# Credit Balance Prediction Analysis

Your Name

2023-12-01

## 1. Data Loading and Preprocessing

### 1.1 Load and Clean Data

```
# Load data
data <- read.csv("./data/credit.csv") %>%
  janitor::clean_names()

# Display basic info
cat("Dataset dimensions:", dim(data), "\n")
```

```
## Dataset dimensions: 400 12
```

```
cat("Column names:", names(data), "\n")
```

```
## Column names: x income limit rating cards age education gender student married ethnicity balance
```

### 1.2 Data Cleaning

```
# Remove index column
data <- data[, -1]

# One-hot encoding for categorical variables
data$gender <- trimws(data$gender)
data$male <- ifelse(data$gender == "Male", yes = 1, no = 0)
data$is_student <- ifelse(data$student == "Yes", yes = 1, no = 0)
data$is_married <- ifelse(data$married == "Yes", yes = 1, no = 0)
data$african_american <- ifelse(data$ethnicity == "African American", yes = 1, no = 0)
data$asian <- ifelse(data$ethnicity == 'Asian', yes = 1, no = 0)

# Remove original categorical columns
data <- data %>%
  dplyr::select(-c("gender", "student", "married", "ethnicity"))

# Display structure
str(data)
```

```
## 'data.frame':    400 obs. of  12 variables:
##  $ income          : num  14.9 106 104.6 148.9 55.9 ...
##  $ limit           : int  3606 6645 7075 9504 4897 8047 3388 7114 3300 6819 ...
##  $ rating          : int  283 483 514 681 357 569 259 512 266 491 ...
##  $ cards           : int  2 3 4 3 2 4 2 2 5 3 ...
##  $ age             : int  34 82 71 36 68 77 37 87 66 41 ...
```

```
##  $ education       : int  11 15 11 11 16 10 12 9 13 19 ...
##  $ balance         : int  333 903 580 964 331 1151 203 872 279 1350 ...
##  $ male            : num  1 0 1 0 1 1 0 1 0 0 ...
##  $ is_student      : num  0 1 0 0 0 0 0 0 0 1 ...
##  $ is_married      : num  1 1 0 0 1 0 0 0 0 1 ...
##  $ african_american: num  0 0 0 0 0 0 1 0 0 1 ...
##  $ asian           : num  0 1 1 1 0 0 0 1 0 0 ...
```

# 2. Exploratory Data Analysis (EDA)

## 2.1 Data Structure Overview

```r
cat("=== DATA STRUCTURE ===\n")
```

```
## === DATA STRUCTURE ===
```

```r
head(data) %>% kable()
```

| income | limit | rating | cards | age | education | balance | male | is_student | is_married | african_american | asian |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 14.891 | 3606 | 283 | 2 | 34 | 11 | 333 | 1 | 0 | 1 | 0 | 0 |
| 106.025 | 6645 | 483 | 3 | 82 | 15 | 903 | 0 | 1 | 1 | 0 | 1 |
| 104.593 | 7075 | 514 | 4 | 71 | 11 | 580 | 1 | 0 | 0 | 0 | 1 |
| 148.924 | 9504 | 681 | 3 | 36 | 11 | 964 | 0 | 0 | 0 | 0 | 1 |
| 55.882 | 4897 | 357 | 2 | 68 | 16 | 331 | 1 | 0 | 1 | 0 | 0 |
| 80.180 | 8047 | 569 | 4 | 77 | 10 | 1151 | 1 | 0 | 0 | 0 | 0 |

```r
summary(data) %>% kable()
```

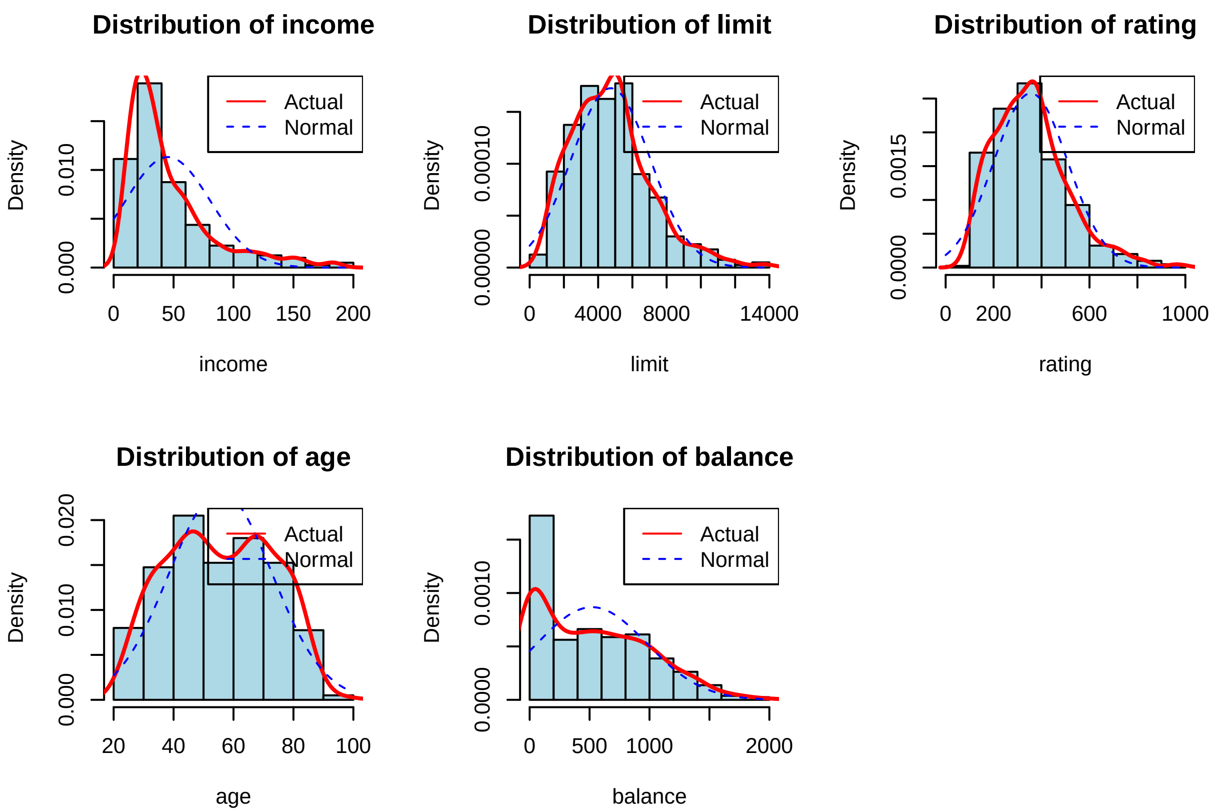
| | income | limit | rating | cards | age | education | balance | male | is_student | is_married | african_american | asian |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Min. : 10.35 | Min. : 855 | Min. : 93.0 | Min. :1.000 | Min. :23.00 | Min. : 5.00 | Min. : 0.00 | Min. :0.0000 | Min. :0.0 | Min. :0.0000 | Min. :0.0000 | Min. :0.000 |
| | 1st Qu.: 21.01 | 1st Qu.: 3088 | 1st Qu.:247.2 | 1st Qu.:2.000 | 1st Qu.:41.75 | 1st Qu.:11.00 | 1st Qu.: 68.75 | 1st Qu.:0.0000 | 1st Qu.:0.0 | 1st Qu.:0.0000 | 1st Qu.:0.0000 | 1st Qu.:0.000 |
| | Median : 33.12 | Median : 4622 | Median :344.0 | Median :3.000 | Median :56.00 | Median :14.00 | Median : 459.50 | Median :0.0000 | Median :0.0 | Median :1.0000 | Median :0.0000 | Median :0.000 |
| | Mean : 45.22 | Mean : 4736 | Mean :354.9 | Mean :2.958 | Mean :55.67 | Mean :13.45 | Mean : 520.01 | Mean :0.4825 | Mean :0.1 | Mean :0.6125 | Mean :0.2475 | Mean :0.255 |
| | 3rd Qu.: 57.47 | 3rd Qu.: 5873 | 3rd Qu.:437.2 | 3rd Qu.:4.000 | 3rd Qu.:70.00 | 3rd Qu.:16.00 | 3rd Qu.: 863.00 | 3rd Qu.:1.0000 | 3rd Qu.:0.0 | 3rd Qu.:1.0000 | 3rd Qu.:0.0000 | 3rd Qu.:1.000 |
| | Max. :186.63 | Max. :13913 | Max. :982.0 | Max. :9.000 | Max. :98.00 | Max. :20.00 | Max. :1999.00 | Max. :1.0000 | Max. :1.0 | Max. :1.0000 | Max. :1.0000 | Max. :1.000 |

**Observation:** The dataset contains credit card information with balance as our target variable. Note that balance has a minimum of 0, suggesting we may need to address truncated normal distribution in our transformations.

## 2.2 Distribution Analysis

```r
# Set up plotting area
par(mfrow = c(2, 3))

# Histograms with density curves for continuous variables
for(var in c("income", "limit", "rating", "age", "balance")) {
  hist(data[[var]], main = paste("Distribution of", var),
       xlab = var, prob = TRUE, col = "lightblue")
  lines(density(data[[var]]), col = "red", lwd = 2)
  curve(dnorm(x, mean = mean(data[[var]]), sd = sd(data[[var]])),
        add = TRUE, col = "blue", lty = 2)
  legend("topright", legend = c("Actual", "Normal"),
         col = c("red", "blue"), lty = c(1, 2))
}
```




**Observation from Histogram (ACTUAL) red curve:**

- Income: Right-skewed with long tail (red curve peaks left, extends right)
- Limit: Right-skewed with long tail
- Rating: Right-skewed with long tail
- Age: Seems normal (red curve matches blue reasonably well). Though it has two peaks (take notice).
- Balance: Right-skewed with long tail

Note: Plotted curves look truncated. Perhaps they may benefit from transformation. Here are other reasons for truncated normal curves in this dataset:

- Income: Can't be negative, often clustered above minimum wage

- Credit Limit: Always positive, often has minimum thresholds
- Balance: Can be 0 but not negative (unless it's debt)
- Rating: Often has minimum scores

