EDA_Missing & Clean data& visualization_ MAY 20th

May 23, 2025

```
[554]: import pandas as pd
```

```
[556]: pd.__version__
```

```
[556]: '2.2.2'
```

```
[558]: emp=pd.read_excel(r'/Users/shashi/Downloads/Rawdata.xlsx')
```

```
[560]: emp
```

```
[560]:      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+
```

```
[562]: id(emp)
```

```
[562]: 5669291024
```

```
[564]: emp.columns
```

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

```
[566]: emp.shape
```

```
[566]: (6, 6)
```

```
[568]: emp.head()
```

```
[568]:      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
```

[570]:

```
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: 420.0+ bytes
```

[572]:

```
emp.tail()
```

[572]:

```
     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+
```

[574]:

```
emp.isnull()
```

[574]:

```
    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
```

[576]:

```
emp.isna()
```

[576]:

```
    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
```

[578]:

```
emp.isnull().sum()
```

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

0.1 DATA CLEANSING

```
[581]: emp['Name']
```

```
[581]: 0      Mike
       1    Teddy^
       2     Uma#r
       3      Jane
       4    Uttam*
       5       Kim
       Name: Name, dtype: object
```

```
[583]: emp['Name']=emp['Name'].str.replace(r'\W','',regex=True) # # removes the␣
         ↪special characters
```

```
[585]: emp['Name']
```

```
[585]: 0     Mike
       1    Teddy
       2     Umar
       3     Jane
       4    Uttam
       5      Kim
       Name: Name, dtype: object
```

```
[587]: emp['Domain']=emp['Domain'].str.replace(r'\W','',regex=True) # removes the␣
         ↪special characters
```

```
[589]: emp['Domain']
```

```
[589]: 0    Datascience
       1        Testing
       2    Dataanalyst
       3      Analytics
       4     Statistics
       5            NLP
       Name: Domain, dtype: object
```

```
[591]: emp['Age']=emp['Age'].str.replace(r'\W','',regex=True) # removes special␣
         ↪characters
```

[593]:
```
emp['Age']
```

[593]:
```
0    34years
1       45yr
2        NaN
3        NaN
4       67yr
5       55yr
Name: Age, dtype: object
```

[595]:
```
emp['Age']=emp['Age'].str.extract('(\d+)')  # extracts only digits
```

[597]:
```
emp['Age']
```

[597]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

[599]:
```
emp['Location']=emp['Location'].str.replace(r'\W','',regex=True)
```

[601]:
```
emp['Location']
```

[601]:
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

[603]:
```
emp['Salary']=emp['Salary'].str.replace(r'\W','',regex=True)
```

[605]:
```
emp['Salary']
```

[605]:
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

[607]:
```
emp['Exp']=emp['Exp'].str.extract('(\d+)')
```

[609]:
```
emp['Exp']
```

[609]:
```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

[611]:
```
emp
```

[611]:
```
    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
```

[613]:
```
clean_data=emp.copy()
```

[615]:
```
clean_data
```

[615]:
```
    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
```

1 MISSING VALUE TREATMENT

[618]:
```
clean_data
```

[618]:
```
    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
```

[620]:
```
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: 420.0+ bytes
```

[622]:
```
import numpy as np
```

[624]:
```
clean_data.head(1)
```

[624]:
```
   Name       Domain Age Location Salary Exp
0  Mike  Datascience  34   Mumbai   5000   2
```

[626]:
```
clean_data['Age']
```

[626]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

[628]:
```
clean_data['Age']=clean_data['Age'].fillna(np.mean(pd.
 ↪to_numeric(clean_data['Age'])))  # fills missing values with mean
```

[630]:
```
clean_data['Age']
```

[630]:
```
0       34
1       45
2    50.25
3    50.25
4       67
5       55
Name: Age, dtype: object
```

[632]:
```
clean_data['Exp']=clean_data['Exp'].fillna(np.mean(pd.
 ↪to_numeric(clean_data['Exp'])))  # fills missing values with mean
```

[634]:
```
clean_data['Exp']
```

```
[634]: 0      2
       1      3
       2      4
       3    4.8
       4      5
       5     10
       Name: Exp, dtype: object
```

```
[636]: clean_data
```

```
[636]:     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
```

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

```
[640]: clean_data['Location']
```

```
[640]: 0       Mumbai
       1    Bangalore
       2    Bangalore
       3     Hyderbad
       4    Bangalore
       5        Delhi
       Name: Location, dtype: object
```

```
[642]: 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
```

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

```
[646]: clean_data['Age']
```

```
[646]: 0    34
       1    45
       2    50
       3    50
       4    67
       5    55
       Name: Age, dtype: int64
```

```
[648]: clean_data
```

```
[648]:     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  4.8
       4  Uttam   Statistics   67  Bangalore  30000    5
       5    Kim          NLP   55      Delhi  60000   10
```

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

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

```
[654]: clean_data
```

```
[654]:     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
```

```
[656]: clean_data.info()  # object to int
```

```
<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      int64
 3   Location  6 non-null      object
 4   Salary    6 non-null      int64
 5   Exp       6 non-null      int64
dtypes: int64(3), object(3)
```

```
memory usage: 420.0+ bytes
```

```
[658]: clean_data['Name']=clean_data['Name'].astype('category')
```

```
[660]: clean_data['Name']
```

```
[660]: 0     Mike
       1    Teddy
       2     Umar
       3     Jane
       4    Uttam
       5      Kim
       Name: Name, dtype: category
       Categories (6, object): ['Jane', 'Kim', 'Mike', 'Teddy', 'Umar', 'Uttam']
```

```
[662]: clean_data['Domain']=clean_data['Domain'].astype('category')
```

```
[664]: clean_data['Domain']
```

```
[664]: 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']
```

```
[666]: clean_data['Location']=clean_data['Location'].astype('category')
```

```
[668]: clean_data['Location']
```

```
[668]: 0       Mumbai
       1    Bangalore
       2    Bangalore
       3     Hyderbad
       4    Bangalore
       5        Delhi
       Name: Location, dtype: category
       Categories (4, object): ['Bangalore', 'Delhi', 'Hyderbad', 'Mumbai']
```

```
[670]: clean_data
```

```
[670]:     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
```

[672]:
```
clean_data.to_csv('clean_data.csv')
```

[674]:
```
import os

current_directory = os.getcwd()
print("Current Working Directory:", current_directory)
```

```
Current Working Directory: /Users/shashi/Desktop/NARESH IT /Daily work
```

[676]:
```
#Imports excel file to our laptop

import os
os.getcwd()
```

[676]:
```
'/Users/shashi/Desktop/NARESH IT /Daily work '
```

[678]:
```
clean_data.columns
```

[678]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

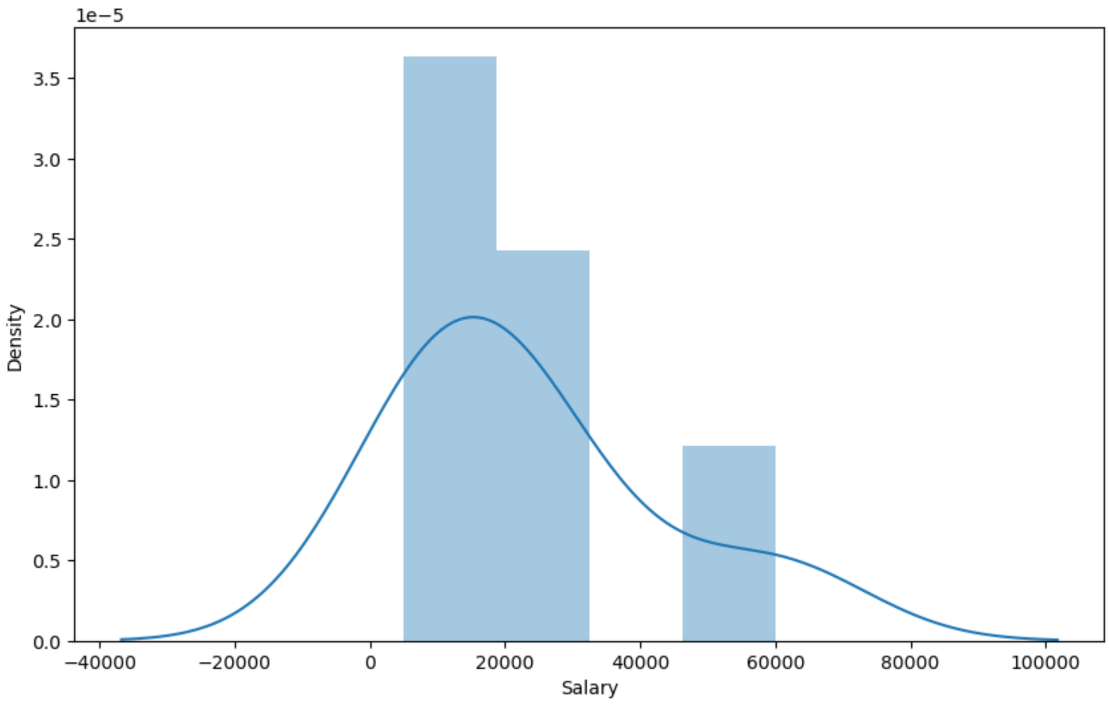
[680]:
```
import matplotlib.pyplot as plt  # visualization
import seaborn as sns    #advance visualization
```

[682]:
```
# Tells Python to suppress all warning messages.

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

[684]:
```
clean_data
```

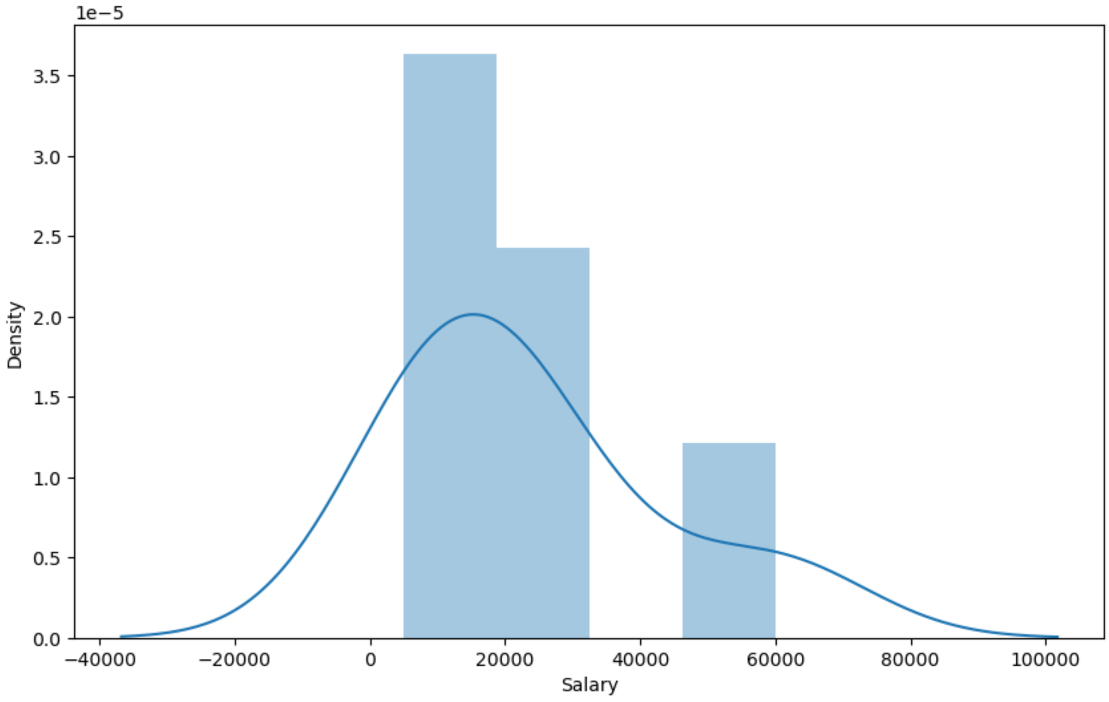
[684]:
```
    Name       Domain  Age   Location  Salary  Exp
0   Mike  Datascience   34     Mumbai    5000    2
1  Teddy      Testing   45  Bangalore   10000    3
2   Umar  Dataanalyst   50  Bangalore   15000    4
3   Jane    Analytics   50   Hyderbad   20000    4
4  Uttam   Statistics   67  Bangalore   30000    5
5    Kim          NLP   55      Delhi   60000   10
```

[686]:
```
vis1=sns.distplot(clean_data['Salary'])   # plots the salary
```



```
[688]: plt.rcParams['figure.figsize']=10,6  # Increases the height
```

```
[690]: vis1=sns.distplot(clean_data['Salary'])   # Increased the size of graph
```



[691]:

```
vis1=plt.distplot(clean_data['Salary']) # error as matplotlib doesnt have a␣
 ↪function distplot
```

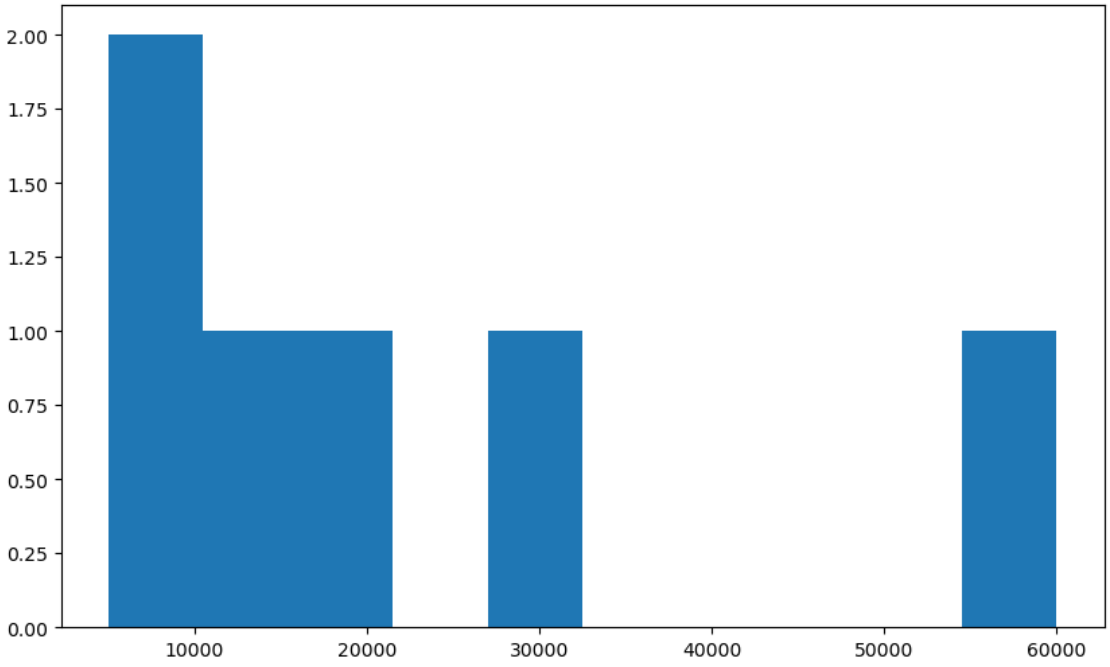
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[691], line 1
----> 1 vis1=plt.distplot(clean_data['Salary'])

AttributeError: module 'matplotlib.pyplot' has no attribute 'distplot'
```

[694]:

```
# OUTLIER DETECTION
# standing out from the rest is called outlier detection - anamoly detection
# Outlier will impact many classification algorithsm - KNN/logistic algorithms
# In the above graph we have salary outlier of 60k
```

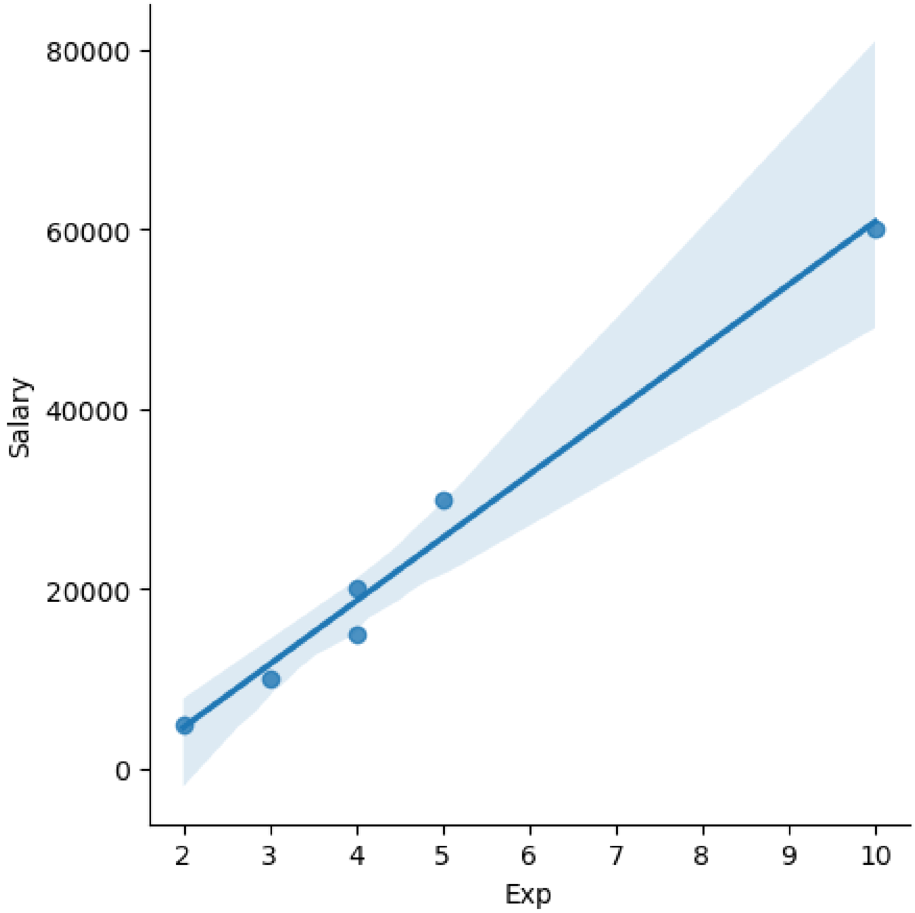
[696]:

```
vis2=plt.hist(clean_data['Salary'])  # we can identify the outlier 60k
```

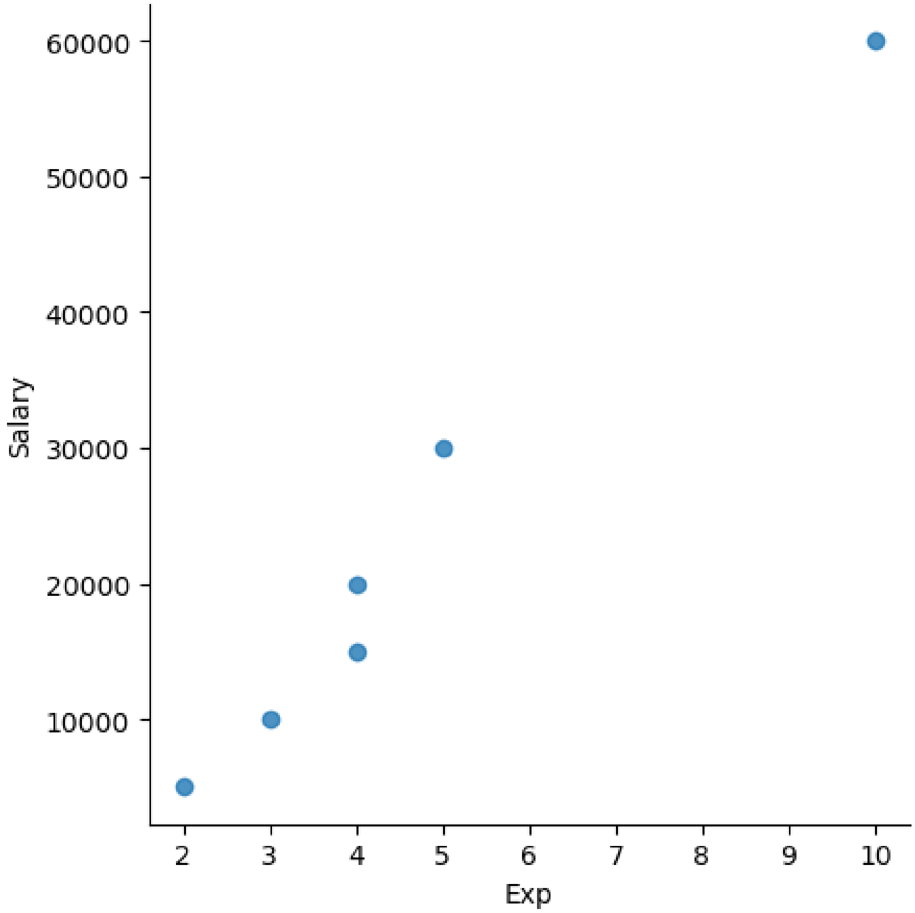


[698]:

```
# Linear Model Plot" - bivariate analysis

vis3=sns.lmplot(data=clean_data,x='Exp',y='Salary')
```

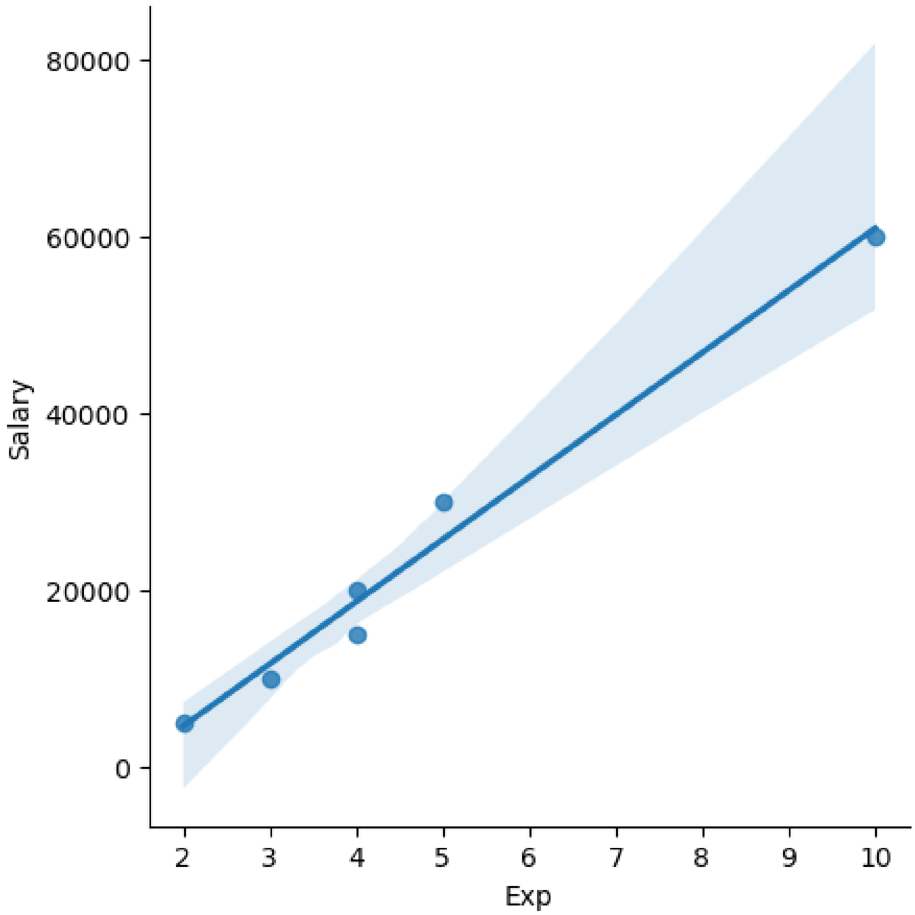


[700]:
```
# Setting fit_reg=False tells Seaborn to only plot the scatter plot (data␣
 ↪points) without fitting or drawing a regression line

vis4=sns.lmplot(data=clean_data,x='Exp',y='Salary',fit_reg=False)
```



[702]:

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



[704]:
```
clean_data[:2]
```

[704]:
```
    Name       Domain  Age   Location  Salary  Exp
0   Mike  Datascience   34     Mumbai    5000    2
1  Teddy      Testing   45  Bangalore   10000    3
```

[706]:
```
clean_data[:]
```

[706]:
```
    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
```

[708]:
```
clean_data[0:1]
```

```
[708]:    Name       Domain  Age Location  Salary  Exp
       0  Mike  Datascience   34   Mumbai    5000    2
```

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

```
[712]: clean_data
```

```
[712]:     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
```

```
[714]: x_iv
```

```
[714]:     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
```

```
[716]: x_iv.columns
```

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

```
[718]: clean_data.columns
```

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

```
[720]: clean_data
```

```
[720]:     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
```

```
[722]: y_dv=clean_data.drop(['Name', 'Domain', 'Age', 'Location', 'Exp'],axis=1) #␣
        ↪Only independent variable filter
```

```
[724]: y_dv
```

```
[724]:    Salary
       0    5000
       1   10000
       2   15000
       3   20000
       4   30000
       5   60000
```

```
[726]: clean_data
```

```
[726]:     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
```

```
[728]: x_iv # Independent variables
```

```
[728]:     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
```

```
[730]: y_iv # dependent vairables
```

```
[730]:    Salary
       0    5000
       1   10000
       2   15000
       3   20000
       4   30000
       5   60000
```

```
[732]: # imputations means value o & 1
```

```
[746]: import pandas as pd

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

```
[750]: imputation
```

```
[750]:    Age  Salary  Exp  Name_Jane  Name_Kim  Name_Mike  Name_Teddy  Name_Umar  \
       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

   Name_Uttam  Domain_Analytics  Domain_Dataanalyst  Domain_Datascience  \
0           0                 0                   0                   1
1           0                 0                   0                   0
2           0                 0                   1                   0
3           0                 1                   0                   0
4           1                 0                   0                   0
5           0                 0                   0                   0

   Domain_NLP  Domain_Statistics  Domain_Testing  Location_Bangalore  \
0           0                  0               0                   0
1           0                  0               1                   1
2           0                  0               0                   1
3           0                  0               0                   0
4           0                  1               0                   1
5           1                  0               0                   0

   Location_Delhi  Location_Hyderbad  Location_Mumbai
0               0                  0                1
1               0                  0                0
2               0                  0                0
3               0                  1                0
4               0                  0                0
5               1                  0                0
```

[ ]:

```
# NEXT STEPS - ML MODEL BUILDING
# FUTURE PREDICTION
# 3 LEVEL TESTS, DEPLOYMENT, AUTOMIZATION
```