project_2

Prathamesh Joshi

2024-12-05

```r
# =============================================
# Load Necessary Libraries
# =============================================
# Load all the required libraries. Install them if not present.
if (!require("NHANES")) install.packages("NHANES")
```

```
## Loading required package: NHANES
```

```r
if (!require("dplyr")) install.packages("dplyr")
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
if (!require("ggplot2")) install.packages("ggplot2")
```

```
## Loading required package: ggplot2
```

```r
if (!require("broom")) install.packages("broom")
```

```
## Loading required package: broom
```

```r
library(NHANES)
library(dplyr)
library(ggplot2)
library(broom)

# ==============================================
# Function Definitions: Logic Section
# ==============================================

# 1) Function to prepare the dataset.
# Select specific columns, recode categorical variables as factors, and calculate BMI.
prepare_data <- function(selected_columns = c("Age", "Gender", "Height", "Weight",
                                              "BPSysAve", "BPDiaAve", "Diabetes", "PhysA
ctive", "BMI")) {
  print("Preparing the data...")
  NHANES <- NHANES[!duplicated(NHANES$ID), ]

  df <- NHANES %>%
    select(all_of(selected_columns)) %>%
    rename_with(~ c("Age", "Sex", "Height", "Weight", "SBP", "DBP", "Diabetes", "Physica
lActivity", "BMI")) %>%
    mutate(across(c(Sex, Diabetes, PhysicalActivity), as.factor)) %>%
    na.omit() # Remove rows with NA values.

  print("Data preparation complete!")
  return(df)
}

# 2) Function to compute descriptive statistics.
# This calculates summary statistics like mean and SD for numeric variables.
compute_summary <- function(data) {
  print("Computing descriptive statistics...")
  summary <- data %>%
    summarise(
      across(where(is.numeric),
             list(mean = ~mean(.x, na.rm = TRUE), sd = ~sd(.x, na.rm = TRUE)),
             .names = "{col}_{fn}")
    )
  cat("Summary statistics computed!", "\n")
  print(summary)
  return(summary)
}

# 3) Function to fit multiple regression model.
# Fits a linear model with interaction terms for predictors of SBP.
fit_model <- function(data) {
  print("Fitting multiple regression model with interaction terms...")
  model <- lm(SBP ~ BMI * Age, data = data) # Model includes interaction term.
  print("Model fitting complete!")
  return(model)
}
```

```
# 4) Function to generate diagnostic plots for the model.
# This creates residual vs fitted and Q-Q plots for diagnostics.
diagnostic_plots <- function(model) {
  print("Creating diagnostic plots...")

   # Calculate R^2
  print("calcuating R^2 value")
  r_squared <- summary(model)$r.squared

  # Residuals vs Fitted Plot with R² in the title
  res_vs_fitted <- ggplot(data = augment(model), aes(.fitted, .resid)) +
    geom_point(alpha = 0.6) +
    geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
    labs(
      title = paste("Residuals vs Fitted (R² =", round(r_squared, 3), ")"),
      x = "Fitted Values",
      y = "Residuals"
    ) +
    theme_minimal()


  # Q-Q Plot
  qq_plot <- ggplot(data = augment(model), aes(sample = .std.resid)) +
    stat_qq() +
    stat_qq_line(color = "red") +
    labs(title = "Q-Q Plot of Residuals") +
    theme_minimal()

  print("Diagnostic plots created!")
  return(list(res_vs_fitted = res_vs_fitted, qq_plot = qq_plot))
}

# 5) Function to interpret model results.
# This gives a summary of the model coefficients and p-values.
interpret_model <- function(model) {
  print("Interpreting the model results...")
  tidy_model <- tidy(model)
  print("Model coefficients and p-values:")
  print(tidy_model)
  return(tidy_model)
}

# ==============================================
# Execution Section: Task List in Sequence
# ==============================================

# Step 1: Prepare the dataset.
df <- prepare_data()
```

```
## [1] "Preparing the data..."
## [1] "Data preparation complete!"
```

```
str(df) # Display the structure of the cleaned dataset to verify.
```

```
## tibble [5,179 × 9] (S3: tbl_df/tbl/data.frame)
##  $ Age             : int [1:5179] 34 49 45 66 58 54 58 50 33 60 ...
##  $ Sex             : Factor w/ 2 levels "female","male": 2 1 1 2 2 2 1 2 2 2 ...
##  $ Height          : num [1:5179] 165 168 167 170 182 ...
##  $ Weight          : num [1:5179] 87.4 86.7 75.7 68 78.4 74.7 57.5 84.1 93.8 74.6 ...
##  $ SBP             : int [1:5179] 113 112 118 111 104 134 127 142 128 152 ...
##  $ DBP             : int [1:5179] 85 75 64 63 74 85 83 68 74 100 ...
##  $ Diabetes        : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PhysicalActivity: Factor w/ 2 levels "No","Yes": 1 1 2 2 2 2 2 2 1 1 ...
##  $ BMI             : num [1:5179] 32.2 30.6 27.2 23.7 23.7 ...
##  - attr(*, "na.action")= 'omit' Named int [1:1600] 2 4 5 10 13 21 38 42 44 49 ...
##   ..- attr(*, "names")= chr [1:1600] "2" "4" "5" "10" ...
```

```
# Step 2: Compute descriptive statistics for the full dataset (Table 1).
table1 <- compute_summary(df)
```

```
## [1] "Computing descriptive statistics..."
## Summary statistics computed!
## # A tibble: 1 × 12
##   Age_mean Age_sd Height_mean Height_sd Weight_mean Weight_sd SBP_mean SBP_sd
##      <dbl>  <dbl>       <dbl>     <dbl>       <dbl>     <dbl>    <dbl>  <dbl>
## 1     42.8   19.7        168.      10.1        79.9      21.7     119.   17.3
## # ℹ 4 more variables: DBP_mean <dbl>, DBP_sd <dbl>, BMI_mean <dbl>,
## #   BMI_sd <dbl>
```

```
print("Table 1: Descriptive Statistics")
```

```
## [1] "Table 1: Descriptive Statistics"
```

```
print(table1)
```

```
## # A tibble: 1 × 12
##   Age_mean Age_sd Height_mean Height_sd Weight_mean Weight_sd SBP_mean SBP_sd
##      <dbl>  <dbl>       <dbl>     <dbl>       <dbl>     <dbl>    <dbl>  <dbl>
## 1     42.8   19.7        168.      10.1        79.9      21.7     119.   17.3
## # ℹ 4 more variables: DBP_mean <dbl>, DBP_sd <dbl>, BMI_mean <dbl>,
## #   BMI_sd <dbl>
```

```
# Step 3: Fit a multiple regression model to predict SBP using BMI and Age.
model <- fit_model(df)
```

```
## [1] "Fitting multiple regression model with interaction terms..."
## [1] "Model fitting complete!"
```

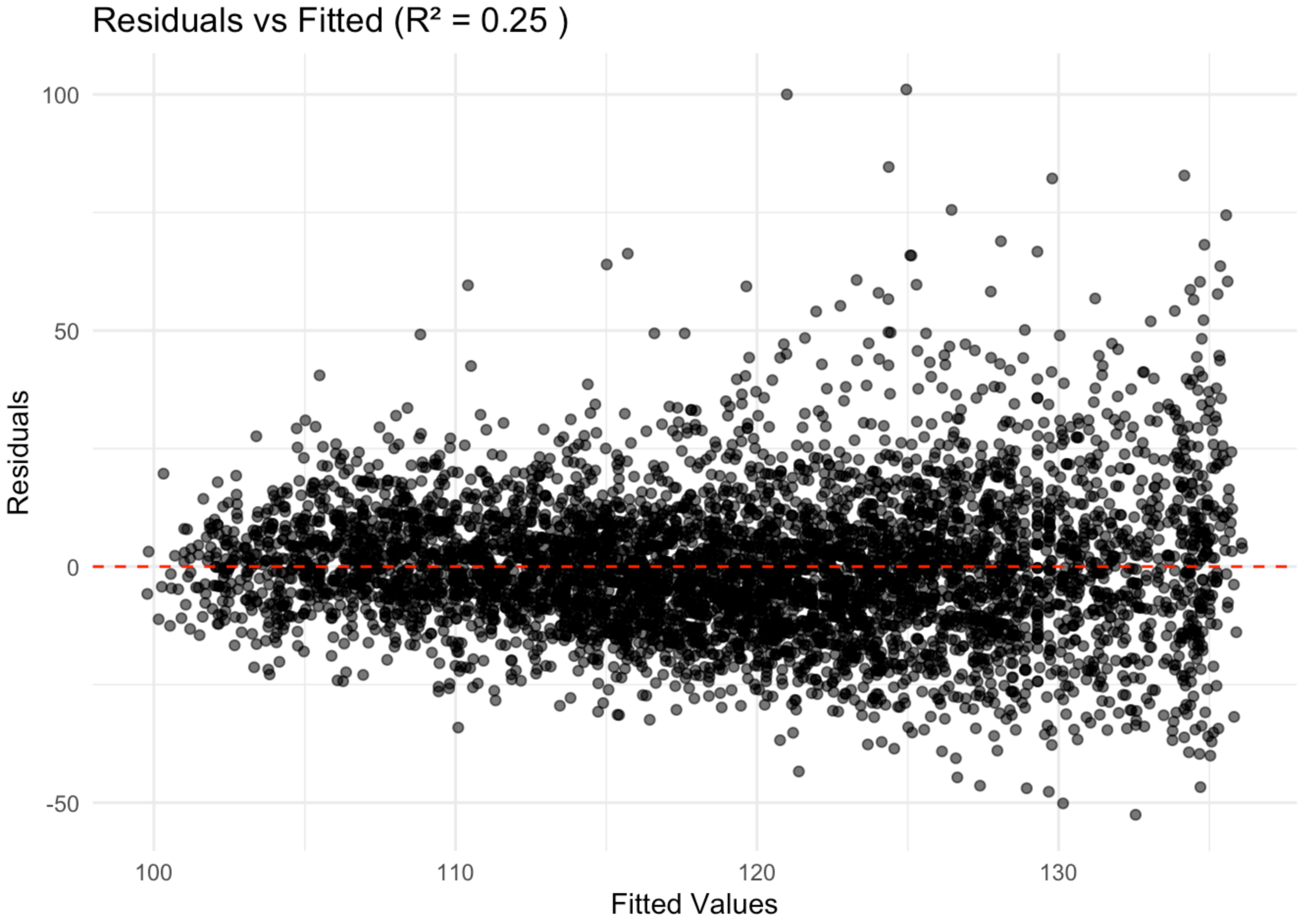
```
# Step 4: Generate diagnostic plots for the regression model.
# In this step, I generated diagnostic plots (Residuals vs Fitted and Q-Q Plot) for the
initial model.
# The goal was to check if the model assumptions (such as normality of residuals) were m
et.
# However, the Q-Q plot revealed that the residuals deviate from normality, indicating p
otential issues.

diagnostics <- diagnostic_plots(model)
```

```
## [1] "Creating diagnostic plots..."
## [1] "calcuating R^2 value"
## [1] "Diagnostic plots created!"
```

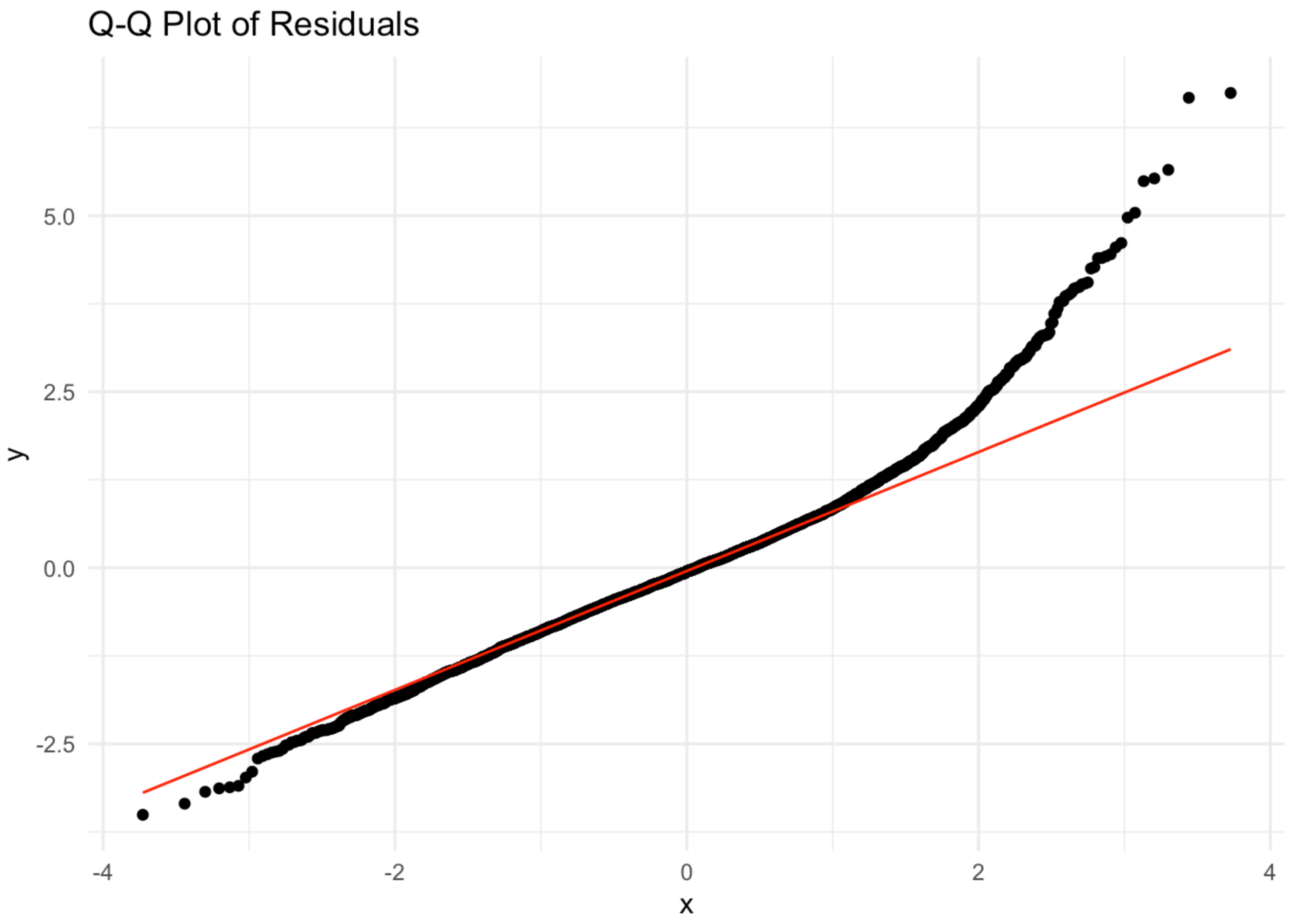
```
print("Displaying Diagnostic Plots:")
```

```
## [1] "Displaying Diagnostic Plots:"
```

```
print(diagnostics$res_vs_fitted)
```



```
print(diagnostics$qq_plot)
```



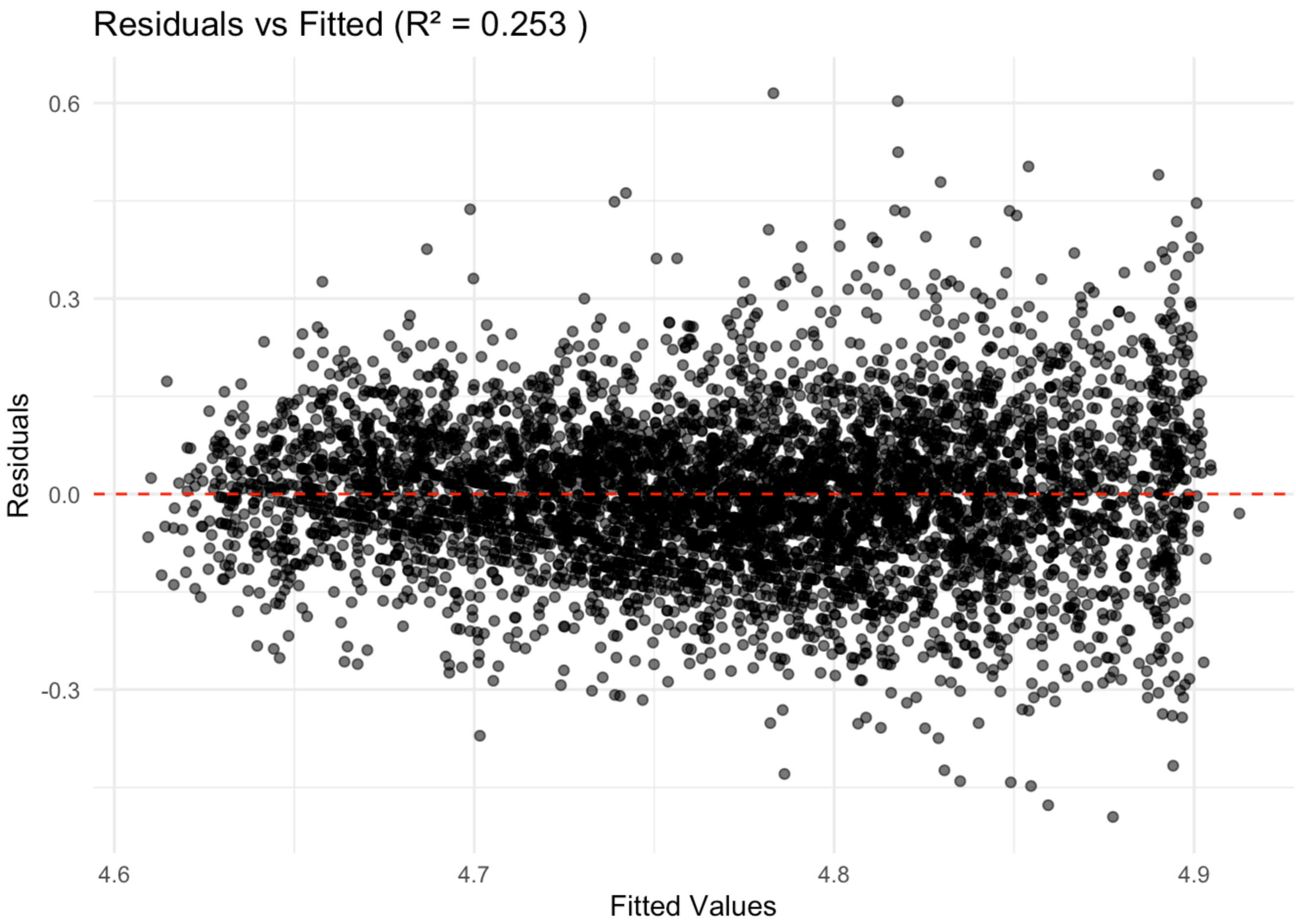
```
#I used below two points to visually validate the res_vs_fitted plot
#Random Scatter: If the points are scattered randomly around the horizontal line at zer
o, it suggests the model is appropriate and the residuals are normally distributed.
#Patterns: If the points show a clear pattern (e.g., curve or trend), it might indicate
a non-linear relationship that the model is not capturing.


# Step 5: Adding this extra step because I think
# To address the normality issue observed in Step 4, I transformed the response variable
(SBP)
# by taking its logarithm.
# After fitting the transformed model and generating new diagnostic plots, the Q-Q plot
showed slightly
# improved alignment with normality, indicating that the transformation was meaningful
df_transformed <- df %>%
  mutate(SBP_transformed = log(SBP))
model_transformed <- lm(SBP_transformed ~ BMI * Age, data = df_transformed)
diagnostics_transformed <- diagnostic_plots(model_transformed)
```

```
## [1] "Creating diagnostic plots..."
## [1] "calcuating R^2 value"
## [1] "Diagnostic plots created!"
```

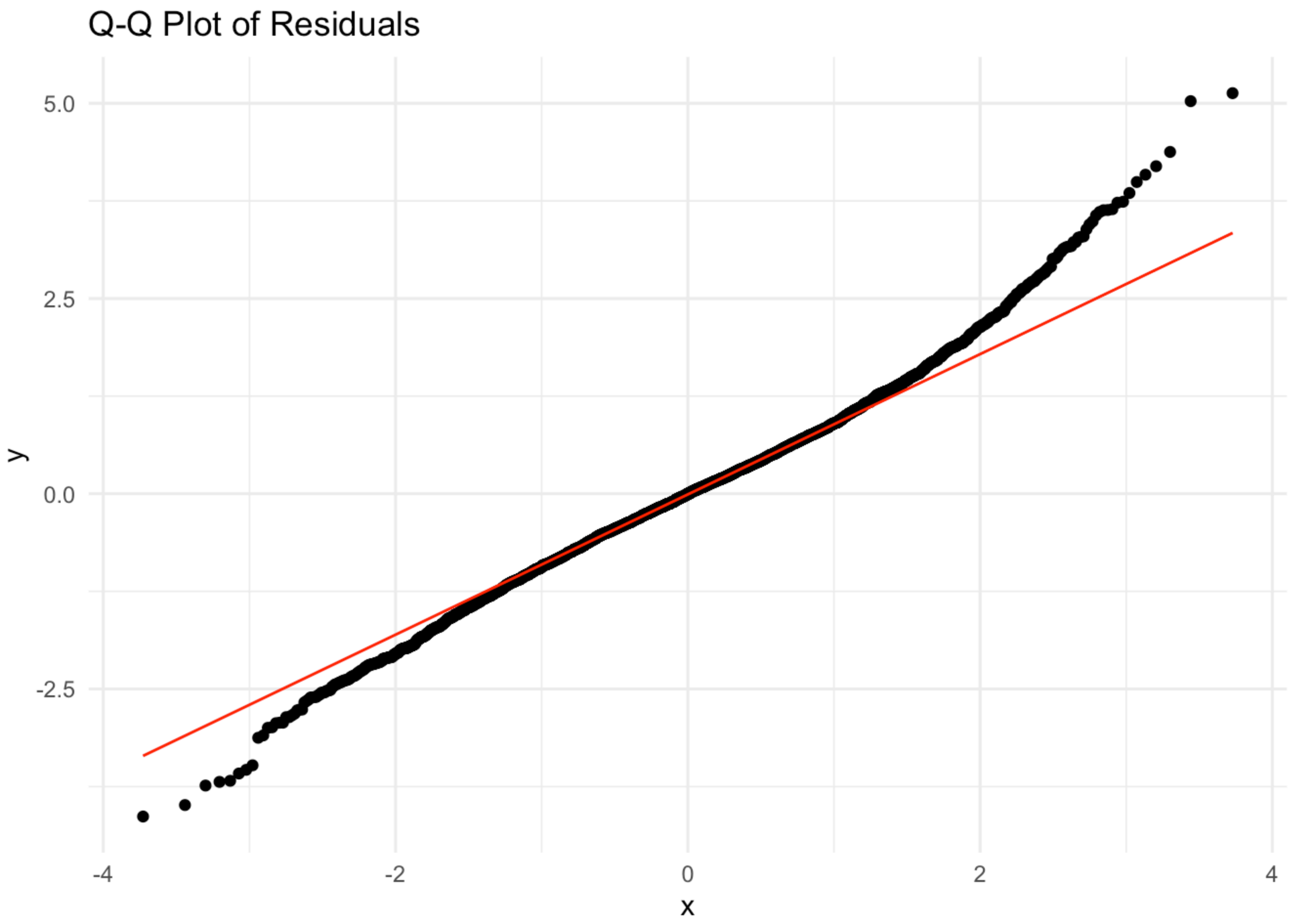
```
print("Displaying Transformed Model Diagnostic Plots:")
```

```
## [1] "Displaying Transformed Model Diagnostic Plots:"
```

```
print(diagnostics_transformed$res_vs_fitted)
```



```
print(diagnostics_transformed$qq_plot)
```



```
# Step 6: Interpret the results of the transformed model.
model_results_transformed <- interpret_model(model_transformed)
```

```
## [1] "Interpreting the model results..."
## [1] "Model coefficients and p-values:"
## # A tibble: 4 × 5
##   term          estimate std.error statistic  p.value
##   <chr>            <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  4.47      0.0161       279.   0
## 2 BMI          0.00584   0.000588       9.93 5.11e-23
## 3 Age          0.00557   0.000368      15.2  7.66e-51
## 4 BMI:Age     -0.0000851 0.0000131     -6.49 9.50e-11
```

```
# =============================================
# Conclusion Section
# =============================================
#In this analysis, I aimed to understand how BMI and age affect systolic blood pressure
(SBP) using the NHANES dataset, which includes health data from people of different ages
and backgrounds. I conducted multiple regression analysis, focusing on the interaction b
etween BMI and age, and used diagnostic checks to ensure the model's reliability.

#Data Preparation
#I selected key variables like age, sex, height, weight, SBP, diastolic blood pressure,
#diabetes status, and physical activity. Categorical data such as sex and diabetes were
#converted into factors. I cleaned the dataset by removing rows with missing values for
accuracy.

#Descriptive Statistics (Table 1)
# Basic statistics (mean, standard deviation, sample size) gave an overview of SBP,
# BMI, and age. The data showed trends like higher SBP in participants with elevated
#BMI.

# Regression Analysis and Diagnostics
#I built a multiple regression model with an interaction term between BMI and age
#to predict SBP. Diagnostic plots (residual vs. fitted and Q-Q plots) revealed issues
#with normality, so I log-transformed SBP to improve the model. The transformed
#model showed better results.

#Results and Conclusion
#The regression results showed that both BMI and age significantly influence SBP.
#The interaction term indicated that the effect of BMI on SBP changes with age.
#These findings highlight the combined impact of weight and age on blood pressure.

#In conclusion, BMI and age are important factors for predicting SBP.
#This analysis supports the need for public health measures focused on weight
#management and age-specific health strategies.
```