Regression Analysis Project: Predicting Glucose Levels

2025-06-03

```
# Loading Libraries
suppressPackageStartupMessages({
  suppressWarnings({
    library(tidyverse)
    library(corrplot)
    library(ggplot2)
    library(reshape2)
    library(gridExtra)
    library(Metrics)
    library(car)
    library(lmtest)
    library(MASS)
  })
})
```

Introduction:

Diabetes is a chronic health issue that affects millions of people all over the world. Diabetes can lead to serious health complications if not diagnosed early.Blood Glucose is a critical indicator when it comes to monitoring diabetes. I have taken this dataset from Kaggle - https://www.kaggle.com/datasets/mathchi/diabetes-data-set (https://www.kaggle.com/datasets/mathchi/diabetes-data-set). The dataset was originally collected by National Institute of Diabetes, Digestive and Kidney Diseases as part of a study on Pima Indian women. The main aim of this project is to identify the most effective model for predicting glucose level and to evaluate its assumptions,accuracy and robustness.

Total number of observations present are 768. We have multiple predictor variables and 1 Outcome variable.

Predictor Variables are Pregnancies: Number of times pregnant. It is int datatype Glucose: Plasma glucose concentration. It is int datatype BloodPressure: Diastolic blood pressure. It is int datatype SkinThickness: Triceps skin fold thickness. It is int datatype Insulin: 2-Hour serum insulin. It is int datatype BMI: Body mass index. It is numerical datatype DiabetesPedigreeFunction: Diabetes pedigree function. It is numerical datatype Age: Age of patient. It is int datatype Target Variable is the Outcome variable which is a Binary Indicator with 0 for No Diabetes and 1 for Diabetes.It is int datatype

We are going to consider Age,BMI,Insulin,BloodPressure and DiabetesPedigreeFunction in this analysis. As they are the prime contributors to increase in Glucose levels and we are going to omit Pregnancies, SkinThickness and Outcome.

Outcome is a binary variable. It shows whether the person has diabetes or not. Since our aim is to predict Glucose’s continuous response variable, binary target is not suitable as a predictor.

Pregnancies is biologically related to females and may introduce gender specific bias. So we are not considering this variable.

SkinThickness is a localized measure of body fat, whereas BMI is a broader aspect and provides a overall body composition.

```
# Loading dataset
df <- read.csv("diabetes.csv")
head(df)
```

```
##   Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           6     148            72            35       0 33.6
## 2           1      85            66            29       0 26.6
## 3           8     183            64             0       0 23.3
## 4           1      89            66            23      94 28.1
## 5           0     137            40            35     168 43.1
## 6           5     116            74             0       0 25.6
##   DiabetesPedigreeFunction Age Outcome
## 1                    0.627  50       1
## 2                    0.351  31       0
## 3                    0.672  32       1
## 4                    0.167  21       0
## 5                    2.288  33       1
## 6                    0.201  30       0
```

We are removing Outcome, Preganancies and Skinthickness in the below part

```
# Excluding 'Outcome', 'Pregnancies', and 'SkinThickness' columns
df_noutcome <- df[, !names(df) %in% c("Outcome", "Pregnancies", "SkinThickness")]
```

Glucose range from 0 to 199, with a mean of 120.9 and a median of 117. BloodPressure range between 0 and 122, with a mean of 69.1. Insulin shows a wide spread with values between 0 to 846 with a median of 30.5. BMI averages around 31.99, with values up to 67.1. DiabetesPedigreeFunction ranges from 0.078 to 2.42 and Age spans from 21 to 81, with a mean of 33.24.

```
# Checking data structure
str(df_noutcome)
```

```
## 'data.frame':    768 obs. of  6 variables:
##  $ Glucose                 : int  148 85 183 89 137 116 78 115 197 125 ...
##  $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
```

```
summary(df_noutcome)
```

```
##     Glucose      BloodPressure       Insulin           BMI       
##  Min.   :  0.0   Min.   :  0.00   Min.   :  0.0   Min.   : 0.00  
##  1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.:  0.0   1st Qu.:27.30  
##  Median :117.0   Median : 72.00   Median : 30.5   Median :32.00  
##  Mean   :120.9   Mean   : 69.11   Mean   : 79.8   Mean   :31.99  
##  3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:127.2   3rd Qu.:36.60  
##  Max.   :199.0   Max.   :122.00   Max.   :846.0   Max.   :67.10  
##  DiabetesPedigreeFunction      Age       
##  Min.   :0.0780           Min.   :21.00  
##  1st Qu.:0.2437           1st Qu.:24.00  
##  Median :0.3725           Median :29.00  
##  Mean   :0.4719           Mean   :33.24  
##  3rd Qu.:0.6262           3rd Qu.:41.00  
##  Max.   :2.4200           Max.   :81.00  
```

```
sum(is.na(df_noutcome))
```

```
## [1] 0
```

```
colnames(df_noutcome)
```

```
## [1] "Glucose"                  "BloodPressure"           
## [3] "Insulin"                  "BMI"                     
## [5] "DiabetesPedigreeFunction" "Age"
```

There are no missing values in the dataset.

```
# Checking missing values
colSums(is.na(df_noutcome))
```

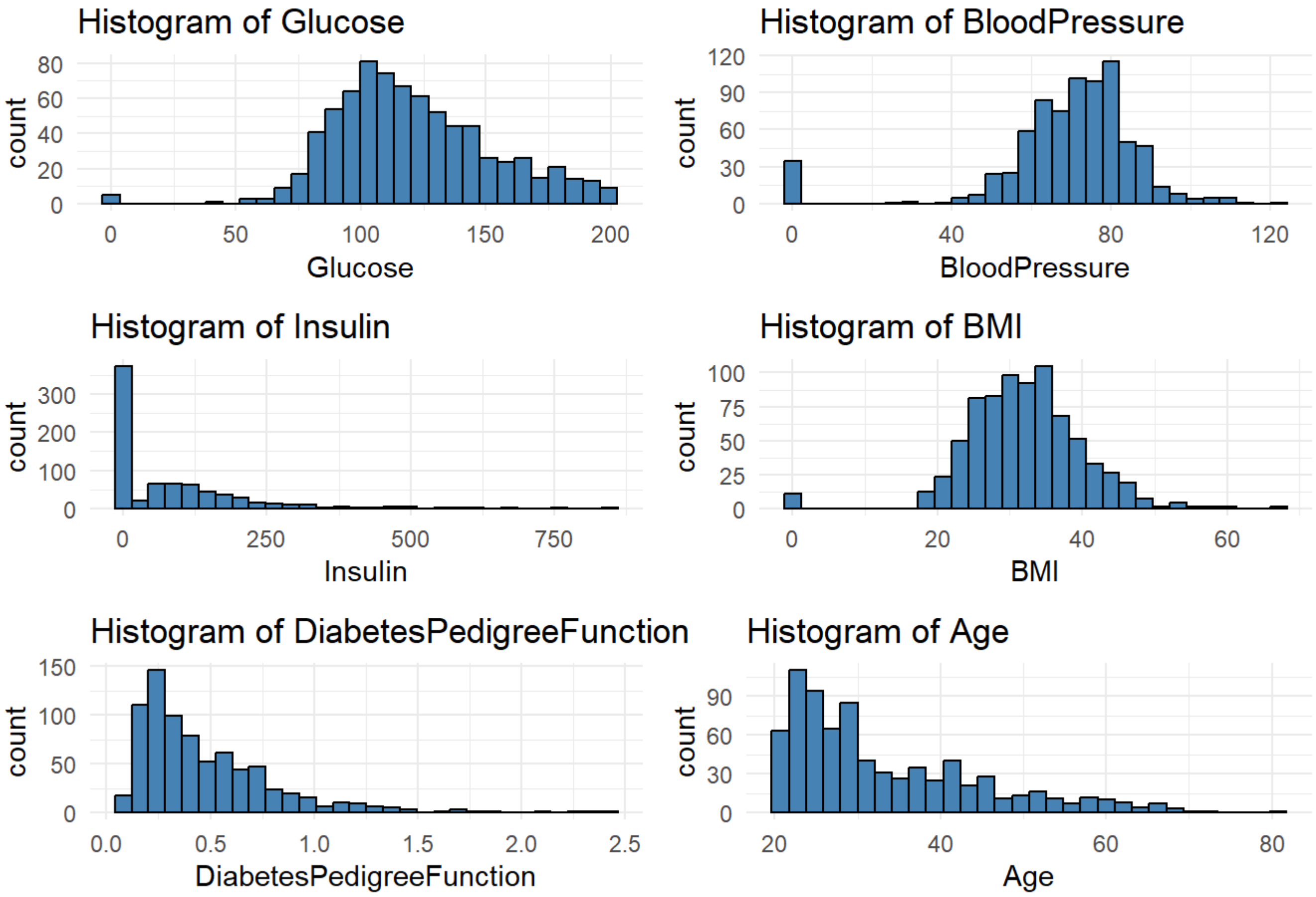
```
##                  Glucose            BloodPressure                  Insulin 
##                        0                        0                        0 
##                      BMI DiabetesPedigreeFunction                      Age 
##                        0                        0                        0
```

From Histogram, we can see that Glucose shows a slightly left skewed distribution with more values at the far higher end of the spectrum.For BloodPressure, we could see a good normal distribution.In terms of Insulin, the distribution is right skewed.For BMI, we could see a fairly normal distribution.For DiabetesPedigreeFunction, we could see a right skewed distribution and for Age also we could see a right skewed distribution.

From Boxplot graph, we can see that Glucose, BloodPressure, Insulin, BMI,DiabetesPedigreeFunction and Age have outliers, we have to deal with these outliers before working on the model.

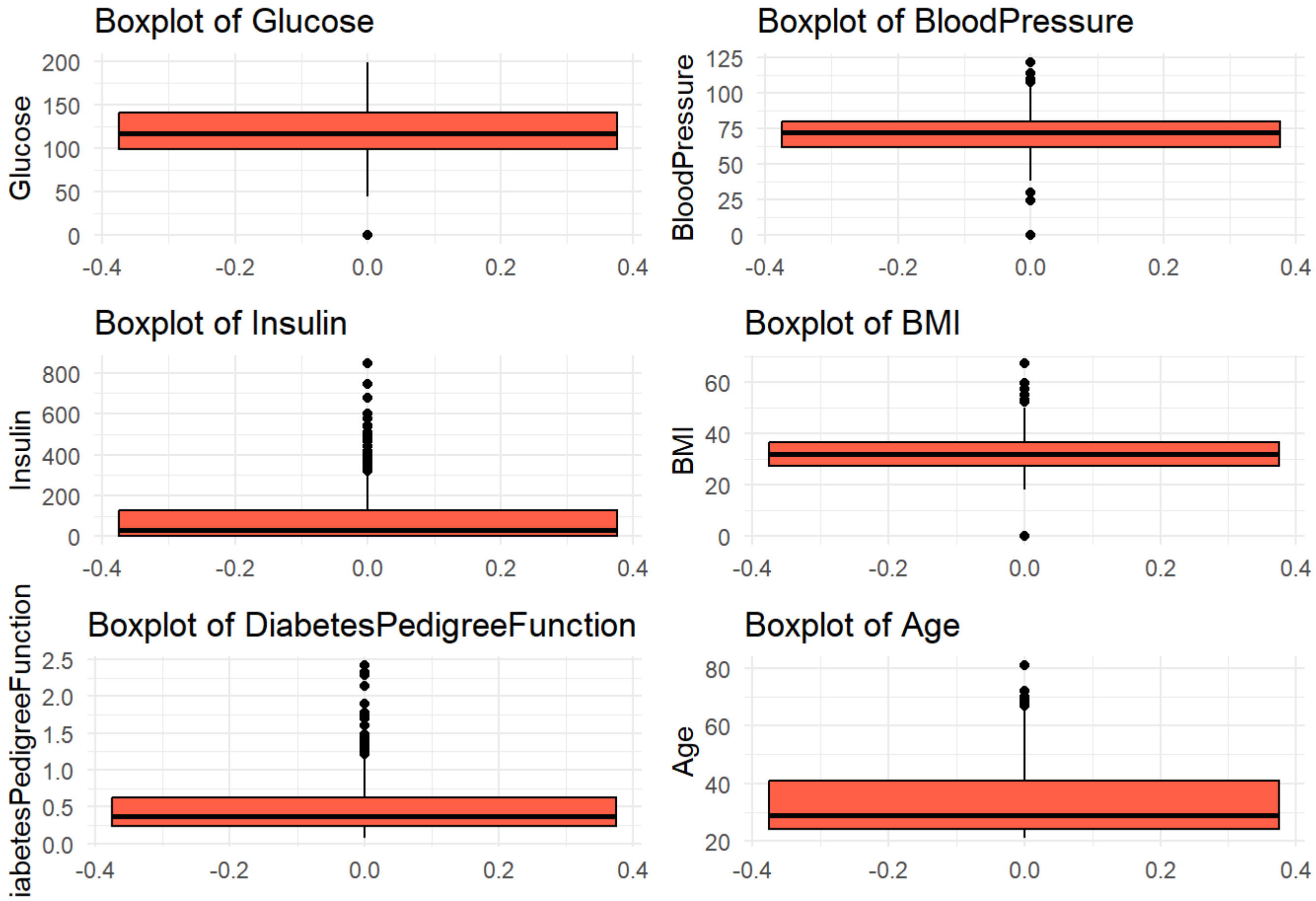
```
# Histogram
hist_plots <- lapply(names(df_noutcome), function(var) {
  ggplot(df_noutcome, aes(x = !!sym(var))) +
    geom_histogram(bins = 30, fill = "steelblue", color = "black") +
    theme_minimal() +
    ggtitle(paste("Histogram of", var))
})

# Boxplots
box_plots <- lapply(names(df_noutcome), function(var) {
  ggplot(df_noutcome, aes(y = !!sym(var))) +
    geom_boxplot(fill = "tomato", color = "black") +
    theme_minimal() +
    ggtitle(paste("Boxplot of", var))
})

# Displaying histogram
do.call("grid.arrange", c(hist_plots, ncol = 2, top = "Figure 1: Histogram of Predictors"))
```

Figure 1: Histogram of Predictors



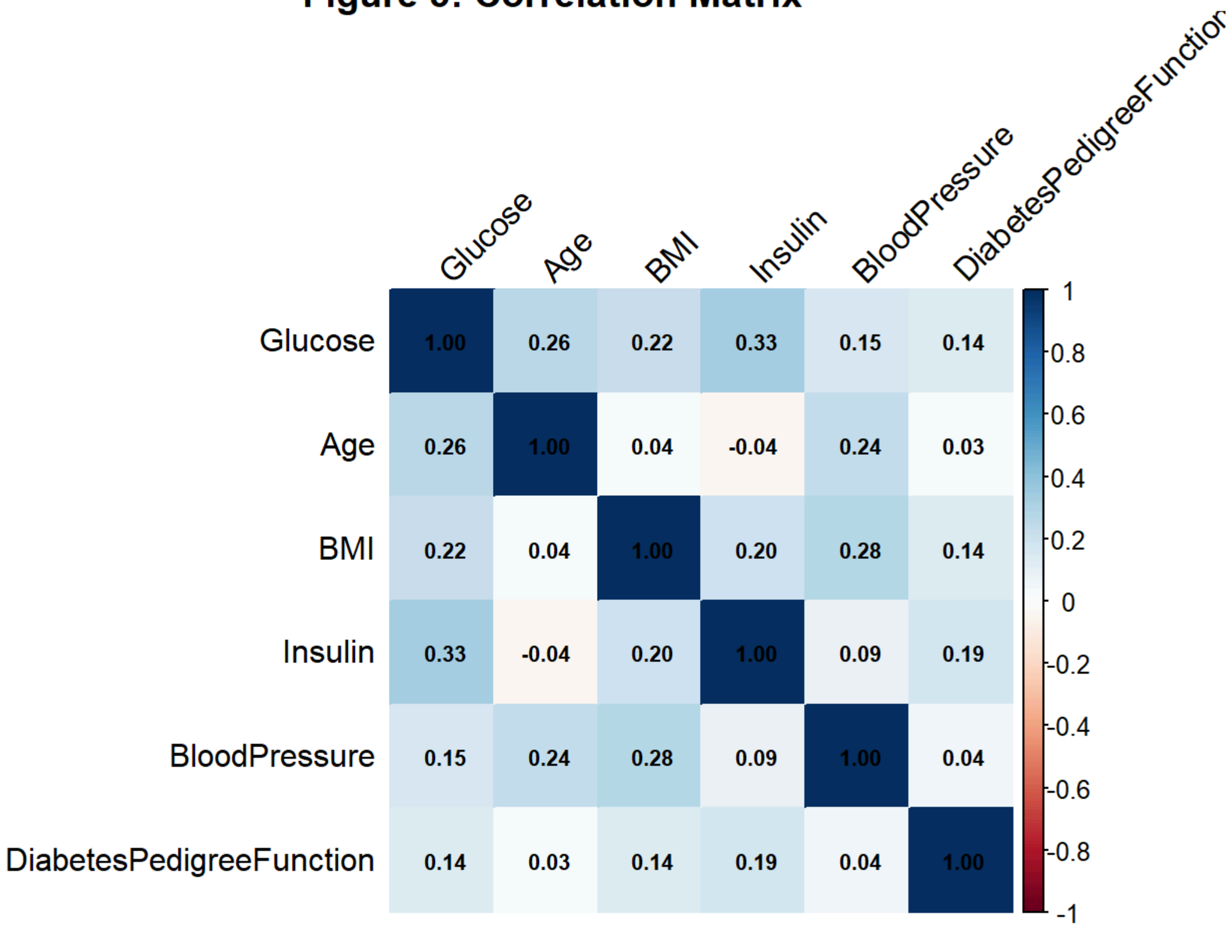
```
# Displaying boxplot
do.call("grid.arrange", c(box_plots, ncol = 2, top = "Figure 2: Boxplots of Predictors"))
```

Figure 2: Boxplots of Predictors



```
# Correlation matrix
cor_data <- df_noutcome[, c("Glucose", "Age", "BMI", "Insulin", "BloodPressure", "DiabetesPedigreeFunction")]
cor_matrix <- cor(cor_data, use = "complete.obs")

# Correlation heatmap
corrplot(cor_matrix, method = "color", addCoef.col = "black", number.cex = 0.7,
         tl.col = "black", tl.srt = 45, title = "Figure 3: Correlation Matrix", mar=c(0,0,1,0))
```

Figure 3: Correlation Matrix



In the Correlation matrix, we are focusing on Glucose vs the predictors, Insulin has 0.33 correlation with Glucose, shows moderate positive correlation.Next is Age with Glucose, which has a value of 0.26, which is weak to moderate positive correlation. BMI has a value of 0.22 which is weak positive correlation. Blood Pressure has a value of 0.15 which is very weak positive correlation. DiabetesPedigreeFunction is the weakest of all the predictors with a value of 0.14 positive correlation. Insulin is the strongest linear predictor of Glucose with Age and BMI being the next strongest predictors, Blood Pressure and DiabetesPedigreeFunction show weak correlation

We are creating a function to count outliers using the IQR method. We will apply this function to the df_noutcome to see the total number of Outliers present in each of the predictors.

Insulin has 34 outliers, Glucose has 5 outliers, BloodPressure has 45 which is the largest number in all of these predictors, Age has 9 outliers,BMI has 19 outliers and DiabetesPedigreeFunction has 29 outliers.

```r
# Function to count outliers using IQR method
count_outliers_iqr <- function(x) {
  Q1 <- quantile(x, 0.25, na.rm = TRUE)
  Q3 <- quantile(x, 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  lower <- Q1 - 1.5 * IQR
  upper <- Q3 + 1.5 * IQR
  sum(x < lower | x > upper, na.rm = TRUE)
}

outlier_counts <- sapply(df_noutcome, count_outliers_iqr)
outlier_counts
```

```
##                  Glucose            BloodPressure                  Insulin
##                        5                       45                       34
##                      BMI DiabetesPedigreeFunction                      Age
##                       19                       29                        9
```

Methods:

For Insulin, we are doing log transformation, the reason is, from the Histogram graph, we can see that the distribution is right skewed, the same applies to DiabetesPedigreeFunction. For BloodPressure, we are doing square root transformation to reduce moderate skewness and stabilize variance.For Age also its right skewed, we are doing log transformation and for BMI we are doing sqrt transformation to reduce skewness.We are not going to disturb Glucose now has it has very less outliers and in the future if we want to perform any transformations like Box-Cox, the outliers will be suppressed automatically.Also Glucose is relatively symmetric so at the moment we are not doing any changes to this response variable.

```
df_transformed <- df_noutcome

# Applying transformations
df_transformed$Insulin <- log1p(df_transformed$Insulin)
df_transformed$DiabetesPedigreeFunction <- log1p(df_transformed$DiabetesPedigreeFunction)
df_transformed$BloodPressure <- sqrt(df_transformed$BloodPressure)
df_transformed$Age <- log1p(df_transformed$Age)
df_transformed$BMI <- sqrt(df_transformed$BMI)

# Viewing summary to confirm
summary(df_transformed)
```

```
##     Glucose      BloodPressure       Insulin           BMI
##  Min.   :  0.0   Min.   : 0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.: 99.0   1st Qu.: 7.874   1st Qu.:0.000   1st Qu.:5.225
##  Median :117.0   Median : 8.485   Median :3.449   Median :5.657
##  Mean   :120.9   Mean   : 8.091   Mean   :2.472   Mean   :5.584
##  3rd Qu.:140.2   3rd Qu.: 8.944   3rd Qu.:4.854   3rd Qu.:6.050
##  Max.   :199.0   Max.   :11.045   Max.   :6.742   Max.   :8.191
##  DiabetesPedigreeFunction      Age
##  Min.   :0.07511          Min.   :3.091
##  1st Qu.:0.21813          1st Qu.:3.219
##  Median :0.31663          Median :3.401
##  Mean   :0.36532          Mean   :3.482
##  3rd Qu.:0.48628          3rd Qu.:3.738
##  Max.   :1.22964          Max.   :4.407
```

```
sapply(df_transformed, count_outliers_iqr)
```

```
##                  Glucose            BloodPressure                  Insulin
##                        5                       41                        0
##                      BMI DiabetesPedigreeFunction                      Age
##                       16                       13                        0
```

After performing all the trnasformations, we could still see some outliers present, so the best thing we can do is construct 2 models 1.model without outliers 2.model with outliers

Compare both the model’s AIC values and other aspects and see which model performs best and use it for further analysis.

We are first fitting a model with outliers.

$$E(\hat{y}) = -16.1852 + 29.2043x_1 + 5.1581x_2 + 1.9237x_3 - 0.2908x_4 + 11.4876x_5$$

```
# Model A: With outliers
model_with_outliers <- lm(Glucose ~ Age + BMI + Insulin + BloodPressure + DiabetesPedigreeFun
ction, data = df_transformed)
summary(model_with_outliers)
```

```
##
## Call:
## lm(formula = Glucose ~ Age + BMI + Insulin + BloodPressure +
##     DiabetesPedigreeFunction, data = df_transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -129.062  -19.002   -2.789   19.402   83.515
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -16.1852    13.5529  -1.194   0.2328
## Age                       29.2043     3.5822   8.153 1.46e-15 ***
## BMI                        5.1581     1.2732   4.051 5.62e-05 ***
## Insulin                    1.9237     0.4657   4.130 4.02e-05 ***
## BloodPressure             -0.2908     0.6061  -0.480   0.6315
## DiabetesPedigreeFunction  11.4876     5.5748   2.061   0.0397 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.87 on 762 degrees of freedom
## Multiple R-squared:  0.1331, Adjusted R-squared:  0.1274
## F-statistic: 23.39 on 5 and 762 DF,  p-value: < 2.2e-16
```

We are removing the outliers using a function based on the IQR rule.

```
remove_outliers <- function(x) {
  Q1 <- quantile(x, 0.25, na.rm = TRUE)
  Q3 <- quantile(x, 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  x[x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR)] <- NA
  return(x)
}

# Model B: Without outliers
df_clean <- as.data.frame(lapply(df_transformed, remove_outliers))
df_clean <- na.omit(df_clean)

model_without_outliers <- lm(Glucose ~ Age + BMI + Insulin + BloodPressure + DiabetesPedigree
Function, data = df_clean)
summary(model_without_outliers)
```

```
##
## Call:
## lm(formula = Glucose ~ Age + BMI + Insulin + BloodPressure +
##     DiabetesPedigreeFunction, data = df_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.596 -18.817  -2.267  17.560  81.438
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -50.8966    17.1304  -2.971 0.003069 **
## Age                       24.5428     3.6925   6.647 6.05e-11 ***
## BMI                        7.4154     2.0007   3.706 0.000227 ***
## Insulin                    1.9655     0.4595   4.278 2.15e-05 ***
## BloodPressure              4.5240     1.7666   2.561 0.010651 *
## DiabetesPedigreeFunction   4.1206     6.1442   0.671 0.502667
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.39 on 697 degrees of freedom
## Multiple R-squared:  0.1422, Adjusted R-squared:  0.136
## F-statistic:  23.1 on 5 and 697 DF,  p-value: < 2.2e-16
```

```
pred_with <- predict(model_with_outliers)
pred_without <- predict(model_without_outliers)

# True values
actual_with <- df_transformed$Glucose
actual_without <- df_clean$Glucose

cat("With Outliers - MAE:", mae(actual_with, pred_with),
    "RMSE:", rmse(actual_with, pred_with),
    "AIC:", AIC(model_with_outliers), "\n")
```

```
## With Outliers - MAE: 23.10728 RMSE: 29.74998 AIC: 7404.874
```

```
cat("Without Outliers - MAE:", mae(actual_without, pred_without),
    "RMSE:", rmse(actual_without, pred_without),
    "AIC:", AIC(model_without_outliers), "\n")
```

```
## Without Outliers - MAE: 22.2815 RMSE: 28.26667 AIC: 6707.434
```

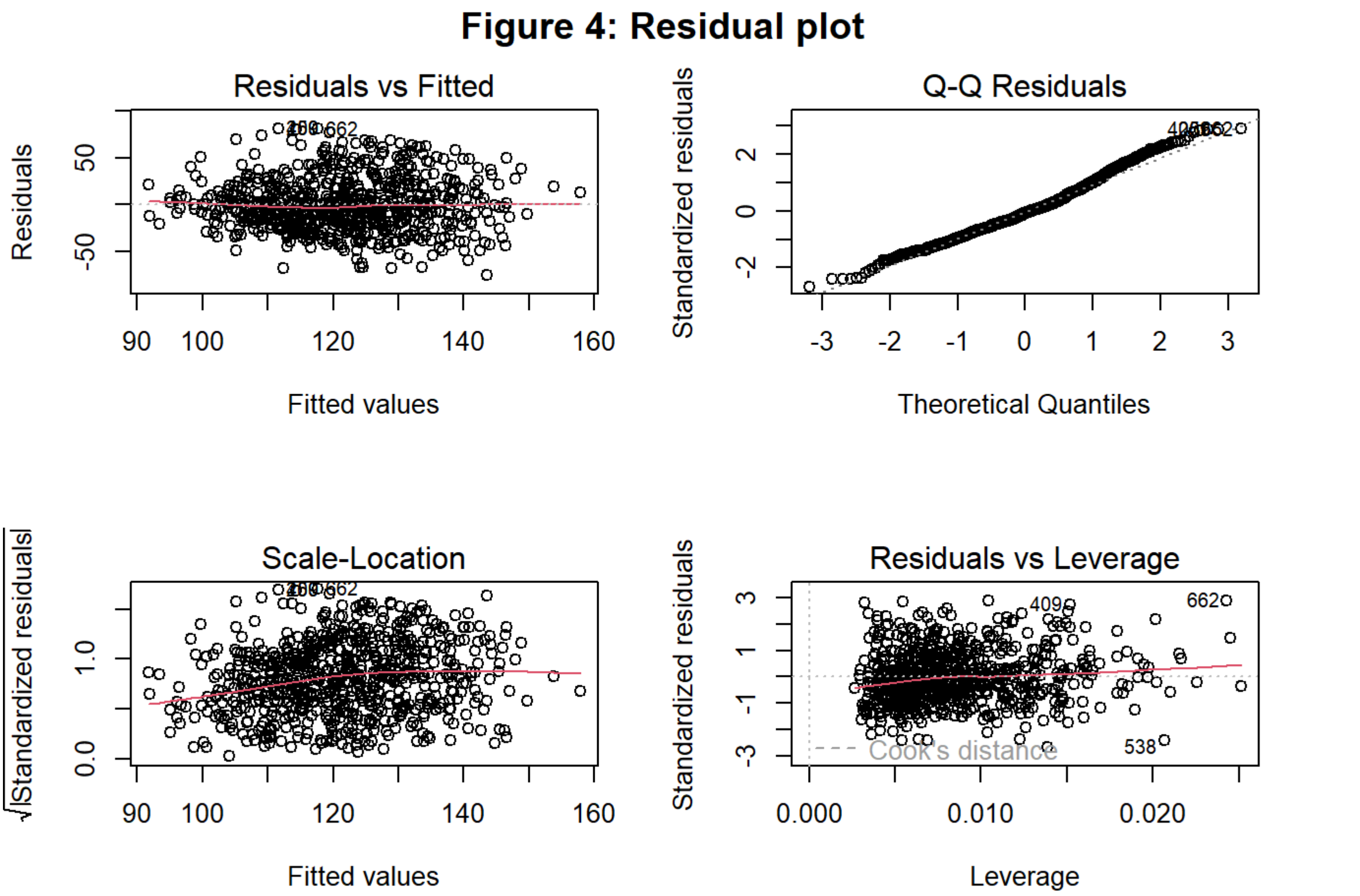
Model A with outliers has MAE value of 23.10, RMSE of 29.74, AIC of 7404.87, r2 of 0.1331, Adjusted R2 of 0.1274 whereas Model B without outliers has MAE value of 22.28, RMSE of 28.26, AIC of 6707.434, r2 of 0.1422, Adjusted R2 of 0.1360. MAE and RMSE are lower in Model B. Predicted value of Glucose are closer to actual values in model B. Lower RMSE suggests fewer extreme prediction errors. Lower the AIC value better is the fit of the model. Model B has lower AIC values compared to that of Model A. Model B has higher Adjusted r2 Values which explains a slightly higher proportion of variance in Glucose. BloodPressure becomes statistically significant in Model B which wasn’t the case in Model A.

So we will go with Model B which is the model without outliers.

$$E(\hat{y}) = -50.8966 + 24.5428x_1 + 7.4154x_2 + 1.9655x_3 + 4.5240x_4 + 4.1206x_5$$

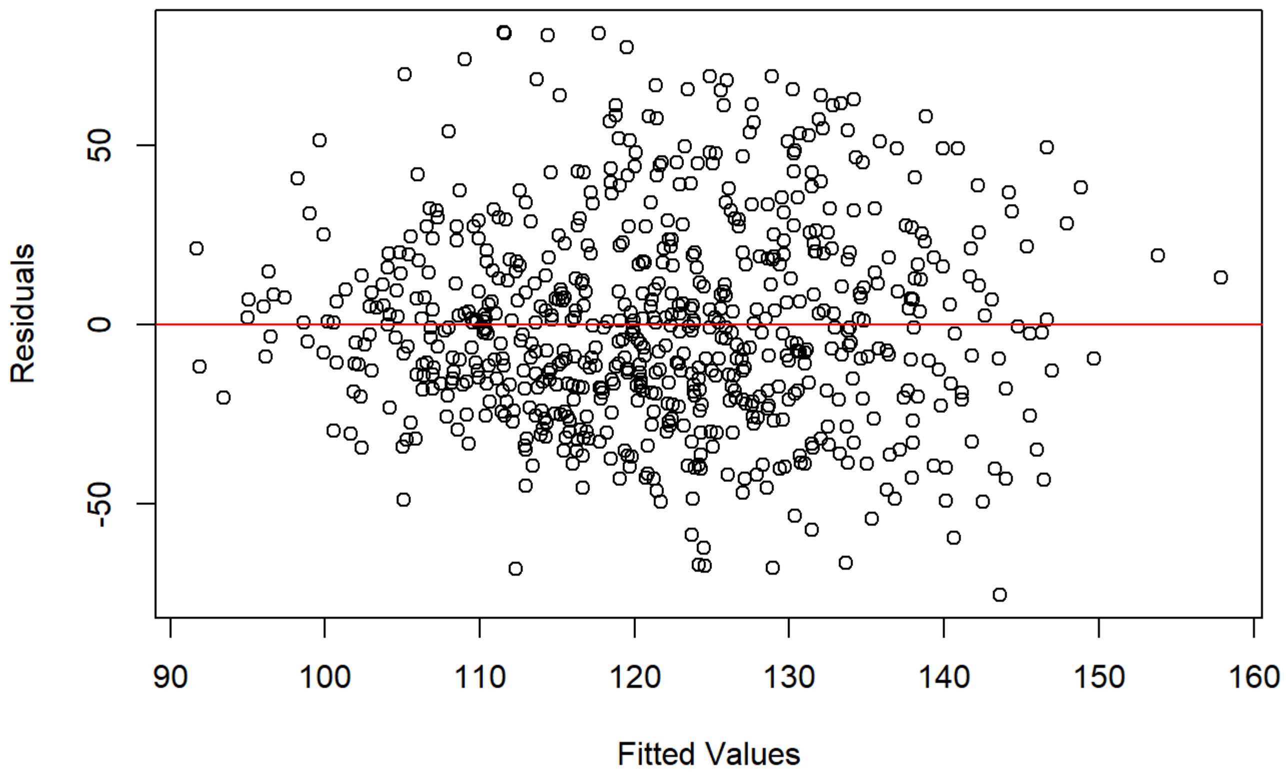
In the Residuals vs Fitted plot of the Residual Plot, we can see that all the plots are randomly scattered around the red line.In QQ-Plot all the points are on the red line, but we could see a slight variation in the top end of the tail.In Scale Location graph, we could see a slight downward trend. In Residuals vs Leverage graph, all the points are on the red line, we will further analyse the outcome of these graph with multiple tests.

```
par(mfrow=c(2,2))
plot(model_without_outliers)
mtext("Figure 4: Residual plot", side = 3, line = -2, outer = TRUE, cex = 1.2, font = 2)
```



```
# Residuals vs Fitted Plot for model_without_outliers
plot(model_without_outliers$fitted.values, model_without_outliers$residuals,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Figure 5: Residuals vs Fitted")
abline(h = 0, col = "red")
```

Figure 5: Residuals vs Fitted



```
# Standardised and Studentised Residuals
standard_res <- rstandard(model_without_outliers)
student_res <- rstudent(model_without_outliers)

# Identifying observations with high residuals
which(abs(student_res) > 2)
```

```
##   3   9  23  41  63  77  93 147 186 221 238 259 261 318 320 328 353 360 361 400
##   3   7  17  35  53  66  80 130 166 200 215 236 238 291 293 301 320 326 327 363
## 409 441 456 463 490 499 538 550 562 580 596 597 648 662 673 676 716 729 738
## 372 401 414 421 446 454 489 501 513 531 545 546 589 603 614 616 651 664 673
```

```
# Cook's Distance
cooks_d <- cooks.distance(model_without_outliers)
plot(cooks_d, type = "h",
     main = "Figure 6: Cook's Distance", ylab = "Cook's Distance")
abline(h = 4 / length(model_without_outliers$fitted.values), col = "red", lty = 2)
```

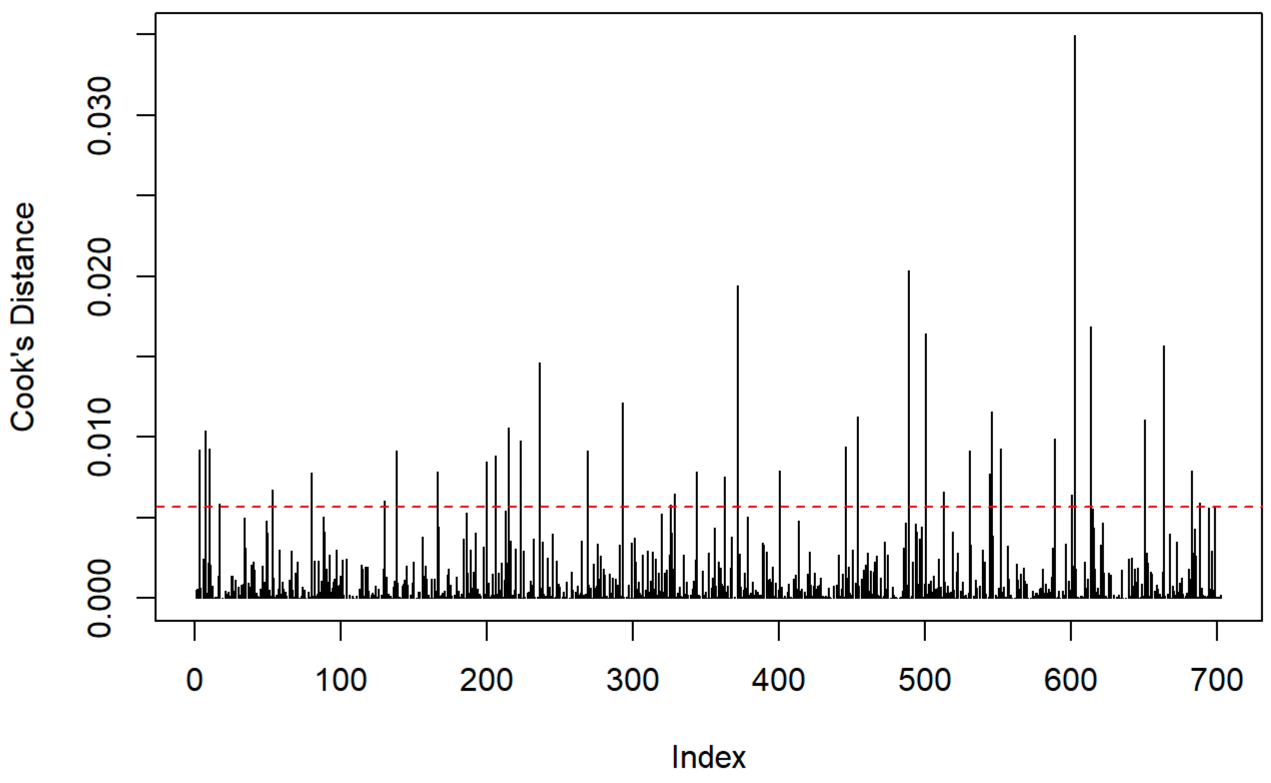


Figure 6: Cook's Distance

```
# Identifying high influence points
which(cooks_d > 4 / length(model_without_outliers$fitted.values))
```

```
##   3   9  14  23  63  93 147 155 186 221 228 238 246 259 295 320 360 363 380 400
##   3   7  10  17  53  80 130 138 166 200 206 215 223 236 269 293 326 329 344 363
## 409 441 490 499 538 550 562 580 596 597 607 648 660 662 673 716 729 748 754 764
## 372 401 446 454 489 501 513 531 545 546 552 589 601 603 614 651 664 683 689 699
```

```
#Leverage (Hat) Values
hat_vals <- hatvalues(model_without_outliers)

# Plotting leverage values
plot(hat_vals, type = "h",
     main = "Figure 7: Leverage Values", ylab = "Leverage")
abline(h = 2 * mean(hat_vals), col = "red", lty = 2)
```

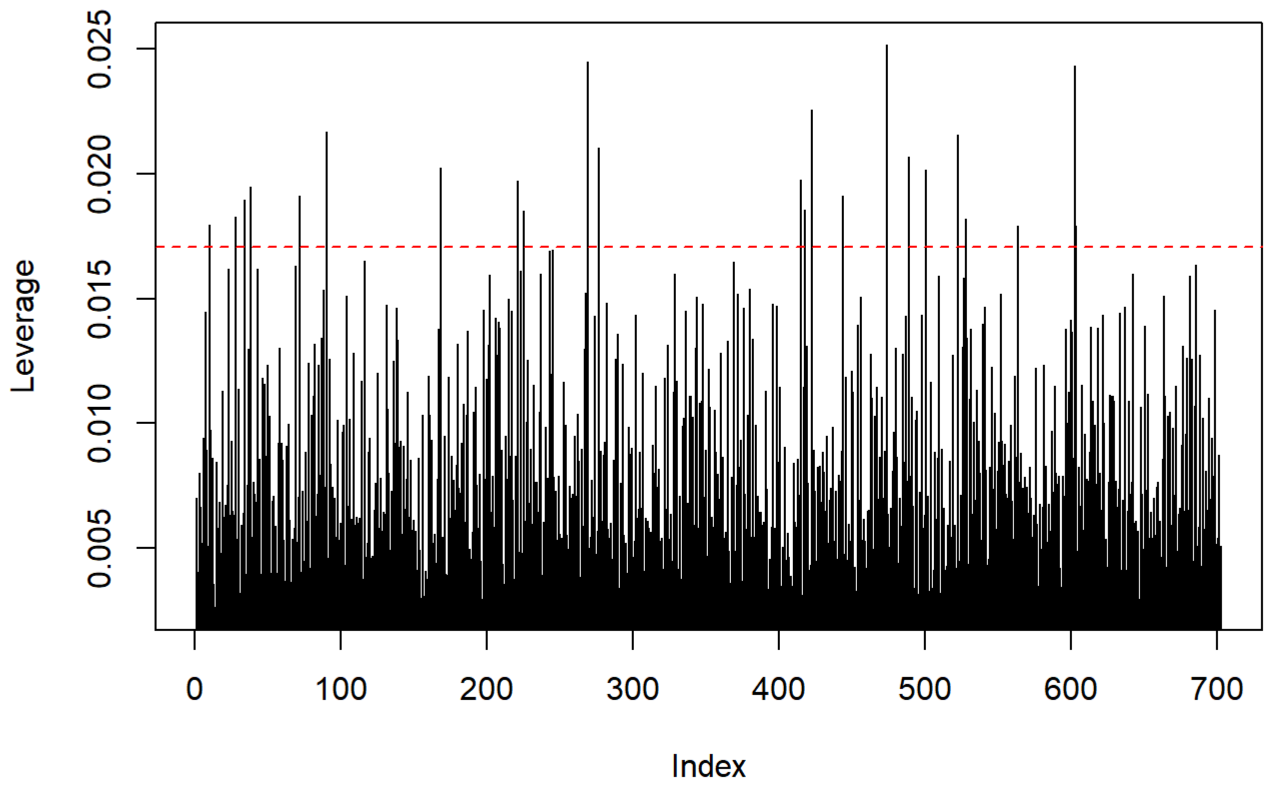


Figure 7: Leverage Values

```
# Identifying high leverage points
high_leverage <- which(hat_vals > 2 * mean(hat_vals))
print("High leverage points:")
```

```
## [1] "High leverage points:"
```

```
print(high_leverage)
```

```
##  14  34  40  44  85 103 188 244 248 295 304 457 460 465 488 520 538 550 572 577
##  10  28  34  38  72  90 168 221 225 269 277 415 418 423 444 474 489 501 523 528
## 619 662 663
## 564 603 604
```

```
# Combining all unique influential indices
final_influential_points <- unique(c(
  which(abs(student_res) > 2),
  which(cooks_d > 4 / length(model_without_outliers$fitted.values)),
  which(hat_vals > 2 * mean(hat_vals))
))

length(final_influential_points)
```

```
## [1] 69
```

Points with studentized residuals greater than 2 are considered to be outliers, we have some points above the threshold Cook’s Distance shows how much the model’s prediction would change if we remove a point. The points where Cook’s D>4/n are considered influential points, they all will impact our regression coefficient. Leverage measures how far an observation’s predictor values are from a mean of all predictor values.Points with hat value >2 * average hat value are considered high leverage. Finally in final_influential_points, we are combining all the observations we got from various influential points approach, we could see a total of 69 observations.

We are going to remove all the 69 observations as they have a great impact on our model.

```
# Creating a cleaned dataset by removing influential points
df_hat_removed_final <- df_clean[-final_influential_points, ]

# Fitting the final model
model_final <- lm(Glucose ~ Age + BMI + Insulin + BloodPressure + DiabetesPedigreeFunction, d
ata = df_hat_removed_final)

# Summary of the model
summary(model_final)
```

```
##
## Call:
## lm(formula = Glucose ~ Age + BMI + Insulin + BloodPressure +
##     DiabetesPedigreeFunction, data = df_hat_removed_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.462 -17.329  -1.221  15.848  57.417
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -66.1183    15.3191  -4.316 1.84e-05 ***
## Age                       26.7340     3.4082   7.844 1.88e-14 ***
## BMI                        7.3335     1.8072   4.058 5.58e-05 ***
## Insulin                    2.1331     0.4069   5.242 2.17e-07 ***
## BloodPressure              5.4367     1.6756   3.245  0.00124 **
## DiabetesPedigreeFunction  -2.4502     5.6604  -0.433  0.66526
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.62 on 628 degrees of freedom
## Multiple R-squared:  0.208,  Adjusted R-squared:  0.2017
## F-statistic:    33 on 5 and 628 DF,  p-value: < 2.2e-16
```

$$E(\hat{y}) = -66.1183 + 26.7340x_1 + 7.3335x_2 + 2.1331x_3 + 5.4367x_4 - 2.4502x_5$$

Hypothesis for Wald Statistics: H0: The predictor has no effect. HA: Predictor has significant effect

Hypothesis for Anova

H0: Variable has no effect on response HA: Variable has effect on response

From the Wald Statistics, Anova and by checking the p-value from the summary of the model, we can see that Age, BMI, Insulin, Bloodpressure have significant effect on the model because their p-value is less than 0.05 whereas DiabetesDegreeFunction has a p-value greater than 0.05 which inturn shows that it has no effect on the model.Except DiabetesDegreeFunction, rest of the predictors donot have their value in 0 for Wald Statistic, hence we reject null hypothesis for Wald Statistics and for Anova, the predictors Age, BMI, Insulin, Bloodpressure have p-value less than 0.05, we reject H0 for Anova and for DiabetesDegreeFunction p-value is greater than 0.05, we fail to reject H0. Therefore Age, BMI, Insulin, Bloodpressure are the ones which are significant predictors.

So model_final is the model without these influential points, we will check this model with other 2 models, i.e Model with outliers and model without outliers and see which model performs best in terms of low AIC value, low MAE values and lower RMSE values.After going through the results, we could see that model_final has lowest AIC, MAE and RMSE values, so we will take this model for further analysis.

```
# ANOVA
anova(model_final)
```

```
## Analysis of Variance Table
##
## Response: Glucose
##                           Df Sum Sq Mean Sq F value    Pr(>F)
## Age                        1  51608   51608 92.5075 < 2.2e-16 ***
## BMI                        1  20242   20242 36.2832 2.906e-09 ***
## Insulin                    1  14179   14179 25.4150 6.056e-07 ***
## BloodPressure              1   5904    5904 10.5829  0.001203 **
## DiabetesPedigreeFunction   1    105     105  0.1874  0.665263
## Residuals                628 350350     558
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Getting coefficients and Standard errors
coefs1 <- summary(model_final)$coefficients

# Computing Wald statistics
wald_stats <- (coefs1[, "Estimate"] / coefs1[, "Std. Error"])^2

# Computing p-values from Chi-squared distribution with df = 1
wald_pvals <- 1 - pchisq(wald_stats, df = 1)

wald_results <- data.frame(
  Estimate = coefs1[, "Estimate"],
  StdError = coefs1[, "Std. Error"],
  WaldStatistic = round(wald_stats, 3),
  p_value = round(wald_pvals, 4),
  Significance = ifelse(wald_pvals < 0.05, "Significant", "Not Significant")
)

print(wald_results)
```

```
##                             Estimate   StdError WaldStatistic p_value
## (Intercept)               -66.118268 15.3190532        18.629  0.0000
## Age                        26.734043  3.4081815        61.530  0.0000
## BMI                         7.333481  1.8071988        16.467  0.0000
## Insulin                     2.133090  0.4068991        27.482  0.0000
## BloodPressure               5.436723  1.6755971        10.528  0.0012
## DiabetesPedigreeFunction  -2.450180  5.6604147         0.187  0.6651
##                             Significance
## (Intercept)                  Significant
## Age                          Significant
## BMI                          Significant
## Insulin                      Significant
## BloodPressure                Significant
## DiabetesPedigreeFunction Not Significant
```

```
# Function to calculate performance metrics
model_metrics <- function(model, data) {
  preds <- predict(model, data)
  actual <- data$Glucose
  mae <- mean(abs(preds - actual))
  rmse <- sqrt(mean((preds - actual)^2))
  aic <- AIC(model)
  return(c(MAE = mae, RMSE = rmse, AIC = aic))
}

# Computing metrics
metrics_with_outliers <- model_metrics(model_with_outliers, df_transformed)
metrics_without_outliers <- model_metrics(model_without_outliers, df_clean)
metrics_final <- model_metrics(model_final, df_hat_removed_final)

comparison <- rbind(
  With_Outliers = metrics_with_outliers,
  Without_Outliers = metrics_without_outliers,
  Final_Cleaned = metrics_final
)

print(round(comparison, 2))
```

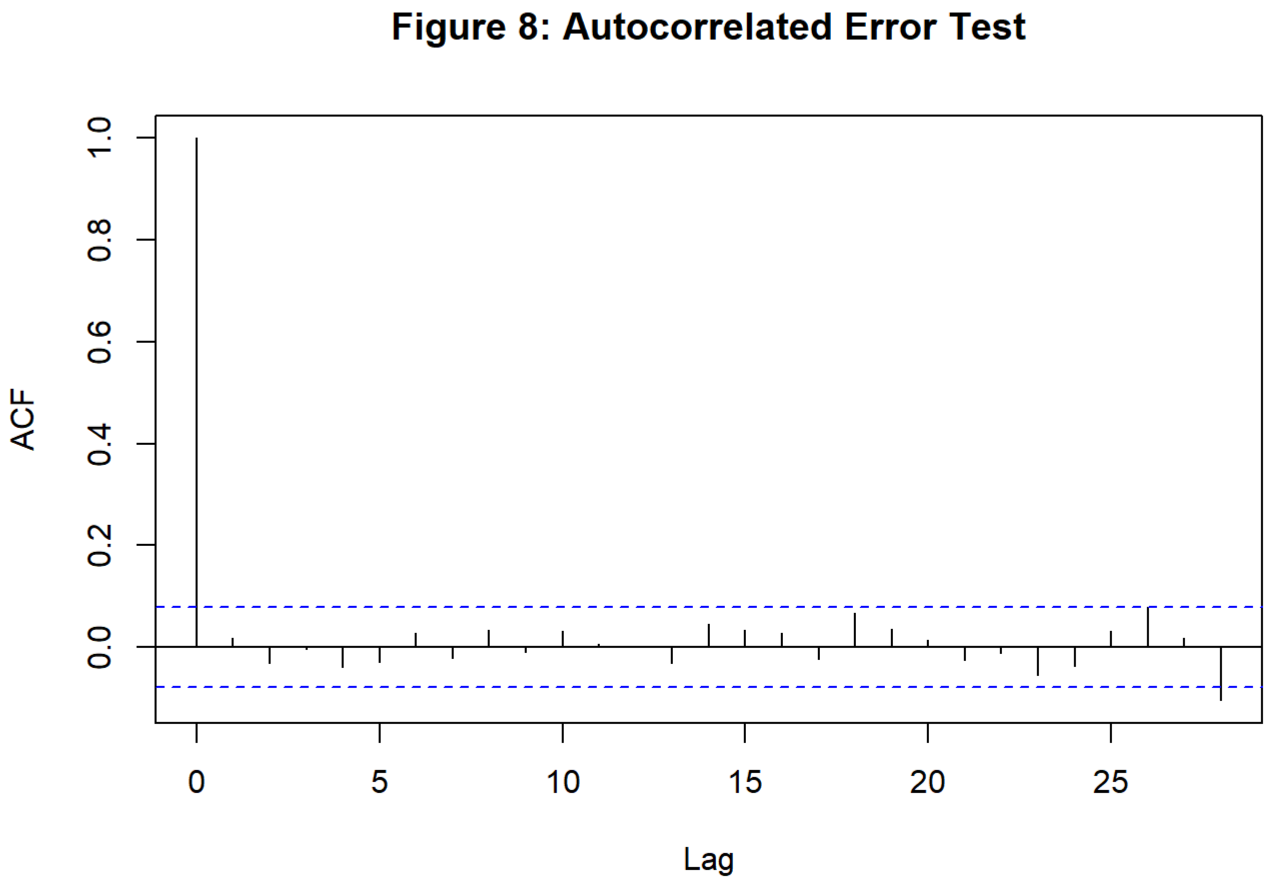
```
##                    MAE  RMSE     AIC
## With_Outliers    23.11 29.75 7404.87
## Without_Outliers 22.28 28.27 6707.43
## Final_Cleaned    19.11 23.51 5816.69
```

Durbin-Watson Test

Hypothesis test: H0:No autocorrelation HA:There is autocorrelation

Since DW is close to 2 and p-value is lesser than 0.05, we fail to reject H0.No correlation detected.

```
acf(model_final$residuals, main = "Figure 8: Autocorrelated Error Test")
```



```
dwtest(model_final)
```

```
##
##  Durbin-Watson test
##
## data:  model_final
## DW = 1.9637, p-value = 0.3241
## alternative hypothesis: true autocorrelation is greater than 0
```

Breusch-Pagan Test (Heteroskedasticity)

Hypothesis test: H0:Variance is constant HA:Variance is not constant

BP = 26.814 and p-value lesser than 0.05, we reject H0. There is evidence of heteroskedasticity.

```
bptest(model_final)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model_final
## BP = 26.814, df = 5, p-value = 6.2e-05
```

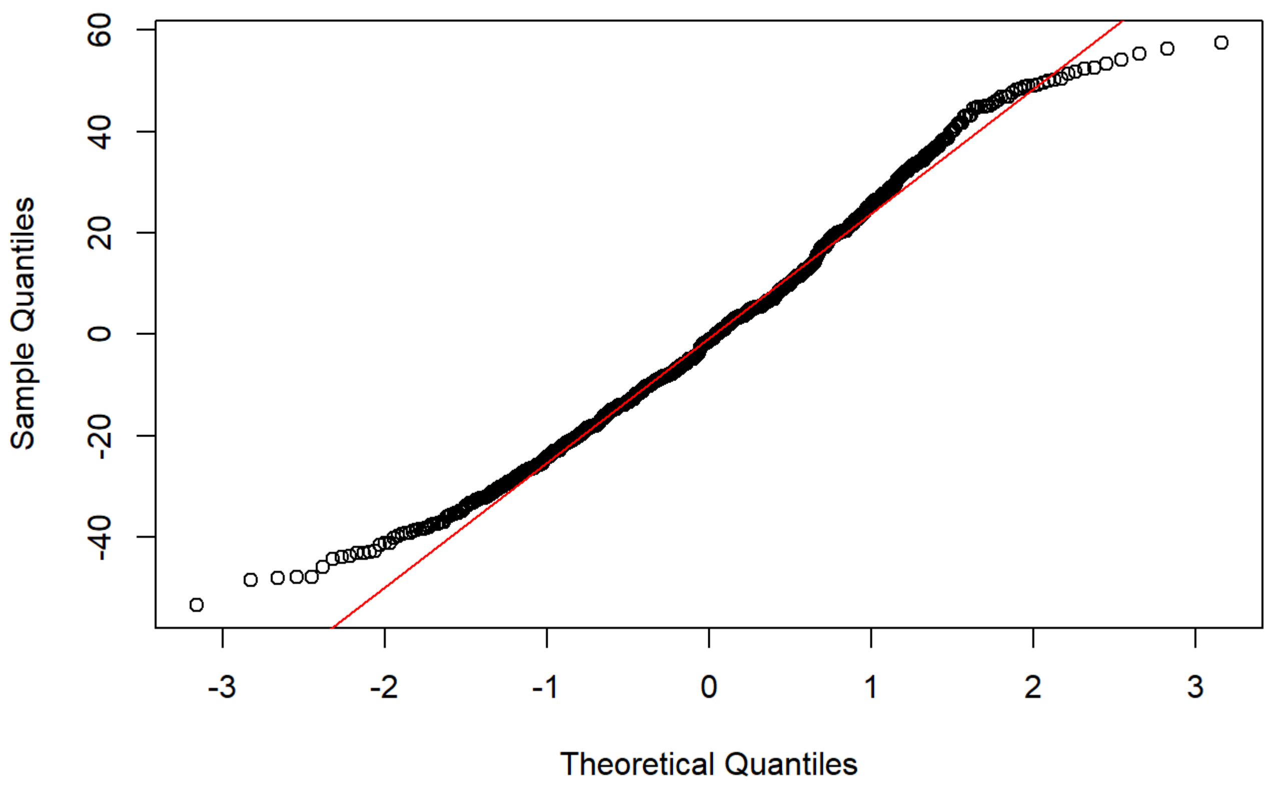
Shapiro-Wilk Test

Hypothesis test: H0:Normal distribution is followed by residuals HA:No normality

Since p-value is lesser than 0.05, we reject H0. Residuals are not normally distributed.

```
# Q-Q Plot
qqnorm(residuals(model_final), main = "Figure 9: Q-Q Plot of Residuals")
qqline(residuals(model_final), col = "red")
```

Figure 9: Q-Q Plot of Residuals



```
# Shapiro-Wilk Test
shapiro.test(residuals(model_final))
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  residuals(model_final)
## W = 0.98556, p-value = 6.472e-06
```

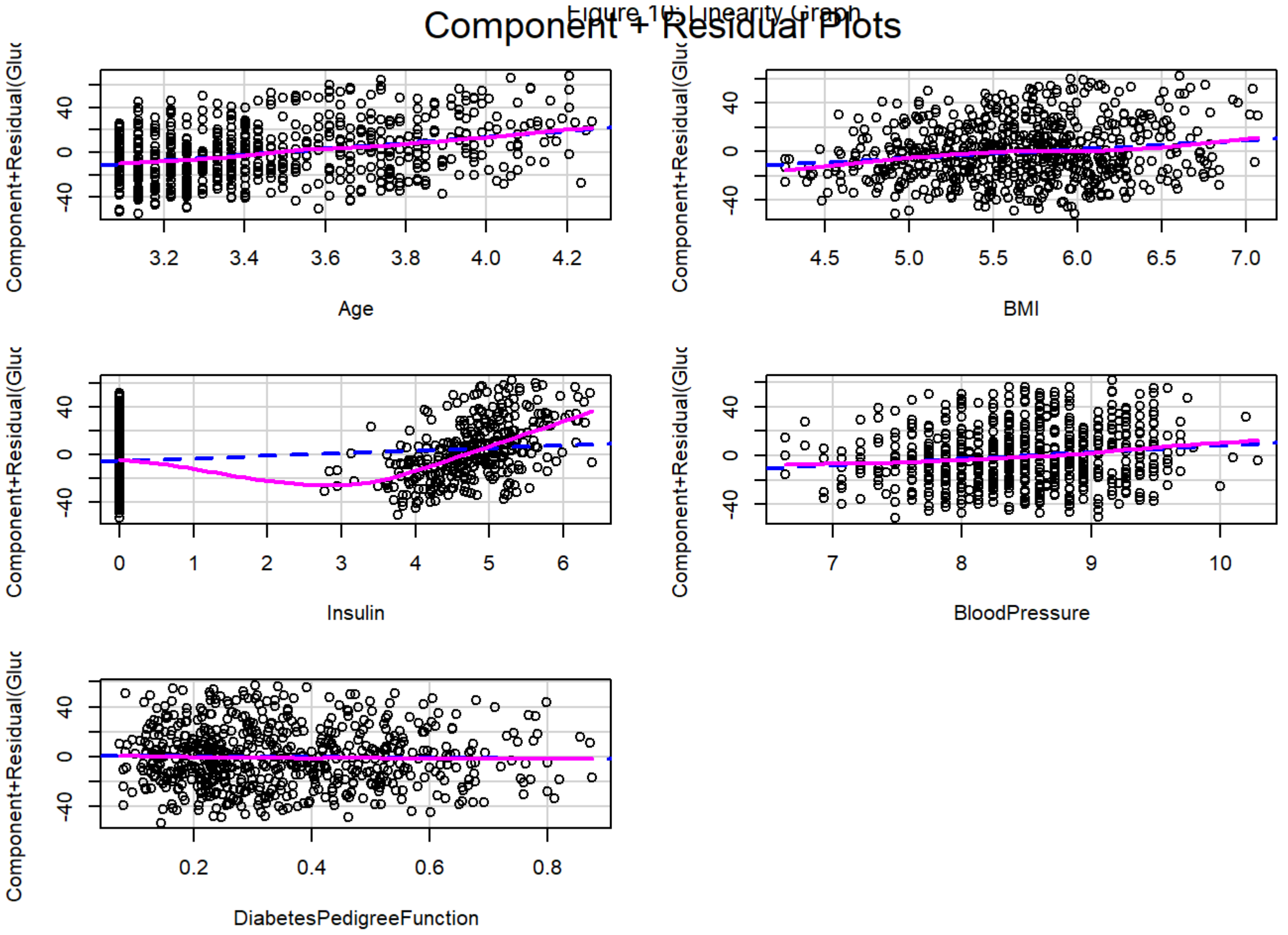
From VIF test, we can see that no predictors have value more than 5. Hence, there is no multicollinearity present

```
# VIF (Multicollinearity)
vif(model_final)
```

```
##                      Age                      BMI                  Insulin 
##                 1.237229                 1.146847                 1.111801 
##            BloodPressure DiabetesPedigreeFunction 
##                 1.317437                 1.066609
```

For Insulin alone, we could see that the pink curve is not passing through the blue line, but for remaining predictors, we can clearly see that they are all aligned properly.

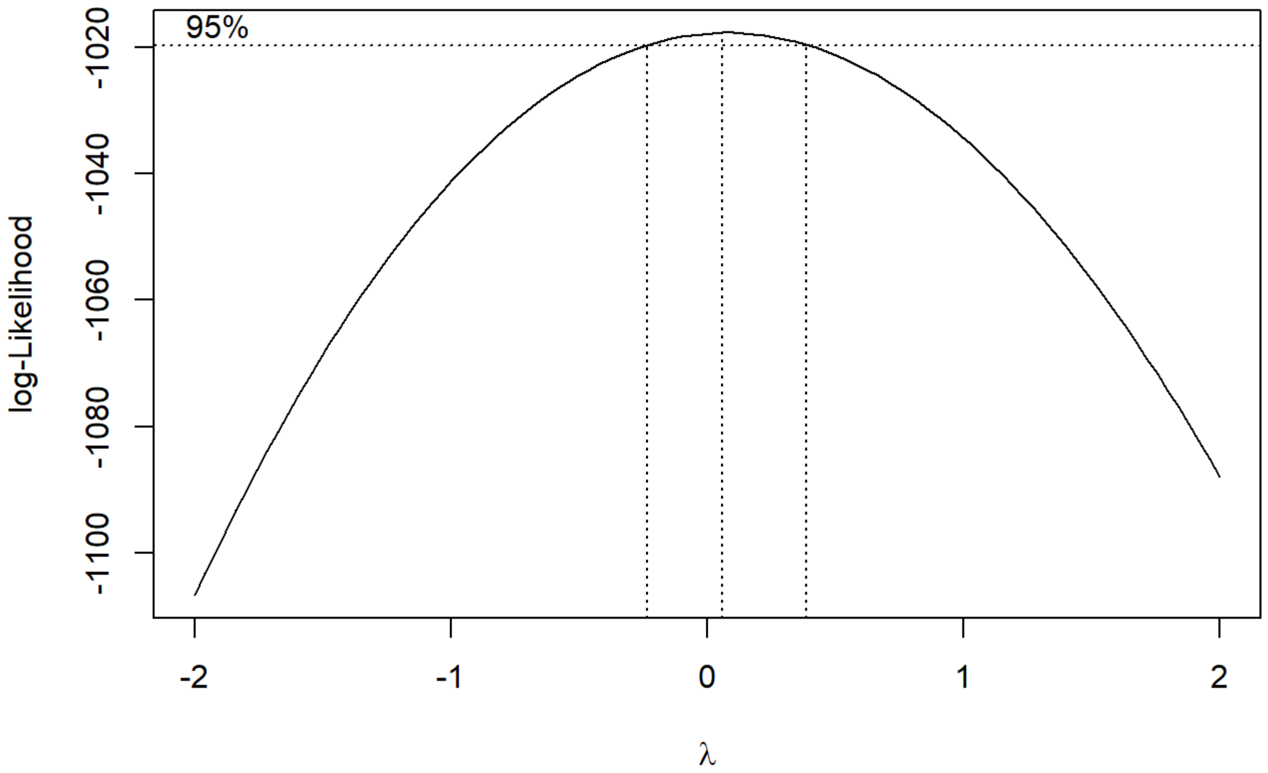
```
# Linearity
car::crPlots(model_final)
mtext("Figure 10: Linearity Graph", side=3, line=3.5, cex=0.8, font=0.5)
```



Since we did not achieve normality and to remove heteroskedasticity., we are going to transform our model by Box-Cox transformation and see that whether we are able to achieve normality.

From Box-cox transformation, we can see that the optimal Lambda is 0.06 which is very close to 0, so we will apply log transformation to response variable Glucose and then we will fit a new model and perform all the model diagnostics.

```
# Applying Box-Cox transformation on the model
boxcox_result <- boxcox(model_final, lambda = seq(-2, 2, 0.1))
```



```
# Finding the lambda with maximum log-likelihood
lambda_optimal <- boxcox_result$x[which.max(boxcox_result$y)]
cat("Optimal lambda:", lambda_optimal, "\n")
```

```
## Optimal lambda: 0.06060606
```

```
# Transforming the response variable based on the optimal lambda
if (abs(lambda_optimal) < 1e-4) {
  df_transformed <- df_hat_removed_final
  df_transformed$Glucose_transformed <- log(df_transformed$Glucose)
} else {
  df_transformed <- df_hat_removed_final
  df_transformed$Glucose_transformed <- (df_transformed$Glucose^lambda_optimal - 1) / lambda_
optimal
}
```

```
model_boxcox <- lm(Glucose_transformed ~ Age + BMI + Insulin + BloodPressure + DiabetesPedigr
eeFunction,
                   data = df_transformed)
summary(model_boxcox)
```

```
## 
## Call:
## lm(formula = Glucose_transformed ~ Age + BMI + Insulin + BloodPressure + 
##     DiabetesPedigreeFunction, data = df_transformed)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.76245 -0.17789  0.00874  0.19255  0.59476 
## 
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)    
## (Intercept)               3.458285   0.171850  20.124  < 2e-16 ***
## Age                       0.289876   0.038233   7.582 1.23e-13 ***
## BMI                       0.081488   0.020273   4.020 6.54e-05 ***
## Insulin                   0.022829   0.004565   5.001 7.40e-07 ***
## BloodPressure             0.063187   0.018797   3.362 0.000822 ***
## DiabetesPedigreeFunction -0.024459   0.063499  -0.385 0.700231    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.265 on 628 degrees of freedom
## Multiple R-squared:  0.2023, Adjusted R-squared:  0.1959 
## F-statistic: 31.85 on 5 and 628 DF,  p-value: < 2.2e-16
```

```
# ANOVA
anova(model_boxcox)
```

```
## Analysis of Variance Table
## 
## Response: Glucose_transformed
##                           Df Sum Sq Mean Sq F value    Pr(>F)    
## Age                        1  6.239  6.2393 88.8704 < 2.2e-16 ***
## BMI                        1  2.519  2.5190 35.8797 3.535e-09 ***
## Insulin                    1  1.615  1.6148 23.0013 2.024e-06 ***
## BloodPressure              1  0.797  0.7969 11.3515    0.0008 ***
## DiabetesPedigreeFunction   1  0.010  0.0104  0.1484    0.7002    
## Residuals                628 44.089  0.0702                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Getting coefficients and Standard errors
coefs2 <- summary(model_boxcox)$coefficients

# Computing Wald statistics
wald_stats1 <- (coefs2[, "Estimate"] / coefs2[, "Std. Error"])^2

# Computing p-values from Chi-squared distribution with df = 1
wald_pvals1 <- 1 - pchisq(wald_stats1, df = 1)

wald_results1 <- data.frame(
  Estimate1 = coefs2[, "Estimate"],
  StdError1 = coefs2[, "Std. Error"],
  WaldStatistic1 = round(wald_stats1, 3),
  p_value1 = round(wald_pvals1, 4),
  Significance1 = ifelse(wald_pvals1 < 0.05, "Significant", "Not Significant")
)

print(wald_results1)
```

```
##                             Estimate1   StdError1 WaldStatistic1 p_value1
## (Intercept)                3.45828485 0.171849546        404.972   0.0000
## Age                        0.28987648 0.038233071         57.484   0.0000
## BMI                        0.08148847 0.020273205         16.157   0.0001
## Insulin                    0.02282918 0.004564605         25.013   0.0000
## BloodPressure              0.06318721 0.018796893         11.300   0.0008
## DiabetesPedigreeFunction -0.02445870 0.063498683          0.148   0.7001
##                            Significance1
## (Intercept)                  Significant
## Age                          Significant
## BMI                          Significant
## Insulin                      Significant
## BloodPressure                Significant
## DiabetesPedigreeFunction Not Significant
```

$$E(\hat{y}) = 3.458285 + 0.289876x_1 + 0.081488x_2 + 0.022829x_3 + 0.063187x_4 - 0.024459x_5$$

Hypothesis for Wald Statistics: H0: The predictor has no effect. HA: Predictor has significant effect

Hypothesis for Anova

H0: Variable has no effect on response HA: Variable has effect on response

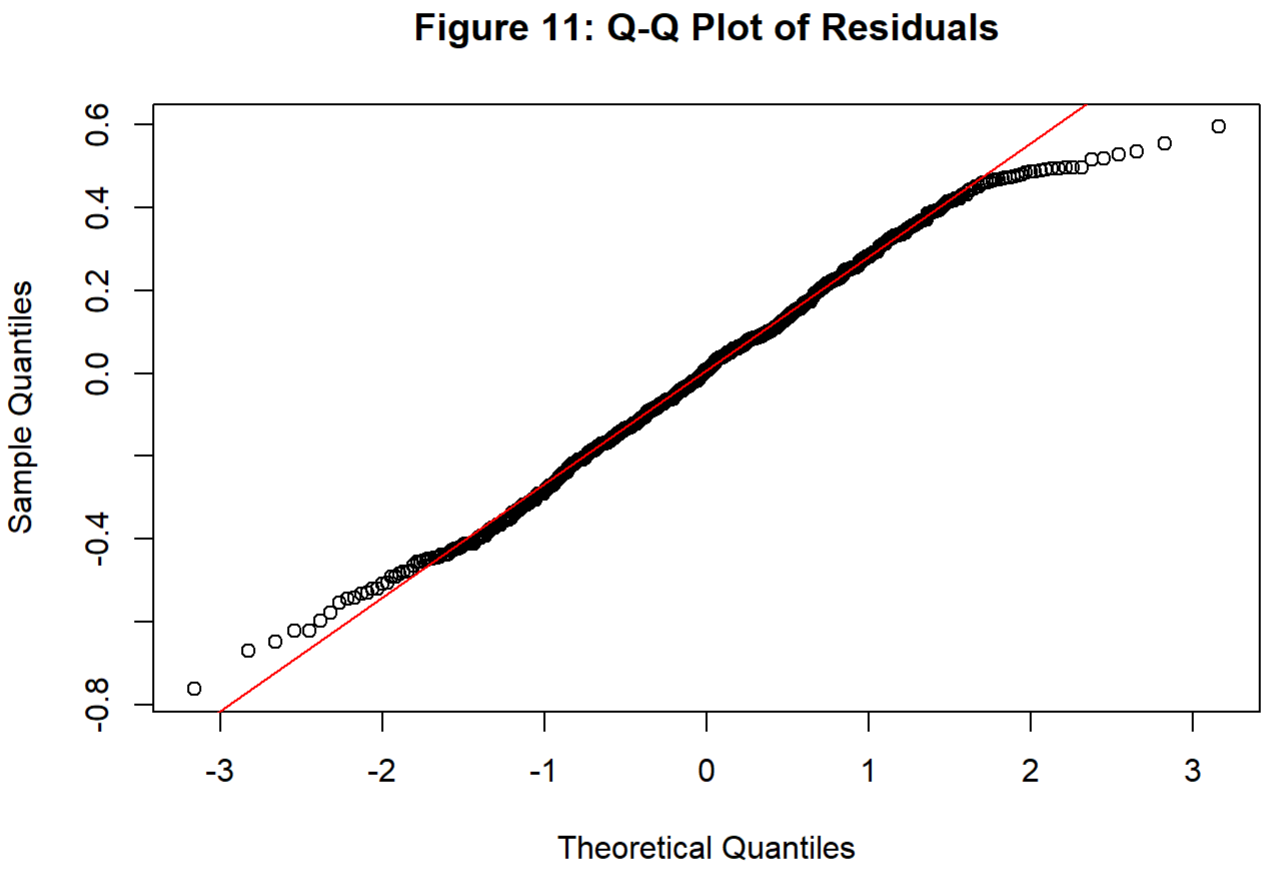
From the Wald Statistics, Anova and by checking the p-value from the summary of the model, we can see that Age, BMI, Insulin, Bloodpressure has significant effects on the model, this is same like model_final, which is the non-transformed model because they have their p-value less than 0.05 whereas DiabetesDegreeFunction has p-value greater than 0.05 and has no effect on the model.Except DiabetesDegreeFunction in this model also, rest of the predictors donot have their value in 0 for Wald Statistic, hence we reject null hypothesis for Wald Statistics and for Anova, the predictors Age, BMI, Insulin, Bloodpressure have p-value less than 0.05, we reject H0 for them and for DiabetesDegreeFunction p-value is greater than 0.05, we fail to reject H0. Therefore Age, BMI, Insulin, Bloodpressure are the ones which are significant predictors in this model too. In Box-Cox model p-value of DiabetesDegreeFunction has slightly increased compared to that of the non-transformed model.

Shapiro-Wilk Test

Hypothesis test: H0:Normal distribution is followed by residuals HA:No Normality

Since p-value is lesser than 0.05, we reject H0. Residuals are not normally distributed.But we can see a increase in p-value compared to non-transformed data.There is not much change in the QQ-plot where both the tails did not fit the points.But we could see a slight improvement in BC transformed model.

```
# Normality
# Q-Q Plot
qqnorm(residuals(model_boxcox), main = "Figure 11: Q-Q Plot of Residuals")
qqline(residuals(model_boxcox), col = "red")
```



```
shapiro.test(residuals(model_boxcox))
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  residuals(model_boxcox)
## W = 0.99129, p-value = 0.0008691
```

Breusch-Pagan Test (Heteroskedasticity)

Hypothesis test: H0:Variance is constant HA:Variance is not constant

p-value is higher than 0.05, we failed to reject H0. There is no evidence of heteroskedasticity. After transformation heteroskedasticity is removed.

```
# Heteroscedasticity
bptest(model_boxcox)
```

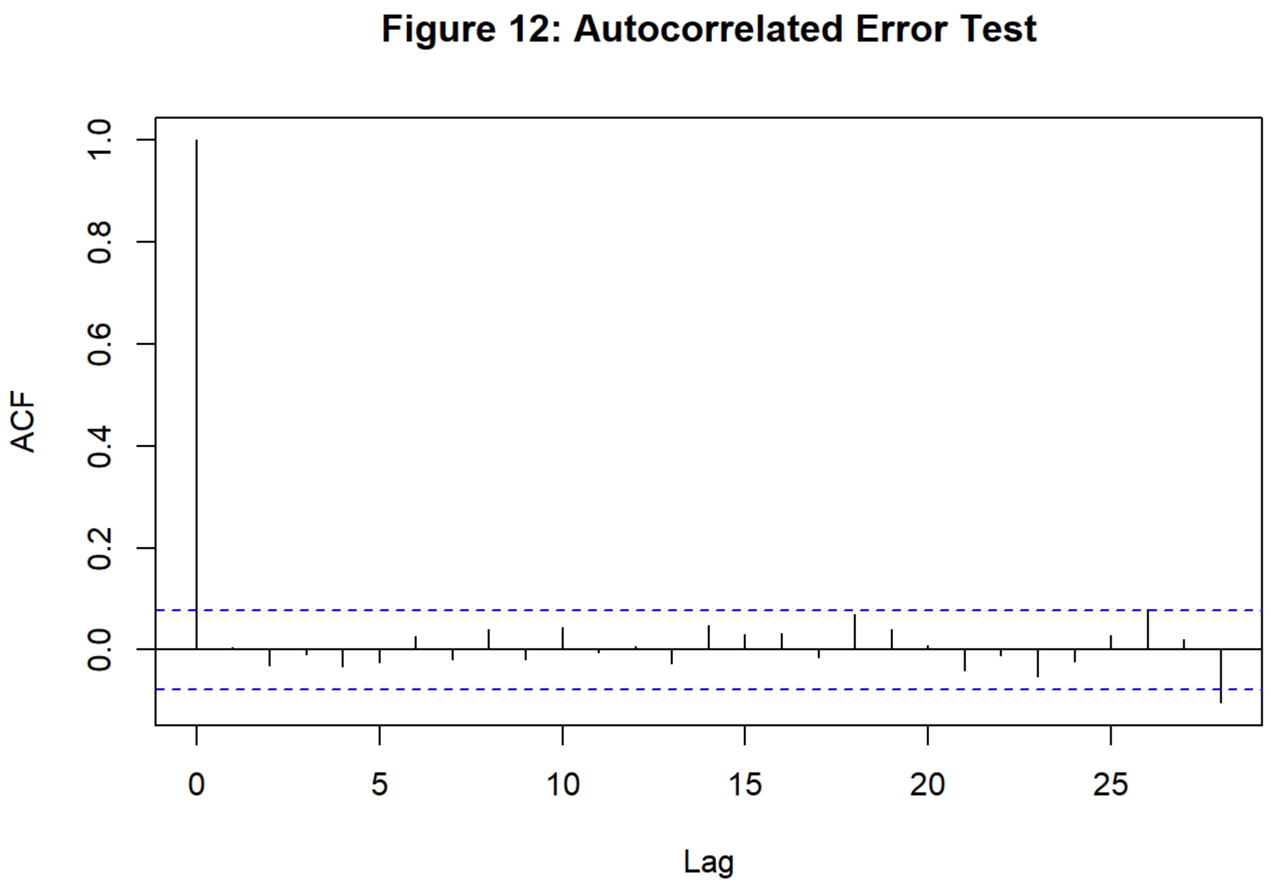
```
## 
##  studentized Breusch-Pagan test
## 
## data:  model_boxcox
## BP = 6.6574, df = 5, p-value = 0.2474
```

Durbin-Watson Test

Hypothesis test: H0:No autocorrelation HA:There is autocorrelation

DW is now very close to 2 and p-value is greater than 0.05, we fail to reject H0.No correlation detected.

```
# Independence
acf(model_boxcox$residuals, main = "Figure 12: Autocorrelated Error Test")
```



```
dwtest(model_boxcox)
```

```
## 
##  Durbin-Watson test
## 
## data:  model_boxcox
## DW = 1.9883, p-value = 0.4418
## alternative hypothesis: true autocorrelation is greater than 0
```

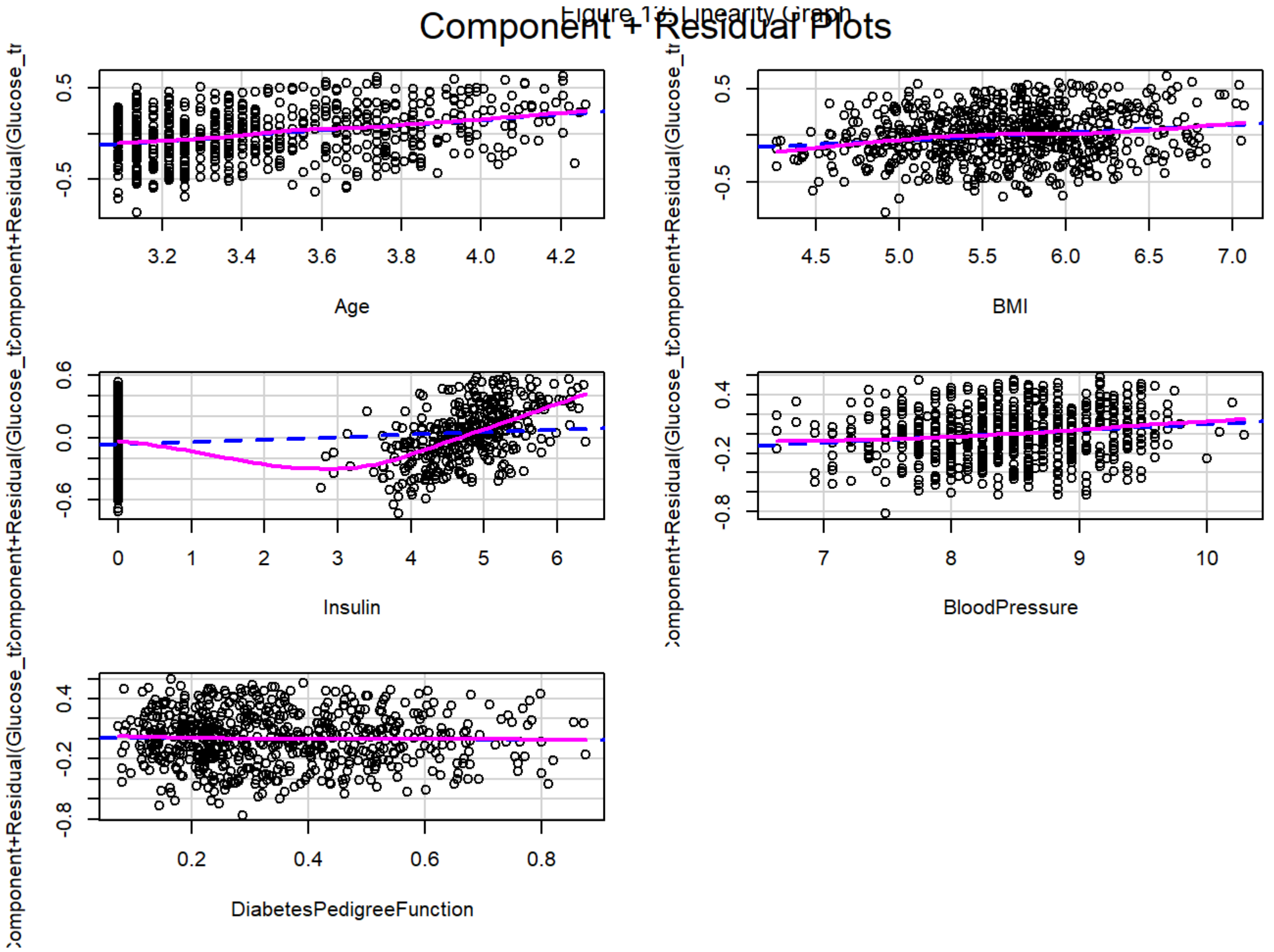
No predictor is above 5, so there is no multicollinearity.

```
#Multicollinearity
vif(model_boxcox)
```

```
##                      Age                      BMI                  Insulin 
##                 1.237229                 1.146847                 1.111801 
##            BloodPressure DiabetesPedigreeFunction 
##                 1.317437                 1.066609
```

Linearity has improved a lot for the BC-Transformed model. Insulin’s value has reduced considerably and both the pink curve and the blue dotted line are very close to each other.

```
# Linearity
car::crPlots(model_boxcox)
mtext("Figure 13: Linearity Graph", side=3, line=3.5, cex=0.8, font=0.5)
```



Since Normality is still not achieved after the Box-Cox Transformation, we will fit robust model which is our second model for this analysis and see whether we are able to achieve normality for this model.

```
# Fitting robust regression on the transformed dataset
robust_model <- rlm(Glucose_transformed ~ Age + BMI + Insulin + BloodPressure + DiabetesPedig
reeFunction,
                    data = df_transformed)

# Summary of the robust model
summary(robust_model)
```

```
##
## Call: rlm(formula = Glucose_transformed ~ Age + BMI + Insulin + BloodPressure +
##     DiabetesPedigreeFunction, data = df_transformed)
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.763921 -0.182059  0.004096  0.192065  0.593921
##
## Coefficients:
##                          Value   Std. Error t value
## (Intercept)               3.4426  0.1873    18.3837
## Age                       0.2907  0.0417     6.9785
## BMI                       0.0777  0.0221     3.5166
## Insulin                   0.0233  0.0050     4.6902
## BloodPressure             0.0676  0.0205     3.3010
## DiabetesPedigreeFunction -0.0313  0.0692    -0.4524
##
## Residual standard error: 0.2748 on 628 degrees of freedom
```

```
# Getting coefficients matrix (no column names)
coefs3 <- summary(robust_model)$coefficients

# Assigning column names manually
colnames(coefs3) <- c("Estimate", "Std.Error", "t.value")

# Computing Wald statistics
wald_stats2 <- (coefs3[, "Estimate"] / coefs3[, "Std.Error"])^2

# Computing p-values from Chi-squared distribution with df = 1
wald_pvals2 <- 1 - pchisq(wald_stats2, df = 1)

# Compiling results
wald_results2 <- data.frame(
  Estimate2 = coefs3[, "Estimate"],
  StdError2 = coefs3[, "Std.Error"],
  WaldStatistic2 = round(wald_stats2, 3),
  p_value2 = round(wald_pvals2, 4),
  Significance2 = ifelse(wald_pvals2 < 0.05, "Significant", "Not Significant")
)

print(wald_results2)
```

```
##                            Estimate2   StdError2 WaldStatistic2 p_value2
## (Intercept)               3.44264210 0.187265704        337.962   0.0000
## Age                       0.29074285 0.041662857         48.699   0.0000
## BMI                       0.07768894 0.022091859         12.367   0.0004
## Insulin                   0.02332931 0.004974083         21.998   0.0000
## BloodPressure             0.06761507 0.020483112         10.897   0.0010
## DiabetesPedigreeFunction -0.03130147 0.069194978          0.205   0.6510
##                            Significance2
## (Intercept)                  Significant
## Age                          Significant
## BMI                          Significant
## Insulin                      Significant
## BloodPressure                Significant
## DiabetesPedigreeFunction Not Significant
```

$E(\hat{y}) = 3.4426 + 0.2907x_1 + 0.0777x_2 + 0.0233x_3 + 0.0676x_4 - 0.0313x_5$

Hypothesis for Wald Statistics: H0: The predictor has no effect. HA: Predictor has significant effect

Age,BMI,Insulin and BloodPressure are significant predictors and DiabetesPedigreeFunction is not significant. We got this from the Wald Statistics.There is no Anova calcualtion for this model.This model’s summary does not contain p-values, we manually calculated this in the Wald statistics section.

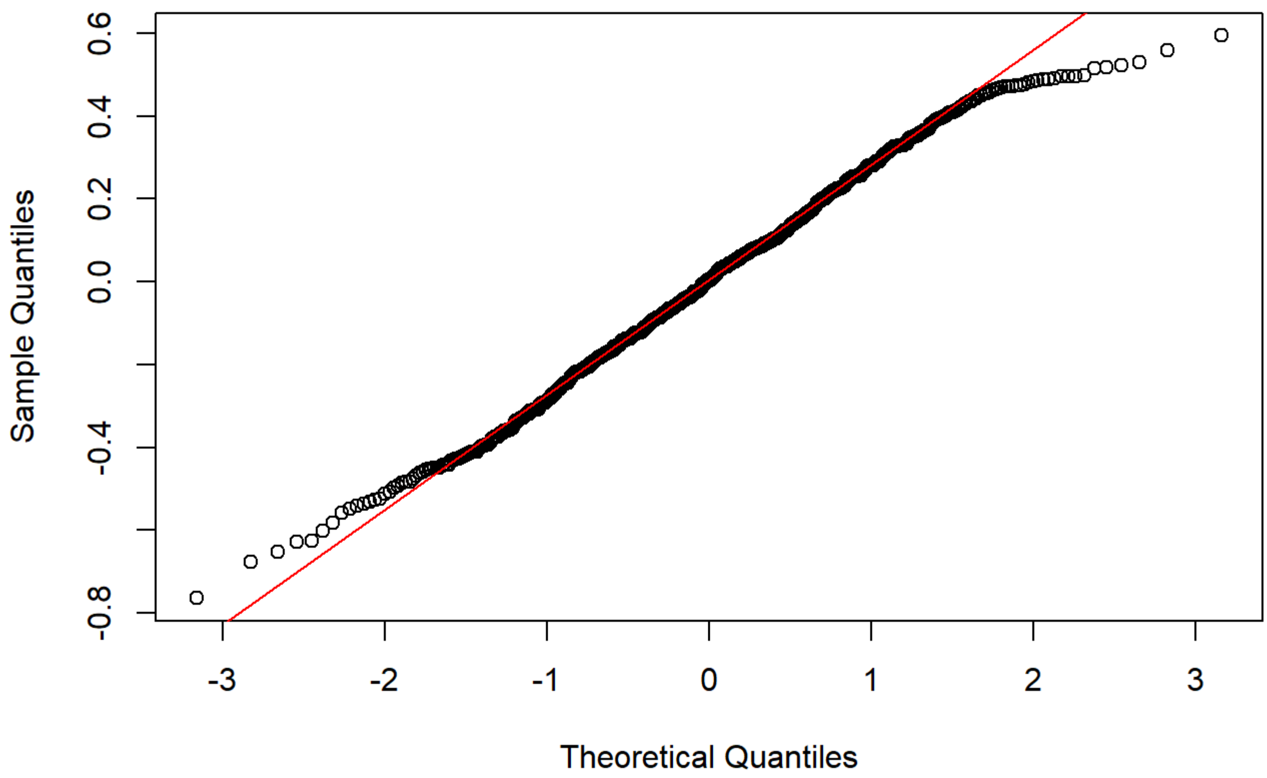
Shapiro-Wilk Test

Hypothesis test: H0:Normal distribution is followed by residuals HA:No Normality

Since p-value is lesser than 0.05, we reject H0. Residuals are not normally distributed.But we can see a increase in p-value compared to Box-Cox model.There is a slight deviation from normality. Because of the large sample size, minor normality tends to occur and in order to satisfy homoscedasticity and other measures, this violation is unlikely to affect model inference.

```
# Normality
# Q-Q Plot
qqnorm(residuals(robust_model), main = "Figure 14: Q-Q Plot of Residuals")
qqline(residuals(robust_model), col = "red")
```

Figure 14: Q-Q Plot of Residuals



```
shapiro.test(residuals(robust_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(robust_model)
## W = 0.99153, p-value = 0.001089
```

Breusch-Pagan Test (Heteroskedasticity)

Hypothesis test: H0:Variance is constant HA:Variance is not constant

p-value is higher than 0.05, we failed to reject H0. There is no evidence of heteroskedasticity.

```
# Heteroscedasticity
bptest(robust_model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  robust_model
## BP = 6.6574, df = 5, p-value = 0.2474
```

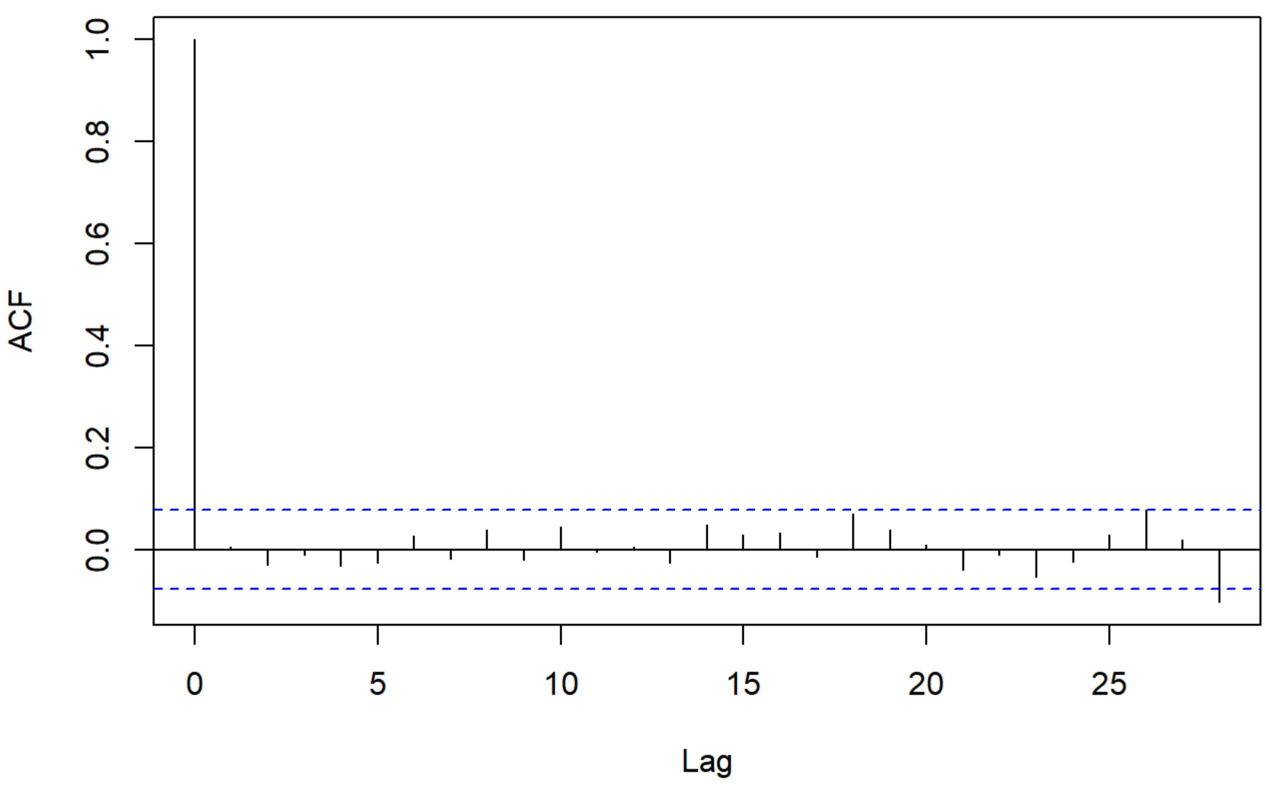
Durbin-Watson Test

Hypothesis test: H0:No autocorrelation HA:There is autocorrelation

Since p-value is greater than 0.05, we fail to reject H0.No correlation detected.

```
# Independence
acf(robust_model$residuals, main = "Figure 15: Autocorrelated Error Test")
```

Figure 15: Autocorrelated Error Test



```
dwtest(robust_model)
```

```
##
##  Durbin-Watson test
##
## data:  robust_model
## DW = 1.9883, p-value = 0.4418
## alternative hypothesis: true autocorrelation is greater than 0
```

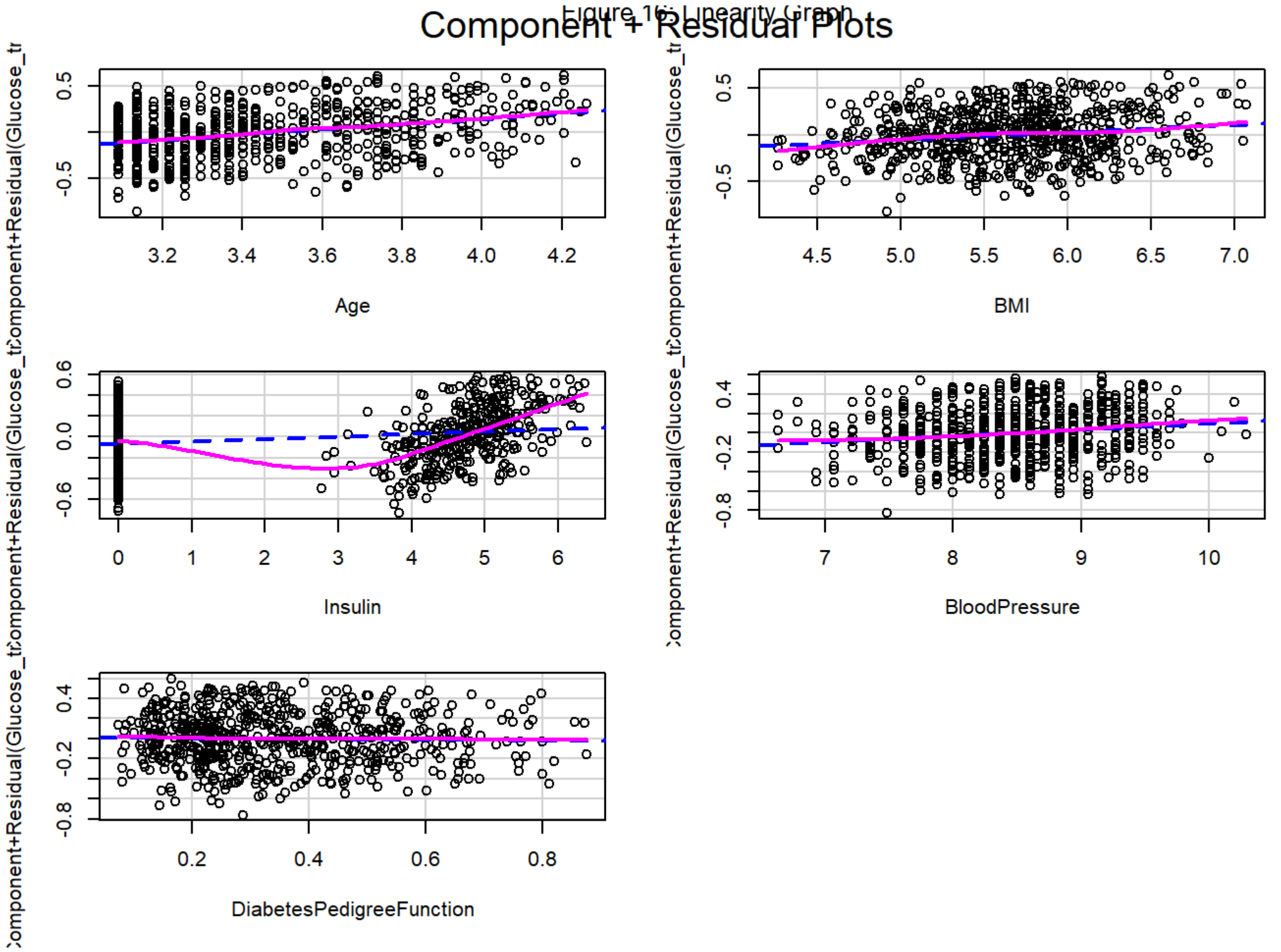
No predictor is above 5, so there is no multicollinearity.

```
#Multicollinearity
vif(robust_model)
```

```
##                      Age                      BMI                  Insulin 
##                 1.237229                 1.146847                 1.111801 
##            BloodPressure DiabetesPedigreeFunction 
##                 1.317437                 1.066609
```

Linearity almost remains same as the Box-Cox model, all the predictor’s both the pink curve and blue dotted line are very close to each other.

```
# Linearity
car::crPlots(robust_model)
mtext("Figure 16: Linearity Graph", side=3, line=3.5, cex=0.8, font=0.5)
```



```
# Making Predictions
pred_boxcox <- predict(model_boxcox)
pred_robust <- predict(robust_model)

# Actual Values
actual <- df_transformed$Glucose_transformed

# Calculate Metrics
mae_boxcox <- mae(actual, pred_boxcox)
rmse_boxcox <- rmse(actual, pred_boxcox)

mae_robust <- mae(actual, pred_robust)
rmse_robust <- rmse(actual, pred_robust)

cat("Box-Cox Model - MAE:", mae_boxcox, "RMSE:", rmse_boxcox, "\n")
```

```
## Box-Cox Model - MAE: 0.2157721 RMSE: 0.2637077
```

```
cat("Robust Model  - MAE:", mae_robust,  "RMSE:", rmse_robust,  "\n")
```

```
## Robust Model  - MAE: 0.2156808 RMSE: 0.2637381
```

```
# Residuals
resid_boxcox <- residuals(model_boxcox)
resid_robust <- residuals(robust_model)
```

```
# Histogram
hist1 <- ggplot(data.frame(resid_boxcox), aes(x = resid_boxcox)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  ggtitle("Figure 17: Residuals Histogram - BoxCox") +
  theme_minimal()

hist2 <- ggplot(data.frame(resid_robust), aes(x = resid_robust)) +
  geom_histogram(bins = 30, fill = "salmon", color = "black") +
  ggtitle("Figure 18: Residuals Histogram - Robust") +
  theme_minimal()
```

```
# QQ plots
qq1 <- ggplot(data.frame(sample = resid_boxcox), aes(sample = sample)) +
  stat_qq() + stat_qq_line() +
  ggtitle("Figure 19: QQ Plot - BoxCox") + theme_minimal()

qq2 <- ggplot(data.frame(sample = resid_robust), aes(sample = sample)) +
  stat_qq() + stat_qq_line() +
  ggtitle("Figure 20: QQ Plot - Robust") + theme_minimal()
```

```
# Fitted vs Actual
fit1 <- ggplot(df_transformed, aes(x = actual, y = pred_boxcox)) +
  geom_point(color = "blue") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  ggtitle("Figure 21: Fitted vs Actual - BoxCox") + xlab("Actual") + ylab("Predicted")

fit2 <- ggplot(df_transformed, aes(x = actual, y = pred_robust)) +
  geom_point(color = "red") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  ggtitle("Figure 22: Fitted vs Actual - Robust") + xlab("Actual") + ylab("Predicted")
```

```
grid.arrange(hist1, hist2, qq1, qq2, fit1, fit2, ncol = 2)
```

Figure 17: Residuals Histogram - BoxCox

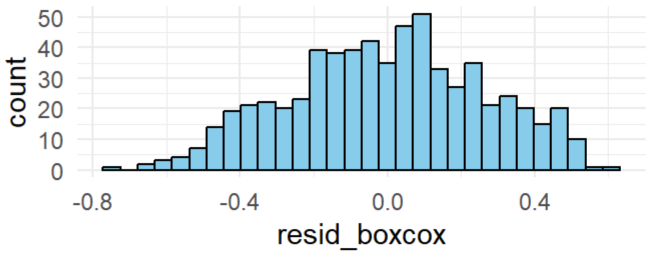


Figure 18: Residuals Histogram - Robu

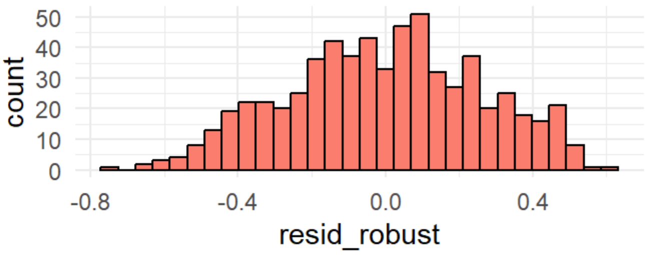


Figure 19: QQ Plot - BoxCox

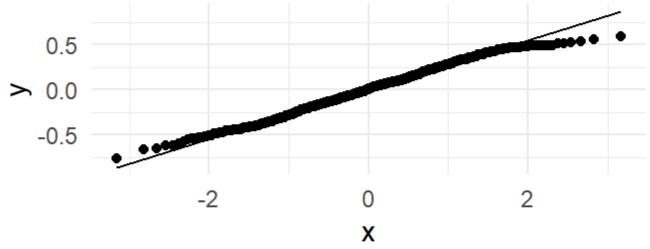


Figure 20: QQ Plot - Robust

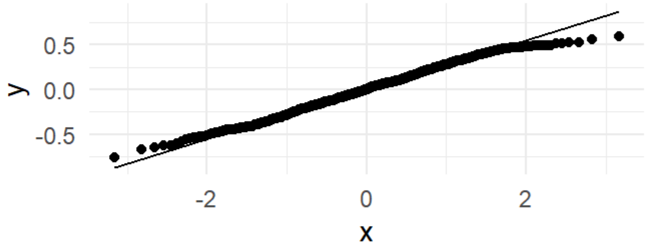


Figure 21: Fitted vs Actual - BoxCox

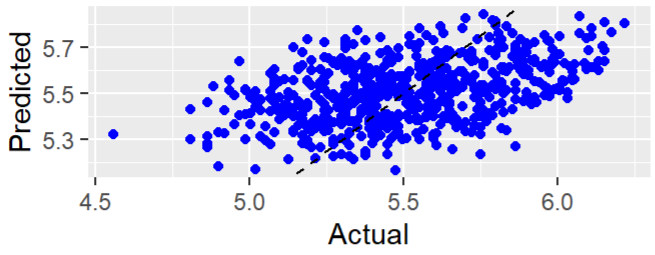
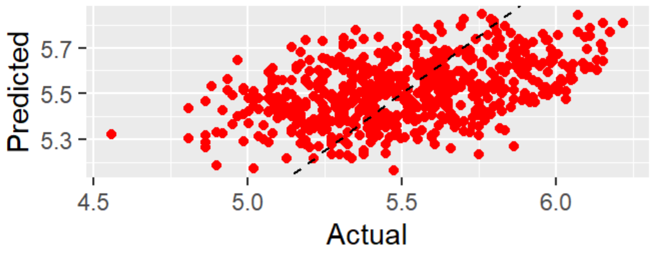


Figure 22: Fitted vs Actual - Robust



From the above graph, Histogram of Box-Cox is nearly symmetrical and Histogram of Robust is slightly more uniform. QQ-Plot of the models show some deviation at the tails. In the Fitted vs Actual graph, both models show similar spread. MAE and RMSE of both the models almost remain same, there is no practical difference in them.

```
# Full model with all predictors
full_model <- lm(Glucose_transformed ~ Age + BMI + Insulin + BloodPressure + DiabetesPedigree
Function,
                 data = df_transformed)

# Stepwise model selection using AIC
model_stepwise <- step(full_model, direction = "both", trace = FALSE)

# Summary and AIC
summary(model_stepwise)
```

```
##
## Call:
## lm(formula = Glucose_transformed ~ Age + BMI + Insulin + BloodPressure,
##     data = df_transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76136 -0.17597  0.00782  0.19060  0.59757
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.458006   0.171732  20.136  < 2e-16 ***
## Age           0.288834   0.038111   7.579 1.26e-13 ***
## BMI           0.080642   0.020140   4.004 6.97e-05 ***
## Insulin       0.022467   0.004464   5.033 6.31e-07 ***
## BloodPressure 0.063320   0.018781   3.371 0.000793 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2648 on 629 degrees of freedom
## Multiple R-squared:  0.2021, Adjusted R-squared:  0.197
## F-statistic: 39.83 on 4 and 629 DF,  p-value: < 2.2e-16
```

```
AIC(model_stepwise)
```

```
## [1] 121.2288
```

```
# Predicting and Evaluating
pred_stepwise <- predict(model_stepwise, df_transformed)
actual <- df_transformed$Glucose_transformed

# Metrics
r2_stepwise <- 1 - sum((actual - pred_stepwise)^2) / sum((actual - mean(actual))^2)
rmse_stepwise <- sqrt(mean((actual - pred_stepwise)^2))
mae_stepwise <- mean(abs(actual - pred_stepwise))

cat("Stepwise Linear Model:\n")
```

```
## Stepwise Linear Model:
```

```
cat("AIC:", AIC(model_stepwise), "\n")
```

```
## AIC: 121.2288
```

```
cat("R2:", round(r2_stepwise, 4), "\nRMSE:", round(rmse_stepwise, 4), "\nMAE:", round(mae_ste
pwise, 4), "\n\n")
```

```
## R2: 0.2021
## RMSE: 0.2637
## MAE: 0.2158
```

From the above Stepwise model, we can see that DiabetesPedigreeFunction did not improve the model’s AIC so stepwise selection removed that predictor.Remaining other predictors p-value’s are lesser than 0.05. AIC of this model is 121.2288, R2 -> which means about 20.2% of the variation in Glucose is explained by this model. Adjusted R-square is 0.197.RMSE is 0.2637 and MAE is 0.2158

```
# Residuals
res <- residuals(model_stepwise)
fitted <- fitted(model_stepwise)
```

Shapiro-Wilk Test

Hypothesis test: H0:Normal distribution is followed by residuals HA:No Normality

Since p-value is lesser than 0.05, we reject H0. Residuals are not normally distributed.

```
# Shapiro-Wilk test
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99131, p-value = 0.0008803
```

Breusch-Pagan Test

Hypothesis test: H0:Variance is constant HA:Variance is not constant

p-value is higher than 0.05, we failed to reject H0. There is no evidence of heteroskedasticity.

```
# Breusch-Pagan test for heteroscedasticity
bptest(model_stepwise)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model_stepwise
## BP = 5.4123, df = 4, p-value = 0.2475
```

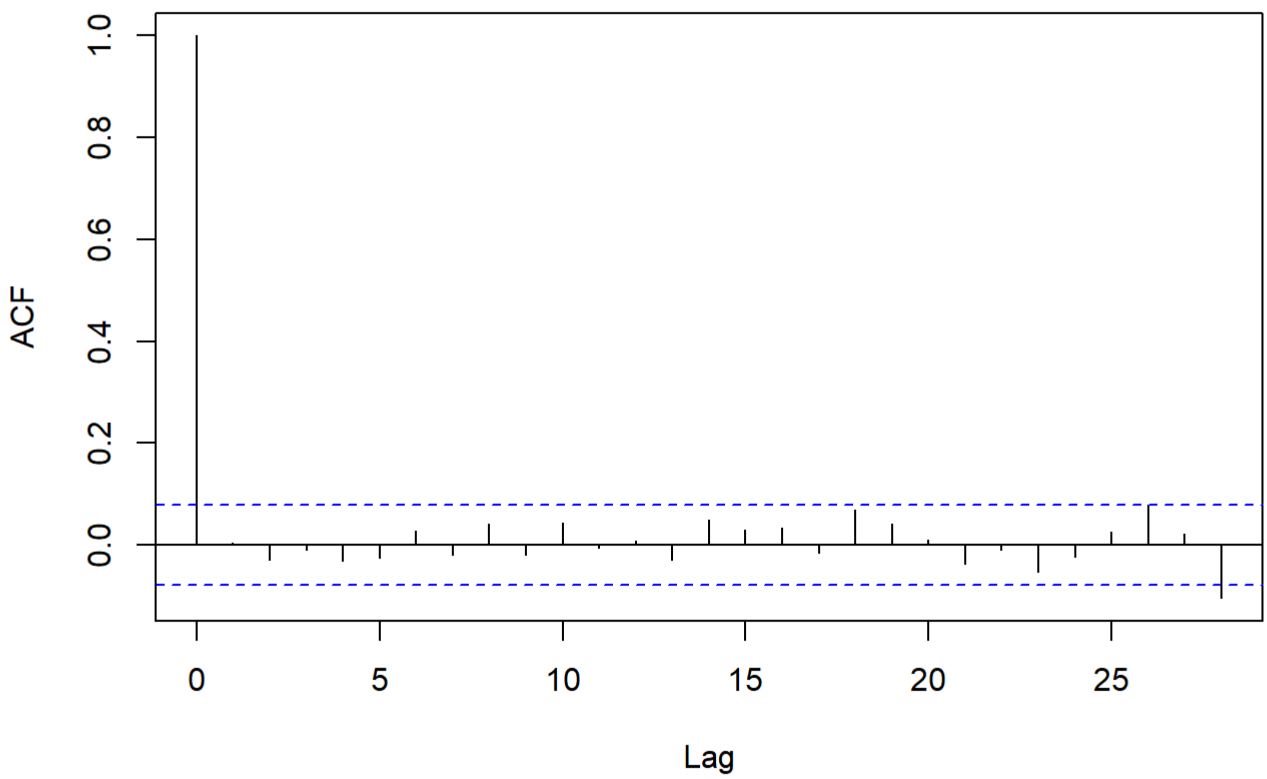
Durbin-Watson Test

Hypothesis test: H0:No autocorrelation HA:There is autocorrelation

Since p-value is greater than 0.05, we fail to reject H0.No correlation detected.

```
# Durbin-Watson test for autocorrelation
acf(model_stepwise$residuals, main = "Figure 23: Autocorrelated Error Test")
```

Figure 23: Autocorrelated Error Test



```
durbinWatsonTest(model_stepwise)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     0.003012036      1.992283   0.938
##  Alternative hypothesis: rho != 0
```

No predictor is above 5, so there is no multicollinearity.

```
# VIF for multicollinearity
vif(model_stepwise)
```

```
##           Age           BMI       Insulin BloodPressure 
##      1.231026      1.133367      1.064726      1.316994
```

Linearity remains same as other 2 models, all the predictor’s pink curve and blue dotted line are very close to each other.

```
# Linearity
car::crPlots(model_stepwise)
mtext("Figure 24: Linearity Graph", side=3, line=3.5, cex=0.8, font=0.5)
```

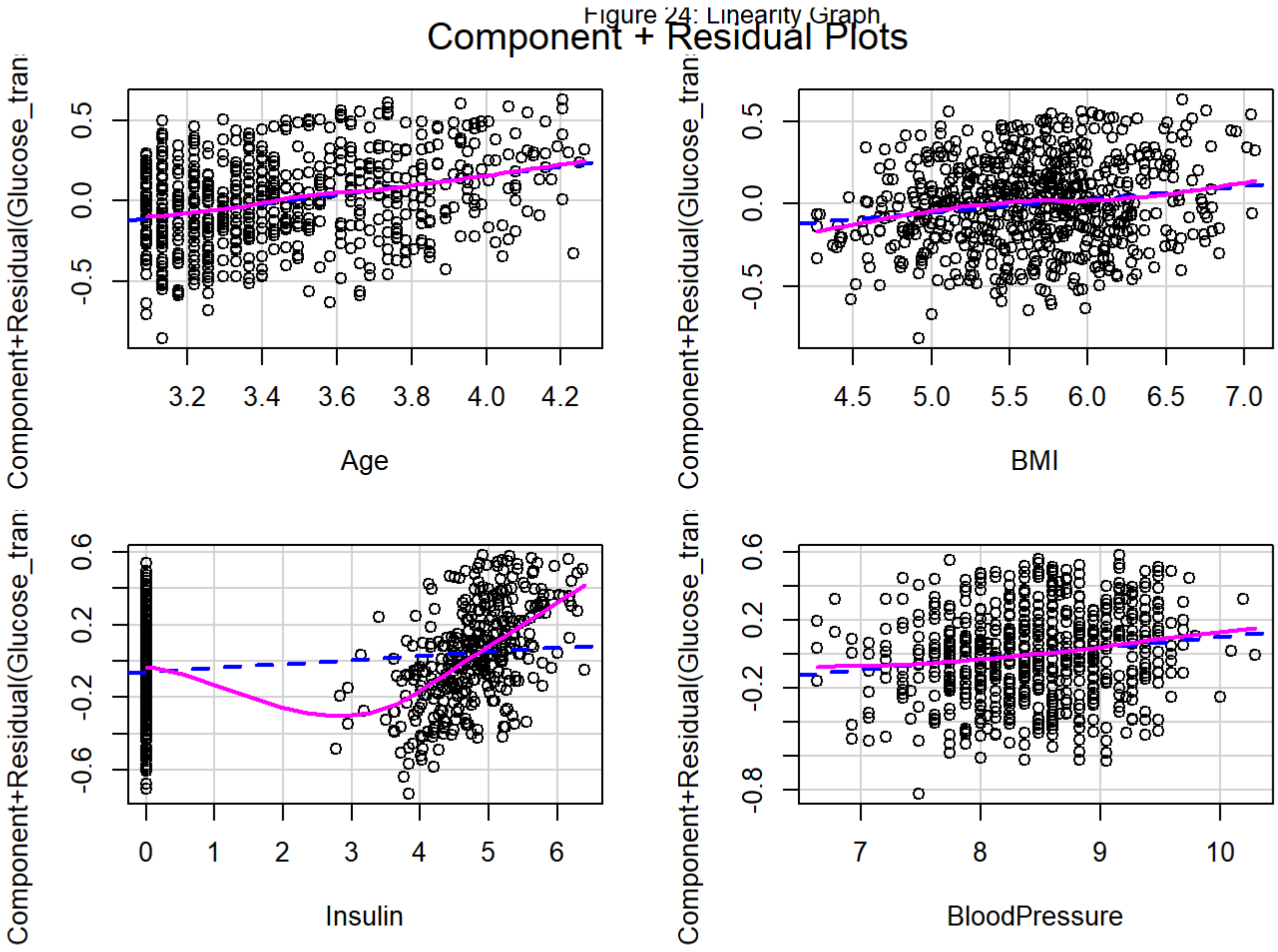
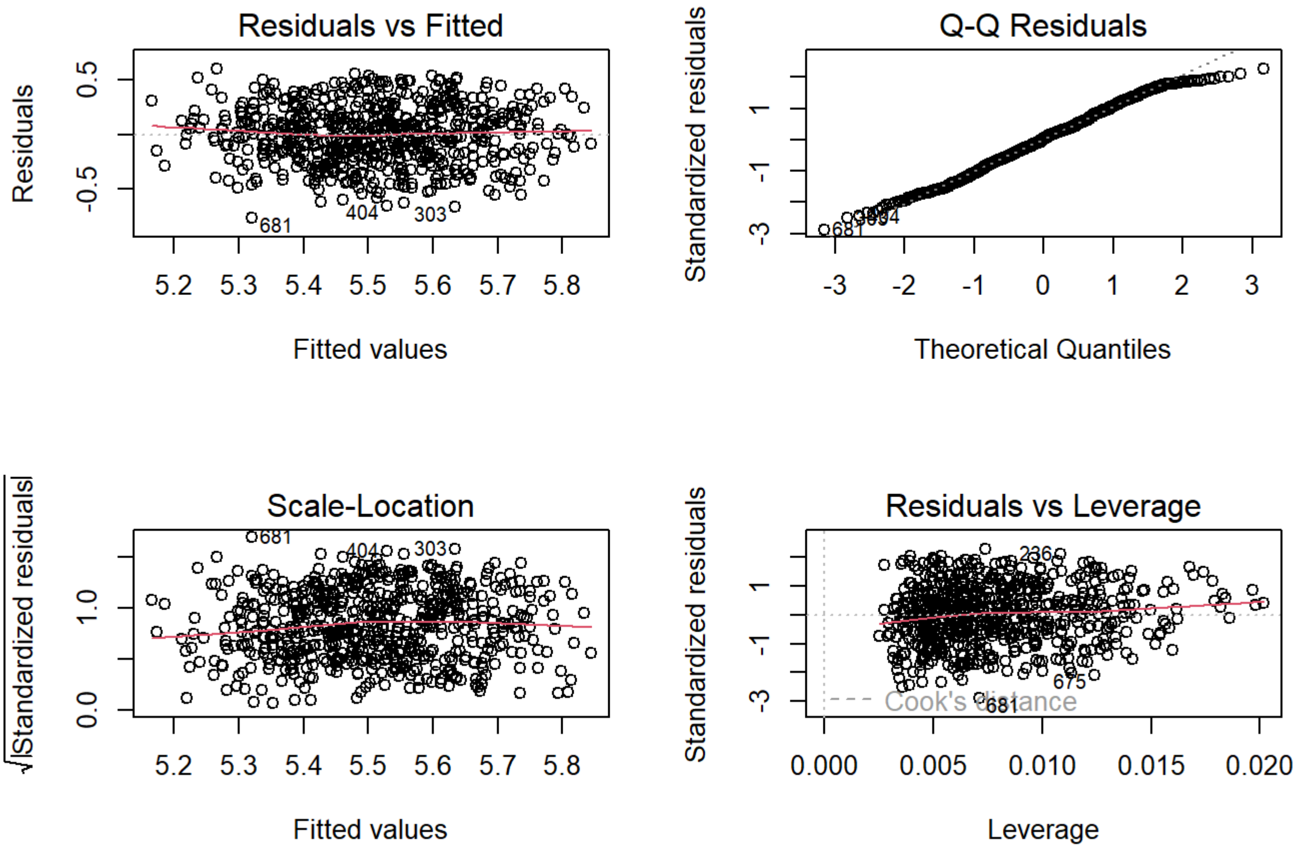


Figure 24: Linearity Graph

```
# Plotting diagnostics
par(mfrow = c(2, 2))
plot(model_stepwise)
mtext("Figure 25: Residual plot", side = 3, line = -2, outer = TRUE, cex = 1.2, font = 2)
```

Figure 25: Residual plot



```
# Histogram and Q-Q plot
par(mfrow = c(1, 2))
hist(res, main = "Figure 26: Histogram of Residuals", xlab = "Residuals", col = "skyblue")
qqnorm((res), main = "Figure 27: QQ Plot of Residuals")
qqline(res, col = "red")
```

Figure 26: Histogram of Residuals

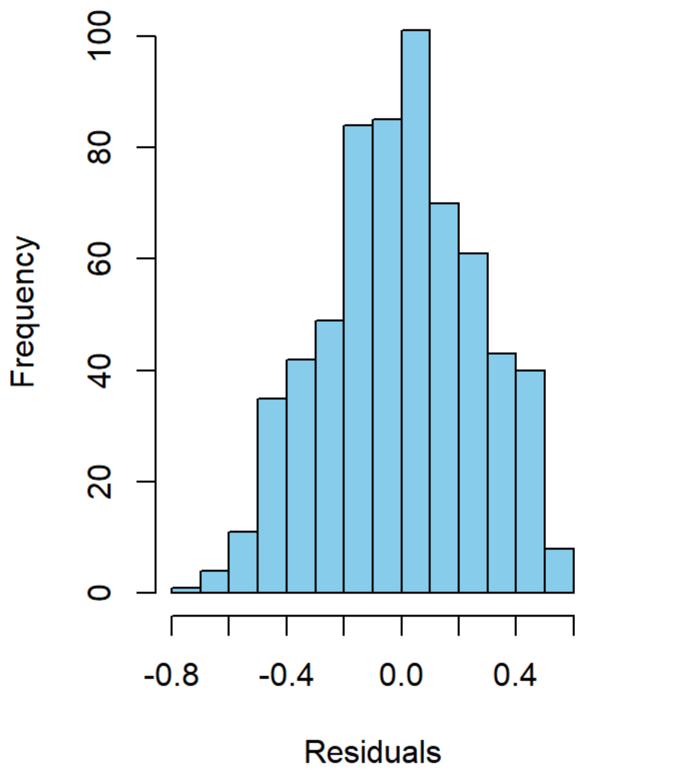
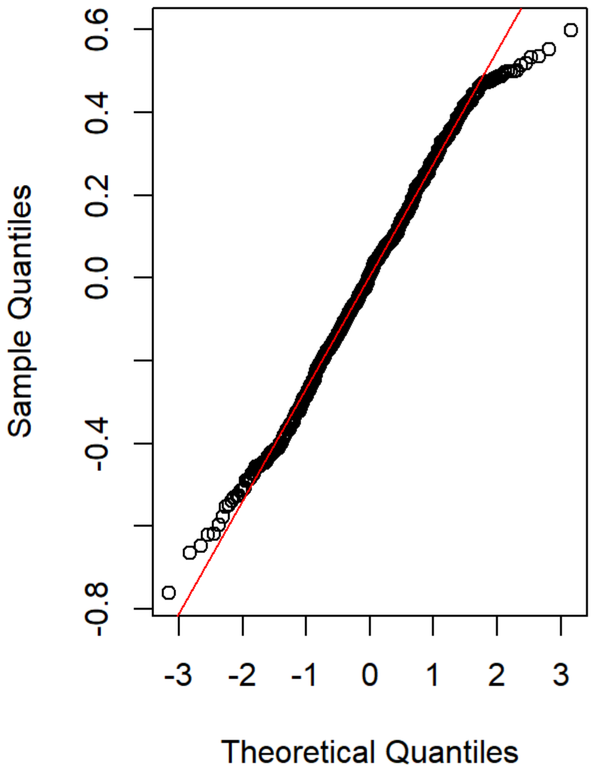


Figure 27: QQ Plot of Residuals



In Residual vs Fitted graph, all the points are well scattered around the red line.In Scale location graph, there is a slight trend of the red line.In Residuals vs Leverage graph, all the points are scattered around the red line.Histogram remains slightly symmetrical. QQ-PLot’s red line tends to capture all the points in the middle but at the tail ends, it was not able to do the same.

```
# Predictions
pred_linear  <- predict(model_final)
pred_robust  <- predict(robust_model)
pred_stepwise     <- predict(model_stepwise)
pred_boxcox <- predict(model_boxcox)

# True values
actual <- df_transformed$Glucose

# Metrics
# Linear
mae_linear  <- mae(actual, pred_linear)
rmse_linear <- rmse(actual, pred_linear)
aic_linear  <- AIC(model_final)

# Robust
mae_robust  <- mae(actual, pred_robust)
rmse_robust <- rmse(actual, pred_robust)
aic_robust  <- AIC(robust_model)

# Stepwise
mae_stepwise     <- mae(actual, pred_stepwise)
rmse_stepwise   <- rmse(actual, pred_stepwise)
aic_stepwise     <- AIC(model_stepwise)

#BoxCox
mae_boxcox <- mae(actual, pred_boxcox)
rmse_boxcox <- rmse(actual, pred_boxcox)
aic_boxcoz  <- AIC(model_boxcox)

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

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

```
cat("-----------------------------------\n")
```

```
## -----------------------------------
```

```
cat("Linear Model   - MAE:", round(mae_linear, 2),
    "| RMSE:", round(rmse_linear, 2),
    "| AIC:", round(aic_linear, 2), "\n")
```

```
## Linear Model   - MAE: 19.11 | RMSE: 23.51 | AIC: 5816.69
```

```
cat("Robust Model   - MAE:", round(mae_robust, 2),
    "| RMSE:", round(rmse_robust, 2),
    "| AIC:", round(aic_robust, 2), "\n")
```

```
## Robust Model   - MAE: 113.17 | RMSE: 116.2 | AIC: 123.23
```

```
cat("Stepwise Model      - MAE:", round(mae_stepwise, 2),
    "| RMSE:", round(rmse_stepwise, 2),
    "| AIC:", round(aic_stepwise, 2), "\n")
```

```
## Stepwise Model      - MAE: 113.18 | RMSE: 116.2 | AIC: 121.23
```

```
cat("BoxCox Model      - MAE:", round(mae_boxcox, 2),
    "| RMSE:", round(rmse_boxcox, 2),
    "| AIC:", round(aic_boxcoz, 2), "\n")
```

```
## BoxCox Model      - MAE: 113.18 | RMSE: 116.2 | AIC: 123.08
```

Results and Discussion:

We have analysed various linear modelling approach to predict Glucose using the predictors Age, BMI, Insulin, BloodPressure, and DiabetesPedigreeFunction. By comparing all the 4 models MAE, RMSE and AIC, we can say that Stepwise model has the lowest AIC value which is 121.23 indicating better model parsimony whereas linear model has lowest MAE and RMSE value indicating best predictive accuracy.But I am considering AIC as the best metric among the other 2 because it balances both model fit and complexity. We should also consider parsimony while choosing a model,which the AIC provides and also it provides a more robust basis for comparison.I am going with a model which offers more efficiency and a more statistically sound result.So Stepwise Model is the best model for this dataset.

Conclusion:

After evaluating multiple linear regression models, the best model for predicting Glucose levels is Stepwise model with four significant predictors: Age,BMI,Insulin and BloodPressure. This model has the least AIC value among all the other models.While normality remains slightly violated due to the large dataset, other assumptions like independence, linearity, and multicollinearity are all satisfied with the analysis.The DiabetesPedigreeFunction had a p-value greater than 0.05 and it was thus non-significant, so it was excluded in Stepwise Model. We are considering 2 persons with Age 45 and 60, BMI is 28 and 35 and Insulin level is 90 and 150 and BloodPressure is 80 and 90. For the first patient, Predicted_Glucose value is 129.67, the value is above the normal range, typically the value should be lesser than 110. The individual is now in early diabetic range. The combination of moderate BMI and Insulin is the reason for the elevated glucose prediction.In second patient, Predicted_Glucose is 147.93, the patient is predicted to have high glucose level. Given the older age, higher BMI and higher Insulin levels, we can see a clear picture of uncontrolled diabetes. These 2 predictions reflect real world examples and clearly explain how the model can assist in early identification of people who are at risk of diabetes and help in quantifying effect of clinical measures on glucose levels.

```
# Preparing new observations
new_data <- data.frame(
  Age = log1p(c(45, 60)),
  BMI = sqrt(c(28, 35)),
  Insulin = log1p(c(90, 150)),
  BloodPressure = sqrt(c(80, 90))
)

# Prediction using stepwise model
predicted_transformed <- predict(model_stepwise, newdata = new_data)

# Applying inverse Box-Cox transformation
# Box-Cox lambda value is
lambda <- 0.06060606

inverse_boxcox <- function(y, lambda) {
  if (abs(lambda) < 1e-4) {
    return(exp(y))
  } else {
    return((y * lambda + 1)^(1 / lambda))
  }
}

predicted_glucose <- inverse_boxcox(predicted_transformed, lambda)

# Displaying results
result_df <- cbind(
  New_Patient = c("Patient 1", "Patient 2"),
  Predicted_Glucose_Transformed = round(predicted_transformed, 4),
  Predicted_Glucose = round(predicted_glucose, 2)
)

print(result_df)
```

```
##   New_Patient Predicted_Glucose_Transformed Predicted_Glucose
## 1 "Patient 1" "5.6583"                      "129.67"
## 2 "Patient 2" "5.8359"                      "147.93"
```

Reference:

1. https://www.kaggle.com/datasets/mathchi/diabetes-data-set (https://www.kaggle.com/datasets/mathchi/diabetes-data-set)
2. https://rmit.instructure.com/courses/140876/pages/week-4-after-class-3?module_item_id=7223180 (https://rmit.instructure.com/courses/140876/pages/week-4-after-class-3?module_item_id=7223180)
3. https://rmit.instructure.com/courses/140876/pages/week-5-after-class-2?module_item_id=7232820 (https://rmit.instructure.com/courses/140876/pages/week-5-after-class-2?module_item_id=7232820)
4. https://rmit.instructure.com/courses/140876/pages/week-6-after-class-2?module_item_id=7247271 (https://rmit.instructure.com/courses/140876/pages/week-6-after-class-2?module_item_id=7247271)
5. https://rmit.instructure.com/courses/140876/pages/week-7-after-class-3?module_item_id=7270551 (https://rmit.instructure.com/courses/140876/pages/week-7-after-class-3?module_item_id=7270551)

6. https://rmit.instructure.com/courses/140876/pages/week-8-after-class-2?module_item_id=7291996 (https://rmit.instructure.com/courses/140876/pages/week-8-after-class-2?module_item_id=7291996)

7. https://rmit.instructure.com/courses/140876/pages/week-9-after-class-2?module_item_id=7307061 (https://rmit.instructure.com/courses/140876/pages/week-9-after-class-2?module_item_id=7307061)

8. https://rmit.instructure.com/courses/140876/pages/week-10-after-class?module_item_id=7317487 (https://rmit.instructure.com/courses/140876/pages/week-10-after-class?module_item_id=7317487)

9. https://rmit.instructure.com/courses/140876/pages/week-11-during-class?module_item_id=7338093 (https://rmit.instructure.com/courses/140876/pages/week-11-during-class?module_item_id=7338093)

Appendix:

Loading libraries

```
suppressPackageStartupMessages({ suppressWarnings({ library(tidyverse) library(ggplot2) library(reshape2)
library(gridExtra) library(Metrics) library(car) library(lmtest) library(MASS) }) })
```

Loading dataset

```
df <- read.csv(“diabetes.csv”) head(df)
```

Excluding ‘Outcome’, ‘Pregnancies’, and ‘SkinThickness’ columns

```
df_noutcome <- df[, !names(df) %in% c(“Outcome”, “Pregnancies”, “SkinThickness”)]
```

Checking data structure

```
str(df_noutcome) summary(df_noutcome) sum(is.na(df_noutcome)) colnames(df_noutcome)
```

Checking missing values

```
colSums(is.na(df_noutcome))
```

Histogram

```
hist_plots <- lapply(names(df_noutcome), function(var) { ggplot(df_noutcome, aes(x = !!sym(var))) +
geom_histogram(bins = 30, fill = “steelblue”, color = “black”) + theme_minimal() + ggtitle(paste(“Histogram of”,
var)) })
```

Boxplots

```
box_plots <- lapply(names(df_noutcome), function(var) { ggplot(df_noutcome, aes(y = !!sym(var))) +
geom_boxplot(fill = “tomato”, color = “black”) + theme_minimal() + ggtitle(paste(“Boxplot of”, var)) })
```

Displaying histogram

```
do.call(“grid.arrange”, c(hist_plots, ncol = 2, top = “Figure 1: Histogram of Predictors”))
```

Displaying boxplot

do.call(“grid.arrange”, c(box_plots, ncol = 2, top = “Figure 2: Boxplots of Predictors”))

Correlation matrix

cor_data <- df_noutcome[, c(“Glucose”, “Age”, “BMI”, “Insulin”, “BloodPressure”, “DiabetesPedigreeFunction”)] cor_matrix <- cor(cor_data, use = “complete.obs”)

Correlation heatmap

corrplot(cor_matrix, method = “color”, addCoef.col = “black”, number.cex = 0.7, tl.col = “black”, tl.srt = 45, title = “Figure 3: Correlation Matrix”, mar=c(0,0,1,0))

Function to count outliers using IQR method

count_outliers_iqr <- function(x) { Q1 <- quantile(x, 0.25, na.rm = TRUE) Q3 <- quantile(x, 0.75, na.rm = TRUE) IQR <- Q3 - Q1 lower <- Q1 - 1.5 * IQR upper <- Q3 + 1.5 * IQR sum(x < lower | x > upper, na.rm = TRUE) }

outlier_counts <- sapply(df_noutcome, count_outliers_iqr) outlier_counts

df_transformed <- df_noutcome

Applying transformations

df_transformed$Insulin < -log1p(df_transformed$Insulin) df_transformed $DiabetesPedigreeFunction < -log1p(df_transformed$DiabetesPedigreeFunction) df_transformed $BloodPressure < -sqrt(df_transformed$BloodPressure) df_transformed $Age < -log1p(df_transformed$Age) df_transformed$BMI < -sqrt(df_transformed$BMI)

Viewing summary to confirm

summary(df_transformed)

sapply(df_transformed, count_outliers_iqr)

Model A: With outliers

model_with_outliers <- lm(Glucose ~ Age + BMI + Insulin + BloodPressure + DiabetesPedigreeFunction, data = df_transformed) summary(model_with_outliers)

remove_outliers <- function(x) { Q1 <- quantile(x, 0.25, na.rm = TRUE) Q3 <- quantile(x, 0.75, na.rm = TRUE) IQR <- Q3 - Q1 x[x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR)] <- NA return(x) }

Model B: Without outliers

df_clean <- as.data.frame(lapply(df_transformed, remove_outliers)) df_clean <- na.omit(df_clean)

model_without_outliers <- lm(Glucose ~ Age + BMI + Insulin + BloodPressure + DiabetesPedigreeFunction, data = df_clean) summary(model_without_outliers)

pred_with <- predict(model_with_outliers) pred_without <- predict(model_without_outliers)

True values

actual_with <- df_transformed$Glucoseactual_{w}ithout < -df_{c}lean$Glucose

cat(“With Outliers - MAE:”, mae(actual_with, pred_with), “RMSE:”, rmse(actual_with, pred_with), “AIC:”, AIC(model_with_outliers), “”)

cat(“Without Outliers - MAE:”, mae(actual_without, pred_without), “RMSE:”, rmse(actual_without, pred_without), “AIC:”, AIC(model_without_outliers), “”)

par(mfrow=c(2,2)) plot(model_without_outliers) mtext(“Figure 4: Residual plot”, side = 3, line = -2, outer = TRUE, cex = 1.2, font = 2)

Residuals vs Fitted Plot for model_without_outliers

plot(model_without_outliers$fitted.values, model_{w}ithout_{o}utliers$residuals, xlab = “Fitted Values”, ylab = “Residuals”, main = “Figure 5: Residuals vs Fitted”) abline(h = 0, col = “red”)

Standardised and Studentised Residuals

standard_res <- rstandard(model_without_outliers) student_res <- rstudent(model_without_outliers)

Identifying observations with high residuals

which(abs(student_res) > 2)

Cook’s Distance

cooks_d <- cooks.distance(model_without_outliers) plot(cooks_d, type = “h”, main = “Figure 6: Cook’s Distance”, ylab = “Cook’s Distance”) abline(h = 4 / length(model_without_outliers$fitted.values), col = “red”, lty = 2)

Identifying high influence points

which(cooks_d > 4 / length(model_without_outliers$fitted.values))

#Leverage (Hat) Values hat_vals <- hatvalues(model_without_outliers)

Plotting leverage values

plot(hat_vals, type = “h”, main = “Figure 7: Leverage Values”, ylab = “Leverage”) abline(h = 2 * mean(hat_vals), col = “red”, lty = 2)

Identifying high leverage points

high_leverage <- which(hat_vals > 2 * mean(hat_vals)) print(“High leverage points:”) print(high_leverage)

Combining all unique influential indices

```
final_influential_points <- unique(c( which(abs(student_res) > 2), which(cooks_d > 4 / length(model_without_outliers$fitted.values)), which(hat_vals > 2 * mean(hat_vals)) ))
```

```
length(final_influential_points)
```

Creating a cleaned dataset by removing influential points

```
df_hat_removed_final <- df_clean[-final_influential_points, ]
```

Fitting the final model

```
model_final <- lm(Glucose ~ Age + BMI + Insulin + BloodPressure + DiabetesPedigreeFunction, data = df_hat_removed_final)
```

Summary of the model

```
summary(model_final)
```

ANOVA

```
anova(model_final)
```

Getting coefficients and Standard errors

```
coefs1 <- summary(model_final)$coefficients
```

Computing Wald statistics

```
wald_stats <- (coefs1[, “Estimate”] / coefs1[, “Std. Error”])^2
```

Computing p-values from Chi-squared distribution with df = 1

```
wald_pvals <- 1 - pchisq(wald_stats, df = 1)
```

```
wald_results <- data.frame( Estimate = coefs1[, “Estimate”], StdError = coefs1[, “Std. Error”], WaldStatistic = round(wald_stats, 3), p_value = round(wald_pvals, 4), Significance = ifelse(wald_pvals < 0.05, “Significant”, “Not Significant”) )
```

```
print(wald_results)
```

Function to calculate performance metrics

```
model_metrics <- function(model, data) { preds <- predict(model, data) actual <- data$Glucose mae <- mean(abs(preds - actual)) rmse <- sqrt(mean((preds - actual)^2)) aic <- AIC(model) return(c(MAE = mae, RMSE = rmse, AIC = aic)) }
```

Computing metrics

metrics_with_outliers <- model_metrics(model_with_outliers, df_transformed) metrics_without_outliers <- model_metrics(model_without_outliers, df_clean) metrics_final <- model_metrics(model_final, df_hat_removed_final)

comparison <- rbind(With_Outliers = metrics_with_outliers, Without_Outliers = metrics_without_outliers, Final_Cleaned = metrics_final)

print(round(comparison, 2))

acf(model_final$residuals, main = “Figure 8: Autocorrelated Error Test”) dwtest(model_final)

bptest(model_final)

Q-Q Plot

qqnorm(residuals(model_final), main = “Figure 9: Q-Q Plot of Residuals”) qqline(residuals(model_final), col = “red”)

Shapiro-Wilk Test

shapiro.test(residuals(model_final))

VIF (Multicollinearity)

vif(model_final)

Linearity

car::crPlots(model_final) mtext(“Figure 10: Linearity Graph”, side=3, line=3.5, cex=0.8, font=0.5)

Applying Box-Cox transformation on the model

boxcox_result <- boxcox(model_final, lambda = seq(-2, 2, 0.1))

Finding the lambda with maximum log-likelihood

lambda_optimal <- boxcox_result$x[which.\,max(boxcox_{r}esult$y)] cat(“Optimal lambda:”, lambda_optimal, “”)

Transforming the response variable based on the optimal lambda

if (abs(lambda_optimal) < 1e-4) { df_transformed <- df_hat_removed_final df_transformed $Glucose_{t}ransformed < -log(df_{t}ransformed$Glucose) } else { df_transformed <- df_hat_removed_final df_transformed$Glucose_{t}ransformed < -(df_{t}ransformed$

```
Glucose^lambda_optimal - 1) / lambda_optimal }
```

```
model_boxcox <- lm(Glucose_transformed ~ Age + BMI + Insulin + BloodPressure + DiabetesPedigreeFunction, data = df_transformed) summary(model_boxcox)
```

ANOVA

```
anova(model_boxcox)
```

Getting coefficients and Standard errors

```
coefs2 <- summary(model_boxcox)$coefficients
```

Computing Wald statistics

```
wald_stats1 <- (coefs2[, “Estimate”] / coefs2[, “Std. Error”])^2
```

Computing p-values from Chi-squared distribution with df = 1

```
wald_pvals1 <- 1 - pchisq(wald_stats1, df = 1)
```

```
wald_results1 <- data.frame( Estimate1 = coefs2[, “Estimate”], StdError1 = coefs2[, “Std. Error”], WaldStatistic1 = round(wald_stats1, 3), p_value1 = round(wald_pvals1, 4), Significance1 = ifelse(wald_pvals1 < 0.05, “Significant”, “Not Significant”) )
```

```
print(wald_results1)
```

Normality

Q-Q Plot

```
qqnorm(residuals(model_boxcox), main = “Figure 11: Q-Q Plot of Residuals”) qqline(residuals(model_boxcox), col = “red”)
```

```
shapiro.test(residuals(model_boxcox))
```

Heteroscedasticity

```
bptest(model_boxcox)
```

Independence

```
acf(model_boxcox$residuals, main = “Figure 12: Autocorrelated Error Test”) dwtest(model_boxcox)
```

```
#Multicollinearity vif(model_boxcox)
```

Linearity

```
car::crPlots(model_boxcox) mtext(“Figure 13: Linearity Graph”, side=3, line=3.5, cex=0.8, font=0.5)
```

Fitting robust regression on the transformed dataset

robust_model <- rlm(Glucose_transformed ~ Age + BMI + Insulin + BloodPressure + DiabetesPedigreeFunction, data = df_transformed)

Summary of the robust model

summary(robust_model)

Getting coefficients matrix (no column names)

coefs3 <- summary(robust_model)$coefficients

Assigning column names manually

colnames(coefs3) <- c(“Estimate”, “Std.Error”, “t.value”)

Computing Wald statistics

wald_stats2 <- (coefs3[, “Estimate”] / coefs3[, “Std.Error”])^2

Computing p-values from Chi-squared distribution with df = 1

wald_pvals2 <- 1 - pchisq(wald_stats2, df = 1)

Compiling results

wald_results2 <- data.frame(Estimate2 = coefs3[, “Estimate”], StdError2 = coefs3[, “Std.Error”], WaldStatistic2 = round(wald_stats2, 3), p_value2 = round(wald_pvals2, 4), Significance2 = ifelse(wald_pvals2 < 0.05, “Significant”, “Not Significant”))

print(wald_results2)

Normality

Q-Q Plot

qqnorm(residuals(robust_model), main = “Figure 14: Q-Q Plot of Residuals”) qqline(residuals(robust_model), col = “red”)

shapiro.test(residuals(robust_model))

Heteroscedasticity

bptest(robust_model)

Independence

acf(robust_model$residuals, main = “Figure 15: Autocorrelated Error Test”) dwtest(robust_model)

#Multicollinearity vif(robust_model)

Linearity

car::crPlots(robust_model) mtext(“Figure 16: Linearity Graph”, side=3, line=3.5, cex=0.8, font=0.5)

Making Predictions

pred_boxcox <- predict(model_boxcox) pred_robust <- predict(robust_model)

Actual Values

actual <- df_transformed$Glucose_transformed

Calculate Metrics

mae_boxcox <- mae(actual, pred_boxcox) rmse_boxcox <- rmse(actual, pred_boxcox)

mae_robust <- mae(actual, pred_robust) rmse_robust <- rmse(actual, pred_robust)

cat(“Box-Cox Model - MAE:”, mae_boxcox, “RMSE:”, rmse_boxcox, “”) cat(“Robust Model - MAE:”, mae_robust, “RMSE:”, rmse_robust, “”)

Residuals

resid_boxcox <- residuals(model_boxcox) resid_robust <- residuals(robust_model)

Histogram

hist1 <- ggplot(data.frame(resid_boxcox), aes(x = resid_boxcox)) + geom_histogram(bins = 30, fill = “skyblue”, color = “black”) + ggtitle(“Figure 17: Residuals Histogram - BoxCox”) + theme_minimal()

hist2 <- ggplot(data.frame(resid_robust), aes(x = resid_robust)) + geom_histogram(bins = 30, fill = “salmon”, color = “black”) + ggtitle(“Figure 18: Residuals Histogram - Robust”) + theme_minimal()

QQ plots

qq1 <- ggplot(data.frame(sample = resid_boxcox), aes(sample = sample)) + stat_qq() + stat_qq_line() + ggtitle(“Figure 19: QQ Plot - BoxCox”) + theme_minimal()

qq2 <- ggplot(data.frame(sample = resid_robust), aes(sample = sample)) + stat_qq() + stat_qq_line() + ggtitle(“Figure 20: QQ Plot - Robust”) + theme_minimal()

Fitted vs Actual

fit1 <- ggplot(df_transformed, aes(x = actual, y = pred_boxcox)) + geom_point(color = “blue”) + geom_abline(slope = 1, intercept = 0, linetype = “dashed”) + ggtitle(“Figure 21: Fitted vs Actual - BoxCox”) + xlab(“Actual”) + ylab(“Predicted”)

fit2 <- ggplot(df_transformed, aes(x = actual, y = pred_robust)) + geom_point(color = “red”) + geom_abline(slope = 1, intercept = 0, linetype = “dashed”) + ggtitle(“Figure 22: Fitted vs Actual - Robust”) + xlab(“Actual”) + ylab(“Predicted”)

grid.arrange(hist1, hist2, qq1, qq2, fit1, fit2, ncol = 2)

Full model with all predictors

full_model <- lm(Glucose_transformed ~ Age + BMI + Insulin + BloodPressure + DiabetesPedigreeFunction, data = df_transformed)

Stepwise model selection using AIC

model_stepwise <- step(full_model, direction = “both”, trace = FALSE)

Summary and AIC

summary(model_stepwise) AIC(model_stepwise)

Predicting and Evaluating

pred_stepwise <- predict(model_stepwise, df_transformed) actual <- df_transformed$Glucose_transformed

Metrics

r2_stepwise <- 1 - sum((actual - pred_stepwise)^2) / sum((actual - mean(actual))^2) rmse_stepwise <- sqrt(mean((actual - pred_stepwise)^2)) mae_stepwise <- mean(abs(actual - pred_stepwise))

cat(“Stepwise Linear Model:”) cat(“AIC:”, AIC(model_stepwise), “”) cat(“R2:”, round(r2_stepwise, 4), “:”, round(rmse_stepwise, 4), “:”, round(mae_stepwise, 4), “”)

Residuals

res <- residuals(model_stepwise) fitted <- fitted(model_stepwise)

Shapiro-Wilk test

shapiro.test(res)

Breusch-Pagan test for heteroscedasticity

bptest(model_stepwise)

Durbin-Watson test for autocorrelation

acf(model_stepwise$residuals, main = “Figure 23: Autocorrelated Error Test”)
durbinWatsonTest(model_stepwise)

VIF for multicollinearity

vif(model_stepwise)

Linearity

car::crPlots(model_stepwise) mtext(“Figure 24: Linearity Graph”, side=3, line=3.5, cex=0.8, font=0.5)

Plotting diagnostics

par(mfrow = c(2, 2)) plot(model_stepwise) mtext(“Figure 25: Residual plot”, side = 3, line = -2, outer = TRUE, cex = 1.2, font = 2)

Histogram and Q-Q plot

par(mfrow = c(1, 2)) hist(res, main = “Figure 26: Histogram of Residuals”, xlab = “Residuals”, col = “skyblue”) qqnorm((res), main = “Figure 27: QQ Plot of Residuals”) qqline(res, col = “red”)

Predictions

pred_linear <- predict(model_final) pred_robust <- predict(robust_model) pred_stepwise <- predict(model_stepwise) pred_boxcox <- predict(model_boxcox)

True values

actual <- df_transformed$Glucose

Metrics

Linear

mae_linear <- mae(actual, pred_linear) rmse_linear <- rmse(actual, pred_linear) aic_linear <- AIC(model_final)

Robust

mae_robust <- mae(actual, pred_robust) rmse_robust <- rmse(actual, pred_robust) aic_robust <- AIC(robust_model)

Stepwise

mae_stepwise <- mae(actual, pred_stepwise) rmse_stepwise <- rmse(actual, pred_stepwise) aic_stepwise <- AIC(model_stepwise)

#BoxCox mae_boxcox <- mae(actual, pred_boxcox) rmse_boxcox <- rmse(actual, pred_boxcox) aic_boxcoz <- AIC(model_boxcox)

```
cat(“Performance Comparison:”) cat(“——————————————————”) cat(“Linear Model - MAE:”, round(mae_linear,
2), “| RMSE:”, round(rmse_linear, 2), “| AIC:”, round(aic_linear, 2), “”) cat(“Robust Model - MAE:”,
round(mae_robust, 2), “| RMSE:”, round(rmse_robust, 2), “| AIC:”, round(aic_robust, 2), “”) cat(“Stepwise Model
- MAE:”, round(mae_stepwise, 2), “| RMSE:”, round(rmse_stepwise, 2), “| AIC:”, round(aic_stepwise, 2), “”)
cat(“BoxCox Model - MAE:”, round(mae_boxcox, 2), “| RMSE:”, round(rmse_boxcox, 2), “| AIC:”,
round(aic_boxcoz, 2), “”)
```

Preparing new observations

```
new_data <- data.frame( Age = log1p(c(45, 60)),
BMI = sqrt(c(28, 35)),
Insulin = log1p(c(90, 150)),
BloodPressure = sqrt(c(80, 90))
)
```

Prediction using stepwise model

```
predicted_transformed <- predict(model_stepwise, newdata = new_data)
```

Applying inverse Box-Cox transformation

Box-Cox lambda value is

```
lambda <- 0.06060606
```

```
inverse_boxcox <- function(y, lambda) { if (abs(lambda) < 1e-4) { return(exp(y))
} else { return((y * lambda + 1)^(1 / lambda))
} }
```

```
predicted_glucose <- inverse_boxcox(predicted_transformed, lambda)
```

Displaying results

```
result_df <- cbind( New_Patient = c(“Patient 1”, “Patient 2”), Predicted_Glucose_Transformed =
round(predicted_transformed, 4), Predicted_Glucose = round(predicted_glucose, 2) )
```

```
print(result_df)
```