heart-disease-diagnostic-analysis

March 28, 2024

[130]:
```
# importing the required libraries for calculation

import numpy as np
import pandas as pd
```

[131]:
```
# importing the required libraries for visualisation

import matplotlib.pyplot as plt
import seaborn as sns
```

[132]:
```
# importing the required libraries for prediction

from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
```

[133]:
```
# Loading the dataset

data = pd.read_csv('Downloads\Heart Disease data.csv')
```

Copying the data for further exploration and Analysis

[134]:
```
hrt_dis = data.copy()
```

[135]:
```
# Loading the dataset

hrt_dis
```

[135]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	\
0	52	1	0	125	212	0	1	168	0	1.0	
1	53	1	0	140	203	1	0	155	1	3.1	
2	70	1	0	145	174	0	1	125	1	2.6	
3	61	1	0	148	203	0	1	161	0	0.0	
4	62	0	0	138	294	1	1	106	0	1.9	
…	…	…	..	…	…	…	…	…	…	…	
1020	59	1	1	140	221	0	1	164	1	0.0	
1021	60	1	0	125	258	0	0	141	1	2.8	
1022	47	1	0	110	275	0	0	118	1	1.0	

```
1023   50    0   0       110   254    0        0      159      0      0.0
1024   54    1   0       120   188    0        1      113      0      1.4

      slope  ca  thal  target
0         2   2     3       0
1         0   0     3       0
2         0   0     3       0
3         2   1     3       0
4         1   3     2       0
…     …   ..  …     …
1020      2   0     2       1
1021      1   1     3       0
1022      1   1     2       0
1023      2   0     2       1
1024      1   1     3       0

[1025 rows x 14 columns]
```

Exploring the Data

[136]:
```
# Finding the total records and features provided in the dataset

hrt_dis.shape
```

[136]:
```
(1025, 14)
```

[137]:
```
# Finding the dimension

hrt_dis.ndim
```

[137]:
```
2
```

[138]:
```
# Memory storage used for each column

hrt_dis.memory_usage(index = False,deep = True)
```

[138]:
```
age         8200
sex         8200
cp          8200
trestbps    8200
chol        8200
fbs         8200
restecg     8200
thalach     8200
exang       8200
oldpeak     8200
slope       8200
```

```
ca          8200
thal        8200
target      8200
dtype: int64
```

[139]:
```
# Getting information regarding all columns

hrt_dis.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
 1   sex       1025 non-null   int64
 2   cp        1025 non-null   int64
 3   trestbps  1025 non-null   int64
 4   chol      1025 non-null   int64
 5   fbs       1025 non-null   int64
 6   restecg   1025 non-null   int64
 7   thalach   1025 non-null   int64
 8   exang     1025 non-null   int64
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64
 11  ca        1025 non-null   int64
 12  thal      1025 non-null   int64
 13  target    1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

[140]:
```
# Gettinng statistical information of all numeric features of dataset

hrt_dis.describe()
```

[140]:

	age	sex	cp	trestbps	chol	\
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	
mean	54.434146	0.695610	0.942439	131.611707	246.00000	
std	9.072290	0.460373	1.029641	17.516718	51.59251	
min	29.000000	0.000000	0.000000	94.000000	126.00000	
25%	48.000000	0.000000	0.000000	120.000000	211.00000	
50%	56.000000	1.000000	1.000000	130.000000	240.00000	
75%	61.000000	1.000000	2.000000	140.000000	275.00000	
max	77.000000	1.000000	3.000000	200.000000	564.00000	

	fbs	restecg	thalach	exang	oldpeak	\
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	

```
mean      0.149268     0.529756   149.114146     0.336585     1.071512
std       0.356527     0.527878    23.005724     0.472772     1.175053
min       0.000000     0.000000    71.000000     0.000000     0.000000
25%       0.000000     0.000000   132.000000     0.000000     0.000000
50%       0.000000     1.000000   152.000000     0.000000     0.800000
75%       0.000000     1.000000   166.000000     1.000000     1.800000
max       1.000000     2.000000   202.000000     1.000000     6.200000

              slope           ca         thal       target
count  1025.000000  1025.000000  1025.000000  1025.000000
mean      1.385366     0.754146     2.323902     0.513171
std       0.617755     1.030798     0.620660     0.500070
min       0.000000     0.000000     0.000000     0.000000
25%       1.000000     0.000000     2.000000     0.000000
50%       1.000000     0.000000     2.000000     1.000000
75%       2.000000     1.000000     3.000000     1.000000
max       2.000000     4.000000     3.000000     1.000000
```

[141]:
```
# for getting the total number of unique values

hrt_dis["sex"].value_counts()
```

[141]:
```
1    713
0    312
Name: sex, dtype: int64
```

0 –> Female

1 –> Male

[142]:
```
hrt_dis["cp"].value_counts()
```

[142]:
```
0    497
2    284
1    167
3     77
Name: cp, dtype: int64
```

0 –> typical angina

1 –> atypical angina

2 –> non-anginal pain

3 –> asymptomatic

[143]:
```
hrt_dis["fbs"].value_counts()
```

[143]:
```
0    872
1    153
```

```
Name: fbs, dtype: int64
```

Normal = > 120 mg/dl

1 –> true

0 –> false

[144]:
```
hrt_dis["restecg"].value_counts()
```

[144]:
```
1    513
0    497
2     15
Name: restecg, dtype: int64
```

0 –> normal

1 –> having ST-T wave abnormality

2 –> showing probable or definite left ventricular hypertrophy by Estes' criteria

[145]:
```
hrt_dis["exang"].value_counts()
```

[145]:
```
0    680
1    345
Name: exang, dtype: int64
```

1 –> yes

0 –> no

[146]:
```
hrt_dis["slope"].value_counts()
```

[146]:
```
1    482
2    469
0     74
Name: slope, dtype: int64
```

0 –> upsloping

1 –> flat

2 –> downsloping

[147]:
```
hrt_dis["ca"].value_counts()
```

[147]:
```
0    578
1    226
2    134
3     69
4     18
Name: ca, dtype: int64
```

[148]:
```
hrt_dis["thal"].value_counts()
```

[148]:
```
2    544
3    410
1     64
0      7
Name: thal, dtype: int64
```

0 –> normal

1 –> fixed defect

2 –> reversable defect

[149]:
```
hrt_dis["target"].value_counts()
```

[149]:
```
1    526
0    499
Name: target, dtype: int64
```

0 –> No Heart Disease

1 –> Heart Disease

Data Cleaning

[150]:
```
# Checking for missing values

hrt_dis.isnull().sum()
```

[150]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

There is NO missing values in the dataset, proceed to further analysis

[151]:
```
# Defining function for converting numerical data to categorical data
```

```
def hr_d (row):
    if row == 0:
        return 'Not Present'
    elif row == 1:
        return 'Present'
```

[152]:
```
#Creating new column and applying function

hrt_dis["heart_disease"] = hrt_dis['target'].apply(hr_d)
hrt_dis.head(5)
```

[152]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	\
0	52	1	0	125	212	0	1	168	0	1.0	2	
1	53	1	0	140	203	1	0	155	1	3.1	0	
2	70	1	0	145	174	0	1	125	1	2.6	0	
3	61	1	0	148	203	0	1	161	0	0.0	2	
4	62	0	0	138	294	1	1	106	0	1.9	1	

	ca	thal	target	heart_disease
0	2	3	0	Not Present
1	0	3	0	Not Present
2	0	3	0	Not Present
3	1	3	0	Not Present
4	3	2	0	Not Present

[153]:
```
# Defininf function for creating age category

def age_cat (a):
    if a < 18:
        return 'young'
    elif (a >= 18) &  (a < 40):
        return 'Adult'
    elif (a >= 40) & (a < 60):
        return "middle age"
    elif a >=60:
        return "senior citizen"
```

[154]:
```
# Applying the def to new column

hrt_dis["age_group"] = hrt_dis['age'].apply(age_cat)

hrt_dis.head(5)
```

[154]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	\
0	52	1	0	125	212	0	1	168	0	1.0	2	
1	53	1	0	140	203	1	0	155	1	3.1	0	
2	70	1	0	145	174	0	1	125	1	2.6	0	

```
3   61    1   0       148   203    0        1      161      0      0.0      2
4   62    0   0       138   294    1        1      106      0      1.9      1

   ca  thal  target heart_disease       age_group
0   2     3       0   Not Present      middle age
1   0     3       0   Not Present      middle age
2   0     3       0   Not Present  senior citizen
3   1     3       0   Not Present  senior citizen
4   3     2       0   Not Present  senior citizen
```

[155]:
```
# creating new column categorical sex data

hrt_dis['gender'] = np.where(hrt_dis['sex'] == 0, 'Female','Male')

hrt_dis.head(5)
```

[155]:
```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   52    1   0       125   212    0        1      168      0      1.0      2
1   53    1   0       140   203    1        0      155      1      3.1      0
2   70    1   0       145   174    0        1      125      1      2.6      0
3   61    1   0       148   203    0        1      161      0      0.0      2
4   62    0   0       138   294    1        1      106      0      1.9      1

   ca  thal  target heart_disease       age_group  gender
0   2     3       0   Not Present      middle age    Male
1   0     3       0   Not Present      middle age    Male
2   0     3       0   Not Present  senior citizen    Male
3   1     3       0   Not Present  senior citizen    Male
4   3     2       0   Not Present  senior citizen  Female
```

[156]:
```
# def function for new column relating to chol

def chol_range (c):
    if c  <200 :
        return 'Normal'
    elif (c >= 200) & (c <= 239):
        return 'Borderline high'
    elif c > 240:
        return 'High'
```

[157]:
```
hrt_dis['chol_range'] = hrt_dis['chol'].apply(chol_range)
hrt_dis.head(5)
```

[157]:
```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   52    1   0       125   212    0        1      168      0      1.0      2
1   53    1   0       140   203    1        0      155      1      3.1      0
2   70    1   0       145   174    0        1      125      1      2.6      0
```

