

data set link: <https://www.kaggle.com/competitions/playground-series-s5e10/data>

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

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

```
df = pd.read_csv('/content/drive/MyDrive/Project 1/heart.csv')
```

```
df.head()
```

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

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
df.dtypes
```

	0
age	int64
sex	int64
cp	int64
trestbps	int64
chol	int64
fbs	int64
restecg	int64
thalach	int64
exang	int64
oldpeak	float64
slope	int64
ca	int64
thal	int64
target	int64

dtype: object

```
df.columns
```

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

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

```

    0
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
```

```
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='object')
```

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
for column in df.select_dtypes(include=['object']).columns:
    df[column] = label_encoder.fit_transform(df[column])
df.head()
```

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

Next steps: [Generate code with df](#) [New interactive sheet](#)

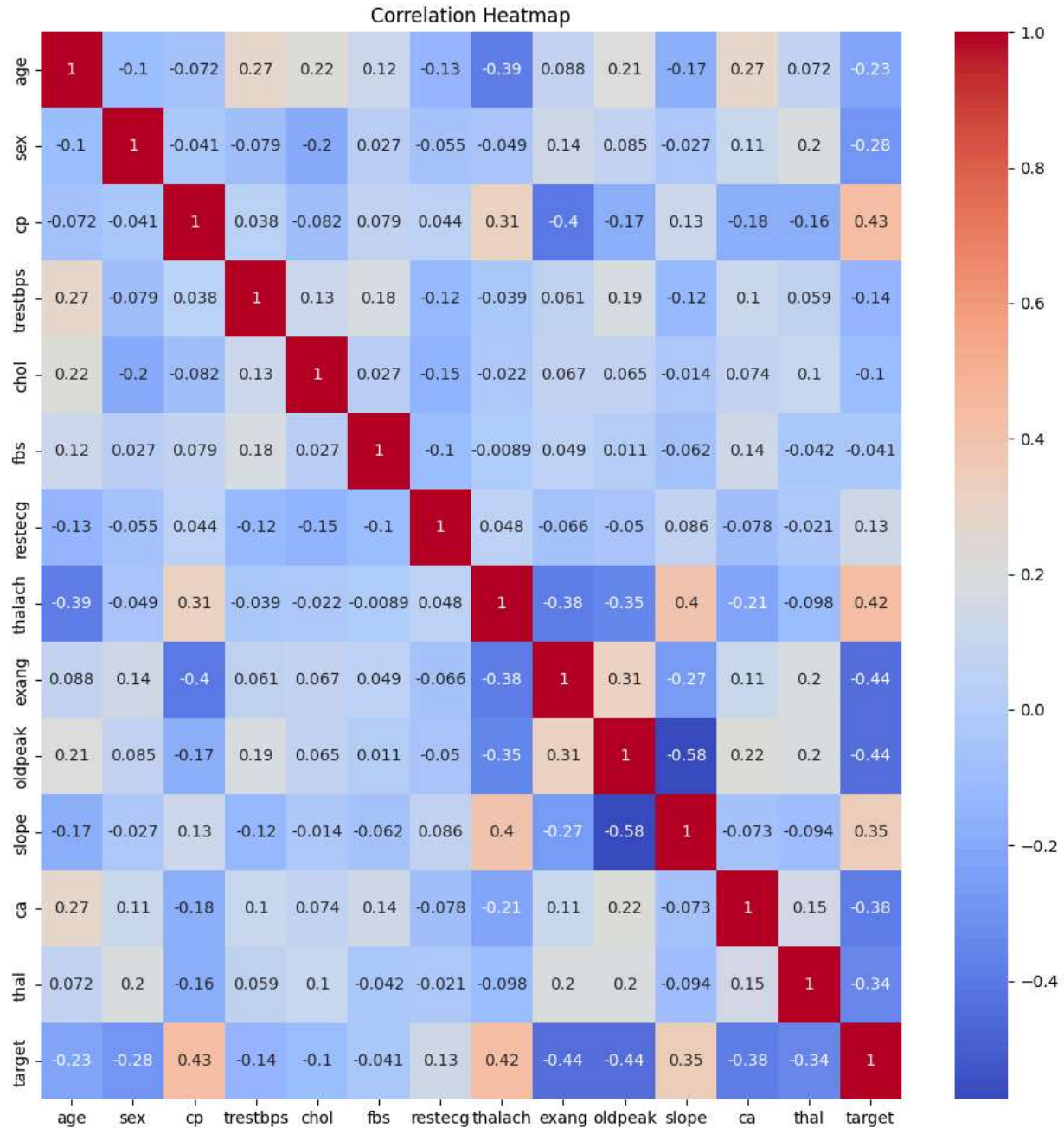
```
for column in df.select_dtypes(include=['float64']).columns:
    mean_value = df[column].mean()
```

```
for column in df.select_dtypes(include=['object']).columns:
    mode_value = df[column].mode()[0]
    df[column] = df[column].fillna(mode_value)
    df.isnull().sum()
```

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

```
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
```

```
Text(0.5, 1.0, 'Correlation Heatmap')
```



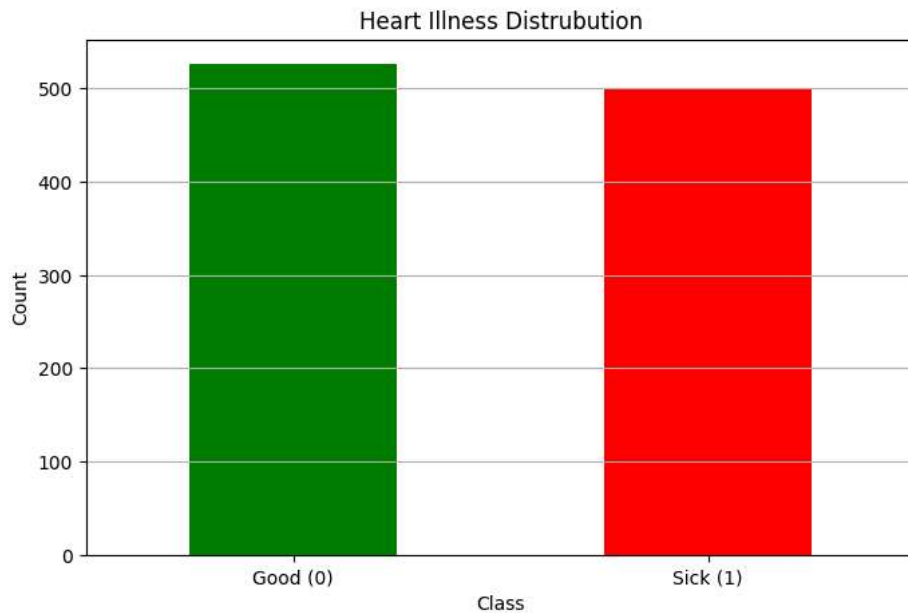
```
df['target'].value_counts()
```

```
count
target
1      526
0      499
```

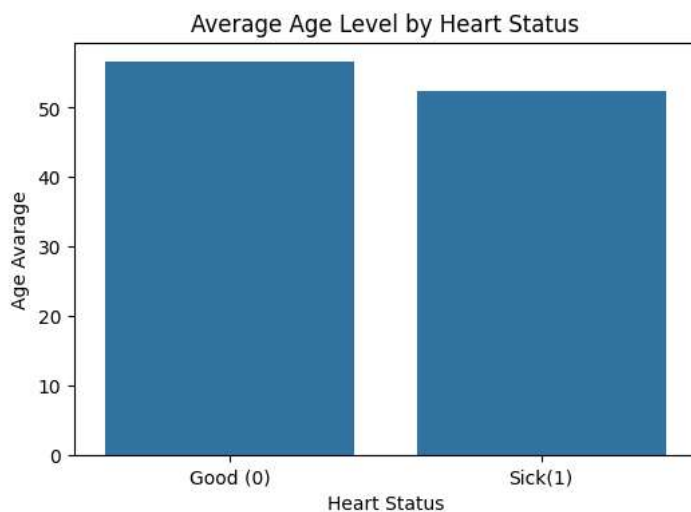
```
dtype: int64
```

```
plt.figure(figsize=(8, 5))
df['target'].value_counts().plot(kind='bar', color=['green',
'red'])
plt.xticks([0, 1], ['Good (0)', 'Sick (1)'], rotation=0)
plt.title('Heart Illness Distribution')
plt.xlabel('Class')
```

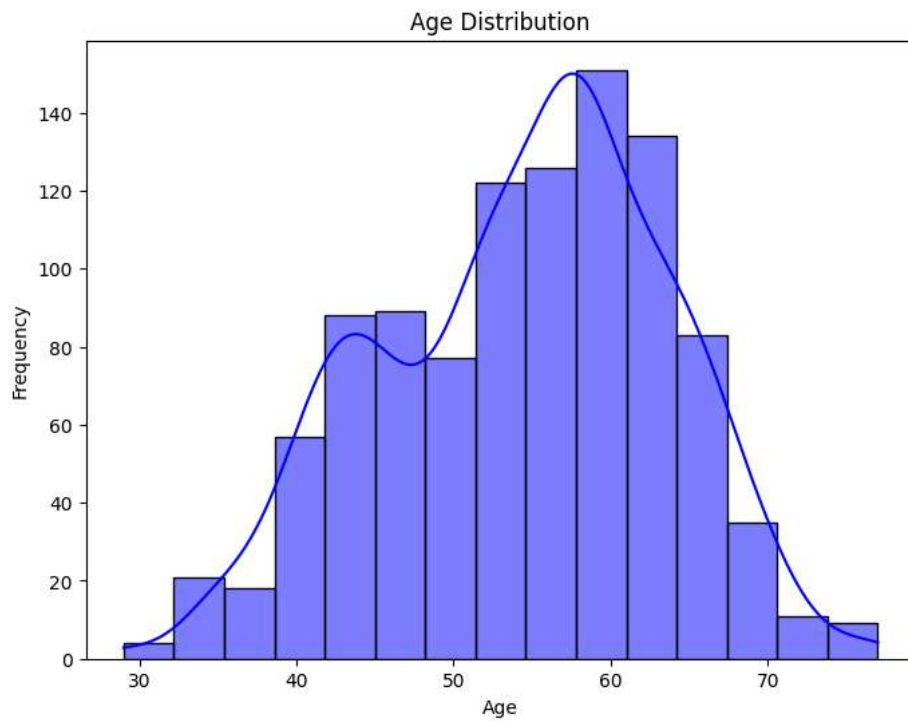
```
plt.ylabel('Count')
plt.grid(axis='y')
plt.show()
```



```
import seaborn as sns
plt.figure(figsize=(6,4))
sns.barplot(x='target', y='age', data=df, estimator='mean',
ci=None)
plt.title("Average Age Level by Heart Status")
plt.xlabel("Heart Status")
plt.ylabel("Age Avarage")
plt.xticks([0, 1], ['Good (0)', 'Sick(1)'], rotation=0)
plt.show()
```



```
plt.figure(figsize=(8, 6))
sns.histplot(df['age'], bins=15, kde=True, color='blue')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



```
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()
```

