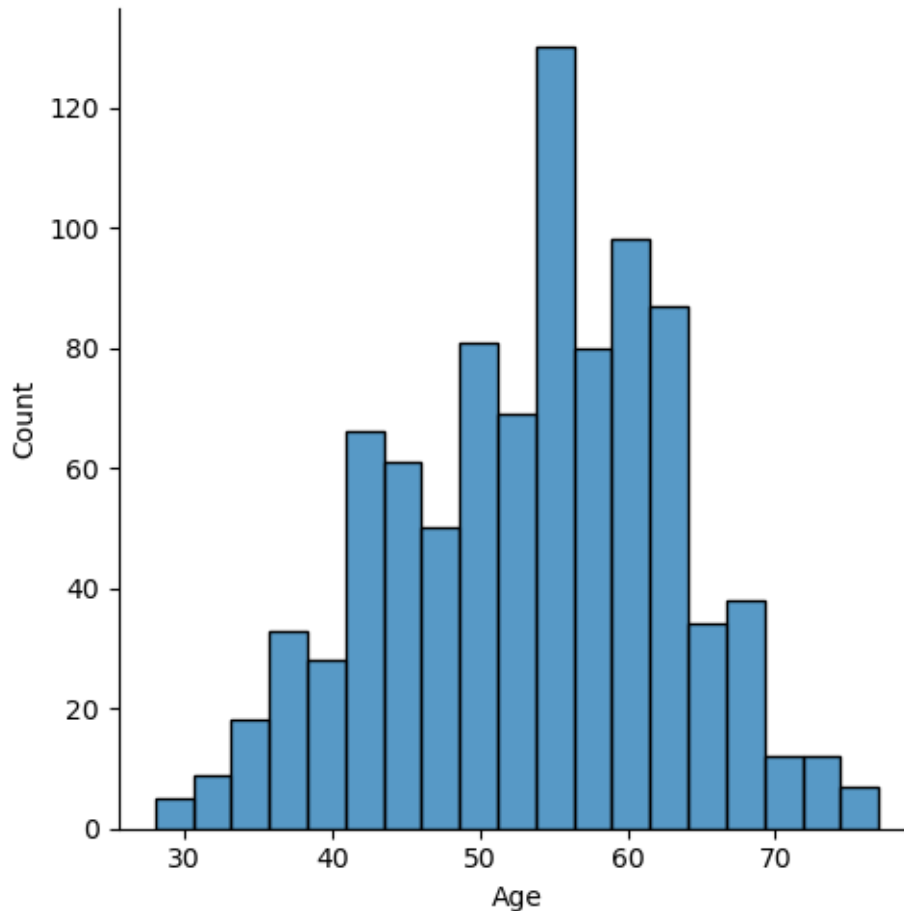
```
[5]: import seaborn as sb
```

```
[6]: #Plotting a graph for Age Distribution

sb.displot(df['Age'])
```

C:\Users\harsh\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:  
 The figure layout has changed to tight  
 self.\_figure.tight\_layout(\*args, \*\*kwargs)

```
[6]: <seaborn.axisgrid.FacetGrid at 0x2157aa4fa90>
```



## 2 Distribution Plot for Numeric Columns

```
[7]: # Distribution of age & plotting multiple graph types at same time
sb.distplot(df['Age'], kde=True, hist=True, rug=True, color='Blue')
```

C:\Users\harsh\AppData\Local\Temp\ipykernel\_17704\2342594979.py:2: UserWarning:

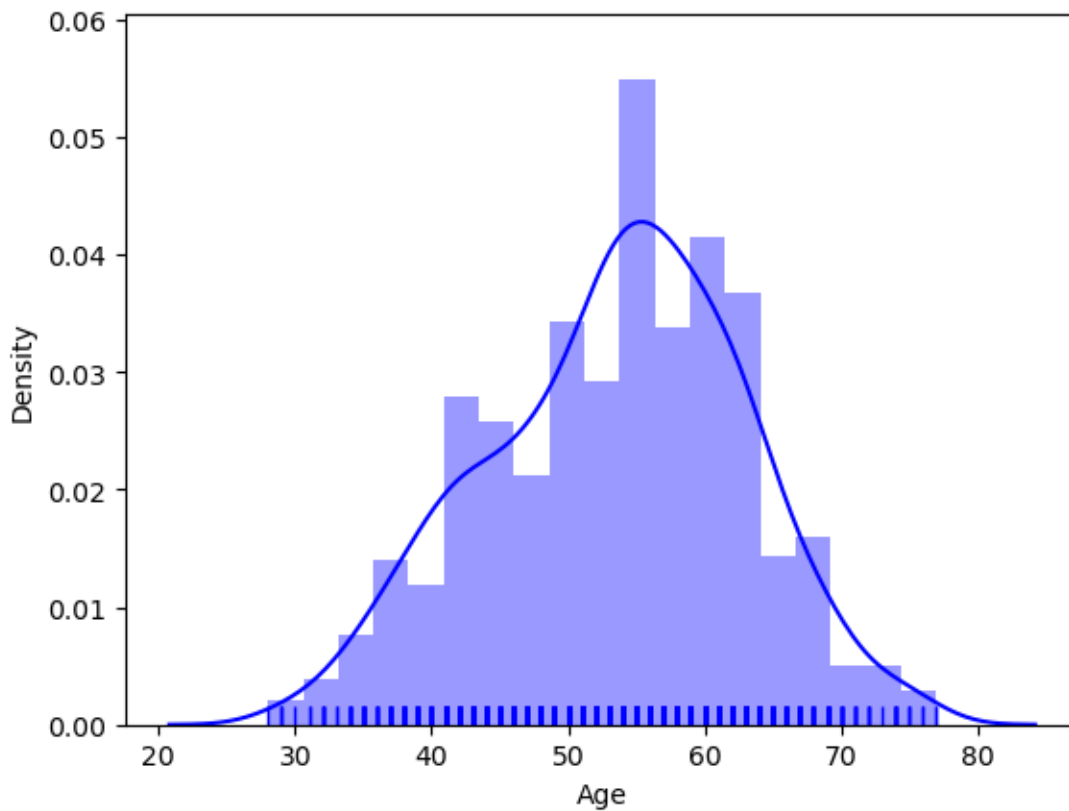
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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>

```
sb.distplot(df['Age'], kde=True, hist=True, rug=True, color='Blue')
```

```
[7]: <Axes: xlabel='Age', ylabel='Density'>
```



```
[8]: # Divide the dataset into 15 bins  
sb.distplot(df['Age'], bins=15)
```

C:\Users\harsh\AppData\Local\Temp\ipykernel\_17704\2536180916.py:2: UserWarning:

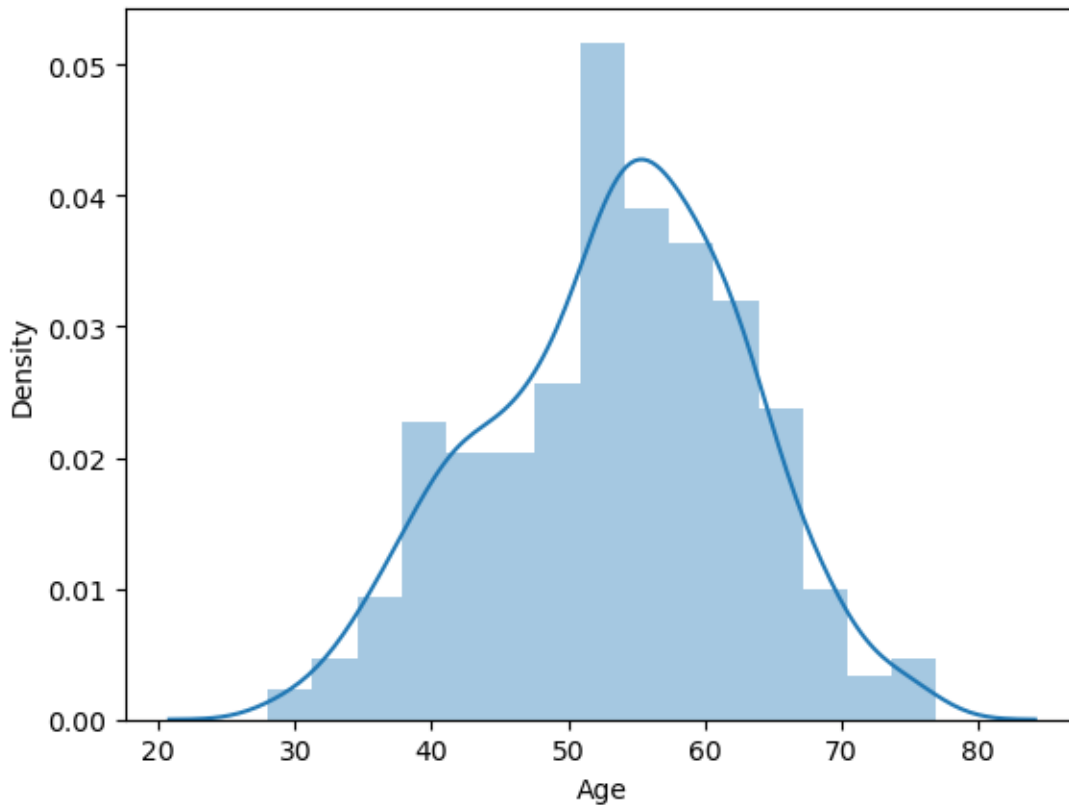
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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>

```
sb.distplot(df['Age'], bins=15)
```

```
[8]: <Axes: xlabel='Age', ylabel='Density'>
```



```
[9]: # Distribution of RestingBP
sb.distplot(df['RestingBP'], kde=True, color='Red')
```