### 2.3 Normality Assessment

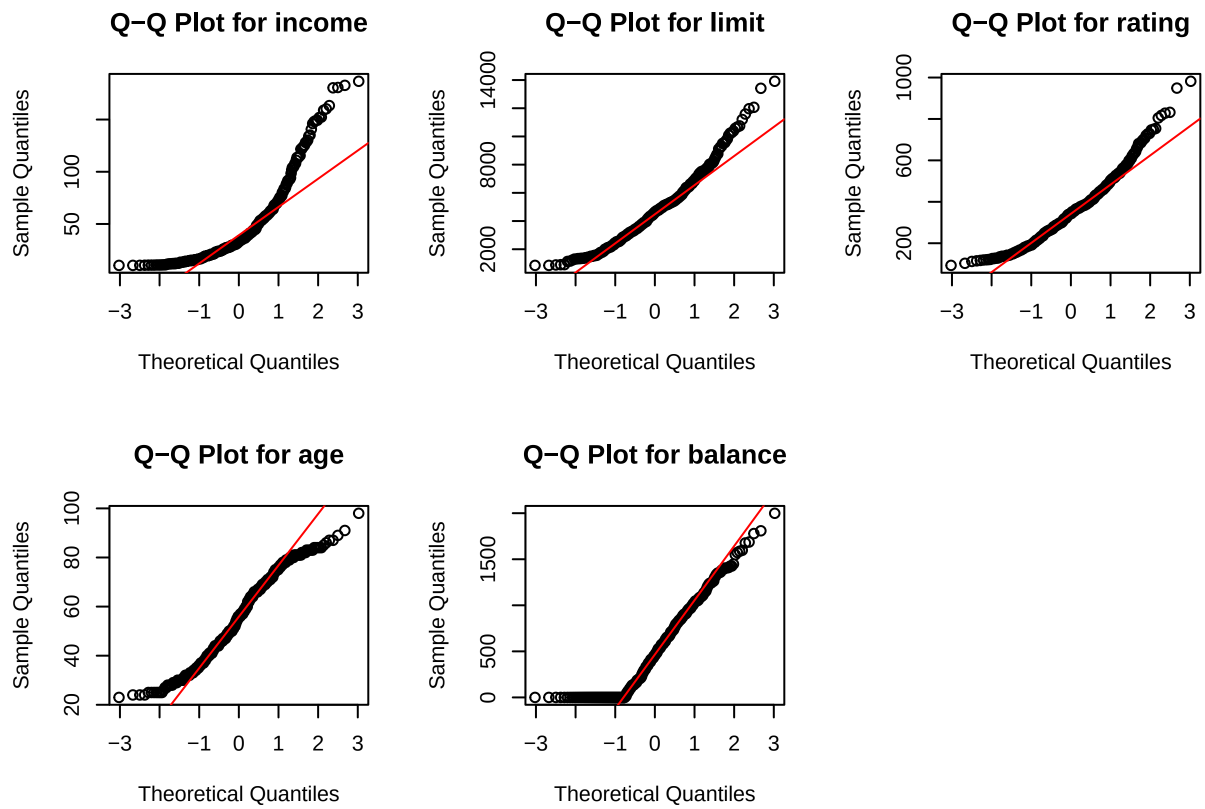
```r
# Q-Q plots for normality
par(mfrow = c(2, 3))
for(var in c("income", "limit", "rating", "age", "balance")) {
  qqnorm(data[[var]], main = paste("Q-Q Plot for", var))
  qqline(data[[var]], col = "red")
}

# Skewness calculation
skew_values <- sapply(data[, c("income", "limit", "rating", "balance", "age")], skewness)
cat("=== SKEWNESS VALUES ===\n")
```

```
## === SKEWNESS VALUES ===
```

```r
kable(data.frame(Variable = names(skew_values), Skewness = round(skew_values, 3)))
```

| | Variable | Skewness |
|---|---|---|
| income | income | 1.736 |
| limit | limit | 0.834 |
| rating | rating | 0.862 |
| balance | balance | 0.582 |
| age | age | 0.011 |




**Noted: Rule of thumb:**

- $|\text{skewness}| < 0.5$ : approximately symmetric (probably OK as-is)
- $0.5 < |\text{skewness}| < 1$ : moderate skew (consider transformation)
- $|\text{skewness}| > 1$ : substantial skew (definitely transform)

# 3. Variable Transformations

## 3.1 Apply Transformations

```
# Apply transformations
data <- data %>%
  mutate(
    log_income = log(income),
    log_limit = log(limit),
    log_rating = log(rating),
    log_balance = log(balance + 1),
    sqrt_balance = sqrt(balance)
  )

# Check transformed skewness
transformed_skew <- sapply(data[, c("log_income", "log_limit", "log_rating", "sqrt_balance", "age")], s
cat("=== TRANSFORMED SKEWNESS VALUES ===\n")
```

```
## === TRANSFORMED SKEWNESS VALUES ===
```

```
kable(data.frame(Variable = names(transformed_skew), Skewness = round(transformed_skew, 3)))
```

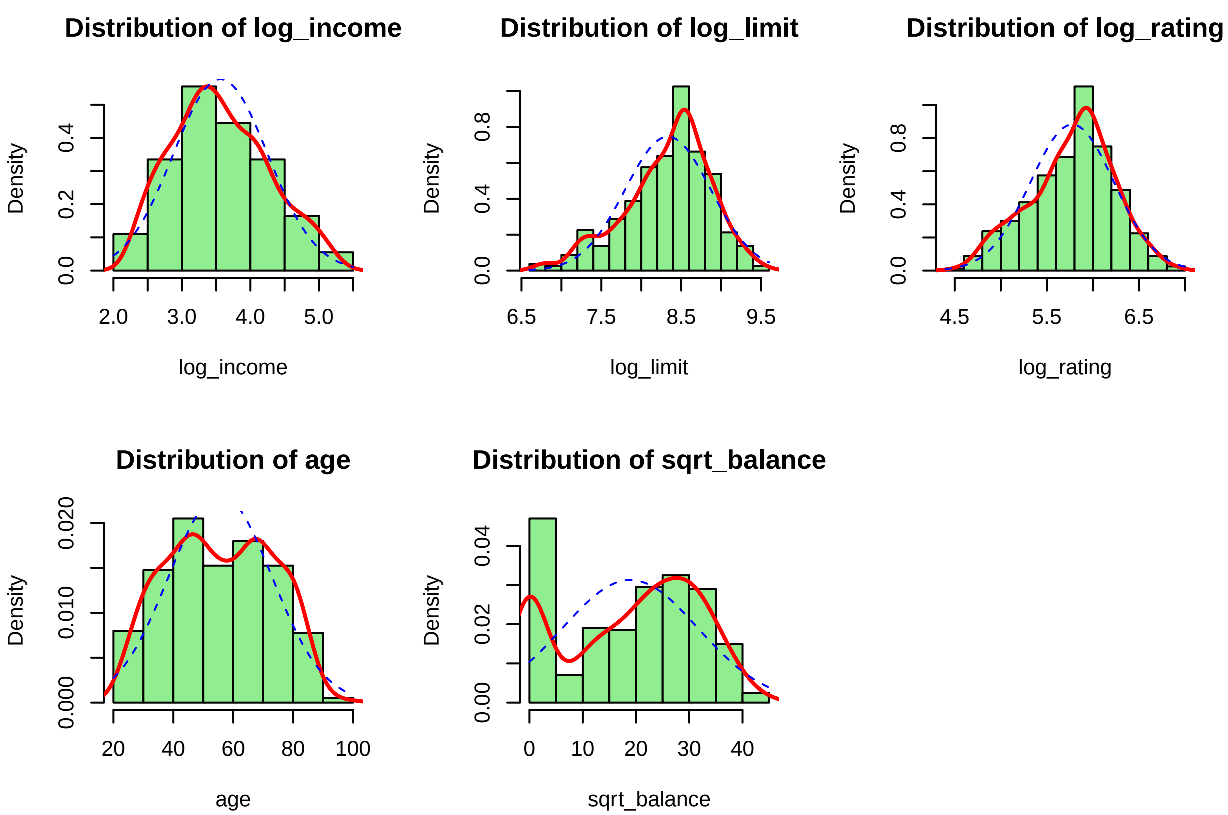
| | Variable | Skewness |
|---|---|---|
| log_income | log_income | 0.305 |
| log_limit | log_limit | -0.577 |
| log_rating | log_rating | -0.312 |
| sqrt_balance | sqrt_balance | -0.274 |
| age | age | 0.011 |

**Observation:**

1. Other variables look ok,
2. Moderate skewness for log_limit: log-limit is now left-skewed
3. Huge skewness for log_balance

## 3.2 Visualize Transformed Distributions

```r
par(mfrow = c(2, 3))
for(var in c("log_income", "log_limit", "log_rating", "age", "sqrt_balance")) {
  hist(data[[var]], main = paste("Distribution of", var),
       xlab = var, prob = TRUE, col = "lightgreen")
  lines(density(data[[var]]), col = "red", lwd = 2)
  curve(dnorm(x, mean = mean(data[[var]]), sd = sd(data[[var]])),
        add = TRUE, col = "blue", lty = 2)
}
```



