```
In [27]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as se
```

```
In [28]: sp=pd.read_csv("/home/student/Desktop/Employee_Salary_Dataset.csv")
```

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

Out[4]: ID                        1
        Experience_Years          1
        Age                      17
        Gender               Female
        Salary                 3000
        dtype: object
```

```
In [5]: sp.max()

Out[5]: ID                          35
        Experience_Years            27
        Age                         62
        Gender                    Male
        Salary                10000000
        dtype: object
```

```
In [6]: sp.std()

        /tmp/ipykernel_2740/2171739191.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[6]: ID                  1.024695e+01
        Experience_Years    7.552950e+00
        Age                 1.464355e+01
        Salary              3.170124e+06
        dtype: float64
```

```
In [7]: sp.mean()

        /tmp/ipykernel_2740/3291234476.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[7]: ID                  1.800000e+01
        Experience_Years    9.200000e+00
        Age                 3.548571e+01
        Salary              2.059147e+06
        dtype: float64
```

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

```
Out[8]:
        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
   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
```

	ID	Experience_Years	Age	Gender	Salary
23	24	NaN	NaN	NaN	NaN

In [9]:
```
sp.mean(axis=1)[0:4]
```

```
/tmp/ipykernel_2740/2676889982.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[9]:
```
0    62508.50
1    12506.00
2    42507.25
3     6257.00
dtype: float64
```

In [10]:
```
sp.groupby(['Experience_Years'])['Age'].mean()
```

Out[10]:
```
Experience_Years
1     19.250000
2     21.600000
3     22.500000
4     25.500000
5     28.500000
6     29.000000
10    35.000000
11    40.000000
14    39.000000
15    54.000000
16    49.000000
19    53.666667
20    55.000000
25    62.000000
27    62.000000
Name: Age, dtype: float64
```

In [15]:
```
sp_u = sp.rename(columns={'Experience_Years':'E_year'},inplace=Fals
```

In [16]:
```
(sp_u.groupby(['Age']).E_year.mean())
```

Out[16]:

```
Age
17      1.000000
18      1.000000
```

In [17]:
```
sp.head()
```

Out[17]:

	ID	E_year	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

In [19]:
```
sp.std(axis=1)[0:4]
```

```
/tmp/ipykernel_2740/2545776365.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.std(axis=1)[0:4]
```

Out[19]:
```
0    124994.333900
1     24996.001694
2     84995.167190
3     12495.336570
dtype: float64
```

In [20]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

In [24]:
```
sp['Gender'] = le.fit_transform(sp['Gender'])
newdf = sp
sp
```

Out[24]:

	ID	E_year	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

	ID	E_year	Age	Gender	Salary
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
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 [25]: sp.isnull()
```

Out[25]:

	ID	E_year	Age	Gender	Salary
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	False	False	False

	ID	E_year	Age	Gender	Salary
10	False	False	False	False	False
11	False	False	False	False	False
12	False	False	False	False	False
13	False	False	False	False	False
14	False	False	False	False	False
15	False	False	False	False	False
16	False	False	False	False	False
17	False	False	False	False	False
18	False	False	False	False	False
19	False	False	False	False	False
20	False	False	False	False	False
21	False	False	False	False	False
22	False	False	False	False	False
23	False	False	False	False	False
24	False	False	False	False	False
25	False	False	False	False	False
26	False	False	False	False	False
27	False	False	False	False	False
28	False	False	False	False	False
29	False	False	False	False	False
30	False	False	False	False	False
31	False	False	False	False	False
32	False	False	False	False	False
33	False	False	False	False	False

In [29]:
```
sp.dropna(how='all')
```

Out[29]:

	ID	Experience_Years	Age	Gender	Salary
0	1	NaN	28	Female	250000
1	2	1.0	21	Male	50000
2	3	3.0	23	Female	170000
3	4	2.0	22	Male	25000
4	5	1.0	17	Male	10000
5	6	25.0	62	Male	5001000
6	7	19.0	54	Female	800000
7	8	2.0	21	Female	9000

	ID	Experience_Years	Age	Gender	Salary
8	9	10.0	36	Female	61500
9	10	15.0	54	Female	650000
10	11	4.0	26	Female	250000
11	12	6.0	29	Male	1400000
12	13	14.0	39	Male	6000050
13	14	11.0	40	Male	220100
14	15	2.0	23	Male	7500
15	16	4.0	27	Female	87000
16	17	10.0	34	Female	930000
17	18	15.0	54	Female	7900000
18	19	2.0	21	Male	15000
19	20	10.0	36	Male	330000
20	21	15.0	54	Male	6570000
21	22	4.0	26	Male	25000
22	23	5.0	29	Male	6845000
23	24	1.0	21	Female	6000
24	25	4.0	23	Female	8900
25	26	3.0	22	Female	20000
26	27	1.0	18	Male	3000
27	28	27.0	62	Female	10000000
28	29	19.0	54	Female	5000000
29	30	2.0	21	Female	6100
30	31	10.0	34	Male	80000
31	32	15.0	54	Male	900000
32	33	20.0	55	Female	1540000
33	34	19.0	53	Female	9300000

In [30]:

```
sp=pd.read_csv("/home/student/Desktop/Iris.csv")
```

In [34]:

```
from sklearn import preprocessing
enc = preprocessing.OneHotEncoder()
enc_df= pd.DataFrame(enc.fit_transform(sp[['SepalWidthCm']]).toarra
enc_df
```

Out[34]:

	0	1	2	3	4	5	6	7	8	9	...	13	14	15	16	17	18	19	20
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	0.0
1	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	0.0	0.0	0.0
2	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	0.0

	0	1	2	3	4	5	6	7	8	9	...	13	14	15	16	17	18	19	20
3	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	0.0
4	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
145	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	0.0	0.0	0.0
146	0.0	0.0	0.0	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
147	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	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	...	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
149	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	0.0	0.0	0.0

```
In [36]: df_encode = sp_u.join(enc_df)
         df_encode
```

Out[36]:

	ID	E_year	Age	Gender	Salary	0	1	2	3	4	...	13	14	15	16	17
0	1	5	28	Female	250000	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0
1	2	1	21	Male	50000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
2	3	3	23	Female	170000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
3	4	2	22	Male	25000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
4	5	1	17	Male	10000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0
5	6	25	62	Male	5001000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
6	7	19	54	Female	800000	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0
7	8	2	21	Female	9000	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0
8	9	10	36	Female	61500	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
9	10	15	54	Female	650000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
10	11	4	26	Female	250000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0
11	12	6	29	Male	1400000	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0
12	13	14	39	Male	6000050	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
13	14	11	40	Male	220100	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
14	15	2	23	Male	7500	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
15	16	4	27	Female	87000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
16	17	10	34	Female	930000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
17	18	15	54	Female	7900000	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0
18	19	2	21	Male	15000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0
19	20	10	36	Male	330000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0
20	21	15	54	Male	6570000	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0
21	22	4	26	Male	25000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0
22	23	5	29	Male	6845000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0
23	24	1	21	Female	6000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0

	ID	E_year	Age	Gender	Salary	0	1	2	3	4	...	13	14	15	16	17
24	25	4	23	Female	8900	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0
25	26	3	22	Female	20000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
26	27	1	18	Male	3000	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0
27	28	27	62	Female	10000000	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0
28	29	19	54	Female	5000000	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0
29	30	2	21	Female	6100	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
30	31	10	34	Male	80000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
31	32	15	54	Male	900000	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	0.0
32	33	20	55	Female	1540000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
33	34	19	53	Female	9300000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
34	35	16	49	Male	7600000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0

In [41]:
```
irisSet = (sp['Species']== 'Iris-setosa')
print('Iris-setosa')
print(sp[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 [42]:
```
irisVer = (sp['Species']== 'Iris-setosa')
print('Iris-setosa')
print(sp[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
```

In [44]:

```
irisVir = (sp['Species']=='Iris-setosa')
print('Iris-setosa')
print(sp[irisVir].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 [ ]: