МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

**Звіт**

До лабораторної роботи №8

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

Виконав:

ст. гр. ІТ-21сп

**Артур Крохін**

Прийняв:

доц. каф. ІСМ

**Сергій ЩЕРБАК**

Львів ‒ 2023

**Лабораторна робота №8**

**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.

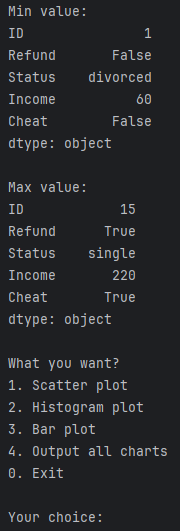
Завдання 9: Експорт і обмін

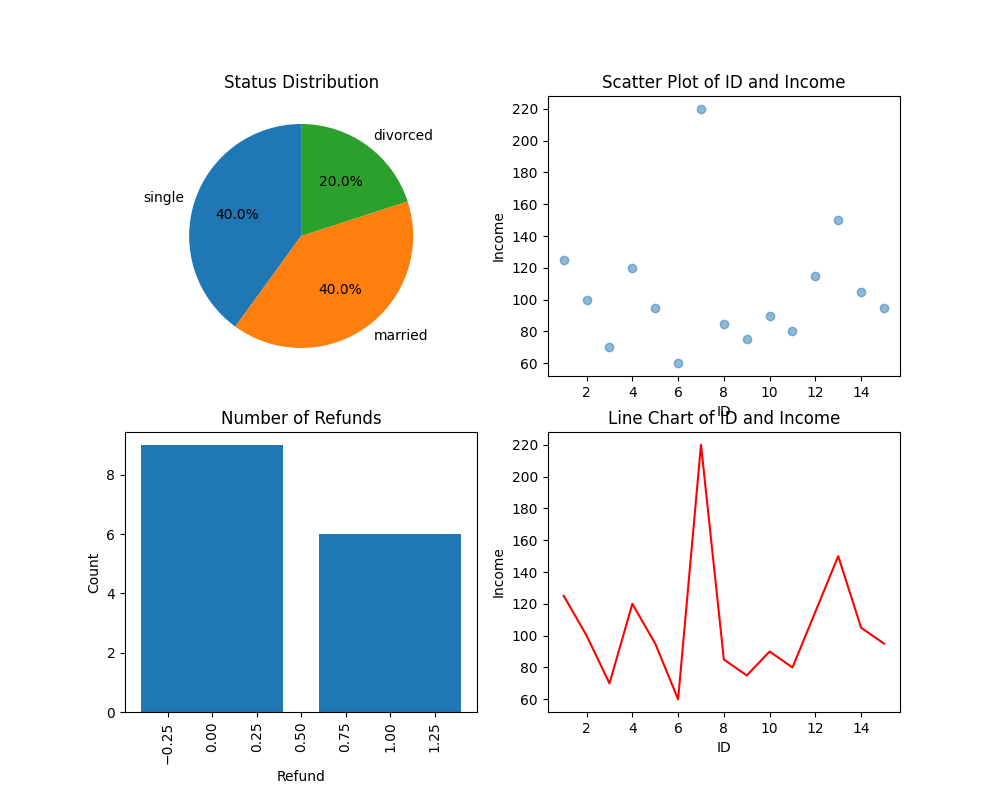
Реалізуйте функціональність для експорту візуалізацій як зображень (наприклад, PNG, SVG) або інтерактивних веб-додатків (наприклад, HTML)

Файл **data\_visualizer.py**

import os  
  
import pandas as pd  
import matplotlib.pyplot as plt  
import plotly.express as px  
from Shared.Save.file\_save import FileSaver  
  
  
class DataVisualizer:  
  
 def \_\_init\_\_(self):  
 self.data = pd.read\_csv("Classes/labwork\_8/tax.csv")  
 self.columns = self.data.columns.tolist()  
 self.file\_save = FileSaver("Data/labwork\_8")  
  
 def explore\_data(self):  
  
 min\_values = self.data.min()  
 max\_values = self.data.max()  
  
 print("Min value:")  
 print(min\_values)  
  
 print("\nMax value:")  
 print(max\_values)  
  
 def save\_plots(self, data, name\_file):  
  
 path = "Data/labwork\_8"  
 full\_path = os.path.join(path, name\_file)  
 data.savefig(full\_path)  
 print(f"Plot saved at: {full\_path}")  
  
 def visualize\_data(self, user\_choice):  
 """  
 Visualize data based on user input.  
  
 :param user\_choice: User's choice for the type of visualization.  
 """  
 if user\_choice == "1":  
 self.scatter\_plot()  
 elif user\_choice == "2":  
 self.histogram\_plot()  
 elif user\_choice == "3":  
 self.bar\_plot()  
 elif user\_choice == "4":  
 self.multiple\_subplots()  
 else:  
 print("Invalid choice. Please enter a number from 1 to 4.")  
  
 def scatter\_plot(self):  
  
 scatter\_fig, scatter\_ax = plt.subplots()  
 scatter\_ax.scatter(self.data['ID'], self.data['Income'], alpha=0.5)  
 scatter\_ax.set\_xlabel('ID')  
 scatter\_ax.set\_ylabel('Income')  
 scatter\_ax.set\_title('Scatter Plot of Id and Last Income')  
 self.save\_plots(scatter\_fig, 'scatter\_plot')  
 plt.show()  
  
 def histogram\_plot(self):  
  
 histogram\_fig, histogram\_ax = plt.subplots()  
 self.data['Income'].plot(kind='hist', bins=10, color='green', alpha=0.7, ax=histogram\_ax)  
 histogram\_ax.set\_title('Histogram')  
 histogram\_ax.set\_xlabel('Value')  
 histogram\_ax.set\_ylabel('Number of people')  
 self.save\_plots(histogram\_fig, 'histogram\_plot')  
 plt.show()  
  
 def bar\_plot(self):  
  
 bar\_fig, bar\_ax = plt.subplots(figsize=(8, 7))  
 self.data['Status'].value\_counts().plot(kind='bar', color='skyblue', ax=bar\_ax)  
 bar\_ax.set\_title('Number of employees by status')  
 bar\_ax.set\_xlabel('Status')  
 bar\_ax.set\_ylabel('Number of employees')  
 self.save\_plots(bar\_fig, 'bar\_plot')  
 plt.show()  
  
 def multiple\_subplots(self):  
  
 status\_distribution = self.data['Status'].value\_counts()  
 axes[0, 0].pie(status\_distribution, autopct='%1.1f%%', labels=status\_distribution.index, startangle=90)  
 axes[0, 0].set\_title('Status Distribution')  
  
 # plot the scatter plot of ID and Income in the first subplot  
 axes[0, 1].scatter(self.data['ID'], self.data['Income'], alpha=0.5)  
 axes[0, 1].set\_xlabel('ID')  
 axes[0, 1].set\_ylabel('Income')  
 axes[0, 1].set\_title('Scatter Plot of ID and Income')  
  
 # plot the bar chart of the refund counts in the second subplot  
 refund\_counts = self.data['Refund'].value\_counts()  
 axes[1, 0].bar(refund\_counts.index, refund\_counts)  
 axes[1, 0].set\_xlabel('Refund')  
 axes[1, 0].set\_ylabel('Count')  
 axes[1, 0].set\_title('Number of Refunds')  
 axes[1, 0].tick\_params(axis='x', labelrotation=90)  
  
 # plot the line chart of ID and Income in the third subplot  
 axes[1, 1].plot(self.data['ID'], self.data['Income'], color='red')  
 axes[1, 1].set\_xlabel('ID')  
 axes[1, 1].set\_ylabel('Income')  
 axes[1, 1].set\_title('Line Chart of ID and Income')  
  
 # show the plot  
 plt.show()  
  
 # adjust the spacing between the subplots  
 plt.subplots\_adjust(wspace=0.5, hspace=0.5)  
  
 # save the plot as a PNG file and output it as an HTML file  
 self.save\_plots(fig, 'all\_plots.png')  
 fig = px.scatter(self.data, x='ID', y='Income', color='Status',  
 title='Scatter Plot of ID and Income')  
 fig.write\_html('Data/labwork\_8/output\_plot.html')  
 plt.show()

Результат виконання всіх завдань:





**Рис.1** Результат виконання

**Висновок:** Виконавши ці завдання, я розробив додаток для візуалізації CSV-наборів даних за допомогою Matplotlib та базових принципів ООП (наслідування, інкапсуляція, поліморфізм).