CS574 HW1

Kanak Choudhury

2/13/2020

library(glmnet)

library(ggplot2)

## 

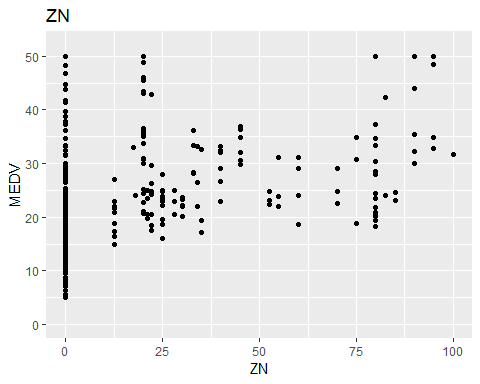
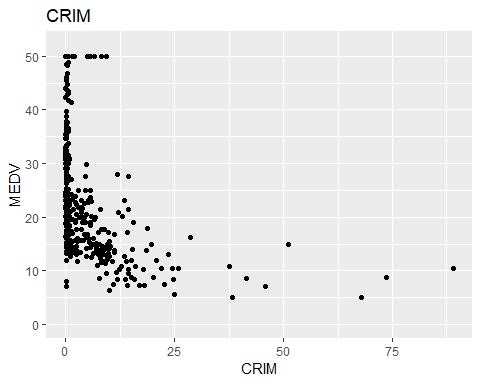
## Problem 2

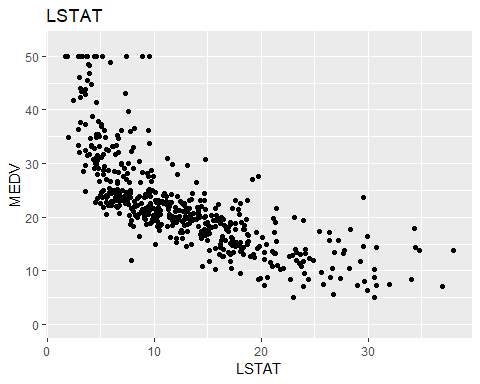
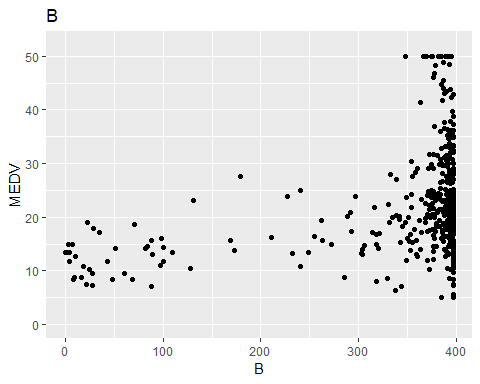
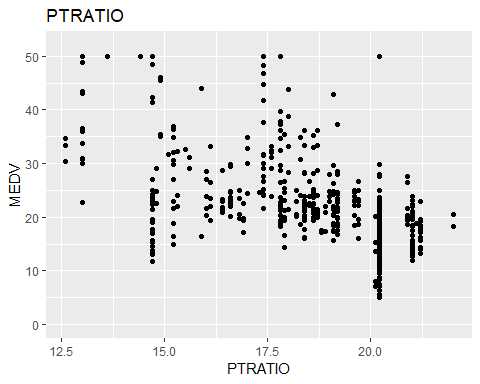
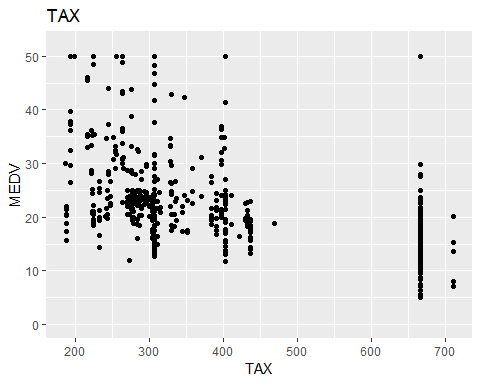
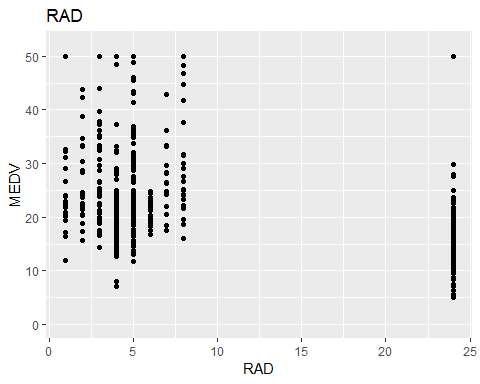
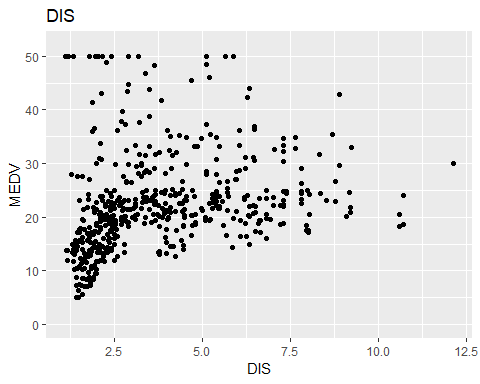
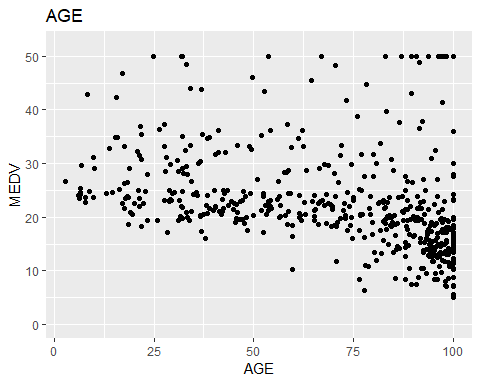
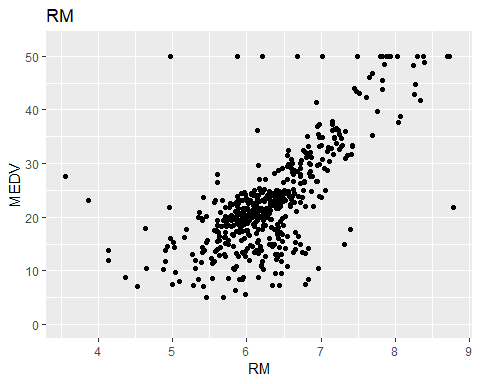
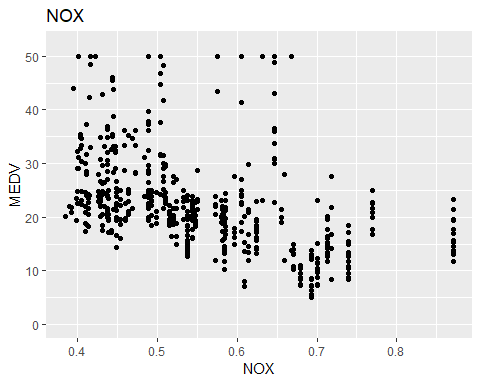
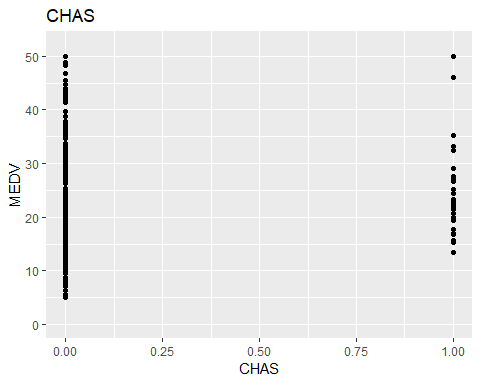
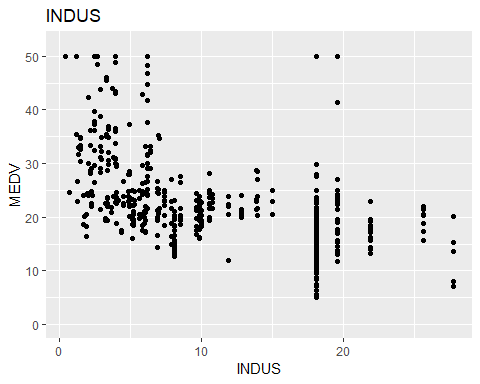
path = "D:\\ISU\\COMS 574 - Introduction to Machine Learning\\HW\\HW1\\"  
  
dt = read.csv(paste(path, "housingdata.csv", sep = ""), header = T)  
varname = names(dt)

## 

## (a)

for (i in c(1:13)){  
 print(ggplot(dt, aes\_string(x=varname[i], y=varname[14])) +   
 ylim(0,max(dt$MEDV+2))+  
 ggtitle(varname[i]) +  
 geom\_point())  
}





Based on scatter plot, LSTAT (% lower status of the population) (negative relation) and RM (average number of rooms per dwelling) (positive relation) have moderate (highest among the features) linear association (-0.74 and 0.69) with MEDV (Median value of owner-occupied homes in $1000’s). On the other hand, B ( where is the proportion of black residents by town) has the least relation with the response variable. Though, CHAS is a binary variable, it seems that there is a high relation with response variable. It is clear from the scatter plot that there is nonlinear or quadratic relation between DIS (weighted distances to ﬁve Boston employment centres) and MEDV. All other variables have some relation with response variable. However, with respect to high order space (interaction with other variables or higher order) there might have some strong relation with the response variable.

## (b)

dt\_train = dt[1:400,]  
dt\_test = dt[401:dim(dt)[1],]  
varn = c("AGE", "INDUS", "NOX", "RM", "TAX")  
  
reg\_res = list()  
tr\_mse = list()  
k = 1  
for (i in c(0:length(varn))){  
 comb = combn(varn, i)  
 for (j in c(1:dim(comb)[2])){  
 if (dim(comb)[1]==0){  
 formu = "MEDV ~ 1"  
 comb = matrix(c("Intercept"), nrow = 1, ncol = 1)  
 } else{  
 formu = paste("MEDV ~ 1", paste((comb[,j]), collapse = "+"), sep = "+")   
 }  
   
 reg\_res[[k]] = lm(as.formula(formu), data = dt\_train)  
 tr\_mse[[k]] = list(i = i, var = paste((comb[,j]), collapse = ","),   
 tr\_mse = mean(reg\_res[[k]]$residuals^2),   
 vel\_mse = mean((predict(reg\_res[[k]], newdata = dt\_test)-dt\_test$MEDV)^2))  
 k = k+1  
 }  
   
}  
  
res = do.call(rbind.data.frame, tr\_mse)  
names(res) = c("subset", "variables", "tr\_mse", "ts\_mse")  
  
trset = NULL  
for (i in c(0:length(varn))){  
 aa = which(res$tr\_mse == res[res$subset == i,][which.min(res[res$subset == i,3]),3])  
 trset = rbind(trset, res[aa,])  
 res1 = reg\_res[[aa]]  
 len1 = length(res1$coefficients)  
 if (i==0){  
 print(paste("For subset model ", i, ": y^hat = ", round(res1$coefficients[1], 3) ,  
 " with MSE = ", round(res[aa,3],3)))   
 } else {  
   
 print(paste("For subset model ", i, ": y^hat = ", round(res1$coefficients[1], 3), " + " ,  
 paste(round(res1$coefficients[2:len1], 3),   
 names(res1$coefficients)[2:len1], sep = " \* ", collapse = " + "),  
 " with training MSE = ", round(res[aa, 3],3)))  
 }  
}

print(trset)

aa = which.min(res$tr\_mse)  
res1 = reg\_res[[aa]]  
len1 = length(res1$coefficients)  
print(paste("For subset model ", res[aa,]$subset, ": y^hat = ", round(res1$coefficients[1], 3),   
 " + " , paste(round(res1$coefficients[2:len1], 3),   
 names(res1$coefficients)[2:len1], sep = " \* ", collapse = " + "),  
 " with training MSE = ", round(res[aa, 3],3), " and testing MSE = ", round(res[aa, 4],3)))

aa = which.min(res$ts\_mse)  
res1 = reg\_res[[aa]]  
len1 = length(res1$coefficients)  
print(paste("For subset model ", res[aa,]$subset, ": y^hat = ", round(res1$coefficients[1], 3),   
 " + " ,  
 paste(round(res1$coefficients[2:len1], 3),   
 names(res1$coefficients)[2:len1], sep = " \* ", collapse = " + "),  
 " with training MSE = ", round(res[aa, 3],3), " and testing MSE = ", round(res[aa, 4],3)))

