

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

```
In [3]: sp = pd.read_csv("/home/student/Desktop/Employee_Salary_Dataset.csv")
```

```
In [4]: sp.head(6)
```

```
Out[4]:
```

	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 [5]: pd.read_csv("/home/student/Desktop/Employee_Salary_Dataset.csv")
```

Out[5]:

	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
6	7	19	54	Female	800000
7	8	2	21	Female	9000
8	9	10	36	Female	61500
9	10	15	54	Female	650000
10	11	4	26	Female	250000
11	12	6	29	Male	1400000
12	13	14	39	Male	6000050
13	14	11	40	Male	220100
14	15	2	23	Male	7500
15	16	4	27	Female	87000
16	17	10	34	Female	930000
17	18	15	54	Female	7900000
18	19	2	21	Male	15000
19	20	10	36	Male	330000
20	21	15	54	Male	6570000
21	22	4	26	Male	25000
22	23	5	29	Male	6845000
23	24	1	21	Female	6000
24	25	4	23	Female	8900
25	26	3	22	Female	20000
26	27	1	18	Male	3000
27	28	27	62	Female	10000000
28	29	19	54	Female	5000000
29	30	2	21	Female	6100
30	31	10	34	Male	80000
31	32	15	54	Male	900000
32	33	20	55	Female	1540000
33	34	19	53	Female	9300000

	ID	Experience_Years	Age	Gender	Salary
34	35	16	49	Male	7600000

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

Out[6]:

ID	1.800000e+01
Experience_Years	9.200000e+00
Age	3.548571e+01
Salary	2.059147e+06

dtype: float64

In [7]: `sp.loc[:, 'Age'].mean()`

Out[7]: 35.48571428571429

In [8]: `sp.mean(axis=1)[0:4]`

Out[8]:

0	62508.50
1	12506.00
2	42507.25
3	6257.00

dtype: float64

In [9]: `sp.median()`

Out[9]:

ID	18.0
Experience_Years	6.0
Age	29.0
Salary	250000.0

dtype: float64

In [11]: `sp.loc[:, 'Age'].median()`

Out[11]: 29.0

In [12]: `sp.median(axis=1)[0:4]`

Out[12]:

0	16.5
1	11.5
2	13.0
3	13.0

dtype: float64

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

Out[13]:

	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
33	34	NaN	NaN	NaN	NaN

	ID	Experience_Years	Age	Gender	Salary
34	35	NaN	NaN	NaN	NaN

In [15]: `sp.mode(axis=1)[0:4]`

Out[15]:

	0	1	2	3	4
0	1	5.0	28.0	Female	250000.0
1	2	1.0	21.0	Male	50000.0
2	3	NaN	NaN	NaN	NaN
3	4	2.0	22.0	Male	25000.0

In [19]: `sp.loc[:, 'Age'].mode()`

Out[19]: 0 54
Name: Age, dtype: int64

In [20]: `sp.min()`

Out[20]: ID 1
Experience_Years 1
Age 17
Gender Female
Salary 3000
dtype: object

In [21]: `sp.loc[:, 'Salary'].min(skipna=False)`

Out[21]: 3000

In [22]: `sp.max()`

Out[22]: ID 35
Experience_Years 27
Age 62
Gender Male
Salary 10000000
dtype: object

In [23]: `sp.loc[:, 'Salary'].max(skipna=False)`

Out[23]: 10000000

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

Out[24]: ID 1.024695e+01
Experience_Years 7.552950e+00
Age 1.464355e+01
Salary 3.170124e+06
dtype: float64

In [25]: `sp.loc[:, 'Age'].std()`

Out[25]: 14.643551940884361

```
In [26]: sp.std(axis=1)[0:4]
```

Out[26]:

0	124994.333900
1	24996.001694
2	84995.167190
3	12495.336570

dtype: float64

```
In [27]: sp.groupby(['Salary'])['Age'].mean()
```

Out[27]:

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 [37]: sp_u=sp.rename(columns= {'ID': 'EMP_ID'},inplace=False)
(sp_u.groupby(['Salary']).Age.mean())
```

```
Out[37]: 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 [31]: sp.std()
```

```
Out[31]: ID          1.024695e+01
Experience_Years  7.552950e+00
Age             1.464355e+01
Salary          3.170124e+06
dtype: float64
```

```
In [34]: sp.head()
```

```
Out[34]:
```

	EMP_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

```
In [38]: from sklearn import preprocessing
enc=preprocessing.OneHotEncoder()
```

```
enc_sp=pd.DataFrame(enc.fit_transform(sp[['Salary']]).toarray())  
enc_sp
```


Out[38]:

[illegible]

	0	1	2	3	4	5	6	7	8	9	...	23	24	25	26	27	28	29	30	31	32
34	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	0.0	0.0	0.0

35 rows × 33 columns

```
In [39]: sp_encode=sp_u.join(enc_sp)
          sp_encode
```

Out[39]:

	EMP_ID	Experience_Years	Age	Gender	Salary	0	1	2	3	4	...	23	24	25	26	...
0	1	5	28	Female	250000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	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	0.0	0.0	0.0
5	6	25	62	Male	5001000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.0
6	7	19	54	Female	800000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
7	8	2	21	Female	9000	0.0	0.0	0.0	0.0	0.0	...	0.0	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	0.0	0.0
11	12	6	29	Male	1400000	0.0	0.0	0.0	0.0	0.0	...	0.0	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	1.0	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	1.0	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	0.0	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	0.0
19	20	10	36	Male	330000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
20	21	15	54	Male	6570000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0
21	22	4	26	Male	25000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
22	23	5	29	Male	6845000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
23	24	1	21	Female	6000	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
24	25	4	23	Female	8900	0.0	0.0	0.0	0.0	1.0	...	0.0	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	1.0	0.0	0.0	0.0	0.0	...	0.0	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	0.0	0.0	0.0	0.0
28	29	19	54	Female	5000000	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0
29	30	2	21	Female	6100	0.0	0.0	1.0	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	...	0.0	0.0	0.0	0.0	0.0
32	33	20	55	Female	1540000	0.0	0.0	0.0	0.0	0.0	...	1.0	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

	EMP_ID	Experience_Years	Age	Gender	Salary	0	1	2	3	4	...	23	24	25	26	...
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 [40]: iris=pd.read_csv("/home/student/Desktop/Iris.csv")
```

```
In [41]: import pandas as pd
```

```
In [42]: iris.head()
```

```
Out[42]:
```

	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

```
In [43]: irisSet=(iris['Species']=='Iris-versicolor')
print('Iris-versicolor')
print(iris[irisSet].describe())
```

```
Iris-versicolor
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.000000	50.000000	50.000000	50.000000
mean	75.50000	5.936000	2.770000	4.260000	1.326000
std	14.57738	0.516171	0.313798	0.469911	0.197753
min	51.00000	4.900000	2.000000	3.000000	1.000000
25%	63.25000	5.600000	2.525000	4.000000	1.200000
50%	75.50000	5.900000	2.800000	4.350000	1.300000
75%	87.75000	6.300000	3.000000	4.600000	1.500000
max	100.00000	7.000000	3.400000	5.100000	1.800000

```
In [44]: irisVer=(iris['Species']=='Iris-versicolor')
print('Iris-versicolor')
print(iris[irisVer].describe())
```

```
Iris-versicolor
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.000000	50.000000	50.000000	50.000000
mean	75.50000	5.936000	2.770000	4.260000	1.326000
std	14.57738	0.516171	0.313798	0.469911	0.197753
min	51.00000	4.900000	2.000000	3.000000	1.000000
25%	63.25000	5.600000	2.525000	4.000000	1.200000
50%	75.50000	5.900000	2.800000	4.350000	1.300000
75%	87.75000	6.300000	3.000000	4.600000	1.500000
max	100.00000	7.000000	3.400000	5.100000	1.800000

```
In [45]: irisVir=(iris['Species']=='Iris-versicolor')
print('Iris-versicolor')
print(iris[irisVir].describe())
```

Iris-versicolor

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.000000	50.000000	50.000000	50.000000
mean	75.50000	5.936000	2.770000	4.260000	1.326000
std	14.57738	0.516171	0.313798	0.469911	0.197753
min	51.00000	4.900000	2.000000	3.000000	1.000000
25%	63.25000	5.600000	2.525000	4.000000	1.200000
50%	75.50000	5.900000	2.800000	4.350000	1.300000
75%	87.75000	6.300000	3.000000	4.600000	1.500000
max	100.00000	7.000000	3.400000	5.100000	1.800000

In []: