Homework # 3

STA 4210

Due: Friday, February 14

The submission must be typed. Show all work. Print all input and output code if using software to solve. Points will be deducted for insufficient work or missing steps to solving the problems.

Use `R` to answer the following questions. You may use `R` functions to find the information necessary to answer the problems. You do not need to solve any of these problems manually. All data files can be found here:

`https://people.clas.ufl.edu/dlindberg/sta-4210/`

Problem 1. A baseball statistician wonders if the number of games a Major League Baseball (MLB) team wins can be predicted by the average age of the players on that team. At the conclusion of the 2024 MLB season, the number of wins and average age for the players on the team were collected for all 30 MLB teams.

(a) Fit a simple linear regression used to predict wins (Y) from the average player age (X). State the fitted equation. Interpret the slope and the intercept (if possible) in the context of the problem.

(b) Conduct an F test to test whether or not $\beta_1 = 0$. Let $\alpha = 0.05$. State the null and alternative hypotheses, the test statistic, the p-value, α, and state your conclusion.

(c) Create a Residual Plot where e_i is on the Y-axis and average player age is on the X-axis. Give the plot a title and label the axes.

(d) Create a Standardized (Studentized) Residual Plot where e_i^* is on the Y-axis and average player age is on the X-axis. Give the plot a title and label the axes. Recall, $e_i^* = e_i/\sqrt{MSE}$. Add dotted horizontal lines at $e_i^* = 3$ and $e_i^* = -3$.

(e) Identify one of the violations from the Standardized Residual Plot.

(f) If you found a violation in part (e), use one of the remedies discussed in class and redo parts (a)-(d). Did this resolve the violation identified in part (e)? Also compare the conclusions made from the hypothesis test in part (b) for each scenario.

Problem 2. An experiment was conducted to determine the relation of area of wires (1/100,000 in) of couplings and the deflection of galvonometer in explosion experiments. The researchers were interested in determining whether there is a linear relationship between these two variables.

(a) Fit a simple linear regression used to predict deflection (Y) from the area (X). State the fitted equation.

(b) Create a Residual Plot where e_i is on the Y-axis and area is on the X-axis. Give the plot a title and label the axes.

(c) Identify one of the violations from the Residual Plot.

(d) Use the `boxcox()` function to identify a transformation. Clearly state the transformation function and which variable it's applied to. Use the most appropriate transformation from the notes (i.e. choose a value for λ from the ones listed in the notes that is closest to the calculated $\hat{\lambda}$). You'll need to load the `MASS` library.

(e) Make the transformation from part (d) and redo parts (a) and (b) using the transformed Y as the outcome. Did this resolve the violation identified in part (c)?

Problem 3. Data was collected for one independent variable X and an outcome Y. Assume a simple linear regression model to determine the relationship between X and Y.

(a) Produce a Q-Q Plot on the Outcome Y. What is this plot used to determine? Make a comment on your observations of this plot and if you notice any violations.

(b) Use the `boxcox()` function to identify a transformation. Clearly state the transformation function and which variable it's applied to. Use the most appropriate transformation from the notes (i.e. choose a value for λ from the ones listed in the notes that is closest to the calculated $\hat{\lambda}$). You'll need to load the `MASS` library.

(c) Make the transformation from part (b) and redo part (a) using the transformed Y as the outcome. Did this resolve the violation identified?

Azreen Haque

2/14/2025

Solutions

Problem 1

Solutions below.

Part A

```
data <- read.csv("hw03pr01.csv", header = TRUE, sep = ",")
colnames(data)
```

```
## [1] "Team"   "AvgAge" "Wins"
```

```
model <- lm(Wins ~ AvgAge, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = Wins ~ AvgAge, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.484  -3.929   2.009   7.622  17.676
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -20.237     58.312  -0.347   0.7312
## AvgAge         3.623      2.086   1.737   0.0934 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.06 on 28 degrees of freedom
## Multiple R-squared:  0.09725,    Adjusted R-squared:  0.06501
## F-statistic: 3.016 on 1 and 28 DF,  p-value: 0.09342
```

```
# Extract coefficients
intercept <- coef(model)[1]  # 0 (Intercept)
slope <- coef(model)[2]       # 1 (Slope)

# Print the fitted regression equation
cat("The fitted regression equation is:\n")
```

```
## The fitted regression equation is:
```

```
cat("Wins(Y) =", round(intercept, 2), "+", round(slope, 2), "* AvgAge(X)\n")
```

```
## Wins(Y) = -20.24 + 3.62 * AvgAge(X)
```

```
# Interpretation
cat("Interpretation of Slope:\n")
```

```
## Interpretation of Slope:
```

```
cat("Each additional year in AvgAge increases wins by", slope, "games.\n\n")
```

```
## Each additional year in AvgAge increases wins by 3.623029 games.
```

```
cat("Interpretation of Intercept:\n")
```

```
## Interpretation of Intercept:
```

```
cat("If AvgAge = 0, predicted wins =", intercept, ".\n")
```

```
## If AvgAge = 0, predicted wins = -20.23662 .
```

Part B

```
anova_results <- anova(model)
print(anova_results)
```

```
## Analysis of Variance Table
##
## Response: Wins
##           Df Sum Sq Mean Sq F value  Pr(>F)
## AvgAge     1  438.5  438.51  3.0164 0.09342 .
## Residuals 28 4070.5  145.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# State hypotheses
cat("Hypothesis Test for F-statistic:\n")
```

```
## Hypothesis Test for F-statistic:
```

```
cat("H0:  1 = 0 (AvgAge does NOT affect Wins)\n")
```

```
## H0:  1 = 0 (AvgAge does NOT affect Wins)
```

```
cat("HA:  1   0 (AvgAge DOES affect Wins)\n\n")
```

```
## HA:  1   0 (AvgAge DOES affect Wins)
```

```r
f_stat <- anova_results$`F value`[1]
p_value <- anova_results$`Pr(>F)`[1]
cat("F-statistic:", round(f_stat, 3), "\n")
```

```
## F-statistic: 3.016
```

```r
cat("p-value:", round(p_value, 5), "\n")
```

```
## p-value: 0.09342
```

```r
alpha <- 0.05

if (p_value < alpha) {
  cat("Conclusion: Reject H0. AvgAge significantly predicts Wins.\n")
} else {
  cat("Conclusion: Fail to reject H0. No significant relationship.\n")
}
```

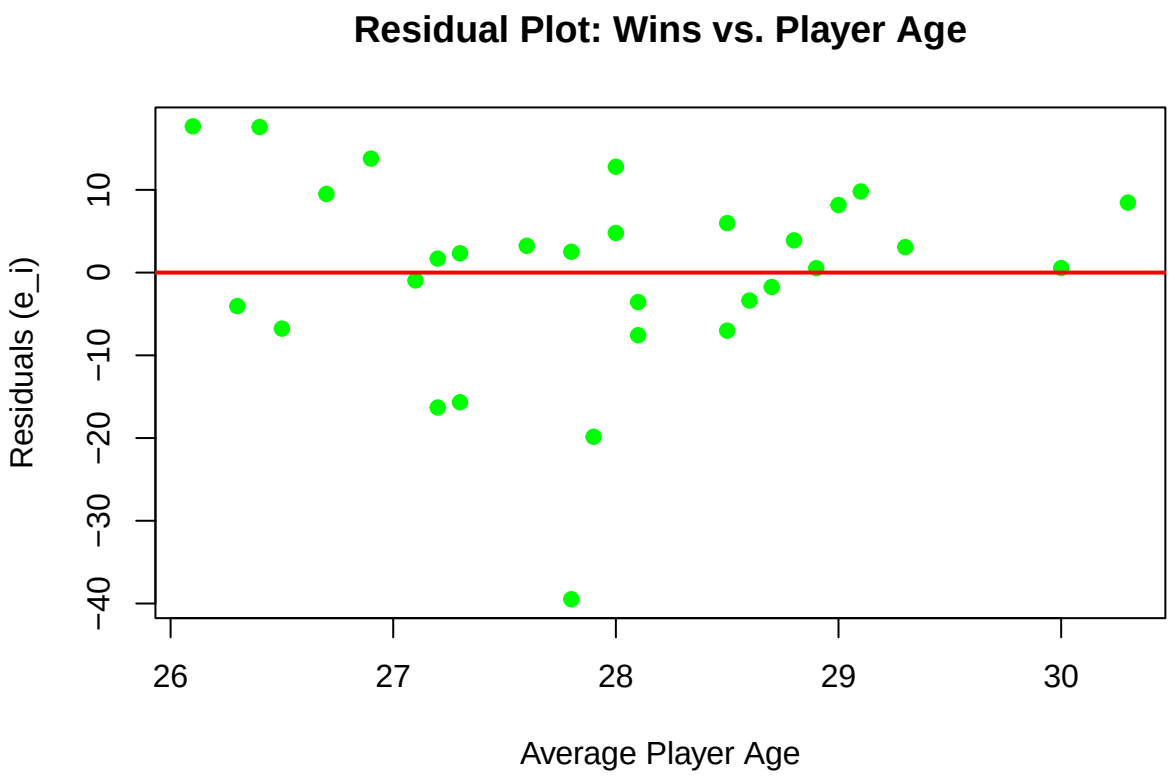
```
## Conclusion: Fail to reject H0. No significant relationship.
```

Part C

```r
# Compute residuals
residuals <- resid(model)

# Create the residual plot
plot(data$AvgAge, residuals,
     main = "Residual Plot: Wins vs. Player Age",
     xlab = "Average Player Age",
     ylab = "Residuals (e_i)",
     pch = 19, col = "green")

# Add a horizontal line at y = 0
abline(h = 0, col = "red", lwd = 2)
```



Part D

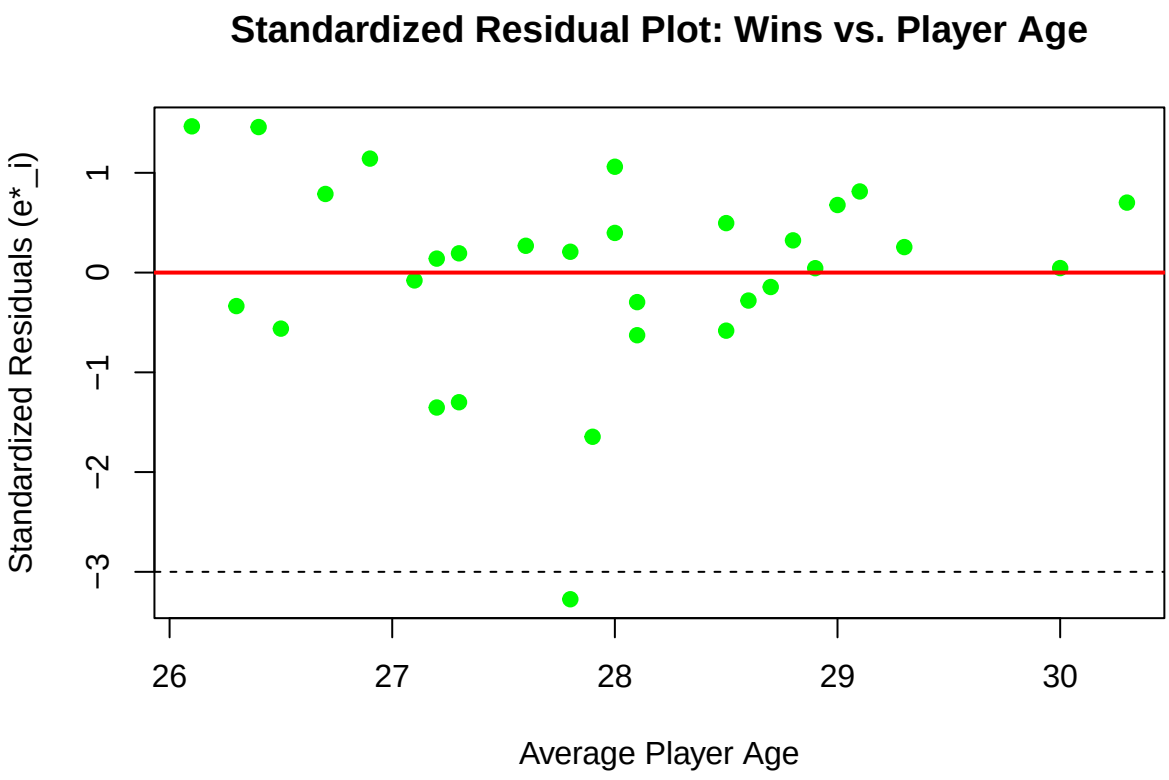
```r
# Compute MSE (Mean Squared Error)
mse <- sum(residuals^2) / df.residual(model)

# Compute standardized residuals
std_residuals <- residuals / sqrt(mse)

# Plot standardized residuals vs. AvgAge
plot(data$AvgAge, std_residuals,
     main = "Standardized Residual Plot: Wins vs. Player Age",
     xlab = "Average Player Age",
     ylab = "Standardized Residuals (e*_i)",
     pch = 19, col = "green")

# Add horizontal reference line at y = 0
abline(h = 0, col = "red", lwd = 2)

# Add dotted horizontal lines at e*_i = ±3
abline(h = 3, col = "black", lty = 2)
abline(h = -3, col = "black", lty = 2)
```



Part E

```
cat("Violation 4: Model fits all but one outlying observation\n")
```

```
## Violation 4: Model fits all but one outlying observation
```

Part F

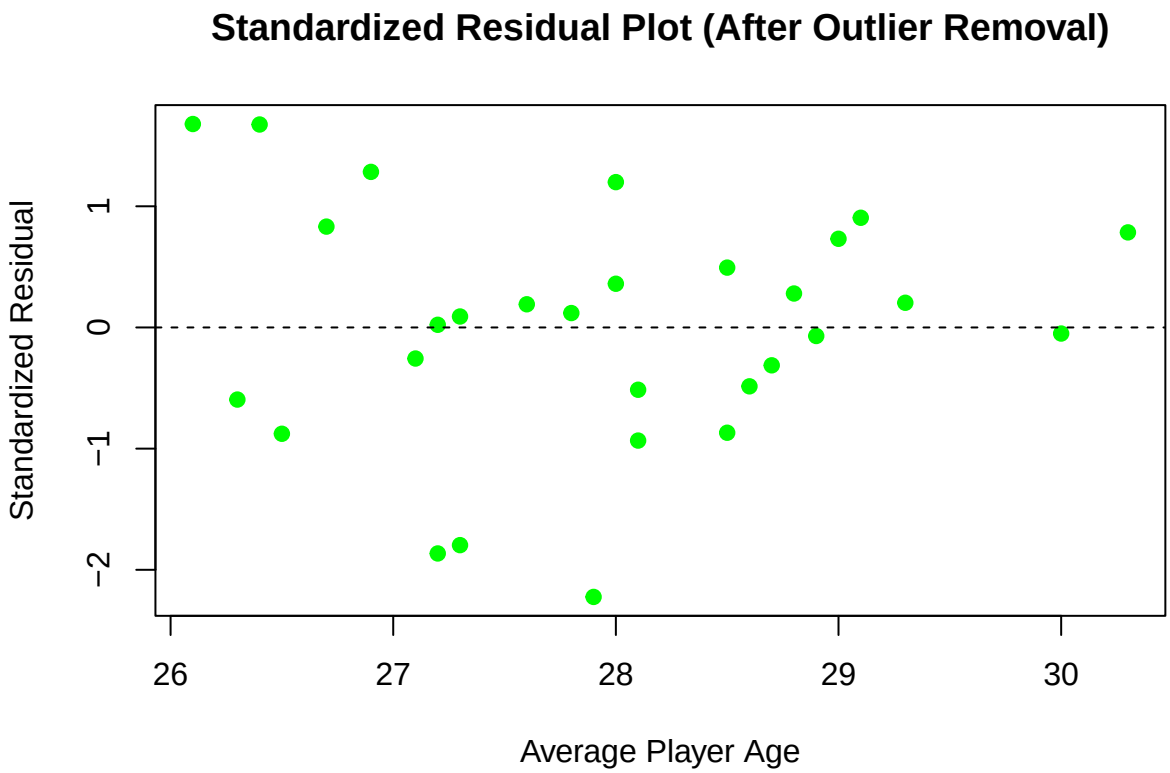
```
# Identify non-outlier data points based on standardized residuals
data_clean <- data[abs(resid(model) / summary(model)$sigma) < 3, ]

# Refit the regression model without outliers
model_clean <- lm(Wins ~ AvgAge, data = data_clean)

# Plot Standardized Residuals
plot(data_clean$AvgAge, resid(model_clean) / summary(model_clean)$sigma,
     main = "Standardized Residual Plot (After Outlier Removal)",
     ylab = "Standardized Residual",
     xlab = "Average Player Age",
     pch = 19, col = "green")

# Add reference lines
```

```r
abline(a = 0, b = 0, lty = 2)  # Zero reference line
abline(a = -3, b = 0, lty = 3) # Lower bound (-3)
abline(a = 3, b = 0, lty = 3)  # Upper bound (+3)
```




```r
# Run ANOVA for both models
anova_original <- anova(model)
anova_clean <- anova(model_clean)

# Print F-statistics and p-values for comparison
cat("Before Removing Outliers:\n")
```

```
## Before Removing Outliers:
```

```r
cat("F-statistic:", round(anova_original$`F value`[1], 3), "\n")
```

```
## F-statistic: 3.016
```

```r
cat("p-value:", round(anova_original$`Pr(>F)`[1], 5), "\n")
```

```
## p-value: 0.09342
```

```r
cat("\nAfter Removing Outliers:\n")
```

```
##
## After Removing Outliers:
```

```r
cat("F-statistic:", round(anova_clean$`F value`[1], 3), "\n")
```

```
## F-statistic: 4.392
```

```r
cat("p-value:", round(anova_clean$`Pr(>F)`[1], 5), "\n")
```

```
## p-value: 0.04561
```

```r
# Compare hypothesis test decision
if (round(anova_clean$`Pr(>F)`[1], 5) < 0.05) {
  cat("Conclusion: Reject H0. AvgAge significantly predicts Wins.\n")
} else {
  cat("Conclusion: Fail to reject H0. No significant relationship.\n")
}
```

```
## Conclusion: Reject H0. AvgAge significantly predicts Wins.
```

```r
cat("The violation was resolved leading to a corrected conclusion.\n")
```

```
## The violation was resolved leading to a corrected conclusion.
```

Problem 2

Part A

```r
data <- read.csv("hw03pr02.csv", header = TRUE, sep = ",")
colnames(data)
```

```
## [1] "Area"       "Deflection"
```

```r
model <- lm(Deflection ~ Area, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = Deflection ~ Area, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.318  -7.000  -1.821   6.244  16.247
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 187.5259     3.0966   60.56   <2e-16 ***
## Area         -0.6925     0.0333  -20.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.747 on 25 degrees of freedom
## Multiple R-squared:  0.9454, Adjusted R-squared:  0.9432
## F-statistic: 432.6 on 1 and 25 DF,  p-value: < 2.2e-16
```

```r
intercept <- coef(model)[1]
slope <- coef(model)[2]
cat("The fitted regression equation is:\n")
```

```
## The fitted regression equation is:
```

```r
cat("Deflection(Y) =", round(intercept, 2), "+", round(slope, 2), "* Area(X)\n")
```

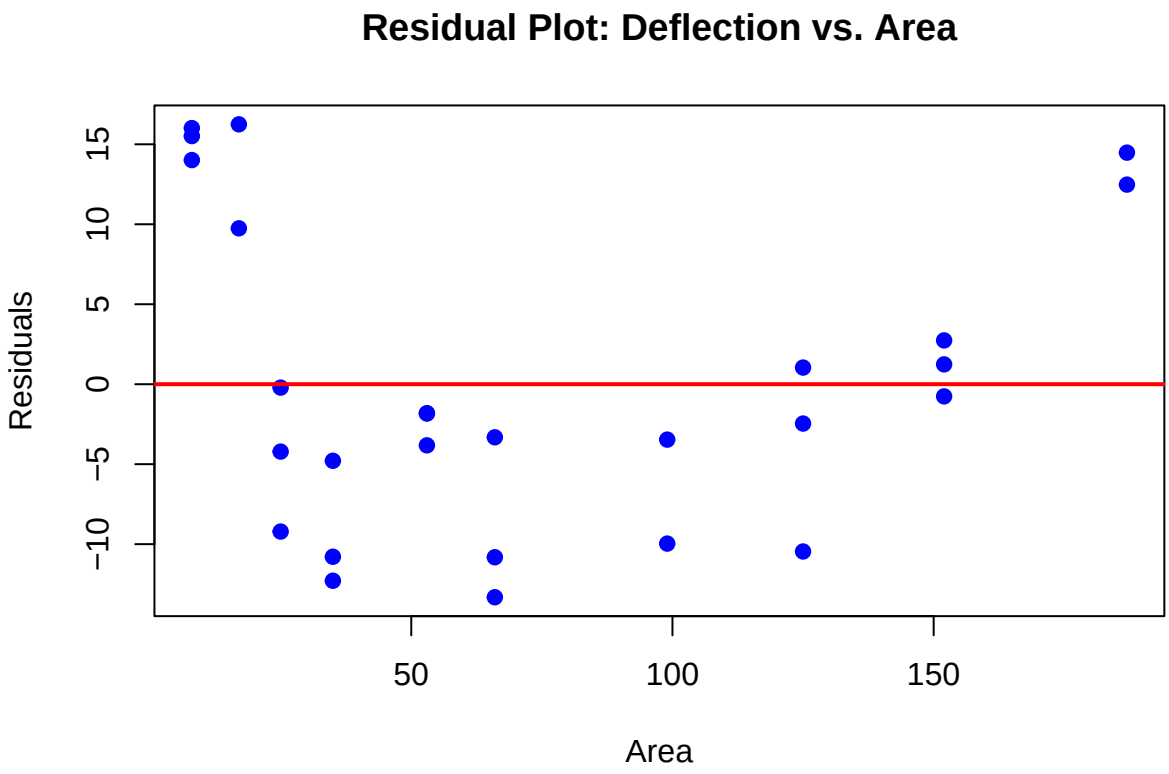
```
## Deflection(Y) = 187.53 + -0.69 * Area(X)
```

Part B

```r
# Compute residuals
residuals <- resid(model)

# Create residual plot
plot(data$Area, residuals,
     main = "Residual Plot: Deflection vs. Area",
     xlab = "Area",
     ylab = "Residuals",
     pch = 19, col = "blue")

# Add a horizontal line at y = 0
abline(h = 0, col = "red", lwd = 2)
```



Part C

```
cat("Violation 1: The regression function is not linear.")
```

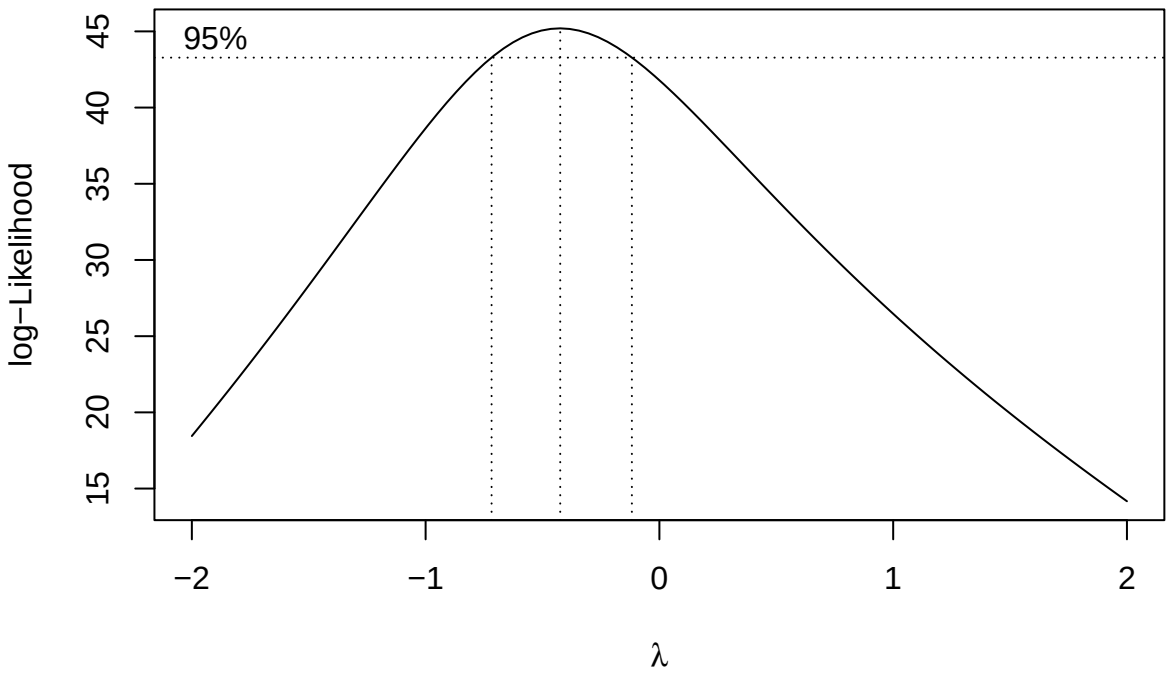
```
## Violation 1: The regression function is not linear.
```

Part D

```
# Load necessary library
library(MASS)

# Fit the initial linear model
model <- lm(Deflection ~ Area, data = data)

# Apply Box-Cox transformation to find optimal lambda
boxcox_result <- boxcox(model, lambda = seq(-2, 2, by = 0.1))  # Generates Box-Cox plot
```




```r
# Find the optimal lambda (maximizing log-likelihood)
lambda_opt <- boxcox_result$x[which.max(boxcox_result$y)]
cat("Optimal Lambda:", lambda_opt, "\n")
```

```
## Optimal Lambda: -0.4242424
```

```r
# Select the closest transformation from the notes
if (lambda_opt >= 1.9 & lambda_opt <= 2.1) {
    data$Deflection_Transformed <- data$Deflection^2  # Y^2
    cat("Chosen Transformation: Y^2\n")
} else if (lambda_opt >= 0.4 & lambda_opt <= 0.6) {
    data$Deflection_Transformed <- sqrt(data$Deflection)  # sqrt(Y)
    cat("Chosen Transformation: sqrt(Y)\n")
} else if (lambda_opt >= -0.1 & lambda_opt <= 0.1) {
    data$Deflection_Transformed <- log(data$Deflection + abs(min(data$Deflection)) + 0.01)  # log(Y)
    cat("Chosen Transformation: log(Y)\n")
} else if (lambda_opt >= -0.6 & lambda_opt <= -0.4) {
    data$Deflection_Transformed <- 1 / sqrt(data$Deflection)  # 1/sqrt(Y)
    cat("Chosen Transformation: 1/sqrt(Y)\n")
} else if (lambda_opt >= -1.1 & lambda_opt <= -0.9) {
    data$Deflection_Transformed <- 1 / data$Deflection  # 1/Y
    cat("Chosen Transformation: 1/Y\n")
} else {
    cat("Chosen Transformation: No common transformation found\n")
}
```

```
## Chosen Transformation: 1/sqrt(Y)
```

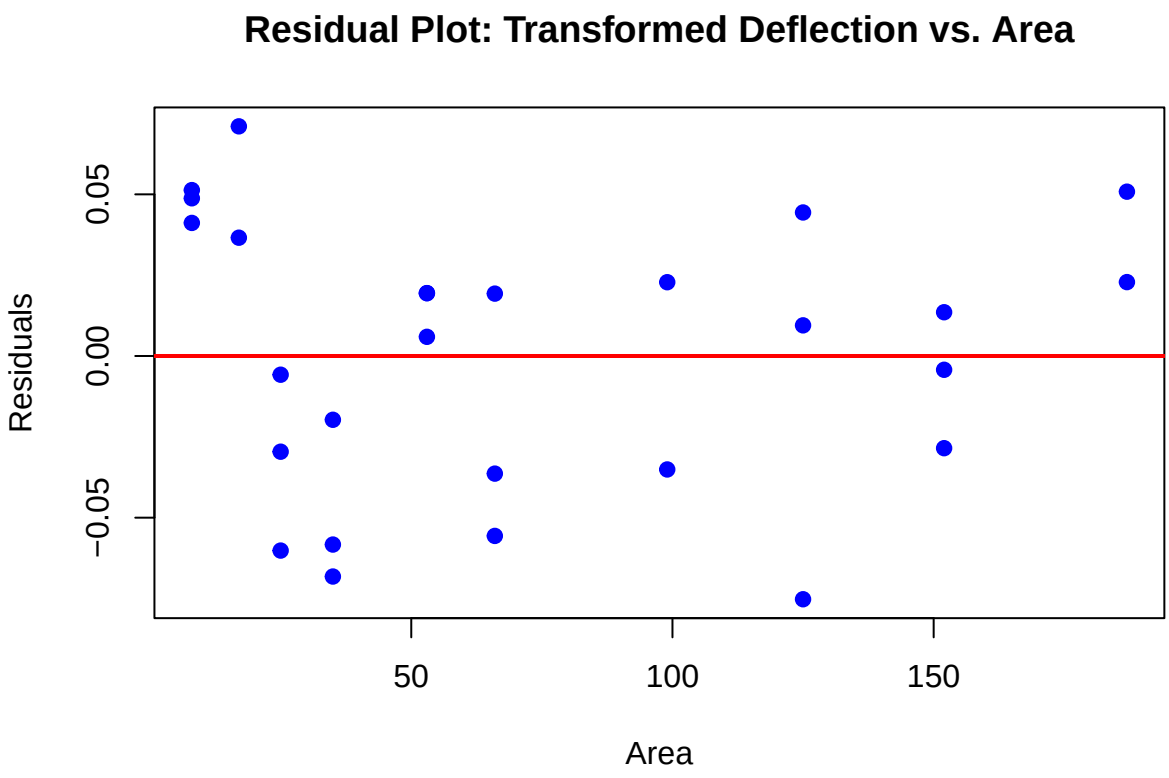
```
# Fit the new model with the transformed Y (if applicable)
if (exists("data$Deflection_Transformed")) {
    model_transformed <- lm(Deflection_Transformed ~ Area, data = data)
    summary(model_transformed)
}
```

Part E

```
data$Transformed_Deflection <- log(data$Deflection)
# Fit the linear model with transformed Y
model_transformed <- lm(Transformed_Deflection ~ data$Area, data = data)
summary(model_transformed)
```

```
##
## Call:
## lm(formula = Transformed_Deflection ~ data$Area, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.075230 -0.032344  0.009477  0.029723  0.071031
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.2818345  0.0135154   390.8   <2e-16 ***
## data$Area   -0.0056100  0.0001453   -38.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04254 on 25 degrees of freedom
## Multiple R-squared:  0.9835, Adjusted R-squared:  0.9828
## F-statistic:  1490 on 1 and 25 DF,  p-value: < 2.2e-16
```

```
# Create the residual plot for transformed Y
plot(data$Area, resid(model_transformed),
     main = "Residual Plot: Transformed Deflection vs. Area",
     xlab = "Area", ylab = "Residuals",
     col = "blue", pch = 19)
abline(h = 0, col = "red", lwd = 2)
```




```
# Check if the violation is resolved
cat("Check residual plot: the spread is more random, transformation was effective.\n")
```

```
## Check residual plot: the spread is more random, transformation was effective.
```

Problem 3

Part A

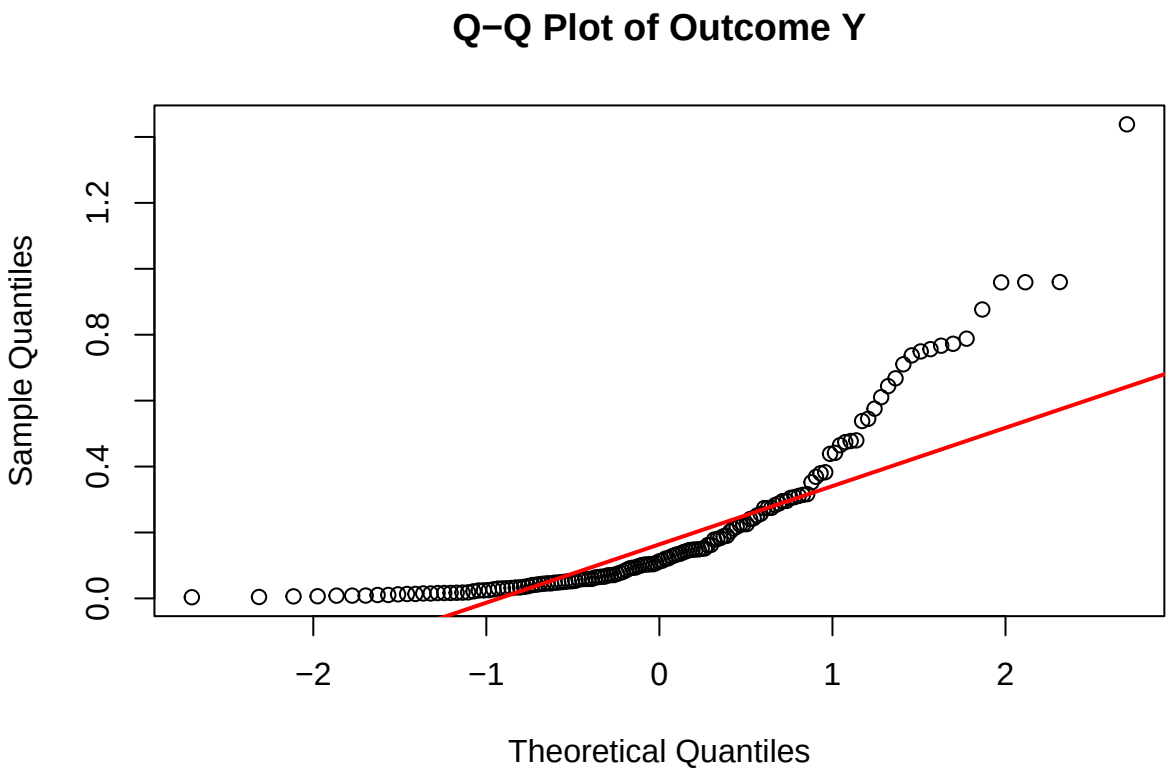
```
# Load necessary library
library(ggplot2)

# Load the dataset
data <- read.csv("hw03pr03.csv", header = TRUE, sep = ",")

colnames(data)
```

```
## [1] "X" "x" "y"
```

```
# Generate a Q-Q plot for the outcome variable (ensure correct column name)
qqnorm(data$y, main = "Q-Q Plot of Outcome Y")
qqline(data$y, col = "red", lwd = 2)  # Adds a reference line
```




```
# observations
cat("Observations:\n",
    "- The Q-Q plot is used to assess whether the residuals follow a normal distribution.\n",
    "- If the points lie along the straight red line, the residuals are normally distributed.\n",
    "- In this case, the points deviate significantly from the line, especially at the tails, indicatin
    "- This suggests that a transformation might be necessary to make the residuals more normal.\n")
```

```
## Observations:
##  - The Q-Q plot is used to assess whether the residuals follow a normal distribution.
##  - If the points lie along the straight red line, the residuals are normally distributed.
##  - In this case, the points deviate significantly from the line, especially at the tails, indicating
##  - This suggests that a transformation might be necessary to make the residuals more normal.
```

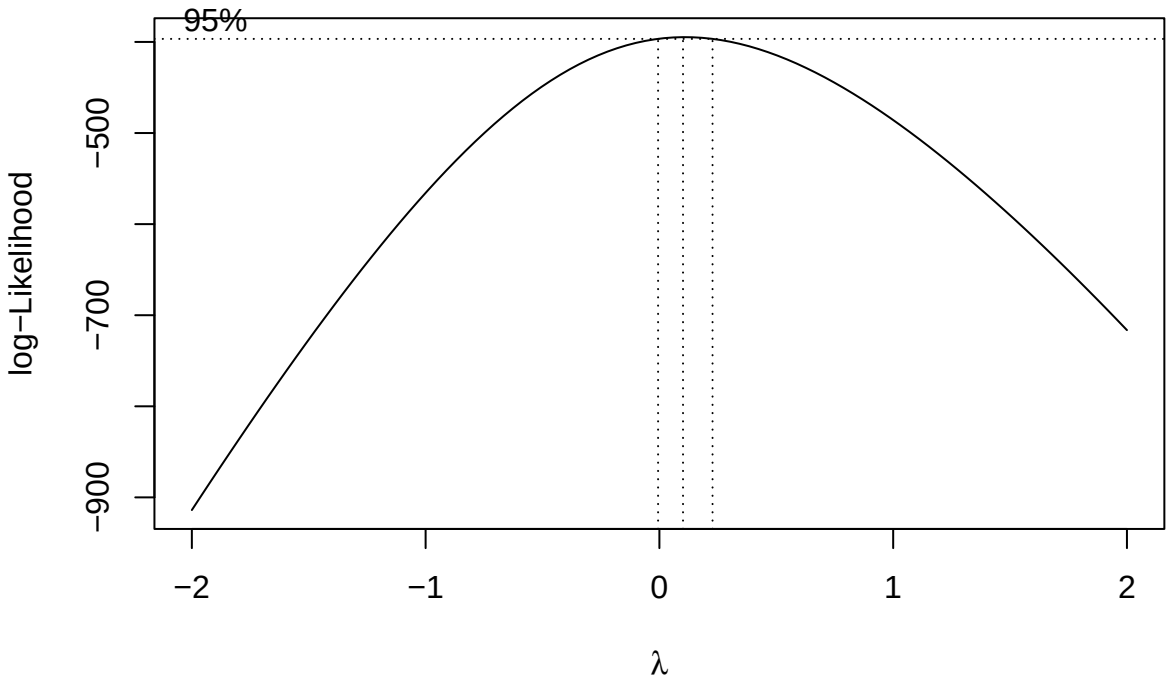
Part b

```
# Load necessary library
library(MASS)

# Fit the linear model
# Fit the linear model
model <- lm(y ~ x, data = data)

# Perform the Box-Cox transformation
```

```r
library(MASS)
boxcox_result <- boxcox(model, lambda = seq(-2, 2, by = 0.1))
```




```r
# Find the optimal lambda
lambda_optimal <- boxcox_result$x[which.max(boxcox_result$y)]

# Print the transformation applied
cat("The optimal lambda ( ) found is:", lambda_optimal, "\n")
```

```
## The optimal lambda ( ) found is: 0.1010101
```

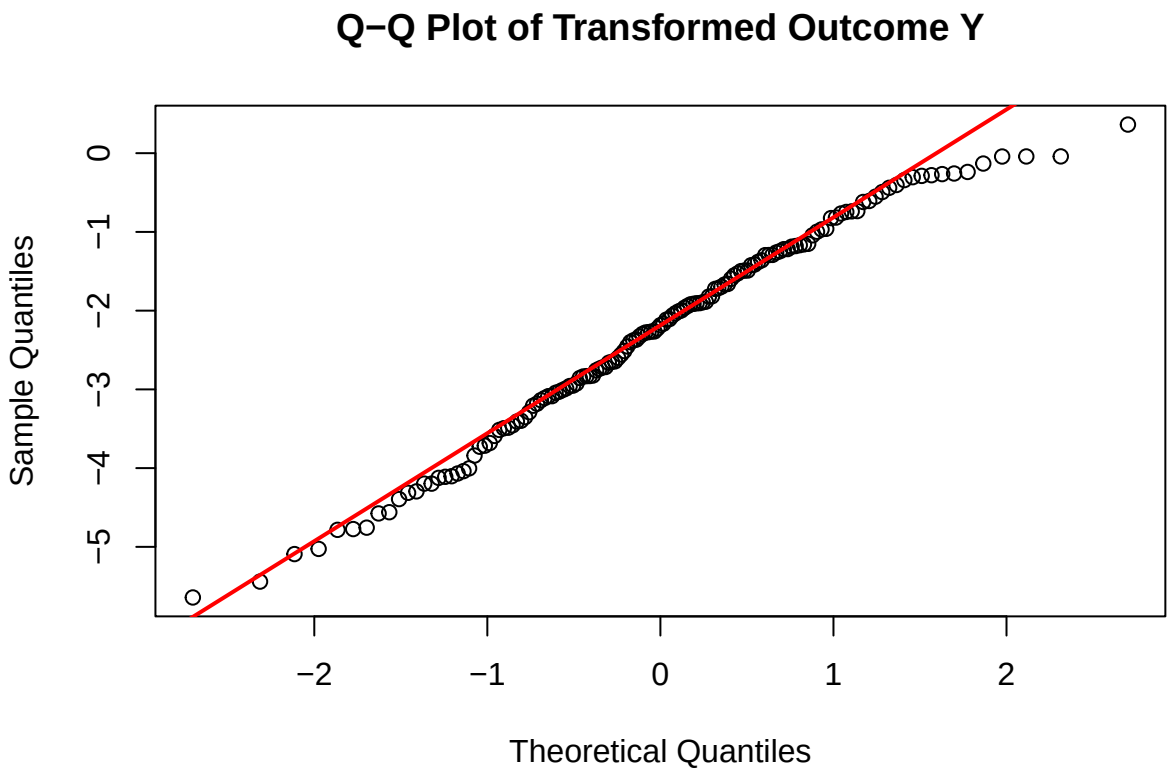
```r
# Apply the transformation based on the closest lambda value
if (lambda_optimal > 1.5) {
    data$y_transformed <- data$y^2
    cat("Transformation applied: Y' = Y^2\n")
} else if (lambda_optimal > 0.25) {
    data$y_transformed <- sqrt(data$y)
    cat("Transformation applied: Y' = sqrt(Y)\n")
} else if (lambda_optimal > -0.25) {
    data$y_transformed <- log(data$y)
    cat("Transformation applied: Y' = log(Y)\n")
} else if (lambda_optimal > -0.75) {
    data$y_transformed <- 1 / sqrt(data$y)
    cat("Transformation applied: Y' = 1 / sqrt(Y)\n")
```

```
} else {
    data$y_transformed <- 1 / data$y
    cat("Transformation applied: Y' = 1 / Y\n")
}
```

```
## Transformation applied: Y' = log(Y)
```

```
# Apply the transformation to Y
data$Y_transformed <- log(data$y)

# Generate Q-Q plot for transformed Y
qqnorm(data$Y_transformed, main = "Q-Q Plot of Transformed Outcome Y")
qqline(data$Y_transformed, col = "red", lwd = 2)
```




```
# Print observation
cat("The Q-Q plot for the transformed outcome Y now aligns more closely with the normality assumption.
The data points are much closer to the diagonal reference line compared to the original Q-Q plot,
suggesting that the transformation helped correct the non-normality issue.
There may still be minor deviations in the tails, but overall, normality is significantly improved.")
```

```
## The Q-Q plot for the transformed outcome Y now aligns more closely with the normality assumption.
## The data points are much closer to the diagonal reference line compared to the original Q-Q plot,
## suggesting that the transformation helped correct the non-normality issue.
## There may still be minor deviations in the tails, but overall, normality is significantly improved.
```