2020

DataMining

HW2

192STG11 우나영

1. Description

이번 과제에서는 첫 번째로 ISL(II)의 3.3.3 Potential problem, 3.4 Marketing plan 그리고 3.5 Comparison of linear regression with K-Nearest-Neighbor 을 정리하는 것이다. 두 번째는 ISL(II)의 3.6 Lab: linear regression 을 읽고 linear regression model fitting 의 기초가 되는 R 코드를 작성하는 것이다. 마지막으로는 이러한 내용들을 바탕으로 ISL(II) 121 페이지에 나오는 exercise 8, 9, 13, 14 를 풀어본다.

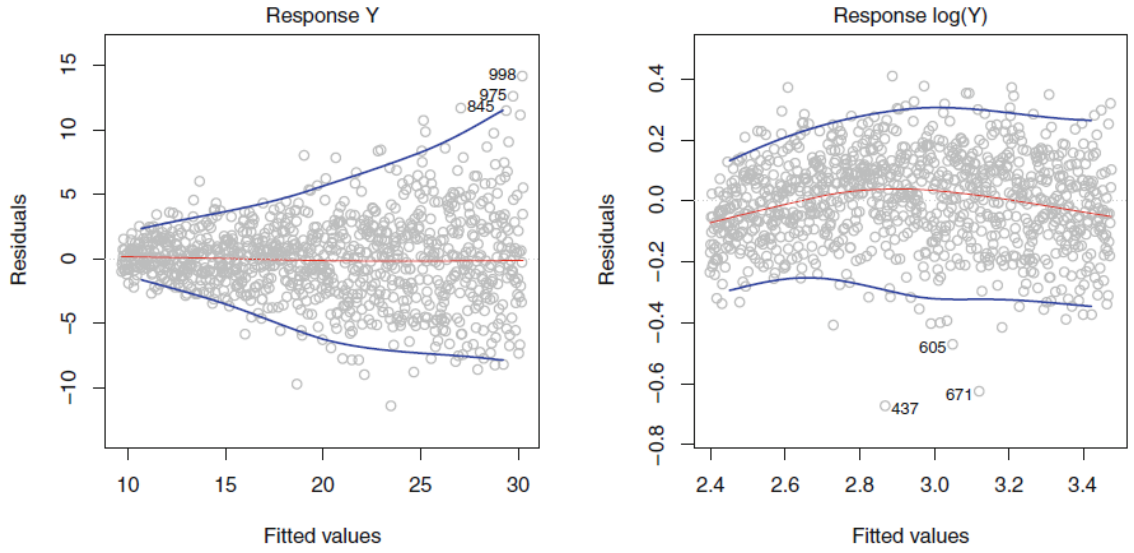
2. Implementation

3.3.3 Potential problem 은 데이터를 linear regression 에 적합할 때 발생하는 6 가지 문제들에 대해 다룬다. 6 가지 문제는 Non-linearity of the response-predictor relationship, Correlation of error terms, Non-constant variance of error terms, Outliers, High-leverage point, 그리고 Collinearity 이다.

Non-linearity 는 linear regression 모델이 기본적으로 predictor 와 response variable 의 straight-line relationship 을 가정하기 때문에 발생하는 문제이다. 만약 predictor 와 response 가 linear relationship 이 아니라면 모델의 예측력이 떨어진다. 따라서, predictor 와 response variable 의 linearity 를 판별하는 방법으로 residual plot 을 사용한다. Residual plot 은 simple linear model 일때, residual $e_i = y_i - \hat{y}_i$ 즉, response variable 과 prediction 의 오차를 y 축으로 predictor variable x_i를 x 축으로 놓고 그린다. 반면에, Multiple linear model 일때는, residual 을 y 축으로 예측치인 $\hat{y}_i$을 x 축으로 놓고 그린다. Predictor 와 response 가 linear 관계에 있다면 residual plot 의 점들은 랜덤하게 분포할 것이다. 그러나 residual plot 의 점들이 특정한 패턴을 보인다면 이는 predictor 와 response 가 linear 관계가 아니라는 것이다. 이때 가장 쉽게 대처하는 방법은 predictor 를 $\log X, \sqrt{X}, X^2$과 같이 non-linear 로 transform 하는 것이다.

linear model 의 중요한 가정 중 하나는 error term $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$이 uncorrelated 하다는 것이다. 즉 ϵ_i 값이 ϵ_{i+1} 값에 어떠한 영향도 미치지 않는다는 것이다. 만약 error term 이 uncorrelated 하다는 가정을 만족하지 않는다면 잘못된 parameter 추정구간을 얻게 된다. 즉, 모델의 신뢰성을 잃는다. 이러한 error term 의 correlation 은 주로 시계열(time serial) 데이터에서 자주 발생한다. 또한 linear model 은 error term 의 등분산성 즉, $\mathrm{Var}(\epsilon_i) = \sigma^2$ 가정이 필요하다. error term 의 등분산성의 가정을 바탕으로 모델의 confidence interval, standard errors 그리고 가설검정을 수행하기 때문에 error term 의 등분산성의 성립은 중요하다. 이분산성(heteroscedasticity)이 발생했을 때 해결할 수 있는 방법 중 하나는 바로

response variable 을 $\log Y$ 또는 $\sqrt{Y}$ 로 변형시키는 것이다. 아래에 좌측에는 이분산성을 보여주는 residual plot 과 우측에는 response variable 을 변형시킨 후 residual plot 을 보여준다.



또 다른 방법으로는 weighted least square 이다. 이는 i번째 response variable 이 n_i개의 raw observations 의 평균값이고 각각의 raw observation 들이 variance σ^2에 uncorrelated 할 때 사용할 수 있다. 이때 error term 의 variance 는 $\sigma_i^2 = \sigma^2/n_i$이며 weight 는 $w_i = n_i$이다.

Outlier 는 model 의 추정치와 큰 차이를 보이는 관측치 y_i이다. Outlier 여부에 따라 모델의 정확도가 달라지기 때문에 outlier 를 판단하고 제거 여부를 판단하는 것은 모델 진단의 중요한 과정 중 하나이다. Outlier 를 판단하는 대표적인 방법 중 하나가 바로 residual plot 이다. 다른 residual 과 달리 동떨어져 있는 residual 값을 갖는 관측치가 outlier 이다. 동떨어져 있는 정도를 판단하는 측도는 studentized residuals 으로 각각의 residual e_i를 추정된 standard error 로 나눈 값이다. studentized residuals 의 절대치가 3 이상인 관측치를 outlier 라고 진단한다. Outlier 는 모델의 적합에 큰 영향을 끼치기 때문에 outlier 제거 여부는 outlier 의 발생 원인에 대한 조사와 domain 지식을 바탕으로 신중하게 결정되어야 한다. High leverage point 는 outlier 와 달리 비정상적인 predictor variable x_i 값을 가진 관측치이다. High leverage point 는 이름 그대로 least squares line 추정에 큰 영향을 준다. 관측치의 leverage 측도는 simple linear regression 일 경우 아래와 같다.

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

h_i값은 x_i가 $\bar{x}$에서 멀어질수록 커진다. leverage statistic 은 항상 $1/n$과 1 사이에 존재하기 때문에 leverage statistics 의 평균은 항상 $(p+1)/n$이다. 따라서 관측치의 leverage statistic 이 $(p+1)/n$을 넘는다면 high leverage point 로 의심할 수 있다. 특히 high leverage 인 동시에 outlier 인 경우는 최악의 조합이라고 할 수 있다.

마지막으로 linear model 적합에서 발생하는 이슈는 collinearity 이다. Collinearity 는 두 개 이상의 predictor variable 이 서로 correlation 이 있는 경우를 말한다. Predictor variable 간에 collinearity 가 있는 경우 predictor variable 각각이 response 에 얼마나 영향을 끼치는지 측정하기가 어렵기 때문에 regression 적합에서 문제가 된다. Collinearity 는 회귀 계수의 정확도를 떨어뜨리기 때문에 회귀계수 $\hat{\beta_j}$의 standard error 가 커지게 된다. 이때 t statistics 는 $\hat{\beta_j}$을 $\hat{\beta_j}$의 standard error 로 나눈 값으로 $\hat{\beta_j}$의 standard error 이 커짐에 따라 작아진다. 따라서 collinearity 로 인해 가설검정의 측도인 t-statics 의 값이 작아져 $H_0: \beta_j = 0$ 가설검정 power 가 줄어든다. Collinearity 는 correlation matrix 를 통해 서로 영향을 주는 predictor variable 짝을 찾아 해결할 수 있다. 그러나 실제로 collinearity 는 두 개이상의 predictor variable 의 synergy 로 발생하는 경우가 많기 때문에 correlation matrix 로 진단하기 어렵다. 따라서 이러한 multi predictor variable 사이의 collinearity 를 multicollinearity 라 부르며 이를 진단하는 측도를 VIF(Variance Inflation Factor)라 부른다. 계산 식은 아래와 같다.

$$\mathrm{VIF}(\hat{\beta_j}) = \frac{1}{1 - R^2_{X_j|-X_j}}$$

$R^2_{X_j|-X_j}$은 X_j를 나머지 predictor variable 과 linear regression 으로 적합했을 때 R^2값을 말한다. $R^2_{X_j|-X_j}$이 1 에 가까워지면 VIF 값은 커진다. 경험적으로 5~10 이면 predictor variable 간의 multicollinearity 가 존재한다고 진단할 수 있다.

3.4 는 chapter 3 에서 다뤘던 Advertising 데이터에 대한 전반적인 분석 내용을 말해준다. 첫 번째 advertising sales 와 budget 이 연관성에 대한 질문이다. 즉 $H_0: \beta_{TV} = \beta_{Radio} = \beta_{Newspaper} = 0$을 가설검정 하는 것이다. Table 3.6 에 따르면 F-statistics 의 p-value 값이 매우 작으므로 H_0을 기각하여 advertising sales 와 budget 이 관련이 있다고 판단할 수 있다. 다음으로는 이 연관성이 얼마나 강한지에 대한 질문이다. Table 3.6 의 RSE(Residual Standard Error)에 따르면 이 모델은 예측치가 12%정도 빗겨 나간다. 그리고 R^2 은 response 의 variance 가 모델에 의해 얼마나 설명되는지 보여주는 척도로 table 3.6 의 수치에 따르면 90%이다. 그렇다면 TV, radio, 그리고 newspaper 중 어떤 매체가 advertising sales 에 영향을 미치는지에 대해 의문이 들 수 있다. 이는 각 매체의 회귀 계수의 t-statistics 로 가설검정 함으로써 알 수 있다. Table 3.4 에 따르면 TV 와 radio 의 회귀계수 p-value 는 매우 낮은 반면 newspaper 는 높다는 것을 알 수 있다. 따라서 TV 와 radio 는 sales 에 영향을 미치는 반면 newspaper 는 그렇지 못하다고 판단할 수 있다. TV 와 radio 의 sales 에 미치는 영향 정도는 각각의 회귀계수의 confidence interval 을 통해 알 수 있다. TV 와 radio 의 confidence interval 은 구간이 좁고 0 에서 멀리 떨어져 있어 sales 에 영향을 미친다는 것을 알 수 있지만 newspaper 회귀계수의 confidence interval 은 0 을 포함하고 있어 sales 에 통계적으로 영향이 없다고 판단할 수 있다.

우리는 advertising 데이터로 적합한 모델로부터 얼마나 정확하게 sales 값을 추정할 수 있을까? response 는 두 가지로 예측될 수 있다. 하나는 response 각각의 값 $Y = f(X) + \epsilon$ 와 response 의 average 값 $f(X)$ 이다. 전자는 prediction interval 을 사용하고 후자는 confidence interval 을 사용한다. Prediction interval 은 ϵ에 대한 uncertainty 를 포함하고 있기 때문에 일반적으로 confidence interval 보다 범위가 넓다. 다음으로는 Advertising 데이터는 response 와 predictor 와 linear 한 관계를 갖고 있는지에 대한 질문이다. Figure 3.5 의 residual plot 에서 residual 이 일정한 패턴을 보인다는 것을 통해 non linear 한 관계라고 판단할 수 있다. 마지막으로는 이 advertising media 들 간의 synergy 효과가 있는지에 대한 질문이다. 일반적인 linear regression model 은 additive relationship 을 가정한다. 앞선 Figure 3.5 을 통해 이 모델이 linear 관계가 아니라는 판단을 할 수 있었다. 또한 interaction term 의 추가로 R^2의 값이 90%에서 97%까지 상승한다는 사실을 통해 advertising media 들 간의 synergy 효과가 있다고 판단할 수 있다.

3.5 는 Linear regression 과 K-Nearest Neighbor 를 비교한다. Linear regression 은 linear functional form 인 f(X)를 가정하여 parameter 를 추정하는 parametric 모델인 반면 KNN(K-Nearest-Neighbor)은 parametric form 을 가정하지 않는 non-parametric 모델이다. KNN 은 prediction point x_0와 K 값이 주어졌을 때, x_0와 가까운 K training 관측치(N_0)의 response 값을 average 하는 방식으로 $f(x_0)$을 예측한다. 수식으로 표현하면 아래와 같다.

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_0} y_i$$

K 값이 작다면 flexible 하게 적합이 되며 low bias 를 갖지만 높은 variance 를 갖는다. 반면 K 값이 크다면 smoother 하고 variance 가 적은 적합을 갖지만 높은 bias 를 갖게 된다.

linear regression 과 같은 parametric 모델은 true 모델이 가정한 parametric 모델에 가까울 경우 KNN 과 같은 non-parametric 모델보다 성능이 높다. 하지만 현실적으로 response variable Y 와 predictor variable X 의 관계가 linear 인 경우는 거의 없다. 따라서 대부분의 경우에는 linear regression 보다 KNN regression 이 X 와 Y 의 관계가 linear 여부에 관계 없이 성능이 좋은 경우가 많다. 그러나 KNN 은 데이터의 차원이 증가함에 따라 curse of dimensionality 현상으로 인해 성능이 떨어진다. 또한 linear regression 은 모델의 해석이 용이하기 때문에 linear model 의 성능이 KNN 보다 크게 떨어지지 않는다면 linear model 을 채택한다.

3.6 Lab : Linear Regression 을 통해 새롭게 알게 된 함수들이 많았다. fix()함수로 package 에 내장된 promise 된 데이터셋을 사용할 수 있다. lm()함수로 linear regression 적합시킨 후 confint(lm.fit)을 이용하면 회귀계수의 confidence interval 을 구할 수 있다. 또한 predict 함수로 confidence 와 prediction interval 을 구할 수 있다. plot(lm.fit)는 residual vs

fitted value plot, Normal QQ plot, standardized residual 그리고 leverage plot 까지 4 개의 모델 진단 함수를 보여준다. residual(), rstudent() 그리고 hatvalues()는 잔차, standardized 잔차 그리고 leverage statistics 를 구해준다. summary(lm.fit)$sigma 는 RSE(Residual Standard Error)값을 출력해준다. Polynomial 을 해주기 위해 lm 의 formula 에 I()를 넣거나 poly()를 넣어주는 두 가지 방법이 있다. 모델의 적합도에 대한 가설검정을 위해 anova()함수를 사용할 수 있다. anova(H_0, H_1)의 p-value 가 작다면 null hypothesis H_0을 기각하고 alternative hypothesis H_1을 채택해야 한다. lm 함수는 class 가 factor 인 변수의 경우 자동으로 dummy variable 로 만든다. constrast()함수를 사용하면 dummy variable 의 코딩 방법을 알려준다. 이처럼 다양한 R 의 기본 내장 함수를 통해 linear regression 적합과 모형 진단이 가능하다는 것을 배울 수 있었다. 다음으로는 위의 내용을 바탕으로 ISL(II) Chapter 3 의 exercise 8, 9, 13, 14 번 문제를 풀이할 것이다.

8. This question involves the use of simple linear regression on the Auto data set.

(a) Use the lm() function to perform a simple linear regression with mpg as the response and horsepower as the predictor. Use the summary() function to print the results. Comment on the output.

Auto data 는 392 개의 관측치와 9 개의 변수로 이루어진 dataset 이다.

변수명	설명	종속/독립 변수
mpg	miles per gallon	종속
cylinders	Number of cylinders between 4 and 8	독립
displacement	Engine displacement (cu. inches)	독립
horsepower	Engine horsepower	독립
weight	Vehicle weight (lbs.)	독립
acceleration	Time to accelerate from 0 to 60 mph (sec.)	독립
year	Model year (modulo 100)	독립
origin	Origin of car (1. American, 2. European, 3. Japanese)	독립
name	Vehicle name	독립

$Coefficients$	$Estimate$	$Std.Error$	$t-value$	$Pr(>\|t\|)$
$(Intercept)$	39.94	0.72	55.66	<2e-16
$horsepower$	-0.16	0.01	-24.49	<2e-16

RSE	4.906 on 390 df	$Adjusted\ R^2$	0.6049
R^2	0.6059	$p-value$	<2.2e-16
$F-statistics$	599.7 on 1 and 390 df		

ⅰ. Is there a relationship between the predictor and the response?

horsepower 의 회귀계수 p-value 가 매우 작으므로 mpg 와 horsepower 의 선형관계가 있다고 판단할 수 있다.

ⅱ. How strong is the relationship between the predictor and the response?

mpg 의 mean 은 23.45 이고 RSE 는 4.906 이다. 즉 20.92%의 에러를 갖고 있다. Adjusted R^2 값은 0.6049 로 이 모델이 response variable 인 mpg 분산의 60.49%를 설명하고 있다는 뜻이다.

ⅲ. Is the relationship between the predictor and the response positive or negative?

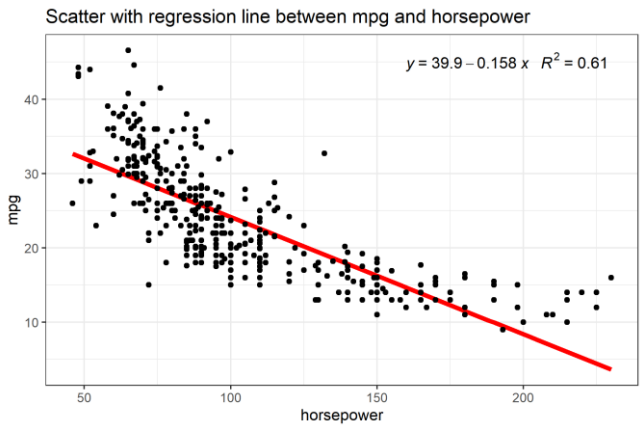
horsepower 의 회귀계수가 -0.16 으로 음수이기 때문에 predictor 인 horsepower 와 response 인 mpg 는 negative 관계에 있다고 말할 수 있다. 즉, horsepower 가 커질수록 mpg fuel 의 효율성은 줄어든다.

ⅳ. What is the predicted mpg associated with a horsepower of 98? What are the associated 95% confidence and prediction intervals?

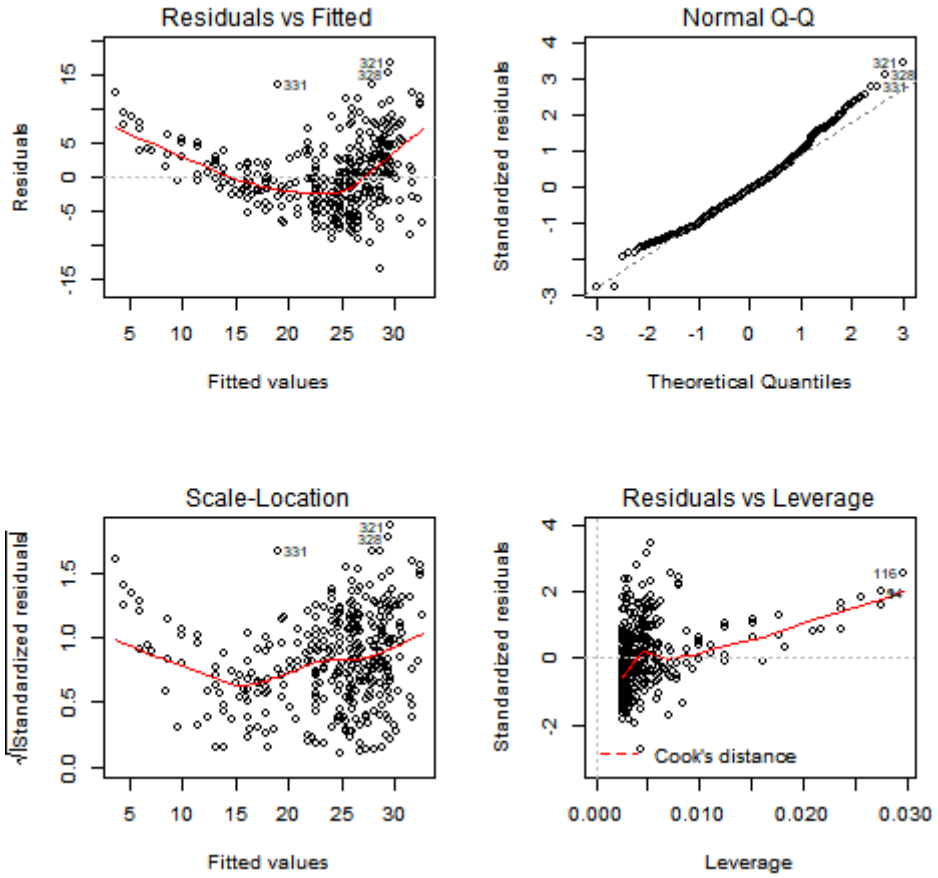
R 의 내장함수인 predict 를 사용하였다.

$Predicted\ Value$	$95\%\ Confidence\ Interval$	$95\%\ Prediction\ Interval$
24.47	(23.97, 24.96)	(14.81, 34.12)

(b) Plot the response and predictor. Use the abline() function to display the least squares regression line.



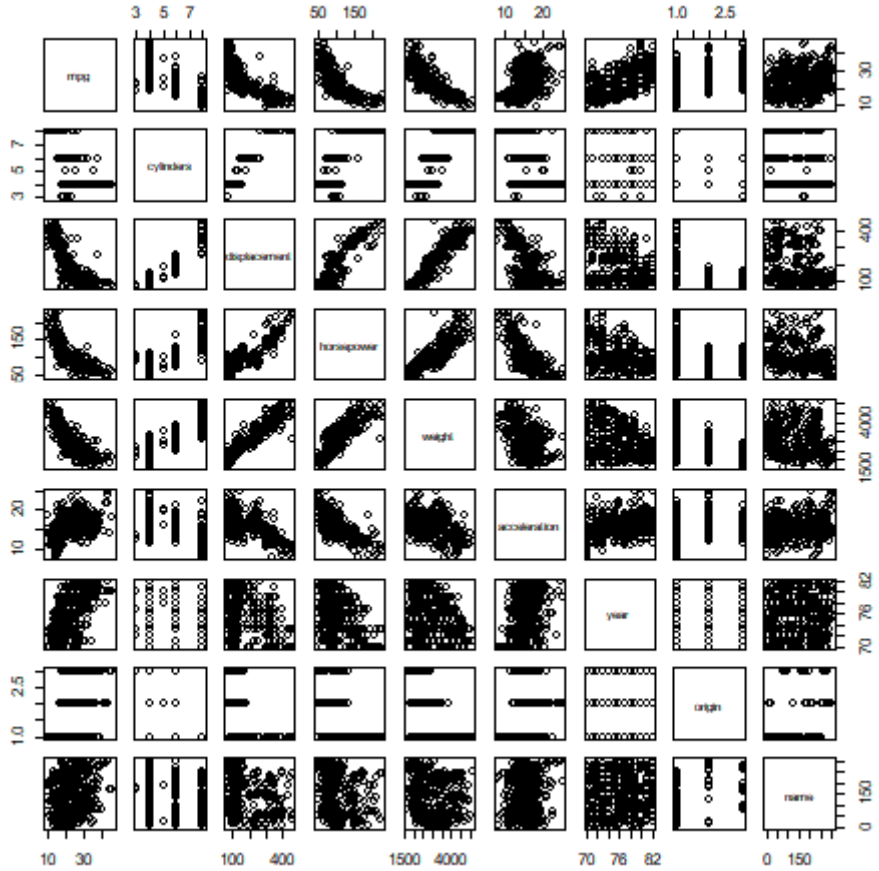
(c) Use the plot() function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.



residual vs fitted plot 에서 non-linear 패턴이 관측된다. Normal QQ plot 의 직선을 따라 점들이 분포하므로 데이터가 정규분포에 근사함을 알 수 있다. Scale-Location plot 의 standardized 잔차들이 수평을 이룬다고 볼 수 있으므로 등분산성을 만족한다고 판단할 수 있다. 마지막으로 Residuals vs Leverage plot 을 통해 cook's distance dashed line 이 없으므로 influential point 가 거의 없지만 116 번 관측치의 leverage 가 매우 높음을 확인할 수 있다.

9. This question involves the use of multiple linear regression on the Auto data set.

(a) Produce a scatterplot matrix which includes all of the variables in the data set.



(b) Compute the matrix of correlations between the variables using the function cor(). You will need to exclude the name variables which is qualitative.

	mpg	$cylinders$	$displace-ment$	$horse-power$	$weight$	$acceler-ation$	$year$	$origin$
mpg	1	-0.78	-0.81	-0.78	-0.83	0.42	0.58	0.57
$cylinder$		1	0.95	0.84	0.90	-0.50	-0.35	-0.57
$displace-ment$			1	0.90	0.93	-0.54	-0.37	-0.61
$horse-power$				1	0.86	-0.69	-0.42	-0.46
$weight$					1	-0.42	-0.31	-0.59
$acceler-ation$						1	0.29	0.21
$year$							1	0.18
$origin$								1

(c) Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the summary() function to print the results. Comment on the output. For instance:

$Coefficients$	$Estimate$	$Std.Error$	$t-value$	$Pr(>\|t\|)$
$(Intercept)$	-17.22	4.64	-3.71	0.00024***
$cylinder$	-0.49	0.32	-1.53	0.12780
$displacement$	0.02	0.01	2.65	0.00844**
$horsepower$	-0.02	0.01	-1.23	0.21963
$weight$	-0.01	0.00	-9.93	<2e-16***
$accleration$	0.08	0.10	0.82	0.41548
$year$	0.75	0.05	14.73	<2e-16***
$origin$	1.43	0.28	5.13	4.67e-07***

RSE	3.328 on 384 df	$Adjusted\ R^2$	0.8182
R^2	0.8215	$p-value$	<2.2e-16
$F-statistics$	252.4 on 7 and 384 df		

ⅰ. Is there a relationship between the predictors and the response?

F-검정의 p-value 가 작으므로 $H_0: \beta_j = 0 \text{ for all j} = 1, \dots, 7$을 기각한다. 따라서 적어도 하나의 predictor 는 response 와 선형관계를 갖는다.

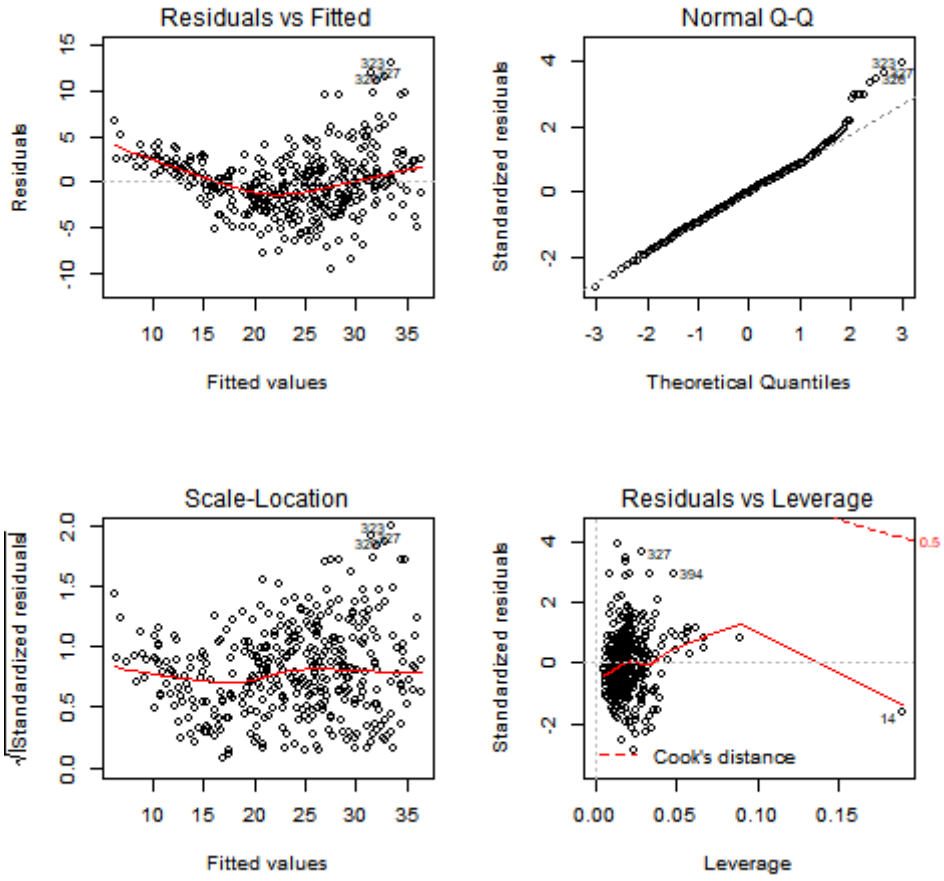
ⅱ. Which predictors appear to have a statistically significant relationship to the response?

유의수준 $\alpha = 0.05$를 기준으로 회귀계수의 t 검정 결과 displacement, weight, year 그리고 origin 이 통계적으로 유의하다고 볼 수 있다.

ⅲ. What does the coefficient for the year variable suggest?

year 의 회귀계수는 0.75 로 양수이다. 따라서 mpg 와 year 는 양의 선형관계가 있다고 말할 수 있다. 즉, 다른 predictor 들이 고정되었다고 가정하고 year 의 한 단위가 증가할 때, mpg 의 평균 증가분이 0.75 이다.

(d) Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?



Residual vs fitted plot 에서 non linear 패턴이 관측된다. 또한 323, 327, 그리고 326 번 관측치가 outlier 라는 것을 파악할 수 있다. Residual vs Leverage plot 에서 standardized residual 이 절대값 2 를 넘는 outlier 들이 존재하는 것을 알 수 있다. 또한 14 번 관측치의 경우 높은 leverage 값을 갖는다.

(e) Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

name	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	35.47889	53.13579	0.667702	0.504748
cylinders	6.988576	8.247971	0.847309	0.397381
displacement	-0.47854	0.189353	-2.52722	0.011921*
horsepower	0.503434	0.346999	1.450821	0.147693

weight	0.004133	0.017594	0.234901	0.814418
acceleration	-5.85917	2.173621	-2.69558	0.007354*
year	0.69743	0.60967	1.143947	0.2534
origin	-20.8956	7.097091	-2.94424	0.003446*
cylinders:displacement	-0.00338	0.006455	-0.52412	0.600513
cylinders:horsepower	0.011613	0.024198	0.479927	0.631568
cylinders:weight	3.57E-04	8.95E-04	0.39918	0.689995
cylinders:acceleration	0.277872	0.166422	1.669688	0.095843
cylinders:year	-0.17413	0.097141	-1.7925	0.073885
cylinders:origin	0.402168	0.492622	0.816382	0.414817
displacement:horsepower	-8.49E-05	2.88E-04	-0.29434	0.768666
displacement:weight	2.47E-05	1.47E-05	1.682049	0.093419
displacement:acceleration	-0.00348	0.003342	-1.04107	0.298534
displacement:year	0.005934	0.002391	2.482019	0.013516*
displacement:origin	0.023981	0.019465	1.231995	0.218748
horsepower:weight	-1.97E-05	2.92E-05	-0.67321	0.501243
horsepower:acceleration	-0.00721	0.003719	-1.9392	0.053252
horsepower:year	-0.00584	0.003938	-1.48218	0.139161
horsepower:origin	0.002233	0.029301	0.076191	0.939309
weight:acceleration	2.35E-04	2.29E-04	1.025183	0.30596
weight:year	-2.25E-04	2.13E-04	-1.05568	0.291816
weight:origin	-5.79E-04	0.001591	-0.36379	0.716229
acceleration:year	0.055622	0.025582	2.174265	0.030331*
acceleration:origin	0.458316	0.15666	2.925552	0.003655***
year:origin	0.139257	0.07399	1.882119	0.06062

RSE	2.695 on 363 df	$Adjusted\ R^2$	0.8808
R^2	0.8893	$p-value$	<2.2e-16
$F-statistics$	104.2 on 28 and 363 df		

유의수준 $\alpha = 0.05$ 를 기준으로 회귀계수의 t 검정 결과 interaction term 중에서 displacement:year, acceleration:year, 그리고 acceleration:origin 이 통계적으로 유의하다.

(f) Try a few different transformations of the variables, such as $log(X), \sqrt{X}, X^2$. Comment on your findings.

name 을 제외한 Auto data 를 1 차항으로 회귀 적합 후 step function 으로 가장 AIC 가 낮은 모델을 선정했다. 그 결과 아래 같은 predictor variables 들이 채택되었다.

$$mpg \sim cylinders + displacement + horsepower + weight + year + origin$$

각각의 predictor variable 들을 log, 루트, 그리고 제곱으로 transformation 시킨 후 모델들의 AIC 값을 비교해보았다.

Transformation of Predictors	*AIC*
X	949.1795
$\log(X)$	894.6317
$\sqrt{X}$	921.3492
X^2	1002.3231

따라서, predictor variable 의 아래의 수식과 같이 log transformation 한 모델이 가장 AIC 값이 작으므로 log transformation 이 적절하다고 말할 수 있다.

$$mpg \sim log(cylinders) + log(displacement) + log(horsepower) + log(weight) + log(year) + log(origin)$$

13. In this exercise you will create some simulated data and will fit simple linear regression models to it. Make sure to use set.seed(1) prior to starting part (a) to ensure consistent results.

(a) Using the rnorm() function, create a vector, x, containing 100 observations drawn from a N(0,1) distribution. This represents a feature, X.

Statistics	*Value*
$Mean(X)$	0.1089
$Standard\ deviation(X)$	0.8982

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

Statistics	*Value*
$Mean(eps)$	-0.0189
$Standard\ deviation(eps)$	0.4789

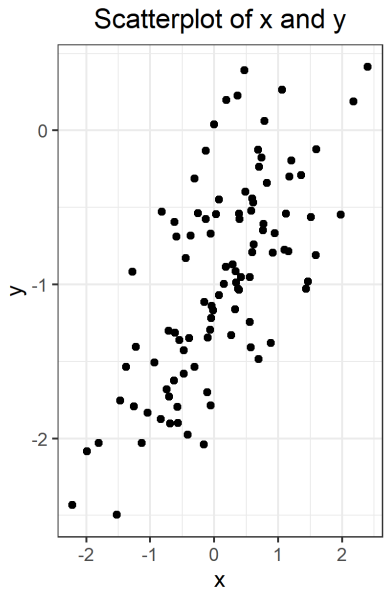
(c) Using x and eps, generate a vector y according to the model

$$\mathrm{Y} = -1 + 0.5\mathrm{X} + \epsilon. \quad (3.39)$$

What is the length of the vector y? What are the values of β_0 and β_1 in this linear model? length of y is 100.

Parameter	*Value*
β_0	-1
β_1	0.5

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



Scatterplot 을 통해 x 와 y 가 선형관계에 있음을 유추해 볼 수 있다.

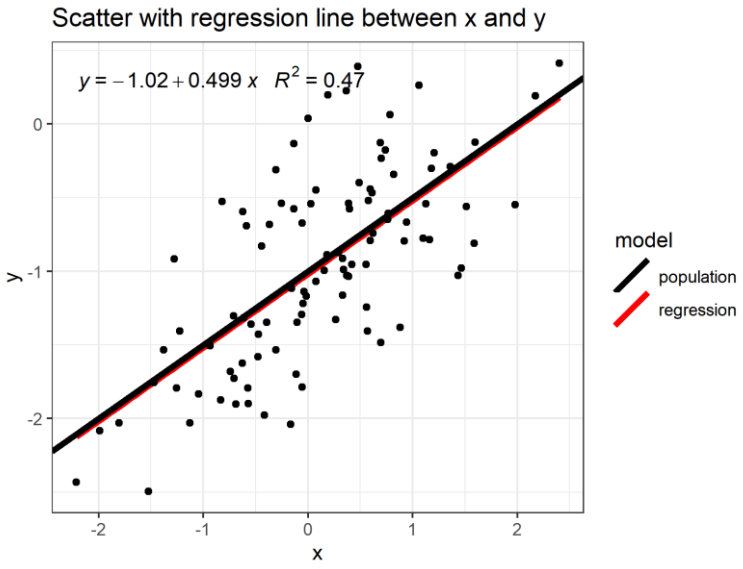
(e) Fit a least squares linear model to predict y using x. Comment on the model obtained. How do $\widehat{\beta_0}$ and $\widehat{\beta_1}$ compare to β_0 and β_1?

$Coefficients$	$Estimate$	$Std.Error$	$t-value$	$Pr(>\|t\|)$
$\widehat{\beta_0}$	-1.01885	0.04849	-21.010	<2e-16***
$\widehat{\beta_1}$	0.49947	0.05386	9.273	4.58e-15***

RSE	0.4814 on 98 df	$Adjusted\ R^2$	0.4674
R^2	0.4619	$p-value$	4.583e-15
$F-statistics$	85.99 on 1 and 98 df		

$\widehat{\beta_0}$ and $\widehat{\beta_1}$은 각각 β_0 and β_1에 근사한 값을 갖는다. 이 모델은 F-검정의 p-value가 매우 작기 때문에 null hypothesis를 기각한다.

(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() command to create an appropriate legend.



population과 regression의 직선이 거의 완벽하게 겹쳐져 구분하기 힘들다.

(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?

$Model\ 1 : y \sim x$

$Model\ 2 : y \sim x + x^2$

$Res.Df$	RSS	Df	$Sum\ of\ Sq$	F	$Pr(>F)$
98	22.709				
97	22.257	1	0.45163	1.9682	0.1638

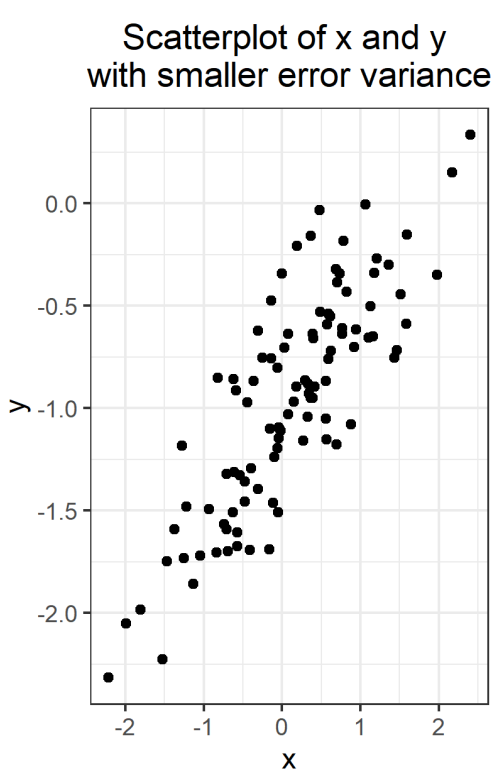
model1 과 model2 를 선형 회귀에 적합 후 anova 테스트를 한 결과 위의 표와 같은 결과가 나왔다. 귀무가설은 'model1 이 포함하지 않은 model2 의 회귀계수는 0 이다' 이며 대립가설은 'model1 이 포함하지 않은 model2 의 회귀계수 중 0 이 아닌 것이 존재한다' 이다. F-test 결과 유의 수준 $\alpha = 0.05$ 에서 귀무가설을 기각하지 못해 2 차모델보다 1 차모델이 더 적합하다는 결론을 내릴 수 있다.

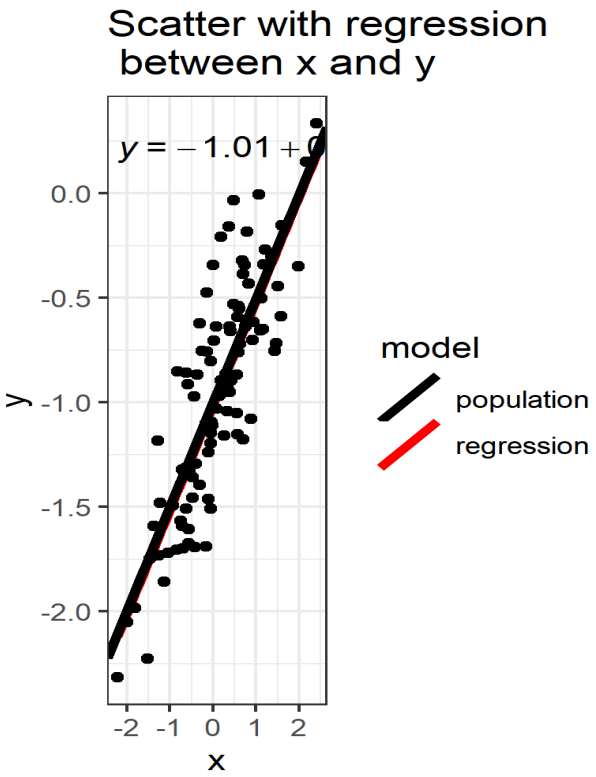
(h) Repeat (a)-(f) after modifying the data generation process in such a way that there is less noise in the data. The model(3.39) should remain the same. You can do this by decreasing the variance of the normal distribution used to generate the error term ϵ in (b). Describe your results.

(h)-(a, b) $x_1, \dots, x_{100} \sim N(0,1), \epsilon_1, \dots, \epsilon_{100} \sim N(0, 0.1)$

$Statistics$	$Value$
$Mean(X)$	0.1088874
$Standard\ deviation(X)$	0.8981994
$Mean(eps)$	-0.01195596
$Standard\ deviation(eps)$	0.302908

(h)-(d,e,f)





$Coefficients$	$Estimate$	$Std.Error$	$t-value$	$Pr(>\|t\|)$
$\widehat{\beta_0}$	-1.01192	0.03067	-32.99	<2e-16 ***
$\widehat{\beta_1}$	0.49966	0.03407	14.67	<2e-16 ***

RSE	0.3044 on 98 df	$Adjusted\ R^2$	0.6838
R^2	0.687	$p-value$	< 2.2e-16
$F-statistics$	215.1 on 1 and 98 DF		

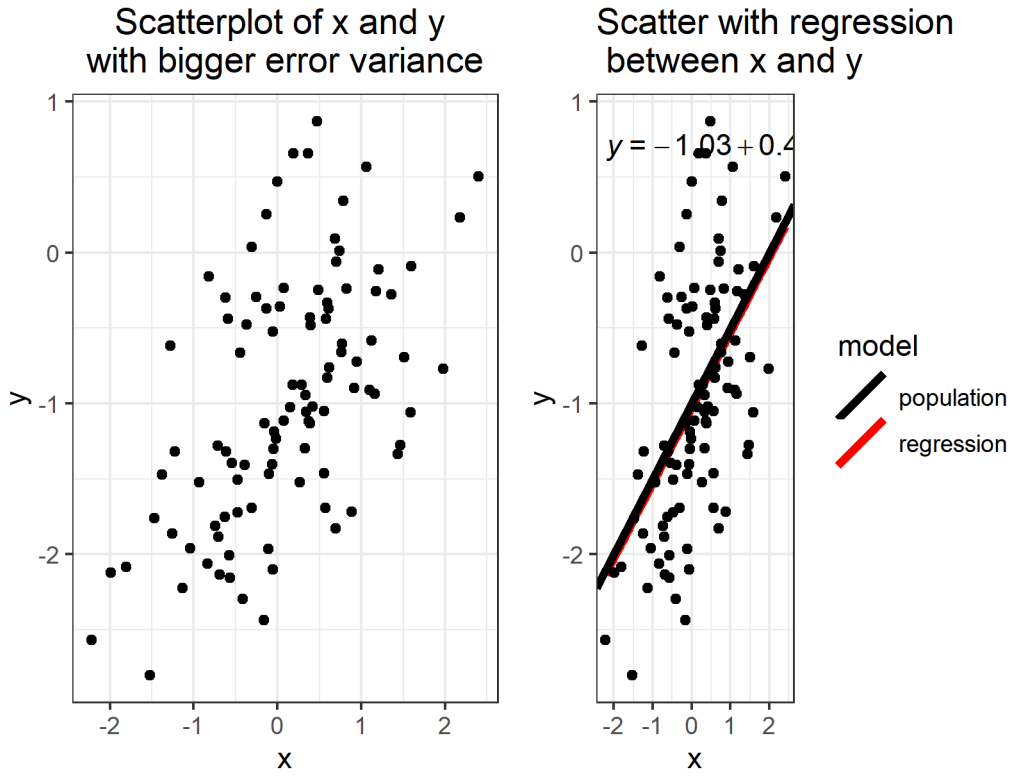
scatterplot 을 통해 오차의 분산의 감소로 인해 x 와 y 가 선형관계가 더욱 뚜렷해졌음을 알 수 있다. 또한 ϵ의 분산이 0.25 일 때보다 추정된 회귀계수 $\widehat{\beta_0}$와 $\widehat{\beta_1}$의 standard error 도 더 작아졌으며 참값에 더 가까워졌다. Regression 과 population 직선이 거의 겹쳐져 구분이 어려워졌다. 또한 R^2도 더 높아져 ϵ의 분산이 작은 데이터가 true model 을 더 잘 적합 한다고 말할 수 있다.

(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.

(i)-(a, b) $x_1, \dots, x_{100} \sim N(0,1), \epsilon_1, \dots, \epsilon_{100} \sim N(0, 0.5)$

Statistics	*Value*
$Mean(X)$	0.1088874
$Standard\ deviation(X)$	0.8981994
$Mean(eps)$	-0.02673435
$Standard\ deviation(eps)$	0.6773228

(i)-(d,e,f)



Coefficients	*Estimate*	*Std. Error*	$t - value$	$Pr(>\|t\|)$
$\widehat{\beta_0}$	-1.02665	0.06858	-14.970	< 2e-16 ***
$\widehat{\beta_1}$	0.49925	0.07617	6.554	2.62e-09 ***

RSE	0.6808 on 98 df	*Adjusted* R^2	0.2976
R^2	0.3047	$p - value$	2.624e-09
$F - statistics$	42.96 on 1 and 98 DF		

ϵ 의 분산이 커짐에 따라 Scatterplot 에서 뚜렷한 선형성을 파악하기가 어려워졌다. 또한 회귀 추정선도 population 직선과 멀어졌다. 추정된 회귀계수의 standard error 도 커졌다. R^2 도 작아져 ϵ 이 커짐에 따라 데이터의 선형관계 정도가 약해짐을 알 수 있다.

(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.

$Data$	$confidence\ interval\ of\ \widehat{\beta_0}$	$confidence\ interval\ of\ \widehat{\beta_1}$
$Original$	(-1.1150804, 0.3925794)	(-0.9226122, 0.6063602)
$less\ noisy$	(-1.0727832, 0.4320613)	(-0.9510557, 0.5672681)
$noisier$	(-1.1627482, 0.3480843)	(-0.8905572, 0.6504160)

noise 가 커질수록 신뢰구간이 넓어진다는 것을 확인할 수 있다.

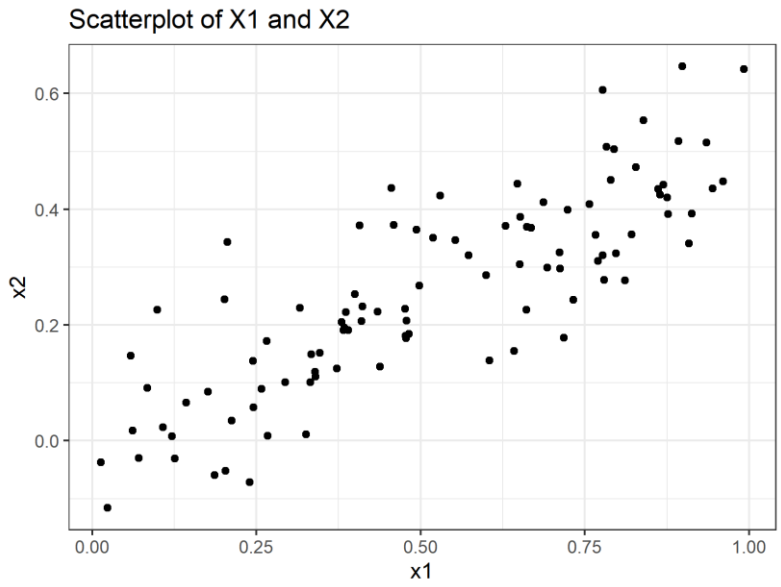
14. This problem focuses on the collinearity problem.

(a)

$y = 2 + 2 * X_1 + 0.3 * X_2 + \epsilon$ where $\beta_0 = 2, \beta_1 = 2, \beta_2 = 0.3$

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

$\mathrm{cor}(X_1, X_2) = 0.84$



(c) Using this data, fit a least squares regression to predict y using x1 and x2. Describe the results obtained. What $\widehat{\beta_0}$, $\widehat{\beta_1}$, and $\widehat{\beta_2}$? How do these relate to the true β_0, β_1, and β_2? Can you reject the null hypothesis $H_0: \beta_1 = 0$? How about the null hypothesis $H_0: \beta_2 = 0$?

$Coefficients$	$Estimate$	$Std.Error$	$t-value$	$Pr(>\|t\|)$
$\widehat{\beta_0}$	2.1305	0.2319	9.188	7.61e-15 ***
$\widehat{\beta_1}$	1.4396	0.7212	1.996	0.0487 *
$\widehat{\beta_2}$	1.0097	1.1337	0.891	0.3754

RSE	1.056 on 97 df	$Adjusted\ R^2$	0.1925
R^2	0.2088	$p-value$	1.164e-05
$F-statistics$	12.8 on 2 and 97 DF		

$\widehat{\beta_0}$, $\widehat{\beta_1}$, and $\widehat{\beta_2}$의 값은 위와 같다. $\widehat{\beta_0}$만 true β_0에 가깝다. 유의수준 $\alpha = 0.05$라 할 때 $H_0: \beta_1 = 0$을 가까스로 기각한다. $H_0: \beta_2 = 0$는 기각하지 못한다.

(d) Now fit a least squares regression to predict y using only x1. Comment on your results. Can you reject the null hypothesis $H_0: \beta_1 = 0$?

$Coefficients$	$Estimate$	$Std.Error$	$t-value$	$Pr(>\|t\|)$
$\widehat{\beta_0}$	2.1124	0.2307	9.155	8.27e-15 ***
$\widehat{\beta_1}$	1.9759	0.3963	4.986	2.66e-06 ***

RSE	1.055 on 98 df	$Adjusted\ R^2$	0.1942
R^2	0.2024	$p-value$	2.661e-06
$F-statistics$	24.86 on 1 and 98 DF		

유의수준 $\alpha = 0.05$라 할 때 $\widehat{\beta_1}$의 t test 결과 p-value 가 매우 작으므로 $H_0: \beta_1 = 0$를 기각한다.

(e) Now fit a least squares regression to predict "y" using only "x2". Comment on your results. Can you reject the null hypothesis $H_0: \beta_1 = 0$?

$Coefficients$	$Estimate$	$Std.Error$	$t-value$	$Pr(>\|t\|)$
$\widehat{\beta_0}$	2.3899	0.1949	12.26	< 2e-16 ***
$\widehat{\beta_1}$	2.8996	0.6330	4.58	1.37e-05 ***

RSE	1.072 on 98 df	$Adjusted\ R^2$	0.1679
R^2	0.1763	$p-value$	1.366e-05
$F-statistics$	20.98 on 1 and 98 DF		

유의수준 $\alpha = 0.05$라 할 때 $\widehat{\beta_1}$의 t test 결과 p-value 가 매우 작으므로 $H_0: \beta_1 = 0$를 기각한다.

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

x1 과 x2 를 동시에 회귀 직선에 적합 시켰을 때 각 회귀계수의 t test 의 p-value 값이 컸고 심지어 x2 의 회귀계수는 유의하지 않다는 결론을 얻었다. 그러나 x1 과 x2 를 각각 회귀 직선에 적합 시켰을 때 x1 과 x2 의 각 회귀계수는 유의하다는 결과를 얻었다. 이는 x1 과 x2 의 높은 상관관계로 인한 다중공선성 문제에서 발생한 문제이다. 따라서 회귀모형에서 다중공선성 관계에 있는 두 개의 변수 x1 과 x2 를 동시에 적합 시켰을 때 회귀계수의 추정에 오류가 생길 수 있는 것이다.

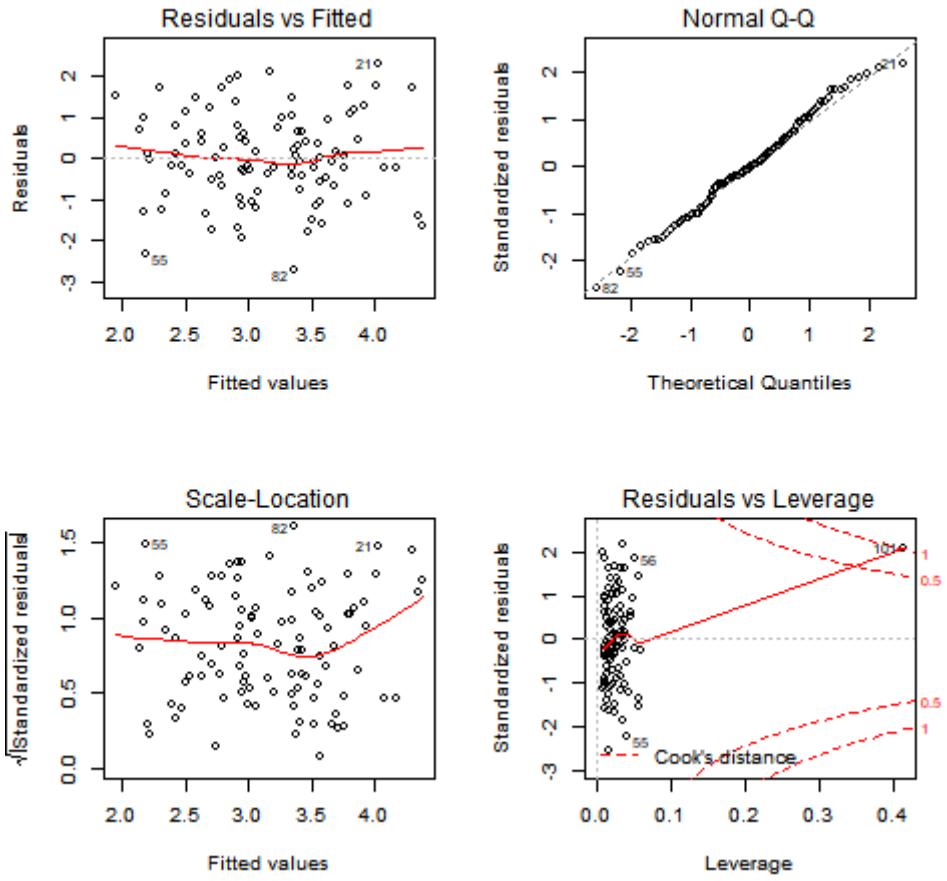
(g) Now suppose we obtain one additional observation, which was unfortunately mismeasured. Re-fit the linear models from (c) to (e) using this new data. What effect does this new observation have on the each of the models? In each model, is this observation an outlier? A high-leverage point? Both? Explain your answers.

(c)

$Coefficients$	$Estimate$	$Std.Error$	$t-value$	$Pr(>\|t\|)$
$\widehat{\beta_0}$	2.2267	0.2314	9.624	7.91e-16 ***
$\widehat{\beta_1}$	0.5394	0.5922	0.911	0.36458
$\widehat{\beta_2}$	2.5146	0.8977	2.801	0.00614 **

RSE	1.075 on 98 df	$Adjusted\ R^2$	0.2029
R^2	0.2188	$p-value$	5.564e-06
$F-statistics$	13.72 on 2 and 98 DF		

새로운 관측치 추가 후 x1 과 x2 를 동시에 선형 회귀 적합한 결과가 위와 같다. 관측치 추가 전과 달리 유의수준 $\alpha = 0.05$라 할 때 $H_0: \beta_1 = 0$ 대신 $H_0: \beta_2 = 0$을 기각한다.



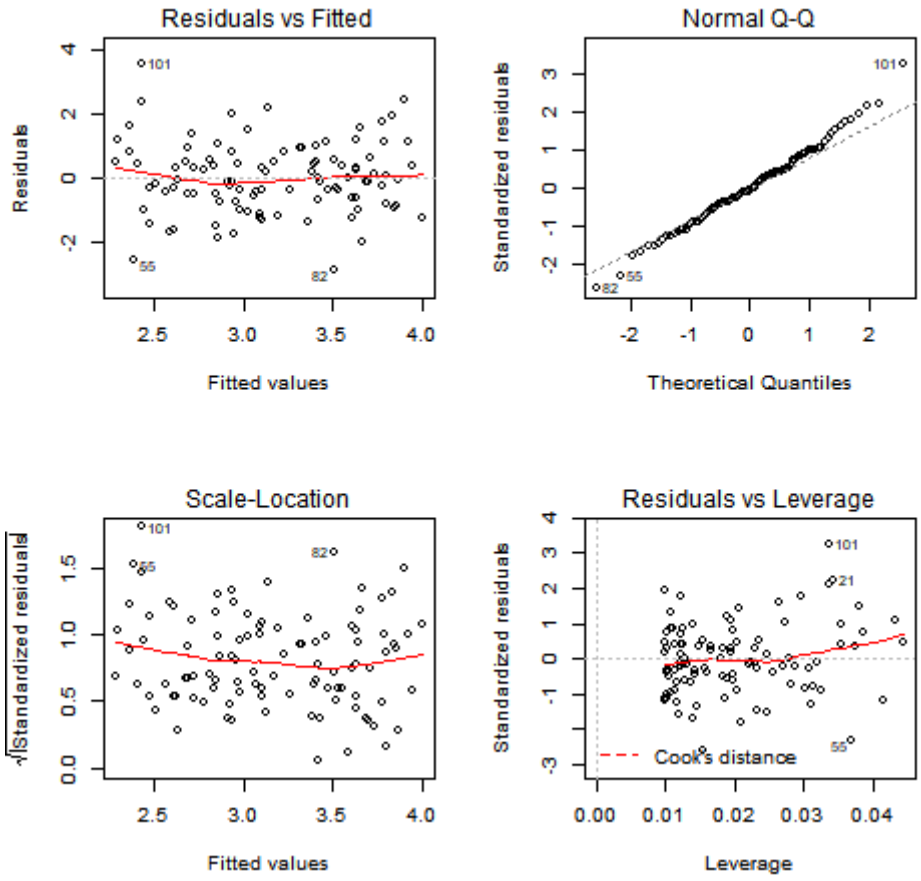
Residuals vs Leverage plot 을 통해 새롭게 추가된 101 번 관측치가 cook's distance 경계를 넘었으므로 high leverage point 라는 것을 확인할 수 있다.

(d)

$Coefficients$	$Estimate$	$Std.Error$	$t-value$	$Pr(>\|t\|)$
$\widehat{\beta_0}$	2.2569	0.2390	9.445	1.78e-15 ***
$\widehat{\beta_1}$	1.7657	0.4124	4.282	4.29e-05 ***

RSE	1.111 on 99 df	$Adjusted\ R^2$	0.1477
R^2	0.1562	$p-value$	4.295e-05
$F-statistics$	18.33 on 1 and 99 DF		

새로운 관측치 추가 후 x1 만 선형 회귀 적합한 결과가 위와 같다. 유의수준 $\alpha = 0.05$라 할 때 $H_0: \beta_1 = 0$ 을 기각한다.



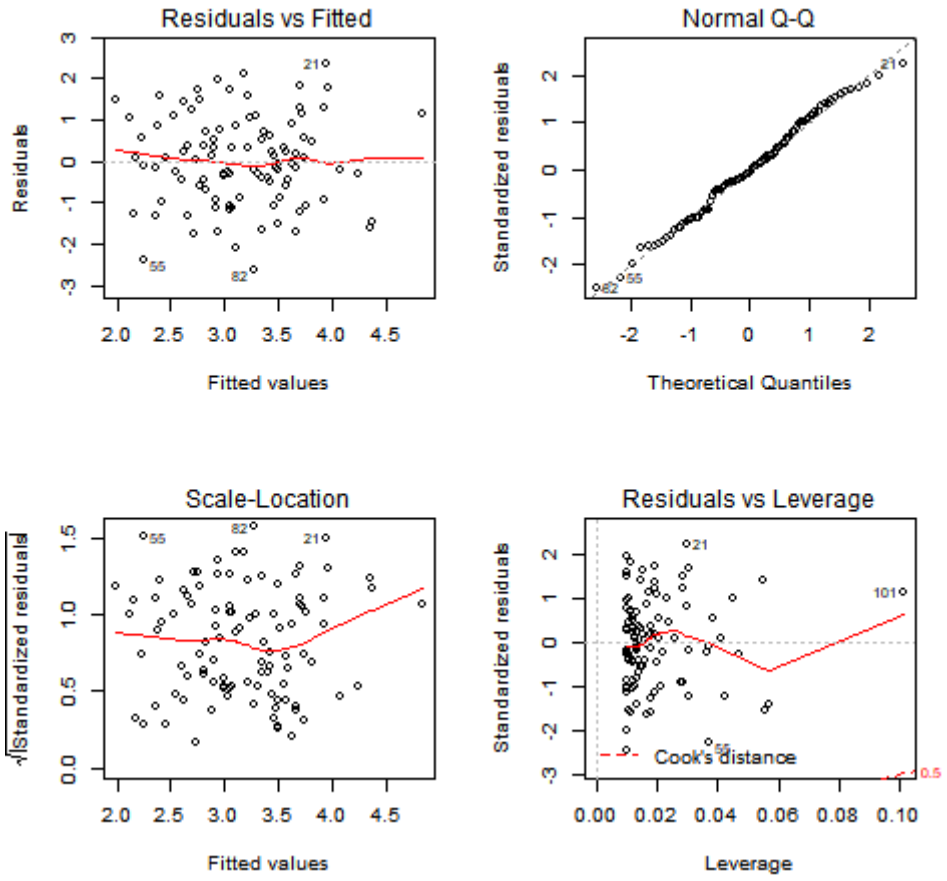
Residual vs Fitted plot 과 Residuals vs Leverage plot 을 통해 x1 만 적합한 모델에서 새롭게 추가된 관측치 101 번은 standardized residual 의 절대치 2 를 넘기 때문에 outlier 로 판정할 수 있다.

(e)

$Coefficients$	$Estimate$	$Std.Error$	$t-value$	$Pr(>\|t\|)$
$\widehat{\beta_0}$	2.3451	0.1912	12.264	< 2e-16 ***
$\widehat{\beta_1}$	3.1190	0.6040	5.164	1.25e-06 ***

RSE	1.074 on 99 df	$Adjusted\ R^2$	0.2042
R^2	0.2122	$p-value$	1.253e-06
$F-statistics$	26.66 on 1 and 99 DF		

새로운 관측치 추가 후 x2 만 선형 회귀 적합한 결과가 위와 같다. 유의수준 $\alpha = 0.05$라 할 때 $H_0: \beta_1 = 0$ 을 기각한다.



Residual vs Fitted plot 과 Residuals vs Leverage plot 을 통해 x2 만 적합한 모델에서 새롭게 추가된 관측치 101 번은 outlier 는 아니지만 비교적 높은 leverage 를 갖고 있어 high leverage point 에 해당한다고 볼 수 있다.

3. Discussion

이번 과제를 통해 Linear regression 의 해석 그리고 모형 진단 방법에 대한 이론적인 내용과 R 실습을 공부할 수 있었다. Linear regression 은 모형의 해석이 용이하다는 장점이 있지만 제약조건이 많기 때문에 그만큼 고려해야할 사항도 많다. 교재 ISL(II)는 회귀 적합 시 고려해야할 이슈 6 가지 문제로 Non-linearity of the response-predictor relationship, Correlation of error terms, Non-constant variance of error terms, Outliers, High-leverage point, 그리고 Collinearity 을 소개했다. 또한 3.6 Lab 을 통해 회귀 적합과 모형을 진단하는 코드를 배울 수 있었다. lm()함수의 argument 기능을 통해 다양한 회귀식을 적합할 수 있고 confint(), prediction, 그리고 plot() 함수를 통해 회귀 모형을 해석하고 진단할 수 있다는 것을 배웠다. 마지막으로 위에서 배운 이론과 R 코드를 연습문제에 적용하면서 linear regression 을 전반적으로 깊게 학습할 수 있었다.

4. Appendix

```
pacman::p_load(ggplot2, tidyverse, dplyr,
plotly, processx)

rm(list=ls(all=TRUE))



# HW3 - exercise 8

library(ISLR)

fix(Auto)

attach(Auto)

lm.fit <- lm(mpg ~ horsepower)

summary(lm.fit)


predict(lm.fit,
data.frame(horsepower=c(98)), interval =
"confidence")

predict(lm.fit,
data.frame(horsepower=c(98)), interval =
"prediction")


library(ggpmisc)

my.formula <- y ~ x

ggplot(aes(x = horsepower, y = mpg), data =
Auto) +

  geom_smooth(method = "lm", se=FALSE,
color="red",size = 1.5, formula =
my.formula) +

  stat_poly_eq(formula = my.formula,

               aes(label =
paste(..eq.label.., ..rr.label.., sep =
"~~~")),

               parse = TRUE, label.x = "right",
label.y = "top") +

  geom_point()+ labs(title = "Scatter with
regression line between mpg and horsepower")
+ theme(plot.title = element_text(hjust =
0.5, size = 15)) + theme_bw()

ggsave("reg.png")


png("diag.png")

par(mfrow = c(2,2))

plot(lm.fit)

dev.off()


# HW3 - exercise 9

png("all.png")

pairs(Auto)

dev.off()


cor(Auto[,1:8])


lm.fit2 <- lm(mpg ~.-name, data = Auto)

summary(lm.fit2)


png("diag2.png")

par(mfrow = c(2,2))

plot(lm.fit2)

dev.off()


lm.fit3 <- lm(mpg ~.^2, data = Auto[,1:8])

sum_lm <- summary(lm.fit3)

sum_lm <- as.data.frame(sum_lm$coefficients)

sum_lm$name <- rownames(sum_lm)

write_csv(sum_lm, "summary.csv")


step(lm.fit2)

lm.fit4 <- lm(mpg ~ log(cylinders) +
log(displacement) + log(horsepower) +
log(weight) + log(year) + log(origin), data
= Auto)

lm.fit5 <- lm(formula = mpg ~ cylinders +
displacement + horsepower + weight +

              year + origin, data = Auto)

lm.fit6 <- lm(mpg ~ sqrt(cylinders) +
sqrt(displacement) + sqrt(horsepower) +
sqrt(weight) + sqrt(year) +
sqrt(origin),data = Auto)

lm.fit7 <- lm(mpg ~ I(cylinders^2) +
I(displacement^2) +
I(horsepower^2)+I(weight^2) + I(year^2) +
I(origin^2), data = Auto)

data.frame(AIC = c(extractAIC(lm.fit4)[2],
extractAIC(lm.fit5)[2],extractAIC(lm.fit6)[2
],extractAIC(lm.fit7)[2]))


library(gridExtra)

ggplot(aes(mpg, cylinders), data = Auto) +
geom_point()


#HW3 - exercise 13

set.seed(1)

X <- rnorm(100, 0, 1)

mean(X); sd(X)


eps <- rnorm(100, 0, sqrt(0.25))

mean(eps); sd(eps)


y <- -1 + 0.5*X + eps

mean(y); sd(y)


df <- data.frame(x = X, y = y)

ggplot(aes(x=x, y = y), data = df) +
geom_point() + theme_bw() +
```

```
  labs(title = "Scatterplot of x and y") +
theme(plot.title = element_text(hjust =
0.5))

ggsave('scatter.png')


fit <- lm(y ~ X)

summary(fit)


my.formula = y ~ x

ggplot(aes(x = x, y = y), data = df) +

  geom_smooth(aes(col="regression"), method =
"lm", se=FALSE, size = 1.5, formula =
my.formula, show.legend = FALSE) +

  stat_poly_eq(formula = my.formula,

               aes(label =
paste(..eq.label.., ..rr.label.., sep =
"~~~")),

               parse = TRUE, label.x = "left",
label.y = "top") +

  geom_point()+ labs(title = "Scatter with
regression line between x and y") +
theme(plot.title = element_text(hjust = 0.5,
size = 15)) + theme_bw()+

geom_abline(aes(col = "population", slope =
0.5, intercept = -1), size = 1.5) +
scale_color_manual("model", values = c(1,
2))

ggsave('comp.png')


fit2 <- lm(y ~ x + I(x^2), data = df)

anova(fit, fit2)




#h

set.seed(1)

X <- rnorm(100, 0, 1)

eps <- rnorm(100, 0, sqrt(0.1))

mean(X); sd(X); mean(eps); sd(eps)

y <- -1 + 0.5 * X  + eps

df <- data.frame(x = X, y = y)


ggplot(aes(x=x, y = y), data = df) +
geom_point() + theme_bw() +

  labs(title = "Scatterplot of x and y \nwith
smaller error variance") + theme(plot.title
= element_text(hjust = 0.5))

ggsave('last.png')


fit_1 <- lm(y ~ x, data =df)

summary(fit_1)


my.formula = y ~ x

ggplot(aes(x = x, y = y), data = df) +

  geom_smooth(aes(col="regression"), method =
"lm", se=FALSE, size = 1.5, formula =
my.formula, show.legend = FALSE) +

  stat_poly_eq(formula = my.formula,

               aes(label =
paste(..eq.label.., ..rr.label.., sep =
"~~~")),

               parse = TRUE, label.x = "left",
label.y = "top") +

  geom_point()+ labs(title = "Scatter with
regression \n between x and y") +
theme(plot.title = element_text(hjust = 0.5,
size = 15)) + theme_bw()+

  geom_abline(aes(col = "population", slope =
0.5, intercept = -1), size = 1.5) +
scale_color_manual("model", values = c(1,
2))

ggsave('comp2.png')


#i

set.seed(1)

X <- rnorm(100, 0, 1)

eps <- rnorm(100, 0, sqrt(0.5))

mean(X); sd(X); mean(eps); sd(eps)

y <- -1 + 0.5 * X  + eps

df <- data.frame(x = X, y = y)


ggplot(aes(x=x, y = y), data = df) +
geom_point() + theme_bw() +

  labs(title = "Scatterplot of x and y \nwith
bigger error variance") + theme(plot.title =
element_text(hjust = 0.5))

ggsave('scatter_big.png')


fit_1 <- lm(y ~ x, data =df)

summary(fit_1)


my.formula = y ~ x

ggplot(aes(x = x, y = y), data = df) +

  geom_smooth(aes(col="regression"), method =
"lm", se=FALSE, size = 1.5, formula =
my.formula, show.legend = FALSE) +

  stat_poly_eq(formula = my.formula,

               aes(label =
paste(..eq.label.., ..rr.label.., sep =
"~~~")),

               parse = TRUE, label.x = "left",
label.y = "top") +

  geom_point()+ labs(title = "Scatter with
regression \n between x and y") +
theme(plot.title = element_text(hjust = 0.5,
size = 15)) + theme_bw()+

  geom_abline(aes(col = "population", slope =
0.5, intercept = -1), size = 1.5) +
```

```
scale_color_manual("model", values = c(1,
2))
ggsave('comp3.png')



#j
my_function <- function(var){
  set.seed(1)
  x <- rnorm(100, 0, 1)
  eps <- rnorm(100, 0, sqrt(var))
  y <- -1 + 0.5*x +eps
  fit <- lm(y ~ x)
  result = confint(fit)
  return(result)
}
var <- c(0.25, 0.1, 0.5)
sapply(var, function(var) my_function(var))




# HW3 - exercise 14
#a.
set.seed(1)
x1 <- runif(100)
x2 <- 0.5 * x1 + rnorm(100)/10
y <- 2 + 2*x1+0.3*x2+rnorm(100)

#b.
cor(x1, x2)
df <- data.frame(yy = y, x1 = x1, x2= x2)
ggplot(aes(x=x1, y=x2), data=df) +
geom_point()+
  theme(plot.title = element_text(hjust =
0.5))+
  labs(title = "Scatterplot of X1 and X2")+
  theme_bw()
ggsave('plot1.png')

#c.
fit <- lm(yy ~ x1 + x2, data =df)
summary(fit)

#d.
lm_fit <- lm(yy ~ x1, data = df)
summary(lm_fit)

#e.
lm_fit2 <- lm(yy ~ x2, data = df)
summary(lm_fit2)

#g.
x1 <- c(x1, 0.1)
x2 <- c(x2, 0.8)
y <- c(y, 6)
df <- data.frame(y = y, x1 = x1, x2 = x2)

#g-c
fit <- lm(y ~ x1 + x2, data = df)
summary(fit)
png("fit1.png")
par(mfrow = c(2,2))
plot(fit)
dev.off()

#g-d
lm_fit <- lm(y ~ x1, data = df)
summary(lm_fit)
png("fit2.png")
par(mfrow = c(2,2))
plot(lm_fit)
dev.off()

#g-e
lm_fit2 <- lm(y ~ x2, data = df)
summary(lm_fit2)
png("fit3.png")
par(mfrow = c(2,2))
plot(lm_fit2)
dev.off()
```