



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Experiment No. 11
Program to demonstrate data frame creation and Manipulation using Pandas
Date of Performance: 10/04/2024
Date of Submission: 17/04/2024



Experiment No. 11

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

Aim: To study and implement data frame creation and Manipulation using 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.



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

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": "pallavi dhandar", "Teena": "pranali chakor", "Smart": "anushka
chavan", "Yami": "Shruti borahde"}

# 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:

The screenshot shows the OnlineGDB beta IDE interface. The top bar includes a lightning bolt icon, the text 'OnlineGDB beta', and a description 'online compiler and debugger for c/c++'. Below this is a navigation menu with links: 'code. compile. run. debug. share.', 'IDE', 'My Projects', 'Classroom new', 'Learn Programming', 'Programming Questions', 'Jobs new', 'Sign Up', and 'Login'. The main editor area displays a Python script named 'main.py' with the following code:

```
1 import matplotlib.pyplot as plt
2
3
4 # Sample DataFrame creation
5 data = {'name': ["Anjali", "Teena", "Smart", "Yami", "Anjali", "Teena", "Smart", "Yami"],
6         'age': [25, 30, 35, 40, 22, 28, 36, 42]}
7 df = pd.DataFrame(data)
8
9 # Mapping of old names to new names
10 name_mapping = {"Anjali": "pallavi dhandar", "Teena": "pranali chakor", "Smart": "anushka chavan", "Yami": "shruti borahde"}
11
12 # Replace names in the DataFrame
13 df['name'].replace(name_mapping, inplace=True)
14
15 # Display DataFrame type and content
16 print(df)
```

The output window shows the following text:

```
Type: <class 'pandas.core.frame.DataFrame'>
DataFrame:
   name  age
0  pallavi dhandar  25
1  pranali chakor  30
2  anushka chavan  35
3  shruti borahde  40
4  pallavi dhandar  22
5  pranali chakor  28
6  anushka chavan  36
7  shruti borahde  42

/home/main.py:28: UserWarning: Matplotlib is currently using agg, which is a non-GUI backend, so cannot show the figure.
  plt.show()

...Program finished with exit code 0
Press ENTER to exit console.
```



Conclusion:

In conclusion, Pandas provides a powerful framework for creating and manipulating DataFrames, which are fundamental to data analysis in Python. With Pandas, users can efficiently handle structured data, perform various operations like filtering, grouping, and merging, and apply transformations easily. Its intuitive syntax and extensive functionality make it a preferred choice for data manipulation tasks. By leveraging Pandas, analysts and data scientists can streamline their workflows, extract insights, and derive value from their datasets effectively.