Converting-raw-data-to-clean-data-using-Python-and-Exploratory-Data-Analysis-EDA

In [171…
```
import pandas as pd
```

In [172…
```
pd.__version__
```

Out[172…
```
'2.2.2'
```

In [173…
```
emp=pd.read_excel(r"C:\Users\admin\Downloads\Rawdata.xlsx")
```

In [174…
```
emp
```

Out[174…

	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 [175…
```
#nan means missing values
#and data is not cleaned
```

In [176…
```
id(emp)
```

Out[176…
```
2117413066592
```

In [177…
```
emp.columns
```

Out[177…
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [178…
```
emp.shape
```

Out[178…
```
(6, 6)
```

In [179…
```
emp.head()
```

Out[179…

	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 [180…
```
emp.tail()
```

Out[180…

	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 [181…
```
emp.info()
#info of dataframes
```

```
<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 [182…
```
emp
```

Out[182...

	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 [183...

```
emp.isnull()
#gives t or false
```

Out[183...

	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 [184...

```
emp.isnull().sum()
#shows missing value of columns
```

Out[184...

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

In [185...

```
emp.columns
```

Out[185...

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

In [186...

```
emp
```

Out[186…

	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 [1]:
```
#data cleaning or cleansing
```

In [188…
```
emp['Name']
```

Out[188…
```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

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

regex comma fullstop hash etc

Explanation: r'\W' matches any non-word character (equivalent to [^a-zA-Z0-9_]). '' to remove extra space regex=True specifies that you are using a regular expression.

In [190…
```
emp['Name']
```

Out[190…
```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

In [191…
```
emp
```

Out[191…

	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 [192…
```
emp['Domain']
```

Out[192…
```
0     Datascience#$
1           Testing
2    Dataanalyst^^#
3       Ana^^lytics
4        Statistics
5               NLP
Name: Domain, dtype: object
```

In [193…
```
emp['Domain']=emp['Domain'].str.replace(r'\W','',regex=True)
```

In [194…
```
emp['Domain']
```

Out[194…
```
0    Datascience
1        Testing
2    Dataanalyst
3      Analytics
4     Statistics
5            NLP
Name: Domain, dtype: object
```

In [195…
```
emp['Age']
```

Out[195…
```
0    34 years
1      45' yr
2         NaN
3         NaN
4       67-yr
5        55yr
Name: Age, dtype: object
```

In [196…
```
emp['Age']=emp['Age'].str.replace(r'\W','',regex=True)
```

In [197…
```
emp['Age']
```

Out[197…
```
0    34years
1       45yr
2        NaN
3        NaN
4       67yr
5       55yr
Name: Age, dtype: object
```

In [198…
```
emp['Age']=emp['Age'].str.extract('(\d+)')
```

```
<>:1: SyntaxWarning: invalid escape sequence '\d'
<>:1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\admin\AppData\Local\Temp\ipykernel_19984\3771958390.py:1: SyntaxWarning: in
valid escape sequence '\d'
  emp['Age']=emp['Age'].str.extract('(\d+)')
```

In [200…
```
emp['Salary']
```

Out[200…
```
0     5^00#0
1    10%%000
2    1$5%000
3     2000^0
4     30000-
5    6000^$0
Name: Salary, dtype: object
```

In [201…
```
emp['Salary']=emp['Salary'].str.replace(r'\W','',regex=True)
```

In [202…
```
emp['Salary']
```

Out[202…
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

In [203…
```
emp['Exp']
```

Out[203…
```
0         2+
1         <3
2     4> yrs
3        NaN
4    5+ year
5        10+
Name: Exp, dtype: object
```

In [204…
```
emp['Exp']=emp['Exp'].str.replace(r'\W','',regex=True)
```

In [205…
```
emp['Exp']
```

Out[205…
```
0        2
1        3
2     4yrs
3      NaN
4    5year
5       10
Name: Exp, dtype: object
```

In [206…
```
emp
```

Out[206…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	Dataanalyst	NaN	NaN	15000	4yrs
3	Jane	Analytics	NaN	Hyderbad	20000	NaN
4	Uttam*	Statistics	67	NaN	30000	5year
5	Kim	NLP	55	Delhi	60000	10

In [207…
```
emp['Age']
```

Out[207…
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [208…
```
emp['Location']
```

Out[208…
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [209…
```
emp['Location']=emp['Location'].str.replace(r'\W','',regex=True)
```

In [210…
```
emp['Location']
```

Out[210…
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [211…
```
emp.head()
```

Out[211…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	Dataanalyst	NaN	NaN	15000	4yrs
3	Jane	Analytics	NaN	Hyderbad	20000	NaN
4	Uttam*	Statistics	67	NaN	30000	5year

In [212…
```
emp['Exp']
```

Out[212…
```
0        2
1        3
2     4yrs
3      NaN
4    5year
5       10
Name: Exp, dtype: object
```

In [213…
```
emp['Exp']=emp['Exp'].str.extract('(\d+)')
```

```
<>:1: SyntaxWarning: invalid escape sequence '\d'
<>:1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\admin\AppData\Local\Temp\ipykernel_19984\1466635560.py:1: SyntaxWarning: in
valid escape sequence '\d'
  emp['Exp']=emp['Exp'].str.extract('(\d+)')
```

In [214…
```
emp['Exp']
```

Out[214…
```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

In [215…
```
emp
```

Out[215…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	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 [216…
```
clean_data=emp.copy()
```

In [217…
```
clean_data
```

Out[217…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	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 [218…
```
clean_data.isnull().sum()
##eda and gets missing vakue data
```

Out[218…
```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

In [219…
```
#if numerical data is misiing we use mean median mode
#categorical means mode
#fill na give fill missing value
```

In [220…
```
clean_data['Age']
```

Out[220…
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [221…
```
import numpy as np
```

In [222…
```
clean_data['Age']=clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['Age'])
```

In [223…
```
clean_data['Age']
```

Out[223…
```
0       34
1       45
2    50.25
3    50.25
4       67
5       55
Name: Age, dtype: object
```

In [224…
```
clean_data['Exp']
```

Out[224…
```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

In [225…
```
clean_data['Exp']=clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['Exp'])
```

In [226…
```
clean_data['Exp']
```

Out[226…
```
0      2
1      3
2      4
3    4.8
4      5
5     10
Name: Exp, dtype: object
```

In [227…
```
clean_data
```

Out[227...

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	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 [228...
```
clean_data['Location'].isnull().sum()
```

Out[228...
```
2
```

In [229...
```
clean_data['Location']
```

Out[229...
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [230...
```
clean_data['Location']=clean_data['Location'].fillna(clean_data['Location'].mode()[
```

In [231...
```
clean_data['Location']
```

Out[231...
```
0       Mumbai
1    Bangalore
2    Bangalore
3     Hyderbad
4    Bangalore
5        Delhi
Name: Location, dtype: object
```

In [232...
```
clean_data
```

Out[232...

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	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 [233…
```
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 [234…
```
clean_data['Age'] = clean_data['Age'].astype(int)
clean_data['Salary'] = clean_data['Salary'].astype(int)
clean_data['Exp'] = clean_data['Exp'].astype(int)
```

In [235…
```
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 [236…
```
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 [237…
```
clean_data
```

Out[237…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	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 [239…
```
clean_data.to_csv('CLEANED_DATA@MAruth.csv')
#Now lets save this cleaned data as a new csv file with name
#LEANED_DATA
```

In [240…
```
clean_data
```

Out[240…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	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 [241…
```
import os
os.getcwd()
```

Out[241…
```
'C:\\Users\\admin'
```

In [242…
```
clean_data
```

Out[242…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	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 [142…
```
EDA TEchnique
```

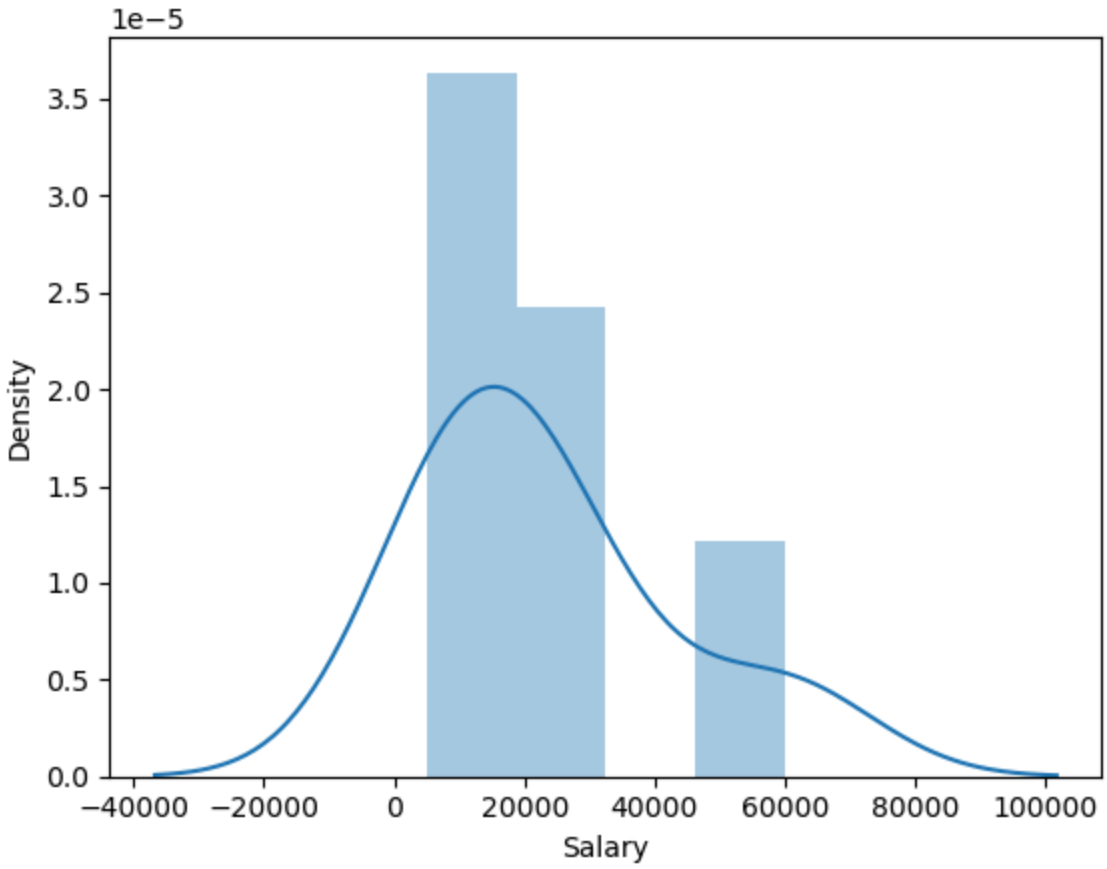
In [243…
```
import matplotlib.pyplot as plt # matplotlib.pyplot is a powerful tool for data vis
import seaborn as sns #Seaborn is a statistical data visualization library built o
```

In [244…
```
import warnings
warnings.filterwarnings('ignore')
```

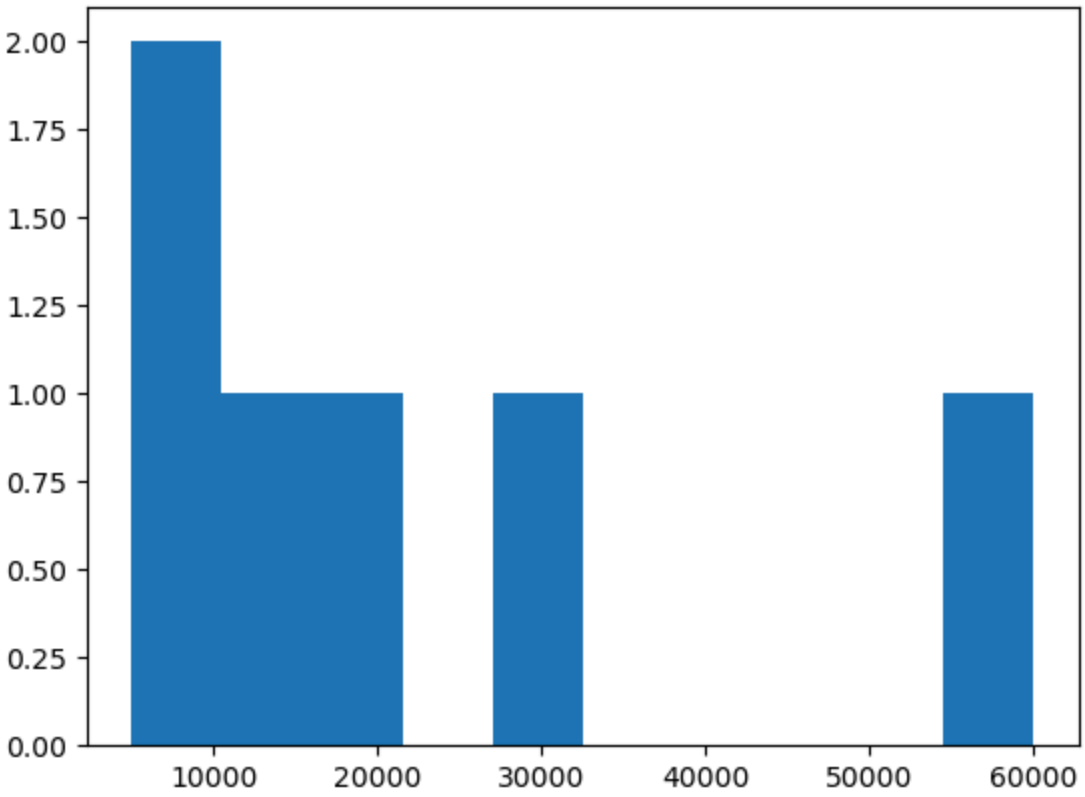
In [245…
```
clean_data['Salary']
```

Out[245…
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: int32
```

In [246…
```
vis1 = sns.distplot(clean_data['Salary'])
```

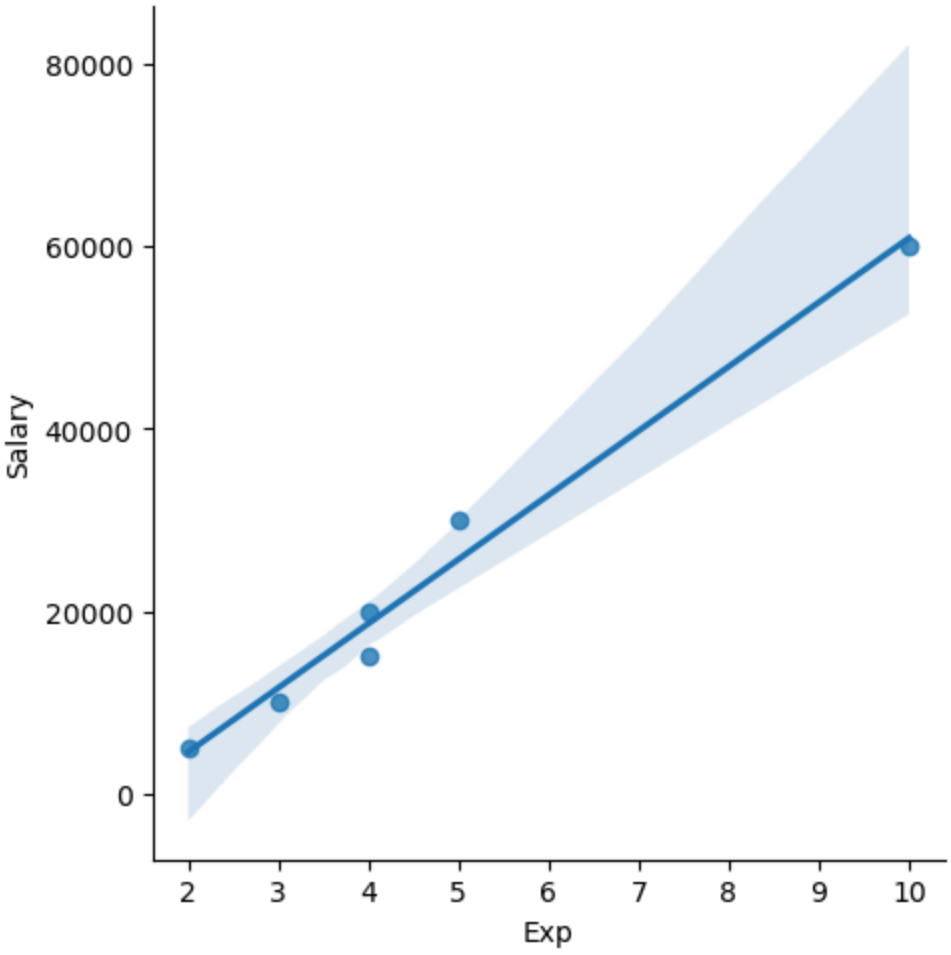


In [247…
```
vis2 = plt.hist(clean_data['Salary'])
```

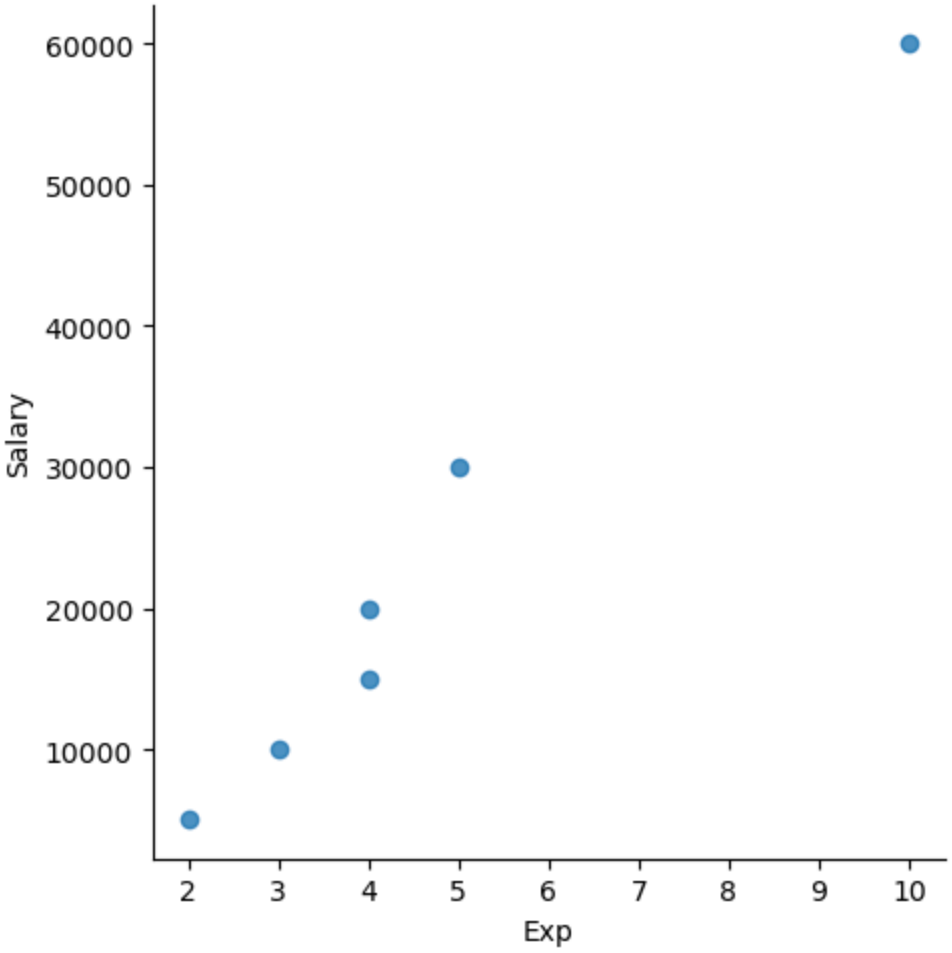


In [248…

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



In [249…
```
vis5 = sns.lmplot(data = clean_data, x = 'Exp', y = 'Salary', fit_reg = False)
```



In [250…
```
clean_data[:]
```

Out[250…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	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 [251…
```
clean_data[0:6:2]
```

Out[251...

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
2	Uma#r	Dataanalyst	50	Bangalore	15000	4
4	Uttam*	Statistics	67	Bangalore	30000	5

In [252...
```
clean_data[::-1]
```

Out[252...

	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	Uma#r	Dataanalyst	50	Bangalore	15000	4
1	Teddy^	Testing	45	Bangalore	10000	3
0	Mike	Datascience	34	Mumbai	5000	2

In [253...
```
clean_data.columns
```

Out[253...
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [255...
```
x_iv = clean_data[['Name', 'Domain', 'Age', 'Location', 'Exp']]
```

In [256...
```
x_iv
```

Out[256...

	Name	Domain	Age	Location	Exp
0	Mike	Datascience	34	Mumbai	2
1	Teddy^	Testing	45	Bangalore	3
2	Uma#r	Dataanalyst	50	Bangalore	4
3	Jane	Analytics	50	Hyderbad	4
4	Uttam*	Statistics	67	Bangalore	5
5	Kim	NLP	55	Delhi	10

In [257...
```
y_dv = clean_data[['Salary']]
```

In [258...
```
y_dv
```

Out[258...

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

In [259...
```
emp
```

Out[259...

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	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 [260...
```
clean_data
```

Out[260...

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy^	Testing	45	Bangalore	10000	3
2	Uma#r	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 [261...
```
imputation=pd.get_dummies(clean_data)
```

In [262...
```
imputation
```

Out[262...

	Age	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy^	Name_Uma#r	N
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 [263...
```
imputation = pd.get_dummies(clean_data).astype(int)
```

In [264...
```
imputation
```

Out[264...

	Age	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy^	Name_Uma#r	N
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 [ ]:
```
```