In [1]:
```
import pandas as pd
```

In [2]:
```
pd.__version__
```

Out[2]:
```
'2.2.2'
```

In [3]:
```
eda=pd.read_excel(r'C:\Users\HP\Downloads\Rawdata.xlsx')
eda
```

Out[3]:

	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 [4]:
```
eda.head()
```

Out[4]:

	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 [5]:
```
eda.tail()
```

Out[5]:

	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 [6]:
```
eda.isnull().any().any
```

Out[6]:
```
<bound method Series.any of Name        False
Domain      False
Age          True
Location     True
Salary      False
Exp          True
dtype: bool>
```

In [7]:
```
id(eda)
```

Out[7]:
```
1271245177920
```

In [8]:
```
eda.columns
```

Out[8]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [9]:
```
eda.shape
```

Out[9]:
```
(6, 6)
```

In [10]:
```
eda.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 [11]:
```
eda
```

Out[11]:

	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 [12]:
```
eda.isnull()
```

Out[12]:

	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 [13]:
```
eda.isna()
```

Out[13]:

	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 [14]:
```
eda.isnull().sum()
```

Out[14]:
```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

In [15]:
```
eda.columns
```

Out[15]:
```
Index(['Name', 'Domain', 'Age', 'Location', 'Salary', 'Exp'], dtype='object')
```

In [16]:
```
eda
```

Out[16]:

	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+

Data Cleaning or Data Cleansing

In [18]:
```
eda['Name']
```

Out[18]:
```
0      Mike
1    Teddy^
2     Uma#r
3      Jane
4    Uttam*
5       Kim
Name: Name, dtype: object
```

In [19]:
```
eda['Name'] = eda['Name'].str.replace(r'\W','',regex=True) # type only capital '
# It removes all speicla characters. # '' = it replaces and returns empty.
# '\W' = This is a regular expression (regex) pattern.
# In regex, \W matches any non-word character.
# A "word character" is defined as any letter (a-z, A-Z), digit (0-9), or unders
# So, \W matches anything that is not a letter, digit, or underscore—
# such as spaces, punctuation marks (e.g., !, ?, ., ,), or special characters (e
```

In [20]:
```
eda['Name']
```

Out[20]:
```
0     Mike
1    Teddy
2     Umar
3     Jane
4    Uttam
5      Kim
Name: Name, dtype: object
```

In [21]:
```
eda['Domain'] = eda['Domain'].str.replace(r'\W','',regex=True)
eda['Domain']
```

Out[21]:
```
0    Datascience
1        Testing
2    Dataanalyst
3      Analytics
4     Statistics
5            NLP
Name: Domain, dtype: object
```

In [22]:
```
eda['Age'] = eda['Age'].str.replace(r'\W','',regex=True)
eda['Age']
```

Out[22]:
```
0    34years
1       45yr
2        NaN
3        NaN
4       67yr
5       55yr
Name: Age, dtype: object
```

In [23]:
```
eda['Age'] = eda['Age'].str.extract('(\d+)')
eda['Age'] # \d: Matches any single digit (0-9), "+" = one or more occurences
```

```
<>:1: SyntaxWarning: invalid escape sequence '\d'
<>:1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\HP\AppData\Local\Temp\ipykernel_11484\3740922015.py:1: SyntaxWarning: in
valid escape sequence '\d'
  eda['Age'] = eda['Age'].str.extract('(\d+)')
```

Out[23]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [24]:
```
eda
```

Out[24]:

	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 [25]:
```
eda['Location'] = eda['Location'].str.replace(r'\W','',regex=True)
eda['Location']
```

Out[25]:
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [26]:
```
eda['Salary'] = eda['Salary'].str.replace(r'\W','',regex=True)
eda['Salary']
```

Out[26]:
```
0     5000
1    10000
2    15000
3    20000
4    30000
5    60000
Name: Salary, dtype: object
```

In [27]:
```
eda['Exp'] = eda['Exp'].str.replace(r'\W','',regex=True)
eda['Exp']
```

Out[27]:
```
0        2
1        3
2     4yrs
3      NaN
4    5year
5       10
Name: Exp, dtype: object
```

In [28]:
```
eda['Exp'] = eda['Exp'].str.extract('(\d+)')
eda['Exp'] # if any errors use this r(r'(\\d+)')
```

```
<>:1: SyntaxWarning: invalid escape sequence '\d'
<>:1: SyntaxWarning: invalid escape sequence '\d'
C:\Users\HP\AppData\Local\Temp\ipykernel_11484\1090334937.py:1: SyntaxWarning: in
valid escape sequence '\d'
  eda['Exp'] = eda['Exp'].str.extract('(\d+)')
```

Out[28]:
```
0      2
1      3
2      4
3    NaN
4      5
5     10
Name: Exp, dtype: object
```

In [29]:
```
eda
```

Out[29]:

	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 [30]:
```
clean_data = eda.copy()
```

In [31]:
```
clean_data
```

Out[31]:

	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 [32]:
```
# till now we removed
```

Lets Apply EDA Techniques

1. Missing value treatment

In [34]:
```
clean_data.isnull().sum()
```

Out[34]:
```
Name        0
Domain      0
Age         2
Location    2
Salary      0
Exp         1
dtype: int64
```

In [35]:
```
clean_data['Age']
```

Out[35]:
```
0     34
1     45
2    NaN
3    NaN
4     67
5     55
Name: Age, dtype: object
```

In [36]:
```
import numpy as np
```

In [37]:
```
clean_data['Age'] = clean_data['Age'].fillna(np.mean(pd.to_numeric(clean_data['A
clean_data['Age'] # .fillna() fills the null values using mean strategy.we can a
```

Out[37]:
```
0       34
1       45
2    50.25
3    50.25
4       67
5       55
Name: Age, dtype: object
```

In [38]:
```
clean_data['Exp'] = clean_data['Exp'].fillna(np.mean(pd.to_numeric(clean_data['E
clean_data['Exp']
```

Out[38]:
```
0      2
1      3
2      4
3    4.8
4      5
5     10
Name: Exp, dtype: object
```

In [39]:
```
clean_data
```

Out[39]:

	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 [40]:
```
clean_data['Location'].isnull().sum()
```

Out[40]:
```
2
```

In [41]:
```
clean_data['Location']
```

Out[41]:
```
0       Mumbai
1    Bangalore
2          NaN
3     Hyderbad
4          NaN
5        Delhi
Name: Location, dtype: object
```

In [42]:
```
clean_data['Location'] = clean_data['Location'].fillna(clean_data['Location'].mo
clean_data['Location'] # give mode value as "0". don't change. It's the main log
# whenever we use mode use "0". this is for categorical values only.
```

Out[42]:
```
0       Mumbai
1    Bangalore
2    Bangalore
3     Hyderbad
4    Bangalore
5        Delhi
Name: Location, dtype: object
```

In [43]:
```
clean_data
```

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	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 [44]:
```
clean_data.info() # data is cleaned
```

```
<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: 420.0+ bytes
```

In [45]:
```
clean_data['Age'] = clean_data['Age'].astype(int)
clean_data['Age'] # astype() converts system datatype to user datatype.
```

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

In [46]:
```
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 [47]:
```
clean_data['Salary'] = clean_data['Salary'].astype(int)
clean_data['Exp'] = clean_data['Exp'].astype(int)
```

In [48]:
```
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 [49]:
```
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 [50]:
```
clean_data.to_csv('clean_data.csv')
```

In [51]:
```
import os
os.getcwd()
```

Out[51]:
```
'C:\\Users\\HP'
```

In [108…
```
from PIL import Image
```

In [110…
```
data = Image.open(r'C:\Users\HP\Pictures\Screenshots\EDA-Cleansed-data.png')
data
```

Out[110…

RAW-DATA

	A	B	C	D	E	F	G
1	**Name**	**Domain**	**Age**	**Location**	**Salary**	**Exp**	
2	Mike	Datascience#$	34 years	Mumbai	5^00#0	2+	
3	Teddy^	Testing	45' yr	Bangalore	10%%000	<3	
4	Uma#r	Dataanalyst^^#			1$5%000	4> yrs	
5	Jane	Ana^^lytics		Hyderbad	2000^0		
6	Uttam*	Statistics	67-yr		30000-	5+ year	
7	Kim	NLP	55yr	Delhi	6000^$0	10+	
8							

CLEANED-DATA

	A	B	C	D	E	F	G	H
1		Name	Domain	Age	Location	Salary	Exp	
2	0	Mike	Datascience	34	Mumbai	5000	2	
3	1	Teddy	Testing	45	Bangalore	10000	3	
4	2	Umar	Dataanalyst	50	Bangalore	15000	4	
5	3	Jane	Analytics	50	Hyderbad	20000	4	
6	4	Uttam	Statistics	67	Bangalore	30000	5	
7	5	Kim	NLP	55	Delhi	60000	10	
8								

In [ ]: