Azreen Haque

3/27/2025

Solutions

Problem 1

Solutions below.

Part A

```r
data <- read.csv("hw05pr01.csv", header = TRUE, sep = ",")
names(data)
```

```
## [1] "Children" "Hours"
```

```r
# Extracting X and Y data
Xi <- data$Children
Yi <- data$Hours
n <- length(Xi)

# Compute sums manually
sum_X <- sum(Xi)
sum_Y <- sum(Yi)
sum_X2 <- sum(Xi^2)
sum_XY <- sum(Xi * Yi)

# Compute X'X manually
XtX <- matrix(c(n, sum_X,
                sum_X, sum_X2),
              nrow = 2, byrow = TRUE)

# Compute X'Y manually
XtY <- matrix(c(sum_Y, sum_XY), nrow = 2, byrow = TRUE)

# Compute determinant manually
determinant <- n * sum_X2 - (sum_X)^2

# Compute inverse (X'X)^(-1) manually
XtX_inv <- (1 / determinant) * matrix(c(sum_X2, -sum_X,
                                        -sum_X, n),
                                      nrow = 2, byrow = TRUE)

# Display results
cat("X'X =\n")
```

```
## X'X =
```

```r
print(XtX)
```

```
##      [,1] [,2]
## [1,]    6   24
## [2,]   24  142
```

```
cat("\nX'Y =\n")
```

```
##
## X'Y =
```

```
print(XtY)
```

```
##      [,1]
## [1,]   82
## [2,]  385
```

```
cat("\n(X'X)^(-1) =\n")
```

```
##
## (X'X)^(-1) =
```

```
print(XtX_inv)
```

```
##             [,1]        [,2]
## [1,]  0.51449275 -0.08695652
## [2,] -0.08695652  0.02173913
```

Part B

```
X_data <- data$Children
Y_data <- data$Hours
n <- length(X_data)

# Design matrix X and response matrix Y
X <- cbind(1, X_data)
Y <- matrix(Y_data, nrow = n, ncol = 1)

# Calculate matrices in R
XtX <- t(X) %*% X
XtY <- t(X) %*% Y
XtX_inv <- solve(XtX)

# Calculate b (Regression coefficients)
b <- XtX_inv %*% XtY

# Calculate Y-hat (Fitted values)
Y_hat <- X %*% b

# Calculate e (Residuals)
e <- Y - Y_hat

# Output all matrices
cat("X'X:\n"); print(XtX)
```

```
## X'X:

##            X_data
##          6     24
## X_data 24    142
```

```
cat("\nDimension of X'X:", dim(XtX), "\n\n")
```

```
##
## Dimension of X'X: 2 2
```

```
cat("X'Y:\n"); print(XtY)
```

```
## X'Y:

##         [,1]
##           82
## X_data   385
```

```
cat("\nDimension of X'Y:", dim(XtY), "\n\n")
```

```
##
## Dimension of X'Y: 2 1
```

```
cat("(X'X)^-1:\n"); print(XtX_inv)
```

```
## (X'X)^-1:

##                             X_data
##          0.51449275 -0.08695652
## X_data -0.08695652  0.02173913
```

```
cat("\nDimension of (X'X)^-1:", dim(XtX_inv), "\n\n")
```

```
##
## Dimension of (X'X)^-1: 2 2
```

```
cat("b (coefficients):\n"); print(b)
```

```
## b (coefficients):

##             [,1]
##         8.710145
## X_data 1.239130
```

```
cat("\nDimension of b:", dim(b), "\n\n")
```

```
##
## Dimension of b: 2 1
```

```r
cat("Y-hat (fitted values):\n"); print(Y_hat)
```

```
## Y-hat (fitted values):
```

```
##           [,1]
## [1,] 11.188406
## [2,] 16.144928
## [3,] 13.666667
## [4,] 11.188406
## [5,]  9.949275
## [6,] 19.862319
```

```r
cat("\nDimension of Y-hat:", dim(Y_hat), "\n\n")
```

```
##
## Dimension of Y-hat: 6 1
```

```r
cat("e (residuals):\n"); print(e)
```

```
## e (residuals):
```

```
##            [,1]
## [1,]  1.8115942
## [2,]  0.8550725
## [3,] -2.6666667
## [4,]  0.8115942
## [5,] -0.9492754
## [6,]  0.1376812
```

```r
cat("\nDimension of e:", dim(e), "\n\n")
```

```
##
## Dimension of e: 6 1
```

```r
# Fitted regression equation
cat(sprintf("Fitted Regression Equation: Y_hat = %.3f + %.3f*X\n", b[1], b[2]))
```

```
## Fitted Regression Equation: Y_hat = 8.710 + 1.239*X
```

Part C

```r
# Perform Simple Linear Regression using lm()
model <- lm(Y_data ~ X_data, data = data)

# Display model summary
model_summary <- summary(model)
print(model_summary)
```

```
##
## Call:
## lm(formula = Y_data ~ X_data, data = data)
##
## Residuals:
##       1       2       3       4       5       6
##  1.8116  0.8551 -2.6667  0.8116 -0.9493  0.1377
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.7101     1.2782   6.814  0.00242 **
## X_data        1.2391     0.2627   4.716  0.00920 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.782 on 4 degrees of freedom
## Multiple R-squared:  0.8476, Adjusted R-squared:  0.8095
## F-statistic: 22.24 on 1 and 4 DF,  p-value: 0.009199
```

```r
# Extract coefficients and MSE
coefficients <- model$coefficients
MSE <- sum(model$residuals^2) / (n - 2)

# Display fitted equation
cat(sprintf("\nFitted Regression Equation from lm(): Y_hat = %.3f + %.3f*X\n",
            coefficients[1], coefficients[2]))
```

```
##
## Fitted Regression Equation from lm(): Y_hat = 8.710 + 1.239*X
```

```r
# Display MSE
cat(sprintf("Mean Squared Error (MSE) = %.3f\n", MSE))
```

```
## Mean Squared Error (MSE) = 3.176
```

```r
# Output comparison statement
cat("The fitted equations from manual calculations and the lm() function match closely.\n")
```

```
## The fitted equations from manual calculations and the lm() function match closely.
```

Part D

```r
# Extract Mean Squared Error (MSE) from model summary
MSE <- summary(model)$sigma^2
cat("Mean Squared Error (MSE):", MSE, "\n\n")
```

```
## Mean Squared Error (MSE): 3.175725
```

```r
# Variance-covariance matrix calculation manually using MSE
Var_b <- MSE * XtX_inv
cat("Variance-Covariance matrix of b (manually):\n")
```

```
## Variance-Covariance matrix of b (manually):
```

```r
print(Var_b)
```

```
##                        X_data
##          1.633887 -0.27614997
## X_data -0.276150  0.06903749
```

```r
# Extract variance and covariance values from Var(b)
Var_b0 <- Var_b[1,1]    # Variance of Intercept (b0)
Var_b1 <- Var_b[2,2]    # Variance of slope (b1)
Cov_b0b1 <- Var_b[1,2]  # Covariance between b0 and b1

cat("\nVariance of b0:", Var_b0, "\n")
```

```
##
## Variance of b0: 1.633887
```

```r
cat("Variance of b1:", Var_b1, "\n")
```

```
## Variance of b1: 0.06903749
```

```r
cat("Covariance between b0 and b1:", Cov_b0b1, "\n\n")
```

```
## Covariance between b0 and b1: -0.27615
```

```r
# Compare manually computed variance-covariance with vcov() output
Var_cov_matrix_builtin <- vcov(model)
cat("Variance-Covariance Matrix from manual calculation:\n")
```

```
## Variance-Covariance Matrix from manual calculation:
```

```r
print(Var_b)
```

```
##                        X_data
##          1.633887 -0.27614997
## X_data -0.276150  0.06903749
```

```r
cat("\nVariance-Covariance Matrix using vcov() function:\n")
```

```
##
## Variance-Covariance Matrix using vcov() function:
```

```
print(Var_cov_matrix_builtin)
```

```
##             (Intercept)      X_data
## (Intercept)    1.633887 -0.27614997
## X_data        -0.276150  0.06903749
```

Part E

```
# Hat Matrix (H) calculation
H <- X %*% XtX_inv %*% t(X)

# Display Hat matrix and its dimension
cat("Hat Matrix (H):\n")
```

```
## Hat Matrix (H):
```

```
print(H)
```

```
##             [,1]       [,2]      [,3]        [,4]        [,5]        [,6]
## [1,]  0.25362319 0.07971014 0.1666667  0.25362319  0.29710145 -0.05072464
## [2,]  0.07971014 0.25362319 0.1666667  0.07971014  0.03623188  0.38405797
## [3,]  0.16666667 0.16666667 0.1666667  0.16666667  0.16666667  0.16666667
## [4,]  0.25362319 0.07971014 0.1666667  0.25362319  0.29710145 -0.05072464
## [5,]  0.29710145 0.03623188 0.1666667  0.29710145  0.36231884 -0.15942029
## [6,] -0.05072464 0.38405797 0.1666667 -0.05072464 -0.15942029  0.71014493
```

```
cat("\nDimension of H:", dim(H), "\n\n")
```

```
##
## Dimension of H: 6 6
```

```
# Compute fitted values using Hat matrix
Y_hat_H <- H %*% Y
cat("Fitted values (Y_hat) using Hat Matrix:\n")
```

```
## Fitted values (Y_hat) using Hat Matrix:
```

```
print(Y_hat)
```

```
##           [,1]
## [1,] 11.188406
## [2,] 16.144928
## [3,] 13.666667
## [4,] 11.188406
## [5,]  9.949275
## [6,] 19.862319
```

```
# Verify if the fitted values match those from lm()
Y_hat_lm <- fitted(model)
match_fitted <- all.equal(as.vector(Y_hat), as.vector(Y_hat_lm), tolerance=1e-5)

if(isTRUE(match_fitted)){
  cat("\nFitted values from Hat Matrix match those from lm().\n\n")
} else {
  cat("Fitted values do NOT match.\n")
}
```

```
##
## Fitted values from Hat Matrix match those from lm().
```

Part F

```
H_transpose <- t(H)
HH <- H %*% H

# Check symmetry
is_symmetric <- all.equal(H, H_transpose, tolerance=1e-5)

# Check idempotency
is_idempotent <- all.equal(HH, H, tolerance=1e-5)

cat("\nHat Matrix symmetry check (H' = H):", is_symmetric, "\n")
```

```
##
## Hat Matrix symmetry check (H' = H): TRUE
```

```
cat("Hat Matrix idempotent check (HH = H):", is_idempotent, "\n")
```

```
## Hat Matrix idempotent check (HH = H): TRUE
```

Part G

```
# Prediction using predict() function
new_data <- data.frame(X_data = 5)
predict_lm <- predict(model, newdata = new_data)

cat("Prediction using predict() function (X=5):", predict_lm, "\n")
```

```
## Prediction using predict() function (X=5): 14.9058
```

```
# Prediction manually using matrices Xh and b
Xh <- matrix(c(1, 5), nrow = 1)
predict_manual <- Xh %*% b

cat("Prediction manually calculated using matrices (X=5):", predict_manual, "\n")
```

```
## Prediction manually calculated using matrices (X=5): 14.9058
```

```r
# Compare the predictions
prediction_match <- all.equal(as.numeric(predict_lm), as.numeric(Xh %*% b), tolerance = 1e-5)

if (isTRUE(prediction_match)){
  cat("\nThe predictions from predict() and manual calculation match closely.\n")
} else {
  cat("\nThe predictions from predict() and manual calculation do NOT match.\n")
}
```

```
##
## The predictions from predict() and manual calculation match closely.
```

Problem 2

Part A

```r
data <- read.csv("hw05pr02.csv", header = TRUE, sep = ",")
names(data)
```

```
## [1] "satisfaction" "age"          "illness"      "anxiety"
```

```r
model <- lm(satisfaction ~ age + illness + anxiety, data = data)

# Display the regression summary
summary(model)
```

```
##
## Call:
## lm(formula = satisfaction ~ age + illness + anxiety, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.6138  -7.3235   0.6604   8.9471  18.3025
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 155.3427    17.5413   8.856 3.71e-11 ***
## age          -1.1431     0.2172  -5.263 4.51e-06 ***
## illness      -0.4476     0.4617  -0.970    0.338
## anxiety     -13.2427     6.0418  -2.192    0.034 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.1 on 42 degrees of freedom
## Multiple R-squared:  0.6786, Adjusted R-squared:  0.6556
## F-statistic: 29.56 on 3 and 42 DF,  p-value: 1.948e-10
```

```r
# Extract coefficients
coefficients <- model$coefficients
cat(sprintf("\nFitted Regression Equation:\nSatisfaction = %.3f + %.3f*Age + %.3f*Illness + %.3f*Anxiet
            coefficients[1], coefficients[2], coefficients[3], coefficients[4]))
```

```
##
## Fitted Regression Equation:
## Satisfaction = 155.343 + -1.143*Age + -0.448*Illness + -13.243*Anxiety
```

```r
# Interpretation of coefficients
cat("\nInterpretation of Coefficients:\n")
```

```
##
## Interpretation of Coefficients:
```

```r
cat("1. Intercept (b0): The predicted satisfaction when age,\n")
```

```
## 1. Intercept (b0): The predicted satisfaction when age,
```

```r
cat("   illness, and anxiety are all zero. In a real-world\n")
```

```
##    illness, and anxiety are all zero. In a real-world
```

```r
cat("   context, this may not be meaningful as a patient\n")
```

```
##    context, this may not be meaningful as a patient
```

```r
cat("   with zero age is unrealistic in regards to years & satisfaction.\n\n")
```

```
##    with zero age is unrealistic in regards to years & satisfaction.
```

```r
cat("2. Age (b1): A one-year increase in age decreases\n")
```

```
## 2. Age (b1): A one-year increase in age decreases
```

```r
cat("   satisfaction by", round(coefficients[2], 3), "points,\n")
```

```
##    satisfaction by -1.143 points,
```

```r
cat("   holding illness and anxiety constant. Since\n")
```

```
##    holding illness and anxiety constant. Since
```

```r
cat("   p-value < 0.05, age is statistically significant.\n\n")
```

```
##    p-value < 0.05, age is statistically significant.
```

```r
cat("3. Illness (b2): A one-unit increase in illness\n")
```

```
## 3. Illness (b2): A one-unit increase in illness
```

```r
cat("   decreases satisfaction by", round(coefficients[3], 3), "points.\n")
```

```
##    decreases satisfaction by -0.448 points.
```

```r
cat("   Since p-value > 0.05, illness is not statistically\n")
```

```
##    Since p-value > 0.05, illness is not statistically
```

```r
cat("   significant. In practice, we may not interpret this\n")
```

```
##    significant. In practice, we may not interpret this
```

```r
cat("   coefficient as reliable.\n\n")
```

```
##    coefficient as reliable.
```

```r
cat("4. Anxiety (b3): A one-unit increase in anxiety\n")
```

```
## 4. Anxiety (b3): A one-unit increase in anxiety
```

```r
cat("   decreases satisfaction by", round(coefficients[4], 3), "points.\n")
```

```
##    decreases satisfaction by -13.243 points.
```

```r
cat("   P-value < 0.05, meaning anxiety is statistically\n")
```

```
##    P-value < 0.05, meaning anxiety is statistically
```

```r
cat("   significant in predicting satisfaction.\n\n")
```

```
##    significant in predicting satisfaction.
```

Part B

```r
X <- as.matrix(cbind(1, data$age, data$illness, data$anxiety))
Y <- as.matrix(data$satisfaction)

XtX <- t(X) %*% X # Compute X'X
XtX_inv <- solve(XtX) # Compute (X'X)^-1
XtY <- t(X) %*% Y # Compute X'Y
b <- XtX_inv %*% XtY # Compute b (coefficients)

# Output matrices and their dimensions
cat("\nMatrix X'X:\n")
```

```
##
## Matrix X'X:
```

```
print(XtX)
```

```
##         [,1]    [,2]   [,3]    [,4]
## [1,]   46.0  1716.0   2267  104.70
## [2,] 1716.0 67544.0  85607 3970.30
## [3,] 2267.0 85607.0 112605 5197.00
## [4,]  104.7  3970.3   5197  242.85
```

```
cat("Dimensions of X'X:", dim(XtX), "\n\n")
```

```
## Dimensions of X'X: 4 4
```

```
cat("Matrix (X'X)^-1:\n")
```

```
## Matrix (X'X)^-1:
```

```
print(XtX_inv)
```

```
##              [,1]          [,2]         [,3]         [,4]
## [1,]  3.01796532  0.0102446204 -0.062133777 -0.138957994
## [2,]  0.01024462  0.0004626133 -0.000408794 -0.003231719
## [3,] -0.06213378 -0.0004087940  0.002090678 -0.011269560
## [4,] -0.13895799 -0.0032317190 -0.011269560  0.358030458
```

```
cat("Dimensions of (X'X)^-1:", dim(XtX_inv), "\n\n")
```

```
## Dimensions of (X'X)^-1: 4 4
```

```
cat("Matrix X'Y:\n")
```

```
## Matrix X'Y:
```

```
print(XtY)
```

```
##          [,1]
## [1,]   2783.0
## [2,]  98464.0
## [3,] 135081.0
## [4,]   6183.8
```

```
cat("Dimensions of X'Y:", dim(XtY), "\n\n")
```

```
## Dimensions of X'Y: 4 1
```

```
cat("Vector b (coefficients):\n")
```

```
## Vector b (coefficients):
```

```
print(b)
```

```
##             [,1]
## [1,] 155.342680
## [2,]  -1.143071
## [3,]  -0.447615
## [4,] -13.242700
```

```
cat("Dimensions of b:", dim(b), "\n")
```

```
## Dimensions of b: 4 1
```

Part C

```
# Extract MSE from model summary
MSE <- summary(model)$sigma^2

# Compute Variance-Covariance Matrix
Var_b <- MSE * XtX_inv

# Extract variances and covariances
Var_b0 <- Var_b[1,1]  # Variance of Intercept
Var_b1 <- Var_b[2,2]  # Variance of Age
Var_b2 <- Var_b[3,3]  # Variance of Illness
Var_b3 <- Var_b[4,4]  # Variance of Anxiety

Cov_b1b2 <- Var_b[2,3]  # Covariance between Age and Illness
Cov_b1b3 <- Var_b[2,4]  # Covariance between Age and Anxiety
Cov_b2b3 <- Var_b[3,4]  # Covariance between Illness and Anxiety

# Compare with built-in vcov() function
Var_b_builtin <- vcov(model)

cat("\nVariance-Covariance Matrix of b (Manual Calculation):\n", Var_b, "\n")
```

```
##
## Variance-Covariance Matrix of b (Manual Calculation):
##  307.698 1.044495 -6.334877 -14.16752 1.044495 0.04716594 -0.04167877 -0.3294914 -6.334877 -0.041678
```

```
cat("\nVariance-Covariance Matrix using vcov():\n", Var_b_builtin, "\n")
```

```
##
## Variance-Covariance Matrix using vcov():
##  307.698 1.044495 -6.334877 -14.16752 1.044495 0.04716594 -0.04167877 -0.3294914 -6.334877 -0.041678
```

Part D

```r
# Perform ANOVA
anova_results <- anova(model)

# Extract Sum of Squares
SSR <- sum(anova_results[1:3,2])
SSE <- anova_results[4,2]
SST <- SSR + SSE

# Extract Degrees of Freedom
df_regression <- sum(anova_results[1:3,1])
df_error <- anova_results[4,1]
df_total <- df_regression + df_error

# Compute Mean Squares
MSR <- SSR / df_regression
MSE <- SSE / df_error

# Manually Constructed ANOVA Table
anova_manual <- data.frame(
  Source = c("Regression", "Error", "Total"),
  DF = c(df_regression, df_error, df_total),
  Sum_Sq = c(SSR, SSE, SST),
  Mean_Sq = c(MSR, MSE, NA)
)

# Print ANOVA Table
print(anova_manual)
```

```
##       Source DF    Sum_Sq   Mean_Sq
## 1 Regression  3  9041.371 3013.7904
## 2      Error 42  4282.129  101.9554
## 3      Total 45 13323.500        NA
```

Part E

```r
# Extract Sum of Squares values from ANOVA table
SSTO <- 13323.500
SSR <- 9041.371
SSE <- 4282.129

# Compute R-squared
R_squared <- SSR / SSTO

# Display result
cat(sprintf("\nR-squared (from ANOVA table) = %.4f\n", R_squared))
```

```
##
## R-squared (from ANOVA table) = 0.6786
```

```r
# Compare with summary(model) R-squared
summary_R2 <- summary(model)$r.squared
cat(sprintf("\nR-squared (from model summary) = %.4f\n", summary_R2))
```

```
##
## R-squared (from model summary) = 0.6786
```

```r
# Interpretation
cat("\nInterpretation:\n")
```

```
##
## Interpretation:
```

```r
cat("\nInterpretation:\n")
```

```
##
## Interpretation:
```

```r
cat("R-squared represents the proportion of variability in satisfaction\n")
```

```
## R-squared represents the proportion of variability in satisfaction
```

```r
cat("explained by age, illness, and anxiety.\n")
```

```
## explained by age, illness, and anxiety.
```

```r
cat(sprintf("With R^2 = %.4f, this means that %.2f%% of the variation in satisfaction\n",
            R_squared, R_squared * 100))
```

```
## With R^2 = 0.6786, this means that 67.86% of the variation in satisfaction
```

```r
cat("is explained by the predictors in the model, while the remaining 32.14%\n")
```

```
## is explained by the predictors in the model, while the remaining 32.14%
```

```r
cat("is due to other factors not included in the model.\n")
```

```
## is due to other factors not included in the model.
```

```r
cat("The values from the ANOVA table and model summary should match closely.\n")
```

```
## The values from the ANOVA table and model summary should match closely.
```

Part F

```r
# Compute SSTO, SSE, and SSR using summation formulas
# Total Sum of Squares (SSTO)
sst_total <- sum((data$satisfaction - mean(data$satisfaction))^2)
cat("\nSSTO (Summation Formula):", sst_total, "\n")
```

```
##
## SSTO (Summation Formula): 13323.5
```

```r
# Error Sum of Squares (SSE)
predicted_values <- predict(model)
sse_error <- sum((data$satisfaction - predicted_values)^2)
cat("SSE (Summation Formula):", sse_error, "\n")
```

```
## SSE (Summation Formula): 4282.129
```

```r
# Regression Sum of Squares (SSR)
ssr_regression <- sst_total - sse_error
cat("SSR (Summation Formula):", ssr_regression, "\n")
```

```
## SSR (Summation Formula): 9041.371
```

```r
# Compute SSTO, SSE, and SSR using matrix form
# Define response variable Y and design matrix X
Y <- as.matrix(data$satisfaction)
X <- model.matrix(model)

# Compute hat matrix H
H <- X %*% solve(t(X) %*% X) %*% t(X)

# Compute SSTO (Total Sum of Squares)
sst_matrix <- t(Y - mean(Y)) %*% (Y - mean(Y))
cat("\nSSTO (Matrix Form):", sst_matrix[1,1], "\n")
```

```
##
## SSTO (Matrix Form): 13323.5
```

```r
# Compute SSE (Error Sum of Squares)
sse_matrix <- t(Y - H %*% Y) %*% (Y - H %*% Y)
cat("SSE (Matrix Form):", sse_matrix[1,1], "\n")
```

```
## SSE (Matrix Form): 4282.129
```

```r
# Compute SSR (Regression Sum of Squares)
ssr_matrix <- sst_matrix - sse_matrix
cat("SSR (Matrix Form):", ssr_matrix[1,1], "\n")
```

```
## SSR (Matrix Form): 9041.371
```

```
# Compare the two methods
cat("\nComparison:\n")
```

```
##
## Comparison:
```

```
cat("Difference in SSTO:", abs(sst_total - sst_matrix[1,1]), "\n")
```

```
## Difference in SSTO: 0
```

```
cat("Difference in SSE:", abs(sse_error - sse_matrix[1,1]), "\n")
```

```
## Difference in SSE: 2.073648e-10
```

```
cat("Difference in SSR:", abs(ssr_regression - ssr_matrix[1,1]), "\n")
```

```
## Difference in SSR: 2.073648e-10
```

Problem 3

Part A

```
# Given Matrices and Vectors
E_V <- matrix(c(4, -1), nrow = 2)
E_U <- matrix(c(-2, 3), nrow = 2)
Var_V <- matrix(c(2,1,1,3), nrow = 2)
Var_U <- matrix(c(1,-1,-1,2), nrow = 2)
A <- matrix(c(-3,1,2,-5), nrow = 2, byrow = TRUE)
B <- matrix(c(0,-2,1,4), nrow = 2, byrow = TRUE)
C <- matrix(c(-1,2), nrow = 2)

# (a) E(AV + BU)
E_AV_BU <- A %*% E_V + B %*% E_U
cat("E(AV + BU):\n")
```

```
## E(AV + BU):
```

```
print(E_AV_BU)
```

```
##      [,1]
## [1,]  -19
## [2,]   23
```

Part B

```r
Var_AV <- A %*% Var_V %*% t(A)
cat("\nVar(AV + C):\n")
```

```
##
## Var(AV + C):
```

```r
print(Var_AV) # C doesn't affect variance
```

```
##      [,1] [,2]
## [1,]   15  -10
## [2,]  -10   63
```

Part C

```r
# (c) Var(BU + C)
Var_BU <- B %*% Var_U %*% t(B)
cat("\nVar(BU + C):\n")
```

```
##
## Var(BU + C):
```

```r
print(Var_BU) # C does not affect variance,omitted
```

```
##      [,1] [,2]
## [1,]    8  -14
## [2,]  -14   25
```