

Ridge Regression Assignment

Ignasi Mañé, Antoni Company & Chiara Barbi

19/02/2019

Introduction

Write a report that contains the results of the computations that you are asked to carry out below, as well as the explanation of what you are doing. The main text (2 or 3 pages) should include pieces of source code and graphical and numerical output.

First, we load the Boston data set from the MASS package and the libraries required to do the exercise. We will use the caret package to split the data into k random folds whenever it is needed.

```
library(MASS)
library(caret)
data("Boston")
set.seed(5647)
#we set a seed to generate pseudo random numbers
```

1. Choosing the penalization parameter λ

1. Write an R function implementing the ridge regression penalization parameter λ choice based on the minimization of the mean squared prediction error in a validation set ($MSPE_{val}(\lambda)$).

Input: Matrix X and vector y corresponding to the training sample; matrix X_{val} and vector y_{val} corresponding to the validation set; a vector `lambda.v` of candidate values for λ .

Output: For each element λ in `lambda.v`, the value of $MSPE_{val}(\lambda)$. Additionally you can plot these values against $\log(1 + \lambda) - 1$ graphic, or against $df(\lambda)$.

We will create a function that it is used later on in the document.

```
ridgereg <- function(x,y,xval,yval,lambda,plot=TRUE) {

  p <- dim(x)[2] #number of parameters
  n <- dim(x)[1] #number of observations

  #empty results matrix
  results <- matrix(0,nrow = length(lambda), ncol = 2)
  colnames(results) <- c("lambda","pmse")

  for (i in 1:length(lambda)) {

    mspe <- fit.ridge(x,y,xval,yval,lambda=lambda[i])$mspe
    results[i,] <- c(lambda[i],mspe)

  }
}
```

```

df <- cal.edf(x=x,lambda=lambda)
results <- cbind(results,df)

if(plot == TRUE){

  plot.lam(results=results,lambda=lambda,df=df)

}

return(results)

}

```

2. Write an R function implementing the ridge regression penalization parameter $df(\lambda)$ choice based on k-fold cross-validation ($MSP_{E_{val}}(\lambda)$). Input, output and graphics as before.

```

ridgereg.cv <- function(x,y,k,lambda,plot=TRUE) {

  p <- dim(x)[2]
  n <- dim(x)[1]
  flds <- createFolds(y, k = k, list = TRUE, returnTrain = FALSE)
  kresults <- matrix(0,nrow=length(lambda) , ncol = k)
  results <- matrix(0,nrow = length(lambda), ncol = 2)
  colnames(results) <- c("lambda","pmse")

  for (j in 1:length(flds)) {

    filt <- unlist(flds[j], use.names=FALSE)
    mpse <- ridgereg(x=x[-filt,],y=y[-filt,],
                    xval=x[filt,],yval=y[filt,],
                    lambda = lambda,
                    plot = FALSE)
    kresults[,j] <- mpse[,2]

  }

  results[,1] <- lambda
  results[,2] <- rowMeans(kresults)

  df <- cal.edf(x=x,lambda=lambda)
  results <- cbind(results,df)

  if (plot==TRUE) {

    plot.lam(results=results,lambda=lambda,df=df)

  }

  value <- results[which.min(results[,2]),]
  return(value)

}

```

3. Consider the prostate data used in class. Use your routines to choose the penalization parameter λ by the following criteria: behavior in the validation set (the 30 observations not being in the training sample); 5-fold and 10-fold cross-validation. Compare your results with those obtained when using leave-one-out and generalized cross-validation.

We will split the data set into a train set and a test set to calculate lambda in exercise 3.1. However, we will use the entire data set in exercises from 3.2 to 3.4

```
prostate <- read.table("prostate_data.txt", header=TRUE, row.names = 1)
train.index <- which(prostate$train==TRUE)
test <- prostate[-train.index,]
train <- prostate[train.index,]

#Data for point 3.1
Y <- scale( train$lpsa, center=TRUE, scale=FALSE)
X <- scale( as.matrix(train[,1:8]), center=TRUE, scale=TRUE)
Yval <- scale( test$lpsa, center=TRUE, scale=FALSE)
Xval <- scale( as.matrix(test[,1:8]), center=TRUE, scale=TRUE)

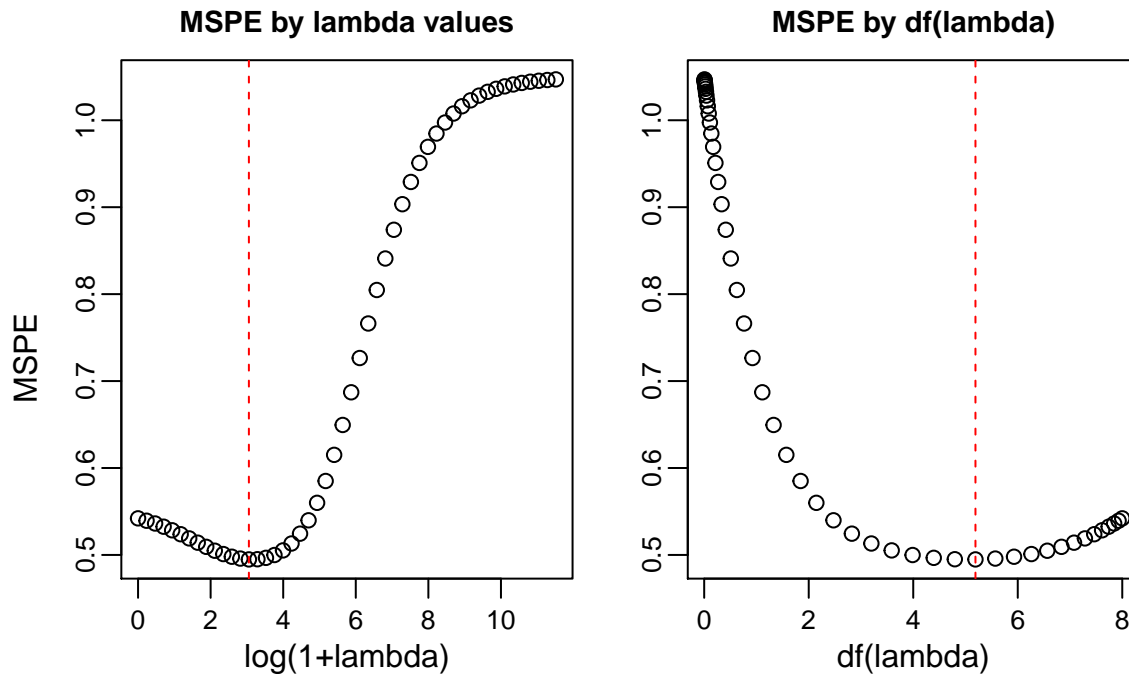
#Data for points 3.2, 3.3, 3.4
Xtot <- scale(as.matrix(prostate[,1:8]),center=TRUE , scale = TRUE)
Ytot <- scale(prostate$lpsa, center=TRUE , scale=FALSE)

#We create a matrix to store the optimal lambdas
optim.lambdas <- matrix(0,ncol=4,nrow=5)
colnames(optim.lambdas) <- c("method","lambda","pmse","df")
optim.lambdas <- data.frame(optim.lambdas)

#We generate a sequence of lambdas
lambda.max <- 1e5
n.lambdas <- 50
lambda.v <- exp(seq(0,log(lambda.max+1),length=n.lambdas))-1
```

3.1. Mean squared prediction error

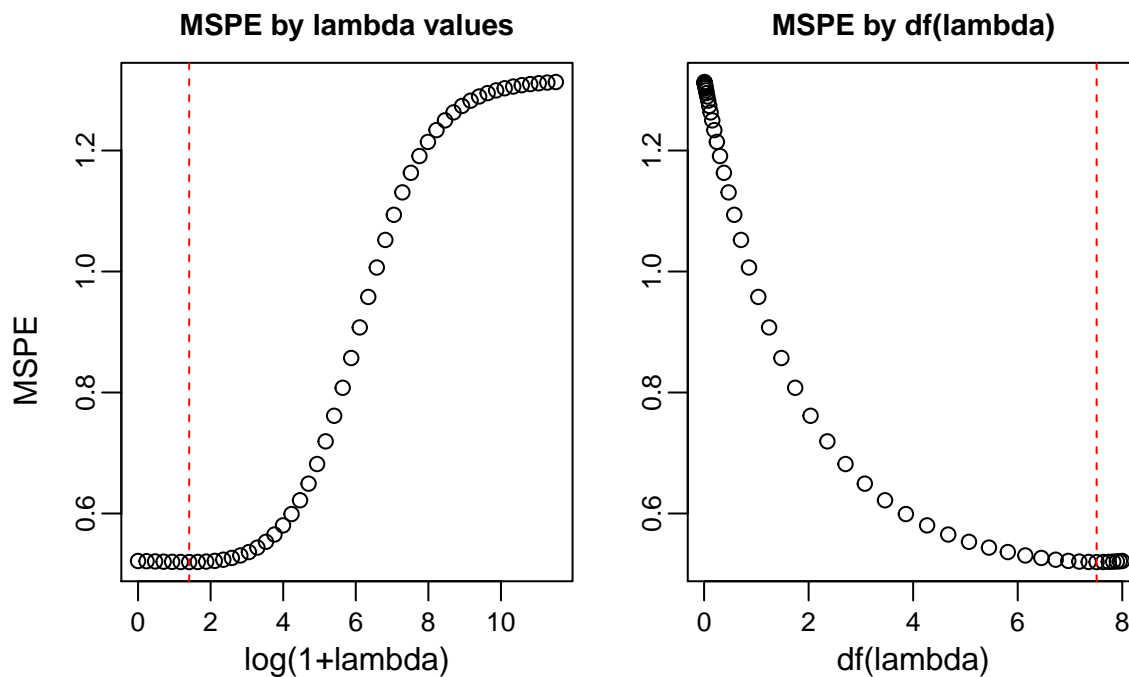
```
results.mspe <- ridgereg(x = X, y = Y, xval = Xval , yval = Yval ,lambda = lambda.v,plot = TRUE)
```



```
value <- results.mspe[which.min(results.mspe[,2]),]
optim.lambdas[1,] <- c("mspe",round(value,4))
```

3.2. 5-Folds Cross-Validation

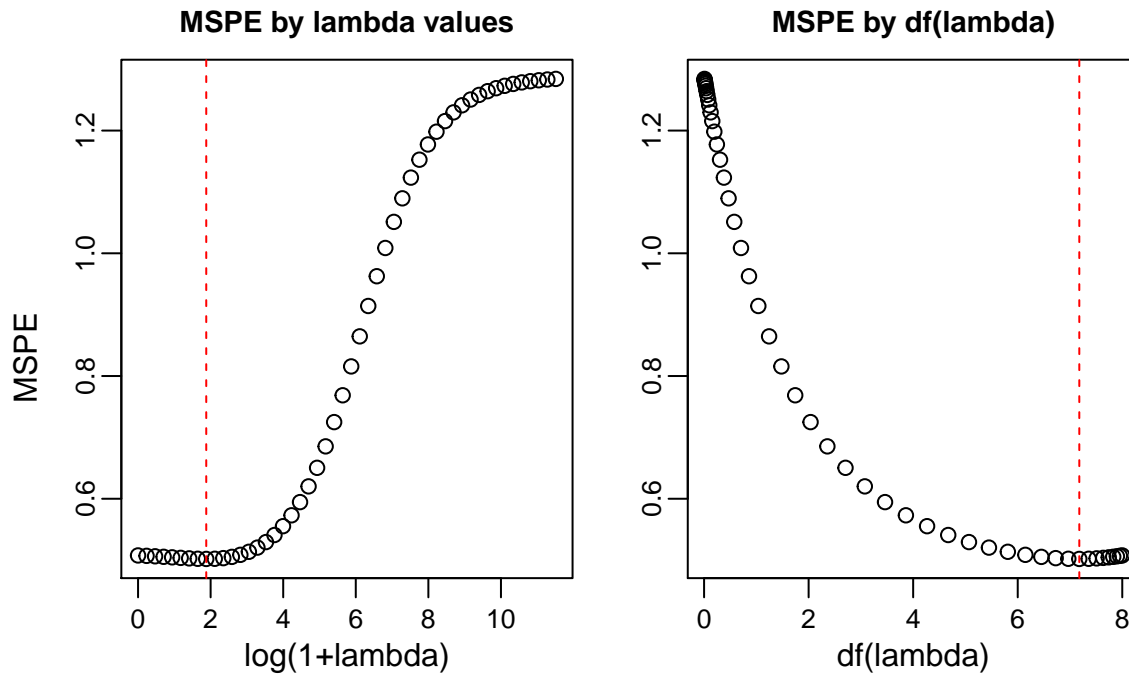
```
results.cv5 <- ridgereg.cv(x=Xtot,y=Ytot,k=5,lambda=lambda.v,plot=TRUE)
```



```
optim.lambdas[2,] <- c("5 fold cv",round(results.cv5,4))
```

3.3. 10-Folds Cross-Validation

```
results.cv10 <- ridgereg.cv(x=Xtot,y=Ytot,k=10,lambda=lambda.v,plot=TRUE)
```

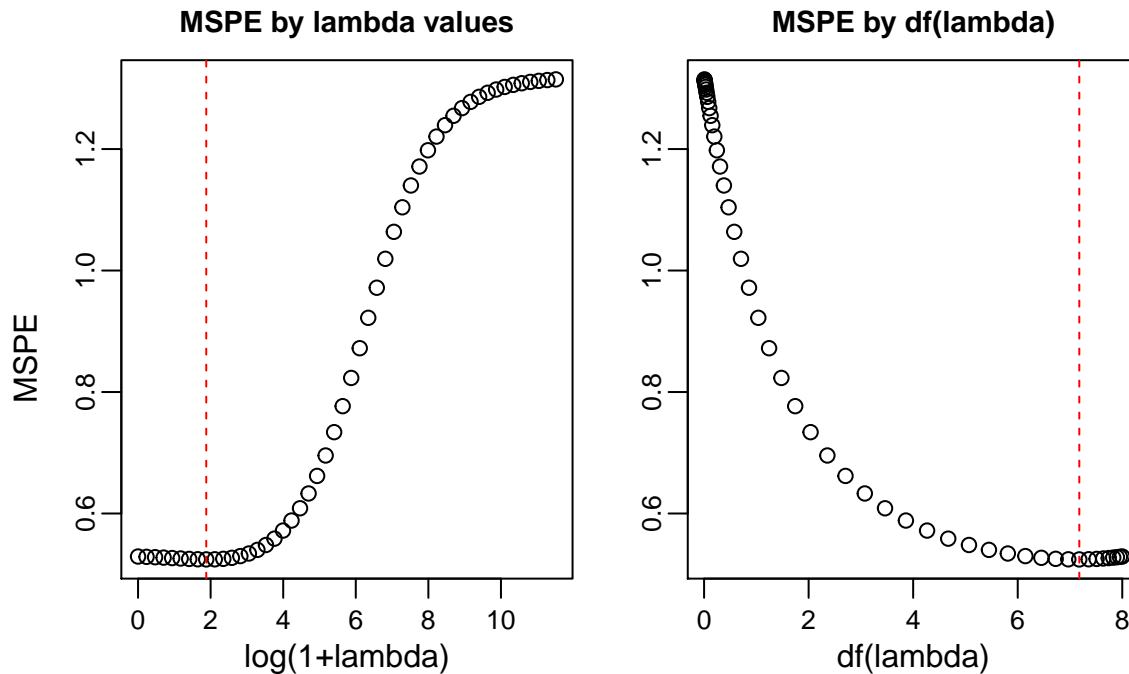


```
optim.lambdas[3,] <- c("10 fold cv",round(results.cv10,4))
```

3.3. Leave One Out Cross-Validation

Leave one out cross-validation is the particular case of kfold cross-validation in which $k=n$. However, since the algorithm that computes the kfold cv is very expensive, we will use a useful theoretical result for linear models that requires much less computational power.

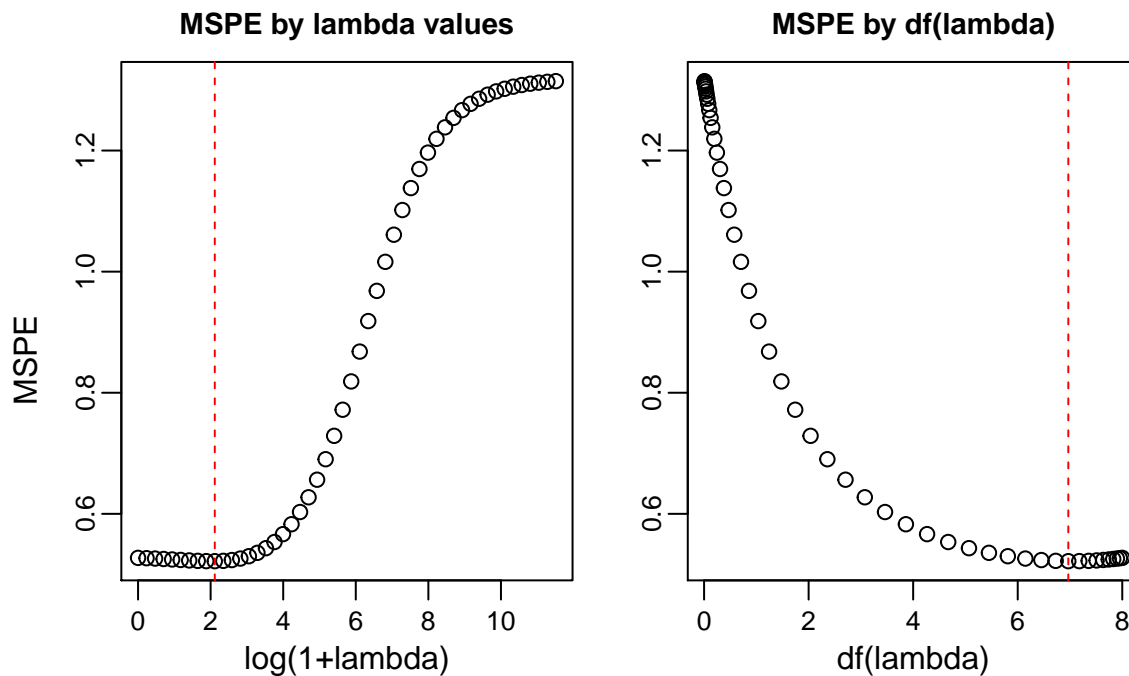
```
result.loo <- loo.g.cv(x=Xtot,y=Ytot,lambda=lambda.v,type="loocv")
```



```
optim.lambdas[4,] <- c("loo-cv",round(result.loo,4))
```

3.4. Generalized Cross-Validation

```
result.loo <- loo.g.cv(x=Xtot,y=Ytot,lambda=lambda.v,type="gcv")
```



```
optim.lambdas[5,] <- c("gcv",round(result.loo,4))
```

We obtain the following results:

```
##      method  lambda  pmse  df
```

```
## 1      mspe 20.2096 0.4949 5.1904
## 2  5 fold cv  3.0949 0.5197 7.5092
## 3 10 fold cv  5.5513 0.5019 7.1779
## 4    loo-cv  5.5513 0.5244 7.1779
## 5      gcv  7.2864 0.5218  6.968
```

For each of the cross-validation methods used to estimate the error as a function of lambda we select the value of lambda that minimizes it as the optimal value λ^* .

Method	λ^*	$df(\lambda^*)$	$E[MSPE]$
mspe	20.21	5.19	0.50
cv. 5 folds	3.10	7.51	0.52
cv. 10 folds	5.55	7.18	0.50
loo-cv	5.55	7.18	0.52
gcv	7.29	6.97	0.52

2. Ridge regression for the Boston Housing data

1. The Boston House-price corrected dataset (available in `boston.Rdata`) contains the same data (with some corrections) and it also includes the UTM coordinates of the geographical centers of each neighborhood. For the Boston House-price corrected dataset use ridge regression to fit the regression model where the response is MEDV and the explanatory variables are the remaining 13 variables in the previous list.

First we choose three different ridge regression penalization parameters λ^* using: 5-fold cross validation method, 10-fold cross-validation method and leave-one-out method.

```
data(Boston)

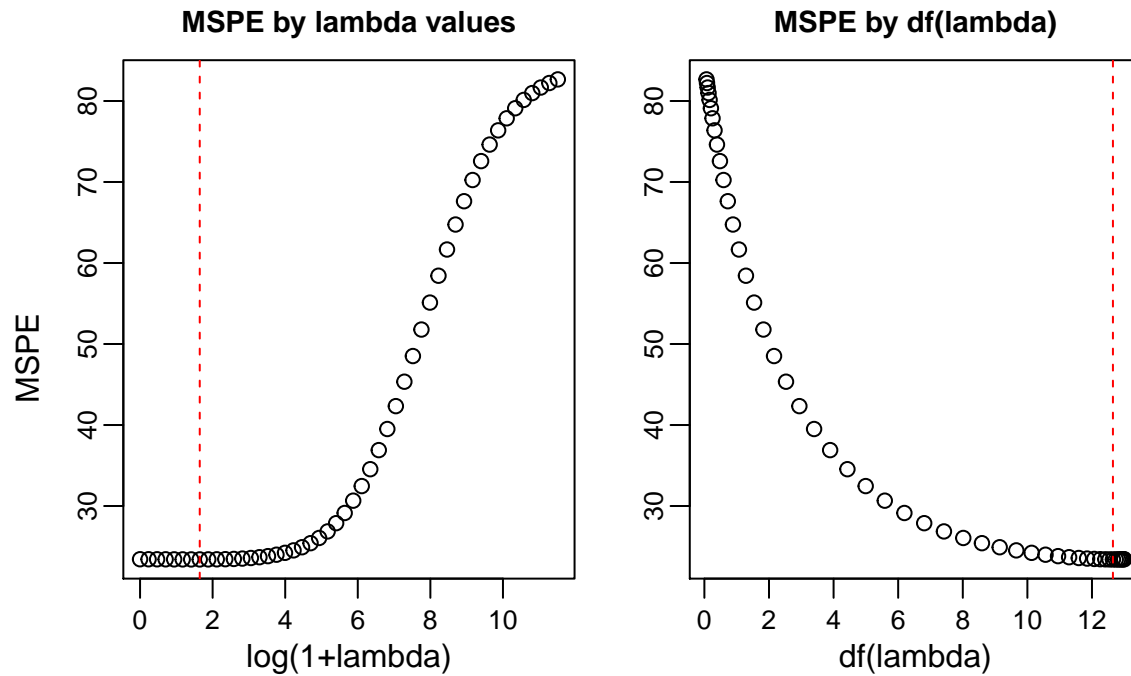
Y <- scale( Boston$medv, center=TRUE, scale=FALSE)
X <- scale( as.matrix(Boston[, -c(14)]), center=TRUE, scale=TRUE)
p <- dim(X)[2]
n <- dim(X)[1]
df <- data.frame(X, Y)

optim.lambda <- matrix(0, ncol=4, nrow=3)
colnames(optim.lambda) <- c("method", "lambda", "pmse", "df")
optim.lambda <- data.frame(optim.lambda)

lambda.max <- 1e5
n.lambdas <- 50
lambda.v <- exp(seq(0, log(lambda.max+1), length=n.lambdas))-1
```

1.1. 5-Fold Cross Validation

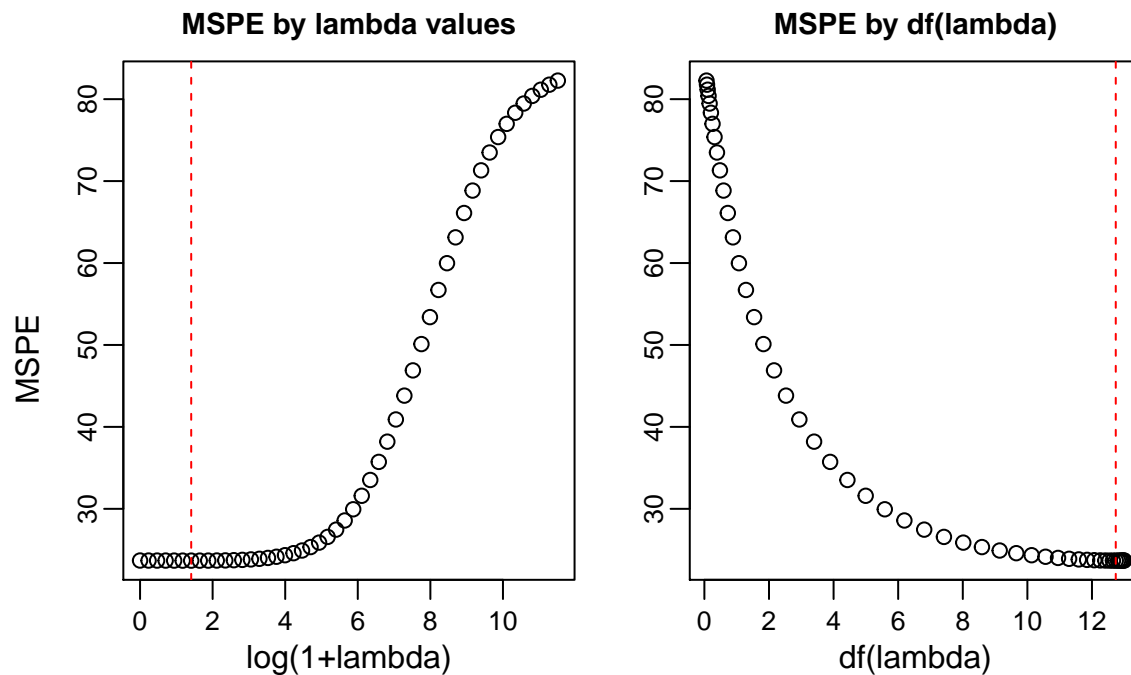
```
# k- fold 5 cross-validation
results.cv5 <- ridgereg.cv(x=X, y=Y, k=5, lambda=lambda.v, plot=TRUE)
```



```
optim.lambda[1,] <- c("5 fold cv",round(results.cv5,4))
```

1.2. 10-Fold Cross Validation

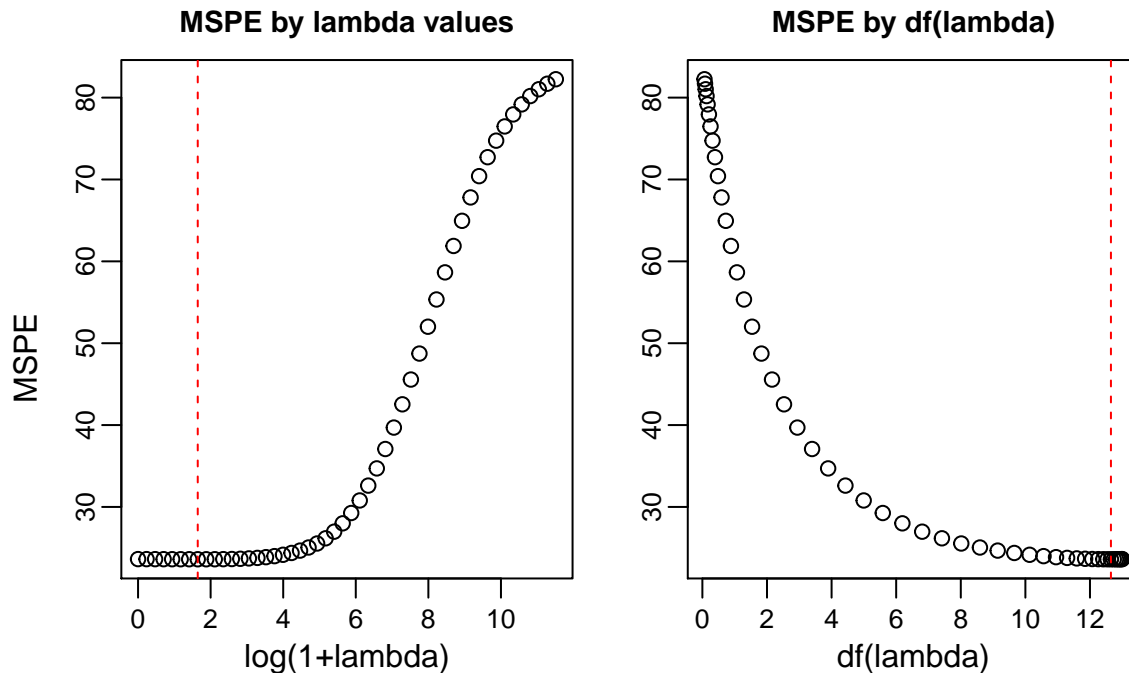
```
#k - fold 10 cross-validation
results.cv10 <- ridgereg.cv(x=X,y=Y,k=10,lambda=lambda.v,plot=TRUE)
```



```
optim.lambda[2,] <- c("10 fold cv",round(results.cv10,4))
```


1.3. Leave One Out Cross Validation

```
#loo
results.loo <- loo.g.cv(x=X,y=Y,lambda=lambda.v,type="loocv")
```



```
optim.lambda[3,] <- c("loo",round(results.loo,4))
optim.lambda
```

```
##      method lambda    pmse    df
## 1  5 fold cv 4.1795 23.4056 12.6491
## 2 10 fold cv 3.0949 23.6786 12.7361
## 3      loo 4.1795  23.61 12.6491
```

The method that provides the lowest value of the Mean Square Predicted Error is the 5-fold cross validation. Anyway there is not difference between this method and the Leave-one-out method, both in terms of degrees of freedom and λ^* value.

Finally, fitting the Ridge regression model using the optimal λ^* resulting from the 5-fold cross validation method provides us with the following results:

1.4. Beta Estimation for the Selected Lambda (scaled)

```
model <- fit.ridge(x=X,y=Y,xval=X,yval=Y,lambda=as.numeric(optim.lambda[1,2]))
model$beta
```

```
##      [,1]
## crim -0.896678766
## zn    1.023426610
## indus 0.051610792
## chas  0.695265953
## nox   -1.947957499
## rm    2.709761003
```

```
## age      -0.005057122
## dis      -2.998224860
## rad       2.393028311
## tax      -1.827300027
## ptratio  -2.029732770
## black     0.848302592
## lstat    -3.694627956
```