```
import pandas as pd
```

Data Exploration:

```
df=pd.read_csv('/content/Employee.csv')
df
```

	Company	Age	Salary	Place	Country	Gender
0	TCS	20.0	NaN	Chennai	India	0
1	Infosys	30.0	NaN	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0
...	...	...	...	...	...	...
143	TCS	33.0	9024.0	Calcutta	India	1
144	Infosys	22.0	8787.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	TCS	33.0	5034.0	Mumbai	India	1
147	Infosys	22.0	8202.0	Cochin	India	0

148 rows × 6 columns

Next steps: View recommended plots New interactive sheet

The data set for the employee and its contain Company name, Employee age, Employee salary,Employee place,country and Gender

Complany- Company Name Age- Employee age Salary - Employee salary Place - Employee Place Country - Employee country Gender- Emplyee Gender

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148 entries, 0 to 147
Data columns (total 6 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   Company  140 non-null    object 
 1   Age      130 non-null    float64
 2   Salary   124 non-null    float64
 3   Place    134 non-null    object 
 4   Country  148 non-null    object 
 5   Gender   148 non-null    int64  
dtypes: float64(2), int64(1), object(3)
memory usage: 7.1+ KB
```

```
df.describe()
```

```
              Age       Salary      Gender
count  130.000000   124.000000  148.000000
mean    30.484615  5312.467742    0.222973
std     11.096640  2573.764683    0.417654
min      0.000000  1089.000000    0.000000
25%     22.000000  3030.000000    0.000000
50%     32.500000  5000.000000    0.000000
75%     37.750000  8000.000000    0.000000
max     54.000000  9876.000000    1.000000
```

List down the unique values

```
df['Company'].value_counts()
```

```
                            count
                   Company
            TCS                53
          Infosys              45
            CTS                36
Tata Consultancy Services       2
        Congnizant              2
      Infosys Pvt Lmt           2

dtype: int64
```

```
df['Place'].value_counts()
```

	count
Place	
Mumbai	37
Calcutta	33
Chennai	14
Delhi	14
Cochin	13
Noida	8
Hyderabad	8
Podicherry	3
Pune	2
Bhopal	1
Nagpur	1

dtype: int64

```
df['Country'].value_counts()
```

	count
Country	
India	148

dtype: int64

```
df['Gender'].value_counts()
```

	count
Gender	
0	115
1	33

dtype: int64

```
#Rename the coulms
df2=df.copy()
```

```
df2=df2.rename({'Company':'Comp_name','Age':'Emp_age','Salary':'Emp_salary','Place':'Emp_place','Country':'
df2
```

	Comp_name	Emp_age	Emp_salary	Emp_place	Emp_country	Emp_gender
0	TCS	20.0	NaN	Chennai	India	0
1	Infosys	30.0	NaN	Mumbai	India	0
2	TCS	35.0	2300.0	Calcutta	India	0
3	Infosys	40.0	3000.0	Delhi	India	0
4	TCS	23.0	4000.0	Mumbai	India	0
...	...	...	...	...	...	...
143	TCS	33.0	9024.0	Calcutta	India	1
144	Infosys	22.0	8787.0	Calcutta	India	1
145	Infosys	44.0	4034.0	Delhi	India	1
146	TCS	33.0	5034.0	Mumbai	India	1
147	Infosys	22.0	8202.0	Cochin	India	0

148 rows × 6 columns

Next steps: View recommended plots New interactive sheet

```
df2.shape
```

```
(148, 6)
```

Data Cleaning:

```
#Find duplicates
df2.duplicated().sum()
```

```
4
```

```
#Remove duplicates
df2.drop_duplicates(inplace=True)
df2.shape
```

```
(144, 6)
```

```
#Find null values
df2.isnull().sum()
```

	0
Comp_name	8
Emp_age	17
Emp_salary	23
Emp_place	14
Emp_country	0
Emp_gender	0

dtype: int64

```
round(df2.isnull().mean()*100,2)
```

	0
Comp_name	5.56
Emp_age	11.81
Emp_salary	15.97
Emp_place	9.72
Emp_country	0.00
Emp_gender	0.00

dtype: float64

```
#remove rows with null values in company name
df2.dropna(subset=['Comp_name'],axis=0,inplace=True)
```

```
#Replace the value 0 in age as NaN
df2['Emp_age']=df2['Emp_age'].fillna(0)
```

```
df2.shape
```

```
(136, 6)
```

```
round(df2.isnull().mean()*100,2)
```

```
                     0
Comp_name         0.00
Emp_age           0.00
Emp_salary       14.71
Emp_place         9.56
Emp_country       0.00
Emp_gender        0.00

dtype: float64
```

```
#Treat the null values in all columns using any measures(removing/ replace the values with mean/median/mode
```

```
round(df2.isnull().mean()*100,2)
```

```
                     0
Comp_name         0.00
Emp_age           0.00
Emp_salary       14.71
Emp_place         9.56
Emp_country       0.00
Emp_gender        0.00

dtype: float64
```

```
#Replace Emp_salary nulll values with mean
mean=df2['Emp_salary'].mean()
df2['Emp_salary'].fillna(mean,inplace=True)
```

```
round(df2.isnull().mean()*100,2)
```

```
                    0
Comp_name        0.00
Emp_age          0.00
Emp_salary       0.00
Emp_place        9.56
Emp_country      0.00
Emp_gender       0.00

dtype: float64
```

```
#Replace Emp_place nulll values with mean
mod=df2['Emp_place'].mode()

df2['Emp_place'].fillna(mod[0],inplace=True)
```

```
round(df2.isnull().mean()*100,2)
```

	0
Comp_name	0.0
Emp_age	0.0
Emp_salary	0.0
Emp_place	0.0
Emp_country	0.0
Emp_gender	0.0

dtype: float64

Find the Outliers

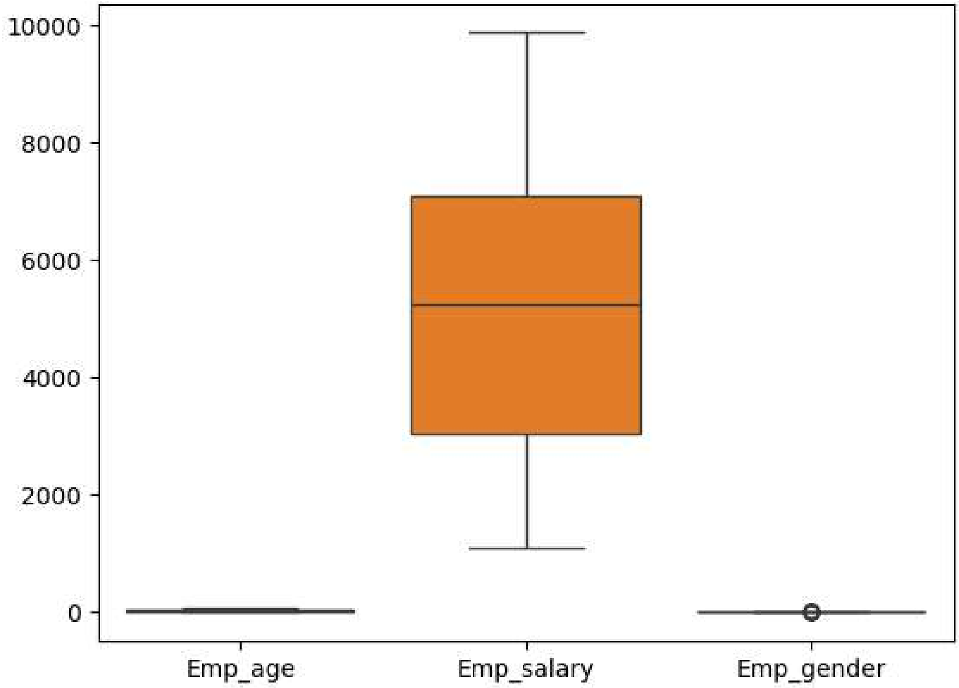
```
num=df2.select_dtypes('number')
num.skew()
```

	0
Emp_age	-0.643296
Emp_salary	0.214116
Emp_gender	1.311559

dtype: float64

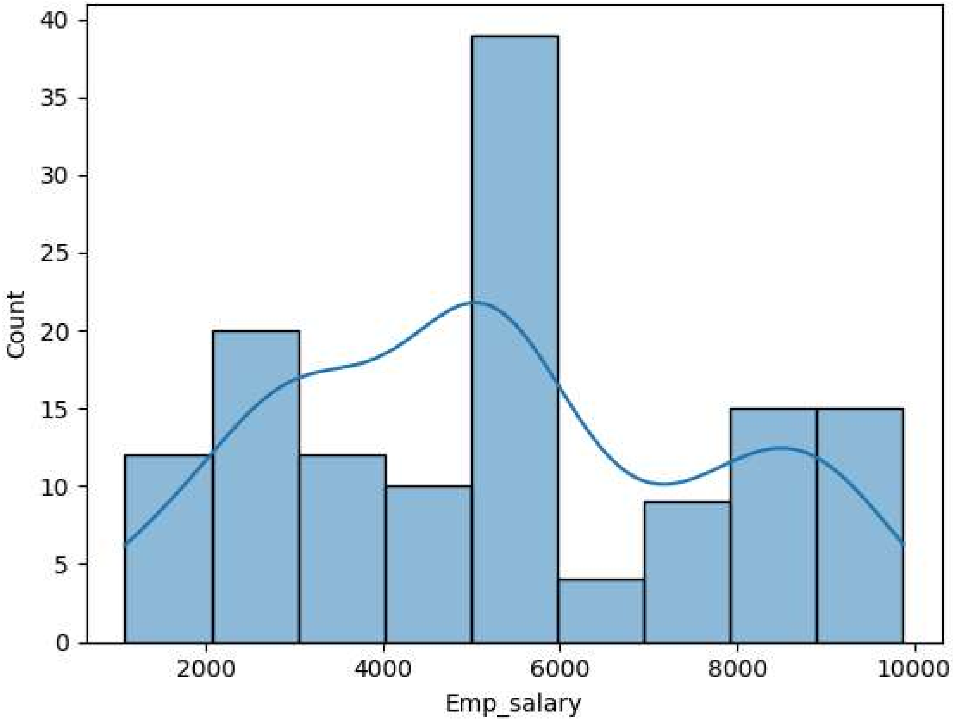
```
import seaborn as sns
sns.boxplot(df2)
```

```
<Axes: >
```



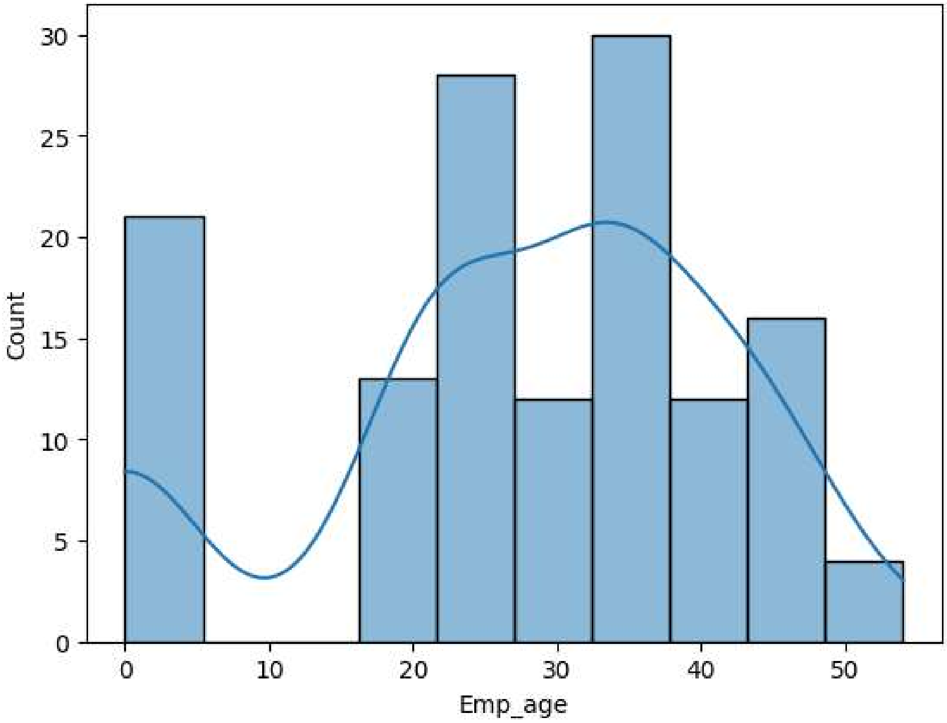
```
sns.histplot(df2['Emp_salary'],kde=True)
```

```
<Axes: xlabel='Emp_salary', ylabel='Count'>
```



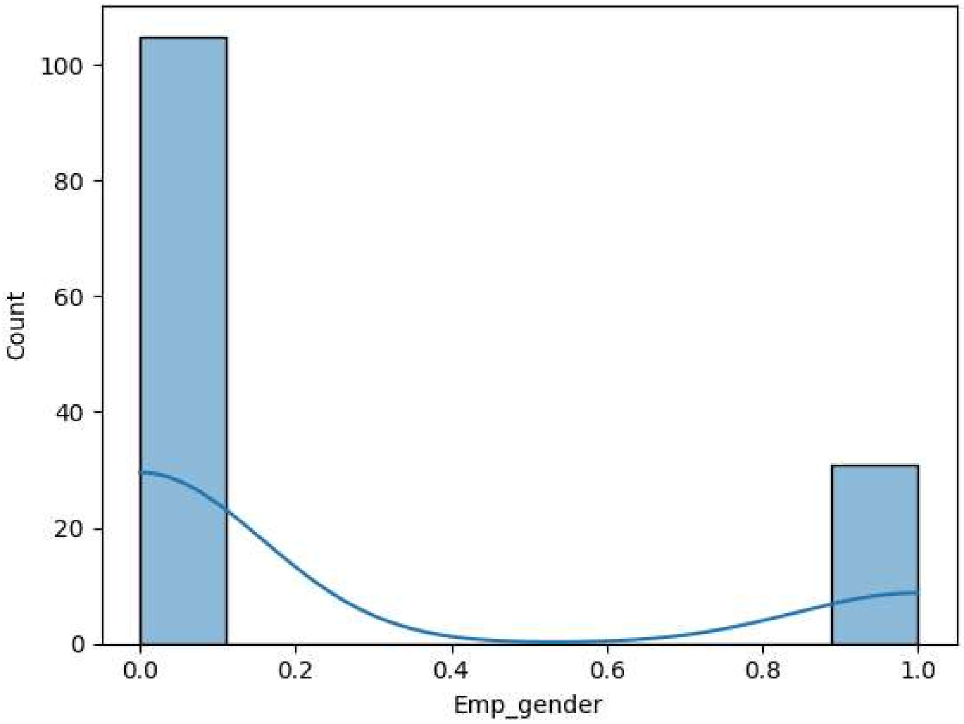
```
sns.histplot(df2['Emp_age'],kde=True)
```

```
<Axes: xlabel='Emp_age', ylabel='Count'>
```



```
sns.histplot(df2['Emp_gender'],kde=True)
```

```
<Axes: xlabel='Emp_gender', ylabel='Count'>
```



Data Analysis:

Filter the data with age >40 and salary<5000

```
filtred_data=df2[(df2['Emp_age']>40)&(df2['Emp_salary']<5000)]
filtred_data
```

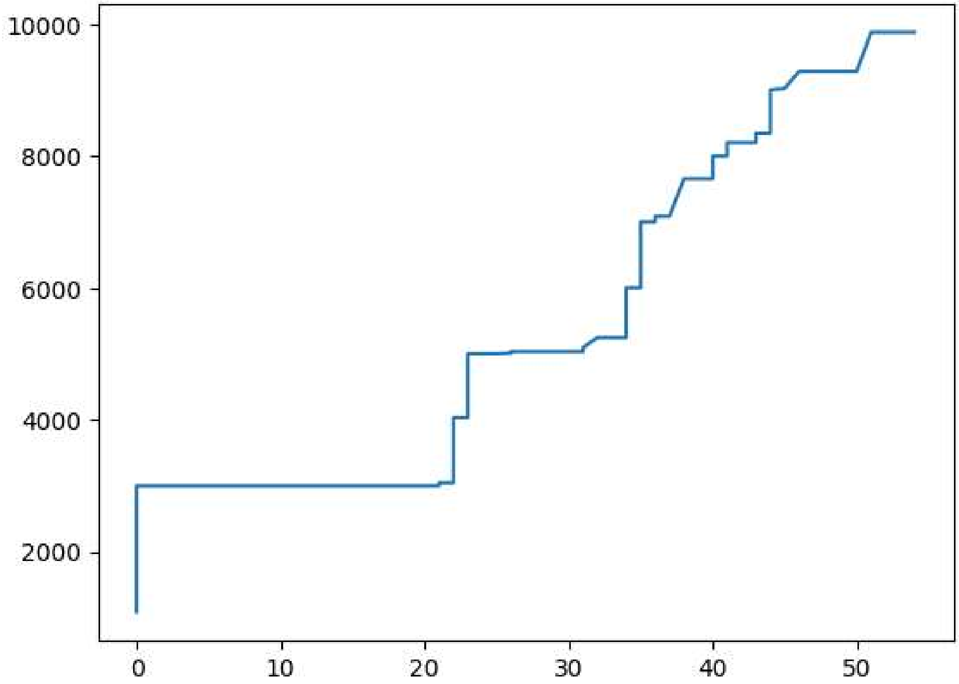
	Comp_name	Emp_age	Emp_salary	Emp_place	Emp_country	Emp_gender
21	Infosys	50.0	3184.0	Delhi	India	0
32	Infosys	45.0	4034.0	Calcutta	India	0
39	Infosys	41.0	3000.0	Mumbai	India	0
50	Infosys	41.0	3000.0	Chennai	India	0
57	Infosys	51.0	3184.0	Hyderabad	India	0
68	Infosys	43.0	4034.0	Mumbai	India	0
75	Infosys	44.0	3000.0	Cochin	India	0
86	Infosys	41.0	3000.0	Delhi	India	0
93	Infosys	54.0	3184.0	Mumbai	India	0
104	Infosys	44.0	4034.0	Delhi	India	0
122	Infosys	44.0	3234.0	Mumbai	India	0
129	Infosys	50.0	3184.0	Calcutta	India	0
138	CTS	44.0	3033.0	Cochin	India	0
140	Infosys	44.0	4034.0	Hyderabad	India	0
145	Infosys	44.0	4034.0	Delhi	India	1

Next steps: View recommended plots New interactive sheet

Plot the chart with age and salary

```
import matplotlib.pyplot as plt


x=df2['Emp_age'].sort_values(ascending=True)
y=df2['Emp_salary'].sort_values(ascending=True)
plt.plot(x,y)
plt.show()
```



Count the number of people from each place and represent it visually

```
df2['Emp_place'].value_counts()
```

	count
Emp_place	
Mumbai	46
Calcutta	30
Chennai	13
Delhi	13
Cochin	13
Noida	7
Hyderabad	7
Podicherry	3
Pune	2
Bhopal	1
Nagpur	1

dtype: int64

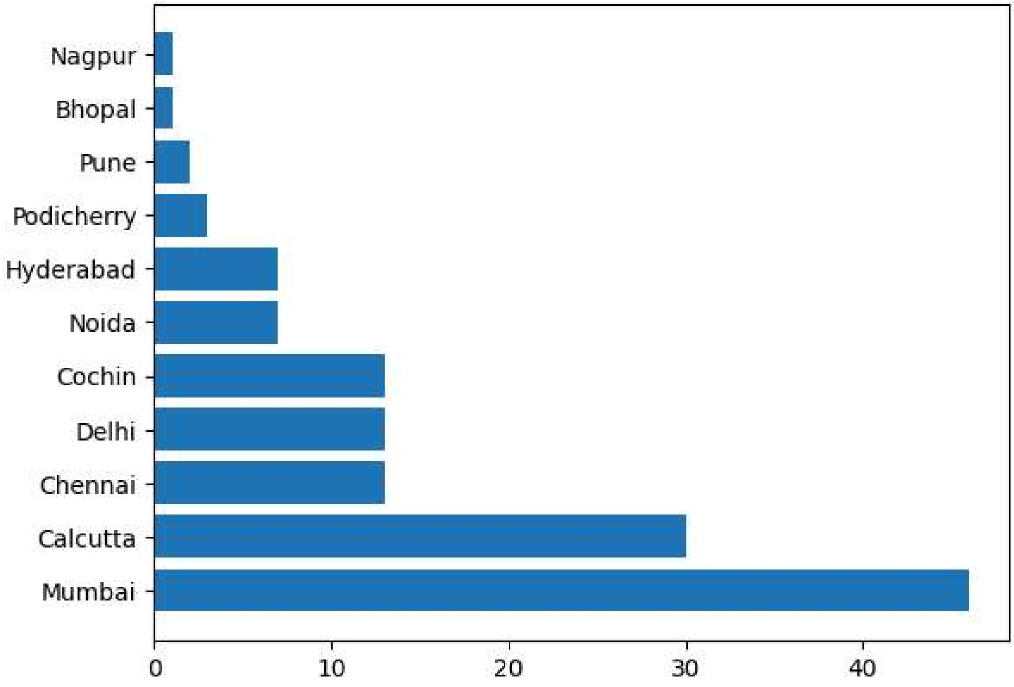
```
data=df2['Emp_place'].value_counts()
x=list(data.index)
x
```

```
['Mumbai',
 'Calcutta',
       'Chennai',
       'Delhi',
       'Cochin',
       'Noida',
       'Hyderabad',
       'Podicherry',
       'Pune',
       'Bhopal',
       'Nagpur']
```

```
y=df2['Emp_place'].value_counts().values
y
```

```
array([46, 30, 13, 13, 13,  7,  7,  3,  2,  1,  1])
```

```
plt.barh(x,y)
plt.show()
```



Data Encoding:

```
df2
```

	Comp_name	Emp_age	Emp_salary	Emp_place	Emp_country	Emp_gender
0	TCS	20.0	5244.974138	Chennai	India	0
1	Infosys	30.0	5244.974138	Mumbai	India	0
2	TCS	35.0	2300.000000	Calcutta	India	0
3	Infosys	40.0	3000.000000	Delhi	India	0
4	TCS	23.0	4000.000000	Mumbai	India	0
...	...	...	...	...	...	...
142	Infosys Pvt Lmt	22.0	8202.000000	Mumbai	India	0
143	TCS	33.0	9024.000000	Calcutta	India	1
145	Infosys	44.0	4034.000000	Delhi	India	1
146	TCS	33.0	5034.000000	Mumbai	India	1
147	Infosys	22.0	8202.000000	Cochin	India	0

136 rows × 6 columns

Next steps: View recommended plots New interactive sheet

label encoding

```
from sklearn import preprocessing
lbl_encoder=preprocessing.LabelEncoder()


df2['place_lbl_encoded']=lbl_encoder.fit_transform(df2['Emp_place'])
df2['Comp_name_lbl_encoded']=lbl_encoder.fit_transform(df2['Comp_name'])
df2
```

	Comp_name	Emp_age	Emp_salary	Emp_place	Emp_country	Emp_gender	place_lbl_encoded	Comp_name_l
0	TCS	20.0	5244.974138	Chennai	India	0	2	
1	Infosys	30.0	5244.974138	Mumbai	India	0	6	
2	TCS	35.0	2300.000000	Calcutta	India	0	1	
3	Infosys	40.0	3000.000000	Delhi	India	0	4	
4	TCS	23.0	4000.000000	Mumbai	India	0	6	
...	...	...	...	...	...	...	...	
142	Infosys Pvt Lmt	22.0	8202.000000	Mumbai	India	0	6	
143	TCS	33.0	9024.000000	Calcutta	India	1	1	
145	Infosys	44.0	4034.000000	Delhi	India	1	4	
146	TCS	33.0	5034.000000	Mumbai	India	1	6	
147	Infosys	22.0	8202.000000	Cochin	India	0	3	

136 rows × 8 columns

Next steps: View recommended plots New interactive sheet

Feature Scaling:

minmaxscaler

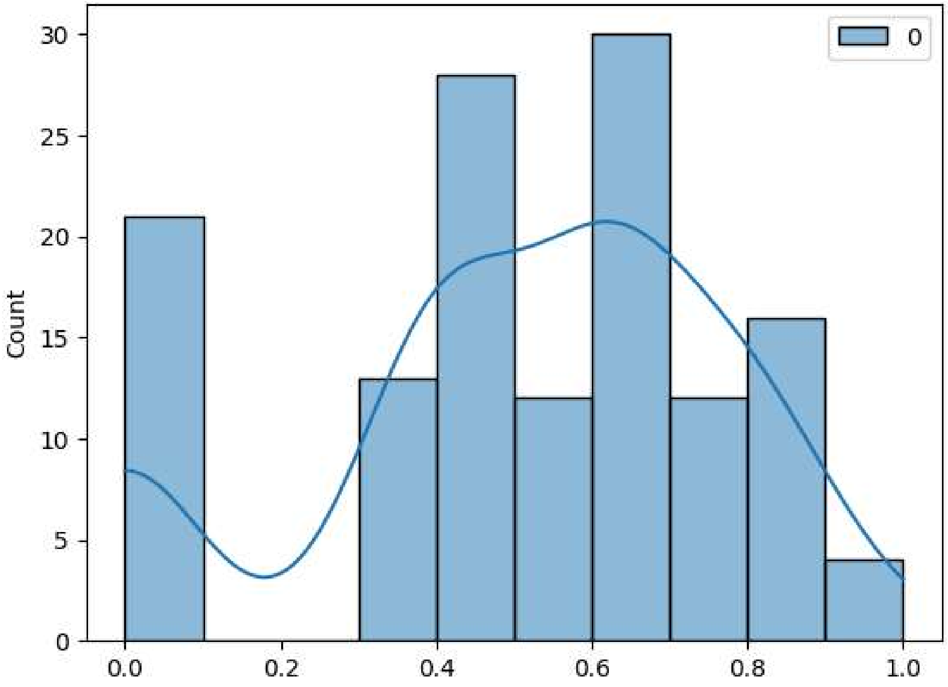
```
from sklearn.preprocessing import MinMaxScaler,StandardScaler
```

```
min_max_scaler=MinMaxScaler()
x=df2[['Emp_age']]
age_min_max_scale= min_max_scaler.fit_transform(x)
age_min_max_scale
```

```
array([[0.37037037],
       [0.55555556],
       [0.64814815],
       [0.74074074],
       [0.42592593],
       [0.        ],
       [0.        ],
       [0.42592593],
       [0.62962963],
       [0.83333333],
       [0.42592593],
       [0.62962963],
       [0.83333333],
       [0.33333333],
       [0.74074074],
       [0.42592593],
       [0.42592593],
            [0.62962963],
            [0.40740741],
            [0.59259259],
            [0.68518519],
            [0.92592593],
            [0.38888889],
            [0.        ],
            [0.        ],
            [0.42592593],
            [0.62962963],
            [0.83333333],
            [0.42592593],
            [0.64814815],
            [0.85185185],
            [0.37037037],
            [0.83333333],
            [0.66666667],
            [0.48148148],
            [0.64814815],
            [0.59259259],
            [0.62962963],
            [0.75925926],
            [0.44444444],
            [0.        ],
            [0.        ],
            [0.46296296],
            [0.64814815],
            [0.85185185],
            [0.44444444],
            [0.59259259],
            [0.7962963 ],
            [0.35185185],
            [0.75925926],
            [0.38888889],
            [0.64814815],
            [0.38888889],
            [0.59259259],
            [0.7037037 ],
            [0.94444444],
            [0.42592593],
            [0.        ]
```

```
import seaborn as sns
sns.histplot(age_min_max_scale,kde=True)
```

```
<Axes: ylabel='Count'>
```



Standard scaler

```
x=df2[['Emp_salary']]
standard_scaler=StandardScaler()
sal_standard_scale=standard_scaler.fit_transform(x)
sal_standard_scale
```

```
       [-3.88318071e-01],
       [ 8.01222022e-02],
       [ 1.30718811e-01],
       [ 1.49345415e+00],
       [ 1.49345415e+00],
       [ 1.95261838e+00],
       [-3.83477899e-16],
       [-1.75231833e+00],
       [-3.83477899e-16],
```