In [1]:

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

In [2]:

```
df=pd.read_excel(r'C:\Users\Dell\Downloads\Rawdata\Rawdata.xlsx')
```

In [3]:

```
df
```

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]:

```
df.shape
```

Out[4]:

```
(6, 6)
```

In [5]:

```
df.info
```

Out[5]:

```
<bound method DataFrame.info of      Name          Domain       Age    Loca
tion   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]:

```
df['Name']=df['Name'].str.replace(r'\W','')
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_13224\2336627936.py:1: FutureWa
rning: The default value of regex will change from True to False in a futu
re version.
  df['Name']=df['Name'].str.replace(r'\W','')
```

In [7]:

```
df
```

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	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 [8]:

```
df['Domain']=df['Domain'].str.replace(r'\W','')
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_13224\1578638787.py:1: FutureWa
rning: The default value of regex will change from True to False in a futu
re version.
  df['Domain']=df['Domain'].str.replace(r'\W','')
```

In [9]:

```
df
```

Out[9]:

	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 [10]:

```
df['Location']=df['Location'].str.replace(r'\W','')
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_13224\1080459875.py:1: FutureWa
rning: The default value of regex will change from True to False in a futu
re version.
  df['Location']=df['Location'].str.replace(r'\W','')
```

In [11]:

```
df
```

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	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 [12]:

```
cat_data=df[['Name','Domain','Location']]
cat_data
```

Out[12]:

	Name	Domain	Location
0	Mike	Datascience	Mumbai
1	Teddy	Testing	Bangalore
2	Umar	Dataanalyst	NaN
3	Jane	Analytics	Hyderbad
4	Uttam	Statistics	NaN
5	Kim	NLP	Delhi

In [13]:

```
df
```

Out[13]:

	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 [14]:

```
df['Age']=df['Age'].str.extract('(\d)')
```

In [15]:

```
df['Age']
```

Out[15]:

```
0      3
1      4
2    NaN
3    NaN
4      6
5      5
Name: Age, dtype: object
```

In [16]:

```
df['Salary']=df['Salary'].str.replace(r'\W','')
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_13224\1559525201.py:1: FutureWa
rning: The default value of regex will change from True to False in a futu
re version.
  df['Salary']=df['Salary'].str.replace(r'\W','')
```

In [17]:

```
df['Salary']
```

Out[17]:

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

In [18]:

```
df
```

Out[18]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	3	Mumbai	5000	2+
1	Teddy	Testing	4	Bangalore	10000	<3
2	Umar	Dataanalyst	NaN	NaN	15000	4> yrs
3	Jane	Analytics	NaN	Hyderbad	20000	NaN
4	Uttam	Statistics	6	NaN	30000	5+ year
5	Kim	NLP	5	Delhi	60000	10+

In [19]:

```
df['Exp']=df['Exp'].str.extract('(\d+)')
```

In [20]:

```
df
```

Out[20]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	3	Mumbai	5000	2
1	Teddy	Testing	4	Bangalore	10000	3
2	Umar	Dataanalyst	NaN	NaN	15000	4
3	Jane	Analytics	NaN	Hyderbad	20000	NaN
4	Uttam	Statistics	6	NaN	30000	5
5	Kim	NLP	5	Delhi	60000	10

In [21]:

```
import numpy as np
```

In [22]:

```
clean_data=df.copy()
```

In [23]:

```
clean_data
```

Out[23]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	3	Mumbai	5000	2
1	Teddy	Testing	4	Bangalore	10000	3
2	Umar	Dataanalyst	NaN	NaN	15000	4
3	Jane	Analytics	NaN	Hyderbad	20000	NaN
4	Uttam	Statistics	6	NaN	30000	5
5	Kim	NLP	5	Delhi	60000	10

In [24]:

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

Out[24]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	3	Mumbai	5000	2
1	Teddy	Testing	4	Bangalore	10000	3
2	Umar	Dataanalyst	4.5	NaN	15000	4
3	Jane	Analytics	4.5	Hyderbad	20000	NaN
4	Uttam	Statistics	6	NaN	30000	5
5	Kim	NLP	5	Delhi	60000	10

In [25]:

```
clean_data
```

Out[25]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	3	Mumbai	5000	2
1	Teddy	Testing	4	Bangalore	10000	3
2	Umar	Dataanalyst	4.5	NaN	15000	4
3	Jane	Analytics	4.5	Hyderbad	20000	NaN
4	Uttam	Statistics	6	NaN	30000	5
5	Kim	NLP	5	Delhi	60000	10

In [26]:

```
clean_data['Age']=clean_data['Age'].astype(int)
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 [27]:

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

Out[27]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	3	Mumbai	5000	2
1	Teddy	Testing	4	Bangalore	10000	3
2	Umar	Dataanalyst	4	NaN	15000	4
3	Jane	Analytics	4	Hyderbad	20000	4.8
4	Uttam	Statistics	6	NaN	30000	5
5	Kim	NLP	5	Delhi	60000	10

In [28]:

```
clean_data
```

Out[28]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	3	Mumbai	5000	2
1	Teddy	Testing	4	Bangalore	10000	3
2	Umar	Dataanalyst	4	NaN	15000	4
3	Jane	Analytics	4	Hyderbad	20000	4.8
4	Uttam	Statistics	6	NaN	30000	5
5	Kim	NLP	5	Delhi	60000	10

In [29]:

```
clean_data['Exp']=clean_data['Exp'].astype(int)
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       6 non-null      int32 
dtypes: int32(2), object(4)
memory usage: 368.0+ bytes
```

In [30]:

```
clean_data
```

Out[30]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	3	Mumbai	5000	2
1	Teddy	Testing	4	Bangalore	10000	3
2	Umar	Dataanalyst	4	NaN	15000	4
3	Jane	Analytics	4	Hyderbad	20000	4
4	Uttam	Statistics	6	NaN	30000	5
5	Kim	NLP	5	Delhi	60000	10

In [31]:

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

In [32]:

```
import os
os.getcwd()
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [33]:

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

Out[33]:

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

In [34]:

```
clean_data
```

Out[34]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	3	Mumbai	5000	2
1	Teddy	Testing	4	Bangalore	10000	3
2	Umar	Dataanalyst	4	NaN	15000	4
3	Jane	Analytics	4	Hyderbad	20000	4
4	Uttam	Statistics	6	NaN	30000	5
5	Kim	NLP	5	Delhi	60000	10

In [35]:

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

In [36]:

```
clean_data['Location']
```

Out[36]:

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

In [37]:

```
clean_data
```

Out[37]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	3	Mumbai	5000	2
1	Teddy	Testing	4	Bangalore	10000	3
2	Umar	Dataanalyst	4	Bangalore	15000	4
3	Jane	Analytics	4	Hyderbad	20000	4
4	Uttam	Statistics	6	Bangalore	30000	5
5	Kim	NLP	5	Delhi	60000	10

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  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 [39]:

```
clean_data.Name=clean_data.Name.astype('category')
clean_data.Domain=clean_data.Name.astype('category')
clean_data.Location=clean_data.Name.astype('category')
clean_data['Salary']=clean_data['Salary'].astype(int)
```

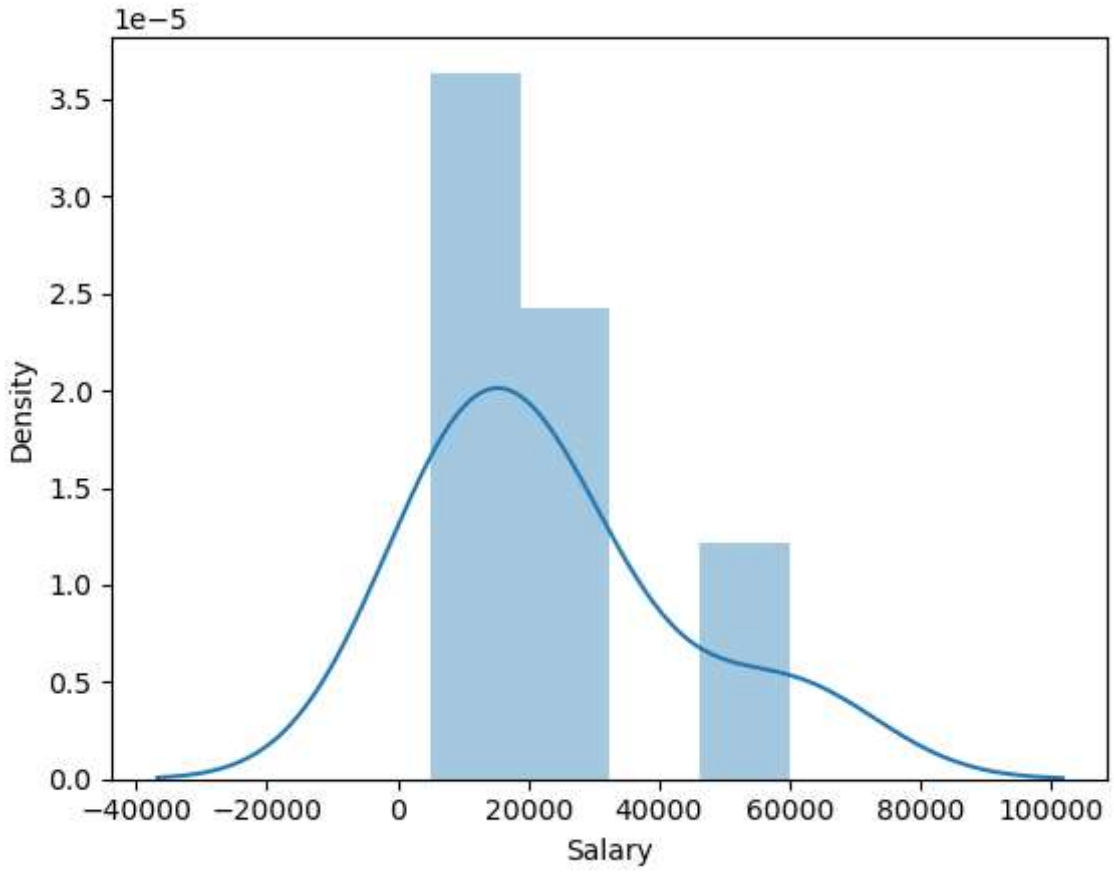
In [40]:

```
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: 878.0 bytes
```

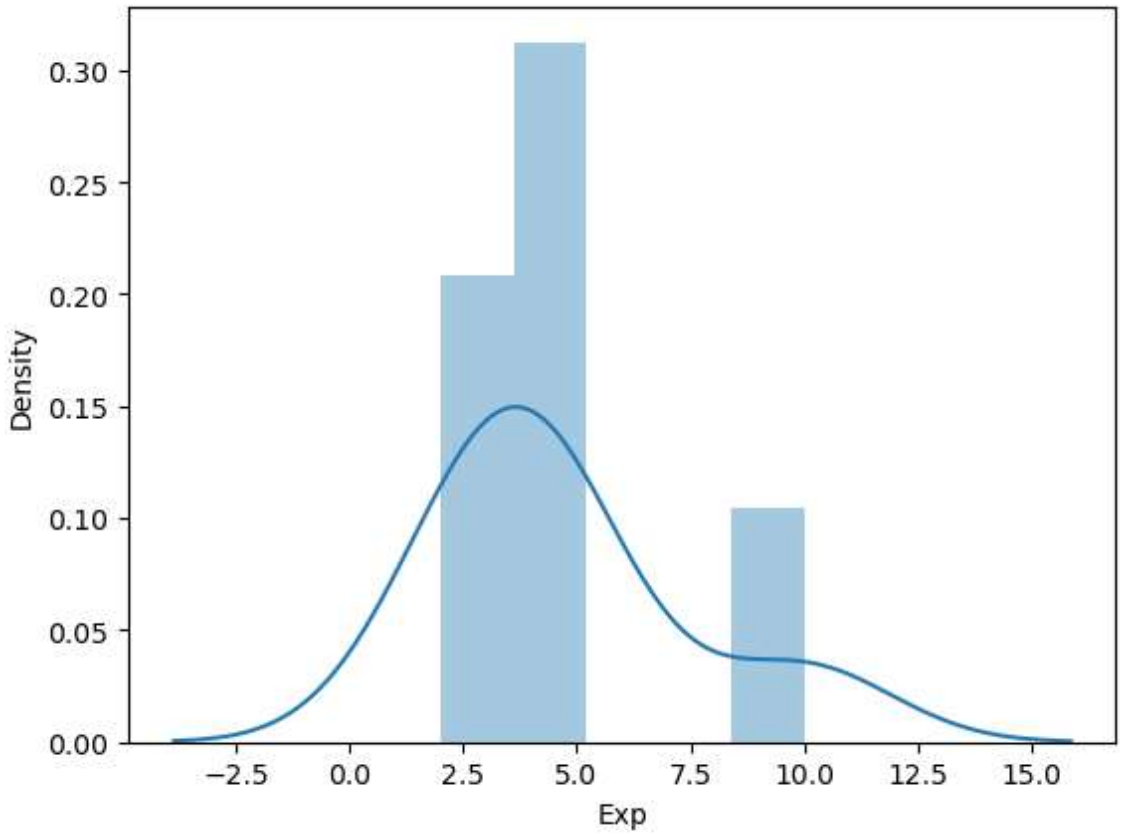
In [41]:

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



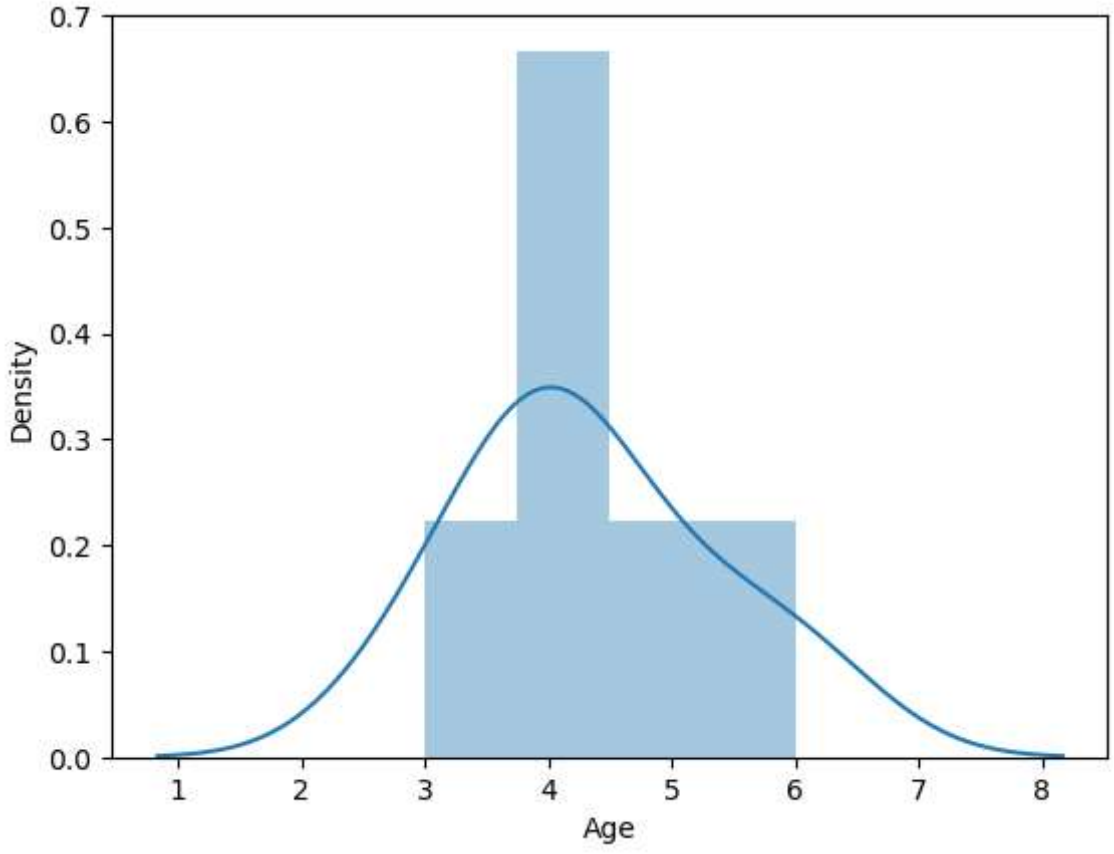
In [42]:

```
vis1=sns.distplot(clean_data['Exp'])
```



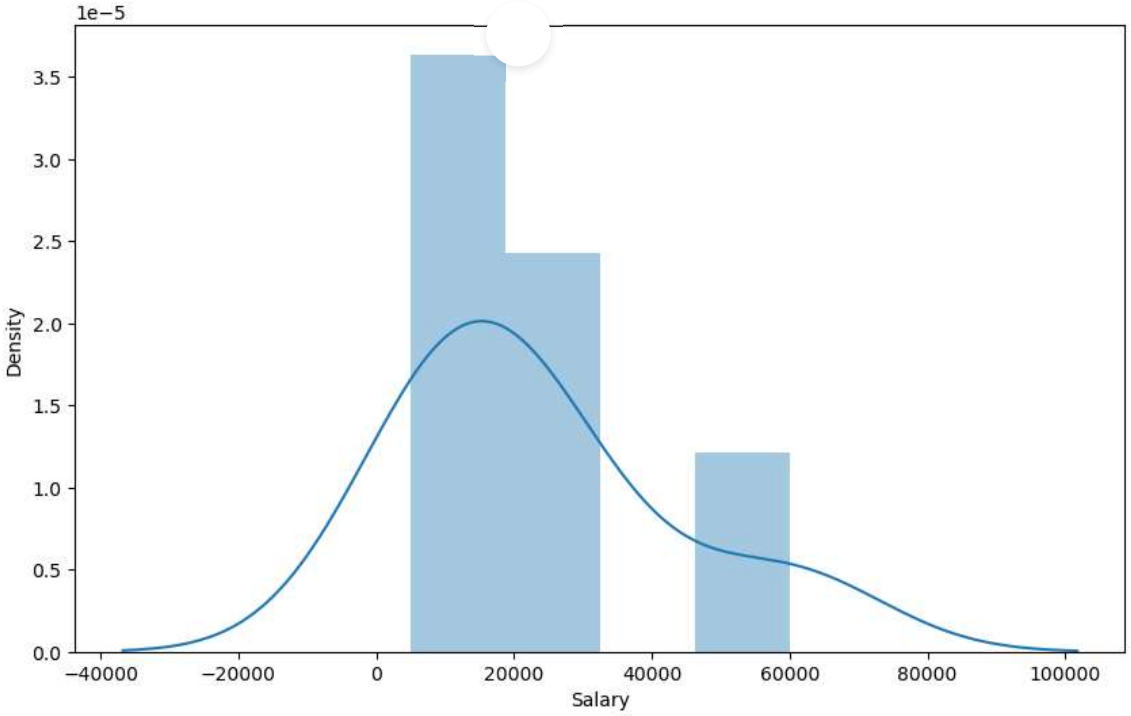
In [43]:

