Lab 1

Part-1 (60 pts):

Use the heart.csv dataset present under the Lab 1 section in canvas.
Please perform the following questions in Python (jupyter notebook) and submit an ipynb file. Please submit Part-1, Part-2 in a single ipynb file.

You should submit one ipynb file and one pdf file. (Do not zip ipynb file and pdf file)

*** Please write all your explanation in jupyter notebook markdown cells (Do not write analysis or explanation of the graph by comments).**
*** All the graphs should be properly labeled.**

About the data:
Each following row contains the information of one patient. The first row is the name of the observed variables. There are 10 variables:

- sbp: Systolic blood pressure
- tobacco: Cumulative tobacco consumption, in kg
- ldl: Low-density lipoprotein cholesterol
- adiposity: Adipose tissue concentration
- famhist: Family history of heart disease (1=Present, 0=Absent)
- typea: Score on test designed to measure type-A behavior
- obesity: Obesity
- alcohol: Current consumption of alcohol
- age: Age of subject
- chd: Coronary heart disease at baseline; 1=Yes 0=No.
(chd= 0 means alive and chd= 1 means dead.)

1. Look at each variables(features) in the dataset and decide if it belongs to one of these four groups (10 pt):

1. **Nominal Categorical**: Categories with no specific order (like colors or names).
2. **Binary Categorical**: Two possible categories (like yes/no or 0/1).
3. **Discrete**: Whole numbers (like number of items).
4. **Continuous**: Numbers that can take any value within a range (like height or temperature)

2. Find the number of null values for each column. (10 pts)
3. Descriptive Analysis (40 pts / 4 pts for each sub question):
   1) Show the general descriptive statistics by using **describe** function.
   2) Find the age of the oldest person and print the people with that age by filtering
   3) Find the youngest person and print the people with that age by filtering
   4) Find the average and standard deviation of age column
   5) Find median age
   6) Draw a bar chart that represents the relationship between the deaths and ages and draw an insights from the chart (you can filter by chd==1 and draw a histogram)
   7) Find the age groups whose survival rate is the largest (Please include a graph that proves it)

8) Find a relationship between ’famhist’ and ‘chd’. Please write what you have found.
9) Get more visuals on data distributions
   i. Plot Correlation Matrix
   ii. Plot Scatter Matrix
   iii. Plot Per Column Distribution
   iiii. Plot a heat map for missing values
10) Please research and provide your opinion on additional techniques for handling null values, excluding the 'drop NA' feature.

Part 2 (40 pts)
Part 2 is for linear algebra. Please follow the directions of each question. You should write a comment for the important code lines that describe what it is.

1. Basic Matrix Multiplication (20 pts)

Write a Python function that takes two matrices as input and returns their product. Do not use built-in matrix multiplication functions such as `np.dot`**.**

```
import numpy as np

def matrix_multiply(A, B):
    """
    Multiplies two matrices A and B.

    Parameters:
    A (numpy.ndarray): First matrix.
    B (numpy.ndarray): Second matrix.

    Returns:
    numpy.ndarray: The product of matrices A and B.
    """
    # Your code here

# Example usage:

A = np.array([[12, 21], [2, 8]])
B = np.array([[13, 7], [7, 8]])
print(matrix_multiply(A, B))
```

2. Compute the Determinant (10 pts)

Write a Python function that computes the determinant of a square matrix using the `numpy.linalg` module.

```
import numpy as np

def compute_determinant(A):
    """
    Computes the determinant of a square matrix A.
```

```
    Parameters:
    A (numpy.ndarray): Square matrix.

    Returns:
    float: Determinant of the matrix A.
    """
    # Your code here

# Example usage:
A = np.array([[11, 13], [15, 17]])
print(compute_determinant(A))
```

3. Solve a System of Linear Equations (10 pts)

Write a function that solves a system of linear equations given in matrix form, `Ax = b`, where `A` is a coefficient matrix and `b` is a constant vector. Use `numpy.linalg.solve()`.

```
import numpy as np

def solve_linear_system(A, b):
    """
    Solves the system of linear equations Ax = b.

    Parameters:
    A (numpy.ndarray): Coefficient matrix.
    b (numpy.ndarray): Constant vector.

    Returns:
    numpy.ndarray: Solution vector x.
    """
    # Your code here

# Example usage:
A = np.array([[3, 1], [1, 2]])
b = np.array([9, 8])
print(solve_linear_system(A, b))
```

What to submit to Canvas

Before submitting your notebook in Canvas perform the following to make sure it works:

1. Clear all the out
2. Restart the kernel
3. Run all the cells
4. Please include a pdf version of your ipynb file.