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

In [2]:
```
emp = pd.read_excel(r'C:\Users\user\Eda and stat\14th - Eda practicle\EDA- Practicl
```

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.columns
```

Out[4]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [5]:
```
emp.shape  # Ruturns the shape of the array
```

Out[5]:
```
(6, 6)
```

In [6]:
```
emp.head() #Diaplay the first few rows
```

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** | 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 |

In [7]:
```
emp.tail() # The function is usd to view the last few rows of a data frame
```

Out[7]:

|   | Name | Domain | Age | Location | Salary | Exp |
|---|---|---|---|---|---|---|
| **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 [8]:

```
emp.info() #Display the summary of the dataframe
```

```
<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 [9]:

```
emp
```

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** | 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.isnull() # To Detecting the missing values
```

Out[12]:

|   | 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 |

In [13]:

```
emp.isnull().sum() # In Pandas, emp.isnull().sum() is used to identify and count mi
```

Out[13]:

```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

In [14]:

```
emp.columns
```

Out[14]:

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

In [15]:

```
emp
```

Out[15]:

|   | 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 [16]:

```
emp['Name']
```

Out[16]:

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

In [18]:

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

In [19]:

```
emp['Name']
```

Out[19]:

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

In [20]:

```
emp
```

Out[20]:

|   | 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 [21]:

```
emp['Domain']
```

Out[21]:

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

In [23]:

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

In [24]:

```
emp['Domain']
```

Out[24]:

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

In [25]:

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

```
Out[25]: 0    34 years
         1      45' yr
         2         NaN
         3         NaN
         4       67-yr
         5        55yr
         Name: Age, dtype: object
```

```
In [26]: emp['Age'] = emp['Age'].str.replace(r'\W','',regex=True)
```

```
In [27]: emp['Age']
```

```
Out[27]: 0    34years
         1       45yr
         2        NaN
         3        NaN
         4       67yr
         5       55yr
         Name: Age, dtype: object
```

```
In [28]: emp['Age'] = emp['Age'].str.extract('(\d+)')
```

```
In [29]: emp['Age']
```

```
Out[29]: 0     34
         1     45
         2    NaN
         3    NaN
         4     67
         5     55
         Name: Age, dtype: object
```

```
In [30]: emp
```

Out[30]:

| | 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 [31]: emp['Location']
```

Out[31]:

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

In [32]:

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

In [33]:

```
emp['Location']
```

Out[33]:

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

In [34]:

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

Out[34]:

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

In [35]:

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

In [36]:

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

Out[36]:

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

In [37]:

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

Out[37]:

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

In [38]:

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

Out[38]:

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

In [39]:

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

In [40]:

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

Out[40]:

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

In [41]:

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

In [43]:

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

Out[43]:

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

In [44]:

```
emp
```

Out[44]:

| | 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 [45]:

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

. Till now we have rawdata we use regex to to clean the data and removed all noises characted from the dataset

.You can also work in same thing in sql query as well

# EDA TECHNIQUE LETS APPPLY

. Missing values treatment for numeric data

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 | 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 [47]:

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

Out[47]:

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

In [49]:

```
import numpy as np
```

In [50]:

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

In [51]:

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

Out[51]:

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

In [52]:

```
emp
```

Out[52]:

| | 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 [53]:

```
clean_data
```

Out[53]:

| | 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 [54]:

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

Out[54]:

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

In [55]:

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

In [56]:

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

Out[56]:

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

In [57]:

```
clean_data
```

Out[57]:

| | 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 [58]:
```
clean_data['Location'].isnull().sum()
```

Out[58]:
```
2
```

In [59]:
```
clean_data['Location'] = clean_data['Location'].fillna(clean_data['Location'].mode(
```

In [60]:
```
clean_data['Location']
```

Out[60]:
```
0       Mumbai
1    Bangalore
2    Bangalore
3     Hyderbad
4    Bangalore
5        Delhi
Name: Location, dtype: object
```

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

Out[61]:

| | 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 [63]:
```
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      object
 3   Location  6 non-null      object
 4   Salary    6 non-null      object
 5   Exp       6 non-null      object
dtypes: object(6)
memory usage: 420.0+ bytes
```

In [65]:

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

In [66]:

```
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      object
dtypes: int32(1), object(5)
memory usage: 396.0+ bytes
```

In [69]:

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

In [71]:

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

In [72]:

```
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 [74]:

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

In [75]:
```
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: 866.0 bytes
```

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

In [78]:
```
import os
os.getcwd()
```

Out[78]:
```
'C:\\Users\\user\\Eda and stat\\14th - Eda practicle\\EDA- Practicle'
```

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 |

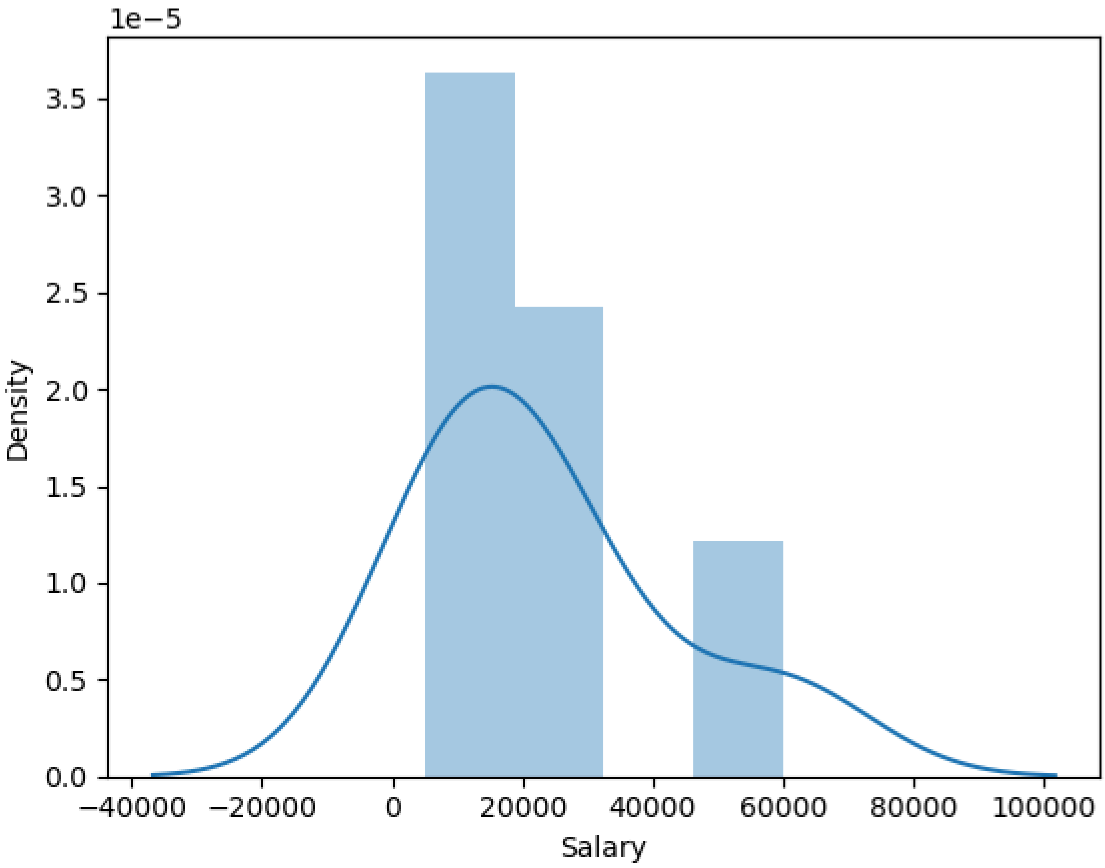
```
In [80]: import matplotlib.pyplot as plt #Visualization
         import seaborn as sns
```

```
In [81]: import warnings
         warnings.filterwarnings('ignore')
```

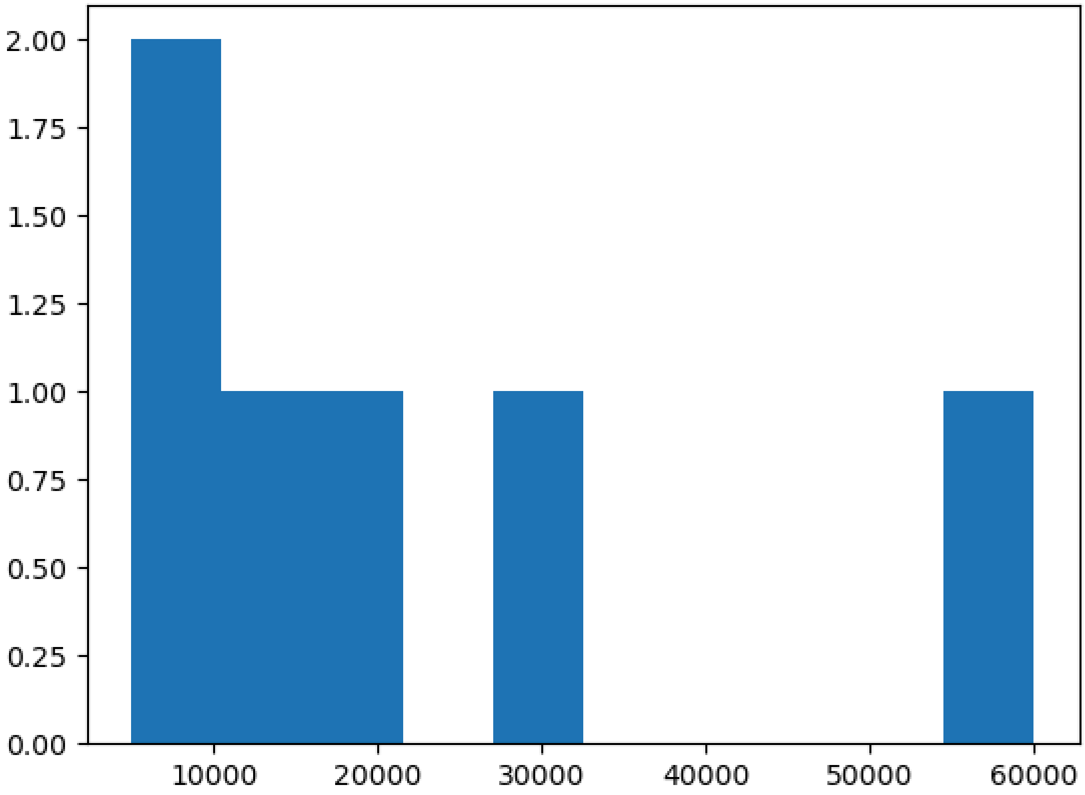
```
In [83]: clean_data['Salary']
```

```
Out[83]: 0     5000
         1    10000
         2    15000
         3    20000
         4    30000
         5    60000
         Name: Salary, dtype: int32
```

```
In [85]: vis1 = sns.distplot(clean_data['Salary'])
```



```
In [86]: vis2 = plt.hist(clean_data['Salary'])
```

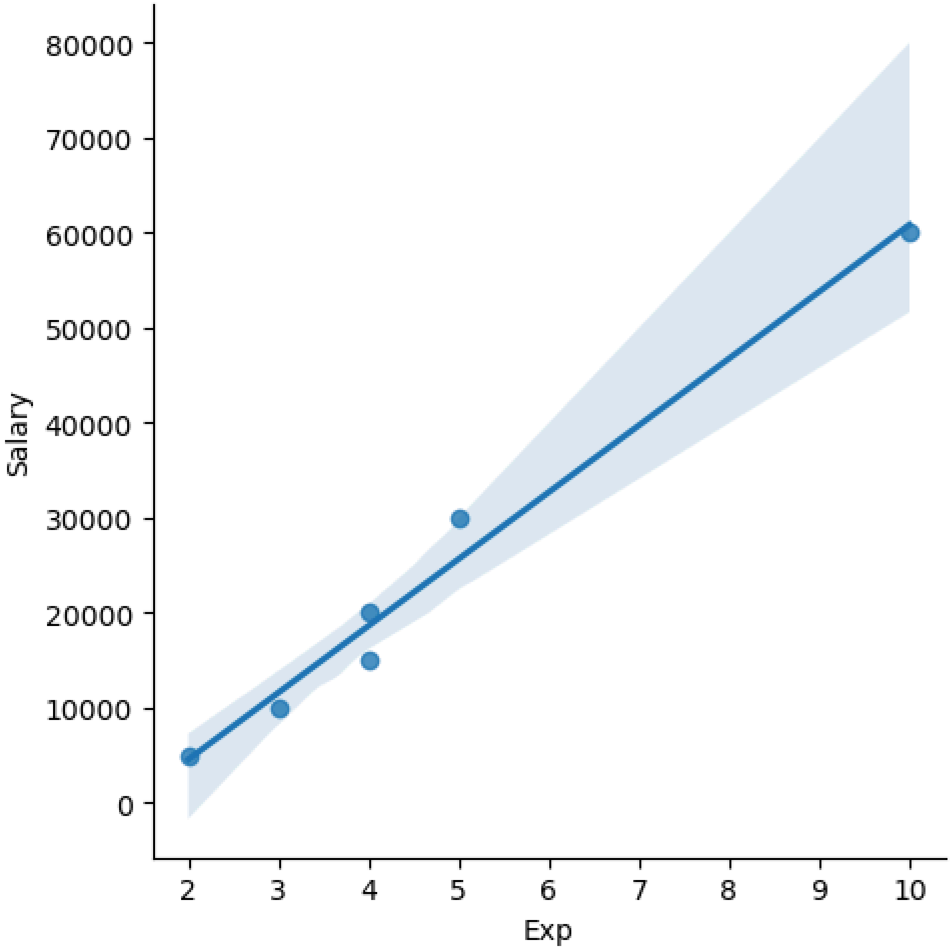


In [87]:
```
clean_data
```

Out[87]:

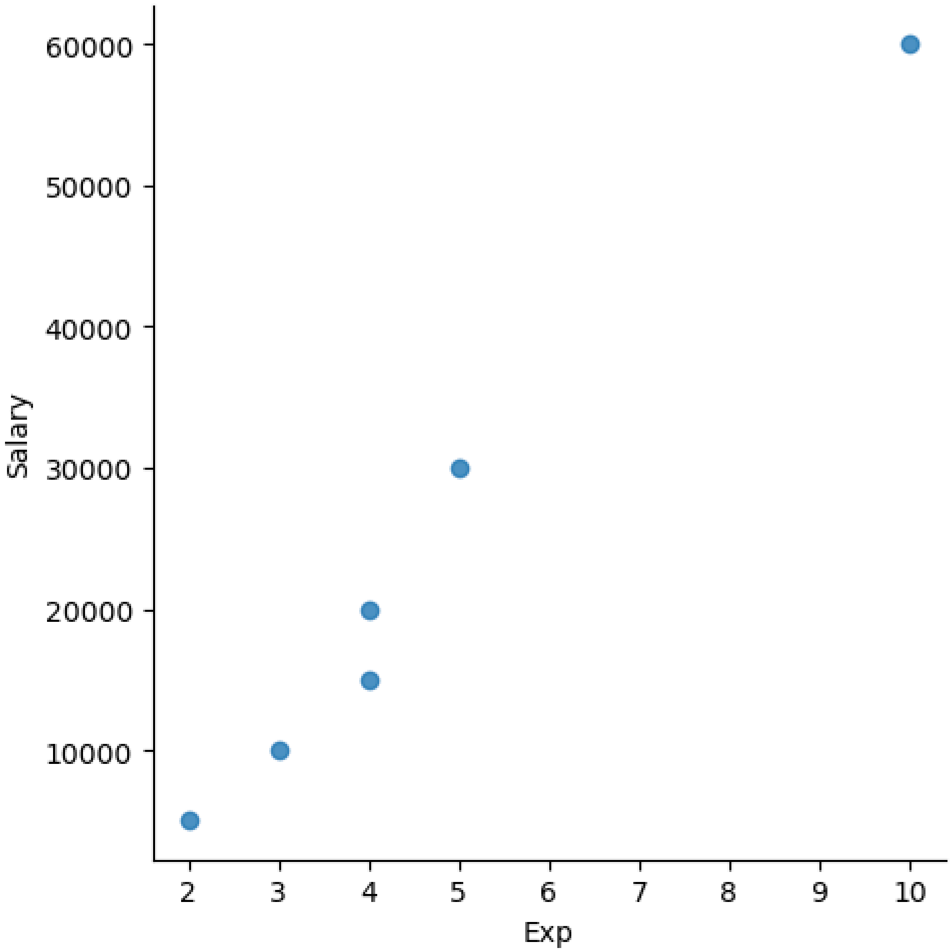
|  | Name | Domain | Age | Location | Salary | Exp |
|---|---|---|---|---|---|---|
| **0** | Mike | Datascience | 34 | Mumbai | 5000 | 2 |
| **1** | Teddy | Testing | 45 | Bangalore | 10000 | 3 |
| **2** | Umar | Dataanalyst | 50 | Bangalore | 15000 | 4 |
| **3** | Jane | Analytics | 50 | Hyderbad | 20000 | 4 |
| **4** | Uttam | Statistics | 67 | Bangalore | 30000 | 5 |
| **5** | Kim | NLP | 55 | Delhi | 60000 | 10 |

In [89]:
```
vis4 = sns.lmplot(data=clean_data,x = 'Exp', y='Salary')
```



In [90]:

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



In [91]:

```
clean_data[:]
```

Out[91]:

|   | 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 [92]:

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

Out[92]:

|   | 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 [93]:
```
clean_data[::-1]
```

Out[93]:

| | 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 |

In [94]:
```
clean_data.columns
```

Out[94]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [95]:
```
clean_data
```

Out[95]:

| | 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 [100…
```
X_iv = clean_data[['Name','Domain','Age','Location','Salary','Exp']]
```

In [97]:
```
X_iv
```

Out[97]:

| | 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 [98]:
```
y_dv = clean_data['Salary']
```

In [99]:
```
y_dv
```

Out[99]:
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: int32
```

In [101…
```
emp
```

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 | 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 [102…
```
clean_data
```

Out[102…

| | 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 [103…
```
X_iv
```

Out[103…

| | 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 [104…

```
y_dv
```

Out[104…

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

In [105…

```
clean_data
```

Out[105…

| | 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 [107…

```
imputation = pd.get_dummies(clean_data ,dtype=int)
```

In [108…

```
imputation
```

Out[108…

|   | Age | Salary | Exp | Name_Jane | Name_Kim | Name_Mike | Name_Teddy | Name_Umar | Nan |
|---|---|---|---|---|---|---|---|---|---|
| **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 [109…

```
clean_data
```

Out[109…

|   | 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 [110…

```
len(clean_data)
```

Out[110…

```
6
```

In [111…

```
imputation.columns
```

Out[111…

```
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 [112…

```
len(imputation.columns)
```

Out[112…

```
19
```

In [ ]: