```
1 #import pandas library
2 import pandas as pd
```

```
1 #import carsdataset to a variable cds using pandas library
2 cds = pd.read_csv("/content/CarsData.csv")
```

```
1 cds  #prints complete dataset
```

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylin
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	
...	...	...	...	...	...	...	...	...	
427	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	
428	Volvo	C70 HPT convertible 2dr	Sedan	Europe	Front	$42,565	$40,083	2.3	
429	Volvo	S80 T6	Sedan	Europe	Front	$45,210	$42,573	2.9	

```
1 cds.head() #prints first 5 rows of the dataset
```

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	

```
1 cds.tail() #prints last 5 rows of the dataset
```

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylin
427	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	

```
1 cds.shape #returns the number of rows and columns of the dataset
```

```
(432, 15)
```

Data Cleaning

```
1 # isnull function is used to check the null values present in the dataset
2 # here false means no null value is present here
3 cds.isnull()
```

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	H
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	
427	False	False	False	False	False	False	False	False	False	
428	False	False	False	False	False	False	False	False	False	
429	False	False	False	False	False	False	False	False	False	
430	False	False	False	False	False	False	False	False	False	
431	False	False	False	False	False	False	False	False	False	

432 rows × 15 columns

```
1 #sum() function used to find all null values count based on the column - in
2 cds.isnull().sum()
```

```
Make          4
Model         4
Type          4
Origin        4
DriveTrain    4
MSRP          4
Invoice       4
EngineSize    4
Cylinders     6
Horsepower    4
MPG_City      4
```

```
MPG_Highway    4
Weight         4
Wheelbase      4
Length         4
dtype: int64
```

fill null values of each column with mean value of that column or with a default value

```
cds['Make'].fillna('Acura',inplace=True)
cds['Model'].fillna('C70 LPT convertible 2dr',inplace=True)
cds['Type'].fillna('Sedan',inplace=True)
cds['Origin'].fillna('Europe',inplace=True)
cds['DriveTrain'].fillna('Front',inplace=True)
cds['MSRP'].fillna('$40,565',inplace=True)
cds['Invoice'].fillna('$40,083',inplace=True)
# fillna() func used to fill the null values with given values in the entire
# Here Make,Model,Type,Origin,DriveTrain,MSRP,Invoice is str so we cannot fi
# here inplace=True makes the permanent changes of the null values to the gi
```

```
cds['EngineSize'].fillna(cds['EngineSize'].mean(),inplace=True)
cds['Cylinders'].fillna(cds['Cylinders'].mean(),inplace=True)
cds['Horsepower'].fillna(cds['Horsepower'].mean(),inplace=True)
cds['MPG_City'].fillna(cds['MPG_City'].mean(),inplace=True)
cds['MPG_Highway'].fillna(cds['MPG_Highway'].mean(),inplace=True)
cds['Weight'].fillna(cds['Weight'].mean(),inplace=True)
cds['Wheelbase'].fillna(cds['Wheelbase'].mean(),inplace=True)
cds['Length'].fillna(cds['Length'].mean(),inplace=True)
# this all are int values so we can find mean and replace the null values
```

```
cds.isnull().sum()
# Now the data has no null values
```

```
Make           0
Model          0
Type           0
Origin         0
DriveTrain     0
MSRP           0
Invoice        0
EngineSize     0
Cylinders      0
Horsepower     0
MPG_City       0
MPG_Highway    0
Weight         0
Wheelbase      0
Length         0
dtype: int64
```

Value Count function

this function returns the count of each value present in the entire column of the dataset.

```
cds['Make'].value_counts()
# values with there frequencies in the Make column
# Similarly we can find for remaining columns also
```

```
Toyota           28
Chevrolet        27
Mercedes-Benz    26
Ford             23
BMW              20
Audi             19
Honda            17
Nissan           17
Volkswagen       15
Chrysler         15
Dodge            13
Mitsubishi       13
Volvo            12
Jaguar           12
Hyundai          12
Subaru           11
Pontiac          11
Mazda            11
Acura            11
Lexus            11
Kia              11
Buick             9
Mercury           9
Lincoln           9
Saturn            8
Cadillac          8
Suzuki            8
Infiniti          8
GMC               8
Porsche           7
Saab              7
Land Rover        3
Oldsmobile        3
Jeep              3
Scion             2
Isuzu             2
MINI              2
Hummer            1
Name: Make, dtype: int64
```

```
cds['Origin'].value_counts()
```

```
Asia      158
USA       147
Europe    127
Name: Origin, dtype: int64
```

Instruction (Filtering)

isin() function - return the records which it has the given value names in it.

```
1 cds[cds['Make'].isin(['Jeep','Hummer'])]
```

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cyli
169	Hummer	H2	SUV	USA	All	$49,995	$45,815	6.0	
205	Jeep	Grand Cherokee Laredo	SUV	USA	Front	$27,905	$25,686	4.0	
		Liberty							

```
1 cds[cds['Origin'].isin(['USA'])]
```

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cyl
48	Buick	Rainier	SUV	USA	All	$37,895	$34,357	4.2	
49	Buick	Rendezvous CX	SUV	USA	Front	$26,545	$24,085	3.4	
50	Buick	Century Custom 4dr	Sedan	USA	Front	$22,180	$20,351	3.1	
51	Buick	LeSabre Custom 4dr	Sedan	USA	Front	$26,470	$24,282	3.8	
52	Buick	Regal LS 4dr	Sedan	USA	Front	$24,895	$22,835	3.8	
...	...	...	...	...	...	...	...	...	
351	Saturn	lon3 4dr	Sedan	USA	Front	$15,825	$14,811	2.2	
352	Saturn	lon2 quad coupe 2dr	Sedan	USA	Front	$14,850	$13,904	2.2	
353	Saturn	lon3 quad	Sedan	USA	Front	$16,350	$15,299	2.2	

Instruction (Remove unwanded records)

- Remove records where weight is above 4000

```
1 # records with weight below 4000
2 cds[cds['Weight']<4000]
```

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cyl
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	
5	Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	$46,100	$41,100	3.5	
...	...	...	...	...	...	...	...	...	
427	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	
		C70 HPT							

```
1 # records with weight above 4000
2 cds[ ~(cds['Weight']<4000) ]  # we can use negation or <
```

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5
15	Audi	A4 3.0 Quattro convertible 2dr	Sedan	Europe	All	$44,240	$40,075	3.0
17	Audi	A6 4.2 Quattro 4dr	Sedan	Europe	All	$49,690	$44,936	4.2
18	Audi	A8 L Quattro 4dr	Sedan	Europe	All	$69,190	$64,740	4.2
20	Audi	RS 6 4dr	Sports	Europe	Front	$84,600	$76,417	4.2
...	...	...	...	...	...	...	...	...
405	Volkswagen	Touareg V6	SUV	Europe	All	$35,515	$32,243	3.2
415	Volkswagen	Phaeton	Sedan	Europe	Front	$65,000	$59,912	4.2

Instruction (Apply function on a column)

- Increase values of MPG_City by 3

apply() function - it is used to apply a function or a operation on a column of the dataset

```
1 cds['MPG_City'] = cds['MPG_City'].apply(lambda x:x+3)
```

```
2 # lambda used for mathematical functions or operations
```

```
1 cds
```

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylin
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	
...	...	...	...	...	...	...	...	...	
427	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	
428	Volvo	C70 HPT convertible 2dr	Sedan	Europe	Front	$42,565	$40,083	2.3	
429	Volvo	S80 T6 4dr	Sedan	Europe	Front	$45,210	$42,573	2.9	

✓ 0s completed at 8:23 PM