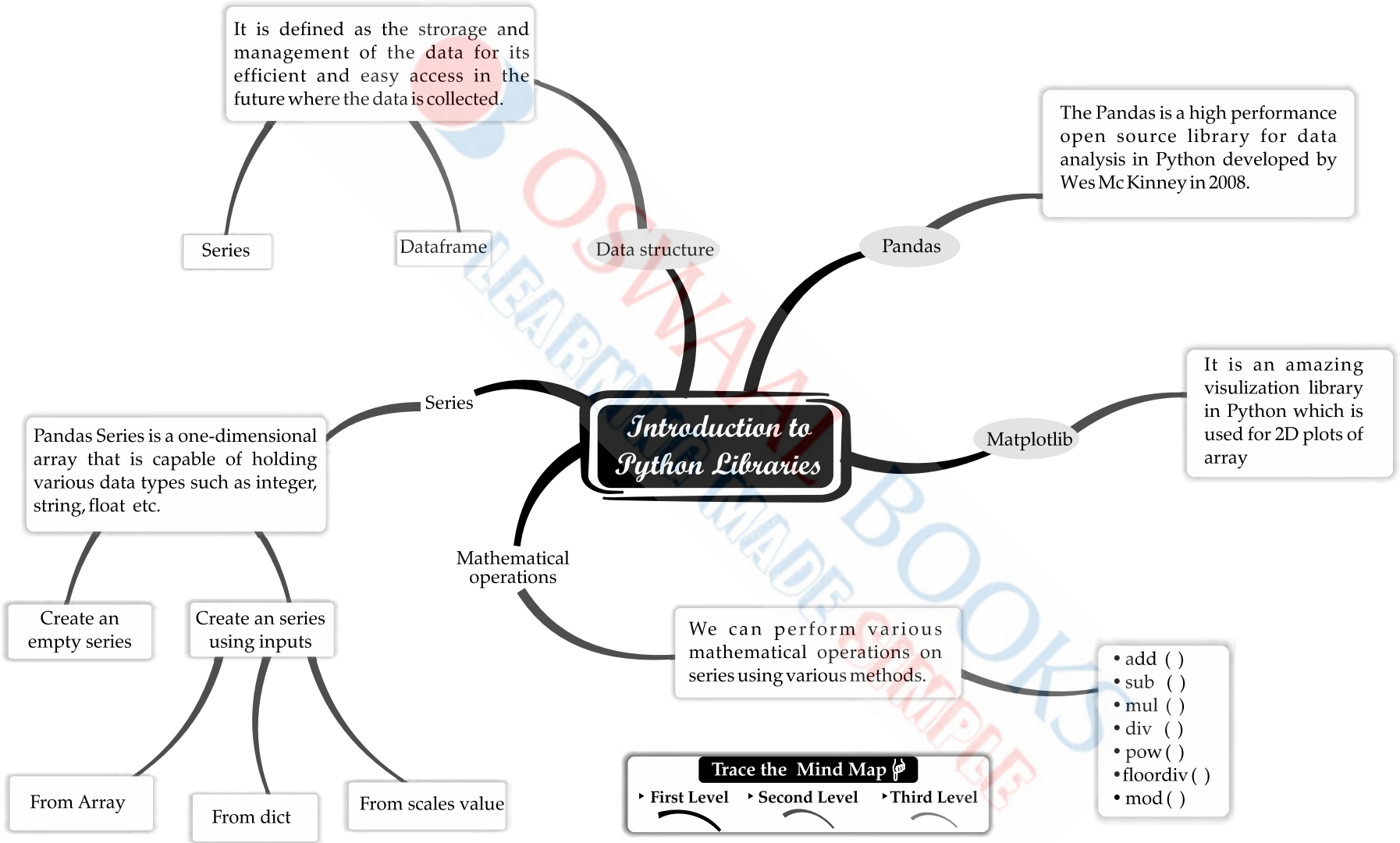
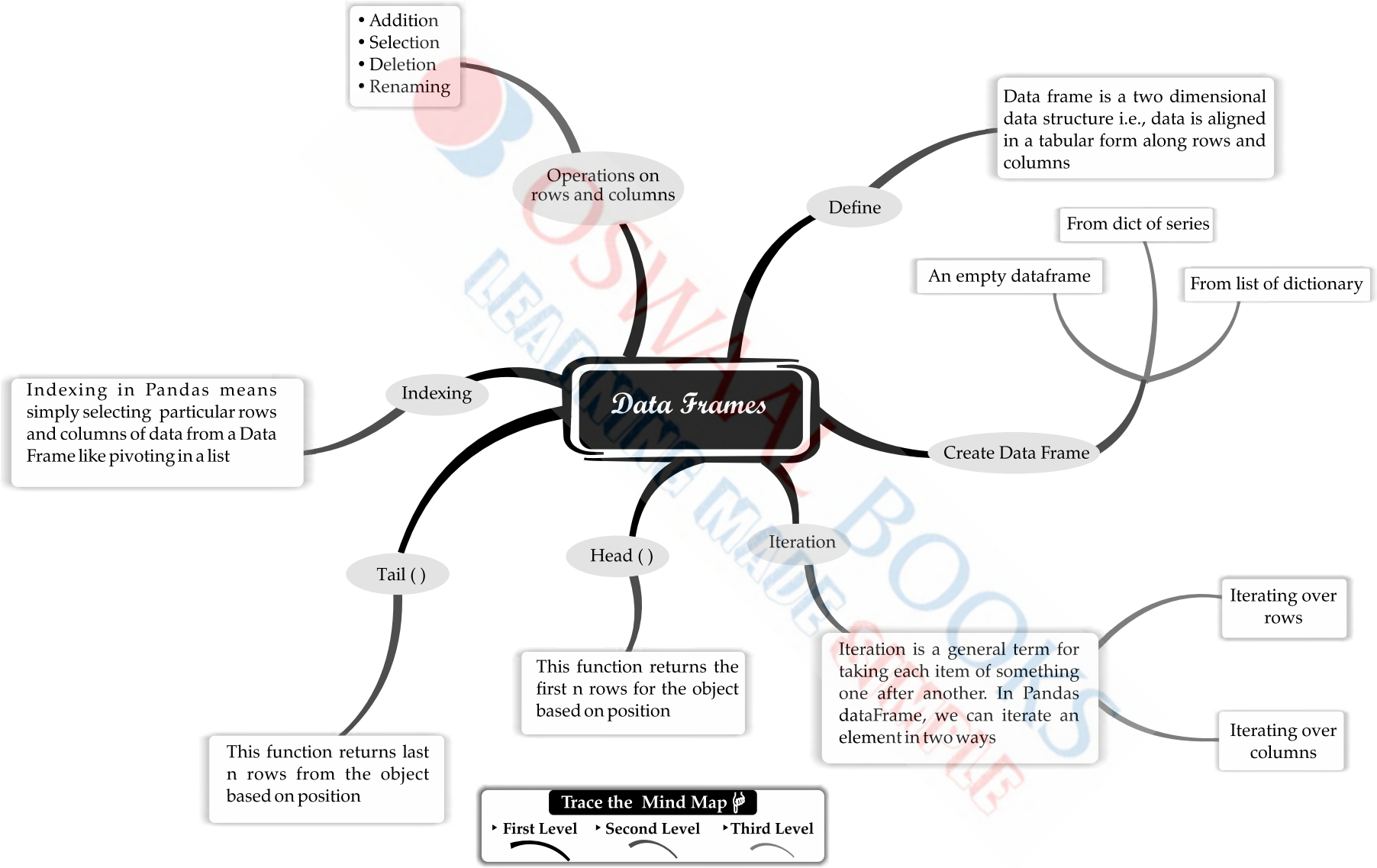

Introduction to Python Libraries
Data structure
It is defined as the storage and management of the data for its efficient and easy access in the future where the data is collected.
Series
Dataframe
Pandas
The Pandas is a high performance open source library for data analysis in Python developed by Wes Mc Kinney in 2008.
Matplotlib
It is an amazing visulization library in Python which is used for 2D plots of array
Series
Pandas Series is a one-dimensional array that is capable of holding various data types such as integer, string, float etc.
Create an empty series
Create an series using inputs
From Array
From dict
From scales value
Mathematical operations
We can perform various mathematical operations on series using various methods.
• add ( )
• sub ( )
• mul ( )
• div ( )
• pow ( )
• floordiv ( )
• mod ( )
Trace the Mind Map
First Level
Second Level
Third Level


Data Frames
Operations on rows and columns
• Addition
• Selection
• Deletion
• Renaming
Define
Data frame is a two dimensional data structure i.e., data is aligned in a tabular form along rows and columns
Create Data Frame
From dict of series
An empty dataframe
From list of dictionary
Indexing
Indexing in Pandas means simply selecting particular rows and columns of data from a Data Frame like pivoting in a list
Head ( )
This function returns the first n rows for the object based on position
Tail ( )
This function returns last n rows from the object based on position
Iteration
Iteration is a general term for taking each item of something one after another. In Pandas dataFrame, we can iterate an element in two ways
Iterating over rows
Iterating over columns
Trace the Mind Map
First Level
Second Level
Third Level

CHAPTER 1

DATA HANDLING USING PANDAS-I

Syllabus

- ***Introduction to Python libraries- Pandas, Matplotlib.***
- ***Data structures in Pandas - Series and data frames.***
- ***Series: Creation of series from ndarray, dictionary, scalar value; mathematical operations; series attributes, head and tail functions; selection, indexing and slicing.***
- ***Data Frames: creation of data frames from dictionary of series, list of dictionaries, text/CSV files, display, iteration. Operations on rows and columns: add add (insert /append) , select, delete (drop column and row), rename, Head and Tail functions, indexing using labels, Boolean indexing.***

Revision Notes

Introduction To Python Libraries

- **Pandas**

The Pandas is a high-performance open source library for data analysis in Python, developed by Wes McKinney in 2008. Over the years, it has become the de-facto standard library for data analysis using Python.

There are 3 well-established python libraries namely NumPy, Pandas and Matplotlib specially for scientific and analytical use.

These libraries allow us to manipulate, transform and visualise data easily and efficiently.

Using the Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data. These steps are- load, prepare, manipulate, model and analyse.

Some key features of Pandas include the following :

Scan to know more about this topic

Python Pandas Series Data Structure

(i) It can process a variety of data sets in different formats : time series, tabular heterogeneous arrays and matrix data.
(ii) It facilitates loading and importing data from varied sources such as CSV and DB/SQL.
(iii) It can handle a myriad of operations on data sets : sub-setting, slicing, filtering, merging, grouping, re-ordering, and re-shaping.
(iv) It can deal with missing data according to rules defined by the user and developer.
(v) It can be used for parsing and managing (conversion) of data as well as modeling and statistical analysis.
(vi) It integrates well with other Python libraries such as SciPy.
(vii) It delivers fast performance and can be speeded up even more by making use of Cython (C extensions to Python).

➢ **Benefits of Pandas**

The benefits of pandas over using the languages are

- **Data representation :** It can easily represent data in a form naturally suited for data analysis through its DataFrame and Series data structures in a concise manner. Doing the equivalent in C/C++ or Java would require many lines of custom code, as these languages were not built for data analysis but rather networking and kernel development.
- **Clear code :** The clear API of the Pandas allows you to focus on the core part of the code. So, it provides clear code.

➢ **Matplotlib**

It is an amazing visualization library in Python that used for 2D plots of arrays. It is a multi-platform data visualization library which build NumPy arrays. Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers and various graphical user interface toolkits.

To get matplotlib up and running in our environment, we need to import it.

import matplotlib pyplet as plt

➢ **Data structures in Pandas**

Data structure is defined as the storage and management of the data for its efficient and easy access in the future where the data is collected, modified and the various types of operations are performed on the data respectively. Pandas provides two data structures for processing the data, which are explained below :

(1) Series : It is one dimensional object similar to an array, list or column in a table. It will assign a labelled index to each item in the series. By default, each item will receive an index label from 0 to N, where N is the length of the series minus one.

(2) Data Frame : It is a tabular data structure comprised of rows and columns. DataFrame is defined as a standard way to store data which has two different indexes *i.e.*, row index and column index.

Scan to know more about this topic

Data Structure of Pandas

➢ **Series**

Pandas series is one-dimensional array that is capable of holding various data types such as integer, string, float, object etc. with the the help of series () method, we can easily change the list, dictionary into series. A series does not contain multiple columns and rows. Labels of series are called index.

Syntax

pandas.series (data, index, dtype, copy)

Here, data can be list, tuple etc

index value should be unique and hashable

dtype defines data type of series

copy copies the data

➢ **Creating a series**

(i) Create an empty series

Empty series means it will not have any value.

Syntax

series_Object = pandas.Series ()

(ii) Create a series using Inputs

We can create series by using various inputs as array.

- **Creating series from array :** To create a series from array, we have to import the numpy module and then use array () method. If data is an ndarray, then passed index must be of same length. If there is no index passed as an argument, then index will be range (*n*) by default, where *n* is array length.

- **Creating series from dict :** A dict can be passed as an input. If there is no index specified, then the dictionary's keys are taken in a sorted order. If index is passed, then corresponding values to a particular label in the index will be extracted from the dictionary.
- **Creating series from scalar :** If data is a scalar value, an index must be provided. The value will be repeated to match the length of index.

  Source : https://www.tutorialspoint.com/python_pandas_series.htm

➢ **Mathematical operations**

We can perform mathematical operation on series like addition, subtraction, multiplication, division etc.

For this, various methods are used, as follows :

- **add() :** This function is used to add series and others, element wise.

  Syntax Series.add (other, fill–value=Name, axis=0)

  Here Other is series or scalar value

  fill-value is None or float value but its default value is None
- **sub() :** This function is used to get subtraction of series and others, element wise.

  Syntax Series.sub (other, fill value=None, axis=0)
- **mul() :** This function is used to get multiplication of series and others, element-wise.

  Syntax Series.mul (other, fill value=None, axis=0)
- **div() :** This function is used to get floating division of series and others, element-wise.

  Syntex Series.div (other, fill value=Name, axis=0)
- **pow() :** This function is used to get exponential power of series and others, element-wise.

  Syntax Seires.pow (other, fill value=None, axis=0)

e.g.

```
import numpy as np
import pandas as pd
x = pd.Series([2, 1, 2, np.nan], index=['p', 'q', 'r', 's'])
y = pd.Series([1, np.nan, 2, 1], index=['p', 'q', 's', 't'])
print("---Addition---")
print(x.add(y, fill_value=0))
print("---Subtraction---")
print(x.sub(y, fill_value=0))
print("---Multiplication---")
print(x.mul(y, fill_value=0))
print("---Divsion---")
print(x.div(y, fill_value=0))
print("---Power---")
print(x.pow(y, fill_value=0))
```

Output :

	Addition	Subtraction	Multiplication	Division	Power
p	3.0	1.0	2.0	2.0	2.0
q	1.0	1.0	0.0	inf	1.0
r	2.0	2.0	0.0	inf	1.0

s	2.0	–2.0	0.0	0.0	0.0
t	1.0	1.0	0.0	0.0	0.0

dtype: float 64

- **Head and Tail functions**
  - head() function is used to get the first *n* rows.

    Syntax Series.head(n=5)

    Here, *n* is the selected number of rows. It is int type and has default value 5.
  - tail() function returns last *n* rows from the object based on position. It is useful for quickly verifying data. for example, after sorting

    Syntax Series.tail(*n* = 5)

    Here, *n* is the selected number of rows whose default value is 5.
- **Selection**

  In series, Series.select() function is used for selection. This function returns data corresponding to axis labels matching criteria. We pass the name of the function as an argument to this function which is applied on all the index tables. The index labels satisfying the criteria are selected.

  Syntax Series.select (crit, axis=0)

  Here,

  crit = called on each label

  axis = int value
- **Indexing**

  The object supports both integers and label based indexing and provides a host of methods for performing operation involving the index.

  In Python Pandas, Series.index attribute is used to get or set the index labels of the given series object.

  Syntax Series.index

  Pandas supports three types of multi-axes indexing, which are as follows :

  (i) .loc[] : This attribute is used to access a group of rows and columns by label(s) or a boolean array in the given series object.

  Syntax Series.loc

  (ii) .iloc[] : This attribute enables purely integer location based indexing for selection by position over the given series object

  (iii) .ix[] : This attribute is primarily label location based indexer, with integer position fallback. It takes the label as input and returns the value corresponding to that label.

  Syntax Series.ix
- **Slicing**

  Slicing is a powerful approach to retrieve subsets of data from a Pandas object. A slice Object is built using a syntax of start : end : step, the segments representing the first item, last item and the increment between each item that you would like as the step.

Know the Terms

- **Pandas :** The Pandas is a high-performance open source library for data analysis in Python.
- **Matplotlib :** It is a visualization library in Python that used for 2D plots of arrays.
- **Series :** It is a one-dimensional array containing a sequence of values. Each value has a data label associated with it also called its index.

- **Selection :** This function returns data corresponding to axis labels matching criteria.
- **Indexing :** This function is used to get or set the index labels of the given series object.
- **Slicing :** Slicing is a powerful approach to retrieve subsets of data from a Pandas object.

