Performing Linear Regression Techinques on a Cardiovascular Disease Dataset

Perla Molina, Malik Khouma, Elisabeth Snider

2022-12-15

Introduction

The Dataset we are working examining for this project evaluates a number of factors related to cardiovascular health. There are 12 features and 70,000 observations. IN looking at this dataset, we are most interested in identifying factors that contribute to high blood pressure. Our project aims to evaluate those factors in a clear and quantifiable way.

Blood pressure can be measured in two ways, systolic blood pressure (the top number) and diastolic blood pressure(the bottom number). We made the choice to focus on systolic blood pressure as high systolic blood pressure is associated with more negative health outcomes, and therefore is of more interest to us.

Running Libraries and Functions

We created a regression function that chooses the best model for any purpose. It cleans the data, removes NAs, and lets the user choose from a Predictive or Explanatory model. It then fits Lasso, Ridge, and OLS and compares the MSE and Adjusted R-Squared values to choose the best model. It also checks for outliers and runs diagnostic plots to evaluate normality and constant variance to make sure all assumptions are met.

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-6
```

```
library(matlib)
library(ggplot2)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(MASS)

outlier_helper <- function(X, y, fitted.values) {

  # High Leverage

  hat_diags <- hatvalues(lm(y ~ ., data.frame(X)))

  influential.point <- 2 * mean(hat_diags)
  influential.points <- which(hat_diags > influential.point)

  # Points of Influence
  residuals <- fitted.values - y
  mse <- mean(residuals^2)
  n <- dim(X)[1]
  p <- dim(X)[2]

  si2 <- ((n - p)/(n - p - 1) * mse) - (residuals ^ 2 / ((n - p - 1) * (1 - hat_diags)))
  si <- sqrt(si2)
  di <- residuals / (1 - hat_diags)

  di.real <- abs(di / si)
  high.leverage <- which(di.real > qt(p = 0.05, df = (n-p-1), lower.tail = FALSE))

  outliers <- list(high.leverage = high.leverage,
                   influential.points = influential.points,
                   outliers = union(high.leverage, influential.points))
}



diagnostic_helper <- function(fitted.values, actual.values) {

  diagnostics <- list()

  residuals <- fitted.values - actual.values

  base.plot <- ggplot() + theme_minimal()

  diagnostics$residual.hist <- base.plot + geom_histogram(aes(residuals)) +
    labs(x = "Residuals")
  diagnostics$qq <- base.plot + geom_qq(aes(sample = residuals)) +
    labs(x = "Theoretical", y = "Sample")
  diagnostics$cv <- base.plot + geom_point(aes(fitted.values, residuals)) +
    labs(x = "Fitted Values", y = "Residuals")
  diagnostics$ks <- ks.test(residuals, "pnorm")

  if (length(residuals) > 3 && length(residuals) < 5000) {
    diagnostics$sw <- shapiro.test(residuals)
  }

  diagnostics
}
```

```r
predictive_helper <- function(X, y) {

  ridge.model <- cv.glmnet(X, y, alpha = 0)
  ridge.lambda <- ridge.model$lambda.min
  lasso.model <- cv.glmnet(X, y, alpha = 1)
  lasso.lambda <- lasso.model$lambda.min

  train <- sample(c(TRUE, FALSE), nrow(X), replace = TRUE, prob = c(0.8, 0.2))

  y.train <- y[train]
  y.test <- y[!train]

  X.train <- X[train,]
  X.test <- X[!train,]

  ridge.model <- glmnet(X.train, y.train, alpha = 0, lambda = ridge.model$lambda.min)
  lasso.model <- glmnet(X.train, y.train, alpha = 1, lambda = lasso.model$lambda.min)
  ols.model <- lm(y.train ~ ., data = data.frame(X.train))


  models <- list(ridge = ridge.model, lasso = lasso.model, ols = ols.model)

  test.mse <- c()
  test.mse[1] <- unname(assess.glmnet(ridge.model, newx = X.test, newy = y.test)$mse[1])
  test.mse[2] <- unname(assess.glmnet(lasso.model, newx = X.test, newy = y.test)$mse[1])

  pred <- predict.lm(ols.model, newdata = data.frame(X.test))
  test.mse[3] <- mean((pred - y.test)^2)

  min.mse <- which.min(test.mse)

  if (min.mse == 1) {
    return(glmnet(X, y, alpha = 0, lambda = ridge.lambda, intercept = FALSE))
  }

  if (min.mse == 2) {
    return(glmnet(X, y, alpha = 1, lambda = lasso.lambda, intercept = FALSE))
  }

  return(lm(y ~ ., data = data.frame(X), y = TRUE))
}



explanatory_helper <- function(X, y) {

  lasso.model <- cv.glmnet(X, y, alpha = 1)
  lasso.model <- glmnet(X, y, alpha = 1, lambda = lasso.model$lambda.min, intercept = FALSE)
  lasso.coef <- coef(lasso.model)
  relevant.vars <- names(lasso.coef[lasso.coef[, 1] != 0,])
  relevant.vars <- relevant.vars[-1]
```

```
  X <- subset(X, select = relevant.vars)

  lm(y ~ ., data = data.frame(X), y = TRUE)
}



cleanup_helper <- function(X, y, na.tolerance = 0) {

  n <- dim(X)[1]
  na.sums <- colSums(is.na(X))
  na.sums <- (na.sums/n) < na.tolerance
  summary(na.sums)

  for (i in 1:dim(X)[2]) {
    if (na.sums[[i]]) {
      col.na <- !is.na(X[, i])
      y <- y[col.na]
      X <- X[col.na,]
    }
  }


  for (i in 1:dim(X)[2]) {
    if (class(X[, i]) == "factor") {
      X[, i] <- factor(X[, i], levels = c(levels(X[, i]), NA),
                       labels = c(levels(X[, i]), "None"), exclude = NULL)
    }
    if (is.numeric(X[, i])){
      X[, i][is.na(X[, i])] <- 0
    }
  }

  X <- model.matrix(~ ., X)
  output <- list()
  output$X <- X
  output$y <- y

  output
}



important_function <- function(X, y, na.tolerance, model.type) {

  cleaned <- cleanup_helper(X, y)
  X <- cleaned$X
  y <- cleaned$y

  if (model.type == "explanatory") {
    model <- explanatory_helper(X, y)
  } else if (model.type == "predictive") {
    model <- predictive_helper(X, y)
```

```
  }

  fitted.values <- predict(model, newx = X)

  model.coefficients <- coef(model)

  if ("lm" %in% class(model)) {
    variables.used <- names(model.coefficients)[-1]
    variables.used <- variables.used[-1]
  } else {
    model.coef <- coef(model)
    variables.used <- names(model.coef[model.coef[, 1] != 0,])
  }

  data.used <- subset(X, select = variables.used)
  outliers <- outlier_helper(data.used, y, fitted.values)

  diagnostics <- diagnostic_helper(fitted.values, y)

  X.data.frame <- data.frame(X)

  if ("lm" %in% class(model)) {
    diagnostics$bp <- bptest(model)
    box <- boxcox(model)
  } else {
    box <- boxcox(lm(y ~ ., data = data.frame(X), y = TRUE))
  }

  bc.lambda <- box.lambda <- box$x[which.max(box$y)]
  suggested.transform <- (y ^ box.lambda - 1) / box.lambda

  list(model = model, diagnostics = diagnostics, outliers = outliers, transform = suggested.transform)
}
```

Loading the Dataset

Since we are primarily interested in systolic blood pressure, we chose to remove diastolic blood pressure readings from the data set to reduce confusion. We also removed the ID column. Our function cleans the data and removes NAs so we don't need to do any further data cleaning before running the function.

```
important_data <- read.table("/Users/perli/OneDrive/Documents/MATH 372 - Desk/final project/cardio_trai
                             sep = ";", header = TRUE, stringsAsFactors = TRUE)

cardio <- read.table("/Users/perli/OneDrive/Documents/MATH 372 - Desk/final project/cardio_train.csv",
                     header = T, sep = ';', stringsAsFactors = T)

# im removing ID
cardio <- cardio[,-1]

# choosing y and X
blood_pressure <- cardio$ap_hi # y / response value
cardio.X <- cardio[, !names(cardio) %in%
```

```
                    c("ap_hi","ap_lo")] # removing response and ap_lo

invalid <- which(blood_pressure <= 0)
cardio.X <- cardio.X[-invalid,]
blood_pressure <- blood_pressure[-invalid]
```

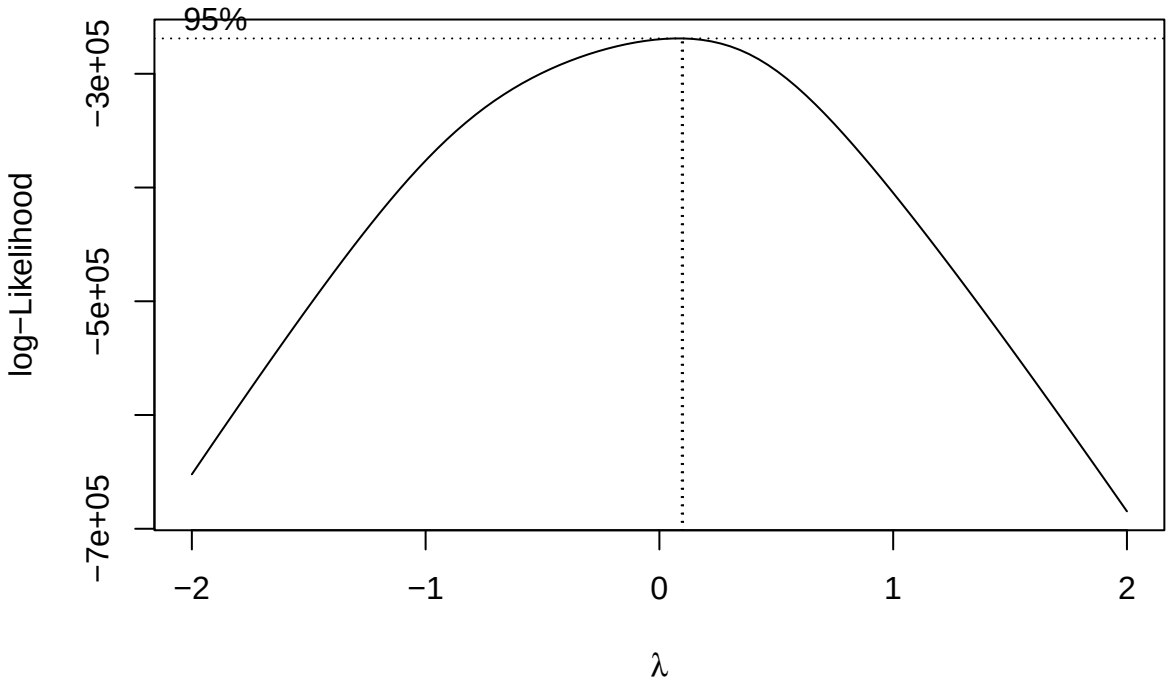
Performing Predictive Modeling

When running our function for prediction, we found that the best predictive model uses a Ridge regression. Of the three models used (i.e Ridge, LASSO, OLS) it had the lowest MSE. The R2 tells us that this model explains 41% of the variance in the data and we can see from the log likelihood plot that the best lambda is 0.5649.

```
output <- important_function(cardio.X, blood_pressure, 0, 'predictive')
```

```
## Warning in predict.lm(ols.model, newdata = data.frame(X.test)): prediction from
## a rank-deficient fit may be misleading
```

```
## Warning in ks.test.default(residuals, "pnorm"): ties should not be present for
## the Kolmogorov-Smirnov test
```



```
predictive.model <- output$model
predictive.model
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(X), y = TRUE)
##
## Coefficients:
##  (Intercept)  X.Intercept.           age        gender        height
##   93.0853921            NA     0.0004323     1.3509265    -0.0069981
##       weight   cholesterol          gluc         smoke          alco
##    0.2135694     2.1016814     0.1715151    -1.5900541     0.3465983
##       active         cardio
##    0.7613376    14.4984969
```

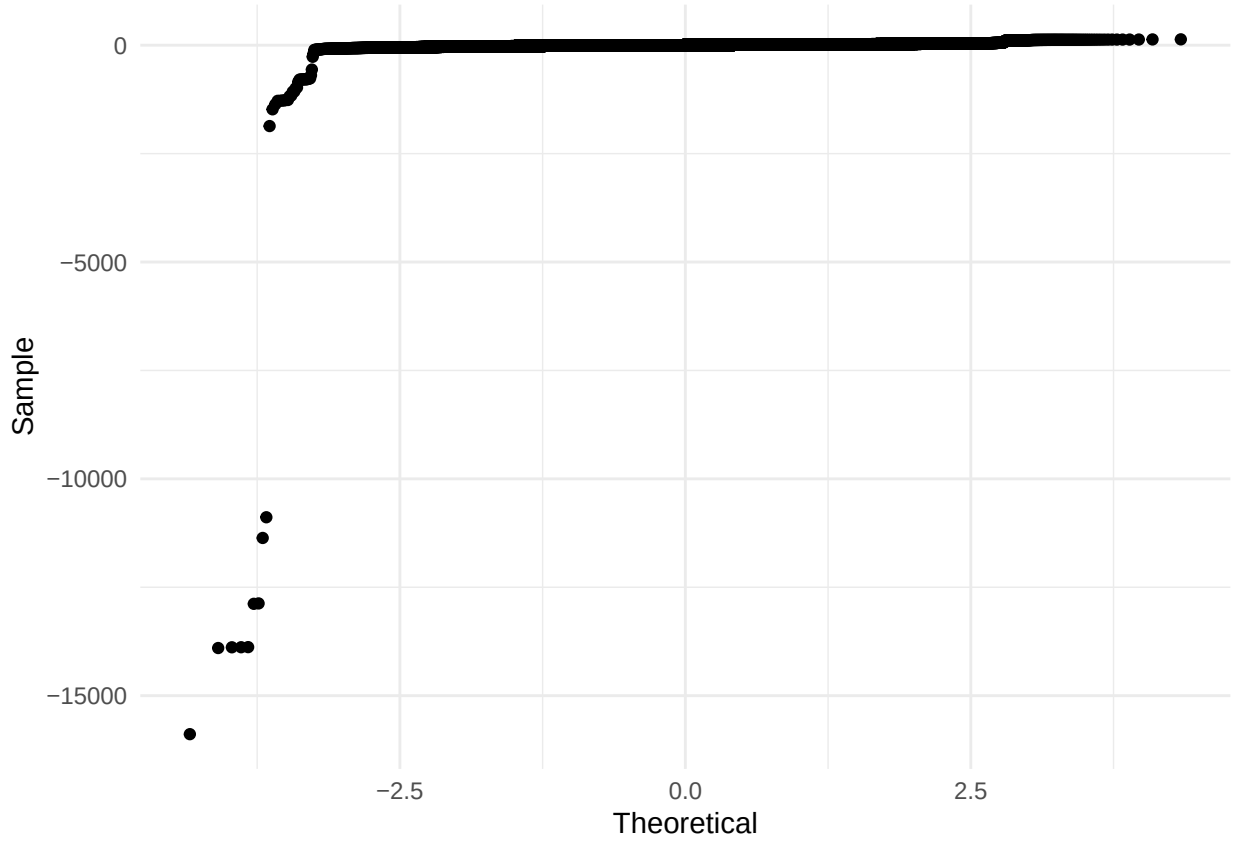
```
predictive.model$beta
```

```
## NULL
```

Initial Diagnostics of Predictive Model

Our initial evaluation of the diagnostic plots for this model tells us that this data isn't normally distributed and does not have constant variance. We hope that by removing outliers and doing a square-root y transformation we will be able to resolve these issues and increase the accuracy of our predictions. We will then run a Kolmogorov-Smirnov test for normality to confirm our results.

```
predictive.diagnostics <- output$diagnostics
predictive.diagnostics$qq
```



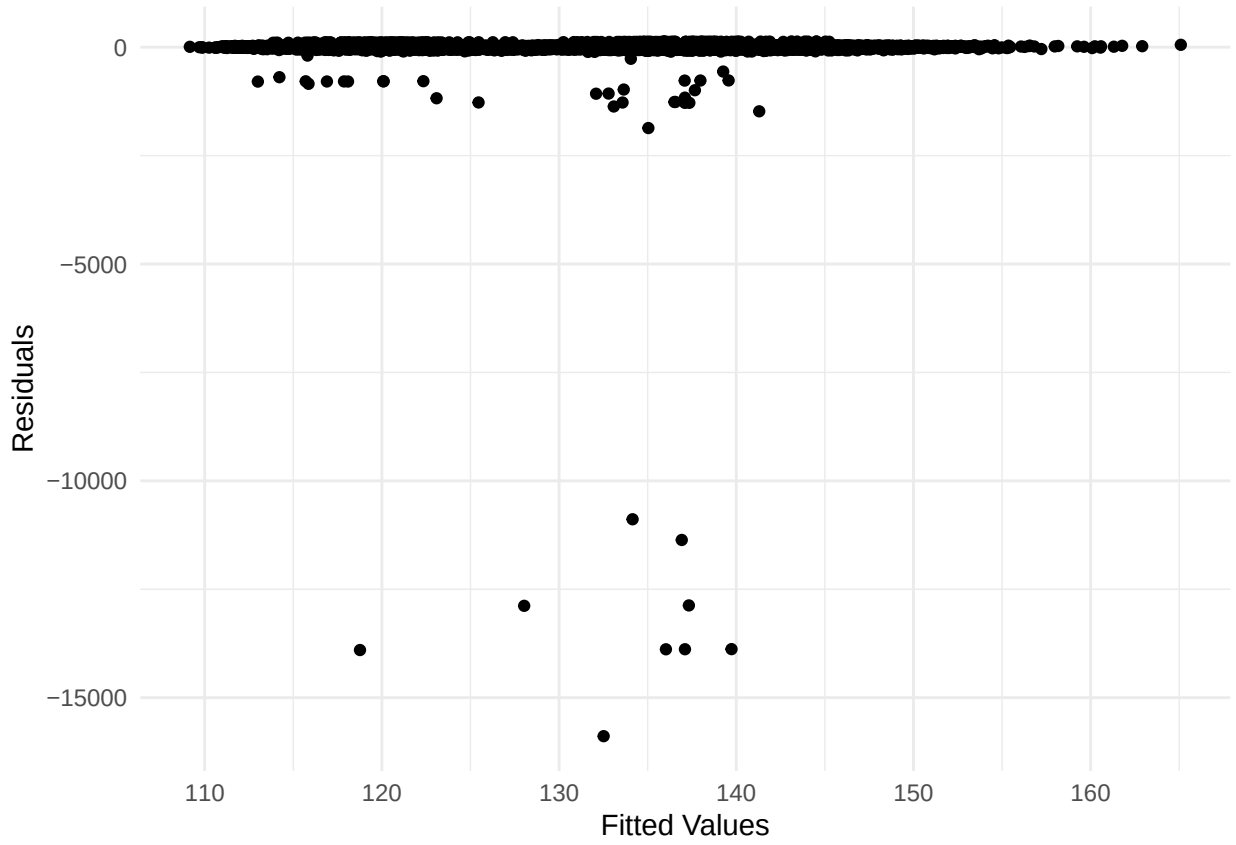
We can see here, by looking at the KS statistic, that D is not as small as we would like. This further implies non-normality in model.

```
predictive.diagnostics$ks
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  residuals
## D = 0.46528, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

And here we can see that the plot of fitted values against residuals does not provide us with constant variance across the data.

```
predictive.diagnostics$cv
```



Removing Outliers & Conducting Transformations

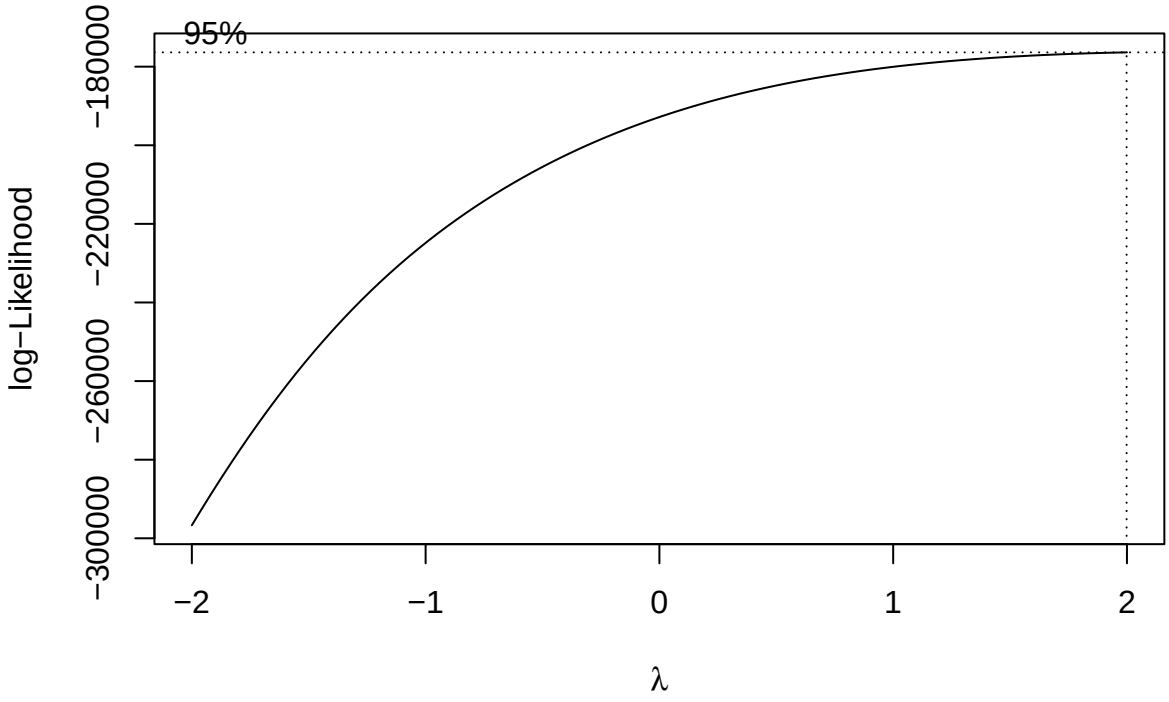
When we remove outliers and refit the data to a new predictive model we see a huge improvement. Once again, our function gives us the best model, based on MSE, and we can tell it has improved in accuracy because the F-statistic is much larger. Additionally, the R2 tells us that this model explains 99% of the variance in the data! With the outliers removed the model also gives us a much smaller lambda value of 0.0008.

```
outliers <- output$outliers$outliers
outlier.data <- cardio.X[-outliers,]
outlier.response <- blood_pressure[-outliers]

outlier.output <- important_function(outlier.data, sqrt(outlier.response), 0, 'predictive')
```

```
## Warning in predict.lm(ols.model, newdata = data.frame(X.test)): prediction from
## a rank-deficient fit may be misleading
```

```
## Warning in ks.test.default(residuals, "pnorm"): ties should not be present for
## the Kolmogorov-Smirnov test
```




```
outlier.output$model
```

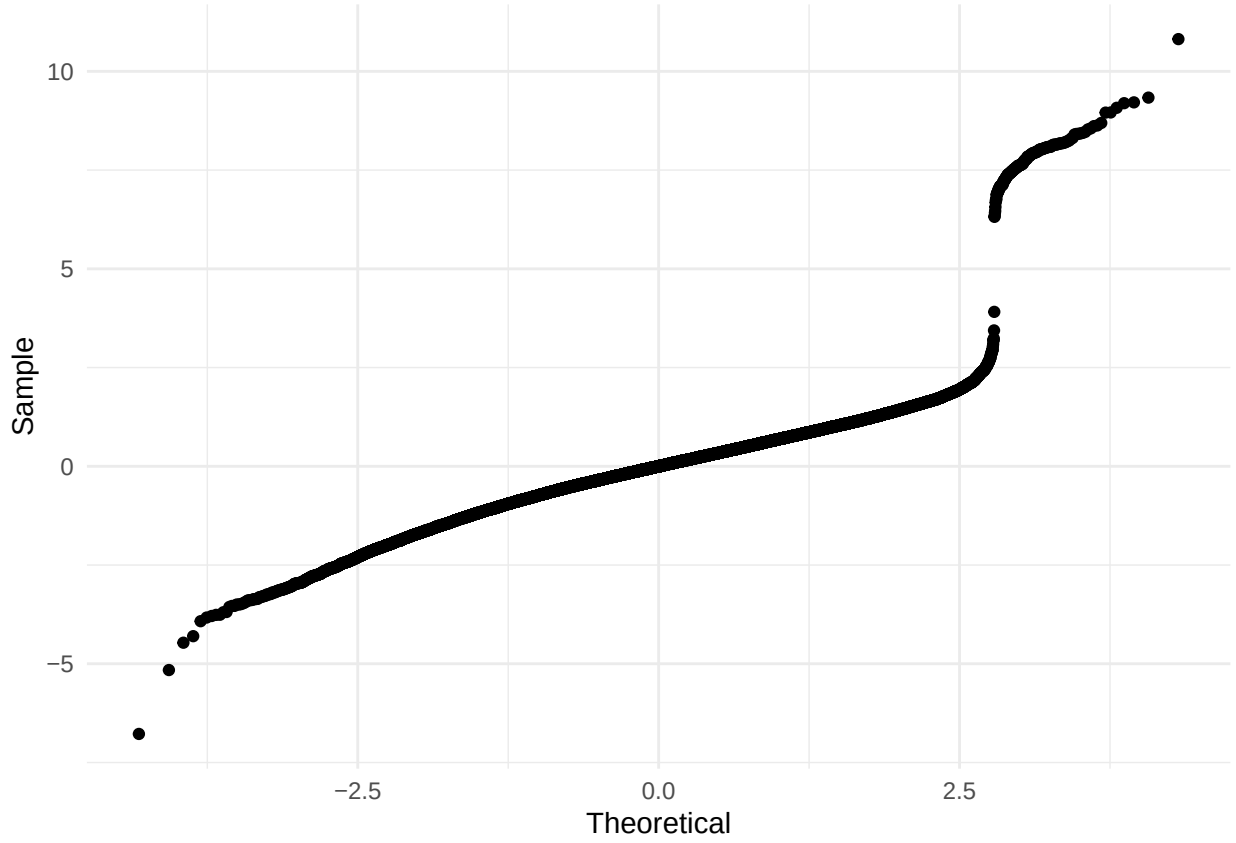
```
##
## Call:  glmnet(x = X, y = y, alpha = 0, lambda = ridge.lambda, intercept = FALSE)
##
##   Df  %Dev  Lambda
## 1  9 99.43 0.03176
```

```
summary(outlier.output$model)
```

```
##           Length Class     Mode
## a0         1     -none-    numeric
## beta      11     dgCMatrix S4
## df         1     -none-    numeric
## dim        2     -none-    numeric
## lambda     1     -none-    numeric
## dev.ratio  1     -none-    numeric
## nulldev    1     -none-    numeric
## npasses    1     -none-    numeric
## jerr       1     -none-    numeric
## offset     1     -none-    logical
## call       6     -none-    call
## nobs       1     -none-    numeric
```

After removing outliers and performing a square root transformation our diagnostic plots look a lot better. The QQ plot looks much more linear, indicating that the data distribution is much closer to normal.

```
outlier.output$diagnostics$qq
```



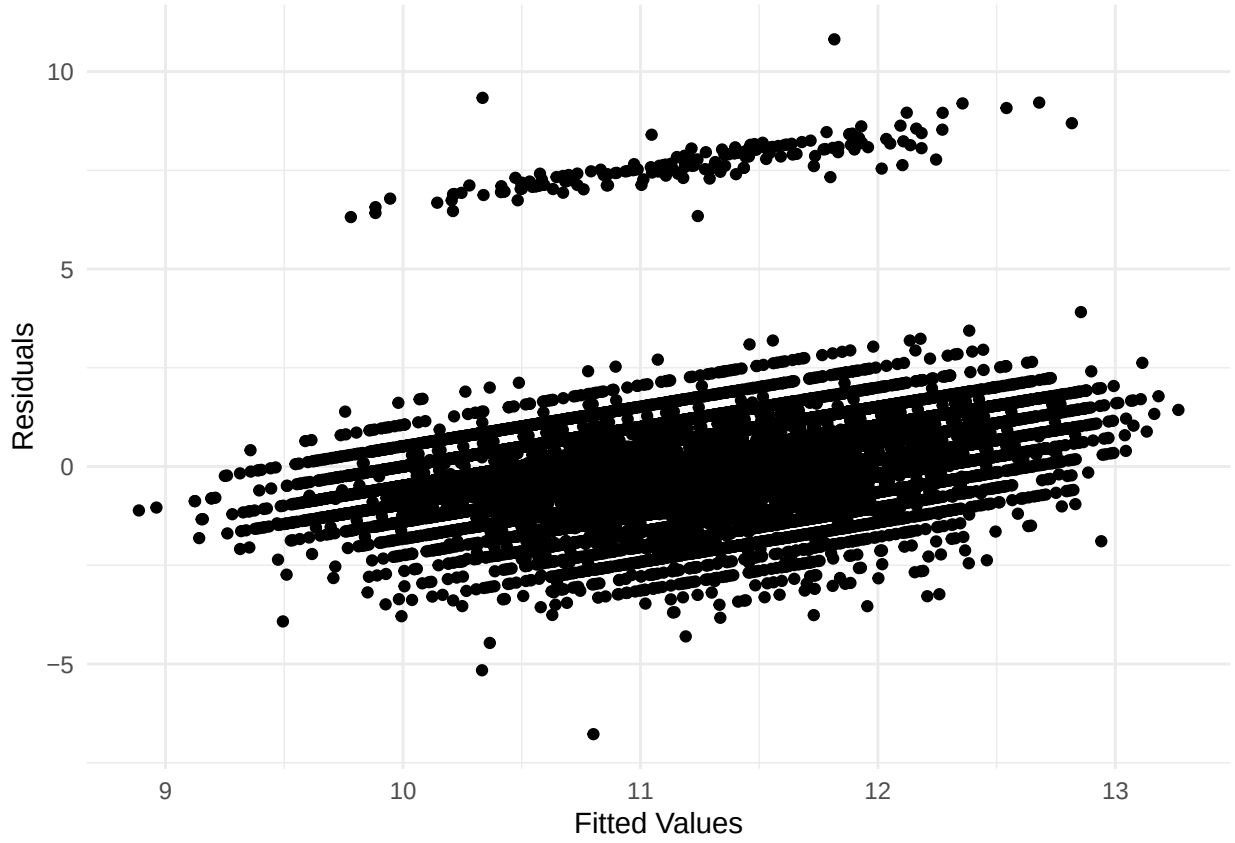
In running another Kolmogorov-Smirnov test we can see that D has become significantly smaller. We are now able to accept H0 for normality.

```
outlier.output$diagnostics$ks
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  residuals
## D = 0.089469, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Examining the fitted vs. residuals plot we are also able to confirm homoskedasticity.

```
outlier.output$diagnostics$cv
```

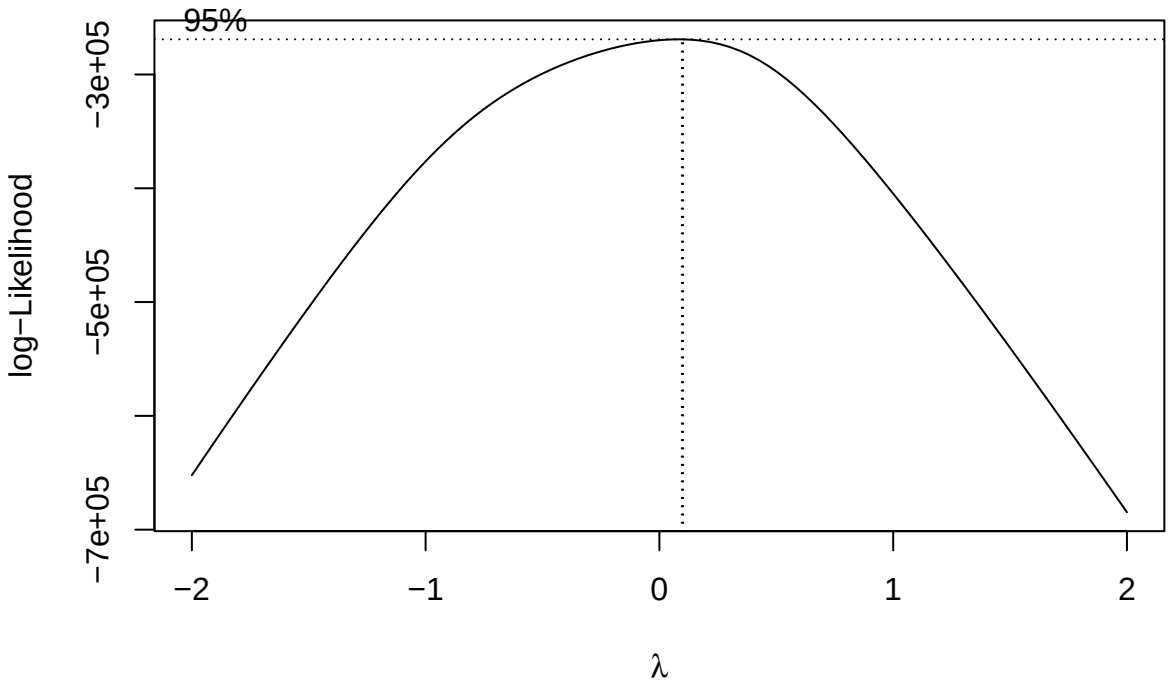


Performing Explanatory Modeling

Just like when we ran our function of predictive modeling, our function cleans our data and gives us the best model for the job. When we run our function and request explanatory output instead of predictive output we are given the results of running Lasso on our data. We are given the following log likelihood plot.

```
explan.output <- important_function(cardio.X, blood_pressure, 0, 'explanatory')
```

```
## Warning in ks.test.default(residuals, "pnorm"): ties should not be present for
## the Kolmogorov-Smirnov test
```



Model Selection

In running Lasso, we are given 8 of the original 10 variables. The adjusted R Squared is 0.0034 which is not great. And the F-Statistic is 30.99 on 8 and 69984 DF, which is also not great. We'd like both of these to be higher so we will remove outliers, like we did for the predicitive model, and see what happens.

```
best.model <- explan.output$model
summary(best.model)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(X), y = TRUE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -135.4   -12.1    -1.5     4.6 15886.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 102.71674   12.84877   7.994 1.32e-15 ***
## gender        1.43906    1.46894   0.980   0.3273
## height       -0.01882    0.08473  -0.222   0.8242
## weight        0.21548    0.04342   4.963 6.96e-07 ***
## cholesterol   2.23860    0.98371   2.276   0.0229 *
```

```
## gluc          0.24182    1.13987   0.212   0.8320
## smoke        -1.65830    2.18095  -0.760   0.4470
## active        0.75823    1.46504   0.518   0.6048
## cardio       14.94362    1.20955  12.355  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 153.7 on 69984 degrees of freedom
## Multiple R-squared:  0.00353,    Adjusted R-squared:  0.003416
## F-statistic: 30.99 on 8 and 69984 DF,  p-value: < 2.2e-16
```

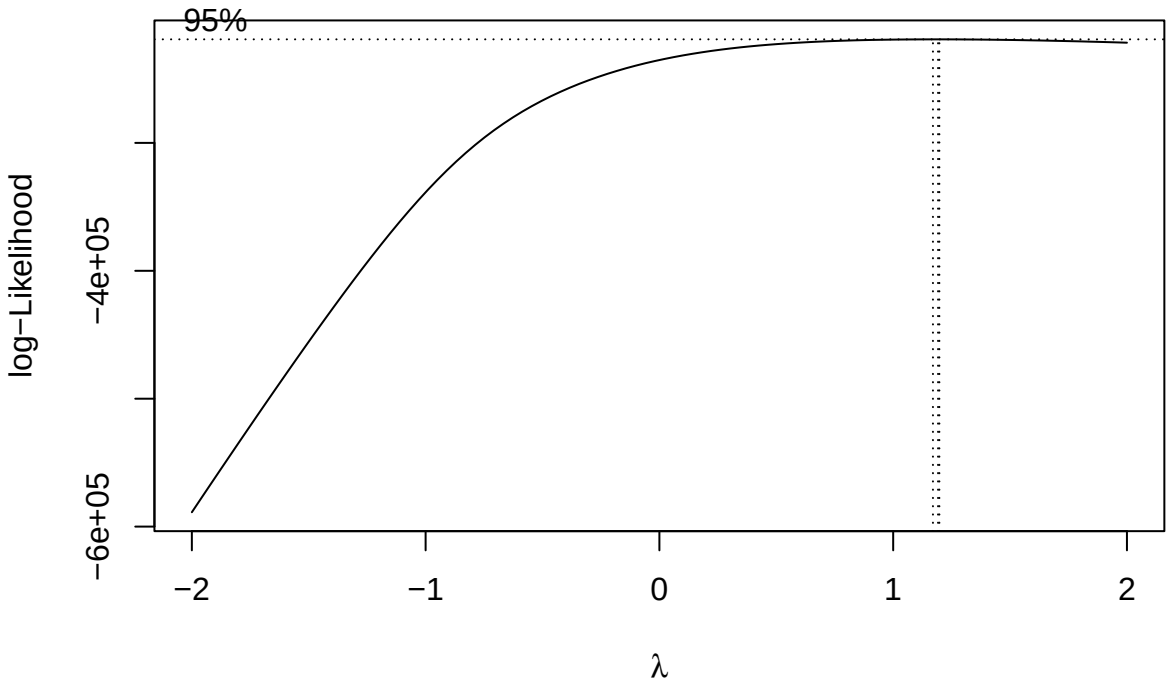
Removing Outliers from Explanatory Model

We removed the outliers and refit the data with Lasso. This ended up adding in another variable to our model. In checking the Adjusted R Squared and F-statistic we can see that these SIGNIFICANTLY increased to 0.2101 and 1870 on 9 and 63247 DF, respectively. Therefore, we rejust the first model and continue with this one.

```
outliers.explan <- explan.output$outliers$outliers
outlierExplan.data <- cardio.X[-outliers.explan,]
outlierExplan.response <- blood_pressure[-outliers.explan]

outlierExplan.output <- important_function(outlierExplan.data, outlierExplan.response, 0, 'explanatory')
```

```
## Warning in ks.test.default(residuals, "pnorm"): ties should not be present for
## the Kolmogorov-Smirnov test
```




```
best.outlier.model <- outlierExplan.output$model
summary(best.outlier.model)
```

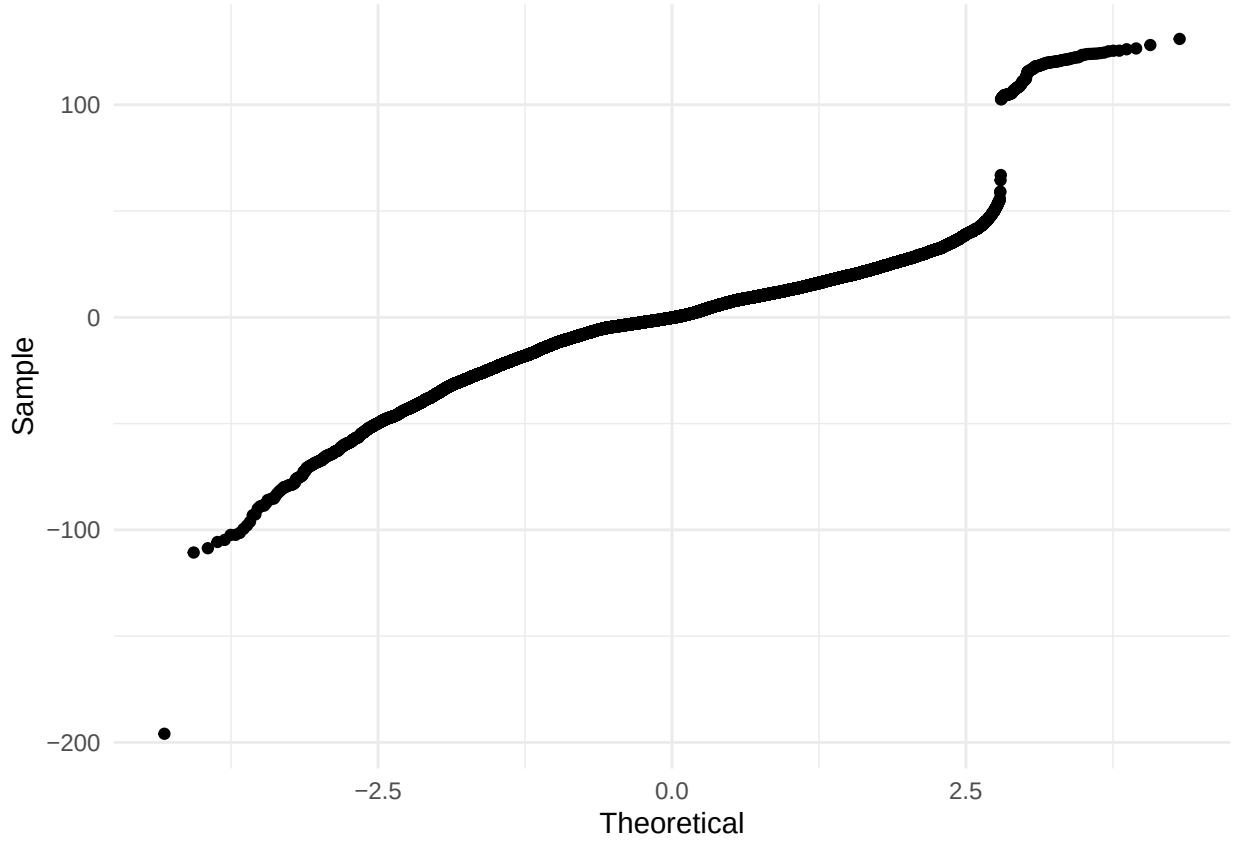
```
##
## Call:
## lm(formula = y ~ ., data = data.frame(X), y = TRUE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -130.953   -9.312    0.205    6.673  195.937
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 123.732067   1.546137  80.027  < 2e-16 ***
## gender        2.286895   0.161355  14.173  < 2e-16 ***
## height       -0.175132   0.010236 -17.109  < 2e-16 ***
## weight        0.252766   0.005336  47.373  < 2e-16 ***
## cholesterol   2.048473   0.120229  17.038  < 2e-16 ***
## gluc          0.128279   0.167672   0.765 0.444237
## smoke         0.655593   0.304195   2.155 0.031152 *
## alco          0.756910   0.322075   2.350 0.018771 *
## active        0.599998   0.167244   3.588 0.000334 ***
## cardio       12.883578   0.132171  97.477  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 15.87 on 63247 degrees of freedom
## Multiple R-squared:  0.2102, Adjusted R-squared:  0.2101
## F-statistic:  1870 on 9 and 63247 DF,  p-value: < 2.2e-16
```

Diagnostics and Transformations on Explanatory Model

Now that we have a model we are happy with, it's time to run diagnostics. We already removed outliers so we hope that our QQ plots will look pretty linear.However, we notice that some points are still a little off, thus our distribution is not entirely normal.

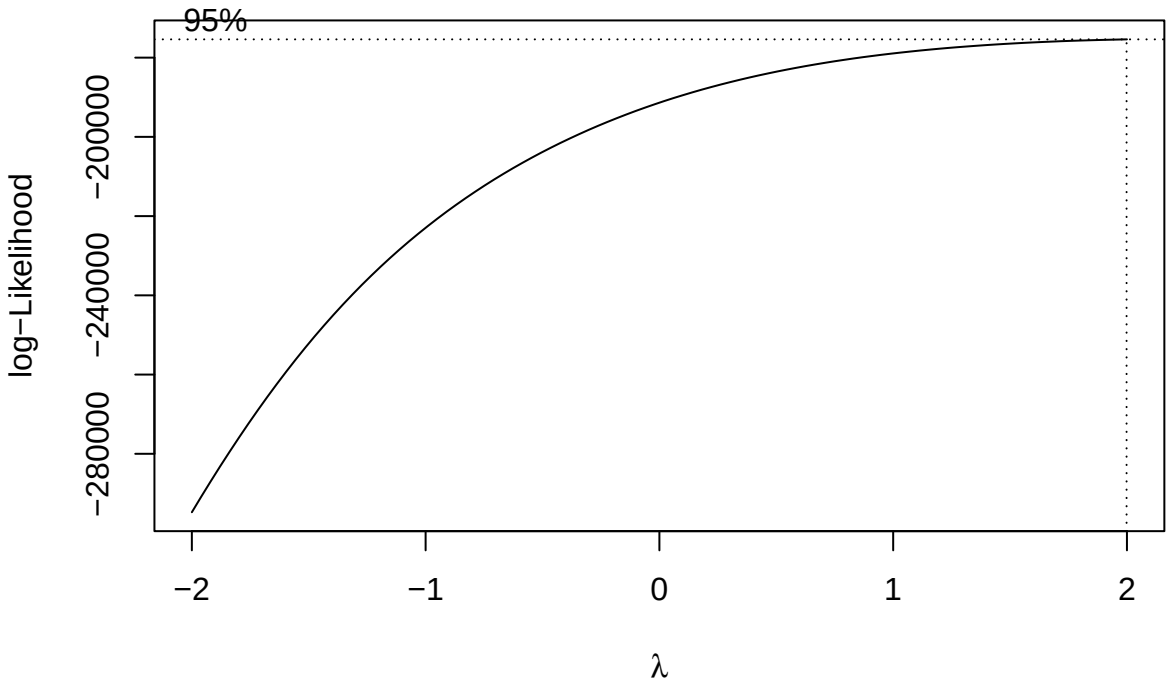
```
outlierExplan.output$diagnostics$qq
```



We will perform a Square root transformation on y and re-run the model in the hopes of achieving normality. The Adjusted R Squared and F-Statistics on this new model are slightly lower, but not by much, meaning they are still sufficiently large. The QQ plot looks a lot better and since we need normality to do any sort of inference, we will go with this model and perform a KS test.

```
explan.output.transform1 <- important_function(outlierExplan.data, sqrt(outlierExplan.response), 0, 'exp
```

```
## Warning in ks.test.default(residuals, "pnorm"): ties should not be present for
## the Kolmogorov-Smirnov test
```

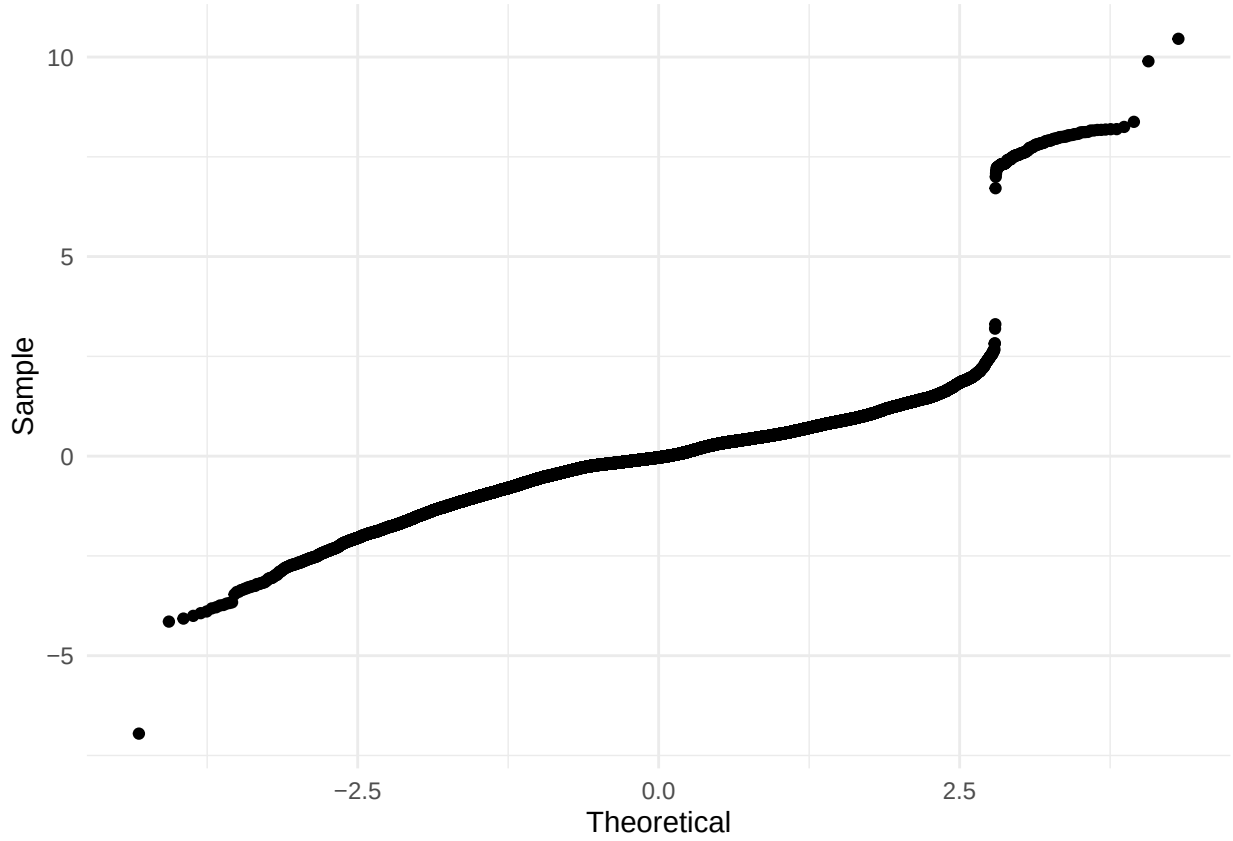



```
best.model.transform1 <- explan.output.transform1$model
summary(best.model.transform1)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(X), y = TRUE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4566  -0.3946   0.0333   0.3178   6.9525
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.0469287  0.0732106 150.893  < 2e-16 ***
## gender       0.1007945  0.0076403  13.193  < 2e-16 ***
## height      -0.0074218  0.0004847 -15.313  < 2e-16 ***
## weight       0.0111305  0.0002526  44.056  < 2e-16 ***
## cholesterol  0.0904030  0.0056929  15.880  < 2e-16 ***
## gluc         0.0059419  0.0079394   0.748 0.454218
## smoke        0.0292585  0.0144038   2.031 0.042228 *
## alco         0.0303287  0.0152504   1.989 0.046738 *
## active       0.0275853  0.0079191   3.483 0.000495 ***
## cardio       0.5636573  0.0062584  90.064  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7516 on 63247 degrees of freedom
## Multiple R-squared:  0.1858, Adjusted R-squared:  0.1857
## F-statistic:  1603 on 9 and 63247 DF,  p-value: < 2.2e-16
```

```
explan.output.transform1$diagnostics$qq
```



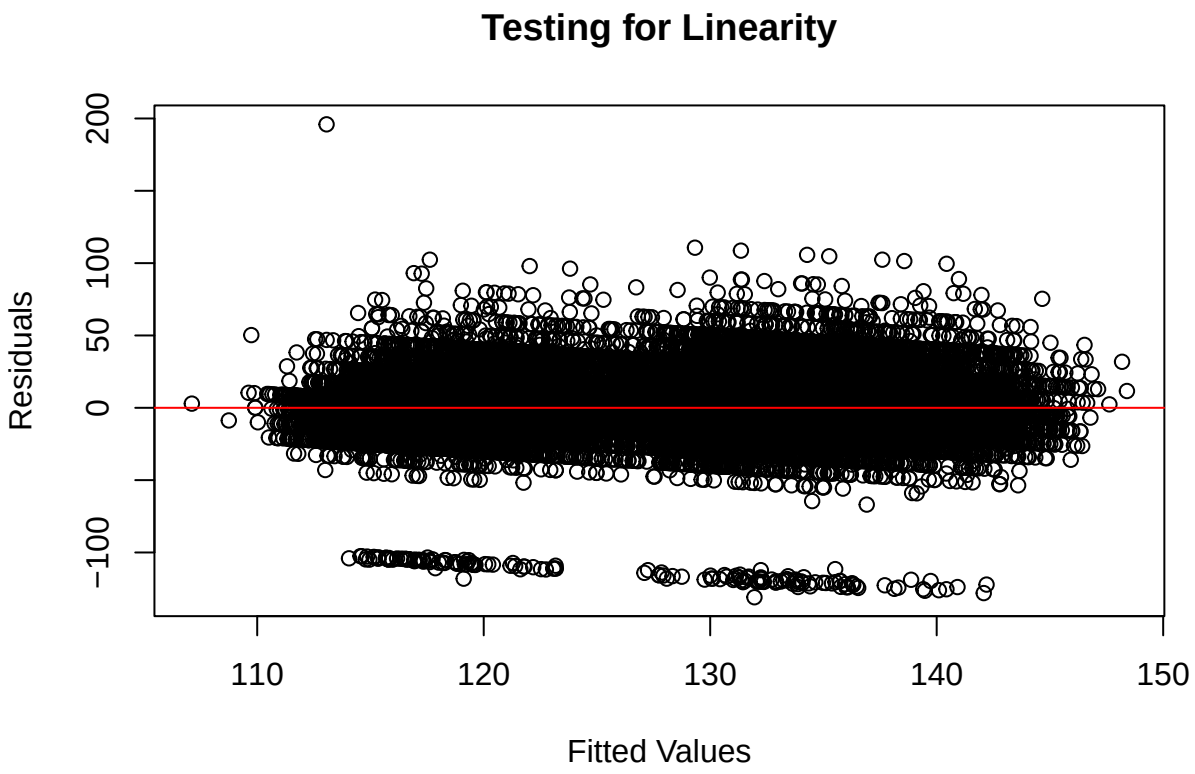
In running a KS test we can see that the D value is 0.136, which is sufficiently small for us accept normality.

```
explan.output.transform1$diagnostics$ks
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  residuals
## D = 0.13601, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

In performing a test for linearity we can see that the plot is relatively linear with a few values that are not constant, indicating heteroskedasticity. When we look at correlation it is VERy close to zero so linearity can be confirmed without a Square root transformation.

```
plot(best.outlier.model$fitted.values, best.outlier.model$residuals,
     xlab = 'Fitted Values', ylab = 'Residuals', main = 'Testing for Linearity')
abline(h = mean(best.outlier.model$residuals), col = "red")
```

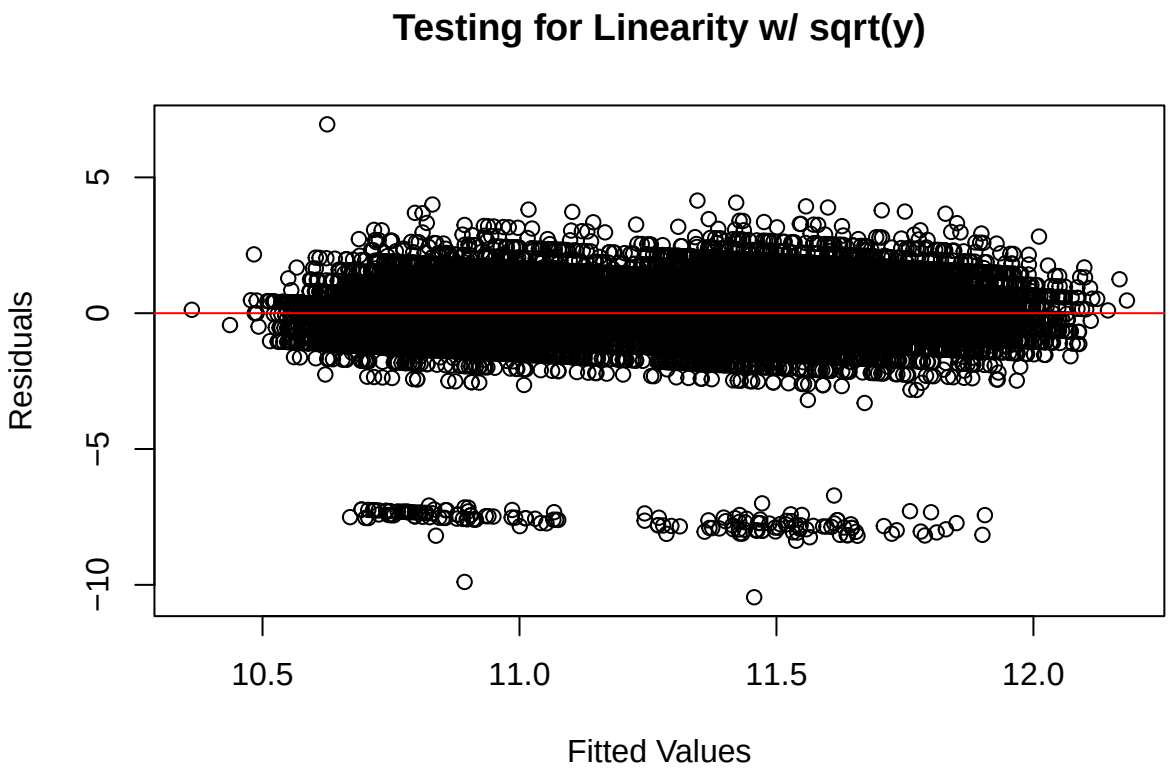



```
cor(best.outlier.model$fitted.values, best.outlier.model$residuals)
```

```
## [1] -1.373107e-14
```

We decided to try looking at the data with a transform anyway, but it looks worse and the correlation is higher, so it seems that by removing the outliers we did everything necessary to capture trends in the data.

```
plot(best.model.transform1$fitted.values, best.model.transform1$residuals,
     xlab = 'Fitted Values', ylab = 'Residuals', main = 'Testing for Linearity w/ sqrt(y)')
abline(h = mean(best.model.transform1$residuals), col = "red")
```

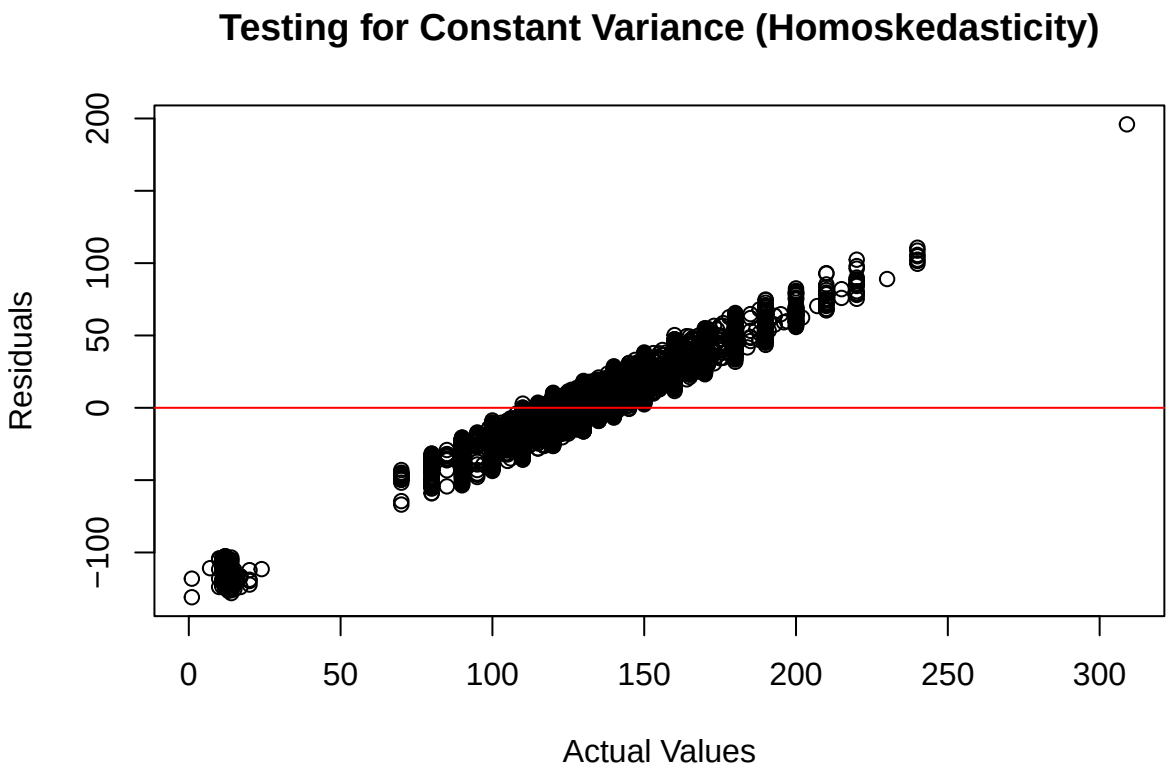



```
cor(best.model.transform1$fitted.values, best.model.transform1$residuals)
```

```
## [1] -1.567336e-14
```

Now we test for homoskedasticity or constant variance y looking at a plot of response values vs. residuals. The points appear to be spread out evenly, but in a linear fashion. This is ok, but not ideal. We will perform a Breush-Pagan test to confirm that variance is constant.

```
plot(outlierExplan.response, best.outlier.model$residuals,
     xlab = "Actual Values", ylab = 'Residuals',
     main = 'Testing for Constant Variance (Homoskedasticity)')
abline(h = mean(best.outlier.model$residuals), col = "red")
```



In running a Breusch-Pagan statistic to test for constant variance in current explanatory model we find that the p-value is significally less than 0.05. This proves there is constant variance and we do not reject H0; there is no need for a transformation.

```
outlierExplan.output$diagnostics$bp
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 698.9, df = 9, p-value < 2.2e-16
```

Conclusion

After cleaning, processing, diagnosing, and analyzing this data set, we found both a model that best predicts high systolic blood pressure and a model that best explains high systolic blood pressure. Through the process of diagnostics, we discovered that this model is very sensitive to outliers and removing them was neccesarity to get any kind of insight.

Our explanatory model tells us the most significant factors associated with high blood pressure are gender, height, weight, cholesterol, activity level and the presence of cardiovascular disease. While smoking and drinking alcohol had some effect, surprisingly, these habits didn't appear to be as strongly correlated with high blood pressure as we expected.

Since gender, height, and pre-existing cardiovascular disease are not factors that subjects have control over, we suggest that further study be done to examine the efficacy of a treatment plan that includes diet changes

to target weight and cholesterol, coupled with increases activity level.