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

In [2]:
```
emp = pd.read_excel(r'D:\EDA\Rawdata.xlsx')
```

In [3]:
```
emp
```

Out[3]:

	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 [4]:
```
emp['Name']
```

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

In [5]:
```
emp['Name'] = emp['Name'].str.replace(r'\W','')
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_15672\389424325.py:1: FutureWarning: The default value of regex will c
hange from True to False in a future version.
  emp['Name'] = emp['Name'].str.replace(r'\W','')
```

In [6]:
```
emp
```

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	Umar	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 [7]:
```
emp['Domain'] = emp['Domain'].str.replace(r'\W','')
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_15672\2360087947.py:1: FutureWarning: The default value of regex will
change from True to False in a future version.
  emp['Domain'] = emp['Domain'].str.replace(r'\W','')
```

In [8]:
```
emp
```

Out[8]:

	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 [9]:
```
emp['Age'] = emp['Age'].str.replace(r'\W','')
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_15672\3358378917.py:1: FutureWarning: The default value of regex will
change from True to False in a future version.
  emp['Age'] = emp['Age'].str.replace(r'\W','')
```

In [10]:
```
emp
```

Out[10]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34years	Mumbai	5^00#0	2+
1	Teddy	Testing	45yr	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	67yr	NaN	30000-	5+ year
5	Kim	NLP	55yr	Delhi	6000^$0	10+

In [12]:
```
emp['Age']=emp['Age'].str.extract('(\d+)')
```

In [13]:
```
emp
```

Out[13]:

	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 [14]:
```
emp['Location'] = emp['Location'].str.replace(r'\W','')
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_15672\3886403992.py:1: FutureWarning: The default value of regex will
change from True to False in a future version.
  emp['Location'] = emp['Location'].str.replace(r'\W','')
```

In [15]:
```
emp['Salary'] = emp['Salary'].str.replace(r'\W','')
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_15672\1304150360.py:1: FutureWarning: The default value of regex will
change from True to False in a future version.
  emp['Salary'] = emp['Salary'].str.replace(r'\W','')
```

In [16]:
```
emp
```

Out[16]:

	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 [17]:
```
emp['Exp'] = emp['Exp'].str.extract('(\d+)')
```

In [18]:
```
emp
```

Out[18]:

	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 [19]:
```
clean_data = emp.copy()
```

In [20]:
```
clean_data
```

Out[20]:

	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 [21]:
```
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       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 [22]:
```
import numpy as np
```

In [23]:
```
clean_data
```

Out[23]:

	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 [24]:
```
clean_data['Age']
```

Out[24]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [25]:
```
clean_data['Age'] = clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['Age'])))
```

In [26]:
```
clean_data
```

Out[26]:

	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	NaN
4	Uttam	Statistics	67	NaN	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [27]:
```
emp
```

Out[27]:

	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 [28]:
```
clean_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	50.25	NaN	15000	4
3	Jane	Analytics	50.25	Hyderbad	20000	NaN
4	Uttam	Statistics	67	NaN	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [29]:
```
clean_data['Exp']= clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['Exp'])))
```

In [30]:
```
clean_data
```

Out[30]:

	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 [31]:
```
clean_data['Location']= clean_data['Location'].fillna(clean_data['Location'].mode()[0])
```

In [32]:
```
clean_data
```

Out[32]:

	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 [33]:
```
clean_data['Age']= clean_data['Age'].astype(int)
```

In [34]:
```
clean_data['Salary']=clean_data['Salary'].astype(int)
```

In [35]:
```
clean_data['Exp']=clean_data['Exp'].astype(int)
```

In [36]:
```
clean_data
```

Out[36]:

	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 [37]:
```
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 [41]:
```
import os
```

In [46]:
```
os.getcwd()
```

Out[46]:
```
'C:\\Users\\HP'
```

In [47]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

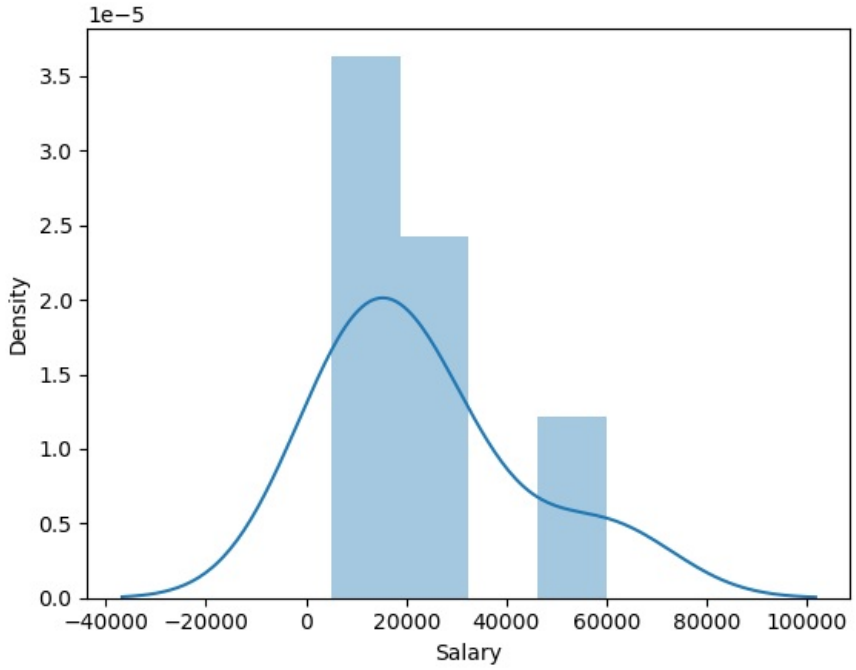
In [48]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [45]:
```
clean_data
```

Out[45]:

	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 [49]:
```
vis1 = sns.distplot(clean_data['Salary'])
```

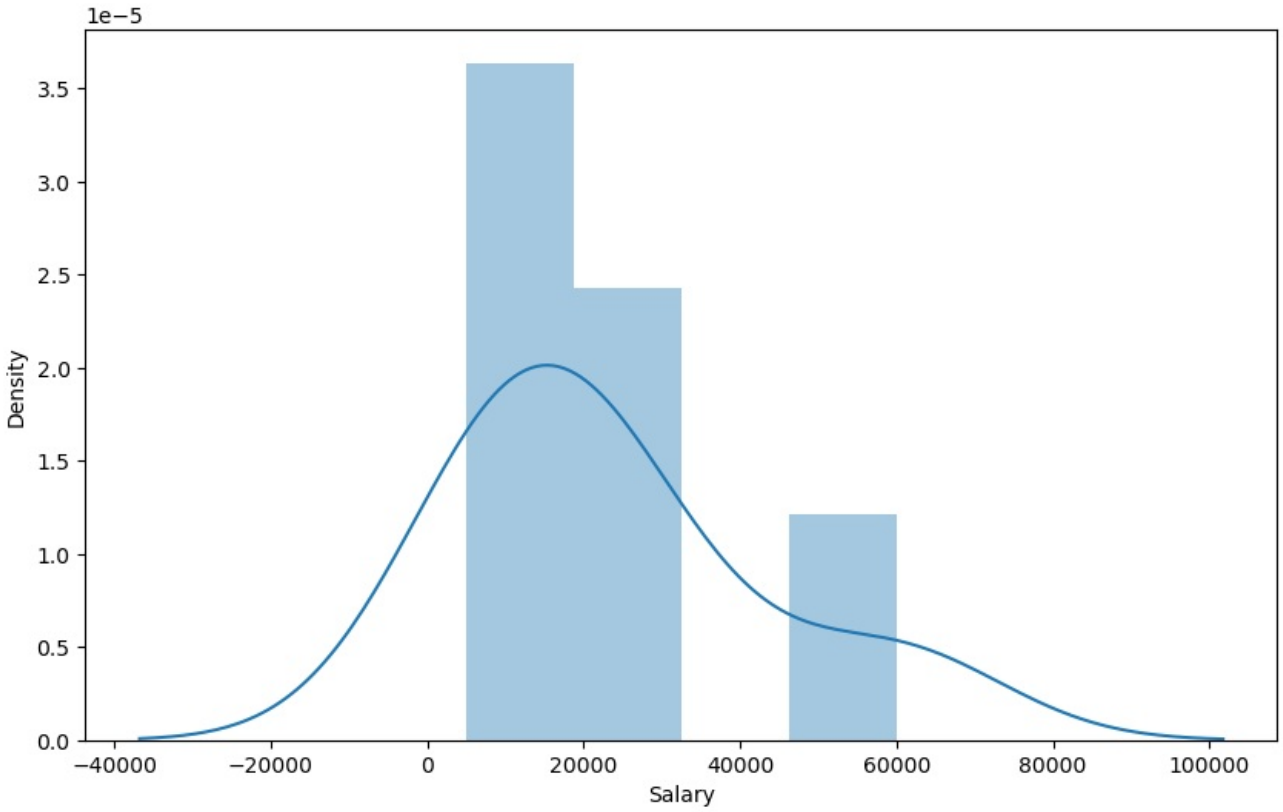


In [50]:
```
plt.rcparams['figure.figsize']=10,6
vis1 = sns.distplot(clean_data['Salary'])
```

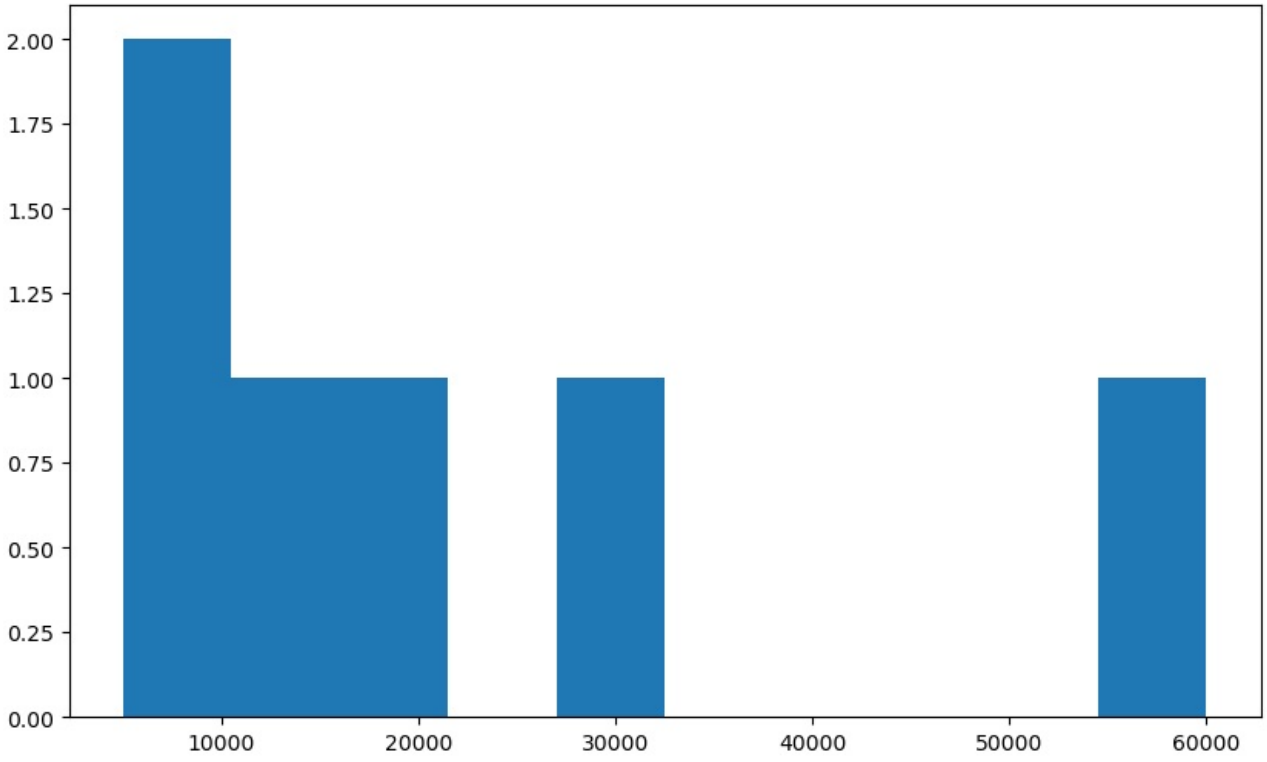
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[50], line 1
----> 1 plt.rcparams['figure.figsize']=10,6
      2 vis1 = sns.distplot(clean_data['Salary'])

AttributeError: module 'matplotlib.pyplot' has no attribute 'rcparams'
```

In [51]:
```
plt.rcParams['figure.figsize'] = 10,6
```

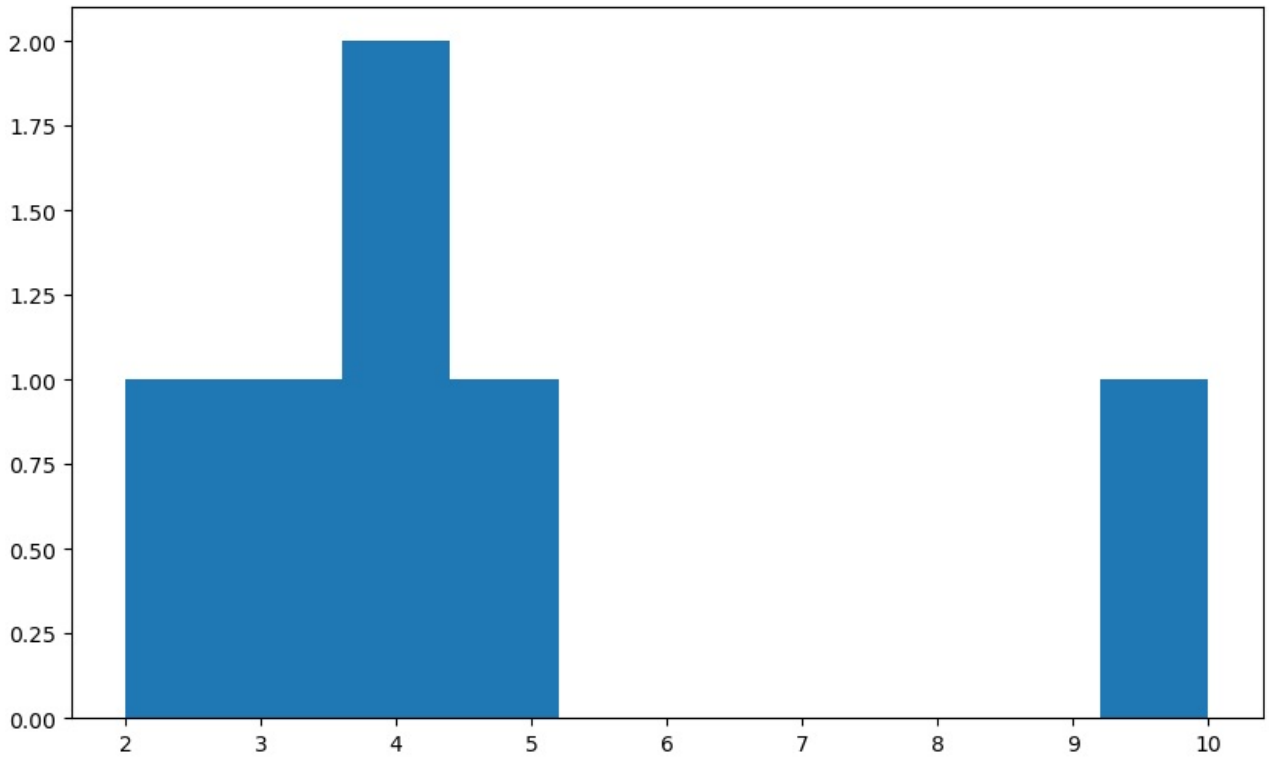
In [52]:
```
vis1 = sns.distplot(clean_data['Salary'])
```



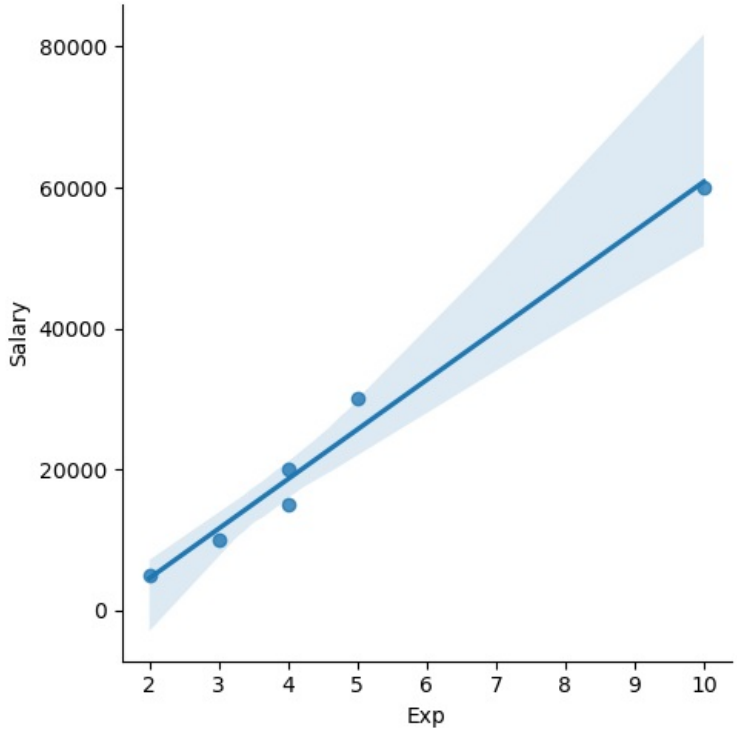
In [53]:
```
vis2 = plt.hist(clean_data['Salary'])
```



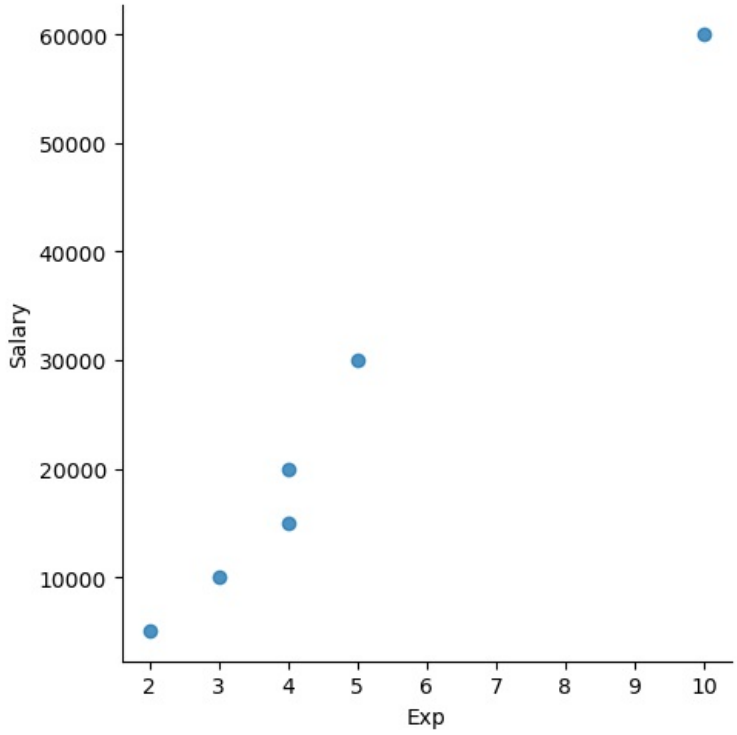
In [54]:
```
vis3 = plt.hist(clean_data['Exp'])
```



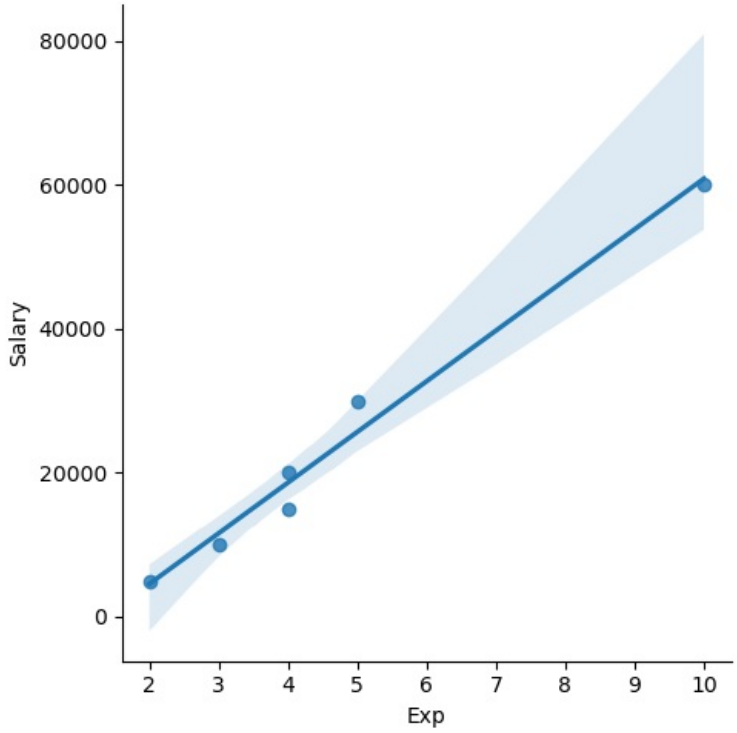
In [55]:
```
vis4 = sns.lmplot(data= clean_data,x='Exp',y='Salary')
```



In [56]:
```
vis5= sns.lmplot(data = clean_data,x="Exp",y="Salary",fit_reg=False)
```



In [57]:
```
vis6 = sns.lmplot(data=clean_data,x="Exp",y="Salary",fit_reg= True)
```



In [58]:
```
clean_data
```

Out[58]:

	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 [59]:
```
clean_data[:]
```

Out[59]:

	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 [60]:
```
clean_data[:2]
```

Out[60]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3

In [61]:
```
clean_data[2:]
```

Out[61]:

	Name	Domain	Age	Location	Salary	Exp
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 [62]:
```
clean_data[0:1]
```

Out[62]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2

In [65]:
```
clean_data(0,3)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[65], line 1
----> 1 clean_data(0,3)

TypeError: 'DataFrame' object is not callable
```

In [67]:
```
clean_data
```

Out[67]:

	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 [69]:
```
x_iv =clean_data.drop(['Salary'],axis=1)
```

In [71]:
```
clean_data
```

Out[71]:

	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 [72]:
```
x_iv
```

Out[72]:

	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 [73]:
```
x_iv.columns
```

Out[73]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Exp'], dtype='object')
```

In [77]:
```
y_dv= clean_data.drop(['Name','Domain','Age','Location','Exp'],axis=1)
```

In [78]:
```
y_dv
```

Out[78]:

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

In [79]:
```
clean_data
```

Out[79]:

	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 [80]:
```
x_iv
```

Out[80]:

	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 [82]:
```
y_dv
```

Out[82]:

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

In [83]:
```
clean_data
```

Out[83]:

	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 [84]:
```
imputation = pd.get_dummies(clean_data)
```

In [86]:
```
imputation
```

Out[86]:

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

In [87]:
```
imputation.columns
```

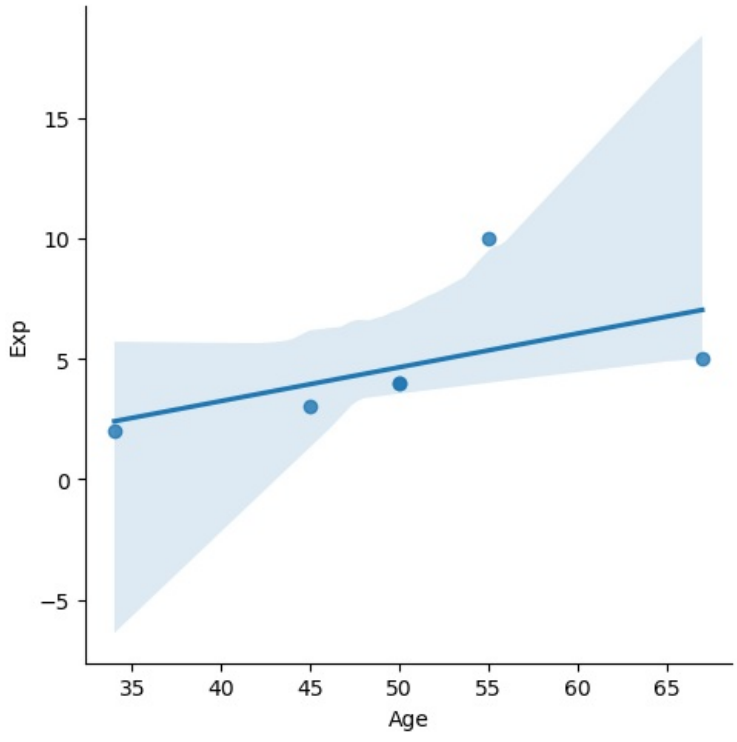
Out[87]:
```
Index(['Age', 'Salary', 'Exp', 'Name_Jane', 'Name_Kim', 'Name_Mike',
       'Name_Teddy', 'Name_Umar', 'Name_Uttam', 'Domain_Analytics',
       'Domain_Dataanalyst', 'Domain_Datascience', 'Domain_NLP',
       'Domain_Statistics', 'Domain_Testing', 'Location_Bangalore',
       'Location_Delhi', 'Location_Hyderbad', 'Location_Mumbai'],
      dtype='object')
```

In [88]:
```
clean_data
```

Out[88]:

	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 [101…
```
vis9 = sns.lmplot(data = clean_data,x='Age',y='Exp')
```



In [ ]:
```
vis7=
```

```
In [ ]: vis7=
```