Heart Disease Dataset

After cleaning the dataset, various attributes will be plotted to show distribution and trends.

This dataset is from 1988, containing 1025 rows and 14 features.

The features are:
Age, sex, **CP** (chest pain type, 4 values), **trestbps** (resting blood pressure), **chol** (serum cholestoral in mg/dl), **fbs** (fasting blood sugar > 120 mg/dl), **restecg** (resting electrocardiographic results, values 0,1,2), **thalach** (maximum heart rate achieved), **exang** (exercise induced angina), **oldpeak** (oldpeak = ST depression induced by exercise relative to rest), **slope** (the slope of the peak exercise ST segment), **ca** (number of major vessels, 0-3 colored by flourosopy), **thal** (0 = normal; 1 = fixed defect; 2 = reversable defect), and **target**.

The dataset can be found at: [https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset]

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
data = pd.read_csv('heart.csv')
```

In [5]:
```
data.head()
```

Out[5]:

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

In [7]:
```
data.shape
```

Out[7]: (1025, 14)

In [16]:
```
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
```

In [20]:
```
#data.isnull().sum()
```

In [24]:
```
data.describe().T
```

Out[24]:

	count	mean	std	min	25%	50%	75%	max
age	1025.0	54.434146	9.072290	29.0	48.0	56.0	61.0	77.0
sex	1025.0	0.695610	0.460373	0.0	0.0	1.0	1.0	1.0
cp	1025.0	0.942439	1.029641	0.0	0.0	1.0	2.0	3.0
trestbps	1025.0	131.611707	17.516718	94.0	120.0	130.0	140.0	200.0
chol	1025.0	246.000000	51.592510	126.0	211.0	240.0	275.0	564.0
fbs	1025.0	0.149268	0.356527	0.0	0.0	0.0	0.0	1.0
restecg	1025.0	0.529756	0.527878	0.0	0.0	1.0	1.0	2.0
thalach	1025.0	149.114146	23.005724	71.0	132.0	152.0	166.0	202.0
exang	1025.0	0.336585	0.472772	0.0	0.0	0.0	1.0	1.0
oldpeak	1025.0	1.071512	1.175053	0.0	0.0	0.8	1.8	6.2
slope	1025.0	1.385366	0.617755	0.0	1.0	1.0	2.0	2.0
ca	1025.0	0.754146	1.030798	0.0	0.0	0.0	1.0	4.0
thal	1025.0	2.323902	0.620660	0.0	2.0	2.0	3.0	3.0
target	1025.0	0.513171	0.500070	0.0	0.0	1.0	1.0	1.0

In [26]:
```
data.duplicated().sum()
```

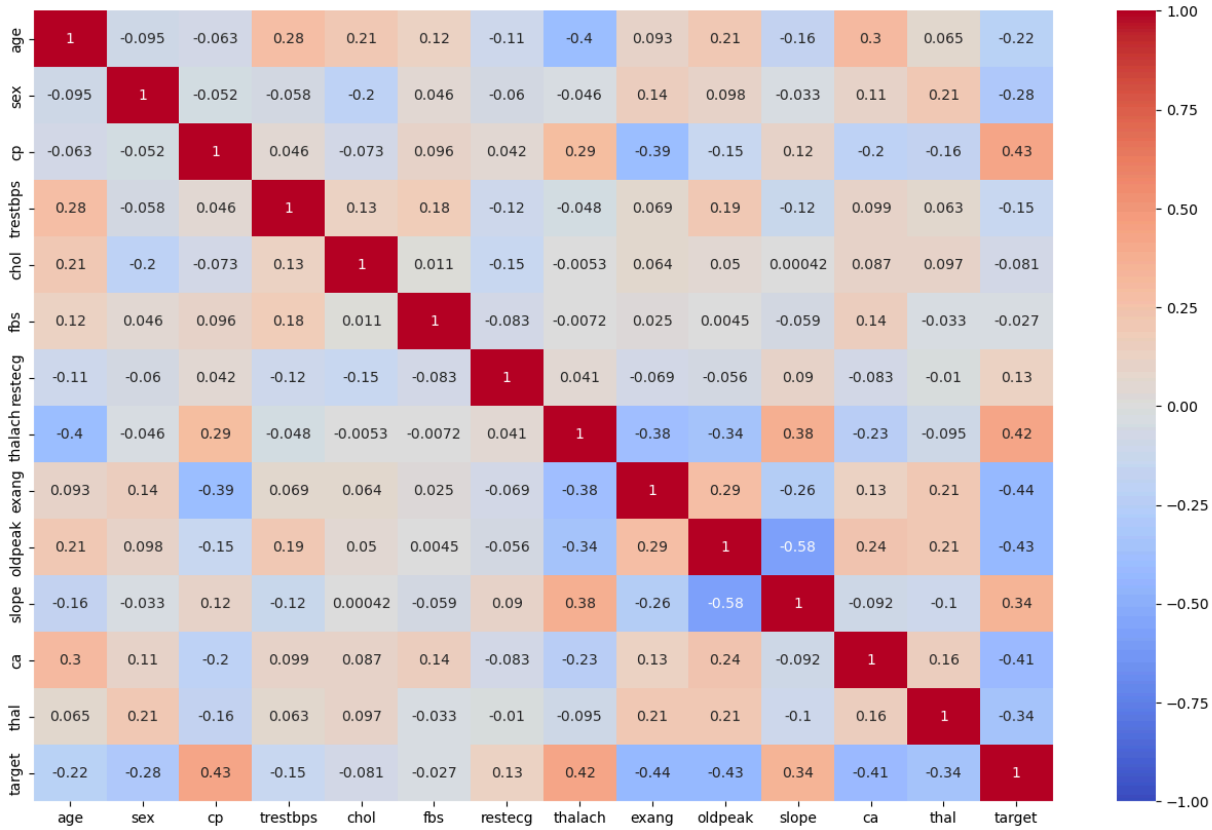
Out[26]: 723

In [28]:
```
data = data.drop_duplicates()
data.shape
```

Out[28]: (302, 14)

View Correlation Matrix

In [37]:
```
plt.figure(figsize=(16,10))
sns.heatmap(data.corr(), cmap='coolwarm', annot=True, vmin=-1, vmax=1)
plt.show()
```



Insights from Correlation Matrix

- Chest pain, maximum heart rate, and slope have high correlation with the target.
- Exercise induced angina, oldpeak, ca, and thal negatively correlate wiht the target.

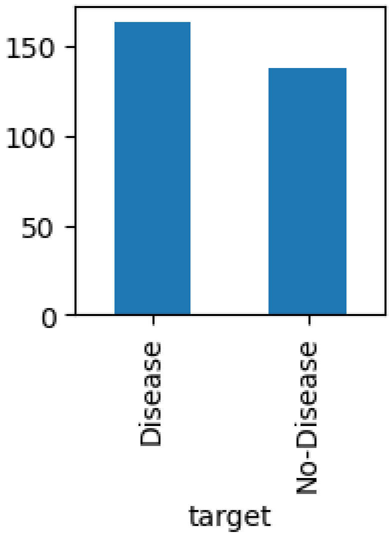
Is the dataset balanced in terms of positive and negative labels?

In [50]:
```
data.target.value_counts(normalize=True)*100
```

Out[50]:
```
target
1    54.304636
0    45.695364
Name: proportion, dtype: float64
```

In [118…

```
data.target.value_counts().plot(kind='bar', figsize=(2,2))
plt.xticks([0,1], ['Disease', 'No-Disease'])
plt.xlabel('target');
```



Do male or female have more heart disease?

In [81]:

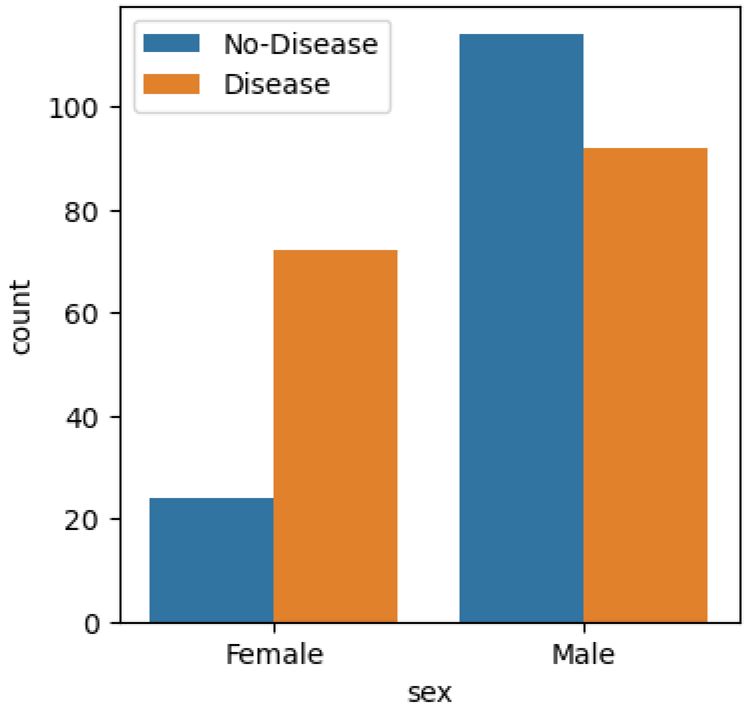
```
data.sex.value_counts()
```

Out[81]:

```
sex
1    206
0     96
Name: count, dtype: int64
```

In [87]:

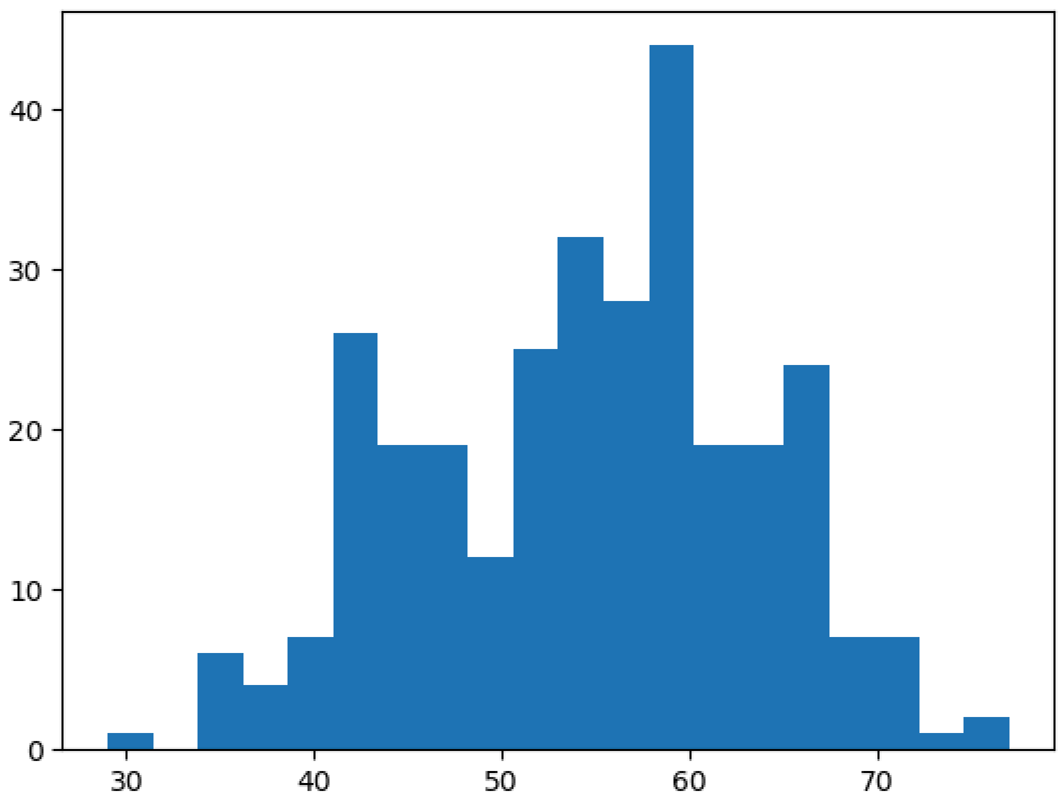
```
plt.figure(figsize=(4,4))
sns.countplot(x='sex',hue='target',data=data)
plt.xticks([1,0],['Male','Female'])
plt.legend(labels=['No-Disease','Disease'])
plt.show()
```



Check age distribution.

In [102…

```
plt.hist(data['age'], bins=20);
```



In [ ]:

```
#sns.distplot(data['age'], bins=20)
#plt.show()
```

Check chest pain type:

0: typeical angina

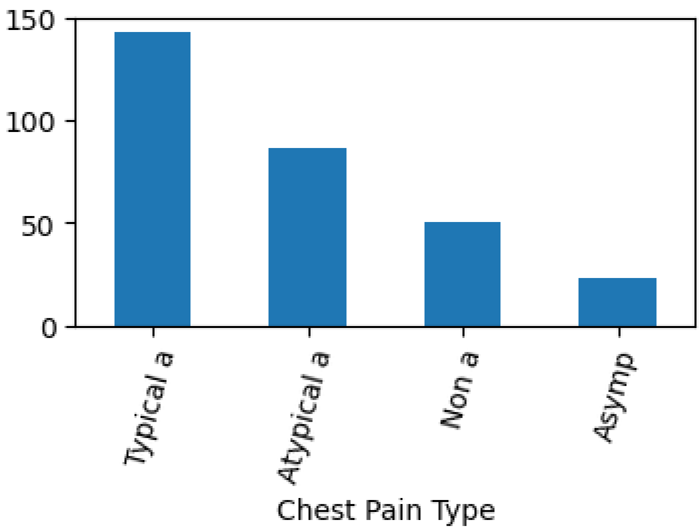
1: atypical angina

2: non-anginal pain

3: asymptomatic

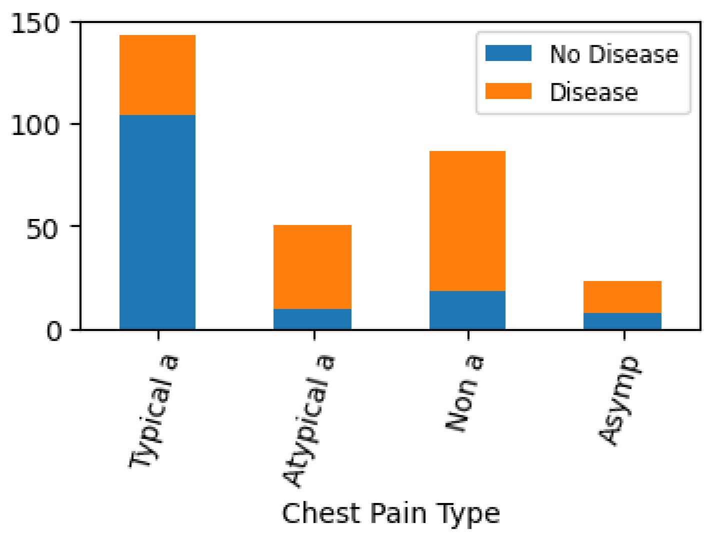
In [114…

```
data['cp'].value_counts().plot(kind= 'bar', figsize=(4,2))
plt.xticks([0,1,2,3], ['Typical a', 'Atypical a', 'Non a', 'Asymp'])
plt.xticks(rotation=75)
plt.xlabel('Chest Pain Type')
plt.show()
```



In [173…

```
pain_disease = data.groupby('cp')['target'].value_counts().unstack()

pain_disease.plot(kind='bar', stacked=True, figsize=(4,2))
plt.xticks([0,1,2,3], ['Typical a', 'Atypical a', 'Non a', 'Asymp'])
plt.xticks(rotation=75)
plt.xlabel('Chest Pain Type')
plt.legend(labels=['No Disease', 'Disease'], fontsize='small')
plt.show()
```



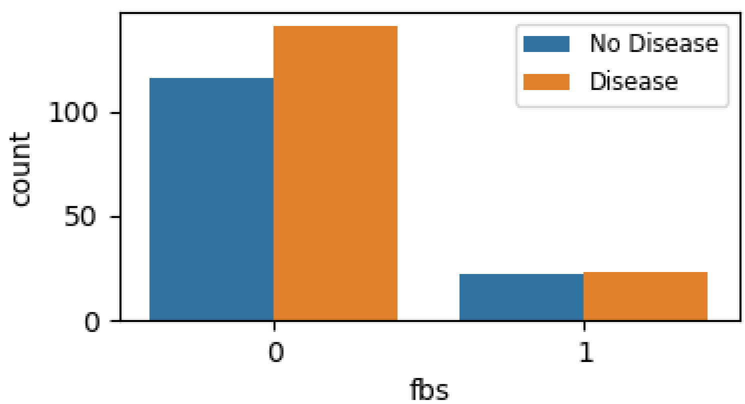
In [146…

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

Show fasting blood sugar distribution according to target variable

In [175…

```
plt.figure(figsize=(4,2))
sns.countplot(x='fbs', hue='target', data=data)
plt.legend(['No Disease', 'Disease'], fontsize='small')
plt.show()
```



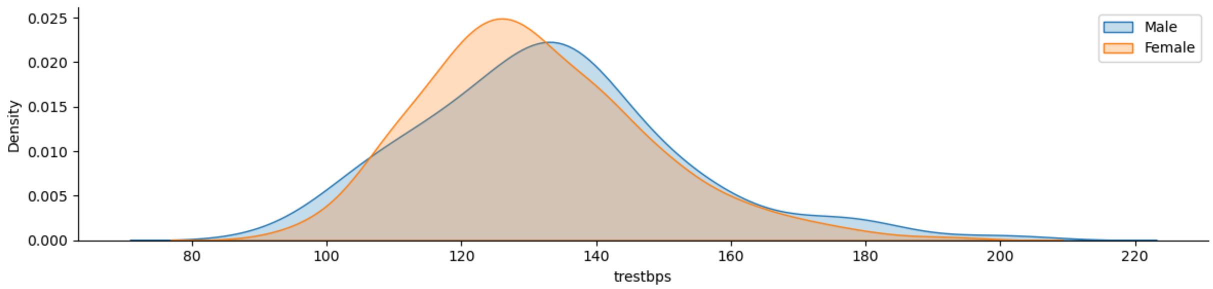
Compare resting blood pressure of the two genders.

In [184…

```
g=sns.FacetGrid(data, hue='sex', aspect=4)
g.map(sns.kdeplot, 'trestbps', shade=True)
plt.legend(labels=['Male', 'Female'])
```

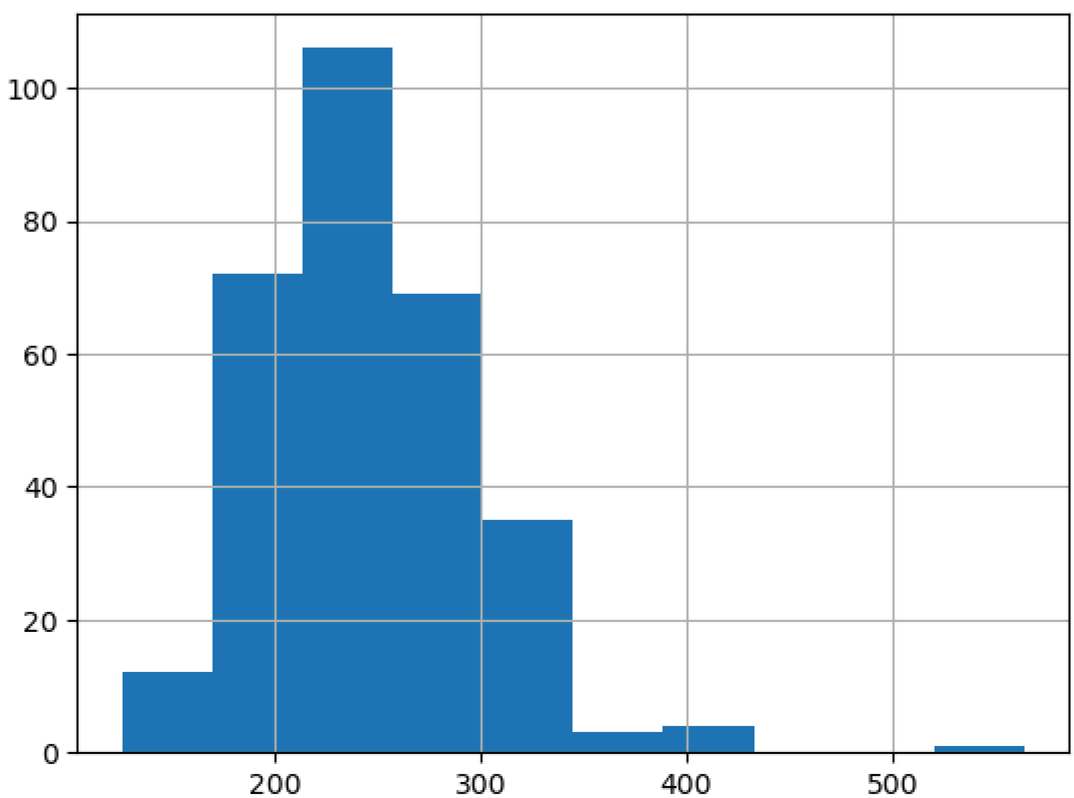
```
C:\Users\maria\anaconda3\Lib\site-packages\seaborn\axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
C:\Users\maria\anaconda3\Lib\site-packages\seaborn\axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
```

Out[184… <matplotlib.legend.Legend at 0x1e8e3a6b770>



In [186…
```
data['chol'].hist()
```

Out[186… <Axes: >



Plot Continuous variables

Instead of manually inpsecting each column, we can write the following code:

In [192…

```
categorical_v = []
continuous_v = []

for column in data.columns:
    if data[column].nunique() <= 10:
        categorical_v.append(column)
    else:
        continuous_v.append(column)
```

In [194…

```
categorical_v
```

Out[194…

```
['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal', 'target']
```

In [196…

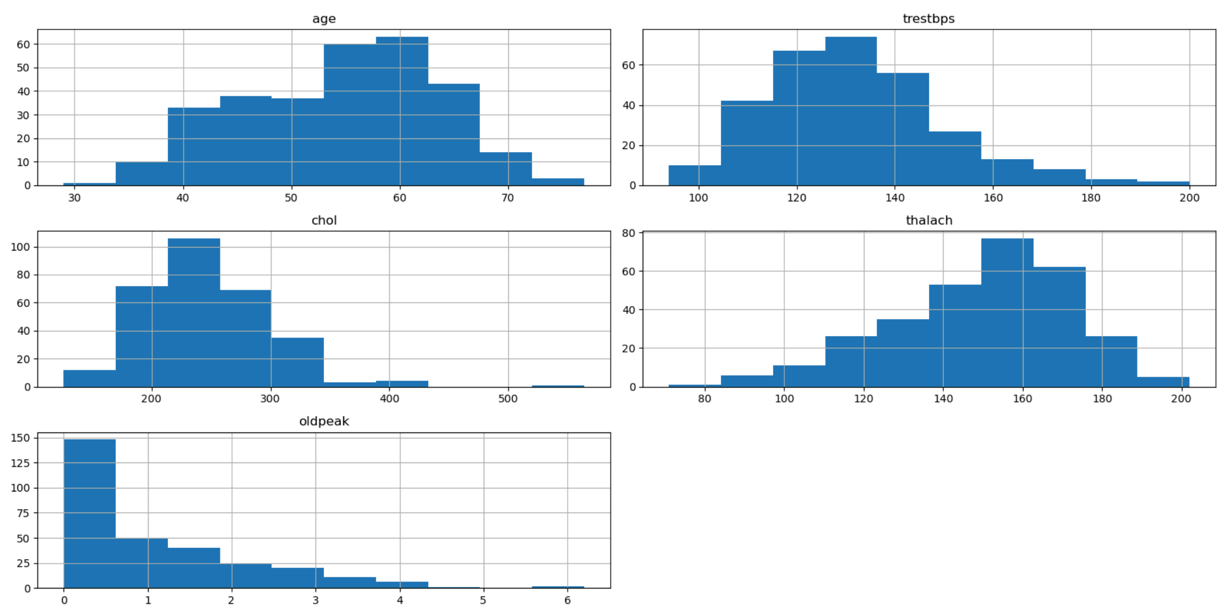
```
continuous_v
```

Out[196…

```
['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
```

In [208…

```
data.hist(continuous_v, figsize=(16,8))
#plt.tight_layout()
plt.show()
```



In [ ]: