

```
In [2]: import numpy as np
import pandas as pd
```

```
In [3]: #import the dataset
data = pd.read_csv('Automobile_data.csv')
```

```
In [4]: #print first 5
data.head(5)
```

Out[4]:

	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

```
In [5]: #last 5 rows
data.tail(5)
```

Out[5]:

	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

```
In [6]: #to find the number of rows having Nan value
data.isna().sum()
```

```
Out[6]: 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           3
dtype: int64
```

```
In [7]: #Most expensive car
data[data.price==data.price.max()]
```

Out[7]:

	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 [8]: #All toyota cars
data[data.company=='toyota']
```

Out[8]:

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

```
In [9]: #Number of total cars per company
data['company'].value_counts()
```

```
Out[9]: toyota      7
        bmw        6
        nissan     5
        mazda      5
        volkswagen 4
        mitsubishi 4
        mercedes-benz 4
        audi       4
        porsche    3
        honda      3
        isuzu      3
        chevrolet  3
        jaguar     3
        alfa-romero 3
        dodge      2
        volvo      2
        Name: company, dtype: int64
```

```
In [10]: #Each company highest price car
data.groupby('company').max()
```

Out[10]:

	index	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
company									
alfa-romero	2	hatchback	94.5	171.2	ohcv	six	154	21	16500.0
audi	6	wagon	105.8	192.7	ohc	four	115	24	18920.0
bmw	15	sedan	110.0	197.0	ohc	six	182	23	41315.0
chevrolet	18	sedan	94.5	158.8	ohc	three	70	47	6575.0
dodge	20	hatchback	93.7	157.3	ohc	four	68	31	6377.0
honda	29	wagon	96.5	175.4	ohc	four	101	30	12945.0
isuzu	32	sedan	94.5	170.7	ohc	four	78	38	6785.0
jaguar	35	sedan	113.0	199.6	ohcv	twelve	262	15	36000.0
mazda	43	sedan	104.9	175.0	rotor	two	101	31	18344.0
mercedes-benz	47	wagon	120.9	208.1	ohcv	five	184	22	45400.0
mitsubishi	52	sedan	96.3	172.4	ohc	four	88	37	8189.0
nissan	57	wagon	100.4	184.6	ohcv	six	152	45	13499.0
porsche	63	hatchback	98.4	175.7	ohcf	six	288	17	37028.0
toyota	79	wagon	104.5	187.8	ohc	six	156	35	15750.0
volkswagen	86	sedan	97.3	171.7	ohc	four	100	37	9995.0
volvo	88	wagon	104.3	188.8	ohc	four	114	23	13415.0

```
In [11]: #Average Mileage of car company
df1=data.groupby('company').mean()
df1['average-mileage']
```

```
Out[11]: company
alfa-romero      20.333333
audi             20.000000
bmw             19.000000
chevrolet       41.000000
dodge           31.000000
honda           26.333333
isuzu           33.333333
jaguar          14.333333
mazda           28.000000
mercedes-benz   18.000000
mitsubishi      29.500000
nissan           31.400000
porsche         17.000000
toyota          28.714286
volkswagen      31.750000
volvo           23.000000
Name: average-mileage, dtype: float64
```

```
In [12]: #Sorting by price  
data.sort_values(by='price',ascending=False)
```

Out[12]:

	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
11	14	bmw	sedan	103.5	193.8	ohc	six	182	16	41300
34	46	mercedes-benz	sedan	120.9	208.1	ohcv	eight	184	14	40900
46	62	porsche	convertible	89.5	168.9	ohcf	six	207	17	37000
12	15	bmw	sedan	110.0	197.0	ohc	six	182	15	36800
26	35	jaguar	sedan	102.0	191.7	ohcv	twelve	262	13	36000
25	34	jaguar	sedan	113.0	199.6	dohc	six	176	15	35500
45	61	porsche	hardtop	89.5	168.9	ohcf	six	207	17	34000
24	33	jaguar	sedan	113.0	199.6	dohc	six	176	15	32200
10	13	bmw	sedan	103.5	189.0	ohc	six	182	16	30700
33	45	mercedes-benz	wagon	110.0	190.9	ohc	five	123	22	28200
32	44	mercedes-benz	sedan	110.0	190.9	ohc	five	123	22	25500
9	11	bmw	sedan	101.2	176.8	ohc	six	121	21	20900
6	6	audi	wagon	105.8	192.7	ohc	five	110	19	18900
31	43	mazda	sedan	104.9	175.0	ohc	four	72	31	18300
4	4	audi	sedan	99.4	176.6	ohc	five	115	18	17400
8	10	bmw	sedan	101.2	176.8	ohc	four	101	23	16900
2	2	alfa-romero	hatchback	94.5	171.2	ohcv	six	154	19	16500
1	1	alfa-romero	convertible	88.6	168.8	dohc	four	111	21	16500
7	9	bmw	sedan	101.2	176.8	ohc	four	101	23	16400
54	79	toyota	wagon	104.5	187.8	dohc	six	156	19	15700
5	5	audi	sedan	99.8	177.3	ohc	five	110	19	15200
3	3	audi	sedan	99.8	176.6	ohc	four	102	24	13900
44	57	nissan	sedan	100.4	184.6	ohcv	six	152	19	13400
0	0	alfa-romero	convertible	88.6	168.8	dohc	four	111	21	13400
60	88	volvo	wagon	104.3	188.8	ohc	four	114	23	13400
19	28	honda	sedan	96.5	175.4	ohc	four	101	24	12900
59	87	volvo	sedan	104.3	188.8	ohc	four	114	23	12900
30	39	mazda	hatchback	95.3	169.0	rotor	two	101	17	11800
20	29	honda	sedan	96.5	169.1	ohc	four	100	25	10300
...
53	71	toyota	wagon	95.7	169.7	ohc	four	62	27	8770
39	52	mitsubishi	sedan	96.3	172.4	ohc	four	88	25	8180
57	82	volkswagen	sedan	97.3	171.7	ohc	four	52	37	7990
56	81	volkswagen	sedan	97.3	171.7	ohc	four	85	27	7970

```
In [13]: #Concatation
GermanCars = {'Company': ['Ford', 'Mercedes', 'BMV', 'Audi'], 'Price':
[23845, 171995, 135925 , 71400]}
japaneseCars = {'Company': ['Toyota', 'Honda', 'Nissan', 'Mitsubishi '],
'Price': [29995, 23600, 61500 , 58900]}
```

```
In [14]: data1 = pd.DataFrame(GermanCars)
data2 = pd.DataFrame(japaneseCars)
```

```
In [15]: frames = [data1,data2]
```

```
In [16]: result = pd.concat(frames,keys=[ 'German', 'Japan'])
result
```

Out[16]:

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

```
In [17]: #Merge
Car_Price = {'Company': ['Toyota', 'Honda', 'BMV', 'Audi'], 'Price':
[23845, 17995, 135925 , 71400]}
car_Horsepower = {'Company': ['Toyota', 'Honda', 'BMV', 'Audi'], 'horsepower':
[141, 80, 182 , 160]}
data3 = pd.DataFrame(Car_Price)
data4 = pd.DataFrame(car_Horsepower)
```

```
In [18]: res = pd.merge(data3,data4)
```

```
In [19]: res
```

Out[19]:

	Company	Price	horsepower
0	Toyota	23845	141
1	Honda	17995	80
2	BMV	135925	182
3	Audi	71400	160