

Module6

Regression



◆ 학습목표

Chi-Square test 및 Regression 분석방법을 학습하고, 통계적 분석과 기계학습 방법의 차이에 대해 학습한다.

-
- I. Chi-Square test
 - II. AB test
 - III. Regression
 - IV. Logistic Regression
 - V. 통계적 분석과 기계학습 분석 방법 비교
-

I. Chi-Square test

Chi-Square test란

❖ 동질성 검정

- 집단간 분포의 동질성(Homogeneity of proportions)을 통해 두 집단간의 차이를 검정
- 사전실험설계: 사전에 그룹의 수를 결정해서 연구할 때
- 코호트연구: 상대적 위험율 (Relative Risk)

❖ 독립성 검정

- 변수들 간의 독립성(independence of variables) 또는 관련성을 검정
- 사후사례대조: 사후의 결과를 토대로 연구할 때
- 사례-대조연구: 오즈비 (Odds ratio)

❖ 적합도 검정

- 각 범주에 속할 확률이 특정 값 또는 이론적 확률과 어느 정도 일치하는지를 검정

❖ 동질성 검정

- 사전(실험)설계일 때: 사전에 그룹의 수를 결정
- 그룹에 따른 차이를 연구할 때
- 예) 비타민과 감기에 대한 연구를 하기 위해, 비타민을 투여할 실험군과 가짜약을 투여할 대조군으로 사전에 구분하여 연구
- 비율기준: 그룹별 자료수
- 관심: 실험군 감기발생 비율 = 대조군 감기 발생 비율

사후

사전

그룹	감기발병		합계
	유	무	
실험군 (비타민)	17 (34.0%)	33 (66.0%)	50 (100.0%)
대조군(Placebo)	38 (76.0%)	12 (24.0%)	50 (100.0%)
합계	55 (55.0%)	45 (45.0%)	100 (100.0%)

$$\frac{17}{50} = \frac{38}{50} \text{ or } \frac{17}{50} \neq \frac{38}{50}$$

$$\frac{33}{50} = \frac{12}{50} \text{ or } \frac{33}{50} \neq \frac{12}{50}$$

❖ 독립성 검정

- 사후사례대조: 사후의 결과를 토대로 연구할 때
- 두 변수간의 관련성을 연구할 때
- 예) 흡연이 폐암과 연관이 있는지를 연구하기 위해 흡연자와 비흡연자를 대상으로 폐암발생여부를 사후에 조사
- 비율기준: 전체 자료수

사전

폐암	흡연					합계
	비흡연군	장기금연군	단기금연군	재흡연군	흡연군	
무	170,867 (52.0%)	51,690 (15.7%)	46,598 (14.2%)	29,178 (8.9%)	27,784 (8.5%)	326,117 (99.3%)
	723 (0.2%)	370 (0.1%)	497 (0.2%)	319 (0.1%)	504 (0.2%)	2,413 (0.7%)
합계	171,590 (52.2%)	52,060 (15.8%)	47,095 (14.4%)	29,497 (9.0%)	28,288 (8.6%)	328,530 (100.0%)

$$\frac{170,867}{328,530} = \frac{170,329}{328,530} \text{ or } \frac{170,867}{328,530} \neq \frac{170,329}{328,530}$$

출처: 한국인에서 흡연과 폐암의 상관관계 및 폐암의 위험인자 분석, 국민건강보험 일산병원 연구소 (2016)

동질성 검정

❖ 문제의 정의

- K 병원에서는 비타민과 감기와의 관계를 연구하고자 한다.
- 감기가 걸리지 않은 사람을 대상으로 비타민을 투여할 실험군과 가짜약을 투여할 대조군으로 구분하고 겨울동안 감기가 걸렸는지를 확인하였다.
- 과연 비타민이 감기에 효과가 있었는지 검증해 보라
- 15_1.PreCH.csv

❖ 가설

- 귀무가설(H_0): 실험군과 대조군 간에는 감기발생 차이가 없다.

$$H_0: \theta_{11} = \theta_{21}, \theta_{12} = \theta_{22} \quad \theta = \frac{\theta_{ij}}{n_i}$$

- 연구가설(H_1): 실험군과 대조군 간에는 감기발생 차이가 있다.

$$H_1: \theta_{11} \neq \theta_{21}, \theta_{12} \neq \theta_{22}$$

❖ 분할표

모집단	범주		계
	1	2	
1	$O_{11} \left(\frac{o_{11}}{n_1} \right)$	$O_{12} \left(\frac{o_{12}}{n_1} \right)$	$n_1 (100\%)$
2	$O_{21} \left(\frac{o_{21}}{n_1} \right)$	$O_{22} \left(\frac{o_{22}}{n_2} \right)$	$n_2 (100\%)$
계	M_1	M_2	n

❖ 가설

$$\frac{o_{11}}{n_1} = \frac{o_{21}}{n_2} \text{ or } \frac{o_{11}}{n_1} \neq \frac{o_{21}}{n_2}$$

$$\frac{o_{12}}{n_1} = \frac{o_{22}}{n_2} \text{ or } \frac{o_{12}}{n_1} \neq \frac{o_{22}}{n_2}$$

❖ 분할표

		감기발병		Total
실험처치		정상	감기	
실험군	Observed	33	17	50
	Expected	22.500	27.500	50.000
대조군	Observed	12	38	50
	Expected	22.500	27.500	50.000
Total	Observed	45	55	100
	Expected	45.000	55.000	100.000

❖ 기대도수

$$\hat{E}_{11} = \frac{50 \times 45}{100} = 22.5 \quad \hat{E}_{21} = \frac{50 \times 55}{100} = 27.5$$

$$\hat{E}_{21} = \frac{50 \times 45}{100} = 22.5 \quad \hat{E}_{22} = \frac{50 \times 55}{100} = 27.5$$

❖ 검정통계량

$$\sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - \hat{E}_{ij})^2}{\hat{E}_{ij}} \sim \chi^2_{(r-1)(c-1)}$$

❖ 사후분석

- 피어슨 잔차 → 수정잔차

$$R_{ij}^{(p)} = \frac{(O_{ij} - \hat{E}_{ij})}{\sqrt{\hat{E}_{ij}}} \quad R_{ij}^{(p)} = \frac{O_{ij} - \hat{E}_{ij}}{\sqrt{\hat{E}_{ij}(1 - p_i)(1 - \hat{\theta}_j)}}$$

❖ 검정통계량

$$\sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - \hat{E}_{ij})^2}{\hat{E}_{ij}} = \frac{(33 - 22.5)^2}{22.5} + \dots + \frac{(38 - 27.5)^2}{27.5} = 17.818 > \chi_1^2 = 3.84$$

❖ 사후분석

- 피어슨 잔차 → 수정잔차

$$\begin{aligned} R_{11}^{(1)} &= \frac{(O_{11} - \hat{E}_{11})}{\sqrt{\hat{E}_{11}}} \\ &= \frac{(33 - 22.5)}{\sqrt{22.5}} \\ &= 2.2 \end{aligned}$$

$$\begin{aligned} R_{11}^{(1)} &= \frac{O_{11} - \hat{E}_{11}}{\sqrt{\hat{E}_{11}(1 - p_i)(1 - \hat{\theta}_j)}} \\ &= \frac{(33 - 22.5)}{\sqrt{22.5 \left(1 - \frac{50}{100}\right) \left(1 - \frac{45}{100}\right)}} \\ &= 4.2 > 1.96 \end{aligned}$$

❖ 상대적 위험율 (Relative Risk)

- 비타민은 복용하면 감기에 걸릴 확률이 그렇지 않은 사람에 비해 2.57배 높게 나타남

그룹	감기발병		합계
	유	무	
실험군 (비타민)	17 (34.0%)	33 (66.0%)	50 (100.0%)
대조군(Placebo)	38 (76.0%)	12 (24.0%)	50 (100.0%)
합계	55 (55.0%)	45 (45.0%)	100 (100.0%)

$$RR = \frac{\frac{O_{11}}{n_1}}{\frac{O_{21}}{n_2}} = \frac{\frac{33}{50}}{\frac{12}{50}} = \frac{0.66}{0.24} = 2.75$$

15_1.Chi-square test(동질성)

LGE Internal Use Only

- ▼ 15_1.Chi-square test(동질성)
 - https://www.statsmodels.org/devel/contingency_tables.html

- ▼ 1.기본 package 설정

```
[ ] # 그래프에서 한글 폰트 인식하기  
!sudo apt-get install -y fonts-nanum  
!sudo fc-cache -fv  
!rm ~/.cache/matplotlib -rf
```

```
[ ] !pip install pingouin
```

```
# *** 런타임 다시 시작
```

```
✓ 2초 [2] # 1.기본  
import numpy as np # numpy 패키지 가져오기  
import matplotlib.pyplot as plt # 시각화 패키지 가져오기  
import seaborn as sns # 시각화  
  
# 2.데이터 가져오기  
import pandas as pd # csv -> dataframe으로 전환  
  
# 3.통계분석 package  
import pingouin as pg  
from scipy import stats  
import statsmodels.api as sm
```

```
✓ 0초 [2] # 기본세팅  
# 테마 설정  
sns.set_theme(style = "darkgrid")
```



2.데이터 불러오기

LGE Internal Use Only

2.데이터 불러오기

2.1 데이터 프레임으로 저장

- 원본데이터(csv)를 dataframe 형태로 가져오기(pandas)

[3] prech_df = pd.read_csv('https://raw.githubusercontent.com/leecho-bigdata/statistics-python/main/15_1.PreCH.csv', encoding="cp949")
prech_df.head()

	실험처치	감기발병
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1

Next steps: View recommended plots

2.2 범주형 변수 처리

- 가변수 처리시 문자로 처리를 해야 변수명 구분이 쉬움

[4] prech_df[['실험처치']].replace({1:'실험군', 2:'대조군'}, inplace=True)
prech_df[['감기발병']].replace({1:'정상', 2:'감기'}, inplace=True)
prech_df[['실험처치']] = prech_df[['실험처치']].astype('category')
prech_df[['감기발병']] = prech_df[['감기발병']].astype('category')
prech_df

	실험처치	감기발병
0	실험군	정상

✓ 0초 오후 9:07에 완료됨

3.Chi-square test(동질성)

LGE Internal Use Only

```
✓ 3.Chi-square test(동질성)
  ✓ 3.1 분할표(contingency table)

[8] tab = pd.crosstab(prech_df['실험처치'], prech_df['감기발병'])
    tab

    감기발병 감기 정상
    실험처치
    대조군    38   12
    실험군    17   33

Next steps: View recommended plots

✓ [9] # 위치 조정
    tab = tab.loc[['실험군', '대조군'], :]
    tab = tab.loc[:, ['정상', '감기']]

✓ [10] tab

    감기발병 정상 감기
    실험처치
    실험군    33   17
    대조군    12   38

Next steps: View recommended plots

  ✓ 3.2 교차분석

✓ 0초 오후 9:07에 완료됨
```

3.Chi-square test(동질성)

LGE Internal Use Only

```
▼ 3.2 교차분석
[11] # ch분석
      result = sm.stats.Table(tab)

[12] # observed
      print(result.table_orig)

      감기발병 정상 감기
      실험처치
      실험군 33 17
      대조군 12 38

[13] # expected
      print(result.fittedvalues)

      감기발병 정상 감기
      실험처치
      실험군 22.5 27.5
      대조군 22.5 27.5

[14] # Pearson
      rslt = result.test_nominal_association()
      print(rslt.pvalue)

      2.4304960694832012e-05

▼ 3.3 표준화잔차
[15] # 표준화 잔차
      result.standardized_resids

      감기발병 정상 감기
      실험처치
      실험군 1.221159 -1.221159
      대조군 0.778840 1.778840
      ✓ 0초 오후 9:07에 완료됨
```

3.Chi-square test(동질성)

LGE Internal Use Only

```
✓ 3.3 표준화잔차
[15] # 표준화 잔차
      result.standardized_resids
      감기발병 정상 감기
      실험처치
      실험군 4.221159 -4.221159
      대조군 -4.221159 4.221159

✓ 3.4 상대적 위험률
[16] table = np.asarray(tab)
      table
      array([[33, 17],
             [12, 38]])

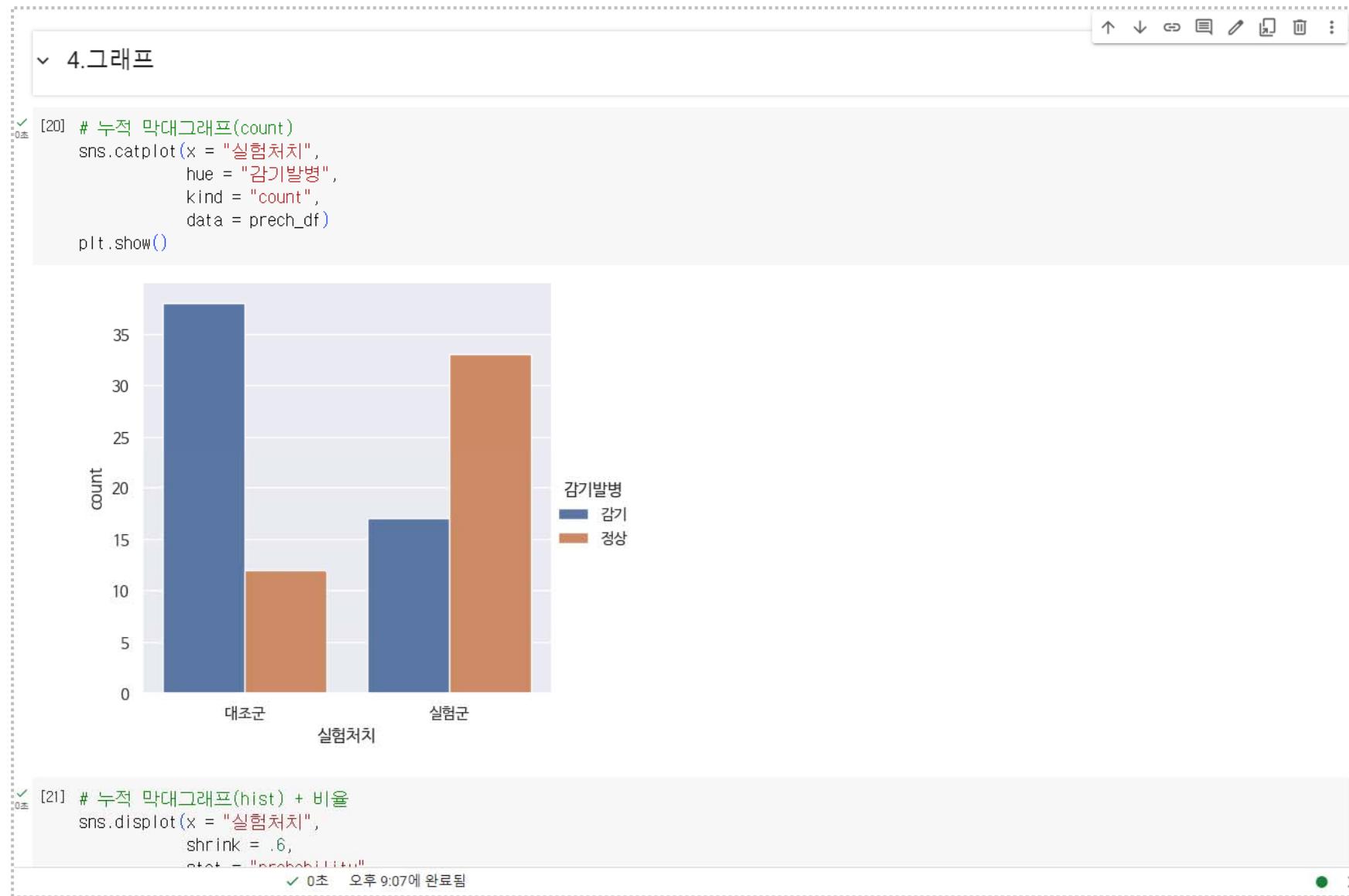
[17] table
      array([[33, 17],
             [12, 38]])

[18] t22 = sm.stats.Table2x2(table)

[19] print(t22.summary())
      Estimate SE LCB UCB p-value
      -----
      Odds ratio 6.147 2.565 14.729 0.000
      Log odds ratio 1.816 0.446 0.942 2.690 0.000
      Risk ratio 2.750 1.616 4.681 0.000
      Log risk ratio 1.012 0.271 0.480 1.543 0.000

✓ 0초 오후 9:07에 완료됨
```

4.그래프



- ❖ 비타민을 투여한 그룹이 그렇지 않은 그룹에 비해 감기발병이 적은 것으로 나타났다($\chi^2 = 17.82, < .001$). 비타민 복용이 어느 정도 영향을 주는지를 분석한 결과, 비타민을 복용하면 감기가 안 걸릴 가능성이 2.75배 높은 것으로 나타났다.

	감기 발병률			χ^2	p	RR
	무	유	전체			
비타민	33 (22.5)	17 (34.0)	50 (100.0)	17.82	<.001	2.75
Placebo	12 (24.0)	38 (76.0)	50 (100.0)			
전체	45 (45.0)	55 (55.0)	100 (100.0)			
RR = Relative Risk						

독립성 검정

❖ 문제의 정의

- K병원에서는 흡연을 많이 하는 사람일수록 폐암에 걸릴 확률이 높다는 것을 발표하였다. 과연 흡연이 폐암과 연관이 있는지를 검증해 보자.
- 흡연자와 비흡연자를 대상으로 폐암발생여부를 파악하여 분석하고자 한다. 이때 국립보건소를 통해 다음과 같은 2차 자료를 구했다고 하자.
- 흡연을 많이 하면 폐암이 걸리는지를 검증해 보자.
- 16_1.PostCH.csv

❖ 가설

- 귀무가설(H_0) : 흡연유무와 폐암유무는 서로 독립적이다.
- 연구가설(H_1) : 흡연유무와 폐암유무는 서로 독립적이지 않다.

❖ 분할표

모집단	범주					계
	1	2	3	4	5	
1	$O_{11} \left(\frac{o_{11}}{n} \right)$	$O_{12} \left(\frac{o_{12}}{n} \right)$	$O_{13} \left(\frac{o_{13}}{n} \right)$	$O_{14} \left(\frac{o_{14}}{n} \right)$	$O_{15} \left(\frac{o_{15}}{n} \right)$	n_1
2	$O_{21} \left(\frac{o_{21}}{n} \right)$	$O_{22} \left(\frac{o_{22}}{n} \right)$	$O_{23} \left(\frac{o_{23}}{n} \right)$	$O_{24} \left(\frac{o_{24}}{n} \right)$	$O_{25} \left(\frac{o_{25}}{n} \right)$	n_2
계	M_1	M_2	M_3	M_4	M_5	n

❖ 검정통계량

$$\sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - \hat{E}_{ij})^2}{\hat{E}_{ij}} \sim \chi^2_{(r-1)(c-1)}$$

❖ 사후분석

- 피어슨 잔차 → 수정잔차

$$R_{ij}^{(p)} = \frac{(O_{ij} - \hat{E}_{ij})}{\sqrt{\hat{E}_{ij}}}$$

$$R_{ij}^{(p)} = \frac{O_{ij} - \hat{E}_{ij}}{\sqrt{\hat{E}_{ij}(1 - \hat{\theta}_{i+})(1 - \hat{\theta}_{+j})}}$$

❖ 분할표

Contingency Tables						
		흡연				
폐암		비흡연	장기흡연	단기흡연	재흡연	흡연
정상	Observed	170867	51690	46598	29178	27784
	Expected	170329.699	51677.628	46749.095	29280.349	28080.229
폐암	Observed	723	370	497	319	504
	Expected	1260.301	382.372	345.905	216.651	207.771
Total	Observed	171590	52060	47095	29497	28288
	Expected	171590	52060	47095	29497	28288

❖ 기대도수

$$\hat{E}_{11} = \frac{326,117 \times 171,590}{328,530} = 170,329.7$$

❖ 검정통계량

$$\sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - \hat{E}_{ij})^2}{\hat{E}_{ij}} = \frac{(170,867 - 170,330)^2}{170,330} + \dots = 771.835 > \chi^2_4 = 9.49$$

❖ 사후분석

$$R_{11}^{(1)} = \frac{(170,867 - 170,330)}{\sqrt{170,330 \left(1 - \frac{326,117}{328,530}\right) \left(1 - \frac{171,590}{328,530}\right)}} = 22.0$$

❖ 오즈비(odds ratio)

- 위험요인과 질병 발생간의 연관성을 1을 기준으로 나타낸 척도

폐암	흡연					합계
	비흡연군	장기금연군	단기금연군	재흡연군	흡연군	
무	170,867 (52.0%)	51,690 (15.7%)	46,598 (14.2%)	29,178 (8.9%)	27,784 (8.5%)	326,117 (99.3%)
유	723 (0.2%)	370 (0.1%)	497 (0.2%)	319 (0.1%)	504 (0.2%)	2,413 (0.7%)
합계	171,590 (52.2%)	52,060 (15.8%)	47,095 (14.4%)	29,497 (9.0%)	28,288 (8.6%)	328,530 (100.0%)

$$OR = \frac{\frac{O_{52}}{O_{51}}}{\frac{O_{12}}{O_{11}}} = \frac{\frac{504}{27,784}}{\frac{723}{170,867}} = \frac{0.0181}{0.0042} = 4.29$$

16_1.Chi-square test(독립성)

LGE Internal Use Only

▼ 16_1.Chi-square test(독립성)

- https://www.statsmodels.org/devel/contingency_tables.html

▼ 1.기본 package 설정

```
[ ] # 그래프에서 한글 폰트 인식하기  
!sudo apt-get install -y fonts-nanum  
!sudo fc-cache -fv  
!rm ~/.cache/matplotlib -rf
```

```
[ ] !pip install pingouin
```

```
# *** 런타임 다시 시작
```

2초 # 1.기본

```
import numpy as np # numpy 패키지 가져오기  
import matplotlib.pyplot as plt # 시각화 패키지 가져오기  
import seaborn as sns # 시각화  
  
# 2.데이터 가져오기  
import pandas as pd # csv -> dataframe으로 전환  
  
# 3.통계분석 package  
import pingouin as pg  
from scipy import stats  
import statsmodels.api as sm
```

0초 [2] # 기본세팅

```
# 테마 설정  
sns.set_theme(style = "darkgrid")
```



2.데이터 불러오기

LGE Internal Use Only

The screenshot shows a Jupyter Notebook interface with the following content:

- Section 2.1:** 2.1 데이터 프레임으로 저장
 - 원본데이터(csv)를 dataframe 형태로 가져오기(pandas)
- Code Cell 3:** [3] postch_df = pd.read_csv('https://raw.githubusercontent.com/leecho-bigdata/statistics-python/main/16_1.PostCH.csv', encoding="cp949")
postch_df.head()

A preview of the DataFrame is shown:

	폐암	흡연	관측치
0	1	1	170867
1	1	2	51690
2	1	3	46598
3	1	4	29178
4	1	5	27784
- Section 2.2:** 2.2 범주형 변수 처리
 - 가변수 처리시 문자로 처리를 해야 변수명 구분이 쉬움
- Code Cell 4:** [4] postch_df['폐암'].replace({1:'정상', 2:'폐암'}, inplace=True)
postch_df['흡연'].replace({1:'비흡연', 2:'장기흡연', 3:'단기흡연', 4:'재흡연', 5:'흡연'}, inplace=True)
postch_df['폐암'] = postch_df['폐암'].astype('category')
postch_df['흡연'] = postch_df['흡연'].astype('category')
postch_df

A preview of the cleaned DataFrame is shown:

	폐암	흡연	관측치
0	정상	비흡연	170867

2.데이터 불러오기

LGE Internal Use Only

Next steps: [View recommended plots](#)

▼ 2.2 범주형 변수 처리

- 가변수 처리시 문자로 처리를 해야 변수명 구분이 쉬움

[4] 0초 postch_df['폐암'].replace({1:'정상', 2:'폐암'}, inplace=True)
postch_df['흡연'].replace({1:'비흡연', 2:'장기흡연', 3:'단기흡연', 4:'재흡연', 5:'흡연'}, inplace=True)
postch_df['폐암'] = postch_df['폐암'].astype('category')
postch_df['흡연'] = postch_df['흡연'].astype('category')
postch_df

	폐암	흡연	관측치
0	정상	비흡연	170867
1	정상	장기흡연	51690
2	정상	단기흡연	46598
3	정상	재흡연	29178
4	정상	흡연	27784
5	폐암	비흡연	723
6	폐암	장기흡연	370
7	폐암	단기흡연	497
8	폐암	재흡연	319
9	폐암	흡연	504

Next steps: [View recommended plots](#)

▼ 2.3 자료구조 살펴보기

[5] 0초 postch_df.shape
(10, 3)

✓ 0초 오후 9:12에 완료됨

3.Chi-square test(독립성)

LGE Internal Use Only

```
▼ 3.Chi-square test(독립성)
  ▼ 3.1 분할표(contingency table)

  ✓ [8] tab = pd.crosstab(postch_df['폐암'],
                         postch_df['흡연'],
                         values = postch_df['관측치'],
                         aggfunc = 'sum')

  ✓ [9] # 위치 조정
        tab = tab.loc[:, ["비흡연", "장기흡연", "단기흡연", "재흡연", "흡연"]]
        tab

        흡연 비흡연 장기흡연 단기흡연 재흡연 흡연
        폐암
        정상 170867 51690 46598 29178 27784
        폐암 723 370 497 319 504

        Next steps:  View recommended plots

  ▼ 3.2 교차분석

  ✓ [10] # ch분석
        result = sm.stats.Table(tab)

  ✓ [11] # observed
        print(result.table_orig)

        흡연 비흡연 장기흡연 단기흡연 재흡연 흡연
        폐암
        정상 170867 51690 46598 29178 27784
        폐암 723 370 497 319 504

scr... ✓ 0초 오후 9:12에 완료됨
```

3.Chi-square test(독립성)

```

0초 [11] # observed
print(result.table_orig)



|    | 흡연     | 비흡연   | 장기흡연  | 단기흡연  | 재흡연   | 흡연 |
|----|--------|-------|-------|-------|-------|----|
| 폐암 |        |       |       |       |       |    |
| 정상 | 170867 | 51690 | 46598 | 29178 | 27784 |    |
| 폐암 | 723    | 370   | 497   | 319   | 504   |    |


0초 [12] # expected
print(result.fittedvalues)



|    | 흡연            | 비흡연          | 장기흡연         | 단기흡연        | 재흡연          | 흡연 |
|----|---------------|--------------|--------------|-------------|--------------|----|
| 폐암 |               |              |              |             |              |    |
| 정상 | 170329.699053 | 51677.627675 | 46749.094801 | 29280.34928 | 28080.229191 |    |
| 폐암 | 1260.300947   | 382.372325   | 345.905199   | 216.65072   | 207.770809   |    |


0초 [13] # Pearson
rslt = result.test_nominal_association()
print(rslt.pvalue)

0.0

▼ 3.3 표준화잔차

0초 [14] # 표준화 잔차
result.standardized_resids



|    | 흡연         | 비흡연       | 장기흡연      | 단기흡연      | 재흡연        | 흡연 |
|----|------------|-----------|-----------|-----------|------------|----|
| 폐암 |            |           |           |           |            |    |
| 정상 | 21.978698  | 0.692265  | -8.809885 | -7.315323 | -21.576857 |    |
| 폐암 | -21.978698 | -0.692265 | 8.809885  | 7.315323  | 21.576857  |    |


0초 오후 9:12에 완료됨

```

3.Chi-square test(독립성)

```

✓ 0초 [14] 정상 21.978698 0.692265 -8.809885 -7.315323 -21.576857
    폐암 -21.978698 -0.692265 8.809885 7.315323 21.576857

    ▾ 3.4 odds ratio

✓ 0초 [15] tab = tab.loc[:, ["비흡연", "흡연"]]
tab

    흡연 비흡연 흡연
    폐암
    정상 170867 27784
    폐암 723 504

Next steps: View recommended plots

✓ 0초 ⏴ table = np.asarray(tab)
table
array([[170867, 27784],
       [ 723,   504]])

✓ 0초 [17] t22 = sm.stats.Table2x2(table)

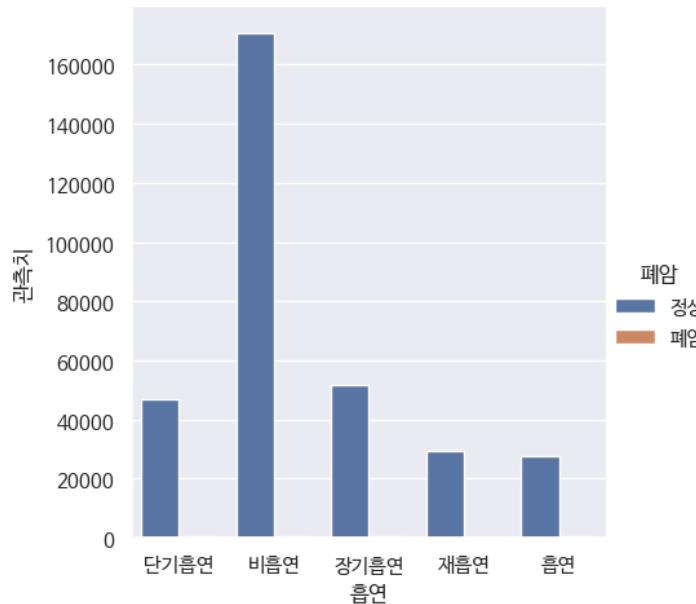
✓ 0초 [18] print(t22.summary())
      Estimate SE LCB UCB p-value
-----
Odds ratio      4.287  3.823 4.807  0.000
Log odds ratio  1.456  0.058 1.341 1.570  0.000
Risk ratio      1.460  1.393 1.530  0.000
Log risk ratio  0.378  0.024 0.332 0.425  0.000
-----
```

✓ 0초 오후 9:12에 완료됨

4.그래프

4.그래프

```
[19] sns.catplot(x = "흡연",
                 y = "관측치",
                 hue = "폐암",
                 kind = "bar",
                 data = postch_df)
plt.show()
```



```
[19]
```

✓ 0초 오후 9:12에 완료됨

- ❖ 흡연 유무와 폐암발생 유무와 관련이 있는지를 검증한 결과, 흡연 유무와 폐암 발생 유무와 관련이 있는 것으로 나타났다($\chi^2 = 79.71, <.001$). 흡연 유무가 폐암 발생에 어느 정도 영향을 주는지를 분석한 결과, 흡연을 하면 흡연을 하지 않은 사람에 비해 폐암에 걸릴 가능성이 4.29배나 높은 것으로 나타났다.

폐암	흡연					합계	χ^2	p
	비흡연군	장기금연군	단기금연군	재흡연군	흡연군			
무	170,867 (52.0%)	51,690 (15.7%)	46,598 (14.2%)	29,178 (8.9%)	27,784 (8.5%)	326,117 (99.3%)	771.835	0.000
유	723 (0.2%)	370 (0.1%)	497 (0.2%)	319 (0.1%)	504 (0.2%)	2,413 (0.7%)		
합계	171,590 (52.2%)	52,060 (15.8%)	47,095 (14.4%)	29,497 (9.0%)	28,288 (8.6%)	328,530 (100.0%)		
OR		1.69	2.52	2.58	4.29			

적합도 검정

❖ 적합도 검정

- 관찰된 도수가 어떤 측정한 이론분포를 따르는가를 검정
- 재배한 강낭콩이 멘델의 유전법칙을 따르는가?
- 여성운전자의 자동차 구매경향이 일반인의 구매경향과 일치하는가?

	관찰수	자동차 판매비율	기대도수
소나타	131	0.46	138
그랜저	75	0.22	66
아이오닉	29	0.13	39
아반떼	52	0.15	45
제네시스	13	0.04	12
계	300	100	300

❖ 가설

- 귀무가설 : 여성운전자의 자동차 구매경향이 일반인의 구매경향과 일치한다

$$H_0: (p_1, p_2, \dots, p_k) = (p_{10}, p_{20}, \dots, p_{k0})$$

- 연구가설 : 적어도 하나의 p_i 는 가정된 p_{i0} 와 다르다

❖ 검정통계량

$$\sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - \hat{E}_{ij})^2}{\hat{E}_{ij}} = 5.32 < \chi^2_{0.05,4} = 9.49$$

$$p-value = 0.256 > \alpha = 0.05$$

❖ 문제의 정의

- G텔레콤의 고객 이탈율은 9%이다. 500명의 고객을 샘플로 이탈가능성을 조사하였다.
- 500명 중 50명이 앞으로 이탈할 것으로 나타났다.
- 유의수준 5%로 고객 이탈율이 9%라고 할 수 있는가?

	정상	이탈	합계
관찰수	450	50	500
이탈율	0.91	0.09	1

- 교차분석(적합도 검정)

$$\sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - \hat{E}_{ij})^2}{\hat{E}_{ij}} = 0.611 < \chi^2_4 = 3.84 \quad * Z \sim N(0,1) \Rightarrow Z^2 \sim \chi^2_1$$

- 모비율분석

$$z_{cal} = \frac{p - \pi_0}{\sqrt{\frac{p(1-p)}{n}}} = \frac{0.10 - 0.09}{\sqrt{\frac{0.100(1-0.100)}{500}}} = \frac{0.01}{\sqrt{0.0013}} = 0.745$$

$$p-value = P(|z| > 0.745) = 0.434$$

17_1.Chi-square test(적합성)

LGE Internal Use Only

- ✓ 17_1.Chi-square test(적합성)
 - https://www.statsmodels.org/devel/contingency_tables.html

- ✓ 1.기본 package 설정

```
3초 [1] # 1.기본
import numpy as np # numpy 패키지 가져오기
import matplotlib.pyplot as plt # 시각화 패키지 가져오기
import seaborn as sns # 시각화

# 2.데이터 가져오기
import pandas as pd # csv -> dataframe으로 전환

# 3.통계분석 package
#import pingouin as pg
from scipy import stats
import statsmodels.api as sm
```

- ✓ 2.데이터 불러오기

- ✓ 2.1 데이터 프레임으로 저장

- 원본데이터(csv)를 dataframe 형태로 가져오기(pandas)

```
0초 [2] god_df = pd.read_csv('https://raw.githubusercontent.com/leecho-bigdata/statistics-python/main/17_1.GofCH.csv', encoding="cp949")
god_df
```

종류	관찰수	판매비율	선택
0	1	131	0.46

✓ 0초 오전 10:33에 완료됨



2.데이터 불러오기

LGE Internal Use Only

The screenshot shows a Jupyter Notebook interface with the following content:

- Section 2.데이터 불러오기 (2. Data Import)**
 - Section 2.1 데이터 프레임으로 저장 (2.1 Save as DataFrame)**
 - 원본데이터(csv)를 dataframe 형태로 가져오기(pandas)
 - Code cell 1:** god_df = pd.read_csv('https://raw.githubusercontent.com/leecho-bigdata/statistics-python/main/17_1_GofCH.csv', encoding="cp949")
god_df
 - Data Preview:** A table showing the first 5 rows of the DataFrame.

종류	관찰수	판매비율
0	131	0.46
1	75	0.22
2	29	0.13
3	52	0.15
4	13	0.04
 - Next steps:** View recommended plots
- Section 3.Chi-square test(적합도) (3. Chi-square test (Goodness of Fit))**
 - Section 3.1 분할표(contingency table) (3.1 Contingency Table)**
 - Code cell 2:** [3] observed = god_df['관찰수']
expected = god_df['관찰수'].sum() * god_df['판매비율']
 - Code cell 3:** [4] np.array(observed)
np.array(expected)
array([138., 66., 39., 45., 12.])

3.Chi-square test(적합도)

LGE Internal Use Only

The screenshot shows a Jupyter Notebook cell with the following content:

```
[3]: observed = god_df['관찰수']
      expected = god_df['관찰수'].sum() * god_df['판매비율']

[4]: np.array(observed)
      np.array(expected)

array([138.,  66.,  39.,  45.,  12.])

[5]: stats.chisquare(f_obs=observed, f_exp=expected)

Power_divergenceResult(statistic=5.318669977365629, pvalue=0.2561342507779879)
```

At the bottom of the cell, there is a status bar with the text "✓ 0초 오전 10:33에 완료됨". The cell has a standard Jupyter interface with a toolbar at the top and a scroll bar on the right.

연습문제

연습문제1

❖ 문제의 정의

- 압박붕대 사용에 대한 실험연구 (Callam et.al, 1992)
- Bandage: 1:elastic, 2: inelastic
- Healed : 1:Yes, 2>No
- 1.압방붕대 사용이 치료에 효과가 있었는가?
- 2.압박붕대 사용했을 때와 사용하지 않았을 때 상대위험률은 얼마인가?
- 15_2.bandage.csv

Bandage	Healed		Total
	Yes	No	
Elastic	35	30	65
Inelastic	19	48	67
Total	54	78	132

출처: https://www-users.york.ac.uk/~mb55/yh_stats/chiodds.htm#table4

연습문제2

❖ 문제의 정의

- 2006년 "약물 사용: 대학생 운동선수의 약물 사용에 대한 NCAA 연구 보고서"
- 1.운동선수의 스테로이드 사용과 Division과 관계가 있는가?
- 2.Division1과 Division3의 odds ratio는 얼마인가?
- 16_2.ncaa.csv

Steroid Use	Division			Total
	Division I	Division II	Division III	
Yes	103	52	65	220
No	8,440	4,289	6,428	19,157
Total	8,543	4,341	6,493	19,377

출처: <https://courses.lumenlearning.com/wmopen-concepts-statistics/chapter/test-of-homogeneity/>

연습문제3

❖ 문제의 정의

- 캐나다 하키 선수가 되기 위해서는 빠른 생일이 유리한가?
- 2008~2009 시즌 Ontario Hockey League (OHL)의 선수 분포(15~20세)은 캐나다의 전체 정상 출산 비율(1989년)과 같다고 말할 수 있는가?
- 17_2ohl.csv

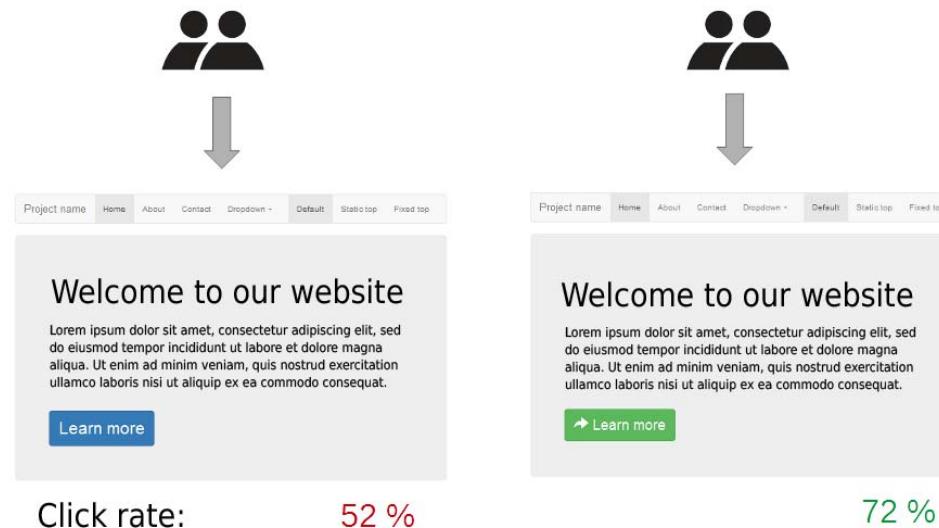
Steroid Use	quarter				Total
	Q1	Q2	Q3	Q4	
OHL players	147	110	52	50	359
% of Canadian births	23.7%	25.9%	25.9%	24.5%	1

출처: <https://pages.stat.wisc.edu/~larger/stat302/sol10.pdf>

II. AB test

❖ A/B 테스트(버킷 테스트 또는 분할-실행 테스트)

- 마케팅, 제품 개발, 웹 디자인에서 사용
- 두 가지 버전(일반적으로 웹페이지, 이메일 또는 애플리케이션)을 비교하여 어느 버전이 더 나은 성능을 내는지 확인하는 데 사용되는 방법
- 두 개의 변형 A와 B를 사용하는 종합 대조 실험(controlled experiment)
- 버전 A: 현재 사용되는 버전(control) - 버전 B: 수정 버전(treatment)
- 웹 디자인: 텍스트, 레이아웃, 이미지, 색상과 같은 요소들을 테스트



https://ko.wikipedia.org/wiki/A/B_테스트

AB test

Obama campaign analysis

- 대선자금 마련 캠페인으로 온라인 웹사이트를 통한 뉴스레터 구독자 모집
- 담당자: 댄 시로커(Dan Siroker, 광고 전략 전문가)
- 실험: 상단의 '미디어' 섹션과 클릭 유도 문구 '버튼' 테스트
- 4개 버튼 문구과 6개 미디어(3개의 이미지와 3개의 비디오)
- Google 최적화 도구를 사용하여 다변량 테스트 실행



출처: https://www.optimizely.com/insights/blog/how-obama-raised-60-million-by-running-a-simple-experiment/obama_test_sections/

AB test

❖ 결과

Combinations (24)		Page Sections (2)		Download: XML CSV TSV Print		
Relevance Rating	Variation	Est. conv. rate		Chance to Beat Orig.	Observed Improvement	Conv/Visitors
Button	Original	7.51% ± 0.2%	—	—	—	5851 / 77858
	Learn More	8.91% ± 0.2%	Winner	100%	18.6%	6927 / 77729
	Join Us Now	7.62% ± 0.2%	Inconclusive	73.9%	-1.3%	3913 / 77644
	Sign Up Now	7.34% ± 0.2%	Loser	13.7%	-2.38%	5660 / 77151
Media	Original	8.54% ± 0.2%	—	—	—	4425 / 51794
	Family Image	9.66% ± 0.2%	Winner	100%	13.1%	4996 / 51696
	Change Image	8.61% ± 0.2%	Inconclusive	92.2%	3.85%	4595 / 51790
	Barack's Video	7.76% ± 0.2%	Loser	0.04%	-9.14%	3992 / 51427
	Sam's Video	6.29% ± 0.2%	Loser	0.00%	-26.4%	3261 / 51864
Springfield Video	Springfield Video	5.95% ± 0.2%	Loser	0.00%	-30.3%	3084 / 51811

Combinations (24)		Page Sections (2)		Download: XML CSV TSV Print		
Combination	Status	Est. conv. rate		Chance to Beat Orig.	Observed Improvement	Conv/Visitors
Original	Enabled	8.26% ± 0.5%	—	—	—	1088 / 13167
★ Top high-confidence winners. Run a follow-up experiment »						
Combination 11	Enabled	11.6% ± 0.6%	Winner	100%	40.6%	1504 / 12947
Combination 7	Enabled	10.3% ± 0.6%	Winner	100%	24.0%	1340 / 13073
Combination 3	Enabled	9.80% ± 0.6%	Winner	99.7%	18.7%	1277 / 13025
Combination 10	Enabled	9.23% ± 0.6%	Inconclusive	95.9%	11.7%	1203 / 13031
Combination 8	Enabled	9.03% ± 0.6%	Inconclusive	91.6%	9.28%	1178 / 13046
Combination 9	Enabled	8.77% ± 0.6%	Inconclusive	81.8%	6.10%	1111 / 12672
Combination 6	Enabled	8.64% ± 0.5%	Inconclusive	75.3%	4.58%	1108 / 12822



출처: https://www.optimizely.com/insights/blog/how-obama-raised-60-million-by-running-a-simple-experiment/obama_test_sections/

AB test

❖ 결과

- Combination 11 : Learn more 버튼 + 미디어(가족)
- 가입율: 8.26% → 11.6% (40.6% 증가)
- 페이지 가입자 수: 7,120,000 명 → 1,000만명(2,880,000명 증가)
- 기부금 : 평균 21달러 → 6천만 달러 추가 기부금 달성



출처: https://www.optimizely.com/insights/blog/how-obama-raised-60-million-by-running-a-simple-experiment/obama_test_sections/

AB test

❖ 사례

- G회사는 홈페이지를 통해 고객 등록을 유도하고자 한다. 이번에 홈페이지를 수정하기 위해 A안과 B안을 놓고 1달 동안 AB test를 실시하였다.
- A안과 B안 중에서 클릭률은 어떤 것이 좋은가?
- A안과 B안 중에서 등록률은 어떤 것이 좋은가?

	user_id	timestamp	group	landing_page	converted
0	763854	2017-01-21 03:43:17.188315	control	old_page	0
1	690555	2017-01-18 06:38:13.079449	control	old_page	0
2	861520	2017-01-06 21:13:40.044766	control	old_page	0
3	630778	2017-01-05 16:42:36.995204	control	old_page	0
4	656634	2017-01-04 15:31:21.676130	control	old_page	0

출처: https://github.com/renatofillinich/ab_test_guide_in_python/blob/master/AB%20testing%20with%20Python.ipynb

AB test

Date	A안					B안				
	Pageviews	Clicks	Clicks(%)	Enrollments	Enrollments(%)	Pageviews	Clicks	Clicks(%)	Enrollments	Enrollments(%)
2024-01-01	7,723	687	0.089	134	0.017	7,716	840	0.109	143	0.019
2024-01-02	9,102	779	0.086	147	0.016	9,288	939	0.101	154	0.017
2024-01-03	10,511	909	0.086	167	0.016	10,480	1038	0.099	183	0.017
2024-01-04	9,871	836	0.085	156	0.016	9,867	981	0.099	176	0.018
2024-01-05	10,014	837	0.084	163	0.016	9,793	986	0.101	178	0.018
2024-01-06	9,670	823	0.085	138	0.014	9,500	942	0.099	167	0.018
2024-01-07	9,008	748	0.083	146	0.016	9,088	934	0.103	165	0.018
2024-01-08	7,434	632	0.085	110	0.015	7,664	806	0.105	132	0.017
2024-01-09	8,459	691	0.082	131	0.015	8,434	851	0.101	158	0.019
2024-01-10	10,667	861	0.081	165	0.015	10,496	1014	0.097	191	0.018
2024-01-11	10,660	867	0.081	196	0.018	10,551	1018	0.096	181	0.017
2024-01-12	9,947	838	0.084	162	0.016	9,737	955	0.098	166	0.017
2024-01-13	8,324	665	0.080	127	0.015	8,176	796	0.097	160	0.020
2024-01-14	9,434	673	0.071	220	0.023	9,402	851	0.091	232	0.025
2024-01-15	8,687	691	0.080	176	0.020	8,669	823	0.095	165	0.019
2024-01-16	8,896	708	0.080	161	0.018	8,881	847	0.095	191	0.022
2024-01-17	9,535	759	0.080	233	0.024	9,655	925	0.096	251	0.026
2024-01-18	9,363	736	0.079	154	0.016	9,396	890	0.095	200	0.021
2024-01-19	9,327	739	0.079	196	0.021	9,262	881	0.095	239	0.026
2024-01-20	9,345	734	0.079	167	0.018	9,308	882	0.095	245	0.026
2024-01-21	8,890	706	0.079	174	0.020	8,715	876	0.101	220	0.025
2024-01-22	8,460	681	0.080	156	0.018	8,448	849	0.100	180	0.021
2024-01-23	8,836	693	0.078	206	0.023	8,836	878	0.099	220	0.025
2024-01-24	9,437	788	0.084	134	0.014	9,359	943	0.101	143	0.015
2024-01-25	9,420	781	0.083	147	0.016	9,427	897	0.095	154	0.016
2024-01-26	9,570	805	0.084	167	0.017	9,633	962	0.100	183	0.019
2024-01-27	9,921	830	0.084	156	0.016	9,842	985	0.100	176	0.018
2024-01-28	9,424	781	0.083	163	0.017	9,272	921	0.099	178	0.019
2024-01-29	9,010	756	0.084	138	0.015	8,969	914	0.102	167	0.019
2024-01-30	9,656	825	0.085	146	0.015	9,697	1004	0.104	165	0.017
합계	278,601	22859	0.082	4836	0.017	344,660	34,023	0.099	6,683	0.019

AB test

❖ 교차분석

	Pageviews	Clicks	Non_Clicks	Enrollments	Non_Enrollments
A안	278,601	22,859	255,742	4,836	273,765
B안	344,660	34,023	310,637	6,683	337,977

❖ 비율분석

	Pageviews	Clicks	Clicks(%)	Enrollments	Enrollments(%)
A안	278,601	22,859	0.082	4,836	0.017
B안	344,660	34,023	0.099	6,683	0.019

❖ t-test

	Group	N	Mean	Median	SD	SE
Clicks(%)	A안	30	0.082	0.083	0.0034	0.0006
	B안	30	0.099	0.099	0.0037	0.0007
Enrollments(%)	A안	30	0.017	0.016	0.0027	0.0005
	B안	30	0.020	0.019	0.0033	0.0006

AB test

❖ 교차분석

	Pageviews	Clicks	Non_Clicks	Enrollments	Non_Enrollments
A안	278,601	22,859	255,742	4,836	273,765
B안	344,660	34,023	310,637	6,683	337,977

- Clicks

$$\sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - \hat{E}_{ij})^2}{\hat{E}_{ij}} = 51.93 \sim \chi^2_{0.05, 4} = 9.49$$

$$p-value = 0.000 < \alpha = 0.05$$

- Enrollments

$$\sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - \hat{E}_{ij})^2}{\hat{E}_{ij}} = 35.07 \sim \chi^2_{0.05, 4} = 9.49$$

$$p-value = 0.000 < \alpha = 0.05$$

AB test

❖ 비율분석

	Pageviews	Clicks	Clicks(%)	Enrollments	Enrollments(%)
A안	278,601	22,859	0.082	4,836	0.017
B안	344,660	34,023	0.099	6,683	0.019

$$\bar{p} = \frac{x_1 + x_2}{n_1 + n_2} = \frac{22,859 + 34,023}{278,601 + 344,660} = 0.091$$

$$t_{cal} = \frac{(p_1 - p_2)}{\sqrt{\bar{p}(1 - \bar{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} = \frac{(0.082 - 0.099)}{\sqrt{0.091(1 - 0.091)\left(\frac{1}{278,601} + \frac{1}{344,660}\right)}} = -22.71$$

$$p-value = P(|t| > 22.71) = 0.000$$

AB test

❖ t-test

	Group	N	Mean	Median	SD	SE
Clicks(%)	A안	30	0.082	0.083	0.0034	0.0006
	B안	30	0.099	0.099	0.0037	0.0007
Enrollments(%)	A안	30	0.017	0.016	0.0027	0.0005
	B안	30	0.020	0.019	0.0033	0.0006

$$t_{cal} = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} = -18.5 > -2.0$$

$$p-value = P(|t| > 18.5) = 0.000 < 0.05$$

18_1.AB test

```
✓ 18_1.AB test
  ✓ 1.기본 package 설정
    [1] # 그래프에서 한글 폰트 인식하기
        !sudo apt-get install -y fonts-nanum
        !sudo fc-cache -fv
        !rm ~/.cache/matplotlib -rf

    [1] !pip install pingouin
        # *** 런타임 다시 시작

    ✓ 2초 [1] # 1.기본
        import numpy as np # numpy 패키지 가져오기
        import matplotlib.pyplot as plt # 시각화 패키지 가져오기
        import seaborn as sns # 시각화

        # 2.데이터 가져오기
        import pandas as pd # csv -> dataframe으로 전환

        # 3.통계분석 package
        import pingouin as pg
        from scipy import stats
        import statsmodels.api as sm

    ✓ 0초 [2] # 기본세팅
        # 테마 설정
        sns.set_theme(style = "darkgrid")

        # 한글 인식
        plt.rc('font', family='NanumBarunGothic')
    ✓ 0초 오전 10:37에 완료됨
```

2.교차분석

LGE Internal Use Only

The screenshot shows a Jupyter Notebook interface with the following content:

- Section 2.1:** 2.1 데이터 프레임으로 저장
- Cell 3:** abch_df = pd.read_csv('https://raw.githubusercontent.com/leecho-bigdata/statistics-python/main/18_2.ABTest(CH).csv', encoding="cp949")
abch_df.head()
- Output 3:** A preview of the DataFrame abch_df. The columns are 그룹, Clicks, 관측치1, Enrollments, and 관측치2. The data is as follows:

	그룹	Clicks	관측치1	Enrollments	관측치2
0	1	1	22859	1	4836
1	1	2	255742	2	273765
2	2	1	34023	1	6683
3	2	2	310637	2	337977

- Cell 4:** abch_df.info()
- Output 4:** DataFrame information:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
 --- 
 0   그룹        4 non-null    int64  
 1   Clicks      4 non-null    int64  
 2   관측치1     4 non-null    int64  
 3   Enrollments 4 non-null    int64  
 4   관측치2     4 non-null    int64  
dtypes: int64(5)
memory usage: 288.0 bytes
```
- Cell 5:** abch_df['그룹'].replace({1:'A안', 2:'B안'}, inplace=True)
abch_df['Clicks'].replace({1:'Click', 2:'Non_Click'}, inplace=True)
abch_df['Enrollments'].replace({1:'Enrollments', 2:'Non_Enrollments'}, inplace=True)
abch_df['그룹'] = abch_df['그룹'].astype('category')
abch_df['Clicks'] = abch_df['Clicks'].cat.set_categories(['Click', 'Non_Click'])
- Feedback:** ✓ 0초 오전 10:37에 완료됨

2.교차분석

```

0초      uctypes.malloc()
memory usage: 288.0 bytes

[5]: abch_df['그룹'].replace({1:'A안', 2:'B안'}, inplace=True)
abch_df['Clicks'].replace({1:'Click', 2:'Non_Click'}, inplace=True)
abch_df['Enrollments'].replace({1:'Enrollments', 2:'Non_Enrollments'}, inplace=True)
abch_df['그룹'] = abch_df['그룹'].astype('category')
abch_df['Clicks'] = abch_df['Clicks'].astype('category')
abch_df['Enrollments'] = abch_df['Enrollments'].astype('category')
abch_df

```

	그룹	Clicks	관측치1	Enrollments	관측치2
0	A안	Click	22859	Enrollments	4836
1	A안	Non_Click	255742	Non_Enrollments	273765
2	B안	Click	34023	Enrollments	6683
3	B안	Non_Click	310637	Non_Enrollments	337977

Next steps: [View recommended plots](#)

▼ 2.2 Clicks(%)

```

0초
[6]: tab = pd.crosstab(abch_df['그룹'],
                      abch_df['Clicks'],
                      values = abch_df['관측치1'],
                      margins = True,
                      aggfunc = 'sum')
tab

```

	Clicks	Click	Non_Click	All
그룹				
A안	22859	255742	278601	
B안	34023	310637	344660	
All	56882	566379	623261	

✓ 0초 오전 10:37에 완료됨

2.교차분석

LGE Internal Use Only

```
▼ 2.2 Clicks(%)

[6]: tab = pd.crosstab(abch_df['그룹'],
                      abch_df['Clicks'],
                      values = abch_df['관측치1'],
                      margins = True,
                      aggfunc = 'sum')
tab

Clicks Click Non_Click All
그룹
A안 22859 255742 278601
B안 34023 310637 344660
All 56882 566379 623261

Next steps:  View recommended plots

[7]: # Pearson
result = sm.stats.Table(tab)
rslt = result.test_nominal_association()
print(rslt)

df      4
pvalue 0.0
statistic 515.9348810835081

▼ 2.3 Enrollments(%)

[8]: tab = pd.crosstab(abch_df['그룹'],
                      abch_df['Enrollments'],
                      values = abch_df['관측치2'],
                      margins = True,
                      aggfunc = 'sum')
✓ 0초 오전 10:37에 완료됨
```

2.교차분석

▼ 2.3 Enrollments(%)

```
[8]: tab = pd.crosstab(abch_df['그룹'],
                      abch_df['Enrollments'],
                      values = abch_df['관측치2'],
                      margins = True,
                      aggfunc = 'sum')
tab
```

	Enrollments	Non_Enrollments	All
그룹			
A안	4836	273765	278601
B안	6683	337977	344660
All	11519	611742	623261

Next steps: View recommended plots

```
[9]: # Pearson
result = sm.stats.Table(tab)
rslt = result.test_nominal_association()
print(rslt)
```

df	4
pvalue	4.501337250717441e-07
statistic	35.06657937860307

▼ 3.비율검정

▼ 3.1 Clicks(%)

```
[10]: from statsmodels.stats.proportion import proportions_ztest
```

SC... ✓ 0초 오전 10:37에 완료됨

3.비율검정

The screenshot shows a Jupyter Notebook interface with the following structure:

- Section 3. 비율검정**
 - Section 3.1 Clicks(%)**
 - Cell [10]:

```
0초 [10] from statsmodels.stats.proportion import proportions_ztest

count = np.array([22859, 34023])      # x1, x2
nobs = np.array([278601, 344660])    # n1, n2

z, p = proportions_ztest(count = count,
                         nobs = nobs,
                         value = 0)
print('z : {}, p : {}'.format(z, p))

z : -22.714199987750114, p : 3.2431643422618867e-114
```
 - Section 4.2 Enrollments(%)**
 - Cell [11]:

```
0초 [11] from statsmodels.stats.proportion import proportions_ztest

count = np.array([4836, 6683])      # x1, x2
nobs = np.array([278601, 344660])    # n1, n2

z, p = proportions_ztest(count = count,
                         nobs = nobs,
                         value = 0)
print('z : {}, p : {}'.format(z, p))

z : -5.921704094144109, p : 3.1862255504964427e-09
```
 - Cell [12]:

```
0초 [12] # z2 = ch2
np.sqrt(rslt.statistic)
```

✓ 0초 오전 10:37에 완료됨



4.t-test

▼ 4.t-test

▼ 4.1 데이터 프레임으로 저장

```
[0초] [13] ab_df = pd.read_csv('https://raw.githubusercontent.com/leecho-bigdata/statistics-python/main/18_1.ABTest.csv', encoding="cp949")
ab_df.head()
```

그룹	Date	Pageviews	Clicks	Clicks(%)	Enrollments	Enrollments(%)
0	2024-01-01	7,723	687	0.089	134	0.017
1	2024-01-02	9,102	779	0.086	147	0.016
2	2024-01-03	10,511	909	0.086	167	0.016
3	2024-01-04	9,871	836	0.085	156	0.016
4	2024-01-05	10,014	837	0.084	163	0.016

Next steps: View recommended plots

```
[0초] [14] ab_df['그룹'].replace({1:'A안', 2:'B안'}, inplace=True)
ab_df['그룹'] = ab_df['그룹'].astype('category')

ab_df.head()
```

그룹	Date	Pageviews	Clicks	Clicks(%)	Enrollments	Enrollments(%)
0	A안	2024-01-01	7,723	687	0.089	134
1	A안	2024-01-02	9,102	779	0.086	147
2	A안	2024-01-03	10,511	909	0.086	167
3	A안	2024-01-04	9,871	836	0.085	156
4	A안	2024-01-05	10,014	837	0.084	163

Next steps: View recommended plots

✓ 0초 오전 10:37에 완료됨

4.t-test

```

✓ [14] 1 A안 2024-01-02 9,102 779 0.086 147 0.016
       2 A안 2024-01-03 10,511 909 0.086 167 0.016
       3 A안 2024-01-04 9,871 836 0.085 156 0.016
       4 A안 2024-01-05 10,014 837 0.084 163 0.016

Next steps: View recommended plots

✓ [15] # 분석변수가 여러개 일 때
num_feature = [ 'Clicks(%)', 'Enrollments(%)' ]
for num in num_feature:
    print("----", num, "----")
    results = ab_df.groupby('그룹')[num].describe().round(4)
    print(results, "#n")

---- Clicks(%) ----
   count      mean      std     min    25%    50%    75%      max
그룹
A안  30.0  0.0821  0.0034  0.071  0.080  0.083  0.084  0.089
B안  30.0  0.0989  0.0037  0.091  0.096  0.099  0.101  0.109

---- Enrollments(%) ----
   count      mean      std     min    25%    50%    75%      max
그룹
A안  30.0  0.0172  0.0027  0.014  0.0152  0.016  0.018  0.024
B안  30.0  0.0197  0.0033  0.015  0.0173  0.019  0.021  0.026

```

▼ 4.2 Clicks(%)

```

✓ [16] x = ab_df[ 'Clicks(%)'][ab_df[ '그룹' ] == 'A안']
y = ab_df[ 'Clicks(%)'][ab_df[ '그룹' ] == 'B안']

✓ [17] # paired = True : paired sample t-test
# correction = False : 등분산일때
pg.ttest(x, y,
          paired = False,
          alternative = "two-sided").

```

✓ 0초 오전 10:37에 완료됨

4.t-test

```
▼ 4.2 Clicks(%)

[16] x = ab_df['Clicks(%)'][ab_df['그룹'] == 'A안']
y = ab_df['Clicks(%)'][ab_df['그룹'] == 'B안']

[17] # paired = True : paired sample t-test
# correction = False : 등분산일때
pg.ttest(x, y,
          paired = False,
          alternative = "two-sided",
          correction = False).round(3)

      T  dof  alternative  p-val    CI95%  cohen-d    BF10  power
T-test -18.32    58   two-sided   0.0 [-0.02, -0.01]    4.73  2.229e+22    1.0

[18] # 그래프
sns.catplot(x = "그룹",
             y = "Clicks(%)",
             kind = "point",
             data = ab_df)
plt.show()


```

4.t-test

```
[19] # 등분산이면 지금까지 분석한 것이 문제 없음
pg.homoscedasticity(ab_df,
                     dv = "Clicks(%)",
                     group = "그룹")

    ┌─────────┐ pval equal_var ┌─────────┐
levene 0.076533 0.783035 True

[20] pg.normality(ab_df,
                  dv = 'Clicks(%)',
                  group = '그룹')

    ┌─────────┐ pval normal ┌─────────┐
그룹
A안 0.917436 0.023024 False
B안 0.960252 0.314403 True

[21] # 한글 폰트 인식
sns.catplot(data = ab_df,
             y = "Clicks(%)",
             hue = "그룹",
             kind = "box")
plt.show()


```

4.t-test

```
✓ 4.3 Enrollments(%)
```

```
[22]: x = ab_df['Enrollments(%)'][ab_df['그룹'] == 'A안']
y = ab_df['Enrollments(%)'][ab_df['그룹'] == 'B안']

pg.ttest(x, y,
          paired = False,
          alternative = "two-sided",
          correction = False).round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-3.275	58	two-sided	0.002	[-0.0, -0.0]	0.846	19.065	0.896

```
[23]: # 그래프
sns.catplot(x = "그룹",
             y = "Enrollments(%)",
             kind = "point",
             data = ab_df)
plt.show()
```

The figure is a scatter plot with a light gray background. The y-axis is labeled 'Enrollments(%)' and ranges from 0.018 to 0.021 with major grid lines every 0.001. The x-axis has two categories: '0초' and '오전 10:37에 완료됨'. The point for '0초' is at approximately (0.0182, 0.0182) and the point for '오전 10:37에 완료됨' is at approximately (0.0198, 0.0200). Each point includes a vertical blue error bar. A legend in the bottom left corner shows a green checkmark next to '0초' and a gray square next to '오전 10:37에 완료됨'. In the bottom right corner, there are three icons: a green dot, a black square with a white 'X', and a black square with a white circle.

4.t-test

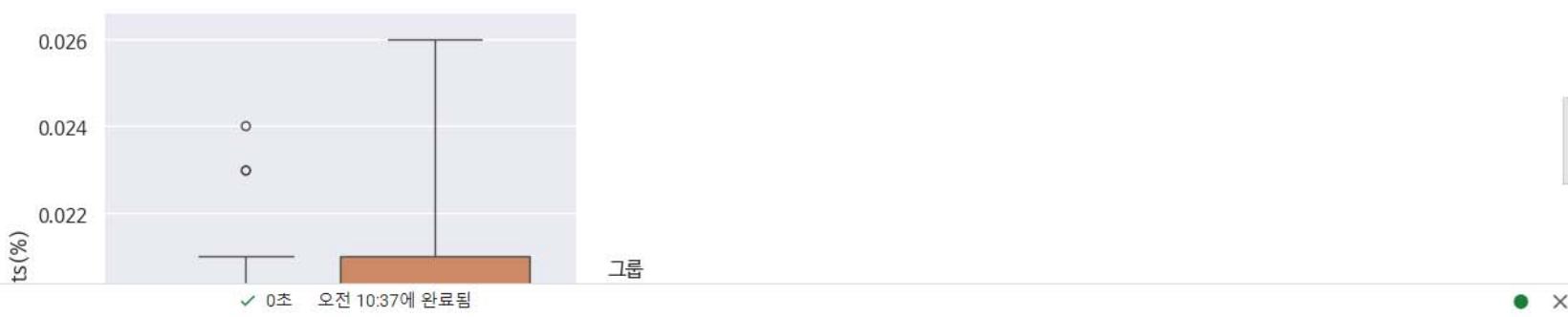
```
[24] # 등분산이면 지금까지 분석한 것이 문제 없음
pg.homoscedasticity(ab_df,
                     dv = "Enrollments(%)",
                     group = "그룹")

      ┌─────────┐ pval equal_var ──────────┐
      ┌─────────┐ levene 0.817621 0.369617 ──────────┐
      ┌─────────┐ ┌─────────┐ ┌─────────┐ ──────────┐
      ┌─────────┐ 그룹 ┌─────────┐ pval  normal ──────────┐
      ┌─────────┐ ┌─────────┐ ┌─────────┐ ┌─────────┐ ──────────┐
      ┌─────────┐ A안 ┌─────────┐ 0.844781 0.000483 ──────────┐
      ┌─────────┐ ┌─────────┐ ┌─────────┐ ┌─────────┐ ──────────┐
      ┌─────────┐ B안 ┌─────────┐ 0.849501 0.000605 ──────────┐

[25] pg.normality(ab_df,
                  dv = 'Enrollments(%)',
                  group = '그룹')

      ┌─────────┐ pval normal ──────────┐
      ┌─────────┐ 그룹 ┌─────────┐ ┌─────────┐ ──────────┐
      ┌─────────┐ ┌─────────┐ ┌─────────┐ ┌─────────┐ ──────────┐
      ┌─────────┐ A안 ┌─────────┐ 0.844781 0.000483 ──────────┐
      ┌─────────┐ ┌─────────┐ ┌─────────┐ ┌─────────┐ ──────────┐
      ┌─────────┐ B안 ┌─────────┐ 0.849501 0.000605 ──────────┐

[26] # 한글 폰트 인식
sns.catplot(data = ab_df,
             y = "Enrollments(%)",
             hue = "그룹",
             kind = "box")
plt.show()


```

4.t-test



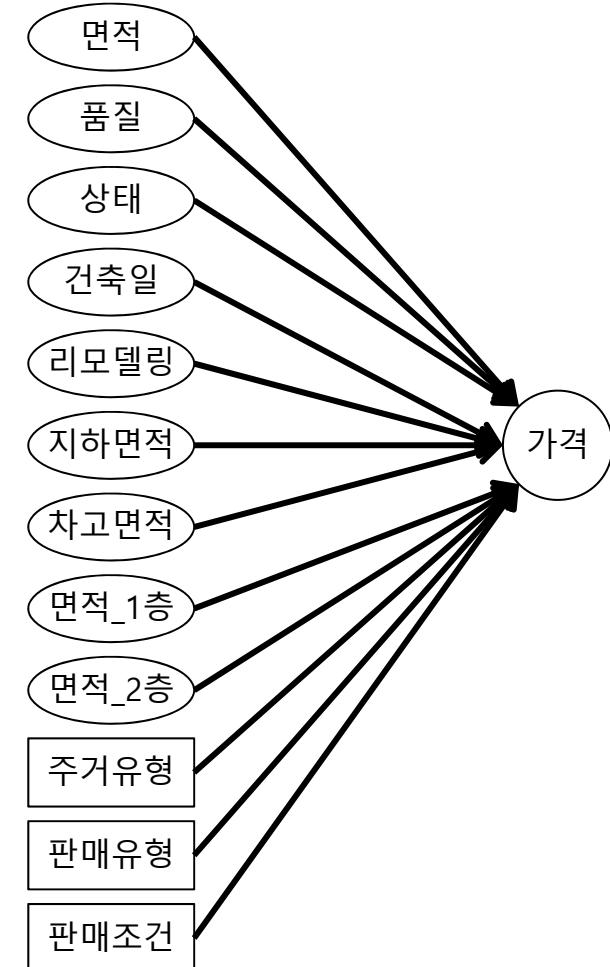
III. Regression

Correlation

Regression(예측)

❖ 문제의 정의

- 가격 - 부동산의 판매 가격(달러)
- 면적: 부지 크기(제곱피트)
- 품질: 전반적인 재료 및 마감 품질(10점)
- 상태: 전반적인 상태 등급 (10점)
- 건축일: 원래 건설 날짜
- 리모델링: 리모델링 날짜
- 지하면적: 지하 면적의 총 평방피트
- 차고면적: 평방 피트 단위의 차고 크기
- 면적_1층: 1층 평방 피트
- 면적_2층: 2층 평방 피트
- 주거유형BldgType: 주거 유형
- 판매유형SaleType: 판매 유형
- 판매조건SaleCondition: 판매 조건



Regression(예측)

❖ 범주형 데이터

- One-hot encoding = dummy variable
- 주거유형

level	label	듀플렉스	주거_기타
1	단독주택	0	0
2	듀플렉스	1	0
3	기타	0	1

- 판매조건

Level	label	판매_신규
1	보증증서	0
2	신규건물	1

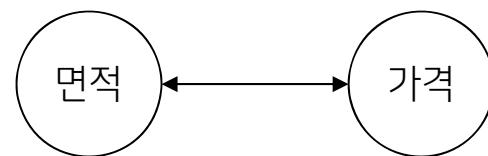
- 판매유형

level	label	조건_압류및공매도
1	정상판매	0
2	압류 및 공매도	1

Correlation

❖ 문제의 정의

- 부동산의 판매 가격과 면적 간의 관계를 살펴보고자 한다. 가격과 면적은 관계가 있는가?
- 또한 다른 변수들(면적 ~ 면적_2층까지의 변수)와는 관계가 있는가?
- 11_1.COR.csv



Correlation(상관분석)

- 가설검정

$$H_0: \rho = 0$$

$$H_1: \rho \neq 0$$

가격	연면적
150750	7388
131500	4435
160000	8800
187500	13031
153900	7892
129900	4224
165500	9600
173000	10852
167000	9937
184000	12416
165000	9500
149000	7200
180500	11851
177000	11512
155000	7931
167000	10004
156500	8400
163000	9120
170000	10192
160000	8658

Correlation

❖ 두 개의 연속변수 사이의 관계성

- 두 변수간 상호의존관계가 있을 때 이 관련성을 통계적으로 분석
- 한 쌍의 변수가 선형 적으로 관련된 정도
- 관련된 두 변수가 있을 때 하나의 변수에 대한 정보를 가지고 다른 변수를 예측하거나 설명할 때
- 두 변수 사이에 강한 관련성이 있을 경우에는 한 변수에 대한 정보를 가지고 다른 변수를 예측 할 수 있음

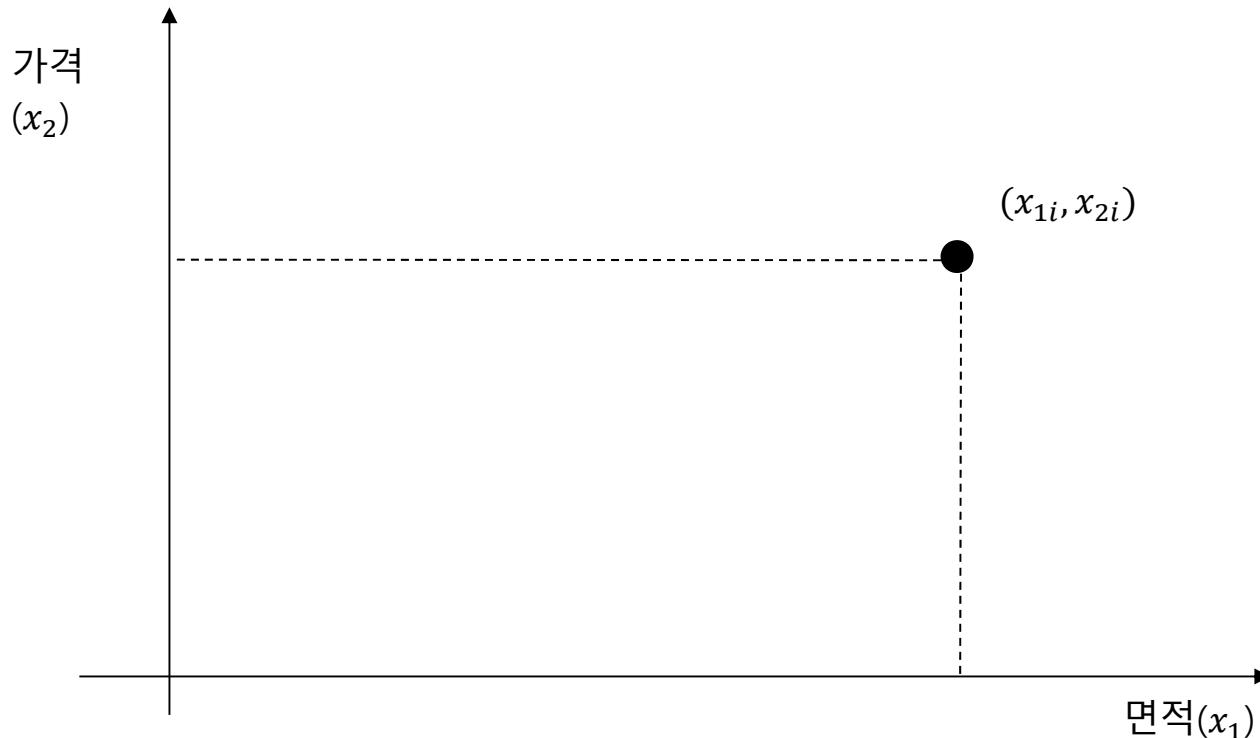
❖ 자료

- $(x_1, x_2), \dots, (x_i, x_j)$
- 부동산 면적과 가격과의 관계
- 야구선수의 훌런수와 연봉액수
- 광고비와 매출액
- 부모의 키 와 자녀 간의 관계

Correlation

- ❖ 산점도(Scatter plot)

- 두 변수 x_1, x_2 의 관측치 (x_i, x_j) 를 좌표평면상에 점으로 나타낸 그림



❖ 공분산(Covariance)

- 두 변수간의 공통분산
- 모집단

$$\sigma_{xy} = \frac{1}{N} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)$$

- 표본집단

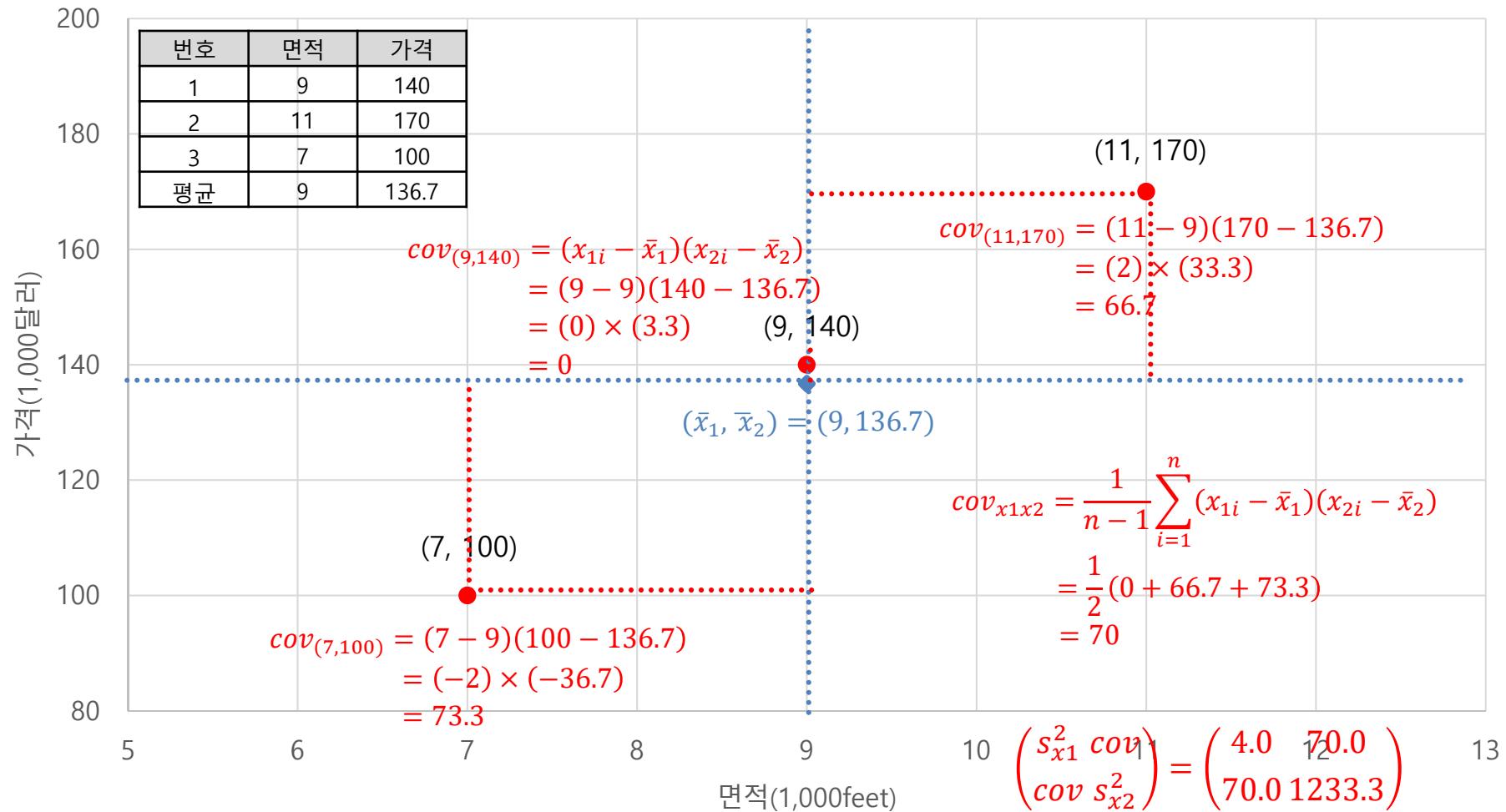
$$s_{xy} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

- 공분산은 척도단위에 따라 민감하게 반응함 → 표준화 필요

Correlation

LGE Internal Use Only

공분산(Covariance)



Correlation

❖ 상관계수(Correlation Coefficient)

- 두 변수의 관계를 하나의 수치로 나타낸 척도

- 모상관계수 : $\rho = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$

- 표본상관계수 : $r = \frac{s_{xy}}{s_x s \sigma_y}$

$$r = \frac{cov(x, y)}{\sqrt{var(x)} \sqrt{var(y)}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

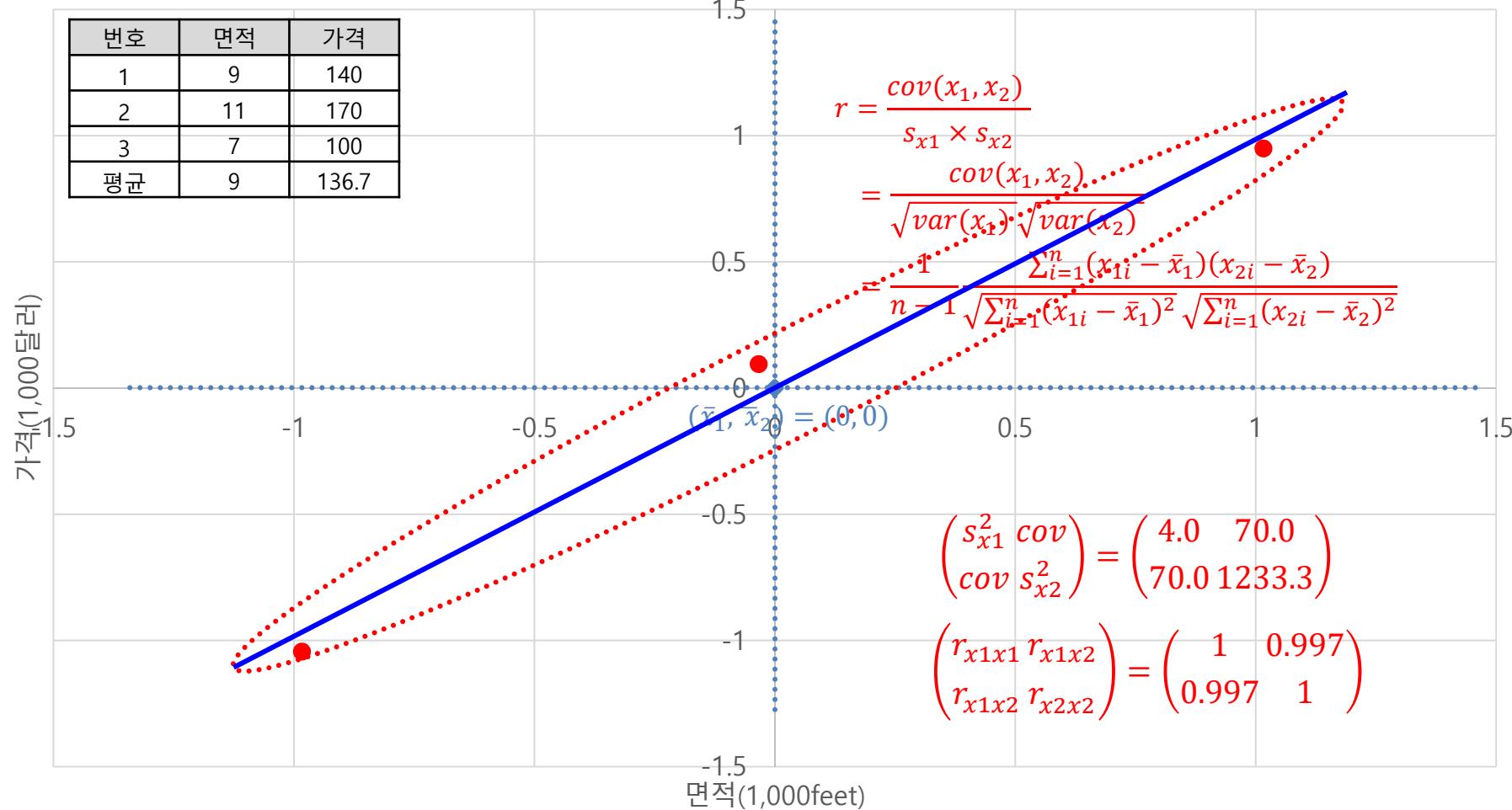
- 크기

$$\rho(r) = \begin{cases} \text{강도: } 0 \sim 1 \\ \text{방향: } - \text{ or } + \end{cases} \quad r = -1 \sim 0 \sim +1$$

Correlation

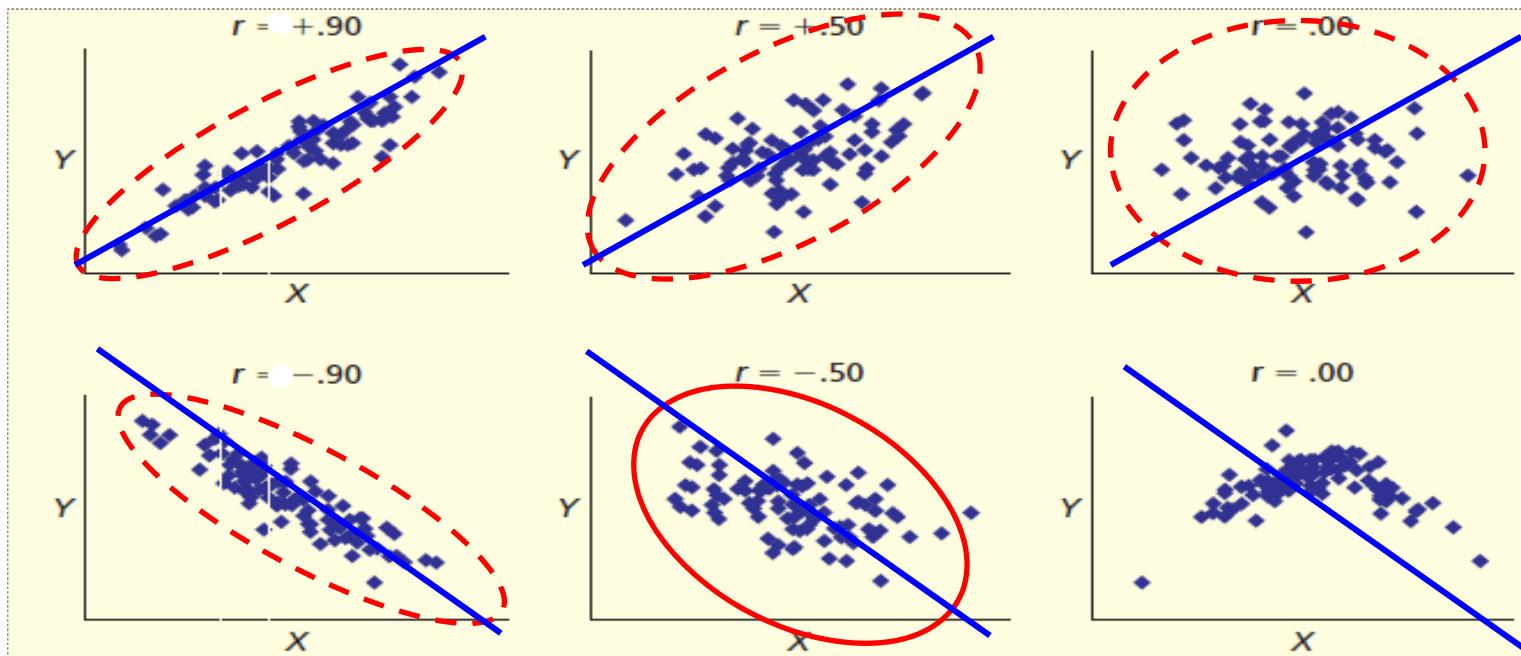
LGE Internal Use Only

상관계수(Correlation)



Correlation

- ❖ 두 개의 연속변수 사이의 관계성(선형성)

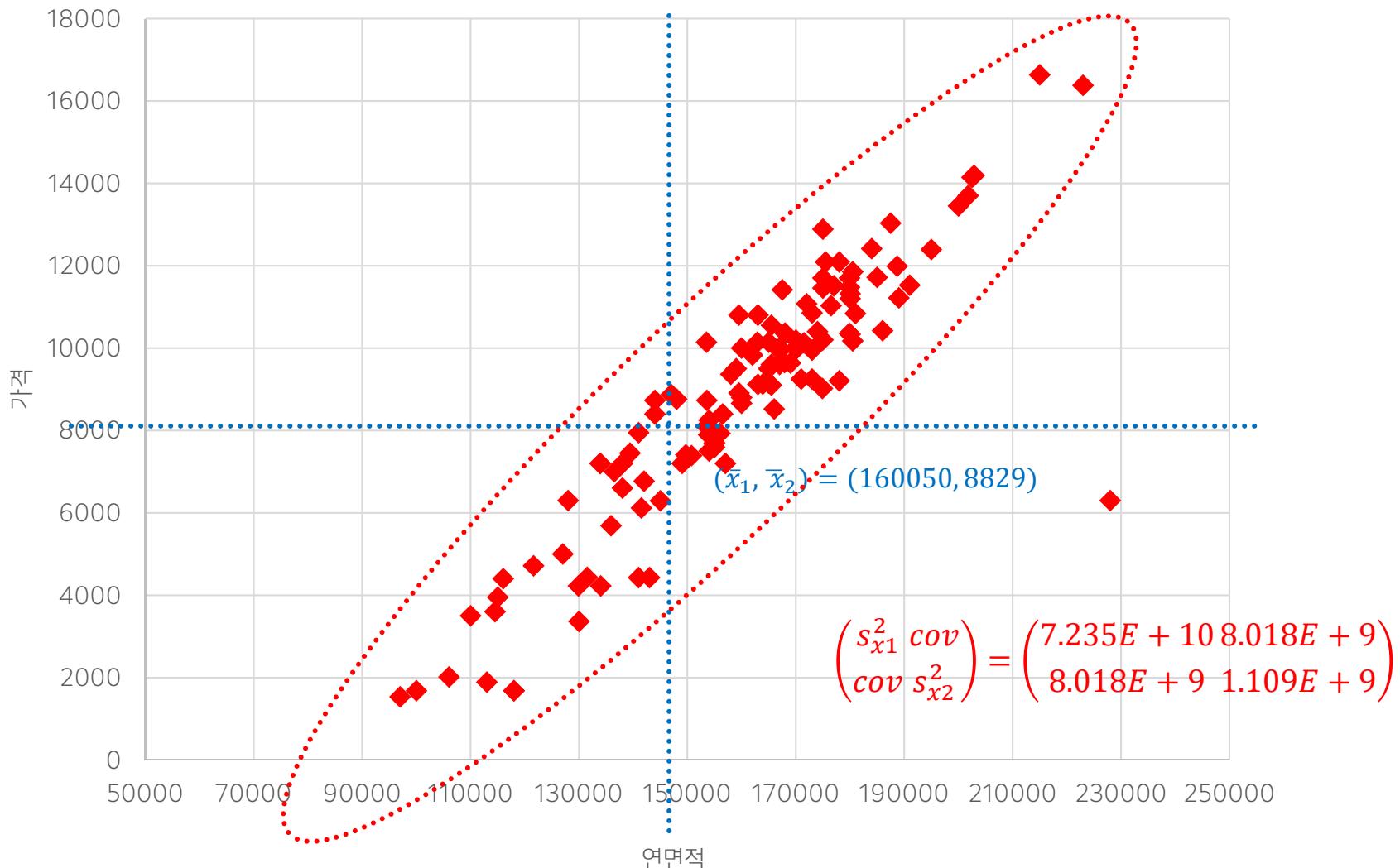


출처: Applied Statistics in Business and Economics, David Doane, Lori Seward, McGraw Hill

Correlation

LGE Internal Use Only

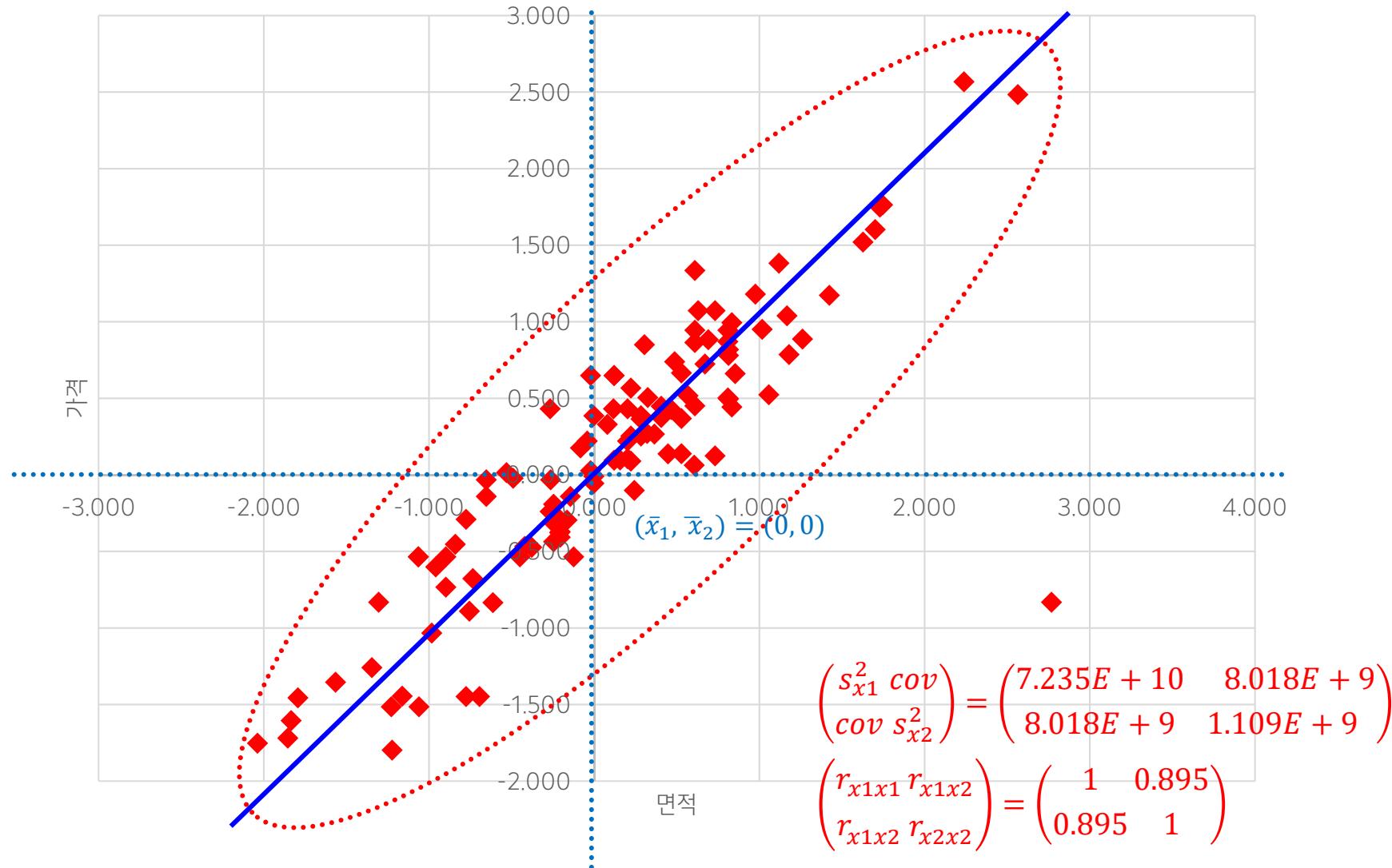
공분산(Covariance)



Correlation

LGE Internal Use Only

상관계수(Correlation)



Correlation

❖ 상관계수

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} = 0.859$$

❖ 검정통계량 (test statistics)

$$t = \frac{r}{\sqrt{\frac{(1 - r^2)}{n - 2}}} = \frac{r\sqrt{(n - 2)}}{\sqrt{(1 - r^2)}} = \frac{0.895\sqrt{(121 - 2)}}{\sqrt{(1 - 0.895^2)}} = 21.888 \quad * t_{cal} = \frac{r}{\sqrt{\frac{(1 - r^2)}{n - 2}}} \sim t_{n-2}$$

❖ 유의확률(*p*-value) 계산

$$p-value = P(|t| > 21.888) = 0.000 < \alpha = 0.05$$

11_1.Correlation

The screenshot shows a Jupyter Notebook interface with the following structure:

- Section 11_1.Correlation**
 - <https://pingouin-stats.org/build/html/generated/pingouin.corr.html#pingouin.corr>
- Section 1.기본 package 설정**
 - In-cell code:

```
[ ] # 그래프에서 한글 폰트 인식하기
!sudo apt-get install -y fonts-nanum
!sudo fc-cache -fv
!rm ~/.cache/matplotlib -rf
```

```
[ ] !pip install pingouin
# *** 런타임 다시 시작
```
 - In-cell code (cell 1):

```
2초 [1] # 1.기본
import numpy as np # numpy 패키지 가져오기
import matplotlib.pyplot as plt # 시각화 패키지 가져오기
import seaborn as sns # 시각화
```

```
# 2.데이터 가져오기
import pandas as pd # csv -> dataframe으로 전환
```

```
# 3.통계분석 package
import pingouin as pg
from scipy import stats
import statsmodels.api as sm
```
 - In-cell code (cell 2):

```
0초 [2] # 기본세팅
# 테마 설정
sns.set_theme(style = "darkgrid")
```

2.데이터 불러오기

LGE Internal Use Only

The screenshot shows a Jupyter Notebook interface with the following structure:

- Section 2.1:** 2.1 데이터 프레임으로 저장
 - 원본데이터(csv)를 dataframe 형태로 가져오기(pandas)
- Cell 3:** [3]

```
cr_df = pd.read_csv('https://raw.githubusercontent.com/leecho-bigdata/statistics-python/main/11_1.CR.csv', encoding="cp949")  
cr_df.head()
```

Output: A Pandas DataFrame with 14 columns and 5 rows. The columns are labeled: id, 가격, 연면적, 품질, 상태, 건축년도, 리모델링년도, 지하면적, 차고면적, 면적_1층, 면적_2층, 주거유형, 판매유형, 판매조건.

	id	가격	연면적	품질	상태	건축년도	리모델링년도	지하면적	차고면적	면적_1층	면적_2층	주거유형	판매유형	판매조건
0	1.0	150750.0	7388.0	5.0	6.0	1959.0	2002.0	1063.0	624.0	1327.0	0.0	1.0	1.0	1.0
1	2.0	131500.0	4435.0	6.0	5.0	2003.0	2003.0	848.0	420.0	848.0	0.0	3.0	2.0	2.0
2	3.0	160000.0	8800.0	6.0	6.0	1964.0	1964.0	1251.0	461.0	1251.0	0.0	1.0	1.0	1.0
3	4.0	187500.0	13031.0	6.0	5.0	1995.0	1996.0	691.0	409.0	691.0	807.0	1.0	1.0	1.0
4	5.0	153900.0	7892.0	6.0	5.0	1993.0	1993.0	1199.0	530.0	1199.0	0.0	3.0	2.0	2.0

Next steps: [View recommended plots](#)
- Cell 4:** [4]

```
cr_df.shape
```

Output: (1180, 14)
- Cell 5:** [5]

```
cr_df.info()
```

Output: Pandas DataFrame information. Total 1180 entries, 0 to 1179. Data columns (total 14 columns): # Column Non-Null Count Dtype --- 0 id 121 non-null float64 1 ... 121 non-null float64

3. 기술통계

```

▼ 3.기술통계
✓ [7] # 그룹별 기술통계
cr_df.describe().round(3).T

    count      mean       std      min     25%     50%     75%      max
  id      121.0   61.000   35.074     1.0    31.0    61.0    91.0   121.0
  가격    121.0 160050.653 24553.521 97000.0 144000.0 163000.0 175000.0 228000.0
  연면적   121.0  8829.157  3040.173   1533.0   7200.0   9247.0  10800.0  16635.0
  품질     121.0    5.901    0.723     4.0      5.0      6.0      6.0     8.0
  상태     121.0    5.975    1.084     3.0      5.0      6.0      7.0     8.0
  건축년도  121.0 1963.603   26.390   1890.0   1957.0   1968.0   1978.0   2009.0
  리모델링년도 121.0 1982.570   18.394   1950.0   1968.0   1988.0   2000.0   2009.0
  지하면적   121.0  967.207  315.738     0.0    731.0   912.0   1196.0   1844.0
  차고면적   121.0  445.248  159.853     0.0    336.0   453.0   530.0   923.0
  면적_1층   121.0 1119.347  317.726   483.0   848.0  1116.0  1350.0  2020.0
  면적_2층   121.0  337.364  373.564     0.0      0.0      0.0    708.0  1101.0
  주거유형   121.0    1.372    0.685     1.0      1.0      1.0      2.0     3.0
  판매유형   121.0    1.182    0.387     1.0      1.0      1.0      1.0     2.0
  판매조건   121.0    1.322    0.469     1.0      1.0      1.0      2.0     2.0

▼ 4.Correlation
✓ [8] columns = [['연면적', '품질', '상태', '건축년도', '리모델링년도',
  '지하면적', '차고면적', '면적_1층', '면적_2층'], ['가격']]
  pg.pairwise_corr(cr_df, columns = columns)

  ✓ 34초 오후 9:17에 완료됨

```

4.Correlation

```

▼ 4.Correlation

✓ [8] columns = [['연면적', '품질', '상태', '건축년도', '리모델링년도',
    '지하면적', '차고면적', '면적_1층', '면적_2층'], ['가격']]
    pg.pairwise_corr(cr_df, columns = columns)

      X   Y   method alternative   n      r      CI95%      p-unc      BF10      power
  0  연면적  가격  pearson  two-sided  121  0.895144  [0.85, 0.93]  1.430389e-43  1.325e+40  1.000000
  1  품질   가격  pearson  two-sided  121  0.397698  [0.24, 0.54]  6.271452e-06  2706.922  0.995720
  2  상태   가격  pearson  two-sided  121  -0.212236 [-0.38, -0.04]  1.943663e-02   1.688  0.651579
  3  건축년도  가격  pearson  two-sided  121  0.255337  [0.08, 0.41]  4.705054e-03   5.884  0.812430
  4  리모델링년도  가격  pearson  two-sided  121  0.163480  [-0.02, 0.33]  7.318311e-02   0.555  0.435829
  5  지하면적  가격  pearson  two-sided  121  0.350390  [0.18, 0.5]  8.150373e-05  243.037  0.978787
  6  차고면적  가격  pearson  two-sided  121  0.325337  [0.16, 0.48]  2.714423e-04  79.393  0.957370
  7  면적_1층  가격  pearson  two-sided  121  0.431133  [0.27, 0.57]  7.947524e-07  1.92e+04  0.998927
  8  면적_2층  가격  pearson  two-sided  121  0.011150  [-0.17, 0.19]  9.033932e-01   0.115  0.051562

✓ [9] variables = ['가격', '연면적', '품질', '상태', '건축년도', '리모델링년도',
    '지하면적', '차고면적', '면적_1층', '면적_2층']
    cr_df[variables].rcorr()

      가격  연면적  품질  상태  건축년도  리모델링년도  지하면적  차고면적  면적_1층  면적_2층
  가격   -      ***    ***     *      **          ***        ***        ***        ***
  연면적  0.895   -      ***          *          ***        ***        ***        ***
  품질   0.398  0.346   -          -          ***        *          *          ***        ***
  상태   -0.212 -0.14   -0.088     -      ***          *          *          ***        ***
  건축년도  0.255  0.2  0.121  -0.444     -      **          *          *          **        **

  ✓ 34초  오후 9:17에 완료됨

```

4.Correlation

```

8   면적_2층  가격  pearson      two-sided 121  0.011150  [-0.17, 0.19]  9.033932e-01  0.115  0.051562

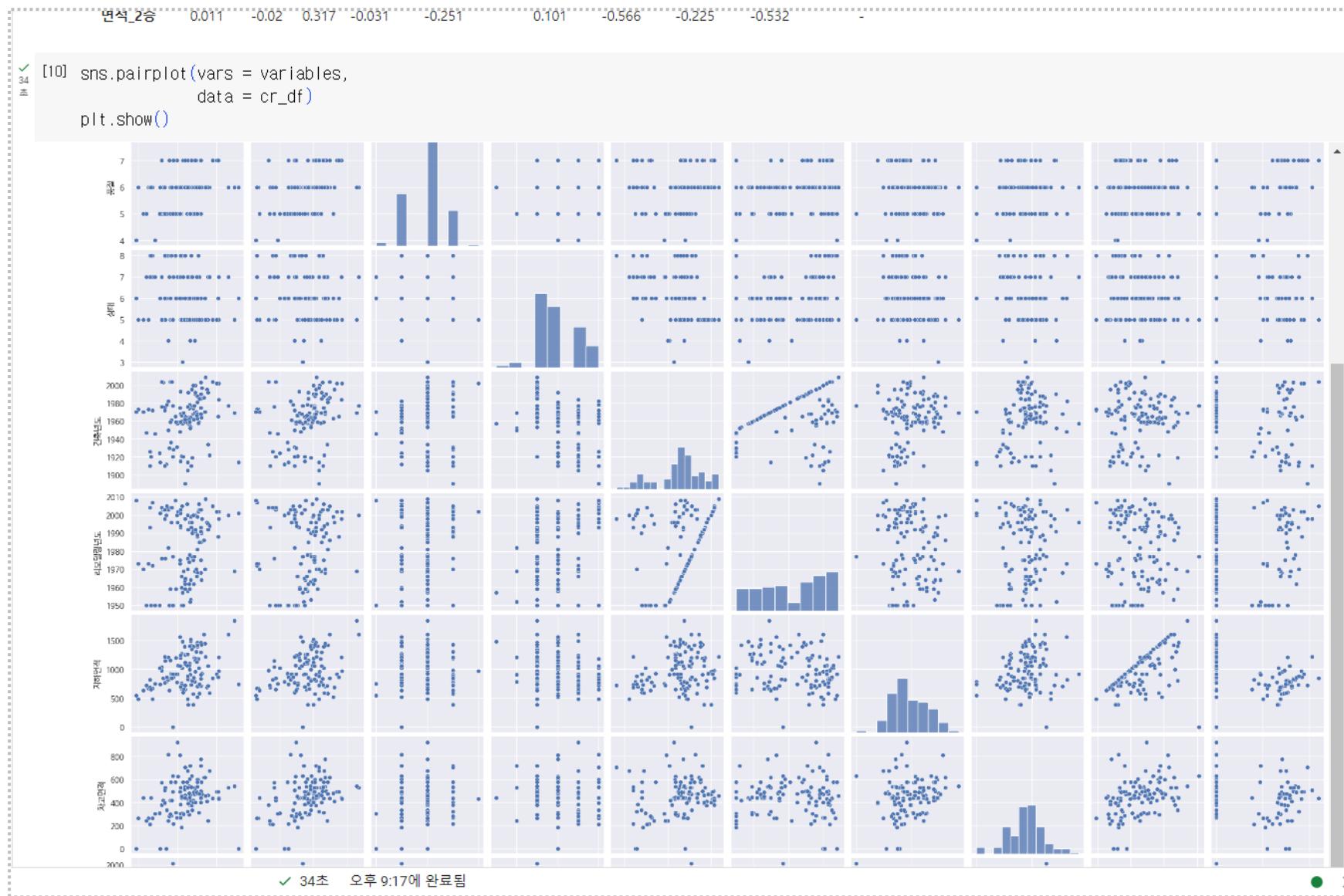
[9] variables = ['가격', '연면적', '품질', '상태', '건축년도', '리모델링년도',
    '지하면적', '차고면적', '면적_1층', '면적_2층']

cr_df[variables].rcorr()

  가격 연면적 품질 상태 건축년도 리모델링년도 지하면적 차고면적 면적_1층 면적_2층
  가격 - *** *** * ** *** *** *** ***
  연면적 0.895 - *** * *** *** *** ***
  품질 0.398 0.346 - - - - - -
  상태 -0.212 -0.14 -0.088 - - *** * * *
  건축년도 0.255 0.2 0.121 -0.444 - - ** * * * *
  리모델링년도 0.163 0.121 0.155 0.211 0.271 - - - -
  지하면적 0.35 0.352 -0.08 -0.212 0.188 -0.129 - - * *** ***
  차고면적 0.325 0.41 0.082 -0.084 0.221 -0.032 0.223 - - *** *
  면적_1층 0.431 0.508 -0.028 -0.145 0.008 -0.082 0.605 0.415 - - ***
  면적_2층 0.011 -0.02 0.317 -0.031 -0.251 0.101 -0.566 -0.225 -0.532 - -
```

[10] sns.pairplot(vars = variables,
 data = cr_df)
plt.show()

4.Correlation



Correlation

LGE Internal Use Only

- ❖ 가격과 연면적에는 양의 상관관계가 있는 것으로 나타났다 ($r = .857$, $p < .000$).

	가격	연면적	품질	상태	건축년도	리모델링년도	지하면적	차고면적	면적_1층
가격									
연면적	0.90***								
품질	0.40***	0.35***							
상태	-0.21*	-0.14	-0.09						
건축년도	0.26**	0.20*	0.12	-0.44***					
리모델링년도	0.16	0.12	0.16	0.21*	0.27**				
지하면적	0.35***	0.35***	-0.08	-0.21*	0.19*	-0.13			
차고면적	0.33***	0.41***	0.08	-0.08	0.22*	-0.03	0.22*		
면적_1층	0.43***	0.51***	-0.03	-0.15	0.01	-0.08	0.61***	0.42***	
면적_2층	0.01	-0.02	0.32***	-0.03	-0.25**	0.10	-0.57***	-0.23*	-0.53***

Note. * $p < .05$, ** $p < .01$, *** $p < .001$

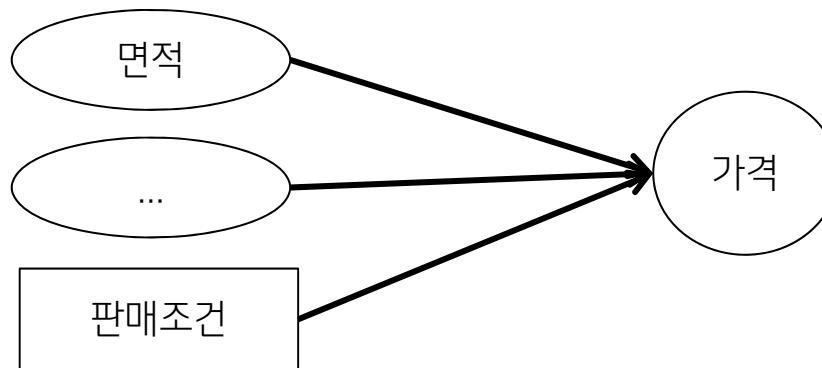
Regression[예측]

Linear Regression(예측)

LGE Internal Use Only

❖ 문제의 정의

- 부동산의 판매 가격을 예측하고자 한다. 부동산 가격과 관계가 있는 변수들은 무엇인가?
- 부동산 판매 가격을 예측할 수 있는 예측모델을 만들어 보자.
- 12_1.MR(예측).csv



Regression(회귀분석)

- 가설검정

$$H_0: \beta_1 = \beta_2 = \beta_3 = \dots = \beta_n = 0$$

$$H_1: \text{not } H_0$$

Regression(예측)

LGE Internal Use Only

❖ 회귀분석(Regression)

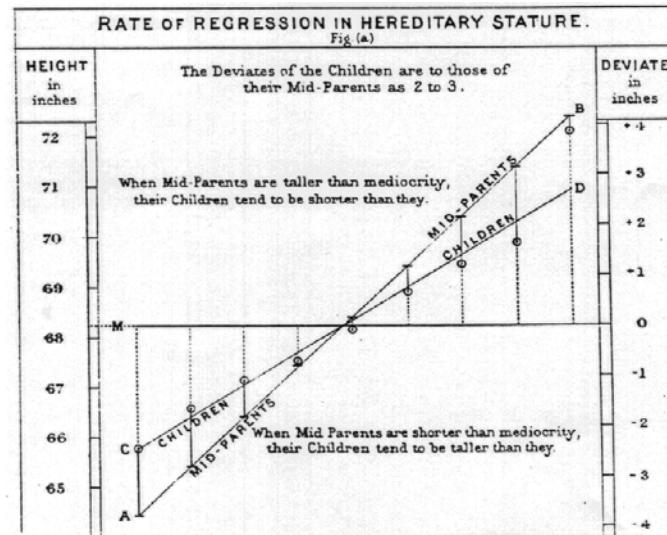
- 영국의 유전학자 Francis Galton(1822~1911)에 의해 도입
- "REGRESSION towards MEDIOCRITY in HEREDITARY STATURE", Journal of the Anthropological Institute 15 (1886), 246–263
- Karl Pearson(1903)에 의해 모형 정립

TABLE I.

NUMBER OF ADULT CHILDREN OF VARIOUS STATURES BORN OF 205 MID-PARENTS OF VARIOUS STATURES.
(All Female heights have been multiplied by 1·08).

Heights of the Mid-parents in inches.	Heights of the Adult Children.												Total Number of Adult Children.	Medians.		
	Below	62·2	63·2	64·2	65·2	66·2	67·2	68·2	69·2	70·2	71·2	72·2	73·2			
Above	4	5	
72·5	1	1	1	1	2	2	2	4	19	
71·5	1	3	4	3	5	10	4	9	2	2	43	
70·5	1	1	1	1	12	18	14	7	4	3	3	11	
69·5	1	16	4	17	27	20	33	25	20	11	4	183	
68·5	1	7	11	6	25	31	34	48	21	18	4	3	41	
67·5	3	5	1	15	30	38	28	30	10	11	4	..	219	
66·5	3	3	5	8	17	17	12	13	4	211	
65·5	1	9	7	11	11	7	7	5	2	1	78	
64·5	1	4	1	6	5	5	2	2	20	
Below	..	1	2	4	1	2	2	1	1	14	1	
Totals	..	5	7	32	59	45	117	138	120	167	99	64	41	17	928	205
Medians	66·3	67·8	67·9	67·7	67·9	68·3	68·5	69·0	69·0	70·0

NOTE.—In calculating the Medians, the entries have been taken as referring to the middle of the squares in which they stand. The reason why the headings run 62·2, 63·2, &c., instead of 62·5, 63·5, &c., is that the observations are unequally distributed between 62 and 63, 63 and 64, &c., there being a strong bias in favour of integral inches. After careful consideration, I concluded that the headings, as adopted, best satisfied the conditions. This inequality was not apparent in the case of the Mid-parents.



❖ 회귀분석(Regression)

- 인과관계를 검정하는 분석방법
- 하나 또는 여러 개의 원인변수(독립변수)가 다른 변수(종속변수)에 영향을 미칠 때
- 독립변수 (Independent Variables: IV) : 종속변수에 영향을 주는 변수
- 종속변수 (Dependent Variable: DV) : 다른 변수에 의해 영향을 받는 변수

$$X(IV) \rightarrow f(\text{process}) \rightarrow Y(DV)$$



$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots$$

- 변수들 간의 관계를 설명하고 예측하는 분석방법

❖ 회귀식을 통한 예측

- 독립변수의 변화에 따라 종속변수의 값이 어떻게 변할지를 예측
- 예) 주택면적에 따른 가격예측

$$Y(\text{가격}) = \beta_0 + \beta_1 x_1 = 93,737 + 7.429 \times \text{면적}$$

❖ 두 변수 사이의 영향관계를 설명

- 온라인게임의 충성도에 영향을 주는 요인

Model Coefficients - 충성도

Predictor	Estimate	SE	t	p	Stand. Estimate
Intercept	-1.360	0.349	-3.894	< .001	
도구	0.020	0.031	0.647	0.518	0.018
보상	0.078	0.019	4.091	< .001	0.113
정보	0.110	0.027	4.097	< .001	0.116
디자인	0.157	0.016	9.799	< .001	0.289
공동체	0.136	0.019	7.277	< .001	0.245
몰입	0.319	0.027	11.669	< .001	0.400

❖ 선형회귀분석

- 독립변수와 종속변수와의 관계가 선형(직선)일 때 이용

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \varepsilon_i$$

❖ 비선형회귀분석

- 독립변수와 종속변수의 관계가 선형이 아닐 때

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2^2 + \cdots + \varepsilon_i$$

❖ 로지스틱 회귀분석

- 종속변수의 값이 이분형(명목변수)일 때

$$Y(0,1) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \varepsilon_i$$

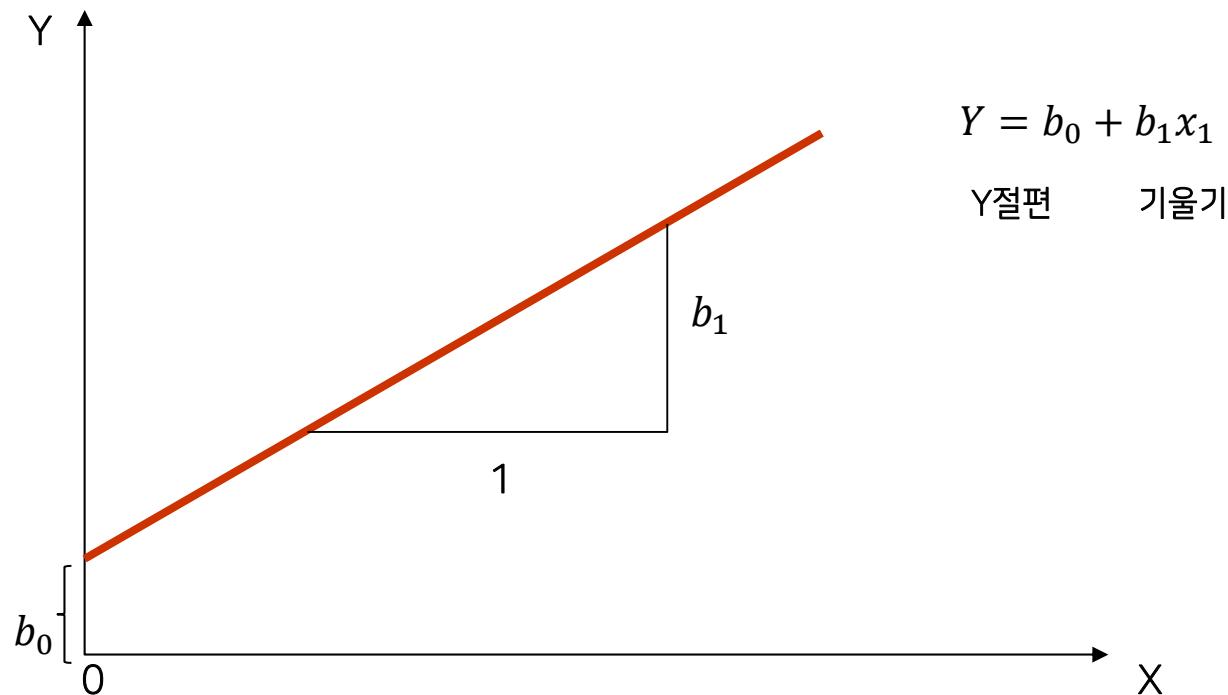
Regression(예측)

LGE Internal Use Only

- ❖ 선형회귀모형

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \varepsilon_i$$

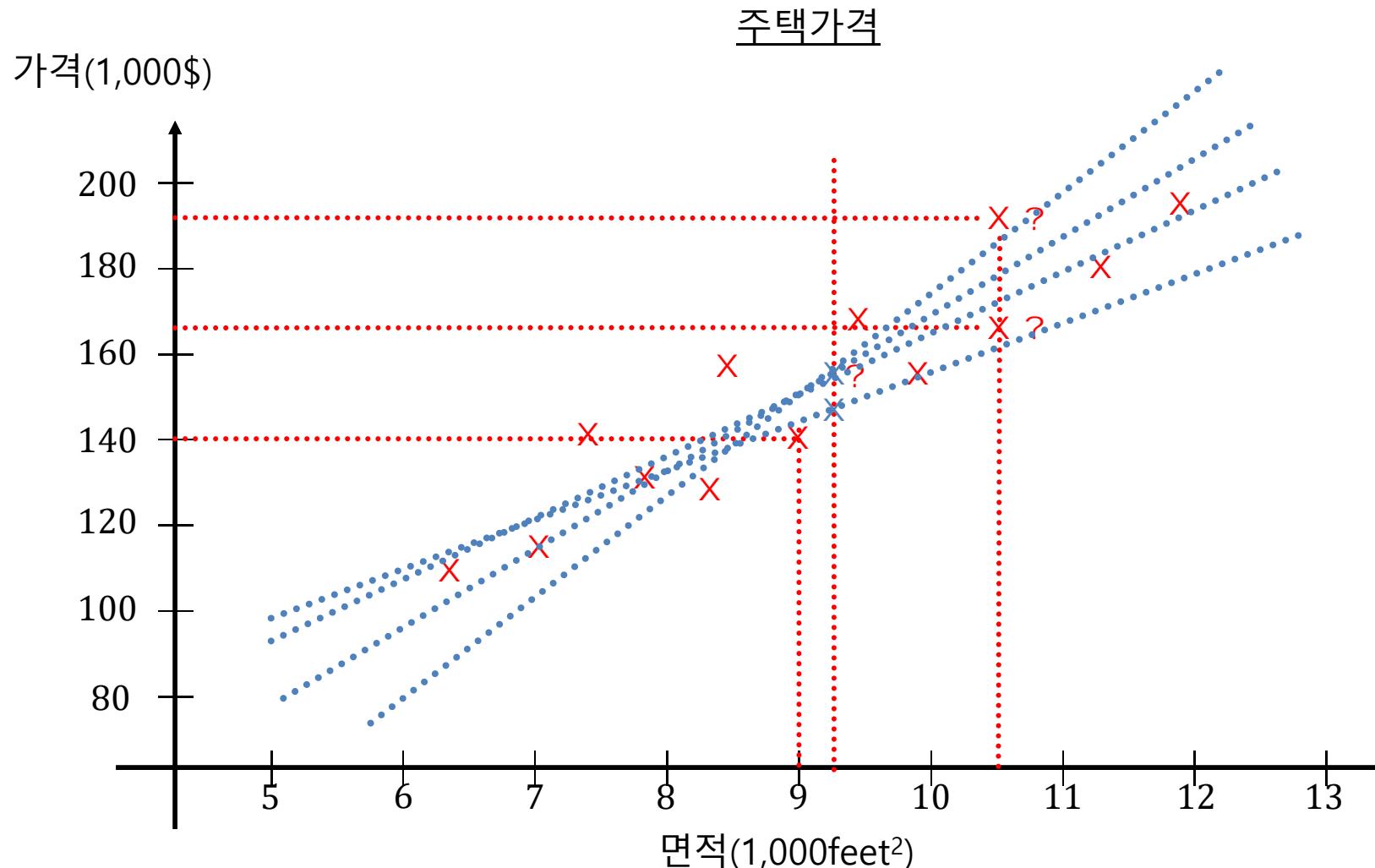
모수(β_0, β_1) \leftarrow 추정값(b_0, b_1)



Regression(예측)

LGE Internal Use Only

- ❖ 어떤 직선이 더 좋은가?

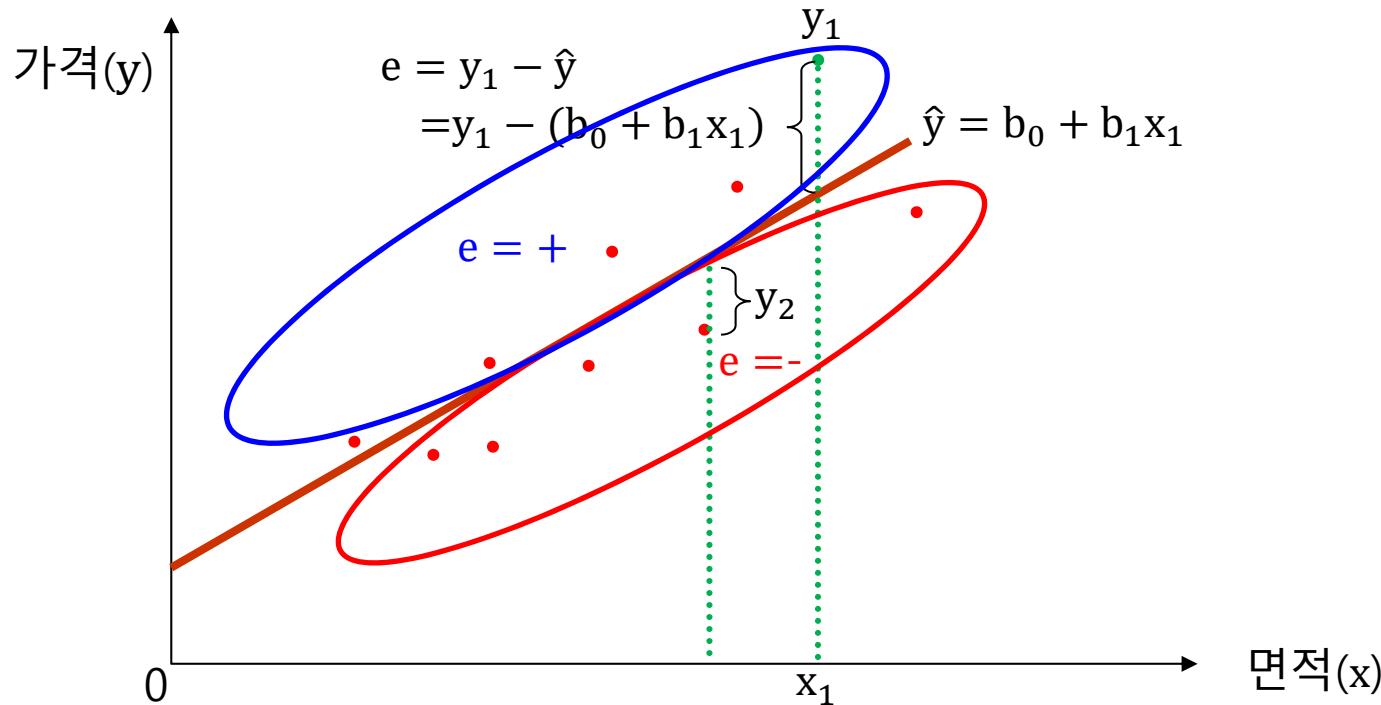


Regression(예측)

LGE Internal Use Only

- ❖ 최소제곱법(Method of least Squares)
 - +, -를 없애주기 위해서 square($(e)^2$) error 사용

$$MSE(e) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^n (y_i - (b_0 + b_1 x_i))^2$$



❖ 분산 모델 이용

$$\sum_{i=1}^n (x_i - \bar{x})^2 \longrightarrow \sum_{i=1}^n (y_i - (b_0 + b_1 x_i))^2$$

❖ 편차(deviation)

- 자료가 평균을 중심으로 어떻게 분포 → 분산 및 표준편차
- 편차(개체값-평균): $(x_i - \bar{x})$

❖ 잔차(residuals)

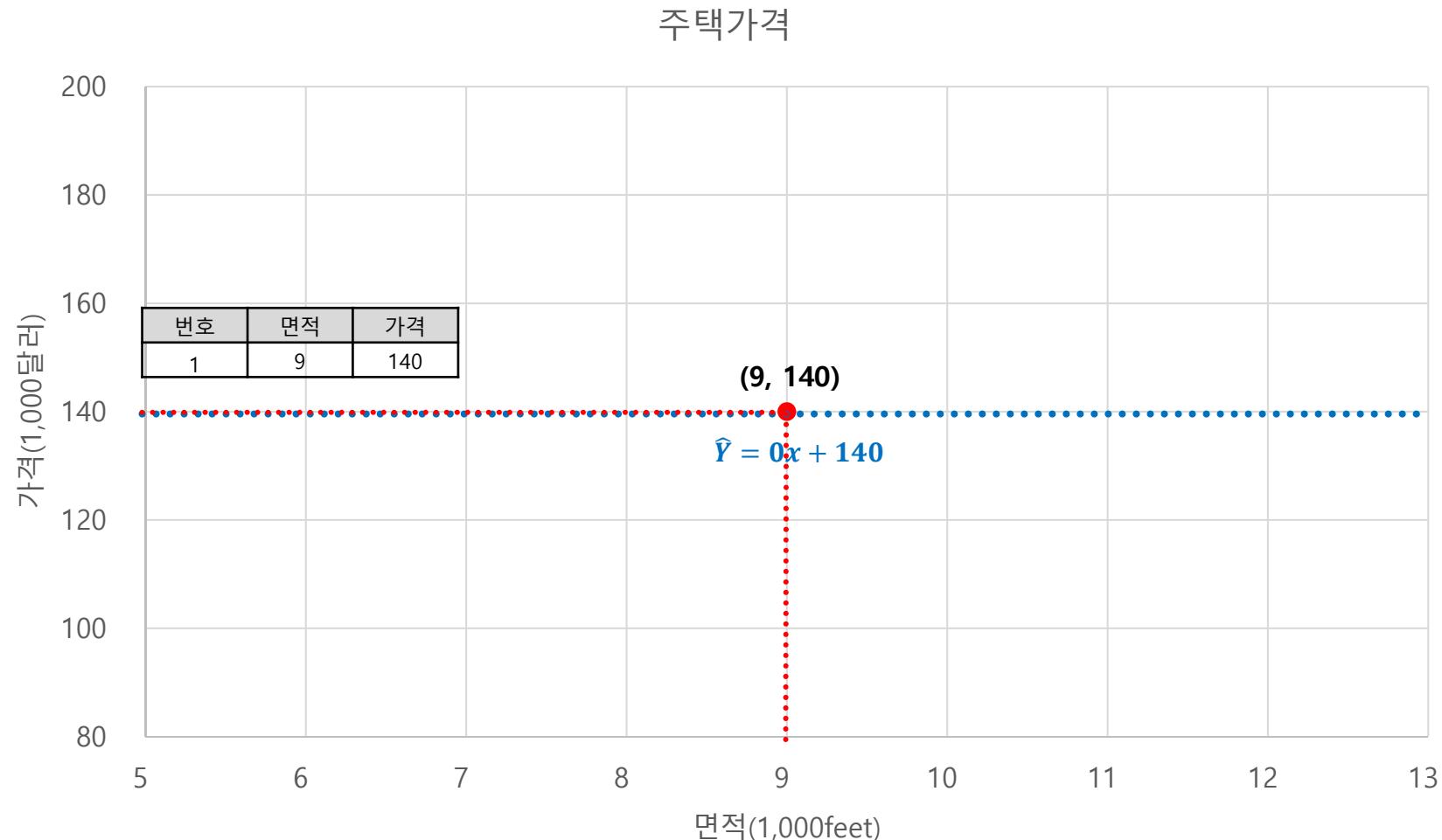
- 회귀분석에서 사용 → 모델의 적합도(회귀직선)
- $e_i = (y_i - \hat{y}_i)$

❖ 오차(error)

- 데이터마이닝 성능평가에서 사용 → 모형의 성능 (실제값 예측)
- $e_i = (y_i - \hat{y}_i)$

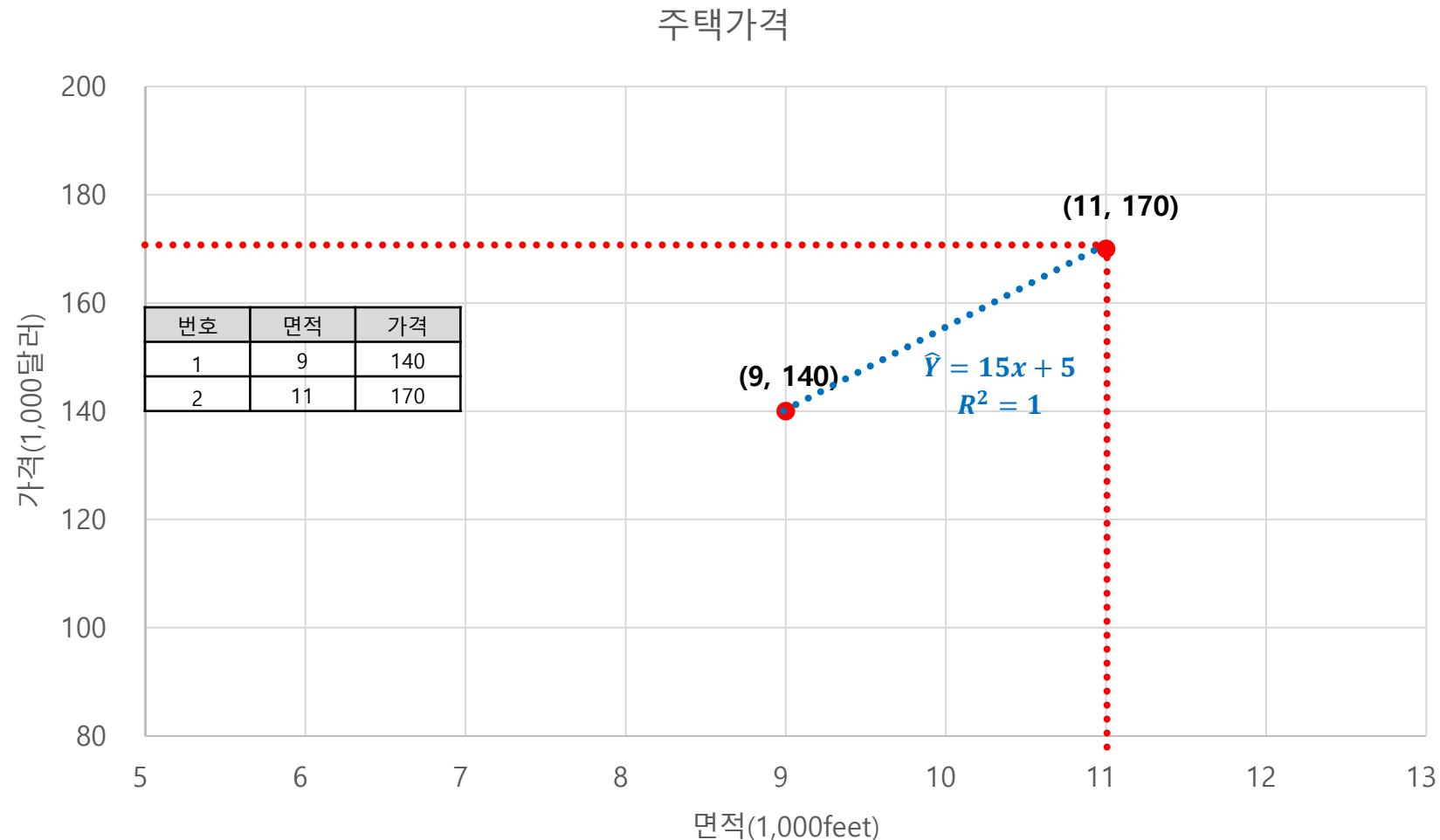
Regression(예측)

LGE Internal Use Only



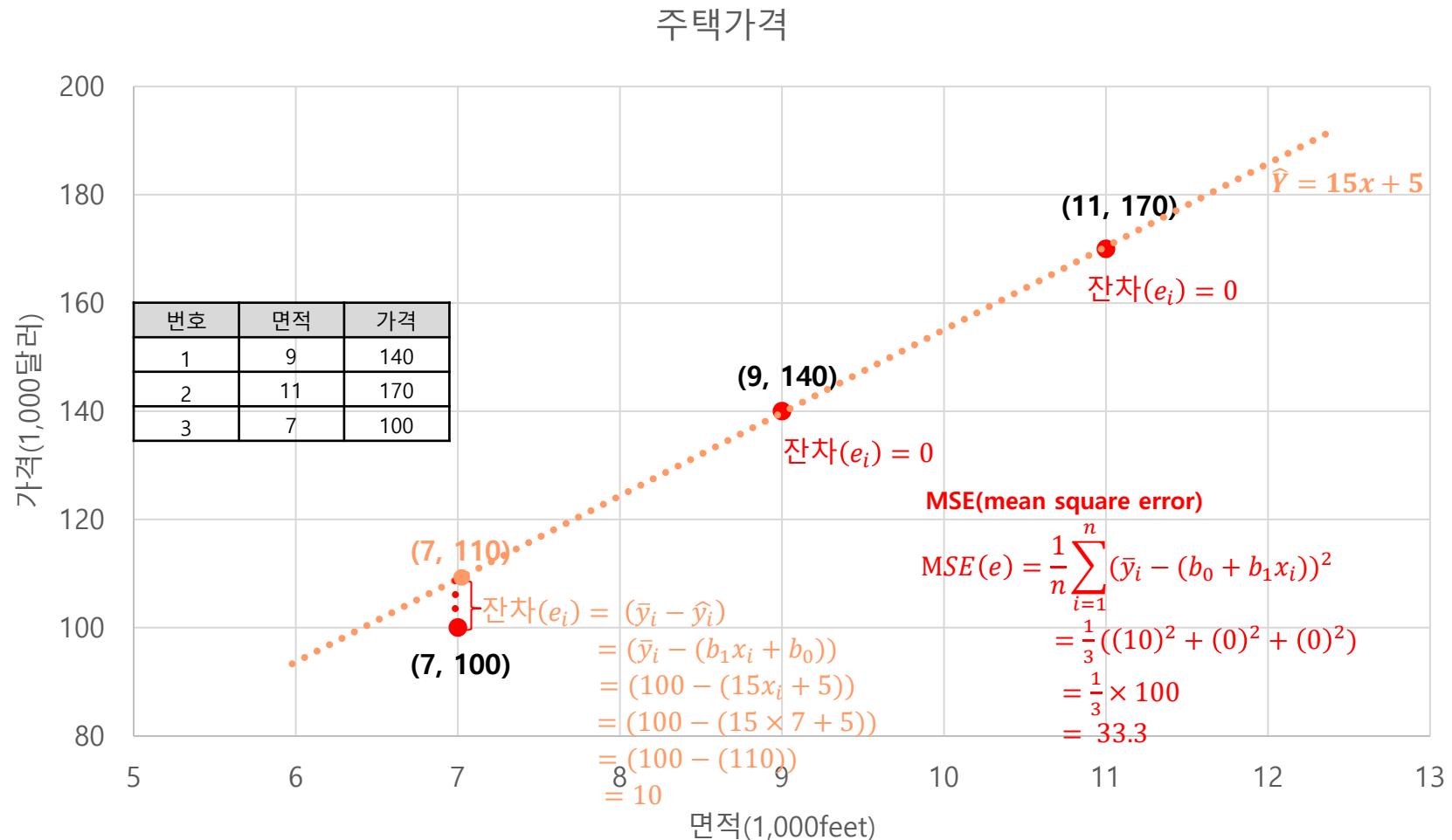
Regression(예측)

LGE Internal Use Only



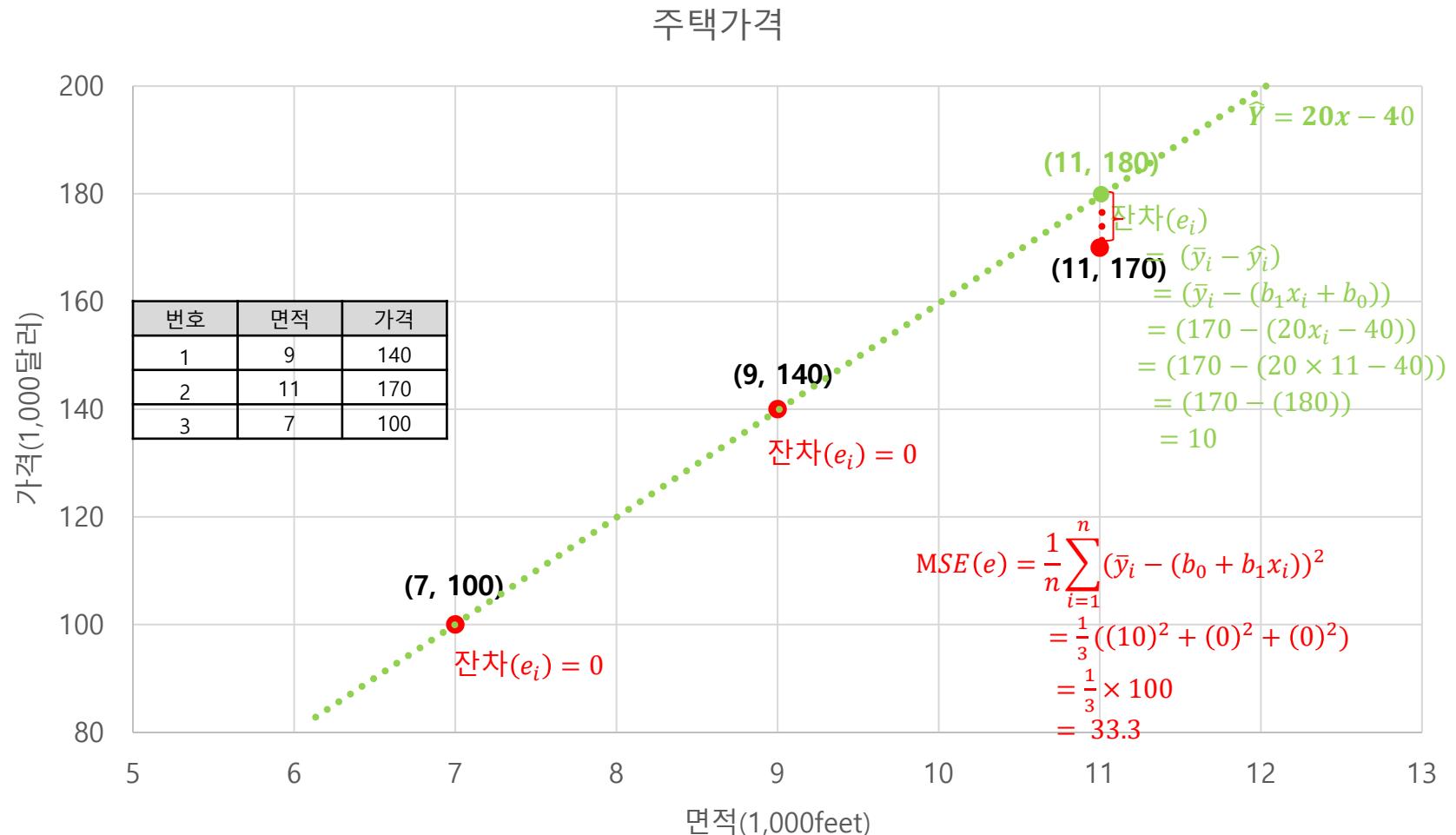
Regression(예측)

LGE Internal Use Only



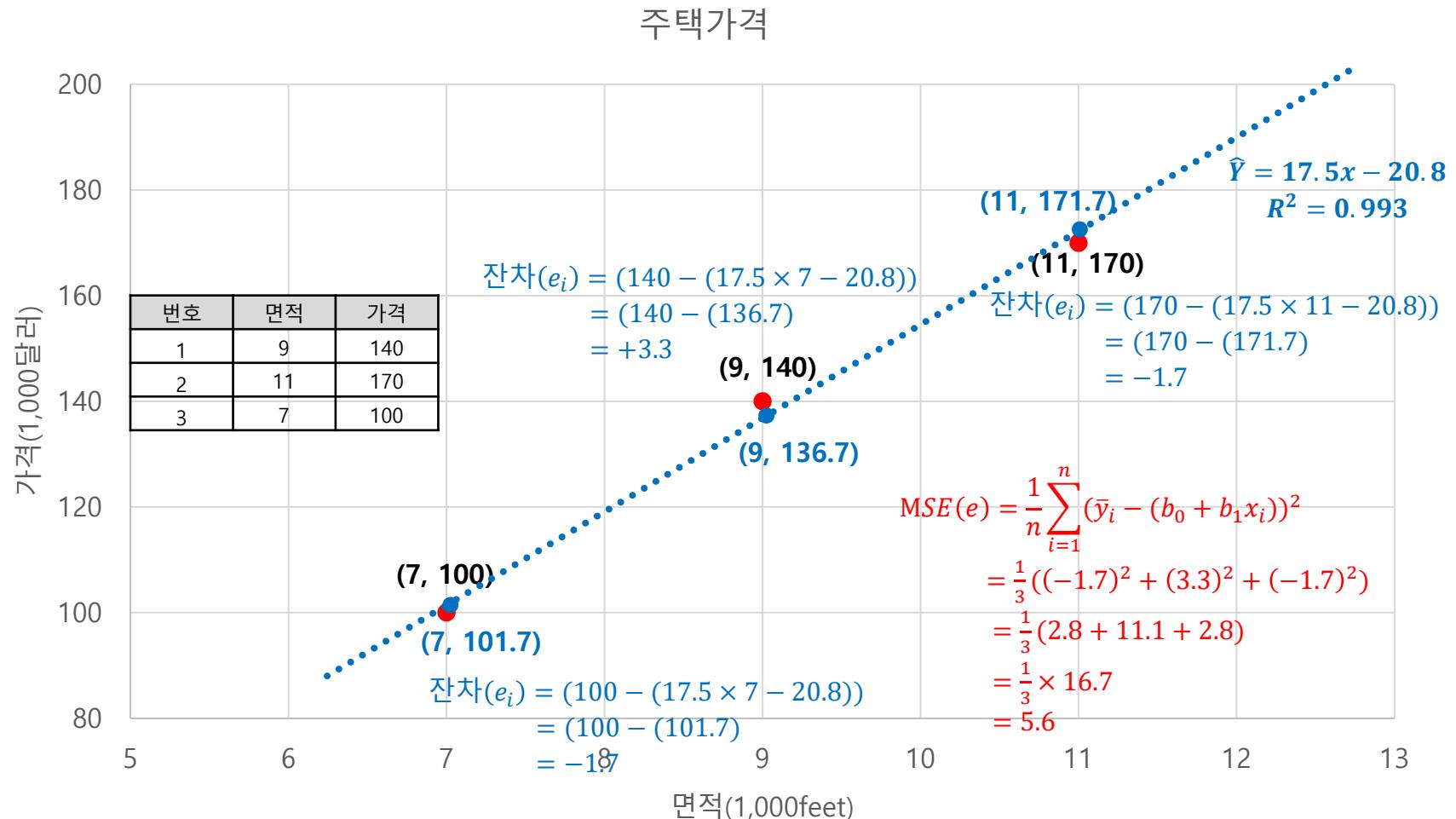
Regression(예측)

LGE Internal Use Only



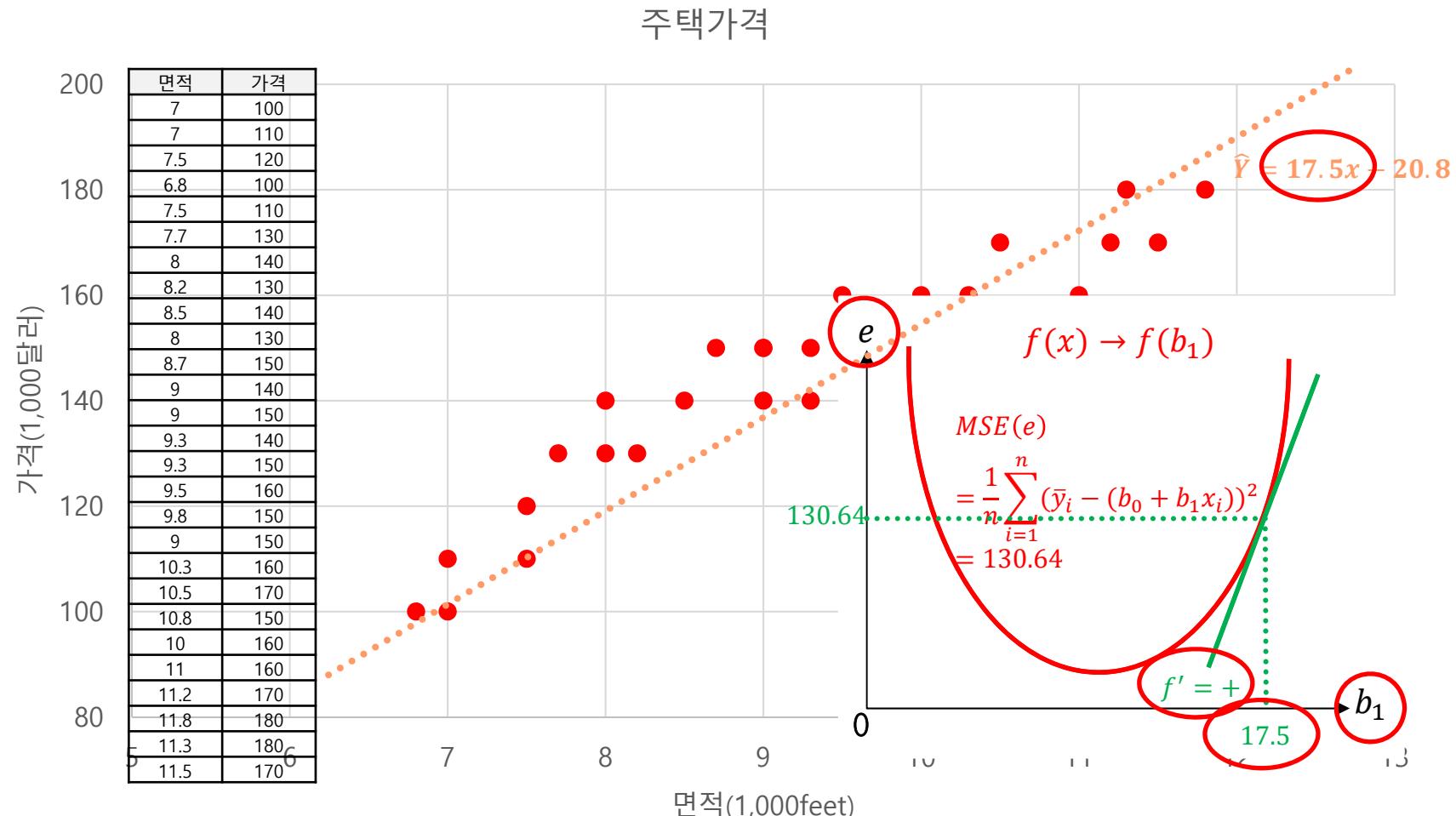
Regression(예측)

LGE Internal Use Only



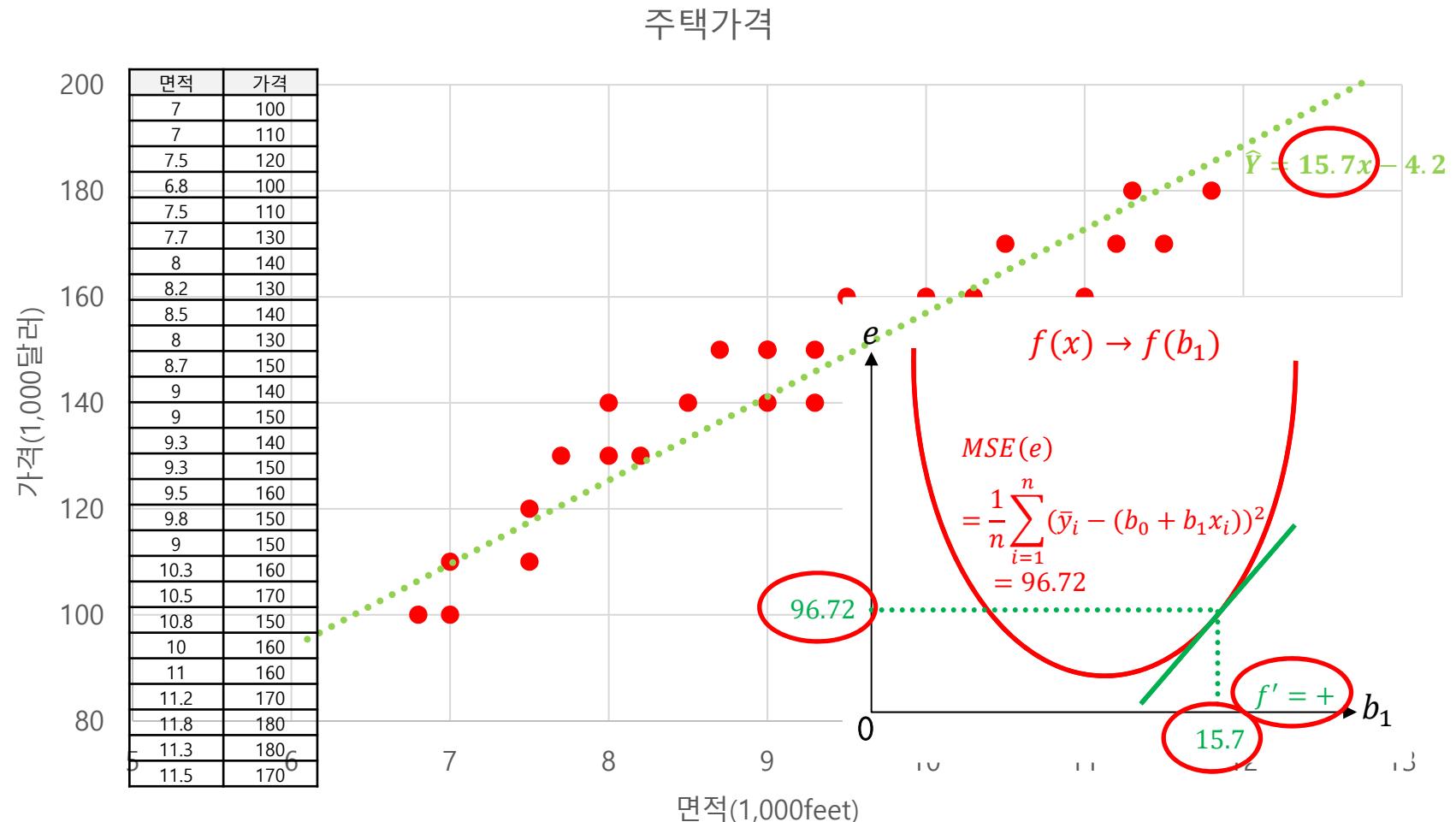
Regression(예측)

LGE Internal Use Only



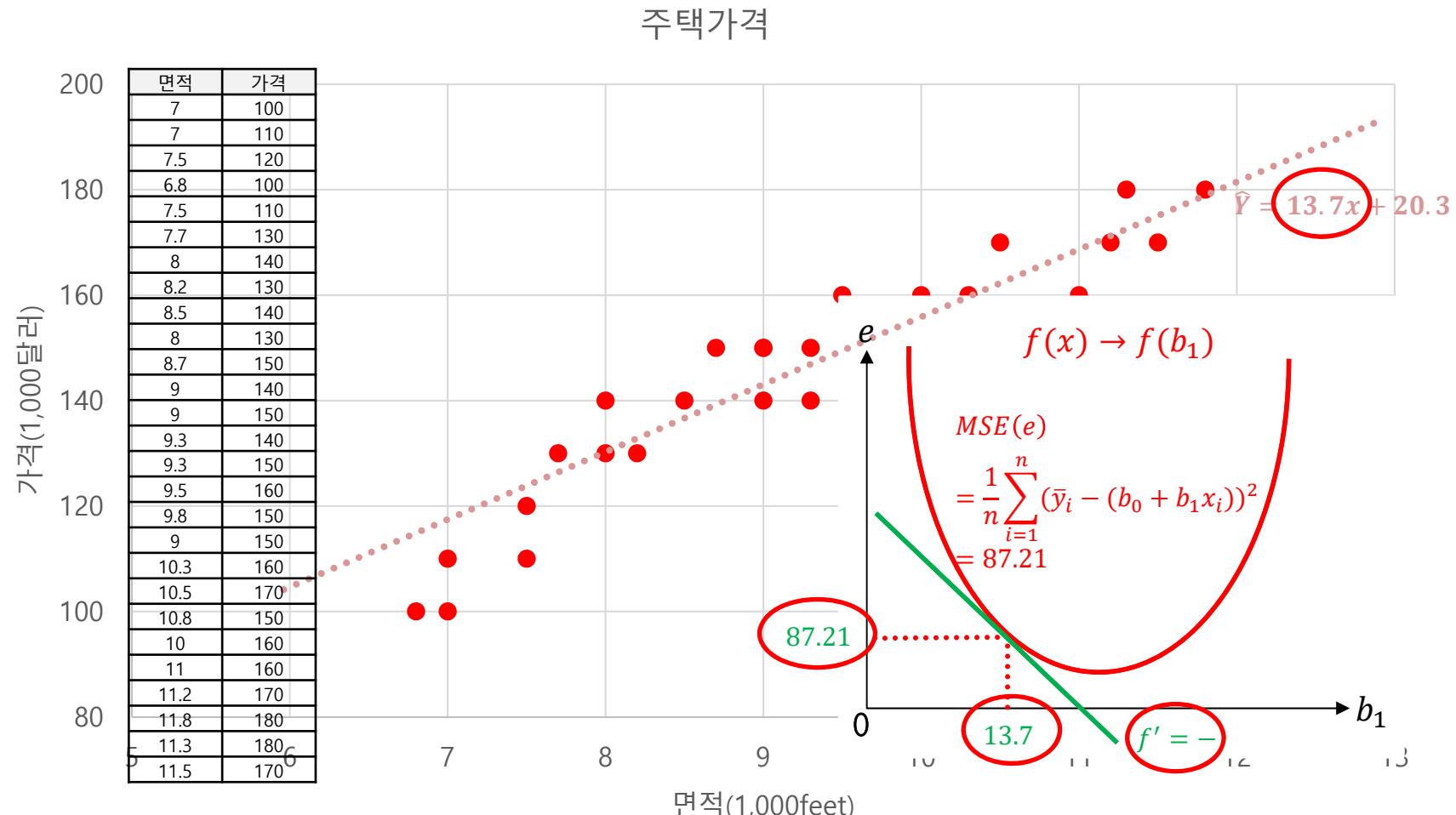
Regression(예측)

LGE Internal Use Only



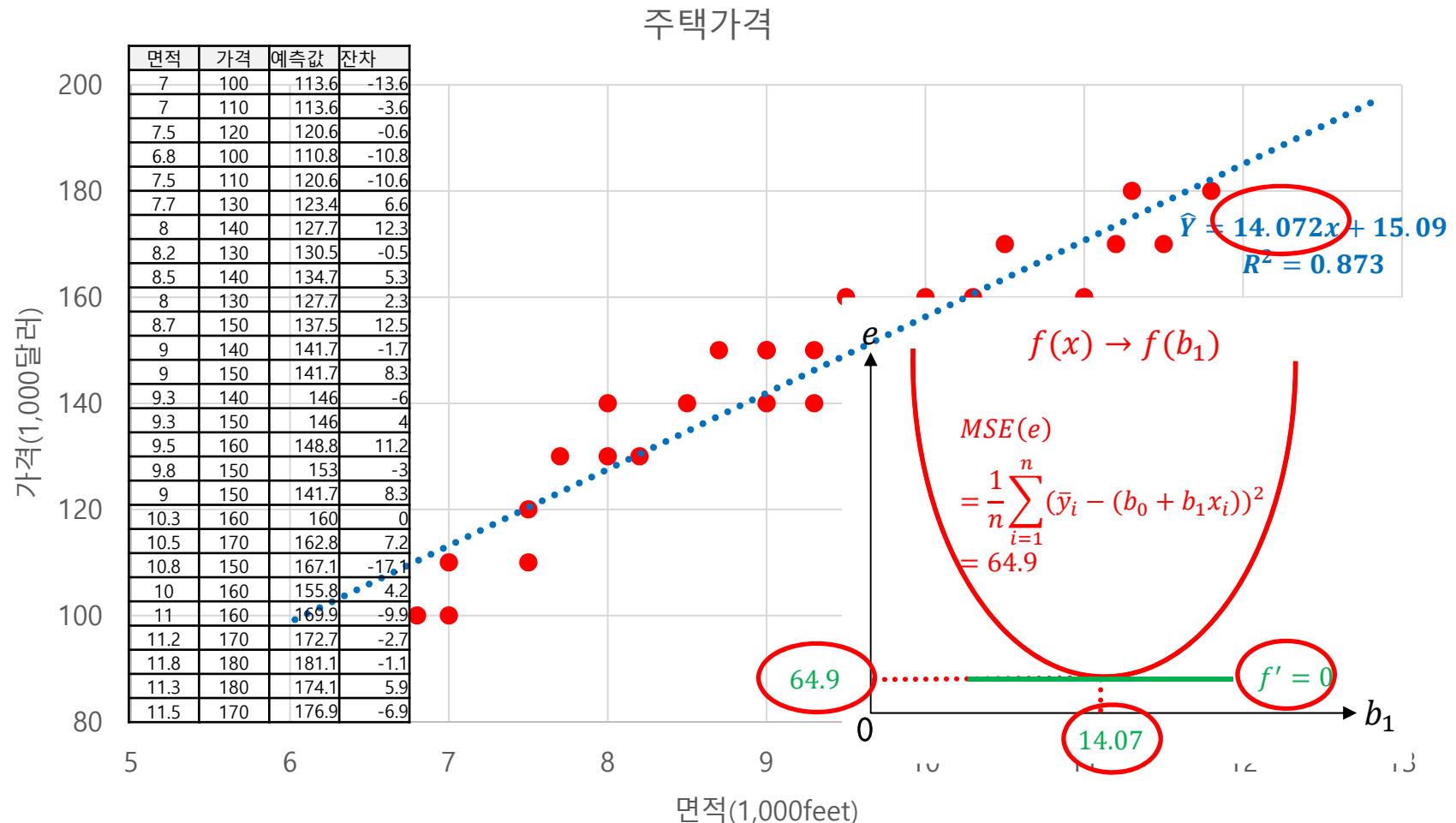
Regression(예측)

LGE Internal Use Only



Regression(예측)

LGE Internal Use Only

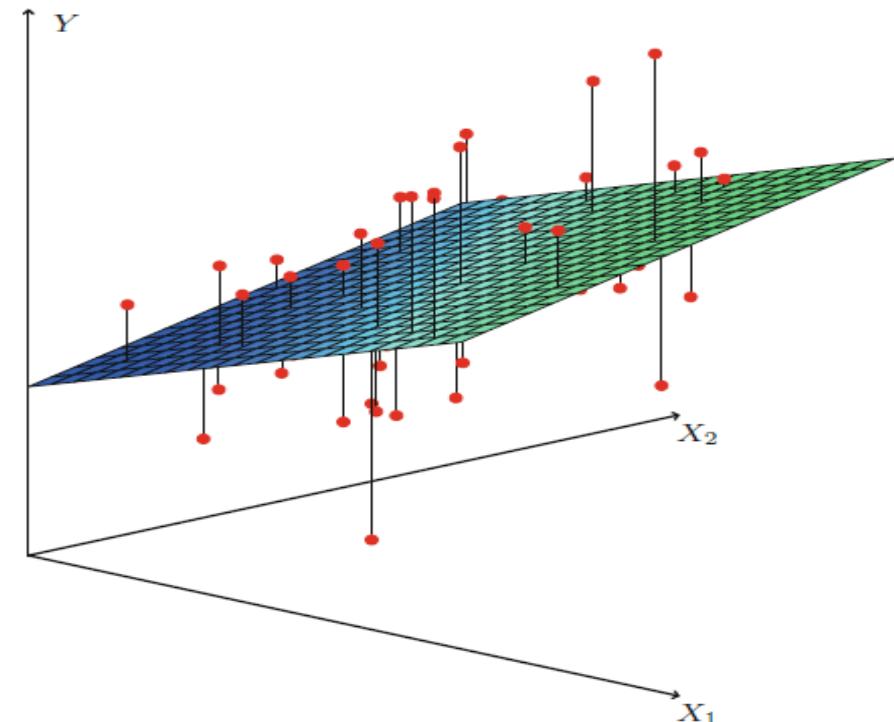
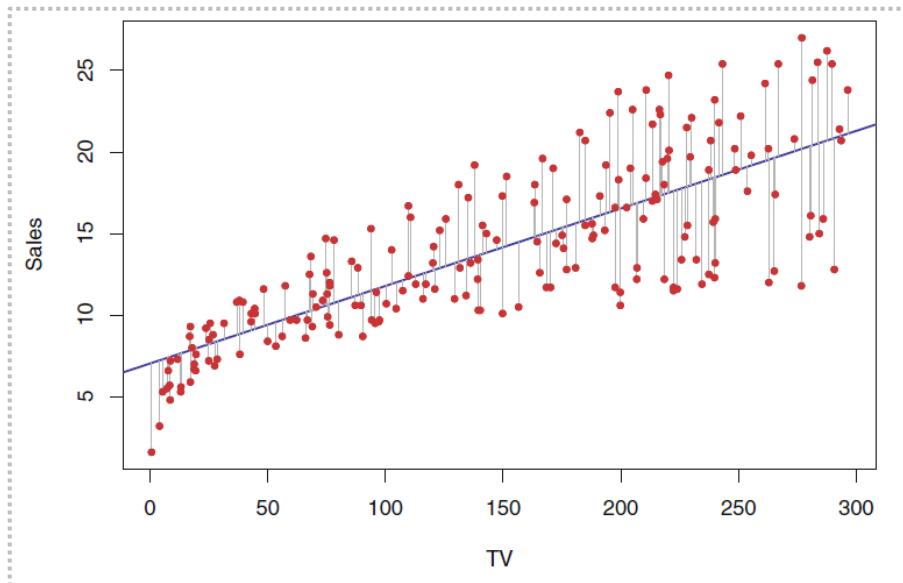


Regression(예측)

LGE Internal Use Only

$$Y = \beta_0 + \beta_1 x_1$$

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$



출처: https://godongyoung.github.io/머신러닝/2018/01/20/ISL-linear-regression_ch3.html

Regression(예측)

LGE Internal Use Only

❖ 최소제곱법(Method of least Squares)

- 적합된 회기식에 의한 예측치 \hat{y}_i 와 관찰치 y_i 의 차이인 잔차들의 제곱의 합이 최소가 되도록 회귀계수를 추정하는 방법(미분)

$$\sum_{i=1}^n (e_i)^2 = \sum_{i=1}^n (y_i - (b_0 + b_1 x_i))^2$$

$$\begin{aligned}\frac{\partial D}{\partial b_0} &= -2 \sum_{i=1}^n (y_i - (b_0 + b_1 x_i)) = 0 \\ \frac{\partial D}{\partial b_1} &= -2 \sum_{i=1}^n x_i (y_i - (b_0 + b_1 x_i)) = 0\end{aligned}$$

$$b_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} = \frac{S_{xy}}{S_{xx}} \longrightarrow \hat{\beta}_1$$

$$\beta_1 \rightarrow b_1 \rightarrow \hat{\beta}_1$$

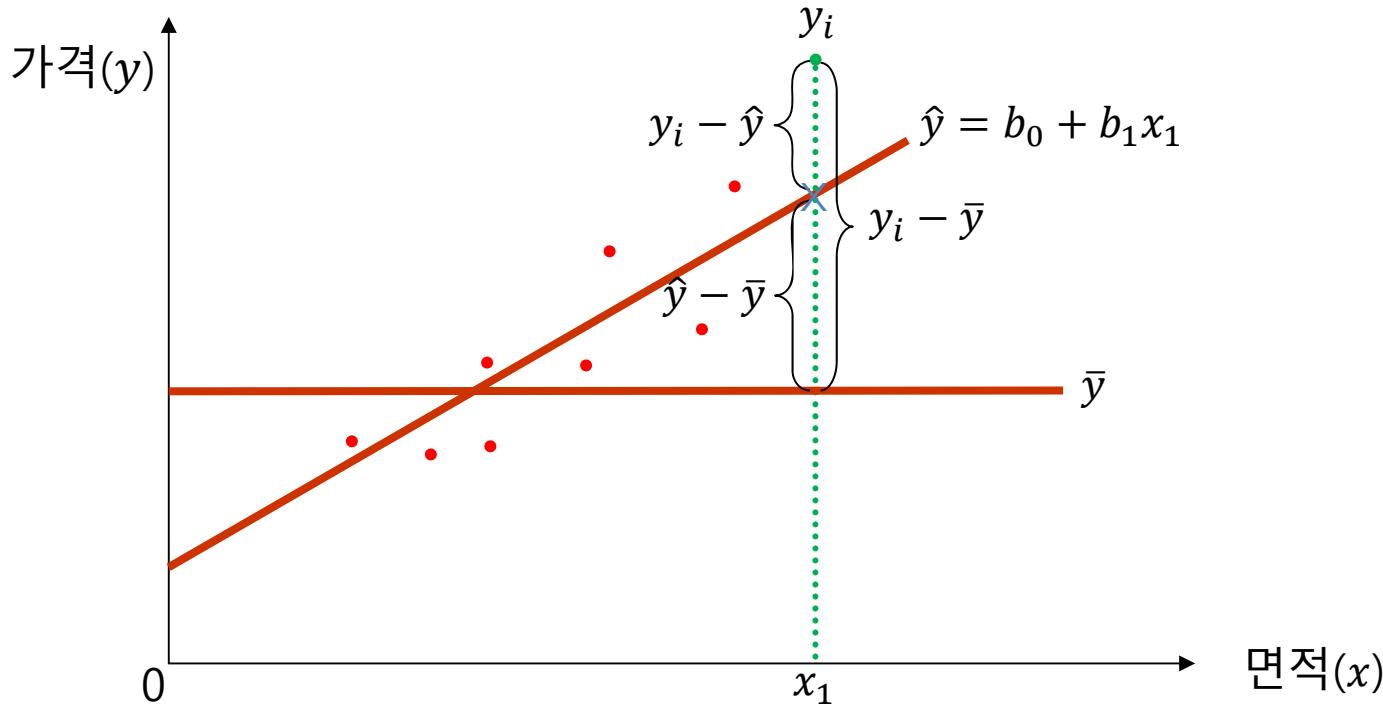
$$b_0 = \bar{y} - b_1 \bar{x} \longrightarrow \hat{\beta}_0$$

$$\beta_0 \rightarrow b_0 \rightarrow \hat{\beta}_0$$

Regression(예측)

LGE Internal Use Only

❖ 변동의 분해(분산분석)



$$SST = SSR + SSE$$

Regression(예측)

LGE Internal Use Only

❖ 회귀식의 분산분석

요인	제곱합 (SS)	자유도(df)	평균제곱 (MS)	F
회귀 모형	$SSR = \sum (\hat{y} - \bar{y})^2$	k	$MSR = \frac{SSR}{k}$	$\frac{MSR}{MSE}$
잔차	$SSE = \sum (y_i - \hat{y})^2$	$n - (k - 1)$	$MSE = \frac{SSE}{n - (k - 1)}$	
총계	$SST = \sum (y_i - \bar{y})^2$	$n - 1$	$MSR = \frac{SSR}{k}$	

❖ 검정통계량

$$F = \frac{MSR}{MSE} \sim F(k, n - k - 1)$$

Regression(예측)

LGE Internal Use Only

❖ 회귀식의 분산분석

	제곱합	df	평균 제곱	F	유의수준
회귀분석	60,855,385,221	1	60,855,385,221	1,051	.000
잔차	6,834,073,748	119	57,915,879		
총계	67,689,458,969	120			

❖ 검정통계량

$$F = \frac{MSR}{MSE} = \frac{60,855,385,221}{57,915,879} = 1,051 > F_{(0.05, 1, 119)} = 3.92$$

$$p-value = P(F > 479.842) = 0.000 < \alpha = 0.05$$

❖ 회귀식의 적합도

- 회귀모형이 얼마나 종속변수를 잘 설명하고 있는지
- 결정계수(R^2): 총변동 중에서 회귀모형에 의해 설명되는 비율
- 수정 결정계수(adjusted R^2): 결정계수는 독립변수가 많아질수록 증가하기 때문에 이를 수정

$$R^2 = \frac{\text{회귀선에 의해 설명되는 변동}}{\text{전체변동}} = \frac{\sum(\hat{y} - \bar{y})^2}{\sum(y_i - \bar{y})^2} = \frac{SSR}{SST}$$
$$= \frac{60,855,385,221}{67,689,458,969} = 0.899$$

Model Fit Measures			
Model	R	R^2	Adjusted R^2
1	0.948	0.899	0.898

❖ 회귀계수(β) 검정

- 귀무가설(H_0): 두 변수간에는 인과관계(영향력)가 없다.

$$H_0: \beta_1 = 0$$

- 연구가설(H_1): 두 변수간에는 인과관계(영향력) 가 있다.

$$H_1: \beta_1 \neq 0$$

- 검정통계량

$$\frac{\hat{\beta}_1 - \beta_1}{\sqrt{\frac{MSE}{S_{xx}}}} \sim t(n-2)$$

$$\frac{7.429 - 0}{0.229} = 32.415 > 1.980$$

Model Coefficients - 가격

Predictor	Estimate	SE	t	p
Intercept	93736.519	2143.971	43.721	< .001
연면적	7.429	0.229	32.415	< .001

❖ 회귀분석 가정검정

- ANOVA와 같이 잔차검정
- 차이점- 회귀식을 기준으로 한 잔차를 이용
- 회귀모형식

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \varepsilon_i$$

$$\varepsilon_{ij} \rightarrow e_{ij} \sim N(0, \sigma^2)$$

$$* ANOVA: Y_{ij} = \mu + (\mu_i - \mu) + \varepsilon_{ij}$$

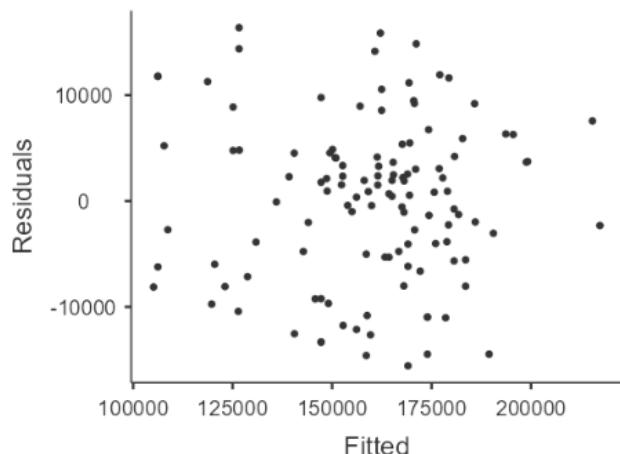
전체평균 + 처리효과 + 측정오차

- 등분산성: 종속변수의 분산은 독립변수의 값에 관계없이 동일해야 한다.
- 정규성: 오차(잔차)는 정규분포를 이루어어야 한다.
- 독립성: 오차(잔차)는 서로 독립적이어야 한다

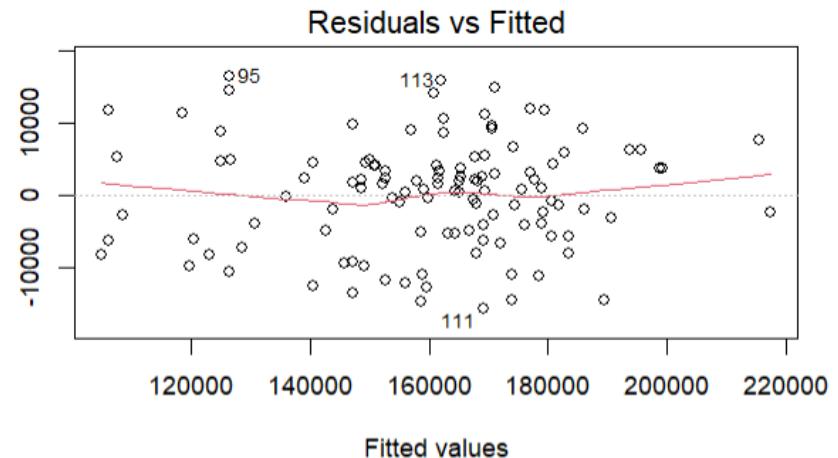
Regression(예측)

❖ 잔차의 등분산성

- 오차 ε_i 는 모든 i 에 대하여 평균이 [0]이며 분산이 σ^2 인 정규분포를 따름
- 만약 등분산성을 만족하지 않으면 종속변수의 변환: $\log(y_i)$
- 비선형 회귀분석으로 변환: $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2^2 + \dots + \varepsilon_i$
- 잔차 plot으로 확인
- R: ncvTest(Breusch-Pagan검정)



<jamovi>

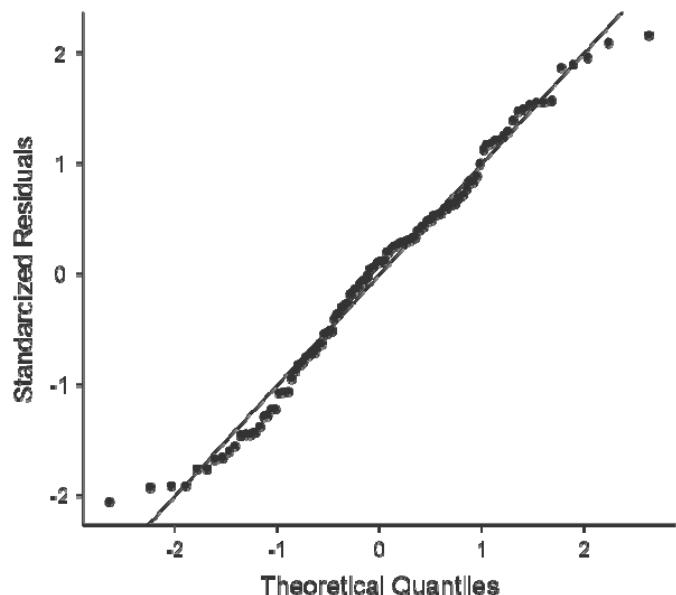


<R>

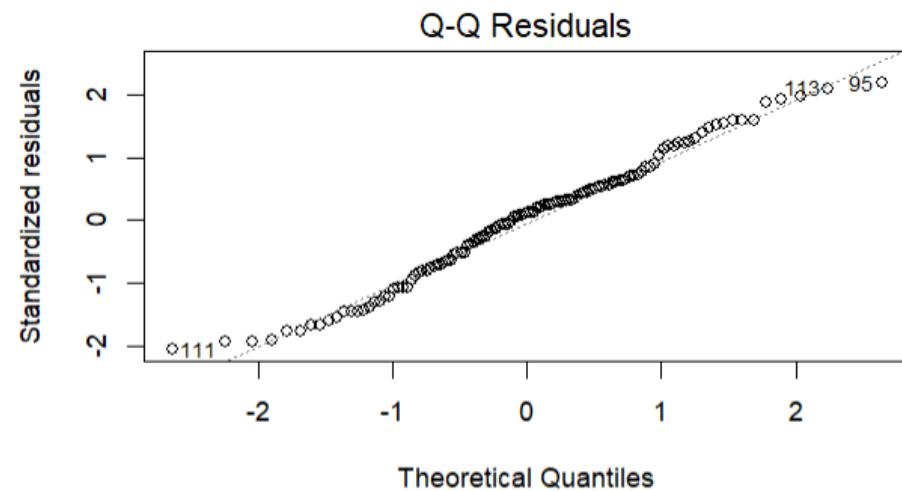
Regression(예측)

❖ 잔차의 정규성

- 모든 잔차는 정규분포를 이루어어야 한다.
- Q-Q, shapiro.test, jarque-bera검정



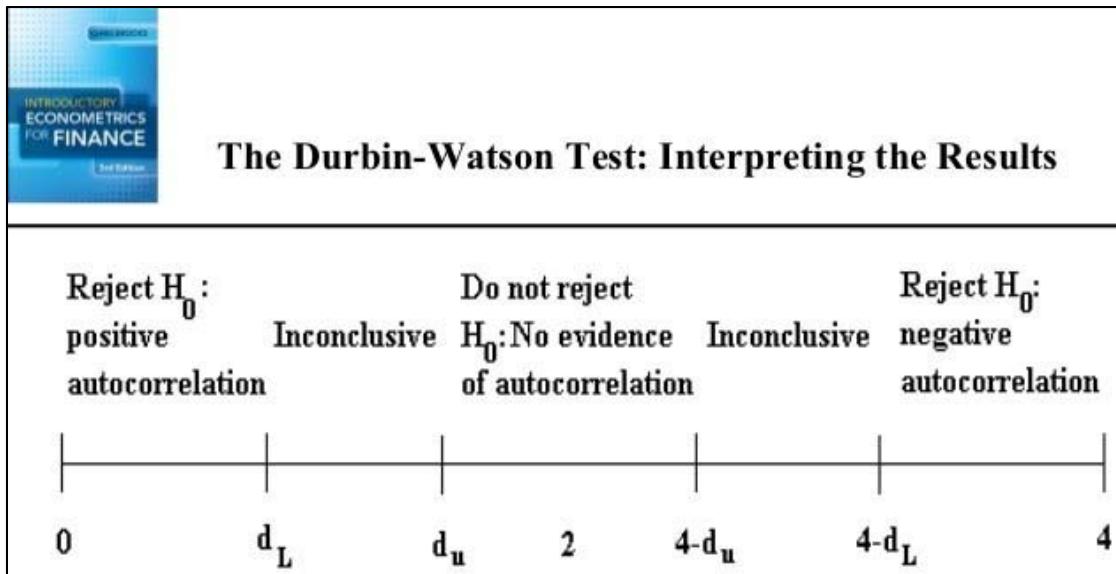
<jamovi>



<R>

❖ 잔차의 독립성

- 회귀분석에서 오차 ε_i 는 서로 독립적이라고 가정
- 오차의 자기상관이란 오차들이 서로 자기상관관계가 있음을 나타냄
- 자기상관이 있으면 값이 높게 나타남
- 검정방법: Durbin-Watson의 통계량



출처: <https://www.researchgate.net/post/How-to-choose-significance-level-for-Durbin-Watson-Statistics>

❖ 다중공선성(multicollinearity)

- 다중 회귀분석에서 독립변수가 많이 투입되면 결정계수(회귀식의 설명력)는 높아지지만 회귀계수는 신뢰하지 못하게 됨
- 다중공선성이 높은 독립변수가 있다면 그 변수는 삭제
- 검정방법:
 - 공차한계(Tolerance) < 0.1,
 - VIF(Variance Inflation Factor)>10이면 다중공선성 있음

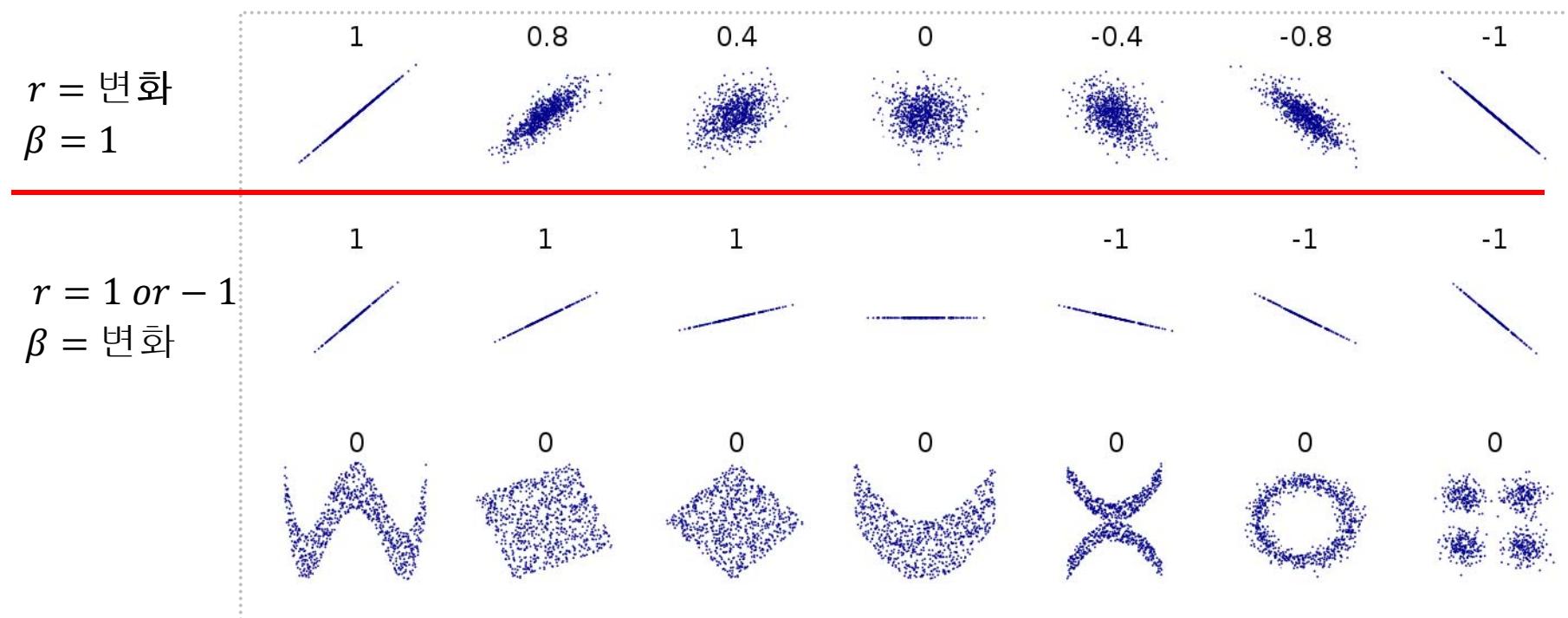
Collinearity Statistics

	VIF	Tolerance
연면적	2.134	0.469
품질	1.329	0.753
상태	1.965	0.509
건축년도	2.478	0.404
리모델링년도	1.485	0.673
지하면적	2.033	0.492
차고면적	1.352	0.740
면적_1층	3.098	0.323
면적_2층	2.903	0.344

Regression(예측)

❖ 상관계수와 회귀계수와의 관계

- 상관계수: 두 변수와의 관계성(연관성→ 선형성: linear relationship)
- 회귀계수: 독립변수에 대한 종속변수의 변화량(기울기:slope) 예) IV → DV



출처: https://en.wikipedia.org/wiki/Correlation_and_dependence

❖ 상관계수와 회귀계수

- 상관계수 : x 와 y 의 관련성(선형성)
- 회귀계수 : x 와 y 의 관련성 \times

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} = \frac{s_{xy}}{s_x s_y}$$
$$b_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} = \frac{s_{xy}}{s_{xx}}$$

❖ 표준화 계수와 비표준화 계수

- 비표준화 계수: 원래 데이터 값을 이용하여 기울기 구함
- 표준화 계수: 원래 데이터를 표준화한 이후에 기울기를 구함

$$Y = \beta_0 + \beta_1 x_1 + \varepsilon_i$$

원데이타

$$Y = \beta_0 + \beta_1 x_1 + \varepsilon_i$$

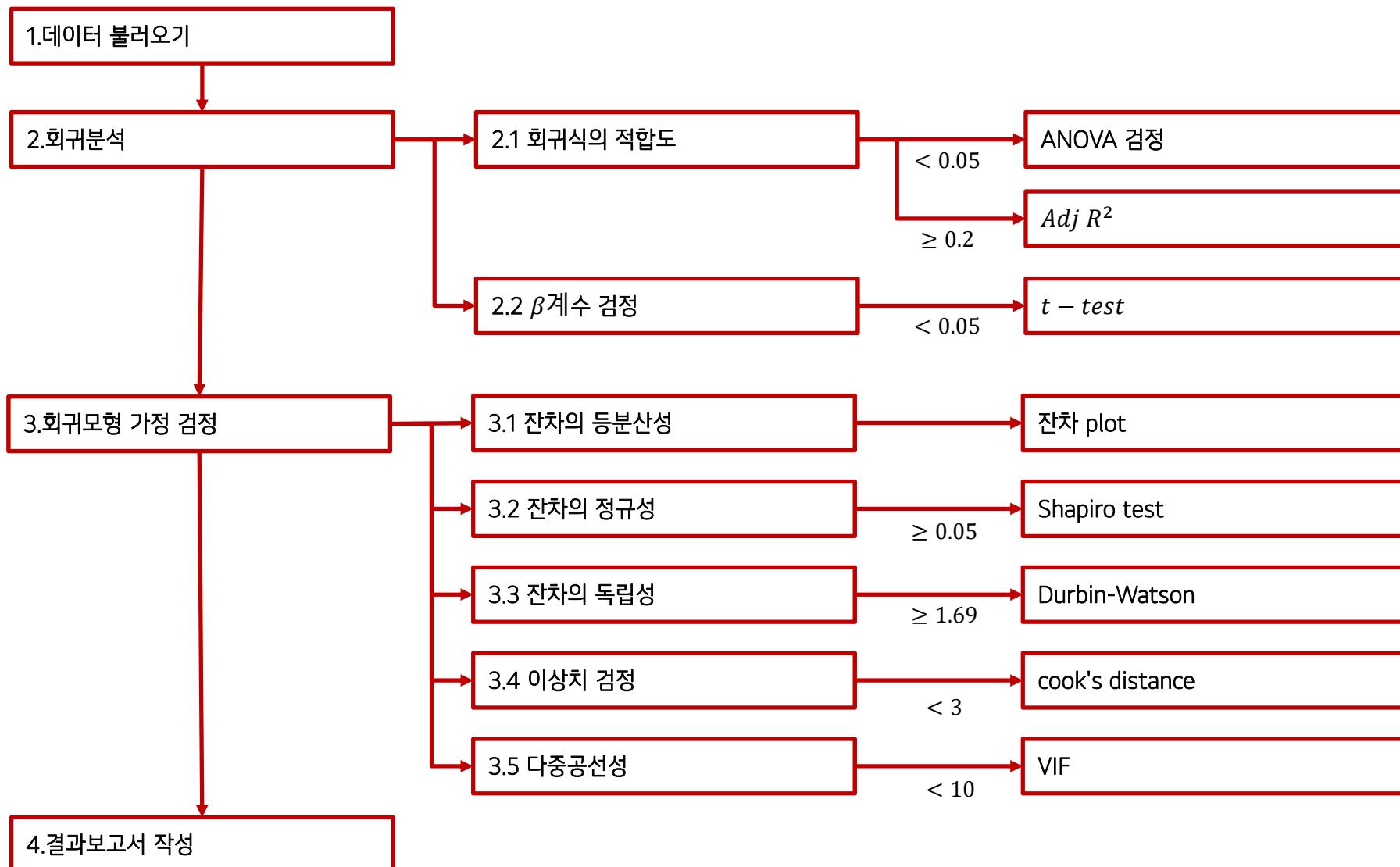
$$y_i = \frac{(y_i - \bar{y})}{\sqrt{s_{yy}}} \quad x_{ji} = \frac{(x_{ij} - \bar{x}_j)}{\sqrt{s_{jj}}}$$

표준화

표준화

Regression(예측)

LGE Internal Use Only



12_1.Regression(예측)

LGE Internal Use Only

The screenshot shows a Jupyter Notebook interface with the following structure:

- Section 1:** 12_1.Regression(예측)
 - <https://www.statsmodels.org/stable/gettingstarted.html>
- Section 2:** 1.기본 package 설정
 - [] # 그래프에서 한글 폰트 인식하기
!sudo apt-get install -y fonts-nanum
!sudo fc-cache -fv
!rm ~/.cache/matplotlib -rf
 - [] !pip install pingouin
*** 런타임 다시 시작
- Section 3:** [1] # 1.기본
 - 2초 import numpy as np # numpy 패키지 가져오기
 - import matplotlib.pyplot as plt # 시각화 패키지 가져오기
 - import seaborn as sns # 시각화
- Section 4:** [2] # 기본세팅
 - 0초 # 테마 설정
sns.set_theme(style = "darkgrid")

At the bottom of the notebook, there is a status bar with the text "✓ 0초 오후 9:23에 완료됨". The top right corner of the window has a toolbar with various icons.

2.데이터 불러오기

LGE Internal Use Only

The screenshot shows a Jupyter Notebook interface with the following structure:

- Section 2.1:** 2.1 데이터 프레임으로 저장
 - 원본데이터(csv)를 dataframe 형태로 가져오기(pandas)
- Code Cell [3]:** mr_df = pd.read_csv('https://raw.githubusercontent.com/echo-bigdata/statistics-python/main/12_1_MR(pred).csv', encoding="cp949")
mr_df.head()
- Data Preview:** A table showing the first 5 rows of the dataset.
- Next steps:** View recommended plots
- Section 2.2:** 2.2 범주형 변수 처리
 - 가변수 처리시 문자로 처리를 해야 변수명 구분이 쉬움
- Code Cell [4]:** mr_df['주거유형'].replace({1:'단독주택', 2:'튜플렉스', 3:'기타'}, inplace=True)
mr_df['판매유형'].replace({1:'보증증서', 2:'신규건물'}, inplace=True)
mr_df['판매조건'].replace({1:'정상판매', 2:'압류(공매도)'}, inplace=True)

mr_df['주거유형'] = mr_df['주거유형'].astype('category')
mr_df['판매유형'] = mr_df['판매유형'].astype('category')
mr_df['판매조건'] = mr_df['판매조건'].astype('category')
mr_df.head()
- Execution Status:** ✓ 0초 오후 9:23에 완료됨

2.데이터 불러오기

LGE Internal Use Only

Next steps: [View recommended plots](#)

▼ 2.2 범주형 변수 처리

- 가변수 처리시 문자로 처리를 해야 변수명 구분이 쉬움

[4] mr_df[['주거유형']].replace({1:'단독주택', 2:'튜플렉스', 3:'기타'}, inplace=True)
mr_df[['판매유형']].replace({1:'보증증서', 2:'신규건물'}, inplace=True)
mr_df[['판매조건']].replace({1:'정상판매', 2:'압류(공매도')}, inplace=True)

mr_df[['주거유형']] = mr_df[['주거유형']].astype('category')
mr_df[['판매유형']] = mr_df[['판매유형']].astype('category')
mr_df[['판매조건']] = mr_df[['판매조건']].astype('category')
mr_df.head()

	id	가격	연면적	품질	상태	건축년도	리모델링년도	지하면적	차고면적	면적_1층	면적_2층	주거유형	판매유형	판매조건	More
0	1	150750	7388	5	6	1959	2002	1063	624	1327	0	단독주택	보증증서	정상판매	More
1	2	131500	4435	6	5	2003	2003	848	420	848	0	기타	신규건물	압류(공매도)	More
2	3	160000	8800	6	6	1964	1964	1251	461	1251	0	단독주택	보증증서	정상판매	More
3	4	187500	13031	6	5	1995	1996	691	409	691	807	단독주택	보증증서	정상판매	More
4	5	153900	7892	6	5	1993	1993	1199	530	1199	0	기타	신규건물	압류(공매도)	More

Next steps: [View recommended plots](#)

▼ 2.3 자료구조 살펴보기

[5] mr_df.shape
(121, 14)

[6] mr_df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121 entries, 0 to 120
```

✓ 0초 오후 9:23에 완료됨

3. 기술통계

✓ 0초 [8] # 그룹별 기술통계
mr_df.describe().round(3).T

	count	mean	std	min	25%	50%	75%	max
id	121.0	61.000	35.074	1.0	31.0	61.0	91.0	121.0
가격	121.0	160050.653	24553.521	97000.0	144000.0	163000.0	175000.0	228000.0
연면적	121.0	8829.157	3040.173	1533.0	7200.0	9247.0	10800.0	16635.0
품질	121.0	5.901	0.723	4.0	5.0	6.0	6.0	8.0
상태	121.0	5.975	1.084	3.0	5.0	6.0	7.0	8.0
건축년도	121.0	1963.603	26.390	1890.0	1957.0	1968.0	1978.0	2009.0
리모델링년도	121.0	1982.570	18.394	1950.0	1968.0	1988.0	2000.0	2009.0
지하면적	121.0	967.207	315.738	0.0	731.0	912.0	1196.0	1844.0
차고면적	121.0	445.248	159.853	0.0	336.0	453.0	530.0	923.0
면적_1층	121.0	1119.347	317.726	483.0	848.0	1116.0	1350.0	2020.0
면적_2층	121.0	337.364	373.564	0.0	0.0	0.0	708.0	1101.0

✓ 0초 [9] # 범주형 변수
lecture_df.columns
categorical_features = ['주거유형', '판매유형', '판매조건']

for col in categorical_features:
 print("----", col, "----")
 results = mr_df[col].value_counts()
 print(results, "\n")

---- 주거유형 ----
단독주택 90
튜플렉스 17
기타 14
마을 주택 1
 ✓ 0초 오후 9:23에 완료됨

3. 기술통계

```
[8] 0초      번식_1층    121.0    1179.347    317.726    483.0    848.0    1116.0    1350.0    2020.0
      면적_2층    121.0    337.364    373.564    0.0      0.0      0.0    708.0    1101.0
```

```
[9] 0초
# 범주형 변수
# lecture_df.columns
categorical_features = ['주거유형', '판매유형', '판매조건']

for col in categorical_features:
    print("----", col, "----")
    results = mr_df[col].value_counts()
    print(results, "\n")

---- 주거유형 ----
단독주택    90
튜플렉스     17
기타        14
Name: 주거유형, dtype: int64

---- 판매유형 ----
보증증서    99
신규건물    22
Name: 판매유형, dtype: int64

---- 판매조건 ----
절상판매    82
임류(공매도) 39
Name: 판매조건, dtype: int64
```

4. Regression(예측)

- <https://www.statsmodels.org/stable/examples/notebooks/generated/ols.html>
- 수치형 + 범주형
- dmatrix 사용

```
[10] 0초
# 기본
formula = "가격 ~ 연면적 + 품질 + 상태 + 건축년도 + 리모델링년도 + 지하면적 +
          #           + 차고면적 + 면적_1층 + 면적_2층 +
          #           + C(주거유형) + C(판매유형) + C(판매조건)"
```

✓ 0초 오후 9:23에 완료됨



4.Regression(예측)

LGE Internal Use Only

▼ 4.Regression(예측)

- <https://www.statsmodels.org/stable/examples/notebooks/generated/ols.html>
- 수치형 + 범주형
- dmatrix 사용

[10] # 기본
#formula = "가격 ~ 연면적 + 품질 + 상태 + 건축년도 + 리모델링년도 + 지하연적 #
+ 차고면적 + 면적_1층 + 면적_2층 #
+ C(주거유형) + C(판매유형) + C(판매조건)"

[11] # 코드 이용
columns = ['연면적', '품질', '상태', '건축년도', '리모델링년도', '지하연적',
'차고면적', '면적_1층', '면적_2층', 'C(주거유형)', 'C(판매유형)', 'C(판매조건)']

formula = "가격 ~ " + " + ".join(columns)
formula

'가격 ~ 연면적 + 품질 + 상태 + 건축년도 + 리모델링년도 + 지하연적 + 차고면적 + 면적_1층 + 면적_2층 + C(주거유형) + C(판매유형) + C(판매조건)'

[12] # dmatrix 이용
from patsy import dmatrices

y, X = dmatrices(formula,
 data = mr_df,
 return_type = 'dataframe')

[13] X

	Intercept	C(주거유형)[T. 단독주택]	C(주거유형)[T. 튜플렉스]	C(판매유형)[T. 신규건물]	C(판매조건)[T. 정상판매]	연면적	품 질	상 태	건축년 도	리모델링 년도	지하연 적	차고면 적	면적_1 층	면적_2 층
0	1.0	1.0	0.0	0.0	1.0	7388.0	5.0	6.0	1959.0	2002.0	1063.0	624.0	1327.0	0.0
1	1.0	0.0	0.0	1.0	0.0	4435.0	6.0	5.0	2003.0	2003.0	848.0	420.0	848.0	0.0

✓ 0초 오후 9:23에 완료됨

4.Regression(예측)

LGE Internal Use Only

```
[12] y, X = dmatrices(formula,
                      data = mr_df,
                      return_type = 'dataframe')

[13] X
```

	Intercept	C(주거유형)[1. 단독주택]	C(주거유형)[1. 투플렉스]	C(판매유형)[1. 신규건물]	C(판매조건)[1. 정상판매]	연면적	품질	상태	건축년도	리모델링년도	지하면적	차고면적	면적_1총	면적_2총
0	1.0	1.0	0.0	0.0	1.0	7388.0	5.0	6.0	1959.0	2002.0	1063.0	624.0	1327.0	0.0
1	1.0	0.0	0.0	1.0	0.0	4435.0	6.0	5.0	2003.0	2003.0	848.0	420.0	848.0	0.0
2	1.0	1.0	0.0	0.0	1.0	8800.0	6.0	6.0	1964.0	1964.0	1251.0	461.0	1251.0	0.0
3	1.0	1.0	0.0	0.0	1.0	13031.0	6.0	5.0	1995.0	1996.0	691.0	409.0	691.0	807.0
4	1.0	0.0	0.0	1.0	0.0	7892.0	6.0	5.0	1993.0	1993.0	1199.0	530.0	1199.0	0.0
...
116	1.0	0.0	0.0	1.0	0.0	1533.0	4.0	6.0	1970.0	2008.0	546.0	0.0	798.0	546.0
117	1.0	0.0	1.0	1.0	0.0	7200.0	5.0	8.0	1972.0	2003.0	768.0	396.0	768.0	0.0
118	1.0	0.0	1.0	1.0	0.0	7200.0	6.0	6.0	1910.0	1998.0	1214.0	506.0	1260.0	1031.0
119	1.0	0.0	1.0	1.0	0.0	6300.0	6.0	6.0	1914.0	2001.0	742.0	0.0	742.0	742.0
120	1.0	0.0	1.0	1.0	0.0	6300.0	6.0	6.0	1914.0	2001.0	742.0	0.0	742.0	742.0

121 rows × 14 columns

Next steps: [View recommended plots](#)

```
[14] model = sm.OLS(y, X) # 모델 생성
      result = model.fit() # 모델 실행

[15] print(result.summary())
```

OLS Regression Results

=====

Dep. Variable:	가격	R-squared:	0.829
Model:	OLS	Adj. R-squared:	0.808
Method:	Least Squares	F-statistic:	39.88
Date:	Sat, 02 Mar 2024	Prob (F-statistic):	5.29e-35

✓ 0초 오후 9:23에 완료됨

4.Regression(예측)

LGE Internal Use Only

```
✓ 0초 [15] print(result.summary())

          OLS Regression Results
=====
Dep. Variable:      가격  R-squared:       0.829
Model:              OLS  Adj. R-squared:   0.808
Method:             Least Squares  F-statistic:    39.88
Date:      Sat, 02 Mar 2024  Prob (F-statistic):  5.29e-35
Time:      12:23:07  Log-Likelihood:     -1287.5
No. Observations:    121  AIC:            2603.
Df Residuals:        107  BIC:            2642.
Df Model:           13
Covariance Type:   nonrobust
=====
      coef  std err      t      P>|t|      [0.025      0.975]
Intercept      -1.19e+05  1.46e+05   -0.815     0.417  -4.09e+05   1.7e+05
C(주거유형)[T.단독주택]  4867.2659  7706.807    0.632     0.529  -1.04e+04  2.01e+04
C(주거유형)[T.튜플렉스]  2678.4172  6797.977    0.394     0.694  -1.08e+04  1.62e+04
C(판매유형)[T.신규건물]  4008.5807  5483.337    0.731     0.466  -6861.495  1.49e+04
C(판매조건)[T.정상판매] -2264.6693  4187.665   -0.541     0.590  -1.06e+04  6036.889
연면적         6.8877   0.657    10.481    0.000      5.585    8.190
품질          3296.7099  1598.331    2.063     0.042    128.204  6465.215
상태          -1409.3099  1335.785   -1.055     0.294  -4057.349  1238.729
건축년도        43.6497   71.316    0.612     0.542    -97.727  185.026
리모델링년도    58.5679   70.365    0.832     0.407    -80.923  198.058
지하면적        6.5719   4.561    1.441     0.153    -2.470  15.614
차고면적        -6.9983   7.325   -0.955     0.342    -21.519  7.522
면적_1층       -1.4080   5.848   -0.241     0.810    -13.001  10.185
면적_2층        1.6977   4.604    0.369     0.713    -7.429  10.824
=====
Omnibus:            141.286  Durbin-Watson:      1.639
Prob(Omnibus):      0.000  Jarque-Bera (JB):  4779.598
Skew:                3.952  Prob(JB):            0.00
Kurtosis:             32.758  Cond. No.        1.47e+06
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.47e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

▼ 5.가정검정

- <https://ethanweed.github.io/pythonbook/05.04-regression.html#regressionnormality>

✓ 0초 오후 9:23에 완료됨



5.가정검정

5.가정검정

- <https://ethanweed.github.io/pythonbook/05.04-regression.html#regressionnormality>
- 잔차의 등분산성: Breusch-Pagan
- 잔차의 정규성: Jarque-Bera, Omnibus(D'Angostino's test)
- 독립성(자기상관): Durbin-Watson
- 다중공선성(VIF): Cond. No

5.1 기본 검정

- 잔차의 정규성: Jarque-Bera, Omnibus(D'Angostino's test)
- 독립성(자기상관): Durbin-Watson
- 다중공선성(VIF): Cond. No

[16] `print(result.summary())`

OLS Regression Results									
Dep. Variable:	가격	R-squared:	0.829						
Model:	OLS	Adj. R-squared:	0.808						
Method:	Least Squares	F-statistic:	39.88						
Date:	Sat, 02 Mar 2024	Prob (F-statistic):	5.29e-35						
Time:	12:23:07	Log-Likelihood:	-1287.5						
No. Observations:	121	AIC:	2603.						
Df Residuals:	107	BIC:	2642.						
Df Model:	13								
Covariance Type:	nonrobust								
	coef	std err	t	P> t	[0.025	0.975]			
Intercept	-1.19e+05	1.46e+05	-0.815	0.417	-4.09e+05	1.7e+05			
CC(주거유형)[T.단독주택]	4867.2659	7706.807	0.632	0.529	-1.04e+04	2.01e+04			
CC(주거유형)[T.튜플렉스]	2678.4172	6797.977	0.394	0.694	-1.08e+04	1.62e+04			
CC(판매유형)[T.신규건물]	4008.5807	5483.337	0.731	0.466	-6861.495	1.49e+04			
CC(판매조건)[T.정상판매]	-2264.6693	4187.665	-0.541	0.590	-1.06e+04	6036.889			
연면적	6.8877	0.657	10.481	0.000	5.585	8.190			
품질	3296.7099	1598.331	2.063	0.042	128.204	6465.215			
실태	-1409.3099	1335.785	-1.055	0.294	-4057.349	1238.729			
건축년도	43.6497	71.316	0.612	0.542	-97.727	185.026			
기타델린년도	58.5679	70.365	0.832	0.407	-80.923	198.058			

✓ 0초 오후 9:23에 완료됨

5. 가정검정

```
✓ 0초 [16] print(result.summary())
      OLS Regression Results
-----
Dep. Variable:                 가격
Model:                          OLS
Method:                         Least Squares
Date: Sat, 02 Mar 2024
Time: 12:23:07
No. Observations:             121
Df Residuals:                  107
Df Model:                      13
Covariance Type:               nonrobust
-----
            coef    std err          t      P>|t|      [0.025      0.975]
Intercept     -1.19e+05   1.46e+05     -0.815     0.417    -4.09e+05    1.7e+05
C(주거유형)[T. 단독주택]  4867.2659   7706.807      0.632     0.529    -1.04e+04   2.01e+04
C(주거유형)[T. 템플렉스]  2578.4172   6797.977      0.394     0.694    -1.08e+04   1.62e+04
C(판매유형)[T. 신규건물]  4008.5807   5483.337      0.731     0.466    -6861.495   1.49e+04
C(판매조건)[T. 정상판매] -2264.6693   4187.665     -0.541     0.590    -1.06e+04   6036.889
연면적           6.8877    0.657     10.481     0.000      5.585     8.190
품질            3296.7099   1598.331      2.063     0.042    128.204    6465.215
실태          -1409.3099   1335.785     -1.055     0.294    -4057.349   1238.729
건축년도        43.6497    71.316      0.612     0.542    -97.727    185.026
리모델링년도    58.5679    70.365      0.832     0.407    -80.923    198.058
지하면적         6.5719    4.561      1.441     0.153     -2.470    15.614
차고면적        -6.9983    7.325     -0.955     0.342    -21.519     7.522
면적_1층       -1.4080    5.848     -0.241     0.810    -13.001    10.185
면적_2층        1.6977    4.604      0.369     0.713     -7.429    10.824
-----
Omnibus:                141.286 Durbin-Watson:           1.639
Prob(Omnibus):           0.000 Jarque-Bera (JB):      4779.598
Skew:                   3.952 Prob(JB):                  0.00
Kurtosis:                32.758 Cond. No.:           1.47e+06
-----
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.47e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

▼ 5.2 잔차의 등분산 검정

- 잔차의 등분산성 테스트: Breush-Pagan 테스트:

✓ 0초 오후 9:23에 완료됨

5.가정검정

▼ 5.2 잔차의 등분산 검정

- 잔차의 등분산성 테스트: Breush-Pagan 테스트:

```
✓ [17] # 잔차의 등분산성 테스트: Breush-Pagan 테스트:
import statsmodels.stats.api as sms
from statsmodels.compat import lzip

name = ["Lagrange multiplier statistic", "p-value", "f-value", "f p-value"]
test = sms.het_breushpagan(result.resid, result.model.exog)
lzip(name, test)

[('Lagrange multiplier statistic', 26.816819635694316),
 ('p-value', 0.013179525935949678),
 ('f-value', 2.343550653851272),
 ('f p-value', 0.00864402200002526)]
```

```
✓ [18] # 잔차 플롯

# 표준화 잔차 생성
influence = result.get_influence()
res_standard = influence.resid_studentized_internal

# 예측값 생성
pred = result.predict(X)

# 데이터 프레임으로 생성
regplot_df = pd.DataFrame({'pred': pred, 'res_standard': res_standard})
regplot_df
```

	pred	res_standard
0	145982.124889	0.467629
1	134423.133237	-0.287447
2	159480.148533	0.049094
3	192100.694563	-0.450222

✓ 0초 오후 9:23에 완료됨

5.가정검정



5.가정검정

LGE Internal Use Only

▼ 5.3 잔차의 정규성 검정

```
[20] # shapiro test  
pg.normality(result.resid)
```

	pval	normal
0	0.726995	1.054879e-13
		False

```
[21] ## QQ plot  
plt.rc("figure", figsize=(8, 5))  
sm.qqplot(res_standard, line = 's')  
sns.despine()
```

Sample Quantiles

Theoretical Quantiles

✓ 0초 오후 9:23에 완료됨

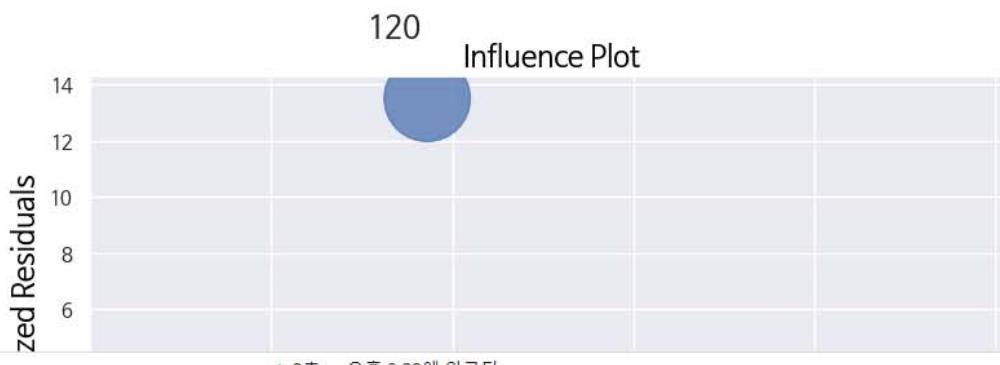
5. 가정검정

▼ 5.4 이상치 제거

```
[22] # 표준화 잔차를 이용한 이상치 확인  
stud_res = result.outlier_test()  
stud_res.sort_values(by = "student_resid", ascending = False).head(5)
```

student_resid	unadj_p	bonf(p)
120	13.504098	8.913759e-25
100	1.782336	7.755752e-02
106	1.590713	1.146522e-01
112	1.400458	1.642965e-01
76	1.312467	1.921972e-01

```
[23] ## cooks_distance를 이용한 이상치 확인  
## https://www.statsmodels.org/dev/examples/notebooks/generated/regression\_plots.html  
  
fig = sm.graphics.influence_plot(result, criterion="cooks")  
plt.rc("figure", figsize=(8, 5))  
plt.rc("font", size = 7)  
fig.tight_layout(pad = 1.0)
```



✓ 0초 오후 9:23에 완료됨

5. 가정검정

```

✓ 0초 [22] 112 1.400458 1.642965e-01 1.000000e+00
    76 1.312467 1.921972e-01 1.000000e+00

✓ 1초 ⏴ ## cooks_distance를 이용한 이상치 확인
## https://www.statsmodels.org/dev/examples/notebooks/generated/regression\_plots.html

fig = sm.graphics.influence_plot(result, criterion="cooks")
plt.rc("figure", figsize=(8, 5))
plt.rc("font", size = 7)
fig.tight_layout(pad = 1.0)

➡ 120
Influence Plot
Studentized Residuals
Leverage

```

Studentized Residuals

Leverage

120

Influence Plot

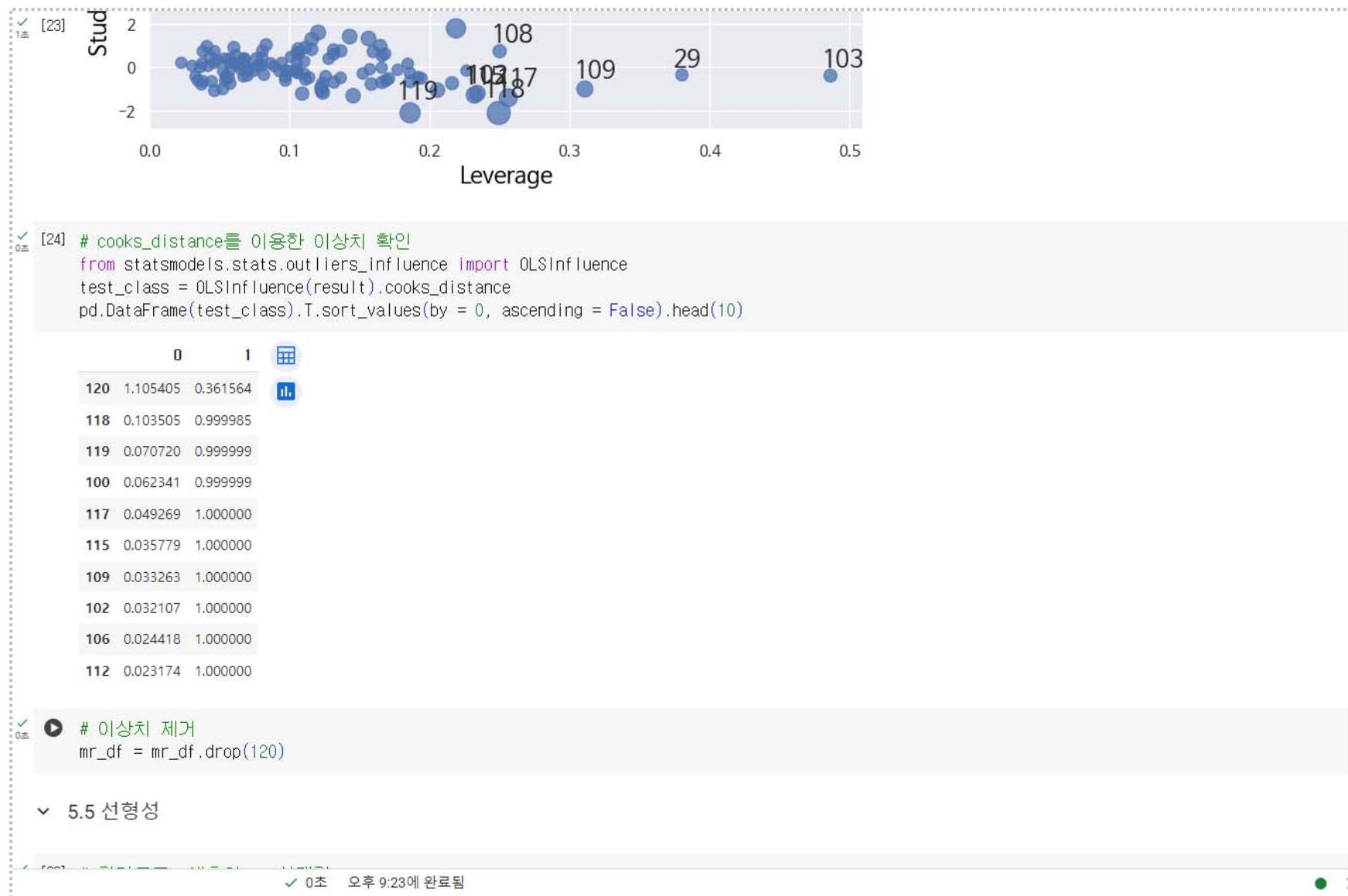
Index	Studentized Residuals	Leverage
112	1.400458	0.1642965
76	1.312467	0.1921972
108	2.0	0.25
109	1.0	0.32
29	-0.8	0.38
103	-0.5	0.48
104	1.5	0.28
117	1.0	0.28
118	-1.5	0.28
119	-1.8	0.22

```

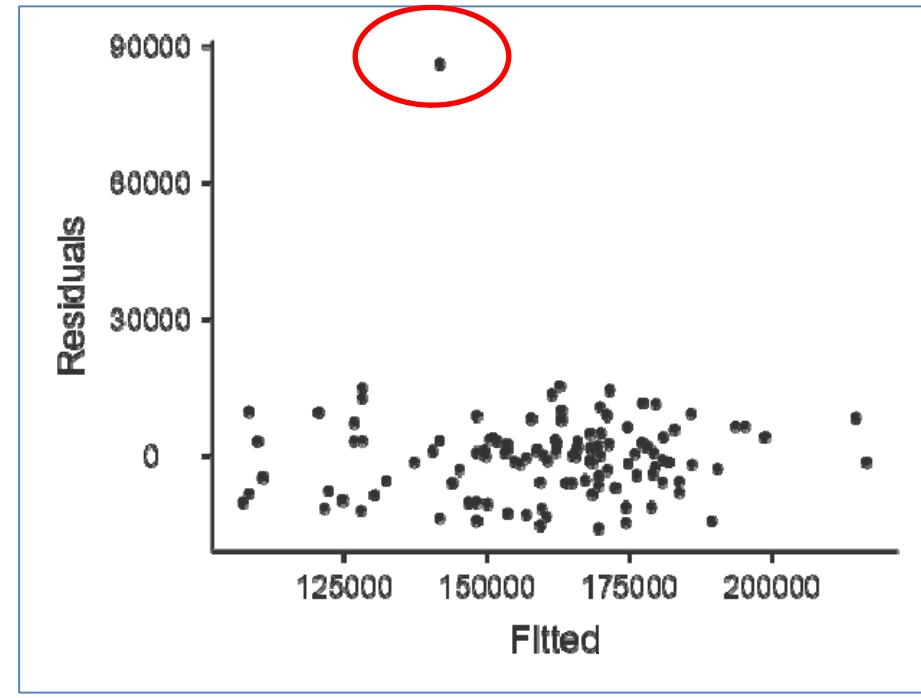
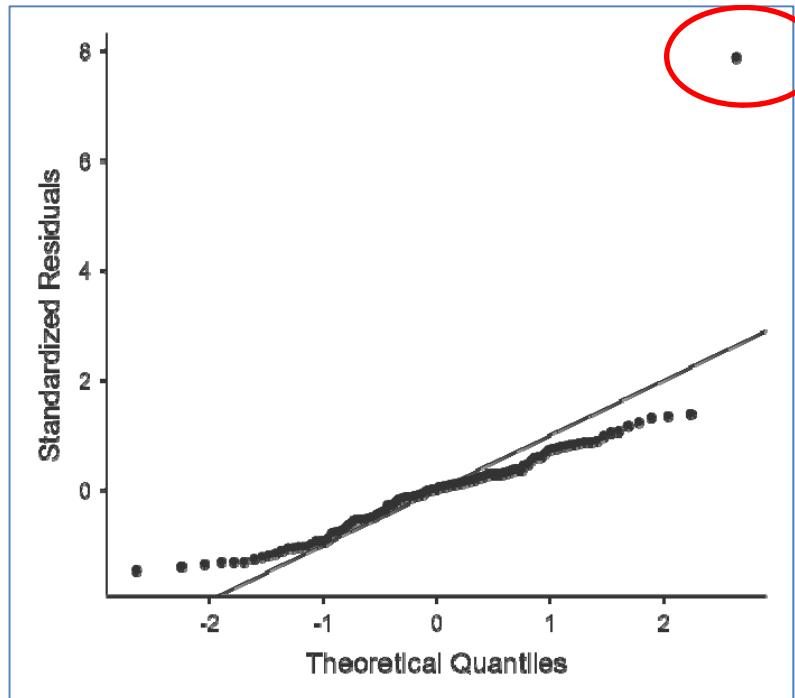
✓ 0초 [24] # cooks_distance를 이용한 이상치 확인
from statsmodels.stats.outliers_influence import OLSInfluence
✓ 0초 오후 9:23에 완료됨

```

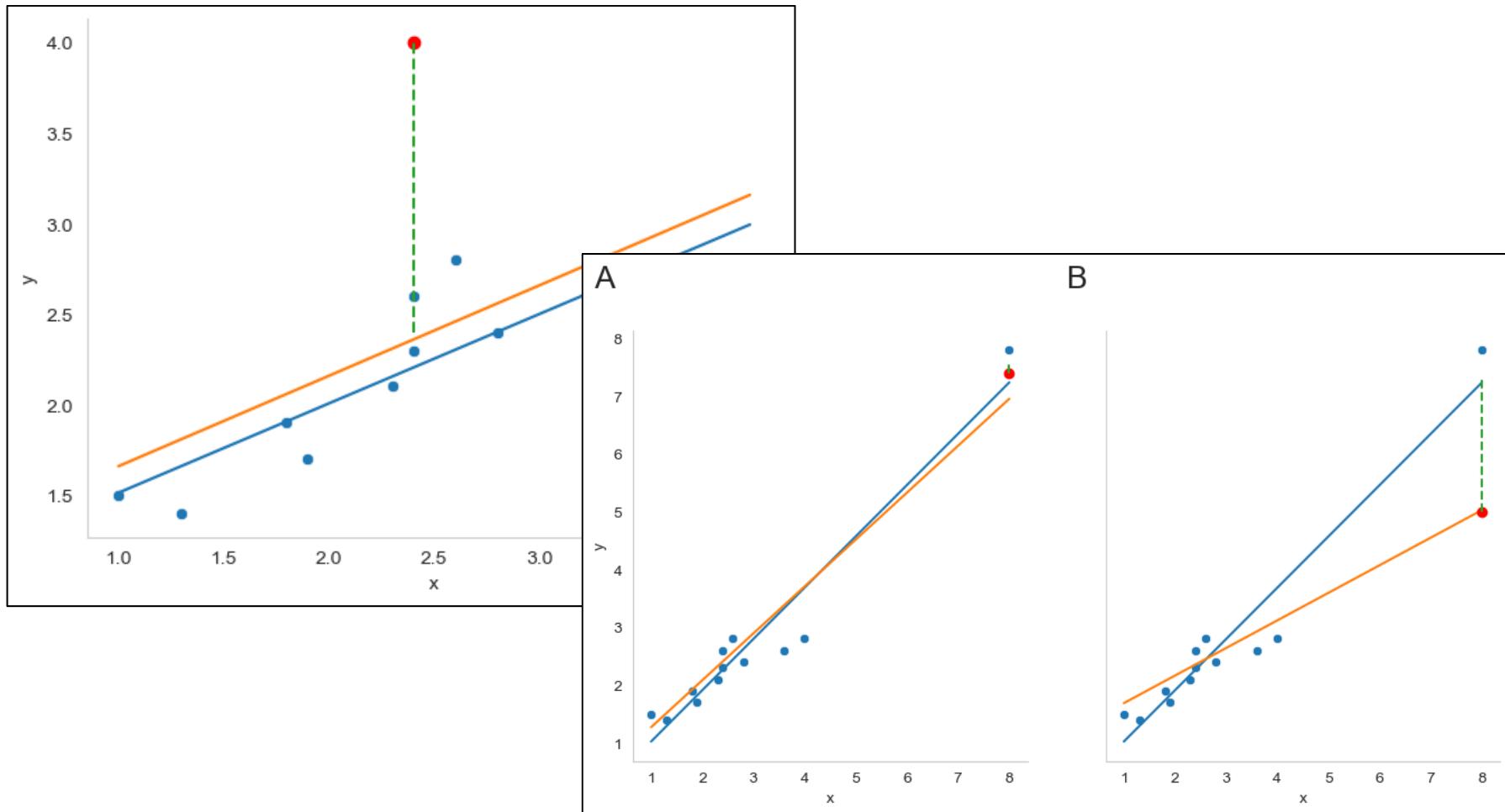
5. 가정검정



이상치 제거



이상치에 따른 회귀선 변화



4.Regression(예측)

LGE Internal Use Only

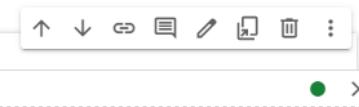
이상치 제거하고 회귀분석 다시 실행

```
[40] model = sm.OLS(y, X) # 모델 생성
    result = model.fit() # 모델 실행

[41] print(result.summary())

OLS Regression Results
=====
Dep. Variable:      가격      R-squared:       0.933
Model:                 OLS      Adj. R-squared:    0.925
Method:              Least Squares      F-statistic:     113.2
Date:        Sat, 02 Mar 2024      Prob (F-statistic): 7.19e-56
Time:          12:30:06      Log-Likelihood:   -1217.3
No. Observations:      120      AIC:             2463.
Df Residuals:         106      BIC:             2502.
Df Model:            13
Covariance Type:    nonrobust
=====
      coef    std err        t      P>|t|      [0.025      0.975]
Intercept   -1.1e+05   8.9e+04   -1.237     0.219   -2.86e+05   6.64e+04
C(주거유형)[T.단독주택]  -4363.9246   4744.105   -0.920     0.360   -1.38e+04   5041.726
C(주거유형)[T._duplex]  -7781.0121   4212.797   -1.847     0.068   -1.61e+04   571.267
C(판매유형)[T.신규건물]  -4911.8793   3404.865   -1.443     0.152   -1.17e+04   1838.597
C(판매조건)[T.정상판매]  -2261.4521   2550.914   -0.887     0.377   -7318.888   2795.984
연면적      6.7375    0.400    16.824     0.000      5.944     7.531
품질      2400.1457   975.884    2.459     0.016     465.361   4334.930
상태      -334.0592   817.579   -0.409     0.684   -1954.990   1286.872
건축년도      90.7995   43.582    2.083     0.040      4.393    177.206
리모델링년도      7.4513   43.030    0.173     0.863    -77.859    92.762
지하면적      6.8445    2.778    2.463     0.015      1.336    12.353
차고면적      4.2832    4.539    0.944     0.348    -4.717    13.283
면적_1층      1.2065    3.568    0.338     0.736    -5.867     8.280
면적_2층      4.2085    2.811    1.497     0.137    -1.364     9.781
=====
Omnibus:            3.419  Durbin-Watson:        2.100
Prob(Omnibus):      0.181  Jarque-Bera (JB):    3.329
Skew:                0.355  Prob(JB):           0.189
Kurtosis:            2.600  Cond. No.        1.47e+06
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.47e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```



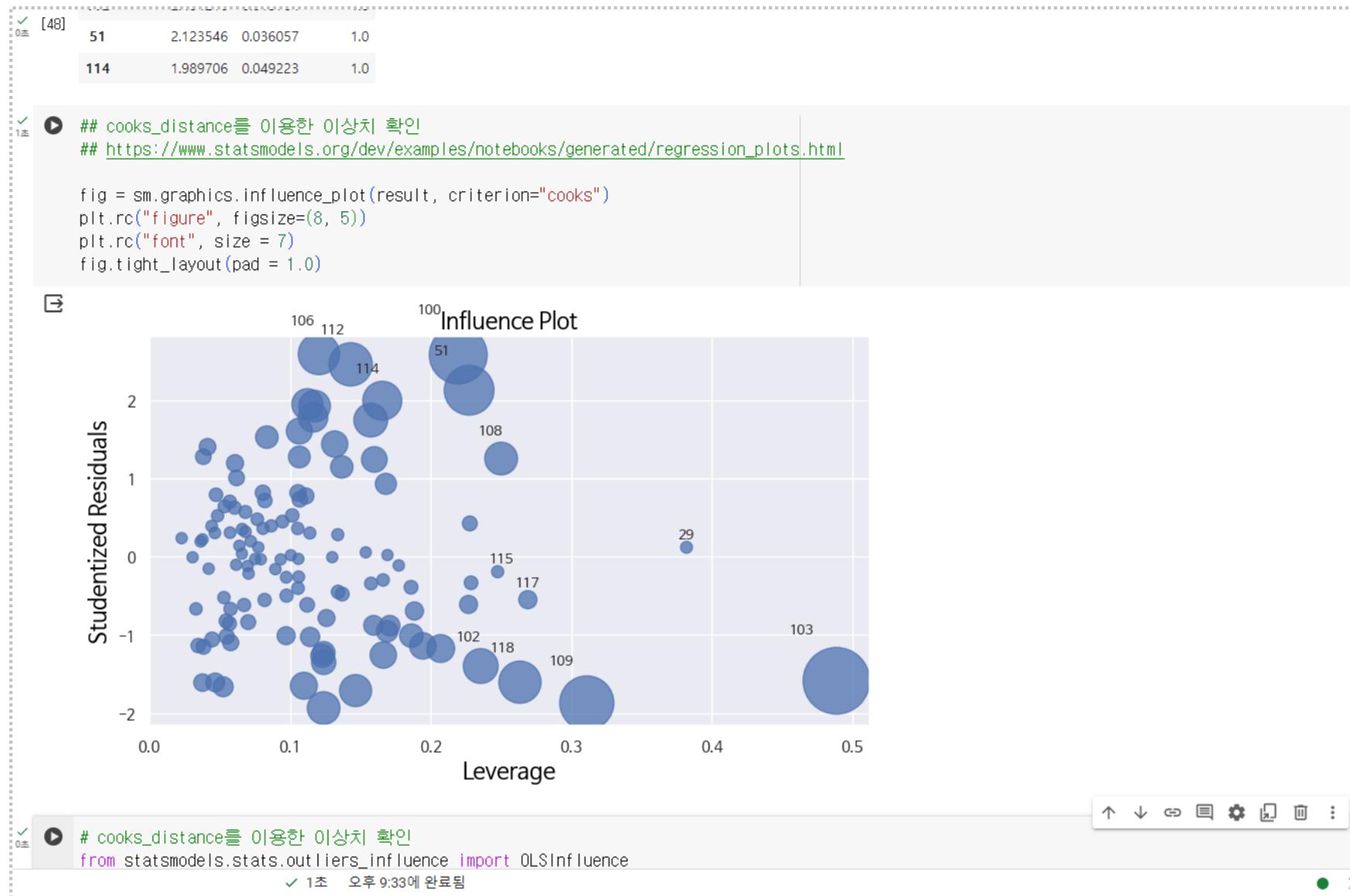
5.가정검정



5.가정검정



5. 가정검정



5.가정검정

```
[25] # 이상치 제거  
mr_df = mr_df.drop(120)
```

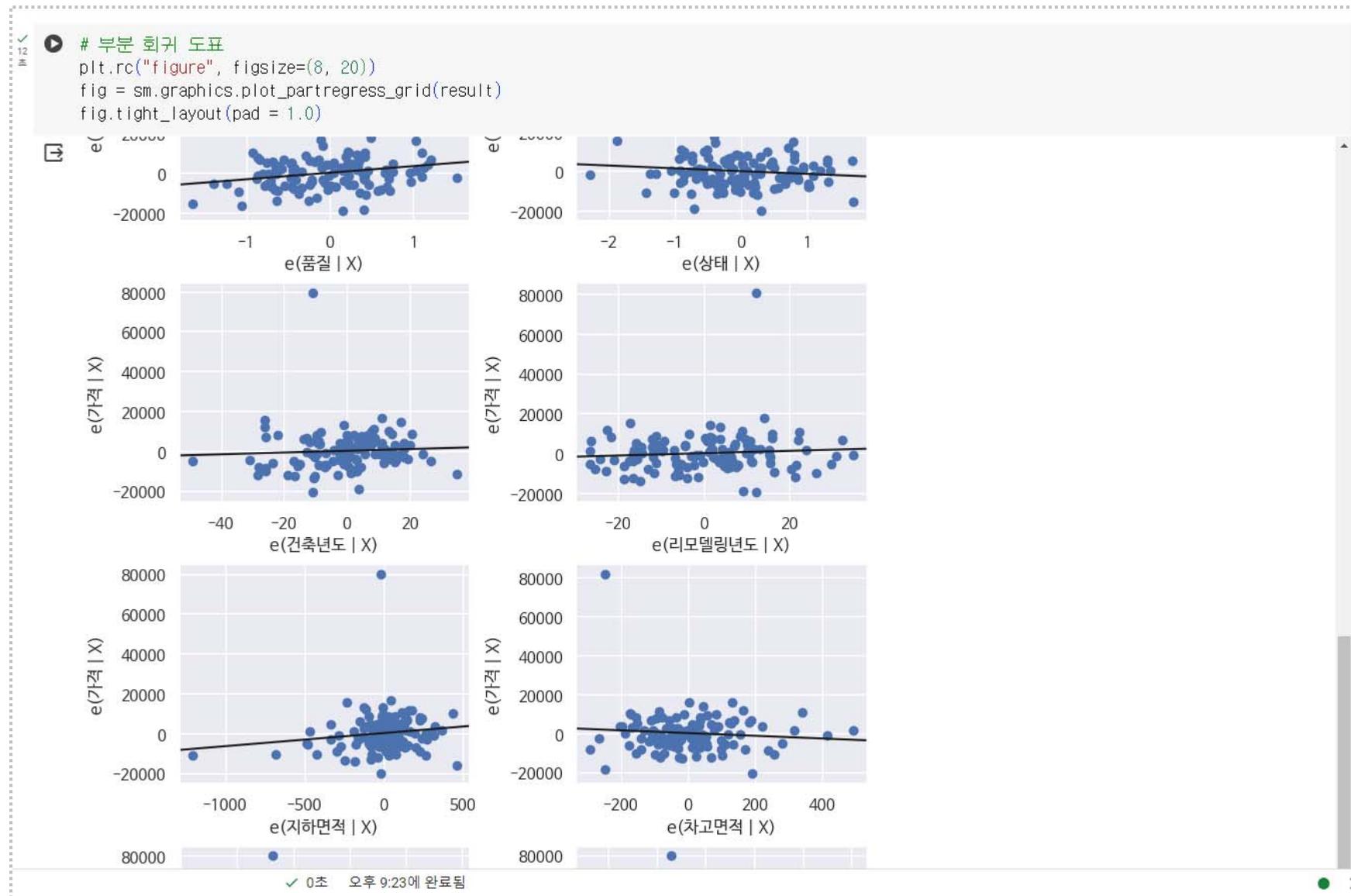
▼ 5.5 선형성

```
[26] # 회귀도표: 예측치 vs 실제값  
sns.scatterplot(x = regplot_df["pred"], y = mr_df["가격"])  
<Axes: xlabel='pred', ylabel='가격'>
```

```
[27] # 부분 회귀 도표  
plt.rc("figure", figsize=(8, 20))  
fig = sm.graphics.plot_partregress_grid(result)  
fig.tight_layout(pad = 1.0)
```

5. 가정검정

LGE Internal Use Only



5. 가정검정

▼ 5.6 다중 공선성

- VIF 10이상 삭제

```
[28] from statsmodels.stats.outliers_influence import variance_inflation_factor

vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.values.shape[1])]
vif["features"] = X.columns
print(vif.round(1))

    VIF Factor      features
0     22312.0      Intercept
1      11.8  C(주거유형)[T.단독주택]
2       5.8  C(주거유형)[T.튜플렉스]
3       4.7  C(판매유형)[T.신규건물]
4       4.0  C(판매조건)[T.점상판매]
5       4.1      연면적
6       1.4      품질
7       2.2      상태
8       3.7     건축년도
9       1.7  리모델링년도
10      2.2      지하면적
11      1.4      차고면적
12      3.6      면적_1층
13      3.1      면적_2층
```

▼ 부록1 formula 사용

- https://www.statsmodels.org/devel/example_formulas.html
- 수치형 + 범주형
- smf.ols 이용
- 잔차검증시 문제점 있음

```
[29] # 코드 이용
columns = ['연면적', '품질', '상태', '건축년도', '리모델링년도', '지하면적', '차고면적',
           '면적_1층', '면적_2층', 'C(주거유형)', 'C(판매유형)', 'C(판매조건)']
```

✓ 0초 오후 9:23에 완료됨

부록1 formula 사용

LGE Internal Use Only

부록1 formula 사용

- https://www.statsmodels.org/devel/example_formulas.html
- 수치형 + 범주형
- smf.ols 이용
- 잔차검증시 문제점 있음

✓ [29] # 코드 이용

```
columns = ['연면적', '품질', '상태', '건축년도', '리모델링년도', '지하면적', '차고면적',
           '면적_1층', '면적_2층', 'C(주거유형)', 'C(판매유형)', 'C(판매조건)']

formula = "가격 ~ " + " + ".join(columns)
formula

'가격 ~ 연면적 + 품질 + 상태 + 건축년도 + 리모델링년도 + 지하면적 + 차고면적 + 면적_1층 + 면적_2층 + C(주거유형) + C(판매유형) + C(판매조건)'
```

✓ [30] import statsmodels.formula.api as smf

```
model = smf.ols(formula = formula, data = mr_df)
result = model.fit()
```

✓ [31] print(result.summary())

OLS Regression Results

```
=====
Dep. Variable:      가격    R-squared:       0.933
Model:                 OLS    Adj. R-squared:   0.925
Method:              Least Squares    F-statistic:     113.2
Date:            Sat, 02 Mar 2024    Prob (F-statistic): 7.19e-56
Time:                12:23:23    Log-Likelihood:   -1217.3
No. Observations:      120    AIC:             2463.
Df Residuals:          106    BIC:             2502.
Df Model:                  13
Covariance Type:    nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
Intercept	-1.1e+05	8.9e+04	-1.237	0.219	-2.86e+05	6.64e+04

✓ 0초 오후 9:33에 완료됨

부록2 변수로 입력

LGE Internal Use Only

▼ 부록2 변수로 입력

- 수치형 변수만 입력
- pd.get_dummies(mr_df)를 이용할 수 있지만 복잡함

[32] # 방법1 상수항 포함: sm.add_constant(X)

```
# X(대문자)와 y(소문자)로 분리
X = mr_df[['id', '가격', '연면적', '품질', '상태', '건축년도', '리모델링년도', '지하면적', '차고면적',
            '면적_1층', '면적_2층']]
y = mr_df['가격']
X = sm.add_constant(X) # 상수항 추가
model = sm.OLS(y, X) # 모델 생성
result = model.fit() # 모델 실행
```

[33] print(result.summary())

Dep. Variable:	가격	R-squared:	1.000			
Model:	OLS	Adj. R-squared:	1.000			
Method:	Least Squares	F-statistic:	1.178e+29			
Date:	Sat, 02 Mar 2024	Prob (F-statistic):	0.00			
Time:	12:23:23	Log-Likelihood:	2500.0			
No. Observations:	120	AIC:	-4976.			
Df Residuals:	108	BIC:	-4942.			
Df Model:	11					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	1.557e-09	2.73e-09	0.571	0.569	-3.85e-09	6.96e-09
id	5.684e-14	6.37e-13	0.089	0.929	-1.21e-12	1.32e-12
가격	1.0000	3.33e-15	3e+14	0.000	1.000	1.000
연면적	-7.105e-15	2.4e-14	-0.296	0.768	-5.48e-14	4.05e-14
품질	-5.002e-12	3.45e-11	-0.145	0.885	-7.33e-11	6.33e-11
상태	-3.638e-12	2.7e-11	-0.135	0.893	-5.71e-11	4.99e-11
건축년도	1.563e-13	1.35e-12	0.116	0.908	-2.51e-12	2.82e-12
리모델링년도	-9.663e-13	1.39e-12	-0.695	0.489	-3.72e-12	1.79e-12
지하면적	-4.263e-14	9.69e-14	-0.440	0.661	-2.35e-13	1.49e-13
차고면적	5.507e-14	1.50e-12	0.340	0.790	2.50e-13	2.50e-13

부록3 pingouin

LGE Internal Use Only

```
▼ 부록3 pingouin
  • https://pingouin-stats.org/build/html/generated/pingouin.linear\_regression.html#pingouin.linear\_regression
  • 분석이 간단하지만 가정검정 지원이 안됨

✓ [34] # dmatrix 이용
from patsy import dmatrices

y, X = dmatrices(formula,
                  data = mr_df,
                  return_type = 'dataframe')

✓ [35] y = mr_df['가격']
y

0    150750
1    131500
2    160000
3    187500
4    153900
...
115   144000
116   97000
117   133900
118   133900
119   128000
Name: 가격, Length: 120, dtype: int64

✓ [36] lm = pg.linear_regression(X, y)
lm.round(3)

      names      coef       se      T   pval     r2  adj_r2    CI [2.5%]    CI [97.5%]
0 Intercept -110012.777 88964.779 -1.237 0.219 0.933 0.925 -286394.104 66368.550
1 C(주거유형)[T.단독주택] -4363.925 4744.105 -0.920 0.360 0.933 0.925 -13769.575 5041.726
2 C(주거유형)[T.튜플렉스] -7781.012 4212.797 -1.847 0.068 0.933 0.925 -16133.292 571.267
3 C(판매유형)[T.신규건물] -4911.879 3404.865 -1.443 0.152 0.933 0.925 -11662.355 1838.597
```

부록3 pingouin

```
Name: '가격', Length: 120, dtype: int64
```

```
[36]: lm = pg.linear_regression(X, y)
lm.round(3)
```

	names	coef	se	T	pval	r2	adj_r2	CI [2.5%]	CI [97.5%]
0	Intercept	-110012.777	88964.779	-1.237	0.219	0.933	0.925	-286394.104	66368.550
1	C(주거유형)[T.단독주택]	-4363.925	4744.105	-0.920	0.360	0.933	0.925	-13769.575	5041.726
2	C(주거유형)[T.튜플렉스]	-7781.012	4212.797	-1.847	0.068	0.933	0.925	-16133.292	571.267
3	C(판매유형)[T.신규건물]	-4911.879	3404.865	-1.443	0.152	0.933	0.925	-11662.355	1838.597
4	C(판매조건)[T.정상판매]	-2261.452	2550.914	-0.887	0.377	0.933	0.925	-7318.888	2795.984
5	연면적	6.738	0.400	16.824	0.000	0.933	0.925	5.944	7.531
6	품질	2400.146	975.884	2.459	0.016	0.933	0.925	465.361	4334.930
7	상태	-334.059	817.579	-0.409	0.684	0.933	0.925	-1954.990	1286.872
8	건축년도	90.800	43.582	2.083	0.040	0.933	0.925	4.393	177.206
9	리모델링년도	7.451	43.030	0.173	0.863	0.933	0.925	-77.859	92.762
10	지하면적	6.845	2.778	2.463	0.015	0.933	0.925	1.336	12.353
11	차고면적	4.283	4.539	0.944	0.348	0.933	0.925	-4.717	13.283
12	면적_1층	1.206	3.568	0.338	0.736	0.933	0.925	-5.867	8.280
13	면적_2층	4.209	2.811	1.497	0.137	0.933	0.925	-1.364	9.781

```
[37]: pg.normality(lm.residuals_)
```

	pval	normal
0	0.979869	0.069245
		True

