

강원지역혁신플랫폼

# 117기 학습

Machine Learning

선형 회귀 분석 실습(2)



## ▼ 학습목표

- 📁 **Mall Customers Clustering Analysis**

데이터 집합으로 군집 분석을 구현할 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

내 드라이브 > PyWork > ML

유형 ▾ 사람 ▾ 수정 날짜 ▾

이름	↑
ch09	
ch10	
ch11	
ch12	
ch13	
ch14	
HelloWorld	



# 01 | 실습

- ◆ “ML > ch11 >” 폴더를 클릭함
- 아래의 [ch11\_03\_선형회귀 실습.ipynb] 스크립트를 클릭함

... > ML > ch11

유형 ▾ 사람 ▾ 수정 날짜 ▾

이름 ↑
ch11_01_선형회귀.ipynb
ch11_02_선형회귀 실습.ipynb
<b>ch11_03_선형회귀 실습.ipynb</b>
student90.csv
student90.txt
student300_outliers.txt
X student300_outliers.xlsx
student300.txt
X student300.xlsx



## 02 | 단순한 선형 회귀 실습



### 단순한 선형 회귀 실습

△ 대학생 300명의 키와 몸무게 데이터 셋으로 **선형 회귀 분석**을 수행해보자.

- ◆ 이 데이터로 키로 몸무게를 예측하는 단순 선형 회귀 모델을 만든다.
  - › 여기서는 **이상치 데이터**를 전처리함
  - › CLRM(Classical Linear Regression Model) 모델의 가정은 무시함
  - › 간단하게 산포도, 회귀직선, 신뢰구간, 모델 학습 및 평가, 예측을 수행함
    - 예측은 **나의 키로 몸무게를 예측**함

성명	성별	학년	키(cm)	몸무게(kg)	취미
학생1	남	1	170.4	69.1	게임
학생2	여	3	169.3	62.0	음악
...	...	...	...	...	...



## 02 | 단순한 선형 회귀 실습

❖ 다음은 대학생 300명의 키와 몸무게 데이터셋을 읽어오는 코드이다.

◆ 실행결과 데이터 형상은 (300, 6)인 것을 알 수 있음

```
std = pd.read_excel(os.getcwd()+'student300_outliers.xlsx')
print(std.shape)    # (300, 6)
print(std.info)
```

```
(300, 6)
<bound method DataFrame.info of      설명 성별 학년   키(cm)   몸무게(kg)   취미
0   학생1 남   1   170.5   69.0   게임
1   학생2 여   1   163.5   51.8   독서
2   학생3 남   3   191.4   60.2   음악
3   학생4 남   2   176.3   70.7   수영
4   학생5 남   2   149.7   57.1   수영
...
295  학생296 여   4   170.5   62.2   음악
296  학생297 여   1   172.6   63.7   등산
297  학생298 남   3   161.0   65.8   등산
298  학생299 남   4   176.4   49.8   수영
299  학생300 여   2   153.4   56.2   수영
```

```
[300 rows x 6 columns]>
```



## 02 | 단순한 선형 회귀 실습

❖ 다음은 키(cm)속성의 평균과 중앙값을 계산한 결과이다.

◆ 실행결과 키의 평균과 중앙값은 거의 같은 것을 알 수 있음

```
mean_height = std['키(cm)'].mean()  
median_height = std['키(cm)'].median()  
print(f"키 평균: {mean_height} cm")      # 키 평균: 170.56866666666667 cm  
print(f"키 중앙값: {median_height} cm")  # 키 중앙값: 170.0 cm
```



## 02 | 단순한 선형 회귀 실습

❖ 다음은 몸무게(kg) 속성의 평균과 중앙값을 계산한 결과이다.

◆ 실행결과 몸무게의 평균과 중앙값은 거의 같은 것을 알 수 있음

```
mean_weight = std['몸무게(kg)'].mean()  
median_weight = std['몸무게(kg)'].median()  
print(f"몸무게 평균: {mean_weight} kg")    # 몸무게 평균: 64.83899999999998 kg  
print(f"몸무게 중앙값: {median_weight} kg") # 몸무게 중앙값: 65.0 kg
```

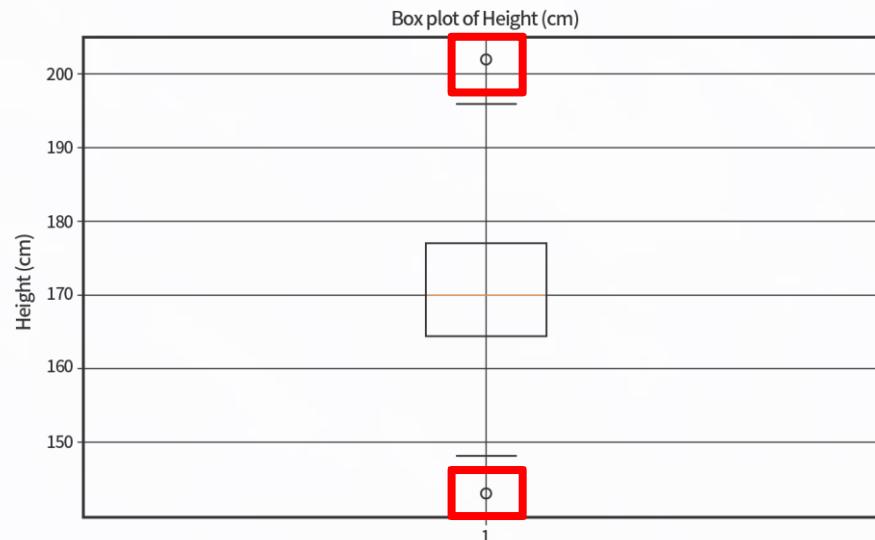


## 02 | 단순한 선형 회귀 실습

❖ 다음은 키(cm)속성으로 상자그림을 그린 결과이다.

◆ 실행결과 키 속성에는 이상치 데이터가 포함된 것을 알 수 있음

```
plt.figure(figsize=(10, 6))
plt.boxplot(std['키(cm)'])
plt.title('Box Plot of Height (cm)')
plt.ylabel('Height (cm)')
plt.grid(True)
plt.show()
```





## 02 | 단순한 선형 회귀 실습

❖ 다음은 키(cm) 속성의 이상치 데이터를 출력한 결과이다.

◆ 실행결과 두 명(학생014, 학생149)의 학생 키가 출력된 것을 볼 수 있음

```
# 이상치 탐지: 상자그림을 이용한 IQR 방법 적용  
Q1 = std["키(cm)"].quantile(0.25)  
Q3 = std["키(cm)"].quantile(0.75)  
IQR = Q3 - Q1  
  
# 이상치 경계 설정  
lower_bound = Q1 - 1.5 * IQR  
upper_bound = Q3 + 1.5 * IQR  
  
# 이상치 데이터 필터링  
outliers = std[(std["키(cm)"] < lower_bound) | (std["키(cm)"] > upper_bound)]  
  
# 이상치 데이터 출력  
print(outliers)
```

	성명	성별	학년	키(cm)	몸무게(kg)	취미
13	학생014	여	1학년	201.9	56.9	게임
148	학생149	여	3학년	143.0	77.4	게임



## 02 | 단순한 선형 회귀 실습

❖ 다음은 키(cm) 속성의 이상치 데이터를 클린징하는 코드이다.

◆ 학생014의 키를 201.9cm에서 160cm로 수정함

› 학생014의 경우 몸무게가 56.9kg인 것을 감안하여 160cm로 조정함

◆ 학생149의 키를 143cm에서 180cm로 수정함

› 학생149의 경우 몸무게가 77.4kg인 것을 감안하여 180cm로 조정함

```
std.loc[std['성명'] == '학생014', '키(cm)'] = 160
```

```
std.loc[std['성명'] == '학생149', '키(cm)'] = 180
```

	성명	성별	학년	키(cm)	몸무게(kg)	취미
13	학생014	여	1학년	201.9	56.9	게임
148	학생149	여	3학년	143.0	77.4	게임

변경전 키와 몸무게

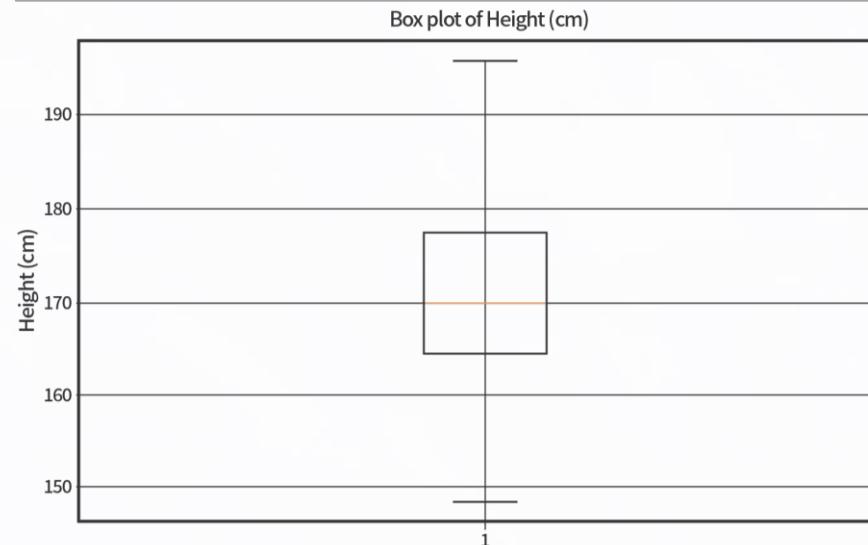


## 02 | 단순한 선형 회귀 실습

❖ 다음은 데이터 클린징을 수행한 키(cm) 속성으로 상자그림을 그린 결과이다.

◆ 실행결과 키 속성에는 이상치 데이터가 없는 것을 알 수 있음

```
plt.figure(figsize=(10, 6))
plt.boxplot(std['키(cm)'])
plt.title('Box Plot of Height (cm)')
plt.ylabel('Height (cm)')
plt.grid(True)
plt.show()
```



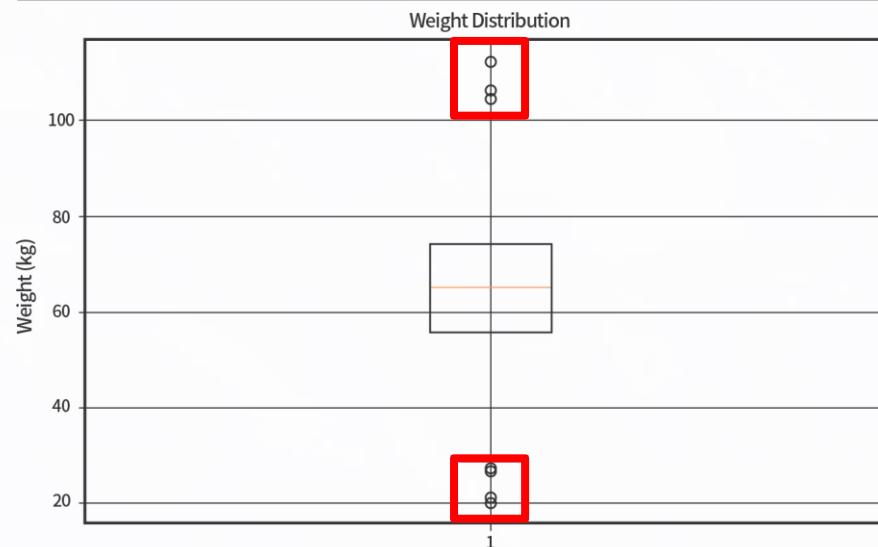


## 02 | 단순한 선형 회귀 실습

❖ 다음은 몸무게(kg) 속성으로 상자그림을 그린 결과이다.

◆ 실행결과 몸무게 속성에는 이상치 데이터가 포함된 것을 알 수 있음

```
plt.figure(figsize=(10, 6))
plt.boxplot(std['몸무게(kg)'])
plt.title('Box Plot of Weight (kg)')
plt.ylabel('Weight (kg)')
plt.grid(True)
plt.show()
```





## 02 | 단순한 선형 회귀 실습

❖ 다음은 **몸무게(kg)** 속성의 이상치 데이터를 출력한 결과이다.

◆ 실행결과 7명의 학생 몸무게가 출력된 것을 볼 수 있음

```
# 이상치 탐지: 상자그림을 이용한 IQR 방법 적용  
Q1_weight = std["몸무게(kg)"].quantile(0.25)  
Q3_weight = std["몸무게(kg)"].quantile(0.75)  
IQR_weight = Q3_weight - Q1_weight  
  
# 이상치 경계 설정  
lower_bound_weight = Q1_weight - 1.5 * IQR_weight  
upper_bound_weight = Q3_weight + 1.5 * IQR_weight  
  
# 이상치 데이터 필터링  
outliers_weight = std[(std["몸무게(kg)"] < lower_bound_weight) | (std["몸무게(kg)"] >  
upper_bound_weight)]  
  
# 이상치 데이터 출력  
print(outliers_weight)
```

	성명	성별	학년	키(cm)	몸무게(kg)	취미
24	학생025	여	2학년	165.5	27.0	수영
32	학생033	남	4학년	166.7	104.7	게임
55	학생056	남	3학년	183.2	112.1	게임
69	학생070	여	2학년	166.3	20.9	독서
122	학생123	남	4학년	160.6	19.7	게임
133	학생134	남	1학년	153.8	106.4	수영
216	학생217	남	2학년	167.9	26.6	수영



## 02 | 단순한 선형 회귀 실습

- ❖ 다음은 **몸무게(kg)** 속성의 **이상치 데이터**를 **클린징**하는 코드이다.
  - ◆ 학생025의 몸무게를 **27.9kg**에서 **60kg**으로 수정함
    - › 학생025의 경우 키가 165.5cm인 것을 감안하여 60kg로 조정함
  - ◆ 학생033의 몸무게를 **104.7kg**에서 **61kg**으로 수정함
    - › 학생033의 경우 키가 166.7cm인 것을 감안하여 61kg으로 조정함

	성명	성별	학년	키(cm)	몸무게(kg)	취미
24	학생025	여	2학년	165.5	27.0	수영
32	학생033	남	4학년	166.7	104.7	게임
55	학생056	남	3학년	183.2	112.1	게임
69	학생070	여	2학년	166.3	20.9	독서
122	학생123	남	4학년	160.6	19.7	게임
133	학생134	남	1학년	153.8	106.4	수영
216	학생217	남	2학년	167.9	26.6	수영

변경전 키와 몸무게



## 02 | 단순한 선형 회귀 실습

- ◆ 학생056의 몸무게를 **112.1kg**에서 **70kg**으로 수정함
  - › 학생056의 경우 키가 183.2cm인 것을 감안하여 70kg로 조정함
- ◆ 학생070의 몸무게를 **20.9kg**에서 **60kg**으로 수정함
  - › 학생070의 경우 키가 166.3cm인 것을 감안하여 60kg으로 조정함
- ◆ 학생123의 몸무게를 **19.7kg**에서 **50kg**으로 수정함
  - › 학생123의 경우 키가 160.6cm인 것을 감안하여 50kg으로 조정함

	성명	성별	학년	키(cm)	몸무게(kg)	취미
24	학생025	여	2학년	165.5	27.0	수영
32	학생033	남	4학년	166.7	104.7	게임
55	학생056	남	3학년	183.2	112.1	게임
69	학생070	여	2학년	166.3	20.9	독서
122	학생123	남	4학년	160.6	19.7	게임
133	학생134	남	1학년	153.8	106.4	수영
216	학생217	남	2학년	167.9	26.6	수영



## 02 | 단순한 선형 회귀 실습

- ◆ 학생134의 몸무게를 **106.4kg**에서 **49kg**으로 수정함
  - › 학생134의 경우 키가 153.8cm인 것을 감안하여 49kg로 조정함
  
- ◆ 학생217의 몸무게를 **26.6kg**에서 **62kg**으로 수정함
  - › 학생217의 경우 키가 167.9cm인 것을 감안하여 62kg으로 조정함

```
std.loc[std['성명'] == '학생025', '몸무게(kg)'] = 60  
std.loc[std['성명'] == '학생033', '몸무게(kg)'] = 61  
std.loc[std['성명'] == '학생056', '몸무게(kg)'] = 70  
std.loc[std['성명'] == '학생070', '몸무게(kg)'] = 60  
std.loc[std['성명'] == '학생123', '몸무게(kg)'] = 50  
std.loc[std['성명'] == '학생134', '몸무게(kg)'] = 49  
std.loc[std['성명'] == '학생217', '몸무게(kg)'] = 62
```

	성명	성별	학년	키(cm)	몸무게(kg)	취미
24	학생025	여	2학년	165.5	27.0	수영
32	학생033	남	4학년	166.7	104.7	게임
55	학생056	남	3학년	183.2	112.1	게임
69	학생070	여	2학년	166.3	20.9	독서
122	학생123	남	4학년	160.6	19.7	게임
133	학생134	남	1학년	153.8	106.4	수영
216	학생217	남	2학년	167.9	26.6	수영

변경전 키와 몸무게

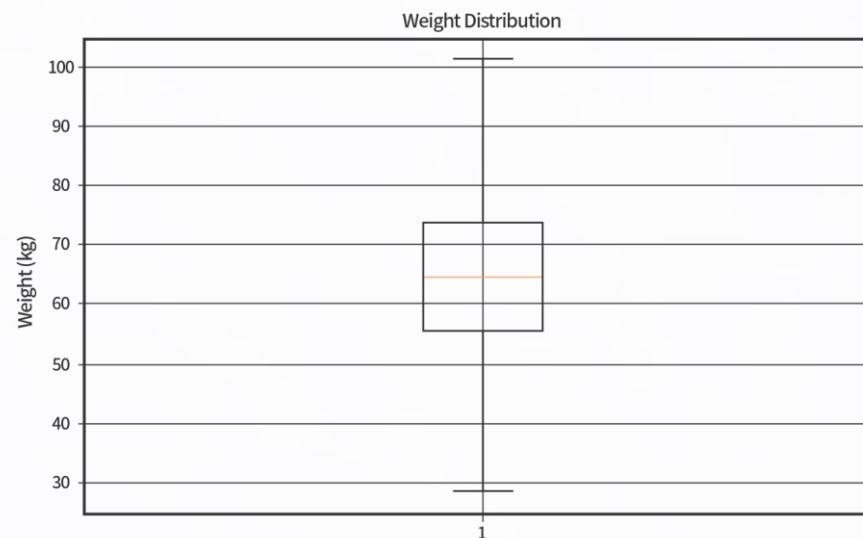


## 02 | 단순한 선형 회귀 실습

❖ 다음은 데이터 클린징을 수행한 몸무게(kg) 속성으로 상자그림을 그린 결과이다.

◆ 실행결과 몸무게 속성에는 이상치 데이터가 없는 것을 알 수 있음

```
plt.figure(figsize=(10, 6))
plt.boxplot(std['몸무게(kg)'])
plt.title('Box Plot of Weight (kg)')
plt.ylabel('Weight (kg)')
plt.grid(True)
plt.show()
```





## 02 | 단순한 선형 회귀 실습

❖ 다음은 대학생 300명의 키와 몸무게 데이터셋의 산점도 그래프이다.

◆ 여기에서 키와 몸무게 평균도 함께 표시함

```
# 몸무게평균  
w_avg = np.mean(std['몸무게(kg)'])  
print('몸무게 평균:', w_avg)  
  
# 키평균  
h_avg = np.mean(std['키(cm)'])  
print('키 평균:', h_avg)  
  
# 키와 몸무게로 산점도 그리기  
plt.scatter(std['키(cm)'], std['몸무게(kg)'])  
plt.title('대학생 300명 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(171,123,"키의 평균")  
plt.text(220,73,"몸무게의 평균")  
plt.show()
```

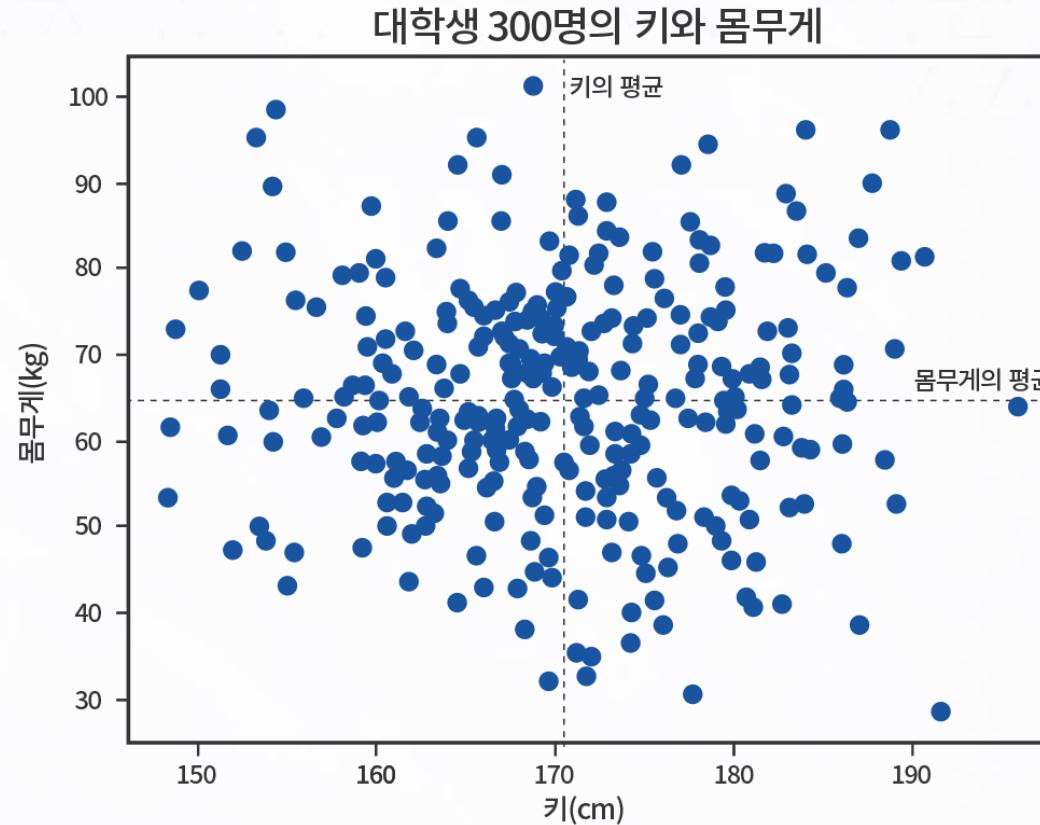


## 02 | 단순한 선형 회귀 실습

- ◆ 아래 그림과 같이 **몸무게 평균은 약 65kg, 키 평균은 약 170cm**인 것을 알 수 있음
  - › 산점도 그래프에서 **키와 몸무게 데이터의 분산이 조금 줄어든 것**을 알 수 있음

몸무게 평균: 64.82100000000001

키 평균: 170.4523333333334





## 02 | 단순한 선형 회귀 실습

▲ 다음은 대학생 300명의 키와 몸무게 데이터 셋으로 산점도에 회귀직선을 추가한다.

◆ 여기에서 키와 몸무게 평균도 함께 표시함

```
# 몸무게 평균  
w_avg = np.mean(std['몸무게(kg)'])  
  
# 키 평균  
h_avg = np.mean(std['키(cm)'])  
  
# x를 설명변수, y를 반응변수로 하는 1차 회귀 곡선(즉 직선을 적합)  
b1, b0 = np.polyfit(std['키(cm)'], std['몸무게(kg)'], 1) # 지울기(=b1), 절편(=b0)을 반환  
print('b0=', b0, 'b1=', b1)  
fit = b0 + b1 * std['키(cm)']  
  
# 키와 몸무게로 산점도, 회귀직선 그리기  
plt.scatter(std['키(cm)'], std['몸무게(kg)']) # 산점도  
plt.plot(std['키(cm)'], fit, color='red') # polyfit() 함수 : 절편, 기울기 계산  
plt.title('대학생 300명 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(171, 100, "키의 평균")  
plt.text(190, 66, "몸무게의 평균")  
plt.show()
```



## 02 | 단순한 선형 회귀 실습

- 실행결과 산점도에 회귀직선이 추가된 것을 볼 수 있음

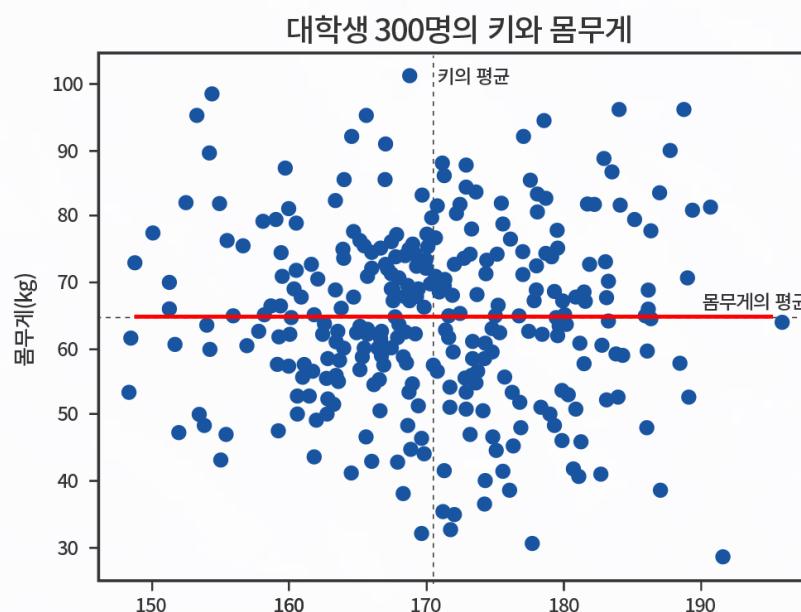
$$\text{학생 몸무게} = \boxed{66.207} - \boxed{0.0081} * \text{학생의 키}$$

절편 계수

몸무게 평균: 64.82100000000001

키 평균: 170.4523333333334

B0= 66.20700325428407 b1= -0.00813132461832365





## 02 | 단순한 선형 회귀 실습

▲ 다음은 대학생 300명의 키와 몸무게 데이터 셋으로 산점도에 회귀직선과 신뢰구간을 그려보자.

◆ 여기에서 키와 몸무게 평균, 신뢰구간은 유의수준 95%로 한다.

```
# 몸무게 평균  
w_avg = np.mean(std['몸무게(kg)'])  
  
# 키 평균  
h_avg = np.mean(std['키(cm)'])  
  
# x를 설명변수, y를 반응변수로 하는 1차 회귀 곡선(즉 직선을 적합)  
b1, b0 = np.polyfit(std['키(cm)'], std['몸무게(kg)'], 1) # 지울기(=b1), 절편(=b0)을 반환  
print('b0=', b0, 'b1=', b1)  
fit = b0 + b1 * std['키(cm)']  
  
# 키와 몸무게로 산점도, 선형회귀선, 95% 신뢰구간 그리기  
plt.scatter(std['키(cm)'], std['몸무게(kg)']) # 산점도  
sns.regplot(x='키(cm)', y='몸무게(kg)', data=std) # 회귀직선  
plt.title('대학생 300명 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(171, 100, "키의 평균")  
plt.text(190, 66, "몸무게의 평균")  
plt.show()
```



## 02 | 단순한 선형 회귀 실습

› 실행결과 산점도에 회귀직선, 신뢰구간(95%)이 추가된 것을 볼 수 있음

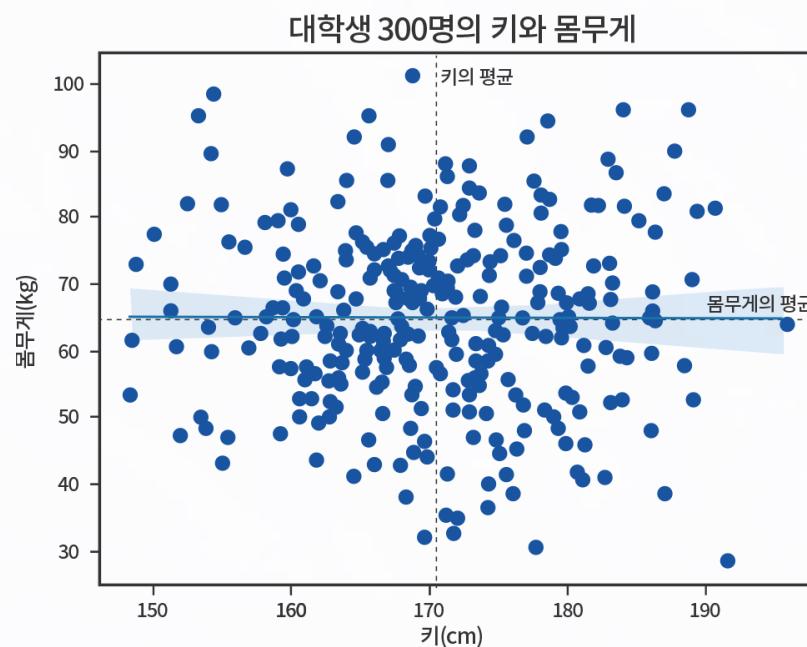
$$\text{학생 몸무게} = \boxed{66.207} - \boxed{-0.0081} * \text{학생의 키}$$

절편 계수

몸무게 평균: 64.82100000000001

키 평균: 170.4523333333334

B0= 66.20700325428407 b1= -0.00813132461832365





## 02 | 단순한 선형 회귀 실습

- ▲ 다음은 대학생 300명의 키와 몸무게 데이터 셋으로 모델 생성 및 학습을 수행하는 코드이다.
    - ◆ 모델 생성 및 학습 결과 다음과 같은 회귀식이 계산된 것을 알 수 있음

```
# 모형 생성 및 학습하기  
model_lr = LinearRegression().fit(np.c_[std['키(cm)']], np.c_[std['몸무게(kg)']])  
  
# 회귀 계수 : 절편, 기울기  
print("intercept=", model_lr.intercept_) # 절편 intercept= [66.20700325]  
print("coef=", model_lr.coef_) # 기울기(계수) coef= [[-0.00813132]]  
  
# 학생 몸무게(kg) = 66.20700325 - 0.00813132 * 학생의 키(cm)
```



## 02 | 단순한 선형 회귀 실습

❖ 다음은 학습된 모델로 모델 성능평가를 수행하는 코드이다.

◆ 여기서는 모델 성능평가 지표로 MSE를 이용함

› 실행결과 MSE = 186.8135 인 것을 볼 수 있다.

```
# 모델의 성능 확인  
mse = mean_squared_error(y_true = std['몸무게(kg)'], y_pred = model_lr.predict(np.c_[std['키(cm)']]))  
mse      # 186.81357035381873
```



## 02 | 단순한 선형 회귀 실습

- ❖ 다음은 학습된 모델로 예측을 수행하는 코드이다.
  - ◆ 여기서는 새로운 학생의 키가 175cm 임
    - > 실행결과 몸무게가 약 64.78(kg) 인 것을 볼 수 있음

```
X_new = [[175]]          # 새로운 학생의 키(cm) = 175
print("Predict=", model_lr.predict(X_new)) # 새로운 학생 키에 대한 예측 결과 = [[64.78402145]]
```



## 02 | 단순한 선형 회귀 실습

❖ 다음은 학습된 두 모델로 예측을 수행한 결과를 비교한 것이다.

◆ 여기서는 새로운 학생의 키가 175cm 임

이상치 데이터를 제거하지 않은 경우

모델이 예측한 몸무게 약 71.02(kg)

모델 성능 평가 결과 MSE = 267.791

이상치 데이터를 제거한 경우

모델이 예측한 몸무게 약 64.78(kg)

모델 성능 평가 결과 MSE 186.8135

두 모델의 예측결과는 약 6.24(kg) 차이