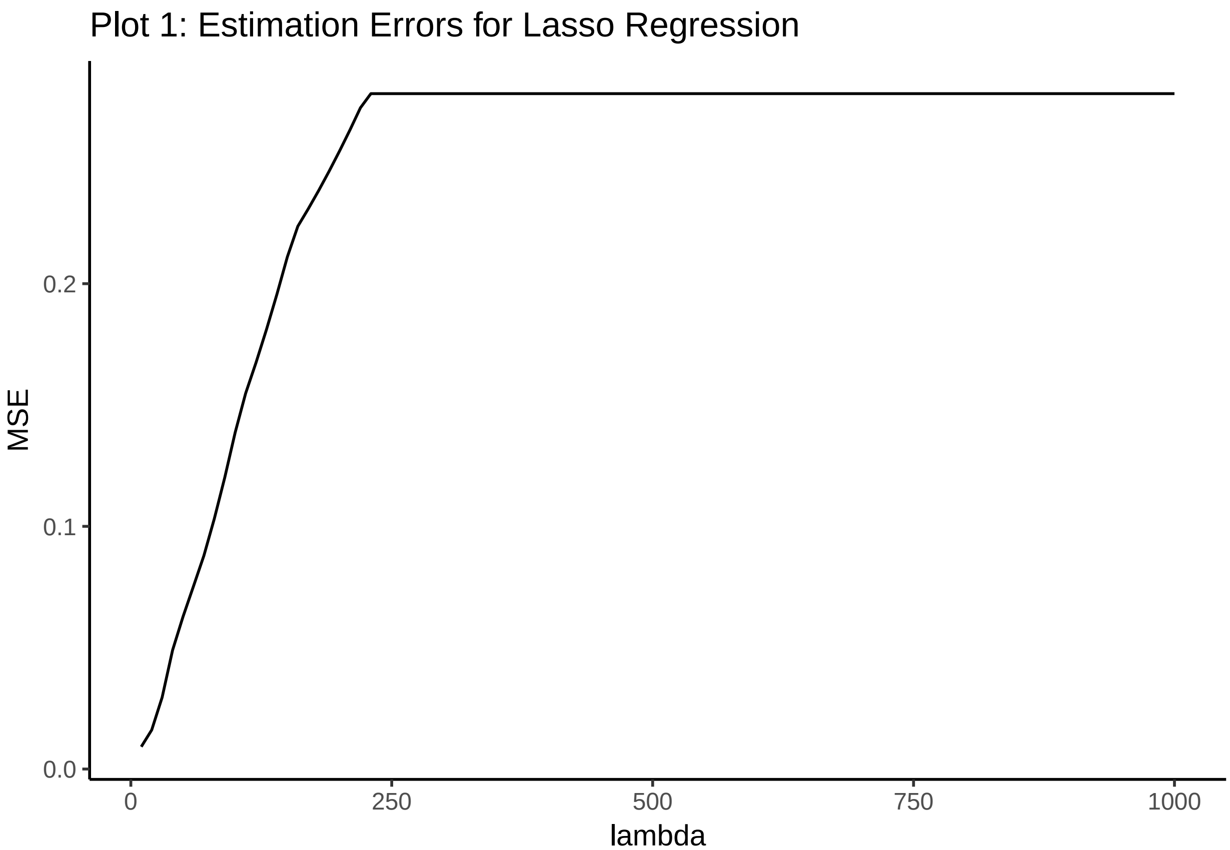
202A-HW7

Jiayu Wu

2017/11/26

Estimation Error for Lasso Regression

```
library(LMjw)
# training data
set.seed(10086)
n <- 50; p <- 200
X <- matrix(rnorm(n * p), nrow = n)
beta <- c(1:5,rep(0,195))
Y <- 1+X %*% beta+rnorm(n)
# model fitting and plotting
lambda_all <- (100:1)*10
beta_all <- myLasso(X,Y,lambda_all)
library(ggplot2)
mse <- apply(beta_all, 2, function(b){mean((b-c(1,beta))^2)})
p_mse <- ggplot(data.frame(lambda=lambda_all,MSE=mse),aes(x=lambda,y=MSE))+geom_line()
p_mse <- p_mse+theme_classic()+ggtitle("Plot 1: Estimation Errors for Lasso Regression")
p_mse
```



Regression Analysis

Analyze datasets “mtcars” with my linear regression package “LMjw” to study the response variable mpg (Miles/gallon).

```
# data
library(knitr)
kable(head(mtcars), align = 'c',caption = "Dataset mtcars with 32 observations")
```

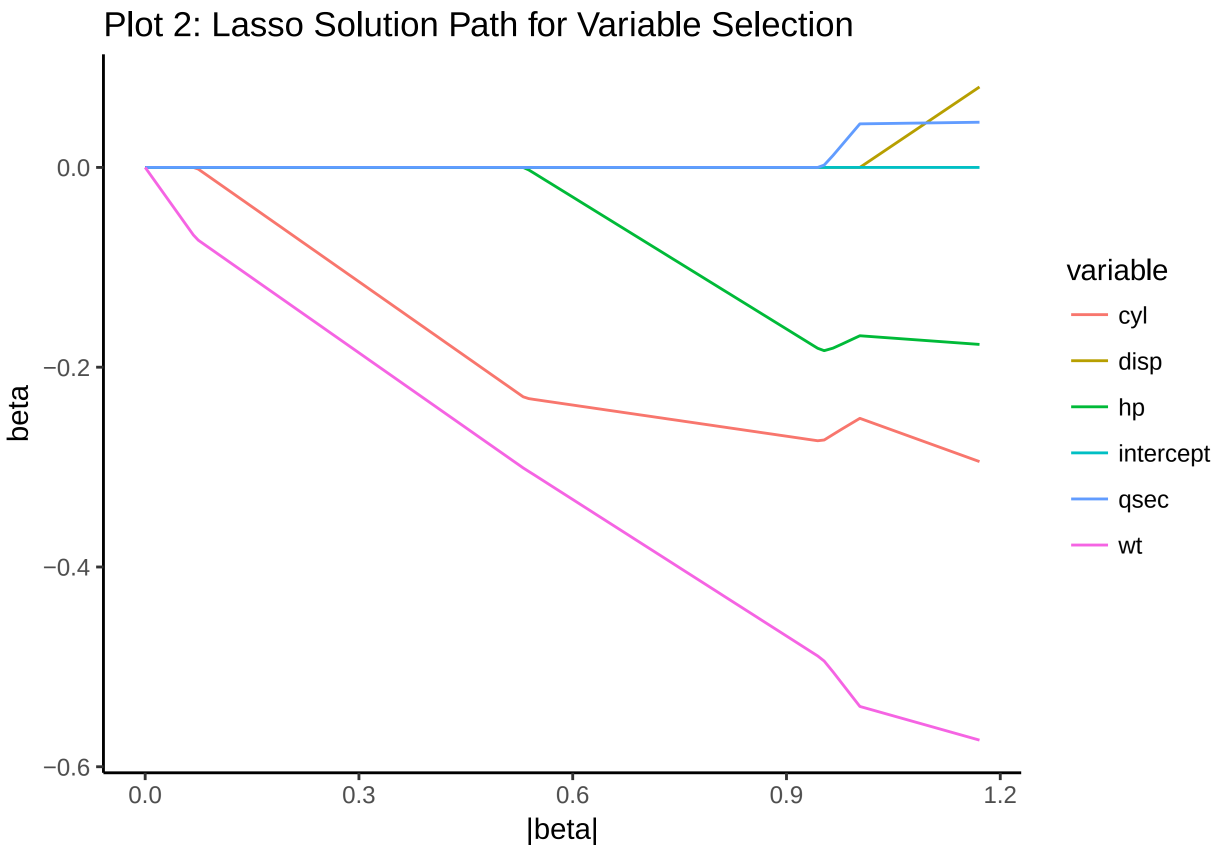
Table 1: Dataset mtcars with 32 observations

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

Lasso Regression for variable selection

Scale the data, regress mpg on all the other five continuous variables: number of cylinders (cyl), horse power (hp), weight (wt: 1000lbs), speed (qsec: 1/4 mile time) and displacement (disp: cu.in.) with Lasso regularization, and plot the solution path.

```
library(LMjw)
X <- as.matrix(mtcars[,c("cyl","hp","wt","qsec","disp")])
X <- scale(X)
Y <- as.matrix(mtcars[,c("mpg")])
Y <- scale(Y)
lambda_all <- (500:1)/5
beta_all <- myLasso(X,Y,lambda_all)
p <- ncol(X)
beta_sum <- t(matrix(rep(1, (p+1)), nrow = 1)%*%abs(beta_all))
d <- data.frame(x=rep(beta_sum,(p+1)),beta=as.numeric(t(beta_all)),
       variable=rep(c("intercept","cyl","hp","wt","qsec","disp"),each=length(lambda_all)))
mp <- ggplot(d,aes(x,beta,color=variable))+geom_line()+theme_classic()+xlab("|beta|")
mp <- mp+ggtitle("Plot 2: Lasso Solution Path for Variable Selection")
mp
```



According to the solution path, we can choose three variables that are firstly admitted: weight (wt: 1000lbs), horse power (hp) and number of cylinders (cyl). Thus, we get the following multiple linear model:

$$mpg = wt + hp + cyl + \epsilon$$

Least square estimation and Ridge shrinkage

With the above data, fit the model with ordinary least square powered by QR decomposition.

Sample half of the observations (16) from the dataset for testing, and plot training/testing error to find a reasonable λ_{goal} through cross-validation. Then fit the model with ridge regression penalized by λ_{goal}.

```
library(LMjw)
X <- as.matrix(mtcars[,c("hp","wt","cyl")])
Y <- as.matrix(mtcars[,c("mpg")])
# least square regression
ls <- myLM(X,Y)
# ridge regression
set.seed(1)
train <- sample(32,16)
lambda <- seq(0.1,6.1,0.1)
train_er <- apply(t(lambda),2,function(lambda){
       beta<-myRidge(X[train,],Y[train,],lambda)
       mean(((Y[train,]-cbind(rep(1,16),X[train,])%*%beta))^2)
       })
test_er<-apply(t(lambda),2,function(lambda){
       beta<-myRidge(X[train,],Y[train,],lambda)
       mean(((Y[-train,]-cbind(rep(1,16),X[-train,])%*%beta))^2)})
```

```r
lambda_goal <- lambda[which.min(test_er-train_er)]
library(ggplot2)
er <- data.frame(lambda=rep(lambda, 2),type=rep(c("testing","training"),
              each=length(lambda)),error=c(test_er,train_er))
p_er <- ggplot(er,aes(x=lambda,y=error,color=type))+geom_line()+geom_vline(xintercept = lambda_goal)
p_er <- p_er+annotate("text",x=2.9,y=7.5,label=paste0("lambda_goal","=",lambda_goal))
p_er <- p_er+theme_classic()+ggtitle("Plot 3: Training and Testing Errors for Ridge Regression")
p_er
```


```r
ridge <- myRidge(X,Y,lambda_goal)
t<-rbind(ls$beta_ls,t(ridge))
rownames(t)<-c("OLS","Ridge(lambda=2)")
colnames(t)<-c("intercept","hp","wt","cyl")
library(knitr)
kable(t, align = 'c',caption = "Regression Coefficients")
```

Table 2: Regression Coefficients

	intercept	hp	wt	cyl
OLS	38.75179	-0.0180381	-3.166973	-0.9416168
Ridge(lambda=2)	38.30987	-0.0196643	-2.762324	-1.0420453

According to Plot 3, we use $\lambda_{goal} = 2$ to penalize overfitting in ridge regression. The regression result is compared with least square in Table 2. Ridge regression shrinks the largest coefficient in OLS regression towards 0 to avoid overfitting.

The regrssion indiates negative correlations between dependent variable and independent variables. In general, the car with bigger weights, more cylinders and bigger horsepower tend to consume more oil in the same mileage.

Principal Component Analysis

With scaled data, we conduct PCA on the design matrix consists of cyl, hp, wt, qsec and disp based on eigen decomposition.

```
library(LMjw)
X <- as.matrix(mtcars[,c("cyl","hp","wt","qsec","disp")])
X <- scale(X)
# PCA
pca<-myEigen_QR(var(X))
t<-rbind(pca$D,sqrt(pca$D)/sum(sqrt(pca$D))*100)
rownames(t) <- c("eigen_value", "proportion of variance")
colnames(t) <- c("Comp_1","Comp_2","Comp_3","Comp_4","Comp_5")
library(knitr)
kable(t, align = 'c',caption = "Principal Component Analysis Result")
```

Table 3: Principal Component Analysis Result

	Comp_1	Comp_2	Comp_3	Comp_4	Comp_5
eigen_value	3.766095	0.9240209	0.1541901	0.0906482	0.0650454
proportion of variance	50.397252	24.9633056	10.1974008	7.8188141	6.6232274

```
C <- X%*%pca$V[,1:2]
colnames(C)<-c("Comp_1","Comp_2")
save(C,file="C.RData")
```

According to Table 4, after orthogonal transformation the first two components explain 75% of the total variance, so they reserve the main information in the original data. Thus, we reduce the dimension of the data from five to two, and the two vectors have the nice property of being perpendicular.
The transformed data is obtained by multiply design matrix X and eigen vector, and reserved for further logistic regression analysis.

Logistic Regression

The median of mpg equals to 19.2. Therefore, we regard a car with mpg lower than 19.2 as high-mileage and marked with 1, otherwise a car is low-mileage and marked with 0. With the first two components from PCA, we can conduct logistic regression.

```
library(LMjw)
load("C.RData")
Y <- as.matrix(mtcars[,c("mpg")])
# test logistic Regression
m <- median(Y)
YL <- ifelse(Y<m,1,0)
set.seed(1)
train <- sample(32,26)
lgt<-myLogistic(C[train,],YL[train,])
(C[-train,]%*%lgt$beta>0)==YL[-train,]
```

```
##                   [,1]
## Mazda RX4         TRUE
## Merc 280          TRUE
## Toyota Corona     TRUE
## Dodge Challenger TRUE
## Fiat X1-9         TRUE
## Ferrari Dino      TRUE
```

```r
lgt<-myLogistic(C,YL)
t <- rbind(t(lgt$beta),lgt$se)
rownames(t)<-c("coefficient","standard_error")
colnames(t)<-c("Comp_1","Comp_2")
library(knitr)
kable(t, align = 'c',caption = "Logistic Regression Result")
```

Table 4: Logistic Regression Result

	Comp_1	Comp_2
coefficient	-1.8337544	1.131034
standard_error	0.6190652	0.814886

Firstly randomly sample 6 observations as testing data for cross-validation, after fitting the model, all 6 is classfied right. Then we conduct logistic regression on the whole dataset, and the result is reported in table 5.