

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 Python

Виконав:
студент групи РІ-21сп
Владислав ДМИТРЕНКО

Львів – 2024

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

План роботи

Завдання 1: Оберіть CSV-набір даних, який ви хочете візуалізувати. Переконайтеся, що він містить відповідні дані для створення змістовних візуалізацій.

Завдання 2: Напишіть код для завантаження даних з CSV-файлу в ваш додаток Python. Використовуйте бібліотеки, такі як Pandas, для спрощення обробки даних.

Завдання 3: Визначте екстремальні значення по стовцям

Завдання 4: 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: Попередньо обробіть набір даних за необхідністю для візуалізації. Це може включати виправлення даних, фільтрацію, агрегацію або трансформацію.

Завдання 6: 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: Реалізуйте більш складні візуалізації, виходячи з характеристик набору. Поекспериментуйте з різними функціями Matplotlib та налаштуваннями.

Завдання 8: Навчіться створювати кілька піддіаграм в межах одного малюнка для відображення декількох візуалізацій поруч для кращого порівняння.

Завдання 9: Реалізуйте функціональність для експорту візуалізацій як зображень (наприклад, PNG, SVG) або інтерактивних веб-додатків (наприклад, HTML)

Результати тестування:

```
PS C:\Users\Blxxd\Documents\GitHub\DegraCalc> python main.py
1. Display extreme values
2. Show Temperature Line Chart
3. Show Temperature vs Humidity Scatter Plot
4. Show Humidity Histogram
5. Show Multiple Subplots
6. Export Chart as PNG
Select an option (or 'q' to quit):
```

Рис. 1. Користувацький інтерфейс

```
Select an option (or 'q' to quit): 1
Temperature: Min=-8.751103306122198, Max=34.334323546035684
Humidity: Min=20.10315523092394, Max=99.34085119341344
WindSpeed: Min=0.0205347525669341, Max=19.902143508601725
Rainfall: Min=0.0, Max=2.5
1. Display extreme values
2. Show Temperature Line Chart
3. Show Temperature vs Humidity Scatter Plot
4. Show Humidity Histogram
5. Show Multiple Subplots
6. Export Chart as PNG
Select an option (or 'q' to quit):
```

Рис. 2. Вивід даних

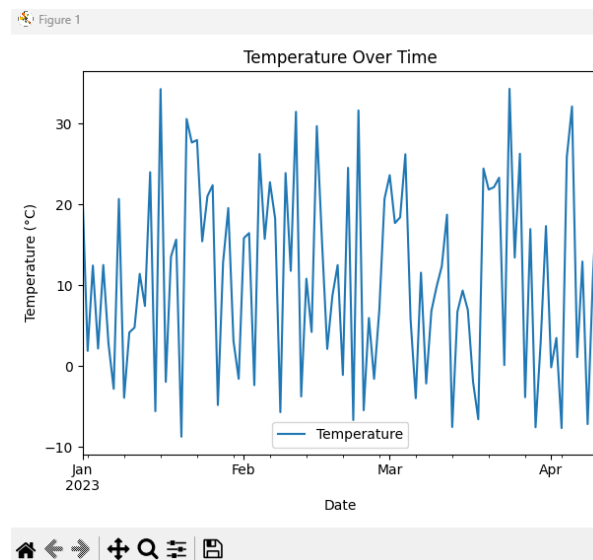


Рис 3. Вигляд лінійної діаграми

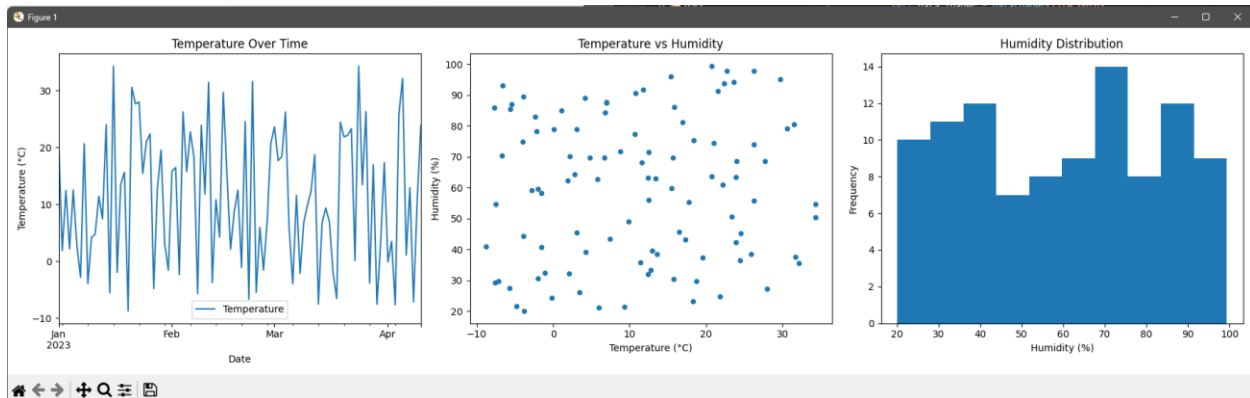


Рис. 4. Вигляд всіх діаграм разом

Текст контроллера даних

```
class DataController:
    def __init__(self, file_path):
        self.data_loader = DataLoader(file_path)
        self.data = self.data_loader.load_data()
        if self.data.empty():
            print("Warning: The data is empty or could not be loaded correctly.")
        self.analysis = DataAnalysis(self.data)

    def display_extremes(self):
        extremes = self.analysis.get_extreme_values()
        for key, value in extremes.items():
            print(f'{key}: Min={value[0]}, Max={value[1]}')

    def get_plot_instance(self, plot_type):
        if plot_type == "line":
            return LineChart()
        elif plot_type == "scatter":
            return ScatterPlot()
        elif plot_type == "histogram":
            return Histogram()
        else:
            return None

    def plot_chart(self, plot_type):
        plot_instance = self.get_plot_instance(plot_type)
        if plot_instance:
            plot_instance.plot(self.data)
            plot_instance.show()
```

Текст функції експорту діаграми в PNG

```
class ExportController:
    def export_plot(self, plot_instance, filename, format="png"):
        plot_instance.save(filename, format)
```

```
def export_as_html(self, plot_instance, filename):  
    plot_instance.save_as_html(filename)  
  
def prompt_and_export(self, plot_instance, filename_base, format="png"):  
    save_choice = input("Do you want to save this chart? (y/n): ")  
    if save_choice.lower() == 'y':  
        filename = f"{filename_base}.{format}"  
        self.export_plot(plot_instance, filename, format)  
        print(f"Chart saved as {filename}.")
```

Висновки: В ході виконання лабораторної роботи було створено багатофункціональний додаток для візуалізації CSV-наборів даних за допомогою Matplotlib.