

# ge

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
In [3]: emp=pd.read_excel(r'C:\Users\user\Desktop\20th- EDA Practice\EDA- Practice\Raw
emp
```

Out[3]:

	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 [4]: emp.shape
```

Out[4]: (6, 6)

```
In [5]: len(emp)
```

Out[5]: 6

```
In [6]: emp.columns
```

Out[6]: Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')

```
In [7]: 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 [8]: emp
```

Out[8]:

	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 [9]: emp['Name']

```
Out[9]: 0      Mike
        1      Teddy^
        2      Uma#r
        3      Jane
        4      Uttam*
        5      Kim
Name: Name, dtype: object
```

In [10]: emp['Domain']

```
Out[10]: 0      Datascienc#$
        1      Testing
        2      Dataanalyst^#
        3      Ana^^lytics
        4      Statistics
        5      NLP
Name: Domain, dtype: object
```

In [11]: emp['Age']

```
Out[11]: 0      34 years
        1      45' yr
        2      NaN
        3      NaN
        4      67-yr
        5      55yr
Name: Age, dtype: object
```

In [12]: emp['Location']

```
Out[12]: 0      Mumbai
        1      Bangalore
        2      NaN
        3      Hyderbad
        4      NaN
        5      Delhi
Name: Location, dtype: object
```

In [13]: emp['Salary']

```
Out[13]: 0      5^00#0
          1      10%000
          2      1$5%000
          3      2000^0
          4      30000-
          5      6000^$0
Name: Salary, dtype: object
```

```
In [14]: emp['Exp']
```

```
Out[14]: 0      2+
          1      <3
          2      4> yrs
          3      NaN
          4      5+ year
          5      10+
Name: Exp, dtype: object
```

```
In [15]: emp[['Name', 'Domain']]
```

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

```
In [16]: emp[['Name', 'Domain', 'Age']]
```

	Name	Domain	Age
0	Mike	Datascience#\$	34 years
1	Teddy^	Testing	45' yr
2	Uma#r	Dataanalyst^^#	NaN
3	Jane	Ana^^lytics	NaN
4	Uttam*	Statistics	67-yr
5	Kim	NLP	55yr

```
In [17]: emp[['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp']]
```

Out[17]:

	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 [18]:

emp.isnull()

Out[18]:

	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 [19]:

emp['Name']

Out[19]:

```

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

```

In [20]:

pip install pandas

```

Requirement already satisfied: pandas in c:\users\user\anaconda3\lib\site-packages (2.2.2)
Requirement already satisfied: numpy>=1.26.0 in c:\users\user\anaconda3\lib\site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\user\anaconda3\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\user\anaconda3\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\user\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\user\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

```

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

```
Out[31]: 0    Datascience
          1    Testing
          2    Dataanalyst
          3    Analytics
          4    Statistics
          5    NLP
Name: Domain, dtype: object
```

```
In [32]: emp['Age'] = emp['Age'].str.replace(r'\W', ' ', regex=True)
emp['Age']
```

```
Out[32]: 0    34years
          1    45yr
          2    NaN
          3    NaN
          4    67yr
          5    55yr
Name: Age, dtype: object
```

```
In [33]: emp['Age'] = emp['Age'].str.extract('(\d+)')
emp['Age']
```

```
<>:1: SyntaxWarning: invalid escape sequence '\d'
<>:1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\user\AppData\Local\Temp\ipykernel_7932\1503651561.py:1: SyntaxWarning: i
nvalid escape sequence '\d'
    emp['Age'] = emp['Age'].str.extract('(\d+)')
```

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

```
In [34]: emp
```

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

```
Out[36]: 0      Mumbai
1      Bangalore
2      NaN
3      Hyderbad
4      NaN
5      Delhi
Name: Location, dtype: object
```

```
In [38]: emp['Salary']=emp['Salary'].str.replace(r'\W', '', regex=True)
emp['Salary']
```

```
Out[38]: 0      5000
1      10000
2      15000
3      20000
4      30000
5      60000
Name: Salary, dtype: object
```

```
In [39]: emp
```

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

```
In [42]: emp['Exp'] = emp['Exp'].str.extract('(\d+)')
emp['Exp']
```

```
<>>1: SyntaxWarning: invalid escape sequence '\d'
<>>1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\user\AppData\Local\Temp\ipykernel_7932\4181881072.py:1: SyntaxWarning: i
nvalid escape sequence '\d'
    emp['Exp'] = emp['Exp'].str.extract('(\d+)')
```

```
Out[42]: 0      2
         1      3
         2      4
         3    NaN
         4      5
         5     10
Name: Exp, dtype: object
```

```
In [43]: emp
```

```
Out[43]:   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 [44]: clean_data=emp.copy()
```

```
In [45]: clean_data
```

```
Out[45]:   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
```

## MISSING VALUE TREATMENT

```
In [46]: clean_data
```

```
Out[46]:   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 [47]: `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
```

In [48]: `import numpy as np`

In [49]: `clean_data`

Out[49]:

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

In [51]: `clean_data.head(1)`

Out[51]:

	Name	Domain	Age	Location	Salary	Exp
<b>0</b>	Mike	Datascience	34	Mumbai	5000	2

In [52]: `clean_data['Age']`

Out[52]:

0	34
1	45
2	NaN
3	NaN
4	67
5	55

Name: Age, dtype: object

In [53]: `clean_data['Age']=clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['Age'])))`

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

```
Out[54]: 0      34
         1      45
         2    50.25
         3    50.25
         4      67
         5      55
Name: Age, dtype: object
```

```
In [56]: clean_data
```

	Name	Domain	Age	Location	Salary	Exp
<b>0</b>	Mike	Datascience	34	Mumbai	5000	2
<b>1</b>	Teddy	Testing	45	Bangalore	10000	3
<b>2</b>	Umar	Dataanalyst	50.25	NaN	15000	4
<b>3</b>	Jane	Analytics	50.25	Hyderabad	20000	NaN
<b>4</b>	Uttam	Statistics	67	NaN	30000	5
<b>5</b>	Kim	NLP	55	Delhi	60000	10

```
In [57]: clean_data['Exp'] = clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['Exp'])))
```

```
Out[57]: 0      2
         1      3
         2      4
         3    4.8
         4      5
         5     10
Name: Exp, dtype: object
```

```
In [59]: clean_data
```

	Name	Domain	Age	Location	Salary	Exp
<b>0</b>	Mike	Datascience	34	Mumbai	5000	2
<b>1</b>	Teddy	Testing	45	Bangalore	10000	3
<b>2</b>	Umar	Dataanalyst	50.25	NaN	15000	4
<b>3</b>	Jane	Analytics	50.25	Hyderabad	20000	4.8
<b>4</b>	Uttam	Statistics	67	NaN	30000	5
<b>5</b>	Kim	NLP	55	Delhi	60000	10

```
In [60]: clean_data['Location'] = clean_data['Location'].fillna(clean_data['Location'].mode().values)
```

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

```
In [61]: clean_data['Age'] = clean_data['Age'].astype(int)
clean_data['Salary'] = clean_data['Salary'].astype(int)
clean_data['Exp'] = clean_data['Exp'].astype(int)
clean_data
```

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

```
In [62]: 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 [63]: clean_data['Name'] = clean_data['Name'].astype('category')
clean_data['Domain'] = clean_data['Domain'].astype('category')
clean_data['Location'] = clean_data['Location'].astype('category')
clean_data
```

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datasience	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 [64]: `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 [65]: `clean_data.to_csv('clean_data.csv')`

In [66]: `import os  
os.getcwd()`

Out[66]: 'C:\\\\Users\\\\user'

In [67]: `clean_data.columns`

Out[67]: Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')

## VISUALIZATIONS

In [68]: `import matplotlib.pyplot as plt  
import seaborn as sns`

In [69]: `import warnings  
warnings.filterwarnings('ignore')`

In [70]: `clean_data`

Out[70]:

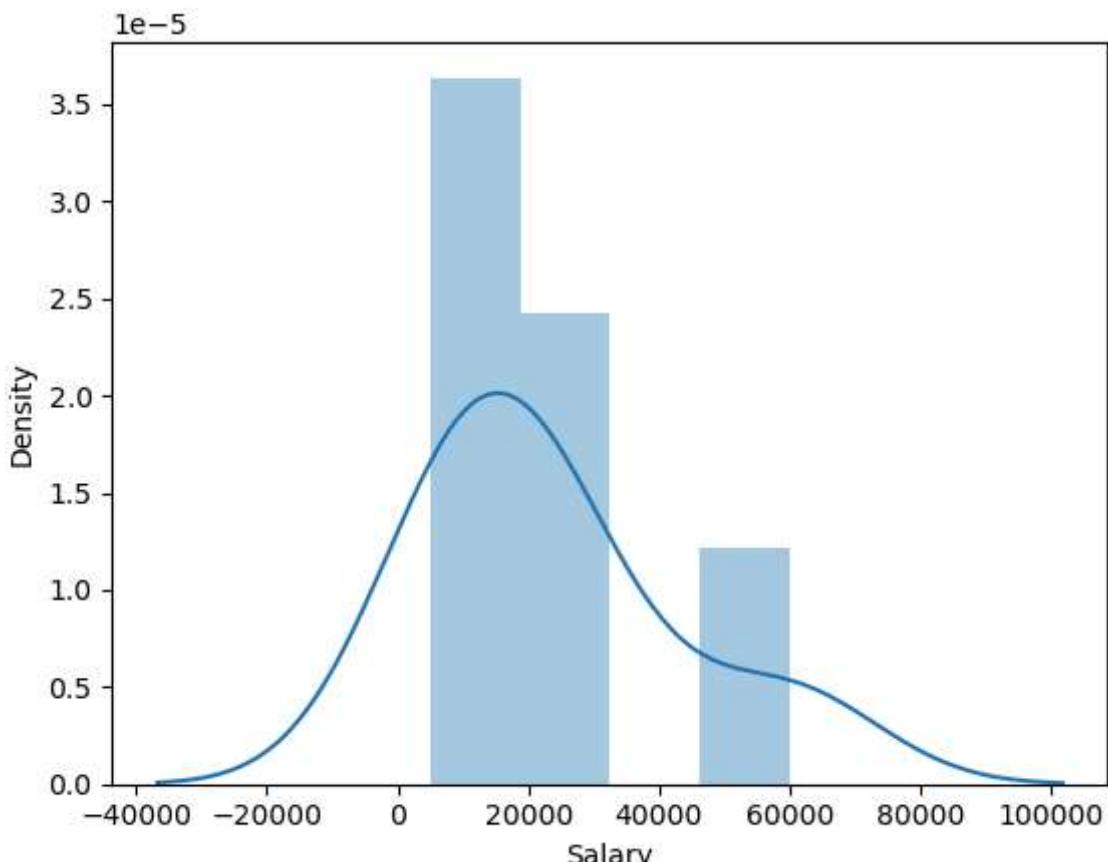
	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2
1	Teddy	Testing	45	Bangalore	10000	3
2	Umar	Dataanalyst	50	Bangalore	15000	4
3	Jane	Analytics	50	Hyderabad	20000	4
4	Uttam	Statistics	67	Bangalore	30000	5
5	Kim	NLP	55	Delhi	60000	10

In [71]: `clean_data['Salary']`

Out[71]:

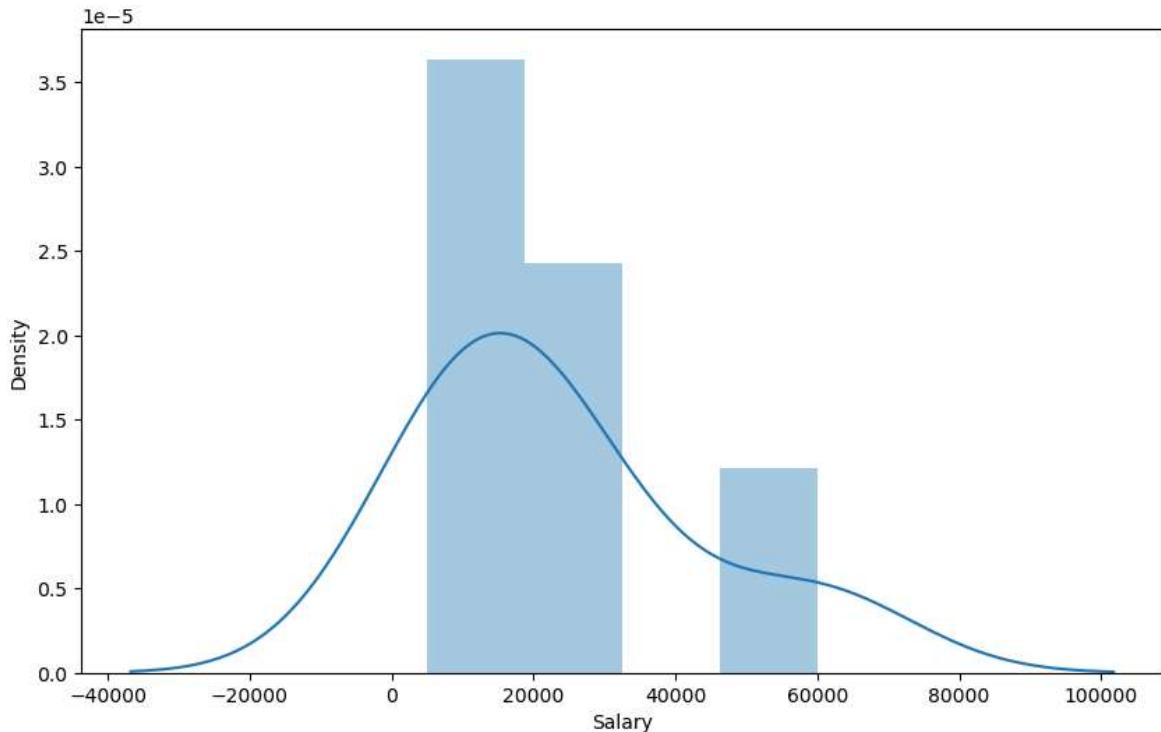
```
0      5000
1     10000
2     15000
3     20000
4     30000
5     60000
Name: Salary, dtype: int32
```

In [72]: `vis1 = sns.distplot(clean_data['Salary'])`

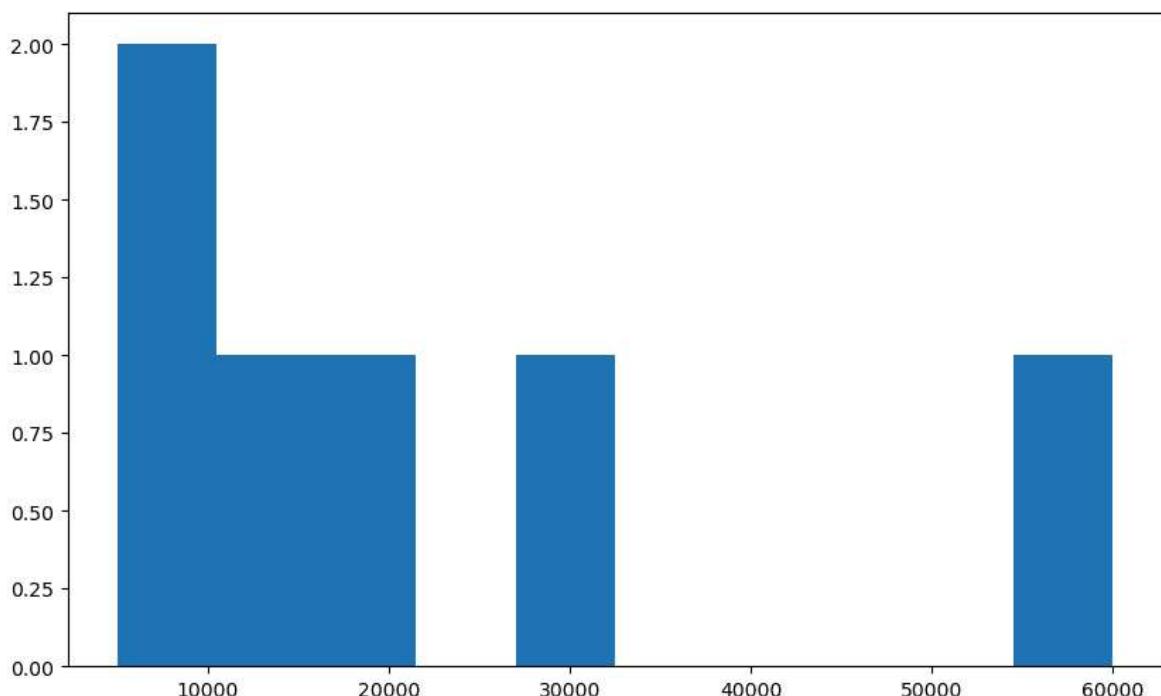


In [73]: `plt.rcParams['figure.figsize'] = 10,6`

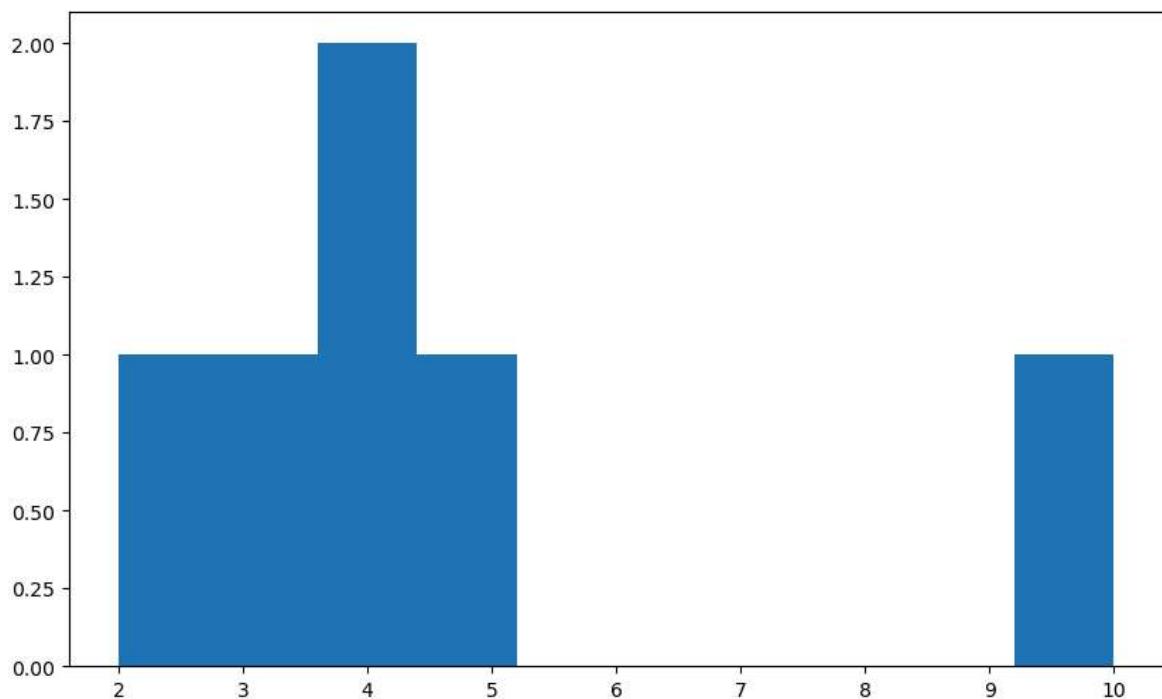
In [74]: `vis1 = sns.distplot(clean_data['Salary'])`



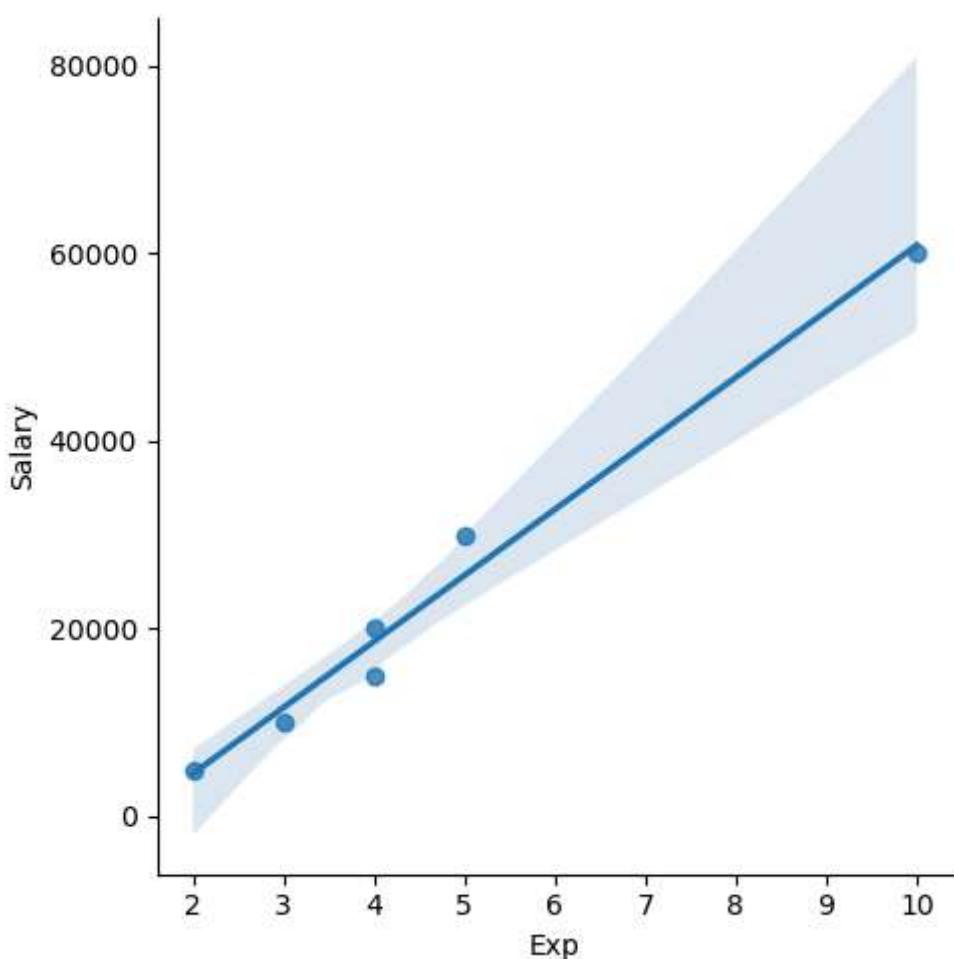
```
In [75]: vis2 = plt.hist(clean_data['Salary'])
```



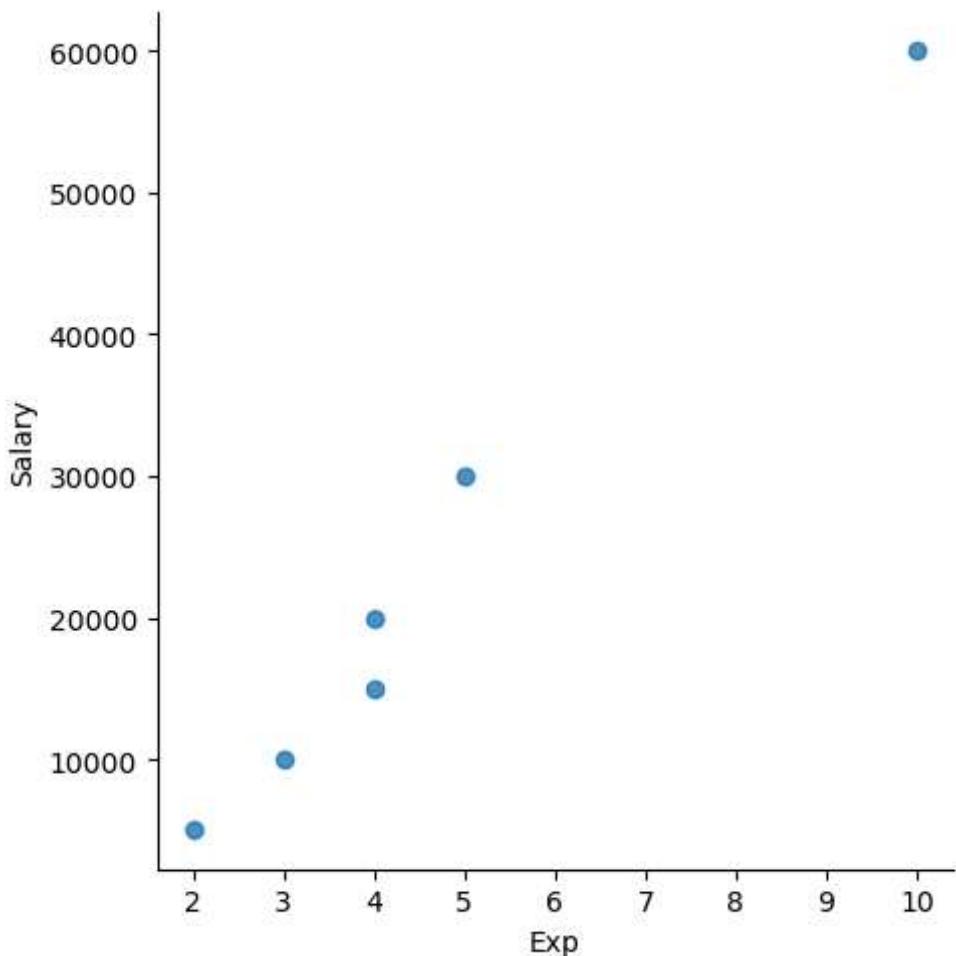
```
In [76]: vis3 = plt.hist(clean_data['Exp'])
```



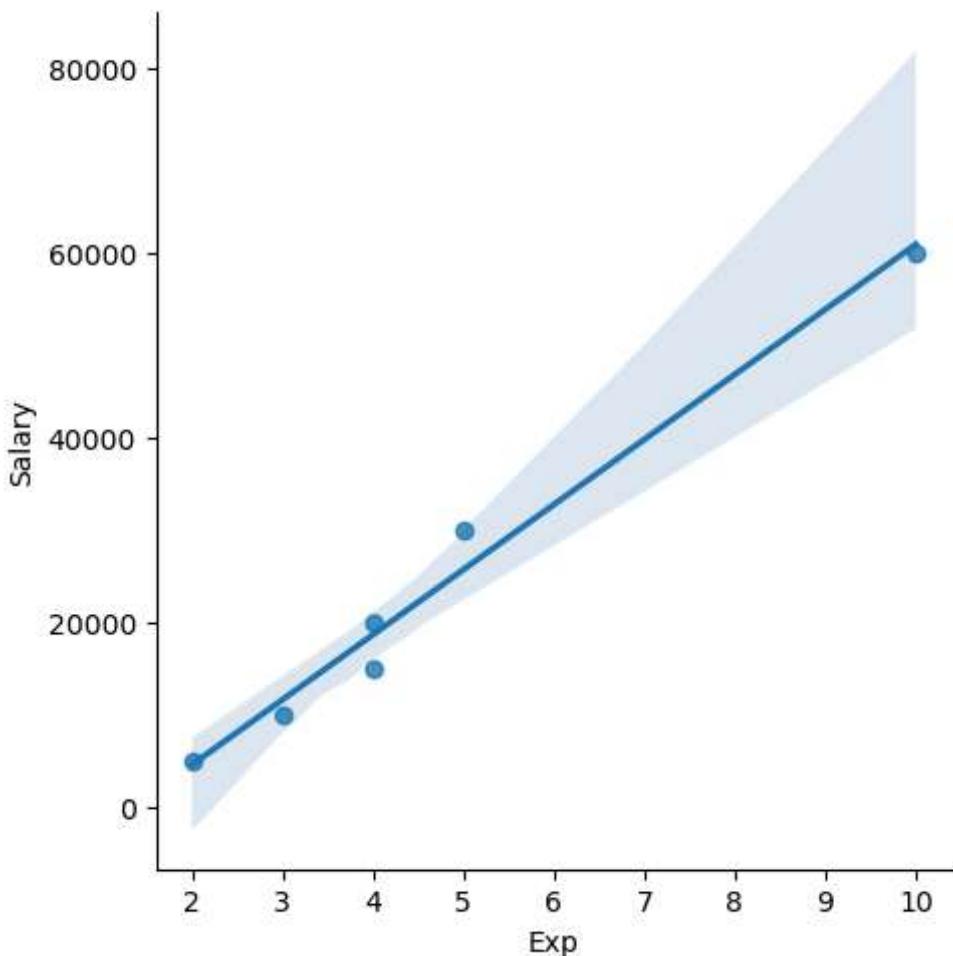
```
In [77]: vis4 = sns.lmplot(data=clean_data,x = 'Exp', y='Salary')
```



```
In [78]: vis5 = sns.lmplot(data=clean_data,x = 'Exp', y='Salary', fit_reg = False)
```



```
In [79]: vis6 = sns.lmplot(data=clean_data,x = 'Exp', y='Salary', fit_reg = True)
```



```
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	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 [81]: clean_data[:]
```

Out[81]:

	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 [82]: `clean_data[:2]`

Out[82]:

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

In [83]: `clean_data[2:]`

Out[83]:

	Name	Domain	Age	Location	Salary	Exp
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 [84]: `clean_data[0:1]`

Out[84]:

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2

In [86]: `clean_data[0,3]`

```
KeyError Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3805, in Index.get_
_loc(self, key)
    3804     try:
-> 3805         return self._engine.get_loc(casted_key)
    3806     except KeyError as err:

File index.pyx:167, in pandas._libs.index.IndexEngine.get_loc()

File index.pyx:196, in pandas._libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:7081, in pandas._libs.hashtable.Py
ObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:7089, in pandas._libs.hashtable.Py
ObjectHashTable.get_item()

KeyError: (0, 3)
```

The above exception was the direct cause of the following exception:

```
KeyError Traceback (most recent call last)
Cell In[86], line 1
----> 1 clean_data[0,3]

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:4102, in DataFrame.__getitem__(self, key)
    4100 if self.columns.nlevels > 1:
    4101     return self._getitem_multilevel(key)
-> 4102 indexer = self.columns.get_loc(key)
    4103 if is_integer(indexer):
    4104     indexer = [indexer]

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3812, in Index.get_loc(self, key)
    3807     if isinstance(casted_key, slice) or (
    3808         isinstance(casted_key, abc.Iterable)
    3809         and any(isinstance(x, slice) for x in casted_key)
    3810     ):
    3811         raise InvalidIndexError(key)
-> 3812     raise KeyError(key) from err
    3813 except TypeError:
    3814     # If we have a listlike key, _check_indexing_error will raise
    3815     # InvalidIndexError. Otherwise we fall through and re-raise
    3816     # the TypeError.
    3817     self._check_indexing_error(key)

KeyError: (0, 3)
```

```
In [87]: x_iv = clean_data.drop(['Salary'],axis=1)
```

In [88]: clean\_data

Out[88]:

	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 [89]: `x_iv`

Out[89]:

	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 [90]: `x_iv.columns`

Out[90]: `Index(['Name', 'Domain', 'Age', 'Location', 'Exp'], dtype='object')`

In [91]: `clean_data.columns`

Out[91]: `Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')`

In [92]: `clean_data`

Out[92]:

	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 [93]: `y_dv = clean_data.drop(['Name', 'Domain', 'Age', 'Location', 'Exp'], axis=1)`  
`clean_data`

Out[93]:

	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 [94]: clean\_data

Out[94]:

	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 [95]: x\_iv

Out[95]:

	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 [97]: y\_dv

Out[97]:

	Salary
<b>0</b>	5000
<b>1</b>	10000
<b>2</b>	15000
<b>3</b>	20000
<b>4</b>	30000
<b>5</b>	60000

In [ ]: