21-07-2023

Importing libraries

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

1.Create any series and print the output

In [3]:
```
a=pd.Series([2,3,4,5,6,7,8])
a
```

Out[3]:
```
0    2
1    3
2    4
3    5
4    6
5    7
6    8
dtype: int64
```

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

In [6]:
```
data = [{'a': 5, 'b': 10,'c':20,'d':7,'e':9}]
df = pd.DataFrame(data, index=['A','B','C','D','E','F','G','H','I','J'],columns=['a', '
df
```

Out[6]:

	a	b1	c	d1	e
A	5	NaN	20	NaN	9
B	5	NaN	20	NaN	9
C	5	NaN	20	NaN	9
D	5	NaN	20	NaN	9
E	5	NaN	20	NaN	9
F	5	NaN	20	NaN	9
G	5	NaN	20	NaN	9
H	5	NaN	20	NaN	9
I	5	NaN	20	NaN	9
J	5	NaN	20	NaN	9

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

In [7]:
```
df.head(7)
```

Out[7]:

	a	b1	c	d1	e
A	5	NaN	20	NaN	9
B	5	NaN	20	NaN	9
C	5	NaN	20	NaN	9
D	5	NaN	20	NaN	9
E	5	NaN	20	NaN	9
F	5	NaN	20	NaN	9
G	5	NaN	20	NaN	9

In [8]:
```
df.tail(6)
```

Out[8]:

	a	b1	c	d1	e
E	5	NaN	20	NaN	9
F	5	NaN	20	NaN	9
G	5	NaN	20	NaN	9
H	5	NaN	20	NaN	9
I	5	NaN	20	NaN	9
J	5	NaN	20	NaN	9

4.Fill with constant value and print the output

In [9]:
```
df.fillna(value=8)
```

Out[9]:

	a	b1	c	d1	e
A	5	8.0	20	8.0	9
B	5	8.0	20	8.0	9
C	5	8.0	20	8.0	9
D	5	8.0	20	8.0	9
E	5	8.0	20	8.0	9
F	5	8.0	20	8.0	9

	a	b1	c	d1	e
G	5	8.0	20	8.0	9
H	5	8.0	20	8.0	9
I	5	8.0	20	8.0	9
J	5	8.0	20	8.0	9

5.Drop the row with missing values and print the output

In [13]:
```
df = pd.DataFrame({'A':  [1,2,3,4,5,np.nan,6,7,np.nan,np.nan,8,9,10,np.nan],
        'B': [11,12,np.nan,13,14,np.nan,15,16,np.nan,np.nan,17,np.nan,19,np.nan],
        'C': [20,21,22,23,np.nan,24,np.nan,26,27,np.nan,np.nan,28,29,30]
       })
df
```

Out[13]:

	A	B	C
0	1.0	11.0	20.0
1	2.0	12.0	21.0
2	3.0	NaN	22.0
3	4.0	13.0	23.0
4	5.0	14.0	NaN
5	NaN	NaN	24.0
6	6.0	15.0	NaN
7	7.0	16.0	26.0
8	NaN	NaN	27.0
9	NaN	NaN	NaN
10	8.0	17.0	NaN
11	9.0	NaN	28.0
12	10.0	19.0	29.0
13	NaN	NaN	30.0

In [15]:
```
df.dropna()
```

Out[15]:

	A	B	C
0	1.0	11.0	20.0
1	2.0	12.0	21.0
3	4.0	13.0	23.0

	A	B	C
7	7.0	16.0	26.0
12	10.0	19.0	29.0

6.Drop the column with missing values and print the output

In [16]:
```
data = [{'a': 5, 'b': 10,'c':20,'d':7,'e':9}]
df = pd.DataFrame(data, index=['A','B','C','D','E','F','G','H','I','J'],columns=['a', '
df
```

Out[16]:

	a	b1	c	d1	e
A	5	NaN	20	NaN	9
B	5	NaN	20	NaN	9
C	5	NaN	20	NaN	9
D	5	NaN	20	NaN	9
E	5	NaN	20	NaN	9
F	5	NaN	20	NaN	9
G	5	NaN	20	NaN	9
H	5	NaN	20	NaN	9
I	5	NaN	20	NaN	9
J	5	NaN	20	NaN	9

In [17]:
```
df.dropna(axis=1,how='any')
```

Out[17]:

	a	c	e
A	5	20	9
B	5	20	9
C	5	20	9
D	5	20	9
E	5	20	9
F	5	20	9
G	5	20	9
H	5	20	9
I	5	20	9
J	5	20	9

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

In [18]:
```
data = [{'a': 5, 'b': 10,'c':20,'d':7,'e':9}]
df = pd.DataFrame(data, index=['A','B','C','D','E','F','G','H','I','J'],columns=['a', '
df
```

Out[18]:

	a	b1	c	d1	e
A	5	NaN	20	NaN	9
B	5	NaN	20	NaN	9
C	5	NaN	20	NaN	9
D	5	NaN	20	NaN	9
E	5	NaN	20	NaN	9
F	5	NaN	20	NaN	9
G	5	NaN	20	NaN	9
H	5	NaN	20	NaN	9
I	5	NaN	20	NaN	9
J	5	NaN	20	NaN	9

In [19]:
```
pd.isna(df)
```

Out[19]:

	a	b1	c	d1	e
A	False	True	False	True	False
B	False	True	False	True	False
C	False	True	False	True	False
D	False	True	False	True	False
E	False	True	False	True	False
F	False	True	False	True	False
G	False	True	False	True	False
H	False	True	False	True	False
I	False	True	False	True	False
J	False	True	False	True	False

In [20]:
```
np.isnan(df)
```

Out[20]:

	a	b1	c	d1	e
A	False	True	False	True	False
B	False	True	False	True	False
C	False	True	False	True	False
D	False	True	False	True	False
E	False	True	False	True	False
F	False	True	False	True	False
G	False	True	False	True	False
H	False	True	False	True	False
I	False	True	False	True	False
J	False	True	False	True	False

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

In [22]:

```
data = [{'a': 5, 'b': 10,'c':20,'d':7,'e':9}]
df = pd.DataFrame(data, index=['A','B','C','D','E','F','G','H','I','J'],columns=['a', '
df
```

Out[22]:

	a	b	c	d	e
A	5	10	20	7	9
B	5	10	20	7	9
C	5	10	20	7	9
D	5	10	20	7	9
E	5	10	20	7	9
F	5	10	20	7	9
G	5	10	20	7	9
H	5	10	20	7	9
I	5	10	20	7	9
J	5	10	20	7	9

In [27]:

```
df[df["b"]>=10]
```

Out[27]:

	a	b	c	d	e
A	5	10	20	7	9
B	5	10	20	7	9

	a	b	c	d	e
C	5	10	20	7	9
D	5	10	20	7	9
E	5	10	20	7	9
F	5	10	20	7	9
G	5	10	20	7	9
H	5	10	20	7	9
I	5	10	20	7	9
J	5	10	20	7	9

In [29]:
```
df[df["a"]<2]
```

Out[29]:

	a	b	c	d	e

In [36]:
```
df[df["a"]%5==0]
```

Out[36]:

	a	b	c	d	e
A	5	10	20	7	9
B	5	10	20	7	9
C	5	10	20	7	9
D	5	10	20	7	9
E	5	10	20	7	9
F	5	10	20	7	9
G	5	10	20	7	9
H	5	10	20	7	9
I	5	10	20	7	9
J	5	10	20	7	9

9.Display your output using loc and iloc ,row,column heading

In [52]:
```
data = [{'a': 'B', 'b': 10,'c':20,'d':7,'e':9}]
df = pd.DataFrame(data, index=['A','B','C','D','E','F','G','H','I','J'],columns=['a', '
df
```

Out[52]:

	a	b	c	d	e
A	B	10	20	7	9

	a	b	c	d	e
B	B	10	20	7	9
C	B	10	20	7	9
D	B	10	20	7	9
E	B	10	20	7	9
F	B	10	20	7	9
G	B	10	20	7	9
H	B	10	20	7	9
I	B	10	20	7	9
J	B	10	20	7	9

In [53]:
```
df.iloc[0:4]
```

Out[53]:

	a	b	c	d	e
A	B	10	20	7	9
B	B	10	20	7	9
C	B	10	20	7	9
D	B	10	20	7	9

In [42]:
```
df.loc["A":"D"]
```

Out[42]:

	a	b	c	d	e
A	5	10	20	7	9
B	5	10	20	7	9
C	5	10	20	7	9
D	5	10	20	7	9

In [44]:
```
df.columns
```

Out[44]:
```
Index(['a', 'b', 'c', 'd', 'e'], dtype='object')
```

In [46]:
```
df.index
```

Out[46]:
```
Index(['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J'], dtype='object')
```

10.Display the statistical summary of data

In [47]:
```
df.describe()
```

Out[47]:

	a	b	c	d	e
count	10.0	10.0	10.0	10.0	10.0
mean	5.0	10.0	20.0	7.0	9.0
std	0.0	0.0	0.0	0.0	0.0
min	5.0	10.0	20.0	7.0	9.0
25%	5.0	10.0	20.0	7.0	9.0
50%	5.0	10.0	20.0	7.0	9.0
75%	5.0	10.0	20.0	7.0	9.0
max	5.0	10.0	20.0	7.0	9.0

In [ ]:
```

```