In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
auto_master = pd.read_csv(r"D:\PG-DAI\MachineLearning\9 December\Automobile\Automobile_data.csv")
```

Exercise 1: From the given dataset print the first and last five rows

In [3]:

```
auto_master.head(10)
```

Out[3]:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
0	0	alfa-romero	convertible	88.6	168.8	dohc	four	111	21	13495.0
1	1	alfa-romero	convertible	88.6	168.8	dohc	four	111	21	16500.0
2	2	alfa-romero	hatchback	94.5	171.2	ohcv	six	154	19	16500.0
3	3	audi	sedan	99.8	176.6	ohc	four	102	24	13950.0
4	4	audi	sedan	99.4	176.6	ohc	five	115	18	17450.0
5	5	audi	sedan	99.8	177.3	ohc	five	110	19	15250.0
6	6	audi	wagon	105.8	192.7	ohc	five	110	19	18920.0
7	9	bmw	sedan	101.2	176.8	ohc	four	101	23	16430.0
8	10	bmw	sedan	101.2	176.8	ohc	four	101	23	16925.0
9	11	bmw	sedan	101.2	176.8	ohc	six	121	21	20970.0

In [4]:

```
auto_master.tail(10)
```

Out[4]:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
51	69	toyota	wagon	95.7	169.7	ohc	four	62	31	6918.0
52	70	toyota	wagon	95.7	169.7	ohc	four	62	27	7898.0
53	71	toyota	wagon	95.7	169.7	ohc	four	62	27	8778.0
54	79	toyota	wagon	104.5	187.8	dohc	six	156	19	15750.0
55	80	volkswagen	sedan	97.3	171.7	ohc	four	52	37	7775.0

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
56	81	volkswagen	sedan	97.3	171.7	ohc	four	85	27	7975.0
57	82	volkswagen	sedan	97.3	171.7	ohc	four	52	37	7995.0
58	86	volkswagen	sedan	97.3	171.7	ohc	four	100	26	9995.0
59	87	volvo	sedan	104.3	188.8	ohc	four	114	23	12940.0
60	88	volvo	wagon	104.3	188.8	ohc	four	114	23	13415.0

Exercise 2: Clean the dataset and update the CSV file

In [7]:

```
auto_master.unique.sum()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-7-59d2f8492655> in <module>
----> 1 auto_master.unique.sum()

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
   5137             if self._info_axis._can_hold_identifiers_and_holds_name(name):
   5138                 return self[name]
-> 5139             return object.__getattribute__(self, name)
   5140 
   5141     def __setattr__(self, name: str, value) -> None:

AttributeError: 'DataFrame' object has no attribute 'unique'
```

In [24]:

```
auto_master.isna().sum()
```

Out[24]:

```
index               0
company             0
body-style          0
wheel-base          0
length              0
engine-type         0
num-of-cylinders    0
horsepower          0
average-mileage     0
price               0
dtype: int64
```

In [18]:

```
auto_master[auto_master['price']==auto_master['price'].isna()]
```

Out[18]:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price

In [19]:

```
auto_master[auto_master.isna().any(axis=1)]
```

Out[19]:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
22	31	isuzu	sedan	94.5	155.9	ohc	four	70	38	NaN
23	32	isuzu	sedan	94.5	155.9	ohc	four	70	38	NaN
47	63	porsche	hatchback	98.4	175.7	dohcv	eight	288	17	NaN

In [23]:

```
auto_master = auto_master.dropna(axis=0)
```

Exercise 3: Find the most expensive car company name

In [31]:

```
auto_master[auto_master.price == auto_master.price.max()]
```

Out[31]:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
35	47	mercedes-benz	hardtop	112.0	199.2	ohcv	eight	184	14	45400.0

In [38]:

```
# Exercise 4: Print All Toyota Cars details

auto_master[auto_master.company == 'toyota' ]
```

Out[38]:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
48	66	toyota	hatchback	95.7	158.7	ohc	four	62	35	5348.0
49	67	toyota	hatchback	95.7	158.7	ohc	four	62	31	6338.0
50	68	toyota	hatchback	95.7	158.7	ohc	four	62	31	6488.0
51	69	toyota	wagon	95.7	169.7	ohc	four	62	31	6918.0
52	70	toyota	wagon	95.7	169.7	ohc	four	62	27	7898.0
53	71	toyota	wagon	95.7	169.7	ohc	four	62	27	8778.0
54	79	toyota	wagon	104.5	187.8	dohc	six	156	19	15750.0

In [42]:

```
# Exercise 5: Count total cars per company

auto_master['company'].value_counts()
```

Out[42]:

```
toyota           7
bmw              6
nissan           5
mazda            5
volkswagen       4
mercedes-benz    4
audi             4
mitsubishi       4
honda            3
alfa-romero      3
jaguar           3
chevrolet        3
porsche          2
dodge            2
volvo            2
isuzu            1
Name: company, dtype: int64
```

In [63]:

```
# Exercise 6: Find each company’s Highest price car
car_comp = auto_master.groupby('company')
car_comp['price'].max()
```

Out[63]:

```
company
alfa-romero      16500.0
audi             18920.0
bmw              41315.0
chevrolet         6575.0
dodge             6377.0
honda            12945.0
isuzu             6785.0
jaguar           36000.0
mazda            18344.0
mercedes-benz    45400.0
mitsubishi        8189.0
nissan           13499.0
porsche          37028.0
toyota           15750.0
volkswagen        9995.0
volvo            13415.0
Name: price, dtype: float64
```

In [66]:

```
# Exercise 7: Find the average mileage of each car making company
```

```
car_compa = auto_master.groupby('company')
car_compa['average-mileage'].mean()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-66-b9eb1b651928> in <module>
      2 
      3 car_compa = auto_master.groupby('company')
----> 4 car_compa['average-mileage'].mean().sort()

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
   5137             if self._info_axis._can_hold_identifiers_and_holds_name(name):
   5138                 return self[name]
-> 5139             return object.__getattribute__(self, name)
   5140 
   5141     def __setattr__(self, name: str, value) -> None:

AttributeError: 'Series' object has no attribute 'sort'
```

In [67]:

```
# Exercise 8: Sort all cars by Price column
auto_master.sort_values('price')
```

Out[67]:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
13	16	chevrolet	hatchback	88.4	141.1	l	three	48	47	5151.0
27	36	mazda	hatchback	93.1	159.1	ohc	four	68	30	5195.0
48	66	toyota	hatchback	95.7	158.7	ohc	four	62	35	5348.0
36	49	mitsubishi	hatchback	93.7	157.3	ohc	four	68	37	5389.0
28	37	mazda	hatchback	93.1	159.1	ohc	four	68	31	6095.0
37	50	mitsubishi	hatchback	93.7	157.3	ohc	four	68	31	6189.0
17	20	dodge	hatchback	93.7	157.3	ohc	four	68	31	6229.0
14	17	chevrolet	hatchback	94.5	155.9	ohc	four	70	38	6295.0
49	67	toyota	hatchback	95.7	158.7	ohc	four	62	31	6338.0
16	19	dodge	hatchback	93.7	157.3	ohc	four	68	31	6377.0
50	68	toyota	hatchback	95.7	158.7	ohc	four	62	31	6488.0
15	18	chevrolet	sedan	94.5	158.8	ohc	four	70	38	6575.0

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
41	54	nissan	sedan	94.5	165.3	ohc	four	69	31	6649.0
21	30	isuzu	sedan	94.3	170.7	ohc	four	78	24	6785.0
29	38	mazda	hatchback	93.1	159.1	ohc	four	68	31	6795.0
42	55	nissan	sedan	94.5	165.3	ohc	four	69	31	6849.0
51	69	toyota	wagon	95.7	169.7	ohc	four	62	31	6918.0
38	51	mitsubishi	sedan	96.3	172.4	ohc	four	88	25	6989.0
40	53	nissan	sedan	94.5	165.3	ohc	four	55	45	7099.0
18	27	honda	wagon	96.5	157.1	ohc	four	76	30	7295.0
43	56	nissan	wagon	94.5	170.2	ohc	four	69	31	7349.0
55	80	volkswagen	sedan	97.3	171.7	ohc	four	52	37	7775.0
52	70	toyota	wagon	95.7	169.7	ohc	four	62	27	7898.0
56	81	volkswagen	sedan	97.3	171.7	ohc	four	85	27	7975.0
57	82	volkswagen	sedan	97.3	171.7	ohc	four	52	37	7995.0
39	52	mitsubishi	sedan	96.3	172.4	ohc	four	88	25	8189.0
53	71	toyota	wagon	95.7	169.7	ohc	four	62	27	8778.0
58	86	volkswagen	sedan	97.3	171.7	ohc	four	100	26	9995.0
20	29	honda	sedan	96.5	169.1	ohc	four	100	25	10345.0
30	39	mazda	hatchback	95.3	169.0	rotor	two	101	17	11845.0
59	87	volvo	sedan	104.3	188.8	ohc	four	114	23	12940.0
19	28	honda	sedan	96.5	175.4	ohc	four	101	24	12945.0
60	88	volvo	wagon	104.3	188.8	ohc	four	114	23	13415.0
0	0	alfa-romero	convertible	88.6	168.8	dohc	four	111	21	13495.0
44	57	nissan	sedan	100.4	184.6	ohcv	six	152	19	13499.0
3	3	audi	sedan	99.8	176.6	ohc	four	102	24	13950.0

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
5	5	audi	sedan	99.8	177.3	ohc	five	110	19	15250.0
54	79	toyota	wagon	104.5	187.8	dohc	six	156	19	15750.0
7	9	bmw	sedan	101.2	176.8	ohc	four	101	23	16430.0
1	1	alfa-romero	convertible	88.6	168.8	dohc	four	111	21	16500.0
2	2	alfa-romero	hatchback	94.5	171.2	ohcv	six	154	19	16500.0
8	10	bmw	sedan	101.2	176.8	ohc	four	101	23	16925.0
4	4	audi	sedan	99.4	176.6	ohc	five	115	18	17450.0
31	43	mazda	sedan	104.9	175.0	ohc	four	72	31	18344.0
6	6	audi	wagon	105.8	192.7	ohc	five	110	19	18920.0
9	11	bmw	sedan	101.2	176.8	ohc	six	121	21	20970.0
32	44	mercedes-benz	sedan	110.0	190.9	ohc	five	123	22	25552.0
33	45	mercedes-benz	wagon	110.0	190.9	ohc	five	123	22	28248.0
10	13	bmw	sedan	103.5	189.0	ohc	six	182	16	30760.0
24	33	jaguar	sedan	113.0	199.6	dohc	six	176	15	32250.0
45	61	porsche	hardtop	89.5	168.9	ohcf	six	207	17	34028.0
25	34	jaguar	sedan	113.0	199.6	dohc	six	176	15	35550.0
26	35	jaguar	sedan	102.0	191.7	ohcv	twelve	262	13	36000.0
12	15	bmw	sedan	110.0	197.0	ohc	six	182	15	36880.0
46	62	porsche	convertible	89.5	168.9	ohcf	six	207	17	37028.0
34	46	mercedes-benz	sedan	120.9	208.1	ohcv	eight	184	14	40960.0
11	14	bmw	sedan	103.5	193.8	ohc	six	182	16	41315.0
35	47	mercedes-benz	hardtop	112.0	199.2	ohcv	eight	184	14	45400.0

In [69]:

```
# Exercise 9: Concatenate two data frames using the following conditions
```

```
GermanCars = {'Company': ['Ford', 'Mercedes', 'BMV', 'Audi'], 'Price': [23845, 171995, 135925 , 71400]}
japaneseCars = {'Company': ['Toyota', 'Honda', 'Nissan', 'Mitsubishi '], 'Price': [29995, 23600, 61500 , 58900]}
```

In [72]:

```
cars_German = pd.DataFrame.from_dict(GermanCars)
cars_Japanese = pd.DataFrame.from_dict(japaneseCars)
pd.concat([cars_German, cars_Japanese], keys=["Germany", "Japanese"])
```

Out[72]:

		Company	Price
Germany	**0**	Ford	23845
	1	Mercedes	171995
	2	BMV	135925
	3	Audi	71400
Japanese	**0**	Toyota	29995
	1	Honda	23600
	2	Nissan	61500
	3	Mitsubishi	58900

In [77]:

```
# Exercise 10: Merge two data frames using the following condition
# *Create two data frames using the following two Dicts, Merge two data frames, and append the second data frame as a new column t

Car_Price = {'Company': ['Toyota', 'Honda', 'BMV', 'Audi'], 'Price': [23845, 17995, 135925 , 71400]}
Car_Horsepower = {'Company': ['Toyota', 'Honda', 'BMV', 'Audi'], 'horsepower': [141, 80, 182 , 160]}
```

In [90]:

```
Car_Price = pd.DataFrame.from_dict(Car_Price)
Car_Horsepower = pd.DataFrame.from_dict(Car_Horsepower)
```

In [93]:

```
Car_Price.merge(Car_Horsepower)
```

Out[93]:

	Company	Price	horsepower
0	Toyota	23845	141
1	Honda	17995	80
2	BMV	135925	182

	Company	Price	horsepower
3	Audi	71400	160