```
#importing necessary packages
import numpy as np
```

```
import pandas as pd
```

```
#reading a tabular data as data frame
cars=pd.read_csv('E:\sweetlin-official\Sweetlin_2020\Folder D\Personal\Learning\dataset\Toyota.csv')
```

```
#to display the features in the data set
cars.columns
```

```
Index(['Unnamed: 0', 'Price', 'Age', 'KM', 'FuelType', 'HP', 'MetColor',
       'Automatic', 'CC', 'Doors', 'Weight'],
      dtype='object')
```

```
#print the value in the data set
cars
```

	Unnamed: 0	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	0	13500	23.0	46986	Diesel	90	1.0	0	2000	three	1165
1	1	13750	23.0	72937	Diesel	90	1.0	0	2000	3	1165
2	2	13950	24.0	41711	Diesel	90	NaN	0	2000	3	1165
3	3	14950	26.0	48000	Diesel	90	0.0	0	2000	3	1165
4	4	13750	30.0	38500	Diesel	90	0.0	0	2000	3	1170
...	...	...	...	...	...	...	...	...	...	...	...
1431	1431	7500	NaN	20544	Petrol	86	1.0	0	1300	3	1025
1432	1432	10845	72.0	??	Petrol	86	0.0	0	1300	3	1015
1433	1433	8500	NaN	17016	Petrol	86	0.0	0	1300	3	1015
1434	1434	7250	70.0	??	NaN	86	1.0	0	1300	3	1015
1435	1435	6950	76.0	1	Petrol	110	0.0	0	1600	5	1114

1436 rows × 11 columns

```
#to get the dimension of the data set
cars.shape
```

```
(1436, 11)
```

```
#to get the information of the data set
cars.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1436 entries, 0 to 1435
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  1436 non-null   int64
 1   Price       1436 non-null   int64
 2   Age        1336 non-null   float64
 3   KM         1436 non-null   object
 4   FuelType   1336 non-null   object
 5   HP         1436 non-null   object
 6   MetColor   1286 non-null   float64
 7   Automatic  1436 non-null   int64
 8   CC         1436 non-null   int64
 9   Doors      1436 non-null   object
 10  Weight     1436 non-null   int64
dtypes: float64(2), int64(5), object(4)
memory usage: 123.5+ KB
```

```
#to get the frequency count of unique values of a column in the data frame
cars['KM'].value_counts()
```

```
??       15
1         8
36000     7
59000     7
43000     7
         ..
27301     1
63135     1
98500     1
78785     1
32000     1
Name: KM, Length: 1256, dtype: int64
```

```
#to display top 2 records of a column in the data frame
cars['KM'].head(2)
```

```
0    46986
1    72937
Name: KM, dtype: object
```

```
#to display the unique values of a column in the data frame
cars['HP'].unique()
```

```
array(['90', '????', '192', '110', '97', '71', '116', '98', '69', '86',
       '72', '107', '73'], dtype=object)
```

```
#reading the file with a few more parameters besides the file name to get a clean data
cars=pd.read_csv('E:\sweetlin-official\Sweetlin_2020\Folder D\Personal\Learning\dataset\Toyota.csv',
```

```
#to take a copy of a data frame
car2=cars.copy()
```

```
car2.columns
```

```
Index(['Price', 'Age', 'KM', 'FuelType', 'HP', 'MetColor', 'Automatic', 'CC',
       'Doors', 'Weight'],
      dtype='object')
```

```
car2.shape
```

```
car2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1436 entries, 0 to 1435
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Price      1436 non-null   int64  
 1   Age        1336 non-null   float64
 2   KM         1421 non-null   float64
 3   FuelType   1336 non-null   object 
 4   HP         1430 non-null   float64
 5   MetColor   1286 non-null   float64
 6   Automatic  1436 non-null   int64  
 7   CC         1436 non-null   int64  
 8   Doors      1436 non-null   object 
 9   Weight     1436 non-null   int64  
dtypes: float64(4), int64(4), object(2)
memory usage: 123.4+ KB
```

```
#checking for the presence of missingness
car2.isnull().sum()
```

```
Price          0
Age          100
KM            15
FuelType     100
HP             6
MetColor     150
Automatic      0
CC             0
Doors          0
Weight         0
dtype: int64
```

```
#summary statistics
car2.describe()
```

	Price	Age	KM	HP	MetColor	Automatic	
count	1436.000000	1336.000000	1421.000000	1430.000000	1286.000000	1436.000000	1436.0000
mean	10730.824513	55.672156	68647.239972	101.478322	0.674961	0.055710	1566.8279
std	3626.964585	18.589804	37333.023589	14.768255	0.468572	0.229441	187.1824
min	4350.000000	1.000000	1.000000	69.000000	0.000000	0.000000	1300.0000
25%	8450.000000	43.000000	43210.000000	90.000000	0.000000	0.000000	1400.0000
50%	9900.000000	60.000000	63634.000000	110.000000	1.000000	0.000000	1600.0000
75%	11950.000000	70.000000	87000.000000	110.000000	1.000000	0.000000	1600.0000
max	32500.000000	80.000000	243000.000000	192.000000	1.000000	1.000000	2000.0000

Central Tendency Measures

```
#to compute mean of a column in the data frame
cars['Age'].mean()
```

```
55.67215568862275
```

```
#to compute median of a column in the data frame
cars['Age'].median()
```

```
60.0
```

```
#to compute mode of a column in the data frame
cars['Age'].mode()
```

```
0    65.0
dtype: float64
```

```
#to compute the quantile of a column in the data frame. check for quantile (0), quantile(1), quantil
cars['Age'].quantile()
```

```
60.0
```

Measures of dispersion

```
#to compute variance of a column in the data frame
cars['Weight'].var()
```

```
2771.0875661196496
```

```
#to compute standard deviation of a column in the data frame
cars['Weight'].std()
```

```
52.6411204869316
```

```
#correlation -to check whether linear relationship exists between 2 variables
cars['Price'].corr(cars['Age'])
```

```
-0.8784074093622005
```

```
cars['Age'].value_counts()
```

```
65.0    62
68.0    60
80.0    52
62.0    41
78.0    41
        ..
12.0     2
10.0     1
6.0      1
18.0     1
2.0      1
Name: Age, Length: 77, dtype: int64
```

```
pd.value_counts(cars['FuelType'])
```

```
Petrol    1177
Diesel     144
CNG         15
Name: FuelType, dtype: int64
```

```
#To create frequency table
pd.crosstab(index=cars['FuelType'],columns='count',dropna=True)
```

col_0	count
FuelType	
CNG	15
Diesel	144
Petrol	1177

```
#To create two-way table
pd.crosstab(index=cars['Automatic'],columns=cars['FuelType'],dropna=True)
```

FuelType	CNG	Diesel	Petrol
Automatic			
0	15	144	1104
1	0	0	73

```
#Two-way table -Joint probability
pd.crosstab(index=cars['Automatic'],columns=cars['FuelType'],normalize=True,dropna=True)
```

FuelType	CNG	Diesel	Petrol
Automatic			
0	0.011228	0.107784	0.826347
1	0.000000	0.000000	0.054641

```
#Two-way table -Margin probability
pd.crosstab(index=cars['Automatic'],columns=cars['FuelType'],normalize=True,margins=True,dropna=True
```

FuelType	CNG	Diesel	Petrol	All
Automatic				
0	0.011228	0.107784	0.826347	0.945359
1	0.000000	0.000000	0.054641	0.054641
All	0.011228	0.107784	0.880988	1.000000

```
#Two-way table -conditional probability
pd.crosstab(index=cars['Automatic'],columns=cars['FuelType'],normalize='index',margins=True,dropna=T
```

FuelType	CNG	Diesel	Petrol
Automatic			
0	0.011876	0.114014	0.874109
1	0.000000	0.000000	1.000000
All	0.011228	0.107784	0.880988

```
#Two-way table -conditional probability
pd.crosstab(index=cars['Automatic'],columns=cars['FuelType'],normalize='columns',margins=True,dropna
```

FuelType	CNG	Diesel	Petrol	All
Automatic				
0	1.0	1.0	0.937978	0.945359
1	0.0	0.0	0.062022	0.054641

```
#correlation - to consider the columns having only numerical values
num_data=cars.select_dtypes(exclude=[object])
```

```
num_data.columns
```

```
Index(['Price', 'Age', 'KM', 'HP', 'MetColor', 'Automatic', 'CC', 'Weight'], dtype='object')
```

```
#correlation matrix
corr_val=num_data.corr()
corr_val
```

	Price	Age	KM	HP	MetColor	Automatic	CC	Weight
Price	1.000000	-0.878407	-0.574720	0.309902	0.112041	0.033081	0.165067	0.581198
Age	-0.878407	1.000000	0.512735	-0.157904	-0.099659	0.032573	-0.120706	-0.464299
KM	-0.574720	0.512735	1.000000	-0.335285	-0.093825	-0.081248	0.299993	-0.026271
HP	0.309902	-0.157904	-0.335285	1.000000	0.064749	0.013755	0.053758	0.086737
MetColor	0.112041	-0.099659	-0.093825	0.064749	1.000000	-0.013973	0.029189	0.057142
Automatic	0.033081	0.032573	-0.081248	0.013755	-0.013973	1.000000	-0.069321	0.057249
CC	0.165067	-0.120706	0.299993	0.053758	0.029189	-0.069321	1.000000	0.651450
Weight	0.581198	-0.464299	-0.026271	0.086737	0.057142	0.057249	0.651450	1.000000