Part A

Load required libraries

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

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
heart_disease = fetch_ucirepo(id=45)

# data (as pandas dataframes)
X = heart_disease.data.features
y = heart_disease.data.targets

# metadata
print(heart_disease.metadata)

# variable information
print(heart_disease.variables)
```

```
0        age  Feature      Integer       Age
1        sex  Feature  Categorical       Sex
2         cp  Feature  Categorical      None
3   trestbps  Feature      Integer      None
4       chol  Feature      Integer      None
5        fbs  Feature  Categorical      None
6    restecg  Feature  Categorical      None
7    thalach  Feature      Integer      None
8      exang  Feature  Categorical      None
9    oldpeak  Feature      Integer      None
10     slope  Feature  Categorical      None
11        ca  Feature      Integer      None
12      thal  Feature  Categorical      None
13       num   Target      Integer      None

                                          description  units missing_values
0                                                None  years             no
1                                                None   None             no
2                                                None   None             no
3   resting blood pressure (on admission to the ho...  mm Hg             no
4                                   serum cholestoral  mg/dl             no
5                         fasting blood sugar > 120 mg/dl   None             no
6                                                None   None             no
7                             maximum heart rate achieved   None             no
8                                exercise induced angina   None             no
9   ST depression induced by exercise relative to ...   None             no
10                                               None   None             no
11  number of major vessels (0-3) colored by flour...   None            yes
12                                               None   None            yes
13                             diagnosis of heart disease   None             no
```

Variables Description:

1-age: Age of the patient in years
2-sex: Male/Female
3-cp: chest pain type: typical angina, atypical angina, non-anginal, asymptomatic
4-trestbps: resting blood pressure (resting blood pressure (in mm Hg on admission to the hospital))
5-chol: serum cholesterol in mg/dl
6-fbs: if fasting blood sugar > 120 mg/dl
7-restecg: resting electrocardiographic results Values: [normal, stt abnormality, lv hypertrophy]
8-thalach: maximum heart rate achieved
9-exang: exercise-induced angina (True/ False)
10-oldpeak: ST depression induced by exercise relative to rest
11-slope: the slope of the peak exercise ST segment
12-ca: number of major vessels (0-3) colored by fluoroscopy
13-thal: [normal; fixed defect; reversible defect]
14-num: the predicted attribute

```python
# Convert data into pandas DataFrame
df = pd.DataFrame(data=X, columns=heart_disease.feature_names)

# Add the target variable to the DataFrame
df['num'] = y

# Display the head of the DataFrame
print(df.head())
```

```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   63    1   1       145   233    1        2      150      0      2.3      3
1   67    1   4       160   286    0        2      108      1      1.5      2
2   67    1   4       120   229    0        2      129      1      2.6      2
3   37    1   3       130   250    0        0      187      0      3.5      3
4   41    0   2       130   204    0        2      172      0      1.4      1

    ca  thal  num
0  0.0   6.0    0
1  3.0   3.0    2
2  2.0   7.0    1
3  0.0   3.0    0
4  0.0   3.0    0
```

```python
## Display basic info about the dataset
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64  
 11  ca        299 non-null    float64
 12  thal      301 non-null    float64
 13  num       303 non-null    int64  
dtypes: float64(3), int64(11)
memory usage: 33.3 KB
None
```

```python
## check for missing values
print(df.isnull().sum())
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          4
thal        2
num         0
dtype: int64
```

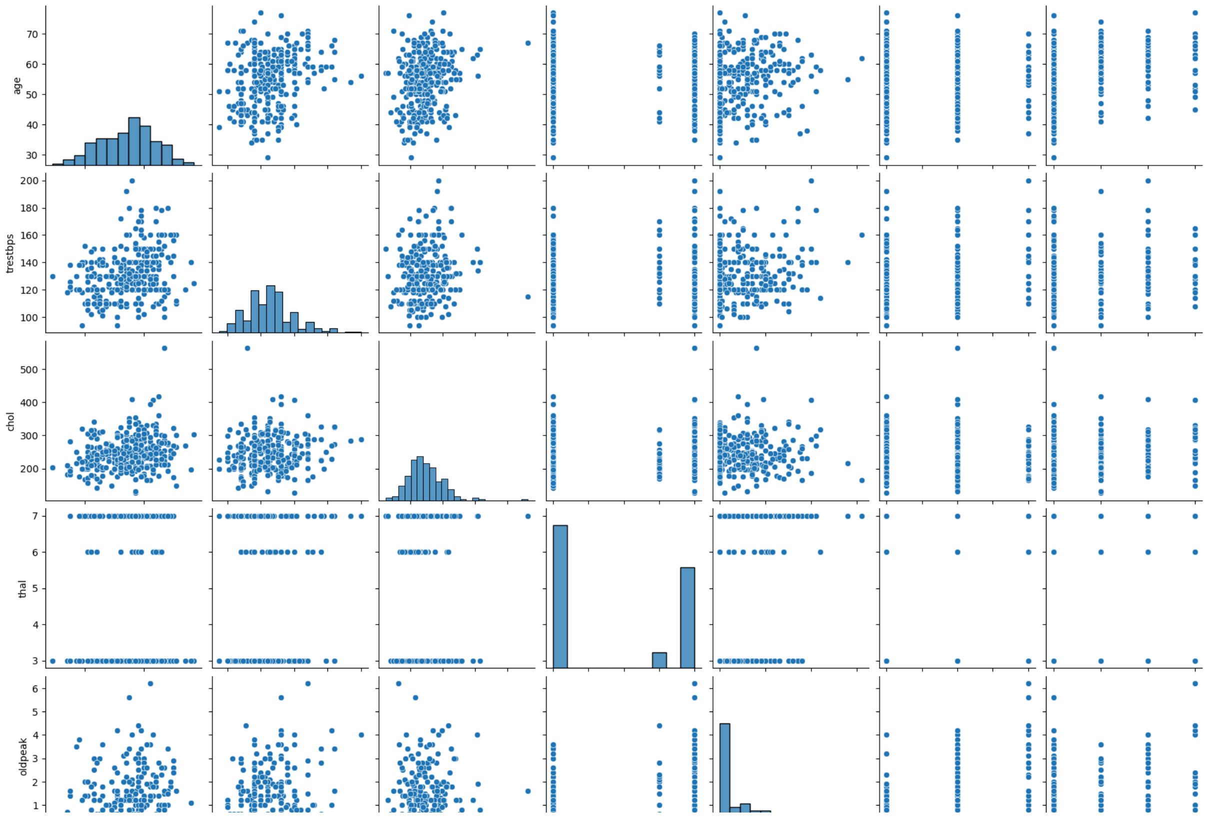
```python
df.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.438944	0.679868	3.158416	131.689769	246.693069	0.148515	0.990099	149.607261
std	9.038662	0.467299	0.960126	17.599748	51.776918	0.356198	0.994971	22.875003
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000
25%	48.000000	0.000000	3.000000	120.000000	211.000000	0.000000	0.000000	133.500000
50%	56.000000	1.000000	3.000000	130.000000	241.000000	0.000000	1.000000	153.000000
75%	61.000000	1.000000	4.000000	140.000000	275.000000	0.000000	2.000000	166.000000
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000

```
sns.pairplot(df[['age','trestbps','chol','thal','oldpeak','slope', 'ca']])
```

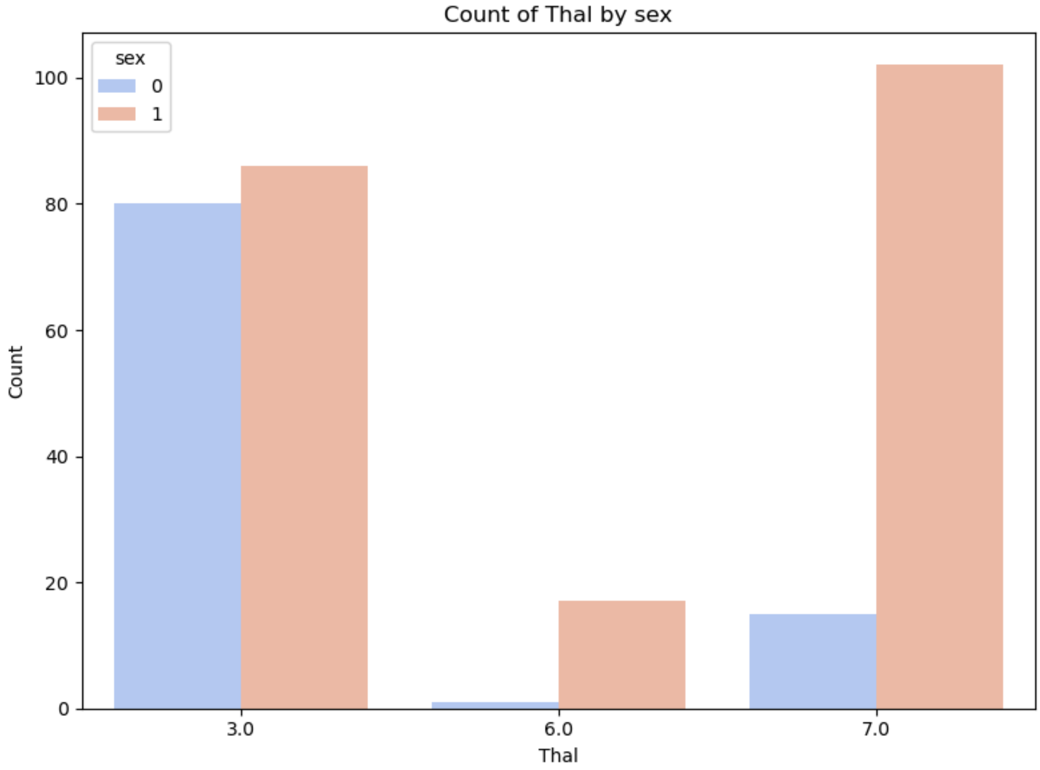
```
/Users/cardinle/anaconda3/lib/python3.11/site-packages/seaborn/axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

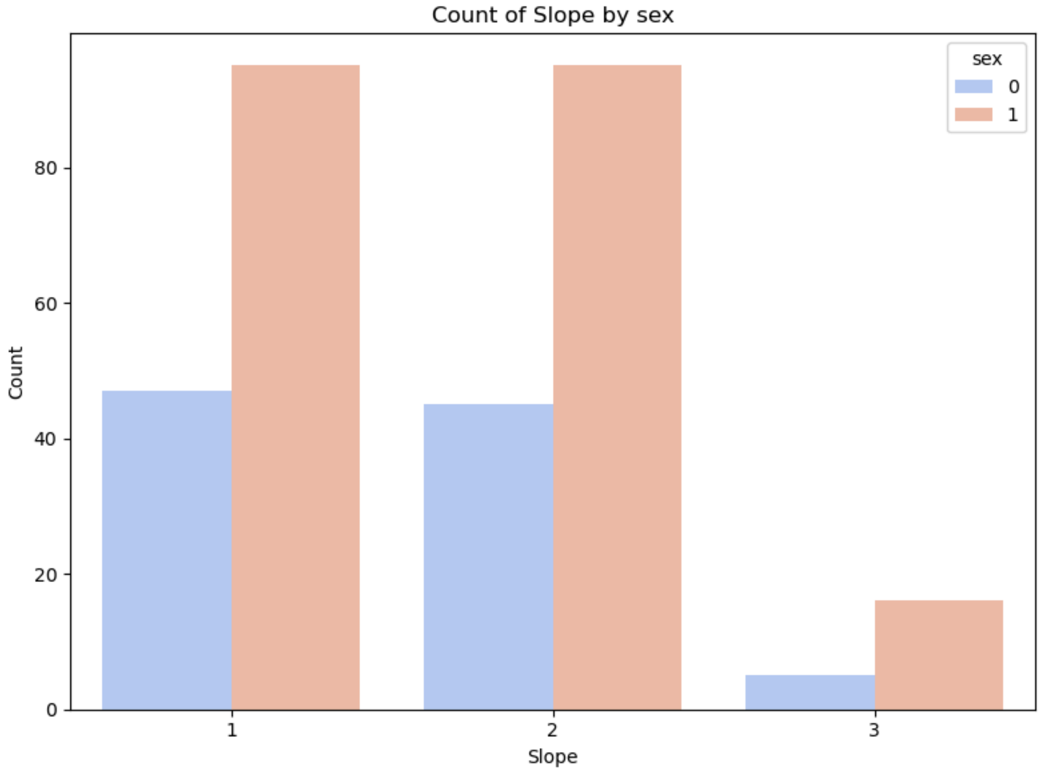
```
<seaborn.axisgrid.PairGrid at 0x12f0c2e10>
```

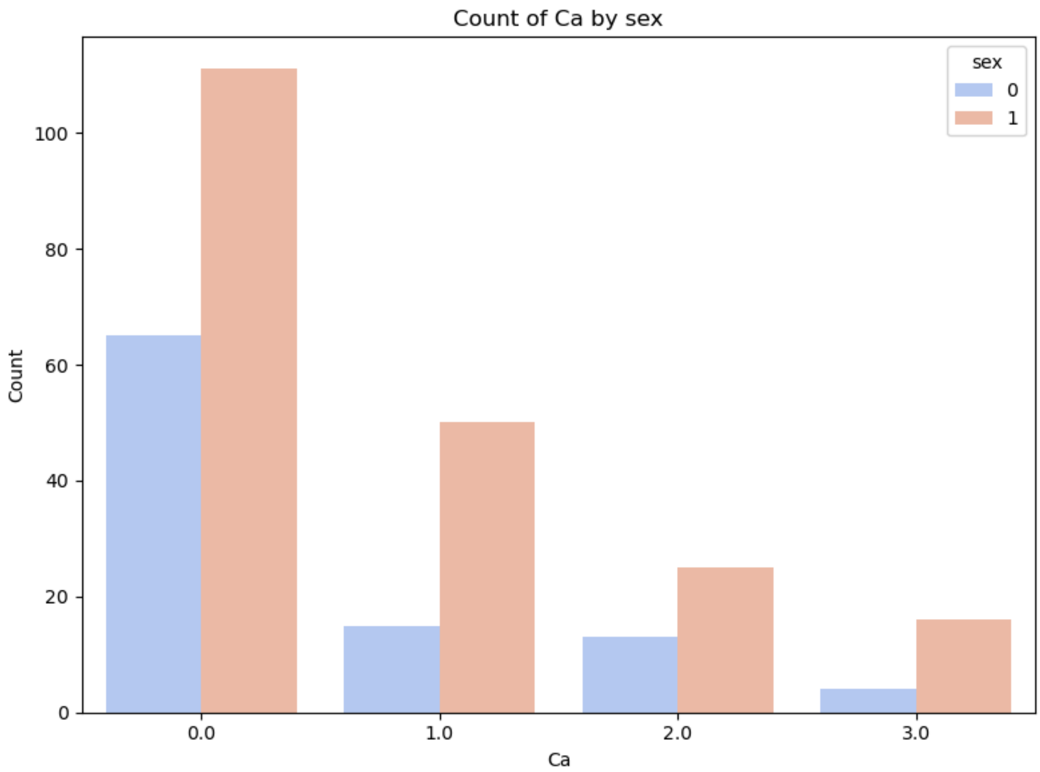


```python
# Calculate and visualize correlation matrix
correlation_matrix = df[['age','trestbps','chol','thal','oldpeak', 'slope', 'ca']].corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=.5)
plt.title('Correlation Matrix')
plt.show()
```

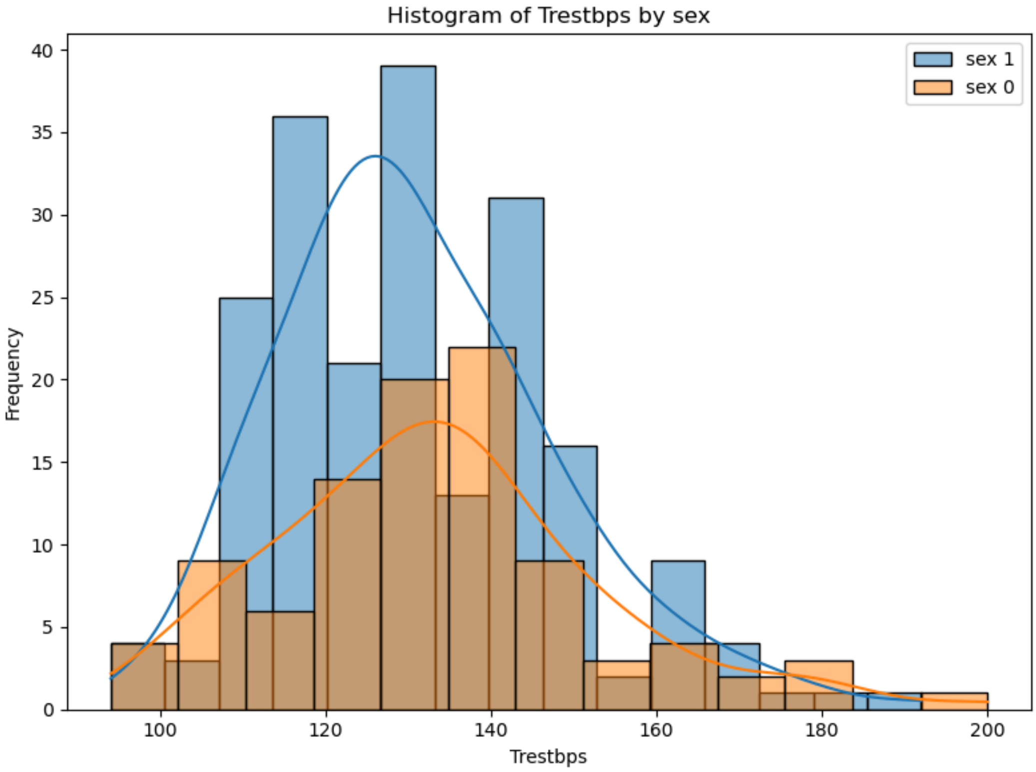
```
cat_var = 'sex'
cont_var = ['thal', 'slope', 'ca']

# Create a facet grid of bar plots for each continuous variable
for cont_var in cont_var:
    plt.figure(figsize=(8, 6))
    sns.countplot(data=df, x=cont_var, hue=cat_var, palette='coolwarm')
    plt.title(f'Count of {cont_var.capitalize()} by {cat_var}')
    plt.xlabel(cont_var.capitalize())
    plt.ylabel('Count')
    plt.legend(title=cat_var)
    plt.tight_layout()
    plt.show()
```

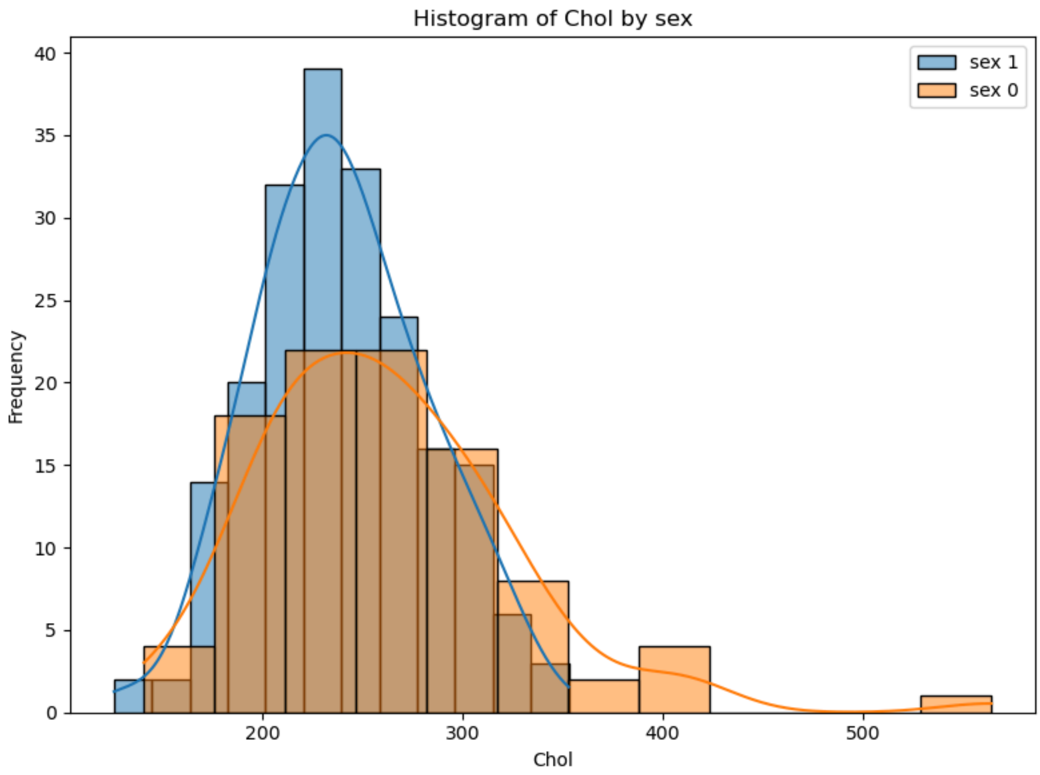


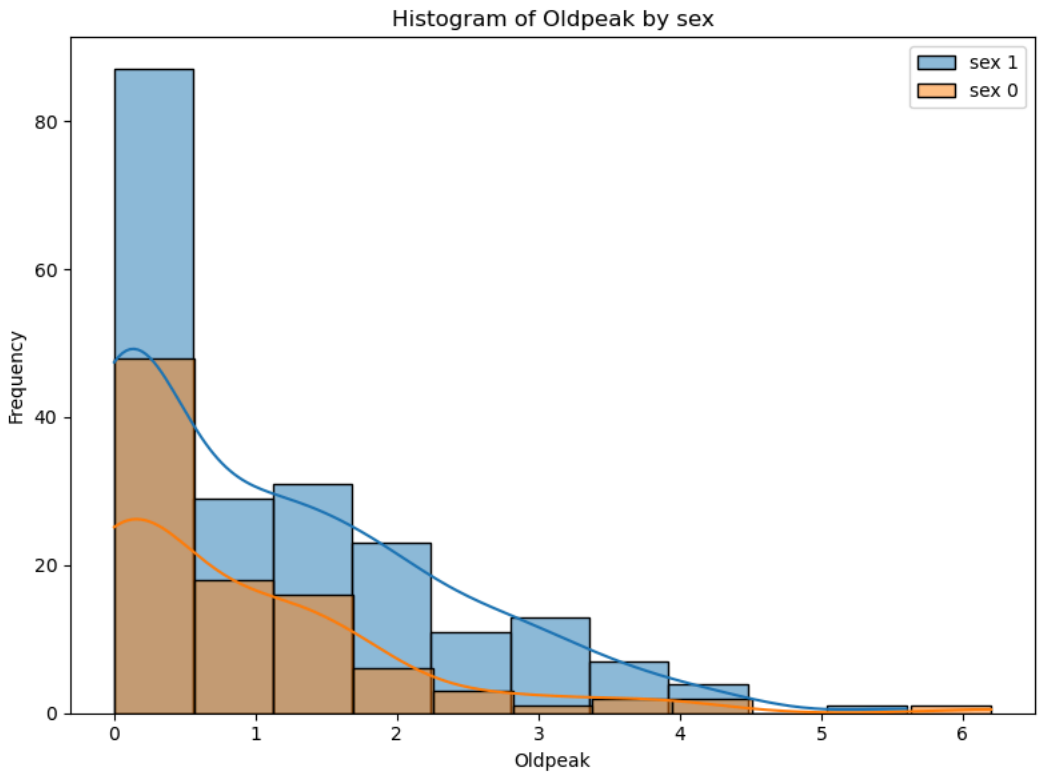



Count of Ca by sex
sex
0
1
Count
0
20
40
60
80
100
0.0
1.0
2.0
3.0
Ca

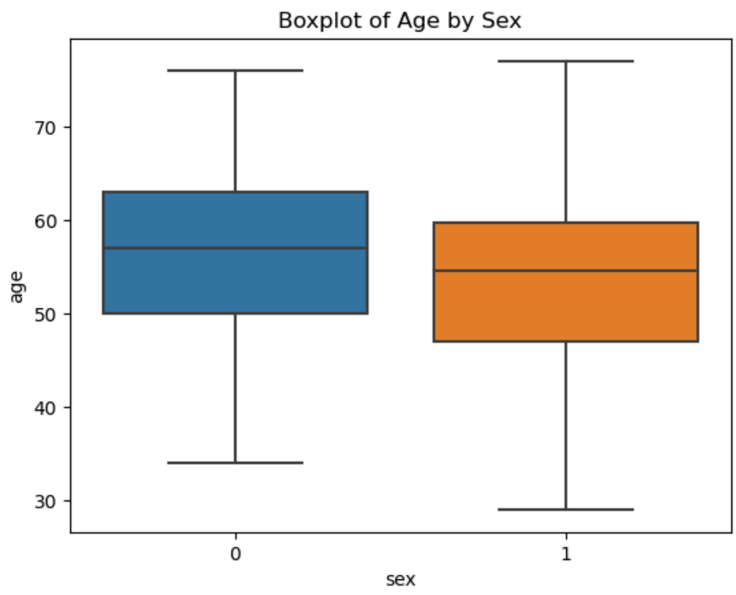
```
# Define categorical and continuous variables
cat_var = 'sex'
cont_var2 = ['trestbps', 'chol', 'oldpeak']

# Create a facet grid of histograms for each continuous variable
for cont_var in cont_var2:
    plt.figure(figsize=(8, 6))
    for category in df[cat_var].unique():
        sns.histplot(df[df[cat_var] == category][cont_var], kde=True, label=f'{cat_var} {category}')
    plt.title(f'Histogram of {cont_var.capitalize()} by {cat_var}')
    plt.xlabel(cont_var.capitalize())
    plt.ylabel('Frequency')
    plt.legend()
    plt.tight_layout()
    plt.show()
```

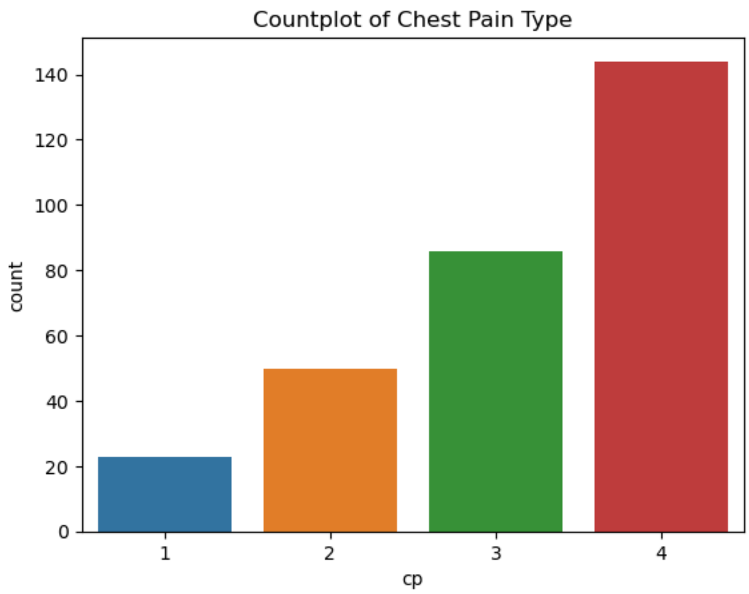



Histogram of Chol by sex
sex 1
sex 0
Frequency
Chol
0
5
10
15
20
25
30
35
40
200
300
400
500

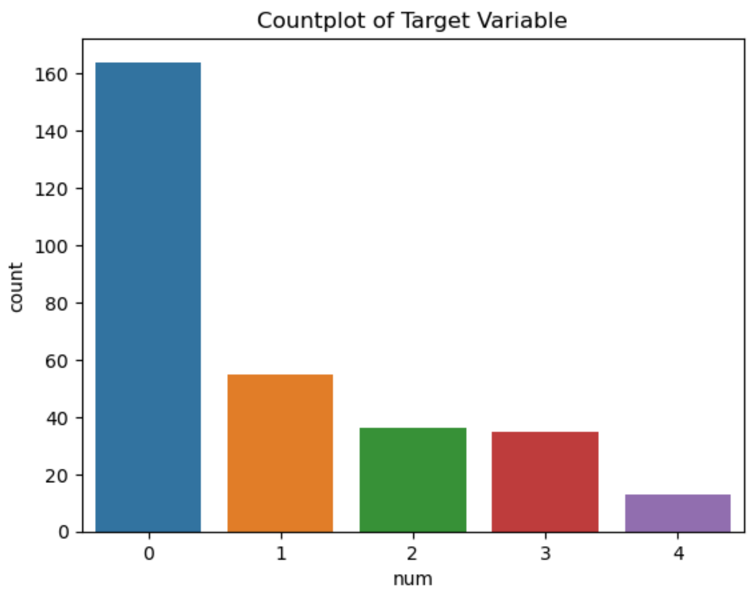

Histogram of Oldpeak by sex
sex 1
sex 0
Frequency
Oldpeak
0
20
40
60
80
0
1
2
3
4
5
6

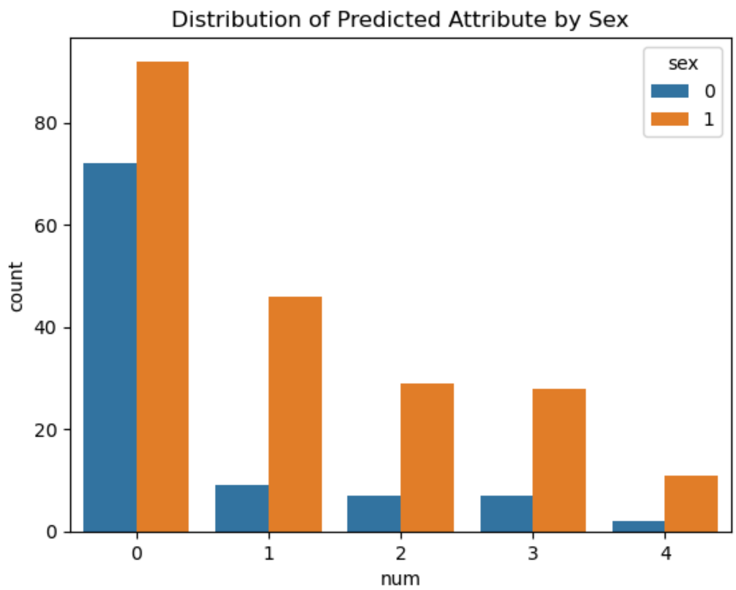
```
# Boxplot for categorical variable against a numerical variable
sns.boxplot(x='sex', y='age', data=df)
plt.title('Boxplot of Age by Sex')
plt.show()

# Countplot for a categorical variable
sns.countplot(x='cp', data=df)
plt.title('Countplot of Chest Pain Type')
plt.show()
```





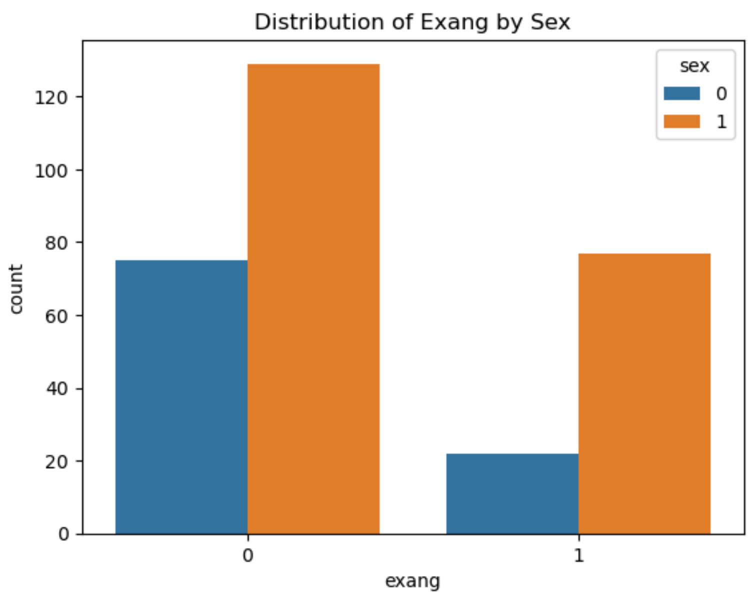
```
# Countplot for the target variable
sns.countplot(x='num', data=df)
plt.title('Countplot of Target Variable')
plt.show()

# Explore the relationship between the target and other variables
sns.countplot(x='num', hue='sex', data=df)
plt.title('Distribution of Predicted Attribute by Sex')
plt.show()
```





```
##  Association between sex and exang:

from scipy.stats import chi2_contingency

# Contingency table
contingency_table = pd.crosstab(df['sex'], df['exang'])

# Perform chi-square test
chi2, p, dof, expected = chi2_contingency(contingency_table)

# Print the results
print("Chi-square statistic:", chi2)
print("p-value:", p)
print("Degrees of freedom:", dof)
print("Expected frequencies table:")
print(expected)

# Explore the relationship between the Exang and Sex
sns.countplot(x='exang', hue='sex', data=df)
plt.title('Distribution of Exang by Sex')
plt.show()
```

```
Chi-square statistic: 5.825653331982741
p-value: 0.015794101260699977
Degrees of freedom: 1
Expected frequencies table:
[[ 65.30693069  31.69306931]
 [138.69306931  67.30693069]]
```



The chi-square test results indicate a significant association between 'sex' and 'exang' with a chi-square statistic of 5.83 (p-value = 0.016).

Part B

Business Question:
Given the patient health data, what factors are most indicative of the likelihood of a heart-related condition?

Background:
The dataset contains information about various health indicators for patients, such as age, sex, chest pain type, blood pressure, cholesterol levels, and other relevant factors. Understanding the relationships and patterns within this data can provide valuable insights into the factors that may contribute to or indicate a higher risk of heart-related conditions.

Thought Process:

Identification of Target Variable: The 'num' column appears to be the predicted attribute or target variable, possibly indicating the presence or absence of a heart-related condition.

Exploratory Data Analysis (EDA): By conducting exploratory data analysis, we can identify variables that show significant patterns or correlations with the target variable. For example, examining the distribution of target values based on age, gender, chest pain type, etc.

Feature Importance: Utilizing statistical techniques or machine learning models, we can assess the importance of each feature in predicting the likelihood of a heart-related condition. This helps in identifying key indicators.

Decision Support for Healthcare Providers: The insights gained from the analysis can assist healthcare providers in making informed decisions. For instance, they can prioritize certain risk factors during patient assessments, recommend preventive measures for individuals with specific characteristics, or tailor interventions based on identified patterns.

Part C

The outcome variable is "target": it represents the likelhood of a heart related condition. It is a binary variable where 1 represents the "presence of the condition" and 0 represents the "absence of the condition."

Link with Business Question:

The business question aims to understand the factors that are most indicative of likelihood of the heart related condition.
In this context, the 'target' variable becomes the key outcome that we are trying to predict and analyze. By exploring the relationships between 'target' and other variables in the dataset. We can identify patterns and factors contributing to the presence or absence of heart-related conditions.

Part D

Prediction/modeling method:

To decide on a prediction method for the business question of predicting the likelihood of heart-related conditions based on a 'target' variable, we need to consider the nature of the outcome variable and the characteristics of the dataset.
In our case, the outcome variabe is binary or the (presence or absence of a heart-related condition), thus two common modeling methods, can be used, which are logistic regression and decision tree.

Prediction method: Logistic Regression:

Reasoning:

1-Binary Classification: Since the outcome variable('target') is binary (indicating the presence or absence of a heart-related condition), logistic regression is well-suited for binary classification problems.

2-Interpretability: Logistic regression provides interpretable results, making it easier to understand and explain the relationship between the independent and the likelhood of the outcome.

3-Assumption of Linearity: Logistic regression assumes a linear relationship between the independent variable and the log-odds of the outcome. If the relationships are expected to be roughly linear, logistic regression can be effective.

4-Probability Estimation: Logistic regression models provide probabilities, allowing for a clear interpretation of the likelihood of a particular outcome.

Consideration for decision tree:

While decision trees are also a valid choice for classification problems, logistic regression is often preferred in cases where interpretability and understanding the impact of individual features are crucial.

Decision Trees may be more suitable when the relationships between features and the outcome are non-linear or when feature interaction are complex.

Final decision:

Given the nature of the problem, and the binary classification task, and the interpretability requirement, we will use logistic regression as the predictive modeling method for this project. The next steps will be conducting exploratory data analysis, feature engineering, and building and evaluating a logistic regression model to predict the likelihood of heart related conditions.