In [2]:
```
import numpy as np
import pandas as pd
```

1)Create any series and print the output

In [3]:
```
a=pd.Series([1,2,3,4])
a
```

Out[3]:
```
0    1
1    2
2    3
3    4
dtype: int64
```

2)Create any dataframe of 10x5 with few nan values and print the output

In [4]:
```
b=pd.DataFrame(np.random.randn(10,5))
b
```

Out[4]:

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **0** | -1.716845 | -1.682953 | 0.710344 | 0.065341 | 2.189372 |
| **1** | -0.751919 | 0.334650 | -1.004745 | -0.780546 | -0.899033 |
| **2** | -0.951698 | 0.216146 | -0.596814 | 2.386887 | 1.941154 |
| **3** | -0.575552 | 0.086148 | -0.052183 | -0.857394 | -0.692277 |
| **4** | 0.662474 | 0.802218 | -0.720891 | -0.094154 | -0.091665 |
| **5** | 0.220488 | -2.241827 | 1.091333 | -0.101199 | 0.977486 |
| **6** | 0.139861 | -0.557000 | -1.168093 | 0.090117 | -1.396452 |
| **7** | -0.161203 | -1.631497 | -0.587880 | -1.535733 | -0.452674 |
| **8** | -1.764292 | -0.364935 | 0.180476 | 0.490930 | -0.253134 |
| **9** | -1.014942 | 0.426999 | 0.912173 | -0.029895 | -1.216904 |

3)Display top 7 and last 6 rows and print the output

In [5]:
```
b.head(7)
```

Out[5]:

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **0** | -1.716845 | -1.682953 | 0.710344 | 0.065341 | 2.189372 |
| **1** | -0.751919 | 0.334650 | -1.004745 | -0.780546 | -0.899033 |
| **2** | -0.951698 | 0.216146 | -0.596814 | 2.386887 | 1.941154 |
| **3** | -0.575552 | 0.086148 | -0.052183 | -0.857394 | -0.692277 |
| **4** | 0.662474 | 0.802218 | -0.720891 | -0.094154 | -0.091665 |
| **5** | 0.220488 | -2.241827 | 1.091333 | -0.101199 | 0.977486 |

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **6** | 0.139861 | -0.557000 | -1.168093 | 0.090117 | -1.396452 |

In [6]:
```
b.tail(6)
```

Out[6]:

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **4** | 0.662474 | 0.802218 | -0.720891 | -0.094154 | -0.091665 |
| **5** | 0.220488 | -2.241827 | 1.091333 | -0.101199 | 0.977486 |
| **6** | 0.139861 | -0.557000 | -1.168093 | 0.090117 | -1.396452 |
| **7** | -0.161203 | -1.631497 | -0.587880 | -1.535733 | -0.452674 |
| **8** | -1.764292 | -0.364935 | 0.180476 | 0.490930 | -0.253134 |
| **9** | -1.014942 | 0.426999 | 0.912173 | -0.029895 | -1.216904 |

4)Fill with a constant value and print the output

In [7]:
```
c=pd.DataFrame(
{
    'a':[1,2,3,np.nan,4],
    'b':[11,12,np.nan,13,44],
})
c.fillna(value=0)
```

Out[7]:

| | a | b |
|---|---|---|
| **0** | 1.0 | 11.0 |
| **1** | 2.0 | 12.0 |
| **2** | 3.0 | 0.0 |
| **3** | 0.0 | 13.0 |
| **4** | 4.0 | 44.0 |

5) Drop the column with missing values and print the output

In [8]:
```
d=pd.DataFrame(
{
    'a':[1,2,3,np.nan,4],
    'b':[11,12,np.nan,13,44],
})
d.dropna()
```

Out[8]:

| | a | b |
|---|---|---|
| **0** | 1.0 | 11.0 |
| **1** | 2.0 | 12.0 |
| **4** | 4.0 | 44.0 |

6)Drop the row with missing value and print the output

In [9]:

```
d=pd.DataFrame(
{
    'a':[1,2,3,np.nan,4],
    'b':[11,12,np.nan,13,44],
})
d.dropna()
```

Out[9]:

| | a | b |
|---|---|---|
| **0** | 1.0 | 11.0 |
| **1** | 2.0 | 12.0 |
| **4** | 4.0 | 44.0 |

7)To check the presence of missing values in your dataframe

In [10]:

```
d=pd.DataFrame(
{
    'a':[1,2,3,np.nan,4],
    'b':[11,12,np.nan,13,44],
})
d.isna()
```

Out[10]:

| | a | b |
|---|---|---|
| **0** | False | False |
| **1** | False | False |
| **2** | False | True |
| **3** | True | False |
| **4** | False | False |

8)Use operators and check the condition and print the output

In [11]:

```
d=pd.DataFrame(
{
    'a':[1,2,3,np.nan,4],
    'b':[11,12,np.nan,13,44],
})
d[d['a']>3]
```

Out[11]:

| | a | b |
|---|---|---|
| **4** | 4.0 | 44.0 |

9)Display your output using loc and iloc,rows and columns headings

In [12]:

```
d=pd.DataFrame(
{
    'a':[1,2,3,np.nan,4],
    'b':[11,12,np.nan,13,44],
})
d.loc[0:1]
```

Out[12]:

| | a | b |
|---|---|---|
| **0** | 1.0 | 11.0 |
| **1** | 2.0 | 12.0 |

10)Display the statistical summary of data

In [13]:

```
b.describe()
```

Out[13]:

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **count** | 10.000000 | 10.000000 | 10.000000 | 10.000000 | 10.000000 |
| **mean** | -0.591363 | -0.461205 | -0.123628 | -0.036565 | 0.010587 |
| **std** | 0.811631 | 1.045001 | 0.816874 | 1.036024 | 1.269804 |
| **min** | -1.764292 | -2.241827 | -1.168093 | -1.535733 | -1.396452 |
| **25%** | -0.999131 | -1.362872 | -0.689872 | -0.610709 | -0.847344 |
| **50%** | -0.663736 | -0.139394 | -0.320032 | -0.062024 | -0.352904 |
| **75%** | 0.064595 | 0.305024 | 0.577877 | 0.083923 | 0.710198 |
| **max** | 0.662474 | 0.802218 | 1.091333 | 2.386887 | 2.189372 |