C:\Users\harsh\AppData\Local\Temp\ipykernel\_17704\3275766499.py:2: UserWarning:

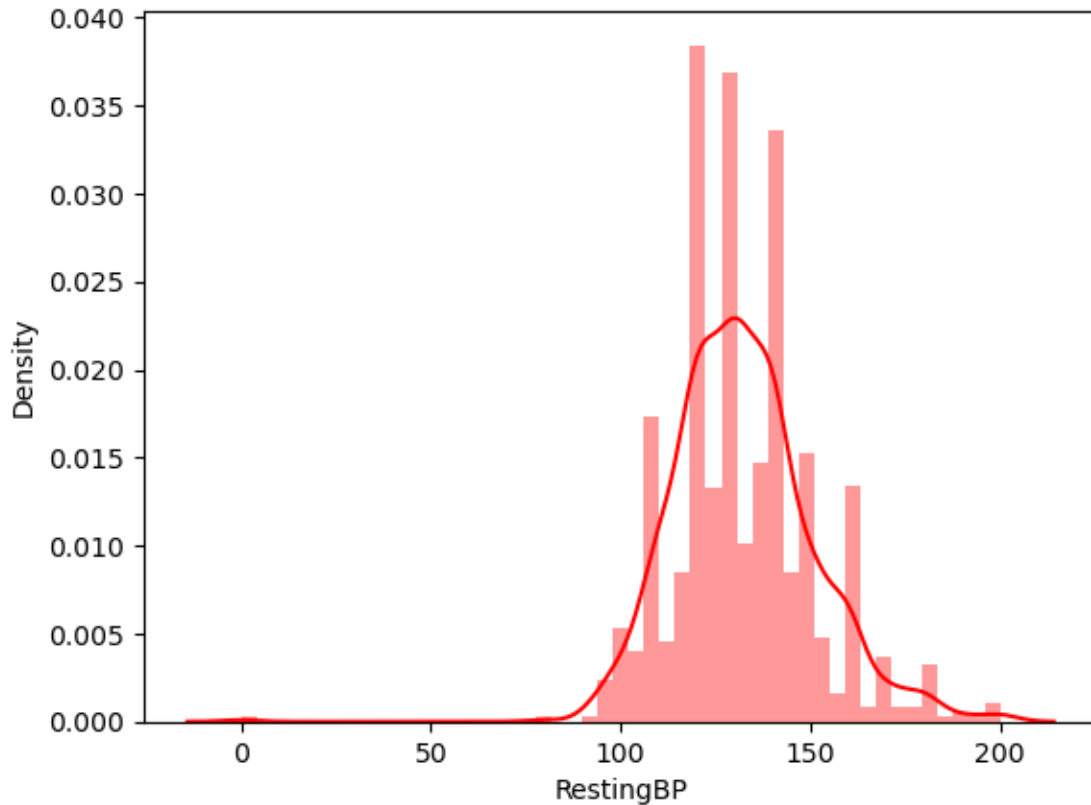
``distplot`` is a deprecated function and will be removed in seaborn v0.14.0.

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>

```
sb.distplot(df['RestingBP'], kde=True, color='Red')
```

```
[9]: <Axes: xlabel='RestingBP', ylabel='Density'>
```



```
[10]: # Distribution of Cholesterol
sb.distplot(df['Cholesterol'], kde=True, color='Green')
```

C:\Users\harsh\AppData\Local\Temp\ipykernel\_17704\2560055095.py:2: UserWarning:

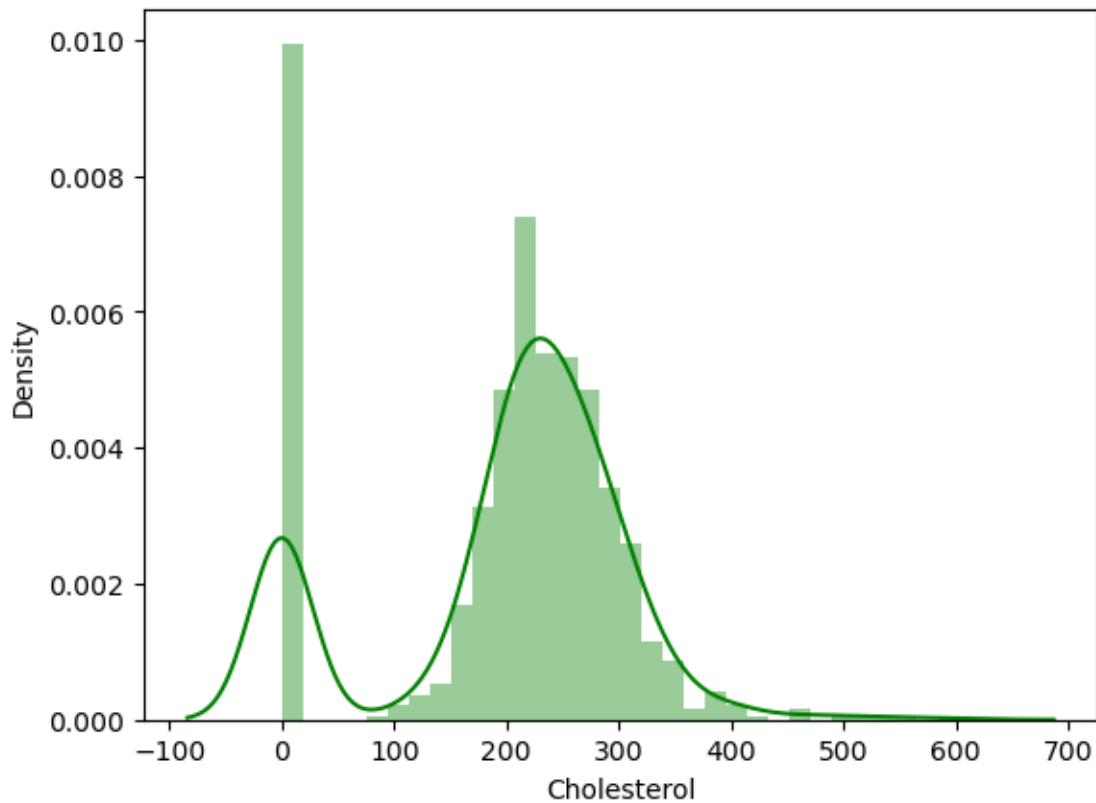
``distplot`` is a deprecated function and will be removed in seaborn v0.14.0.

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>

```
sb.distplot(df['Cholesterol'], kde=True, color='Green')
```

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



```
[11]: # Distribtuion of MaxHR
sb.distplot(df['MaxHR'], kde=True, color='purple')
```

C:\Users\harsh\AppData\Local\Temp\ipykernel\_17704\3372172032.py:2: UserWarning:

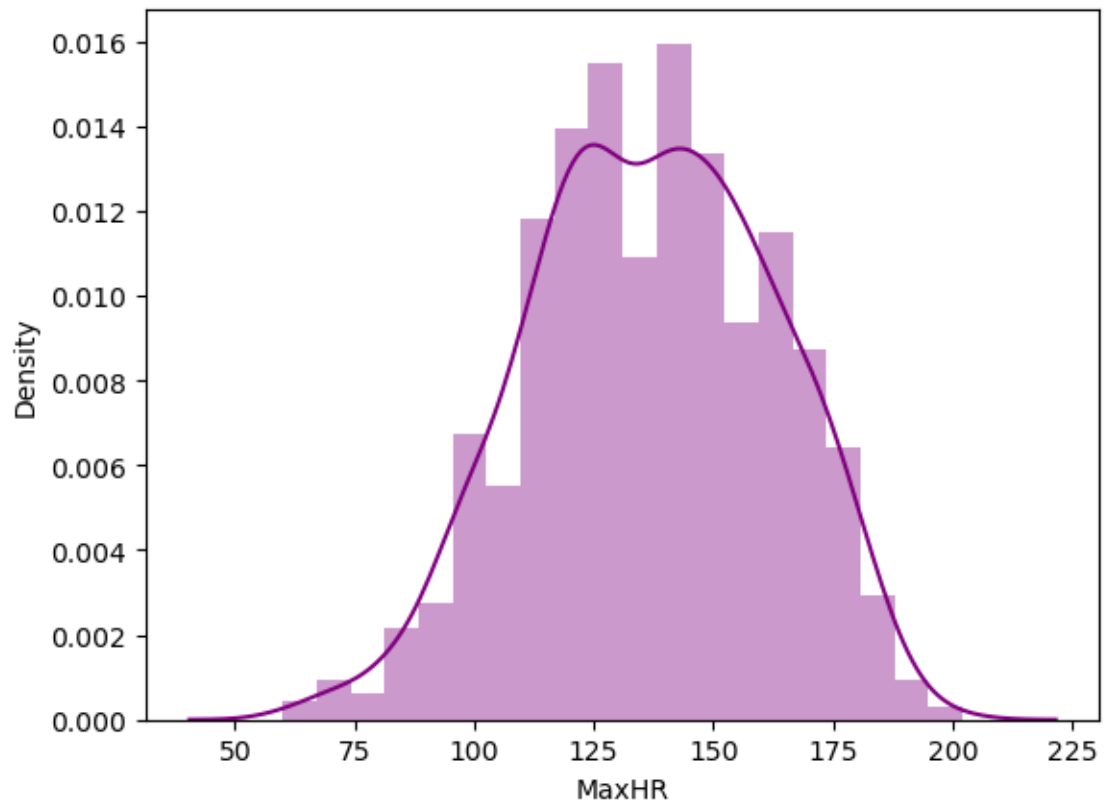
``distplot`` is a deprecated function and will be removed in seaborn v0.14.0.

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>

```
sb.distplot(df['MaxHR'], kde=True, color='purple')
```

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



### 3 Pie Chart's for Categorical columns

```
[12]: # Grouping of Gender's
df.groupby('Sex').size()
```

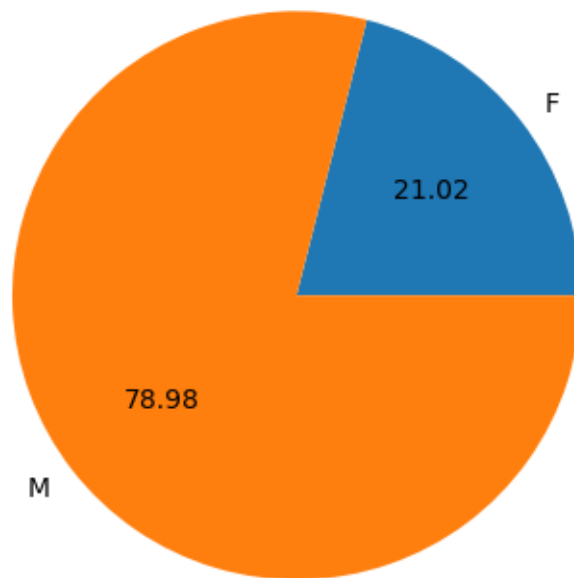
```
[12]: Sex
F      193
M      725
dtype: int64
```

```
[13]: # Distribution of Gender - Pie Chart
# autopct : auto punctuations

df.groupby('Sex').size().plot( kind = 'pie', autopct='%.2f')
```

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



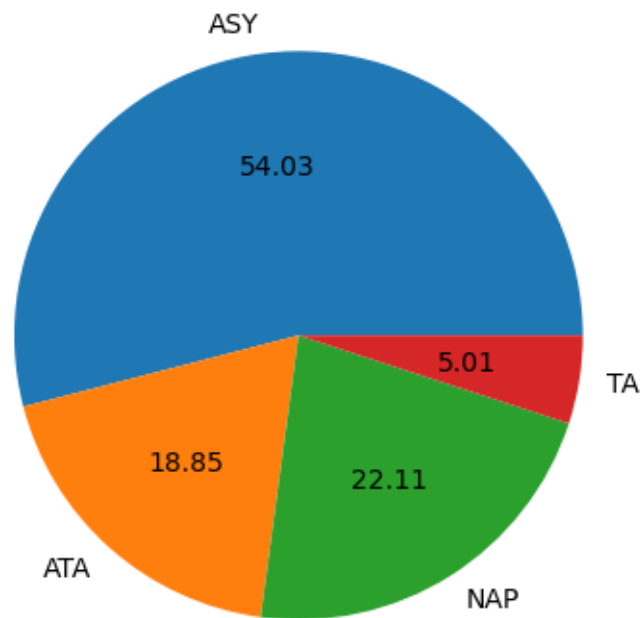


```
[14]: # Grouping of ChestPainType  
df.groupby('ChestPainType').size()
```

```
[14]: ChestPainType  
ASY    496  
ATA    173  
NAP    203  
TA      46  
dtype: int64
```

```
[15]: # Distribution of ChestPainType - Pie Chart  
df.groupby('ChestPainType').size().plot(kind = 'pie', autopct='%.2f')
```

```
[15]: <Axes: >
```

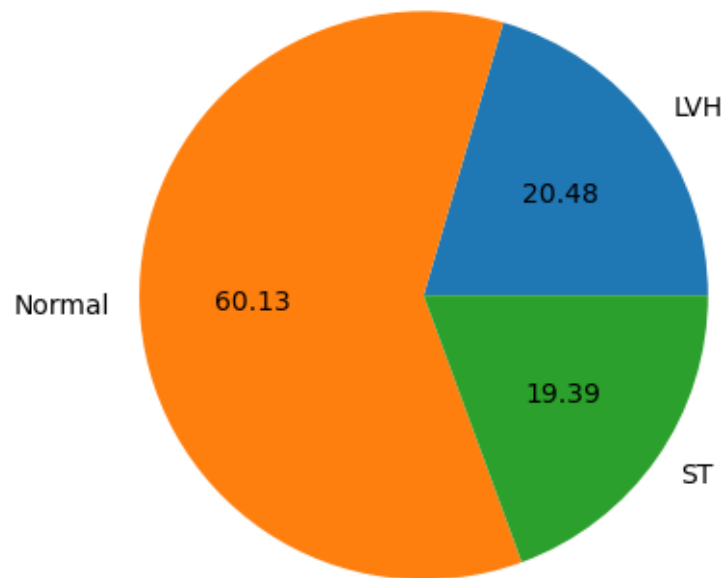


```
[16]: # Grouping of RestingECG
df.groupby('RestingECG').size()
```

```
[16]: RestingECG
      LVH      188
      Normal  552
      ST      178
      dtype: int64
```

```
[17]: # Distribution of RestingECG
df.groupby('RestingECG').size().plot( kind = 'pie', autopct='%.2f')
```

```
[17]: <Axes: >
```

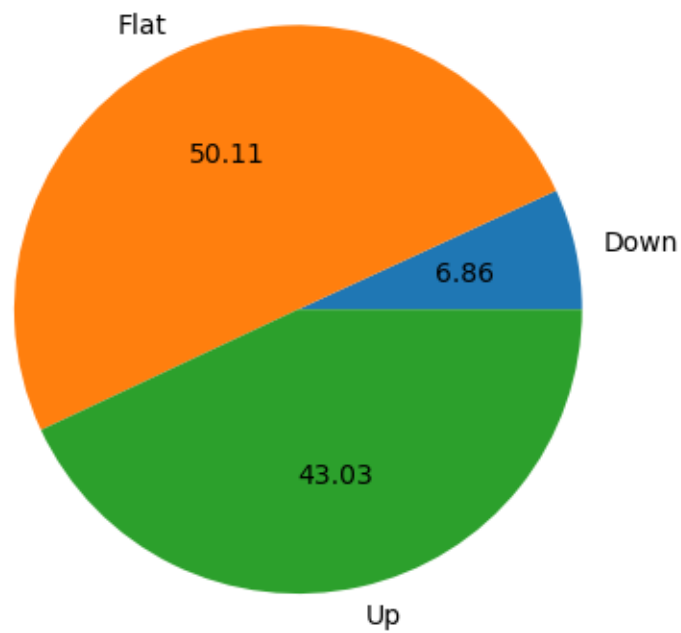


```
[18]: # Grouping of ST_Slope  
df.groupby('ST_Slope').size()
```

```
[18]: ST_Slope  
Down      63  
Flat     460  
Up       395  
dtype: int64
```

```
[19]: # Distribution of ST_Slope  
df.groupby('ST_Slope').size().plot( kind = 'pie', autopct='%.2f')
```

```
[19]: <Axes: >
```

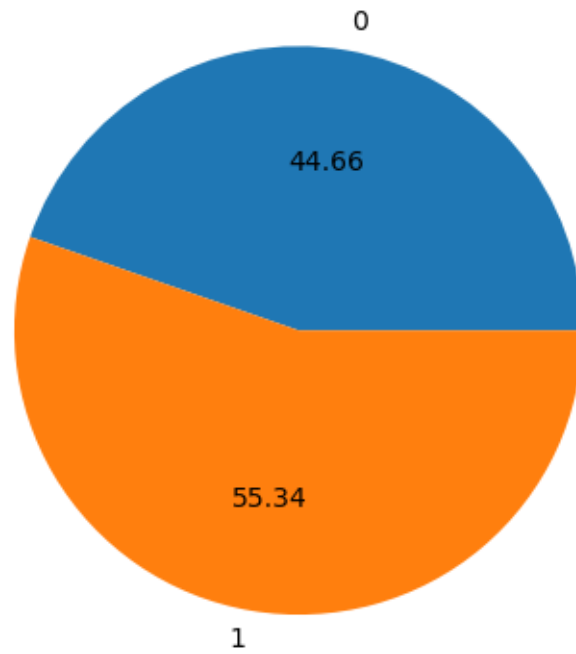


```
[20]: # Grouping of HeartDisease  
df.groupby('HeartDisease').size()
```

```
[20]: HeartDisease  
0    410  
1    508  
dtype: int64
```

```
[21]: # Distribution of HeartDisease  
df.groupby('HeartDisease').size().plot( kind = 'pie', autopct='% .2f')
```

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

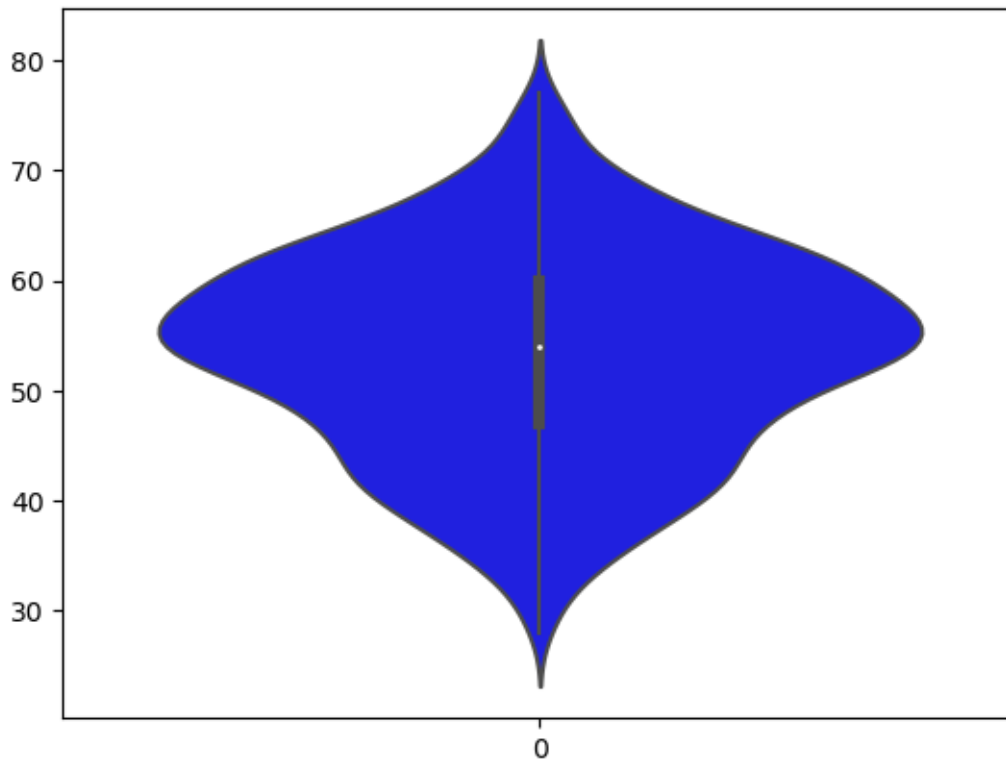


## 4 ViolinPlot

For Multiple Column Analysis

```
[22]: # region toward's 0 denotes Lower amounts  
      sb.violinplot(df['Age'], color='blue')
```

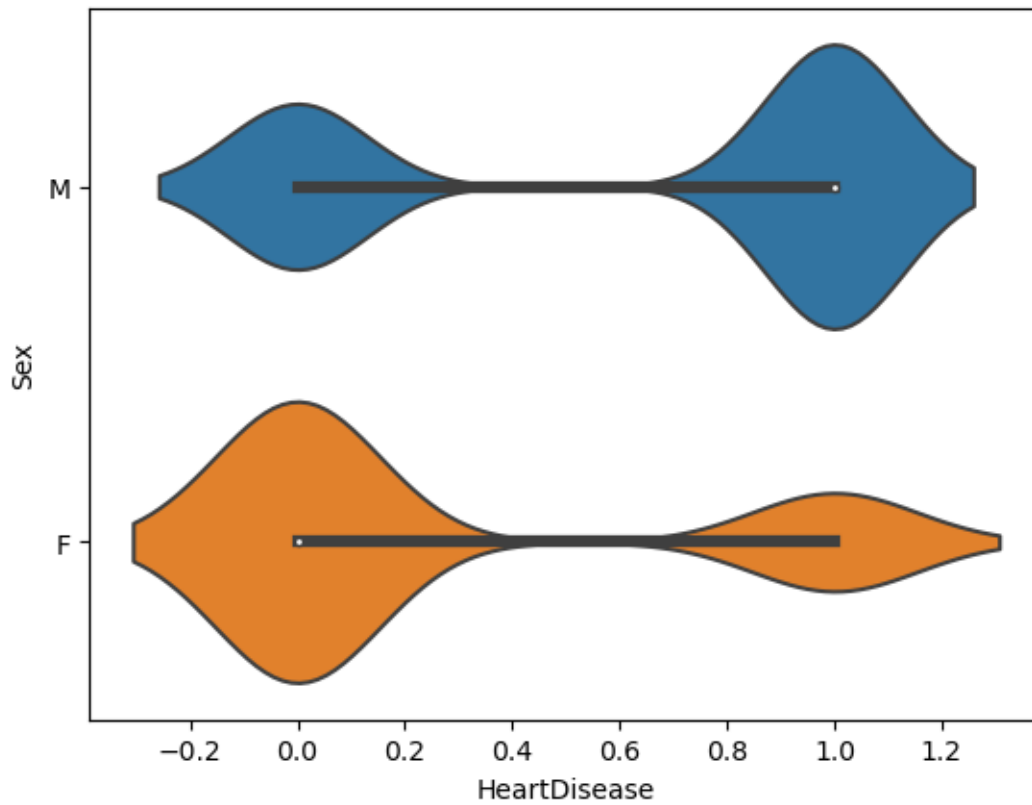
```
[22]: <Axes: >
```



#### 4.1 HeartDisease correlation with Sex

```
[23]: sb.violinplot(y=df['Sex'], x=df['HeartDisease'])
```

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

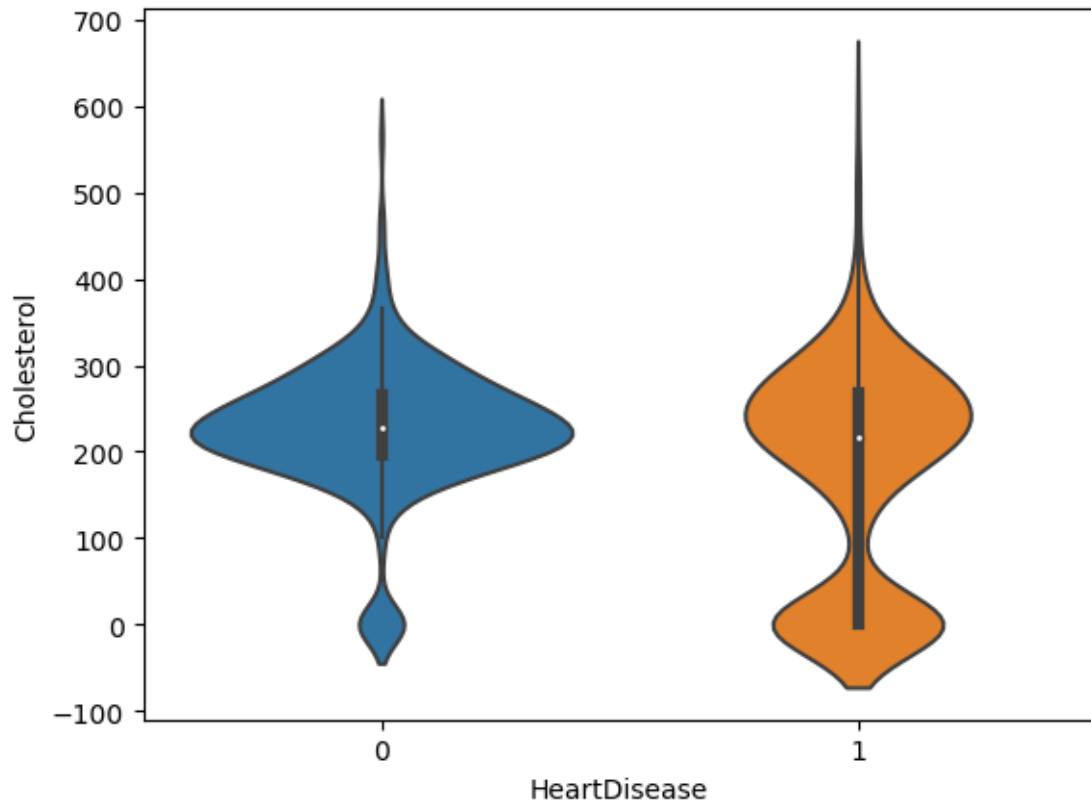


Denoting More Males have HearDisease

## 4.2 HeartDisease correlation with Cholesterol

```
[24]: sb.violinplot(y=df['Cholesterol'], x=df['HeartDisease'])
```

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



## 5 Correlation of Heatmap

Correlation ranges btw 1 to -1 more closer to 1 better correlation between 2 columns Closer to -1 denotes negative correlation that means if one increases other decreases It defines the dependency btw 2 columns, like if one values fluctuates then how other value is affected

```
[27]: numeric_df = df.select_dtypes(include=['number'])
      numeric_df.corr()
```

```
[27]:
```

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

```
HeartDisease
Age          0.282039
RestingBP    0.107589
```



```
Cholesterol    -0.232741
FastingBS      0.267291
MaxHR          -0.400421
Oldpeak        0.403951
HeartDisease    1.000000
```

```
[28]: # Let's select the columns with datatype as numbers
num_df=df.select_dtypes(include=['number'])
num_df
```

```
[28]:
```

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
0	40	140	289	0	172	0.0	0
1	49	160	180	0	156	1.0	1
2	37	130	283	0	98	0.0	0
3	48	138	214	0	108	1.5	1
4	54	150	195	0	122	0.0	0
..	...	...	...	...	...	...	...
913	45	110	264	0	132	1.2	1
914	68	144	193	1	141	3.4	1
915	57	130	131	0	115	1.2	1
916	57	130	236	0	174	0.0	1
917	38	138	175	0	173	0.0	0

[918 rows x 7 columns]

```
[29]: num_df.corr()
```

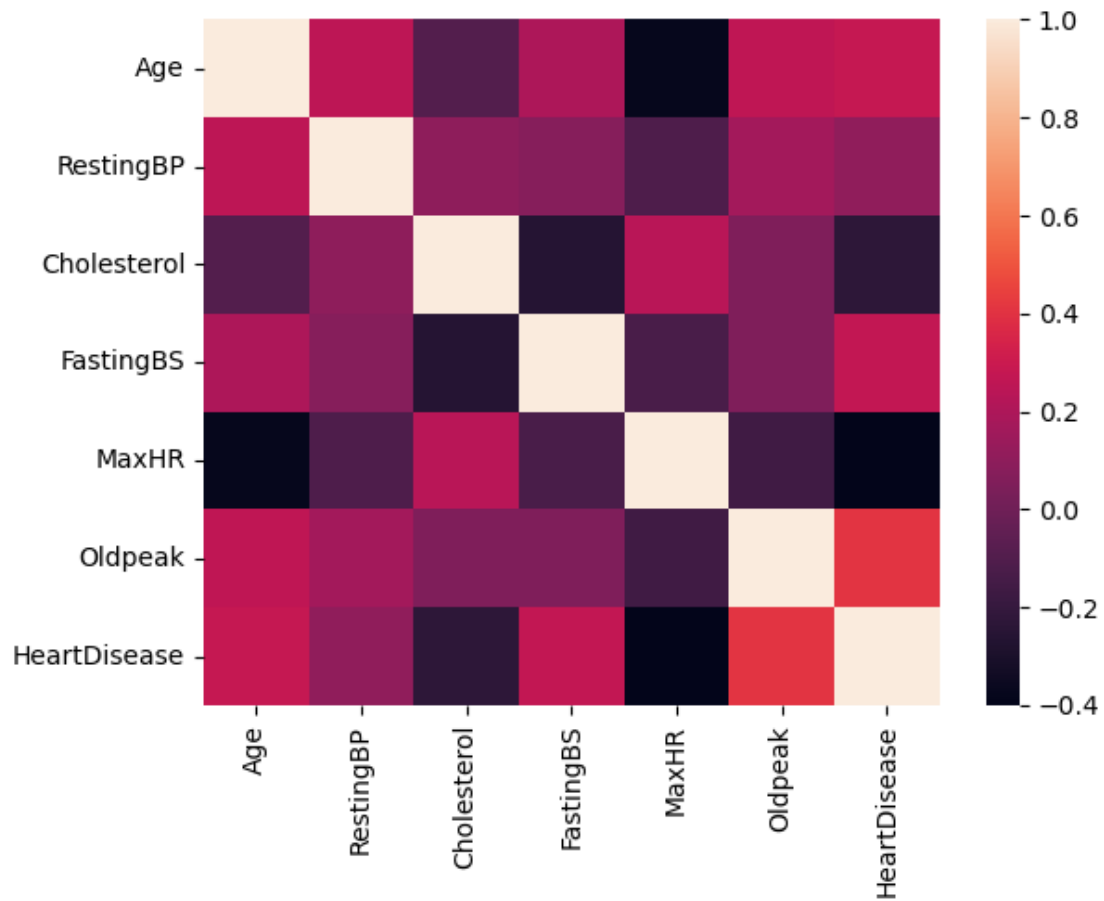
```
[29]:
```

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

```
HeartDisease
Age          0.282039
RestingBP    0.107589
Cholesterol  -0.232741
FastingBS    0.267291
MaxHR        -0.400421
Oldpeak      0.403951
HeartDisease 1.000000
```

```
[31]: sb.heatmap(num_df.corr())
```

