Experimental Design and Analysis
HW02 Solution

Problem 1

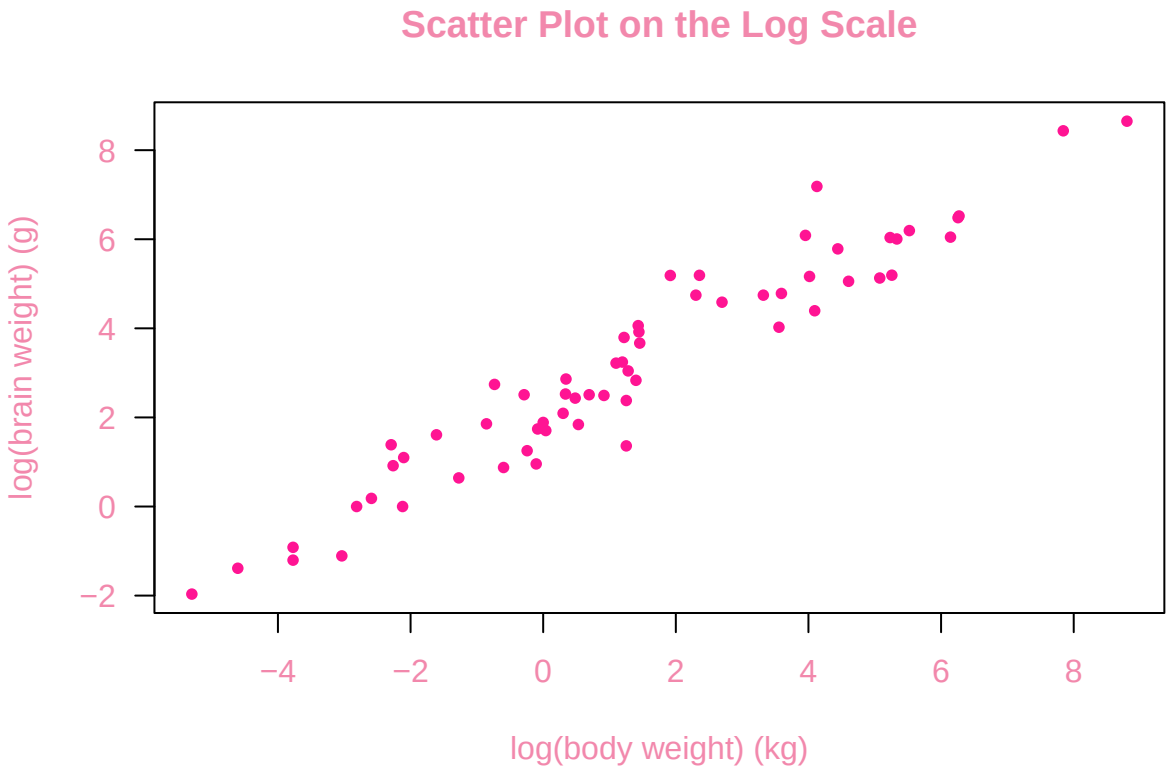
(a)

Draw a scatter plot of **brain weight** *versus* **body weight**.

The upper-right zoomed-in scatter plot indicates a positive relationship between the two variables.

(b)

Take log-transformation of both the variables, and plot **log(body weight)** *versus* **log(brain weight)**.



The log-scale scatter plot reveals a more evident positive linear relationship between the two variables.

(c)

We fit a regression model

$$\textbf{log(brain weight)} = \beta_0 + \beta_1 \, \textbf{log(body weight)} + \varepsilon,$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

```
##
## Call:
## lm(formula = brain ~ body, data = log(data_1))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71550 -0.49228 -0.06162  0.43597  1.94829
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.13479    0.09604   22.23   <2e-16 ***
```

```
## body         0.75169    0.02846   26.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6943 on 60 degrees of freedom
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9195
## F-statistic: 697.4 on 1 and 60 DF,  p-value: < 2.2e-16
```

This model explains 92.08 %of the total variation.

(d)

Consider the regression model fitted at **part (c)** :

$$\textbf{log(brain weight)} = \beta_0 + \beta_1 \, \textbf{log(body weight)} + \varepsilon,$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

A 95 % confidence interval for $\mathbb{E}\Big\{\textbf{log(brain weight)} \,\Big|\, \textbf{log(body weight)} = x^*\Big\}$ *is*

$$\hat{\mu}^* \pm t_{(62-2,\ 0.05/2)} \ s.e.(\hat{\mu}^*),$$

where

$$\begin{cases} \hat{\mu}^* = \begin{pmatrix} 1 & x^* \end{pmatrix} \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} \\ s.e.(\hat{\mu}^*) = \hat{\sigma} \sqrt{\begin{pmatrix} 1 & x^* \end{pmatrix} \left(\mathbf{X}'\mathbf{X} \right)^{-1} \begin{pmatrix} 1 \\ x^* \end{pmatrix}} \end{cases}$$

Since we have already obtained the coefficient estimates

$$\begin{cases} \hat{\beta}_0 = 2.1347886768 \\ \hat{\beta}_1 = 0.7516859362 \\ \hat{\sigma} = 0.694294731 \end{cases},$$

we can manually compute the desired confidence interval:

```r
options(digits = 12)

x_star <- log(250)
X <- model.matrix(fit_1)
beta_0_hat <- fit_1$coefficients[1]
beta_1_hat <- fit_1$coefficients[2]
sigma_hat  <- sqrt(sum((fit_1$residuals)^2) / fit_1$df.residual)
# or sigma_hat <- summary(fit_1)$sigma

mu_star_hat <- t(c(1, x_star)) %*% c(beta_0_hat, beta_1_hat)
SE_mu_star_hat <- sigma_hat * sqrt(t(c(1, x_star)) %*% solve(t(X) %*% X) %*% c(1, x_star))
```

```r
exp(c(mu_star_hat - SE_mu_star_hat * qt(0.05/2, 62-2, lower.tail = FALSE),
      mu_star_hat + SE_mu_star_hat * qt(0.05/2, 62-2, lower.tail = FALSE)))
```

```
## [1] 398.931576369 721.690573214
```

Alternatively, we can use the built-in **R** *function to compute this confidence interval:*

```r
options(digits = 12)

exp(predict(fit_1, data.frame("body" = log(250)), interval = "confidence"))
```

```
##             fit           lwr           upr
## 1 536.567943529 398.931576369 721.690573214
```

Both methods yield the same 95 % C.I. for $\mathbb{E}\Big\{\textbf{brain weight} \,\Big|\, \textbf{body weight} = 250\ kg\Big\}$:

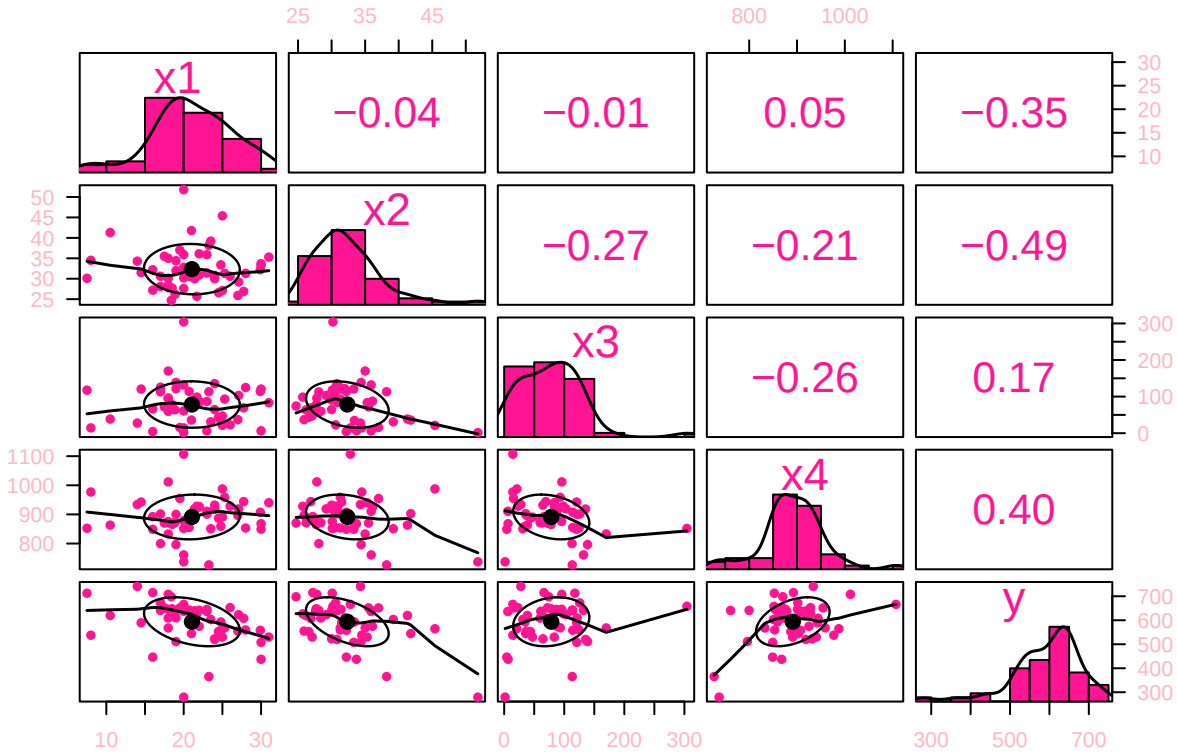
$$\Big(398.931576369,\ 721.690573214\Big)$$

$$\Big(\textit{INTENTIONALLY LEFT BLANK}\Big)$$

Problem 2

(a)

Use the **pairs.panels** *function in the* **psych** *package to draw scatter plots for every pair of variables and compute the corresponding correlations.*



The following observations can be made:

- *The pairwise correlations among the explanatory variables are all relatively small in magnitude, so severe collinearity does not appear to be a major concern.*
- *Among the predictors,* x_2 and x_4 show the strongest marginal associations with y. In particular, x_2 is moderately negatively correlated with y, whereas x_4 is moderately positively correlated with y.
- *There is a negative association between* x_1 and y, although the relationship appears weaker than those for x_2 and x_4.
- *There is no strong linear relationship between* x_3 and y. At most, the association appears to be weak.
- *For* x_2 and y, the scatter plot suggests a negative trend. Even if the lower-right outlying observation is ignored, the negative association still seems to remain.
- *For* x_4 and y, the scatter plot suggests a positive trend, although one or two relatively extreme observations may strengthen the apparent relationship.
- *Some scatter plots suggest mild curvature rather than a perfectly linear pattern, especially for the relationships involving* x_2 and possibly x_3. Thus, a simple linear model may not capture all features of the data.

- *From the histograms on the diagonal,* x_1 appears roughly symmetric, whereas x_2 and x_3 appear somewhat right-skewed. The response variable y is concentrated in the middle range, with a few relatively low observations.
- *Overall, if the predictors are ranked by their marginal linear relationships with* y, then x_2 and x_4 appear to be the most informative, x_1 appears moderately informative, and x_3 appears to be the weakest.

(b)

Fit a multiple linear regression model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \varepsilon,$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

```
##
## Call:
## lm(formula = y ~ ., data = data_2)
##
## Residuals:
##           Min            1Q        Median            3Q           Max
## -151.94217016  -30.75676480    2.44287437   41.20132557  115.26157761
##
## Coefficients:
##                  Estimate    Std. Error  t value   Pr(>|t|)
## (Intercept) 439.974326473 171.833934725  2.56046 0.01380117 *
## x1           -6.292693794   1.715472709 -3.66820 0.00063234 ***
## x2           -6.171796400   1.872585596 -3.29587 0.00189515 **
## x3            0.276618841   0.184318381  1.50077 0.14024929
## x4            0.520987879   0.149885712  3.47590 0.00112230 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.3045783 on 46 degrees of freedom
## Multiple R-squared:  0.503425422,    Adjusted R-squared:  0.460245023
## F-statistic: 11.6586563 on 4 and 46 DF,  p-value: 1.28021168e-06
```

Based on the p-values from the individual t-tests, x_1, x_2, *and* x_4 *are seen to significantly affect the response.*

(c)

The following observations can be made:

- *The multiple regression output suggests that* x_1, x_2, and x_4 have significant effects on y, while x_3 does not. The overall model is significant and explains about 50% of the variation in the response.
- *For* x_1 (state gasoline tax), the estimated coefficient is negative and highly significant. This means that, holding the other variables fixed, higher fuel tax is associated with lower gasoline consumption. This result is also consistent with the scatter plot, where x_1 and y show a negative association.

- *For* x_2 (per capita income), the estimated coefficient is also negative and significant. Although one might initially expect higher income to increase gasoline consumption, the scatter plot shows a fairly clear negative association between x_2 and y. A possible explanation is that higher-income areas may be more urbanized, more congested, or better served by public transportation, so gasoline consumption is lower.
- *For* x_3 (paved highways), the estimated coefficient is positive, but it is not statistically significant. This suggests that, after accounting for the other variables, x_3 does not provide strong additional information about y. The scatter plot also shows that the relationship between x_3 and y is weak and not clearly linear.
- *For* x_4 (licensed drivers per 1000 persons in population of 16 years olds or older), the estimated coefficient is positive and significant. This is intuitively reasonable, since a higher proportion of licensed drivers should lead to greater gasoline consumption. The scatter plot supports this conclusion, as x_4 and y show a positive association.

(d) (Supplementary)

```r
# Response and predictors
response <- "y"
predictors <- setdiff(names(data_2), response)

# 1. Fit the full model
full_model <- lm(y ~ ., data = data_2)
summary(full_model)
```

```
##
## Call:
## lm(formula = y ~ ., data = data_2)
##
## Residuals:
##           Min            1Q        Median            3Q           Max
## -151.94217016  -30.75676480    2.44287437   41.20132557  115.26157761
##
## Coefficients:
##                  Estimate    Std. Error  t value   Pr(>|t|)
## (Intercept) 439.974326473 171.833934725  2.56046 0.01380117 *
## x1           -6.292693794   1.715472709 -3.66820 0.00063234 ***
## x2           -6.171796400   1.872585596 -3.29587 0.00189515 **
## x3            0.276618841   0.184318381  1.50077 0.14024929
## x4            0.520987879   0.149885712  3.47590 0.00112230 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.3045783 on 46 degrees of freedom
## Multiple R-squared:  0.503425422,    Adjusted R-squared:  0.460245023
## F-statistic: 11.6586563 on 4 and 46 DF,  p-value: 1.28021168e-06
```

(i)

```r
# 2. Best subset selection using BIC and Cp
if (!require(leaps)) install.packages("leaps")
library(leaps)

regfit <- regsubsets(y ~ ., data = data_2, nvmax = length(predictors), method = "exhaustive")
regsum <- summary(regfit)

# Find the model with the smallest BIC
id_bic <- which.min(regsum$bic)

# Find the model with the smallest Cp
id_cp <- which.min(regsum$cp)

# Helper function: convert the model selected by regsubsets into a formula
make_formula_from_regsubsets <- function(regfit_obj, id, response = "y") {
  vars <- names(coef(regfit_obj, id = id))
  vars <- setdiff(vars, "(Intercept)")
  if (length(vars) == 0) {
    as.formula(paste(response, "~ 1"))
  } else {
    as.formula(paste(response, "~", paste(vars, collapse = " + ")))
  }
}

bic_formula <- make_formula_from_regsubsets(regfit, id_bic, response)
cp_formula  <- make_formula_from_regsubsets(regfit, id_cp, response)
bic_model <- lm(bic_formula, data = data_2)
cp_model  <- lm(cp_formula,  data = data_2)

cat("Best subset selection results\n")
```

```
## Best subset selection results
```

```r
cat("BIC-selected model:\n")
```

```
## BIC-selected model:
```

```r
print(formula(bic_model), showEnv = FALSE)
```

```
## y ~ x1 + x2 + x4
```

```r
cat("\nCp-selected model:\n")
```

```
##
## Cp-selected model:
```

```r
print(formula(cp_model), showEnv = FALSE)
```

```
## y ~ x1 + x2 + x3 + x4
```

```
cat("\nBIC values by model size:\n")
```

```
##
## BIC values by model size:
```

```
print(regsum$bic, showEnv = FALSE)
```

```
## [1]  -5.92374970531 -11.90162303777 -17.53589205464 -16.04197335341
```

```
cat("\nCp values by model size:\n");
```

```
##
## Cp values by model size:
```

```
print(regsum$cp, showEnv = FALSE)
```

```
## [1] 23.69124975189 13.20748626864  5.25229991037  5.00000000000
```

(ii)

```
# 3. Backward elimination (based on p-values)
backward_elimination_p <- function(data, response = "y", alpha_out = 0.05) {
  remaining <- setdiff(names(data), response)

  current_formula <- as.formula(
    paste(response, "~", paste(remaining, collapse = " + "))
  )
  current_model <- lm(current_formula, data = data)

  repeat {
    pvals <- summary(current_model)$coefficients[-1, 4]  # Exclude the intercept p-value

    if (length(pvals) == 0) break

    worst_p <- max(pvals, na.rm = TRUE)

    if (worst_p <= alpha_out) break

    worst_var <- names(which.max(pvals))
    remaining <- setdiff(remaining, worst_var)

    if (length(remaining) == 0) {
      current_formula <- as.formula(paste(response, "~ 1"))
    } else {
      current_formula <- as.formula(
        paste(response, "~", paste(remaining, collapse = " + "))
      )
    }
```

```
    current_model <- lm(current_formula, data = data)
  }

  return(current_model)
}

backward_model <- backward_elimination_p(data_2, response = "y", alpha_out = 0.05)

cat("Backward elimination result\n")
```

```
## Backward elimination result
```

```
print(formula(backward_model), showEnv = FALSE)
```

```
## y ~ x1 + x2 + x4
```

```
summary(backward_model)
```

```
##
## Call:
## lm(formula = current_formula, data = data)
##
## Residuals:
##          Min           1Q       Median           3Q          Max
## -161.5894726  -31.5460334   11.8648885   43.3880726  123.6589032
##
## Coefficients:
##                  Estimate    Std. Error  t value   Pr(>|t|)
## (Intercept) 560.476432774 153.931074137  3.64109 0.00067475 ***
## x1           -6.314372579   1.738114955 -3.63289 0.00069168 ***
## x2           -7.132316467   1.783106530 -3.99994 0.00022285 ***
## x4            0.445469483   0.143055099  3.11397 0.00314049 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 64.1423992 on 47 degrees of freedom
## Multiple R-squared:  0.47911162, Adjusted R-squared:  0.445863425
## F-statistic: 14.4101545 on 3 and 47 DF,  p-value: 8.6672531e-07
```

