Raw data to Clean Data

In [2]:
```
import pandas as pd
```

In [6]:
```
emp = pd.read_excel(r'/Users/babarhussain/Documents/Rawdata.xlsx')
emp
```

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
5	Kim	NLP	55yr	Delhi	6000^$0	10+

In [8]:
```
emp.shape
```

Out[8]:
```
(6, 6)
```

In [10]:
```
emp.columns
```

Out[10]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='obj
ect')
```

In [12]:
```
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 [14]:
```
emp.isnull()
```

Out[14]:

	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 [18]:
```
emp.isnull().any().any()
```

Out[18]: True

In [20]:
```
emp.describe()
```

Out[20]:

	Name	Domain	Age	Location	Salary	Exp
count	6	6	4	4	6	5
unique	6	6	4	4	6	5
top	Mike	Datascience#$	34 years	Mumbai	5^00#0	2+
freq	1	1	1	1	1	1

cleaning Data

In [23]:
```
emp
```

Out[23]:

	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 [25]:
```
emp['Name']
```

Out[25]:
```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

In [27]:
```
emp['Name'] = emp['Name'].str.replace(r'\W','',regex=True)
emp['Name']
```

Out[27]:
```
0     Mike
1    Teddy
2     Umar
3     Jane
4    Uttam
5      Kim
Name: Name, dtype: object
```

In [29]:
```
emp['Domain'] = emp['Domain'].str.replace(r'\W','', regex=True)
emp['Domain']
```

Out[29]:
```
0    Datascience
1        Testing
2    Dataanalyst
3      Analytics
4     Statistics
5            NLP
Name: Domain, dtype: object
```

In [31]:
```
emp['Age']
```

Out[31]:
```
0    34 years
1      45' yr
2         NaN
3         NaN
4       67-yr
5        55yr
Name: Age, dtype: object
```

In [33]:
```
emp['Age'] = emp['Age'].replace(r'^0-9.','',regex=True)
emp['Age']
```

Out[33]:
```
0    34 years
1      45' yr
2         NaN
3         NaN
4       67-yr
5        55yr
Name: Age, dtype: object
```

In [35]:
```
emp['Age'] = emp['Age'].replace(r'[^0-9.]','',regex=True)
emp['Age']
```

Out[35]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [37]:
```
emp
```

Out[37]:

	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 [39]:
```
emp['Salary']
```

Out[39]:
```
0     5^00#0
1    10%%000
2    1$5%000
3     2000^0
4     30000-
5    6000^$0
Name: Salary, dtype: object
```

In [43]:
```
emp['Salary'] = emp['Salary'].replace(r'[^0-9.]','',regex=True)
emp['Salary']
```

Out[43]:
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

In [45]:
```
emp['Exp']
```

Out[45]:
```
0         2+
1         <3
2     4> yrs
3        NaN
4    5+ year
5        10+
Name: Exp, dtype: object
```

In [47]:
```
emp['Exp'] = emp['Exp'].replace(r'[^0-9.]','',regex=True)
emp['Exp']
```

Out[47]:
```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

In [49]:
```
emp
```

Out[49]:

	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 [51]:
```
clean_data = emp.copy()
clean_data
```

Out[51]:

	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

EDA Technique Apply

find Missing Values

In [55]:
```
clean_data
```

Out[55]:

	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 [57]:
```
clean_data.isnull().any().any()
```

Out[57]: True

In [59]:
```
clean_data.isnull()
```

Out[59]:

	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 [61]:
```
clean_data.isna()
```

Out[61]:

	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 [63]:
```
clean_data.isnull().sum()
```

Out[63]:
```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

In [65]:
```
clean_data['Age']
```

Out[65]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [67]:
```
import numpy as np
```

In [71]:
```
clean_data['Age'] = clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_
clean_data['Age']
```

Out[71]:
```
0       34
1       45
2    50.25
3    50.25
4       67
5       55
Name: Age, dtype: object
```

In [73]:
```
clean_data
```

Out[73]:

	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 [75]:
```
emp['Exp'].isnull()
```

Out[75]:
```
0    False
1    False
2    False
3     True
4    False
5    False
Name: Exp, dtype: bool
```

In [83]:
```
clean_data['Exp'] = emp['Exp'].fillna(np.mean(pd.to_numeric(clean_data['E
clean_data['Exp']
```

Out[83]:
```
0      2
1      3
2      4
3    4.8
4      5
5     10
Name: Exp, dtype: object
```

In [85]:
```
clean_data
```

Out[85]:

	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 [87]:
```
emp
```

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	NaN	NaN	15000	4
3	Jane	Analytics	NaN	Hyderbad	20000	4.8
4	Uttam	Statistics	67	NaN	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [89]:
```
clean_data['Location'] = clean_data['Location'].fillna(clean_data['Locati
clean_data['Location']
```

Out[89]:
```
0       Mumbai
1    Bangalore
2    Bangalore
3     Hyderbad
4    Bangalore
5        Delhi
Name: Location, dtype: object
```

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	50.25	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 [93]:
```
clean_data.isnull().any().any()
```

Out[93]:
```
False
```

In [95]:
```
clean_data.isnull().sum()
```

Out[95]:
```
Name        0
Domain      0
Age         0
Location    0
Salary      0
Exp         0
dtype: int64
```

In [99]:
```
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 [103…
```
clean_data[['Age','Salary','Exp']] = clean_data[['Age','Salary','Exp']].a
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      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
```

In [107…
```
clean_data[['Name','Domain','Location']] = clean_data[['Name','Domain','L
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      int64   
 3   Location  6 non-null      category
 4   Salary    6 non-null      int64   
 5   Exp       6 non-null      int64   
dtypes: category(3), int64(3)
memory usage: 938.0 bytes
```

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 [119…
```
clean_data.to_csv('clean_data.csv', index=False)
```

In [121…
```
import os
os.getcwd()
```

Out[121… '/Users/babarhussain/EDA Example'

In [123…
```
clean_data
```

Out[123…

	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

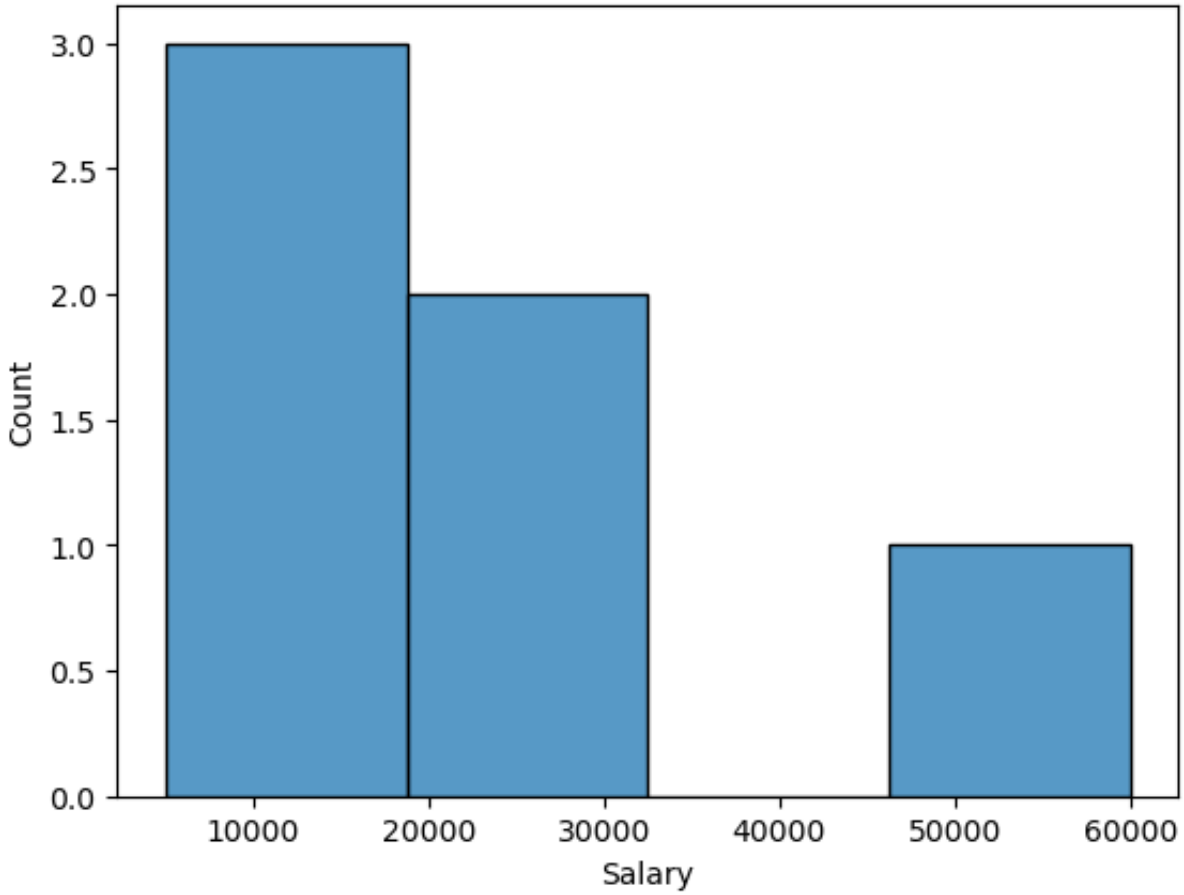
In [125…
```
import matplotlib.pyplot as plt
import seaborn as ss
```

In [127…
```
import warnings
warnings.filterwarnings('ignore')
```

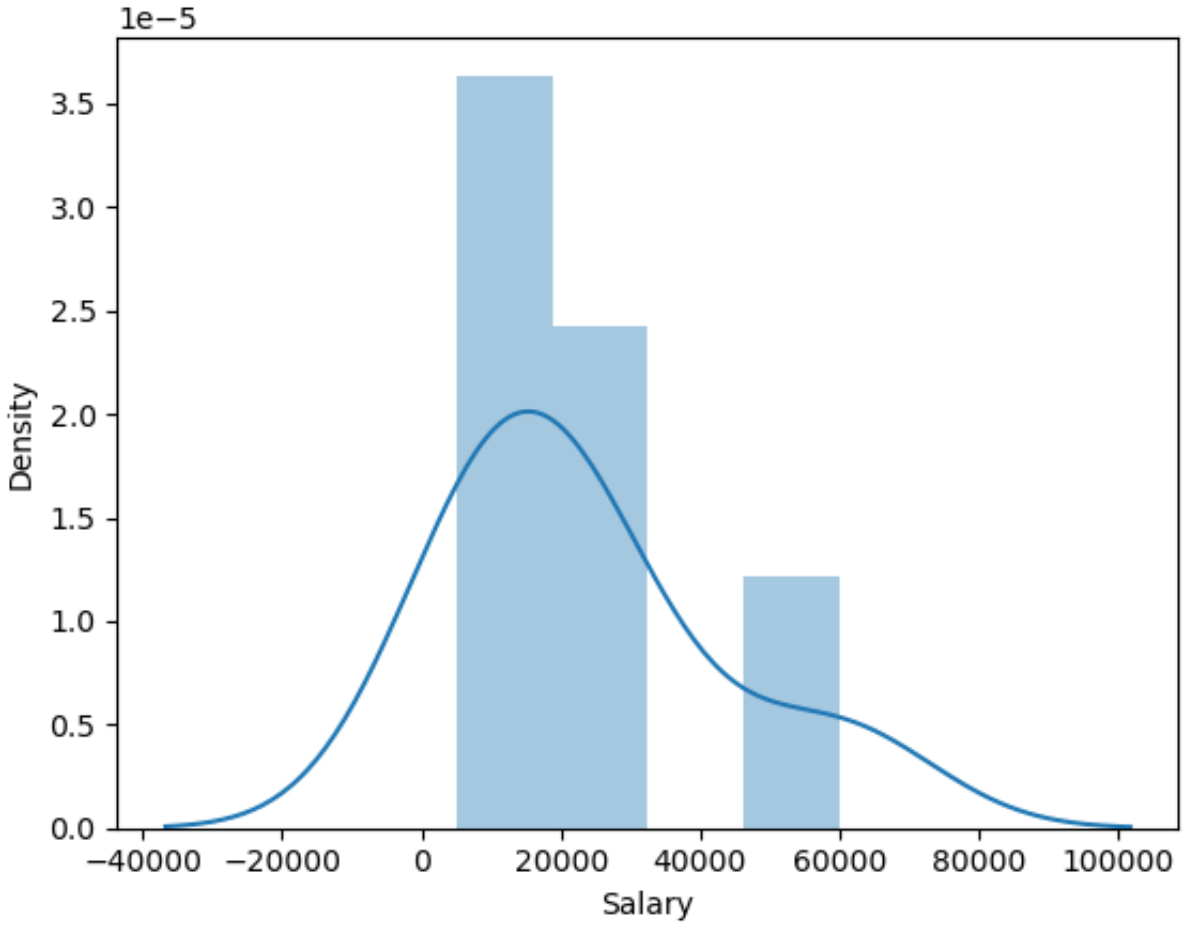
In [129…
```
clean_data['Salary']
```

Out[129…
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: int64
```

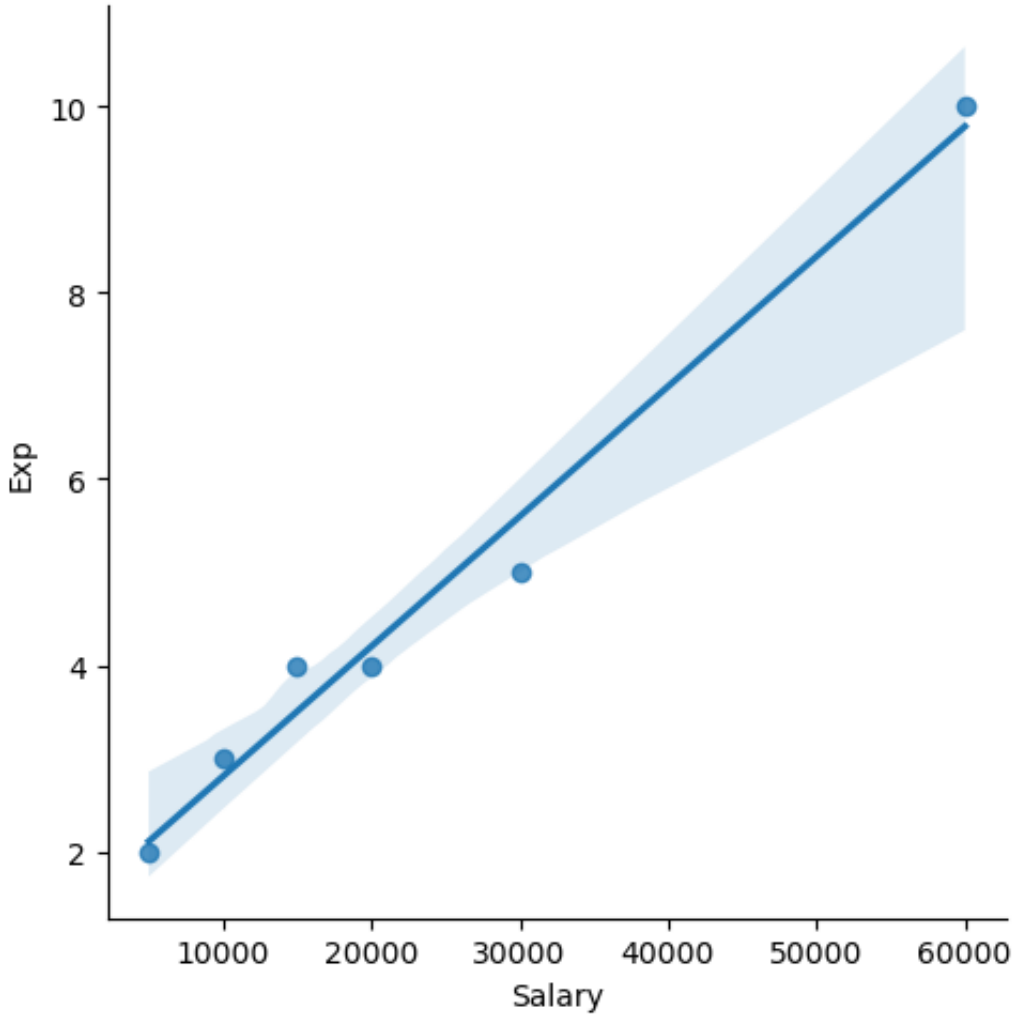
In [131…
```
vs = ss.histplot(clean_data['Salary']) #Outlier Treatment
```



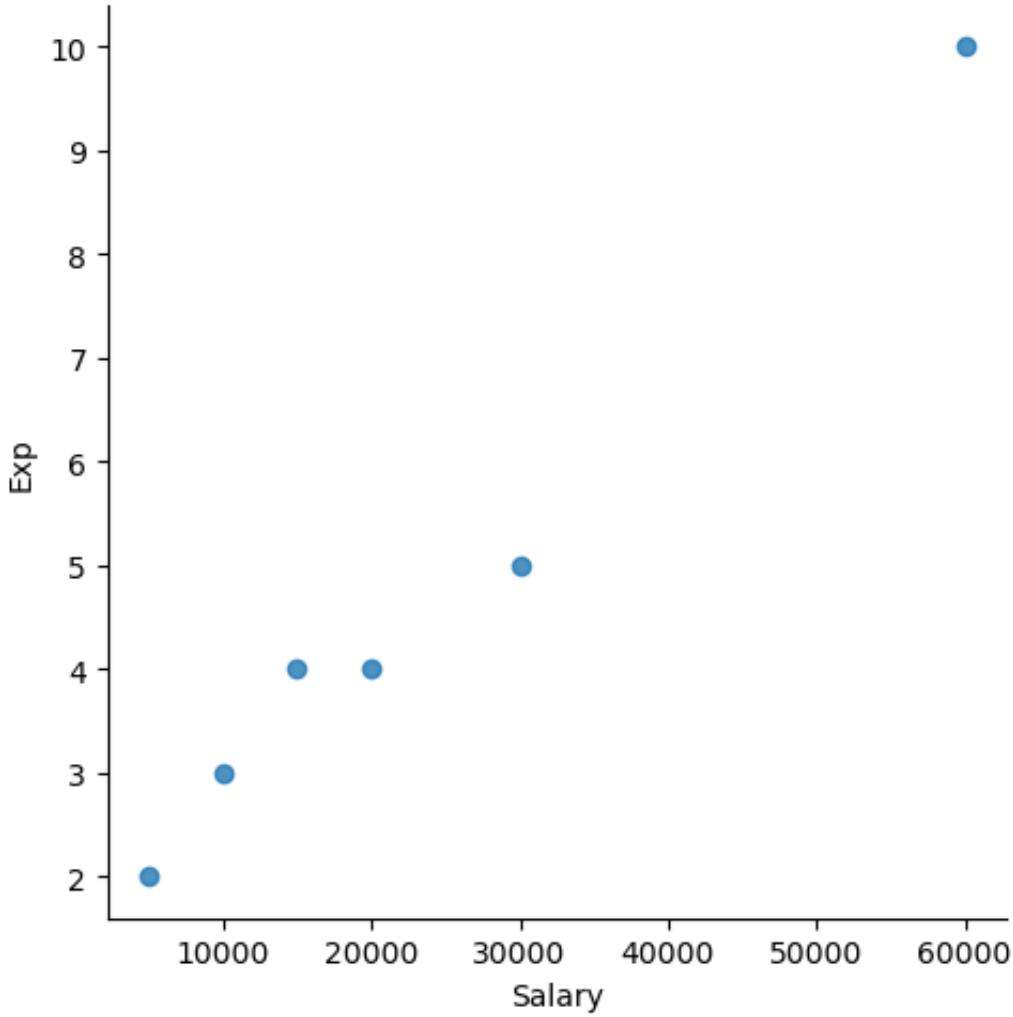
In [135…
```
vs1 = ss.distplot(clean_data['Salary']) # univariate Analysis
```



In [137…
```
vs3 = ss.lmplot(data=clean_data, x='Salary', y='Exp') # Bivariate Analysi
```



In [141…
```
vs3 = ss.lmplot(data=clean_data, x='Salary', y='Exp', fit_reg =False) # B
```



In [143…
```
# Variable Identification
```

In [145…
```
x_vr = clean_data[['Name','Domain','Age','Location','Exp']] # here all ar
x_vr
```

Out[145…

	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 [147…
```
y_vr = clean_data[['Salary']] # here Slary is dependent variable
y_vr
```

Out[147…

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

In [151…

```
imputation = pd.get_dummies(clean_data, dtype=int) #dummy variables (also
imputation
```

Out[151…

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

In [155…

```
imputation = pd.get_dummies(clean_data, columns=['Name'], dtype=int,drop_
imputation
```

Out[155…

	Domain	Age	Location	Salary	Exp	Name_Jane	Name_Kim	Name_Mike
0	Datascience	34	Mumbai	5000	2	0	0	1
1	Testing	45	Bangalore	10000	3	0	0	0
2	Dataanalyst	50	Bangalore	15000	4	0	0	0
3	Analytics	50	Hyderbad	20000	4	1	0	0
4	Statistics	67	Bangalore	30000	5	0	0	0
5	NLP	55	Delhi	60000	10	0	1	0

In [ ]:

```
```

In [ ]:

```
```