

## [데이터프레임과 시리즈 (Pandas)-II]

### 8. 데이터 그룹화 및 집계

(1) **Group by**: 열(column)값을 이용하여 그룹화한다. (단일 열 / 다중 열 모두 가능)

ex) iris데이터셋의 데이터를, 종별, 꽃받침 길이 별로 그룹화하여 평균 구하기

```
iris.groupby([ iris["Species"], iris["Sepal.Length"] ]).mean() # 다중열의 경우, 그룹화할 열을 리스트로
```

(2) **레이블(원핫인코딩)**

ex) `le = LabelEncoder()` # 객체 생성

```
iris_df['species'] = le.fit_transform(iris_df['Species']) # Species열의 데이터값들을 컴퓨터가 인식할 수 있도록 함
```

### 9. 데이터 구조 변경

(1) 와이드 포맷 / 롱 포맷

1) 와이드 포맷(wide format): 열 단위의 데이터 구조 ex) pivot table

2) 롱 포맷(long format): 행 단위 데이터 구조 es)unpivot table

(2) **melt()**를 이용한 언피벗팅: `df.melt(id_vars= ['기준열...'], var_name='변수', value_name='값')`

(3) **pivot\_table()**을 이용한 피벗팅: `df.pivot_table(index=['기준열..'], ...)`

### 10. 데이터프레임에 함수 적용시키기

(1) **apply()**: 행 또는 열 단위로 함수 적용 / 데이터프레임, 시리즈 모두 적용 가능

ex) 데이터 프레임 df의 열별 합계 출력 `df.apply = (np.sum, axis=0)`

(2) **applymap()**: 각 요소별로 함수 적용 / 데이터프레임에서만 적용 가능

ex) 데이터 프레임 X의 각 요소값들을 제공하여 출력 `X.applymap( lambda x : x**2)`

(3) **map()**: 시리즈 타입의 벡터에만 적용 가능

### 11. 결측치나 특정값 일괄 변경하기

(1) **fillna()** : 결측치를 어떤 특정값으로 변경

(2) **replace(to\_value, new\_value, inplace=False)**: to\_value를 new\_value로 변경하여 반환만 함

(3) **where()**: 조건을 만족하는 요소는 그대로 출력 ( $\leftrightarrow$  **mask()**)