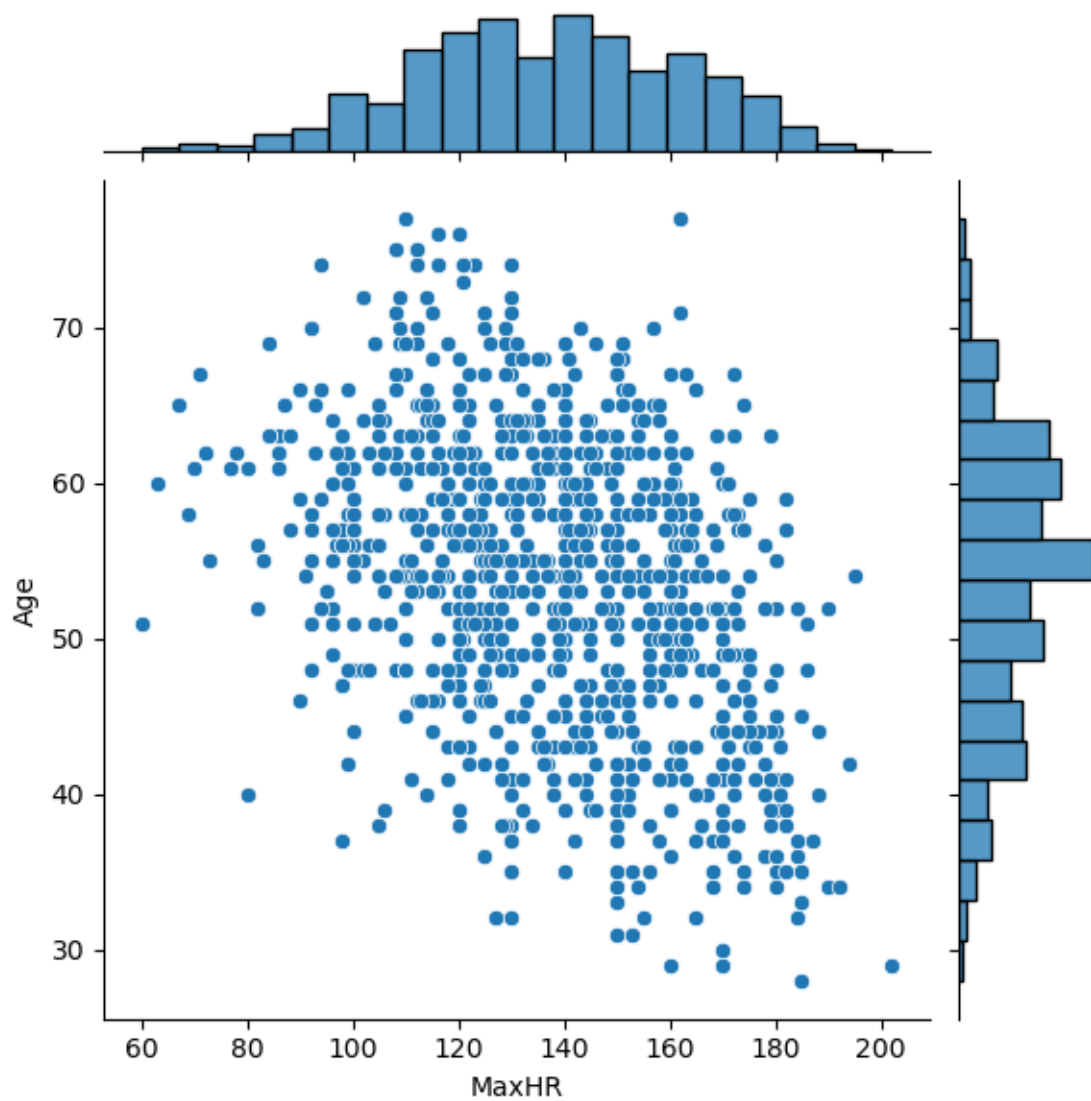
[31]: <Axes: >



## 6 Correlation - JointPlot

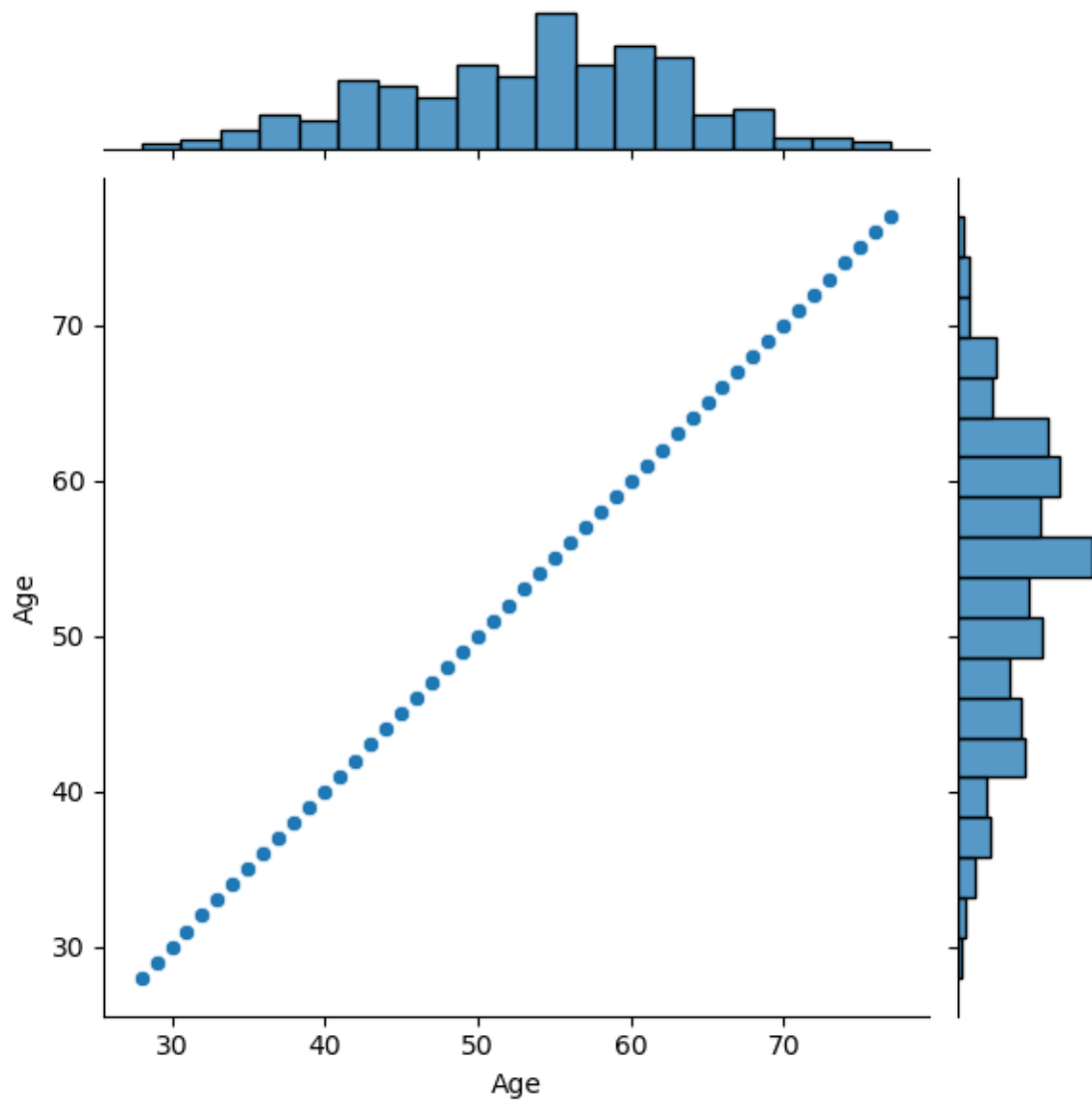
```
[33]: sb.jointplot(y='Age', x='MaxHR', data=df)
```

[33]: <seaborn.axisgrid.JointGrid at 0x2157fb708d0>



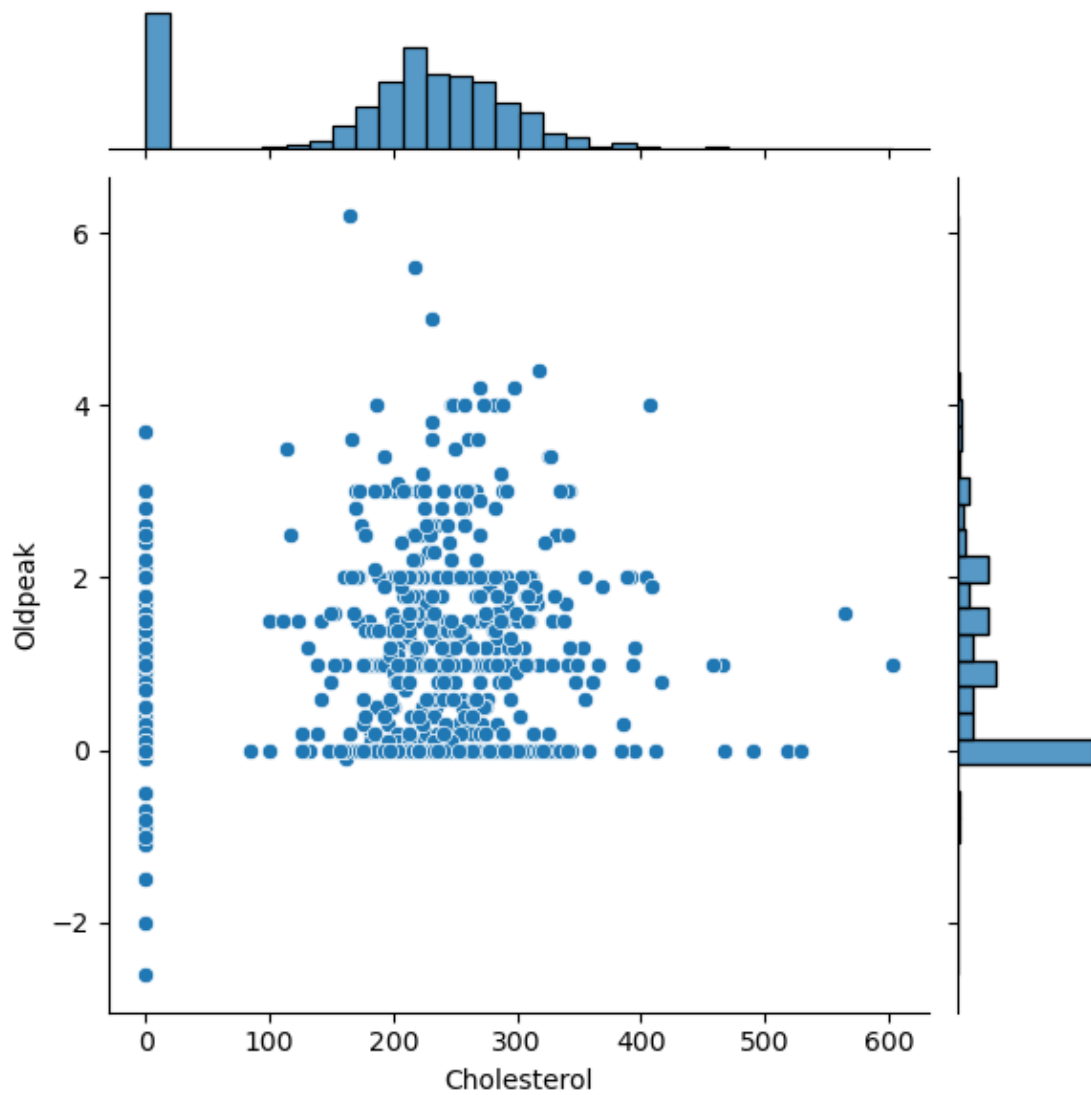
```
[35]: # 100% Correlation
      sb.jointplot(y='Age', x='Age', data=df)
```

```
[35]: <seaborn.axisgrid.JointGrid at 0x21503615850>
```



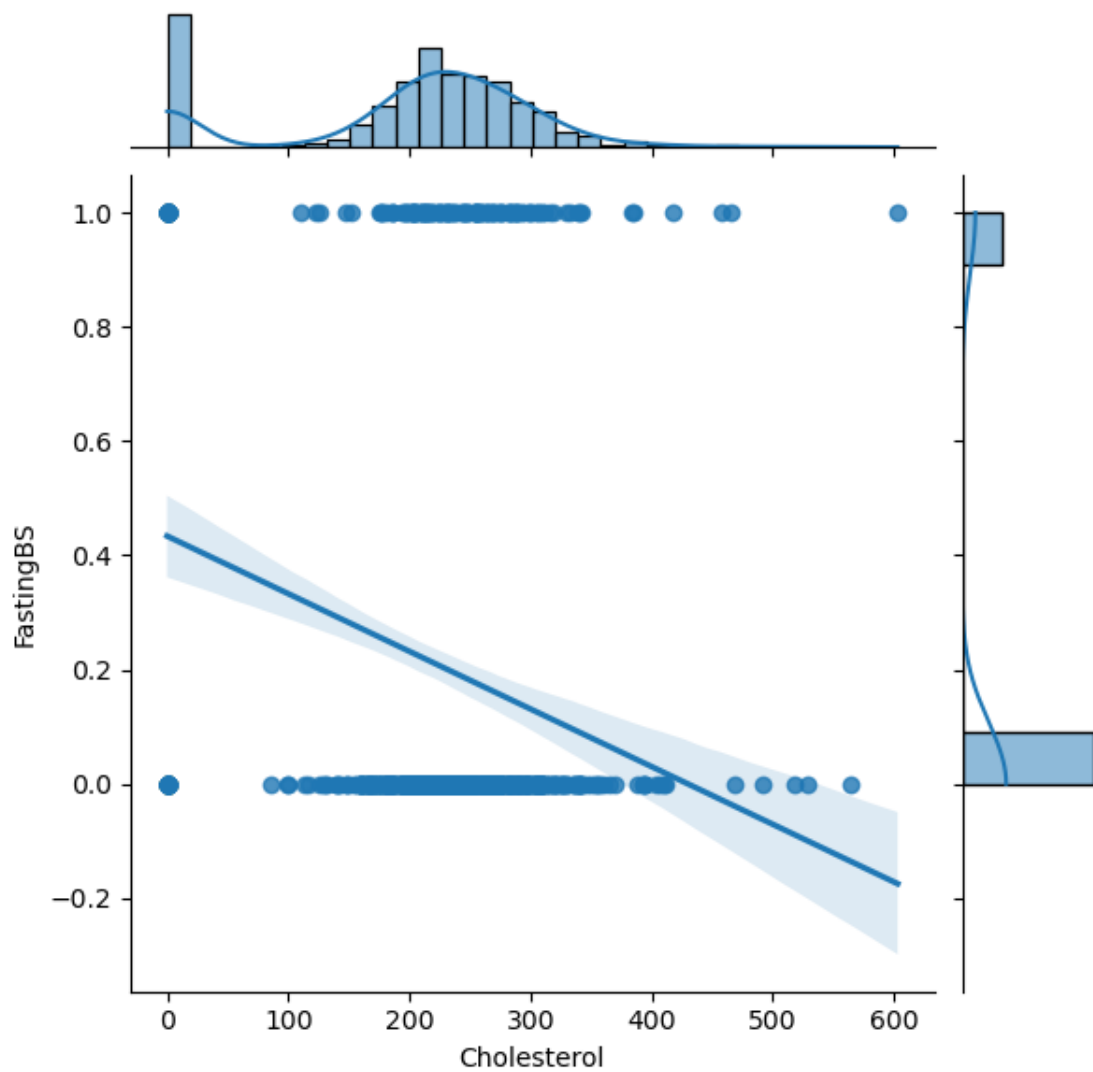
```
[36]: sb.jointplot(y='Oldpeak', x='Cholesterol', data=df)
```

```
[36]: <seaborn.axisgrid.JointGrid at 0x2150371ead0>
```



