

EDA TECHNIQUE

```
In [697... import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
In [699... import warnings
```

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

```
In [701... pd.__version__
```

```
Out[701... '2.2.2'
```

```
In [703... emp = pd.read_excel(r"D:\FSDS Material\Dataset\EDA.xlsx")
```

```
In [705... emp
```

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascienc#\$	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 [707... id(emp)
```

```
Out[707... 1856296473840
```

```
In [709... emp.columns
```

```
Out[709... Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

```
In [711... emp.shape
```

```
Out[711... (6, 6)
```

```
In [713... emp.head()
```

```
Out[713...

|   | 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 [715... emp.tail()
```

```
Out[715...

|   | 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 [717... 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 [719... `emp['Domain']`

```
Out[719... 0      Datascience#$
 1          Testing
 2      Dataanalyst^^#
 3      Ana^^lytics
 4          Statistics
 5          NLP
Name: Domain, dtype: object
```

In [721... `emp.isnull()`

```
Out[721...    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 [723... `emp.isnull().sum()`

```
Out[723...]: Name      0  
          Domain     0  
          Age        2  
          Location    2  
          Salary      0  
          Exp         1  
          dtype: int64
```

```
In [725...]: emp['Name']
```

```
Out[725...]: 0      Mike  
1      Teddy^  
2      Uma#r  
3      Jane  
4      Uttam*  
5      Kim  
Name: Name, dtype: object
```

```
In [727...]: emp['Domain']
```

```
Out[727...]: 0      Datascience#$  
1      Testing  
2      Dataanalyst^^#  
3      Ana^^lytics  
4      Statistics  
5      NLP  
Name: Domain, dtype: object
```

```
In [729...]: emp['Age']
```

```
Out[729...]: 0      34 years  
1      45' yr  
2      NaN  
3      NaN  
4      67-yr  
5      55yr  
Name: Age, dtype: object
```

```
In [731...]: emp['Location']
```

```
Out[731...]: 0      Mumbai  
             1    Bangalore  
             2      NaN  
             3   Hyderabad  
             4      NaN  
             5      Delhi  
Name: Location, dtype: object
```

```
In [733...]: emp['Salary']
```

```
Out[733...]: 0      5^00#0  
             1    10%0000  
             2   1$5%000  
             3     2000^0  
             4    30000-  
             5   6000^$0  
Name: Salary, dtype: object
```

```
In [735...]: emp['Exp']
```

```
Out[735...]: 0      2+  
             1      <3  
             2    4> yrs  
             3      NaN  
             4   5+ year  
             5     10+  
Name: Exp, dtype: object
```

```
In [737...]: emp[['Name', 'Domain']]
```

Out[737...]

	Name	Domain
0	Mike	Datascience#\$
1	Teddy^	Testing
2	Uma#r	Dataanalyst^^#
3	Jane	Ana^^lytics
4	Uttam*	Statistics
5	Kim	NLP

In [739...]

```
emp[['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp']]
```

Out[739...]

	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 [741...]

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

Out[741...]

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

It looks for any non-word character (like spaces, punctuation, or special symbols).

Removes those characters by replacing them with an empty string.

```
In [752... emp['Name'] = emp['Name'].str.replace(r'\W', '', regex=True)
```

```
In [754... emp['Name']
```

```
Out[754... 0    Mike
           1    Teddy
           2    Umar
           3    Jane
           4    Uttam
           5    Kim
Name: Name, dtype: object
```

```
In [756... emp['Domain']
```

```
Out[756... 0    Datascience
           1    Testing
           2    Dataanalyst
           3    Analytics
           4    Statistics
           5    NLP
Name: Domain, dtype: object
```

```
In [758... emp['Domain'] = emp['Domain'].str.replace(r'\W', '', regex=True)
```

```
In [760... emp['Domain']
```

```
Out[760... 0    Datascience  
           1      Testing  
           2   Dataanalyst  
           3     Analytics  
           4    Statistics  
           5       NLP  
Name: Domain, dtype: object
```

```
In [762... emp['Age'] = emp['Age'].str.replace(r'\W', ' ', regex=True)
```

```
In [764... emp['Age']
```

```
Out[764... 0    34years  
           1     45yr  
           2      NaN  
           3      NaN  
           4    67yr  
           5    55yr  
Name: Age, dtype: object
```

```
In [766... emp['Location'] = emp['Location'].str.replace(r'\W', ' ', regex=True)
```

```
In [768... emp['Location']
```

```
Out[768... 0      Mumbai  
           1   Bangalore  
           2      NaN  
           3  Hyderabad  
           4      NaN  
           5      Delhi  
Name: Location, dtype: object
```

```
In [770... emp['Salary'] = emp['Salary'].str.replace(r'\W', ' ', regex=True)
```

```
In [772... emp['Salary']
```

```
Out[772...]: 0    5000
              1   10000
              2   15000
              3   20000
              4   30000
              5   60000
Name: Salary, dtype: object
```

```
In [774...]: emp['Exp']
```

```
Out[774...]: 0      2+
              1      <3
              2     4> yrs
              3      NaN
              4    5+ year
              5     10+
Name: Exp, dtype: object
```

```
In [776...]: emp['Exp'] = emp['Exp'].str.replace(r'\W', ' ', regex=True)
```

```
In [778...]: emp['Exp']
```

```
Out[778...]: 0      2
              1      3
              2     4yrs
              3      NaN
              4    5year
              5     10
Name: Exp, dtype: object
```

```
In [780...]: emp['Exp'] = emp['Exp'].str.extract('(\d+)')
```

```
In [782...]: emp['Exp']
```

```
Out[782...]: 0      2
              1      3
              2      4
              3      NaN
              4      5
              5     10
Name: Exp, dtype: object
```

```
In [784... emp
```

```
Out[784...    Name      Domain     Age   Location  Salary  Exp
              0   Mike  Datascience  34years  Mumbai    5000    2
              1   Teddy   Testing    45yr  Bangalore  10000   3
              2   Umar  Dataanalyst  NaN      NaN    15000   4
              3   Jane  Analytics   NaN  Hyderabad  20000  NaN
              4   Uttam  Statistics  67yr      NaN    30000   5
              5   Kim    NLP        55yr    Delhi    60000   10
```

```
In [786... emp.dtypes
```

```
Out[786... Name      object
          Domain    object
          Age       object
          Location  object
          Salary    object
          Exp       object
          dtype: object
```

```
In [788... clean_data = emp.copy()
```

```
In [790... clean_data
```

Out[790...]

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

In [792...]

clean_data['Age']

Out[792...]

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

In [794...]

clean_data['Domain']

Out[794...]

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

In [796...]

clean_data['Age'] = clean_data['Age'].str.replace(r'\D', '', regex=True)

In [798...]

clean_data['Age']

```
Out[798... 0    34  
1    45  
2    NaN  
3    NaN  
4    67  
5    55  
Name: Age, dtype: object
```

```
In [800... clean_data['Age'] = clean_data['Age'].astype('float')
```

```
In [804... clean_data.dtypes
```

```
Out[804... Name      object  
Domain     object  
Age        float64  
Location   object  
Salary     object  
Exp        object  
dtype: object
```

Missing value treatment

```
In [807... clean_data['Age'] = clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['Age']))))
```

```
In [809... clean_data['Age']
```

```
Out[809... 0    34.00  
1    45.00  
2    50.25  
3    50.25  
4    67.00  
5    55.00  
Name: Age, dtype: float64
```

```
In [811... clean_data['Exp'] = clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['Exp']))))
```

```
In [813... clean_data['Exp']
```

```
Out[813...]: 0      2  
             1      3  
             2      4  
             3    4.8  
             4      5  
             5     10  
Name: Exp, dtype: object
```

```
In [815...]: clean_data
```

```
Out[815...]:   Name      Domain  Age  Location  Salary  Exp  
0   Mike  Datascience  34.00    Mumbai    5000    2  
1  Teddy       Testing  45.00  Bangalore  10000    3  
2  Umar  Dataanalyst  50.25      NaN  15000    4  
3   Jane      Analytics  50.25  Hyderbad  20000  4.8  
4  Uttam      Statistics  67.00      NaN  30000    5  
5    Kim          NLP  55.00    Delhi  60000   10
```

```
In [817...]: clean_data['Location'].isnull().sum()
```

```
Out[817...]: 2
```

```
In [819...]: clean_data['Location']
```

```
Out[819...]: 0      Mumbai  
             1    Bangalore  
             2      NaN  
             3    Hyderbad  
             4      NaN  
             5      Delhi  
Name: Location, dtype: object
```

```
In [821...]: clean_data['Location'] = clean_data['Location'].fillna(clean_data['Location'].mode()[0])
```

```
In [823...]: clean_data['Location']
```

```
Out[823...]: 0      Mumbai  
1    Bangalore  
2    Bangalore  
3    Hyderabad  
4    Bangalore  
5      Delhi  
Name: Location, dtype: object
```

Clean the data

```
In [826...]: clean_data
```

```
Out[826...]:   Name     Domain  Age  Location  Salary  Exp  
0  Mike  Datascience  34.00    Mumbai    5000    2  
1  Teddy      Testing  45.00  Bangalore  10000    3  
2  Umar  Dataanalyst  50.25  Bangalore  15000    4  
3  Jane      Analytics  50.25  Hyderabad  20000  4.8  
4  Uttam      Statistics  67.00  Bangalore  30000    5  
5  Kim        NLP  55.00      Delhi  60000   10
```

```
In [828...]: 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      float64 
 3   Location   6 non-null      object  
 4   Salary     6 non-null      object  
 5   Exp        6 non-null      object  
dtypes: float64(1), object(5)
memory usage: 420.0+ bytes
```

```
In [830...]: clean_data['Age'] = clean_data['Age'].astype(int)
```

```
In [832...]: clean_data['Age']
```

```
Out[832...]: 0    34
 1    45
 2    50
 3    50
 4    67
 5    55
Name: Age, dtype: int32
```

```
In [834...]: 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 [836...]: clean_data['Salary'] = clean_data['Salary'].astype(int)
clean_data['Exp'] = clean_data['Exp'].astype(int)
```

```
In [838...]: clean_data['Salary']
```

```
Out[838...]: 0      5000
 1     10000
 2    15000
 3   20000
 4   30000
 5   60000
Name: Salary, dtype: int32
```

```
In [840...]: clean_data['Exp']
```

```
Out[840...]: 0      2
 1      3
 2      4
 3      4
 4      5
 5     10
Name: Exp, dtype: int32
```

```
In [842...]: 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 [844...]: 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 [846... `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 [848... `clean_data`

Out[848...

	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 [850... `clean_data.to_csv('clean_data.csv')`

In [852... `import os`
`os.getcwd()`

Out[852... 'C:\\Users\\satyabrata'

In [854... `clean_data`Out[854...

	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

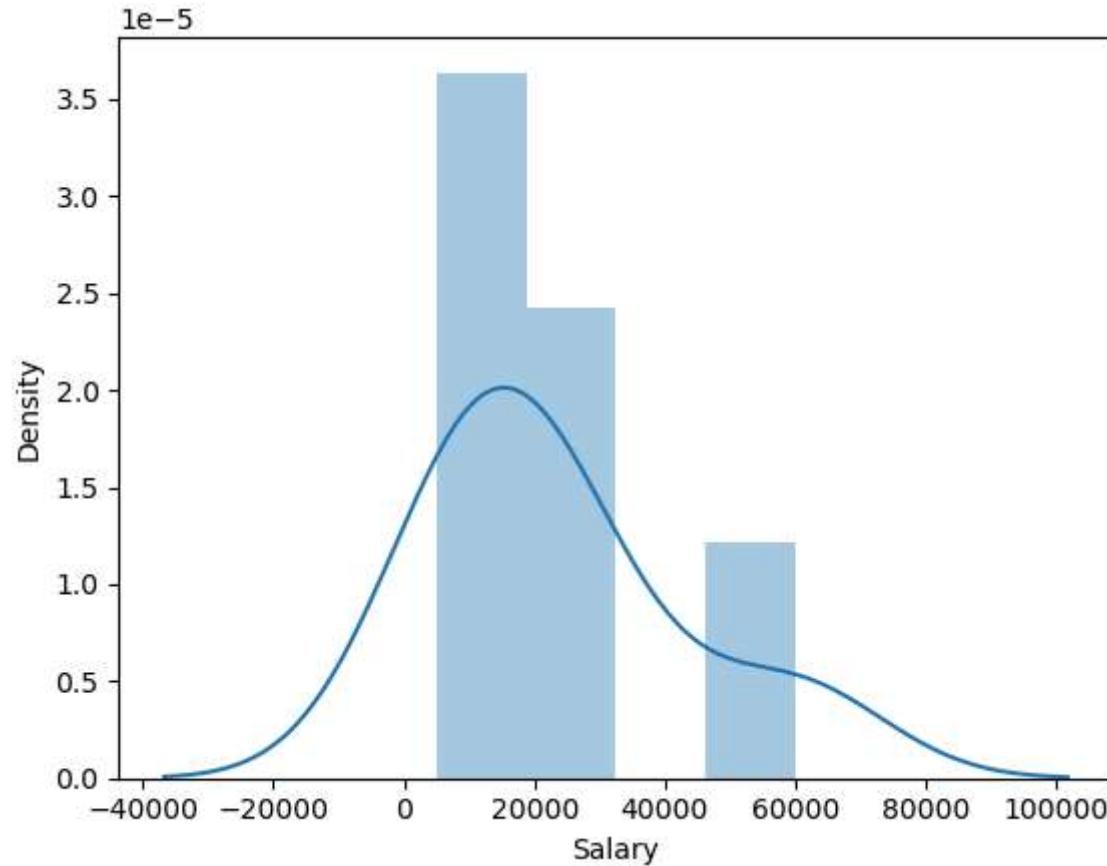
In [856... `clean_data['Salary']`Out[856...

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

Name: Salary, dtype: int32

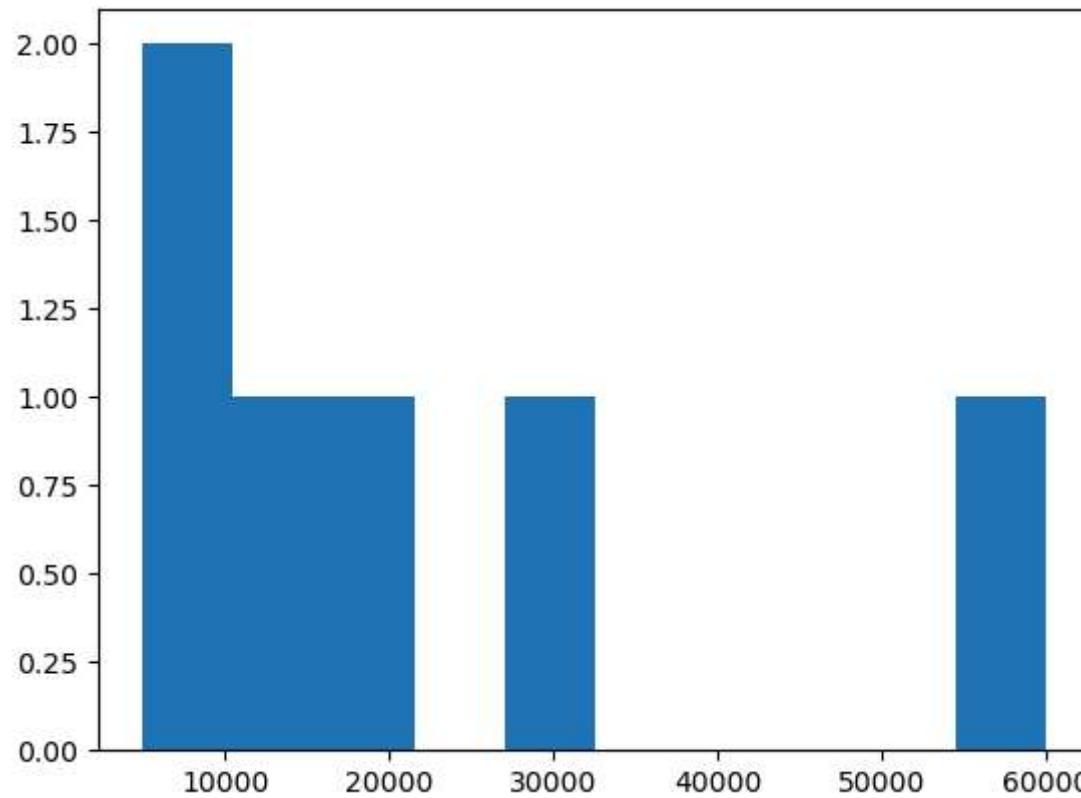
Univariate Analysis

In [859... `data = sns.distplot(clean_data['Salary'])`



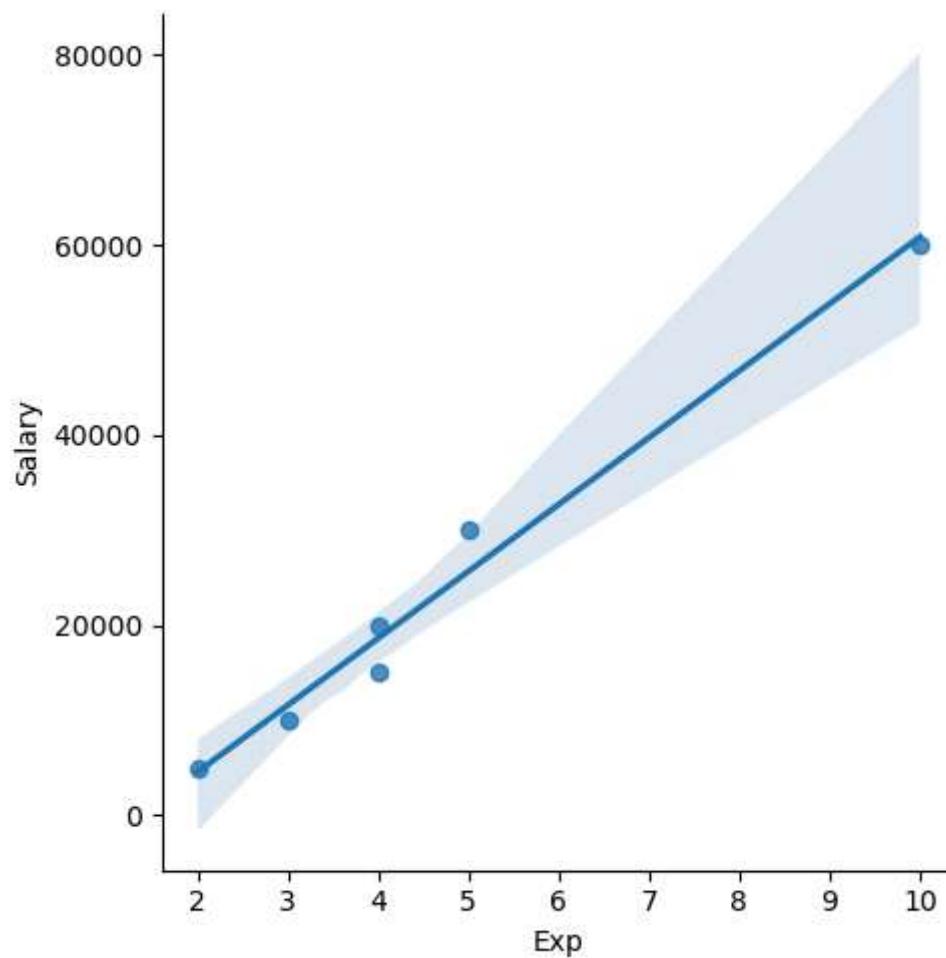
Outlier treatment

```
In [862]: data2 = plt.hist(clean_data['Salary'])
```

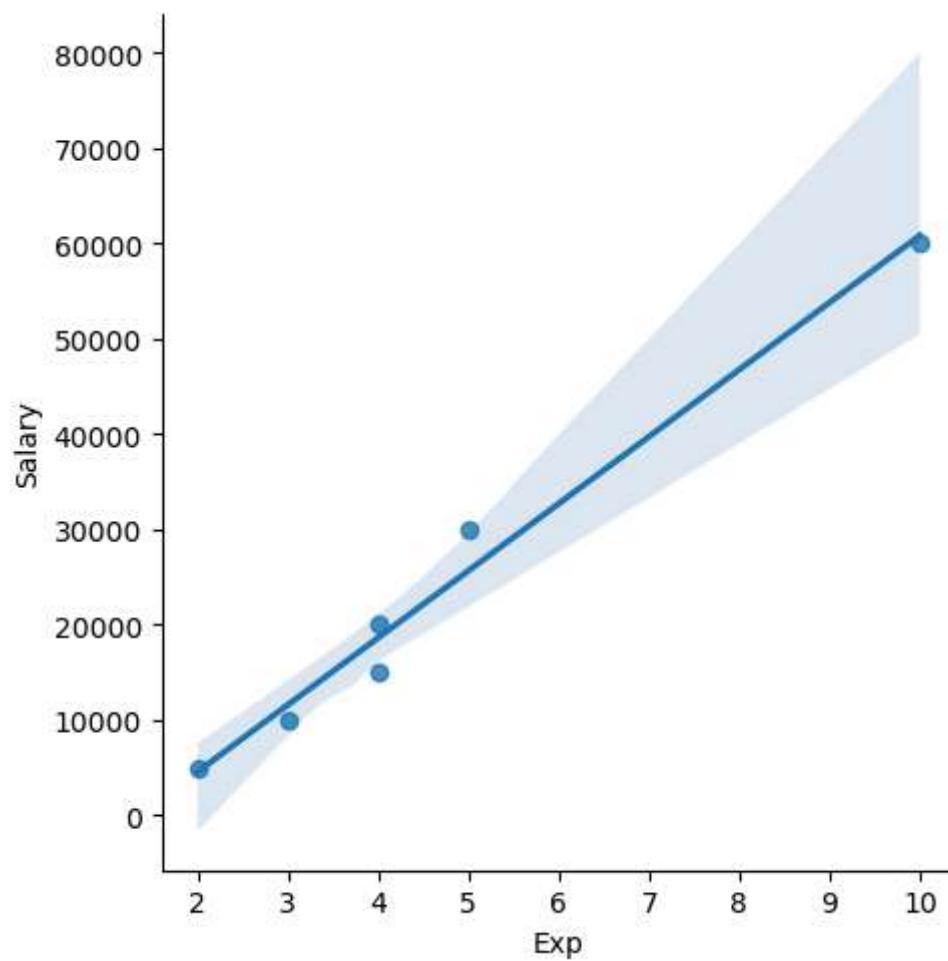


Bi-variate analysis

```
In [865...]: data3 = sns.lmplot(data=clean_data, x = 'Exp', y='Salary')
```



```
In [867]: vis4 = sns.lmplot(data=clean_data, x = 'Exp', y='Salary', fit_reg = True)
```



```
In [868]: clean_data[:]
```

Out[868...]

	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 [871...]

clean_data[0:6:2]

Out[871...]

	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 [873...]

clean_data[::-1]

Out[873...]

	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 [875...]

clean_data.columns

```
Out[875]: Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

Variable Identification

```
In [878]: X_iv = clean_data[['Name', 'Domain', 'Age', 'Location', 'Exp']]
```

```
In [880]: X_iv
```

```
Out[880]:
```

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

```
In [884]: y_dv
```

```
Out[884]:
```

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

In [886... emp

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

In [888... clean_data

	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 [890... X_iv

Out[890...]

	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 [892...]

y_dv

Out[892...]

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

In [894...]

clean_data

Out[894...]

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascienc	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

Variable transformation

In [897...]

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

In [899...]

```
imputation
```

Out[899...]

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



In [901...]

```
clean_data
```

Out[901...]

	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 [903...]

imputation

Out[903...]

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



Variable Creation

In [906...]

len(clean_data)

Out[906...]

6

In [908...]

imputation.columns

```
Out[908]: Index(['Age', 'Salary', 'Exp', 'Name_Jane', 'Name_Kim',  
                 'Name_Teddy', 'Name_Umar', 'Name_Uttam', 'Domain_Analytics',  
                 'Domain_Dataanalyst', 'Domain_Datascience', 'Domain_NLP',  
                 'Domain_Statistics', 'Domain_Testing', 'Location_Bangalore',  
                 'Location_Delhi', 'Location_Hyderabad', 'Location_Mumbai'],  
                dtype='object')
```

```
In [910]: len(imputation.columns)
```

```
Out[910]: 19
```

```
In [ ]:
```