Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Лабораторна роботи №8

з дисципліни «Спеціалізовані мови програмування»

на тему

«Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python»

Виконав:

Бабич О. Ю.

Перевірив:

Щербак С. С.

Львів 2024

**Мета:** Розробка додатка для візуалізації CSV-наборів даних за допомогою Matplotlib та базових принципів ООП (наслідування, інкапсуляція, поліморфізм)

**План роботи**

Завдання 1: Вибір CSV-набору даних

Оберіть CSV-набір даних, який ви хочете візуалізувати. Переконайтеся, що він містить відповідні дані для створення змістовних візуалізацій.

Завдання 2: Завантаження даних з CSV

Напишіть код для завантаження даних з CSV-файлу в ваш додаток Python. Використовуйте бібліотеки, такі як Pandas, для спрощення обробки даних.

Завдання 3: Дослідження даних

Визначте екстремальні значення по стовцям

Завдання 4: Вибір типів візуалізацій

Визначте, які типи візуалізацій підходять для представлення вибраних наборів даних. Зазвичай це може бути лінійні графіки, стовпчикові діаграми, діаграми розсіювання, гістограми та секторні діаграми.

Завдання 5: Підготовка даних

Попередньо обробіть набір даних за необхідністю для візуалізації. Це може включати виправлення даних, фільтрацію, агрегацію або трансформацію.

Завдання 6: Базова візуалізація

Створіть базову візуалізацію набору даних, щоб переконатися, що ви можете відображати дані правильно за допомогою Matplotlib. Розпочніть з простої діаграми для візуалізації однієї змінної.

Завдання 7: Розширені візуалізації

Реалізуйте більш складні візуалізації, виходячи з характеристик набору. Поекспериментуйте з різними функціями Matplotlib та налаштуваннями.

Завдання 8: Декілька піддіаграм

Навчіться створювати кілька піддіаграм в межах одного малюнка для відображення декількох візуалізацій поруч для кращого порівняння.

Завдання 9: Експорт і обмін

Реалізуйте функціональність для експорту візуалізацій як зображень (наприклад, PNG, SVG) або інтерактивних веб-додатків (наприклад, HTML)

**Реалізація:**

**Папка classes, файл base\_visualizer.py:**

import matplotlib.pyplot as plt

from abc import ABC, abstractmethod

class BaseVisualizer(ABC):

def \_\_init\_\_(self, data):

self.data = data

@abstractmethod

def plot(self):

pass

def save\_plot(self, filename):

plt.savefig(filename)

print(f"Файл збережено як {filename}")

plt.close()

def show\_plot(self):

plt.show()

**Папка classes, файл data\_explorer.py:**

class DataExplorer:

def \_\_init\_\_(self, data):

self.data = data

def print\_extreme\_values(self):

print("Максимальні значення:\n", self.data.max())

print("Мінімальні значення:\n", self.data.min())

**Папка classes, файл data\_loader.py:**import pandas as pd

class DataLoader:

def \_\_init\_\_(self, filepath):

self.filepath = filepath

def load\_data(self):

data = pd.read\_csv(self.filepath)

return data

**Папка classes, файл histogram\_visualizer.py:**

import matplotlib.pyplot as plt

from classes.base\_visualizer import BaseVisualizer

class HistogramVisualizer(BaseVisualizer):

def plot(self):

if 'y\_column' not in self.data.columns:

print("Дані для візуалізації не знайдені!")

return

plt.figure()

plt.hist(self.data['y\_column'], bins=10, color='g', edgecolor='black')

plt.title("Гістограма")

plt.xlabel("Значення")

plt.ylabel("Частота")

self.save\_plot("histogram\_plot.png")

**Папка classes, файл error\_handler.py:**

class ErrorHandler:

def handle(self, error):

print(f"Помилка: {error}")

**Папка classes, файл line\_plot\_visualizer.py:**

import matplotlib.pyplot as plt

from classes.base\_visualizer import BaseVisualizer

class LinePlotVisualizer(BaseVisualizer):

def plot(self):

if 'x\_column' not in self.data.columns or 'y\_column' not in self.data.columns:

print("Дані для візуалізації не знайдені!")

return

plt.figure()

plt.plot(self.data['x\_column'], self.data['y\_column'], marker='o', color='b', linestyle='-')

plt.title("Лінійний графік")

plt.xlabel("X-вісь")

plt.ylabel("Y-вісь")

plt.grid(True)

self.save\_plot("line\_plot.png")

**Папка classes, файл subplots.py:**

import matplotlib.pyplot as plt

class Subplots:

def \_\_init\_\_(self, data):

self.data = data

def create\_subplots(self):

fig, axs = plt.subplots(1, 2, figsize=(10, 5))

axs[0].plot(self.data['x\_column'], self.data['y\_column'], marker='o', color='b', linestyle='-')

axs[1].hist(self.data['y\_column'])

plt.show()

**Файл main.py:**

from classes.data\_loader import DataLoader

from classes.data\_explorer import DataExplorer

from classes.line\_plot\_visualizer import LinePlotVisualizer

from classes.histogram\_visualizer import HistogramVisualizer

from classes.subplots import Subplots

def main():

# Завантаження даних

loader = DataLoader("data.csv")

data = loader.load\_data()

# Дослідження даних

explorer = DataExplorer(data)

explorer.print\_extreme\_values()

# Вибір візуалізації

visualizer = LinePlotVisualizer(data) # Використовуємо лінійний графік

visualizer.plot()

# Інша візуалізація

visualizer\_hist = HistogramVisualizer(data) # Використовуємо гістограму

visualizer\_hist.plot()

# Декілька піддіаграм

subplots = Subplots(data)

subplots.create\_subplots()

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Файл data.csv:**

x\_column,y\_column

1,5

2,27

3,20

4,25

5,35

6,45

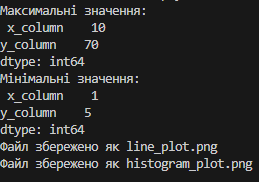
7,50

8,55

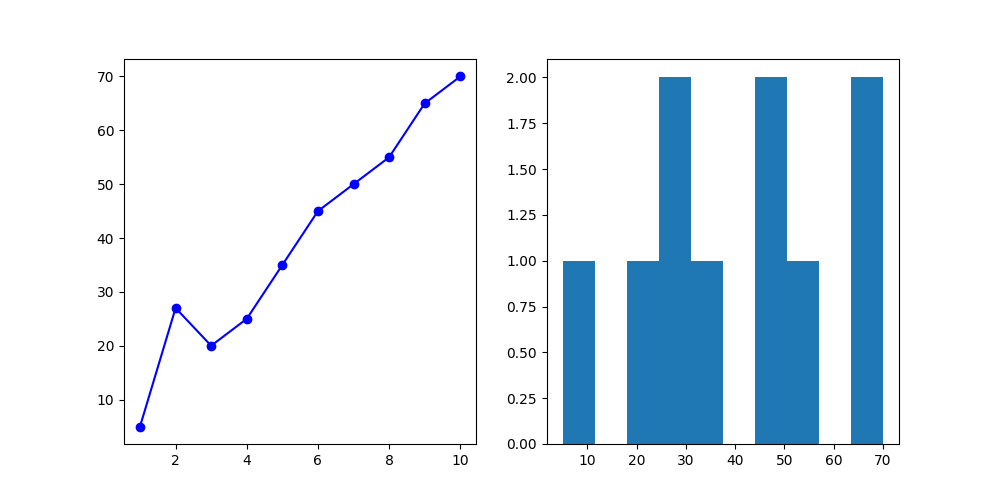
9,65

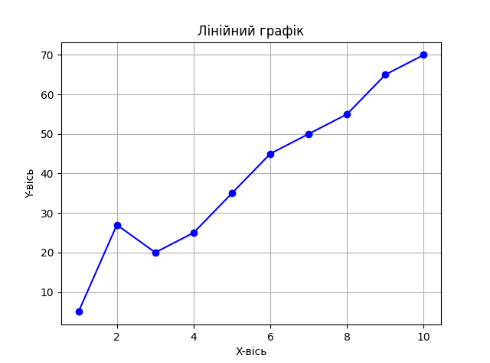
10,70

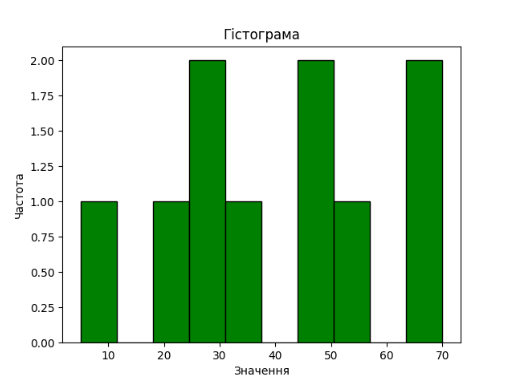
**Результат виконання:**

****

**Діаграми:**

****

****

****

**Висновок:** на цій лабораторній роботі я створив багатофункціональний додаток для візуалізації CSV-наборів даних за допомогою Matplotlib