МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №8

з дисципліни

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

на тему

ВІЗУАЛІЗАЦІЯ ТА ОБРОБКА ДАНИХ ЗА ДОПОМОГОЮ СПЕЦІАЛІЗОВАНИХ БІБЛІОТЕК В PYTHON

Виконала:

ст. гр. ІТ-21сп

Кобленц Д.А.

Прийняв:

Щербак С.С.

Львів-2023

**Мета роботи:** 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**

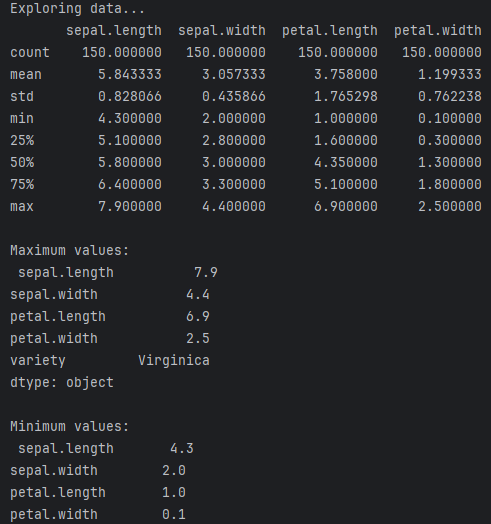
Навчіться створювати кілька піддіаграм в межах одного малюнка для відображення декількох візуалізацій поруч для кращого порівняння.

**Завдання 9: Експорт і обмін**

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

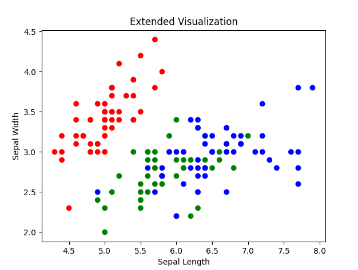
**Код програми:**  
class DataProcessor:  
 """  
 The DataProcessor class processes and visualizes data.  
 """  
  
 def \_\_init\_\_(self):  
 """  
 Initializes a DataProcessor object with a DataFrame of data and an output directory.  
 """  
 relative\_file\_path = os.path.join('..', 'Data', 'Lab8', 'Input', 'iris.csv')  
  
 self.df = pd.read\_csv(relative\_file\_path)  
  
 self.output\_directory = os.path.join('..', 'Data', 'Lab8', 'Output')  
  
 def explore\_data(self):  
 """  
 Explores the data by printing the descriptive statistics, maximum values, and minimum values of the DataFrame.  
 """  
 print(self.df.describe())  
  
 print("\nMaximum values:\n", self.df.max())  
  
 print("\nMinimum values:\n", self.df.min())  
  
 def visualize\_data(self, save=False):  
 """  
 Visualizes the data with various plots and saves the plots if specified.  
 """  
 plt.scatter(self.df['sepal.length'], self.df['sepal.width'])  
 plt.title('Basic Visualization')  
 plt.xlabel('Sepal Length')  
 plt.ylabel('Sepal Width')  
  
 if save:  
 plt.savefig(os.path.join(self.output\_directory, 'basic\_visualization.svg'), format='svg')  
 plt.savefig(os.path.join(self.output\_directory, 'basic\_visualization.png'), format='png')  
 else:  
 plt.show()  
  
 colors = {'Setosa': 'r', 'Versicolor': 'g', 'Virginica': 'b'}  
 plt.scatter(self.df['sepal.length'], self.df['sepal.width'], c=self.df['variety'].apply(lambda x: colors[x]))  
 plt.title('Extended Visualization')  
 plt.xlabel('Sepal Length')  
 plt.ylabel('Sepal Width')  
  
 if save:  
 plt.savefig(os.path.join(self.output\_directory, 'extended\_visualization.svg'), format='svg')  
 plt.savefig(os.path.join(self.output\_directory, 'extended\_visualization.png'), format='png')  
 else:  
 plt.show()  
  
 fig, axs = plt.subplots(2)  
 fig.suptitle('Vertically stacked subplots')  
 axs[0].scatter(self.df['sepal.length'], self.df['sepal.width'], c=self.df['variety'].apply(lambda x: colors[x]))  
 axs[0].set(xlabel='Sepal Length', ylabel='Sepal Width')  
 axs[1].scatter(self.df['petal.length'], self.df['petal.width'], c=self.df['variety'].apply(lambda x: colors[x]))  
 axs[1].set(xlabel='Petal Length', ylabel='Petal Width')  
  
 if save:  
 plt.savefig(os.path.join(self.output\_directory, 'multiple\_subplots.svg'), format='svg')  
 plt.savefig(os.path.join(self.output\_directory, 'multiple\_subplots.png'), format='png')  
 else:  
 plt.show()  
  
 plt.figure(figsize=(10, 6))  
 sns.violinplot(x='variety', y='sepal.length', data=self.df, hue='variety', palette='Set1', legend=False)  
 plt.title('Sepal Length Distribution by Variety')  
 plt.xlabel('Flower Variety')  
 plt.ylabel('Sepal Length')  
  
 if save:  
 plt.savefig(os.path.join(self.output\_directory, 'violin\_plot.svg'), format='svg')  
 plt.savefig(os.path.join(self.output\_directory, 'violin\_plot.png'), format='png')  
 else:  
 plt.show()  
  
 plt.figure(figsize=(10, 6))  
 sns.barplot(x='variety', y='sepal.length', data=self.df)  
 plt.title('Bar Plot of Sepal Length by Variety')  
 plt.xlabel('Flower Variety')  
 plt.ylabel('Sepal Length')  
  
 if save:  
 plt.savefig(os.path.join(self.output\_directory, 'bar\_plot.svg'), format='svg')  
 plt.savefig(os.path.join(self.output\_directory, 'bar\_plot.png'), format='png')  
 else:  
 plt.show()

На рисунку 1 зображено результат програми(мінімальні та максимальні значення).



**Рис. 1** Результат виконання програми(значення)

На рисунку 2 зображено результат програми(візуалізація).



**Рис. 2** Результат виконання програми(візуалізація)

**Висновок:** я створив багатофункціональний додаток для візуалізації CSV-наборів даних за допомогою Matplotlib.