```
[37]
```

✓ 0초 오후 9:33에 완료됨

Regression(예측)

- ❖ 주택가격에 영향을 주는 요인을 분석한 결과, 연면적, 품질, 건축년도, 지하면적이 유의한 것으로 나타났다. ($F=113.163$, $p=.000$).
- ❖ 가격 예측 모델을 구하면 다음과 같다.

$$\text{가격} = -116,638 + 6.738\text{연면적} + 2,400\text{품질} + 90.8\text{건축년도} + 6.845\text{지하면적}$$

Predictor	Estimate	SE	t	p
Intercept	-116,638.154	87,244.185	-1.337	0.184
연면적	6.738	0.400	16.824	<.001
품질	2400.146	975.884	2.459	0.016
상태	-334.059	817.579	-0.409	0.684
건축년도	90.800	43.582	2.083	0.040
리모델링년도	7.451	43.030	0.173	0.863
지하면적	6.845	2.778	2.463	0.015
차고면적	4.283	4.539	0.944	0.348
면적_1층	1.206	3.568	0.338	0.736
면적_2층	4.209	2.811	1.497	0.137
주거유형:				
튜플렉스 – 단독주택	-3,417.087	3,005.353	-1.137	0.258
기타 – 단독주택	4,363.925	4,744.105	0.920	0.360
판매유형:				
신규건물 – 보증증서	-4,911.879	3,404.865	-1.443	0.152
판매조건:				
압류(공매도) – 정상판매	2,261.452	2,550.914	0.887	0.377

Regression(설명)

❖ 문제의 정의

- 온라인게임의 충성도에 영향을 주는 요인이 무엇인지를 연구하고자 한다.
- 영향을 주는 변수로는 도구, 보상, 정보, 디자인, 공동체, 몰입이 있다.
- 온라인게임 몰입에 영향을 주는 변수는 무엇이고, 어떤 변수가 온라인게임 몰입에 가장 큰 영향을 주는 변수인가?
- 13_1.MR(설명).csv

❖ 가설

- 귀무가설(H_0): 독립변수들은 종속변수에 영향을 주지 않는다.

$$H_0: \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$$

- 연구가설(H_1): 독립변수들 중 하나의 변수는 종속변수에 영향을 준다.

$$H_1: \text{not } H_0$$

Regression(설명)

LGE Internal Use Only

❖ 표준화 회귀계수 검정

Model Coefficients - 충성도

Predictor	Estimate	SE	t	p	Stand. Estimate
Intercept	-1.360	0.349	-3.894	< .001	
도구	0.020	0.031	0.647	0.518	0.018
보상	0.078	0.019	4.091	< .001	0.113
정보	0.110	0.027	4.097	< .001	0.116
디자인	0.157	0.016	9.799	< .001	0.289
공동체	0.136	0.019	7.277	< .001	0.245
몰입	0.319	0.027	11.669	< .001	0.400

13_1.Regression(설명)

LGE Internal Use Only

The screenshot shows a Jupyter Notebook interface with the following structure:

- Section 1:** 13_1.Regression(설명)
 - <https://www.statsmodels.org/stable/gettingstarted.html>
- Section 2:** 1.기본 package 설정
 - [] # 그래프에서 한글 폰트 인식하기
!sudo apt-get install -y fonts-nanum
!sudo fc-cache -fv
!rm ~/.cache/matplotlib -rf
 - [] !pip install pingouin
*** 런타임 다시 시작
- Section 3:** [1] # 1.기본
 - import numpy as np # numpy 패키지 가져오기
 - import matplotlib.pyplot as plt # 시각화 패키지 가져오기
 - import seaborn as sns # 시각화
- Section 4:** [2] # 기본세팅
 - # 테마 설정
sns.set_theme(style = "darkgrid")

The code cells contain Python and shell commands related to setting up a Jupyter notebook environment for regression analysis, specifically using the statsmodels library.

2.데이터 불러오기

LGE Internal Use Only

2.데이터 불러오기

2.1 데이터 프레임으로 저장

- 원본데이터(csv)를 dataframe 형태로 가져오기(pandas)

```
[3]: mr_df = pd.read_csv('https://raw.githubusercontent.com/echo-bigdata/statistics-python/main/13_1_MR(expl).csv', encoding='cp949')
mr_df.head()
```

no	성별	결혼	학력	연령	도구	보상	정보	디자인	공동체	몰입	충성도
0	1	2	3	18	2	7	6	12	12	8	6
1	2	2	2	22	2	8	3	11	6	4	4
2	3	2	2	26	2	8	7	14	18	12	9
3	4	1	2	34	2	8	7	18	15	4	7
4	5	1	2	28	2	9	8	16	23	14	11

Next steps: View recommended plots

2.2 범주형 변수 처리

- 가변수 처리시 문자로 처리를 해야 변수명 구분이 쉬움

```
[4]: mr_df['성별'].replace({1:'남자', 2:'여자'}, inplace=True)
mr_df['결혼'].replace({1:'결혼', 2:'미혼'}, inplace=True)
mr_df['학력'].replace({1:'초중고생', 2:'고졸', 3:'대학생', 4:'대졸'}, inplace=True)

mr_df['성별'] = mr_df['성별'].astype('category')
mr_df['결혼'] = mr_df['결혼'].astype('category')
mr_df['학력'] = mr_df['학력'].astype('category')
mr_df
```

✓ 0초 오후 9:38에 완료됨

2.데이터 불러오기

LGE Internal Use Only

Next steps: [View recommended plots](#)

▼ 2.2 범주형 변수 처리

- 가변수 처리시 문자로 처리를 해야 변수명 구분이 쉬움

[4] mr_df['성별'].replace({1:'남자', 2:'여자'}, inplace=True)
mr_df['결혼'].replace({1:'결혼', 2:'미혼'}, inplace=True)
mr_df['학력'].replace({1:'초중고생', 2:'고졸', 3:'대학생', 4:'대졸'}, inplace=True)

mr_df['성별'] = mr_df['성별'].astype('category')
mr_df['결혼'] = mr_df['결혼'].astype('category')
mr_df['학력'] = mr_df['학력'].astype('category')

mr_df

no	성별	결혼	학력	연령	도구	보상	정보	디자인	공동체	몰입	충성도	
0	1	남자	미혼	대학생	18	2	7	6	12	12	8	6
1	2	여자	미혼	고졸	22	2	8	3	11	6	4	4
2	3	여자	미혼	고졸	26	2	8	7	14	18	12	9
3	4	남자	미혼	대학생	34	2	8	7	18	15	4	7
4	5	남자	미혼	대졸	28	2	9	8	16	23	14	11
...
371	372	남자	결혼	대학생	35	10	10	9	26	20	13	12
372	373	여자	미혼	대학생	24	10	10	13	28	29	19	15
373	374	여자	미혼	고졸	23	10	12	5	22	18	12	9
374	375	여자	미혼	대졸	19	10	15	7	14	16	15	10
375	376	남자	미혼	대졸	28	10	16	7	20	14	10	9

376 rows × 12 columns

Next steps: [View recommended plots](#)

▼ 2.3 자료구조 살펴보기

✓ 0초 오후 9:38에 완료됨

3. 기술통계

LGE Internal Use Only

```
✓ 0초 [8] # 그룹별 기술통계  
mr_df.describe().round(3).T
```

	count	mean	std	min	25%	50%	75%	max
no	376.0	188.500	108.686	1.0	94.75	188.5	282.25	376.0
연령	376.0	26.646	6.938	11.0	22.00	26.0	30.00	50.0
도구	376.0	6.231	1.534	2.0	5.00	6.0	8.00	10.0
보상	376.0	10.492	2.591	4.0	9.00	11.0	12.00	18.0
정보	376.0	8.915	1.874	3.0	8.00	9.0	10.00	15.0
디자인	376.0	18.936	3.274	8.0	17.00	19.0	21.00	28.0
공동체	376.0	19.375	3.218	6.0	18.00	19.0	21.00	29.0
몰입	376.0	12.689	2.227	4.0	12.00	13.0	14.00	20.0
충성도	376.0	10.223	1.781	3.0	9.00	10.0	11.00	15.0

```
✓ 0초 [9] # 범주형 변수  
# lecture_df.columns  
categorical_features = ['성별', '결혼', '학력']  
  
for col in categorical_features:  
    print("----", col, "----")  
    results = mr_df[col].value_counts()  
    print(results, "\n")  
  
---- 성별 ----  
여자    209  
남자    167  
Name: 성별, dtype: int64  
  
---- 결혼 ----  
미혼    275  
결혼    101
```

✓ 0초 오후 9:38에 완료됨

4.Regression(설명)

LGE Internal Use Only

▼ 4.Regression(설명)

- <https://www.statsmodels.org/stable/examples/notebooks/generated/ols.html>
- 수치형 + 범주형
- dmatrix 사용

▼ 4.1 Regression

0초 [10] # 코드 이용

```
columns = ['도구', '보상', '정보', '디자인', '공동체', '몰입', '연령',
           'C(성별)', 'C(결혼)', 'C(학력)']

formula = "충성도 ~ " + " + ".join(columns)

formula
```

'충성도 ~ 도구 + 보상 + 정보 + 디자인 + 공동체 + 몰입 + 연령 + C(성별) + C(결혼) + C(학력)'

0초 [11] # dmatrix 이용

```
from patsy import dmatrices

y, X = dmatrices(formula,
                  data = mr_df,
                  return_type = 'dataframe')
```

0초 [12] X.head()

	Intercept	C(성별)[T.여자]	C(결혼)[T.미혼]	C(학력)[T.대출]	C(학력)[T.대학생]	C(학력)[T.초중고생]	도구	보상	정보	디자인	공동체	몰입	연령	
0	1.0	0.0	1.0	0.0	1.0		0.0	2.0	7.0	6.0	12.0	12.0	8.0	18.0
1	1.0	1.0	1.0	0.0	0.0		0.0	2.0	8.0	3.0	11.0	6.0	4.0	22.0
2	1.0	1.0	1.0	0.0	0.0		0.0	2.0	8.0	7.0	14.0	18.0	12.0	26.0
3	1.0	0.0	1.0	1.0	0.0		0.0	2.0	8.0	7.0	18.0	15.0	4.0	34.0
4	1.0	0.0	1.0	1.0	0.0		0.0	2.0	9.0	8.0	16.0	23.0	14.0	28.0

#... ✓ 0초 오후 9:38에 완료됨

4.Regression(설명)

Next steps: [View recommended plots](#)

```

0초 [13] model = sm.OLS(y, X) # 모델 생성
      result = model.fit() # 모델 실행

0초 [14] print(result.summary())

```

OLS Regression Results

Dep. Variable:	총성도	R-squared:	0.775			
Model:	OLS	Adj. R-squared:	0.768			
Method:	Least Squares	F-statistic:	104.3			
Date:	Sat, 02 Mar 2024	Prob (F-statistic):	1.21e-109			
Time:	12:38:18	Log-Likelihood:	-469.48			
No. Observations:	376	AIC:	965.0			
Df Residuals:	363	BIC:	1016.			
Df Model:	12					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
Intercept	-1.0909	0.466	-2.342	0.020	-2.007	-0.175
CC(성별)[T.여자]	0.1463	0.091	1.600	0.110	-0.033	0.326
CC(결혼)[T.미혼]	-0.0571	0.132	-0.434	0.664	-0.316	0.202
CC(학력)[T.대출]	0.3300	0.129	2.554	0.011	0.076	0.584
CC(학력)[T.대학생]	0.1317	0.129	1.025	0.306	-0.121	0.384
CC(학력)[T.초중고생]	0.2207	0.155	1.425	0.155	-0.084	0.525
도구	0.0237	0.031	0.761	0.447	-0.038	0.085
보상	0.0728	0.019	3.835	0.000	0.035	0.110
정보	0.1111	0.027	4.168	0.000	0.059	0.164
디자인	0.1573	0.016	9.824	0.000	0.126	0.189
승동체	0.1347	0.019	7.211	0.000	0.098	0.171
몰입	0.3175	0.027	11.597	0.000	0.264	0.371
연령	-0.0161	0.008	-1.926	0.055	-0.033	0.000
Omnibus:	0.670	Durbin-Watson:	1.872			
Prob(Omnibus):	0.715	Jarque-Bera (JB):	0.776			
Skew:	-0.047	Prob(JB):	0.678			
Kurtosis:	2.798	Cond. No.	462.			

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

▼ 4.2 표준화 계수

✓ 0초 오후 9:38에 완료됨

4.Regression(설명)

▼ 4.2 표준화 계수

```

0초 [15] # 수치형 자료만 있을때
# model_std = sm.OLS(zscore(y), zscore(X)) # 모델 생성
# result_std = model_std.fit() # 모델 실행
# print(result_std.summary())

0초 [16] X.columns
Index(['Intercept', 'C(성별)[T.여자]', 'C(결혼)[T.미혼]', 'C(학력)[T.대출]',
       'C(학력)[T.대학생]', 'C(학력)[T.초중고생]', '도구', '보상', '정보',
       '디자인', '공동체', '몰입',
       '연경'],
      dtype='object')

0초 [17] X_cat = X[['C(성별)[T.여자]', 'C(결혼)[T.미혼]', 'C(학력)[T.대출]', 'C(학력)[T.대학생]', 'C(학력)[T.초중고생]']]
X_cat

```

	C(성별)[T.여자]	C(결혼)[T.미혼]	C(학력)[T.대출]	C(학력)[T.대학생]	C(학력)[T.초중고생]
0	0.0	1.0	0.0	1.0	0.0
1	1.0	1.0	0.0	0.0	0.0
2	1.0	1.0	0.0	0.0	0.0
3	0.0	1.0	1.0	0.0	0.0
4	0.0	1.0	1.0	0.0	0.0
...
371	0.0	0.0	0.0	1.0	0.0
372	1.0	1.0	0.0	1.0	0.0
373	1.0	1.0	0.0	0.0	0.0
374	1.0	1.0	1.0	0.0	0.0
375	0.0	1.0	1.0	0.0	0.0

376 rows × 5 columns

✓ 0초 오후 9:38에 완료됨

4.Regression(설명)

LGE Internal Use Only

```
[18] from scipy.stats.mstats import zscore  
  
X_std = X[['연령', '도구', '보상', '정보', '디자인', '공동체', '몰입']]  
X_std = zscore(X_std)  
X_std  
  
연령 도구 보상 정보 디자인 공동체 몰입  
0 -1.247846 -2.761391 -1.349413 -1.557321 -2.121169 -2.294612 -2.107816  
1 -0.670559 -2.761391 -0.962986 -3.160111 -2.426982 -4.161416 -3.905975  
2 -0.093272 -2.761391 -0.962986 -1.023058 -1.509544 -0.427809 -0.309656  
3 1.061302 -2.761391 -0.962986 -1.023058 -0.286293 -1.361211 -3.905975  
4 0.195372 -2.761391 -0.576558 -0.488794 -0.897918 1.127860 0.589423  
... ... ... ... ... ... ... ...  
371 1.205624 2.459391 -0.190131 0.045469 2.160209 0.194459 0.139883  
372 -0.381915 2.459391 -0.190131 2.182523 2.771835 2.994664 2.837123  
373 -0.526237 2.459391 0.582725 -2.091584 0.936958 -0.427809 -0.309656  
374 -1.103524 2.459391 1.742007 -1.023058 -1.509544 -1.050077 1.038963  
375 0.195372 2.459391 2.128435 -1.023058 0.325333 -1.672345 -1.208736  
376 rows × 7 columns  
  
Next steps:  View recommended plots  
  
[19] X_std = pd.concat([X_cat, X_std], axis=1)  
X_std  
  
C(성별)[T.여자] C(결혼)[T.미혼] C(학력)[T.대출] C(학력)[T.대학생] C(학력)[T.초중고생] 연령 도구 보상 정보 디자인 공동체 몰입  
0 0.0 1.0 0.0 1.0 0.0 -1.247846 -2.761391 -1.349413 -1.557321 -2.121169 -2.294612 -2.107816  
1 1.0 1.0 0.0 0.0 0.0 -0.670559 -2.761391 -0.962986 -3.160111 -2.426982 -4.161416 -3.905975  
2 1.0 1.0 0.0 0.0 0.0 -0.093272 -2.761391 -0.962986 -1.023058 -1.509544 -0.427809 -0.309656  
3 0.0 1.0 1.0 1.0 0.0 0.0 1.061302 -2.761391 -0.962986 -1.023058 -0.286293 -1.361211 -3.905975  
✓ 0초 오후 9:38에 완료됨
```

4.Regression(설명)

```

0초 [19] X_std = pd.concat([X_cat, X_std], axis=1)
X_std

      C(성별)[T.여자]  C(결혼)[T.미혼]  C(학력)[T.대졸]  C(학력)[T.대학생]  C(학력)[T.초중고생]  연령  도구  보상  정보  디자인  공동체  물입
0            0.0           1.0           0.0           1.0           0.0 -1.247846 -2.761391 -1.349413 -1.557321 -2.121169 -2.294612 -2.107816
1            1.0           1.0           0.0           0.0           0.0 -0.670559 -2.761391 -0.962986 -3.160111 -2.426982 -4.161416 -3.905975
2            1.0           1.0           0.0           0.0           0.0 -0.093272 -2.761391 -0.962986 -1.023058 -1.509544 -0.427809 -0.309656
3            0.0           1.0           1.0           0.0           0.0  1.061302 -2.761391 -0.962986 -1.023058 -0.286293 -1.361211 -3.905975
4            0.0           1.0           1.0           0.0           0.0  0.195372 -2.761391 -0.576558 -0.488794 -0.897918  1.127860  0.589423
...
371           0.0           0.0           0.0           1.0           0.0  1.205624  2.459391 -0.190131  0.045469  2.160209  0.194459  0.139883
372           1.0           1.0           0.0           1.0           0.0 -0.381915  2.459391 -0.190131  2.182523  2.771835  2.994664  2.837123
373           1.0           1.0           0.0           0.0           0.0 -0.526237  2.459391  0.582725 -2.091584  0.936958 -0.427809 -0.309656
374           1.0           1.0           1.0           0.0           0.0 -1.103524  2.459391  1.742007 -1.023058 -1.509544 -1.050077  1.038963
375           0.0           1.0           1.0           0.0           0.0  0.195372  2.459391  2.128435 -1.023058  0.325333 -1.672345 -1.208736
376 rows × 12 columns

Next steps:  View recommended plots

0초 [20] y_std = mr_df[["총성도"]]
y_std = zscore(y_std)
y_std

      총성도
0   -2.374777
1   -3.499357
2   -0.687908
3   -1.812488
4   -0.426672

```

✓ 0초 오후 9:38에 완료됨

4.Regression(설명)

```

[21] model_std = sm.OLS(y_std, X_std) # 모델 생성
      result_std = model_std.fit() # 모델 실행
      print(result_std.summary())

OLS Regression Results
=====
Dep. Variable: 충성도 R-squared (uncentered): 0.774
Model: OLS Adj. R-squared (uncentered): 0.766
Method: Least Squares F-statistic: 103.7
Date: Sat, 02 Mar 2024 Prob (F-statistic): 1.76e-109
Time: 12:38:18 Log-Likelihood: -254.15
No. Observations: 376 AIC: 532.3
Df Residuals: 364 BIC: 579.5
Df Model: 12
Covariance Type: nonrobust
=====
            coef    std err        t    P>|t|    [0.025    0.975]
C(성별)[T.여자]  0.0535   0.048    1.121    0.263    -0.040    0.147
C(결혼)[T.미혼] -0.1051   0.056   -1.892    0.059    -0.214    0.004
C(학력)[T.대학]  0.1299   0.062    2.080    0.038     0.007    0.253
C(학력)[T.대학생]  0.0153   0.061    0.252    0.801    -0.104    0.135
C(학력)[T.초중고생]  0.0685   0.079    0.868    0.386    -0.087    0.223
연령          -0.0847   0.029   -2.901    0.004    -0.142    -0.027
도구           0.0199   0.027    0.741    0.459    -0.033    0.073
보살           0.1102   0.027    4.006    0.000     0.056    0.164
정보           0.1162   0.028    4.135    0.000     0.061    0.171
디자인         0.2868   0.029    9.740    0.000     0.229    0.345
공동체         0.2421   0.034    7.161    0.000     0.176    0.309
불입           0.4010   0.034   11.725    0.000     0.334    0.468
=====
Omnibus:          0.733 Durbin-Watson:       1.883
Prob(Omnibus):   0.693 Jarque-Bera (JB):  0.828
Skew:             -0.042 Prob(JB):        0.661
Kurtosis:          2.786 Cond. No.       6.73
=====

Notes:
[1] R2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[22] print("==== 비표준화 계수 ====")
      print(result.params)
      print("\n")
      print("==== 표준화 계수 ====")

```

✓ 0초 오후 9:38에 완료됨

4.Regression(설명)

```
[0초] [1] R2 is computed without centering (uncentered) since the model does not contain a constant.  
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
[0초] [22] print("===== 비표준화 계수 =====")  
print(result.params)  
print("\n")  
print("===== 표준화 계수 =====")  
print(result_std.params)
```

===== 비표준화 계수 =====

Intercept	-1.090870
C(성별)[T.여자]	0.146277
C(결혼)[T.미혼]	-0.057101
C(학력)[T.대출]	0.329953
C(학력)[T.대학생]	0.131721
C(학력)[T.초중고생]	0.220665
도구	0.023737
보상	0.072752
정보	0.111140
디자인	0.157301
공동체	0.134706
몰입	0.317508
연령	-0.016131

dtype: float64

===== 표준화 계수 =====

C(성별)[T.여자]	0.053465
C(결혼)[T.미혼]	-0.105115
C(학력)[T.대출]	0.129922
C(학력)[T.대학생]	0.015272
C(학력)[T.초중고생]	0.068452
연령	-0.084729
도구	0.019932
보상	0.110160
정보	0.116202
디자인	0.286784
공동체	0.242061
몰입	0.401034

dtype: float64

▼ 5.가정검정

✓ 0초 오후 9:38에 완료됨



5.가정검정

5.가정검정

- <https://ethanweed.github.io/pythonbook/05.04-regression.html#regressionnormality>
- 잔차의 등분산성: Breusch-Pagan
- 잔차의 정규성: Jarque-Bera, Omnibus(D'Angostino's test)
- 독립성(자기상관): Durbin-Watson
- 다중공선성(VIF): Cond. No

5.1 기본 검정

- 잔차의 정규성: Jarque-Bera, Omnibus(D'Angostino's test)
- 독립성(자기상관): Durbin-Watson
- 다중공선성(VIF): Cond. No

[23] `print(result.summary())`

OLS Regression Results

Dep. Variable:	총성도	R-squared:	0.775			
Model:	OLS	Adj. R-squared:	0.768			
Method:	Least Squares	F-statistic:	104.3			
Date:	Sat, 02 Mar 2024	Prob (F-statistic):	1.21e-109			
Time:	12:38:18	Log-Likelihood:	-469.48			
No. Observations:	376	AIC:	965.0			
Df Residuals:	363	BIC:	1016.			
Df Model:	12					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
Intercept	-1.0909	0.466	-2.342	0.020	-2.007	-0.175
CI(성별)[T.여자]	0.1463	0.091	1.600	0.110	-0.033	0.326
CI(결혼)[T.미혼]	-0.0571	0.132	-0.434	0.664	-0.316	0.202
CI(학력)[T.대출]	0.3300	0.129	2.554	0.011	0.076	0.584
CI(학력)[T.대학생]	0.1317	0.129	1.025	0.306	-0.121	0.384
CI(학력)[T.초중고생]	0.2207	0.155	1.425	0.155	-0.084	0.525
도구	0.0237	0.031	0.761	0.447	-0.038	0.085
보상	0.0728	0.019	3.835	0.000	0.035	0.110
정보	0.1111	0.027	4.168	0.000	0.059	0.164
디자인	0.1573	0.016	9.824	0.000	0.126	0.189

✓ 0초 오후 9:38에 완료됨

5. 가정검정

5.2 잔차의 등분산 검정

- 잔차의 등분산성 테스트: Breush-Pagan 테스트:

```
✓ [24] # 잔차의 등분산성 테스트: Breush-Pagan 테스트:
import statsmodels.stats.api as sms
from statsmodels.compat import lzip

name = ["Lagrange multiplier statistic", "p-value", "f-value", "f p-value"]
test = sms.het_breuschpagan(result.resid, result.model.exog)
lzip(name, test)

[('Lagrange multiplier statistic', 13.692360414295386),
 ('p-value', 0.32078345537272),
 ('f-value', 1.1432105130492498),
 ('f p-value', 0.32361681851581087)]
```

✓ [25] # 잔차 풀웃

```
# 표준화 잔차 생성
influence = result.get_influence()
res_standard = influence.resid_studentized_internal

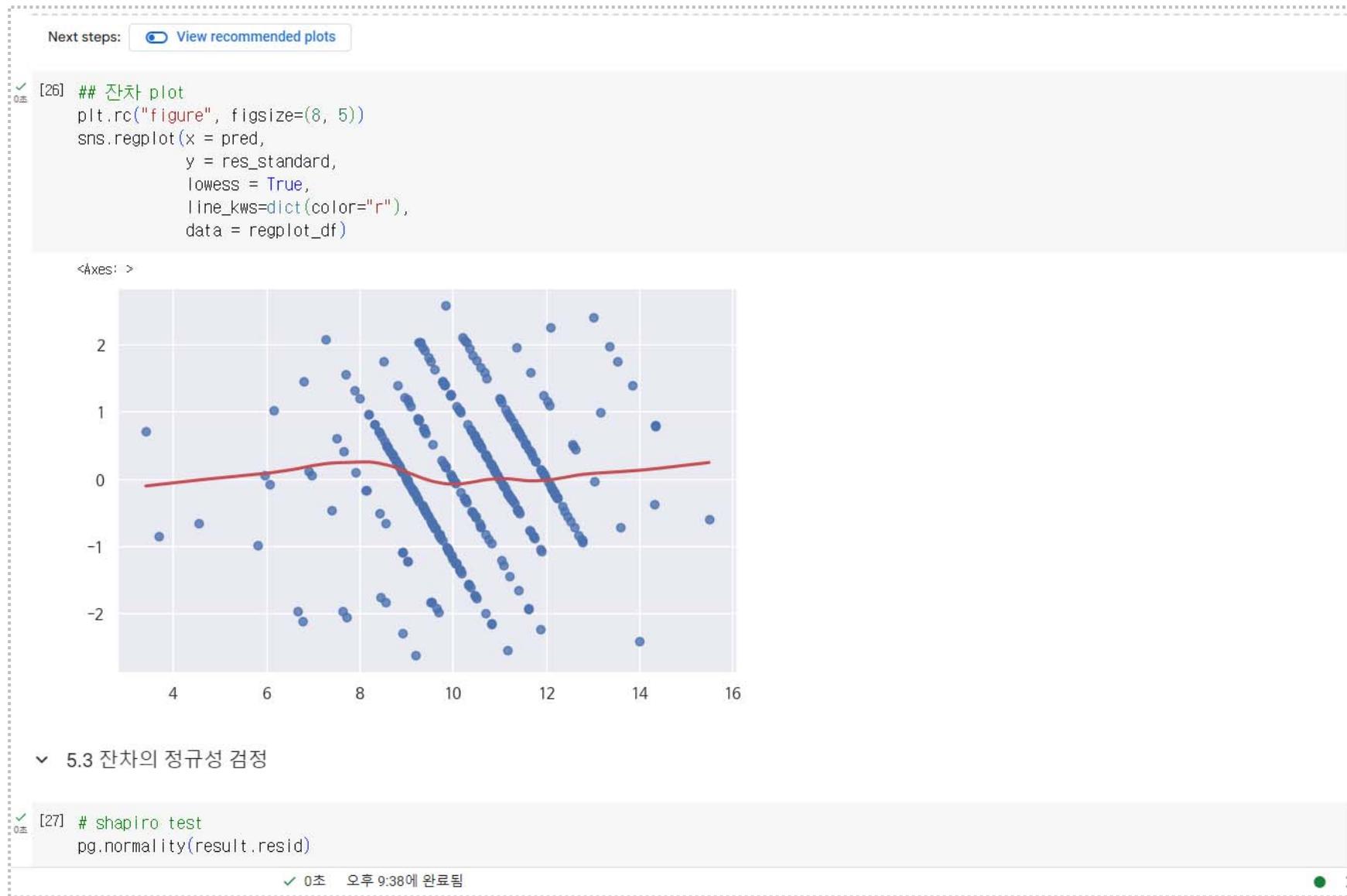
# 예측값 생성
pred = result.predict(X)

# 데이터 프레임으로 생성
regplot_df = pd.DataFrame({'pred': pred, 'res_standard': res_standard})
regplot_df
```

	pred	res_standard	grid icon
0	5.961116	0.046449	grid icon
1	3.414913	0.709523	grid icon
2	8.423383	0.686386	grid icon
3	6.163037	1.016076	grid icon
4	10.221127	0.720010	grid icon

✓ 0초 오후 9:38에 완료됨

5.가정검정



5.가정검정

▼ 5.3 잔차의 정규성 검정

```
[27] # shapiro test  
pg.normality(result.resid)
```

	pval	normal
0	0.995166	0.292246
		True

```
[28] ## QQ plot  
plt.rc("figure", figsize=(8, 5))  
sm.qqplot(res_standard, line = 's')  
sns.despine()
```

Sample Quantiles

Theoretical Quantiles

✓ 0초 오후 9:38에 완료됨

5.가정검정

▼ 5.4 이상치 제거

```
[29] # 표준화 잔차를 이용한 이상치 확인
stud_res = result.outlier_test()
stud_res.sort_values(by = "student_resid", ascending = False).head(5)
```

student_resid	unadj_p	bonf(p)
25	2.595779	0.009822
126	2.412650	0.016333
154	2.271076	0.023729
205	2.120766	0.034621
53	2.084467	0.037818

▶ ## cooks_distance를 이용한 이상치 확인
https://www.statsmodels.org/dev/examples/notebooks/generated/regression_plots.html

```
fig = sm.graphics.influence_plot(result, criterion="cooks")
plt.rc("figure", figsize=(8, 5))
plt.rc("font", size = 7)
fig.tight_layout(pad = 1.0)
```

5.가정검정

```
[31] # cooks_distance를 이용한 이상치 확인
from statsmodels.stats.outliers_influence import OLSInfluence
test_class = OLSInfluence(result).cooks_distance
pd.DataFrame(test_class).T.sort_values(by = 0, ascending = False).head(10)

          0      1
231  0.034821  1.0
126  0.034684  1.0
25   0.025885  1.0
296  0.023968  1.0
53   0.021942  1.0
283  0.021776  1.0
282  0.021569  1.0
222  0.017651  1.0
154  0.017189  1.0
365  0.016932  1.0

[32] # 이상치 제거
# mr_df = mr_df.drop(120)

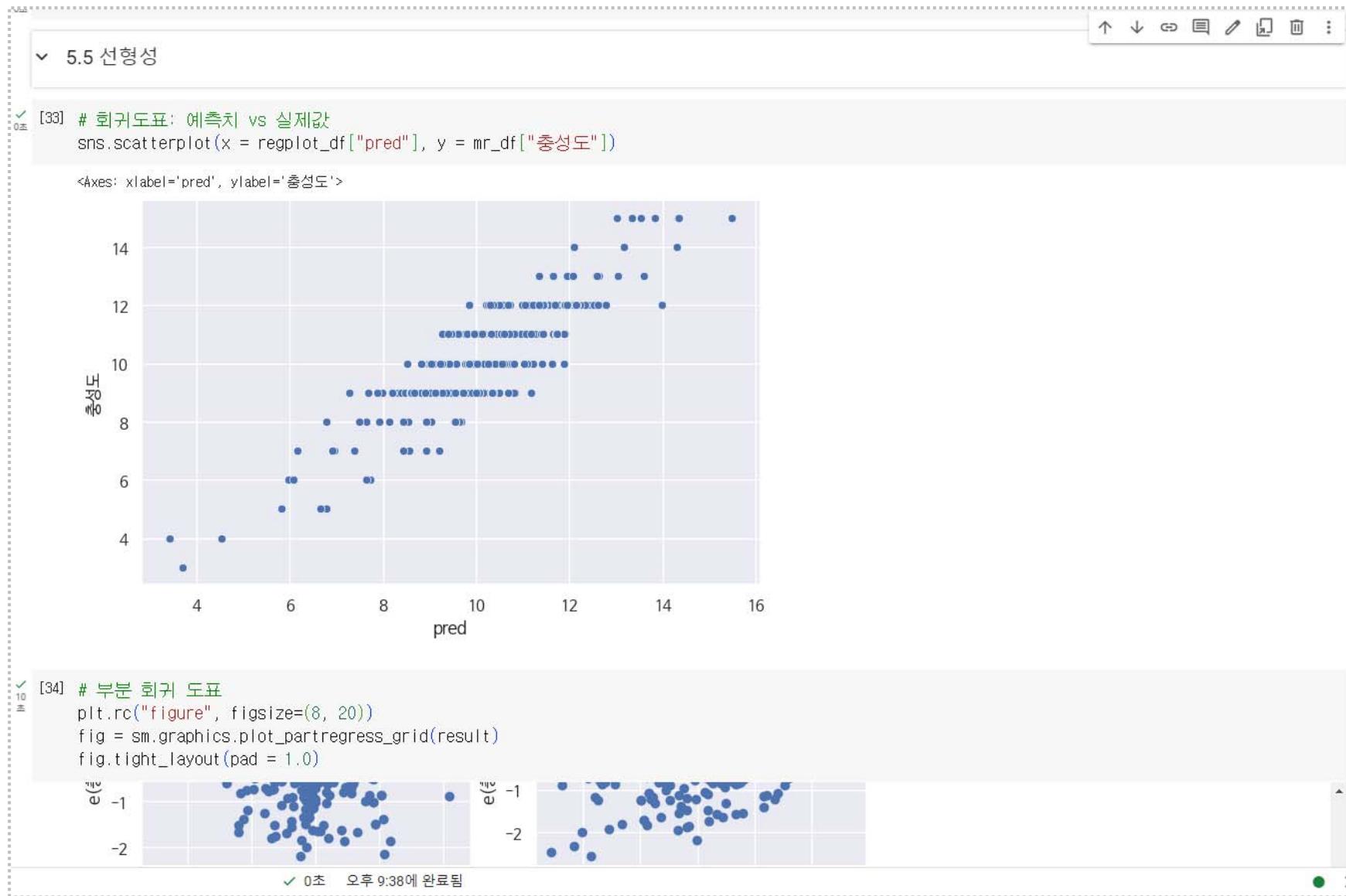
▼ 5.5 선형성

[33] # 회귀도표: 예측치 vs 실제값
sns.scatterplot(x = regplot_df["pred"], y = mr_df["충성도"])

<Axes: xlabel='pred', ylabel='충성도'>

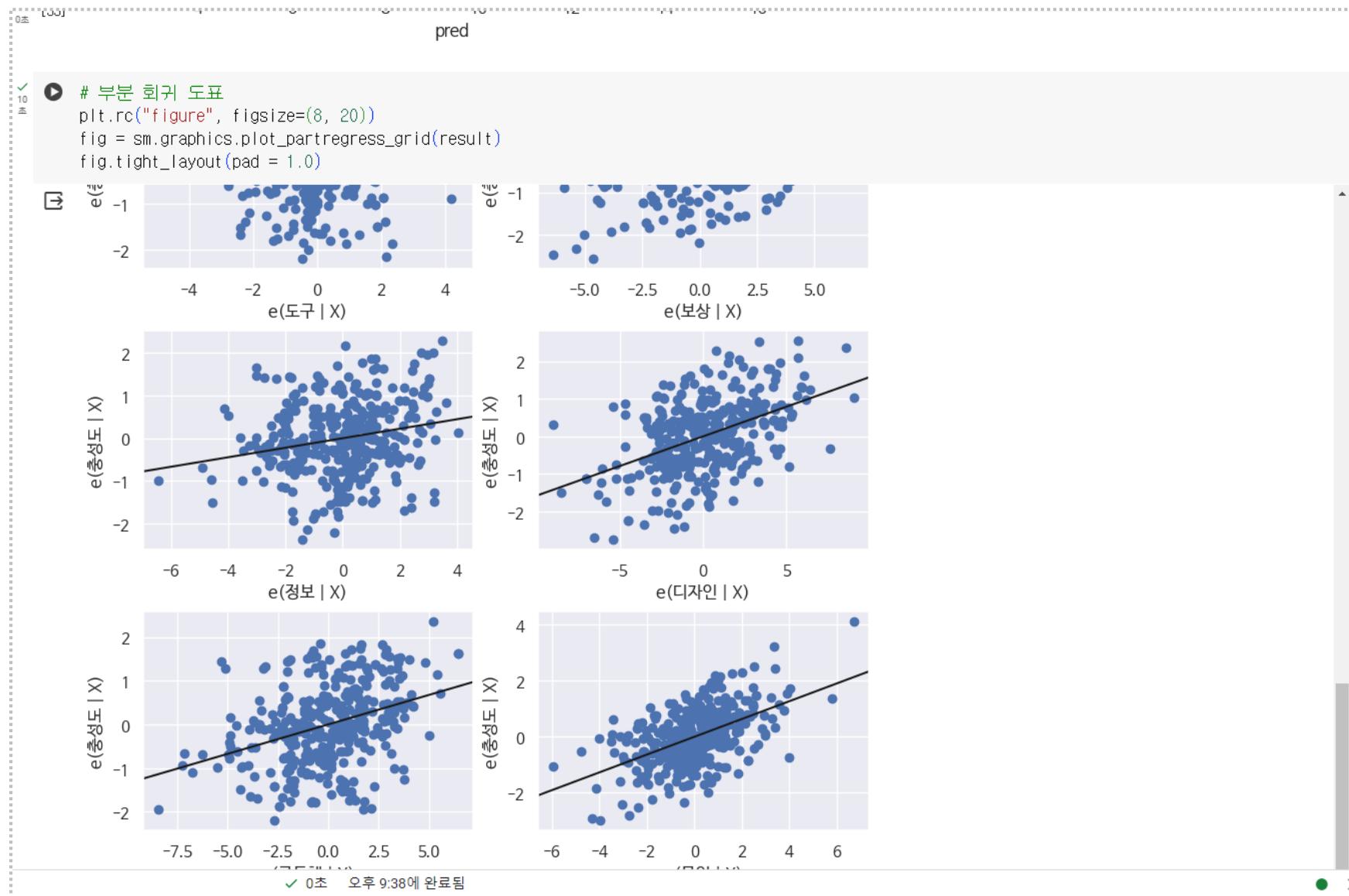
```

5.가정검정



5. 가정검정

LGE Internal Use Only



✓ 0초 오후 9:38에 완료됨

5.가정검정



Regression(설명)

LGE Internal Use Only

- ❖ 온라인게임 충성도에 영향을 주는 요인을 분석한 결과, 보상, 정보, 디자인, 공동체, 몰입이 영향을 주는 것으로 나타났다($F=22.183$, $p=-.000$). 설명력($Adj R^2$)은 0.768으로 나타났다. 이 중에서 가장 중요한 요인으로는 몰입(0.397)으로 나타났다.

독립변수	Estimate	SE	Stand. Estimate	t	p
Intercept	-0.870	0.482		-1.805	0.072
도구	0.024	0.031	0.020	0.761	0.447
보상	0.073	0.019	0.106	3.835	<.001
정보	0.111	0.027	0.117	4.168	<.001
디자인	0.157	0.016	0.289	9.824	<.001
공동체	0.135	0.019	0.243	7.211	<.001
몰입	0.318	0.027	0.397	11.597	<.001
연령	-0.016	0.008	-0.063	-1.926	0.055
성별:					
여자 – 남자	0.146	0.091	0.082	1.600	0.110
결혼:					
미혼 – 결혼	-0.057	0.132	-0.032	-0.434	0.664
학력:					
고졸 – 초중고생	-0.221	0.155	-0.124	-1.425	0.155
대학생 – 초중고생	-0.089	0.139	-0.050	-0.638	0.524
대졸 – 초중고생	0.109	0.140	0.061	0.782	0.435

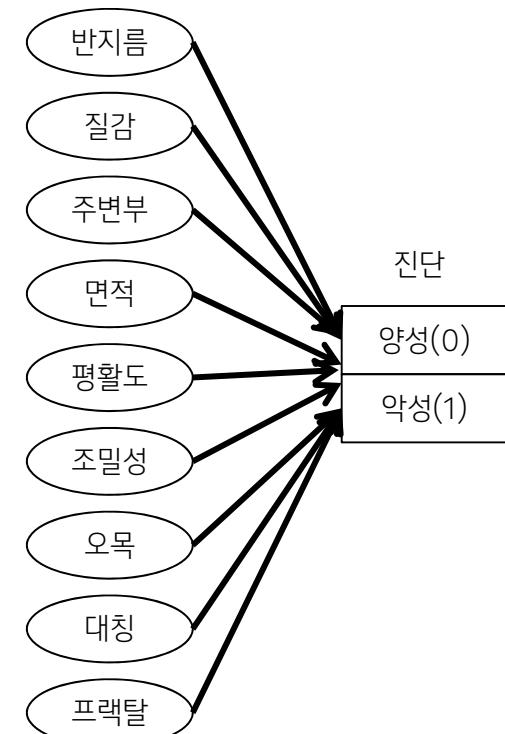
IV. Logistic Regression

Logistic Regression이란

Logistic regression

❖ 문제의 정의

- 유방암 진단 (0 = 양성, 1 = 악성)
- 세포조직의 FNA(Fine Needle Aspirate)의 디지털화된 이미지
- 반지름(주변의 중심에서 점까지의 거리 평균)
- 질감(그레이 스케일 값의 표준 편차)
- 주변부
- 면적
- 평활도(반지름 길이의 국지적 변동)
- 조밀성(주변² /면적 - 1.0)
- 오목(윤곽의 오목 부분의 심각도)
- 대칭
- 프랙탈 차원: 공간에 패턴을 얼마나 조밀하게 채우는지 나타내는 비율
- 14_1.LR.csv



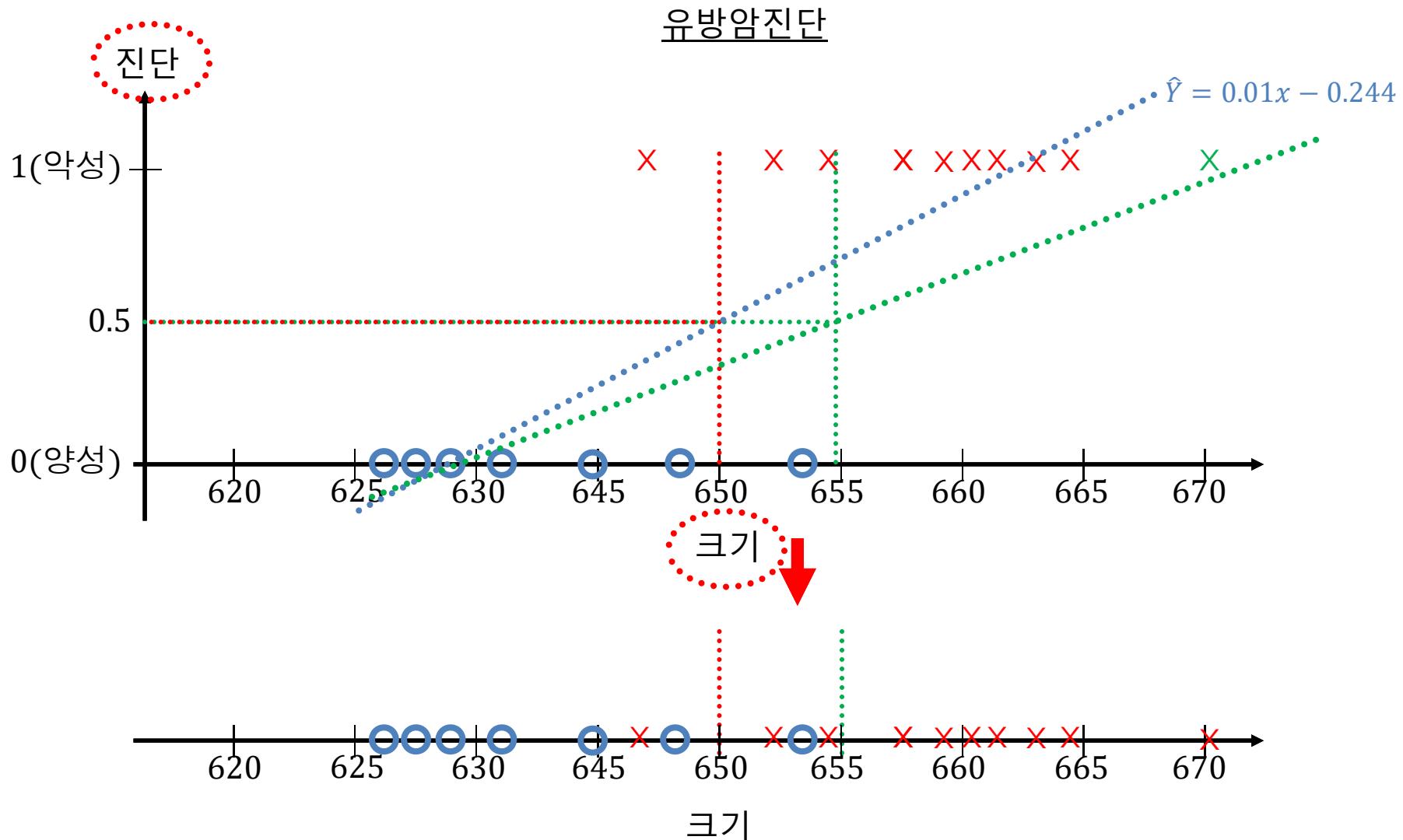
Logistic regression

❖ 데이터

id	diagnosis	radius	texture	perimeter	area	smoothness	compactness	concavity	symmetry	fractal_dimension
번호	진단	반지름	질감	주변부	크기	평활도	조밀성	오목	대칭	프랙탈
1	1	18.0	10.4	122.8	1001.0	0.118	0.278	0.300	0.242	0.079
2	1	20.6	17.8	132.9	1326.0	0.085	0.079	0.087	0.181	0.057
3	1	19.7	21.3	130.0	1203.0	0.110	0.160	0.197	0.207	0.060
4	0	13.5	14.4	87.5	566.3	0.098	0.081	0.067	0.189	0.058
5	0	13.1	15.7	85.6	520.0	0.108	0.127	0.046	0.197	0.068
6	0	9.5	12.4	60.3	273.9	0.102	0.065	0.030	0.182	0.069
7	1	11.4	20.4	77.6	386.1	0.143	0.284	0.241	0.260	0.097
8	1	20.3	14.3	135.1	1297.0	0.100	0.133	0.198	0.181	0.059
9	1	12.5	15.7	82.6	477.1	0.128	0.170	0.158	0.209	0.076
10	1	18.3	20.0	119.6	1040.0	0.095	0.109	0.113	0.179	0.057
11	1	13.7	20.8	90.2	577.9	0.119	0.165	0.094	0.220	0.075
12	1	13.0	21.8	87.5	519.8	0.127	0.193	0.186	0.235	0.074
...										
566	1	20.1	28.3	131.2	1261.0	0.098	0.103	0.144	0.175	0.055
567	1	16.6	28.1	108.3	858.1	0.085	0.102	0.093	0.159	0.056
568	1	20.6	29.3	140.1	1265.0	0.118	0.277	0.351	0.240	0.070
569	0	7.8	24.5	47.9	181.0	0.053	0.044	0.000	0.159	0.059

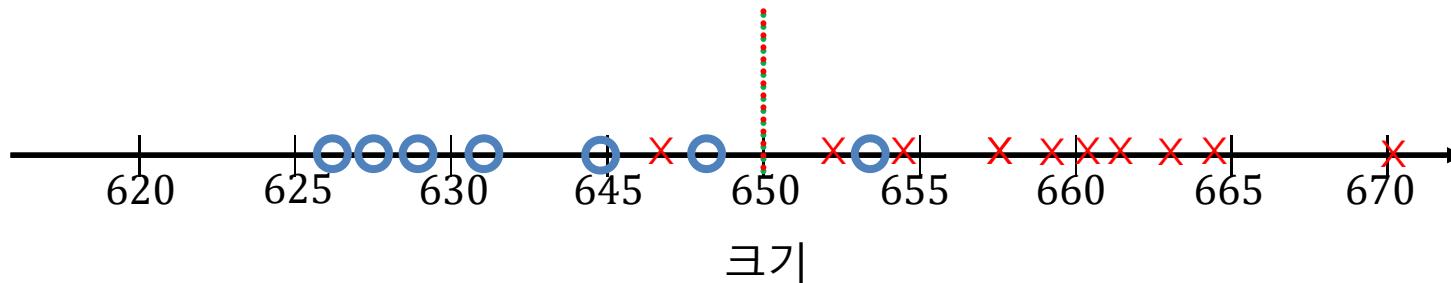
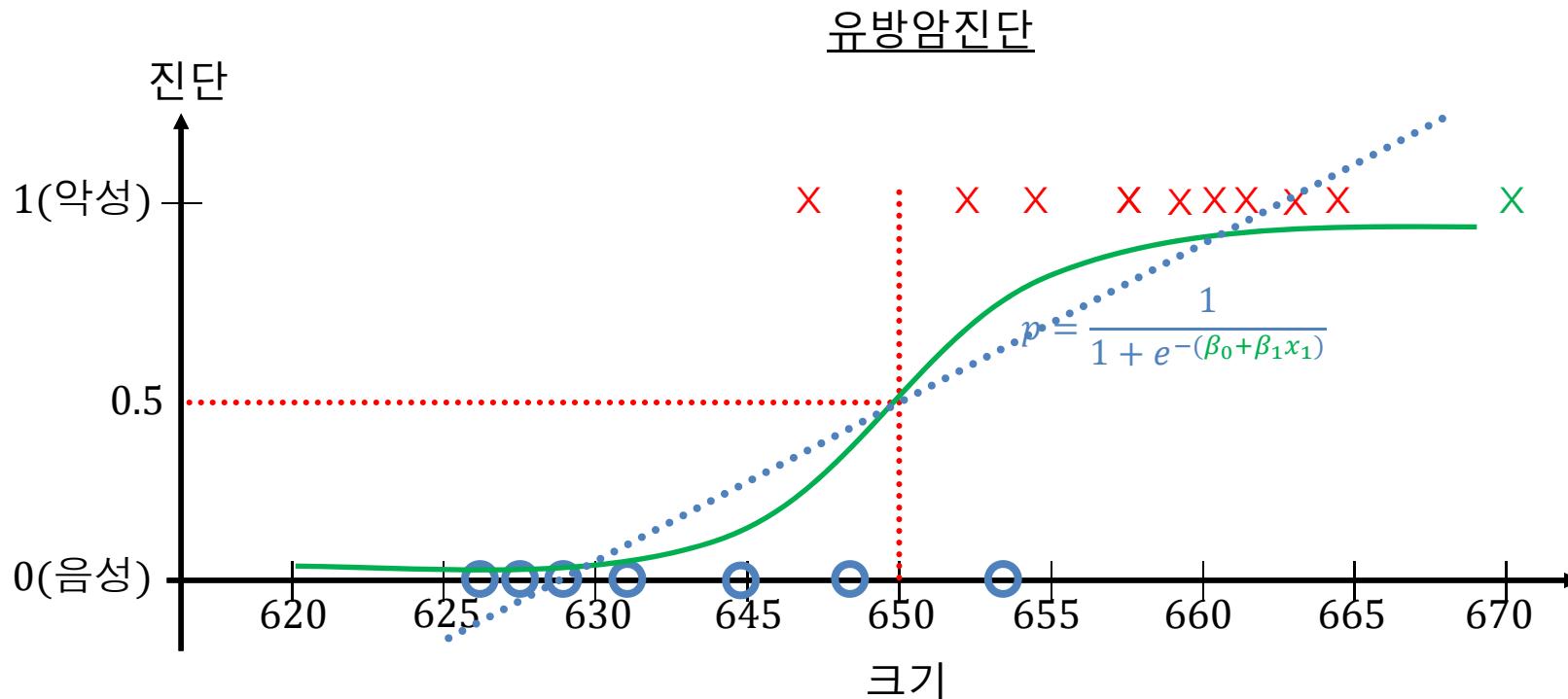
Logistic regression(독립변수1개)

LGF Internal Use Only



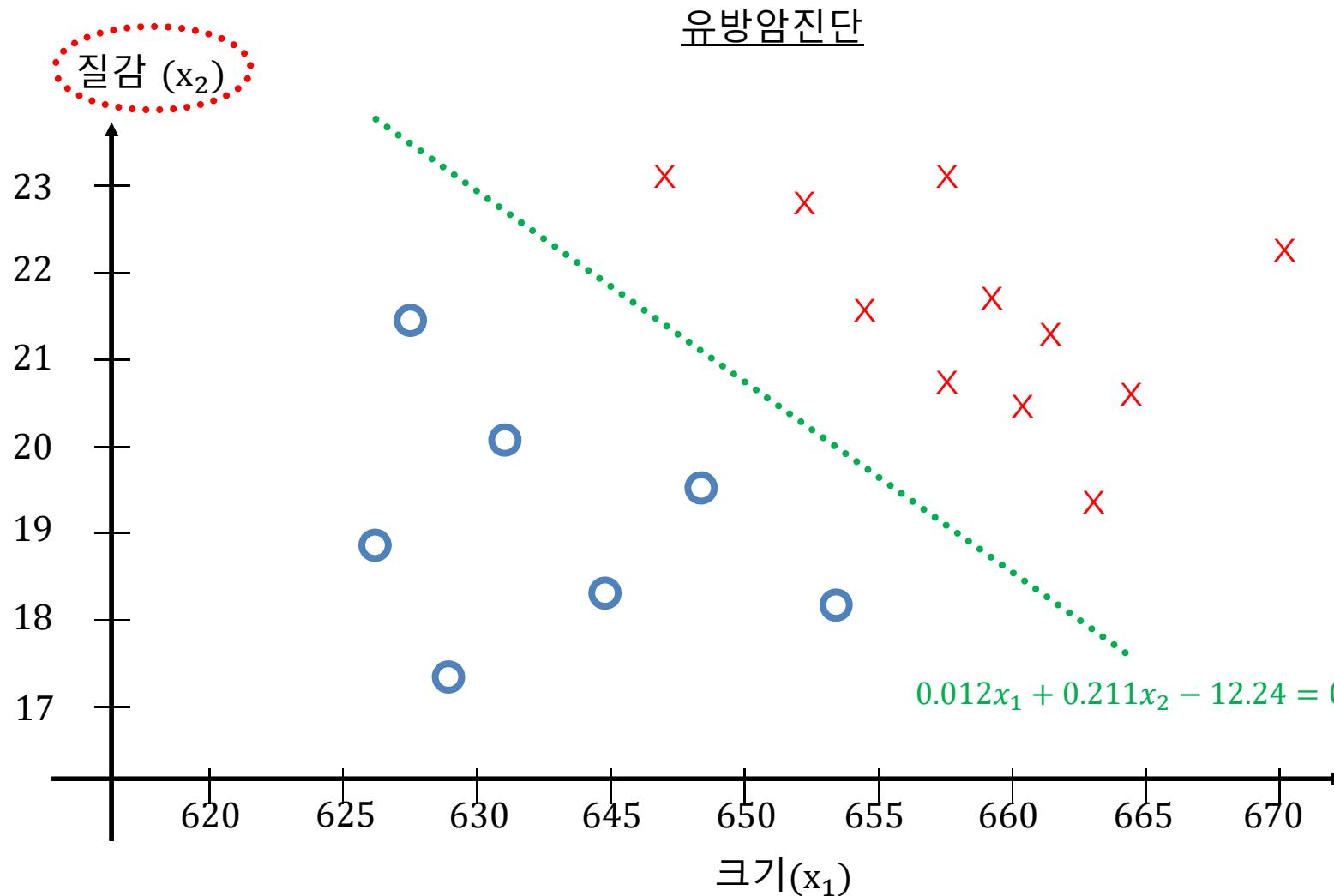
Logistic regression(독립변수1개)

LGF Internal Use Only



Logistic regression(독립변수2개)

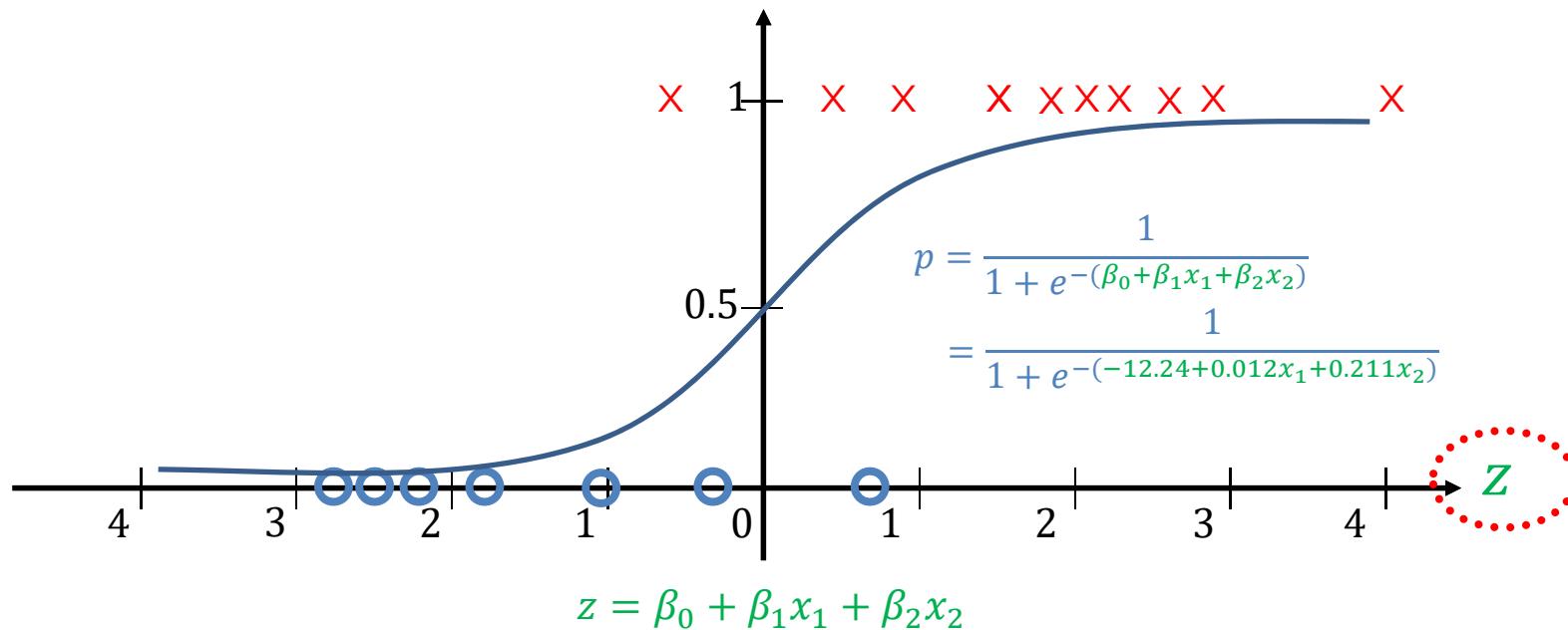
LGE Internal Use Only



Logistic regression(독립변수2개)

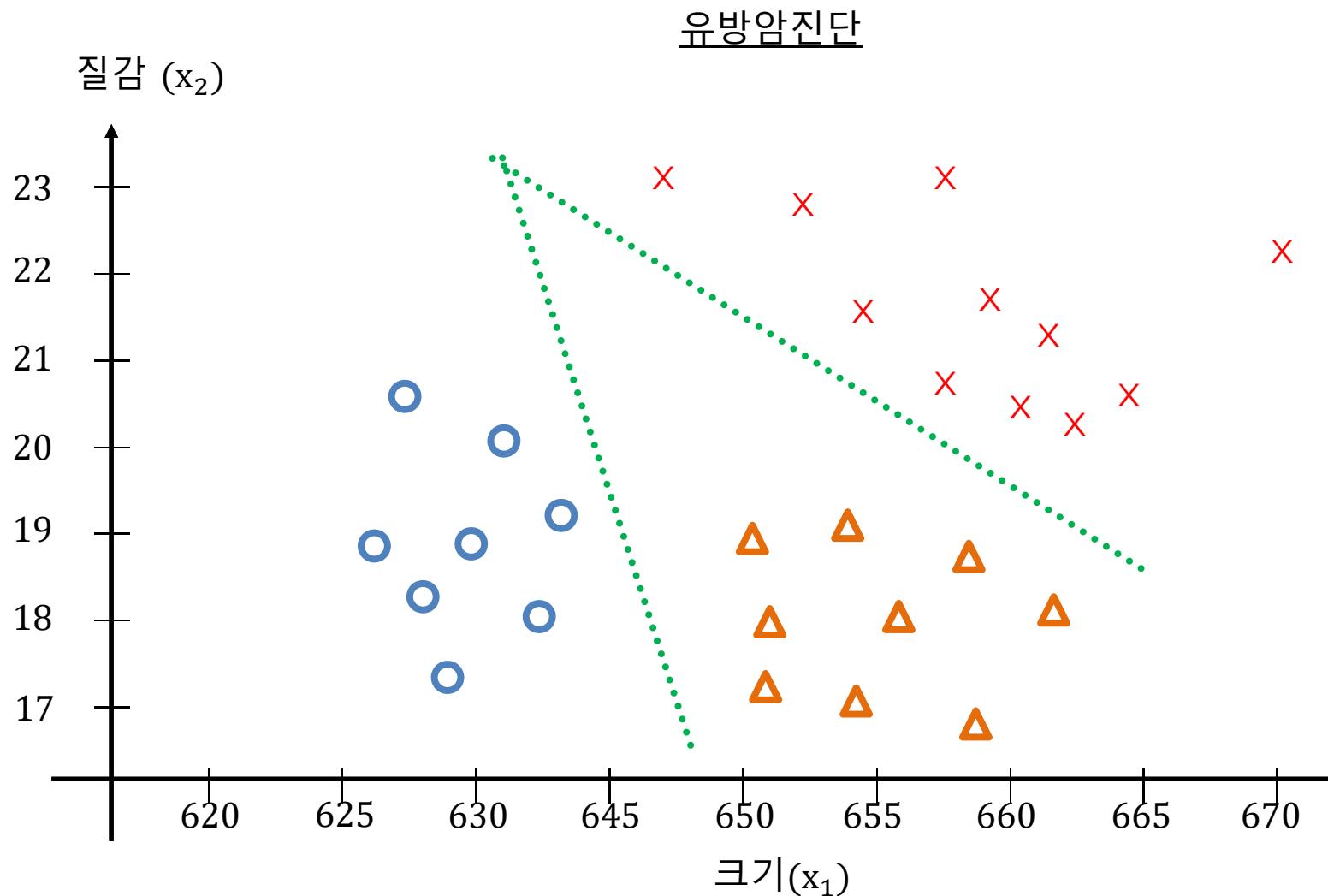
LGE Internal Use Only

유방암진단



Logistic regression(범주 3개)

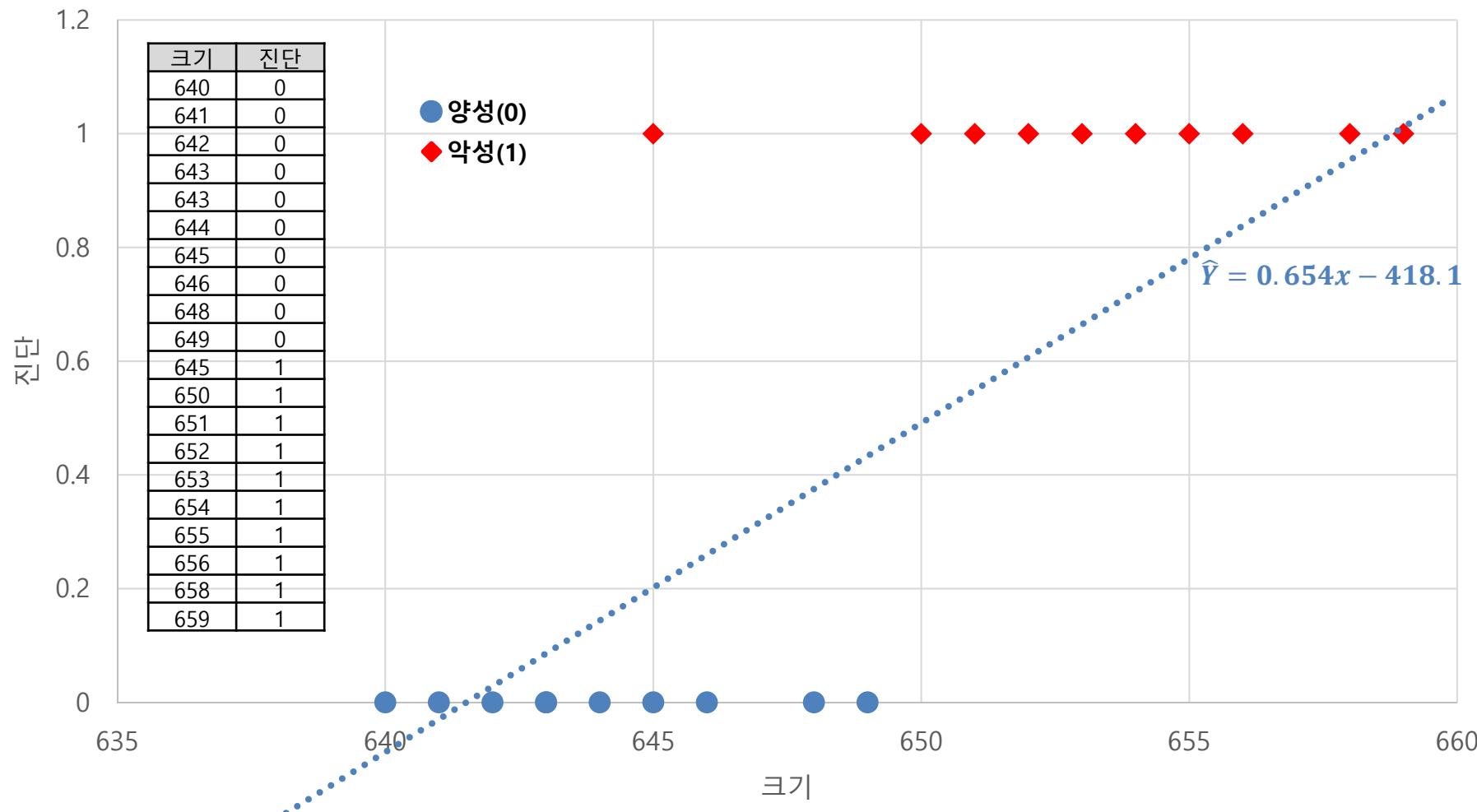
LGE Internal Use Only



Logistic regression(독립변수1개)

LGF Internal Use Only

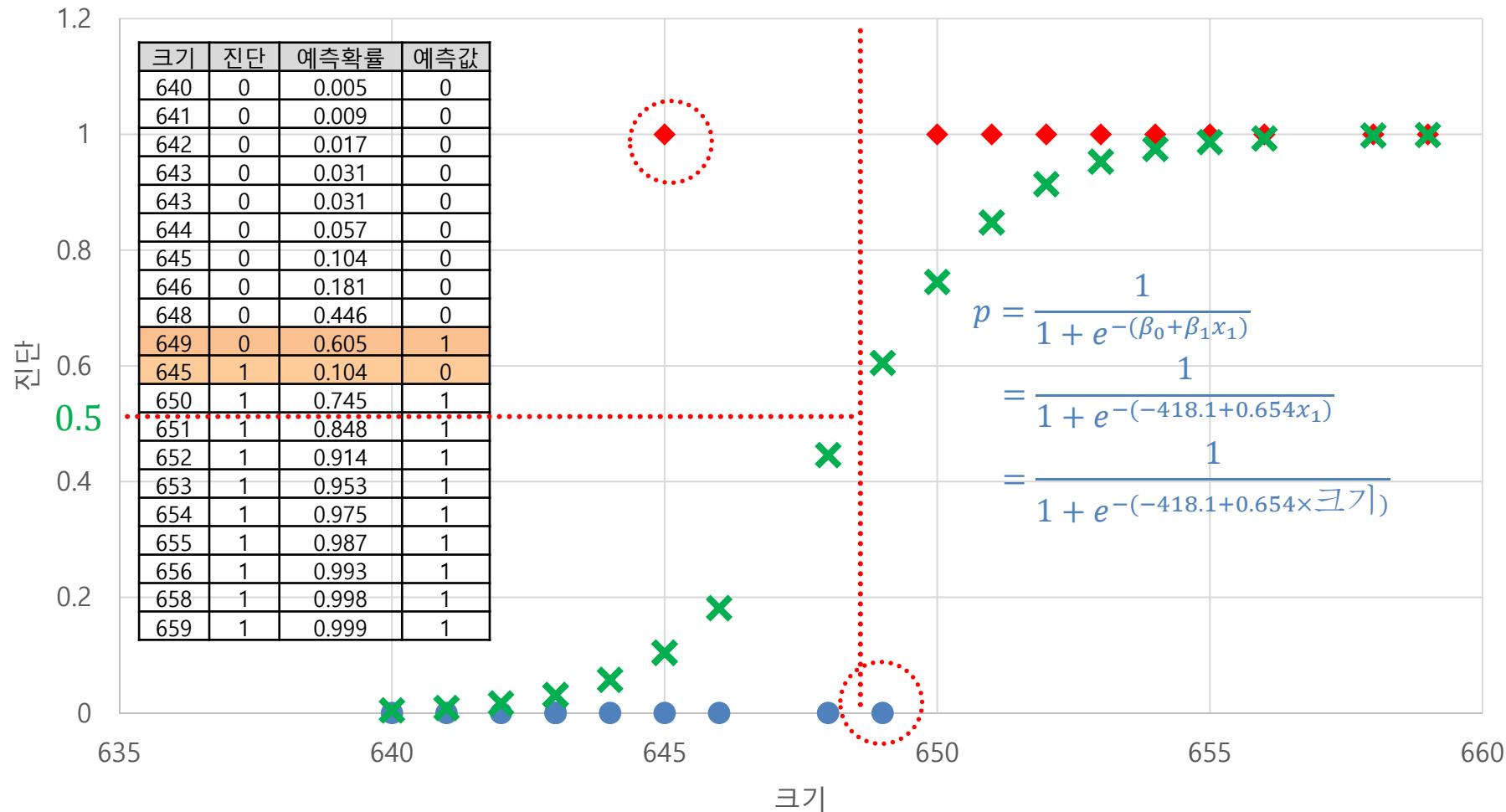
유방암 진단



Logistic regression(독립변수 1개)

LGF Internal Use Only

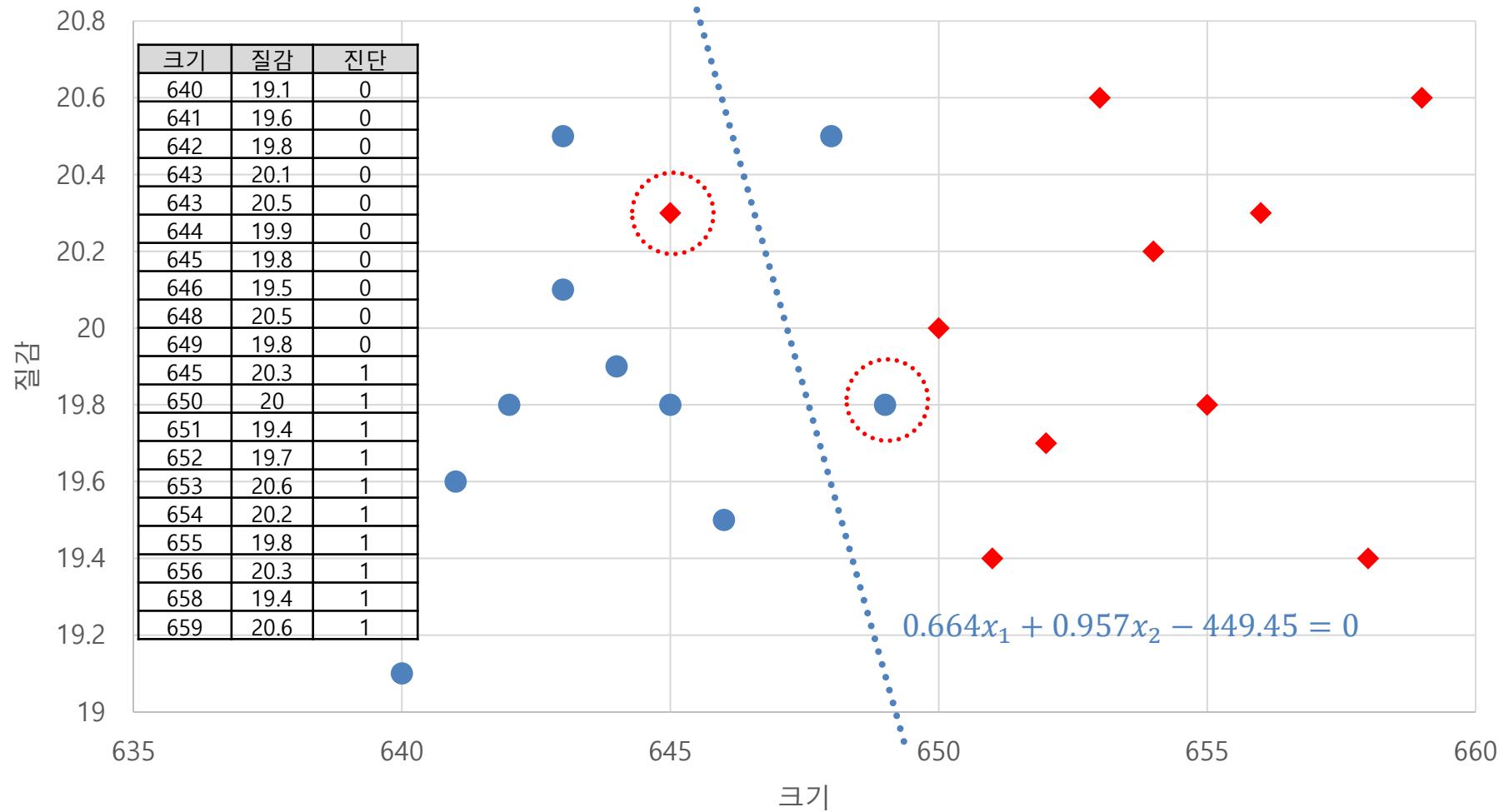
유방암 진단



Logistic regression(독립변수 2개)

LGE Internal Use Only

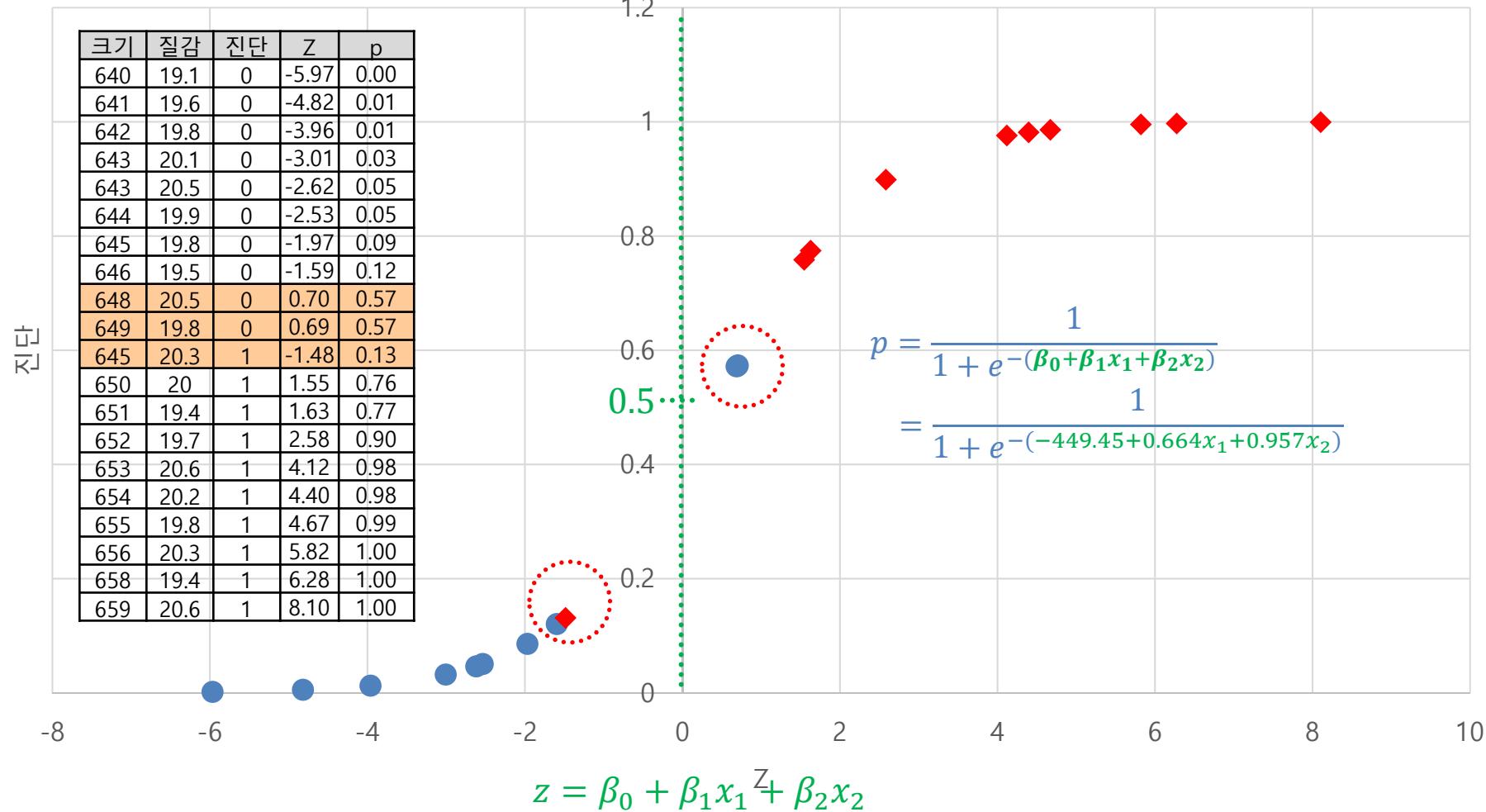
유방암 진단



Logistic regression(독립변수 2개)

LGE Internal Use Only

유방암 진단



Logistic regression

Model Coefficients - 진단

Predictor	Estimate	SE	Z	p	Odds ratio
Intercept	-34.914	7.612	-4.587	< .001	0.000
질감	0.476	0.080	5.926	< .001	1.609
크기	0.017	0.003	5.330	< .001	1.017
평활도	152.492	30.847	4.943	< .001	1.6839843467521157e+66
조밀성	0.445	12.814	0.035	0.972	1.561
오목	29.013	6.784	4.276	< .001	3980800197966.975
대칭	23.301	12.354	1.886	0.059	13161416601.154
프랙탈	-119.861	90.354	-1.327	0.185	0.000

Note. Estimates represent the log odds of "진단 = 악성" vs. "진단 = 양성"

$$P(\text{악성}) = \frac{1}{1 + e^{(-34.914 + 0.476\text{질감} + 0.017\text{크기} + \dots)}}$$

Logistic regression

- ❖ 클래스 1에 속할 확률 추정
 - 실제값 Y 는 0 or 1의 값만 가짐 → 확률로 변환

$$\begin{cases} p = P(Y = 1) \\ 1 - p = P(Y = 0) \end{cases}$$

- ❖ 오즈(Odds)
 - 클래스 0에 속하는 확률에 대한 클래스 1에 속하는 확률의 비

$$Odds(Y = 1) = \frac{Y = 1}{Y = 0} = \frac{p}{1 - p}$$

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots)}} = \frac{Odds}{1 + Odds}$$

$$Odds(Y = 1) = e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots)}$$

Logistic regression

❖ Exp(B) 해석방법

- Odds

$$Odds(\text{진단} = \text{악성}) = e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots)}$$

- 대조 : 다른 변수는 고정, area가 1 증가할 때 Odds

$$Odds(\text{진단} = \text{악성} | \text{크기} = 0) = e^{(0.017 \times 0)} = 1$$

$$Odds(\text{진단} = \text{악성} | \text{크기} = 1) = e^{(0.017 \times 1)} = 1.017$$

$$Odds(\text{진단} = \text{악성} | \text{크기} = 30) = e^{(0.017 \times 10)} = 1.665$$

$$Odds(\text{진단} = \text{악성} | \text{크기} = 100) = e^{(0.017 \times 100)} = 5.474$$

- 실제 Odds

$$Odds(\text{진단} = \text{악성} | \text{크기} = 0) = e^{(-34.914 + 0.017 \times 0)} = 6.87E-16$$

$$Odds(\text{진단} = \text{악성} | \text{크기} = 100) = e^{(-34.914 + 0.017 \times 100)} = 3.76E-15$$

Logistic regression

❖ 비교

- $Odds \approx 1.6$

$$Odds(\text{진단 = 악성} | \text{크기} = 30) = e^{(0.017 \times 10)} = 1.665$$

$$Odds(\text{진단 = 악성} | \text{질감} = 1.1) = e^{(0.0476 \times 1.1)} = 1.688$$

$$Odds(\text{진단 = 악성} | \text{평활도} = 0.0034) = e^{(152.49 \times 0.0034)} = 1.580$$

$$Odds(\text{진단 = 악성} | \text{크기} = 30) = e^{(-34.914 + 0.017 \times 30)} = 1.14E-15$$

$$Odds(\text{진단 = 악성} | \text{질감} = 1.1) = e^{(-34.914 + 0.0476 \times 1.1)} = 1.16E-15$$

$$Odds(\text{진단 = 악성} | \text{평활도} = 0.0034) = e^{(-34.914 + 152.49 \times 0.0034)} = 1.15E-15$$

Logistic regression

❖ 비교

- $Odds \approx 5.4$

$$Odds(\text{진단} = \text{악성} | \text{크기} = 100) = e^{(0.017 \times 100)} = 5.474$$

$$Odds(\text{진단} = \text{악성} | \text{질감} = 3.5) = e^{(0.0476 \times 3.5)} = 5.291$$

$$Odds(\text{진단} = \text{악성} | \text{평활도} = 0.011) = e^{(152.49 \times 0.011)} = 5.331$$

$$Odds(\text{진단} = \text{악성} | \text{크기} = 100) = e^{(-34.914 + 0.017 \times 100)} = 3.76E-15$$

$$Odds(\text{진단} = \text{악성} | \text{질감} = 3.5) = e^{(-34.914 + 0.0476 \times 3.5)} = 3.64E-15$$

$$Odds(\text{진단} = \text{악성} | \text{평활도} = 0.011) = e^{(-34.914 + 152.49 \times 0.011)} = 3.68E-15$$

Logistic regression

❖ 상대비교 - 표준화

	B	S.E.	Wald	df	유의수준	Exp(B)
Z질감	2.050	.346	35.116	1	.000	7.764
Z크기	5.992	1.124	28.407	1	.000	400.024
Z평활도	2.152	.435	24.438	1	.000	8.604
Z조밀성	.024	.678	.001	1	.972	1.024
Z오목	2.318	.542	18.288	1	.000	10.158
Z대칭	.641	.340	3.557	1	.059	1.898
Z프랙탈	-.851	.642	1.760	1	.185	.427
상수	-.574	.268	4.572	1	.032	.563

14_1.Regression(Logistic)

The screenshot shows a Jupyter Notebook interface with the following structure:

- Section 1: 14_1.Regression(Logistic)**
 - [Link to statsmodels documentation](https://www.statsmodels.org/stable/discretemod.html)
- Section 2: 1.기본 package 설정**

```
[1] # 그래프에서 한글 폰트 인식하기
!sudo apt-get install -y fonts-nanum
!sudo fc-cache -fv
!rm ~/.cache/matplotlib -rf

[2] !pip install pingouin

# *** 런타임 다시 시작
```
- Section 3: # 1.기본**

```
[3] # 1.기본
import numpy as np # numpy 패키지 가져오기
import matplotlib.pyplot as plt # 시각화 패키지 가져오기
import seaborn as sns # 시각화

# 2.데이터 가져오기
import pandas as pd # csv -> dataframe으로 전환

# 3.통계분석 package
import pingouin as pg
from scipy import stats
import statsmodels.api as sm
```
- Section 4: # 기본세팅**

```
[4] # 기본세팅
# 테마 설정
sns.set_theme(style = "darkgrid")
```

At the bottom of the notebook, there is a status bar indicating "✓ 0초 오후 9:43에 완료됨". The top right corner of the window has standard window control buttons (minimize, maximize, close).

2.데이터 불러오기

LGE Internal Use Only

▼ 2.데이터 불러오기

▼ 2.1 데이터 프레임으로 저장

- 원본데이터(csv)를 dataframe 형태로 가져오기(pandas)

[3] lr_df = pd.read_csv('https://raw.githubusercontent.com/echo-bigdata/statistics-python/main/14_1_LR.csv', encoding='cp949')
lr_df.head()

	id	진단	반지름	질감	주변부	크기	평활도	조밀성	오목	대칭	프랙탈
0	1	1	18.0	10.4	122.8	1001.0	0.118	0.278	0.300	0.242	0.079
1	2	1	20.6	17.8	132.9	1326.0	0.085	0.079	0.087	0.181	0.057
2	3	1	19.7	21.3	130.0	1203.0	0.110	0.160	0.197	0.207	0.060
3	4	0	13.5	14.4	87.5	566.3	0.098	0.081	0.067	0.189	0.058
4	5	0	13.1	15.7	85.6	520.0	0.108	0.127	0.046	0.197	0.068

Next steps: [View recommended plots](#)

▼ 2.2 범주형 변수 처리

- 가변수 처리시 문자로 처리를 해야 변수명 구분이 쉬움

[4] lr_df['진단'].replace({0:'양성', 1:'악성'}, inplace=True)
lr_df['진단'] = lr_df['진단'].astype('category')
lr_df

	id	진단	반지름	질감	주변부	크기	평활도	조밀성	오목	대칭	프랙탈
0	1	악성	18.0	10.4	122.8	1001.0	0.118	0.278	0.300	0.242	0.079
1	2	악성	20.6	17.8	132.9	1326.0	0.085	0.079	0.087	0.181	0.057
2	3	악성	19.7	21.3	130.0	1203.0	0.110	0.160	0.197	0.207	0.060

3. 기술통계

```

▼ 3. 기술통계
[8] # 그룹별 기술통계
lr_df.describe().round(3).T

  count   mean    std    min    25%    50%    75%    max
id      565.0 283.000 163.246  1.000 142.000 283.000 424.000 565.000
반지름  565.0 14.143  3.534   7.000 11.700 13.400 15.900 28.100
질감    565.0 19.292  4.310   9.700 16.200 18.800 21.800 39.300
주변부  565.0 92.034  24.368  43.800 75.200 86.300 104.300 188.500
크기    565.0 655.836 352.944 143.500 419.800 551.100 788.500 2501.000
평활도  565.0 0.096   0.014   0.053   0.086   0.096   0.105   0.163
조밀성  565.0 0.105   0.053   0.019   0.065   0.094   0.131   0.345
오목    565.0 0.089   0.080   0.000   0.030   0.062   0.132   0.427
대칭    565.0 0.181   0.027   0.106   0.162   0.179   0.196   0.304
프랙탈  565.0 0.063   0.007   0.050   0.058   0.062   0.066   0.097

[9] # 범주형 변수
# lecture_df.columns
categorical_features = ['진단']

for col in categorical_features:
    print("----", col, "----")
    results = lr_df[col].value_counts()
    print(results, "\n")

---- 진단 ----
알설    357
악설    208
Name: 진단, dtype: int64

✓ 0초 오후 9:43에 완료됨

```

4.Logistic Regression

▼ 4.Logistic Regression

- <https://www.statsmodels.org/stable/examples/notebooks/generated/ols.html>
- 수치형 + 범주형
- dmatrix 사용

▼ 4.1 회귀분석

```
[10] columns = ['반지름', '질감', '주변부', '크기', '평활도', '조밀성', '오목', '대칭', '프랙탈']

# 다중공선성 제거: 반지름, 주변부
# columns = ['질감', '크기', '평활도', '조밀성', '오목', '대칭', '프랙탈']

formula = "진단 ~ " + " + ".join(columns)
formula

'진단 ~ 반지름 + 질감 + 주변부 + 크기 + 평활도 + 조밀성 + 오목 + 대칭 + 프랙탈'

[11] # dmatrix 이용
from patsy import dmatrices

y, X = dmatrices(formula,
                  data = lr_df,
                  return_type = 'dataframe')

[12] X.head()
```

	Intercept	반지름	질감	주변부	크기	평활도	조밀성	오목	대칭	프랙탈
0	1.0	18.0	10.4	122.8	1001.0	0.118	0.278	0.300	0.242	0.079
1	1.0	20.6	17.8	132.9	1326.0	0.085	0.079	0.087	0.181	0.057
2	1.0	19.7	21.3	130.0	1203.0	0.110	0.160	0.197	0.207	0.060
3	1.0	13.5	14.4	87.5	566.3	0.098	0.081	0.067	0.189	0.058

scroll... ✓ 0초 오후 9:43에 완료됨

4.Logistic Regression

Next steps: [View recommended plots](#)

```
[13] y = y.drop('진단[양성]', axis = 1)
y.head()
```

진단[양성]	0
0	1.0
1	1.0
2	1.0
3	0.0
4	0.0

```
[14] model = sm.Logit(y, X) # 모델 생성
result = model.fit() # 모델 실행

Optimization terminated successfully.
    Current function value: 0.104391
    Iterations 11
```

```
[15] print(result.summary())
```

Dep. Variable:	진단[양성]	No. Observations:	565			
Model:	Logit	Df Residuals:	555			
Method:	MLE	Df Model:	9			
Date:	Sat, 02 Mar 2024	Pseudo R-squ.:	0.8413			
Time:	12:43:11	Log-Likelihood:	-58.981			
converged:	True	LL-Null:	-371.75			
Covariance Type:	nonrobust	LLR p-value:	6.918e-129			
	coef	std err	z	P> z	[0.025	0.975]
Intercept	-20.9352	14.824	-1.412	0.158	-49.989	8.118
반지름	0.4226	4.051	0.104	0.917	-7.517	8.362
질감	0.4805	0.081	5.906	0.000	0.321	0.640
주변부	-0.3373	0.536	-0.629	0.529	-1.388	0.713
크기	0.0370	0.019	1.976	0.048	0.000	0.074
평활도	151.2655	31.354	4.824	0.000	89.812	212.719

✓ 0초 오후 9:43에 완료됨

4.Logistic Regression

```
[14] Optimization terminated successfully.
      Current function value: 0.104391
      Iterations 11

[15] print(result.summary())

      Logit Regression Results
=====
Dep. Variable: 진단[악성] No. Observations: 565
Model: Logit Df Residuals: 555
Method: MLE Df Model: 9
Date: Sat, 02 Mar 2024 Pseudo R-squ.: 0.8413
Time: 12:43:11 Log-Likelihood: -58.981
converged: True LL-Null: -371.75
Covariance Type: nonrobust LLR p-value: 6.918e-129
=====

            coef    std err          z      P>|z|      [0.025      0.975]
Intercept -20.9352   14.824     -1.412     0.158    -49.989     8.118
반지름    0.4226    4.051      0.104     0.917     -7.517     8.362
질감      0.4805    0.081      5.906     0.000      0.321     0.640
주변부    -0.3373    0.536     -0.629     0.529     -1.388     0.713
크기       0.0370    0.019      1.976     0.048     0.000     0.074
평활도    151.2655   31.354     4.824     0.000     89.812    212.719
조밀성    20.5338   22.419      0.916     0.360     -23.406    64.473
오른       27.6345   7.220      3.827     0.000     13.483    41.786
대칭      22.8285   12.195     1.872     0.061     -1.073    46.730
프랙탈   -173.2395  98.940     -1.751     0.080    -367.158   20.679
=====

Possibly complete quasi-separation: A fraction 0.32 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.

▼ 4.2 odds

[16] print("===== 계수 =====")
print(result.params)
print("\n")
print("===== odds =====")
print(np.exp(result.params))

===== 계수 =====
  ✓ 0초 오후 9:43에 완료됨
```

4.Logistic Regression

```
▼ 4.2 odds
[16]: print("===== 계수 =====")
print(result.params)
print("\n")
print("===== odds =====")
print(np.exp(result.params))

===== 계수 =====
Intercept      -20.935205
반지름          0.422618
질감            0.480465
주변부          -0.337342
크기             0.036962
평활도          151.265482
조밀성          20.533808
오목             27.634471
대칭             22.828524
프랙탈          -173.239492
dtype: float64

===== odds =====
Intercept      8.090137e-10
반지름          1.525951e+00
질감            1.616826e+00
주변부          7.136646e-01
크기             1.037653e+00
평활도          4.940425e+65
조밀성          8.274079e+08
오목             1.003456e+12
대칭             8.209221e+09
프랙탈          5.794882e-76
dtype: float64

▼ 5. 가정검정


- https://ethanweed.github.io/pythonbook/05.04-regression.html#regressionnormality
- 잔차의 등분산성: Breusch-Pagan
- 잔차의 정규성: Jarque-Bera, Omnibus(D'Angostino's test)

✓ 0초 오후 9:43에 완료됨
```

5.가정검정

5.가정검정

- <https://ethanweed.github.io/pythonbook/05.04-regression.html#regressionnormality>
- 잔차의 등분산성: Breusch-Pagan
- 잔차의 정규성: Jarque-Bera, Omnibus(D'Angostino's test)
- 독립성(자기상관): Durbin-Watson
- 다중공선성(VIF): Cond. No

5.1 다중 공선성

- VIF 10이상 삭제

```
[17] from statsmodels.stats.outliers_influence import variance_inflation_factor  
  
vif = pd.DataFrame()  
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.values.shape[1])]  
vif["features"] = X.columns  
print(vif.round(1))
```

	VIF Factor	features
0	1193.8	Intercept
1	1502.3	반지름
2	1.2	질감
3	1836.4	주변부
4	56.2	크기
5	2.2	평활도
6	21.5	조밀성
7	8.0	오목
8	1.8	대칭
9	6.4	프랙탈

✓ 0초 오후 9:43에 완료됨



4.Logistic Regression

4.Logistic Regression

- <https://www.statsmodels.org/stable/examples/notebooks/generated/ols.html>
- 수치형 + 범주형
- dmatrix 사용

4.1 회귀분석

다중공선성 제거후 다시 실행

```
[19] # columns = ['반지름', '질감', '주변부', '크기', '평활도', '조밀성', '오목', '대칭', '프랙탈']

# 다중공선 제거: 반지름, 주변부
columns = ['질감', '크기', '평활도', '조밀성', '오목', '대칭', '프랙탈']

formula = "진단 ~ " + " + ".join(columns)
formula

'진단 ~ 질감 + 크기 + 평활도 + 조밀성 + 오목 + 대칭 + 프랙탈'
```

✓ [20] # dmatrix 이용
from patsy import dmatrices

y, X = dmatrices(formula,
 data = lr_df,
 return_type = 'dataframe')

✓ [21] X.head()

	Intercept	질감	크기	평활도	조밀성	오목	대칭	프랙탈
0	1.0	10.4	1001.0	0.118	0.278	0.300	0.242	0.079
1	1.0	17.8	1326.0	0.085	0.079	0.087	0.181	0.057
2	1.0	21.3	1203.0	0.110	0.160	0.197	0.207	0.060
3	1.0	14.4	566.3	0.098	0.081	0.067	0.189	0.058

scroll...

✓ 0초 오후 9:45에 완료됨

4.Logistic Regression

```

4      0.0

[23] model = sm.Logit(y, X)  # 모델 생성
result = model.fit()  # 모델 실행

Optimization terminated successfully.
    Current function value: 0.105996
    Iterations 11

[24] print(result.summary())

Logit Regression Results
=====
Dep. Variable: 진단[악성] No. Observations: 565
Model: Logit Df Residuals: 557
Method: MLE Df Model: 7
Date: Sat, 02 Mar 2024 Pseudo R-squ.: 0.8390
Time: 12:45:47 Log-Likelihood: -59.854
converged: True LL-Null: -371.75
Covariance Type: nonrobust LLR p-value: 1.835e-130
=====
            coef  std err      z   P>|z|   [0.025   0.975]
Intercept -34.9140   7.612   -4.587   0.000  -49.833  -19.995
질감       0.4755   0.080    5.926   0.000    0.318    0.633
크기        0.0170   0.003    5.330   0.000    0.011    0.023
평활도     152.4918  30.847    4.943   0.000   92.033   212.951
조밀성      0.4453  12.814    0.035   0.972  -24.669   25.560
오록      29.0125   6.784    4.276   0.000   15.716   42.309
대칭      23.3006  12.354    1.886   0.069   -0.912   47.513
프랙탈    -119.8613  90.354   -1.327   0.185  -296.952   57.230
=====

Possibly complete quasi-separation: A fraction 0.31 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.

▼ 4.2 odds

[25] print("===== 계수 =====")
print(result.params)
print("\n")
===== 계수 =====
                params
Intercept      -34.9140
질감           0.4755
크기            0.0170
평활도         152.4918
조밀성          0.4453
오록            29.0125
대칭            23.3006
프랙탈         -119.8613

```

4.Logistic Regression

```
[24] Possibly complete quasi-separation: A fraction 0.31 of observations can be perfectly predicted. This might indicate that there is complete quasi-separation. In this case some parameters will not be identified.

▽ 4.2 odds

[25] print("===== 계수 =====")
print(result.params)
print("\n")
print("===== odds =====")
print(np.exp(result.params))

===== 계수 =====
Intercept      -34.913962
질감          0.475503
크기          0.016976
평활도        152.491779
조밀성        0.445319
오목         29.012504
대칭         23.300555
프랙탈       -119.861261
dtype: float64

===== odds =====
Intercept     6.871619e-16
질감          1.608823e+00
크기          1.017121e+00
평활도        1.683984e+66
조밀성        1.560989e+00
오목          3.980800e+12
대칭          1.316142e+10
프랙탈       8.808781e-53
dtype: float64
```

▽ 5. 가정검정

- <https://ethanweed.github.io/pythonbook/05.04-regression.html#regressionnormality>
- 잔차의 등분산성: Breusch-Pagan
- 잔차의 정규성: Jarque-Bera, Omnibus(D'Angostino's test)

ed.github.io%2Fpythonbook%2F05.04-regre... 오후 9:45에 완료됨

5.가정검정

▼ 5.가정검정

- <https://ethanweed.github.io/pythonbook/05.04-regression.html#regressionnormality>
- 잔차의 등분산성: Breusch-Pagan
- 잔차의 정규성: Jarque-Bera, Omnibus(D'Angostino's test)
- 독립성(자기상관): Durbin-Watson
- 다중공선성(VIF): Cond. No

▼ 5.1 다중 공선성

- VIF 10이상 삭제

```
✓ 0초 ▶ from statsmodels.stats.outliers_influence import variance_inflation_factor  
  
vif = pd.DataFrame()  
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.values.shape[1])]  
vif["features"] = X.columns  
print(vif.round(1))
```

	VIF Factor	features
0	392.3	Intercept
1	1.2	질감
2	4.7	크기
3	2.2	평활도
4	8.6	조밀성
5	7.4	오록
6	1.8	대칭
7	4.3	프랙탈

[] + 코드 + 텍스트

✓ 0초 오후 9:45에 완료됨



Logistic Regression

- ❖ 유방암 진단에 영향을 주는 변수로 질감, 크기, 평활도, 오목으로 나타났다. 질감, 크기, 평활도, 오목이
클수록 악성일 확률이 높은 것으로 나타났다.

Predictor	Estimate	SE	Z	p	Odds ratio
Intercept	-34.91	7.61	-4.59	<.001	0.000
질감	0.48	0.08	5.93	<.001	1.610
크기	0.02	0.00	5.33	<.001	1.017
평활도	152.49	30.85	4.94	<.001	1.684E+66
조밀성	0.45	12.81	0.03	0.97	1.561
오목	29.01	6.78	4.28	<.001	3.98E+12
대칭	23.30	12.35	1.89	0.06	1.32E+10
프랙탈	-119.86	90.35	-1.33	0.19	0.000

연습문제

연습문제1

❖ 문제의 정의

- G의류에서는 새로운 옷을 디자인하려고 하는데, 키와 몸무게가 어떤 관계가 있는지를 보고자 한다. 키와 몸무게는 상관관계가 있는가?
- 11_2.weight.csv

몸무게	키
72	176
72	172
70	182
43	160
48	163
54	165
51	168
52	163
73	182
45	148
60	170
62	166
64	172
47	160
51	163
74	170
88	182
64	174
56	164
56	160
62	178
70	175
73	173
82	188
75	180

연습문제2

❖ 문제의 정의

- G병원에서 혈액의 콜레스테롤 수치를 이용해 중성지방 수치를 알아보고자 한다.
- 1. 콜레스테롤 수치와 중성지방 수치가 관련이 있는가?
- 2. 회귀분석의 가정을 만족하는가?
- 3. 콜레스테롤과 중성지방 사이의 관련성을 회귀식으로 추정하세요.
- 12_2.NeutralFat.csv

$$\text{중성지방} = b_0 + b_1 \text{콜레스테롤}$$

col	fat
108.4	44.1
110.4	40.9
127.1	44.3
128.2	50.5
131.7	77.3
134.8	52.5
136.7	73.9
142.4	50.5
142.5	52.3
143.6	41.7
144.5	74.2
145.0	112.7
145.5	73.9
146.0	162.3
148.4	67.9
150.0	45.1
153.9	137.3
154.3	123.0
154.7	57.9
158.3	51.6
162.9	63.4
164.1	161.1
164.1	132.5
164.3	49.9
164.8	105.1

연습문제3

❖ 문제의 정의

- Toyota 중고차 가격을 결정하는 모델을 만들고자 한다.
- price: 가격 (유로)
- km: 주행킬로미터 (kilometers driven)
- 1. 주행거리와 중고차 가격은 관련이 있는가?
- 2. 회귀분석의 가정을 만족하는가?
- 3. 주행거리와 중고차 가격 사이의 관련성을 회귀식으로 추정하세요.
- 12_3.toyota.csv

$$price = b_0 + b_1 km$$

id	price	km
1	12900	23000
2	8500	61977
3	7750	69000
4	9799	59000
5	9950	57948
6	13950	13748
7	9695	43000
8	9000	61165
9	8950	38900
10	11895	39439
11	10850	47768
12	11895	27170
13	13900	22000
14	7750	65400
15	7995	60724
16	13450	17003
17	6550	72328
18	7950	72222
19	9950	58000
20	11950	32781
21	9500	56214
22	8750	56307
23	10950	40214
24	8450	49291
25	6950	66000

연습문제4

❖ 문제의 정의

- K대학에서는 재학생을 대상으로 교육수요자 만족도조사를 실시하였다.
- 1. 교양만족도, 전공만족도, 비교과만족도는 전체 만족도에 영향을 주는가?
- 2. 회귀분석의 가정을 만족하는가?
- 3. 가장 중요한 변수는 무엇인가?
- 07_3.Education.omv를 복사해서 사용하고 결과는 13_2.Education.omv로 저장하세요.

학..	학년	교양만족도	전공만족도	비교과만...
빅경	1학년	47.6	40.5	40.0
빅경	1학년	33.3	35.7	33.3
빅경	1학년	50.0	52.4	50.0
빅경	1학년	35.7	28.5	40.0
빅경	1학년	54.7	92.8	43.3
빅경	1학년	39.3	53.6	55.0
빅경	1학년	46.4	46.4	45.0
빅경	1학년	42.9	67.9	35.0
사복	1학년	42.9	50.0	25.0
사복	1학년	32.1	28.6	30.0
사복	1학년	50.0	50.0	50.0
사복	1학년	50.0	50.0	50.0
사복	1학년	78.6	64.3	70.0
사복	1학년	42.9	53.6	25.0
간호	1학년	42.8	73.8	46.6
간호	1학년	80.5	90.5	76.6
간호	1학년	50.0	46.4	25.0
간호	1학년	25.0	32.1	40.0
간호	1학년	39.3	50.0	35.0
간호	1학년	39.3	57.1	20.0
간호	1학년	28.6	39.3	25.0
간호	1학년	50.0	75.0	45.0
간호	1학년	85.7	92.9	20.0
간호	1학년	25.0	46.4	30.0
간호	1학년	50.0	50.0	50.0

연습문제5

❖ 문제의 정의

- K기업의 인사담당인 이부장은 신체적 건강과 심리적건강, 조직몰입, 이직경험이 이직의도에 영향을 준다고 보고, 이들간의 인과관계를 연구하고자 한다
- 이직의도: 1:없음, 2:있음
- 이직경험: 1:없음, 2:있음
- 1. 신체적 건강과 심리적건강, 조직몰입, 이직경험이 이직의도에 영향을 주는가?
- 2. 회귀분석의 가정을 만족하는가?
- 14_2.HR.csv

id	이직의도	신체적건강	심리적건강
1	1	43	18
2	1	54	27
3	1	60	30
4	1	57	17
5	1	60	30
6	2	42	27
7	1	48	21
8	1	46	18
9	1	57	30
10	1	59	24
11	1	47	23
12	1	36	18
13	1	43	24
14	2	46	17
15	2	43	23
16	1	52	20
17	1	51	27
18	2	49	23
19	2	39	21
20	2	36	16
21	1	41	24
22	1	58	30
23	1	57	30
24	1	40	21
25	1	46	24

연습문제6

❖ 문제의 정의

- 유니버설 은행에서는 target marketing을 활용한 캠페인을 진행하려고 한다.
- 대출 제안에 대한 수락(1)에 영향을 미치는 변수는 무엇인가?
- 대출의도: 0:거절, 1:수락
- 카드보유유무: 0: 없음, 1:있음
- 1. 나이, 경력 등은 대출수락(1)에 영향을 주는가?
- 2. 회귀분석의 가정을 만족하는가?
- 14_3.UniversalBank.csv

id	대출의도	나이	경력	수입
1	0	25	1	49
2	0	45	19	34
3	0	39	15	11
4	0	35	9	100
5	0	35	8	45
6	0	37	13	29
7	0	53	27	72
8	0	50	24	22
9	0	35	10	81
10	1	34	9	180
11	0	65	39	105
12	0	29	5	45
13	0	48	23	114
14	0	59	32	40
15	0	67	41	112
16	0	60	30	22
17	1	38	14	130
18	0	42	18	81
19	1	46	21	193
20	0	55	28	21
21	0	56	31	25
22	0	57	27	63
23	0	29	5	62
24	0	44	18	43
25	0	36	11	152

V. 통계적 분석과 기계학습 분석 방법 비교

통계적 방법과 기계학습적 방법은 어떻게 다른가요?

통계적 방법과 기계학습적 방법의 차이

LGE Internal Use Only

❖ 통계적 방법과 기계학습적 방법의 차이

구분	통계적	기계학습적
데이터	<ul style="list-style-type: none">표본 데이터(소수)	<ul style="list-style-type: none">대용량 데이터(빅데이터)
가정	<ul style="list-style-type: none">통계적 가정 필요모수통계(정규분포)비모수통계회귀분석 가정(선형성, 등분산성)	<ul style="list-style-type: none">없음
방법	<ul style="list-style-type: none">가설검정(귀무가설)	<ul style="list-style-type: none">데이터를 통한 스스로 학습
검정	<ul style="list-style-type: none">유의수준(p-value)	<ul style="list-style-type: none">예측 정확도
특징	<ul style="list-style-type: none">모든 데이터 사용 → overfitting, underfitting 문제설명	<ul style="list-style-type: none">Training, Validation, Testing 구분parameter tuning 가능여러 모델 비교 가능예측 및 분류의 정확도 → 변수와의 관계 설명이 어려움

- ❖ 최소제곱법(Method of least Squares)

- 적합된 회기식에 의한 예측치 \hat{y}_i 와 관찰치 y_i 의 차이인 잔차들의 제곱의 합이 최소가 되도록 회귀계수를 추정하는 방법(미분)

$$\sum_{i=1}^n (e_i)^2 = \sum_{i=1}^n (y_i - (b_0 + b_1 x_i))^2 \quad \begin{array}{l} \xrightarrow{\hspace{1cm}} \frac{\partial D}{\partial b_0} = -2 \sum_{i=1}^n (y_i - (b_0 + b_1 x_i)) = 0 \\ \xrightarrow{\hspace{1cm}} \frac{\partial D}{\partial b_1} = -2 \sum_{i=1}^n x_i(y_i - (b_0 + b_1 x_i)) = 0 \end{array}$$

$$b_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} = \frac{S_{xy}}{S_{xx}} \quad \longrightarrow \quad \hat{\beta}_1 \qquad \qquad \beta_1 \rightarrow b_1 \rightarrow \hat{\beta}_1$$

$$b_0 = \bar{y} - b_1 \bar{x} \quad \longrightarrow \quad \hat{\beta}_0 \qquad \qquad \beta_0 \rightarrow b_0 \rightarrow \hat{\beta}_0$$

❖ 회귀계수(β) 검정

- 귀무가설(H_0): 두 변수간에는 인과관계(영향력)가 없다.

$$H_0: \beta_1 = 0$$

- 연구가설(H_1): 두 변수간에는 인과관계(영향력) 가 있다.

$$H_1: \beta_1 \neq 0$$

- 검정통계량

$$\frac{\hat{\beta}_1 - \beta_1}{\sqrt{\frac{MSE}{S_{xx}}}} \sim t(n-2)$$

$$\frac{7.429 - 0}{0.229} = 32.415 > 1.980$$

Model Coefficients - 가격

Predictor	Estimate	SE	t	p
Intercept	93736.519	2143.971	43.721	< .001
연면적	7.429	0.229	32.415	< .001

❖ 회귀분석 가정검정

- ANOVA와 같이 잔차검정
- 차이점- 회귀식을 기준으로 한 잔차를 이용
- 회귀모형식

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \varepsilon_i$$

$$\varepsilon_{ij} \rightarrow e_{ij} \sim N(0, \sigma^2)$$

$$* ANOVA: Y_{ij} = \mu + (\mu_i - \mu) + \varepsilon_{ij}$$

전체평균 + 처리효과 + 측정오차

- 등분산성: 종속변수의 분산은 독립변수의 값에 관계없이 동일해야 한다.
- 정규성: 오차(잔차)는 정규분포를 이루어어야 한다.
- 독립성: 오차(잔차)는 서로 독립적이어야 한다

회귀분석(통계적 방법)

LGE Internal Use Only

계수^a

모형	비표준 계수		표준 계수	t	유의수준
	B	표준 오차			
1 (상수)	-1289334.699	69245.265		-18.620	.000
언면적	1.768	.239	.112	7.390	.000
품질	11480.177	718.681	.263	15.974	.000
상태	6202.797	598.978	.133	10.356	.000
건축년도	474.064	31.470	.264	15.064	.000
리모델링년도	152.139	38.066	.059	3.997	.000
지하면적	21.951	2.570	.152	8.542	.000
차고면적	29.316	3.752	.105	7.813	.000
면적_1층	47.429	3.076	.295	15.421	.000
면적_2층	43.865	1.791	.339	24.493	.000
주거_2가구변경	-9388.093	4106.264	-.025	-2.286	.022
주거_듀플렉스	-22407.373	3402.774	-.074	-6.585	.000
주거_타운젠트바깥쪽	-1725.418	2576.710	-.009	-.670	.503
주거_타운젠트안쪽	-11032.945	3578.707	-.038	-3.083	.002
판매_법원증서	-8284.469	3330.622	-.029	-2.487	.013
조건_안류및공매도	-7954.314	2275.568	-.040	-3.496	.000

a. 종속 변수: 가격

회귀분석(통계적 방법)

LGE Internal Use Only

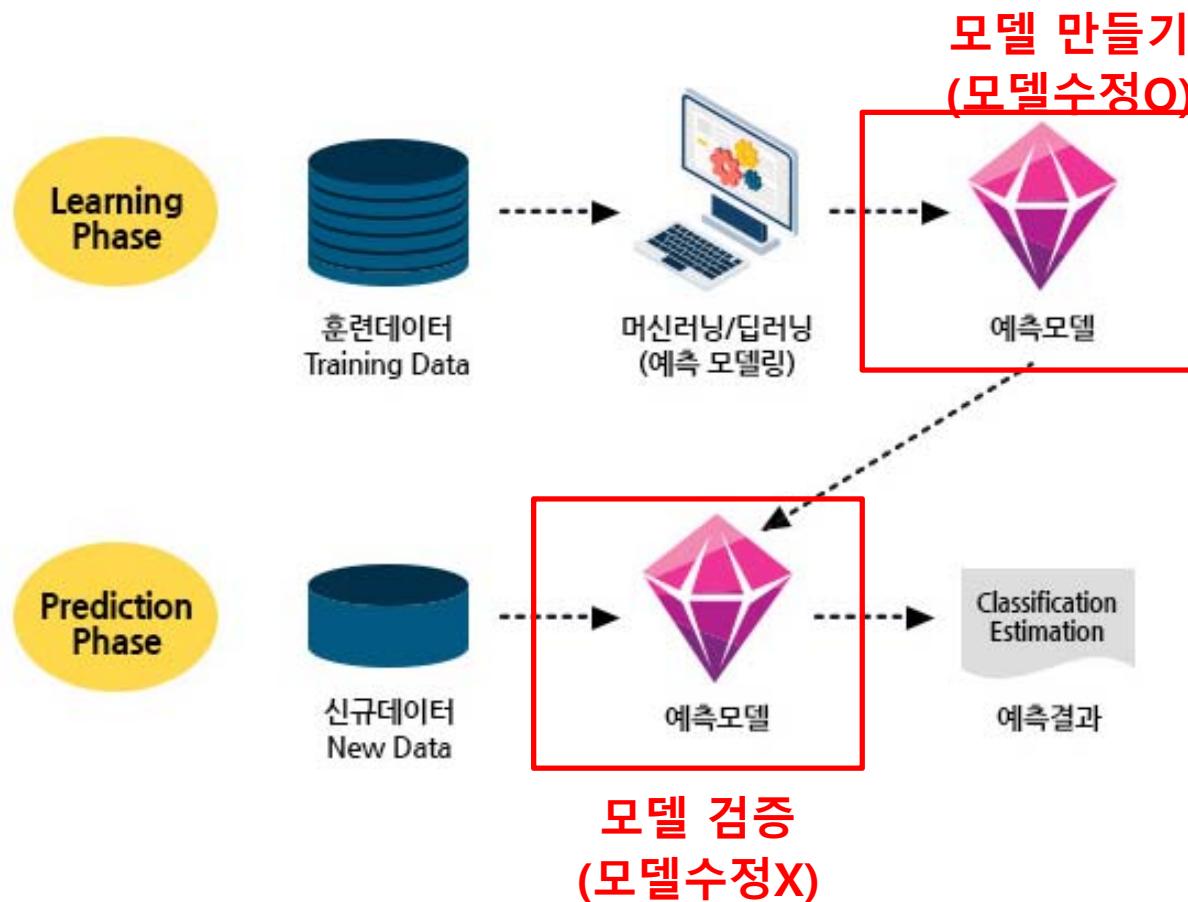
❖ 데이터수에 따른 유의수준 변화

	n= 200			n = 690			n= 1300		
	표준 계수	t	유의수준	표준 계수	t	유의수준	표준 계수	t	유의수준
(상수)		-7.732	.000		-14.775	.000		-18.620	.000
연면적	.099	2.951	.004	.103	5.041	.000	.112	7.390	.000
품질	.271	7.365	.000	.271	12.332	.000	.263	15.974	.000
상태	.109	4.141	.000	.142	8.245	.000	.133	10.356	.000
건축년도	.286	8.153	.000	.273	11.356	.000	.264	15.064	.000
리모델링년도	.054	1.726	.086	.065	3.252	.001	.059	3.997	.000
지하면적	.129	2.879	.004	.144	5.744	.000	.152	8.542	.000
차고면적	.120	3.973	.000	.094	5.180	.000	.105	7.813	.000
면적_1층	.280	6.144	.000	.304	11.435	.000	.295	15.421	.000
면적_2층	.346	11.533	.000	.340	18.391	.000	.339	24.493	.000
주거_2가구변경	-.014	-.596	.552	-.022	-1.470	.142	-.025	-2.286	.022
주거_듀플렉스	-.045	-1.844	.067	-.072	-4.828	.000	-.074	-6.585	.000
주거_타운젠트바깥쪽	-.045	-1.697	.091	-.026	-1.469	.142	-.009	-.670	.503
주거_타운젠트안쪽	-.078	-2.814	.005	-.032	-1.913	.056	-.038	-3.083	.002
판매_법원증서	.017	.662	.509	-.027	-1.694	.091	-.029	-2.487	.013
조건_압류및공매도	-.022	-.863	.389	-.046	-2.981	.003	-.040	-3.496	.000

❖ 통계적 방법과 기계학습 방법의 차이

구분	통계적	기계학습적
데이터	<ul style="list-style-type: none"> 표본 데이터(소수) 	<ul style="list-style-type: none"> 대용량 데이터(빅데이터)
가정	<ul style="list-style-type: none"> 통계적 가정 필요 모수통계(정규분포) 비모수통계 회귀분석 가정(선형성, 등분산성) 	<ul style="list-style-type: none"> 없음
방법	<ul style="list-style-type: none"> 가설검정(귀무가설) 	<ul style="list-style-type: none"> 데이터를 통한 스스로 학습
검정	<ul style="list-style-type: none"> 유의수준(p-value) 	<ul style="list-style-type: none"> 예측 정확도
특징	<ul style="list-style-type: none"> 모든 데이터 사용 → overfitting, underfitting 문제 설명 	<ul style="list-style-type: none"> Training, Validation, Testing 구분 parameter tuning 가능 여러 모델 비교 가능 예측 및 분류의 정확도 → 변수와의 관계 설명이 어려움

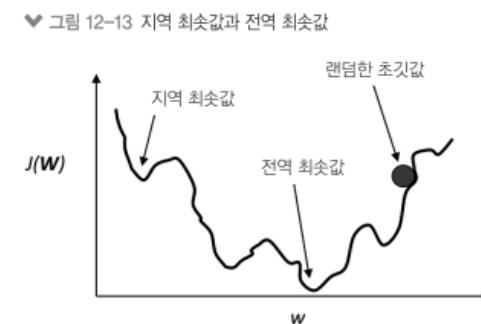
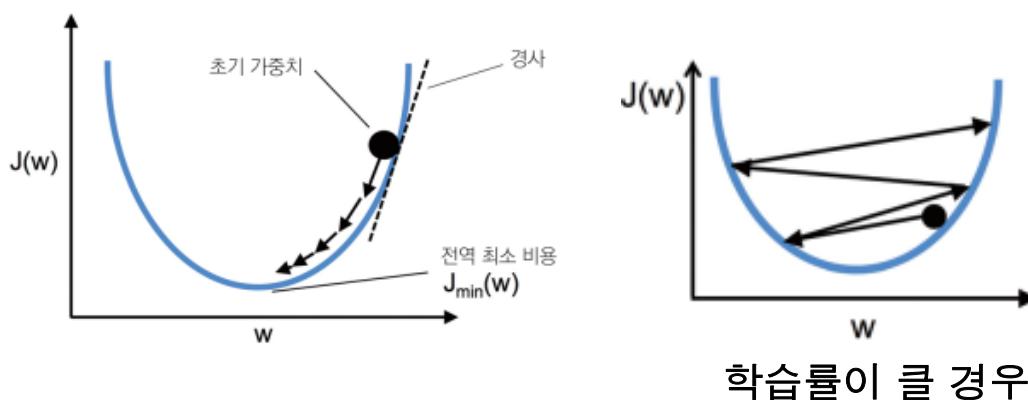
❖ 훈련과 검증



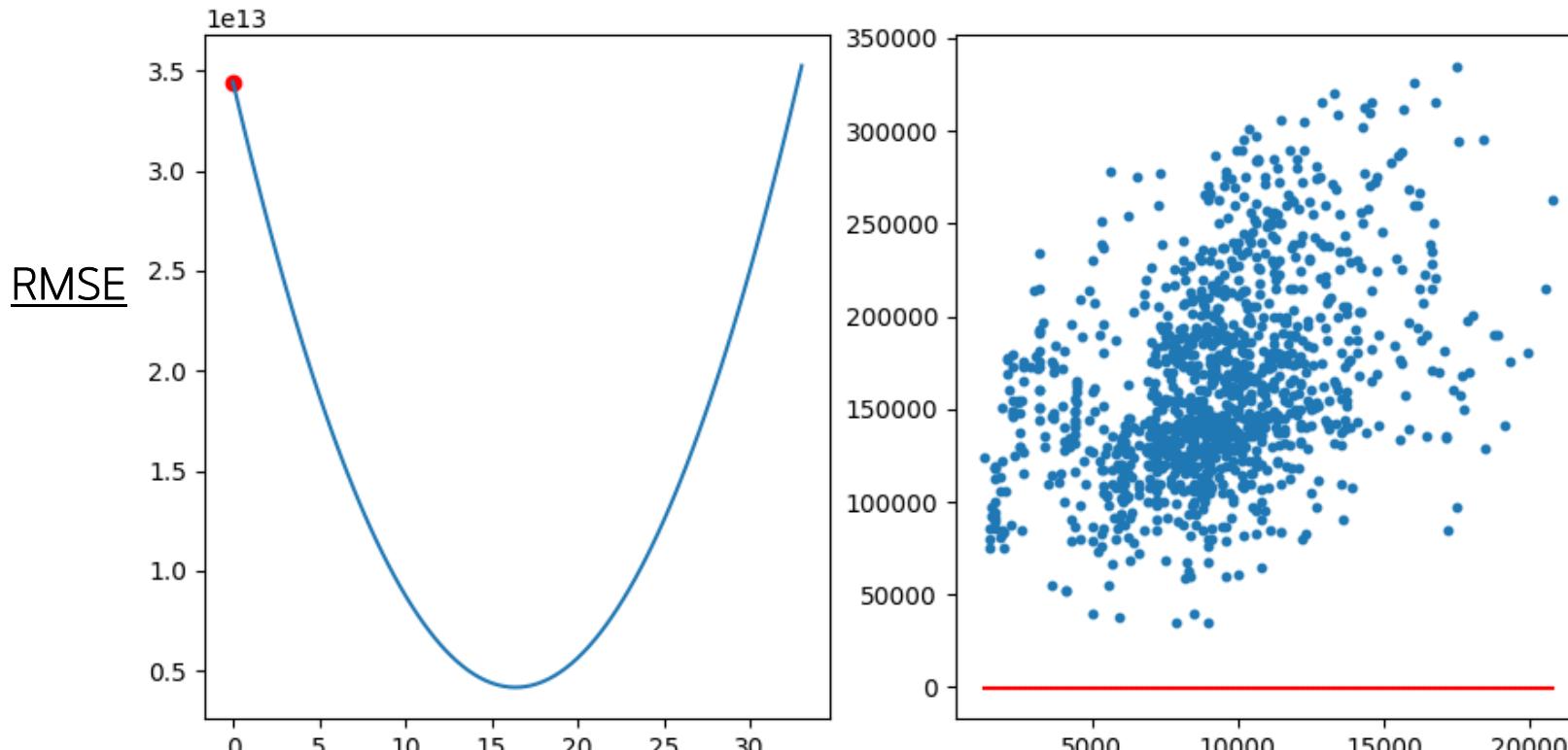
출처: http://spss.datasolution.kr/solution/solution_bigdata.asp

- ❖ 경사하강법(Gradient descent)

- 미분을 이용하여 기울기가 최소인 지점 찾기
- 학습률(Learning Rate)
 - 새로운 정보를 얼마나 반영할지를 조절
 - 학습률(Learning Rate)을 이용하여 점진적으로 변화량 조절
- 모멘텀(Momentum): 어느 정도 기존의 방향을 유지할 것인지 조정



출처: 세巴斯찬 라시카 외, “머신러닝 교과서 with 파이썬, 사이킷런, 텐서플로,” 2019

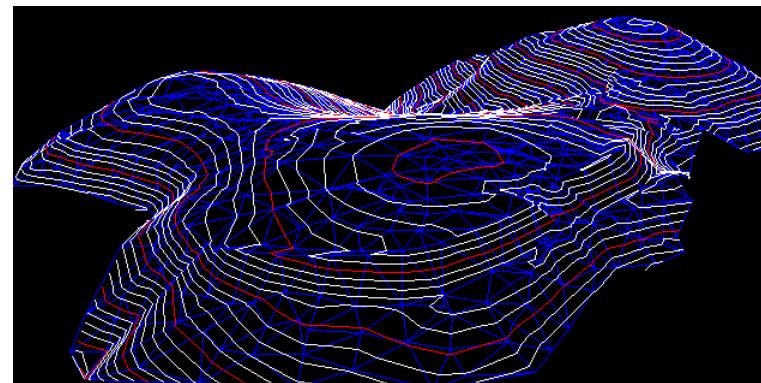
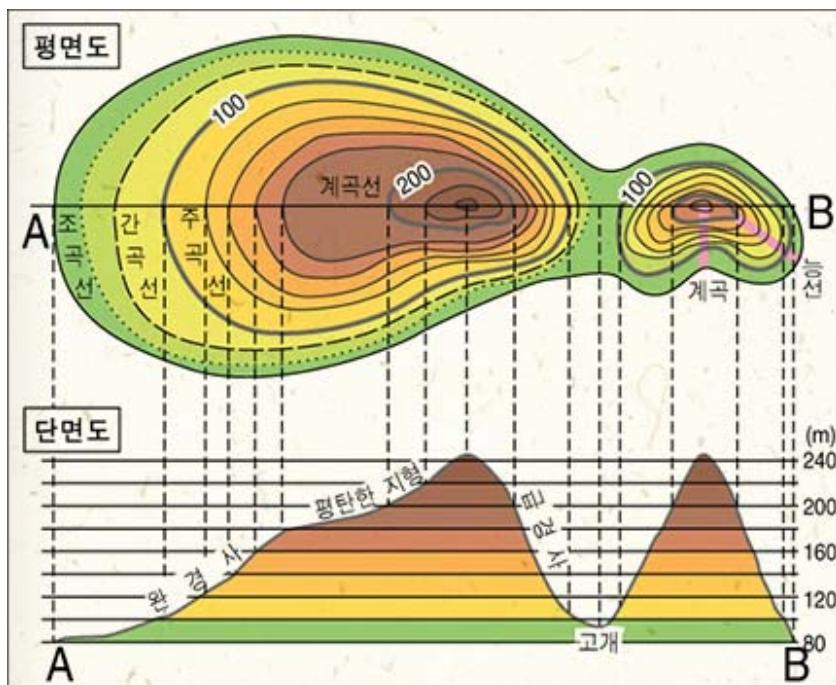


연면적(b1)

회귀분석(기계학습적 방법)

LGE Internal Use Only

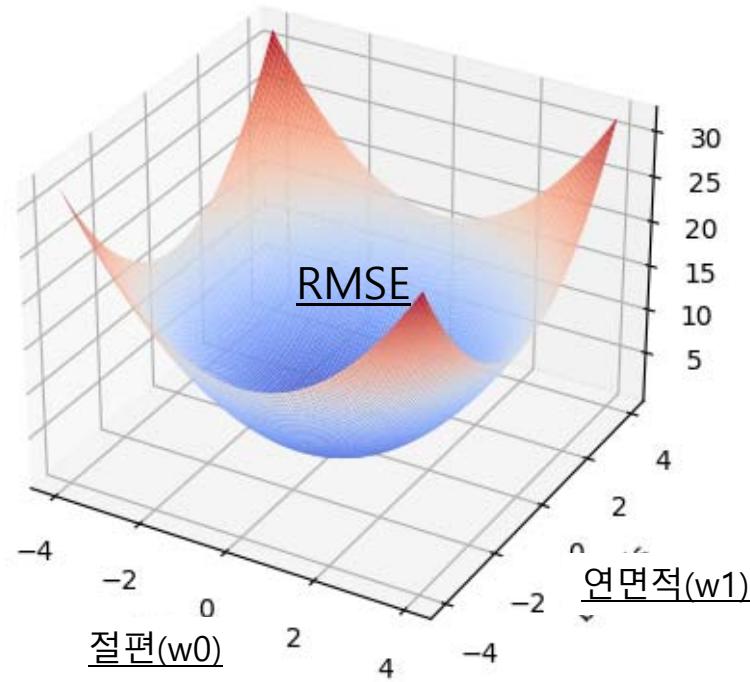
- ❖ 변수가 2개일때 3차원 → 2차선 (등고선)



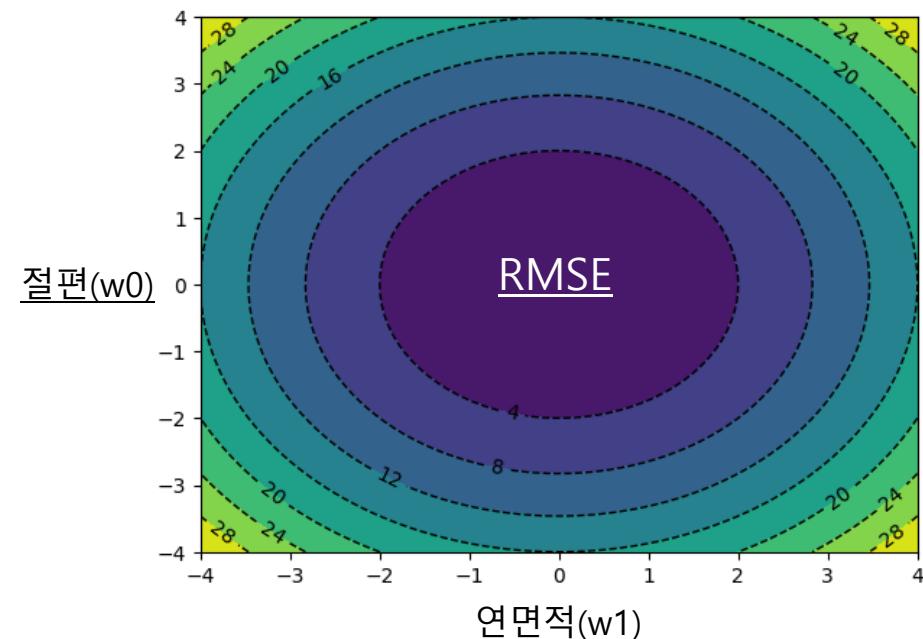
출처: https://www.youtube.com/watch?v=0kns1gXLYg4&list=PLLssT5z_DsK-h9vYZkQkYNWcItqhIRJLN&index=7

회귀분석(기계학습적 방법)

LGE Internal Use Only



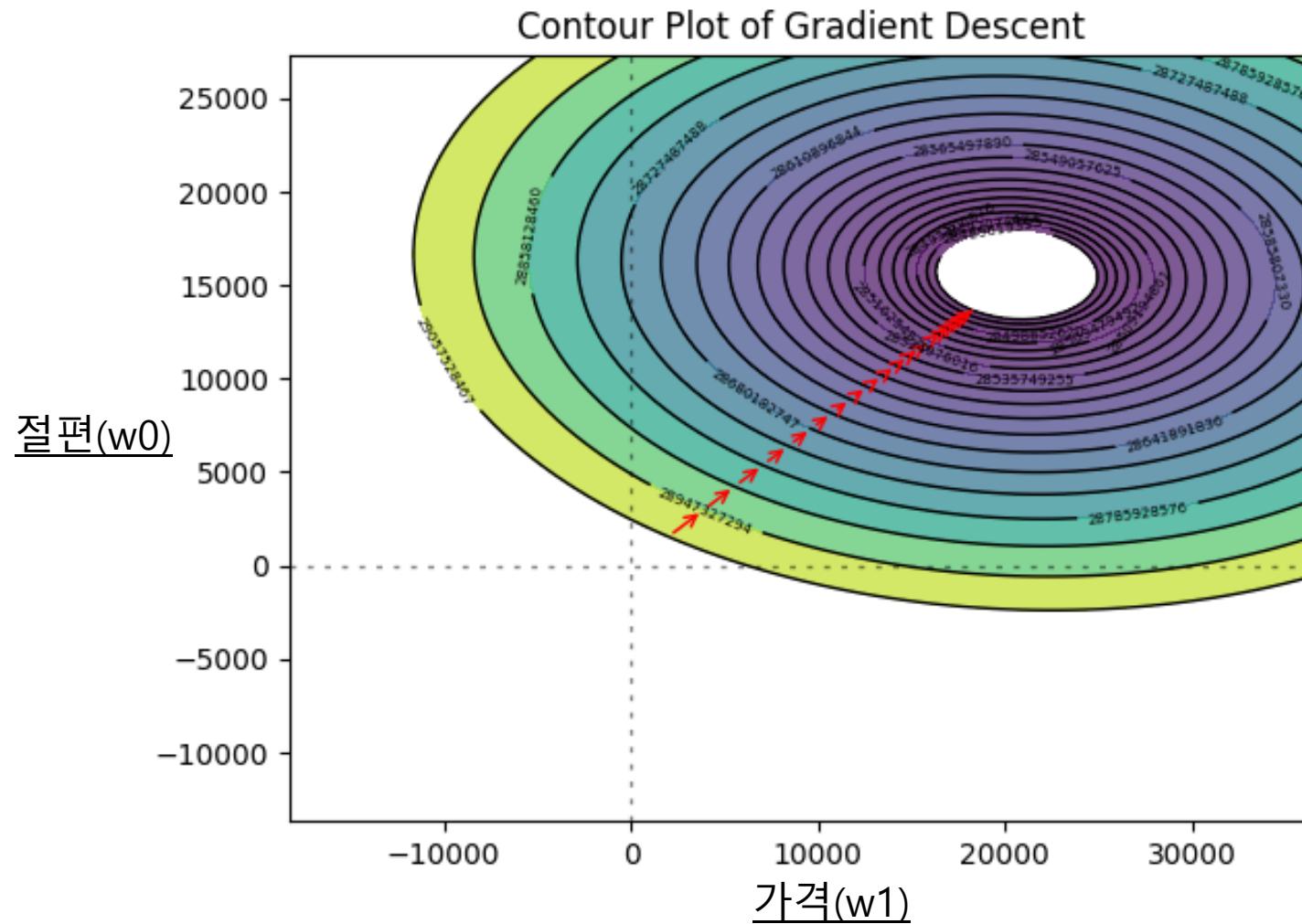
RMSE(3차원)



RMSE(등고선, 2차원)

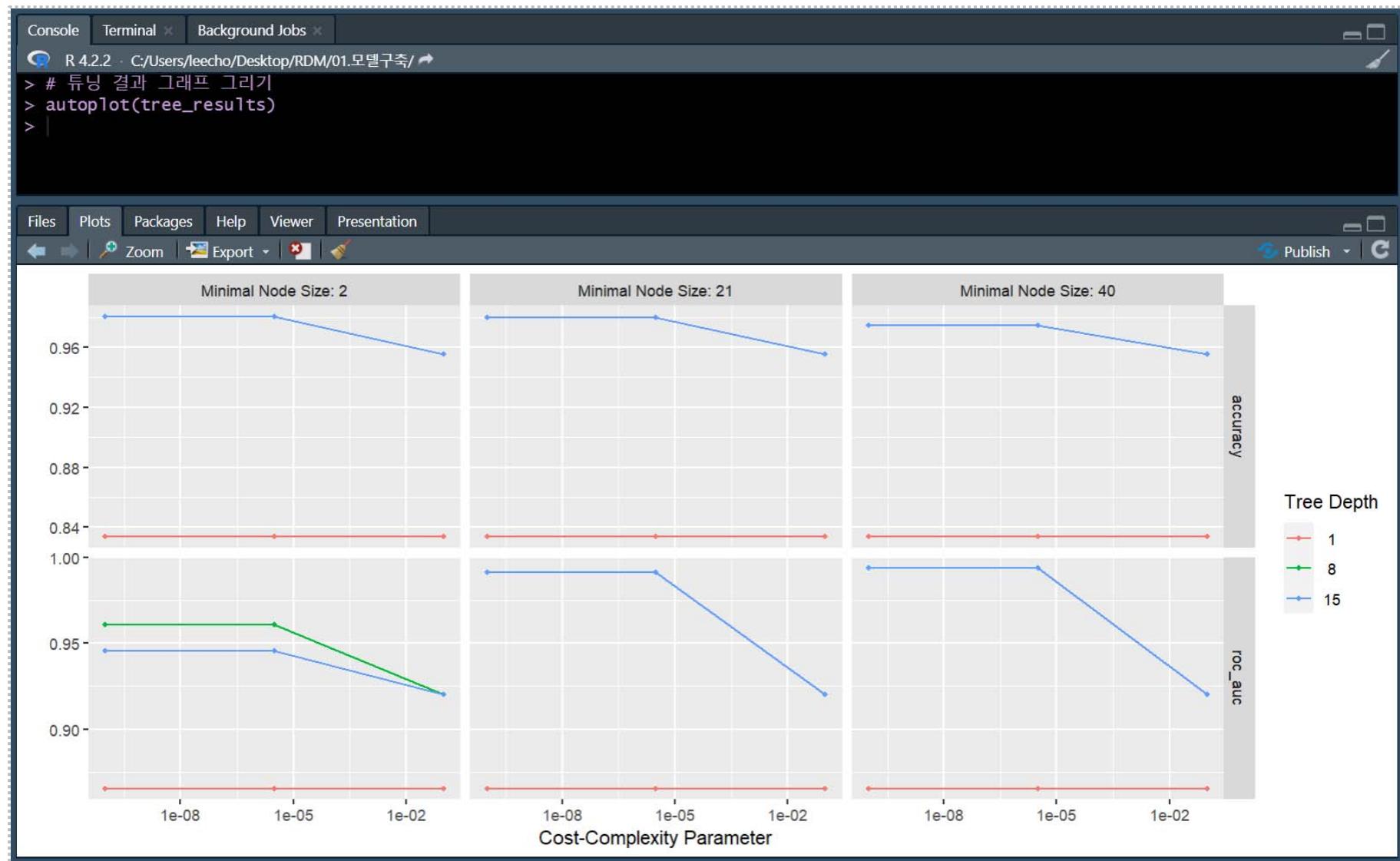
회귀분석(기계학습적 방법)

LGE Internal Use Only



회귀분석(기계학습적 방법)

LGE Internal Use Only



회귀분석(기계학습적 방법)

LGE Internal Use Only

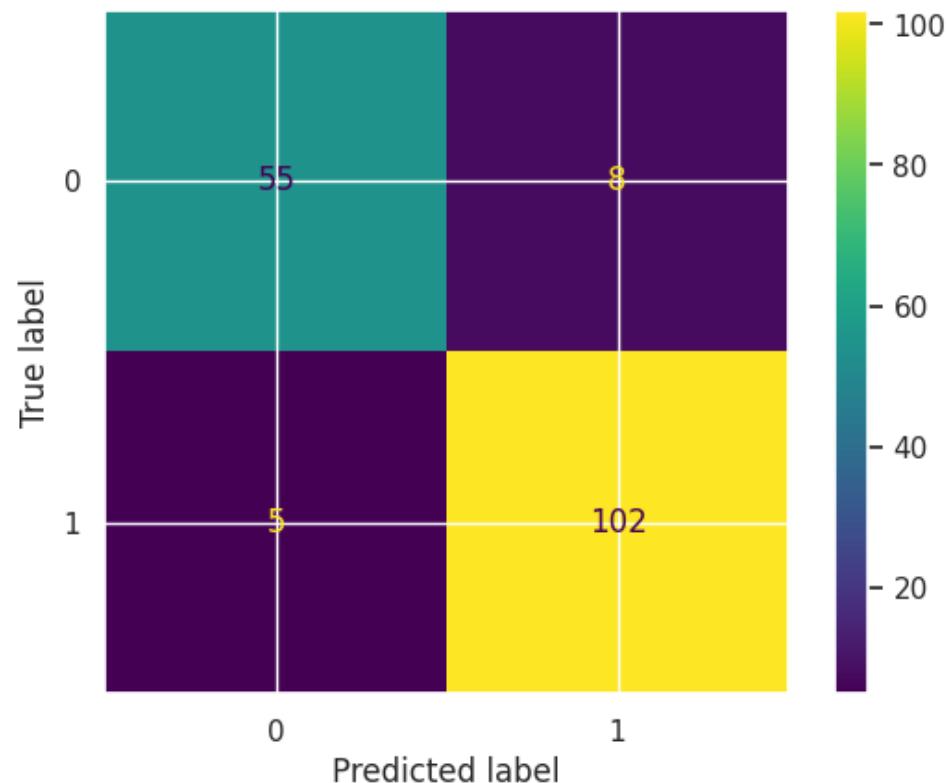
The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** 04_02.Support Vector Machine.ipynb
- Code Cell Content:** A Python script for generating validation curves. It iterates over masks, stacks them, finds the best parameter mask, and then plots validation and training scores against various parameters.
- Plots:** Three validation curves are displayed:
 - Validation Curve (C vs Score):** Shows validation score (blue circles) and training score (orange triangles) as C increases from 0 to 1000. The validation score drops sharply from ~0.998 to ~0.980, while the training score remains high (~0.998).
 - Validation Curve (Gamma vs Score):** Shows validation score (blue circles) and training score (orange triangles) as Gamma increases from 0.0 to 1.0. The validation score decreases from ~0.998 to ~0.980, while the training score increases from ~0.992 to ~0.998.
 - Validation Curve (Kernel vs Score):** Shows validation score (blue circles) and training score (orange triangles) for RBF and linear kernels. The validation score is higher than the training score for both kernels.

❖ 모델검정

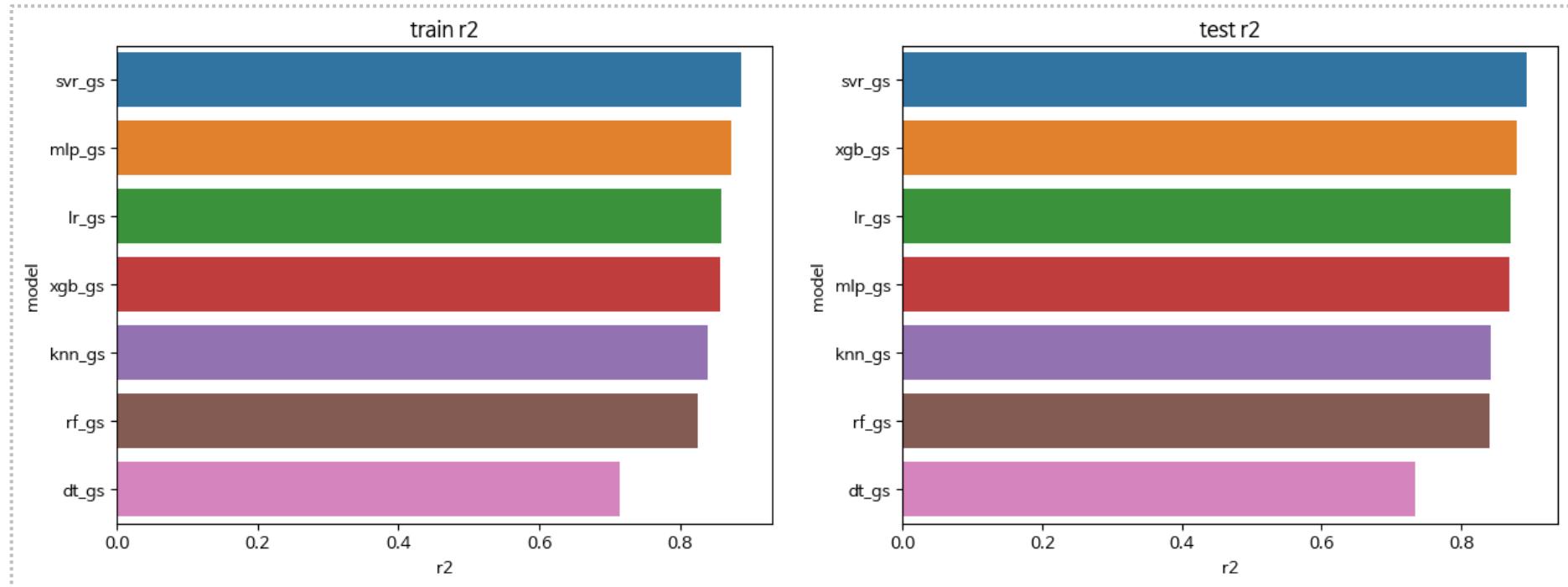
- 수치형: RMSE, R^2
- 범주형: accuracy, ROC

```
> final_pred %>%  
+   metrics(truth = 가격,  
+             estimate = .pred)  
# A tibble: 3 × 3  
  .metric .estimator .estimate  
  <chr>   <chr>      <dbl>  
1 rmse    standard     0.365  
2 mae     standard     0.275  
3 rsq     standard     0.875
```



기계학습(여러 모델 비교): Python

LGE Internal Use Only



수고하셨습니다.