Week 10 Report

Dylan Denner

3/30/2021

## Text, letter Description automatically generated

Text, letter

Description automatically generated

A picture containing table

Description automatically generated

Text, letter

Description automatically generated

Text, letter

Description automatically generated

## Applied Homework

1. In Chapter 4, we used logistic regression to predict the probability of default using income and balance on the Default data set. We will now estimate the test error of this logistic regression model using the validation set approach. Do not forget to set a random seed before beginning your analysis.
2. **Fit a logistic regression model that uses income and balance to predict default.**

Library(ISLR)  
data <- Default  
  
log\_model <- glm(default ~ income + balance, data = data, family = binomial)

1. Using the validation set approach, estimate the test error of this model.

Set.seed(100)  
  
training\_data\_index <- sample(10000, 5000, replace = FALSE)  
  
train\_model <- glm(default ~ income + balance, data = data, family = binomial, subset = training\_data\_index)  
  
predict\_prob <- predict(train\_model, data[-training\_data\_index,], type = “response”)  
  
test\_data <- data[-training\_data\_index,]  
  
test\_data$predict\_prob <- predict\_prob  
test\_data$predict\_default <- “No”  
test\_data$predict\_default[test\_data$predict\_prob > 0.5] <- “Yes”  
  
df <- subset(test\_data, test\_data$default != test\_data$predict\_default)  
  
test\_error\_rate = length(df$default)/length(test\_data$default)  
test\_error\_rate

**## [1] 0.0264**

Repeat the process in (b) three times, using three different splits of the observations into a training set and a validation set. Comment on the results obtained.

training\_data\_index2 <- sample(10000, 5000, replace = FALSE)  
  
train\_model2 <- glm(default ~ income + balance, data = data, family = binomial, subset = training\_data\_index2)  
  
predict\_prob2 <- predict(train\_model2, data[-training\_data\_index2,], type = "response")  
  
test\_data2 <- data[-training\_data\_index2,]  
  
test\_data2$predict\_prob2 <- predict\_prob2  
test\_data2$predict\_default2 <- "No"  
test\_data2$predict\_default2[test\_data2$predict\_prob2 > 0.5] <- "Yes"  
  
df2 <- subset(test\_data2, test\_data2$default != test\_data2$predict\_default2)  
  
test\_error\_rate2 = length(df2$default)/length(test\_data2$default)  
test\_error\_rate2

**## [1] 0.028**

training\_data\_index3 <- sample(10000, 5000, replace = FALSE)  
  
train\_model3 <- glm(default ~ income + balance, data = data, family = binomial, subset = training\_data\_index3)  
  
predict\_prob3 <- predict(train\_model3, data[-training\_data\_index3,], type = "response")  
  
  
test\_data3 <- data[-training\_data\_index3,]  
  
test\_data3$predict\_prob3 <- predict\_prob3  
test\_data3$predict\_default3 <- "No"  
test\_data3$predict\_default3[test\_data3$predict\_prob3 > 0.5] <- "Yes"  
  
df3 <- subset(test\_data3, test\_data3$default != test\_data3$predict\_default3)  
  
test\_error\_rate3 = length(df3$default)/length(test\_data3$default)  
test\_error\_rate3

**## [1] 0.026**

training\_data\_index4 <- sample(10000, 5000, replace = FALSE)  
  
train\_model4 <- glm(default ~ income + balance, data = data, family = binomial, subset = training\_data\_index4)  
  
predict\_prob4 <- predict(train\_model4, data[-training\_data\_index4,], type = "response")  
  
test\_data4 <- data[-training\_data\_index4,]  
  
test\_data4$predict\_prob4 <- predict\_prob4  
test\_data4$predict\_default4 <- "No"  
test\_data4$predict\_default4[test\_data4$predict\_prob4 > 0.5] <- "Yes"  
  
df4 <- subset(test\_data4, test\_data4$default != test\_data4$predict\_default4)  
  
test\_error\_rate4 = length(df4$default)/length(test\_data4$default)  
test\_error\_rate4

**## [1] 0.0254**

**We see that in all three simulations, the test error rates are comparable.**

1. Now consider a logistic regression model that predicts the probability of default using income, balance, and a dummy variable for student. Estimate the test error for this model using the validation set approach. Comment on whether or not including a dummy variable for student leads to a reduction in the test error rate.

training\_data\_index <- sample(10000, 5000, replace = FALSE)  
  
train\_model\_stu <- glm(default ~ income + balance + student, data = data, family = binomial, subset = training\_data\_index)  
  
predict\_prob <- predict(train\_model\_stu, data[-training\_data\_index,], type = "response")  
  
test\_data <- data[-training\_data\_index,]  
  
test\_data$predict\_prob <- predict\_prob  
test\_data$predict\_default <- "No"  
test\_data$predict\_default[test\_data$predict\_prob > 0.5] <- "Yes"  
  
df <- subset(test\_data, test\_data$default != test\_data$predict\_default)  
  
test\_error\_rate\_stu = length(df$default)/length(test\_data$default)  
test\_error\_rate\_stu

**## [1] 0.0262**

1. In Sections 5.3.2 and 5.3.3, we saw that the cv.glm() function can be used in order to compute the LOOCV test error estimate. Alternatively, one could compute those quantities using just the glm() and predict.glm() functions, and a for loop. You will now take this approach in order to compute the LOOCV error for a simple logistic regression model on the Weekly data set. Recall that in the context of classification problems, the LOOCV error is given in (5.4).
2. Fit a logistic regression model that predicts Direction using Lag1 and Lag2.

data <- Weekly  
  
model\_1 <- glm(Direction ~ Lag1 + Lag2, data = data, family = binomial)

1. Fit a logistic regression model that predicts Direction using Lag1 and Lag2 using all but the first observation.

model\_2 <- glm(Direction ~ Lag1 + Lag2, data = data[-1,], family = binomial)

1. Use the model from (b) to predict the direction of the first observation. You can do this by predicting that the first observation will go up if P(Direction=“Up”|Lag1, Lag2) > 0.5. Was this observation correctly classified?

predict\_prob <- predict(model\_2, data[1,], type = "response")  
print(data$Direction[[1]])

## [1] Down  
## Levels: Down Up

print(if(predict\_prob > 0.5) "Up" else "Down")

## [1] "Up"

**Here we see that the model predicts up while the actual outcome is Down**

1. Write a for loop from i = 1 to i = n, where n is the number of observations in the data set, that performs each of the following steps:
2. Fit a logistic regression model using all but the ith observation to predict Direction using Lag1 and Lag2.
3. Compute the posterior probability of the market moving up for the ith observation.
4. Use the posterior probability for the ith observation in order to predict whether or not the market moves up.
5. Determine whether or not an error was made in predicting the direction for the ith observation. If an error was made, then indicate this as a 1, and otherwise indicate it as a 0.

output\_list <- vector(mode = "numeric", length = length(data$Year))  
  
for (i in 1:length(data$Year)){  
 temp\_model <- glm(Direction ~ Lag1 + Lag2, data = data[-i,], family = binomial)  
 temp\_predict\_prob <- predict(temp\_model, data[i,], type = "response")  
 dir <- if(temp\_predict\_prob >= 0.5) "Up" else "Down"  
   
 output\_list[i] <- if(data$Direction[i] != dir) 1 else 0  
}

1. Take the average of the n numbers obtained in (d)iv in order to obtain the LOOCV estimate for the test error. Comment on the results.

CV\_error\_rate = sum(output\_list)/length(output\_list)  
CV\_error\_rate

**## [1] 0.4499541**

## 

## Final Project

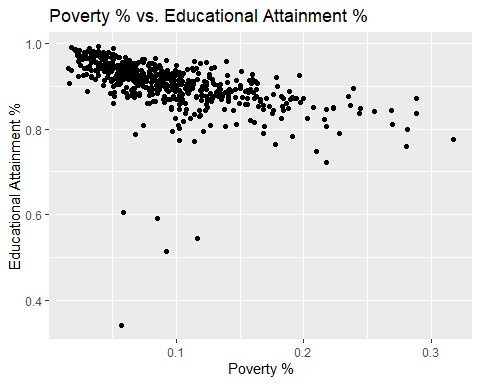
This week, Jackson and I investigated the assumptions for our linear model and potential remedies. From the textbook, we have 6 potential issues.

Potential Problems 3.3.3

1. Non-Linearity of the response-predictor relationships
2. Correlation of error terms
3. Non-constant variance of error terms
4. Outliers
5. High-leverage points
6. Collinearity

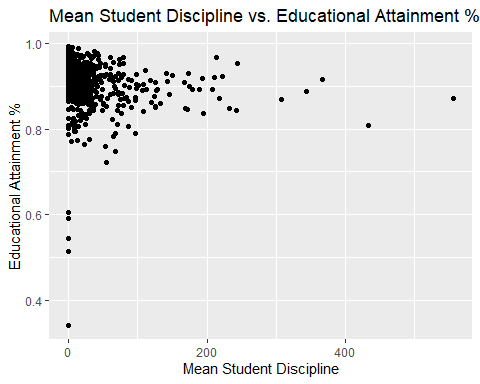
**Poverty % vs. Educational Attainment %**

ggplot(model\_data, aes(x=poverty\_percentage, y=HS\_PLUS\_percentage)) + geom\_point() +   
 ggtitle("Poverty % vs. Educational Attainment %") +  
 xlab("Poverty %") +   
 ylab("Educational Attainment %")



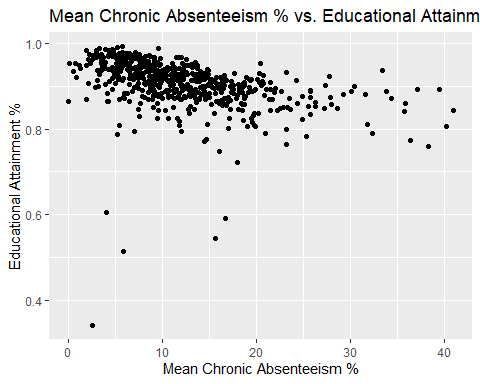
**Mean Student Discipline vs. Educational Attainment %**

ggplot(model\_data, aes(x=mean\_total\_students\_discipline, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Mean Student Discipline vs. Educational Attainment %") +  
 xlab("Mean Student Discipline") +  
 ylab("Educational Attainment %")



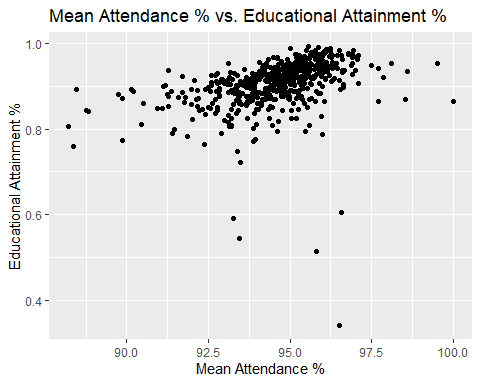
**Mean Chronic Absenteeism % vs. Educational Attainment %**

ggplot(model\_data, aes(x=mean\_chronic\_absenteeism, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Mean Chronic Absenteeism % vs. Educational Attainment %") +  
 xlab("Mean Chronic Absenteeism %") +  
 ylab("Educational Attainment %")



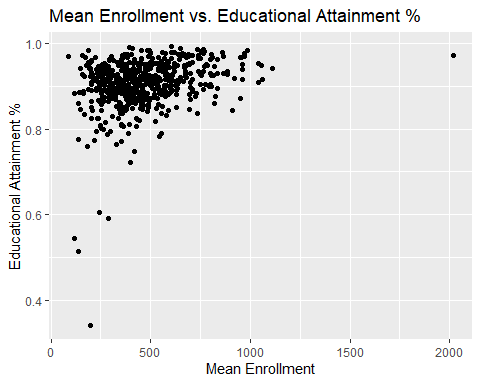
**Mean Attendance % vs. Educational Attainment %**

ggplot(model\_data, aes(x=mean\_attendance, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Mean Attendance % vs. Educational Attainment %") +  
 xlab("Mean Attendance %") +  
 ylab("Educational Attainment %")



**Mean Enrollment vs. Educational Attainment %**

ggplot(model\_data, aes(x=mean\_enrollment, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Mean Enrollment vs. Educational Attainment %") +  
 xlab("Mean Enrollment") +  
 ylab("Educational Attainment %")

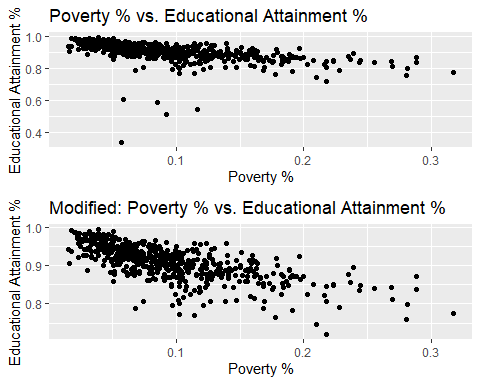


Now we remove outliers from the data due to assumed sampling errors and replot to show the difference in the scatter plots.

mod\_model\_data <- subset(model\_data, model\_data$HS\_PLUS\_percentage > 0.65)  
require(gridExtra)

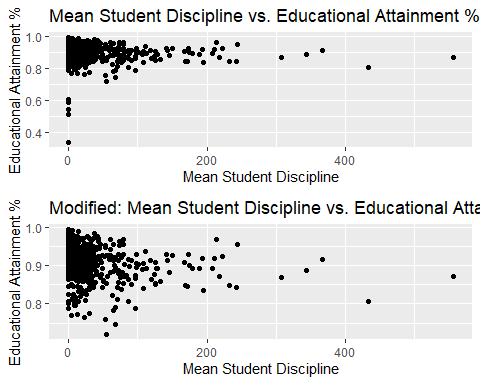
**Poverty % vs. Educational Attainment %**

plot1 <- ggplot(model\_data, aes(x=poverty\_percentage, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Poverty % vs. Educational Attainment %") +  
 xlab("Poverty %") +  
 ylab("Educational Attainment %")  
  
plot2 <- ggplot(mod\_model\_data, aes(x=poverty\_percentage, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Modified: Poverty % vs. Educational Attainment %") +  
 xlab("Poverty %") +  
 ylab("Educational Attainment %")  
  
grid.arrange(plot1, plot2)



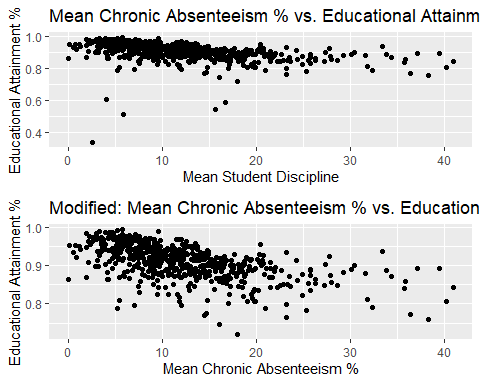
**Mean Student Discipline vs. Educational Attainment %**

plot1 <- ggplot(model\_data, aes(x= mean\_total\_students\_discipline, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Mean Student Discipline vs. Educational Attainment %") +  
 xlab("Mean Student Discipline") +  
 ylab("Educational Attainment %")  
  
plot2 <- ggplot(mod\_model\_data, aes(x= mean\_total\_students\_discipline, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Modified: Mean Student Discipline vs. Educational Attainment %") +  
 xlab("Mean Student Discipline") +  
 ylab("Educational Attainment %")  
  
grid.arrange(plot1, plot2)



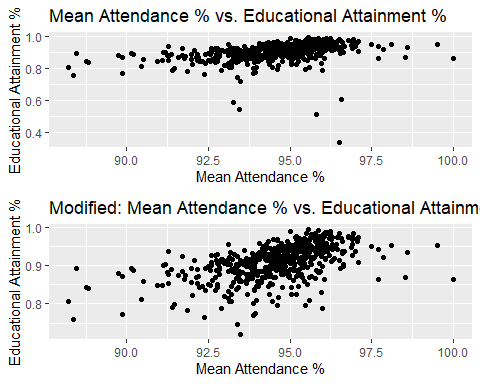
**Mean Chronic Absenteeism % vs. Educational Attainment %**

plot1 <- ggplot(model\_data, aes(x=mean\_chronic\_absenteeism, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Mean Chronic Absenteeism % vs. Educational Attainment %") +  
 xlab("Mean Student Discipline") +  
 ylab("Educational Attainment %")  
  
plot2 <- ggplot(mod\_model\_data, aes(x=mean\_chronic\_absenteeism, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Modified: Mean Chronic Absenteeism % vs. Educational Attainment %") +  
 xlab("Mean Chronic Absenteeism %") +  
 ylab("Educational Attainment %")  
  
grid.arrange(plot1, plot2)



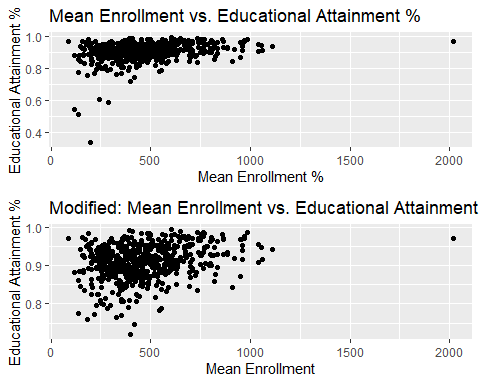
**Mean Attendance and Educational Attainment %**

plot1 <- ggplot(model\_data, aes(x=mean\_attendance, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Mean Attendance % vs. Educational Attainment %") +  
 xlab("Mean Attendance %") +  
 ylab("Educational Attainment %")  
  
plot2 <- ggplot(mod\_model\_data, aes(x=mean\_attendance, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Modified: Mean Attendance % vs. Educational Attainment %") +  
 xlab("Mean Attendance %") +  
 ylab("Educational Attainment %")  
  
grid.arrange(plot1, plot2)



**Mean Enrollment and Educational Attainment %**

plot1 <- ggplot(model\_data, aes(x = mean\_enrollment, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Mean Enrollment vs. Educational Attainment %") +  
 xlab("Mean Enrollment %") +  
 ylab("Educational Attainment %")  
  
plot2 <- ggplot(mod\_model\_data, aes(x= mean\_enrollment, y=HS\_PLUS\_percentage)) + geom\_point() +  
 ggtitle("Modified: Mean Enrollment vs. Educational Attainment %") +  
 xlab("Mean Enrollment") +  
 ylab("Educational Attainment %")  
  
grid.arrange(plot1, plot2)



# Simple Linear Regression Models  
  
POVERTY\_lm <- lm(HS\_PLUS\_percentage ~ poverty\_percentage, data = mod\_model\_data)  
DISCIPLINE\_lm <- lm(HS\_PLUS\_percentage ~ mean\_total\_students\_discipline, data = mod\_model\_data)  
CHRONIC\_lm <- lm(HS\_PLUS\_percentage ~ mean\_chronic\_absenteeism, data = mod\_model\_data)  
ATTENDANCE\_lm <- lm(HS\_PLUS\_percentage ~ mean\_attendance, data = mod\_model\_data)  
ENROLLMENT\_lm <- lm(HS\_PLUS\_percentage ~ mean\_enrollment, data = mod\_model\_data)

**Shapiro-Wilk Normality Test for Poverty %**

POVERTY\_sresid <- studres(POVERTY\_lm)  
shapiro.test(POVERTY\_sresid)

##   
## Shapiro-Wilk normality test  
##   
## data: POVERTY\_sresid  
## W = 0.95023, p-value = 3.341e-13

**Shapiro-Wilk Normality Test for Mean Student Discipline**

DISCIPLINE\_sresid <- studres(DISCIPLINE\_lm)  
shapiro.test(DISCIPLINE\_sresid)

##   
## Shapiro-Wilk normality test  
##   
## data: DISCIPLINE\_sresid  
## W = 0.95965, p-value = 1.215e-11

**Shapiro-Wilk Normality Test for Mean Chronic Absenteeism %**

CHRONIC\_sresid <- studres(CHRONIC\_lm)  
shapiro.test(CHRONIC\_sresid)

##   
## Shapiro-Wilk normality test  
##   
## data: CHRONIC\_sresid  
## W = 0.94531, p-value = 6.133e-14

**Shapiro-Wilk Normality Test for Mean Attendance %**

ATTENDANCE\_sresid <- studres(ATTENDANCE\_lm)  
shapiro.test(ATTENDANCE\_sresid)

##   
## Shapiro-Wilk normality test  
##   
## data: ATTENDANCE\_sresid  
## W = 0.94197, p-value = 2.056e-14

**Shapiro-Wilk Normality Test for Mean Enrollment**

ENROLLMENT\_sresid <- studres(ENROLLMENT\_lm)  
shapiro.test(ENROLLMENT\_sresid)

##   
## Shapiro-Wilk normality test  
##   
## data: ENROLLMENT\_sresid  
## W = 0.96928, p-value = 8.867e-10

**Score Test for Non-Constant Error Variance: Poverty %**

#heterodastic  
ncvTest(POVERTY\_lm)

## Non-constant Variance Score Test   
## Variance formula: ~ fitted.values   
## Chisquare = 26.53847, Df = 1, p = 2.5834e-07

**Score Test for Non-Constant Error Variance: Mean Student Discipline**

#homoscedastic  
ncvTest(DISCIPLINE\_lm)

## Non-constant Variance Score Test   
## Variance formula: ~ fitted.values   
## Chisquare = 0.08208011, Df = 1, p = 0.7745

**Score Test for Non-Constant Error Variance: Mean Chronic Absenteeism %**

#heteroscedastic  
ncvTest(CHRONIC\_lm)

## Non-constant Variance Score Test   
## Variance formula: ~ fitted.values   
## Chisquare = 6.226651, Df = 1, p = 0.012584

**Score Test for Non-Constant Error Variance: Mean Attendance %**

#homoscedastic  
ncvTest(ATTENDANCE\_lm)

## Non-constant Variance Score Test   
## Variance formula: ~ fitted.values   
## Chisquare = 0.00171754, Df = 1, p = 0.96694

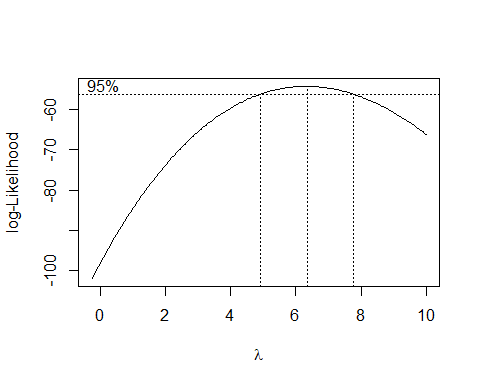
**Score Test for Non-Constant Error Variance: Mean Enrollment**

#heteroscedastic  
ncvTest(ENROLLMENT\_lm)

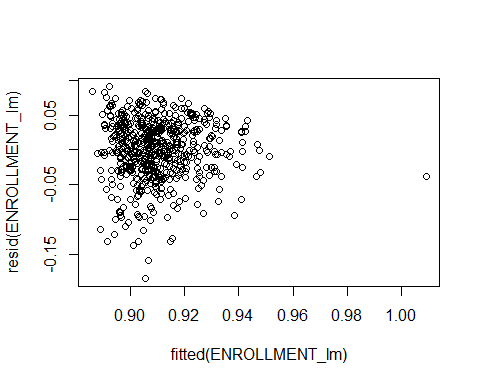
## Non-constant Variance Score Test   
## Variance formula: ~ fitted.values   
## Chisquare = 11.81214, Df = 1, p = 0.00058846

BOXCOX Transformations

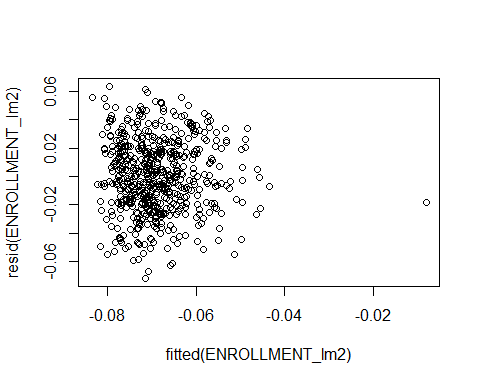
library(MASS)  
boxcox(ENROLLMENT\_lm, lambda = seq(-0.25, 10, by = 0.05), plotit = TRUE)



plot1 <- plot(fitted(ENROLLMENT\_lm), resid(ENROLLMENT\_lm))



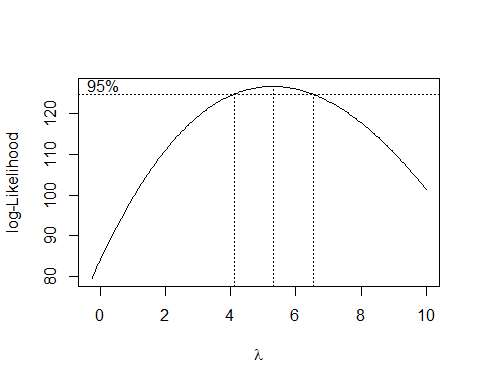
ENROLLMENT\_lm2 <- lm((((HS\_PLUS\_percentage ^ 6) - 1) / 6) ~ mean\_enrollment, data = mod\_model\_data)  
plot2 <- plot(fitted(ENROLLMENT\_lm2), resid(ENROLLMENT\_lm2))



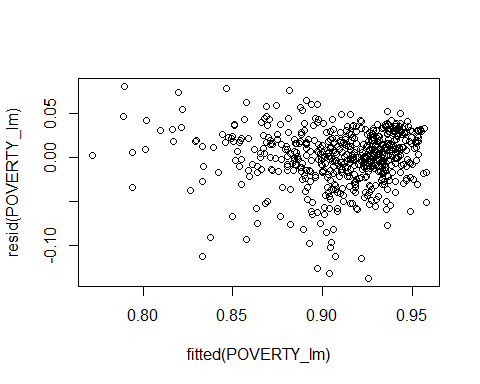
ENROLLMENT\_sresid2 <- studres(ENROLLMENT\_lm2)  
shapiro.test(ENROLLMENT\_sresid2)

##   
## Shapiro-Wilk normality test  
##   
## data: ENROLLMENT\_sresid2  
## W = 0.99584, p-value = 0.1196

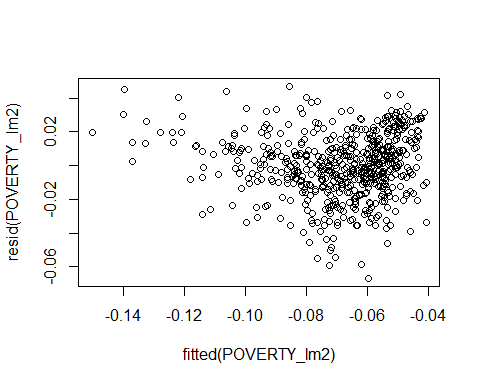
#Better P-value for normality, but still significantly Different from Normal  
POVERTY\_lm <- lm(HS\_PLUS\_percentage ~ poverty\_percentage, data = mod\_model\_data)  
boxcox(POVERTY\_lm, lambda = seq(-0.25, 10, by = 0.05), plotit = TRUE)



plot1 <- plot(fitted(POVERTY\_lm), resid(POVERTY\_lm))



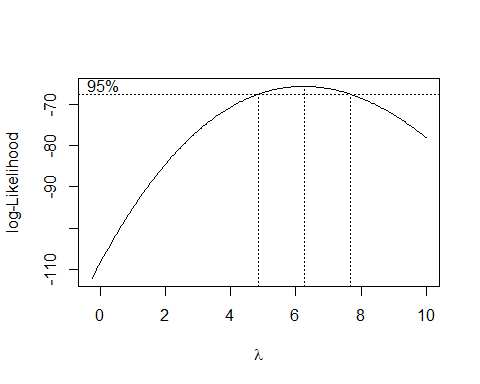
POVERTY\_lm2 <- lm((((HS\_PLUS\_percentage ^ 6) - 1) / 6) ~ poverty\_percentage, data = mod\_model\_data)  
plot2 <- plot(fitted(POVERTY\_lm2), resid(POVERTY\_lm2))



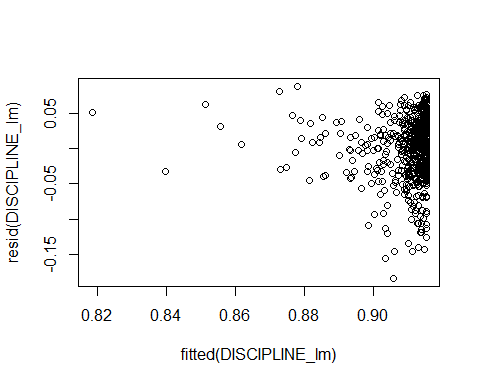
POVERTY\_sresid2 <- studres(POVERTY\_lm2)  
shapiro.test(POVERTY\_sresid2)

##   
## Shapiro-Wilk normality test  
##   
## data: POVERTY\_sresid2  
## W = 0.99205, p-value = 0.003028

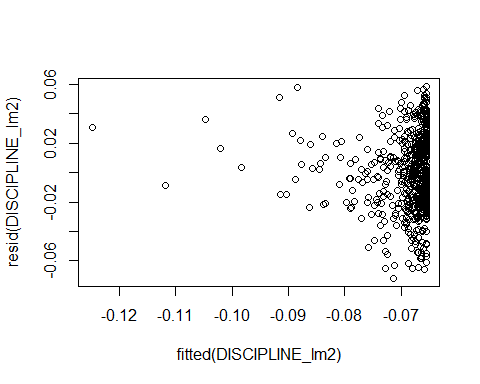
#Better P-value for normality, but still significantly Different from Normal  
DISCIPLINE\_lm <- lm(HS\_PLUS\_percentage ~ mean\_total\_students\_discipline, data = mod\_model\_data)  
boxcox(DISCIPLINE\_lm, lambda = seq(-0.25, 10, by = 0.05), plotit = TRUE)



plot1 <- plot(fitted(DISCIPLINE\_lm), resid(DISCIPLINE\_lm))



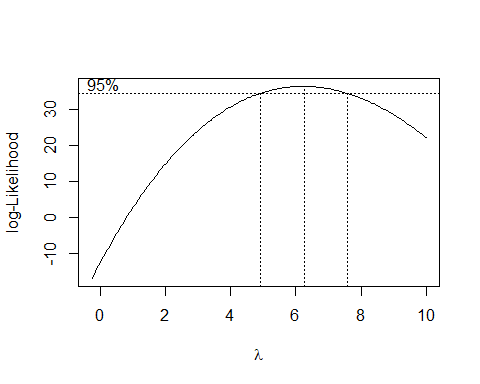
DISCIPLINE\_lm2 <- lm((((HS\_PLUS\_percentage ^ 6) - 1) / 6) ~ mean\_total\_students\_discipline, data = mod\_model\_data)  
plot2 <- plot(fitted(DISCIPLINE\_lm2), resid(DISCIPLINE\_lm2))



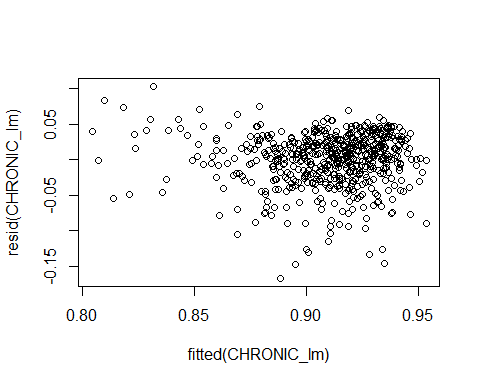
DISCIPLINE\_sresid2 <- studres(DISCIPLINE\_lm2)  
shapiro.test(DISCIPLINE\_sresid2)

##   
## Shapiro-Wilk normality test  
##   
## data: DISCIPLINE\_sresid2  
## W = 0.99339, p-value = 0.0108

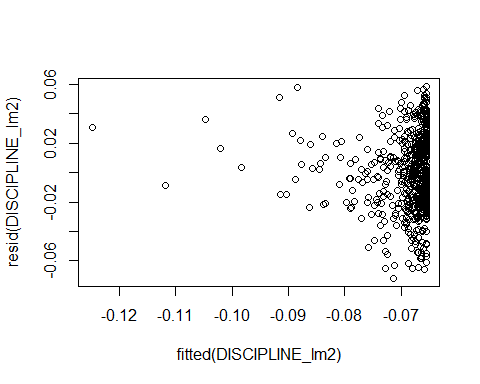
#Better P-value for normality, but still significantly Different from Normal  
CHRONIC\_lm <- lm(HS\_PLUS\_percentage ~ mean\_chronic\_absenteeism, data = mod\_model\_data)  
boxcox(CHRONIC\_lm, lambda = seq(-0.25, 10, by = 0.05), plotit = TRUE)



plot1 <- plot(fitted(CHRONIC\_lm), resid(CHRONIC\_lm))



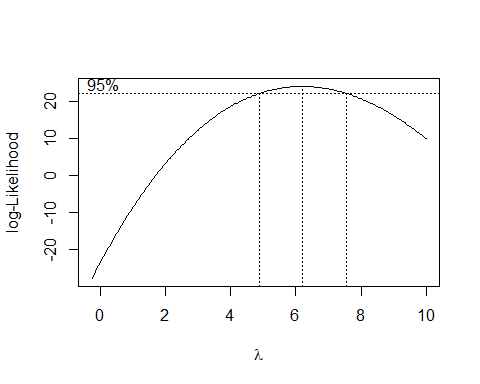
CHRONIC\_lm2 <- lm((((HS\_PLUS\_percentage ^ 6) - 1) / 6) ~ mean\_chronic\_absenteeism, data = mod\_model\_data)  
plot2 <- plot(fitted(DISCIPLINE\_lm2), resid(DISCIPLINE\_lm2))



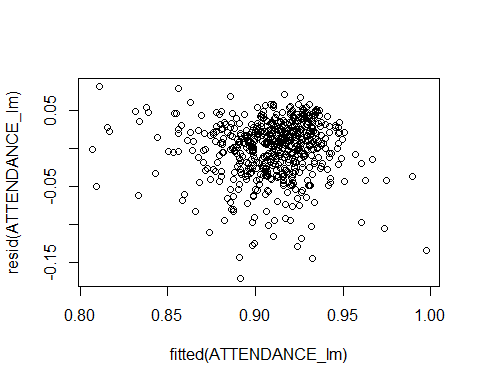
CHRONIC\_sresid2 <- studres(CHRONIC\_lm2)  
shapiro.test(DISCIPLINE\_sresid2)

##   
## Shapiro-Wilk normality test  
##   
## data: DISCIPLINE\_sresid2  
## W = 0.99339, p-value = 0.0108

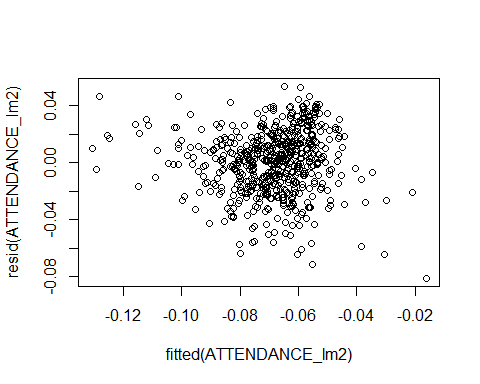
#Better P-value for normality, but still significantly Different from Normal  
ATTENDANCE\_lm <- lm(HS\_PLUS\_percentage ~ mean\_attendance, data = mod\_model\_data)  
boxcox(ATTENDANCE\_lm, lambda = seq(-0.25, 10, by = 0.05), plotit = TRUE)



plot1 <- plot(fitted(ATTENDANCE\_lm), resid(ATTENDANCE\_lm))



ATTENDANCE\_lm2 <- lm((((HS\_PLUS\_percentage ^ 6) - 1) / 6) ~ mean\_attendance, data = mod\_model\_data)  
plot2 <- plot(fitted(ATTENDANCE\_lm2), resid(ATTENDANCE\_lm2))



ATTENDANCE\_sresid2 <- studres(ATTENDANCE\_lm2)  
shapiro.test(ATTENDANCE\_sresid2)

##   
## Shapiro-Wilk normality test  
##   
## data: ATTENDANCE\_sresid2  
## W = 0.98915, p-value = 0.0002359