Pandas Exercise

In this exercise, we are using **Automobile Dataset** for data analysis. This Dataset has different characteristics of an auto such as body-style, wheel-base, engine-type, price, mileage, horsepower, etc.

What included in this Pandas exercise?

- It contains 10 questions. The solution is provided for each question.
- Each question includes a specific Pandas topic you need to learn.

When you complete each question, you get more familiar with data analysis using pandas.

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

	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

	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

```python
import pandas as pd
df = pd.read_csv("D:\\Python\\Articles\\pandas\\automobile-
dataset\\Automobile_data.csv")
df.head(5)
df.tail(5)
```

Exercise 2: Clean the dataset and update the CSV file

Replace all column values which contain `?`, `n.a`, or `NaN`.

```python
'price':["?","n.a"],
'stroke':["?","n.a"],
'horsepower':["?","n.a"],
'peak-rpm':["?","n.a"],
'average-mileage':["?","n.a"]})
```

```
print (df)
```

Exercise 3: Find the most expensive car company name

Print most expensive car's company name and price.

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

	company	price
35	mercedes-benz	45400.0

Exercise 4: Print All Toyota Cars details

	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

```
car_Manufacturers = df.groupby('company')
toyotaDf = car_Manufacturers.get_group('toyota')
toyotaDf
```

Exercise 5: Count total cars per company

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

```python
df['company'].value_counts()
```

Exercise 6: Find each company's Highest price car

	company	price
company		
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

```python
car_Manufacturers = df.groupby('company')
priceDf = car_Manufacturers['company','price'].max()
```

```
priceDf
```

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

	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

```
car_Manufacturers = df.groupby('company')
mileageDf = car_Manufacturers['company','average-mileage'].mean()
mileageDf
```

Exercise 8: Sort all cars by Price column

	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

```
carsDf = pd.read_csv("D:\\Python\\Articles\\pandas\\automobile-
dataset\\Automobile_data.csv")
carsDf = carsDf.sort_values(by=['price', 'horsepower'], ascending=False)
carsDf.head(5)
```

Exercise 9: Concatenate two data frames using the following conditions

Create two data frames using the following two [dictionaries](#).

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

```
GermanCars = {'Company': ['Ford', 'Mercedes', 'BMV', 'Audi'], 'Price': [23845,
171995, 135925 , 71400]}
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
```

		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

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 to the first data frame.

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

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

```
Car_Price = {'Company': ['Toyota', 'Honda', 'BMV', 'Audi'], 'Price': [23845,
17995, 135925 , 71400]}
carPriceDf = pd.DataFrame.from_dict(Car_Price)

car_Horsepower = {'Company': ['Toyota', 'Honda', 'BMV', 'Audi'], 'horsepower':
[141, 80, 182 , 160]}
carsHorsepowerDf = pd.DataFrame.from_dict(car_Horsepower)

carsDf = pd.merge(carPriceDf, carsHorsepowerDf, on="Company")
carsDf
```