HW3

Robert Chandler

2024-06-30

Setup

```r
# Read dataset
le <- read.csv("../../datasets/life_expectancy.csv")

matrix_string_list <- function(
    matrix,
    inline = FALSE,
    maxrows = 7,
    maxcols = 7,
    show_dimension = NULL) {
  nrows <- nrow(matrix)
  ncols <- ncol(matrix)

  begin <- "\\begin{bmatrix}"

  if (is.null(show_dimension)) {
    if (nrows > maxrows || ncols > maxcols) {
      show_dimension <- TRUE
    } else {
      show_dimension <- FALSE
    }
  }

  dimension_string <- if (show_dimension) {
    paste(
      "_{", nrows, " \\times ", ncols, "}",
      sep = ""
    )
  } else {
    ""
  }

  end <- paste("\\end{bmatrix}", dimension_string, sep = "")

  if (!inline) {
    begin <- c("$$", begin)
    end <- c(end, "$$")
  }

  if (nrows > maxrows) {
    matrix <- matrix[c(seq_len(maxrows - 1), nrows), ]
```

```r
  }

  matrix_strings <- apply(matrix, 1, function(row) {
    str <- if (length(row) > maxcols) {
      paste(
        c(
          row[seq_len(maxcols - 1)],
          "\\dots",
          tail(row, 1)
        ),
        collapse = " & "
      )
    } else {
      paste(row, collapse = " & ")
    }

    str <- paste(str, "\\\\")
  })

  if (nrows > maxrows) {
    dots <- rep("\\vdots", min(ncols, maxcols))
    if (ncols > maxcols) {
      dots <- append(dots, "\\ddots", length(dots) - 1)
    }
    dots_string <- paste(paste(dots, collapse = " & "), "\\\\")
    matrix_strings <- append(
      matrix_strings, dots_string, length(matrix_strings) - 1
    )
  }

  c(begin, matrix_strings, end)
}

print_matrix <- function(matrix, maxrows = 10, maxcols = 10) {
  writeLines(matrix_string_list(
    matrix,
    inline = FALSE, maxrows = maxrows, maxcols = maxcols
  ))
}

matrix_string <- function(
    matrix, maxrows = 7, maxcols = 7, show_dimension = NULL) {
  paste(matrix_string_list(
    matrix,
    inline = TRUE, maxrows = maxrows, maxcols = maxcols, show_dimension =
      show_dimension
  ), collapse = "\n")
}

# Utility function for latex-printing matrices
```

```r
y <- le$X2015Life.expectancy
x1 <- le$Medical.doctors
x2 <- le$Nurses
```

```r
x3 <- le$Pharmacists
```

Utilize the life expectancy data and examine a simple linear regression $Y \sim X$, where $X = X_3$, and Y represents life expectancy.

Set X accordingly:

```r
x <- x3
```

Problem 1

At a confidence level of 95%…

1.a

```r
alpha <- 0.05
n <- length(y)
p <- 2
t_crit <- qt(1 - alpha / 2, n - p)
crit_bonf <- qt(1 - alpha / 4, n - p)
```

To find the individual confidence intervals for β_0, the critical value is denoted by $t(1 - \alpha/2, n - 2)$ and has a value of $t(1 - 0.05/2, 178 - 2) = 1.9735344$.

1.b

To find the individual confidence intervals for β_1, the critical value is denoted by $t(1 - \alpha/2, n - 2)$ and has a value of $t(1 - 0.05/2, 178 - 2) = 1.9735344$.

1.c

To calculate the Bonferroni joint confidence intervals for all parameters using an $\alpha = 0.05$, the critical value is denoted by $B = t(1 - \frac{\alpha}{2p}, DFE) = t(1 - \frac{\alpha}{4}, n - p)$ and has a value of $t(1 - \frac{0.05}{4}, 178 - 2) = 2.2607406$.

1.d

Calculate the Working-Hotelling joint confidence intervals for all parameters using an $\alpha = 0.05$.

```r
w <- sqrt(p * qf(1 - alpha, p, n - p))
model <- lm(y ~ x)
model_summary <- summary(model)
model_summary
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1354  -3.7151   0.3917   4.3968  14.2473
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 66.88378    0.63965  104.56   <2e-16 ***
## x            1.09415    0.09978   10.96   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.27 on 176 degrees of freedom
## Multiple R-squared:  0.4059, Adjusted R-squared:  0.4025
## F-statistic: 120.2 on 1 and 176 DF,  p-value: < 2.2e-16
```

```r
b0 <- model$coefficients[["(Intercept)"]]
b1 <- model$coefficients[["x"]]
t0 <- model_summary$coefficients["(Intercept)", "t value"]
t1 <- model_summary$coefficients["x", "t value"]

b0_interval <- c(b0 - w * t0, b0 + w * t0)
b1_interval <- c(b1 - w * t1, b1 + w * t1)
```

The interval for b_0 is $(-191.2554462, 325.0230075)$, and the interval for b_1 is $(-25.9765722, 28.1648684)$

The critical value is denoted by $W = \sqrt{2F(1-\alpha; p, n-p)}$ and has a value of $W = \sqrt{2F(1-\alpha; p, n-p)} = \sqrt{2F(1-0.05; 2, 178-2)} = 2.4687271$.

1.e

Which joint confidence interval is better? Briefly explain

For joint estimation of the *parameters*, the Bonferroni interval is better when $n - p \geq 2$ because it yields a smaller critical value: $B = 2.2607406 < 2.4687271 = W$. This is consistent across all models when we are estimating the parameters/weights/coefficients simultaneously for a linear model. That is, $W/B > 1$.

Problem 2

At the confidence level of 95%...

2.a

```r
t_crit <- qt(1 - alpha / 2, n - p)
mse <- anova(model)["Residuals", "Mean Sq"]
std_err_mean <- sqrt(mse / n)
```

To find the individual confidence interval for the mean response $\hat{Y}_h$, where X_h = sample mean, the critical value is denoted by

$$t(1-\alpha/2, n-2) = t(1-0.05/2, 178-2)$$

and has a value of 1.9735344, the standard error can be computed with the formula

$$s\{\hat{Y}_h\} = \sqrt{MSE\left[\frac{1}{n} + \frac{(X_h - \bar{X})^2}{\sum(X_i - \bar{X})^2}\right]} = \sqrt{39.3179977\left[\frac{1}{178} + 0\right]}$$

and has a value of 0.4699868.

2.b

```
x_h <- median(x)
xbar <- mean(x)
sst_x <- sum((x - xbar)^2)
std_err_median <- sqrt(mse * (1 / n + (x_h - xbar)^2 / sst_x))
```

To find the individual confidence interval for the mean response $\hat{Y}_h$, where X_h = sample median, the critical value is denoted by

$$t(1-\alpha/2, n-2) = t(1-0.05/2, 178-2)$$

and has a value of 1.9735344, the standard error can be computed with the formula

$$s\{\hat{Y}_h\} = \sqrt{MSE\left[\frac{1}{n} + \frac{(X_h-\bar{X})^2}{\sum(X_i-\bar{X})^2}\right]} = \sqrt{39.3179977\left[\frac{1}{178} + \frac{(2.32-4.3483989)^2}{\sum(X_i-4.3483989)^2}\right]}$$

and has a value of 0.5117147.

2.c

To compute the joint confidence interval for the mean responses for 5 different X_h levels (data points), the Bonferroni critical value is denoted by

$$B = t\left(1-\frac{\alpha}{2g}, n-p\right)$$

and has a value of

```
g <- 5
crit_bonf_5 <- qt(1 - alpha / (2 * g), n - p)
```

$$t\left(1-\frac{0.05}{2*5}, 178-2\right) = 2.6040517$$

The WH critical value is denoted by

$$W = \sqrt{2F(1-\alpha; 2, n-p)}$$

and has a value of

$$W = \sqrt{2F(1-\alpha; 2, n-p)} = \sqrt{2F(1-0.05; 2, 178-2)} = 2.4687271$$

Which method is better here and why?

For joint estimation of the *mean response* of 5 values, the Working-Hotelling interval is better because it yields a smaller critical value: $W = 2.4687271 < 2.6040517 = B$. Working-Hotelling gives a better estimate than Bonferroni any time $g \geq 3$.

Which method should you suggest if you are not sure what X_h levels are to be predicted?

When the X_h levels are not known, it is better to use the Working-Hotelling method since it yields the same interval for all levels of X_h while the Bonferroni method changes with the number of levels, g.

2.d

To calculate the Bonferroni joint confidence interval for the single response prediction at 3 X_h levels, the critical value is denoted by

$$B = t\left(1 - \frac{\alpha}{2g}, n - p\right)$$

and has a value of

```
g <- 3
crit_bonf_3 <- qt(1 - alpha / (2 * g), n - p)
```

$$t\left(1 - \frac{0.05}{2 * 3}, 178 - 2\right) = 2.6040517$$

2.e

To calculate the Scheffe joint confidence interval for the single response predictions at 3 X_h levels, the critical value is denoted by

$$S = \sqrt{gF(1 - \alpha; g, n - p)}$$

and has a value of

```
crit_scheffe_3 <- qf(1 - alpha, g, n - p)
```

$$S = \sqrt{3F(1 - 0.05; 3, 178 - 2)} = 2.6559389$$

MLR Setup

For Problems 3-8, consider a multiple linear regression (MLR) model where $Y \sim X_1 + X_2 + X_3$ in the life expectancy data.

```
model <- lm(y ~ x1 + x2 + x3)
p <- 4
```

Problem 3

Use R to complete the response value vector, **Y** and the design matrix **X**, hat matrix **H**, and the parameter matrix β. Use correct matrix notations and type all general formulas clearly, show R code and output. Note that partial results are allowed for large matrix output such as **Y**, **X**, and **H**, but make sure to describe the dimension of all matrices. Then compute the SS terms with the following matrix operation.

The response value vector **Y** is $65, 77.8, 75.6, 52.4, 76.4, 76.3$ ... $69.4, 72, 76, 65.7, 61.8, 67$ with dimension (178).

The design matrix **X** is

```
x <- cbind(1, x1, x2, x3)
```

$$\begin{bmatrix} 1 & 2.54 & 4.52 & 0.29 \\ 1 & 18.83 & 58.31 & 10.86 \\ 1 & 17.32 & 15.59 & 4.49 \\ 1 & 2.11 & 4.01 & 0.74 \\ 1 & 28.98 & 95.77 & 1.83 \\ 1 & 38.95 & 54.47 & 4.97 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1.89 & 20.29 & 1.02 \end{bmatrix}_{178\times 4}$$

Dimensions are shown in the subscript.

The hat matrix H is:

```
hat <- x %*% solve(t(x) %*% x) %*% t(x)
```

$$\begin{bmatrix} 0.012 & 0.003 & 0.007 & 0.012 & 0.002 & 0.002 & \dots & 0.011 \\ 0.003 & 0.028 & 0.009 & 0.005 & -0.011 & -0.004 & \dots & 0.006 \\ 0.007 & 0.009 & 0.012 & 0.008 & -0.005 & 0.009 & \dots & 0.004 \\ 0.012 & 0.005 & 0.008 & 0.012 & 0.001 & 0.001 & \dots & 0.011 \\ 0.002 & -0.011 & -0.005 & 0.001 & 0.03 & 0.008 & \dots & 0.005 \\ 0.002 & -0.004 & 0.009 & 0.001 & 0.008 & 0.022 & \dots & -0.003 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.011 & 0.006 & 0.004 & 0.011 & 0.005 & -0.003 & \dots & 0.012 \end{bmatrix}_{178\times 178}$$

The parameter matrix β is

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$$

which is estimated by **b**, which for this model has a value of

```
b <- matrix(model$coefficients)
```

$$\mathbf{b} = [64.4798191, 0.2341811, 0.0257442, 0.3211588]'$$

with dimension $(4, 1)$.

We can compute the different sum of squares terms as follows:

```
j <- matrix(1, n, n)
ssr <- drop(t(y) %*% (hat - j / n) %*% y)

i <- diag(n)
sse <- drop(t(y) %*% (i - hat) %*% y)

sst <- drop(t(y) %*% (i - j / n) %*% y)
```

$$SSR = (\hat{Y}_i - \bar{Y})^2 = b'X'Y - \frac{1}{n}Y'JY = Y'\left[H - \frac{J}{n}\right]Y = 6815.098591$$

$$SSE = (Y_i - \hat{Y}_i)^2 = e'e = (Y - Xb)'(Y - Xb) = Y'Y - b'X'Y = Y'(I - H)Y = 4832.5137686$$

$$SST = (Y_i - \bar{Y})^2 = Y'Y - \frac{1}{n}Y'JY = Y'\left[I - \frac{J}{n}\right]Y = 1.1647612 \times 10^4$$

Problem 4

Use R to complete the ANOVA table and verify your computation in Problem 3.

```
anova_results <- anova(model)
anova_results
```

```
## Analysis of Variance Table
##
## Response: y
##            Df Sum Sq Mean Sq  F value    Pr(>F)
## x1          1 6428.4  6428.4 231.4630 < 2.2e-16 ***
## x2          1  200.9   200.9   7.2325  0.007856 **
## x3          1  185.8   185.8   6.6897  0.010515 *
## Residuals 174 4832.5    27.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

SSR

We can verify SSR by summing the `x1`, `x2`, and `x3` rows of the `Sum Sq` column of our ANOVA results and asserting that the sum is equal to the value calculated in Problem 3.

```
ssr_anova <- sum(anova_results[c("x1", "x2", "x3"), "Sum Sq"])

print(paste("SSR from ANOVA:", ssr_anova))
```

```
## [1] "SSR from ANOVA: 6815.09859096319"
```

```
print(paste("SSR from matrix algebra:", ssr))
```

```
## [1] "SSR from matrix algebra: 6815.09859096366"
```

```
stopifnot(all.equal(drop(ssr), ssr_anova))
```

SSE

We can verify SSE by checking the `Residuals` row of the `Sum Sq` column:

```
sse_anova <- anova_results["Residuals", "Sum Sq"]

print(paste("SSE from ANOVA:", sse_anova))
```

```
## [1] "SSE from ANOVA: 4832.51376858737"
```

```
print(paste("SSE from matrix algebra:", sse))
```

```
## [1] "SSE from matrix algebra: 4832.51376858694"
```

```
stopifnot(all.equal(drop(sse), sse_anova))
```

SST

We can verify SST by summing the entire `Sum Sq` column:

```r
sst_anova <- sum(anova_results[, "Sum Sq"])

print(paste("SSE from ANOVA:", sst_anova))
```

```
## [1] "SSE from ANOVA: 11647.6123595506"
```

```r
print(paste("SSE from matrix algebra:", sst))
```

```
## [1] "SSE from matrix algebra: 11647.6123595506"
```

```r
stopifnot(all.equal(drop(sst), sst_anova))
```

Problem 5

Use R to compute the variance-covariance matrix of the residuals $\Sigma[\mathbf{e}]$, where $\Sigma[\mathbf{e}] = \sigma^2(I - H)$, and σ^2 can be estimated by MSE. Note that this reflects the actual variance-covariance of the residual. The less assumption violation there is, the closer this matrix is to the variance-covariance matrix of the random error.

```r
mse <- anova_results["Residuals", "Mean Sq"]
cov <- mse * (i - hat)
```

$$\Sigma[\mathbf{e}] = \begin{bmatrix} 27.45 & -0.093 & -0.205 & -0.322 & -0.069 & -0.049 & \dots & -0.3 \\ -0.093 & 26.999 & -0.238 & -0.143 & 0.292 & 0.113 & \dots & -0.156 \\ -0.205 & -0.238 & 27.434 & -0.216 & 0.138 & -0.26 & \dots & -0.121 \\ -0.322 & -0.143 & -0.216 & 27.448 & -0.025 & -0.027 & \dots & -0.302 \\ -0.069 & 0.292 & 0.138 & -0.025 & 26.951 & -0.231 & \dots & -0.143 \\ -0.049 & 0.113 & -0.26 & -0.027 & -0.231 & 27.154 & \dots & 0.072 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -0.3 & -0.156 & -0.121 & -0.302 & -0.143 & 0.072 & \dots & 27.438 \end{bmatrix}_{178 \times 178}$$

5.a

What is the dimension of the matrix?

The dimension is $(178, 178)$.

5.b

Under assumption of independence and constant variance, what is the variance-covariance matrix of the random error $\Sigma[\varepsilon]$?

If the variables are independent, then their covariances would all be zero, so all of the off-diagonal elements of the matrix would be zero, and the diagonal elements would all be equal to the constant variance value.

Problem 6

Use R to compute the variance-covariance matrix of the parameters $\Sigma[\mathbf{b}]$, where $\Sigma[\mathbf{b}] = \sigma^2(X'X)^-1$, and σ^2 can be estimated by MSE.

```r
cov_b <- mse * solve(t(x) %*% x)
```

$$\Sigma[\mathbf{b}] = \begin{bmatrix} 0.367 & -0.007 & -0.001 & -0.005 \\ -0.007 & 0.002 & 0 & -0.002 \\ -0.001 & 0 & 0 & -0.001 \\ -0.005 & -0.002 & -0.001 & 0.015 \end{bmatrix}$$

6.a

What is the dimension of the matrix?

The dimension is $(4, 4)$

6.b

What is the standard error of b_1, b_2, and b_3?

We can find the standard error by simply taking the square root of our covariance matrix and observing the 3 last diagonal values.

```
std_err_b123 <- diag(cov_b^0.5)[c("x1", "x2", "x3")]
std_err_b123
```

```
##         x1         x2         x3
## 0.03936016 0.01466801 0.12417016
```

We can check these against our model summary's values and assert that they are equal

```
model_summary <- summary(model)
std_err_b123_ms <- model_summary$coefficients[c("x1", "x2", "x3"), "Std. Error"]
std_err_b123_ms
```

```
##         x1         x2         x3
## 0.03936016 0.01466801 0.12417016
```

```
stopifnot(all.equal(std_err_b123, std_err_b123_ms))
```

6.c

What is the 95% individual confidence interval for β_1, β_2, and β_3, respectively?

First, calculate the critical t-vallue using $n-4$ degrees of freedom now instead of $n-2$ since we have two more independent variables than in SLR:

```
t_crit <- qt(1 - alpha / 2, n - p)
```

The critical value is the same for all parameters, and we multiply it against the standard error of each parameter to get the margin of error for each interval. Therefore, the intervals are:

```
b1 <- model$coefficients["x1"]
b2 <- model$coefficients["x2"]
b3 <- model$coefficients["x3"]
ci_b1 <- c(
  b1 - t_crit * std_err_b123["x1"], b1 + t_crit * std_err_b123["x1"]
)
ci_b2 <- c(
  b2 - t_crit * std_err_b123["x2"], b2 + t_crit * std_err_b123["x2"]
)
ci_b3 <- c(
  b3 - t_crit * std_err_b123["x3"], b3 + t_crit * std_err_b123["x3"]
)
```

$$b_1 \pm t(1-\alpha/2; n-p)s[b_1] = 0.2341811 \pm t(1-0.05/2; 178-4)0.0393602 = (\mathbf{0.1564963}, \mathbf{0.3118659})$$
$$b_2 \pm t(1-\alpha/2; n-p)s[b_2] = 0.0257442 \pm t(1-0.05/2; 178-4)0.014668 = (-\mathbf{0.0032059}, \mathbf{0.0546944})$$
$$b_3 \pm t(1-\alpha/2; n-p)s[b_3] = 0.3211588 \pm t(1-0.05/2; 178-4)0.1241702 = (\mathbf{0.0760853}, \mathbf{0.5662324})$$

We can confirm these values using `confint()`:

```r
confint(model)
```

```
##                    2.5 %      97.5 %
## (Intercept) 63.284537614 65.67510062
## x1           0.156496305  0.31186592
## x2          -0.003205879  0.05469436
## x3           0.076085264  0.56623243
```

Problem 7

In order to perform a global F test for the significance of the MLR model, the critical value can be denoted by

$$F(1-\alpha, p-1, n-p)$$

and has a value of

```r
f_crit <- qf(1 - alpha, p - 1, n - p)
```

$$F(1-0.05, 4-1, 178-4) = 2.6565324$$

Calculating F^* for the model:

```r
msr <- ssr / (p - 1)
f_star <- msr / mse
```

$$F^* = \frac{MSR}{MSE} = 81.7950527 \gg 2.6565324$$

Based on the R output, since F^* is **greater than** than the critical value, the model is **significant**.

This agrees with the F and p values at the bottom of the model summary:

```r
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7701  -3.0922  -0.1913   3.8118  10.9957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 64.47982    0.60561 106.471  < 2e-16 ***
## x1           0.23418    0.03936   5.950 1.44e-08 ***
## x2           0.02574    0.01467   1.755   0.0810 .
## x3           0.32116    0.12417   2.586   0.0105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.27 on 174 degrees of freedom
## Multiple R-squared:  0.5851, Adjusted R-squared:  0.578
## F-statistic:  81.8 on 3 and 174 DF,  p-value: < 2.2e-16
```

Problem 8

At the confidence level of 85%...

```
alpha <- 0.15
```

8.a

To calculate the Bonferroni joint confidence intervals for parameters β_1, β_2, and β_3, the critical value is denoted by

$$B = t\left(1 - \frac{\alpha}{2g}; n - p\right)$$

and has a value of

```
g <- 3
crit_bonf_params <- qt(1 - alpha / (2 * g), n - p)
```

$$t\left(1 - \frac{0.15}{2 \cdot 3}; 178 - 4\right) = 1.9736914$$

8.b

Calculate the Working-Hotelling joint confidence intervals for parameters β_1, β_2, and β_3, the critical value is denoted by

$$W = \sqrt{pF(1 - \alpha; p, n - p)}$$

and has a value of

```
w <- sqrt(p * qf(1 - alpha, p, n - p))
```

$$\sqrt{4F(1 - 0.15; 4, 178 - 4)} = 2.6148609$$

8.c

Which joint confidence interval is better? Briefly explain.

As stated in 1.e, for joint estimation of the *parameters*, the Bonferroni interval is better when $n - p \geq 2$ because it yields a smaller critical value: $B = 1.9736914 < 2.6148609 = W$. This is consistent across all models when we are estimating the parameters/weights/coefficients simultaneously for a linear model. That is, $W/B > 1$.

Problem 9

Use R to compute the MLR model $Y \sim X_1 + X_2 + X_3$, model summary, and ANOVA table. Then compute the remaining question by hand.

```
model_summary
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7701  -3.0922  -0.1913   3.8118  10.9957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 64.47982    0.60561 106.471  < 2e-16 ***
## x1           0.23418    0.03936   5.950 1.44e-08 ***
## x2           0.02574    0.01467   1.755   0.0810 .
## x3           0.32116    0.12417   2.586   0.0105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.27 on 174 degrees of freedom
## Multiple R-squared:  0.5851, Adjusted R-squared:  0.578
## F-statistic:  81.8 on 3 and 174 DF,  p-value: < 2.2e-16
```

```r
anova_results
```

```
## Analysis of Variance Table
##
## Response: y
##            Df Sum Sq Mean Sq  F value    Pr(>F)
## x1          1 6428.4  6428.4 231.4630 < 2.2e-16 ***
## x2          1  200.9   200.9   7.2325  0.007856 **
## x3          1  185.8   185.8   6.6897  0.010515 *
## Residuals 174 4832.5    27.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

9.a

The 85% simultaneous confidence interval on β_1, β_2, and β_3. Use both Bonferroni and WH method.

Bonferroni

```r
ci_b1_joint <- c(
  b1 - crit_bonf_params * std_err_b123["x1"],
  b1 + crit_bonf_params * std_err_b123["x1"]
)

ci_b2_joint <- c(
  b2 - crit_bonf_params * std_err_b123["x2"],
  b2 + crit_bonf_params * std_err_b123["x2"]
)

ci_b3_joint <- c(
  b3 - crit_bonf_params * std_err_b123["x3"],
  b3 + crit_bonf_params * std_err_b123["x3"]
)
```

$$b_1 \pm t\left(1-\frac{\alpha}{2g}; n-p\right) s[b_1] = 0.2341811 \pm t\left(1-\frac{0.15}{2\cdot 3}; 178-4\right) 0.0393602 = (\mathbf{0.1564963}, \mathbf{0.3118659})$$
$$b_2 \pm t\left(1-\frac{\alpha}{2g}; n-p\right) s[b_2] = 0.0257442 \pm t\left(1-\frac{0.15}{2\cdot 3}; 178-4\right) 0.014668 = (-\mathbf{0.0032059}, \mathbf{0.0546944})$$
$$b_3 \pm t\left(1-\frac{\alpha}{2g}; n-p\right) s[b_3] = 0.3211588 \pm t\left(1-\frac{0.15}{2\cdot 3}; 178-4\right) 0.1241702 = (\mathbf{0.0760853}, \mathbf{0.5662324})$$

Working-Hotelling

```
ci_b1_joint <- c(
  b1 - w * std_err_b123["x1"],
  b1 + w * std_err_b123["x1"]
)

ci_b2_joint <- c(
  b2 - w * std_err_b123["x2"],
  b2 + w * std_err_b123["x2"]
)

ci_b3_joint <- c(
  b3 - w * std_err_b123["x3"],
  b3 + w * std_err_b123["x3"]
)
```

$$b_1 \pm \sqrt{pF(1-\alpha; p, n-p)} s[b_1] = 0.2341811 \pm \sqrt{4F(1-0.15; 4, 178-4)} 0.0393602 = (\mathbf{0.1312598}, \mathbf{0.3371025})$$
$$b_2 \pm \sqrt{pF(1-\alpha; p, n-p)} s[b_2] = 0.0257442 \pm \sqrt{4F(1-0.15; 4, 178-4)} 0.014668 = (-\mathbf{0.0126106}, \mathbf{0.064099})$$
$$b_3 \pm \sqrt{pF(1-\alpha; p, n-p)} s[b_3] = 0.3211588 \pm \sqrt{4F(1-0.15; 4, 178-4)} 0.1241702 = (-\mathbf{0.0035289}, \mathbf{0.6458465})$$

9.b

Based on the results in part (a), do you think all three predictors have significant impact on Y?

If all three predictors have a significant impact on Y, we would expect to see all three predictors have nonzero values. If zero falls within in any of the intervals for the parameters, then we cannot say that all three predictors have significant impact on Y.

Since the interval for β_2 overlaps with 0 in both intervals calculated above, which indicates no impact, we cannot say that all three predictors have a simultaneously/jointly significant impact on Y at a confidence level of 85%. *Some* parameters may have an impact, but we just cannot say that *all three* have a *joint* impact.

9.c

The 95% confidence interval for the mean response prediction when X_1, X_2, and X_3 each takes the median value in the sample.

```
alpha <- 0.05
x_h_df <- data.frame(t(apply(x, 2, median)))
x_h <- matrix(apply(x, 2, median))
y_h <- predict(model, newdata = x_h_df)

std_err_y_h <- sqrt(t(x_h) %*% vcov(model) %*% x_h)
```

```r
t_crit <- qt(1 - alpha / 2, n - p)

ci_y_h <- c(y_h - t_crit * std_err_y_h, y_h + t_crit * std_err_y_h)
```

The 95% confidence interval for the prediction of the mean response $E[Y_h]$ when each X component is the median of the component is:

$$\hat{Y}_h \pm t(1-\alpha/2; n-p)s[\hat{Y}_h] = 69.7799818 \pm t(1-0.05/2; 178-4)0.4338523 = (68.9236912, 70.6362724)$$

We can confirm this against the output from `ci.reg()`:

```r
ci.reg(model, newdata = x_h_df)
```

```
##   X.Intercept.     x1     x2   x3      Fit Lower.Band Upper.Band
## 1            1 15.715 33.985 2.32 69.77998   68.92369   70.63627
```