

## Pandas Practice Question

*print first and last five rows*

In [20]:

```
import pandas as pd
data = pd.read_csv("Automobile_data.csv")
```

*print first and last five rows*

In [21]:

```
data.head(5)
```

Out[21]:

	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 [22]:

```
data.tail(5)
```

Out[22]:

	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

*Print most expensive car's company name and price.*

In [23]:

```
data1 = data[['company', 'price']][data.price == data['price'].max()]
data1
```

Out[23]:

	company	price
35	mercedes-benz	45400.0

*Print All Toyota Cars details*

In [24]:

```
df = data.groupby('company')
```

```
toyotadf = df.get_group('toyota')
toyotadf
```

Out[24]:

	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

### Count total cars per company

In [28]:

```
total_count = data['company'].value_counts()
total_count
```

Out[28]:

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

### Find each company's Higesht price car

In [32]:

```
import warnings
warnings.filterwarnings("ignore")
com = data.groupby('company')
highest = com['company', 'price'].max()
print(highest)
```

```
company      company  price
alfa-romero  alfa-romero  16500.0
audi         audi       18920.0
bmw          bmw        41315.0
chevrolet    chevrolet   6575.0
dodge        dodge      6377.0
honda        honda      12945.0
isuzu        isuzu       6785.0
jaguar       jaguar      36000.0
mazda        mazda      18344.0
mercedes-benz mercedes-benz 45400.0
mitsubishi   mitsubishi   8189.0
nissan        nissan      13499.0
porsche      porsche     37028.0
```

```
toyota          toyota  15750.0
volkswagen      volkswagen  9995.0
volvo           volvo   13415.0
```

### average mileage of each car making company

In [35]:

```
df1 = data.groupby('company')
avg_mileage = df1['average-mileage'].mean()
avg_mileage
```

Out[35]:

average-mileage	
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

### Sort all cars by Price column

In [40]:

```
cars_price = data.sort_values(by=['price'], ascending = False)
cars_price.head()
```

Out[40]:

	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
11	14	bmw	sedan	103.5	193.8	ohc	six	182	16	41315.0
34	46	mercedes-benz	sedan	120.9	208.1	ohcv	eight	184	14	40960.0
46	62	porsche	convertible	89.5	168.9	ohcf	six	207	17	37028.0
12	15	bmw	sedan	110.0	197.0	ohc	six	182	15	36880.0

### Create two data frames using the following two Dicts, Concatenate those two data frames and create a key for each data frame.

In [41]:

```
GermanCars = {'Company': ['Ford', 'Mercedes', 'BMW', 'Audi'], 'Price': [23845, 171995, 135925, 71400]}
carsDf1 = pd.DataFrame.from_dict(GermanCars)
```

```
carsDf1 = pd.DataFrame.from_dict(GermanCars)

japaneseCars = {'Company': ['Toyota', 'Honda', 'Nissan', 'Mitsubishi '], 'Price': [29995, 23600, 61500, 58900]}
carsDf2 = pd.DataFrame.from_dict(japaneseCars)

carsDf = pd.concat([carsDf1, carsDf2], keys=["Germany", "Japan"])
carsDf
```

Out[41]:

		Company	Price
Germany	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 [ ]: