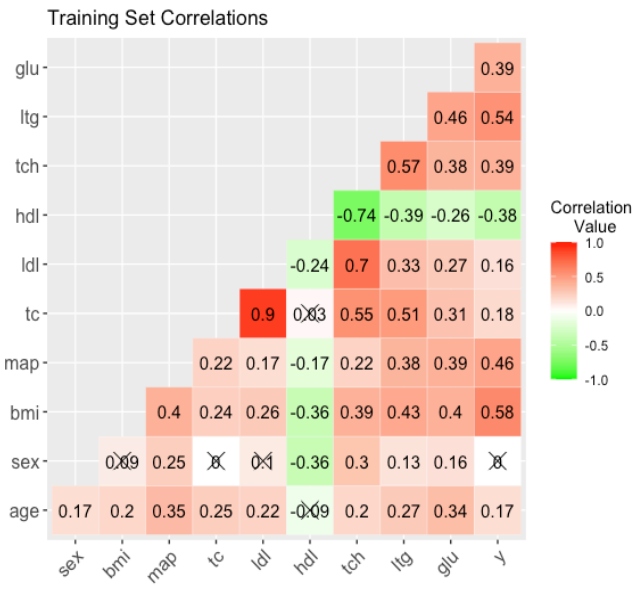
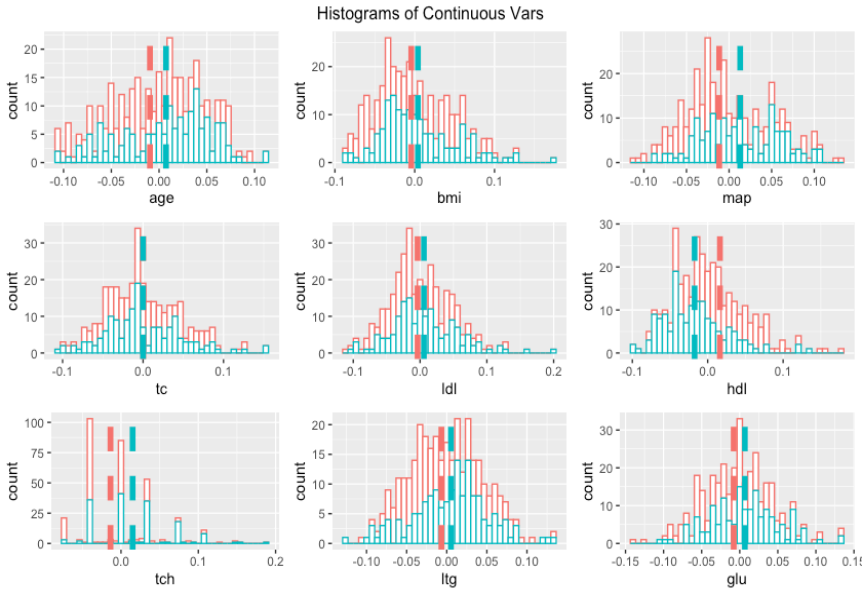
Leonid Rempel
ADEC7430.02 Big Data Econometrics
Individual Project 1
September 25, 2020

1. Introduction and Data Summary

This project employs the diabetes data in Efron et al. (2003) and examines the effects of ten baseline predictor variables[1] on a quantitative measure of disease progression one year after baseline[2]. The dataset contains measurements from 442 diabetes patients, and five different techniques are employed to study the progression of disease. This dataset was first broken down into a training set, containing a random sample of 75% of the data, and a testing set, consisting of the remaining 25% of the data.

The first step in the analysis is to determine any patterns inherent within the data itself. The left graph below summarizes the correlations among the variables and response used in the training set. The right graph summarizes the distribution of the continuous predictor variables[3] by sex.





From the correlation graph, we can see that the response variable has a significant correlation with most variables with the exception of the dummy variable sex, which is itself correlated significantly to map, hdl, tch, ltg, and glu. Another point of interest is the negative correlations that hdl has with the response and other predictor and the highly correlated nature of tc and ldl. We should expect tradeoffs with regards to these variables in the models. Lastly, the histograms tell us that, with the possible exception of tch, all of our predictor variables for both sexes follow a roughly normal distribution. Only map, hdl, and tch show some significant sex related differences, but otherwise we will have some degree efficiency in our linear predictor coefficients due to the normally distributed nature of the predictor variables.

---

[1] age, sex, body mass index (bmi), average blood pressure (map), and six blood serum measurements (tc, ldl, hdl, tch, ltg, glu)
[2] Rereferred to as 'y' throughout the study.
[3] The predictor variable, 'y' is excluded from this graph

2. Model Summary and Analysis

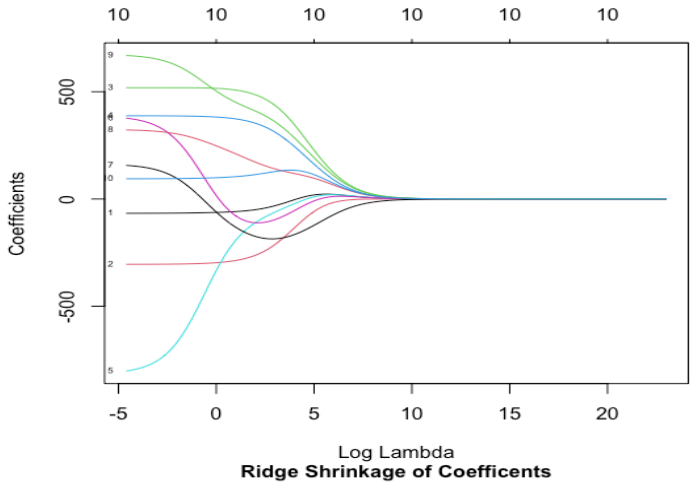
Model	Predictors Used	Test MSE	Std. Error of MSE
Full OLS	All ten	3111.265	361.091
BIC Best Subsets	sex, bmi, map, tc, tch, ltg	3095.483	369.753
CV Best Subsets	sex, bmi, map, tc, tch, ltg	3095.483	369.753
Ridge	All ten	3070.94	350.557
Lasso	sex, bmi, map, hdl, ltg, glu	2920.051	346.229

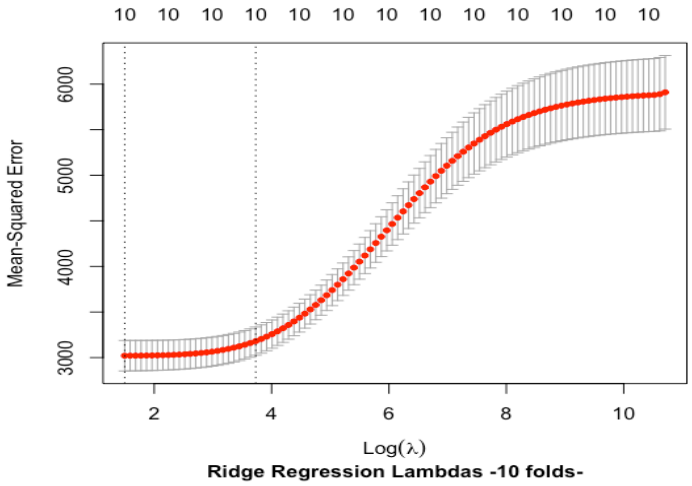
The table above summarizes the predictors used, the test data mean prediction error, and the standard errors of the mean prediction errors of each of the five models employed. The coefficient estimates produced for the predictors in each model can be found in appendix A, along with any additional model information such as the predictor significance and the diagnostics of the model itself.

The first model analyzed, which is an ordinary least squares regression that uses all of the predictors, has the poorest test MSE performance. This is to be expected from our initial data analysis of the correlations between predictors where we noted that, at the very minimum, a trade off needs to be made between using tc or ldl, if not also with tch and hdl. Surprisingly, the standard error of the mean prediction error ends up being better than both of the best subsets models, but this may be due to the luck of the specific random train test split.

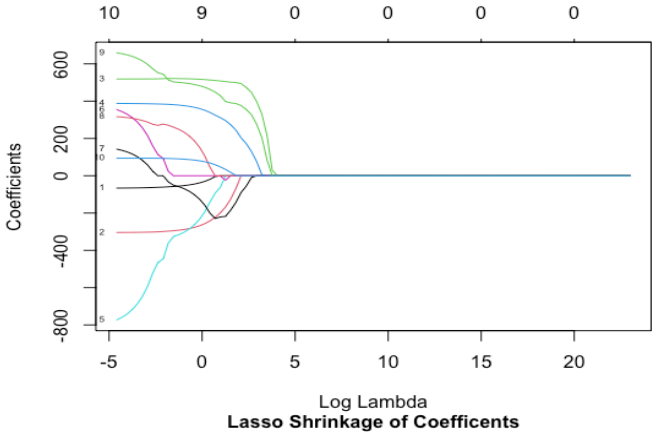
The next two models analyzed are both created using the best subset selection method. The first employs the BIC criteria to select the best subset model while and the latter used a ten-fold cross-validation for model selection. Both the BIC criterion and cross-validation method end up using the exact same predictors and therefore produced the same test MSE and the standard error of MSE. These models, on their own, suggest that it is better to use tc over ldl and tch over hdl for predicting the response.

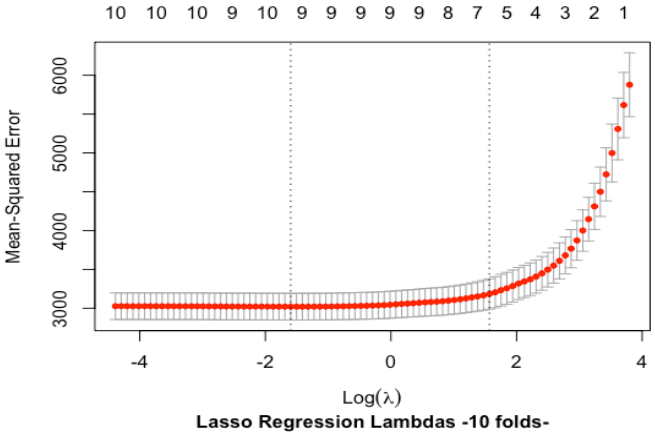
The ridge regression model preforms better than the preceding linear regression models. The increase in bias from using a lambda of around 42 to shrink the coefficients, as displayed in the graph on the next page, is less than the decrease in variance of the model when the random test dataset is used. The choice of lambda ensures that all of the ten available predictor variables are used, as none are shrunk close enough to be considered zero, which would be the case if a larger value of lambda was used. This on its own suggests that there is a specific form that a linear model has to take in order to be effective in predicting the progression of the disease.





The lasso model has the best performance out of the all the models employed, with the lowest MSE and standard error of MSE. Unlike the ridge, the lasso is capable of shrinking coefficients all the way to zero, and thus it results in a six feature model which makes different tradeoffs than the best subsets models. The lasso model, whose shrinkage of features is shown in the graph below, with lambda around 4.8, chooses hdl over tch and uses glu, unlike the other models. This goes against the grain of what the previous models have suggested thus far, but the reason why this unique six feature selection works is because it results in the least amount of overfitting to training data - which the other models suffer too much from in comparison.





3. Conclusion

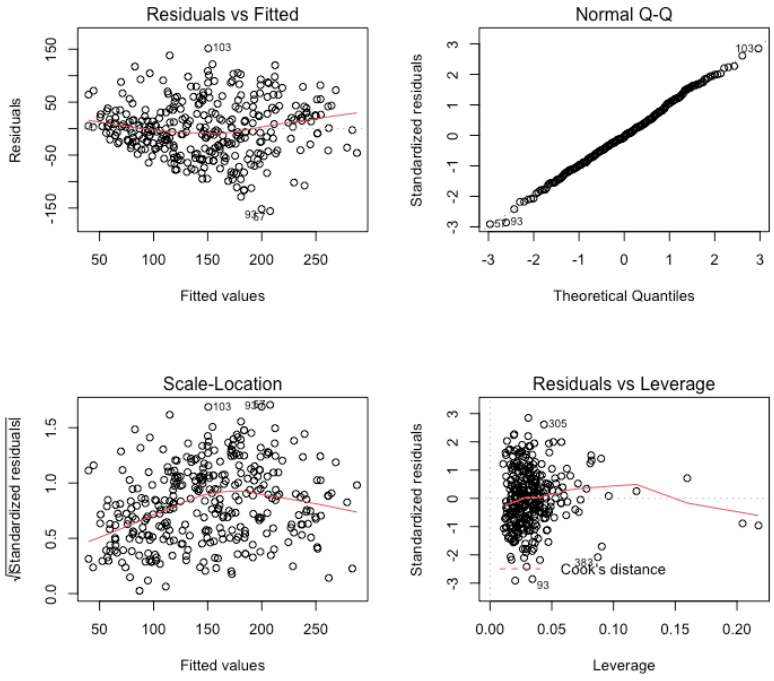
The study of the dataset gives us a good idea of which predictors, out of the ten provided, may be important in determining the progression of diabetes in the patients observed over the course of the year. Although sex is directly uncorrelated with the progression of disease, when used in

tandem with other variables, it becomes a useful predictor of diabetes development in patients. All the models agree that bmi, map, and ltg are also important predictors of the disease. Further study is necessary towards demining whether hdl really is more effective for predicting the response than tc or tch. As it stands now, the regularization models, the ridge and lasso, suggest there is a specific structure that a linear model has to adhere to in order to be useful. This makes sense since diabetes develops in a wide variety of patients and thus, most datasets will have a significant degree of variability from one study to the next. The next steps I would suggest is employing transformations within the linear model, since the plots of the residuals from the first three models have a sort of curve to it. It may be possible that at some point the effect from a factor, say bmi, has an increasing – not constant – effect on the progression of diabetes, as may be the case in obese patients. For the future, I would also suggest trying a random forest model. It would be excellent for the study of diabetes, especially if node purity can be achieved so that it can be determined which factors definitely do not contribute towards the progression of the disease. Such a technique has been used for other medical studies and it would perhaps find great effect here as well.

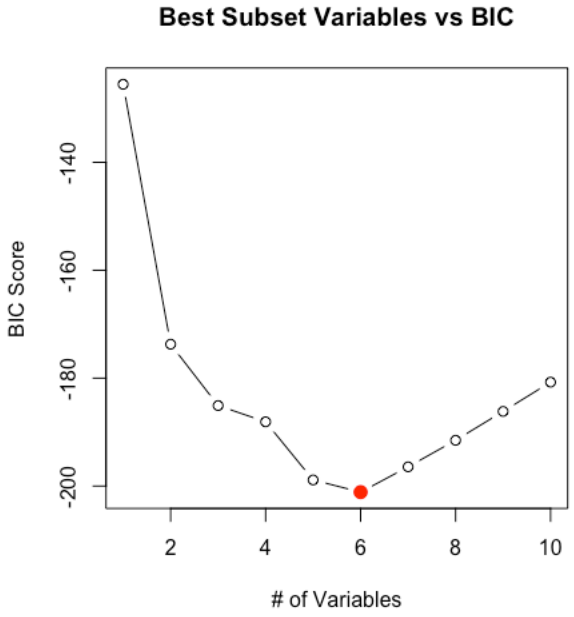
Appendix A: Model Details

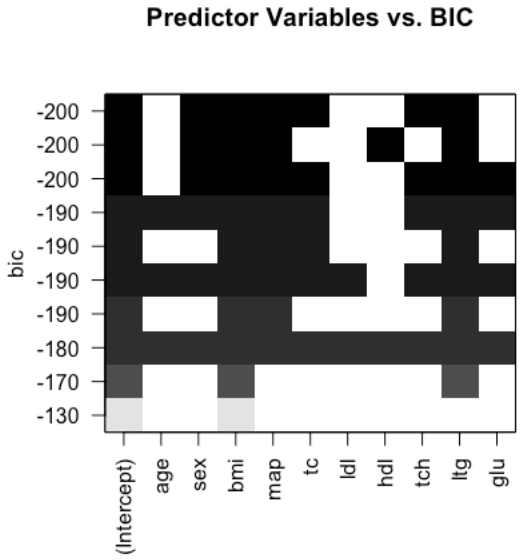
Least Squares Regression

```
Call:
lm(formula = y ~ ., data = data.train)

Residuals:
     Min       1Q   Median       3Q      Max
-155.726  -36.065   -2.758   35.039  151.509

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  149.920      2.976  50.382  < 2e-16 ***
age          -66.758     68.946  -0.968  0.33364
sex         -304.651     69.847  -4.362 1.74e-05 ***
bmi          518.663     76.573   6.773 6.01e-11 ***
map          388.111     72.755   5.335 1.81e-07 ***
tc          -815.268    537.549  -1.517  0.13034
ldl          387.604    439.162   0.883  0.37811
hdl          162.903    269.117   0.605  0.54539
tch          323.832    186.803   1.734  0.08396 .
ltg          673.620    206.888   3.256  0.00125 **
glu           94.219     79.590   1.184  0.23737
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 54.05 on 321 degrees of freedom
Multiple R-squared:  0.5213,    Adjusted R-squared:  0.5064
F-statistic: 34.96 on 10 and 321 DF,  p-value: < 2.2e-16
```

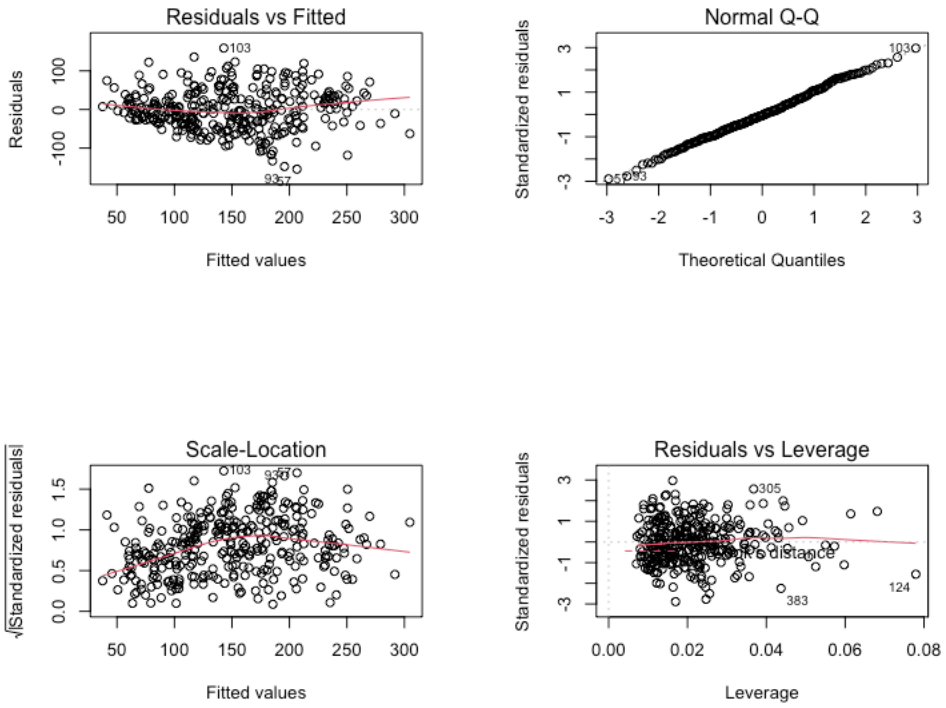


Best Subsets Regression using BIC

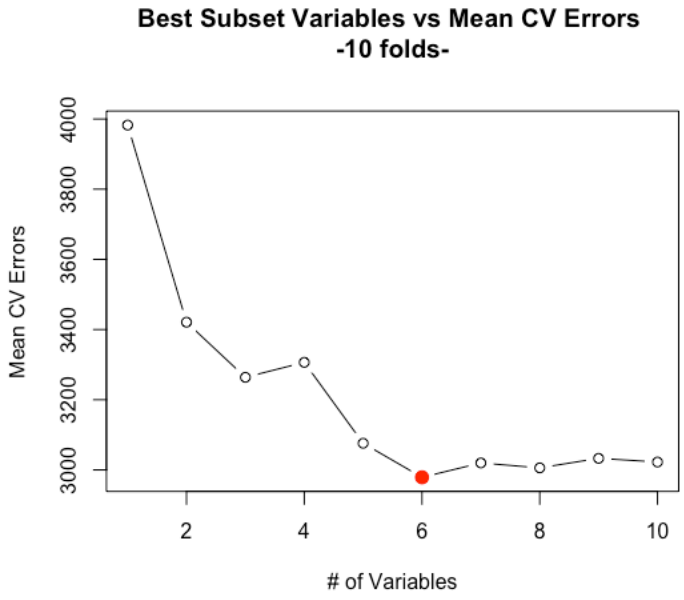




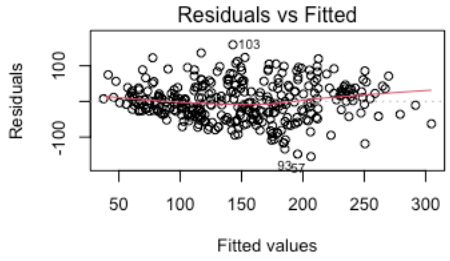

```
Call:
lm(formula = y ~ sex + bmi + map + tc + tch + ltg, data = data.train)

Residuals:
     Min       1Q   Median       3Q      Max
-154.549  -34.821   -2.544   35.859  159.067

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  150.117      2.962  50.683  < 2e-16 ***
sex         -306.042     69.041  -4.433 1.27e-05 ***
bmi          538.827     74.219   7.260 2.88e-12 ***
map          389.067     69.916   5.565 5.49e-08 ***
tc          -379.038     82.271  -4.607 5.87e-06 ***
tch          332.674     88.968   3.739 0.000218 ***
ltg          527.566     87.067   6.059 3.78e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 53.94 on 325 degrees of freedom
Multiple R-squared:  0.5172,    Adjusted R-squared:  0.5083
F-statistic: 58.03 on 6 and 325 DF,  p-value: < 2.2e-16
```

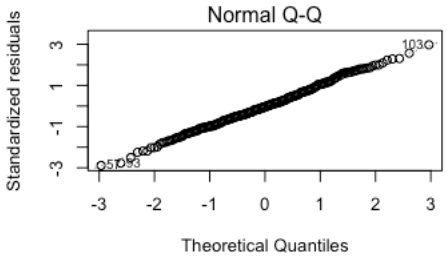


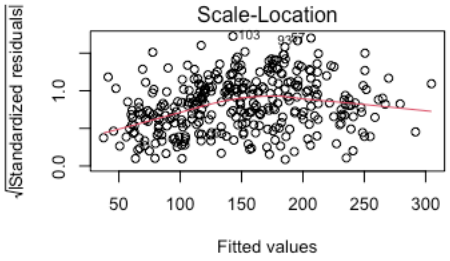
Best Subsets Regression using 10-fold Cross Validation:

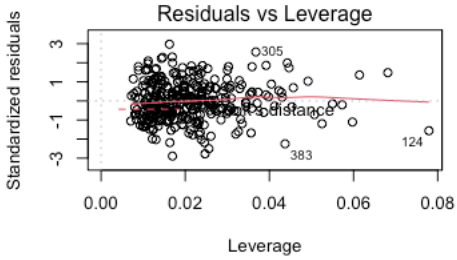


```
Call:
lm(formula = y ~ sex + bmi + map + tc + tch + ltg, data = data.train)

Residuals:
     Min       1Q   Median       3Q      Max
-154.549  -34.821   -2.544   35.859  159.067

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  150.117      2.962  50.683  < 2e-16 ***
sex         -306.042     69.041  -4.433 1.27e-05 ***
bmi          538.827     74.219   7.260 2.88e-12 ***
map          389.067     69.916   5.565 5.49e-08 ***
tc          -379.038     82.271  -4.607 5.87e-06 ***
tch          332.674     88.968   3.739 0.000218 ***
ltg          527.566     87.067   6.059 3.78e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 53.94 on 325 degrees of freedom
Multiple R-squared:  0.5172,    Adjusted R-squared:  0.5083
F-statistic: 58.03 on 6 and 325 DF,  p-value: < 2.2e-16
```









Ridge Regression:

(Intercept)	149.99068
age	-11.33162
sex	-156.91053
bmi	374.44939
map	264.89998
tc	-31.96990
ldl	-66.89724
hdl	-174.01202
tch	123.97204
ltg	307.68646
glu	134.48120

Ridge lambda used: 41.67209

Lasso Regression:

(Intercept)	149.95298
age	.
sex	-119.62208
bmi	501.56473
map	270.92614
tc	.
ldl	.
hdl	-180.29437
tch	.
ltg	390.55001
glu	16.58881

Lasso lambda used: 4.791278

Appendix B: R Code

```
cat("\014")
rm(list = ls())
gc()

# For replicability and to have the correct random draw from seeds
RNGversion('3.5.3')

### starter code provided to create test and train dataset

# Load the diabetes data
```

```
library(lars)
data(diabetes)
data.all <- data.frame(cbind(diabetes$x, y = diabetes$y))

# Partition the patients into two groups: training (75%) and test (25%)
n <- dim(data.all)[1]          # sample size = 442
set.seed(1306)                 # set random number generator seed to enable
                          # repeatability of results
test <- sample(n, round(n/4))   # randomly sample 25% test
data.train <- data.all[-test,]
data.test <- data.all[test,]
x <- model.matrix(y ~ ., data = data.all)[,-1] # define predictor matrix
                                    # excl intercept col of 1s
x.train <- x[-test,]           # define training predictor matrix
x.test <- x[test,]             # define test predictor matrix
y <- data.all$y                # define response variable
y.train <- y[-test]            # define training response variable
y.test <- y[test]              # define test response variable
n.train <- dim(data.train)[1]   # training sample size = 332
n.test <- dim(data.test)[1]     # test sample size = 110

#################################################################
# Looking at relationships within the dataset
pairs(data.train)

# Packages for data exploration
library(ggplot2)
library(ggcorrplot)
library(gridExtra)
library(ggthemes)
library(plyr)

# Creating a Correlation Graph
corr <- cor(data.train)
p.mat <- cor_pmat(data.train)

ggcorrplot(corr, type = "lower", outline.col = "white",
           p.mat = p.mat, ggtheme = 'theme_minimal',
           legend.title = 'Correlation\n     Value',
           colors = c('green', 'white', 'red'),
           title = 'Training Set Correlations', lab = T)

# Creating Histogram Graphs
mu <- ddply(data.train, 'sex', summarise, grp.mean=mean(age))
age.hist <- ggplot(data.train, aes(x=age, color = as.factor(sex))) +
             geom_histogram(fill = 'white', bins = 40) +
```

```
                geom_vline(data = mu, aes(xintercept=grp.mean, color=as.factor(sex)),
                           size = 2, linetype = 'dashed') +
                theme(legend.position = 'none')

mu <- ddply(data.train, 'sex', summarise, grp.mean=mean(bmi))
bmi.hist <- ggplot(data.train, aes(x=bmi, color = as.factor(sex))) +
                geom_histogram(fill = 'white', bins = 40) +
                geom_vline(data = mu, aes(xintercept=grp.mean, color=as.factor(sex)),
                           size = 2, linetype = 'dashed') +
                theme(legend.position = 'none')

mu <- ddply(data.train, 'sex', summarise, grp.mean=mean(map))
map.hist <- ggplot(data.train, aes(x=map, color = as.factor(sex))) +
                geom_histogram(fill = 'white', bins = 40) +
                geom_vline(data = mu, aes(xintercept=grp.mean, color=as.factor(sex)),
                           size = 2, linetype = 'dashed') +
                theme(legend.position = 'none')


mu <- ddply(data.train, 'sex', summarise, grp.mean=mean(tc))
tc.hist <- ggplot(data.train, aes(x=tc, color = as.factor(sex))) +
                geom_histogram(fill = 'white', bins = 40) +
                geom_vline(data = mu, aes(xintercept=grp.mean, color=as.factor(sex)),
                           size = 2, linetype = 'dashed') +
                theme(legend.position = 'none')

mu <- ddply(data.train, 'sex', summarise, grp.mean=mean(ldl))
ldl.hist <- ggplot(data.train, aes(x=ldl, color = as.factor(sex))) +
                geom_histogram(fill = 'white', bins = 40) +
                geom_vline(data = mu, aes(xintercept=grp.mean, color=as.factor(sex)),
                           size = 2, linetype = 'dashed') +
                theme(legend.position = 'none')

mu <- ddply(data.train, 'sex', summarise, grp.mean=mean(hdl))
hdl.hist <- ggplot(data.train, aes(x=hdl, color = as.factor(sex))) +
                geom_histogram(fill = 'white', bins = 40) +
                geom_vline(data = mu, aes(xintercept=grp.mean, color=as.factor(sex)),
                           size = 2, linetype = 'dashed') +
                theme(legend.position = 'none')

mu <- ddply(data.train, 'sex', summarise, grp.mean=mean(tch))
tch.hist <- ggplot(data.train, aes(x=tch, color = as.factor(sex))) +
                geom_histogram(fill = 'white', bins = 40) +
                geom_vline(data = mu, aes(xintercept=grp.mean, color=as.factor(sex)),
                           size = 2, linetype = 'dashed') +
                theme(legend.position = 'none')
```

```
mu <- ddply(data.train, 'sex', summarise, grp.mean=mean(ltg))
ltg.hist <- ggplot(data.train, aes(x=ltg, color = as.factor(sex))) +
              geom_histogram(fill = 'white', bins = 40) +
              geom_vline(data = mu, aes(xintercept=grp.mean, color=as.factor(sex)),
                         size = 2, linetype = 'dashed') +
              theme(legend.position = 'none')

mu <- ddply(data.train, 'sex', summarise, grp.mean=mean(glu))
glu.hist <- ggplot(data.train, aes(x=glu, color = as.factor(sex))) +
              geom_histogram(fill = 'white', bins = 40) +
              geom_vline(data = mu, aes(xintercept=grp.mean, color=as.factor(sex)),
                         size = 2, linetype = 'dashed') +
              theme(legend.position = 'none')

grid.arrange(age.hist, bmi.hist, map.hist, tc.hist, ldl.hist, hdl.hist,
          tch.hist, ltg.hist, glu.hist, ncol = 3,
          top = 'Histograms of Continuous Vars')

par(mfrow=c(1,1))

#################################################################
# Least Squares Regression Model

reg.OLS.all <- lm(y ~., data = data.train)
summary(reg.OLS.all)
OLS.pred <- predict.lm(reg.OLS.all, newdata = data.test, se.fit = T)
OLS.MSE <- mean((y.test-OLS.pred$fit)^2)

# The standard error of the prediction or the model is taken by dividing the standard
# deviation of the errors by the square root of the sample size
OLS.se <- sd((data.test$y - OLS.pred$fit)^2)/sqrt(n.test)

# Model Diagnostics
par(mfrow=c(2,2))
plot(reg.OLS.all)

#################################################################
# Best Subset Regression Model with min BIC

library(leaps)

reg.OLS.best <- regsubsets(y~., data = data.train, nvmax = 11)
sum.reg.OLS.best <- summary(reg.OLS.best)

par(mfrow=c(1,2))
```

```
plot(x = 1:10, y = sum.reg.OLS.best$bic, type = 'b', ylab = 'BIC Score', xlab = '# of Variables')
title(main = 'Best Subset Variables vs BIC')
points(6, min(sum.reg.OLS.best$bic), col = "red", cex = 2, pch = 20)
plot(reg.OLS.best, scale = "bic", main = "Predictor Variables vs. BIC")


min.bic.model <- coef(reg.OLS.best, which.min(sum.reg.OLS.best$bic))
min.bic.model # Printing out the coefficents of the best subsets model according to BIC
OLS.best.model <- lm(y~ sex + bmi + map + tc + tch + ltg, data.train)
summary(OLS.best.model)

test.mat <- model.matrix(y~., data = data.test)
OLS.best.pred <- test.mat[,names(min.bic.model)] %*% min.bic.model

OLS.best.MSE <- mean((y.test - OLS.best.pred)^2)
OLS.best.se <- sd((y.test - OLS.best.pred)^2)/sqrt(n.test)

# Model Diagnostics
par(mfrow=c(2,2))
plot(OLS.best.model)

#################################################################
# Best Subset with 10 fold CV

predict.regsubsets <- function(object, newdata, id, ...) {
  form <- as.formula(object$call [[2]])
  mat <- model.matrix(form, newdata)
  coefi <- coef(object, id=id)
  xvars <- names(coefi)
  return(mat[,xvars ]%*%coefi)
}

k <- 10
set.seed(1306)
folds <- sample(1:k, nrow(data.train), replace = TRUE)
cv.errors <- matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))

for (i in 1:k) {
  model <- regsubsets(y~., data = data.train[folds != i,], nvmax = 10)
  for (j in 1:10) {
    pred <- predict(model, data.train[folds == i,], id = j)
    cv.errors[i, j] = mean((data.train$y[folds == i] - pred)^2)
  }
}
mean.cv.errors = apply(cv.errors, 2, mean)
```

```
par(mfrow=c(1,1))
plot(mean.cv.errors, type = 'b', ylab = 'Mean CV Errors', xlab = '# of Variables')
title(main='Best Subset Variables vs Mean CV Errors \n-10 folds-')
points(6, min(mean.cv.errors), col = "red", cex = 2, pch = 20)

# We create the 6 variable model
reg.OLS.best.CV <- regsubsets(y~., data=data.train, nvmax = 11)
# First we look at the coefficients
coef(reg.OLS.best.CV, id = 6) # They end up being the same as the best subsets with BIC
OLS.best.CV.model <- lm(y~ sex + bmi + map + tc + tch + ltg, data.train)
summary(OLS.best.CV.model)

OLS.best.CV.pred <- predict.regsubsets(reg.OLS.best.CV, data.test, id = 6)
OLS.best.CV.MSE <- mean((y.test - OLS.best.CV.pred)^2)
OLS.best.CV.se <- sd((y.test - OLS.best.CV.pred)^2)/sqrt(n.test)

# Model Diagnostics
par(mfrow=c(2,2))
plot(OLS.best.CV.model)

#############################################################
# Ridge with 10 fold CV

library(glmnet)

par(mfrow = c(1,2))
grid <- 10^seq(10, -2, length = 100)

ridge.models <- glmnet(x.train, y.train, alpha = 0, lambda = grid, thresh = 1e-12)
plot(ridge.models, xvar = "lambda", label = TRUE)
title(sub = 'Ridge Shrinkage of Coefficents', font.sub = 2)

set.seed(1306)
cv.out.ridge <- cv.glmnet(x.train, y.train, alpha = 0)
plot(cv.out.ridge)
title(sub ='Ridge Regression Lambdas -10 folds-', font.sub = 2)
ridge.bestlamb <- cv.out.ridge$lambda.1se

# Print our the ridge best lambda within 1 standard error
ridge.bestlamb

# Determining the model coefficients
ridge.model <- glmnet(x.train, y.train, alpha = 0, lambda = ridge.bestlamb)
ridge.coef <- predict(ridge.model, s = ridge.bestlamb, type = "coefficients")

# Print out the coefficients
```

```
ridge.coef

# Determining the model measures of interests
ridge.pred <- predict(cv.out.ridge, s = ridge.bestlamb, newx = x.test)
ridge.MSE <- mean((y.test - ridge.pred)^2)
ridge.se <- sd((y.test-ridge.pred)^2)/sqrt(n.test)

# Print out the error measurements
ridge.MSE
ridge.se

############################################################
# Lasso with 10 fold CV

par(mfrow = c(1,2))

lasso.models <- glmnet(x.train, y.train, alpha = 1, lambda = grid, thresh = 1e-12)
plot(lasso.models, xvar = "lambda", label = TRUE)
title(sub = 'Lasso Shrinkage of Coefficents', font.sub = 2)

set.seed(1306)
cv.out.lasso <- cv.glmnet(x.train, y.train, alpha = 1)
plot(cv.out.lasso)
title(sub ='Lasso Regression Lambdas -10 folds-', font.sub = 2)
lasso.bestlamb <- cv.out.lasso$lambda.1se

# Print out the best lasso lambda within 1 standard error
lasso.bestlamb

# Determining the model coefficients
lasso.model <- glmnet(x.train, y.train, alpha = 1, lambda = lasso.bestlamb)
lasso.coef <- predict(lasso.model, s = lasso.bestlamb, type = "coefficients")

# Print out the coefficients
lasso.coef

# Determining the model measures of interests
lasso.pred <- predict(cv.out.lasso, s = lasso.bestlamb, newx = x.test)
lasso.MSE <- mean((y.test - lasso.pred)^2)
lasso.se <- sd((y.test-lasso.pred)^2)/sqrt(n.test)

# Print out the error measurements
lasso.MSE
lasso.se
```