Daily Assessment

In [1]:

```
import numpy as np
import pandas as pd
```

1. Create any Series and print the output

In [3]:

```
a=pd.Series([45,12,78,46,13,24])
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 [5]:

```
b=pd.DataFrame(
{
    'A':np.arange(1,11),
    'B':['sara','man','tea','karma','tamil','bot','sale','leave','mention','flash'],
    'C':[11,23,12,45,23,67,12,45,67,34],
    'D':pd.date_range('20230721',periods=10),
    'E':pd.Series(np.nan,index=list(range(10))),
})
b
```

Out[5]:

	A	B	C	D	E
0	1	sara	11	2023-07-21	NaN
1	2	man	23	2023-07-22	NaN
2	3	tea	12	2023-07-23	NaN
3	4	karma	45	2023-07-24	NaN
4	5	tamil	23	2023-07-25	NaN
5	6	bot	67	2023-07-26	NaN
6	7	sale	12	2023-07-27	NaN
7	8	leave	45	2023-07-28	NaN
8	9	mention	67	2023-07-29	NaN
9	10	flash	34	2023-07-30	NaN

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

In [8]:

```
b.head(7)
```

Out[8]:

	A	B	C	D	E
0	1	sara	11	2023-07-21	NaN
1	2	man	23	2023-07-22	NaN
2	3	tea	12	2023-07-23	NaN
3	4	karma	45	2023-07-24	NaN
4	5	tamil	23	2023-07-25	NaN
5	6	bot	67	2023-07-26	NaN
6	7	sale	12	2023-07-27	NaN

In [10]:

```
b.tail(6)
```

Out[10]:

	A	B	C	D	E
4	5	tamil	23	2023-07-25	NaN
5	6	bot	67	2023-07-26	NaN
6	7	sale	12	2023-07-27	NaN
7	8	leave	45	2023-07-28	NaN
8	9	mention	67	2023-07-29	NaN
9	10	flash	34	2023-07-30	NaN

4. Fill with a constant value and print the output

In [13]:

```
b.fillna(value=2)
```

Out[13]:

	A	B	C	D	E
0	1	sara	11	2023-07-21	2.0
1	2	man	23	2023-07-22	2.0
2	3	tea	12	2023-07-23	2.0
3	4	karma	45	2023-07-24	2.0
4	5	tamil	23	2023-07-25	2.0
5	6	bot	67	2023-07-26	2.0
6	7	sale	12	2023-07-27	2.0
7	8	leave	45	2023-07-28	2.0
8	9	mention	67	2023-07-29	2.0
9	10	flash	34	2023-07-30	2.0

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

In [14]:

```
b.dropna(axis=1,how='any')
```

Out[14]:

	A	B	C	D
0	1	sara	11	2023-07-21
1	2	man	23	2023-07-22
2	3	tea	12	2023-07-23
3	4	karma	45	2023-07-24
4	5	tamil	23	2023-07-25
5	6	bot	67	2023-07-26
6	7	sale	12	2023-07-27
7	8	leave	45	2023-07-28
8	9	mention	67	2023-07-29
9	10	flash	34	2023-07-30

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

In [16]:

```
b.dropna()
```

Out[16]:

	A	B	C	D	E

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

In [18]:

```
b.isna()
```

Out[18]:

	A	B	C	D	E
0	False	False	False	False	True
1	False	False	False	False	True
2	False	False	False	False	True
3	False	False	False	False	True
4	False	False	False	False	True
5	False	False	False	False	True
6	False	False	False	False	True
7	False	False	False	False	True
8	False	False	False	False	True
9	False	False	False	False	True

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

In [34]:

```
c=pd.Series([12,34,56,65,78,73])
print(c[(c>30)&(c<70)])
```

```
1    34
2    56
3    65
dtype: int64
```

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

In [19]:

```
b.iloc[0:3]
```

Out[19]:

	A	B	C	D	E
0	1	sara	11	2023-07-21	NaN
1	2	man	23	2023-07-22	NaN
2	3	tea	12	2023-07-23	NaN

In [20]:

```
b.loc['A':'C']
```

Out[20]:

	A	B	C	D	E

In [21]:

```
b.loc[3:7]
```

Out[21]:

	A	B	C	D	E
3	4	karma	45	2023-07-24	NaN
4	5	tamil	23	2023-07-25	NaN
5	6	bot	67	2023-07-26	NaN
6	7	sale	12	2023-07-27	NaN
7	8	leave	45	2023-07-28	NaN

10. Display the statistical summary of data

In [22]:

```
b.describe()
```

Out[22]:

	A	C	E
count	10.00000	10.000000	0.0
mean	5.50000	33.900000	NaN
std	3.02765	21.496511	NaN
min	1.00000	11.000000	NaN
25%	3.25000	14.750000	NaN
50%	5.50000	28.500000	NaN
75%	7.75000	45.000000	NaN
max	10.00000	67.000000	NaN

In [ ]: