Working on Real Time Project Using Python

(A Part of Big Data Analysis)

Cars Data Set

Here the data of different car is given with their specififcation.

The data is available as a CSV file. We will analyze this data using Pandas DataFrame.

The commands that we used in this project :

import pandas as pd

To import Pandas library

In [4]:
```
import pandas as pd
```

pd.read_csv

To import the CSV file in Jupyter notebook

In [5]:
```
data = pd.read_csv("Cars_Data.csv")
```

In [6]:
```
data
```

Out[6]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepowe
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265.(
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200.(
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	200.(
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6.0	270.(
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6.0	225.(
...	...	...	...	...	...	...	...	...	...	.
427	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	5.0	197.(
428	Volvo	C70 HPT convertible 2dr	Sedan	Europe	Front	$42,565	$40,083	2.3	5.0	242.(
429	Volvo	S80 T6 4dr	Sedan	Europe	Front	$45,210	$42,573	2.9	6.0	268.(
430	Volvo	V40	Wagon	Europe	Front	$26,135	$24,641	1.9	4.0	170.(
431	Volvo	XC70	Wagon	Europe	All	$35,145	$33,112	2.5	5.0	208.(

432 rows × 15 columns

.head()

It shows the first N rows in the data (by default, N=5)

In [7]:
```
data.head()
```

Out[7]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265.0	
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200.0	
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	200.0	
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6.0	270.0	
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6.0	225.0	

.shape

It shows the total no. of rows and no. of columns of the dataframe

In [8]:
```
data.shape
```

Out[8]:
```
(432, 15)
```

df.isnull( ).sum( )

It detects the missing values from each column of the dataframe.

fillna()

To fill the null values of a column with some particular value

value_counts

In a column, it shows all the unique values with their count. It can be applied to a single column only.

isin()

To show all records including particular elements

apply()

To apply a function along any axis of DataFrame

1) Instruction (For Data Cleaning) - Find all Null Values in the dataset. If there is any null value in any column, then fill it with the mean of that column.

In [11]:
```
data.isnull() #shows the column in boolean having null value
```

Out[11]:

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

432 rows × 15 columns

In [12]:
```
data.isnull().sum()
```

Out[12]:
```
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
```

In [13]:
```
data.notnull()
```

Out[13]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG
0	True	True	True	True	True	True	True	True	True	True	

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG
1	True	True	True	True	True	True	True	True	True	True	
2	True	True	True	True	True	True	True	True	True	True	
3	True	True	True	True	True	True	True	True	True	True	
4	True	True	True	True	True	True	True	True	True	True	
...	...	...	...	...	...	...	...	...	...	...	
427	True	True	True	True	True	True	True	True	True	True	
428	True	True	True	True	True	True	True	True	True	True	
429	True	True	True	True	True	True	True	True	True	True	
430	True	True	True	True	True	True	True	True	True	True	
431	True	True	True	True	True	True	True	True	True	True	

432 rows × 15 columns

In [14]:
```
data.notnull().sum()
```

Out[14]:
```
Make           428
Model          428
Type           428
Origin         428
DriveTrain     428
MSRP           428
Invoice        428
EngineSize     428
Cylinders      426
Horsepower     428
MPG_City       428
MPG_Highway    428
Weight         428
Wheelbase      428
Length         428
dtype: int64
```

In [16]:
```
data['Cylinders'].fillna(data['Cylinders'].mean()) #removing the null value from Cylind
```

Out[16]:
```
0      6.0
1      4.0
2      4.0
3      6.0
4      6.0
      ...
427    5.0
428    5.0
429    6.0
430    4.0
431    5.0
Name: Cylinders, Length: 432, dtype: object
```

In [18]:
```
data['Cylinders'].fillna(data['Cylinders'].mean(), inplace= True)
```

In [19]:
```
data.isnull().sum()
```

Out[19]:
```
Make          4
Model         4
Type          4
Origin        4
DriveTrain    4
MSRP          4
Invoice       4
EngineSize    4
Cylinders     0
Horsepower    4
MPG_City      4
MPG_Highway   4
Weight        4
Wheelbase     4
Length        4
dtype: int64
```

2) Question (Based on Value Counts)- Check what are the different types of Make are there in our dataset. And, what is the count (occurrence) of each Make in the data ?

In [20]:
```
data['Make'].value_counts(0)
```

Out[20]:
```
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
Lexus            11
Kia              11
Buick             9
Mercury           9
Lincoln           9
Saturn            8
Cadillac          8
Suzuki            8
Infiniti          8
GMC               8
Acura             7
Porsche           7
```

```
Saab              7
Land Rover        3
Oldsmobile        3
Jeep              3
Scion             2
Isuzu             2
MINI              2
Hummer            1
Name: Make, dtype: int64
```

3) Instruction (Filtering) - Show all the records where Origin is Asia or Europe.

In [21]:
```
data[data['Origin'].isin(['Asia','Europe'])]
```

Out[21]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepowe
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265.(
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200.(
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	200.(
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6.0	270.(
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6.0	225.(
...	...	...	...	...	...	...	...	...	...	.
427	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	5.0	197.(
428	Volvo	C70 HPT convertible 2dr	Sedan	Europe	Front	$42,565	$40,083	2.3	5.0	242.(
429	Volvo	S80 T6 4dr	Sedan	Europe	Front	$45,210	$42,573	2.9	6.0	268.(
430	Volvo	V40	Wagon	Europe	Front	$26,135	$24,641	1.9	4.0	170.(
431	Volvo	XC70	Wagon	Europe	All	$35,145	$33,112	2.5	5.0	208.(

281 rows × 15 columns

4)Instruction (Removing unwanted records) - Remove all the records (rows) where Weight is above 4000.

In [23]:
```
data[~(data['Weight']  > 4000)]
```

Out[23]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepov
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	20
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	20

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepov
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6.0	27
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6.0	22
5	Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	$46,100	$41,100	3.5	6.0	22
...	...	...	...	...	...	...	...	...	...	
427	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	5.0	19
428	Volvo	C70 HPT convertible 2dr	Sedan	Europe	Front	$42,565	$40,083	2.3	5.0	24
429	Volvo	S80 T6 4dr	Sedan	Europe	Front	$45,210	$42,573	2.9	6.0	26
430	Volvo	V40	Wagon	Europe	Front	$26,135	$24,641	1.9	4.0	17
431	Volvo	XC70	Wagon	Europe	All	$35,145	$33,112	2.5	5.0	20

329 rows × 15 columns

In [ ]:
```
#number of rows is 329 now
```

5) Instruction (Applying function on a column) - Increase all the values of 'MPG_City' column by 3.

In [26]:
```
data.head(2)
```

Out[26]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265.0	
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200.0	

In [29]:
```
data['MPG_City'] = data['MPG_City'].apply(lambda x:x+3)
```

In [30]:
```
data #we can see MPG city value got increased  by 3 above it was 17.
```

Out[30]:

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepowe
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265.(
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200.(

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepowe
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	200.(
3	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299	3.2	6.0	270.(
4	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014	3.5	6.0	225.(
...	...	...	...	...	...	...	...	...	...	.
427	Volvo	C70 LPT convertible 2dr	Sedan	Europe	Front	$40,565	$38,203	2.4	5.0	197.(
428	Volvo	C70 HPT convertible 2dr	Sedan	Europe	Front	$42,565	$40,083	2.3	5.0	242.(
429	Volvo	S80 T6 4dr	Sedan	Europe	Front	$45,210	$42,573	2.9	6.0	268.(
430	Volvo	V40	Wagon	Europe	Front	$26,135	$24,641	1.9	4.0	170.(
431	Volvo	XC70	Wagon	Europe	All	$35,145	$33,112	2.5	5.0	208.(

432 rows × 15 columns

In [ ]: