Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.

Installation of Pandas

```
!pip install pandas
```

```
Requirement already satisfied: pandas in
/usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.22.4 in
/usr/local/lib/python3.10/dist-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2-
>pandas) (1.17.0)
```

Import Pandas

```
import pandas as pd #importing pandas and providing it with an alias
```

Series

A **Pandas Series** is like a column in a table.

It is a *one-dimensional array* holding data of any type.

```
a = [1, 7, 2, 4, 9, 8]

myNum = pd.Series(a)

print(myNum)
```

```
0    1
1    7
2    2
3    4
4    9
5    8
dtype: int64
```

```
print(myNum[0])
print(myNum[5])
```

```
1
8
```

Labels

With the *index* argument, you can name your own labels.

```
a = [10, 17, 21]

myNum = pd.Series(a, index = ["a", "b", "c"])

print(myNum)

a    10
b    17
c    21
dtype: int64

print(myNum["a"])

10
```

Key/Value Objects as Series

```
running = {"day1": 2, "day2": 3, "day3": 5}

myRun = pd.Series(running)

print(myRun)

day1    2
day2    3
day3    5
dtype: int64
```

DataFrames

Data sets in Pandas are usually *multi-dimensional* tables, called **DataFrames**.

Series is like a column, a DataFrame is the whole table.

```
data = {
  "kilometers": [4, 3, 5],
  "duration": [50, 40, 45]
}

myRun = pd.DataFrame(data)

print(myRun)

   kilometers  duration
0           4        50
1           3        40
2           5        45
```

```
print(myRun.info())       # Overview of the dataset
print(myRun.describe())   # Summary statistics

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   kilometers  3 non-null      int64
 1   duration    3 non-null      int64
dtypes: int64(2)
memory usage: 176.0 bytes
None
       kilometers  duration
count         3.0       3.0
mean          4.0      45.0
std           1.0       5.0
min           3.0      40.0
25%           3.5      42.5
50%           4.0      45.0
75%           4.5      47.5
max           5.0      50.0

data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35]}
df = pd.DataFrame(data)
print(df)

      Name  Age
0    Alice   25
1      Bob   30
2  Charlie   35
```

Loading Data from a File

```
mydf =
pd.read_csv('https://raw.githubusercontent.com/gagan-iitb/DataAnalytic
sAndVisualization/refs/heads/main/Lab-W25/dataset/names.csv')
```

Download CSV - names.csv

```
print(mydf.head())  # Display the first 5 rows

      Name  Age
0    Alice   25
1      Bob   30
2  Charlie   35
3    James   23
4     John   26
```

```
print(mydf.head(7))  # Display the first 7 rows
```

```
      Name  Age
0    Alice   25
1      Bob   30
2  Charlie   35
3    James   23
4     John   26
5  William   28
6    Caleb   25
```

```
print(mydf['Name'])  # Single column
```

```
0      Alice
1        Bob
2    Charlie
3      James
4       John
5    William
6      Caleb
7      Helen
Name: Name, dtype: object
```

```
print(mydf[['Age', 'Name']])  # Multiple columns
```

```
   Age     Name
0   25    Alice
1   30      Bob
2   35  Charlie
3   23    James
4   26     John
5   28  William
6   25    Caleb
7   30    Helen
```

Filtering Rows

```
print(mydf[mydf['Age'] > 25])
```

```
      Name  Age
1      Bob   30
2  Charlie   35
4     John   26
5  William   28
7    Helen   30
```

Adding/Updating Columns

```
mydf['Salary'] = [50000, 60000, 50000, 50000, 30000, 70000, 90000,
80000]
print(mydf)
```

```
      Name  Age  Salary
0    Alice   25   50000
1      Bob   30   60000
2  Charlie   35   50000
3    James   23   50000
4     John   26   30000
5  William   28   70000
6    Caleb   25   90000
7    Helen   30   80000
```

Saving to a File

```
mydf.to_csv('myDataframe.csv', index=False)
```

Dropping Columns

```
mydf = mydf.drop('Salary', axis=1)  # Drop column

print(mydf)
```

```
      Name  Age
0    Alice   25
1      Bob   30
2  Charlie   35
3    James   23
4     John   26
5  William   28
6    Caleb   25
7    Helen   30
```

```
#Create/Append two new columns named Marks, Department in mydf and
display it
import pandas as pd

mydf['Marks'] = [85, 90, 88, 75, 80, 95, 85, 90]
mydf['Department'] = ['Design', 'Marketing', 'Development',
'Management', 'Marketing', 'Design', 'Development', 'Design']
print(mydf)
```

```
      Name  Age  Marks   Department
0    Alice   25     85       Design
1      Bob   30     90    Marketing
2  Charlie   35     88  Development
3    James   23     75   Management
4     John   26     80    Marketing
5  William   28     95       Design
```

```
6    Caleb   25     85  Development
7    Helen   30     90       Design
```

```
#Save the newly create mydf to a csv file. (Name of file =
myDataframe_YourIDNumber.csv)
mydf.to_csv('myDataframe_12340220.csv', index=False)
print("DataFrame has been saved to 'myDataframe_12340220.csv'.")
```

```
DataFrame has been saved to 'myDataframe_12340220.csv'.
```

```
#Filter all the rows where Age falls between 25-30.
filtered_df = mydf[(mydf['Age'] >= 25) & (mydf['Age'] <= 30)]
print(filtered_df)
```

```
      Name  Age  Marks   Department
0    Alice   25     85       Design
1      Bob   30     90    Marketing
4     John   26     80    Marketing
5  William   28     95       Design
6    Caleb   25     85  Development
7    Helen   30     90       Design
```

Unique() function

```
mydf.Age.unique()
```

```
array([25, 30, 35, 23, 26, 28])
```

Sorting

```
mydf.sort_values(by=['Age'])
```

```
{"summary":"{\n  \"name\": \"mydf\",\n  \"rows\": 8,\n  \"fields\": [\
n    {\n      \"column\": \"Name\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 8,\n
\"samples\": [\n          \"Alice\",\n          \"Bob\",\n
\"James\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Age\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 3,\n        \"min\": 23,\n        \"max\": 35,\n
\"num_unique_values\": 6,\n        \"samples\": [\n          23,\n
25,\n          35\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Marks\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 6,\n        \"min\": 75,\n        \"max\": 95,\n
\"num_unique_values\": 6,\n        \"samples\": [\n          75,\n
85,\n          88\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Department\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 4,\n        \"samples\":
```

```
[\n          \"Design\",\n          \"Marketing\",\n
\"Management\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```
#Sort mydf dataframe on the basis of Name,Marks.
mydf.sort_values(by=['Name', 'Marks'])
```

```
{"summary":"{\n  \"name\": \"mydf\",\n  \"rows\": 8,\n  \"fields\": [\
n    {\n      \"column\": \"Name\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 8,\n
\"samples\": [\n          \"Bob\",\n          \"James\",\n
\"Alice\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Age\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 3,\n        \"min\": 23,\n        \"max\": 35,\n
\"num_unique_values\": 6,\n        \"samples\": [\n          25,\n
30,\n          28\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Marks\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 6,\n        \"min\": 75,\n        \"max\": 95,\n
\"num_unique_values\": 6,\n        \"samples\": [\n          85,\n
90,\n          95\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Department\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          \"Marketing\",\n          \"Management\",\n
\"Design\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

Missing Data

```
data = {
    "Name": ["Alice", "Bob", "Charlie", "Diana", "Eve", "Frank",
"Grace", "Hank"],
    "Gender": ["Female", "Male", None, "Female", None, "Male",
"Female", None],
}
df = pd.DataFrame(data)
print(df)
```

```
      Name  Gender
0    Alice  Female
1      Bob    Male
2  Charlie    None
3    Diana  Female
4      Eve    None
5    Frank    Male
6    Grace  Female
7     Hank    None
```

```
print("\nCheck for missing values:")
print(pd.isnull(df))
```

```
Check for missing values:
    Name  Gender
0  False   False
1  False   False
2  False    True
3  False   False
4  False    True
5  False   False
6  False   False
7  False    True
```

```
print("\nCheck for missing values(Column):")
print(pd.isnull(df['Gender']))
```

```
Check for missing values(Column):
0    False
1    False
2     True
3    False
4     True
5    False
6    False
7     True
Name: Gender, dtype: bool
```

```
# Fill missing values in the 'Gender' column with a default value
df['Gender'] = df['Gender'].fillna("Not Specified")

#updated dataframe
print(df)
```

```
      Name         Gender
0    Alice         Female
1      Bob           Male
2  Charlie  Not Specified
3    Diana         Female
4      Eve  Not Specified
5    Frank           Male
6    Grace         Female
7     Hank  Not Specified
```

```
#Read myStudentDataFrame.csv
import pandas as pd
df = pd.read_csv('myDataframe_12340220.csv')
print(df)
```

```
      Name  Age  Marks   Department
0    Alice   25     85       Design
1      Bob   30     90    Marketing
2  Charlie   35     88  Development
3    James   23     75   Management
4     John   26     80    Marketing
5  William   28     95       Design
6    Caleb   25     85  Development
7    Helen   30     90       Design
```

```
#Check for missing data in all columns using appropriate pandas functions.
missing_data = df.isnull().sum()
print("Missing Data in Columns:\n", missing_data)
```

```
Missing Data in Columns:
 Name          0
Age           0
Marks         0
Department    0
dtype: int64
```

```
#Drop Rows with Missing Data
df_cleaned = df.dropna()

#Compute Summary Statistics (AVG,MEAN,MAX,MIN)
summary_statistics = {
    "Mean": df_cleaned['Marks'].mean(),
    "Max": df_cleaned['Marks'].max(),
    "Min": df_cleaned['Marks'].min(),
    "Avg": df_cleaned['Marks'].mean()
}
print("\nSummary Statistics:\n", summary_statistics)
```

```

Summary Statistics:
 {'Mean': 86.0, 'Max': 95, 'Min': 75, 'Avg': 86.0}
```

```
#Filter Data and Compute Pass/Fail
#mark >= 40: Pass
#mark < 40: Fail
#Add a new column Result to the DataFrame indicating Pass or Fail.

df_cleaned['Result'] = df_cleaned['Marks'].apply(lambda x: 'Pass' if x >= 40 else 'Fail')

#Save the Final DataFrame
#Save the updated DataFrame (with the Result column) to a new CSV file named Result_YourIDNumber.csv.

df_cleaned.to_csv('Result_12340220.csv', index=False)
```

```
print("\nUpdated DataFrame with 'Result' column saved to
'Result_12340220.csv'.")
```

```
Updated DataFrame with 'Result' column saved to 'Result_12340220.csv'.
```

Additional Practice Questions - Click Here