In [56]:
```
import pandas as pd
```

In [57]:
```
emp=pd.read_excel(r'C:\Users\yogay\OneDrive\Documents\python\Rawdata.xlsx')
```

In [58]:
```
emp
```

Out[58]:

	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 [59]:
```
emp.shape
```

Out[59]:
```
(6, 6)
```

In [60]:
```
len(emp)
```

Out[60]:
```
6
```

In [61]:
```
emp.columns
```

Out[61]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [62]:
```
len(emp.columns)
```

Out[62]:
```
6
```

In [63]:
```
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 [64]:
```
emp
```

Out[64]:

	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 [65]:
```
emp['Name']
```

Out[65]:
```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

In [66]:
```
emp['Domain']
```

Out[66]:
```
0     Datascience#$
1           Testing
2    Dataanalyst^^#
3       Ana^^lytics
4        Statistics
5               NLP
Name: Domain, dtype: object
```

In [67]:
```
emp['Age']
```

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

In [68]:
```
emp['Location']
```

Out[68]:
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [69]:
```
emp['Salary']
```

Out[69]:
```
0     5^00#0
1    10%%000
2    1$5%000
3     2000^0
4     30000-
5    6000^$0
Name: Salary, dtype: object
```

In [70]:
```
emp['Exp']
```

Out[70]:
```
0         2+
1         <3
2     4> yrs
3        NaN
4    5+ year
5        10+
Name: Exp, dtype: object
```

In [71]:
```
emp[['Name','Domain']]
```

Out[71]:

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

In [72]:
```
emp[['Name','Domain','Age']]
```

Out[72]:

	Name	Domain	Age
0	Mike	Datascience#$	34 years
1	Teddy^	Testing	45' yr
2	Uma#r	Dataanalyst^^#	NaN
3	Jane	Ana^^lytics	NaN
4	Uttam*	Statistics	67-yr
5	Kim	NLP	55yr

In [73]:
```
emp[['Name','Domain','Age','Location','Salary','Exp']]
```

Out[73]:

	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 [74]:
```
emp['Name']
```

Out[74]:
```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

In [75]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [76]:
```
emp['Name']=emp['Name'].str.replace(r'\W','')
```

In [77]:
```
emp
```

Out[77]:

	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+

data cleaning

In [78]:
```
emp['Domain']=emp['Domain'].str.replace(r'\W','')
```

In [79]:
```
emp['Domain']
```

Out[79]:
```
0    Datascience
1        Testing
2    Dataanalyst
3      Analytics
4     Statistics
5            NLP
Name: Domain, dtype: object
```

In [80]:
```
emp
```

Out[80]:

	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 [81]:
```
emp['Age']=emp['Age'].str.replace(r'\W','')
```

In [82]:
```
emp['Age']
```

Out[82]:
```
0    34years
1       45yr
2        NaN
3        NaN
4       67yr
5       55yr
Name: Age, dtype: object
```

In [83]:
```
emp['Age']=emp['Age'].str.extract('(\d+)')
```

In [84]:
```
emp['Age']
```

Out[84]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [85]:
```
emp
```

Out[85]:

	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 [86]:
```
emp['Salary'] = emp['Salary'].str.replace(r'\W','')
```

In [87]:
```
emp['Salary']
```

Out[87]:
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

In [88]:
```
emp
```

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	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 [89]:
```
emp['Exp']=emp['Exp'].str.extract('(\d+)')
```

In [90]:
```
emp
```

Out[90]:

	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 [91]:
```
clean_data = emp.copy()
```

In [92]:
```
clean_data
```

Out[92]:

	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

Handling Missing Values

In [93]:
```
clean_data
```

Out[93]:

	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 [94]:
```
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 [95]:
```
import numpy as np
```

In [96]:
```
clean_data.head(1)
```

Out[96]:

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

In [97]:
```
clean_data['Age']
```

Out[97]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [98]:
```
clean_data['Age'] = clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['Age'])))
```

In [99]:
```
clean_data['Age']
```

Out[99]:
```
0       34
1       45
2    50.25
3    50.25
4       67
5       55
Name: Age, dtype: object
```

In [100…
```
emp
```

Out[100]:

	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 [101…
```
clean_data
```

Out[101]:

	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 [103…
```
clean_data['Exp'] = clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['Exp'])))
```

In [104…
```
clean_data['Exp']
```

Out[104]:
```
0      2
1      3
2      4
3    4.8
4      5
5     10
Name: Exp, dtype: object
```

In [105…
```
clean_data['Location'] = clean_data['Location'].fillna(clean_data['Location'].mode()[0])
```

In [106…
```
clean_data['Location']
```

Out[106]:
```
0       Mumbai
1    Bangalore
2    Bangalore
3     Hyderbad
4    Bangalore
5        Delhi
Name: Location, dtype: object
```

In [107…
```
clean_data
```

Out[107]:

	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

change types of objects

In [108…
```
clean_data['Age'] = clean_data['Age'].astype(int)
```

In [109…
```
clean_data['Salary'] = clean_data['Salary'].astype(int)
```

In [110…
```
clean_data['Exp'] = clean_data['Exp'].astype(int)
```

In [113…
```
clean_data['Domain'] = clean_data['Domain'].astype('category')
```

In [114…
```
clean_data['Name'] = clean_data['Name'].astype('category')
```

In [115…
```
clean_data['Location'] = clean_data['Location'].astype('category')
```

In [116…
```
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 [117…
```
clean_data
```

Out[117]:

	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 [120…
```
clean_data.to_csv('clean_data.csv')  #get dataset into the system
```

In [118…
```
import os
os.getcwd()
```

Out[118]:
```
'C:\\Users\\yogay'
```

In [121…
```
clean_data.columns
```

Out[121]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [122…
```
import matplotlib.pyplot as plt # visualization
import seaborn as sns # Advanced visualization
```

In [123…
```
import warnings
warnings.filterwarnings('ignore')
```

In [124…
```
clean_data
```

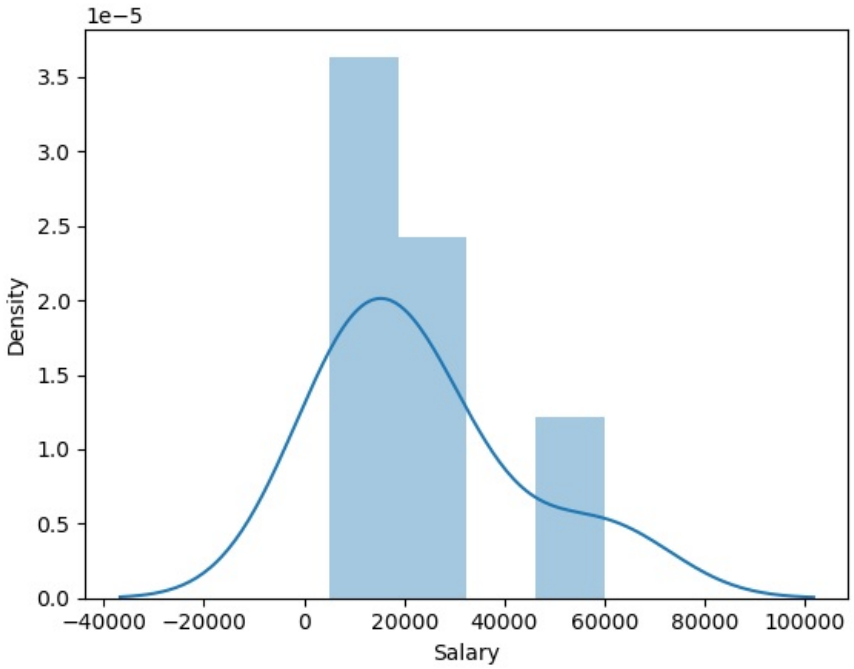
Out[124]:

	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 [125…
```
clean_data['Salary']
```

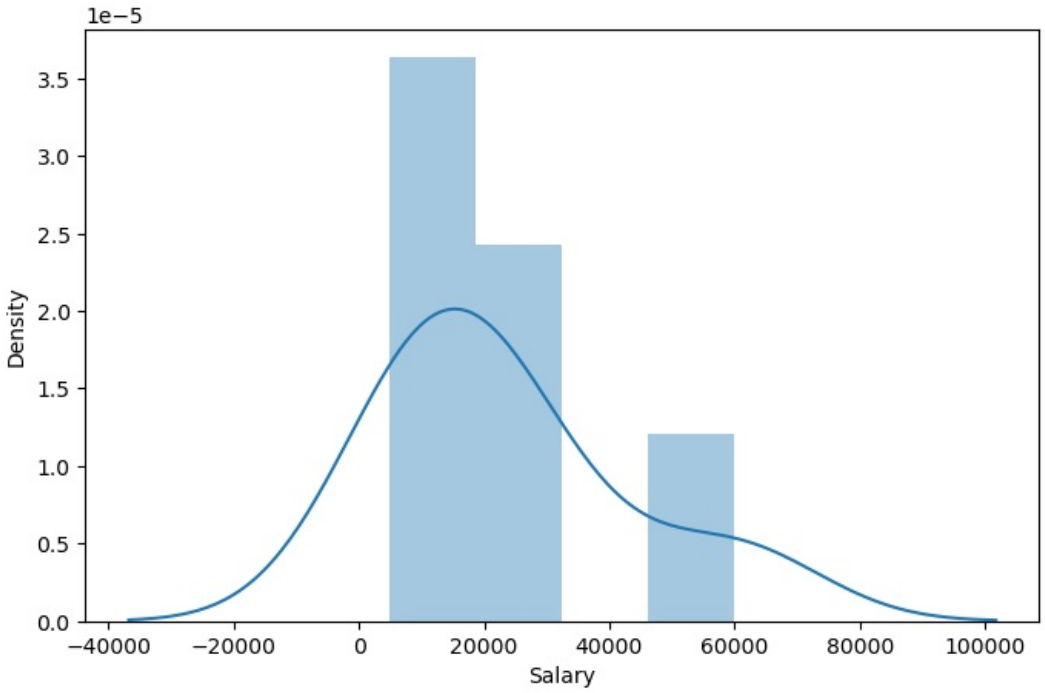
Out[125]:
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: int32
```

In [126…
```
vis1 = sns.distplot(clean_data['Salary'])
```

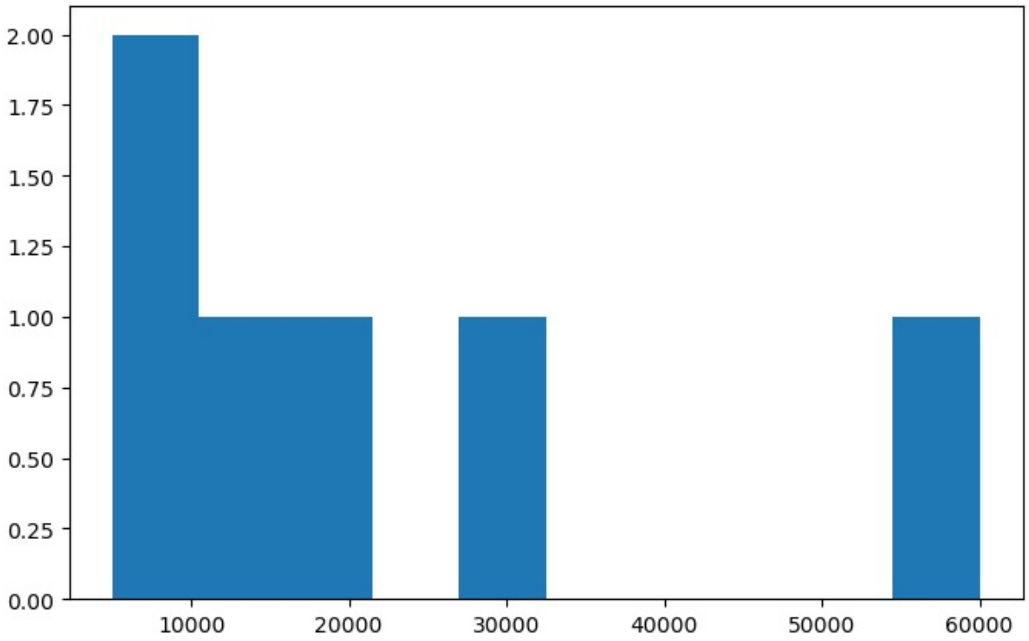


In [127…
```
plt.rcParams['figure.figsize'] = 8,5
```

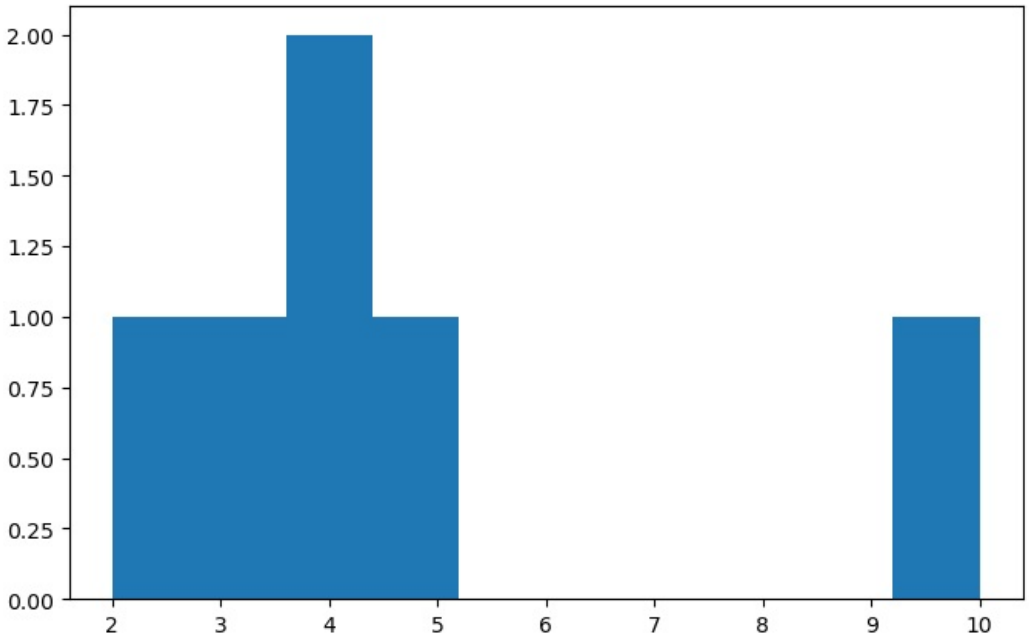
In [128…
```
vis1 = sns.distplot(clean_data['Salary'])
```



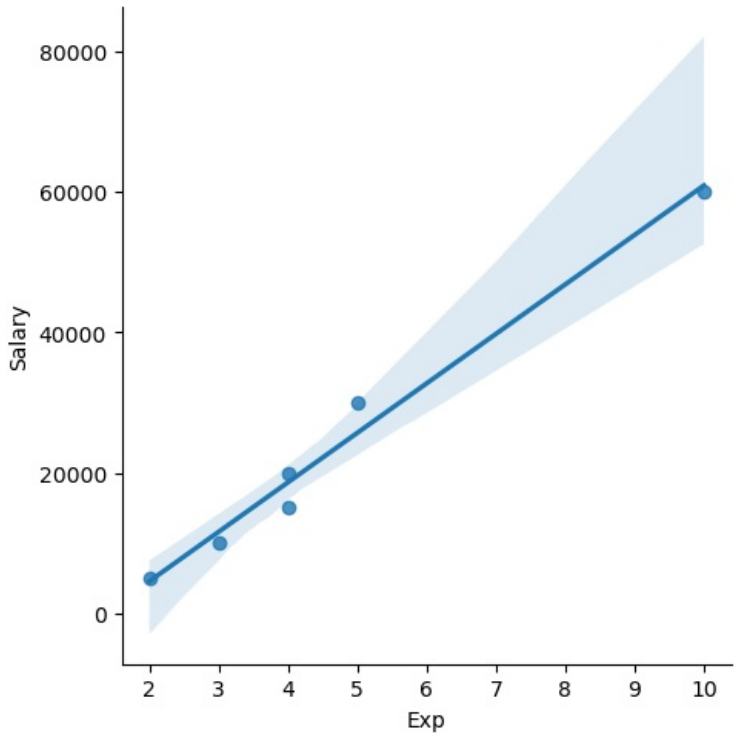
In [129…
```
vis2 = plt.hist(clean_data['Salary'])
```



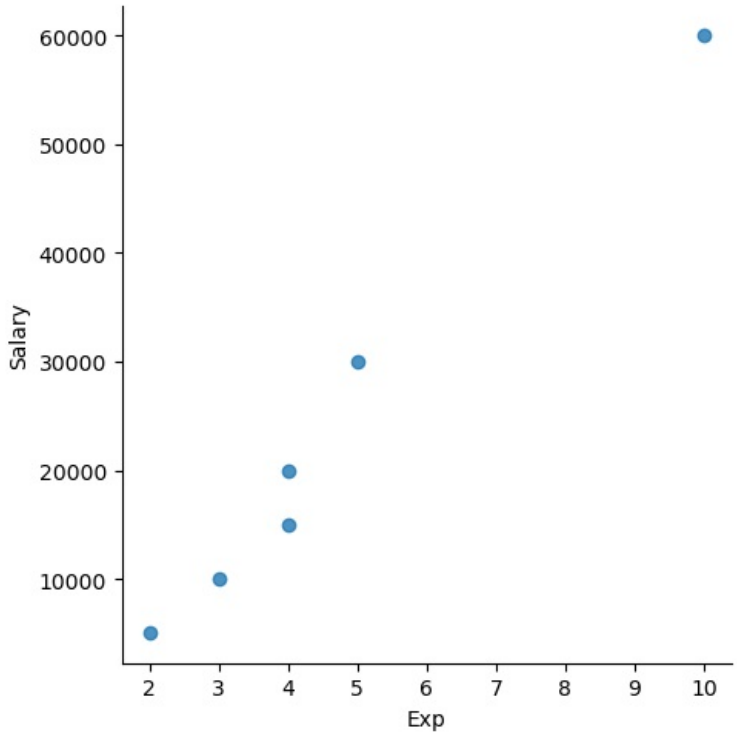
In [130…
```
vis3 = plt.hist(clean_data['Exp'])
```



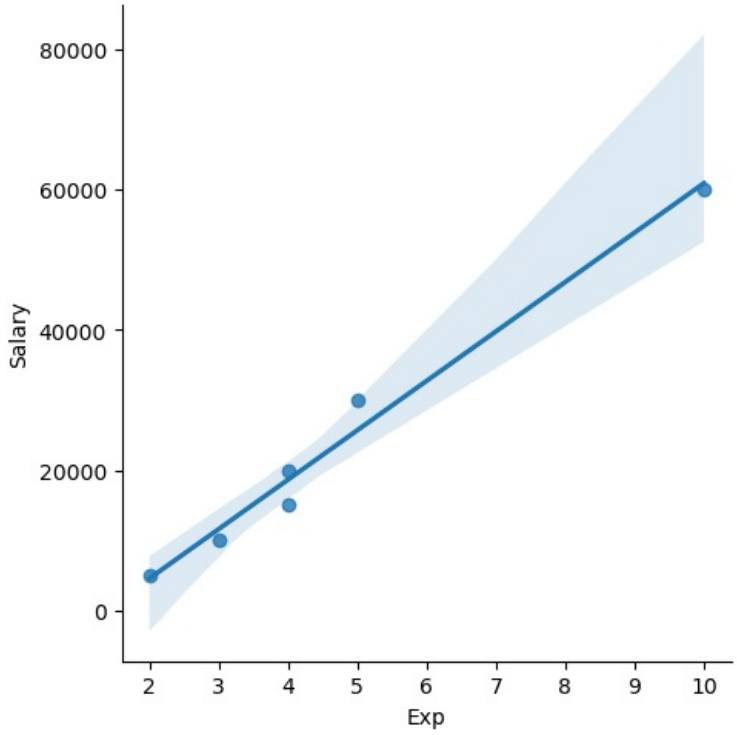
In [131…
```
vis4 = sns.lmplot(data=clean_data,x = 'Exp', y='Salary')
```



In [132…
```
vis5 = sns.lmplot(data=clean_data,x = 'Exp', y='Salary', fit_reg = False)
```



In [135…
```
vis6 = sns.lmplot(data=clean_data,x = 'Exp', y='Salary', fit_reg = True)
```



In [136…
```
clean_data
```

Out[136]:

	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 [137…
```
clean_data[:]
```

Out[137]:

	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 [138…
```
clean_data[:2]
```

Out[138]:

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

In [139…
```
clean_data[2:]
```

Out[139]:

	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 [140…
```
clean_data[0:1]
```

Out[140]:

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

In [141…
```
x,iy = clean_data.drop(['Salary'],axis=1)
```

In [141…
```
x_iv = clean_data.drop(['Salary'],axis=1)
```

In [142…
```
clean_data
```

Out[142]:

	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 [143…
```
x_iv
```

Out[143]:

	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 [144…
```
x_iv.columns
```

Out[144]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Exp'], dtype='object')
```

In [145…
```
clean_data.columns
```

Out[145]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [146…
```
y_dv = clean_data.drop(['Name', 'Domain', 'Age', 'Location','Exp'],axis=1)
```

In [147…
```
y_dv
```

Out[147]:

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

In [148…
```
clean_data
```

Out[148]:

	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 [149…
```
x_iv
```

Out[149]:

	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 [150…
```
y_dv
```

Out[150]:

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

In [151…
```
imputation = pd.get_dummies(clean_data)
```

In [152…
```
imputation
```

Out[152]:

	Age	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Umar	Name_Uttam	Domain_Analytics	Domain_Dataanalyst
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 [153…
```
clean_data.shape
```

Out[153]:
```
(6, 6)
```

In [154…
```
clean_data.columns
```

Out[154]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [155…
```
imputation.columns
```

Out[155]:
```
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 [156…
```
imputation.describe()
```

Out[156]:

	Age	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Umar	Name_Uttam	Domain_Analytics
count	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000	6.000000
mean	50.166667	23333.333333	4.666667	0.166667	0.166667	0.166667	0.166667	0.166667	0.166667	0.166667
std	10.907184	19916.492328	2.804758	0.408248	0.408248	0.408248	0.408248	0.408248	0.408248	0.408248
min	34.000000	5000.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	46.250000	11250.000000	3.250000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	50.000000	17500.000000	4.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	53.750000	27500.000000	4.750000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	67.000000	60000.000000	10.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

In [157…
```
imputation.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 19 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Age                 6 non-null      int32
 1   Salary              6 non-null      int32
 2   Exp                 6 non-null      int32
 3   Name_Jane           6 non-null      uint8
 4   Name_Kim            6 non-null      uint8
 5   Name_Mike           6 non-null      uint8
 6   Name_Teddy          6 non-null      uint8
 7   Name_Umar           6 non-null      uint8
 8   Name_Uttam          6 non-null      uint8
 9   Domain_Analytics    6 non-null      uint8
 10  Domain_Dataanalyst  6 non-null      uint8
 11  Domain_Datascience  6 non-null      uint8
 12  Domain_NLP          6 non-null      uint8
 13  Domain_Statistics   6 non-null      uint8
 14  Domain_Testing      6 non-null      uint8
 15  Location_Bangalore  6 non-null      uint8
 16  Location_Delhi      6 non-null      uint8
 17  Location_Hyderbad   6 non-null      uint8
 18  Location_Mumbai     6 non-null      uint8
dtypes: int32(3), uint8(16)
memory usage: 296.0 bytes
```

In [ ]: