In [15]:

```
import pandas as pd
#to use python libraries we import the required libraries using "import"
# NOTE : Python is case Sensitive
data = {
    'Name':['Alice','Bob','Charlie'],
    'Age':[25,30,35]
}
df = pd.DataFrame(data)
#print(df)
#print(df.head(8))
#print(df.tail(5))
```

In [16]:

```
df = pd.read_csv('data.csv')
print(df)
#To print the data set  uploaded
```

```
   ID      Name   Age  Gender         City     Salary  Experience
0   1      John  28.0    Male     New York    50000.0           3
1   2      Sara   NaN  Female  Los Angeles    52000.0           4
2   3     David  35.0    Male      Chicago        NaN           8
3   4     Linda  29.0  Female     New york    58000.0           5
4   5   Michael  41.0    Male        Miami    62000.0          10
5   6      Emma   NaN  Female  Los Angeles    54000.0           4
6   7    Robert  23.0    Male      CHICAGO    49000.0           2
7   8    Olivia  32.0  Female        Miami  1000000.0           7
8   9     James  27.0    Male     New York    51000.0           3
9  10  Patricia  38.0  Female  Los Angeles        NaN           9
```

In [5]:

```
print(df.to_string())
#Use df.to_string() to print with some start value and a end value when the data is too large
```

```
      Name  Age
0    Alice   25
1      Bob   30
2  Charlie   35
```

In [17]:

```
print(df.head(8))
```

```
   ID     Name   Age  Gender         City     Salary  Experience
0   1     John  28.0    Male     New York    50000.0           3
1   2     Sara   NaN  Female  Los Angeles    52000.0           4
2   3    David  35.0    Male      Chicago        NaN           8
3   4    Linda  29.0  Female     New york    58000.0           5
4   5  Michael  41.0    Male        Miami    62000.0          10
5   6     Emma   NaN  Female  Los Angeles    54000.0           4
6   7   Robert  23.0    Male      CHICAGO    49000.0           2
7   8   Olivia  32.0  Female        Miami  1000000.0           7
```

In [18]:

```
print(df.tail(5))
```

```
   ID      Name   Age  Gender         City     Salary  Experience
5   6      Emma   NaN  Female  Los Angeles    54000.0           4
6   7    Robert  23.0    Male      CHICAGO    49000.0           2
7   8    Olivia  32.0  Female        Miami  1000000.0           7
8   9     James  27.0    Male     New York    51000.0           3
9  10  Patricia  38.0  Female  Los Angeles        NaN           9
```

In [19]:

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   ID          10 non-null     int64  
 1   Name        10 non-null     object 
 2   Age         8 non-null      float64
 3   Gender      10 non-null     object 
 4   City        10 non-null     object 
 5   Salary      8 non-null      float64
 6   Experience  10 non-null     int64  
dtypes: float64(2), int64(2), object(3)
memory usage: 692.0+ bytes
None
```

In [20]:

```
print(df.describe())
```

```
             ID        Age          Salary  Experience
count  10.00000   8.000000        8.000000   10.000000
mean    5.50000  31.625000   172000.000000    5.500000
std     3.02765   6.045955   334591.008333    2.798809
min     1.00000  23.000000    49000.000000    2.000000
25%     3.25000  27.750000    50750.000000    3.250000
50%     5.50000  30.500000    53000.000000    4.500000
75%     7.75000  35.750000    59000.000000    7.750000
max    10.00000  41.000000  1000000.000000   10.000000
```

In [22]:

```
print(df.isnull().sum())
#Number of all null values in the data set
```

```
ID            0
Name          0
Age           2
Gender        0
City          0
Salary        2
Experience    0
dtype: int64
```

In [23]:

```
a =df['Age'].mean()#mean of age
print(a)
```

```
31.625
```

In [25]:

```
b = df['Salary'].mean()
print(b)
#sum() Sum
#std standard deviation
```

```
172000.0
```

In [26]:

```
#data types of data
print(df.dtypes)
```

```
ID              int64
Name           object
Age           float64
Gender         object
City           object
Salary        float64
Experience      int64
dtype: object
```

In [27]:

```
#print columns in data set
print(df.columns)
```

```
Index(['ID', 'Name', 'Age', 'Gender', 'City', 'Salary', 'Experience'], dtype='object')
```

In [37]:

```
#convert Column heading to capital letters
df.columns = df.columns.str.upper()
print(df)
```

```
   ID      NAME   AGE  GENDER         CITY     SALARY  EXPERIENCE
0   1      John  28.0    Male     New York    50000.0           3
1   2      Sara   NaN  Female  Los Angeles    52000.0           4
2   3     David  35.0    Male      Chicago        NaN           8
3   4     Linda  29.0  Female     New york    58000.0           5
4   5   Michael  41.0    Male        Miami    62000.0          10
5   6      Emma   NaN  Female  Los Angeles    54000.0           4
6   7    Robert  23.0    Male      CHICAGO    49000.0           2
7   8    Olivia  32.0  Female        Miami  1000000.0           7
8   9     James  27.0    Male     New York    51000.0           3
9  10  Patricia  38.0  Female  Los Angeles        NaN           9
```

In [39]:

```
#display the unique values in data set
print(df.nunique())
```

```
ID            10
NAME          10
AGE            8
GENDER         2
CITY           6
SALARY         8
EXPERIENCE     8
dtype: int64
```

In [36]:

```
#check null value in column in each
print(df.isnull().sum()/len(df)*100)
```

```
ID             0.0
NAME           0.0
AGE           20.0
GENDER         0.0
CITY           0.0
SALARY        20.0
EXPERIENCE     0.0
dtype: float64
```

In [40]:

```
#check the status of null values in data set
print(df.isnull())
```

```
      ID   NAME    AGE  GENDER   CITY  SALARY  EXPERIENCE
0  False  False  False   False  False   False       False
1  False  False   True   False  False   False       False
2  False  False  False   False  False    True       False
3  False  False  False   False  False   False       False
4  False  False  False   False  False   False       False
5  False  False   True   False  False   False       False
6  False  False  False   False  False   False       False
7  False  False  False   False  False   False       False
8  False  False  False   False  False   False       False
9  False  False  False   False  False    True       False
```

In [41]:

```
#print total no of columns and rows in data set
print(df.shape)
#10 rows and 7 columns)
```

```
(10, 7)
```

In [44]:

```
#remove null values from the data set and print
print(df)
print()
df1= df.dropna()#remove null vaues in columns
print(df1)
```

```
   ID      NAME   AGE  GENDER         CITY     SALARY  EXPERIENCE
0   1      John  28.0    Male     New York    50000.0           3
1   2      Sara   NaN  Female  Los Angeles    52000.0           4
2   3     David  35.0    Male      Chicago        NaN           8
3   4     Linda  29.0  Female     New york    58000.0           5
4   5   Michael  41.0    Male        Miami    62000.0          10
5   6      Emma   NaN  Female  Los Angeles    54000.0           4
6   7    Robert  23.0    Male      CHICAGO    49000.0           2
7   8    Olivia  32.0  Female        Miami  1000000.0           7
8   9     James  27.0    Male     New York    51000.0           3
9  10  Patricia  38.0  Female  Los Angeles        NaN           9

   ID     NAME   AGE  GENDER      CITY     SALARY  EXPERIENCE
0   1     John  28.0    Male  New York    50000.0           3
3   4    Linda  29.0  Female  New york    58000.0           5
4   5  Michael  41.0    Male     Miami    62000.0          10
6   7   Robert  23.0    Male   CHICAGO    49000.0           2
7   8   Olivia  32.0  Female     Miami  1000000.0           7
8   9    James  27.0    Male  New York    51000.0           3
```

In [45]:

```
#copy dataframe to dataframe2
df2 = df
print(df)
print()
print(df2)
```

```
   ID      NAME   AGE  GENDER         CITY     SALARY  EXPERIENCE
0   1      John  28.0    Male     New York    50000.0           3
1   2      Sara   NaN  Female  Los Angeles    52000.0           4
2   3     David  35.0    Male      Chicago        NaN           8
3   4     Linda  29.0  Female     New york    58000.0           5
4   5   Michael  41.0    Male        Miami    62000.0          10
5   6      Emma   NaN  Female  Los Angeles    54000.0           4
6   7    Robert  23.0    Male      CHICAGO    49000.0           2
7   8    Olivia  32.0  Female        Miami  1000000.0           7
8   9     James  27.0    Male     New York    51000.0           3
9  10  Patricia  38.0  Female  Los Angeles        NaN           9

   ID      NAME   AGE  GENDER         CITY     SALARY  EXPERIENCE
0   1      John  28.0    Male     New York    50000.0           3
1   2      Sara   NaN  Female  Los Angeles    52000.0           4
2   3     David  35.0    Male      Chicago        NaN           8
3   4     Linda  29.0  Female     New york    58000.0           5
4   5   Michael  41.0    Male        Miami    62000.0          10
5   6      Emma   NaN  Female  Los Angeles    54000.0           4
6   7    Robert  23.0    Male      CHICAGO    49000.0           2
7   8    Olivia  32.0  Female        Miami  1000000.0           7
8   9     James  27.0    Male     New York    51000.0           3
9  10  Patricia  38.0  Female  Los Angeles        NaN           9
```

In [47]:

```
#data Cleaning
#Fill the missing values in the data set with 0
df2.fillna(0,inplace=True)# opr perform to original dataset without creating or returning a n
print(df2)
```

```
   ID      NAME   AGE  GENDER         CITY     SALARY  EXPERIENCE
0   1      John  28.0    Male     New York    50000.0           3
1   2      Sara   0.0  Female  Los Angeles    52000.0           4
2   3     David  35.0    Male      Chicago        0.0           8
3   4     Linda  29.0  Female     New york    58000.0           5
4   5   Michael  41.0    Male        Miami    62000.0          10
5   6      Emma   0.0  Female  Los Angeles    54000.0           4
6   7    Robert  23.0    Male      CHICAGO    49000.0           2
7   8    Olivia  32.0  Female        Miami  1000000.0           7
8   9     James  27.0    Male     New York    51000.0           3
9  10  Patricia  38.0  Female  Los Angeles        0.0           9
```

In [60]:

```
#Fill missing values with mean of the column
#df.fillna(df.mean())
```

```
#print(df)
df.fillna(df.mean(numeric_only=True), inplace=True)
print(df)
#This will work onlu if all columns are numeric
```

```
   ID      NAME   AGE  GENDER         CITY     SALARY  EXPERIENCE
4   5   Michael  41.0    Male        Miami    62000.0          10
9  10  Patricia  38.0  Female  Los Angeles        0.0           9
2   3     David  35.0    Male      Chicago        0.0           8
7   8    Olivia  32.0  Female        Miami  1000000.0           7
3   4     Linda  29.0  Female     New york    58000.0           5
0   1      John  28.0    Male     New York    50000.0           3
8   9     James  27.0    Male     New York    51000.0           3
6   7    Robert  23.0    Male      CHICAGO    49000.0           2
5   6      Emma   0.0  Female  Los Angeles    54000.0           4
1   2      Sara   0.0  Female  Los Angeles    52000.0           4
```

In [55]:

```
#Sort the values in the dataset based on age in ascending order
#df.sort_values("Age",inplace=True)
df.sort_values("AGE",inplace=True)
print(df)
```

```
   ID      NAME   AGE  GENDER         CITY     SALARY  EXPERIENCE
1   2      Sara   0.0  Female  Los Angeles    52000.0           4
5   6      Emma   0.0  Female  Los Angeles    54000.0           4
6   7    Robert  23.0    Male      CHICAGO    49000.0           2
8   9     James  27.0    Male     New York    51000.0           3
0   1      John  28.0    Male     New York    50000.0           3
3   4     Linda  29.0  Female     New york    58000.0           5
7   8    Olivia  32.0  Female        Miami  1000000.0           7
2   3     David  35.0    Male      Chicago        0.0           8
9  10  Patricia  38.0  Female  Los Angeles        0.0           9
4   5   Michael  41.0    Male        Miami    62000.0          10
```

In [58]:

```
#Sort values in data set in desc order
df.sort_values("AGE",ascending=False,inplace=True)
print(df)
```

```
   ID      NAME   AGE  GENDER         CITY     SALARY  EXPERIENCE
4   5   Michael  41.0    Male        Miami    62000.0          10
9  10  Patricia  38.0  Female  Los Angeles        0.0           9
2   3     David  35.0    Male      Chicago        0.0           8
7   8    Olivia  32.0  Female        Miami  1000000.0           7
3   4     Linda  29.0  Female     New york    58000.0           5
0   1      John  28.0    Male     New York    50000.0           3
8   9     James  27.0    Male     New York    51000.0           3
6   7    Robert  23.0    Male      CHICAGO    49000.0           2
5   6      Emma   0.0  Female  Los Angeles    54000.0           4
1   2      Sara   0.0  Female  Los Angeles    52000.0           4
```

In [59]:

```
#Find duplicated values in the data set
print(df.duplicated())
#Return ture for every row that is a duplicate , otherwise False
```

```
4    False
9    False
2    False
7    False
3    False
0    False
8    False
6    False
5    False
1    False
dtype: bool
```