Stat 420 - Homework 2

Fawad Khan

Fall 2025

Exercise 1 (Using LM)

a

```r
cat_model <- lm(Hwt ~ Bwt, data = cats)
summary(cat_model)
```

```
##
## Call:
## lm(formula = Hwt ~ Bwt, data = cats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5694 -0.9634 -0.0921  1.0426  5.1238
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3567     0.6923  -0.515    0.607
## Bwt           4.0341     0.2503  16.119   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.452 on 142 degrees of freedom
## Multiple R-squared:  0.6466, Adjusted R-squared:  0.6441
## F-statistic: 259.8 on 1 and 142 DF,  p-value: < 2.2e-16
```

b

```r
coef(cat_model)
```

```
## (Intercept)         Bwt
##  -0.3566624   4.0340627
```

Beta 0 hat is not very useful in the real world since this is telling us the mean heart weight when body weight is 0 kg. A cat will not weigh 0 kg in real life. Beta 1 hat is telling us that for every 1 kg increase in body weight, the estimated mean heart weight increases by 4.0340627 grams.

c

```r
predict(cat_model, newdata = data.frame(Bwt = 3.1))
```

```
##        1
## 12.14893
```

```r
range(cats$Bwt)
```

```
## [1] 2.0 3.9
```

The estimated heart weight of a cat that weights 3.1 kg is 12.14893 grams. We feel confidant in this prediction as it lies within the range of observed body weight from 2.0 - 3.9. This is considered interpolation.

d

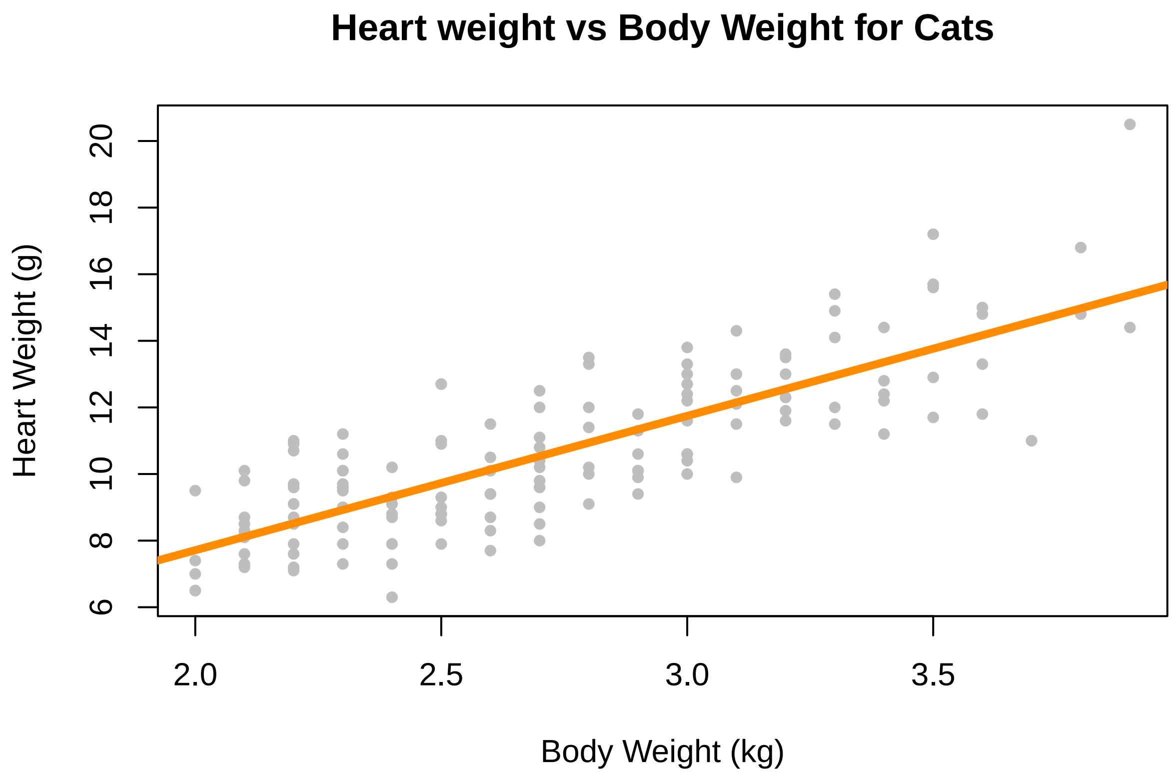
```r
predict(cat_model, newdata = data.frame(Bwt = 1.5))
```

```
##        1
## 5.694432
```

The estimated heart weight of a cat that weights 1.5 kg is 5.694432 grams. We do NOT feel confidant in this prediction as it lies outside the range of observed body weight from 2.0 - 3.9. This is considered extrapolation.

e

```r
plot(Hwt ~ Bwt, data = cats,
     xlab = 'Body Weight (kg)',
     ylab = 'Heart Weight (g)',
     main = 'Heart weight vs Body Weight for Cats',
     pch = 20,
     col = 'grey')
abline(cat_model, lwd = 4, col = 'darkorange')
```



f

```r
summary(cat_model)$r.squared
```

```
## [1] 0.6466209
```

Exercise 2 (Simulating SLR)

```r
birthday = 19970614
set.seed(birthday)
```

a

```r
n <- 25
x = runif(n = 25, 0, 10)
sigma <- sqrt(10.24)

sim_slr = function(x, beta_0 = 10, beta_1 = 5, sigma = 1) {
  n = length(x)
  epsilon = rnorm(n, mean = 0, sd = sigma)
  y = beta_0 + beta_1 * x + epsilon
  data.frame(predictor = x, response = y)
}
```

```r
sim_data <- sim_slr(x = x, beta_0 = 5, beta_1 = -3, sigma = sigma)
sim_data
```

```
##    predictor    response
## 1  6.8766826 -12.7590997
## 2  9.9178137 -25.9904373
## 3  1.0819739   3.7221847
## 4  3.1275773  -7.0070102
## 5  5.4072787 -11.6300011
## 6  0.5304602   4.4806159
## 7  9.6477942 -18.5903701
## 8  5.5197700 -12.5975133
## 9  6.0173403 -11.4553947
## 10 8.2005218 -15.0388991
## 11 1.3795505  -0.3066208
## 12 2.9103220  -0.8573657
## 13 5.8895766  -8.0588520
## 14 5.4827536 -12.8953812
## 15 2.8702116  -3.3413296
## 16 0.2625114   2.9356192
## 17 6.2649121 -13.6159879
## 18 7.7910873 -19.2570328
## 19 3.0619635  -5.7746593
## 20 7.1537374 -19.7035927
## 21 2.6672684  -5.3725165
## 22 5.6563735 -20.1041936
## 23 5.6123512  -9.9392838
## 24 3.6861575  -6.8922720
## 25 1.7610144   4.9054820
```

b

```r
sim_fit = lm(response ~ predictor, data = sim_data)
coef(sim_fit)
```

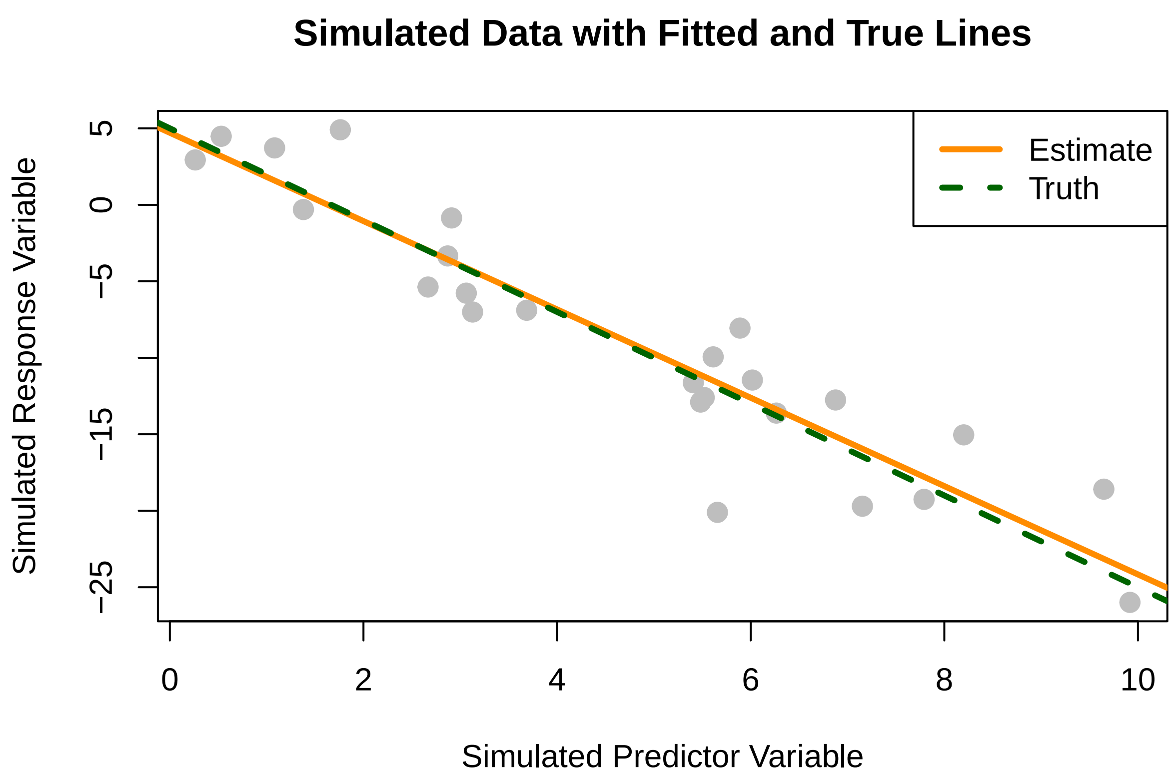
```
## (Intercept)   predictor
##    4.712927   -2.887487
```

We should be expecting values close to 5 (beta 0) and -3 (beta 1), but not exactly since we are factoring in noise. Based on our 25 data points, our results are fairly close to the true parameters.

c

```r
plot(sim_data$predictor, sim_data$response,
     main = "Simulated Data with Fitted and True Lines",
     xlab = "Simulated Predictor Variable",
     ylab = "Simulated Response Variable",
     cex = 2,
     pch = 20, col = "grey")

abline(sim_fit, col = "darkorange", lwd = 3)
```

```r
abline(a = 5, b = -3, col = "darkgreen", lwd = 3, lty = 2)
legend("topright", c("Estimate", "Truth"), lty = c(1, 2), lwd = 3,
       col = c("darkorange", "darkgreen"))
```



d

```r
num_samples <- 1000
beta_0 <- 5
beta_1 <- -3

beta_hat_1 <- rep(0, num_samples)

for (i in 1:num_samples) {
  eps <- rnorm(25, mean = 0, sd = sigma)
  y <- beta_0 + beta_1 * x + eps

  sim_model <- lm(y ~ x)
  beta_hat_1[i] <- coef(sim_model)[2]
}
```

e

```r
mean(beta_hat_1)
```

```
## [1] -2.991287
```

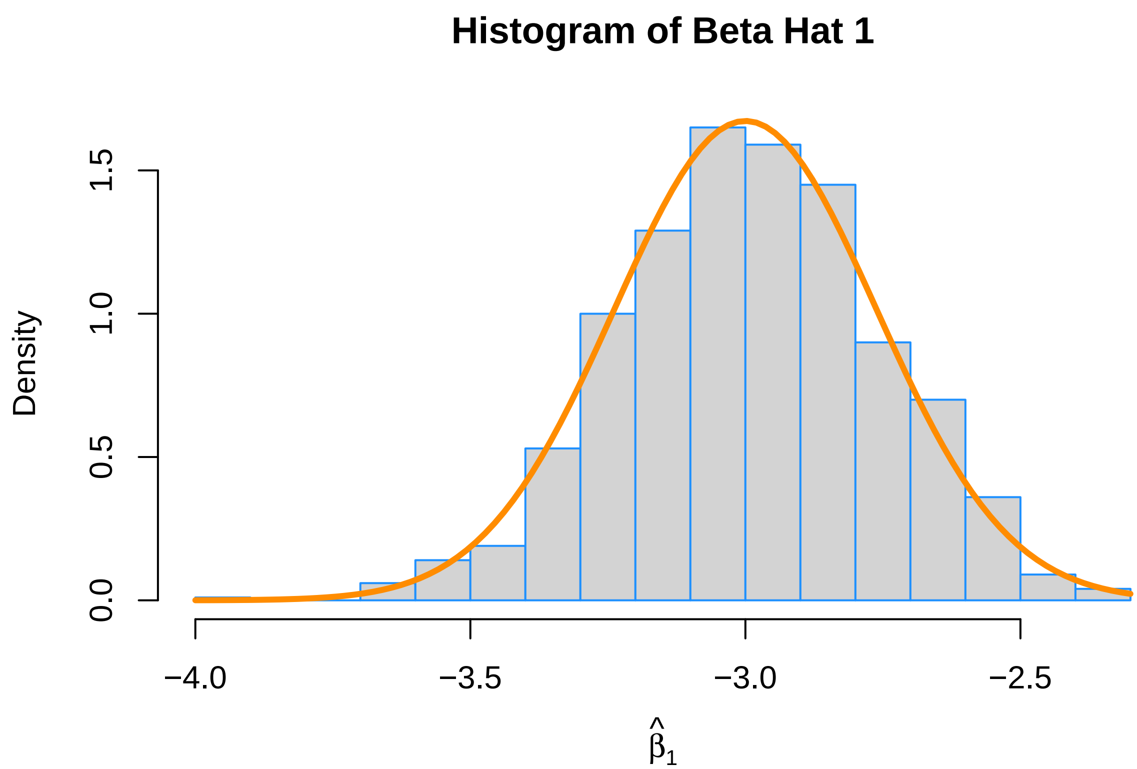
```r
sd(beta_hat_1)
```

```
## [1] 0.2385121
```

The mean of beta hat 1, should be close to the true mean of -3, which it is. The standard deviation measures how much this value varies from sample to sample due to noise.

f

```r
var_beta_1_hat <- var(beta_hat_1)

hist(beta_hat_1, prob = TRUE, breaks = 20,
     xlab = expression(hat(beta)[1]),
     main = "Histogram of Beta Hat 1",
     col = "lightgray", border = "dodgerblue")

curve(dnorm(x, mean = beta_1, sd = sqrt(var_beta_1_hat)),
      col = "darkorange", lwd = 3, add = TRUE)
```



The distribution of beta hat 1 follows a normal distribution, and is symmetrical and bell-shaped. The mean is around the true mean of -3 and the standard deviation is in line with what we found earlier.

Exercise 3 (Using LM for Inference)

a

```r
cat_model <- lm(Hwt ~ Bwt, data = cats)
summary_cat <- summary(cat_model)

t_value <- summary_cat$coefficients["Bwt", "t value"]
p_value <- summary_cat$coefficients["Bwt", "Pr(>|t|)"]

t_value
```

```
## [1] 16.11939
```

```r
p_value
```

```
## [1] 6.969045e-34
```

Null Hypothesis: Body weight (kg) has no effect in the heart weight (g) in cats.

Alternative Hypothesis: Body weight (kg) does have an effect in the heart weight (g) of cats.

Test Statistic: 16.11939

P-Value: 6.969045e-34

Decision & Conclusion: At alpha = 0.05, we will make the statistical decision to reject the null hypothesis. This is because our p-value is much smaller than 0.05. There is therefore significant evidence at that level that body weight is significantly associated with heart weight in cats.

b

```r
confint(cat_model, level = 0.95)
```

```
##                 2.5 %   97.5 %
## (Intercept) -1.725163 1.011838
## Bwt          3.539343 4.528782
```

We are 95% confidant that for every additional kg of body weight, the mean heart weight increases between 3.539343 and 4.528782 grams.

c

```r
confint(cat_model, '(Intercept)', level = 0.95)
```

```
##                 2.5 %   97.5 %
## (Intercept) -1.725163 1.011838
```

We are 95% confidant that the true heart weight of a cat with a body weight of 0 kg lies between -1.725163 and 1.011838 grams. In the real world we know the weight can never be negative.

d

```r
new_data <- data.frame(Bwt = c(2.1, 2.8))
predictions <- predict(cat_model, newdata = new_data, interval = "confidence", level = 0.95)
predictions
```

```
##         fit       lwr       upr
## 1  8.114869  7.724455  8.505284
## 2 10.938713 10.696491 11.180935
```

Estimated mean heart weight of cat with a body weight of 2.1 is between 7.724455 and 8.505284 grams.

Estimated mean heart weight of cat with a body weight of 2.8 is between 10.696491 and 11.180935 grams.

The interval is larger for a prediction at 2.1 kg of body weight. This is because 2.1 is further from the true mean, whereas 2.8 is much closer to it. We would expect the confidence interval to be wider the further from the true mean, so this result is expected.

e

```r
new_data_pred <- data.frame(Bwt = c(2.8, 4.2))
predict(cat_model, newdata = new_data_pred, interval = "prediction", level = 0.95)
```

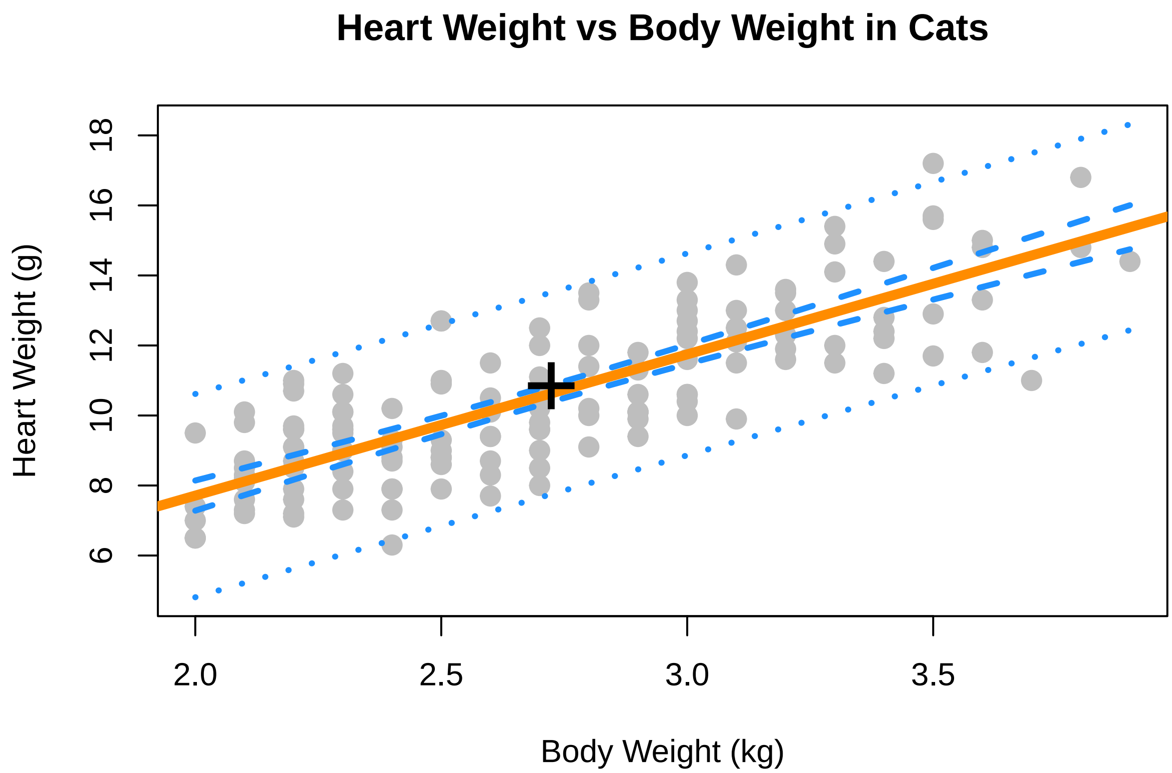
```
##        fit       lwr      upr
## 1 10.93871  8.057446 13.81998
## 2 16.58640 13.614238 19.55856
```

Estimated mean heart weight of cat with a body weight of 2.8 is between 8.057446 and 13.81998 grams.

Estimated mean heart weight of cat with a body weight of 4.2 is between 13.614238 and 19.55856 grams.

f

```r
bwt_grid <- seq(min(cats$Bwt), max(cats$Bwt), by = 0.01)

hwt_ci_band <- predict(cat_model,
                       newdata = data.frame(Bwt = bwt_grid),
                       interval = "confidence",
                       level = 0.95)

hwt_pi_band <- predict(cat_model,
                       newdata = data.frame(Bwt = bwt_grid),
                       interval = "prediction",
                       level = 0.95)

plot(Hwt ~ Bwt, data = cats,
     xlab = "Body Weight (kg)",
     ylab = "Heart Weight (g)",
     main = "Heart Weight vs Body Weight in Cats",
     pch  = 20,
     cex  = 2,
     col  = "grey",
     ylim = c(min(hwt_pi_band), max(hwt_pi_band)))

abline(cat_model, lwd = 5, col = "darkorange")

lines(bwt_grid, hwt_ci_band[, "lwr"], col = "dodgerblue", lwd = 3, lty = 2)
lines(bwt_grid, hwt_ci_band[, "upr"], col = "dodgerblue", lwd = 3, lty = 2)

lines(bwt_grid, hwt_pi_band[, "lwr"], col = "dodgerblue", lwd = 3, lty = 3)
lines(bwt_grid, hwt_pi_band[, "upr"], col = "dodgerblue", lwd = 3, lty = 3)

points(mean(cats$Bwt), mean(cats$Hwt), pch = "+", cex = 3)
```



g

The point of the confidence interval is to show where the mean response lies for a given predictor value. Most data points fall outside the confidence bands, as they vary more than the mean.

h

```
beta_hat <- summary(cat_model)$coefficients["Bwt", "Estimate"]
se_beta <- summary(cat_model)$coefficients["Bwt", "Std. Error"]
beta_0 <- 3.5
t_stat <- (beta_hat - beta_0) / se_beta
df <- df.residual(cat_model)
p_value <- 2 * pt(-abs(t_stat), df)
t_stat
```

```
## [1] 2.134019
```

```
p_value
```

```
## [1] 0.03455924
```

Test Statistic: 2.134019

P-Value: 0.03455924

Decision: At alpha = 0.05, we will reject the null hypothesis as the p-value is less than 0.05. There is evidence that the slope is significantly different from 3.5

Exercise 4 (More inference for LM)

```r
library(mlbench)
data(Ozone, package = "mlbench")
Ozone = Ozone[, c(4, 6, 7, 8)]
colnames(Ozone) = c("ozone", "wind", "humidity", "temp")
Ozone = Ozone[complete.cases(Ozone), ]
```

a

```r
ozone_wind_model <- lm(ozone ~ wind, data = Ozone)
summary_ozone <- summary(ozone_wind_model)
t_value <- summary_ozone$coefficients["wind", "t value"]
p_value <- summary_ozone$coefficients["wind", "Pr(>|t|)"]
t_value
```

```
## [1] -0.2189811
```

```r
p_value
```

```
## [1] 0.8267954
```

Null Hypothesis: Wind speed has no effect on ozone.

Alternative Hypothesis: Wind speed does have an effect on ozone.

Test Statistic: -0.2189811

P-Value: 0.8267954

Decision & Conclusion: Since the p-value is greater than alpha = 0.01, we fail to reject the null hypothesis. There is insufficient evidence at alpha = 0.01 to conclude that wind speed has an effect on ozone.

b

```r
ozone_temp_model <- lm(ozone ~ temp, data = Ozone)
summary_temp_model <- summary(ozone_temp_model)
t_value <- summary_temp_model$coefficients["temp", "t value"]
p_value <- summary_temp_model$coefficients["temp", "Pr(>|t|)"]
t_value
```

```
## [1] 22.84896
```

```r
p_value
```

```
## [1] 8.153764e-71
```

Null Hypothesis: Temperature has no effect on ozone.

Alternative Hypothesis: Temperature does have an effect on ozone.

Test Statistic: 22.84896

P-Value: 8.153764e-71

Decision & Conclusion: Since the p-value is smaller than alpha = 0.01, we will reject the null hypothesis. There is significant evidence at alpha = 0.01 that temperature does have an effect on ozone.

Exercise 5 (Simulating Confidence Intervals)

a

```
birthday <- 19970614
set.seed(birthday)

n <- 25
x <- seq(0, 2.5, length = n)

beta_0 <- 5
beta_1 <- 2
sigma <- 3
num_samples <- 2500

beta_hat_1 <- numeric(num_samples)
se <- numeric(num_samples)

for (i in 1:num_samples) {
  eps <- rnorm(n, mean = 0, sd = sigma)
  y <- beta_0 + beta_1 * x + eps

  sim_model <- lm(y ~ x)

  beta_hat_1[i] <- coef(sim_model)[2]
  se[i] <- summary(sim_model)$coefficients["x", "Std. Error"]
}
```

b

```
df <- n - 2
t_crit <- qt(0.975, df)
lower_95 <- beta_hat_1 - t_crit * se
upper_95 <- beta_hat_1 + t_crit * se
```

c

```
contains_true <- (lower_95 <= beta_1) & (upper_95 >= beta_1)
coverage <- mean(contains_true)
coverage
```

```
## [1] 0.9448
```

d

```
reject_null <- (lower_95 > 0) | (upper_95 < 0)
rejection_rate <- mean(reject_null)
rejection_rate
```

```
## [1] 0.6872
```

e

```r
t_crit_99 <- qt(0.995, df)
lower_99 <- beta_hat_1 - t_crit_99 * se
upper_99 <- beta_hat_1 + t_crit_99 * se
```

f

```r
contains_true_99 <- (lower_99 <= beta_1) & (upper_99 >= beta_1)
coverage_99 <- mean(contains_true_99)
coverage_99
```

```
## [1] 0.992
```

g

```r
reject_null_99 <- (lower_99 > 0) | (upper_99 < 0)
rejection_rate_99 <- mean(reject_null_99)
rejection_rate_99
```

```
## [1] 0.4072
```

Exercise 6 (Recreating LM())

a

```r
my_lm1 <- function(x, y) {
  x_bar <- mean(x)
  y_bar <- mean(y)

  SXX <- sum((x - x_bar)^2)
  SXY <- sum((x - x_bar) * (y - y_bar))

  b1 <- SXY / SXX
  b0 <- y_bar - b1 * x_bar

  coefficients <- c(b0, b1)
  names(coefficients) <- c("(Intercept)", "x")
  return(coefficients)
}

set.seed(2025)
x <- 1:10
y <- 2 + 3 * x + rnorm(10, 0, 1)

my_lm1(x, y)
```

```
## (Intercept)           x
##    2.670360    2.943289
```

b

```r
my_lm2 <- function(x, y) {
  X <- cbind(1, x)

  beta <- solve(t(X) %*% X) %*% t(X) %*% y

  coefficients <- as.vector(beta)
  names(coefficients) <- c("(Intercept)", "x")

  return(coefficients)
}

my_lm2(x, y)
```

```
## (Intercept)           x
##    2.670360    2.943289
```

Both answers are the same.