```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy import stats
import seaborn as sns
df = pd.read_csv('heart.csv')

print(df)
```

```
     age  sex  cp  trestbps  chol  fbs  ...  exang  oldpeak  slope  ca  thal  target
0     63    1   3       145   233    1  ...      0      2.3      0   0     1       1
1     37    1   2       130   250    0  ...      0      3.5      0   0     2       1
2     41    0   1       130   204    0  ...      0      1.4      2   0     2       1
3     56    1   1       120   236    0  ...      0      0.8      2   0     2       1
4     57    0   0       120   354    0  ...      1      0.6      2   0     2       1
..   ...  ...  ..       ...   ...  ...  ...    ...      ...    ...  ..   ...     ...
298   57    0   0       140   241    0  ...      1      0.2      1   0     3       0
299   45    1   3       110   264    0  ...      0      1.2      1   0     3       0
300   68    1   0       144   193    1  ...      0      3.4      1   2     3       0
301   57    1   0       130   131    0  ...      1      1.2      1   1     3       0
302   57    0   1       130   236    0  ...      0      0.0      1   1     2       0

[303 rows x 14 columns]
```

## Checking if for any NULL values

```
print("No of NULL values in the dataset ",len(df)-len(df.isna()))

```

```
No of NULL values in the dataset  0
```

## Removing random NULL values

```
for col in df.columns:
    df.loc[df.sample(frac=0.2).index, col] = np.nan
```

```
df
```

```
     age  sex   cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  slope  ca
0    NaN  NaN  3.0     145.0    NaN  1.0      0.0    150.0    0.0      2.3    0.0  Na
1   37.0  1.0  2.0     130.0  250.0  0.0      1.0    187.0    0.0      3.5    0.0  0.
2   41.0  0.0  1.0     130.0    NaN  NaN      0.0    172.0    0.0      1.4    2.0  0.
3   56.0  1.0  1.0     120.0    NaN  0.0      1.0      NaN    NaN      0.8    2.0  0.
4    NaN  NaN  0.0     120.0  354.0  0.0      NaN    163.0    1.0      0.6    2.0  0.
...  ...  ...  ...       ...    ...  ...      ...      ...    ...      ...    ...  .
```

Removing NULL values from the dataset

```
for col in df.columns:
  df[col].fillna(df[col].mode()[0], inplace=True)
```

```
df
```

```
      age  sex   cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  slope   ca
0    58.0  1.0  3.0     145.0  234.0  1.0      0.0    150.0    0.0      2.3    0.0  0.0
1    37.0  1.0  2.0     130.0  250.0  0.0      1.0    187.0    0.0      3.5    0.0  0.0
2    41.0  0.0  1.0     130.0  234.0  0.0      0.0    172.0    0.0      1.4    2.0  0.0
3    56.0  1.0  1.0     120.0  234.0  0.0      1.0    160.0    0.0      0.8    2.0  0.0
4    58.0  1.0  0.0     120.0  354.0  0.0      0.0    163.0    1.0      0.6    2.0  0.0
...   ...  ...  ...       ...    ...  ...      ...      ...    ...      ...    ...  ...
298  57.0  0.0  0.0     140.0  241.0  0.0      1.0    123.0    1.0      0.2    1.0  0.0
299  45.0  1.0  3.0     110.0  264.0  0.0      1.0    132.0    0.0      1.2    1.0  0.0
300  68.0  1.0  0.0     144.0  193.0  1.0      1.0    160.0    0.0      3.4    1.0  0.0
301  57.0  1.0  0.0     120.0  234.0  0.0      0.0    115.0    1.0      1.2    1.0  1.0
302  58.0  0.0  1.0     130.0  236.0  0.0      0.0    160.0    0.0      0.0    2.0  1.0

303 rows × 14 columns
```

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 58.0 | 1.0 | 3.0 | 145.0 | 234.0 | 1.0 | 0.0 | 150.0 | 0.0 | 2.3 | 0.0 | 0.0 |
| **1** | 37.0 | 1.0 | 2.0 | 130.0 | 250.0 | 0.0 | 1.0 | 187.0 | 0.0 | 3.5 | 0.0 | 0.0 |
| **2** | 41.0 | 0.0 | 1.0 | 130.0 | NaN | NaN | 0.0 | 172.0 | 0.0 | 1.4 | 2.0 | 0.0 |
| **3** | 56.0 | 1.0 | 1.0 | 120.0 | NaN | 0.0 | 1.0 | NaN | NaN | 0.8 | 2.0 | 0.0 |
| **4** | NaN | NaN | 0.0 | 120.0 | 354.0 | 0.0 | NaN | 163.0 | 1.0 | 0.6 | 2.0 | 0.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **298** | 57.0 | 0.0 | 0.0 | 140.0 | 241.0 | 0.0 | 1.0 | 123.0 | 1.0 | 0.2 | 1.0 | NaN |
| **299** | 45.0 | NaN | 3.0 | 110.0 | 264.0 | NaN | 1.0 | 132.0 | 0.0 | 1.2 | 1.0 | 0.0 |
| **300** | 68.0 | 1.0 | 0.0 | 144.0 | 193.0 | 1.0 | 1.0 | NaN | 0.0 | 3.4 | 1.0 | NaN |

## Computing Mean

```
print(df.mean())
```

```
age          55.115512
sex           0.755776
cp            0.693069
trestbps    127.838284
chol        242.254125
fbs           0.122112
restecg       0.349835
thalach     153.056106
exang         0.231023
oldpeak       0.754785
slope         1.600660
ca            0.498350
thal          2.188119
target        0.676568
dtype: float64
```

## Computing Median

```
print(df.median())
```

```
age          58.0
sex           1.0
cp            0.0
trestbps    120.0
chol        234.0
fbs           0.0
restecg       0.0
thalach     160.0
exang         0.0
```

```
oldpeak       0.0
slope         2.0
ca            0.0
thal          2.0
target        1.0
dtype: float64
```

## Computing Mode

```
for col in df.columns:
  print("Mode of column ",col," is",df[col].mode()[0])
```

```
Mode of column  age  is 58
Mode of column  sex  is 1
Mode of column  cp  is 0
Mode of column  trestbps  is 120
Mode of column  chol  is 197
Mode of column  fbs  is 0
Mode of column  restecg  is 1
Mode of column  thalach  is 162
Mode of column  exang  is 0
Mode of column  oldpeak  is 0.0
Mode of column  slope  is 2
Mode of column  ca  is 0
Mode of column  thal  is 2
Mode of column  target  is 1
```

```
for col in df.columns:
  print(col,max(df[col])-min(df[col]))
```

```
age 47.0
sex 1.0
cp 3.0
trestbps 106.0
chol 268.0
fbs 1.0
restecg 2.0
thalach 106.0
exang 1.0
oldpeak 6.2
slope 2.0
ca 4.0
thal 3.0
target 1.0
```

```
Q1 = np.percentile(df, 25, interpolation = 'midpoint')
Q3 = np.percentile(df, 25, interpolation = 'midpoint')

IQR = Q3 - Q1

print(IQR)
print(Q1)
print(Q3)
print(Q3)
```

```
0.0
0.0
0.0
```

```
IQR = stats.iqr(df, interpolation = 'midpoint')
```

```
print(IQR)
```

```
55.0
```

```
for col in df.columns:
  IQR = stats.iqr(df[col], interpolation = 'midpoint')
  print("IQR for feature ",col," ",IQR)
```

```
IQR for feature  age   13.5
IQR for feature  sex   1.0
IQR for feature  cp   2.0
IQR for feature  trestbps   20.0
IQR for feature  chol   63.5
IQR for feature  fbs   0.0
IQR for feature  restecg   1.0
IQR for feature  thalach   32.5
IQR for feature  exang   1.0
IQR for feature  oldpeak   1.6
IQR for feature  slope   1.0
IQR for feature  ca   1.0
IQR for feature  thal   1.0
IQR for feature  target   1.0
```

```
df.std()
```

```
age          9.082101
sex          0.466011
cp           1.032052
trestbps    17.538143
chol        51.830751
fbs          0.356198
restecg      0.525860
thalach     22.905161
exang        0.469794
oldpeak      1.161075
slope        0.616226
ca           1.022606
thal         0.612277
target       0.498835
dtype: float64
```

```
pd.qcut(range(4),4,labels=["typical","atypical","non-anginal","asymptomatic"])
```

```
['typical', 'atypical', 'non-anginal', 'asymptomatic']
```

```
Categories (4, object): ['typical' < 'atypical' < 'non-anginal' < 'asymptomatic']
```

```
df
```

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | tha |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 63 | 1 | 3 | 145 | 233 | 1 | 0 | 150 | 0 | 2.3 | 0 | 0 | |
| **1** | 37 | 1 | 2 | 130 | 250 | 0 | 1 | 187 | 0 | 3.5 | 0 | 0 | |
| **2** | 41 | 0 | 1 | 130 | 204 | 0 | 0 | 172 | 0 | 1.4 | 2 | 0 | |
| **3** | 56 | 1 | 1 | 120 | 236 | 0 | 1 | 178 | 0 | 0.8 | 2 | 0 | |
| **4** | 57 | 0 | 0 | 120 | 354 | 0 | 1 | 163 | 1 | 0.6 | 2 | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **298** | 57 | 0 | 0 | 140 | 241 | 0 | 1 | 123 | 1 | 0.2 | 1 | 0 | |
| **299** | 45 | 1 | 3 | 110 | 264 | 0 | 1 | 132 | 0 | 1.2 | 1 | 0 | |
| **300** | 68 | 1 | 0 | 144 | 193 | 1 | 1 | 141 | 0 | 3.4 | 1 | 2 | |
| **301** | 57 | 1 | 0 | 130 | 131 | 0 | 1 | 115 | 1 | 1.2 | 1 | 1 | |
| **302** | 57 | 0 | 1 | 130 | 236 | 0 | 0 | 174 | 0 | 0.0 | 1 | 1 | |

303 rows × 14 columns

```
pd.cut(df.cp, bins=4, labels=["typical","atypical","non-anginal","asymptomatic"],right=T
```

```
0      asymptomatic
1       non-anginal
2          atypical
3          atypical
4           typical
           ...
298         typical
299    asymptomatic
300         typical
301         typical
302        atypical
Name: cp, Length: 303, dtype: category
Categories (4, object): ['typical' < 'atypical' < 'non-anginal' < 'asymptomatic']
```

```
df
```

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | tha |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 63 | 1 | 3 | 145 | 233 | 1 | 0 | 150 | 0 | 2.3 | 0 | 0 | |
| **1** | 37 | 1 | 2 | 130 | 250 | 0 | 1 | 187 | 0 | 3.5 | 0 | 0 | |
| **2** | 41 | 0 | 1 | 130 | 204 | 0 | 0 | 172 | 0 | 1.4 | 2 | 0 | |
| **3** | 56 | 1 | 1 | 120 | 236 | 0 | 1 | 178 | 0 | 0.8 | 2 | 0 | |
| **4** | 57 | 0 | 0 | 120 | 354 | 0 | 1 | 163 | 1 | 0.6 | 2 | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **298** | 57 | 0 | 0 | 140 | 241 | 0 | 1 | 123 | 1 | 0.2 | 1 | 0 | |

```
labels=["typical angina","atypical angina","non-anginal pain","asymptomatic"]
df["Class"]=pd.cut(df.cp, bins=4, right=True,labels=labels)
df[["Class","cp"]]
df0=df[df.Class == labels[0]]
df1 = df[df.Class == labels[1]]
df2 = df[df.Class == labels[2]]
df3 = df[df.Class == labels[3]]
```

```
df0
```

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | tha |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **4** | 57 | 0 | 0 | 120 | 354 | 0 | 1 | 163 | 1 | 0.6 | 2 | 0 | |
| **5** | 57 | 1 | 0 | 140 | 192 | 0 | 1 | 148 | 0 | 0.4 | 1 | 0 | |
| **10** | 54 | 1 | 0 | 140 | 239 | 0 | 1 | 160 | 0 | 1.2 | 2 | 0 | |
| **18** | 43 | 1 | 0 | 150 | 247 | 0 | 1 | 171 | 0 | 1.5 | 2 | 0 | |
| **20** | 59 | 1 | 0 | 135 | 234 | 0 | 1 | 161 | 0 | 0.5 | 1 | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **296** | 63 | 0 | 0 | 124 | 197 | 0 | 1 | 136 | 1 | 0.0 | 1 | 0 | |
| **297** | 59 | 1 | 0 | 164 | 176 | 1 | 0 | 90 | 0 | 1.0 | 1 | 2 | |

```
df1
```

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | tha |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **2** | 41 | 0 | 1 | 130 | 204 | 0 | 0 | 172 | 0 | 1.4 | 2 | 0 | |
| **3** | 56 | 1 | 1 | 120 | 236 | 0 | 1 | 178 | 0 | 0.8 | 2 | 0 | |
| **6** | 56 | 0 | 1 | 140 | 294 | 0 | 0 | 153 | 0 | 1.3 | 1 | 0 | |
| **7** | 44 | 1 | 1 | 120 | 263 | 0 | 1 | 173 | 0 | 0.0 | 2 | 0 | |
| **12** | 49 | 1 | 1 | 130 | 266 | 0 | 1 | 171 | 0 | 0.6 | 2 | 0 | |
| **25** | 71 | 0 | 1 | 160 | 302 | 0 | 1 | 162 | 0 | 0.4 | 2 | 2 | |
| **30** | 41 | 0 | 1 | 105 | 198 | 0 | 1 | 168 | 0 | 0.0 | 2 | 1 | |
| **32** | 44 | 1 | 1 | 130 | 219 | 0 | 0 | 188 | 0 | 0.0 | 2 | 0 | |
| **41** | 48 | 1 | 1 | 130 | 245 | 0 | 0 | 180 | 0 | 0.2 | 1 | 0 | |
| **45** | 52 | 1 | 1 | 120 | 325 | 0 | 1 | 172 | 0 | 0.2 | 2 | 0 | |
| **55** | 52 | 1 | 1 | 134 | 201 | 0 | 1 | 158 | 0 | 0.8 | 2 | 1 | |
| **61** | 54 | 1 | 1 | 108 | 309 | 0 | 1 | 156 | 0 | 0.0 | 2 | 0 | |
| **63** | 41 | 1 | 1 | 135 | 203 | 0 | 1 | 132 | 0 | 0.0 | 1 | 0 | |
| **67** | 45 | 0 | 1 | 130 | 234 | 0 | 0 | 175 | 0 | 0.6 | 1 | 0 | |
| **68** | 44 | 1 | 1 | 120 | 220 | 0 | 1 | 170 | 0 | 0.0 | 2 | 0 | |
| **72** | 29 | 1 | 1 | 130 | 204 | 0 | 0 | 202 | 0 | 0.0 | 2 | 0 | |
| **75** | 55 | 0 | 1 | 135 | 250 | 0 | 0 | 161 | 0 | 1.4 | 1 | 0 | |
| **77** | 59 | 1 | 1 | 140 | 221 | 0 | 1 | 164 | 1 | 0.0 | 2 | 0 | |
| **78** | 52 | 1 | 1 | 128 | 205 | 1 | 1 | 184 | 0 | 0.0 | 2 | 0 | |
| **81** | 45 | 1 | 1 | 128 | 308 | 0 | 0 | 170 | 0 | 0.0 | 2 | 0 | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **87** | 46 | 1 | 1 | 101 | 197 | 1 | 1 | 156 | 0 | 0.0 | 2 | 0 |
| **93** | 54 | 0 | 1 | 132 | 288 | 1 | 0 | 159 | 1 | 0.0 | 2 | 1 |
| **94** | 45 | 0 | 1 | 112 | 160 | 0 | 1 | 138 | 0 | 0.0 | 1 | 0 |
| **102** | 63 | 0 | 1 | 140 | 195 | 0 | 1 | 179 | 0 | 0.0 | 2 | 2 |
| **108** | 50 | 0 | 1 | 120 | 244 | 0 | 1 | 162 | 0 | 1.1 | 2 | 0 |
| **114** | 55 | 1 | 1 | 130 | 262 | 0 | 1 | 155 | 0 | 0.0 | 2 | 0 |
| **118** | 46 | 0 | 1 | 105 | 204 | 0 | 1 | 172 | 0 | 0.0 | 2 | 0 |
| **125** | 34 | 0 | 1 | 118 | 210 | 0 | 1 | 192 | 0 | 0.7 | 2 | 0 |
| **129** | 74 | 0 | 1 | 120 | 269 | 0 | 0 | 121 | 1 | 0.2 | 2 | 1 |

```
df2
```

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | tha |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 37 | 1 | 2 | 130 | 250 | 0 | 1 | 187 | 0 | 3.5 | 0 | 0 | |
| **8** | 52 | 1 | 2 | 172 | 199 | 1 | 1 | 162 | 0 | 0.5 | 2 | 0 | |
| **9** | 57 | 1 | 2 | 150 | 168 | 0 | 1 | 174 | 0 | 1.6 | 2 | 0 | |
| **11** | 48 | 0 | 2 | 130 | 275 | 0 | 1 | 139 | 0 | 0.2 | 2 | 0 | |
| **15** | 50 | 0 | 2 | 120 | 219 | 0 | 1 | 158 | 0 | 1.6 | 1 | 0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |

```
df3
```

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | tha |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 63 | 1 | 3 | 145 | 233 | 1 | 0 | 150 | 0 | 2.3 | 0 | 0 | |
| **13** | 64 | 1 | 3 | 110 | 211 | 0 | 0 | 144 | 1 | 1.8 | 1 | 0 | |
| **14** | 58 | 0 | 3 | 150 | 283 | 1 | 0 | 162 | 0 | 1.0 | 2 | 0 | |
| **17** | 66 | 0 | 3 | 150 | 226 | 0 | 1 | 114 | 0 | 2.6 | 0 | 0 | |
| **19** | 69 | 0 | 3 | 140 | 239 | 0 | 1 | 151 | 0 | 1.8 | 2 | 2 | |
| **24** | 40 | 1 | 3 | 140 | 199 | 0 | 1 | 178 | 1 | 1.4 | 2 | 0 | |
| **34** | 51 | 1 | 3 | 125 | 213 | 0 | 0 | 125 | 1 | 1.4 | 2 | 1 | |
| **58** | 34 | 1 | 3 | 118 | 182 | 0 | 0 | 174 | 0 | 0.0 | 2 | 0 | |
| **62** | 52 | 1 | 3 | 118 | 186 | 0 | 0 | 190 | 0 | 0.0 | 1 | 0 | |
| **83** | 52 | 1 | 3 | 152 | 298 | 1 | 1 | 178 | 0 | 1.2 | 1 | 0 | |
| **100** | 42 | 1 | 3 | 148 | 244 | 0 | 0 | 178 | 0 | 0.8 | 2 | 2 | |
| **101** | 59 | 1 | 3 | 178 | 270 | 0 | 0 | 145 | 0 | 4.2 | 0 | 0 | |
| **106** | 69 | 1 | 3 | 160 | 234 | 1 | 0 | 131 | 0 | 0.1 | 1 | 1 | |
| **117** | 56 | 1 | 3 | 120 | 193 | 0 | 0 | 162 | 0 | 1.9 | 1 | 0 | |
| **147** | 60 | 0 | 3 | 150 | 240 | 0 | 1 | 171 | 0 | 0.9 | 2 | 0 | |
| **152** | 64 | 1 | 3 | 170 | 227 | 0 | 0 | 155 | 0 | 0.6 | 1 | 0 | |
| **222** | 65 | 1 | 3 | 138 | 282 | 1 | 0 | 174 | 0 | 1.4 | 1 | 1 | |
| **228** | 59 | 1 | 3 | 170 | 288 | 0 | 0 | 159 | 0 | 0.2 | 1 | 0 | |
| **254** | 59 | 1 | 3 | 160 | 273 | 0 | 0 | 125 | 0 | 0.0 | 2 | 0 | |
| **259** | 38 | 1 | 3 | 120 | 231 | 0 | 1 | 182 | 1 | 3.8 | 1 | 0 | |
| **271** | 61 | 1 | 3 | 134 | 234 | 0 | 1 | 145 | 0 | 2.6 | 1 | 2 | |
| **286** | 59 | 1 | 3 | 134 | 204 | 0 | 1 | 162 | 0 | 0.8 | 2 | 2 | |

```
print(labels[0])
print()
print("1.Mean")
print(df0.mean())
print()
print("2.Median")
print(df0.median())
print()
print("3.Mode")
print(df0.mode().transpose())
print()
print("4.Range")
for col in df0.columns:
    if(col!='Class'):
```

```
    if(col!='Class'):
        #omitting Class because its a string
        print(col,max(df0[col])-min(df0[col]))
print()
print("5.IQR")
for col in df0.columns:
    if(col!='Class'):
        IQR = stats.iqr(df0[col], interpolation = 'midpoint')
        print("IQR for feature ",col," ",IQR)
        print()
print()
print("6.Standard Deviation")
print(df0.std())
```

```
cp 0
trestbps 100
chol 278
fbs 1
restecg 2
thalach 115
exang 1
oldpeak 6.2
slope 2
ca 4
thal 3
target 1

5.IQR
IQR for feature  age   11.0

IQR for feature  sex   1.0

IQR for feature  cp   0.0

IQR for feature  trestbps   20.0

IQR for feature  chol   73.0

IQR for feature  fbs   0.0

IQR for feature  restecg   1.0

IQR for feature  thalach   33.5

IQR for feature  exang   1.0

IQR for feature  oldpeak   2.0500000000000003

IQR for feature  slope   1.0

IQR for feature  ca   2.0

IQR for feature  thal   1.0

IQR for feature  target   1.0
```

```
6.Standard Deviation
age          8.312752
sex          0.446927
cp           0.000000
trestbps    18.036141
chol        51.540390
fbs          0.332873
restecg      0.541674
thalach     22.999317
exang        0.498199
oldpeak      1.297559
slope        0.589978
ca           1.057586
thal         0.678423
target       0.446927
dtype: float64
```

Q7) Comparing the statistical values in the graph for the original dataset with the values of the changed dataset based on class labels, we cn observe that:-

i) One observation we can make is regarding the mean, where we see that the Mean of the individual Classes do not vary much with respect to the mean of all the values of the dataset taken together.

ii) Since median is the 2nd quartile value, it also does not change for the individual classes with respect to the overall dataset.

iii) Mode shows more variation than the mean or median since it gives us the most frequently occuring values and spliting the dataset into classes tends to affect the frequency of particular values too. So, mode is different for many features.

Q8) Line Plots showing the variations of :-

(i) Age vs Chest Pain Type

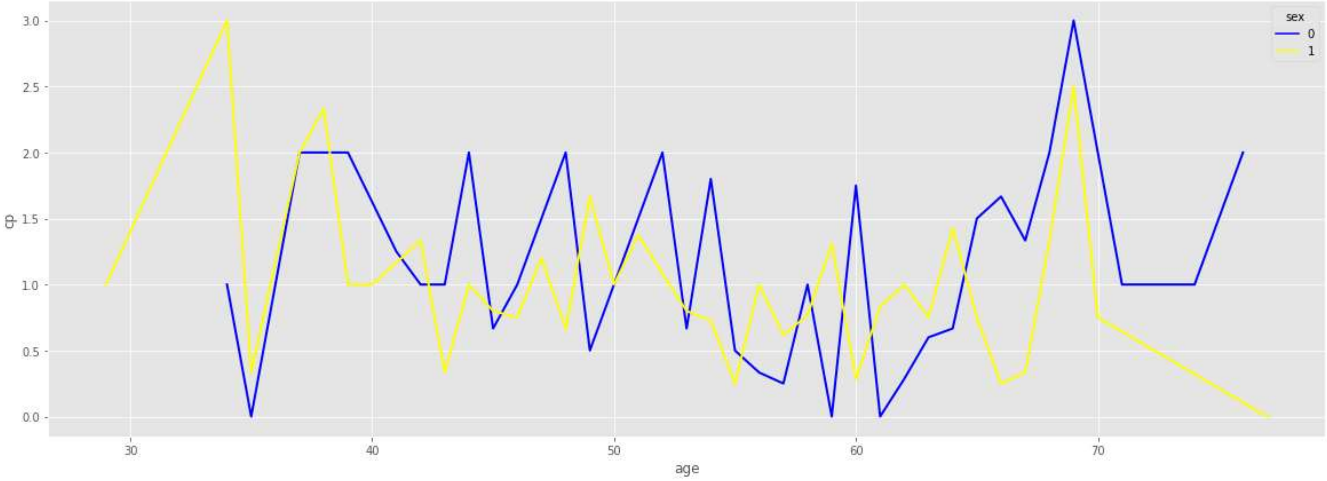
(ii) Chest Pain Type vs Cholestrol

(iii) Age vs Thal

(iv) Sex vs Chest Pain type

```
plt.figure(figsize= (20,7))
plt.figure(figsize=(20,7))
plt.style.use('ggplot')
sns.lineplot(df['age'],df['cp'],linewidth=2,hue=df['sex'],ci=None,palette=['blue','yello
```

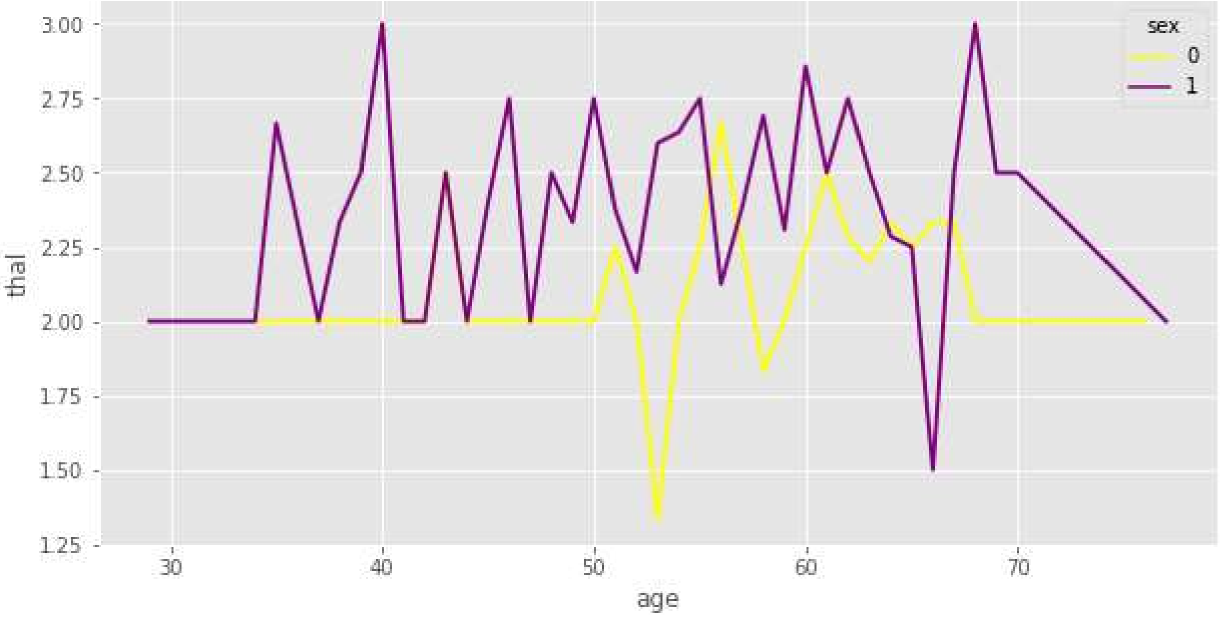
```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass t
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f7095351e48>
```



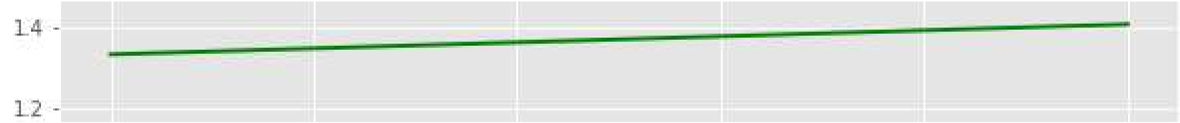
```
plt.figure(figsize=(10,5))
plt.style.use('ggplot')
sns.lineplot(df["cp"],df['chol'],linewidth=2,hue=df['sex'],ci=None,palette=['purple','ye
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass t
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f709507d240>
```

```
plt.figure(figsize=(10,5))
plt.style.use('ggplot')
sns.lineplot(df["age"],df['thal'],linewidth=2,hue=df['sex'],ci=None,palette=['yellow','p
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass t
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f617ad5d860>
```



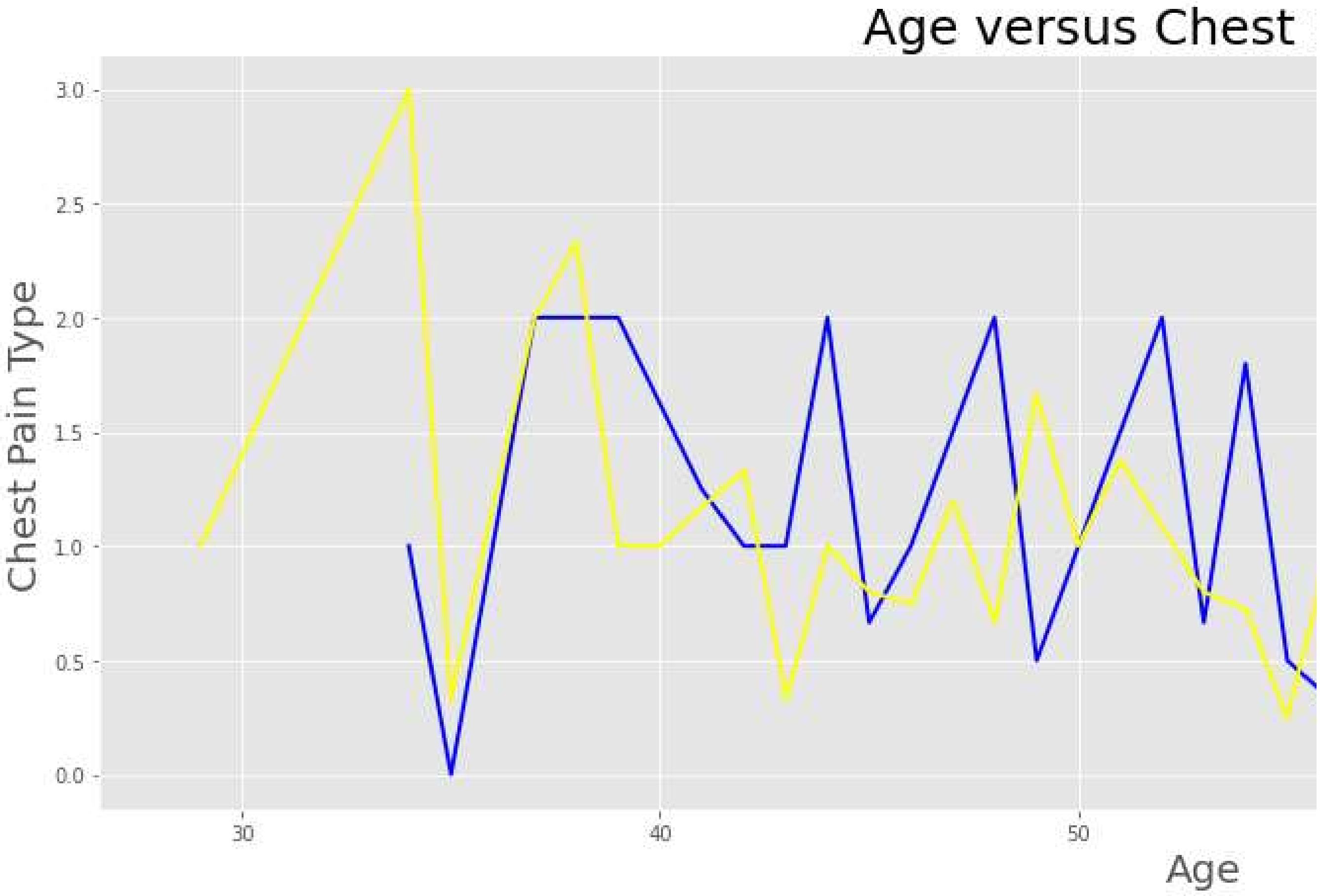
```
plt.figure(figsize=(10,5))
plt.style.use('ggplot')
sns.lineplot(df["sex"],df['cp'],linewidth=2,hue=df['target'],ci=None,palette=['red','gre
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass t
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f7095134828>
```



Plots Shown along with the labels

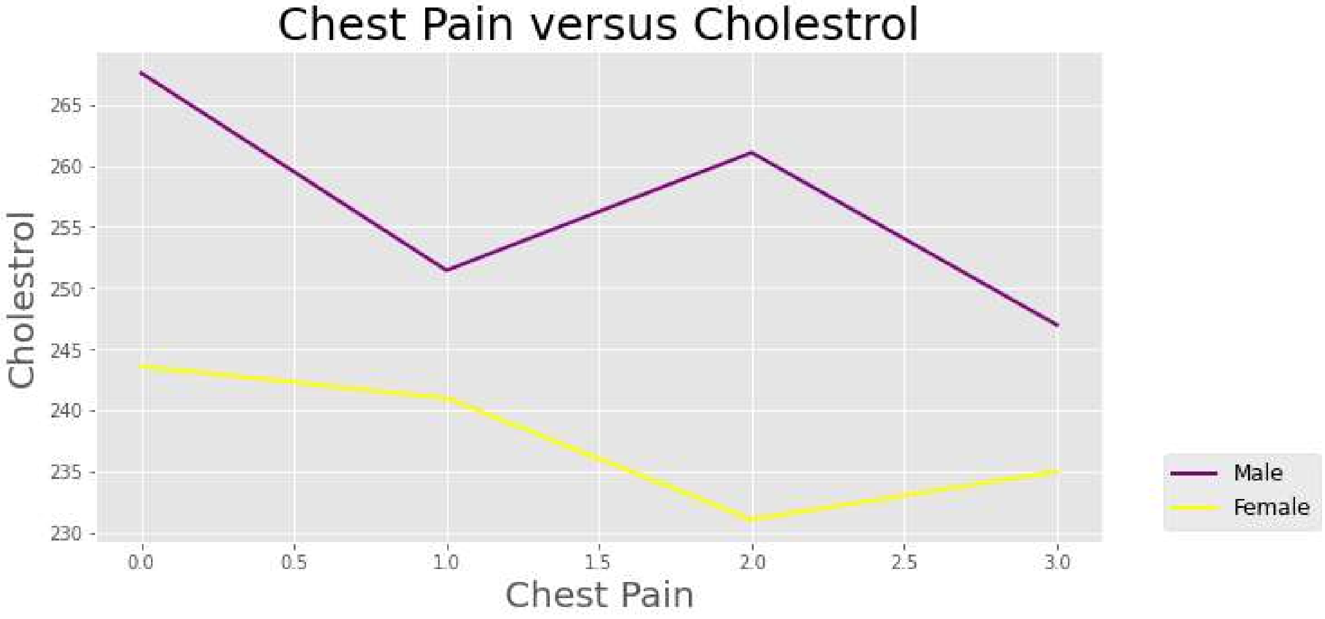
```
plt.figure(figsize= (20,7))
plt.style.use('ggplot')
sns.lineplot(df['age'],df['cp'],linewidth=2,hue=df['sex'],ci=None,palette=['blue','yell
plt.xlabel("Age",fontsize=20)
plt.ylabel("Chest Pain Type",fontsize=20)
plt.title("Age versus Chest Pain Type",fontsize=25)
plt.legend(labels=['Male','Female'],frameon=True,fontsize='large',bbox_to_anchor=(1.05,
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass t
  FutureWarning
```



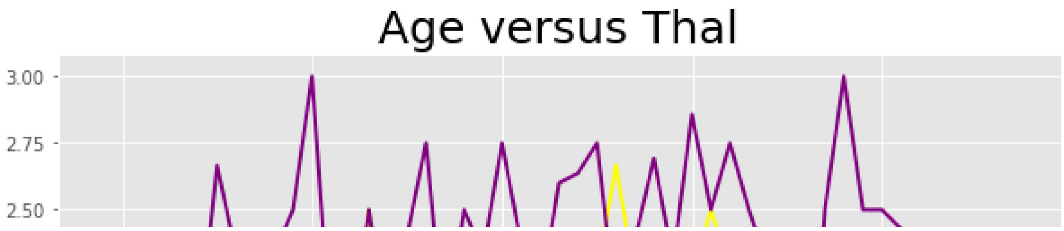
```
plt.figure(figsize=(10,5))
plt.style.use('ggplot')
sns.lineplot(df["cp"],df['chol'],linewidth=2,hue=df['sex'],ci=None,palette=['purple','ye
plt.xlabel("Chest Pain",fontsize=20)
plt.ylabel("Cholestrol",fontsize=20)
plt.title("Chest Pain versus Cholestrol",fontsize=25)
plt.legend(labels=['Male','Female'],frameon=True,fontsize='large',bbox_to_anchor=(1.05,
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass t
  FutureWarning
```



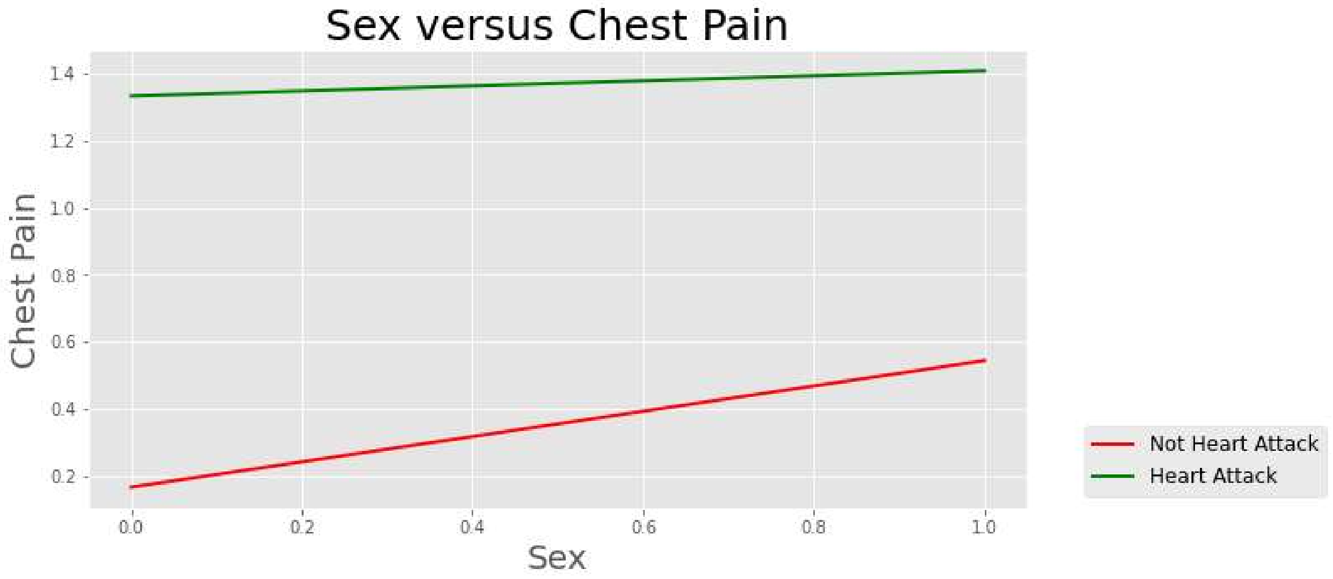
```
plt.figure(figsize=(10,5))
plt.style.use('ggplot')
sns.lineplot(df["age"],df['thal'],linewidth=2,hue=df['sex'],ci=None,palette=['yellow','p
plt.xlabel("Age",fontsize=20)
plt.ylabel("Thal",fontsize=20)
plt.title("Age versus Thal",fontsize=25)
plt.legend(labels=['Male','Female'],frameon=True,fontsize='large',bbox_to_anchor=(1.05,
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass t
  FutureWarning
```



```
plt.figure(figsize=(10,5))
plt.style.use('ggplot')
sns.lineplot(df["sex"],df['cp'],linewidth=2,hue=df['target'],ci=None,palette=['red','gre
plt.xlabel("Sex",fontsize=20)
plt.ylabel("Chest Pain",fontsize=20)
plt.title("Sex versus Chest Pain",fontsize=25)
plt.legend(labels=['Not Heart Attack','Heart Attack'],frameon=True,fontsize='large',bbox
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass t
  FutureWarning
```



Q10)

Observations from the above plots:-

(i) For Age vs Chest Pain Type plot,we have an irregular graph for both male and female.

(ii) The Chest Pain vs Cholestrol plot indicates that the cholestrol in men is higher with increase in chest pain than in women.

(iii) For Cholestrol vs Chest Pain also we have an irregular graph for both male and female.

(iv) If the person has a lower chest pain, then he has a lesser chance of getting a heart attack.

(v) Sex Vs Cholestrol The graph is a linear graph which gives us the conculsion that Female's have more chances of not getting a heart attack even with a high Chest pain (i.e they can bear a lot of Chest Pain) than their male counterparts who are not able to bear it to that extent.

Conclusions:-

Women have higher cholestrol compared to men according to these plots.

Men have higher chest pain which could also be inferred from the above plots.