

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

health = pd.read_csv("heart.csv")
health.columns=health.columns.str.strip()
```

```
health.head
```

```
950
```

```
950
```

```
health.head(5)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak
0	63	1	3	145	233	1	0	150	0	2.3
1	37	1	2	130	250	0	1	187	0	3.5
2	41	0	1	130	204	0	0	172	0	1.4
3	56	1	1	120	236	0	1	178	0	0.8
4	57	0	0	120	354	0	1	163	1	0.6

	ca	thal	target
0	0	1	1
1	0	2	1
2	0	2	1
3	0	2	1
4	0	2	1

```
health.tail(5)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak
298	57	0	0	140	241	0	1	123	1	0.2
299	45	1	3	110	264	0	1	132	0	1.2
300	68	1	0	144	193	1	1	141	0	3.4
301	57	1	0	130	131	0	1	115	1	1.2
302	57	0	1	130	236	0	0	174	0	0.0

	slope	ca	thal	target
298	1	0	3	0
299	1	0	3	0
300	1	2	3	0
301	1	1	3	0
302	1	1	2	0

health.shape

(303, 14)

```
health.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
```

```
health.isna()
```

[illegible]

```

...
298 False False False False False False False False
False
299 False False False False False False False False
False
300 False False False False False False False False
False
301 False False False False False False False False
False
302 False False False False False False False False
False

```

	oldpeak	slope	ca	thal	target
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
..
298	False	False	False	False	False
299	False	False	False	False	False
300	False	False	False	False	False
301	False	False	False	False	False
302	False	False	False	False	False

```
[303 rows x 14 columns]
```

```
health.isna().sum()
```

```

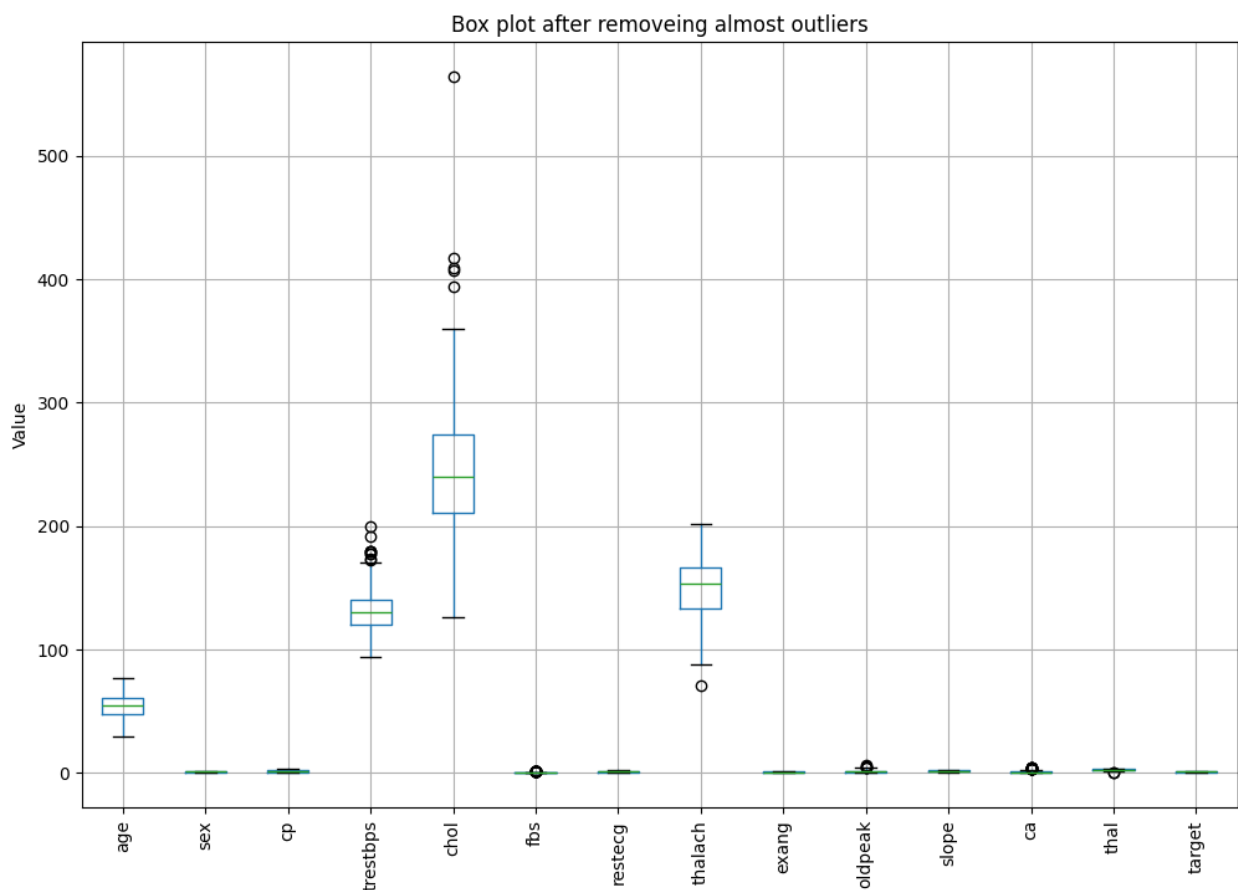
age          0
sex          0
cp           0
trestbps     0
chol         0
fbs          0
restecg      0
thalach      0
exang        0
oldpeak      0
slope        0
ca           0
thal         0
target       0
dtype: int64

```

Remove outliers

```
ZScores = stats.zscore(health)
Threshold = 3
outliers = (ZScores > Threshold) | (ZScores < -Threshold)
new_data = health[~outliers.any(axis=1)]

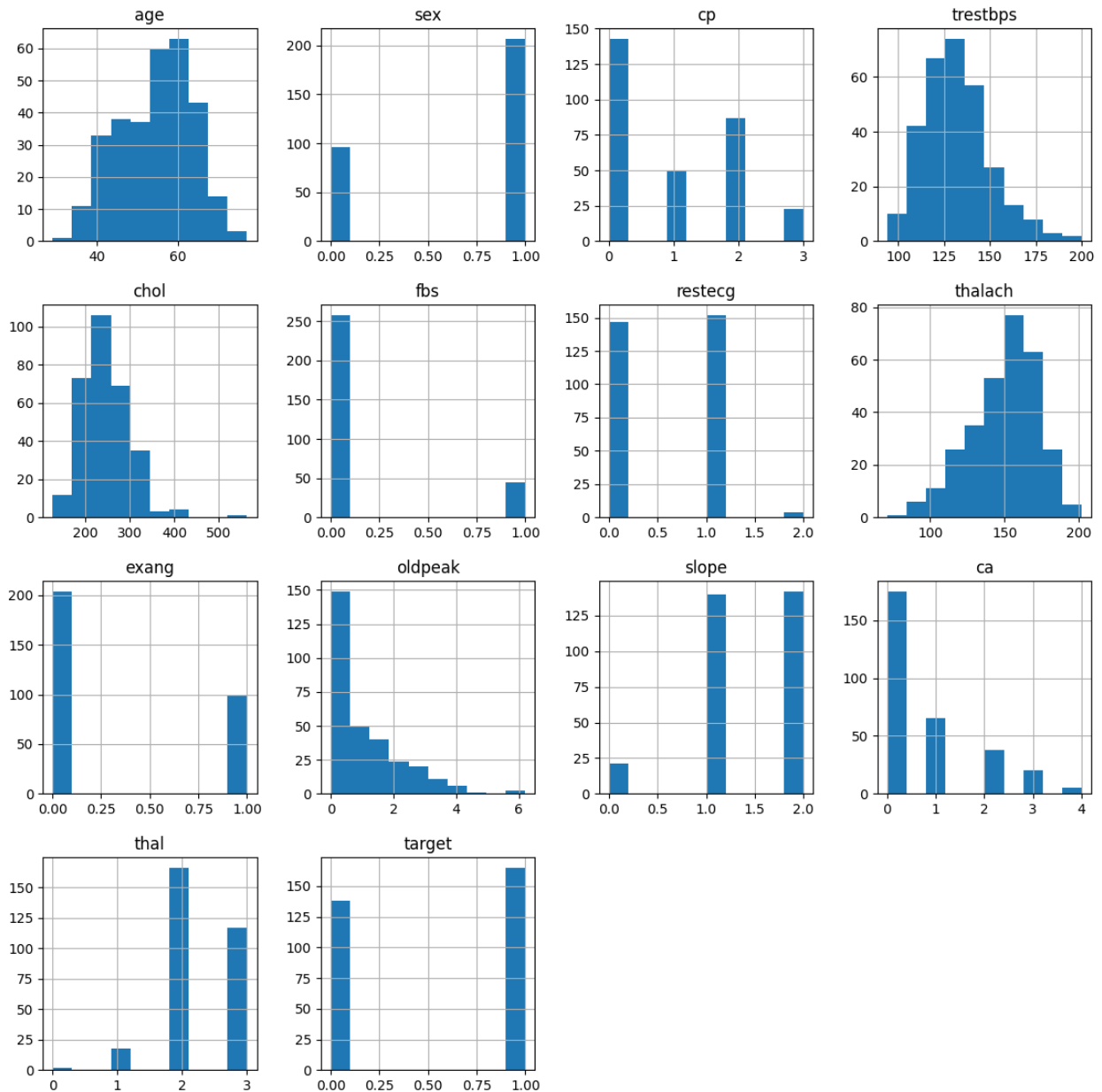
plt.figure(figsize=(12,8))
health.boxplot()
plt.title("Box plot after removeing almost outliers")
plt.ylabel('Value')
plt.xticks(rotation=90)
plt.show()
```



Visualizing the features of dataset

```
health.hist(figsize=(14,14))
plt.show

<function matplotlib.pyplot.show(close=None, block=None)>
```



Patient demographics

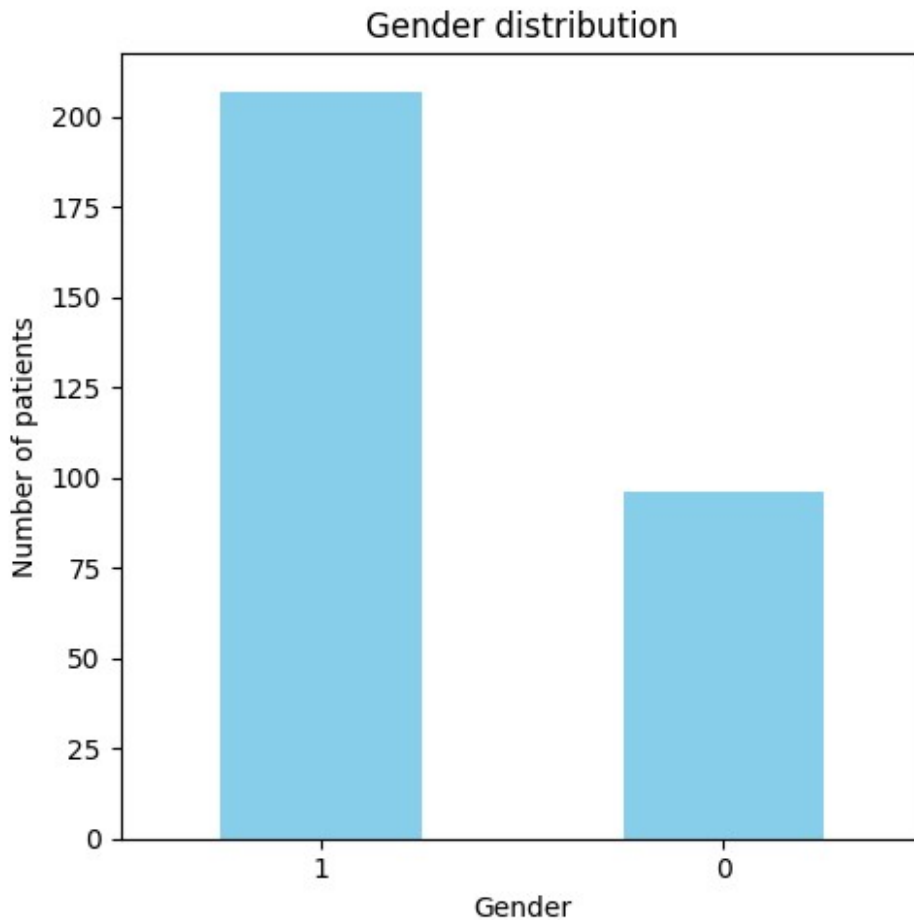
```
#count the number of patients for each gender
Gender = health['sex'].value_counts()
chestpain = health['cp'].value_counts()
heart_rate = health['exang'].value_counts()

#Plot bar plots for each demographic Category
plt.figure(figsize=(5, 5))

#bar plot for gender distribution
Gender.plot(kind='bar',color='skyblue')
plt.title('Gender distribution')
```

```
plt.xlabel('Gender')
plt.ylabel('Number of patients')
plt.xticks(rotation=0)

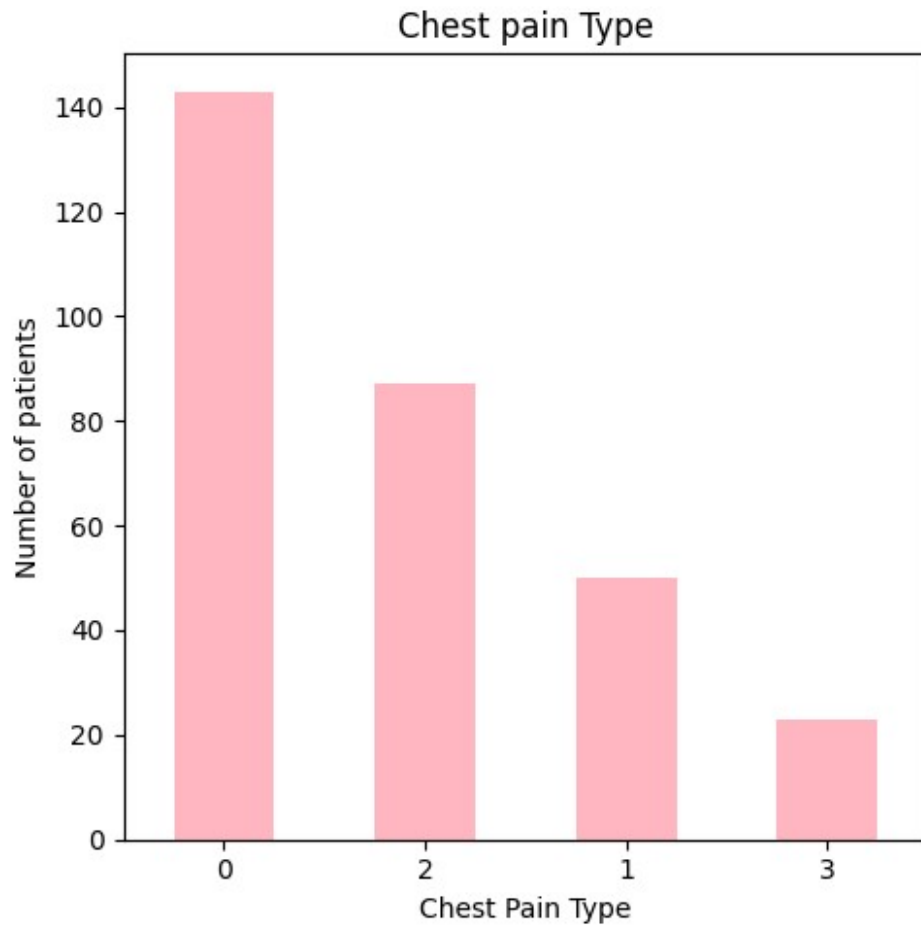
plt.tight_layout()
plt.show()
```



```
#Plot bar plots for each demographic Category
plt.figure(figsize=(5, 5))

#bar plot for Chest pain distribution
chestpain.plot(kind='bar',color='lightpink')
plt.title('Chest pain Type')
plt.xlabel('Chest Pain Type')
plt.ylabel('Number of patients')
plt.xticks(rotation=0)

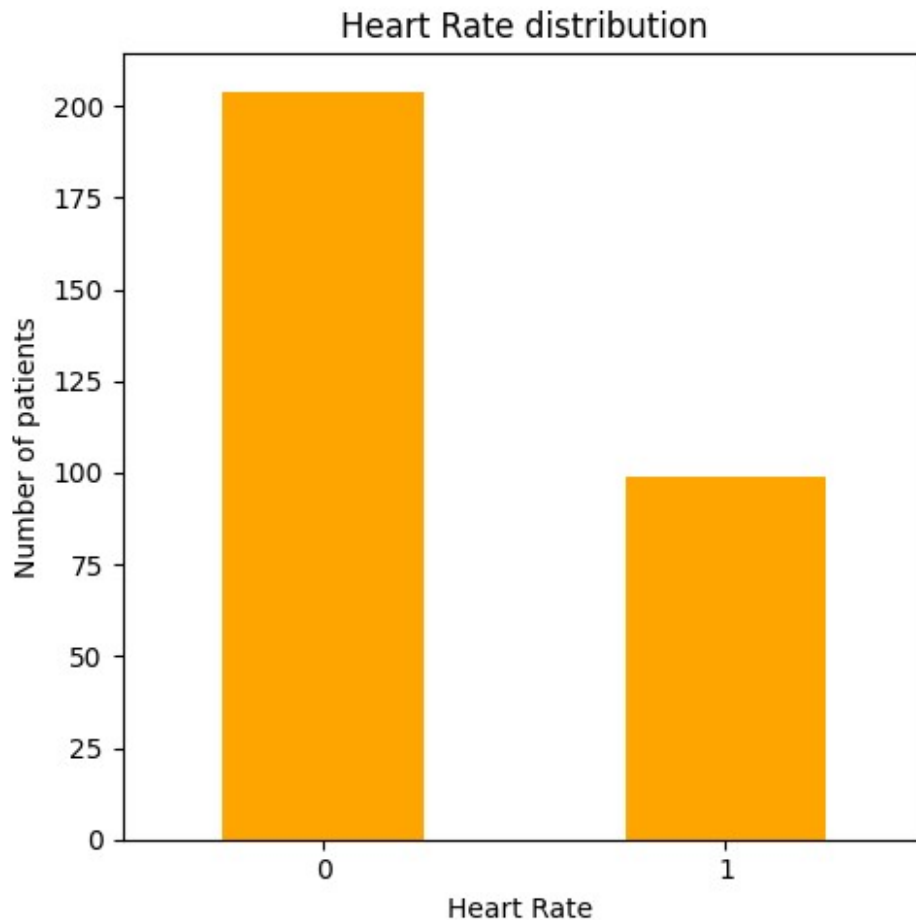
plt.tight_layout()
plt.show()
```



```
#Plot bar plots for each demographic Category
plt.figure(figsize=(5, 5))

#bar plot for Heart Rate distribution
heart_rate.plot(kind='bar',color='orange')
plt.title('Heart Rate distribution')
plt.xlabel('Heart Rate ')
plt.ylabel('Number of patients')
plt.xticks(rotation=0)

plt.tight_layout()
plt.show()
```



Analyze health metrics such as Blood Sugar, Cholesterol Levels

```
health.columns
```

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

```
#Plot histograms for cholesterol levels
```

```
plt.figure(figsize=(12,6))
```

```
#histogram for cholesterol levels
```

```
plt.subplot(1, 2, 2)
```

```
plt.hist(health['chol'], bins=20, color="skyblue")
```

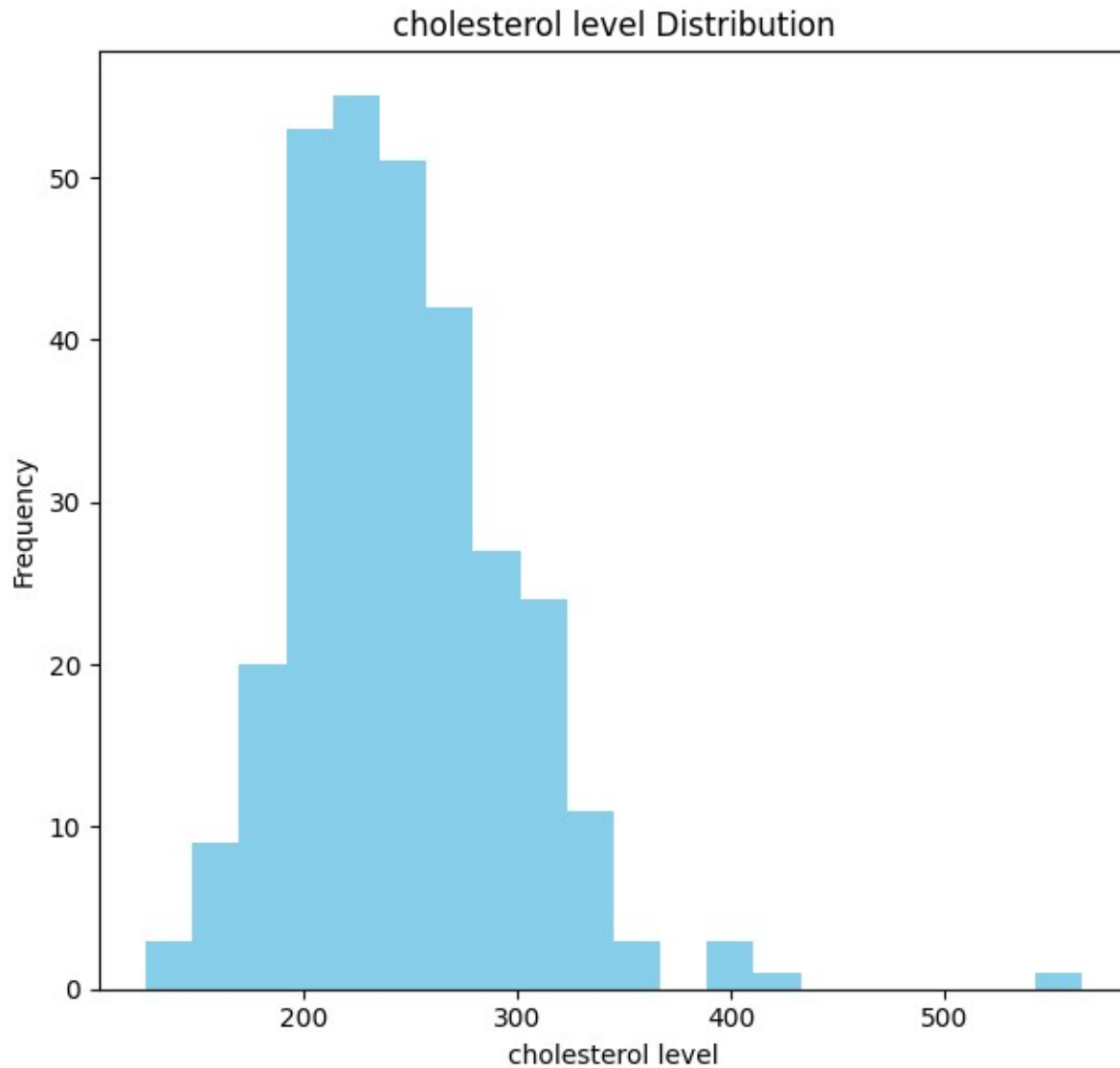
```
plt.title('cholesterol level Distribution')
```

```
plt.xlabel("cholesterol level")
```

```
plt.ylabel('Frequency')
```

```
plt.tight_layout()
```

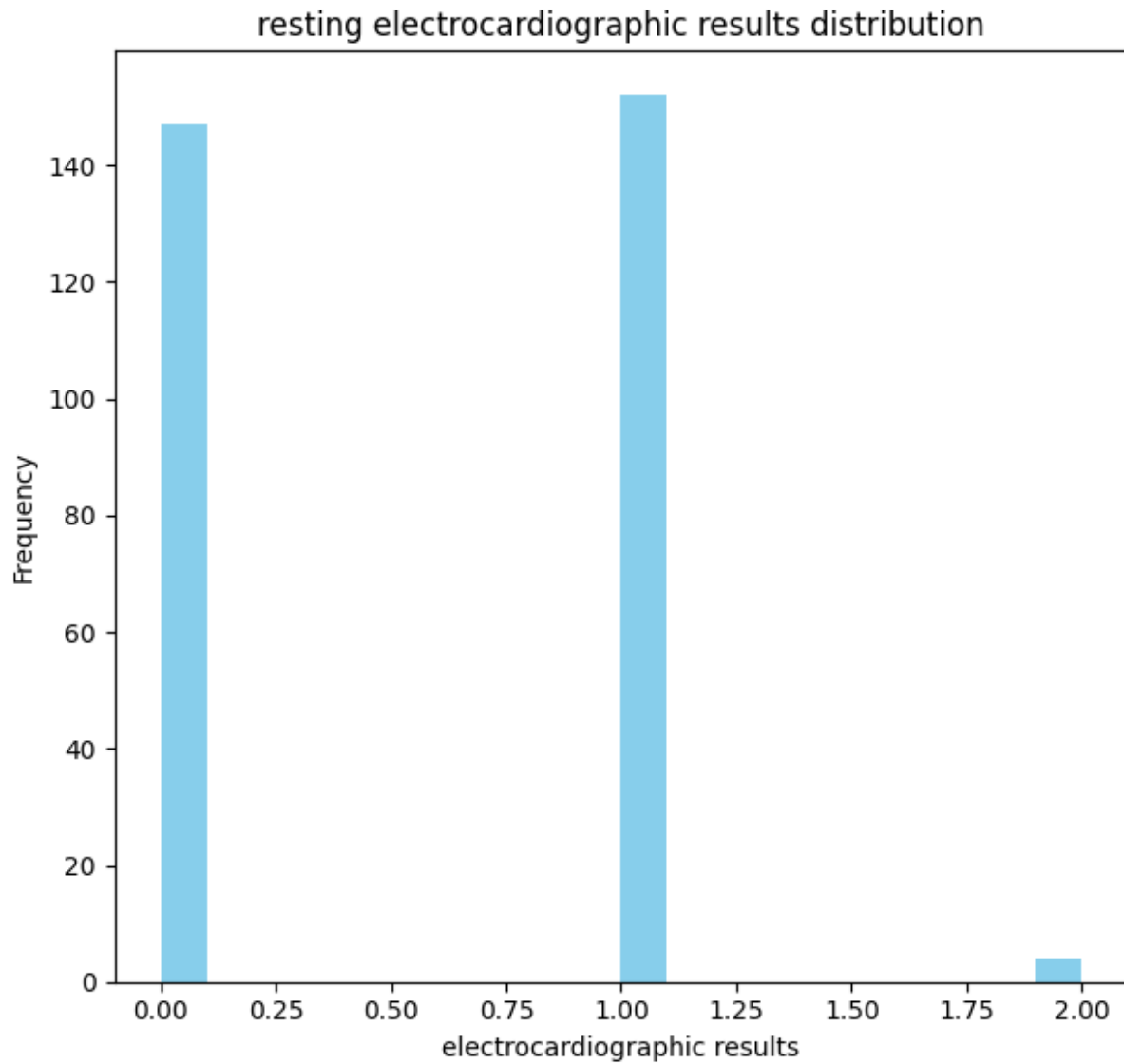
```
plt.show()
```

```
#Plot histograms for resting electrocardiographic results levels
plt.figure(figsize=(12,6))

#histogram for resting electrocardiographic results levels
plt.subplot(1, 2, 2)
plt.hist(health['restecg'], bins=20, color="skyblue")
plt.title('resting electrocardiographic results distribution')
plt.xlabel("electrocardiographic results")
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```

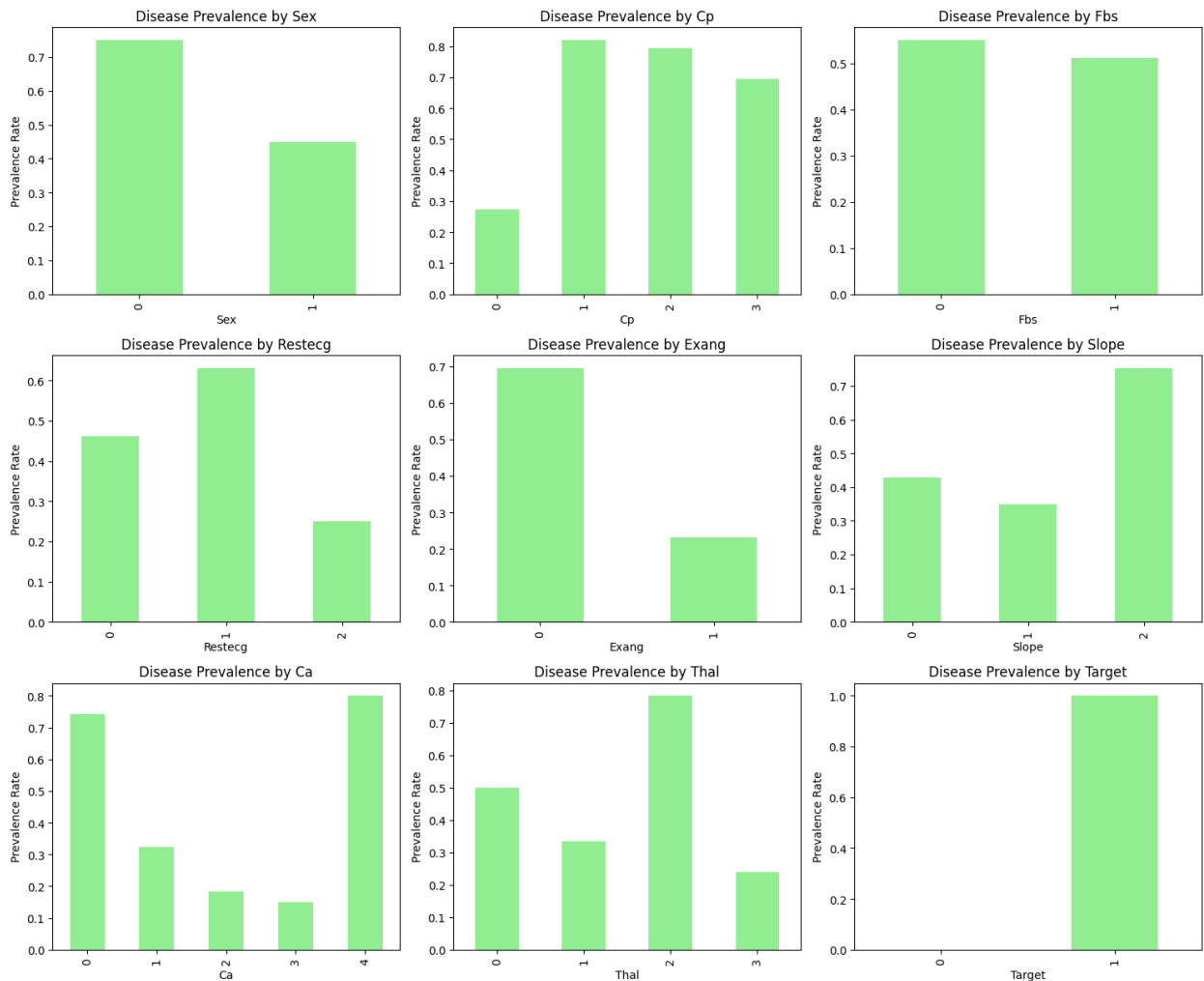


Disease Prevalence Rates

```
variables = ['sex', 'cp', 'fbs', 'restecg',  
            'exang', 'slope', 'ca', 'thal', 'target']  
  
plt.figure(figsize=(15,20))  
  
for i, var in enumerate(variables, 1):  
    plt.subplot(5, 3, i)  
    #Calculate disease prevalence rates for each category in the  
    variable  
    prevalence_rates = health.groupby(var)  
    ['target'].mean().sort_index()  
  
    #plot bar plot  
    prevalence_rates.plot(kind='bar', color='lightgreen')
```

```
plt.title(f'Disease Prevalence by {var.capitalize()}')
plt.xlabel(var.capitalize())
plt.ylabel('Prevalence Rate')
```

```
plt.tight_layout()
plt.show()
```



```
cleaned_data_filename = "cleaned_heart.csv"
health.to_csv(cleaned_data_filename, index=False)

print(f"Cleand data saved to {cleaned_data_filename}")

Cleaned data saved to cleaned_heart.csv
```