hw3 draft

Yujia

2022-10-04

Typed Problem 1

Build a linearly dependant matrix m1:

```
x1 <- c(6,2,9,1)
x2 <- c(2,4,1,2)
x3 <- c(4,-2,8,-1)
m1 <- cbind(x1,x2,x3)
lmod <- lm(x1~x3+x2)
summary(lmod)
```

```
## Warning in summary.lm(lmod): essentially perfect fit: summary may be unreliable
```

```
##
## Call:
## lm(formula = x1 ~ x3 + x2)
##
## Residuals:
##          1          2          3          4
## -1.846e-16  5.430e-17  1.086e-16  2.172e-17
##
## Coefficients:
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) 4.441e-16  4.964e-16 8.950e-01    0.535
## x3          1.000e+00  4.734e-17 2.112e+16   <2e-16 ***
## x2          1.000e+00  1.748e-16 5.721e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.22e-16 on 1 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 4.158e+32 on 2 and 1 DF,  p-value: < 2.2e-16
```

x1, x2, x3 are linearly dependent. Regress x1 on x2 and x3, we will get that the R-squared equals to 1 and RSS=0. Also, since x1 is depend on both x1 and x2, the regression p-value of x2 and x3 are both less than 0.05. In general, A set of vectors $\{x_1, x_2, \cdots, x_p\}$ is linearly dependent if there exist numbers $\beta_1, \beta_2, \cdots \beta_p$,not all equal to zero, such that
$\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_P = 0$
It can be re-written as
$x_1 = -(\beta_2 x_2 + \cdots + \beta_p x_P)/\beta_1$
Therefore, we can get a perfect fit and the rss will be 0.

Typed Problem 2

Build a linearly independant matrix m2:

```
m2 <- matrix(rnorm(10000* 2),     # Create random matrix
                 ncol = 2)
cor(m2[,1],m2[,2])
```

```
## [1] -0.02293641
```

The correlation of this sample is very closed to 0. We can conclude that in general, if numerical predictors are linearly independent, then they are uncorrelated.

Typed Problem 3

a

```
myOLS <- function(Y, X, is1 = TRUE) {
  if (is1 == TRUE)
n <- nrow(X)
p <- ncol(X)
if (is1 == FALSE){x1 <- cbind(rep(1, n),X)}
else {x1 <- X}

beta <- solve(crossprod(x1,x1),crossprod(x1,Y))
xtxi <- solve(t(x1) %*% x1)
H <- x1 %*% xtxi %*% t(x1)
Rss <- t(Y) %*% (diag(1,n)-H) %*% Y
sigma <- sqrt(Rss/(n-p))
se <- sqrt(diag(xtxi))*sigma
list(beta,se)
}
n = 30
set.seed(0)
p = 3
X = matrix(runif(n*p),nrow=n)*2-1
b = seq(1,p,by=1)
Y = X%*%b + rnorm(n)
fit1 = lm(Y ~ X); summary(fit1); # regression with an intercept
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3701 -0.3304  0.1082  0.4938  2.3930
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1291     0.1708  -0.756  0.45648
## X1            0.9214     0.2987   3.085  0.00478 **
## X2            2.4021     0.3468   6.926 2.36e-07 ***
## X3            2.7482     0.3452   7.960 1.94e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9235 on 26 degrees of freedom
## Multiple R-squared:  0.802,  Adjusted R-squared:  0.7792
## F-statistic: 35.11 on 3 and 26 DF,  p-value: 2.711e-09
```

```r
myOLS(Y,X,FALSE)
```

```
## Warning in sqrt(diag(xtxi)) * sigma: Recycling array of length 1 in vector-array arithmetic is depre
##   Use c() or as.vector() instead.

## [[1]]
##            [,1]
## [1,] -0.1291359
## [2,]  0.9214173
## [3,]  2.4020865
## [4,]  2.7482181
##
## [[2]]
## [1] 0.1676341 0.2930771 0.3403307 0.3387858
```

```r
fit0 = lm(Y ~ -1 + X); summary(fit0) # regression without an intercept
```

```
##
## Call:
## lm(formula = Y ~ -1 + X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.48956 -0.47760 -0.00264  0.37870  2.29687
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## X1   0.8929     0.2939   3.038  0.00523 **
## X2   2.3868     0.3435   6.949 1.81e-07 ***
## X3   2.7167     0.3400   7.991 1.38e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9161 on 27 degrees of freedom
## Multiple R-squared:  0.7982, Adjusted R-squared:  0.7758
## F-statistic: 35.61 on 3 and 27 DF,  p-value: 1.583e-09
```

```r
myOLS(Y,X,TRUE)
```

```
## Warning in sqrt(diag(xtxi)) * sigma: Recycling array of length 1 in vector-array arithmetic is depre
##   Use c() or as.vector() instead.

## [[1]]
##          [,1]
## [1,] 0.892872
## [2,] 2.386783
## [3,] 2.716730
##
## [[2]]
## [1] 0.2939026 0.3434637 0.3399867
```

b