(iii)

```
# 4. Stepwise regression (based on p-values)
stepwise_selection_p <- function(data, response = "y", alpha_in = 0.05, alpha_out = 0.05) {
  all_vars <- setdiff(names(data), response)
  selected <- character(0)
  changed <- TRUE

  while (changed) {
    changed <- FALSE
    # ---------- Forward step ----------
```

```
    candidates <- setdiff(all_vars, selected)

    if (length(candidates) > 0) {
      pvals_in <- sapply(candidates, function(v) {
        vars_try <- c(selected, v)
        f_try <- as.formula(paste(response, "~", paste(vars_try, collapse = " + ")))
        fit_try <- lm(f_try, data = data)
        coef(summary(fit_try))[v, "Pr(>|t|)"]
      })

      best_var <- names(which.min(pvals_in))
      best_p   <- min(pvals_in)

      if (best_p < alpha_in) {
        selected <- c(selected, best_var)
        changed <- TRUE
      }
    }
    # ---------- Backward step ----------
    if (length(selected) > 0) {
      repeat {
        f_now <- as.formula(paste(response, "~", paste(selected, collapse = " + ")))
        fit_now <- lm(f_now, data = data)
        pvals_now <- coef(summary(fit_now))[-1, "Pr(>|t|)"]

        if (length(pvals_now) == 0) break

        worst_p <- max(pvals_now)

        if (worst_p > alpha_out) {
          worst_var <- names(which.max(pvals_now))
          selected <- setdiff(selected, worst_var)
          changed <- TRUE

          if (length(selected) == 0) break
        } else {
          break
        }
      }
    }
  }

  if (length(selected) == 0) {
    final_formula <- as.formula(paste(response, "~ 1"))
  } else {
    final_formula <- as.formula(paste(response, "~", paste(selected, collapse = " + ")))
  }

  lm(final_formula, data = data)
}

stepwise_model <- stepwise_selection_p(data_2, response = "y", alpha_in = 0.05, alpha_out = 0.05)
```

```r
cat("Stepwise regression result\n")
```

```
## Stepwise regression result
```

```r
print(formula(stepwise_model), showEnv = FALSE)
```

```
## y ~ x2 + x1 + x4
```

```r
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = final_formula, data = data)
##
## Residuals:
##          Min           1Q       Median           3Q          Max
## -161.5894726  -31.5460334   11.8648885   43.3880726  123.6589032
##
## Coefficients:
##                  Estimate    Std. Error  t value   Pr(>|t|)
## (Intercept) 560.476432774 153.931074137  3.64109 0.00067475 ***
## x2           -7.132316467   1.783106530 -3.99994 0.00022285 ***
## x1           -6.314372579   1.738114955 -3.63289 0.00069168 ***
## x4            0.445469483   0.143055099  3.11397 0.00314049 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 64.1423992 on 47 degrees of freedom
## Multiple R-squared:  0.47911162, Adjusted R-squared:  0.445863425
## F-statistic: 14.4101545 on 3 and 47 DF,  p-value: 8.6672531e-07
```

```r
# 5. Compare whether the selected models are consistent
get_terms <- function(model) attr(terms(model), "term.labels")

res <- list(
  BIC      = get_terms(bic_model),
  Cp       = get_terms(cp_model),
  Backward = get_terms(backward_model),
  Stepwise = get_terms(stepwise_model)
)

cat("Selected predictors by each method\n")
```

```
## Selected predictors by each method
```

```r
print(res)
```

```
## $BIC
## [1] "x1" "x2" "x4"
##
```

```
## $Cp
## [1] "x1" "x2" "x3" "x4"
##
## $Backward
## [1] "x1" "x2" "x4"
##
## $Stepwise
## [1] "x2" "x1" "x4"
```

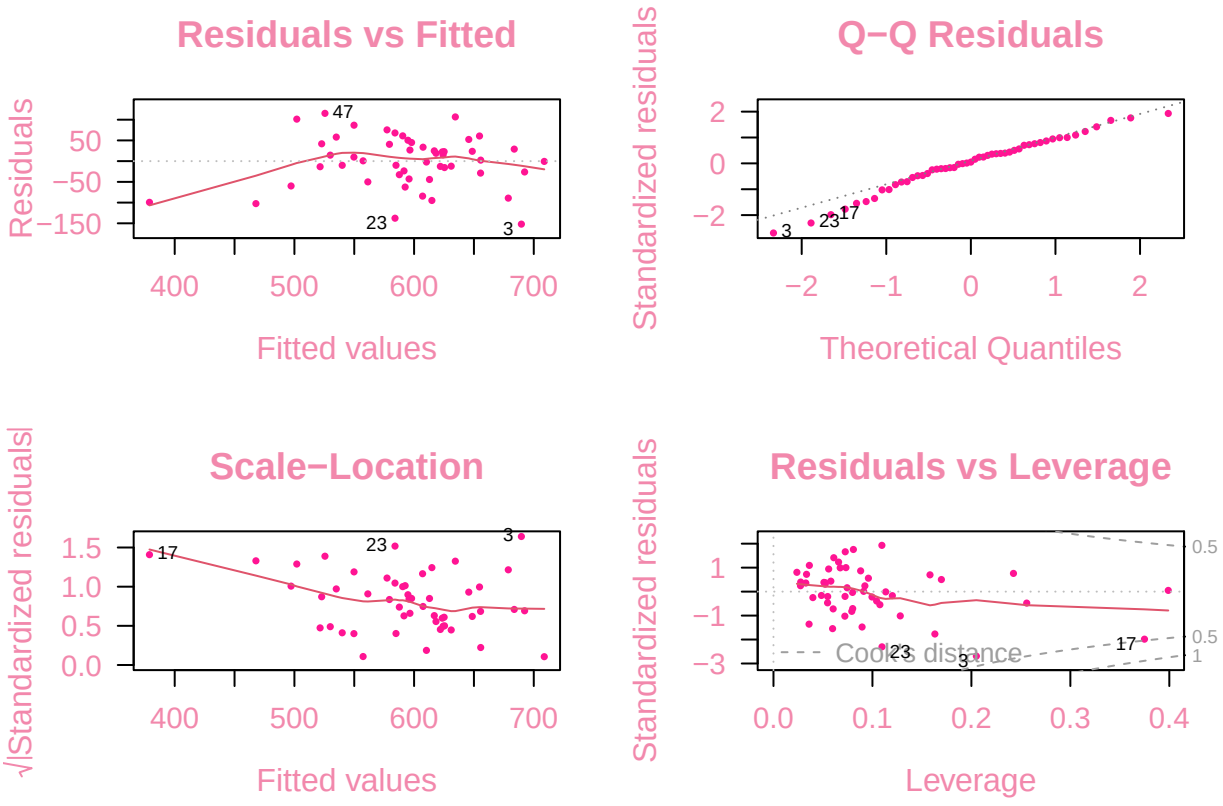
Conclusions are

$$\begin{cases} \textbf{BIC}: & y \sim x_1 + x_2 + x_4 \\ \textbf{C}_\textbf{p}: & y \sim x_1 + x_2 + x_3 + x_4 \\ \textbf{Backward}: & y \sim x_1 + x_2 + x_4 \\ \textbf{Stepwise}: & y \sim x_2 + x_1 + x_4 \end{cases}$$

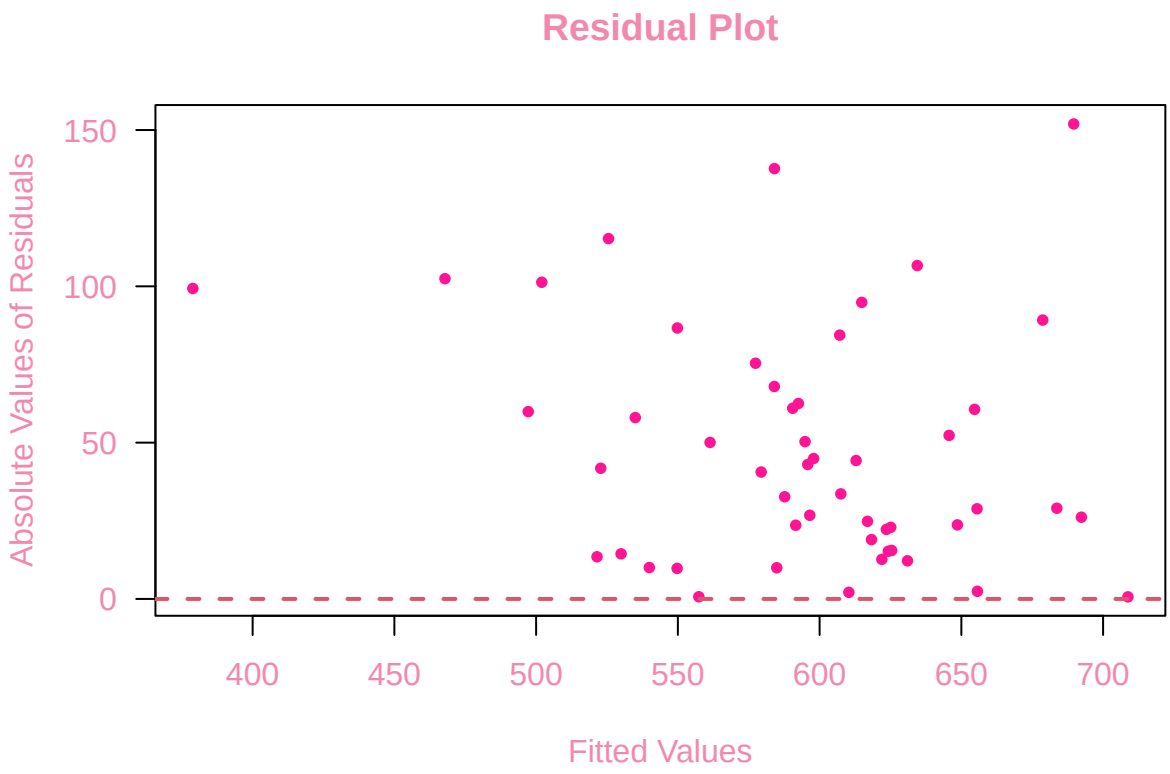
Hence, the results are not consistent.

(e)

It is preferable to first conduct diagnostic checks on the model:



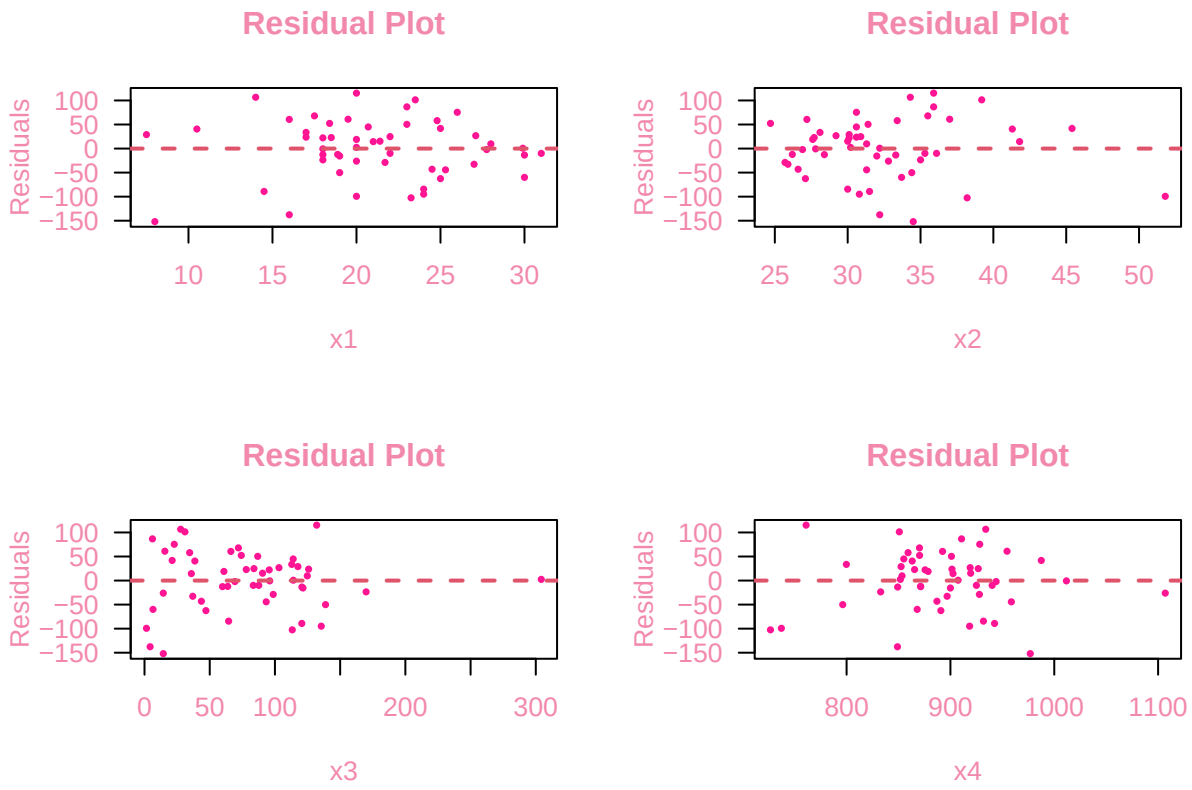
Plot the absolute values of residuals versus the fitted values:



There is no obvious evidence that the constant variance assumption is violated.

$\left(\textit{INTENTIONALLY LEFT BLANK}\right)$

Plot the residuals versus each predictor:



The plots for x_2 and x_3 suggest slight non-constant variance.
Therefore, it may be appropriate to first apply WLS or transform y and then refit the model.

(The data used in this problem are derived from the tables at http://www.fhwa.dot.gov/policy/ohim/hs05/.)