Pandas

Pandas

Pandas is an open-source Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.
It is mainly popular for importing and analyzing data much easier.

Key Features of Pandas

- Fast and efficient DataFrame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Label-based slicing, indexing and subsetting of large data sets.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.

Installing Pandas

```
!pip install pandas
```

Importing Pandas

```
import pandas as pd
```

Checking Pandas Version

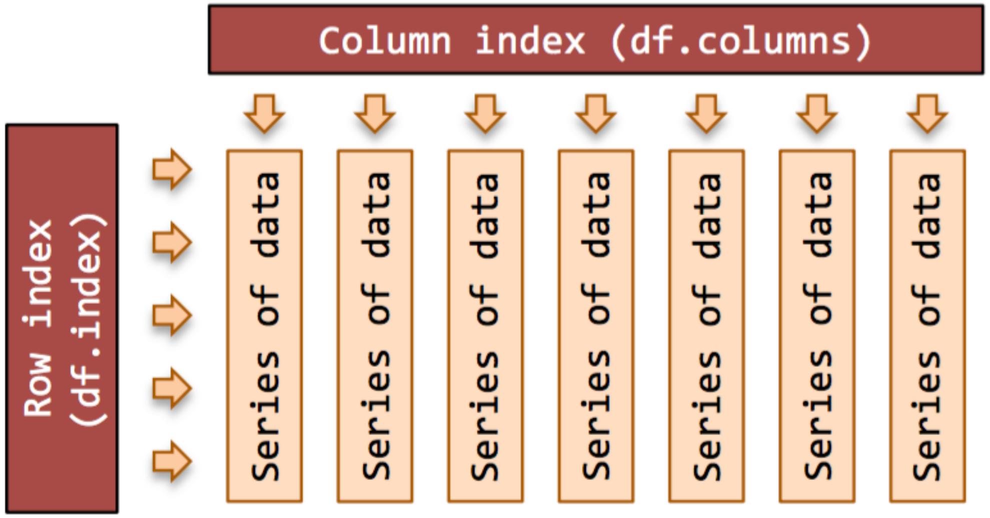
```
pd.__version__
```

Pandas Series

Series is a one-dimensional labelled array capable of holding data of any type (integer, string, float, python objects, etc.).
The axis labels are collectively called index.

Syntax
pd.Series(data, index, dtype)

A series can be created using various inputs like –
•Array
•Dict
•Scalar value or constant



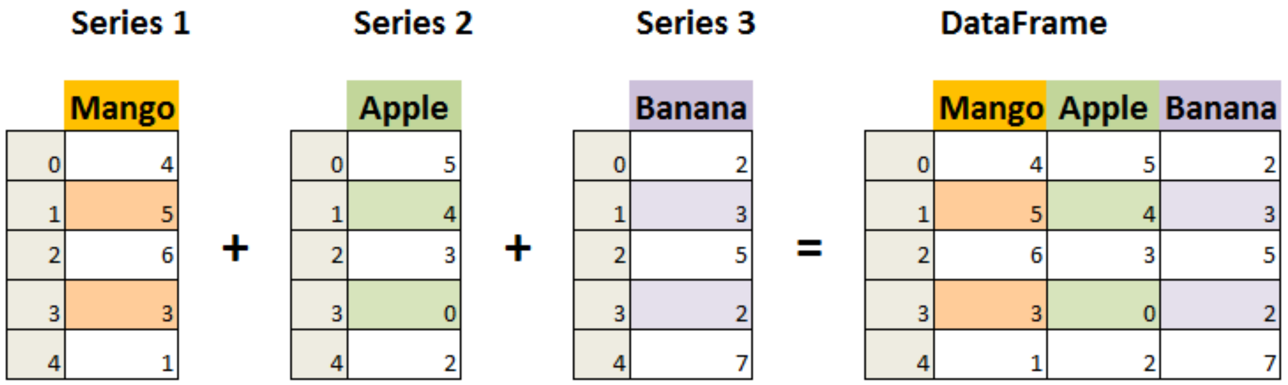
Pandas DataFrame

A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.

Syntax

pd. DataFrame(data, index, columns, dtype, copy)

A pandas DataFrame can be created using various inputs like –
•Lists
•dict
•Series
•Numpy ndarrays



Series 1

	Mango
0	4
1	5
2	6
3	3
4	1

+

Series 2

	Apple
0	5
1	4
2	3
3	0
4	2

+

Series 3

	Banana
0	2
1	3
2	5
3	2
4	7

=

DataFrame

	Mango	Apple	Banana
0	4	5	2
1	5	4	3
2	6	3	5
3	3	0	2
4	1	2	7

Reading CSV files using Pandas

Most of the data for analysis is available in the form of a tabular format such as Excel and Comma Separated files(CSV).
To access data from csv file, we require a function read_csv() that retrieves data in the form of data frame.

Syntax

pd. read_csv(filepath*, sep, usecols, nrows, index_col)

filepath : URL or location of file

sep : Stands for separator, default is ‘, ‘ as in csv

usecols : Only uses the passed col[string list] to make data frame

nrows : It means number of rows to be displayed from the dataset.

index_col : Makes passed column as index instead of 0, 1, 2, 3…,n

Indexing DataFrames

function	Description	Syntax
shape	returns the number of rows and columns	df.shape
head	Return the first n rows. (Default first 5 rows)	df.head(n)
tail	Return the last n rows. (Default last 5 rows)	df.tail()
loc	Label based indexing	df.loc[row range, [column names]]
iloc	Index based indexing	df.iloc[row range , column range]

Other useful Functions

Finding missing values

df.isna().sum() : To check missing values of all the columns

df[columnnames].isna().sum() : To check missing values of specified columns

Print columns of DataFrame

df.columns

Sorting DataFrame

df.sort_values(by = [columns],ascending=[order])

Crosstab and Group By

Crosstab

A cross tabulation (or crosstab) report is used to analyze the relationship between two or more variables.
This method is used to compute a simple cross-tabulation of two (or more) factors.

Syntax

```
pd.crosstab(df[column1] , df[column2])
```

Group By

In many situations, we split the data into sets and we apply some functionality on each subset. In the apply functionality, we can perform the following operations −

- Aggregation − computing a summary statistic
- Transformation − perform some group-specific operation
- Filtration − discarding the data with some condition

Syntax

```
df.groupby([columns]).agg({col1:aggfunction, col2:aggfunction, …})
```

Pivot tables

Pivot Tables

A PivotTable is an interactive way to quickly summarize large amounts of data.
You can use a PivotTable to analyze numerical data in detail, and answer unanticipated questions about your data.
In pandas we have a function pivot_table() which helps us to create pivot table in python.

Syntax

```
pd.pivot_table(data, values, index, columns, aggfunc)
```

data : DataFrame
values : column to aggregate, optional
index: column, Grouper, array, or list of the previous
columns: column, Grouper, array, or list of the previous
aggfunc: function, list of functions, dict, default numpy.mean

Concat and Joins

Concat

In order to concat DataFrame, we use concat() function which helps in concatenating a DataFrame.

Syntax

pd.concat([df1, df2] ,axis)

axis : Whether we want to concat DataFrames row wise or column wise (1 for columns,0 for rows, default is 0)

Joins

Pandas has full-featured, high performance in-memory join operations very similar to relational databases like SQL. Pandas provides a single function, merge, as the entry point for all standard database join operations between DataFrame objects.

Syntax

pd.merge(leftdf, rightdf, how, on, left_on, right_on,suffixes)

leftdf – A DataFrame object.

rightdf – Another DataFrame object.

how – One of 'left', 'right', 'outer', 'inner'. Defaults to inner. Each method has been described below.

on – Columns (names) to join on. Must be found in both the left and right DataFrame objects.

left_on – Columns from the left DataFrame to use as keys. Can either be column names or arrays with length equal to the length of the DataFrame.

right_on – Columns from the right DataFrame to use as keys. Can either be column names or arrays with length equal to the length of the DataFrame.

suffixes – List indicating the suffix to add in case of overlapping columns

Other useful functions

pd.read_clipboard(sep)
Read text from clipboard
sep : A string or regex delimiter. The default of ‘s+’ denotes one or more whitespace characters.

df.to_csv(path, index)
Write dataframe to a comma-separated values (csv) file.
path : File path
Index : Whether to write row names (**True** : If we want row names else **False**, Default is True)

pd.read_excel(path, sheet_name)
Read an Excel file into a pandas DataFrame.
path : File path
Sheet_name : Sheet of excel file which we want to read.

df.to_excel(path, index, sheet_name)
Write object to an Excel sheet.
path : File path
Index : Whether to write row names (**True** : If we want row names else **False**, Default is True)
sheet_name : Sheet of excel file which should store dataframe.