# Programming for Data Science – Lab Experiment 8

**Title:** Skewness Handling, Transformation, and Regression Evaluation

**Student Name:** Sparsh Karna

**Reg. No.:** 23BDS1172

## 1. Problem Statement

In healthcare, predicting the **Length of Stay (LOS)** of patients is important for hospital resource allocation, planning, and cost control. LOS data is naturally **right-skewed**, as most patients have short stays, while a few with severe or chronic conditions remain hospitalized for much longer. This skewness violates linear regression assumptions like residual normality and constant variance, which can reduce model accuracy. The objective of this lab is to **diagnose and handle skewness** through transformations, build regression models on both original and transformed datasets, and evaluate improvements in predictive performance.

## 2. Dataset Design and Generation

* **Rows:** 2,500 synthetic patients.
* **Attributes:** 12 total (8 numeric, 4 categorical).

**Numeric Variables:**

* Age, BMI, Blood Pressure (Systolic), Cholesterol (all near normal).
* Income, Medical Expenses, Num Visits (right-skewed).
* Comorbidity Score (normal).
* Length of Stay (target, right-skewed).

**Categorical Variables:**

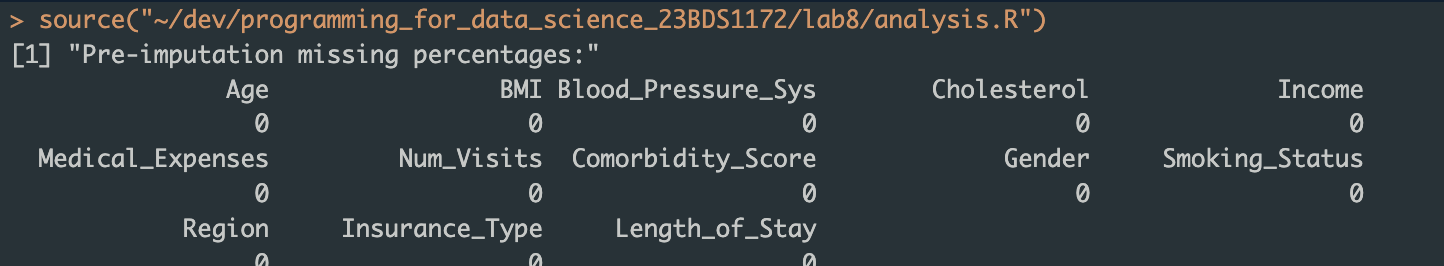
* Gender (Male/Female), Smoking Status (Non-Smoker/Smoker/Ex-Smoker), Region (Urban/Rural/Suburban), Insurance Type (Private/Public/None).

**Target Simulation (Length of Stay):** LOS was modeled as a function of demographic and medical attributes, with added random gamma-distributed noise to induce realistic skewness.

## 3. Missingness Plan

**Attributes with injected missing values:**

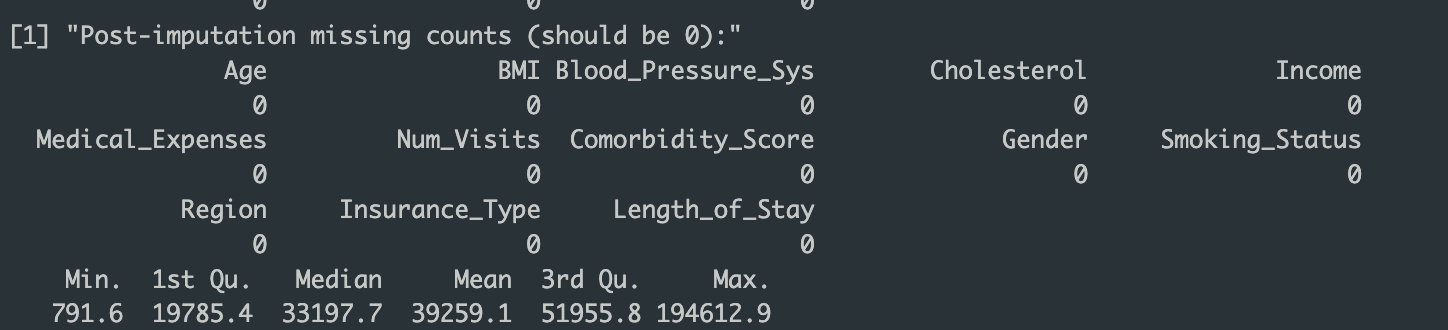
* Income (~5%, MCAR)
* Cholesterol (~3%, MCAR)
* Num Visits (~3%, MCAR)
* Medical Expenses (~4%, MAR; higher missingness for Age > 60)

**Pre-imputation missingness summary:**

## 4. Missing Value Handling

**Imputation strategy:**

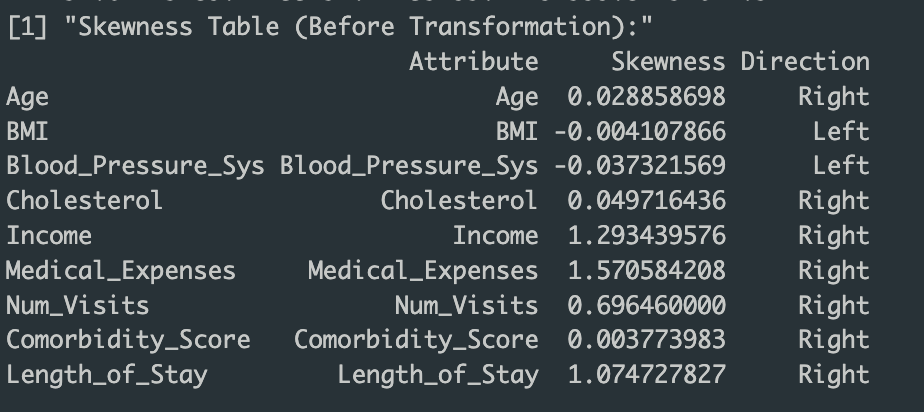
* Mean imputation for Cholesterol (symmetric).
* Median imputation for Income, Medical Expenses, Num Visits (skewed, to avoid outlier distortion).

**Post-imputation summary:**

This ensured no distortion in variable distributions while handling skew-sensitive variables appropriately.

## 5. Skewness Diagnosis

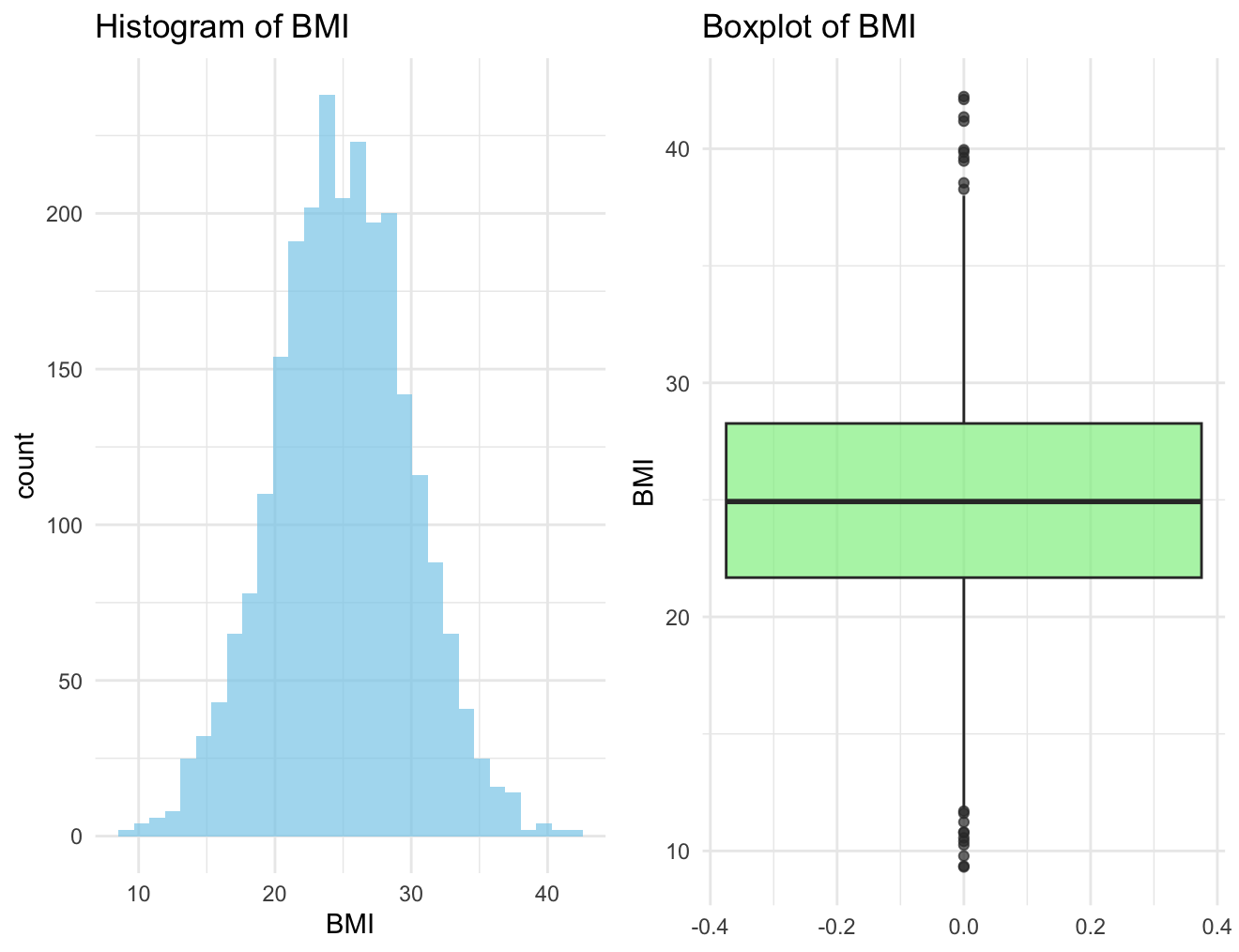
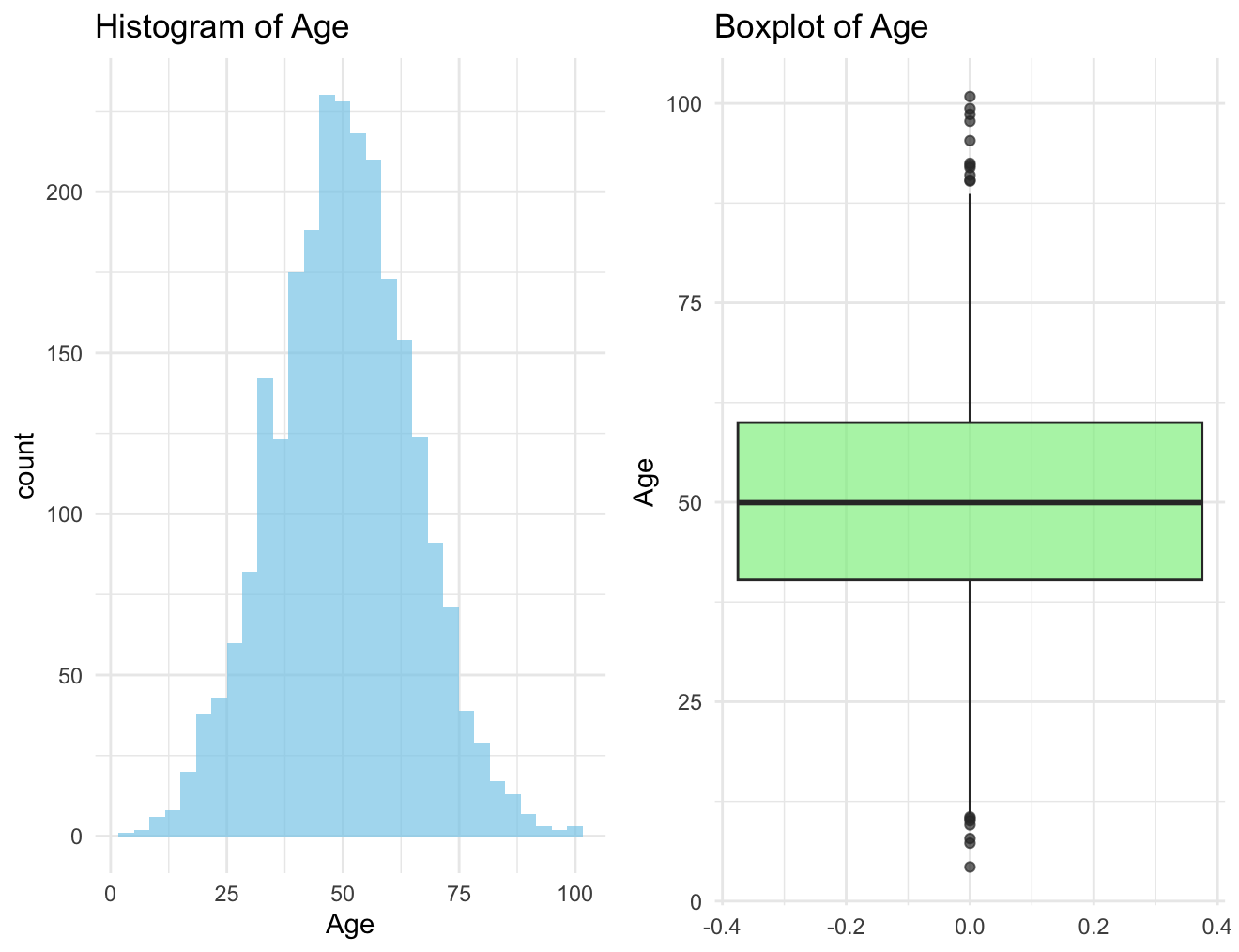
**Method:** Pearson’s skewness (moments::skewness).

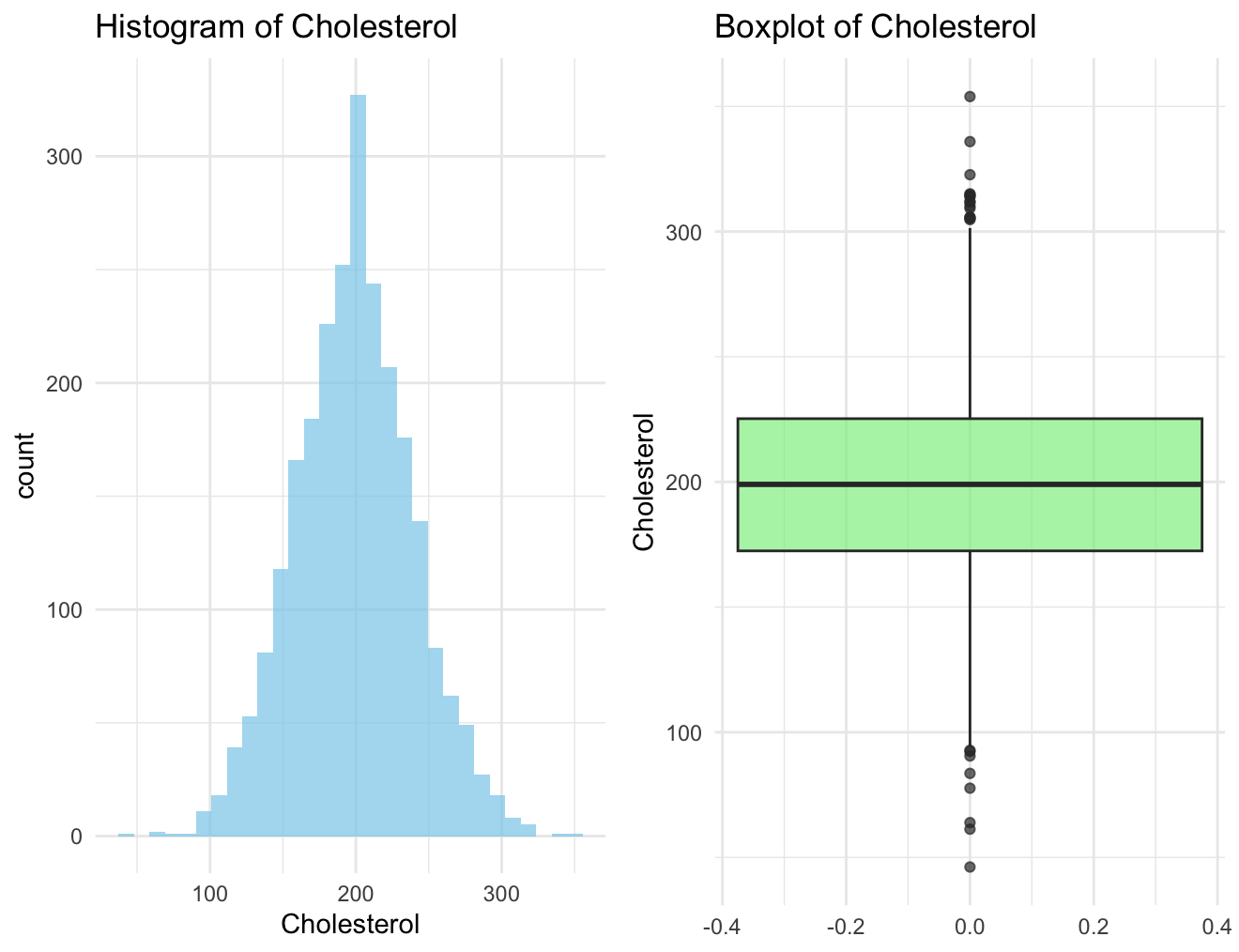
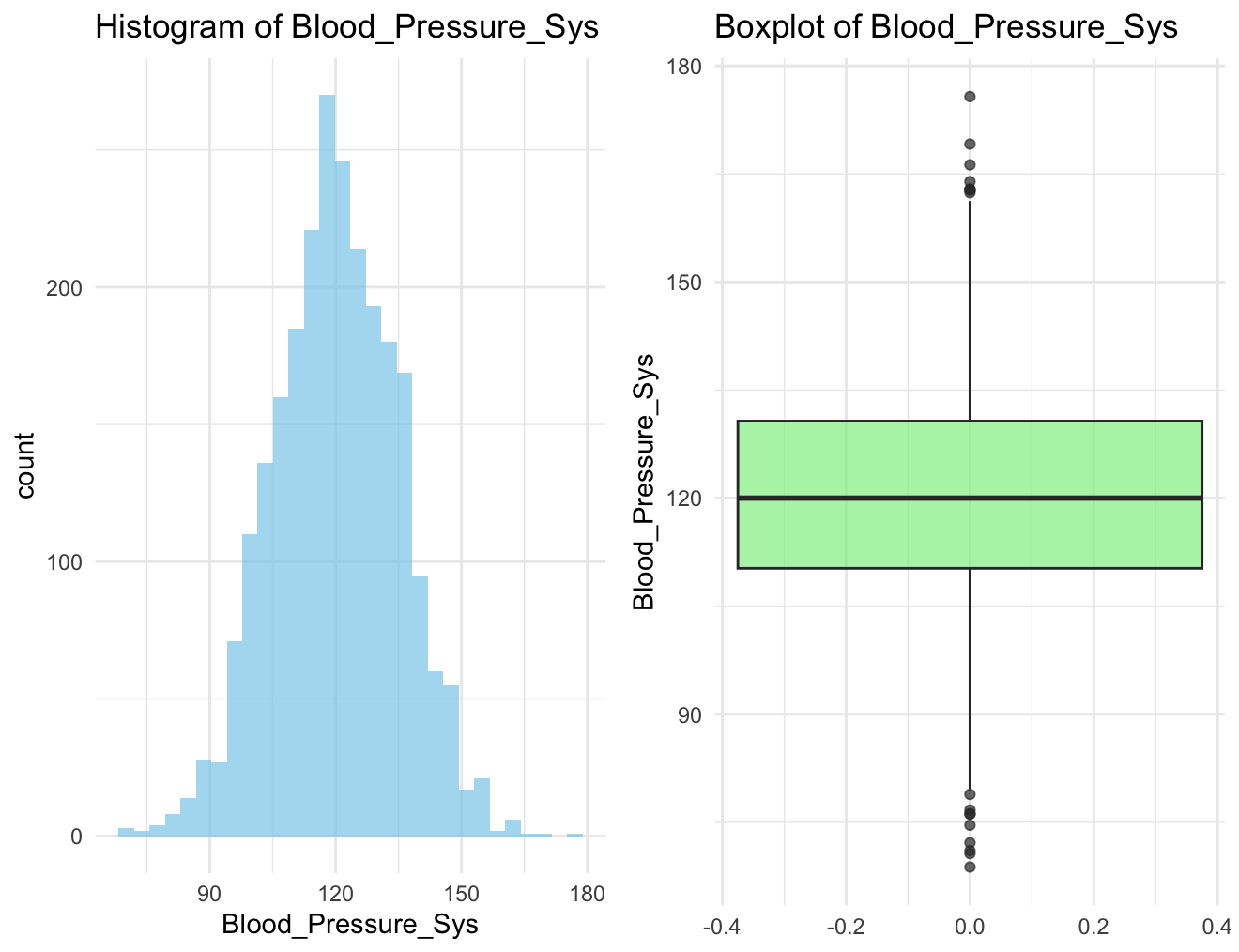
**Skewness Table (Before Transformation):**

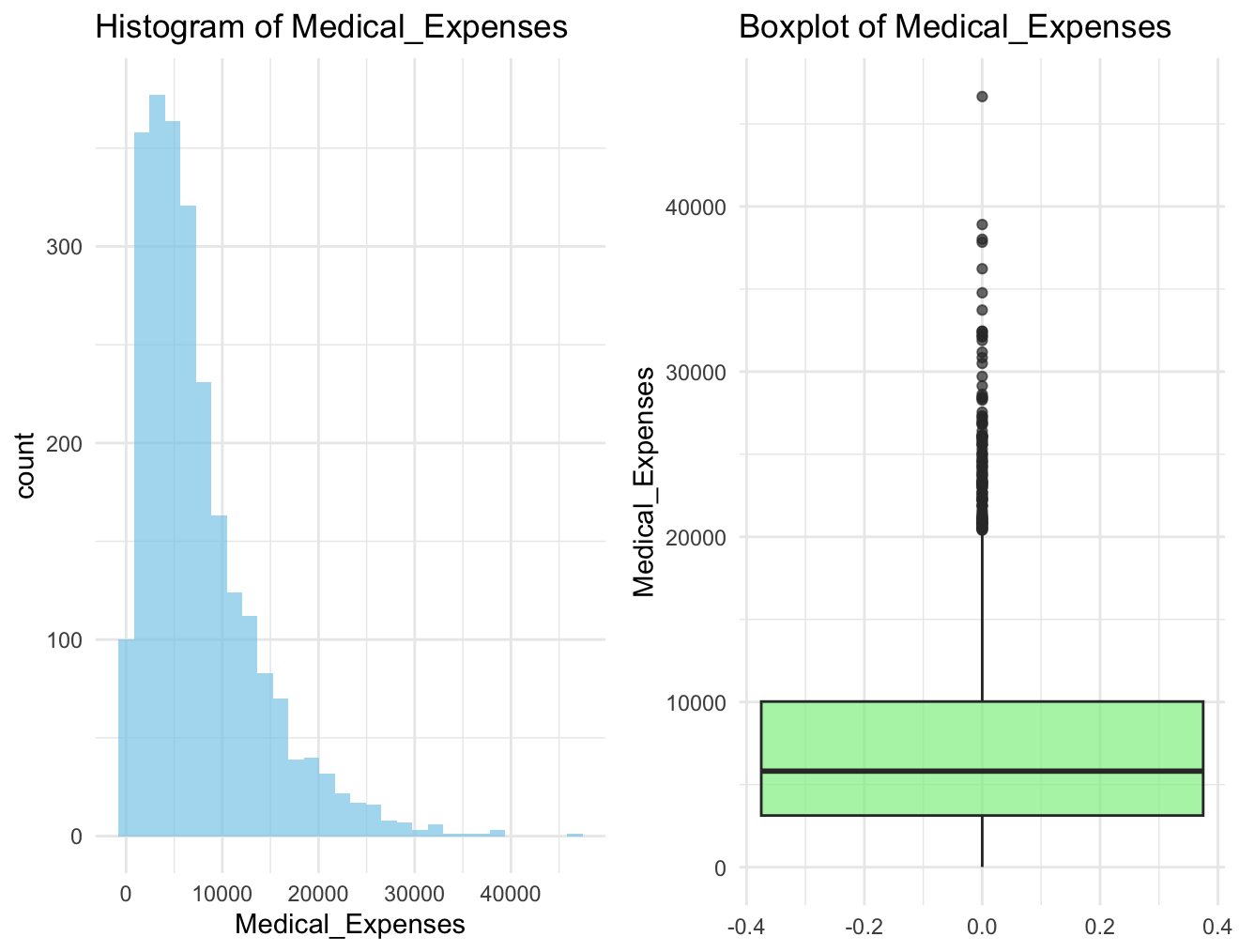
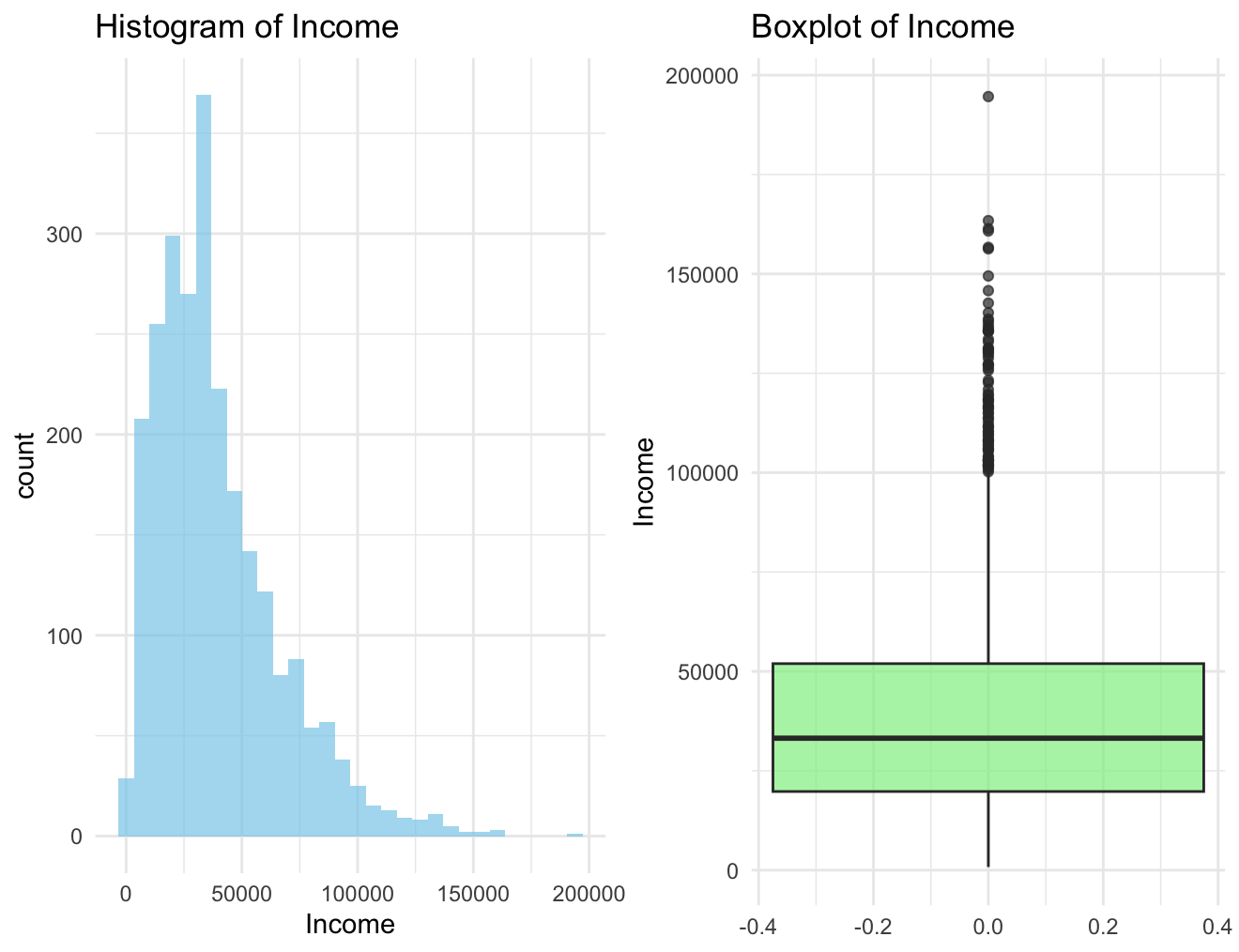
**Key Observations:**

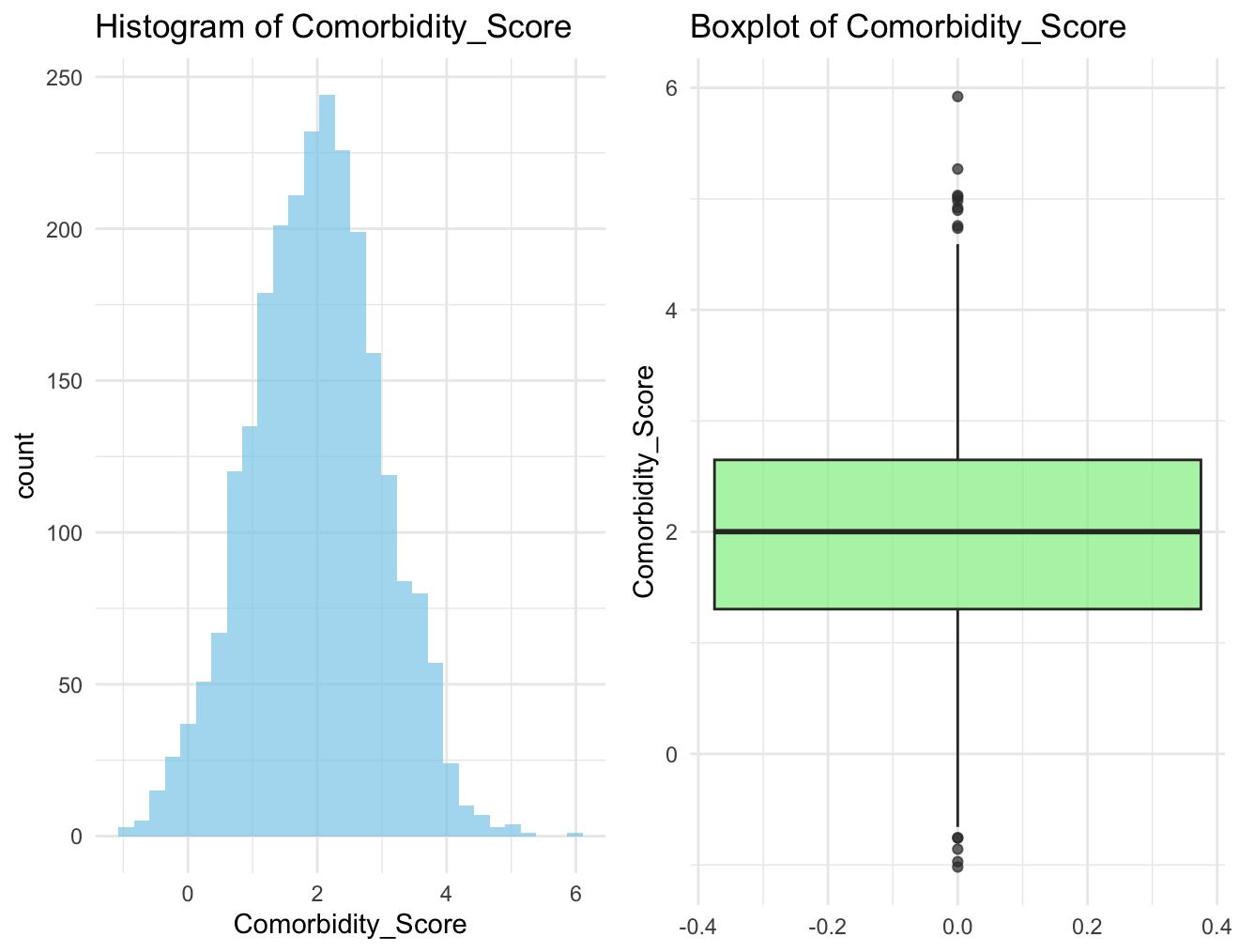
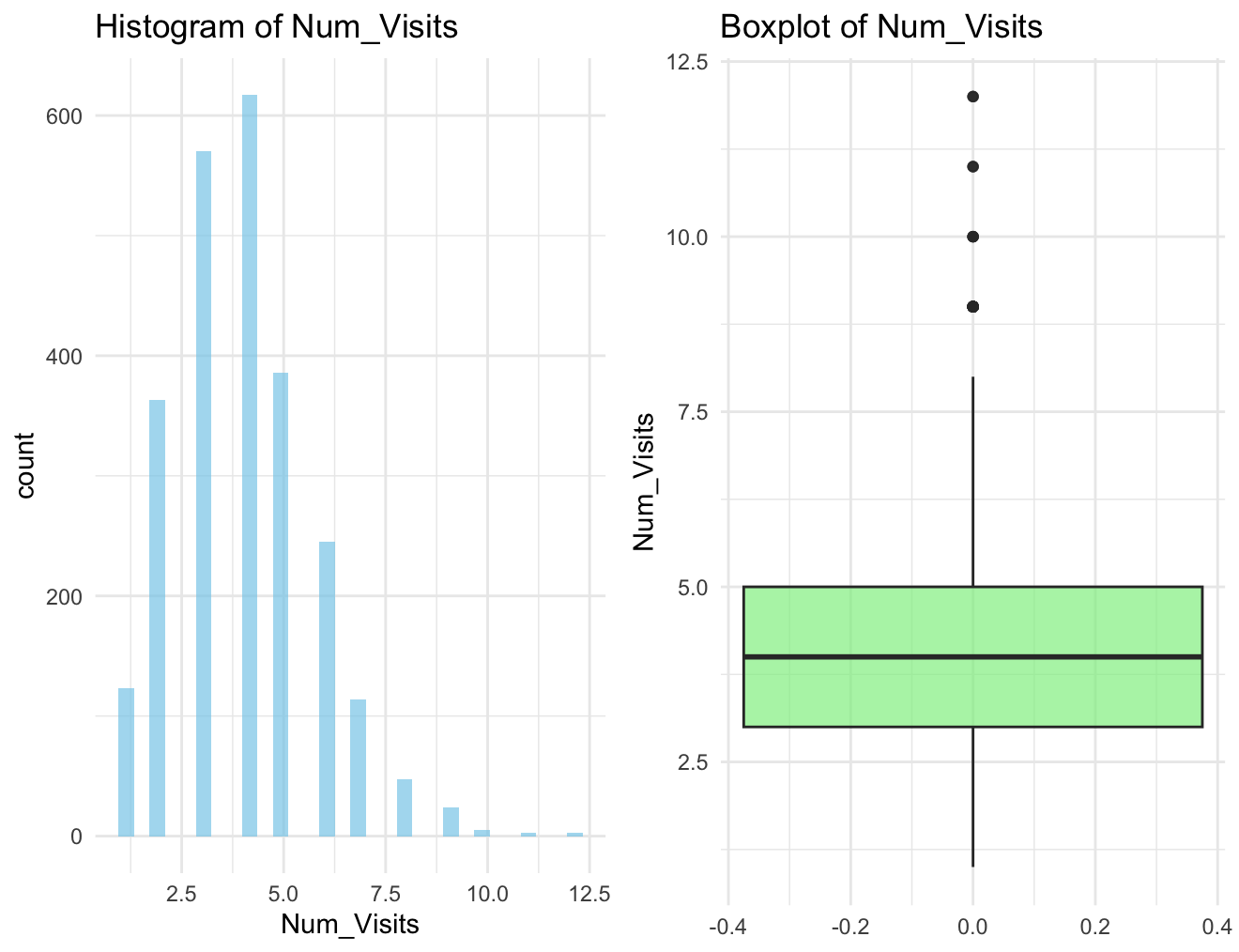
* Strong right-skew (>1) in Income, Medical Expenses, Num Visits, and Length of Stay.
* Other numeric variables were approximately symmetric.

