In [6]:
```
import pandas as pd
```

In [8]:
```
pd.__version__
```

Out[8]:
```
'2.3.1'
```

In [70]:
```
emp = pd.read_excel(r"C:\Users\milind rajput\Downloads\Rawdata.xlsx")
```

In [72]:
```
emp
```

Out[72]:

	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 [74]:
```
id(emp)
```

Out[74]:
```
1742942710784
```

In [78]:
```
emp.head()
```

Out[78]:

	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 [80]:
```
emp.columns
```

Out[80]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [82]:
```
emp.shape
```

Out[82]:
```
(6, 6)
```

In [86]:
```
emp.tail()
```

Out[86]:

	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 [88]:
```
emp.head()
```

Out[88]:

	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 [90]:
```
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 [92]:
```
emp.isnull()
```

Out[92]:

	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 [94]:
```
emp.isna()
```

Out[94]:

	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 [98]:
```
emp.isnull().sum()
```

Out[98]:
```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

In [ ]:
```
# Data cleaning or data cleansing
```

In [100…
```
emp
```

Out[100…

	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 [126…
```
emp['Name']= emp['Name'].str.replace(r'\W','',regex = True)
emp['Name']
```

Out[126…
```
0     Mike
1    Teddy
2     Umar
3     Jane
4    Uttam
5      Kim
Name: Name, dtype: object
```

In [128…
```
emp
```

Out[128...

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience#$	34years	Mumbai	5^00#0	2+
1	Teddy	Testing	45yr	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	67yr	NaN	30000-	5+ year
5	Kim	NLP	55yr	Delhi	6000^$0	10+

In [130...
```
emp['Domain']=emp['Domain'].str.replace(r'\W','',regex = True)
emp['Domain']
```

Out[130...
```
0    Datascience
1        Testing
2    Dataanalyst
3      Analytics
4     Statistics
5            NLP
Name: Domain, dtype: object
```

In [132...
```
emp
```

Out[132...

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34years	Mumbai	5^00#0	2+
1	Teddy	Testing	45yr	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	67yr	NaN	30000-	5+ year
5	Kim	NLP	55yr	Delhi	6000^$0	10+

In [134...
```
emp['Age']=emp['Age'].str.replace(r'\W','',regex = True)
emp['Age']
```

Out[134...
```
0    34years
1       45yr
2        NaN
3        NaN
4       67yr
5       55yr
Name: Age, dtype: object
```

In [136...
```
emp
```

Out[136…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34years	Mumbai	5^00#0	2+
1	Teddy	Testing	45yr	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	67yr	NaN	30000-	5+ year
5	Kim	NLP	55yr	Delhi	6000^$0	10+

In [140…
```
emp['Age'] = emp['Age'].str.extract('(\\d+)')
emp['Age']
```

Out[140…
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [142…
```
emp
```

Out[142…

	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 [144…
```
emp['Location'].str.replace(r'\W','',regex = True)
emp['Location']
```

Out[144…
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [146…
```
emp
```

Out[146...

	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 [150...
```
emp['Salary']=emp['Salary'].str.replace(r'\W','',regex =True)
emp['Salary']
```

Out[150...
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

In [152...
```
emp
```

Out[152...

	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 [156...
```
emp['Exp']= emp['Exp'].str.extract('(\\d+)')
emp['Exp']
```

Out[156...
```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

In [158...
```
emp
```

Out[158...

	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 [326...

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

Out[326...

	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 Techniques We Apply

In [328...

```
clean_data
```

Out[328...

	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 [ ]:

```
# for Numerical values apply mean(),Median(),Mode()
```

In [330...

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

Out[330…
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [332…
```
import numpy as np
```

In [168…
```
# fillna() filling Nan values
clean_data['Age'] = clean_data['Age'].fillna (np.mean(pd.to_numeric(clean_data['
```

In [334…
```
clean_data['Age']
```

Out[334…
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [ ]:
```
# in Machine Learning we use transforemer
```

In [336…
```
# fillna() filling Nan values/
# Exp column we filling Missing values
clean_data['Exp'] = clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['E
```

In [338…
```
clean_data['Exp']
```

Out[338…
```
0      2
1      3
2      4
3    4.8
4      5
5     10
Name: Exp, dtype: object
```

In [340…
```
# Categorical Missing Values
# Mode(),KNN
# Location column clean
clean_data['Location']= clean_data['Location'].fillna(clean_data['Location'].mod
```

In [342…
```
clean_data['Location']
 # Choosing Banglore as Nan Value behind that their is euclidean distance
```

Out[342…
```
0       Mumbai
1    Bangalore
2    Bangalore
3     Hyderbad
4    Bangalore
5        Delhi
Name: Location, dtype: object
```

In [344…
```
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
```

emp

In [346…
```
emp
```

Out[346…

	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 [348…
```
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  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 [193…
```
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 [350…

```
clean_data
```

Out[350…

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

In [352…

```
# astype() system build in data type to user wants data type
clean_data['Age'] = clean_data['Age'].astype(int)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[352], line 2
      1 # astype() system build in data type to user wants data type
----> 2 clean_data['Age'] = clean_data['Age'].astype(int)

File ~\anaconda3\Lib\site-packages\pandas\core\generic.py:6662, in NDFrame.astype
(self, dtype, copy, errors)
   6656     results = [
   6657         ser.astype(dtype, copy=copy, errors=errors) for _, ser in self.it
ems()
   6658     ]
   6660 else:
   6661     # else, only a single dtype is given
-> 6662     new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)
   6663     res = self._constructor_from_mgr(new_data, axes=new_data.axes)
   6664     return res.__finalize__(self, method="astype")

File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:430, in Base
BlockManager.astype(self, dtype, copy, errors)
    427 elif using_copy_on_write():
    428     copy = False
--> 430 return self.apply(
    431     "astype",
    432     dtype=dtype,
    433     copy=copy,
    434     errors=errors,
    435     using_cow=using_copy_on_write(),
    436 )

File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:363, in Base
BlockManager.apply(self, f, align_keys, **kwargs)
    361         applied = b.apply(f, **kwargs)
    362     else:
--> 363         applied = getattr(b, f)(**kwargs)
    364     result_blocks = extend_blocks(applied, result_blocks)
    366 out = type(self).from_blocks(result_blocks, self.axes)

File ~\anaconda3\Lib\site-packages\pandas\core\internals\blocks.py:784, in Block.
astype(self, dtype, copy, errors, using_cow, squeeze)
    781         raise ValueError("Can not squeeze with more than one column.")
    782     values = values[0, :]  # type: ignore[call-overload]
--> 784 new_values = astype_array_safe(values, dtype, copy=copy, errors=errors)
    786 new_values = maybe_coerce_values(new_values)
    788 refs = None

File ~\anaconda3\Lib\site-packages\pandas\core\dtypes\astype.py:237, in astype_ar
ray_safe(values, dtype, copy, errors)
    234     dtype = dtype.numpy_dtype
    236 try:
--> 237     new_values = astype_array(values, dtype, copy=copy)
    238 except (ValueError, TypeError):
    239     # e.g. _astype_nansafe can fail on object-dtype of strings
    240     #  trying to convert to float
    241     if errors == "ignore":

File ~\anaconda3\Lib\site-packages\pandas\core\dtypes\astype.py:182, in astype_ar
ray(values, dtype, copy)
    179     values = values.astype(dtype, copy=copy)
    181 else:
```

```
--> 182     values = _astype_nansafe(values, dtype, copy=copy)
    184 # in pandas we don't store numpy str dtypes, so convert to object
    185 if isinstance(dtype, np.dtype) and issubclass(values.dtype.type, str):

File ~\anaconda3\Lib\site-packages\pandas\core\dtypes\astype.py:133, in _astype_n
ansafe(arr, dtype, copy, skipna)
    129     raise ValueError(msg)
    131 if copy or arr.dtype == object or dtype == object:
    132     # Explicit copy, or required since NumPy can't view from / to object.
--> 133     return arr.astype(dtype, copy=True)
    135 return arr.astype(dtype, copy=copy)

ValueError: cannot convert float NaN to integer
```

In [ ]:
```
clean_data.info()
```

In [205…
```
clean_data['Slary'] = clean_data['Salary'].astype(int)
clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 7 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
 6   Slary     6 non-null      int32 
dtypes: int32(2), object(5)
memory usage: 420.0+ bytes
```

In [ ]:
```
clean_data['Exp'] = clean_data['Exp'].astype(int)
clean_data.info()
```

In [356…
```
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 [358…
```
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       4 non-null      object  
 3   Location  6 non-null      category
 4   Salary    6 non-null      object  
 5   Exp       6 non-null      object  
dtypes: category(3), object(3)
memory usage: 938.0+ bytes
```

In [360…
```
clean_data
```

Out[360…

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

In [362…
```
clean_data.to_csv('clean_data.csv')
```

In [364…
```
import os
os.getcwd()
```

Out[364…
```
'C:\\Users\\milind rajput\\FSDS Course'
```

In [366…
```
clean_data
```

Out[366…

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

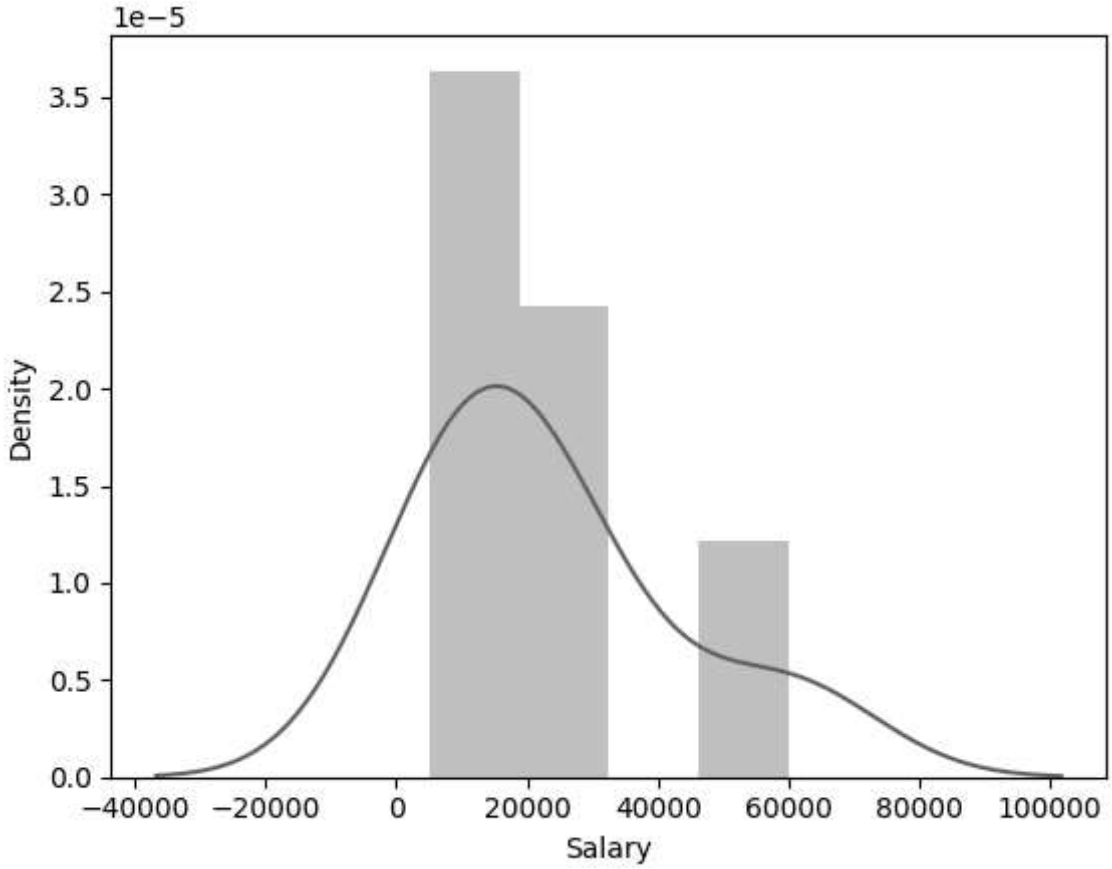
In [368…
```
# Univariat
# Bivariat
import matplotlib.pyplot as plt
import seaborn as sns
```

In [370…
```
import warnings
warnings.filterwarnings('ignore')
```

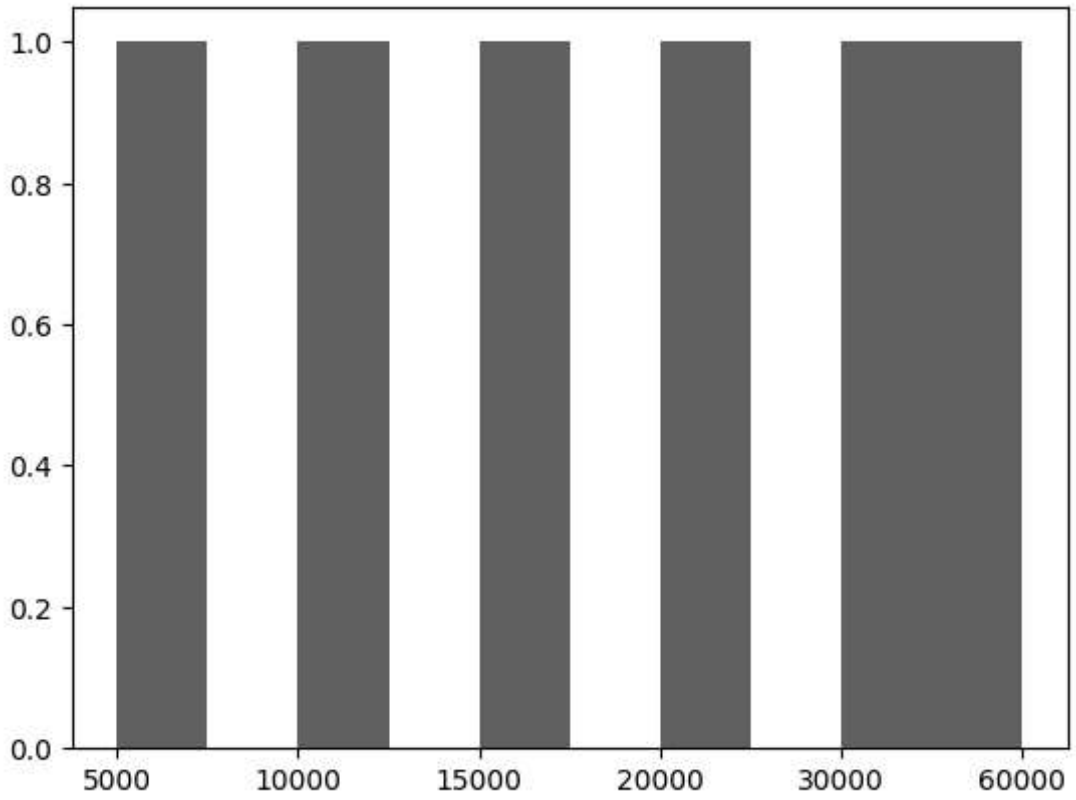
In [372…
```
clean_data['Salary']
```

Out[372…
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

In [374…
```
vis1 = sns.distplot(clean_data['Salary'])
# univariat  Analysis
```



In [376…
```
vis2 = plt.hist(clean_data['Salary'])
```



In [378…
```
clean_data
```

Out[378...

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

In [392...

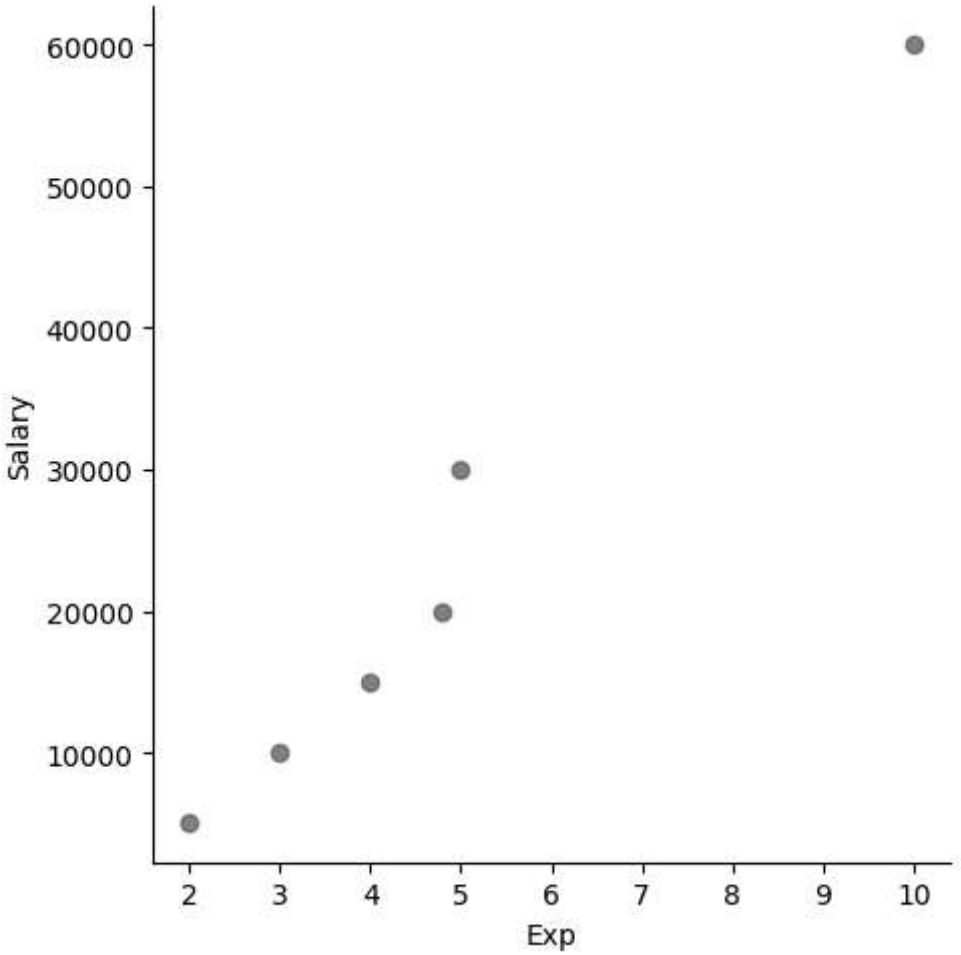
```
# Bivariat Analysis
import seaborn as sns
import matplotlib.pyplot as plt

# Drop missing and make sure types are numeric
clean_data = clean_data.dropna(subset=['Exp', 'Salary'])
clean_data['Exp'] = pd.to_numeric(clean_data['Exp'], errors='coerce')
clean_data['Salary'] = pd.to_numeric(clean_data['Salary'], errors='coerce')

# Bivariate Analysis plot
vis4 = sns.lmplot(data=clean_data, x='Exp', y='Salary', fit_reg=True, height=5,

plt.title("Experience vs Salary")
plt.xlabel("Experience (Years)")
plt.ylabel("Salary")
plt.show()
```

In [396...

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

Out[396...

```
<seaborn.axisgrid.FacetGrid at 0x19583180cb0>
```



In [398…
```
clean_data
```

Out[398…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2.0
1	Teddy	Testing	45	Bangalore	10000	3.0
2	Umar	Dataanalyst	NaN	Bangalore	15000	4.0
3	Jane	Analytics	NaN	Hyderbad	20000	4.8
4	Uttam	Statistics	67	Bangalore	30000	5.0
5	Kim	NLP	55	Delhi	60000	10.0

In [400…
```
clean_data[0:6:2]
```

Out[400…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2.0
2	Umar	Dataanalyst	NaN	Bangalore	15000	4.0
4	Uttam	Statistics	67	Bangalore	30000	5.0

In [402…
```
clean_data
```

Out[402...

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2.0
1	Teddy	Testing	45	Bangalore	10000	3.0
2	Umar	Dataanalyst	NaN	Bangalore	15000	4.0
3	Jane	Analytics	NaN	Hyderbad	20000	4.8
4	Uttam	Statistics	67	Bangalore	30000	5.0
5	Kim	NLP	55	Delhi	60000	10.0

In [404...
```
clean_data[::-1]
```

Out[404...

	Name	Domain	Age	Location	Salary	Exp
5	Kim	NLP	55	Delhi	60000	10.0
4	Uttam	Statistics	67	Bangalore	30000	5.0
3	Jane	Analytics	NaN	Hyderbad	20000	4.8
2	Umar	Dataanalyst	NaN	Bangalore	15000	4.0
1	Teddy	Testing	45	Bangalore	10000	3.0
0	Mike	Datascience	34	Mumbai	5000	2.0

In [406...
```
clean_data.columns
```

Out[406...
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [408...
```
X_iv = clean_data[['Name', 'Domain', 'Age', 'Location', 'Exp']]
```

In [410...
```
X_iv
```

Out[410...

	Name	Domain	Age	Location	Exp
0	Mike	Datascience	34	Mumbai	2.0
1	Teddy	Testing	45	Bangalore	3.0
2	Umar	Dataanalyst	NaN	Bangalore	4.0
3	Jane	Analytics	NaN	Hyderbad	4.8
4	Uttam	Statistics	67	Bangalore	5.0
5	Kim	NLP	55	Delhi	10.0

In [412...
```
y_dv = clean_data[['Salary']]
```

In [414...
```
y_dv
```

Out[414…

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

In [416…
```
emp
```

Out[416…

	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 [420…
```
x_iv
```

Out[420…
```
[['Name', 'Domain', 'Age', 'Loction', 'Exp']]
```

In [422…
```
clean_data
```

Out[422…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2.0
1	Teddy	Testing	45	Bangalore	10000	3.0
2	Umar	Dataanalyst	NaN	Bangalore	15000	4.0
3	Jane	Analytics	NaN	Hyderbad	20000	4.8
4	Uttam	Statistics	67	Bangalore	30000	5.0
5	Kim	NLP	55	Delhi	60000	10.0

In [424…
```
imputation = pd.get_dummies(clean_data)
```

In [426…
```
imputation
```

Out[426…

	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Umar	Name
0	5000	2.0	False	False	True	False	False	
1	10000	3.0	False	False	False	True	False	
2	15000	4.0	False	False	False	False	True	
3	20000	4.8	True	False	False	False	False	
4	30000	5.0	False	False	False	False	False	
5	60000	10.0	False	True	False	False	False	

6 rows × 22 columns

In [428…

```
clean_data
```

Out[428…

	Name	Domain	Age	Location	Salary	Exp
0	Mike	Datascience	34	Mumbai	5000	2.0
1	Teddy	Testing	45	Bangalore	10000	3.0
2	Umar	Dataanalyst	NaN	Bangalore	15000	4.0
3	Jane	Analytics	NaN	Hyderbad	20000	4.8
4	Uttam	Statistics	67	Bangalore	30000	5.0
5	Kim	NLP	55	Delhi	60000	10.0

In [430…

```
imputation
```

Out[430…

	Salary	Exp	Name_Jane	Name_Kim	Name_Mike	Name_Teddy	Name_Umar	Name
0	5000	2.0	False	False	True	False	False	
1	10000	3.0	False	False	False	True	False	
2	15000	4.0	False	False	False	False	True	
3	20000	4.8	True	False	False	False	False	
4	30000	5.0	False	False	False	False	False	
5	60000	10.0	False	True	False	False	False	

6 rows × 22 columns

In [ ]:

In [ ]: