```
#
# load data
#
LAozone = read.table("LAozone.data",sep=",",head=T)

#
## choose 300 as training, use the remining ones as test
#
n <- dim(LAozone)[1]
ntrain <- 300
ntest <- n-ntrain
set.seed(2013)
select <- sample(1:n)
trainID <- select[1:ntrain]
x.train <- LAozone[trainID,-1]
y.train <- LAozone[trainID,1]
x.test <- LAozone[-trainID,-1]
y.test <- LAozone[-trainID,1]

#
# fit OLS
#
ols.fit<- lm(y.train~.,data=cbind(y.train,x.train))
ols.pred <- predict(ols.fit, x.test)
ols.coef <- ols.fit$coef[-1]
ols.resid <- y.test-ols.pred
ols.result <- list(mean(ols.resid^2), sd(ols.resid^2)/sqrt
(ntest))


#
# fit LASSO
#
library(lars)
lasso.path<-lars(as.matrix(x.train),y.train,type="lasso")
myK <- 10
set.seed(2003)
lasso.cv<-cv.lars(as.matrix(x.train),y.train,K=myK,
type="lasso",mode="fraction")

# minimal cv
mincv <- which.min(lasso.cv$cv)
best_s <- lasso.cv$index[mincv]
lasso.coef<-coef(lasso.path,s=best_s,mode="fraction")
lasso.pred<-predict.lars(lasso.path,as.matrix
(x.test),s=best_s,type=c("fit"),mode="fraction")
lasso.resid <- y.test-lasso.pred$fit
lasso.result <- list(mean(lasso.resid^2),sd(lasso.resid^2)/sqrt
(ntest))

#
#print prediction results
```

```
#
print(ols.result)
print(lasso.result)

#
#print coefficient
#
print(ols.coef)
print(lasso.coef)

#
#1-se rule
#
#bound<-lasso.cv$cv[mincv]+lasso.cv$cv.error[mincv]
#tmpcv <- c(lasso.cv$cv[1:mincv],rep(0,length(lasso.cv$cv)-
#mincv))
#best_s<-lasso.cv$index[min(which(tmpcv<bound))]
```