model2

2025-11-21

```
library(MASS)
library(car)
```

```
## Loading required package: carData
```

```
credit_data <- read.csv("credit_card_data.csv")

set.seed(7)
train_index <- sample(1:nrow(credit_data), size = nrow(credit_data) * 0.7)
train_data <- credit_data[train_index, ]
test_data  <- credit_data[-train_index, ]

head(train_data) #923 rows
```

```
##      card reports      age income        share expenditure owner selfemp
## 476   yes       0 21.50000 2.3779 0.0179990800    35.16667    no      no
## 706   yes       0 23.41667 1.8600 0.2670521000   413.59750    no      no
## 218   yes       0 29.16667 2.5000 0.0767256000   159.59500    no      no
## 630   yes       0 24.08333 1.6200 0.0105080200    13.68583    no      no
## 1016   no       0 30.08333 3.1200 0.0003846154     0.00000    no      no
## 835   yes       1 36.33333 7.3500 0.0712285700   436.27500    no      no
##      dependents months majorcards active
## 476           0      7          0      5
## 706           0      3          1      4
## 218           0      7          1     21
## 630           0     12          0      4
## 1016          2     12          0      5
## 835           0     51          0     14
```

```
# head(test_data) #396 rows
```

Step 1 — Fit Your First-Order Model (Individual Work)

```
model1 <- lm(expenditure ~ income + share + dependents + months + active, data = train_data)
summary(model1)
```

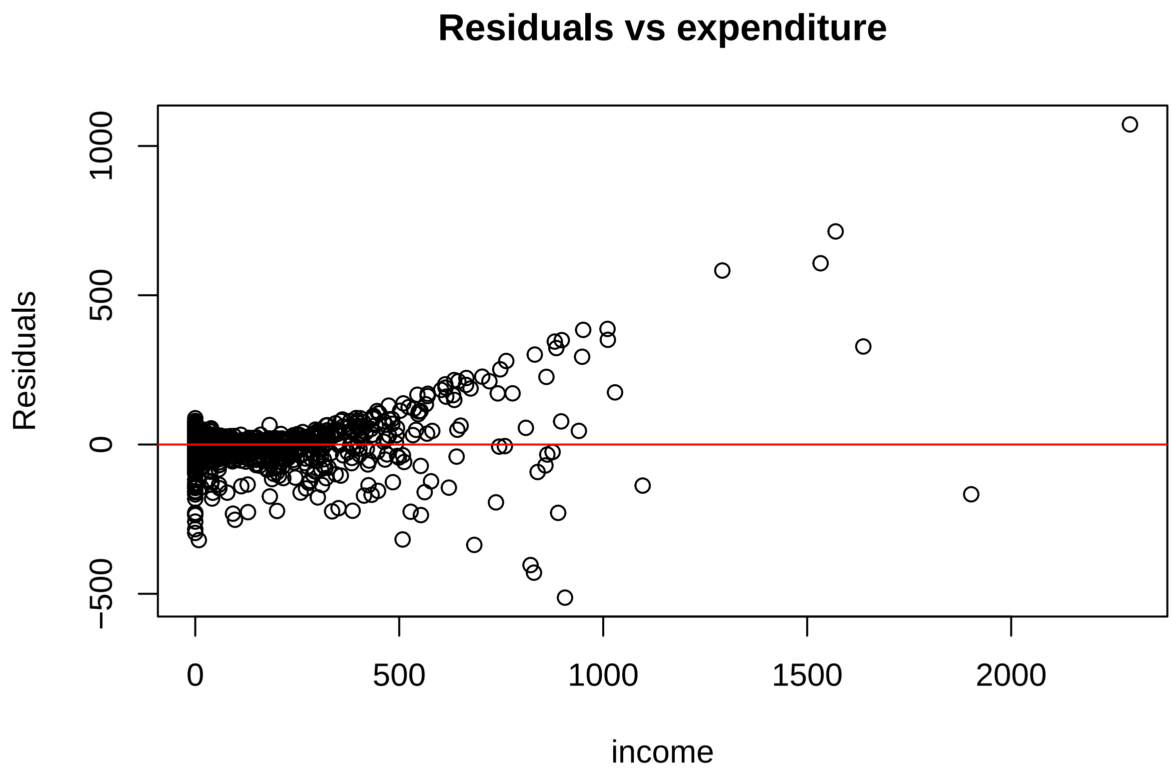
```
##
## Call:
## lm(formula = expenditure ~ income + share + dependents + months +
```

```
##     active, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -512.80  -29.22    3.86   28.43 1072.21
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -129.64194    8.34869 -15.528   <2e-16 ***
## income        41.34601    2.03885  20.279   <2e-16 ***
## share       2375.97225   35.16316  67.570   <2e-16 ***
## dependents     6.37492    2.73331   2.332   0.0199 *
## months        -0.09740    0.04771  -2.041   0.0415 *
## active         0.78291    0.51004   1.535   0.1251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 96.01 on 917 degrees of freedom
## Multiple R-squared:  0.8403, Adjusted R-squared:  0.8395
## F-statistic: 965.4 on 5 and 917 DF,  p-value: < 2.2e-16
```

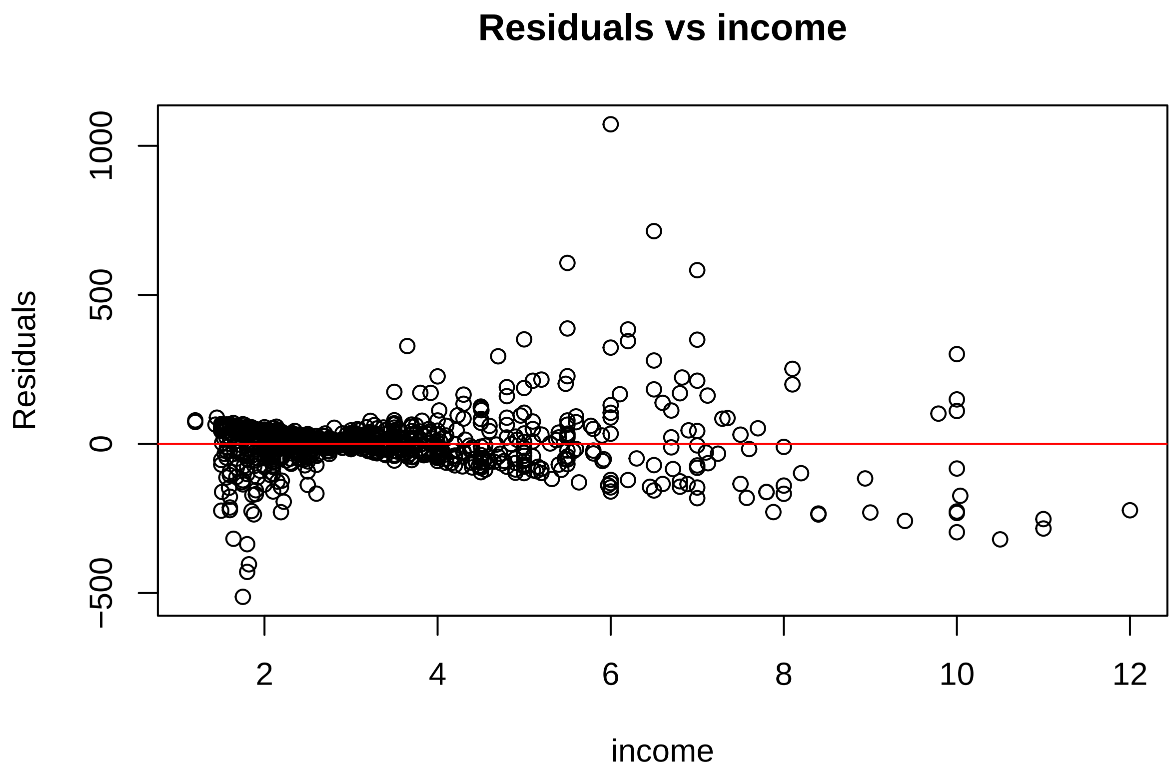
Step 2 — Explore Curvature: Higher-Order Polynomial Terms (Individual Work)

Evaluate whether the predictor exist curvature through residual plots.

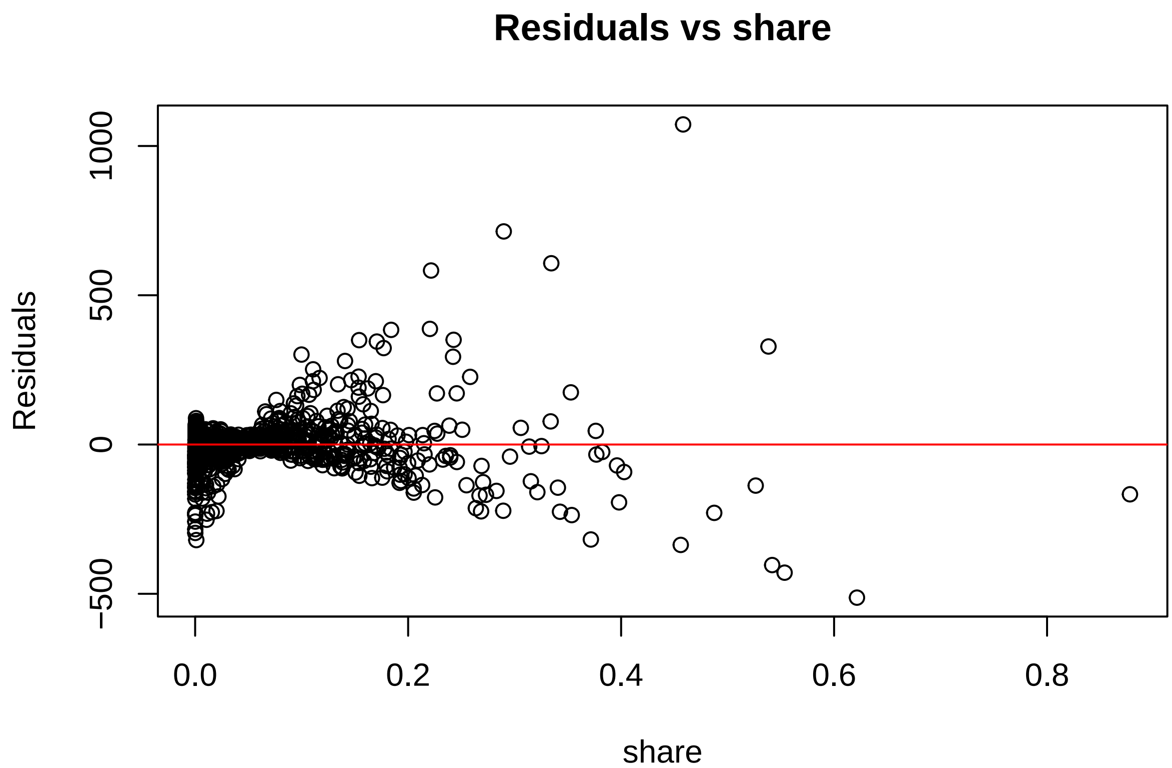
```r
res <- residuals(model1)

# expenditure
plot(train_data$expenditure, res,
     xlab = "income",
     ylab = "Residuals",
     main = "Residuals vs expenditure")
abline(h = 0, col = "red")
```



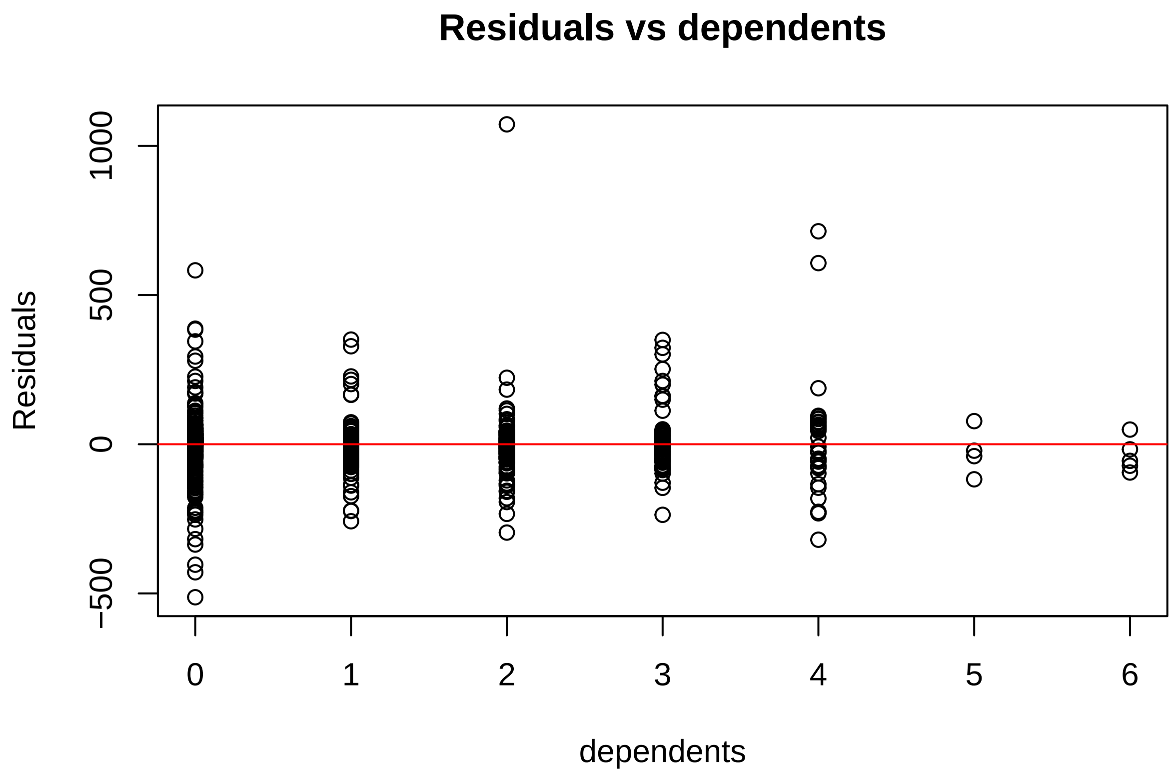

```r
# income
plot(train_data$income, res,
     xlab = "income",
     ylab = "Residuals",
     main = "Residuals vs income")
abline(h = 0, col = "red")
```



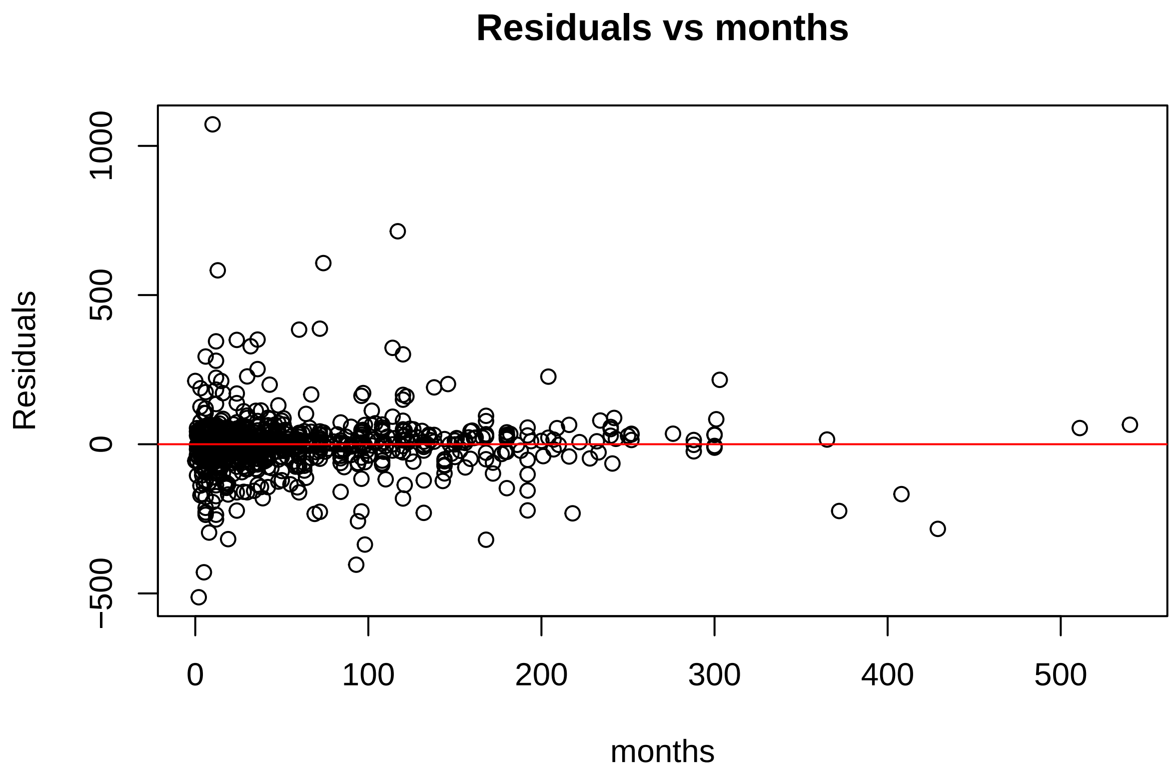

```r
# share
plot(train_data$share, res,
     xlab = "share",
     ylab = "Residuals",
     main = "Residuals vs share")
abline(h = 0, col = "red")
```



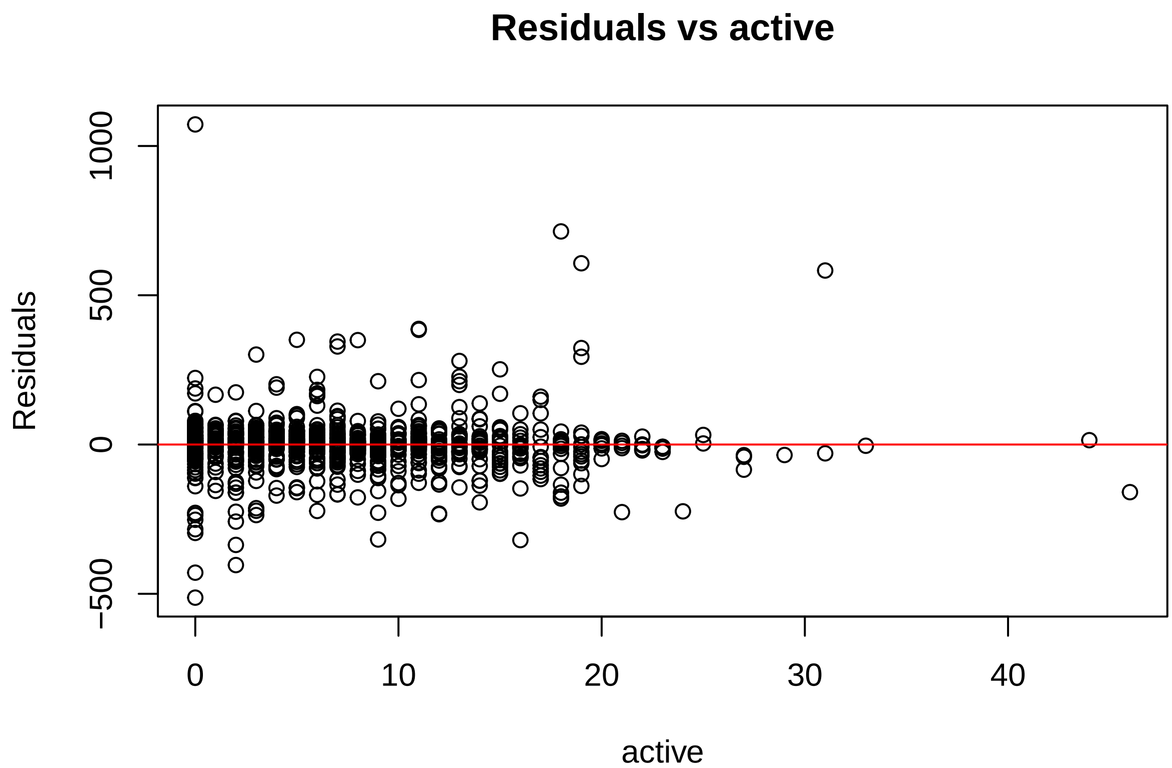

```r
# dependents
plot(train_data$dependents, res,
     xlab = "dependents",
     ylab = "Residuals",
     main = "Residuals vs dependents")
abline(h = 0, col = "red")
```




```r
# months
plot(train_data$months, res,
     xlab = "months",
     ylab = "Residuals",
     main = "Residuals vs months")
abline(h = 0, col = "red")
```

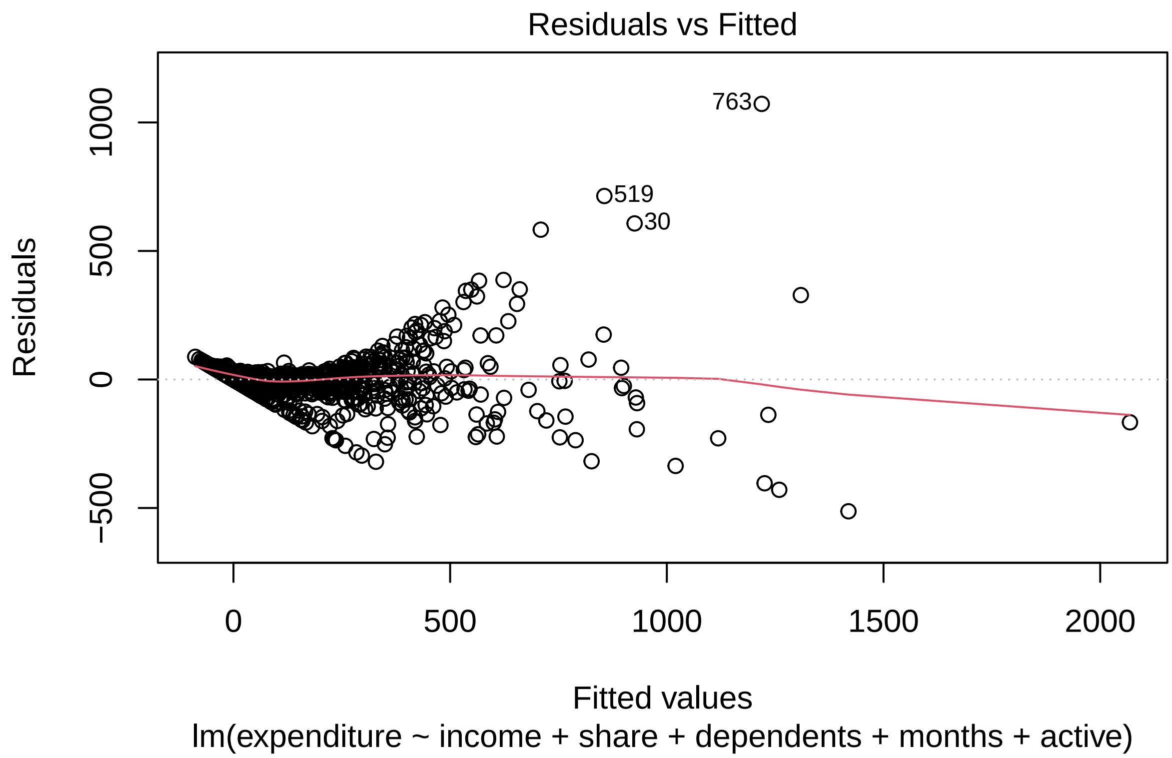



```r
# active
plot(train_data$active, res,
     xlab = "active",
     ylab = "Residuals",
     main = "Residuals vs active")
abline(h = 0, col = "red")
```

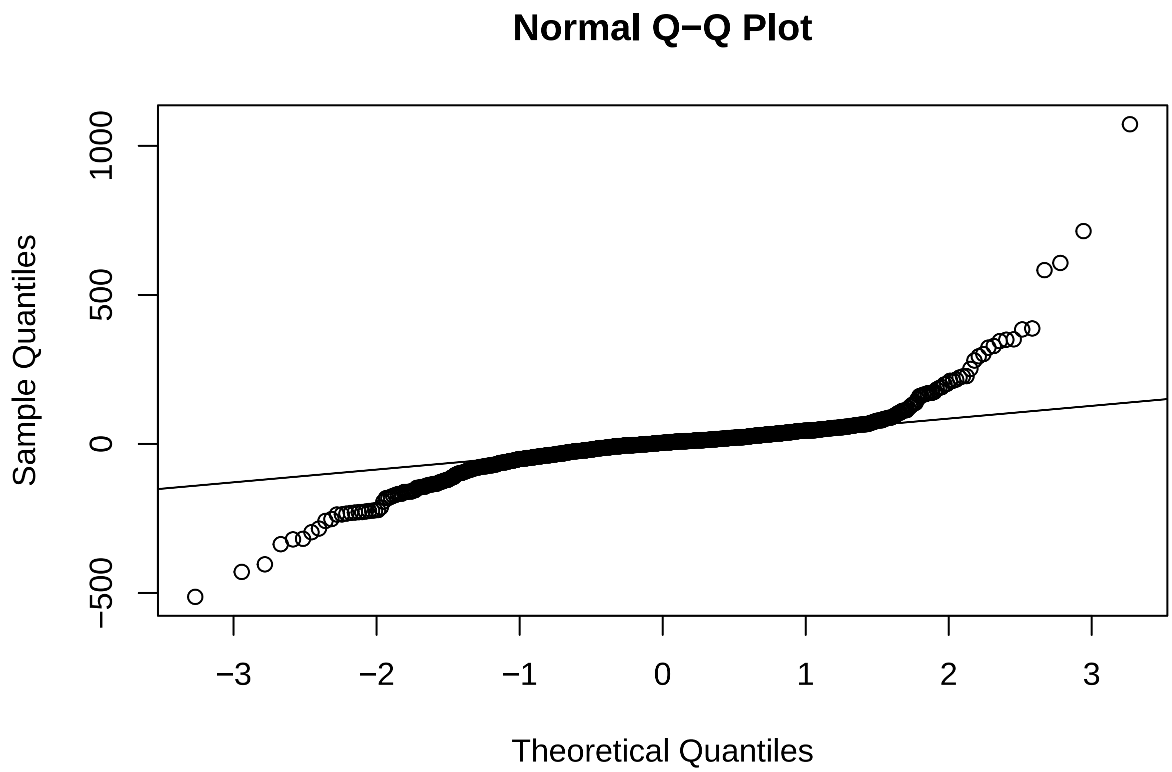


From the performance of residual plots, there are curvature existed in the variable income, share, and month.

```r
# create a reusable function to plot residual vs fitted plot
plot_diagnostics <- function(model, title = NULL) {
  res <- residuals(model)
  fit <- fitted(model)

  # Residuals vs Fitted
  plot(fit, res,
       xlab = "Fitted Values",
       ylab = "Residuals",
       main = ifelse(is.null(title),
                     "Residuals vs Fitted",
                     paste(title, "- Residuals vs Fitted")))
  abline(h = 0, col = "red", lwd = 2)

  # Q-Q Plot
  qqnorm(res,
         main = ifelse(is.null(title),
                       "Normal Q-Q Plot",
                       paste(title, "- Q-Q Plot")))
  qqline(res, col = "red", lwd = 2)
}

plot(model1, which=1) # residuals vs fitted
```




```
qqnorm(resid(model1)); qqline(resid(model1)) # normality
```




```r
# the residual scatter plot shows a cone shape with high variance. This means that the residuals expone
```

```r
# Polynomial terms
train_data$months_c    <- train_data$months - mean(train_data$months)
train_data$months_c2   <- train_data$months_c^2

train_data$income2 <- train_data$income^2
train_data$share2  <- train_data$share^2
train_data$months2 <- train_data$months^2
```

```r
# income + share
model2 <- lm(expenditure ~ income + income2 + share  + share2  + months + dependents + active, data = t
summary(model2)
```

```
##
## Call:
## lm(formula = expenditure ~ income + income2 + share + share2 +
##     months + dependents + active, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -351.46  -36.67   -4.28   30.25 1091.68
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -228.8530    14.5803 -15.696  < 2e-16 ***
## income        88.1860     6.8354  12.901  < 2e-16 ***
## income2       -4.7251     0.6604  -7.155 1.71e-12 ***
## share       2740.3686    64.5730  42.438  < 2e-16 ***
## share2      -921.9591   140.9000  -6.543 1.00e-10 ***
## months        -0.1040     0.0454  -2.291   0.0222 *
## dependents     4.1192     2.6248   1.569   0.1169
## active         0.4368     0.4869   0.897   0.3699
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 91.35 on 915 degrees of freedom
## Multiple R-squared:  0.8558, Adjusted R-squared:  0.8547
## F-statistic: 775.6 on 7 and 915 DF,  p-value: < 2.2e-16
```

```r
# adj r^2: 0.8547

# all quadratic terms
model3 <- lm(expenditure ~ income + income2 + share  + share2  + months + months2 + dependents + active
summary(model3)
```

```
##
## Call:
## lm(formula = expenditure ~ income + income2 + share + share2 +
##     months + months2 + dependents + active, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -350.81  -36.44   -4.13   30.53 1092.57
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.295e+02  1.464e+01 -15.681  < 2e-16 ***
## income       8.794e+01  6.852e+00  12.833  < 2e-16 ***
## income2     -4.698e+00  6.625e-01  -7.091 2.66e-12 ***
## share        2.740e+03  6.460e+01  42.421  < 2e-16 ***
## share2      -9.211e+02  1.410e+02  -6.535 1.06e-10 ***
## months      -5.487e-02  9.932e-02  -0.552    0.581
## months2     -1.757e-04  3.158e-04  -0.556    0.578
## dependents   3.955e+00  2.642e+00   1.497    0.135
## active       4.201e-01  4.880e-01   0.861    0.390
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 91.39 on 914 degrees of freedom
## Multiple R-squared:  0.8558, Adjusted R-squared:  0.8546
## F-statistic: 678.2 on 8 and 914 DF,  p-value: < 2.2e-16
```

```r
# adj r^2: 0.8546

# transformation
# model4 <- lm(expenditure ~ log_income + log_share + dependents + months_c + months_c2 + active, data =
# summary(model4)
```

```r
# # adj r^2: 0.8523
#
# # all sqrt (including polynomial months on sqrt scale)
# model5 <- lm(expenditure ~ log_income + log_share + dependents + sqrt_months_c + sqrt_months_c2 + act
# summary(model5)
# # adj r^2: 0.8524
```

```r
# compare the adjusted R^2 for the models that performed better
cat("Adjusted R^2: \nModel 1: ", summary(model1)$adj.r.squared, "\nModel 2: ", summary(model2)$adj.r.squ
    "\nModel 3: ", summary(model3)$adj.r.squared)
```

```
## Adjusted R^2:
## Model 1:  0.8394792
## Model 2:  0.8546664
## Model 3:  0.8545566
```

```r
# compare the VIF for multicollinearity
cat("\nVIF: \nModel 1: ")
```

```
##
## VIF:
## Model 1:
```

```r
print(vif(model1), type = "predictor")
```

```
##     income      share dependents     months     active
##   1.180984   1.015459   1.148239   1.036684   1.034308
```

```r
cat ("\nModel 2: ")
```

```
##
## Model 2:
```

```r
print(vif(model2), type = "predictor")
```

```
##     income    income2      share     share2     months dependents     active
##  14.660940  14.031714   3.782285   3.770604   1.036922   1.169552   1.041060
```

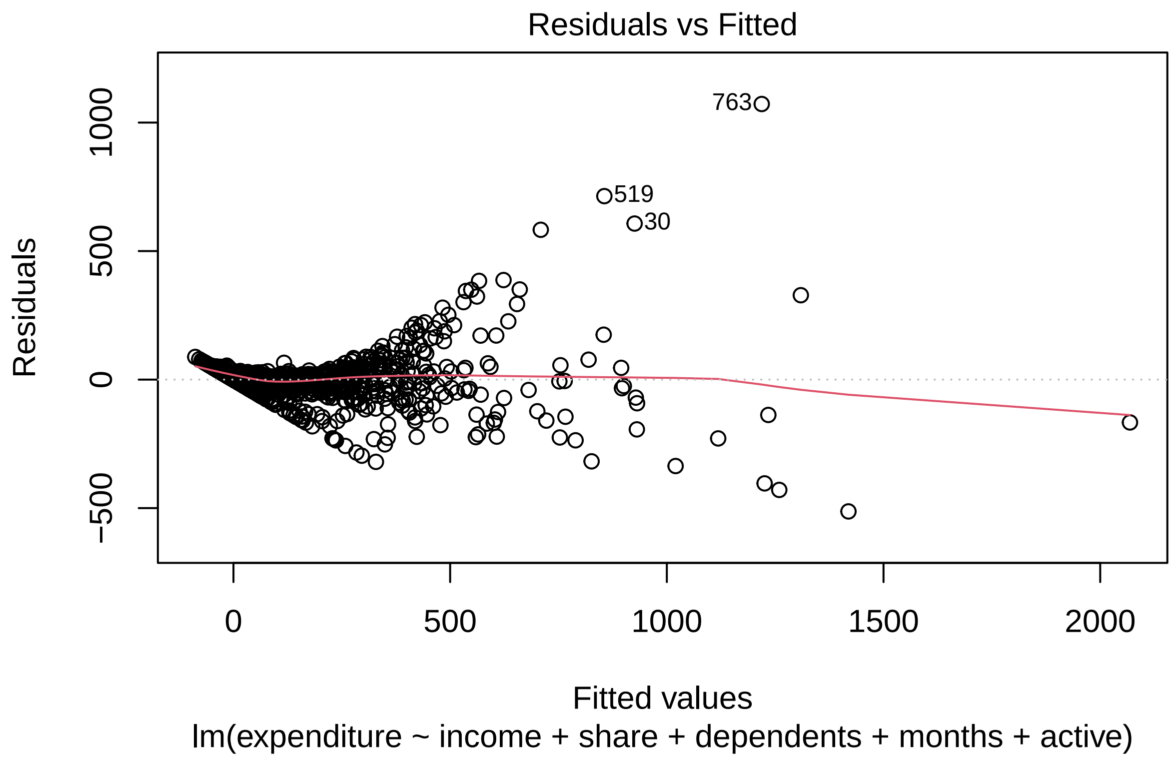
```r
cat("\nModel 3: ")
```

```
##
## Model 3:
```

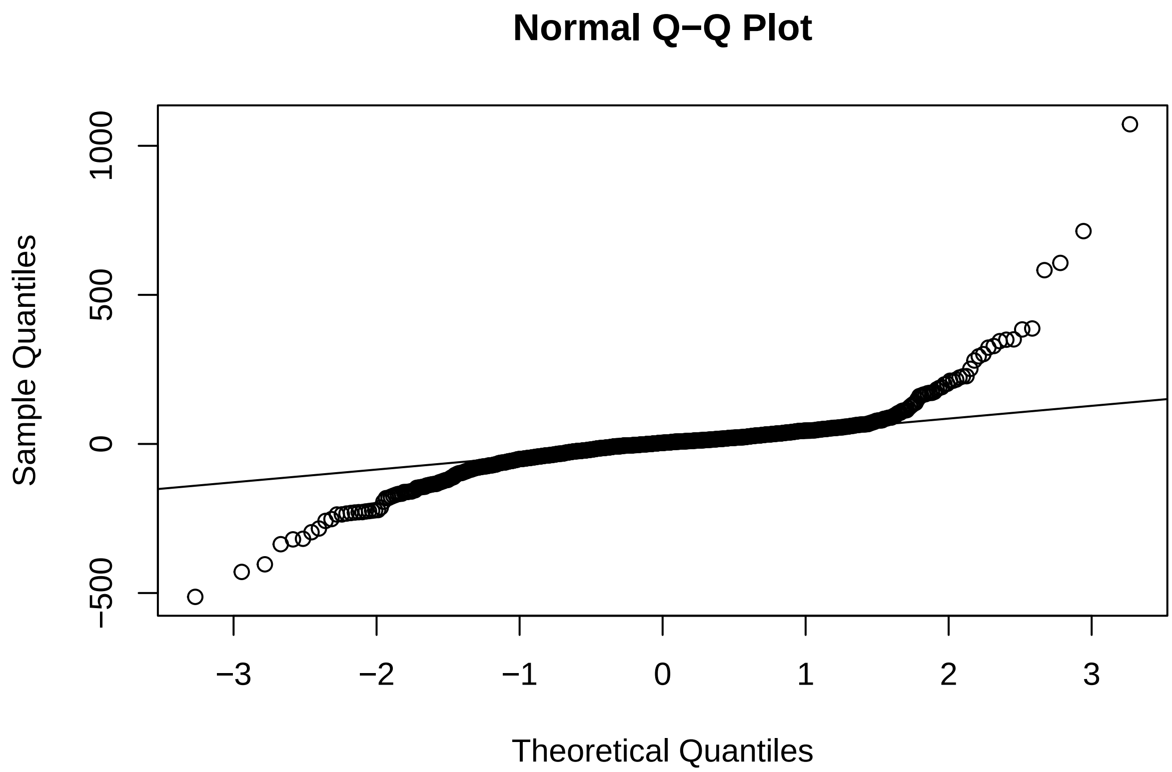
```r
print(vif(model3), type = "predictor")
```

```
##     income    income2      share     share2     months    months2 dependents
##  14.723042  14.109987   3.782295   3.771032   4.959193   4.889934   1.184302
##     active
##   1.045025
```

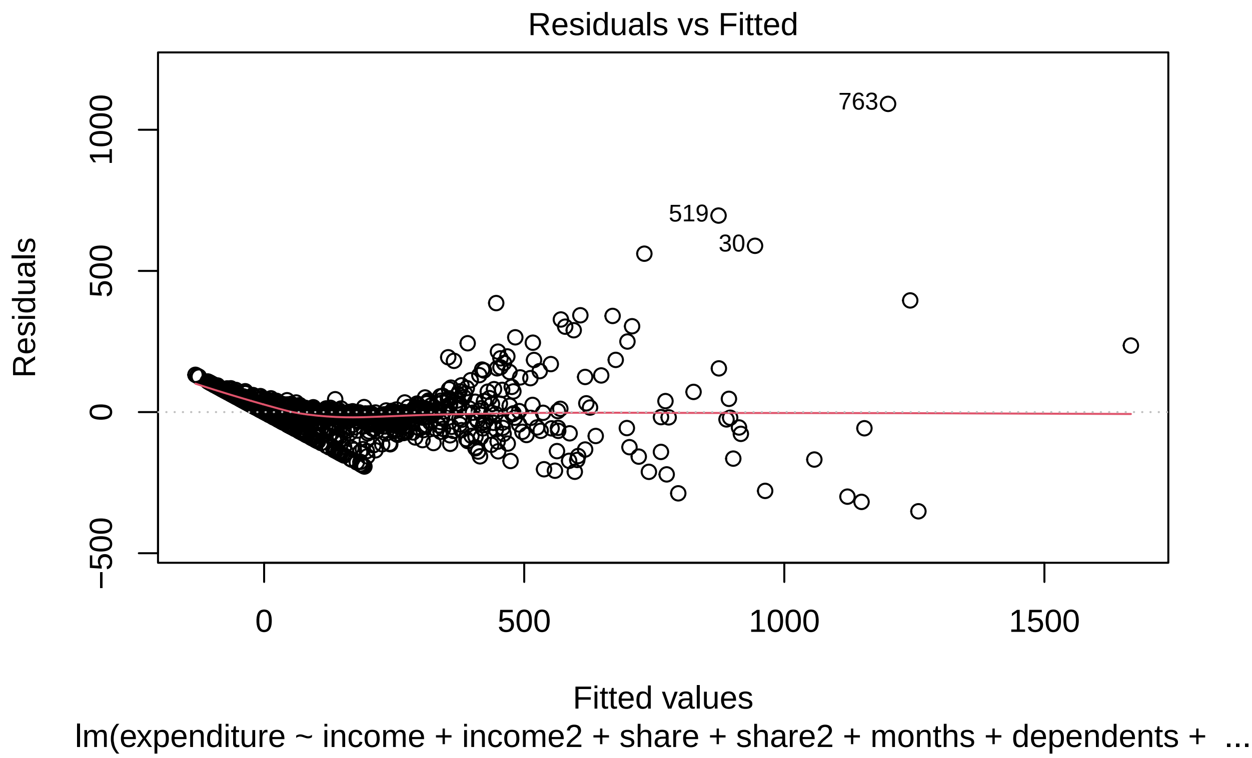
```r
# residual plot
plot(model1, which=1) # residuals vs fitted
```



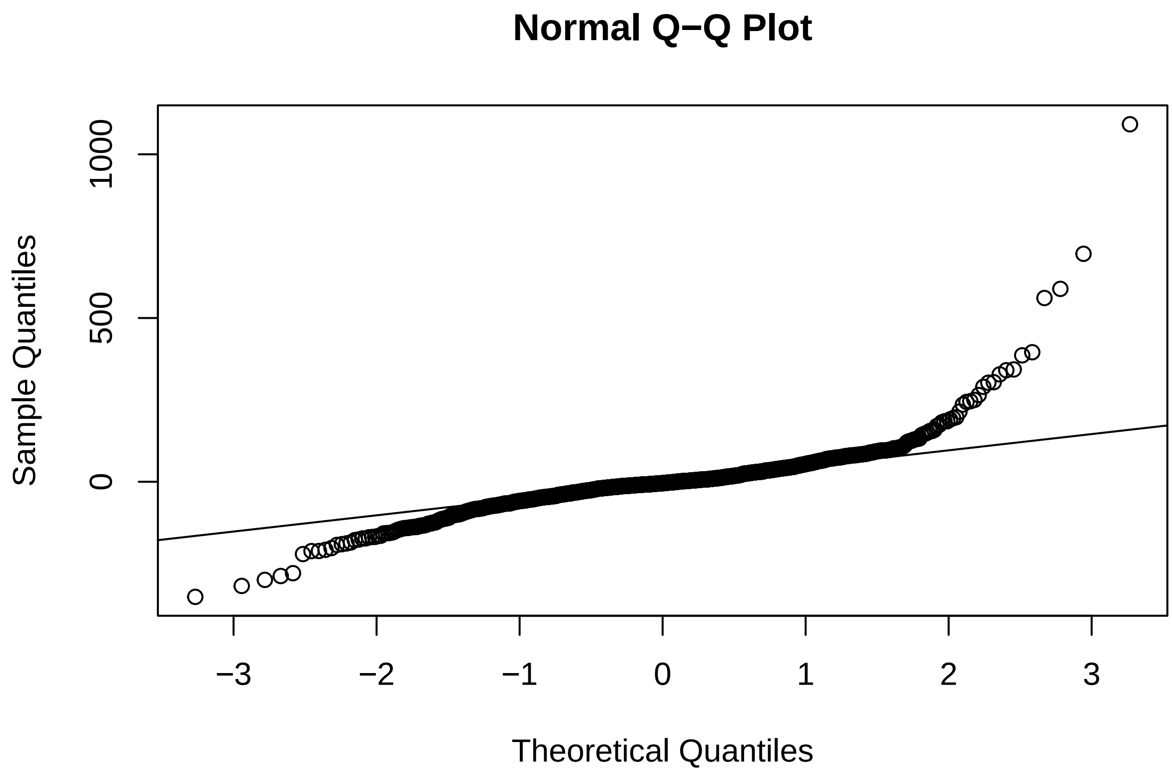

```r
qqnorm(resid(model1)); qqline(resid(model1)) # normality
```



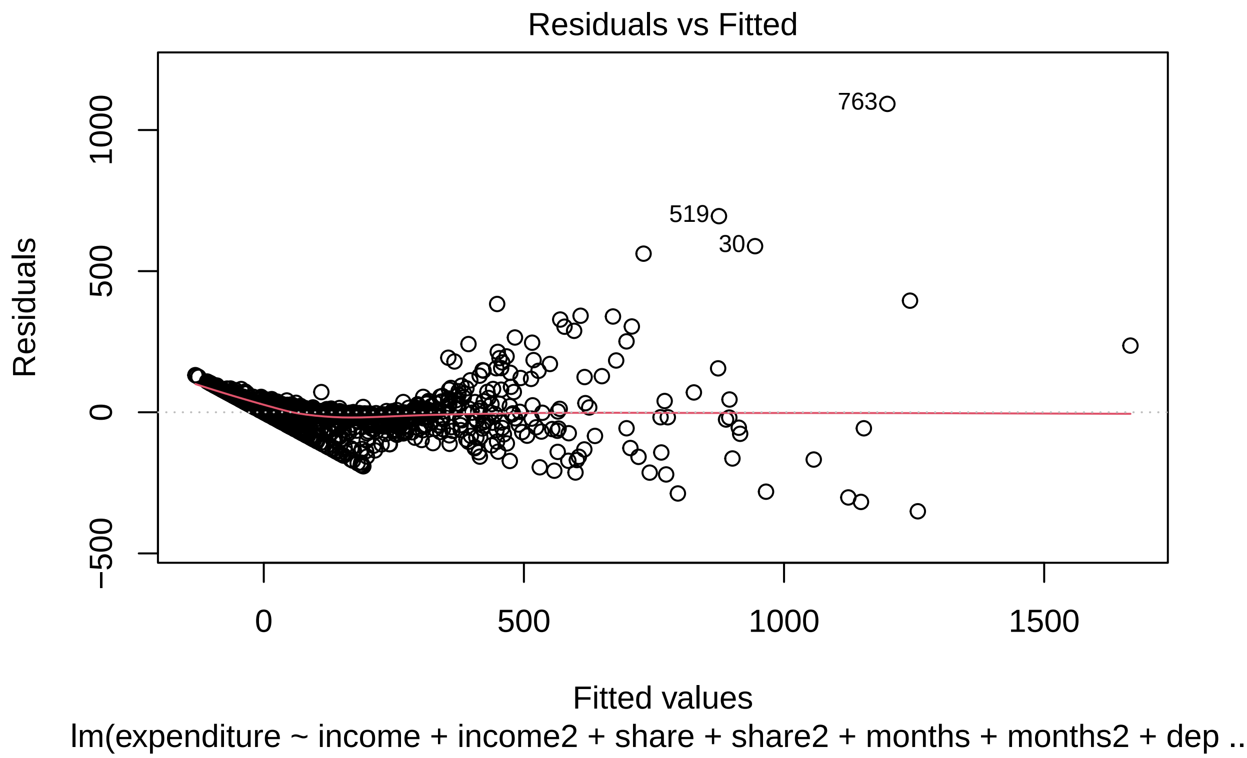

```r
plot(model2, which=1) # residuals vs fitted
```



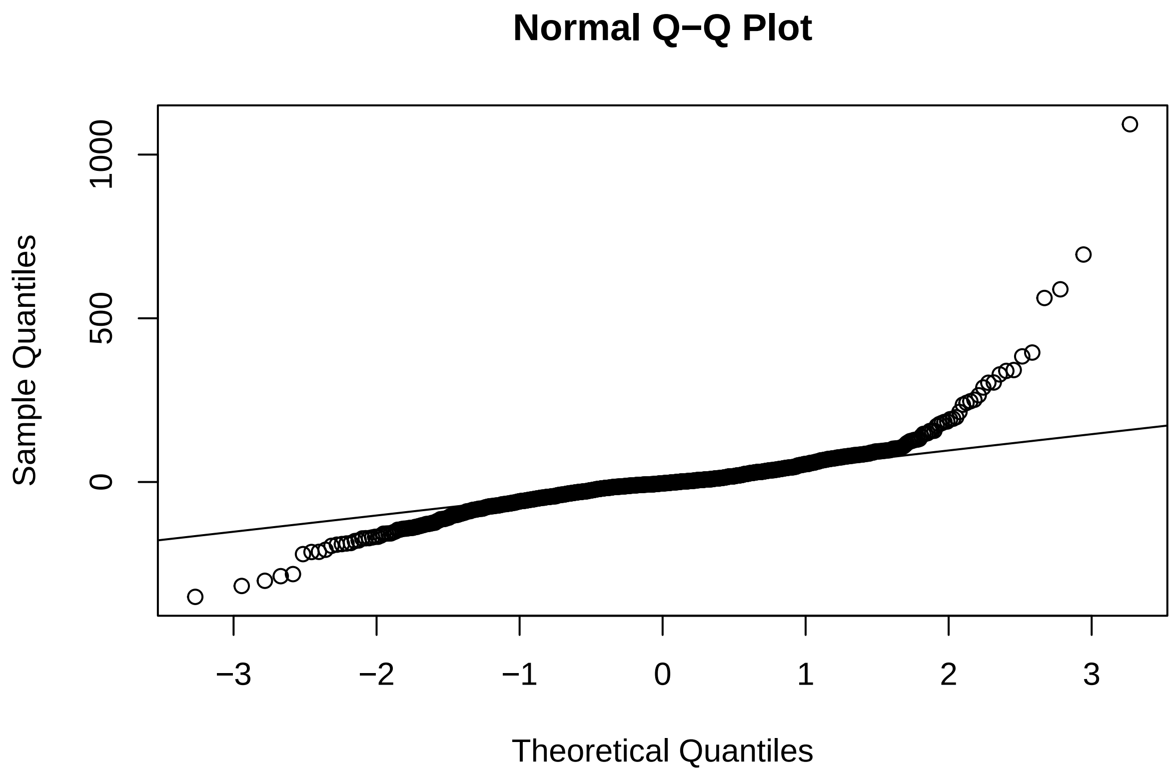

```
qqnorm(resid(model2)); qqline(resid(model2)) # normality
```




```
plot(model3, which=1) # residuals vs fitted
```




```
qqnorm(resid(model3)); qqline(resid(model3)) # normality
```



After checking for curvature in the predictors, we found evidence of nonlinear patterns in income, share, and months, so we explored adding quadratic terms instead of applying transformations at this stage. Among all fitted curvature models, the two best-performing ones were: (1) the model including quadratic terms for income and share, and (2) the model including quadratic terms for income, share, and months. Although the residual-versus-fitted and Q–Q plots did not show substantial visual improvement over the baseline model (only a little bit), these two models achieved the highest adjusted R^2 values among all curvature combinations and surpassed the baseline model (model 1). Therefore, these two curvature-enhanced models are retained for further consideration in the next modeling stage.

Step 3 — Explore Interaction Terms (Individual Work)

Each member is focusing on different predictors, and I was assigned to explore more about the potentials of variable *months*.

```r
#fit model with different combination of interaction with months
model4 <- lm(expenditure ~ income + share + dependents * months + active, data = train_data)
model5 <- lm(expenditure ~ income + share * months + dependents + active, data = train_data)
model6 <- lm(expenditure ~ income * months + share + dependents + active, data = train_data)
model7 <- lm(expenditure ~ income + share + dependents + active * months, data = train_data)

cat("Adjusted R^2: \nModel 4: ", summary(model4)$adj.r.squared,
    "\nModel 5: ", summary(model5)$adj.r.squared,
    "\nModel 6: ", summary(model6)$adj.r.squared,
    "\nModel 7: ", summary(model7)$adj.r.squared)
```

```
## Adjusted R^2:
## Model 4:  0.8394281
## Model 5:  0.8398366
## Model 6:  0.8412528
## Model 7:  0.8393493
```

```r
# Use F test to see whether the interaction term is statistically significant
anova(model1, model4)
```

```
## Analysis of Variance Table
##
## Model 1: expenditure ~ income + share + dependents + months + active
## Model 2: expenditure ~ income + share + dependents * months + active
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1    917 8452561
## 2    916 8446032  1    6529.8 0.7082 0.4003
```

```r
anova(model1, model5)
```

```
## Analysis of Variance Table
##
## Model 1: expenditure ~ income + share + dependents + months + active
## Model 2: expenditure ~ income + share * months + dependents + active
##   Res.Df     RSS Df Sum of Sq      F  Pr(>F)
## 1    917 8452561
## 2    916 8424545  1     28016 3.0462 0.08126 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

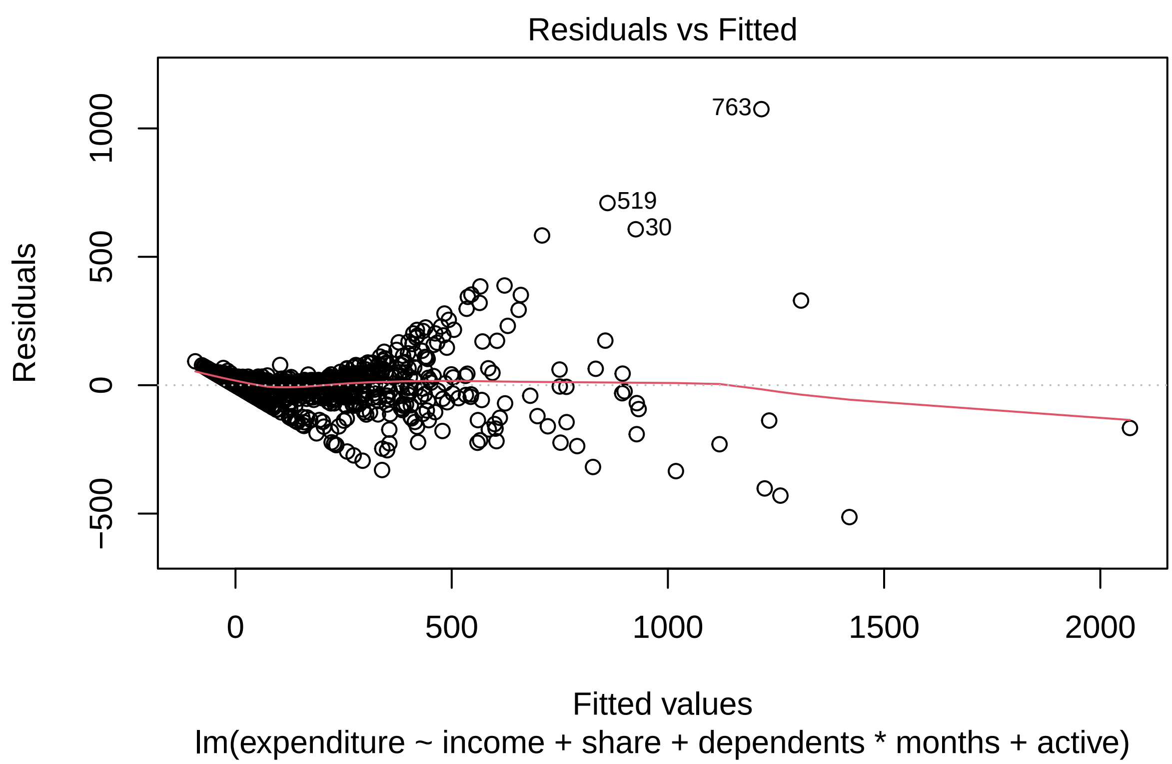
```r
anova(model1, model6)
```

```
## Analysis of Variance Table
##
## Model 1: expenditure ~ income + share + dependents + months + active
## Model 2: expenditure ~ income * months + share + dependents + active
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1    917 8452561
## 2    916 8350052  1    102509 11.245 0.000831 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

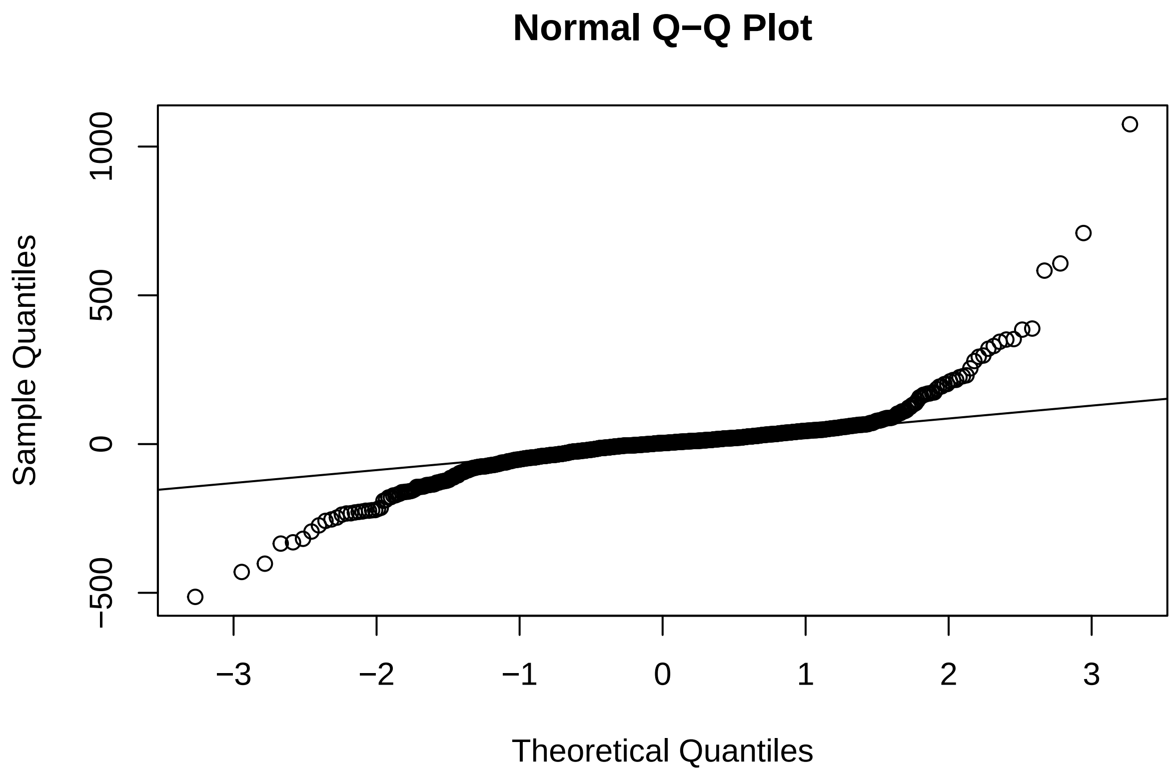
```r
anova(model1, model7)
```

```
## Analysis of Variance Table
##
## Model 1: expenditure ~ income + share + dependents + months + active
## Model 2: expenditure ~ income + share + dependents + active * months
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1    917 8452561
## 2    916 8450173  1    2388.3 0.2589  0.611
```

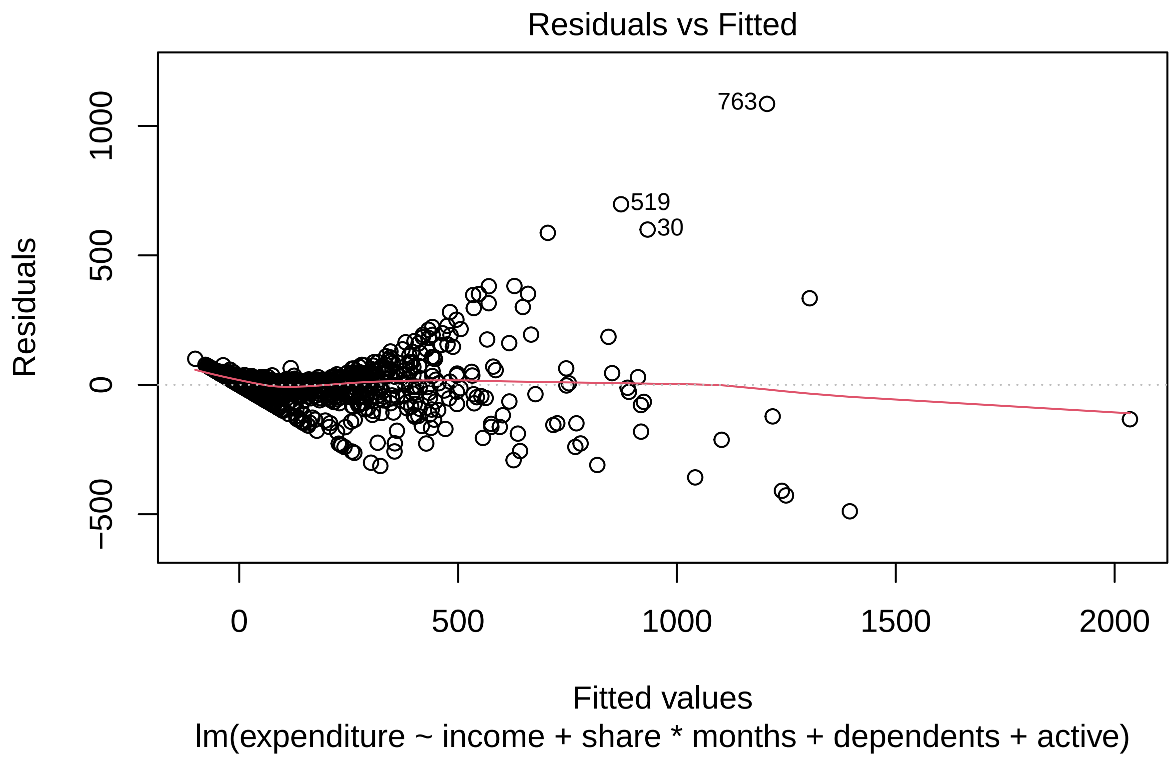
```r
# plot the residual vs fitted model and qq plot to see any improvements
plot(model4, which=1) # residuals vs fitted
```



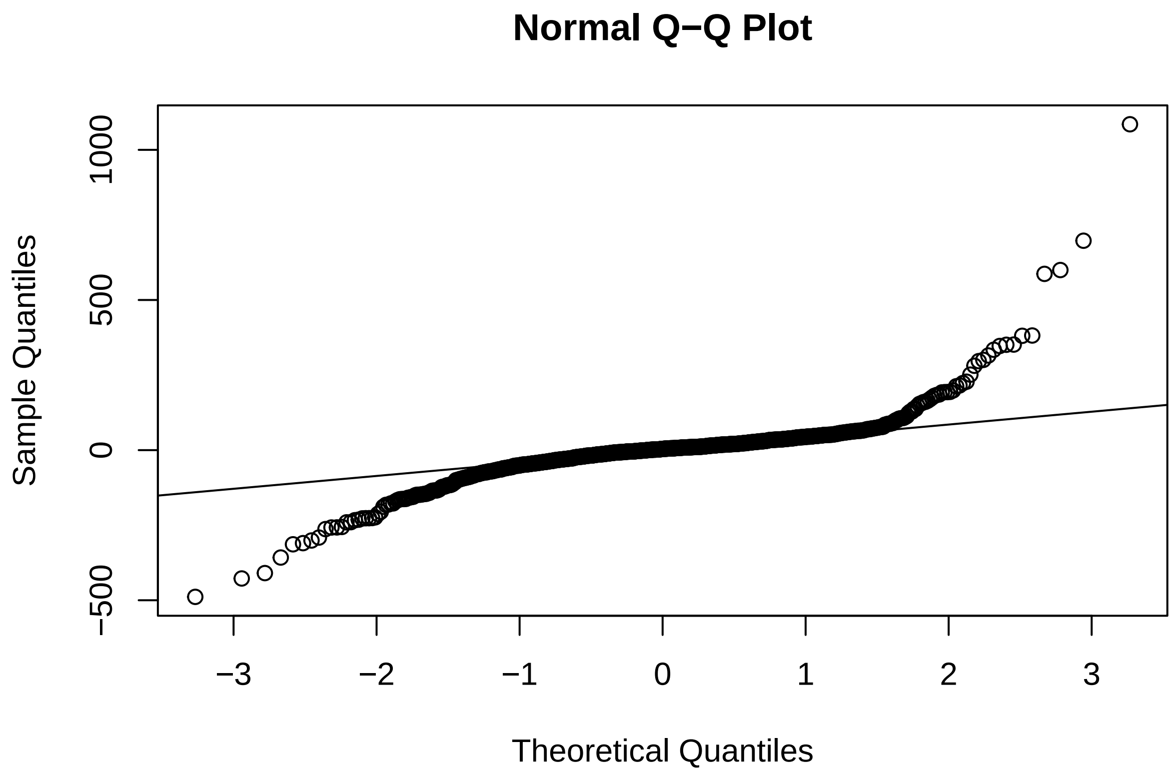

```r
qqnorm(resid(model4)); qqline(resid(model4)) # normality
```



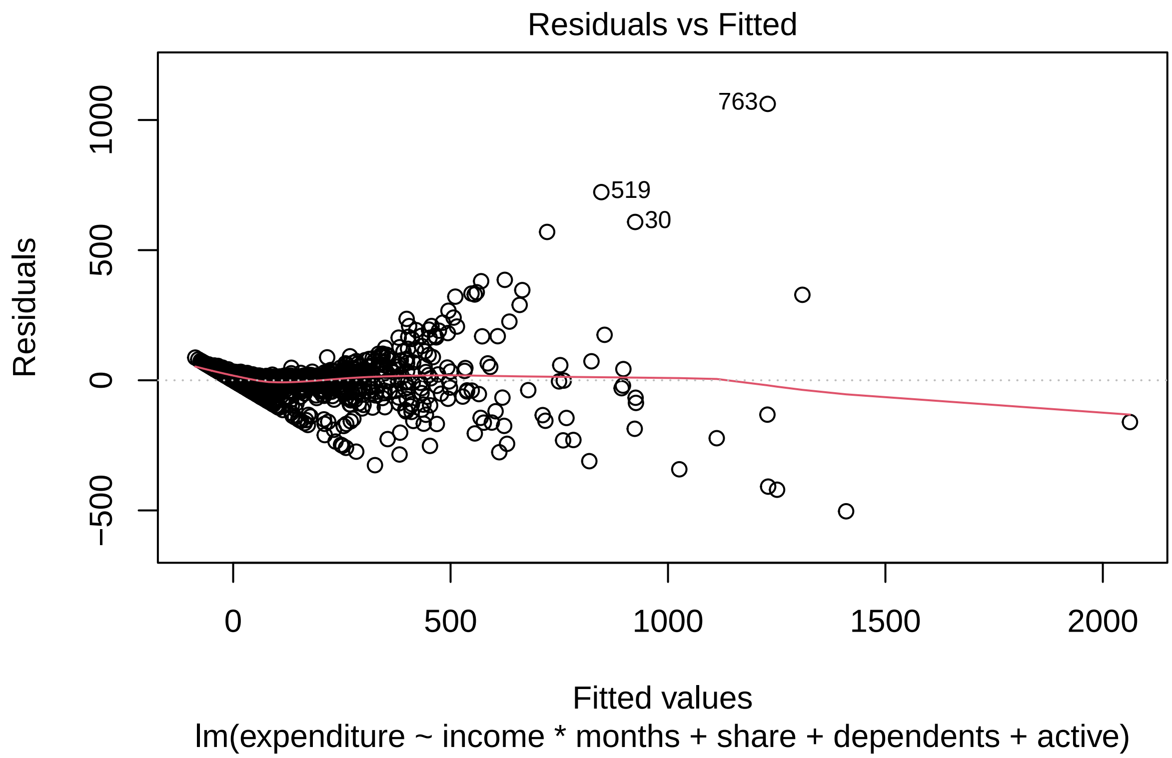

```r
plot(model5, which=1) # residuals vs fitted
```



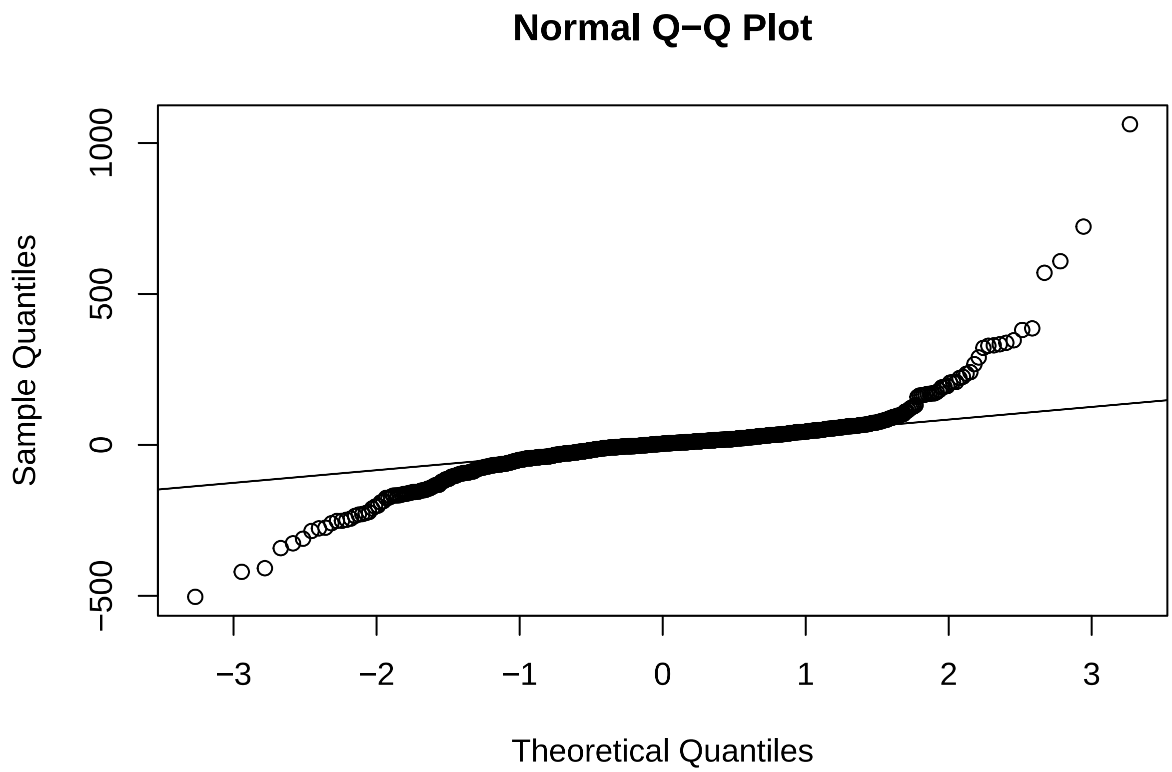

```r
qqnorm(resid(model5)); qqline(resid(model5)) # normality
```



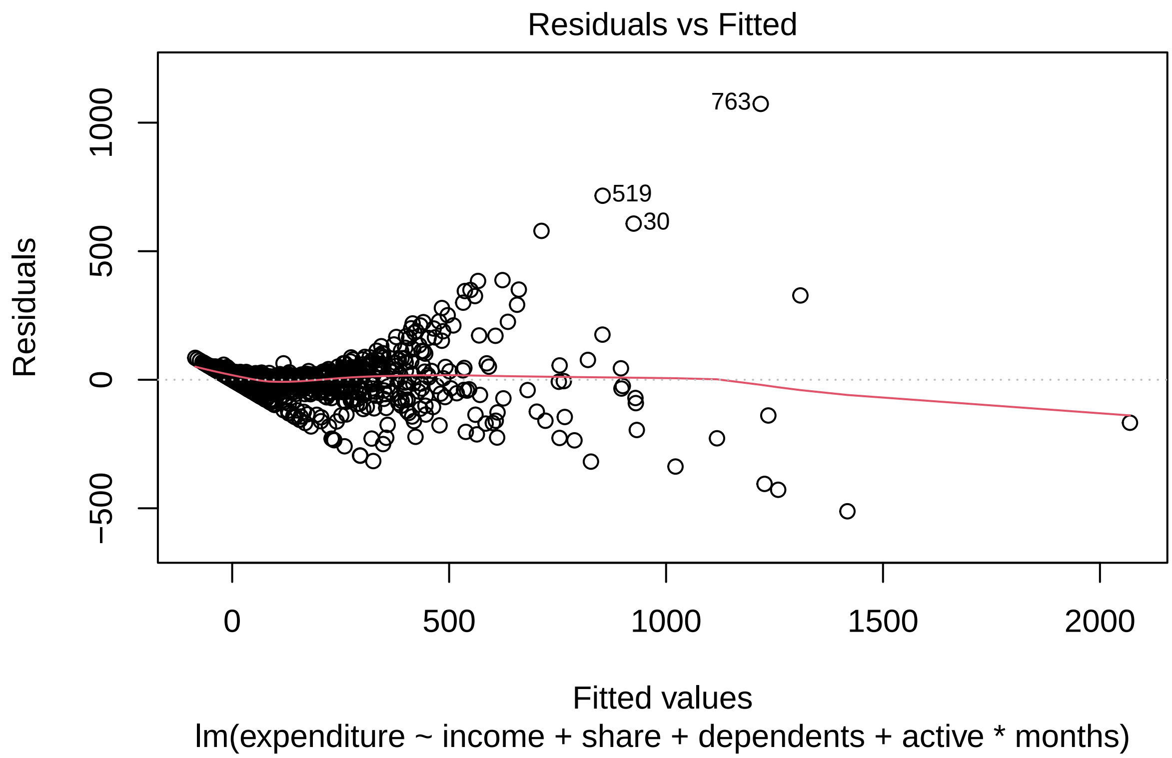

```
plot(model6, which=1) # residuals vs fitted
```



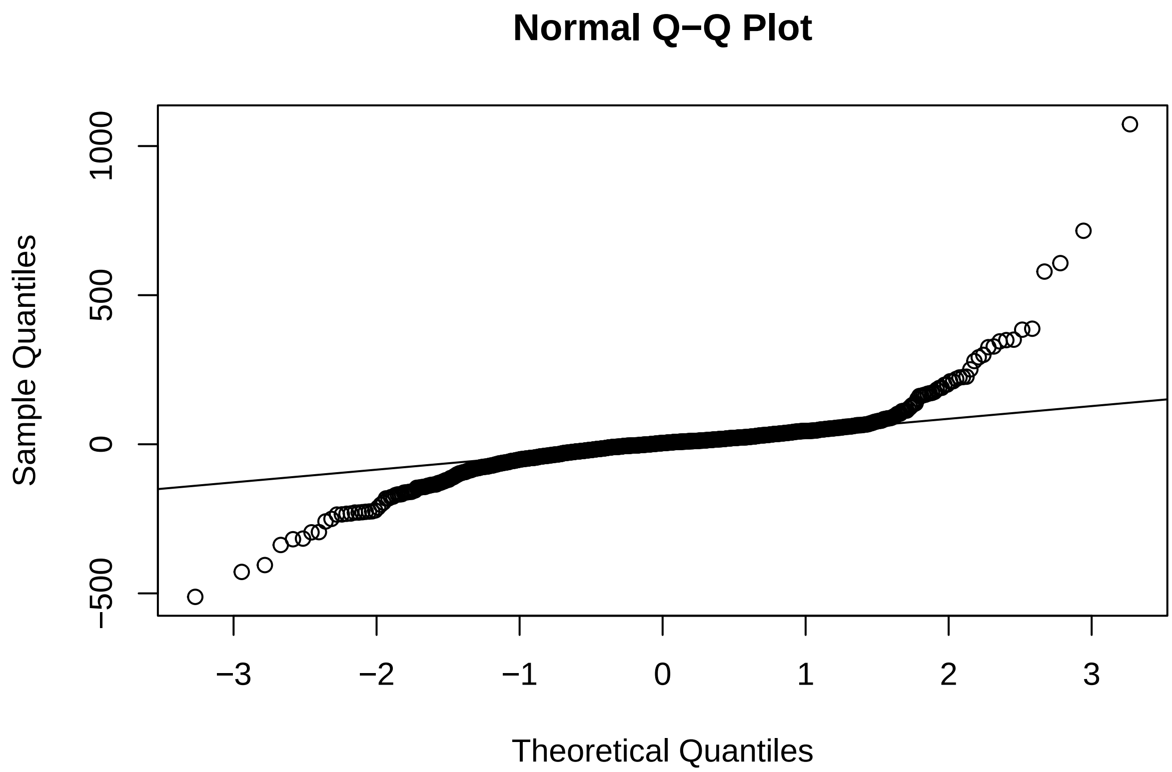

```
qqnorm(resid(model6)); qqline(resid(model6)) # normality
```




```
plot(model7, which=1) # residuals vs fitted
```




```r
qqnorm(resid(model7)); qqline(resid(model7)) # normality
```



Based on the F-tests comparing models 4–7 against model 1, models 4 and 7 do not show significant improvement, while models 5 and 6 are statistically significant. Although the residual diagnostics did not show noticeable changes across these models, model 6 achieves the highest adjusted R^2 among models 4–7 and provides a statistically significant improvement over model 1. Therefore, we retain the interaction term $income * months$ for further consideration. [specify income * months]

Step 4 — Checkpoint: Determine Whether Screening Is Needed

```
n <- nrow(train_data)
n
```

```
## [1] 923
```

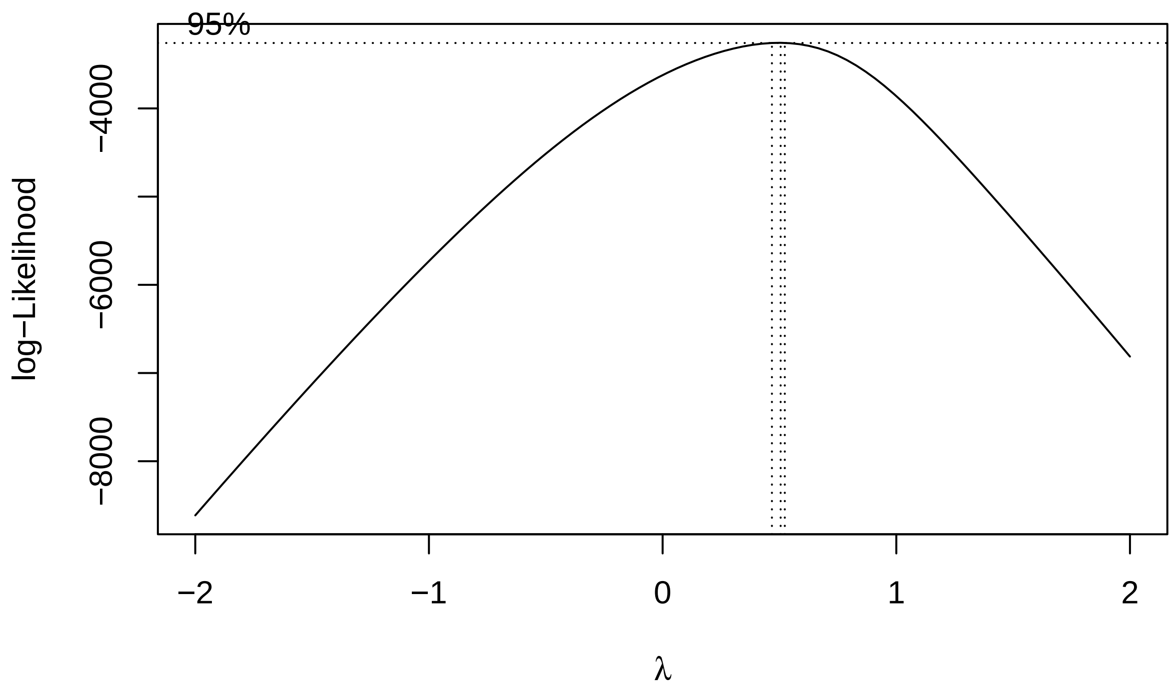
Because the train dataset contains 923 observations, the sample size is more than sufficient to support models that include quadratic and interaction terms. When evaluating multicollinearity using VIF across all expanded models so far, none of the predictors, including quadratic and interaction terms, exceeded a VIF of 5. This indicates that the models are stable, and there is no evidence of harmful multicollinearity. Since (1) the sample size is large, (2) the expanded models do not overload the data, and (3) multicollinearity remains at acceptable levels, variable screening is not required at this stage. You can safely proceed to the next modeling step using the full set of first-order predictors along with the candidate curvature and interaction terms identified earlier.

Step 5 — Evaluate Need for Transformations (Individual Work)

```
# Log transforms
train_data$log_income      <- log(train_data$income)
train_data$log_share       <- log(train_data$share + 1)     # +1 if share has zeros
train_data$log_months      <- log(train_data$months)

# Sqrt transforms
train_data$sqrt_income <- sqrt(train_data$income)
train_data$sqrt_share  <- sqrt(train_data$share)
train_data$sqrt_months <- sqrt(train_data$months)

# Centering log transforms
train_data$log_income_c      <- train_data$log_income - mean(train_data$log_income)
train_data$log_income_c2     <- train_data$log_income_c^2

train_data$log_share_c       <- train_data$log_share - mean(train_data$log_share)
train_data$log_share_c2      <- train_data$log_share_c^2

train_data$log_months_c      <- train_data$log_months - mean(train_data$log_months)
train_data$log_months_c2     <- train_data$log_months_c^2

# Centering sqrt transforms
train_data$sqrt_income_c     <- train_data$sqrt_income - mean(train_data$sqrt_income)
train_data$sqrt_income_c2    <- train_data$sqrt_income_c^2

train_data$sqrt_share_c      <- train_data$sqrt_share - mean(train_data$sqrt_share)
train_data$sqrt_share_c2     <- train_data$sqrt_share_c^2

train_data$sqrt_months_c     <- train_data$sqrt_months - mean(train_data$sqrt_months)
train_data$sqrt_months_c2    <- train_data$sqrt_months_c^2
```

Based on previous analyses from last two weeks, we found that all variables that we are using, including both predictors and the response, are right-skewed. Furthermore, in part 2, we identified four variables, which are expenditure, income, share, and months, that exhibit curvature, violating the assumptions of normality and constant variance. Therefore, it is necessary to perform a transformation. Regarding Y, the expenditure, I will use Box-Cox transformation to assess whether transforming the response variable can improve these assumptions. Regarding X, I will explore potential transformations (such as log or square-root) for the three predictor variables.

```
# box cox requires variables to be all positive, so we use log expenditure squared
bc <- boxcox(lm((expenditure+1) ~ income + share + dependents + months + active, data=train_data))
```

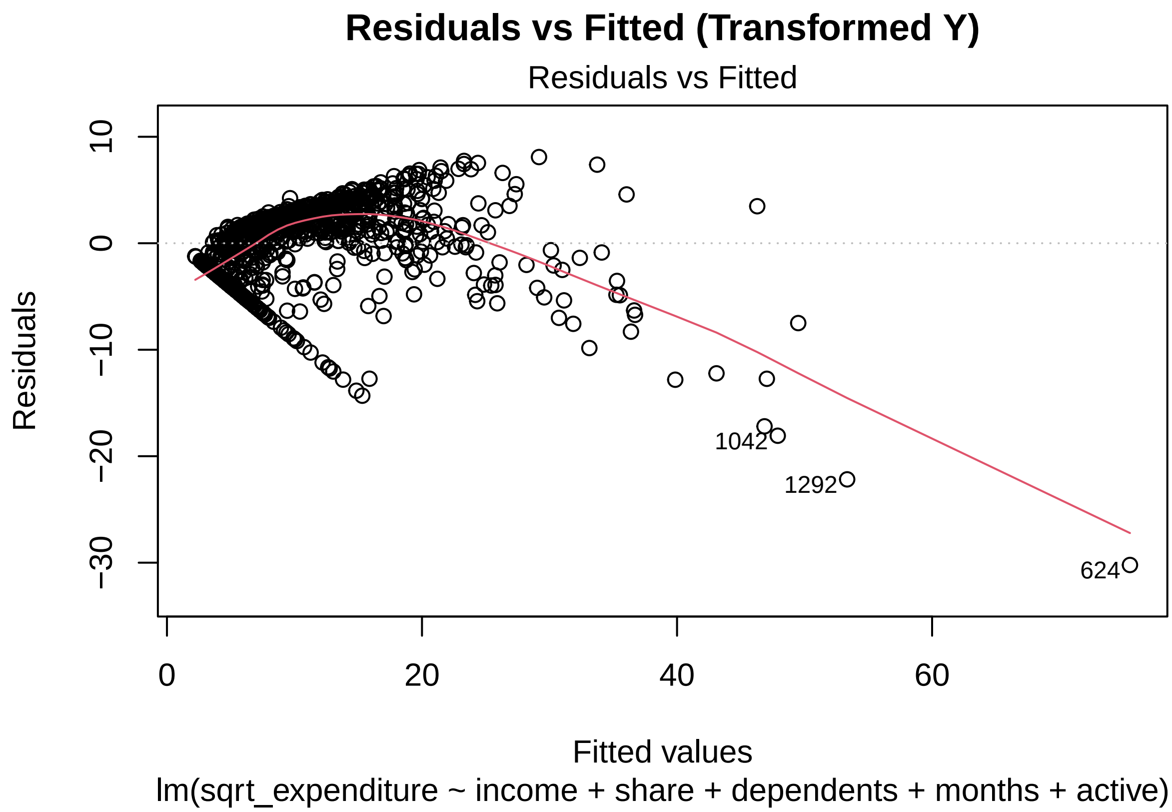



```r
lambda <- bc$x[which.max(bc$y)]
lambda # 0.5050505
```

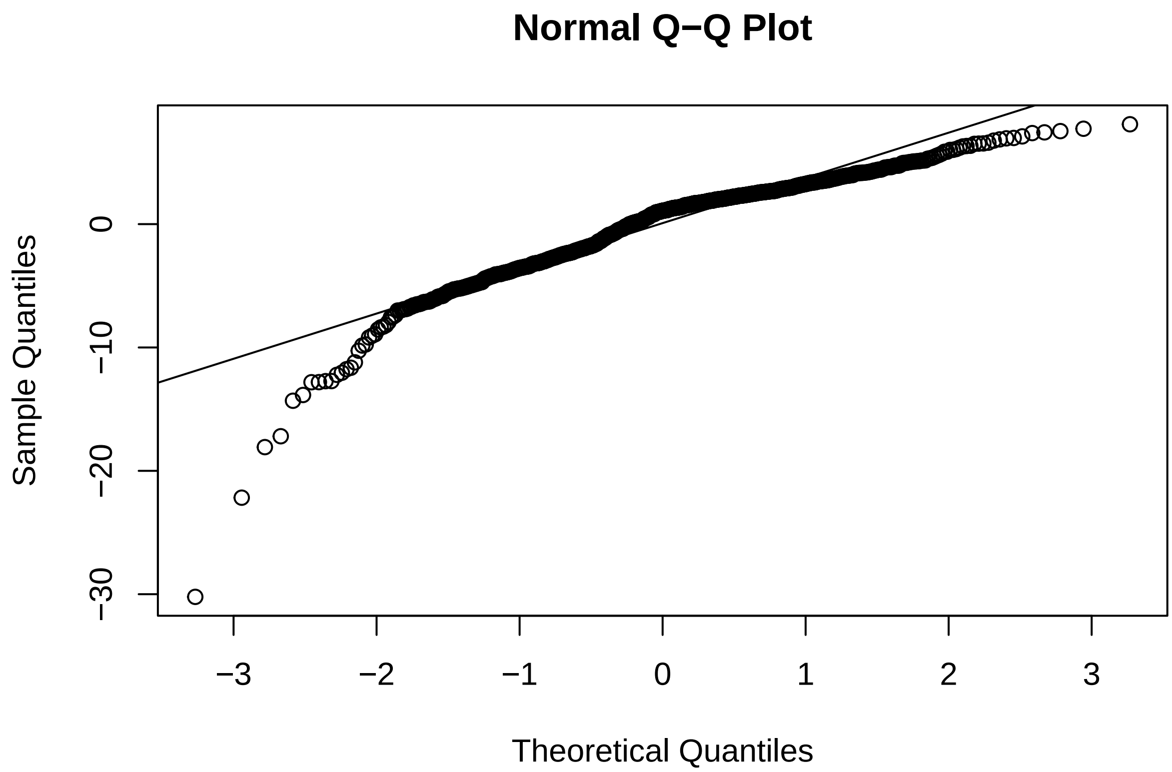
```
## [1] 0.5050505
```

To address the issue of negative values existed in expenditure (Y), I applied Box-Cox to expenditure + 1 and found that the optimal lambda is 0.505. This result indicates that a square-root transformation of the response variable is most appropriate for stabilizing variance and improving the normality of residuals.

```r
train_data$sqrt_expenditure <- (train_data$expenditure + 1)^0.505

model_trans <- lm(sqrt_expenditure ~ income + share + dependents + months + active, data=train_data)

plot(model_trans, which=1, main="Residuals vs Fitted (Transformed Y)")
```




```
qqnorm(resid(model_trans)); qqline(resid(model_trans), main="QQ Plot (Transformed Y)")
```



The plot diagnosises show that transform Y makes the situation worse. there is still a clear curvature exist in the residual vs fitted plot, and even more deviations in the QQ plot.

Step 6 — Build Your Refined Model (Individual Work)

- First-order terms: expenditure ~ income + share + dependents + months + active
- Centered polynomial terms: combination of $income^2 + share^2 + share^2$
- Interaction terms: $incomemonths$
- Transformed predictors: $share$
- transformed response: $sqrt(expenditure)$

```r
# Interaction term, use the centered variable
train_data$income_c      <- train_data$income - mean(train_data$income)
train_data$share_c      <- train_data$share - mean(train_data$share)
train_data$income_months_share <- train_data$income_c * train_data$months * train_data$share_c
# Centered polynomial terms
train_data$income_c2   <- train_data$income_c^2
train_data$log_share_c2 <- train_data$log_share_c^2

refined_model3 <- lm(sqrt_expenditure ~ income_c + income_c2 + sqrt_share_c + log_share_c2 +
                     months + income_months_share + dependents + active,
```

```r
                      data = train_data) # adjusted R^2: 0.971
summary(refined_model3)
```

```
##
## Call:
## lm(formula = sqrt_expenditure ~ income_c + income_c2 + sqrt_share_c +
##     log_share_c2 + months + income_months_share + dependents +
##     active, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4202 -0.5621 -0.0057  0.6437 10.5039
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.120e+01  9.856e-02 113.604  < 2e-16 ***
## income_c             1.672e+00  4.458e-02  37.512  < 2e-16 ***
## income_c2           -7.415e-02  1.080e-02  -6.864 1.23e-11 ***
## sqrt_share_c         5.241e+01  3.531e-01 148.416  < 2e-16 ***
## log_share_c2        -1.254e+01  3.097e+00  -4.050 5.56e-05 ***
## months               2.540e-03  7.475e-04   3.398 0.000709 ***
## income_months_share  7.872e-02  3.966e-03  19.850  < 2e-16 ***
## dependents           3.530e-02  4.136e-02   0.853 0.393634
## active               2.392e-03  7.651e-03   0.313 0.754633
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.435 on 914 degrees of freedom
## Multiple R-squared:  0.9712, Adjusted R-squared:  0.971
## F-statistic:  3853 on 8 and 914 DF,  p-value: < 2.2e-16
```

```r
cat("The final refined model:\nAdjusted R^2: ", summary(refined_model3)$adj.r.square)
```

```
## The final refined model:
## Adjusted R^2:  0.9709521
```

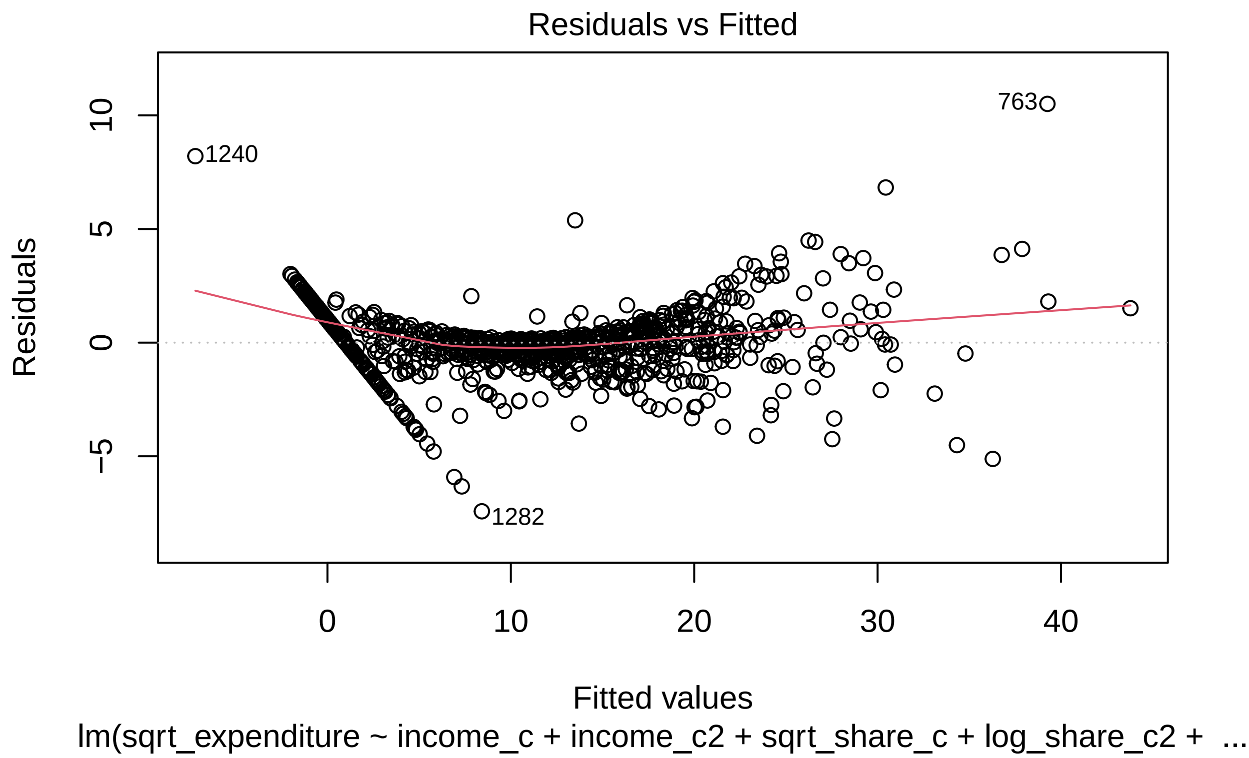
```r
cat("\nVIF: \n")
```

```
##
## VIF:
```

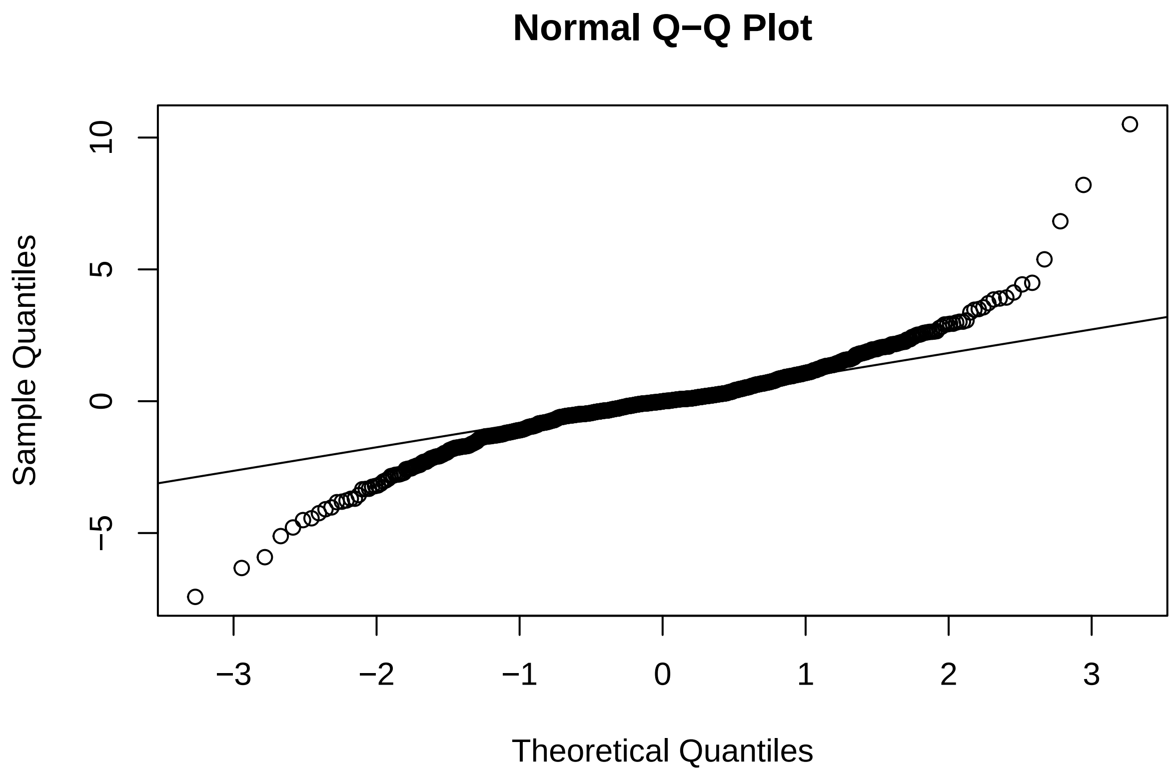
```r
vif(refined_model3)
```

```
##            income_c           income_c2        sqrt_share_c        log_share_c2
##            2.526913            2.403627            1.392025            1.413630
##              months income_months_share          dependents              active
##            1.138728            1.280959            1.176237            1.041303
```

```r
# Check residuals
plot(refined_model3, which=1) # residuals vs fitted
```




```r
qqnorm(resid(refined_model3)); qqline(resid(refined_model3)) # normality
```



Step 7- Evaluate predictive performance on the test set (Individual Work)

```
test_data$income_c <- test_data$income - mean(train_data$income)
test_data$share_c <- test_data$share - mean(train_data$income)
test_data$income_months_share <- test_data$income_c * test_data$months * test_data$share_c
# Centered polynomial terms
test_data$income_c2 <- test_data$income_c^2

test_data$log_share <- log(test_data$share + 1)      # +1 if share has zeros
test_data$log_share_c <- test_data$log_share - mean(train_data$log_share)

test_data$log_share_c2 <- test_data$log_share_c^2

test_data$sqrt_share <- sqrt(test_data$share)
test_data$sqrt_share_c <- test_data$sqrt_share - mean(train_data$sqrt_share)

pred_test <- predict(refined_model3, newdata = test_data)
RMSE_pred <- sqrt(mean((test_data$expenditure - pred_test)^2))
RMSE_pred
```

```
## [1] 389.095
```

```r
final_model <- lm(expenditure ~ income + sqrt(share) + dependents + months + active + share:income,
                  data = train_data)
pred_test <- predict(final_model, newdata = test_data)
# RMSE: square root of mean squared errors
RMSE_final <- sqrt(mean((test_data$expenditure - pred_test)^2))
RMSE_final
```

```
## [1] 0.2173686
```

```r
# Baseline prediction = mean of Y in training data
baseline_pred <- rep(mean(train_data$expenditure), nrow(test_data))

# Baseline RMSE
RMSE_baseline <- sqrt(mean((test_data$expenditure - baseline_pred)^2))

RMSE_baseline
```

```
## [1] 336.2129
```

The RMSE of the transformed model on the test data is 389.095, which is extremely large. Even though this model achieved a very high adjusted R^2 of 0.971 on the training set, the poor test performance indicates that it is heavily overfitting. It captures noise and overly specific patterns from the training data rather than the true underlying relationships. Its training RMSE of 181.257 further shows that it fits the training set well but does not generalize to new observations, the testing set.

In contrast, the refined final model, built using the original response scale and simpler predictor structure, achieves an RMSE of only 0.217 on the test set. This value is dramatically smaller than the baseline RMSE (approximately equal to the standard deviation of the outcome, 336.2129), meaning the refined model predicts extremely well relative to the natural variability in the data. The fact that this model performs far better on the test set, despite being simpler, confirms that reducing unnecessary transformations and complexity improves predictive accuracy and stability.

Conclusion: The comparison clearly shows that the highly transformed model was overfitting, while the simplified final model generalizes substantially better. The huge gap between training and testing RMSE for the transformed model signals high variance and poor predictive reliability. In contrast, the final model's exceptionally small RMSE (0.217) indicates strong predictive power and suggests it is the more appropriate model for this problem.