1) Data Wrangling

Report on data wrangling step.

Firstly, we came up with the dataset from "Kaggle" site and chose it that accept cleaning.

The dataset name is bike_buyers . it’s about some information about people who buy bikes like age, marital status, gender, income , number of children they have ,their Education , Occupation , and number of cars they have.

We found some quality and some tidiness issues , so we cleaned the dataset.

But before we clean, we copied the dataset.

```
In [6]: bike_clean = dataset.copy()
```

Quality issues:

We found some quality issues like :

- change ID type from integer to string to be more accurate because we don’t deal with ID as a number .

```
In [7]: #change ID type from int to string
        bike_clean.ID = bike_clean.ID.astype(str)
```

- change cars type from float to integer because it has no meaning to have 1.5 car it will cause inaccurate data .

```
In [9]: #change Cars type from float to int
        bike_clean.Cars = bike_clean.Cars.astype(int)
```

- change income type from float to integer to be more obvious and easy to read .

```
In [11]:
#change Income type from float to int
bike_clean.Income = bike_clean.Income.astype(int)
```

- change children type from float to integer because also like cars it has no meaning to have 1.5 children and also make it easy to read .

```
In [13]:
#change Children type from float to int
bike_clean.Children = bike_clean.Children.astype(int)
```

- change gender type from string to categorical to be easy to classify or clustering .

```
In [16]:
#change Gender type to catagorical
bike_clean.Gender = bike_clean.Gender.astype('category')
```

- change age type from float to integer to be easier to read and understand.

```
In [19]:
#change age type from float to int
bike_clean.Age = bike_clean.Age.astype(int)
```

Tidiness issue :

Also, in our project we have a little tidiness issue like:

- NAN values in cars, income, children, homeowner, gender, age, and marital status.
- And we replaced these missing values with the mode in homeowner, gender, and marital status.
- And the other (cars , income , children , age) we replaced it’s missing values with the median .

In [14]:
```
#replace missing Home Owner values with the mode
bike_clean['Home Owner'] = bike_clean['Home Owner'].fillna(bike_clean['Home Owner'].mode()[0])
```

In [15]:
```
#replace missing Gender values with the mode
bike_clean.Gender = bike_clean.Gender.fillna(bike_clean.Gender.mode()[0])
```

In [17]:
```
#replace missing Marital Status values with the mode
bike_clean['Marital Status'] = bike_clean['Marital Status'].fillna(bike_clean['Marital Status'].mode()[0])
```

In [18]:
```
#replace missing Age values with median
bike_clean.Age = bike_clean.Age.fillna(bike_clean.Age.median())
```

In [12]:
```
#replace missing Children values with median
bike_clean.Children = bike_clean.Children.fillna(bike_clean.Children.median())
```

In [10]:
```
#replace missing Income values with median
bike_clean.Income = bike_clean.Income.fillna(bike_clean.Income.median())
```

```
In [8]:  #replace missing cars values with median
         bike_clean.Cars = bike_clean.Cars.fillna(bike_clean.Cars.median())
```

we handle these issues and clean data using "pandas" and "numpy" libraries. these used for manipulating and analyzing

data .

Numpy:

we use "numpy" because it has important attributes, we can know the size of each column, dimensions, data types, and so on

Pandas:

we use "pandas " in this stage because of it's features

Some commonly used data structures in pandas are:

Series objects: 1D array, similar to a column in a dataset

DataFrame objects: 2D table, similar to a data set Panel objects: Dictionary of DataFrames