```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix


df=pd.read_csv("/content/heart.csv")


df
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
0	52	1	0	125	212	0	1	168	0	1.0	2	2
1	53	1	0	140	203	1	0	155	1	3.1	0	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1
4	62	0	0	138	294	1	1	106	0	1.9	1	3
...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0
1021	60	1	0	125	258	0	0	141	1	2.8	1	1
1022	47	1	0	110	275	0	0	118	1	1.0	1	1
1023	50	0	0	110	254	0	0	159	0	0.0	2	0
1024	54	1	0	120	188	0	1	113	0	1.4	1	1

1025 rows × 14 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.describe()
```

	age	sex	cp	trestbps	chol	fbs	
	0000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025
	4146	0.695610	0.942439	131.611707	246.00000	0.149268	0
	2290	0.460373	1.029641	17.516718	51.59251	0.356527	0
	0000	0.000000	0.000000	94.000000	126.00000	0.000000	0
	0000	0.000000	0.000000	120.000000	211.00000	0.000000	0
	0000	1.000000	1.000000	130.000000	240.00000	0.000000	1
	0000	1.000000	2.000000	140.000000	275.00000	0.000000	1
	0000	1.000000	3.000000	200.000000	564.00000	1.000000	2

```
df.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
```

```
df.shape
```

```
(1025, 14)
```

```
df.isnull().sum().sum()
```

```
np.int64(0)
```

```
df.duplicated().sum()
```

```
np.int64(723)
```

```
df[df.duplicated()]
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
15	34	0	1	118	210	0	1	192	0	0.7	2	0
31	50	0	1	120	244	0	1	162	0	1.1	2	0
43	46	1	0	120	249	0	0	144	0	0.8	2	0
55	55	1	0	140	217	0	1	111	1	5.6	0	0
61	66	0	2	146	278	0	0	152	0	0.0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0
1021	60	1	0	125	258	0	0	141	1	2.8	1	1
1022	47	1	0	110	275	0	0	118	1	1.0	1	1
1023	50	0	0	110	254	0	0	159	0	0.0	2	0
1024	54	1	0	120	188	0	1	113	0	1.4	1	1

723 rows × 14 columns

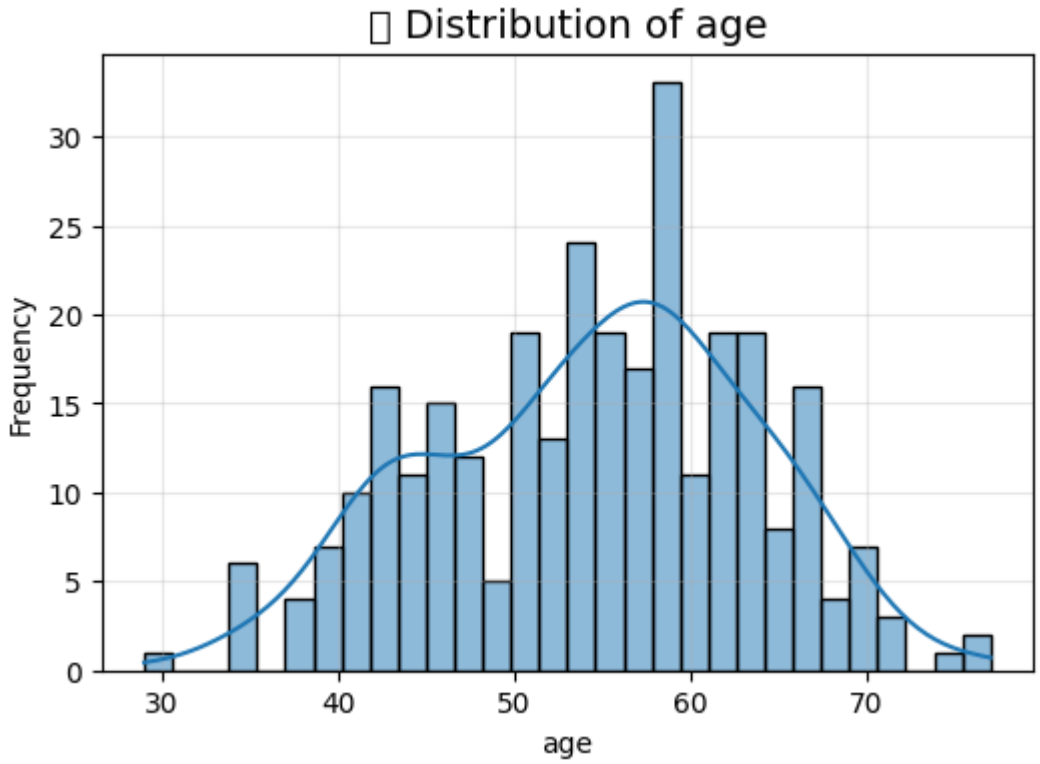
```
df = df.groupby(df.columns.tolist()).size().reset_index(name='count')
```

```
df.duplicated().sum()
```

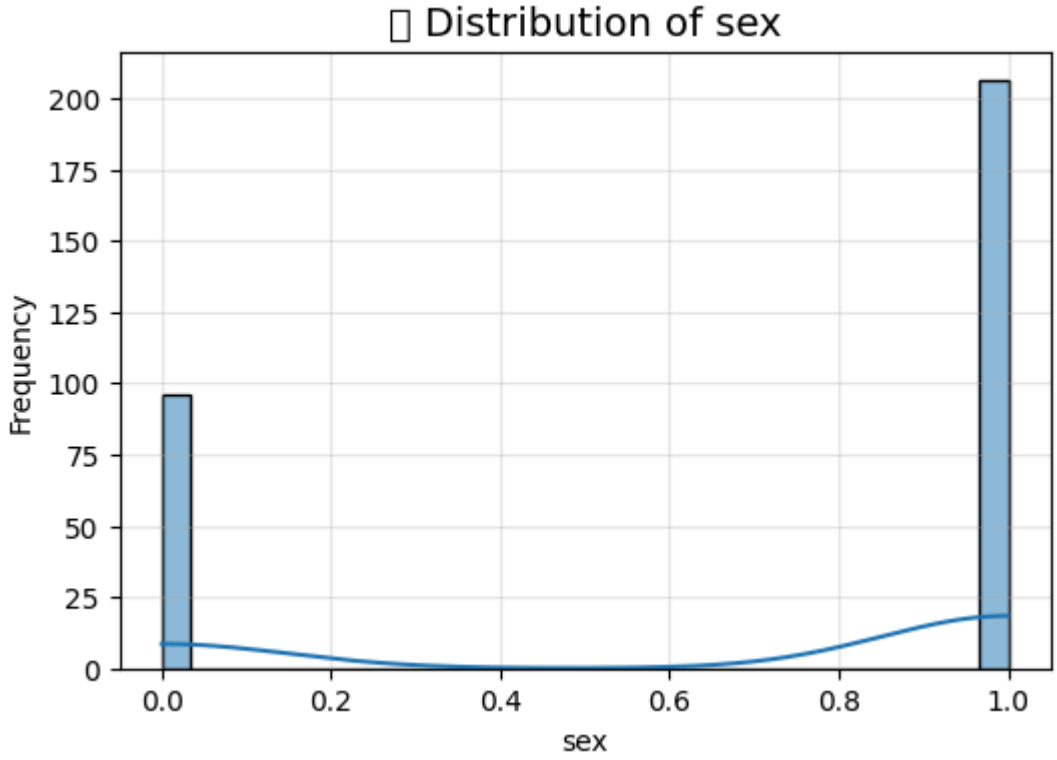
```
np.int64(0)
```

```
num_cols = df.columns.tolist()
for col in num_cols:
    plt.figure(figsize=(6, 4))
    sns.histplot(df[col], kde=True, bins=30)
    plt.title(f"📈 Distribution of {col}", fontsize=14)
    plt.xlabel(col)
    plt.ylabel("Frequency")
    plt.grid(True, alpha=0.3)
    plt.show()
```

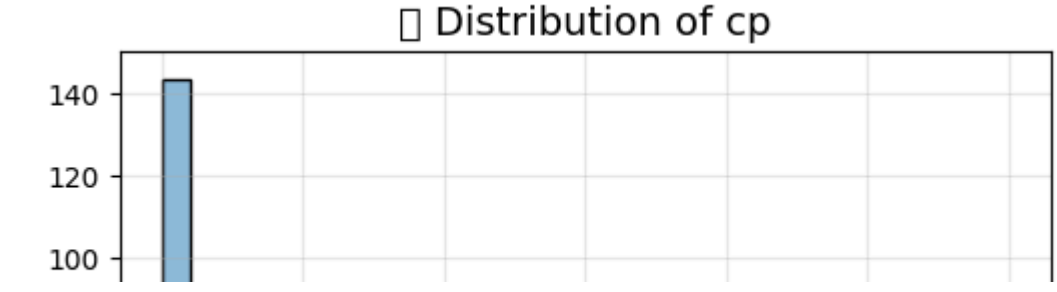
```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

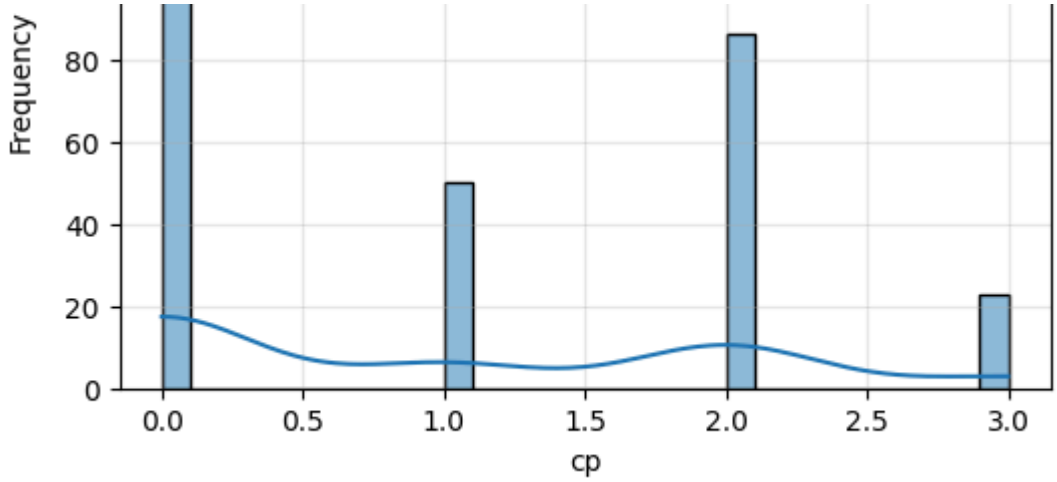



```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

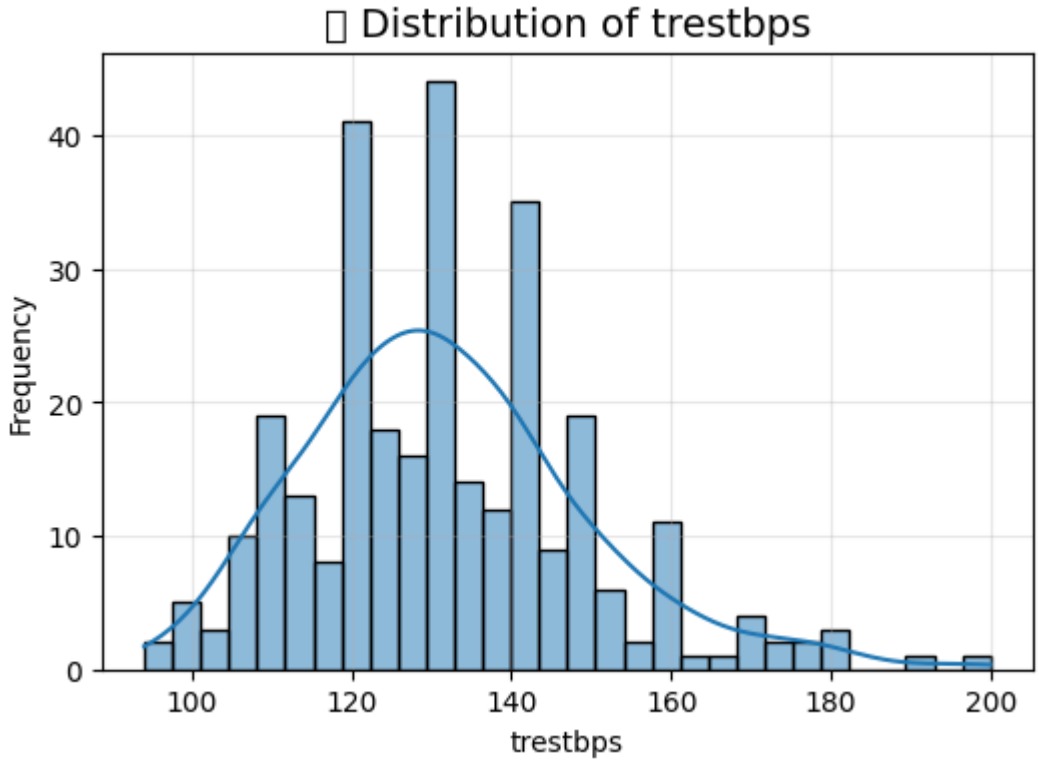



```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

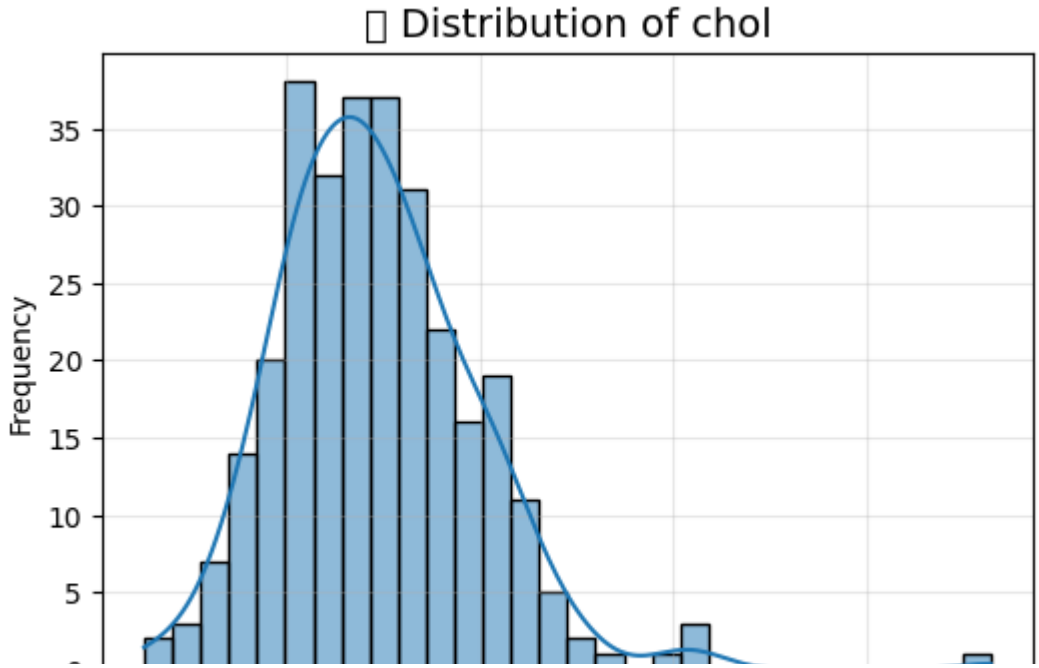


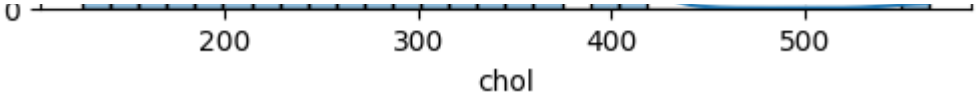



```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

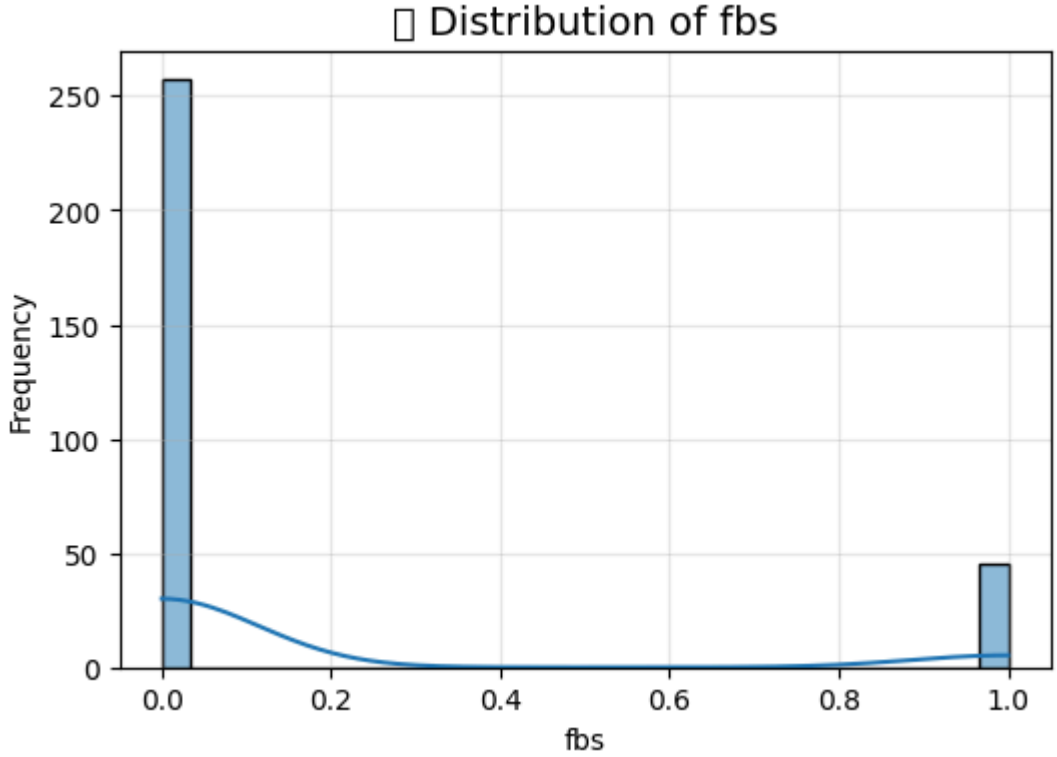



```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

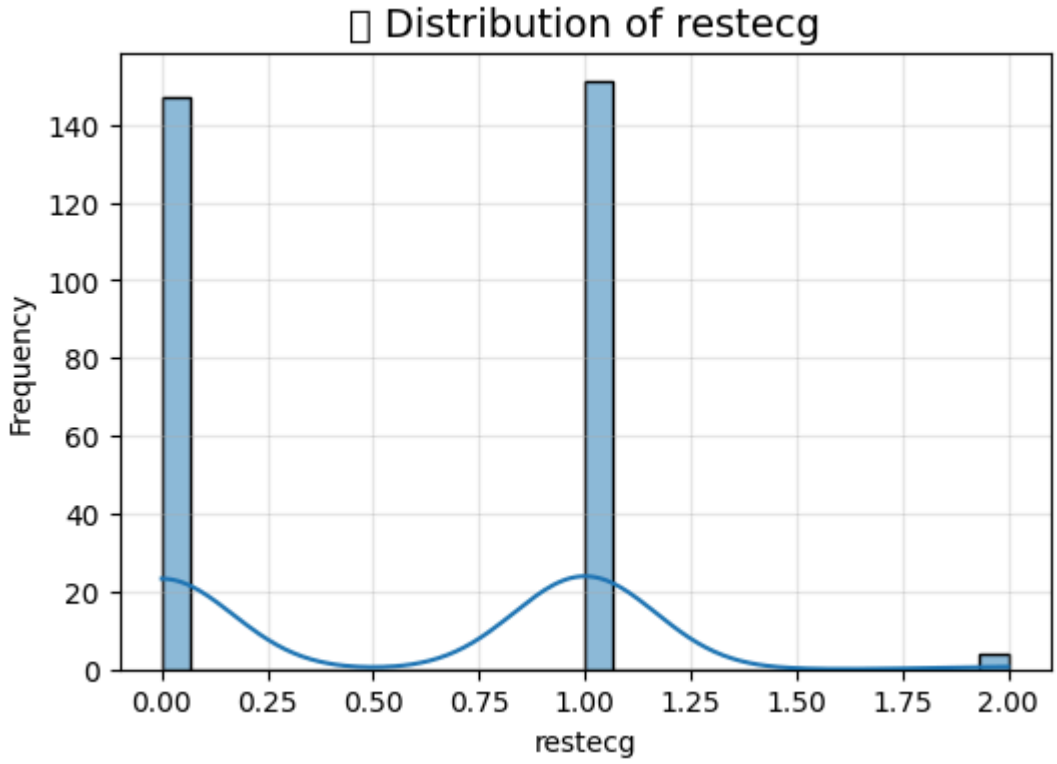




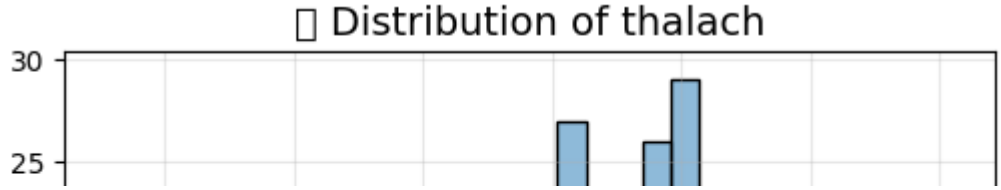

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

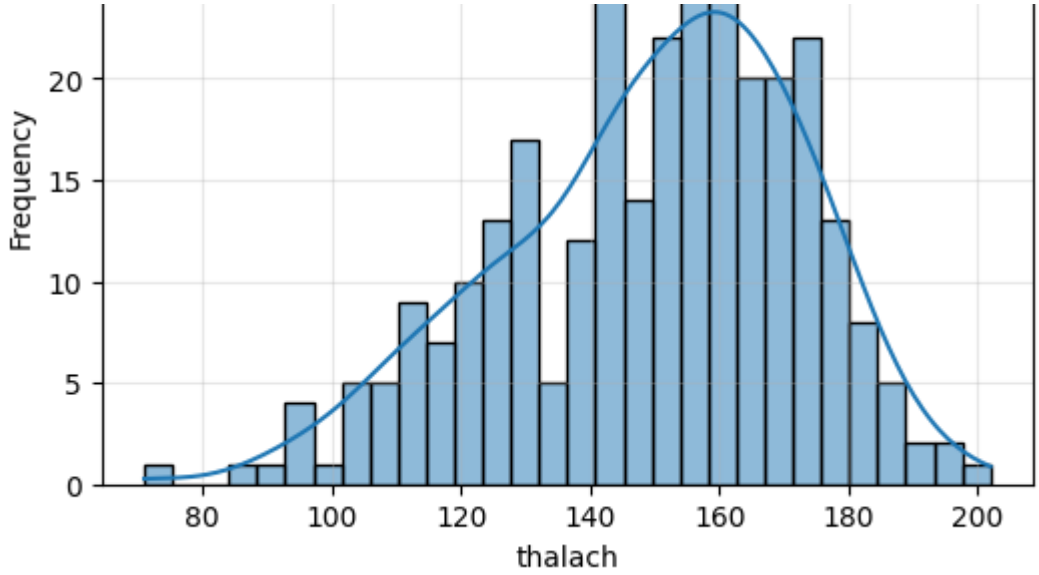



```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

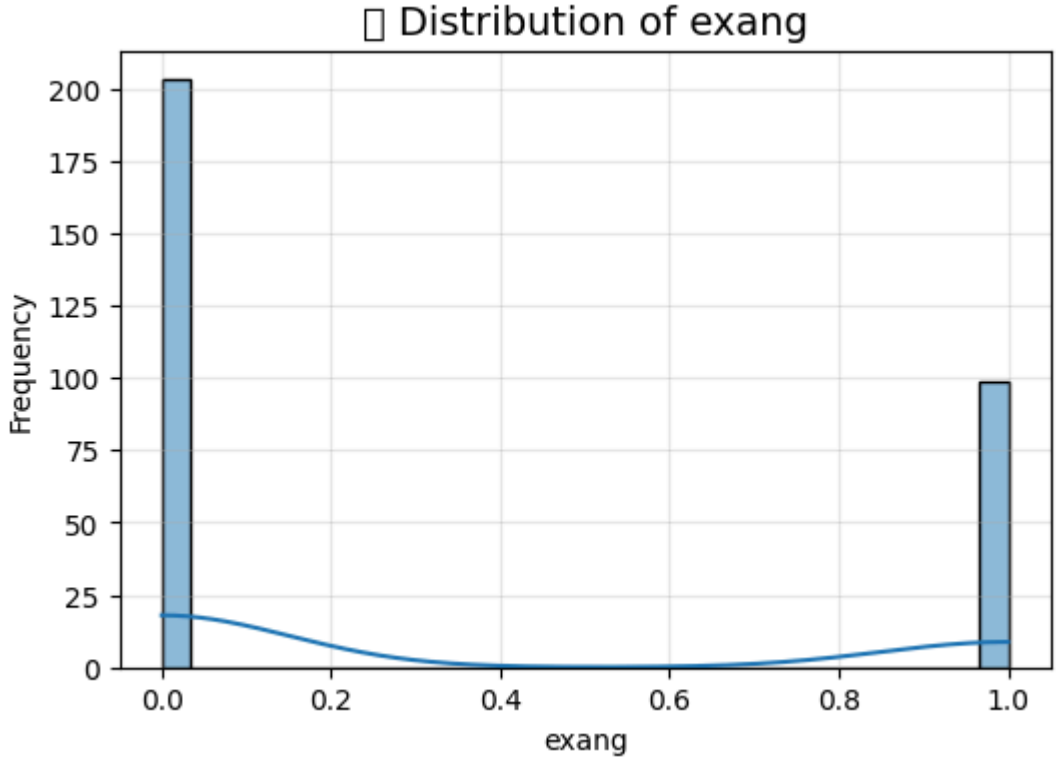



```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

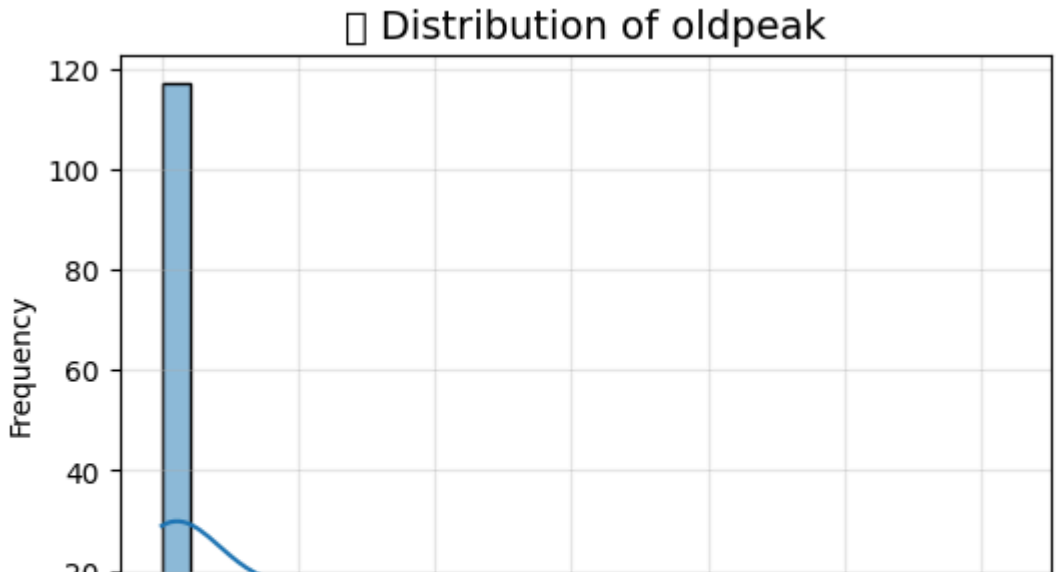


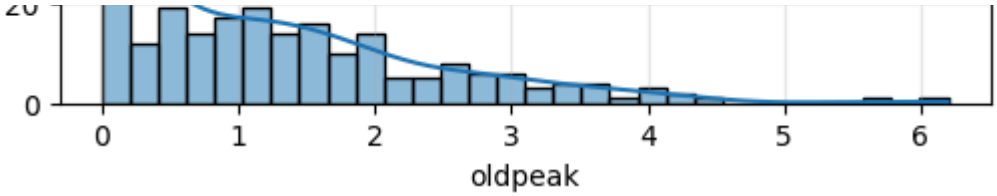



```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

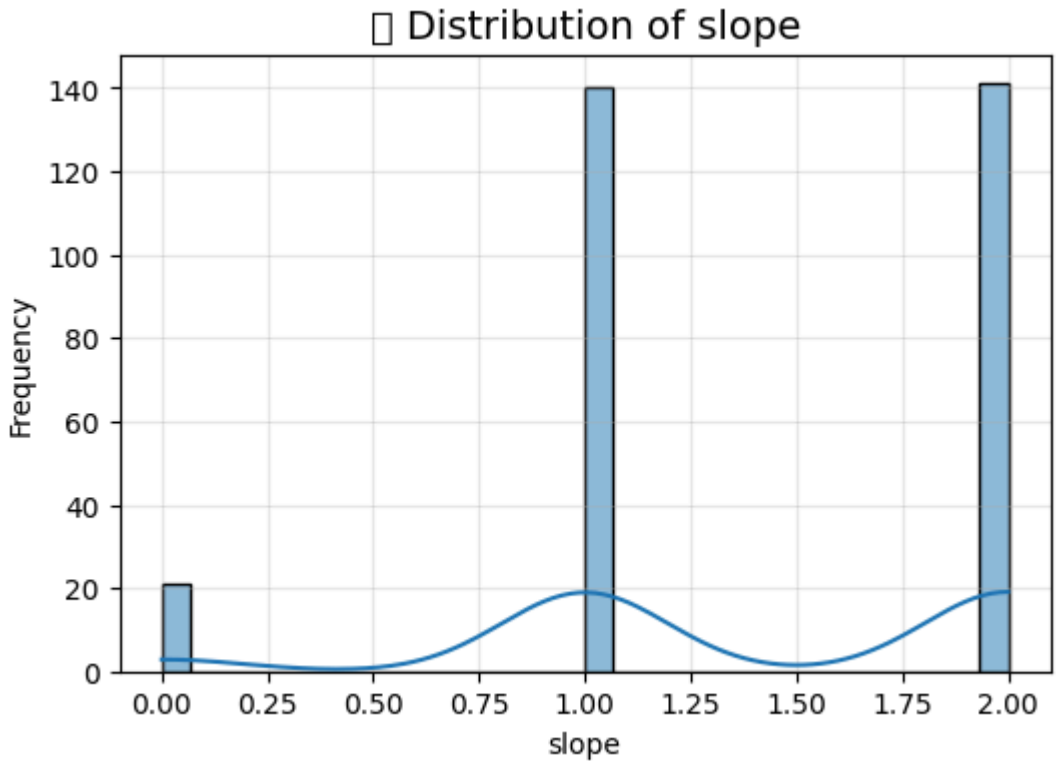



```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

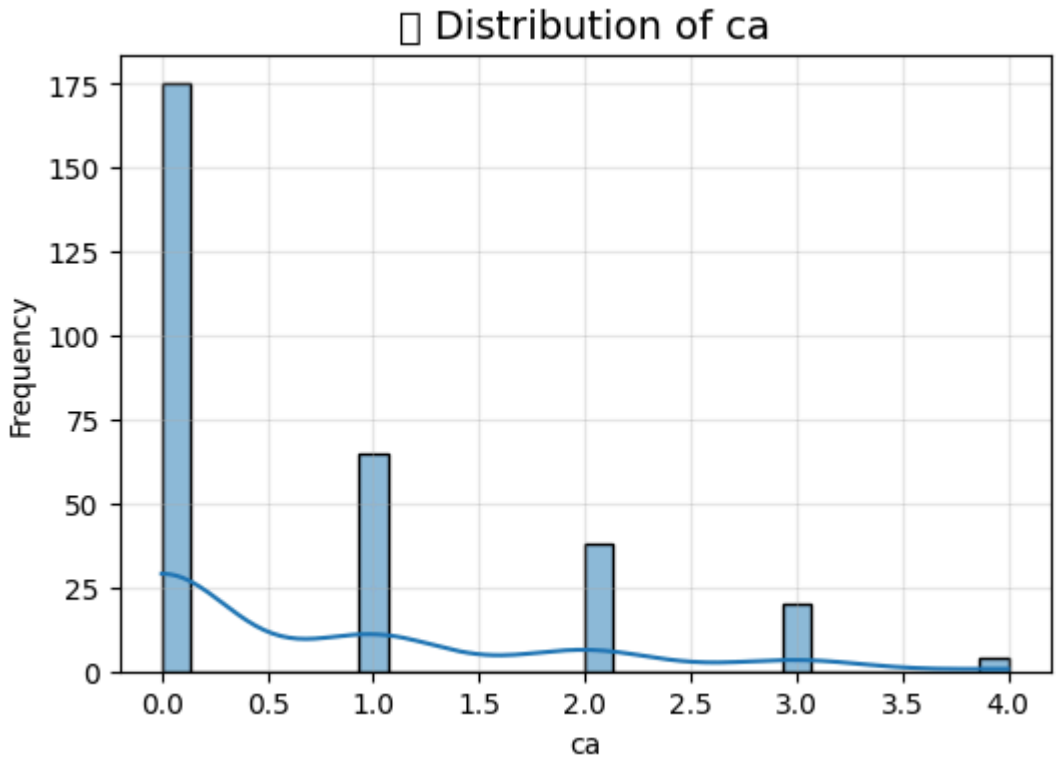




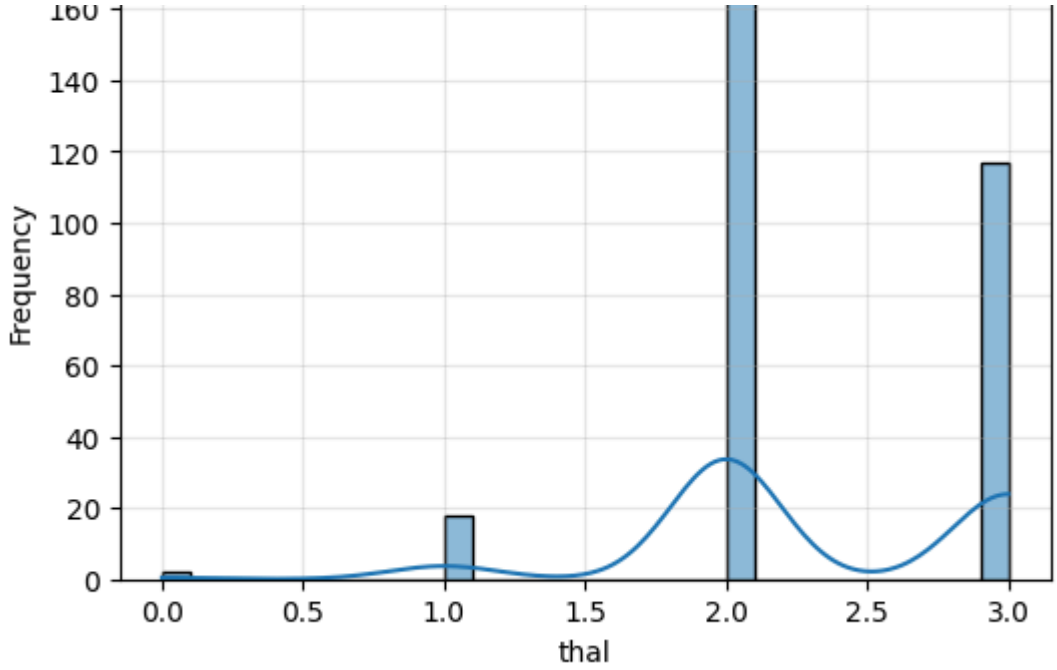

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```



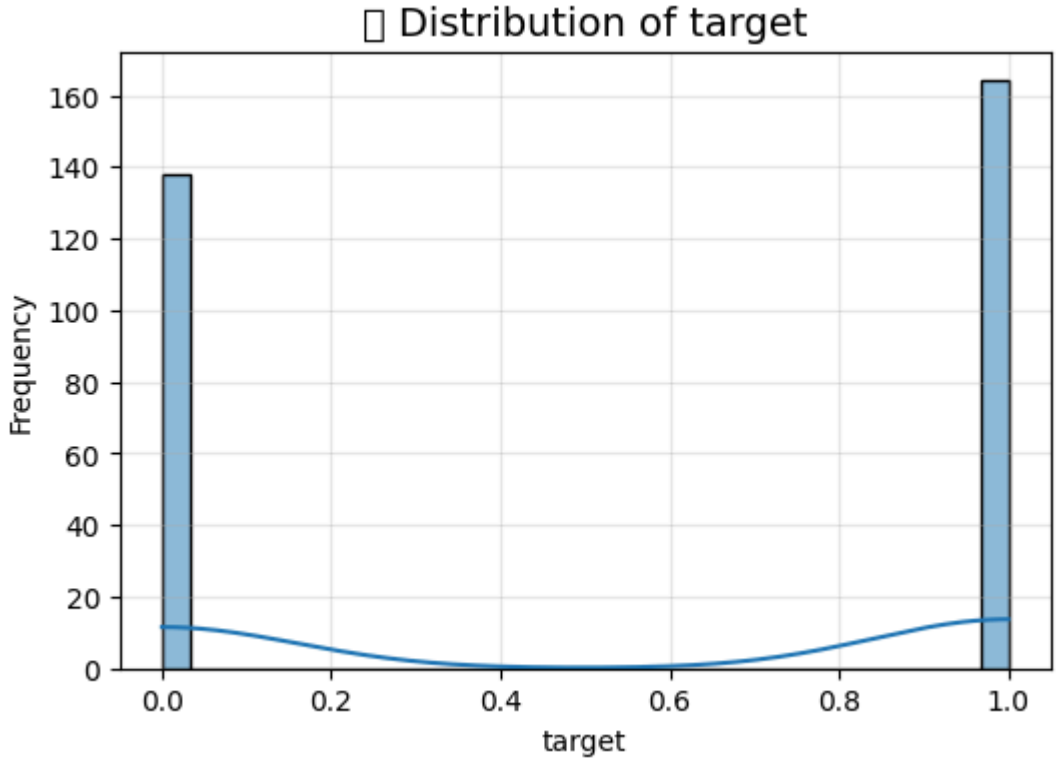

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```



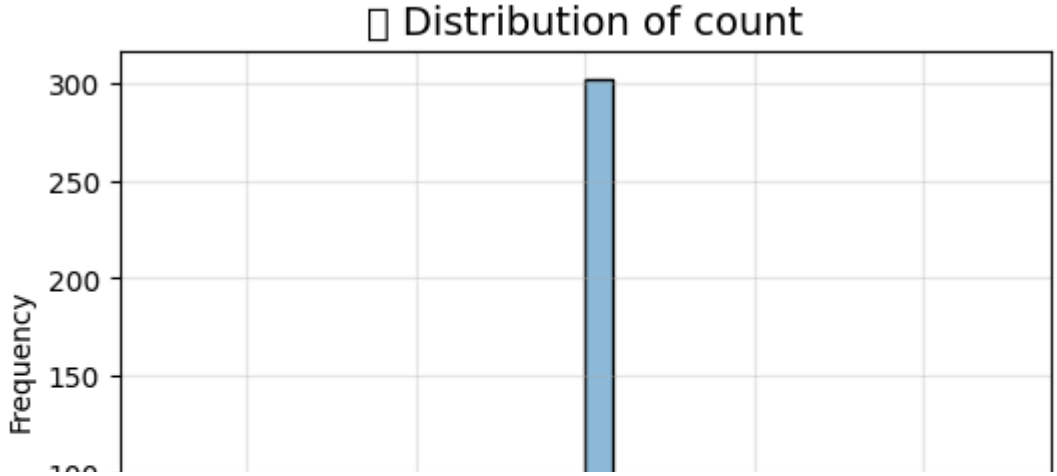

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

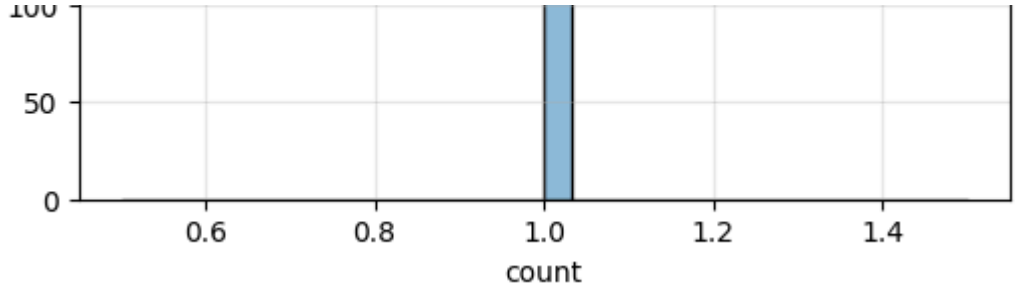



```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

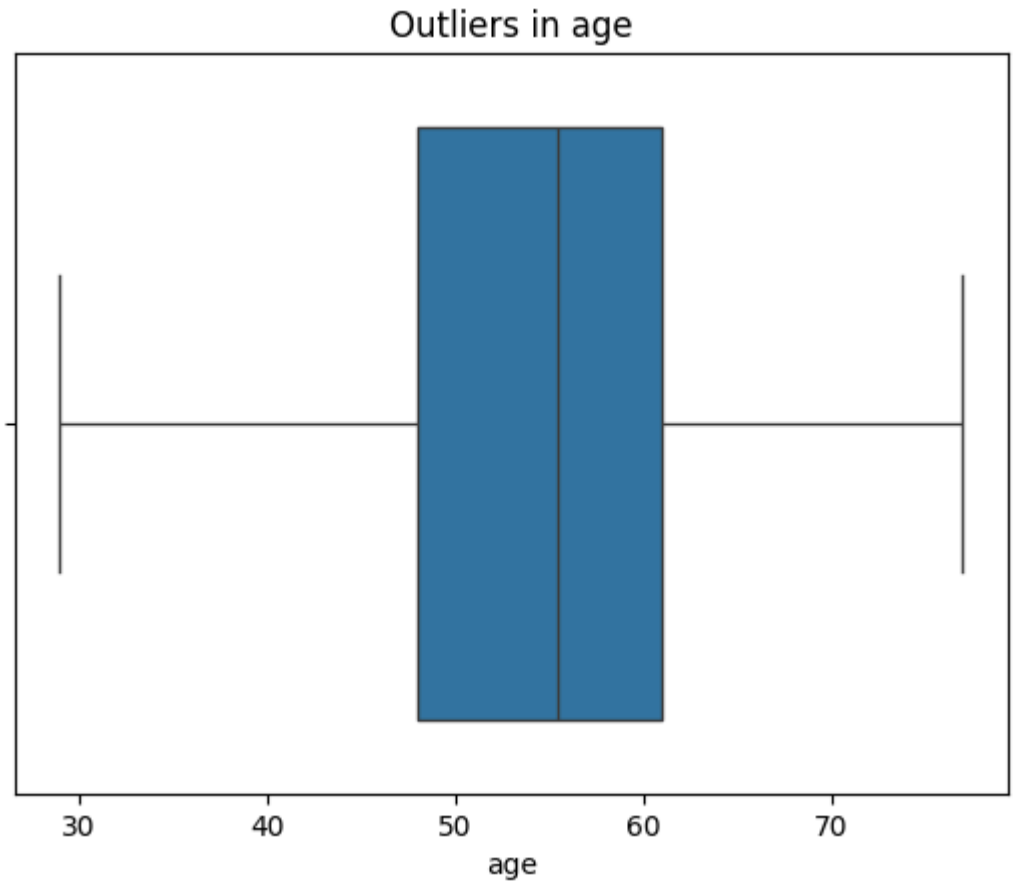


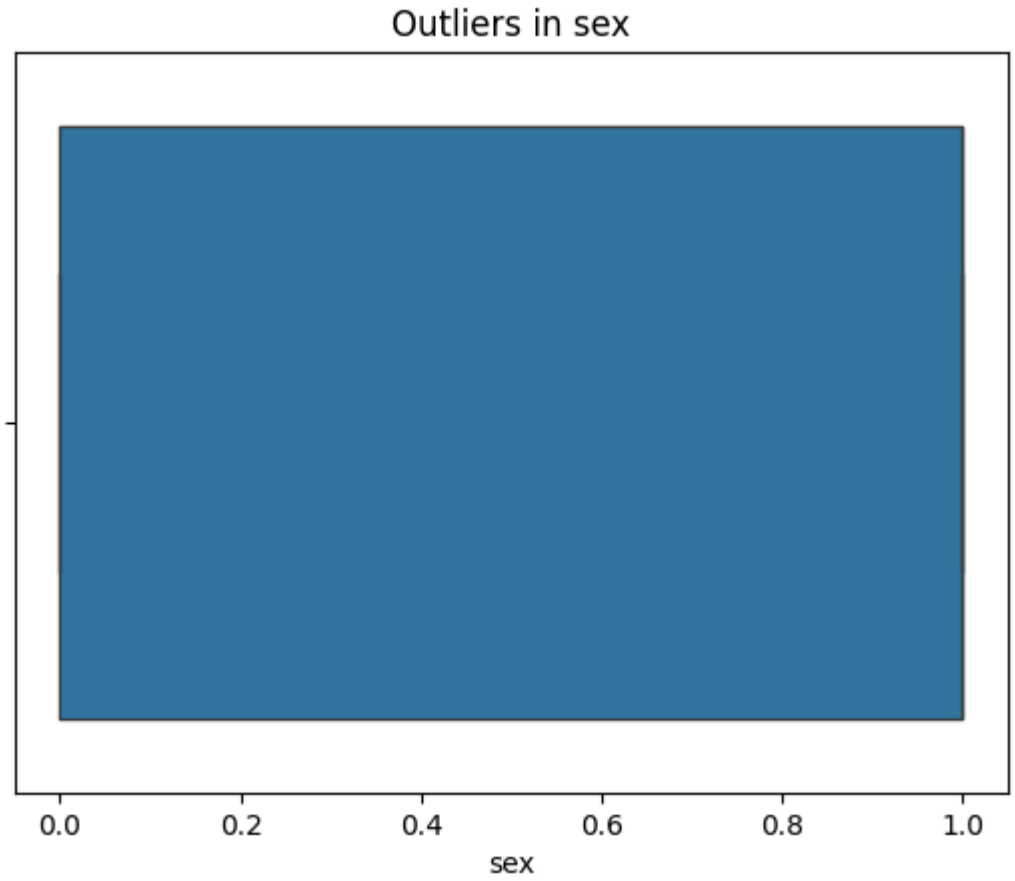

```
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

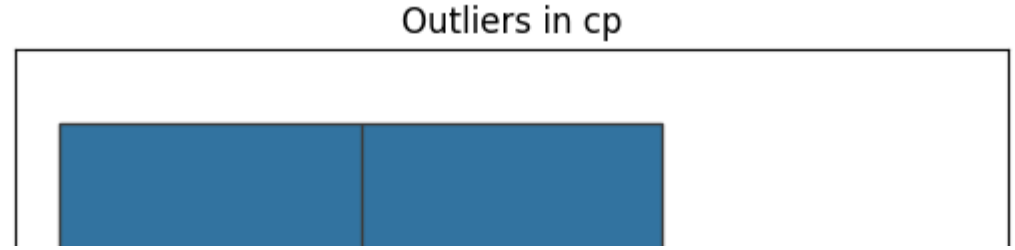


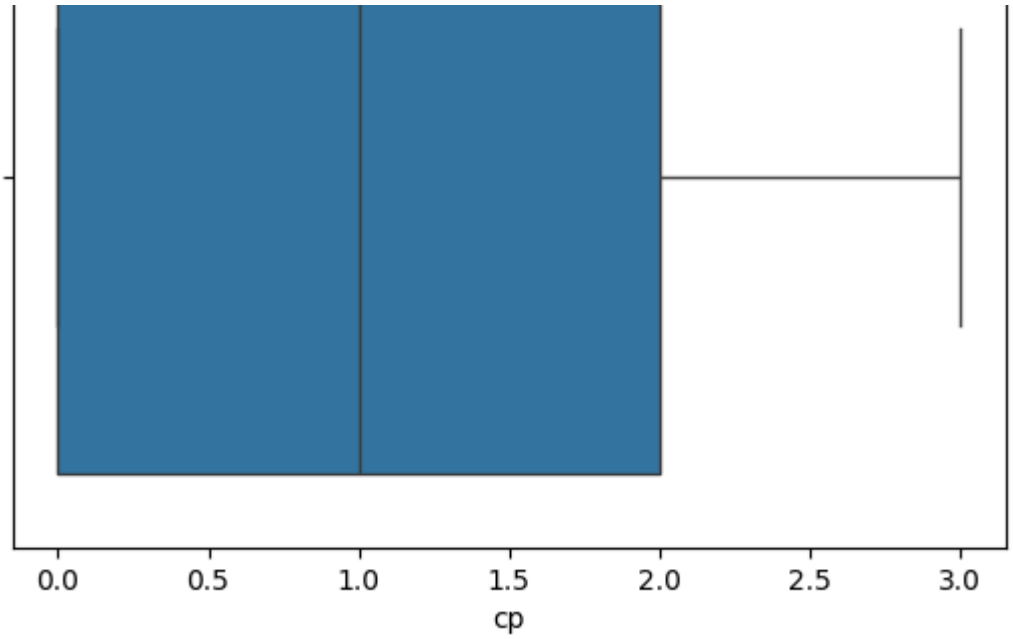

100
50
0
0.6
0.8
1.0
1.2
1.4
count

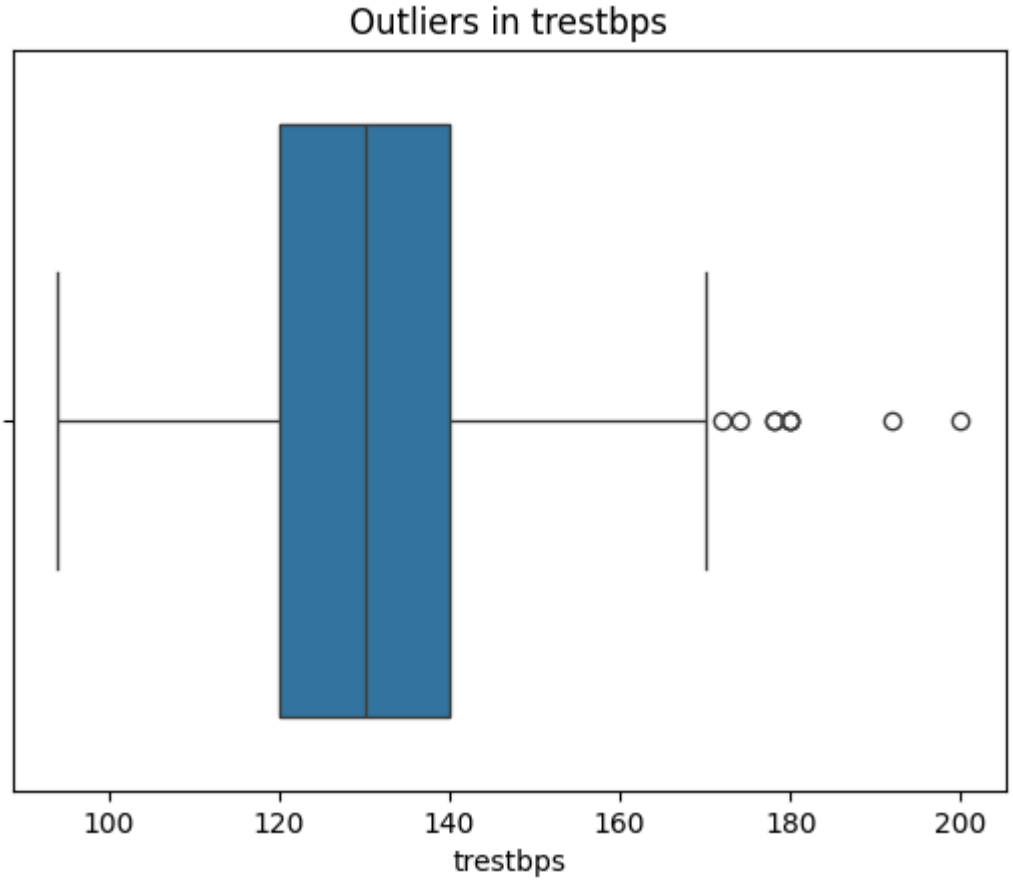
```
    for col in num_cols:
        sns.boxplot(x=df[col])
        plt.title(f"Outliers in {col}")
        plt.show()
```

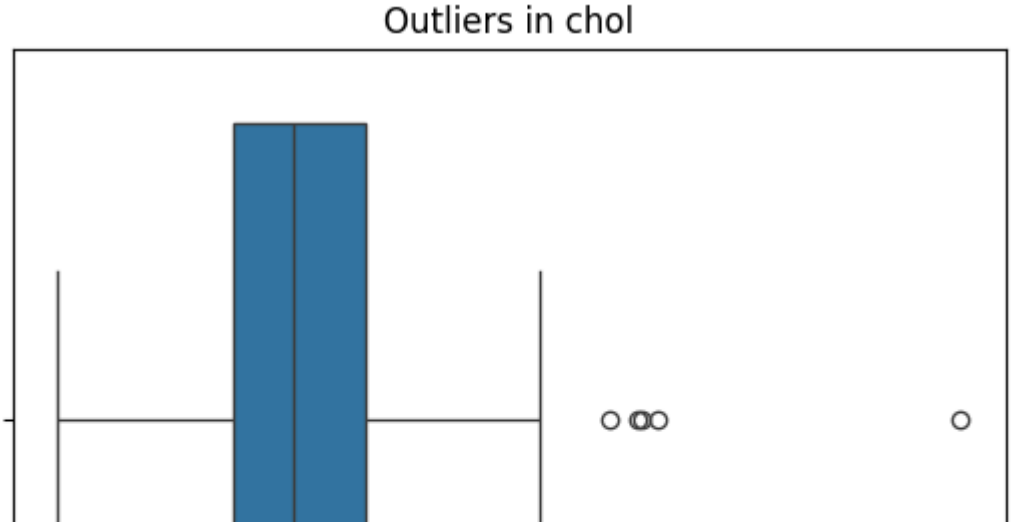

Outliers in age
30
40
50
60
70
age

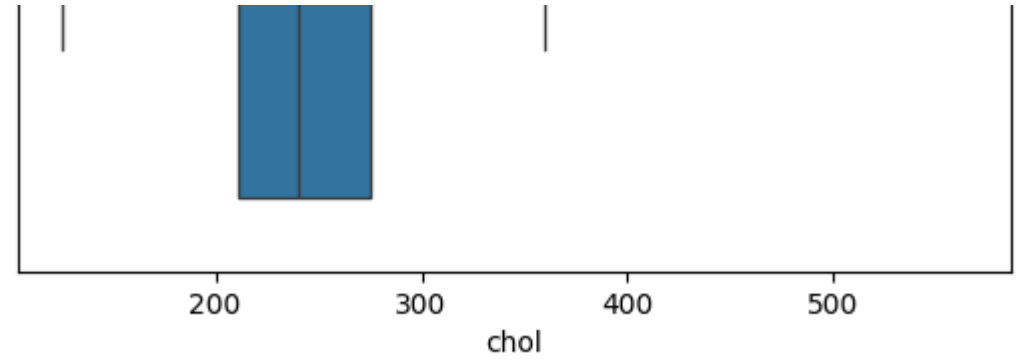

Outliers in sex
0.0
0.2
0.4
0.6
0.8
1.0
sex

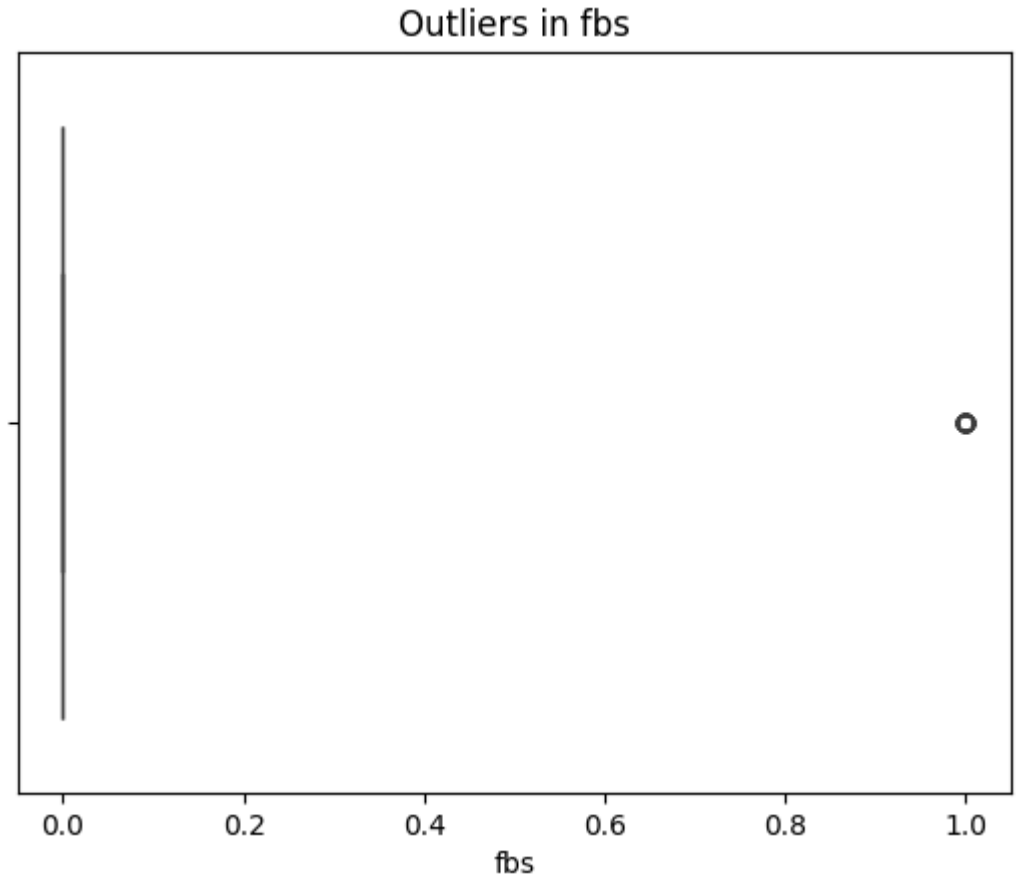

Outliers in cp

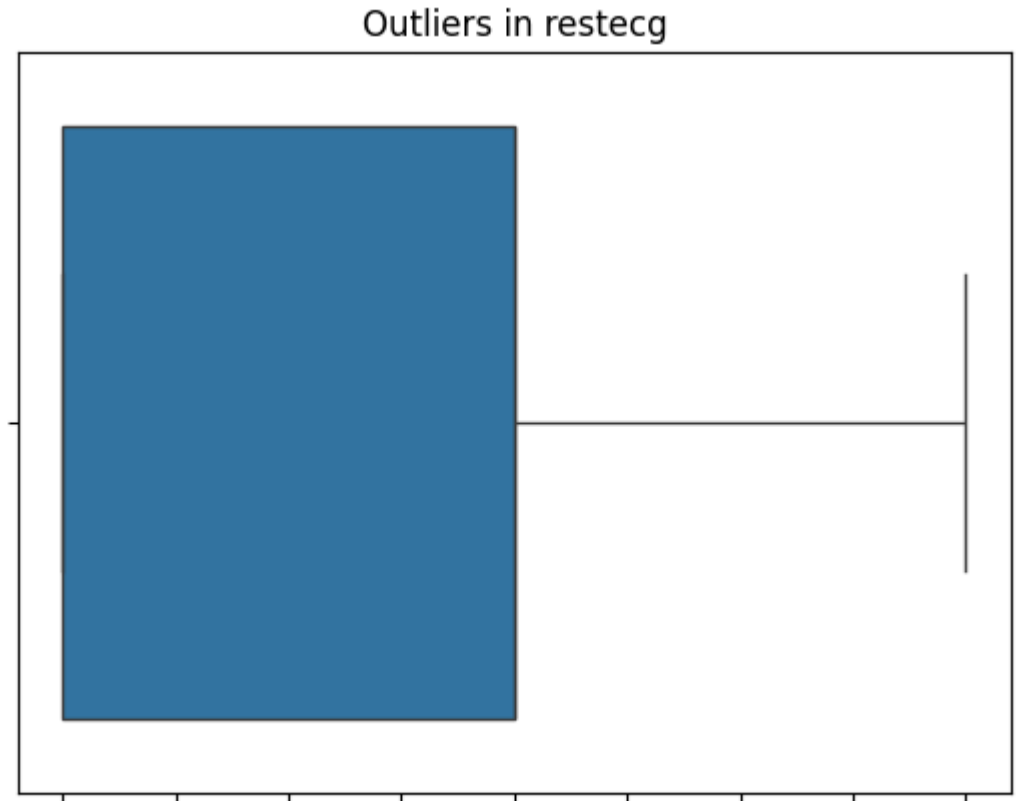

0.0
0.5
1.0
1.5
2.0
2.5
3.0
cp

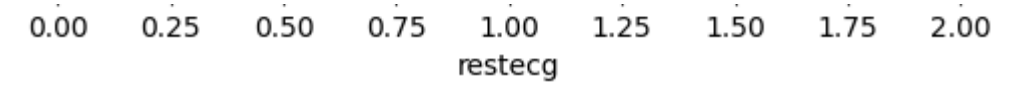

Outliers in trestbps
100
120
140
160
180
200
trestbps

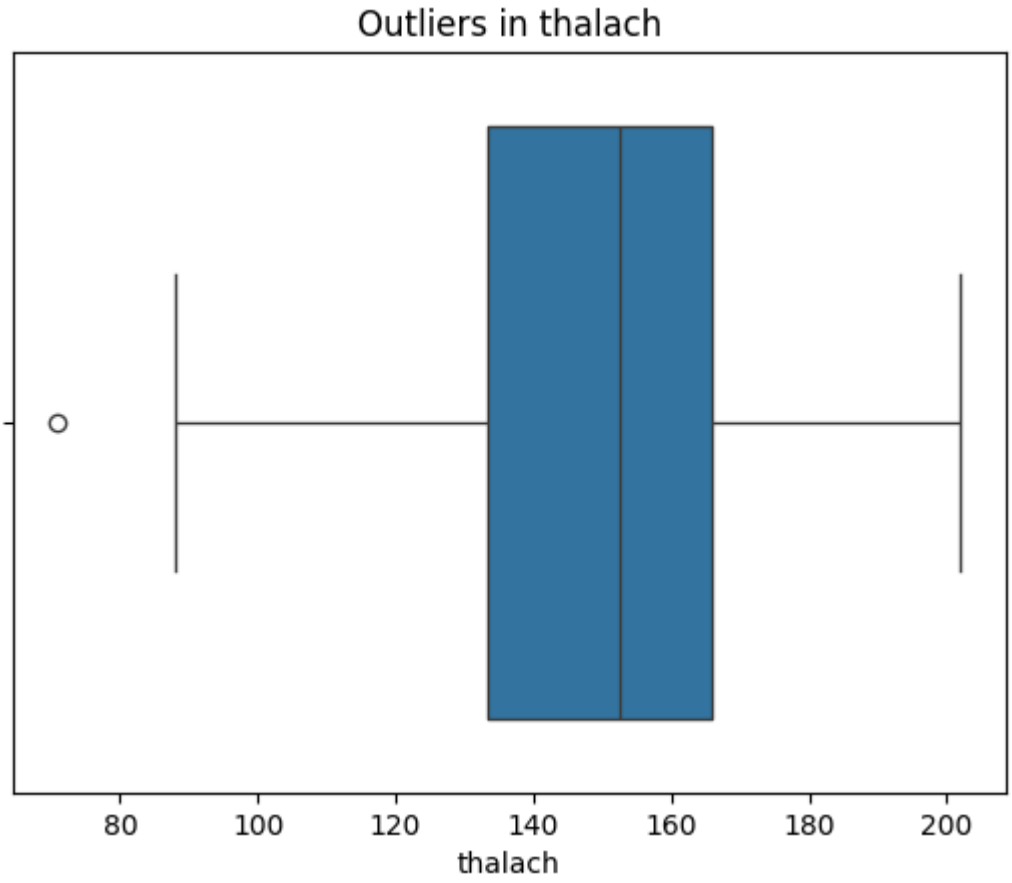

Outliers in chol

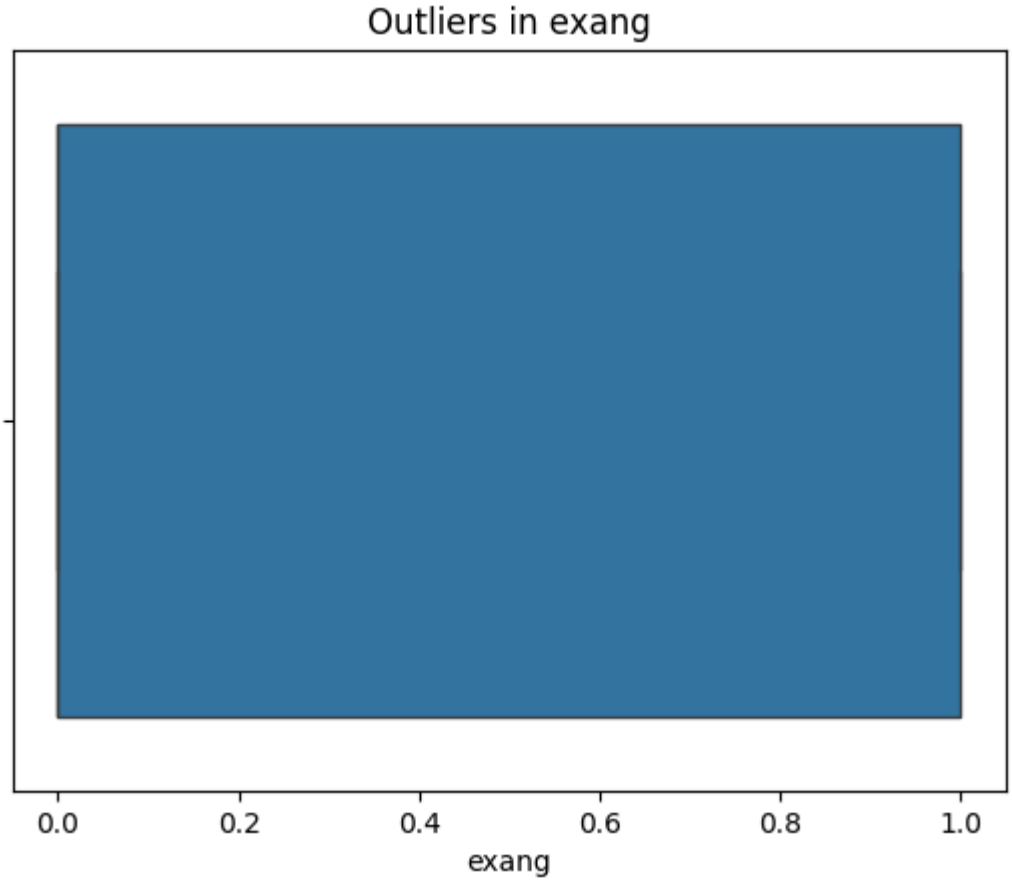

200
300
400
500
chol

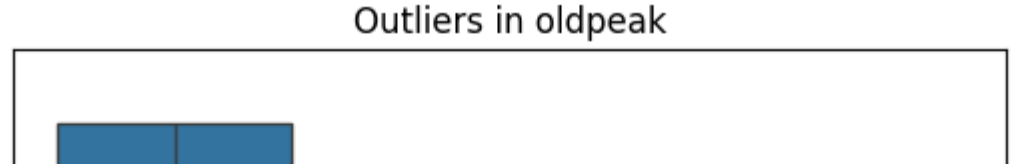

Outliers in fbs
0.0
0.2
0.4
0.6
0.8
1.0
fbs

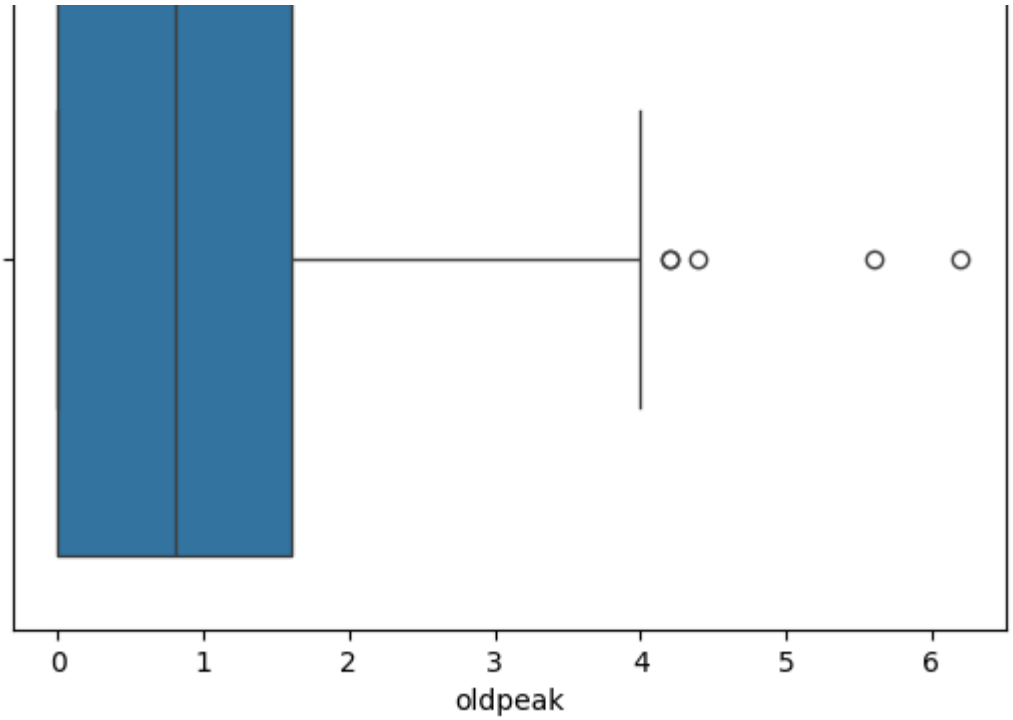

Outliers in restecg

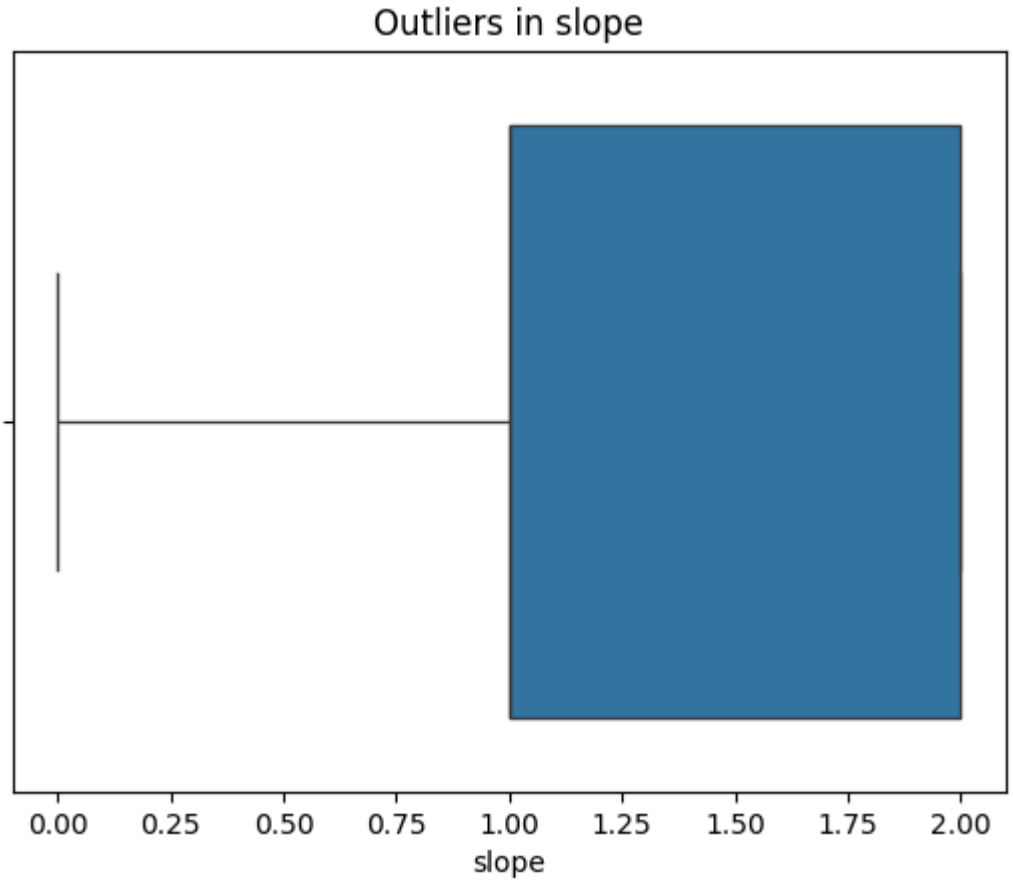

0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
restecg

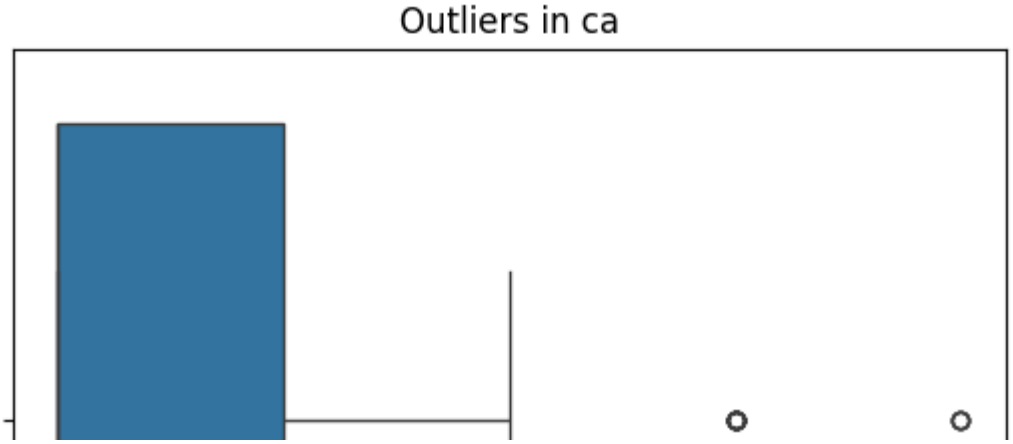

Outliers in thalach
80
100
120
140
160
180
200
thalach

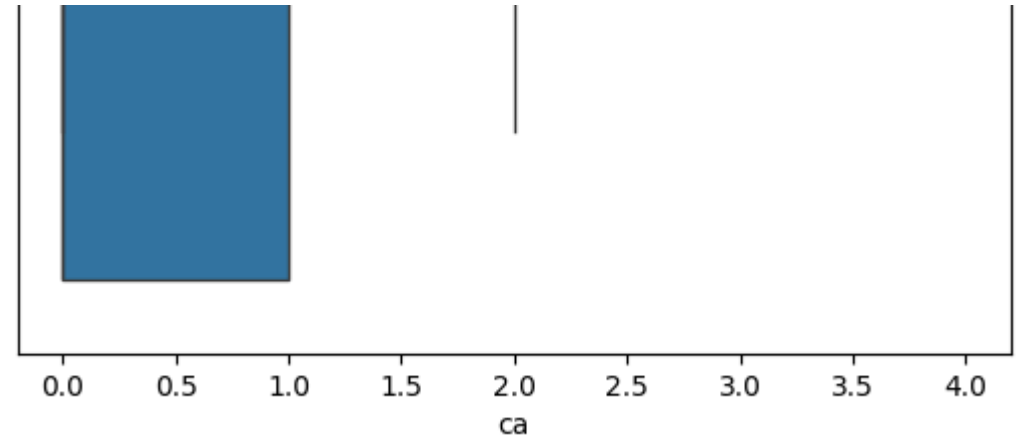

Outliers in exang
0.0
0.2
0.4
0.6
0.8
1.0
exang

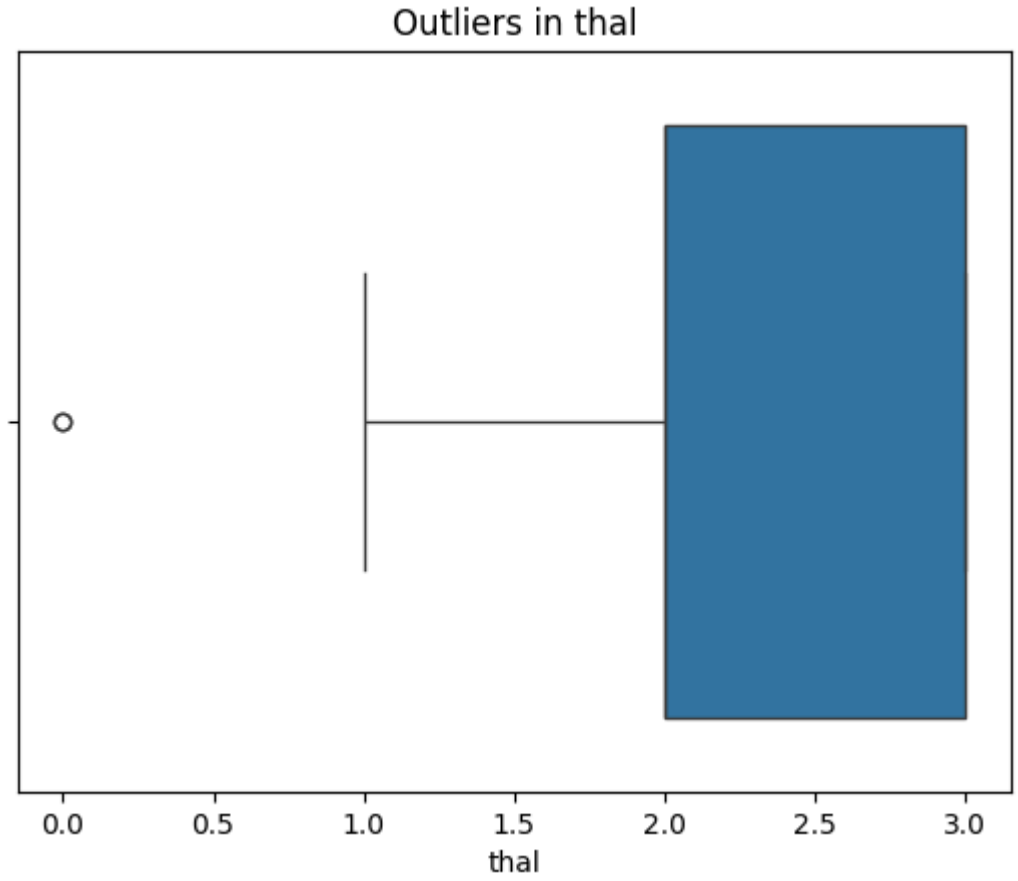

Outliers in oldpeak

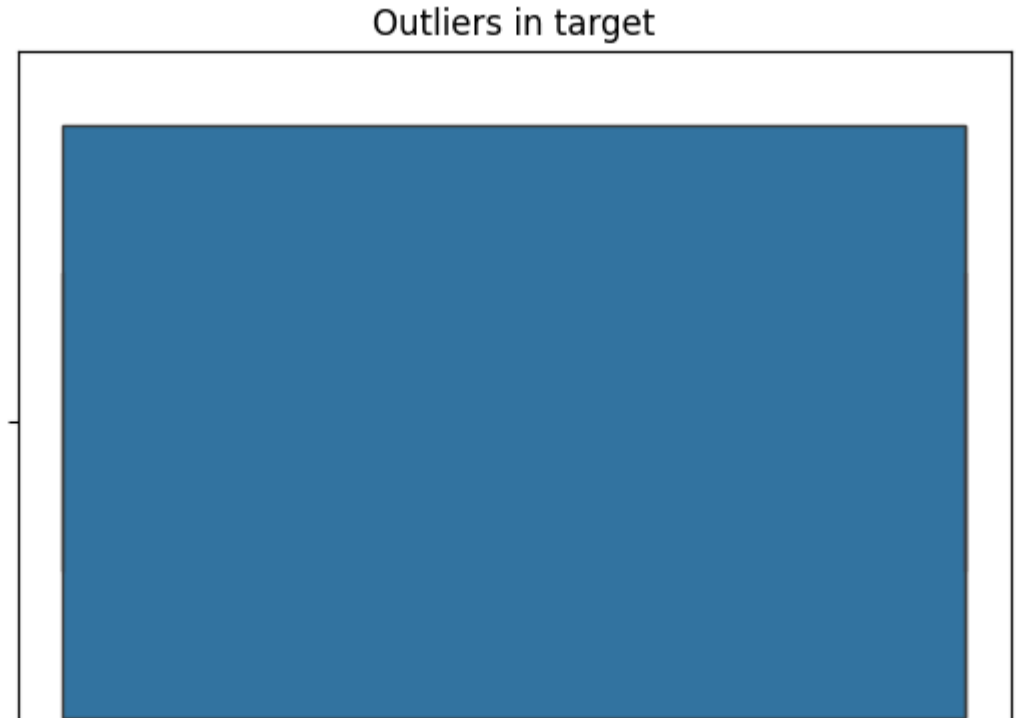

0
1
2
3
4
5
6
oldpeak


Outliers in slope
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
slope


Outliers in ca


0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
ca


Outliers in thal
0.0
0.5
1.0
1.5
2.0
2.5
3.0
thal


Outliers in target