강원지역혁신플랫폼

기계학습

Machine Learning

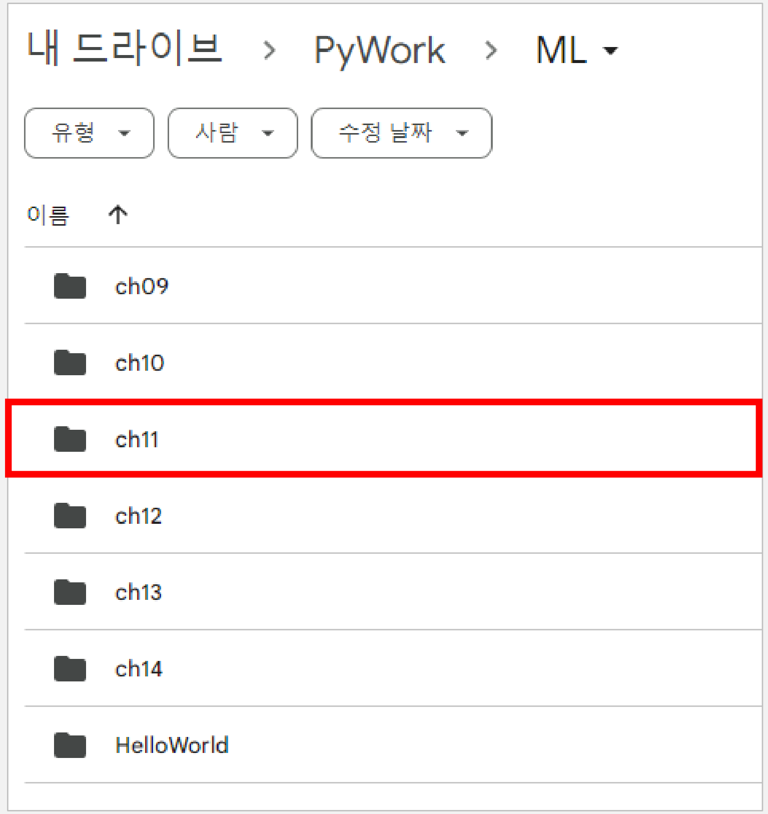
선형 회귀 분석 알고리즘

학습목표

선형회귀 알고리즘을 설명할 수 있습니다.

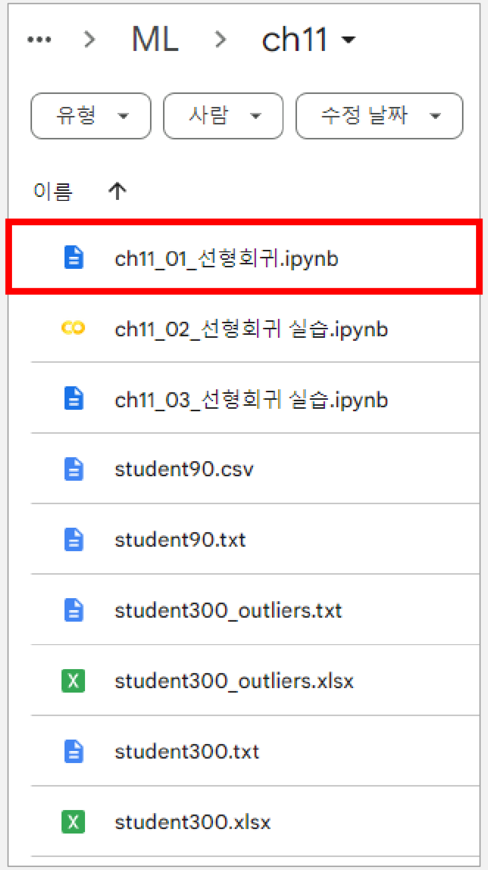
01 | 실습

- (권장) 아래와 같은 경로에 실행 소스가 존재하면 환경 구축 완료
  - 구글 드라이브 “PyWork > ML” 폴더로 이동함
    - 아래의 [ch11] 폴더를 클릭하면 됨



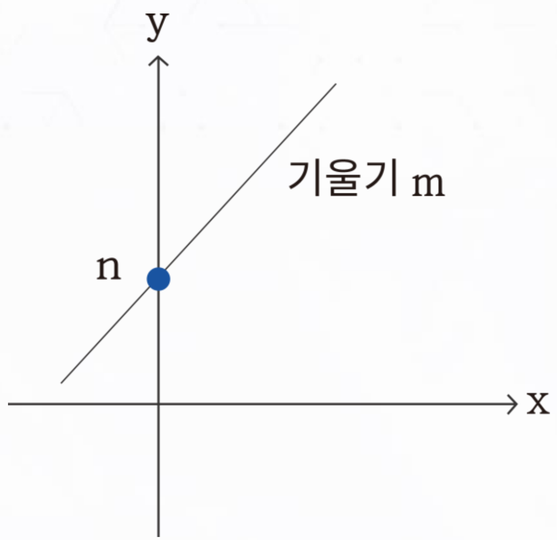
01 | 실습

- “ML > ch11 >” 폴더를 클릭함
  - 아래의 [ch11_01_선형회귀.ipynb] 스크립트를 클릭함



선형 회귀분석의 개념

- 기울기와 y절편이 주어졌을 때 직선의 방정식은 다음과 같다.

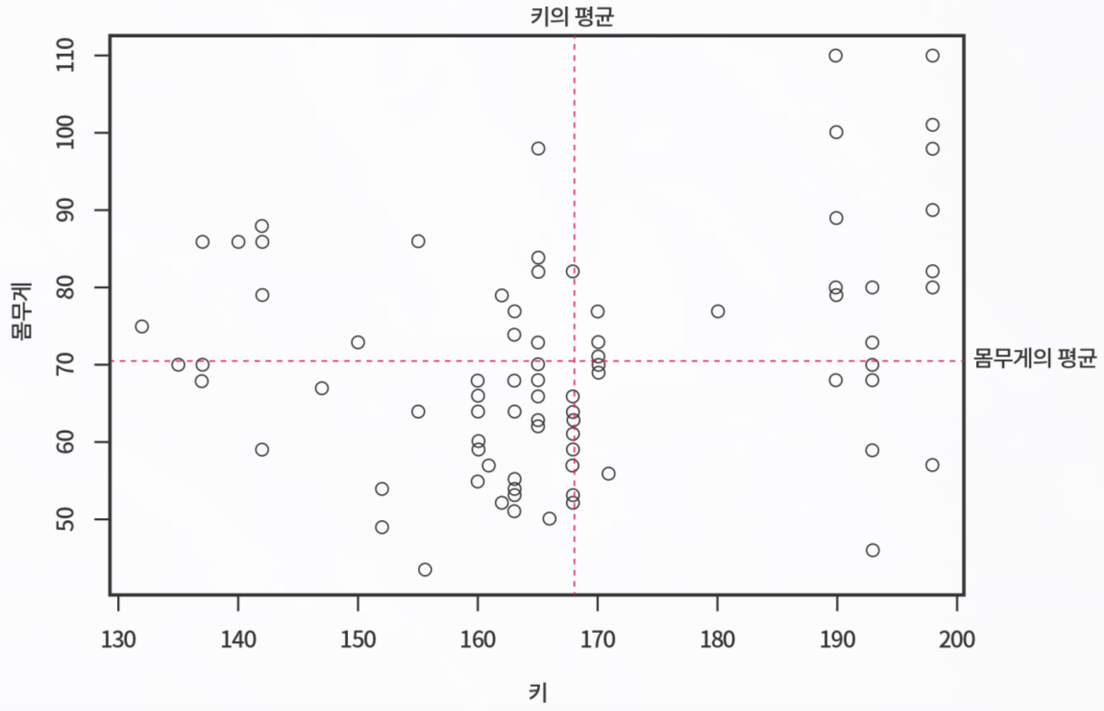


- 기울기가 m이고, y절편이 n인 직선의 방정식 → $y = mx + n$
  - 위의 직선에서 x가 주어지면, y를 구할 수 있다.
    - 하지만, 통계에서는 문제가 그렇게 간단하지 않다.

02 | 선형 회귀분석의 개념

아래의 그림은 대학생 90명의 키(cm)와 몸무게(kg) 데이터로 키 데이터를 x축, 몸무게를 y축에 나타내고 (x, y)를 2차원 점도표로 나타낸 것이다.

- 이를 산포도라고 한다.
- 여기서 어느 학생의 키 x를 보고 몸무게 y를 예측할 수 있을까?
  - 만약, 예측한다면 어떤 방법이 있을까요?



02 | 선형 회귀분석의 개념

회귀분석은 이처럼 어수선한 산점도에 맞는 직선을 찾는 것이다.

- 회귀직선 또는 예측직선은 다음과 같은 형태이다.

$$y = \beta_0 + \beta_1 x$$

  - 여기서 x는 독립변수 또는 예측변수라 하고, y는 종속변수 또는 반응변수라고 한다.

- 예를 들어 학생의 키(cm)로 몸무게(kg)를 예측한다면, 아래와 같은 회귀식으로 나타낼 수 있다.

  - 학생의 몸무게(kg) = $\underbrace{\boxed{\beta_0}}_{\text{절편}} + \underbrace{\boxed{\beta_1}}_{\text{계수}}$ x 학생의 키(cm) ※ 절편과 계수를 회귀 계수

02 | 선형 회귀분석의 개념

- 다음은 대학생 90명의 키와 몸무게 데이터 셋을 읽어오는 코드이다.
  - 아래와 같이 90개의 관측치와 4개의 속성으로 구성된 것을 볼 수 있다.

```
std90 = pd.read_csv(os.getcwd()+'/student90.csv')
print(std90.shape)     # (90, 4)
print(std90.info)
```

```
(90, 4)
<bound method DataFrame.info of     no sex  weight_kg  height_cm
0    1   m         98        198
1    2   m         77        170
2    3   m         70        170
3    4   m         90        198
4    5   m         71        170
..  ..  ..        ...        ...
85  88   f        100        190
86  89   f         54        163
87  90   f         57        161
88  91   f        101        198
89  92   f        110        190

[90 rows x 4 columns]>
```

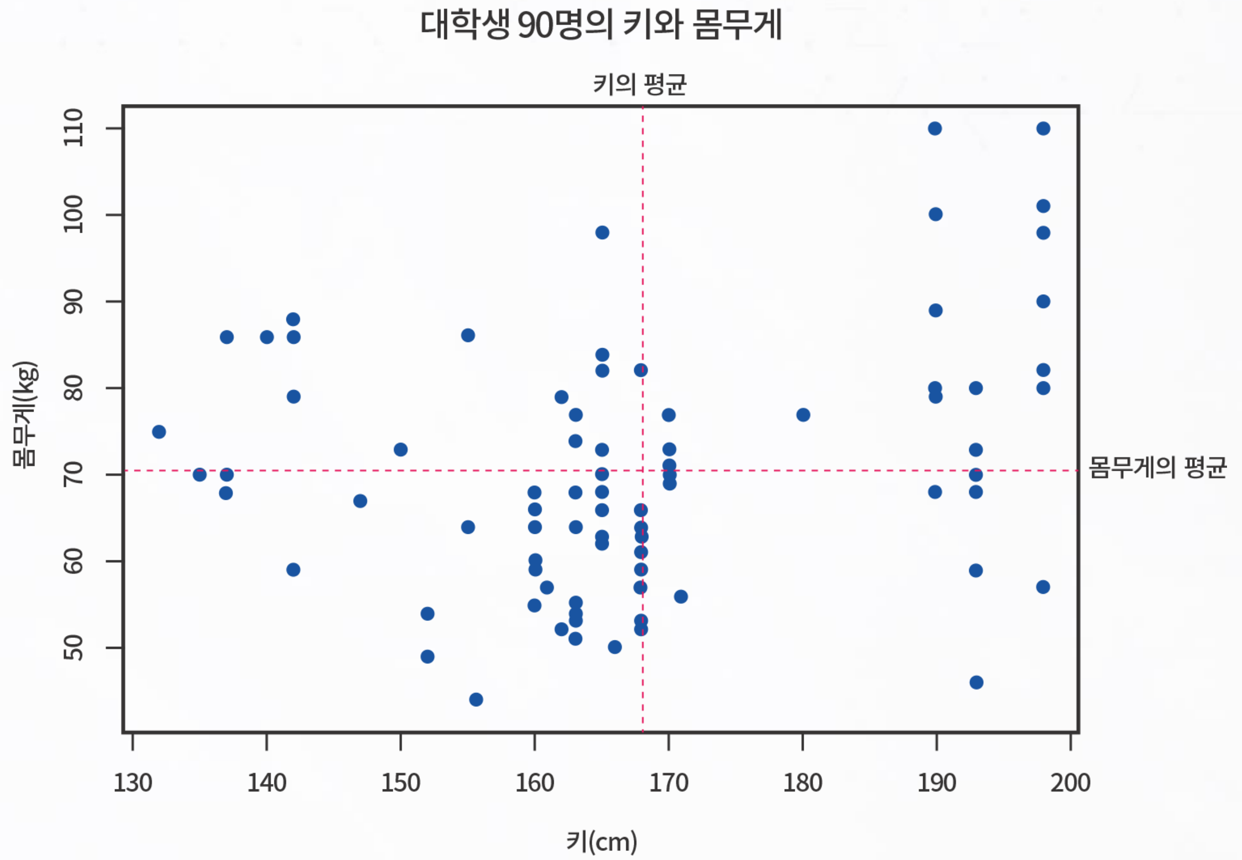
- 다음은 대학생 90명의 **키와 몸무게 데이터 셋**으로 **산점도**를 그리는 코드이다.
  - 여기서는 x축에 키(cm), y축에 몸무게(kg)를 나타낸다.
    - 그리고 몸무게와 키의 평균은 점선으로 표시한다.

```
# 몸무게 평균
w_avg = np.mean(std90['weight_kg'])
print('몸무게 평균:', w_avg)

# 키 평균
h_avg = np.mean(std90['height_cm'])
print('키 평균:', h_avg)

# 키와 몸무게로 산점도 그리기
plt.scatter(std90['height_cm'], std90['weight_kg'])
plt.title('대학생 90명 키와 몸무게', fontsize=16)
plt.xlabel('키(cm)', fontsize=12)
plt.ylabel('몸무게(kg)', fontsize=12)
plt.axhline(w_avg, color='gray', linestyle='--', linewidth=1)
plt.axvline(h_avg, color='gray', linestyle='--', linewidth=1)
plt.text(169,110,"키의 평균")
plt.text(175,72,"몸무게의 평균")
plt.show()
```

02 | 선형 회귀분석의 개념

- 아래의 그림은 실행결과이다.
  - 여기서 몸무게 평균은 약 70.43(kg), 키의 평균은 약 168.13(cm)인 것을 알 수 있다.

```
몸무게 평균: 70.43333333333334
키 평균: 168.13333333333333
```



02 | 선형 회귀분석의 개념

- 다음은 회귀직선을 찾는 과정을 보여주기 위해, 대학생 90명의 키와 몸무게 데이터에서 임의로 10명의 데이터를 추출하는 코드이다.
  - 데이터를 섞어서 임의로 10개를 선택하는 함수는 다음과 같다.
    - pd.DataFrame.sample() 함수: 판다스에서 데이터프레임 행을 섞어서 임의의 항목 샘플을 반환한다.
    - numpy.random.permutation() 함수: 데이터프레임의 인덱스를 섞을 수 있다.
    - sklearn.utils.shuffle() 함수: 데이터프레임 행을 섞는다.
    - 여기서는 shuffle() 함수를 이용해서 데이터에서 임의로 10명의 데이터를 추출한다.

```
import sklearn

std90_shuffled = sklearn.utils.shuffle(std90, n_samples=10,
random_state=1234).reset_index(drop=True)
print(std90_shuffled)
```

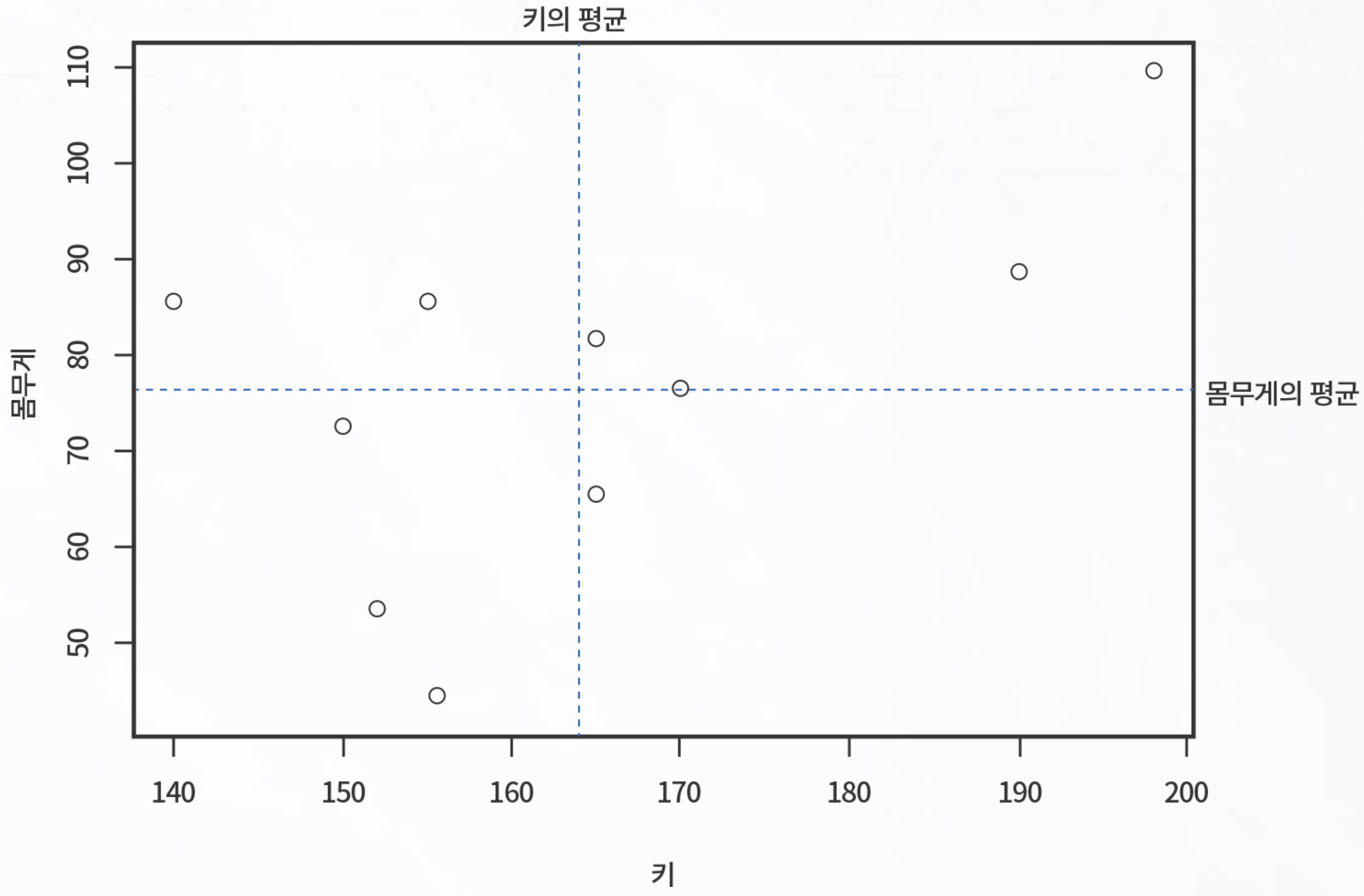
- 실행결과 아래와 같이 임의로 10명의 데이터가 추출된 것을 볼 수 있다.

	weight_kg	height_cm
1	86	140
2	73	150
3	54	152
4	82	165
5	89	190
6	43	156
7	77	170
8	110	198
9	66	165
10	86	155

다음은 임의로 10명을 추출한 데이터로 산점도를 그리는 코드이다.

```
# 몸무게 평균
w_avg = np.mean(std90_shuffled['weight_kg'])
print('몸무게 평균:', w_avg)

# 키 평균
h_avg = np.mean(std90_shuffled['height_cm'])
print('키 평균:', h_avg)

# 키와 몸무게로 산점도 그리기
plt.scatter(std90_shuffled['height_cm'], std90_shuffled['weight_kg'])
plt.title('대학생 10명 키와 몸무게', fontsize=20)
plt.xlabel('키(cm)', fontsize=14)
plt.ylabel('몸무게(kg)', fontsize=14)
plt.axhline(w_avg, color='gray', linestyle='--', linewidth=1)
plt.axvline(h_avg, color='gray', linestyle='--', linewidth=1)
plt.text(168,85,"키의 평균")
plt.text(175,63,"몸무게의 평균")
plt.show()
```

- 아래 그림은 대학생 10명의 키와 몸무게의 산점도이다.
  - 여기서 회귀직선을 어떻게 얻을 수 있을지를 생각해보자.



회귀분석의 역사적 배경

- 19세기 말 유전학자 프랜시스 골턴은 평균으로의 회귀라는 현상을 발견했다.
  - 그는 유전법칙을 찾던 중 부모의 키가 자손으로 전달된다는 사실을 발견했다.
    - 즉, 아버지가 키가 크면 아들은 조금 작고, 반대로 아버지의 키가 작으면 아들은 조금 더 큰 경향이 있었다.
  - 골턴은 자신이 "평균으로의 회귀"라고 불렀던 이 현상을 연구하기 위해 회귀분석법을 개발했다.

회귀직선 (또는 최소제곱 직선)

SSE(Sum of Squares to residual Error, 오차 제곱합)가 최소가 되는 직선을 말한다.

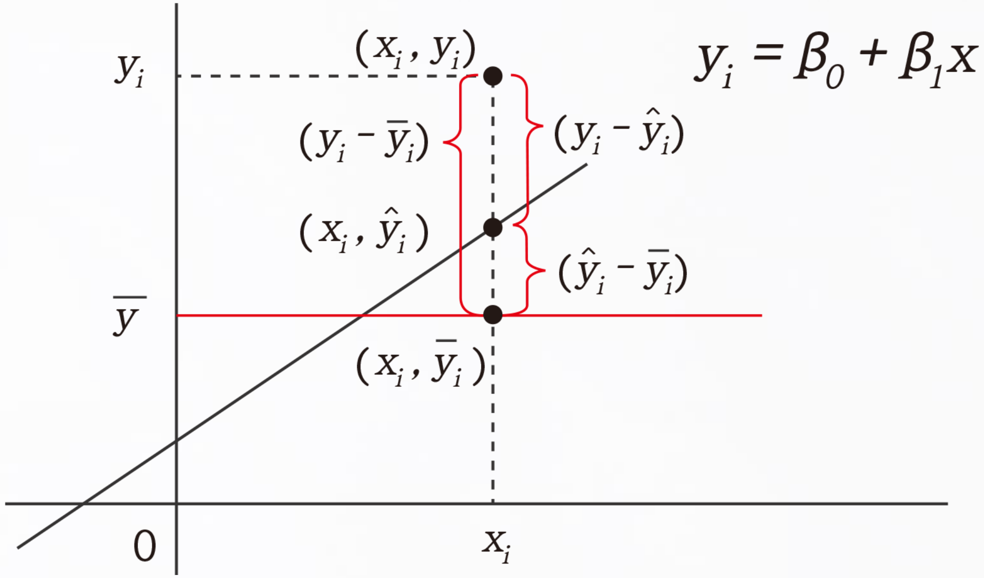
- 최소 제곱법이란 잔차 제곱의 합이 최소가 되도록 값을 정하는 방법이다.
- 회귀 모형에서는 오차의 제곱 합(SSE) $\sum \varepsilon^2$ 이 최소가 되도록 회귀 계수를 정한다.

SST = SSR + SSE

$$SST = \sum (y_i - \bar{y})^2$$

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

$$SSR = \sum (\hat{y}_i - \bar{y})^2$$



※ SST(Total Sum of Squares): 총 제곱 합
SSR(Sum of Squares due to Regression): 회귀 제곱 합
SSE(Sum of Squares to residual Error): 잔차(오차) 제곱 합

04 | 회귀직선 (또는 최소제곱 직선)

회귀직선의 식은 아래와 같다.

$$y = \beta_0 + \beta_1 x$$

여기서 $\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$, $\beta_0 = \bar{y} - \beta_1 \bar{x}$ 이다.

$$SS_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$SS_{yy} = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

평균을 중심으로 한 거리의 제곱의 합, x_i와 y_i의 산포도 측정

$$SS_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

외적(교차곱)은 SS_{xx}와 함께 계수 β_1을 결정

04 | 회귀직선 (또는 최소제곱 직선)

회귀직선을 찾는 과정을 보여주기 위해, 대학생 90명의 키와 몸무게 데이터에서 임의로 10명의 데이터를 추출하여 사용한다.

✦ 아래의 표는 이 데이터 셋으로 분산계산표를 계산한 결과이다.

No	x_i	y_i	$(x_i - \bar{x})$	$(y_i - \bar{y})$	$(x_i - \bar{x})^2$	$(y_i - \bar{y})^2$	$(x_i - \bar{x})(y_i - \bar{y})$
1	140	86	-24.1	9.4	580.81	88.36	-226.54
2	150	73	-14.1	-3.6	198.81	12.96	50.76
3	152	54	-12.1	-22.6	146.41	510.76	273.46
4	165	82	0.9	5.4	0.81	29.16	4.86
5	190	89	25.9	12.4	670.81	153.76	321.16
6	156	43	-8.1	-33.6	65.61	1128.96	272.16
7	170	77	5.9	0.4	34.81	0.16	2.36
8	198	110	33.9	33.4	1149.21	1115.56	1132.26
9	165	66	0.9	-10.6	0.81	112.36	-9.54
10	155	86	-9.1	9.4	82.81	88.36	-85.54
합계	1641	766	0	0	SS_{xx}=2930.9	SS_{yy}=3240.4	SS_{xy}=1735.4
평균	164.1	76.6					

04 | 회귀직선 (또는 최소제곱 직선)

앞의 분산 계산표에서 β_0, β_1 의 값은 다음과 같다.

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} = \frac{SS_{xy}}{SS_{xx}} = \frac{1735.4}{2930.9} = 0.5921048$$

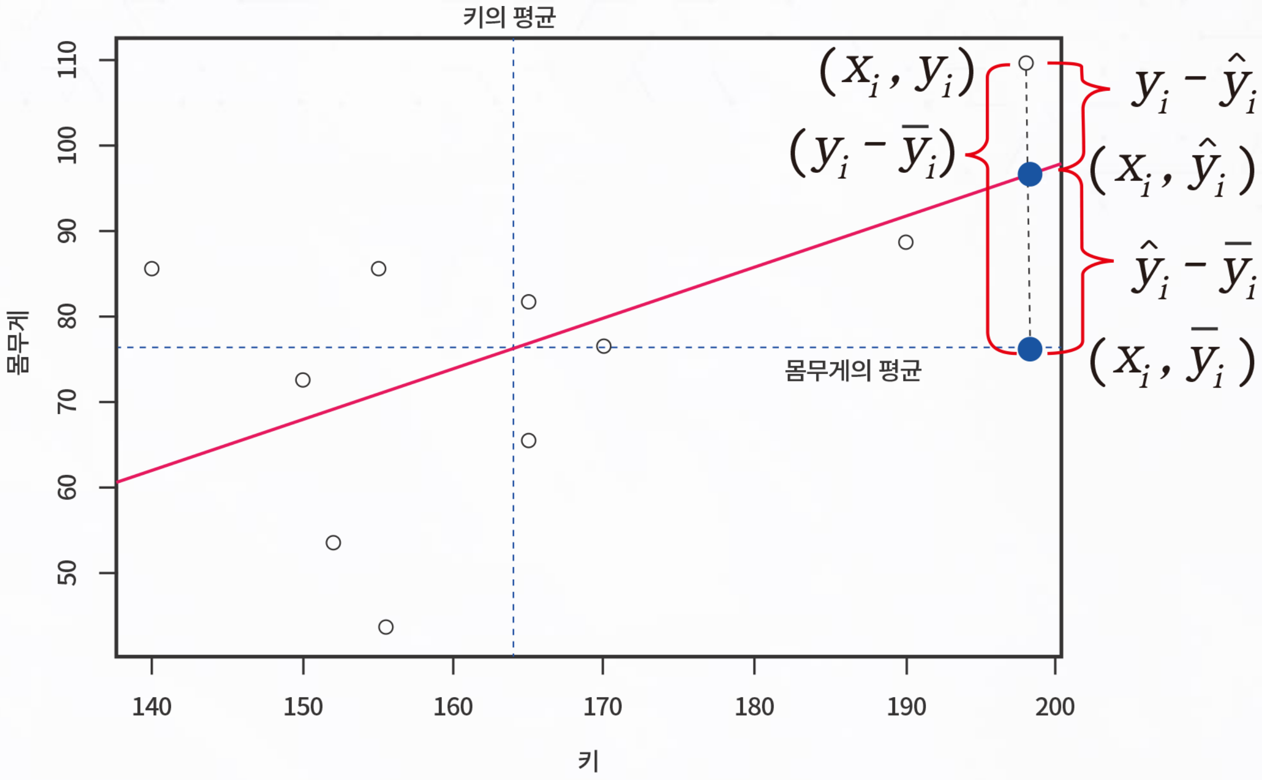
$$\beta_0 = \overline{y} - \beta_1 \overline{x} = 76.6 - (0.592) * 164.1 = -20.5472$$

즉 회귀직선은 다음과 같다.

$$y = -20.5472 + 0.592x$$

04 | 회귀직선 (또는 최소제곱 직선)

- 회귀직선은 항상 평균 점($\bar{x}$, $\bar{y}$)을 지나간다.
  - 즉, 회귀직선은 x축의 키와 y축의 몸무게 평균값을 지나간다.



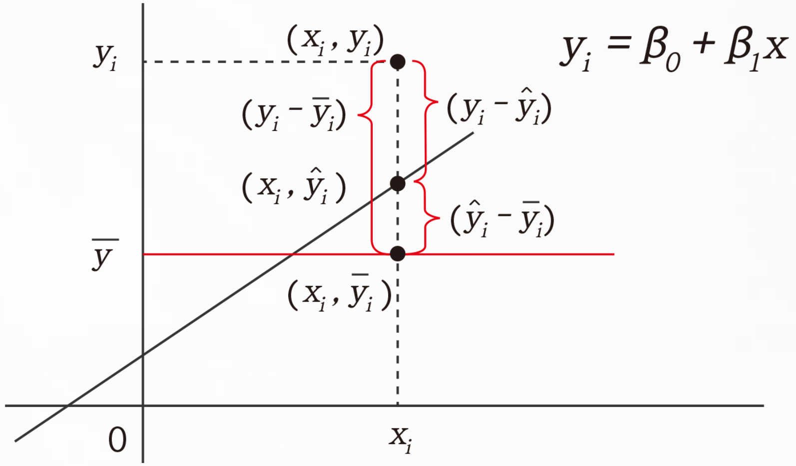
분산분석(Analysis of variance, ANOVA)

- 아래 그림에서 직선이 최상의 적합한 직선이라면, 어느 정도 정확한가?
  - 그 답은 데이터 점들이 어느 정도 흩어져 있는지에 달려 있다.
    - 데이터의 전체 편차(SST)보다 SSE가 어느 정도 큰지에 달려 있다.
      - 즉, SSE가 작은 값이면 정확도가 높다.

$$SST = \sum (y_i - \bar{y})^2$$

$$SSE = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

$$SSR = \sum (y_i - \hat{y}_i)^2$$

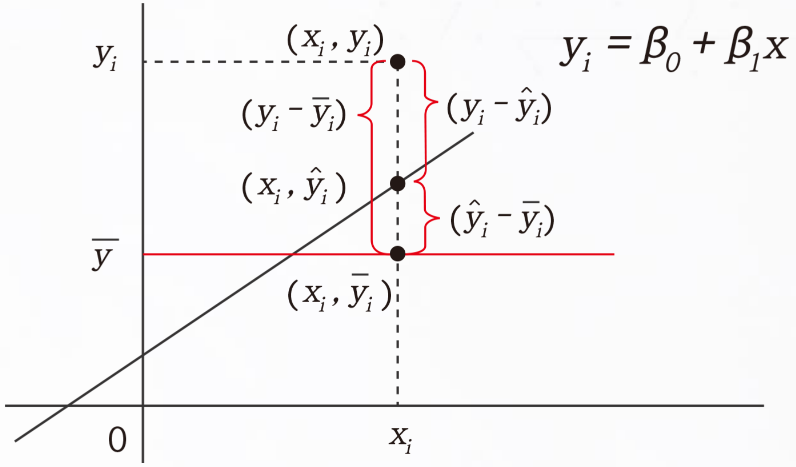


※ SST(Total Sum of Squares): 총 제곱 합
SSR(Sum of Squares due to Regression): 회귀 제곱 합
SSE(Sum of Squares to residual Error): 잔차(오차) 제곱 합

총편차제곱합(Total Sum of Squared Deviation) 또는 총제곱합(Sum of Squares Total)

- 여기서 표본 구성요소 y_i와 총 평균 $\bar{y}$간의 차이 $(y_i - \bar{y})$를 y_i와 $\bar{y}$간의 편차(deviation)라고 한다.
- 각각의 표본 관찰값(y_i)과 총 평균($\bar{y}$)간의 편차를 제곱하여 모두 더한 것이다.

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2$$



※ SST(Total Sum of Squares): 총 제곱 합

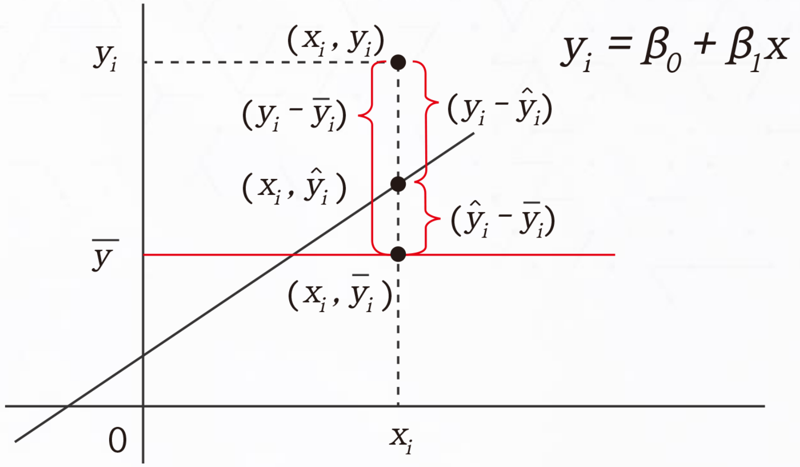
05 | 분산분석

데이터 점 들이 어느 정도 흩어져 있는지 계량화하면, 아래와 같이 정리할 수 있다.

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

$$SSR = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$$

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

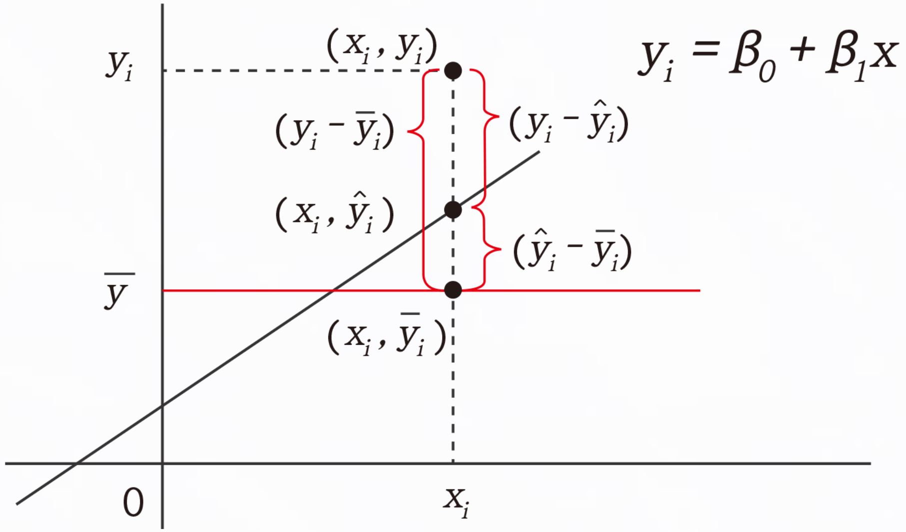


※ SST(Total Sum of Squares): 총 제곱 합
SSR(Sum of Squares due to Regression): 회귀 제곱 합
SSE(Sum of Squares to residual Error): 잔차(오차) 제곱 합

아래의 그래프의 직선 식(회귀직선)은 아래와 같다.

$$\hat{y}_i = \underbrace{\boxed{\beta_0}}_{\text{절편}} + \underbrace{\boxed{\beta_1}}_{\text{계수}} x_i$$

$\hat{y}_i$는 회귀직선으로 결정되는 몸무게 예측치이다 (여기서는 대학생 키로 몸무게를 예측한다).



분산계산표

✦ 대학생 키와 몸무게 데이터에서 10명의 데이터를 추출하여 아래와 같은 직선을 찾아냈다 (아래의 표는 분산계산표이다).

$$\hat{y}_i = -20.5472 + 0.592x_i$$

No	x_i	y_i	$\hat{y}_i$	$(\hat{y}_i - \bar{y})$	$(\hat{y}_i - \bar{y})^2$	$(y_i - \hat{y}_i)$	$(y_i - \hat{y}_i)^2$	$(y_i - \bar{y}_i)$	$(y_i - \bar{y}_i)^2$
1	140	86	62.3328	-14.2672	203.5530	23.6672	560.1364	9.4	88.36
2	150	73	68.2528	-8.3472	69.6757	4.7472	22.5359	-3.6	12.96
3	152	54	69.4368	-7.1632	51.3114	-15.4368	238.295	-22.6	510.76
4	165	82	77.1328	0.5328	0.2839	4.8672	23.6896	5.4	29.16
5	190	89	91.9328	15.3328	235.0948	-2.9328	8.6013	12.4	153.76
6	156	43	71.8048	-4.7952	22.9939	-28.8048	829.7165	-33.6	1128.96
7	170	77	80.0928	3.4928	12.1997	-3.0928	9.5654	0.4	0.16
8	198	110	96.6688	20.0688	402.7567	13.3312	177.7209	33.4	1115.56
9	165	66	77.1328	0.5328	0.2839	-11.1328	123.9392	-10.6	112.36
10	155	86	71.2128	-5.3872	29.0219	14.7872	218.6613	9.4	88.36
합계	1641	766		$SSR = 1027.175$		$SSE = 2212.861$		$SST = 3240.4$	
평균	164.1	76.6							

대학생 10명의 데이터로 분산계산표를 정리하면 다음과 같다.

✦ 앞의 분산계산표에서 SST = SSR + SSE 인 것을 알 수 있다.

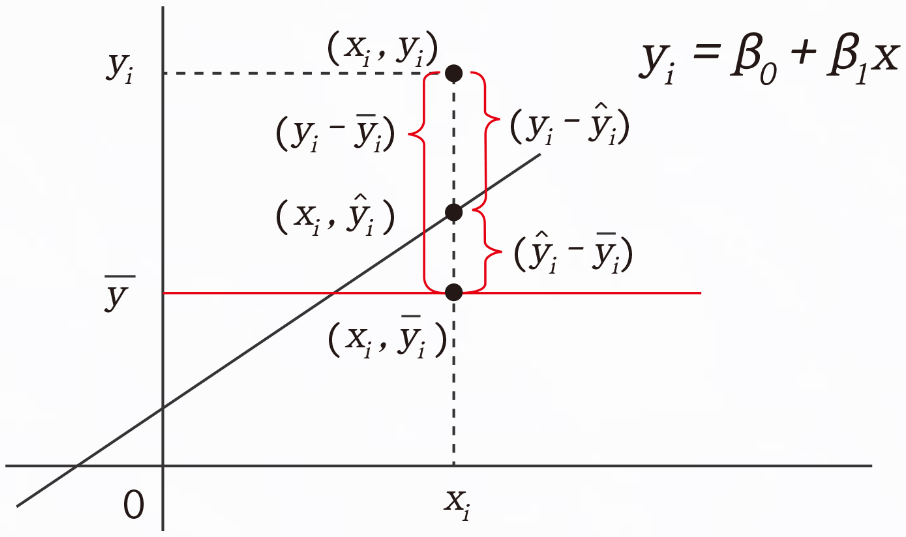
변동요인	제곱합	임의 데이터에 대한 계산 값
회귀	$SSR = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$	1027.175
잔차	$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	2212.861
합계	$SS_{yy} = \sum_{i=1}^{n}(y_i - \bar{y})^2$	3240.4

※ SST(Total Sum of Squares): 총 제곱 합
SSR(Sum of Squares due to Regression): 회귀 제곱 합
SSE(Sum of Squares to residual Error): 잔차(오차) 제곱 합

SSR은 회귀로 생긴 변동이다.

- 즉, y의 예측값들을 측정한다.

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$



※ SSR(Sum of Squares due to Regression): 회귀 제곱 합

총 편차에 대한 잔차의 비율은 다음과 같다.

$$\frac{SSE}{SST}$$

$$SST(SS_{yy}) = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

$$SSR = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$$

SST = SSR + SSE

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

※ SST(Total Sum of Squares): 총 제곱 합
SSR(Sum of Squares due to Regression): 회귀 제곱 합
SSE(Sum of Squares to residual Error): 잔차(오차) 제곱 합

결정 계수(R-squared; R^2로 표기)

결정 계수는 총 변동 중 회귀로 생긴 변동이 기여하는 비율로서 다음과 같이 정의한다.

$$\boldsymbol{R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}} \quad (0 \le R^2 \le 1)$$

✦ 왜냐하면 SSR = SST − SSE 이기 때문이다.

※ SST(Total Sum of Squares): 총 제곱 합
SSR(Sum of Squares due to Regression): 회귀 제곱 합
SSE(Sum of Squares to residual Error): 잔차(오차) 제곱 합

06 | 결정 계수

앞의 대학생 10명 데이터의 경우 결정 계수는 다음과 같다.

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} = 1 - \frac{2212.861}{3240.4} = 1 - 0.6829 = 0.3171$$

✦ 위 결과에서 몸무게 변화의 31.71%는 키로 설명된다는 뜻이다.

✦ 나머지 68.29%는 '오차' 이다.

※ SST(Total Sum of Squares): 총 제곱 합
SSR(Sum of Squares due to Regression): 회귀 제곱 합
SSE(Sum of Squares to residual Error): 잔차(오차) 제곱 합

07 | 수정 결정 계수

수정 결정 계수

- 결정 계수(R^2)은 독립 변수가 늘어나면 값이 커지는 성질이 있다.
  - 서로 다른 개수의 독립 변수를 사용하는 모델 간의 비교에 사용하기에는 적합하지 않다.
  - 따라서, R^2을 자유도로 나눈 수정 결정 계수(Adjusted R-squared; ${R_{adj}}^2$ 으로 표기)를 더 많이 사용한다.

$$\mathrm{R_{adj}}^2 = 1 - \frac{\frac{\mathrm{SSE}}{(\mathrm{n}-\mathrm{k}-1)}}{\frac{\mathrm{SST}}{(\mathrm{n}-1)}}$$

> 여기서 SSE는 잔차 제곱 합, SST는 총 제곱 합이고, n은 데이터의 수, k는 독립변수의 수이다.

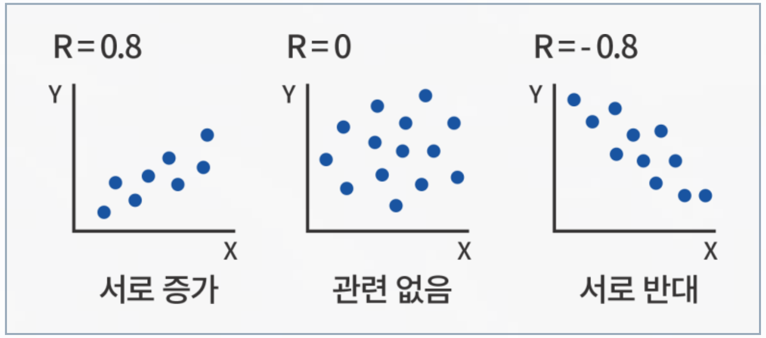
상관 계수(correlation coefficient)

- 상관계수는 결정계수 대신 사용하는 R^2의 제곱근이고, 부호는 $\boldsymbol{\beta}_1$과 같다.

$$r = (\boldsymbol{\beta}_1\text{와 같은 부호})\sqrt{R^2}$$

  - r은 회귀선이 오른쪽 위로 향할 때는 +이고, 아래로 향할 때는 −가 된다.

- 상관계수(r)는 회귀선의 정확도를 측정하고, x가 증가할 때 y가 증가하는지 또는 감소하는지를 말해준다.
  - 흔히 상관 계수라고 하면 피어슨 상관 계수를 뜻한다.



예측(prediction) 또는 외삽(extrapolation)

- 도출한 회귀 모형을 바탕으로 주어지지 않은 값에 대해서 추측하는 것을 예측 또는 외삽이라고 한다.
  - 예를 들어 $y = -20.5472 + 0.592x$ 라는 회귀식을 구했고,
    새로운 값 학생 홍길동의 키가 175cm를 대입해보고 싶다는 것이다.
    - 즉, 키로 몸무게를 예측해보고 싶다는 것이다.
  - 예를 들어 $y = 3x + 2$라는 회귀식을 구했고, 우리가 가지고 있는 $x = 1, 2, 3, 4$ 뿐이라면,
    새로운 값 5를 대입해보고 싶다는 것이다.
    - 즉, 예측 결과는 $y=3*5+2=17$이다.

모델 평가

- 평균 제곱근 오차(Root Mean Square Error; RMSE)는 모델에 의해 예측된 값과 실제 환경에서 관찰되는 값의 차이를 다룰 때 많이 사용하는 측도이며 모델의 정밀도(precision)를 표현하는데 적합하다.

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

여기서 y_i는 관측값, $\hat{y}_i$는 추정값

- 평균 절대 오차(Mean Absolute Error; MAE)는 두 연속형 변수 간의 차이를 측정하는 통계량이다.

$$MAE = \frac{\sum_{i=1}^{n}|y_i-x_i|}{n}$$

여기서 y_i는 추정값, x_i 는 관측값

- RMSE와 MAE는 0에 가까운 값일수록 정확도가 높은 것으로 해석한다.

포뮬러(Formula) 해석하기

$$Y1 + Y2 + \cdots + Yn \sim X1 + X2 + \cdots + Xm$$

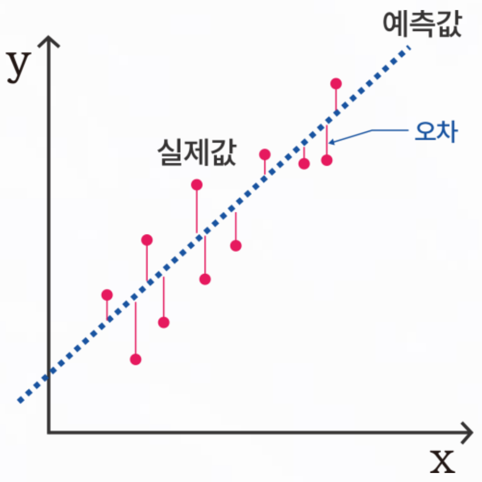
- 포뮬러의 정확한 의미는 사용하는 함수에 따라 다르나 일반적으로 "$(Y1, Y2, \ldots, Yn)$의 순서쌍을 $(X1, X2, \ldots, Xm)$의 순서쌍으로 모델링한다"고 볼 수 있다.
  - ✦ 이때 상수항은 임시적으로 허용된다.

- $Y \sim X1$은 선형 회귀에서 $Y = a * X1 + b$를 의미한다.

회귀 분석의 종류

회귀 분석이란 변수 사이의 인과 관계를 증명하는 방법으로 단순 회귀 분석, 다중 회귀 분석, 로지스틱 회귀 분석 등과 같이 종류가 다양하다.

- 독립 변수가 한 개인 경우 단순 회귀 분석이라고 한다.
- 독립 변수가 두 개 이상인 경우 다중 회귀 분석이라고 한다.



Simple Linear Regression

$$y = b_0 + b_1 * x_1$$

Multiple Linear Regression

Dependent variable (DV) — Independent variables (IVs)

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \cdots + b_n * x_n$$

Constant — Coefficients

단순한 선형 회귀 실습

- 대학생 90명의 키와 몸무게 데이터 셋으로 선형 회귀 분석을 수행해보자.
  - 이 데이터로 키로 몸무게를 예측하는 단순 선형 회귀 모델을 만들어 보자.
  - 그리고 나의 키로 몸무게를 예측해본다.
    - 여기에서는 CLRM(Classical Linear Regression Model) 모델의 가정은 무시함
      - 간단하게 산포도, 회귀직선, 신뢰구간, 모델 학습 및 평가, 예측을 수행함

13 | 단순한 선형 회귀 실습

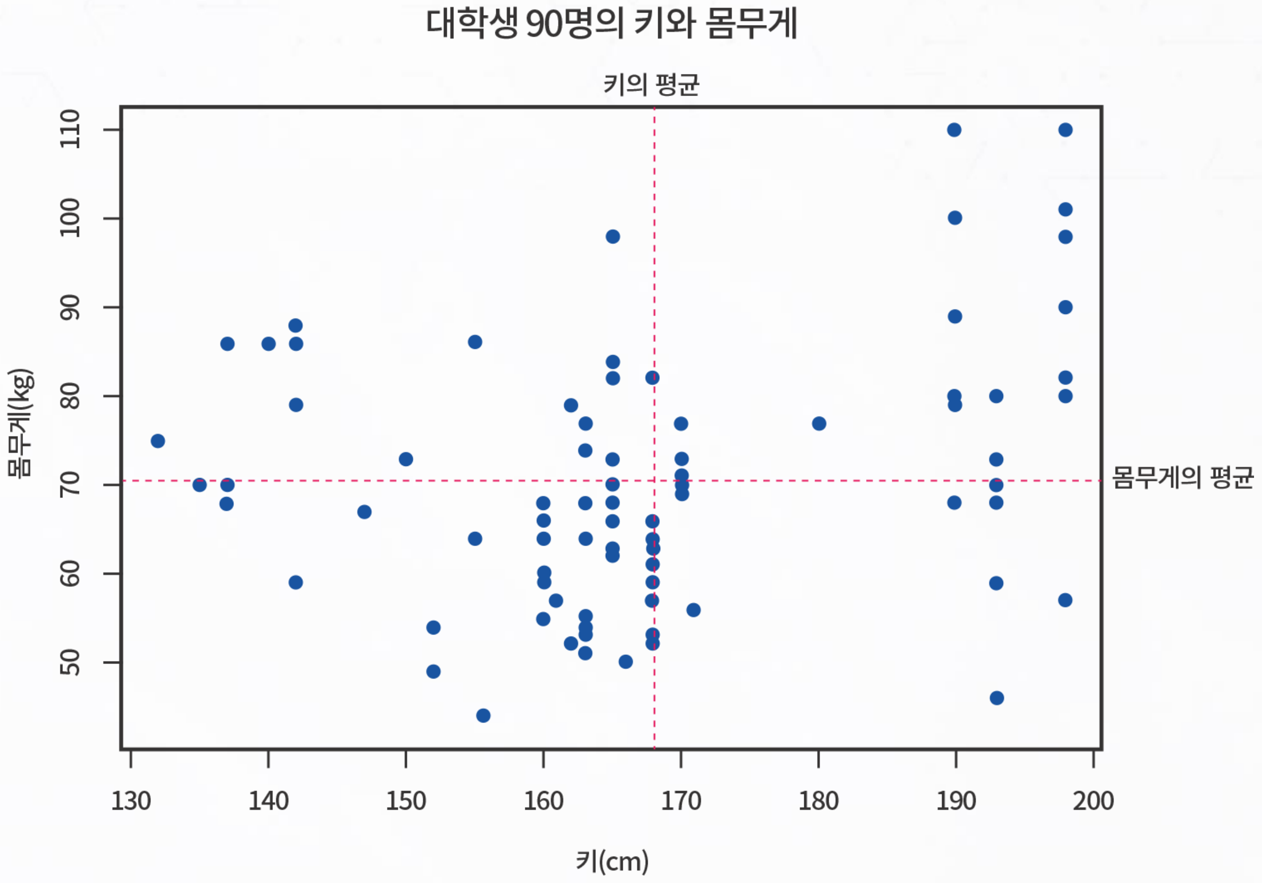
다음은 대학생 90명의 키와 몸무게 데이터 셋으로 **산점도** 그래프를 그려보자.

- 여기에서 **키와 몸무게 평균**도 함께 **표시**한다.

```
# 몸무게 평균
w_avg = np.mean(std90['weight_kg'])

# 키 평균
h_avg = np.mean(std90['height_cm'])

# 키와 몸무게로 산점도 그리기
plt.scatter(std90['height_cm'], std90['weight_kg'])
plt.title('대학생 90명 키와 몸무게', fontsize=16)
plt.xlabel('키(cm)', fontsize=12)
plt.ylabel('몸무게(kg)', fontsize=12)
plt.axhline(w_avg, color='gray', linestyle='--', linewidth=1)
plt.axvline(h_avg, color='gray', linestyle='--', linewidth=1)
plt.text(169,110,"키의 평균")
plt.text(175,72,"몸무게의 평균")
plt.show()
```

> 실행결과 산점도 그래프에 키와 몸무게 평균이 표시된 것을 볼 수 있다.

몸무게 평균: 70.43333333333334
키 평균: 168.13333333333333



13 | 단순한 선형 회귀 실습

다음은 **대학생 90명**의 **키와 몸무게** 데이터 셋으로 **산점도**에 **회귀직선**을 그려보자.

- 여기에서 **키와 몸무게 평균**도 함께 **표시**한다.

```
# 몸무게 평균
w_avg = np.mean(std90['weight_kg'])

# 키 평균
h_avg = np.mean(std90['height_cm'])

# x를 설명변수, y를 반응변수로 하는 1차 회귀 곡선(즉 직선을 적합)
b1, b0 = np.polyfit(std90['height_cm'], std90['weight_kg'], 1)  # 지울기(=b1), 절편(=b0)을 반환
print('b0=', b0, 'b1=', b1)
fit = b0 + b1 * std90['height_cm']

# 키와 몸무게로 산점도, 회귀직선 그리기
plt.scatter(std90['height_cm'], std90['weight_kg']) # 산점도
plt.plot(std90['height_cm'], fit, color='red')    # polyfit() 함수 : 절편, 기울기 계산
plt.title('대학생 10명 키와 몸무게', fontsize=20)
plt.xlabel('키(cm)', fontsize=14)
plt.ylabel('몸무게(kg)', fontsize=14)
plt.axhline(w_avg, color='gray', linestyle='--', linewidth=1)
plt.axvline(h_avg, color='gray', linestyle='--', linewidth=1)
plt.text(169,110,"키의 평균")
plt.text(175,72,"몸무게의 평균")
plt.show()
```

13 | 단순한 선형 회귀 실습

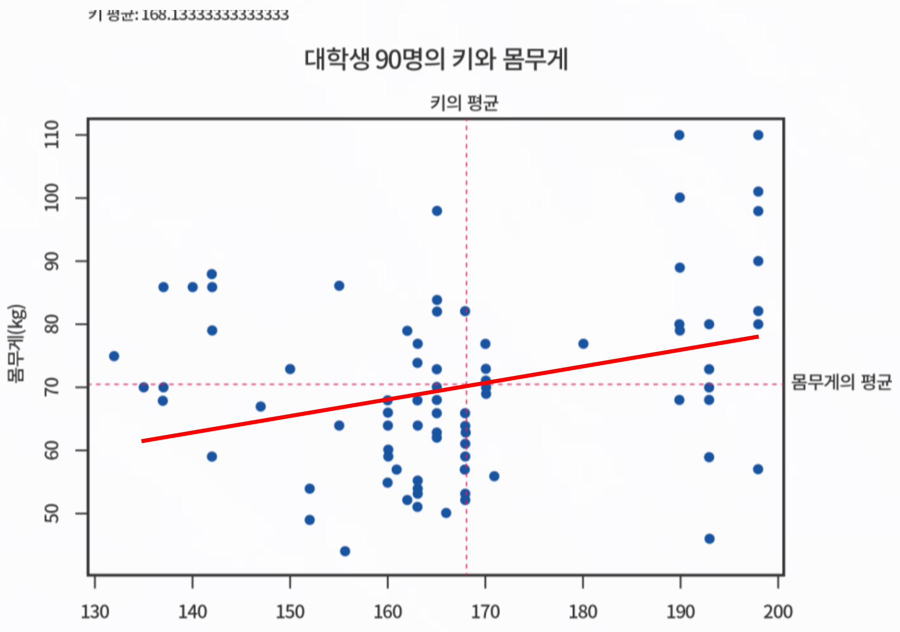
> 실행결과 산점도에 **회귀직선**이 추가된 것을 볼 수 있음

학생 몸무게 = $\underbrace{\boxed{32.660}}_{\text{절편}}$ + $\underbrace{\boxed{0.224}}_{\text{계수}}$ * 학생의 키

몸무게 평균: 70.43333333333334

키 평균: 168.13333333333333

b0= 32.660414362377615 b1= 0.22466050141329735



13 | 단순한 선형 회귀 실습

다음은 대학생 90명의 키와 몸무게 데이터 셋으로 **산점도**에 **회귀직선**과 **신뢰구간**을 그려보자.

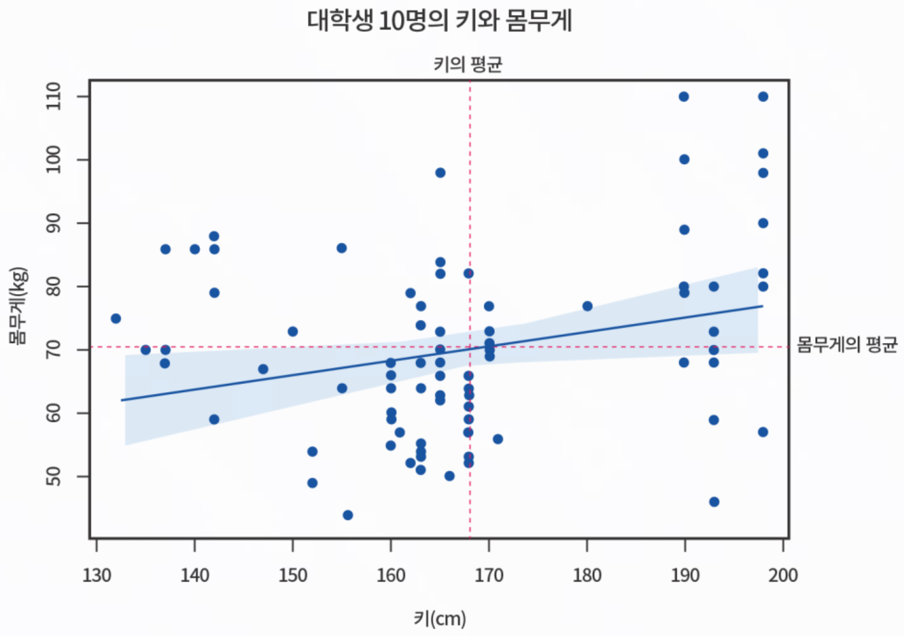
✦ 여기에서 **키와 몸무게 평균**, **신뢰구간**은 **유의수준 95%**로 한다.

```
# 몸무게 평균
w_avg = np.mean(std90['weight_kg'])

# 키 평균
h_avg = np.mean(std90['height_cm'])

# x를 설명변수, y를 반응변수로 하는 1차 회귀 곡선(즉 직선을 적합)
b1, b0 = np.polyfit(std90['height_cm'], std90['weight_kg'], 1)  # 지울기(=b1), 절편(=b0)을 반환
print('b0=', b0, 'b1=', b1)
fit = b0 + b1 * std90['height_cm']

# 키와 몸무게로 산점도, 선형회귀선, 95% 신뢰구간 그리기
plt.scatter(std90['height_cm'], std90['weight_kg'])    # 산점도
sns.regplot(x='height_cm', y='weight_kg', data=std90)        # 회귀직선
plt.title('대학생 10명 키와 몸무게', fontsize=20)
plt.xlabel('키(cm)', fontsize=14)
plt.ylabel('몸무게(kg)', fontsize=14)
plt.axhline(w_avg, color='gray', linestyle='--', linewidth=1)
plt.axvline(h_avg, color='gray', linestyle='--', linewidth=1)
plt.text(169,110,"키의 평균")
plt.text(175,72,"몸무게의 평균")
plt.show()
```

- 실행결과 산점도에 회귀직선, 신뢰구간(95%)이 추가된 것을 볼 수 있다.

학생 몸무게 = $\boxed{32.660}_{\text{절편}}$ + $\boxed{0.224}_{\text{계수}}$ * 학생의 키

```
몸무게 평균: 70.43333333333334
키 평균: 168.13333333333333
b0= 32.660414362377615 b1= 0.22466050141329735
```



13 | 단순한 선형 회귀 실습

다음은 대학생 90명의 키와 몸무게 데이터 셋으로 **모델 생성 및 학습**을 수행하는 코드이다.

- 모델 생성 및 학습결과 다음과 같은 **회귀식**이 계산된 것을 알 수 있다.

학생 몸무게 = $\underbrace{\boxed{32.660}}_{\text{절편}}$ + $\underbrace{\boxed{0.224}}_{\text{계수}}$ * 학생의 키

```
# 모형 생성 및 학습하기
model_lr = LinearRegression().fit(np.c_[std90['height_cm']], np.c_[std90['weight_kg']])

# 회귀 계수 : 절편, 기울기
print("intercept=", model_lr.intercept_)  # 절편 intercept = [32.66041436]
print("coef=", model_lr.coef_)          # 기울기(계수) coef = [[0.2246605]]

# 학생 몸무게(kg) = 32.66041436 + 0.2246605 * 학생의 키(cm)
```

13 | 단순한 선형 회귀 실습

- 다음은 학습된 모델로 **모델 성능평가**를 수행하는 코드이다.
  - 여기서는 모델 성능평가 지표로 MSE를 이용한다.
    - 실행결과 **MSE = 176.6653** 인 것을 볼 수 있다.

```
mse = mean_squared_error(y_true = std90['weight_kg'], y_pred = model_lr.predict(np.c_[std90['height_cm']]))
mse            # 176.6653444314721
```

13 | 단순한 선형 회귀 실습

- 다음은 학습된 모델로 예측을 수행하는 코드이다.
  - 여기서는 새로운 학생의 키가 175cm 이다.
    - 실행결과 몸무게가 약 71.92(kg) 인 것을 볼 수 있다.

$$\text{학생 몸무게} = \underbrace{\boxed{32.660}}_{\text{절편}} + \underbrace{\boxed{0.224}}_{\text{계수}} * \text{학생의 키}$$

```
X_new = [[175]]                                # 새로운 학생의 키(cm) = 175
print("Predict=", model_lr.predict(X_new))   # 새로운 학생 키에 대한 예측 결과 = [[71.97600211]]
```