Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Звіт про виконання лабораторної роботи №8

з дисципліни «Візуалізація та обробка даних за допомогою

спеціалізованих бібліотек Python»

Виконала:  
студентка групи РІ-31  
Заплетнюк Ю.І.

Прийняв:  
Щербак С.С.

**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 програмної реалізації**

**data\_loader.py**

import pandas as pd

import matplotlib.pyplot as plt

import mpld3

class DataLoader:

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

self.file\_path = file\_path

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

self.preprocess\_data()

def preprocess\_data(self):

# конвертація RGB-значень у кортежі

self.data['RGB'] = self.data['RGB'].apply(

lambda x: tuple(map(int, x.strip('rgb()').split(',')))

)

def get\_extreme\_values(self):

extremes = {

'HEX\_min': self.data['HEX'].min(),

'HEX\_max': self.data['HEX'].max(),

'RGB\_min': self.data['RGB'].min(),

'RGB\_max': self.data['RGB'].max()

}

return extremes

def plot\_colors(self):

colors = self.data['HEX']

names = self.data['Name']

plt.figure(figsize=(8, 6))

plt.barh(names, [1] \* len(names), color=colors)

plt.xlabel('Color')

plt.ylabel('Name')

plt.title('Colors Visualization')

plt.show()

def advanced\_visualizations(self):

"""Візуалізація кольорів у вигляді сітки."""

fig, ax = plt.subplots(figsize=(8, 6))

ax.set\_title("Color Grid")

for i, (index, row) in enumerate(self.data.iterrows()):

ax.add\_patch(plt.Rectangle((i % 5, i // 5), 1, 1, color=row['HEX']))

ax.set\_xlim(0, 5)

ax.set\_ylim(0, len(self.data) // 5 + 1)

ax.axis("off")

plt.show()

def multiple\_subplots(self):

"""Створення кількох підграфіків із HEX і RGB значеннями."""

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))

# HEX значення на гістограмі

ax1.bar(self.data['Name'], range(len(self.data)), color=self.data['HEX'])

ax1.set\_title('HEX Colors')

ax1.set\_xticklabels(self.data['Name'], rotation=45, ha='right')

# Значення RGB як діаграма розсіювання

r, g, b = zip(\*self.data['RGB'])

ax2.scatter(r, g, c=self.data['HEX'], s=100)

ax2.set\_title('RGB Values')

ax2.set\_xlabel('Red')

ax2.set\_ylabel('Green')

plt.tight\_layout()

plt.show()

return fig

def export\_visualization(self, fig, filename, file\_format='png'):

"""Експорт візуалізації у формат PNG, SVG або HTML."""

if file\_format in ['png', 'svg']:

fig.savefig(f"{filename}.{file\_format}", format=file\_format)

elif file\_format == 'html':

html = mpld3.fig\_to\_html(fig)

with open(f"{filename}.html", 'w') as f:

f.write(html)

print(f"Visualization saved as {filename}.{file\_format}")

file\_path = '/Users/uliazapletnuk/Desktop/^\_^/3kyrsik/python/lab8/color\_srgb.csv'

data\_loader = DataLoader(file\_path)

data\_loader1 = DataLoader(file\_path)

extremes = data\_loader.get\_extreme\_values()

print("Екстремальні значення:", extremes)

data\_loader.plot\_colors()

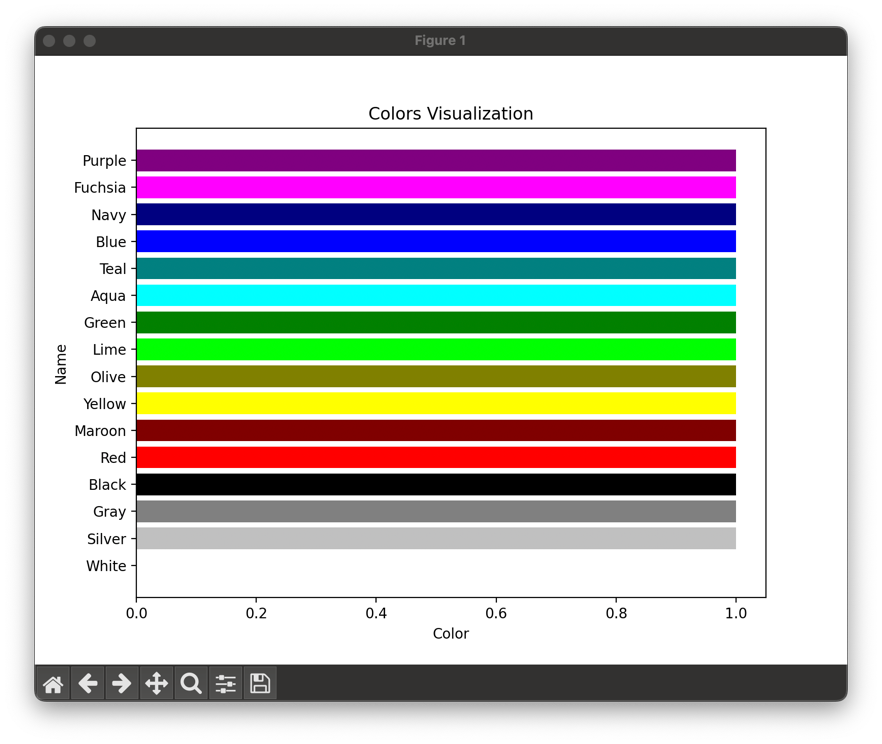
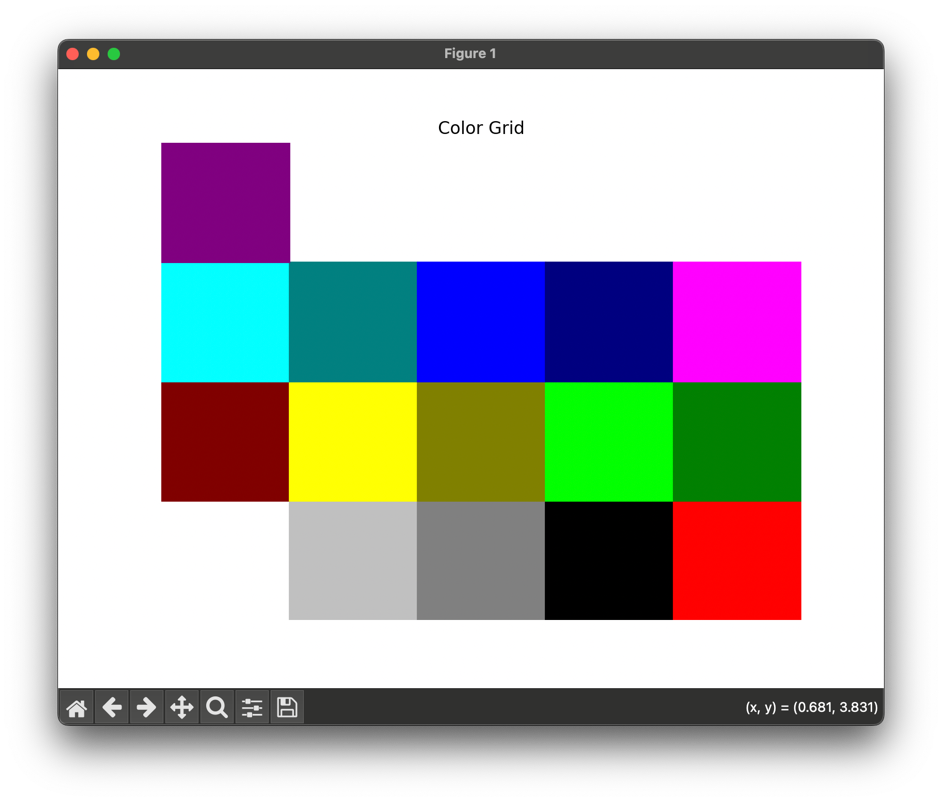
data\_loader.advanced\_visualizations()

fig = data\_loader.multiple\_subplots()

data\_loader.export\_visualization(fig, 'color\_chart', 'png')

data\_loader.export\_visualization(fig, 'color\_chart', 'html')

На рис. 1-3 показано результат виконання завдання.

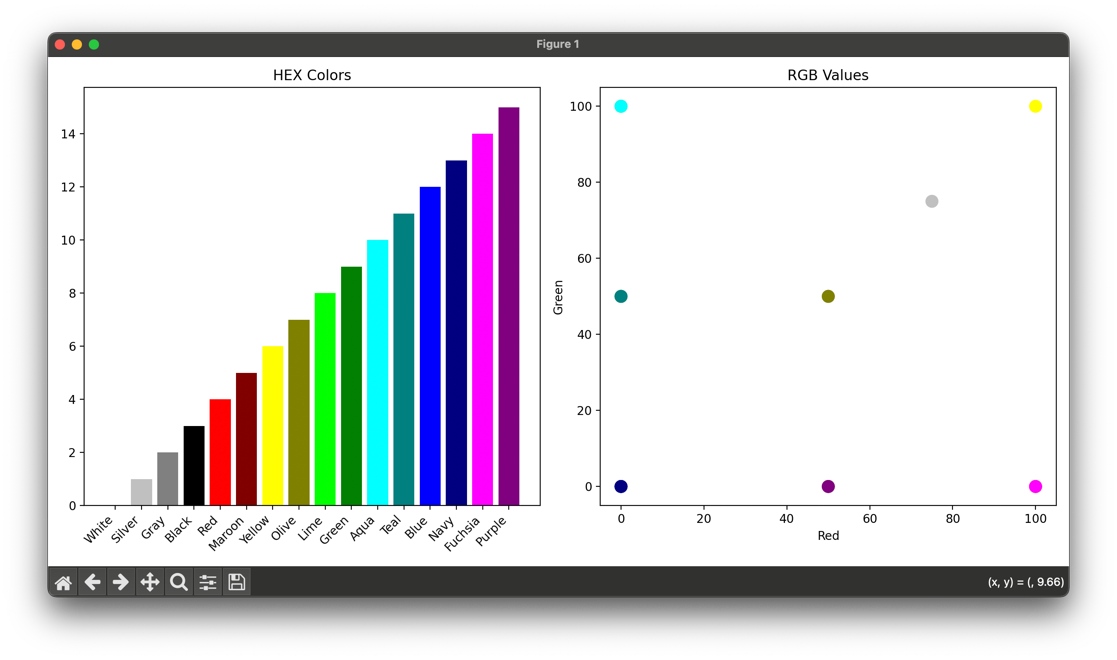
****

Рис. 1-3. Результат виконання завдання

**Висновки:** виконавши лабораторну роботу, було створено багатофункціональний додаток для візуалізації CSV-наборів даних за допомогою Matplotlib. Цей проект покращив мої навички візуалізації даних, дозволяючи досліджувати результати з різноманітними наборами даних