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

In [2]:
```
df = pd.read_csv('/Users/sameerkhan/Documents/DATA CLEANING/Automobile price data _Raw_(before).csv')
```

In [3]:
```
df
```

```
  ?   alfa-     gas     std   two  convertible  rwd  front   88.6  ...  130  mpfi  3.47  2.68   9.0  111  5000  21
      romero
  ?   alfa-     gas     std   two  convertible  rwd  front   88.6  ...  130  mpfi  3.47  2.68   9.0  111  5000  21
      romero
  ?   alfa-     gas     std   two    hatchback  rwd  front   94.5  ...  152  mpfi  2.68  3.47   9.0  154  5000  19
      romero
164    audi     gas     std  four        sedan  fwd  front   99.8  ...  109  mpfi  3.19  3.40  10.0  102  5500  24
164    audi     gas     std  four        sedan  4wd  front   99.4  ...  136  mpfi  3.19  3.40   8.0  115  5500  18
...     ...     ...     ...   ...          ...  ...    ...    ...  ...  ...   ...   ...   ...   ...  ...   ...  ...
 95   volvo     gas     std  four        sedan  rwd  front  109.1  ...  141  mpfi  3.78  3.15   9.5  114  5400  23
 95   volvo     gas   turbo  four        sedan  rwd  front  109.1  ...  141  mpfi  3.78  3.15   8.7  160  5300  19
 95   volvo     gas     std  four        sedan  rwd  front  109.1  ...  173  mpfi  3.58  2.87   8.8  134  5500  18
 95   volvo  diesel   turbo  four        sedan  rwd  front  109.1  ...  145   idi  3.01  3.40  23.0  106  4800  26
```

In [50]:
```
df.shape
```

Out[50]:
```
(205, 26)
```

In [4]:
```
'''
1. Check for errors in datasets
2. Replace them with a suitable approach and explain why you opted for that method or technique.
3. Final report on the dataset before and after cleaning.
4. The final dataset needs to be saved to .csv format.
'''
df.describe()
```

Out[4]:

	symboling	wheel-base	length	width	height	curb-weight	engine-size	compression-ratio	city-mpg	highway-mpg
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000
mean	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317	10.142537	25.219512	30.751220
std	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693	3.972040	6.542142	6.886443
min	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	7.000000	13.000000	16.000000
25%	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000	8.600000	19.000000	25.000000
50%	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	9.000000	24.000000	30.000000
75%	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000	9.400000	30.000000	34.000000
max	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000	23.000000	49.000000	54.000000

In [ ]:
```
# cleaning 4wd errors and checking unique values to catogorical coumns
```

In [5]:
```
import re

df.loc[df['drive-wheels']=='4wd','drive-wheels'] = 'fwd'
```

```
In [6]: df
```

Out[6]:

ıake	fuel- type	aspiration	num- of- doors	body- style	drive- wheels	engine- location	wheel- base	...	engine- size	fuel- system	bore	stroke	compression- ratio	horsepower	peak- rpm	city mpı
alfa- nero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0	111	5000	2
alfa- nero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0	111	5000	2
alfa- nero	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68	3.47	9.0	154	5000	1
audi	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19	3.40	10.0	102	5500	2
audi	gas	std	four	sedan	fwd	front	99.4	...	136	mpfi	3.19	3.40	8.0	115	5500	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
ıolvo	gas	std	four	sedan	rwd	front	109.1	...	141	mpfi	3.78	3.15	9.5	114	5400	2
ıolvo	gas	turbo	four	sedan	rwd	front	109.1	...	141	mpfi	3.78	3.15	8.7	160	5300	1
ıolvo	gas	std	four	sedan	rwd	front	109.1	...	173	mpfi	3.58	2.87	8.8	134	5500	1
ıolvo	diesel	turbo	four	sedan	rwd	front	109.1	...	145	idi	3.01	3.40	23.0	106	4800	2
ıolvo	gas	turbo	four	sedan	rwd	front	109.1	...	141	mpfi	3.78	3.15	9.5	114	5400	1

```
In [10]: df['normalized-losses'].unique()

Out[10]: array(['?', '164', '158', '192', '188', '121', '98', '81', '118', '148',
                '110', '145', '137', '101', '78', '106', '85', '107', '104', '113',
                '150', '129', '115', '93', '142', '161', '153', '125', '128',
                '122', '103', '168', '108', '194', '231', '119', '154', '74',
                '186', '83', '102', '89', '87', '77', '91', '134', '65', '197',
                '90', '94', '256', '95'], dtype=object)

In [11]: df['num-of-doors'].unique()

Out[11]: array(['two', 'four', '?'], dtype=object)

In [12]: df['bore'].unique()

Out[12]: array(['3.47', '2.68', '3.19', '3.13', '3.50', '3.31', '3.62', '2.91',
                '3.03', '2.97', '3.34', '3.60', '2.92', '3.15', '3.43', '3.63',
                '3.54', '3.08', '?', '3.39', '3.76', '3.58', '3.46', '3.80',
                '3.78', '3.17', '3.35', '3.59', '2.99', '3.33', '3.70', '3.61',
                '3.94', '3.74', '2.54', '3.05', '3.27', '3.24', '3.01'],
               dtype=object)

In [13]: df['stroke'].unique()

Out[13]: array(['2.68', '3.47', '3.40', '2.80', '3.19', '3.39', '3.03', '3.11',
                '3.23', '3.46', '3.90', '3.41', '3.07', '3.58', '4.17', '2.76',
                '3.15', '?', '3.16', '3.64', '3.10', '3.35', '3.12', '3.86',
                '3.29', '3.27', '3.52', '2.19', '3.21', '2.90', '2.07', '2.36',
                '2.64', '3.08', '3.50', '3.54', '2.87'], dtype=object)

In [14]: df['horsepower'].unique()

Out[14]: array(['111', '154', '102', '115', '110', '140', '160', '101', '121',
                '182', '48', '70', '68', '88', '145', '58', '76', '60', '86',
                '100', '78', '90', '176', '262', '135', '84', '64', '120', '72',
                '123', '155', '184', '175', '116', '69', '55', '97', '152', '200',
                '95', '142', '143', '207', '288', '?', '73', '82', '94', '62',
                '56', '112', '92', '161', '156', '52', '85', '114', '162', '134',
                '106'], dtype=object)

In [15]: df['peak-rpm'].unique()

Out[15]: array(['5000', '5500', '5800', '4250', '5400', '5100', '4800', '6000',
                '4750', '4650', '4200', '4350', '4500', '5200', '4150', '5600',
                '5900', '5750', '?', '5250', '4900', '4400', '6600', '5300'],
               dtype=object)
```

```
In [22]: type(df['price'].value_counts())

Out[22]: pandas.core.series.Series
```

```
In [24]: df['stroke'] = pd.to_numeric(df['stroke'], errors = 'coerce')
         #converting it to numeric values from nan
```

```
In [27]: df[['stroke']].mode()
         #checking mode as its categorical column cant use mean to fill null values
         # as it will disturb the data
         df[['stroke']].mode().iloc[0]

Out[27]: stroke    3.4
         Name: 0, dtype: float64
```

```
In [28]: #replacing nan values with mode of column
         df['stroke'].replace(to_replace=np.nan,value=df['stroke'].mode()[0],inplace=True)
```

```
In [30]: df[['stroke']].isnull().sum()

Out[30]: stroke    0
         dtype: int64
```

```
In [31]: #same with other
         df['normalized-losses'] = pd.to_numeric(df['normalized-losses'], errors = 'coerce')
```

```
In [34]: df['normalized-losses'].mode().iloc[0]

Out[34]: 161.0
```

```
In [35]: df['normalized-losses'].replace(to_replace=np.nan,value=df['normalized-losses'].mode()[0],inplace=True)
```

```
In [36]: df['bore'] = pd.to_numeric(df['bore'], errors = 'coerce')
```

```
In [38]: df['bore'].mode().iloc[0]

Out[38]: 3.62
```

```
In [39]: df['bore'].replace(to_replace=np.nan,value=df['bore'].mode()[0],inplace=True)
```

```
In [43]: df['horsepower'].mode()

Out[43]: 0    68.0
         dtype: float64
```

```
In [41]: df['horsepower'] = pd.to_numeric(df['horsepower'], errors = 'coerce')
```

```
In [42]: df['horsepower'].replace(to_replace=np.nan,value=df['horsepower'].mode()[0],inplace=True)
```

```
In [44]: df['peak-rpm'] = pd.to_numeric(df['peak-rpm'], errors = 'coerce')
```

```
In [46]: df['peak-rpm'].mode()

Out[46]: 0    5500.0
         dtype: float64
```

```
In [47]: df['peak-rpm'].replace(to_replace=np.nan,value=df['peak-rpm'].mode()[0],inplace=True)
```

```
In [48]: df['num-of-doors'].mode()

Out[48]: 0    four
         dtype: object
```

```
In [49]: df['num-of-doors'].replace(to_replace='?',value=df['num-of-doors'].mode()[0],inplace=True)
```

In [59]:
```
df.info()
df.shape
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   symboling          205 non-null    int64  
 1   normalized-losses  205 non-null    float64
 2   make               205 non-null    object 
 3   fuel-type          205 non-null    object 
 4   aspiration         205 non-null    object 
 5   num-of-doors       205 non-null    object 
 6   body-style         205 non-null    object 
 7   drive-wheels       205 non-null    object 
 8   engine-location    205 non-null    object 
 9   wheel-base         205 non-null    float64
 10  length             205 non-null    float64
 11  width              205 non-null    float64
 12  height             205 non-null    float64
 13  curb-weight        205 non-null    int64  
 14  engine-type        205 non-null    object 
```

In [63]:
```
df.isna().sum()
```

```
body-style           0
drive-wheels         0
engine-location      0
wheel-base           0
length               0
width                0
height               0
curb-weight          0
engine-type          0
num-of-cylinders     0
engine-size          0
fuel-system          0
bore                 0
stroke               0
compression-ratio    0
horsepower           0
peak-rpm             0
city-mpg             0
highway-mpg          0
price                0
dtype: int64
```

In [66]:
```
df.to_csv('/Users/sameerkhan/Documents/DATA CLEANING/Auto_mobile_dataset_cleaned1.csv')
```

In [68]:
```
df['price'].unique()
```

```
       '12945', '10345', '6785', '11048', '32250', '35550', '36000',
       '5195', '6095', '6795', '6695', '7395', '10945', '11845', '13645',
       '15645', '8495', '10595', '10245', '10795', '11245', '18280',
       '18344', '25552', '28248', '28176', '31600', '34184', '35056',
       '40960', '45400', '16503', '5389', '6189', '6669', '7689', '9959',
       '8499', '12629', '14869', '14489', '6989', '8189', '9279', '5499',
       '7099', '6649', '6849', '7349', '7299', '7799', '7499', '7999',
       '8249', '8949', '9549', '13499', '14399', '17199', '19699',
       '18399', '11900', '13200', '12440', '13860', '15580', '16900',
       '16695', '17075', '16630', '17950', '18150', '12764', '22018',
       '32528', '34028', '37028', '9295', '9895', '11850', '12170',
       '15040', '15510', '18620', '5118', '7053', '7603', '7126', '7775',
       '9960', '9233', '11259', '7463', '10198', '8013', '11694', '5348',
       '6338', '6488', '6918', '7898', '8778', '6938', '7198', '7788',
       '7738', '8358', '9258', '8058', '8238', '9298', '9538', '8449',
       '9639', '9989', '11199', '11549', '17669', '8948', '10698', '9988',
       '10898', '11248', '16558', '15998', '15690', '15750', '7975',
       '7995', '8195', '9495', '9995', '11595', '9980', '13295', '13845',
       '12290', '12940', '13415', '15985', '16515', '18420', '18950',
       '16845', '19045', '21485', '22470', '22625'], dtype=object)
```

In [72]:
```
df['price'] = pd.to_numeric(df['price'], errors = 'coerce')
```

In [76]:
```
df['price'].mean()
```

Out[76]:
```
13207.129353233831
```

```
In [77]: df.isna().sum()
```

```
Out[77]: symboling            0
         normalized-losses    0
         make                 0
         fuel-type            0
         aspiration           0
         num-of-doors         0
         body-style           0
         drive-wheels         0
         engine-location      0
         wheel-base           0
         length               0
         width                0
         height               0
         curb-weight          0
         engine-type          0
         num-of-cylinders     0
         engine-size          0
         fuel-system          0
         bore                 0
```

```
In [78]: df['price'].isna().sum()
```

```
Out[78]: 4
```

```
In [79]: df['price'].replace(to_replace=np.nan,value=df['price'].mean(),inplace=True)
```

```
In [80]: df['price'].isna().sum()
```

```
Out[80]: 0
```

```
In [81]: df.to_csv('/Users/sameerkhan/Documents/DATA CLEANING/Auto_mobile_dataset_cleaned_final.csv')
```

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

```
Out[82]: (205, 26)
```

```
In [ ]:
```