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

In [2]:
```
emp = pd.read_excel(r'Downloads\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.shape
```

Out[4]:
```
(6, 6)
```

In [5]:
```
emp.columns
```

Out[5]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='objec
t')
```

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

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

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

Out[7]:
```
6
```

In [8]:
```
emp.describe()
```

Out[8]:

	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 [9]:
```
emp.columns
```

Out[9]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='objec
t')
```

In [10]:
```
emp[['Name','Domain']]
```

Out[10]:

	Name	Domain
0	Mike	Datascience#$
1	Teddy^	Testing
2	Uma#r	Dataanalyst^^#
3	Jane	Ana^^lytics
4	Uttam*	Statistics
5	Kim	NLP

In [11]:
```
emp[['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp']]
```

Out[11]:

	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 [12]:
```
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: 416.0+ bytes
```

In [13]:
```
emp['Name']
```

Out[13]:
```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

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

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_8372\389424325.py:1: FutureWarni
ng: The default value of regex will change from True to False in a future ve
rsion.
  emp['Name'] = emp['Name'].str.replace(r'\W','')
```

In [15]:
```
emp['Name']
```

Out[15]:
```
0     Mike
1    Teddy
2     Umar
3     Jane
4    Uttam
5      Kim
Name: Name, dtype: object
```

In [16]:
```
emp['Domain'] = emp['Domain'].str.replace(r'\W','')
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_8372\2360087947.py:1: FutureWarn
ing: The default value of regex will change from True to False in a future v
ersion.
  emp['Domain'] = emp['Domain'].str.replace(r'\W','')
```

In [17]:
```
emp
```

Out[17]:

	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 [18]:
```
emp['Location'] = emp['Location'].str.replace(r'\W','')
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_8372\3886403992.py:1: FutureWarn
ing: The default value of regex will change from True to False in a future v
ersion.
  emp['Location'] = emp['Location'].str.replace(r'\W','')
```

In [19]:
```
emp
```

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

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_8372\3358378917.py:1: FutureWarn
ing: The default value of regex will change from True to False in a future v
ersion.
  emp['Age'] = emp['Age'].str.replace(r'\W','')
```

In [21]:
```
emp
```

Out[21]:

	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 [22]:
```
emp['Age'] = emp['Age'].str.extract('(\d+)')
```

```
In [23]: emp
```

Out[23]:

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

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_8372\1304150360.py:1: FutureWarn
ing: The default value of regex will change from True to False in a future v
ersion.
  emp['Salary'] = emp['Salary'].str.replace(r'\W','')
```

```
In [25]: emp
```

Out[25]:

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

```
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 = emp.copy()
```

In [29]:

```
emp
```

Out[29]:

	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 [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	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 [31]:

```
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: 416.0+ bytes
```

In [32]:

```
import numpy as np
```

In [33]:
```
clean_data
```

Out[33]:

	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 [35]:
```
clean_data["Age"]=clean_data["Age"].fillna(np.mean(pd.to_numeric
                                                    (clean_data["Age"]) ))
```

In [36]:
```
clean_data['Age']
```

Out[36]:
```
0       34
1       45
2    50.25
3    50.25
4       67
5       55
Name: Age, dtype: object
```

In [37]:
```
clean_data['Age'] = clean_data['Age'].astype(int)
```

In [38]:
```
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  4 non-null      object
 4   Salary    6 non-null      object
 5   Exp       5 non-null      object
dtypes: int32(1), object(5)
memory usage: 392.0+ bytes
```

In [39]:
```
clean_data['Exp']=clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['
In [40]: clean_data['Exp']

Out[40]: 0      2
         1      3
         2      4
         3    4.8
         4      5
         5     10
         Name: Exp, dtype: object
```

```
In [41]: clean_data['Exp'] = clean_data['Exp'].astype(int)
```

```
In [42]: clean_data['Location'] = clean_data['Location'].fillna(np.mode(pd.to_numeric((
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[42], line 1
----> 1 clean_data['Location'] = clean_data['Location'].fillna(np.mode(pd.to
_numeric(clean_data['Location'])))

File ~\anaconda3\lib\site-packages\numpy\__init__.py:311, in __getattr__(att
r)
    308     from .testing import Tester
    309     return Tester
--> 311 raise AttributeError("module {!r} has no attribute "
    312                      "{!r}".format(__name__, attr))

AttributeError: module 'numpy' has no attribute 'mode'
```

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

```
In [45]: clean_data['Location']

Out[45]: 0       Mumbai
         1    Bangalore
         2    Bangalore
         3     Hyderbad
         4    Bangalore
         5        Delhi
         Name: Location, dtype: object
```

In [46]:
```
clean_data
```

Out[46]:

	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 [47]:
```
clean_data.to_csv('clean_data.csv')
```

In [48]:
```
import os
os.getcwd()
```

Out[48]:
```
'C:\\Users\\Admin'
```

In [49]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

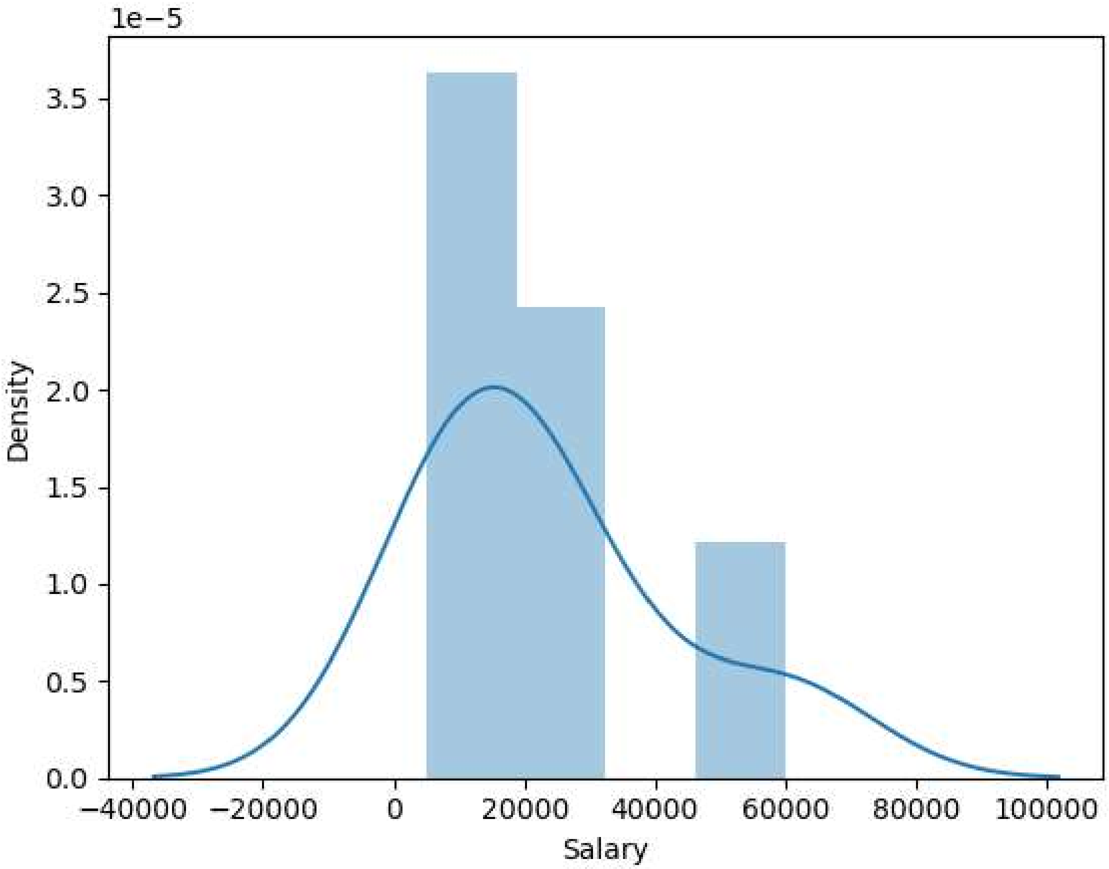
In [50]:
```
%matplotlib inline

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

In [51]:
```
clean_data['Salary']
```

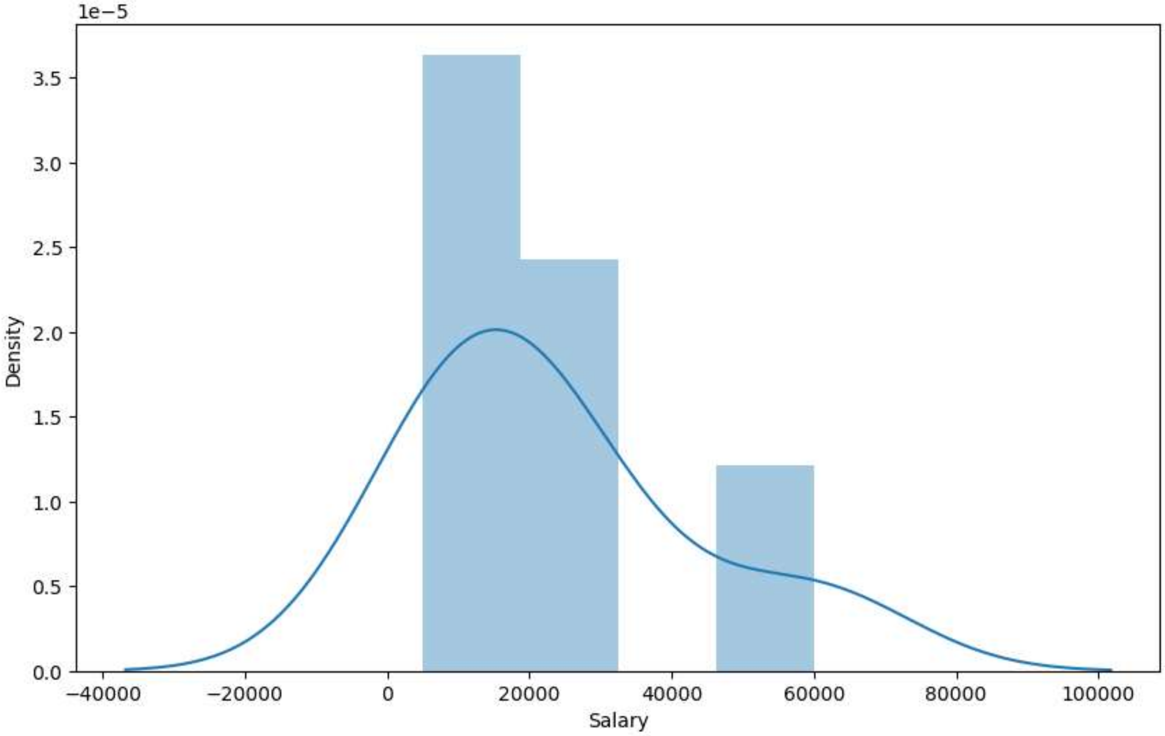
Out[51]:
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

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

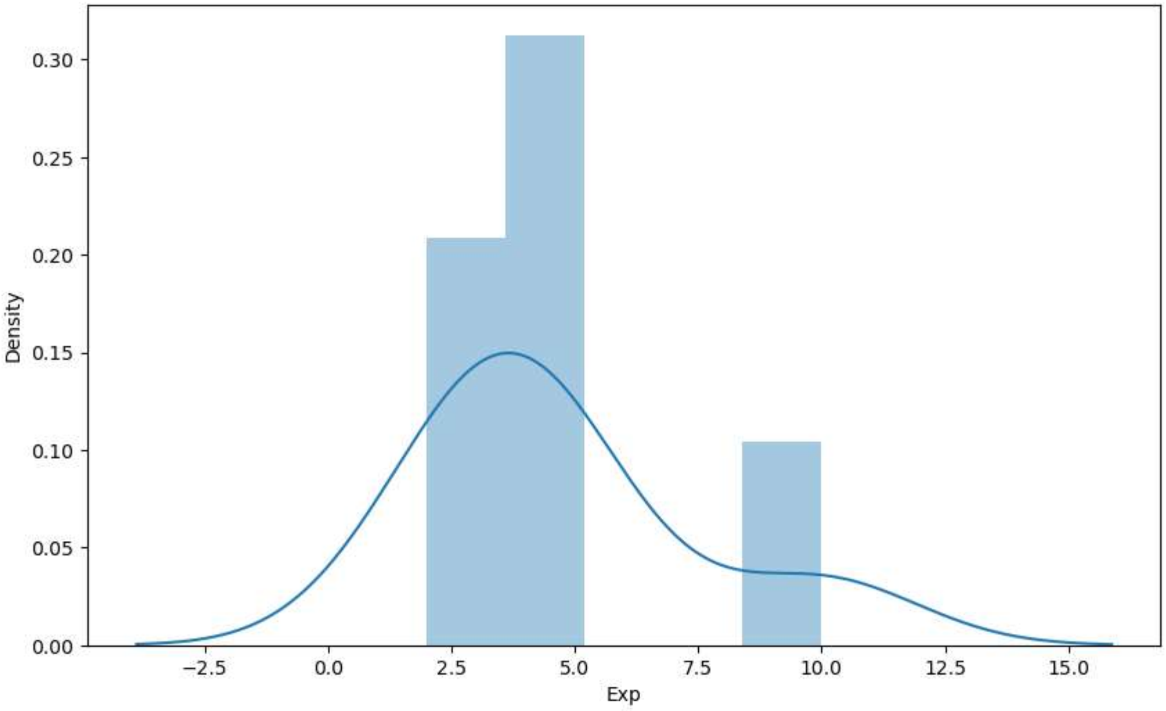


In [55]:
```
plt.rcParams['figure.figsize']=10,6
```

In [56]:
```
vis1 = sns.distplot(clean_data['Salary'])
```

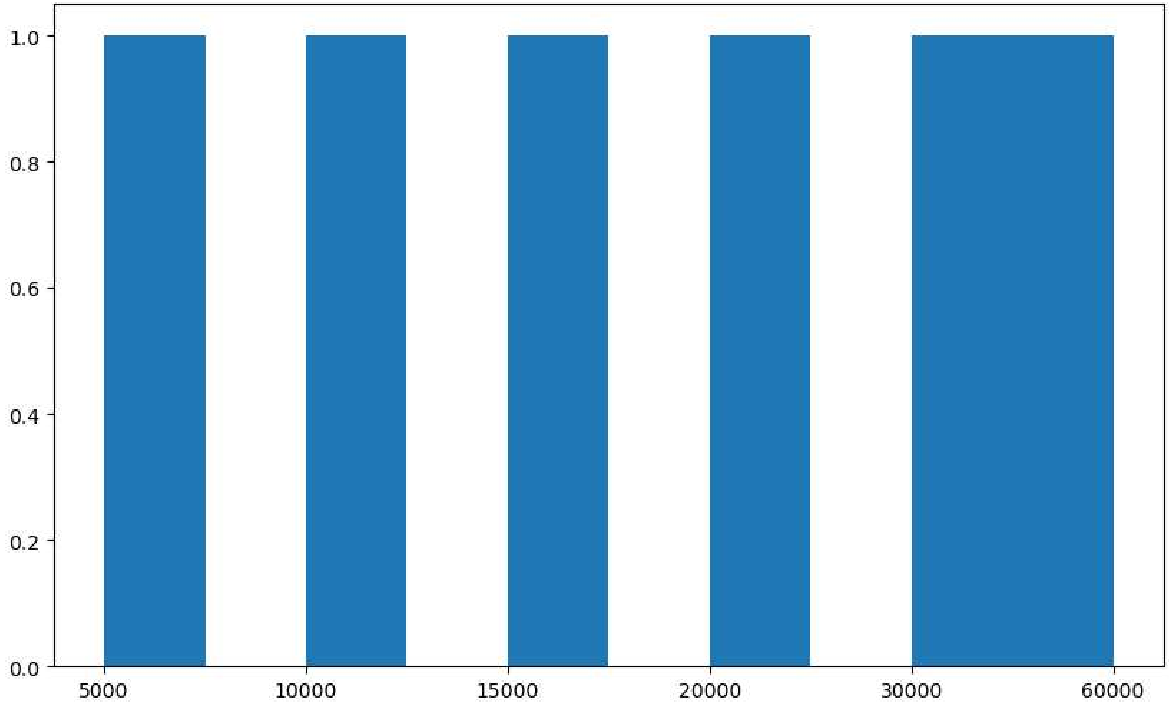


In [57]:
```
vis2 = sns.distplot(clean_data['Exp'])
```



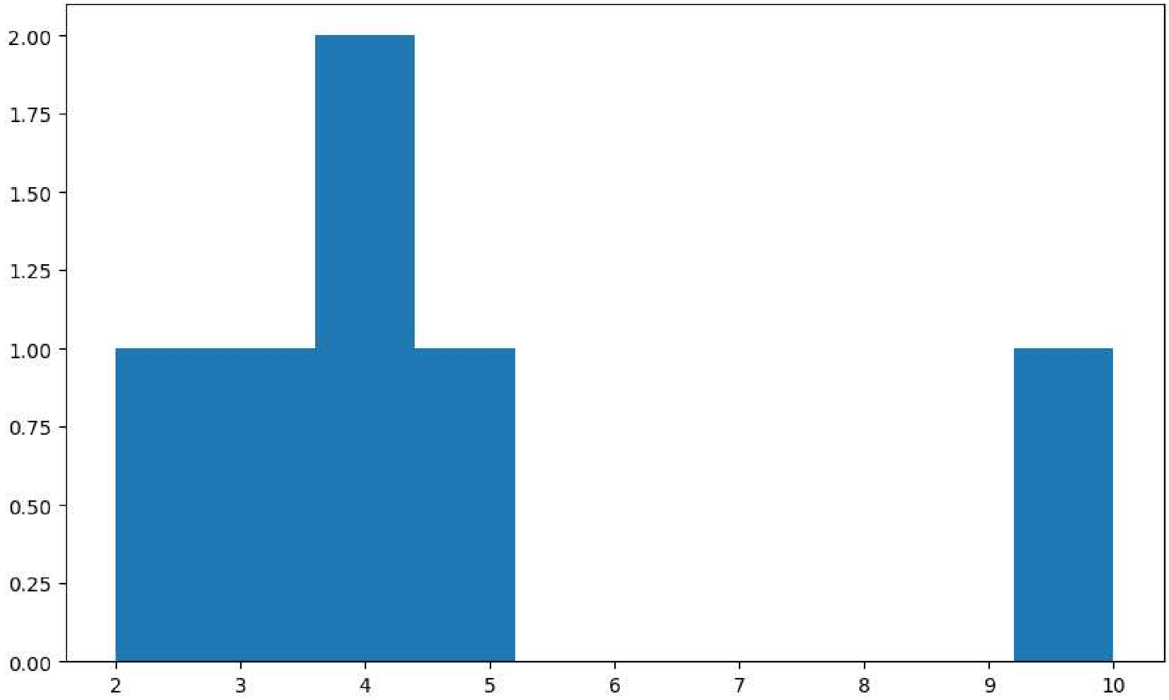
In [58]:

```
vis3 = plt.hist(clean_data['Salary'])
```



In [59]:

```
vis4 = plt.hist(clean_data['Exp'])
```



In [60]:

```
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      int32 
dtypes: int32(2), object(4)
memory usage: 368.0+ bytes
```

In [61]:

```
clean_data.Name = clean_data.Name.astype('category')
```

In [62]:

```
clean_data.Domain = clean_data.Domain.astype('category')
```

In [63]:

```
clean_data.Location = clean_data.Location.astype('category')
```

In [64]:

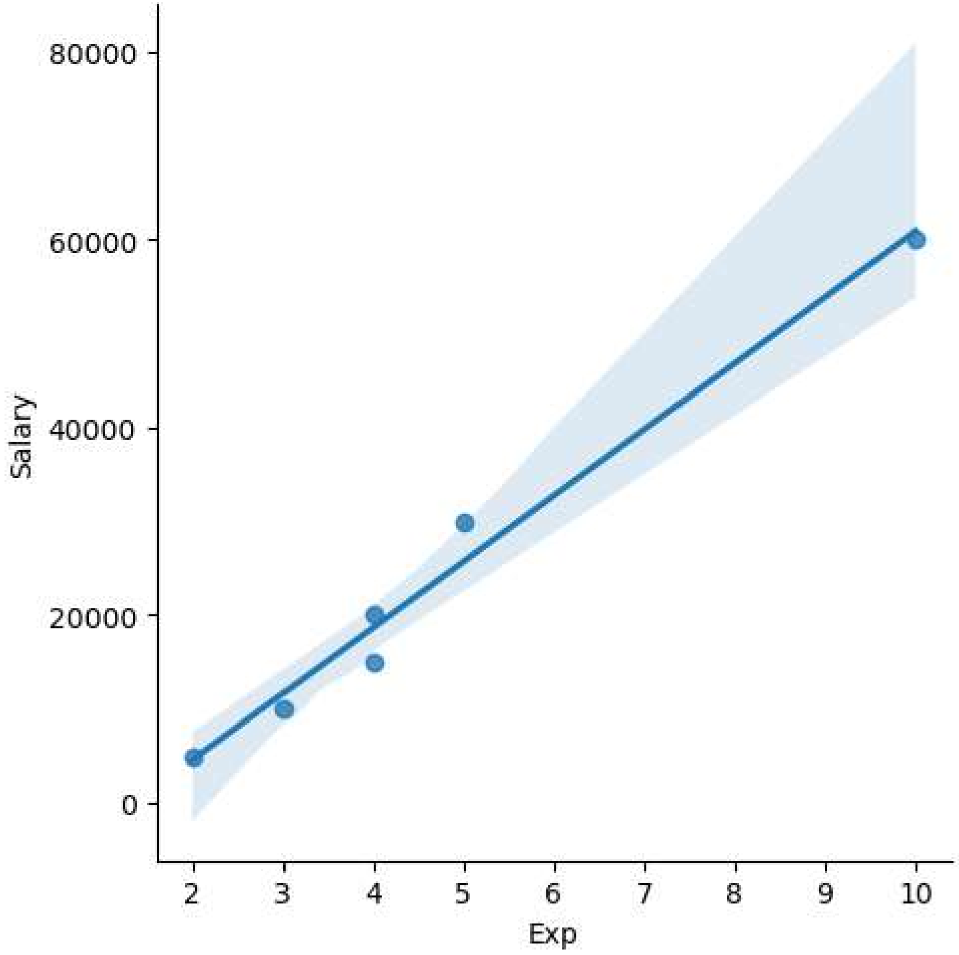
```
clean_data['Salary'] = clean_data['Salary'].astype(int)
```

In [65]:

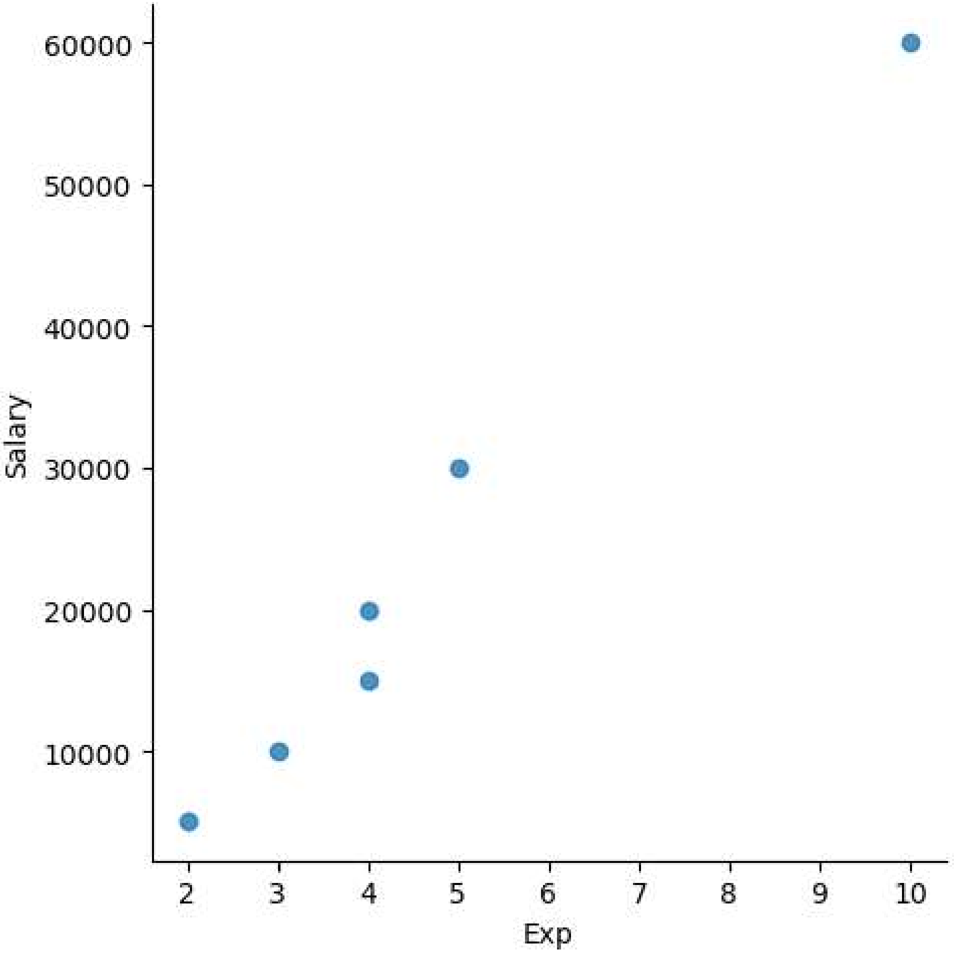
```
clean_data.info()
```

```
<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: 862.0 bytes
```

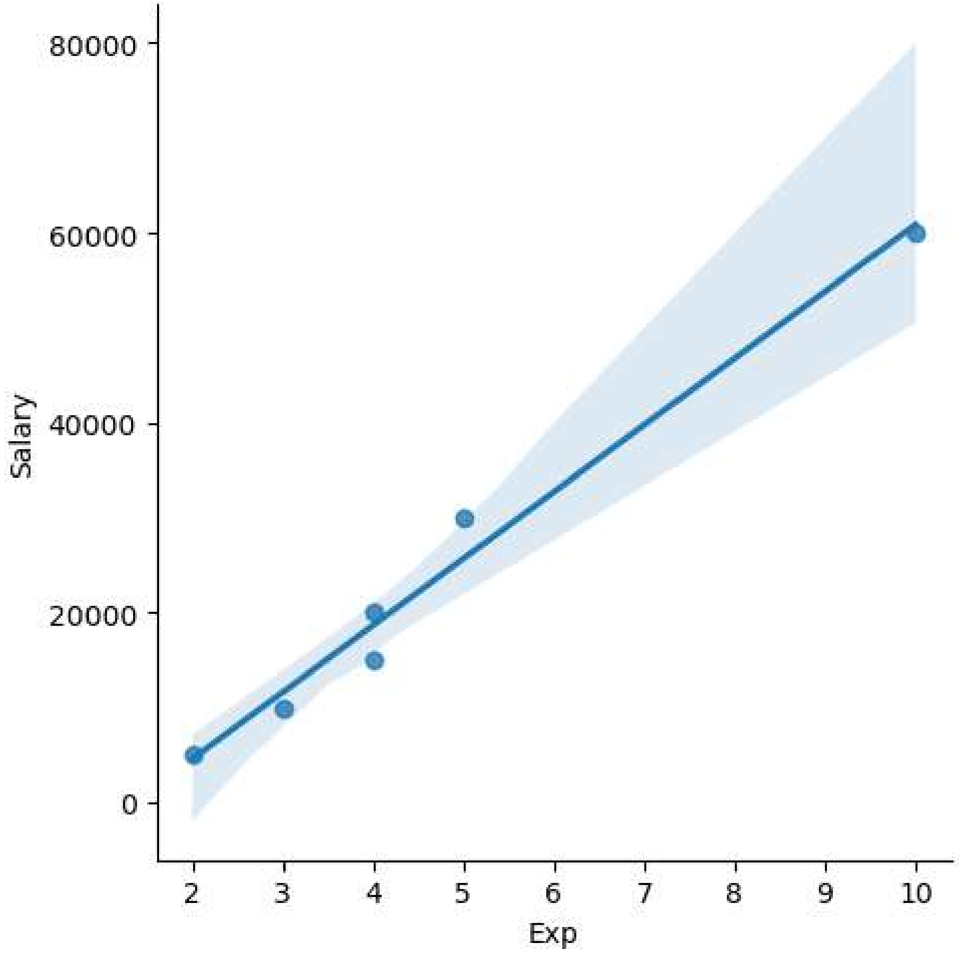
```
In [66]: vis6 = sns.lmplot(data = clean_data, x = "Exp", y='Salary')
```



```
In [67]: vis6 = sns.lmplot(data = clean_data, x = "Exp", y='Salary', fit_reg = False)
```



```
In [68]: vis6 = sns.lmplot(data = clean_data, x = "Exp", y='Salary', fit_reg = True)
```



```
In [69]: clean_data[0:5:2]
```

Out[69]:

	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 [70]: clean_data
```

Out[70]:

	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 [71]: emp
```

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

```
In [73]: x_iv
```

Out[73]:

	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 [74]: y_dv = clean_data.drop(['Name','Domain','Age','Location','Exp'],axis = 1)
In [75]: y_dv
```

Out[75]:

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

```
In [76]: clean_data
```

Out[76]:

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

```
In [78]: imputation
```

Out[78]:

	Age	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Umar	Name_Utt
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	4	1	0	0	0	0	
4	67	30000	5	0	0	0	0	0	
5	55	60000	10	0	1	0	0	0	

```
In [ ]:
```