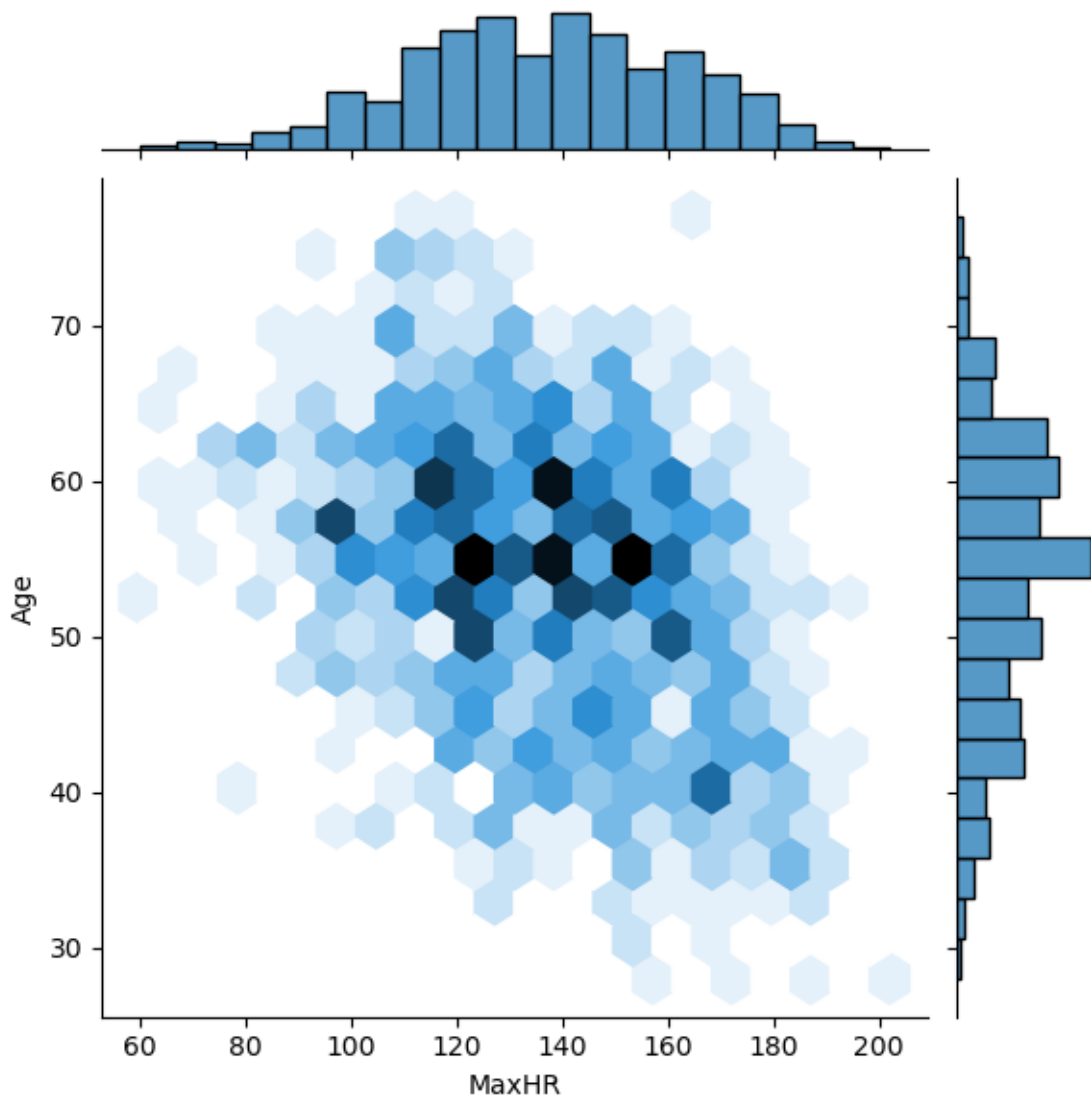
```
[38]: # Line doesn't fit the data : no correlation
      sb.jointplot(x='Cholesterol',y='FastingBS',kind='reg',data=df)
```

```
[38]: <seaborn.axisgrid.JointGrid at 0x215036d8a90>
```



```
[40]: sb.jointplot(y='Age', x='MaxHR', data=df, kind='hex')
```

```
[40]: <seaborn.axisgrid.JointGrid at 0x2150534c5d0>
```



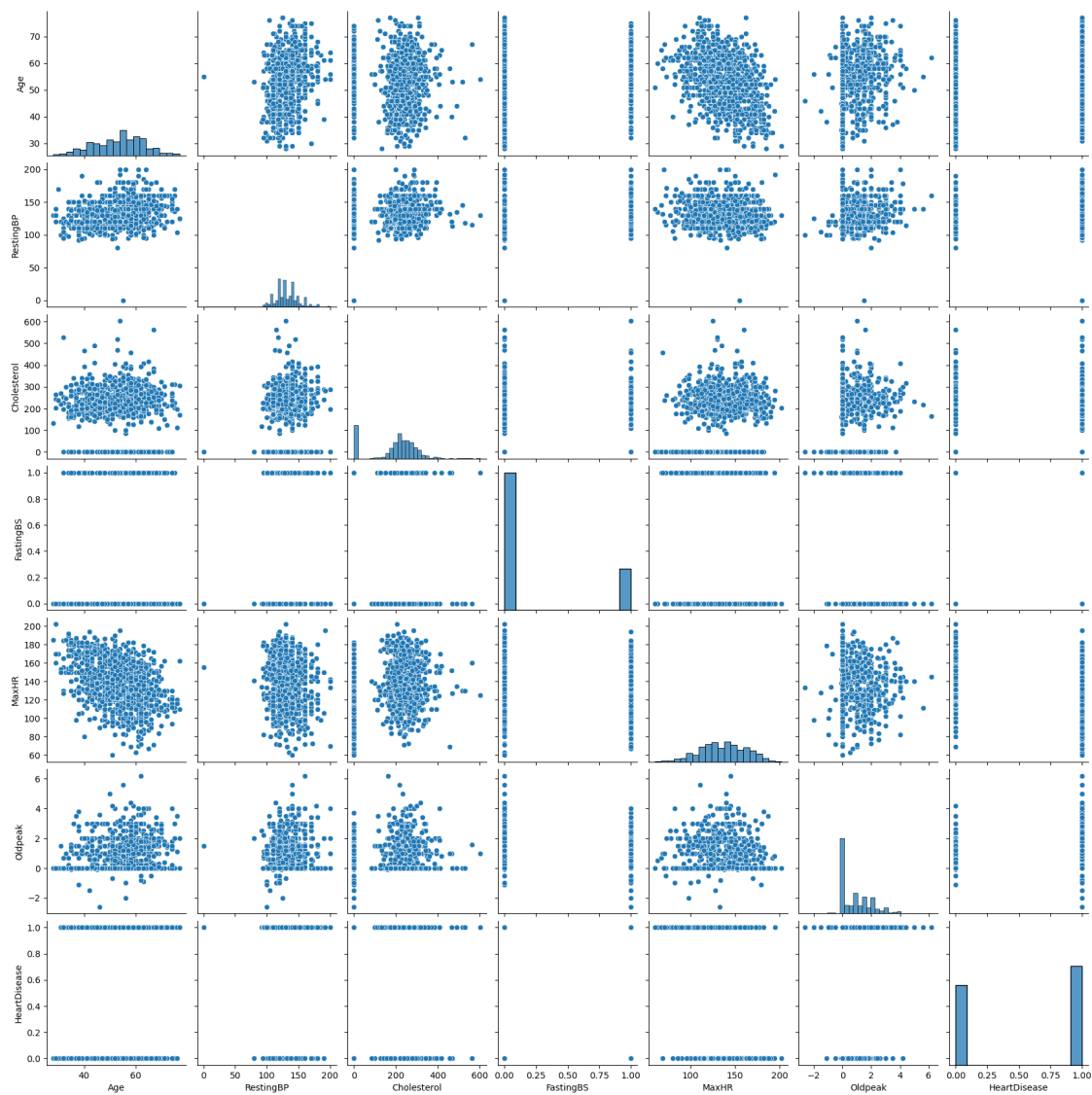
## 7 Correlation- PairPlot

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.

```
[41]: sb.pairplot(df)
```

```
C:\Users\harsh\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

[41]: <seaborn.axisgrid.PairGrid at 0x21503628150>



[ ]: