

▼ Import Libraries

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

▼ Data Preview

```
df = pd.read_csv("heart.csv")
df.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

▼ Part 1: Handle Missing Values

Check Missing Values

```
print("Missing values:\n", df.isnull().sum())
```

```
Missing values:
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
```

Result: There is no missing values in this dataset.

▼ Part 2: Handle Outliers

```
def remove_outliers_iqr(data, columns):
    for col in columns:
        Q1 = data[col].quantile(0.25)
        Q3 = data[col].quantile(0.75)
        IQR = Q3 - Q1
        lower = Q1 - 1.5 * IQR
        upper = Q3 + 1.5 * IQR
        data = data[(data[col] >= lower) & (data[col] <= upper)]
    return data

numeric_cols = df.select_dtypes(include=['float64', 'int64']).columns
df_clean = remove_outliers_iqr(df, numeric_cols)

print(f"Shape before removing outliers: {df.shape}")
print(f"Shape after removing outliers: {df_clean.shape}")
```

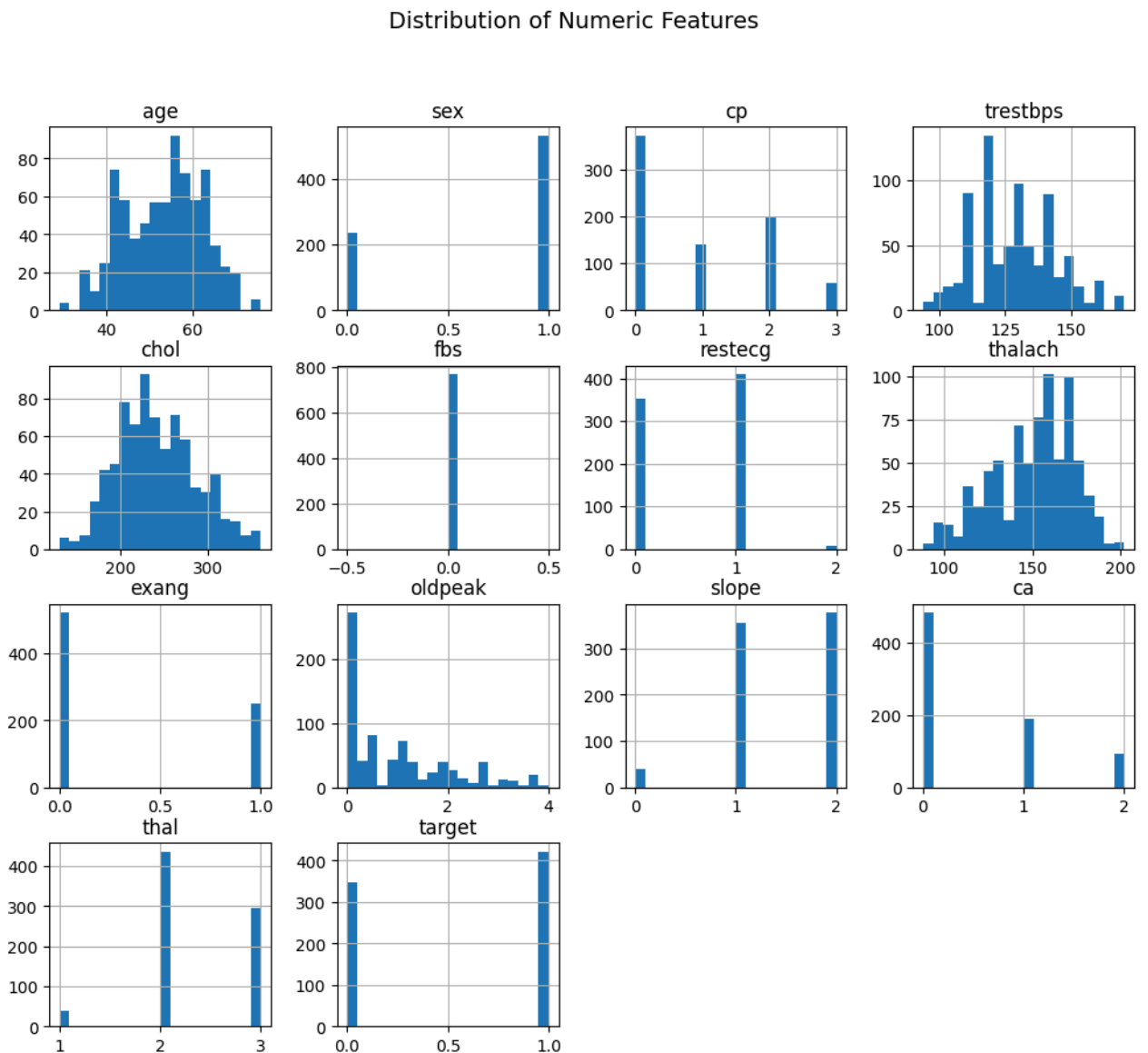
```
Shape before removing outliers: (1025, 14)
Shape after removing outliers: (769, 14)
```

Result: There is no remaining outliers values in this dataset.

Part 3: Perform Data Visualization

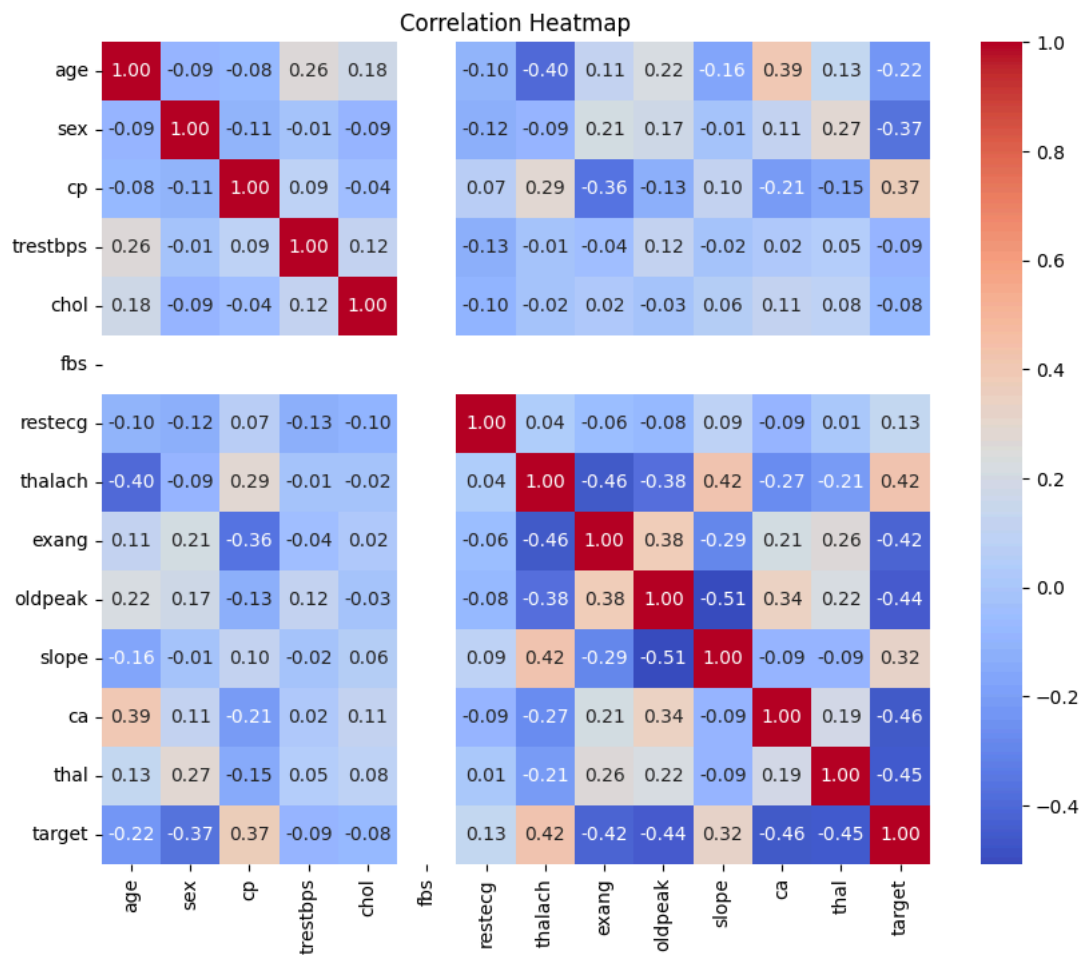
Histogram plot Visualization

```
# Histogram for numeric columns
df_clean.hist(figsize=(12, 10), bins=20)
plt.suptitle("Distribution of Numeric Features", fontsize=14)
plt.show()
```



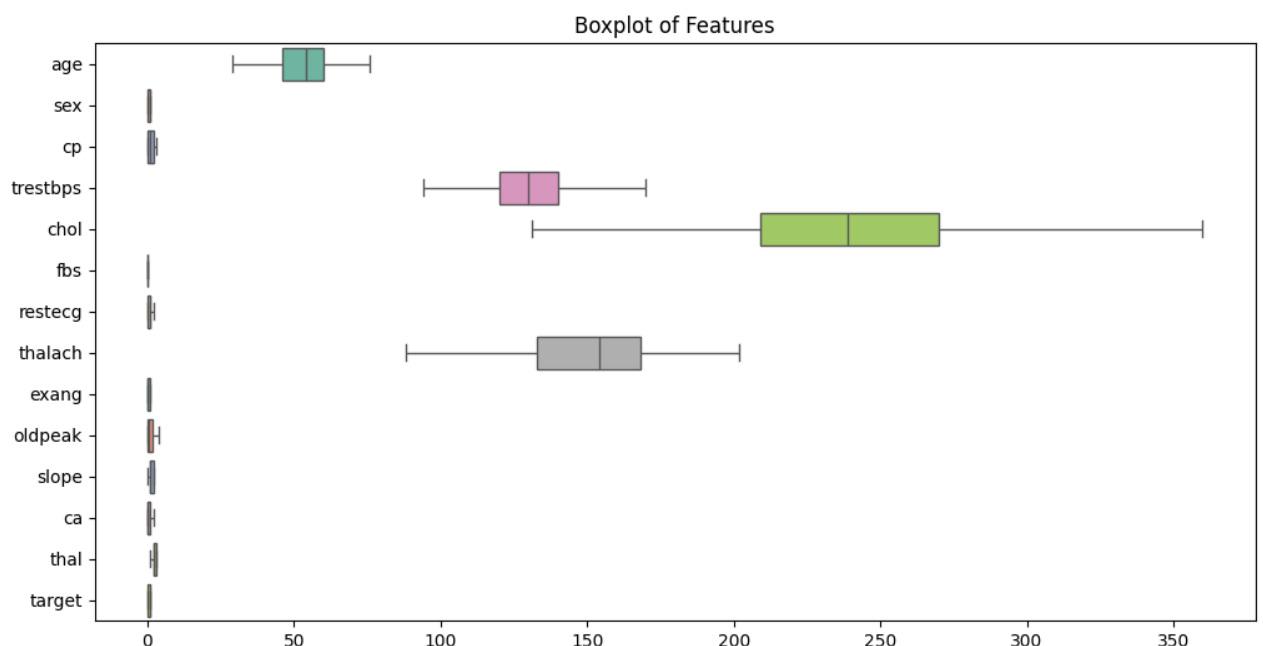
Correlation heatmap Visualization

```
# Correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(df_clean.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```



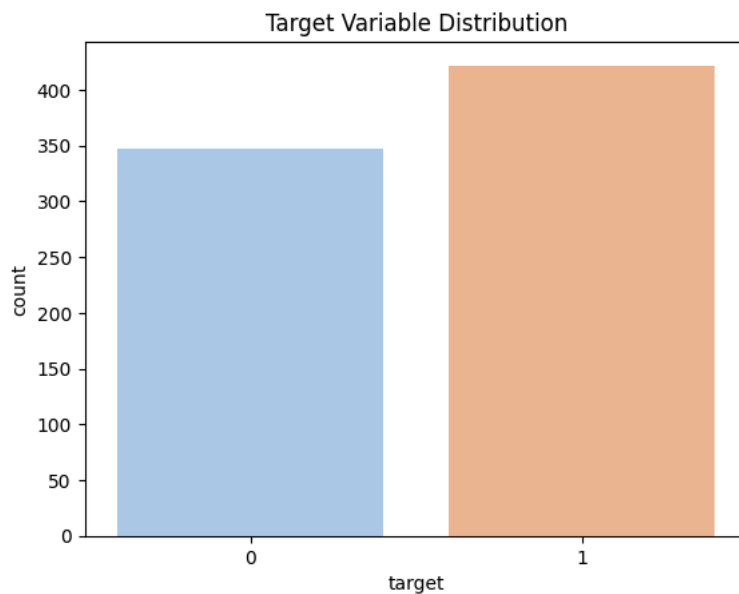
Boxplot Visualization

```
# Boxplots to visualize outliers
plt.figure(figsize=(12, 6))
sns.boxplot(data=df_clean, orient="h", palette="Set2")
plt.title("Boxplot of Features")
plt.show()
```



▼ Target Distribution Plot

```
# Target distribution plot
if 'target' in df_clean.columns:
    sns.countplot(x='target', hue='target', data=df_clean, palette='pastel', legend=False)
    plt.title("Target Variable Distribution")
    plt.show()
```



Result: These are different types of visualizations in this dataset.

✅ All preprocessing steps completed successfully — missing values handled, outliers removed, and visualizations generated gracefully.