Summary Report

This is a summary report for the Heart Attack Analysis & Prediction Dataset.

Dataset Description

- Age : Age of the patient
- Sex : Sex of the patient
- exang: exercise induced angina (1 = yes; 0 = no)
- ca: number of major vessels (0-3)
- cp : Chest Pain type chest pain type
  - Value 1: typical angina
  - Value 2: atypical angina
  - Value 3: non-anginal pain
  - Value 4: asymptomatic
- trtbps : resting blood pressure (in mm Hg)
- chol : cholesterol in mg/dl fetched via BMI sensor
- fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
- rest_ecg : resting electrocardiographic results
  - Value 0: normal
  - Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
  - Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
- thalach : maximum heart rate achieved
- target : 0= less chance of heart attack 1= more chance of heart attack

Import pandas and matplotlib.pyplot

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

Read the Dataframe heart.csv

In [ ]:
```
from rylib.lib import readfile
readfile("heart.csv")
```

Out[ ]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

Generates Summary Statistics for heart.csv

In [ ]:
```
def summary(file_path):
    df = readfile(file_path)
    summary_stats = df.describe()
    return summary_stats

summary("heart.csv")
```

Out[ ]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Calculate the Median Value for Each Column in heart.csv

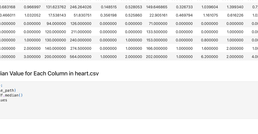
In [ ]:
```
def median(file_path):
    df = readfile(file_path)
    median_values = df.median()
    return median_values

median("heart.csv")
```

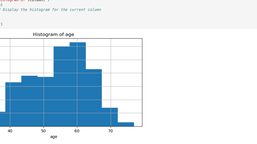
Out[ ]:
```
age         55.0
sex          1.0
cp           1.0
trtbps     130.0
chol       240.0
fbs          0.0
restecg      1.0
thalachh   153.0
exng         0.0
oldpeak      0.8
slp          1.0
caa          0.0
thall        2.0
output       1.0
dtype: float64
```

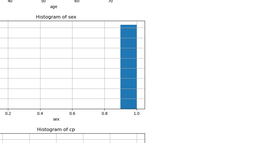
Generate Histogram for Each Column in heart.csv

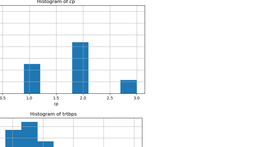
In [ ]:
```
def histogram(file_path):
    df = readfile(file_path)
    columns = df.columns

    for column in columns:
        plt.figure(figsize=(5, 4))  # Create a new figure for each column
        plt.hist(df[column])
        plt.xlabel(column)
        plt.ylabel("Frequency")
        plt.title(f"Histogram of {column}")
        plt.grid(True)
        plt.show()  # Display the histogram for the current column

histogram("heart.csv")
```

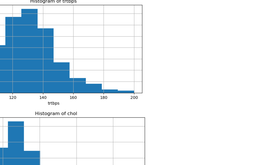
Out[ ]:

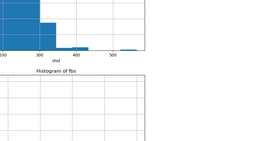


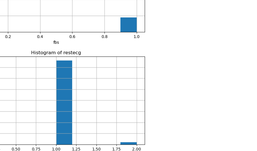


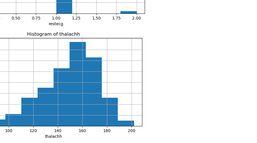


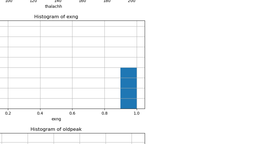


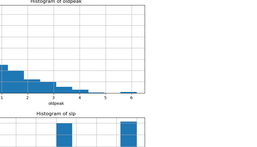


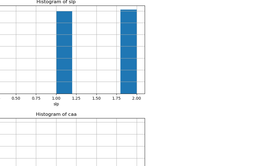


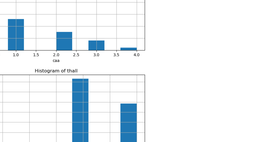












Generate Scatter Plot For Resting Blood Pressure and Age in heart.csv

In [ ]:
```
def scatter_age_blood_pressure(file_path):
    df = readfile(file_path)
    x = df.iloc[:, 0]  # 1st column (Age)
    y = df.iloc[:, 3]  # 4th column (Resting Blood pressure)
    plt.scatter(x, y, alpha=0.5, label="Data Points")
    plt.xlabel("Age")
    plt.ylabel("Resting Blood Pressure (mm Hg)")
    plt.title("Scatter Plot: Age vs. Resting Blood Pressure")
    plt.grid(True)  # Add grid lines for reference
    plt.legend()
    plt.show()

scatter_age_blood_pressure("heart.csv")
```

Out[ ]: