

Raw Data To Clean Data Conversion Using Python EDA

In []:

Data cleaning how to implement eda technique on dataset to build ml model the steps which we follow today any interviewer ask the datacleaning .isna() - check missing value .fillna() - fill missing vlaue

In [2]: `import pandas as pd`

In [4]: `pd.__version__`

Out[4]: '2.2.2'

In [6]: `emp = pd.read_excel(r'E:\Data Science & AI\Dataset files\Rawdata.xlsx')`

In [8]: `emp`

Out[8]:

	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 [10]: `id(emp)`

Out[10]: 2556826072608

In [12]: `emp.columns`

Out[12]: Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')

In [14]: `emp.shape`

Out[14]: (6, 6)

In [16]: `emp.head()`

Out[16]:

	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 [18]: `emp.tail()`

Out[18]:

	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 [20]: `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
```

In [24]: `emp`

Out[24]:

	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 [22]: `emp['Domain']`

```
Out[22]: 0    Datascience#$
          1      Testing
          2    Dataanalyst^^#
          3      Ana^^lytics
          4      Statistics
          5          NLP
          Name: Domain, dtype: object
```

```
In [26]: emp.isnull()#emp.isna()
```

```
Out[26]:
```

	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 [28]: emp.isnull().sum()
```

```
Out[28]: Name      0
          Domain    0
          Age       2
          Location  2
          Salary    0
          Exp       1
          dtype: int64
```

Data Cleaning or Data Cleansing

```
In [30]: emp['Name']
```

```
Out[30]: 0    Mike
          1    Teddy^
          2    Uma#r
          3    Jane
          4    Uttam*
          5    Kim
          Name: Name, dtype: object
```

```
In [34]: emp['Name'] = emp['Name'].str.replace(r'\W', '', regex=True)#nonword charac
```

```
In [36]: emp['Name']
```

```
Out[36]: 0    Mike
          1    Teddy
          2    Umar
          3    Jane
          4    Uttam
          5    Kim
          Name: Name, dtype: object
```

```
In [42]: emp['Domain'] = emp['Domain'].str.replace(r'\W', '', regex=True)
```

```
In [44]: emp['Domain']
```

```
Out[44]: 0    Datascience
          1      Testing
          2    Dataanalyst
          3      Analytics
          4    Statistics
          5         NLP
          Name: Domain, dtype: object
```

```
In [46]: emp['Age'] = emp['Age'].str.replace(r'\W','',regex=True)
```

```
In [48]: emp['Age']
```

```
Out[48]: 0    34years
          1    45yr
          2      NaN
          3      NaN
          4    67yr
          5    55yr
          Name: Age, dtype: object
```

```
In [52]: emp['Age'] = emp['Age'].str.extract(r'(\d+)')
```

```
In [54]: emp['Age']
```

```
Out[54]: 0    34
          1    45
          2    NaN
          3    NaN
          4    67
          5    55
          Name: Age, dtype: object
```

```
In [56]: emp
```

```
Out[56]:
```

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

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

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

```
In [66]: emp['Salary'] = emp['Salary'].str.replace(r'\W', '', regex=True)
```

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

```
Out[68]: 0      5000
1     10000
2     15000
3     20000
4     30000
5     60000
Name: Salary, dtype: object
```

```
In [70]: emp
```

```
Out[70]:
```

	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	Hyderabad	20000	NaN
4	Uttam	Statistics	67	NaN	30000	5+ year
5	Kim	NLP	55	Delhi	60000	10+

```
In [72]: emp['Exp'] = emp['Exp'].str.extract(r'(\d+)')
```

```
In [74]: emp['Exp']
```

```
Out[74]: 0      2
1      3
2      4
3     NaN
4      5
5     10
Name: Exp, dtype: object
```

```
In [76]: emp
```

```
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	NaN	NaN	15000	4
3	Jane	Analytics	NaN	Hyderabad	20000	NaN
4	Uttam	Statistics	67	NaN	30000	5
5	Kim	NLP	55	Delhi	60000	10

```
In [78]: clean_data = emp.copy()
```

>>>>>>>Till now we have raw data we use regex to clean the data and removed all noise characted from the dataset
>>>>>>>you can also work in same things in sql query as well>Missing Values Treatment for Numerical data

```
In [80]: clean_data
```

```
Out[80]:
```

	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	Hyderabad	20000	NaN
4	Uttam	Statistics	67	NaN	30000	5
5	Kim	NLP	55	Delhi	60000	10

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

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

```
In [84]: import numpy as np
```

```
In [86]: clean_data['Age'] = clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['A
```

```
In [88]: clean_data['Age']
```

```
Out[88]: 0    34
1    45
2    50.25
3    50.25
4    67
5    55
Name: Age, dtype: object
```

```
In [90]: clean_data['Exp']
```

```
Out[90]: 0      2
         1      3
         2      4
         3    NaN
         4      5
         5     10
         Name: Exp, dtype: object
```

```
In [92]: clean_data['Exp'] = clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['E
```

```
In [94]: clean_data['Exp']
```

```
Out[94]: 0      2
         1      3
         2      4
         3    4.8
         4      5
         5     10
         Name: Exp, dtype: object
```

```
In [96]: clean_data
```

```
Out[96]:
```

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

```
Out[98]: 2
```

```
In [100... clean_data['Location']
```

```
Out[100... 0      Mumbai
         1    Bangalore
         2         NaN
         3    Hyderbad
         4         NaN
         5      Delhi
         Name: Location, dtype: object
```

```
In [102... clean_data['Location'] = clean_data['Location'].fillna(clean_data['Location'].mo
```

```
In [104... clean_data['Location']
```

```
Out[104...] 0      Mumbai
            1      Bangalore
            2      Bangalore
            3      Hyderabad
            4      Bangalore
            5        Delhi
            Name: Location, dtype: object
```

```
In [106...] clean_data
```

```
Out[106...]   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  Hyderabad  20000  4.8
4   Uttam   Statistics   67  Bangalore  30000    5
5     Kim      NLP     55     Delhi  60000   10
```

```
In [108...] 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 [110...] clean_data['Age'] = clean_data['Age'].astype(int)
```

```
In [112...] 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 [114...] clean_data['Salary'] = clean_data['Salary'].astype(int)
            clean_data['Exp'] = clean_data['Exp'].astype(int)
```


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      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 [118... `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 [120... `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 [122... `clean_data`

Out[122...

	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	Hyderabad	20000	4
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [124... `clean_data.to_csv('clean_data.csv')`

In [126... `import os`
`os.getcwd()`

Out[126... `'C:\\Users\\roy62\\Data Science & AI'`

In [128... clean_data

Out[128...

	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

EDA TECHNIQUE LETS APPLY

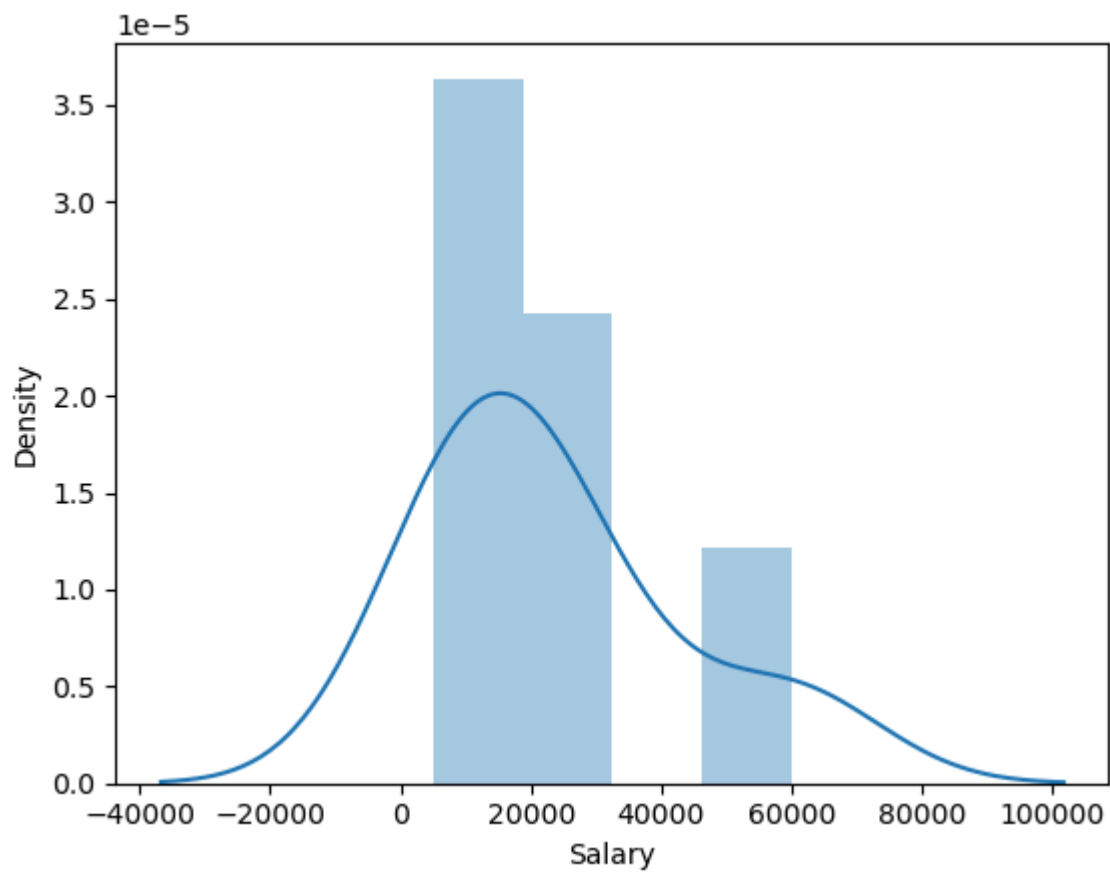
In [130... `import matplotlib.pyplot as plt # visualization`
`import seaborn as sns`

In [131... `import warnings`
`warnings.filterwarnings('ignore')`

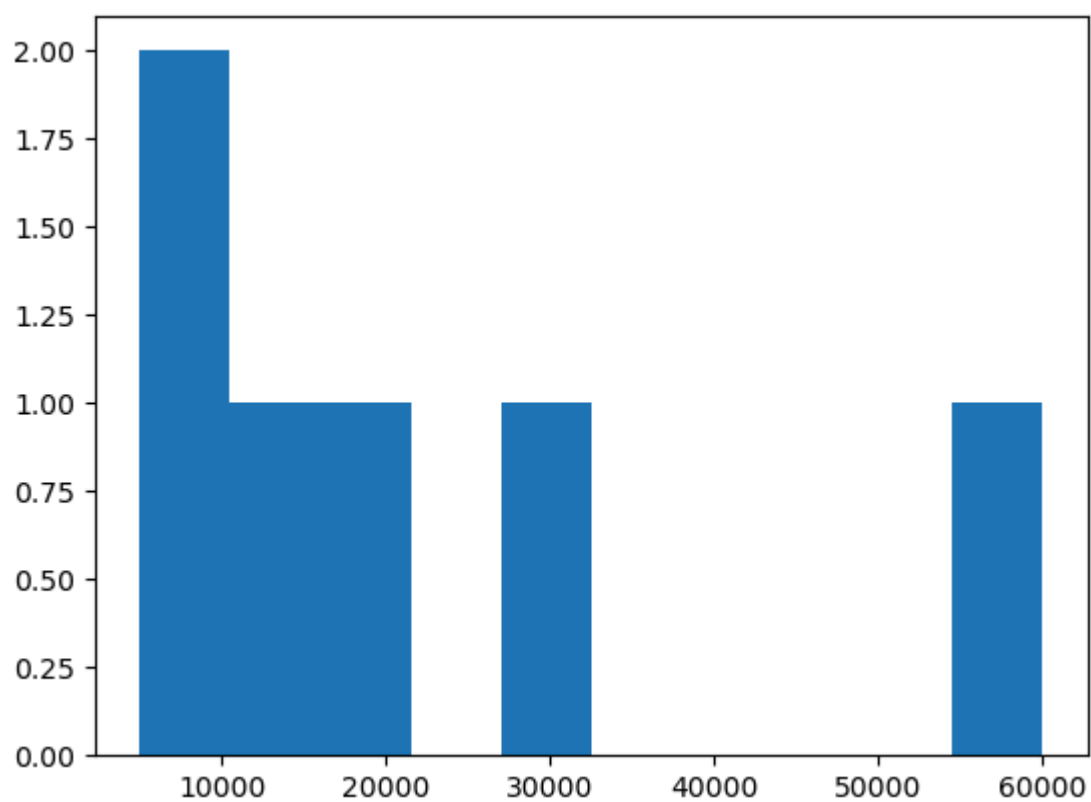
In [134... `clean_data['Salary']`

Out[134... 0 5000
1 10000
2 15000
3 20000
4 30000
5 60000
Name: Salary, dtype: int32

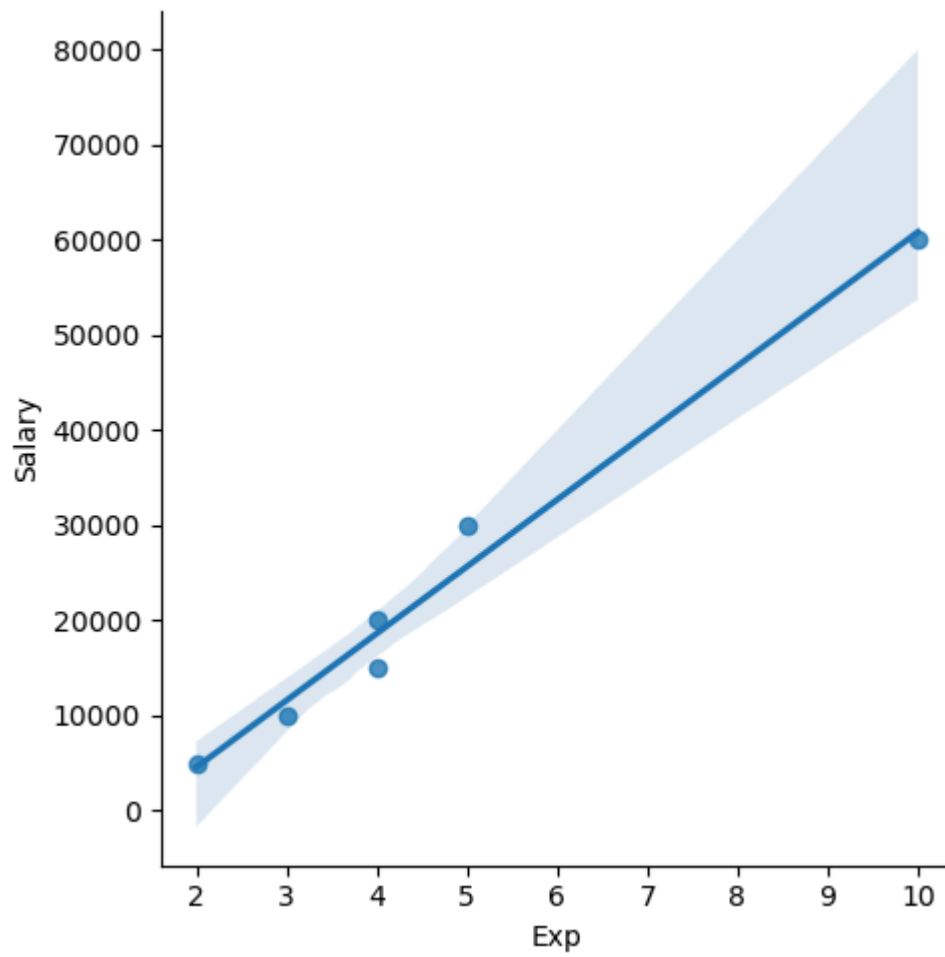
In [136... `vis1 = sns.distplot(clean_data['Salary'])`



In [138... `vis2 = plt.hist(clean_data['Salary'])`

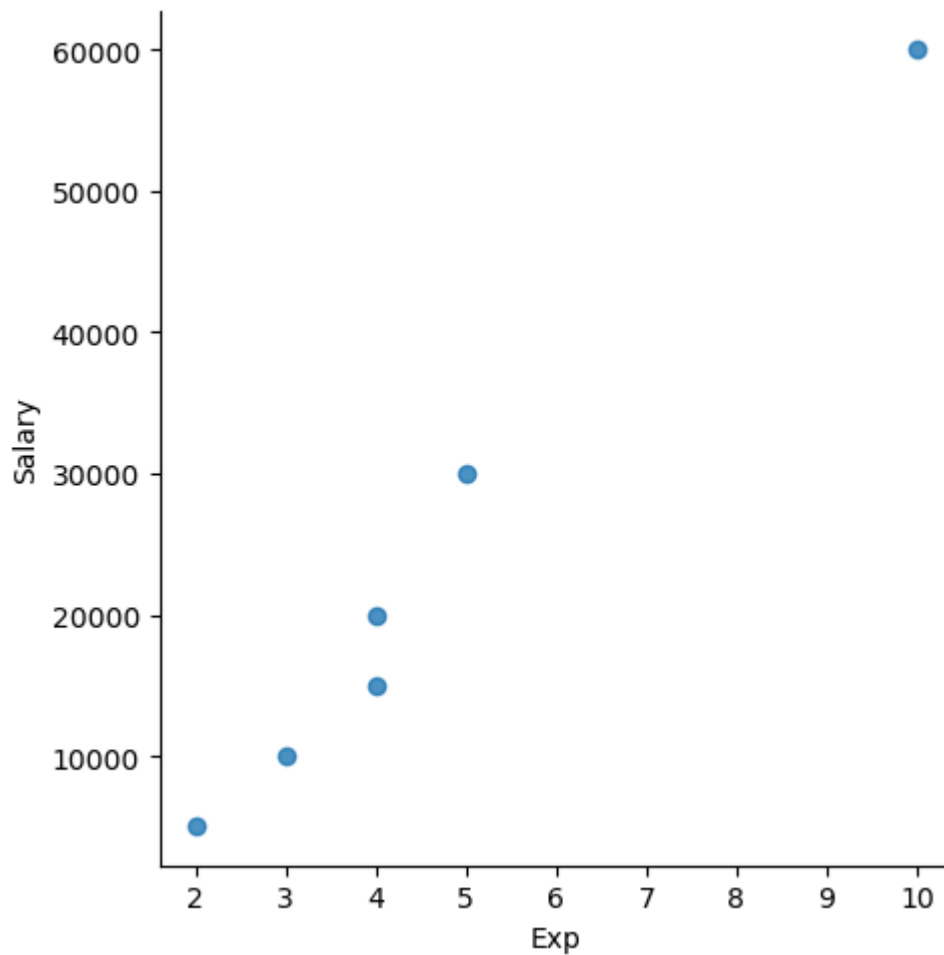


In [140... `vis4 = sns.lmplot(data=clean_data, x = 'Exp', y='Salary')`



In [142...

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



In [144... `clean_data[:]`

Out[144...

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

Out[146...

	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 [148... `clean_data[:, :-1]`

Out[148...

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

In [150...

```
clean_data.columns
```

Out[150...

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

In [152...

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

In [154...

```
X_iv
```

Out[154...

	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	Hyderabad	4
4	Uttam	Statistics	67	Bangalore	5
5	Kim	NLP	55	Delhi	10

In [156...

```
y_dv = clean_data[['Salary']]
```

In [158...

```
y_dv
```

Out[158...

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

In [160...

```
emp
```

Out[160...

	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 [162...

clean_data

Out[162...

	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 [164...

X_iv

Out[164...

	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 [166...

y_dv

Out[166...

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

In [168...

clean_data

Out[168...

	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 [170...

imputation = pd.get_dummies(clean_data)

In [172...

imputation

Out[172...

	Age	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Umar
0	34	5000	2	False	False	True	False	False
1	45	10000	3	False	False	False	True	False
2	50	15000	4	False	False	False	False	True
3	50	20000	4	True	False	False	False	False
4	67	30000	5	False	False	False	False	False
5	55	60000	10	False	True	False	False	False



In [174...

clean_data