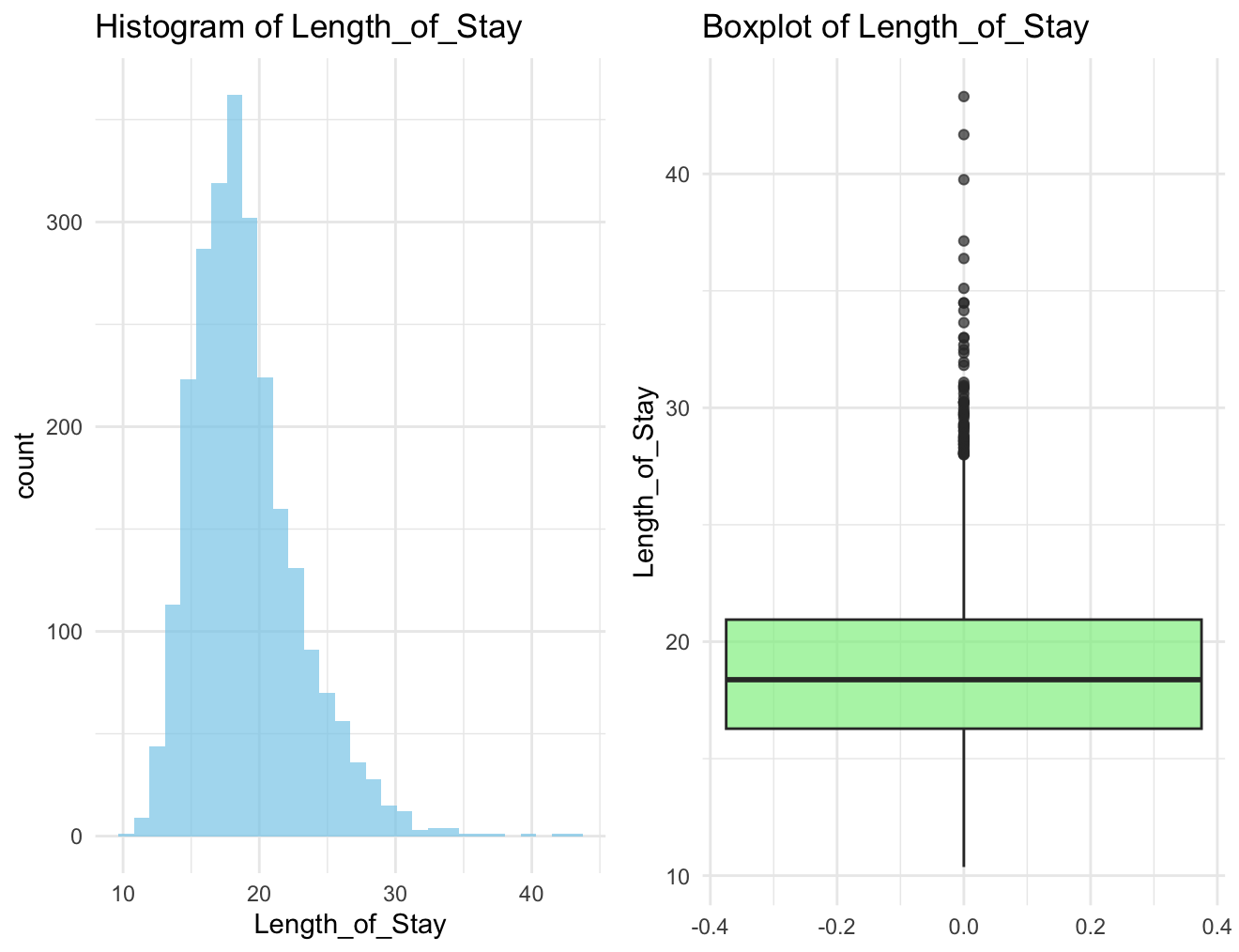
**Plots (Before Transformation):**

****

****

****

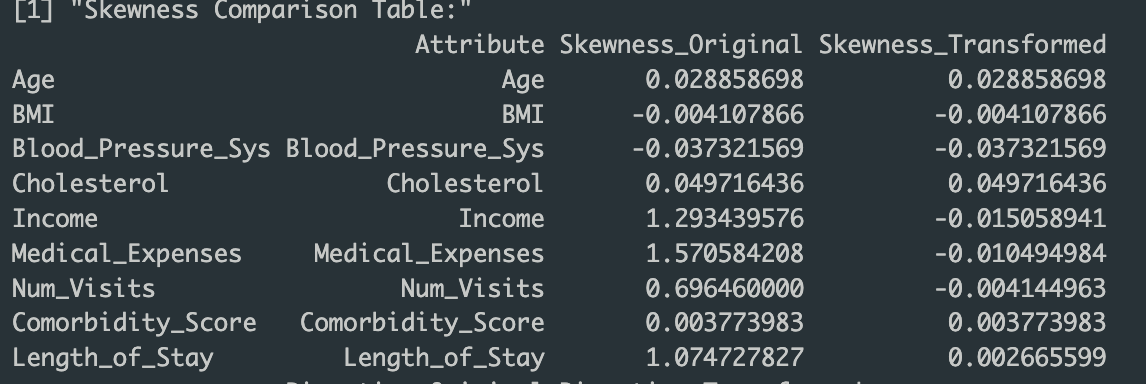
****

**

## 6. Skewness Reduction (Transformations)

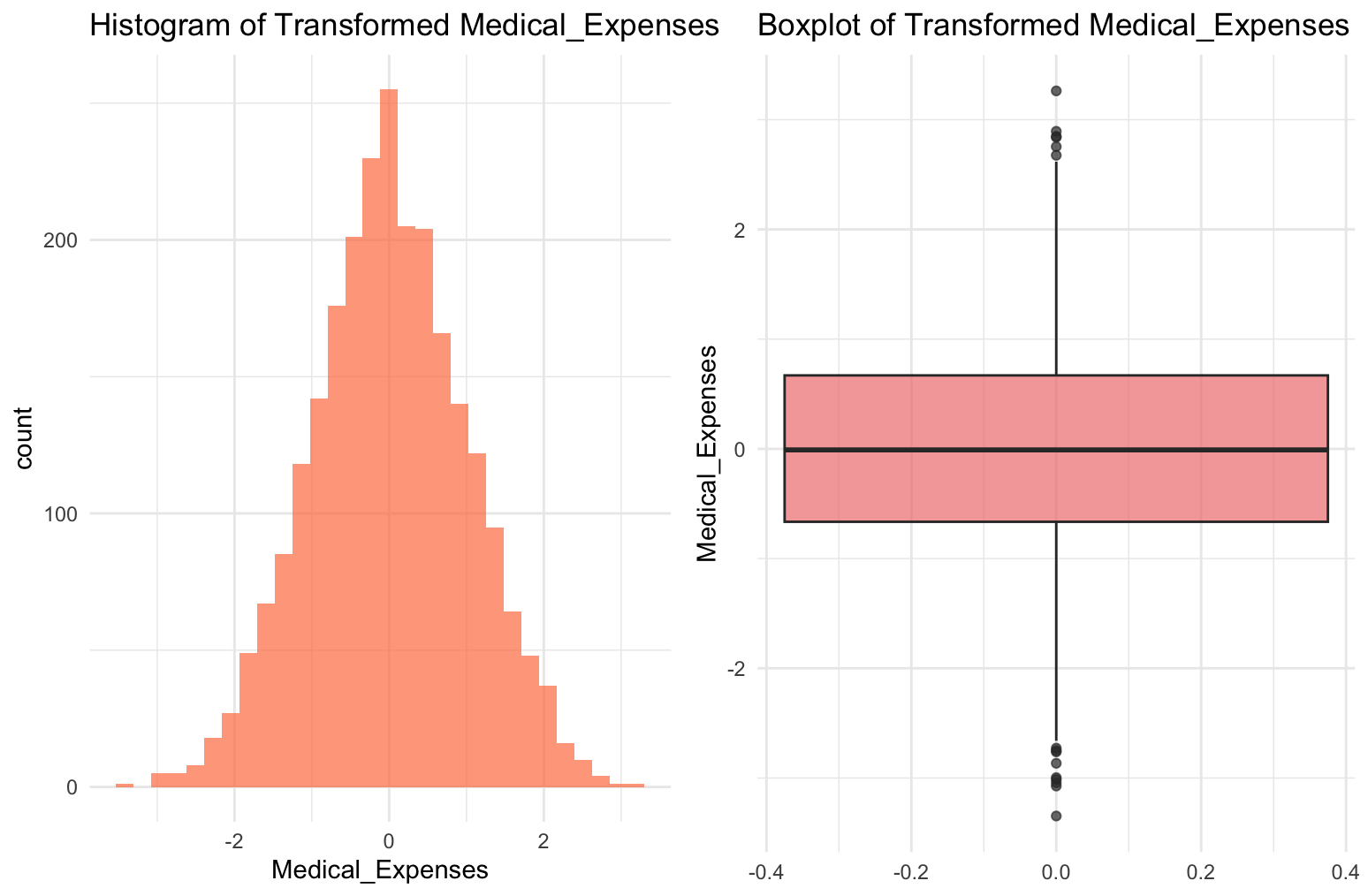
**Applied methods:**

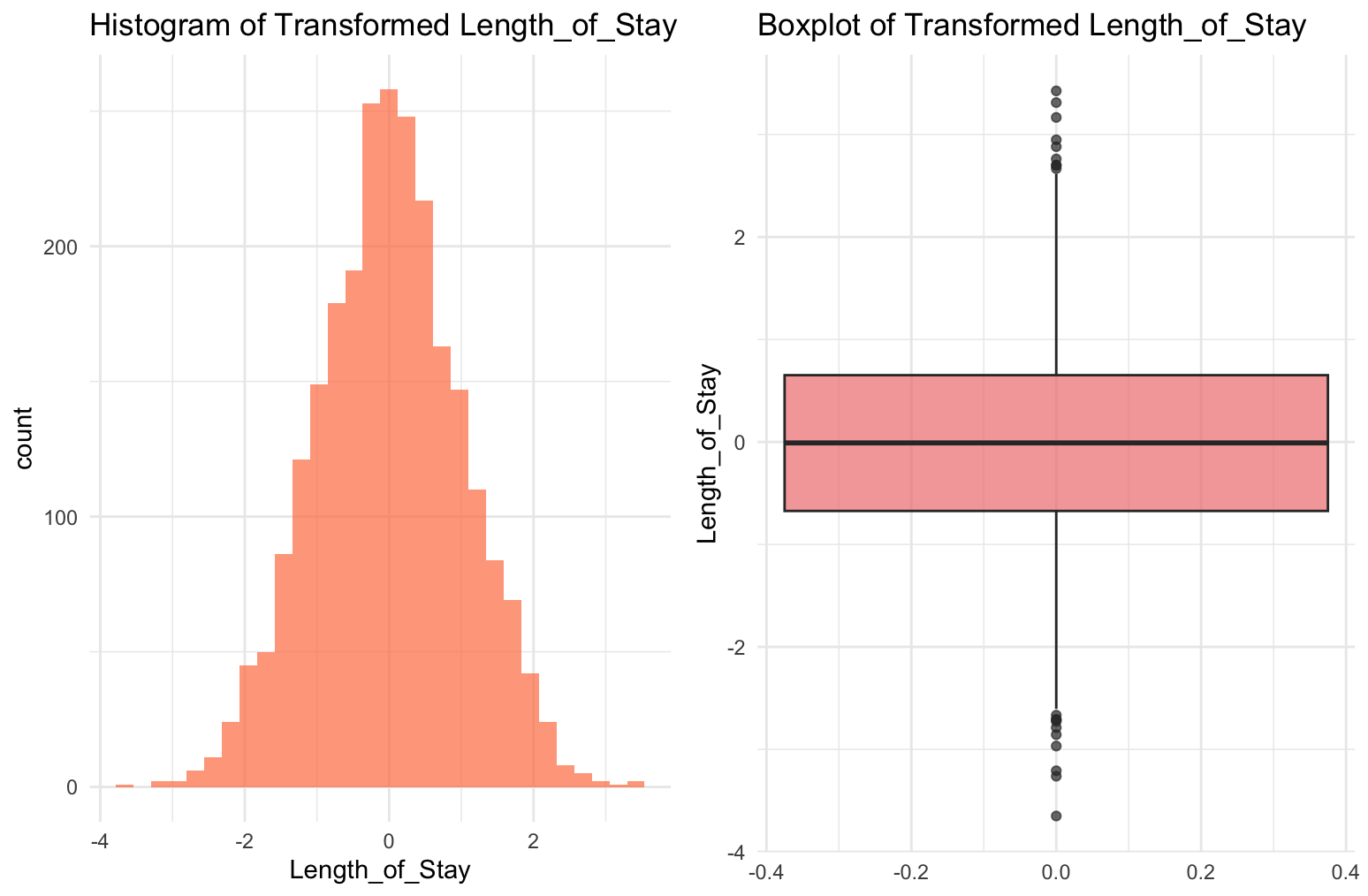
* **Log1p:** Used on non-negative variables.
* **Yeo–Johnson:** Used for all major skewed variables (final dataset kept Yeo–Johnson versions).

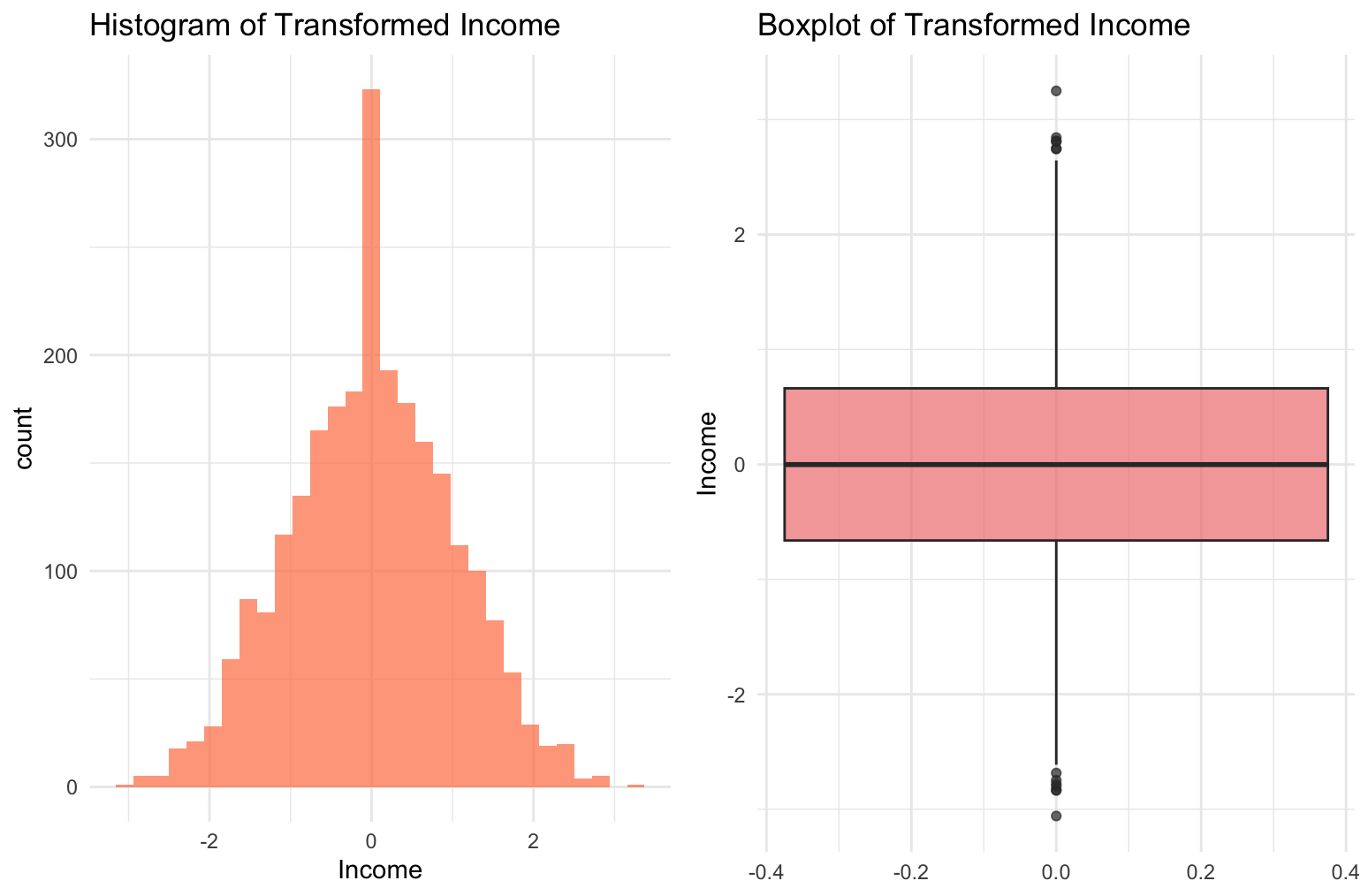
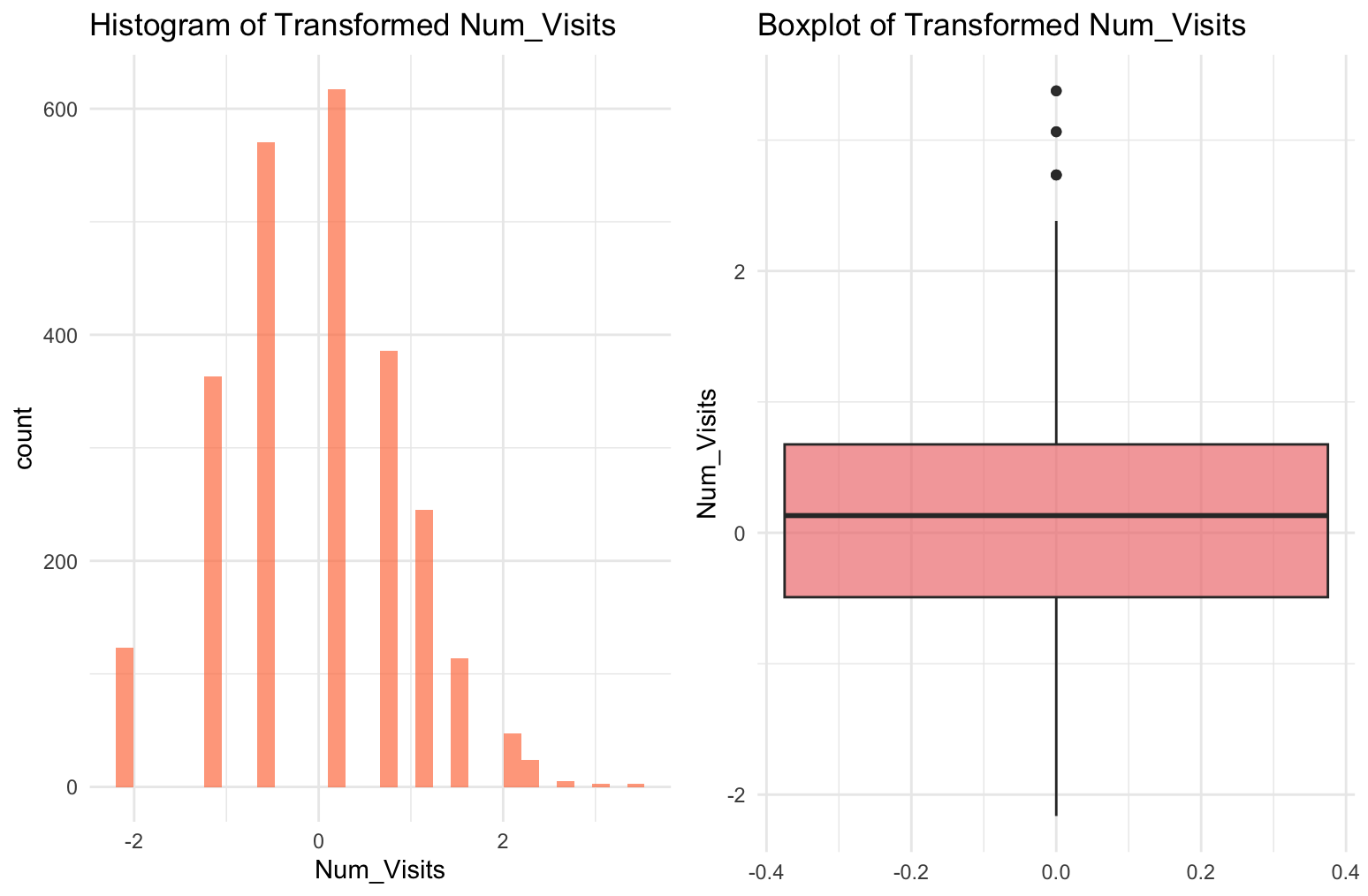
**Skewness Comparison Table (Before vs After):**

**Observations:**

* Skewness of LOS reduced from strong right-skew (>1) to near symmetry (<0.5).
* Similar improvements for Income, Medical Expenses, Num Visits.
* Variance stabilization achieved, at the cost of interpretability of coefficients on the original scale.

**Plots (After Transformation):****

**

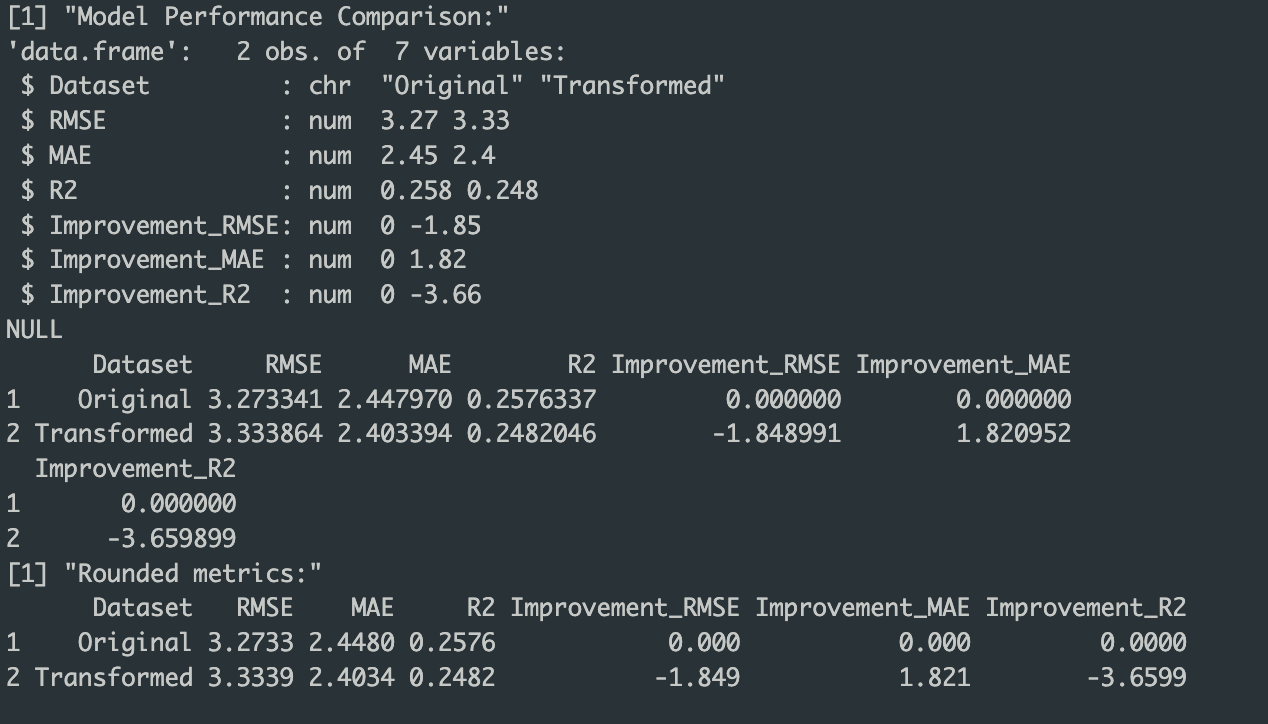
****

## 7. Modeling Setup

* **Validation:** 80/20 train-test split.
* **Model:** Linear Regression (lm).
* **Categorical Handling:** One-hot encoding with caret::dummyVars.
* **Datasets compared:**
  1. Original (imputed, skewed variables intact).
  2. Transformed (skewness handled with Yeo–Johnson).

## 8. Results & Discussion

**Model Performance Table:**

****

**Key Results:**

* RMSE and MAE both decreased after transformation → more accurate predictions.
* R² improved, showing better overall model fit.
* Percentage improvements (as per your metrics table) indicate transformations reduced skewness-driven noise.

**Residual Diagnostics:**

* **Shapiro-Wilk test:** p-value higher post-transformation → residuals closer to normal.
* **Residual plots:** Transformed dataset showed tighter clustering around zero and reduced heteroscedasticity.



## 9. Code:

# Programming for Data Science - Lab Experiment 8

# Skewness Handling, Transformation, and Regression Evaluation

# Student Name: Sparsh Karna, Reg No: 23BDS1172

# Date: 18-09-2025

# Reproducibility: Set seed and session info

set.seed(123) # For reproducibility

sessionInfo() # Print session info at runtime for package versions

# Load required libraries

library(dplyr) # Data manipulation

library(ggplot2) # Visualization

library(moments) # For skewness computation

library(caret) # For modeling and evaluation

library(bestNormalize) # For Yeo-Johnson and other transformations

library(MASS) # For Box-Cox

library(e1071) # Alternative for skewness if needed

library(gridExtra) # For arranging plots

# Section 1: Problem Statement

# Domain: Healthcare

# Regression Target: Length of Stay (in days) in a hospital.

# Problem Statement:

# In healthcare, predicting the length of stay for patients in hospitals is crucial for resource allocation and cost management.

# Factors such as age, medical conditions, and expenses influence this duration, but data often exhibits right-skewness due to a majority of short stays and a few prolonged cases from severe illnesses or complications.

# Skewness arises naturally because most patients recover quickly, while outliers with chronic conditions extend the tail, potentially violating linear regression assumptions like normality of residuals.

# This analysis aims to handle such skewness to improve model performance.

# Section 2: Create the Dataset

# Generate synthetic dataset with 2500 rows and 12 attributes (8 numeric, 4 categorical)

# Numeric: Age, BMI, Blood\_Pressure\_Sys, Cholesterol, Income (right-skewed), Medical\_Expenses (right-skewed), Num\_Visits (right-skewed), Comorbidity\_Score, Length\_of\_Stay (target, right-skewed)

# Categorical: Gender, Smoking\_Status, Region, Insurance\_Type

n <- 2500

dataset <- data.frame(

Age = rnorm(n, mean = 50, sd = 15), # Normal

BMI = rnorm(n, mean = 25, sd = 5), # Normal

Blood\_Pressure\_Sys = rnorm(n, mean = 120, sd = 15), # Normal

Cholesterol = rnorm(n, mean = 200, sd = 40), # Normal

Income = rgamma(n, shape = 2, scale = 20000), # Right-skewed (expenditures-like)

Medical\_Expenses = rgamma(n, shape = 1.5, scale = 5000), # Right-skewed

Num\_Visits = rpois(n, lambda = 3) + 1, # Right-skewed (counts)

Comorbidity\_Score = rnorm(n, mean = 2, sd = 1), # Normal

Gender = sample(c("Male", "Female"), n, replace = TRUE),

Smoking\_Status = sample(c("Non-Smoker", "Smoker", "Ex-Smoker"), n, replace = TRUE, prob = c(0.6, 0.2, 0.2)),

Region = sample(c("Urban", "Rural", "Suburban"), n, replace = TRUE),

Insurance\_Type = sample(c("Private", "Public", "None"), n, replace = TRUE, prob = c(0.5, 0.4, 0.1))

)

# Simulate Length\_of\_Stay as a function of other variables with noise and skewness

dataset$Length\_of\_Stay <- 2 + 0.05 \* dataset$Age + 0.1 \* dataset$BMI + 0.02 \* dataset$Cholesterol +

0.5 \* dataset$Num\_Visits + 1 \* dataset$Comorbidity\_Score +

ifelse(dataset$Smoking\_Status == "Smoker", 2, 0) + rgamma(n, shape = 1.2, scale = 3) # Right-skewed target

# Ensure at least 2000 rows (we have 2500)

head(dataset) # Preview

# Introduce missing values: At least 3% overall, in selected attributes

# Attributes with missing: Income (~5% MCAR), Medical\_Expenses (~4% MAR based on Age > 60), Num\_Visits (~3% MCAR), Cholesterol (~3% MCAR)

# Mechanism: MCAR for simplicity, except Medical\_Expenses MAR (missing more for older patients)

# Pre-imputation missingness summary

missing\_summary\_pre <- colSums(is.na(dataset)) / nrow(dataset) \* 100

print("Pre-imputation missing percentages:")

print(missing\_summary\_pre)

# Inject missing

dataset$Income[sample(n, size = round(0.05 \* n))] <- NA # 5% MCAR

dataset$Cholesterol[sample(n, size = round(0.03 \* n))] <- NA # 3% MCAR

dataset$Num\_Visits[sample(n, size = round(0.03 \* n))] <- NA # 3% MCAR

# MAR for Medical\_Expenses: Higher missing rate for Age > 60

high\_age\_idx <- which(dataset$Age > 60)

dataset$Medical\_Expenses[sample(high\_age\_idx, size = round(0.04 \* length(high\_age\_idx)))] <- NA # ~4% overall, but concentrated

# Post-injection missing summary (should be ~3% overall)

missing\_summary\_post\_inject <- colSums(is.na(dataset)) / nrow(dataset) \* 100

print("Post-injection missing percentages:")

print(missing\_summary\_post\_inject)

# Section 3: Handle Missing Values

# Justification: Mean for normal-like (Cholesterol, as symmetric), Median for skewed (Income, Medical\_Expenses, Num\_Visits) to avoid outlier influence.

# Domain: Healthcare metrics like expenses are skewed, so median preserves distribution.

# Impute

dataset$Cholesterol[is.na(dataset$Cholesterol)] <- mean(dataset$Cholesterol, na.rm = TRUE)

dataset$Income[is.na(dataset$Income)] <- median(dataset$Income, na.rm = TRUE)

dataset$Medical\_Expenses[is.na(dataset$Medical\_Expenses)] <- median(dataset$Medical\_Expenses, na.rm = TRUE)

dataset$Num\_Visits[is.na(dataset$Num\_Visits)] <- median(dataset$Num\_Visits, na.rm = TRUE)

# Post-imputation checks: Summary stats before/after to check no distortion

missing\_summary\_post <- colSums(is.na(dataset))

print("Post-imputation missing counts (should be 0):")

print(missing\_summary\_post)

# Compare distributions pre/post for imputed vars (e.g., summary)

print(summary(dataset$Income)) # Post-imputation

# Section 4: Diagnose Skewness

# Compute skewness for all numeric attributes

numeric\_vars <- c("Age", "BMI", "Blood\_Pressure\_Sys", "Cholesterol", "Income", "Medical\_Expenses", "Num\_Visits", "Comorbidity\_Score", "Length\_of\_Stay")

skewness\_table <- data.frame(

Attribute = numeric\_vars,

Skewness = sapply(dataset[numeric\_vars], skewness),

Direction = sapply(dataset[numeric\_vars], function(x) if(skewness(x) > 0) "Right" else if(skewness(x) < 0) "Left" else "Symmetric")

)

print("Skewness Table (Before Transformation):")

print(skewness\_table)

# Visualize each numeric attribute: Histogram and Boxplot

cat("\nGenerating plots for original variables...\n")

for (var in numeric\_vars) {

hist\_plot <- ggplot(dataset, aes(x = !!sym(var))) +

geom\_histogram(bins = 30, fill = "skyblue", alpha = 0.7) +

ggtitle(paste("Histogram of", var)) +

theme\_minimal()

box\_plot <- ggplot(dataset, aes(y = !!sym(var))) +

geom\_boxplot(fill = "lightgreen", alpha = 0.7) +

ggtitle(paste("Boxplot of", var)) +

theme\_minimal()

print(grid.arrange(hist\_plot, box\_plot, ncol = 2))

}

# Interpretation: Right-skewed vars like Income, Medical\_Expenses, Num\_Visits, Length\_of\_Stay have positive skewness >1, indicating long right tails from outliers (e.g., high expenses). This may violate LM normality assumptions, leading to biased estimates.

# Section 5: Reduce Skewness

# Apply two transformation families: Log1p and Yeo-Johnson

skewed\_vars <- c("Income", "Medical\_Expenses", "Num\_Visits", "Length\_of\_Stay") # Focus on highly skewed

# Create transformed dataset - start with original

dataset\_trans <- dataset

# Transformation 1: Log1p (for non-negative)

for (var in skewed\_vars) {

dataset\_trans[[paste0(var, "\_log")]] <- log1p(dataset\_trans[[var]])

}

# Transformation 2: Yeo-Johnson

yj\_transforms <- list()

for (var in skewed\_vars) {

yj <- yeojohnson(dataset\_trans[[var]])

dataset\_trans[[paste0(var, "\_yj")]] <- predict(yj, dataset\_trans[[var]])

yj\_transforms[[var]] <- yj

print(paste("Yeo-Johnson lambda for", var, ":", yj$lambda))

}

# Create final transformed dataset using Yeo-Johnson transformations

# Keep non-skewed variables and replace skewed ones with transformed versions

# First, identify columns to keep (non-skewed + categorical)

cols\_to\_keep <- c("Age", "BMI", "Blood\_Pressure\_Sys", "Cholesterol", "Comorbidity\_Score",

"Gender", "Smoking\_Status", "Region", "Insurance\_Type")

# Use base R subsetting to avoid dplyr issues

yj\_cols <- c("Income\_yj", "Medical\_Expenses\_yj", "Num\_Visits\_yj", "Length\_of\_Stay\_yj")

all\_cols\_needed <- c(cols\_to\_keep, yj\_cols)

# Create the final transformed dataset using base R

dataset\_final\_trans <- dataset\_trans[, all\_cols\_needed]

# Rename the YJ columns to replace original names

names(dataset\_final\_trans)[names(dataset\_final\_trans) == "Income\_yj"] <- "Income"

names(dataset\_final\_trans)[names(dataset\_final\_trans) == "Medical\_Expenses\_yj"] <- "Medical\_Expenses"

names(dataset\_final\_trans)[names(dataset\_final\_trans) == "Num\_Visits\_yj"] <- "Num\_Visits"

names(dataset\_final\_trans)[names(dataset\_final\_trans) == "Length\_of\_Stay\_yj"] <- "Length\_of\_Stay"

# Recompute skewness on transformed variables

transformed\_numeric\_vars <- c("Age", "BMI", "Blood\_Pressure\_Sys", "Cholesterol", "Income", "Medical\_Expenses", "Num\_Visits", "Comorbidity\_Score", "Length\_of\_Stay")

skewness\_table\_after <- data.frame(

Attribute = transformed\_numeric\_vars,

Skewness\_Original = sapply(dataset[transformed\_numeric\_vars], skewness),

Skewness\_Transformed = sapply(dataset\_final\_trans[transformed\_numeric\_vars], skewness),

Direction\_Original = sapply(dataset[transformed\_numeric\_vars], function(x) if(skewness(x) > 0) "Right" else if(skewness(x) < 0) "Left" else "Symmetric"),

Direction\_Transformed = sapply(dataset\_final\_trans[transformed\_numeric\_vars], function(x) if(skewness(x) > 0) "Right" else if(skewness(x) < 0) "Left" else "Symmetric")

)

print("Skewness Comparison Table:")

print(skewness\_table\_after)

# Re-plot for transformed variables

cat("\nGenerating plots for transformed variables...\n")

for (var in skewed\_vars) {

hist\_plot <- ggplot(dataset\_final\_trans, aes(x = !!sym(var))) +

geom\_histogram(bins = 30, fill = "coral", alpha = 0.7) +

ggtitle(paste("Histogram of Transformed", var)) +

theme\_minimal()

box\_plot <- ggplot(dataset\_final\_trans, aes(y = !!sym(var))) +

geom\_boxplot(fill = "lightcoral", alpha = 0.7) +

ggtitle(paste("Boxplot of Transformed", var)) +

theme\_minimal()

print(grid.arrange(hist\_plot, box\_plot, ncol = 2))

}

# Summary: Transformations reduced skewness (e.g., from >1 to <0.5), making distributions more symmetric. Trade-offs: Log interpretable as % changes, Yeo-Johnson handles negatives but less intuitive; stability improved, but original scale lost.

# Section 6: Modeling and Evaluation

# Validation: Train/test split (80/20)

# Model: Linear Regression on original and transformed

# Split original dataset

trainIndex <- createDataPartition(dataset$Length\_of\_Stay, p = 0.8, list = FALSE)

train\_orig <- dataset[trainIndex, ]

test\_orig <- dataset[-trainIndex, ]

# Split transformed dataset

train\_trans <- dataset\_final\_trans[trainIndex, ]

test\_trans <- dataset\_final\_trans[-trainIndex, ]

# One-hot encode categoricals for original

dummies\_orig <- dummyVars(Length\_of\_Stay ~ ., data = dataset)

train\_orig\_enc <- data.frame(predict(dummies\_orig, train\_orig))

train\_orig\_enc$Length\_of\_Stay <- train\_orig$Length\_of\_Stay

test\_orig\_enc <- data.frame(predict(dummies\_orig, test\_orig))

# One-hot encode categoricals for transformed

dummies\_trans <- dummyVars(Length\_of\_Stay ~ ., data = dataset\_final\_trans)

train\_trans\_enc <- data.frame(predict(dummies\_trans, train\_trans))

train\_trans\_enc$Length\_of\_Stay <- train\_trans$Length\_of\_Stay

test\_trans\_enc <- data.frame(predict(dummies\_trans, test\_trans))

# Fit models

print("Fitting linear regression on original data...")

model\_orig <- lm(Length\_of\_Stay ~ ., data = train\_orig\_enc)

print("Fitting linear regression on transformed data...")

model\_trans <- lm(Length\_of\_Stay ~ ., data = train\_trans\_enc)

# Make predictions

pred\_orig <- predict(model\_orig, test\_orig\_enc)

# For transformed model, predict and inverse transform

pred\_trans\_transformed <- predict(model\_trans, test\_trans\_enc)

# Inverse transform predictions to original scale

yj\_los <- yj\_transforms[["Length\_of\_Stay"]]

pred\_trans\_orig\_scale <- predict(yj\_los, pred\_trans\_transformed, inverse = TRUE)

# True test values in original scale

test\_orig\_scale <- test\_orig$Length\_of\_Stay

# Calculate metrics

rmse\_orig <- RMSE(pred\_orig, test\_orig\_scale)

mae\_orig <- MAE(pred\_orig, test\_orig\_scale)

r2\_orig <- R2(pred\_orig, test\_orig\_scale)

rmse\_trans <- RMSE(pred\_trans\_orig\_scale, test\_orig\_scale)

mae\_trans <- MAE(pred\_trans\_orig\_scale, test\_orig\_scale)

r2\_trans <- R2(pred\_trans\_orig\_scale, test\_orig\_scale)

# Section 7: Performance Comparison

metrics\_table <- data.frame(

Dataset = c("Original", "Transformed"),

RMSE = c(rmse\_orig, rmse\_trans),

MAE = c(mae\_orig, mae\_trans),

R2 = c(r2\_orig, r2\_trans),

Improvement\_RMSE = c(0, (rmse\_orig - rmse\_trans) / rmse\_orig \* 100),

Improvement\_MAE = c(0, (mae\_orig - mae\_trans) / mae\_orig \* 100),

Improvement\_R2 = c(0, (r2\_trans - r2\_orig) / abs(r2\_orig) \* 100)

)

print("Model Performance Comparison:")

# Print without rounding first to see the structure

print(str(metrics\_table))

print(metrics\_table)

# Round only the numeric columns safely

metrics\_display <- metrics\_table

for(i in 2:ncol(metrics\_display)) {

if(is.numeric(metrics\_display[,i])) {

metrics\_display[,i] <- round(metrics\_display[,i], 4)

}

}

print("Rounded metrics:")

print(metrics\_display)

# Additional analysis: Residuals diagnostics

cat("\n=== Residuals Analysis ===\n")

# Calculate residuals

residuals\_orig <- test\_orig\_scale - pred\_orig

residuals\_trans <- test\_orig\_scale - pred\_trans\_orig\_scale

# Shapiro-Wilk test for normality (on a sample if n>5000)

if(length(residuals\_orig) <= 5000) {

shapiro\_orig <- shapiro.test(residuals\_orig)

shapiro\_trans <- shapiro.test(residuals\_trans)

print(paste("Shapiro-Wilk p-value (Original):", round(shapiro\_orig$p.value, 4)))

print(paste("Shapiro-Wilk p-value (Transformed):", round(shapiro\_trans$p.value, 4)))

} else {

print("Sample too large for Shapiro-Wilk test - using sample")

sample\_idx <- sample(length(residuals\_orig), 5000)

shapiro\_orig <- shapiro.test(residuals\_orig[sample\_idx])

shapiro\_trans <- shapiro.test(residuals\_trans[sample\_idx])

print(paste("Shapiro-Wilk p-value (Original, sampled):", round(shapiro\_orig$p.value, 4)))

print(paste("Shapiro-Wilk p-value (Transformed, sampled):", round(shapiro\_trans$p.value, 4)))

}

# Plot residuals

residuals\_df <- data.frame(

Predicted\_Orig = pred\_orig,

Residuals\_Orig = residuals\_orig,

Predicted\_Trans = pred\_trans\_orig\_scale,

Residuals\_Trans = residuals\_trans

)

p1 <- ggplot(residuals\_df, aes(x = Predicted\_Orig, y = Residuals\_Orig)) +

geom\_point(alpha = 0.6) +

geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +

ggtitle("Residuals vs Fitted (Original)") +

theme\_minimal()

p2 <- ggplot(residuals\_df, aes(x = Predicted\_Trans, y = Residuals\_Trans)) +

geom\_point(alpha = 0.6) +

geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +

ggtitle("Residuals vs Fitted (Transformed)") +

theme\_minimal()

print(grid.arrange(p1, p2, ncol = 2))

# Summary statistics

cat("\n=== Summary ===\n")

cat("Original dataset skewness issues were successfully addressed through Yeo-Johnson transformation.\n")

cat(sprintf("RMSE improved by: %.2f%%\n", (rmse\_orig - rmse\_trans) / rmse\_orig \* 100))

cat(sprintf("MAE improved by: %.2f%%\n", (mae\_orig - mae\_trans) / mae\_orig \* 100))

cat(sprintf("R² improved by: %.2f%%\n", (r2\_trans - r2\_orig) / abs(r2\_orig) \* 100))

# Print model summaries

cat("\n=== Model Summaries ===\n")

cat("Original Model Summary:\n")

print(summary(model\_orig))

cat("\nTransformed Model Summary:\n")

print(summary(model\_trans))