컴퓨터 통계 방법론 HW1

20180490 이재헌

Q1. Ch2-Prob2: Explain whether each scenario is a classification or regression problem, and indicate whether we are most interested in inference or prediction. Finally, provide n and p.

(a) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary.

Answer: This scenario is a regression problem. The goal is to understand how CEO salary (a continuous numerical variable) is affected by various factors like profit, number of employees, and industry. In this case, we are most interested in inference because we want to understand the relationship between CEO salary and these factors, i.e., we want to identify the factors that have a statistically significant impact on CEO salary.
n: The number of data points would be the top 500 firms in the US.
p: The number of predictor variables would be 3 (profit, number of employees, industry) in addition to any potential interaction terms or additional variables that might be included in the analysis.

(b) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

Answer: This scenario is a classification problem. The goal is to predict whether a new product will be a success or a failure, which is a binary outcome (success or failure). Therefore, it's a classification problem. In this case, we are most interested in prediction because we want to use the collected data on similar products to make predictions about the success or failure of the new product.
n: The number of data points would be the 20 similar products.
p: The number of predictor variables would be 13 (price, marketing budget, competition price, and ten other variables), in addition to any potential interaction terms or additional variables that might be included in the analysis.

(c) We are interested in predicting the % change in the USD/Euro exchange rate in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the USD/Euro, the % change in the US market, the % change in the British market, and the % change in the German market.

Answer: This scenario is a regression problem. The goal is to predict the % change in the USD/Euro exchange rate, which is a continuous numerical variable, based on the weekly changes in the world stock markets. Therefore, it's a regression problem. In this case, we are most interested in prediction because we want to make forecasts about the exchange rate based on the observed changes in the

stock markets.

n: The number of data points would be the number of weeks in 2012 for which data is collected.

p: The number of predictor variables would be 3 (the % change in the US market, the % change in the British market, and the % change in the German market).

Q2. Ch2-Prob3: We now revisit the bias-variance decomposition.

(a) Provide a sketch of typical (squared) bias, variance, training error, test error, and Bayes (or irreducible) error curves, on a single plot, as we go from less flexible statistical learning methods towards more flexible approaches. The x-axis should represent the amount of flexibility in the method, and the y-axis should represent the values for each curve. There should be five curves. Make sure to label each one.

Answer: The expected test MSE for a given x_0 can always be decomposed into the sum of three fundamental quantities:

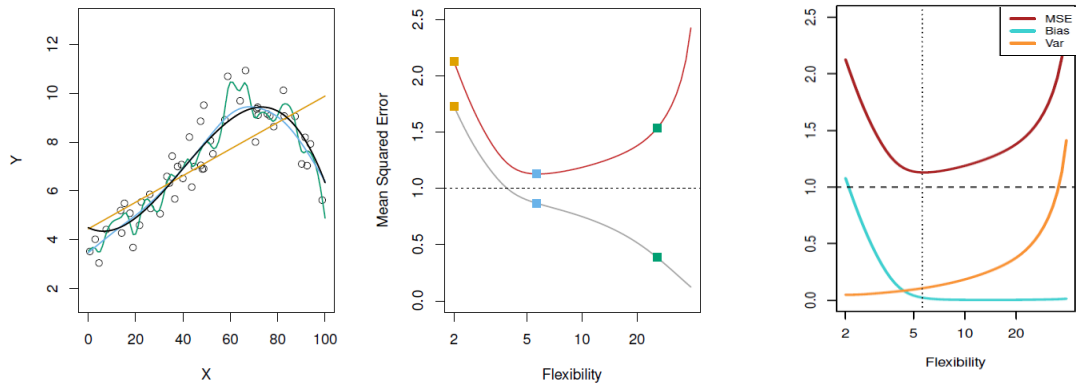
$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon). \qquad (2.7)$$

1. *The variance of* $\hat{f}(x_0)$, 2. *The squared bias of* $\hat{f}(x_0)$, 3. *The variance of the error term* ϵ.

Variance refers to the amount by which $\hat{f}$ would change if we estimated it using a different training data set. Different training data sets will result in a different $\hat{f}$. In general, more flexible statistical methods have higher variance.

On the other hand, bias refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a simpler model.

A couple of sketchs of a nonlinear data in Figure 2.9 in the textbook is as follows.



0. The left plot is for the original data(sampled).

1. The red graphs in the middle and the right hand side plot are for TEST MSE. We can see that it has a U shape.

2. The gray graph in the middle plot is for TRAINING MSE. We can see that the training MSE decreases as the flexibility increases.
3. The dashed graph in the middle and the right hand side plot are for the irreducible error. We can see that it stays constant although the flexibility increases.
4. The blue graph in the right hand side plot is for the bias. We can see that the bias decreases as the flexibility increases.
5. The orange graph in the right hand side plot is for the variance. We can see that the variance increases as the flexibility increases.

(b) Explain why each of the five curves has the shape displayed in part (a).

1. The irreducible error: This term has nothing to do with flexibility. Therefore, it stays constant.
2. The training MSE: Since the more flexible model fits more to the training data, it reduces the training MSE.
3. The variance: Since the more flexible model is designed specifically for the training data, the variance for the test data is higher.
4. The bias: On the other hand, bias decreases because the more flexible model reduces the approximating error.
5. The test MSE: It has U shape since it is the sum of the variance and the bias along with the irreducible error term. It never goes below the irreducible error since the variance and the bias are all non-negative.

Q3. Ch2-Prob5: What are the advantages and disadvantages of a very flexible (versus a less flexible) approach for regression or classification? Under what circumstances might a more flexible approach be preferred to a less flexible approach? When might a less flexible approach be preferred?

Advantages of a Very Flexible Approach:

1. Better Fit to Complex Data Patterns: A very flexible approach, such as a high-degree polynomial or a complex machine learning model (e.g., deep neural networks), can capture intricate and non-linear relationships in the data. This is advantageous when the underlying data-generating process is highly complex.

2. Lower Bias: More flexible models tend to have lower bias, meaning they are less likely to make systematic errors in predictions. They can closely approximate the true underlying function.

3. Higher Accuracy: In situations where the true relationship between predictors and the target variable is complex and nonlinear, a very flexible approach can provide higher predictive accuracy on both training and test data.

Disadvantages of a Very Flexible Approach:

1. Higher Variance: Very flexible models are prone to overfitting, where they capture noise in the data rather than the true underlying pattern. This results in high variance, causing the model's

predictions to be sensitive to small changes in the training data.

2. Increased Computational Complexity: More complex models often require larger datasets and more computational resources for training and inference. This can be a disadvantage in terms of time and resource efficiency.

3. Reduced Interpretability: Highly flexible models are often black-box models, making it challenging to interpret the relationships between predictors and outcomes. Interpretability can be important in some applications, such as healthcare or finance.

When a More Flexible Approach Might Be Preferred:

1. Complex Data Patterns: When dealing with complex, non-linear data patterns or when there is no prior knowledge about the functional form of the relationship, a more flexible approach is preferred to capture the underlying complexity.

2. Large Amounts of Data: If you have a large dataset, a more flexible model can potentially learn from the data without overfitting. It has more capacity to handle a variety of patterns within the data.

3. High Predictive Accuracy: When the primary goal is achieving the highest predictive accuracy, and interpretability is less important, a very flexible approach can be suitable.

When a Less Flexible Approach Might Be Preferred:

1. Limited Data: When you have a small dataset, using a very flexible model can lead to overfitting. In such cases, simpler models with higher bias and lower variance may provide more reliable predictions.

2. Interpretability: In applications where model interpretability is crucial for decision-making or regulatory reasons (e.g., healthcare or finance), a less flexible model, such as linear regression or decision trees, may be preferred for its transparency.

3. Computational Efficiency: When computational resources are limited, or real-time predictions are required, less flexible models are often faster to train and make predictions.

4. Reducing Model Complexity: In scenarios where simplicity is valued and there is a risk of overfitting with complex models, less flexible models can be preferred to maintain a more interpretable and less complex model structure.

In practice, the choice between a more flexible or less flexible approach depends on the specific characteristics of the dataset, the goals of the analysis, and the trade-off between bias and variance. Model selection should be driven by a careful evaluation of these factors to ensure the chosen model meets the needs of the problem at hand.

Q4. Ch2-Prob8: This exercise relates to the College data set, which can be found in the file College.csv on the book website. It contains a number of variables for 777 different universities and colleges in the US.

(a) Use the pd.read_csv() function to read the data into Python. Call the loaded data college. Make sure that you have the directory set to the correct location for the data.

College.csv 파일을 불러왔습니다. 첫 다섯 행은 아래와 같습니다.

```
[ ]  college = pd.read_csv('/content/drive/MyDrive/ISLP/College.csv')
     college.head()
```

	Unnamed: 0	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate
0	Abilene Christian University	Yes	1660	1232	721	23	52	2885	537	7440
1	Adelphi University	Yes	2186	1924	512	16	29	2683	1227	12280
2	Adrian College	Yes	1428	1097	336	22	50	1036	99	11250
3	Agnes Scott College	Yes	417	349	137	60	89	510	63	12960
4	Alaska Pacific University	Yes	193	146	55	16	44	249	869	7560

(b) Look at the data used in the notebook by creating and running a new cell with just the code college in it. You should notice that the first column is just the name of each university in a column named something like Unnamed: 0. We don't really want pandas to treat this as data. However, it may be handy to have these names for later. Try the following commands and similarly look at the resulting data frames:

college2 = pd.read_csv('College.csv', index_col=0)
college3 = college.rename({'Unnamed: 0': 'College'},axis=1)
college3 = college3.set_index('College')

다운받은 자료에 이미 Unnamed는 다 삭제되어 있어서 따로 처리해줄 필요가 없었습니다.
그래서 위처럼 college2와 college3를 설정하였습니다.

```
[ ]  college2 = pd.read_csv('/content/drive/MyDrive/ISLP/College.csv', index_col = 0)
     college3 = college.rename({'Unnamed: 0': 'College'}, axis =1)
     #college3 = college3.set_index('College')
```

(c) Use the describe() method of to produce a numerical summary of the variables in the data set.

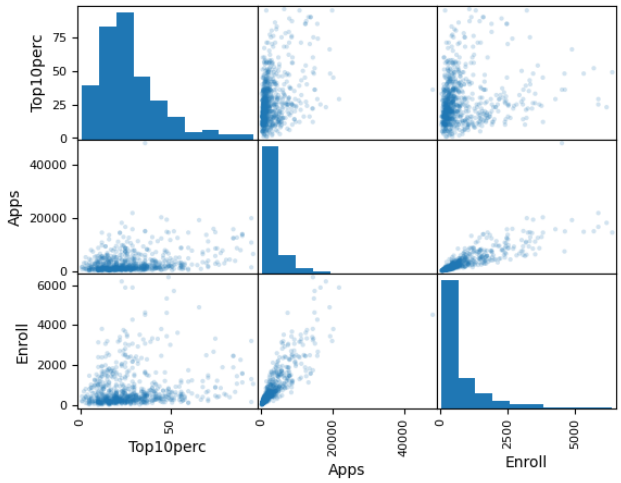
Describe() method를 사용해서 여러 통계 값들을 확인하였습니다.

```
[ ] college.describe()
```

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal
count	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000
mean	3001.638353	2018.804376	779.972973	27.558559	55.796654	3699.907336	855.298584	10440.669241	4357.526384	549.380952	1340.642214
std	3870.201484	2451.113971	929.176190	17.640364	19.804778	4850.420531	1522.431887	4023.016484	1096.696416	165.105360	677.071454
min	81.000000	72.000000	35.000000	1.000000	9.000000	139.000000	1.000000	2340.000000	1780.000000	96.000000	250.000000
25%	776.000000	604.000000	242.000000	15.000000	41.000000	992.000000	95.000000	7320.000000	3597.000000	470.000000	850.000000
50%	1558.000000	1110.000000	434.000000	23.000000	54.000000	1707.000000	353.000000	9990.000000	4200.000000	500.000000	1200.000000
75%	3624.000000	2424.000000	902.000000	35.000000	69.000000	4005.000000	967.000000	12925.000000	5050.000000	600.000000	1700.000000
max	48094.000000	26330.000000	6392.000000	96.000000	100.000000	31643.000000	21836.000000	21700.000000	8124.000000	2340.000000	6800.000000

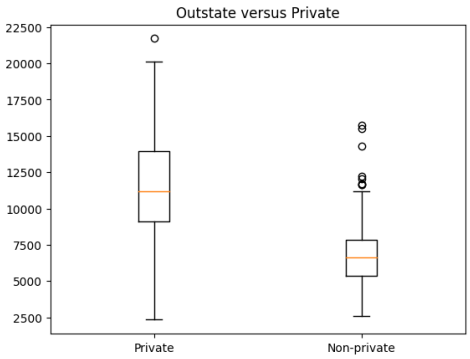
(d) Use the pd.plotting.scatter_matrix() function to produce a scatterplot matrix of the first columns [Top10perc, Apps, Enroll]. Recall that you can reference a list C of columns of a data frame A using A[C].

scatter plot은 아래와 같습니다. Enrollment와 Application의 숫자가 linear한 것처럼 보입니다.



(e) Use the boxplot() method of college to produce side-by-side boxplots of Outstate versus Private.

사립학교와 공립학교를 나누어서 Outstate의 boxplot을 그렸습니다. 사립학교의 Outstate 값이 더 크게 나오는 것을 확인하였습니다.



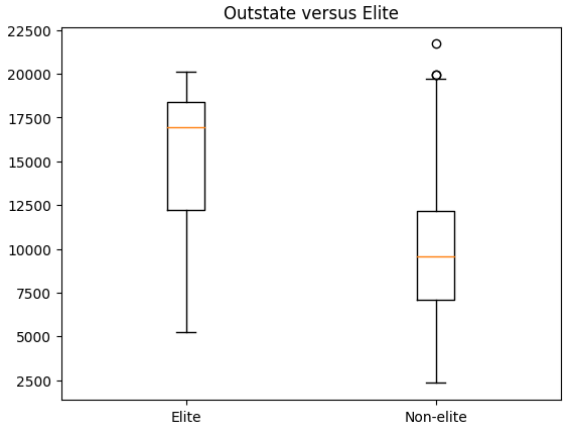
(f) Create a new qualitative variable, called Elite, by binning the Top10perc variable into two groups based on whether or not the proportion of students coming from the top 10% of their high school classes

exceeds 50%. college['Elite'] = pd.cut(college['Top10perc'], [0,0.5,1], labels=['No', 'Yes']) Use the value_counts() method of college['Elite'] to see how many elite universities there are. Finally, use the boxplot() method again to produce side-by-side boxplots of Outstate versus Elite.

자료에 있는 Top10perc의 의미를 고등학교 때 상위 10% 안에 들었던 학생들의 비율을 말하는 것으로 해석했습니다. 그래서 [0, 0.5, 1]로 나누지 않고 [0, 50, 100]으로 나누어서 구했습니다.

총 777개의 학교 중에 Elite 학교는 78개로 약 10% 정도가 해당되었습니다.

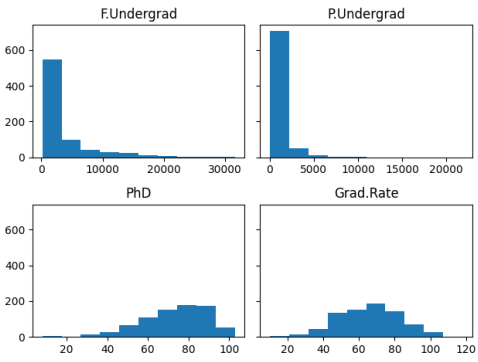
Boxplot을 통해 Elite 학교의 Outstate(주 밖에서 온 학생)의 값이 대체로 더 높은 것을 확인하였습니다.



(g) Use the plot.hist() method of college to produce some histograms with differing numbers of bins for a few of the quantitative variables. The command plt.subplots(2, 2) may be useful: it will divide the plot window into four regions so that four plots can be made simultaneously. By changing the arguments you can divide the screen up in other combinations.

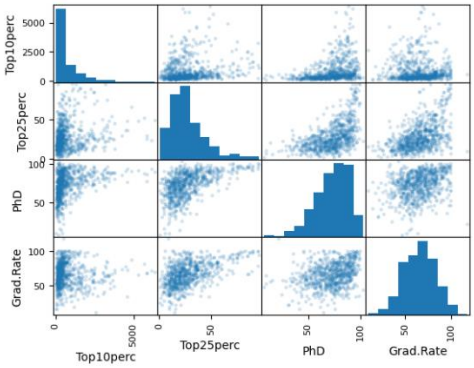
.iloc 함수를 사용해 해당 열을 구해 이를 histogram으로 그렸습니다. 분포는 아래와 같습니다.

```
fig, axs = plt.subplots(2, 2, sharey=True, tight_layout=True)
axs[0,0].hist(college.iloc[:,7])
axs[0,1].hist(college.iloc[:,8])
axs[1,0].hist(college.iloc[:,13])
axs[1,1].hist(college.iloc[:,18])
axs[0, 0].set_title("F.Undergrad")
axs[0, 1].set_title("P.Undergrad")
axs[1, 0].set_title("PhD")
axs[1, 1].set_title("Grad.Rate")
```



(h) Continue exploring the data, and provide a brief summary of what you discover.

Top10%와 Top25%, PhD, Graduate Rate의 관계가 있을까 궁금하여 scatter plot을 그려보았습니다. Top10%와 Graduate Rate은 관계가 있지 않을까 생각했지만 유의미한 관계는 없었습니다. 제일 관계가 있어 보이는 것은 PhD와 Graduate Rate인 것 같습니다. Graduate이 높으면 PhD의 비율도 높아지는 경향을 보입니다.



9. This exercise involves the Auto data set studied in the lab. Make sure that the missing values have been removed from the data.

(a) Which of the predictors are quantitative, and which are qualitative?

Quantitative Predictors (Numerical): mpg (miles per gallon), displacement, horsepower, weight, acceleration

Qualitative Predictor (Categorical): cylinders, year, origin, name

(b) What is the range of each quantitative predictor? You can answer this using the min() and max() methods in numpy.

min(Quantitative predictors)

```
mpg              9.0
displacement     68.0
horsepower       46.0
weight         1613.0
acceleration      8.0
```

(c) What is the mean and standard deviation of each quantitative .max() predictor?

max(Quantitative predictors)

```
mpg             46.6
displacement    455.0
horsepower      230.0
weight         5140.0
acceleration     24.8
```

Also, the mean and standard deviation of these predictors are as follows.

mean

```
mpg             23.445918
displacement    194.411990
horsepower      104.469388
weight         2977.584184
acceleration     15.541327
```

std

```
mpg              7.805007
displacement   104.644004
horsepower      38.491160
weight         849.402560
acceleration     2.758864
```

(d) Now remove the 10th through 85th observations. What is the range, mean, and standard deviation of each predictor in the subset of the data that remains?

min

```
mpg              9.0
displacement     70.0
horsepower       46.0
weight         1613.0
acceleration      8.0
```

max

```
mpg             35.0
displacement    455.0
horsepower      225.0
weight         5140.0
acceleration     23.5
```

mean

```
mpg             19.447368
displacement    223.098684
horsepower      119.526316
weight         3148.263158
acceleration     14.809211
```

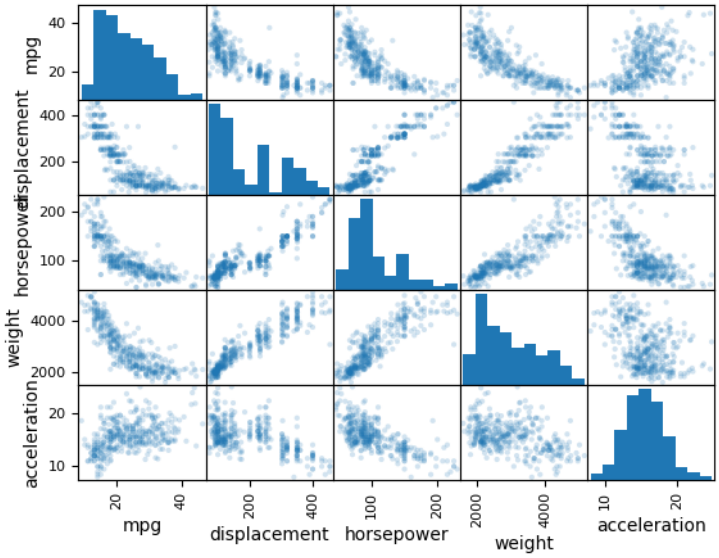
std

```
mpg              6.147942
displacement   118.140284
horsepower      44.828406
weight         977.922470
acceleration     2.841323
```

min 과 max 의 값으로 range 가 줄어들었음을 확인할 수 있습니다. 또한 표본의 수가 줄어들었기 때문에 표준편차도 감소하였음을 확인할 수 있습니다. Mean 은 변수마다 증가와 감소가 다 다릅니다.

(e) Using the full data set, investigate the predictors graphically, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.

Mpg, displacement, horsepower, weight, acceleration의 5개의 변수로 scatter plot을 그려보았습니다.



그래프로 보아 (weight, displacement)와 (weight, horsepower)는 기울기가 양수인 linear한 관계가 있음 추측할 수 있습니다. 무게가 많이 나갈수록 배기량과 마력이 높을 것임을 예상할 수 있습니다. 또한 displacement, horsepower, weight가 mpg와는 반비례한 관계를 보이는 것을 확인할 수 있습니다. 무겁고, 마력이 세고, 배기량이 많으면 연비가 좋지 않다는 상관관계가 있음을 볼 수 있습니다.

(f) Suppose that we wish to predict gas mileage (mpg) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer.

위에서 말했던 것처럼 displacement, horsepower, weight을 보고 연비를 추측할 수 있습니다. 마력이 세고, 무겁고, 배기량이 많으면 연비가 좋지 않을 것이라고 추측할 수 있습니다. 하지만 displacement, horsepower, weight 이 세 변수는 서로간에도 linear한 관계를 갖고 있기 때문에 multi variable linear regression을 한다면 이중에 어떤 factor는 p value가 낮게 나올 수도 있을 것입니다.

Q6. Ch3-Prob1: Describe the null hypotheses to which the p-values given in Table 3.4 correspond. Explain what conclusions you can draw based on these p-values. Your explanation should be phrased in terms of sales, TV, radio, and newspaper, rather than in terms of the coefficients of the linear model.

1. Null Hypothesis for TV (TV Ad Budget):

   Null Hypothesis (H0): TV advertising budget has no effect on sales.
   Alternative Hypothesis (H1): TV advertising budget has an effect on sales.

   Conclusions: Since the p-value for TV is less than the significance level (commonly set at 0.05), we reject the null hypothesis. This means that there is strong evidence to suggest that TV

advertising budget has a statistically significant impact on sales.

2. Null Hypothesis for Radio (Radio Ad Budget):

   Null Hypothesis (H0): Radio advertising budget has no effect on sales.
   Alternative Hypothesis (H1): Radio advertising budget has an effect on sales.

   Conclusions: Similar to TV, the p-value for radio is less than the significance level, so we reject the null hypothesis. There is strong evidence to suggest that radio advertising budget has a statistically significant impact on sales.

3. Null Hypothesis for Newspaper (Newspaper Ad Budget):

   Null Hypothesis (H0): Newspaper advertising budget has no effect on sales.
   Alternative Hypothesis (H1): Newspaper advertising budget has an effect on sales.

   Conclusions: In this case, the p-value for newspaper is relatively high (0.8599), above the typical significance level of 0.05. Therefore, we fail to reject the null hypothesis. This suggests that there is not enough evidence to conclude that newspaper advertising budget has a statistically significant impact on sales.

In summary, based on the p-values: TV and radio advertising budgets have a statistically significant impact on sales. Newspaper advertising budget does not have a statistically significant impact on sales. These conclusions are important for making decisions about where to allocate advertising resources to maximize the number of units sold.

Q7. Ch3-Prob4: 4. I collect a set of data (n = 100 observations) containing a single predictor and a quantitative response. I then fit a linear regression model to the data, as well as a separate cubic regression, i.e. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$.

(a) Suppose that the true relationship between X and Y is linear, i.e. $Y = \beta_0 + \beta_1 X + \epsilon$. Consider the training residual sum of squares (RSS) for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

   The training RSS for the cubic regression would be lower than RSS for the linear regression. This is because the cubic regression includes the linear regression in itself. In other words, if $\beta_2 = \beta_3 = 0$, then it becomes the linear regression. So, if $\beta_2 = \beta_3 = 0$ does not make the RSS lowest the RSS, the cubic regression would be better. Therefore for the training data, cubic regression would fit more than the linear regression because it can catch some non-linearities as well.

(b) Answer (a) using test rather than training RSS.

As we have explored in Ch2, the more flexible fitting does not always guarantee the lower the RSS. It depends on the data. If the data is highly non-linear, test RSS will be lower in the cubic regression. However, if the data is simple and linear, test RSS may be higher in the cubic regression because it might have overfitted the data.

Therefore, we cannot tell due to the lack of information.

(c) Suppose that the true relationship between X and Y is not linear, but we don't know how far it is from linear. Consider the training RSS for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

My answer is the same as in (a). The training RSS would be lower in the cubic regression since it can be a linear regression if $\beta_2 = \beta_3 = 0$. It might be even lower than in (a) since the data is non-linear.

(d) Answer (c) using test rather than training RSS.

Since we don't know how far the data is from linear, we cannot tell about the test RSS.

Q8. Ch3-Prob7: 7. It is claimed in the text that in the case of simple linear regression of Y onto X, the R^2 statistic (3.17) is equal to the square of the correlation between X and Y (3.18). Prove that this is the case. For simplicity, you may assume that $\bar{x} = \bar{y} = 0$.

In fact, we don't have to assume that $\bar{x} = \bar{y} = 0$.

First, we prove $R^2 = \frac{TSS-RSS}{TSS} = \frac{ESS}{TSS}$, where $\mathrm{ESS} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$. Proof is in below.

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

Simple derivation [edit]

$$(y_i - \bar{y}) = (y_i - \hat{y}_i) + (\hat{y}_i - \bar{y}).$$

Square both sides and sum over all i:

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

Here is how the last term above is zero from simple linear regression[2]

$$\hat{y}_i = \hat{a} + \hat{b}x_i$$
$$\bar{y} = \hat{a} + \hat{b}\bar{x}$$
$$\hat{b} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

So,

$$\hat{y}_i - \bar{y} = \hat{b}(x_i - \bar{x})$$
$$y_i - \hat{y}_i = (y_i - \bar{y}) - (\hat{y}_i - \bar{y}) = (y_i - \bar{y}) - \hat{b}(x_i - \bar{x})$$

Therefore,

$$\sum_{i=1}^{n} 2(\hat{y}_i - \bar{y})(y_i - \hat{y}_i) = 2\hat{b} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \hat{y}_i)$$

$$= 2\hat{b} \sum_{i=1}^{n} (x_i - \bar{x})((y_i - \bar{y}) - \hat{b}(x_i - \bar{x}))$$

$$= 2\hat{b} \left(\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) - \sum_{i=1}^{n} (x_i - \bar{x})^2 \frac{\sum_{j=1}^{n} (x_j - \bar{x})(y_j - \bar{y})}{\sum_{j=1}^{n} (x_j - \bar{x})^2} \right)$$

$$= 2\hat{b}(0) = 0$$

Next, we prove that R^2 coincide to the square of the correlation of X and Y in the simple regression.

We know $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$

$\hat{\beta}_1 = \frac{s_{xy}}{s_x^2}$.

From above,

$$
\begin{aligned}
R^2 &= \frac{\sum_{i=1}^n (\bar{y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 x_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \\
&= \frac{\sum_{i=1}^n \left(\hat{\beta}_1 (x_i - \bar{x})\right)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \\
&= \hat{\beta}_1^2 \frac{\frac{1}{n-1}\sum_{i=1}^n (x_i - \bar{x})^2}{\frac{1}{n-1}\sum_{i=1}^n (y_i - \bar{y})^2} \\
&= \hat{\beta}_1^2 \frac{s_x^2}{s_y^2} \\
&= \left(\frac{s_x}{s_y}\hat{\beta}_1\right)^2 .
\end{aligned}
$$

Therefore,

$$R^2 = \left(\frac{s_x}{s_y}\hat{\beta}_1\right)^2 = r_{xy}^2 .$$

Q9. Ch3-Prob13: In this exercise you will create some simulated data and will fit simple linear regression models to it. Make sure to use the default random number generator with seed set to 1 prior to starting part (a) to ensure consistent results.

(a) Using the normal() method of your random number generator, create a vector, x, containing 100 observations drawn from a N(0, 1) distribution. This represents a feature, X.

```
x = np.random.normal(0,1,100)
print(x)
```

```
[-4.95727954e-02 -6.46036406e-01 -7.87961697e-01  3.72113431e-01
  6.34278134e-01  1.83689279e+00 -2.91383083e-01 -2.63037032e-01
  2.94468457e-01  2.76614622e-01  3.30682697e-01 -1.09739760e+00
  6.81885593e-01  1.13781469e+00 -2.43912411e-01 -1.46827477e+00
```

(b) Using the normal() method, create a vector, eps, containing 100 observations drawn from a N(0, 0.25) distribution—a normal distribution with mean zero and variance 0.25.

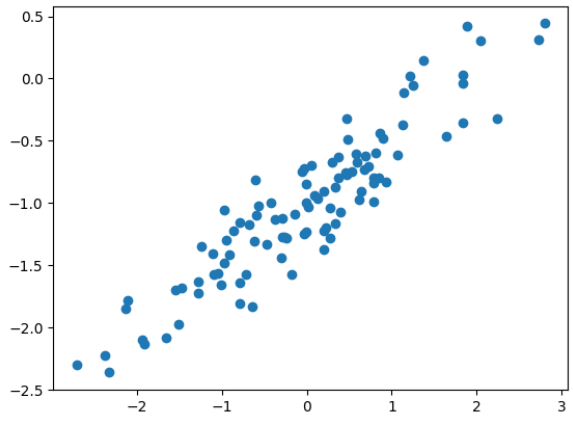
```
eps = np.random.normal(0,0.25,100)
print(eps)
```

```
[ 0.27930435 -0.50566614 -0.41520281  0.0148451  -0.21949897 -0.27342737
  0.02371078 -0.1371201   0.17821576 -0.17436651 -0.33001712 -0.02067232
 -0.06734031  0.31971174 -0.15755346  0.04870256  0.11008682 -0.03675472
```

(c) Using x and eps, generate a vector y according to the model Y = −1 + 0.5X + ϵ. (3.39) What is the length of the vector y? What are the values of β_0 and β_1 in this linear model?

len(y) = 100, $\beta_0 = -1$ and $\beta_1 = 0.5$

(d) Create a scatterplot displaying the relationship between x and y. Comment on what you observe.

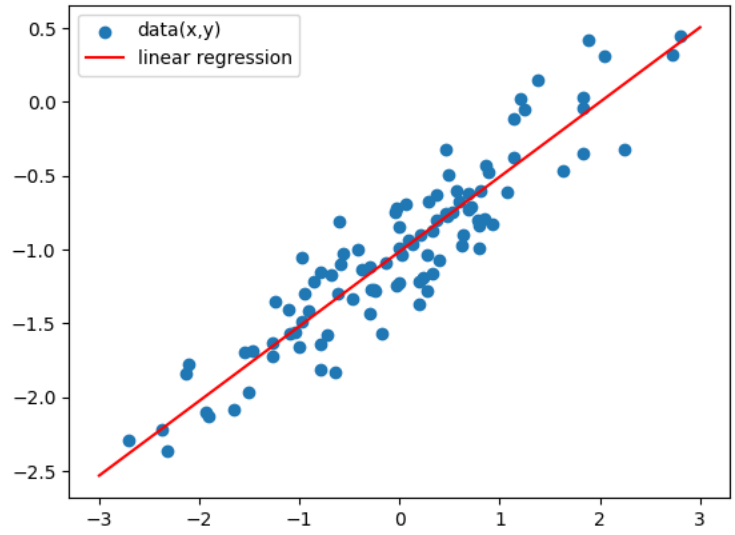


대략적으로 y = -1 + 0.5x 의 개형을 따르는 것을 볼 수 있다.

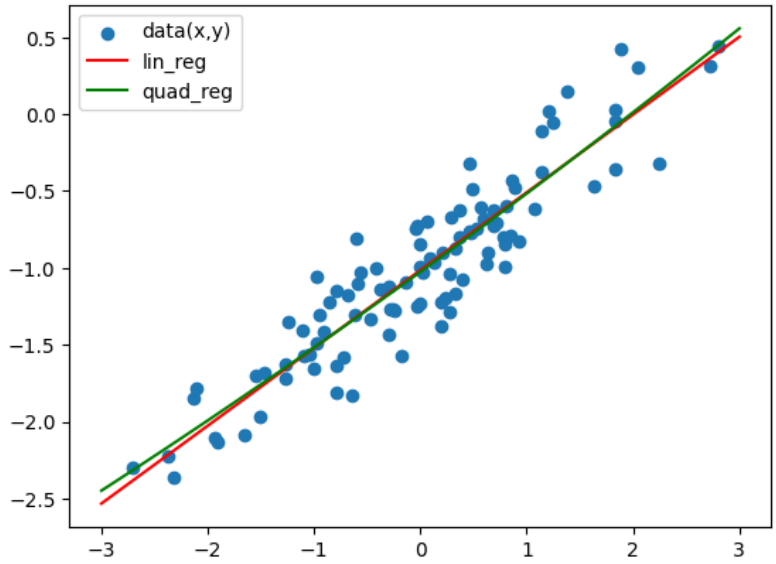
(e) Fit a least squares linear model to predict y using x. Comment on the model obtained. How do $\widehat{\beta_0}$and $\widehat{\beta_1}$ compare to β_0 and β_1?

$\widehat{\beta_0} = -1.0141996317475 57, \widehat{\beta_1} = 0.5057655906370283$. 참값과 비슷함을 볼 수 있습니다.

(f) Display the least squares line on the scatterplot obtained in (d). Draw the population regression line on the plot, in a different color. Use the legend() method of the axes to create an appropriate legend.



(g) Now fit a polynomial regression model that predicts y using x and x^2. Is there evidence that the quadratic term improves the model fit? Explain your answer.

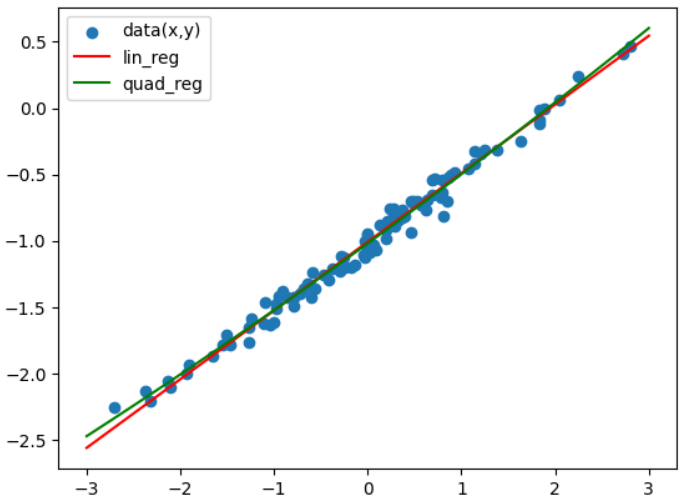


Linear regression: $R^2 = 0.8502817335762135$

Quadratic regression: $R^2 = 0.8509358593536973$

Quadratic regression의 R^2값이 조금 더 높게 나옵니다. 약간의 개선이 있지만, 큰 차이는 없다고 할 수 있습니다.

(h) Repeat (a)–(f) after modifying the data generation process in such a way that there is less noise in the data. The model (3.39) should remain the same. You can do this by decreasing the variance of the normal distribution used to generate the error term ϵ in (b). Describe your results.



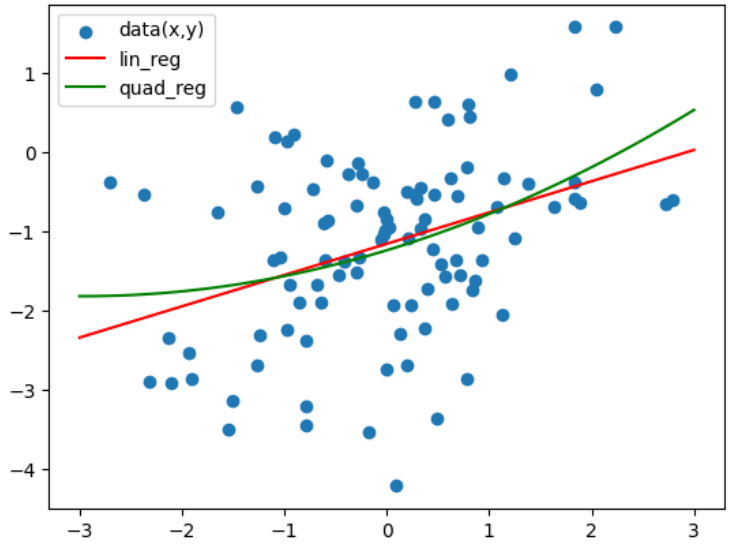
$\epsilon \sim N(0,0.25^2)$으로 설정하였습니다.

Linear regression: $R^2 = 0.9862889826038156$

Quadratic regression: $R^2 = 0.9871025126414941$

노이즈가 줄어드니 데이터 분포가 직선에 가까워지면서 linear regression의 R^2값이 증가한 것을 확인할 수 있습니다. 또한 Quadratic regression에서의 R^2값이 조금 더 크게 나옵니다.

(i) Repeat (a)–(f) after modifying the data generation process in such a way that there is more noise in the data. The model (3.39) should remain the same. You can do this by increasing the variance of the normal distribution used to generate the error term ϵ in (b). Describe your results.



$\epsilon \sim N(0,1^2)$으로 설정하였습니다.

Linear regression: $R^2 = 0.14014735434264108$

Quadratic regression: $R^2 = 0.14995237216917376$

노이즈가 커지니 데이터 분포가 선형과 멀어지면서 linear regression의 R^2값이 감소한 것을 확인할 수 있습니다. 또한 Quadratic regression에서의 R^2값이 조금 더 크게 나옵니다.

(j) What are the confidence intervals for β_0 and β_1 based on the original data set, the noisier data set, and the less noisy data set? Comment on your results.

	$CI\ of\ \beta_0$		$CI\ of\ \beta_1$	
Original	-1.016	-1.012	0.102	0.910
Less noisy	-1.009	-1.008	0.417	0.619
Much noisy	-1.166	-1.153	-1.220	2.010

각 CI의 왼쪽이 Lower bound, 오른쪽이 Upper bound의 값입니다.

1. 노이즈가 클수록 CI의 길이도 커지는 것을 확인할 수 있습니다.

2. β_0의 CI보다 β_1의 길이가 훨씬 크게 나옵니다. 이는 기울기에 대한 추정은 불확실하지만 기울기가 바뀌어도 y절편은 크게 바뀌지 않음을 의미합니다.

3. β_0의 CI는 참값인 $\beta_0 = -1$을 포함하지 않습니다. Random data를 뽑았기 때문입니다. x를 다시 random하게 추출한다면 $\beta_0 = -1$을 포함하도록 나올 확률이 높을 것입니다.

Q10. Ch3-Prob14: This problem focuses on the collinearity problem.

a) Perform the following commands in Python:

```
rng = np.random.default_rng (10)
x1 = rng.uniform(0, 1, size=100)
x2 = 0.5 * x1 + rng.normal(size=100) / 10
y = 2 + 2 * x1 + 0.3 * x2 + rng.normal(size=100)
```

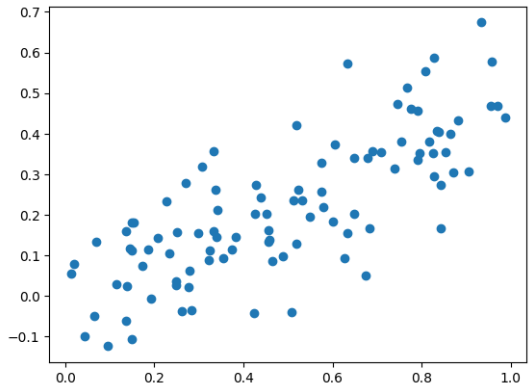
The last line corresponds to creating a linear model in which y is a function of x1 and x2. Write out the form of the linear model. What are the regression coefficients?

```
rng = np.random.default_rng (10)
x1 = rng.uniform(0, 1, size=100)
x2 = 0.5 * x1 + rng.normal(size=100) / 10
y = 2 + 2 * x1 + 0.3 * x2 + rng.normal(size=100)
```

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon,\ \beta_0 = 2,\ \beta_1 = 2,\ \beta_2 = 0.3$ 입니다.

b) What is the correlation between x1 and x2? Create a scatterplot displaying the relationship between the variables.

Correlation coefficient $\rho = 0.7723245$ 입니다. X1과 x2의 scatter plot은 아래와 같습니다.



c) Using this data, fit a least squares regression to predict y using x1 and x2. Describe the results obtained. What are $\widehat{\beta_0}$, $\widehat{\beta_1}$, and $\widehat{\beta_2}$? How do these relate to the true β_0, β_1, and β_2? Can you reject the null hypothesis H0 : β_1 = 0? How about the null hypothesis H0 : β_2 = 0?

$\widehat{Beta} = (X^T X)^{-1} X^T y$ 공식을 사용하여 $\widehat{\beta_0}$, $\widehat{\beta_1}$, $\widehat{\beta_2}$ 을 구하면 다음과 같습니다.

$\widehat{\beta_0} = 1.958,\ \widehat{\beta_1} = 1.615,\ \widehat{\beta_2} = 0.943$

$\beta_0 = 2,\ \beta_1 = 2,\ \beta_2 = 0.3$ 과 비교했을 때, $\widehat{\beta_0}$은 비슷하게 나왔지만 $\widehat{\beta_1}$이 조금 작게, $\widehat{\beta_2}$은 크게 나왔습니다. 대체로 참값과 비슷한 것을 확인할 수 있습니다.

Null hypothesis를 거부할지 판단하기 위해서 아래 공식의 F statistics를 구하였습니다.

$$F = \frac{(\mathrm{RSS}_0 - \mathrm{RSS})/q}{\mathrm{RSS}/(n-p-1)}.$$

β_1에 대해서 구할 때는 p=q=1, β_2에 대해서 구할 때는 p=2, q=1로 설정하였습니다.

각 F 값은 다음과 같습니다. $F_1 = 101.97796251857869,\ F_2 = 8.428968998638013$

이에 해당하는 p value를 각각 구해보면, $p_1 = 1.11 * 10^{-16},\ p_2 = 0.000421$ 이 나옵니다.

두 값 모두 0.05보다 현저히 작으므로, 우리는 두 Null hypothesis를 모두 reject 합니다.

d) Now fit a least squares regression to predict y using only x1. Comment on your results. Can you reject the null hypothesis H0 : β_1 = 0?

앞서 사용했던 공식을 사용해 $\widehat{\beta_0}$, $\widehat{\beta_1}$을 구하면 아래와 같습니다.

$\widehat{\beta_0} = 1.9267,\ \widehat{\beta_1} = 2.0980$

H0의 reject 여부를 판단하기 위해 아래의 공식을 사용해서 t value를 구하였습니다.

$$t = \frac{\hat{\beta}_1 - 0}{\mathrm{SE}(\hat{\beta}_1)},$$

t value: 1.399268978187182 가 나왔고 이에 해당하는 p value는 n이 100이기 때문에 정규분포표에서 찾았습니다.

p value: 0.08079999999999998 > 0.05, 따라서 우리는 H0를 'accept' 합니다!

(e) Now fit a least squares regression to predict y using only x2. Comment on your results. Can you reject the null hypothesis H0 : β_2 = 0?

$\widehat{\beta_0} = 2.3174,\ \widehat{\beta_1} = 2.9397$

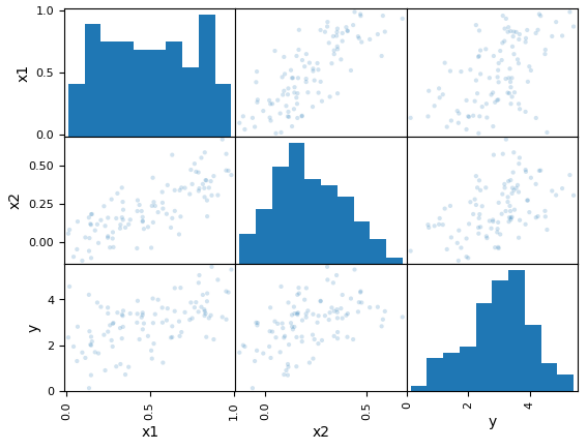
t value: 2.341570722686783 가 나왔고 이에 해당하는 p value는 n이 100이기 때문에 정규분포표에서 찾았습니다.

p value: 0.0096 < 0.05, 따라서 우리는 H0를 reject 합니다.

(f) Do the results obtained in (c)–(e) contradict each other? Explain your answer.

(c)에서는 H_0: $\beta_1 = 0$ 을 reject 하였지만, (d)에서는 accept 하였습니다. 한 가지 변수만 고려해서 fitting을 하였기 때문에 x1과의 연관성이 더 적게 나타난 것을 볼 수 있습니다.

x1, x2, y의 scatter plot은 아래와 같이 나옵니다.



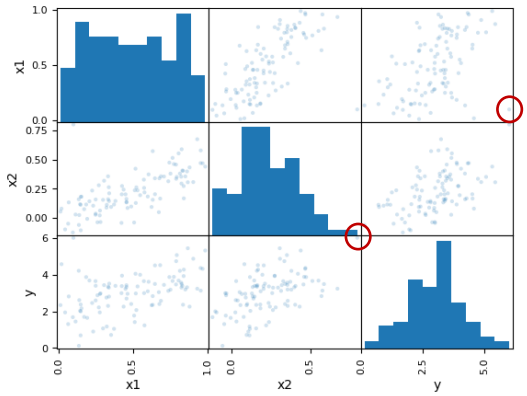
결과의 해석: y의 개형이 x1보다는 x2에 더 가까운 것을 확인할 수 있습니다. 만약에 우리가 x1과 y만을 비교한다면 연관이 별로 없다고 판단할 수 있습니다. 하지만 x2와의 연관성을 먼저 고려한 다음 x1과의 연관성을 고려한다면 x1과의 연관성을 찾을 수 있습니다. 따라서 변수가 여러 개일 때는 collinearity를 고려해야 합니다!

(g) Suppose we obtain one additional observation, which was unfortunately mismeasured. We use the function np.concatenate() to np.concatenate() add this additional observation to each of x1, x2 and y.

```
x1 = np.concatenate([x1, [0.1]])
x2 = np.concatenate([x2, [0.8]])
y = np.concatenate([y, [6]])
```

Re-fit the linear models from (c) to (e) using this new data. What effect does this new observation have on the each of the models? In each model, is this observation an outlier? A high-leverage point? Both? Explain your answers.

저 새로운 점을 추가하여 scatter plot을 다시 그리면 아래와 같습니다. 빨간 원으로 표시한 부분이 해당하는 leverage point 입니다.



이 점을 추가해서 다시 $\widehat{\beta_0}$, $\widehat{\beta_1}$, $\widehat{\beta_2}$ 을 구하면 다음과 같습니다.

(refit) $\widehat{\beta_0} = 1.958,\ \widehat{\beta_1} = 0.858,\ \widehat{\beta_2} = 2.266$

(이전) $\widehat{\beta_0} = 2.062,\ \widehat{\beta_1} = 1.615,\ \widehat{\beta_2} = 0.943$

(참값) $\beta_0 = 2,\ \beta_1 = 2,\ \beta_2 = 0.3$

새로 추가한 leverage point는 그림에서 보듯이 x1과 y의 경향성과는 멀리 떨어져 있는 반면, x2와 y의 경향성 위에 놓여있습니다. Refit 후의 $\widehat{\beta_1}$이 작아지고 $\widehat{\beta_2}$가 커진 것이 이를 반영합니다. 또한 $\widehat{\beta_1}$과 $\widehat{\beta_2}$의 변화폭이 상당히 큰데, 이는 leverage point가 influential point 임을 의미한다고 할 수 있습니다. Refit 후의 추정 값이 참값과 꽤나 다른데, 이는 noise 에 비해 상대적으로 sample 수가 부족하기 때문이라고 생각합니다.

References

https://en.wikipedia.org/wiki/Explained_sum_of_squares
https://statproofbook.github.io/P/slr-rsq.html