

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Інститут телекомунікацій, радіоелектроніки та електронної техніки  
кафедра «Телекомунікацій»



Звіт з лабораторної роботи №1  
з дисципліни «Теорія алгоритмів та структур даних»

Підготував:  
ст.групи ТР-31  
Гейниш Р.Т.

Прийняв:  
Андрущак В.С.

Львів 2024р.

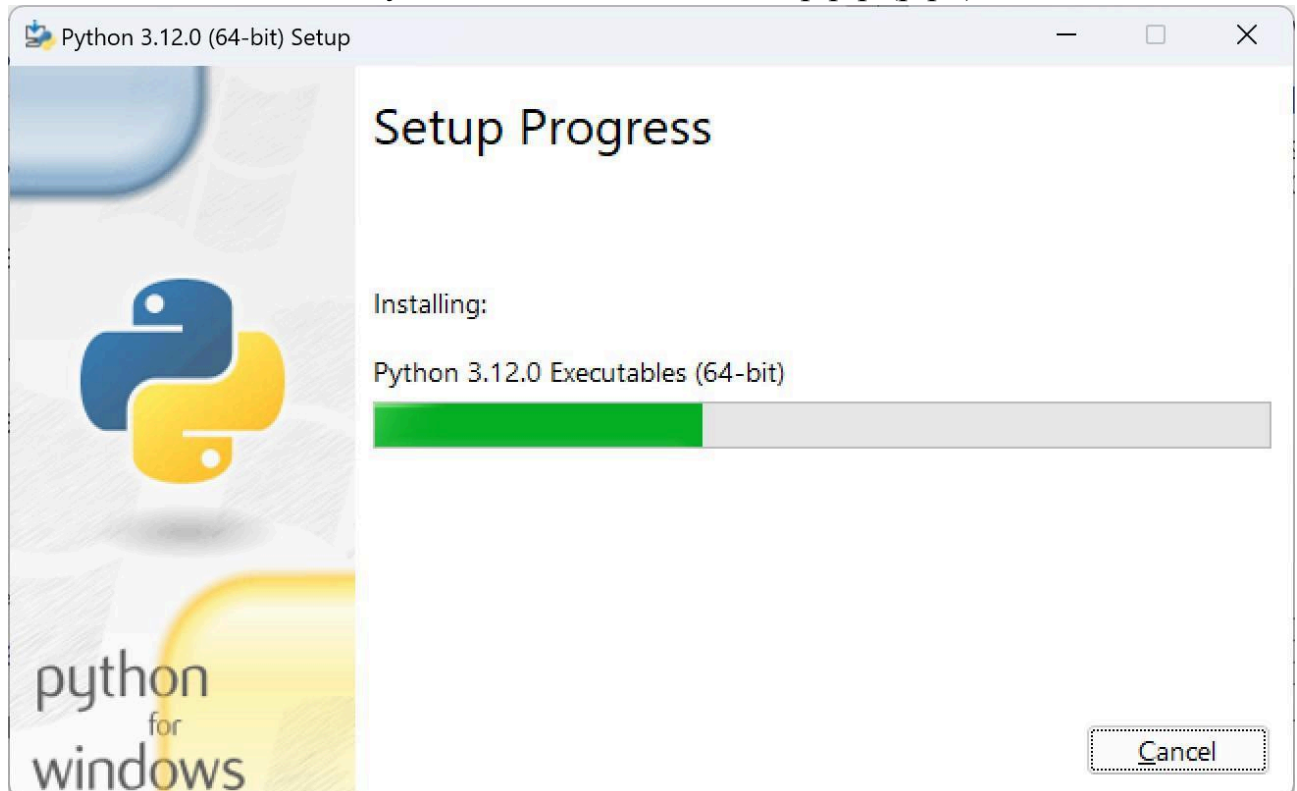
**Тема роботи:** Налаштування середовища роботи Python3 та Jupyter Notebook

**Мета роботи:** налаштувати робоче середовище, вивчити та дослідити основні технічні елементи для дослідження даних та алгоритм

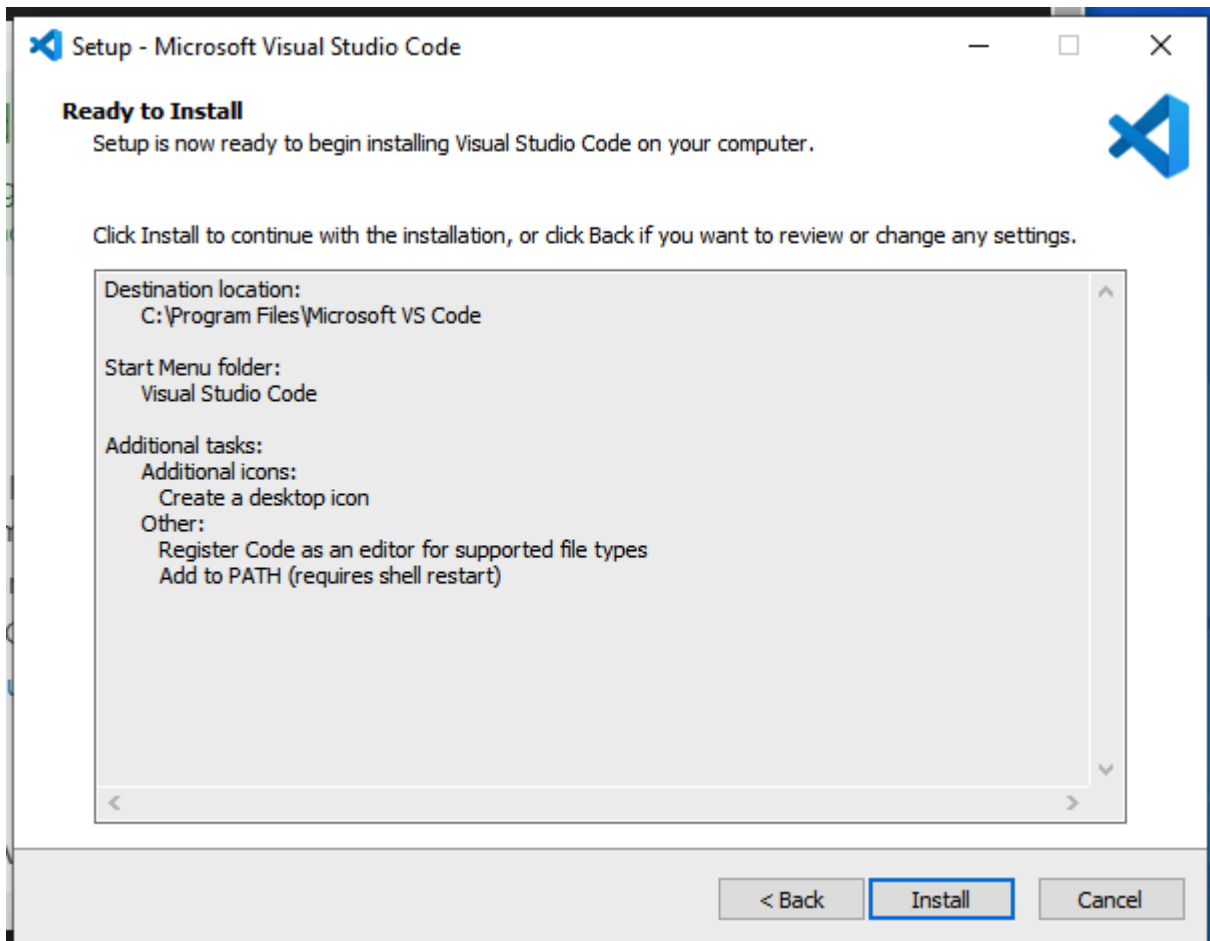
## Завдання до виконання

### Завдання 1. Налаштувати середовище роботи

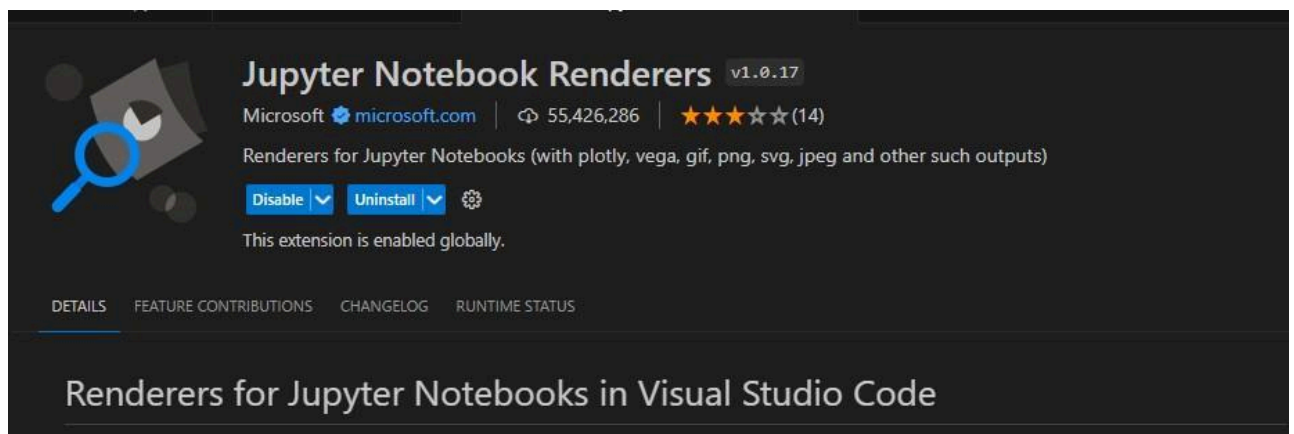
Встановлюємо Python3, пакетний менеджер pip (pip3) та VSCode.



```
C:\Users\suraj>python get-pip.py
Collecting pip
  Downloading pip-22.1.2-py3-none-any.whl (2.1 MB)
    ----- 2.1/2.1 MB 4.3 MB/s eta 0:00:00
Collecting wheel
  Using cached wheel-0.37.1-py2.py3-none-any.whl (35 kB)
Installing collected packages: wheel, pip
  Attempting uninstall: pip
    Found existing installation: pip 22.0.4
    Uninstalling pip-22.0.4:
      Successfully uninstalled pip-22.0.4
Successfully installed pip-22.1.2 wheel-0.37.1
```



Після чого встановлюємо jupyter notebook, знайшовши його у вкладці extensions

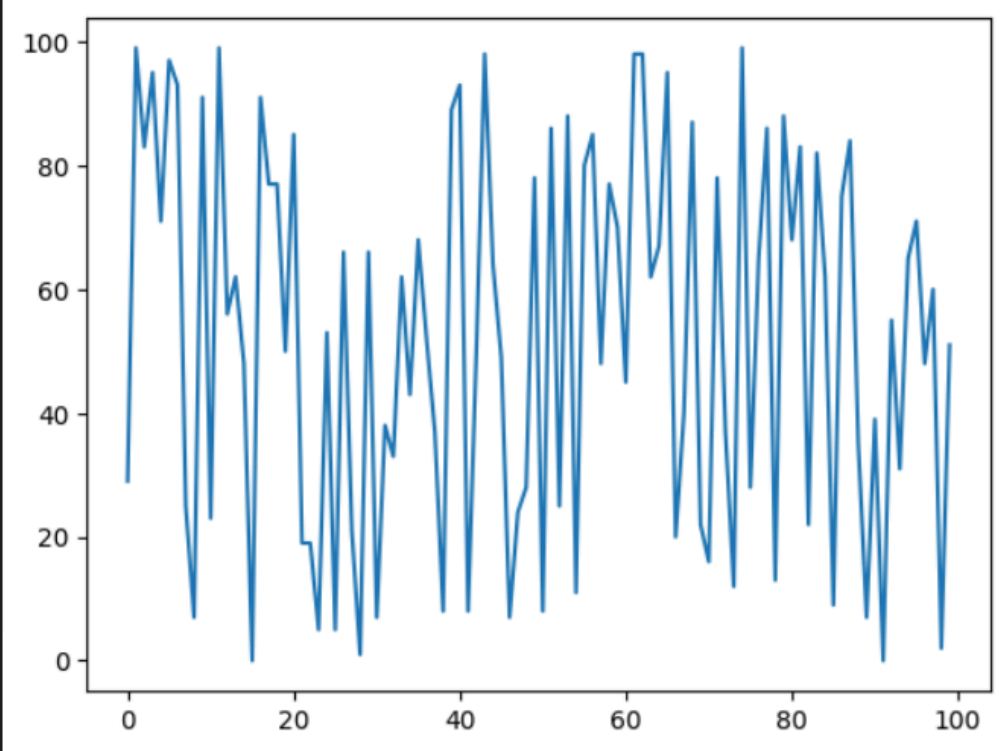


За допомогою пакетного менеджера pip інсталяціюмо бібліотеки numpy, pandas, matplotlib ввівши `pip install <назва бібліотеки>`

Перевіряємо правильність налаштування середовища та бібліотек Підключаємо бібліотеки та виводимо у консоль “Hello, world!”

```
▶ ▼  
print("Hello world")  
[114] ✓ 0.0s  
... Hello world
```

Генеруємо масив з 100 випадкових елементів та візуалізуємо дані

```
▶ ▼  
import numpy as np  
import matplotlib.pyplot as plt  
from numpy import random  
  
x=random.randint(100, size=(100))  
print(x)  
plt.plot(x) # https://www.w3schools.com/python/matplotlib\_plotting.asp  
plt.show()  
[116] ✓ 0.1s  
... [29 99 83 95 71 97 93 25 7 91 23 99 56 62 48 0 91 77 77 50 85 19 19 5  
53 5 66 21 1 66 7 38 33 62 43 68 52 37 8 89 93 8 50 98 64 49 7 24  
28 78 8 86 25 88 11 80 85 48 77 70 45 98 98 62 67 95 20 40 87 22 16 78  
37 12 99 28 65 86 13 88 68 83 22 82 62 9 75 84 35 7 39 0 55 31 65 71  
48 60 2 51]  
...  

```

## Завдання 2. Дослідження датасету

Підключаємо необхідні бібліотеки та завантажуємо датасет. Прочитавши опис до датасету називаємо відповідно колонки.

```
import pandas as pd

url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"

df = pd.read_csv(url)

columns = ["Times Pregnant", "Plasma Glucose", "Diastolic Blood Pressure",
           "Skin Thickness", "Serum Insulin", "Body mass",
           "Diabetes Pedigree", "Age", "Class Variable"]

info = pd.read_csv(url, names=columns)

print(info.shape)
print(info.isnull().sum()) # https://medium.com/analytics-vidhya/python-finding-missing-values-in-a-data-frame-3030a
```

Визначаємо розмір датасету

```
info = pd.read_csv(url, names=columns)

print(info.shape)
```

Отримуємо наступне значення у виводі, де 768 - це кількість рядків, а 9 - кількість стовпців:

```
(768, 9))
```

Наступним кроком визначаємо пропущені дані. Для цього проведемо перевірку на пусті значення (NaN) за допо `isnull()` та виводимо загальну суму NaN значень з кожної колонки

```
print(info.isnull().sum()) #
```

Отримуємо:

```
Times Pregnant      0
Plasma Glucose      0
Diastolic Blood Pressure  0
Skin Thickness      0
Serum Insulin       0
Body mass           0
Diabetes Pedigree   0
Age                 0
Class Variable      0
dtype: int64
```

Отже, даних зі значенням NaN немає. А описі до датасету вказано, що пропущені дані є, перевіряємо колонки, які можуть містити значення "0", але не є логічним, як наприклад колонка `Plasma Glucose`, що містить дані про

концентрацію глюкози в плазмі через 2 години в оральному тесті на толерантність до глюкози, норма становить зазвичай 140 мг/д

```
import pandas as pd
import numpy as np

url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"

df = pd.read_csv(url)

columns = ["Times Pregnant", "Plasma Glucose", "Diastolic Blood Pressure",
           "Skin Thickness", "Serum Insulin", "Body mass",
           "Diabetes Pedigree", "Age", "Class Variable"]

info = pd.read_csv(url, names=columns)

#print(info.columns)

#pregnant = info["Times Pregnant"] # https://pandas.pydata.org/docs/getting_started/intro_tutorials/03_subset_data.
#print(np.where(pregnant<=-1)) # https://stackoverflow.com/questions/13869173/numpy-find-index-of-the-elements-with
#glucose = info["Plasma Glucose"]
#print(np.where(glucose<=0))
💡
missing_data = (info[columns] == 0).sum()

print(missing_data)
```

Отримуємо:

```
Times Pregnant      111
Plasma Glucose       5
Diastolic Blood Pressure  35
Skin Thickness      227
Serum Insulin       374
Body mass           11
Diabetes Pedigree     0
Age                 0
Class Variable      500
dtype: int64
```

Здійснюємо обрахунок середнього арифметичного, дисперсії, середнє квадратичне відхилення для кожної із колонок набору даних:

```
import pandas as pd
import numpy as np

url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"

df = pd.read_csv(url)

columns = ["Times Pregnant", "Plasma Glucose", "Diastolic Blood Pressure",
           "Skin Thickness", "Serum Insulin", "Body mass",
           "Diabetes Pedigree", "Age", "Class Variable"]
💡
info = pd.read_csv(url, names=columns)

# num_of_rows = len(info)

# sum = (info[columns]).sum()
# avg_sum = sum/num_of_rows

print("Average:", "\n", info.mean(), "\n") #https://www.geeksforgeeks.org/create-the-mean-and-standard-deviation-of-
print("Variance:", "\n", info.var(), "\n") #https://www.w3schools.com/python/pandas/ref_df_std.asp#:~:text=The%20var
print("Standart Deviation", "\n", info.std()) #https://www.geeksforgeeks.org/create-the-mean-and-standard-deviation-
```

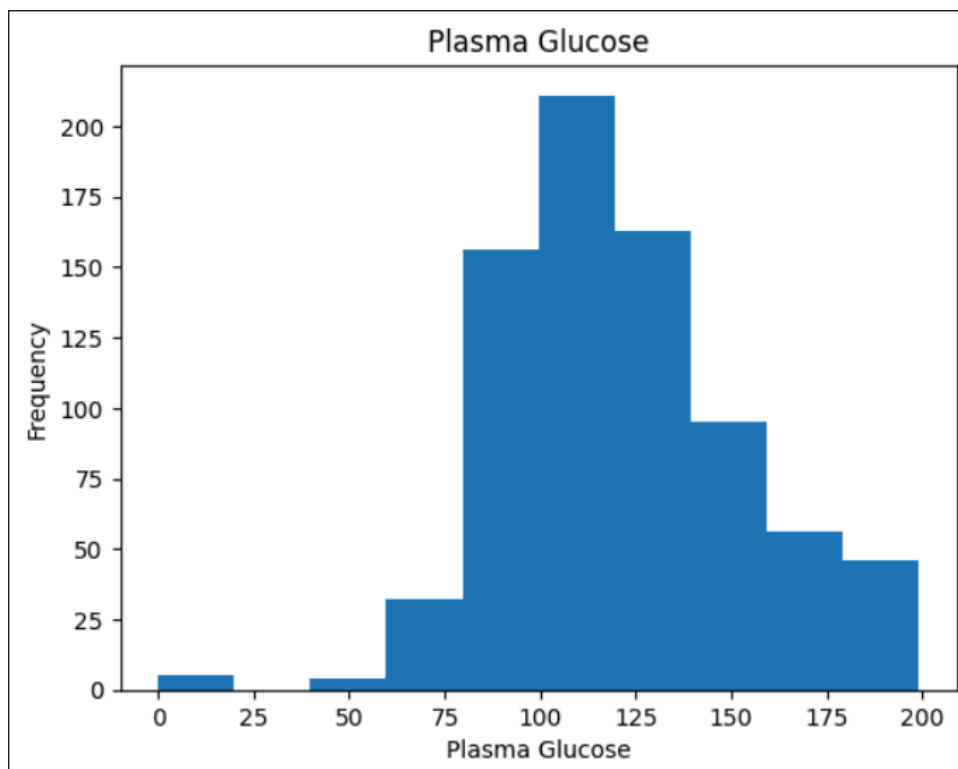
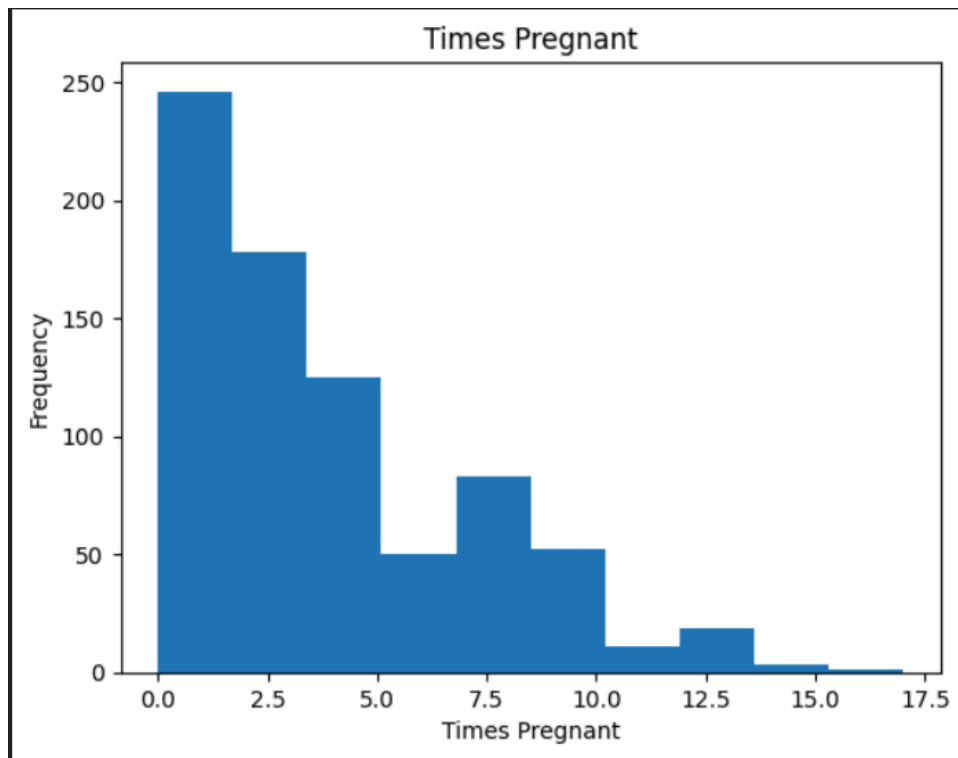
Отримуємо:

```
Average:
| Times Pregnant      3.845052
Plasma Glucose      120.894531
Diastolic Blood Pressure  69.105469
Skin Thickness      20.536458
Serum Insulin       79.799479
Body mass           31.992578
Diabetes Pedigree    0.471876
Age                 33.240885
Class Variable       0.348958
dtype: float64
```

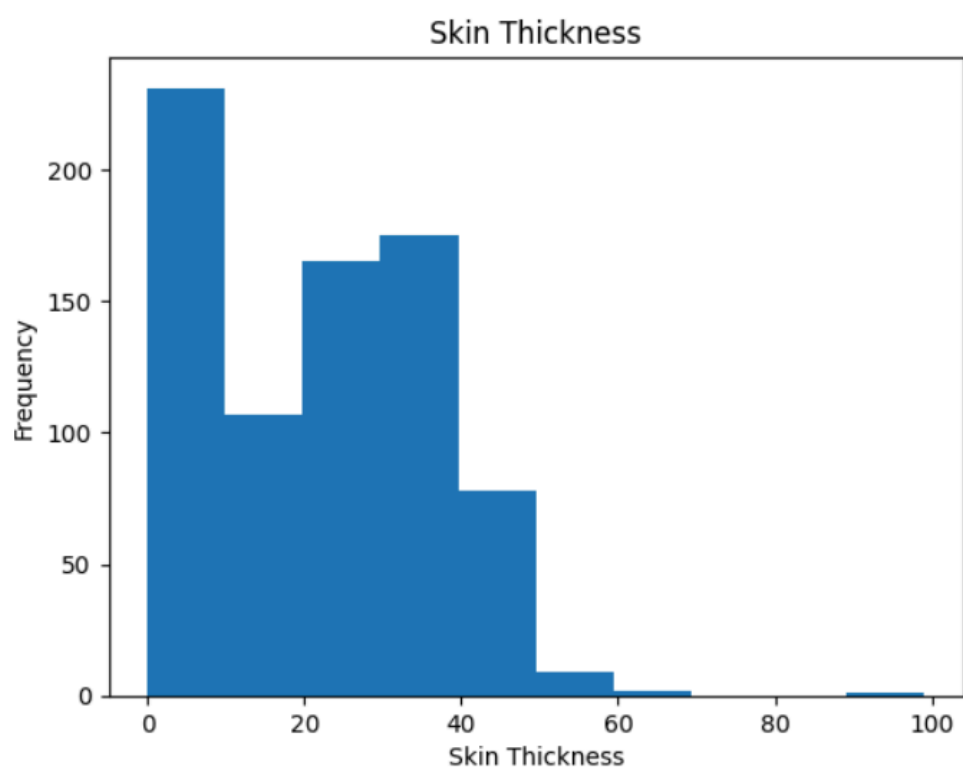
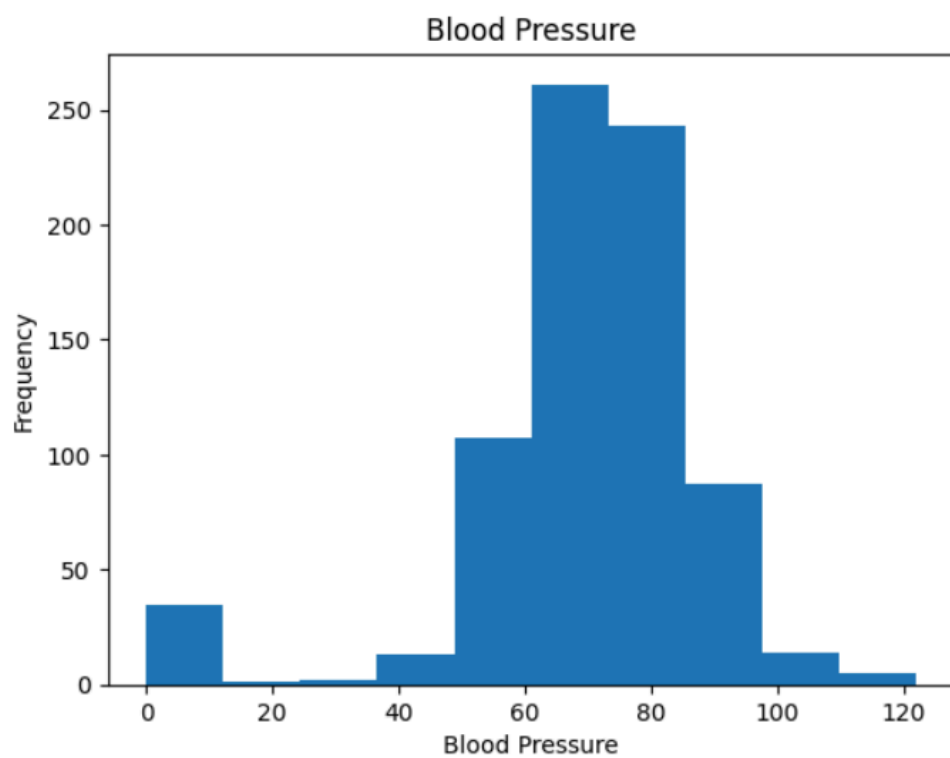
```
Variance:
| Times Pregnant      11.354056
Plasma Glucose      1022.248314
Diastolic Blood Pressure  374.647271
Skin Thickness      254.473245
Serum Insulin       13281.180078
Body mass           62.159984
Diabetes Pedigree    0.109779
Age                 138.303046
Class Variable       0.227483
dtype: float64
```

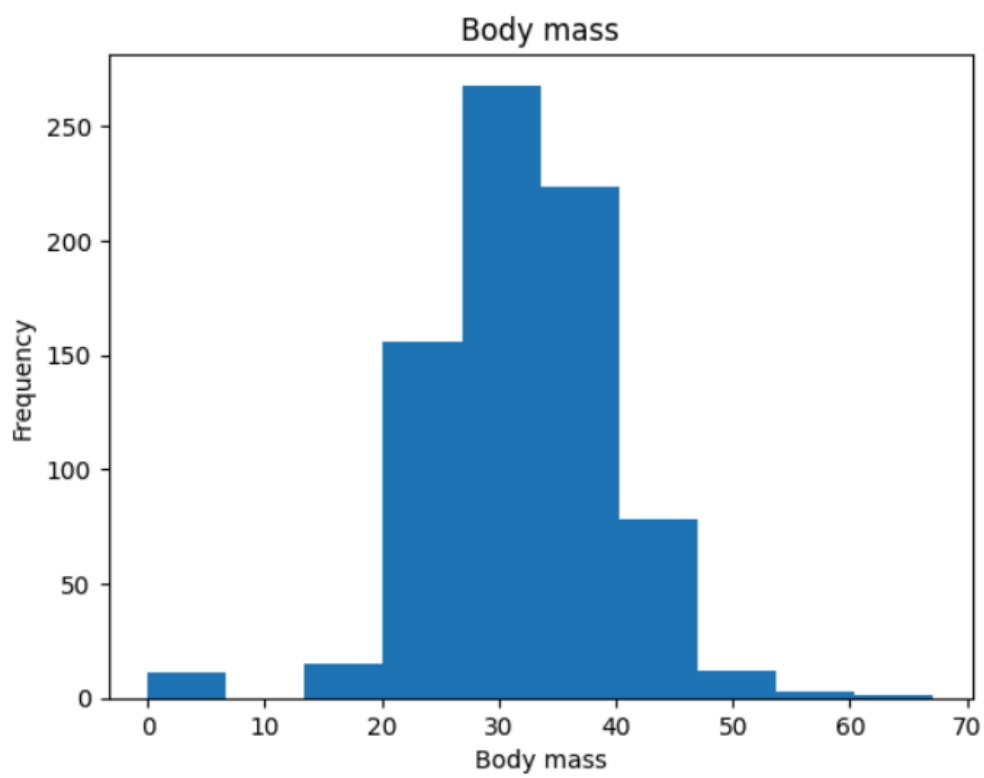
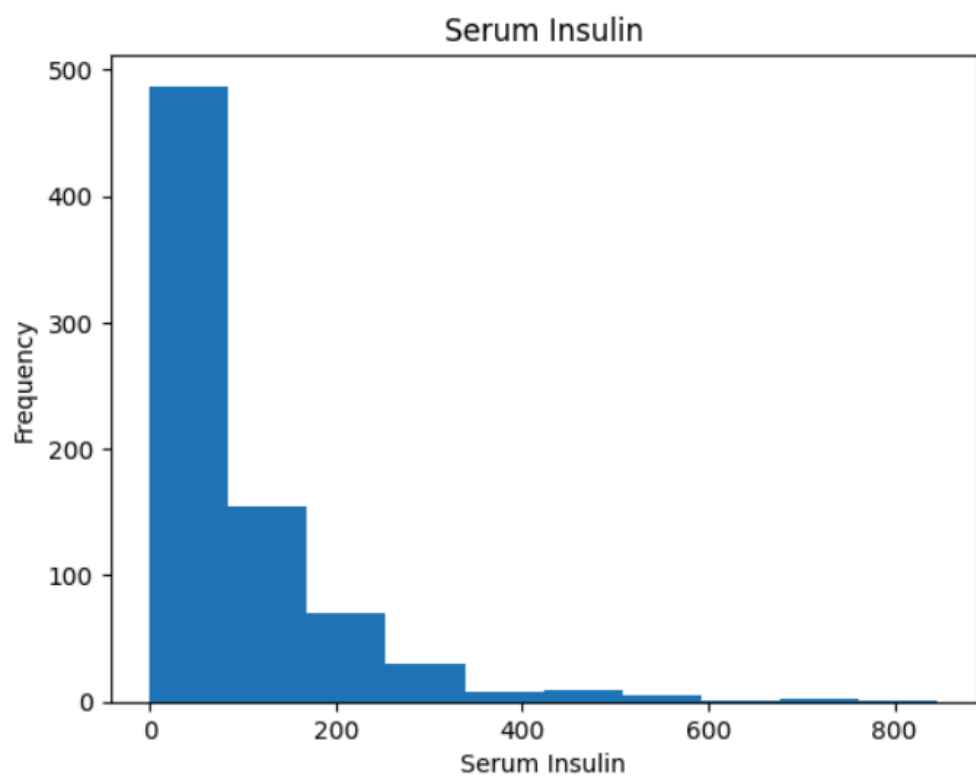
```
Standart Deviation
| Times Pregnant      3.369578
Plasma Glucose       31.972618
Diastolic Blood Pressure  19.355807
Skin Thickness       15.952218
Serum Insulin        115.244002
Body mass            7.884160
Diabetes Pedigree     0.331329
Age                  11.760232
Class Variable        0.476951
dtype: float64
```

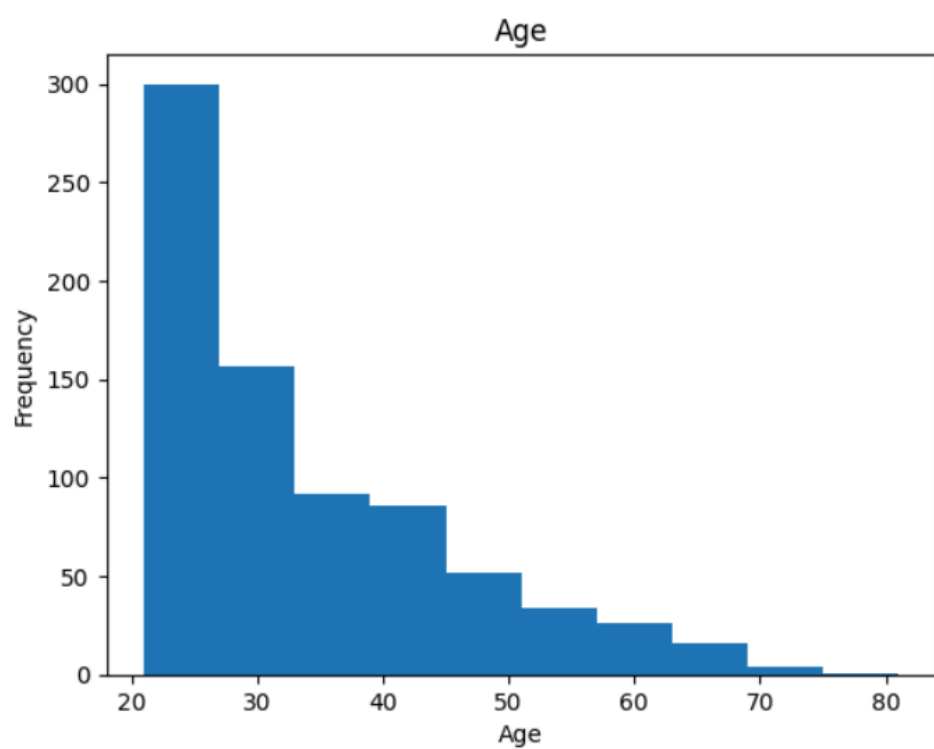
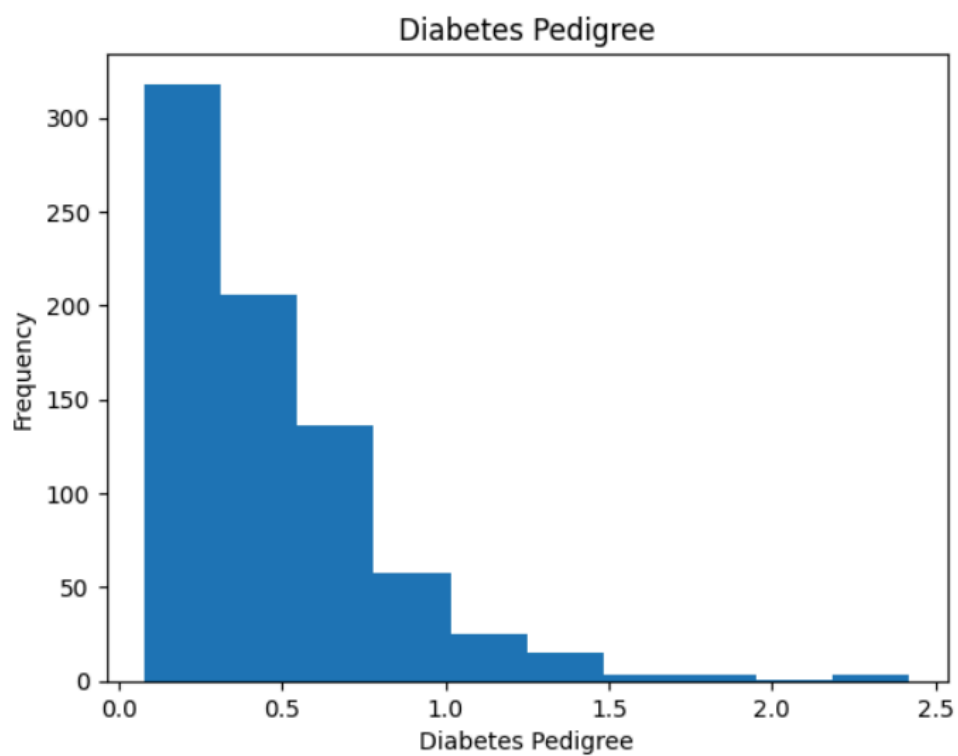
Візуалізімо дані за допомогою гістограм:

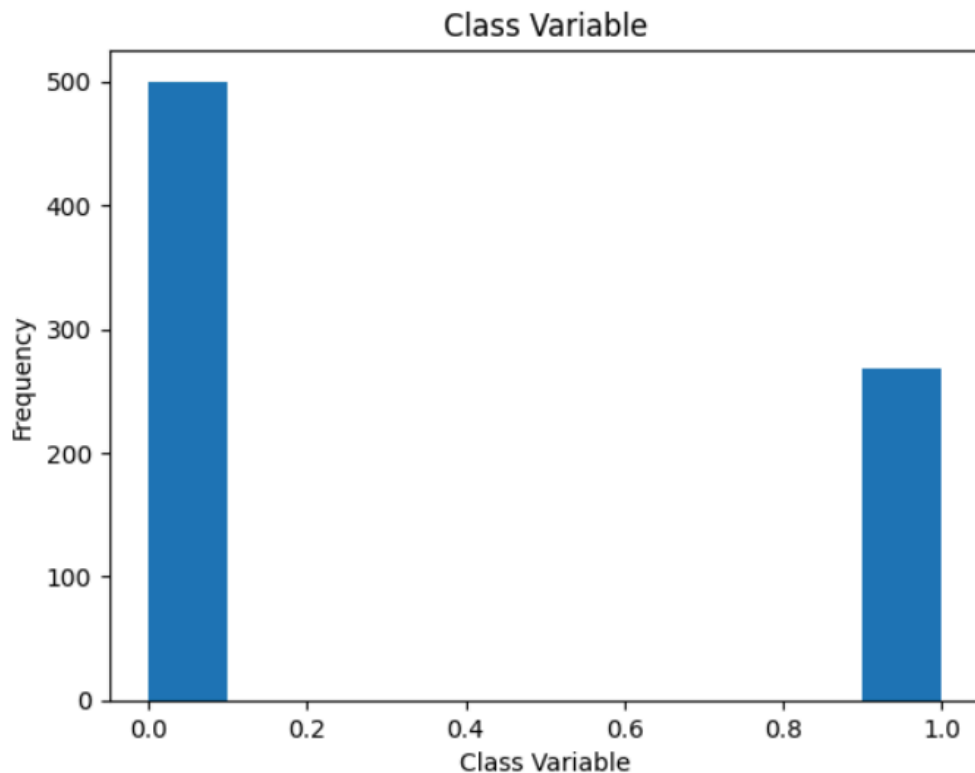












## Висновки:

Отже, на основі даних можна зробити загальні висновки. Можна помітити, що більшість пацієнтів мають менше ніж 5 вагітностей. Більшість пацієнтів також мають нормальне значення рівня глюкози крові, дані зосереджені в діапазоні від 100 до 150, (середнє значення становить 121.6) Дані про артеріальний тиск показують, що більшість пацієнтів мали тиск від 60 до 80. Розподіл товщини шкіри трицепса може бути корисним показником жирової маси, у більшості цей показник був рівний 20-40 мм.

Середнє значення індексу маси тіла (32.46) перевищує діапазон норми (18.5 до 24.9). Вибірка складалася з людей середній вік яких становить 33.2 роки. 500 людей мали негативний тест на діабет, 268 мали позитивний тест.

