

## WEEK 5. 워싱턴의 집 값 예측하기

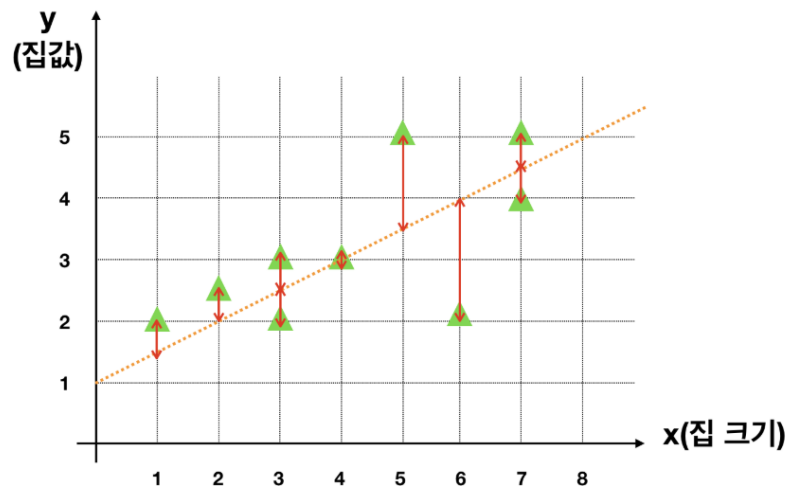
이성민

# 1. Linear Regression

1 선형 회귀 : 종속변수  $y$ 와 독립변수  $x$ 의 상관관계가 식으로 설명되는 회귀분석

2 최적의 식 : 오차가 가장 적은 식  
(거리 제곱의 합)

3 수많은 식을 그려보고 오차가 가장 적은 식 선택



```
for (x, y) in data:
```

```
    predict = a * x + b
```

```
    sums = sums + (y - predict) * (y - predict)
```


## 2. Visualization

■ 필요 없는 값 지우기 : `df.drop( [ '지우고 싶은 값' ], axis=1)`


■ 상관계수 : 변수 간의 관련성 수치화 (관련성 있을수록 1에 가깝다)

피어슨 상관계수 : 두 변수의 선형적 상관관계 표현

양의 상관관계 그래프의 x 축이 커질 수록 y 축 데이터도 커지는 모양

  $+0.7 \sim +1.0 \Rightarrow$  강한 양적 선형관계  
 $+0.3 \sim +0.7 \Rightarrow$  뚜렷한 양적 선형관계  
 $+0.1 \sim +0.3 \Rightarrow$  약한 양적 선형관계

음의 상관관계 그래프의 x 축이 커질 수록 y 축 데이터가 작아지는 모양

  $-1.0 \sim -0.7 \Rightarrow$  강한 음적 선형관계  
 $-0.7 \sim -0.3 \Rightarrow$  뚜렷한 음적 선형관계  
 $-0.3 \sim -0.1 \Rightarrow$  약한 음적 선형관계

$-0.1 \sim +0.1 \Rightarrow$  거의 무시될 수 있는 선형관계

## 2. Visualization

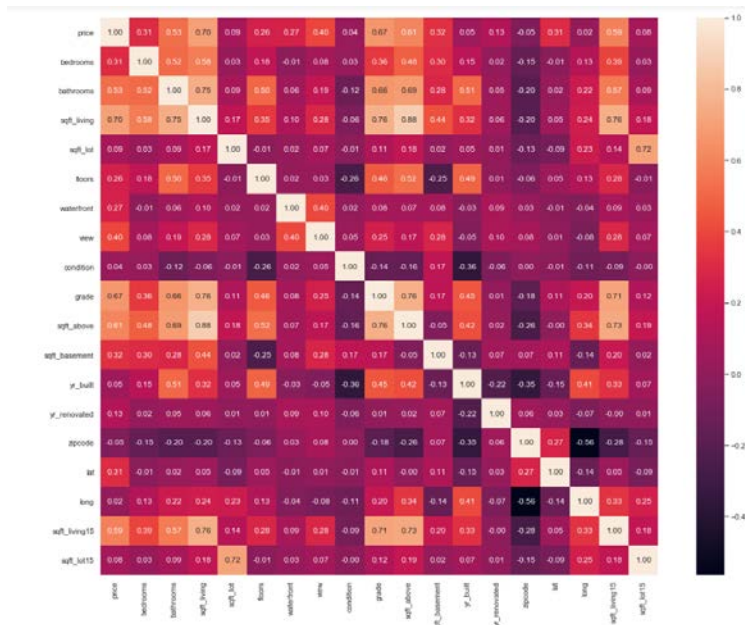
### ■ 히트맵으로 시각화하기

```
plt.figure(figsize=(20, 15))  
sns.heatmap(house_data.corr(), annot=True, fmt='.2f', square=True)  
plt.show()
```

figsize : 맵 크기 설정

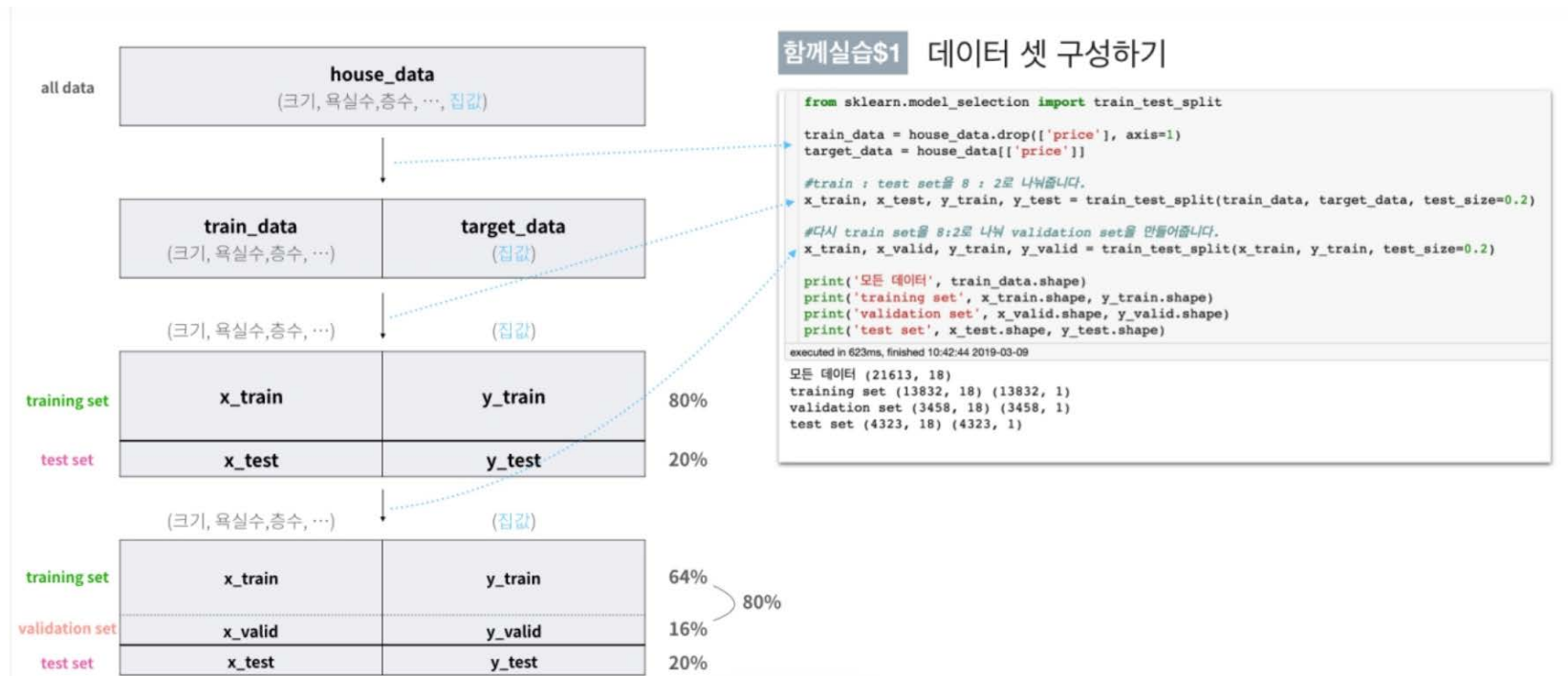
annot : 맵 안의 숫자 유무

fmt : 소수점 뒤의 숫자



### 3. Scikit-learning 으로 Linear Regression 구현

#### ■ Data set 구성



`x_train, x_test, y_train, y_test =`  
`train_test_split(train_data, target_data, test_size=0.2)`

### 3. Scikit-learning 으로 Linear Regression 구현

#### ■ Polynomial Regression

다항식 회귀 : 선형 관계가 아닌 곡선 형태의 그래프를 가질 때

```
from sklearn.preprocessing import PolynomialFeatures
```

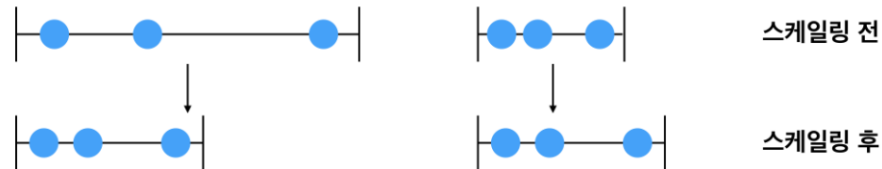
```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(PolynomialFeatures(1),  
                      LinearRegression()).fit(x_train, y_train)
```

## 4. Feature Scaling과 성능 개선 (+)

### Feature Scaling

특정 값의 범위를 균일하게 맞춰주는 작업



```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
```

```
xs_train = scaler.fit_transform(x_train)
```