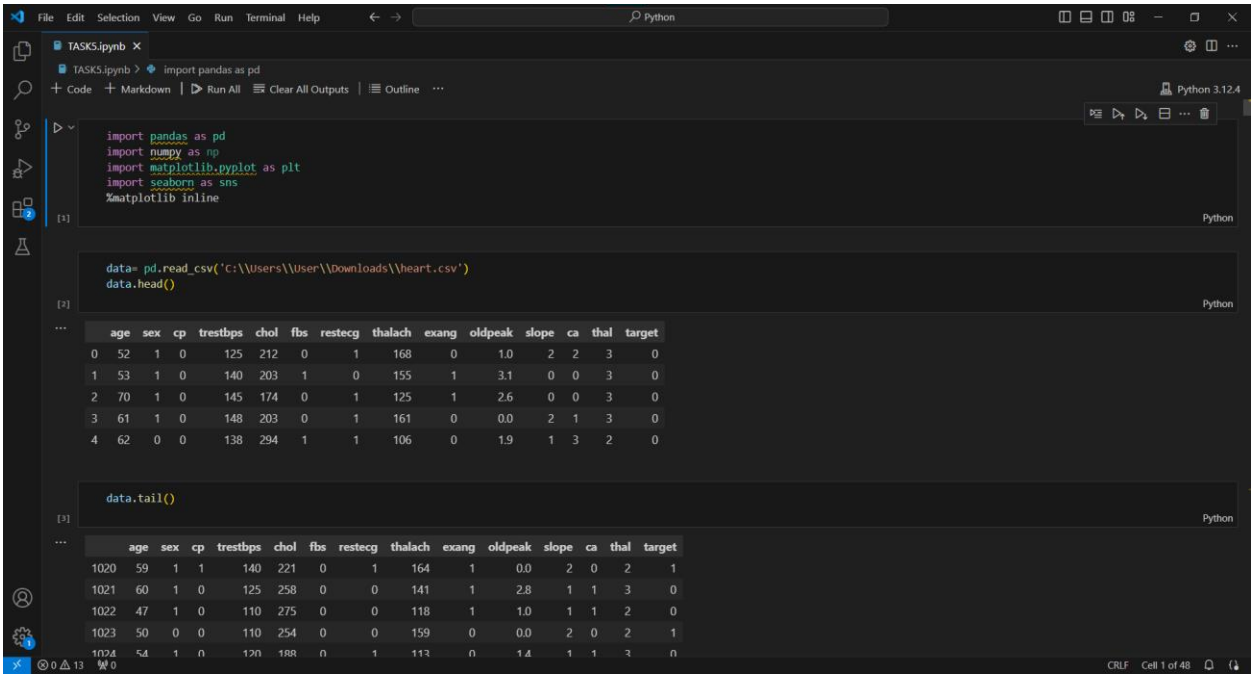
TASK 5



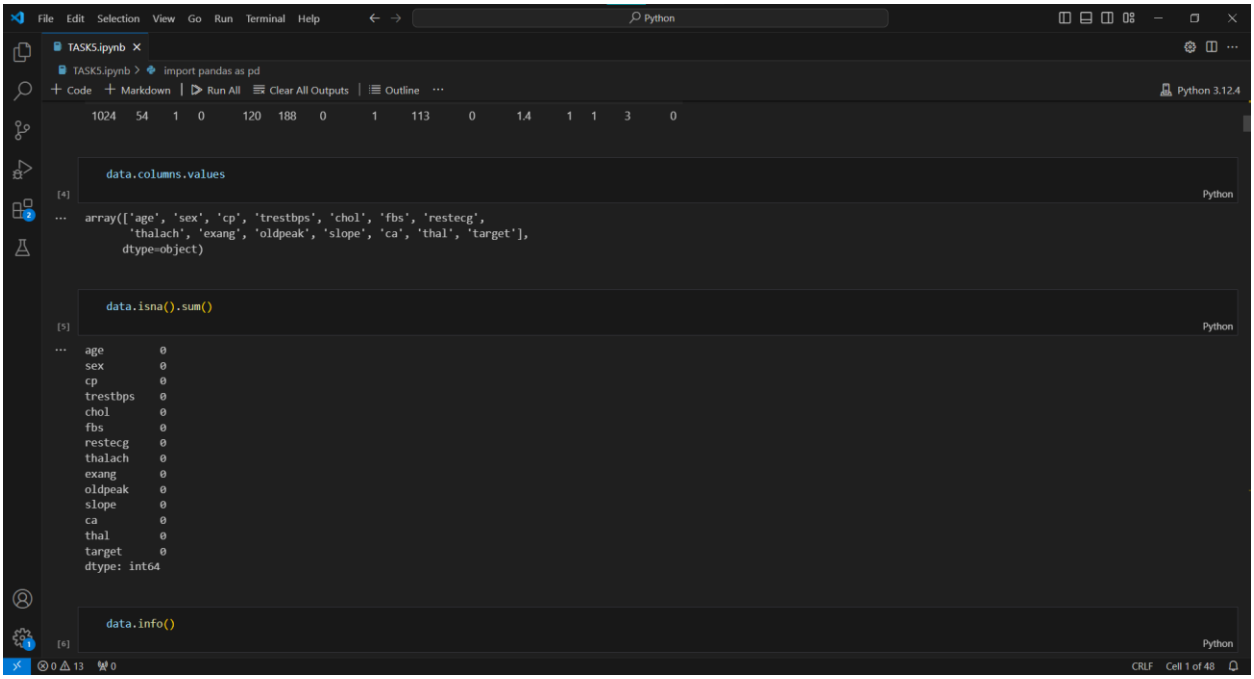
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
data= pd.read_csv('C:\\Users\\User\\Downloads\\heart.csv')
data.head()
```

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

```
data.tail()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0



```
data.columns.values
```

```
array(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg',
       'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype=object)
```

```
data.isna().sum()
```

```
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
```

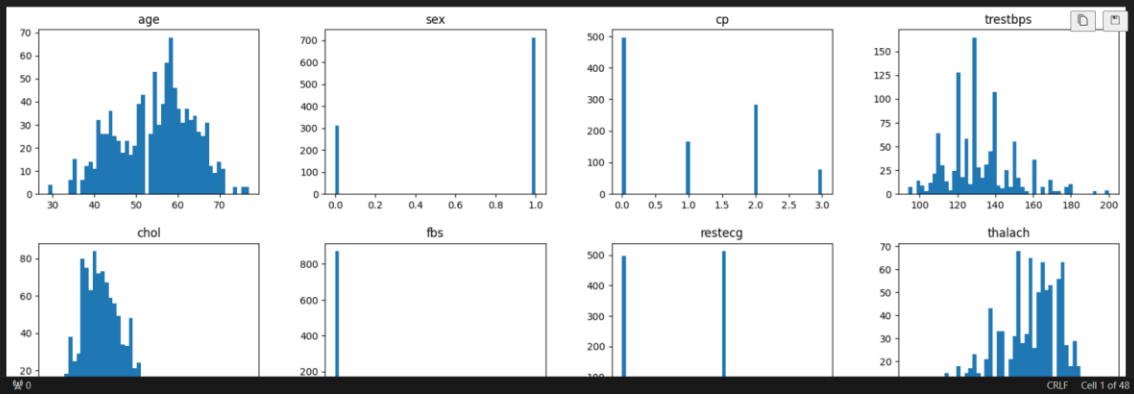
```
data.info()
```

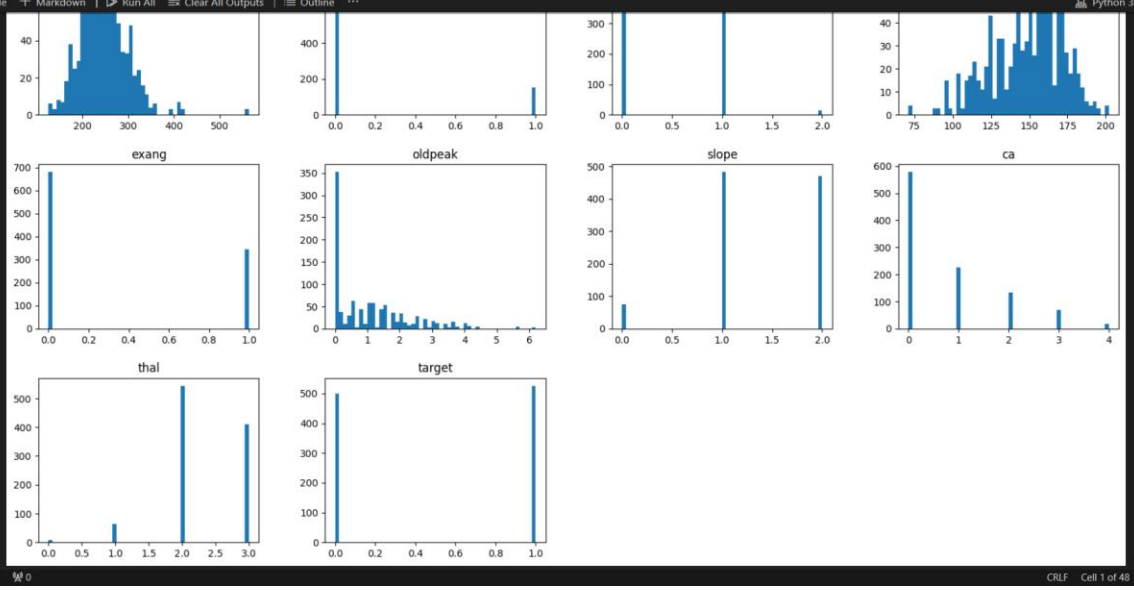
```
data.info()
```

```
<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
```

```
#Plotting histogram of all numeric values
data.hist(bins=50, grid=False, figsize=(20,15))
```

```
array([[<Axes: title={'center': 'age'}>, <Axes: title={'center': 'sex'}>,
        <Axes: title={'center': 'cp'}>,
        <Axes: title={'center': 'trestbps'}>],
       [<Axes: title={'center': 'chol'}>,
        <Axes: title={'center': 'fbs'}>,
        <Axes: title={'center': 'restecg'}>,
        <Axes: title={'center': 'thalach'}>],
       [<Axes: title={'center': 'exang'}>,
        <Axes: title={'center': 'oldpeak'}>,
        <Axes: title={'center': 'slope'}>,
        <Axes: title={'center': 'ca'}>],
       [<Axes: title={'center': 'thal'}>,
        <Axes: title={'center': 'target'}>, <Axes: >, <Axes: >]],
      dtype=object)
```





```
data.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.754146	2.323902	0.513171
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.030798	0.620660	0.500070
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

```
questions= ["1. How many people have heart disease and how many people doesn't have heart disease?",
            "2. People of which sex has most heart disease?",
            "3. People of which sex has which type of chest pain more?",
            "4. People with which chest pain are most prone to have heart disease?",
            "5. What is the average cholesterol level for patients with heart disease versus those without?",
            "6. How does the number of major vessels (ca) affect the likelihood of heart disease?",
            "7. How does the slope of the peak exercise ST segment (slope) vary with age and cholesterol levels?",
            "8. Is there a linear relationship between age and resting blood pressure (`trestbps`)?",
            "9. What is the distribution of serum cholesterol levels (chol) in patients? Are there any outliers?",
            "10. How does the resting blood pressure (trestbps) vary between males and females?"]
questions
```

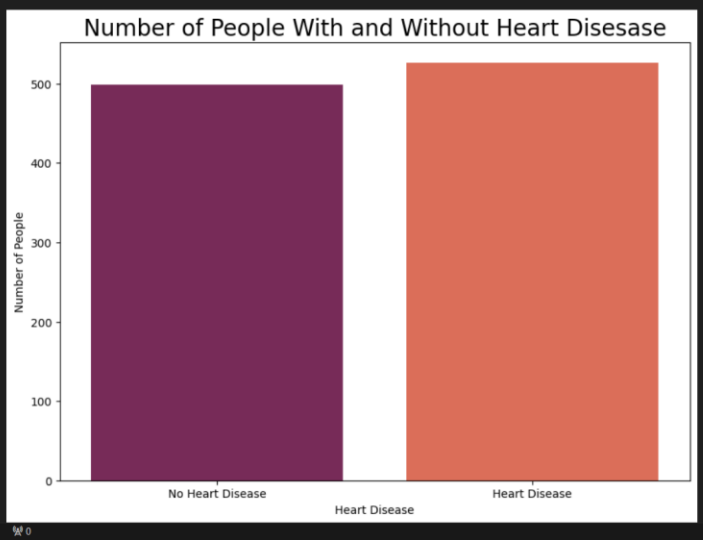
```
["1. How many people have heart disease and how many people doesn't have heart disease?",
 '2. People of which sex has most heart disease?',
 '3. People of which sex has which type of chest pain more?',
 '4. People with which chest pain are most prone to have heart disease?',
 '5. What is the average cholesterol level for patients with heart disease versus those without?',
 '6. How does the number of major vessels (ca) affect the likelihood of heart disease?',
 '7. How does the slope of the peak exercise ST segment (slope) vary with age and cholesterol levels?',
 '8. Is there a linear relationship between age and resting blood pressure (`trestbps`)?',
 '9. What is the distribution of serum cholesterol levels (chol) in patients? Are there any outliers?',
 '10. How does the resting blood pressure (trestbps) vary between males and females?']
```

```
#Let's find the answer for first question

# 1. How many people have heart disease and how many people doesn't have heart disease?

#Getting the values
data.target.value_counts()
```

```
target
1    526
0    499
Name: count, dtype: int64
```

```
#Plotting bar chart

plt.figure(figsize=(10, 7))
sns.barplot(x=data.target.value_counts().index, y=data.target.value_counts().values, palette="rocket")
plt.title("Number of People With and Without Heart Disease", fontsize=20)
plt.xlabel("Heart Disease")
plt.xticks([0,1], ["No Heart Disease", "Heart Disease"])
plt.ylabel("Number of People")
plt.show()
```

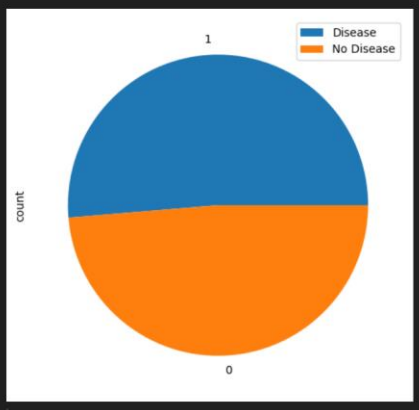
```
sns.barplot(x=data.target.value_counts().index, y=data.target.value_counts().values, palette="rocket")
```



```
#Plotting a pie chart for the same
data.target.value_counts().plot(kind='pie', figsize=(8,6))
plt.legend(["Disease", "No Disease"])
```

[12] Python

```
<matplotlib.legend.Legend at 0x1fffcbdc770>
```



1. Percentage of people with heart disease is slightly more.

```
# 0 represents Female
# 1 represents Male
# Sex column part
# 0 represents 'No Disease'
# 1 represents 'Disease'
# Target Column Part
# Now let's check how many male and female are in the dataset

data.sex.value_counts()
```

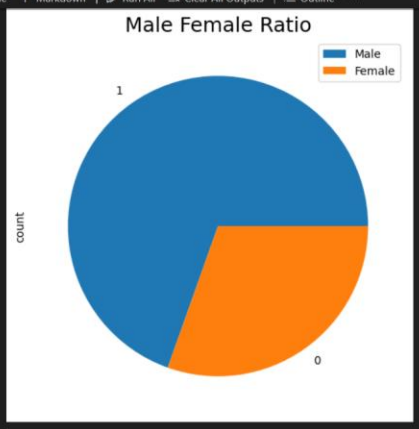
[13] Python

```
sex
1    713
0    312
Name: count, dtype: int64
```

```
#Plotting a pie chart for the same
data.sex.value_counts().plot(kind='pie', figsize=(8,6))
plt.legend(["Male", "Female"])
plt.title("Male Female Ratio", fontsize=18)
```

[14] Python

```
Text(0.5, 1.0, 'Male Female Ratio')
```




```
# Let's find the answer for the 2nd question

# 2. People of which sex has most heart disease?

pd.crosstab(data.target, data.sex)
```

```
pd.crosstab(data.target, data.sex)
```

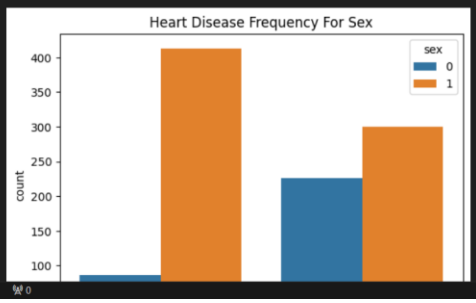
[15] Python

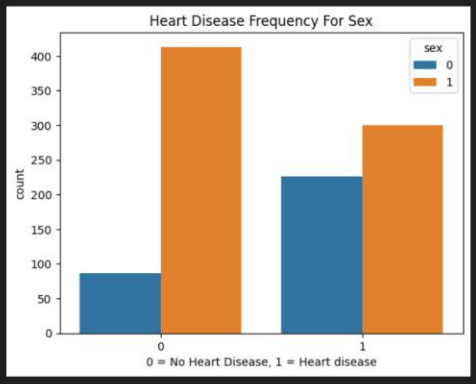
target	sex 0	1
0	86	413
1	226	300

```
sns.countplot(x='target', data=data, hue='sex')
plt.title("Heart Disease Frequency For Sex")
plt.xlabel("0 = No Heart Disease, 1 = Heart disease")
```

[16] Python

```
Text(0.5, 0, '0 = No Heart Disease, 1 = Heart disease')
```





1. Number of Male is more than double in our dataset than female.
2. More than '45% male' has heart disease and '75% male' has heart disease.

```
# Let's move to question 3
```

```
# Let's move to question 3

# 3. People of which sex has which type of chest pain more?

#Counting values for different chest pain

data.cp.value_counts()
```

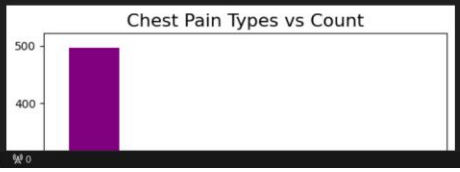
[17] Python

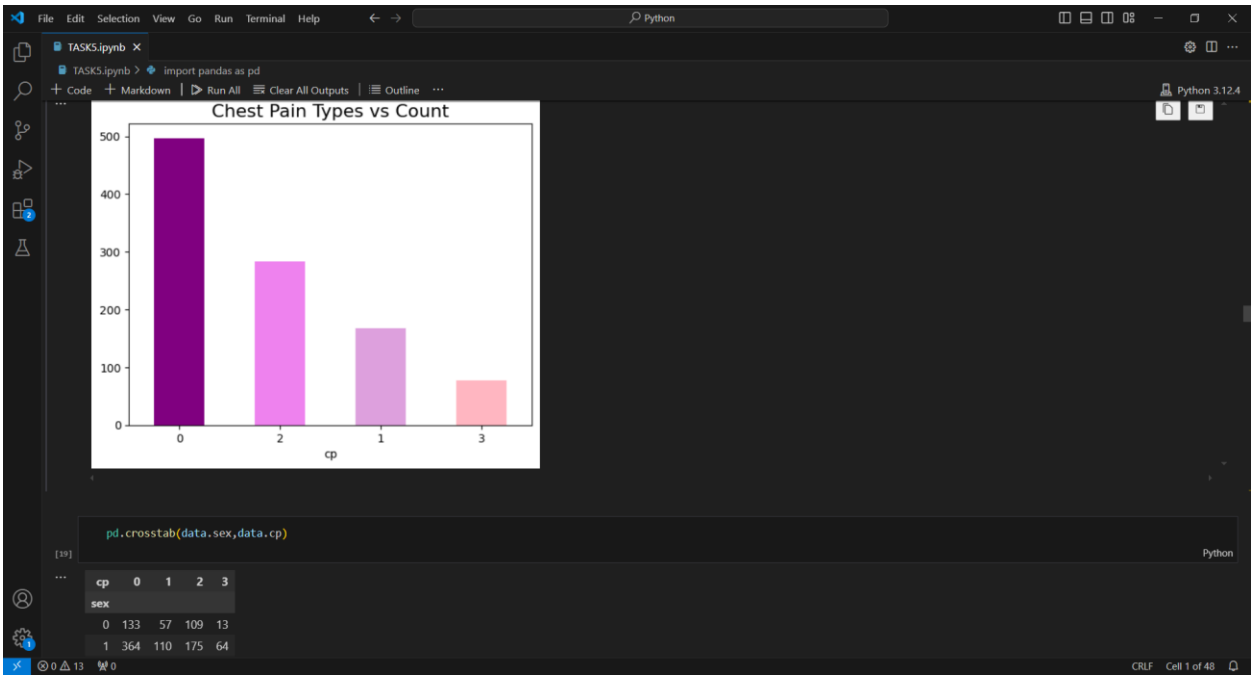
```
cp
0    497
2    284
1    167
3     77
Name: count, dtype: int64
```

```
# Plotting a bar chart
data.cp.value_counts().plot(kind="bar", color=['purple','violet','plum','lightpink'])
plt.title("Chest Pain Types vs Count", fontsize=16)
plt.xticks(rotation=0)
```

[18] Python

```
(array([0, 1, 2, 3]),
 [Text(0, 0, '0'), Text(1, 0, '2'), Text(2, 0, '1'), Text(3, 0, '3')])
```

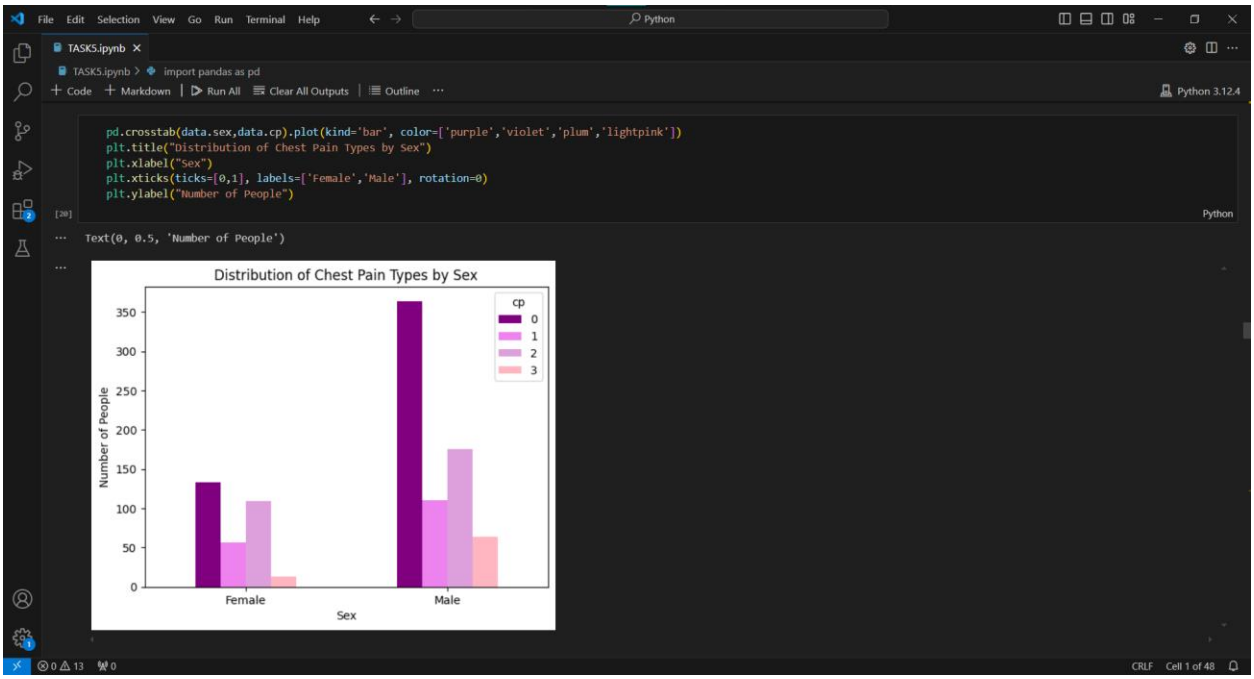





```
pd.crosstab(data.sex,data.cp)
```

[19] Python

sex	cp 0	1	2	3
0	133	57	109	13
1	364	110	175	64




```
pd.crosstab(data.sex,data.cp).plot(kind='bar', color=['purple','violet','plum','lightpink'])
plt.title("Distribution of Chest Pain Types by Sex")
plt.xlabel("Sex")
plt.xticks(ticks=[0,1], labels=['Female','Male'], rotation=0)
plt.ylabel("Number of People")
```

[20] Python

```
Text(0, 0.5, 'Number of People')
```

1. Most of 'Male' has 'Type 0' chest pain and least of 'Type 3' chest pain.
2. In case of 'Female', 'Type 0' and 'Type 2' chest pain's percentage is almost same.

```
# Now question 4

# 4. People with which chest pain are most prone to have heart disease?

pd.crosstab(data.cp,data.target)
```

[21] Python

target	0	1
cp		
0	375	122
1	33	134
2	65	219
3	26	51

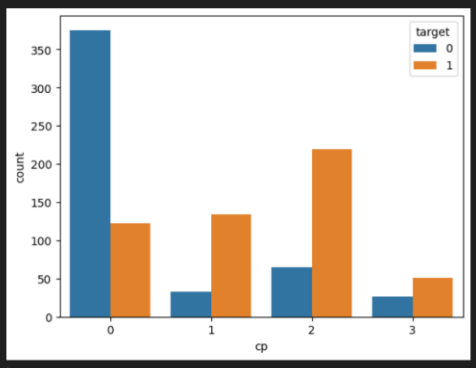
```
sns.countplot(x='cp', data=data, hue='target')
```

[22] Python

```
<Axes: xlabel='cp', ylabel='count'>
```

```
sns.countplot(x='cp', data=data, hue='target')
```

```
<Axes: xlabel='cp', ylabel='count'>
```



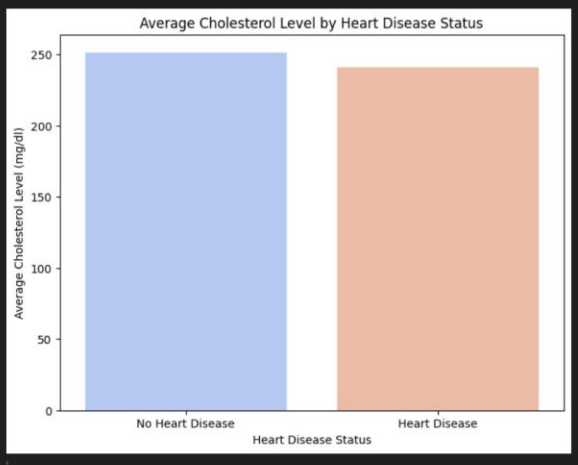
1. Most of the people with 'Type 0' chest pain has less chance of heart disese.
2. As for rest of the chest pain types it is opposite.

```
# Question 5

# 5. What is the average cholesterol level for patients with heart disease versus those without?

# Group by heart disease status and calculate the average cholesterol level
average_cholesterol = data.groupby('target')['chol'].mean()
average_cholesterol
```

```
target
0    251.292585
1    240.979087
Name: chol, dtype: float64
```

```
# Converting groupby result to a DataFrame
average_cholesterol_df = average_cholesterol.reset_index()

average_cholesterol_df['target'] = average_cholesterol_df['target'].map({0: 'No Heart Disease', 1: 'Heart Disease'})

# Plot the barchart
plt.figure(figsize=(8, 6))
sns.barplot(x='target', y='chol', data=average_cholesterol_df, palette='coolwarm')
plt.title("Average Cholesterol Level by Heart Disease Status")
plt.xlabel("Heart Disease Status")
plt.ylabel("Average Cholesterol Level (mg/dl)")
plt.show()
```

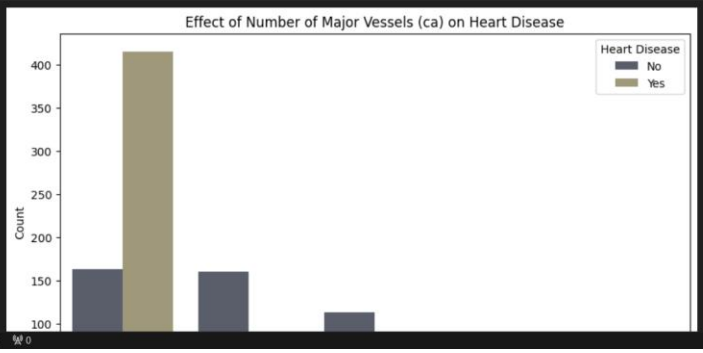
```
C:\Users\User\AppData\Local\Temp\ipykernel_2216\2814331948.py:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='target', y='chol', data=average_cholesterol_df, palette='coolwarm')
```

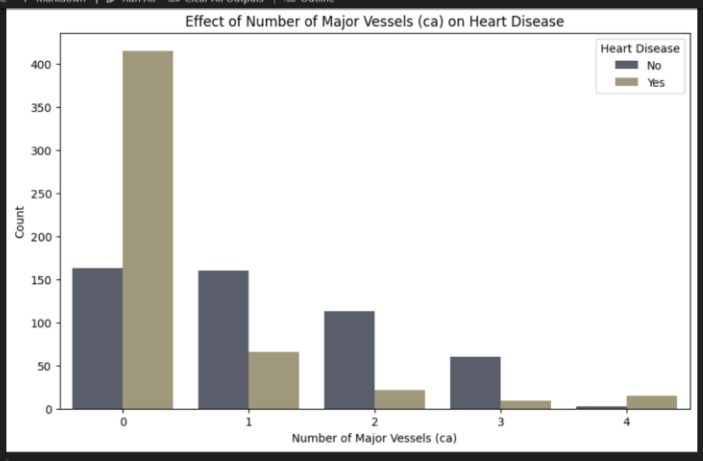


1. Average cholestrol level for patients with no heart disease is more (251.3) than those with heart disease(241).

1. Average cholestrol level for patients with no heart disease is more (251.3) than those with heart disease(241).

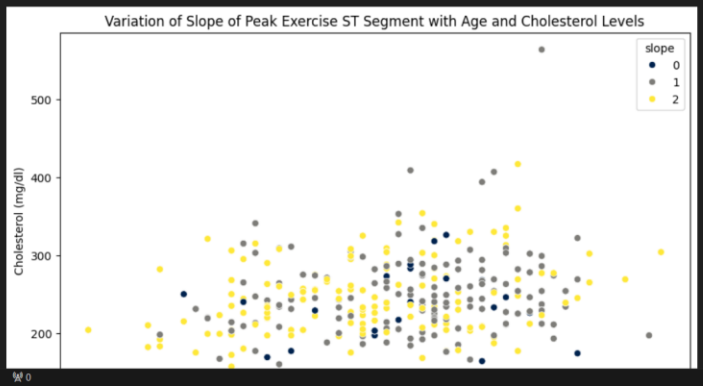
```
# Question 6
# 6. How does the number of major vessels (ca) affect the likelihood of heart disease?
# Create a bar plot
plt.figure(figsize=(10, 6))
sns.countplot(x='ca', hue='target', data=data, palette='cividis')
plt.title('Effect of Number of Major Vessels (ca) on Heart Disease')
plt.xlabel('Number of Major Vessels (ca)')
plt.ylabel('Count')
plt.legend(title='Heart Disease', loc='upper right', labels=['No', 'Yes'])
plt.show()
```

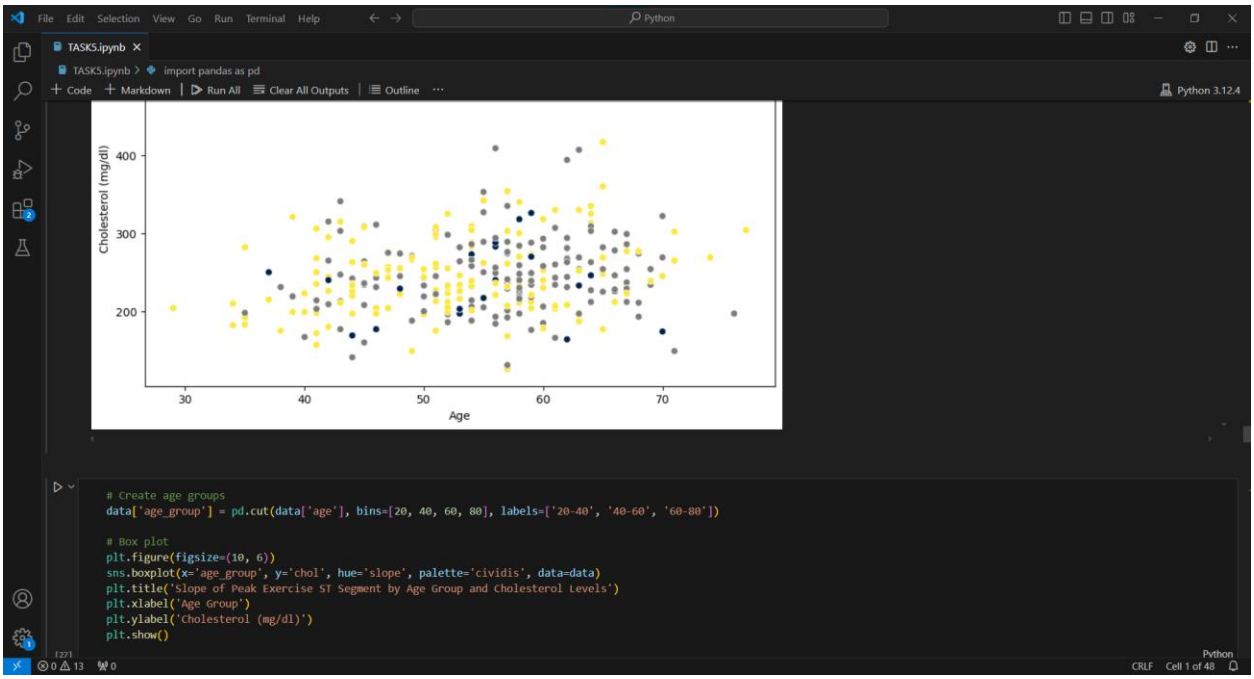




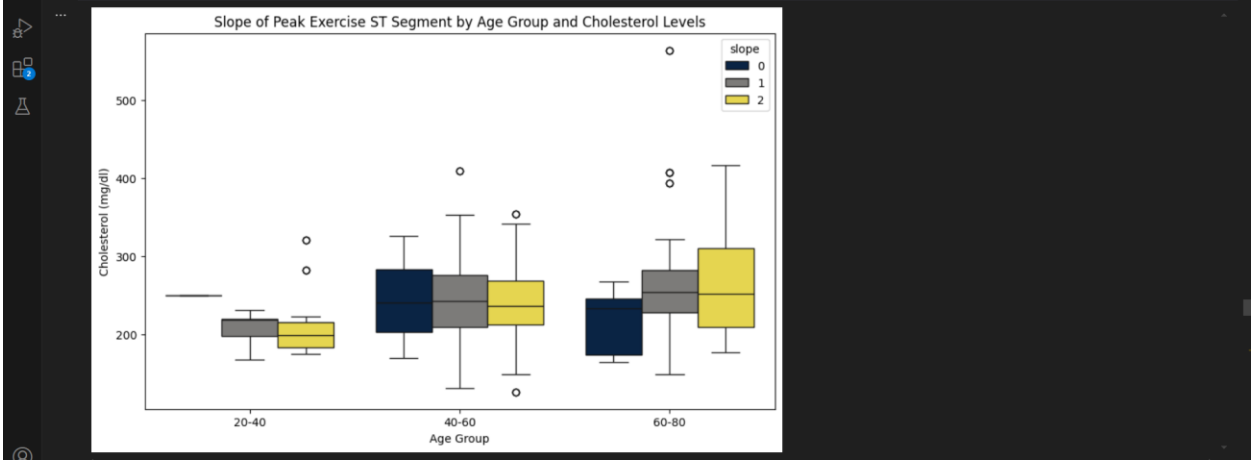
1. The plot shows that number of major vessels (ca) is an important factor in determining heart disease risk.
2. People with fewer major vessels (especially ca=0) are at higher risk of heart disease, whereas those with more major vessels (especially ca=2 or higher) are at a lower risk.

```
# Question 7
# 7. How does the slope of the peak exercise ST segment (slope) vary with age and cholesterol levels?

plt.figure(figsize=(10, 6))
sns.scatterplot(x='age', y='chol', hue='slope', palette='cividis', data=data)
plt.title('Variation of Slope of Peak Exercise ST Segment with Age and Cholesterol Levels')
plt.xlabel('Age')
plt.ylabel('Cholesterol (mg/dl)')
plt.show()
```

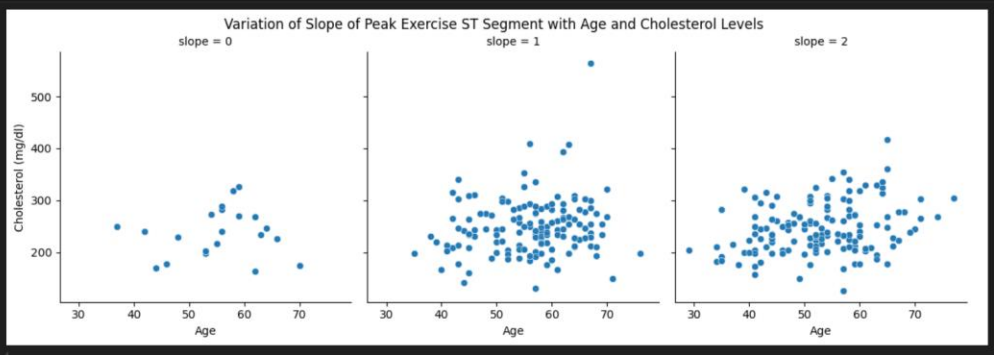





```
# Create age groups
data['age_group'] = pd.cut(data['age'], bins=[20, 40, 60, 80], labels=['20-40', '40-60', '60-80'])

# Box plot
plt.figure(figsize=(10, 6))
sns.boxplot(x='age_group', y='chol', hue='slope', palette='cividis', data=data)
plt.title('Slope of Peak Exercise ST Segment by Age Group and Cholesterol Levels')
plt.xlabel('Age Group')
plt.ylabel('Cholesterol (mg/dl)')
plt.show()
```




```
g = sns.FacetGrid(data, col="slope", height=4, aspect=1)
g.map(sns.scatterplot, "age", "chol", alpha=.7)
g.add_legend()
g.set_axis_labels("Age", "Cholesterol (mg/dl)")
g.fig.suptitle('Variation of Slope of Peak Exercise ST Segment with Age and Cholesterol Levels', y=1.02)
plt.show()
```

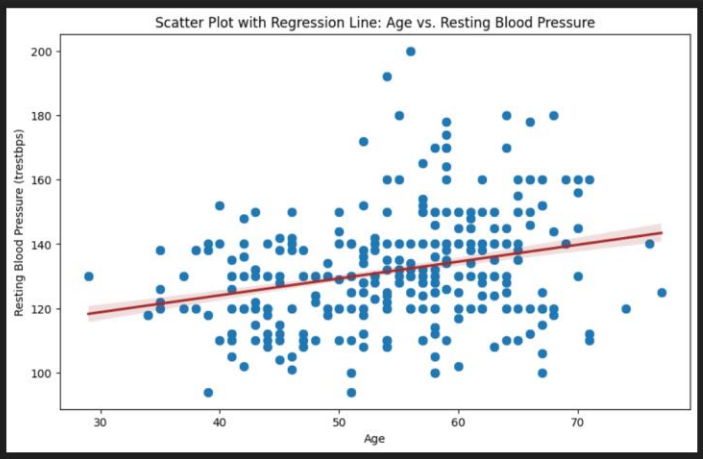


1. Across all slope categories, there doesn't appear to be a strong correlation between age and cholesterol levels. The scatter plots indicate that cholesterol levels vary widely at all ages, regardless of the slope of the ST segment.
2. The plots for slope = 1 and slope = 2 have more data points, indicating that these slope values are more common in the dataset. Both have similar patterns in the distribution of cholesterol levels across different ages.
3. The first plot (slope = 0) has fewer individual and the cholesterol levels are more spread out. This indicate that slope = 0 is less common or associated with different characteristics compared to the other slope categories.

```python
# Question 8
# 8. Is there a linear relationship between age and resting blood pressure (`trestbps`)?

from scipy.stats import pearsonr

# Calculate the Pearson correlation coefficient
correlation, p_value = pearsonr(data['age'], data['trestbps'])

print(f"Pearson correlation coefficient: {correlation}")
print(f"P-value: {p_value}")
```

```
Pearson correlation coefficient: 0.27112140631607595
P-value: 9.96199440464185e-19
```

```python
# Create a scatter plot with a regression line
plt.figure(figsize=(10, 6))
sns.regplot(x='age', y='trestbps', data=data, scatter_kws={'s': 50}, line_kws={'color': 'brown'})
plt.title('Scatter Plot with Regression Line: Age vs. Resting Blood Pressure')
plt.xlabel('Age')
plt.ylabel('Resting Blood Pressure (trestbps)')
plt.show()
```

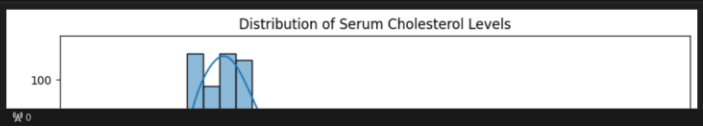


1. A linear relationship between age and resting blood pressure is there in the plot, but it is a weak linear relationship.

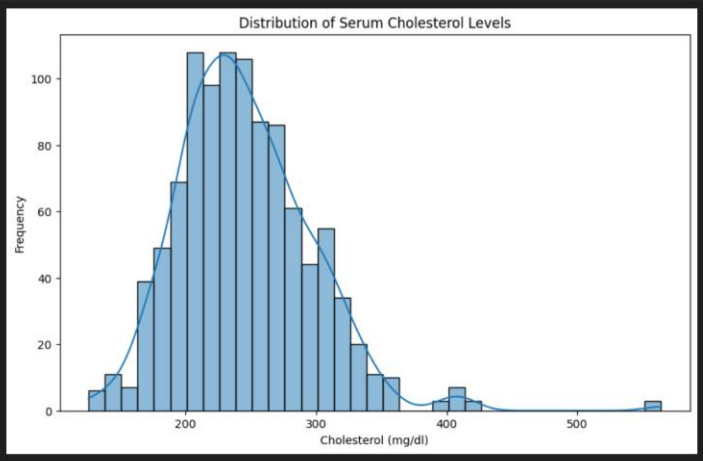
```python
# Question 9
# 9. What is the distribution of serum cholesterol levels (chol) in patients? Are there any outliers?

# Descriptive statistics for cholesterol
print(data['chol'].describe())
```

```
count    1025.00000
mean      246.00000
std        51.59251
min       126.00000
25%       211.00000
50%       240.00000
75%       275.00000
max       564.00000
Name: chol, dtype: float64
```

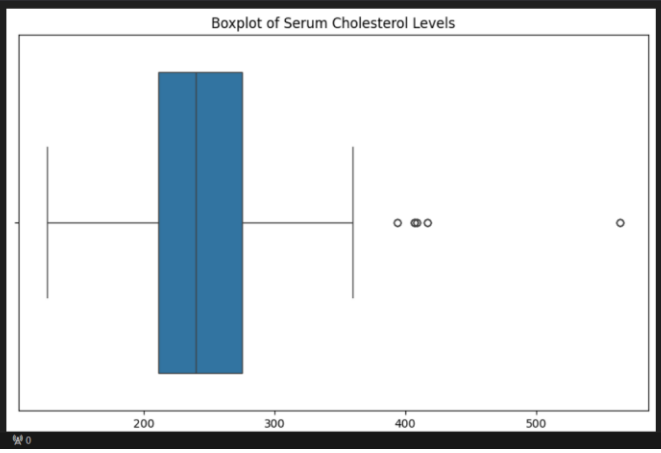
```python
# Histogram for cholesterol distribution
plt.figure(figsize=(10, 6))
sns.histplot(data['chol'], kde=True)
plt.title('Distribution of Serum Cholesterol Levels')
plt.xlabel('Cholesterol (mg/dl)')
plt.ylabel('Frequency')
plt.show()
```

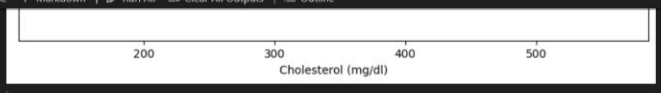


```
plt.ylabel('Frequency')
plt.show()
```




```
# Boxplot to identify outliers in cholesterol levels
plt.figure(figsize=(10, 6))
sns.boxplot(x=data['chol'])
plt.title('Boxplot of Serum Cholesterol Levels')
plt.xlabel('Cholesterol (mg/dl)')
plt.show()
```






```
Q1 = data['chol'].quantile(0.25)
Q3 = data['chol'].quantile(0.75)
IQR = Q3 - Q1

# Identify outliers
outliers = data[(data['chol'] < (Q1 - 1.5 * IQR)) | (data['chol'] > (Q3 + 1.5 * IQR))]
outliers
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	age_group
123	65	0	2	140	417	1	0	157	0	0.8	2	1	2	1	60-80
158	67	0	2	115	564	0	0	160	0	1.6	1	0	3	1	60-80
179	56	0	0	134	409	0	0	150	1	1.9	1	2	3	0	40-60
192	67	0	2	115	564	0	0	160	0	1.6	1	0	3	1	60-80
255	62	0	0	140	394	0	0	157	0	1.2	1	0	2	1	60-80
450	63	0	0	150	407	0	0	154	0	4.0	1	3	3	0	60-80
464	67	0	2	115	564	0	0	160	0	1.6	1	0	3	1	60-80
481	63	0	0	150	407	0	0	154	0	4.0	1	3	3	0	60-80
542	62	0	0	140	394	0	0	157	0	1.2	1	0	2	1	60-80
578	62	0	0	140	394	0	0	157	0	1.2	1	0	2	1	60-80
641	56	0	0	134	409	0	0	150	1	1.9	1	2	3	0	40-60
665	65	0	2	140	417	1	0	157	0	0.8	2	1	2	1	60-80

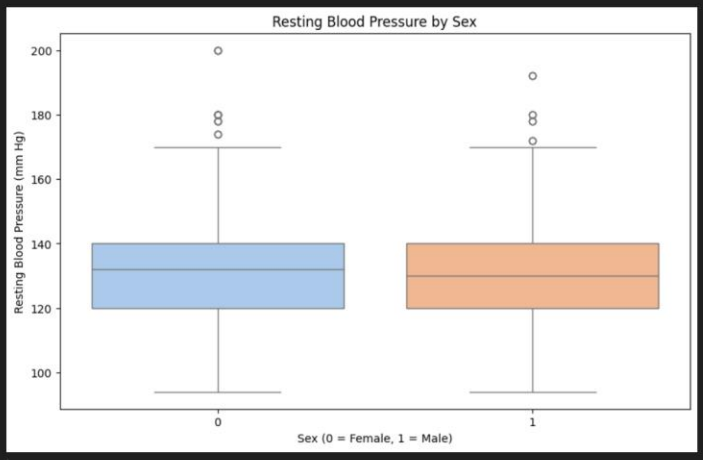
665	65	0	2	140	417	1	0	157	0	0.8	2	1	2	1	60-80
685	63	0	0	150	407	0	0	154	0	4.0	1	3	3	0	60-80
889	63	0	0	150	407	0	0	154	0	4.0	1	3	3	0	60-80
958	65	0	2	140	417	1	0	157	0	0.8	2	1	2	1	60-80
996	56	0	0	134	409	0	0	150	1	1.9	1	2	3	0	40-60

```python
# Question 10
# 10. How does the resting blood pressure (trestbps) vary between males and females?

# Assuming 'sex' is represented as 0 for female and 1 for male
data.groupby('sex')['trestbps'].describe()
```

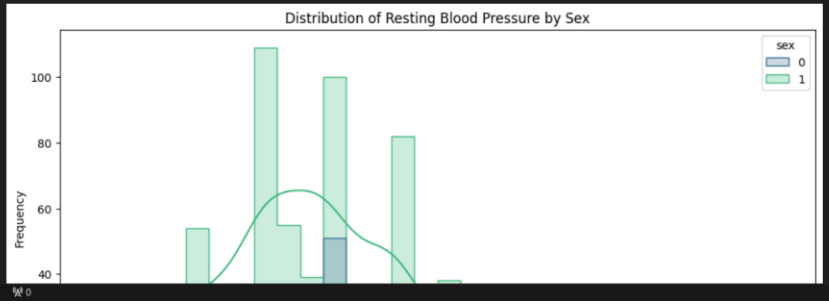
sex	count	mean	std	min	25%	50%	75%	max
0	312.0	133.701923	19.581578	94.0	120.0	132.0	140.0	200.0
1	713.0	130.697055	16.463723	94.0	120.0	130.0	140.0	192.0

```python
plt.figure(figsize=(10, 6))
sns.boxplot(x='sex', y='trestbps', data=data, palette='pastel')
plt.title('Resting Blood Pressure by Sex')
plt.xlabel('Sex (0 = Female, 1 = Male)')
plt.ylabel('Resting Blood Pressure (mm Hg)')
plt.show()
```

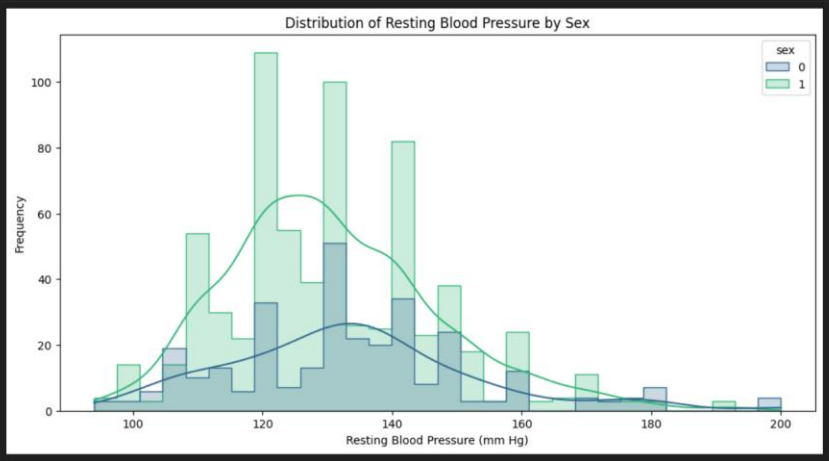
```
C:\Users\User\AppData\Local\Temp\ipykernel_2216\2538616903.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.
```




```python
plt.figure(figsize=(12, 6))
sns.histplot(data=data, x='trestbps', hue='sex', element='step', palette='viridis', bins=30, kde=True)
plt.title('Distribution of Resting Blood Pressure by Sex')
plt.xlabel('Resting Blood Pressure (mm Hg)')
plt.ylabel('Frequency')
plt.show()
```



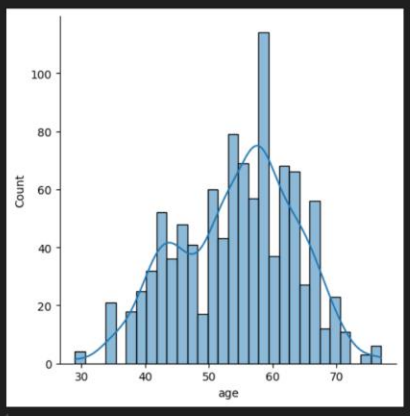
```
    plt.show()
```




```
    # Now let's take a look on our age column
```

```
    # Now let's take a look on our age column
    # Create a distribution plot with normal distribution curve

    sns.displot(x='age', data=data, bins=30, kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x1fffa641730>
```



1. 58-59 year old people are the most in the dataset.

```
    # Let's plot another distribution plot for Maximum Heart Rate.

    sns.displot(x='thalach', data=data, bins=30, kde=True, color='brown')
```

```
<seaborn.axisgrid.FacetGrid at 0x1fffcde9550>
```