Міністерство освіти і науки України

Національний університет «Львівська політехніка»

******

Звіт про виконання лабораторної роботи №8  
з дисципліни  
“Спеціалізовані мови програмування”  
на тему  
“Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python ”

Виконала:  
студентка групи РІ-31,  
Пятницька Вікторія

Прийняв:

Щербак С.С.

**Мета.** 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)

**Код програми:**

**./lab8/data\_handler.py:**

import pandas as pd

class DataHandler:

def \_\_init\_\_(self, file\_path):

self.\_data = pd.read\_csv(file\_path)

def get\_data(self):

return self.\_data

def show\_summary(self):

print(self.\_data.describe())

def find\_extremes(self):

extremes = {}

for column in self.\_data.select\_dtypes(include='number').columns:

extremes[column] = {

'min': self.\_data[column].min(),

'max': self.\_data[column].max()

}

for col, values in extremes.items():

print(f"Column: {col} | Min: {values['min']} | Max: {values['max']}")

return extremes

**./lab8/data\_visualizer.py:**

from matplotlib import pyplot as plt

from data\_handler import DataHandler

from visualization import Visualization

class DataVisualizer:

def \_\_init\_\_(self, data\_handler: DataHandler, strategy: Visualization):

self.data\_handler = data\_handler

self.strategy = strategy

def set\_strategy(self, strategy: Visualization):

self.strategy = strategy

def visualize(self):

self.strategy.visualize(self.data\_handler.get\_data())

**./lab8/data\_visualization.py:**

from abc import ABC, abstractmethod

import matplotlib.pyplot as plt

class Visualization(ABC):

@abstractmethod

def visualize(self, data):

pass

class BarChart(Visualization):

def \_\_init\_\_(self, category\_column, value\_column):

self.category\_column = category\_column

self.value\_column = value\_column

def visualize(self, data):

data\_grouped = data.groupby(self.category\_column)[self.value\_column].mean()

data\_grouped.plot(kind='bar', figsize=(10, 5), color='green')

plt.xlabel(self.category\_column)

plt.ylabel(f"Average {self.value\_column}")

plt.title(f"Average {self.value\_column} by {self.category\_column}")

plt.show()

class Histogram(Visualization):

def \_\_init\_\_(self, column, bins=10):

self.column = column

self.bins = bins

def visualize(self, data):

plt.hist(data[self.column], bins=self.bins, color='skyblue')

plt.title(f'Histogram of {self.column}')

plt.xlabel(self.column)

plt.ylabel('Frequency')

plt.show()

class PieChart(Visualization):

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

self.column = column

def visualize(self, data):

plt.figure(figsize=(8, 8))

data[self.column].value\_counts().plot(kind='pie', autopct='%1.1f%%', colors=['lightblue', 'orange', 'green', 'pink'])

plt.title(f"Distribution of {self.column}")

plt.ylabel('')

plt.show()

class MultiPlot(Visualization):

def visualize(self, data):

fig, axs = plt.subplots(1, 2, figsize=(14, 7))

# Bar chart for 'Number\_of\_Apps\_Used'

data['Number\_of\_Apps\_Used'].value\_counts().plot(kind='bar', ax=axs[0], color='green')

axs[0].set\_title('Distribution of Number of Apps Used')

axs[0].set\_xlabel('Number of Apps')

axs[0].set\_ylabel('Frequency')

