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

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

```
df
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope
0	63	1	3	145	233	1	0	150	0	2.3	0
1	37	1	2	130	250	0	1	187	0	3.5	0
2	41	0	1	130	204	0	0	172	0	1.4	2
3	56	1	1	120	236	0	1	178	0	0.8	2
4	57	0	0	120	354	0	1	163	1	0.6	2
...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1
299	45	1	3	110	264	0	1	132	0	1.2	1
300	68	1	0	144	193	1	1	141	0	3.4	1
301	57	1	0	130	131	0	1	115	1	1.2	1
302	57	0	1	130	236	0	0	174	0	0.0	1

303 rows × 14 columns

```
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        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
```

```
     dtypes: float64(1), int64(13)
     memory usage: 33.3 KB
```

```
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          0
     thal        0
     target      0
     dtype: int64
```

```
df['target'].value_counts()
```

```
     1    165
     0    138
     Name: target, dtype: int64
```

```
df.describe()
```

	age	sex	cp	trestbps	chol	fbs	res
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.00
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.52
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.52
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.00
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.00
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.00
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.00
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.00

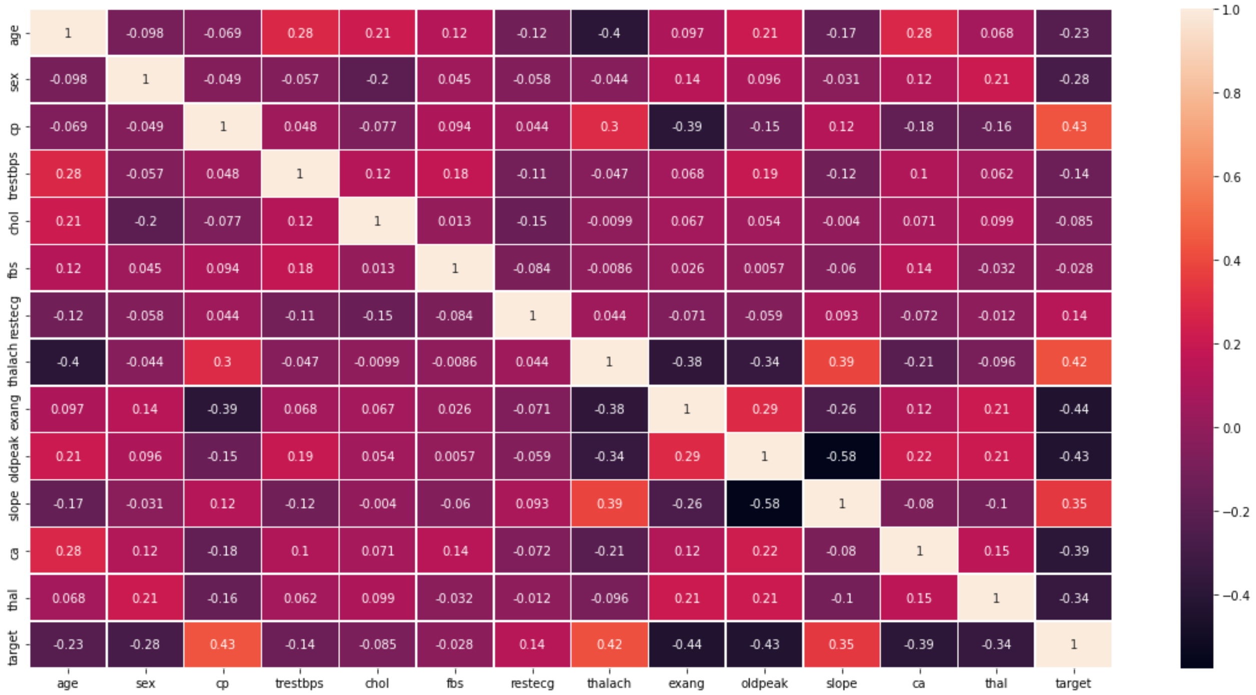
```
corr_matrix = df.corr()
corr_matrix['target'].sort_values(ascending=False)
```

```
     target      1.000000
     cp          0.433798
     thalach     0.421741
     slope       0.345877
     restecg     0.137230
     fbs        -0.028046
     chol       -0.085239
     trestbps   -0.144931
     age        -0.225439
```

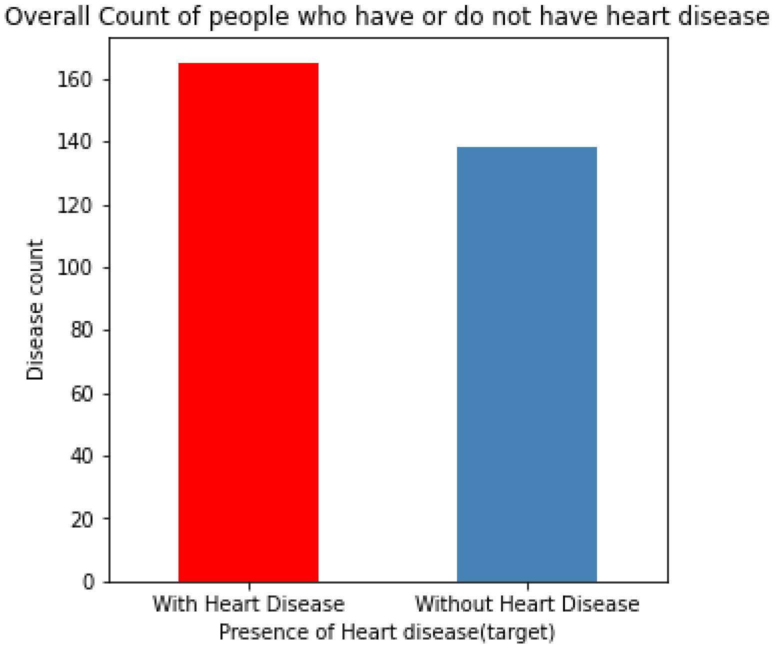
```
sex        -0.280937
thal       -0.344029
ca         -0.391724
oldpeak    -0.430696
exang      -0.436757
Name: target, dtype: float64
```

```
import seaborn as sns
corrdf = df.corr('pearson')

fig, ax = plt.subplots(figsize=(20,10))
sns.heatmap(corrdf, annot=True, linewidths=.5, ax=ax)
```

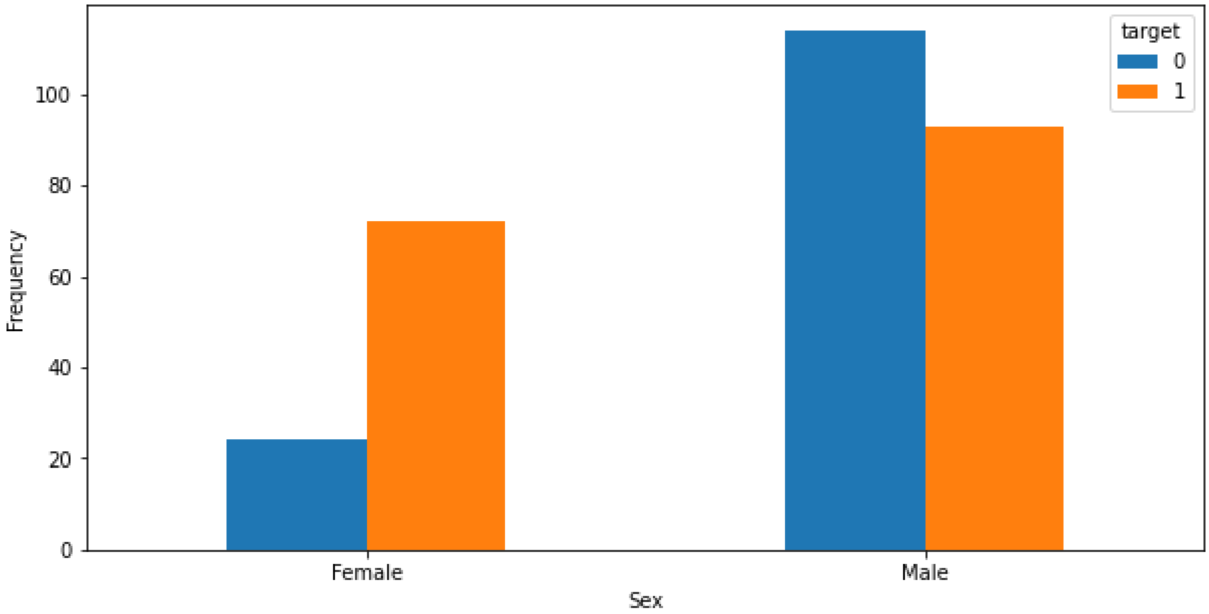
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f77a8e8a090>
```



```
disease_count = df['target'].value_counts()
disease_count.plot(kind='bar', color=['red','steelblue'],figsize=(5,5))
index = [0,1]
plt.xlabel('Presence of Heart disease(target)')
plt.ylabel('Disease count')
plt.xticks(index, ['With Heart Disease', 'Without Heart Disease'],rotation=0)
plt.title('Overall Count of people who have or do not have heart disease')
plt.show()
```

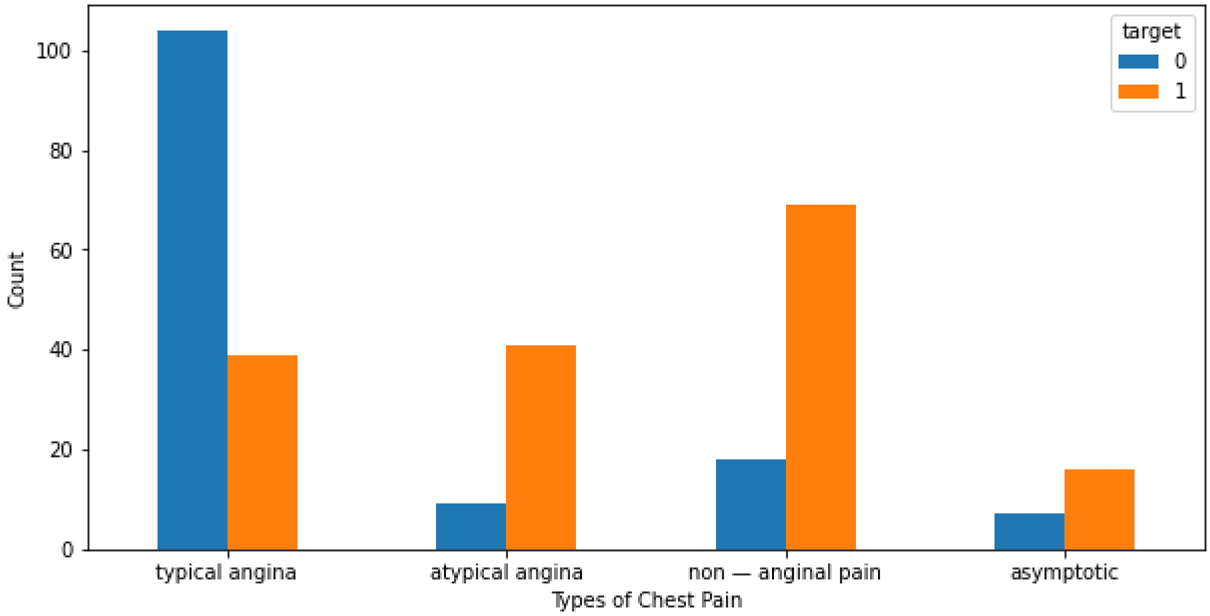


```
sex_disease = df.groupby(['sex'])['target'].value_counts()
plot = sex_disease.loc(axis=0)[:,:]
index = [0,1]
p = plot.groupby(['sex','target']).sum().unstack()
p.plot(kind='bar',figsize=(10,5))
plt.xlabel('Sex')
plt.xticks(index, ['Female', 'Male'],rotation=0)
plt.ylabel('Frequency')
plt.show()
```



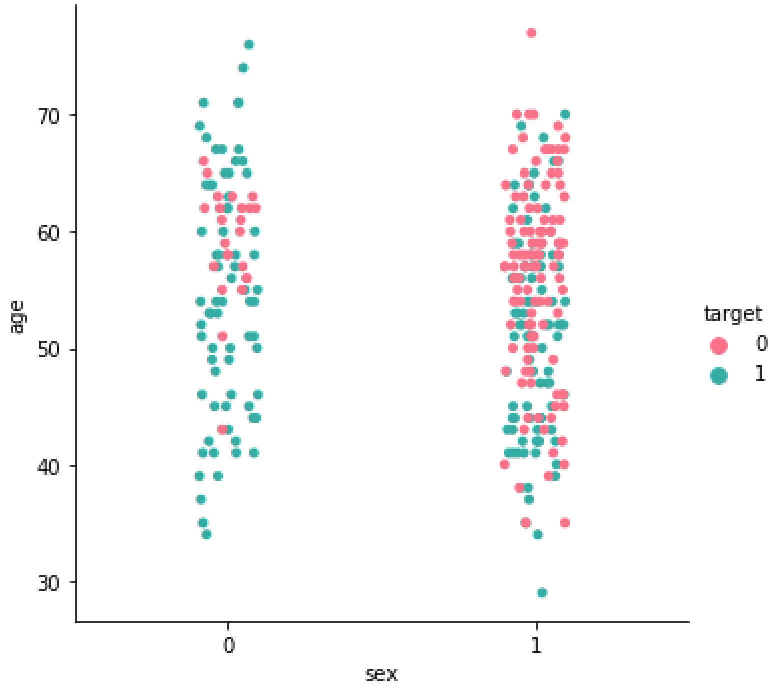
```
cp = df.groupby(['cp'])['target'].value_counts()
plot = cp.loc(axis=0)[:,:]
index = [0,1,2,3]
p = plot.groupby(['cp','target']).sum().unstack()
p.plot(kind='bar',figsize=(10,5))
```

```
plt.xlabel('Types of Chest Pain')
plt.xticks(index,['typical angina','atypical angina','non — anginal pain', 'asympto
plt.ylabel('Count')
plt.show()
```



```
sns.catplot(data=df,x='sex',y='age',hue='target',palette='husl')
```

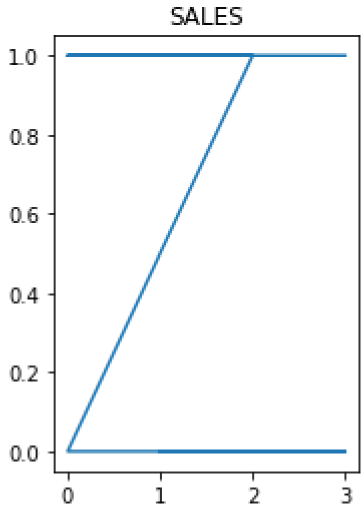
```
<seaborn.axisgrid.FacetGrid at 0x7f77a3b57590>
```



```
#plot 1:
x = df['cp']
y = df['target']

plt.subplot(1, 2, 1)
plt.plot(x,y)
plt.title("SALES")
```

```
#plot 2:
# x = np.array([0, 1, 2, 3])
# y = np.array([10, 20, 30, 40])

# plt.subplot(1, 2, 2)
# plt.plot(x,y)
# plt.title("INCOME")

# plt.show()
```

```
Text(0.5, 1.0, 'SALES')
```



```
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import  MinMaxScaler
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import StandardScaler
StandardScaler = StandardScaler()
columns_to_scale = ['age','trestbps','chol','thalach','oldpeak']
df[columns_to_scale] = StandardScaler.fit_transform(df[columns_to_scale])
```

```
X = df.drop("target", 1)
y = df["target"]
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.25, random_s
```

```
lr = LogisticRegression()
model = lr.fit(X_train,y_train)
prediction1=model.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,prediction1)
```

```
0.7631578947368421
```

● ×