Main Flow Task- 5

Description

The Heart Disease Analysis project involves analysing a dataset related to heart disease to identify key factors that contribute to heart disease occurrence. The goal is to use data analytics techniques to predict the likelihood of heart disease based on various health indicators such as age, cholesterol levels, blood pressure, and other relevant featuresThe analysis aims to provide insights that can help in early diagnosis and prevention.

Responsibility

1. Data Cleaning: Handled missing data, outliers, and inconsistencies to ensure the dataset was suitable for analysis.

2. Exploratory Data Analysis (EDA): Performed EDA to understand the distribution of data, relationships between variables.

3. Question Formulation: Developed specific minimum questions relatedto heart disease, and solve each question by using appropriate functions.

4. Data Visualization:

Created visualizations using tools like Matplotlib Seabor, to effectively present the findings and insights gained from th analysis. This included charts, graphs, and other visual aids to make t e results easy to understand.nd.

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

# import the dataset file
import numpy as np
import pandas as pd

data=pd.read_csv('heart_disease_data.csv')

data
```

```
     Age Sex ChestPainType  RestingBP  Cholesterol  FastingBS
RestingECG  \
0     40   M           ATA        140          289          0
Normal
1     49   F           NAP        160          180          0
Normal
2     37   M           ATA        130          283          0
ST
```

```
3     48   F           ASY        138          214           0
Normal
4     54   M           NAP        150          195           0
Normal
..   ...  ..           ...        ...          ...         ...         .
..
913   45   M            TA        110          264           0
Normal
914   68   M           ASY        144          193           1
Normal
915   57   M           ASY        130          131           0
Normal
916   57   F           ATA        130          236           0
LVH
917   38   M           NAP        138          175           0
Normal

     MaxHR ExerciseAngina  Oldpeak ST_Slope  HeartDisease
0      172              N      0.0       Up             0
1      156              N      1.0     Flat             1
2       98              N      0.0       Up             0
3      108              Y      1.5     Flat             1
4      122              N      0.0       Up             0
..     ...            ...      ...      ...           ...
913    132              N      1.2     Flat             1
914    141              N      3.4     Flat             1
915    115              Y      1.2     Flat             1
916    174              N      0.0     Flat             1
917    173              N      0.0       Up             0

[918 rows x 12 columns]

data.head()

   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
```

```
data.tail()
```

```
     Age Sex ChestPainType  RestingBP  Cholesterol  FastingBS
RestingECG  \
913   45   M            TA        110          264          0
Normal
914   68   M           ASY        144          193          1
Normal
915   57   M           ASY        130          131          0
Normal
916   57   F           ATA        130          236          0
LVH
917   38   M           NAP        138          175          0
Normal

     MaxHR ExerciseAngina  Oldpeak ST_Slope  HeartDisease
913    132              N      1.2     Flat             1
914    141              N      3.4     Flat             1
915    115              Y      1.2     Flat             1
916    174              N      0.0     Flat             1
917    173              N      0.0       Up             0
```

```
data.describe()
```

```
               Age   RestingBP  Cholesterol   FastingBS       MaxHR  \
count  918.000000  918.000000   918.000000  918.000000  918.000000
mean    53.510893  132.396514   198.799564    0.233115  136.809368
std      9.432617   18.514154   109.384145    0.423046   25.460334
min     28.000000    0.000000     0.000000    0.000000   60.000000
25%     47.000000  120.000000   173.250000    0.000000  120.000000
50%     54.000000  130.000000   223.000000    0.000000  138.000000
75%     60.000000  140.000000   267.000000    0.000000  156.000000
max     77.000000  200.000000   603.000000    1.000000  202.000000

          Oldpeak  HeartDisease
count  918.000000    918.000000
mean     0.887364      0.553377
std      1.066570      0.497414
min     -2.600000      0.000000
25%      0.000000      0.000000
50%      0.600000      1.000000
75%      1.500000      1.000000
max      6.200000      1.000000
```

```
data.isnull().sum()
```

```
Age               0
Sex               0
ChestPainType     0
RestingBP         0
Cholesterol       0
FastingBS         0
RestingECG        0
MaxHR             0
ExerciseAngina    0
Oldpeak           0
ST_Slope          0
HeartDisease      0
dtype: int64
```

```
# Calculate and print statistics for each column
print(f"Column: Age")
print(f"Mean: {data['Age'].mean()}")
print(f"Median: {data['Age'].median()}")
print(f"Max: {data['Age'].max()}")
print(f"Min: {data['Age'].min()}\n")

print(f"Cloumn:RestingBP")
print(f"mean:{data['RestingBP'].mean()}")
print(f"median:{data['RestingBP'].median()}")
print(f"max:{data['RestingBP'].max()}")
print(f"min:{data['RestingBP'].min()}")

print(f"column:Cholesterol ")
print(f"mean:{data['Cholesterol'].mean()}")
print(f"medin:{data['Cholesterol'].median()}")
print(f"max:{data['Cholesterol'].max()}")
print(f"min:{data['Cholesterol'].min()}")

print(f"column:FastingBS")
print(f"mean:{data['FastingBS'].mean()}")
print(f"medin:{data['FastingBS'].median()}")
print(f"max:{data['FastingBS'].max()}")
print(f"min:{data['FastingBS'].min()}")

print(f"column:MaxHR")
print(f"mean:{data['MaxHR'].mean()}")
print(f"medin:{data['MaxHR'].median()}")
print(f"max:{data['MaxHR'].max()}")
print(f"min:{data['MaxHR'].min()}")

print(f"column:Oldpeak")
print(f"mean:{data['Oldpeak'].mean()}")
print(f"medin:{data['Oldpeak'].median()}")
print(f"max:{data['Oldpeak'].max()}")
```

```python
print(f"min:{data['Oldpeak'].min()}")

print(f"column:HeartDisease")
print(f"mean:{data['HeartDisease'].mean()}")
print(f"medin:{data['HeartDisease'].median()}")
print(f"max:{data['HeartDisease'].max()}")
print(f"min:{data['HeartDisease'].min()}")
```

```
Column: Age
Mean: 53.510893246187365
Median: 54.0
Max: 77
Min: 28

Cloumn:RestingBP
mean:132.39651416122004
median:130.0
max:200
min:0
column:Cholesterol
mean:198.7995642701525
medin:223.0
max:603
min:0
column:FastingBS
mean:0.2331154684095860 5
medin:0.0
max:1
min:0
column:MaxHR
mean:136.80936819172112
medin:138.0
max:202
min:60
column:Oldpeak
mean:0.8873638344226579
medin:0.6
max:6.2
min:-2.6
column:HeartDisease
mean:0.5533769063180828
medin:1.0
max:1
min:0
```

```python
#find how many people having hart dieses and what is their percentage

# Count the number of people with heart disease (where HeartDisease = 1)
total_heart_disease_cases = data['HeartDisease'].sum()
```

```
# Calculate the total number of people in the dataset
total_people = len(data)
```

```
total_heart_disease_cases
```

```
508
```

```
total_people
```

```
918
```

```
# Calculate the percentage of people with heart disease
percentage_heart_disease = (total_heart_disease_cases / total_people)
* 100
number_of_people_heartdieses=(total_heart_disease_cases /
total_people)
```

```
number_of_people_heartdieses
```

```
0.5533769063180828
```

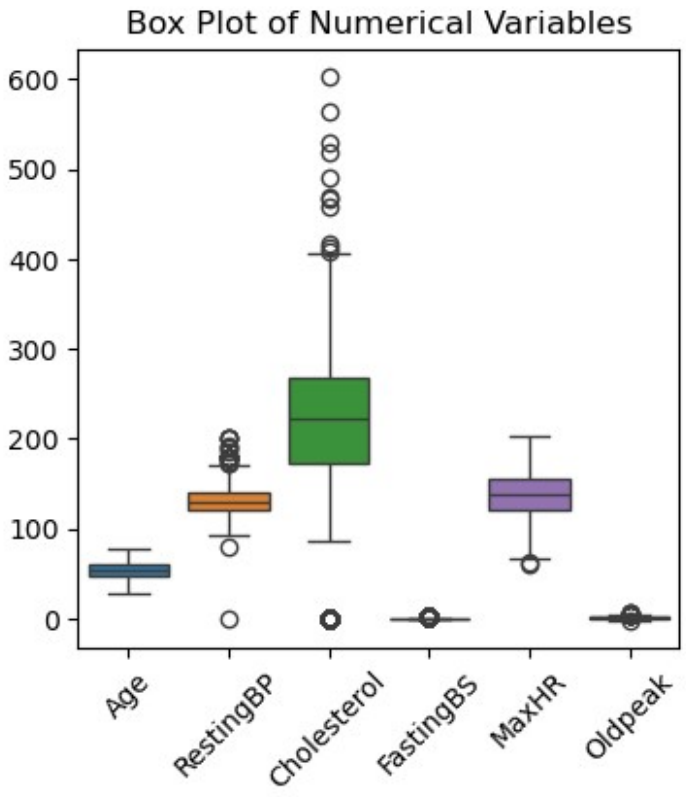
```
percentage_heart_disease
```

```
55.33769063180828
```

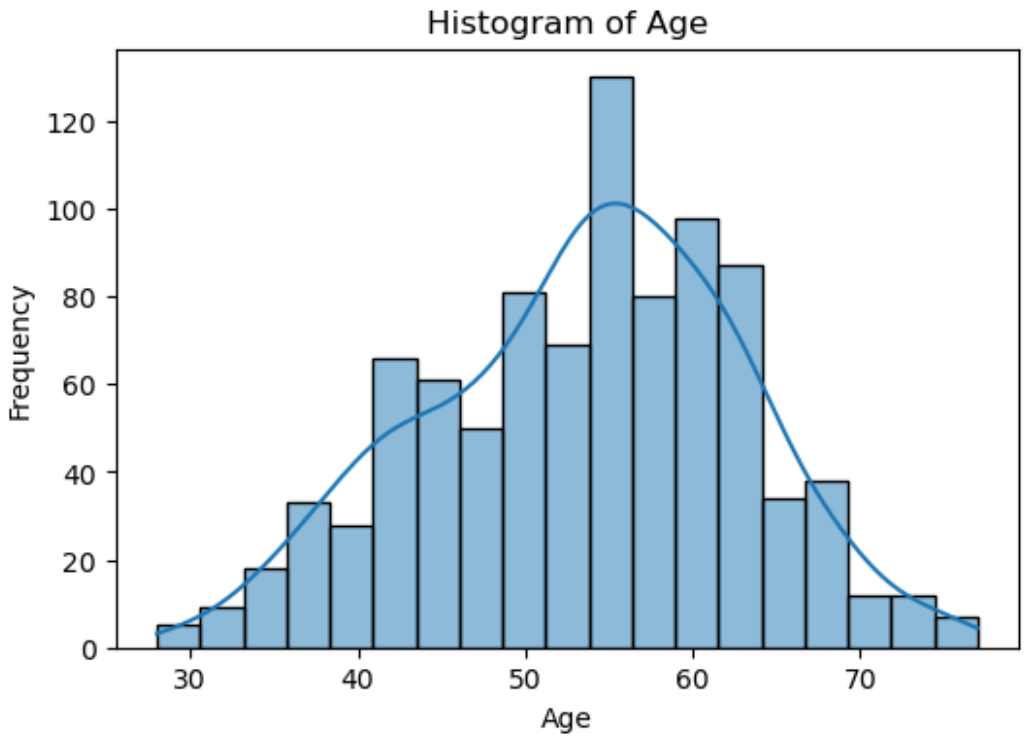
```
print(f"Number of people with heart disease:
{number_of_people_heartdieses}")
print(f"Percentage of people with heart disease:
{percentage_heart_disease}")
```

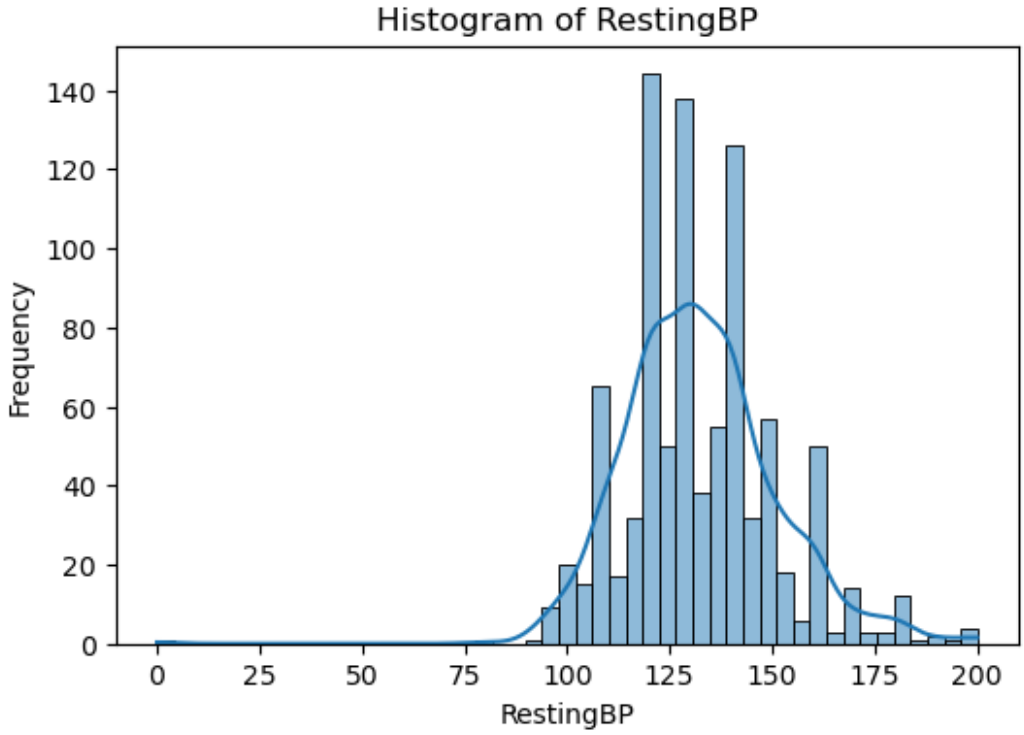
```
Number of people with heart disease: 0.5533769063180828
Percentage of people with heart disease: 55.33769063180828
```

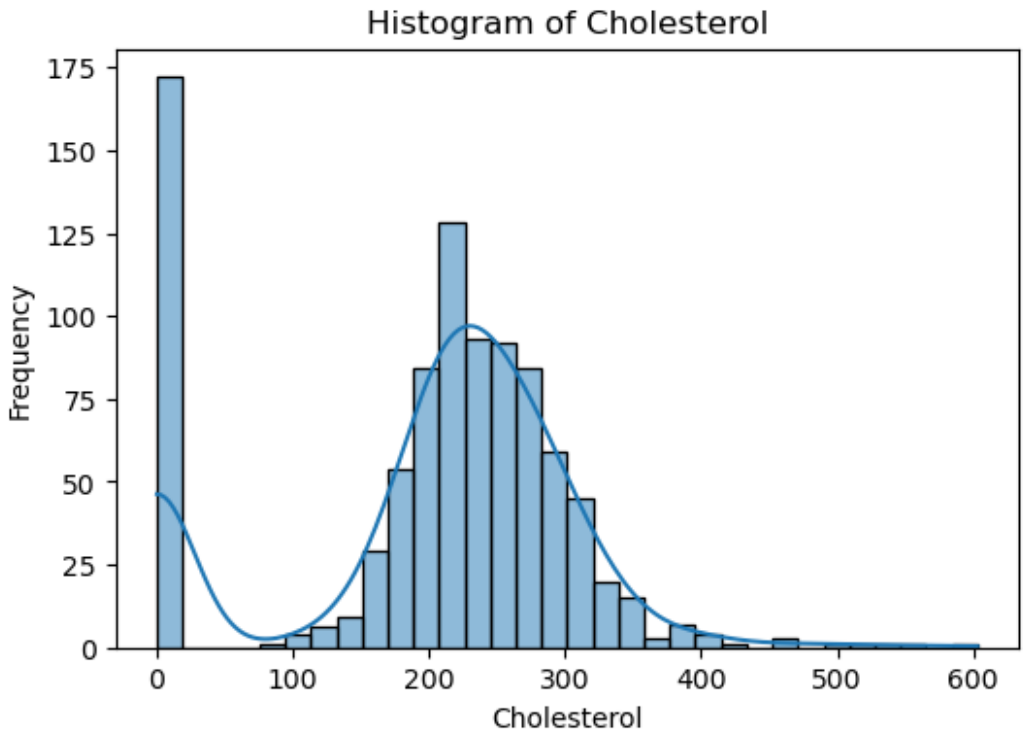
```
# Box plots for numerical variables
numerical_cols = ['Age', 'RestingBP', 'Cholesterol', 'FastingBS',
'MaxHR', 'Oldpeak']
plt.figure(figsize=(4, 4))
sns.boxplot(data=data[numerical_cols])
plt.title('Box Plot of Numerical Variables')
plt.xticks(rotation=45)
plt.show()
```

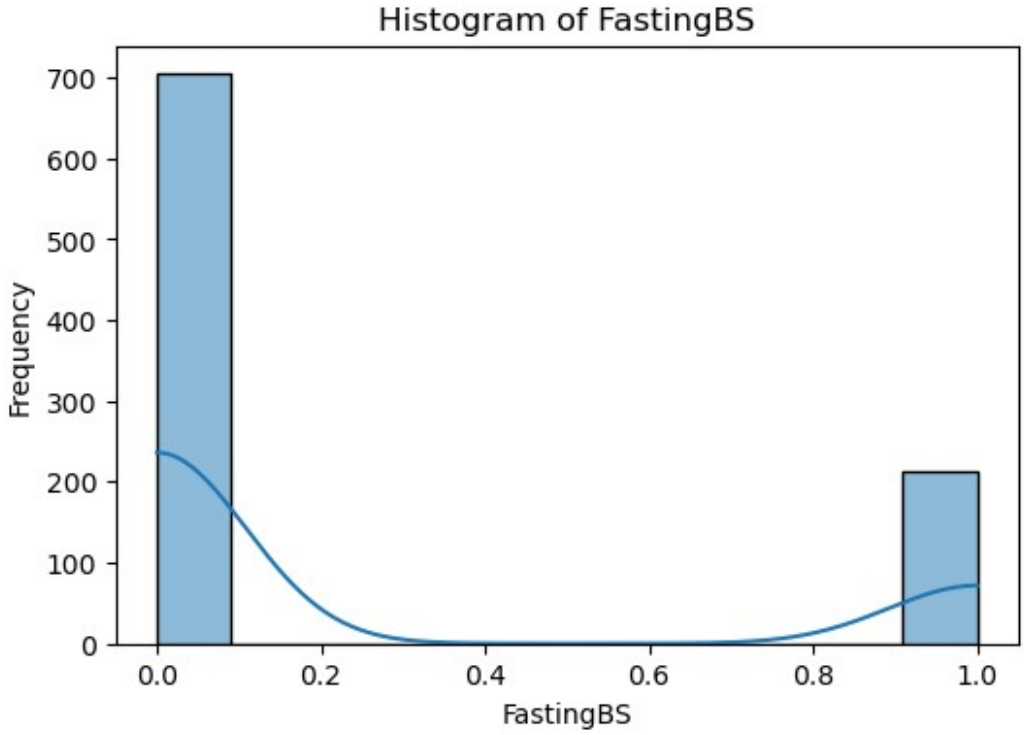


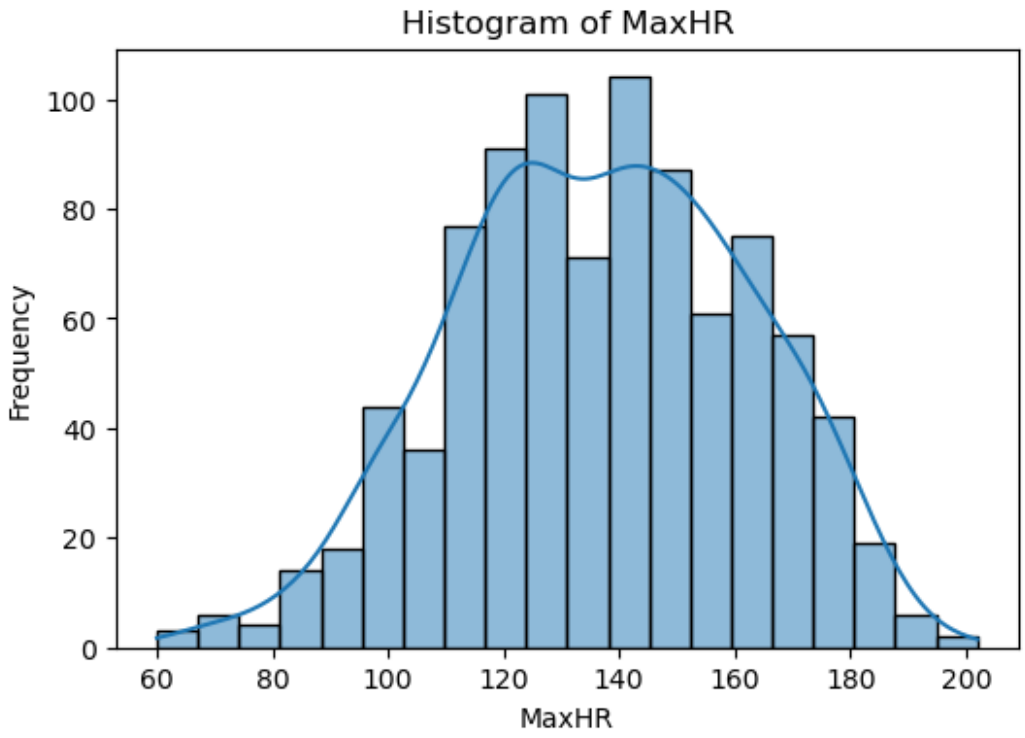

```
# Identify numerical columns
numerical_cols = ['Age', 'RestingBP', 'Cholesterol', 'FastingBS',
'MaxHR', 'Oldpeak']


# Create histograms for each numerical variable
for column in numerical_cols:
    plt.figure(figsize=(6, 4))
    sns.histplot(data[column], kde=True)
    plt.title(f'Histogram of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()
```

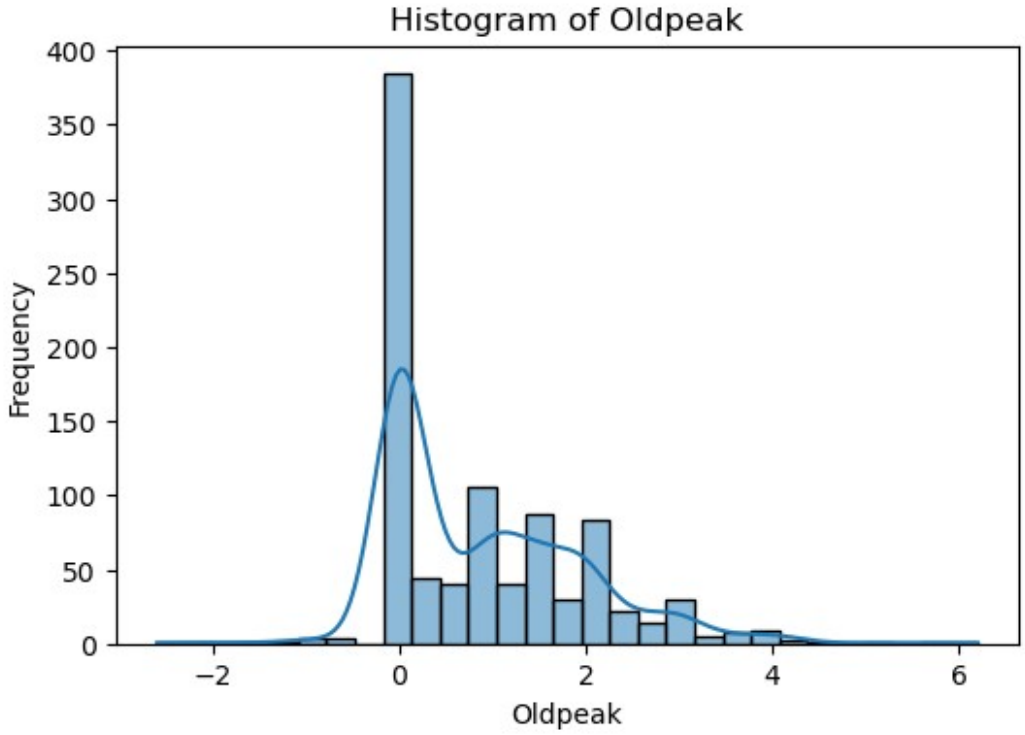

Histogram of Age
Frequency
Age
0
20
40
60
80
100
120
30
40
50
60
70


Histogram of RestingBP
Frequency
RestingBP
0
20
40
60
80
100
120
140
0
25
50
75
100
125
150
175
200


Histogram of Cholesterol
Frequency
Cholesterol
0
25
50
75
100
125
150
175
0
100
200
300
400
500
600


Histogram of FastingBS
Frequency
FastingBS
0
100
200
300
400
500
600
700
0.0
0.2
0.4
0.6
0.8
1.0

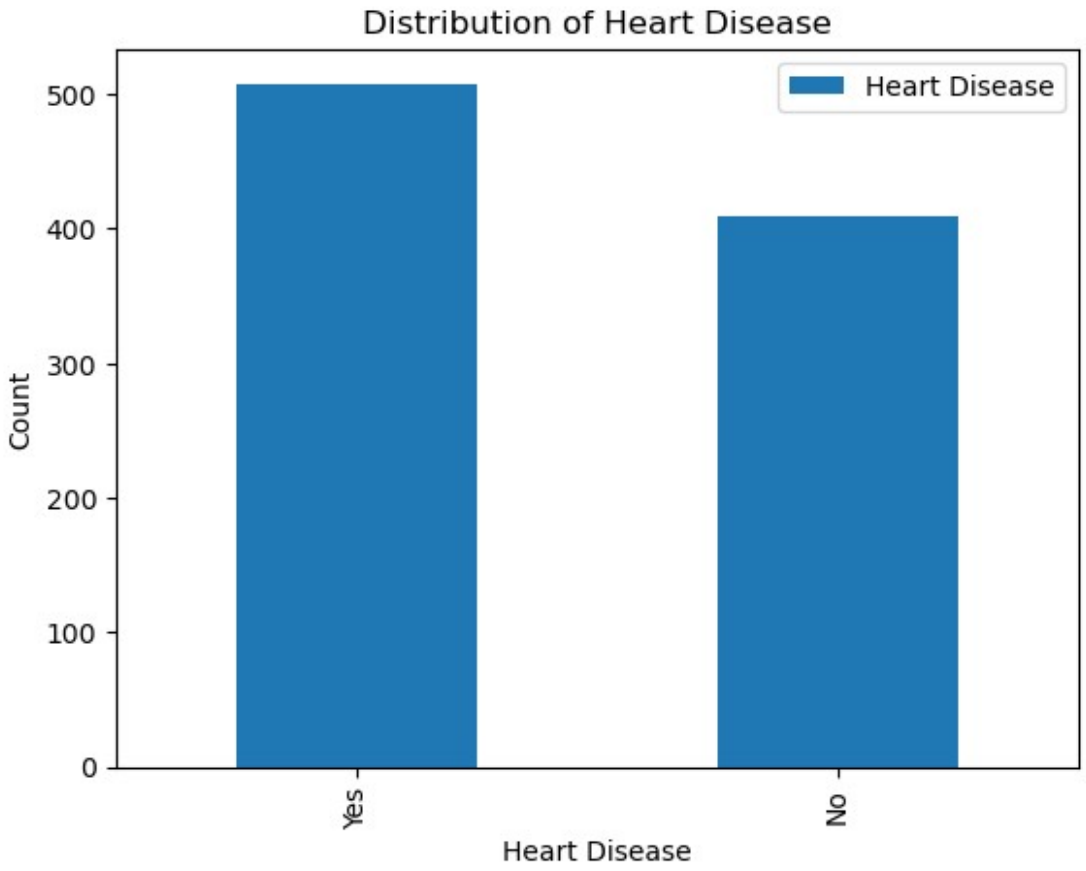




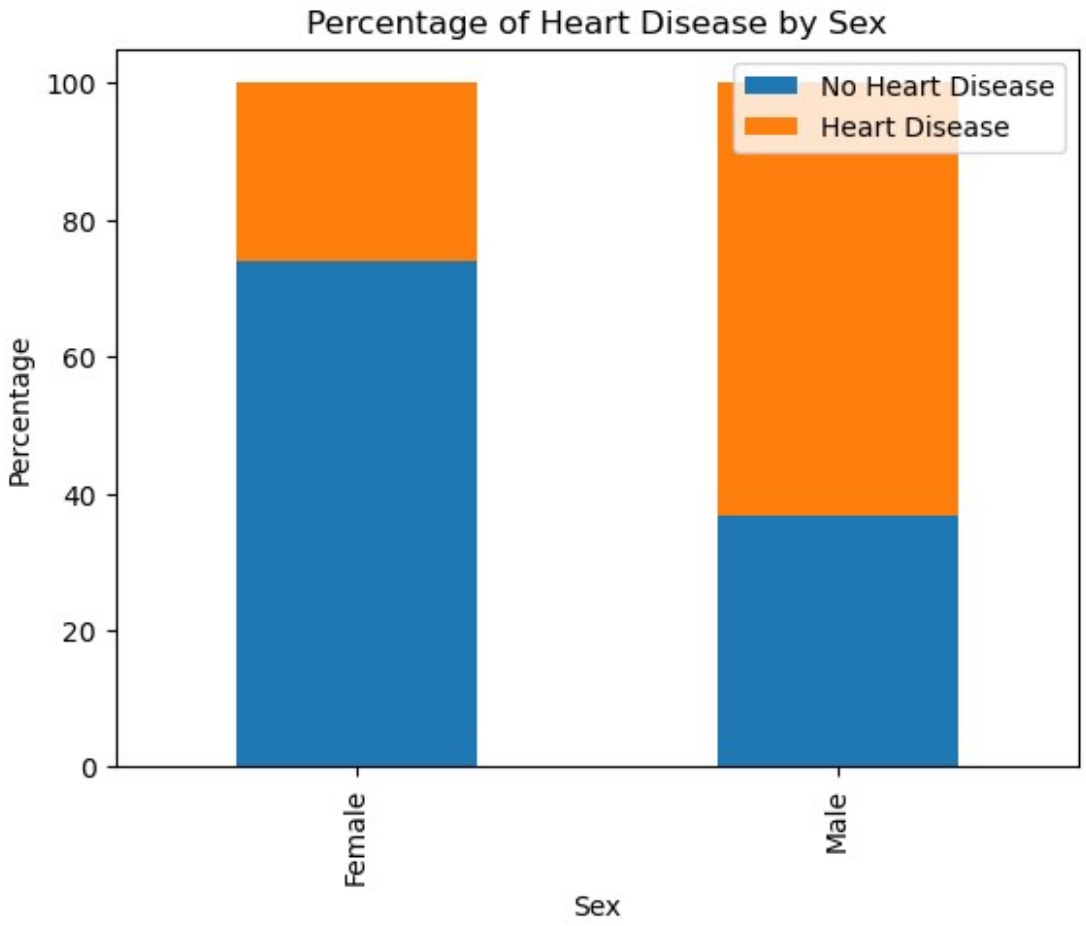

```
#how many people have hart dieses and how many people have doesent
have heart dieses
```

```
# Count the number of people with heart disease (where HeartDisease =
1)
total_heart_disease_cases = data['HeartDisease'].sum()

# Calculate the number of people without heart disease (where
HeartDisease = 0)
total_no_heart_disease_cases = len(data) - total_heart_disease_cases

print(f"Number of people with heart disease:
{total_heart_disease_cases}")
print(f"Number of people without heart disease:
{total_no_heart_disease_cases}")

# Create a new DataFrame to visualize the counts
data = {'Heart Disease': [total_heart_disease_cases,
total_no_heart_disease_cases]}
df_plot = pd.DataFrame(data, index=['Yes', 'No'])

# Plot the histogram
df_plot.plot(kind='bar')
plt.title('Distribution of Heart Disease')
plt.xlabel('Heart Disease')
plt.ylabel('Count')
plt.show()
```

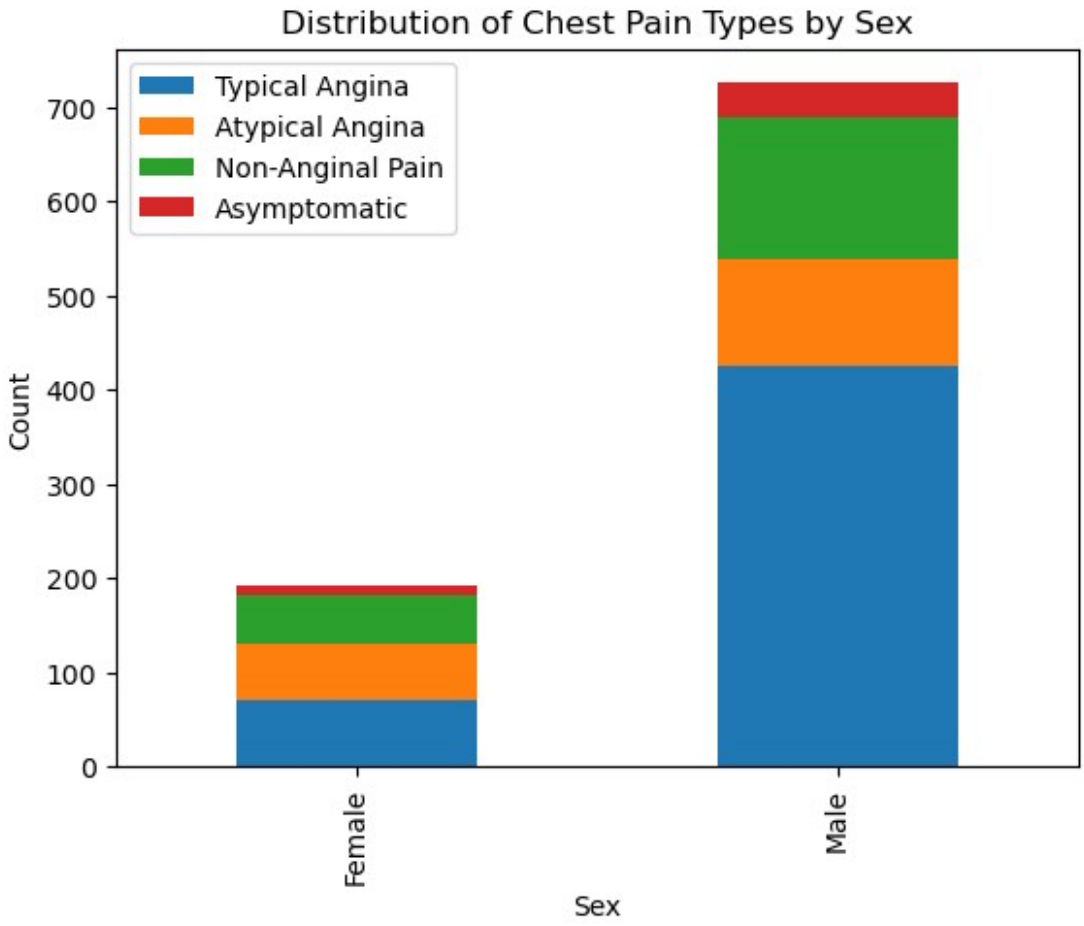
```
Number of people with heart disease: 508
Number of people without heart disease: 410
```




```
#people have which sex have most heart dieses

# Group the data by sex and count the number of occurrences of heart
disease (target = 1)
sex_counts = data.groupby('Sex')
['HeartDisease'].value_counts().unstack()

# Calculate the percentage of heart disease for each sex
sex_percentages = sex_counts.div(sex_counts.sum(axis=1), axis=0) * 100

# Plot the results
sex_percentages.plot(kind='bar', stacked=True)
plt.title('Percentage of Heart Disease by Sex')
plt.xlabel('Sex')
plt.ylabel('Percentage')
plt.xticks([0, 1], ['Female', 'Male'])
plt.legend(['No Heart Disease', 'Heart Disease'])
plt.show()
```




```
#people of which sex has which has which type of chest pain most
# Group the data by sex and chest pain type, and count the occurrences
grouped_data = data.groupby(['Sex', 'ChestPainType']).size().unstack()

# Plot the results
grouped_data.plot(kind='bar', stacked=True)
plt.title('Distribution of Chest Pain Types by Sex')
plt.xlabel('Sex')
plt.ylabel('Count')
plt.xticks([0, 1], ['Female', 'Male'])
plt.legend(['Typical Angina', 'Atypical Angina', 'Non-Anginal Pain',
'Asymptomatic'])
plt.show()
```


Distribution of Chest Pain Types by Sex
Typical Angina
Atypical Angina
Non-Anginal Pain
Asymptomatic
Count
0
100
200
300
400
500
600
700
Female
Male
Sex