# Pie chart for 'Number\_of\_Apps\_Used'

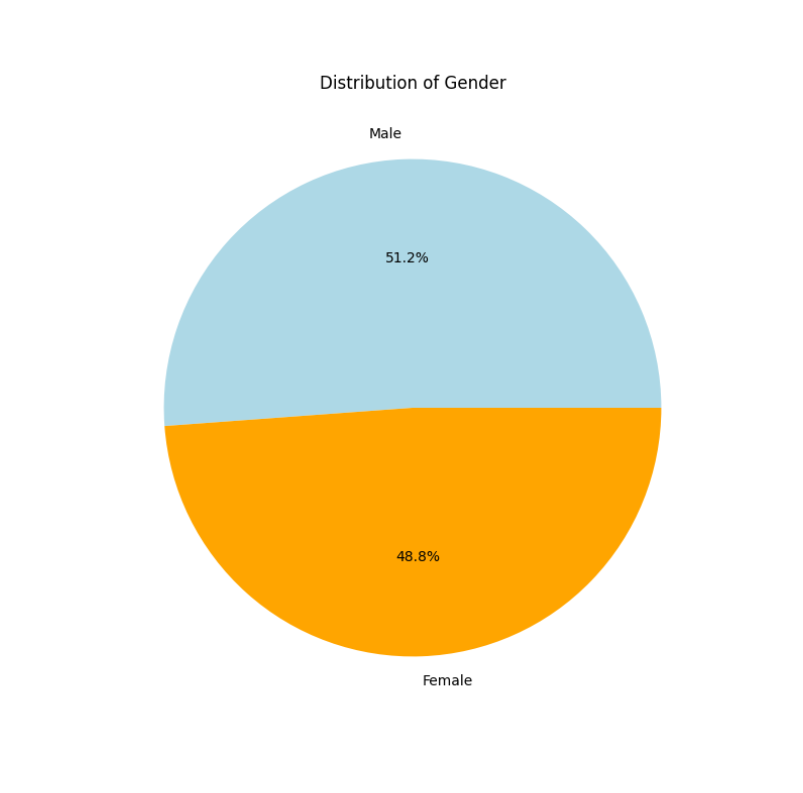
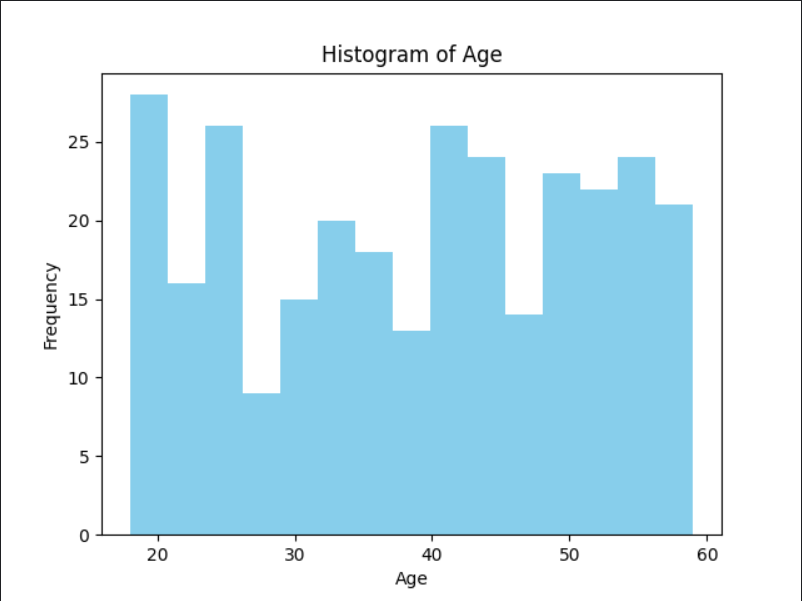
data['Number\_of\_Apps\_Used'].value\_counts().plot(kind='pie', ax=axs[1], autopct='%1.1f%%', colors=['lightblue', 'orange', 'yellow', 'pink'])

axs[1].set\_title('Number of Apps Used Distribution')

axs[1].set\_ylabel('')

plt.tight\_layout()

plt.show()



*Рис.1.Приклад виконання програми*

**Висновки:** внаслідок виконання лабораторної роботи я розробила додаток для візуалізації CSV-наборів даних за допомогою Matplotlib.

**GitHub:** <https://github.com/viksi01/Calculator/tree/lab8>