STAT 983 **Homework 02** **Ramesh A. Kesawan**

Problem 1 (R exercise)

The following report contains the analysis of body fat dataset.

1. Introduction

Despite the fact that body fat causes adverse effect on human body, it is extremely necessary to maintain life and reproductive functions. If we formalize the body fat content, it is often refers to the percentage of fat content in one's body with respect to the body mass. With the advancement in the technology there are several ways to calculate the body fat content. But most accurate method of estimating body fat percentage was to measure that person's average density (total mass divided by total volume) and apply a formula to convert that to body fat percentage. The person average density is often calculated using the old and yet precise method called Underwater weighing, where it is measured by completely submerging a person in water and calculating the volume of the displaced water from the weight of the displaced water. Once you have average body density(ρ) then one can use two formulas to calculate the body fat content. The formulas are,

Brozek formula: $\mathrm{BF} = (4.57/\rho - 4.142) \times 100$

Siri formula is: $\mathrm{BF} = (4.95/\rho - 4.50) \times 100$.

With the aim of analyzing the body fat content a dataset given with 18 variables measured and calculated from 252 men. Among those 16 of them are body related measurements. such as height, weight, density and so on. The remaining two are body fat content given through Brozek and Siri equations. Our task is identify the necessary variables that would explain the body fat content. simply put, what is relationship between those variables and body fat content.

2. Explanatory Data Analysis

So, in the data we have 16 measurements taken from 252 men. As to comply with the saying of precaution is better than the cure, it is often advised to check the data before proceed with any particular analysis. So a concise explanatory analysis would help us to unveil some hidden pattern or information about the variables which will eventually help in building models easily.

Starting from the scatter plot matrix in figure 1, we have shown here the inter-relationship between few variables as it is hard to visualize all of them. We can clearly see some pattern here. As you can see almost all the variables are highly correlated. This could potentially leads to multi-colinearity. But lets discuss more about this on later sections. It is also confirmed from the correlation matrix given in the table 1. If you think clearly this correlation is nothing new, it is stating the obvious fact these body measurements are some what related to each others as these are obtained from an individual.

If see the box plots given in figure 2, you can see the distribution of these variables. For an example the height variable distributed 60- 80 inches, with an outlier of 30 inches. Likewise all of the variables have some outlying observations. We will see how to treat those variables in the next sections.

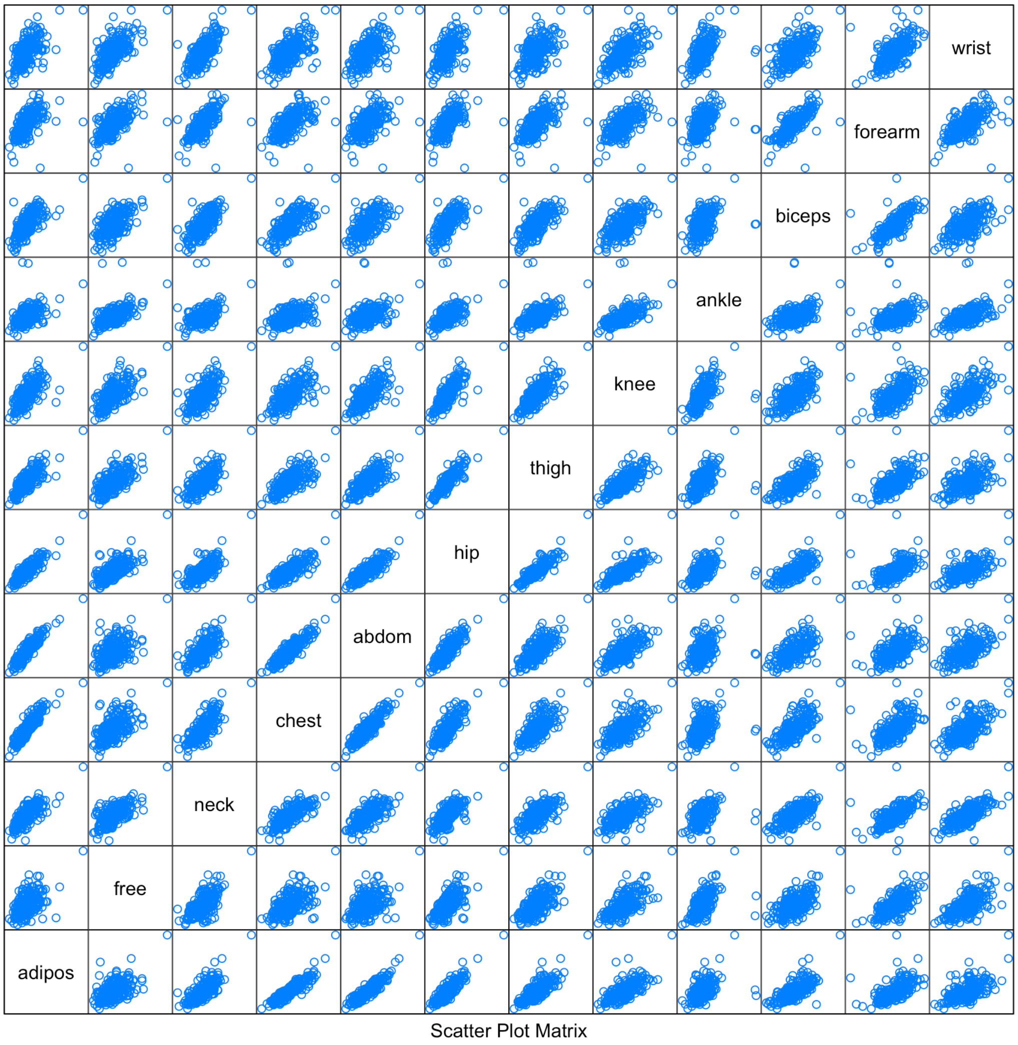

Figure 1: Scatter plot matrix

Table 1: Correlation

	adipos	**free**	**neck**	**chest**	**abdom**	**hip**	**thigh**	**knee**	**ankle**	**biceps**	**forearm**
adipos	1	0.54	0.77	0.91	0.92	0.89	0.81	0.71	0.48	0.74	0.59
free	0.54	1	0.67	0.58	0.48	0.69	0.67	0.69	0.56	0.65	0.6
neck	0.77	0.67	1	0.78	0.75	0.73	0.7	0.67	0.46	0.73	0.67
chest	0.91	0.58	0.78	1	0.92	0.83	0.73	0.71	0.46	0.73	0.62
abdom	0.92	0.48	0.75	0.92	1	0.87	0.77	0.73	0.42	0.68	0.53
hip	0.89	0.69	0.73	0.83	0.87	1	0.9	0.82	0.54	0.74	0.58
thigh	0.81	0.67	0.7	0.73	0.77	0.9	1	0.8	0.53	0.76	0.62
knee	0.71	0.69	0.67	0.71	0.73	0.82	0.8	1	0.58	0.69	0.6
ankle	0.48	0.56	0.46	0.46	0.42	0.54	0.53	0.58	1	0.47	0.42
biceps	0.74	0.65	0.73	0.73	0.68	0.74	0.76	0.69	0.47	1	0.75
forearm	0.59	0.6	0.67	0.62	0.53	0.58	0.62	0.6	0.42	0.75	1

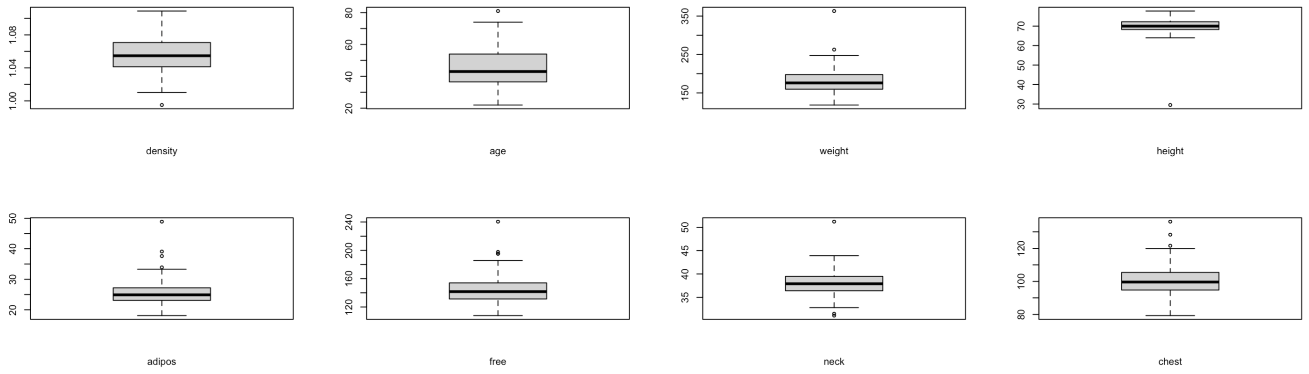

Figure 2: Box plots

Multi-collinearity: Multi-collinearity represents the presence of strong correlation in explanatory variables in a multiple regression setting. This often leads to incorrect results of regression analyses by imposing significant effects on one another. This will eventually makes the parameters unstable. There are several ways to identify the such relations starting from investigating the correlation among the explanatory variables. Once it is identified, it is often recommended to remove one or more explanatory variables as it does not cause any loss of information from a multiple linear regression perspective.

First method of diagnostic is to check the Variance Inflation Factor(VIF). It is the quotient of the variance in a model with multiple terms by the variance of a model with one term alone. There by it provides an index that measures how much the variance of an estimated regression coefficient is increased because of collinearity. The presence of the multi-collinearity is confirmed with the variance inflation factor values greater than 5 to 10. However, variance inflation factor helps to identify the presence of multi-collinearity, it cannot detect the exact explanatory variables that causes the multi-collinearity.

Table 2: VIF of variables

Variable	**VIF**	**Variable**	**VIF**
density	19.962	abdom	20.648
age	2.324	hip	15.565
weight	84.068	thigh	8.351
height	2.193	knee	4.678
adipos	16.91	ankle	1.887
free	36.637	biceps	3.962
neck	4.403	forearm	2.944
chest	11.187	wrist	3.427

The table 2 shows the collinearity of variables in the model. More than 8 variables VIF is greater than the 5. Which indicate the presence of extreme collinearity. As we said earlier the VIF could not be able to tell which variable causes this behavior, so we cannot come to decision of removing certain variable.

Condition Number: The eigenvalues (λ) of the standardized matrix explanatory of variables can also be used to diagnose multi-collinearity. Eigenvalues close to 0 indicate the presence of multi-collinearity, in which explanatory variables are highly inter-correlated and even small changes in the data lead to large changes in regression coefficient estimates. The square root of the ratio between the maximum and each eigenvalue ($\lambda_1, \lambda_2, \ldots, \lambda_\mathrm{k}$) is referred to as the condition index: $K_s = \sqrt{\frac{\lambda_{\max}}{\lambda_s}}(s = 1, 2, \ldots, k)$

The largest condition index is called the condition number. A condition number between 10 and 30 indicates the presence of multi-collinearity and when a value is larger than 30, the multi-collinearity is regarded as strong.

The table 3 indicate the presence shows the large five condition numbers. Which clearly indicate the presence of the multi-collinearity

Table 3: Condition Index

No.	**Condition Index**
1	109804.183
2	13660.76
3	636.741
4	375.531
5	306.606

3. Methods

Here we discuss models we used to describe the relationship between response variable and the explanatory variables.

Multiple Linear Regression: Linear regression is state of art method to describe a linear relationship between a scalar response and one or more predictor variables (also known as dependent and independent variables).

Formula of Multiple Linear Regression

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon$$

where, for $i = n$ observations, y_i = dependent variable, x_i = expanatory variables β_0 = y-intercept (constant term) β_p = slope coefficients for each explanatory variable and ϵ = the model's error term.

The multiple linear regression often been the victim of multi-collinearity as it deals with so many variables. As in our dataset we have seen this. Having so many variables inter-related to each others triggers the multi-collinearity in the model. So reduce the effect of the multi-collinearity we can remove the unnecessary variables. This can be achieve in several ways. Few possibilities are the model selection procedures such as step-wise regression and penalized regression techniques such as Ridge regression and Lasso regression.

The stepwise regression method: The way of selecting the appropriate regression model by systematically removing unnecessary models from group of candidate models. At every step of the procedure, the candidate variables are evaluated by some pre-specified criterion. Few considered approaches are

Forward-selection: This starts with no explanatory variables and then adds variables, one by one, based on which variable is the most statistically significant, until there are no remaining statistically significant variables.

backward-elimination: This starts with all possible explanatory variables and then discards the least statistically significant variables, one by one. Backward elimination is challenging if there is a large number of candidate variables and impossible if the number of candidate variables is larger than the number of observations.

Ridge Regression: Ridge regression is a technique which is similar to usual regression (least squares) except the coefficients are estimated by minimizing a slightly different quantity. The objective function which is minimized in-order to get the coefficients is,

$$\sum_{i=1}^{n}(y_i - \beta_0 - x_i^T\beta)^2 + \lambda||\beta||^2$$

Where the value λ value will be determined separately by cross validation. In least square regression the coefficients are estimated by minimizing the first quantity in the above equation. However, in the ridge regression, the coefficients are estimated by adding a shrinkage penalty to the usual least square estimates.

Ridge Regression is a technique for analyzing multiple regression data that suffer from multi-collinearity. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will be to give estimates that are more reliable. Therefore, it is known that the ridge penalty shrinks the coefficients of correlated predictors towards each other.

Lasso regression: The Technique is similar to ridge regression with an different penalty term - L_1 added to the model. Therefore, Lasso is a regularized regression method that is with the L_1 penalty. Now the objective function which is minimized in-order to get the coefficients is,

$$\sum_{i=1}^{n}(y_i - \beta_0 - x_i^T\beta)^2 + \lambda|\beta|$$

Where the value $\lambda's$ will be determined separately by cross validation. Because of the L_1 penalty added to the model the lasso tends to pick necessary varaibles and discard the others.

4. Results

The dataset is analysed using various techniques. Here we present the results of each of these analyses. We have evaluated each of models by calculating the training and testing errors from two disjoint data. The test set is build from the randomly selected 25 observations from the original dataset. At the same time the rest is kept it for the train set.

Despite the presence of multi-collinearity, we started building a multiple linear regression model. As this model will be regarded as a baseline model with all the variables. Only 8 out of 16 are significant at 5% significant level. The estimated coefficient values are given in the table 4. We also obtained the value of coefficient of determination (adjusted R^2). That is **0.987** and that implies the model's ability to explain the variation in the response values.

Further to measure the performance of the model, we calculated the training mean squared error. It is **0.7646**. and to see the performance of the model on unseen data, we calculated the mean squared error on the test set. It is **0.9794**. Even though there is slight increase in the test error than the training error. We cannot come to any conclusion at this point.

Then we went on to find the best model, by means of keeping necessary and getting rid of unnecessary variables. From that, the best model is selected based on the model which minimizes the training mean squared error. The figure 3 below shows that.

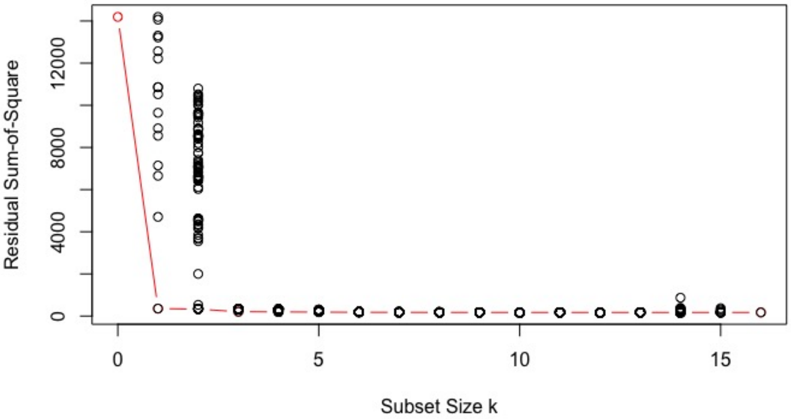

Figure 3: RSS of subsets of variables

The red line indicate the best model in each subset of models. For an example the best model from the set of **5** is with variables *density, weight,adipos,free and chest* and model is able to reduce the training MSE of **0.8304** and testing MSE **0.9939**. Next we considered the Automation procedure to select the best model which minimizes the AIC. It turns out significant variables in the above mentioned model are *density, weight, adipos , free , chest, thigh, forearm and wrist.* The training MSE of this model is **0.7728** where as the test MSE is **1.0016**.

Then we used Ridge regression to see how much it can reduce the training and testing errors. λ is chosen by cross validation that reduces mean squared error. It turns out it is 0.05. The figure 4 shows the behavior of regression coefficients with range of values of λ.

Using the λ obtained from the cross validation, the model is re-built. The model is able to perform with the training error of **0.7648** and testing error of **0.9424**. If you consider the table 4 you can see there isn't much shrinkage applied to the coefficients. So do a better model selection another approach should be considered.

So, we implemented lasso regression with the aim of selecting subset of variables that could explain the above mentioned relationship. The lasso procedure is successful and it founds that variable *neck* is not useful. figure 5 shows the shrinkage behaviour of these parameters. The lasso model performed with training MSE **0.7699** and testing MSE **0.8027**.

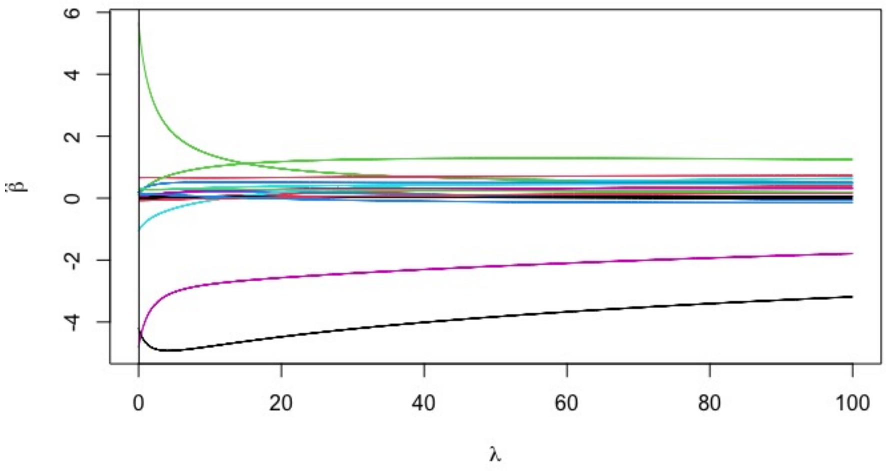

Figure 4: Ridge parameters

Table 4: Estimate coefficients under each model

Coefficients	**OLS**	**Ridge**	**Lasso**
(Intercept)	235.7761	237.5038	
density	-216.0709	-218.0395	-227.253
age	0.0042	0.0041	0.004
weight	0.192	0.1878	0.1775
height	0.007	0.008	0.0103
adipos	-0.2855	-0.2803	-0.2305
free	-0.2634	-0.2597	-0.2444
neck	0.0078	0.0086	0
chest	0.0774	0.0775	0.0685
abdom	0.0105	0.0119	0.0105
hip	0.0254	0.0272	0.0208
thigh	0.0614	0.0607	0.0504
knee	0.0084	0.01	0.013
ankle	-0.0069	-0.0066	-0.0059
biceps	-0.0239	-0.024	-0.0162
forearm	0.1455	0.1451	0.1309
wrist	0.1367	0.1368	0.124

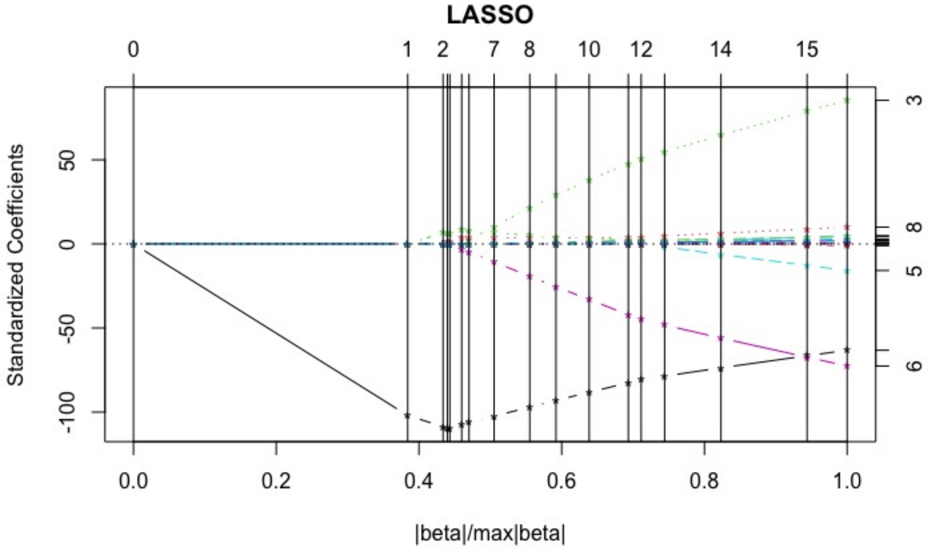

Figure 5: Lasso parameters

The table 5 shows the summary of these results. Non of these model exhibits a much reduction in both training and testing errors. But Lasso model performs much better than the others. But is is not advised use Lasso in the case of multi-collinearity.

Table 5: Summary of all models

	Full model	**Best of K=5**	**Stepwise**	**Ridge**	**Lasso**
Train Errors	**0.7640**	0.8304	0.7728	0.7648	0.7699
Test Errors	0.9794	0.9939	1.0016	0.9424	**0.8027**

The above procedure is sufficient to evaluate the model performance when we sufficiently large training and test set. But, with relatively small set, this results could be misleading. So, as general practice to evaluate the performance a cross-validation technique is preferred. One such approach is Monte Carlo Cross-Validation algorithm that repeats the above computation several times ($B = 100$). That is, for each loop $b = 1, ..., B$; we randomly select, say $n_1 = 25$ observations from the original data as the testing data, and use the remaining data as a training sample.

Table 6: Bootstrap test sample results.

	Full model	**Best of K=5**	**Stepwise**	**Ridge**	**Lasso**
Mean	0.7203	0.8482	0.7504	0.7098	0.7156
Variance	0.0291	0.0068	0.0248	0.05	0.028

Table 6 shows the results of the Monte Carlo Cross-Validation. You can see the mean of the test errors is relatively small in the all the analyses.

5. Findings

We had to analyse a dataset which contains the body fat percentage of 252 men and several body related variables. There are two variables that can be used as the response, but we restricted to use 'Broze' variable as it is the commonly used value to quantify the body fat measurements. We have seen existence of high multi-collinearity among the explanatory variables as they all related to body. Despite the multi- collinearity in the data, we started the model building procedure with regression model. The main reason behind this is to set up a baseline model, which then will be useful in evaluating the other or advanced models.

We then moved to update the regression model based on step-wise model selection procedures. In there we found a best model with 5 set of variables and a best model among all the other models using AIC. Then we build the penalized version of regression models, including ridge and lasso regression. As it is often advised to use these models as they always work well in case of multi-collinearity. We noticed that lasso model rejected the variable *'neck'* as a unnecessary variable in explaining the body fat of men.

As these MSE estimates of testing sets are based on single random sample of data. We argued that it is not viable to consider the superiority of one model over the other as this is based on single test set. So to get a clear picture we considered a several bootstrap random samples of the original dataset as the test set and build the model on that. The results are finally summarized the results in table 7. It is clearly noticeable that almost all the model exhibits same kind of behavior. R code is given at the end of the paper.

Problem 2

1. (Uniqueness of fitted values from the lasso. 5 points) For some $\lambda > 0$, suppose we have two lasso solutions $\hat{\beta}, \hat{\gamma}$ with the same optimal value in the lasso objective.

   1. Show that it must be the case that $\mathbf{X}\hat{\beta} = \mathbf{X}\hat{\gamma}$, meaning that the two solutions must yield the same predicted values. (Hint: If not, then use the strict convexity of the function $f(u) = \|y - u\|_2^2$ and convexity of the L_1 -norm to establish a contradiction.)

      Proof:

      The Lasso is the solution to the optimization problem with the objective function $f(\beta)$

$$f(\beta) = \underset{\beta}{\operatorname{argmin}} \frac{1}{2N}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

      Lets assume that two solutions do not yield the same same predicted values. i.e, $\mathbf{X}\hat{\beta} \neq \mathbf{X}\hat{\gamma}$. For any $0 < t < 1$, we have

$$\begin{aligned}
f(t\hat{\beta} + (1-t)\hat{\gamma}) &= \frac{1}{2N}\|y - X(t\hat{\beta} + (1-t)\hat{\gamma})\|_2^2 + \lambda\|(t\hat{\beta} + (1-t)\hat{\gamma})\|_1 \\
&= \frac{1}{2N}\|y - tX\hat{\beta} + (1-t)X\hat{\gamma}\|_2^2 + \lambda\|(t\hat{\beta} + (1-t)\hat{\gamma})\|_1 \\
&< \frac{1}{2N}\|y - tX\hat{\beta}\|_2^2 + \frac{1}{2N}\|(1-t)X\hat{\gamma}\|_2^2 + \lambda\|(t\hat{\beta} + (1-t)\hat{\gamma})\|_1 \qquad \text{Strict convexity of L2 norm} \\
&\leq \frac{1}{2N}\|y - tX\hat{\beta}\|_2^2 + \frac{1}{2N}\|(1-t)X\hat{\gamma}\|_2^2 + \lambda\|(t\hat{\beta}\|_1 + \lambda\|(1-t)\hat{\gamma})\|_1 \qquad \text{convexity of L1}
\end{aligned}$$

$$= tf\left(\hat{\beta}\right) + (1-t)f\left(\hat{\gamma}\right)$$

That is $t\hat{\beta} + (1-t)\hat{\gamma}$ obtains a lower value, which is a contradiction. Therefore, it must be the case that $\mathbf{X}\hat{\beta} = \mathbf{X}\hat{\gamma}$

2. If $\lambda > 0$, show that we must have $\|\hat{\beta}\|_1 = \|\hat{\gamma}\|_1$.

   Solution:

   By above part, we have proved that any two solutions must yield same predicted values and therefore it must have mean squared error. But the solutions also should get same value in $f(\beta)$ of above and if $\lambda > 0$, then they must have the same ℓ_1 norm. Therefore, $\|\hat{\beta}\|_1 = \|\hat{\gamma}\|_1$

Rcode

```
setwd("˜/Documents/Class/STAT-983")
library(Hmisc)
library(lattice)
library(broom)
library(tidyverse)
library(leaps)
library(MASS)
library(lars)

fat <- read.table(file = "fat.csv", sep=",", header=TRUE)
n = dim(fat)[1]
n1 = round(n/10)
set.seed(06)
flag = sort(sample(1:n, n1))

flag = c(1, 21, 22, 57, 70, 88, 91, 94, 121, 127, 149, 151, 159, 162,
         164, 177, 179, 194, 206, 214, 215, 221, 240, 241, 243);
fat1train = fat[-flag,]
fat1test = fat[flag,]

#Multi-collinearity
splom(fat1train[,7:18], pscales = 0)
summary(fat1train[,-(1:2)])
par(mfrow = c(4,4))
for (i in 3:18) boxplot(fat1train[,i], xlab=names(fat1train)[i])
dev.off()

#correlation
cor1=round(cor(fat1train[,7:18]),2)
write.csv(cor1, file = "corr.csv")

# Fit a linear regression on the training data
model0 <- lm(brozek ˜ .-siri, data = fat1train);
```

```
summary(model0);
X <- model.matrix(model0);
eign1 <- eigen( t(X) %*% X);
round(eign1$val,2)

#Condition Index
CI<-sort(sqrt( eign1$val[1] / eign1$val),decreasing = T)[1:5]
write.csv(round(CI,3), file = "CI.csv")

#VIF
VIF1 <- NULL;
X <- model.matrix(model0);
for (i in 2:17) VIF1 <- cbind(VIF1, 1/(1-summary(lm(X[,i] ~ X[,-i]))$r.squared));
colnames(VIF1) <- colnames(fat1train)[3:18]
write.csv(round(VIF1,3), file = "VIF1.csv")


### Models

#(i) Full model; Linear regression
ytrue=fat1test$brozek
model1 <- lm(brozek ~ .-siri, data = fat1train);
MSE1mod1 <-   mean((resid(model1) )^2)
kj=tidy((model1)) %>% data.frame()
kj[-1,2]
paste(colnames(train)[1], "~",paste(colnames(train)[-1], collapse = kj[-1,2]),sep = "")
pred1a <- predict(model1, fat1test)
MSE2mod1 <-   mean((pred1a - ytrue)^2)

#(ii) Linear regression with the best subset model
leaps <- regsubsets(brozek ~ .-siri, data= fat1train, nbest= 100, really.big= TRUE,nvmax = 16);
models <- summary(leaps)$which
models.size <- as.numeric(attr(models, "dimnames")[[1]])
models.rss <- summary(leaps)$rss

models.best.rss <- tapply(models.rss, models.size, min);
model0 <- lm( brozek ~ 1, data= fat1train);
models.best.rss <- c( sum(resid(model0)^2), models.best.rss);


op2 <- which(models.size == 5);
flag2 <- op2[which.min(models.rss[op2])];

mod2selectedmodel <- models[flag2,];
mod2Xname <- paste(names(mod2selectedmodel)[mod2selectedmodel][-1], collapse="+");
mod2form <- paste ("brozek ~", mod2Xname);
model2 <- lm( as.formula(mod2form), data= fat1train);
```

```
MSE1mod2 <- mean(resid(model2)^2);
pred2 <- predict(model2, fat1test);
MSE2mod2 <-   mean((pred2 - ytrue)^2)

#(iii) Best model stepwise
model1 <- lm(brozek ~ .-siri, data= fat1train);
model3  <- step(model1);

MSE1mod3 <- mean(resid(model3)^2);
pred3 <- predict(model3, fat1test);
MSE2mod3 <-   mean((pred3 - ytrue)^2);

#(iv) Ridge regression
ridge <- lm.ridge( brozek ~ .-siri, data= fat1train, lambda= seq(0,100,0.01));

lambdaopt <- which.min(ridge$GCV);
mod4.coef <- coef(ridge)[lambdaopt,]
round(mod4.coef, 4)

rig1coef <- ridge$coef[,lambdaopt];
rig1intercepts <- ridge$ym - sum(ridge$xm * (rig1coef / ridge$scales));

pred4 <- scale(fat1test[,-(1:2)], center = F, scale = ridge$scales)%*%
rig1coef + rig1intercepts;
MSE2mod4 <- mean( (pred4 - ytrue)^2);
pred8 <- scale(fat1train[,-(1:2)], center = F, scale = ridge$scales)%*%
rig1coef + rig1intercepts;
MSE2mod8 <- mean( (pred8 - ytrain)^2);
MSE2mod8
MSE2mod4

#(v) LASSO
lambdas <- seq(0.001,100,len=100)

lars <- lars(as.matrix(fat1train[,-(1:2)]), fat1train[,1], type= "lasso",
             trace= TRUE,normalize=T);
Cp1  <- summary(lars)$Cp
index1 <- which.min(Cp1)

coef(lars)[index1,]
lars$beta[index1,]

lasso.lambda <- lars$lambda[index1]
coef.lars1 <- predict(lars, s=lasso.lambda, type="coef", mode="lambda")
coef.lars1$coef
fit5 <- predict(lars, as.matrix(fat1train[,-(1:2)]), s=lasso.lambda, type="fit", mode="lambda");
```

```
yhat5 <- fit5$fit;
MSE1mod5 <- mean( (yhat5 - fat1train$brozek)^2);
MSE1mod5;

fit5b <- predict(lars, as.matrix(fat1test[,-(1:2)]), s=lasso.lambda, type="fit", mode="lambda");
yhat5b <- fit5b$fit;
MSE2mod5 <- mean( (yhat5b - fat1test$brozek)^2);
MSE2mod5;

## Comparing Models
data.frame(Train=c(MSE1mod1, MSE1mod2, MSE1mod3,MSE2mod8, MSE1mod5),
Test=c(MSE2mod1, MSE2mod2, MSE2mod3,MSE2mod4, MSE2mod5))


## Monte Carlo cross validation.
fat2<-fat[,-2]
test=fat2test<-fat1test[,-2]
train=fat2train<-fat1train[,-2]

bootModelError<-function(data,testSize=4,method='full',boots,subsetSize=5){
  require(leaps)
  require(lars)
  require(MASS)
  n=dim(data)[1]
  trainErrors<-rep(-99,boots)
  testErrors<-rep(-99,boots)
  for (i in 1:boots){
    testIndex=sort(sample(1:n,replace = F,size = testSize))
    train=data[-testIndex,]
    test=data[testIndex,]
    ytrue=test[,1]
    ytrain=train[,1]
    formulae<-as.formula(paste(colnames(train)[1], "~",
                               paste(colnames(train)[-1], collapse = "+"),
                               sep = ""))


    if(method=='full'){
      model=lm(formulae,data = train)
      trainMSE=mean((resid(model))^2)
      pred<-predict(model, test)
      testMSE<-mean((pred-ytrue)^2)

      trainErrors[i]<-trainMSE
      testErrors[i]<-testMSE
```

```
}

else if(method=='subset'){
  leaps <- regsubsets(formulae, nbest= 100,data= train, really.big= TRUE,
  nvmax =dim(data)[2]-1);
  models <- summary(leaps)$which
  models.size <- as.numeric(attr(models, "dimnames")[[1]]);
  models.rss <- summary(leaps)$rss

  op2 <- which(models.size == subsetSize)
  flag2 <- op2[which.min(models.rss[op2])]

  mod2selectedmodel <- models[flag2,];
  mod2Xname <- paste(names(mod2selectedmodel)[mod2selectedmodel][-1], collapse="+");
  mod2form <- paste (colnames(train)[1],"~", mod2Xname);
  model2 <- lm( as.formula(mod2form), data= train);

  # traning and testing errors
  trainMSE=mean((resid(model2))^2)
  pred<-predict(model2, test)
  testMSE<-mean((pred-ytrue)^2)

  trainErrors[i]<-trainMSE
  testErrors[i]<-testMSE
}
else if(method=='step'){
  model=lm(formulae,data = train)
  model3  <- step(model,trace = F)

  # traning and testing errors
  trainMSE=mean((resid(model3))^2)
  pred<-predict(model3, test)
  testMSE<-mean((pred-ytrue)^2)

  trainErrors[i]<-trainMSE
  testErrors[i]<-testMSE
}
else if(method=='ridge'){
  ridge <- lm.ridge( formulae, data =train, lambda= seq(0,100,0.01));
  #plot(ridge)

  lambdaopt <- which.min(ridge$GCV);

  rig1coef <- ridge$coef[,lambdaopt];

  rig1intercepts <- ridge$ym - sum(ridge$xm * (rig1coef / ridge$scales));
  pred <- scale(test[,-1], center = F, scale = ridge$scales)%*%
```

```
      rig1coef + rig1intercepts;
      fitt <- scale(train[,-1], center = F, scale = ridge$scales)%*%
      rig1coef + rig1intercepts

      trainMSE<-mean((fitt - ytrain)^2)
      testMSE <- mean((pred - ytrue)^2)

      trainErrors[i]<-trainMSE
      testErrors[i]<-testMSE
    }

    else if(method=='lasso'){
      lars <- lars(as.matrix(train[,-1]), train[,1], type= "lasso", trace= F)
      Cp1  <- summary(lars)$Cp
      index1 <- which.min(Cp1)

      fit5 <- predict(lars, as.matrix(train[,-1]), s=lasso.lambda, type="fit", mode="lambda")
      yhat5 <- fit5$fit
      trainMSE<- mean((yhat5 - ytrain)^2)

      fit5b <- predict(lars, as.matrix(test[,-1]), s=lasso.lambda, type="fit", mode="lambda");
      yhat5b <- fit5b$fit
      testMSE <- mean((yhat5b - ytrue)^2)

      trainErrors[i]<-trainMSE
      testErrors[i]<-testMSE
    }

  }

  return(list('method'=method,
  'meanTrEr'=mean(trainErrors),
  'meanTeEr'=mean(testErrors),
  'varTrEr'=var(trainErrors),
  'varTeEr'=var(testErrors),
  'trainErrors'=trainErrors,
  'testErrors'=testErrors)
  )
}
```