```
3   61    1   0       148   203    0        1      161      0      0.0      2
4   62    0   0       138   294    1        1      106      0      1.9      1

   ca  thal  target heart_disease       age_group  gender       chol_range
0   2     3       0   Not Present      middle age    Male  Borderline high
1   0     3       0   Not Present      middle age    Male  Borderline high
2   0     3       0   Not Present  senior citizen    Male           Normal
3   1     3       0   Not Present  senior citizen    Male  Borderline high
4   3     2       0   Not Present  senior citizen  Female             High
```

[158]:
```
# Changing the column names for accessibility

hrt_dis.rename(columns={"cp":"chest_pain",
                        "trestbps":"rest_BPs",
                        "fbs":"FBS",
                        "exang":"Ex_Ang",
                        "oldpeak":"old_peak",
                        "target":"result"},inplace = True)
hrt_dis.head(5)
```

[158]:
```
   age  sex  chest_pain  rest_BPs  chol  FBS  restecg  thalach  Ex_Ang  \
0   52    1           0       125   212    0        1      168       0
1   53    1           0       140   203    1        0      155       1
2   70    1           0       145   174    0        1      125       1
3   61    1           0       148   203    0        1      161       0
4   62    0           0       138   294    1        1      106       0

   old_peak  slope  ca  thal  result heart_disease       age_group  gender  \
0       1.0      2   2     3       0   Not Present      middle age    Male
1       3.1      0   0     3       0   Not Present      middle age    Male
2       2.6      0   0     3       0   Not Present  senior citizen    Male
3       0.0      2   1     3       0   Not Present  senior citizen    Male
4       1.9      1   3     2       0   Not Present  senior citizen  Female

       chol_range
0  Borderline high
1  Borderline high
2           Normal
3  Borderline high
4             High
```

[159]:
```
# finding the unique values in the new columns

hrt_dis['heart_disease'].unique()
```

[159]:
```
array(['Not Present', 'Present'], dtype=object)
```

[160]:
```
hrt_dis['age_group'].unique()
```

[160]:
```
array(['middle age', 'senior citizen', 'Adult'], dtype=object)
```

[161]:
```
hrt_dis['gender'].unique()
```

[161]:
```
array(['Male', 'Female'], dtype=object)
```

[162]:
```
hrt_dis['chol_range'].unique()
```

[162]:
```
array(['Borderline high', 'Normal', 'High', None], dtype=object)
```

0.0.1 Analysis and Visualisation with case studies

[163]:
```
# getting the mean value attributes for Heart patients and non-heart patients

hrt_dis.groupby('result').mean()
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_15492\3838284946.py:3: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  hrt_dis.groupby('result').mean()
```

[163]:

result	age	sex	chest_pain	rest_BPs	chol	FBS	restecg	thalach	Ex_Ang	old_peak	slope	ca	thal
0	56.569138	0.827655	0.482966	134.106212	251.292585	0.164329	0.456914	139.130261	0.549098	1.600200	1.166333	1.158317	2.539078
1	52.408745	0.570342	1.378327	129.245247	240.979087	0.134981	0.598859	158.585551	0.134981	0.569962	1.593156	0.370722	2.119772

1. Correlation matrix of Numerical features

[164]:
```
# subset containing the the numerical columns

hrt_dis1 = hrt_dis.select_dtypes(include = np.number)
```

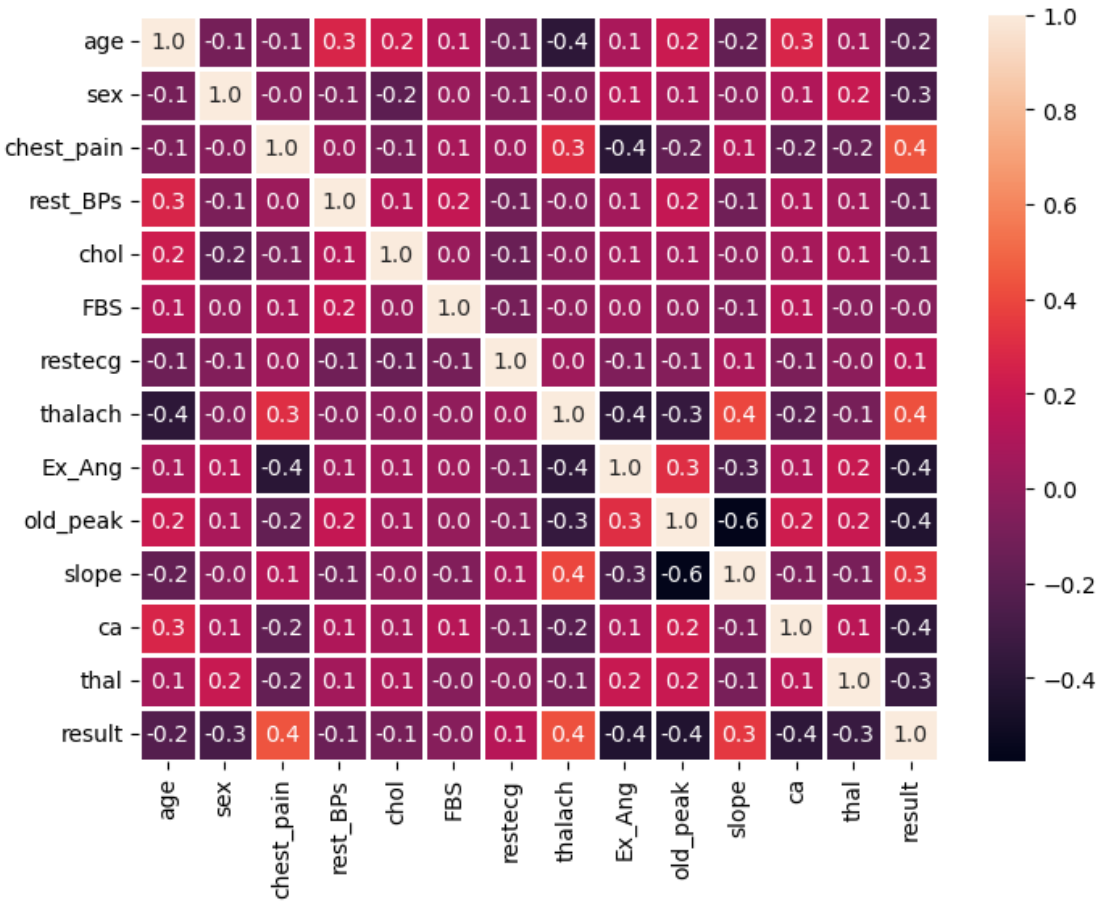
[165]:
```
# correlation metrix of Numerical features

hrt_dis1.corr
```

[165]:
```
<bound method DataFrame.corr of       age  sex  chest_pain  rest_BPs  chol  FBS
restecg  thalach  Ex_Ang  \
0      52    1           0       125   212    0        1      168       0
```

```
1       53    1           0        140   203    1         0       155        1
2       70    1           0        145   174    0         1       125        1
3       61    1           0        148   203    0         1       161        0
4       62    0           0        138   294    1         1       106        0
…    …  …          …        …    …   …      …       …      …
1020    59    1           1        140   221    0         1       164        1
1021    60    1           0        125   258    0         0       141        1
1022    47    1           0        110   275    0         0       118        1
1023    50    0           0        110   254    0         0       159        0
1024    54    1           0        120   188    0         1       113        0

      old_peak  slope  ca  thal  result
0          1.0      2   2     3       0
1          3.1      0   0     3       0
2          2.6      0   0     3       0
3          0.0      2   1     3       0
4          1.9      1   3     2       0
…         …    …   ..   …     …
1020       0.0      2   0     2       1
1021       2.8      1   1     3       0
1022       1.0      1   1     2       0
1023       0.0      2   0     2       1
1024       1.4      1   1     3       0

[1025 rows x 14 columns]>
```

[166]:
```
# Visualisation of correlation matrix using Heatmap


plt.figure(figsize=(8,6))
sns.heatmap(hrt_dis1.corr(), annot = True, fmt = '.1f', linewidth = 1)
```

[166]:
```
<Axes: >
```

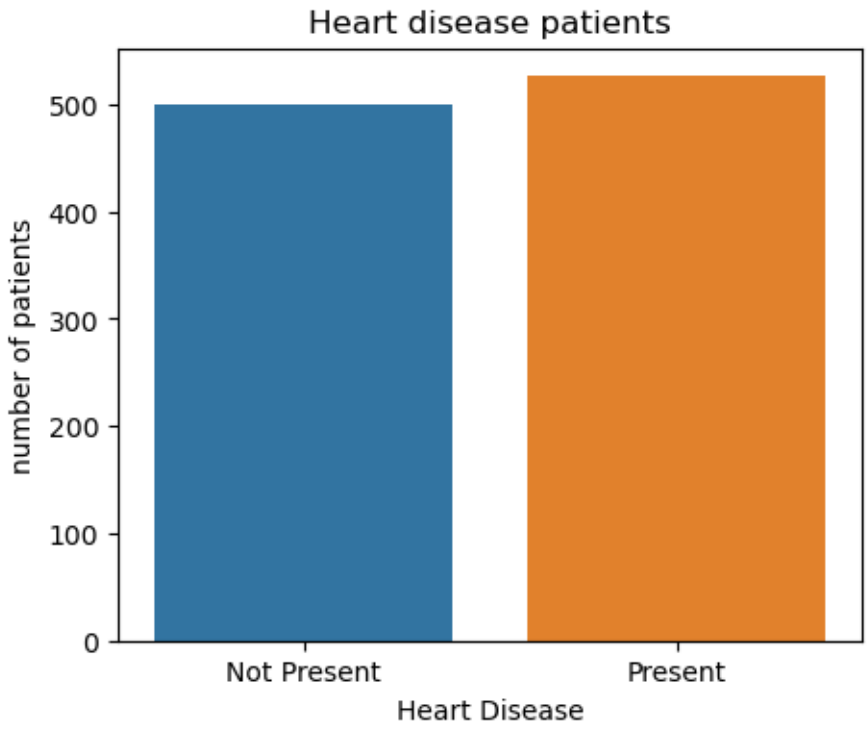


2. Number of Heart disease patients

[167]:
```
hrt_dis["heart_disease"].value_counts()
```

[167]:
```
Present        526
Not Present    499
Name: heart_disease, dtype: int64
```

[168]:
```
# Countplot repressentation of heart disease patients

plt.figure(figsize=(5,4))

sns.countplot (x = 'heart_disease', data = hrt_dis )
plt.xlabel ("Heart Disease")
plt.ylabel ("number of patients")
plt.title ("Heart disease patients")

plt.show()
```



Note:

–> There are a TOTAL OF 1025 PATIENTS, out of them, 526 PATIENTS HAVE HEART ISSUES which is Higher than the number of non-heart disease patients

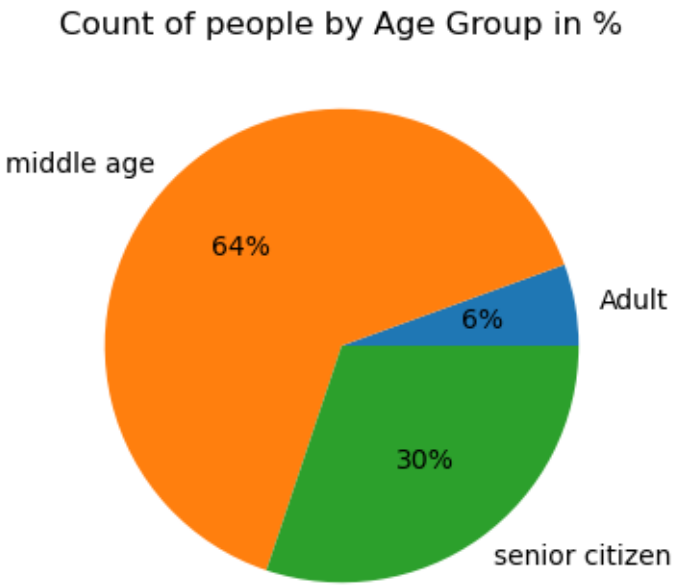
3. Number of people in each Age Group

[169]:
```
# number of patients in each age category

ag = hrt_dis.groupby (by ='age_group')['age'].count()
ag
```

[169]:
```
age_group
Adult              57
middle age        659
senior citizen    309
Name: age, dtype: int64
```

[170]:
```
# pie chart of age population in (%)

plt.figure(figsize=(5,4))

plt.pie(ag, labels=['Adult','middle age','senior citizen'], autopct = '%0.0f%%')
plt.title ("Count of people by Age Group in %")
```

[170]: Text(0.5, 1.0, 'Count of people by Age Group in %')



Note:

–> Most number of patients are MIDDLE-AGED

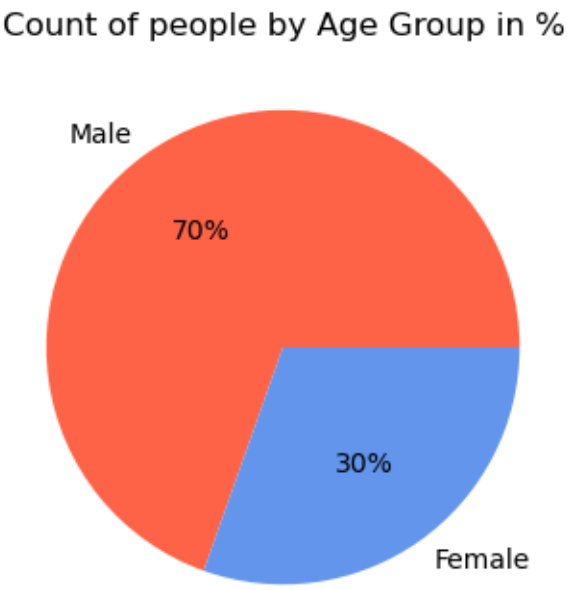
4. Number of people in each Gender

[171]:
```
g = hrt_dis['gender'].value_counts()
g
```

[171]:
```
Male      713
Female    312
Name: gender, dtype: int64
```

[172]:
```
# pie chart of gender population in (%)

plt.figure(figsize=(5,4))

plt.pie(g, labels=['Male','Female'], autopct = '%0.0f%%', colors = ['tomato',␣
↪'cornflowerblue'])
plt.title ("Count of people by Age Group in %")
```

[172]: Text(0.5, 1.0, 'Count of people by Age Group in %')



Note:

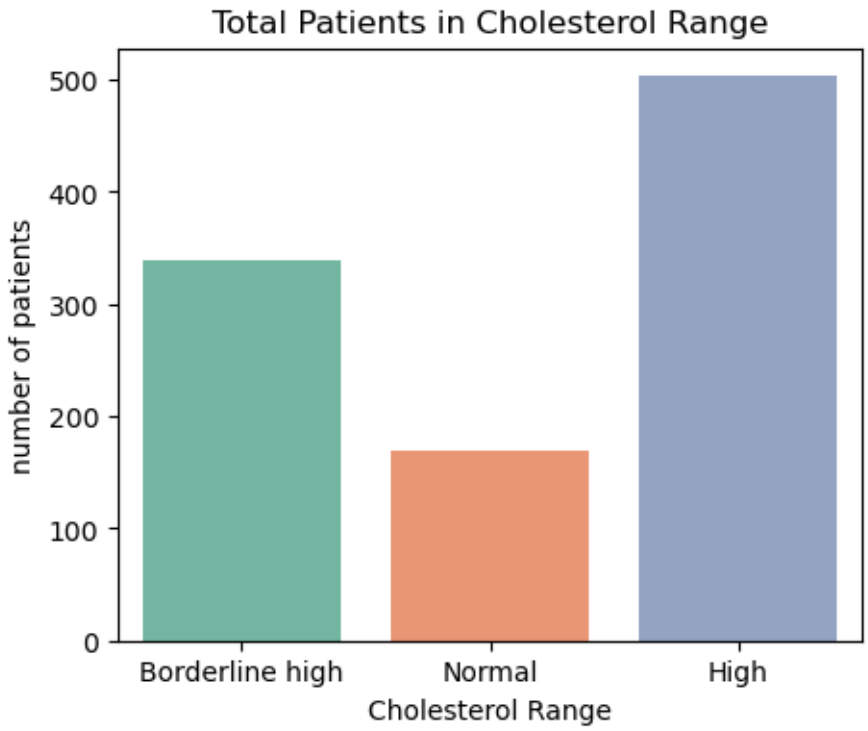
–> Total of 713 patients (70%) are men, while 312 are women(30%)

5. Total Patients in Cholesterol Range

[173]:
```
c = hrt_dis['chol_range'].value_counts()
c
```

[173]:
```
High               503
Borderline high    339
Normal             169
Name: chol_range, dtype: int64
```

[174]:
```
# Countplot  of patients chol range wise

plt.figure(figsize=(5,4))

sns.countplot (x = 'chol_range', data = hrt_dis, palette='Set2' )
plt.title ("Total Patients in Cholesterol Range")
plt.xlabel ("Cholesterol Range")
plt.ylabel ("number of patients")

plt.show()
```

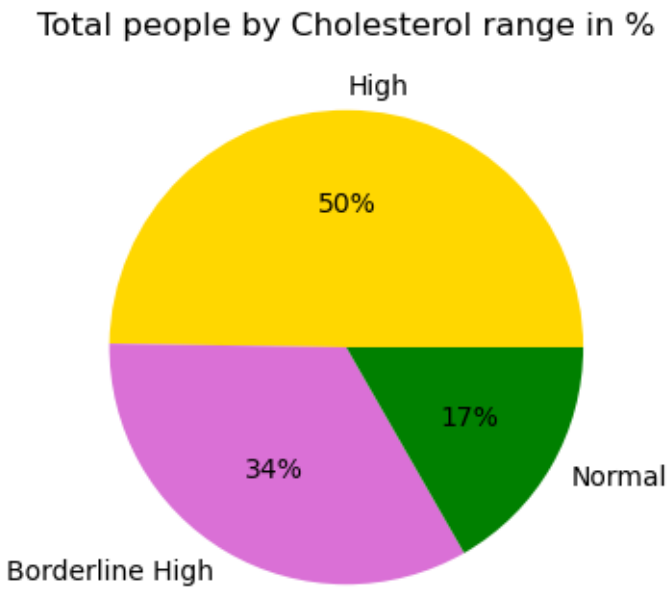


Note:

–> out of total, 503 patients are suffering from High cholesterol

[175]:
```
# pie chart of Cgolesterol range in (%)

plt.figure(figsize=(5,4))

plt.pie(c, labels=['High','Borderline High','Normal'], autopct = '%0.0f%%',␣
 ↪colors = ['gold', 'orchid', 'green'])
plt.title ("Total people by Cholesterol range in %")
```

[175]:
```
Text(0.5, 1.0, 'Total people by Cholesterol range in %')
```



Note:

–> 50% of population falls under the category 'High', which means having a rate of more than 240 mg/dL

6. Grouping of patients by Heart disease status and age

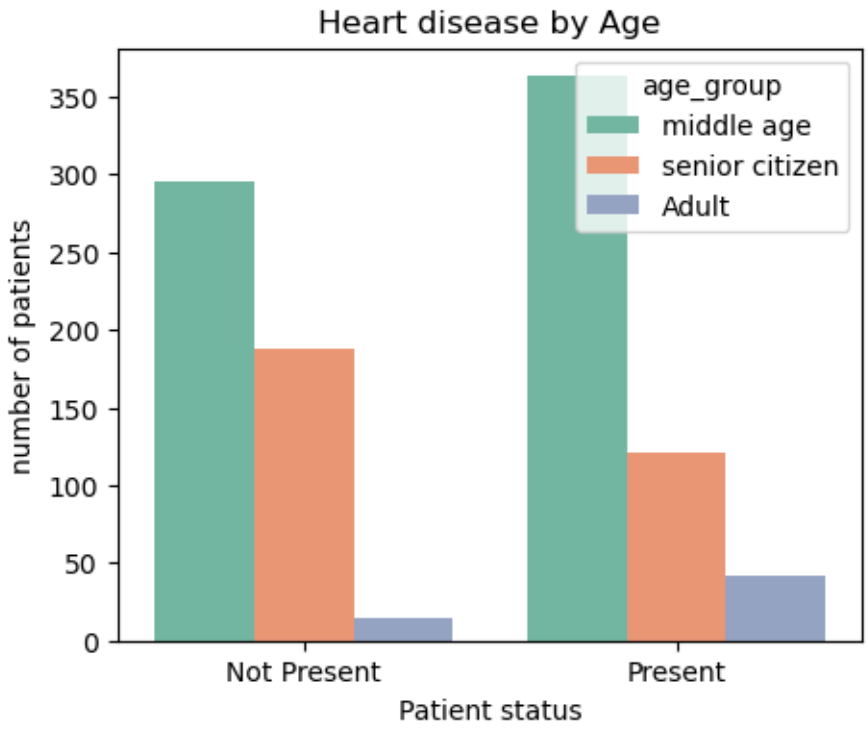
[176]:
```
# Filtring the data by age

hrt_dis.groupby(by = ["heart_disease","age_group"])['result'].count()
```

[176]:
```
heart_disease  age_group
Not Present    Adult              15
               middle age        296
               senior citizen    188
Present        Adult              42
               middle age        363
               senior citizen    121
Name: result, dtype: int64
```

[177]:
```
# Countplot of heart disease patients by age

plt.figure (figsize=(5,4))

sns.countplot (x = 'heart_disease', hue = 'age_group', data = hrt_dis, palette␣
↪= 'Set2')
plt.title ("Heart disease by Age")
```

```
plt.xlabel ("Patient status")
plt.ylabel ("number of patients")
```

[177]: `Text(0, 0.5, 'number of patients')`



Note:

–> Heart diseases are most commonly diagnosed in Adult and Middle-aged citizens

7. Grouping of patients by Heart disease and Gender

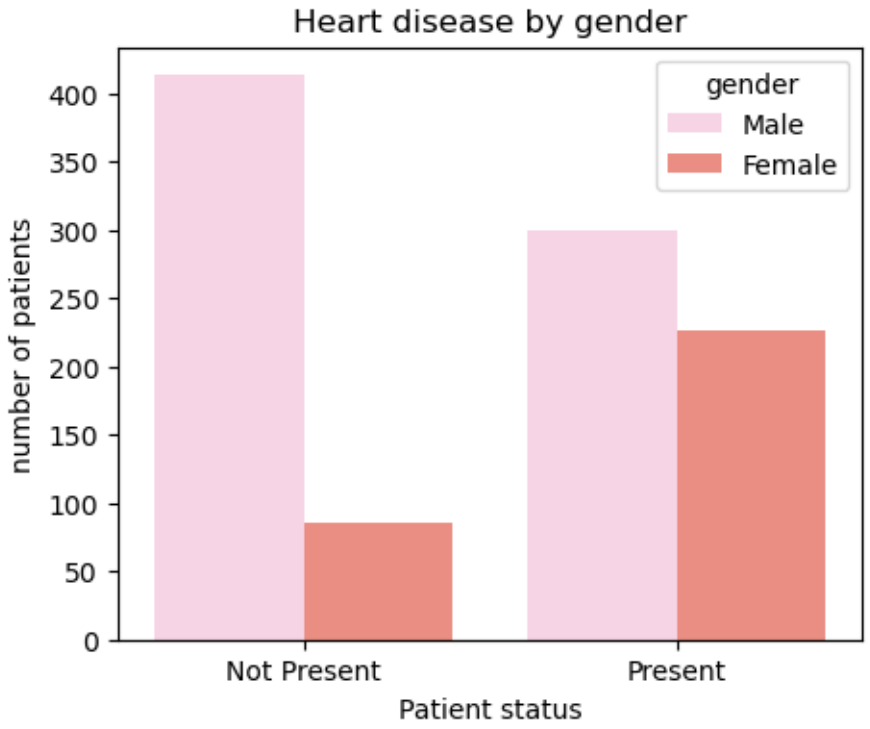
[178]:
```
# Filtring data by gender

hrt_dis.groupby(by=['heart_disease','gender'])['result'].count()
```

[178]:
```
heart_disease  gender
Not Present    Female     86
               Male      413
Present        Female    226
               Male      300
Name: result, dtype: int64
```

[179]:
```
# Countplot of heart disease patients by age
```

```
plt.figure (figsize=(5,4))

sns.countplot (x = 'heart_disease', hue = 'gender', data = hrt_dis, palette =␣
↪'Set3_r')
plt.title ("Heart disease by gender")
plt.xlabel ("Patient status")
plt.ylabel ("number of patients")
```

[179]: Text(0, 0.5, 'number of patients')



Note:

–> Comapring to women, the rate of Heart disease in men ARE HIGHER

8. Finding the Relatioship between Age and Chest pain

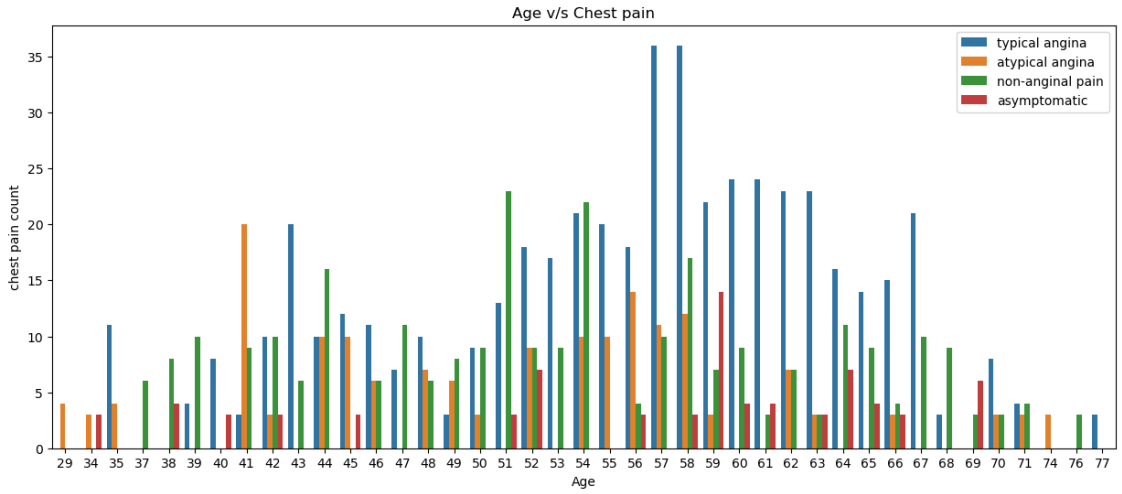
[180]:
```
# calculating the number of patients belongs to different types of chest pains

hrt_dis.groupby(by= 'chest_pain')['age'].count()
```

[180]:
```
chest_pain
0    497
1    167
2    284
```

```
3     77
Name: age, dtype: int64
```

[181]:
```
# using coutplot finding the how common different types of chest pain is in␣
↪people


plt.figure(figsize =(15,6))
sns.countplot(hrt_dis, x='age', hue='chest_pain')
plt.title ("Age v/s Chest pain")
plt.xlabel ("Age")
plt.ylabel ("chest pain count")
plt.legend (labels=['typical angina','atypical angina','non-anginal␣
↪pain','asymptomatic'])
```

[181]:
```
<matplotlib.legend.Legend at 0x25a9defc400>
```



Note:

–> Typical angina and non anginal pain are the most common chest pain types diagnosed among people

–> Typical Angina is found in persons of 56-58 years old

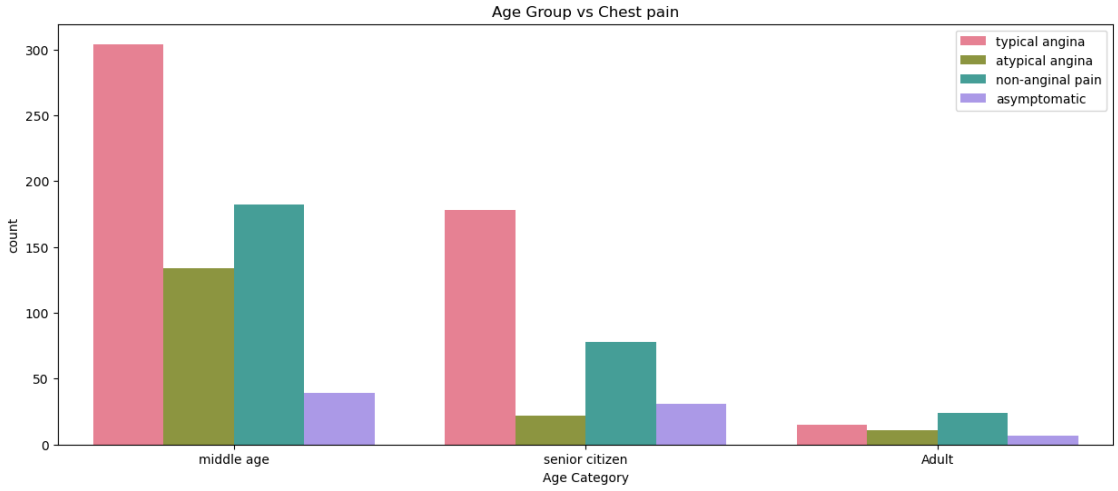
9. Chance of Chest pain occurence grouped by Age category

[182]:
```
# chest pain appearence based on age category

hrt_dis.groupby(by= ['age_group','chest_pain'])['chest_pain'].count()
```

[182]:
```
age_group       chest_pain
Adult           0              15
```

```
                  1              11
                  2              24
                  3               7
middle age        0             304
                  1             134
                  2             182
                  3              39
senior citizen    0             178
                  1              22
                  2              78
                  3              31
Name: chest_pain, dtype: int64
```

[183]:
```
plt.figure(figsize =(15,6))
sns.countplot(hrt_dis, x='age_group', hue='chest_pain', palette = 'husl')
plt.title ("Age Group vs Chest pain")
plt.xlabel ("Age Category")
plt.legend (labels=['typical angina','atypical angina','non-anginal␣
 ↪pain','asymptomatic'])

plt.show()
```



Note:

–> Middle-aged and Senior citizens suffers due to Typical Angina most.

–> Among Aduts, The rate of Non-Anginal pain is found higher than other types

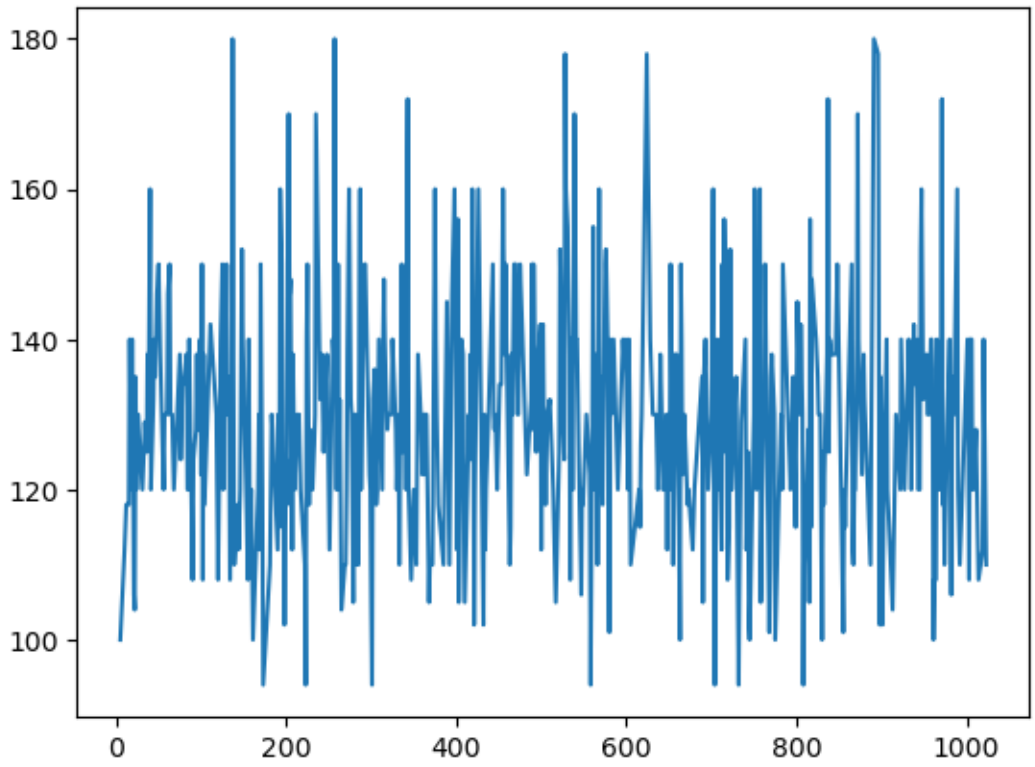
10. Comparing Blood Pressure and Cholesterol for Heart Disease Patients

[184]:
```
# creating a filter for Resting Blood Pressure rates of Heart patients and␣
 ↪storing in bp
```

```
bp = hrt_dis['rest_BPs'][(hrt_dis['heart_disease'] == 'Present')]

# creating a filter for Cholesterol rates of Heart patients and storing in ch

ch = hrt_dis['chol'][(hrt_dis['heart_disease'] == 'Present')]
```

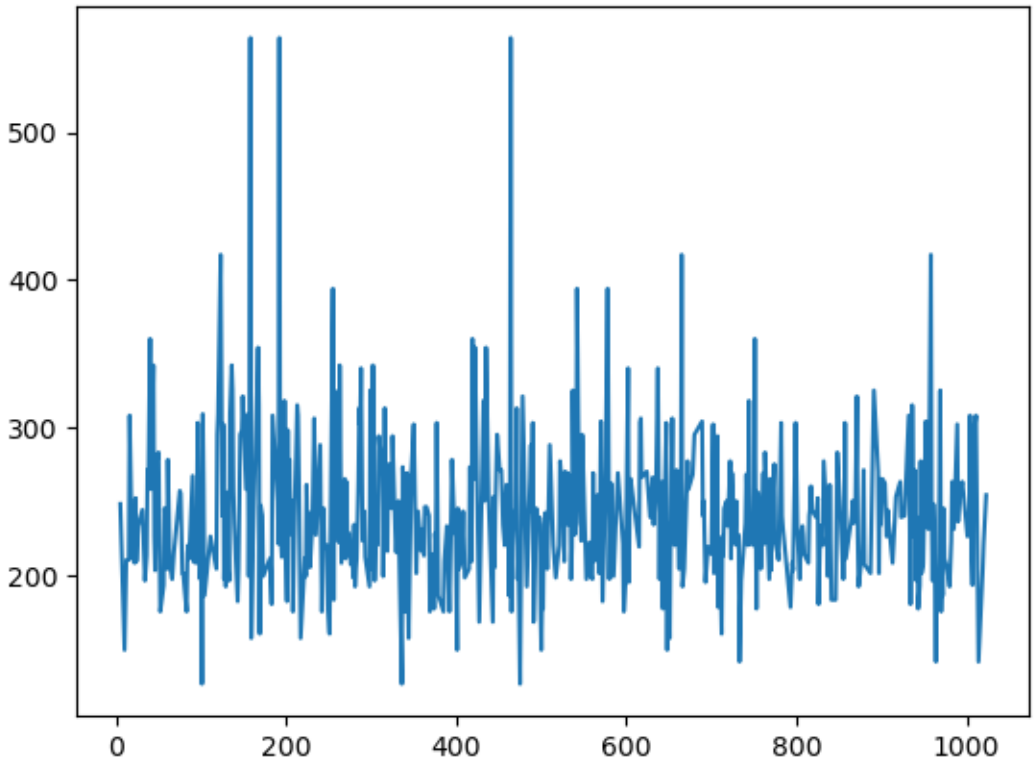
[185]:
```
# line plot of series bp

plt.plot (bp)
```

[185]: [<matplotlib.lines.Line2D at 0x25a9e7e3340>]



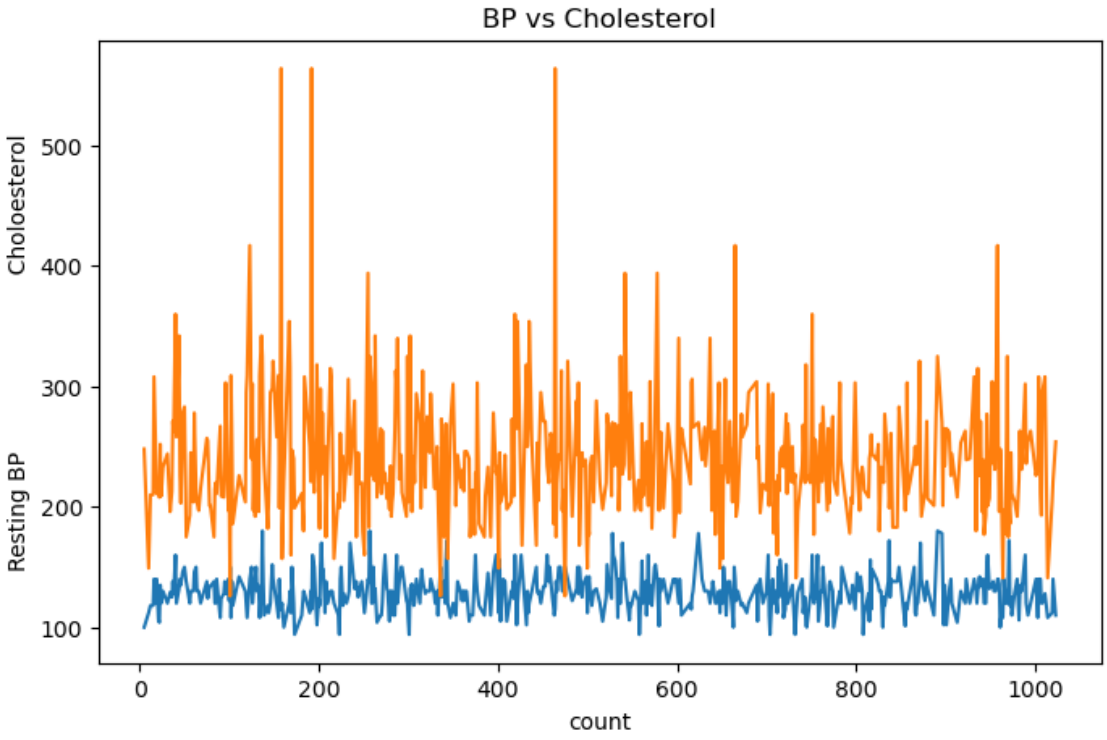
[186]:
```
# line plot of series ch

plt.plot (ch)
```

[186]: [<matplotlib.lines.Line2D at 0x25a9e860490>]



[187]:
```
# Combined linechart of bp and ch

plt.figure(figsize=(8,5))

plt.plot (bp)
plt.plot (ch)
plt.title ("BP vs Cholesterol")
plt.xlabel ("count")
plt.ylabel ("Resting BP                          Choloesterol")
```

[187]:
```
Text(0, 0.5, 'Resting BP                          Choloesterol')
```



Note:

–> There is a direct relationship between Chelosterol rate and Blood pressure rate in Heart disease patients

[188]:
```
hrt_dis.groupby(by=['chol_range'])['heart_disease'].count()
```

[188]:
```
chol_range
Borderline high    339
High               503
Normal             169
Name: heart_disease, dtype: int64
```

11. Percentage of Diabetic patients

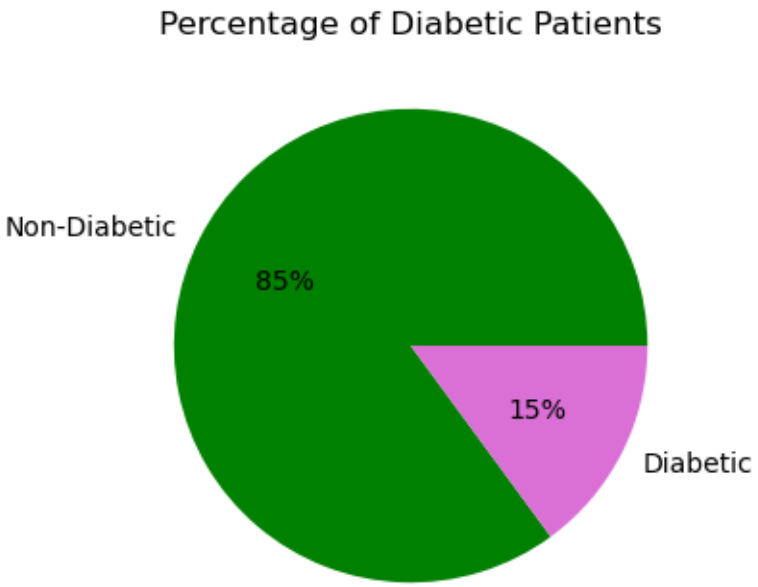
[189]:
```
a = hrt_dis['FBS'].value_counts()
a
```

[189]:
```
0    872
1    153
Name: FBS, dtype: int64
```

[190]:
```
# pie chart of age population in (%)
```

```
plt.figure(figsize=(5,4))

plt.pie(a,labels = ['Non-Diabetic','Diabetic'], autopct = '%0.0f%%',␣
 ↪colors=['green','orchid'])
plt.title ("Percentage of Diabetic Patients")
```

[190]:
```
Text(0.5, 1.0, 'Percentage of Diabetic Patients')
```



Note:

–> Diabetic patients are only 15% of the population

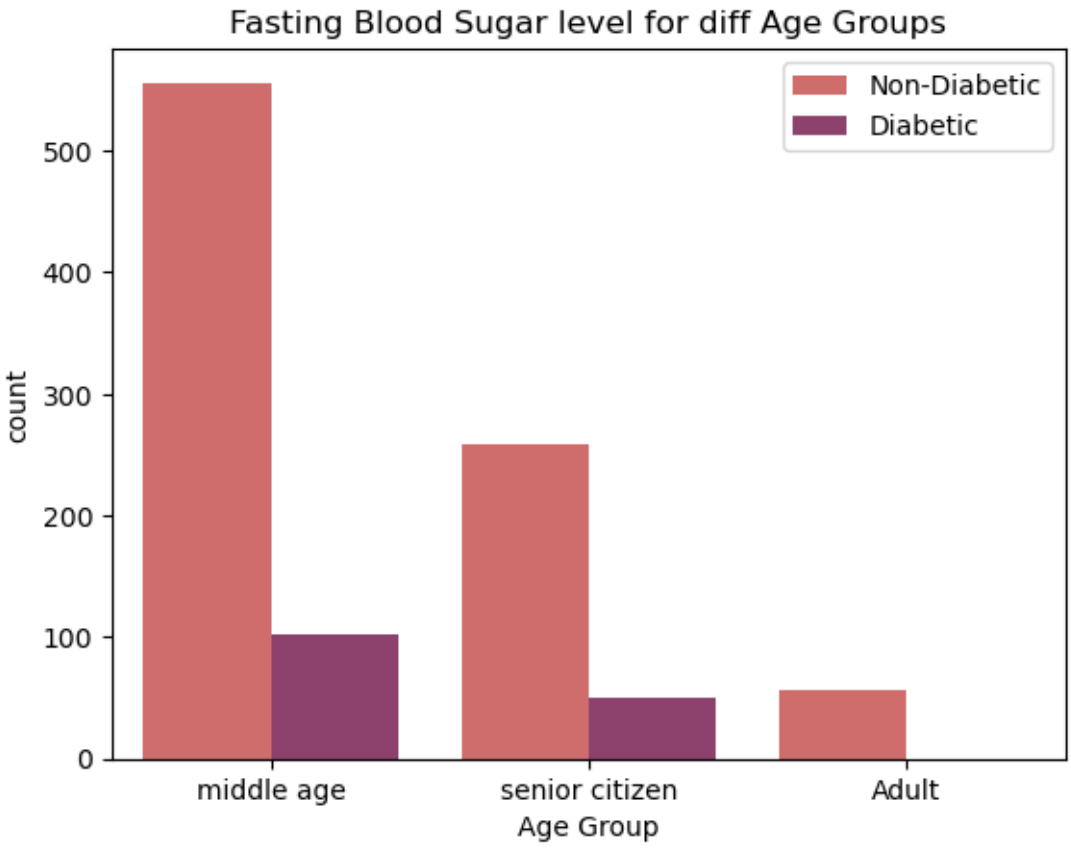
12. Blood sugar level for different age groups

[191]:
```
# Blood sugar level for different age group

a= hrt_dis.groupby (by =['age_group','FBS'])['FBS'].count()
a
```

[191]:
```
age_group       FBS
Adult           0       57
middle age      0      556
                1      103
senior citizen  0      259
                1       50
Name: FBS, dtype: int64
```

[192]:
```
# Countplot visual for sugar level

sns.countplot(x = 'age_group', hue = 'FBS', data = hrt_dis, palette = 'flare')
plt.title ("Fasting Blood Sugar level for diff Age Groups")
plt.xlabel ("Age Group")
plt.legend (labels=['Non-Diabetic','Diabetic'])
```

[192]:
```
<matplotlib.legend.Legend at 0x25a9e1f9510>
```



Note:

–> In Adults FBS level is normal,ie, they are Non-diabetc

–> comparing to senior citizens Middle aged suffer from diabetics

13. Blood sugar rate grouped by Gender

[193]:
```
# Blood sugar level for different age group

hrt_dis.groupby (by =['gender','FBS'])['FBS'].count()
```

[193]:
```
gender  FBS
Female  0      270
        1       42
Male    0      602
        1      111
Name: FBS, dtype: int64
```

[194]:
```
# Countplot visual for sugar level

sns.barplot(x = 'gender', y = 'FBS', data = hrt_dis, palette = 'viridis')
plt.title ("gender wise Blood Sugar rate")
plt.xlabel ("Gender")
```

[194]:
```
Text(0.5, 0, 'Gender')
```

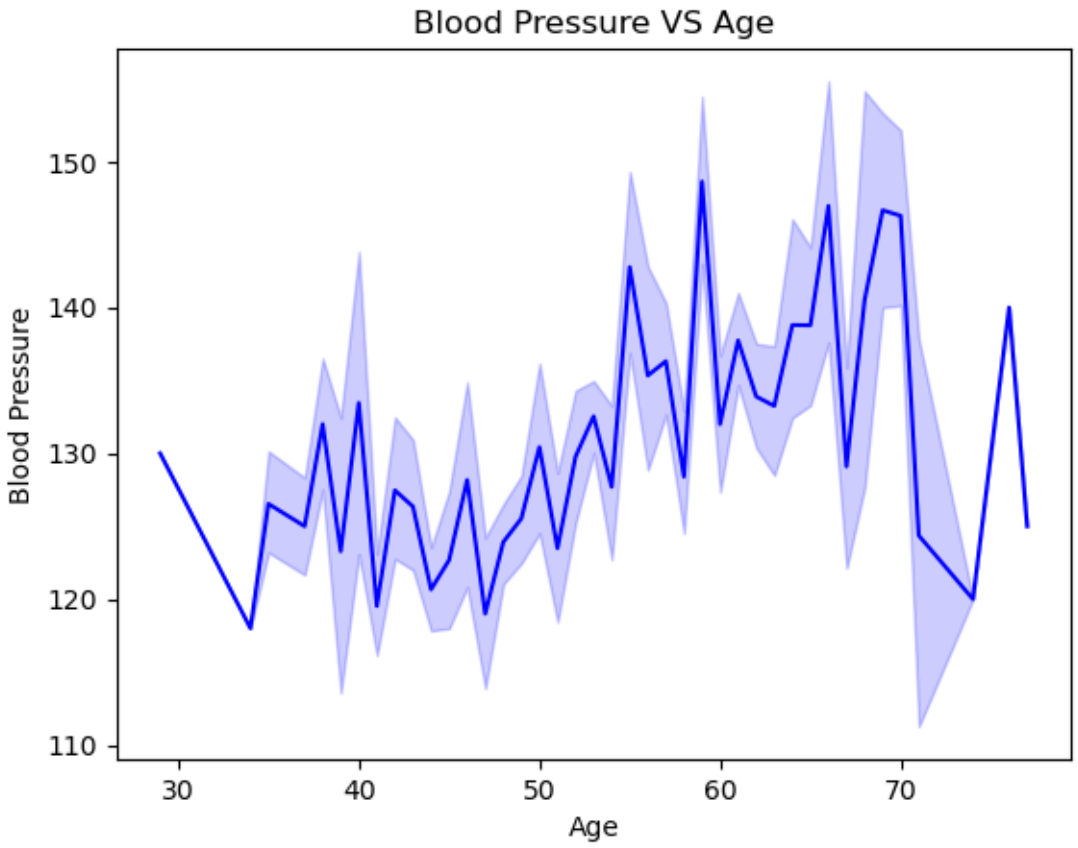
Note:

–> Blood sugar level is higher in men compared to women

14. Relationship of Blood Pressure vs Age

[195]:
```
#Line Plot of Blood Pressure VS Age

sns.lineplot(x='age', y='rest_BPs', data=hrt_dis, color='b')
plt.title('Blood Pressure VS Age')
plt.xlabel('Age')
plt.ylabel('Blood Pressure')
```

[195]:
```
Text(0, 0.5, 'Blood Pressure')
```



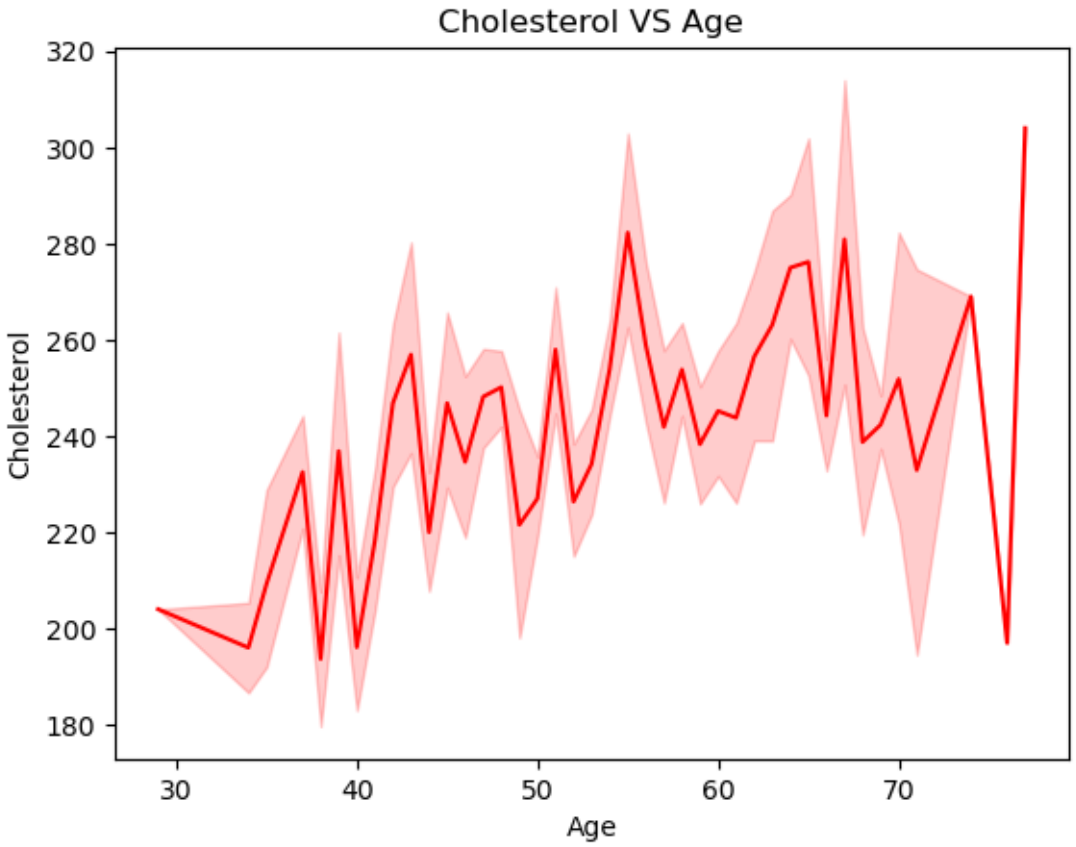
Note:

–> Blood pressure is higher for peple in the middle age, constant increase from the age of 40

15. Relationship of Cholesterol level vs Age

[196]:
```
#Line Plot Creation of Cholestrol VS Age

sns.lineplot(x='age', y='chol', data=data, color='r')
plt.title('Cholesterol VS Age')
plt.xlabel('Age')
plt.ylabel('Cholesterol')
```

[196]: Text(0, 0.5, 'Cholesterol')



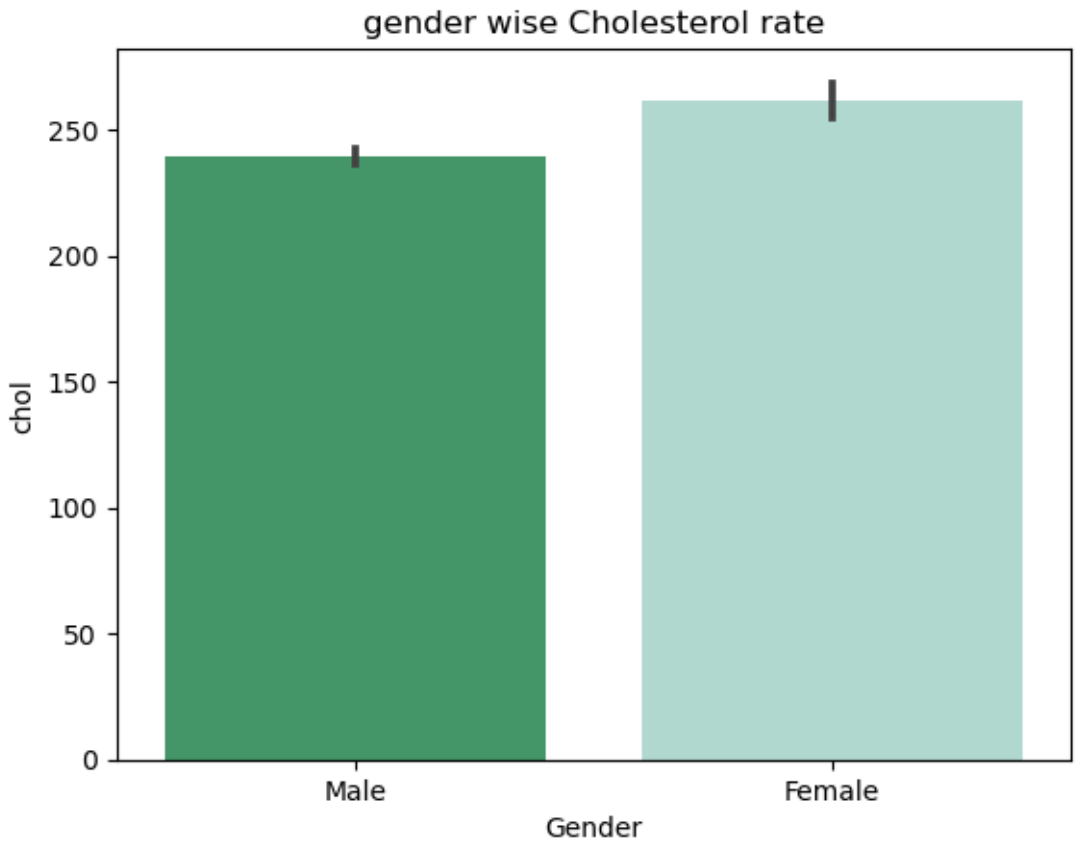
Note:

–> from the age 50 cholesterol level increases

16. Cholesterol rate grouped by Gender

[197]:
```
# Countplot for cholesterol grouped by gender

sns.barplot(x = 'gender', y = 'chol', data = hrt_dis, palette = "BuGn_r")
plt.title ("gender wise Cholesterol rate")
plt.xlabel ("Gender")
```

[197]: Text(0.5, 0, 'Gender')



Note:

–> Cholesterol rate is considerably higher in females

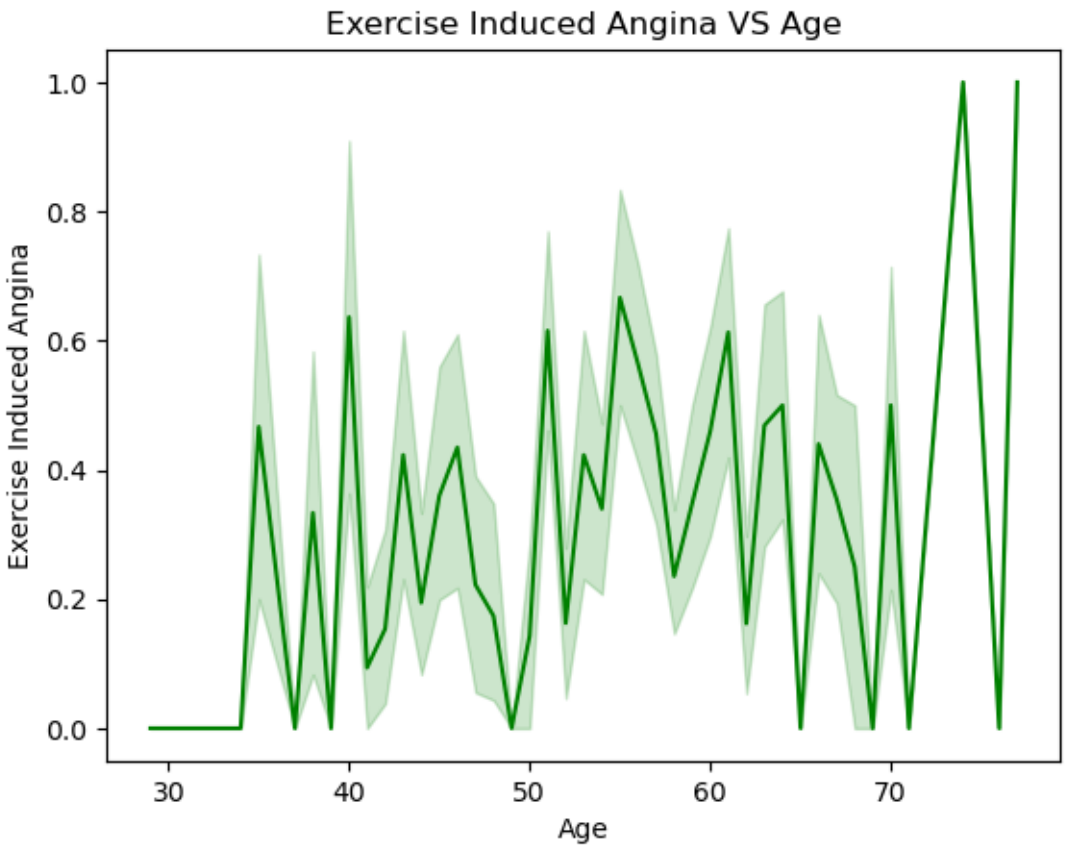
17. Comparing Exercise Induced Angina with Age

[198]:
```
# Calculating the chance of getting exercised induced angina

hrt_dis.groupby(by =['Ex_Ang'])['age'].count()
```

[198]:
```
Ex_Ang
0    680
1    345
Name: age, dtype: int64
```

[199]:
```
#Line Plot showing of Exercise induced Angina VS Age

sns.lineplot(x='age', y='Ex_Ang', data=hrt_dis, color='g')
plt.title('Exercise Induced Angina VS Age')
plt.xlabel('Age')
plt.ylabel('Exercise Induced Angina')
```

[199]: Text(0, 0.5, 'Exercise Induced Angina')



Note:

–> Angina is pain in the chest that comes on with exercise, stress, or other things that make the heart work harder.

–> the rate is high in people between 50 to 60 years and also above 70

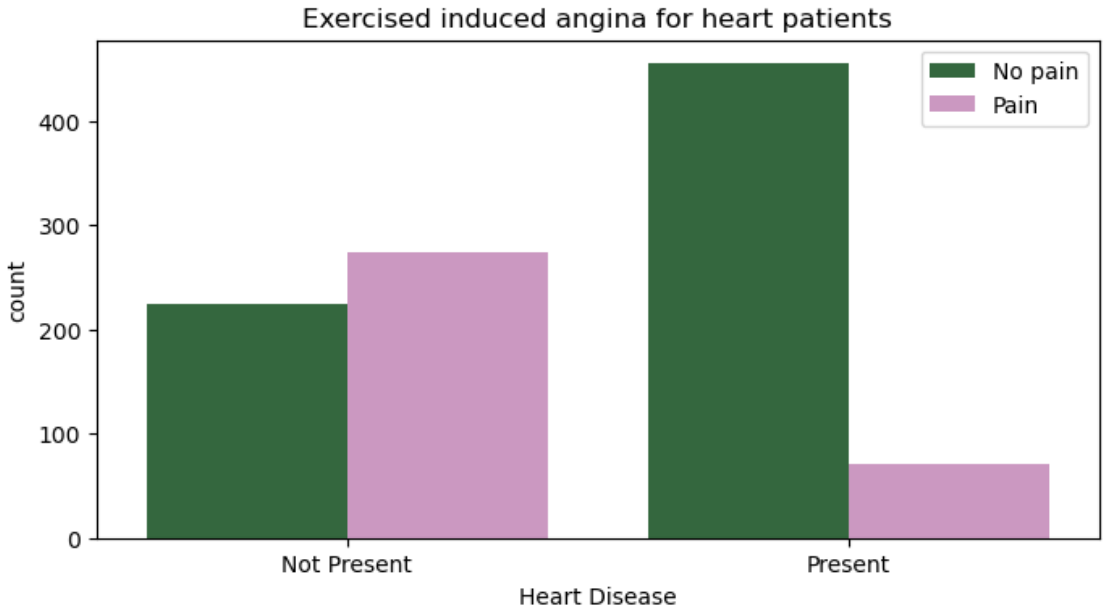
18. Comparing the exercised induced angina with Heart patients

[200]:
```
# Calculating the chance of getting exercised induced angina for heart patients

hrt_dis.groupby(by=['heart_disease','Ex_Ang'])['Ex_Ang'].count()
```

[200]:
```
heart_disease  Ex_Ang
Not Present    0         225
               1         274
Present        0         455
               1          71
Name: Ex_Ang, dtype: int64
```

[201]:
```
# countplot visual for exercised induced angina for heart patients

plt.figure(figsize=(8,4))

sns.countplot(x = 'heart_disease', hue = 'Ex_Ang', data = hrt_dis, palette =␣
↪'cubehelix')
plt.title ("Exercised induced angina for heart patients")
plt.xlabel ("Heart Disease")
plt.legend (labels=['No pain','Pain'])
```

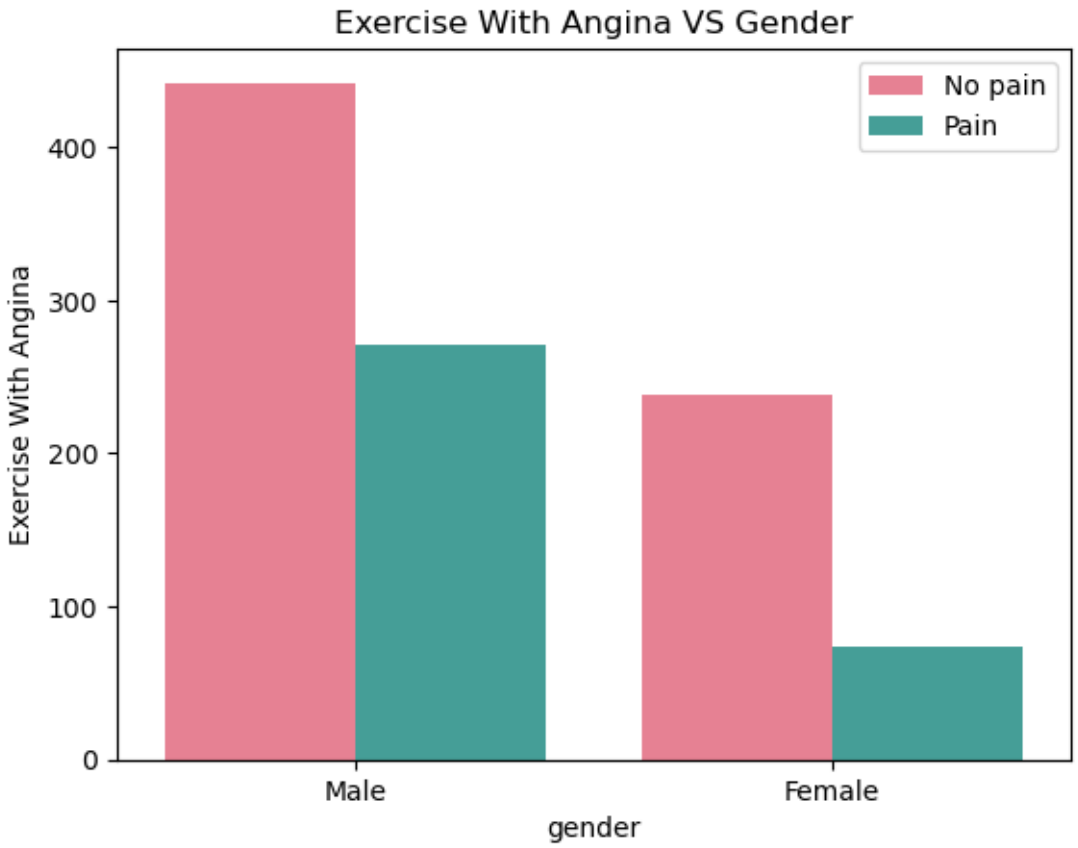
[201]:
```
<matplotlib.legend.Legend at 0x25a9de809d0>
```



19. Exercise Induced Angina with Gender

[202]:
```
#Bar Plot Creation of Exercise With Angina VS Gender

sns.countplot( x = 'gender', hue = 'Ex_Ang', data=hrt_dis, palette = 'husl' )
plt.title('Exercise With Angina VS Gender')
plt.xlabel('gender')
plt.ylabel('Exercise With Angina')
plt.legend (labels=['No pain','Pain'])
```

[202]:
```
<matplotlib.legend.Legend at 0x25a9e4d86a0>
```



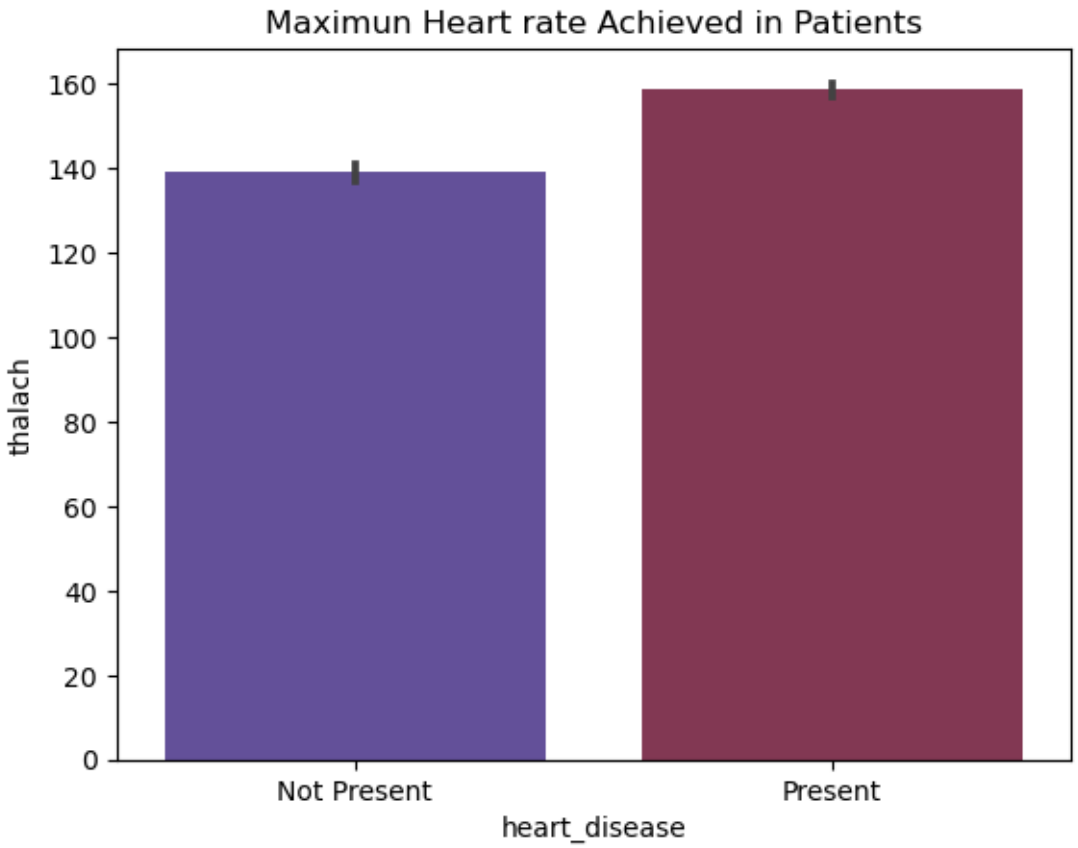
Note:

–>The chance of having paing if you are a male is higher than female

20. Maximum Heart rate Achieved in Patients

[203]:
```
sns.barplot( x= 'heart_disease', y= 'thalach', data= hrt_dis, palette =␣
↪'twilight')
plt.title("Maximun Heart rate Achieved in Patients")
```

[203]:
```
Text(0.5, 1.0, 'Maximun Heart rate Achieved in Patients')
```



Note:

–> thalach or The maximun heart rate achieved in patients is considerably higher in Heart patients

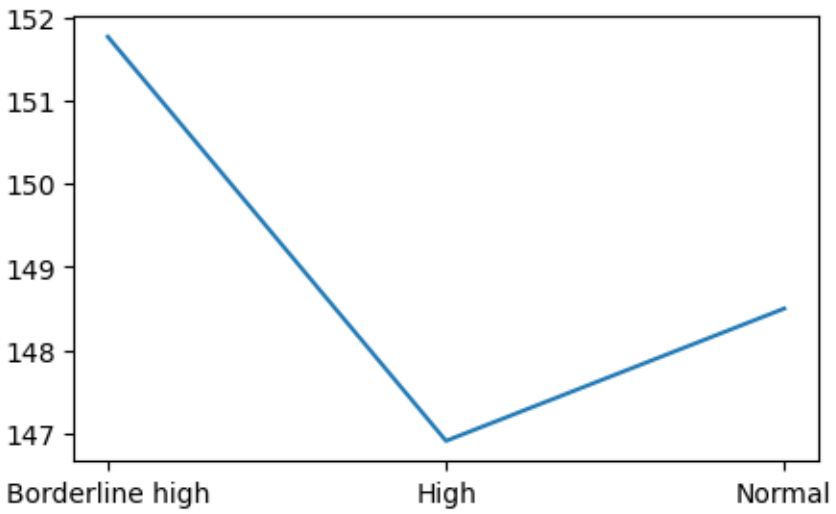
21. Average thalach for cholesterol range

[204]:
```
t= hrt_dis.groupby(by= ['chol_range'])['thalach'].mean()
t
```

[204]:
```
chol_range
Borderline high    151.764012
High               146.906561
Normal             148.497041
Name: thalach, dtype: float64
```

[205]:
```
plt.figure(figsize=(5,3))
plt.plot (t)
```

[205]:
```
[<matplotlib.lines.Line2D at 0x25a9ff58130>]
```



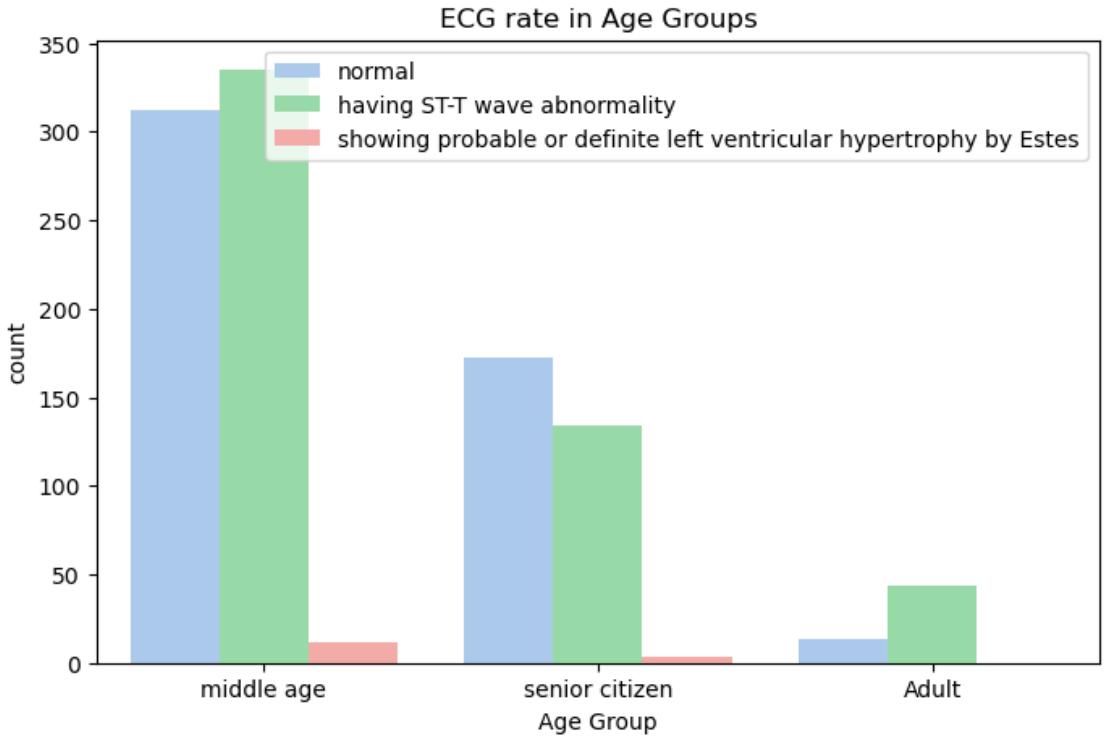
Note:

–> category ‘High’ is having a low thalach average (146.90)

22. ECG rate examined in different Age groups

[206]:
```
# Countplot visual for ECG

plt.figure(figsize=(8,5))
sns.countplot(x = 'age_group', hue = 'restecg', data = hrt_dis,␣
 ↪palette='pastel6')
plt.title ("ECG rate in Age Groups")
plt.xlabel ("Age Group")
plt.legend (labels=['normal','having ST-T wave abnormality','showing probable␣
 ↪or definite left ventricular hypertrophy by Estes'])
```

[206]:
```
<matplotlib.legend.Legend at 0x25a9e2c5180>
```



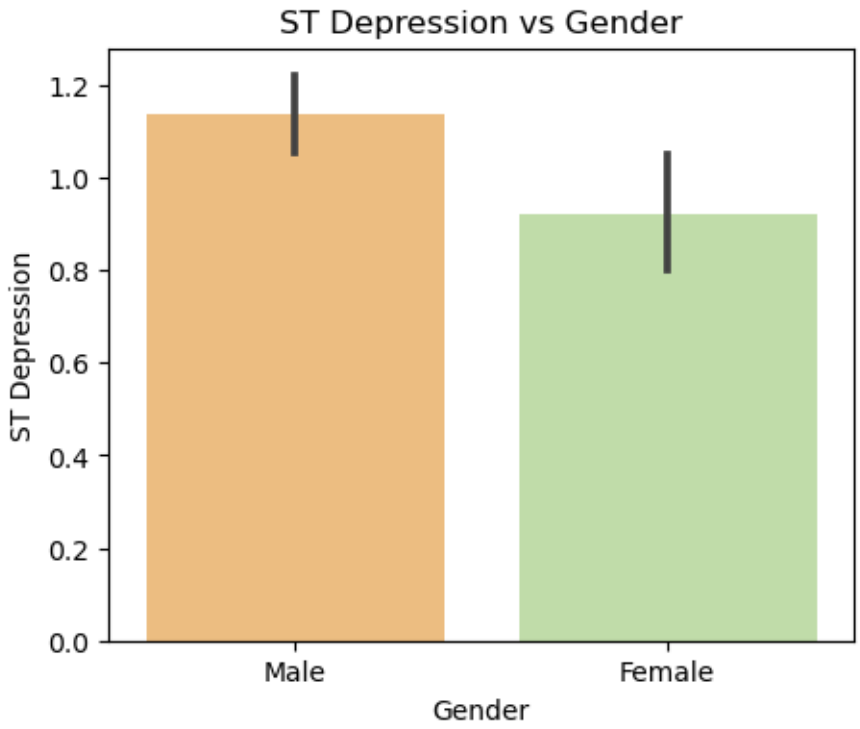
Note:

–>Middle aged people having the high ECG rate having ST-T wave abnormality

23. Relationship of ST Depression vs Gender

[207]:
```
# Bar Plot Creation of ST Depression vs Gender

plt.figure(figsize=(5,4))
sns.barplot(x='gender', y='old_peak', data=hrt_dis, palette ='Spectral')
plt.title('ST Depression vs Gender')
plt.xlabel('Gender')
plt.ylabel('ST Depression')
```

[207]:
```
Text(0, 0.5, 'ST Depression')
```



Note:

–> More Males are diagnosed with ST depression as compare to females

Final Impressions:

```
* There are a total of 1025 record and 14 features in the dataset
* Minimum age of patient in the population is 29 and maximum is 77
* The average rates of features calculated as;
        Blood pressure - 131.61
        Cholesterol    - 246
        Blood Sugar    - 0.149
        ECG            - 0.53
* Total of 713 patients (70%) are men, while 312 are women(30%)
* Out of the population, 526 patients are suffering from heart diseases
* 64% of patients belongs to middle aged, 30% senior citizen and 6% Adults
* 85% of population are non diabetic
* out of total 602 men are Diabetic
* Diabetics in adults are normal in rate
* out of total, 503 patients are suffering from High cholesterol
* 50% of population falls under the category 'High', which means having a rate of more than 24
* The risk of having ST-T wave abnormality is high in patients of 40-60 age
* 363 patients suffers from heart disease belong to middle aged
* The chance of having heart disease is high in men
* Middle-aged and Senior citizens suffers due to Typical Angina most.
```

* Among Adults, The rate of Non-Anginal pain is found higher than other types
* More Males are diagnosed with ST depression as compared to females
* Cholesterol and BP is high in heart disease patients
* After the age of 50, there is a considerable hike in cholesterol rate
* High cholesterol is diagnosed in women compared to men
* more than 400 men has a risk of Exercise induced Angina
* category 'High' is having a low thalach average (146.90)
* Thalach or The maximum heart rate achieved in patients is considerably higher in Heart patie
* Thalach minimum in population is 71 and max is 202