Empoloyee Data Project
Data Preprocessing
Concept

Employee Data

- Data Analyzing
- Treat Missing values
- Apply a Encoding concept
- Aplly a One hot coder
- Apply the Dummy Variable(n-1)

In [1]:
```
## import a operating system path
import os
os.getcwd()
```

Out[1]:
```
'C:\\Users\\ap983'
```

In [2]:
```
# import a Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
### read data with the help of pandas
data = pd.read_csv('C:/Users/ap983/Desktop/Pandey  IMp/COMPLETE PYTHON PROJECT/
data
```

Out[3]:

	Employee Id	First Name	Last Name	Department	Age	Experience	Salary
0	1	Joy	Bass	Sales and Marketing	28.0	3.0	32889
1	2	Sheila	Garza	Sales and Marketing	22.0	1.0	15944
2	3	John	Bryant	Customer Relations	22.0	1.0	40343
3	4	Christian	Farley	Customer Relations	22.0	1.0	19018
4	5	Colorado	Bowen	Accounting	27.0	0.0	24795
...	...	...	...	...	...	...	...
95	96	Tyrone	Barber	Sales and Marketing	47.0	22.0	79077
96	97	Urielle	Herrera	Sales and Marketing	46.0	20.0	81187
97	98	Brendan	Solis	Customer Relations	44.0	18.0	83847
98	99	Holmes	Nelson	Customer Relations	47.0	22.0	89158
99	100	Avye	Davidson	Accounting	44.0	21.0	91645

100 rows × 7 columns

In [4]:
```
# first few observation
data.head()
```

Out[4]:

	Employee Id	First Name	Last Name	Department	Age	Experience	Salary
0	1	Joy	Bass	Sales and Marketing	28.0	3.0	32889
1	2	Sheila	Garza	Sales and Marketing	22.0	1.0	15944
2	3	John	Bryant	Customer Relations	22.0	1.0	40343
3	4	Christian	Farley	Customer Relations	22.0	1.0	19018
4	5	Colorado	Bowen	Accounting	27.0	0.0	24795

In [5]:
```
# last few observation
data.tail()
```

Out[5]:

	Employee Id	First Name	Last Name	Department	Age	Experience	Salary
95	96	Tyrone	Barber	Sales and Marketing	47.0	22.0	79077
96	97	Urielle	Herrera	Sales and Marketing	46.0	20.0	81187
97	98	Brendan	Solis	Customer Relations	44.0	18.0	83847
98	99	Holmes	Nelson	Customer Relations	47.0	22.0	89158
99	100	Avye	Davidson	Accounting	44.0	21.0	91645

In [6]:
```
## check the shape of the dataset
data.shape
```

Out[6]:
```
(100, 7)
```

In [7]:
```
## check the columns name in the data set
data.columns
```

Out[7]:
```
Index(['Employee Id', 'First Name', 'Last Name', 'Department', 'Age',
       'Experience', 'Salary'],
      dtype='object')
```

In [8]:
```
# collect the information
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Employee Id  100 non-null    int64  
 1   First Name   100 non-null    object 
 2   Last Name    100 non-null    object 
 3   Department   81 non-null     object 
 4   Age          82 non-null     float64
 5   Experience   90 non-null     float64
 6   Salary       100 non-null    int64  
dtypes: float64(2), int64(2), object(3)
memory usage: 5.6+ KB
```

In [9]:
```
# check the null value
data.isna().sum()
```

Out[9]:
```
Employee Id     0
First Name      0
Last Name       0
Department     19
Age            18
Experience     10
Salary          0
dtype: int64
```

In [10]:
```
## check the statistics
data.describe(include = 'all')
```

Out[10]:

	Employee Id	First Name	Last Name	Department	Age	Experience	Salary
count	100.000000	100	100	81	82.000000	90.000000	100.000000
unique	NaN	97	93	3	NaN	NaN	NaN
top	NaN	Quentin	Davidson	Sales and Marketing	NaN	NaN	NaN
freq	NaN	2	3	30	NaN	NaN	NaN
mean	50.500000	NaN	NaN	NaN	37.975610	14.766667	65066.760000
std	29.011492	NaN	NaN	NaN	9.515388	6.889252	26189.874212
min	1.000000	NaN	NaN	NaN	22.000000	0.000000	11830.000000
25%	25.750000	NaN	NaN	NaN	27.250000	9.250000	48526.000000
50%	50.500000	NaN	NaN	NaN	42.000000	16.500000	73500.500000
75%	75.250000	NaN	NaN	NaN	45.750000	21.000000	86621.250000
max	100.000000	NaN	NaN	NaN	50.000000	25.000000	98180.000000

In [11]:
```
#  percent of  missing data are there in dataset

data.isnull().sum()/len(data)*100
```

Out[11]:
```
Employee Id     0.0
First Name      0.0
Last Name       0.0
Department     19.0
Age            18.0
Experience     10.0
Salary          0.0
dtype: float64
```

In [12]:
```
# observvw thw how many data are missing
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Employee Id  100 non-null    int64
 1   First Name   100 non-null    object
 2   Last Name    100 non-null    object
 3   Department   81 non-null     object
 4   Age          82 non-null     float64
 5   Experience   90 non-null     float64
 6   Salary       100 non-null    int64
dtypes: float64(2), int64(2), object(3)
memory usage: 5.6+ KB
```

In [13]:
```
#  3   Department   81 non-null     object
# approch modal
# how many data are given in the class "Department"
data['Department'].value_counts()
```

Out[13]:
```
Sales and Marketing    30
Accounting             27
Customer Relations     24
Name: Department, dtype: int64
```

In [14]:
```
# check the mode of the department
data['Department'].mode()
```

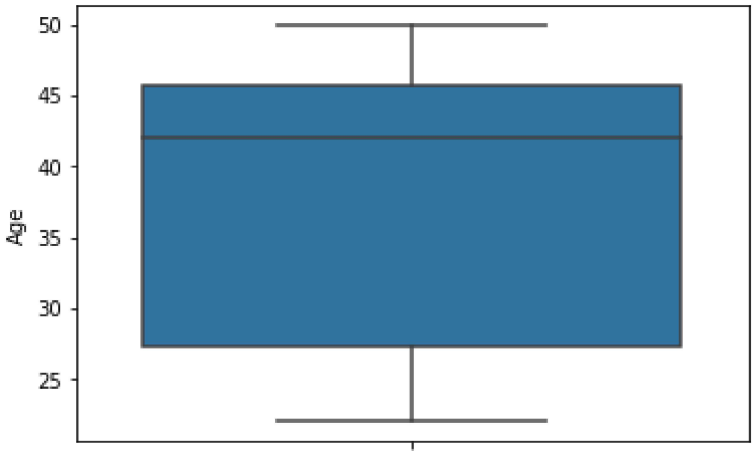
Out[14]:
```
0    Sales and Marketing
Name: Department, dtype: object
```

In [15]:
```
# handling char and missing vakriable
data['Department']=data['Department'].fillna('Sales and Marketing')
```

In [16]:
```
data.isnull().sum()
```

Out[16]:
```
Employee Id     0
First Name      0
Last Name       0
Department      0
Age            18
Experience     10
Salary          0
dtype: int64
```

In [17]:
```
##Age          82 non-null     float64
# 5   Experience   90 non-null     float64
#both age and experiance are numeric velue hence we have we have to do boxpot f
sns.boxplot(y='Age',data=data)
plt.show()
```



In [18]:
```
data['Age'].describe()
```

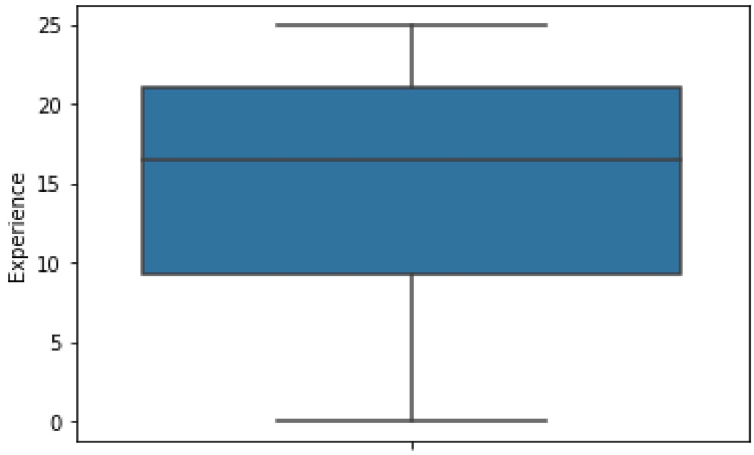
Out[18]:
```
count    82.000000
mean     37.975610
std       9.515388
min      22.000000
25%      27.250000
50%      42.000000
75%      45.750000
max      50.000000
Name: Age, dtype: float64
```

In [19]:
```
# Check the mean value of age
data ['Age'].mean()
```

Out[19]:
```
37.97560975609756
```

In [20]:
```
## fill the mean value of the age in the dataset
data['Age']=data['Age'].fillna(data['Age'].mean())
```

In [21]:
```
# check the outliers of Expeience
sns.boxplot(y='Experience',data=data)
plt.show()
# outliers not found in box plot
```



In [22]:
```
data['Experience'].describe()
```

Out[22]:
```
count    90.000000
mean     14.766667
std       6.889252
min       0.000000
25%       9.250000
50%      16.500000
75%      21.000000
max      25.000000
Name: Experience, dtype: float64
```

In [23]:
```
## check the mean value of Experience
data['Experience'].mean()
```

Out[23]:
```
14.766666666666667
```

In [24]:
```
# check the median value of Experience
data['Experience'].median()
```

Out[24]:
```
16.5
```

In [25]:
```
## fill the mean value of the Experience in Experience data set
data['Experience'] = data['Experience'].fillna(data['Experience'].mean())
```

In [26]:
```
data.isnull().sum()
```

Out[26]:
```
Employee Id    0
First Name     0
Last Name      0
Department     0
Age            0
Experience     0
Salary         0
dtype: int64
```

In [ ]:
```

```

Encoding Concept

Encoding means convert the data into category form here category form means lable here lable means 0,1,2,3 etc.

lablel encoder

In [27]:
```
data.describe(include = 'all')
```

Out[27]:

	Employee Id	First Name	Last Name	Department	Age	Experience	Salary
count	100.000000	100	100	100	100.000000	100.000000	100.000000
unique	NaN	97	93	3	NaN	NaN	NaN
top	NaN	Quentin	Davidson	Sales and Marketing	NaN	NaN	NaN
freq	NaN	2	3	49	NaN	NaN	NaN
mean	50.500000	NaN	NaN	NaN	37.975610	14.766667	65066.760000
std	29.011492	NaN	NaN	NaN	8.606992	6.532050	26189.874212
min	1.000000	NaN	NaN	NaN	22.000000	0.000000	11830.000000
25%	25.750000	NaN	NaN	NaN	30.000000	10.000000	48526.000000
50%	50.500000	NaN	NaN	NaN	39.500000	15.000000	73500.500000
75%	75.250000	NaN	NaN	NaN	44.250000	20.000000	86621.250000
max	100.000000	NaN	NaN	NaN	50.000000	25.000000	98180.000000

In [28]:
```
## check the inside the under of the  daperment  how many type of object  are c
data['Department'].value_counts()
```

Out[28]:
```
Sales and Marketing    49
Accounting             27
Customer Relations     24
Name: Department, dtype: int64
```

In [29]:
```
## convert the Department object in the category form
data['Department'] = data['Department'].astype('category')
```

In [30]:
```
data['Department'] = data['Department'].cat.codes
```

In [31]:
```
data.head()
```

Out[31]:

	Employee Id	First Name	Last Name	Department	Age	Experience	Salary
0	1	Joy	Bass	2	28.0	3.0	32889
1	2	Sheila	Garza	2	22.0	1.0	15944
2	3	John	Bryant	1	22.0	1.0	40343
3	4	Christian	Farley	1	22.0	1.0	19018
4	5	Colorado	Bowen	0	27.0	0.0	24795

In [32]:
```
# here check the three type of category available 0 to 2
data['Department'].value_counts()
```

Out[32]:
```
2    49
0    27
1    24
Name: Department, dtype: int64
```

Part 2 One Hot Coder

One Hot Encoding is a track a dummy varivale in side the data set i can apply Department columns Note : lets suppose inside a dateset three category then one hot coder is a create a three more columns

In [33]:
```
data = pd.get_dummies(data,columns=['Department'])
```

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

Out[34]:

	Employee Id	First Name	Last Name	Age	Experience	Salary	Department_0	Department_1	Departmer
0	1	Joy	Bass	28.0	3.0	32889	0	0	
1	2	Sheila	Garza	22.0	1.0	15944	0	0	
2	3	John	Bryant	22.0	1.0	40343	0	1	
3	4	Christian	Farley	22.0	1.0	19018	0	1	
4	5	Colorado	Bowen	27.0	0.0	24795	1	0	

Part 3 Dummy Variable(n-1) ¶

Here Dummy variable(n-1) means we can created a one hot coder dummy columns then i can remove that department_0 lets show the example i can remove a column

In [35]:
```
# dummy variable
data = data.drop(['Department_0'],axis = 1) # axis=1 means columns
```

In [36]:
```
## here remove the Department_0 show the dataset
data.head()
```

Out[36]:

	Employee Id	First Name	Last Name	Age	Experience	Salary	Department_1	Department_2
0	1	Joy	Bass	28.0	3.0	32889	0	1
1	2	Sheila	Garza	22.0	1.0	15944	0	1
2	3	John	Bryant	22.0	1.0	40343	1	0
3	4	Christian	Farley	22.0	1.0	19018	1	0
4	5	Colorado	Bowen	27.0	0.0	24795	0	0

Thank you Friends