In [2]:

```
import pandas as pd
```

그룹화하여 그룹별 데이터 집계하기

`df.groupby(그룹화 기준 컬럼).통계적용컬럼.통계함수()`

In [3]:

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

Out[3]:

	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
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 [4]:

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

Out[4]:

	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 [5]:

```
len(df)
```

Out[5]:

```
1044
```

1. 그룹의 통계값 계산하기

- `df.groupby( 그룹기준 컬럼 ).통계적용컬럼.통계함수()`
- **count() :** 누락값을 제외한 데이터 수
- **size() :** 누락값을 포함한 데이터 수
- **mean() :** 평균
- **sum() :** 합계

- **std() :** 표준편차
- **min() :** 최소값
- **max() :** 최대값
- **sum() :** 전체 합

1.1. 객실 등급별 생존 통계

In [6]:

```
df.columns
```

Out[6]:

```
Index(['Survived', 'Pclass', 'Sex', 'Age', 'Embarked'], dtype='object')
```

In [11]:

```
# 객실 등급( Pclass )별 탑승자 수를 구한 결과
df.groupby('Pclass').Survived.count()
```

Out[11]:

```
Pclass
1    282
2    261
3    501
Name: Survived, dtype: int64
```

In [12]:

```
# 객실 등급( Pclass )별 탑승자 수를 구한 결과를 데이터프레임 df1으로 만들기
df1 = df.groupby('Pclass').Survived.count().to_frame()
df1
```

Out[12]:

Pclass	Survived
1	282
2	261
3	501

In [13]:

```
# 객실 등급( Pclass )별 생존자 수를 구한 결과를 데이터프레임 df2으로 만들기
df2 = df.groupby('Pclass').Survived.sum().to_frame()
df2
```

Out[13]:

Pclass	Survived
1	168
2	112
3	135

In [14]:

```
# 객실 등급( Pclass )별 생존율 수를 구한 결과를 데이터프레임 df3으로 만들기
df3 = df.groupby('Pclass').Survived.mean().to_frame()
df3
```

Out[14]:

	Survived
Pclass	
1	**0.595745**
2	**0.429119**
3	**0.269461**

In [17]:

```
# 객실 등급( Pclass )별 탑승자수, 생존자 수, 생존율 결과를 데이터프레임 df4으로 만들기 - concat
df4 = pd.concat([df1, df2, df3], axis = 1) # 인덱스 기준으로 concat
df4.columns = ['승선자 수', '생존자 수', '생존율']
df4
```

Out[17]:

	승선자 수	생존자 수	생존율
Pclass			
1	**282**	**168**	**0.595745**
2	**261**	**112**	**0.429119**
3	**501**	**135**	**0.269461**

1.2. 성별 생존 통계

In [18]:

```
# 성별 승선자 수 데이터프레임을 df5로 만들기
df5 = df.groupby('Sex').Survived.count().to_frame()
df5
```

Out[18]:

	Survived
Sex	
female	**386**
male	**658**

In [19]:

```
# 성별 생존자 수 데이터프레임을 df6로 만들기
df6 = df.groupby('Sex').Survived.sum().to_frame()
df6
```

Out[19]:

	Survived
Sex	
female	**322**
male	**93**

In [20]:

```
# 성별 생존율 데이터프레임을 df5로 만들기
df7 = df.groupby('Sex').Survived.mean().to_frame()
df7
```

Out[20]:

	Survived

Sex	Survived
female	0.834197
male	0.141337

In [21]:

```
# 성별 탑승자 수, 생존자 수, 생존율 데이터프레임을 df8로 만들기
df8 = pd.concat([df5, df6, df7], axis = 1)
df8.columns = ['성별 탑승자 수', '성별 생존자 수', '성별 생존율']
df8
```

Out[21]:

Sex	성별 탑승자 수	성별 생존자 수	성별 생존율
female	386	322	0.834197
male	658	93	0.141337

1.3. 성별, 객실 등급별 생존 통계

In [22]:

```
# 성별, 객실등급별 생존율
df.groupby(['Sex', 'Pclass']).Survived.mean().to_frame()
```

Out[22]:

		Survived
Sex	Pclass	
female	1	0.977099
	2	0.941748
	3	0.638158
male	1	0.264901
	2	0.094937
	3	0.108883

2. 그룹에 사용자 정의 함수 적용하기

- `df.gropyby(그룹 기준 컬럼).통계적용컬럼.agg(사용자정의함수, 매개변수들)`

In [23]:

```
def my_mean(values):
    return sum(values) / len(values)
```

In [24]:

```
df.groupby(['Sex', 'Pclass']).Survived.agg(my_mean)
# 매개변수 없으면 0번째 매개변수로 본인이 들어간다.
```

Out[24]:

```
Sex     Pclass
female  1         0.977099
        2         0.941748
        3         0.638158
male    1         0.264901
        2         0.094937
```

```
3          0.108883
Name: Survived, dtype: float64
```

3. 그룹 오브젝트 출력하기

``` **df.groupby(그룹기준컬럼).groups --> 그룹별 인덱스:[데이터리스트] 출력 df.groupby(그룹기준컬럼).get_group(그룹인덱스) --> 그룹별 인덱스에 해당하는 데이터프레임 출력**

In [25]:

```
df20 = df[:20] # 20개만 가져와서 사용해 보자.
df20.head()
```

Out[25]:

| | 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 [26]:

```
len(df20)
```

Out[26]:

```
20
```

In [28]:

```
# Pclass 그룹별 인덱스
df20.groupby('Pclass').groups
```

Out[28]:

```
{1: [1, 3, 6, 11], 2: [9, 15, 20, 21], 3: [0, 2, 4, 7, 8, 10, 12, 13, 14, 16, 18, 22]}
```

In [29]:

```
# Pclass 그룹별 인덱스( 1등급 )
df20.groupby('Pclass').get_group(1)
```

Out[29]:

	Survived	Pclass	Sex	Age	Embarked
1	1	1	female	38.0	C
3	1	1	female	35.0	S
6	0	1	male	54.0	S
11	1	1	female	58.0	S

In [30]:

```
# Pclass 그룹별 인덱스( 2등급 )
df20.groupby('Pclass').get_group(2)
```

Out[30]:

	Survived	Pclass	Sex	Age	Embarked
9	1	2	female	14.0	C
15	1	2	female	55.0	S
```

	Survived	Pclass	Sex	Age	Embarked
20	0	2	male	35.0	S
21	1	2	male	34.0	S

In [31]:

```
# Pclass 그룹별 인덱스( 3등급 )
df20.groupby('Pclass').get_group(3)
```

Out[31]:

	Survived	Pclass	Sex	Age	Embarked
0	0	3	male	22.0	S
2	1	3	female	26.0	S
4	0	3	male	35.0	S
7	0	3	male	2.0	S
8	1	3	female	27.0	S
10	1	3	female	4.0	S
12	0	3	male	20.0	S
13	0	3	male	39.0	S
14	0	3	female	14.0	S
16	0	3	male	2.0	Q
18	0	3	female	31.0	S
22	1	3	female	15.0	Q