table of contents

|  |  |  |
| --- | --- | --- |
| S.no | Contents | Page No. |
| 1. | **Introduction to Pandas:-**   * 1. About pandas   2. Benefits of pandas   3. Installation of pandas   4. Data structures in pandas |  |
| 2. | **Creating objects :-** |  |
| 3. | **Viewing data :-** |  |
| 4. | **Selection:-** |  |
| 5. | Manipulating Data |  |
| 6. | Grouping Data |  |
| 7. | Merging, joining , concatenating |  |
| 8. | Working with date and time |  |
| 9. | Working with text data |  |
| 10. | Working with Csv and excel |  |
| 11. | Introduction Creating objects Viewing Data operations |  |
| 12. | visualization |  |
| 13. | Applications and projects |  |
| 14. | Miscellaneous |  |
|  |  |  |

Introduction to pandas

* Pandas is an open source library.
* It is made mainly for working with relational or labelled data.
* It is fast and efficient.
* Used for reshaping and pivoting of the data sets.
* Provide the functionality of Time Series.
* it integrates with the other libraries such as SciPy, and scikit-learn.
* Handle multiple operations of the data sets such as subsetting, slicing, filtering, groupBy, re-ordering, and re-shaping.

# Benefits of pandas :-

1. Data representation:-

it represents the data in the best form that it is more suited for data analysis

1. clear code :-

it allows you to directly focus on the core part of the code.

# Installation of pandas:-

1. Type cmd command in search box of your pc.
2. And after that using cd in command prompt load the file where you have kept python-pip file.
3. After this write **pip install pandas.**
4. After this write **import pandas as pd.**

# Data structures in pandas:-

There are two types of data structures in pandas :-

1. **Series:-**
2. [Pandas Series](https://www.geeksforgeeks.org/python-pandas-series/) is a one-dimensional labelled array capable of holding data of any type (integer, string, float, python objects, etc.).
3. The axis labels are collectively called indexes.
4. We can easily convert the list, tuple, and dictionary into series using "series' method.
5. A Series cannot contain multiple columns. It has one parameter
6. The object supports both integer and label-based indexing and provides a host of methods for performing operations involving the index.
7. Pandas Series can be created from the lists, dictionary, and from a scalar value etc.
8. Data can be any list , dictionary or scalar value.

Creating a series using arrays:-

Code:-

import pandas as pd

import numpy as np

Arr=np.array(['m' , 'u' , 's' , 'k' , 'a' ,'n'])

Ser=pd.Series(Arr)

print(Ser)

output:-

0    m

1    u

2    s

3    k

4    a

5    n

dtype: object

1. **DataFrame:-**

* A Data frame is a two-dimensional data structure, i.e., data is aligned in rows and columns.
* Pandas DataFrame consists of three principal components, the

data, rows, and columns.

**Code:-**

#Data Frame in pandas:-

import pandas as pd

import numpy as np

Arr=np.array(['m' , 'u' , 's' , 'k' , 'a' ,'n'])

df=pd.DataFrame()

print(df)

df=pd.DataFrame(Arr)

print(df)

**Output:-**

Empty DataFrame

Columns: []

Index: []

   0

0  m

1  u

2  s

3  k

4  a

5  n

### **Q.** Why Pandas is used for Data Science ?

* Pandas allows us to analyze big data and make conclusions based on statistical theories.
* Pandas can clean messy data sets, and make them readable and relevant.
* Relevant data is very important in data science.

imp parameteres used in pandas:-

* **For importing the pandas file:-**

**Syntax:-**  import pandas as pd

* **Load the CSV file:-**

**Syntax:-**  df=pd.read\_csv(‘my data.csv’)

**Syntax:-**  df=pd.read\_csv(‘data/my data.csv’)

(If your csv file is saved in different directory then this syntax will be used)

* **Explore the data :-**

**1. Syntax:-** df.head()

(It will read the fiirst five lines of your data )

**2. Syntax:-** df.head(10)

(It will read the first 10 lines of your data )

**3. Syntax:-** df.describe()

(This command displays the count, mean, standard deviation, minimum, and maximum values for each column of the DataFrame. If your DataFrame contains non-numeric columns, the describe() function will skip them)

* **Manipulate the data :-**

**1. Syntax:-** df.loc[0:5, [“column1”,”column2”]]

(It will print the row from 0 to 5 of column 1 and 2. It used to slice in the data )

**2. Syntax:-** df.iloc[0:5, [0,1]]

(It will print the 0th and 1th column from rows 0 to 1. It only takes the numeric values.)

* **Visualizing the data :-**

**1. Syntax:-** df.[“column1”].plot()

(It will make a line plot of your given column.)

**2. Syntax:-** df.plot.scatter(x=’marks’ , y=’total’)

(It will create the scatter plot between two columns.)

series in pandas

# Creating a series in pandas:-

* + In the real world, a Pandas Series will be created by loading the datasets from existing storage, storage can be SQL Database, CSV file, and Excel file
  + Pandas Series can be created from the lists, dictionary, and from a scalar value etc
  + It has the following parameter:

**Data :**  It can be any list, dictionary, or scalar value.

**Index :**  The value of the index should be unique and hashable. It must be of the same length as data. If we do not pass any index, default **np.arrange(n)** will be used.

**Dtype :**  It refers to the data type of series.

**Copy :**  It is used for copying the data.

* We can create series in two ways :-
  + Empty series
  + series using inputs.

# Empty Series:-

**Code:-**

import pandas as pd

import numpy as np

#creating an empty series:-

seer=pd.Series()

print(seer)

Output:-

Series([], dtype: object)

# Series using inputs:-

1. using arrays,
2. using dictionary
3. using scalar value

## **Using arrays:-**

* If we want to create series using Array then we need to add module Numpy.
* If we do not pass an index, then by default index of **range(n)** is being passed where n defines the length of an array, i.e., [0,1,2,....**range(len(array))-1**].

Code in vs file

## **Using dictionary :-**

* **If the dictionary object is being passed as an input and the index is not specified, then the dictionary keys are taken in a sorted order to construct the index**
* If index is passed, then values correspond to a particular label in the index will be extracted from the **dictionary**.

Code in vs file

## **Using Scalar value:-**

* If we take the scalar values, then the index must be provided. The scalar value will be repeated for matching the length of the index.

Code in vs file

# Acessing elements of series:-

There are two ways through which we can access element of series, they are :

* Accessing Element from Series with Position
* Accessing Element Using Label (index)

## **Accessing elements from series with position:-**

* In order to access the series element refers to the index number.
* Use the index operator [ ] to access an element in a series.
* The index must be an integer. In order to access multiple elements from a series, we use Slice operation.

## **Accessing Element Using Label (index):-**

## In order to access an element from series, we have to set values by index label. A Series is like a fixed-size dictionary in that you can get and set values by index label.

Code in vs file

# Series object attributes:-

The Series attribute is defined as any information related to the Series object such as size, datatype. etc.

|  |  |
| --- | --- |
| Attributes | Description |
|  |  |
| Series.index | Defines the index of the series. |
| Series.shape | It returns a tuple of shape of the data. |
| Series.dtype | It returns the data type of the data. |
| Series.size | It returns the size of the data. |
| Series.empty | It returns True if Series object is empty, otherwise returns false. |
| Series.hasnans | It returns True if there are any NaN values, otherwise returns false. |
| Series.nbytes | It returns the number of bytes in the data. |
| Series.ndim | It returns the number of dimensions in the data. |
| Series.itemsize | It returns the size of the datatype of item. |

Code in vs filef

# Series functions:-

There are some functions used in Series which are as follows:-

|  |  |
| --- | --- |
| Functions | Description |
| Series.map() | Map the values from the two series that have a common column. |
| Series.std() | Used to calculate the standard deviation. |
| Series.to\_frame() | Used to convert the series object to dataframe |
| Series.unique() | Used to print the unique elemts from an column |
| Series.value\_counts() | It returns a series that contains counts of unique values. |

# Series.map() function:-

* The main task of map() is used to map the values from two series that have a common column.
* To map the two Series, the last column of the first Series should be the same as the index column of the second series, and the values should be unique.

**Syntax:-** Series.map(arg, na\_action=None)

# Series.std() function:-

* The Pandas **std()** is defined as a function for calculating the standard deviation of the given set of numbers, DataFrame, column, and rows

**Syntax:-**

Series.std(axis=None, skipna=None, level=None, ddof=1, numeric\_only=None, \*\*kwargs)

# Series.to\_frame() function:-

* The main difference between Series and Data Frame is that Series can only contain a single list with a particular index, whereas the DataFrame is a combination of more than one series that can analyzethe data.
* The Pandas **Series.to\_frame()** function is used to convert the series object to the DataFrame

**Syntax:-** Series.to\_frame(name=None)

# Series.value\_counts() function:-

* The value\_counts() function returns a Series that contain counts of unique values.
* It returns an object that will be in descending order so that its first element will be the most frequently-occurred element.

**Syntax:-** Series.value\_counts(normalize=False, sort=True, ascending=False, bins=None, dropna=True)

dataframe in pandas

# Creating a dataframe in pandas:-

* Pandas DataFrame is a widely used data structure which works with a two-dimensional array with labeled axes (rows and columns).
* DataFrame is defined as a standard way to store data that has two different indexes, i.e., **row index** and **column index.**
* The columns can be heterogeneous types like int, bool, and so on.

It consists of following terms :-

* **data:** It consists of different forms like ndarray, series, map, constants, lists, array.
* **index:** The Default np.arrange(n) index is used for the row labels if no index is passed.
* **columns:** The default syntax is np.arrange(n) for the column labels. It shows only true if no index is passed.
* **dtype:** It refers to the data type of each column.
* **copy():** It is used for copying the data.

We can create a DataFrame using following ways:

* **dict**
* **Lists**
* **Numpy ndarrrays**
* **Series**

Codes in vs filef

# operation apllied on dataframe:-