In [19]:

```
import pandas as pd

# Correct method to read an Excel file
emp = pd.read_excel('/content/drive/MyDrive/Rawdata.xlsx')

print(emp)
```

```
     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 [20]:

```
emp.columns
```

Out[20]:

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

In [21]:

```
emp.shape
```

Out[21]:

```
(6, 6)
```

In [22]:

```
emp.head()
```

Out[22]:

	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 [23]:

```
emp.tail()
```

Out[23]:

	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 [24]:

```
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: 416.0+ bytes
```

In [25]:
```
emp
```

Out[25]:

	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 [26]:
```
emp['Domain']
```

Out[26]:

	Domain
0	Datascience#$
1	Testing
2	Dataanalyst^^#
3	Ana^^lytics
4	Statistics
5	NLP

dtype: object

In [11]:
```
emp.isnull()
```

Out[11]:

	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 [27]:
```
emp.isnull().sum()
```

Out[27]:

	0
Name	0
Domain	0
Age	2
Location	2
Salary	0
Exp	1

dtype: int64

In [28]:

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

Out[28]:

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

dtype: object

In [32]:

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

In [33]:

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

Out[33]:

	Name
0	Mike
1	Teddy
2	Umar
3	Jane
4	Uttam
5	Kim

dtype: object

In [34]:

```
emp['Domain']
```

Out[34]:

	Domain
0	Datascience#$
1	Testing
2	Dataanalyst^^#
3	Ana^^lytics
4	Statistics
5	NLP

dtype: object

In [35]:

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

In [36]:

```
emp['Domain']
```

Out[36]:

	Domain

0	Datascience
1	Testing
2	Dataanalyst
3	Analytics
4	Statistics
5	NLP

dtype: object

In [37]:

```
emp
```

Out[37]:

	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	Umar	Dataanalyst	NaN	NaN	1$5%000	4> yrs
3	Jane	Analytics	NaN	Hyderbad	2000^0	NaN
4	Uttam	Statistics	67-yr	NaN	30000-	5+ year
5	Kim	NLP	55yr	Delhi	6000^$0	10+

In [38]:

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

In [40]:

```
emp['Age']
```

Out[40]:

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

dtype: object

In [41]:

```
emp
```

Out[41]:

	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 [42]:

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

In [43]:
```
emp['Location']
```

Out[43]:

	Location
0	Mumbai
1	Bangalore
2	NaN
3	Hyderbad
4	NaN
5	Delhi

dtype: object

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

Out[44]:

	Salary
0	5^00#0
1	10%%000
2	1$5%000
3	2000^0
4	30000-
5	6000^$0

dtype: object

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

In [47]:
```
emp['Salary']
```

Out[47]:

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

dtype: object

In [48]:
```
emp
```

Out[48]:

	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 [49]:

```
emp['Exp']=emp['Exp'].str.replace(r'\D','',regex=True)
```

In [50]:

```
emp['Exp']
```

Out[50]:

	Exp
0	2
1	3
2	4
3	NaN
4	5
5	10

dtype: object

In [51]:

```
emp
```

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

In [52]:

```
clean_data=emp.copy()
```

In [53]:

```
clean_data
```

Out[53]:

	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 [54]:

```
clean_data['Age']
```

Out[54]:

	Age
0	34

1	45
2	NaN
3	NaN
4	67
5	55

dtype: object

In [55]:

```
import numpy as np
```

In [56]:

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

In [57]:

```
clean_data['Age']
```

Out[57]:

	Age
0	34
1	45
2	50.25
3	50.25
4	67
5	55

dtype: object

In [58]:

```
clean_data['Exp']
```

Out[58]:

	Exp
0	2
1	3
2	4
3	NaN
4	5
5	10

dtype: object

In [59]:

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

In [60]:

```
clean_data['Exp']
```

Out[60]:

	Exp
0	2
1	3
2	4
3	4.8

4	5
5	10

dtype: object

In [61]:
```
clean_data
```

Out[61]:

	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 [62]:
```
clean_data['Location'].isnull().sum()
```

Out[62]:
```
2
```

In [63]:
```
clean_data['Location']
```

Out[63]:

	Location
0	Mumbai
1	Bangalore
2	NaN
3	Hyderbad
4	NaN
5	Delhi

dtype: object

In [64]:
```
clean_data['Location']=clean_data['Location'].fillna(clean_data['Location'].mode()[0])
```

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

Out[65]:

	Location
0	Mumbai
1	Bangalore
2	Bangalore
3	Hyderbad
4	Bangalore
5	Delhi

dtype: object

In [66]:

```
clean_data
```

Out[66]:

	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 [67]:

```
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: 416.0+ bytes
```

In [68]:

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

In [69]:

```
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      object
 5   Exp       6 non-null      object
dtypes: int64(1), object(5)
memory usage: 416.0+ bytes
```

In [70]:

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

In [71]:

```
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: 416.0+ bytes
```

In [72]:

```
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 [73]:

```
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: 934.0 bytes
```

In [74]:

```
clean_data
```

Out[74]:

	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 [75]:

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

In [76]:

```
import os
os.getcwd()
```

Out[76]:

```
'/content'
```

In [77]:

```
clean_data
```

Out[77]:

	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 APPLY **bold text**

In [78]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [79]:
```
import warnings
warnings.filterwarnings('ignore')
```

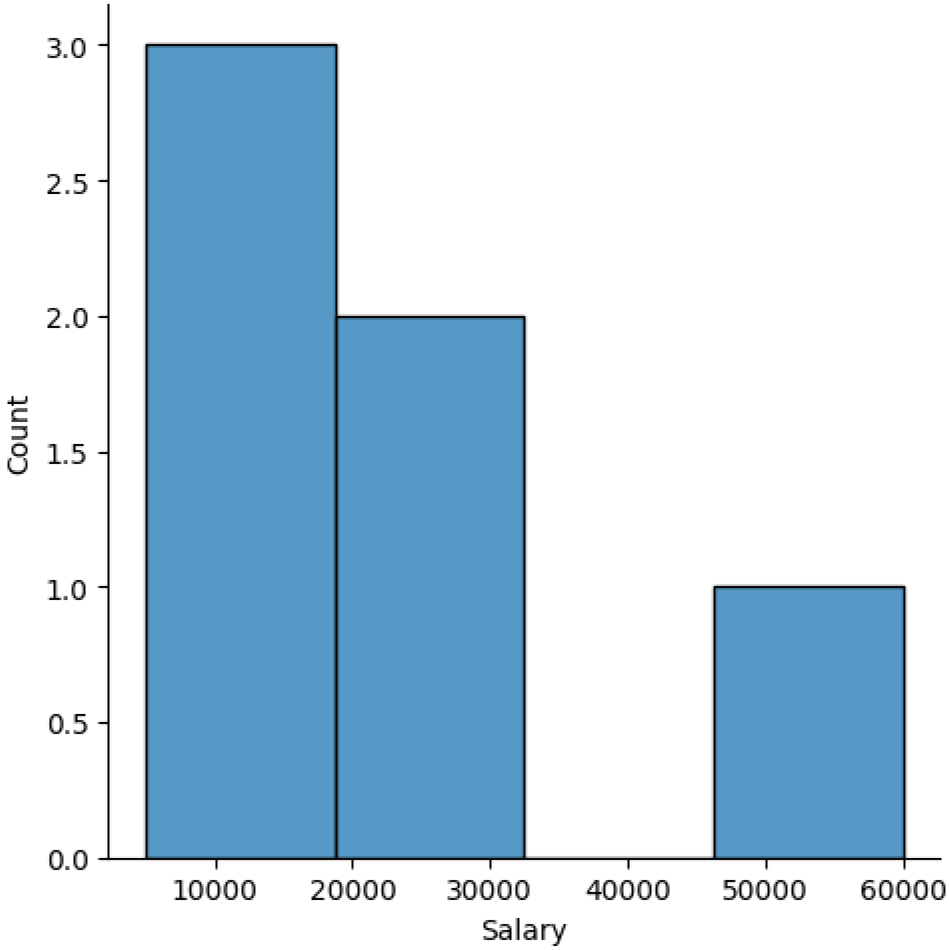
In [80]:
```
clean_data['Salary']
```

Out[80]:

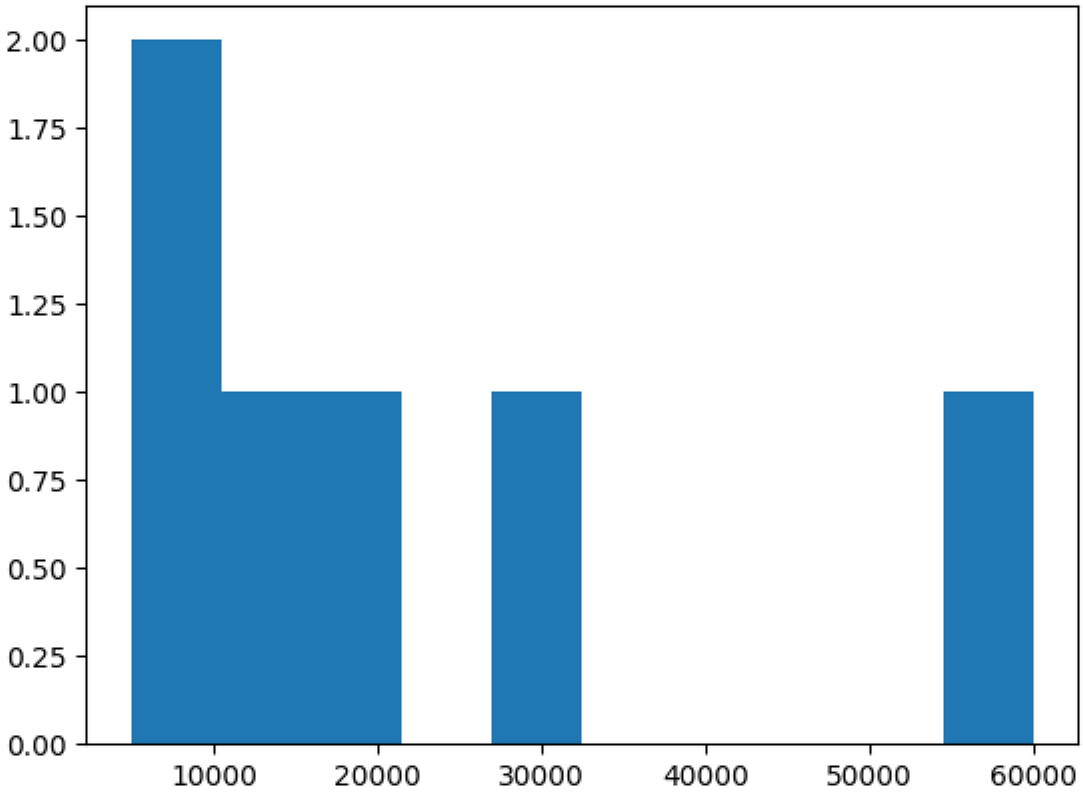
	Salary
0	5000
1	10000
2	15000
3	20000
4	30000
5	60000

dtype: int64

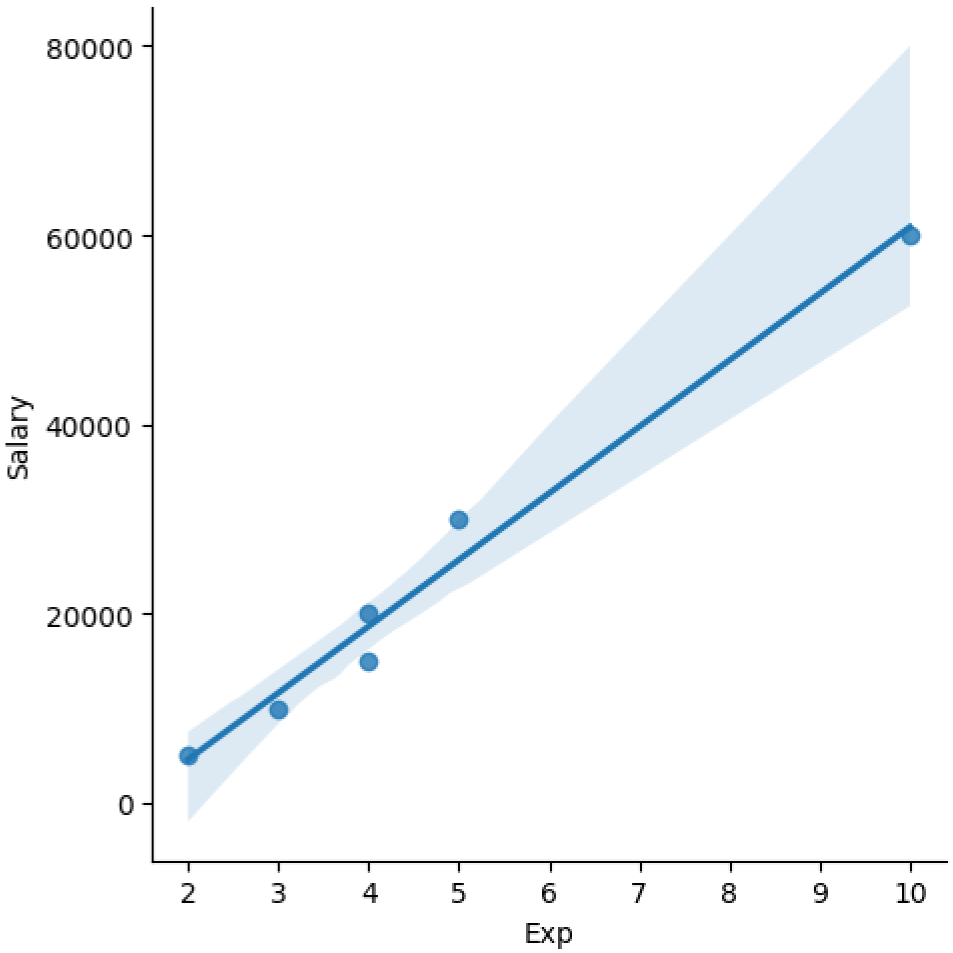
In [83]:
```
vist1=sns.displot(clean_data['Salary'])
```



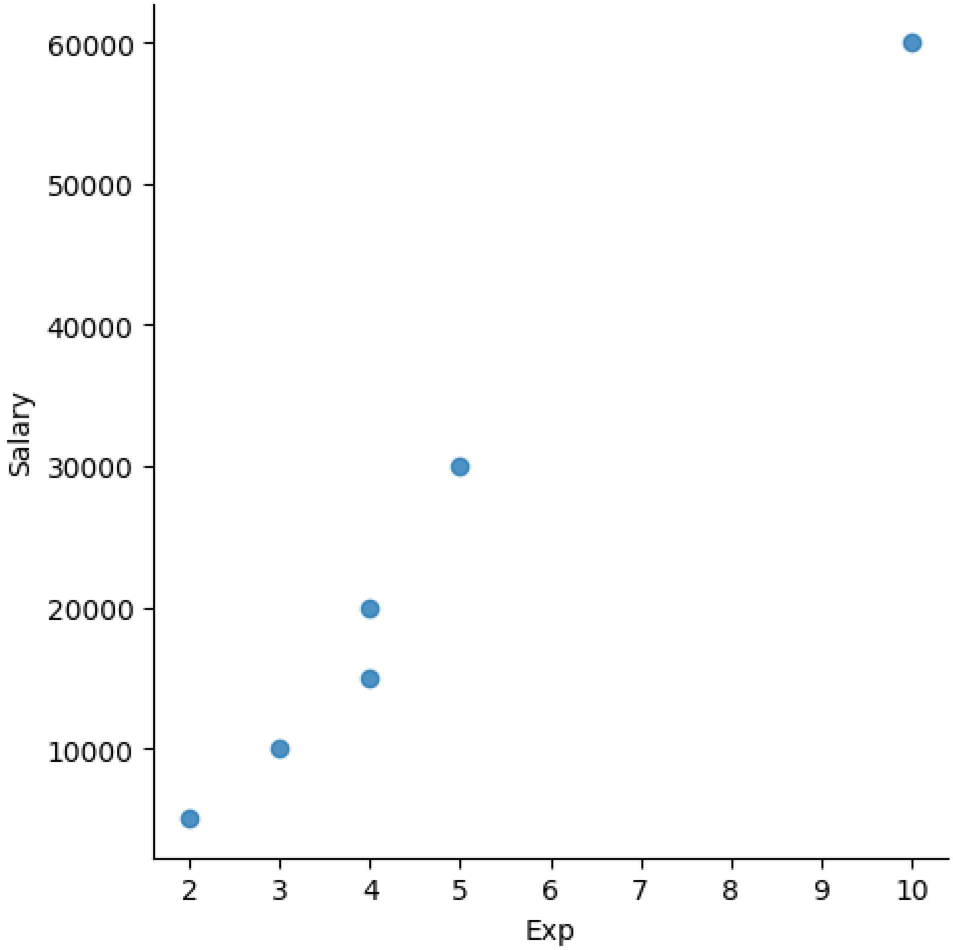
In [84]:
```
vist2=plt.hist(clean_data['Salary'])
```



```
In [85]: vis3=sns.lmplot(data=clean_data,x='Exp',y='Salary')
```



```
In [87]: vis4=sns.lmplot(data=clean_data,x='Exp',y='Salary',fit_reg=False)
```



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

In [89]:
```
clean_data[0:6:2]
```

Out[89]:

	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 [91]:
```
clean_data[::-1]
```

Out[91]:

	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 [92]:

```
clean_data.columns
```

Out[92]:

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

In [93]:

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

Out[93]:

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

In [94]:

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

Out[94]:

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

dtype: int64

In [95]:

```
emp
```

Out[95]:

	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 [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	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 [97]:

```
X_iv
```

Out[97]:

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

In [98]:

```
y_dv
```

Out[98]:

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

dtype: int64

In [99]:

```
clean_data
```

Out[99]:

	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 [108…

```
imputation=pd.get_dummies(clean_data)
imputation
```

Out[108]:

	Age	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Umar	Name_Uttam	Domain_/
0	34	5000	2	False	False	True	False	False	False	

1	45	10000	3	False	False	False	True	False	False
2	50	15000	4	False	False	False	False	True	False
3	50	20000	4	True	False	False	False	False	False
4	67	30000	5	False	False	False	False	False	True
5	55	60000	10	False	True	False	False	False	False

In [101…

```
clean_data
```

Out[101]:

	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 [102…

```
imputation
```

Out[102]:

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

In [103…

```
clean_data=clean_data.replace({True:1,False:0})
clean_data
```

Out[103]:

	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 [104…

```
imputation
```

Out[104]:

	Age	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Umar	Name_Uttam	Domain_
0	34	5000	2	False	False	True	False	False	False	
1	45	10000	3	False	False	False	True	False	False	
2	50	15000	4	False	False	False	False	True	False	

3	50	20000	4	True	False	False	False	False	False
4	67	30000	5	False	False	False	False	False	True
5	55	60000	10	False	True	False	False	False	False

In [105…

```
clean_data=clean_data.astype(int)
clean_data
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/arrays/categorical.py in astype(sel
f, dtype, copy)
    563             try:
--> 564                 new_cats = new_cats.astype(dtype=dtype, copy=copy)
    565                 fill_value = self.categories._na_value

ValueError: invalid literal for int() with base 10: 'Analytics'

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
<ipython-input-105-6d36147c8c60> in <cell line: 1>()
----> 1 clean_data=clean_data.astype(int)
      2 clean_data

/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py in astype(self, dtype, co
py, errors)
   6532         else:
   6533             # else, only a single dtype is given
-> 6534             new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)
   6535             res = self._constructor_from_mgr(new_data, axes=new_data.axes)
   6536             return res.__finalize__(self, method="astype")

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in astype(sel
f, dtype, copy, errors)
    412             copy = False
    413 
--> 414         return self.apply(
    415             "astype",
    416             dtype=dtype,

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in apply(self,
 f, align_keys, **kwargs)
    352                 applied = b.apply(f, **kwargs)
    353             else:
--> 354                 applied = getattr(b, f)(**kwargs)
    355             result_blocks = extend_blocks(applied, result_blocks)
    356 

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/blocks.py in astype(self,
 dtype, copy, errors, using_cow)
    614         values = self.values
    615 
--> 616         new_values = astype_array_safe(values, dtype, copy=copy, errors=errors)
    617 
    618         new_values = maybe_coerce_values(new_values)

/usr/local/lib/python3.10/dist-packages/pandas/core/dtypes/astype.py in astype_array_saf
e(values, dtype, copy, errors)
    236 
    237     try:
--> 238         new_values = astype_array(values, dtype, copy=copy)
    239     except (ValueError, TypeError):
    240         # e.g. _astype_nansafe can fail on object-dtype of strings
```

```
/usr/local/lib/python3.10/dist-packages/pandas/core/dtypes/astype.py in astype_array(values, dtype, copy)
    178     if not isinstance(values, np.ndarray):
    179         # i.e. ExtensionArray
--> 180         values = values.astype(dtype, copy=copy)
    181 
    182     else:

/usr/local/lib/python3.10/dist-packages/pandas/core/arrays/categorical.py in astype(self, dtype, copy)
    573             ):
    574                 msg = f"Cannot cast {self.categories.dtype} dtype to {dtype}"
--> 575                 raise ValueError(msg)
    576 
    577             result = take_nd(

ValueError: Cannot cast object dtype to int64
```

```
In [ ]:
```