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

In [2]:
```
%matplotlib inline
```

In [3]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
data = pd.read_excel(r'D:\AI Course Naresh\2-20-2025\Rawdata.xlsx')
```

In [5]:
```
data
```

Out[5]:

	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 [6]:
```
data.head()
```

Out[6]:

	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 [7]:
```
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 [8]:
```
data.size
```

Out[8]: 36

In [9]:
```
data.columns
```

Out[9]: Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')

In [10]:
```
data.isna()
```

Out[10]:

	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 [11]:
```
data.isnull()
```

Out[11]:

	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 [12]:
```
data.isna().any()
```

Out[12]:
```
Name        False
Domain      False
Age          True
Location     True
Salary      False
Exp          True
dtype: bool
```

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

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

In [14]:
```
data['Name']
```

Out[14]:
```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

In [15]:
```
data['Name'] = data['Name'].str.replace(r'\W','', regex = True)
```

In [16]:
```
data['Name']
```

Out[16]:
```
0     Mike
1    Teddy
2     Umar
3     Jane
4    Uttam
5      Kim
Name: Name, dtype: object
```

In [17]:
```
data['Domain'] = data['Domain'].str.replace(r'\W','', regex= True)
```

In [18]:
```
data['Domain']
```

Out[18]:
```
0    Datascience
1        Testing
2    Dataanalyst
3      Analytics
4     Statistics
5            NLP
Name: Domain, dtype: object
```

In [19]:
```
data
```

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	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 [20]:
```
data['Location'] = data['Location'].str.replace(r'\W','', regex = True)
```

In [21]:
```
data['Location']
```

Out[21]:
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [22]:
```
data['Salary'] = data['Salary'].str.replace(r'\W','', regex= True)
```

In [23]:
```
data['Salary']
```

Out[23]:
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

In [24]:
```
data
```

Out[24]:

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

In [25]:
```
data['Age'] = data['Age'].str.replace(r'\W','', regex = True)
```

In [26]:
```
data['Age'] = data['Age'].str.extract('(\d+)')
```

In [27]:
```
data['Age']
```

Out[27]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [28]:
```
data
```

Out[28]:

	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 [29]:
```
data['Exp'] = data['Exp'].str.extract(r'(\d)')
```

In [30]:
```
data['Exp']
```

Out[30]:
```
0      2
1      3
2      4
3    NaN
4      5
5      1
Name: Exp, dtype: object
```

In [31]:
```
data
```

Out[31]:

	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	1

In [32]:
```
clean_data = data.copy()
```

till now we have raw data we use regex to clean the data and removed all noise characted from the dataset

you can also work in same things in sql query as well

In [34]:
```
clean_data
```

Out[34]:

	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	1

In [35]:
```
import numpy as np
```

In [36]:
```
clean_data['Age'] = clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['Age'
```

In [37]:
```
clean_data['Age']
```

Out[37]:
```
0       34
1       45
2    50.25
3    50.25
4       67
5       55
Name: Age, dtype: object
```

In [38]:
```
clean_data['Age'] = clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['Age'
```

In [39]:
```
clean_data['Exp'] = clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['Exp'
```

In [40]:
```
clean_data['Exp']
```

Out[40]:
```
0      2
1      3
2      4
3    3.0
4      5
5      1
Name: Exp, dtype: object
```

In [41]:
```
clean_data['Location'] = clean_data['Location'].fillna(np.mean(pd.to_numeric(clean_
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
File lib.pyx:2391, in pandas._libs.lib.maybe_convert_numeric()

ValueError: Unable to parse string "Mumbai"

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
Cell In[41], line 1
----> 1 clean_data['Location'] = clean_data['Location'].fillna(np.mean(pd.to_numeric
(clean_data['Location'])))

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\tools\numeric.py:232, in
to_numeric(arg, errors, downcast, dtype_backend)
    230 coerce_numeric = errors not in ("ignore", "raise")
    231 try:
--> 232     values, new_mask = lib.maybe_convert_numeric(  # type: ignore[call-overl
oad]
    233         values,
    234         set(),
    235         coerce_numeric=coerce_numeric,
    236         convert_to_masked_nullable=dtype_backend is not lib.no_default
    237         or isinstance(values_dtype, StringDtype)
    238         and not values_dtype.storage == "pyarrow_numpy",
    239     )
    240 except (ValueError, TypeError):
    241     if errors == "raise":

File lib.pyx:2433, in pandas._libs.lib.maybe_convert_numeric()

ValueError: Unable to parse string "Mumbai" at position 0
```

In [44]:
```
clean_data['Location'] = clean_data['Location'].fillna(clean_data['Location'].mode(
```

In [46]:
```
clean_data['Location']
```

Out[46]:
```
0       Mumbai
1    Bangalore
2    Bangalore
3     Hyderbad
4    Bangalore
5        Delhi
Name: Location, dtype: object
```

In [50]:
```
clean_data
```

Out[50]:

	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	3.0
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	1

In [52]:
```
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 [54]:
```
clean_data['Age'] = clean_data['Age'].astype(int)
```

In [64]:
```
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 [58]:
```
clean_data['Salary'] = clean_data['Salary'].astype(int)
clean_data['Exp'] = clean_data['Exp'].astype(int)
```

In [62]:
```
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 [66]:
```
clean_data
```

Out[66]:

	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	3
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	1

In [68]:
```
clean_data.to_csv('clean_data_2.csv')
```

In [70]:
```
import os
os.getcwd()
```

Out[70]:
```
'C:\\Users\\HOME\\Working folder 2'
```

In [72]:
```
clean_data
```

Out[72]:

	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	3
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	1

EDA TECHNIQUE LETS APPLY

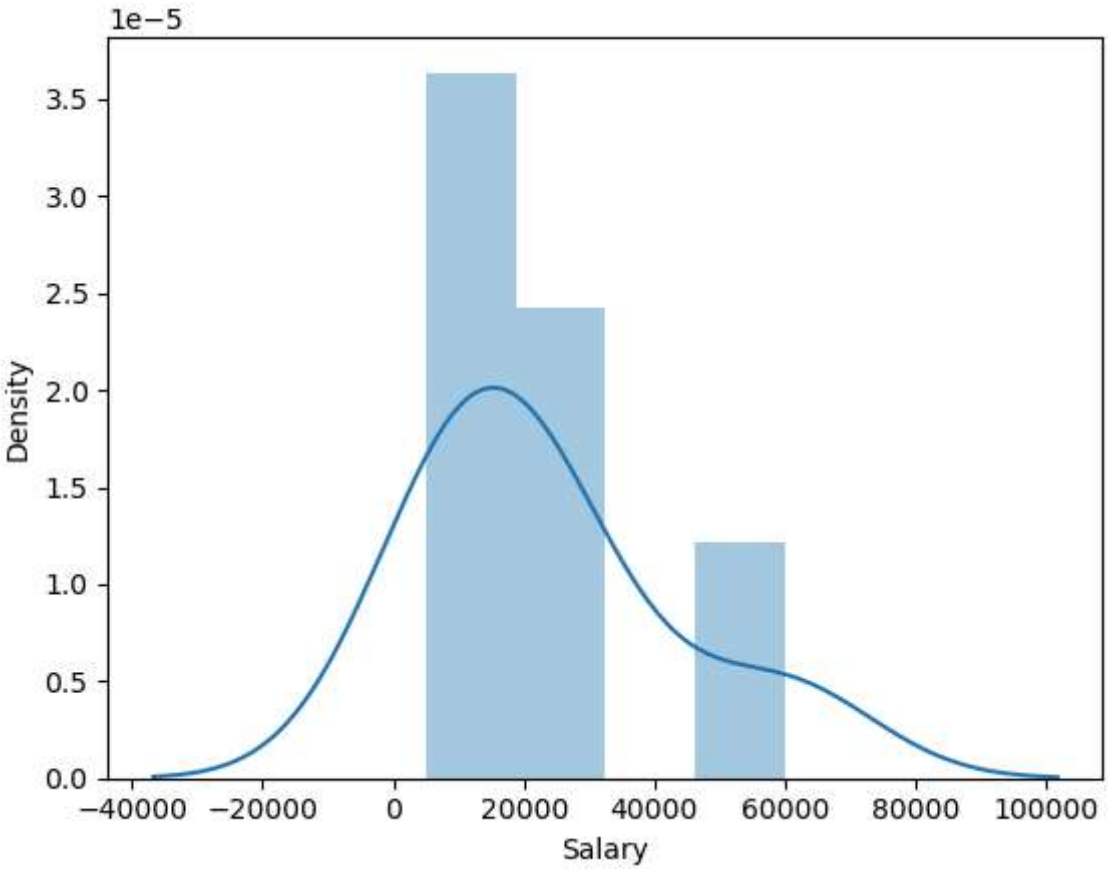
In [75]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [77]:
```
import warnings
warnings.filterwarnings('ignore')
```

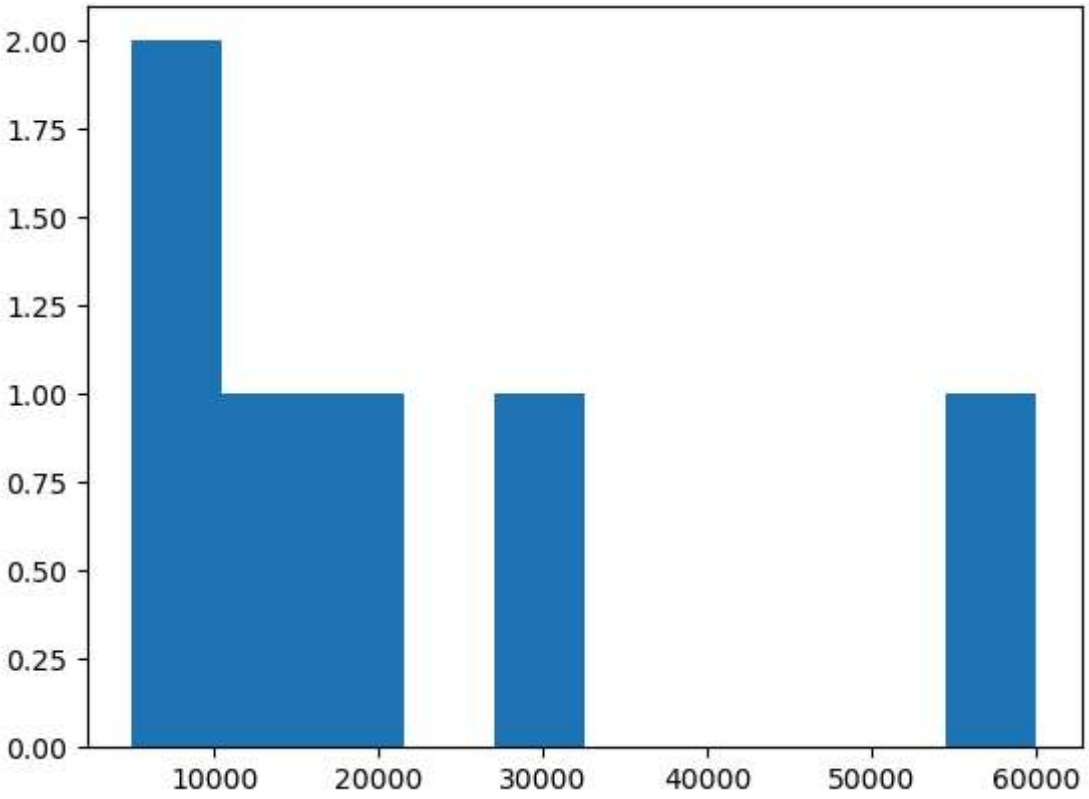
In [79]:
```
clean_data['Salary']
```

Out[79]:
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: int32
```

In [90]:
```
vis1 = sns.distplot(clean_data['Salary'])
plt.show()
```

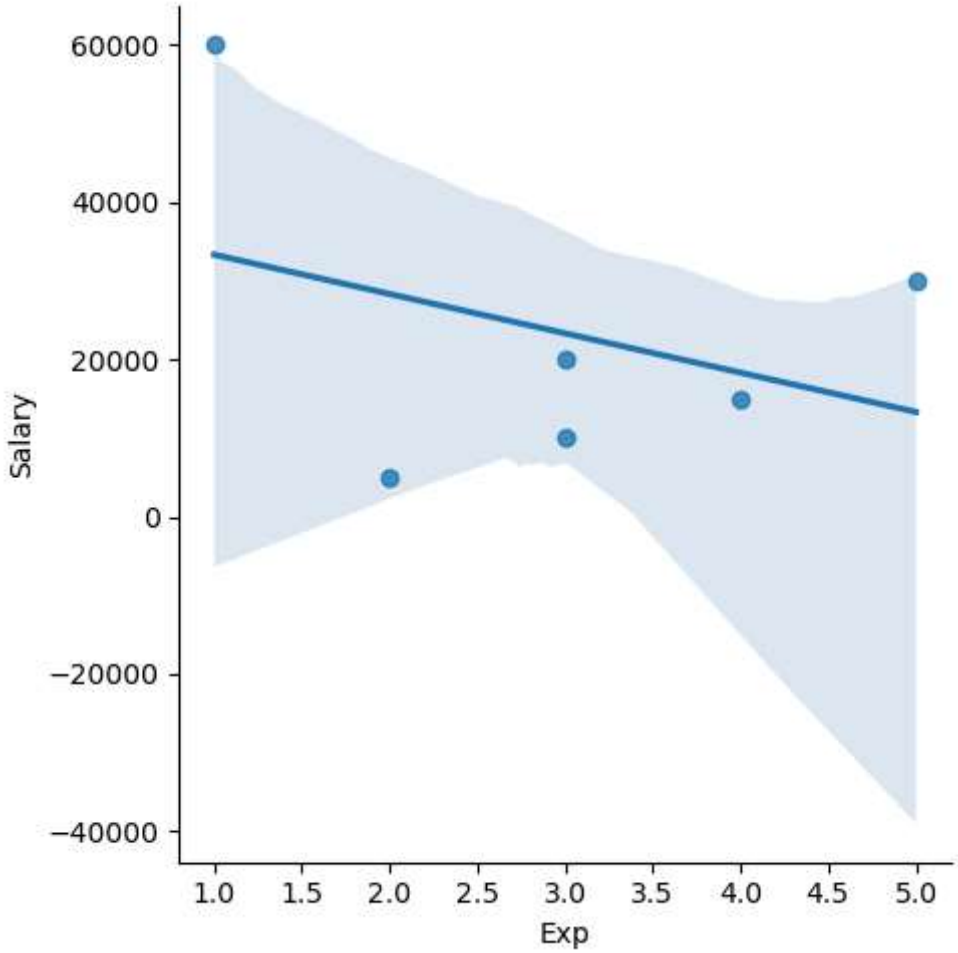


In [96]:
```
vis2 = plt.hist(clean_data['Salary'])
plt.show()
```



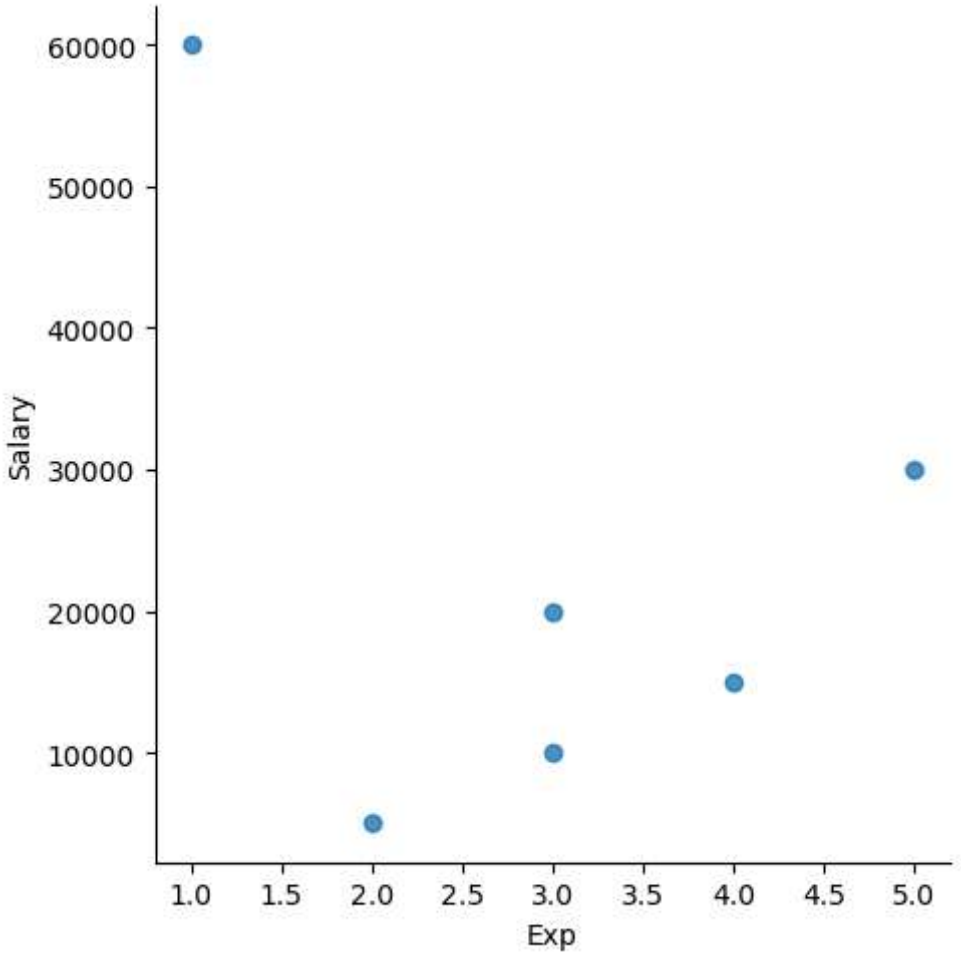
In [100…

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



In [108…

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



In [110…
```
clean_data[:]
```

Out[110…

	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	3
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	1

In [114…
```
clean_data[0:5:2]
```

Out[114…

	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 [116…
```
clean_data[::-1]
```

Out[116…

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

In [118…
```
clean_data.columns
```

Out[118…
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [124…
```
X_iv = clean_data[['Name', 'Domain', 'Age', 'Location', 'Exp']]
```

In [126…
```
X_iv
```

Out[126…

	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	3
4	Uttam	Statistics	67	Bangalore	5
5	Kim	NLP	55	Delhi	1

In [128…
```
y_dv = clean_data[['Salary']]
```

In [130…
```
y_dv
```

Out[130…

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

In [132…
```
clean_data
```

Out[132…

	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	3
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	1

In [134…
```
imputation123 = pd.get_dummies(clean_data)
```

In [136…
```
imputation123
```

Out[136…

	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	3	True	False	False	False	False	
4	67	30000	5	False	False	False	False	False	
5	55	60000	1	False	True	False	False	False	

In [144…
```
imputation = pd.get_dummies(clean_data, dtype = int)
imputation
```

Out[144…

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

In [138…
```
clean_data
```

Out[138…

	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	3
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	1

In [140…
```
imputation123
```

Out[140…

	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	3	True	False	False	False	False	
4	67	30000	5	False	False	False	False	False	
5	55	60000	1	False	True	False	False	False	

In [146…
```
len(clean_data)
```

Out[146… 6

In [148…
```
imputation.columns
```

Out[148…
```
Index(['Age', 'Salary', 'Exp', 'Name_Jane', 'Name_Kim', 'Name_Mike',
       'Name_Teddy', 'Name_Umar', 'Name_Uttam', 'Domain_Analytics',
       'Domain_Dataanalyst', 'Domain_Datascience', 'Domain_NLP',
       'Domain_Statistics', 'Domain_Testing', 'Location_Bangalore',
       'Location_Delhi', 'Location_Hyderbad', 'Location_Mumbai'],
      dtype='object')
```

In [152…
```
len(imputation.columns)
```

Out[152… 19

In [ ]: