In [42]:
```
import pandas as pd
```

In [43]:
```
emp = pd.read_excel(r'C:\Users\lenovo\Desktop\raw data\Rawdata.xlsx')
```

In [44]:
```
emp
```

Out[44]:

	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 [45]:
```
emp['Name']
```

Out[45]:
```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

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

In [47]:
```
emp
```

Out[47]:

	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	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 [48]:
```
emp['Name'] = emp['Name'].str.replace(r'\W', '')
```

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

Out[49]:

	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	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 [50]:
```
emp['Domain'] = emp['Domain'].str.replace(r'\W', '',regex = True)
```

In [51]:
```
emp
```

Out[51]:

	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 [52]:
```
emp['Age'] = emp['Age'].str.replace(r'\W', '',regex = True)
```

In [53]:
```
emp['Age']
```

Out[53]:
```
0    34years
1       45yr
2        NaN
3        NaN
4       67yr
5       55yr
Name: Age, dtype: object
```

In [54]:
```
emp['Age'] =emp['Age'].str.extract('(\d+)')
```

```
<>:1: SyntaxWarning: invalid escape sequence '\d'
<>:1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\lenovo\AppData\Local\Temp\ipykernel_14340\1797230661.py:1: SyntaxWarnin
g: invalid escape sequence '\d'
  emp['Age'] =emp['Age'].str.extract('(\d+)')
```

In [55]:
```
emp['Age']
```

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

In [56]:
```
emp['Location'] = emp['Location'].str.replace(r'\W', '',regex = True)
```

In [57]:
```
emp['Location']
```

Out[57]:
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [58]:
```
emp['Salary'] = emp['Salary'].str.replace(r'\W', '',regex = True)
```

In [59]:
```
emp['Salary']
```

Out[59]:
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

In [60]:
```
emp['Exp'] =emp['Exp'].str.extract('(\d+)')
```

```
<>:1: SyntaxWarning: invalid escape sequence '\d'
<>:1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\lenovo\AppData\Local\Temp\ipykernel_14340\3232909286.py:1: SyntaxWarnin
g: invalid escape sequence '\d'
  emp['Exp'] =emp['Exp'].str.extract('(\d+)')
```

In [61]:
```
emp['Exp']
```

Out[61]:
```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

In [62]:
```
emp
```

Out[62]:

	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 [63]:
```
clean_data = emp.copy()
```

In [64]:
```
clean_data
```

Out[64]:

	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 [65]:
```
clean_data.isnull().sum()
```

Out[65]:
```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

In [66]:
```
clean_data['Age']
```

Out[66]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

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

In [69]:
```
clean_data['Age']= clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['Ag
```

In [70]:
```
clean_data['Age']
```

Out[70]:
```
0       34
1       45
2    50.25
3    50.25
4       67
5       55
Name: Age, dtype: object
```

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

Out[71]:
```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

In [72]:
```
clean_data['Exp']= clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['Ex
```

In [73]:
```
clean_data['Exp']
```

Out[73]:
```
0      2
1      3
2      4
3    4.8
4      5
5     10
Name: Exp, dtype: object
```

In [74]:
```
clean_data['Location']
```

Out[74]:
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [77]:
```
clean_data['Location']= clean_data['Location'].fillna(clean_data['Location'].mod
```

In [78]:
```
clean_data['Location']
```

Out[78]:
```
0       Mumbai
1    Bangalore
2    Bangalore
3     Hyderbad
4    Bangalore
5        Delhi
Name: Location, dtype: object
```

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 [96]:
```
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 [97]:
```
clean_data['Age']=clean_data['Age'].astype(int)
clean_data['Exp']=clean_data['Exp'].astype(int)
```

In [98]:
```
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      object  
 5   Exp       6 non-null      int32   
dtypes: category(3), int32(2), object(1)
memory usage: 890.0+ bytes
```

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 [100…
```
clean_data.to_csv('clean_data.csv')
```

In [101…
```
import os
os.getcwd()
```

Out[101… `'C:\\Users\\lenovo'`

In [102…
```
clean_data
```

Out[102…

	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

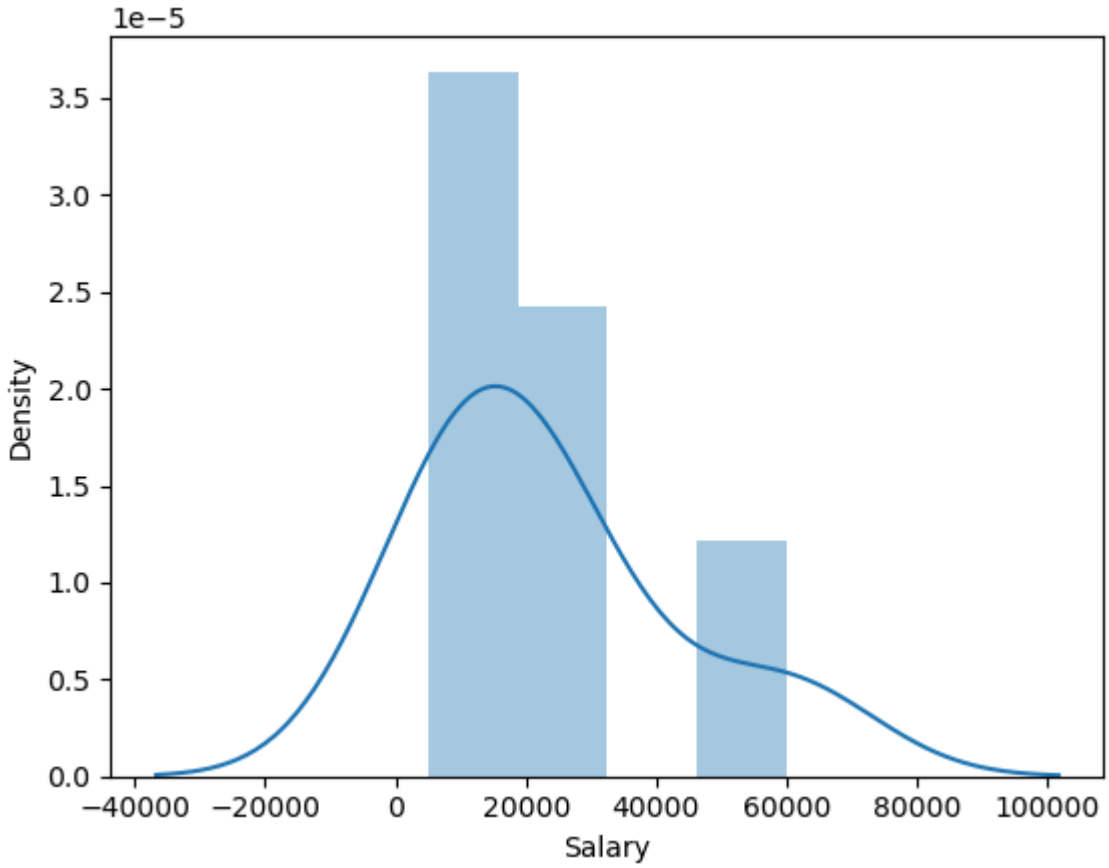
In [103…
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [104…
```
import warnings
warnings.filterwarnings('ignore')
```

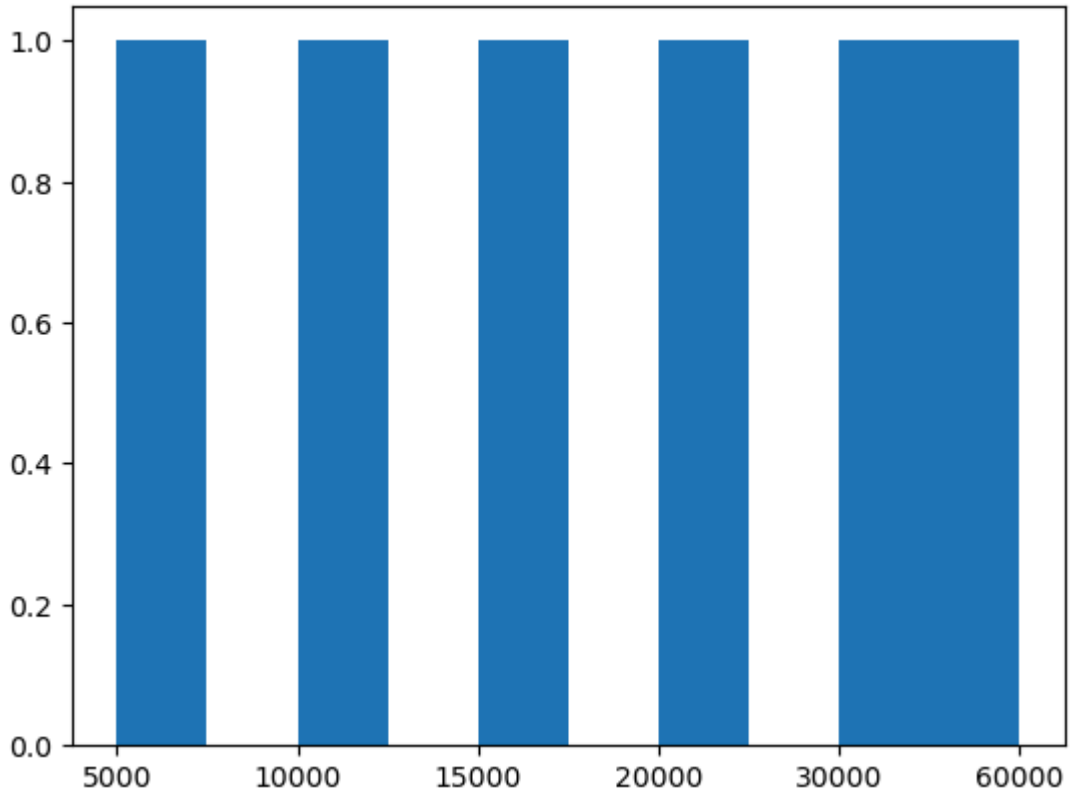
In [105…
```
clean_data['Salary']
```

Out[105…
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

In [106…
```
vis1 = sns.distplot(clean_data['Salary'])
```



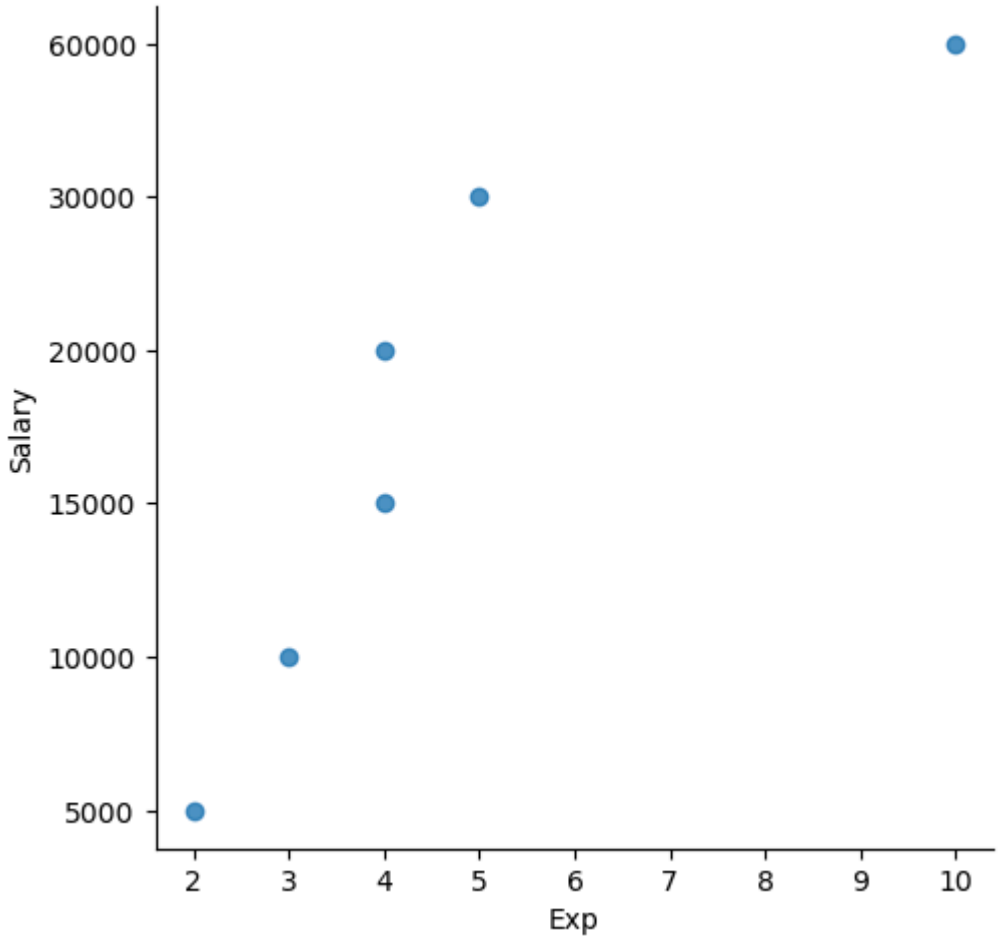
In [107…
```
vis2 = plt.hist(clean_data['Salary'])
```



In [109…
```
vis4 = sns.lmplot(data=clean_data, x='Exp', y='Salary')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[109], line 1
----> 1 vis4 = sns.lmplot(data=clean_data, x='Exp', y='Salary')

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:651, in lmplot(data, x, y, hue, col, row, palette, col_wrap, height, aspect, markers, sharex, sharey, hue_order, col_order, row_order, legend, legend_out, x_estimator, x_bins, x_ci, scatter, fit_reg, ci, n_boot, units, seed, order, logistic, lowess, robust, logx, x_partial, y_partial, truncate, x_jitter, y_jitter, scatter_kws, line_kws, facet_kws)
    642 # Draw the regression plot on each facet
    643 regplot_kws = dict(
    644     x_estimator=x_estimator, x_bins=x_bins, x_ci=x_ci,
    645     scatter=scatter, fit_reg=fit_reg, ci=ci, n_boot=n_boot, units=units,
   (...)
    649     scatter_kws=scatter_kws, line_kws=line_kws,
    650 )
--> 651 facets.map_dataframe(regplot, x=x, y=y, **regplot_kws)
    652 facets.set_axis_labels(x, y)
    654 # Add a legend

File ~\anaconda3\Lib\site-packages\seaborn\axisgrid.py:825, in FacetGrid.map_dataframe(self, func, *args, **kwargs)
    822     kwargs["data"] = data_ijk
    824     # Draw the plot
--> 825     self._facet_plot(func, ax, args, kwargs)
    827 # For axis labels, prefer to use positional args for backcompat
    828 # but also extract the x/y kwargs and use if no corresponding arg
    829 axis_labels = [kwargs.get("x", None), kwargs.get("y", None)]

File ~\anaconda3\Lib\site-packages\seaborn\axisgrid.py:854, in FacetGrid._facet_plot(self, func, ax, plot_args, plot_kwargs)
    852     plot_args = []
    853     plot_kwargs["ax"] = ax
--> 854 func(*plot_args, **plot_kwargs)
    856 # Sort out the supporting information
    857 self._update_legend_data(ax)

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:775, in regplot(data, x, y, x_estimator, x_bins, x_ci, scatter, fit_reg, ci, n_boot, units, seed, order, logistic, lowess, robust, logx, x_partial, y_partial, truncate, dropna, x_jitter, y_jitter, label, color, marker, scatter_kws, line_kws, ax)
    773 scatter_kws["marker"] = marker
    774 line_kws = {} if line_kws is None else copy.copy(line_kws)
--> 775 plotter.plot(ax, scatter_kws, line_kws)
    776 return ax

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:384, in _RegressionPlotter.plot(self, ax, scatter_kws, line_kws)
    381     self.scatterplot(ax, scatter_kws)
    383 if self.fit_reg:
--> 384     self.lineplot(ax, line_kws)
    386 # Label the axes
    387 if hasattr(self.x, "name"):

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:429, in _RegressionPlotter.lineplot(self, ax, kws)
    427 """Draw the model."""
    428 # Fit the regression model
```

```
--> 429 grid, yhat, err_bands = self.fit_regression(ax)
    430 edges = grid[0], grid[-1]
    432 # Get set default aesthetics

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:229, in _RegressionPlotter.fit_regression(self, ax, x_range, grid)
    227     yhat, yhat_boots = self.fit_logx(grid)
    228 else:
--> 229     yhat, yhat_boots = self.fit_fast(grid)
    231 # Compute the confidence interval at each grid point
    232 if ci is None:

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:246, in _RegressionPlotter.fit_fast(self, grid)
    244 X, y = np.c_[np.ones(len(self.x)), self.x], self.y
    245 grid = np.c_[np.ones(len(grid)), grid]
--> 246 yhat = grid.dot(reg_func(X, y))
    247 if self.ci is None:
    248     return yhat, None

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:242, in _RegressionPlotter.fit_fast.<locals>.reg_func(_x, _y)
    241 def reg_func(_x, _y):
--> 242     return np.linalg.pinv(_x).dot(_y)

TypeError: can't multiply sequence by non-int of type 'float'
```

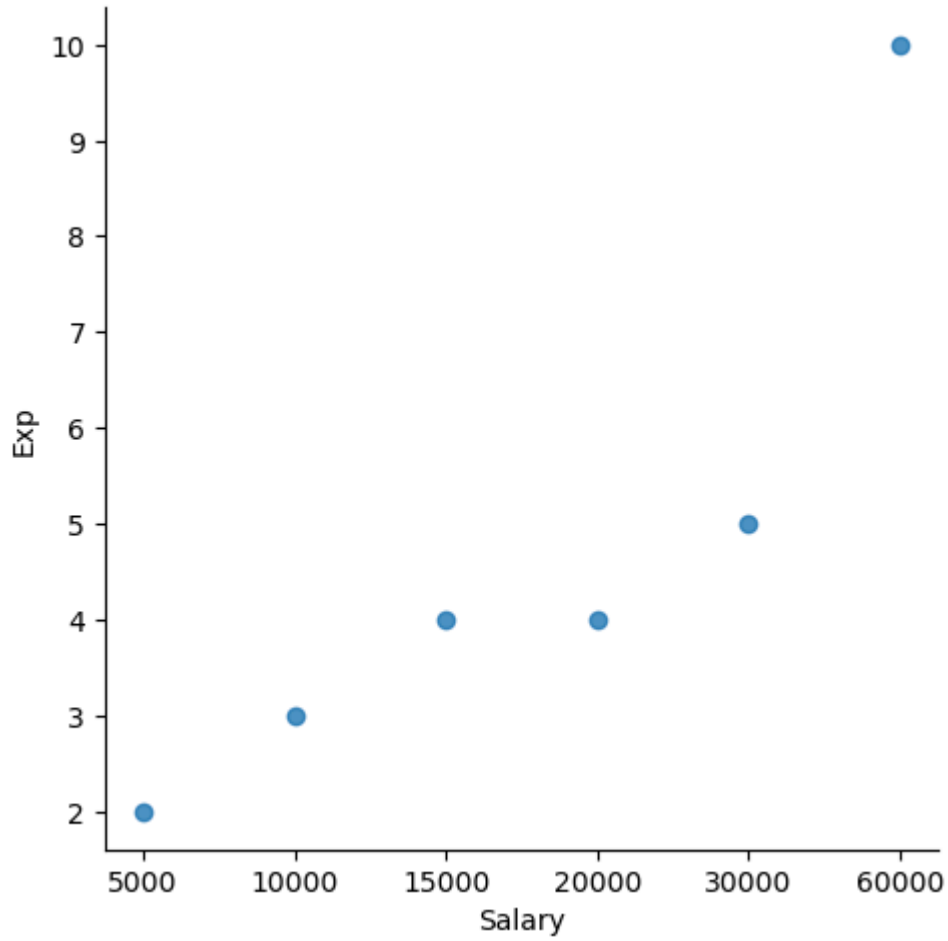


In [112…

```
vis5=sns.lmplot(data=clean_data, y='Exp', x='Salary', fit_reg=True)
```

```
---------------------------------------------------------------------------
UFuncTypeError                            Traceback (most recent call last)
Cell In[112], line 1
----> 1 vis5=sns.lmplot(data=clean_data, y='Exp', x='Salary', fit_reg=True)

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:651, in lmplot(data, x, y, hue, col, row, palette, col_wrap, height, aspect, markers, sharex, sharey, hue_order, col_order, row_order, legend, legend_out, x_estimator, x_bins, x_ci, scatter, fit_reg, ci, n_boot, units, seed, order, logistic, lowess, robust, logx, x_partial, y_partial, truncate, x_jitter, y_jitter, scatter_kws, line_kws, facet_kws)
    642 # Draw the regression plot on each facet
    643 regplot_kws = dict(
    644     x_estimator=x_estimator, x_bins=x_bins, x_ci=x_ci,
    645     scatter=scatter, fit_reg=fit_reg, ci=ci, n_boot=n_boot, units=units,
   (...)
    649     scatter_kws=scatter_kws, line_kws=line_kws,
    650 )
--> 651 facets.map_dataframe(regplot, x=x, y=y, **regplot_kws)
    652 facets.set_axis_labels(x, y)
    654 # Add a legend

File ~\anaconda3\Lib\site-packages\seaborn\axisgrid.py:825, in FacetGrid.map_dataframe(self, func, *args, **kwargs)
    822     kwargs["data"] = data_ijk
    824     # Draw the plot
--> 825     self._facet_plot(func, ax, args, kwargs)
    827 # For axis labels, prefer to use positional args for backcompat
    828 # but also extract the x/y kwargs and use if no corresponding arg
    829 axis_labels = [kwargs.get("x", None), kwargs.get("y", None)]

File ~\anaconda3\Lib\site-packages\seaborn\axisgrid.py:854, in FacetGrid._facet_plot(self, func, ax, plot_args, plot_kwargs)
    852     plot_args = []
    853     plot_kwargs["ax"] = ax
--> 854 func(*plot_args, **plot_kwargs)
    856 # Sort out the supporting information
    857 self._update_legend_data(ax)

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:775, in regplot(data, x, y, x_estimator, x_bins, x_ci, scatter, fit_reg, ci, n_boot, units, seed, order, logistic, lowess, robust, logx, x_partial, y_partial, truncate, dropna, x_jitter, y_jitter, label, color, marker, scatter_kws, line_kws, ax)
    773 scatter_kws["marker"] = marker
    774 line_kws = {} if line_kws is None else copy.copy(line_kws)
--> 775 plotter.plot(ax, scatter_kws, line_kws)
    776 return ax

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:384, in _RegressionPlotter.plot(self, ax, scatter_kws, line_kws)
    381     self.scatterplot(ax, scatter_kws)
    383 if self.fit_reg:
--> 384     self.lineplot(ax, line_kws)
    386 # Label the axes
    387 if hasattr(self.x, "name"):

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:429, in _RegressionPlotter.lineplot(self, ax, kws)
    427 """Draw the model."""
    428 # Fit the regression model
```

```
--> 429 grid, yhat, err_bands = self.fit_regression(ax)
    430 edges = grid[0], grid[-1]
    432 # Get set default aesthetics

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:209, in _RegressionPlotter.fit_regression(self, ax, x_range, grid)
    207         else:
    208             x_min, x_max = ax.get_xlim()
--> 209     grid = np.linspace(x_min, x_max, 100)
    210 ci = self.ci
    212 # Fit the regression

File ~\anaconda3\Lib\site-packages\numpy\core\function_base.py:129, in linspace(start, stop, num, endpoint, retstep, dtype, axis)
    125 div = (num - 1) if endpoint else num
    127 # Convert float/complex array scalars to float, gh-3504
    128 # and make sure one can use variables that have an __array_interface__, gh-6634
--> 129 start = asanyarray(start) * 1.0
    130 stop  = asanyarray(stop)  * 1.0
    132 dt = result_type(start, stop, float(num))

UFuncTypeError: ufunc 'multiply' did not contain a loop with signature matching types (dtype('<U5'), dtype('float64')) -> None
```



In [113…

```
clean_data[:]
```

Out[113…

	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 [115…
```
X_iv = clean_data[['Name', 'Domain', 'Age', 'Location', 'Exp']]
```

In [116…
```
X_iv
```

Out[116…

	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 [119…
```
Y_dv = clean_data[['Salary']]
```

In [120…
```
Y_dv
```

Out[120…

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

In [121…
```
emp
```

Out[121...

	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 [122...
```
clean_data
```

Out[122...

	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 [123...
```
imputation = pd.get_dummies(clean_data, dtype=int)
```

In [124...
```
imputation
```

Out[124...

	Age	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Umar	Name_U
0	34	2	0	0	1	0	0	
1	45	3	0	0	0	1	0	
2	50	4	0	0	0	0	1	
3	50	4	1	0	0	0	0	
4	67	5	0	0	0	0	0	
5	55	10	0	1	0	0	0	

6 rows × 24 columns

In [125...
```
len(clean_data)
```

Out[125... 6

In [126...
```
imputation.columns
```

Out[126…
```
Index(['Age', 'Exp', 'Name_Jane', 'Name_Kim', 'Name_Mike', 'Name_Teddy',
       'Name_Umar', 'Name_Uttam', 'Domain_Analytics', 'Domain_Dataanalyst',
       'Domain_Datascience', 'Domain_NLP', 'Domain_Statistics',
       'Domain_Testing', 'Location_Bangalore', 'Location_Delhi',
       'Location_Hyderbad', 'Location_Mumbai', 'Salary_10000', 'Salary_15000',
       'Salary_20000', 'Salary_30000', 'Salary_5000', 'Salary_60000'],
      dtype='object')
```

In [127…
```
len(imputation.columns)
```

Out[127…
```
24
```