	Marwadi University **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on Pandas Data Structures	
Experiment No: 09	**Date:**	**Enrollment No:**

Aim: Practical based on Pandas Data Structures

IDE:

What is Python Pandas?

Pandas is a powerful, open-source data analysis and manipulation package for Python. It provides data structures and functions needed to work on structured data seamlessly and efficiently.

What Is Pandas Used For?

Pandas is extensively used for:

- Data Cleaning: Handling missing values, duplications, and incorrect data formats.
- Data Manipulation: Filtering, transforming, and merging datasets.
- Data Analysis: Performing statistical analysis and aggregations.
- Data Visualization: Creating plots and charts to visualize data trends and patterns.
- Time Series Analysis: Handling and manipulating time series data.

Run the following command to install Pandas:

```
pip install pandas
```

```
import pandas as pd
```

```
print(pd.__version__)
```

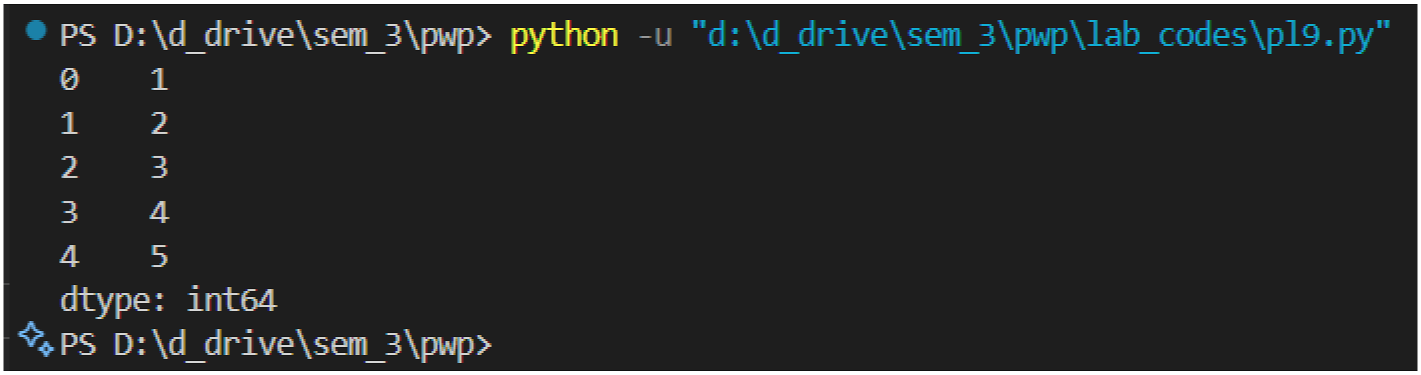
Pandas Series

A Pandas Series is a one-dimensional labeled array capable of holding any data type. It is similar to a column in a spreadsheet or a SQL table.

Example:

```
import pandas as pd
# Creating a Series
data = [1, 2, 3, 4, 5]
series = pd.Series(data)
print(series)
```

Output:



```
PS D:\d_drive\sem_3\pwp> python -u "d:\d_drive\sem_3\pwp\lab_codes\pl9.py"
0    1
1    2
2    3
3    4
4    5
dtype: int64
PS D:\d_drive\sem_3\pwp>
```

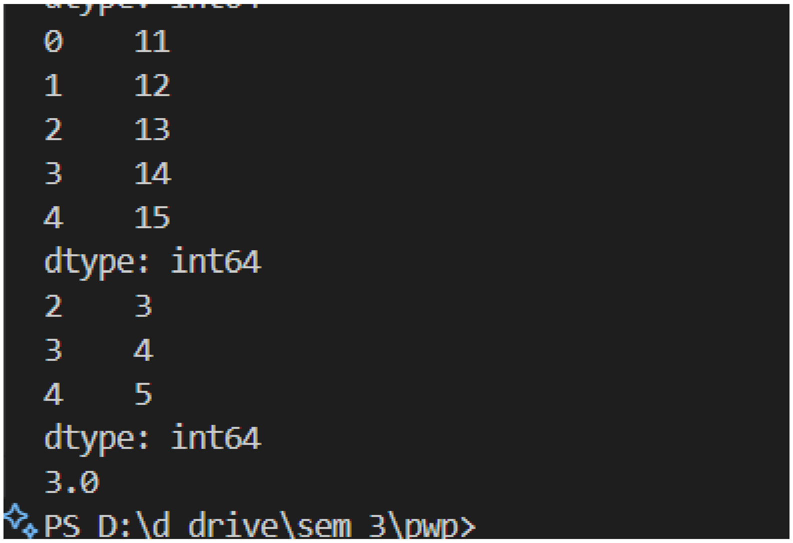
Basic Operations on Series

Perform various operations on Series, such as arithmetic operations, filtering, and statistical calculations.

Example:

```
# Arithmetic Operations
series2 = series + 10
print(series2)
# Filtering
filtered_series = series[series > 2]
print(filtered_series)
# Statistical Calculations
mean_value = series.mean()
print(mean_value)
```

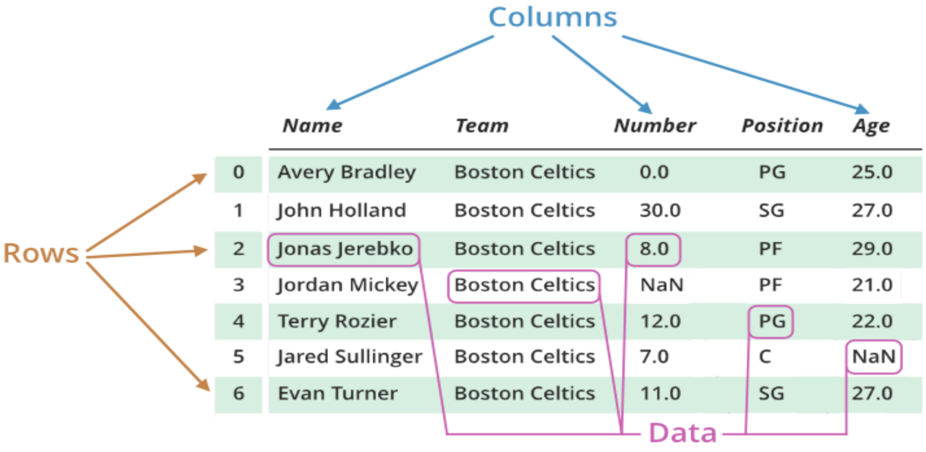
Output



```
dtype: int64
0    11
1    12
2    13
3    14
4    15
dtype: int64
2    3
3    4
4    5
dtype: int64
3.0
PS D:\d drive\sem 3\pwp>
```

Pandas Dataframe

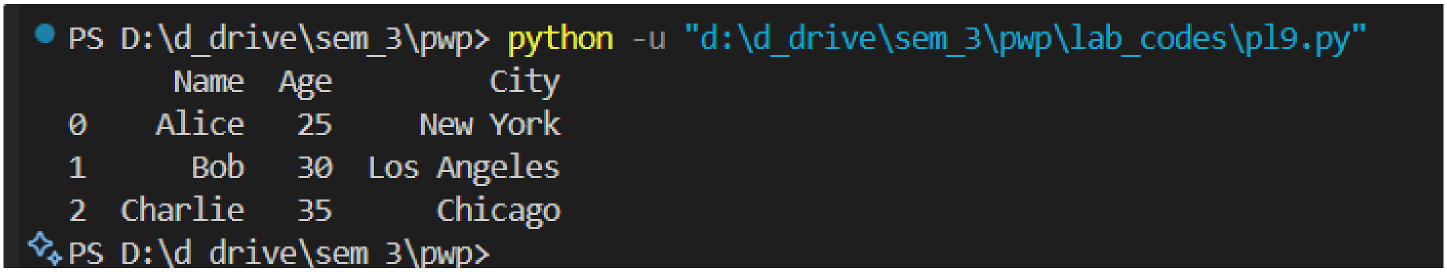
Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.



	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

```
# Creating a DataFrame
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
    'City': ['New York', 'Los Angeles', 'Chicago']
}
df = pd.DataFrame(data)
print(df)
```

Output



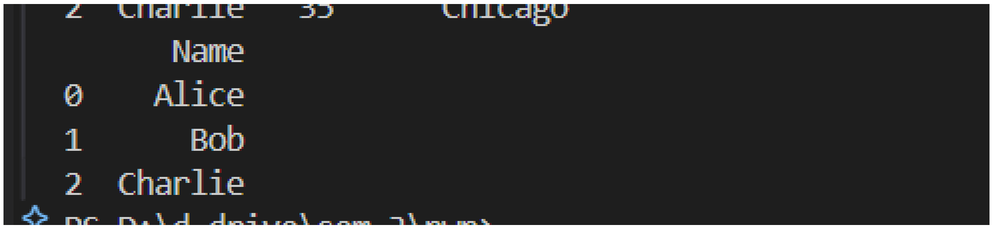
```
PS D:\d_drive\sem_3\pwp> python -u "d:\d_drive\sem_3\pwp\lab_codes\pl9.py"
      Name  Age         City
0    Alice   25     New York
1      Bob   30  Los Angeles
2  Charlie   35      Chicago
PS D:\d_drive\sem_3\pwp>
```

Basic Operations on Dataframes

DataFrames support a wide range of operations for data manipulation and analysis.

```
# Accessing Columns (# select one column)
print(df[['Name']])
```

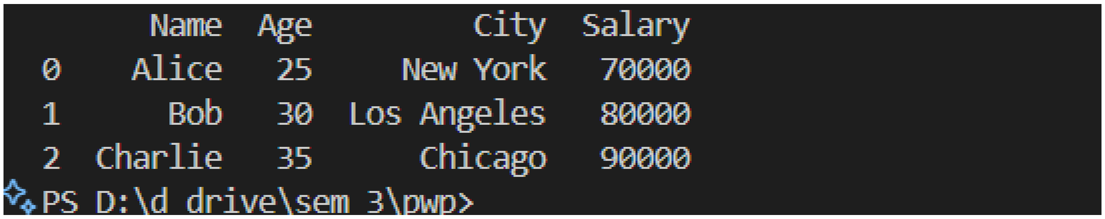
Output



```
2  Charlie   35      Chicago
      Name
0    Alice
1      Bob
2  Charlie
PS D:\d_drive\sem_3\pwp>
```

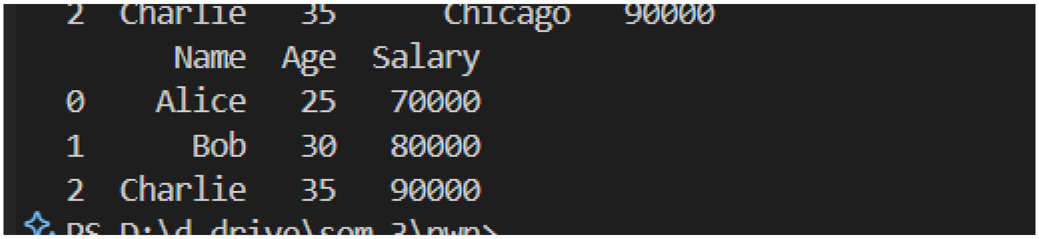
```
# Adding a New Column
df['Salary'] = [70000, 80000, 90000]
print(df)
```

Output



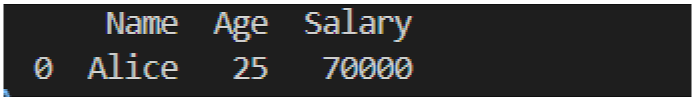
```
      Name  Age         City  Salary
0    Alice   25     New York   70000
1      Bob   30  Los Angeles   80000
2  Charlie   35      Chicago   90000
PS D:\d_drive\sem_3\pwp>
```

# Dropping a Column
df = df.drop('City', axis=1)
print(df)
Output



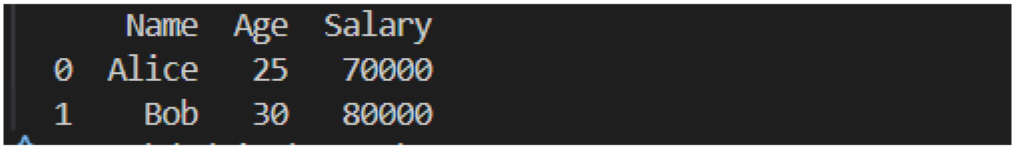
```
  2  Charlie   35      Chicago   90000
        Name  Age  Salary
  0    Alice   25   70000
  1      Bob   30   80000
  2  Charlie   35   90000
PS D:\d drive\sem 3\pwp>
```

The DataFrame is like a table with rows and columns.
Pandas use the loc attribute to return one or more specified row(s)
# Return row 0:
print(df.loc[[0]])
Output



```
     Name  Age  Salary
0  Alice   25   70000
```

#Return row 0 and 1:
#use a list of indexes:
print(df.loc[[0, 1]])
Output

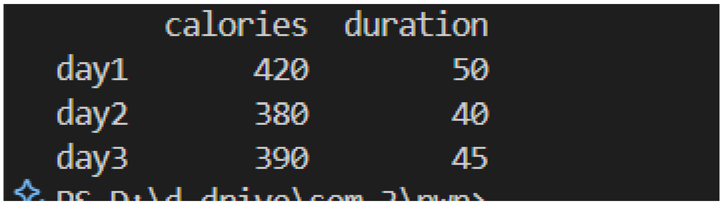


```
     Name  Age  Salary
0  Alice   25   70000
1    Bob   30   80000
```

Named Indexes
With the index argument, you can name your own indexes.
Example:
Add a list of names to give each row a name:
import pandas as pd
data = {
  "calories": [420, 380, 390],

```
  "duration": [50, 40, 45]
}
df = pd.DataFrame(data, index = ["day1", "day2", "day3"])
print(df)
```

Output



```
      calories  duration
day1       420        50
day2       380        40
day3       390        45
PS D:\d_drive\sem_3\pwp>
```

Explanation of Key Pandas Functions

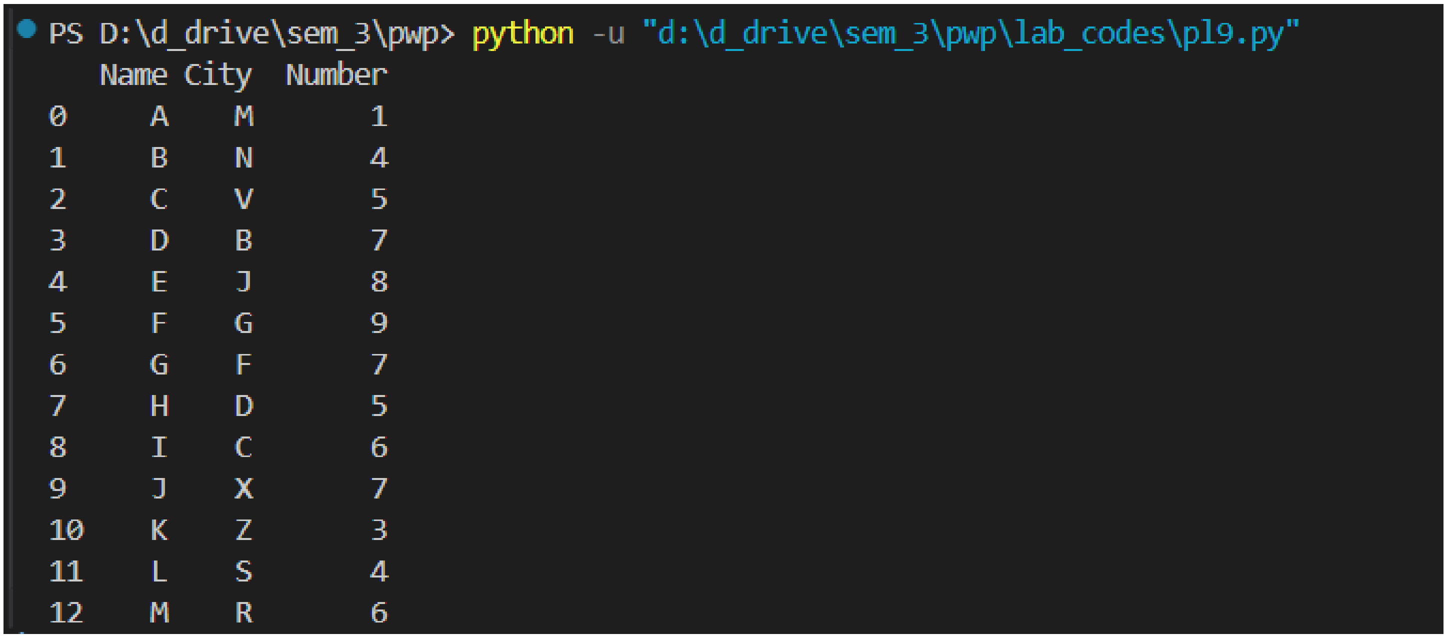
Reading and Writing Data:

Reading Data: Read a CSV file into a DataFrame.

Example:

```
dat = pd.read_csv("data.csv")
print(dat)
```

Output



```
PS D:\d_drive\sem_3\pwp> python -u "d:\d_drive\sem_3\pwp\lab_codes\pl9.py"
   Name City  Number
0     A    M       1
1     B    N       4
2     C    V       5
3     D    B       7
4     E    J       8
5     F    G       9
6     G    F       7
7     H    D       5
8     I    C       6
9     J    X       7
10    K    Z       3
11    L    S       4
12    M    R       6
```

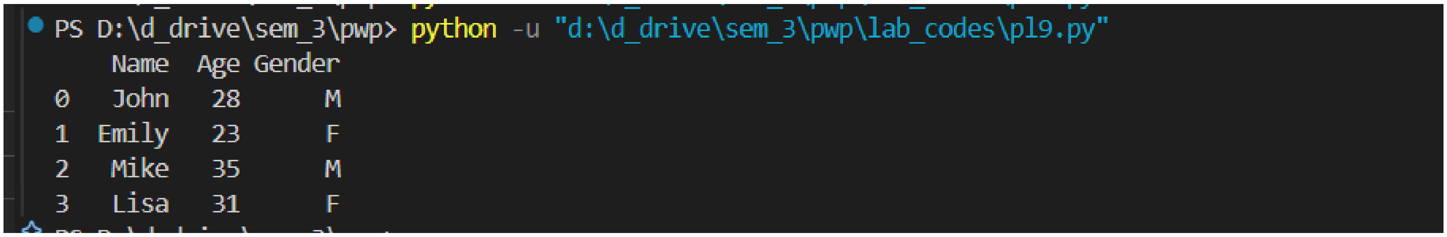
Writing Data: Write a DataFrame to a CSV file.

Note: Other Ways to Save Pandas DataFrames (to_excel(), to_json(), to_hdf(), to_sql(), to_pickle())

Example:

```
Biodata = {'Name': ['John', 'Emily', 'Mike', 'Lisa'],
        'Age': [28, 23, 35, 31],
        'Gender': ['M', 'F', 'M', 'F']
        }
df = pd.DataFrame(Biodata)
# Save the dataframe to a CSV file
df.to_csv('Biodata.csv', index=False)
```

Output



```
PS D:\d_drive\sem_3\pwp> python -u "d:\d_drive\sem_3\pwp\lab_codes\pl9.py"
    Name  Age Gender
0   John   28      M
1  Emily   23      F
2   Mike   35      M
3   Lisa   31      F
```

Data Inspection:

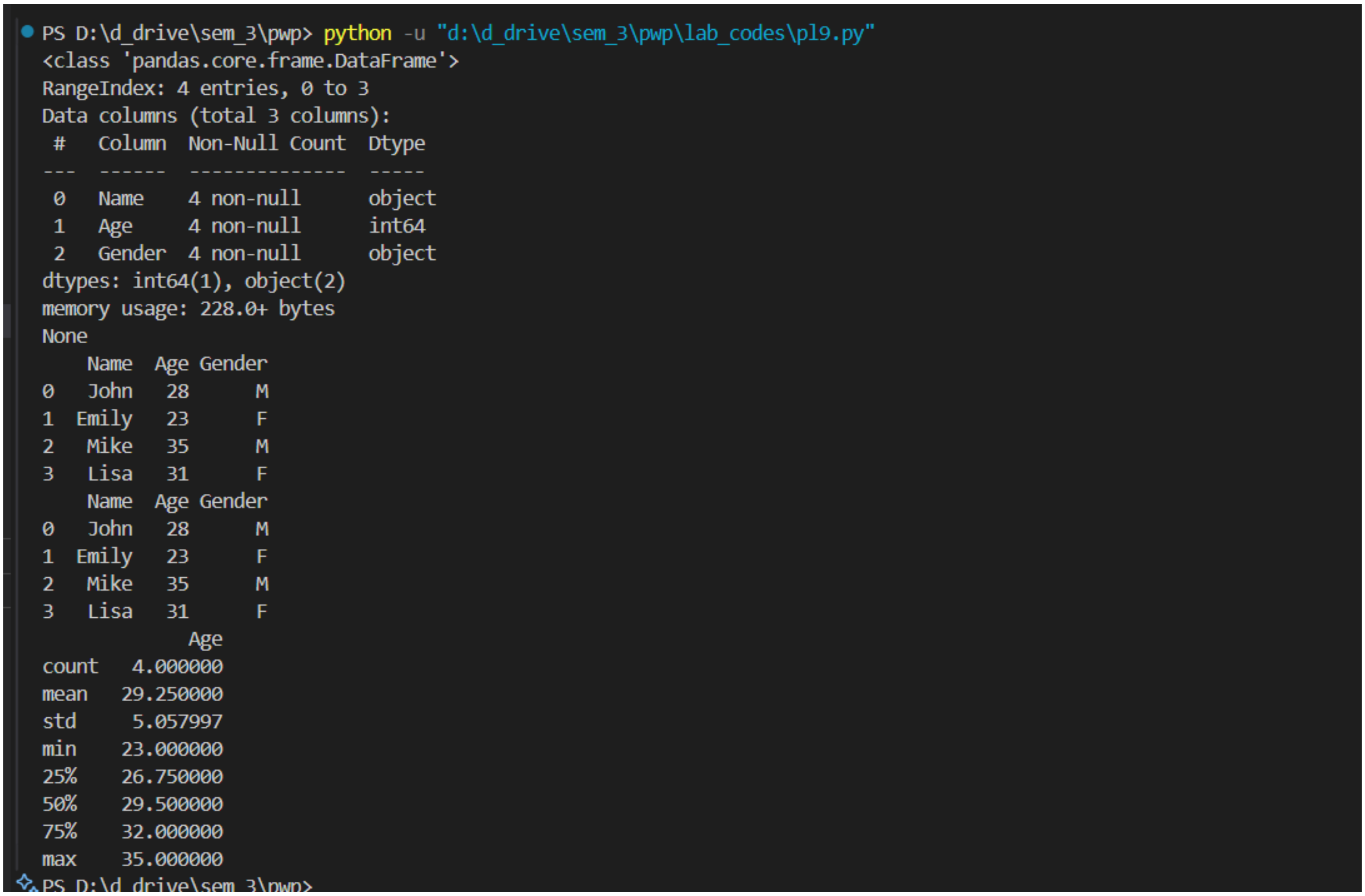
`df.head()`: Display the first few rows of the DataFrame.
`df.tail()`: Display the last few rows of the DataFrame.
`df.info()`: Display a summary of the DataFrame.
`df.describe()`: Provide descriptive statistics for numerical columns. (count: the number of non-null entries, mean: the mean value, std: the standard deviation, min: the minimum value, 25%, 50%, 75%: the lower, median, and upper quartiles, max: the maximum value)

Example:

```
dat = pd.read_csv("data.csv")
print(dat.info())
# shows first and last five rows
```

print(dat.head())
print(dat.tail())
print(dat.describe())

Output



```
PS D:\d_drive\sem_3\pwp> python -u "d:\d_drive\sem_3\pwp\lab_codes\pl9.py"
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Name    4 non-null      object
 1   Age     4 non-null      int64
 2   Gender  4 non-null      object
dtypes: int64(1), object(2)
memory usage: 228.0+ bytes
None
    Name  Age Gender
0   John   28      M
1  Emily   23      F
2   Mike   35      M
3   Lisa   31      F
    Name  Age Gender
0   John   28      M
1  Emily   23      F
2   Mike   35      M
3   Lisa   31      F
             Age
count   4.000000
mean   29.250000
std     5.057997
min    23.000000
25%    26.750000
50%    29.500000
75%    32.000000
max    35.000000
PS D:\d_drive\sem_3\pwp>
```

Data Selection and Indexing:

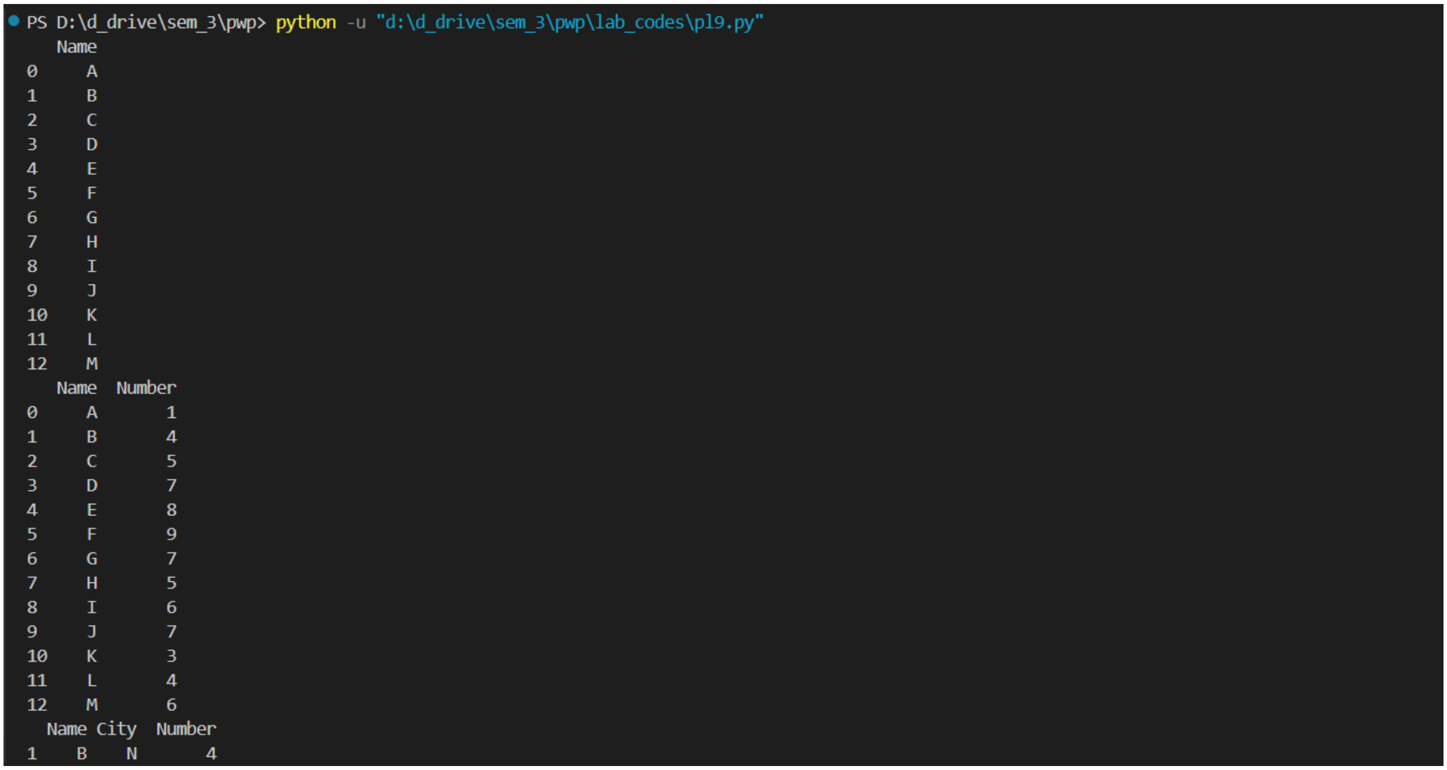
`dat[['A']]`: Select a column.

`dat[['A', 'B']]`: Select multiple columns.

`dat.loc[[0]]`: Select a row by label.

Example:

print(dat[['Name']])
print(dat[['Name','Number']])
print(dat.loc[[1]])
Output



```
PS D:\d_drive\sem_3\pwp> python -u "d:\d_drive\sem_3\pwp\lab_codes\pl9.py"
   Name
0     A
1     B
2     C
3     D
4     E
5     F
6     G
7     H
8     I
9     J
10    K
11    L
12    M
   Name  Number
0     A       1
1     B       4
2     C       5
3     D       7
4     E       8
5     F       9
6     G       7
7     H       5
8     I       6
9     J       7
10    K       3
11    L       4
12    M       6
  Name City  Number
1    B    N       4
```

Data Manipulation:

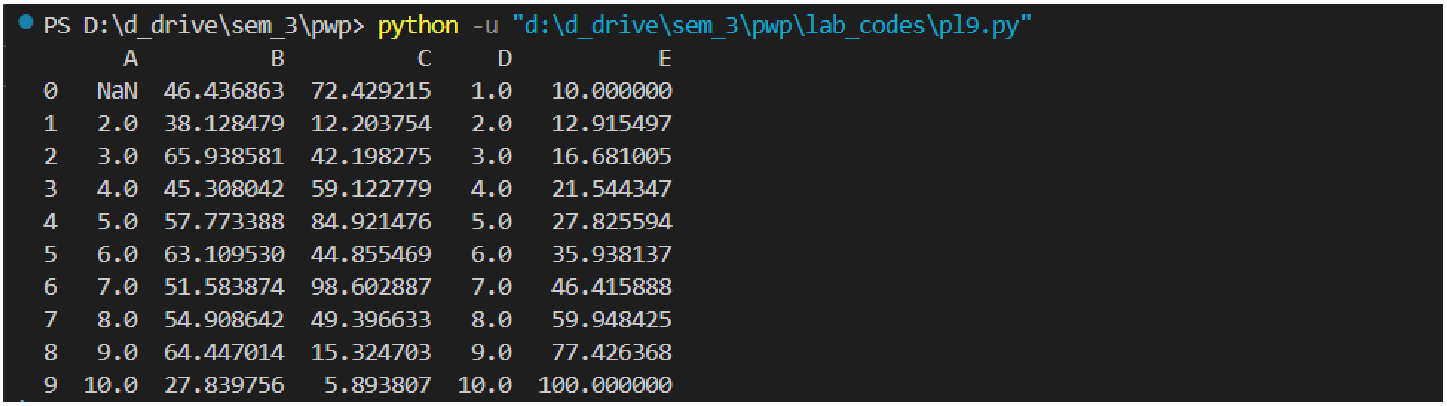
dat['A'] = dat['A'] * 2: Modify a column.
dat['F'] = dat['A'] + dat['B']: Create a new column based on existing columns.
dat.drop(columns=['A']): Drop a column.
dat.drop(index=[0]): Drop a row.

Task

Create a DataFrame with 5 numeric columns

```
data = {
    'A': [np.nan, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'B': np.random.normal(50, 15, 10),
    'C': np.random.rand(10) * 100,
    'D': np.linspace(1, 10, 10),
    'E': np.logspace(1, 2, 10)
}
df = pd.DataFrame(data)
```

Output



```
PS D:\d_drive\sem_3\pwp> python -u "d:\d_drive\sem_3\pwp\lab_codes\pl9.py"
      A          B          C     D           E
0   NaN  46.436863  72.429215   1.0   10.000000
1   2.0  38.128479  12.203754   2.0   12.915497
2   3.0  65.938581  42.198275   3.0   16.681005
3   4.0  45.308042  59.122779   4.0   21.544347
4   5.0  57.773388  84.921476   5.0   27.825594
5   6.0  63.109530  44.855469   6.0   35.938137
6   7.0  51.583874  98.602887   7.0   46.415888
7   8.0  54.908642  49.396633   8.0   59.948425
8   9.0  64.447014  15.324703   9.0   77.426368
9  10.0  27.839756   5.893807  10.0  100.000000
```

Post Lab Exercise:

a. Write a Pandas program to add, subtract, multiple and divide two Pandas Series.

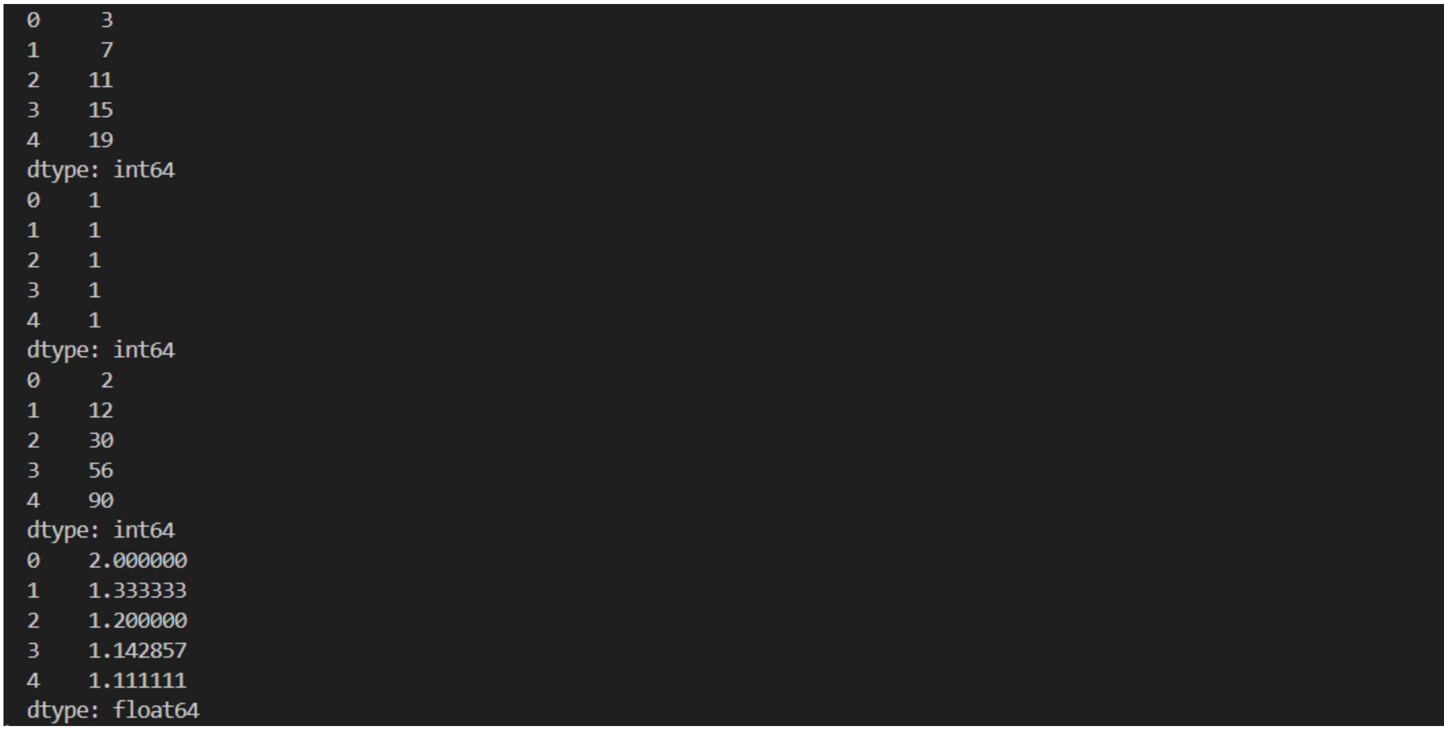
Code :

```
import pandas as pd

s1 = pd.Series([2, 4, 6, 8, 10])
s2 = pd.Series([1, 3, 5, 7, 9])

print(s1 + s2)
print(s1 - s2)
print(s1 * s2)
print(s1 / s2)
```

Output:



```
0     3
1     7
2    11
3    15
4    19
dtype: int64
0    1
1    1
2    1
3    1
4    1
dtype: int64
0     2
1    12
2    30
3    56
4    90
dtype: int64
0    2.000000
1    1.333333
2    1.200000
3    1.142857
4    1.111111
dtype: float64
```

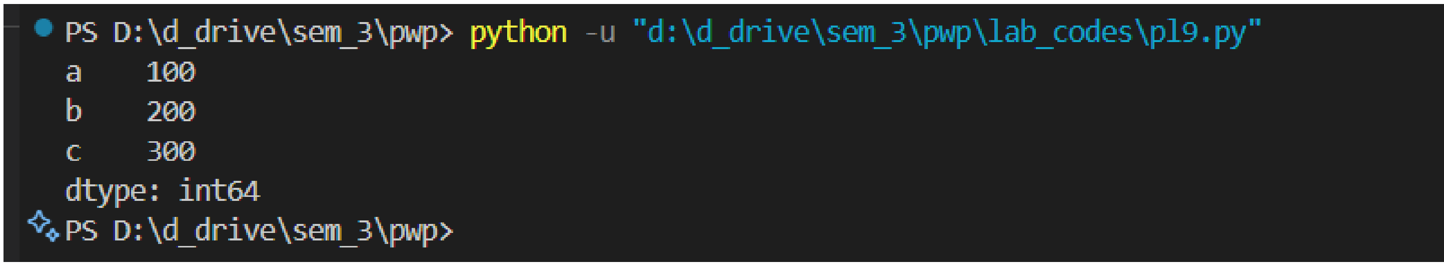
b. Write a Pandas program to convert a dictionary to a Pandas series.
Code :

```
import pandas as pd

data = {'a': 100, 'b': 200, 'c': 300}
s = pd.Series(data)

print(s)
```

Output :



```
PS D:\d_drive\sem_3\pwp> python -u "d:\d_drive\sem_3\pwp\lab_codes\pl9.py"
a    100
b    200
c    300
dtype: int64
PS D:\d_drive\sem_3\pwp>
```

c. Write a Pandas program to create a series from a list, numpy array and dict

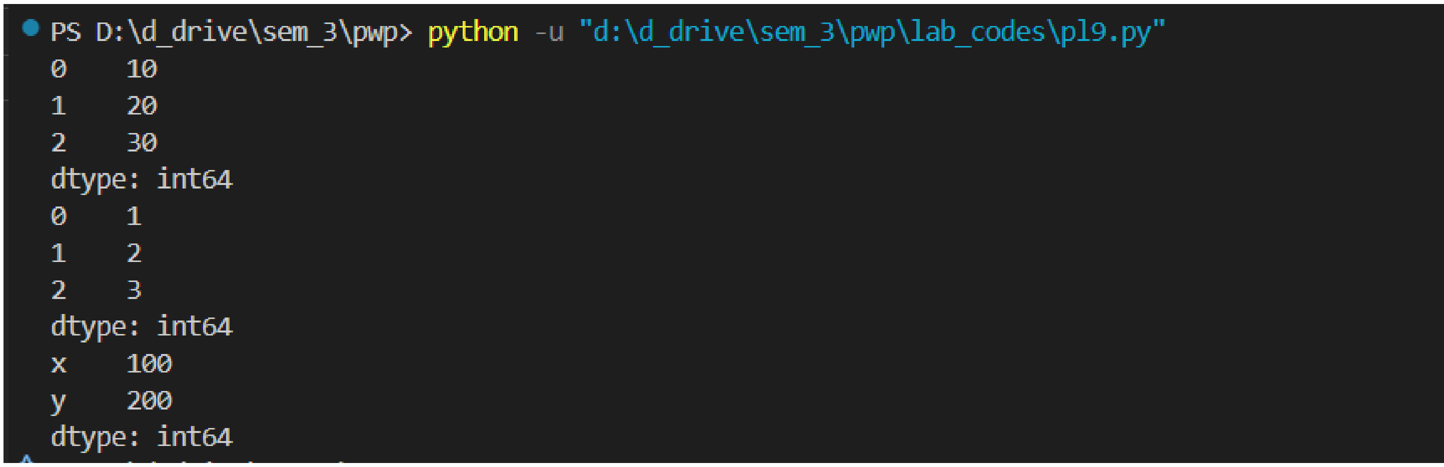
Code :

```
import pandas as pd
import numpy as np

s1 = pd.Series([10, 20, 30])
s2 = pd.Series(np.array([1, 2, 3]))
s3 = pd.Series({'x': 100, 'y': 200})

print(s1)
print(s2)
print(s3)
```

Output :



```
PS D:\d_drive\sem_3\pwp> python -u "d:\d_drive\sem_3\pwp\lab_codes\pl9.py"
0    10
1    20
2    30
dtype: int64
0    1
1    2
2    3
dtype: int64
x    100
y    200
dtype: int64
```

d. Write a Pandas program to stack two series vertically and horizontally

Code :

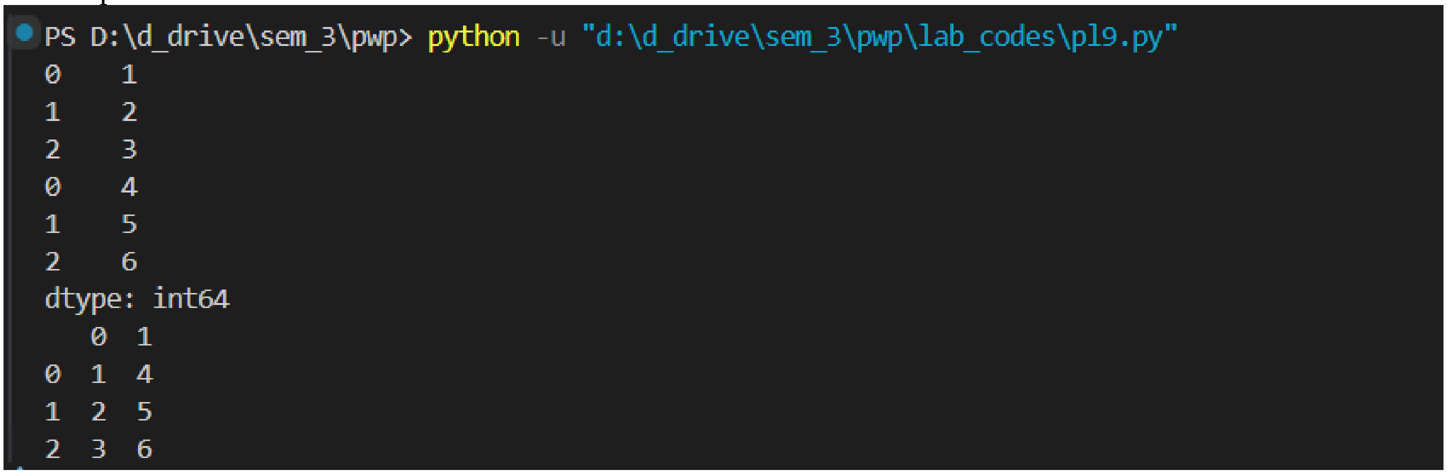
```
import pandas as pd

s1 = pd.Series([1, 2, 3])
s2 = pd.Series([4, 5, 6])

vertical = pd.concat([s1, s2])
horizontal = pd.concat([s1, s2], axis=1)

print(vertical)
print(horizontal)
```

Output :



```
PS D:\d_drive\sem_3\pwp> python -u "d:\d_drive\sem_3\pwp\lab_codes\pl9.py"
0    1
1    2
2    3
0    4
1    5
2    6
dtype: int64
   0  1
0  1  4
1  2  5
2  3  6
```