МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



Лабораторна робота №8

з дисципліни “Спеціалізовані мови програмування”

Виконав:

студент групи ІТ-31

Олександр ГАМАЮНОВ

Прийняв:

доцент кафедри ІСМ

Сергій ЩЕРБАК

Львів-2023

**Тема роботи:** Візуалізація та обробка даних за допомогою спеціалізованих бібліотек Python

**Мета роботи:** 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:  
Main.py:

""" Main file"""

import pandas as pd

import matplotlib.pyplot as plt

# Функція для завантаження та попередньої обробки даних

def load\_and\_preprocess(csv\_file):

    """ Load file and preprocces info"""

    data = pd.read\_csv(csv\_file)

    print("Unique values before mapping:", data['radiant\_win'].unique())

    # Correct the mapping keys to boolean values

    data['radiant\_win'] = data['radiant\_win'].map({True: 1, False: 0})

    print("Unique values after mapping:", data['radiant\_win'].unique())

    data.dropna(subset=['avg\_mmr', 'duration'], inplace=True)

    print(data['avg\_mmr'].describe())

    print(data['duration'].describe())

    print(data['avg\_mmr'].isnull().sum())

    print(data['duration'].isnull().sum())

    print("Shape of DataFrame after dropna:", data.shape)

    return data

def plot\_combined\_chart(ax, data):

    """ combination charts"""

    game\_counts = data['radiant\_win'].value\_counts().sort\_index()

    avg\_mmr\_by\_result = data.groupby('radiant\_win')['avg\_mmr'].mean().sort\_index()

    print("Game Counts:\n", game\_counts)

    print("Average MMR by Result:\n", avg\_mmr\_by\_result)

    ax2 = ax.twinx()

    if not game\_counts.empty:

        ax.bar(game\_counts.index, game\_counts.values, color='blue',

         alpha=0.5, label='Number of Games')

        ax.set\_ylim(0, game\_counts.values.max() \* 1.1)

        ax.set\_xticks(game\_counts.index)

        ax.set\_xticklabels(['Loss', 'Win'])

    if not avg\_mmr\_by\_result.empty:

        ax2.plot(avg\_mmr\_by\_result.index, avg\_mmr\_by\_result.values,

         color='red', marker='o', label='Average MMR')

        ax2.set\_ylim(0, avg\_mmr\_by\_result.values.max() \* 1.1)

    ax.set\_xlabel('Game Result')

    ax.set\_ylabel('Number of Games', color='blue')

    ax2.set\_ylabel('Average MMR', color='red')

    ax.legend(loc='upper left')

    ax2.legend(loc='upper right')

# Функція для побудови гістограми

def plot\_histogram(ax, data, column, title, color):

    """ histogram"""

    ax.hist(data[column], bins=20, color=color, edgecolor='black')

    ax.set\_title(title)

    ax.set\_xlabel(column)

    ax.set\_ylabel('Frequency')

def explore\_data(data):

    """ explore data"""

    for column in data.columns:

        # Переконуємося, що стовпець містить числові дані

        if pd.api.types.is\_numeric\_dtype(data[column]):

            min\_value = data[column].min()

            max\_value = data[column].max()

            print(f"{column}: Мінімальне значення - {min\_value},Максимальне значення - {max\_value}")

def main() -> None:

    """ Main method"""

    data = load\_and\_preprocess('Data/games\_cleaned.csv')

    # Дослідження даних

    explore\_data(data)

    # Створення фігури з трьома піддіаграмами

    fig, axes = plt.subplots(1, 3, figsize=(15, 5))

    # Побудова комбінованого графіка

    plot\_combined\_chart(axes[0], data)

    # Побудова гістограми середнього MMR

    plot\_histogram(axes[1], data, 'avg\_mmr', 'Histogram of Average MMR', 'skyblue')

    # Побудова гістограми тривалості ігор

    plot\_histogram(axes[2], data, 'duration', 'Histogram of Game Duration', 'salmon')

    # Налаштування загального заголовку

    fig.suptitle('Game Results and Statistics')

    # Налаштування макету та відображення графіків

    plt.tight\_layout()

    plt.subplots\_adjust(top=0.88)  # Забезпечити місце для загального заголовку

    plt.savefig('Data/game\_results\_and\_statistics.png')

    plt.show()

# Основний блок коду

if \_\_name\_\_ == '\_\_main\_\_':

    main()

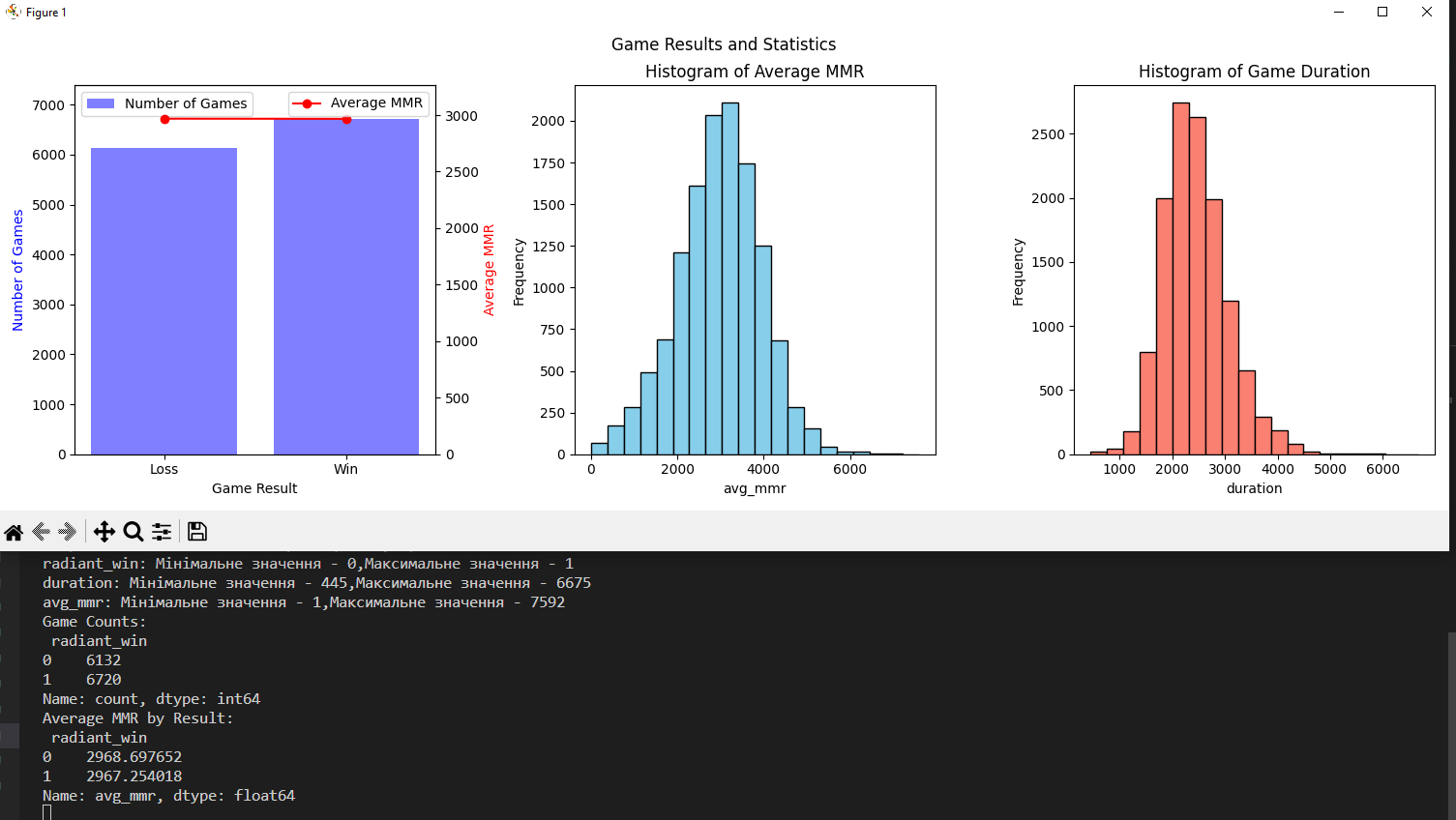


Рис.1 Результат виконання роботи

**Висновок**: Виконуючи 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