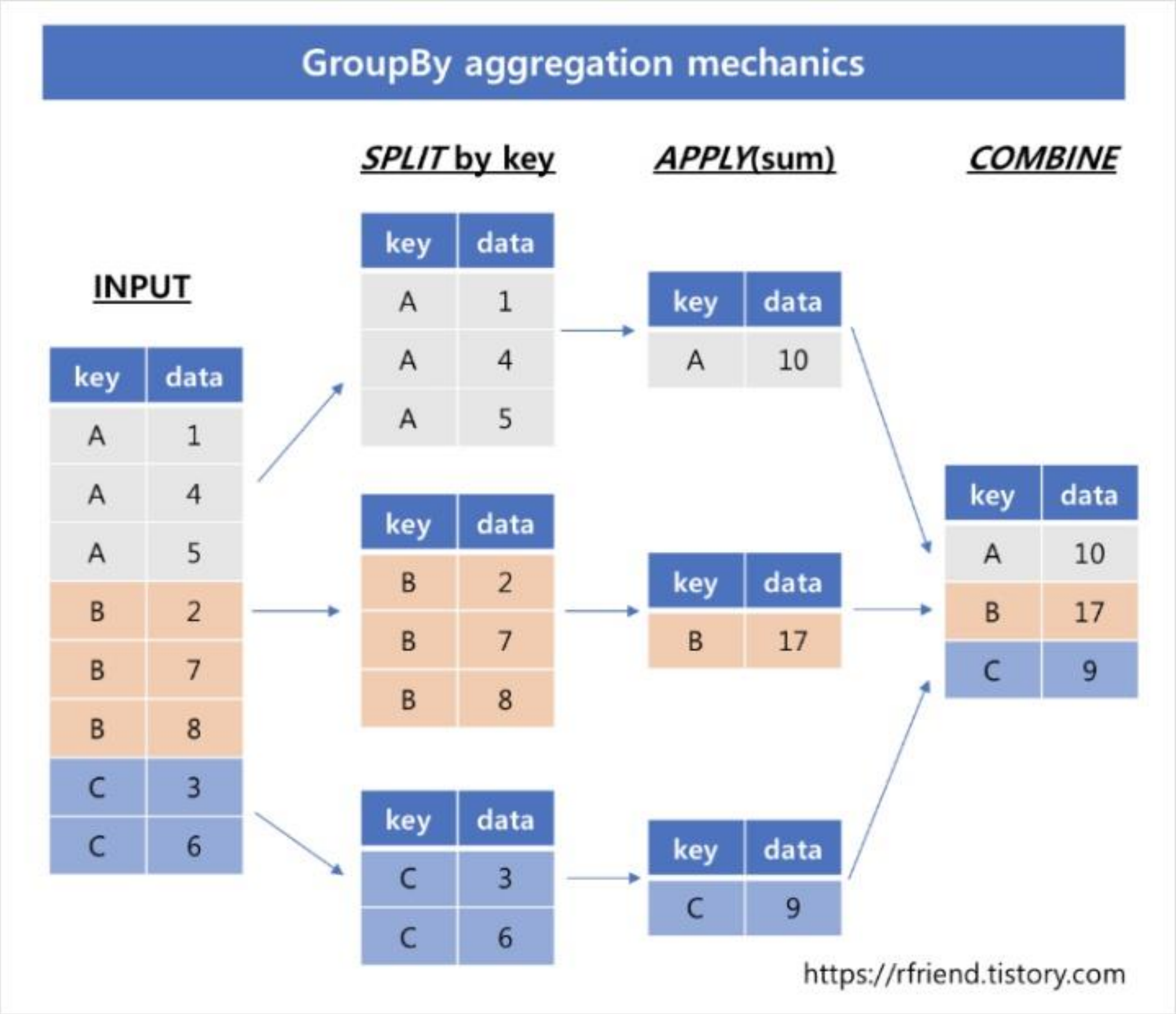
Chapter 03.
Group & Apply

목차

1. Groupby
2. Apply
3. Pivot table

1. Group by



GroupBy aggregation mechanics

INPUT

key	data
A	1
A	4
A	5
B	2
B	7
B	8
C	3
C	6

SPLIT by key

key	data
A	1
A	4
A	5

key	data
B	2
B	7
B	8

key	data
C	3
C	6

APPLY(sum)

key	data
A	10

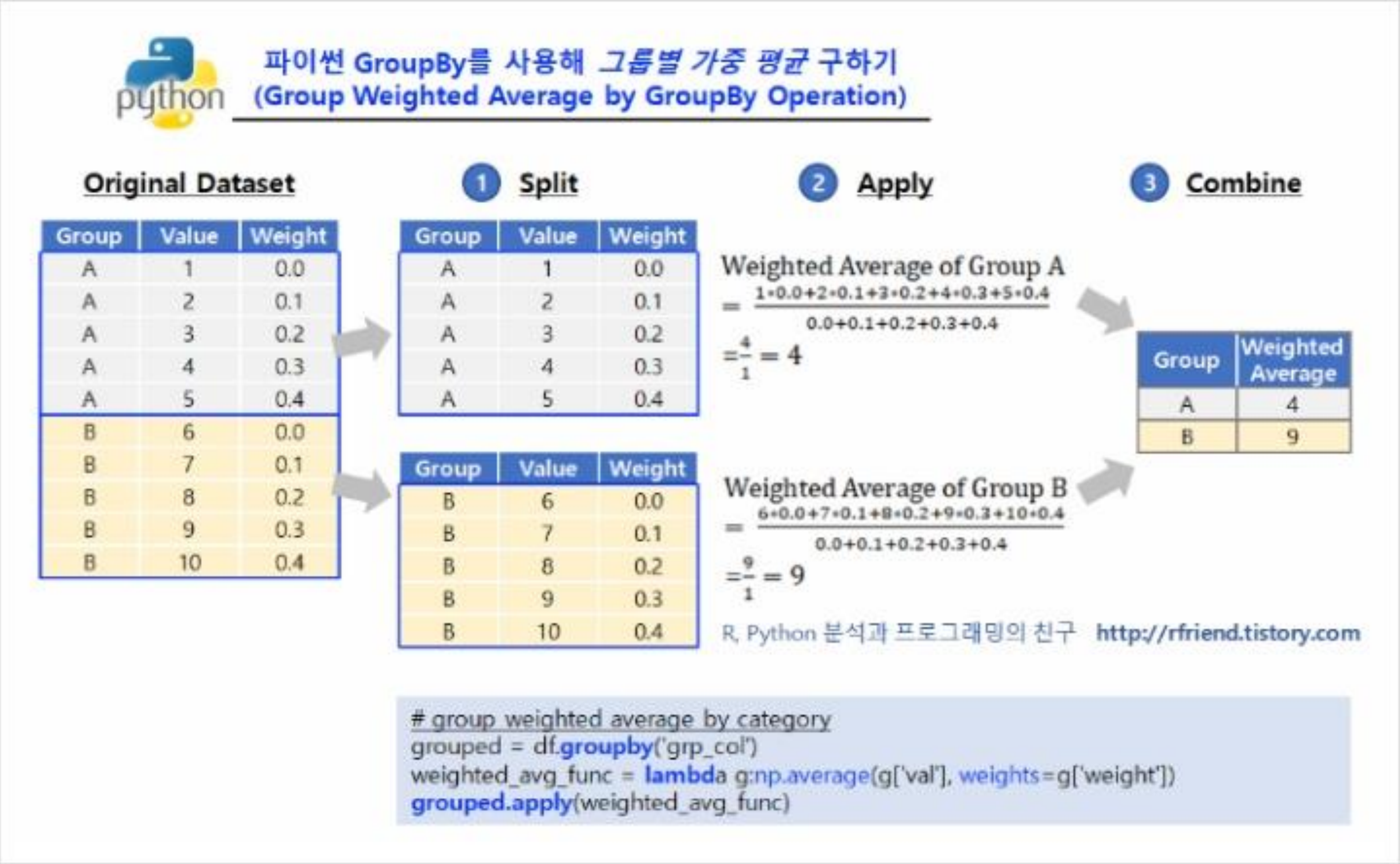
key	data
B	17

key	data
C	9

COMBINE

key	data
A	10
B	17
C	9

https://rfriend.tistory.com



파이썬 GroupBy를 사용해 *그룹별 가중 평균* 구하기
(Group Weighted Average by GroupBy Operation)

Original Dataset

Group	Value	Weight
A	1	0.0
A	2	0.1
A	3	0.2
A	4	0.3
A	5	0.4
B	6	0.0
B	7	0.1
B	8	0.2
B	9	0.3
B	10	0.4

① Split

Group	Value	Weight
A	1	0.0
A	2	0.1
A	3	0.2
A	4	0.3
A	5	0.4

Group	Value	Weight
B	6	0.0
B	7	0.1
B	8	0.2
B	9	0.3
B	10	0.4

② Apply

Weighted Average of Group A

$$= \frac{1 \cdot 0.0 + 2 \cdot 0.1 + 3 \cdot 0.2 + 4 \cdot 0.3 + 5 \cdot 0.4}{0.0 + 0.1 + 0.2 + 0.3 + 0.4}$$

$$= \frac{4}{1} = 4$$

Weighted Average of Group B

$$= \frac{6 \cdot 0.0 + 7 \cdot 0.1 + 8 \cdot 0.2 + 9 \cdot 0.3 + 10 \cdot 0.4}{0.0 + 0.1 + 0.2 + 0.3 + 0.4}$$

$$= \frac{9}{1} = 9$$

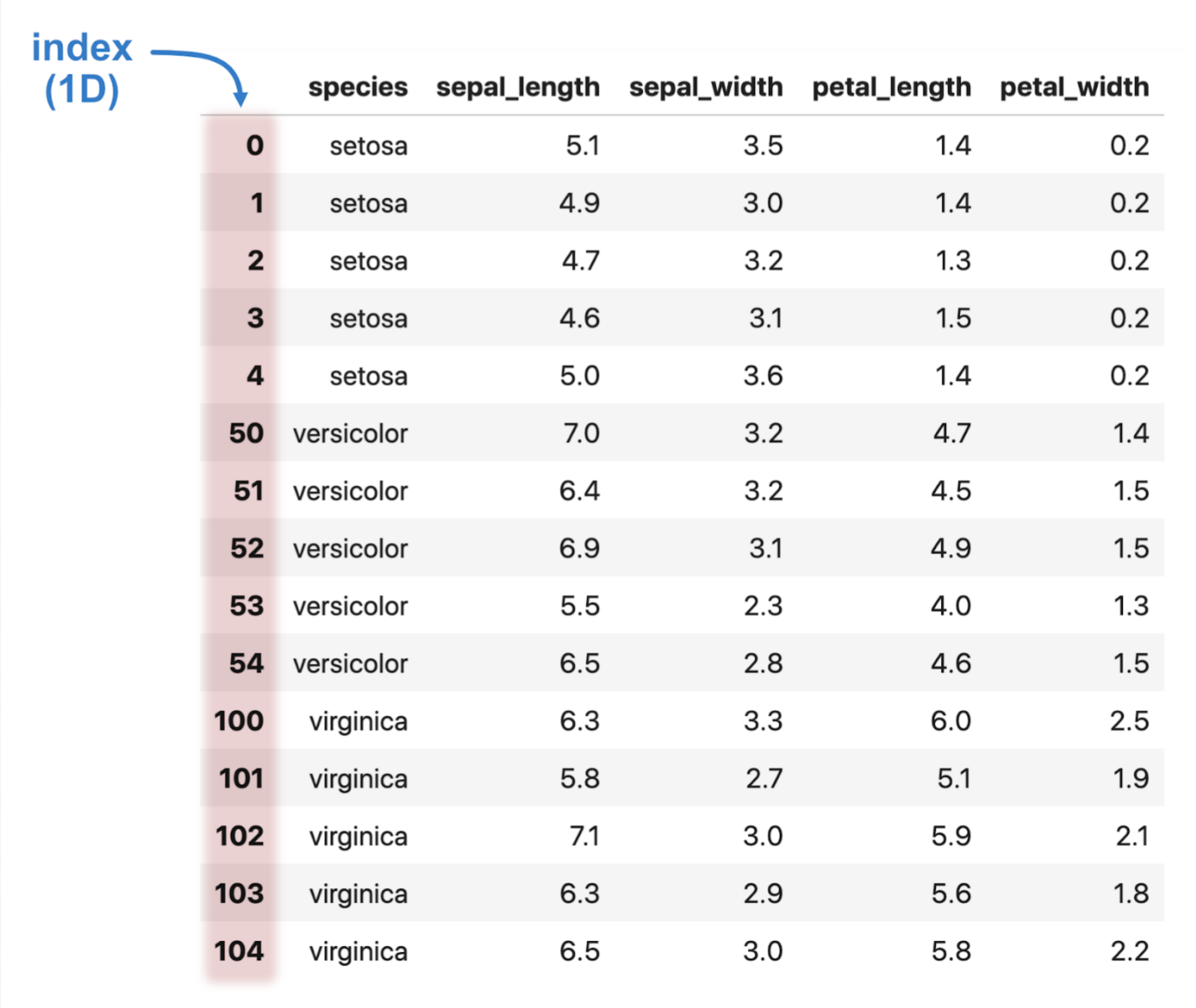
③ Combine

Group	Weighted Average
A	4
B	9

R, Python 분석과 프로그래밍의 친구 **http://rfriend.tistory.com**

```
# group weighted average by category
grouped = df.groupby('grp_col')
weighted_avg_func = lambda g:np.average(g['val'], weights=g['weight'])
grouped.apply(weighted_avg_func)
```

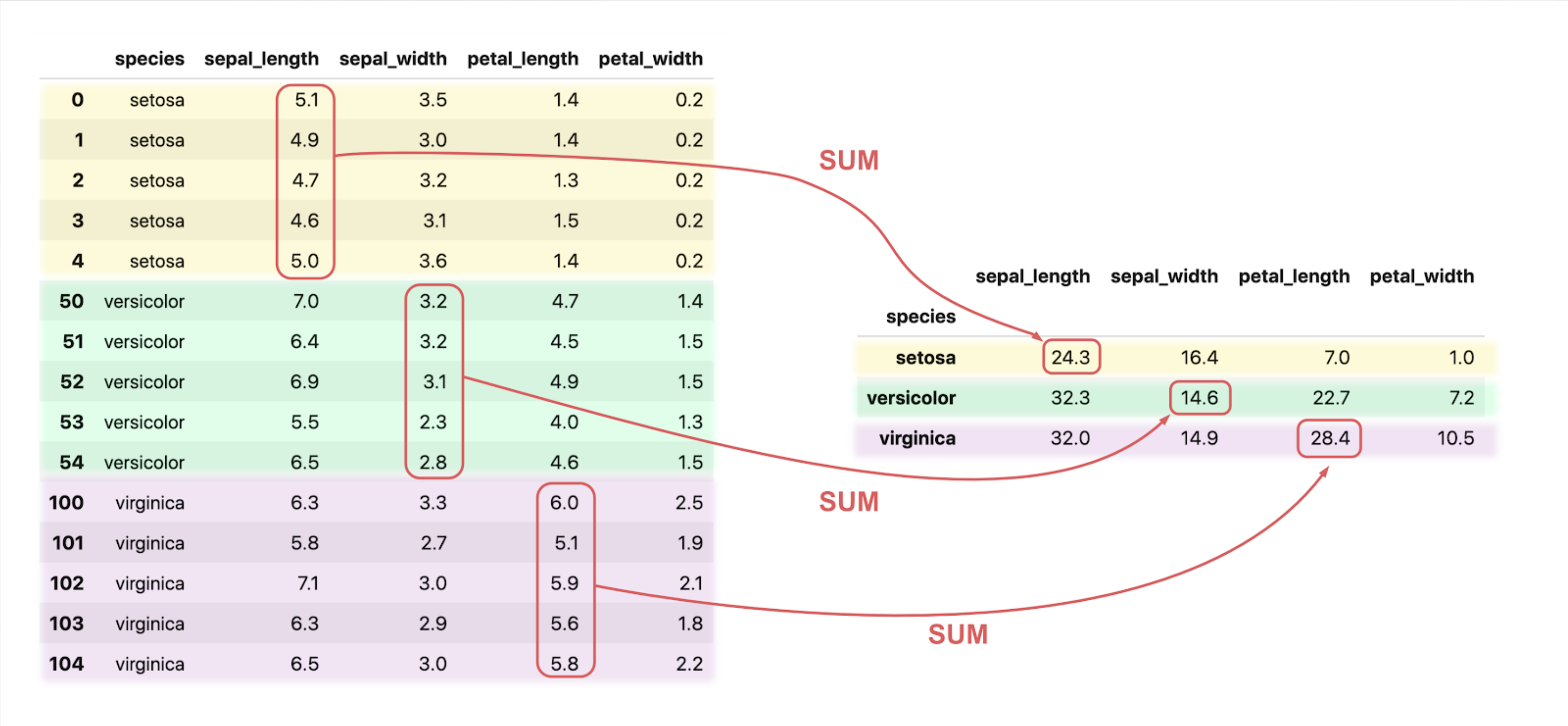
Group by example



	species	sepal_length	sepal_width	petal_length	petal_width
0	setosa	5.1	3.5	1.4	0.2
1	setosa	4.9	3.0	1.4	0.2
2	setosa	4.7	3.2	1.3	0.2
3	setosa	4.6	3.1	1.5	0.2
4	setosa	5.0	3.6	1.4	0.2
50	versicolor	7.0	3.2	4.7	1.4
51	versicolor	6.4	3.2	4.5	1.5
52	versicolor	6.9	3.1	4.9	1.5
53	versicolor	5.5	2.3	4.0	1.3
54	versicolor	6.5	2.8	4.6	1.5
100	virginica	6.3	3.3	6.0	2.5
101	virginica	5.8	2.7	5.1	1.9
102	virginica	7.1	3.0	5.9	2.1
103	virginica	6.3	2.9	5.6	1.8
104	virginica	6.5	3.0	5.8	2.2

- 집단변수 1개를 이용하여 전체 칼럼 그룹화

df.groupby(‘species’).sum()

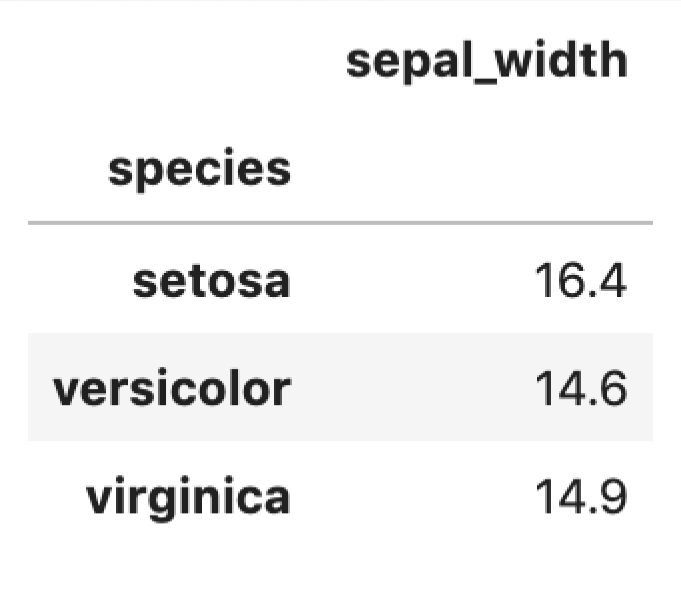


	species	sepal_length	sepal_width	petal_length	petal_width
0	setosa	5.1	3.5	1.4	0.2
1	setosa	4.9	3.0	1.4	0.2
2	setosa	4.7	3.2	1.3	0.2
3	setosa	4.6	3.1	1.5	0.2
4	setosa	5.0	3.6	1.4	0.2
50	versicolor	7.0	3.2	4.7	1.4
51	versicolor	6.4	3.2	4.5	1.5
52	versicolor	6.9	3.1	4.9	1.5
53	versicolor	5.5	2.3	4.0	1.3
54	versicolor	6.5	2.8	4.6	1.5
100	virginica	6.3	3.3	6.0	2.5
101	virginica	5.8	2.7	5.1	1.9
102	virginica	7.1	3.0	5.9	2.1
103	virginica	6.3	2.9	5.6	1.8
104	virginica	6.5	3.0	5.8	2.2

	sepal_length	sepal_width	petal_length	petal_width
species				
setosa	24.3	16.4	7.0	1.0
versicolor	32.3	14.6	22.7	7.2
virginica	32.0	14.9	28.4	10.5

- **집단변수 1개를 이용하여 1개 칼럼 그룹화**

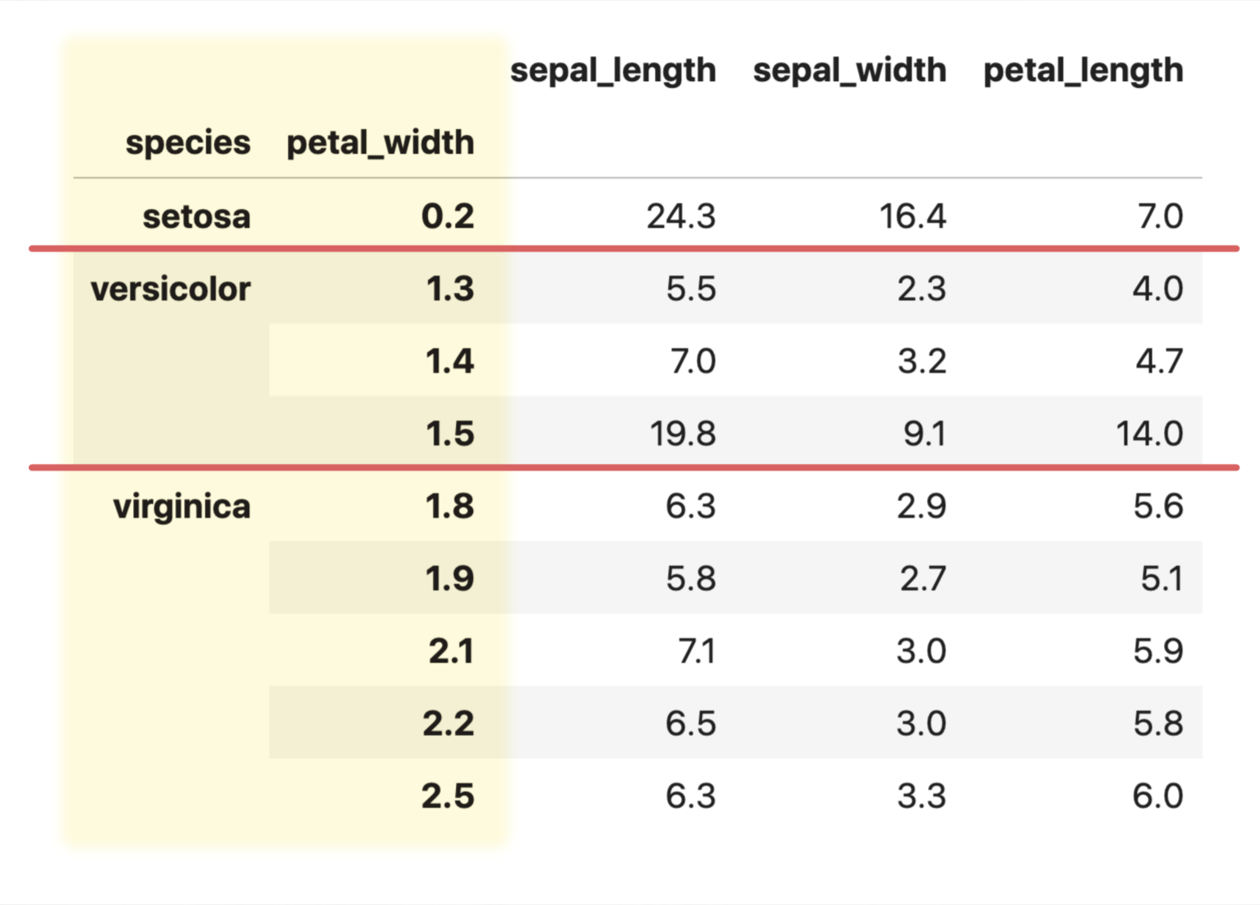
```
df[‘sepal.width’].groupby(df[‘species’]).sum()
```



	sepal_width
species	
setosa	16.4
versicolor	14.6
virginica	14.9

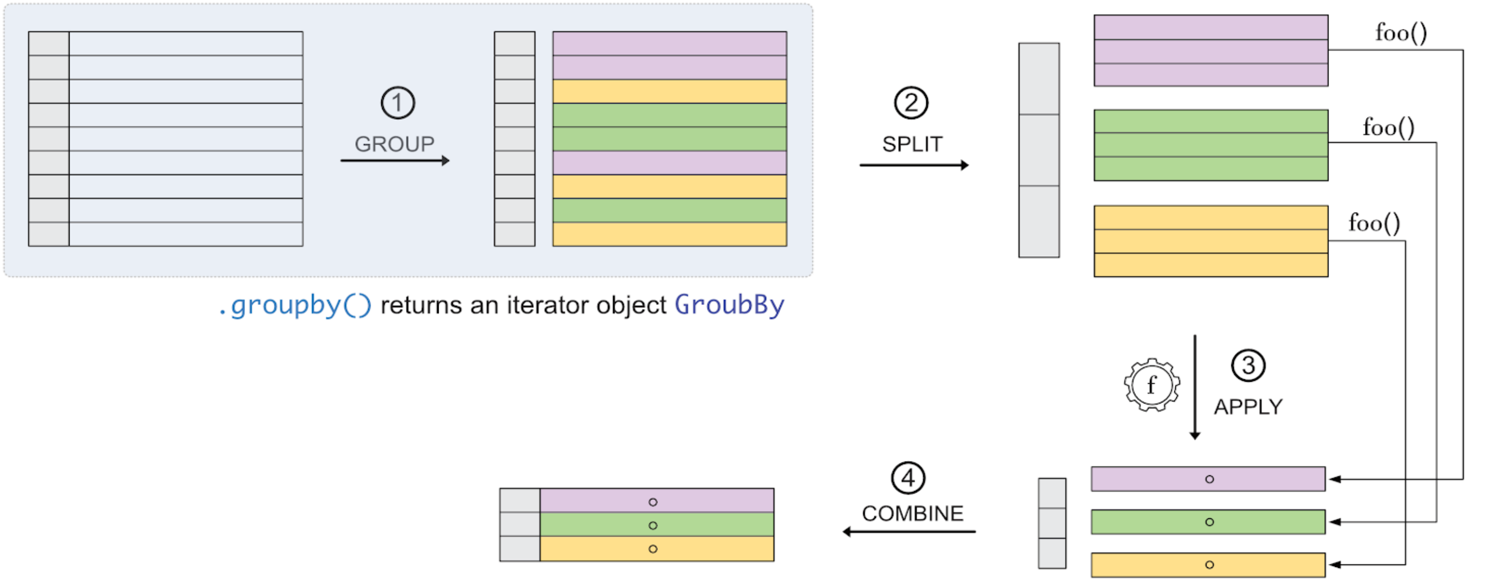
- **2개 변수를 이용하여 나머지 칼럼 그룹화**

multicol_sum = df.groupby([‘species’, ‘petal_width’]).sum()



		sepal_length	sepal_width	petal_length
species	**petal_width**			
setosa	**0.2**	24.3	16.4	7.0
versicolor	**1.3**	5.5	2.3	4.0
	1.4	7.0	3.2	4.7
	1.5	19.8	9.1	14.0
virginica	**1.8**	6.3	2.9	5.6
	1.9	5.8	2.7	5.1
	2.1	7.1	3.0	5.9
	2.2	6.5	3.0	5.8
	2.5	6.3	3.3	6.0

2. Apply

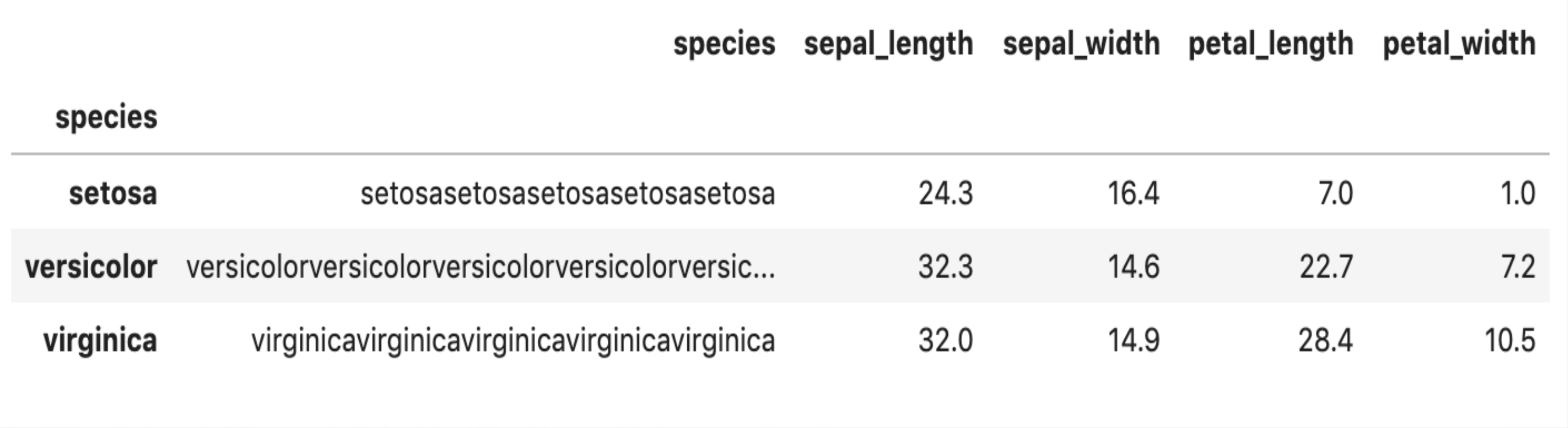


.groupby() returns an iterator object GroubBy

Apply example

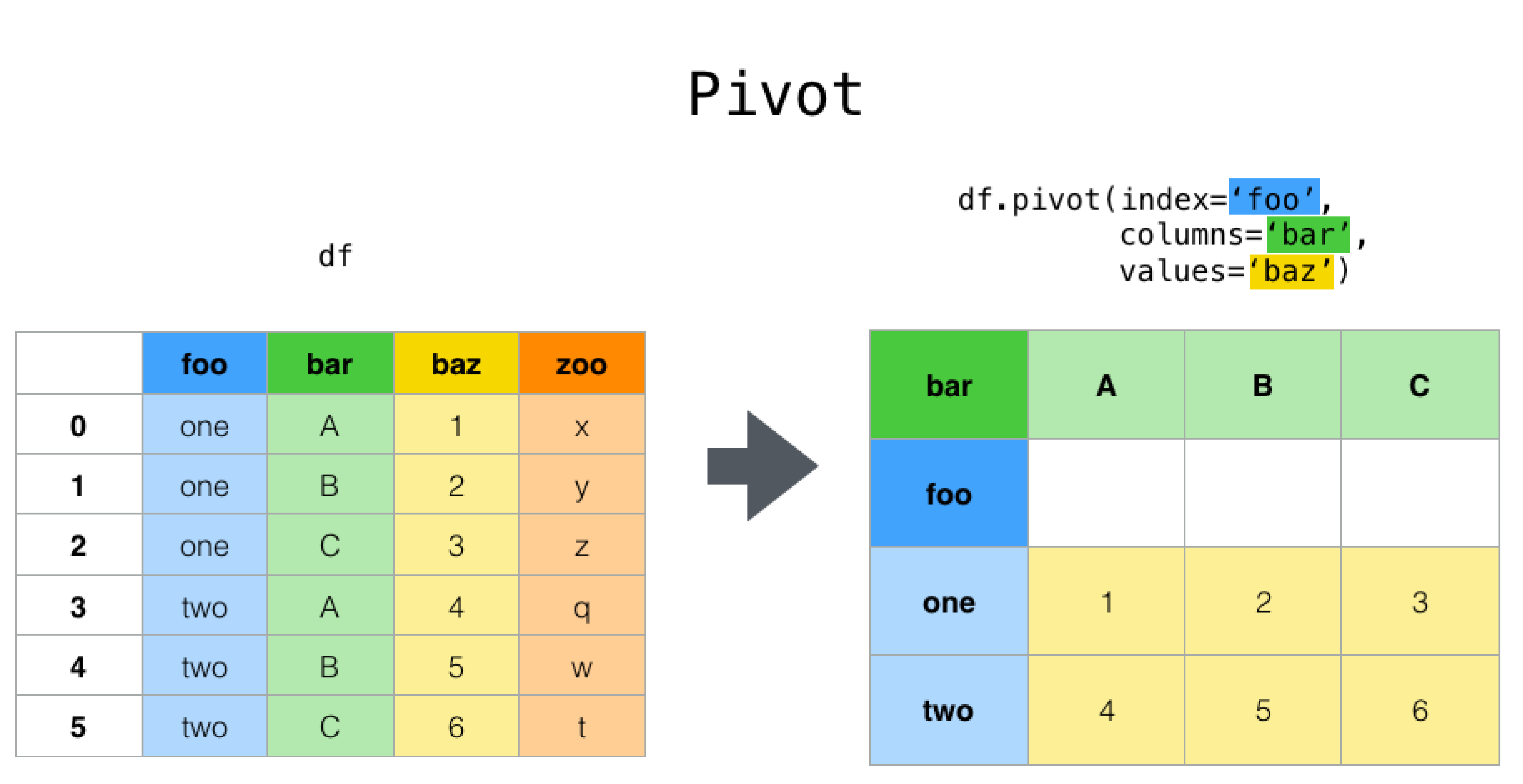
```
# 사용자 함수 정의
def my_sum(gr):
        return gr.sum()

df.groupby(‘species’).apply(my_sum)
```



	species	sepal_length	sepal_width	petal_length	petal_width
species					
setosa	setosasetosasetosasetosasetosa	24.3	16.4	7.0	1.0
versicolor	versicolorversicolorversicolorversicolorversic...	32.3	14.6	22.7	7.2
virginica	virginicavirginicavirginicavirginicavirginica	32.0	14.9	28.4	10.5

3. Pivot table



Pivot

df

	foo	bar	baz	zoo
0	one	A	1	x
1	one	B	2	y
2	one	C	3	z
3	two	A	4	q
4	two	B	5	w
5	two	C	6	t

```
df.pivot(index='foo',
         columns='bar',
         values='baz')
```

bar	A	B	C
foo			
one	1	2	3
two	4	5	6

Pivot table example

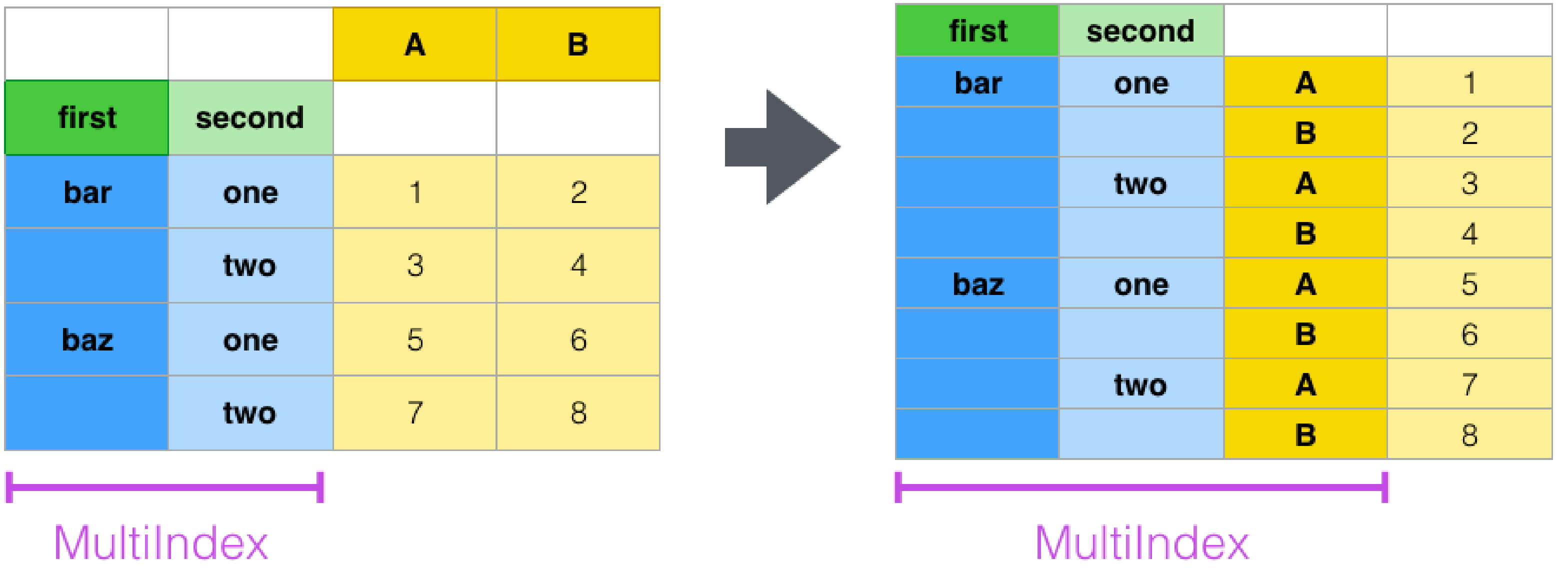
```
In [1]: df
Out[1]:
         date variable     value
0  2000-01-03        A  0.469112
1  2000-01-04        A -0.282863
2  2000-01-05        A -1.509059
3  2000-01-03        B -1.135632
4  2000-01-04        B  1.212112
5  2000-01-05        B -0.173215
6  2000-01-03        C  0.119209
7  2000-01-04        C -1.044236
8  2000-01-05        C -0.861849
9  2000-01-03        D -2.104569
10 2000-01-04        D -0.494929
11 2000-01-05        D  1.071804
```

```
pd.pivot_table(df, index='date', columns='variable', values='value')
variable            A         B         C         D
date
2000-01-03  0.469112 -1.135632  0.119209 -2.104569
2000-01-04 -0.282863  1.212112 -1.044236 -0.494929
2000-01-05 -1.509059 -0.173215 -0.861849  1.071804
```

Stack

`df2`

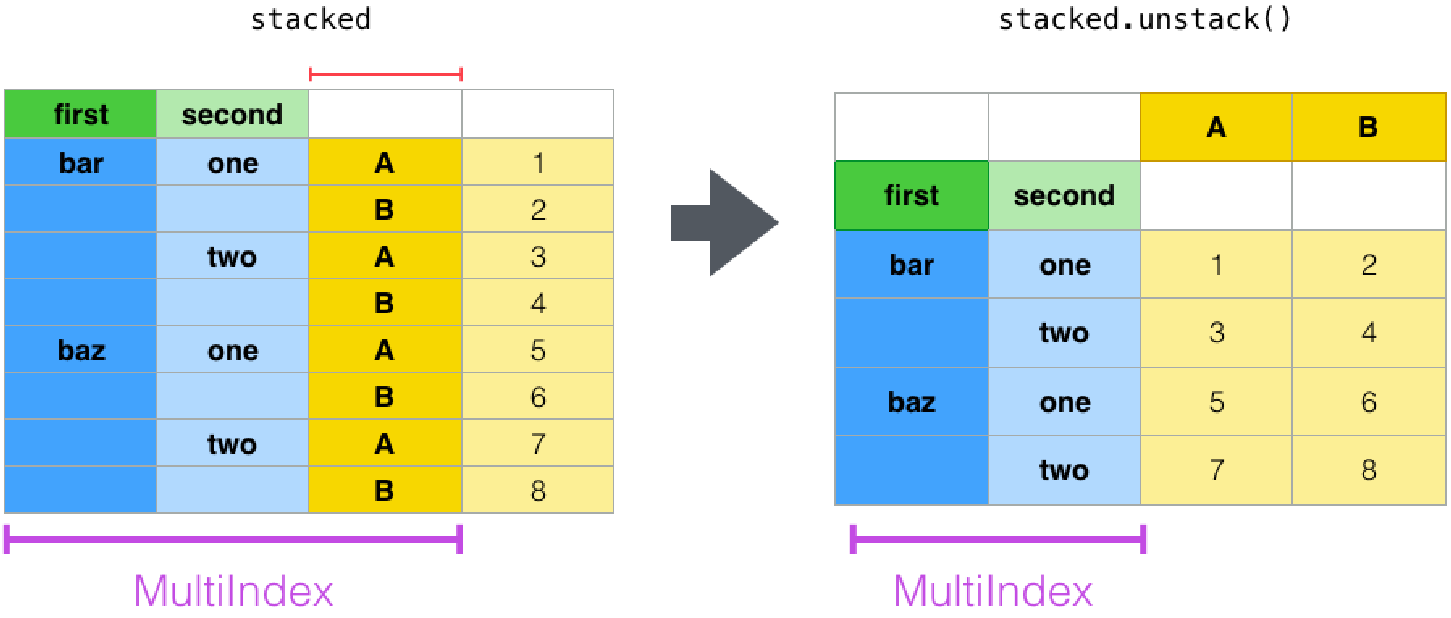
`stacked = df2.stack()`



		A	B
first	second		
bar	one	1	2
	two	3	4
baz	one	5	6
	two	7	8

first	second		
bar	one	A	1
		B	2
	two	A	3
		B	4
baz	one	A	5
		B	6
	two	A	7
		B	8

Unstack



stacked

first	second		
bar	one	A	1
		B	2
	two	A	3
		B	4
baz	one	A	5
		B	6
	two	A	7
		B	8

stacked.unstack()

		A	B
first	second		
bar	one	1	2
	two	3	4
baz	one	5	6
	two	7	8