Azreen Haque

3/27/2025

Solutions

Problem 1

Solutions below.

Part A

```
data <- read.csv("hw06pr01.csv", header = TRUE, sep = ",")
names(data)
```

```
## [1] "brand"     "moisture"  "sweetness" "calories"
```

```
model <- lm(brand ~ moisture + sweetness + calories, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = brand ~ moisture + sweetness + calories, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2893 -2.9927 -0.7011  3.1987  6.5334
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.29067    9.33999   3.029 0.010486 *
## moisture     4.61884    0.46457   9.942 3.81e-07 ***
## sweetness    4.54571    1.01140   4.494 0.000734 ***
## calories     0.02687    0.06622   0.406 0.692047
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.042 on 12 degrees of freedom
## Multiple R-squared:  0.9137, Adjusted R-squared:  0.8921
## F-statistic: 42.33 on 3 and 12 DF,  p-value: 1.169e-06
```

```
cat("Fitted Equation:\n")
```

```
## Fitted Equation:
```

```
cat("Y_hat = ", coef(model)[1],
    " + ", coef(model)[2], " * X1 (moisture)",
    " + ", coef(model)[3], " * X2 (sweetness)",
    " + ", coef(model)[4], " * X3 (calories)\n", sep = "")
```

```
## Y_hat = 28.29067 + 4.618836 * X1 (moisture) + 4.545706 * X2 (sweetness) + 0.02687018 * X3 (calories)
```

Part B

```
model <- lm(brand ~ moisture + sweetness + calories, data = data)
anova_model <- anova(model)
print(anova_model)
```

```
## Analysis of Variance Table
##
## Response: brand
##           Df  Sum Sq Mean Sq  F value    Pr(>F)
## moisture   1 1739.11 1739.11 106.4371 2.554e-07 ***
## sweetness  1  333.06  333.06  20.3841 0.0007081 ***
## calories   1    2.69    2.69   0.1647 0.6920472
## Residuals 12  196.07   16.34
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Extracting sum of squares from the anova table
SS_regression = sum(anova_model$"Sum Sq"[1:3])  # Sum the sum of squares for all predictors
SS_error = anova_model$"Sum Sq"[4]  # Error sum of squares
SS_total = SS_regression + SS_error  # Total sum of squares

# Degrees of freedom
df_regression = sum(anova_model$"Df"[1:3])
df_error = anova_model$"Df"[4]
df_total = df_regression + df_error

# Creating a new data frame for the combined ANOVA table
anova_combined <- data.frame(
  Source = c("Regression", "Error", "Total"),
  `Df` = c(df_regression, df_error, df_total),
  `Sum Sq` = c(SS_regression, SS_error, SS_total),
  `Mean Sq` = c(SS_regression/df_regression, SS_error/df_error, NA)
)

print(anova_combined)
```

```
##       Source Df    Sum.Sq   Mean.Sq
## 1 Regression  3 2074.8654 691.62179
## 2      Error 12  196.0721  16.33934
## 3      Total 15 2270.9375        NA
```

Part C

```
# Define the values for Mean Square Regression and Mean Square Error
MSR <- 691.62179
MSE <- 16.33934

# Calculate the F-statistic
F_calculated <- MSR / MSE
F_calculated
```

```
## [1] 42.32862
```

```
# Retrieve the critical F-value
df_regression <- 3
df_error <- 12
alpha <- 0.05

# Using R's qf function to get the critical F-value from the F-distribution
F_critical <- qf(1 - alpha, df1 = df_regression, df2 = df_error)
F_critical
```

```
## [1] 3.490295
```

```
# Compare the calculated F-statistic with the critical F-value to decide on the hypothesis
if (F_calculated > F_critical) {
  print("Reject the null hypothesis: There is a significant relationship between the variables.")
} else {
  print("Do not reject the null hypothesis: There is no significant relationship between the variables.
}
```

```
## [1] "Reject the null hypothesis: There is a significant relationship between the variables."
```

Part D

```
# Define the number of observations and predictors
n <- 16  # Total number of observations
p <- 3   # Number of predictors

# Extract the sum of squares from the ANOVA table
SSR <- 2074.8654  # Sum of Squares due to Regression
SSTO <- 2270.9375  # Total Sum of Squares
SSE <- 196.0721  # Sum of Squares due to Error

# Calculate the coefficient of multiple determination (R-squared)
R_squared <- SSR / SSTO

# Calculate the adjusted R-squared
R_squared_adjusted <- 1 - ((SSE / SSTO) * ((n - 1) / (n - p - 1)))

# Print the calculated values for verification
cat("Calculated R-squared: ", R_squared, "\n")
```

```
## Calculated R-squared:  0.9136603
```

```
cat("Calculated Adjusted R-squared: ", R_squared_adjusted, "\n")
```

```
## Calculated Adjusted R-squared:  0.8920754
```

```r
# Compare with summary output from the regression model
summary_stats <- summary(model)
cat("From summary(model): R-squared = ", summary_stats$r.squared, ", Adjusted R-squared = ", summary_sta
```

```
## From summary(model): R-squared =  0.9136603 , Adjusted R-squared =  0.8920753
```

Part E

```r
# Extract coefficients
coefficients <- summary_stats$coefficients

# Define significance level
alpha <- 0.05

# Degrees of freedom for t-test
df <- summary_stats$df[2]  # residual degrees of freedom from the model summary

# Critical t-value for two-tailed test at alpha = 0.05
t_critical <- qt(1 - alpha/2, df)

# Function to perform manual t-test
manual_t_test <- function(coef_estimate, coef_stderr, coef_name) {
  t_value <- coef_estimate / coef_stderr
  p_value <- 2 * pt(-abs(t_value), df)  # two-sided p-value

  # Hypotheses
  cat(sprintf("Testing %s:\n", coef_name))
  cat("H0: The coefficient is equal to 0 (no effect)\n")
  cat("Ha: The coefficient is not equal to 0 (has an effect)\n")

  # Test statistic
  cat(sprintf("t-value: %f\n", t_value))

  # Rejection rule
  cat(sprintf("Critical t-value at alpha = %f: +/- %f\n", alpha, t_critical))

  # Conclusion
  if (abs(t_value) > t_critical) {
    cat("Conclusion: Reject H0, there is a significant effect.\n\n")
  } else {
    cat("Conclusion: Do not reject H0, there is no significant effect.\n\n")
  }
}

# Apply the t-test for each predictor
sapply(2:nrow(coefficients), function(i) {
  manual_t_test(coefficients[i, "Estimate"], coefficients[i, "Std. Error"], rownames(coefficients)[i])
})
```

```
## Testing moisture:
## H0: The coefficient is equal to 0 (no effect)
```

```
## Ha: The coefficient is not equal to 0 (has an effect)
## t-value: 9.942279
## Critical t-value at alpha = 0.050000: +/- 2.178813
## Conclusion: Reject H0, there is a significant effect.
##
## Testing sweetness:
## H0: The coefficient is equal to 0 (no effect)
## Ha: The coefficient is not equal to 0 (has an effect)
## t-value: 4.494487
## Critical t-value at alpha = 0.050000: +/- 2.178813
## Conclusion: Reject H0, there is a significant effect.
##
## Testing calories:
## H0: The coefficient is equal to 0 (no effect)
## Ha: The coefficient is not equal to 0 (has an effect)
## t-value: 0.405779
## Critical t-value at alpha = 0.050000: +/- 2.178813
## Conclusion: Do not reject H0, there is no significant effect.

## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
```

Part F

```r
model <- lm(brand ~ moisture + sweetness + calories, data = data)

# Extract estimates, standard errors, and degrees of freedom
summary_model <- summary(model)
betas <- summary_model$coefficients[2:4, 1]         # 1, 2, 3
std_errors <- summary_model$coefficients[2:4, 2]    # SEs
df <- summary_model$df[2]                           # Degrees of freedom (n - p - 1)

# ---- Individual 95% Confidence Intervals ----
alpha <- 0.05
t_indiv <- qt(1 - alpha/2, df)

lower_indiv <- betas - t_indiv * std_errors
upper_indiv <- betas + t_indiv * std_errors

individual_CIs <- data.frame(
  Coefficient = c("moisture", "sweetness", "calories"),
  Lower = lower_indiv,
  Upper = upper_indiv,
  Width = upper_indiv - lower_indiv
)
```

```r
cat("Individual 95% Confidence Intervals:\n")
```

```
## Individual 95% Confidence Intervals:
```

```r
print(individual_CIs)
```

```
##           Coefficient      Lower     Upper     Width
## moisture     moisture  3.6066355 5.6310364 2.0244009
## sweetness   sweetness  2.3420636 6.7493487 4.4072851
## calories     calories -0.1174082 0.1711485 0.2885567
```

```r
# ---- Bonferroni-adjusted Joint 95% Confidence Intervals ----
m <- 3  # number of parameters
alpha_bonf <- alpha / m
t_bonf <- qt(1 - alpha_bonf/2, df)

lower_bonf <- betas - t_bonf * std_errors
upper_bonf <- betas + t_bonf * std_errors

bonferroni_CIs <- data.frame(
  Coefficient = c("moisture", "sweetness", "calories"),
  Lower = lower_bonf,
  Upper = upper_bonf,
  Width = upper_bonf - lower_bonf
)

cat("\nBonferroni-adjusted 95% Confidence Intervals:\n")
```

```
##
## Bonferroni-adjusted 95% Confidence Intervals:
```

```r
print(bonferroni_CIs)
```

```
##           Coefficient      Lower     Upper     Width
## moisture     moisture  3.3275897 5.9100822 2.5824925
## sweetness   sweetness  1.7345582 7.3568541 5.6222959
## calories     calories -0.1571832 0.2109236 0.3681067
```

```r
# ---- Compare with R's built-in confint() ----
cat("\nBuilt-in confint(model) results:\n")
```

```
##
## Built-in confint(model) results:
```

```r
print(confint(model, level = 0.95))
```

```
##                 2.5 %     97.5 %
## (Intercept)  7.9405820 48.6407665
## moisture     3.6066355  5.6310364
## sweetness    2.3420636  6.7493487
## calories    -0.1174082  0.1711485
```

Part G

```r
# Given values from ANOVA table
SSR_X1 <- 1739.11
SSR_X2_given_X1 <- 333.06
SSR_X3_given_X1_X2 <- 2.69

# Calculating combined SSR(X1, X2)
SSR_X1_X2 <- SSR_X1 + SSR_X2_given_X1

# Output results
cat("SSR(X1) - Moisture only: ", SSR_X1, "\n")
```

```
## SSR(X1) - Moisture only:  1739.11
```

```r
cat("SSR(X2|X1) - Sweetness given Moisture: ", SSR_X2_given_X1, "\n")
```

```
## SSR(X2|X1) - Sweetness given Moisture:  333.06
```

```r
cat("SSR(X3|X1, X2) - Calories given Moisture and Sweetness: ", SSR_X3_given_X1_X2, "\n")
```

```
## SSR(X3|X1, X2) - Calories given Moisture and Sweetness:  2.69
```

```r
cat("SSR(X1, X2) - Combined Moisture and Sweetness: ", SSR_X1_X2, "\n")
```

```
## SSR(X1, X2) - Combined Moisture and Sweetness:  2072.17
```

Part H

```r
model_X2_X3_X1 <- lm(brand ~ sweetness + calories + moisture, data = data)
# Generate ANOVA table for the model
anova_result <- anova(model_X2_X3_X1)
print(anova_result)
```

```
## Analysis of Variance Table
##
## Response: brand
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## sweetness  1  333.06  333.06 20.3841 0.0007081 ***
## calories   1  126.68  126.68  7.7529 0.0165153 *
## moisture   1 1615.13 1615.13 98.8489 3.812e-07 ***
## Residuals 12  196.07   16.34
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Extract SSR values directly
SSR_X2 <- anova_result$"Sum Sq"[1]  # Sum of squares for X2
SSR_X3_given_X2 <- anova_result$"Sum Sq"[2]  # Sum of squares for X3 given X2
SSR_X1_given_X2_X3 <- anova_result$"Sum Sq"[3]  # Sum of squares for X1 given X2 and X3
SSR_X2_X3 <- SSR_X2 + SSR_X3_given_X2  # Combined sum of squares for X2 and X3

# Print the results
cat("SSR(X2):", SSR_X2, "\n")
```

```
## SSR(X2): 333.0625
```

```r
cat("SSR(X3|X2):", SSR_X3_given_X2, "\n")
```

```
## SSR(X3|X2): 126.6766
```

```r
cat("SSR(X1|X2, X3):", SSR_X1_given_X2_X3, "\n")
```

```
## SSR(X1|X2, X3): 1615.126
```

```r
cat("SSR(X2, X3):", SSR_X2_X3, "\n")
```

```
## SSR(X2, X3): 459.7391
```

Part I

```r
# Load necessary library
library(stats)

# Fit models for individual predictors
model_X1 <- lm(brand ~ moisture, data = data)
model_X2 <- lm(brand ~ sweetness, data = data)
model_X3 <- lm(brand ~ calories, data = data)

# Fit models for pairs of predictors
model_X1_X2 <- lm(brand ~ moisture + sweetness, data = data)
model_X1_X3 <- lm(brand ~ moisture + calories, data = data)
model_X2_X3 <- lm(brand ~ sweetness + calories, data = data)

# Fit the full model with all predictors
model_full <- lm(brand ~ moisture + sweetness + calories, data = data)
# Calculate R^2 for individual predictors
R1_squared <- summary(model_X1)$r.squared
R2_squared <- summary(model_X2)$r.squared
R3_squared <- summary(model_X3)$r.squared

# Calculate R^2 for pairs of predictors
R12_squared <- summary(model_X1_X2)$r.squared
R13_squared <- summary(model_X1_X3)$r.squared
R23_squared <- summary(model_X2_X3)$r.squared
```

```r
# Full model R^2
R_full_squared <- summary(model_full)$r.squared
# Calculate partial R^2 values
R1_given_23 <- (R_full_squared - R23_squared) / (1 - R23_squared)
R2_given_13 <- (R_full_squared - R13_squared) / (1 - R13_squared)
R3_given_12 <- (R_full_squared - R12_squared) / (1 - R12_squared)
cat("R1^2: ", R1_squared, "\n",
    "R2^2: ", R2_squared, "\n",
    "R3^2: ", R3_squared, "\n",
    "R12^2: ", R12_squared, "\n",
    "R13^2: ", R13_squared, "\n",
    "R23^2: ", R23_squared, "\n",
    "R1|23^2: ", R1_given_23, "\n",
    "R2|13^2: ", R2_given_13, "\n",
    "R3|12^2: ", R3_given_12, "\n")
```

```
## R1^2:  0.7658126
##  R2^2:  0.146663
##  R3^2:  0.06312142
##  R12^2:  0.9124756
##  R13^2:  0.7683188
##  R23^2:  0.2024446
##  R1|23^2:  0.8917445
##  R2|13^2:  0.627334
##  R3|12^2:  0.01353563
```

Part J

```r
# Load necessary library
library(stats)

# Fit the full model with all predictors
model_full <- lm(brand ~ moisture + sweetness + calories, data = data)

# Fit the reduced model without X2 and X3
model_reduced <- lm(brand ~ moisture, data = data)

# Perform ANOVA to compare the two models
model_comparison <- anova(model_reduced, model_full)

# Calculate the F-statistic
SSE_reduced <- sum(model_reduced$residuals^2)
SSE_full <- sum(model_full$residuals^2)
df_reduced <- df.residual(model_reduced)
df_full <- df.residual(model_full)

numerator_df <- (SSE_reduced - SSE_full)
denominator_df <- df_reduced - df_full
F_statistic <- (numerator_df / 2) / (SSE_full / df_full)

# Find the critical F-value for alpha = 0.05
```

```r
F_critical <- qf(0.95, df1 = 2, df2 = df_full)

# Output the calculated F-statistic and critical F-value
cat("Calculated F-statistic: ", F_statistic, "\n")
```

```
## Calculated F-statistic:  10.27437
```

```r
cat("Critical F-value at alpha = 0.05: ", F_critical, "\n")
```

```
## Critical F-value at alpha = 0.05:  3.885294
```

```r
# State the rejection rule and conclusion
if (F_statistic > F_critical) {
  cat("Reject H0: At least one of X2 or X3 significantly contributes to the model.\n")
} else {
  cat("Do not reject H0: X2 and X3 do not significantly contribute to the model.\n")
}
```

```
## Reject H0: At least one of X2 or X3 significantly contributes to the model.
```

Part K

```r
# Load necessary library
library(stats)

# Define the full and reduced models
full_model <- lm(brand ~ moisture + sweetness + calories, data = data)
reduced_model <- lm(brand ~ moisture + sweetness, data = data)

# Perform the ANOVA to compare the models
model_comparison <- anova(reduced_model, full_model)

# Print the ANOVA table to check the output
print(model_comparison)
```

```
## Analysis of Variance Table
##
## Model 1: brand ~ moisture + sweetness
## Model 2: brand ~ moisture + sweetness + calories
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     13 198.76
## 2     12 196.07  1    2.6904 0.1647  0.692
```

```r
# Extract the F-statistic directly from the ANOVA comparison
F_statistic <- model_comparison$F[2]

df_numerator <- model_comparison$Df[2]
residual_df_full <- model_comparison$`Res.Df`[2]
```

```r
# Calculate the critical F-value
F_critical <- qf(0.95, df1 = df_numerator, df2 = residual_df_full)

# Output the F-statistic and critical F-value
cat("Calculated F-statistic: ", F_statistic, "\n")
```

```
## Calculated F-statistic:  0.1646563
```

```r
cat("Critical F-value at alpha = 0.05: ", F_critical, "\n")
```

```
## Critical F-value at alpha = 0.05:  4.747225
```

```r
# Decision Rule: Reject H0 if F_statistic > F_critical
if (F_statistic > F_critical) {
  cat("Reject H0: X3 significantly contributes to the model.\n")
} else {
  cat("Do not reject H0: X3 does not significantly contribute to the model.\n")
}
```

```
## Do not reject H0: X3 does not significantly contribute to the model.
```