REPORT

수강과목	:	회귀분석(I)
담당교수	:	김충락
학 과	:	통계학과
학 번	:	201611531
이 름	:	정호재
제출일자	:	2019.06.17

Regression Analysis (I)
Project 2.

Due June 17, 2019

You may use any statistical packages like R, minitab, spsss, sas, etc.

Generation of random numbers

Let $\beta_0 = \beta_1 = \beta_2 = 1$. For $i = 1, \cdots, 30$,
$X_{1i} \sim U(1,2)$, $X_{2i} \sim U(1,2)$
$\epsilon_i \sim N(0, 0.5^2)$
$Y_i \leftarrow \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \epsilon_i$, $i = 1, \cdots, 30$

Problems

1.(25) Obtain 95% simultaneous C.I for $\beta_0 - \beta_1$, $\beta_0 - \beta_2$, $\beta_1 - \beta_2$ by the Bonferroni's method.

```
> x1<-runif(30,1,2)
> x2<-runif(30,1,2)
> e<-rnorm(30,0,0.5)
> beta0<-1
> beta1<-1
> beta2<-1
> y<-beta0+beta1*x1+beta2*x2+e
> fit<-lm(y~x1+x2)
> summary(fit)

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

Residuals:
     Min       1Q   Median       3Q      Max
-1.1798  -0.2069  -0.0471   0.3032   0.8452

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.3260     0.6598   2.010   0.0546 .
x1            0.7688     0.2859   2.689   0.0121 *
x2            0.8914     0.3814   2.337   0.0271 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.449 on 27 degrees of freedom
Multiple R-squared:  0.355,     Adjusted R-squared:  0.3073
F-statistic: 7.431 on 2 and 27 DF,  p-value: 0.002684
```

```
> beta<-summary(fit)$coefficients[,1]
> beta
(Intercept)          x1          x2
  1.3259811   0.7687878   0.8913589
```

$\hat{\beta}_0$=1.325981, $\hat{\beta}_1$=0.7687878 , $\hat{\beta}_2$=0.8913589

$\hat{\beta}=(X'X)^{-1}X'y$의 방법으로 추정 할 수도 있다.

```
> beta_hat<-solve(t(x_m)%*%x_m)%*%t(x_m)%*%matrix(y, nrow = 30)
> beta_0<-beta_hat[1,1]
> beta_1<-beta_hat[2,1]
> beta_2<-beta_hat[3,1]
```

$\beta_i-\beta_j$에 대한 $100(1-\alpha)\%$ 본페로니동시신뢰구간의 한계를
$q'\beta=q'\hat{\beta}\pm t_{a/2g}(n-p)s\sqrt{q'(X'X)^{-1}q}$을 사용하여 구하면

```
> x_m<-model.matrix(fit)#model.matrix=design matrix를 구해줌
> q1<-c(1,-1,0)
> q2<-c(1,0,-1)
> q3<-c(0,1,-1)
```

a=0.05, g=3임으로

```
> t <- qt(0.05/6,30-3)
> h<-x_m%*%solve(t(x_m)%*%x_m)%*%t(x_m)

> sse<-deviance(fit)#deviance=SSE를 구해줌
> s<-sqrt(sse/30-3)
```

$SSE=y'(I-H)y$식을 활용하여 SSE값을 구할 수도 있다.

```
> sse_1<-t(y)%*%(diag(30)-h)%*%y
> s_1<-sqrt(sse/(30-3))

> (t(q1)%*%beta)+t*s*sqrt(t(q1)%*%solve(t(x_m)%*%x_m)%*%q1)
         [,1]
[1,] 2.102736
> (t(q1)%*%beta)-t*s*sqrt(t(q1)%*%solve(t(x_m)%*%x_m)%*%q1)
          [,1]
[1,] -2.211034
> (t(q1)%*%beta)+t*s*sqrt(t(q2)%*%solve(t(x_m)%*%x_m)%*%q2)
         [,1]
[1,] 2.449526
> (t(q1)%*%beta)-t*s*sqrt(t(q2)%*%solve(t(x_m)%*%x_m)%*%q2)
          [,1]
[1,] -2.557824
> (t(q1)%*%beta)+t*s*sqrt(t(q3)%*%solve(t(x_m)%*%x_m)%*%q3)
         [,1]
[1,] 1.245535
> (t(q1)%*%beta)-t*s*sqrt(t(q3)%*%solve(t(x_m)%*%x_m)%*%q3)
          [,1]
[1,] -1.353833
```

$\beta_0-\beta_1$의 신뢰구간=(-2.211034, 2.102736)
$\beta_0-\beta_2$의 신뢰구간=(-2.557824, 2.449526)
$\beta_1-\beta_2$의 신뢰구간=(-1.353833, 1.245535)

2.(25) Test $H_0:\beta_0=\beta_1=\beta_2$ **at** $\alpha=0.05$.

way1)

가설 $H_0:\beta_0=\beta_1=\beta_2$은 모든 회귀계수들이 같으므로
$\beta_0-\beta_1=0$, $\beta_1-\beta_2=0$의 2개의 가설로 나타낼 수 있다.
이 경우는 C= $\begin{pmatrix}1 & -1 & 0\\ 0 & 1 & -1\end{pmatrix}$ m= $\begin{pmatrix}0\\0\end{pmatrix}$으로 설정하면 된다.

```
> c<-matrix(c(1,-1,0,0,1,-1),byrow=T,ncol=3)
> m<-matrix(c(0,0))
> beta_hat<-as.matrix(fit$coefficients)
> q<-2
> n<-30
> p<-3
> mse<-anova(fit)[3,2]/(30-2)
```

Lagrange방법을 사용하여

$$F_0=\frac{SSE(R)-SSE(F)}{q\cdot\ MSE(F)}=\frac{(C\hat{\beta}-m)'[C(X'X)^{-1}C']^{-1}(C\hat{\beta}-m)}{q\cdot\ MSE(F)}$$ 을 구하면

```
> F0<-t((c%*%beta_hat-m))%*%solve(c%*%solve(t(x_m)%*%x_m)%*%t(c)
+ )%*%(c%*%beta_hat-m)/(q*mse)
> F0>qf(0.05,q,n-p,lower.tail=F)
      [,1]
[1,] FALSE
```

따라서 $\alpha=0.05$ 유의수준에서의 F_0 값이 작으므로 귀무가설을 채택한다.

way2)

1번에서 구한 $\beta_0-\beta_1$, $\beta_0-\beta_2$, $\beta_1-\beta_2$들은 각각
$\beta_0-\beta_1$의 신뢰구간=(-2.211034, 2.102736)
$\beta_0-\beta_2$의 신뢰구간=(-2.557824, 2.449526)
$\beta_1-\beta_2$의 신뢰구간=(-1.353833, 1.245535) 으로 0을 포함한다.

따라서 각각의 $H_0:\beta_0=\beta_1$, $H_0:\beta_0=\beta_2$, $H_0:\beta_1=\beta_2$에 대한 양측검정을 실시한다면 유의수준 0.05하에서 귀무가설을 기각할 수 없게 된다.

따라서 $\beta_0=\beta_1=\beta_2$을 만족하고 최종모델인 $H_0:\beta_0=\beta_1=\beta_2$을 채택한다.

3.(25) Compute residual, leverage, and Cook's distance for each observation.

```
> residuals(fit)

          1           2           3           4           5           6 
 0.44657418 -0.09082397 -0.31970873 -0.07940629  0.09329628  0.24068859 
          7           8           9          10          11          12 
 0.32285722 -0.06440994 -0.19884855 -0.18516023 -0.36580098  0.54432372 
         13          14          15          16          17          18 
-0.18955659  0.14875801 -0.51006687  0.84519201  0.48563491  0.39099832 
         19          20          21          22          23          24 
 0.41125551 -1.17976043 -0.30647394  0.05481547 -0.02294093 -0.12943132 
         25          26          27          28          29          30 
 0.76884335 -0.20951572  0.24416970 -0.02978214 -0.68731427 -0.42840639 
```

```
> hatvalues(fit)

         1          2          3          4          5          6          7 
0.08780122 0.32818584 0.10485659 0.10579341 0.04723440 0.11572931 0.13705403 
         8          9         10         11         12         13         14 
0.17917977 0.06357847 0.09584002 0.15095995 0.09407766 0.03886679 0.03737565 
        15         16         17         18         19         20         21 
0.18258218 0.06390418 0.05475787 0.03744967 0.05896275 0.04304913 0.09973118 
        22         23         24         25         26         27         28 
0.07166718 0.06095987 0.16475912 0.03862048 0.05381919 0.12240402 0.04105955 
        29         30 
0.21767695 0.10206358 
```

```
> cooks.distance(fit)

           1            2            3            4            5 
3.479423e-02 9.917929e-03 2.211658e-02 1.379403e-03 7.488821e-04 
           6            7            8            9           10 
1.417694e-02 3.172079e-02 1.824303e-03 4.740352e-03 6.645845e-03 
          11           12           13           14           15 
4.633334e-02 5.615877e-02 2.499700e-03 1.475823e-03 1.175501e-01 
          16           17           18           19           20 
8.613861e-02 2.389894e-02 1.021761e-02 1.862027e-02 1.081860e-01 
          21           22           23           24           25 
1.911050e-02 4.131595e-04 6.015839e-05 6.541890e-03 4.084154e-02 
          26           27           28           29           30 
4.363355e-03 1.566709e-02 6.548493e-05 2.778110e-01 3.841404e-02 
```

4.(25) Compute the condition number for the design matrix, and the variance inflation factor for X_1 and X_2.

way1 (Compute the condition number for the design matrix)

행렬 X를 비정칙분해하여 만들어진 singular values를 구하면

```
> singval<-svd(x_m)
> singval
$d
[1] 12.8131246  1.3129010  0.6085628

$u
            [,1]         [,2]         [,3]
 [1,] -0.1685347  0.236171784  0.060167691
 [2,] -0.1829682 -0.504172965  0.201291085
 [3,] -0.1544144  0.025256095  0.283504705
 [4,] -0.1555165  0.198421246  0.205516512
 [5,] -0.1892553 -0.105666178 -0.015858381
 [6,] -0.1945403 -0.279069269 -0.001932744
 [7,] -0.2153347 -0.152294658 -0.259790969
 [8,] -0.2207439 -0.034893876 -0.359491761
 [9,] -0.1653191  0.139126340  0.129968893
[10,] -0.1744468  0.255661499 -0.006747273
[11,] -0.1460876  0.054349954  0.355899495
[12,] -0.1704534  0.252648124  0.034528386
[13,] -0.1754182  0.071449941  0.054682221
[14,] -0.1748895  0.039324904  0.072407655
[15,] -0.1951604  0.300896689 -0.232283834
[16,] -0.1781672  0.178754995 -0.014396782
[17,] -0.1907730 -0.134005589 -0.020150686
[18,] -0.1753088  0.003971547  0.081858000
[19,] -0.1676743 -0.032443010  0.172613814
[20,] -0.1713442  0.061887100  0.099298945
[21,] -0.1833003 -0.238808432  0.095408232
[22,] -0.1709359  0.199848595  0.050086418
[23,] -0.1894443 -0.158316173  0.002589188
[24,] -0.2119324  0.111746140 -0.327653144
[25,] -0.1772768  0.077910205  0.033517233
[26,] -0.1967333 -0.044571363 -0.114580023
[27,] -0.1965680 -0.288828230 -0.018528500
[28,] -0.1722740  0.029027967  0.102657676
[29,] -0.2264321 -0.075775750 -0.400828482
[30,] -0.1550439  0.021928374  0.278467443

$v
           [,1]       [,2]       [,3]
[1,] -0.4250557  0.1595977  0.8909861
[2,] -0.6461239 -0.7428572 -0.1751771
[3,] -0.6339176  0.6501475 -0.4188756
```

조건수는 행렬 X의 가장 큰 비정칙지에서 가장 작은 비정칙치를 나눈 값이므로

```
> max(singval$d)/min(singval$d)
[1] 21.05473
```

way2 (Compute the condition number for the design matrix)

$X'X$의 고유치를 구하면

```
> eigenval<-eigen(t(x_m)%*%x_m)
> eigenval
eigen() decomposition
$values
[1] 164.1761628   1.7237089   0.3703487

$vectors
          [,1]       [,2]       [,3]
[1,] 0.4250557  0.1595977  0.8909861
[2,] 0.6461239 -0.7428572 -0.1751771
[3,] 0.6339176  0.6501475 -0.4188756
```

조건수는 행렬 $X'X$의 가장 큰 고유치에서 가장 작은 고유치를 나눈 값이므로

```
> sqrt(max(eigenval$values)/min(eigenval$values))
[1] 21.05473
```

따라서 design matrix의 condition number은 21.05473이다.

way (the variance inflation factor for X_1 and X_2.)

VIF를 구하기 위해 vif함수가 있는 car패키지를 다운받는다.

```
> require(car)
필요한 패키지를 로딩중입니다: car
필요한 패키지를 로딩중입니다: carData
> vif(fit)
      x1       x2
1.022116 1.022116
> sqrt(vif(fit))>10
   x1    x2
FALSE FALSE
```

vif 함수로 fit 모형 검사결과 다중공선성 문제는 없다는 것을 알 수 있다.