Stat632 HW 4

Brandon Keck

Exercise 1.

(a)

Given that $H = (X'X)^{-1}X'$

Show that $HH' = HH = H$

$HH = [X(X'X)^{-1}X'][X(X'X)^{-1}X']$

which then $= [X(X'X)^{-1}][(X'X)(X'X)^{-1}X']$

Because we know that $(X'X)(X'X)^{-1} = I$

Thus:

$HH = X(X'X)^{-1}(X'X)(X'X)^{-1}X'$

$= X(X'X)^{-1}X' = H$

Therefore, $HH' = HH = H$

$\square$

(b)

Show that $E(\hat{Y}) = X\beta$

We know that:

$\hat{Y} = X\hat{\beta} = X(X'X)^{-1}X'Y = HY$

and that the regression model can be written as

$Y = X\beta + \epsilon$

Since $E(\epsilon) = 0$

$\hat{Y} = X\hat{\beta}$

Therefore,

$E(\widehat{\beta}) = E((X'X)^{-1}X'Y)$

$= [(X'X)^{-1}X'][E(Y)]$

$= (X'X)^{-1}X'(X\beta)$

$= \beta$

$E(\widehat{Y}) = E(X\widehat{\beta})$

$= X[E(\widehat{\beta})]$

$= X\beta$

$\square$

(c)

Show that $Var(\hat{Y}) = \sigma^2 H$

$Var(\hat{Y} = HVar(Y)H' = H\sigma^2 IH' = \sigma^2 HH'$

Since $H = H'\ and\ HH = H$

$Var(\hat{Y}) = \sigma^2 H$

$\square$

Exercise 2.

$(X'X)^{-1} = \frac{1}{nS_{XX}} \begin{pmatrix} \sum x_i^2 & -\sum x_i \\ -\sum x_i & n \end{pmatrix}$

where:

Sxx $= \sum(x_i - \bar{x})^2 = \sum x_i 2 - n\bar{x}2$

$Var(\widehat{\beta}) = \frac{\sigma^2}{nSxx} \begin{pmatrix} \sum x_i^2 & -\sum x_i \\ -\sum x_i & n \end{pmatrix}$

$Var(\widehat{\beta}_0) = \sigma^2 \frac{\sum x_i^2}{nSxx} = \sigma^2(\frac{1}{n} + \frac{\bar{x}^2}{Sxx})$

$Var(\widehat{\beta}_1) = \sigma^2 \cdot \frac{n}{nS_{XX}} = \frac{\sigma^2}{S_{XX}}$

$Cov(\widehat{\beta}_0, \widehat{\beta}_1) = -\frac{\sigma^2\bar{x}}{S_{XX}}$

$\square$

Exercise 3.

(a)

```
library(MASS)
library(dplyr)
data(Boston)
```

```
# design matrix with intercept
X <- cbind(Intercept = 1, Boston[,c("dis", "rm", "tax", "chas")])
X <- as.matrix(X)
Y <- Boston$medv

rownames(X) <- Boston$medv
X[1:4,]
```

```
     Intercept    dis    rm tax chas
24           1 4.0900 6.575 296    0
21.6         1 4.9671 6.421 242    0
34.7         1 4.9671 7.185 242    0
33.4         1 6.0622 6.998 222    0
```

```
# manually calculate least squares estimates
betaHat <- solve(t(X) %*% X) %*% t(X) %*% Y
betaHat
```

```
                  [,1]
Intercept -20.16720221
dis        -0.10656777
rm          7.88589232
tax        -0.01647039
chas        3.87901205
```

```
# Compare with lm()
lm1 <- lm(medv ~ dis + rm + tax + chas, data = Boston)
coef(lm1)
```

```
 (Intercept)          dis           rm          tax         chas
-20.16720221  -0.10656777   7.88589232  -0.01647039   3.87901205
```

(b)

```r
n <- nrow(Boston)
p <- 4

# Manually calculate standard errors for least squares estimates
resid <- as.numeric(Y - X %*% betaHat)
sigmaHat2 <- sum(resid^2) / (n-p-1)
covBetaHat <- sigmaHat2 * solve(t(X) %*% X)

covBetaHat
```

```
             Intercept           dis            rm           tax         chas
Intercept  8.510688175 -0.1236974635 -1.0632321031 -3.155777e-03  0.014868410
dis       -0.123697464  0.0233402861 -0.0045240274  1.516224e-04  0.023912819
rm        -1.063232103 -0.0045240274  0.1616761190  1.644565e-04 -0.040647554
tax       -0.003155777  0.0001516224  0.0001644565  3.759877e-06  0.000171935
chas       0.014868410  0.0239128186 -0.0406475538  1.719350e-04  1.151456124
```

```r
seBetaHat <- sqrt(diag(covBetaHat))
seBetaHat
```

```
 Intercept        dis         rm        tax       chas
2.91730838 0.15277528 0.40208969 0.00193904 1.07305924
```

```r
# Compare with lm()
summary(lm1)$coef
```

```
               Estimate Std. Error    t value     Pr(>|t|)
(Intercept) -20.16720221 2.91730838 -6.9129484 1.450596e-11
dis          -0.10656777 0.15277528 -0.6975459 4.857848e-01
rm            7.88589232 0.40208969 19.6122719 5.663118e-64
tax          -0.01647039 0.00193904 -8.4940923 2.286843e-16
chas          3.87901205 1.07305924  3.6149095 3.308341e-04
```