TEAM ARDRA

SOFTWARE SUMMER TRAINING

ASSIGNMENT 1: BASICS OF DATA ANALYSIS

SET 6

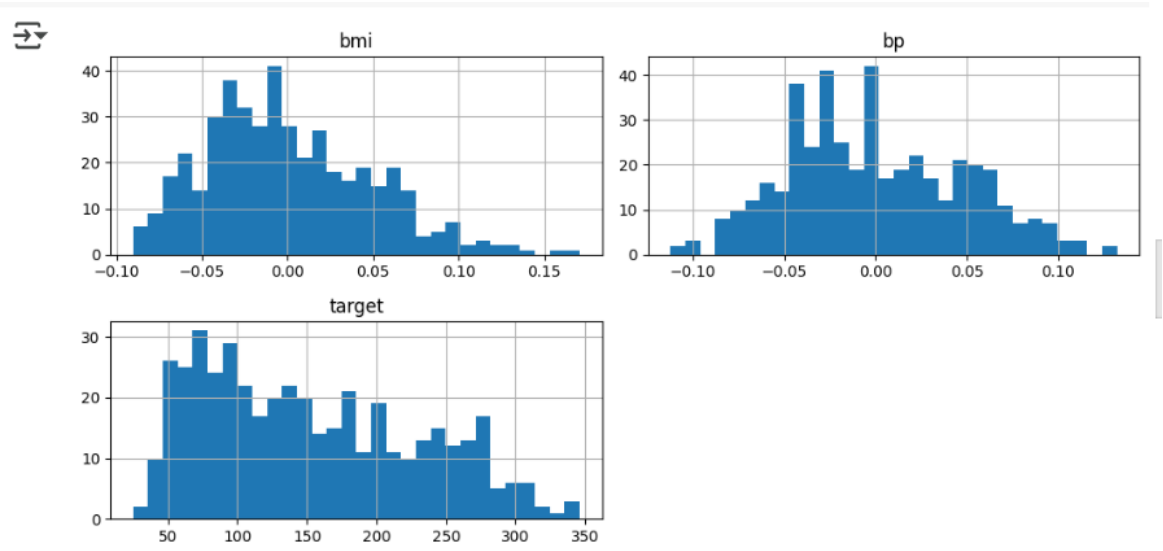
Q1. Dataset: Diabetes Dataset (Scikit-learn)

Google colab link for code:

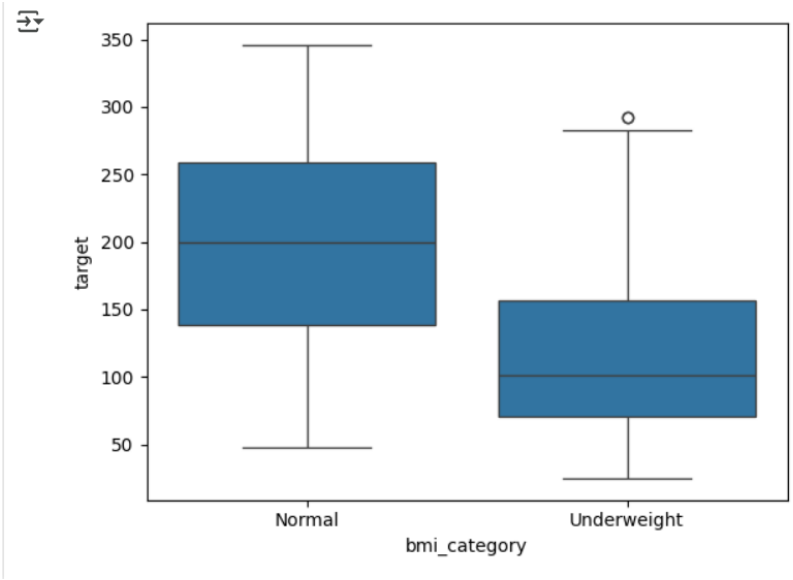
https://colab.research.google.com/drive/1SWH6VD3nC7rpgTAyUkDK7x4NVgERbZ_k?usp=sharing

Visualizations:

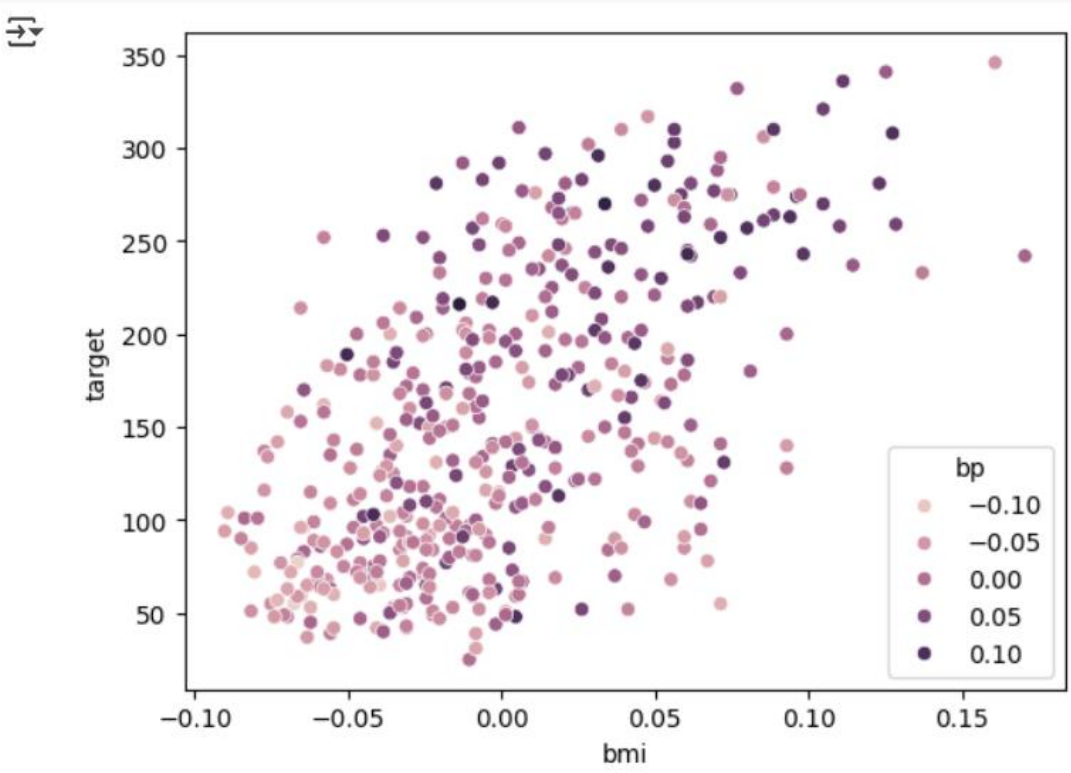
1. Histogram



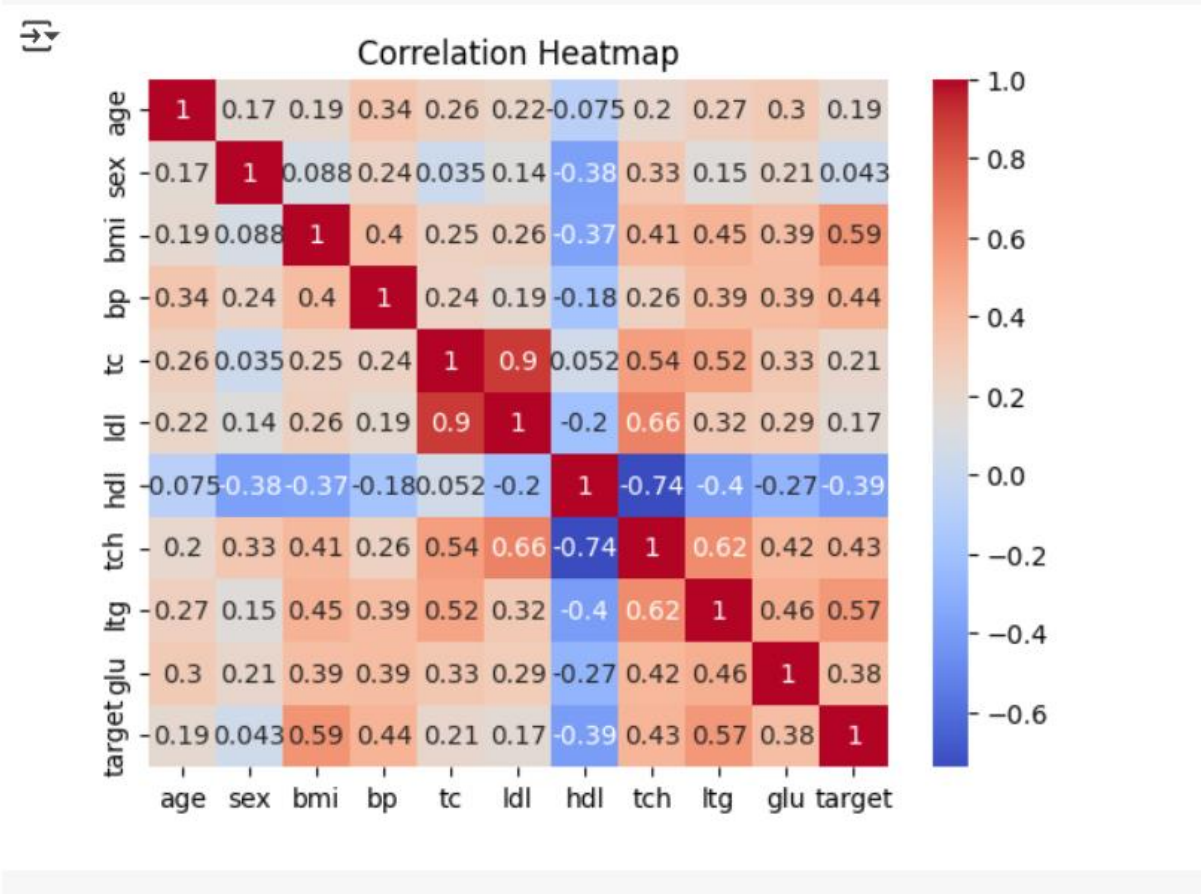
2. Box Plot



3. Scatter Plot

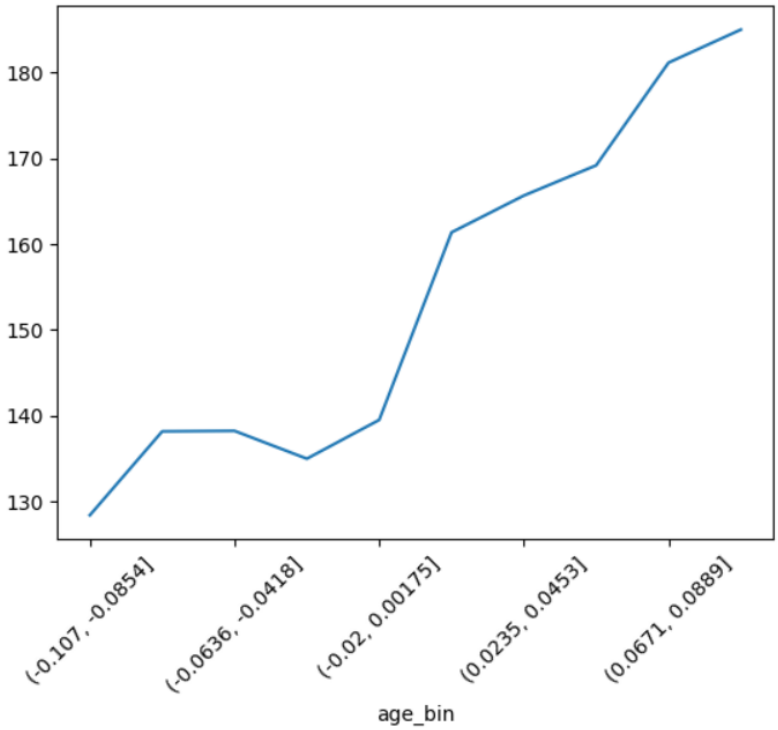


4. Heatmap

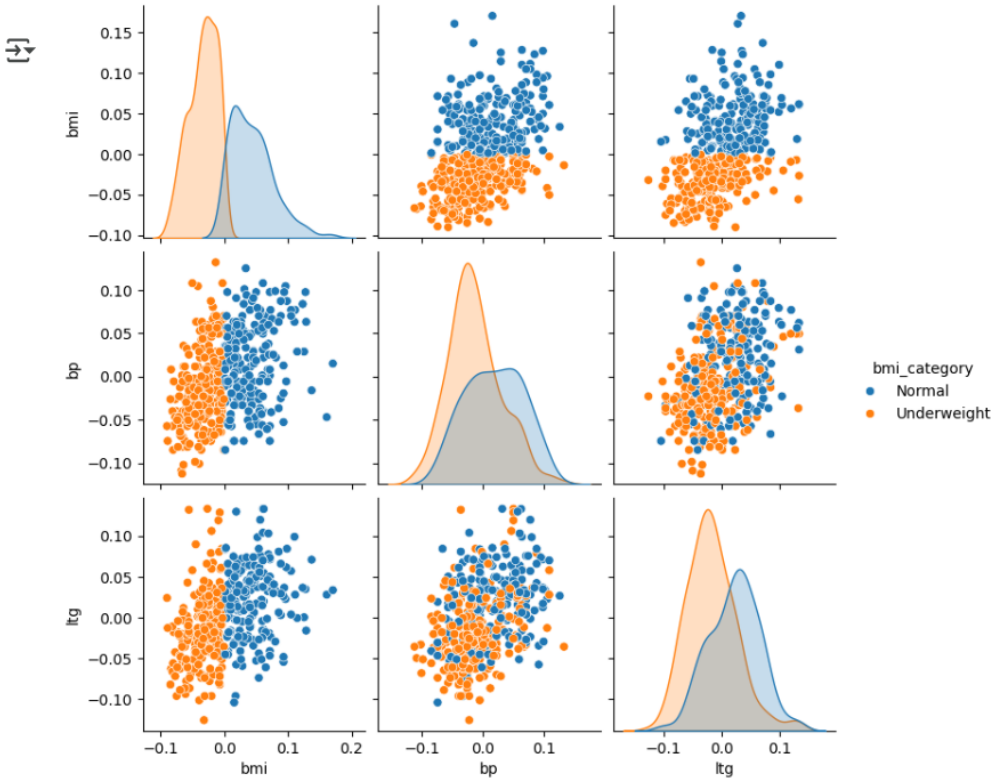

Correlation Heatmap
age 1 0.17 0.19 0.34 0.26 0.22 -0.075 0.2 0.27 0.3 0.19
sex 0.17 1 0.088 0.24 0.035 0.14 -0.38 0.33 0.15 0.21 0.043
bmi 0.19 0.088 1 0.4 0.25 0.26 -0.37 0.41 0.45 0.39 0.59
bp 0.34 0.24 0.4 1 0.24 0.19 -0.18 0.26 0.39 0.39 0.44
tc 0.26 0.035 0.25 0.24 1 0.9 0.052 0.54 0.52 0.33 0.21
ldl 0.22 0.14 0.26 0.19 0.9 1 -0.2 0.66 0.32 0.29 0.17
hdl -0.075 -0.38 -0.37 -0.18 0.052 -0.2 1 -0.74 -0.4 -0.27 -0.39
tch 0.2 0.33 0.41 0.26 0.54 0.66 -0.74 1 0.62 0.42 0.43
ltg 0.27 0.15 0.45 0.39 0.52 0.32 -0.4 0.62 1 0.46 0.57
glu 0.3 0.21 0.39 0.39 0.33 0.29 -0.27 0.42 0.46 1 0.38
target 0.19 0.043 0.59 0.44 0.21 0.17 -0.39 0.43 0.57 0.38 1
age sex bmi bp tc ldl hdl tch ltg glu target
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

5. Line plot

```
<ipython-input-14-6f00ff6c01fd>:3: FutureWarning: The default of observed=False is depr
  df.groupby('age_bin')['target'].mean().plot(kind='line')
```



6. Pair plot



Q2. Simulate Epidemic Spread (SIR Model)

TERMINOLOGIES:

1. Epidemiology: is the study of how diseases spread, who they affect, and how they can be controlled or prevented.
2. Simulate for 100 Time Steps: run the model for 100 steps from Susceptible (S) → Infected (I)→ Recovered (R), then analyse how the disease had spread and declined over time.
   Each step in time can be any no. of days, weeks, or any unit of time.
3. SIR Model: SIR model is a way to simulate how a disease spreads through a population over time. It divides people into three groups as
   S (Susceptible)
   I (Infected)
   R (Recovered)
4. Simulate epidemic spread: Use a mathematical model to imitate how a disease would spread through a population over time.

GIVEN PARAMETERS:

Total population (N) = 1000
Initial infected (I_o) = 1
Initial recovered (R_o) = 0
Initial susceptible (S_o) = N - I_o - R_o = 999
Infection rate (β) = 0.3
Recovery rate (γ) = 0.1
Simulate for 100 time steps

STIMULATION STEPS:

1. At each time step, update the compartments using the SIR model equations:

```
new_infected = (beta * S * I) / N
# new_infected = (infection rate*susceptible*infected)/ total
new_recovered = gamma * I
# new_recovered = recovery rate*infected

S -= new_infected   #update S
I += new_infected - new_recovered    #update I
R += new_recovered    # update R
```

2. Store S, I, R values at each step in a list or NumPy array.
3. Ensure values remain within bounds (non-negative, sum ≈ N

HOW DOES THE DISEASE SPEAD (AT EACH STEP):

1. Let each step in time can be 1 day. That is the data is studied for 100 consecutive days.
2. At every step in time, 2 things are possible:
   some susceptible people get infected
   some infected people recover
3. This is calculated by:
   How many ppl get newly infected at every step:
   new_infected = (β * S * I) / N
   (it directly depends on no. of currently susceptible ppl S, infected ppl I and total population N)
   How many currently infected ppl recover:
   new_recovered = γ * I
   (it directly depends on currently infected ppl only)

SOLUTION:

Aim: To simulate how the number of S, I, and R individuals changes over time using the SIR model

Basic Idea:

1. Start by considering initial values gives as:
   N=1000, S= 999, I=1, R=0
2. For each time step (from 1 to 100):
   i. Calculate how many people get newly infected.
   ii. Calculate how many infected people recover.
   iii. Update the values of S, I, R using those numbers.
   iv. Make sure values don't go negative or exceed the total population, ie check boundary conditions.
   v. Store the values of S, I, and R at each time step into a list/array.
3. At the end, we will have a list/array containing how many people were susceptible, infected, or recovered at each step.

Methodology:
(To store S, I, R values at each time step into a list/array)

Meth1: Storing in lists

1. Create empty lists to store values from time step 1 to 100
   S_values =[]
   I_values=[]
   R_values=[]
2. Set initial values according to given as:
   N=1000, S= 999, I=1, R=0, steps=100
   beta= 0.3, gamma=0.1
3. For each i in steps go through the loop as:

```
for i in range(0, steps):
    # save current values
    S_list.append(S)
    I_list.append(I)
    R_list.append(R)

    # calculate new infected and new recovered
    new_infected = (beta * S * I) / N
    new_recovered = gamma * I

    # update values
    S -= new_infected
    I += new_infected - new_recovered
    R += new_recovered
```

Meth 2: Can store all S, I, R values into numpy array also for easier data visualization

```
import numpy as np

# create arrays to hold values, initialize all to 0
S_array = np.zeros(100)
I_array = np.zeros(100)
R_array = np.zeros(100)
```

```
# set initial values
S = 999, I = 1, R = 0

# for loop
for t in range(100):
   S_array[t] = S
   I_array[t] = I
   R_array[t] = R

   new_infected = (0.3 * S * I) / 1000
   new_recovered = 0.1 * I

   S -= new_infected
   I += new_infected - new_recovered
   R += new_recovered
```

-by Ananya Singh
(24BAI151)