Mulyiple Linear Regression

Kaia Lindberg

12/7/2021

Set up

```
# Import libraries
library(tidyverse)
library(ROCR)
library(MASS)
```

```
# Load data
Data <- read.csv("diabetes2.csv", header=T)
print(head(Data))
```

```
##   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
```

The data has 768 rows and 9 columns. One of these columns (Outcome) will be our response variable for our logistic regression and the others are potential predictors.

```
# Check dimensions of data
print(dim(Data)) # 768 rows and 9 columns
```

```
## [1] 768   9
```

```
# Display names of all columns
print(colnames(Data))
```

```
## [1] "Pregnancies"              "Glucose"
## [3] "BloodPressure"            "SkinThickness"
## [5] "Insulin"                  "BMI"
## [7] "DiabetesPedigreeFunction" "Age"
## [9] "Outcome"
```

Before we go any further with our analysis we will split our data into a training and test set. We've chosen a

random seed of "123" for reproducability. We will do all further analysis, visualization, and model building using our training data and then use our test data to evaluate our model's performance on unseen data.

```
# Randomly split data into two halves
set.seed(123) # For reproducability
sample<-sample.int(nrow(Data), floor(.50*nrow(Data)), replace = F)
train<-Data[sample, ] # Training data
test<-Data[-sample, ] # Test data
```

Data Cleaning

```
# List of columns where zero is non-sensical (i.e. zero indicates unknown)
# All predictor columns other than pregnancies
zero_unknown_cols <-c("Glucose", "BloodPressure", "SkinThickness",
                      "Insulin", "BMI", "DiabetesPedigreeFunction",
                      "Age")
train[,zero_unknown_cols] <- replace(train[,zero_unknown_cols], train[,zero_unknown_cols]==0,NA)
test[,zero_unknown_cols] <- replace(test[,zero_unknown_cols], test[,zero_unknown_cols]==0,NA)
```

```
# Remove variables with high percent missing
train <- dplyr::select(train, -c('SkinThickness', 'Insulin'))
```

```
# For the remaining missing values (<5% in any column) I'll impute with the median
fill_missing_cols <-c("Glucose", "BloodPressure", "BMI",
                      "DiabetesPedigreeFunction", "Age")
for(i in fill_missing_cols) {
  train[ , i][is.na(train[ , i])] <- median(train[ , i], na.rm=TRUE)
}

# Do the same for the test data
for(i in fill_missing_cols) {
  test[ , i][is.na(test[ , i])] <- median(test[ , i], na.rm=TRUE)
}
```

Linear Regression Model

Fit Initial Model

```
# Fit initial regression model
train <- dplyr::select(train, -c('Outcome')) # Remove Outcome from potential predictors
full <- lm(Glucose~., data=train)
summary(full)
```

```
##
## Call:
## lm(formula = Glucose ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -73.809 -18.393  -2.232  16.309  84.525
##
```

```
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               55.7038     9.8363   5.663 2.95e-08 ***
## Pregnancies               -0.1169     0.4767  -0.245 0.806321
## BloodPressure              0.1570     0.1369   1.146 0.252424
## BMI                        1.0095     0.2221   4.546 7.37e-06 ***
## DiabetesPedigreeFunction   8.2598     4.3734   1.889 0.059707 .
## Age                        0.5618     0.1443   3.894 0.000116 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.83 on 378 degrees of freedom
## Multiple R-squared:  0.1414, Adjusted R-squared:  0.1301
## F-statistic: 12.45 on 5 and 378 DF,  p-value: 3.367e-11
```

Only BMI and Age are statistically signficant in predicting Glucose. Let's test whether we can drop Pregnancies, BloodPressure, and DiabetesPedigreeFunction from our model.

Fit reduced model

```
# Fit a reduced model using only BMI and age
reduced <- lm(Glucose~BMI + Age, data=train)
summary(reduced)
```

```
##
## Call:
## lm(formula = Glucose ~ BMI + Age, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -77.397 -18.414  -2.315  16.603  84.185
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  64.5906     7.8399   8.239 2.83e-15 ***
## BMI           1.1416     0.2026   5.635 3.40e-08 ***
## Age           0.6107     0.1210   5.048 6.92e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.89 on 381 degrees of freedom
## Multiple R-squared:  0.1312, Adjusted R-squared:  0.1266
## F-statistic: 28.76 on 2 and 381 DF,  p-value: 2.338e-12
```

```
# Partial f test
anova(reduced, full)
```

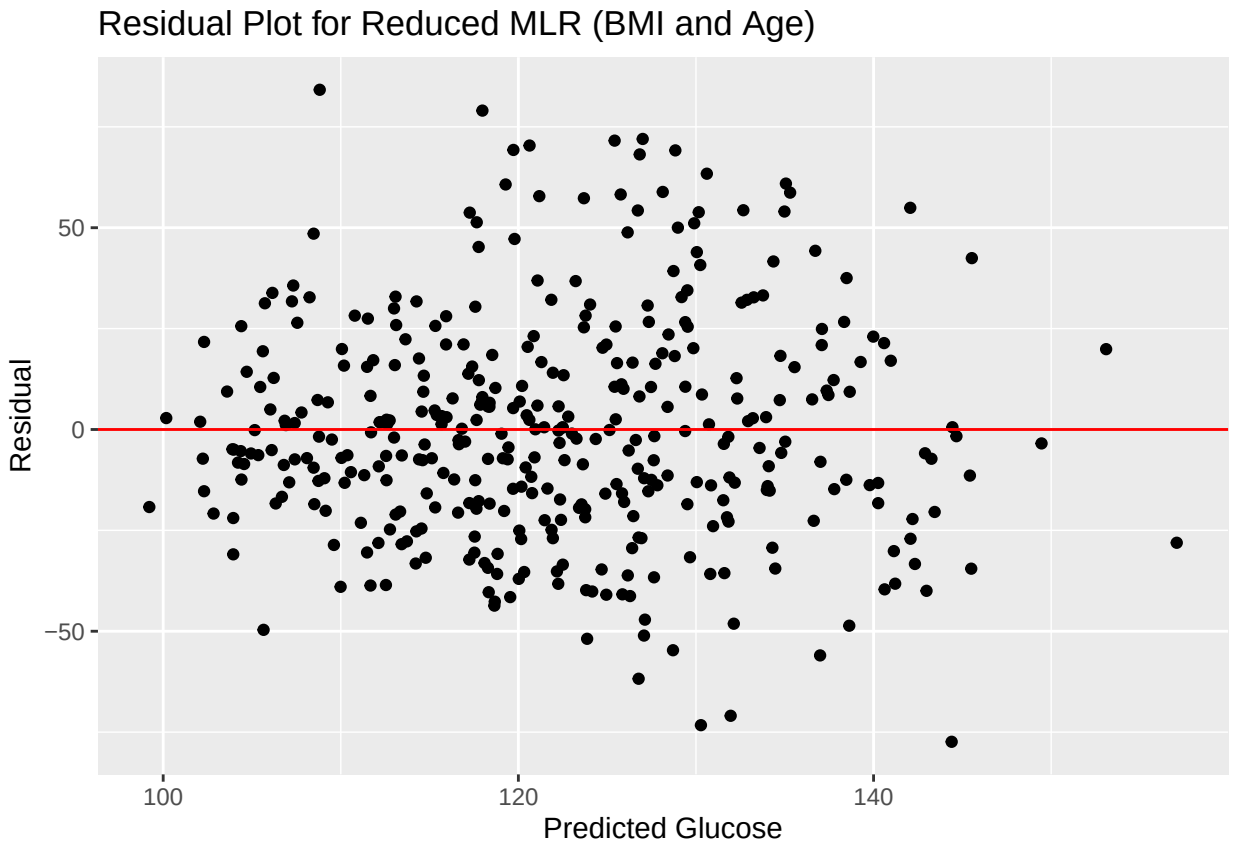
```
## Analysis of Variance Table
##
## Model 1: Glucose ~ BMI + Age
## Model 2: Glucose ~ Pregnancies + BloodPressure + BMI + DiabetesPedigreeFunction +
##     Age
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    381 296368
```

```
## 2    378 292869  3    3499.2 1.5054 0.2127
```

This p-value is larger than an alpha of 0.05 so we fail to reject the null hypothesis. We do not have sufficient evidence to support the claim that at least one of the coefficients in the null hypothesis is non-zero and thus the simpler model (using only BMI and age) is sufficient.

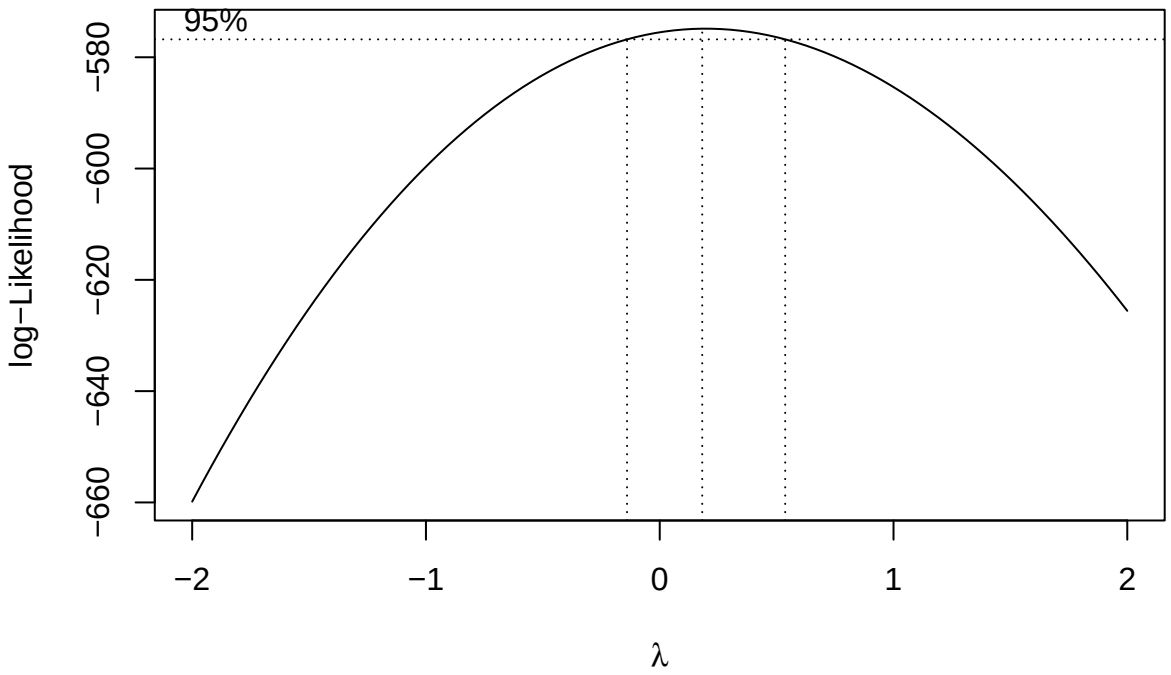
Check regression assumptions

```
# Calculate fitted y and residual
yhat <- reduced$fitted.values
residual <- reduced$residuals
# Add to data
train <- data.frame(train, yhat, residual)
# Create residual plot
ggplot(train, aes(x=yhat, y=residual)) +
geom_point() +
geom_hline(yintercept=0, color="red") +
labs(x="Predicted Glucose", y= "Residual", title="Residual Plot for Reduced MLR (BMI and Age)")
```



The residual plot for this reduced model seems to have non-constant variance as the residuals appear closer to 0 for low predicted glucose and further away from 0 (larger variance) for larger values of predicted glucose. However, there does not appear to be any pattern to the residuals so I believe mean zero assumption is met.

```
boxcox(reduced, lambda = seq(-2,2))
```



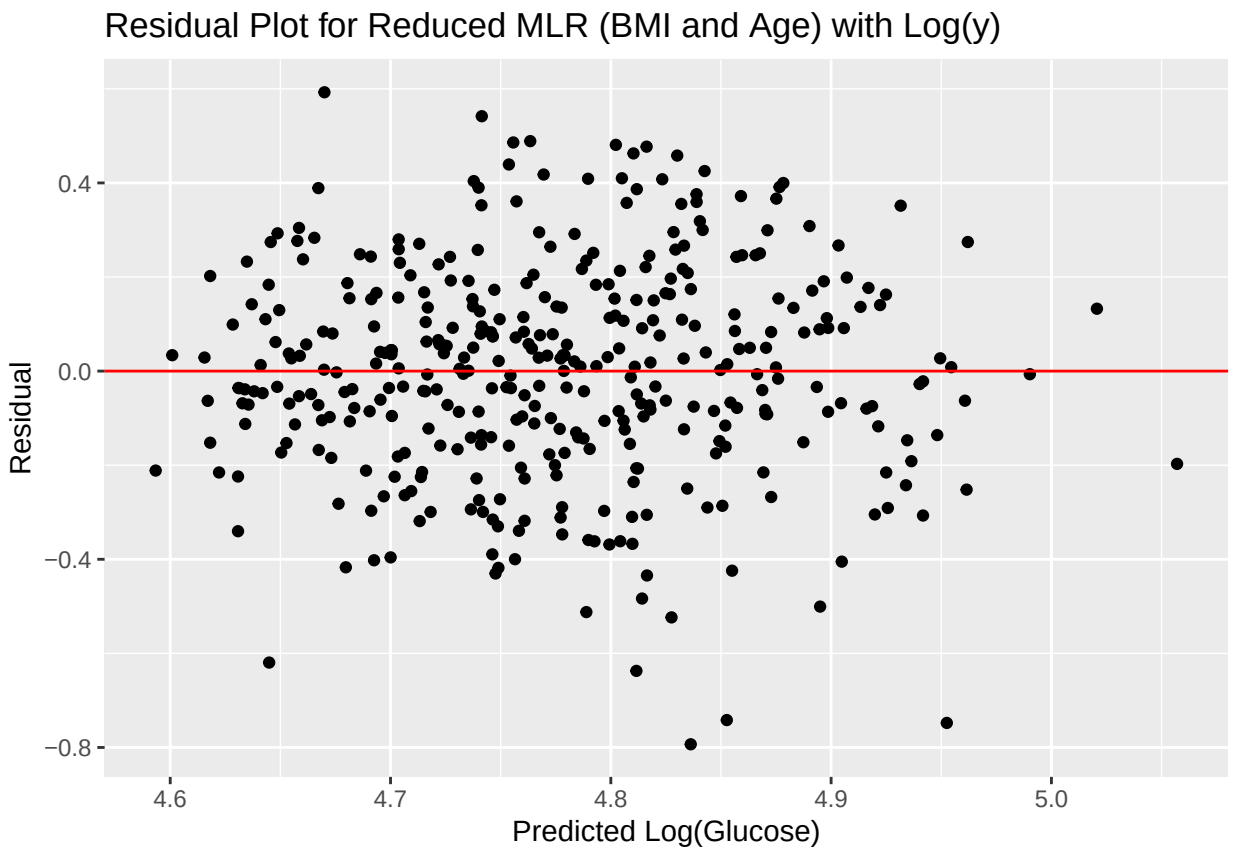
The interval for the box cox plot does not include 1, suggesting that a y-transformation is warranted and could help improve the constant variance assumption. Since 0 is in the interval will do a log transformation, this will also let us maintain interpretability of our coefficients.

```
# Transform response variable
train <- train %>%
  mutate(ystar = log(Glucose))
# Fit reduced model with transformed y
reduced.ystar<-lm(ystar~BMI + Age, data=train)
summary(reduced.ystar)
```

```
##
## Call:
## lm(formula = ystar ~ BMI + Age, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79322 -0.14097  0.00808  0.15338  0.59269
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.3183959  0.0644599  66.994  < 2e-16 ***
## BMI         0.0091628  0.0016656   5.501 6.94e-08 ***
## Age         0.0047619  0.0009946   4.788 2.42e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2293 on 381 degrees of freedom
## Multiple R-squared:  0.123,  Adjusted R-squared:  0.1184
## F-statistic: 26.72 on 2 and 381 DF,  p-value: 1.385e-11
```

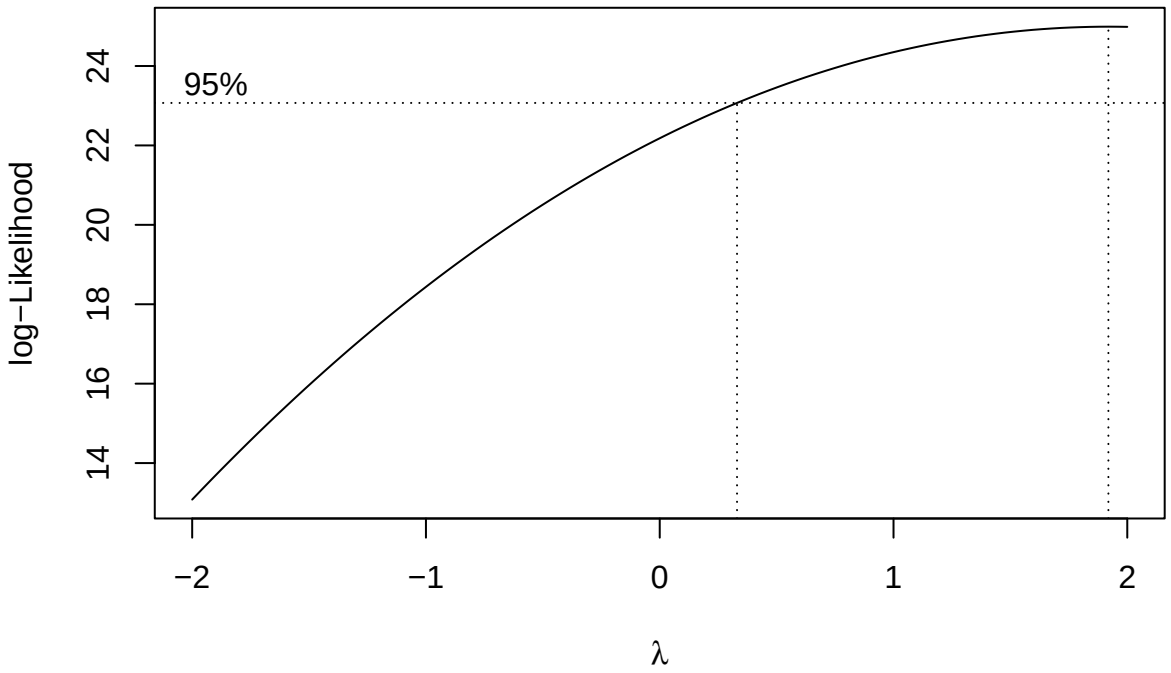
```
# Calculate fitted y and residual for log(y) model
yhat.ystar <- reduced.ystar$fitted.values
residual.ystar <- reduced.ystar$residuals
```

```r
# Add to data
train <- data.frame(train, yhat.ystar, residual.ystar)
# Create residual plot
ggplot(train, aes(x=yhat.ystar, y=residual.ystar)) +
geom_point() +
geom_hline(yintercept=0, color="red") +
labs(x="Predicted Log(Glucose)", y= "Residual", title="Residual Plot for Reduced MLR (BMI and Age) with
```

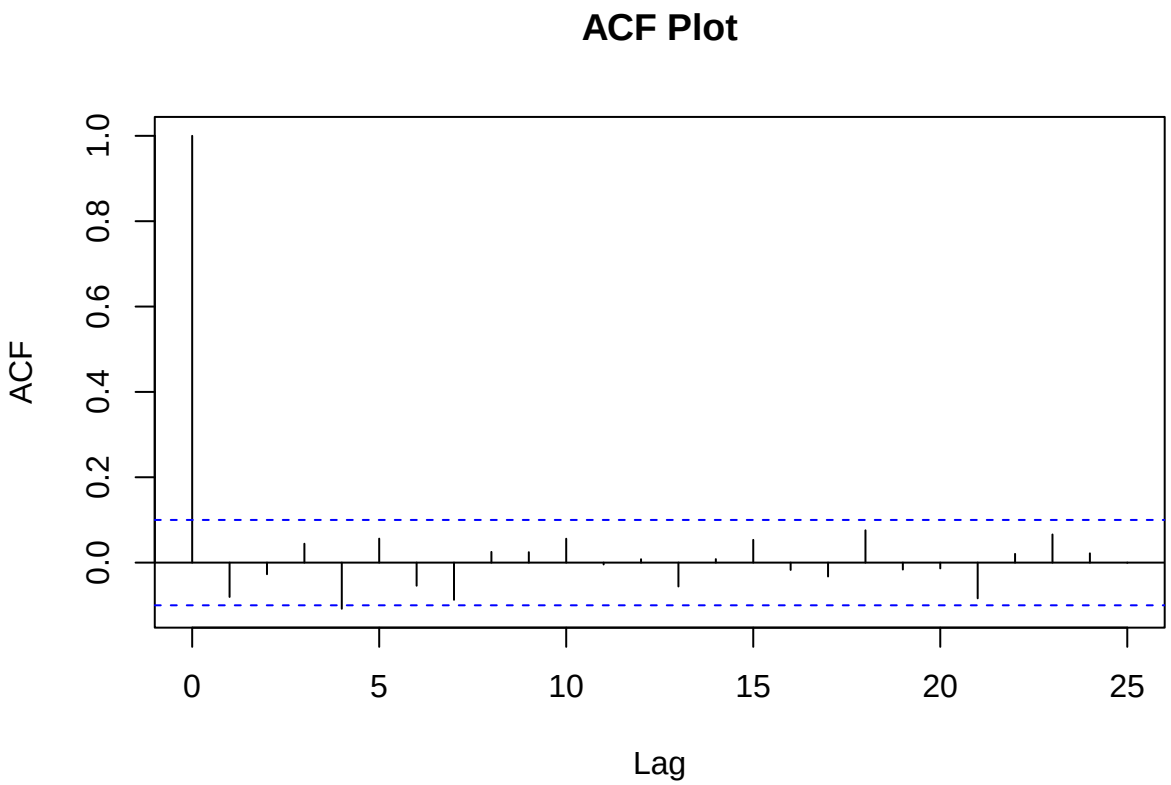


This residual plot looks better. Variance is more constant than before. Confirmed with box cox plot below in which 1 is in the interval and thus we do not need to transform y any further. There also does not appear to be any pattern/shape to the residuals and thus I think the mean zero assumption is met and thus we don't need to transform any of our predictors.

```r
boxcox(reduced.ystar, lambda = seq(-2,2))
```

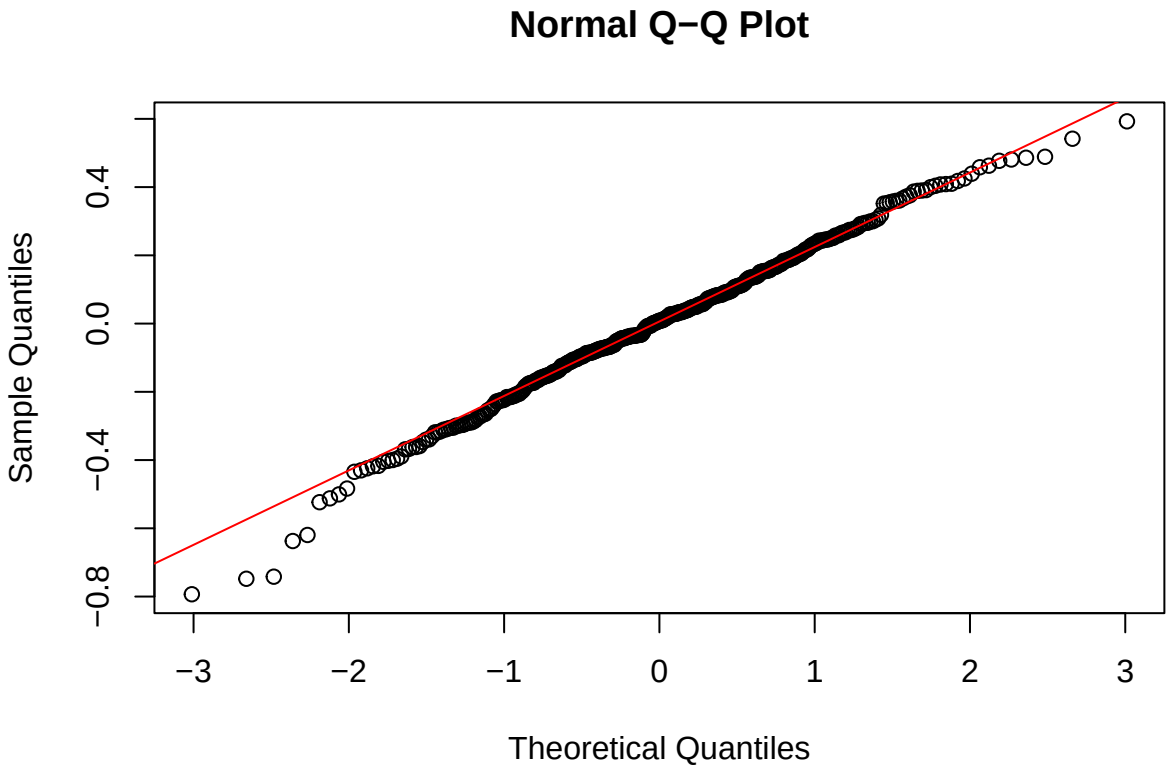



```
acf(train$residual.ystar, main="ACF Plot") #Create ACF plot to see if errors are independent
```



ACF slightly exceeds interval at lag 4, but is minor and all other lags are fine so I'd say this assumption is met.

```
#Check that errors are normally distributed
qqnorm(train$residual.ystar)
qqline(train$residual.ystar, col="red")
```



Based on the QQ plot, the observations generally follow the theoretical values (red straight line) fairly well. There are some minor deviations in the tails, but for the most part the observations follow the red line very well, which suggests that the normality assumption is met and the errors are normally distributed.

Some other exploration

Don't need to use this, just wanted to test out.

Re-check other predictors with transformed y

Now that we've transformed our y variable, I wonder if that changes any of the variables that were insignificant before?

```
# Fit full model with transformed y
full.ystar<-lm(ystar~BMI + Age + Pregnancies + BloodPressure + DiabetesPedigreeFunction, data=train)
summary(full.ystar)
```

```
##
## Call:
## lm(formula = ystar ~ BMI + Age + Pregnancies + BloodPressure +
##     DiabetesPedigreeFunction, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78794 -0.14276  0.00435  0.15633  0.60176
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)               4.2357789  0.0808839  52.369  < 2e-16 ***
## BMI                       0.0079166  0.0018261   4.335 1.87e-05 ***
## Age                       0.0042340  0.0011862   3.569 0.000404 ***
## Pregnancies              -0.0005785  0.0039196  -0.148 0.882752
## BloodPressure             0.0015735  0.0011261   1.397 0.163132
## DiabetesPedigreeFunction  0.0616796  0.0359628   1.715 0.087147 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2289 on 378 degrees of freedom
## Multiple R-squared:  0.1331, Adjusted R-squared:  0.1217
## F-statistic: 11.61 on 5 and 378 DF,  p-value: 1.888e-10
```

No, that did not change anything. Other predictors are still insignificant. Double checked with partial f test, but that suggests we can drop all three predictors (Pregnancies, BloodPressure, and DiabetesPedigreeFunction).

```
# Partial f test
anova(reduced.ystar, full.ystar)
```

```
## Analysis of Variance Table
##
## Model 1: ystar ~ BMI + Age
## Model 2: ystar ~ BMI + Age + Pregnancies + BloodPressure + DiabetesPedigreeFunction
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    381 20.035
## 2    378 19.803  3   0.23174 1.4745 0.2211
```

Automated search procedure(s)

Backward elimination

```
# Declare intercept only model
regnull <- lm(ystar~1, data=train)
# Declare full model
regfull <- lm(ystar~ BMI + Age + Pregnancies + BloodPressure + DiabetesPedigreeFunction, data=train)
# Run backward elimination
step(regfull, scope=list(lower=regnull, upper=regfull), direction="backward")
```

```
## Start:  AIC=-1126.48
## ystar ~ BMI + Age + Pregnancies + BloodPressure + DiabetesPedigreeFunction
##
##                            Df Sum of Sq    RSS     AIC
## - Pregnancies               1   0.00114 19.804 -1128.5
## - BloodPressure             1   0.10229 19.905 -1126.5
## <none>                                  19.803 -1126.5
## - DiabetesPedigreeFunction  1   0.15411 19.957 -1125.5
## - Age                       1   0.66743 20.471 -1115.8
## - BMI                       1   0.98463 20.788 -1109.8
##
## Step:  AIC=-1128.46
## ystar ~ BMI + Age + BloodPressure + DiabetesPedigreeFunction
##
##                            Df Sum of Sq    RSS     AIC
## - BloodPressure             1   0.10125 19.905 -1128.5
```

```
## <none>                                19.804 -1128.5
## - DiabetesPedigreeFunction  1   0.15589 19.960 -1127.5
## - Age                       1   0.81296 20.617 -1115.0
## - BMI                       1   0.98692 20.791 -1111.8
##
## Step:  AIC=-1128.5
## ystar ~ BMI + Age + DiabetesPedigreeFunction
##
##                             Df Sum of Sq    RSS     AIC
## <none>                                    19.905 -1128.5
## - DiabetesPedigreeFunction  1   0.12935 20.035 -1128.0
## - Age                       1   1.13463 21.040 -1109.2
## - BMI                       1   1.51058 21.416 -1102.4
##
## Call:
## lm(formula = ystar ~ BMI + Age + DiabetesPedigreeFunction, data = train)
##
## Coefficients:
##              (Intercept)                       BMI                       Age
##                 4.302842                  0.008955                  0.004635
## DiabetesPedigreeFunction
##                 0.056035
```

Backward selection would suggest we include DiabetesPedigreeFunction as well.

Forward selection

```
# Run forward selection
step(regnull, scope=list(lower=regnull, upper=regfull), direction="forward")
```

```
## Start:  AIC=-1081.62
## ystar ~ 1
##
##                             Df Sum of Sq    RSS     AIC
## + BMI                       1   1.60444 21.240 -1107.6
## + Age                       1   1.21854 21.626 -1100.7
## + BloodPressure             1   1.09534 21.749 -1098.5
## + DiabetesPedigreeFunction  1   0.29852 22.546 -1084.7
## + Pregnancies               1   0.26711 22.578 -1084.1
## <none>                                  22.845 -1081.6
##
## Step:  AIC=-1107.58
## ystar ~ BMI
##
##                             Df Sum of Sq    RSS     AIC
## + Age                       1   1.20543 20.035 -1128.0
## + BloodPressure             1   0.37134 20.869 -1112.3
## + Pregnancies               1   0.26302 20.977 -1110.4
## + DiabetesPedigreeFunction  1   0.20014 21.040 -1109.2
## <none>                                  21.240 -1107.6
##
## Step:  AIC=-1128.02
## ystar ~ BMI + Age
```

```
##
##                             Df Sum of Sq    RSS     AIC
## + DiabetesPedigreeFunction  1  0.129347 19.905 -1128.5
## <none>                                  20.035 -1128.0
## + BloodPressure             1  0.074708 19.960 -1127.5
## + Pregnancies               1  0.001016 20.034 -1126.0
##
## Step:  AIC=-1128.5
## ystar ~ BMI + Age + DiabetesPedigreeFunction
##
##                 Df Sum of Sq    RSS     AIC
## <none>                       19.905 -1128.5
## + BloodPressure  1  0.101253 19.804 -1128.5
## + Pregnancies    1  0.000102 19.905 -1126.5
```

```
##
## Call:
## lm(formula = ystar ~ BMI + Age + DiabetesPedigreeFunction, data = train)
##
## Coefficients:
##              (Intercept)                       BMI                       Age
##                 4.302842                  0.008955                  0.004635
## DiabetesPedigreeFunction
##                 0.056035
```

Forward selection also suggests we include DiabetesPedigreeFunction.

```
# Run stepwise selection
step(regnull, scope=list(lower=regnull, upper=regfull), direction="both")
```

```
## Start:  AIC=-1081.62
## ystar ~ 1
##
##                             Df Sum of Sq    RSS     AIC
## + BMI                        1   1.60444 21.240 -1107.6
## + Age                        1   1.21854 21.626 -1100.7
## + BloodPressure              1   1.09534 21.749 -1098.5
## + DiabetesPedigreeFunction   1   0.29852 22.546 -1084.7
## + Pregnancies                1   0.26711 22.578 -1084.1
## <none>                                   22.845 -1081.6
##
## Step:  AIC=-1107.58
## ystar ~ BMI
##
##                             Df Sum of Sq    RSS     AIC
## + Age                        1   1.20543 20.035 -1128.0
## + BloodPressure              1   0.37134 20.869 -1112.3
## + Pregnancies                1   0.26302 20.977 -1110.4
## + DiabetesPedigreeFunction   1   0.20014 21.040 -1109.2
## <none>                                   21.240 -1107.6
## - BMI                        1   1.60444 22.845 -1081.6
##
## Step:  AIC=-1128.02
## ystar ~ BMI + Age
##
##                             Df Sum of Sq    RSS     AIC
```

```
## + DiabetesPedigreeFunction  1   0.12935 19.905 -1128.5
## <none>                                  20.035 -1128.0
## + BloodPressure             1   0.07471 19.960 -1127.5
## + Pregnancies               1   0.00102 20.034 -1126.0
## - Age                       1   1.20543 21.240 -1107.6
## - BMI                       1   1.59133 21.626 -1100.7
##
## Step:  AIC=-1128.5
## ystar ~ BMI + Age + DiabetesPedigreeFunction
##
##                            Df Sum of Sq    RSS     AIC
## <none>                                  19.905 -1128.5
## + BloodPressure             1   0.10125 19.804 -1128.5
## - DiabetesPedigreeFunction  1   0.12935 20.035 -1128.0
## + Pregnancies               1   0.00010 19.905 -1126.5
## - Age                       1   1.13463 21.040 -1109.2
## - BMI                       1   1.51058 21.416 -1102.4
```

```
##
## Call:
## lm(formula = ystar ~ BMI + Age + DiabetesPedigreeFunction, data = train)
##
## Coefficients:
##              (Intercept)                       BMI                       Age
##                 4.302842                  0.008955                  0.004635
## DiabetesPedigreeFunction
##                 0.056035
```

Same model again, including DiabetesPedigreeFunction. So let's fit that model and check if DiabetesPedigreeFunction. is signficant.

```
# Fit model with transformed y
# Reduced model plus DiabetesPedigreeFunction.
reduced.ystar.dpf<-lm(ystar~BMI + Age + DiabetesPedigreeFunction, data=train)
summary(reduced.ystar.dpf)
```

```
##
## Call:
## lm(formula = ystar ~ BMI + Age + DiabetesPedigreeFunction, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.80586 -0.14017  0.00816  0.15237  0.57996
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               4.3028422  0.0650929  66.103  < 2e-16 ***
## BMI                       0.0089554  0.0016677   5.370 1.37e-07 ***
## Age                       0.0046352  0.0009959   4.654 4.50e-06 ***
## DiabetesPedigreeFunction 0.0560350  0.0356595   1.571    0.117
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2289 on 380 degrees of freedom
## Multiple R-squared:  0.1287, Adjusted R-squared:  0.1218
## F-statistic:  18.7 on 3 and 380 DF,  p-value: 2.458e-11
```

This summary still suggests that we can drop DiabetesPedigreeFunction as it does not add much value in predicting Glucose when Age and BMI are already fit in the model.