In [1]:

```
import pandas as pd
```

1. 행을 열로 보내기(row -> column)

```
df.pivot(index = , columns = , values = )
```

- 행으로 이동하여 각 **row**의 인덱스로 남을 행 **= index**
- 열로 이동하여 각 **column**이 될 행 **= column**
- 행, 열에 모두 들어가지 않고, 데이터프레임 내 값으로 존재할 항목 **= values**

In [2]:

```
# 샘플 데이터
df = pd.DataFrame({
    'item' : ['shirts', 'shirts', 'shirts', 'shirts', 'shirts', 'pants', 'pants', 'pants
', 'pants'],
    'color' : ['white', 'white', 'white', 'black', 'black', 'white', 'white', 'black', '
black'],
    'size' : ['small', 'large', 'large', 'small', 'small', 'large', 'small', 'small', 'l
arge'],
    'sale' : [1, 2, 2, 3, 3, 4, 5, 6, 7],
    'inventory' : [2, 4, 5, 5, 6, 6, 8, 9, 9]
    })
df
```

Out[2]:

	item	color	size	sale	inventory
0	shirts	white	small	1	2
1	shirts	white	large	2	4
2	shirts	white	large	2	5
3	shirts	black	small	3	5
4	shirts	black	small	3	6
5	pants	white	large	4	6
6	pants	white	small	5	8
7	pants	black	small	6	9
8	pants	black	large	7	9

In [3]:

```
# item, size별 재고 합계
df.pivot_table(index='item', columns='size', values = 'inventory', aggfunc = 'sum')
```

Out[3]:

size	large	small
item		
pants	15	17
shirts	9	13

In [4]:

```
# [item, color], size별 재고 합계
df.pivot_table(index = ['item', 'color'], columns = 'size', values = 'inventory', aggfun
c = 'sum')
```

Out[4]:

item	color	size	large	small
pants	black		9.0	9.0
	white		6.0	8.0
shirts	black		NaN	11.0
	white		9.0	2.0

In [6]:

```
# null 값은 0으로 처리
df.pivot_table(index = ['item', 'color'], columns = 'size', values = 'inventory', aggfun
c = 'sum', fill_value = 0)
```

Out[6]:

item	color	size	large	small
pants	black		9	9
	white		6	8
shirts	black		0	11
	white		9	2

In [7]:

```
# ['item', 'color'], size별 판매, 재고 합계
df.pivot_table(index = ['item', 'color'], columns = 'size', values = ['sale', 'inventory
'], aggfunc = 'sum', fill_value=0)
```

Out[7]:

		inventory		sale	
	size	large	small	large	small
item	color				
pants	black	9	9	7	6
	white	6	8	4	5
shirts	black	0	11	0	6
	white	9	2	4	1

타이타닉호 성별, 객실 등급별 생존분석

In [8]:

```
df = pd.read_csv('./data/titanic.csv')
df.head()
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
									STON/O2			

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [10]:

```
df_titanic = df[['Survived', 'Pclass', 'Sex', 'Age', 'Embarked']]
df_titanic = df_titanic.dropna()
df_titanic.head()
```

Out[10]:

	Survived	Pclass	Sex	Age	Embarked
0	0	3	male	22.0	S
1	1	1	female	38.0	C
2	1	3	female	26.0	S
3	1	1	female	35.0	S
4	0	3	male	35.0	S

In [11]:

```
len(df_titanic)
```

Out[11]:

```
1044
```

2.1. 성별, 객실 등급별 승선자 수

- **count**

In [13]:

```
# 인덱스 : 성별 , columns = 객실 등급별
df_titanic.pivot_table(index = 'Sex', columns = 'Pclass', values = 'Survived', aggfunc =
'count', margins = 'True' )
# margins : 각 인덱스별 총합까지 보여준다.
```

Out[13]:

Pclass	1	2	3	All
Sex				
female	131	103	152	386
male	151	158	349	658
All	282	261	501	1044

2.2. 성별, 객실 등급별 생존자 수

- **sum**
- 모든 **csv** 파일에서 이렇게 쓰는 게 아니라
- **Survived** 중 생존은 **1,** 사망은 **0**이기 때문에 **sum** 하면 **1**을 가진 데이터만 더해져 생존자 수만 나온다.
- 이 데이터프레임만 가능한 방법

In [16]:

```
# count 를 sum으로 바꾸면, values에서 1만 더해지므로 총 생존자 수를 구할 수 있다.
```

```
df_titanic.pivot_table(index = 'Sex', columns = 'Pclass', values = 'Survived', aggfunc = 'sum', margins = 'True' )
```

Out[16]:

Pclass	1	2	3	All
Sex				
female	128	97	97	322
male	40	15	38	93
All	168	112	135	415

2.3. 성별, 객실 등급별 생존율

- mean(default)

In [17]:

```
df_titanic.pivot_table(index = 'Sex', columns = 'Pclass', values = 'Survived', aggfunc = 'mean', margins = 'True' )
```

Out[17]:

Pclass	1	2	3	All
Sex				
female	0.977099	0.941748	0.638158	0.834197
male	0.264901	0.094937	0.108883	0.141337
All	0.595745	0.429119	0.269461	0.397510