**Aim**: Create Two-dimensional data with the help of data frames and perform different operations on it.

**Theory:**

**Pandas Data Frame** is a two-dimensional size-mutable, potentially heterogeneous tabular data structure with labelled 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 Data Frame consists of three principal components, the **data**, **rows**, and **columns**.

**Code:**

import pandas as pd

dict1 = {'one' : pd.Series([1., 2., 3.],index=['a', 'b', 'c']),

         'two' : pd.Series([1., 2., 3., 4.],

                           index=['a', 'b', 'c', 'd'])}

df = pd.DataFrame(dict1)

df

#Add column

df['three'] = df['one'] \* df['two']

df['flag'] = df['one'] > 2

df

#Adding a Column Using a Scalar and Assigning to a Data Frame

df['Filler'] = 'HCT'

df['Slic'] = df['one'][:2]

df

# Delete columns

del df['two']

#alternative code

Three = df.pop('three')

df

#insert value on index 1

df.insert(1, 'bar', df['one'])

df

#missing value

df.isna()

#summary

df.describe()

**Output :**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **red** | **green** | **yellow** |
| **a** | -2.097778 | 0.048924 | -2.871512 |
| **b** | -1.094421 | 0.034183 | -0.218595 |
| **c** | 0.137730 | -1.416531 | 0.313404 |

# colum selection

a    0.048924

b    0.034183

c   -1.416531

Name: green, dtype: float64

# multiple colume selection

|  |  |  |
| --- | --- | --- |
|  | **red** | **yellow** |
| **a** | -2.097778 | -2.871512 |
| **b** | -1.094421 | -0.218595 |
| **c** | 0.137730 | 0.313404 |

# insert

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **red** |  | **green** | **color** | **yellow** |
| **a** | -2.097778 |  | 0.048924 | Any | -2.871512 |
| **b** | -1.094421 |  | 0.034183 | Any | -0.218595 |
| **c** | 0.137730 |  | -1.416531 | Any | 0.313404 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **red** | **green** | **color** | **yellow** |
| **b** | -1.094421 | 0.034183 | Any | -0.218595 |
| **c** | 0.137730 | -1.416531 | Any | 0.313404 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **red** | **green** | **color** | **yellow** |
| **a** | -2.097778 | 0.048924 | Any | -2.871512 |
| **b** | -1.094421 | 0.034183 | Any | -0.218595 |
| **c** | 0.137730 | -1.416531 | Any | 0.313404 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **red** | **green** | **yellow** |
| **a** | -2.097778 | 0.048924 | -2.871512 |
| **b** | -1.094421 | 0.034183 | -0.218595 |
| **c** | 0.137730 | -1.416531 | 0.313404 |

# greater than 0

|  |  |  |  |
| --- | --- | --- | --- |
|  | **red** | **green** | **yellow** |
| **a** | False | True | False |
| **b** | False | True | False |
| **c** | True | False | True |

# condition on only one column, will return boolean

a     True

b     True

c    False

Name: green, dtype: bool

**Result:**

We can select any row and column of the DataFrame by passing the name of the rows and column. When you select it from the DataFrame, it becomes one-dimensional and considered as Series. Here we have performed various operations like addition, subtraction, multiplication and finding null values in it..