In [1]:
```
import pandas as pd
```

In [2]:
```
pd.__version__
```

Out[2]:
```
'2.2.2'
```

In [3]:
```
emp = pd.read_excel(r"C:\Users\jayes\OneDrive\Desktop\NareshIT\27_mar\27th - EDA Practicle\27th - EDA Practicle\EDA-
```

In [4]:
```
emp.head()
```

Out[4]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience#$	34 years	Mumbai	5^00#0	2+
1	Teddy^	Testing	45' yr	Bangalore	10%%000	<3
2	Uma#r	Dataanalyst^^#	NaN	NaN	1$5%000	4> yrs
3	Jane	Ana^^lytics	NaN	Hyderbad	2000^0	NaN
4	Uttam*	Statistics	67-yr	NaN	30000-	5+ year

In [5]:
```
emp.isnull()
```

Out[5]:

	Name	Domain	Age	Location	Salary	Exp
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	True	True	False	False
3	False	False	True	False	False	True
4	False	False	False	True	False	False
5	False	False	False	False	False	False

In [6]:
```
len(emp.isnull())
```

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

In [7]:
```
id(emp)
```

Out[7]:
```
2394964040144
```

In [8]:
```
emp.columns
```

Out[8]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [9]:
```
emp.shape
```

Out[9]:
```
(6, 6)
```

In [10]:
```
emp.tail()
```

Out[10]:

	Name	Domain	Age	Location	Salary	Exp
1	Teddy^	Testing	45' yr	Bangalore	10%%000	<3
2	Uma#r	Dataanalyst^^#	NaN	NaN	1$5%000	4> yrs
3	Jane	Ana^^lytics	NaN	Hyderbad	2000^0	NaN
4	Uttam*	Statistics	67-yr	NaN	30000-	5+ year
5	Kim	NLP	55yr	Delhi	6000^$0	10+

In [11]:
```
emp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Name      6 non-null      object
 1   Domain    6 non-null      object
 2   Age       4 non-null      object
 3   Location  4 non-null      object
 4   Salary    6 non-null      object
 5   Exp       5 non-null      object
dtypes: object(6)
memory usage: 420.0+ bytes
```

In [12]:
```
emp.isna()
```

Out[12]:

	Name	Domain	Age	Location	Salary	Exp
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	True	True	False	False
3	False	False	True	False	False	True
4	False	False	False	True	False	False
5	False	False	False	False	False	False

In [13]:
```
len(emp.isna())
```

Out[13]:
```
6
```

In [14]:
```
emp.describe()
```

Out[14]:

	Name	Domain	Age	Location	Salary	Exp
count	6	6	4	4	6	5
unique	6	6	4	4	6	5
top	Mike	Datascience#$	34 years	Mumbai	5^00#0	2+
freq	1	1	1	1	1	1

In [15]:

```
emp.isnull().sum()
```

Out[15]:

```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

data cleaning or data cleansing

In [17]:

```
emp['Name']
```

Out[17]:

```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

In [18]:

```
emp['Name'] = emp['Name'].str.replace(r'\W','',regex='True')
```

In [19]:

```
emp['Name']
```

Out[19]:

```
0     Mike
1    Teddy
2     Umar
3     Jane
4    Uttam
5      Kim
Name: Name, dtype: object
```

In [20]:

```
emp['Domain']
```

Out[20]:

```
0     Datascience#$
1           Testing
2    Dataanalyst^^#
3       Ana^^lytics
4        Statistics
5               NLP
Name: Domain, dtype: object
```

In [21]:

```
emp['Domain'] = emp['Domain'].str.replace(r'\W','',regex=True)
```

In [76]:

```
emp['Domain']
```

Out[76]:

```
0    Datascience
1        Testing
2    Dataanalyst
3      Analytics
4     Statistics
5            NLP
Name: Domain, dtype: object
```

In [23]:

```
emp
```

Out[23]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34 years	Mumbai	5^00#0	2+
1	Teddy	Testing	45' yr	Bangalore	10%%000	<3
2	Umar	Dataanalyst	NaN	NaN	1$5%000	4> yrs
3	Jane	Analytics	NaN	Hyderbad	2000^0	NaN
4	Uttam	Statistics	67-yr	NaN	30000-	5+ year
5	Kim	NLP	55yr	Delhi	6000^$0	10+

In [24]:

```
emp['Age']=emp['Age'].str.replace(r'\W','',regex=True)
```

In [25]:

```
emp['Age']
```

Out[25]:

```
0    34years
1       45yr
2        NaN
3        NaN
4       67yr
5       55yr
Name: Age, dtype: object
```

In [26]:

```
emp['Age']=emp['Age'].str.extract(r'(\d+)')
```

In [27]:

```
emp['Age']
```

Out[27]:

```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [28]:

```
emp
```

Out[28]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5^00#0	2+
1	Teddy	Testing	45	Bangalore	10%%000	<3
2	Umar	Dataanalyst	NaN	NaN	1$5%000	4> yrs
3	Jane	Analytics	NaN	Hyderbad	2000^0	NaN
4	Uttam	Statistics	67	NaN	30000-	5+ year
5	Kim	NLP	55	Delhi	6000^$0	10+

In [29]:
```
emp['Location']=emp['Location'].str.replace(r'\W','',regex=True)
```

In [30]:
```
emp['Location']
```

Out[30]:
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [31]:
```
emp['Salary']=emp['Salary'].str.replace(r'\W','',regex=True)
```

In [32]:
```
emp['Salary']
```

Out[32]:
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

In [33]:
```
emp['Exp']=emp['Exp'].str.extract(r'(\d+)')
```

In [34]:
```
emp['Exp']
```

Out[34]:
```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

In [35]:
```
emp
```

Out[35]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Umar	Dataanalyst	NaN	NaN	15000	4
3	Jane	Analytics	NaN	Hyderbad	20000	NaN
4	Uttam	Statistics	67	NaN	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [40]:
```
clean_data = emp.copy()
```

In [42]:
```
clean_data
```

Out[42]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Umar	Dataanalyst	NaN	NaN	15000	4
3	Jane	Analytics	NaN	Hyderbad	20000	NaN
4	Uttam	Statistics	67	NaN	30000	5
5	Kim	NLP	55	Delhi	60000	10

lets apply EDA techniques

step 1

- missing value treatment

In [44]:
```
clean_data.isnull().sum()
```

Out[44]:
```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

In [48]:
```
clean_data['Age']
```

Out[48]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [50]:
```
import numpy as np
```

In [54]:
```
clean_data['Age']=clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['Age'])))
```

In [56]:
```
clean_data['Age']
```

```
Out[56]: 0       34
         1       45
         2    50.25
         3    50.25
         4       67
         5       55
         Name: Age, dtype: object
```

```
In [58]: clean_data['Exp']
```

```
Out[58]: 0      2
         1      3
         2      4
         3    NaN
         4      5
         5     10
         Name: Exp, dtype: object
```

```
In [60]: clean_data['Exp']=clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['Exp'])))
```

```
In [62]: clean_data['Exp']
```

```
Out[62]: 0      2
         1      3
         2      4
         3    4.8
         4      5
         5     10
         Name: Exp, dtype: object
```

```
In [64]: clean_data.isnull().sum()
```

```
Out[64]: Name        0
         Domain      0
         Age         0
         Location    2
         Salary      0
         Exp         0
         dtype: int64
```

```
In [66]: clean_data['Location']
```

```
Out[66]: 0       Mumbai
         1    Bangalore
         2          NaN
         3     Hyderbad
         4          NaN
         5        Delhi
         Name: Location, dtype: object
```

```
In [70]: clean_data['Location']=clean_data['Location'].fillna(clean_data['Location'].mode()[0])
```

```
In [72]: clean_data['Location']
```

```
Out[72]: 0       Mumbai
         1    Bangalore
         2    Bangalore
         3     Hyderbad
         4    Bangalore
         5        Delhi
         Name: Location, dtype: object
```

```
In [74]: clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Name      6 non-null      object
 1   Domain    6 non-null      object
 2   Age       6 non-null      object
 3   Location  6 non-null      object
 4   Salary    6 non-null      object
 5   Exp       6 non-null      object
dtypes: object(6)
memory usage: 420.0+ bytes
```

```
In [79]: clean_data['Age']=clean_data['Age'].astype(int)
```

```
In [81]: clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Name      6 non-null      object
 1   Domain    6 non-null      object
 2   Age       6 non-null      int32 
 3   Location  6 non-null      object
 4   Salary    6 non-null      object
 5   Exp       6 non-null      object
dtypes: int32(1), object(5)
memory usage: 396.0+ bytes
```

In [83]:
```
clean_data['Salary']=clean_data['Salary'].astype(int)
```

In [85]:
```
clean_data['Exp']=clean_data['Exp'].astype(int)
```

In [89]:
```
clean_data['Name']=clean_data['Name'].astype('category')
```

In [91]:
```
clean_data['Domain']=clean_data['Domain'].astype('category')
```

In [93]:
```
clean_data['Location']=clean_data['Location'].astype('category')
```

In [95]:
```
clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype   
---  ------    --------------  -----   
 0   Name      6 non-null      category
 1   Domain    6 non-null      category
 2   Age       6 non-null      int32   
 3   Location  6 non-null      category
 4   Salary    6 non-null      int32   
 5   Exp       6 non-null      int32   
dtypes: category(3), int32(3)
memory usage: 866.0 bytes
```

In [97]:
```
clean_data
```

Out[97]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Umar	Dataanalyst	50	Bangalore	15000	4
3	Jane	Analytics	50	Hyderbad	20000	4
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [99]:

```
clean_data.to_csv('clean_data.csv')
```

In [101…

```
import os
os.getcwd()
```

Out[101…

```
'C:\\Users\\jayes'
```

step 2

- univariate analysis

In [103…

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [106…

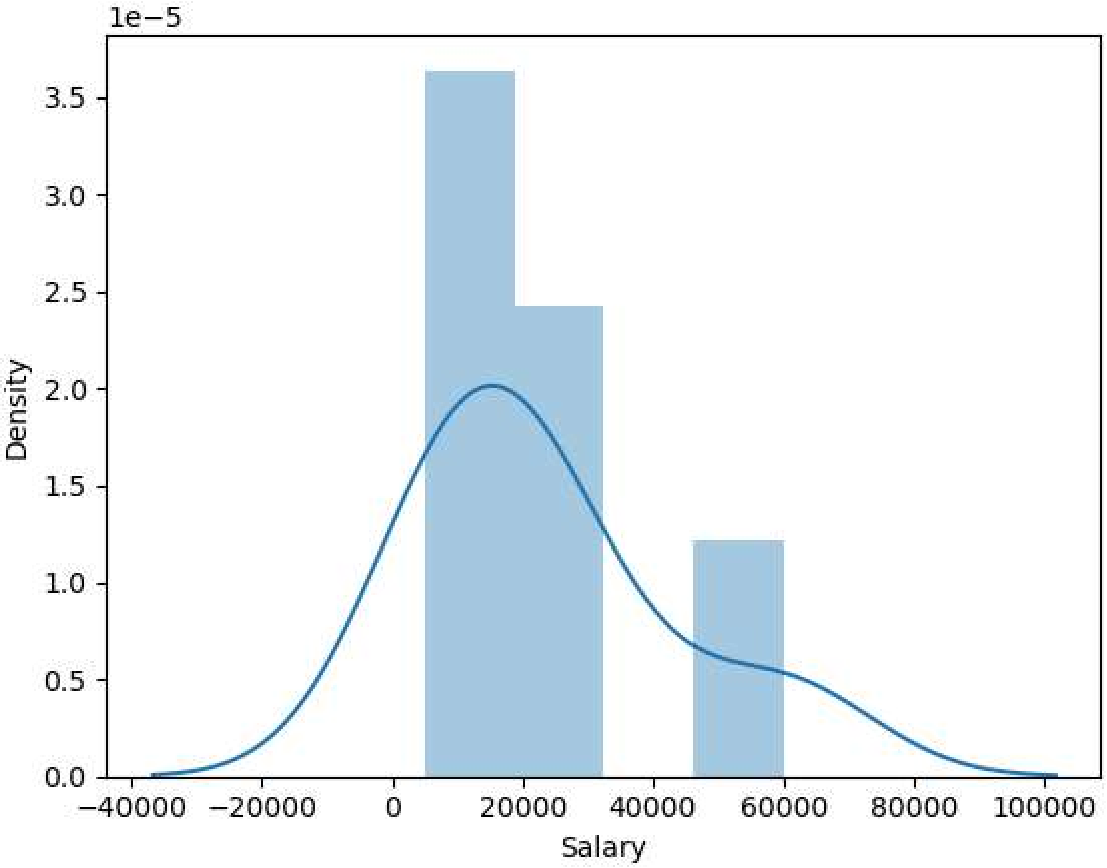
```
import warnings
warnings.filterwarnings('ignore')
```

In [108…

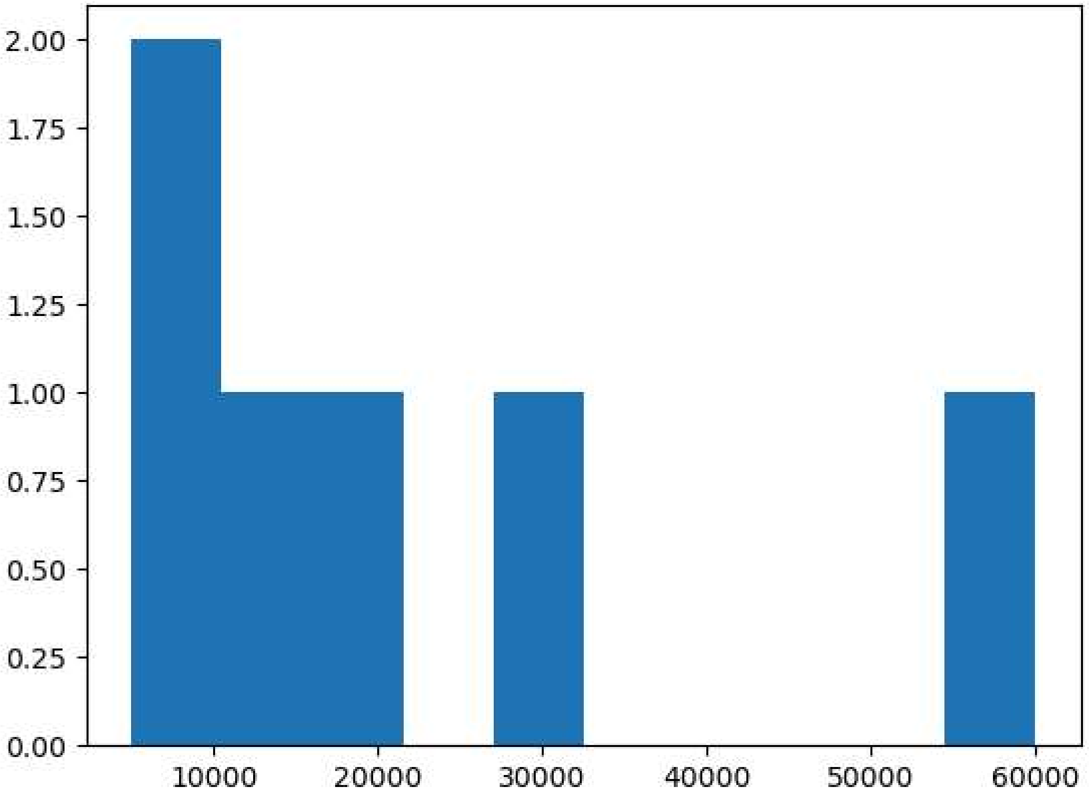
```
clean_data['Salary']
```

```
Out[108…  0     5000
          1    10000
          2    15000
          3    20000
          4    30000
          5    60000
          Name: Salary, dtype: int32
```

```
In [110…  vis1 = sns.distplot(clean_data['Salary'])
```



```
In [112…  vis2 = plt.hist(clean_data['Salary'])
```



In [117…

```
clean_data
```

Out[117…

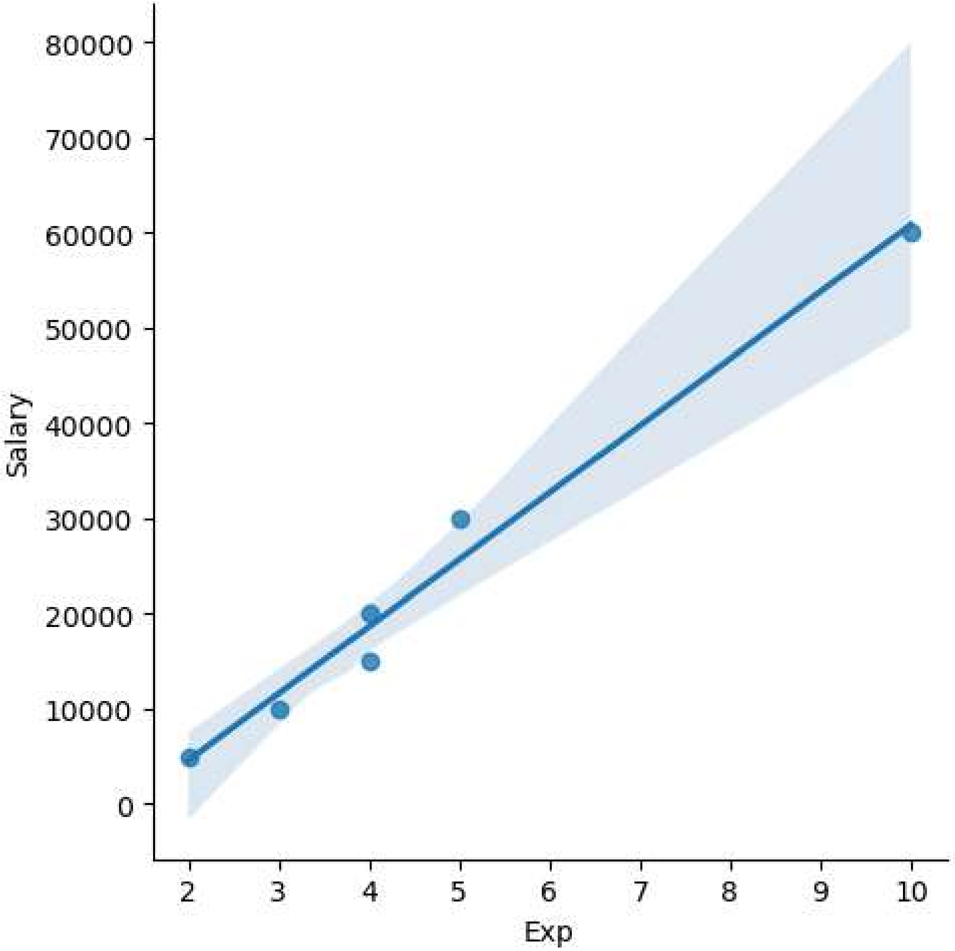
	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Umar	Dataanalyst	50	Bangalore	15000	4
3	Jane	Analytics	50	Hyderbad	20000	4
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	10

step 3 and step 4

- bivariate analysis
- outlier detection

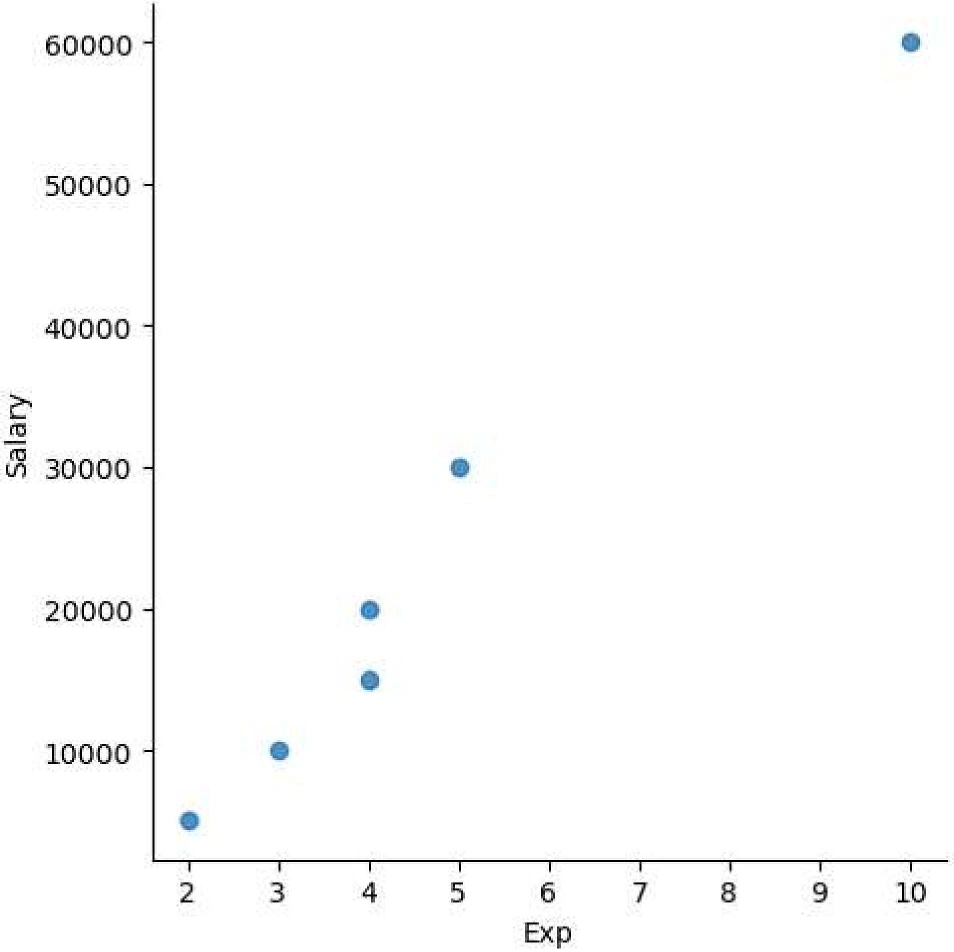
In [122…

```
vis2 = sns.lmplot(data = clean_data, x = 'Exp', y = 'Salary')
```



In [124…

```
vis3 = sns.lmplot(data = clean_data, x = 'Exp', y = 'Salary', fit_reg = False)
```



step 5

- variable creation

In [126…

```
clean_data
```

Out[126…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Umar	Dataanalyst	50	Bangalore	15000	4
3	Jane	Analytics	50	Hyderbad	20000	4
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [130…

```
X_iv = clean_data[['Name','Domain','Age','Location','Exp']]
```

In [132…

```
X_iv
```

Out[132…

	Name	Domain	Age	Location	Exp
0	Mike	Datascience	34	Mumbai	2
1	Teddy	Testing	45	Bangalore	3
2	Umar	Dataanalyst	50	Bangalore	4
3	Jane	Analytics	50	Hyderbad	4
4	Uttam	Statistics	67	Bangalore	5
5	Kim	NLP	55	Delhi	10

In [134…

```
Y_dv = clean_data['Salary']
```

In [136…

```
Y_dv
```

Out[136…

```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: int32
```

step 6 and step 7

- imputation
- variable creation for using in machine learning

In [140…

```
imputation = pd.get_dummies(clean_data,dtype=int)
```

In [142…

```
imputation
```

Out[142…

	Age	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Umar	Name_Uttam	Domain_Analytics	Domain
0	34	5000	2	0	0	1	0	0	0	0	
1	45	10000	3	0	0	0	1	0	0	0	
2	50	15000	4	0	0	0	0	1	0	0	
3	50	20000	4	1	0	0	0	0	0	1	
4	67	30000	5	0	0	0	0	0	1	0	
5	55	60000	10	0	1	0	0	0	0	0	

In [144…

```
len(clean_data.columns)
```

Out[144…

```
6
```

In [146…

```
len(imputation.columns)
```

Out[146…

```
19
```