

1. Create any Series and print the output

```
In [1]: import numpy as np
import pandas as pd
```

```
In [20]: df=pd.Series([1,2,3,4,5,6])
df
```

```
Out[20]: 0    1
1    2
2    3
3    4
4    5
5    6
dtype: int64
```

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

```
In [43]: df=pd.DataFrame({"A": [1,2,3,4,np.nan,2,3,np.nan,5,6],
                           "B": [2,3,1,5,6,2,3,np.nan,5,6],
                           "C": [2,3,np.nan,5,6,2,3,np.nan,5,6],
                           "D": [5,6,7,np.nan,12,2,3,np.nan,5,6],
                           "E": [1,2,3,7,10,12,4,np.nan,34,5]})
df
```

```
Out[43]:
```

	A	B	C	D	E
0	1.0	2.0	2.0	5.0	1.0
1	2.0	3.0	3.0	6.0	2.0
2	3.0	1.0	NaN	7.0	3.0
3	4.0	5.0	5.0	NaN	7.0
4	NaN	6.0	6.0	12.0	10.0
5	2.0	2.0	2.0	2.0	12.0
6	3.0	3.0	3.0	3.0	4.0
7	NaN	NaN	NaN	NaN	NaN
8	5.0	5.0	5.0	5.0	34.0
9	6.0	6.0	6.0	6.0	5.0

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

```
In [34]: df.head(7)
```

Out[34]:

	A	B	C	D	E
0	1.0	2.0	2.0	5.0	1.0
1	2.0	3.0	3.0	6.0	2.0
2	3.0	NaN	NaN	7.0	3.0
3	4.0	5.0	5.0	NaN	7.0
4	NaN	6.0	6.0	12.0	10.0
5	2.0	2.0	2.0	2.0	12.0
6	3.0	3.0	3.0	3.0	4.0

In [35]:

df.tail(6)

Out[35]:

	A	B	C	D	E
4	NaN	6.0	6.0	12.0	10.0
5	2.0	2.0	2.0	2.0	12.0
6	3.0	3.0	3.0	3.0	4.0
7	NaN	NaN	NaN	NaN	NaN
8	5.0	5.0	5.0	5.0	34.0
9	6.0	6.0	6.0	6.0	5.0

4. Fill with a constant value and print the output

In [39]:

df1=pd.DataFrame({"A":1,"B":pd.Series(10,index=list(range(10)))})
df1

Out[39]:

	A	B
0	1	10
1	1	10
2	1	10
3	1	10
4	1	10
5	1	10
6	1	10
7	1	10
8	1	10
9	1	10

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

In [44]: `df.dropna(axis=0)`

Out[44]:

	A	B	C	D	E
0	1.0	2.0	2.0	5.0	1.0
1	2.0	3.0	3.0	6.0	2.0
5	2.0	2.0	2.0	2.0	12.0
6	3.0	3.0	3.0	3.0	4.0
8	5.0	5.0	5.0	5.0	34.0
9	6.0	6.0	6.0	6.0	5.0

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

In [45]: `df.dropna(0)`

Out[45]:

	A	B	C	D	E
0	1.0	2.0	2.0	5.0	1.0
1	2.0	3.0	3.0	6.0	2.0
5	2.0	2.0	2.0	2.0	12.0
6	3.0	3.0	3.0	3.0	4.0
8	5.0	5.0	5.0	5.0	34.0
9	6.0	6.0	6.0	6.0	5.0

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

In [47]: `df.isnull`

Out[47]:

	A	B	C	D	E
0	1.0	2.0	2.0	5.0	1.0
1	2.0	3.0	3.0	6.0	2.0
2	3.0	1.0	NaN	7.0	3.0
3	4.0	5.0	5.0	NaN	7.0
4	NaN	6.0	6.0	12.0	10.0
5	2.0	2.0	2.0	2.0	12.0
6	3.0	3.0	3.0	3.0	4.0
7	NaN	NaN	NaN	NaN	NaN
8	5.0	5.0	5.0	5.0	34.0
9	6.0	6.0	6.0	6.0	5.0

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

In [48]: `df[df["A"]>=1]`

Out[48]:

	A	B	C	D	E
0	1.0	2.0	2.0	5.0	1.0
1	2.0	3.0	3.0	6.0	2.0
2	3.0	1.0	NaN	7.0	3.0
3	4.0	5.0	5.0	NaN	7.0
5	2.0	2.0	2.0	2.0	12.0
6	3.0	3.0	3.0	3.0	4.0
8	5.0	5.0	5.0	5.0	34.0
9	6.0	6.0	6.0	6.0	5.0

In [51]: `df[df["C"]<=6]`

Out[51]:

	A	B	C	D	E
0	1.0	2.0	2.0	5.0	1.0
1	2.0	3.0	3.0	6.0	2.0
3	4.0	5.0	5.0	NaN	7.0
4	NaN	6.0	6.0	12.0	10.0
5	2.0	2.0	2.0	2.0	12.0
6	3.0	3.0	3.0	3.0	4.0
8	5.0	5.0	5.0	5.0	34.0
9	6.0	6.0	6.0	6.0	5.0

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

In [58]:

```
df=pd.DataFrame({"A":[1,2,3,"bcd",np.nan,2,3,np.nan,5,6],
                  "B":[2,3,1,5,6,2,3,np.nan,5,6],
                  "C":[2,3,np.nan,5,6,2,3,np.nan,5,6],
                  "D":[5,6,7,np.nan,12,2,3,np.nan,5,6],
                  "E":[1,2,3,7,10,12,4,np.nan,34,5]})
df.loc["A":"C"]
```

Out[58]:

	A	B	C	D	E
--	---	---	---	---	---

In [62]: `df.iloc[1:6]`

Out[62]:

	A	B	C	D	E
1	2	3.0	3.0	6.0	2.0
2	3	1.0	NaN	7.0	3.0
3	bcd	5.0	5.0	NaN	7.0
4	NaN	6.0	6.0	12.0	10.0
5	2	2.0	2.0	2.0	12.0

In [64]:

```
df.columns
```

Out[64]: Index(['A', 'B', 'C', 'D', 'E'], dtype='object')

In [65]:

```
df.index
```

Out[65]: RangeIndex(start=0, stop=10, step=1)

10. Display the statistical summary of data

In [66]:

```
df.describe()
```

Out[66]:

	B	C	D	E
count	9.000000	8.000000	8.000000	9.000000
mean	3.666667	4.000000	5.750000	8.666667
std	1.870829	1.690309	3.011881	10.173495
min	1.000000	2.000000	2.000000	1.000000
25%	2.000000	2.750000	4.500000	3.000000
50%	3.000000	4.000000	5.500000	5.000000
75%	5.000000	5.250000	6.250000	10.000000
max	6.000000	6.000000	12.000000	34.000000

In []: