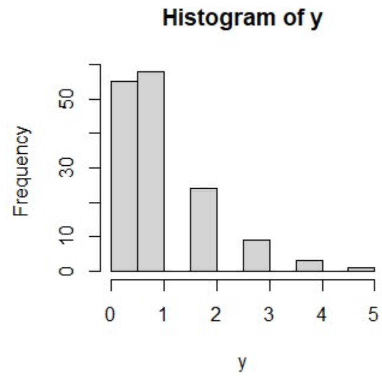
DL HW02 Yongha Kwak 2014121047

```
##(a) step1

N=150

P=50

X=matrix(NA,nrow=N,ncol=P)


# collinearity issue: X_ij ~ N(0,1)으로 제시되어 있으며, iid 조건이 명시되어 있지 않아 다중공선성 문제가 있는 사례로 분석하는 과정을 시행했습니다. (참고) 부분에서는 iid가 성립하는 경우를 별도로 진행했습니다.

covmat=matrix(rnorm(P^2,sd=2),nrow=P)

covmat=covmat+t(covmat)

U=eigen(covmat)$vectors

D=diag(rexp(P,rate=10))

covmat=U%*%D%*%t(U)


library(mvtnorm)

for(i in 1:N){

    X[i,]=rmvnorm(1,mean=rep(0,P),sigma=covmat)

}

X=data.frame(X)


#Step2: 첫 10개의 베타에 대하여만 의미가 있다고 보는 것

betas.true=c(rep(0.5,10),rep(0,P-10))


#Step3

X1=as.matrix(X)

y=rpois(N, lambda = exp(X1%*%betas.true))

hist(y)
```



#Step4

alldata=data.frame(cbind(y,X1))

names(alldata)[1] <- "y"

train=alldata[1:100,] #train data

test=alldata[101:150,] #test data

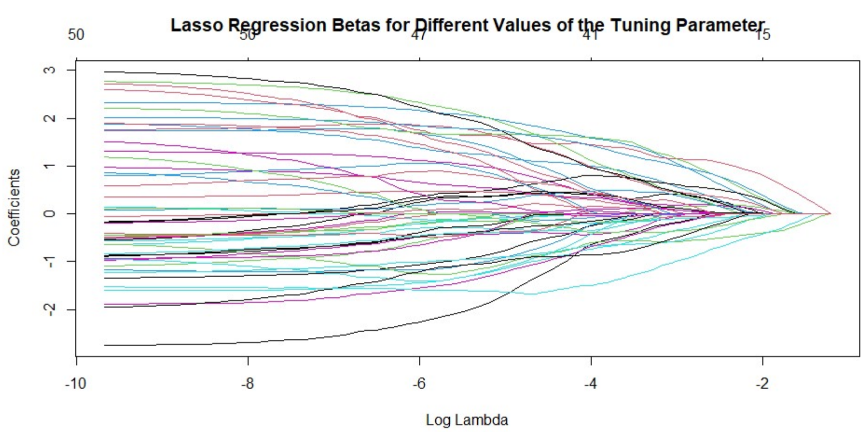
(b)

#fit lasso (trying 100 different lambda values)

lasso=glmnet(x=as.matrix(train[,-1]),y=as.numeric(train[,1]),alpha=1,nlambda=100,intercept = FALSE)

#intercept=FALSE, considering the model without intercep

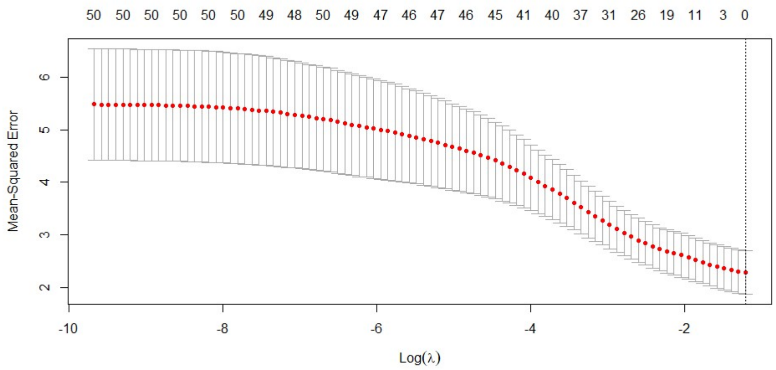
plot(lasso,xvar="lambda",main="Lasso Regression Betas for Different Values of the Tuning Parameter")



use 10-fold crossvalidation to find the best lambda

cv.lasso=cv.glmnet(x=as.matrix(train[,-1]),y=as.numeric(train[,1]),alpha=1,nfolds=10,intercept=FALSE)

plot(cv.lasso)



get lambda and best lasso fit

lambda.lasso=cv.lasso$lambda.min

lambda.lasso

0.3258969

(c)

betas.lasso = coef(cv.lasso, s='lambda.min')[-1]

MSE.lasso = mean((betas.lasso-betas.true)^2)

MSE.lasso

0.08440828

(d) MSPE

yhat.lasso=predict(cv.lasso,newx=as.matrix(test[,-1]),s="lambda.min")

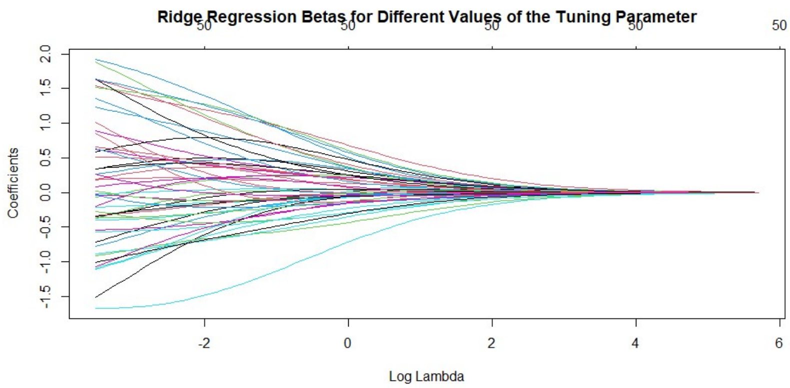
mspe.lasso=mean((test$y-yhat.lasso)^2)

mspe.lasso

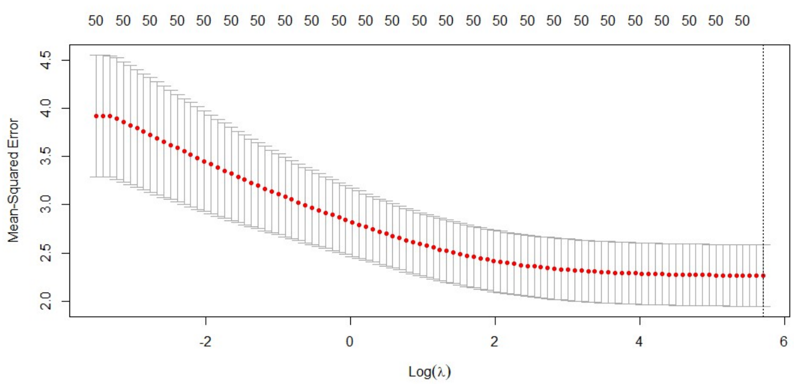
2.707174

(e) <b> ## fit ridge (trying 100 different lambda values)

```
rr=glmnet(x=as.matrix(train[,-1]),y=as.numeric(train[,1]),alpha=0,nlambda=100, intercept = FALSE)

plot(rr,xvar="lambda",main="Ridge Regression Betas for Different Values of the Tuning Parameter")
```




```
## use 10-fold crossvalidation to find the best lambda

cv.rr=cv.glmnet(x=as.matrix(train[,-1]),y=as.numeric(train[,1]),alpha=0,nfolds=10,nlambda=100, intercept=FALSE)

plot(cv.rr)
```




```
## get lambda and best rr fit

lambda.rr=cv.rr$lambda.min

lambda.rr
```

1.9537

<c>

```
betas.rr = coef(cv.rr, s='lambda.min')[-1]
```

```
mse.rr = mean((betas.rr-betas.true)^2)

mse.rr

0.1124435
```

<d>

```
yhat.rr = predict(cv.rr, newx = as.matrix(test[,-1]),s='lambda.min')

mspe.rr = mean((yhat.rr-test$y)^2)

mspe.rr

3.242378
```