```
vis1=sns.distplot(clean_data['Age'])
```



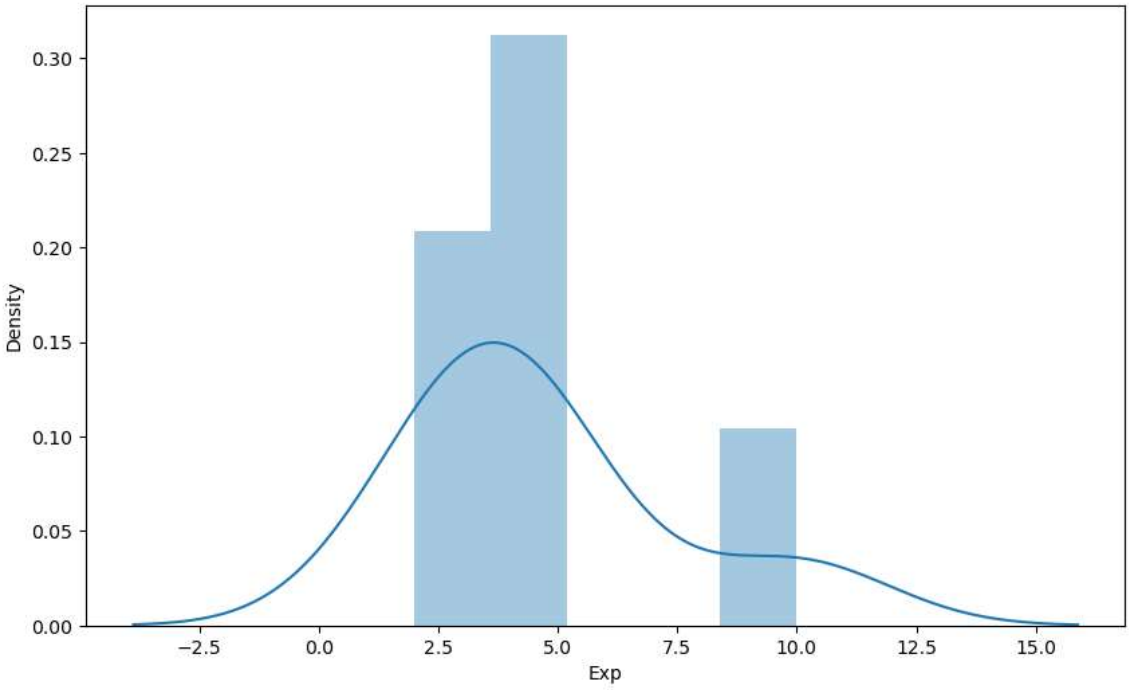
In [44]:

```
plt.rcParams['figure.figsize']=10,6
vis1=sns.distplot(clean_data['Salary'])
```



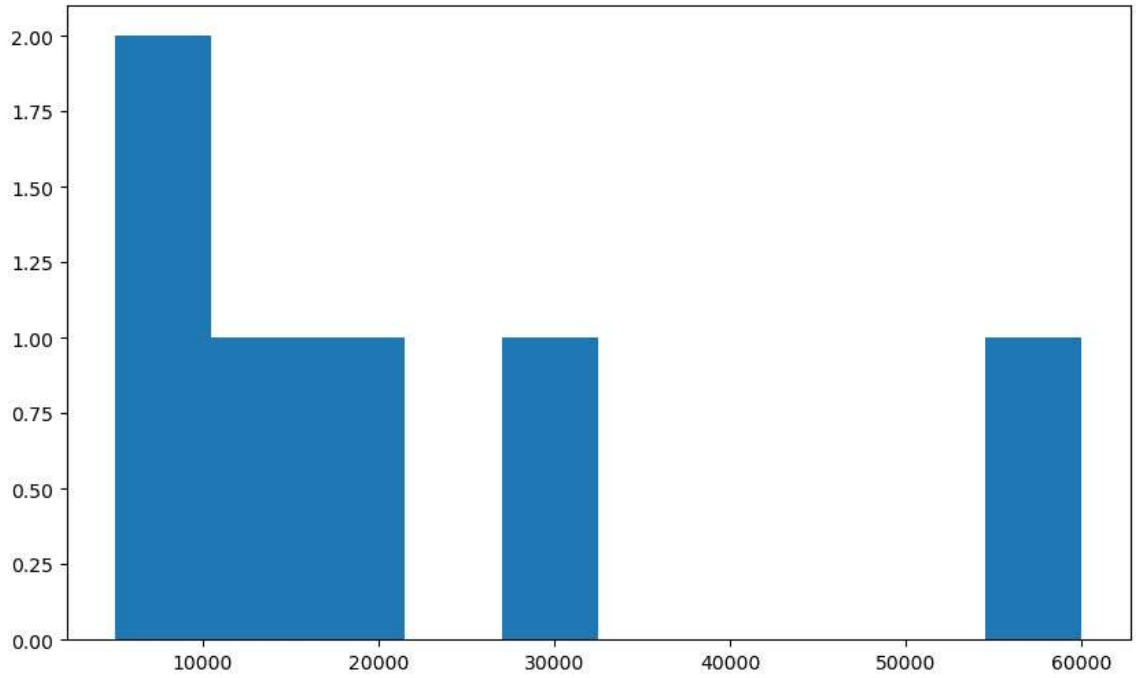
In [45]:

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



In [46]:

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



In [47]:

```
clean_data[0:4:2]
```

Out[47]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Mike	3	Mike	5000	2
2	Umar	Umar	4	Umar	15000	4

In [48]:

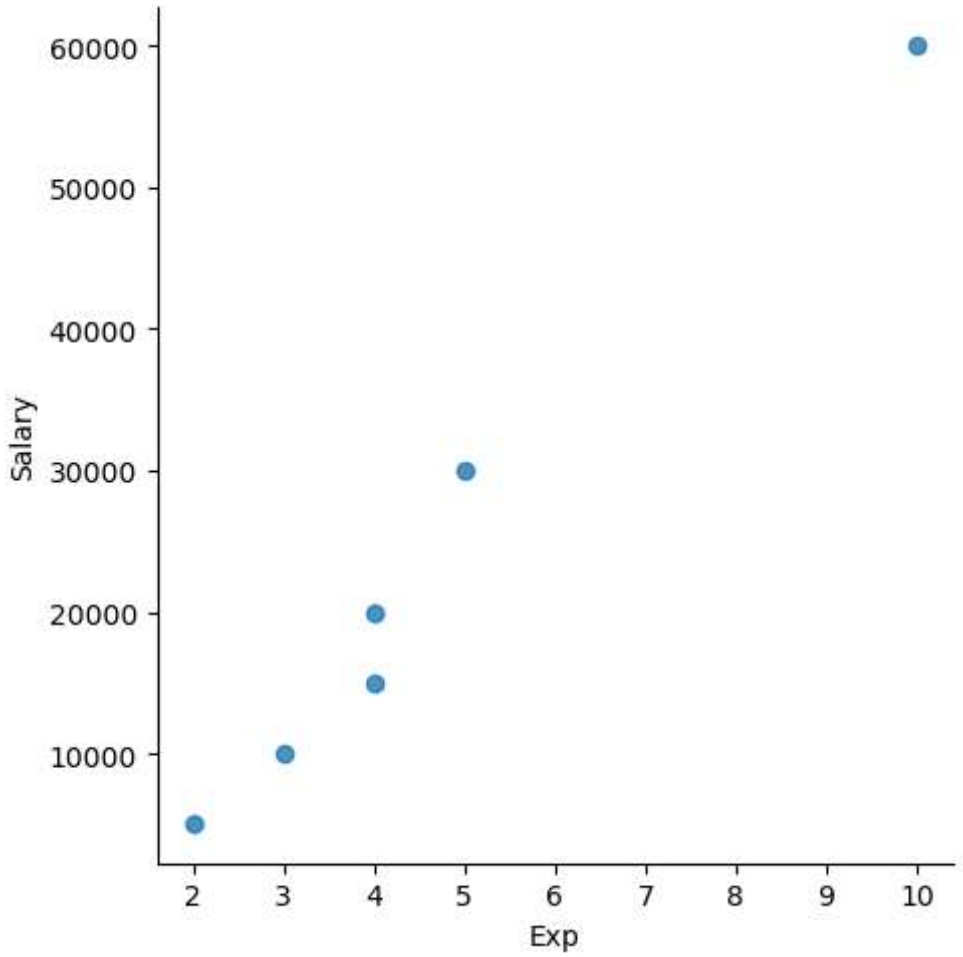
```
y_dv=clean_data.drop(['Name','Domain','Age','Location','Exp'],axis=1)
y_dv
```

Out[48]:

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

In [49]:

```
vis6=sns.lmplot(data=clean_data,x='Exp',y='Salary',fit_reg=False)
```



In [50]:

```
clean_data.head(2)
```

Out[50]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Mike	3	Mike	5000	2
1	Teddy	Teddy	4	Teddy	10000	3

In [51]:

```
imputation=pd.get_dummies(clean_data)
imputation
```

Out[51]:

	Age	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Umar	Name_
0	3	5000	2	0	0	1	0	0	
1	4	10000	3	0	0	0	1	0	
2	4	15000	4	0	0	0	0	1	
3	4	20000	4	1	0	0	0	0	
4	6	30000	5	0	0	0	0	0	
5	5	60000	10	0	1	0	0	0	

6 rows × 21 columns