회귀분석 R 과제

20210858 정보통계학과 이지윤

1 차 과제

#**2.2**

```
#2.2

A = matrix(c(1,-1,2,  -1,3,3,  2,3,4),nc=3)
B = matrix(c(3,-2,4,  -2,1,0,  4,0,2),nc=3)
x = c(1,2,3)
y = c(3,4,5)
```

(a)

```
#a
A+B
```

```
> #a
> A+B
     [,1] [,2] [,3]
[1,]    4   -3    6
[2,]   -3    4    3
[3,]    6    3    6
```

(b)

```
#b
t(B)
```

```
> #b
> t(B)
     [,1] [,2] [,3]
[1,]    3   -2    4
[2,]   -2    1    0
[3,]    4    0    2
```

(c)

```
#c
t(x)%*%A%*%y

```

```
> #c
> t(x)%*%A%*%y
     [,1]
[1,]  171
>
```

(d)

```
#d
t(x)%*%x
```

```
> #d
> t(x)%*%x
     [,1]
[1,]   14
>
```

(e)

```
#e
t(x)%*%A%*%x
```

```
> #e
> t(x)%*%A%*%x
     [,1]
[1,]   93
```

(f)

```
#f
t(x)%*%y
```

```
> #f
> t(x)%*%y
     [,1]
[1,]   26
```

(g)

```
#g
t(A)%*%A
```

```
> #g
> t(A)%*%A
     [,1] [,2] [,3]
[1,]    6    2    7
[2,]    2   19   19
[3,]    7   19   29
```

(h)

```
#h
A%*%B
```

```
> #h
> A%*%B
     [,1] [,2] [,3]
[1,]   13   -3    8
[2,]    3    5    2
[3,]   16   -1   16
>
```

(i)

```
#i
t(y)%*%B
```

```
> #i
> t(y)%*%B
     [,1] [,2] [,3]
[1,]   21   -2   22
>
```

(j)

```
#j
x%*%t(x)
```

```
> #j
> x%*%t(x)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    4    6
[3,]    3    6    9
>
```

(k)

```
#k
x+y
```

```
> #k
> x+y
[1] 4 6 8
>
```

(l)

```
#l
x-y
```

```
> #l
> x-y
[1] -2 -2 -2
>
```

(m)

```
#m
t(x-y)
```

```
> #m
> t(x-y)
     [,1] [,2] [,3]
[1,]   -2   -2   -2
>
```

(n)

```
#n
x%*%t(y)
```

```
> #n
> x%*%t(y)
     [,1] [,2] [,3]
[1,]    3    4    5
[2,]    6    8   10
[3,]    9   12   15
>
```

(o)

```
#o
A-B
```

```
> #o
> A-B
     [,1] [,2] [,3]
[1,]   -2    1   -2
[2,]    1    2    3
[3,]   -2    3    2
>
```

(p)

```
#p
t(A)+t(B)
```

```
> #p
> t(A)+t(B)
     [,1] [,2] [,3]
[1,]    4   -3    6
[2,]   -3    4    3
[3,]    6    3    6
```

(q)

```
#q
t(A+B)
```

```
> #q
> t(A+B)
     [,1] [,2] [,3]
[1,]    4   -3    6
[2,]   -3    4    3
[3,]    6    3    6
```

(r)

```
#r
3*x
```

```
> #r
> 3*x
[1] 3 6 9
```

(s)

```
#s
(t(x)%*%y)**2
```

```
> #s
> (t(x)%*%y)**2
     [,1]
[1,]  676
```

(t)

```r
#t
B%*%A
```

```
> #t
> B%*%A
     [,1] [,2] [,3]
[1,]   13    3   16
[2,]   -3    5   -1
[3,]    8    2   16
```

(u)

```r
#u
library(Matrix)
rankMatrix(A)
```

```
> #u
> library(Matrix)
> rankMatrix(A)
[1] 3
attr(,"method")
[1] "tolNorm2"
attr(,"useGrad")
[1] FALSE
attr(,"tol")
[1] 6.661338e-16
```

(v)

```r
#v
library(MASS)
ginv(A)
```

```
> #v
> library(MASS)
> ginv(A)
      [,1]          [,2]  [,3]
[1,] -0.12 -4.000000e-01  0.36
[2,] -0.40  8.326673e-17  0.20
[3,]  0.36  2.000000e-01 -0.08
>
```

(w)

```r
#w
sum(diag(A))
sum(diag(B))
```

```
> #w
> sum(diag(A))
[1] 8
> sum(diag(B))
[1] 6
```

(x)

```r
#x
C = sum(diag(A+B))
D = sum(diag(A)) + sum(diag(B))
C==D
```

```
> #x
> C = sum(diag(A+B))
> D = sum(diag(A)) + sum(diag(B))
> C==D
[1] TRUE
```

(y)

```r
#y
E = sum(diag(A%*%B))
G = sum(diag(B%*%A))
E == G
```

```
> #y
> E = sum(diag(A%*%B))
> G = sum(diag(B%*%A))
> E == G
[1] TRUE
```

(z)

```
#z
O = t(A%*%B)
P = t(B)%*%t(A)
O==P
```

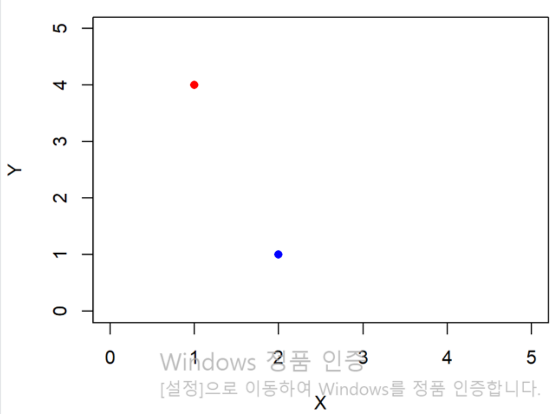
```
> #z
> O = t(A%*%B)
> P = t(B)%*%t(A)
> O==P
     [,1] [,2] [,3]
[1,] TRUE TRUE TRUE
[2,] TRUE TRUE TRUE
[3,] TRUE TRUE TRUE
> 
```

#2.3

```
#2.3
A = matrix(c(2,1,  1,4),nc=2)
```

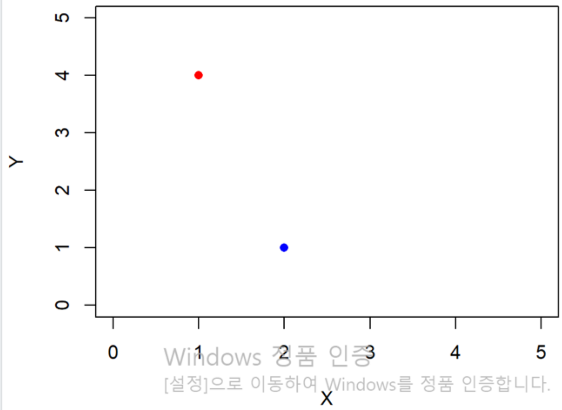
(a)

```
#a
b1 = matrix(c(2,1),nc=2)
b2 = matrix(c(1,4),nc=2)
plot(b1[,1], b1[,2], type = "p", pch = 16, col = "blue", xlim = c(0, 5), ylim = c(0, 5), xlab = "X", ylab = "Y")
points(b2[,1], b2[,2], pch = 16, col = "red")
```



(b)

```r
#b
a1 = c(2,1)
a2 = c(1,4)
plot(a1[1], a1[2], type = "p", pch = 16, col = "blue", xlim = c(0, 5), ylim = c(0, 5), xlab = "X", ylab = "Y")
points(a2[1], a2[2], pch = 16, col = "red")
```



(c)

```r
#c
library(Matrix)
rankMatrix(A)
```

```
> #c
> library(Matrix)
> rankMatrix(A)
[1] 2
attr(,"method")
[1] "tolNorm2"
attr(,"useGrad")
[1] FALSE
attr(,"tol")
[1] 4.440892e-16
```

(d)

```r
#d
eigen(A)
```

```
> #d
> eigen(A)
eigen() decomposition
$values
[1] 4.414214 1.585786

$vectors
          [,1]       [,2]
[1,] 0.3826834 -0.9238795
[2,] 0.9238795  0.3826834
```

(e)

```r
#e
library(MASS)
ginv(A)
```

```
> #e
> library(MASS)
> ginv(A)
           [,1]       [,2]
[1,]  0.5714286 -0.1428571
[2,] -0.1428571  0.2857143
> 
```

#2.5

```r
#2.5

A = matrix(c(5,-4,3,  -4,8,6,  3,6,9),nc=3)
```

(a)

```r
#a
eigen(A)
```

```
> #a
> eigen(A)
eigen() decomposition
$values
[1] 14.554216  8.844169 -1.398385

$vectors
            [,1]       [,2]       [,3]
[1,] -0.06655815  0.7859942 -0.6146407
[2,]  0.69553198 -0.4051259 -0.5933871
[3,]  0.71540567  0.4669970  0.5197197
```

(b)

```r
#b
B = sum(diag(A))
C = 14.554216 + 8.844169  + (-1.398385)
B == C
```

```
> #b
> B = sum(diag(A))
> C = 14.554216 + 8.844169  + (-1.398385)
> B == C
[1] TRUE
```

(c)

```
#c
D = det(A)
E = 14.554216 * 8.844169  * (-1.398385)
D == E

```

```
> #c
> D = det(A)
> E = 14.554216 * 8.844169  * (-1.398385)
> D == E
[1] FALSE
```

(d)

```
#d
library(MASS)
ginv(A)

```

```
> #d
> library(MASS)
> ginv(A)
           [,1]       [,2]       [,3]
[1,] -0.2000000 -0.3000000  0.2666667
[2,] -0.3000000 -0.2000000  0.2333333
[3,]  0.2666667  0.2333333 -0.1333333
>
```

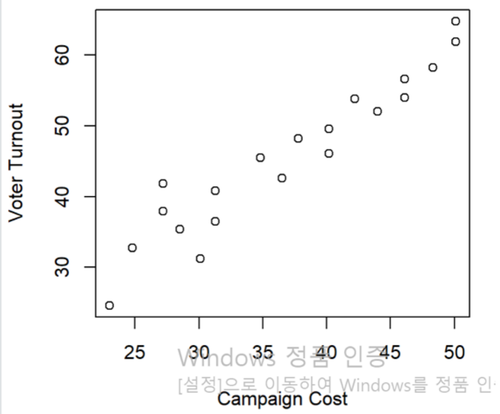
2 차 과제

3.9

```r
tv <- data.frame(
  x = c(28.5, 48.3, 40.2, 34.8, 50.1, 44.0, 27.2, 37.8, 27.2, 46.1,
        31.3, 50.1, 31.3, 24.8, 42.2, 23.0, 30.1, 36.5, 40.2, 46.1 ),
  y = c(35.4, 58.2, 46.1, 45.5, 64.8, 52.0, 37.9, 48.2, 41.8, 54.0,
        40.8, 61.9, 36.5, 32.7, 53.8, 24.6, 31.2, 42.6, 49.6, 56.6)
)
```

#a

```r
#a
plot(tv$x, tv$y ,
     xlab = "Campaign Cost",
     ylab = "Voter Turnout" )
```



#b

```r
#b
cor(tv$x, tv$y)
```

```
> #b
> cor(tv$x, tv$y)
[1] 0.9540002
>
```

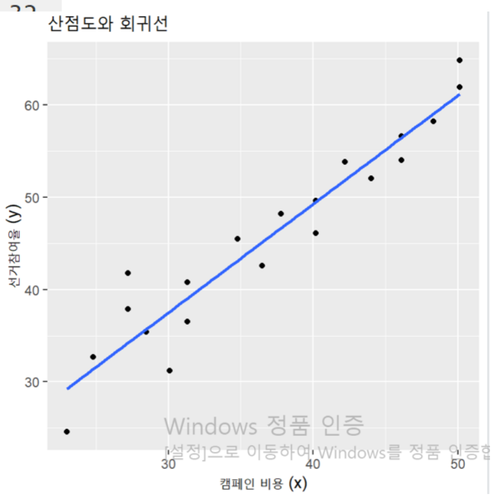
#c

```r
#c
tvlm = lm(tv$x ~ tv$y)
summary(tvlm)
```

```
> summary(tvlm)

Call:
lm(formula = tv$x ~ tv$y)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.7665 -1.7122  0.3963  1.9727  4.3302 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.64365    2.68692   0.612    0.548    
tv$y         0.77327    0.05728  13.500 7.41e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.701 on 18 degrees of freedom
Multiple R-squared:  0.9101,	Adjusted R-squared:  0.9051 
F-statistic: 182.3 on 1 and 18 DF,  p-value: 7.409e-11
```

#d

```r
#d
# ggplot2 패키지 설치 (한 번만 실행)
install.packages("ggplot2")

library(ggplot2)

ggplot(tv, aes(x = x, y = y)) +
  geom_point() +  # 산점도
  geom_smooth(method = "lm", se = FALSE) +  # 회귀선 그리기 (lm: 선형회귀)
  labs(x = "캠페인 비용 (x)", y = "선거참여율 (y)") +  # 축 레이블 지정
  ggtitle("산점도와 회귀선")  # 그래프 제목
```



#e

```r
#e
cor.test(x=tv$x, y=tv$y)
```

```
> #e
> cor.test(x=tv$x, y=tv$y)

	Pearson's product-moment correlation

data:  tv$x and tv$y
t = 13.5, df = 18, p-value = 7.409e-11
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8851650 0.9819684
sample estimates:
      cor 
0.9540002 
```

#f

```r
#f
confidence_interval <- confint(tvlm)
confidence_interval
```

```
> #f
> confidence_interval <- confint(tvlm)
> confidence_interval
                 2.5 %    97.5 %
(Intercept) -4.0013685 7.2886603
tv$y         0.6529371 0.8936109
> 
```

#g

```r
#g (R-squared)
summary(tvlm)
tv_sum <- summary(lm(tv$x ~ tv$y))
r_sq <- tv_sum$r.squared
r_sq
```

```
> tv_sum <- summary(lm(tv$x ~ tv$y))
> r_sq <- tv_sum$r.squared
> r_sq
[1] 0.9101164
```

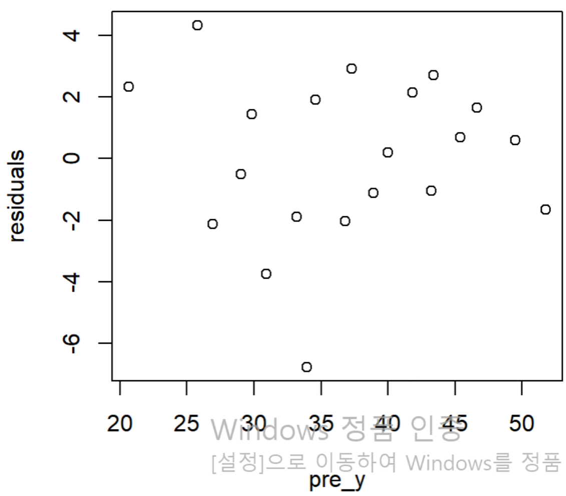
#h

```r
#h
residuals(tvlm)   #잔차
deviance(tvlm)   #잔차제곱합
```

```
> #h
> residuals(tvlm)   #잔차
         1          2          3          4          5          6          7          8 
-0.5175452  1.6518079  2.9084231 -2.0276125 -1.6518005  2.1461066 -3.7507301 -1.1154522 
         9         10         11         12         13         14         15         16 
-6.7664987  2.6995586 -1.8932247  0.5906941  1.4318535 -2.1297054 -1.0457866  2.3338139 
        17         18         19         20 
 4.3302056  1.9148821  0.2019642  0.6890462 
> deviance(tvlm)  #잔차제곱합
[1] 131.2785
```

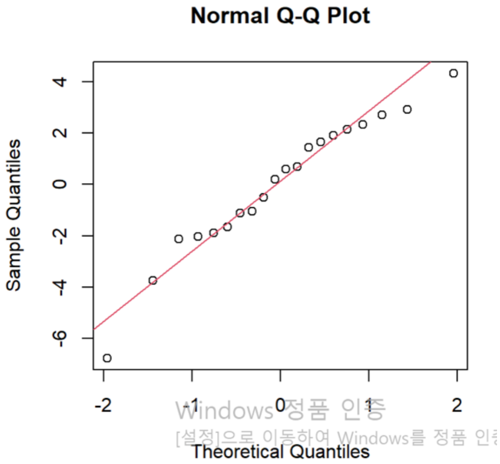
#i

```r
#i
# 잔차 계산
residuals <- residuals(tvlm)
# 이미 모델에 포함된 x를 사용하여 y를 추정
pre_y <- fitted(tvlm)
# (y의 추정치, 잔차) 그림
plot(pre_y, residuals)
```



#j

```r
#j
# 잔차 계산
residuals <- residuals(tvlm)
# Q-Q 그림 그리기
qqnorm(residuals)
qqline(residuals, col = 2)
```



#k

```r
#k
# 캠페인 비용 입력
campaign_cost <- 50
# 모델을 사용하여 선거 참여 비율 예측
predicted_participation <- predict(tvlm, newdata = data.frame(x = campaign_cost))
predicted_participation
```

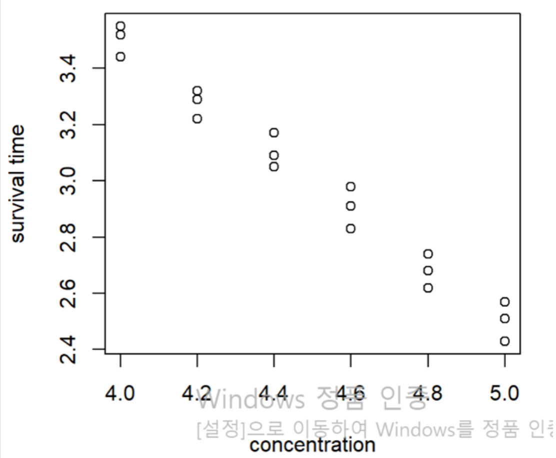
```
> predicted_participation
       1        2        3        4        5        6        7        8        9       10 
29.01755 46.64819 37.29158 36.82761 51.75180 41.85389 30.95073 38.91545 33.96650 43.40044 
      11       12       13       14       15       16       17       18       19       20 
33.19322 49.50931 29.86815 26.92971 43.24579 20.66619 25.76979 34.58512 39.99804 45.41095 
> 
```

3.9

```r
catfish <- data.frame(
  x = c(5.0, 5.0, 5.0, 4.8, 4.8, 4.8, 4.6, 4.6, 4.6,
        4.4, 4.4, 4.4, 4.2, 4.2, 4.2, 4.0, 4.0, 4.0 ),
  y = c(2.51, 2.57, 2.43, 2.62, 2.74, 2.68, 2.83, 2.91, 2.98,
        3.17, 3.05, 3.09, 3.32, 3.22, 3.29, 3.44, 3.52, 3.55)
)
```

#a

```r
#a
plot(catfish$x, catfish$y ,
      xlab = "concentration",
      ylab = "survival time" )
```



#b

```r
#b
cor(catfish$x, catfish$y)
```

```
> #b
> cor(catfish$x, catfish$y)
[1] -0.9882052
>
```

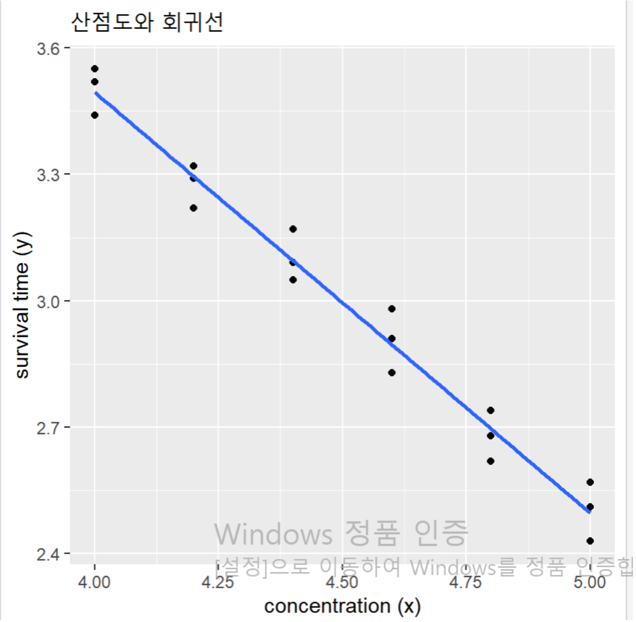
#c

```r
#c
catlm = lm(catfish$x ~ catfish$y)
summary(catlm)
```

```
> summary(catlm)

Call:
lm(formula = catfish$x ~ catfish$y)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.080401 -0.051693  0.002764  0.038088  0.084780 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   7.4309     0.1143   65.02  < 2e-16 ***
catfish$y    -0.9784     0.0379  -25.81 1.82e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05548 on 16 degrees of freedom
Multiple R-squared:  0.9765,	Adjusted R-squared:  0.9751 
F-statistic: 666.3 on 1 and 16 DF,  p-value: 1.815e-14

> 
```

#d

```r
#d
library(ggplot2)

ggplot(catfish, aes(x = x, y = y)) +
  geom_point() +  # 산점도
  geom_smooth(method = "lm", se = FALSE) +  # 회귀선 그리기 (lm: 선형회귀)
  labs(x = "concentration (x)", y = "survival time (y)") +  # 축 레이블 지정
  ggtitle("산점도와 회귀선")  # 그래프 제목
```



#e

```r
#e
cor.test(x=catfish$x, y=catfish$y)
```

```
> #e
> cor.test(x=catfish$x, y=catfish$y)

	Pearson's product-moment correlation

data:  catfish$x and catfish$y
t = -25.813, df = 16, p-value = 1.815e-14
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.9956970 -0.9678791
sample estimates:
       cor 
-0.9882052 
```

#f

```r
#f
confidence_interval <- confint(catlm)
confidence_interval
```

```
> #f
> confidence_interval <- confint(catlm)
> confidence_interval
                2.5 %    97.5 %
(Intercept)  7.188595  7.673187
catfish$y   -1.058767 -0.898059
> 
```

#g

```r
#g (R-squared)
cat_sum <- summary(lm(catfish$x ~ catfish$y))
r_sq <- cat_sum$r.squared
r_sq
```

```
> #g (R-squared)
> cat_sum <- summary(lm(catfish$x ~ catfish$y))
> r_sq <- cat_sum$r.squared
> r_sq
[1] 0.9765494
> 
```

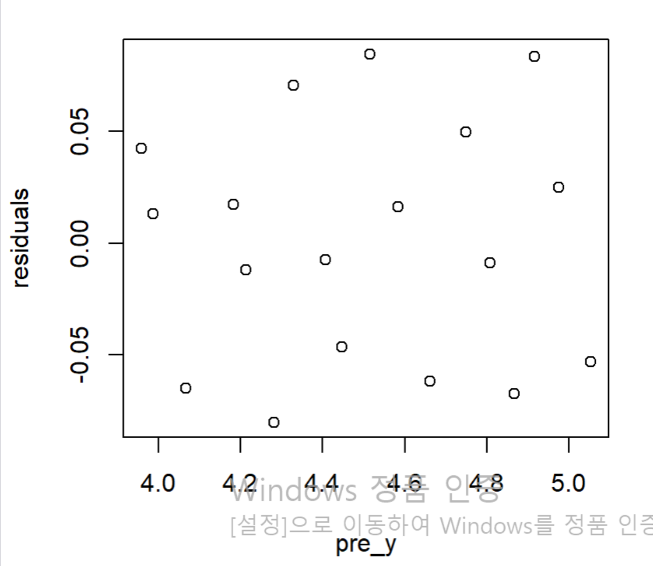
#h

```r
#h
residuals(catlm)   #잔차
deviance(catlm)   #잔차제곱합
```

```
> #h
> residuals(catlm)   #잔차
           1            2            3            4            5            6 
 0.024926090  0.083630875 -0.053346957 -0.067448470  0.049961100 -0.008743685 
           7            8            9           10           11           12 
-0.061981722  0.016291325  0.084780241  0.070678727 -0.046730843 -0.007594320 
          13           14           15           16           17           18 
 0.017440690 -0.080400618 -0.011911702 -0.065149739  0.013123308  0.042475700 
> deviance(catlm)   #잔차제곱합
[1] 0.04924617
```

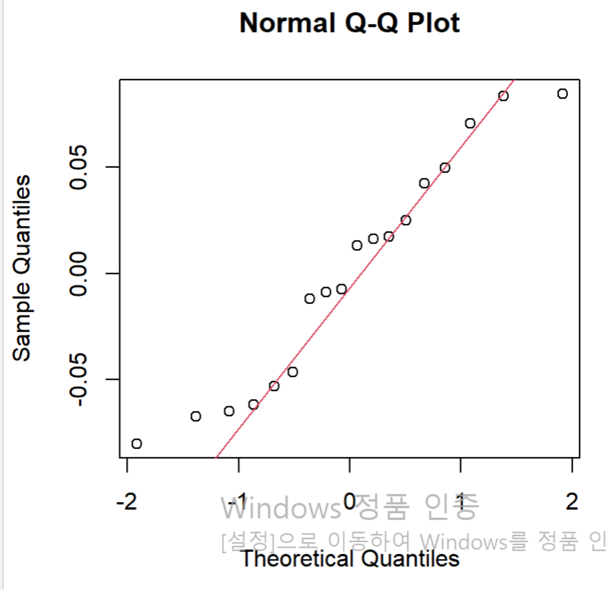
#i

```r
#i
# 잔차 계산
residuals <- residuals(catlm)
# 이미 모델에 포함된 x를 사용하여 y를 추정
pre_y <- fitted(catlm)
# (y의 추정치, 잔차) 그림
plot(pre_y, residuals)
```



#j

```r
#j
# 잔차 계산
residuals <- residuals(catlm)
# Q-Q 그림 그리기
qqnorm(residuals)
qqline(residuals, col = 2)
```



#k

```r
#k
# 선형 회귀 모델 피팅
model <- lm(y ~ x, data = catfish)
# 예측을 위한 새로운 데이터 프레임 생성
new_data <- data.frame(x = 5.5)
# 오염물질 양이 5.5인 경우 생존시간 예측
predicted_survival_time <- predict(model, new_data)
predicted_survival_time
```

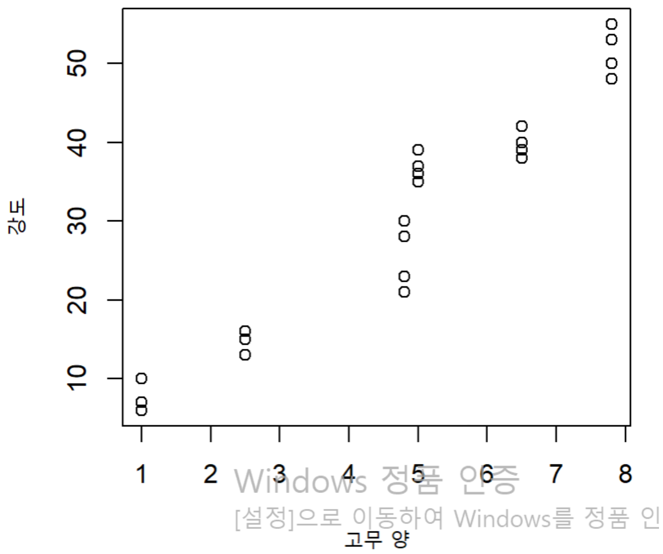
```
> predicted_survival_time
      1
1.99746
>
```

3.14

```r
plastic <- data.frame(
  x = c(1.0, 1.0, 1.0, 1.0, 2.5, 2.5, 2.5, 2.5, 4.8, 4.8, 4.8, 4.8,
        5.0, 5.0, 5.0, 5.0, 6.5, 6.5, 6.5, 6.5, 7.8, 7.8, 7.8, 7.8),
  y = c(6.0, 6.0, 7.0, 10.0, 13.0, 13.0, 15.0, 16.0, 21.0, 23.0, 28.0, 30.0,
        36.0, 39.0, 37.0, 35.0, 38.0, 39.0, 40.0, 42.0, 50.0, 53.0, 48.0, 55.0)
)

```

#a

```r
#a
plot(plastic$x, plastic$y ,
     xlab = "고무 양",
     ylab = "강도" )
```



#b

```r
#b
cor(plastic$x, plastic$y)

```

```
> #b
> cor(plastic$x, plastic$y)
[1] 0.9683688
> 
```

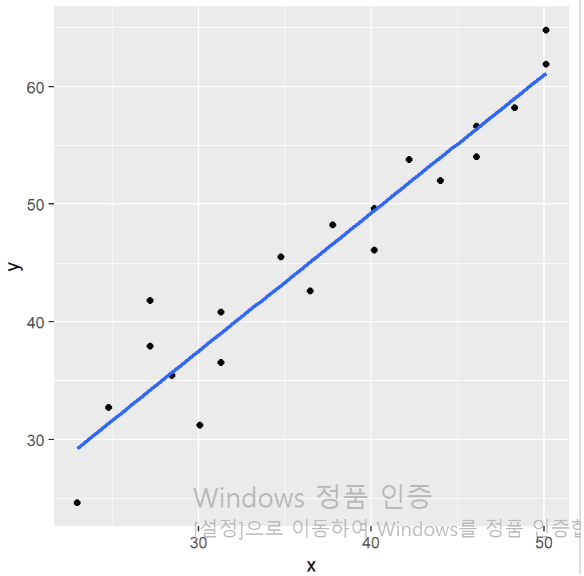
#c

```
#c
plalm = lm(plastic$x ~ plastic$y)
summary(plalm)
```

```
> summary(plalm)

Call:
lm(formula = plastic$x ~ plastic$y)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.01953 -0.41055 -0.00376  0.34419  1.37893 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.389520   0.261388    1.49     0.15    
plastic$y   0.144359   0.007931   18.20 9.44e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5964 on 22 degrees of freedom
Multiple R-squared:  0.9377,	Adjusted R-squared:  0.9349 
F-statistic: 331.3 on 1 and 22 DF,  p-value: 9.441e-15
```

#d

```
#d
library(ggplot2)

ggplot(tv, aes(x = x, y = y)) +
  geom_point() +  # 산점도
  geom_smooth(method = "lm", se = FALSE)
  labs(x = "고무 양 (x)", y = "강도 (y)")
  ggtitle("산점도와 회귀선")
```



#e

```
#e
cor.test(x=plastic$x, y=plastic$y)
```

```
> #e
> cor.test(x=plastic$x, y=plastic$y)

        Pearson's product-moment correlation

data:  plastic$x and plastic$y
t = 18.203, df = 22, p-value = 9.441e-15
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9271517 0.9864298
sample estimates:
      cor
0.9683688
```

#f

```
#f
confidence_interval <- confint(catlm)
confidence_interval
```

```
> #f
> confidence_interval <- confint(catlm)
> confidence_interval
                 2.5 %    97.5 %
(Intercept)  7.188595  7.673187
catfish$y   -1.058767 -0.898059
>
```

#g

```
#g
# 단순 회귀 모델 피팅
model <- lm(y ~ x, data = plastic)
# 적합결여 분산분석표 작성
anova_table <- anova(model)
anova_table
```

```
> anova_table
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)    
x          1 5303.2  5303.2  331.35 9.441e-15 ***
Residuals 22  352.1    16.0                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

과제 코드

3.5

```
tv <- data.frame(
  x = c(28.5, 48.3, 40.2, 34.8, 50.1, 44.0, 27.2, 37.8, 27.2, 46.1,
        31.3, 50.1, 31.3, 24.8, 42.2, 23.0, 30.1, 36.5, 40.2, 46.1 ),
  y = c(35.4, 58.2, 46.1, 45.5, 64.8, 52.0, 37.9, 48.2, 41.8, 54.0,
        40.8, 61.9, 36.5, 32.7, 53.8, 24.6, 31.2, 42.6, 49.6, 56.6)
)

#a
plot(tv$x, tv$y ,
     xlab = "Campaign Cost",
     ylab = "Voter Turnout" )

#b
cor(tv$x, tv$y)


#c
tvlm = lm(tv$x ~ tv$y)
summary(tvlm)

#d
```

```
# ggplot2 패키지 설치 (한 번만 실행)
install.packages("ggplot2")

library(ggplot2)

ggplot(tv, aes(x = x, y = y)) +
  geom_point() +  # 산점도
  geom_smooth(method = "lm", se = FALSE) +  # 회귀선 그리기 (lm: 선형회귀)
  labs(x = "캠페인 비용 (x)", y = "선거참여율 (y)") +  # 축 레이블 지정
  ggtitle("산점도와 회귀선")  # 그래프 제목


#e
cor.test(x=tv$x, y=tv$y)


#f
confidence_interval <- confint(tvlm)
confidence_interval


#g (R-squared)
summary(tvlm)
tv_sum <- summary(lm(tv$x ~ tv$y))
r_sq <- tv_sum$r.squared
r_sq



#h
residuals(tvlm)  #잔차
deviance(tvlm)  #잔차제곱합


#i
# 잔차 계산
residuals <- residuals(tvlm)
# 이미 모델에 포함된 x를 사용하여 y를 추정
pre_y <- fitted(tvlm)
# (y의 추정치, 잔차) 그림
plot(pre_y, residuals)


#j
# 잔차 계산
residuals <- residuals(tvlm)
# Q-Q 그림 그리기
qqnorm(residuals)
qqline(residuals, col = 2)


#k
```

```
# 캠페인 비용 입력
campaign_cost <- 50
# 모델을 사용하여 선거 참여 비율 예측
predicted_participation <- predict(tvlm, newdata = data.frame(x = campaign_cost))
predicted_participation
```

3.9

```
catfish <- data.frame(
  x = c(5.0, 5.0, 5.0, 4.8, 4.8, 4.8, 4.6, 4.6, 4.6,
        4.4, 4.4, 4.4, 4.2, 4.2, 4.2, 4.0, 4.0, 4.0 ),
  y = c(2.51, 2.57, 2.43, 2.62, 2.74, 2.68, 2.83, 2.91, 2.98,
        3.17, 3.05, 3.09, 3.32, 3.22, 3.29, 3.44, 3.52, 3.55)
)


#a
plot(catfish$x, catfish$y ,
     xlab = "concentration",
     ylab = "survival time" )

#b
cor(catfish$x, catfish$y)

#c
catlm = lm(catfish$x ~ catfish$y)
summary(catlm)


#d
library(ggplot2)

ggplot(catfish, aes(x = x, y = y)) +
  geom_point() +  # 산점도
  geom_smooth(method = "lm", se = FALSE) +  # 회귀선 그리기 (lm: 선형회귀)
  labs(x = "concentration (x)", y = "survival time (y)") +  # 축 레이블 지정
  ggtitle("산점도와 회귀선")  # 그래프 제목


#e
cor.test(x=catfish$x, y=catfish$y)
```

```
#f
confidence_interval <- confint(catlm)
confidence_interval



#g (R-squared)
cat_sum <- summary(lm(catfish$x ~ catfish$y))
r_sq <- cat_sum$r.squared
r_sq


#h
residuals(catlm)   #잔차
deviance(catlm)   #잔차제곱합


#i
# 잔차 계산
residuals <- residuals(catlm)
# 이미 모델에 포함된 x 를 사용하여 y 를 추정
pre_y <- fitted(catlm)
# (y 의 추정치, 잔차) 그림
plot(pre_y, residuals)





#j
# 잔차 계산
residuals <- residuals(catlm)
# Q-Q 그림 그리기
qqnorm(residuals)
qqline(residuals, col = 2)


#k
# 선형 회귀 모델 피팅
model <- lm(y ~ x, data = catfish)
# 예측을 위한 새로운 데이터 프레임 생성
new_data <- data.frame(x = 5.5)
# 오염물질 양이 5.5 인 경우 생존시간 예측
predicted_survival_time <- predict(model, new_data)
predicted_survival_time
```

3.14

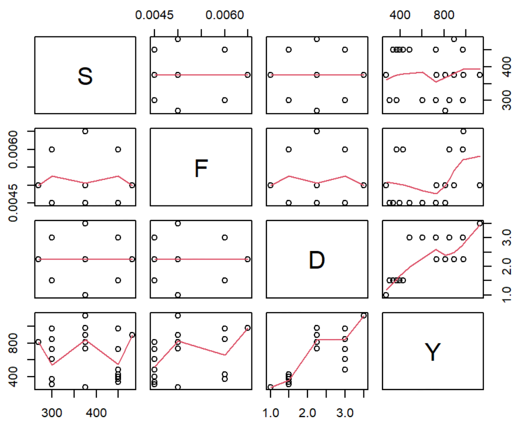
```
plastic <- data.frame(
  x = c(1.0, 1.0, 1.0, 1.0, 2.5, 2.5, 2.5, 2.5, 4.8, 4.8, 4.8, 4.8,
        5.0, 5.0, 5.0, 5.0, 6.5, 6.5, 6.5, 6.5, 7.8, 7.8, 7.8, 7.8),
  y = c(6.0, 6.0, 7.0, 10.0, 13.0, 13.0, 15.0, 16.0, 21.0, 23.0, 28.0, 30.0,
        36.0, 39.0, 37.0, 35.0, 38.0, 39.0, 40.0, 42.0, 50.0, 53.0, 48.0, 55.0)
)

#a
plot(plastic$x, plastic$y ,
     xlab = "고무 양",
     ylab = "강도" )

#b
cor(plastic$x, plastic$y)


#c
plalm = lm(plastic$x ~ plastic$y)
summary(plalm)


#d
library(ggplot2)

ggplot(tv, aes(x = x, y = y)) +
  geom_point() +  # 산점도
  geom_smooth(method = "lm", se = FALSE)
  labs(x = "고무 양 (x)", y = "강도 (y)")
  ggtitle("산점도와 회귀선")


#e
cor.test(x=plastic$x, y=plastic$y)


#f
confidence_interval <- confint(catlm)
confidence_interval

#g
# 단순 회귀 모델 피팅
model <- lm(y ~ x, data = plastic)
# 적합결여 분산분석표 작성
anova_table <- anova(model)
anova_table
```

3 차 과제

4.4

```
drill <- data.frame(
  S = c( 450, 450, 300, 300, 481, 375, 450, 450, 375, 300, 300, 375, 450, 450, 300, 300, 375,
375, 375, 375, 375, 375, 375, 375, 375, 269, 450, 450, 300, 300, 375),
  F = c(0.006, 0.006, 0.0045, 0.0045, 0.005, 0.0045, 0.006, 0.006, 0.0065, 0.0045, 0.0045,
0.005, 0.0045, 0.0045, 0.006, 0.006, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005,
0.005, 0.005, 0.0045, 0.0045, 0.006, 0.006, 0.005),
  D = c(1.5, 1.5, 1.5, 1.5, 2.25, 2.25, 3, 3, 2.25, 3, 3, 1, 1.5, 1.5, 1.5, 1.5, 2.25, 2.25, 2.25,
2.25, 2.25, 2.25, 2.25, 2.25, 2.25, 2.25, 3, 3, 3, 3, 3.5),
  Y = c(430, 368, 306, 306, 894, 813, 969, 969, 976, 727, 606, 276, 338, 399, 368, 368,
894,732, 813, 894, 732, 813, 813, 894, 813, 813, 727, 485, 969, 847, 1126 )
)
```

(a) pairs(drill[,1:4], panel=panel.smooth)



(b)

```
# 다중선형회귀 모형 적합
b_model <- lm(Y ~ S + F + D, data = drill)
```

(c)

적합된 모형 요약 출력

summary(b_model)

#유의수준 5%

summary(b_model)$coefficients

```
> # 적합된 모형 요약 출력
> summary(b_model)

Call:
lm(formula = Y ~ S + F + D, data = drill)

Residuals:
    Min      1Q  Median      3Q     Max 
-394.33 -121.35  -30.17  133.05  214.05 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -467.7713   328.6909  -1.423    0.166    
S               0.2017     0.4914   0.410    0.685    
F           79899.3103 48191.6915   1.658    0.109    
D             298.9278    47.3265   6.316 9.23e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 164.8 on 27 degrees of freedom
Multiple R-squared:  0.6133,	Adjusted R-squared:  0.5703 
F-statistic: 14.27 on 3 and 27 DF,  p-value: 9.176e-06

> #유의수준 5%
> summary(b_model)$coefficients
                Estimate   Std. Error    t value     Pr(>|t|)
(Intercept)  -467.7712821 3.286909e+02 -1.4231344 1.661501e-01
S               0.2017035 4.913864e-01  0.4104785 6.846943e-01
F           79899.3103448 4.819169e+04  1.6579478 1.089032e-01
D             298.9278351 4.732651e+01  6.3162877 9.234009e-07
```

1. (Intercept):
- 검정통계량 (t-value): -1.423
- p-value: 0.166

2. S:
- 검정통계량 (t-value): 0.410
- p-value: 0.686

3. F:
- 검정통계량 (t-value): 1.658
- p-value: 0.109

4. D:
- 검정통계량 (t-value): 6.316
- p-value: 9.23e-07

(d)

결정계수 계산

rsquared <- summary(b_model)$r.squared

결과 출력

cat("Multiple R-squared:", rsquared)

```
> # 결정계수 계산
> rsquared <- summary(b_model)$r.squared
> # 결과 출력
> cat("Multiple R-squared:", rsquared)
Multiple R-squared: 0.6132513
```

(e)

```
# 다중회귀 모형 적합
e_model<- lm(Y ~ S + F + D + S:F + S:D + F:D, data = drill)
```

(f)

```
# 적합된 모형 요약 출력
summary(e_model)
#유의수준 5%
summary(e_model)$coefficients
```

```
> summary(e_model)

Call:
lm(formula = Y ~ S + F + D + S:F + S:D + F:D, data = drill)

Residuals:
    Min      1Q  Median      3Q     Max 
-319.75  -92.91  -64.78  133.05  214.05 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.199e+02  1.816e+03   0.507    0.617
S           -2.489e-01  4.207e+00  -0.059    0.953
F           -2.211e+05  3.253e+05  -0.680    0.503
D           -1.652e+02  4.742e+02  -0.348    0.731
S:F          1.761e+02  7.359e+02   0.239    0.813
S:D         -2.067e-01  7.441e-01  -0.278    0.784
F:D          1.044e+05  7.336e+04   1.424    0.167

Residual standard error: 167.4 on 24 degrees of freedom
Multiple R-squared:  0.6452,	Adjusted R-squared:  0.5565 
F-statistic: 7.274 on 6 and 24 DF,  p-value: 0.000164

> #유의수준 5%
> summary(e_model)$coefficients
                 Estimate   Std. Error     t value  Pr(>|t|)
(Intercept)  9.198572e+02 1.816010e+03  0.50652642 0.6171091
S           -2.488939e-01 4.206862e+00 -0.05916378 0.9533115
F           -2.211376e+05 3.252845e+05 -0.67982836 0.5031184
D           -1.651964e+02 4.742345e+02 -0.34834323 0.7306195
S:F          1.760747e+02 7.359479e+02  0.23924892 0.8129439
S:D         -2.066667e-01 7.440721e-01 -0.27775086 0.7835823
F:D          1.044484e+05 7.336416e+04  1.42369786 0.1674057
```

(Intercept):

- 검정통계량 (t-value): 0.507
- p-value: 0.617

S:

- 검정통계량 (t-value): -0.059
- p-value: 0.953

F:

- 검정통계량 (t-value): -0.680
- p-value: 0.503

D:

- 검정통계량 (t-value): -0.348
- p-value: 0.731

S:F:

- 검정통계량 (t-value): 0.239
- p-value: 0.813

S:D:

- 검정통계량 (t-value): -0.278
- p-value: 0.784

F:D:

- 검정통계량 (t-value): 1.424
- p-value: 0.167

(g)

```
# 결정계수 계산
e_rsquared <- summary(e_model)$r.squared
# 결과 출력
cat("Multiple R-squared:", e_rsquared)
```

```
> # 결정계수 계산
> e_rsquared <- summary(e_model)$r.squared
> # 결과 출력
> cat("Multiple R-squared:", e_rsquared)
Multiple R-squared: 0.6452023
```

(h)

```
#h
# 다중회귀 모형 (상호작용과 이차항 포함)
h_model <- lm(Y ~ S + F + D + S:F + S:D + F:D + I(S^2) + I(F^2) +
I(D^2), data
= drill)
```

(i)

적합된 모형 요약 출력

summary(h_model)

#유의수준 5%

summary(h_model)$coefficients

```
> # 적합된 모형 요약 출력
> summary(h_model)

Call:
lm(formula = Y ~ S + F + D + S:F + S:D + F:D + I(S^2) + I(F^2) +
    I(D^2), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-204.25  -37.67  -28.09   48.31  198.71

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.510e+03  2.331e+03  -2.792  0.01091 *
S            1.061e+01  5.581e+00   1.901  0.07108 .
F            1.532e+06  7.935e+05   1.931  0.06714 .
D            6.040e+02  3.975e+02   1.520  0.14349
I(S^2)      -1.448e-02  6.377e-03  -2.271  0.03382 *
I(F^2)      -1.629e+08  7.122e+07  -2.287  0.03270 *
I(D^2)      -1.709e+02  5.110e+01  -3.346  0.00307 **
S:F          1.761e+02  5.031e+02   0.350  0.72985
S:D         -2.067e-01  5.087e-01  -0.406  0.68865
F:D          1.044e+05  5.016e+04   2.083  0.04971 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 114.5 on 21 degrees of freedom
Multiple R-squared:  0.8549,	Adjusted R-squared:  0.7927
F-statistic: 13.75 on 9 and 21 DF,  p-value: 6.038e-07

> #유의수준 5%
> summary(h_model)$coefficients
                 Estimate   Std. Error    t value    Pr(>|t|)
(Intercept) -6.510332e+03 2.331424e+03 -2.7924274 0.010913309
S            1.061113e+01 5.581147e+00  1.9012451 0.071080872
F            1.531900e+06 7.934699e+05  1.9306340 0.067141896
D            6.040430e+02 3.974654e+02  1.5197374 0.143490801
I(S^2)      -1.448003e-02 6.377434e-03 -2.2705104 0.033823001
I(F^2)      -1.628656e+08 7.122102e+07 -2.2867637 0.032695154
I(D^2)      -1.709421e+02 5.109550e+01 -3.3455410 0.003065451
S:F          1.760747e+02 5.031276e+02  0.3499604 0.729853900
S:D         -2.066667e-01 5.086817e-01 -0.4062790 0.688648073
F:D          1.044484e+05 5.015509e+04  2.0825084 0.049709849
```

(Intercept):
- 검정통계량 (t-value): -2.792
- p-value: 0.01091 (유의수준 5%에서 유의함)

S:
- 검정통계량 (t-value): 1.901
- p-value: 0.07108

F:
- 검정통계량 (t-value): 1.931
- p-value: 0.06714

D:
- 검정통계량 (t-value): 1.520
- p-value: 0.14349

I(S^2):
- 검정통계량 (t-value): -2.271
- p-value: 0.03382 (유의수준 5%에서 유의함)

I(F^2):
- 검정통계량 (t-value): -2.287
- p-value: 0.03270 (유의수준 5%에서 유의함)

I(D^2):
- 검정통계량 (t-value): -3.346
- p-value: 0.00307 (유의수준 5%에서 유의함)

S:F:
- 검정통계량 (t-value): 0.350
- p-value: 0.72985

S:D:
- 검정통계량 (t-value): -0.406
- p-value: 0.68865

F:D:
- 검정통계량 (t-value): 2.083
- p-value: 0.04971 (유의수준 5%에서 유의함)

(j)

```
# 결정계수 계산
h_rsquared <- summary(h_model)$r.squared
# 결과 출력
cat("Multiple R-squared:", h_rsquared)
```

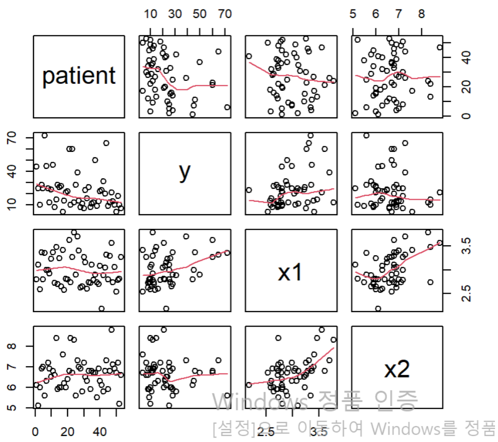
```
> # 결정계수 계산
> h_rsquared <- summary(h_model)$r.squared
> # 결과 출력
> cat("Multiple R-squared:", h_rsquared)
Multiple R-squared: 0.8549055
```

4.5

```
psychology <- data.frame(
  patient = 1:53,
  y = c(44, 25, 10, 28, 25, 72, 45, 25, 12, 24, 46, 8, 15, 28, 26, 27, 4, 14, 21, 22, 60, 10, 60, 12, 28, 39, 14, 8, 11, 7, 23, 16, 26, 8, 11, 12, 50, 9, 13, 22, 23, 31, 20, 65, 9, 12, 21, 13, 10, 4, 18, 10, 7),
  x1 = c(2.8, 3.1, 2.59, 3.36, 2.8, 3.35, 2.99, 2.99, 2.92, 3.23, 3.37, 2.72, 3.47, 2.7, 3.24, 2.65, 3.41, 2.58, 2.81, 2.8, 3.62, 2.74, 3.27, 3.78, 2.9, 3.7, 3.4, 2.63, 2.65, 3.26, 3.15, 2.6, 2.74, 2.72, 3.11, 2.79, 2.9, 2.74, 2.7, 3.08, 2.18, 2.88, 3.04, 3.32, 2.8, 3.29, 3.56, 2.74, 3.06, 2.54, 2.78, 2.81, 3.26),
  x2 = c(6.1, 5.1, 6, 6.9, 7, 5.6, 6.3, 7.2, 6.9, 6.5, 6.8, 6.6, 8.4, 5.9, 6, 6, 7.6, 6.2, 6, 6.4, 6.8, 8.4, 6.7, 8.3, 5.6, 7.3, 7, 6.9, 5.8, 7.2, 6.5, 6.3, 6.8, 5.9, 6.8, 6.7, 6.7, 5.5, 6.9, 6.3, 6.1, 5.8, 6.8, 7.3, 5.9, 6.8, 8.8, 7.1, 6.9, 6.7, 7.2, 5.2, 6.6)
)
```

(a)

```
pairs(psychology[,1:4], panel=panel.smooth)
```



(b)

```
# 다중선형회귀 모형 적합
bb_model <- lm(y ~ x1 + x2, data = psychology)
```

(c)

적합된 모형 요약 출력

summary(bb_model)

#유의수준 5%

summary(bb_model)$coefficients

```
> summary(bb_model)

Call:
lm(formula = y ~ x1 + x2, data = psychology)

Residuals:
    Min      1Q  Median      3Q     Max 
-22.137  -9.490  -2.019   6.438  39.407 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  -0.6354    20.9683  -0.030  0.97595   
x1           23.4514     6.8385   3.429  0.00122 **
x2           -7.0726     3.0109  -2.349  0.02282 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.86 on 50 degrees of freedom
Multiple R-squared:  0.1997,	Adjusted R-squared:  0.1677 
F-statistic: 6.238 on 2 and 50 DF,  p-value: 0.003815

> #유의수준 5%
> summary(bb_model)$coefficients
              Estimate Std. Error     t value    Pr(>|t|)
(Intercept) -0.6353502  20.968334 -0.03030046 0.975948055
x1          23.4514352   6.838510  3.42931961 0.001220793
x2          -7.0726092   3.010924 -2.34898271 0.022817535
```

(Intercept):

- 검정통계량 (t-value): −0.0303
- p-value: 0.97595

x1:

- 검정통계량 (t-value): 3.4293
- p-value: 0.00122 (유의수준 5%에서 유의함)

x2:

- 검정통계량 (t-value): −2.349
- p-value: 0.02282 (유의수준 5%에서 유의함)

(d)

결정계수 계산

rsquared <- summary(bb_model)$r.squared

결과 출력

cat("Multiple R-squared:", rsquared)

```
> # 결정계수 계산
> rsquared <- summary(bb_model)$r.squared
> # 결과 출력
> cat("Multiple R-squared:", rsquared)
Multiple R-squared: 0.1996837
```

(e)

#e

모형 1: 설명변수 1 개

model_1 <- lm(y ~ x1, data = psychology)

모형 2: 설명변수 2 개

model_2 <- lm(y ~ x1 + x2, data = psychology)

결정계수(R-squared) 출력

rsquared_1 <- summary(model_1)$r.squared

rsquared_2 <- summary(model_2)$r.squared

비교 결과 출력

cat("R-squared for Model 1:", rsquared_1, "₩n")

cat("R-squared for Model 2:", rsquared_2, "₩n")

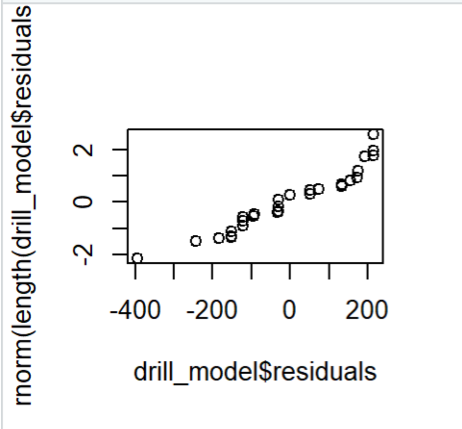
```
> cat("R-squared for Model 1:", rsquared_1, "\n")
R-squared for Model 1: 0.1113652
> cat("R-squared for Model 2:", rsquared_2, "\n")
R-squared for Model 2: 0.1996837
```

5.4

```
drill <- data.frame(
   S = c( 450, 450, 300, 300, 481, 375, 450, 450, 375, 300, 300, 375, 450, 450, 300, 300, 375,
375, 375, 375, 375, 375, 375, 375, 375, 269, 450, 450, 300, 300, 375),
   F = c(0.006, 0.006, 0.0045, 0.0045, 0.005, 0.0045, 0.006, 0.006, 0.0065, 0.0045, 0.0045,
0.005, 0.0045, 0.0045, 0.006, 0.006, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005,
0.005, 0.005, 0.0045, 0.0045, 0.006, 0.006, 0.005),
   D = c(1.5, 1.5, 1.5, 1.5, 2.25, 2.25, 3, 3, 2.25, 3, 3, 1, 1.5, 1.5, 1.5, 1.5, 2.25, 2.25, 2.25,
2.25, 2.25, 2.25, 2.25, 2.25, 2.25, 2.25, 3, 3, 3, 3, 3.5),
   Y = c(430, 368, 306, 306, 894, 813, 969, 969, 976, 727, 606, 276, 338, 399, 368, 368,
894,732, 813, 894, 732, 813, 813, 894, 813, 813, 727, 485, 969, 847, 1126 )
)
```

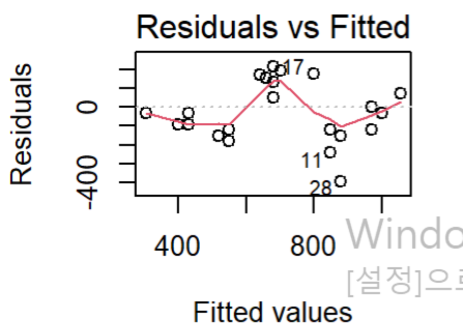
(a)

```
# 잔차의 정규성 확인
qqplot(x = drill_model$residuals, y = rnorm(length(drill_model$residuals)))
```



(b)

```
# 잔차의 등분산성 확인
plot(drill_model, which = 1)
```



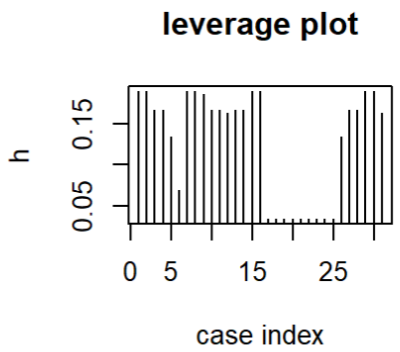
(c)

h <- hat(model.matrix(drill_model)) ; h

지렛대 그림

plot(h, type = "h", xlab = "case index", main = "leverage plot")

```
> plot(h, type = "h", xlab = "case index", main = "leverage plot")
> h <- hat(model.matrix(drill_model)) ; h
 [1] 0.18881799 0.18881799 0.16605937 0.16605937 0.13438318 0.06965
517 0.18881799 0.18881799
 [9] 0.18551724 0.16605937 0.16605937 0.16334874 0.16605937 0.16605
937 0.18881799 0.18881799
[17] 0.03448276 0.03448276 0.03448276 0.03448276 0.03448276 0.03448
276 0.03448276 0.03448276
[25] 0.03448276 0.13438318 0.16605937 0.16605937 0.18881799 0.18881
799 0.16334874
```



(d)

이상점 검정

outlier_test <- outlierTest(drill_model)

outlier_test

```
> # 이상점 검정
> outlier_test <- outlierTest(drill_model)
> outlier_test
No Studentized residuals with Bonferroni p < 0.05
Largest |rstudent|:
   rstudent unadjusted p-value Bonferroni p
28 -2.97756          0.0062145      0.19265
```

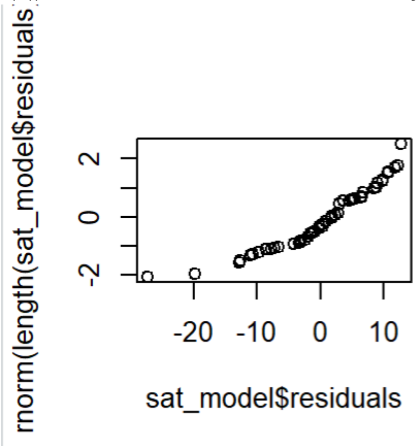
(e)

influencePlot(drill_model, main="Influence Plot")

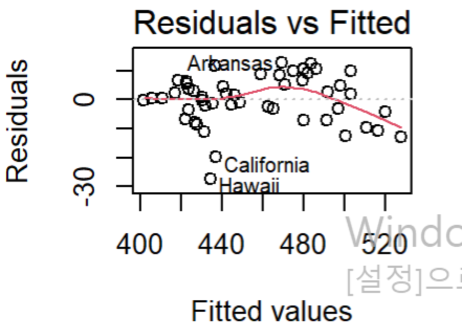
```
> #e
> influencePlot(drill_model, main="Influence Plot")
        StudRes       Hat        CookD
1  -0.808533730 0.1888180 3.853605e-02
11 -1.667587790 0.1660594 1.298691e-01
28 -2.977560083 0.1660594 3.417847e-01
29  0.000534569 0.1888180 1.726883e-08
```

5.5
install.packages("faraway")
library(faraway)
data(sat)
sat

(a)
# 잔차의 정규성 확인
qqplot(x = sat_model$residuals, y = rnorm(length(sat_model$residuals)))

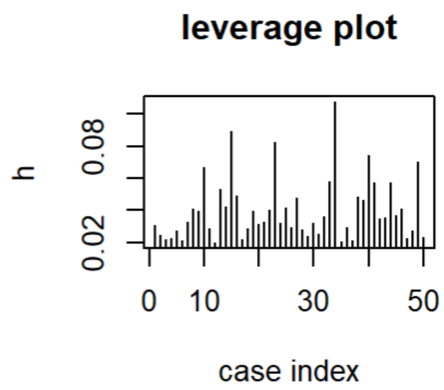


(b)
# 잔차의 등분산성 확인
plot(sat_model, which = 1)



(c)
#c
h <- hat(model.matrix(sat_model)) ; h
# 지렛대 그림
plot(h, type = "h", xlab = "case index", main = "leverage plot")

```
> h <- hat(model.matrix(sat_model)) ; h
 [1] 0.03077979 0.02493972 0.02206211 0.02255299 0.02713959 0.02107
328 0.03275139 0.04099637
 [9] 0.03997925 0.06664135 0.02905463 0.02006222 0.05312294 0.04203
873 0.08954911 0.04935649
[17] 0.02220654 0.02867991 0.03997925 0.03119693 0.03275139 0.04042
367 0.08225459 0.03230596
[25] 0.04145190 0.02935462 0.04815151 0.02839103 0.02399129 0.03196
154 0.02568513 0.03616331
[33] 0.05788731 0.10743899 0.02048846 0.02935462 0.02120761 0.04882
319 0.04646070 0.07463079
[41] 0.05711665 0.03478460 0.03527246 0.05711665 0.03707942 0.04099
637 0.02275803 0.02775269
[49] 0.07046133 0.02332163
```



(d)
# 이상점 검정
outlier_test <- outlierTest(drill_model)
outlier_test

```
> # 이상점 검정
> outlier_test <- outlierTest(drill_model)
> outlier_test
No Studentized residuals with Bonferroni p < 0.05
Largest |rstudent|:
   rstudent unadjusted p-value Bonferroni p
28 -2.97756          0.0062145      0.19265
```

(e)
influencePlot(drill_model, main="Influence Plot")

```
> influencePlot(drill_model, main="Influence Plot")
        StudRes       Hat        CookD
1  -0.808533730 0.1888180 3.853605e-02
11 -1.667587790 0.1660594 1.298691e-01
28 -2.977560083 0.1660594 3.417847e-01
29  0.000534569 0.1888180 1.726883e-08
```

4 차 과제

6.2

```
gala_data <- data.frame(
  Island = c("Baltra", "Bartolome", "Caldwell", "Champion", "Coamano", "Daphne.Major",
             "Daphne.Minor", "Darwin", "Eden", "Enderby", "Espanola", "Fernandina",
             "Gardner1", "Gardner2", "Genovesa", "Isabela", "Marchena", "Onslow",
             "Pinta", "Pinzon", "Las.Plazas", "Rabida", "SanCristobal", "SanSalvador",
             "SantaCruz", "SantaFe", "SantaMaria", "Seymour", "Tortuga", "Wolf"),
  Species = c(58, 31, 3, 25, 2, 18, 24, 10, 8, 2, 97, 93, 58, 5, 40, 347, 51, 2, 104,
              108, 12, 70, 280, 237, 444, 62, 285, 44, 16, 21),
  Endemics = c(23, 21, 3, 9, 1, 11, 0, 7, 4, 2, 26, 35, 17, 4, 19, 89, 23, 2, 37, 33,
               9, 30, 65, 81, 95, 28, 73, 16, 8, 12),
  Area = c(25.09, 1.24, 0.21, 0.1, 0.05, 0.34, 0.08, 2.33, 0.03, 0.18, 58.27, 634.49,
           0.57, 0.78, 17.35, 4669.32, 129.49, 0.01, 59.56, 17.95, 0.23, 4.89, 551.62,
           572.33, 903.82, 24.08, 170.92, 1.84, 1.24, 2.85),
  Elevation = c(346, 109, 114, 46, 77, 119, 93, 168, 71, 112, 198, 1494, 49, 227, 76,
                1707, 343, 25, 777, 458, 94, 367, 716, 906, 864, 259, 640, 147, 186, 253),
  Nearest = c(0.6, 0.6, 2.8, 1.9, 1.9, 8, 6, 34.1, 0.4, 2.6, 1.1, 4.3, 1.1, 4.6, 47.4,
              0.7, 29.1, 3.3, 29.1, 10.7, 0.5, 4.4, 45.2, 0.2, 0.6, 16.5, 2.6, 0.6, 6.8,
              34.1),
  Scruz = c(0.6, 26.3, 58.7, 47.4, 1.9, 8, 12, 290.2, 0.4, 50.2, 88.3, 95.3, 93.1, 62.2,
            92.2, 28.1, 85.9, 45.9, 119.6, 10.7, 0.6, 24.4, 66.6, 19.8, 0, 16.5, 49.2, 9.6,
            50.9, 254.7),
  Adjacent = c(1.84, 572.33, 0.78, 0.18, 903.82, 1.84, 0.34, 2.85, 17.95, 0.1, 0.57, 4669.32,
               58.27, 0.21, 129.49, 634.49, 59.56, 0.1, 129.49, 17.95, 25.09, 572.33, 0.57,
               4.89, 0.52, 0.52, 0.1, 25.09, 17.95, 2.33)
)
```

```
#a
# Endemics에 대한 Species에 대한 비율 열 추가
gala_data$Species_Endemics_Ratio <- gala_data$Endemics / gala_data$Species

# 다중회귀모델 생성
model_1 <- lm(Species_Endemics_Ratio ~ Area + Elevation + Nearest + Scruz + Adjacent, data = gala_data)
summary(model_1)
```

```
> # Endemics에 대한 Species에 대한 비율 열 추가
> gala_data$Species_Endemics_Ratio <- gala_data$Endemics /
gala_data$Species
> # 다중회귀모델 생성
> model_1 <- lm(Species_Endemics_Ratio ~ Area + Elevation + Nearest +
Scruz + Adjacent, data = gala_data)
> summary(model_1)

Call:
lm(formula = Species_Endemics_Ratio ~ Area + Elevation + Nearest +
    Scruz + Adjacent, data = gala_data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.52958 -0.06388 -0.01503  0.07617  0.43024

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.746e-01  7.294e-02   7.877 4.15e-08 ***
```

```
Area          6.604e-05  8.543e-05   0.773   0.4471
Elevation    -4.093e-04  2.045e-04  -2.001   0.0568 .
Nearest      -3.115e-03  4.014e-03  -0.776   0.4453
Scruz         9.783e-04  8.202e-04   1.193   0.2447
Adjacent      5.944e-05  6.746e-05   0.881   0.3869
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2322 on 24 degrees of freedom
Multiple R-squared:  0.2638,	Adjusted R-squared:  0.1104
F-statistic:  1.72 on 5 and 24 DF,  p-value: 0.1684
```

#b

```
# Elevation 변수에 결측값이 존재하는 경우 결측값이 있는 행 제거
if (sum(is.na(data$Elevation)) > 0) {
  data_no_missing <- na.omit(data)

  # 다중회귀모델 생성
  model_no_missing <- lm(Endemics ~ Species + Area + Elevation + Nearest + Scruz + Adjacent, data =
data_no_missing)

  # 모델 요약
  summary(model_no_missing)
} else {
  print("Elevation 변수에 결측값이 없습니다.")
}
```

```
[1] "Elevation 변수에 결측값이 없습니다."
```

#c

```
cor(gala_data[2:7])
```

```
> cor(gala_data[2:7])
              Species      Endemics       Area   Elevation       Near
est        Scruz
Species    1.00000000  0.970876516  0.6178431  0.73848666 -0.014094
067 -0.17114244
Endemics   0.97087652  1.000000000  0.6169791  0.79290437  0.005994
286 -0.15426432
Area       0.61784307  0.616979087  1.0000000  0.75373492 -0.111103
196 -0.10078493
Elevation  0.73848666  0.792904369  0.7537349  1.00000000 -0.011076
984 -0.01543829
Nearest   -0.01409407  0.005994286 -0.1111032 -0.01107698  1.000000
000  0.61541036
Scruz     -0.17114244 -0.154264319 -0.1007849 -0.01543829  0.615410
357  1.00000000
```

vif(gala_model)

```
> # 다중공선성이 높은 변수 : Elevation
> vif(gala_model)
  Species      Area Elevation   Nearest     Scruz  Adjacent 
 4.276096  3.071133  9.912893  1.765814  1.762073  3.192579 
```

: 결과에 따라 elevation 의 다중공선성이 가장 높음 -> 변수선택: Elevation

# 다중공선성이 높은 변수 제거
gala_model_e <- lm(Endemics ~ Species + Area + Nearest + Scruz + Adjacent, data = gala_data)
vif(gala_model_e)

```
> # 다중공선성이 높은 변수 제거
> gala_model_e <- lm(Endemics ~ Species + Area + Nearest + Scruz + 
Adjacent, data = gala_data)
> vif(gala_model_e)
 Species     Area  Nearest    Scruz Adjacent 
1.724319 1.722934 1.727807 1.740346 1.078217 
```

: 제거 후 값이 변함

#d

# 단계별 변수선택법 수행
model <- lm(Endemics ~ Species + Area + Elevation + Nearest + Scruz + Adjacent, data = gala_data)
selected_model <- step(model)

```
> selected_model <- step(model)
Start:  AIC=111.86
Endemics ~ Species + Area + Elevation + Nearest + Scruz + Adjacent

            Df Sum of Sq    RSS    AIC
- Nearest    1      1.86  784.9 109.93
- Scruz      1     14.43  797.5 110.41
<none>                    783.1 111.86
- Adjacent   1    114.35  897.4 113.95
- Area       1    124.61  907.7 114.29
- Elevation  1    367.44 1150.5 121.40
- Species    1   2833.74 3616.8 155.76

Step:  AIC=109.93
Endemics ~ Species + Area + Elevation + Scruz + Adjacent

            Df Sum of Sq    RSS    AIC
- Scruz      1     14.02  798.9 108.46
<none>                    784.9 109.93
- Adjacent   1    126.41  911.3 112.41
- Area       1    136.09  921.0 112.73
- Elevation  1    383.40 1168.3 119.86
- Species    1   2834.01 3618.9 153.78

Step:  AIC=108.46
Endemics ~ Species + Area + Elevation + Adjacent

            Df Sum of Sq    RSS    AIC
<none>                    798.9 108.46
- Adjacent   1     115.6  914.6 110.52
- Area       1     125.2  924.1 110.83
- Elevation  1     371.1 1170.0 117.91
- Species    1    3197.2 3996.2 154.76
```

# 선택된 모델의 요약
summary(selected_model)

```
> # 선택된 모델의 요약
> summary(selected_model)

Call:
lm(formula = Endemics ~ Species + Area + Elevation + Adjacent,
    data = gala_data)

Residuals:
    Min      1Q  Median      3Q     Max
-10.042  -2.792  -0.535   1.946  13.805

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.304114   1.570041   2.104  0.04556 *
Species      0.181838   0.018179  10.002 3.19e-10 ***
Area        -0.004069   0.002056  -1.979  0.05889 .
Elevation    0.025545   0.007496   3.408  0.00222 **
Adjacent    -0.003970   0.002087  -1.902  0.06872 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.653 on 25 degrees of freedom
Multiple R-squared:  0.9631,	Adjusted R-squared:  0.9572
F-statistic: 163.2 on 4 and 25 DF,  p-value: < 2.2e-16
```

#e
Elevation 의 결측값을 추정하는 방법

1. 평균이나 중앙값으로 대체:
   - elevation 변수의 평균 또는 중앙값을 계산하고, 결측값을 해당 값으로 대체합니다.
2. 회귀분석을 활용한 예측
   -다른 변수들을 이용하여 Elevation 을 예측하는 회귀분석 모델을 만들고, 모델을 사용하여 결측값을 예측합니다.
3. 다른 변수들 간의 관계 고려