```
print('hai')
```

```
hai
```

```
import pandas as pd

emp= pd.read_excel(r"C:\Users\ttwrd\Downloads\Rawdata.xlsx")
emp
```

```
     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+
```

```
emp.columns
```

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

```
emp.shape
```

```
(6, 6)
```

```
id(emp)
```

```
3141807330352
```

```
emp.info
```

```
<bound method DataFrame.info of      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+>
```

```
emp.head()
```

```
     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
```

```
emp.tail
```

```
<bound method NDFrame.tail of      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+>
```

```
emp.isnull()
```

```
    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
```

```
emp.isna()
```

```
    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
```

```
emp
```

```
    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+
```

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

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

```
emp['Name']=emp['Name'].str.replace(r'\W','',regex=True)

emp['Name']
```

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

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

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

```
emp['Domain']=emp['Domain'].str.replace(r'\W','',regex=True)
emp['Domain']
```

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

```
emp['Age']
```

```
0    34 years
1      45' yr
2         NaN
3         NaN
4       67-yr
5        55yr
Name: Age, dtype: object
```

```
emp["Age"]=emp["Age"].astype(str)
```

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

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

```
emp['Age'] = pd.to_numeric(emp['Age'])
```

```
emp['Age']
```

```
0    34.0
1    45.0
2     NaN
3     NaN
4    67.0
5    55.0
Name: Age, dtype: float64
```

```
emp['Exp']
```

```
0         2+
1         <3
2     4> yrs
3        NaN
4    5+ year
5        10+
Name: Exp, dtype: object
```

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

```
emp['Exp']
```

```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

```
emp
```

```
     Name       Domain   Age   Location   Salary  Exp
0    Mike  Datascience  34.0     Mumbai   5^00#0    2
1  Teddy^      Testing  45.0  Bangalore  10%%000    3
2   Uma#r  Dataanalyst   NaN        NaN  1$5%000    4
3    Jane    Analytics   NaN   Hyderbad   2000^0  NaN
4  Uttam*   Statistics  67.0        NaN   30000-    5
5     Kim          NLP  55.0      Delhi  6000^$0   10
```

```
emp['Salary']
```

```
0     5000
1    10000
```

```
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

```
emp
```

```
    Name       Domain   Age   Location Salary  Exp
0   Mike  Datascience  34.0     Mumbai   5000    2
1  Teddy      Testing  45.0  Bangalore  10000    3
2   Umar  Dataanalyst   NaN        NaN  15000    4
3   Jane    Analytics   NaN   Hyderbad  20000  NaN
4  Uttam   Statistics  67.0        NaN  30000    5
5    Kim          NLP  55.0      Delhi  60000   10
```

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

```
    Name       Domain   Age   Location Salary  Exp
0   Mike  Datascience  34.0     Mumbai   5000    2
1  Teddy      Testing  45.0  Bangalore  10000    3
2   Umar  Dataanalyst   NaN        NaN  15000    4
3   Jane    Analytics   NaN   Hyderbad  20000  NaN
4  Uttam   Statistics  67.0        NaN  30000    5
5    Kim          NLP  55.0      Delhi  60000   10
```

```
clean_data['Exp']
```

```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

```
import numpy as np
clean_data['Exp']=clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean
_data['Exp'])))
```

```
clean_data['Exp']
```

```
0      2
1      3
2      4
3    4.8
4      5
5     10
Name: Exp, dtype: object
```

```
clean_data
```

```
    Name       Domain   Age   Location Salary  Exp
0   Mike  Datascience  34.0     Mumbai   5000    2
1  Teddy      Testing  45.0  Bangalore  10000    3
2   Umar  Dataanalyst   NaN        NaN  15000    4
3   Jane    Analytics   NaN   Hyderbad  20000  4.8
4  Uttam   Statistics  67.0        NaN  30000    5
5    Kim          NLP  55.0      Delhi  60000   10
```

```
clean_data['Domain']
```

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

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

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

```
0    Datascience
1        Testing
2    Dataanalyst
3      Analytics
4     Statistics
5            NLP
Name: Domain, dtype: category
Categories (6, object): ['Analytics', 'Dataanalyst', 'Datascience',
'NLP', 'Statistics', 'Testing']
```

```
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      category
 2   Age       4 non-null      float64
 3   Location  4 non-null      category
 4   Salary    6 non-null      object
 5   Exp       6 non-null      object
```

```
dtypes: category(2), float64(1), object(3)
memory usage: 760.0+ bytes
```

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

```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: category
Categories (4, object): ['Bangalore', 'Delhi', 'Hyderbad', 'Mumbai']
```

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

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

```
0       Mumbai
1    Bangalore
2    Bangalore
3     Hyderbad
4    Bangalore
5        Delhi
Name: Location, dtype: category
Categories (4, object): ['Bangalore', 'Delhi', 'Hyderbad', 'Mumbai']
```

```
clean_data
```

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34.0	Mumbai	5000	2
1	Teddy	Testing	45.0	Bangalore	10000	3
2	Umar	Dataanalyst	NaN	Bangalore	15000	4
3	Jane	Analytics	NaN	Hyderbad	20000	4.8
4	Uttam	Statistics	67.0	Bangalore	30000	5
5	Kim	NLP	55.0	Delhi	60000	10

```
clean_data['Age']
```

```
0    34.0
1    45.0
2     NaN
3     NaN
4    67.0
5    55.0
Name: Age, dtype: float64
```

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

```
clean_data['Age']
```

```
0    34.00
1    45.00
2    50.25
3    50.25
4    67.00
5    55.00
Name: Age, dtype: float64
```

```
clean_data
```

```
    Name       Domain    Age   Location Salary  Exp
0   Mike  Datascience  34.00     Mumbai   5000    2
1  Teddy      Testing  45.00  Bangalore  10000    3
2   Umar  Dataanalyst  50.25  Bangalore  15000    4
3   Jane    Analytics  50.25   Hyderbad  20000  4.8
4  Uttam   Statistics  67.00  Bangalore  30000    5
5    Kim          NLP  55.00      Delhi  60000   10
```

```
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      float64
 3   Location  6 non-null      category
 4   Salary    6 non-null      object
 5   Exp       6 non-null      object
dtypes: category(3), float64(1), object(2)
memory usage: 938.0+ bytes
```

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

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

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

```
0     5000
1    10000
2    15000
3    20000
```

```
4    30000
5    60000
Name: Salary, dtype: int32
```

```
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      float64
 3   Location  6 non-null      category
 4   Salary    6 non-null      int32
 5   Exp       6 non-null      object
dtypes: category(3), float64(1), int32(1), object(1)
memory usage: 914.0+ bytes
```

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

```
0     2
1     3
2     4
3     4
4     5
5    10
Name: Exp, dtype: int32
```

```
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      float64
 3   Location  6 non-null      category
 4   Salary    6 non-null      int32
 5   Exp       6 non-null      int32
dtypes: category(3), float64(1), int32(2)
memory usage: 890.0 bytes
```

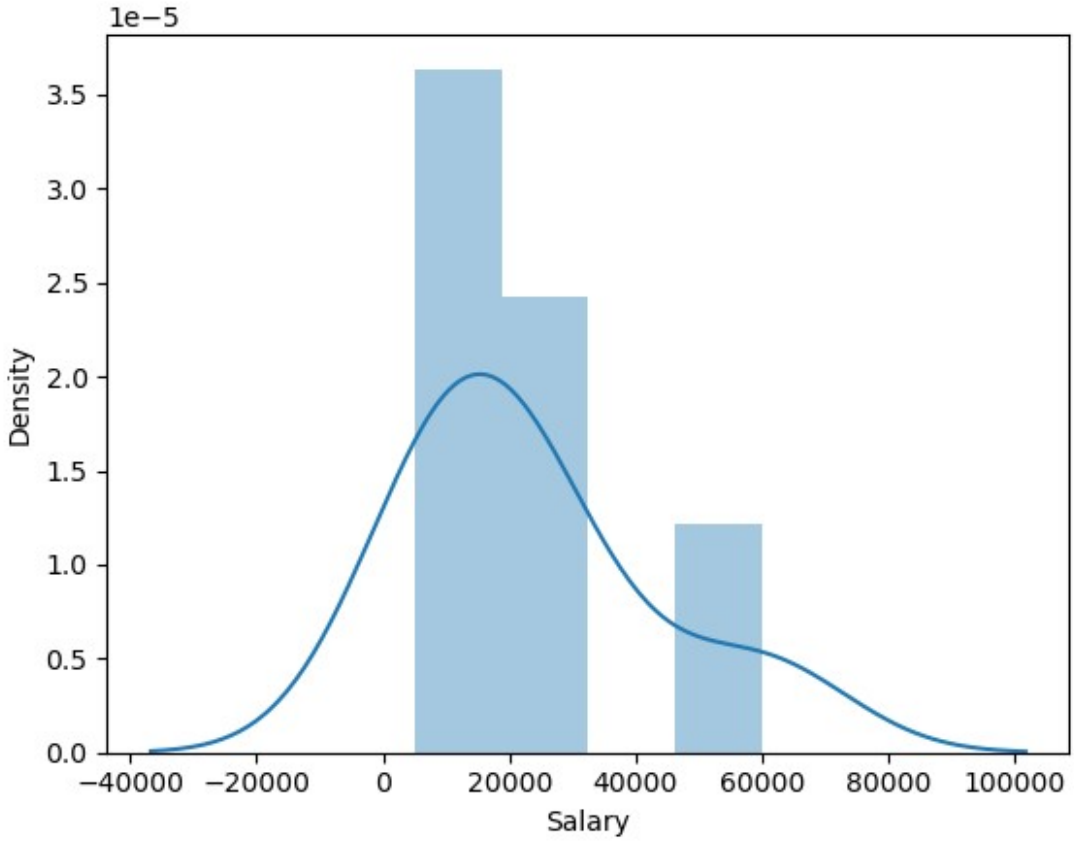
```
clean_data['Age']=clean_data['Age'].astype(int)
```

```
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')



clean_data.to_csv("clean_data.csv")

import os
os.getcwd()
```

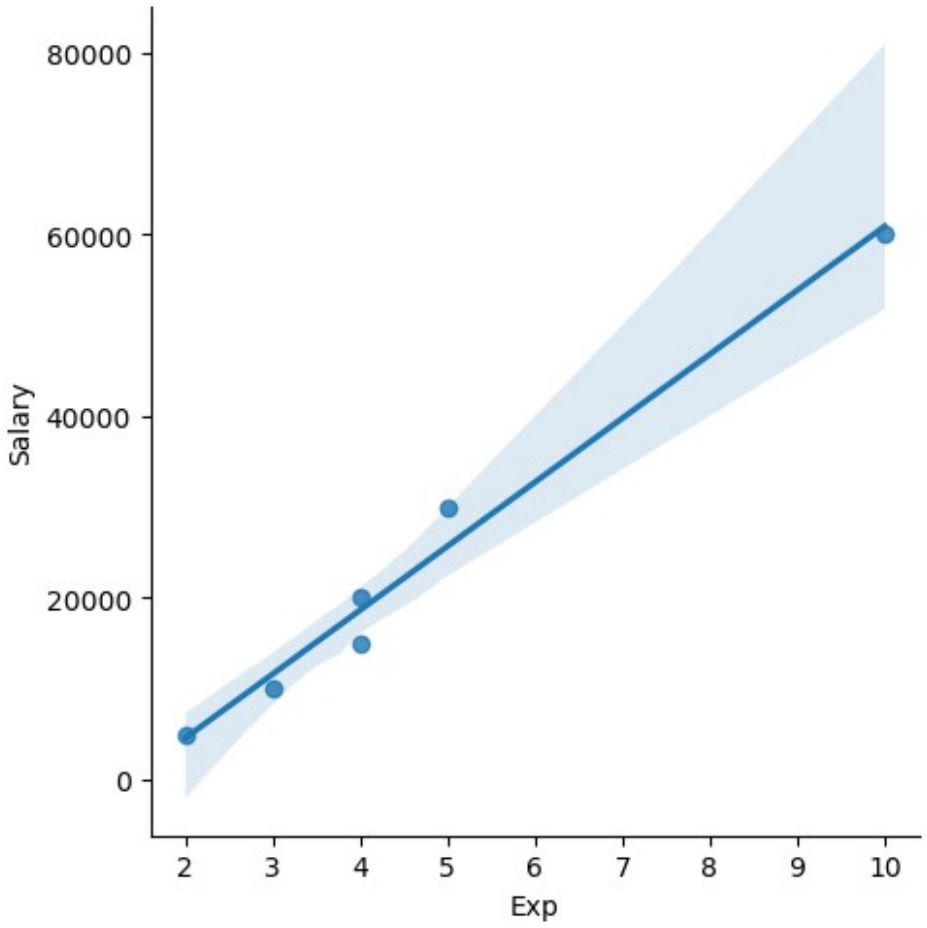
```
'C:\\Users\\ttwrd\\Downloads'
```

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

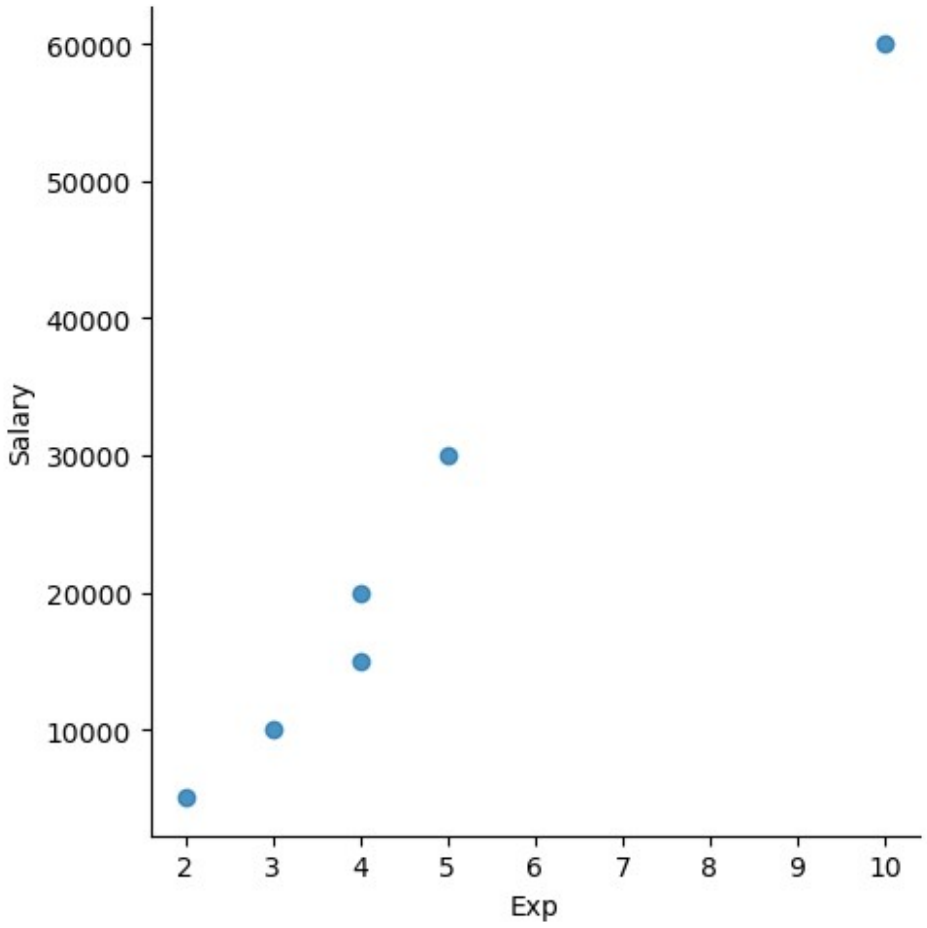
```
<Axes: xlabel='Salary', ylabel='Density'>
```



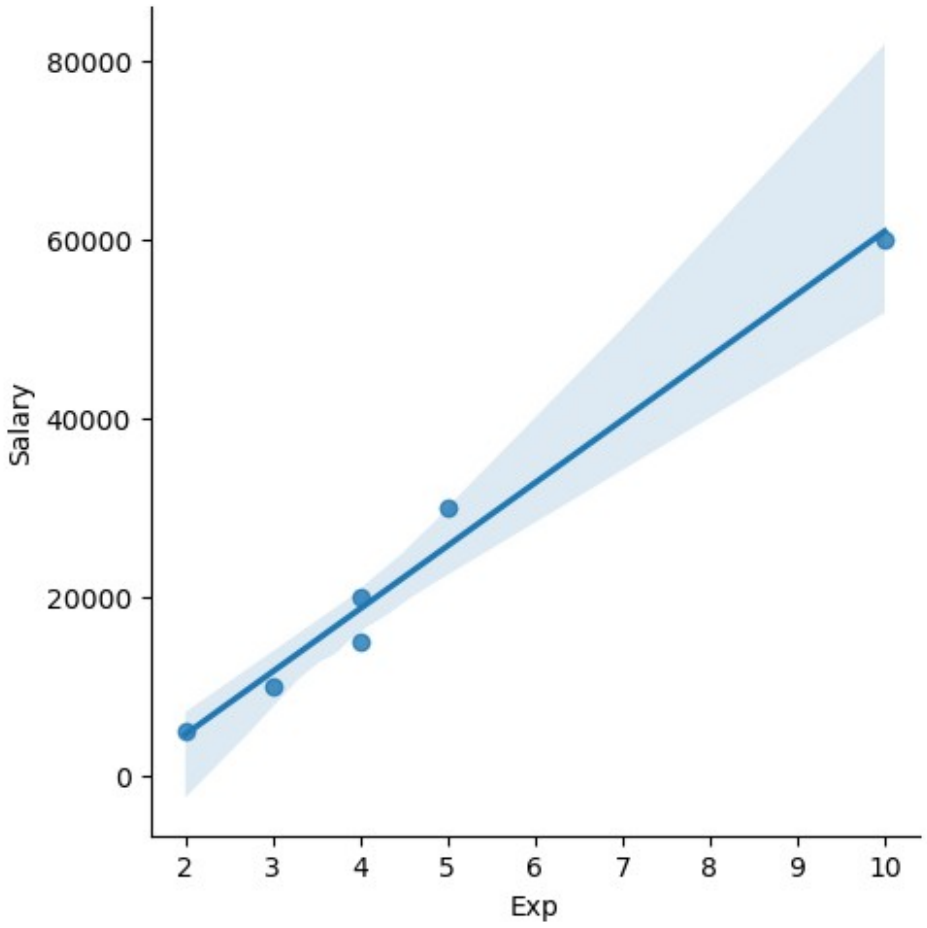

```
vis2=sns.lmplot(data=clean_data,x='Exp',y='Salary',fit_reg=True)
```




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




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




```
clean_data[:]
```

	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

```
clean_data[0:6:2]
```

	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

```
clean_data[::-1]
```

	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
```

```
clean_data.columns
```

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

```
x_indv=clean_data[['Name','Domain','Age','Location','Exp']]
```

```
type(x_indv)
```

```
pandas.core.frame.DataFrame
```

```
y_depvariable=clean_data['Salary']
type(y_depvariable)
```

```
pandas.core.series.Series
```

```
x_indv
```

```
    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
```

```
y_depvariable
```

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