Stat 420 - Homework 3

Fawad Khan

```
nutrition <- read.csv("nutrition-2018.csv")
```

Fall 2025

Exercise 1 (using LM)

a

```
model <- lm(Calories ~ Fat + Sugar + Sodium, data = nutrition)
model_summary <- summary(model)
```

```
f_stat <- model_summary$fstatistic[1]
df1 <- model_summary$fstatistic[2]
df2 <- model_summary$fstatistic[3]
p_value <- pf(f_stat, df1, df2, lower.tail = FALSE)
```

```
f_stat
```

```
##   value
## 6590.94
```

```
p_value
```

```
## value
##     0
```

Null Hypothesis: Fat, sugar, and sodium have no effect on calories.

Alternative Hypothesis: Fat, sugar, and sodium do have an effect on calories.

Test Statistic: 6590.94

P-Value: $<$ 2.2e-16

Decision and Conclusion: Since the p-value is extremely low, at alpha $= 0.01$ we will reject the null hypothesis. There is significant evidence that fat, sugar and sodium are related to calories.

b

```
coef_estimates <- coef(model)
coef_estimates
```

```
##  (Intercept)          Fat        Sugar       Sodium
## 1.004561e+02 8.483289e+00 3.900517e+00 6.165246e-03
```

Beta Hat 0: When fat, sugar, and sodium are all zero, the model predicts about ~ 100.004 calories for a food item.

Beta hat 1: For every addition gram of fat, assuming sugar and sodium are kept constant, calories in a food item is expected to increase by ~ 8.483 calories.

Beta hat 2: For every addition gram of sugar, assuming fat and sodium are kept constant, calories in a food item is expected to increase by ~ 3.900 calories.

Beta Hat 3: For every addition milligram of sodium, assuming fat and sugar are kept constant, calories in a food item is expected to increase by ~ -.006165 calories. We know calories cannot be negative so this is less useful.

c

```
fish_fillet <- data.frame(Fat = 18, Sugar = 5, Sodium = 580)

predicted_calories <- predict(model, newdata = fish_fillet)
predicted_calories
```

```
##        1
## 276.2337
```

d

```
sy <- sd(nutrition$Calories)
sy
```

```
## [1] 168.05
```

```
se <- summary(model)$sigma
se
```

```
## [1] 80.8543
```

The standard deviation of calories is 168.05. This means that across all foods in the data-set, calories vary by about 168.05 from the mean.

The residual standard error is 80.8543. This means that after accounting for fat, sugar and sodium, the models predictions are off by about 80.8543 calories.

e

```
r_squared <- summary(model)$r.squared
r_squared
```

```
## [1] 0.7686281
```

Approximately 76.862% of the variation in calories can be explained by the amount of fat, sugar, and sodium in the food.

f

```r
confint(model, "Sugar", level = 0.90)
```

```
##            5 %     95 %
## Sugar 3.783051 4.017983
```

We are 90% confidant for that each additional gram of sugar in food, the amount of calories increases by between 3.783051 and 4.017983, assuming fat and sodium are constant.

g

```r
confint(model, "(Intercept)", level = 0.95)
```

```
##                2.5 %   97.5 %
## (Intercept) 97.69443 103.2177
```

We are 95% confidant that the expected amount of calories for a food item containing zero fat, zero sugar and zero sodium, is between 97.69443 and 103.2177 calories.

h

```r
fries <- data.frame(Fat = 15, Sugar = 0, Sodium = 260)

pred_conf_int <- predict(model, newdata = fries, interval = "confidence", level = 0.99)
pred_conf_int
```

```
##        fit      lwr     upr
## 1 229.3084 226.1657 232.451
```

For a medium order of McDonald's french fries containing 15 grams of fat, 0 grams of sugar, and 260 milligrams of sodium, we are 99% confidant that the mean calorie count lies between 226.1657 and 232.451 calories.

i

```r
taco <- data.frame(Fat = 11, Sugar = 2, Sodium = 340)

pred_pred_int <- predict(model, newdata = taco, interval = "prediction", level = 0.99)
pred_pred_int
```

```
##        fit       lwr      upr
## 1 203.6695 -4.684481 412.0234
```

For a crunchy taco supreme containing 11 grams of fat, 2 grams of sugar, and 340 milligrams of sodium, we are 99% confidant that the calories for a single crunchy taco supreme lies between -4.684481 and 412.0234 calories. We know that in reality calories cannot be negative so we can consider the range starting from zero calories.

Exercise 2 (More LM for Multiple Regression)

```r
goalies <- read.csv("goalies17.csv")
```

```r
goalies_clean <- subset(goalies, select = -c(Player, GS, L, TOL, SV_PCT, GAA))
```

```r
model1 <- lm(W ~ GA + SV, data = goalies_clean)

model2 <- lm(W ~ GA + SV + SA + MIN + SO, data = goalies_clean)

model3 <- lm(W ~ ., data = goalies_clean)
```

a

```r
anova(model1, model2)
```

```
## Analysis of Variance Table
##
## Model 1: W ~ GA + SV
## Model 2: W ~ GA + SV + SA + MIN + SO
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    500 310758
## 2    497  77763  3    232996 496.38 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Null Hypothesis: The additional variables in model 2: shots against, minutes and shutouts, do not significantly improve the model.

Test Statistic: 496.38

P-Value: 2.2e-16

Decision and Conclusion: The p-value is very small, therefore at alpha = 0.05 we will reject the null hypothesis. This means that the additional variables in model 2 do have an effect on the model.

Preferred Model: Based on our results, we prefer model 2.

b

```r
anova(model2, model3)
```

```
## Analysis of Variance Table
##
## Model 1: W ~ GA + SV + SA + MIN + SO
## Model 2: W ~ First + Last + Active + GP + GA + SA + SV + SO + PIM + MIN
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    497 77763
## 2    492 69133  5    8629.8 12.283 3.073e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Null Hypothesis: The additional variables in model 3: First, Last, Active, GP and PIM, do not significantly improve the model.

Test Statistic: 12.283

P-Value: 3.073e-11

Decision and Conclusion: The p-value is very small, therefore at alpha = 0.05 we will reject the null hypothesis. This means that the additional variables in model 3 do have an effect on the model.

Preferred Model: Based on our results, we prefer model 3.

c

```
coef_summary <- summary(model3)$coefficients
t_value <- coef_summary["SV", "t value"]
p_value <- coef_summary["SV", "Pr(>|t|)"]
t_value
```

```
## [1] -4.077014
```

```
p_value
```

```
## [1] 5.319041e-05
```

Test Statistic: -4.077014

P-Value: 5.319041e-05

Decision and Conclusion: The p-value is very small, therefore at alpha = 0.05 we will reject the null hypothesis. We can conclude that saves do have a significant effect on wins.

Exercise 3 (Regression without LM)

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

a

```
y <- Ozone$ozone

X <- as.matrix(cbind(Intercept = 1, Ozone[, c("wind", "humidity", "temp")]))

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

beta_hat_no_lm <- as.vector(beta_hat)

sum_beta_squared <- sum(beta_hat_no_lm^2)

beta_hat_no_lm
```

```
## [1] -16.38178539  -0.18594444   0.08340014   0.38984294
```

```
sum_beta_squared
```

```
## [1] 268.5564
```

b

```
model_lm <- lm(ozone ~ wind + humidity + temp, data = Ozone)

beta_hat_lm <- as.vector(coef(model_lm))

sum_beta_hat_lm_sq <- sum(beta_hat_lm^2)

beta_hat_lm
```

```
## [1] -16.38178539  -0.18594444   0.08340014   0.38984294
```

```
sum_beta_hat_lm_sq
```

```
## [1] 268.5564
```

c

```
all.equal(beta_hat_no_lm, unname(beta_hat_lm))
```

```
## [1] TRUE
```

d

```
n <- nrow(X)
p <- ncol(X)

y_hat <- X %*% beta_hat_no_lm
residuals <- y - y_hat

RSS <- sum(residuals^2)

sigma_squared <- RSS / (n - p)

var_beta_hat <- sigma_squared * solve(t(X) %*% X)

std_errors_no_lm <- sqrt(diag(var_beta_hat))

std_errors_no_lm
```

```
##  Intercept       wind   humidity       temp
## 1.32204661 0.12567040 0.01437271 0.01925454
```

```r
model_lm <- lm(ozone ~ wind + humidity + temp, data = Ozone)

std_errors_lm <- summary(model_lm)$coefficients[, "Std. Error"]

all.equal(unname(std_errors_no_lm), unname(std_errors_lm))
```

```
## [1] TRUE
```

e

```r
y_mean <- mean(y)
SS_tot <- sum((y - y_mean)^2)

SS_res <- sum((y - (X %*% beta_hat_no_lm))^2)

R_squared_no_lm <- 1 - (SS_res / SS_tot)

R_squared_no_lm
```

```
## [1] 0.6398887
```

```r
model_lm <- lm(ozone ~ wind + humidity + temp, data = Ozone)

R_squared_lm <- summary(model_lm)$r.squared

all.equal(R_squared_no_lm, R_squared_lm)
```

```
## [1] TRUE
```

Exercise 4 (F Test for nested models vs single t-test)

a

```r
summary_model <- summary(model)

coeff_table <- summary_model$coefficients

t_value_sodium <- coeff_table["Sodium", "t value"]
p_value_sodium <- coeff_table["Sodium", "Pr(>|t|)"]

t_value_sodium
```

```
## [1] 5.983251
```

```r
p_value_sodium
```

```
## [1] 2.314599e-09
```

b

```r
reduced_model <- lm(Calories ~ Fat + Sugar, data = nutrition)
anova_result <- anova(reduced_model, model)
```

```r
F_value <- anova_result$F[2]
df1 <- anova_result$Df[2]
df2 <- anova_result$Res.Df[2]
p_value <- anova_result$`Pr(>F)`[2]
F_value
```

```
## [1] 35.79929
```

```r
df1
```

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

```r
df2
```

```
## [1] 5952
```

```r
p_value
```

```
## [1] 2.314599e-09
```

Test Statistic: 35.79929

Distribution of the test statistic: Between 1 numerator degrees of freedom and 5952 denominator degrees of freedom.

P-Value: 2.314599e-09

c

```r
model_reduced <- lm(W ~ GA + SV + SA + MIN, data = goalies_clean)
model_full <- lm(W ~ GA + SV + SA + MIN + SO, data = goalies_clean)

anova_result <- anova(model_reduced, model_full)
anova_result
```

```
## Analysis of Variance Table
##
## Model 1: W ~ GA + SV + SA + MIN
## Model 2: W ~ GA + SV + SA + MIN + SO
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1    498 78435
## 2    497 77763  1    671.84 4.2939 0.03876 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
F_value <- anova_result$F[2]
p_value_F <- anova_result$`Pr(>F)`[2]
F_value
```

```
## [1] 4.29388
```

```
p_value_F
```

```
## [1] 0.03876487
```

```
summary_full <- summary(model_full)
t_value <- summary_full$coefficients["SO", "t value"]
p_value_t <- summary_full$coefficients["SO", "Pr(>|t|)"]
t_value
```

```
## [1] 2.072168
```

```
p_value_t
```

```
## [1] 0.03876487
```

```
all.equal(F_value, t_value^2)
```

```
## [1] TRUE
```

```
all.equal(p_value_F, p_value_t)
```

```
## [1] TRUE
```

Exercise 5 (Simulating Multiple Regression)

a

```
set.seed(400)
sample_size = 40
```

```
x0 <- rep(1, sample_size)
x1 <- rnorm(sample_size, mean = 0, sd = 2)
x2 <- runif(sample_size, min = 0, max = 4)
x3 <- rnorm(sample_size, mean = 0, sd = 1)
x4 <- runif(sample_size, min = -2, max = 2)
x5 <- rnorm(sample_size, mean = 0, sd = 2)

X <- cbind(x0, x1, x2, x3, x4, x5)
C <- solve(t(X) %*% X)

y <- rep(0, sample_size)

sim_data <- data.frame(y = y, x1 = x1, x2 = x2, x3 = x3, x4 = x4, x5 = x5)
```

```
sum_diag_C <- sum(diag(C))
sum_diag_C
```

```
## [1] 0.1763287
```

```r
sim_data[5, ]
```

```
##   y        x1       x2          x3       x4         x5
## 5 0 -1.203677 3.394114 -0.09208766 1.743326 -0.7808329
```

b

```r
beta_hat_1 <- numeric(2500)
beta_3_pval <- numeric(2500)
beta_5_pval <- numeric(2500)
```

c

```r
beta_0 <- 2
beta_1 <- -0.75
beta_2 <- 1.6
beta_3 <- 0
beta_4 <- 0
beta_5 <- 2
sigma <- 5

for (i in 1:2500) {
  epsilon <- rnorm(sample_size, mean = 0, sd = sigma)
  y_new <- beta_0 +
           beta_1 * sim_data$x1 +
           beta_2 * sim_data$x2 +
           beta_3 * sim_data$x3 +
           beta_4 * sim_data$x4 +
           beta_5 * sim_data$x5 +
           epsilon

  sim_data$y <- y_new

  model <- lm(y ~ x1 + x2 + x3 + x4 + x5, data = sim_data)

  beta_hat_1[i] <- coef(model)["x1"]
  beta_3_pval[i] <- summary(model)$coefficients["x3", "Pr(>|t|)"]
  beta_5_pval[i] <- summary(model)$coefficients["x5", "Pr(>|t|)"]
}
```

d

```r
mean_beta1 <- -0.75
var_beta1 <- 25 * C[2, 2]
sd_beta1 <- sqrt(var_beta1)
var_beta1
```

```
## [1] 0.2230073
```

```r
sd_beta1
```

```
## [1] 0.4722365
```

Beta 1 hat is normally distributed with a mean of -0.75, a variation of 0.2230073 and a standard deviation of 0.4722365.

e

```
mean_beta_hat_1 <- mean(beta_hat_1)
var_beta_hat_1 <- var(beta_hat_1)
mean_beta1
```

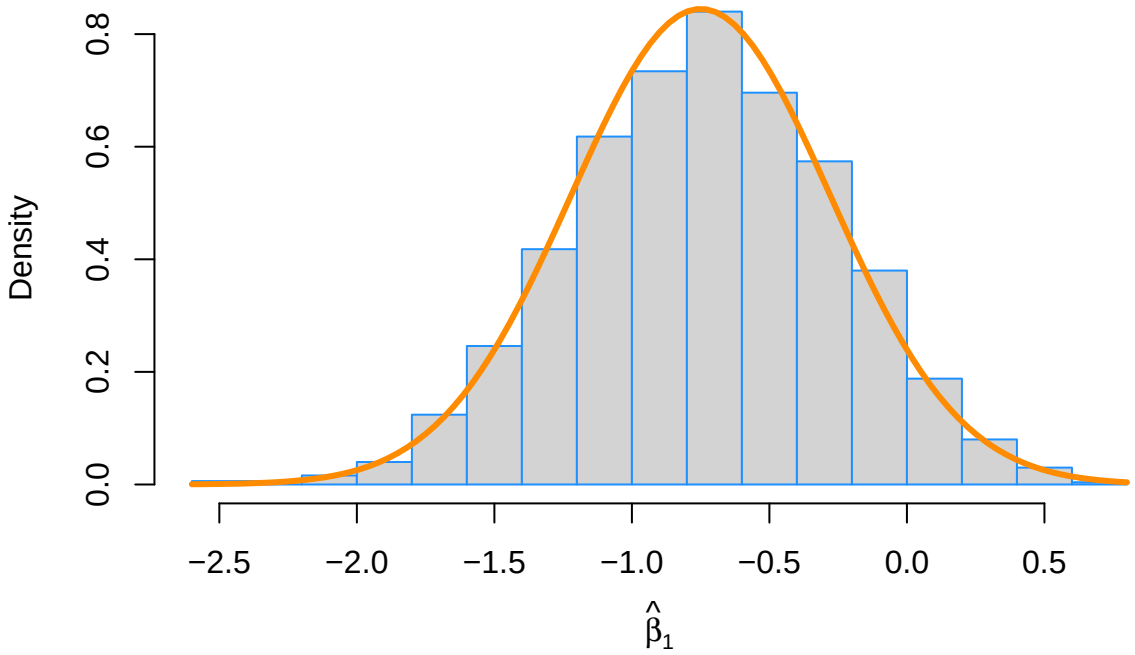
```
## [1] -0.75
```

```
var_beta_hat_1
```

```
## [1] 0.2352889
```

```
hist(beta_hat_1, prob = TRUE, breaks = 20,
     xlab = expression(hat(beta)[1]),
     main = "",
     border = "dodgerblue")

curve(dnorm(x, mean = beta_1, sd = sqrt(sigma^2 * C[2, 2])),
      col = "darkorange", add = TRUE, lwd = 3)
```



The curve does seem to match the histogram, showing a bell-shaped distribution with the center being at the true mean of -0.75. The mean and variance is also as expected.

f

```
prop_less_0.10 <- mean(beta_3_pval < 0.10)
prop_less_0.10
```

```
## [1] 0.1012
```

The proportion of p-values less than 0.10 is 0.1012. We are expecting the p-values to be uniformly distributed with about 10% falling underneath 0.10. This result is expected.

g

```
prop_less_0.01 <- mean(beta_5_pval < 0.01)
prop_less_0.01
```

```
## [1] 0.8564
```

The proportion of p-values less than 0.01 is 0.8564. We are expecting a high proportion of small p-values which indicates strong evidence against the null hypothesis. This result is expected.