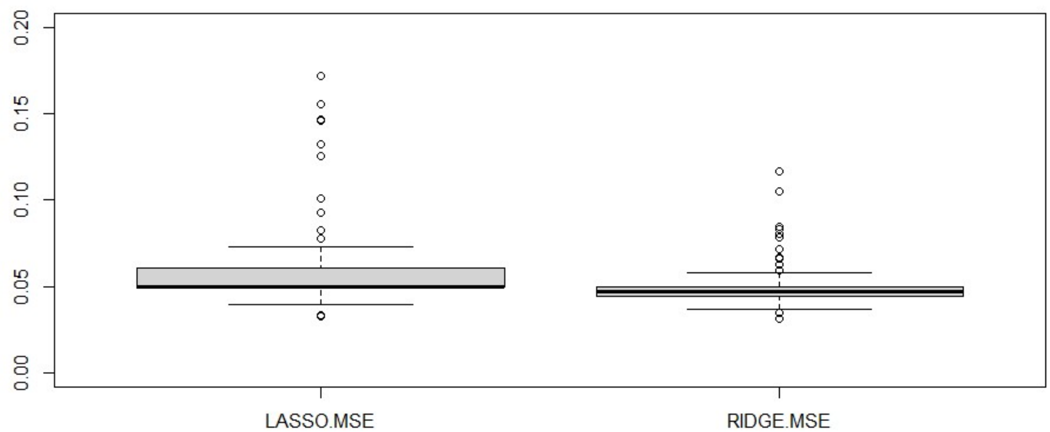
(f)

```
LASSO.MSE <- c()

LASSO.MSPE <- c()

RIDGE.MSE <- c()

RIDGE.MSPE <- c()


for (i in 1:100){

  set.seed(1000+i)

  #(1) step1

  X = matrix(rnorm(N*P,mean=0,sd=1),nrow=N,ncol=P)

  Y = rpois(N*P, lambda = exp(X%*%betas.true))

  #step2

  betas.true=c(rep(0.5,10),rep(0,P-10))

  #Step3

  X1=as.matrix(X)

  y=rpois(N, lambda = exp(X1%*%betas.true))

  #step4
```

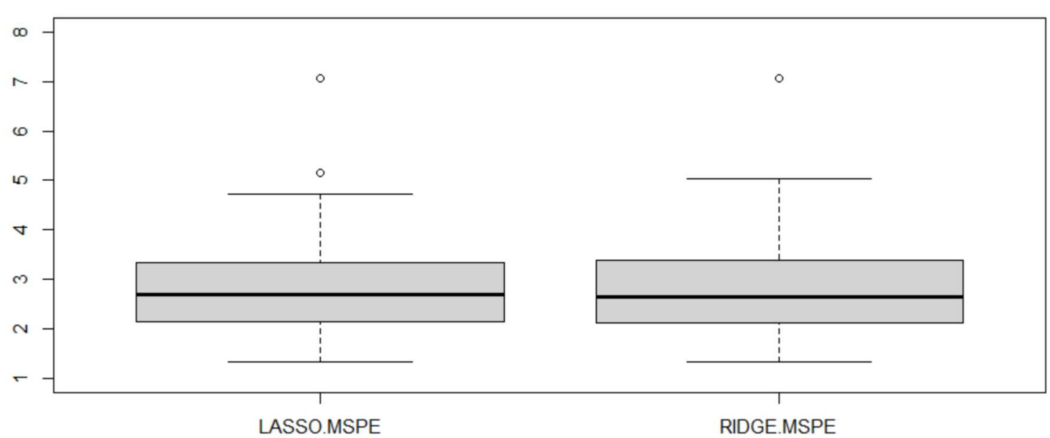
```
alldata=data.frame(cbind(y,X1))

names(alldata)[1] <- "y"

train=alldata[1:100,]

test=alldata[101:150,]

#(b)

lasso=glmnet(x=as.matrix(train[,-1]),y=as.numeric(train[,1]),alpha=1,nlambda=100,intercept = FALSE)

cv.lasso=cv.glmnet(x=as.matrix(train[,-1]),y=as.numeric(train[,1]),alpha=1,nfolds=10,intercept=FALSE)

lambda.lasso=cv.lasso$lambda.min

betas.lasso=coef(cv.lasso,s="lambda.min")

#(c)

betas.lasso = coef(cv.lasso, s='lambda.min')[-1]

MSE.lasso = mean((betas.lasso-betas.true)^2)

#(d)

yhat.lasso=predict(cv.lasso,newx=as.matrix(test[,-1]),s="lambda.min")

mspe.lasso=mean((test$y-yhat.lasso)^2)


##(e)

#<b>

rr=glmnet(x=as.matrix(train[,-1]),y=as.numeric(train[,1]),alpha=0,nlambda=100, intercept = FALSE)

cv.rr=cv.glmnet(x=as.matrix(train[,-1]),y=as.numeric(train[,1]),alpha=0,nfolds=10,nlambda=100,
intercept=FALSE)

lambda.rr=cv.rr$lambda.min

#<c>

betas.rr = coef(cv.rr, s='lambda.min')[-1]

mse.rr = mean((betas.rr-betas.true)^2)

#<d>

yhat.rr = predict(cv.rr, newx = as.matrix(test[,-1]),s='lambda.min')

mspe.rr = mean((yhat.rr-test$y)^2)
```

```
  ##insert results into sulten matrixes
  LASSO.MSE[i] <- as.numeric(MSE.lasso)
  LASSO.MSPE[i] <- as.numeric(mspe.lasso)
  RIDGE.MSE[i] <- as.numeric(mse.rr)
  RIDGE.MSPE[i] <- as.numeric(mspe.rr)
}
```

boxplot(LASSO.MSE,RIDGE.MSE,ylim=c(0,0.2),names=c("LASSO.MSE","RIDGE.MSE"))



boxplot(LASSO.MSPE, RIDGE.MSPE,ylim=c(1,8), names = c("LASSO.MSPE","RIDGE.MSPE"))



mLE<-mean(LASSO.MSE)

mRE<-mean(RIDGE.MSE)

mLE - mRE

0.01149082

mLP<-mean(LASSO.MSPE)

mRP<-mean(RIDGE.MSPE)

mLP-mRP

0.002413065

릿지 회귀분석을 사용할 경우가 MLE, MSPE 모두 라쏘 회귀분석에 비해 그 값이 작다. 이는 릿지 회귀분석이 변수 간 상관관계가 높은 상황에서 좋은 예측 성능을 가지기 때문입니다.

#VIF를 구한 결과는 다음과 같습니다. 초기 설정에서 알 수 있듯이 다중공선성 문제가 발생하고 있습니다.

```
library(car)

fit = lm(y~.,data=train)

vif(fit)

X1        X2        X3        X4        X5        X6
16.123761  6.820354  9.253963  2.722208 11.089990  5.595898
       X7        X8        X9       X10       X11       X12
 4.303775 17.052376 16.228399  8.636735 11.678696 15.941842
      X13       X14       X15       X16       X17       X18
14.801238 22.111869  4.342377 23.545487  7.214285  6.680663
      X19       X20       X21       X22       X23       X24
 7.267211  5.768597 57.677988 11.350326 10.015530  6.931919
      X25       X26       X27       X28       X29       X30
 6.924607  6.407969 39.930937 15.065471 22.094092  9.851024
      X31       X32       X33       X34       X35       X36
49.617375  7.975280 13.597775 17.087464  3.833062  9.263560
      X37       X38       X39       X40       X41       X42
12.134946 18.437190 12.775512 25.766544 15.228404 13.653576
      X43       X44       X45       X46       X47       X48
17.961733 11.525902 47.281932  3.172324  2.675902 49.685306
      X49       X50
22.867360 34.047339
```

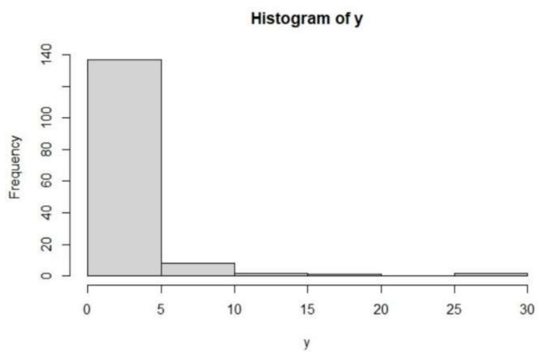
(참고)

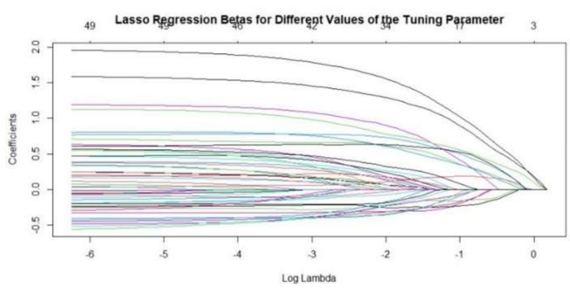
변수 간 상 관관계가 유의미하지 않은 경우에 대한 요약한 결과는 다음과 같습니다.

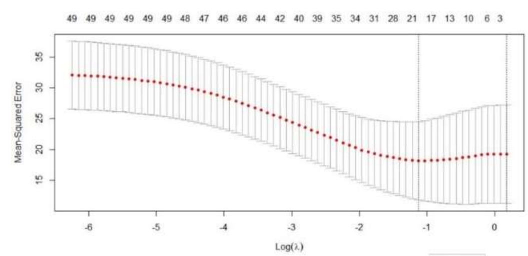
<코드> 달라진 부분 중심으로

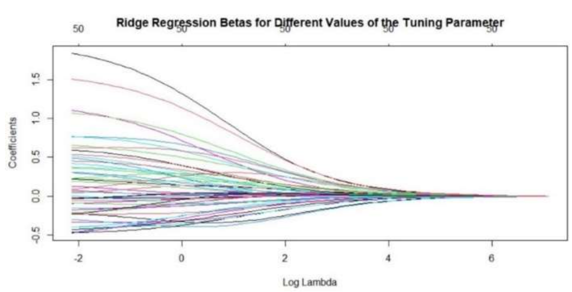
covmat=diag(P) **#iid를 가정하였습니다.**

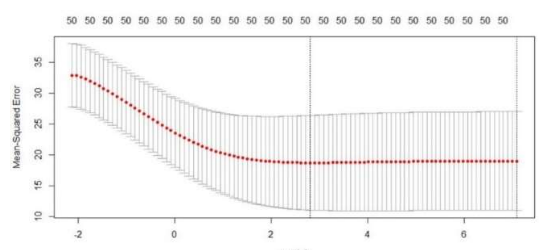
<결과>



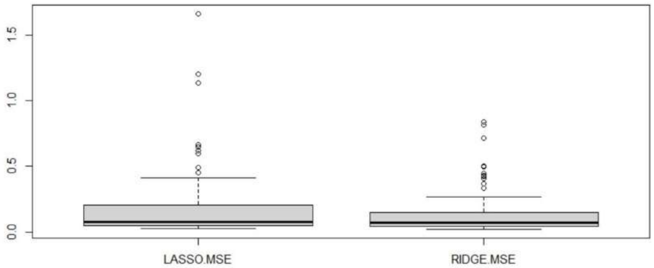








<비교 결과>



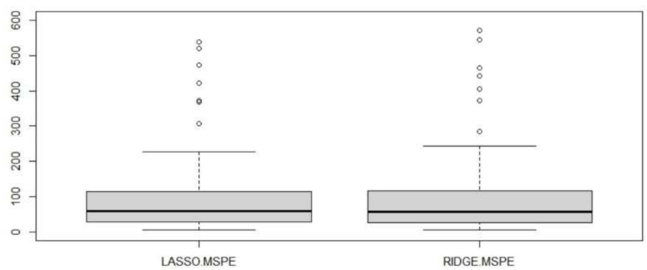
라쏘 회귀분석에 따른 MSE의 평균이 릿지 회귀분석의 그것에 비해 더 크다. 또한 라쏘의 경우 그 분산이 더 크다.

mLE<-mean(LASSO.MSE)

mRE<-mean(RIDGE.MSE)

mLE - mRE

0.04614203



라쏘의 MSPE의 평균이 릿지의 그것보다 작다. 또한 라쏘의 경우 그 분산이 더 작다.

mLP<-mean(LASSO.MSPE)

mRP<-mean(RIDGE.MSPE)

mLP-mRP

-0.5196767

Collinearity 문제가 작거나 없을 경우 릿지 회귀분석의 라쏘 회귀분석에 대한 비교우위가 잘 나타나지 않습니다.

참고_VIF

vif(fit)

X1	X2	X3	X4	X5	X6	X7
1.993286	1.974975	2.238591	1.763671	1.891699	1.918485	1.756981
X8	X9	X10	X11	X12	X13	X14
1.913731	2.198773	1.817111	2.112167	1.809683	2.344941	1.858455
X15	X16	X17	X18	X19	X20	X21
1.954206	2.029176	2.441336	1.809721	1.704674	1.719231	2.043217
X22	X23	X24	X25	X26	X27	X28
1.919856	1.824421	2.118079	1.553621	1.942006	1.786699	2.528623
X29	X30	X31	X32	X33	X34	X35
2.287209	1.981115	1.673997	2.138704	1.808017	2.447955	2.044879
X36	X37	X38	X39	X40	X41	X42
1.819848	2.227709	1.680455	1.664327	2.867414	1.772337	2.260365
X43	X44	X45	X46	X47	X48	X49
2.466501	1.594502	1.859811	1.716531	2.094346	1.969476	2.263376
X50						
1.862868						