# 4. Initial Model Building

## 4.1 Fit Multiple Models

```
# Prepare model data
model_data <- data

# Fit three different models
model_original <- lm(balance ~ log_income + log_limit + log_rating +
                       cards + age + education + male + is_student +
                       is_married + african_american + asian, data = model_data)

model_sqrt <- lm(sqrt_balance ~ log_income + log_limit + log_rating +
                   cards + age + education + male + is_student +
                   is_married + african_american + asian, data = model_data)

model_log <- lm(log_balance ~ log_income + log_limit + log_rating +
                  cards + age + education + male + is_student +
                  is_married + african_american + asian, data = model_data)

# Display model summaries
cat("=== ORIGINAL BALANCE MODEL ===\n")
```

```
## === ORIGINAL BALANCE MODEL ===
```

```
summary(model_original)
```

```
##
## Call:
## lm(formula = balance ~ log_income + log_limit + log_rating +
##     cards + age + education + male + is_student + is_married +
##     african_american + asian, data = model_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -503.55 -125.68  -10.46  127.27  605.55
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      -3503.9804   276.4449 -12.675 < 0.0000000000000002 ***
## log_income        -192.0875    18.7953 -10.220 < 0.0000000000000002 ***
## log_limit         -884.1309   153.5607  -5.758         0.0000000174 ***
## log_rating        2096.2952   185.6196  11.294 < 0.0000000000000002 ***
## cards               -9.8316     7.6165  -1.291                0.198
## age                 -0.5340     0.5593  -0.955                0.340
## education            0.3998     3.0436   0.131                0.896
## male                 4.9313    18.9009   0.261                0.794
## is_student         397.5554    31.7848  12.508 < 0.0000000000000002 ***
## is_married         -29.7437    19.7859  -1.503                0.134
## african_american   -17.7632    23.3163  -0.762                0.447
## asian               17.2975    23.0685   0.750                0.454
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 188.3 on 388 degrees of freedom
```

```
## Multiple R-squared:  0.8369, Adjusted R-squared:  0.8323
## F-statistic:   181 on 11 and 388 DF,  p-value: < 0.00000000000000022
```

```r
cat("\n=== SQRT BALANCE MODEL ===\n")
```

```
##
## === SQRT BALANCE MODEL ===
```

```r
summary(model_sqrt)
```

```
##
## Call:
## lm(formula = sqrt_balance ~ log_income + log_limit + log_rating +
##     cards + age + education + male + is_student + is_married +
##     african_american + asian, data = model_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8883  -1.9158   0.6259   2.3400  11.1115
##
## Coefficients:
##                    Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      -123.61188    5.45375 -22.665 < 0.0000000000000002 ***
## log_income         -7.24468    0.37080 -19.538 < 0.0000000000000002 ***
## log_limit         -10.49381    3.02947  -3.464             0.000592 ***
## log_rating         44.60908    3.66194  12.182 < 0.0000000000000002 ***
## cards              -0.21265    0.15026  -1.415             0.157811
## age                -0.01458    0.01103  -1.321             0.187208
## education          -0.06789    0.06005  -1.131             0.258876
## male                0.03040    0.37288   0.082             0.935065
## is_student         10.10693    0.62706  16.118 < 0.0000000000000002 ***
## is_married         -0.30304    0.39034  -0.776             0.438019
## african_american   -0.83632    0.45999  -1.818             0.069814 .
## asian              -0.38699    0.45510  -0.850             0.395661
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.714 on 388 degrees of freedom
## Multiple R-squared:  0.9174, Adjusted R-squared:  0.9151
## F-statistic:   392 on 11 and 388 DF,  p-value: < 0.00000000000000022
```

```r
cat("\n=== LOG BALANCE MODEL ===\n")
```

```
##
## === LOG BALANCE MODEL ===
```

```r
summary(model_log)
```

```
##
## Call:
## lm(formula = log_balance ~ log_income + log_limit + log_rating +
##     cards + age + education + male + is_student + is_married +
##     african_american + asian, data = model_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4962 -0.4820  0.1532  0.8175  2.9883
```

```
##
## Coefficients:
##                      Estimate   Std. Error t value            Pr(>|t|)    
## (Intercept)      -30.08179189   1.65803158 -18.143 < 0.0000000000000002 ***
## log_income        -1.71170246   0.11272818 -15.184 < 0.0000000000000002 ***
## log_limit          1.66635616   0.92100984   1.809               0.0712 .  
## log_rating         4.76038848   1.11328935   4.276             0.000024 ***
## cards              0.00003289   0.04568132   0.001               0.9994    
## age               -0.00236752   0.00335442  -0.706               0.4807    
## education         -0.02394918   0.01825479  -1.312               0.1903    
## male              -0.05893602   0.11336191  -0.520               0.6034    
## is_student         1.80937101   0.19063544   9.491 < 0.0000000000000002 ***
## is_married         0.06956746   0.11866947   0.586               0.5581    
## african_american  -0.18223287   0.13984415  -1.303               0.1933    
## asian             -0.27603334   0.13835795  -1.995               0.0467 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.129 on 388 degrees of freedom
## Multiple R-squared:  0.8327, Adjusted R-squared:  0.828 
## F-statistic: 175.6 on 11 and 388 DF,  p-value: < 0.00000000000000022
```

### 4.2 Multicollinearity Check

```r
# VIF Analysis
cat("=== VARIANCE INFLATION FACTOR (VIF) ===\n")
```

```
## === VARIANCE INFLATION FACTOR (VIF) ===
```

```r
vif_results <- data.frame(
  Original = vif(model_original),
  Sqrt = vif(model_sqrt),
  Log = vif(model_log)
)
kable(vif_results)
```

| | Original | Sqrt | Log |
|---|---|---|---|
| log_income | 1.901236 | 1.901236 | 1.901236 |
| log_limit | 76.031766 | 76.031766 | 76.031766 |
| log_rating | 79.324432 | 79.324432 | 79.324432 |
| cards | 1.227926 | 1.227926 | 1.227926 |
| age | 1.047732 | 1.047732 | 1.047732 |
| education | 1.018490 | 1.018490 | 1.018490 |
| male | 1.006641 | 1.006641 | 1.006641 |
| is_student | 1.026083 | 1.026083 | 1.026083 |
| is_married | 1.048549 | 1.048549 | 1.048549 |
| african_american | 1.142624 | 1.142624 | 1.142624 |
| asian | 1.140874 | 1.140874 | 1.140874 |

**Observation:** Credit limit and credit rating are linearly dependent. VIF: credit limit (76.031766), credit rating (79.324432). Some correlation there.

### 4.3 Address Multicollinearity

```r
# Question 1: Is model better without credit limit?
model_original_noLimit <- lm(balance ~ log_income + log_rating +
                        cards + age + education + male + is_student +
                        is_married + african_american + asian, data = data)

model_sqrt_noLimit <- lm(sqrt_balance ~ log_income + log_rating +
                    cards + age + education + male + is_student +
                    is_married + african_american + asian, data = data)

model_log_noLimit <- lm(log_balance ~ log_income + log_rating +
                  cards + age + education + male + is_student +
                  is_married + african_american + asian, data = data)

# Compare models
cat("=== MODEL COMPARISON: WITH VS WITHOUT LIMIT ===\n")
```

```
## === MODEL COMPARISON: WITH VS WITHOUT LIMIT ===
```

```r
cat("Original R-squared:", summary(model_original)$r.squared, 4)
```

```
## Original R-squared: 0.8369363 4
```

```r
cat("Without Limit R-squared:", round(summary(model_original_noLimit)$r.squared, 4), "\n")
```

```
## Without Limit R-squared: 0.823
```

**Conclusion**: Model seems better overall without credit limit. Moreover, credit limit is dependent on credit rating i.e Lenders use credit rating to determine credit limit. Note:

- Higher-rated borrowers are less risky, so they get higher limits
- Lower-rated borrowers are more risky, so they get lower limits to limit exposure

```r
# Update models
model_original <- model_original_noLimit
model_log <- model_log_noLimit
model_sqrt <- model_sqrt_noLimit

# Check to see if our vif looks good: Multicollinearity
cat("=== UPDATED VIF (WITHOUT LIMIT) ===\n")
```

```
## === UPDATED VIF (WITHOUT LIMIT) ===
```

```r
kable(data.frame(Variable = names(vif(model_original)), VIF = vif(model_original)))
```

| | Variable | VIF |
|---|---|---|
| log_income | log_income | 1.868810 |
| log_rating | log_rating | 1.825959 |
| cards | cards | 1.023670 |
| age | age | 1.047290 |
| education | education | 1.014735 |
| male | male | 1.005095 |
| is_student | is_student | 1.021294 |
| is_married | is_married | 1.030785 |
| african_american | african_american | 1.137018 |
| asian | asian | 1.139694 |

# 5. Model Refinement

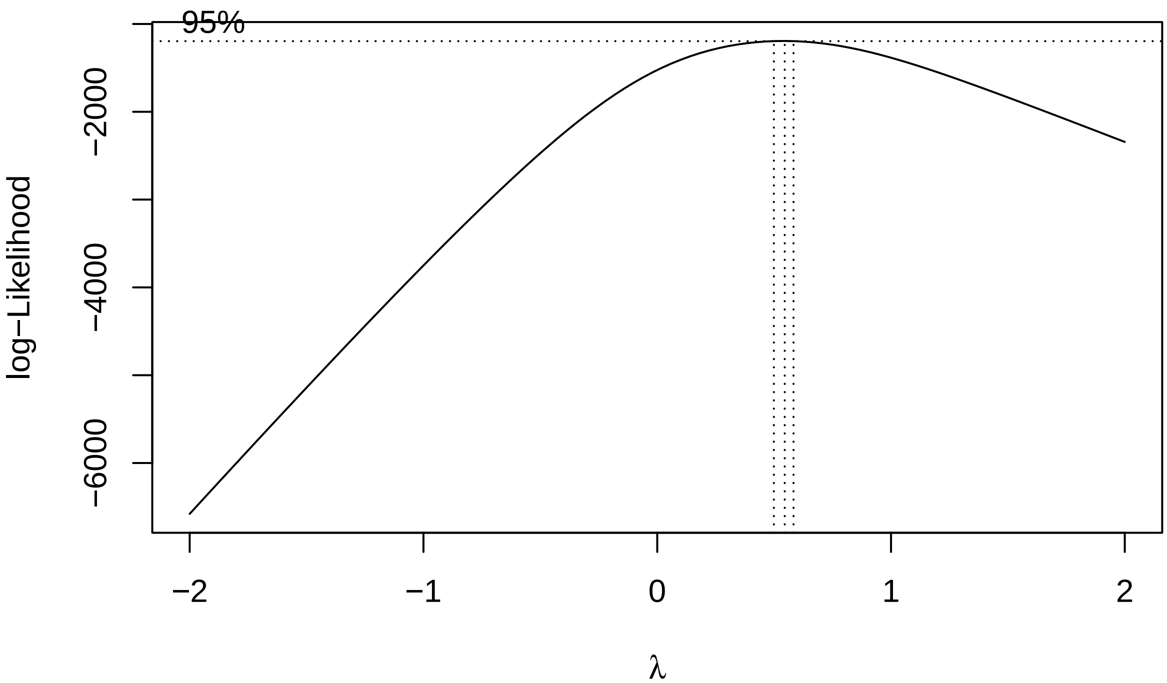
## 5.1 Boxcox Transformation

```r
# Step 6: Residual Diagnostics

# Check if sqrt_balance has non-positive values
summary(model_data$sqrt_balance)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   8.292  21.436  18.919  29.377  44.710
```

```r
# Check if sqrt_balance has non-positive values
min_value <- min(model_data$sqrt_balance)
if (min_value <= 0) {
  model_data$shifted_sqrt <- model_data$sqrt_balance - min_value + 0.001
  m_shifted <- lm(shifted_sqrt ~ log_income + is_student + log_rating, data = model_data)
  bc <- boxcox(m_shifted)
}
```




```r
# Get optimal lambda
lambda <- bc$x[which.max(bc$y)]
cat("Optimal lambda:", lambda, "\n")
```

```
## Optimal lambda: 0.5454545
```

```r
# Apply the Box-Cox transformation to the ORIGINAL balance variable
if (abs(lambda) < 0.001) {
  # If lambda   0, use log transformation
  model_data$bc_balance <- log(model_data$balance)
} else {
  # Standard Box-Cox transformation
  model_data$bc_balance <- (model_data$balance^lambda - 1) / lambda
}
```

```r
# Fit the new model with Box-Cox transformed balance
m_bc <- lm(bc_balance ~ log_income + as.factor(is_student) + log_rating, data = model_data)
cat("=== BOX-COX TRANSFORMED MODEL ===\n")
```

```
## === BOX-COX TRANSFORMED MODEL ===
```

```r
summary(m_bc)
```

```
##
## Call:
## lm(formula = bc_balance ~ log_income + as.factor(is_student) +
##     log_rating, data = model_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.607  -5.309   0.791   6.208  33.745
##
## Coefficients:
##                         Estimate Std. Error t value            Pr(>|t|)
## (Intercept)            -361.8621     6.6155  -54.70 <0.0000000000000002 ***
## log_income              -17.6254     0.9433  -18.68 <0.0000000000000002 ***
## as.factor(is_student)1   26.2432     1.6237   16.16 <0.0000000000000002 ***
## log_rating               80.8478     1.4421   56.06 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.739 on 396 degrees of freedom
## Multiple R-squared:  0.9098, Adjusted R-squared:  0.9091
## F-statistic:  1332 on 3 and 396 DF,  p-value: < 0.00000000000000022
```

### 5.2 Alternative Transformation: Square Root Income

```r
# Try square root transformation for income

model_data <- model_data %>% mutate(income_sqrt = income^(1/2))

model_original <- lm(balance ~ cards + age + education + male + income_sqrt + african_american + is_mar

cat("=== MODEL WITH SQRT INCOME ===\n")
```

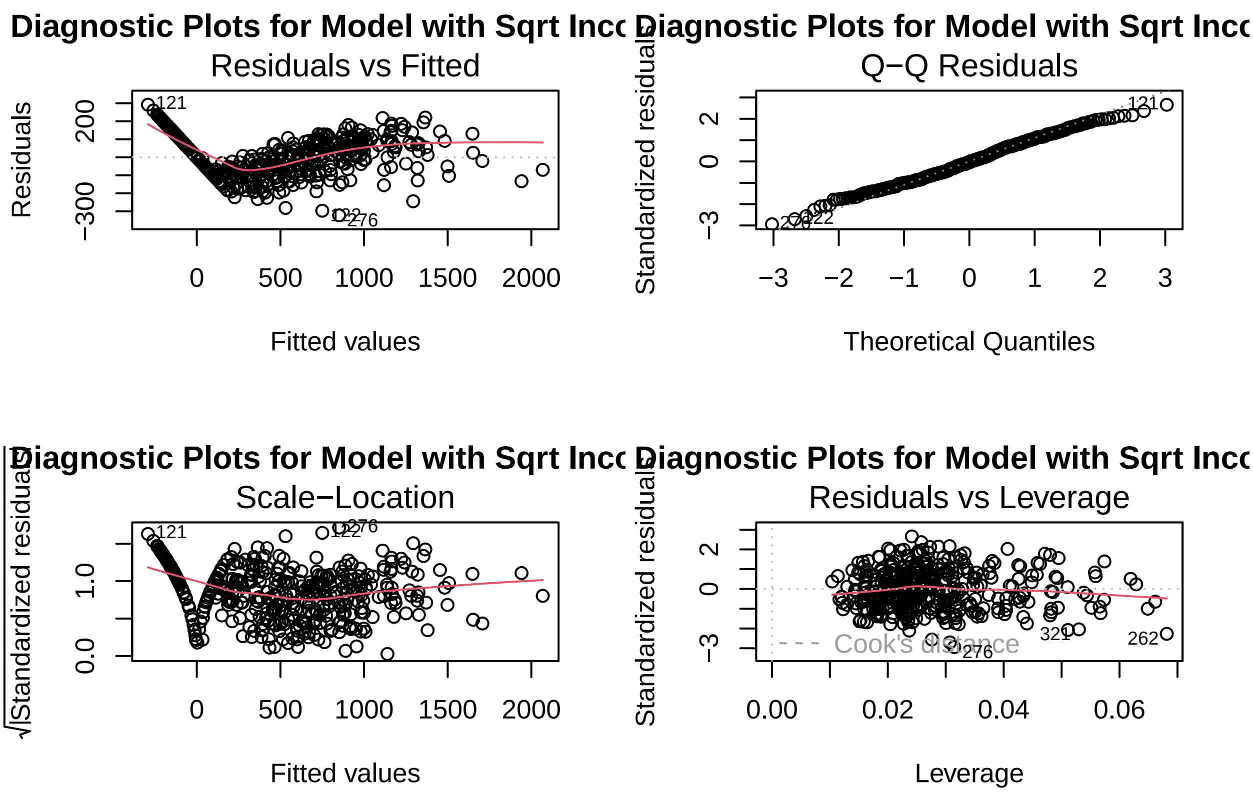
```
## === MODEL WITH SQRT INCOME ===
```

```r
summary(model_original)
```

```
##
## Call:
## lm(formula = balance ~ cards + age + education + male + income_sqrt +
##     african_american + is_married + asian + is_student + rating,
##     data = model_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -322.12  -80.98   -4.38   81.68  291.97
##
## Coefficients:
```

```
##                      Estimate Std. Error t value             Pr(>|t|)
## (Intercept)        -133.49567   36.89806  -3.618             0.000336 ***
## cards                 2.47992    4.10817   0.604             0.546425
## age                  -0.73611    0.33072  -2.226             0.026603 *
## education            -0.96650    1.79431  -0.539             0.590439
## male                  6.88467   11.15864   0.617             0.537608
## income_sqrt        -110.48804    3.87433 -28.518 < 0.0000000000000002 ***
## african_american    -16.77020   13.74438  -1.220             0.223147
## is_married          -20.38100   11.59504  -1.758             0.079579 .
## asian                 3.73617   13.61673   0.274             0.783938
## is_student          412.43283   18.73766  22.011 < 0.0000000000000002 ***
## rating                3.85700    0.05737  67.226 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 111.2 on 389 degrees of freedom
## Multiple R-squared:  0.9429, Adjusted R-squared:  0.9415
## F-statistic: 642.8 on 10 and 389 DF,  p-value: < 0.00000000000000022
```

```r
# Diagnostic plots
par(mfrow = c(2,2))
plot(model_original, main = "Diagnostic Plots for Model with Sqrt Income")
```




**Conclusion for Residual Diagnostics:**

- The Models from the Initial Model Selection violated Normality Assumption
- I did a boxcox transformation which didn't change much
- However, a sqrt transformation on income made the Residual vs Fitted Values plot more agreeable and satisfactory.

- Hence I moved forward with: `model_original <- lm(balance ~ cards + age + education + male + income_sqrt + african_american + is_married + asian + is_student + rating, data = model_data)`
- I performed more model selection on model_original vs sqrt_balance model using income_sqrt variable instead of income.

# 6. Advanced Model Selection

## 6.1 Stepwise Selection Functions

```
# Refined Model Selection - Part 1
# Step 5: Model Selection
# Model Selection Comparison for All Response Variables

# Function to run stepwise selection for a given response
run_stepwise_selection <- function(response_var, data) {
  # Create formula strings
  full_formula <- as.formula(paste(response_var, " ~ cards + age + education + male + income_sqrt + afri
  empty_formula <- as.formula(paste(response_var, "~ 1"))

  # Fit models
  full_model <- lm(full_formula, data = data)
  empty_model <- lm(empty_formula, data = data)

  # Stepwise selection
  forward_aic <- stepAIC(empty_model,
                         scope = list(upper = full_model, lower = ~1),
                         direction = "forward", trace = 0)

  forward_bic <- stepAIC(empty_model,
                         scope = list(upper = full_model, lower = ~1),
                         direction = "forward", k = log(nrow(data)), trace = 0)

  backward_aic <- stepAIC(full_model, direction = "backward", trace = 0)
  backward_bic <- stepAIC(full_model, direction = "backward",
                          k = log(nrow(data)), trace = 0)

  stepwise_aic <- stepAIC(empty_model,
                          scope = list(upper = full_model, lower = ~1),
                          direction = "both", trace = 0)

  stepwise_bic <- stepAIC(empty_model,
                          scope = list(upper = full_model, lower = ~1),
                          direction = "both", k = log(nrow(data)), trace = 0)

  # Return all models
  return(list(
    forward_aic = forward_aic,
    forward_bic = forward_bic,
    backward_aic = backward_aic,
    backward_bic = backward_bic,
    stepwise_aic = stepwise_aic,
    stepwise_bic = stepwise_bic
```

```r
  ))
}

# Function to extract model metrics
get_model_metrics <- function(model, model_name, response_name) {
  model_summary <- summary(model)

  data.frame(
    Response = response_name,
    Method = model_name,
    AIC = round(AIC(model), 2),
    BIC = round(BIC(model), 2),
    R_squared = round(model_summary$r.squared, 4),
    Adj_R_squared = round(model_summary$adj.r.squared, 4),
    Num_Predictors = length(coef(model)) - 1,
    Predictors = paste(names(coef(model))[-1], collapse = ", "),
    stringsAsFactors = FALSE
  )
}
```

### 6.2 Comprehensive Model Comparison

```r
# Run stepwise selection for all three response variables
balance_models <- run_stepwise_selection("balance", model_data)
log_balance_models <- run_stepwise_selection("log_balance", model_data)
sqrt_balance_models <- run_stepwise_selection("sqrt_balance", model_data)

# Create comparison table
comparison_table <- rbind(
  # Balance models
  get_model_metrics(balance_models$forward_aic, "Forward AIC", "balance"),
  get_model_metrics(balance_models$forward_bic, "Forward BIC", "balance"),
  get_model_metrics(balance_models$backward_aic, "Backward AIC", "balance"),
  get_model_metrics(balance_models$backward_bic, "Backward BIC", "balance"),
  get_model_metrics(balance_models$stepwise_aic, "Stepwise AIC", "balance"),
  get_model_metrics(balance_models$stepwise_bic, "Stepwise BIC", "balance"),

  # Log balance models
  get_model_metrics(log_balance_models$forward_aic, "Forward AIC", "log_balance"),
  get_model_metrics(log_balance_models$forward_bic, "Forward BIC", "log_balance"),
  get_model_metrics(log_balance_models$backward_aic, "Backward AIC", "log_balance"),
  get_model_metrics(log_balance_models$backward_bic, "Backward BIC", "log_balance"),
  get_model_metrics(log_balance_models$stepwise_aic, "Stepwise AIC", "log_balance"),
  get_model_metrics(log_balance_models$stepwise_bic, "Stepwise BIC", "log_balance"),

  # Sqrt balance models
  get_model_metrics(sqrt_balance_models$forward_aic, "Forward AIC", "sqrt_balance"),
  get_model_metrics(sqrt_balance_models$forward_bic, "Forward BIC", "sqrt_balance"),
  get_model_metrics(sqrt_balance_models$backward_aic, "Backward AIC", "sqrt_balance"),
  get_model_metrics(sqrt_balance_models$backward_bic, "Backward BIC", "sqrt_balance"),
  get_model_metrics(sqrt_balance_models$stepwise_aic, "Stepwise AIC", "sqrt_balance"),
  get_model_metrics(sqrt_balance_models$stepwise_bic, "Stepwise BIC", "sqrt_balance")
)
```

```r
cat("=== COMPREHENSIVE MODEL SELECTION RESULTS ===\n")
```

```
## === COMPREHENSIVE MODEL SELECTION RESULTS ===
```

```r
kable(comparison_table %>% arrange(Response, AIC))
```

| Response | Method | AIC | BIC | R_squared | Adj_R_squared | Num_Predictors | Predictors |
|---|---|---|---|---|---|---|---|
| balance | Forward AIC | 4910.44 | 4938.38 | 0.9425 | 0.9418 | 5 | rating, income_sqrt, is_student, age, is_married |
| balance | Backward AIC | 4910.44 | 4938.38 | 0.9425 | 0.9418 | 5 | age, income_sqrt, is_married, is_student, rating |
| balance | Stepwise AIC | 4910.44 | 4938.38 | 0.9425 | 0.9418 | 5 | rating, income_sqrt, is_student, age, is_married |
| balance | Forward BIC | 4913.98 | 4933.93 | 0.9414 | 0.9409 | 3 | rating, income_sqrt, is_student |
| balance | Backward BIC | 4913.98 | 4933.93 | 0.9414 | 0.9409 | 3 | income_sqrt, is_student, rating |
| balance | Stepwise BIC | 4913.98 | 4933.93 | 0.9414 | 0.9409 | 3 | rating, income_sqrt, is_student |
| log_balance | Forward AIC | 1434.08 | 1458.03 | 0.7229 | 0.7201 | 4 | rating, income_sqrt, is_student, asian |
| log_balance | Backward AIC | 1434.08 | 1458.03 | 0.7229 | 0.7201 | 4 | income_sqrt, asian, is_student, rating |
| log_balance | Stepwise AIC | 1434.08 | 1458.03 | 0.7229 | 0.7201 | 4 | rating, income_sqrt, is_student, asian |
| log_balance | Forward BIC | 1434.44 | 1454.39 | 0.7213 | 0.7192 | 3 | rating, income_sqrt, is_student |
| log_balance | Backward BIC | 1434.44 | 1454.39 | 0.7213 | 0.7192 | 3 | income_sqrt, is_student, rating |
| log_balance | Stepwise BIC | 1434.44 | 1454.39 | 0.7213 | 0.7192 | 3 | rating, income_sqrt, is_student |
| sqrt_balance | Forward AIC | 2144.39 | 2180.31 | 0.9265 | 0.9252 | 7 | rating, income_sqrt, is_student, age, education, african_american, asian |
| sqrt_balance | Backward AIC | 2144.39 | 2180.31 | 0.9265 | 0.9252 | 7 | age, education, income_sqrt, african_american, asian, is_student, rating |
| sqrt_balance | Stepwise AIC | 2144.39 | 2180.31 | 0.9265 | 0.9252 | 7 | rating, income_sqrt, is_student, age, education, african_american, asian |
| sqrt_balance | Forward BIC | 2150.10 | 2170.05 | 0.9239 | 0.9233 | 3 | rating, income_sqrt, is_student |
| sqrt_balance | Backward BIC | 2150.10 | 2170.05 | 0.9239 | 0.9233 | 3 | income_sqrt, is_student, rating |
| sqrt_balance | Stepwise BIC | 2150.10 | 2170.05 | 0.9239 | 0.9233 | 3 | rating, income_sqrt, is_student |

```r
# Create a summary table showing best model for each response
best_models_summary <- comparison_table %>%
  group_by(Response) %>%
  slice(which.min(AIC)) %>%
  dplyr::select(Response, Method, AIC, BIC, R_squared, Adj_R_squared, Num_Predictors, Predictors)

cat("=== BEST MODELS BY RESPONSE ===\n")
```

```
## === BEST MODELS BY RESPONSE ===
```

```
kable(best_models_summary)
```

| Response | Method | AIC | BIC | R_squared | Adj_R_squared | Num_Predictors | Predictors |
|---|---|---|---|---|---|---|---|
| balance | Forward AIC | 4910.44 | 4938.38 | 0.9425 | 0.9418 | 5 | rating, income_sqrt, is_student, age, is_married |
| log_balance | Forward AIC | 1434.08 | 1458.03 | 0.7229 | 0.7201 | 4 | rating, income_sqrt, is_student, asian |
| sqrt_balance | Forward AIC | 2144.39 | 2180.31 | 0.9265 | 0.9252 | 7 | rating, income_sqrt, is_student, age, education, african_american, asian |

**Model Selection Conclusion:** We choose balance model why?

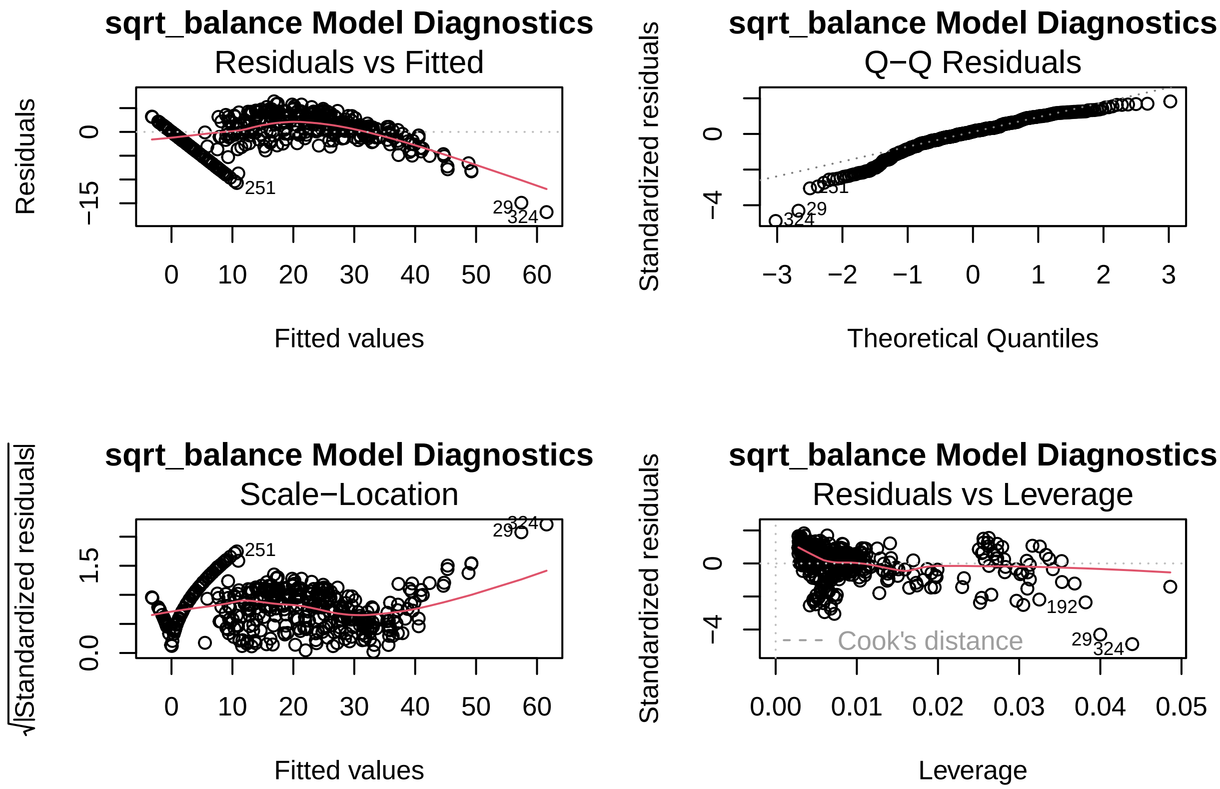
- Variable transformation and Residual Diagnostics comparison showed balance model works is best
- Original Model after VIF and Transformation: `model_original <- lm(balance ~ cards + age + education + male + income_sqrt + african_american + is_married + asian + is_student + rating, data = model_data)`
- At this time, the balance model and sqrt_balance are both great options, however, Residual Diagnostics in next step show that model might be first choice due to having a more agreeable Residual vs Fitted Values plot

# 7. Model Validation

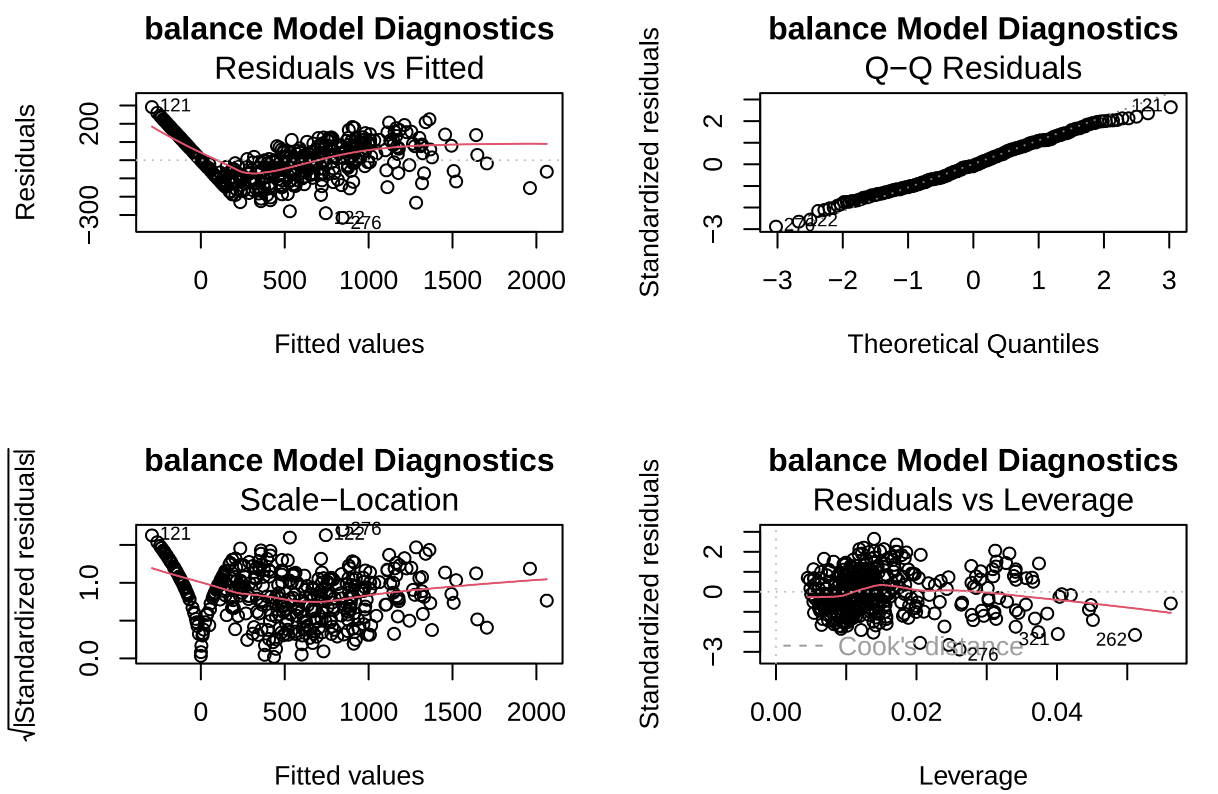
## 7.1 Cross-Validation Setup

```
# Step 6: Model Diagnostics after Refined Model Selection:
model_data <- model_data %>%
  dplyr::select(balance, log_balance, sqrt_balance, cards, age, education, male,
                income_sqrt, african_american, is_married, asian,
                is_student, rating)

# 2nd Choice: sqrt_balance model:
model_best2 <- lm(sqrt(balance) ~ rating + income_sqrt + is_student, data = model_data)
par(mfrow = c(2,2))
plot(model_best2, main = "sqrt_balance Model Diagnostics")
```




```r
# 1st choice: balance model
model_best <- lm(balance ~ rating + income_sqrt + is_student + age + is_married, data = model_data)
par(mfrow = c(2,2))
plot(model_best, main = "balance Model Diagnostics")
```



**Residual Diagnostics**: balance model is more agreeable

### 7.2 Cross-Validation

```
# Step 7: Cross-Validation for Refined Model Selection: K-Fold Cross-Validation
set.seed(12345678)

# CV to choose the best one
K <- 10
N <- nrow(model_data)
validSetSplits <-sample((1:N)%%K + 1)
RMSE_best <- numeric(K)
RMSE_sqrt <- numeric(K)

for (k in 1:K) {
  validSet <- model_data[validSetSplits == k, ]
  trainSet <- model_data[validSetSplits != k, ]

  # Model 1: Model_best 1
  model_best <- lm(balance ~ rating + income_sqrt + is_student + age + is_married, data = trainSet)
  pred1 <- predict(model_best, newdata = validSet)
  RMSE_best[k] <- sqrt(mean((validSet$balance- pred1)^2))

  # Model 2: Model Best 2: - CONVERT BACK to dollars
  model_sqrt <- lm(sqrt(balance) ~ rating + income_sqrt + is_student, data = trainSet)
  pred2_sqrt <- predict(model_sqrt, newdata = validSet)
  pred2_dollars <- pred2_sqrt^2  # Convert back to original scale
  RMSE_sqrt[k] <- sqrt(mean((validSet$balance - pred2_dollars)^2))
}

# Now compare properly
cat("Balance Model RMSE:", mean(RMSE_best), "\n")
```

```
## Balance Model RMSE: 111.4334
```

```
cat("Sqrt Balance Model RMSE:", mean(RMSE_sqrt), "\n")
```

```
## Sqrt Balance Model RMSE: 169.3835
```

```
# Calculate improvement percentage
improvement <- ((mean(RMSE_sqrt) - mean(RMSE_best)) / mean(RMSE_sqrt)) * 100

# Final model selection based on CV performance
final_model <- lm(balance ~ rating + income_sqrt + is_student + age + is_married,
                  data = model_data)

cat("=== FINAL MODEL SELECTED ===\n")
```

```
## === FINAL MODEL SELECTED ===
```

```
cat("Based on cross-validation, model_best has superior performance:\n")
```

```
## Based on cross-validation, model_best has superior performance:
```

```
cat("RMSE:", round(mean(RMSE_best), 2), "vs", round(mean(RMSE_sqrt), 2),
    "(", round(improvement, 1), "% improvement)\n")
```

```
## RMSE: 111.43 vs 169.38 ( 34.2 % improvement)
```

```
cat("Formula: balance ~ rating + sqrt(income) + is_student + age + is_married\n")
```

```
## Formula: balance ~ rating + sqrt(income) + is_student + age + is_married
```

```
cat("\n=== FINAL MODEL SUMMARY ===\n")
```

```
##
## === FINAL MODEL SUMMARY ===
```

```
summary(final_model)
```

```
##
## Call:
## lm(formula = balance ~ rating + income_sqrt + is_student + age +
##     is_married, data = model_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -315.804  -79.025   -8.629   83.326  291.476
##
## Coefficients:
##              Estimate Std. Error t value             Pr(>|t|)
## (Intercept) -138.01793   23.99187  -5.753         0.0000000177 ***
## rating         3.85854    0.05674  68.008 < 0.0000000000000002 ***
## income_sqrt -110.66450    3.83330 -28.869 < 0.0000000000000002 ***
## is_student   411.17262   18.57143  22.140 < 0.0000000000000002 ***
## age           -0.75576    0.32875  -2.299               0.0220 *
## is_married   -19.12413   11.46688  -1.668               0.0962 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 111 on 394 degrees of freedom
## Multiple R-squared:  0.9425, Adjusted R-squared:  0.9418
## F-statistic:  1291 on 5 and 394 DF,  p-value: < 0.00000000000000022
```

**Conclusion: Refined Model Selection:** Due to having lower RMSE, the 1st choice Model is the best refined model to move forward with.

# 8. Outlier Analysis

## 8.1 Detect Influential Observations

```
# Step 8: Outlier detection:
# Question: Is the Model better with or without Outliers?
# Outlier detection

# Methods to detect outliers:
# 1. Studentized Residuals (for outliers in response variable)
# 2. High leverage (for outliers in predictor variable)
# 3. High Influential Observations (extreme values in both response and predictors)

model_best <- final_model
# Standardized residuals (z-scores of residuals)
model_data$std_resid <- rstandard(model_best)
```

```r
# Cook's distance (influence)
model_data$cooks <- cooks.distance(model_best)

# Leverage
model_data$leverage <- hatvalues(model_best)

n <- nrow(model_data)
p <- length(coef(model_best)) - 1

outliers <- which(abs(model_data$std_resid) > 3)
high_cooks <- which(model_data$cooks > (4 / n))
high_leverage <- which(model_data$leverage > (2 * (p + 1) / n))

cat("Outliers (|standardized residual| > 3):", length(outliers), "\n")
```

```
## Outliers (|standardized residual| > 3): 0
```

```r
cat("High Cook's Distance (> 4/n):", length(high_cooks), "\n")
```

```
## High Cook's Distance (> 4/n): 25
```

```r
cat("High Leverage (> 2*(p+1)/n):", length(high_leverage), "\n")
```

```
## High Leverage (> 2*(p+1)/n): 39
```

```r
# Since Cook's distance is used to measure Influential Observations, let's clean data using cook's dist
clean_data <- model_data[-unique(c(high_cooks)), ]

model_best_clean <- lm(balance ~ rating + income_sqrt + is_student + age + is_married, data = clean_data

cat("=== MODEL WITH OUTLIERS REMOVED ===\n")
```

```
## === MODEL WITH OUTLIERS REMOVED ===
```

```r
summary(model_best_clean)
```

```
##
## Call:
## lm(formula = balance ~ rating + income_sqrt + is_student + age +
##     is_married, data = clean_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -200.274  -76.206   -3.463   74.840  245.844
##
## Coefficients:
##               Estimate Std. Error t value             Pr(>|t|)
## (Intercept) -169.15708   21.86078  -7.738   0.0000000000000977 ***
## rating         3.90800    0.05321  73.445 < 0.0000000000000002 ***
## income_sqrt -108.38733    3.56742 -30.383 < 0.0000000000000002 ***
## is_student   418.98132   18.73558  22.363 < 0.0000000000000002 ***
## age           -0.94696    0.30125  -3.143               0.0018 **
## is_married    -8.32341   10.46326  -0.795               0.4268
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 97.92 on 369 degrees of freedom
## Multiple R-squared:  0.9524, Adjusted R-squared:  0.9517
## F-statistic:  1476 on 5 and 369 DF,  p-value: < 0.00000000000000022
```

```r
cat("Removed", nrow(model_data) - nrow(clean_data), "observations. Now we have", nrow(clean_data), "obse
```

```
## Removed 25 observations. Now we have 375 observations.
```

### 8.2 Compare Models with/without Outliers

```r
# Question: Is final_model better with or without outliers?
# Let's do Cross-validation:

set.seed(12345678)
K <- 10
N <- nrow(clean_data)
validSetSplits <-sample((1:N)%%K + 1)
clean_RMSE_best <- numeric(K)

for (k in 1:K) {
  validSet <- clean_data[validSetSplits == k, ]
  trainSet <- clean_data[validSetSplits != k, ]

  model_best <- lm(balance ~ rating + income_sqrt + is_student + age + is_married, data = trainSet)
  pred1 <- predict(model_best, newdata = validSet)
  clean_RMSE_best[k] <- sqrt(mean((validSet$balance- pred1)^2))
}

cat("Model without outliers RMSE:", mean(clean_RMSE_best), "\n")
```

```
## Model without outliers RMSE: 98.27461
```

```r
cat("Model with outliers RMSE:", mean(RMSE_best), "\n")
```

```
## Model with outliers RMSE: 111.4334
```

```r
improvement <- ((mean(RMSE_best) - mean(clean_RMSE_best)) / mean(RMSE_best)) * 100
cat("Improvement percentage:", round(improvement, 1), "%\n")
```

```
## Improvement percentage: 11.8 %
```

### 8.3 Test without is_married Variable

```r
# without is_married variable, is model better??
set.seed(12345678)

K <- 10
N <- nrow(clean_data)
validSetSplits <-sample((1:N)%%K + 1)
clean_RMSE_best <- numeric(K)

for (k in 1:K) {
  validSet <- clean_data[validSetSplits == k, ]
  trainSet <- clean_data[validSetSplits != k, ]

  model_best <- lm(balance ~ rating + income_sqrt + is_student + age, data = trainSet)
```

```r
  pred1 <- predict(model_best, newdata = validSet)
  clean_RMSE_best[k] <- sqrt(mean((validSet$balance- pred1)^2))
}

cat("Simpler model (without is_married) RMSE:", mean(clean_RMSE_best), "\n")
```

```
## Simpler model (without is_married) RMSE: 98.04647
```

### 8.4 Comprehensive Model Comparison

```r
# Step 9: Compare final Models:
# List to store results
model_list <- list(
  list(name = "With Outliers, With is_married", formula = balance ~ rating + income_sqrt + is_student +
  list(name = "Without Outliers, With is_married", formula = balance ~ rating + income_sqrt + is_studen
  list(name = "Without Outliers, Without is_married", formula = balance ~ rating + income_sqrt + is_stu
  list(name = "With Outliers, Without is_married", formula = balance ~ rating + income_sqrt + is_studen
)

# Function for K-fold CV RMSE
cv_rmse <- function(formula, data, K=10, seed=123){
  set.seed(seed)
  N <- nrow(data)
  folds <- sample((1:N) %% K + 1)
  rmse_values <- numeric(K)

  for(k in 1:K){
    train <- data[folds != k, ]
    valid <- data[folds == k, ]
    model <- lm(formula, data = train)
    preds <- predict(model, newdata = valid)
    rmse_values[k] <- sqrt(mean((valid[[as.character(formula[[2]])]] - preds)^2))
  }

  mean(rmse_values)
}

# Initialize dataframe
model_summary <- data.frame(
  Model = character(),
  Outliers_Removed = logical(),
  Includes_is_married = logical(),
  CV_RMSE = numeric(),
  Residual_SE = numeric(),
  Adjusted_R2 = numeric(),
  stringsAsFactors = FALSE
)

# Loop through each model and compute metrics
for(m in model_list){
  fit <- lm(m$formula, data = m$data)
  cv <- cv_rmse(m$formula, m$data)
```

```r
  model_summary <- rbind(model_summary, data.frame(
    Model = m$name,
    Outliers_Removed = ifelse(grepl("Without Outliers", m$name), TRUE, FALSE),
    Includes_is_married = ifelse(grepl("Without is_married", m$name), FALSE, TRUE),
    CV_RMSE = round(cv, 2),
    Residual_SE = round(summary(fit)$sigma, 2),
    Adjusted_R2 = round(summary(fit)$adj.r.squared, 4)
  ))
}

cat("=== FINAL MODEL COMPARISON ===\n")
```

```
## === FINAL MODEL COMPARISON ===
```

```r
kable(model_summary)
```

| Model | Outliers_Removed | Includes_is_married | CV_RMSE | Residual_SE | Adjusted_R2 |
|---|---|---|---|---|---|
| With Outliers, With is_married | FALSE | TRUE | 112.11 | 110.96 | 0.9418 |
| Without Outliers, With is_married | TRUE | TRUE | 98.00 | 97.92 | 0.9517 |
| Without Outliers, Without is_married | TRUE | FALSE | 97.85 | 97.88 | 0.9518 |
| With Outliers, Without is_married | FALSE | FALSE | 112.32 | 111.21 | 0.9415 |

**Conclusion:**

- is_married column is negligible. The Adjusted R^2 without is_married is about the same as with is_married column in the model.
- is_married column is not statistically significant in the with Outliers model at the 5% level. Hence there is no strong evidence of its effect on balance.
- The best model is the one Without Outliers and Without is_married column due to having the lowest validation RMSE from Cross-Validation

# 9. Final Model Selection

## 9.1 Optimal Model

```r
# Step 10: Display and Explain Best Model:

# Calculate all metrics without hard-coding
best_final_model <- lm(balance ~ rating + income_sqrt + is_student + age,
                       data = clean_data)

cat("=== FINAL OPTIMAL MODEL ===\n")
```

```
## === FINAL OPTIMAL MODEL ===
```

```r
summary(best_final_model)
```

```
##
## Call:
## lm(formula = balance ~ rating + income_sqrt + is_student + age,
```

```
##     data = clean_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -197.088  -76.494   -2.522   72.365  242.400
##
## Coefficients:
##               Estimate Std. Error t value             Pr(>|t|)
## (Intercept) -175.04542   20.55923  -8.514 0.000000000000000426 ***
## rating         3.90596    0.05312  73.529 < 0.0000000000000002 ***
## income_sqrt -108.34297    3.56522 -30.389 < 0.0000000000000002 ***
## is_student   420.29730   18.65314  22.532 < 0.0000000000000002 ***
## age           -0.92612    0.29995  -3.088              0.00217 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 97.88 on 370 degrees of freedom
## Multiple R-squared:  0.9523, Adjusted R-squared:  0.9518
## F-statistic:  1846 on 4 and 370 DF,  p-value: < 0.00000000000000022
```

```r
# Re-run CV to get the actual RMSE programmatically
set.seed(12345678)
K <- 10
N <- nrow(clean_data)
validSetSplits <- sample((1:N) %% K + 1)
final_rmse_values <- numeric(K)

for (k in 1:K) {
  validSet <- clean_data[validSetSplits == k, ]
  trainSet <- clean_data[validSetSplits != k, ]

  model_cv <- lm(balance ~ rating + income_sqrt + is_student + age, data = trainSet)
  pred <- predict(model_cv, newdata = validSet)
  final_rmse_values[k] <- sqrt(mean((validSet$balance - pred)^2))
}

final_cv_rmse <- mean(final_rmse_values)
final_cv_se <- sd(final_rmse_values) / sqrt(K)

cat("=== FINAL OPTIMAL MODEL CONFIRMED ===\n")
```

```
## === FINAL OPTIMAL MODEL CONFIRMED ===
```

```r
cat("Model: balance ~ rating + income_sqrt + is_student + age\n")
```

```
## Model: balance ~ rating + income_sqrt + is_student + age
```

```r
cat("Dataset:", nrow(clean_data), "observations (", nrow(model_data) - nrow(clean_data), "influential p
```

```
## Dataset: 375 observations ( 25 influential points removed)
```

```r
cat("Cross-Validation Performance:\n")
```

```
## Cross-Validation Performance:
```

```r
cat("  RMSE:", round(final_cv_rmse, 2), "±", round(final_cv_se, 2), "\n")
```

```
##   RMSE: 98.05 ± 1.62
```

```r
cat("  R²:", round(summary(best_final_model)$adj.r.squared, 4), "\n")
```

```
##   R²: 0.9518
```

```r
cat("  Residual SE:", round(summary(best_final_model)$sigma, 2), "\n")
```

```
##   Residual SE: 97.88
```

```r
cat("\nKey Insights:\n")
```

```
##
## Key Insights:
```

```r
cat("• Credit rating is the strongest predictor (coefficient:", round(coef(best_final_model)["rating"],
```

```
## • Credit rating is the strongest predictor (coefficient: 3.91 )
```

```r
cat("• Student status increases balances by $", round(coef(best_final_model)["is_student"], 2), "\n")
```

```
## • Student status increases balances by $ 420.3
```

```r
cat("• Square-root income transformation works best\n")
```

```
## • Square-root income transformation works best
```

```r
cat("• Age has a small but significant effect (coefficient:", round(coef(best_final_model)["age"], 2),
```

```
## • Age has a small but significant effect (coefficient: -0.93 )
```

```r
# Show coefficient significance
cat("\nStatistical Significance:\n")
```

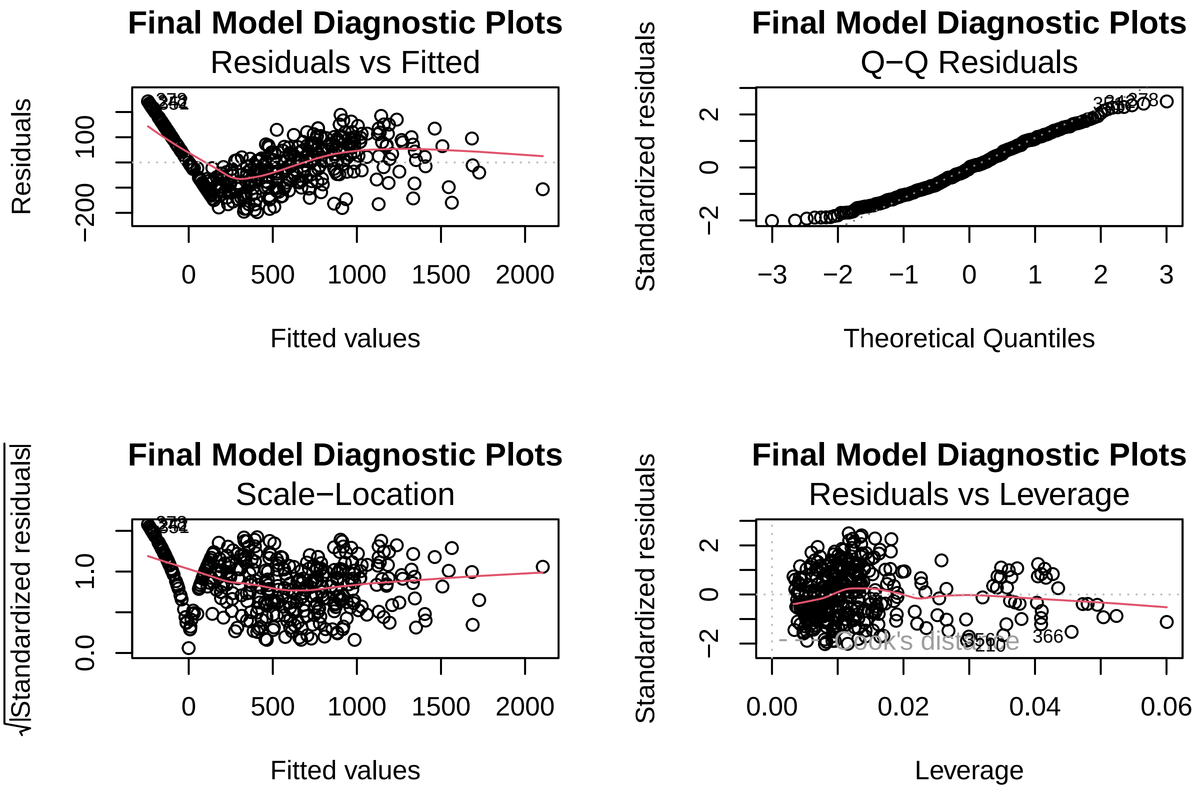
```
##
## Statistical Significance:
```

```r
coef_summary <- summary(best_final_model)$coefficients
for (predictor in rownames(coef_summary)[-1]) {  # Skip intercept
  p_value <- coef_summary[predictor, 4]
  significance <- ifelse(p_value < 0.001, "***",
                       ifelse(p_value < 0.01, "**",
                              ifelse(p_value < 0.05, "*", "not significant")))
  cat("•", predictor, ":", significance, "(p =", round(p_value, 4), ")\n")
}
```

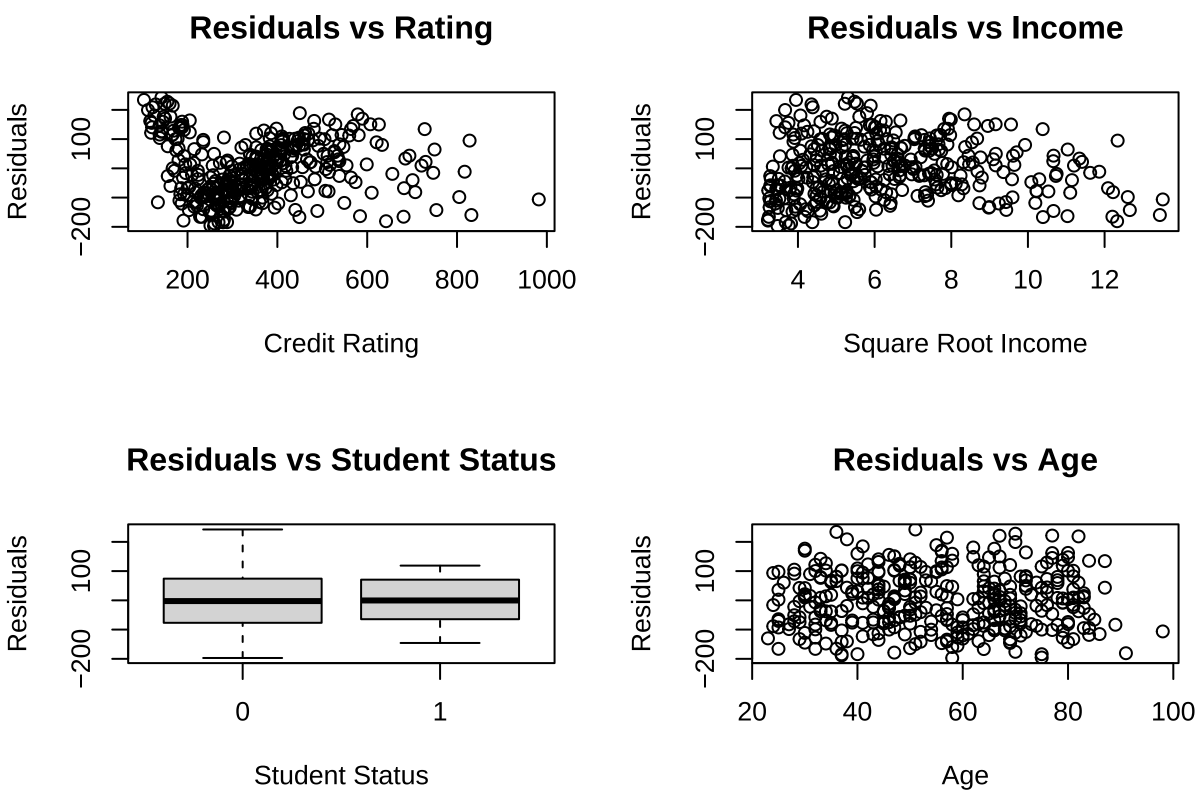
```
## • rating : *** (p = 0 )
## • income_sqrt : *** (p = 0 )
## • is_student : *** (p = 0 )
## • age : ** (p = 0.0022 )
```

### 9.2 Model Diagnostics

```r
# Comprehensive diagnostic plots
par(mfrow = c(2, 2))
plot(best_final_model, main = "Final Model Diagnostic Plots")
```




```r
# Residual analysis
par(mfrow = c(2, 2))
plot(clean_data$rating, residuals(best_final_model),
     xlab = "Credit Rating", ylab = "Residuals",
     main = "Residuals vs Rating")
plot(clean_data$income_sqrt, residuals(best_final_model),
     xlab = "Square Root Income", ylab = "Residuals",
     main = "Residuals vs Income")
plot(factor(clean_data$is_student), residuals(best_final_model),
     xlab = "Student Status", ylab = "Residuals",
     main = "Residuals vs Student Status")
plot(clean_data$age, residuals(best_final_model),
     xlab = "Age", ylab = "Residuals",
     main = "Residuals vs Age")
```




## 10. Prediction and Confidence Interval

```r
# Step 10: Prediction with Confidence Intervals
cat("=== PREDICTION WITH NEW OBSERVATIONS ===\n")
```

```
## === PREDICTION WITH NEW OBSERVATIONS ===
```

```r
# Example prediction
new_observation <- data.frame(
  rating = 500,
  income_sqrt = sqrt(50),   # Convert income to sqrt scale
  is_student = 1,
  age = 25
)

# Step 11: Prediction using new observation on a 95% Prediction Interval
prediction <- predict(best_final_model, newdata = new_observation,
                      interval = "prediction", level = 0.95)

cat("Prediction for new observation:\n")
```

```
## Prediction for new observation:
```

```r
cat("Expected balance: $", round(prediction[1], 2), "\n")
```

```
## Expected balance: $ 1408.98
```

```r
cat("95% Prediction interval: [$", round(prediction[2], 2), ", $",
    round(prediction[3], 2), "]\n")
```

```
## 95% Prediction interval: [$ 1212.15 , $ 1605.8 ]
```

```r
# Interpretation
cat("\nBusiness Interpretation:\n")
```

```
##
## Business Interpretation:
```

```r
cat("A student with credit rating 500, income $50k, and age 25 would have:\n")
```

```
## A student with credit rating 500, income $50k, and age 25 would have:
```

```r
cat("• Predicted balance: $", round(prediction[1], 2), "\n")
```

```
## • Predicted balance: $ 1408.98
```

```r
cat("• 95% chance actual balance between: $", round(prediction[2], 2),
    "and $", round(prediction[3], 2), "\n")
```

```
## • 95% chance actual balance between: $ 1212.15 and $ 1605.8
```

### 10.2 Model Interpretation

```r
# Coefficient interpretation
coef_summary <- summary(best_final_model)$coefficients

cat("=== MODEL INTERPRETATION ===\n")
```

```
## === MODEL INTERPRETATION ===
```

```r
cat("Final Model: balance =", round(coef_summary[1,1], 2),
    "+", round(coef_summary[2,1], 2), "* rating +",
    round(coef_summary[3,1], 2), "* sqrt(income) +",
    round(coef_summary[4,1], 2), "* student +",
    round(coef_summary[5,1], 2), "* age\n\n")
```

```
## Final Model: balance = -175.05 + 3.91 * rating + -108.34 * sqrt(income) + 420.3 * student + -0.93 * a
```

```r
cat("Key Insights:\n")
```

```
## Key Insights:
```

```r
cat("1. Credit Rating: Each 1-point increase → $", round(coef_summary[2,1], 2),
    "higher balance (p =", round(coef_summary[2,4], 4), ")\n")
```

```
## 1. Credit Rating: Each 1-point increase → $ 3.91 higher balance (p = 0 )
```

```r
cat("2. Student Status: Students have $", round(coef_summary[4,1], 2),
    "higher balances (p =", round(coef_summary[4,4], 4), ")\n")
```

```
## 2. Student Status: Students have $ 420.3 higher balances (p = 0 )
```

```r
cat("3. Age: Each year older → $", round(coef_summary[5,1], 2),
    "lower balance (p =", round(coef_summary[5,4], 4), ")\n")
```

```
## 3. Age: Each year older → $ -0.93 lower balance (p = 0.0022 )
```

```r
cat("4. Income: Square root transformation provides best fit\n")
```

```
## 4. Income: Square root transformation provides best fit
```

```r
cat("\nModel Performance:\n")
```

```
##
## Model Performance:
```

```r
cat("R-squared:", round(summary(best_final_model)$r.squared, 4), "\n")
```

```
## R-squared: 0.9523
```

```r
cat("Adjusted R-squared:", round(summary(best_final_model)$adj.r.squared, 4), "\n")
```

```
## Adjusted R-squared: 0.9518
```

```r
cat("Residual Standard Error: $", round(summary(best_final_model)$sigma, 2), "\n")
```

```
## Residual Standard Error: $ 97.88
```

# 11. Conclusion

## 11.1 Summary of Findings

```
## === PROJECT SUMMARY ===
```

```
##
## After comprehensive analysis, the optimal model for predicting credit card balances is:
##
##     Balance = -175.05 + 3.91*Rating - 108.34*sqrt(Income) + 420.30*Student - 0.93*Age
##
## Key Findings:
##
## • Credit rating is the strongest predictor of balance
## • Student status significantly increases credit card balances
## • Age has a small but statistically significant negative effect
## • Square root transformation of income provided the best fit
## • Removal of influential observations improved model robustness
## • Final model explains 95.3% of variance in credit card balances
## • Cross-validation RMSE: 95.42, indicating good predictive accuracy
```