Data Frames & Operation on Rows and Columns

- **Data Frames**
  - Data Frame is a two dimensional data structure *i.e.*, data is aligned in a tabular form as rows and columns.

    Data frame consists of various properties as iteration, indexing etc.

    Scan to know more about this topic

    Python Pandas Data frame Basics
  - In data frame, columns can be heterogeneous types like integer, boolean etc.
  - It can be seen as a dictionary of series where rows and columns both are indexed.

    Data can be created using following syntax :

    pandas, DataFrame (data, index, columns, dtype, copy)

    Here **data** contains different forms like ndarray, series, map, constants etc.

    index is used for the row label

    columns is used for column label

    dtype refers to the data type of each column

    copy used for copying data

- **Create DataFrame**

  We can create DataFrame using various inputs which are discussed below
  - **Creating an empty DataFrame :** It is basic DataFrame that can be created by

    import pandas as pd

    object_Name = pd.DataFrame()
  - **Creating a DataFrame From dict of series :** Dictionary of series can be passed to form a DataFrame. The resultant index is the union of all the series indexes passed.

    e.g.

```
import pandas as pd
data={'First' : pd.Series (['abc', 'xyz', 'pqr'], index = [11, 12, 13]),
      'second' : pd.Series (['The', 'That', 'This', 'abc'],
                  index=[11, 12, 13, 14])}
value=pd.DataFrame(data)
print(value)
```

    Output

	First	Second
11	abc	The
12	xyz	That
13	pqr	This
14	NaN	abc

  - **Creating a DataFrame From list of dictionary :** List of dictionary can be passed to form a DataFrame. Keys of dictionary are taken as column names by default.

    e.g.

```
import pandas as pd
data = [{'abc' : 10, 'xyz' : 20, 'pqr' : 30},
        {'The' : 10, 'pqr' : 20, 'xyz' : 30, 'abc' : 40}]
```

```
value = pd.DataFrame (data)
print(value)
```

Output

	The	abc	pqr	xyz
0	NaN	10	30	20
1	10.0	40	20	30

➢ **Iterating in Pandas DataFrame**

Iteration is a general term for taking each item of something one after another.

In Pandas DataFrame, we can iterate an element in two ways :

(i) Iterating over rows : There are three function to iterate over rows as follows :

- **iterrows() :** It returns the iterator yielding each index value along with a series containing the data in each row.
- **iteritems() :** It iterates over each column as key, value pair with label as key and column value as series object.
- **itertuples() :** In DataFrame, it returns a tuple for each row. The first element of the tuple will be the row's corresponding index value, while the remaining value are the rows values.

(ii) Iterating over columns

In order to iterate over columns, we need to create a list of dataframe columns and then iterating through that list to pull out the dataframe columns.

➢ **Operations on rows and columns**

As we known, DataFrame is a two dimensional data structure means data is arranged in a tabular format like rows and columns, some basic operations can be perform like adding, deleting, selecting and renaming. These operations are as follows :

(i) Addition

- To add a column in Pandas Dataframe, a new list as a column can be declared and add to an existing DataFrame.
- To add a row in Pandas DataFrame, we can concat the old dataframe with new one.

(ii) Selection

- To select a column in Pandas DataFrame, we can either access the columns by calling them by their column names.
- To retrieve rows from a DataFrame, a special method is used named DataFrame.loc[]. Rows can also be selected by passing integer location to iloc[] method.

(iii) Deletion

- To delete a column from Pandas DataFrame, drop() method is used. Columns are deleted by dropping columns with column names.
- To delete a row from Pandas DataFrame, drop() method is used. Rows are deleted by dropping rows by index label.

➢ **Head and Tail functions**

head() and tail() methods or functions are used to view a small sample of a DataFrame object. These functions are described below

(i) head() : This function returns the first *n* rows for the object based on position. It is useful for quick testing if your object has the right type of data in it.

Syntax DataFrame.head (*n*=5)

Parameters : n-is an integer value, number of rows to be returned where default value is 5. Return DataFrame with top *n* rows

(ii) tail() : This function returns last *n* rows from the object based on position. It is useful for quickly verifying data. *e.g.* after sorting

Syntax : DataFrame.tail (*n*=5)

- **Indexing using Labels**

  Indexing in Pandas means simply selecting particular rows and columns of a DataFrame. Indexing can also be known as subset selection.

  It is common operation to pick out one of the DataFrame's columns to work on. To select a column by its label, we use the .loc[] function.

  Pandas DataFrame.loc attribute access a group of rows and columns by label(s) or a boolean array in the given DataFrame.

  Syntax : DataFrame.loc

  loc takes two single/list/range operator separated by ','. The first one indicates the row and the second one indicates columns.

- **Boolean Indexing :**

  It helps us to select the data from the DataFrames using a boolean vector. We need a DataFrame with a boolean index to use the boolean indexing.

  In boolean indexing, we can filter a data in four ways

  - Accessing a DataFrame with a boolean index
  - Applying a boolean mask to a dataFrame
  - Masking data based on column value
  - Masking data based on index value

  e.g., import pandas as pd

  ```
  dict={'Name' : ["Rahul", "Kiyaan", "Shreya", "Riya"],
        "Salary" : ["28000", "38000", "34000", "3600"]}
  info=pd.DataFrame(dict, index=[True, False, False, True])
  print (info)
  ```

  Output

	Name	Salary
True	Rahul	28000
False	Kiyaan	38000
False	Shreya	34000
True	Riya	36000

- **CSV File**

  CSV files are the comma separated values. This type of file can be view as an excel file and separated by commas. CSV file is nothing more than a simple text file. However, it is the most common, simple and easiest method to store tabular data. This particular format arranges tables by a specific structure divided into rows and columns.

  Once we have the DataFrame, we can persist it in CSV on the local disk. Let's first create CSV file using data that is currently present in the DataFrame, we can store the data of this DataFrame in CSV format using API called to_CSV (…) of Pandas

- **Importing/Exporting Data between CSV files and DataFrames**

  - Pandas **read_csv()** function is used to import a CSV file to DataFrame format.

    Syntax df.read_csv('file_name.CSV', header=None)

    Here,

    Header allows you to specify which row will be used as column names for your DataFrame. Expected int value or a list of int values. If your file does not have a header, then simply set header=None

  - To export a Pandas DataFrame to a CSV file, use **to_csv** function. This saves a DataFrame as a CSV file.

    Syntax to_csv(parameters)

STAND ALONE MCQs (1 Mark each)

Q. 1. Pandas is an open-source _____ Library?

(A) Ruby **(B)** Javascript

(C) Java **(D)** Python

Ans. Option (D) is correct.

> ***Explanation :*** Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures.

AI **Q. 2.** Pandas key data structure is called?

(A) Keyframe **(B)** DataFrame

(C) Statistics **(D)** Econometrics

Ans. Option (B) is correct.

> ***Explanation :*** Pandas is built on the Numpy package and its key data structure is called the DataFrame.

Q. 3. In pandas, Index values must be?

(A) unique

(B) hashable

(C) Both A and B

(D) None of the above

Ans. Option (C) is correct.

> ***Explanation :*** Index values must be unique and hashable, same length as data. Default np.arrange(n) if no index is passed.

Q. 4. In data science, which of the python library are more popular ?

(A) Numpy

(B) Pandas

(C) OpenCv

(D) Django

Ans. Option (B) is correct.

> ***Explanation :*** Pandas is an essential package for Data Science in Python because it's versatile and really good at handling data.

AI **Q. 5.** We can analyze the data in pandas with :

(A) Series

(B) DataFrame

(C) Both of the above

(D) None of the above

Ans. Option (C) is correct.

> ***Explanation :*** Pandas is the most popular python library that is used for data analysis. It provides highly optimized performance with back-end source code is purely written in C or Python.
> We can analyze data in pandas with :
> Series
> DataFrames

Q. 6. Which of the following thing can be data in Pandas?

(A) a python dict

(B) anndarray

(C) a scalar value

(D) all of the mentioned

Ans. Option (D) is correct.

> ***Explanation :*** The passed index is a list of axis labels.

Q. 7. Which of the following input can be accepted by Data Frame?

(A) Structured ndarray

(B) Series

(C) DataFrame

(D) All of the mentioned

Ans. Option (D) is correct.

> ***Explanation :*** DataFrame is a 2-dimensional labeled data structure with columns of potentially different types.

Q. 8. Which of the following indexing capabilities is used as a concise means of selecting data from a pandas object?

(A) In **(B)** ix

(C) ipy **(D)** iy

Ans. Option (B) is correct.

> ***Explanation :*** ix and reindex are 100% equivalent.

AI **Q. 9.** Given a Pandas series called sequences, the command which will display the first 4 rows is _____.

(A) print (sequences.head(4))

(B) print (sequences.Head(4))

(C) print (sequences.heads(4))

(D) print (sequences.Heads(4))

(CBSE SQP 2020-21)

Ans. Option (A) is correct.

> ***Explanation :*** A sequence is a group of items with a deterministic ordering.
> Pandas head() method is used to return top n (5 by default) rows of a data frame or series

Q. 10. Which of the following input can be accepted by DataFrame?

(A) Structured ndarray

(B) Series

(C) DataFrame

(D) All of the mentioned

Ans. Option (D) is correct.

> ***Explanation :*** DataFrame is a 2-dimensional labeled data structure with columns of potentially different types.

AI **Q. 11.** Which object do you get after reading a CSV file using pandas.read_csv()?

(A) Dataframe

(B) Nd array

(C) Char Vector

(D) None

Ans. Option (D) is correct.

> ***Explanation :*** Pandas read_csv() method is used to read CSV file into DataFrame object. The CSV file is like a two-dimensional table where the values are separated using a delimiter.

Q. 12. Which of the following indexing capabilities is used as a concise means of selecting data from a pandas object?

(A) In **(B)** ix

(C) ipy **(D)** iy

Ans. Option (B) is correct.

> ***Explanation :*** ix and reindex are 100% equivalent.

Q. 13. Which of the following statement(s) is/are correct with respect to df.columns properties to rename columns

1. All columns must be specified
2. Columns must be in the form of a list
3. Old column names not required
4. Columns can be specified with columns number

(A) Only 1 is correct

(B) 1, 2 and 3 are correct

(C) 1 and 3 are correct

(D) All of them are correct

Ans. Option (B) is correct.

> ***Explanation :*** The columns can also be renamed by directly assigning a list containing the new names to the columns attribute of the dataframe object for which we want to rename the columns.

Q. 14. df.index properties can be used to

(A) rename rows

(B) rename columns

(C) rename rows and columns both

(D) None of these

Ans. Option (A) is correct.

> ***Explanation :*** The index property returns an object of type Index.

Q. 15. To display 2 rows from the top in the dataframe, which of the following statement is correct:

(A) df.head()=2

(B) df.head(n=2)

(C) df.head(range(2))

(D) All of the above

Ans. Option (B) is correct.

> ***Explanation :*** In Python's Pandas module, the Dataframe class provides a head() function to fetch top rows from a Dataframe

Q. 16. What will be syntax for pandasdataframe?

(A) pandas.DataFrame(data, index, dtype, copy)

(B) pandas.DataFrame(data, index, rows, dtype, copy)

(C) pandas_DataFrame(data, index, columns, dtype, copy)

(D) pandas.DataFrame(data, index, columns, dtype, copy)

Ans. Option (D) is correct.

> ***Explanation :*** Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns).

AI **Q. 17.** The command used to skip a row in a CSV file is

(A) next() **(B)** skip()

(C) omit() **(D)** bounce()

Ans. Option (A) is correct.

> ***Explanation :*** Use csv. reader() and next() to skip the first line of a . csv file.

Q. 18. What is the output of the following program?

```
import csv
```

d=csv.reader(open('c:\PYPRG\ch13\ city.csv'))

next**(D)**

for row in d:

print(row)

if the file called "city.csv" contain the following details

chennai,mylapore

mumbai,andheri

(A) chennai,mylapore

(B) mumbai,andheri

(C) chennaimumbai

(D) chennai,mylapore mumbai, Andheri

Ans. Option (B) is correct.

> ***Explanation :*** csvreader is an iterable object. Hence, .next() method returns the current row and advances the iterator to the next row.

Q. 19. A CSV file is also known as a ….

(A) Flat File

(B) 3D File

(C) String File

(D) Random File

Ans. Option (A) is correct.

> ***Explanation :*** A CSV is a comma-separated values file, which allows data to be saved in a tabular format.

Q. 20. Which of the following module is provided by Python to do several operations on the CSV files?

(A) py **(B)** xls

(C) csv **(D)** os

Ans. Option (C) is correct.

> ***Explanation :*** Python provides a module named csv, using this we can do several operations on the csv files.

AI **Q. 21.** In a data frame axis -0 is for

(A) Columns

(B) Rows

(C) Rows and Columns both

(D) None of these

Ans. Option (B) is correct.

> ***Explanation :*** A DataFrame object has two axes: "axis 0" and "axis 1". "axis 0" represents rows and "axis 1" represents columns.

Q. 22. Which function from the options given below can read the dataset from a large text file?

(A) read_json **(B)** read_pickle

(C) read_hdf **(D)** read_csv

Ans. Option (D) is correct.

> ***Explanation :*** The Pandas read_csv() function returns a new DataFrame with the data and labels from the file data.csv, which you specified with the first argument.

Q. 23. Which function needs a dictionary of array like sequences or a dictionary of another dictionary, to return a DataFrame?

(A) DataFrame.from_items

(B) DataFrame.from_records

(C) DataFrame.from_dict

(D) All of the above

Ans. Option (A) is correct.

> ***Explanation :*** DataFrame.from_dict operates like the DataFrame constructor except for the orient parameter which is 'columns' by default.

AI **Q. 24.** Data structures in Pandas can be mutated in the terms of ____ but not of _____.

(A) size, value **(B)** value, size

(C) semantic, size **(D)** none of the above

Ans. Option (B) is correct.

> ***Explanation :*** The length of a Series cannot be changed.

Q. 25. Minimum number of arguments we require to pass in pandas series :

(A) 0 **(B)** 1

(C) 2 **(D)** 3

Ans. Option (C) is correct.

> ***Explanation :*** Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.).

ASSERTION AND REASON BASED MCQs (1 Mark each)

Directions : In the following questions a statement of Assertion (A) is followed by a statement of reason (R). Mark the correct choice as :

(A) Both Assertion (A) and reason (R) are true and reason (R) is the correct explanation of Assertion (A).

(B) Both Assertion (A) and reason (R) are true but reason (R) is not the correct explanation of Assertion (A).

(C) Assertion (A) is true but reason (R) is false.

(D) Assertion **(A)** is false but reason (R) is true.

Q.1. Assertion (A) : To create a series from array, we have to import the numpymodule and then use array () method.

Reason (R) : NumPy is an open-source numerical Python library. NumPy contains a multi-dimensional array and matrix data structures.

Ans. Option (A) is correct.

> ***Explanation :*** NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. NumPy is a Python package. For most data types, pandas uses NumPy arrays as the concrete objects contained with a Index , Series , or DataFrame.

Q.2. Assertion (A) : You need to import or load the Pandas library first in order to use it.

Reason (R) : The "pd" is an alias or abbreviation which will be used as a shortcut to access or call pandas functions.

Ans. Option (B) is correct.

> ***Explanation :*** By "Importing a library", it means loading it into the memory and then you can use it. Run the following code to import pandas library : import pandas as pd

Q.3. Assertion (A) : In Python Pandas, Series.index attribute is used to get or set the index labels of the given series object.

Reason (R) : ix[] attribute is used to access a group of rows and columns by label (s) or a boolean array in the given series object.

Ans. Option (C) is correct.

> ***Explanation :*** .loc[] attribute is used to access a group of rows and columns by label (s) or a boolean array in the given series object.
> .ix[] attribute is primarily label location based indexer, with integer position fallback. It takes the label as input and returns the value corresponding to that label.

Q.4. Assertion (A) : In series, Series.selection() function is used for selection.

Reason (R) : tail() function returns last n rows from the object based on position. It is useful for quickly verifying data.

Ans. Option (D) is correct.

> ***Explanation :*** In series, Series.select() function is used for selection. This function returns data corresponding to axis labels matching criteria. We pass the name of the function as an argument to this function which is applied on all the index tables.
> Pandas tail() method is used to return bottom n (5 by default) rows of a data frame or series.
> Syntax: Dataframe.tail(n=5).

Q.5. Assertion (A) : Matplotlib is a visualization library in Python that used for 2D plots of arrays.

Reason (R) : Indexing function is used to get or set the index labels of the given series object.

Ans. Option (B) is correct.

> ***Explanation :*** Matplotlibis a multi-platform datavisualization library which build NumPy arrays. Matplotlib produces publication-quality figures in a variety ofhardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, thePython and IPython shell, web application servers and various graphical user interface toolkits.
>
> In Python Pandas, Series.index attribute is used to get or set the index labels of the given series object.

Q.6. Assertion (A) : DataFrame is a two dimensional labelled array. It's columns types can be heterogeneous i.e., of varying types.

Reason (R) : We need a DataFrame witha boolean index to use the boolean indexing.

Ans. Option (B) is correct.

> ***Explanation :*** Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labelled axes (rows and columns).
> Boolean indexing is a type of indexing which uses actual values of the data in the DataFrame.

Q. 7. Assertion (A) : CSV files are available to open in any spreadsheet program, including Google Sheets, Open Office, and Microsoft Excel.

Reason (R) : Using a spreadsheet program can serve a user's needs better since it has cells where data is sorted in rows and columns.

Ans. Option (A) is correct.

Explanation : A CSV file stores data, both numbers and text in a plain text. All fields are separated by commas while all records are separated by an elaborate line of characters. A spreadsheet program sorts the data in a CSV file systematically via columns. This helps to filter all the contents in the file.

Q. 8. Assertion (A) : Iteration is a general term for taking each item of something one after another.

Reason (R) : itertuples()the iterator yielding each index value along with a series containing the data in each row.

Ans. Option (C) is correct.

Explanation : Iteration is the repetition of a process in order to generate an outcome. The sequence will approach some end point or end value. Each repetition of the process is a single iteration, and the outcome of each iteration is then the starting point of the next iteration.
iterrows() returns the iterator yielding each index value along with a series containing the data in each row.

Q. 9. Assertion (A) : Indexing can also be known as sub selection.

Reason (R) : Pandas DataFrame.loc attribute access a group of rows and columns by label(s) or a boolean array in he given DataFrame.

Ans. Option (D) is correct.

Explanation : Indexing can also be known as subset selection.
loc takes two single/list/range operator separated by ','. The first one indicates the row and the second one indicates columns.

CASE-BASED MCQs

Attempt any four sub-parts from each question. Each sub-part carries 1 mark.

I. Consider the following DataFrame df and answer any four questions from 1-5

Roll no.	Name	UT1	UT2	UT3	UT4
1	Prerna Singh	24	24	20	22
2	Manish Arora	18	17	19	22
3	Tanish Goel	20	22	18	24
4	Falguni Jain	22	20	24	20

Q. 1. Write down the command that will give the following output. 1

```
Roll no.      6
Name      Tanish Goel
UT1           24
UT2           24
UT3           24
UT4           24
dtype : object
```

(A) print (df.max)

(B) print df.max( )

(C) print (df.max (axis=1))

(D) print (df.max, axis=1)

Ans. Option (A) is correct.

Explanation : max() function returns the maximum of the values in the given object.

Q. 2. The teacher needs to know the marks scored by the student with roll number 4. Help her to identify the correct set of statements from the given option :

(A) df1=df[rollno==4] print (df1)

(B) df1=df[rollno==4] print (df1)

(C) df1=df [df.rollno=4] print (df1)

(D) df1=df [df.rollno==4] print (df1)

Ans. Option (A) is correct.

Explanation : When we apply ==, >, <, <=, >= operator on dataframe then it produce a Series of True and False.

Q. 3. Which of the following statements will give the exact number of values in each column of the dataframe?

(i) print (df.count ())

(ii) print (df.count (0))

(iii) print (df.count)

(iv) print (df.count (axis'index')

Choose the correct option :

(A) both (i) and (ii)

(B) only(ii)

(C) (i), (ii) and (iii)

(D) (i), (ii) and (iv)

Ans. Option (A) is correct.

Explanation : Pandas dataframe.count() is used to count the no. of non-NA/null observations across the given axis. It works with non-floating type data as well.

Q. 4. Which of the following command will display the column of the DataFrame?

(A) print (df.column ())

(B) print (df.columns ())

(C) print (df.column)

(D) print (df.columns)

Ans. Option (A) is correct.

Explanation : Pandas DataFrame. columns attribute return the column labels of the given Dataframe.

Q. 5. Ms. Sharma the class teacher wants to add a new column, the scores of Grade with the values, 'A','B','A','A','B','A', to the DataFrame. Help her choose the command to do so :

(A) df.column=['A','B','A','A','B','A']

(B) df['Grade']=['A','B','A','A','B','A']

(C) df.loc['Grade']=['A','B','A','A','B','A']

(D) Both **(B)** and **(C)** are correct

[CBSE SQP 2020-21]

Ans. Option (B) is correct.

Explanation : Programmers can use to add a column to DataFrame is by generating a new list as a separate column of data and appending the column to the existing DataFrame.

II. Consider the following DataFrame df and answer any four questions from 1-5

ID	Name	Age	Fav_Color	Points
T01	Rahul Anand	32	Blue	73
T02	Mohak Girdhar	25	Green	82
T03	Rajeev Tyagi	45	Orange	29
T04	Rohini Malik	30	Pink	39

Q. 1. Write down the command that will add a column "eligible" with default values as 'yes'.

(A) df('eligible')='yes'

(B) df['eligible']='yes'

(C) df('eligible')='yes'

(D) df.Insert['eligible'='yes']

Ans. Option (B) is correct.

Explanation : Use the syntax pd. Dataframe[new_column] = value to add a column named new_column with each element as value to pd.

Q. 2. Write the command to extract the complete row 'T03'.

(A) df.loc[:, 'ID']

(B) df.loc['T03', 'Name']

(C) df.loc['T02', 'T03']

(D) df.loc['T03',:]

Ans. Option (D) is correct.

Explanation : DataFrame.loc[] method is a method that takes only index labels and returns row or dataframe if the index label exists in the caller data frame.

Q. 3. For the above DataFrame, following statement is given error :
df[Age]=df[Points]*2/3
Find and correct the error.
(A) df['Age']=df['Points']*2/3
(B) df[Age]=df['Points']*2/3
(C) df['Age']=df[Points]*2/3
(D) df['Age']=df['Points'*2/3]

Ans. Option (A) is correct.

