Regression Diagostics

November 17, 2018

```
mydat <- read.csv("data.csv",sep=",")
predictor_x <- mydat$Population
response_y <- mydat$Peakmembership
```

Regression Diagnostics

The question we want to seek is whether there may exist a relationship between the predictor and the response variable. If you suspect a relationship, state it as an expression.

The steps for regression diagnostics follows this flow chart:

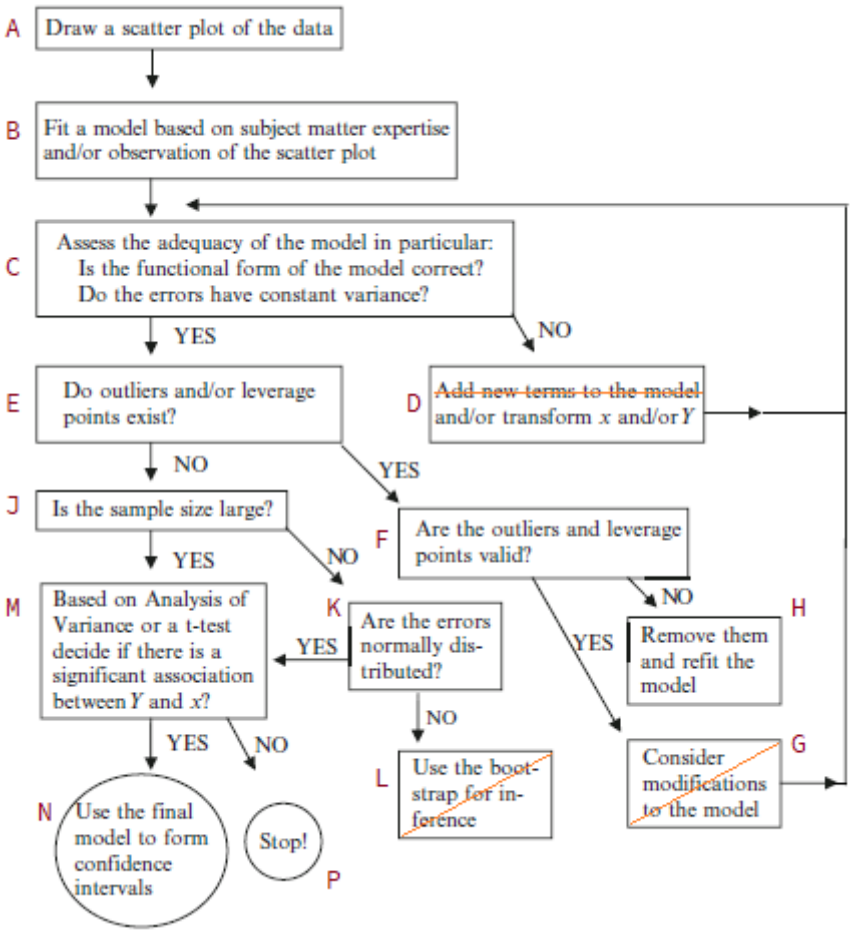
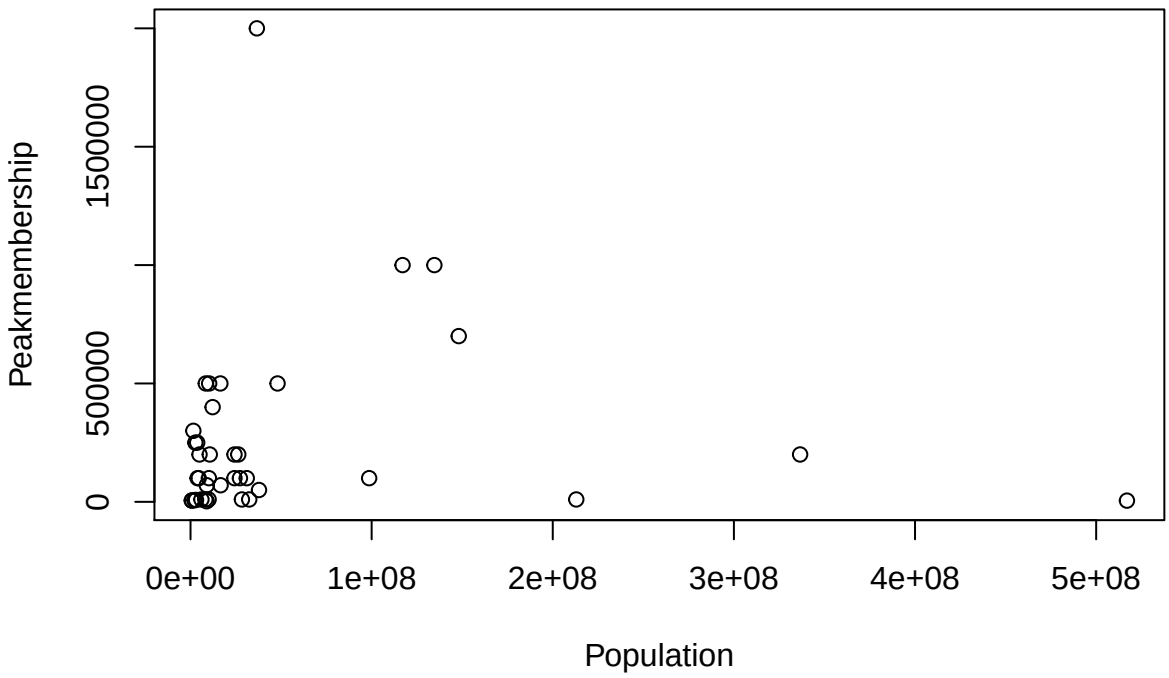


Figure 1: Regression Flow Chart

A: Draw a scatter plot of the data

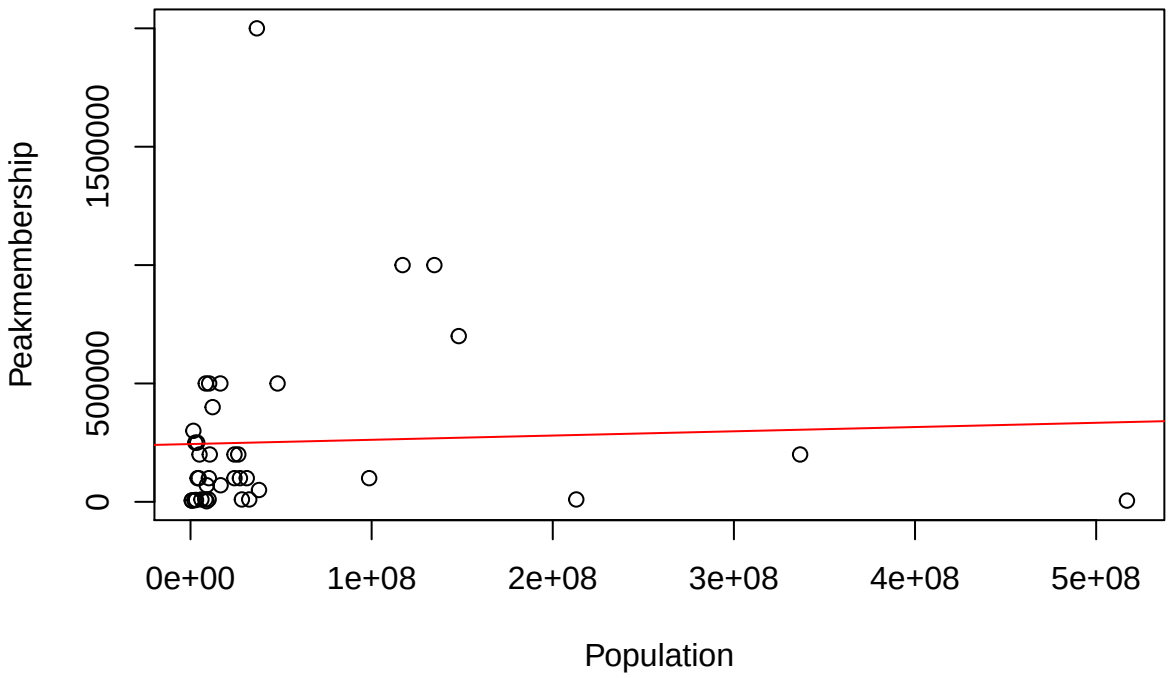
```
plot(x = predictor_x, y = response_y, xlab = "Population", ylab = "Peakmembership")
```



B: Fit a model based on subject matter expertise and/or observation of the scatter plot

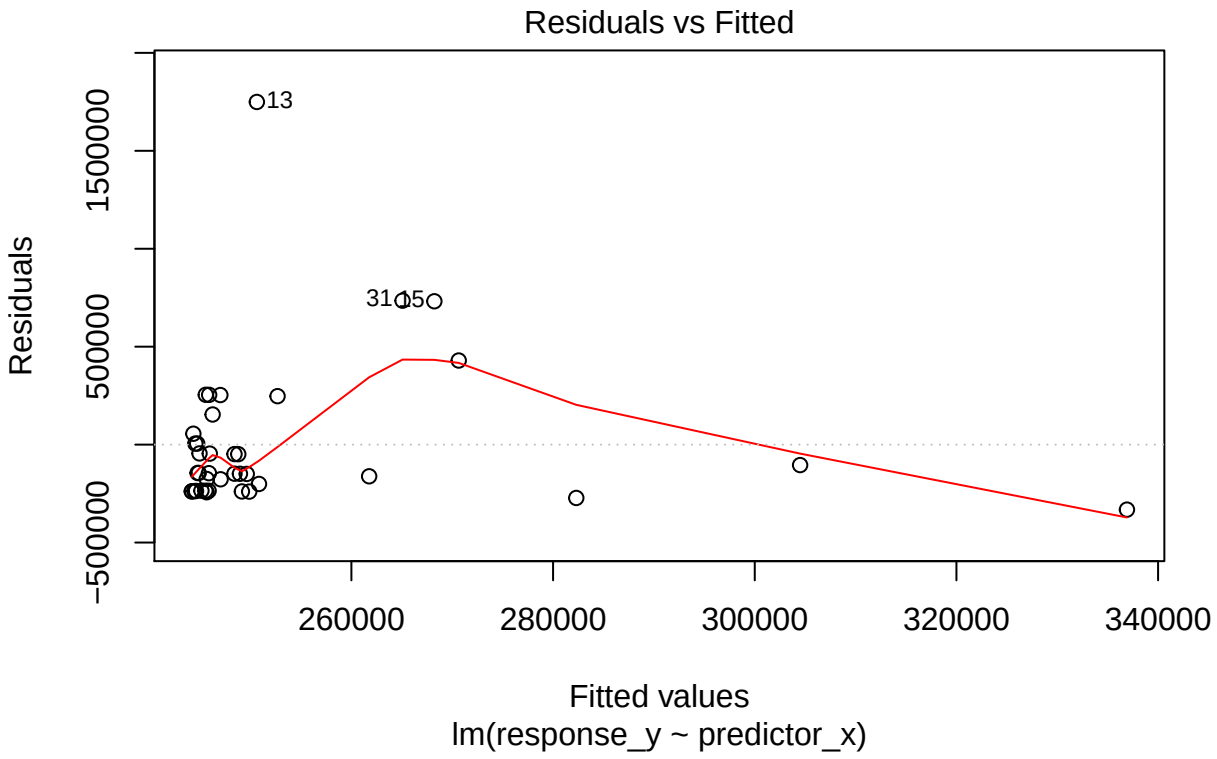
We will try to fit a linear regression line to the data.

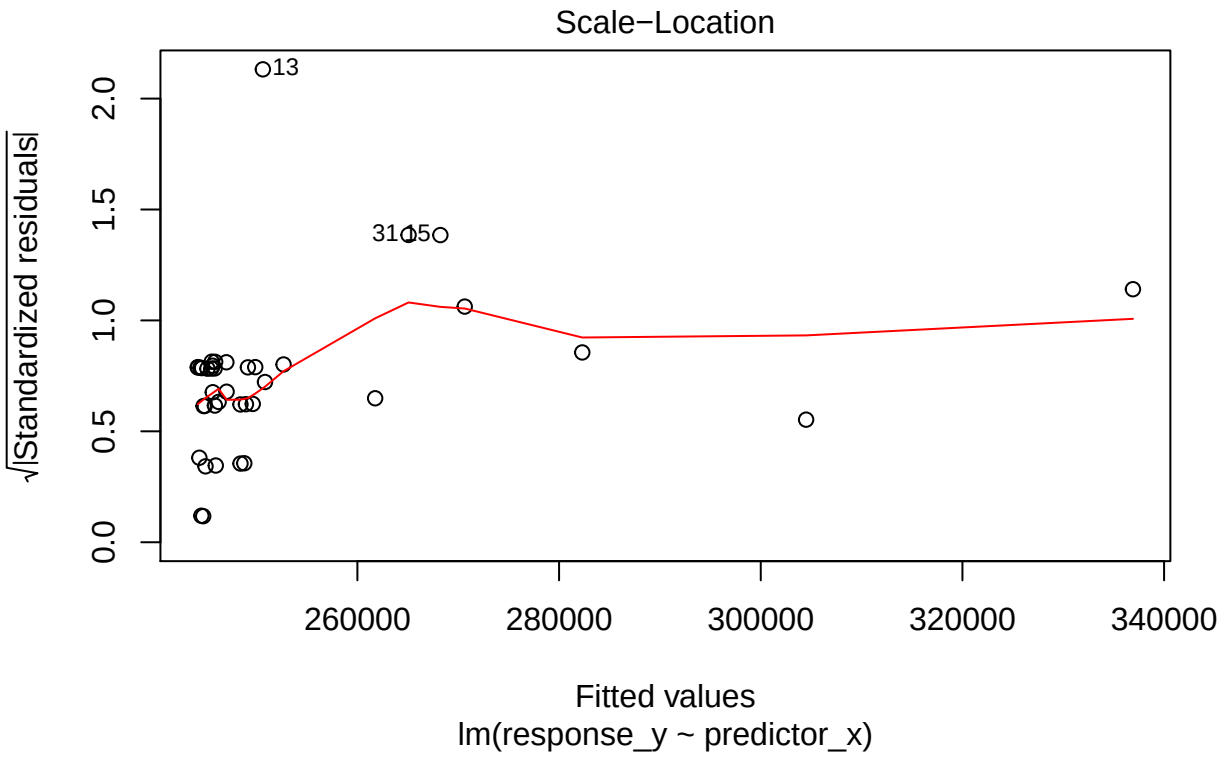
```
plot(x = predictor_x, y = response_y, xlab = "Population", ylab = "Peakmembership")
lm_reg <- lm(formula = response_y ~ predictor_x)
abline(lm_reg, col="red")
```



C: Assess the adequacy of the model in particular: Is the functional form of the model correct? Do the errors have constant variance?

```
plot(lm_reg, which = c(1,3))
```


Residuals vs Fitted
Residuals
1500000
500000
−500000
13
31
5
260000
280000
300000
320000
340000
Fitted values
lm(response_y ~ predictor_x)

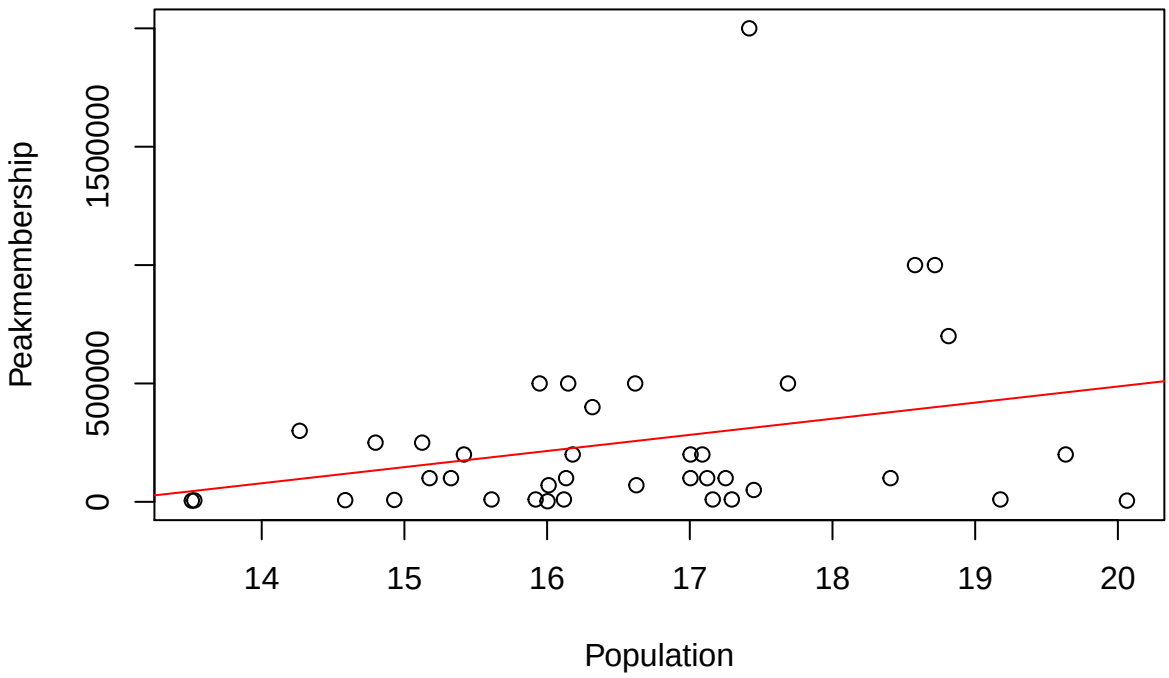


Conclusions from plots of raw data:
1.From the plot of residuals vs fitted values, even though the variance of the residuals appears to be constant, but most of the residuals are clustered to the left.
2.From the square root of standardized residuals vs fitted values plot, the residuals does not appear to be randomly spread.
This shows that we need to apply transformation to x and/or Y.

D: Transform x and/or Y

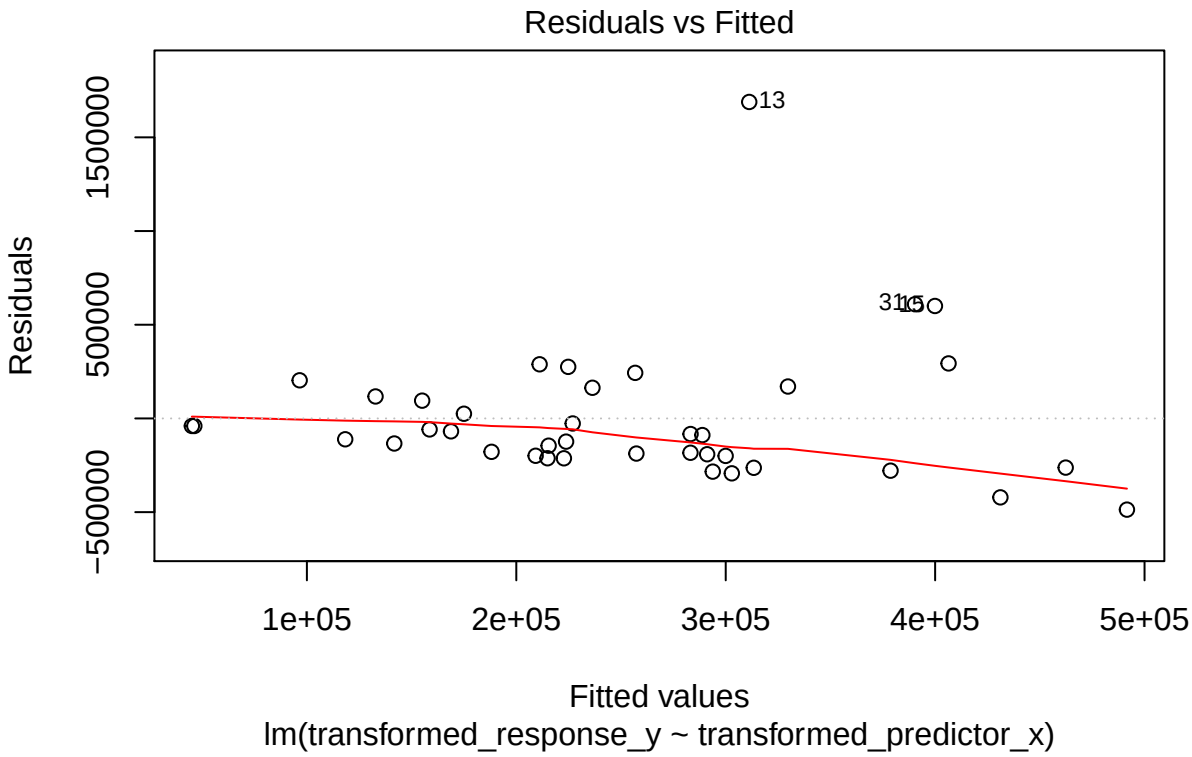
We want the predictor_x to be more spread out, so we apply log transformation to x variable.

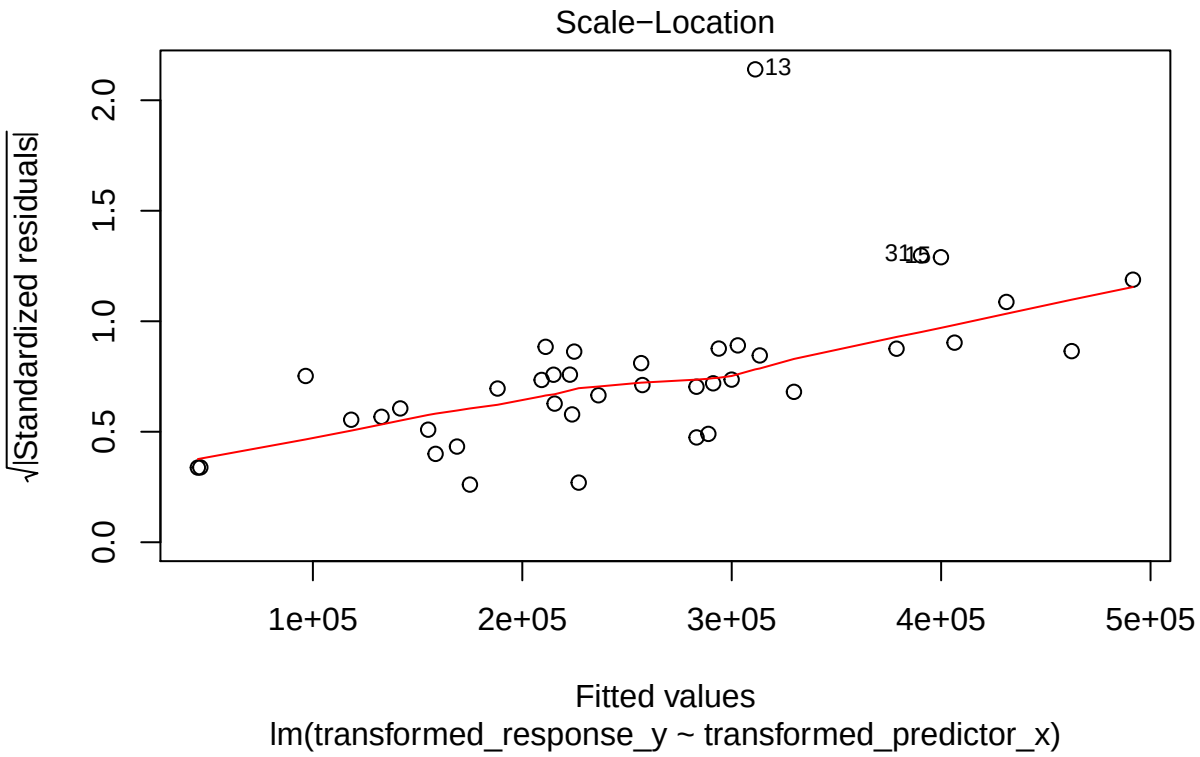
```
transformed_predictor_x <- log(predictor_x)
transformed_response_y <- response_y
new_lm_reg <- lm(formula = transformed_response_y ~ transformed_predictor_x)
plot(x = transformed_predictor_x, y = transformed_response_y, xlab = "Population",
     ylab = "Peakmembership")
abline(new_lm_reg, col = 'red')
```



C: Assess the adequacy of the model in particular: Is the functional form of the model correct? Do the errors have constant variance?

```
plot(new_lm_reg, c(1,3))
```

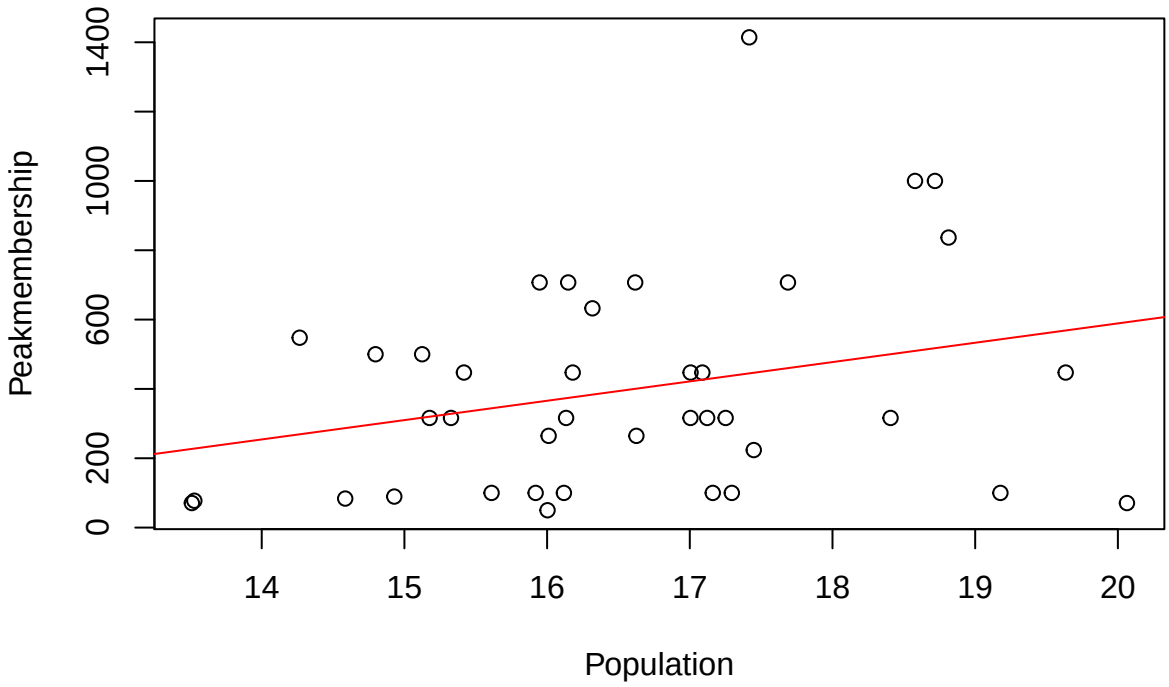

Residuals vs Fitted
Residuals
1500000
500000
−500000
13
31
15
1e+05
2e+05
3e+05
4e+05
5e+05
Fitted values
lm(transformed_response_y ~ transformed_predictor_x)



Conclusions from plots of transformed data:
1. It seems that log transformation to X variable does spread out the data point along the x-axis.
2. However, it seems that the assumption of constant variance is violated. The residuals seem to increase as fitted value increases. The plot of square root of standardized residuals vs fitted value suggests the same. This means we need to transform x and/or Y in order to eliminate such relationship between resuduals and fitted value.

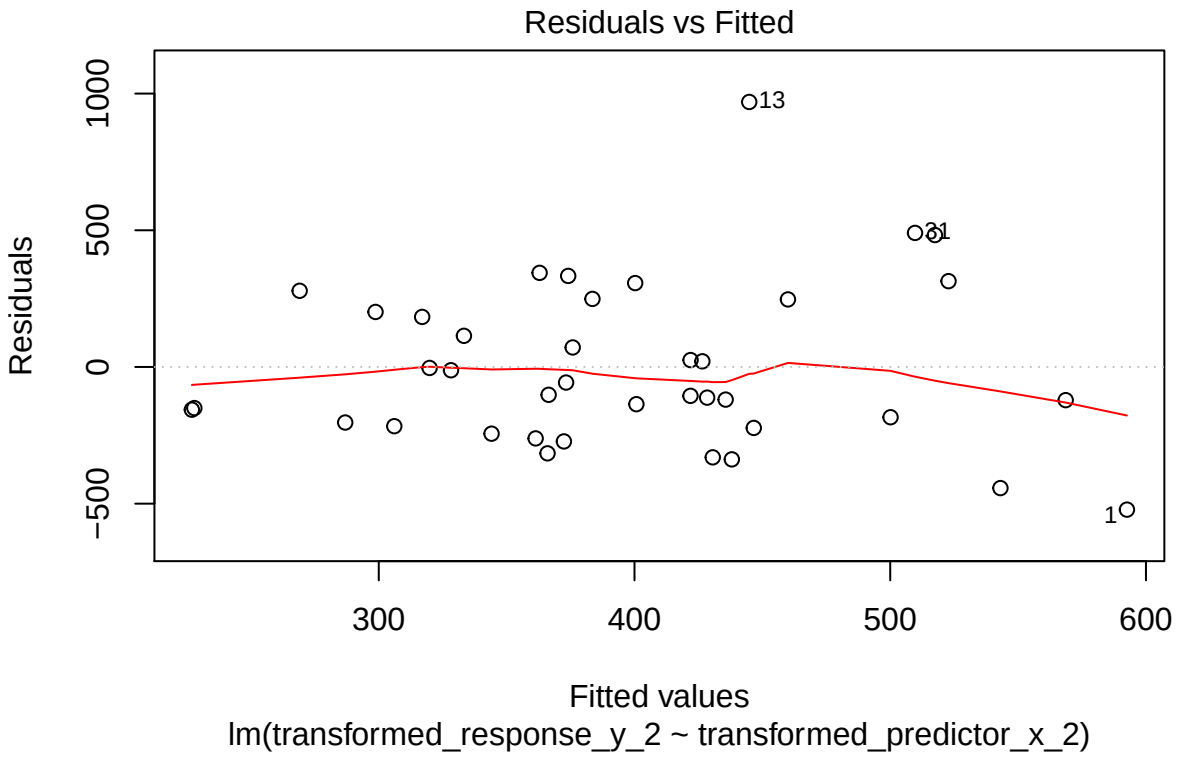
D: Transform x and/or Y

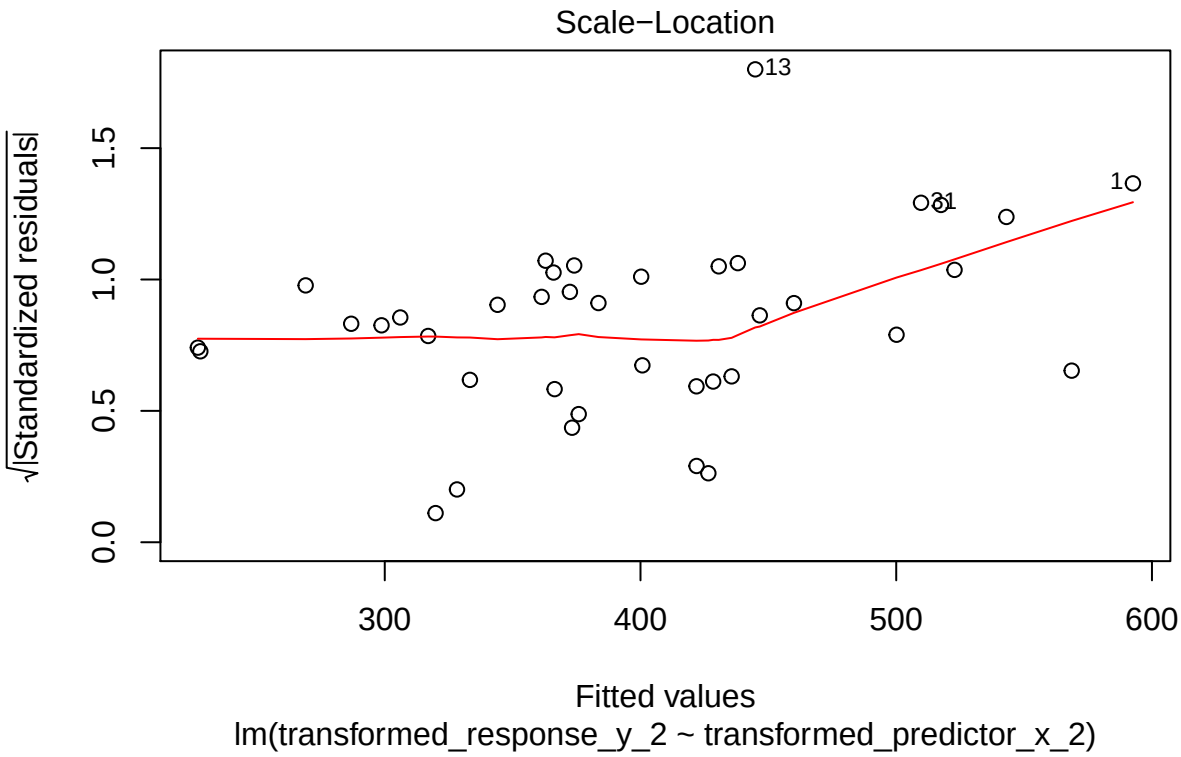
```
transformed_predictor_x_2 <- log(predictor_x)
transformed_response_y_2 <- sqrt(response_y)
new_lm_reg_2 <- lm(formula = transformed_response_y_2 ~ transformed_predictor_x_2)
plot(x = transformed_predictor_x_2, y = transformed_response_y_2, xlab = "Population",
     ylab = "Peakmembership")
abline(new_lm_reg_2, col = 'red')
```



C: Assess the adequacy of the model in particular: Is the functional form of the model correct? Do the errors have constant variance?

```
plot(new_lm_reg_2, c(1,3))
```


Residuals vs Fitted
Residuals
1000
500
0
−500
13
31
1
300
400
500
600
Fitted values
lm(transformed_response_y_2 ~ transformed_predictor_x_2)



Conclusions from plots of 2nd time transformed data:
1. It seems that the residuals are randomly spreaded; 2. It seems that the increasing square root of standardized residuals of the second part is caused by leverage points. 3. It seems that the assumption of homoscedasticity is satisfied;

E: Do outliers and/or leverage points exist?

```
influence_info <- influence.measures(new_lm_reg_2)
# DFBETAS0 > 1
influence_info$infmat[,6][influence_info$infmat[,1] > 1]
```

```
## named numeric(0)
```

```
# DFBETAS1 > 1
influence_info$infmat[,6][influence_info$infmat[,2] > 1]
```

```
## named numeric(0)
```

```
# DFFITS > 1
influence_info$infmat[,6][influence_info$infmat[,3] > 1]
```

```
## named numeric(0)
```

```
# Cook's Distance > 1
influence_info$infmat[,6][influence_info$infmat[,5] > 1]
```

```
## named numeric(0)
```

```
# Leverage points > 4/n
influence_info$infmat[,6][influence_info$infmat[,6] > 4/39]
```

```
##         1         2        17        26
## 0.1552869 0.1253987 0.1237346 0.1248661
```

```
# Outliers sd > 2
rstandard(new_lm_reg_2)[rstandard(new_lm_reg_2) > 2]
```

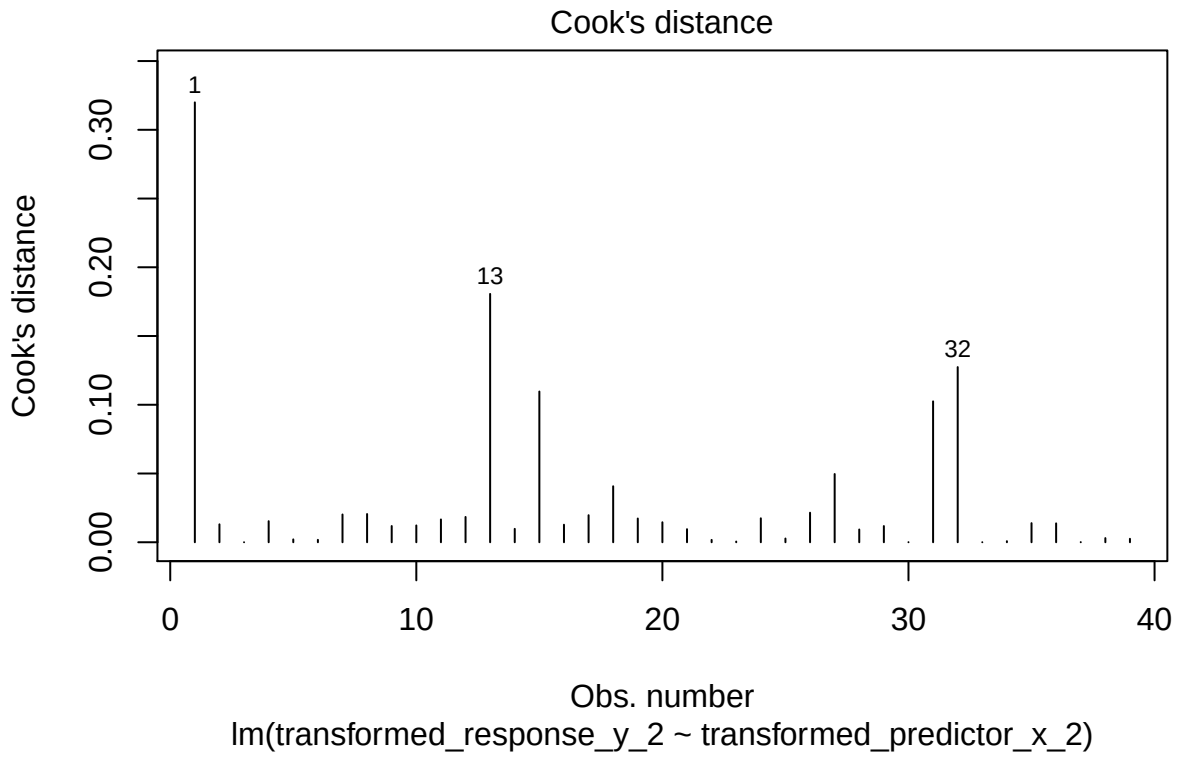
```
##       13
## 3.239567
```

There are four leverage points and one outlier.

F: Are the outliers and leverage points valid?

We will be focusing on points with largre cook's distance, and especially on point 1, 2, 13, 17, 26

```
plot(new_lm_reg_2, 4)
```



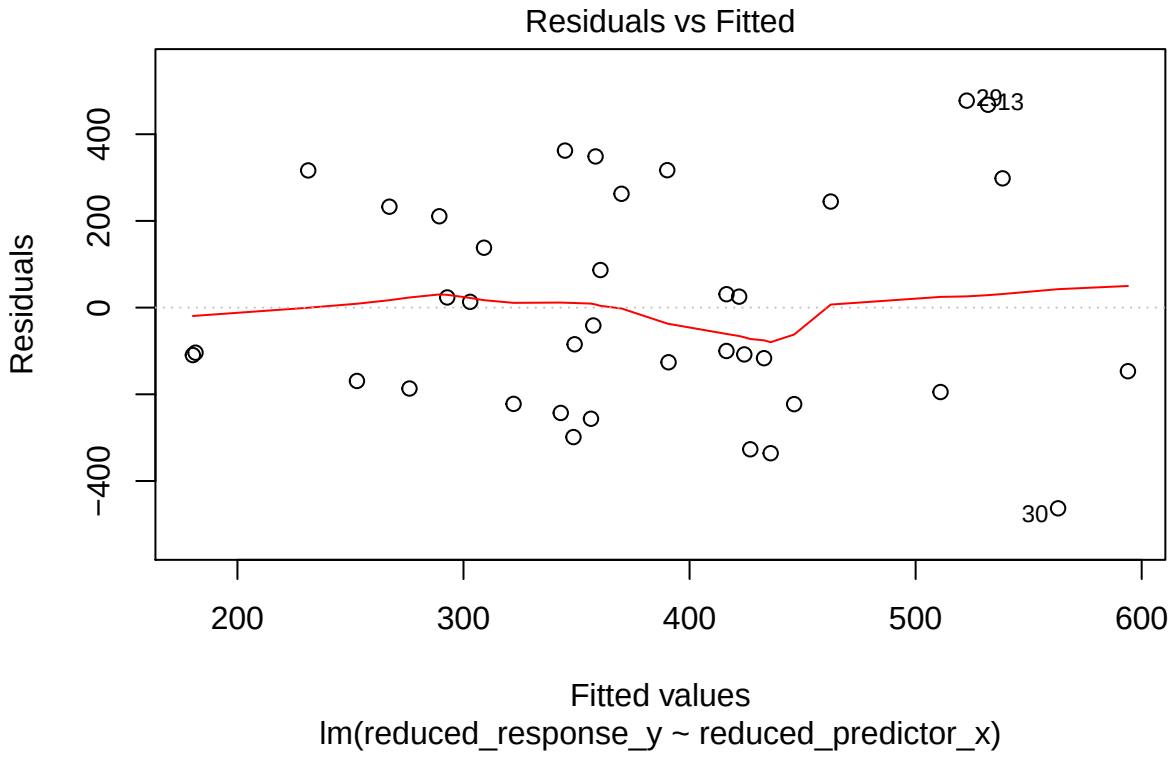
It seems that point 1 has unsually large leverage and point 13 has unusually large standardized residual value, and both of their cook's distance are relatively large.

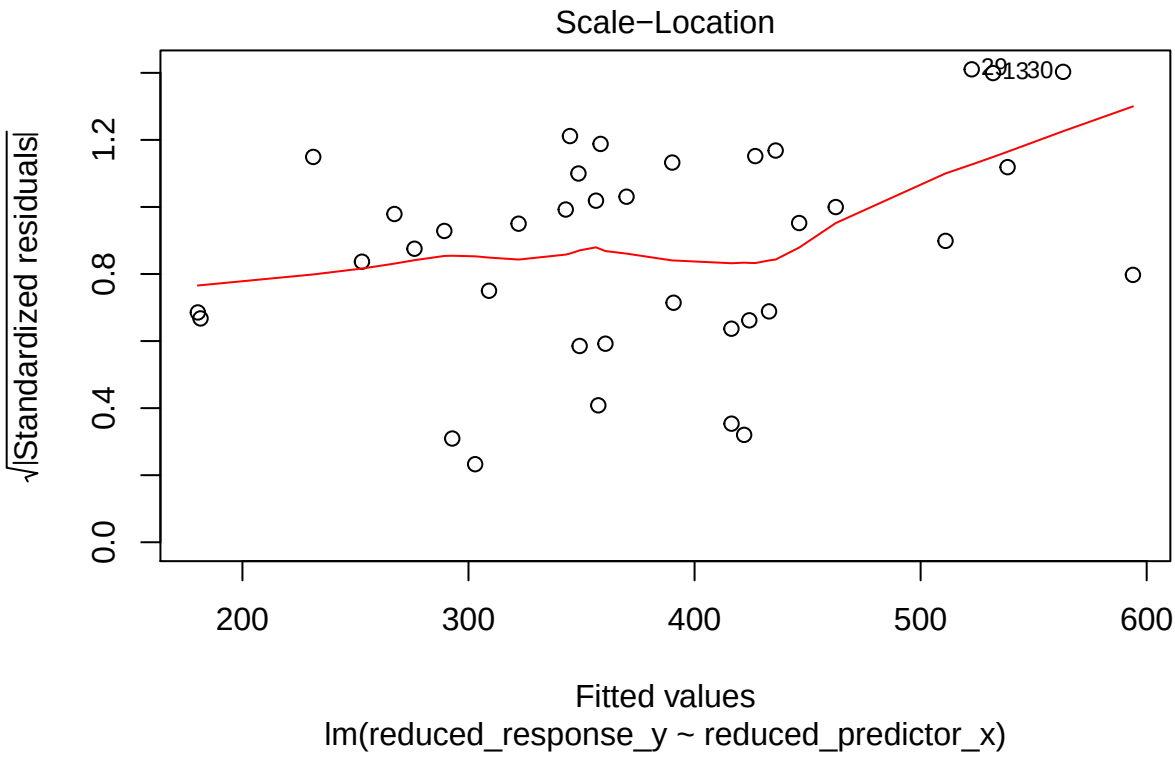
H: Remove them and refit the model

```
invalid_pts <- c(1,13)
reduced_predictor_x <- transformed_predictor_x_2[-invalid_pts]
reduced_response_y <- transformed_response_y_2[-invalid_pts]
reduced_fitted_lm_reg <- lm(formula = reduced_response_y~reduced_predictor_x)
```

C: Assess the adequacy of the model in particular: Is the functional form of the model correct? Do the errors have constant variance?

```
plot(reduced_fitted_lm_reg, c(1,3))
```





Conclusions from plots of reduced data:
1. It seems that the residuals are randomly spreaded; 2. It seems that the increasing square root of standardized residuals of the second part is caused by leverage points, namely point 13, 29, 30. 3. It seems that the assumption of homoscedasticity is satisfied;

E: Do outliers and/or leverage points exist?

```
influence_info <- influence.measures(reduced_fitted_lm_reg)
# DFBETAS0 > 1
influence_info$infmat[,6][influence_info$infmat[,1] > 1]
```

```
## named numeric(0)
```

```
# DFBETAS1 > 1
influence_info$infmat[,6][influence_info$infmat[,2] > 1]
```

```
## named numeric(0)
```

```
# DFFITS > 1
influence_info$infmat[,6][influence_info$infmat[,3] > 1]
```

```
## named numeric(0)
```

```r
# Cook's Distance > 1
influence_info$infmat[,6][influence_info$infmat[,5] > 1]
```

```
## named numeric(0)
```

```r
# Leverage points
influence_info$infmat[,6][influence_info$infmat[,6] > 4/37]
```

```
##         1        15        24        30
## 0.1524698 0.1327822 0.1340513 0.1189982
```

```r
# Outliers
rstandard(reduced_fitted_lm_reg)[rstandard(reduced_fitted_lm_reg) > 2]
```

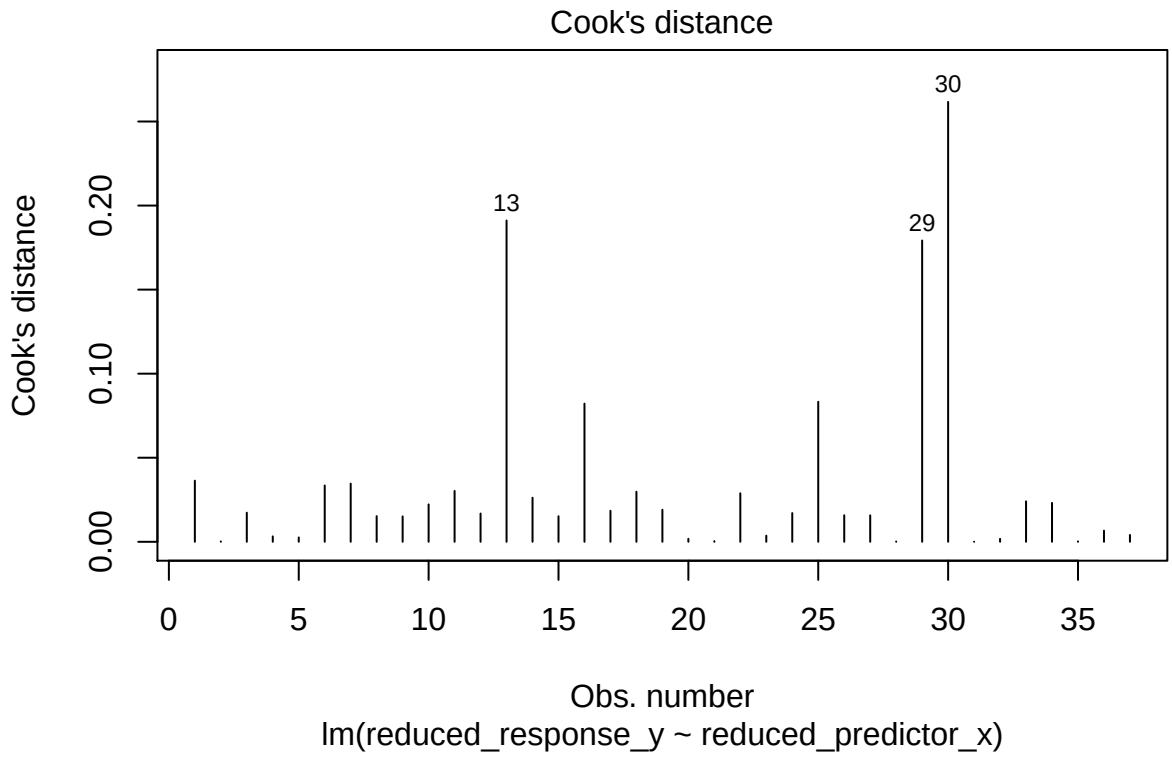
```
## named numeric(0)
```

There are only four leverage points and no outliers.

F: Are the outliers and leverage points valid?

We will be focusing on points with largre cook's distance, and especially on 1, 15, 24, 30.

```r
plot(reduced_fitted_lm_reg, 4)
```



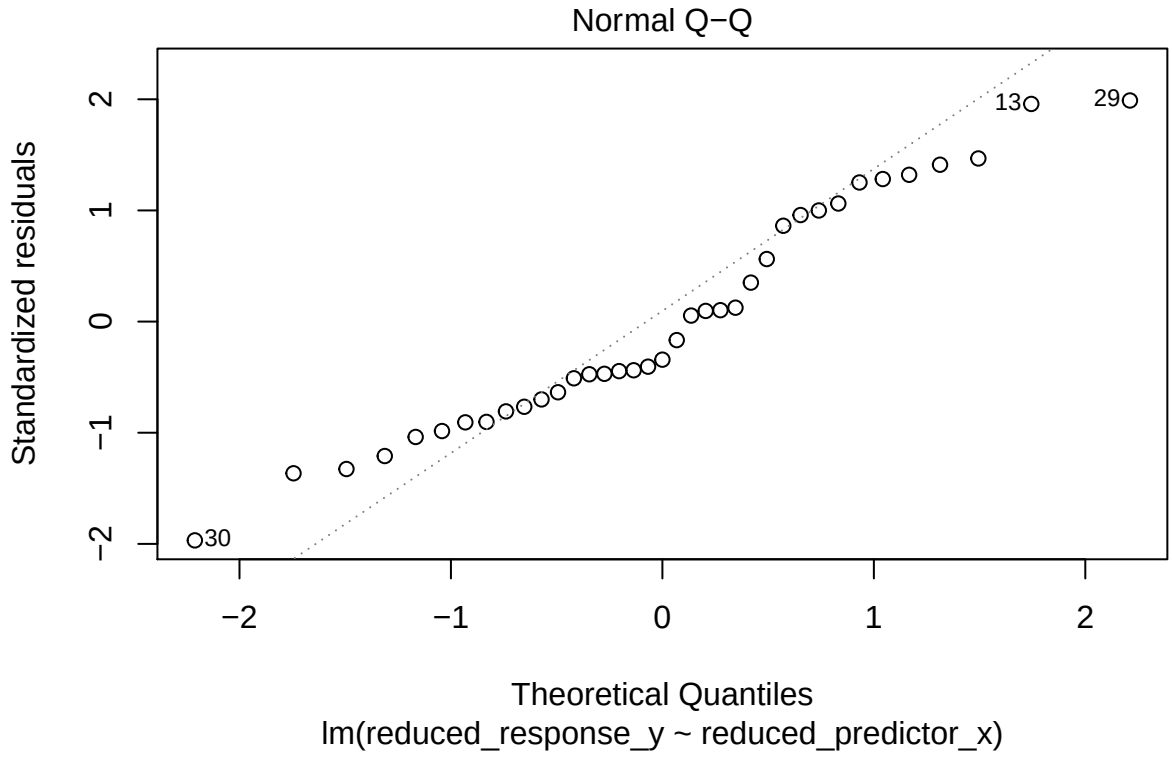
It seems that all points have cook's distance way smaller than 1, therefore we can conclude that the leverage points are "good" leverage points.

J: Is the sample size large ?

There are only 37 observations after the deletion of an outlier and a leverage point, and we can conclude that the sample size is too small for CLT to apply.

K: Are the errors normally distributed?

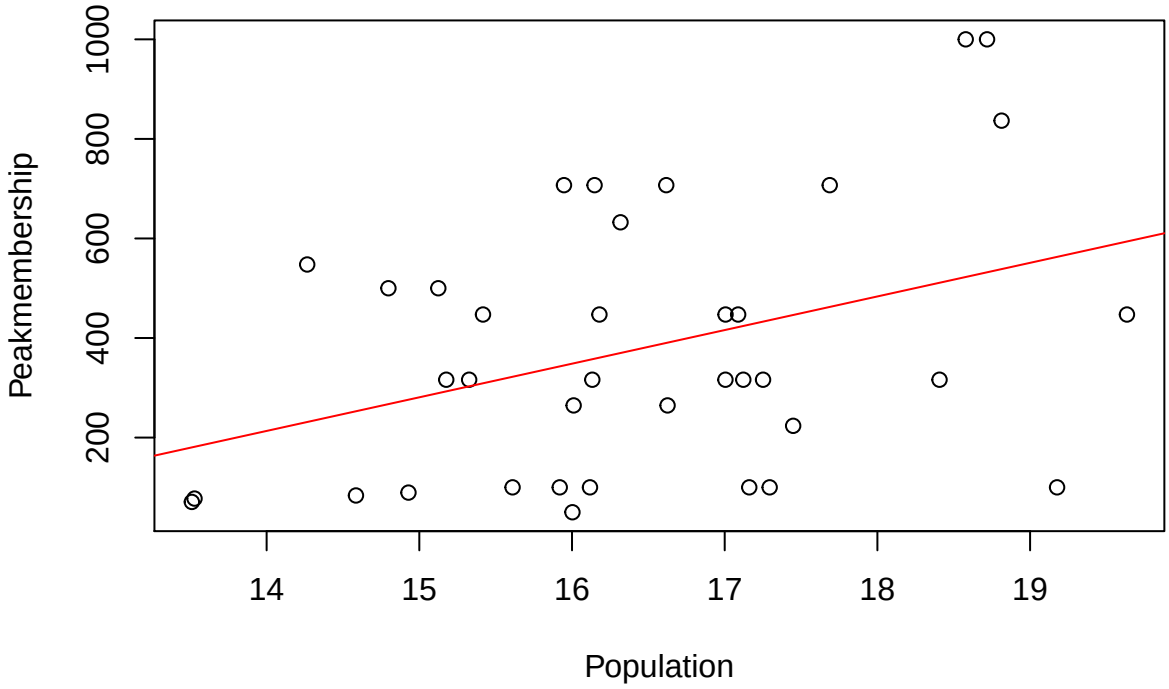
```
plot(reduced_fitted_lm_reg, 2)
```



It seems that most points do not reside on Normal Q-Q plot, and it also looks like the cure has heavy left tail and right tail. Therefore we can conclude that the errors are not normally distributed.

M: Based on Analysis of Variance or a t-test decide if there is a significant association between Y and x?

```
plot(x = reduced_predictor_x, y = reduced_response_y, xlab = "Population", ylab = "Peakmembership")
abline(reduced_fitted_lm_reg, col="red")
```




```
summary(reduced_fitted_lm_reg)
```

```
##
## Call:
## lm(formula = reduced_response_y ~ reduced_predictor_x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -463.01 -186.65  -84.59  232.78  477.40
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -732.31     460.53  -1.590   0.1208
## reduced_predictor_x    67.55      27.88   2.423   0.0207 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 250.6 on 35 degrees of freedom
## Multiple R-squared:  0.1436, Adjusted R-squared:  0.1191
## F-statistic: 5.869 on 1 and 35 DF,  p-value: 0.02072
```

The p-value for β_1 is 0.0207 and is smaller than 0.05, therefore we can conclude that the there is statistically significant evident agains the null hypothesis that $H_0 : \beta_1 = 0$. We can conclude that there is a significant association between transformed Y and transformed x.

N: Use the final model to form confidence intervals

```r
# Confidence Interval for y_hat
conf.intervals <- function(x) {

  n <- length(reduced_response_y)

  # Extract fitted coefficients from model object
  b0 <- reduced_fitted_lm_reg$coefficients[1]
  b1 <- reduced_fitted_lm_reg$coefficients[2]

  # Find SSE and MSE
  rss <- sum((reduced_response_y - reduced_fitted_lm_reg$fitted.values)^2)
  mse <- rss / (n - 2)

  t.val <- qt(0.975, n - 2) # Calculate critical t-value

  # fitted y value
  y.fit <- b1 * x + b0

  # Find the standard error of the regression line
  se <- sqrt(mse) * sqrt(1 / n + (x - mean(reduced_predictor_x))^2 / sum((x - mean(reduced_predictor_x)

  # Warnings of mismatched lengths are suppressed
  lower_bd <- y.fit - t.val * se
  upper_bd <- y.fit + t.val * se

  result <- data.frame(fitted_value = y.fit, lower_bound = lower_bd, upper_bound = upper_bd)

  return(result)
}

head(conf.intervals(reduced_predictor_x))
```

```
##   fitted_value lower_bound upper_bound
## 1     593.9238    395.2621    792.5854
## 2     292.8135    182.3646    403.2623
## 3     276.0973    156.0305    396.1641
## 4     424.2063    332.3483    516.0644
## 5     416.3068    326.9817    505.6319
## 6     344.8879    256.5261    433.2497
```

```r
# Confidence Interval for beta0 and beta1
confint(reduced_fitted_lm_reg)
```

```
##                           2.5 %   97.5 %
## (Intercept)         -1667.23498 202.6170
## reduced_predictor_x    10.94309 124.1504
```

Multi-Linear Regression

(a) Assume that we have a dataset for which SLR under the Gauss-Markov conditions is appropriate. We're given sigma2, the variance of the model error term, and x, a vector of n observations of a predictor variable. The R code is to find X and H, the response variable matrix and the hat matrix, and thence the covariance matrix of the n residuals.The answer is saved as an R matrix named V.

```
residual.covariance_matrix <- function(x, sigma2) {
  # Concatenating 1s as the firts column as X
  X <- matrix(nrow = length(x), ncol = 2)
  X[,1] = 1
  X[,2] = x
  # Calculate Hat matrix
  X_prime_X <- t(X) %*% X
  H <- X %*% solve(X_prime_X) %*% t(X)
  # Calculate Variance-Covariance matrix
  V <- sigma2 * (diag(1, nrow = length(H[,1]), ncol = length(H[1,])) - H)
  return(V)
}
```

(b) Use your work from Q2(a) to calculate V for sigma2=2 and x=seq(1,4).

```
sigma2 <- 2
x <- seq(1,4)
V <- residual.covariance_matrix(x, sigma2)
V
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  0.6 -0.8 -0.2  0.4
## [2,] -0.8  1.4 -0.4 -0.2
## [3,] -0.2 -0.4  1.4 -0.8
## [4,]  0.4 -0.2 -0.8  0.6
```