Advanced Applied Statistics Homework 1 Answer Key

Joshua D. Ingram

2022-09-08

1. Linear Model Theory

Consider the following linear model for the response vector $\boldsymbol{y}$:

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where we assume

$$\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 I)$$

and hence

$$\boldsymbol{\mu} = E[\boldsymbol{y}] = X\boldsymbol{\beta}$$

Note that to multiply two matrices `A` and `B` in R, you write `A %*% B`, and to get the inverse of a matrix `A`, you write `solve(A)`. To get the transpose of a matrix `A`, use `t(A)`.

a. Suppose $\boldsymbol{y} = (2, 4, 3)^t$ and there is only one predictor $\boldsymbol{x} = (1, 2, 3)^t$. You fit the simple linear regression model with an intercept. Show the model matrix X in R.

Answer:

The model matrix X is given by

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}.$$

In R, this is given by

```r
# Create response vector y
y <- matrix(c(2,4,3), nrow = 3, ncol = 1)
# Create model matrix X
X <- matrix(c(1, 1, 1, 1, 2, 3), nrow = 3)
print(X)
```

```
##      [,1] [,2]
## [1,]    1    1
## [2,]    1    2
## [3,]    1    3
```

b. (theoretical) State the least squares estimate $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ in general matrix form.

Answer:

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

c. (practical) Plug your X matrix from part a into the formula from part b and show the result. Does it agree with the result from `coefficients(fit)` where `fit <- lm(y~x)`? Hint: In R, make sure you define `y` as a matrix, i.e., `y <- matrix(c(2,4,3), nrow=3,ncol=1)` and not simply `y <- c(2,4,3)`, as otherwise you can't do proper matrix multiplication in R.

Answer:

```r
# Calculate beta-hat vector
beta_hat <- solve(t(X)%*%X) %*% t(X) %*% y

# fit simple linear regression model
fit <- lm(y~ X[,2])

# Coefficients from lm()
coef <- coefficients(fit)

# Coefficients from formula in part b
print(beta_hat)
```

```
##      [,1]
## [1,]  2.0
## [2,]  0.5
```

```r
# Cofficients from fit using lm():
print(coef)
```

```
## (Intercept)      X[, 2]
##         2.0         0.5
```

The manual calculations agree with the results from the built in `coefficients(fit)`.

d. (practical) In class, we showed that $\text{Var}[\hat{\beta}] = \sigma^2(X^tX)^{-1}$. Find the estimate of σ from `summary(fit)` (I told you where to find this on the output), square it to get the estimate of σ^2, and then use this estimate to estimate $\text{Var}[\hat{\beta}]$. Print out this estimated variance-covariance matrix and compare with what you get from `vcov(fit)`.

Answer:

```r
# sigma from summary fit function (listed as residual standard error when calling summary(fit))
sigma <- summary(fit)$sigma

# variance-covariance matrix
var_beta_hat <- sigma^2 * solve(t(X)%*%X)

# variance-covariance from vcov()
vcov <- vcov(fit)

# Calculated Variance-Covariance Matrix
print(var_beta_hat)
```

```
##      [,1]  [,2]
## [1,]  3.5 -1.50
## [2,] -1.5  0.75
```

```r
# Variance-Covariance from vcov()
print(vcov)
```

```
##             (Intercept) X[, 2]
## (Intercept)         3.5  -1.50
```

```
## X[, 2]              -1.5   0.75
```

The manual calculation of the variance-covariance matrix agree with the results from the built in `vcov(fit)`.

e. (practical) Refer to the estimate of the variance-covariance matrix you just computed in part d. Get its diagonal elements (R command `diag`) and take the square root of the diagonal elements? Compare what you get with the output from `summary(fit)`, especially the standard errors!

Answer:

```r
se_beta_hat <- sqrt(diag(var_beta_hat))

sum_fit <- summary(fit)

# Calculated standard errors of coefficients
print(se_beta_hat)
```

```
## [1] 1.8708287 0.8660254
```

```r
# Standard Errors of coefficients from lm()
print(sum_fit$coefficients)
```

```
##             Estimate Std. Error   t value  Pr(>|t|)
## (Intercept)      2.0  1.8708287 1.0690450 0.4787636
## X[, 2]           0.5  0.8660254 0.5773503 0.6666667
```

The standard errors given in `summary(fit)$coefficients` are the same as the square root of the diagonals of the estimated variance-covariance matrix we calculate.

f. (theoretical) Give the hat matrix H for which $\hat{\boldsymbol{y}} = H\boldsymbol{y}$.

Answer:

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

Thus,

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

g. (practical) Plug the X matrix from part a into H and compute H. Also, just for fun, compute i) H^t and ii) HH. What do you notice? Finally, use H to compute $\hat{\boldsymbol{y}}$. Does this result agree with the R command `fitted(fit)`?

Answer:

```r
H <- X %*% solve(t(X)%*%X) %*% t(X)
print(H)
```

```
##            [,1]      [,2]       [,3]
## [1,]  0.8333333 0.3333333 -0.1666667
## [2,]  0.3333333 0.3333333  0.3333333
## [3,] -0.1666667 0.3333333  0.8333333
```

i)

```r
H_t <- t(H)
print(H_t)
```

```
##            [,1]      [,2]       [,3]
## [1,]  0.8333333 0.3333333 -0.1666667
```

```
## [2,]  0.3333333 0.3333333  0.3333333
## [3,] -0.1666667 0.3333333  0.8333333
```

ii)

```r
HH <- H %*% H
print(HH)
```

```
##            [,1]      [,2]       [,3]
## [1,]  0.8333333 0.3333333 -0.1666667
## [2,]  0.3333333 0.3333333  0.3333333
## [3,] -0.1666667 0.3333333  0.8333333
```

```r
y_hat <- H %*% y
# Predictions from calculated y-hat
print(y_hat)
```

```
##      [,1]
## [1,]  2.5
## [2,]  3.0
## [3,]  3.5
```

```r
# Predictions from fitted()
fitted(fit)
```

```
##   1   2   3
## 2.5 3.0 3.5
```

H has the properties of symmetry and idempotency, so $HH = H$ and $H' = H$. Using the hat matrix H to compute $\hat{\boldsymbol{y}} = H\boldsymbol{y}$ gives the same results as using the `fitted(fit)` function.

h. (theoretical) Show that $\mathrm{E}[\hat{\boldsymbol{y}}] = \boldsymbol{\mu}$.

Answer:

$$E[\hat{\boldsymbol{y}}] = E[H\mathbf{y}] = HE[\mathbf{y}] = HX\boldsymbol{\beta} + HE[\boldsymbol{\epsilon}]$$

$$= HX\boldsymbol{\beta} + \mathbf{0} = X(X'X)^{-1}X'X\boldsymbol{\beta} = X\boldsymbol{\beta} = \boldsymbol{\mu}.$$

i. (theoretical) Show that $\mathrm{Var}[\hat{\boldsymbol{y}}] = \sigma^2 H$, using the fact (see part g) that $HH = H$. (H is "idempotent")

Answer:

$$Var[\hat{\boldsymbol{y}}] = Var[H\boldsymbol{y}] = HVar[\boldsymbol{y}]H' = H\sigma^2 IH' = \sigma^2 HIH = \sigma^2 H.$$

Note that $HH = H$ due to the symmetry and idempotency of H.

j. (practical) Find the estimate of σ from `summary(fit)` (I told you where to find this on the output), square it to get the estimate of σ^2, and then use it and your computed H from part g to find the estimate of $\sigma^2 H$. Then, take the diagonal of this matrix (R command `diag`) and then take the square root of these diagonal elements. Compare these to what you get from the output of `predict(fit, se=TRUE, interval='confidence')`. (This output is what you would use to find a confidence interval for the mean response at the given x-values. To compute the confidence interval, you need the standard errors.)

Answer:

```r
sqrt(diag(sigma^2 * H))
```

```
## [1] 1.1180340 0.7071068 1.1180340
```

```r
predict(fit, se=TRUE, interval="confidence")
```

```
## $fit
##   fit        lwr      upr
## 1 2.5 -11.705969 16.70597
## 2 3.0  -5.984644 11.98464
## 3 3.5 -10.705969 17.70597
##
## $se.fit
## [1] 1.1180340 0.7071068 1.1180340
##
## $df
## [1] 1
##
## $residual.scale
## [1] 1.224745
```

We obtain the standard error used to calculate the confidence interval for the mean response of the given x-value.

k. (theoretical) The residuals are defined as $\boldsymbol{r} = \boldsymbol{y} - \hat{\boldsymbol{y}}$. Show that $\boldsymbol{r} = (I - H)\boldsymbol{y}$.

Answer:

$$\boldsymbol{r} = \boldsymbol{y} - \hat{\boldsymbol{y}} = \boldsymbol{y} - H\boldsymbol{y} = (I - H)\boldsymbol{y}$$