Linear regression.rmd

선형 회귀를 위한 프로그래밍 연습

Goal1. 회귀식에서 각 변수의 적절한 계수를 찾는다.
Goal2. 해당 계수가 0인지 여부를 테스트한다.

회귀식 세팅

X1, X2, X3의 변수를 가지는 회귀식 Y를 데이터프레임에 세팅

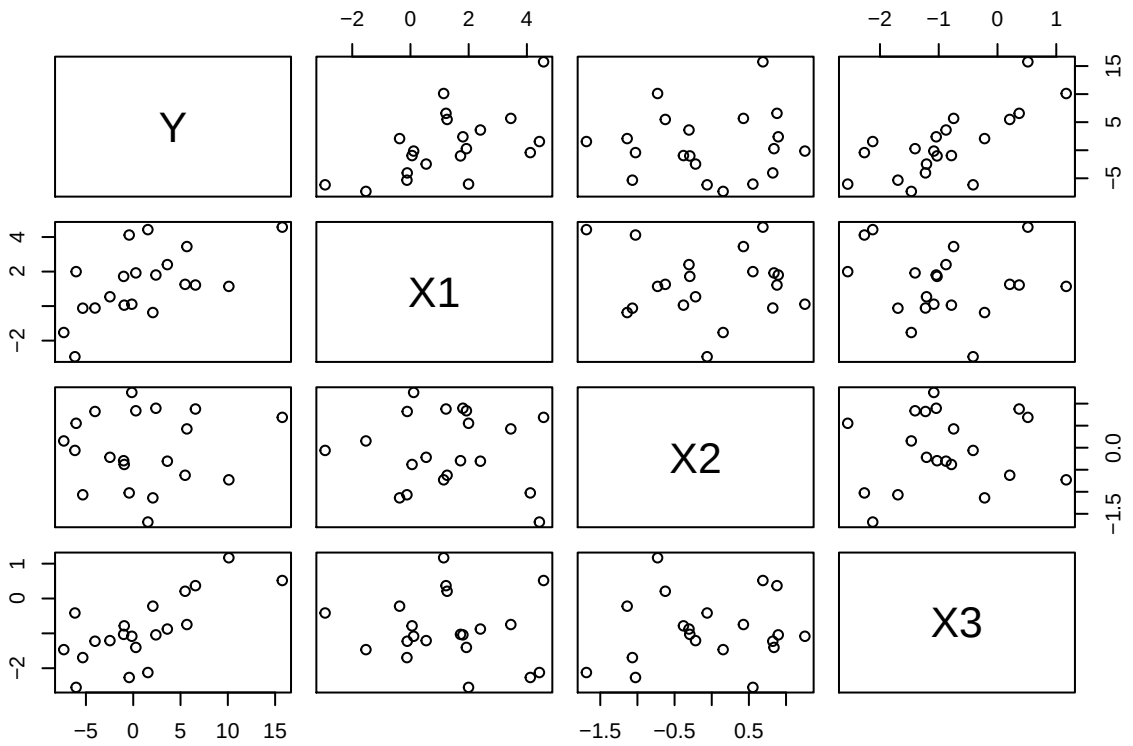
```
set.seed(123)
N <- 20
X1 <- rnorm(N, 1, 2)
X2 <- rnorm(N, 0, 1)
X3 <- rnorm(N, -1, 1)
Y <- 3 + 2*X1 + 0*X2 + 5*X3 + rnorm(N, 0, 1)
(DF <- data.frame(Y, X1, X2, X3))
```

```
##             Y          X1          X2         X3
## 1  -5.3357980 -0.12095129 -1.06782371 -1.6947070
## 2  -2.4626198  0.53964502 -0.21797491 -1.2079173
## 3  -0.4253559  4.11741663 -1.02600445 -2.2653964
## 4  10.1082380  1.14101678 -0.72889123  1.1689560
## 5   5.4851697  1.25857547 -0.62503927  0.2079620
## 6   1.5482457  4.43012997 -1.68669331 -2.1231086
## 7   0.2774504  1.92183241  0.83778704 -1.4028848
## 8  -7.3405175 -1.53012247  0.15337312 -1.4666554
## 9   2.0746817 -0.37370570 -1.13813694 -0.2200349
## 10 -0.1494085  0.10867606  1.25381492 -1.0833691
## 11  5.6718886  3.44816359  0.42646422 -0.7466815
## 12 -1.0126473  1.71962765 -0.29507148 -1.0285468
## 13  2.3944720  1.80154290  0.89512566 -1.0428705
## 14  6.5765415  1.22136543  0.87813349  0.3686023
## 15 -4.0402281 -0.11168227  0.82158108 -1.2257710
## 16 15.7555769  4.57382627  0.68864025  0.5164706
## 17 -6.0371351  1.99570096  0.55391765 -2.5487528
## 18 -6.1641176 -2.93323431 -0.06191171 -0.4153863
## 19  3.6059983  2.40271180 -0.30596266 -0.8761458
## 20 -0.9503491  0.05441718 -0.38047100 -0.7840584
```

EDA

변수 간 산점도 시각화

```
pairs(DF)
```



상관행렬도로 상관계수 확인

```
cor(DF)
```

```
##             Y          X1          X2          X3
## Y  1.00000000  0.58587054  0.05761168  0.71397869
## X1 0.58587054  1.00000000 -0.09172278 -0.12516064
## X2 0.05761168 -0.09172278  1.00000000  0.09778865
## X3 0.71397869 -0.12516064  0.09778865  1.00000000
```

통계 테스트

X행렬로 변환

```
(X <- as.matrix(cbind(rep(1,N), X1, X2, X3)))
```

```
##                  X1          X2         X3
##   [1,] 1 -0.12095129 -1.06782371 -1.6947070
##   [2,] 1  0.53964502 -0.21797491 -1.2079173
##   [3,] 1  4.11741663 -1.02600445 -2.2653964
##   [4,] 1  1.14101678 -0.72889123  1.1689560
```

```
##  [5,] 1  1.25857547 -0.62503927  0.2079620
##  [6,] 1  4.43012997 -1.68669331 -2.1231086
##  [7,] 1  1.92183241  0.83778704 -1.4028848
##  [8,] 1 -1.53012247  0.15337312 -1.4666554
##  [9,] 1 -0.37370570 -1.13813694 -0.2200349
## [10,] 1  0.10867606  1.25381492 -1.0833691
## [11,] 1  3.44816359  0.42646422 -0.7466815
## [12,] 1  1.71962765 -0.29507148 -1.0285468
## [13,] 1  1.80154290  0.89512566 -1.0428705
## [14,] 1  1.22136543  0.87813349  0.3686023
## [15,] 1 -0.11168227  0.82158108 -1.2257710
## [16,] 1  4.57382627  0.68864025  0.5164706
## [17,] 1  1.99570096  0.55391765 -2.5487528
## [18,] 1 -2.93323431 -0.06191171 -0.4153863
## [19,] 1  2.40271180 -0.30596266 -0.8761458
## [20,] 1  0.05441718 -0.38047100 -0.7840584
```

각 변수들의 회귀계수 추정

```
(Bhat <- solve(t(X)%*%X) %*% t(X) %*% Y)
```

```
##        [,1]
##    2.671753
## X1 2.062702
## X2 0.301570
## X3 4.839594
```

추정된 회귀계수와 X로 Y값 추정

```
(Yhat <- X %*% Bhat)
```

```
##              [,1]
##  [1,] -6.10145087
##  [2,] -2.12668399
##  [3,] -0.10825284
##  [4,] 10.46279100
##  [5,]  6.08577774
##  [6,]  1.02615278
##  [7,]  0.09917948
##  [8,] -7.53619798
##  [9,]  0.49280165
## [10,] -1.96903427
## [11,]  6.29926143
## [12,]  1.15209939
## [13,]  1.61067300
## [14,]  7.23977003
## [15,] -3.24308411
## [16,] 14.81337594
## [17,] -5.37959374
## [18,] -5.40760801
## [19,]  3.29537314
## [20,] -1.12526386
```

추정된 Y값으로 RSS 도출

```
(RSS <- sum((Y-Yhat)^2))
```

```
## [1] 16.23097
```

RSS를 통해 RSE 도출

```
(RSE <- sqrt(RSS/(N-3-1)))
```

```
## [1] 1.007192
```

RSE를 통해 각 회귀계수 별 표준오차와 공분산 행렬 도출

```
(SE2 <- (RSE^2) * solve(t(X)%*%X))
```

```
##                            X1           X2           X3
##     0.113506586 -0.015300997 -0.004794861  0.048567295
## X1 -0.015300997  0.014426625  0.002713702  0.003439045
## X2 -0.004794861  0.002713702  0.078772642 -0.005987783
## X3  0.048567295  0.003439045 -0.005987783  0.059637915
```

위 행렬의 대각선에 위치하는 각 변수별 표준오차 저장

```
(se <- sqrt(SE2[row(SE2)==col(SE2)]))
```

```
## [1] 0.3369074 0.1201109 0.2806646 0.2442088
```

추정된 회귀계수가 정규분포를 따른다고 가정하고, 95%의 신뢰구간을 도출

```
(CI <- cbind(Bhat-1.96*se, Bhat+1.96*se))
```

```
##          [,1]      [,2]
##     2.0114142 3.3320911
## X1  1.8272850 2.2981197
## X2 -0.2485327 0.8516727
## X3  4.3609447 5.3182430
```

추정된 회귀계수가 0과 같은지 아닌지 확인하기 위해 T value 생성

```
(T <- (Bhat-0)/se)
```

```
##         [,1]
##     7.930229
## X1 17.173317
## X2  1.074485
## X3 19.817446
```

T분포 상에서 T value의 값을 통해 유의확률을 도출

```
(P <- 2*(1-pt(T, N-3-1)))
```

```
##            [,1]
##    6.211711e-07
## X1 9.885870e-12
## X2 2.985428e-01
## X3 1.102229e-12
```

lm 메서드 사용

위와 같은 공식을 동일하게 수행하는 lm메서드 사용

```
f <- lm(Y~X1+X2+X3, data=DF)
summary(f)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3, data = DF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.16475 -0.63491 -0.07109  0.58298  1.81963
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.6718     0.3369   7.930 6.21e-07 ***
## X1            2.0627     0.1201  17.173 9.89e-12 ***
## X2            0.3016     0.2807   1.074    0.299
## X3            4.8396     0.2442  19.817 1.10e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.007 on 16 degrees of freedom
## Multiple R-squared:  0.9748, Adjusted R-squared:  0.9701
## F-statistic: 206.1 on 3 and 16 DF,  p-value: 5.402e-13
```