

✓ Cardiac Arrest Data Analysis(Python Project)

Dataset Overview

1. Rows: 1,025
2. Columns: 14
3. Target Column: target — Likely indicates presence (1) or absence (0) of heart disease.
4. Includes features like:
 - age, sex, cp (chest pain type), trestbps (resting BP), chol (cholesterol), fbs (fasting blood sugar)
 - restecg, thalach (max heart rate), exang (exercise-induced angina)
 - oldpeak, slope, ca (major vessels), thal

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

```
# Load the uploaded dataset
df = pd.read_csv('/content/drive/MyDrive/cardiac_arrest/cardiac_arrest_dataset.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](#) [View recommended plots](#) [New interactive sheet](#)


```
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
```

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

```
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 6)
```



```
basic_stats = df.describe()
basic_stats
```



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.
mean	54.434146	0.695610	0.942439	131.611707	246.000000	0.149268	0.529756	149.114146	0.336585	1.071512	1.
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.
25%	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	132.000000	0.000000	0.000000	1.
50%	56.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	152.000000	0.000000	0.800000	1.
75%	61.000000	1.000000	2.000000	140.000000	275.000000	0.000000	1.000000	166.000000	1.000000	1.800000	2.
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.

Next steps: [Generate code with basic_stats](#) [View recommended plots](#) [New interactive sheet](#)

Basic Exploration Summary

-  Basic Statistics:
- Average Age: ~54.4 years
 - Average Maximum Heart Rate (thalach): ~149 bpm
 - Cholesterol (chol): Mean ~246 mg/dL, with a high outlier up to 564
 - Target values are balanced: about 51% have heart disease (1)
-  Missing Values: There are no missing values in any column.

```
missing_values = df.isnull().sum()
missing_values
```



	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

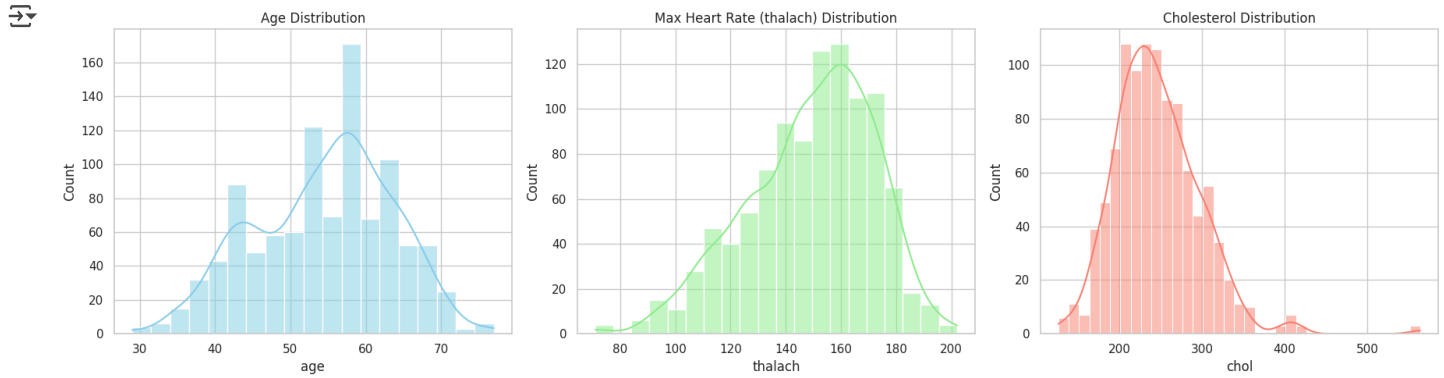
Visual Analysis

```
# 1. Distribution of Age, Thalach, Cholesterol
fig, axs = plt.subplots(1, 3, figsize=(18, 5))
sns.histplot(df['age'], kde=True, ax=axs[0], color="skyblue")
axs[0].set_title("Age Distribution")

sns.histplot(df['thalach'], kde=True, ax=axs[1], color="lightgreen")
axs[1].set_title("Max Heart Rate (thalach) Distribution")

sns.histplot(df['chol'], kde=True, ax=axs[2], color="salmon")
axs[2].set_title("Cholesterol Distribution")
```

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



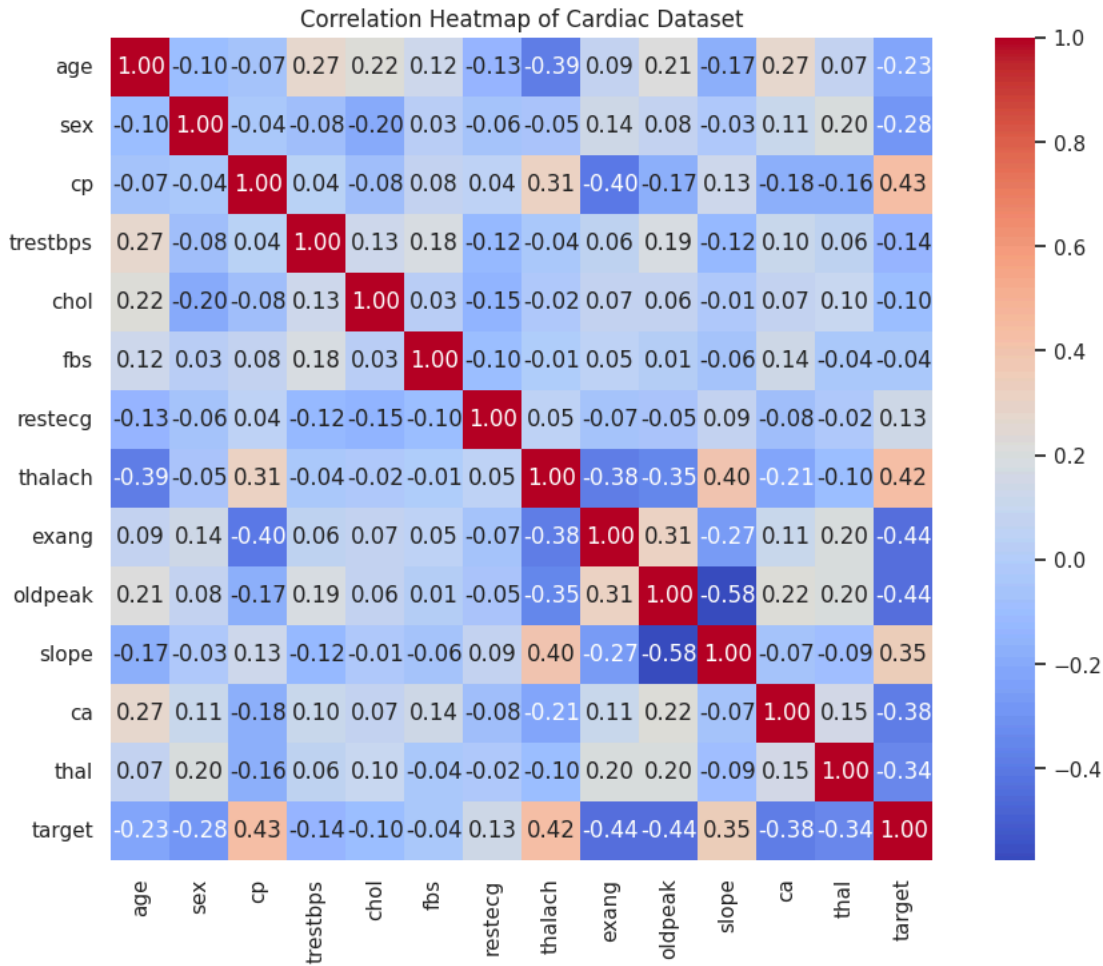
Insights from Distribution Plots

1. Age: Most patients are between 45–65 years.
2. Max Heart Rate (thalach): Most values range from 130–170 bpm, which aligns with typical stress-test results.
3. Cholesterol: Distribution is skewed, with some outliers above 400 mg/dL, which may indicate risk.

Correlation Heatmap

```
# Calculate correlation matrix
correlation_matrix = df.corr()

# Plot heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap="coolwarm", square=True)
plt.title("Correlation Heatmap of Cardiac Dataset")
plt.show()
```



🔥 Correlation Heatmap Insights

- thalach (max heart rate) has a strong negative correlation with target → Lower max heart rate is often linked to heart disease.
- cp (chest pain type) shows a positive correlation with target → Certain chest pain types are associated with heart disease.
- exang (exercise-induced angina) has a strong negative correlation with target → If angina is triggered during exercise, risk is higher.
- oldpeak, ca, slope, and thal also show notable relationships with the target.

Visual Comparison of Key Features by Heart Disease Status (target)

```
# Create a figure with subplots
fig, axs = plt.subplots(1, 3, figsize=(18, 5))

# Max Heart Rate by Target
sns.boxplot(data=df, x='target', y='thalach', ax=axs[0], palette="Set2")
axs[0].set_title("Max Heart Rate vs Target")
axs[0].set_xlabel("Heart Disease (0 = No, 1 = Yes)")
axs[0].set_ylabel("Max Heart Rate (thalach)")

# Cholesterol by Target
sns.boxplot(data=df, x='target', y='chol', ax=axs[1], palette="Set1")
axs[1].set_title("Cholesterol vs Target")
axs[1].set_xlabel("Heart Disease (0 = No, 1 = Yes)")
axs[1].set_ylabel("Cholesterol (mg/dL)")

# Chest Pain Type by Target
sns.countplot(data=df, x='cp', hue='target', ax=axs[2], palette="pastel")
axs[2].set_title("Chest Pain Type vs Target")
axs[2].set_xlabel("Chest Pain Type")
axs[2].set_ylabel("Count")
axs[2].legend(title="Heart Disease")

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

↗ /tmp/ipython-input-10-4113898720.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend`

```
sns.boxplot(data=df, x='target', y='thalach', ax=axes[0], palette="Set2")
```

/tmp/ipython-input-10-4113898720.py:11: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend`

```
sns.boxplot(data=df, x='target', y='chol', ax=axes[1], palette="Set1")
```