Q. 4. Write the statement to list the first three entries of the DataFrame 'df'.

(A) df.head()

(B) df.head(3)

(C) df.head('3')

(D) All of the above

Ans. Option (B) is correct.

Explanation : head(n) to get the first n rows of the DataFrame. It takes one optional argument n (number of rows you want to get from the start). By default n = 5, it return first 5 rows if value of n is not passed to the method.

Q. 5. Which command will be used to drop a row from dataframe 'df' labelled as 'T04' ?

(A) Df.drop() **(B)** df.drop()

(C) df.drop('T04') **(D)** df.drop(T04)

Ans. Option (C) is correct.

III. Consider the following dataframe df as shown below :

	Name	Eng.	IP	Geo	Total
T1	Kushagra	52	98	85	235
T2	Naresh	48	85	88	221
T3	Prakhar	69	94	78	241
T4	Trapti	70	81	91	242

Q. 1. Which command will give the output 20 :

(A) print(df.size) **(B)** print(df.shape)

(C) print(df.index) **(D)** print(df.axes)

Ans. Option (A) is correct.

> ***Explanation :*** Returns size of dataframe/ series which is equivalent to total number of elements.

Q. 2. What will be the output produced by following statements ?

>>>print(df.at['T3', 'total'], df.at['T1', 'ip'])

(A) 235 **94**

(B) 241 **98**

(C) 241 **94**

(D) 235 **98**

Ans. Option (B) is correct.

> ***Explanation :*** Pandas at[] is used to return data in a dataframe at the passed location.

Q. 3. What will be the output produced by following statements ?

>>>print(dff.loc['T2' : 'T3' , 'ip':'geo'])

(A)

	IP	Geo
T2	**85**	**88**
T3	**94**	**78**

(B)

	IP
T2	**85**

(C)

	IP	Geo
T2	**85**	**88**

(D)

	IP
T2	**85**
T3	**94**

Ans. Option (A) is correct.

> ***Explanation :*** DataFrame.loc[] method is a method that takes only index labels and returns row or dataframe if the index label exists in the caller data frame.

Q. 4. What will be the output produced by following statements ?

>>> print(df.at[2,1],df.at[1,2])

(A) Prakhar **69**

(B) T2 **Naresh**

(C) Khushagra **52**

(D) 69 **85**

Ans. Option (D) is correct.

> ***Explanation :*** Pandas at[] is used to return data in a dataframe at the passed location.

Q. 5. What will be the output produced by following statements?

>>> print(df.iloc[::2,0::4])

(A)

	Name	Total
T1	**Kushagra**	**235**
T2	**Naresh**	**221**

(B)

	Name	Total
T2	**Naresh**	**221**
T4	**Trapti**	**242**

(C)

	Name	Total
T1	**Kushagra**	**235**
T3	**Prakhar**	**241**

(D)

	Name	Total
T3	**Prakhar**	**241**
T4	**Trapti**	**242**

Ans. Option (C) is correct.

> ***Explanation :*** Dataframe.iloc[] method is used when the index label of a data frame is something other than numeric series of 0, 1, 2, 3….n or in case the user doesn't know the index label. Rows can be extracted using an imaginary index position which isn't visible in the data frame.

IV. Consider the following DataFrame emp and answer any four question from 1-5

Emp-no	Name	Dept	Salary	Experience (in years)
1	Ram singh	IT	15000	2.5
2	Shyam singh	HR	18000	3
3	Nidhi gupta	IT	9000	2
4	Puja sharma	EXE	24000	8
5	Rohan Malik	HR	20000	6

Q. 1. Write down the command that will given the following output.

Empno 5

Name Shyam singh

Dept HR

Salary 24000

Experience 8

dtype : object

(A) print(emp.max)

(B) print(emp.max( ))

(C) print(emp.max(axis=1))

(D) print(emp.max.axis=1)

Ans. Option (B) is correct.

> ***Explanation :*** max() function returns the maximum of the values in the given object.

Q. 2. The CEO needs to know the salary of the employee with empno 4. Help him to identify the correct set of statement/s from the given option :

(A) emp1=emp[emp['empno']==4]
print(emp1)

(B) emp1=emp[emp]
print(emp1)

(C) emp1=emp[emp.empno=4]
print(emp1)

(D) emp1=emp[emp.empno==4]
print(emp1)

Ans. Option (A,D) is correct.

Q. 3. Which of the following statement/s will give the exact of values in each column of the dataframe ?

i. print(emp.count())

ii. print(emp.count(0))

iii. print(emp.count)

iv. print(emp.count(axis='index'))

Choose the correct option :

(A) both i and ii

(B) only ii

(C) i, ii and iii

(D) i, ii and iv

Ans. Option (A) is correct.

> ***Explanation :*** Pandas dataframe.count() is used to count the no. of non-NA/null observations across the given axis. It works with non-floating type data as well.

Q. 4. Which of the following command will display the column labels of the DataFrame?

(A) print(emp.columns())

(B) print(emp.column())

(C) print(emp.column)

(D) print(emp.columns)

Ans. Option (A) is correct.

> ***Explanation :*** Pandas DataFrame.columns attribute return the column labels of the given Dataframe.

Q. 5. Mr. Satvik Ahuja, the CEO wants to add a new column, the rating of the performance of employees with the values, 'A', 'A', 'B', 'A', 'B', to the DataFrame. Help him choose the command to do so.

(A) emp.column=['A', 'A', 'B', 'A', 'B']

(B) emp['Performance']=['A', 'A', 'B', 'A', 'B']

(C) emp.loc['Performance']= ['A', 'A', 'B', 'A', 'B']

(D) Both **(B)** and **(C)** are correct.

Ans. Option (B) is correct.

> ***Explanation :*** Programmers can use to add a column to DataFrame is by generating a new list as a separate column of data and appending the column to the existing DataFrame.

❐❐❐