

```
In [1]: import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
In [2]: df = pd.read_csv ('heartattack1.csv')
```

```
In [3]: df.head()
```

```
Out[3]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	3.0	145.0	233.0	1	0.0	150.0	0.0	2.3	0	0	1
1	37	1	2.0	130.0	250.0	0	1.0	187.0	0.0	3.5	0	0	2
2	41	0	1.0	130.0	204.0	0	0.0	172.0	0.0	1.4	2	0	2
3	56	1	1.0	120.0	236.0	0	1.0	178.0	0.0	0.8	2	0	2
4	57	0	0.0	120.0	354.0	0	1.0	163.0	1.0	0.6	2	0	2

```
In [4]: df.dtypes
```

```
Out[4]: age          int64
sex          int64
cp          float64
trestbps    float64
chol        float64
fbs         int64
restecg     float64
thalach     float64
exang       float64
oldpeak     float64
slope       int64
ca          int64
thal        int64
target      int64
dtype: object
```

```
In [5]: df.columns
```

```
Out[5]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
              'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
              dtype='str')
```

```
In [6]: df.isnull().sum()
```

```
Out[6]: age      0
sex      0
cp       13
trestbps 3
chol     4
fbs      0
restecg  11
thalach  3
exang    1
oldpeak  2
slope    0
ca       0
thal     0
target   0
dtype: int64
```

```
In [7]: missing_percentage = df.isnull().mean() * 100
threshold = 30
columns_to_drop = missing_percentage[missing_percentage >
threshold].index
df = df.drop(columns=columns_to_drop)
print(f"Columns dropped: {list(columns_to_drop)}")
print(f"Remaining columns: {df.columns}")
```

Columns dropped: []

Remaining columns: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'], dtype='str')

```
In [11]: for column in df.select_dtypes(include=['float64']).columns:
mean_value = df[column].mean()
df[column] = df[column].fillna(mean_value)
```

```
In [12]: df.isnull().sum()
```

```
Out[12]: 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
```

```
In [16]: df.describe()
```

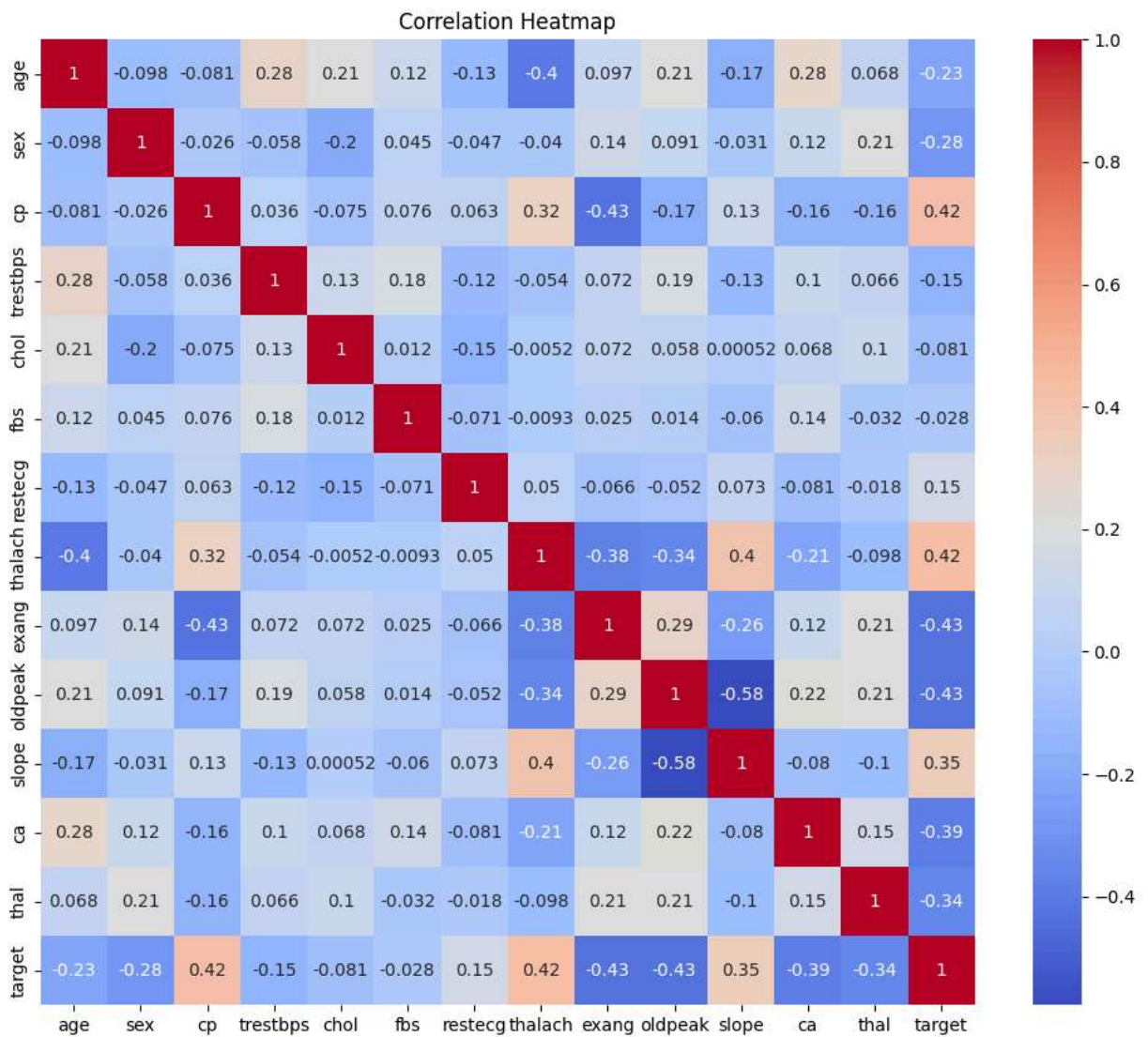
Out[16]:

	age	sex	cp	trestbps	chol	fbs	restecg
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.927586	131.506667	246.525084	0.148515	0.534247
std	9.082101	0.466011	0.989145	17.492570	51.743527	0.356198	0.516796
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000
50%	55.000000	1.000000	0.927586	130.000000	243.000000	0.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000

In [17]: `df.shape`

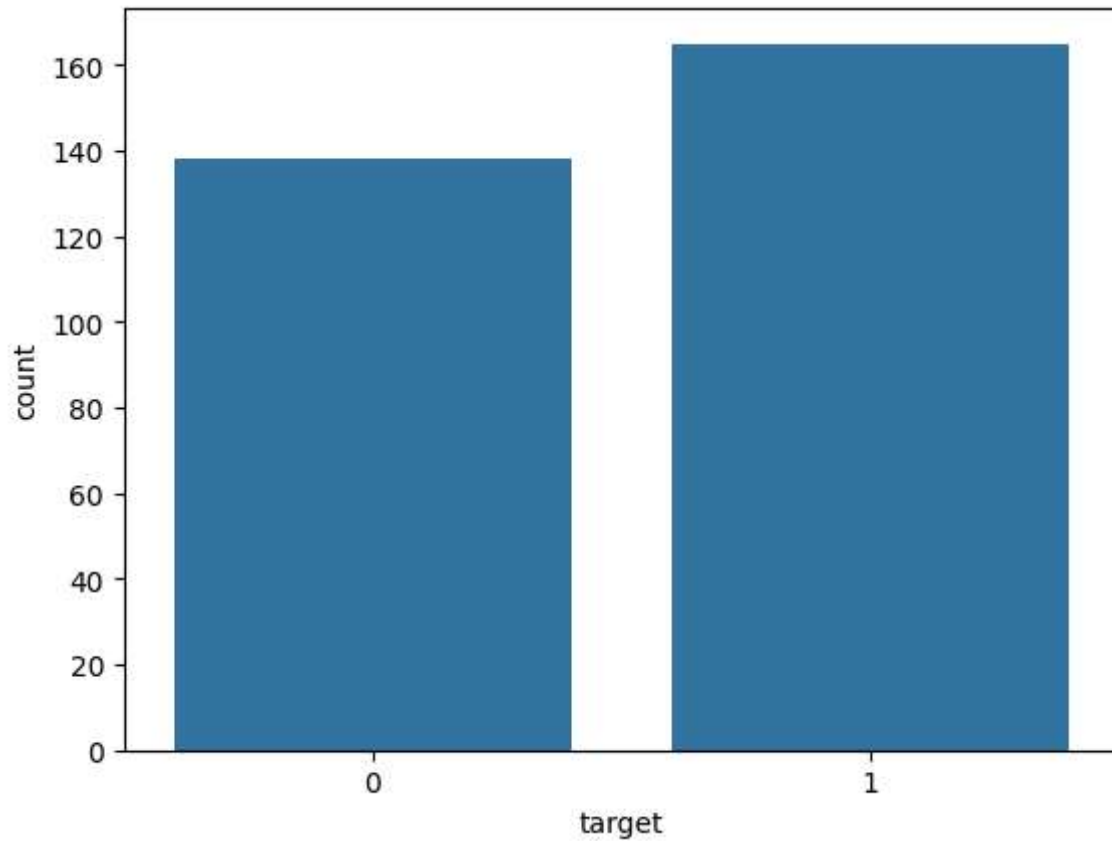
Out[17]: (303, 14)

In [18]: `plt.figure(figsize=(12, 10))`
`sns.heatmap(df.corr(), annot=True, cmap='coolwarm')`
`plt.title('Correlation Heatmap')`
`plt.show()`

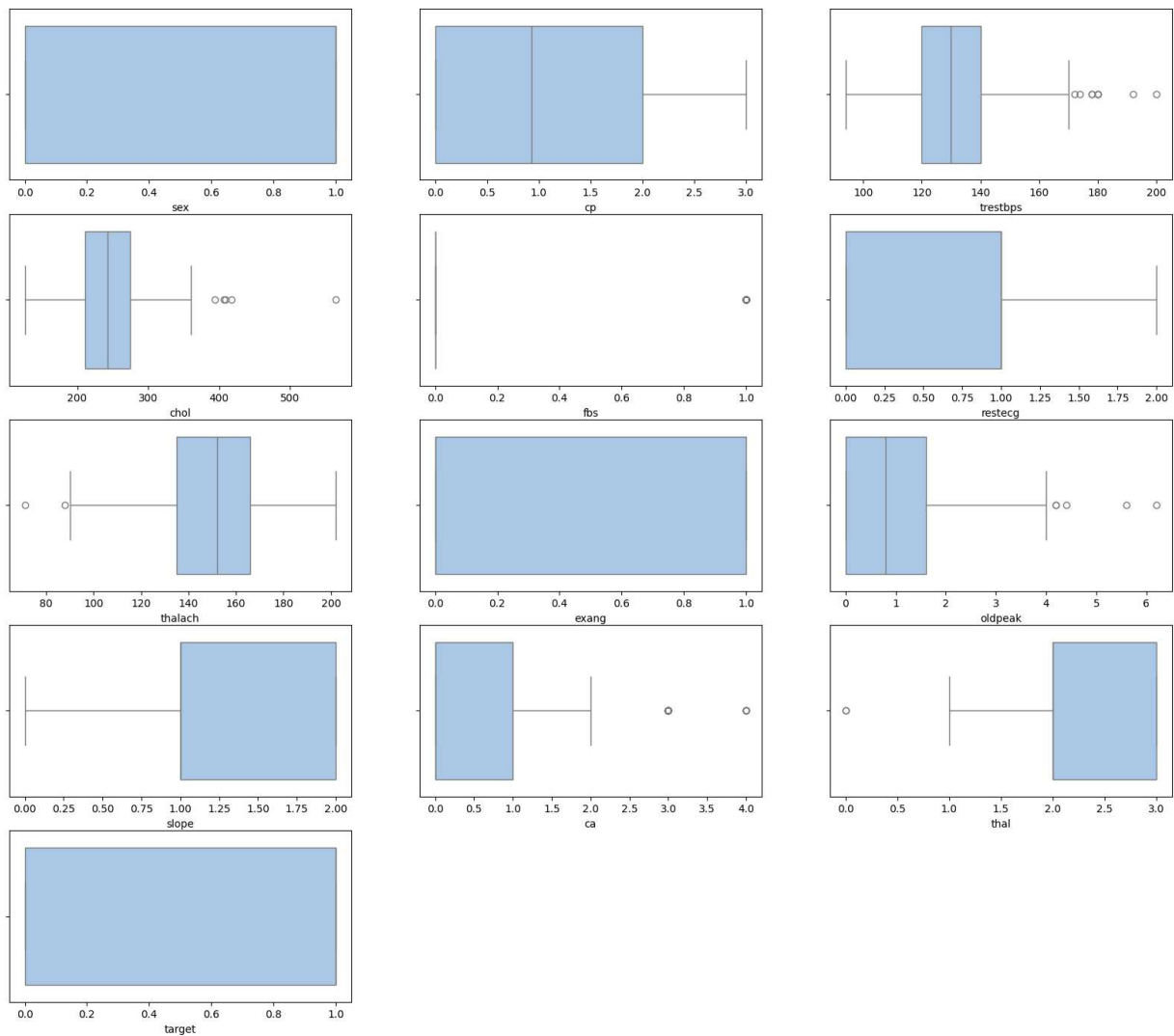


```
In [19]: sns.countplot(x=
'target'
,data=
df)
```

```
Out[19]: <Axes: xlabel='target', ylabel='count'>
```



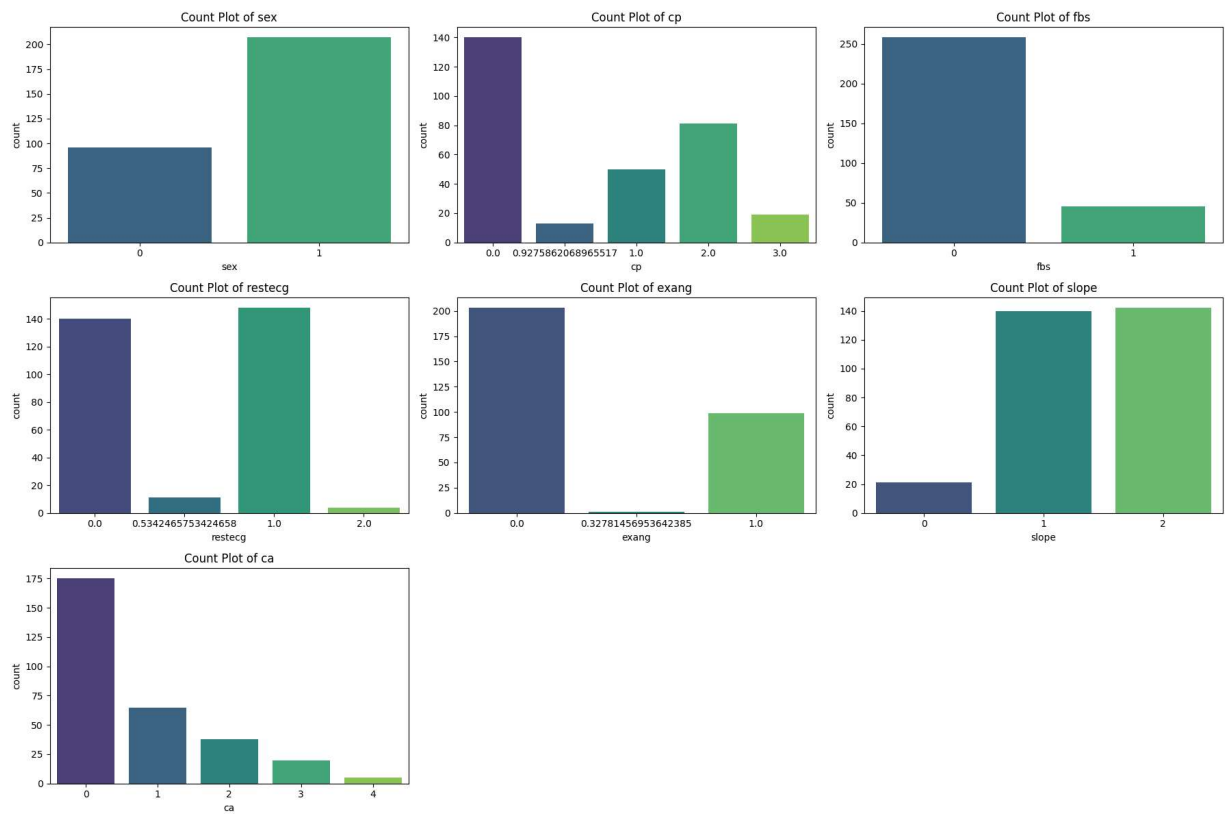
```
In [21]: plt.figure(figsize=(20,35))
plotnumber = 1
for column in df.columns[1:]:
    ax = plt.subplot(10,3,plotnumber)
    sns.boxplot(data = df, x = column, palette='pastel')
    plt.xlabel(column)
    plotnumber+=1
plt.show()
```



```
In [27]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
In [31]: numerical_cols = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
categorical_cols = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope',
'ca']
```

```
In [33]: plt.figure(figsize=(18, 12))
for i, col in enumerate(categorical_cols):
    plt.subplot(3, 3, i + 1)
    sns.countplot(x=col, data=df, palette='viridis')
    plt.title(f'Count Plot of {col}')
plt.tight_layout()
plt.show()
```



```
In [ ]:
```