Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**ЗВІТ**

про виконання лабораторної роботи № 8

з курсу:

«Спеціалізовані мови програмування»

**Виконала:**

студент гр. ІТ-31

Полапа МАКСИМ

**Прийняв:**

Сергій ЩЕРБАК

Львів 2023

**Тема:** Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python

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

**Хід 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)*

import pandas as pd

import matplotlib.pyplot as plt

import logging

class DataVisualizer:

    def \_\_init\_\_(self, csv\_file):

        self.data = pd.read\_csv(csv\_file, delimiter=';')

    def describe\_data(self):

        return self.data.describe()

    def plot\_line\_chart(self, x, y):

        if x not in self.data.columns or y not in self.data.columns:

            print(f"Стовпці '{x}' або '{y}' не знайдені у датафреймі.")

            logging.info("ERROR Стовпці '{x}' або '{y}' не знайдені у датафреймі.")

            return

        plt.plot(self.data[x], self.data[y])

        plt.xlabel(x)

        plt.ylabel(y)

        plt.title(f'{y} over {x}')

        plt.grid(True)

    def plot\_bar\_chart(self, x, y):

        if x not in self.data.columns or y not in self.data.columns:

            print(f"Стовпці '{x}' або '{y}' не знайдені у датафреймі.")

            return

        plt.bar(self.data[x], self.data[y])

        plt.xlabel(x)

        plt.ylabel(y)

        plt.title(f'{y} by {x}')

    def multiple\_subplots(self, x, y):

        if x not in self.data.columns or y not in self.data.columns:

            print(f"Стовпці '{x}' або '{y}' не знайдені у датафреймі.")

            return

        fig, ax = plt.subplots(1, 2, figsize=(12, 6))

        ax[0].plot(self.data[x], self.data[y])

        ax[0].set\_title(f'Line Chart: {y} over {x}')

        ax[1].bar(self.data[x], self.data[y], color='orange')

        ax[1].set\_title(f'Bar Chart: {y} by {x}')

        plt.tight\_layout()

        return fig

visualizer = DataVisualizer('D:/My\_Project/Python/Labu/Laba8/table.csv')

print(visualizer.describe\_data())

def main():

    plt.ion()

    visualizer = DataVisualizer('D:/My\_Project/Python/Labu/Laba8/table.csv')

    print(visualizer.describe\_data())

    def plot\_line\_chart():

        plt.figure()

        visualizer.plot\_line\_chart('Score', 'Year')

        plt.show()

    def plot\_bar\_chart():

        plt.figure()

        visualizer.plot\_bar\_chart('Score', 'Year')

        plt.show()

    def save\_multiple\_subplots():

        fig = visualizer.multiple\_subplots('Score', 'Year')

        fig.savefig('my\_visualization.png')

    # Тепер викликаємо функції візуалізації

    plot\_line\_chart()

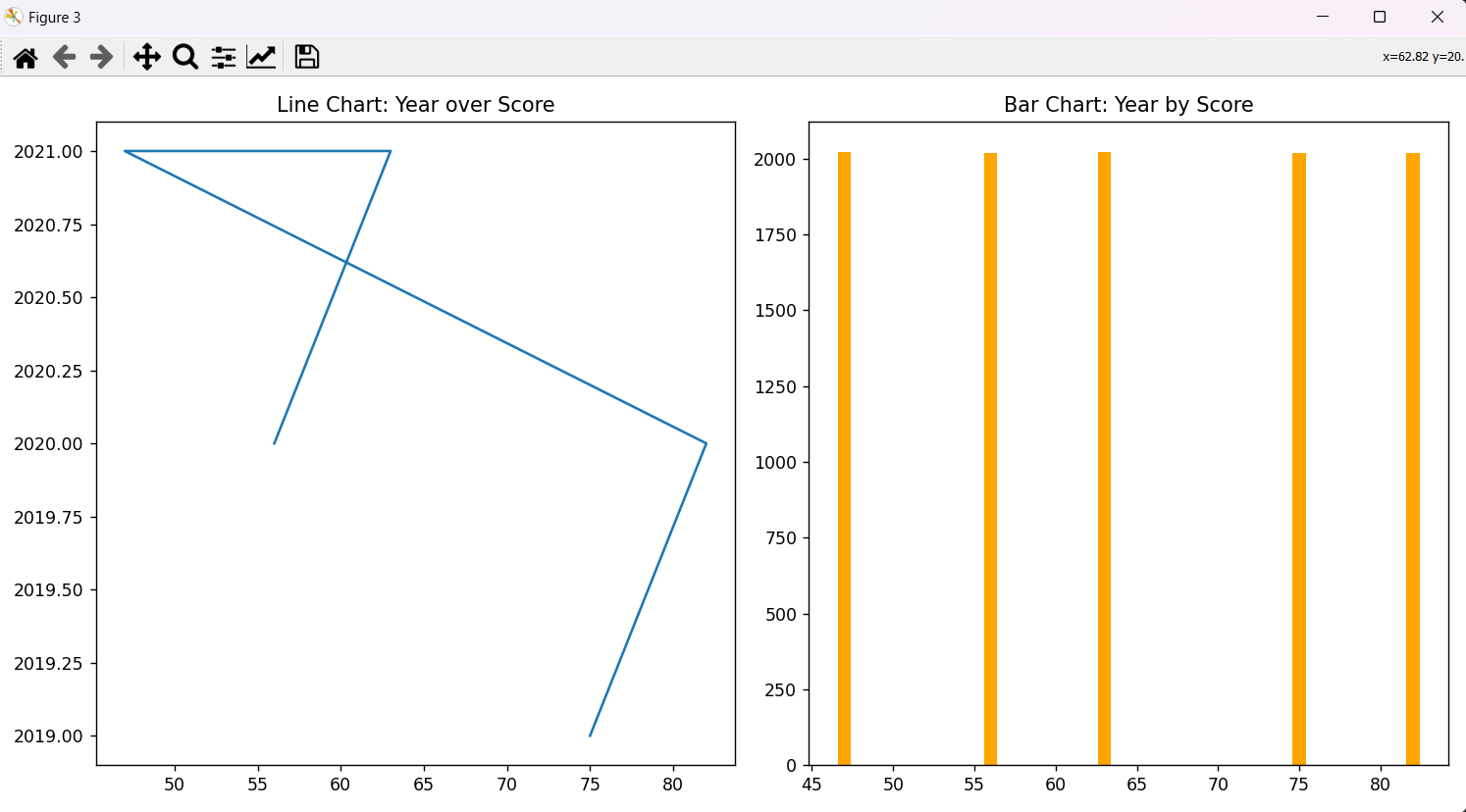
    plot\_bar\_chart()

    save\_multiple\_subplots()

    plt.ioff()

if \_\_name\_\_ == '\_\_main\_\_':

    main()



*Рис.1 – Результат виконання програми*

**Висновок:** на даній лабораторній роботі я створив проект, який надав мені цінний досвід роботи з API, дизайну користувацького інтерфейсу, валідації введення, обробки помилок та тестування.