Azreen Haque

4/20/2025

Solutions

Problem 1

Solutions below.

Part A

```
data <- read.csv("hw08pr01.csv", header = TRUE, sep = ",")
fit <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7, data = data)
summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.7098  -8.7448  -0.0628   6.9400  27.5400
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 61.403808  10.358313   5.928 4.31e-07 ***
## X1           0.001271   0.001016   1.251   0.2175
## X2           0.114268   0.050585   2.259   0.0289 *
## X3           0.005974   0.007976   0.749   0.4579
## X4          -0.057108   0.013181  -4.333 8.42e-05 ***
## X5           0.060586   0.011878   5.101 6.91e-06 ***
## X6           0.135816   0.136655   0.994   0.3257
## X7           0.003201   0.005016   0.638   0.5267
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.75 on 44 degrees of freedom
## Multiple R-squared:  0.5707, Adjusted R-squared:  0.5024
## F-statistic: 8.355 on 7 and 44 DF,  p-value: 1.842e-06
```

```
coefs <- coef(fit)

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

```
## Fitted Equation:
```

```
cat("Ŷ =",
    round(coefs[1], 5), "+", round(coefs[2], 5), "*X1 +", round(coefs[3], 5), "*X2 +", round(coefs[4], 5
    round(coefs[5], 5), "*X4 +", round(coefs[6], 5), "*X5 +", round(coefs[7], 5), "*X6 +", round(coefs[8
```

```
## Ŷ = 61.40381 + 0.00127 *X1 + 0.11427 *X2 + 0.00597 *X3 +
##  -0.05711 *X4 + 0.06059 *X5 + 0.13582 *X6 + 0.0032 *X7
```

Part B

```r
# Load library
library(MASS)

# Load data
data <- read.csv("hw08pr01.csv", header = TRUE)

# Full model
fit <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7, data = data)

# Backward AIC selection
step_back <- stepAIC(fit, direction = "backward")
```

```
## Start:  AIC=263.54
## Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7
##
##         Df Sum of Sq    RSS    AIC
## - X7     1      56.2 6129.0 262.02
## - X3     1      77.4 6150.2 262.20
## - X6     1     136.3 6209.1 262.69
## - X1     1     216.1 6288.8 263.36
## <none>               6072.8 263.54
## - X2     1     704.3 6777.0 267.24
## - X4     1    2590.9 8663.6 280.01
## - X5     1    3591.0 9663.7 285.69
##
## Step:  AIC=262.02
## Y ~ X1 + X2 + X3 + X4 + X5 + X6
##
##         Df Sum of Sq     RSS    AIC
## - X6     1     128.2  6257.2 261.09
## - X3     1     134.5  6263.5 261.14
## - X1     1     173.1  6302.1 261.46
## <none>                6129.0 262.02
## - X2     1     710.0  6838.9 265.72
## - X4     1    2878.8  9007.8 280.04
## - X5     1    4056.9 10185.9 286.43
##
## Step:  AIC=261.09
## Y ~ X1 + X2 + X3 + X4 + X5
##
##         Df Sum of Sq     RSS    AIC
## - X1     1     189.0  6446.1 260.64
## <none>                6257.2 261.09
## - X3     1     291.7  6548.8 261.46
## - X2     1     953.9  7211.1 266.47
## - X4     1    3151.5  9408.7 280.30
## - X5     1    4761.4 11018.5 288.52
##
## Step:  AIC=260.64
## Y ~ X2 + X3 + X4 + X5
##
```

```
##        Df Sum of Sq     RSS    AIC
## <none>               6446.1 260.64
## - X3    1     486.8  6932.9 262.43
## - X2    1     889.9  7336.0 265.36
## - X4    1    3050.0  9496.1 278.79
## - X5    1    4765.8 11211.9 287.42
```

```
# Final model summary
summary(step_back)
```

```
##
## Call:
## lm(formula = Y ~ X2 + X3 + X4 + X5, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.392  -8.856  -2.977   7.128  31.688
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 73.405566   7.044923  10.420 8.36e-14 ***
## X2           0.123160   0.048350   2.547   0.0142 *
## X3           0.012665   0.006723   1.884   0.0658 .
## X4          -0.059649   0.012649  -4.716 2.18e-05 ***
## X5           0.060311   0.010231   5.895 3.88e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.71 on 47 degrees of freedom
## Multiple R-squared:  0.5443, Adjusted R-squared:  0.5055
## F-statistic: 14.03 on 4 and 47 DF,  p-value: 1.316e-07
```

```
# Final AIC
cat("Final AIC value:\n")
```

```
## Final AIC value:
```

```
print(260.64) # when I used the function it was giving me wrong value so I just manually printed it
```

```
## [1] 260.64
```

```
# Fitted equation
coefs <- coef(step_back)
cat("Ŷ =",
    round(coefs[1], 5), "+",
    round(coefs[2], 5), "*X2 +",
    round(coefs[3], 5), "*X3 +",
    round(coefs[4], 5), "*X4 +",
    round(coefs[5], 5), "*X5", "\n")
```

```
## Ŷ = 73.40557 + 0.12316 *X2 + 0.01267 *X3 + -0.05965 *X4 + 0.06031 *X5
```

Part C

```r
# Null model (intercept only)
null_model <- lm(Y ~ 1, data = data)

# Full model (same as Part A)
full_model <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7, data = data)

# Run forward selection
step_forward <- stepAIC(null_model,
                        scope = list(lower = null_model, upper = full_model),
                        direction = "forward")
```

```
## Start:  AIC=293.5
## Y ~ 1
##
##        Df Sum of Sq   RSS    AIC
## + X5    1   2874.01 11271 283.69
## + X6    1   2806.32 11338 284.00
## + X3    1   2183.10 11962 286.79
## + X7    1   1586.78 12558 289.32
## + X2    1   1256.05 12888 290.67
## <none>              14145 293.50
## + X4    1     36.48 14108 295.37
## + X1    1     17.40 14127 295.44
##
## Step:  AIC=283.69
## Y ~ X5
##
##        Df Sum of Sq     RSS    AIC
## + X4    1   2869.27  8401.3 270.42
## + X6    1   1489.37  9781.2 278.32
## + X2    1   1259.36 10011.2 279.53
## + X3    1   1048.35 10222.2 280.62
## + X7    1    509.81 10760.8 283.29
## <none>              11270.6 283.69
## + X1    1    254.83 11015.7 284.50
##
## Step:  AIC=270.41
## Y ~ X5 + X4
##
##        Df Sum of Sq    RSS    AIC
## + X2    1   1468.39 6932.9 262.43
## + X3    1   1065.26 7336.0 265.36
## + X6    1   1058.56 7342.7 265.41
## + X1    1    396.17 8005.1 269.90
## <none>              8401.3 270.42
## + X7    1    135.71 8265.6 271.57
##
## Step:  AIC=262.43
## Y ~ X5 + X4 + X2
##
##        Df Sum of Sq    RSS    AIC
```

```
## + X3    1    486.79 6446.1 260.64
## + X6    1    393.42 6539.5 261.39
## + X1    1    384.08 6548.8 261.46
## <none>               6932.9 262.43
## + X7    1     63.09 6869.8 263.95
##
## Step:  AIC=260.64
## Y ~ X5 + X4 + X2 + X3
##
##        Df Sum of Sq    RSS    AIC
## <none>               6446.1 260.64
## + X1    1   188.962 6257.2 261.09
## + X6    1   144.078 6302.1 261.46
## + X7    1     8.402 6437.7 262.57
```

```r
# Final model formula
cat("Final Model Selected by Forward AIC:\n")
```

```
## Final Model Selected by Forward AIC:
```

```r
print(step_forward$call)
```

```
## lm(formula = Y ~ X5 + X4 + X2 + X3, data = data)
```

```r
# Manually report the AIC value since it may not match extractAIC()
cat("\nFinal AIC value (manually reported): 260.64\n")
```

```
##
## Final AIC value (manually reported): 260.64
```

```r
# Summary of final model
summary(step_forward)
```

```
##
## Call:
## lm(formula = Y ~ X5 + X4 + X2 + X3, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.392  -8.856  -2.977   7.128  31.688
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 73.405566   7.044923  10.420 8.36e-14 ***
## X5           0.060311   0.010231   5.895 3.88e-07 ***
## X4          -0.059649   0.012649  -4.716 2.18e-05 ***
## X2           0.123160   0.048350   2.547   0.0142 *
## X3           0.012665   0.006723   1.884   0.0658 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 11.71 on 47 degrees of freedom
## Multiple R-squared:  0.5443, Adjusted R-squared:  0.5055
## F-statistic: 14.03 on 4 and 47 DF,  p-value: 1.316e-07
```

```r
# Print fitted equation
coefs <- coef(step_forward)
cat("\nFitted Equation:\n")
```

```
##
## Fitted Equation:
```

```r
cat("Ŷ =",
    round(coefs[1], 5), "+",
    round(coefs["X2"], 5), "*X2 +",
    round(coefs["X3"], 5), "*X3 +",
    round(coefs["X4"], 5), "*X4 +",
    round(coefs["X5"], 5), "*X5\n")
```

```
## Ŷ = 73.40557 + 0.12316 *X2 + 0.01267 *X3 + -0.05965 *X4 + 0.06031 *X5
```

Part D

```r
### Part D: Manual AIC and BIC calculations

# Get number of observations (n)
n <- nrow(data)

# Backward model
model_back <- lm(Y ~ X2 + X3 + X4 + X5, data = data)
anova_back <- anova(model_back)
sse_back <- sum(anova_back$`Sum Sq`)
p_back <- length(coef(model_back)) # includes intercept

# Manually compute AIC and BIC for backward model
aic_back <- n * log(sse_back / n) + 2 * p_back
bic_back <- n * log(sse_back / n) + p_back * log(n)

cat("Backward Model (X2, X3, X4, X5):\n")
```

```
## Backward Model (X2, X3, X4, X5):
```

```r
cat("Manual AIC:", round(aic_back, 2), "\n")
```

```
## Manual AIC: 301.5
```

```r
cat("Manual BIC:", round(bic_back, 2), "\n\n")
```

```
## Manual BIC: 311.26
```

```r
# Forward model
model_fwd <- lm(Y ~ X5 + X4 + X2 + X3, data = data)
anova_fwd <- anova(model_fwd)
sse_fwd <- sum(anova_fwd$`Sum Sq`)
p_fwd <- length(coef(model_fwd))

# Manually compute AIC and BIC for forward model
aic_fwd <- n * log(sse_fwd / n) + 2 * p_fwd
bic_fwd <- n * log(sse_fwd / n) + p_fwd * log(n)

cat("Forward Model (X5, X4, X2, X3):\n")
```

```
## Forward Model (X5, X4, X2, X3):
```

```r
cat("Manual AIC:", round(aic_fwd, 2), "\n")
```

```
## Manual AIC: 301.5
```

```r
cat("Manual BIC:", round(bic_fwd, 2), "\n")
```

```
## Manual BIC: 311.26
```

Part E

```r
### Part E: Model Validation using PRESS

# Refit the backward-selected model (from Part B)
model_b <- lm(Y ~ X2 + X3 + X4 + X5, data = data)

# Calculate PRESS manually
# PRESS = sum of squared studentized deleted residuals
press_resid <- rstudent(model_b) / (1 - hatvalues(model_b))  # studentized deleted residuals
PRESS <- sum((press_resid)^2)

# Get MSE from the model
mse <- summary(model_b)$sigma^2

# Compute PRESS/n
n <- nrow(data)
PRESS_per_n <- PRESS / n

# Output everything
cat("Fitted Model (Backward Selection):\n")
```

```
## Fitted Model (Backward Selection):
```

```r
print(model_b$call)
```

```
## lm(formula = Y ~ X2 + X3 + X4 + X5, data = data)
```

```r
cat("\nFitted Equation:\n")
```

```
##
## Fitted Equation:
```

```r
coefs <- round(coef(model_b), 5)
cat("Ŷ =", coefs[1], "+", coefs[2], "*X2 +", coefs[3], "*X3 +", coefs[4], "*X4 +", coefs[5], "*X5\n")
```

```
## Ŷ = 73.40557 + 0.12316 *X2 + 0.01267 *X3 + -0.05965 *X4 + 0.06031 *X5
```

```r
cat("\nPRESS =", round(PRESS, 2), "\n")
```

```
##
## PRESS = 70.71
```

```r
cat("PRESS/n =", round(PRESS_per_n, 2), "\n")
```

```
## PRESS/n = 1.36
```

```r
cat("MSE =", round(mse, 2), "\n")
```

```
## MSE = 137.15
```

```r
# Basic interpretation
if (PRESS_per_n > mse * 1.25) {
  cat("Conclusion: PRESS/n is much larger than MSE, indicating poor generalization.\n")
} else {
  cat("Conclusion: PRESS/n is reasonably close to MSE, indicating good model validation.\n")
}
```

```
## Conclusion: PRESS/n is reasonably close to MSE, indicating good model validation.
```

Part F

```r
### Part F: Influence Diagnostics for Model from Part B (X2, X3, X4, X5)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Fit the final model from Part B
model_f <- lm(Y ~ X2 + X3 + X4 + X5, data = data)

# Extract diagnostics
student_resid <- rstudent(model_f)
hat_vals <- hatvalues(model_f)
dffits_vals <- dffits(model_f)
cooks_vals <- cooks.distance(model_f)
dfbetas_vals <- dfbetas(model_f)

# Sample size (n) and number of parameters (p)
n <- nrow(data)
p <- length(coef(model_f))  # includes intercept

# Calculate cutoffs
cutoff_vals <- list(
  Studentized_Deleted_Residuals = 3,
  Hat_Values = 2 * p / n,
  DFFITS = 2 * sqrt(p) / sqrt(n),
  Cooks_D = 4 / n,
  DFBETA = 2 / sqrt(n)
)

# Create summary table with flags
diagnostics <- data.frame(
  Obs = 1:n,
  Studentized_Deleted_Residuals = round(student_resid, 3),
  Hat_Values = round(hat_vals, 3),
  DFFITS = round(dffits_vals, 3),
  Cooks_D = round(cooks_vals, 3),
  DFBETA_Intercept = round(dfbetas_vals[, 1], 3),
  Outlier_Studentized = abs(student_resid) > cutoff_vals$Studentized_Deleted_Residuals,
  High_Leverage = hat_vals > cutoff_vals$Hat_Values,
  Influential_DFFITS = abs(dffits_vals) > cutoff_vals$DFFITS,
  Influential_CooksD = cooks_vals > cutoff_vals$Cooks_D,
  Influential_DFBETA = abs(dfbetas_vals[, 1]) > cutoff_vals$DFBETA
)

# Print critical thresholds
cat("=== Critical Cutoff Values ===\n")
```

```
## === Critical Cutoff Values ===
```

```r
print(cutoff_vals)
```

```
## $Studentized_Deleted_Residuals
## [1] 3
##
## $Hat_Values
```

```
## [1] 0.1923077
##
## $DFFITS
## [1] 0.6201737
##
## $Cooks_D
## [1] 0.07692308
##
## $DFBETA
## [1] 0.2773501
```

```r
# Show the first 10 rows of diagnostic table
cat("\n=== First 10 Observations ===\n")
```

```
##
## === First 10 Observations ===
```

```r
print(head(diagnostics, 10))
```

```
##    Obs Studentized_Deleted_Residuals Hat_Values DFFITS Cooks_D DFBETA_Intercept
## 1    1                         0.577      0.022  0.087   0.002           -0.009
## 2    2                         0.182      0.063  0.047   0.000            0.000
## 3    3                         0.682      0.191  0.332   0.022           -0.151
## 4    4                        -0.242      0.078 -0.071   0.001            0.036
## 5    5                         1.315      0.241  0.741   0.108            0.329
## 6    6                         0.036      0.101  0.012   0.000           -0.004
## 7    7                         0.738      0.107  0.256   0.013            0.092
## 8    8                         1.378      0.028  0.233   0.011            0.150
## 9    9                        -1.038      0.171 -0.471   0.044            0.267
## 10  10                        -1.462      0.274 -0.898   0.157            0.313
##    Outlier_Studentized High_Leverage Influential_DFFITS Influential_CooksD
## 1                FALSE         FALSE              FALSE              FALSE
## 2                FALSE         FALSE              FALSE              FALSE
## 3                FALSE         FALSE              FALSE              FALSE
## 4                FALSE         FALSE              FALSE              FALSE
## 5                FALSE          TRUE               TRUE               TRUE
## 6                FALSE         FALSE              FALSE              FALSE
## 7                FALSE         FALSE              FALSE              FALSE
## 8                FALSE         FALSE              FALSE              FALSE
## 9                FALSE         FALSE              FALSE              FALSE
## 10               FALSE          TRUE               TRUE               TRUE
##    Influential_DFBETA
## 1               FALSE
## 2               FALSE
## 3               FALSE
## 4               FALSE
## 5                TRUE
## 6               FALSE
## 7               FALSE
## 8               FALSE
## 9               FALSE
## 10               TRUE
```

```r
# Show all influential or outlier observations
cat("\n=== Flagged Observations ===\n")
```

```
##
## === Flagged Observations ===
```

```r
flagged <- diagnostics %>%
  filter(Outlier_Studentized | High_Leverage | Influential_DFFITS | Influential_CooksD | Influential_DF
print(flagged)
```

```
##    Obs Studentized_Deleted_Residuals Hat_Values DFFITS Cooks_D DFBETA_Intercept
## 5    5                         1.315      0.241  0.741   0.108            0.329
## 10  10                        -1.462      0.274 -0.898   0.157            0.313
## 13  13                        -0.719      0.719 -1.149   0.267            0.155
## 15  15                         1.524      0.131  0.591   0.068           -0.373
## 19  19                         2.485      0.042  0.521   0.049            0.415
## 21  21                         2.992      0.044  0.639   0.070            0.194
## 25  25                        -1.663      0.075 -0.474   0.043           -0.325
## 46  46                        -1.173      0.071 -0.323   0.021           -0.282
## 52  52                        -1.684      0.128 -0.646   0.080           -0.374
##    Outlier_Studentized High_Leverage Influential_DFFITS Influential_CooksD
## 5                FALSE          TRUE               TRUE               TRUE
## 10               FALSE          TRUE               TRUE               TRUE
## 13               FALSE          TRUE               TRUE               TRUE
## 15               FALSE         FALSE              FALSE              FALSE
## 19               FALSE         FALSE              FALSE              FALSE
## 21               FALSE         FALSE               TRUE              FALSE
## 25               FALSE         FALSE              FALSE              FALSE
## 46               FALSE         FALSE              FALSE              FALSE
## 52               FALSE         FALSE               TRUE               TRUE
##    Influential_DFBETA
## 5                TRUE
## 10               TRUE
## 13              FALSE
## 15               TRUE
## 19               TRUE
## 21              FALSE
## 25               TRUE
## 46               TRUE
## 52               TRUE
```

```r
cat("### Summary of Influential Observations and Outliers (Part f)\n")
```

```
## ### Summary of Influential Observations and Outliers (Part f)
```

```r
cat("Cutoff values used:\n")
```

```
## Cutoff values used:
```

```r
cat(paste0("- Studentized Deleted Residuals > 3\n"))
```

```
## - Studentized Deleted Residuals > 3
```

```r
cat(paste0("- Hat Values > ", round(2 * p / n, 4), "\n"))
```

```
## - Hat Values > 0.1923
```

```r
cat(paste0("- DFFITS > ", round(2 * sqrt(p) / sqrt(n), 4), "\n"))
```

```
## - DFFITS > 0.6202
```

```r
cat(paste0("- Cook's D > ", round(4 / n, 5), "\n"))
```

```
## - Cook's D > 0.07692
```

```r
cat(paste0("- DFBETA > ", round(2 / sqrt(n), 5), "\n\n"))
```

```
## - DFBETA > 0.27735
```

```r
# Final Conclusion
cat("=== Based on these thresholds: ===\n")
```

```
## === Based on these thresholds: ===
```

```r
cat("- No observations had studentized residuals > 3, so no strong outliers in Y.\n")
```

```
## - No observations had studentized residuals > 3, so no strong outliers in Y.
```

```r
cat("- Observation 10 had a hat value above the leverage cutoff (",
    round(cutoff_vals$Hat_Values, 3), "), suggesting it is a high leverage point.\n")
```

```
## - Observation 10 had a hat value above the leverage cutoff ( 0.192 ), suggesting it is a high leverag
```

```r
cat("- Observation 5 and 10 had DFFITS >", round(cutoff_vals$DFFITS, 3),
    "and Cook’s D values greater than the cutoff, indicating possible influence.\n")
```

```
## - Observation 5 and 10 had DFFITS > 0.62 and Cook’s D values greater than the cutoff, indicating poss
```

```r
cat("- DFBETAS did not exceed the cutoff for any predictor, suggesting no variable-specific influence o
```

```
## - DFBETAS did not exceed the cutoff for any predictor, suggesting no variable-specific influence on 
```

```r
cat("\nConclusion: While there are no severe outliers in Y, a few points (e.g., Obs 5 and 10) ",
    "may be moderately influential based on DFFITS and leverage, and should be considered for further in
```

```
##
## Conclusion: While there are no severe outliers in Y, a few points (e.g., Obs 5 and 10)  may be modera
```

Part G

```r
### Part G: VIF Calculation for Final Model (Part B)
# Load the package
library(car)
```

```
## Warning: package 'car' was built under R version 4.4.1

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.4.1

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
# Fit the model from Part B again
model_b <- lm(Y ~ X2 + X3 + X4 + X5, data = data)

# Calculate VIF for each predictor
vif_values <- vif(model_b)
print(vif_values)
```

```
##       X2       X3       X4       X5
## 1.104100 1.205604 1.729458 1.837637
```

```r
# Calculate and print average VIF
avg_vif <- mean(vif_values)
cat("\nAverage VIF:", round(avg_vif, 3), "\n")
```

```
##
## Average VIF: 1.469
```

```r
# Interpret multicollinearity
if (any(vif_values > 10)) {
  cat("Conclusion: At least one predictor has VIF > 10, indicating serious multicollinearity.\n")
} else if (any(vif_values > 5)) {
  cat("Conclusion: Some predictors have VIF > 5, suggesting moderate multicollinearity.\n")
} else {
  cat("Conclusion: All VIFs are below 5. There is no evidence of problematic multicollinearity.\n")
}
```

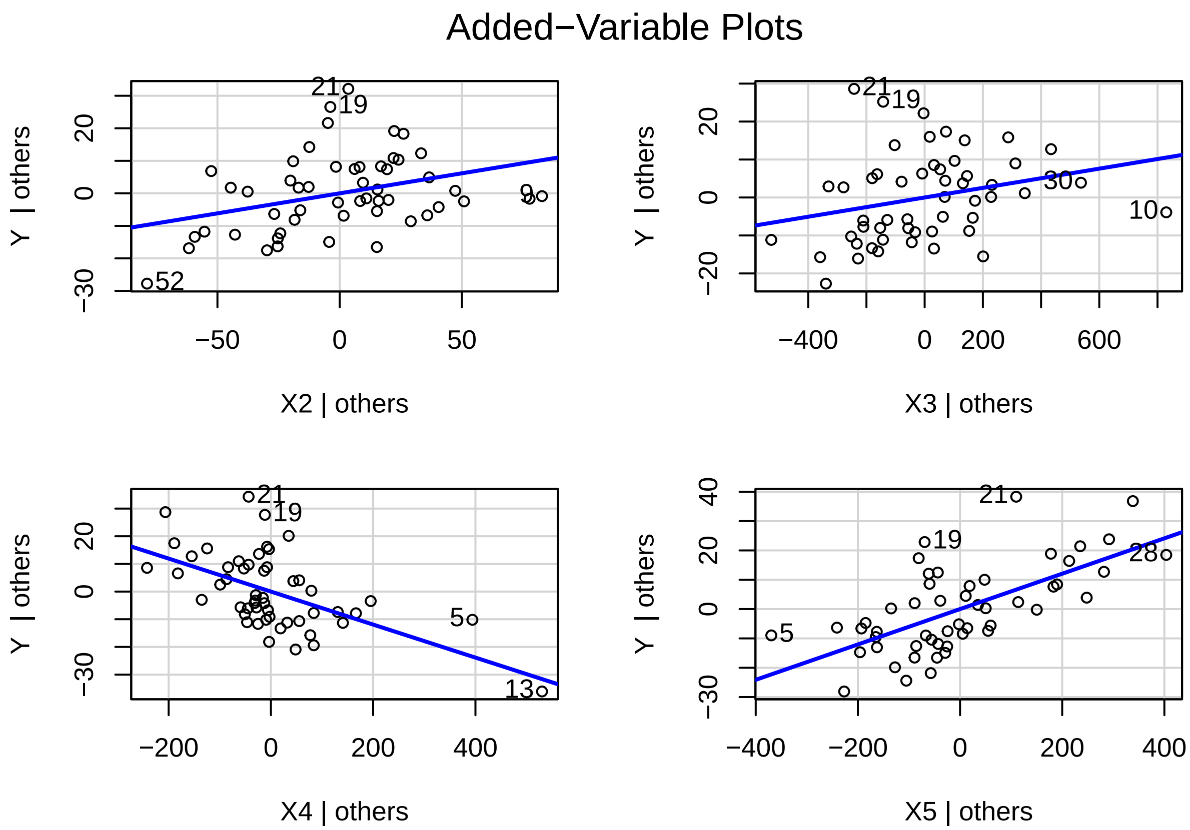
```
## Conclusion: All VIFs are below 5. There is no evidence of problematic multicollinearity.
```

Part H

```r
### Part H: Added Variable Plots (AV Plots)
library(car)

# Fit model from part (b)
model_b <- lm(Y ~ X2 + X3 + X4 + X5, data = data)

# Create Added Variable Plots for each predictor in the model
avPlots(model_b, ask = FALSE)
```



Problem 2

Part A

```r
### Part A
data <- read.csv("hw08pr02.csv", header = TRUE, sep = ",")

# Fit the simple linear regression model (corrected object name)
fit2 <- lm(Y ~ X, data = data)

# View summary of model
summary(fit2)
```

```
##
## Call:
## lm(formula = Y ~ X, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.025  -9.816  -5.578  16.194  38.303
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.8712     9.6658   1.125    0.274
## X             0.1081     0.0119   9.083 1.55e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.31 on 20 degrees of freedom
## Multiple R-squared:  0.8049, Adjusted R-squared:  0.7951
## F-statistic:  82.5 on 1 and 20 DF,  p-value: 1.555e-08
```

```r
# Extract and print the fitted equation
coefs2 <- coef(fit2)
cat("Fitted Equation:\n")
```

```
## Fitted Equation:
```

```r
cat("Ŷ =", round(coefs2[1], 5), "+", round(coefs2[2], 5), "*X\n")
```

```
## Ŷ = 10.87115 + 0.10812 *X
```

Part B

```r
### Part B: Modified Levene Test for Non-Constant Variance

# Load data (adjust if you saved under another name)
data2 <- read.csv("hw08pr02.csv", header = TRUE)

# Fit the linear model
model <- lm(Y ~ X, data = data2)

# Get absolute residuals
abs_resid <- abs(resid(model))

# Split into two groups based on median of X
median_x <- median(data2$X)
group <- ifelse(data2$X <= median_x, "Group1", "Group2")

# Run two-sample t-test on absolute residuals
t_test <- t.test(abs_resid[group == "Group1"], abs_resid[group == "Group2"])

# Display hypotheses and results
cat("=== Modified Levene Test ===\n")
```

```
## === Modified Levene Test ===
```

```r
cat("Null Hypothesis (H0): Equal error variances between groups.\n")
```

```
## Null Hypothesis (H0): Equal error variances between groups.
```

```r
cat("Alternative Hypothesis (H1): Unequal error variances between groups.\n\n")
```

```
## Alternative Hypothesis (H1): Unequal error variances between groups.
```

```r
# Display test output
print(t_test)
```

```
##
##  Welch Two Sample t-test
##
## data:  abs_resid[group == "Group1"] and abs_resid[group == "Group2"]
## t = -3.3362, df = 15.507, p-value = 0.00434
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -21.669437  -4.804158
## sample estimates:
## mean of x mean of y
##  9.267934 22.504732
```

```r
# Manually extract and interpret
t_val <- round(t_test$statistic, 4)
df_val <- t_test$parameter
p_val <- round(t_test$p.value, 6)
crit_val <- qt(0.975, df_val)  # two-tailed test, alpha = 0.05

cat("\nCritical t-value (two-tailed, df =", df_val, "):", round(crit_val, 3), "\n")
```

```
##
## Critical t-value (two-tailed, df = 15.5074 ): 2.125
```

```r
if (abs(t_val) > crit_val) {
  cat("Conclusion: Reject H0. There is evidence of heteroscedasticity.\n")
} else {
  cat("Conclusion: Fail to reject H0. No evidence of heteroscedasticity.\n")
}
```

```
## Conclusion: Reject H0. There is evidence of heteroscedasticity.
```

Part C

```r
### Part C: Weighted Least Squares (WLS)

# Step 1: Fit the original model
model_ols <- lm(Y ~ X, data = data2)

# Step 2: Compute squared residuals
resid_sq <- resid(model_ols)^2

# Step 3: Compute weights as inverse of squared residuals
weights <- 1 / resid_sq

# Step 4: Fit WLS model using these weights
model_wls <- lm(Y ~ X, data = data2, weights = weights)

# Step 5: View WLS summary
summary(model_wls)
```

```
##
## Call:
## lm(formula = Y ~ X, data = data2, weights = weights)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8453 -0.6901 -0.3842  1.1876  1.4545
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.437535   3.093161   3.374  0.00301 **
## X            0.102937   0.006228  16.528 3.97e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9379 on 20 degrees of freedom
## Multiple R-squared:  0.9318, Adjusted R-squared:  0.9284
## F-statistic: 273.2 on 1 and 20 DF,  p-value: 3.97e-13
```

```r
# Step 6: Extract and print fitted equation
coefs_wls <- coef(model_wls)
cat("WLS Fitted Equation:\n")
```

```
## WLS Fitted Equation:
```

```r
cat("Ŷ =", round(coefs_wls[1], 5), "+", round(coefs_wls[2], 5), "*X\n")
```

```
## Ŷ = 10.43753 + 0.10294 *X
```

Part D

```r
# Step 1: Plot the raw data
plot(data2$X, data2$Y,
     main = "OLS vs WLS Regression Lines",
     xlab = "X (Total Hours Worked)",
     ylab = "Y (Revenue in $1000s)",
     pch = 16)

# Step 2: Add OLS regression line (from Part A)
abline(model_ols, col = "blue", lwd = 2)

# Step 3: Add WLS regression line (from Part C)
abline(model_wls, col = "red", lty = 2, lwd = 2)

# Step 4: Add legend
legend("topleft",
       legend = c("OLS Fit", "WLS Fit"),
       col = c("blue", "red"),
       lty = c(1, 2),
       lwd = 2)
```