Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра інформаційних систем та мереж

ЗВІТ

про виконання лабораторної роботи № 8

**“ Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python ”**

з дисципліни **“** **Спеціалізовані мови програмування”**

Виконав:

студент групи ІТ-32

ОЛЕНЮК О. М.

Прийняв:

ЩЕРБАК С. С.

**Львів – 2023**

**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 піддіагам

Навчіться створювати кілька піддіаграм в межах одного малюнка для відображення декількох візуалізіцій 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:**

consumer.py:

class Consumer:  
  
 def \_\_init\_\_(self, data):  
 self.first\_name = data[0]  
 self.last\_name = data[1]  
 self.company\_name = data[2]  
 self.address = data[3]  
 self.city = data[4]  
 self.country = data[5]  
 self.state = data[6]  
 self.zip = data[7]  
 self.phone1 = data[8]  
 self.phone2 = data[9]  
 self.email = data[10]

runner.py:

from service import ConsumerServiceImpl  
  
  
def main():  
 data\_file = "./files/consumers.csv"  
 service = ConsumerServiceImpl(data\_file)  
  
 while True:  
 print(f"1. Display states histogram\n"  
 f"2. Display states pie chart\n"  
 f"3. Display combined states diagram\n"  
 f"0. Exit\n")  
 option = input("Enter your choice: ")  
  
 if option == "1":  
 has\_to\_be\_downloaded = True if input(  
 "Do you want to download the histogram? Enter 'y' or anything else not to download: ") == "y" else False  
 service.create\_states\_histogram(has\_to\_be\_downloaded)  
 elif option == "2":  
 has\_to\_be\_downloaded = True if input(  
 "Do you want to download the pie chart? Enter 'y' or anything else not to download: ") == "y" else False  
 max\_quantity = int(input("Enter the maximum quantity of states: "))  
 service.create\_states\_pie\_chart(has\_to\_be\_downloaded, max\_quantity=max\_quantity)  
 elif option == "3":  
 has\_to\_be\_downloaded = True if input(  
 "Do you want to download the bar chart? Enter 'y' or anything else not to download: ") == "y" else False  
 max\_quantity = int(input("Enter the maximum quantity of states in the pie chart diagram included: "))  
 service.create\_combined\_diagram(has\_to\_be\_downloaded, max\_quantity=max\_quantity)  
 elif option == "0":  
 exit(0)  
 else:  
 print("Incorrect input. Enter again.")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

service.py

import abc  
from src.utility.FileProcessor import CsvProcessor as csv\_processor  
from consumer import Consumer  
from matplotlib import pyplot as plt  
from collections import Counter  
  
  
@abc.abstractmethod  
class ConsumerService(abc.ABC):  
  
 def \_\_init\_\_(self):  
 self.\_consumers = []  
  
  
class ConsumerServiceImpl(ConsumerService):  
 def \_\_init\_\_(self, file\_path: str):  
 super().\_\_init\_\_()  
 consumers\_dataframe = csv\_processor.read(file\_path)  
 for data in consumers\_dataframe.values:  
 self.\_consumers.append(Consumer(data))  
  
 def get\_states(self):  
 return [consumer.state for consumer in self.\_consumers]  
  
 def create\_states\_histogram(self, has\_to\_be\_downloaded: bool):  
 states = self.get\_states()  
 state\_counter = Counter(states)  
  
 plt.figure(figsize=(8, 7))  
 plt.bar([key for key in state\_counter], [state\_counter[key] for key in state\_counter],  
 color='green')  
  
 if has\_to\_be\_downloaded:  
 plt.savefig('./files/state-bar-chart.png')  
 plt.title('Bar Chart')  
 plt.xlabel('States')  
 plt.ylabel('Frequency')  
 plt.xticks(fontsize=6)  
 plt.yticks(fontsize=10)  
 plt.xticks(rotation=-90)  
  
 plt.show()  
  
 def create\_states\_pie\_chart(self, has\_to\_be\_downloaded: bool, max\_quantity: int = 10):  
 states = self.get\_states()  
 state\_counter = Counter(states)  
 sorted\_dict\_values = dict(sorted(state\_counter.items(), key=lambda item: item[1], reverse=True)[:max\_quantity])  
  
 labels = list(sorted\_dict\_values.keys())  
 sizes = list(sorted\_dict\_values.values())  
  
 plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)  
  
 if has\_to\_be\_downloaded:  
 plt.savefig('./files/state-pie-chart.png')  
  
 plt.title('Pie Chart')  
 plt.show()  
  
 def create\_combined\_diagram(self, has\_to\_be\_downloaded: bool, max\_quantity: int = 10):  
 states = self.get\_states()  
 state\_counter = Counter(states)  
 sorted\_dict\_values = dict(sorted(state\_counter.items(), key=lambda item: item[1], reverse=True)[:max\_quantity])  
  
 # Create a figure with two subplots: one for the bar chart and one for the pie chart  
 fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 7))  
  
 # Bar Chart  
 ax1.bar([key for key in state\_counter], [state\_counter[key] for key in state\_counter], color='green')  
 ax1.set\_title('Bar Chart')  
 ax1.set\_xlabel('States')  
 ax1.set\_ylabel('Frequency')  
 ax1.tick\_params(axis='x', rotation=45) # Adjust rotation if needed  
  
 # Pie Chart  
 labels = list(sorted\_dict\_values.keys())  
 sizes = list(sorted\_dict\_values.values())  
 ax2.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)  
 ax2.set\_title('Pie Chart')  
  
 # Adjust layout  
 plt.tight\_layout()  
  
 # Save or show the plot  
 if has\_to\_be\_downloaded:  
 plt.savefig('./files/combined-chart.png')  
 plt.show()

**Результат програми:**

На рис.1, 2, 3 зображений результат виконання роботи у програмі.

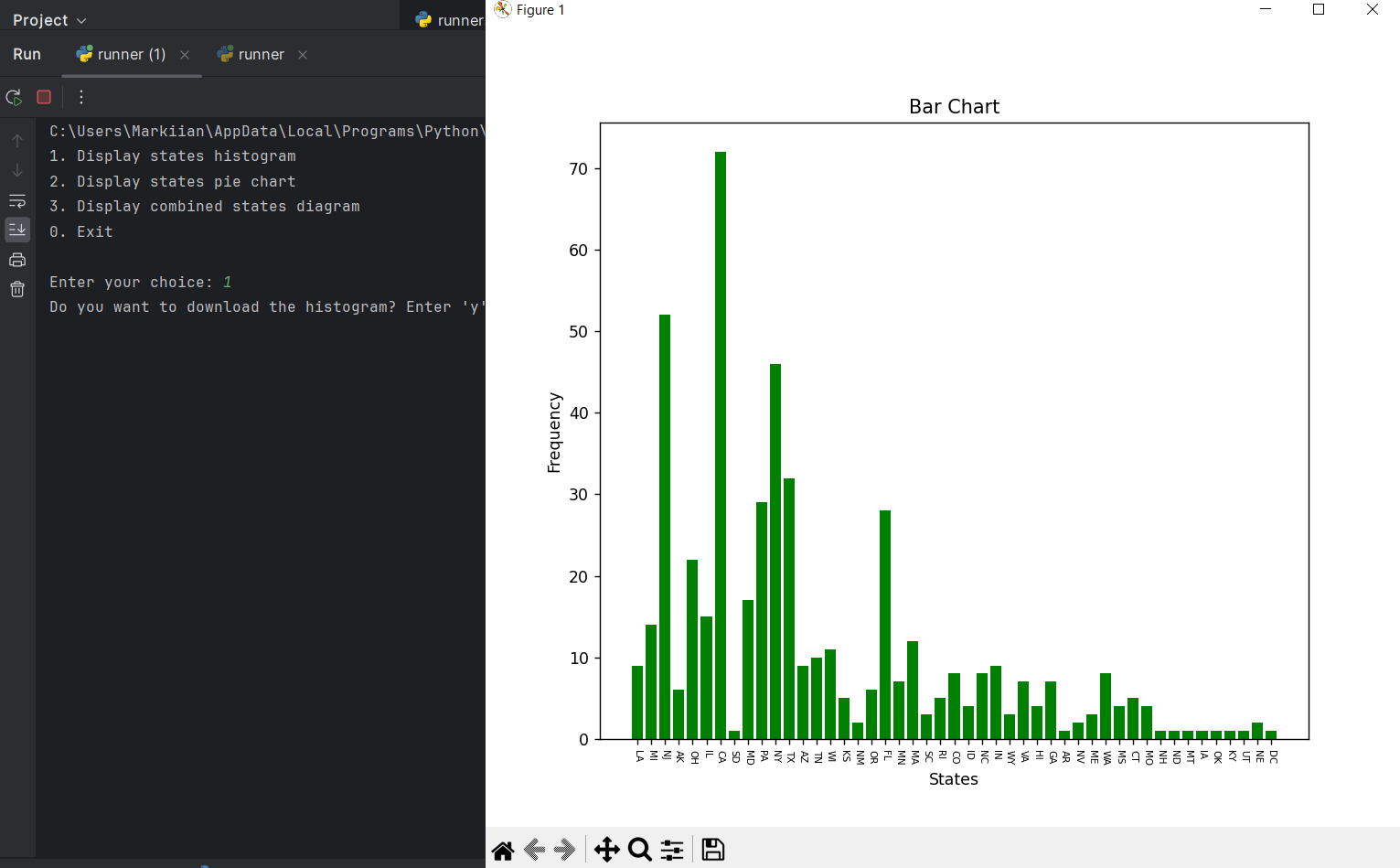


Рис.1

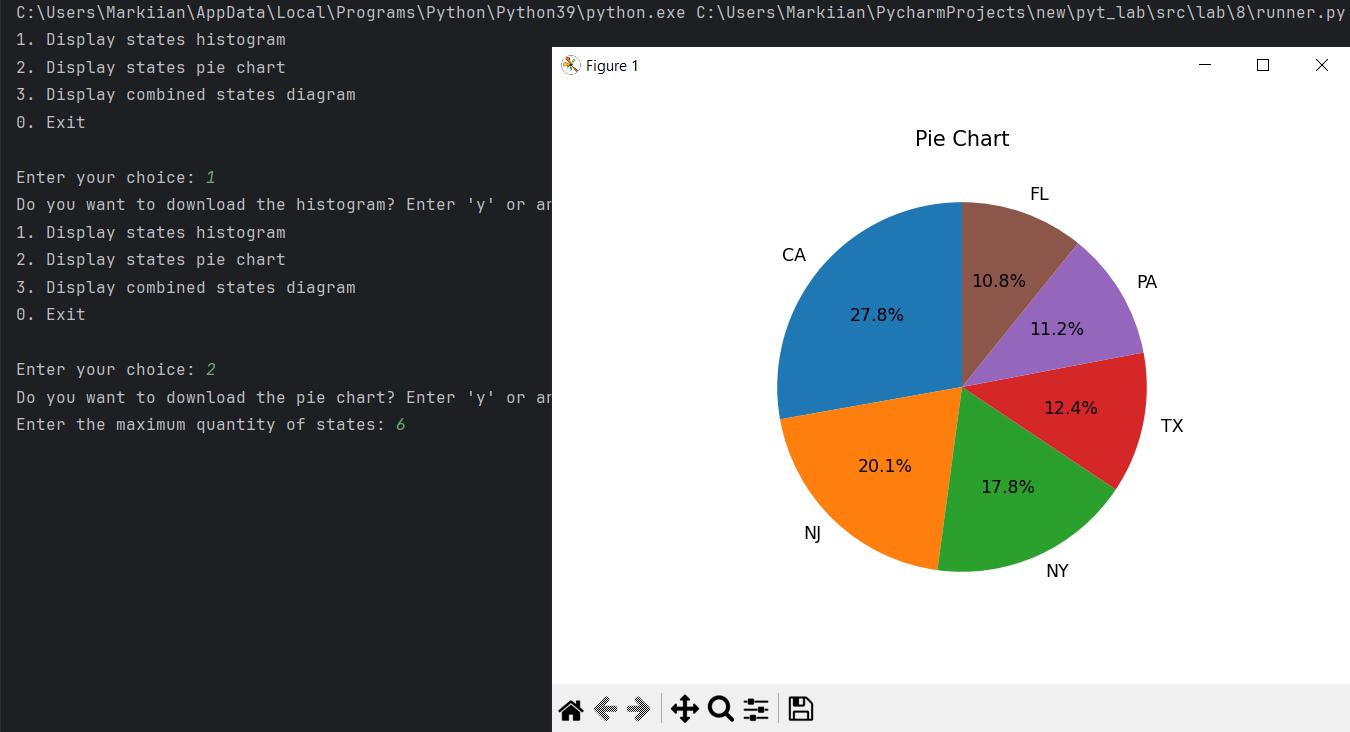


Рис.2

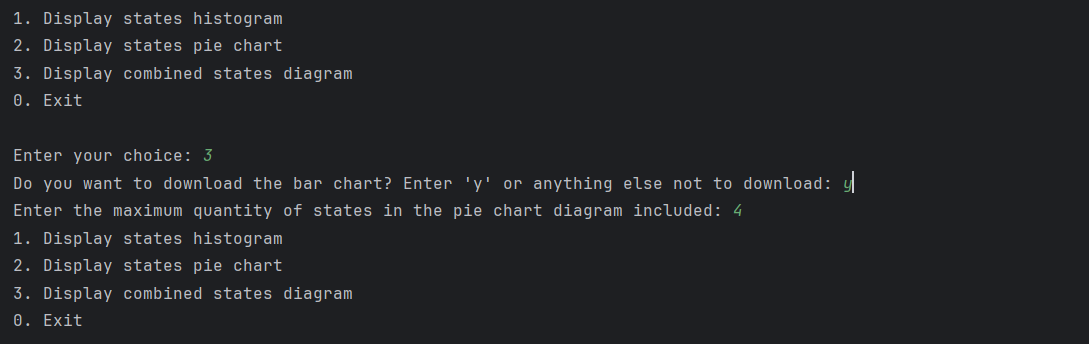


Рис.3

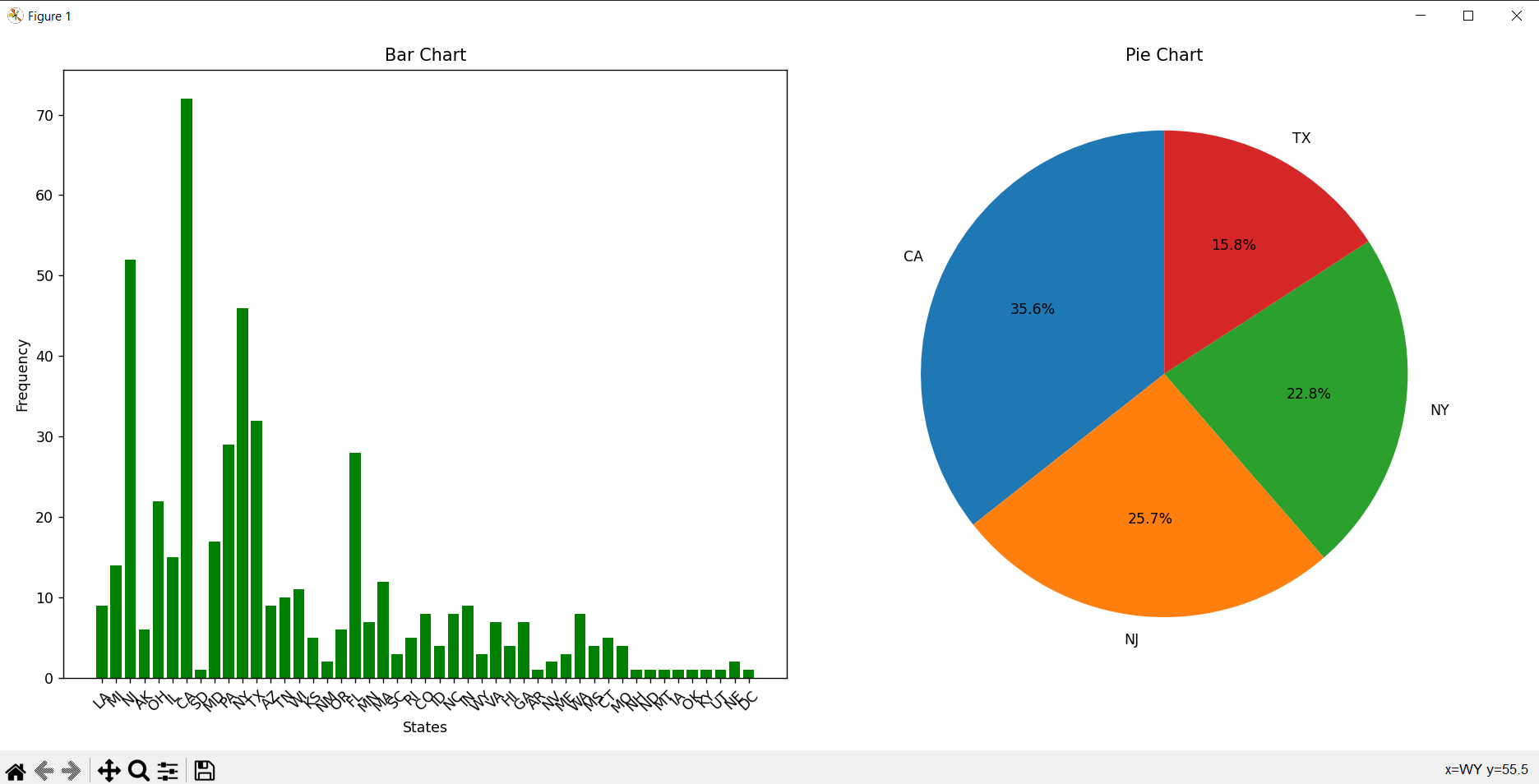


Рис.4

**Посилання на репозиторій**: <https://github.com/Oleksandr2004Oleniuk/SMP.git>

**Висновок**. у цій лабораторній роботі я створиd багатофункціональний додаток для візуалізації CSV-наборів даних за допомогою Matplotlib. Цей проект покращив мої навички візуалізації даних, дозволяючи мені досліджувати результати з різноманітними наборами даних.