```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files

# Upload the dataset
uploaded = files.upload()

# Load the dataset
df = pd.read_csv('heart.csv')  # Adjust the filename if different
```

Choose files heart.csv

- **heart.csv**(text/csv) - 38114 bytes, last modified: 22/08/2024 - 100% done

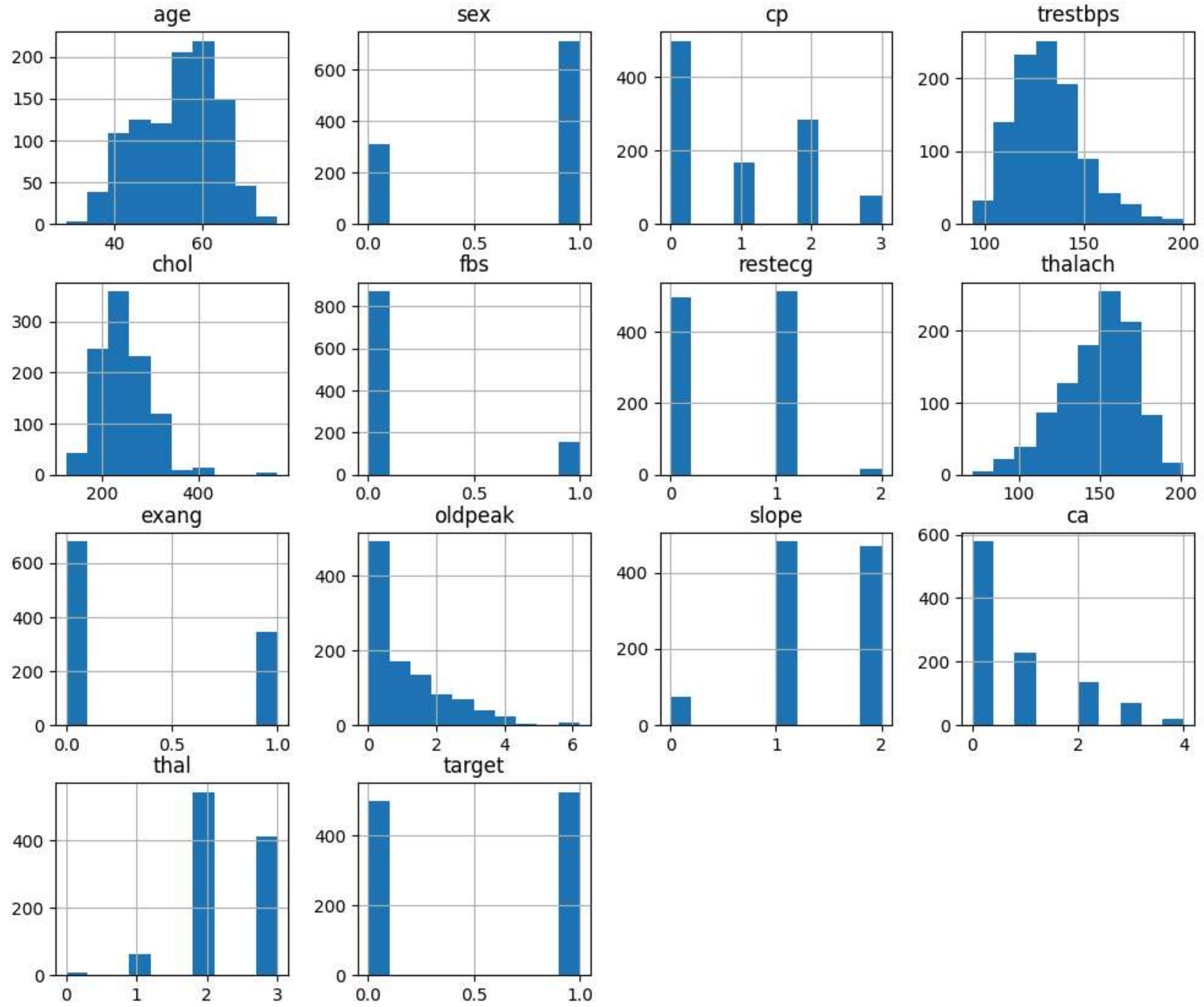
```
Saving heart.csv to heart (8).csv
```

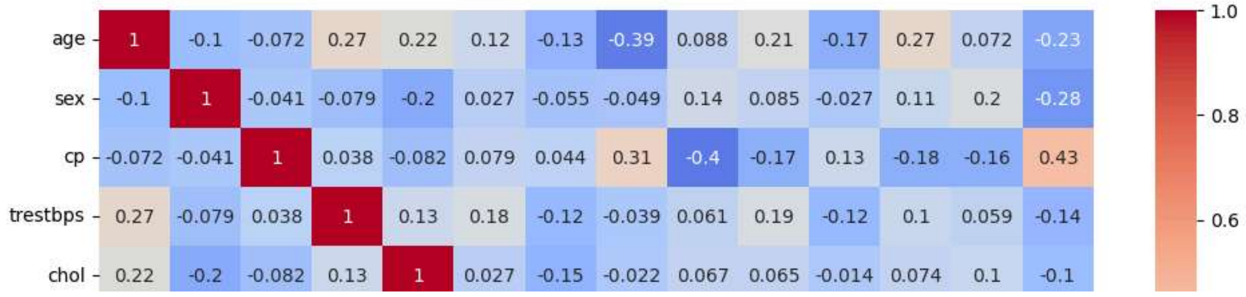
```
import numpy as np  # Make sure NumPy is imported

# Drop rows with missing values
df.dropna(inplace=True)

# Handle outliers (example: removing rows with cholesterol beyond 3 standard deviations)
if 'cholesterol' in df.columns:
    df = df[(np.abs(df['cholesterol'] - df['cholesterol'].mean()) <= (3 * df['cholesterol'].std()))]
else:
    print("The 'cholesterol' column does not exist in the dataframe.")

# Check the first few rows after cleaning
print(df.head())
```

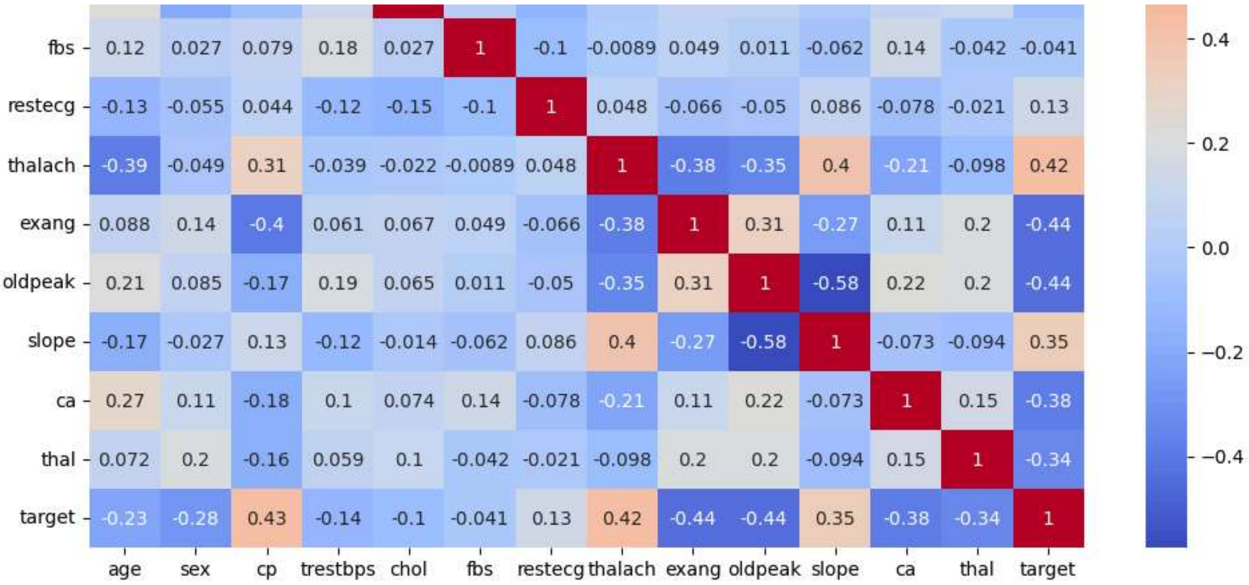
```
The 'cholesterol' column does not exist in the dataframe.
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   52    1   0       125   212    0        1      168      0      1.0      2
1   53    1   0       140   203    1        0      155      1      3.1      0
2   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
0   2     3       0
1   0     3       0
2   0     3       0
3   1     3       0
4   3     2       0
```

```
# Summary statistics
print(df.describe())

# Distribution of data (Histograms)
df.hist(figsize=(12, 10))
plt.show()

# Correlation matrix
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.show()
```

```
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
```







```
# Check if the dataframe is loaded correctly and inspect the first few rows
print("First few rows of the dataframe:")
print(df.head())

# Check if the columns 'heart_disease' and 'age' exist in the dataframe
print("\nColumn Names in DataFrame:")
print(df.columns)

# If both 'heart_disease' and 'age' columns exist, check for missing values
if 'heart_disease' in df.columns and 'age' in df.columns:
    print("\nMissing Values in 'heart_disease' and 'age' Columns:")
    print(df[['heart_disease', 'age']].isnull().sum())
else:
    print("\nError: One or both of the columns 'heart_disease' and 'age' do not exist in the dataframe.")
```

```
First few rows of the dataframe:
   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
0   2     3       0
1   0     3       0
2   0     3       0
3   1     3       0
4   3     2       0

Column Names in DataFrame:
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')

Error: One or both of the columns 'heart_disease' and 'age' do not exist in the dataframe.
```

```
# Check if the dataframe is loaded correctly and inspect the first few rows
print("First few rows of the dataframe:")
print(df.head())

# Check if the columns 'heart_disease' and 'age' exist in the dataframe
print("\nColumn Names in DataFrame:")
print(df.columns)

# If both 'heart_disease' and 'age' columns exist, check for missing values
if 'heart_disease' in df.columns and 'age' in df.columns:
    print("\nMissing Values in 'heart_disease' and 'age' Columns:")
    print(df[['heart_disease', 'age']].isnull().sum())

    # Pairplot to visualize relationships between variables
    sns.pairplot(df, hue='heart_disease')
    plt.show()

    # Heatmap of correlation matrix with a focus on variables related to heart disease
    plt.figure(figsize=(10, 6))
    sns.heatmap(df.corr(), annot=True, cmap='coolwarm', center=0)
    plt.title('Correlation Heatmap')
```