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

In [3]:
```
pd.__version__
```

Out[3]:
```
'2.2.2'
```

In [5]:
```
emp_data=pd.read_excel(r"C:\Users\Arabinda\Downloads\Rawdata.xlsx")
```

In [7]:
```
emp_data
```

Out[7]:

	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
5	Kim	NLP	55yr	Delhi	6000^$0	10+

In [9]:
```
id(emp_data)
```

Out[9]:
```
1343822704672
```

In [13]:
```
emp_data.columns
```

Out[13]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [15]:
```
emp_data.shape
```

Out[15]:
```
(6, 6)
```

In [17]:
```
emp_data.isnull().any().any()
```

Out[17]:
```
True
```

In [19]:
```
emp_data.head()
```

Out[19]:

	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 [23]:
```
emp_data.tail()
```

Out[23]:

	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 [25]:
```
emp_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       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 [27]:
```
emp_data.isnull().sum()
```

Out[27]:
```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

Data Cleaning or Data Cleanging

In [30]:
```
emp_data["Name"]
```

Out[30]:
```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

In [36]:
```
#Eliminate regex '\W' meaning it denotes extra character
emp_data["Name"]=emp_data["Name"].str.replace(r'\W','',regex=True)
```

In [38]:
```
emp_data['Name']
```

Out[38]:
```
0     Mike
1    Teddy
2     Umar
3     Jane
4    Uttam
5      Kim
Name: Name, dtype: object
```

In [40]:
```
#Eliminate regex '\W' meaning it denotes extra character
emp_data["Domain"]=emp_data["Domain"].str.replace(r'\W','',regex=True)
```

In [42]:
```
emp_data['Domain']
```

Out[42]:
```
0    Datascience
1        Testing
2    Dataanalyst
3      Analytics
4     Statistics
5            NLP
Name: Domain, dtype: object
```

In [44]:
```
emp_data
```

Out[44]:

	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 [46]:
```
#Eliminate regex '\W' meaning it denotes extra character
emp_data["Age"]=emp_data["Age"].str.replace(r'\W','',regex=True)
```

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

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

In [62]:
```
emp_data["Age"]=emp_data["Age"].str.extract('(\\d+)')  # Clean all the character ex
```

In [64]:
```
emp_data["Age"]
```

Out[64]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [72]:
```
emp_data["Location"]=emp_data["Location"].str.replace('\W',"",regex=True)
```

```
<>:1: SyntaxWarning: invalid escape sequence '\W'
<>:1: SyntaxWarning: invalid escape sequence '\W'
C:\Users\Arabinda\AppData\Local\Temp\ipykernel_6924\1648145934.py:1: SyntaxWarning:
invalid escape sequence '\W'
  emp_data["Location"]=emp_data["Location"].str.replace('\W',"",regex=True)
```

In [68]:
```
emp_data["Salary"]=emp_data["Salary"].str.replace(r'\W','',regex=True)
```

In [74]:
```
emp_data
```

Out[74]:

	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> yrs
3	Jane	Analytics	NaN	Hyderbad	20000	NaN
4	Uttam	Statistics	67	NaN	30000	5+ year
5	Kim	NLP	55	Delhi	60000	10+

In [76]:
```
emp_data["Exp"]=emp_data["Exp"].str.extract('(\\d+)')  #remove all character
```

In [78]:
```
emp_data
```

Out[78]:

	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 [80]:
```
clean_data=emp_data.copy()
```

Apply EDA Techniques

Missing Value Treatment

In [84]:
```
clean_data.isnull().sum()
```

Out[84]:
```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

In [86]:
```
clean_data['Age']
```

Out[86]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [88]:
```
import numpy as np
```

In [90]:
```
clean_data['Age']=clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['Age'])
```

In [92]:
```
clean_data['Age']
```

Out[92]:
```
0       34
1       45
2    50.25
3    50.25
4       67
5       55
Name: Age, dtype: object
```

In [94]:
```
clean_data['Exp']
```

Out[94]:
```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

In [96]:
```
clean_data['Exp']=clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['Exp']))
```

In [98]:
```
clean_data['Exp']
```

Out[98]:
```
0      2
1      3
2      4
3    4.8
4      5
5     10
Name: Exp, dtype: object
```

In [100…
```
clean_data
```

Out[100…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Umar	Dataanalyst	50.25	NaN	15000	4
3	Jane	Analytics	50.25	Hyderbad	20000	4.8
4	Uttam	Statistics	67	NaN	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [102…
```
clean_data['Location']
```

Out[102…
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [108…
```
clean_data['Location']=clean_data['Location'].fillna(clean_data['Location'].mode()[
```

In [110…
```
clean_data['Location']
```

Out[110…
```
0       Mumbai
1    Bangalore
2    Bangalore
3     Hyderbad
4    Bangalore
5        Delhi
Name: Location, dtype: object
```

In [112…
```
clean_data
```

Out[112…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Umar	Dataanalyst	50.25	Bangalore	15000	4
3	Jane	Analytics	50.25	Hyderbad	20000	4.8
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [114…
```
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 [116…
```
clean_data['Age']=clean_data['Age'].astype(int)   #convert System data type to user
```

In [118…
```
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 [120…
```
clean_data['Exp']=clean_data['Exp'].astype(int)
```

In [122…
```
clean_data['Salary']=clean_data['Salary'].astype(int)
```

In [124…
```
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      int32 
 5   Exp       6 non-null      int32 
dtypes: int32(3), object(3)
memory usage: 348.0+ bytes
```

In [130…
```
clean_data['Name']=clean_data['Name'].astype('category')
clean_data['Domain']=clean_data['Domain'].astype('category')
clean_data['Location']=clean_data['Location'].astype('category')
```

In [132…
```
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 [134…
```
clean_data
```

Out[134…

	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 [136…
```
# import data from jupyter note book  to excel file
clean_data.to_csv('clean_data.csv')
```

In [138…
```
import os
os.getcwd()
```

Out[138… 'C:\\Users\\Arabinda'

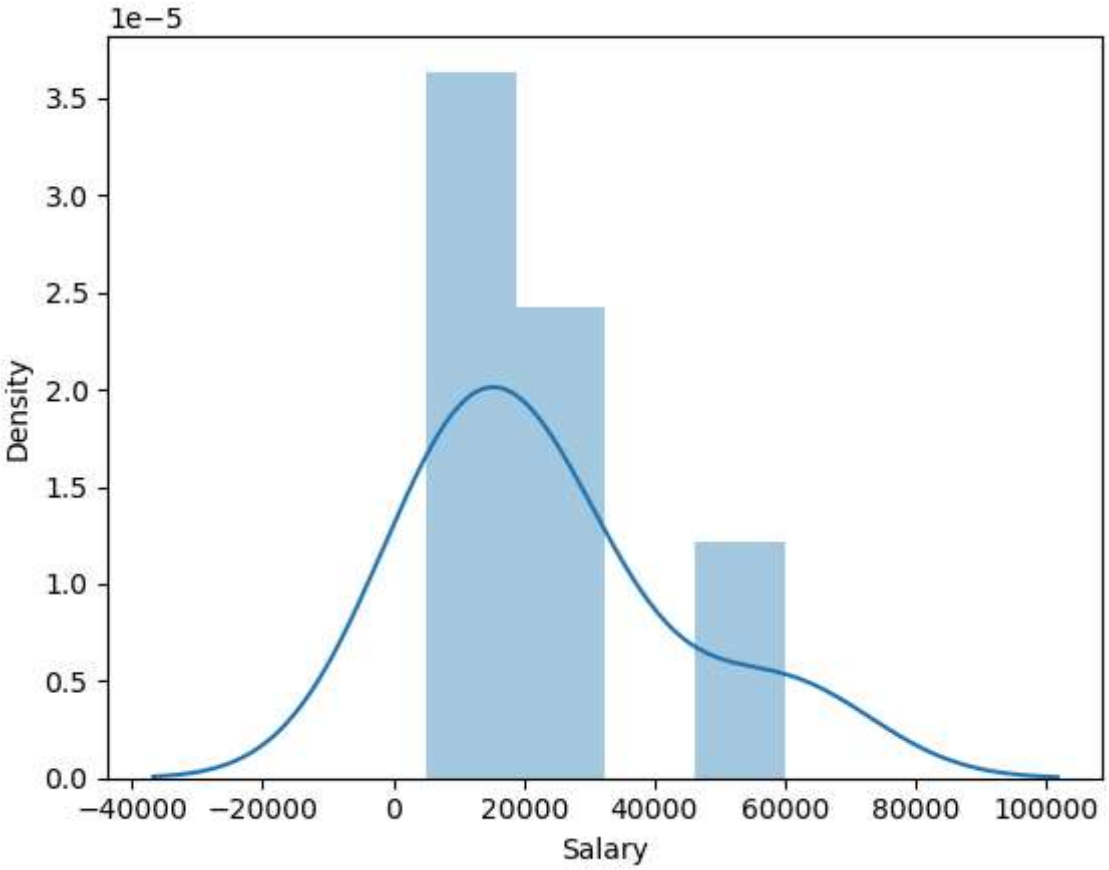
In [144…
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [146…
```
import warnings
warnings.filterwarnings('ignore')
```

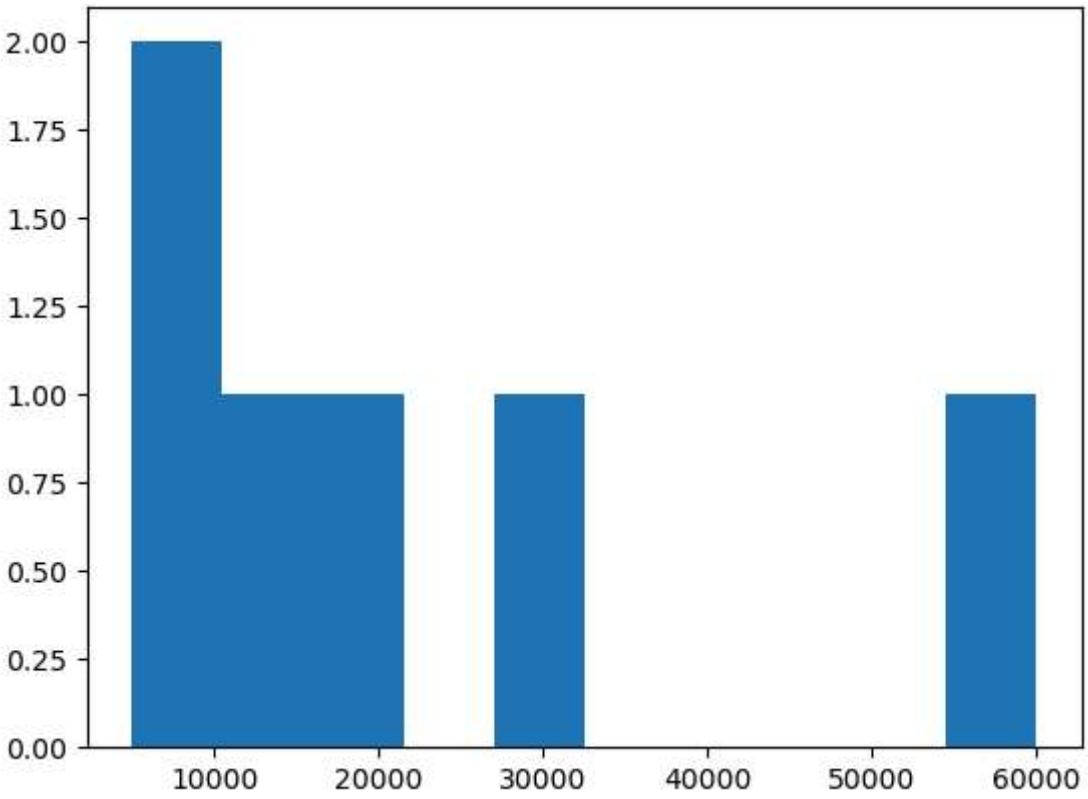
In [148…
```
clean_data['Salary']
```

Out[148…
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: int32
```

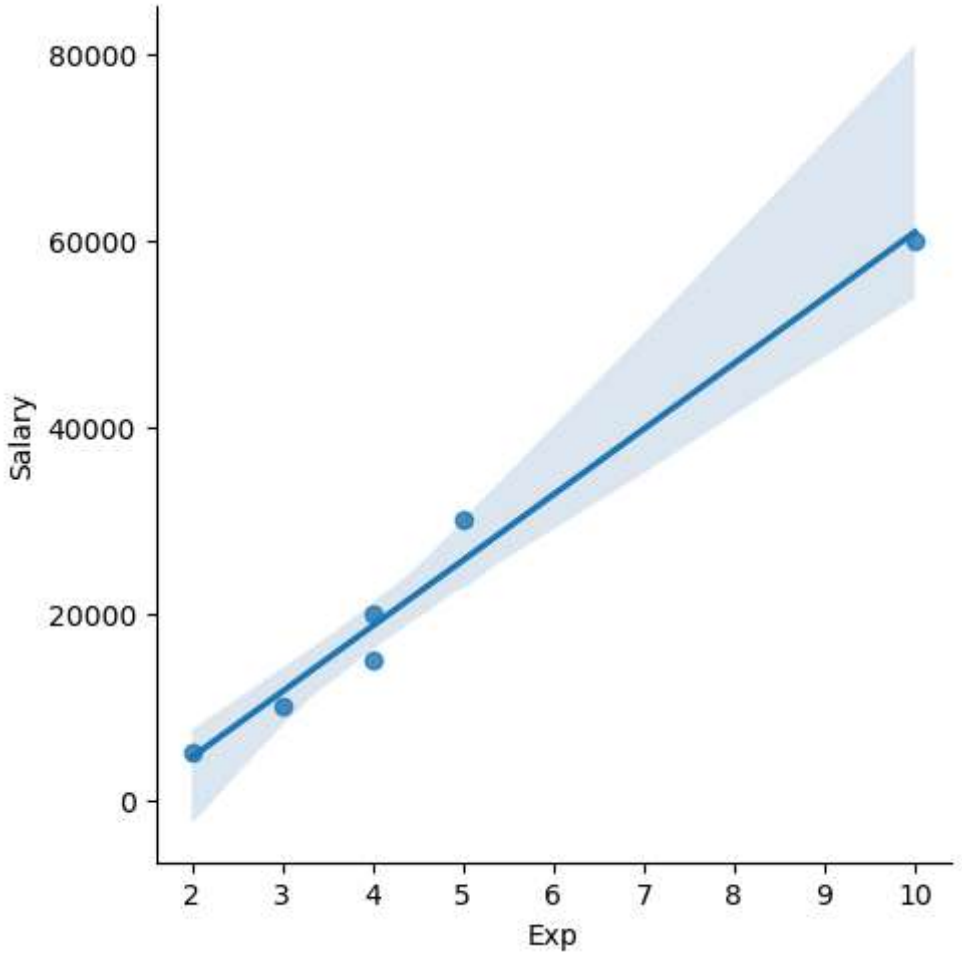
In [150…
```
vis1=sns.distplot(clean_data['Salary'])
```



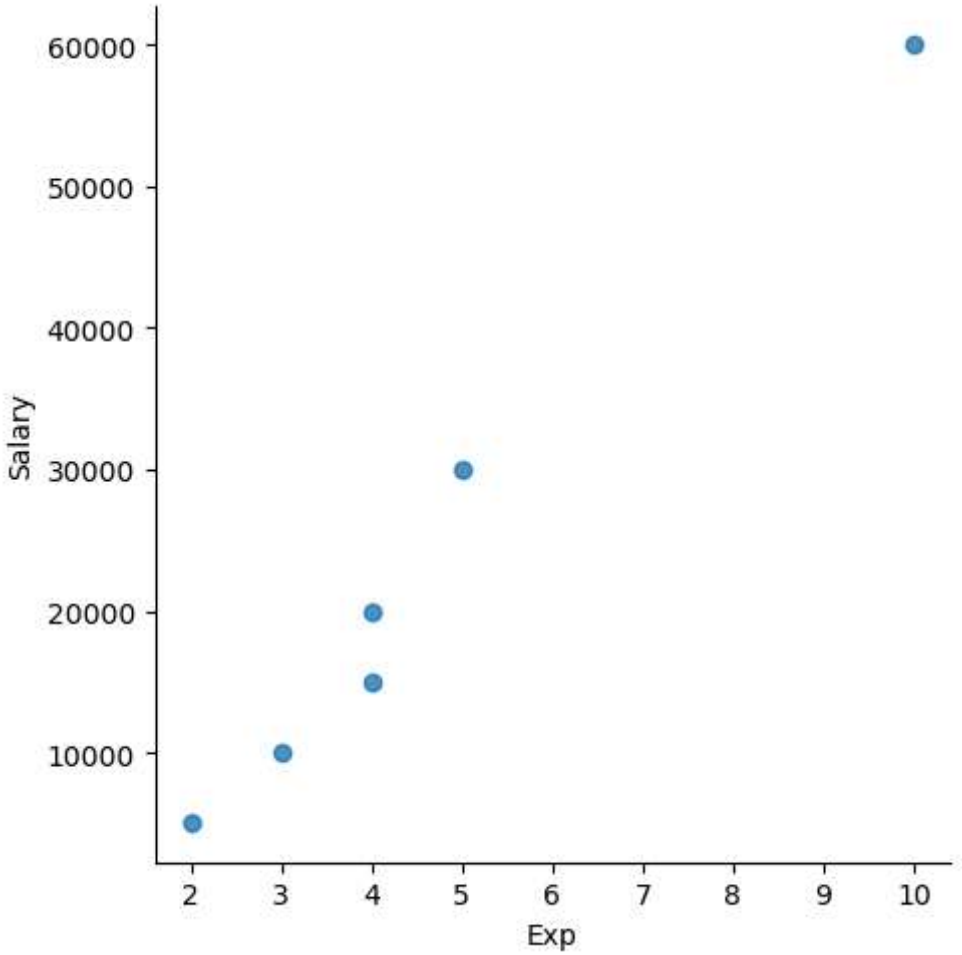
In [152…
```
vis2=plt.hist(clean_data['Salary'])
```



In [156…
```
vis3=sns.lmplot(data=clean_data,x='Exp',y='Salary')
```



In [158…
```
vis4=sns.lmplot(data=clean_data,x='Exp',y='Salary',fit_reg=False)
```



In [160…
```
clean_data[0:6:2]
```

Out[160…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
2	Umar	Dataanalyst	50	Bangalore	15000	4
4	Uttam	Statistics	67	Bangalore	30000	5

In [162…
```
clean_data[::-1]
```

Out[162…

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

In [164…
```
clean_data.columns
```

Out[164…
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [166…
```
X_indepVar=clean_data[['Name', 'Domain', 'Age', 'Location', 'Exp']]
```

In [170…
```
X_DepVar=clean_data[['Salary']]
```

In [172…
```
X_indepVar
```

Out[172…

	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 [174…
```
X_DepVar
```

Out[174…

	Salary
0	5000
1	10000
2	15000
3	20000
4	30000
5	60000

In [176…
```
clean_data
```

Out[176…

	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 [180…
```
imputation=pd.get_dummies(clean_data)   #Using Dummy variable Techniques
imputation
```

Out[180…

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

In [ ]: