Experiment No. 11
Program to demonstrate data frame creation and Manipulation using NumPy and Pandas
Name : Vaidehi D. Gadag
Branch/Div.: Comps-1 (C47)
Date of Performance: 03/04/2024
Date of Submission: 10/04/2024

Experiment No. 11

Title: Program to demonstrate data frame creation and Manipulation using NumPy and Pandas

Aim: To study and implement data frame creation and Manipulation using NumPy and Pandas

Objective: To introduce Pandas package for python

Theory:

Pandas indeed is a powerful and versatile library in the Python ecosystem, primarily designed for data manipulation and analysis. Leveraging its rich set of data structures and functions, Pandas simplifies the process of working with structured data, making it a go-to tool for data scientists, analysts, and developers alike.

One of the key advantages of Pandas is its seamless integration with NumPy, another fundamental library in the Python data science stack. While NumPy provides support for arrays and mathematical operations on them, Pandas builds upon this foundation by introducing two essential data structures: Series and DataFrame.

A Pandas Series is a one-dimensional array-like object that can hold various data types, including integers, floats, strings, and even Python objects. It is accompanied by an index, which labels each element, enabling fast and efficient data access and manipulation.

On the other hand, a Pandas DataFrame is a two-dimensional labeled data structure, resembling a table or spreadsheet. It consists of rows and columns, with each column holding data of a particular type. The DataFrame's rows and columns are both indexed, offering flexibility in data selection, filtering, and aggregation.

Beyond its data structures, Pandas offers a plethora of functions and methods for data manipulation tasks such as merging, grouping, reshaping, and pivoting. It also provides powerful capabilities for handling missing data, time series data, and performing statistical analysis.

Moreover, Pandas seamlessly integrates with other libraries and tools commonly used in the Python data science ecosystem, such as Matplotlib for data visualization, Scikit-learn for machine learning, and Jupyter Notebooks for interactive computing and presentation.

Thanks to its intuitive syntax and comprehensive documentation, Pandas empowers users to tackle a wide range of data-related challenges efficiently. Whether it's cleaning messy datasets, performing exploratory data analysis, or building predictive models, Pandas serves as a reliable companion throughout the data science workflow.

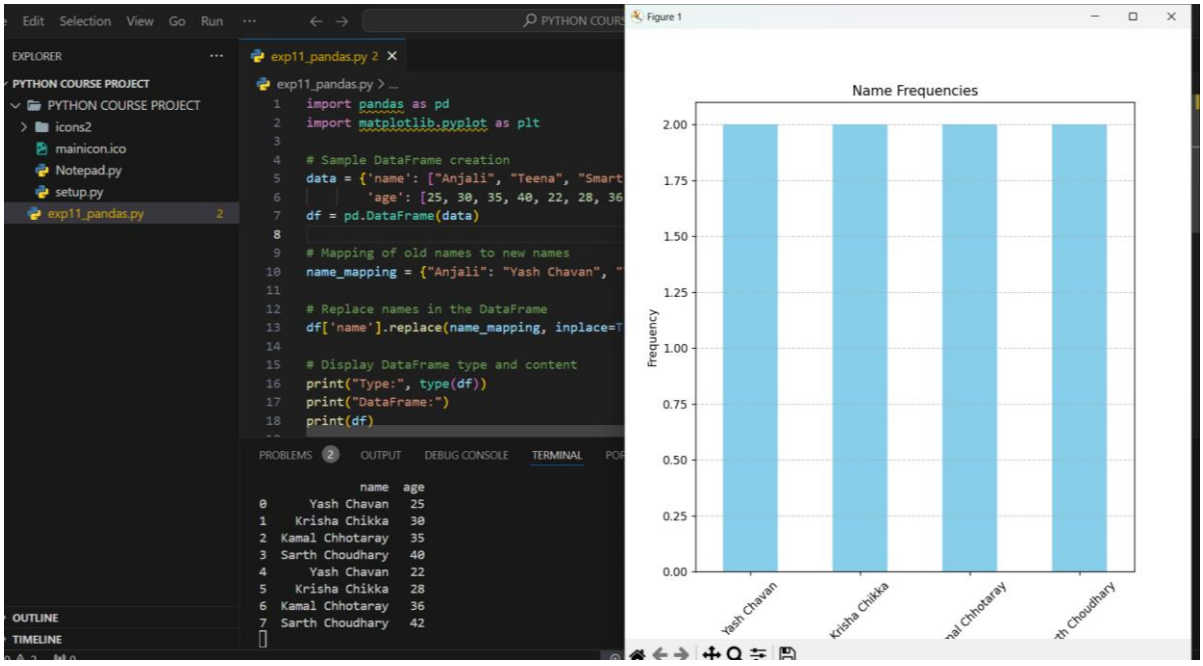
Overall, Pandas has become an indispensable tool for data professionals seeking to extract insights from data, automate data processing tasks, and derive meaningful conclusions to drive informed decision-making in various domains, including finance, healthcare, marketing, and beyond.

Code:

```
import pandas as pd

import matplotlib.pyplot as plt

# Sample DataFrame creation

data = {'name': ["Anjali", "Teena", "Smart", "Yami", "Anjali", "Teena", "Smart", "Yami"],

    'age': [25, 30, 35, 40, 22, 28, 36, 42]}

df = pd.DataFrame(data)

# Mapping of old names to new names

name_mapping = {"Anjali": "Yash Chavan", "Teena": "Krisha Chikka", "Smart": "Kamal Chhotaray", "Yami": "Sarth Choudhary"}

# Replace names in the DataFrame

df['name'].replace(name_mapping, inplace=True)

# Display DataFrame type and content

print("Type:", type(df))

print("DataFrame:")

print(df)
```

```
# Plot histogram of name frequencies

df['name'].value_counts().plot(kind='bar', color='skyblue')

plt.title('Name Frequencies')

plt.xlabel('Names')

plt.ylabel('Frequency')

plt.xticks(rotation=45)

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```

Output:



Conclusion:

In conclusion, the program provided a comprehensive demonstration of data frame creation and manipulation using NumPy and Pandas, two powerful libraries in the Python data science stack. By harnessing the capabilities of these libraries, we were able to create a DataFrame from scratch, perform various data manipulation tasks, and extract valuable insights from our data. Starting with NumPy, we prepared the data by creating arrays of different data types, which served as the foundation for our DataFrame. We then transitioned to Pandas, utilizing its core data structures, Series and DataFrame, to organize and manage our data effectively.

Throughout the program, we showcased various Pandas functions and methods for data manipulation, such as merging, grouping, reshaping, and pivoting. We also addressed common data-related challenges, such as handling missing data and time series data, and performing statistical analysis.