In [130]:

```
import pandas as pd
import numpy as np
```

In [131]:

```
c_data = pd.read_excel("D:\KSR\Python\ExcelFile\Cust_Data.xlsx")
```

In [156]:

```
class DataCleaning:
    def __init__(self, data):
        self.data = data
    def volume_check(self):
        return self.data.shape
    def sample_data(self):
        return self.data.head(5)
    def row_count(self):
        print("The no of rows in given data:", self.data.shape[0])
    def column_count(self):
        print("The no of columns in given data:", data.shape[1])
    def column_list(self):
        return list(self.data.columns)
    def missing_values(self):
        return self.data.isna().sum()
    def missing_values_percent(self):
        return (self.data.isna().sum() / self.data.shape[0]) * 100
    def data_types(self):
        return self.data.dtypes
    def list_numerical_columns(self):
        return list(self.data.select_dtypes(include = np.number).columns)
    def list_categorical_columns(self):
        return list(self.data.select_dtypes(exclude = np.number).columns)
    def impute_missing_values_mean(self, column_name):
        self.data[column_name] = round(self.data[column_name].fillna(self.data[column_name]
        print("imputed sucessfully")
    def imput_missing_values_mode(self,col_name):
        self.data[col_name] = self.data[col_name].fillna(self.data[col_name].mode()[0])
        print("imputed sucessfully")
    def impute_missing_values_guest(self,col_name):
        self.data[col_name] = self.data[col_name].fillna("Guest/NA")
        print("imputed sucessfully")
    def Replace_with_NA(self,col_name):
        import pandas as pd
        self.data[col_name] = self.data[col_name].replace([" ", "NaN", "None", ""], pd.NA)
        print("NA got Replaced Successfully")
    def drop_column_NA(self,col_name):
        self.data = self.data.dropna(subset = [col_name])
```

```
        print("imputed sucessfully")
```

In [133]:

```
df = DataCleaning(c_data)
```

In [8]:

```
df.column_list()
```

Out[8]:

```
['CustID', 'CustName', 'CustLoc', 'CustAge']
```

In [9]:

```
df.row_count()
```

```
The no of rows in given data: 8
```

In [10]:

```
df.data
```

Out[10]:

	CustID	CustName	CustLoc	CustAge
0	1	Kiran	Pune	29
1	1	Kiran	Pune	29
2	2	Sathish	Maldives	30
3	3	Eshwar	Mumbai	34
4	5	Santhosh	Maldives	30
5	6	Mahesh	Tirupati	32
6	NaN	NaN	NaN	NaN
7	7	NaN	Bangalore	NaN

In [11]:

```
df.volume_check()
```

Out[11]:

```
(8, 4)
```

In [40]:

```
df.missing_values_percent()
```

Out[40]:

```
CustID      12.5
```

```
CustName    25.0
CustLoc     12.5
CustAge     25.0
dtype: float64
```

In [14]:

```
car_data = pd.read_csv("D:\KSR\Python\ExcelFile\Car_Data.csv")
```

In [96]:

```
df1 = DataCleaning(car_data)
```

In [16]:

```
df1.row_count()
```

```
The no of rows in given data: 301
```

In [17]:

```
df1.sample_data()
```

Out[17]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type
0	ritz	2014	3.35	5.59	27000	Petrol
1	sx4	2013	4.75	9.54	43000	Diesel
2	ciaz	2017	7.25	9.85	6900	Petrol
3	wagon r	2011	2.85	4.15	5200	Petrol
4	swift	2014	4.6	6.87	42450	Diesel

In [20]:

```
list(data.select_dtypes(include = np.number).columns)
```

Out[20]:

```
['CustID', 'CustAge']
```

In [21]:

```
list(data.select_dtypes(exclude = np.number).columns)
```

Out[21]:

```
['CustName', 'CustLoc']
```

In [25]:

```
df.list_numerical_columns()
```

Out[25]:

['CustID', 'CustAge']

In [26]:

df1.list_numerical_columns()

Out[26]:

['Year', 'Selling_Price', 'Present_Price', 'Kms_Driven']

In [30]:

df.data_types()

Out[30]:

```
CustID      float64
CustName     object
CustLoc      object
CustAge     float64
dtype: object
```

In [42]:

bike_data = pd.read_csv("D:\KSR\Python\ExcelFile\Bike_Data.csv")

In [97]:

bd = DataCleaning(bike_data)

In [91]:

bd.column_list()

Out[91]:

```
['Region',
 'Country',
 'Customer',
 'BusinessSegment',
 'Category',
 'Model',
 'Color',
 'SalesDate',
 'ListPrice',
 'UnitPrice',
```

'OrderQty']

In [44]:

```
bd.missing_values_percent()
```

Out[44]:

```
Region              0.000000
Country             0.000000
Customer            0.000000
BusinessSegment     0.000000
Category            0.000000
Model               0.000000
Color               8.578604
SalesDate           0.000000
ListPrice           0.000000
UnitPrice           0.000000
OrderQty            0.000000
dtype: float64
```

In [162]:

```
cust_data = pd.read_excel("D:\KSR\Python\ExcelFile\Cust_Data.xlsx")
```

In [163]:

```
cust_data.head(10)
```

Out[163]:

	CustID	CustName	CustLoc	CustAge
0	1	Kiran	Pune	29
1	1	Kiran	Pune	29
2	2	Sathish	Maldives	30
3	3	Eshwar	Mumbai	34
4	5	Santhosh	Maldives	30
5	6	Mahesh	Tirupati	32
6	NaN	NaN	NaN	NaN
7	7	NaN	Bangalore	NaN

In [164]:

```
cd = DataCleaning(cust_data)
```

In [165]:

```
cd.impute_missing_values_guest('CustName')
```

```
imputed sucessfully
```

In [166]:

```
cd.impute_missing_values_mean('CustAge')
```

```
imputed sucessfully
```

In [167]:

```
cd.imput_missing_values_mode('CustLoc')
```

```
imputed sucessfully
```

In [168]:

```
cust_data.head(10)
```

Out[168]:

	CustID	CustName	CustLoc	CustAge
0	1	Kiran	Pune	29
1	1	Kiran	Pune	29
2	2	Sathish	Maldives	30
3	3	Eshwar	Mumbai	34
4	5	Santhosh	Maldives	30
5	6	Mahesh	Tirupati	32
6	NaN	Guest/NA	Maldives	31
7	7	Guest/NA	Bangalore	31

In [169]:

```
cd.Replace_with_NA('CustID')
```

```
NA got Replaced Successfully
```

In [175]:

```
cd.data.head(10)
```

Out[175]:

	CustID	CustName	CustLoc	CustAge
0	1	Kiran	Pune	29
1	1	Kiran	Pune	29
2	2	Sathish	Maldives	30
3	3	Eshwar	Mumbai	34
4	5	Santhosh	Maldives	30
5	6	Mahesh	Tirupati	32
7	7	Guest/NA	Bangalore	31

In [171]:

```
cd.drop_column_NA('CustID')
```

imputed sucessfully

In [174]:

cd.data.head(10)

Out[174]:

	CustID	CustName	CustLoc	CustAge
0	1	Kiran	Pune	29
1	1	Kiran	Pune	29
2	2	Sathish	Maldives	30
3	3	Eshwar	Mumbai	34
4	5	Santhosh	Maldives	30
5	6	Mahesh	Tirupati	32
7	7	Guest/NA	Bangalore	31

In [ ]:

```
.mean()),0)
```

Seller_Type	Transmission	Car_Owner
Dealer	Manual	1st Owner
Dealer	Manual	1st Owner
Dealer	Manual	1st Owner
Dealer	Manual	1st Owner
Dealer	Manual	2nd Owner