SHC 798 Assignment 2, 2025

Richard Lubega

2025-10-03

SHC 798 Assignment 2, 2025

Multiple Linear Analysis (MLR)

Question 1: Concrete Strength

```r
pacman::p_load(tidymodels)

# Getting started with the dataset in concrete.csv
concrete <- read.csv(file.choose(), header = TRUE, na.strings = c("NA")) # open dataset
head(concrete) # View first few rows of the dataset
```

```
##   cement  wcr age strength
## 1  369.9 0.48  14     12.7
## 2  344.5 0.49  28     12.7
## 3  375.9 0.44  14     14.7
## 4  410.9 0.44  28     25.0
## 5  340.6 0.54  28      7.9
## 6  340.6 0.57  28      4.9
```

```r
summary(concrete) # Get an overview of the dataset
```

```
##      cement           wcr               age           strength
##  Min.   :271.6   Min.   :0.3700   Min.   : 7.00   Min.   :-0.700
##  1st Qu.:322.5   1st Qu.:0.4775   1st Qu.:14.00   1st Qu.: 5.950
##  Median :340.8   Median :0.5000   Median :28.00   Median : 8.800
##  Mean   :343.8   Mean   :0.4997   Mean   :21.47   Mean   : 9.485
##  3rd Qu.:366.2   3rd Qu.:0.5200   3rd Qu.:28.00   3rd Qu.:11.900
##  Max.   :424.1   Max.   :0.6200   Max.   :56.00   Max.   :25.000
```

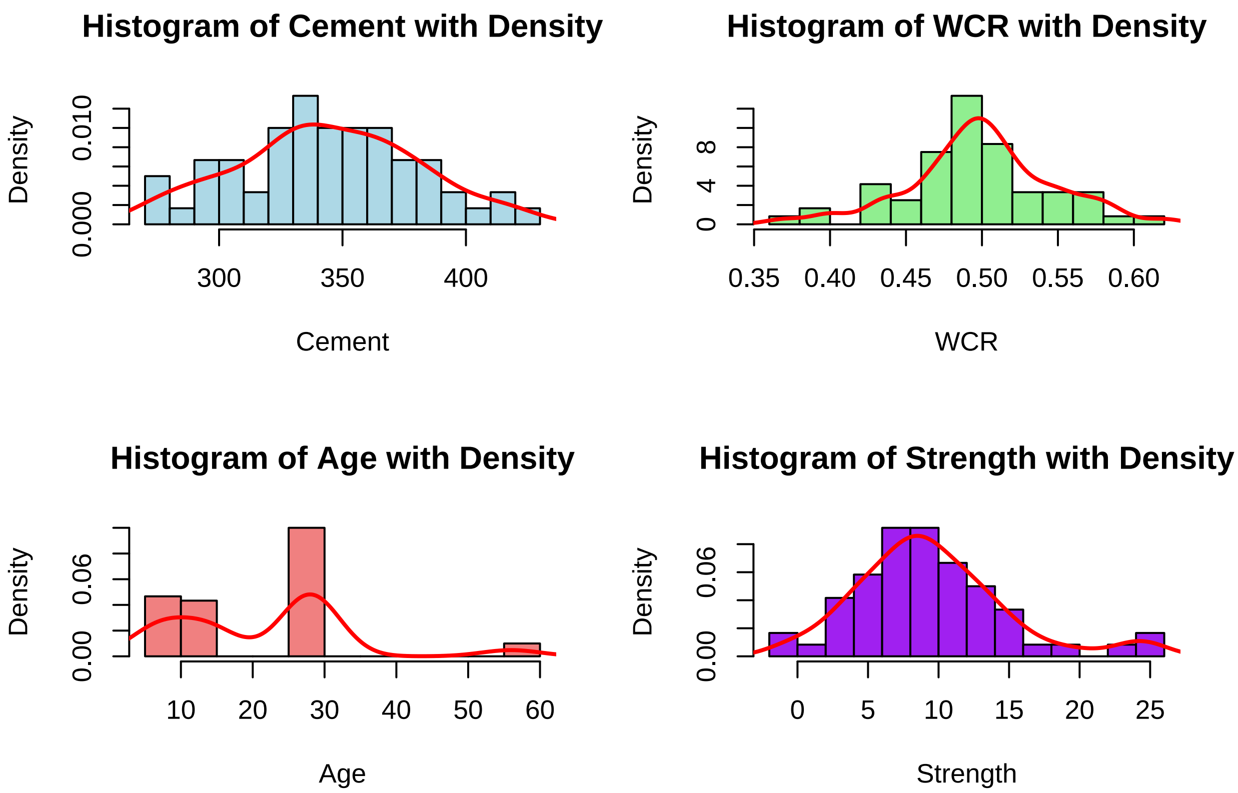
```r
str(concrete) # inspect the dataset and viewing column data types
```

```
## 'data.frame':    60 obs. of  4 variables:
##  $ cement  : num  370 344 376 411 341 ...
##  $ wcr     : num  0.48 0.49 0.44 0.44 0.54 0.57 0.5 0.55 0.52 0.47 ...
##  $ age     : int  14 28 14 28 28 28 7 28 14 7 ...
##  $ strength: num  12.7 12.7 14.7 25 7.9 4.9 7.5 8.9 4.5 9.4 ...
```

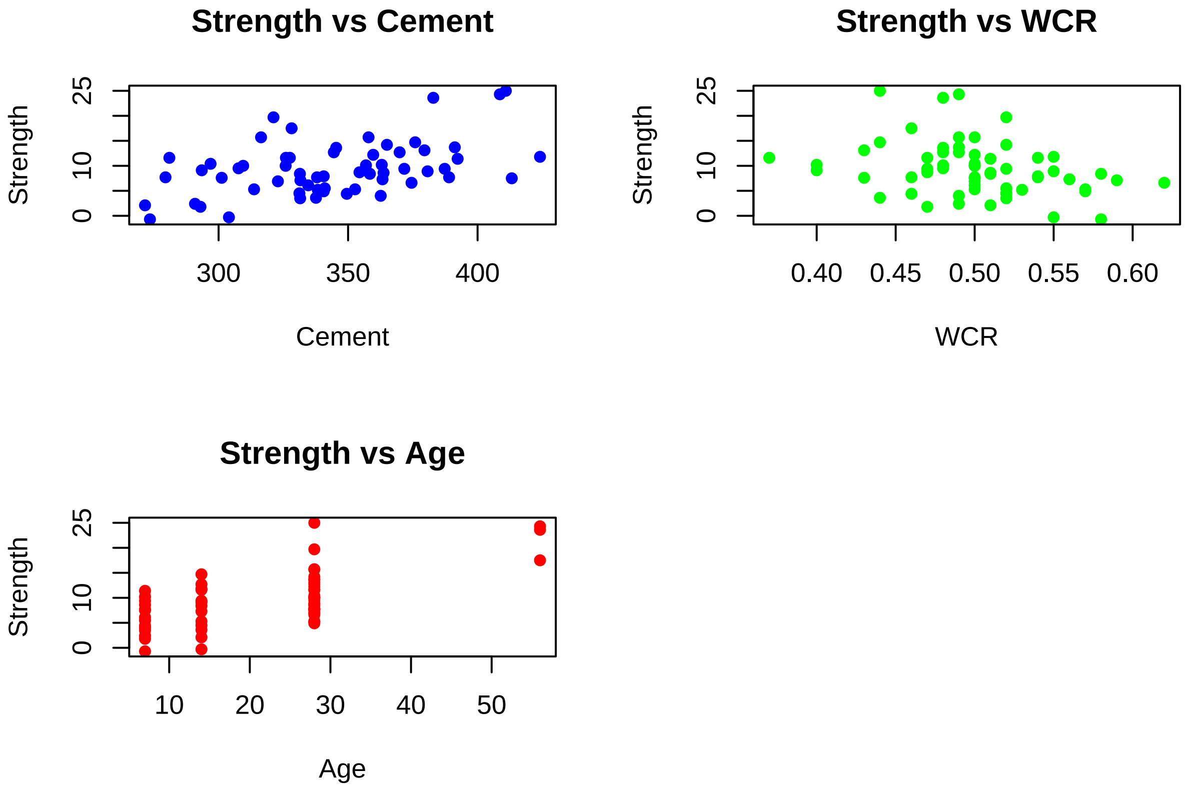
```r
# view(concrete)
unique(concrete$age)
```

```
## [1] 14 28  7 56
```

```r
# Part a): Data Preparation
# [a-1] Histograms with overlaid marginal density distributions
par(mfrow = c(2, 2))

# cement
hist(concrete$cement, main = "Histogram of Cement with Density",
     xlab = "Cement", col = "lightblue", probability = TRUE, breaks = 15)
lines(density(concrete$cement), col = "red", lwd = 2)

# wcr
hist(concrete$wcr, main = "Histogram of WCR with Density",
     xlab = "WCR", col = "lightgreen", probability = TRUE, breaks = 15)
lines(density(concrete$wcr), col = "red", lwd = 2)

# age
hist(concrete$age, main = "Histogram of Age with Density",
     xlab = "Age", col = "lightcoral", probability = TRUE, breaks = 15)
lines(density(concrete$age), col = "red", lwd = 2)

# strength
hist(concrete$strength, main = "Histogram of Strength with Density",
     xlab = "Strength", col = "purple", probability = TRUE, breaks = 15)
lines(density(concrete$strength), col = "red", lwd = 2)
```



```r
par(mfrow = c(1, 1))

# [a-2] Scatter Plots of Strength against each Predictor
par(mfrow = c(2, 2))

# Strength vs Cement
plot(concrete$cement, concrete$strength, main = "Strength vs Cement",
     xlab = "Cement", ylab = "Strength", pch = 16, col = "blue")


# Strength vs WCR
plot(concrete$wcr, concrete$strength, main = "Strength vs WCR",
     xlab = "WCR", ylab = "Strength", pch = 16, col = "green")

# Strength vs Age
plot(concrete$age, concrete$strength, main = "Strength vs Age",
     xlab = "Age", ylab = "Strength", pch = 16, col = "red")

par(mfrow = c(1, 1))
```



Commenting on the Trend and Need for Variable Transformation The marginal plots are not skewed. No clear need for variable transformations

The scatter plot for strength vs age indicates has distinct values (7, 14, 28, 56) which suggests a discrete or categorical nature rather than continuous. The marginal plots also show spikes at these specific ages rather than a smooth distribution.

Therefore, age may be as a **categorical variable** (factor) in regression to account for its discrete levels. Including *interaction terms* (e.g., cement:age, wcr:age) in such a regression model may be also necessary.

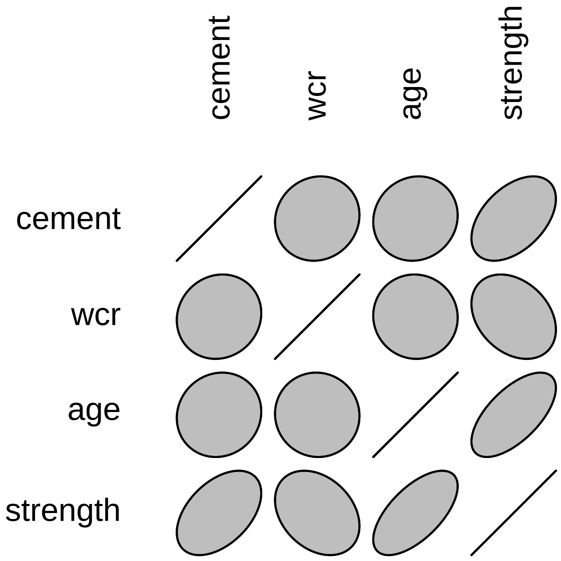
```r
# Part b): Multicollinearity among predictors

# (i) Pearson correlation coefficients
cor(concrete, method = "pearson")
```

```
##              cement         wcr         age   strength
## cement   1.00000000  0.08414330  0.07698239  0.4657863
## wcr      0.08414330  1.00000000 -0.02466868 -0.3063764
## age      0.07698239 -0.02466868  1.00000000  0.6345642
## strength 0.46578632 -0.30637643  0.63456425  1.0000000
```

```r
# Compute the correlation matrix - Same!
cor_matrix <- cor(concrete[, c("cement","wcr","age","strength")])
print(cor_matrix)
```

```
##              cement         wcr         age   strength
## cement   1.00000000  0.08414330  0.07698239  0.4657863
## wcr      0.08414330  1.00000000 -0.02466868 -0.3063764
## age      0.07698239 -0.02466868  1.00000000  0.6345642
## strength 0.46578632 -0.30637643  0.63456425  1.0000000
```

```
# (ii) An ellipse plot to visualise collinearity
pacman::p_load(ellipse)
plotcorr(cor(concrete))
```



```
# (iii) Variance Inflation Factors (VIFs)
pacman::p_load(car)
conc_model <- lm(strength ~ cement + wcr + age, data = concrete)
vif(conc_model)
```

```
##   cement      wcr      age
## 1.013514 1.008121 1.006951
```

From the above **collinearity audit** checks (Pearson correlation coefficients and the ellipse plot), the somewhat elongated ellipses, particularly between cement and strength (0.46578632), and age and strength (0.6345642), suggest potential multicollinearity among these predictors.

This indicates that the predictors may be highly correlated with each other and with the response variable, but Since all VIF values are very close to 1 (well below 5), there is **no significant multicollinearity** among the predictors. This suggests that the predictors are largely independent of each other, which is ideal for a stable regression model.

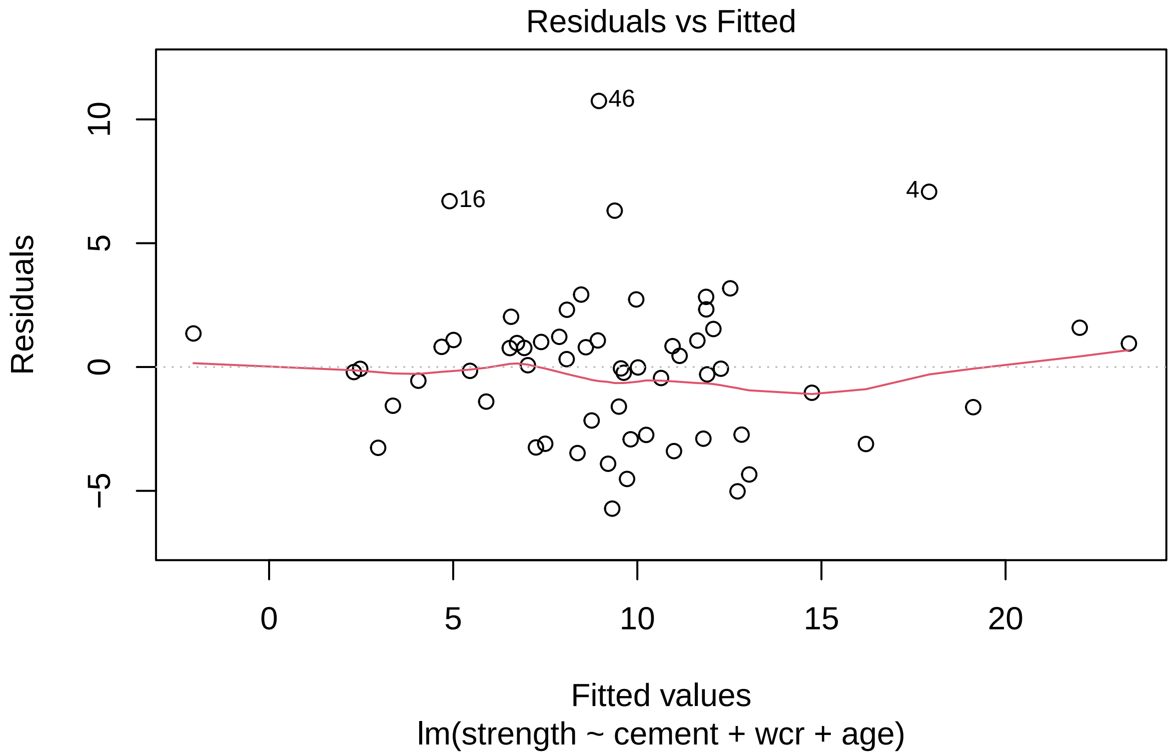
```
# Part-C-1 Model
conc_model <- lm(strength ~ cement + wcr + age, data = concrete)
summary(conc_model)
```

```
##
## Call:
## lm(formula = strength ~ cement + wcr + age, data = concrete)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## age           0.26614    0.03383   7.868 1.27e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

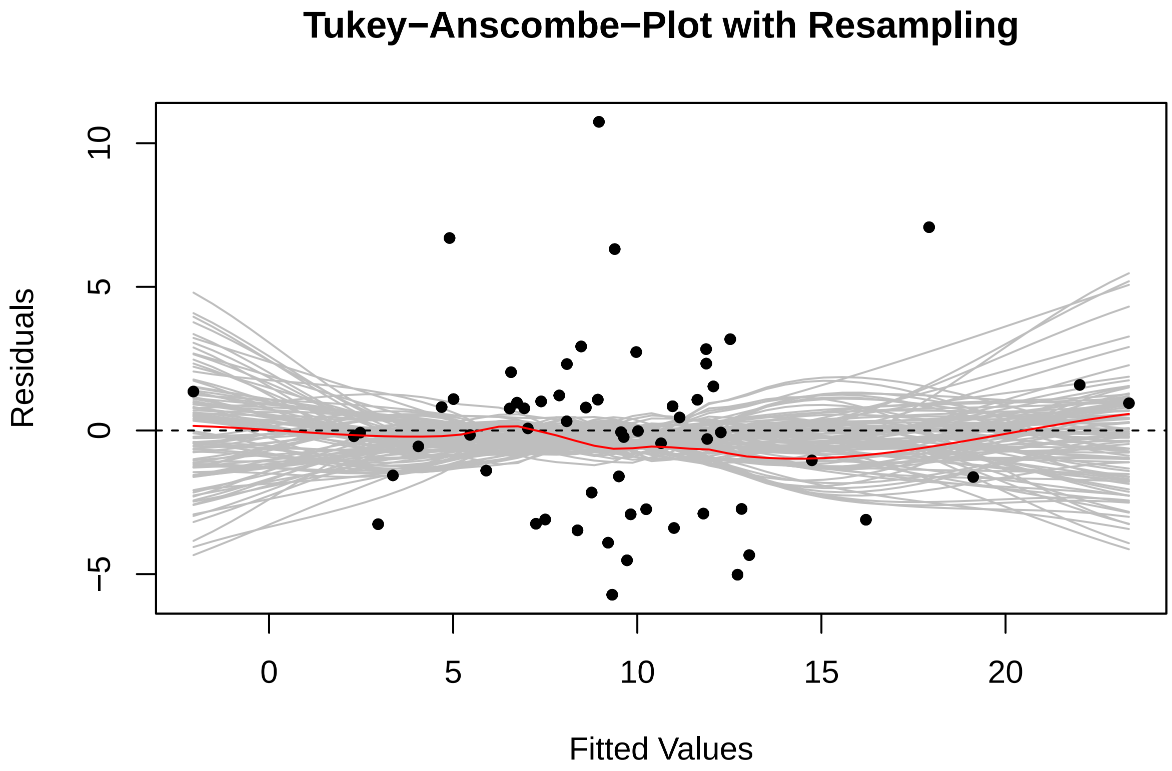
```
confint(conc_model)
```

```
##                    2.5 %       97.5 %
## (Intercept) -11.51290192  10.70239966
## cement        0.04410025   0.08904318
## wcr         -54.58857939 -20.30764934
## age           0.19838020   0.33389978
```

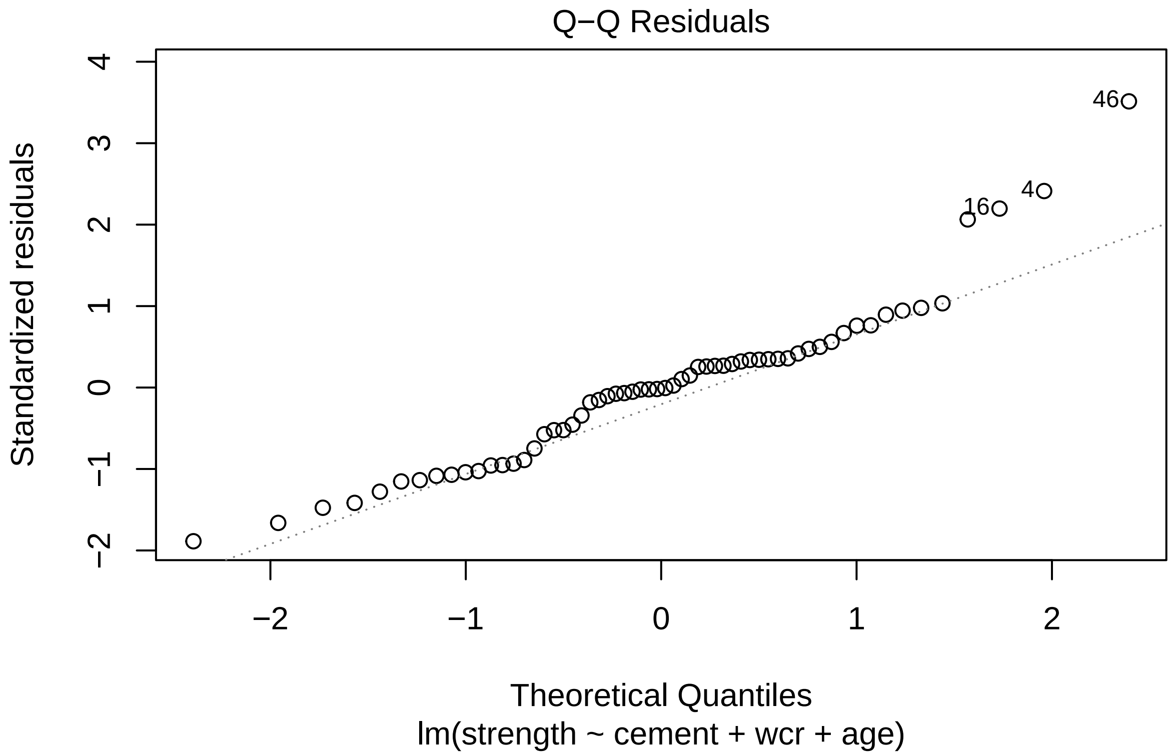
```
# Part C-2: Comment on the Model output
# -Regression coefficients
# -Model significance
# -Adequacy of fit, and
# -Appropriateness of fit

## Residual analysis
plot(conc_model, which=1)
```



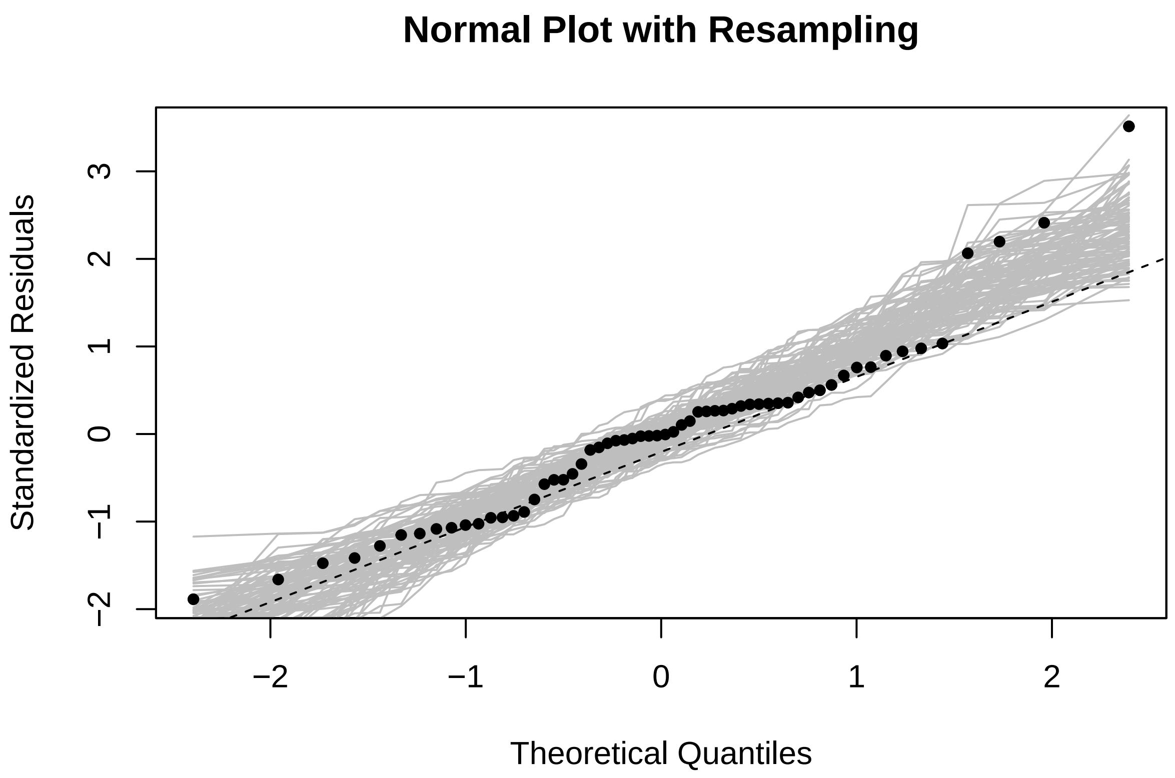
```
resplot(conc_model, plots = 1)
```



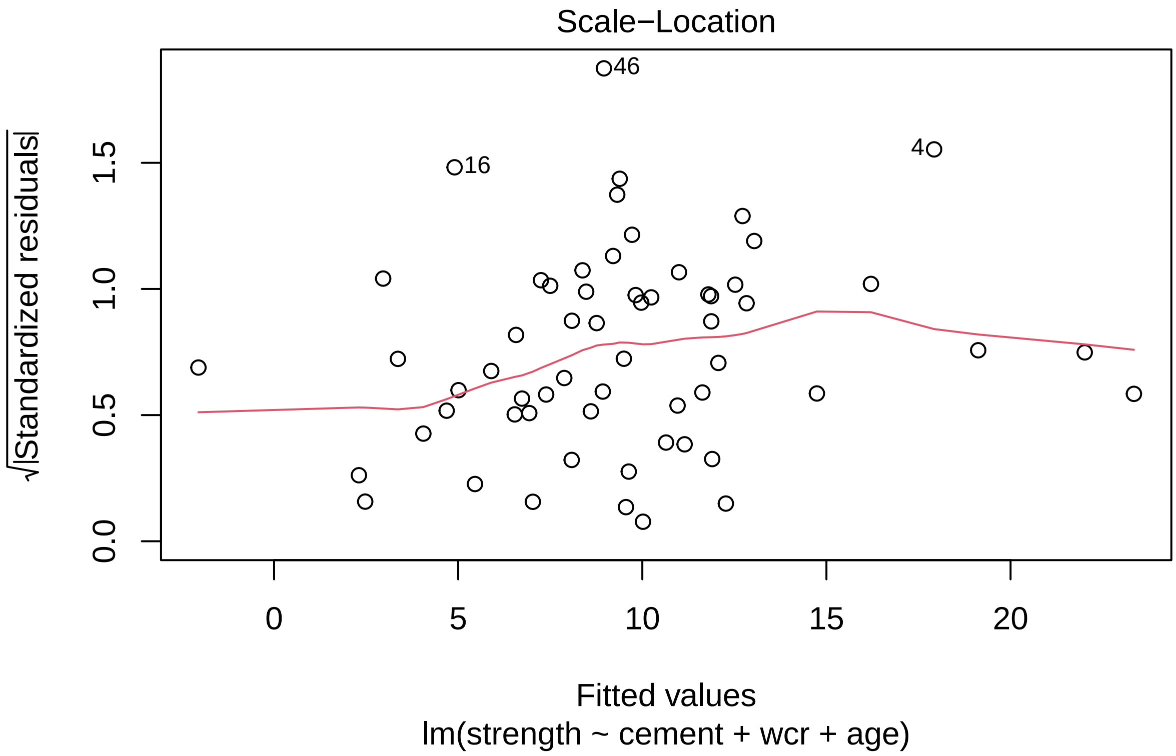
```r
plot(conc_model, which = 2)
```



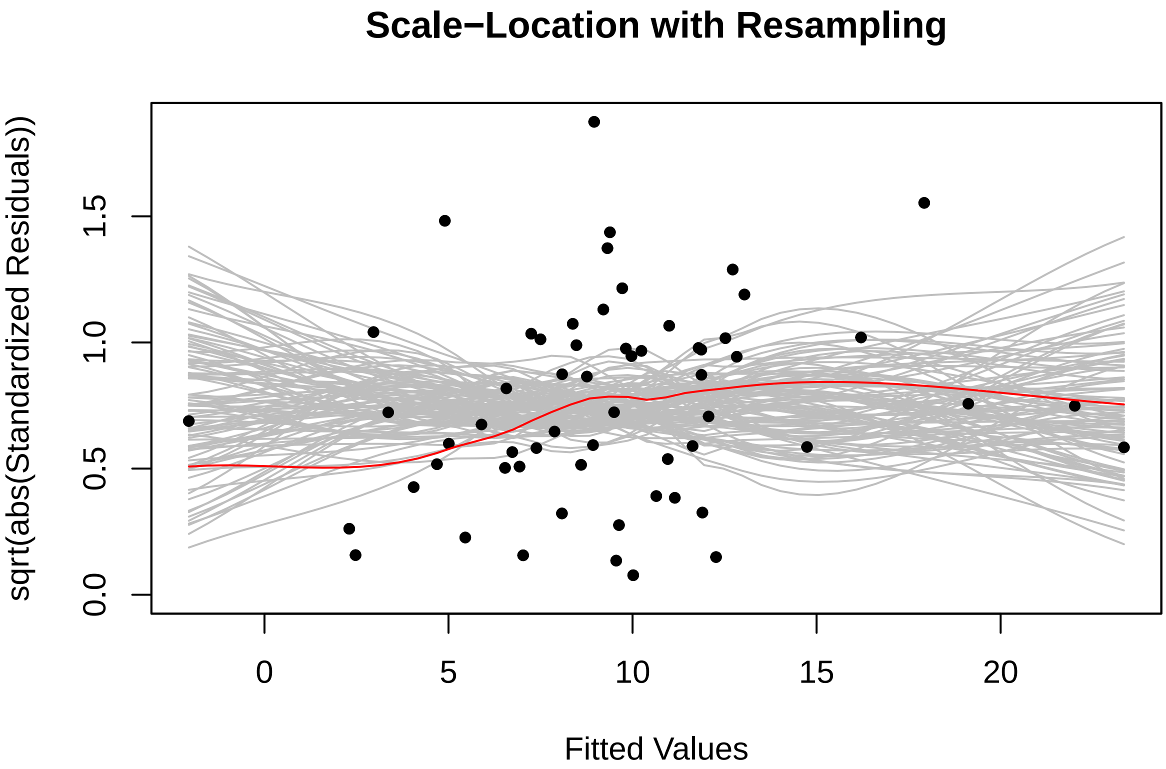
```
resplot(conc_model, plots = 2)
```



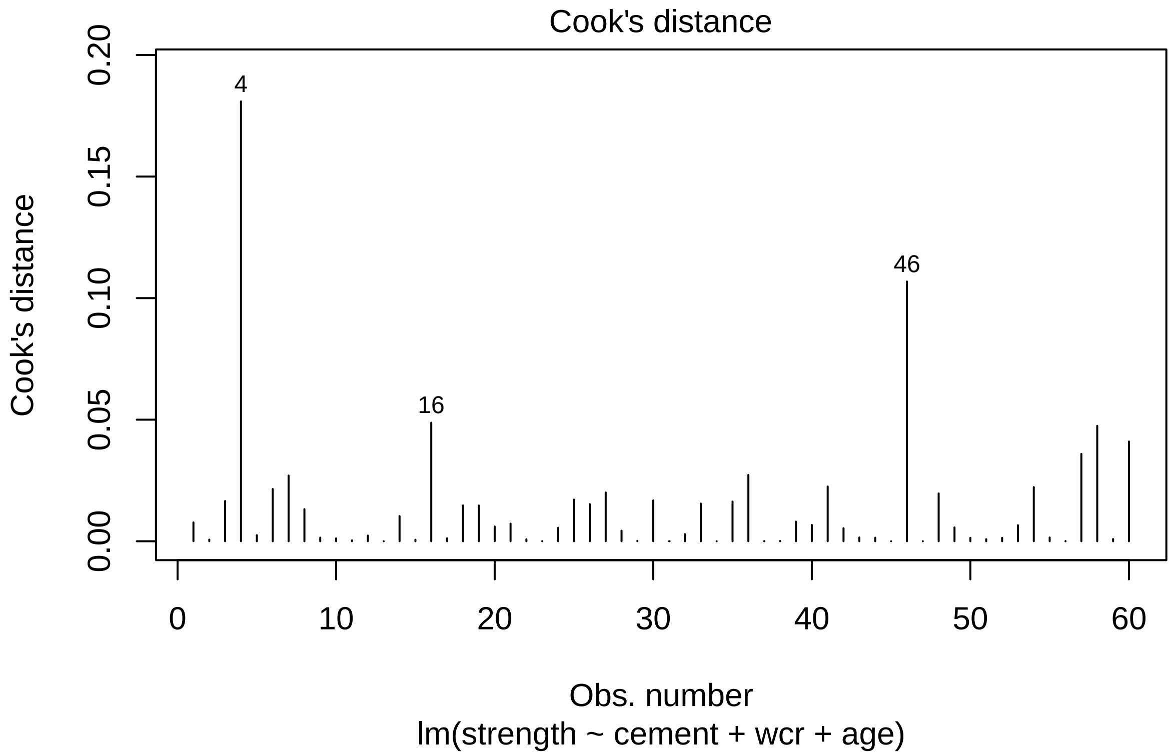
```
## Scale-location plot
plot(conc_model, which = 3)
```



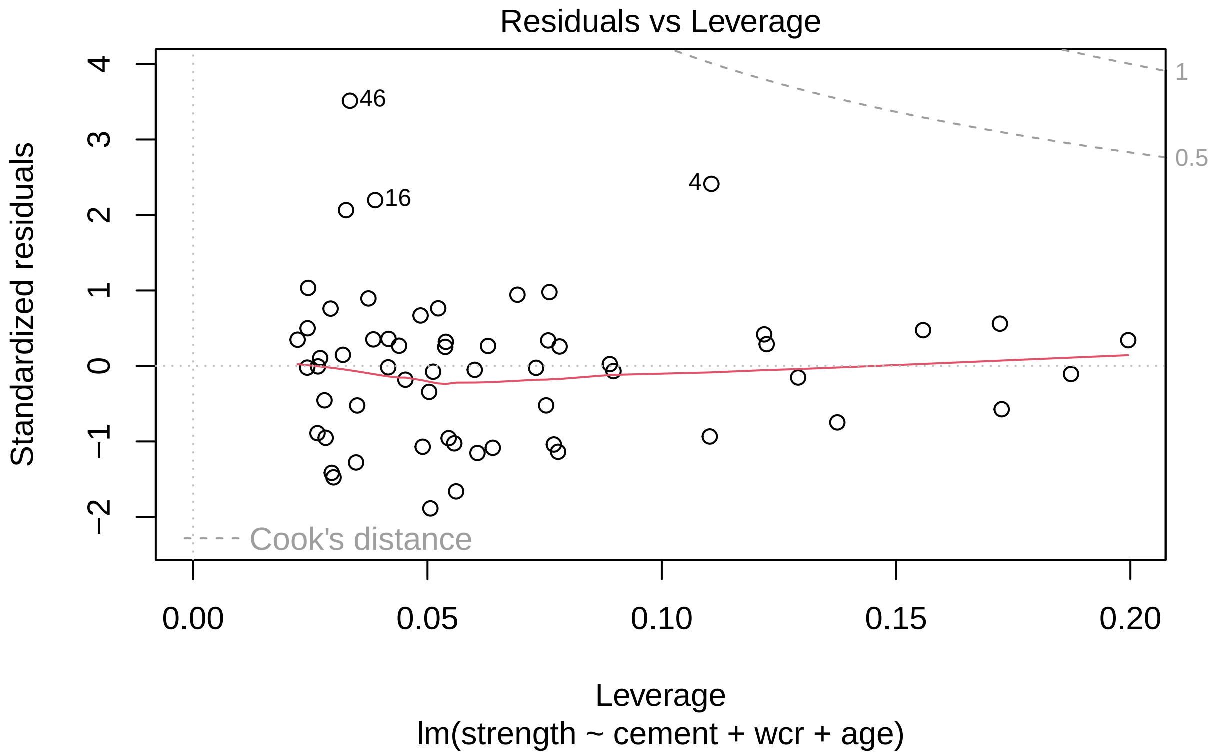
```r
resplot(conc_model, plots = 3)
```



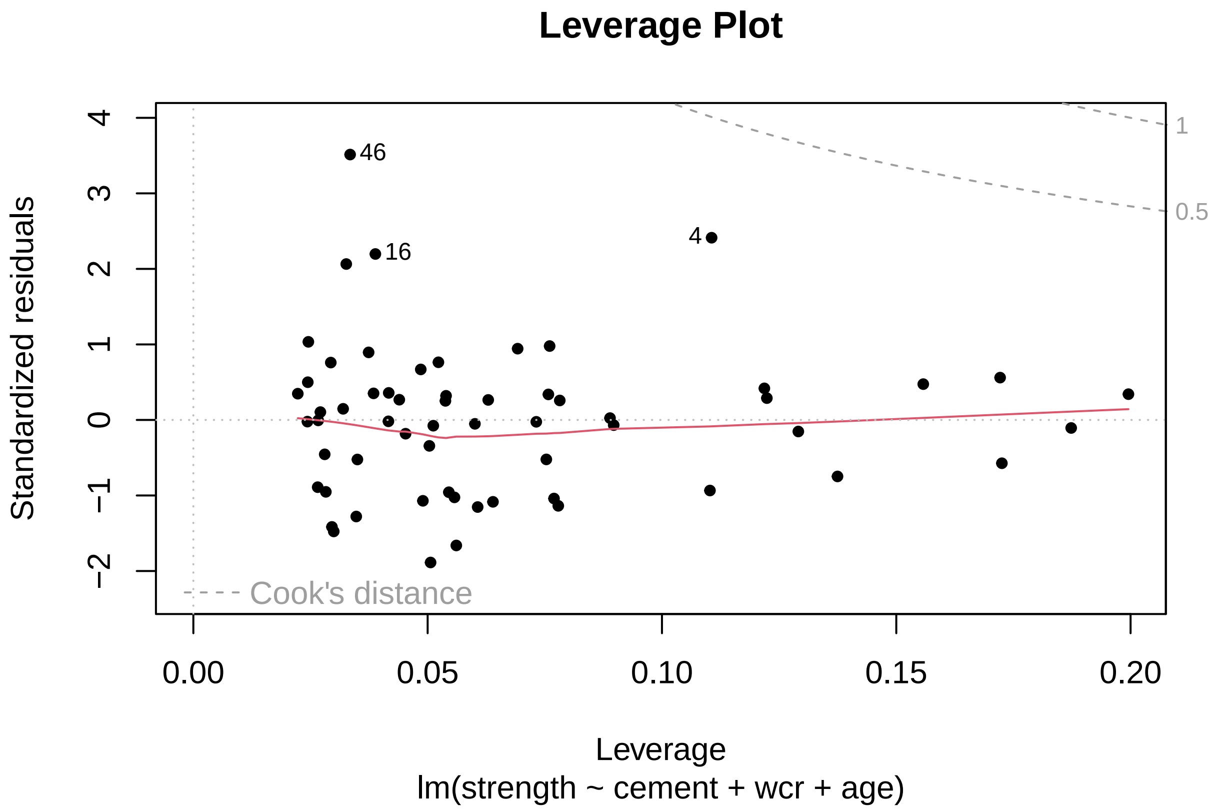
```r
## Cook's Distance plot
plot(conc_model, which = 4)
```



```r
plot(conc_model, which = 5)
```



```r
resplot(conc_model, plots = 4)
```



```
# Autocorrelation using the Durbin-Watson test
pacman::p_load(lmtest)
dwtest(conc_model)
```

```
##
##  Durbin-Watson test
##
## data:  conc_model
## DW = 1.8426, p-value = 0.2733
## alternative hypothesis: true autocorrelation is greater than 0
```

Assumptions not violated, No Autocorrelation.

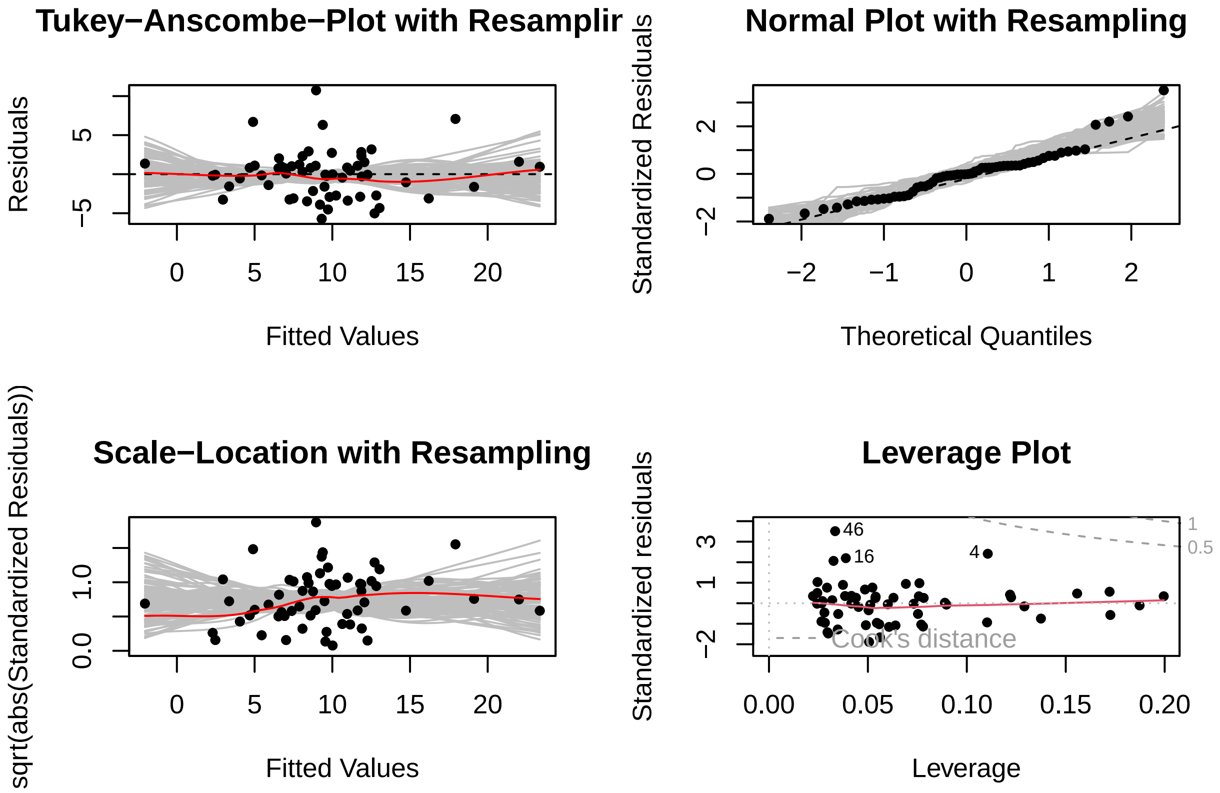
```
# Part d): Variable Selection
# Backward Elimination with AIC
conc.back <- stats::step(conc_model, direction="backward")
```

```
## Start:  AIC=140.02
## strength ~ cement + wcr + age
##
##          Df Sum of Sq     RSS    AIC
## <none>                 541.67 140.02
## - wcr     1    185.28  726.95 155.67
## - cement  1    340.67  882.34 167.29
## - age     1    598.81 1140.49 182.69
```

```r
summary(conc.back)
```

```
##
## Call:
## lm(formula = strength ~ cement + wcr + age, data = concrete)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## age           0.26614    0.03383   7.868 1.27e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

```r
resplot(conc.back)
```



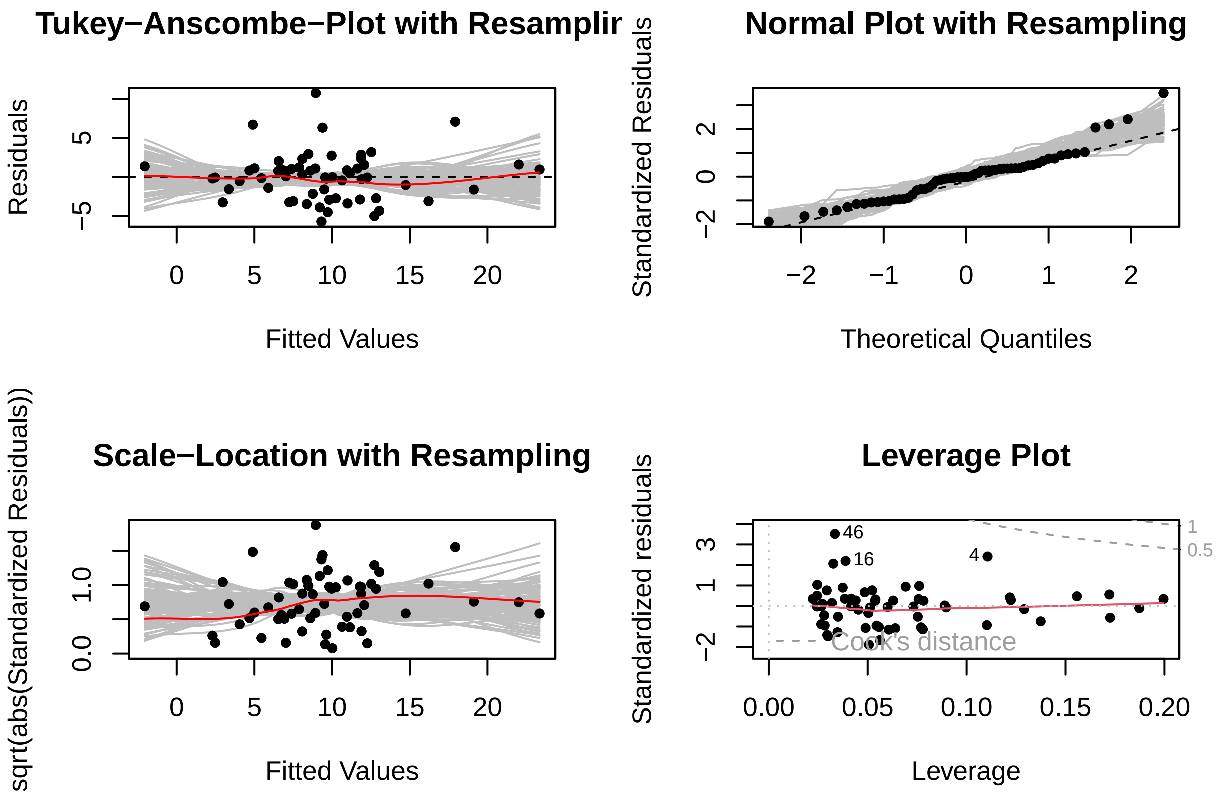
```r
# Forward Selection with AIC
conc_null <- lm(strength ~ 1, data = concrete) # Intercept-only model
sc <- list(lower=conc_null, upper=conc_model)
conc.forw <- stats::step(conc_null, scope=sc, direction="forward", k=2)
```

```
## Start:  AIC=203.37
## strength ~ 1
##
##          Df Sum of Sq    RSS    AIC
## + age     1    692.91 1027.9 174.45
## + cement  1    373.33 1347.4 190.70
## + wcr     1    161.52 1559.2 199.46
## <none>                1720.8 203.37
##
## Step:  AIC=174.45
## strength ~ age
##
##          Df Sum of Sq     RSS    AIC
## + cement  1    300.92  726.95 155.67
## + wcr     1    145.53  882.34 167.29
## <none>                1027.87 174.45
##
## Step:  AIC=155.67
## strength ~ age + cement
##
##        Df Sum of Sq    RSS    AIC
## + wcr   1    185.28 541.67 140.02
## <none>              726.95 155.67
##
## Step:  AIC=140.02
## strength ~ age + cement + wcr
```

```r
summary(conc.forw)
```

```
##
## Call:
## lm(formula = strength ~ age + cement + wcr, data = concrete)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## age           0.26614    0.03383   7.868 1.27e-10 ***
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

```
resplot(conc.forw)
```



```
# AIC Stepwise Model Search: Both Directions Approach
# starting with the null model
conc.b1 <- stats::step(conc_null, scope = sc, direction = "both")
```

```
## Start:  AIC=203.37
## strength ~ 1
##
##           Df Sum of Sq    RSS    AIC
## + age      1    692.91 1027.9 174.45
## + cement   1    373.33 1347.4 190.70
## + wcr      1    161.52 1559.2 199.46
## <none>                 1720.8 203.37
##
## Step:  AIC=174.45
## strength ~ age
##
##           Df Sum of Sq     RSS    AIC
## + cement   1    300.92  726.95 155.67
## + wcr      1    145.53  882.34 167.29
## <none>                 1027.87 174.45
## - age     1    692.91 1720.78 203.37
##
## Step:  AIC=155.67
## strength ~ age + cement
##
##          Df Sum of Sq     RSS    AIC
## + wcr     1    185.28  541.67 140.02
## <none>                 726.95 155.67
## - cement  1    300.92 1027.87 174.45
## - age     1    620.49 1347.44 190.70
##
## Step:  AIC=140.02
## strength ~ age + cement + wcr
##
##          Df Sum of Sq     RSS    AIC
## <none>                 541.67 140.02
## - wcr     1    185.28  726.95 155.67
## - cement  1    340.67  882.34 167.29
## - age     1    598.81 1140.49 182.69
```

```
summary(conc.b1)
```

```
##
## Call:
## lm(formula = strength ~ age + cement + wcr, data = concrete)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## age           0.26614    0.03383   7.868 1.27e-10 ***
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

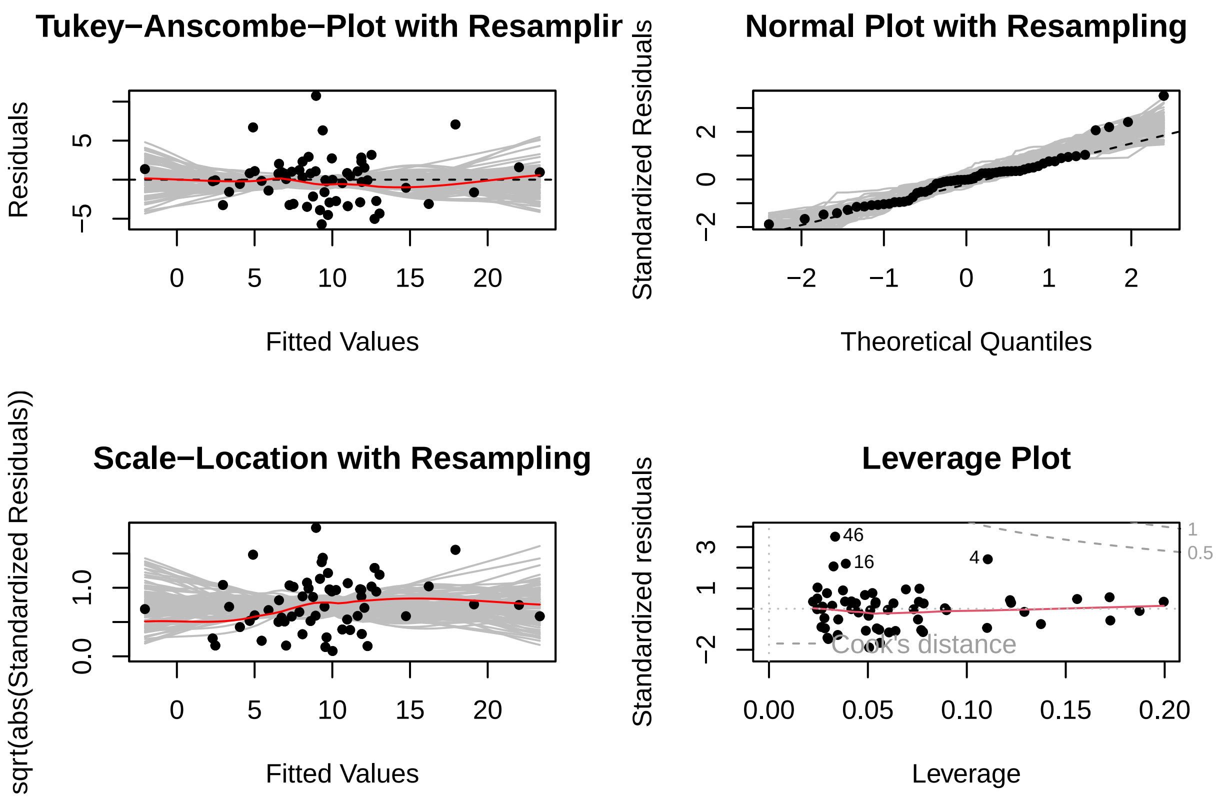
```
resplot(conc.b1)

# starting with the full model
conc.b2 <- stats::step(conc_model, scope = sc, direction = "both")
```

```
## Start:  AIC=140.02
## strength ~ cement + wcr + age
##
##          Df Sum of Sq     RSS    AIC
## <none>                 541.67 140.02
## - wcr     1    185.28  726.95 155.67
## - cement  1    340.67  882.34 167.29
## - age     1    598.81 1140.49 182.69
```

```
summary(conc.b2)
```

```
##
## Call:
## lm(formula = strength ~ cement + wcr + age, data = concrete)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## age           0.26614    0.03383   7.868 1.27e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

```
resplot(conc.b2)
```



```
# starting with a model somewhere in the middle
conc_mid <- lm(strength ~  wcr + age, data = concrete)
conc.b3 <- stats::step(conc_mid, scope = sc, direction = "both")
```

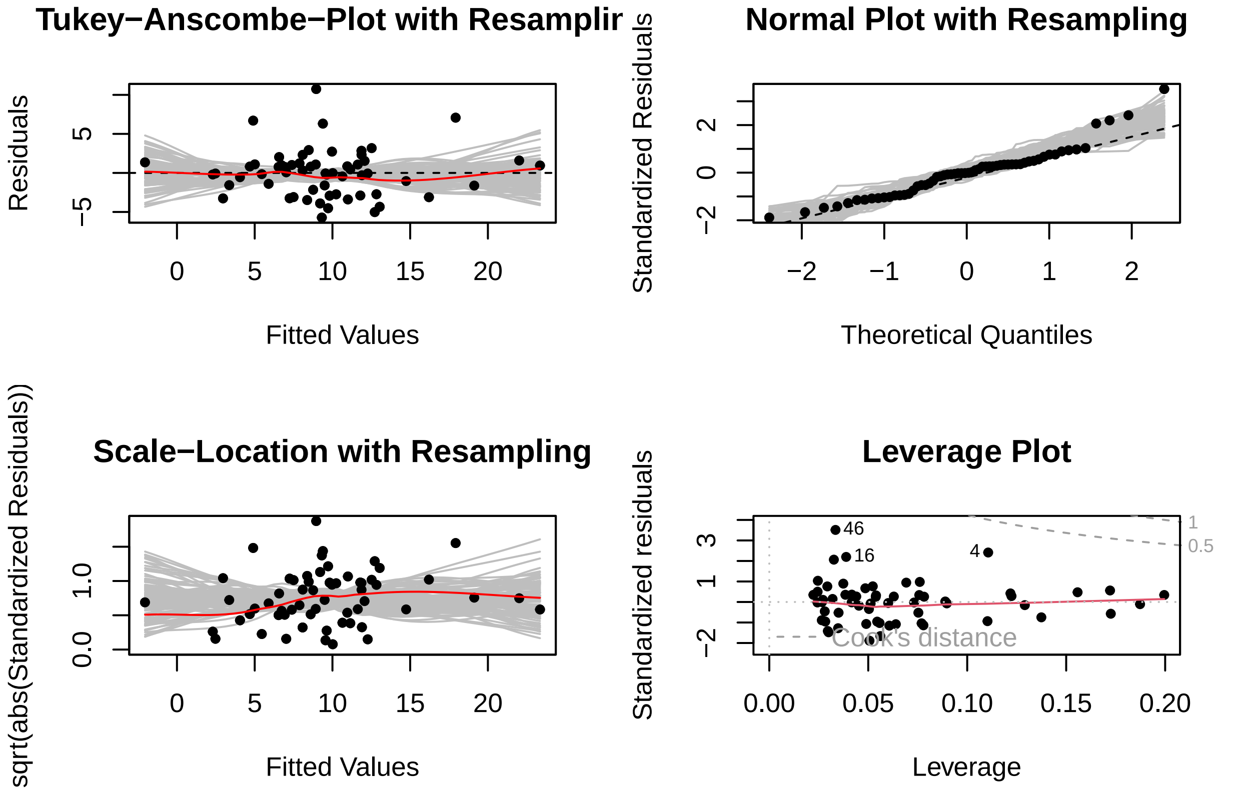
```
## Start:  AIC=167.29
## strength ~ wcr + age
##
##            Df Sum of Sq     RSS    AIC
## + cement    1    340.67  541.67 140.02
## <none>                   882.34 167.29
## - wcr       1    145.53 1027.87 174.45
## - age       1    676.91 1559.25 199.46
##
## Step:  AIC=140.02
## strength ~ wcr + age + cement
##
##            Df Sum of Sq     RSS    AIC
## <none>                   541.67 140.02
## - wcr       1    185.28  726.95 155.67
## - cement    1    340.67  882.34 167.29
## - age       1    598.81 1140.49 182.69
```

```
summary(conc.b3)
```

```
##
```

```
## Call:
## lm(formula = strength ~ wcr + age + cement, data = concrete)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.718 -2.303 -0.037  1.123 10.743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.40525    5.54484  -0.073    0.942
## wcr         -37.44811    8.55637  -4.377 5.31e-05 ***
## age           0.26614    0.03383   7.868 1.27e-10 ***
## cement        0.06657    0.01122   5.935 1.94e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.11 on 56 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6684
## F-statistic: 40.63 on 3 and 56 DF,  p-value: 4.441e-14
```

```
resplot(conc.b3)
```



```
# AIC is used when the principal aim is the prediction
```

All predictors are kept in all 6 models. There are no major improvements in residual plots for all the models. Also, no noticeable changes on predictor significance or model fit.

```r
# Part e) 5-fold cross validation
# Full Model is (strength ~ cement + wcr + age)
# Reduced Model is (strength ~ cement + age); wcr is dropped to see effect on prediction performance

set.seed(123) # Set seed for reproducibility

n <- nrow(concrete) # Number of observations
k <- 5 # Number of folds
sb <- round(seq(0, n, length = (k + 1)))  # Fold boundaries

# Initialize vectors to store MSPE for each model
mspe_full <- numeric(k)
mspe_reduced <- numeric(k)

# 5-fold cross-validation for full model (strength ~ cement + wcr + age)
for (i in 1:k) {
  test <- (sb[k + 1 - i] + 1):sb[k + 2 - i]
  train <- (1:n)[-test]
  fit_full <- lm(strength ~ cement + wcr + age, data = concrete[train, ])
  pred_full <- predict(fit_full, newdata = concrete[test, ])
  mspe_full[i] <- mean((concrete$strength[test] - pred_full)^2, na.rm = TRUE)
}

# 5-fold cross-validation for reduced model (strength ~ cement + age)
for (i in 1:k) {
  test <- (sb[k + 1 - i] + 1):sb[k + 2 - i]  # Same fold split comparability
  train <- (1:n)[-test]
  fit_reduced <- lm(strength ~ cement + age, data = concrete[train, ])
  pred_reduced <- predict(fit_reduced, newdata = concrete[test, ])
  mspe_reduced[i] <- mean((concrete$strength[test] - pred_reduced)^2, na.rm = TRUE)
}

# Calculating overall MSPE for each model
mspe_full_mean <- mean(mspe_full, na.rm = TRUE)
mspe_reduced_mean <- mean(mspe_reduced, na.rm = TRUE)

# Report results
cat("MSPE per fold for Full Model:", mspe_full, "\n")
```

```
## MSPE per fold for Full Model: 10.68408 15.07415 8.694391 6.984834 11.73812
```

```r
cat("MSPE per fold for Reduced Model:", mspe_reduced, "\n")
```

```
## MSPE per fold for Reduced Model: 15.09433 13.59734 7.99105 9.425239 20.52089
```

```r
cat("MSPE for Full Model:", mspe_full_mean, "\n")
```

```
## MSPE for Full Model: 10.63511
```

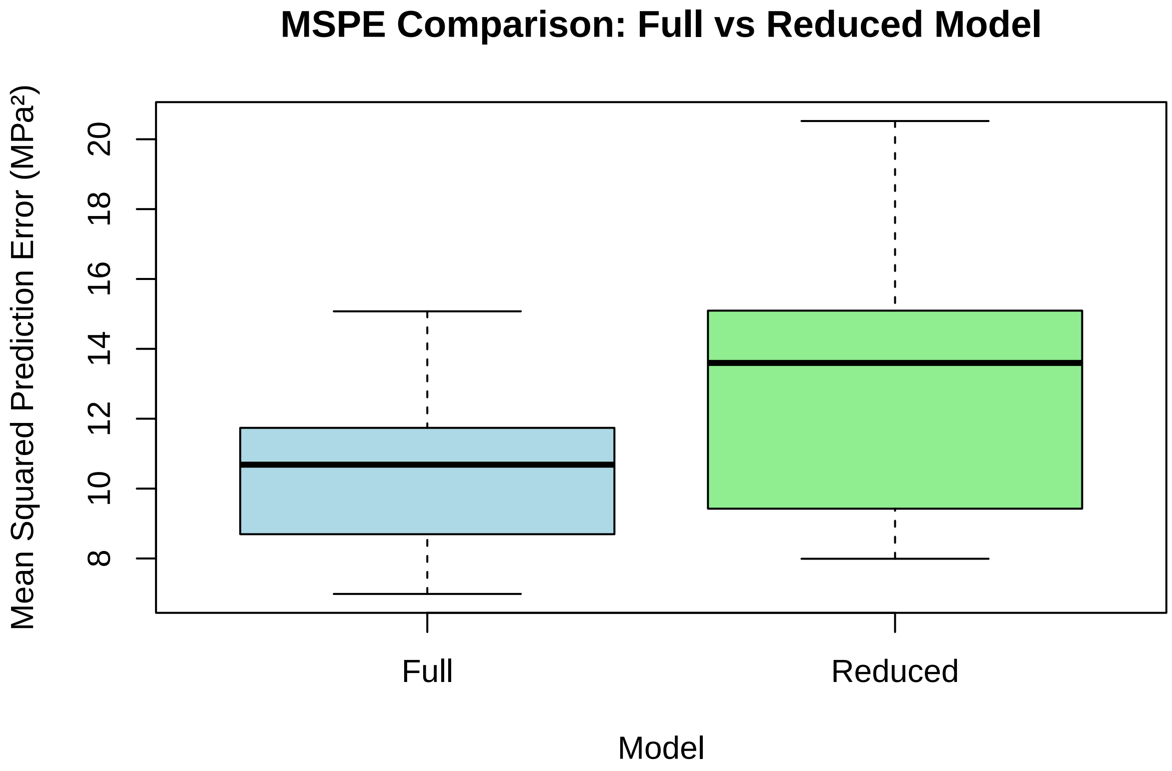
```r
cat("MSPE for Reduced Model:", mspe_reduced_mean, "\n")
```

```
## MSPE for Reduced Model: 13.32577
```

```
# Relative increase in MSPE
relative_increase <- ((mspe_reduced_mean - mspe_full_mean) / mspe_full_mean) * 100
cat("Relative increase in MSPE (%):", relative_increase, "\n")
```

```
## Relative increase in MSPE (%): 25.29973
```

```
# Box plots using MSPEs
# Combining MSPEs into a data frame for plotting
mspe_data <- data.frame(
  MSPE = c(mspe_full, mspe_reduced),
  Model = factor(rep(c("Full", "Reduced"), each = k))
)

# Generating box plots
boxplot(MSPE ~ Model, data = mspe_data,
        main = "MSPE Comparison: Full vs Reduced Model",
        ylab = "Mean Squared Prediction Error (MPa²)",
        col = c("lightblue", "lightgreen"),
        border = "black")
```



From the cross-validation exercise, The MSPE for the reduced is substantially higher (25.29973%) than the full model. Therefore, the variable wcr adds predictive power and the full model is preferable for prediction purposes.

```r
# Part f): Prediction
conc.str <- data.frame(cement=350, wcr=0.5, age=28)
# predict(conc_model, newdata = conc.str, interval = "conf")
predict(conc_model, newdata = conc.str, interval = "pred")
```

```
##        fit      lwr      upr
## 1 11.62271 5.324389 17.92103
```