빅데이터 분석처리 과정
회귀 분석

1. 단순선형회귀

본 수업의 내용

- 머신러닝의 지도 학습 방식을 이해한다
- 선형 회귀 이론을 이해하고 파이썬을 이용한 프로그래밍을 구현한다
- 사이킷런을 사용한 단순 선형 회귀 분석을 구현하고 예측을 수행할 수 있다.
- 스탯츠모델을 사용한 단순 선형 회귀 분석을 구현하고 예측을 수행할 수 있다.

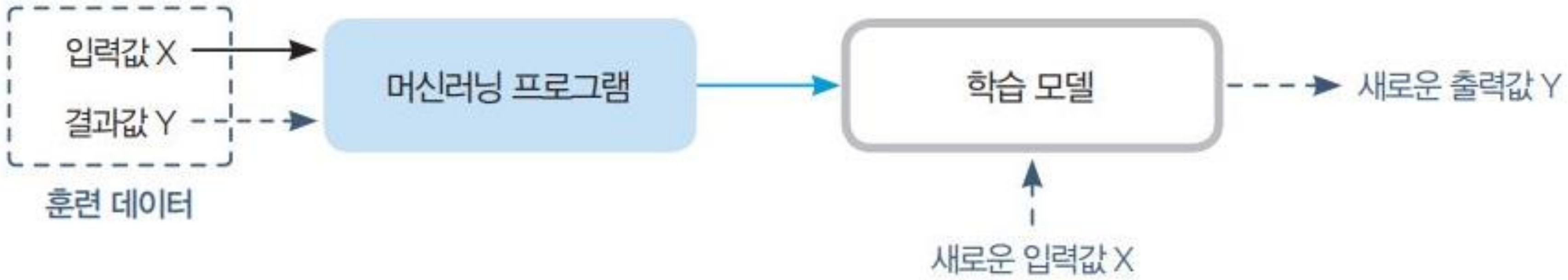
1. 단순선형회귀
Simple Linear Regresson

- 회귀 분석의 개념
- 회귀 분석 유형
- 단순 선형 회귀 절차
- 사이킷런 실습
- 스탯츠 모델 실습

<복습> 머신러닝Machine Learning

- 지도 학습supervised learning:
  - <u>이미 결과를 알고 있는 데이터로 모델을 훈련하고, 이후에 아직 결과를 모르는 데이터에 적용하는 프로세스</u>
  - 학습을 하기 위한 훈련 데이터training data에 입력과 출력을 같이 제공하므로 문제(입력)에 대한 답(출력, 결과값)을 아는 상태에서 학습하는 방식.
  - 입력: 예측변수predict variable, 속성, 특징feature
  - 출력: 반응변수response variable, 목표변수target variable, 클래스class, 레이블label
  - 대표적 유형: 회귀, 분류

- 머신러닝 지도 학습 방식



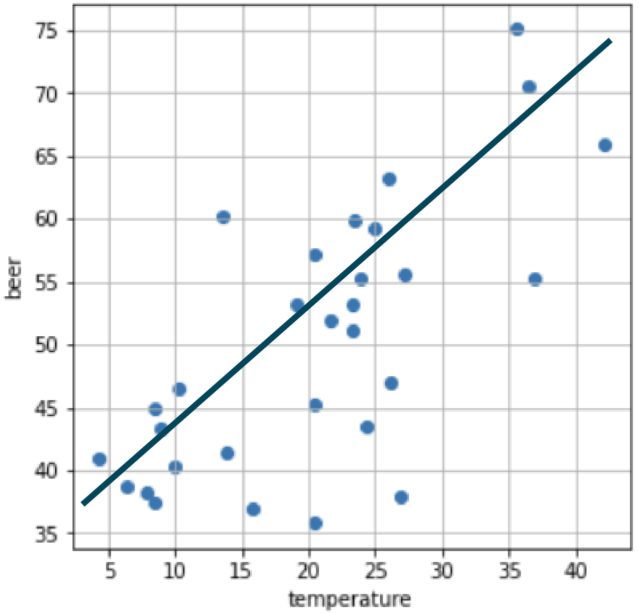
<출처: 데이터과학기반의 파이썬 빅데이터 분석, p308, 한빛아카데미>

회귀Regression

- 회귀Regression이란?
  - 시초: 19C 통계학자 프랜시스 골턴Francis Galton, 키가 큰 사람의 아이가 부모보다 더 크지 않다는 사실을 관찰하고, “평균으로 회귀한다 " 는 표현을 사용
  - 데이터의 값은 평균과 같은 기존의 경향으로 돌아가려는 경향이 있다는 것
  - 여러 변수들 간의 상관 관계를 파악하여 어떤 특정 변수의 값을 다른 변수들의 값을 이용하여 설명, 예측하는 수리식을 찾는 방법
  - 회귀식을 찾는 것

- 예제
  - 날씨(온도)와 맥주의 매상 데이터 관계
  - 근속 연수와 연봉 데이터 관계
  - 집의 면적(크기)와 집값 데이터 관계
  - 나이와 키 관계

온도	매상
20.5	45.3
25	59.3
10	40.4
26.9	38
15.8	37
4.2	40.9
…	…



회귀Regression 분석 유형

- 회귀 분석의 유형
  - 변수의 개수 및 계수의 형태에 따라 구분한다.

  - 독립 변수의 개수에 따라
    - 단순회귀분석: 독립변수가 1개인 경우, 단일 회귀분석이라고도 함.
    - 다중회귀분석: 독립변수가 여러 개인 경우

  - 회귀 계수의 형태에 따라
    - 선형: 계수를 선형 결합으로 표현할 수 있는 경우
    - 비선형: 계수를 선형 결합으로 표현할 수 없는 경우

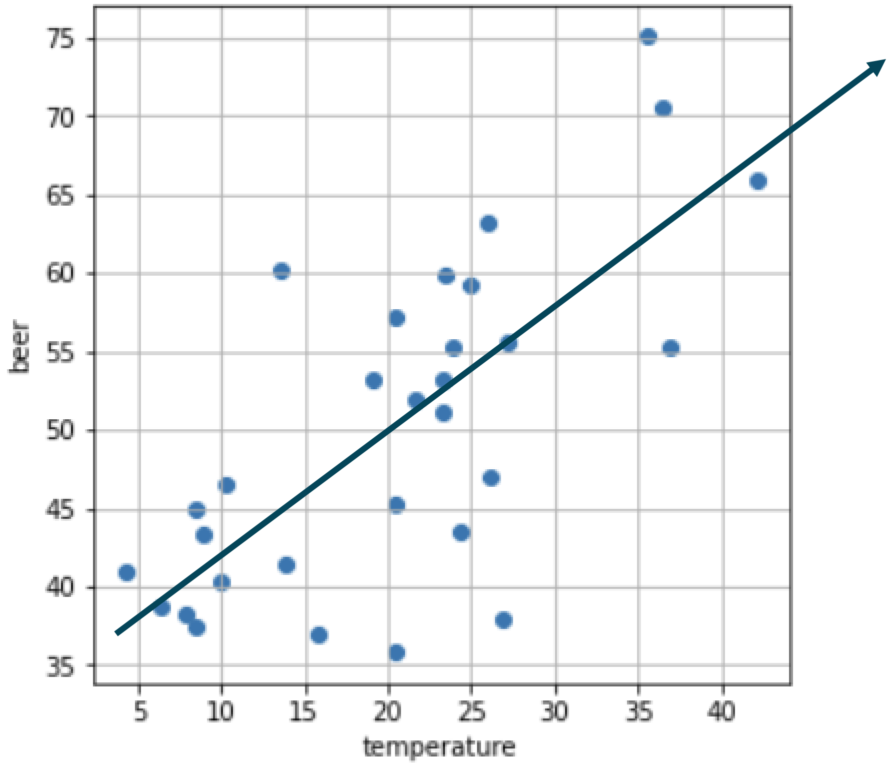
  - 종속 변수의 개수에 따라
    - 단변량univariate 회귀모델 – 종속변수가 1개인 경우
    - 다변량multivaiate 회귀모델 – 종속변수가 여러 개인 경우

선형 회귀

- 선형 회귀란?
  - 종속 변수가 독립 변수와 회귀 계수의 선형 조합으로 표현 가능한 경우
  - 파라미터(계수)에 대한 선형성만을 가정함
- 독립변수, X와 종속 변수, y간의 상호 연관성 정도를 파악하기 위한 분석 기법
- 하나의 변수가 변함에 따라 대응되는 변수가 어떻게 변하는지를 측정하는 것
- 변수 간의 인과관계를 분석할 때 많이 사용
- 독립 변수가 1개이면 단순 회귀 분석, 두 개 이상이면 다중 회귀 분석
- 선형 회귀식: $y = \omega_0 + \omega_1 x_1 + \omega_2.x_2. + \omega_3 x_3 + \cdots + \omega_{\boldsymbol{n}}.x_n$

1. 단순 선형 회귀에 대한 이해 – 회귀식, 계수, 예측값, 잔차(오차)
2. 오차와 손실함수(Cost function)
3. 모델 파라미터 최적화 - 최소제곱법

단순 선형 회귀 Simple Linear Regression



- **단순선형회귀(Simple Linear Regression)**

- 데이터의 분포를 직선으로 가정하고 데이터를 학습해서 **예측**
- 독립변수가 1개이고, 종속변수도 1개
- 독립변수와 종속변수 간의 관계를 선형적으로 파악하는 회귀 방식
- **독립변수와 종속변수의 관계식** : $y(x) = \omega_1 x + \omega_0$

- **회귀 계수(coefficient) :** ω_1

- 독립변수가 종속변수에 끼치는 영향력(설명력)의 정도로서, 직선의 기울기(slope)

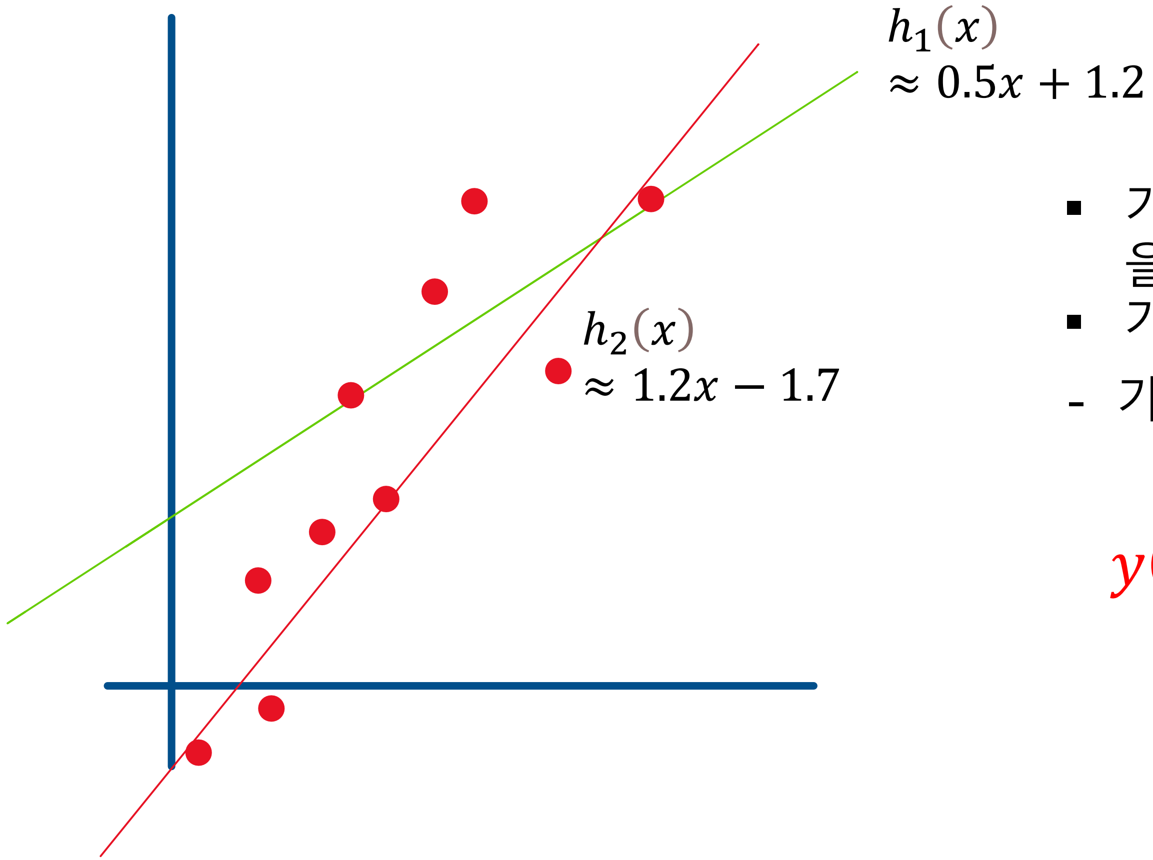
- **절편(intercept):** ω_0

- 독립변수가 0일 때 상수값

※ 이 직선이 데이터에 부합하려면, ω_0 와 ω_1 을 어떻게 정해야 할까?(데이터를 설명하는 가장 적절한 기울기와 절편을 어떻게 찾아야 할까?)

오차와 손실함수Cost Function

- 학습모델(회귀식)과 회귀 계수

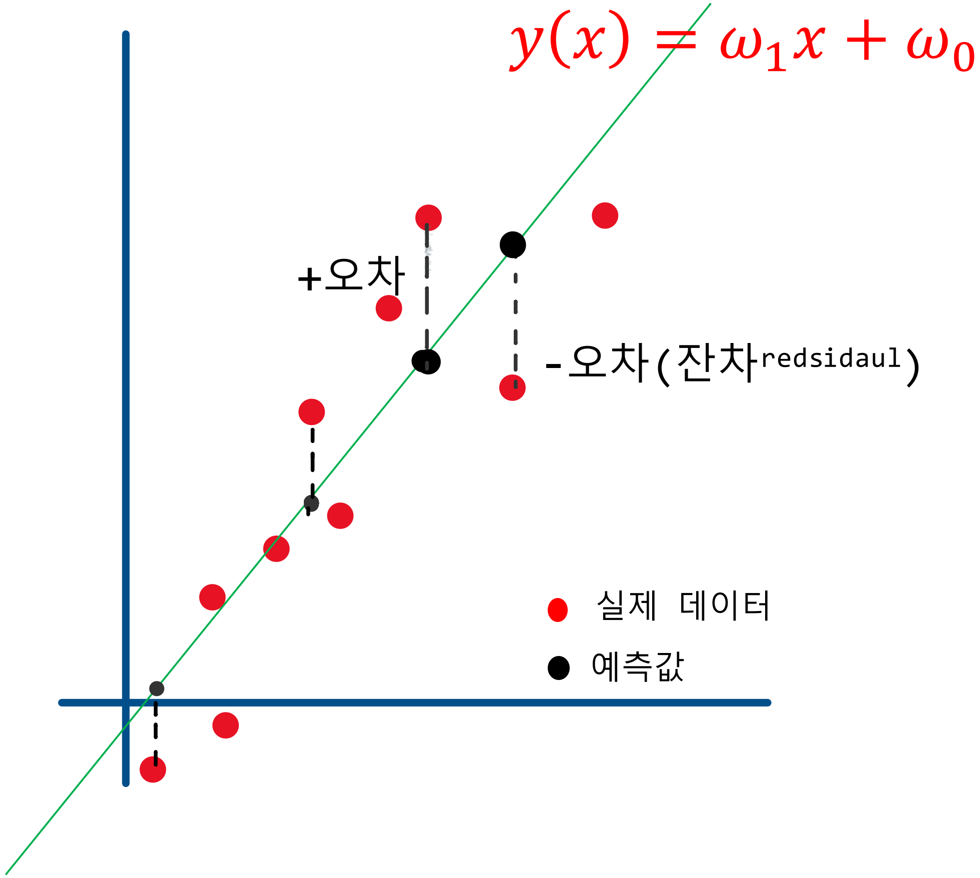


- 기울기와 절편의 값에 따라 여러 개의 직선이 있을 수 있음
- 가장 데이터를 잘 설명하는 직선의 방정식을 추정

\- 가장 적절한 파라미터를 추정

$$y(x) = \omega_1 x + \omega_0$$

잔차 제곱 오차 함수Residual Sum of Squares



- **잔차(residual)/오차**
  - 실제 값과 회귀 모델식에 의해 계산한 추정값(점추정치)의 차이
  - 잔차 값의 작을수록, 구해진 회귀식이 데이터를 잘 설명하고 있다고 볼 수 있다.
  - x_n에서의 오차(점추정치)

- **잔차 제곱합(RSS: Resiual Sum of Squares)**
  - 모든 데이터 점의 잔차를 제곱하여 합한 것
  - RSS $=\sum(y_i - (\omega_1 x_i + \omega_0))^2$
    - x_i: 독립변수 집합 X의 원소
    - y_i: 종속변수 집합 Y의 원소

- **손실함수(loss function) / 비용함수(cost function)**
  - 회귀계수를 추정할 때 손실을 최소화할 목적으로 사용하는 함수

- 직선 모델이 “데이터에 부합”하려면?
  - 가장 적은 오류를 발생시키는 직선을 찾음
  - 손실함수값이 최소가 되는 회귀모형을 만듦

여러 가지 오차식

- 평균 제곱 오차 MSE(Mean Square Error)

  MSE = $\frac{\boldsymbol{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(y_i - \hat{y}_i)^2$

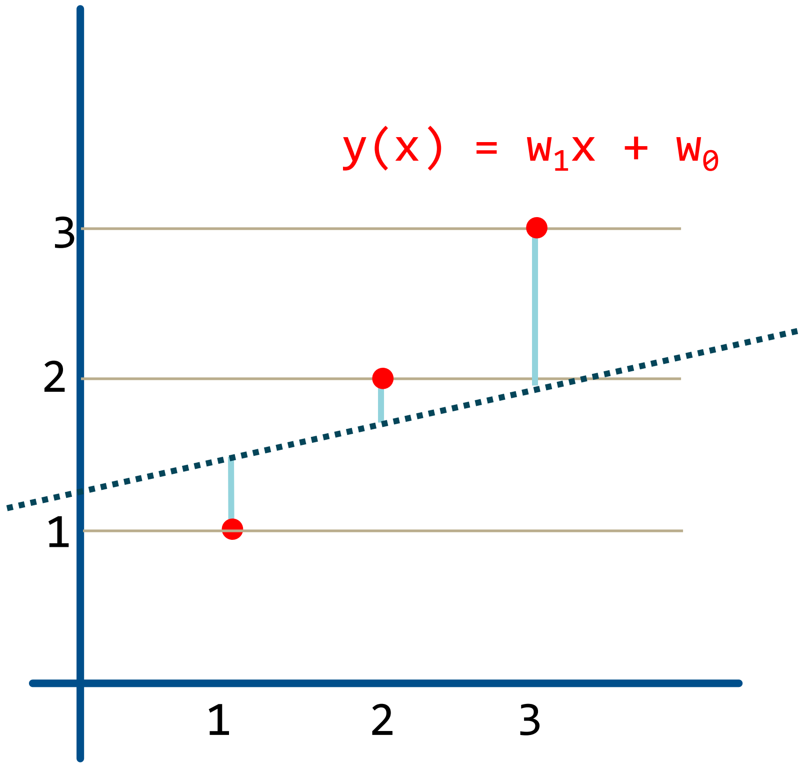
- 평균 절대값 오차 MAE(Mean Absolute Error)

  MAE = $\frac{\boldsymbol{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}|y_i - \hat{y}_i|$

- 평균 제곱근 오차 RMSE(Root Mean Square Error)

  RMSE = $\sqrt{\frac{\boldsymbol{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(y_i - \hat{y}_i)^2}$

회귀 분석 손실함수Cost Function



오차의 합: $\dfrac{(y_1-\hat{y}_1)^2+((y_2-\hat{y}_2)^2+(y_3-\hat{y}_3)^2}{3}$

$$\mathbf{cost} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\mathbf{cost}(w_1, w_0) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

회귀 계수 탐색(선형 방정식 찾기)

$$\mathbf{cost}(w_1, w_0) = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} (y_i - \hat{y}_i)^2$$

<데이터>

X	y
1	1
2	2
3	3

w_1의 값을 변경하며 손실 함수의 값을 최소로 하는 w_1을 탐색한다.

w_1 = -4, cost = $\frac{(1-(-4)*(1))^2+(2-(-4)*(2))^2+(3-(-4)*(3))^2}{3}$ = 116.66

w_1 = -3.5, cost = $\frac{(1-(-3.5)*(1))^2+(2-(-3.5)*(2))^2+(3-(-3.5)*(3))^2}{3}$ = 94.5

w_1 = -3, cost = $\frac{(1-(-3)*(1))^2+(2-(-3)*(2))^2+(3-(-3)*(3))^2}{3}$ = 74.66

$\approx$

w_1 = 0.5, cost = $\frac{(1-(0.5)*(1))^2+(2-(0.5)*(2))^2+(3-(0.5)*(3))^2}{3}$ = 1.16

w_1 = 1.0, cost = $\frac{(1-(1)*(1))^2+(2-(1)*(2))^2+(3-(1)*(3))^2}{3}$ = 0

$\approx$

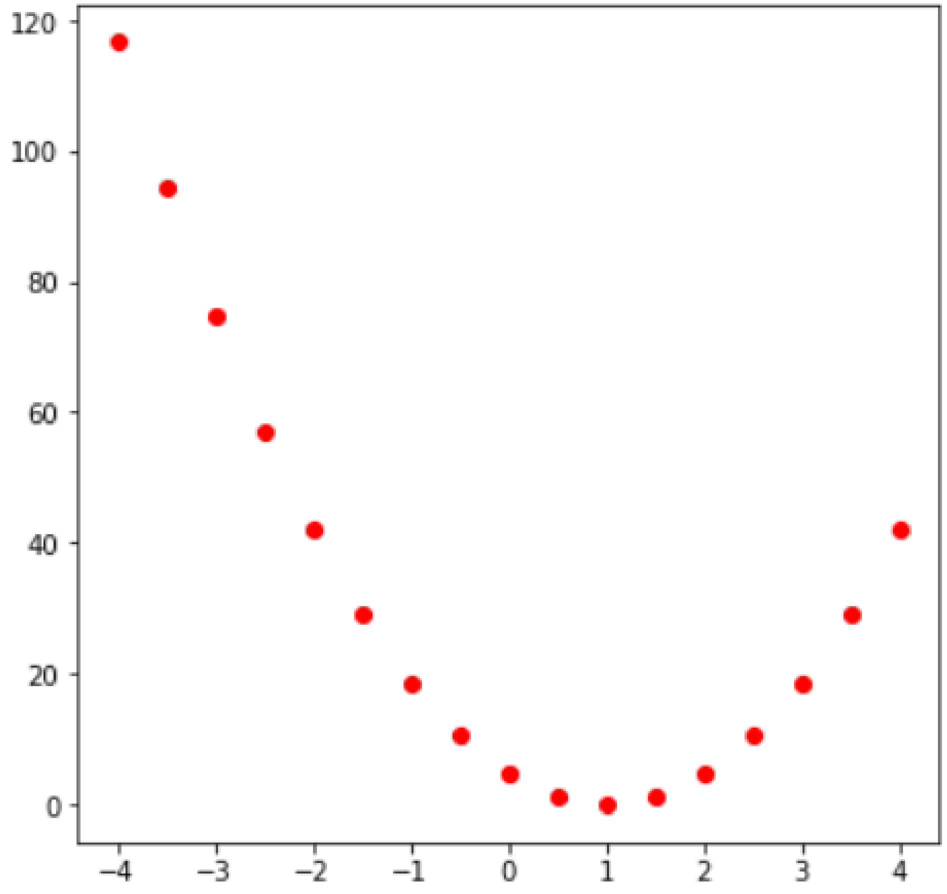
w_1 = 4, cost = $\frac{(1-(4)*(1))^2+(2-(4)*(2))^2+(3-(4)*(3))^2}{3}$ = 42.0

회귀 계수 탐색(선형 방정식 찾기)

- w_1 VS. 손실함수 값

```
[{-4.0: 116.66666666666667},
 {-3.5: 94.5},
 {-3.0: 74.66666666666667},
 {-2.5: 57.166666666666664},
 {-2.0: 42.0},
 {-1.5: 29.166666666666668},
 {-1.0: 18.666666666666668},
 {-0.5: 10.5},
 {0.0: 4.666666666666667},
 {0.5: 1.1666666666666667},
 {1.0: 0.0},
 {1.5: 1.1666666666666667},
 {2.0: 4.666666666666667},
 {2.5: 10.5},
 {3.0: 18.666666666666668},
 {3.5: 29.166666666666668},
 {4.0: 42.0}]
```

- 시각화



회귀 계수 탐색(선형 방정식 찾기) > 최소제곱법Ordinary Least Squares

- 손실함수로 잔차제곱합을 사용하여 손실을 최소로 하는 파라미터(회귀 계수)를 채용하는 방법
- 선형 회귀의 경우 실제로 파라미터 추정을 할 때는 오차의 기울기를 따라 최적의 파라미터를 찾아 가는 방법보다 “최소제곱법(OLS)”이 매우 효율적인 계산법으로 사용될 수 있음.

▪ **최소제곱법(Ordinary least squares 혹은 linear least squares) / 정규 방정식(Normal Equation)**

- 학습에 사용되는 각 데이터 인스턴스 $x^{(\mathrm{i})}$를 i번째 행으로 하는 행렬을 X라고 정의할 때 최적의 파라미터 (손실함수의 값이 최소가 되는) W는 다음과 같음

$$w = (X^T X)^{-1} X^T y$$

▪ sklearn의 최소제곱법 적용.

sklearn.linear_model.LinearRegression

class `sklearn.linear_model.LinearRegression`(**, fit_intercept=True, normalize='deprecated', copy_X=True, n_jobs=None, positive=False*) [source]

Ordinary least squares Linear Regression.

LinearRegression fits a linear model with coefficients w = (w1, ..., wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

선형 방정식 찾기

- 최적의 회귀 모형을 만든다는 것: $y = \omega_1 \boldsymbol{X} + \omega_0$
  - ✓ 손실함수(비용함수)인 MSE값이 최소가 되는 회귀 계수(ω_1, ω_0)**를 찾는 것**

1) 수치해적 방법: 경사하강법
   - 반복적인 계산에 의해 근사값을 구하는 수치 계산법

2) 해석학적 방법(최소제곱법:OLS$^{\text{Ordinary Least Sqaures}}$)
   - 직선 모델의 경우 근사값이 아니라 정규방정식$^{\text{Normal Equation}}$이라는 닫힌 형태의 계산식을 직접 풀어 정확한 해를 구하는 방법(Ax=B, $A^TAx = A^TB$) – 가우스/르장드르
   - 반복계산이 아니라 1회의 계산으로 최적의 W를 구할 수 있다.
   - 계산 시간이 빠르고 정확한 답을 제공한다
   - 문제의 본질을 잘 이해할 수 있고, 다차원 데이터에 대응하며 곡선모델로 확장하기 좋다
   - w_0와 w_1으로 손실 함수(단순선형회귀)를 각각 편미분한 값이 0이 되는 연립방정식의 해를 구한다.
   - 특성의 개수(입력의 차원)가 많을수록 느려지고 다양한 문제에 적용할 수 없다

회귀분석 평가지표 > 여러 가지 오차식

- 평균 제곱 오차 MSE(Mean Square Error)

  $$\text{MSE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}}(y_i - \hat{y}_i)^2$$

- 평균 절대값 오차 MAE(Mean Absolute Error)

  $$\text{MAE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}}|y_i - \hat{y}_i|$$

- 평균 제곱근 오차 RMSE(Root Mean Square Error)

  $$\text{RMSE} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}}(y_i - \hat{y}_i)^2}$$

회귀분석 평가지표 > 결정 계수

- 결정 계수(Coefficient of Determinant)

$$R^2 = \frac{\text{예측값의 분산}}{\text{실제값의 분산}} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2} = 1 - \frac{RSS}{\sum(y_i - \bar{y})^2}$$

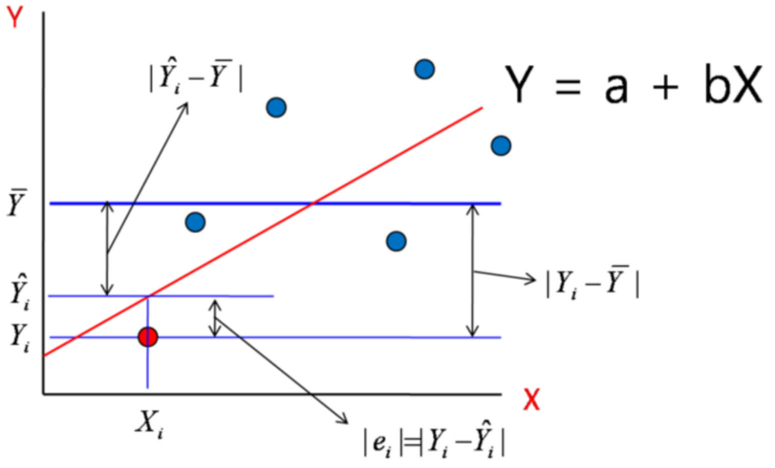
$\hat{y}_i$는 실제값 y_i에 대한 예측값, $\bar{y}$는 실제값들의 평균

- 모델이 얼마나 종속 변수를 잘 설명하는가에 관한 지표
- 만약 모델의 잔차가 매우 큰 경우 결정계수는 마이너스가 나올 수 있다
- 최소제곱법을 이용한 선형 회귀에서는 예측값($\hat{y}_i$)과 실제값(y_i)의 상관 계수를 제곱한 수치가 결정 계수와 같아진다.($0 \le R^2 \le 1$)

- 결정 계수 해석
  - 결정 계수의 값은 0 ≤ R2 ≤ 1이며, 1에 가까울수록 설명력이 강하고 0에 가까울수록 설명력이 약하다

결정 계수(Coefficient of Determinant)

- 결정계수 : 개체(종속 변수)가 가지는 **총 변량 중 회귀식이 설명(차지)하는 변량의 비율**



- $\bar{Y}$는 모든 Y값의 평균, 변수 Y의 대표값이라 할때
- 개체 Y가 가지는 총 변량:

$|Y_i - \bar{Y}|$ = 회귀식이 차지하는 변량($|\hat{Y}_i - \bar{Y}|$) + 회귀식으로 설명할 수 없는 변량($|Y_i - \hat{Y}_i|$)

결정 계수(r^2) = $\dfrac{\text{회귀식으로 설명 가능한 변량의 비율}}{\text{총 변량}}$

$$= \frac{\Sigma(\hat{Y}_i - \bar{Y})^2}{\sum(Y_i - \bar{Y})^2}\cdot$$

$$= \frac{\sum(Y_i - \bar{Y})^2 - (Y_i - \hat{Y}_i)^2}{\sum(Y_i - \bar{Y})^2}\cdot$$

$$= 1 - \frac{\Sigma(Y_i - \hat{Y}_i)^2}{\Sigma(Y_i - \bar{Y})^2}$$

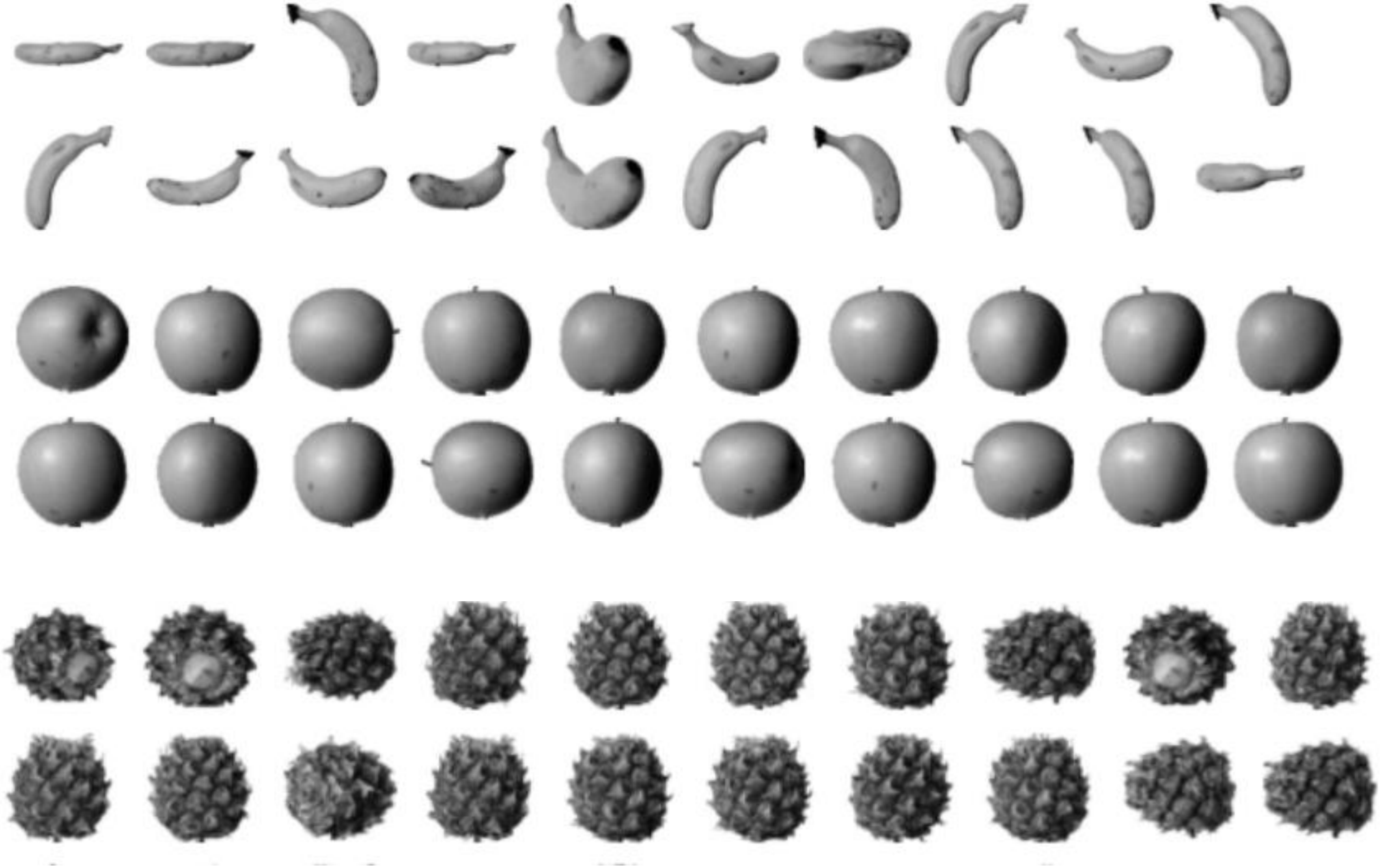
$$= 1 - \frac{Rss}{\Sigma(Y_i - \bar{Y})^2}$$

사이킷런의 회귀 분석 평가 지표

지표	모듈 및 함수	설명	식
MAE	metrics.mean_absolute_error	실제값과 예측값 차이의 절대값의 평균	MAE $= \frac{1}{n}\sum_{i=1}^{n} \lvert y_i - \hat{y}_i \rvert$
MSE	metrics.mean_squared_error	실제값과 예측값의 차이의 제곱들의 평균	MSE $= \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$
RMSE	math또는 numpy 모듈의sqrt	Root of MSE, MSE의 제곱근값	RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$
R^2	metrics.r2_score, 또는 LinearRegression의 score	결정 계수, 실제값의 분산 대비 예측값의 분산 비율	$R^2 = \frac{\text{예측값 분산}}{\text{실제값 분산}}$

- $\hat{y}_i$: 실제값 y_i에 대한 예측값

모델 평가 – train_score VS. test_score



모델의 일반화 성능 평가 용어

- 적합도와 예측 정확도
  - 적합도: 가지고 있는 데이터에 대해 모델을 적용했을 때 들어맞는 정도
  - 예측 정확도: 아직 얻지 못한 데이터에 대해 모델을 적용했을 때 들어맞는 정도
- 과적합(오버피팅) – 적합도는 높은데 예측 정확도는 낮은 경우
- 일반화 오차 – 아직 얻지 못한 데이터에 대한 예측 오차
- 훈련 데이터와 테스트 데이터(검증 데이터)
  - 훈련 데이터(트레이닝 데이터)
    - 파라미터(회귀 계수) 추정에 사용되는 데이터
    - 훈련데이터의 적합도를 평가하는 것으로 모델의 정확도(정밀도)는 구할 수 있지만 일반화 오차를 평가하는 것은 어렵다
  - 테스트 데이터(검증 데이터)
    - 일반화 오차를 추측하기 위해 파라미터 추정을 위한 학습을 할 때 사용하지 않고 남겨둔 데이터
    - 테스트 데이터로 모델의 정확도를 평가하는 것으로 일반화 오차를 어느 정도 추측할 수 있다
- 검증 방법
  - 홀드 아웃 검증
  - 교차 검증
    - 데이터를 일정한 규칙에 따라 훈련 데이터와 테스트 데이터로 나누어 테스트 데이터에 대한 예측 정확도를 평가하는 방법
    - 리브-p-아웃 교차 검증 / Kfold 교차검증

선형 회귀 이론 정리

- 머신 러닝의 개념으로 해석하는 선형회귀
- 가설함수: 최적의 회귀선 $y = \omega_0 + \omega_1 x$
- 손실함수: 예측값과 실제값과의 차이(오차)의 합(평균)
- 회귀 계수를 찾는 방법: 최소 제곱법, 경사하강법
- 회귀 모델 평가 지표
- 모델 성능 평가

단순 선형 회귀 실습(사이킷런, 스탯츠 모델)

- **회귀 모델 구축 후 성능 평가 및 예측**
- **기온(temperature) VS 맥주판매량 회귀 분석 예측**
- **나이(age)VS 키(height) 회귀 분석 예측**
- **응용: 보스턴 집값 예측**

- 기온(temperature) VS 맥주 판매량의 관계
- 예제:

기온에 따른 맥주 판매량 예측하기	
목표	단순선형회귀 분석에 의해 데이터베이스에 없는 기온에 대해 그날의 맥주 판매량을 예측한다.
핵심 개념	머신러닝 프로세스, 지도학습, 입력변수(독립변수), 목표변수(종속변수), 단순선형회귀 분석 회귀식, 잔차제곱법, 비용함수, 경사하강법
데이터수집	30쌍의 기온과 판매량으로 이루어진 데이터 세트
데이터 준비 및 탐색	그래프 그리기
분석모델구축	사이킷런의 선형 회귀 모델 구축

<출처: 파이썬으로 배우는 머신러닝교과서, p168, 한빛미디어>

단순선형회귀 머신러닝 구현(사이킷런)

- 사이킷런에서 최소제곱법(OLS: Ordinary Least Squares)으로 단순 선형 회귀 구현

```
1. 데이터 수집 및 탐색
```

```
2. 모델 클래스 선택
```

```
3. 모델 객체 생성
```

```
4. 모델에 사용할 특성 데이터셋 및 타깃 데이터셋 준비
```

```
5. 객체에 대해 학습 수행: fit()
```

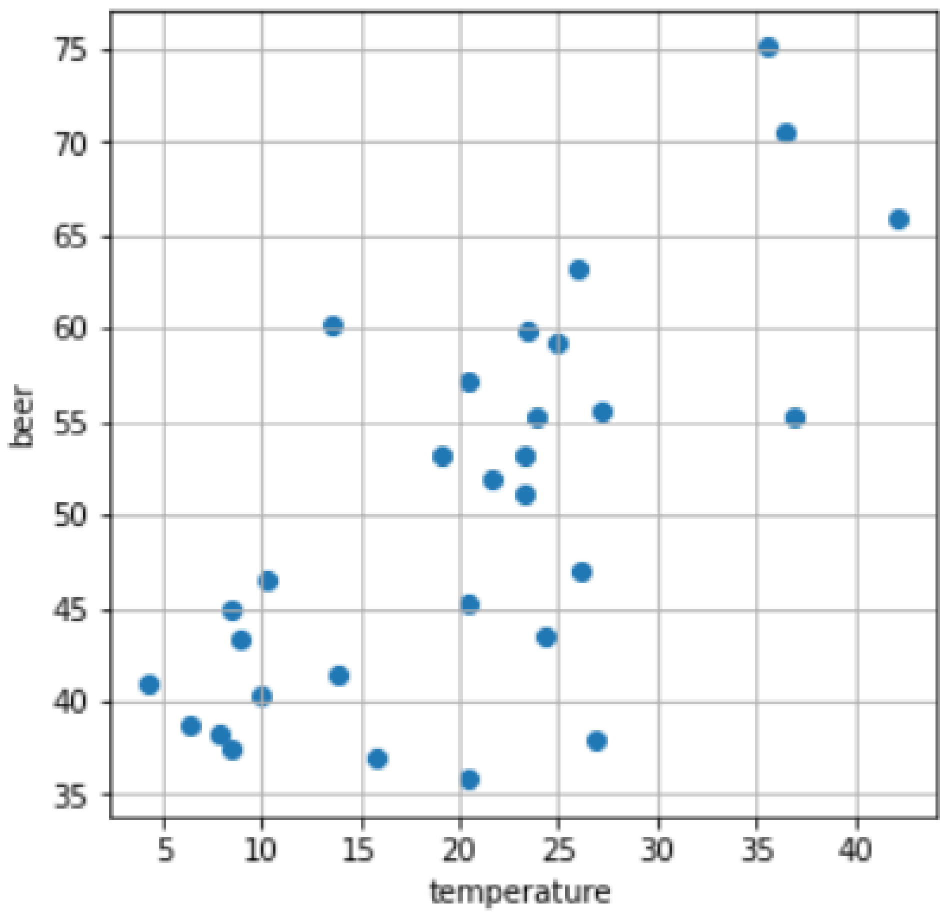
```
6. 실행 객체 또는 추정된 모델에 대해 예측 수행: predict()
```

```
7. 분석 결과를 평가: MSE, RMSE, R2(결정계수)
```

단순선형회귀 머신러닝 구현(사이킷런)

- 맥주 데이터를 활용한 단순 선형 회귀 실습

	beer	temperature
0	45.3	20.5
1	59.3	25.0
2	40.4	10.0
3	38.0	26.9
4	37.0	15.8
5	40.9	4.2
6	60.2	13.5
7	63.3	26.0
8	51.1	23.3
9	44.9	8.5
10	47.0	26.2
11	53.2	19.1
12	43.5	24.3



단순선형회귀 머신러닝 구현(사이킷런)

1. 데이터 수집 및 탐색

필요한 모듈 import

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

데이터 수집

```python
beer = pd.read_csv("./data/beer.csv")
beer
```

]:

	beer	temperature
0	45.3	20.5
1	59.3	25.0
2	40.4	10.0

단순선형회귀 머신러닝 구현(사이킷런)

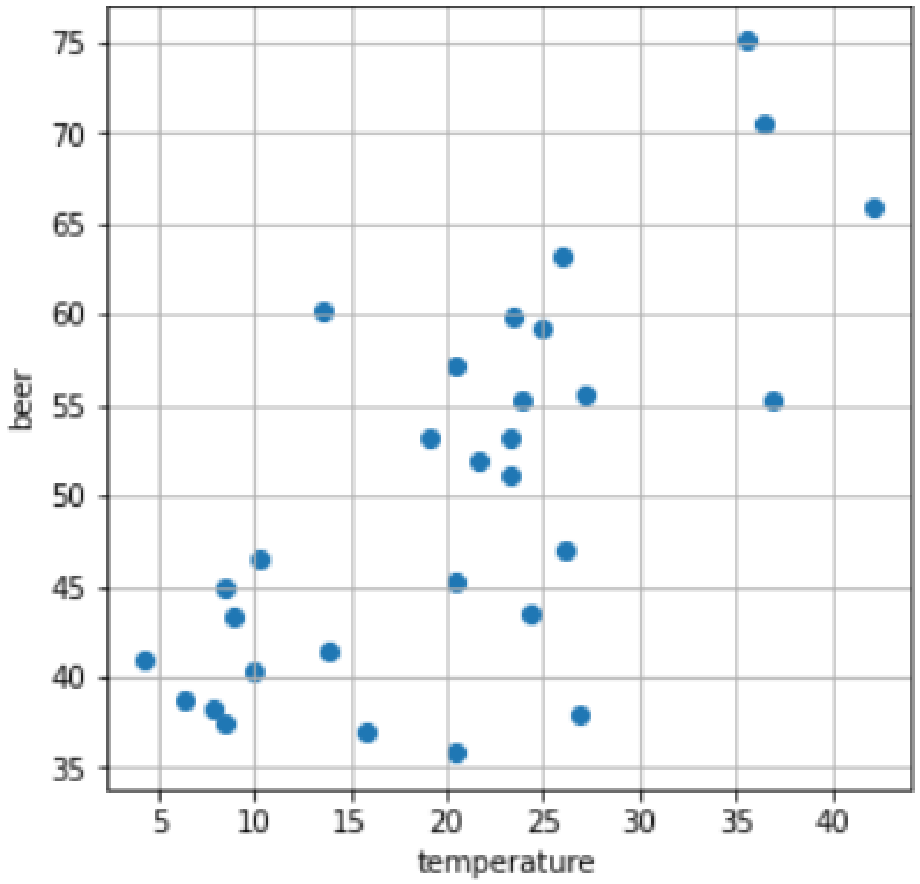
데이터 탐색

```python
plt.figure(figsize=(5,5))
plt.scatter(beer['temperature'], beer['beer'])
plt.xlabel('temperature')
plt.ylabel('beer')
plt.grid()
plt.show()
```

```python
# 독립 변수, 종속 변수 확인
beer['temperature']
```

```
0      20.5
1      25.0
2      10.0
3      26.9
```

```python
# 종속 변수 확인, 레코드 수 확인
print(beer['beer'])
print(len(beer))
```



단순선형회귀 머신러닝 구현(사이킷런)

2. 데이터 준비 및 분할

```python
# 전체 데이터 중 80%는 학습용, 20%는 검증용으로 분리
import numpy as np

#독립변수, 종속변수 데이터셋 준비
X = np.array(beer['temperature']).reshape(-1, 1)
y = beer['beer']

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(\
                                   X, y, test_size=0.2,\
                                              random_state=1)
```

- 독립변수의 특성이 1개 밖에 없더라도 각 값들은 2차원 리스트 또는 배열의 형태일 것

- 첫째 매개변수: 학습용 데이터의 독립변수 집합
- 둘째 매개변수: 학습용 데이터의 종속변수 집합

단순선형회귀 머신러닝 구현(사이킷런)

3. 선형 회귀 모델 객체 생성 - 관계를 모형화하기

```python
from sklearn.linear_model import LinearRegression

#모델 클래스 선택 후 인스턴스 객체 생성

lr = LinearRegression()
```

- 매개변수를 추가 설정할 수 있으나 대부분의 경우 필요하지 않다.
- fit_intercept : 절편 값을 계산할 것인지의 여부를 결정 한다. (기본값은 True)
- normalize : 회귀를 수행하기 전에 데이터를 정규화할 것인지의 여부를 결정한다. (기본값은 False)
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

단순선형회귀 머신러닝 구현(사이킷런)

4. 객체에 대해 학습 수행

- 선형 회귀를 수행할 객체에 대하여 fit() 메소드를 이용하여 학습을 수행
- 회귀식 구현: 회귀계수, 절편

```
# 학습 수행
reg = lr.fit(X_train, y_train)
```

- 첫 번째 매개변수: 훈련용 데이터 독립변수 집합
- 두 번째 매개변수: 검증용 데이터 종속변수 집합

```
# 계수 및 절편 확인: _속성은 학습을 통해 결정되는 속성

reg.coef_, reg.intercept_
```

```
(array([0.69705648]), 36.06666541566105)
```

```
#회귀식
print("y = {:2f}X + {:.3f}".format(reg.coef_[0], reg.intercept_))
```

```
y = 0.697056X + 36.067
```

- $y(x) = w_1 x + w_0$
- w_0 = reg.coef_, w_1 = reg.intercept_

단순선형회귀 머신러닝 구현(사이킷런)

5. 실행 객체 또는 추정된 모델에 대해 예측 수행: predict()

- 모델의 계수를 추정했으므로, predict() 메소드를 사용하면 예측이 가능

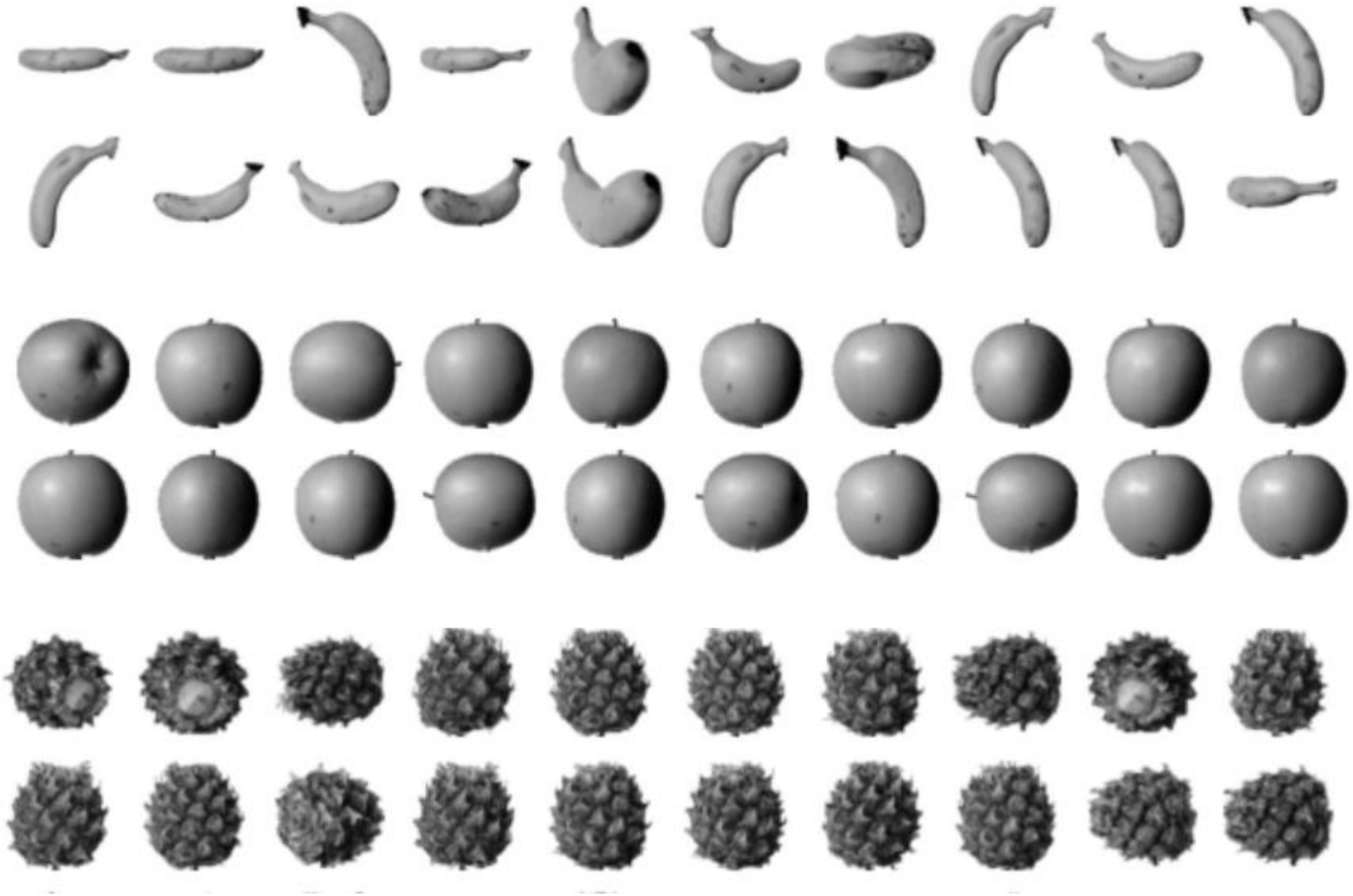
```
# 구축된 모델에서 예측 수행
y_pred = reg.predict(X_test)
print(np.round(y_pred, 2))
```

```
[60.81 50.36 54.33 50.36 41.92 43.18]
```

```
X_test
```

```
array([[35.5],
       [20.5],
       [26.2],
       [20.5],
       [ 8.4],
       [10.2]])
```

학습과 검증



단순선형회귀 머신러닝 구현(사이킷런)

6. 모델의 평가 : MSE, RMSE, 결정계수

- MSE: 사이킷런의 metrics모듈에 있는 mean_squared_error() 함수
- RMSE: MSE의 제곱근을 계산

```
from sklearn.metrics import mean_squared_error, r2_score

# MSE
mse = mean_squared_error(y_test, y_pred)

#RMSE
rmse = np.sqrt(mse)

print("MSE:", np.round(mse, 3))
print("RMSE: ", np.round(rmse, 3))
```

```
MSE: 91.347
RMSE:  9.558
```

[참고] 사이킷런 회귀 추정기의 score() method: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

reg.score(X_test, y_test)

reg.score(X_train, y_train)

단순선형회귀 머신러닝 구현(사이킷런)

6. 모델의 평가 : MSE, RMSE, 결정계수

- 결정계수(R^2) : 가지고 있는 데이터에 대해 모델을 적용했을 때 적합도를 평가

metrics 모듈의 r2_score() 함수를 호출하거나 회귀 객체의 score() 메소드를 호출

$$R^2 = \frac{\text{예측값의 분산}}{\text{실제값의 분산}} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2} = 1 - \frac{RSS}{\sum(y_i - \bar{y})^2}$$

$\hat{y}_i$는 실제값 y_i에 대한 예측값, $\bar{y}$는 실제값들의 평균

- 결정 계수의 값은 $0 \le R2 \le 1$

- 1에 가까울수록 설명력이 강하고 0에 가까울수록 설명력이 약함

```python
#결정계수 R2
r2 = r2_score(y_test, y_pred)
print("R2: ", np.round(r2, 3))
```

```
R2:  0.485
```

단순선형회귀 머신러닝 구현(사이킷런)

7. 모델의 시각화: 회귀식 그래프 시각화

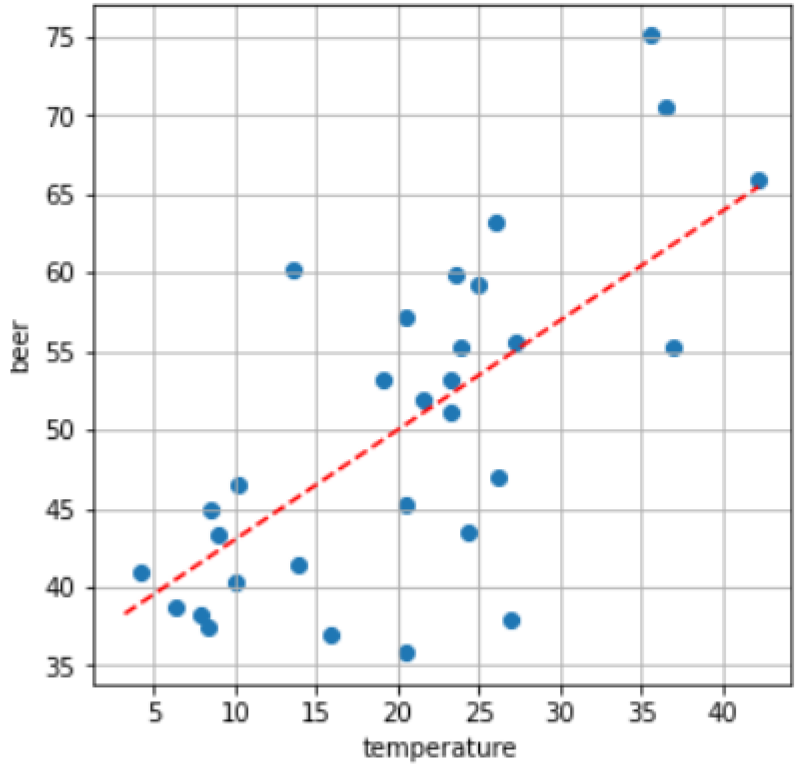
```python
#y = 0.697056X + 36.06 시각화

plt.figure(figsize=(5,5))
xx = np.arange(beer['temperature'].min() - 1,\
               beer['temperature'].max() + 1 )
yy = reg.predict(xx.reshape(len(xx), 1))

plt.plot(xx, yy, linestyle='--',color='red')

# 수집한 데이터셋 시각화
plt.scatter(beer['temperature'], beer['beer'])

plt.xlabel('temperature')
plt.ylabel('beer')
plt.grid()
plt.show()
```



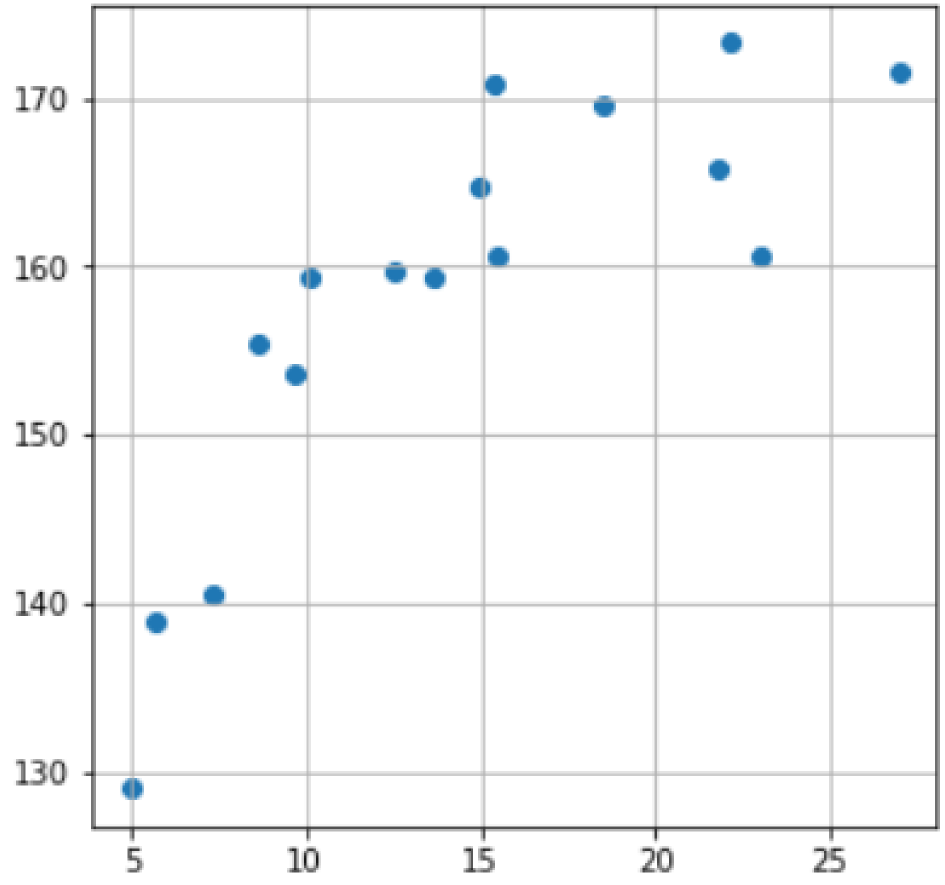
[응용1] 나이(age) VS 키(height) 회귀분석

- 16명의 데이터 셋 준비

```
data_df = pd.read_csv("./data/age_height.csv", )
data_df
```

	age(X)	height(T)
0	15.425550	170.910131
1	23.008112	160.675599
2	5.002859	129.002066
3	12.558314	159.701396
4	8.668897	155.460589
5	7.308465	140.561344
6	9.656505	153.654664
7	13.639018	159.429396
8	14.919187	164.704239
9	18.470418	169.645276
10	15.479863	160.712575
11	22.130488	173.287099
12	10.111306	159.311932

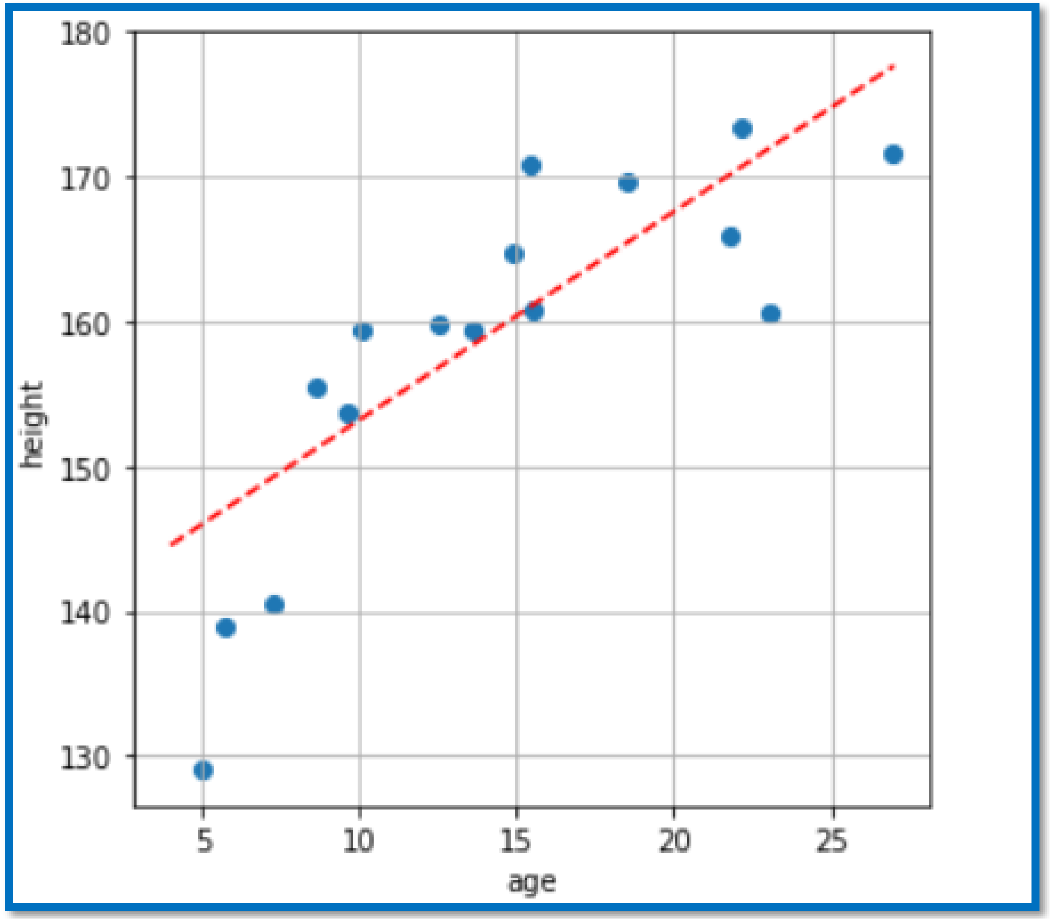
- 데이터 셋 분포 시각화



[응용] 나이(age) VS 키(height) 회귀분석

- 결과

```
MSE: 83.495
RMSE:  9.138
coef:  [1.43863478]
intercetp:  138.78419382830361
R2:  0.663
y = 1.438635X + 138.784
```



단순선형회귀 머신러닝 구현(스탯츠모델Statsmodel)

- 알고리즘: 최소제곱법(OLS: Ordinary Least Squares)
- 스탯츠모델에서 최소제곱법으로 단순 선형 회귀 구현

1. 데이터 수집 및 탐색

2. 데이터 준비

3. 선형회귀 모델 객체 생성

4. 객체에 대해 학습 수행: fit()

5. 실행 객체 또는 추정된 모델에 대해 예측 수행: predict()

6. 분석 결과를 평가: MSE, RMSE, R2(결정계수), p-value,f-통계량

추정을 얼마나 확신할 수 있나

단순선형회귀 머신러닝 구현(스탯츠모델)

- 데이터 준비

```python
#스탯츠모델 import
import statsmodels.api as sm

# 맥주 데이터 생성
beer = pd.read_csv("./data/beer.csv")

import numpy as np

#독립변수, 종속변수 데이터셋 준비
X = np.array(beer['temperature']).reshape(-1, 1)
y = beer['beer']

# 전체 데이터 중 80%는 학습용, 20%는 검증용으로 분리
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(\
                                    X, y, test_size=0.2,\
                                           random_state=1)

X_train = sm.add_constant(X_train)
```

- 상수항 결합으로 독립변수에 데이터를 추가

단순선형회귀 머신러닝 구현(스탯츠모델)

- 선형 회귀 모델 객체 생성

  - sm 모듈에 있는 OLS()을 이용하여 OLS 방법으로 선형 회귀를 수행할 수 있는 객체를 생성

```
# 스탯츠모델
lr = sm.OLS(y_train, X_train)
```

- 학습용 데이터를 객체 생성 시 전달

- 모델 객체에 대한 학습 수행

  - 선형 회귀를 수행할 객체에 대하여 fit() 메소드를 이용하여 학습을 수행
  - 회귀식 구현: 회귀계수, 절편

```
reg = lr.fit()
```

- 학습용 데이터를 객체 생성 시 이미 전달했기 때문에 fit() 메소드에 전달하지 않음.

단순선형회귀 머신러닝 구현(스탯츠모델)

- 모델의 평가 : summary()함수를 사용하여 추정 결과를 표시함

reg.summary()

회귀계수: 0.697

절편: 36.066

Model

R2

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   beer   R-squared:                       0.507
Model:                            OLS   Adj. R-squared:                  0.485
Method:                 Least Squares   F-statistic:                     22.63
Date:                Thu, 01 Jul 2021   Prob (F-statistic):           9.50e-05
Time:                        08:17:45   Log-Likelihood:                -79.854
No. Observations:                  24   AIC:                             163.7
Df Residuals:                      22   BIC:                             166.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         36.0667      3.315     10.881      0.000      29.193      42.941
x1             0.6971      0.147      4.757      0.000       0.393       1.001
==============================================================================
Omnibus:                        1.282   Durbin-Watson:                   1.668
Prob(Omnibus):                  0.527   Jarque-Bera (JB):                0.363
Skew:                          -0.241   Prob(JB):                        0.834
Kurtosis:                       3.360   Cond. No.                         52.2
==============================================================================
```

[응용2] 농어 무게 예측

- 농어의 길이에 기반한 무게를 예측하는 선형 회귀 분석을 수행한다.

농어의 길이가 50CM 이라면 무게는 얼마일까?

[응용3] 보스턴 집값 회귀 분석

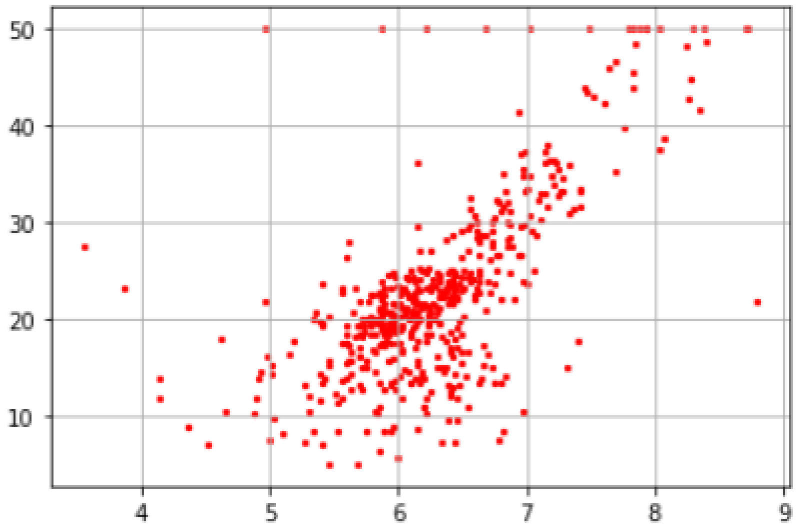
- 보스턴 집값 데이터셋 준비

```
data_df = pd.read_csv('./data/boston_room_price.csv')
data_df.head(10)
```

	RM	price
0	6.575	24.0
1	6.421	21.6
2	7.185	34.7
3	6.998	33.4
4	7.147	36.2
5	6.430	28.7
6	6.012	22.9
7	6.172	27.1
8	5.631	16.5
9	6.004	18.9

- 데이터 셋 분포 시각화

```
plt.scatter(data_df['RM'], data_df['price'], s=5, c='red')
plt.grid();
```

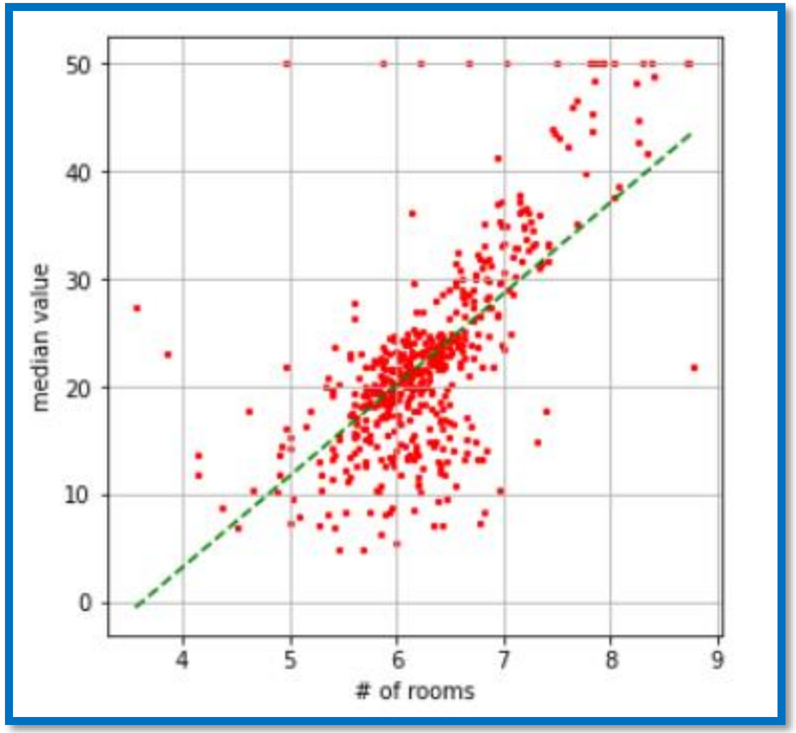


[응용] 보스턴 집값 회귀 분석

✓ **조건 : 학습 7, 검증 3**
✓ **seed = 1로 고정**

- 결과

```
MSE: 36.517
RMSE:  6.043
R2:  0.602
```



?