Analysis

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

1.Create any Series and print the output

In [5]:
```
df = pd.Series(1)
print(df)
```

```
0    1
dtype: int64
```

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

In [7]:
```
df = pd.DataFrame(
{
    "A":5,
    "B":6,
    "C":pd.Timestamp("20231007"),
    "D":78,
    "E":pd.Series(1,index=list(range(10))),
})
df
```

Out[7]:

	A	B	C	D	E
0	5	6	2023-10-07	78	1
1	5	6	2023-10-07	78	1
2	5	6	2023-10-07	78	1
3	5	6	2023-10-07	78	1
4	5	6	2023-10-07	78	1
5	5	6	2023-10-07	78	1
6	5	6	2023-10-07	78	1
7	5	6	2023-10-07	78	1
8	5	6	2023-10-07	78	1
9	5	6	2023-10-07	78	1

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

In [9]:
```
df.head(7)
```

Out[9]:

	A	B	C	D	E
0	5	6	2023-10-07	78	1
1	5	6	2023-10-07	78	1
2	5	6	2023-10-07	78	1
3	5	6	2023-10-07	78	1
4	5	6	2023-10-07	78	1
5	5	6	2023-10-07	78	1
6	5	6	2023-10-07	78	1

In [10]:
```
df.tail(6)
```

Out[10]:

	A	B	C	D	E
4	5	6	2023-10-07	78	1
5	5	6	2023-10-07	78	1
6	5	6	2023-10-07	78	1
7	5	6	2023-10-07	78	1
8	5	6	2023-10-07	78	1
9	5	6	2023-10-07	78	1

4. Fill with a constant value and print the output

In [11]:
```
df1 = pd.DataFrame(
{
    "A":5,
    "B":6,
    "C":pd.Timestamp("20231007"),
    "D":78,
    "E":pd.Series(index=list(range(10))),
})
df1
```

```
C:\Users\user\AppData\Local\Temp/ipykernel_8772/703782190.py:7: DeprecationWarning: The default dtype for empty Series will be
'object' instead of 'float64' in a future version. Specify a dtype explicitly to silence this warning.
  "E":pd.Series(index=list(range(10))),
```

Out[11]:

	A	B	C	D	E
0	5	6	2023-10-07	78	NaN
1	5	6	2023-10-07	78	NaN
2	5	6	2023-10-07	78	NaN
3	5	6	2023-10-07	78	NaN
4	5	6	2023-10-07	78	NaN
5	5	6	2023-10-07	78	NaN
6	5	6	2023-10-07	78	NaN
7	5	6	2023-10-07	78	NaN
8	5	6	2023-10-07	78	NaN
9	5	6	2023-10-07	78	NaN

In [12]:
```
df1.fillna(value=11)
```

Out[12]:

	A	B	C	D	E
0	5	6	2023-10-07	78	11.0
1	5	6	2023-10-07	78	11.0
2	5	6	2023-10-07	78	11.0
3	5	6	2023-10-07	78	11.0
4	5	6	2023-10-07	78	11.0
5	5	6	2023-10-07	78	11.0
6	5	6	2023-10-07	78	11.0
7	5	6	2023-10-07	78	11.0
8	5	6	2023-10-07	78	11.0
9	5	6	2023-10-07	78	11.0

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

In [13]:
```
df2 = pd.DataFrame(
{
    "A":5,
    "B":6,
    "C":pd.Timestamp("20231007"),
    "D":78,
    "E":pd.Series(index=list(range(10))),
})
df2
```

```
C:\Users\user\AppData\Local\Temp/ipykernel_8772/3060633849.py:7: DeprecationWarning: The default dtype for empty Series will be
'object' instead of 'float64' in a future version. Specify a dtype explicitly to silence this warning.
  "E":pd.Series(index=list(range(10))),
```

Out[13]:

	A	B	C	D	E
0	5	6	2023-10-07	78	NaN
1	5	6	2023-10-07	78	NaN
2	5	6	2023-10-07	78	NaN
3	5	6	2023-10-07	78	NaN
4	5	6	2023-10-07	78	NaN
5	5	6	2023-10-07	78	NaN
6	5	6	2023-10-07	78	NaN
7	5	6	2023-10-07	78	NaN
8	5	6	2023-10-07	78	NaN
9	5	6	2023-10-07	78	NaN

In [14]:
```
df2.dropna(axis=1,how='all')
```

Out[14]:

	A	B	C	D
0	5	6	2023-10-07	78
1	5	6	2023-10-07	78
2	5	6	2023-10-07	78
3	5	6	2023-10-07	78
4	5	6	2023-10-07	78
5	5	6	2023-10-07	78
6	5	6	2023-10-07	78
7	5	6	2023-10-07	78
8	5	6	2023-10-07	78
9	5	6	2023-10-07	78

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

In [15]:
```
df3 = pd.DataFrame(
{
    "A":5,
    "B":6,
    "C":pd.Timestamp("20231007"),
    "D":78,
    "E":pd.Series(index=list(range(10))),
})
df3
```

```
C:\Users\user\AppData\Local\Temp/ipykernel_8772/2183112572.py:7: DeprecationWarning: The default dtype for empty Series will be
'object' instead of 'float64' in a future version. Specify a dtype explicitly to silence this warning.
  "E":pd.Series(index=list(range(10))),
```

Out[15]:

	A	B	C	D	E
0	5	6	2023-10-07	78	NaN
1	5	6	2023-10-07	78	NaN
2	5	6	2023-10-07	78	NaN
3	5	6	2023-10-07	78	NaN
4	5	6	2023-10-07	78	NaN
5	5	6	2023-10-07	78	NaN
6	5	6	2023-10-07	78	NaN
7	5	6	2023-10-07	78	NaN
8	5	6	2023-10-07	78	NaN
9	5	6	2023-10-07	78	NaN

In [16]:
```
df3.dropna()
```

Out[16]:

	A	B	C	D	E

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

In [17]:
```
df4 = pd.DataFrame(
{
    "A":5,
    "B":6,
    "C":pd.Timestamp("20231007"),
    "D":pd.Series(index=list(range(10))),
    "E":pd.Series(index=list(range(10))),
})
df4
```

```
C:\Users\user\AppData\Local\Temp/ipykernel_8772/1845060358.py:6: DeprecationWarning: The default dtype for empty Series will be
'object' instead of 'float64' in a future version. Specify a dtype explicitly to silence this warning.
  "D":pd.Series(index=list(range(10))),
C:\Users\user\AppData\Local\Temp/ipykernel_8772/1845060358.py:7: DeprecationWarning: The default dtype for empty Series will be
'object' instead of 'float64' in a future version. Specify a dtype explicitly to silence this warning.
  "E":pd.Series(index=list(range(10))),
```

Out[17]:

	A	B	C	D	E
0	5	6	2023-10-07	NaN	NaN
1	5	6	2023-10-07	NaN	NaN
2	5	6	2023-10-07	NaN	NaN
3	5	6	2023-10-07	NaN	NaN
4	5	6	2023-10-07	NaN	NaN
5	5	6	2023-10-07	NaN	NaN
6	5	6	2023-10-07	NaN	NaN
7	5	6	2023-10-07	NaN	NaN
8	5	6	2023-10-07	NaN	NaN
9	5	6	2023-10-07	NaN	NaN

In [18]:
```
df4.isna()
```

Out[18]:

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

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

In [19]:
```
df4[df4["A"]>=2]
```

Out[19]:

	A	B	C	D	E
0	5	6	2023-10-07	NaN	NaN
1	5	6	2023-10-07	NaN	NaN
2	5	6	2023-10-07	NaN	NaN
3	5	6	2023-10-07	NaN	NaN
4	5	6	2023-10-07	NaN	NaN
5	5	6	2023-10-07	NaN	NaN
6	5	6	2023-10-07	NaN	NaN
7	5	6	2023-10-07	NaN	NaN
8	5	6	2023-10-07	NaN	NaN
9	5	6	2023-10-07	NaN	NaN

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

In [20]:
```
df4.loc["A":"E"]
```

Out[20]:

	A	B	C	D	E

10. Display the statistical summary of data

In [21]:
```
df4.iloc[1:5]
```

Out[21]:

	A	B	C	D	E
1	5	6	2023-10-07	NaN	NaN
2	5	6	2023-10-07	NaN	NaN
3	5	6	2023-10-07	NaN	NaN
4	5	6	2023-10-07	NaN	NaN

In [ ]:
```

```