In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as se
```

In [36]:
```
sp=pd.read_csv("/home/student/Downloads/Demo/Employee_Salary_Datase
```

In [3]:
```
sp.head(6)
```

Out[3]:

	ID	Experience_Years	Age	Gender	Salary
0	1	5	28	Female	250000
1	2	1	21	Male	50000
2	3	3	23	Female	170000
3	4	2	22	Male	25000
4	5	1	17	Male	10000
5	6	25	62	Male	5001000

In [4]:
```
sp.mean()
```

```
/tmp/ipykernel_2609/3291234476.py:1: FutureWarning: The default v
alue of numeric_only in DataFrame.mean is deprecated. In a future
version, it will default to False. In addition, specifying 'numer
ic_only=None' is deprecated. Select only valid columns or specify
the value of numeric_only to silence this warning.
  sp.mean()
```

Out[4]:
```
ID                   1.800000e+01
Experience_Years     9.200000e+00
Age                  3.548571e+01
Salary               2.059147e+06
dtype: float64
```

In [5]:
```
sp.loc[:,'Salary'].mean()
```

Out[5]:
```
2059147.142857143
```

In [6]:
```
sp.loc[:,'Age'].mean()
```

Out[6]:
```
35.48571428571429
```

In [7]:
```
sp.mean(axis=1)[0:4]
```

```
/tmp/ipykernel_2609/2676889982.py:1: FutureWarning: Dropping of n
uisance columns in DataFrame reductions (with 'numeric_only=None
') is deprecated; in a future version this will raise TypeError.
Select only valid columns before calling the reduction.
  sp.mean(axis=1)[0:4]
```

Out[7]:

```
0    62508.50
```

In [8]:
```
sp.median()
```

```
/tmp/ipykernel_2609/1657900362.py:1: FutureWarning: The default v
alue of numeric_only in DataFrame.median is deprecated. In a futu
re version, it will default to False. In addition, specifying 'nu
meric_only=None' is deprecated. Select only valid columns or spec
ify the value of numeric_only to silence this warning.
  sp.median()
```

Out[8]:
```
ID                     18.0
Experience_Years        6.0
Age                    29.0
Salary             250000.0
dtype: float64
```

In [9]:
```
sp.loc[:,'Salary'].median()
```

Out[9]:
```
250000.0
```

In [10]:
```
sp.median(axis=1)[0:4]
```

```
/tmp/ipykernel_2609/3639203685.py:1: FutureWarning: Dropping of n
uisance columns in DataFrame reductions (with 'numeric_only=None
') is deprecated; in a future version this will raise TypeError.
Select only valid columns before calling the reduction.
  sp.median(axis=1)[0:4]
```

Out[10]:
```
0    16.5
1    11.5
2    13.0
3    13.0
dtype: float64
```

In [11]:
```
sp.mode()
```

Out[11]:

	ID	Experience_Years	Age	Gender	Salary
0	1	2.0	54.0	Female	25000.0
1	2	NaN	NaN	NaN	250000.0
2	3	NaN	NaN	NaN	NaN
3	4	NaN	NaN	NaN	NaN
4	5	NaN	NaN	NaN	NaN
5	6	NaN	NaN	NaN	NaN
6	7	NaN	NaN	NaN	NaN
7	8	NaN	NaN	NaN	NaN
8	9	NaN	NaN	NaN	NaN
9	10	NaN	NaN	NaN	NaN
10	11	NaN	NaN	NaN	NaN
11	12	NaN	NaN	NaN	NaN
12	13	NaN	NaN	NaN	NaN

	ID	Experience_Years	Age	Gender	Salary
13	14	NaN	NaN	NaN	NaN
14	15	NaN	NaN	NaN	NaN
15	16	NaN	NaN	NaN	NaN
16	17	NaN	NaN	NaN	NaN
17	18	NaN	NaN	NaN	NaN
18	19	NaN	NaN	NaN	NaN
19	20	NaN	NaN	NaN	NaN
20	21	NaN	NaN	NaN	NaN
21	22	NaN	NaN	NaN	NaN
22	23	NaN	NaN	NaN	NaN
23	24	NaN	NaN	NaN	NaN
24	25	NaN	NaN	NaN	NaN
25	26	NaN	NaN	NaN	NaN
26	27	NaN	NaN	NaN	NaN
27	28	NaN	NaN	NaN	NaN
28	29	NaN	NaN	NaN	NaN
29	30	NaN	NaN	NaN	NaN
30	31	NaN	NaN	NaN	NaN
31	32	NaN	NaN	NaN	NaN
32	33	NaN	NaN	NaN	NaN
33	34	NaN	NaN	NaN	NaN

In [12]:
```
sp.loc[:,'Salary'].mode()
```

Out[12]:
```
0     25000
1    250000
Name: Salary, dtype: int64
```

In [13]:
```
sp.mean()
```

```
/tmp/ipykernel_2609/3291234476.py:1: FutureWarning: The default v
alue of numeric_only in DataFrame.mean is deprecated. In a future
version, it will default to False. In addition, specifying 'numer
ic_only=None' is deprecated. Select only valid columns or specify
the value of numeric_only to silence this warning.
  sp.mean()
```

Out[13]:
```
ID                  1.800000e+01
Experience_Years    9.200000e+00
Age                 3.548571e+01
Salary              2.059147e+06
dtype: float64
```

In [14]:
```
sp.max()
```

Out[14]:
```
ID                        35
Experience_Years          27
Age                       62
Gender                  Male
Salary              10000000
dtype: object
```

In [15]:
```
sp.std()
```

```
/tmp/ipykernel_2609/2171739191.py:1: FutureWarning: The default v
alue of numeric_only in DataFrame.std is deprecated. In a future
version, it will default to False. In addition, specifying 'numer
ic_only=None' is deprecated. Select only valid columns or specify
the value of numeric_only to silence this warning.
  sp.std()
```

Out[15]:
```
ID                  1.024695e+01
Experience_Years    7.552950e+00
Age                 1.464355e+01
Salary              3.170124e+06
dtype: float64
```

In [16]:
```
sp.loc[:'Salary'].std()
```

```
/tmp/ipykernel_2609/843767344.py:1: FutureWarning: The default va
lue of numeric_only in DataFrame.std is deprecated. In a future v
ersion, it will default to False. In addition, specifying 'numeri
c_only=None' is deprecated. Select only valid columns or specify
the value of numeric_only to silence this warning.
  sp.loc[:'Salary'].std()
```

Out[16]:
```
ID                  1.024695e+01
Experience_Years    7.552950e+00
Age                 1.464355e+01
Salary              3.170124e+06
dtype: float64
```

In [17]:
```
sp.loc[:'Age'].std()
```

```
/tmp/ipykernel_2609/3884215968.py:1: FutureWarning: The default v
alue of numeric_only in DataFrame.std is deprecated. In a future
version, it will default to False. In addition, specifying 'numer
ic_only=None' is deprecated. Select only valid columns or specify
the value of numeric_only to silence this warning.
  sp.loc[:'Age'].std()
```

Out[17]:
```
ID                  1.024695e+01
Experience_Years    7.552950e+00
Age                 1.464355e+01
Salary              3.170124e+06
dtype: float64
```

In [18]:
```
sp.std(axis=1)[0:6]
```

```
/tmp/ipykernel_2609/3427172873.py:1: FutureWarning: Dropping of n
```

Out[18]:
```
0    1.249943e+05
1    2.499600e+04
2    8.499517e+04
3    1.249534e+04
4    4.996171e+03
5    2.500485e+06
dtype: float64
```

In [20]:
```
sp.groupby(['Salary'])['Age'].mean()
```

Out[20]:
```
Salary
3000         18.0
6000         21.0
6100         21.0
7500         23.0
8900         23.0
9000         21.0
10000        17.0
15000        21.0
20000        22.0
25000        24.0
50000        21.0
61500        36.0
80000        34.0
87000        27.0
170000       23.0
220100       40.0
250000       27.0
330000       36.0
650000       54.0
800000       54.0
900000       54.0
930000       34.0
1400000      29.0
1540000      55.0
5000000      54.0
5001000      62.0
6000050      39.0
6570000      54.0
6845000      29.0
7600000      49.0
7900000      54.0
9300000      53.0
10000000     62.0
Name: Age, dtype: float64
```

In [23]:
```
sp.rename(columns={'Salary':'salary'},inplace=True)
```

In [24]:
```
sp.head()
```

Out[24]:

	ID	Experience_Years	Age	Gender	salary
0	1	5	28	Female	250000
1	2	1	21	Male	50000
2	3	3	23	Female	170000

	ID	Experience_Years	Age	Gender	salary
3	4	2	22	Male	25000

In [26]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

In [31]:
```
sp['Gender']=le.fit_transform(sp['Gender'])
newdf=sp
sp
```

Out[31]:

	ID	Experience_Years	Age	Gender	salary
0	1	5	28	0	250000
1	2	1	21	1	50000
2	3	3	23	0	170000
3	4	2	22	1	25000
4	5	1	17	1	10000
5	6	25	62	1	5001000
6	7	19	54	0	800000
7	8	2	21	0	9000
8	9	10	36	0	61500
9	10	15	54	0	650000
10	11	4	26	0	250000
11	12	6	29	1	1400000
12	13	14	39	1	6000050
13	14	11	40	1	220100
14	15	2	23	1	7500
15	16	4	27	0	87000
16	17	10	34	0	930000
17	18	15	54	0	7900000
18	19	2	21	1	15000
19	20	10	36	1	330000
20	21	15	54	1	6570000
21	22	4	26	1	25000
22	23	5	29	1	6845000
23	24	1	21	0	6000
24	25	4	23	0	8900
25	26	3	22	0	20000
26	27	1	18	1	3000
27	28	27	62	0	10000000

	ID	Experience_Years	Age	Gender	salary
28	29	19	54	0	5000000
29	30	2	21	0	6100
30	31	10	34	1	80000
31	32	15	54	1	900000
32	33	20	55	0	1540000
33	34	19	53	0	9300000

In [33]:
```
sp.dropna(how="all")
```

Out[33]:

	ID	Experience_Years	Age	Gender	salary
0	1	5	28	0	250000
1	2	1	21	1	50000
2	3	3	23	0	170000
3	4	2	22	1	25000
4	5	1	17	1	10000
5	6	25	62	1	5001000
6	7	19	54	0	800000
7	8	2	21	0	9000
8	9	10	36	0	61500
9	10	15	54	0	650000
10	11	4	26	0	250000
11	12	6	29	1	1400000
12	13	14	39	1	6000050
13	14	11	40	1	220100
14	15	2	23	1	7500
15	16	4	27	0	87000
16	17	10	34	0	930000
17	18	15	54	0	7900000
18	19	2	21	1	15000
19	20	10	36	1	330000
20	21	15	54	1	6570000
21	22	4	26	1	25000
22	23	5	29	1	6845000
23	24	1	21	0	6000
24	25	4	23	0	8900
25	26	3	22	0	20000

	ID	Experience_Years	Age	Gender	salary
26	27	1	18	1	3000
27	28	27	62	0	10000000
28	29	19	54	0	5000000
29	30	2	21	0	6100
30	31	10	34	1	80000
31	32	15	54	1	900000
32	33	20	55	0	1540000

In [38]:
```
spp=pd.read_csv("/home/student/Downloads/Demo/Iris.csv")
```

In [39]:
```
spp
```

Out[39]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [46]:
```
from sklearn import preprocessing
enc=preprocessing.OneHotEncoder()
enc_df=pd.DataFrame(enc.fit_transform(spp[['PetalWidthCm']]).toarra
enc_df
```

Out[46]:

	0	1	2	3	4	5	6	7	8	9	...	12	13	14	15	16	17	18	19
0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	0	1	2	3	4	5	6	7	8	9	...	12	13	14	15	16	17	18	19
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
145	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
146	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
147	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
148	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
149	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0

In [51]:
```
irisSet=(spp['Species']=='Iris-setosa')
print('Iris-setosa')
print(spp[irisSet].describe())
```

```
Iris-setosa
              Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  Peta
lWidthCm
count  50.00000       50.00000     50.000000      50.000000
50.00000
mean   25.50000        5.00600      3.418000       1.464000
0.24400
std    14.57738        0.35249      0.381024       0.173511
0.10721
min     1.00000        4.30000      2.300000       1.000000
0.10000
25%    13.25000        4.80000      3.125000       1.400000
0.20000
50%    25.50000        5.00000      3.400000       1.500000
0.20000
75%    37.75000        5.20000      3.675000       1.575000
0.30000
max    50.00000        5.80000      4.400000       1.900000
0.60000
```

In [54]:
```
irisVer=(spp['Species']=='Iris-setosa')
print('Iris-setosa')
print(spp[irisVer].describe())
```

```
Iris-setosa
              Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  Peta
lWidthCm
count  50.00000       50.00000     50.000000      50.000000
50.00000
mean   25.50000        5.00600      3.418000       1.464000
0.24400
std    14.57738        0.35249      0.381024       0.173511
0.10721
min     1.00000        4.30000      2.300000       1.000000
0.10000
25%    13.25000        4.80000      3.125000       1.400000
0.20000
50%    25.50000        5.00000      3.400000       1.500000
0.20000
75%    37.75000        5.20000      3.675000       1.575000
0.30000
max    50.00000        5.80000      4.400000       1.900000
0.60000
```

In [ ]: