ASSIGNMENT -DAY3

kaviyadevi(20106064)

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

1.printing series

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

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

2.creating dataframe of 10*5

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

Out[5]:

	0	1	2	3	4
0	0.310896	-2.729624	0.023347	2.003032	0.268801
1	0.803039	1.449944	-0.287658	0.114633	0.669356
2	0.482701	1.645162	0.835453	-0.545293	-0.268292
3	0.264689	0.589819	-0.229511	-1.404624	-0.112069
4	-1.684771	0.449952	-1.095002	2.919310	0.233633
5	0.613585	-1.711131	0.836495	-0.098399	0.540936
6	1.349525	-2.332081	1.082465	0.836576	1.080600
7	0.321097	-0.822717	-3.454995	-0.720404	0.660481
8	1.449063	2.608233	3.382435	-0.158817	0.542734
9	-0.357426	0.210969	0.002661	-0.757349	0.358902

3.Displaying top 7 and last 6 rows from dataset

In [6]:
```
a.head()
```

Out[6]:

	0	1	2	3	4
0	0.310896	-2.729624	0.023347	2.003032	0.268801
1	0.803039	1.449944	-0.287658	0.114633	0.669356
2	0.482701	1.645162	0.835453	-0.545293	-0.268292
3	0.264689	0.589819	-0.229511	-1.404624	-0.112069
4	-1.684771	0.449952	-1.095002	2.919310	0.233633

4.constant values

In [7]:
```
df=pd.DataFrame(
    {'a':[10,99,np.nan,65],
     'b':[30,5,69,np.nan],
     'c':[np.nan,56,76,23],
                  })
df.fillna(value=0)
```

Out[7]:

	a	b	c
0	10.0	30.0	0.0
1	99.0	5.0	56.0
2	0.0	69.0	76.0
3	65.0	0.0	23.0

5.dropping the column with missing values

In [11]:
```
df=pd.DataFrame(
{'a':[10,99,np.nan,65],
 'b':[30,5,69,np.nan],

})
df.dropna()
```

Out[11]:

	a	b
0	10.0	30.0
1	99.0	5.0

6.Dropping the values with missing values

In [15]:
```
df=pd.DataFrame(
{'a':[10,99,np.nan,95],
'b':[30,5,69,np.nan],
 'c':[np.nan,56,76,23]
})
df.dropna()
```

Out[15]:

	a	b	c
1	99.0	5.0	56.0

7.Missing values in the dataframe

In [16]:
```
i={'first':[10,99,np.nan,95],
   'second':[30,5,69,np.nan],
   'third':[np.nan,56,76,23]}
df=pd.DataFrame(i)
df.isnull()
df
```

Out[16]:

	first	second	third
0	10.0	30.0	NaN
1	99.0	5.0	56.0
2	NaN	69.0	76.0
3	95.0	NaN	23.0

8.Displaying loc and iloc

In [17]:
```
df.loc[0:2]
```

Out[17]:

	first	second	third
0	10.0	30.0	NaN
1	99.0	5.0	56.0
2	NaN	69.0	76.0

9.loc and iloc using operators

In [18]:
```
df=pd.DataFrame(
{'a':[10,99,np.nan,95],
'b':[30,5,69,np.nan],
 'c':[np.nan,56,76,23]
})
df[df["a"]>2]
```

Out[18]:

	a	b	c
0	10.0	30.0	NaN
1	99.0	5.0	56.0
3	95.0	NaN	23.0

10.Displaying statical summary of data

In [19]:
```
df=pd.DataFrame(
{'a':[10,99,np.nan,95],
'b':[30,5,69,np.nan],
 'c':[np.nan,56,76,23]
})
df.describe()
```

Out[19]:

	a	b	c
count	3.000000	3.000000	3.000000
mean	68.000000	34.666667	51.666667
std	50.269275	32.254199	26.764404
min	10.000000	5.000000	23.000000
25%	52.500000	17.500000	39.500000
50%	95.000000	30.000000	56.000000
75%	97.000000	49.500000	66.000000
max	99.000000	69.000000	76.000000

In [ ]:
```
```