# Predicting the Proportion of Residents of Toronto Neighbourhoods with High Income - Data Analysis

Gavin Pu

2022-12-20

## Packages

This analysis uses the following packages.

```r
library(car)
library(glmnet)
library(opendatatoronto)
library(rms)
```

## Dataset

The original dataset can be found on the City of Toronto's Open Data Portal and is freely licensed under the Open Data License. The opendatatoronto GitHub contains documentation on how to use the `opendatatoronto` package.

```r
# Read the dataset via the `opendatatoronto` package
resources <- list_package_resources("6e19a90f-971c-46b3-852c-0c48c436d1fc")
neighbourhood_profiles <- get_resource(resources[
  resources$id == "f07fe8f0-fa24-4d68-8cb4-326e280b0b05", ])
```

## Data Cleaning

Each variable has a unique row identifier in the `neighbourhood_profiles` data frame. There are five possible predictors that may be used to create the regression model.

1. The first predictor is the proportion of individuals in a neighbourhood who have a university certificate, diploma, or degree at the bachelor level or above (row identifier 1710).
2. The second predictor is the proportion of individuals in a neighbourhood whose occupation is management (row identifier 1923).
3. The third predictor is the proportion of individuals in a neighbourhood who work in professional, scientific, or technical industires (row identifier 1947).
4. The fourth predictor is the proportion of individuals in a neighbourhood who speak both English and French (row identifier 131).
5. The fifth predictor is the portion of individuals in a neighbourhood who studied in a province or territory of Canada outside their original province or territory of residence (row identifier 1861).

The response is the proportion of individuals in a neighbourhood who have an employment income of $100,000 or greater (row identifier 1017).

For brevity, each predictor will henceforth be referred to as "predictor X", where X is the number in the above list. For example, predictor 1 is the proportion of individuals in a neighbourhood who have a university certificate, diploma, or degree at the bachelor level or above.

```r
# Get all neighbourhood names
neighbourhoods <- colnames(neighbourhood_profiles[
  7:ncol(neighbourhood_profiles)])

# Extract the variables of interest from `neighbourhood_profiles`
X_ids <- c(1710, 1923, 1947, 131, 1861, 1017)

# Create a new data frame called `data`
data <- data.frame(Neighbourhood = neighbourhoods)
column_names <- c()

# N = total number of individuals in the neighbourhood
population_2016_X_id <- 3
N <- neighbourhood_profiles[population_2016_X_id, 7:ncol(
  neighbourhood_profiles)][1, ]
N <- as.numeric(gsub(",", "", N))

# Create each proportion and add it to `data`
for (i in X_ids) {
  column_names <- c(column_names, neighbourhood_profiles$Characteristic[i])
  characteristic <- neighbourhood_profiles[i, 7:ncol(neighbourhood_profiles)][
    1, ]
  characteristic <- as.numeric(gsub(",", "", characteristic)) / N
  data <- cbind(data, characteristic)
}

# Modify `data` so that the predictors are `data[1:5]`, the response is
# `data[6]`, and the neighbourhood name is `data[7]`
column_names <- c(column_names, "Neighbourhood")
data <- cbind(data[2:ncol(data)], data[1])
colnames(data) <- column_names

# Assign shorthand names to each variable
pred1 <- data[, 1]
pred2 <- data[, 2]
pred3 <- data[, 3]
pred4 <- data[, 4]
pred5 <- data[, 5]
resp <- data[, 6]
```

### Functions

These functions will be used later in the analysis.

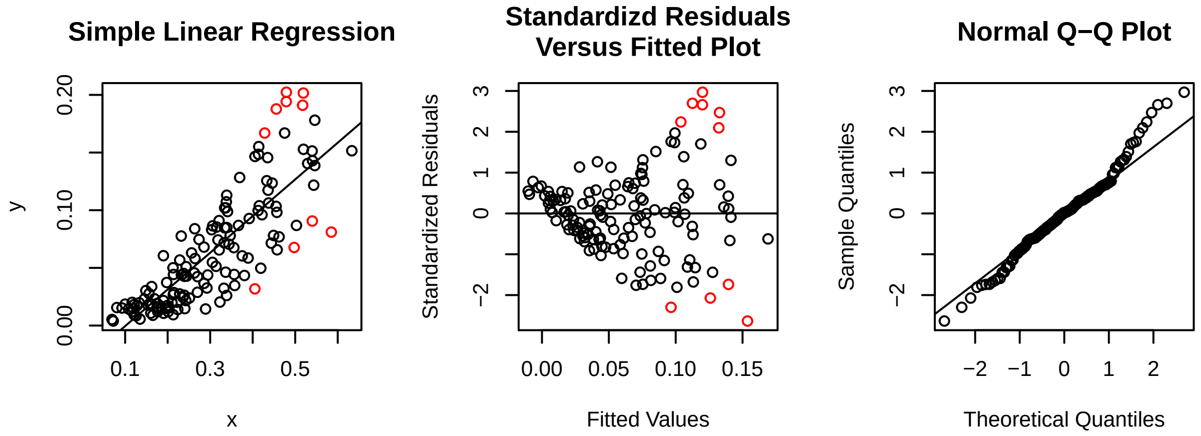
```r
# Perform an SLR analysis
# Precondition: `model` must be a linear model with one predictor
SLR_analysis <- function(model) {
  y <- model$model[[1]]
  x <- model$model[[2]]
  n <- nrow(data)
  st_resid <- rstudent(model)

  # Find influential points
  Di <- cooks.distance(model)
  influential_points <- which(Di > 4 / (n - 2))

  # Create a SLR plot and highlight influential points
  plot(y ~ x, main = "Simple Linear Regression",
       col = ifelse(Di > 4 / (n - 2), "red", "black"))
  abline(model$coefficients)

  # Create a standardized residuals versus fitted values plot and highlight
  # influential points
  plot(st_resid ~ model$fitted.values,
       main = "Standardizd Residuals\nVersus Fitted Plot",
       xlab = "Fitted Values", ylab = "Standardized Residuals",
       col = ifelse(Di > 4 / (n - 2), "red", "black"))
  abline(h = 0)

  # Create a Normal Q-Q plot
  qqnorm(st_resid)
  qqline(st_resid)

  # Perform an ANOVA test and calculate R-squared
  ANOVA <- anova(model)
  RSS <- ANOVA[2, 2]
  SST <- ANOVA[1, 2] + ANOVA[2, 2]
  R_squared <- 1 - (RSS / SST)

  # Return influential points, ANOVA, and R-squared
  return(list(influential_points, ANOVA, R_squared))
}
```

```r
# Perform an MLR analysis
# Precondition: `model` must be a linear model with more than one predictor
MLR_analysis <- function(model) {
  n <- nrow(data)
  p <- length(model$model) - 1
  st_resid <- rstudent(model)

  # Calculate the variance inflation factors (VIFs)
  VIFs <- vif(model)

  # Find influential points
  Di <- cooks.distance(model)
  influential_points <- which(Di > qf(0.5, p + 1, n - p - 1))

  # Create a standardized residuals versus fitted values plot and highlight
  # influential points
  plot(st_resid ~ model$fitted.values,
       main = "Standardizd Residuals\nVersus Fitted Plot",
       xlab = "Fitted Values", ylab = "Standardized Residuals",
       col = ifelse(Di > qf(0.5, p + 1, n - p - 1), "red", "black"))
  abline(h = 0)

  # Create a Normal Q-Q plot
  qqnorm(st_resid)
  qqline(st_resid)
  # Validate the model using cross-validation
  model_cv <- ols(model$terms, model = TRUE, x = TRUE, y = TRUE)
  model_cv <- calibrate(model_cv, method = "crossvalidation", B = 10)

  # Create a calibration plot
  plot(model_cv, main = "Calibration Plot", xlab = "Predicted Response",
       ylab = "Observed Response", subtitles = FALSE, legend = FALSE)
  legend("bottomright", c("Apparent", "Bias-corrected", "Ideal"),
       lty = c(3, 1, 2), bty = "n", cex = 0.6)

  # Perform an ANOVA test and calculate adjusted R-squared
  ANOVA <- anova(model)
  RSS <- ANOVA[nrow(ANOVA), 2]
  SST <- 0
  for (i in 1:nrow(ANOVA)) {
    SST <- SST + ANOVA[i, 2]
  }
  adjusted_R_squared <- 1 - ((RSS / (n - p - 1)) / (SST) / (n - 1))

  # Return VIFs, influential points, ANOVA, and adjusted R-squared
  return(list(VIFs, influential_points, ANOVA, adjusted_R_squared))
}
```

## Simple Linear Regression (SLR)

### Predictor 1

```
par(mfrow = c(1, 3))

# Create and analyze a SLR model with the response and predictor 1
model1 <- lm(resp ~ pred1)
model1_analysis <- SLR_analysis(model1)
```

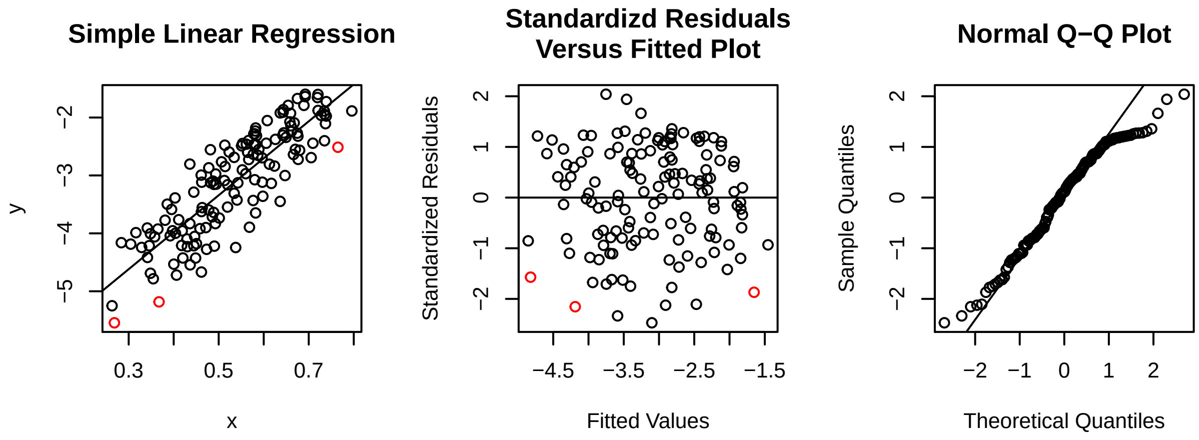


The assumptions of linearity and homoscedasticity are violated because the standardized residuals curve upward slightly and show a cone-shaped pattern. A Box-Cox transformation may help satisfy the conditions of SLR.

```
# Perform a Box-Cox transformation
powerTransform(lm(cbind(resp, pred1) ~ 1))
```

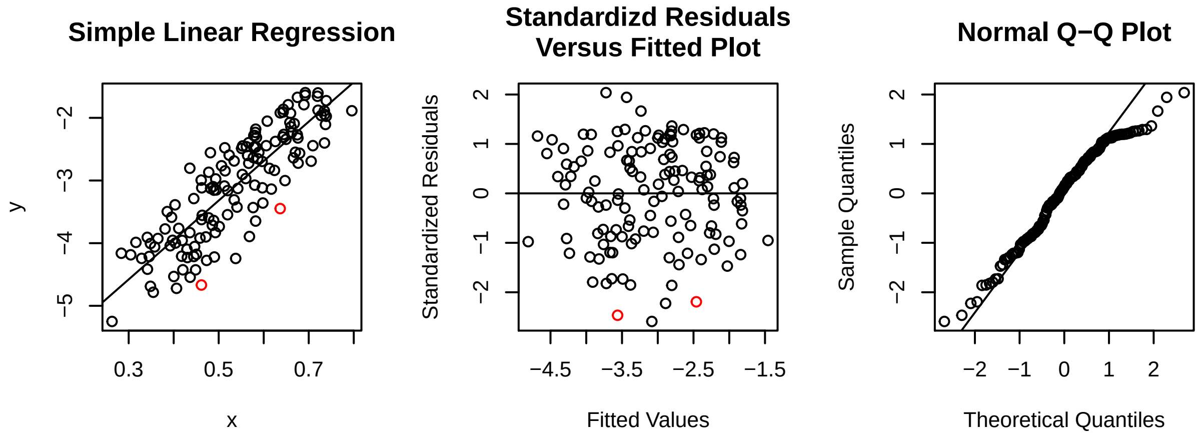
```
## Estimated transformation parameters
##      resp     pred1
## 0.1759367 0.5233750
```

```
# Taking the logarithm of the response and the square root of the predictor
# may help the model satisfy the assumptions of SLR

par(mfrow = c(1, 3))

# Create and analyze an SLR model with the transformed response and transformed
# predictor 1
model1 <- lm(log(resp) ~ sqrt(pred1))
model1_analysis <- SLR_analysis(model1)
```



The assumption of Normality of the errors may not be completely met since some points at the top right of the Normal Q-Q plot deviate from the line. Points coloured in red are influential points. To examine the effects of the 3 influential points, the model will be refit without them.

```
par(mfrow = c(1, 3))

# Refit a model without influential points
influential_points <- as.numeric(names(model1_analysis[[1]]))
model1_rm_influential <- lm(log(resp[-influential_points]) ~
                                    sqrt(pred1[-influential_points]))
model1_rm_influential_analysis <- SLR_analysis(model1_rm_influential)
```



```
# Compare the coefficients
model1$coefficients
```

```
## (Intercept) sqrt(pred1)
##   -6.534092    6.383859
```

```
model1_rm_influential$coefficients
```

```
##                      (Intercept) sqrt(pred1[-influential_points])
##                        -6.467804                         6.295737
```

The influential points have a small effect on the regression coefficients.

```
# ANOVA results
model1_analysis[[2]][1, 5]
```

```
## [1] 1.517986e-41
```

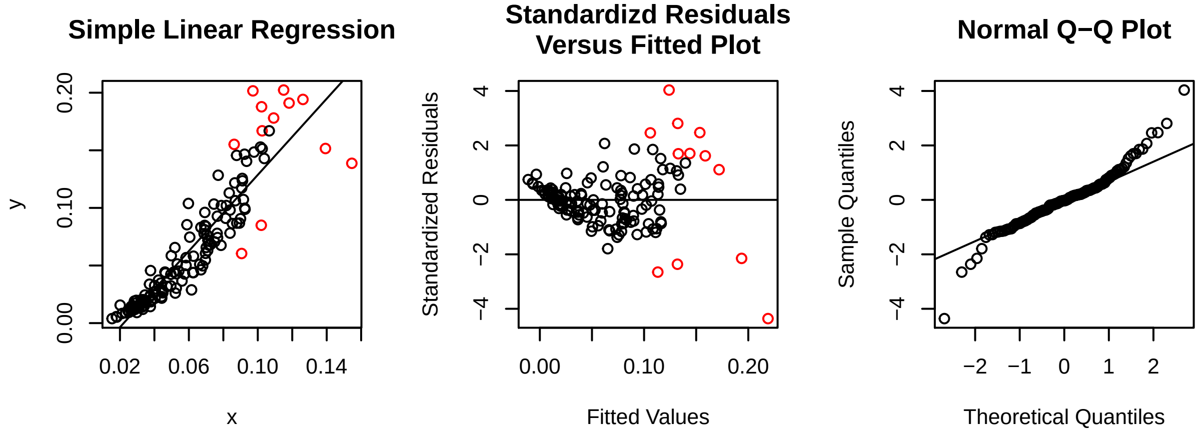
```
# R-squared
model1_analysis[[3]]
```

```
## [1] 0.7342904
```

The $p$-value for the ANOVA test is extremely low and $R^2$ is moderately high, so this model can be kept.

### Predictor 2

```
par(mfrow = c(1, 3))

# Create and analyze an LR model with the response and predictor 2
model2 <- lm(resp ~ pred2)
model2_analysis <- SLR_analysis(model2)
```

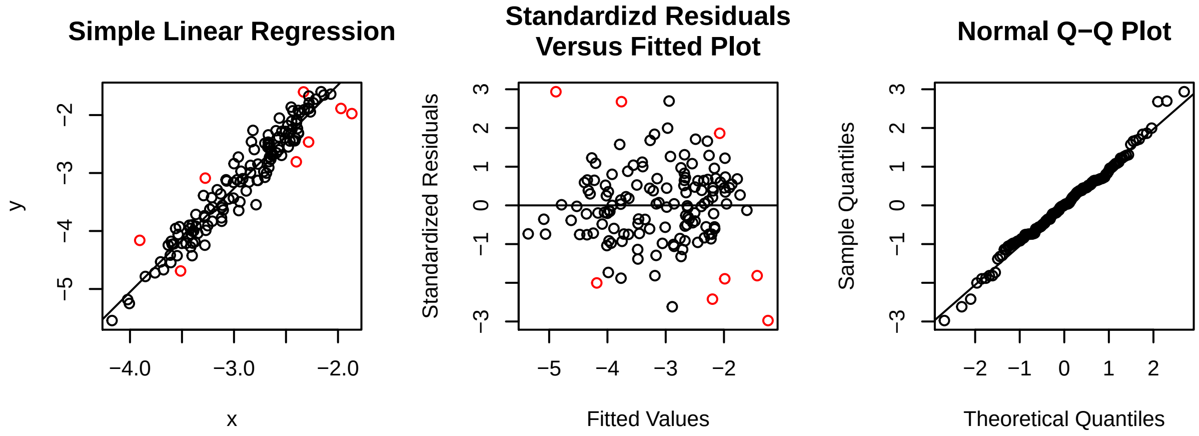


The assumptions of linearity and homoscedasticity are violated because the standardized residuals curve upward slightly and show a cone-shaped pattern. A Box-Cox transformation may help satisfy the conditions of SLR.

```
# Perform a Box-Cox transformation
powerTransform(lm(cbind(resp, pred2) ~ 1))
```

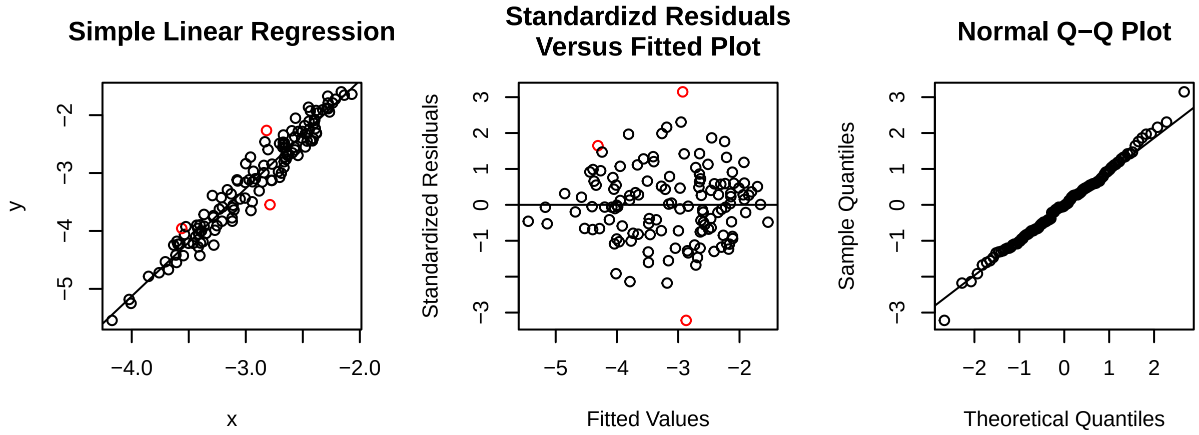
```
## Estimated transformation parameters
##       resp      pred2
## 0.03172192 0.02978311
```

```
# Taking the logarithm of both the response and the predictor may help the
# model satisfy the assumptions of SLR

par(mfrow = c(1, 3))

# Create and analyze an SLR model with the transformed response and transformed
# predictor 2
model2 <- lm(log(resp) ~ log(pred2))
model2_analysis <- SLR_analysis(model2)
```



Points coloured in red are influential points. Of the influential points that appear, there seem to be a few bad leverage points. Comparing this model to a modified version that removes the influential points can determine how dramatically the influential points change the least squares estimates.

```
par(mfrow = c(1, 3))

# Refit a model without influential points
influential_points <- as.numeric(names(model2_analysis[[1]]))
model2_rm_influential <- lm(log(resp[-influential_points]) ~
                              log(pred2[-influential_points]))
model2_rm_influential_analysis <- SLR_analysis(model2_rm_influential)
```



```
# Compare the coefficients
model2$coefficients
```

```
## (Intercept)  log(pred2)
##    2.087602    1.784743
```

```
model2_rm_influential$coefficients
```

```
##                     (Intercept) log(pred2[-influential_points])
##                        2.315701                        1.860412
```

This model may not be the best because the influential points impact the least squares estimates, especially the intercept.

```
# ANOVA p-value
model2_analysis[[2]][1, 5]
```

```
## [1] 5.829043e-78
```

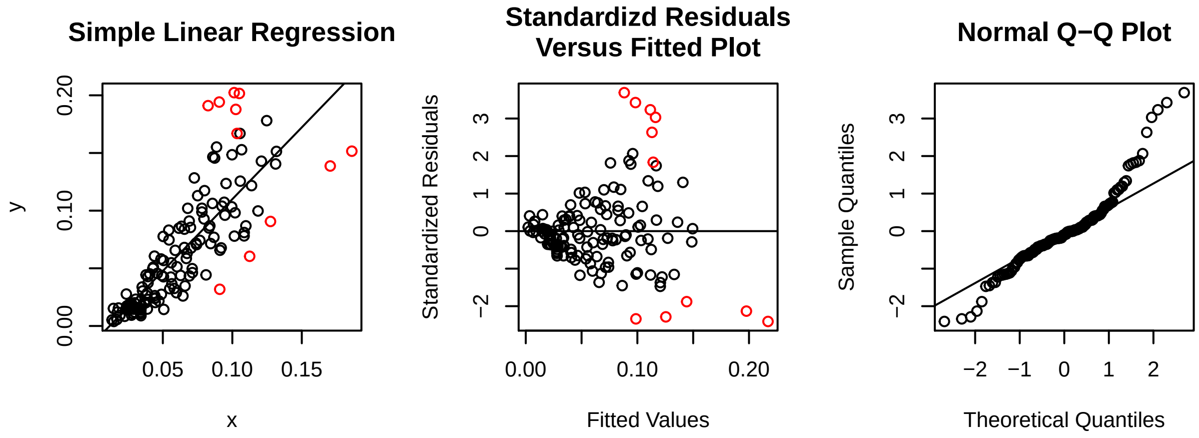
```
# R-squared
model2_analysis[[3]]
```

```
## [1] 0.9210513
```

Nonetheless, the $p$-value for the ANOVA test is extremely low and $R^2$ is very high.

### Predictor 3

```
par(mfrow = c(1, 3))

# Create and analyze an SLR model with the response and predictor 3
model3 <- lm(resp ~ pred3)
model3_analysis <- SLR_analysis(model3)
```

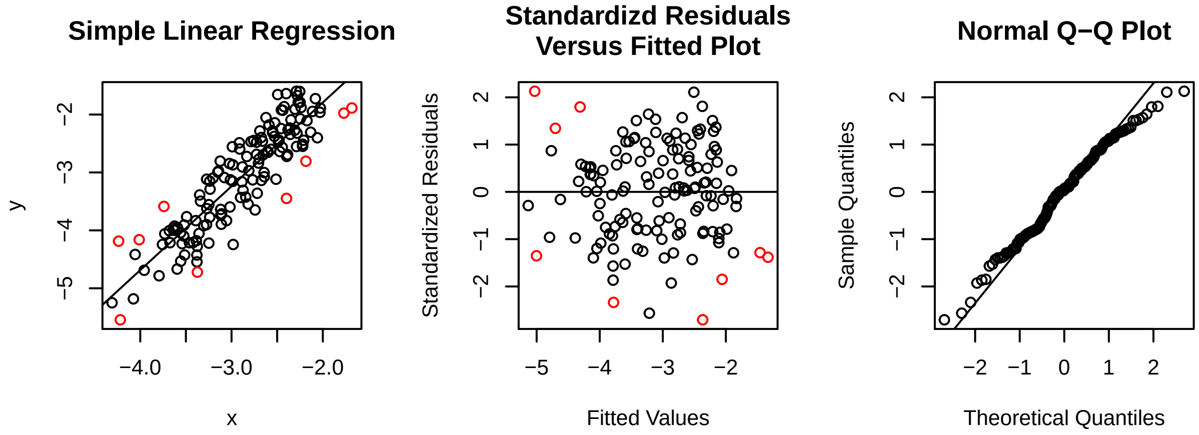


The untransformed model using predictor 3 displays the same problems as the previous two models, so a Box-Cox transformation may help satisfy the linearity and homoscedasticity assumptions.

```r
# Perform a Box-Cox transformation
powerTransform(lm(cbind(resp, pred3) ~ 1))
```

```
## Estimated transformation parameters
##       resp      pred3
## 0.08469673 0.11170378
```

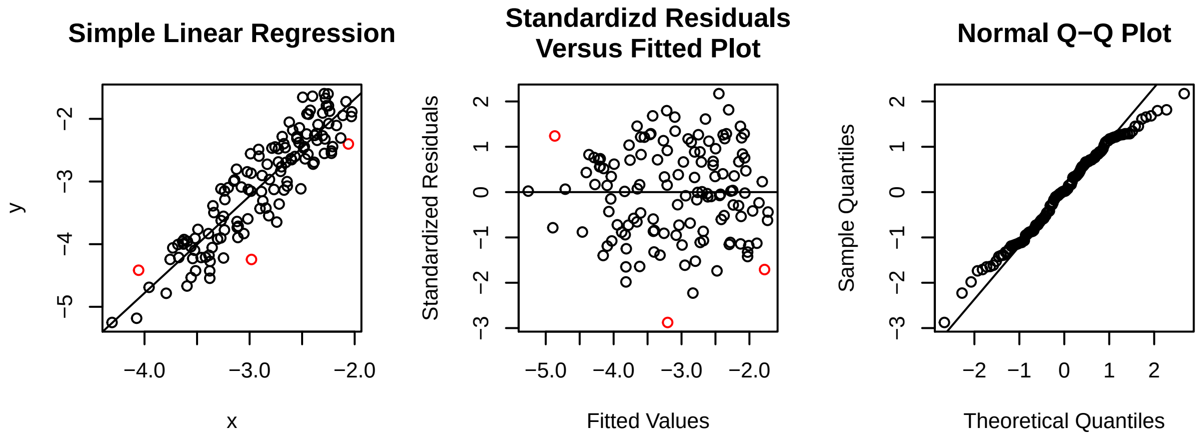
```r
# Taking the logarithm of both the response and the predictor may help the
# model satisfy the assumptions of SLR

par(mfrow = c(1, 3))

# Create and analyze an SLR model with the transformed response and transformed
# predictor 3
model3 <- lm(log(resp) ~ log(pred3))
model3_analysis <- SLR_analysis(model3)
```



This model can be compared to the same model without influential points to find out how much the regression coefficients are affected.

```r
par(mfrow = c(1, 3))
```

```
# Refit a model without influential points
influential_points <- as.numeric(names(model3_analysis[[1]]))
model3_rm_influential <- lm(log(resp[-influential_points]) ~
                              log(pred3[-influential_points]))
model3_rm_influential_analysis <- SLR_analysis(model3_rm_influential)
```



```
# Compare the coefficients
model3$coefficients
```

```
## (Intercept)  log(pred3)
##    1.102931    1.447342
```

```
model3_rm_influential$coefficients
```

```
##                      (Intercept) log(pred3[-influential_points])
##                         1.410744                        1.547620
```

The influential points have a strong effect on the intercept.

```
# ANOVA p-value
model3_analysis[[2]][1, 5]
```

```
## [1] 7.090822e-50
```

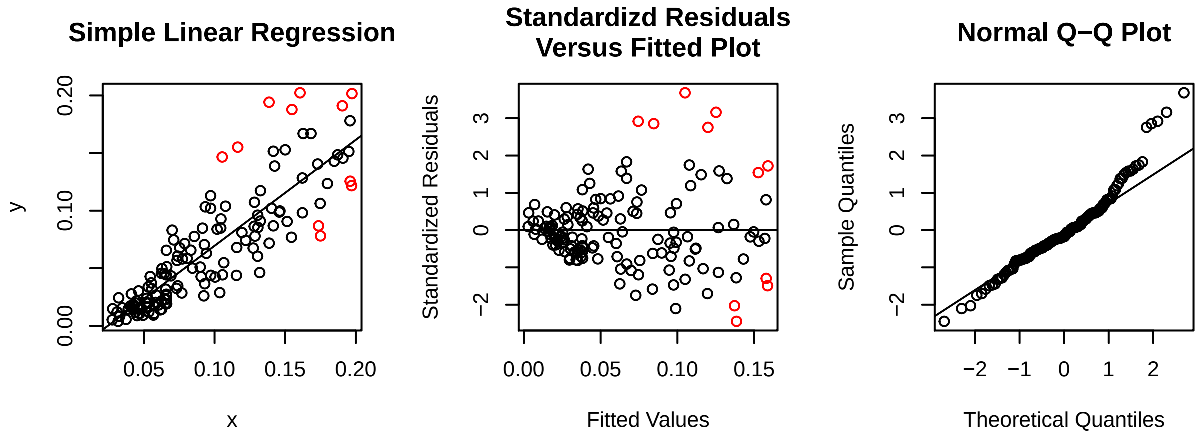
```
# R-squared
model3_analysis[[3]]
```

```
## [1] 0.7986582
```

The $p$-value for the ANOVA test is extremely low, and $R^2$ is relatively high.

### Predictor 4

```
par(mfrow = c(1, 3))

# Create and analyze an SLR model with the response and predictor 4
model4 <- lm(resp ~ pred4)
model4_analysis <- SLR_analysis(model4)
```

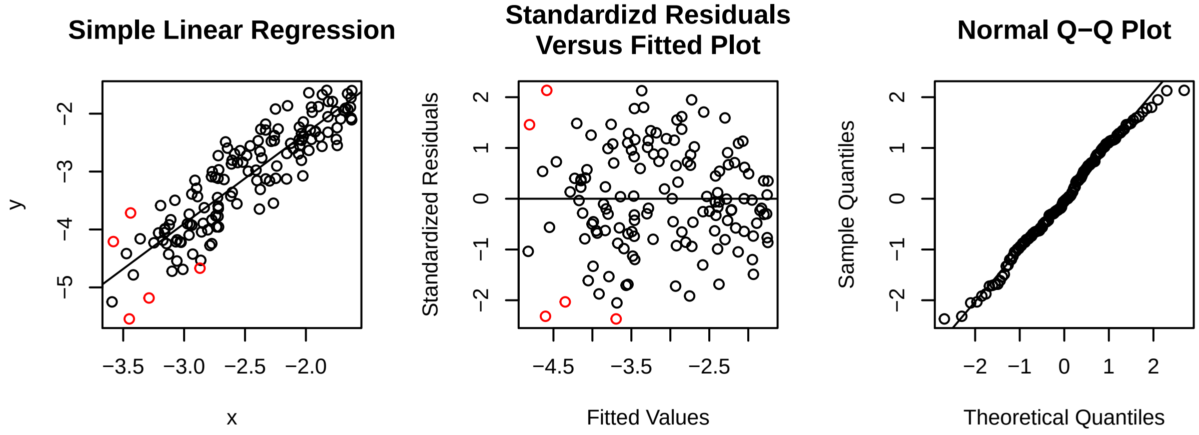


A Box-Cox transformation can remedy the slight violation of linearity and the violation of homoscedasticity.

```
# Perform a Box-Cox transformation
powerTransform(lm(cbind(resp, pred4) ~ 1))
```

```
## Estimated transformation parameters
##       resp      pred4
## 0.20033647 0.03576985
```

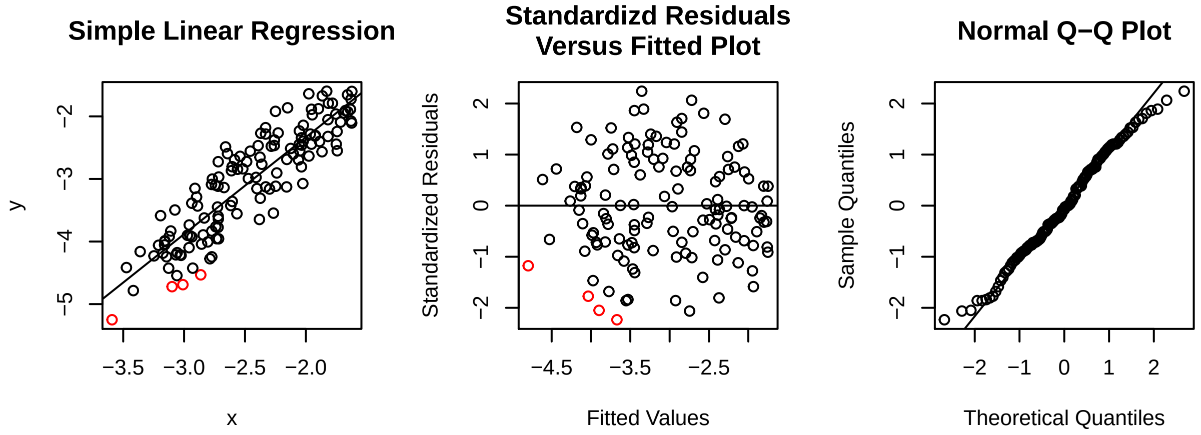
```
# Taking the logarithm of both the response and the predictor may help the
# model satisfy the assumptions of SLR

par(mfrow = c(1, 3))

# Create and analyze an SLR model with the transformed response and transformed
# predictor 4
model4 <- lm(log(resp) ~ log(pred4))
model4_analysis <- SLR_analysis(model4)
```



There appear to be three or four high leverage points. The other influential points could be classified as outliers.

```
par(mfrow = c(1, 3))

# Refit a model without influential points
```

```
influential_points <- as.numeric(names(model4_analysis[[1]]))
model4_rm_influential <- lm(log(resp[-influential_points]) ~
                              log(pred4[-influential_points]))
model4_rm_influential_analysis <- SLR_analysis(model4_rm_influential)
```



```
# Compare the coefficients
model4$coefficients
```

```
## (Intercept)  log(pred4) 
##   0.7894856   1.5627676
```

```
model4_rm_influential$coefficients
```

```
##                     (Intercept) log(pred4[-influential_points]) 
##                       0.7626501                       1.5480166
```

The model is relatively unaffected by the presence influential points.

```
# ANOVA p-value
model4_analysis[[2]][1, 5]
```

```
## [1] 1.167901e-48
```

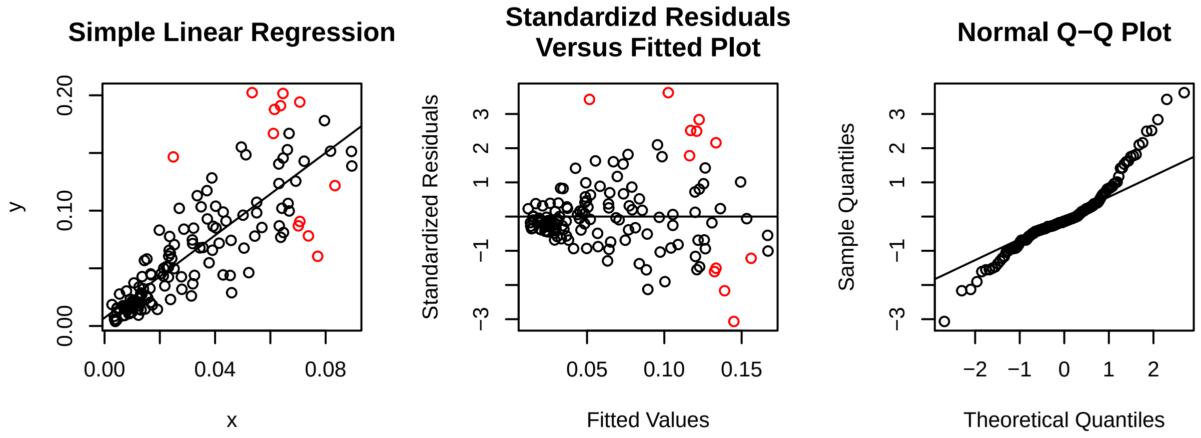
```
# R-squared
model4_analysis[[3]]
```

```
## [1] 0.7903306
```

The $p$-value for the ANOVA test is extremely low, and $R^2$ is high.

### Predictor 5

```
par(mfrow = c(1, 3))

# Create and analyze an SLR model with the response and predictor 5
model5 <- lm(resp ~ pred5)
model5_analysis <- SLR_analysis(model5)
```

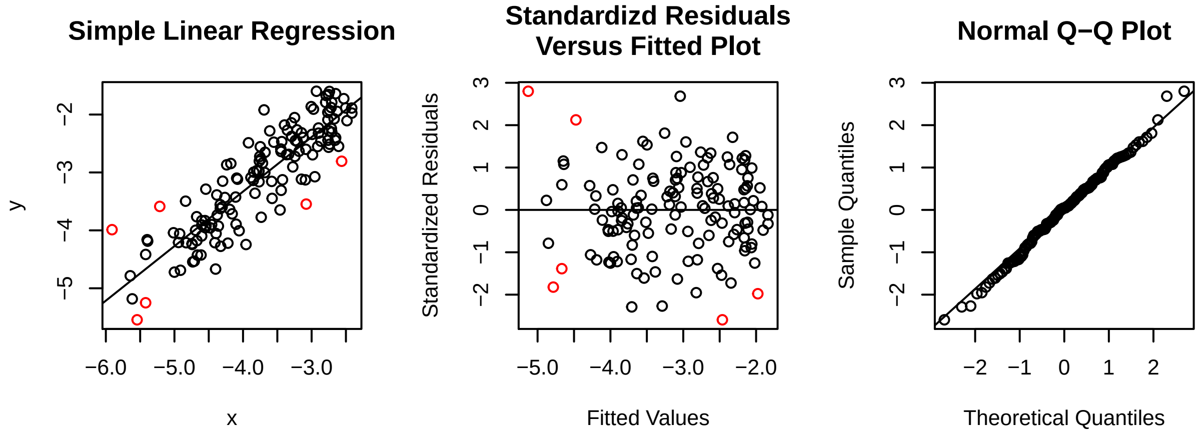


Like the other predictors, a Box-Cox transformation could allow the model to better satisfy the conditions of SLR.

```
# Perform a Box-Cox transformation
powerTransform(lm(cbind(resp, pred5) ~ 1))
```

```
## Estimated transformation parameters
##      resp     pred5
## 0.1275932 0.1596356
```

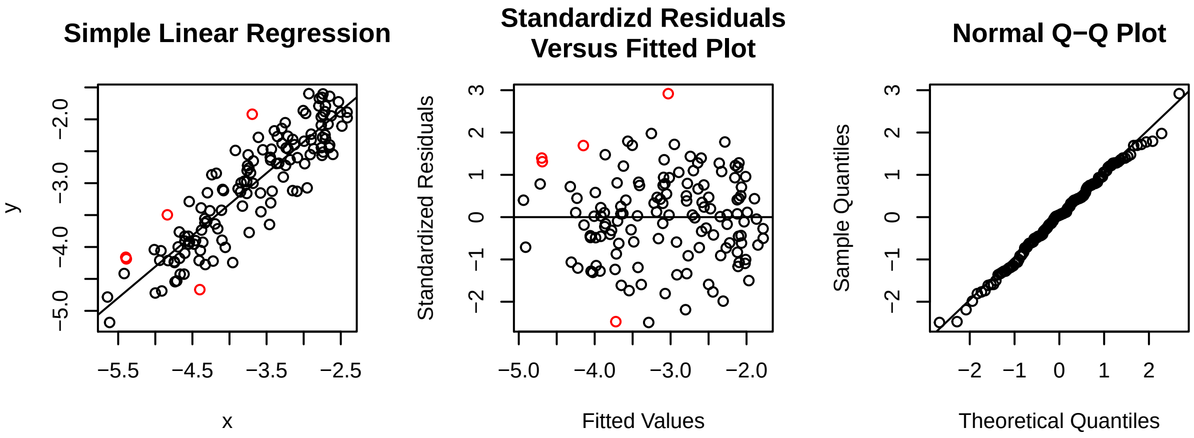
```
# Taking the logarithm of both the response and the predictor may help the
# model satisfy the assumptions of SLR

par(mfrow = c(1, 3))

# Create and analyze an SLR model with the transformed response and transformed
# predictor 5
model5 <- lm(log(resp) ~ log(pred5))
model5_analysis <- SLR_analysis(model5)
```



The point at the top right seems to be a bad leverage point. The other influential points could be classified as outliers.

```
par(mfrow = c(1, 3))
```

```
# Refit a model without influential points
influential_points <- as.numeric(names(model5_analysis[[1]]))
model5_rm_influential <- lm(log(resp[-influential_points]) ~
                              log(pred5[-influential_points]))
model5_rm_influential_analysis <- SLR_analysis(model5_rm_influential)
```



```
# Compare the coefficients
model5$coefficients
```

```
## (Intercept)  log(pred5)
##   0.4368510   0.9417598
```

```
model5_rm_influential$coefficients
```

```
##                     (Intercept) log(pred5[-influential_points])
##                       0.5782154                       0.9772704
```

The intercept changed considerably while the slope increased slightly after influential points were removed. There may be other models that are not as affected by influential points.

```
# ANOVA p-value
model5_analysis[[2]][1, 5]
```

```
## [1] 2.115963e-47
```

```
# R-squared
model5_analysis[[3]]
```

```
## [1] 0.7813583
```

The $p$-value for the ANOVA test is extremely low, and $R^2$ is moderately high.

### Summary

`model1` does not appear to have influential points as bad as those in other models. However, its $R^2$ is lower than those of other models, and its errors may not be entirely Normal.

`model2` has a few bad leverage points that change the least squares estimates. Its $R^2$ is the highest of all the SLR models, however.

`model3` has influential points that affect the intercept significantly. Its $R^2$ is reasonably high.

The least squares estimates of `model4` do not change much after influential points are removed. Again, its $R^2$ value is reasonably high.

model5 has bad leverage points that mostly impact the intercept. Its $R^2$ is similar to those of the previous two models.

The $p$-values for all ANOVA tests indicate that there is strong evidence against the null hypothesis that $H_0 : \beta_1 = 0$.

```
# Means and standard deviations of transformed variables
c(mean(sqrt(pred1)), sd(sqrt(pred1)))
```

```
## [1] 0.5370839 0.1226351
```

```
c(mean(log(pred2)), sd(log(pred2)))
```

```
## [1] -2.9096775  0.4912817
```

```
c(mean(log(pred3)), sd(log(pred3)))
```

```
## [1] -2.9076441  0.5641228
```

```
c(mean(log(pred4)), sd(log(pred4)))
```

```
## [1] -2.4923153  0.5197258
```

```
c(mean(log(pred5)), sd(log(pred5)))
```

```
## [1] -3.7613357  0.8575298
```

```
c(mean(log(resp)), sd(log(resp)))
```

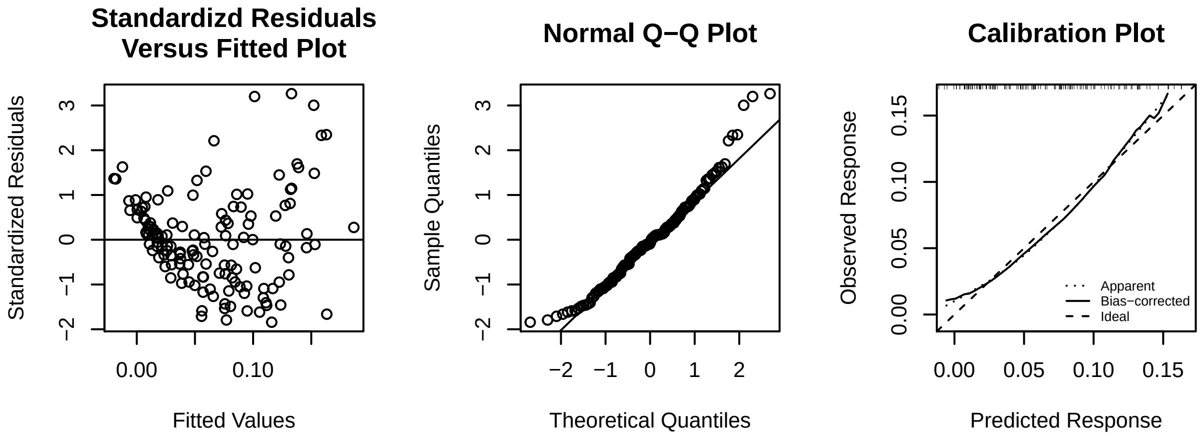
```
## [1] -3.1054240  0.9136173
```

## Multiple Linear Regression (MLR)

### Variable Selection

First, an MLR model must be fit using all predictors.

```
par(mfrow = c(1, 3))

# Create and analyze an MLR model with the response and all 5 predictors
model12345 <- lm(resp ~ pred1 + pred2 + pred3 + pred4 + pred5)
model12345_analysis <- MLR_analysis(model12345)
```



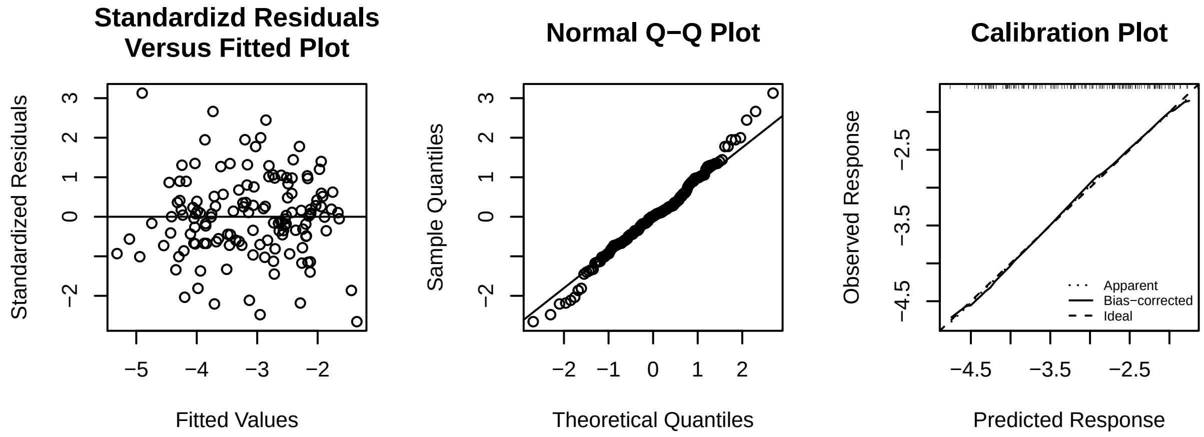
```
##
## n=140   Mean absolute error=0.005   Mean squared error=3e-05
## 0.9 Quantile of absolute error=0.008
```

This model must be transformed and refit because it violates the assumptions of linearity, homoscedasticity, independence of errors, and Normality of errors.

```
# Perform a Box-Cox transformation
powerTransform(lm(cbind(resp, pred1, pred2, pred3, pred4, pred5) ~ 1))
```

```
## Estimated transformation parameters
##       resp      pred1      pred2      pred3      pred4      pred5
## 0.10708252 0.45587044 0.14251473 0.17767577 0.05217486 0.24229721
```

```
# The Box-Cox transformation suggests to use `log(resp)`, 'sqrt(pred1)`,
# `log(pred2)`, `log(pred3)`, `log(pred4)`, and `log(pred5)`

# This matches the transformations used in SLR

par(mfrow = c(1, 3))

# Create and analyze an MLR model with all 5 predictors
model12345 <- lm(log(resp) ~ sqrt(pred1) + log(pred2) + log(pred3) + log(pred4)
                 + log(pred5))
model12345_analysis <- MLR_analysis(model12345)
```



```
##
## n=140   Mean absolute error=0.027   Mean squared error=0.00109
## 0.9 Quantile of absolute error=0.055
```

The transformed model now satisfies all assumptions of MLR. To select the model, stepwise selection can be performed using the AIC and BIC.

```
# Use the AIC to select predictors
modelAIC <- step(model12345, direction = "both", trace = 0, k = 2)
modelAIC$call
```

```
## lm(formula = log(resp) ~ sqrt(pred1) + log(pred2) + log(pred4))
```

```
# Use the BIC to select predictors
n <- nrow(data)
modelBIC <- step(model12345, direction = "both", trace = 0, k = log(n))
modelBIC$call
```

```
## lm(formula = log(resp) ~ sqrt(pred1) + log(pred2))
```

The model with the smallest AIC uses the transformed versions of predictors 1, 2, and 4. Meanwhile, the model with the smallest BIC only uses predictors 1 and 2.

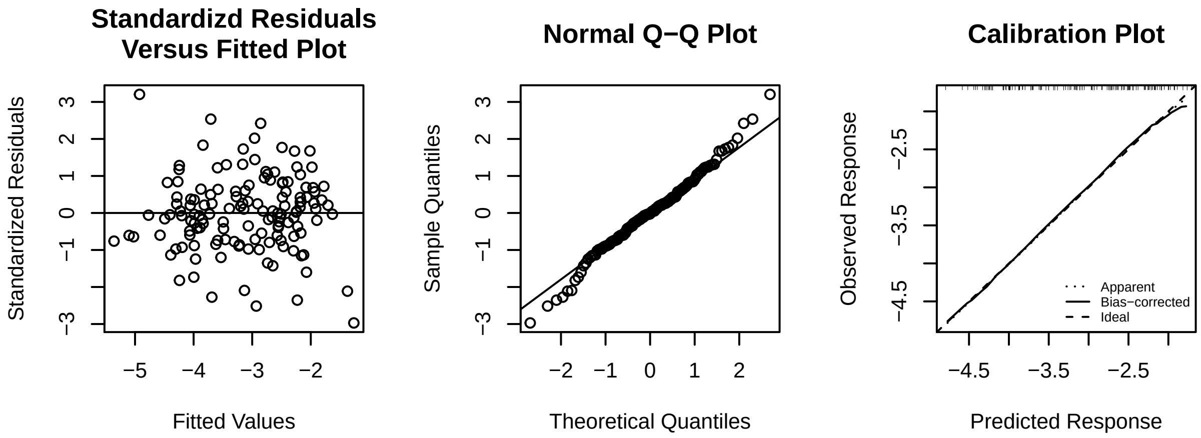
The original model can also be shrunk using LASSO to select another potential model.

```
set.seed(8249)

# Use LASSO to select predictors
pred_matrix <- cbind(sqrt(pred1), log(pred2), log(pred3), log(pred4),
                     log(pred5))
modelLASSO <- glmnet::cv.glmnet(pred_matrix, log(resp))
modelLASSO <- coef(modelLASSO, s = "lambda.1se")
```

LASSO has selected the model with predictors 1, 2, and 4, which is the same model found using stepwise selection with the AIC.

### Predictors 1 and 2

```
par(mfrow = c(1, 3))

# Create and analyze an MLR model
model12 <- lm(log(resp) ~ sqrt(pred1) + log(pred2))
model12_analysis <- MLR_analysis(model12)
```



```
##
## n=140   Mean absolute error=0.023   Mean squared error=0.00095
## 0.9 Quantile of absolute error=0.038
```

```
# Get VIFs
model12_analysis[[1]]
```

```
## sqrt(pred1)  log(pred2)
##    3.750293    3.750293
```

```
# Get all influential points
model12_analysis[[2]]
```

```
## named integer(0)
```

```
# ANOVA p-values
model12_analysis[[3]][5]
```

```
##                  Pr(>F)
## sqrt(pred1) < 2.2e-16 ***
## log(pred2)  < 2.2e-16 ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

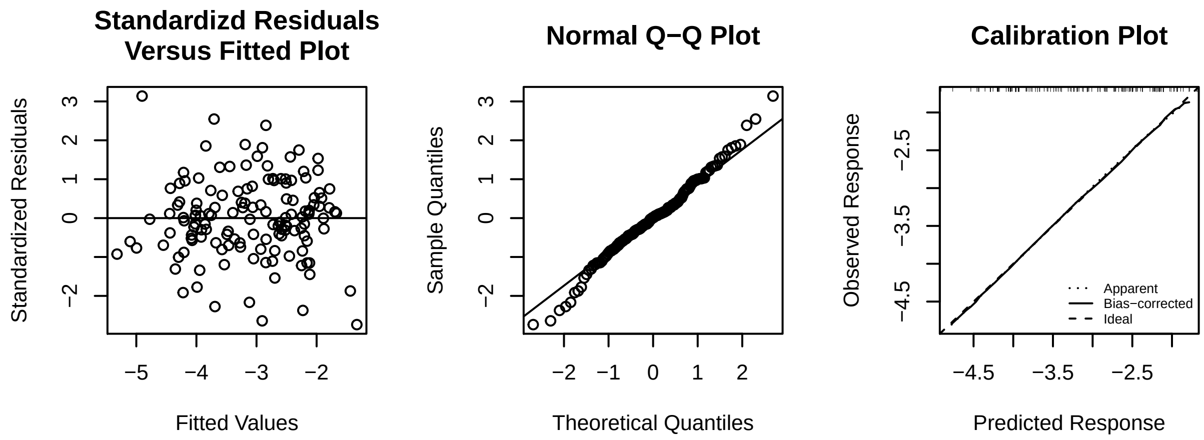
```
# Adjusted R-squared
model12_analysis[[4]]
```

```
## [1] 0.9999961
```

Both VIFs are less than 5. Using Cook's distance, no influential points were found. The calibration plot shows that the observed probabilities largely match the predicted probabilities. From the ANOVA result, all predictors are significant, and $R^2_{adj}$ is high.

### Predictors 1, 2, and 4

```
par(mfrow = c(1, 3))

# Create and analyze an MLR model
model124 <- lm(log(resp) ~ sqrt(pred1) + log(pred2) + log(pred4))
model124_analysis <- MLR_analysis(model124)
```



```
##
## n=140   Mean absolute error=0.011   Mean squared error=0.00022
## 0.9 Quantile of absolute error=0.017
```

```
# Get VIFs
model124_analysis[[1]]
```

```
## sqrt(pred1)  log(pred2)  log(pred4)
##    4.180663    6.409951    5.922429
```

Two of the three VIFs are greater than 5, so this model will not be considered due to the presence of multicollinearity.

### Summary

`model124` had the smallest AIC while `model12` had the smallest BIC. However, `model124` exhibited multicollinearity while `model12` did not. Hence, `model12` appears to be the best MLR model.