## [1] "For subset model 0 : y^hat = 24.334 with MSE = 83.807"  
## [1] "For subset model 1 : y^hat = -35.261 + 9.406 \* RM with training MSE = 36.303"  
## [1] "For subset model 2 : y^hat = -31.076 + -0.034 \* AGE + 9.09 \* RM with training MSE = 35.381"  
## [1] "For subset model 3 : y^hat = -29.222 + -0.028 \* AGE + 8.944 \* RM + -0.004 \* TAX with training MSE = 35.193"  
## [1] "For subset model 4 : y^hat = -30.586 + -0.036 \* AGE + 3.747 \* NOX + 8.991 \* RM + -0.005 \* TAX with training MSE = 35.125"  
## [1] "For subset model 5 : y^hat = -30.628 + -0.034 \* AGE + -0.031 \* INDUS + 4.573 \* NOX + 8.947 \* RM + -0.005 \* TAX with training MSE = 35.11"

## subset variables tr\_mse ts\_mse  
## 2 0 Intercept 83.80701 102.22659  
## 5 1 RM 36.30331 79.62583  
## 9 2 AGE,RM 35.38101 67.97003  
## 22 3 AGE,RM,TAX 35.19273 57.58572  
## 30 4 AGE,NOX,RM,TAX 35.12503 57.85750  
## 32 5 AGE,INDUS,NOX,RM,TAX 35.11013 57.66880

## [1] "For subset model 5 : y^hat = -30.628 + -0.034 \* AGE + -0.031 \* INDUS + 4.573 \* NOX + 8.947 \* RM + -0.005 \* TAX with training MSE = 35.11 and testing MSE = 57.669"

## [1] "For subset model 3 : y^hat = 35.763 + -0.04 \* AGE + -7.286 \* NOX + -0.014 \* TAX with training MSE = 73.333 and testing MSE = 30.002"

The best fitting linear model for every subset of AGE, INDUS, NOX, RM, TAX using the first n = 400 samples based on training MSE is the model with all five features. However, based on test MSE, the based model is the model with variables AGE, NOX and TAX.

## 

## (c)

### (i)

valset = NULL  
for (i in c(0:length(varn))){  
 aa = which(res$ts\_mse == res[res$subset == i,][which.min(res[res$subset == i,4]),4])  
 valset = rbind(valset, res[aa,])  
 res1 = reg\_res[[aa]]  
 len1 = length(res1$coefficients)  
 if (i==0){  
 print(paste("For subset model ", i, ": y^hat = ", round(res1$coefficients[1], 3) ,  
 " with MSE = ", round(res[aa,4],3)))   
 } else {  
   
 print(paste("For subset model ", i, ": y^hat = ", round(res1$coefficients[1], 3), " + " ,  
 paste(round(res1$coefficients[2:len1], 3),   
 names(res1$coefficients)[2:len1], sep = " \* ", collapse = " + "),  
 " with validation MSE = ", round(res[aa,4],3)))  
 }  
}

## [1] "For subset model 0 : y^hat = 24.334 with MSE = 102.227"  
## [1] "For subset model 1 : y^hat = 32.103 + -0.022 \* TAX with validation MSE = 31.616"  
## [1] "For subset model 2 : y^hat = 33.753 + -0.056 \* AGE + -0.017 \* TAX with validation MSE = 30.207"  
## [1] "For subset model 3 : y^hat = 35.763 + -0.04 \* AGE + -7.286 \* NOX + -0.014 \* TAX with validation MSE = 30.002"  
## [1] "For subset model 4 : y^hat = 31.131 + -0.026 \* AGE + -0.388 \* INDUS + 3.887 \* NOX + -0.01 \* TAX with validation MSE = 31.431"  
## [1] "For subset model 5 : y^hat = -30.628 + -0.034 \* AGE + -0.031 \* INDUS + 4.573 \* NOX + 8.947 \* RM + -0.005 \* TAX with validation MSE = 57.669"

print(valset)

## subset variables tr\_mse ts\_mse  
## 2 0 Intercept 83.80701 102.22659  
## 6 1 TAX 75.69373 31.61631  
## 10 2 AGE,TAX 73.59263 30.20671  
## 21 3 AGE,NOX,TAX 73.33258 30.00196  
## 28 4 AGE,INDUS,NOX,TAX 70.78130 31.43145  
## 32 5 AGE,INDUS,NOX,RM,TAX 35.11013 57.66880

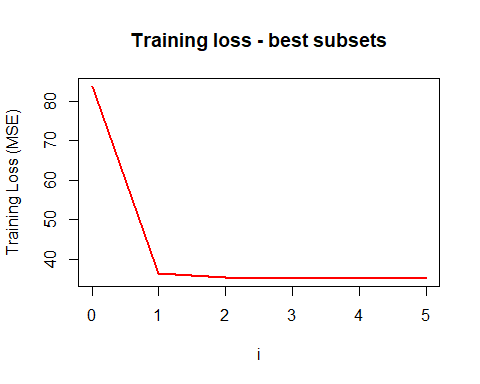
The best subset was for each is given in the above table.

These six models are completely nested with largest order model. If we consider zero for the appropriate parameter(s), we can get any lower order model.

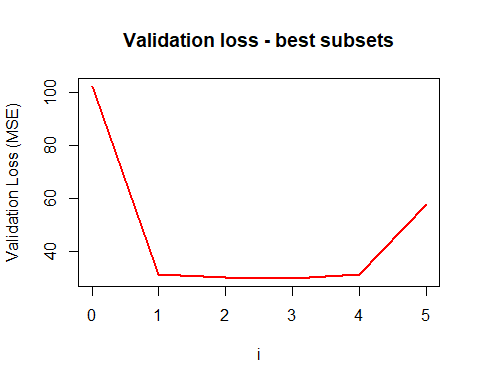
### 

### (ii)

plot(trset$subset, trset$tr\_mse, type = "l", xlab = "i", lwd = 2, col = 2,  
 ylab = "Training Loss (MSE)")  
title("Training loss - best subsets")



plot(valset$subset, valset$ts\_mse, type = "l", xlab = "i", lwd = 2, col = 2,  
 ylab = "Validation Loss (MSE)")  
title("Validation loss - best subsets")



As the number of features increases in the model, the training MSE decreases. However, based on the validation loss, it looks like “U” shaped. That indicates that more complex model is not always better for prediction.

### (iii)

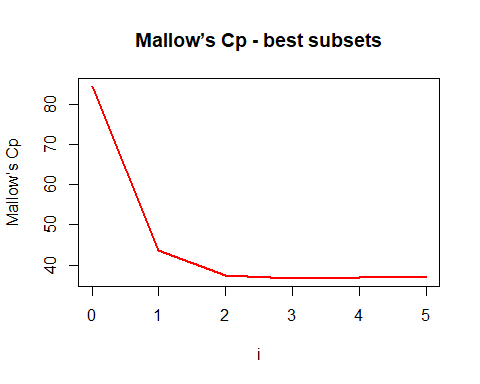
reg\_res\_cp = list()  
tr\_mse\_cp = list()  
k = 1  
for (i in c(0:length(varn))){  
 comb = combn(varn, i)  
 for (j in c(1:dim(comb)[2])){  
 if (dim(comb)[1]==0){  
 formu = "MEDV ~ 1"  
 comb = matrix(c("Intercept"), nrow = 1, ncol = 1)  
 } else{  
 formu = paste("MEDV ~ 1", paste((comb[,j]), collapse = "+"), sep = "+")   
 }  
   
 reg\_res\_cp[[k]] = lm(as.formula(formu), data = dt)  
 trmse = mean(reg\_res\_cp[[k]]$residuals^2)  
 tr\_mse\_cp[[k]] = list(i = i, var = paste((comb[,j]), collapse = ","),   
 tr\_mse\_cp = trmse)  
 k = k+1  
 }  
   
}  
  
res\_cp = do.call(rbind.data.frame, tr\_mse\_cp)  
names(res\_cp) = c("subset", "variables", "tr\_mse")  
res\_cp$cp = res\_cp$tr\_mse + 2\*res\_cp$subset\*res\_cp$tr\_mse[nrow(res\_cp)] / nrow(dt)  
  
  
trset\_cp = NULL  
for (i in c(0:length(varn))){  
 aa = which(res\_cp$cp == res\_cp[res\_cp$subset == i,][which.min(res\_cp[res\_cp$subset == i,4]),4])  
 trset\_cp = rbind(trset\_cp, res\_cp[aa,])  
 res\_cp1 = reg\_res\_cp[[aa]]  
 len1 = length(res\_cp1$coefficients)  
 if (i==0){  
 print(paste("For subset model ", i, ": y^hat = ", round(res\_cp1$coefficients[1], 3) ,  
 " with MSE = ", round(res\_cp[aa,3],3)))   
 } else {  
   
 print(paste("For subset model ", i, ": y^hat = ", round(res\_cp1$coefficients[1], 3), " + " ,  
 paste(round(res\_cp1$coefficients[2:len1], 3),   
 names(res\_cp1$coefficients)[2:len1], sep = " \* ", collapse = " + "),  
 " with Mallow’s Cp = ", round(res\_cp[aa, 4],3)))  
 }  
}

aa = which.min(res\_cp$cp)  
res\_cp1 = reg\_res\_cp[[aa]]  
len1 = length(res\_cp1$coefficients)  
print(paste("For subset model ", res\_cp$subset[aa], ": y^hat = ", round(res\_cp1$coefficients[1], 3), " + " ,  
 paste(round(res\_cp1$coefficients[2:len1], 3),   
 names(res\_cp1$coefficients)[2:len1], sep = " \* ", collapse = " + "),  
 " with Mallow’s Cp = ", round(res\_cp[aa, 4],3)))

## [1] "For subset model 3 : y^hat = -18.955 + -0.037 \* AGE + 7.843 \* RM + -0.013 \* TAX with Mallow’s Cp = 36.73"

plot(trset\_cp$subset, trset\_cp$cp, type = "l", xlab = "i", lwd = 2, col = 2,  
 ylab = "Mallow’s Cp")  
title("Mallow’s Cp - best subsets")

## [1] "For subset model 0 : y^hat = 22.533 with MSE = 84.42"  
## [1] "For subset model 1 : y^hat = -34.671 + 9.102 \* RM with Mallow’s Cp = 43.744"  
## [1] "For subset model 2 : y^hat = -21.233 + 7.993 \* RM + -0.016 \* TAX with Mallow’s Cp = 37.384"  
## [1] "For subset model 3 : y^hat = -18.955 + -0.037 \* AGE + 7.843 \* RM + -0.013 \* TAX with Mallow’s Cp = 36.73"  
## [1] "For subset model 4 : y^hat = -18.803 + -0.035 \* AGE + -0.022 \* INDUS + 7.808 \* RM + -0.012 \* TAX with Mallow’s Cp = 36.866"  
## [1] "For subset model 5 : y^hat = -18.497 + -0.034 \* AGE + -0.016 \* INDUS + -0.942 \* NOX + 7.807 \* RM + -0.012 \* TAX with Mallow’s Cp = 37.005"



The best fitting linear model for every subset of AGE, INDUS, NOX, RM, TAX using all the samples based on is the model with variables AGE, RM and TAX.

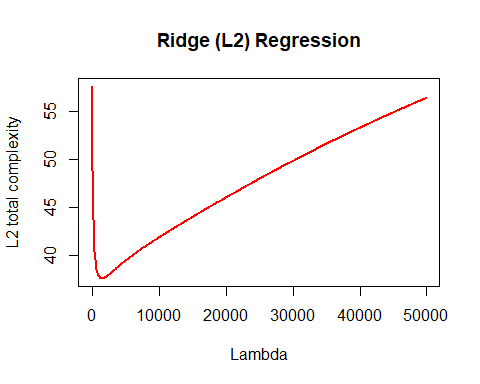
Plot using total complexity and plot using validation set MSE show the same pattern. gives more weight to the model with higher complexity. However, That does not mean and validation set MSE behave the same way. It actually depends on data. For some data set, these two methods can provide the same result. However, if prediction is the main purpose of study, model section based on validation set total loss would be better.

### 

### (iv)

#### (i) - (iii)

dt\_train = dt[1:400,]  
dt\_test = dt[401:dim(dt)[1],]  
varn = c("AGE", "INDUS", "NOX", "RM", "TAX")  
  
lambda = 0  
l2\_res = NULL  
  
loop = TRUE  
  
while (loop) {  
 a = glmnet(x = as.matrix(dt\_train[,varn]), y = dt\_train[,"MEDV"], standardize=FALSE,   
 alpha = 0, lambda = lambda)  
 pred = predict(a, as.matrix(dt\_test[,varn]))  
 l2 = mean((dt\_test$MEDV - pred)^2) + lambda \* sum(a$beta^2)  
 l2\_res = rbind(l2\_res, c(a$lambda, l2))  
 if (lambda > 1400 & lambda <1600){  
 lambda = lambda + 0.05  
 }else {  
 lambda = lambda + 100  
 }  
   
 if(sum(a$beta^2) < 1e-5 | lambda > 50000) loop = FALSE  
}  
  
l2\_res = as.data.frame(l2\_res)  
plot(l2\_res$V1, l2\_res$V2, type = "l", xlab = "Lambda", lwd = 2, col = 2,  
 ylab = "L2 total complexity")  
title("Ridge (L2) Regression")



aa = which.min(l2\_res$V2)  
a = glmnet(x = as.matrix(dt\_train[,varn]), y = dt\_train[,"MEDV"], standardize=FALSE,   
 alpha = 0, lambda = l2\_res[aa,1])  
pred = predict(a, as.matrix(dt\_test[,varn]))  
l2 = mean((dt\_test$MEDV - pred)^2)  
  
print(paste("Model with lambda = ", round(l2\_res[aa,1],3), ": y^hat = ", round(a$a0, 3),   
 " + " ,  
 paste(round(a$beta, 3),   
 row.names(a$beta), sep = " \* ", collapse = " + "),  
 " with testing MSE = ", round(l2,3)))

## [1] "Model with lambda = 1503.3 : y^hat = 33.047 + -0.042 \* AGE + -0.045 \* INDUS + 0 \* NOX + 0.026 \* RM + -0.017 \* TAX with testing MSE = 30.591"

print(paste("Model with lambda = ", l2\_res[aa,1], " has lowest validation MSE, ", round(l2\_res[aa,2],3)))

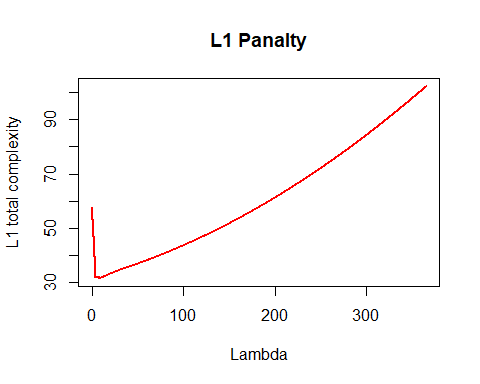
## [1] "Model with lambda = 1503.3 has lowest validation MSE, 37.628"

From this plot, we can see that model with with AGE, INDUS, RM, TAX variables has the smallest validation set total complexity. However, using and validation set MSE, we got smaller set model though both were different.

#### 

#### (iv)

lambda = 0  
l1\_res = NULL  
  
loop = TRUE  
  
while (loop) {  
 a = glmnet(x = as.matrix(dt\_train[,varn]), y = dt\_train[,"MEDV"], standardize=FALSE,   
 alpha = 1, lambda = lambda)  
 pred = predict(a, as.matrix(dt\_test[,varn]))  
 l1 = mean((dt\_test$MEDV - pred)^2) + lambda \* sum(abs(a$beta))  
 l1\_res = rbind(l1\_res, c(a$lambda, l1))  
 lambda = lambda + 0.05  
 if(sum(abs(a$beta)) < 1e-8) loop = FALSE  
}  
  
l1\_res = as.data.frame(l1\_res)  
plot(l1\_res$V1, l1\_res$V2, type = "l", xlab = "Lambda", lwd = 2, col = 2,  
 ylab = "L1 total complexity")  
title("L1 Panalty")



aa = which.min(l1\_res$V2)  
a = glmnet(x = as.matrix(dt\_train[,varn]), y = dt\_train[,"MEDV"], standardize=FALSE,   
 alpha = 1, lambda = l1\_res[aa,1])  
pred = predict(a, as.matrix(dt\_test[,varn]))  
mmse = mean((dt\_test$MEDV - pred)^2)  
  
print(paste("Model with lambda = ", round(l1\_res[aa,1],3), ": y^hat = ", round(a$a0, 3),   
 " + " ,  
 paste(round(a$beta, 3),   
 row.names(a$beta), sep = " \* ", collapse = " + "),  
 " with testing MSE = ", round(mmse,3)))

## [1] "Model with lambda = 8.6 : y^hat = 33.229 + -0.044 \* AGE + -0.001 \* INDUS + 0 \* NOX + 0 \* RM + -0.017 \* TAX with testing MSE = 31.106"

print(paste("Model with lambda = ", round(l1\_res[aa,1], 3), " has lowest validation MSE, ", round(mmse,3)))

## [1] "Model with lambda = 8.6 has lowest validation MSE, 31.106"

LASSO model also shows the same patter as Ridge regression. However, LASSO model selects smaller set variables that the Ridge regression. It includes only AGE, INDUS, TAX variables. It is important to mention that model selection based on different criteria selects different sets of variables in the respective best model.

#### 

#### (v)

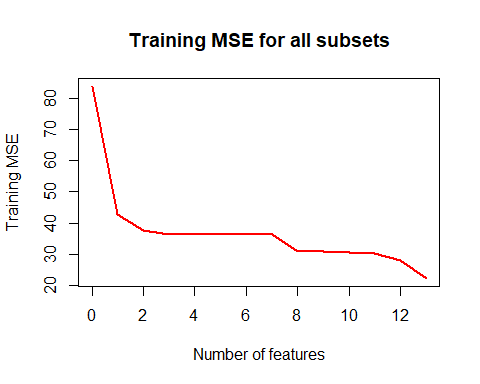
It is important to normalize all features when using L1 or L2 penalty for model estimation. Because, estimated parameters have a same unit as the corresponding variables and when we use these estimated parameters as the penalty factor, it will have high influence if the variable represents with very high values. As a result, prediction based on unnormalize features might be misleading.

### 

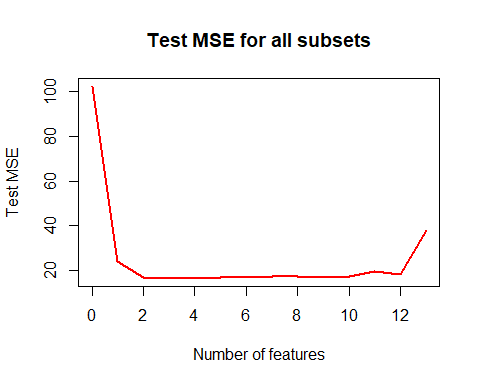
### (d)

#### (i)

dt\_train = dt[1:400,]  
dt\_test = dt[401:dim(dt)[1],]  
varn = c("CRIM", "ZN", "INDUS", "CHAS", "NOX", "RM",  
 "AGE", "DIS", "RAD", "TAX", "PTRATIO", "B" , "LSTAT")  
  
  
varinclude = c()  
varexlude = varn  
fi\_res = list()  
fi\_reg = list()  
k=2  
  
formu = "MEDV ~ 1"  
tem\_reg = lm(as.formula(formu), data = dt\_train)  
temp\_res = list(0, "intercept",   
 mean(tem\_reg$residuals^2),   
 mean((predict(tem\_reg, newdata = dt\_test)-dt\_test$MEDV)^2),  
 formu)  
  
  
fi\_reg[[1]] = tem\_reg  
fi\_res[[1]] = c(temp\_res)  
  
  
while (!is.null(varexlude) & length(varexlude)>0) {  
 tem\_reg = list()  
 temp\_res = list()  
   
   
 for (i in c(1:length(varexlude))){  
   
 if (is.null(varinclude)){  
 formu = paste("MEDV ~ 1", varexlude[i], sep = "+")   
 } else {  
 formu = paste("MEDV ~ 1", paste(varinclude, collapse = "+"), varexlude[i], sep = "+")   
 }  
   
 tem\_reg[[i]] = lm(as.formula(formu), data = dt\_train)  
 temp\_res[[i]] = list(length(varinclude)+1, varexlude[i],   
 mean(tem\_reg[[i]]$residuals^2),   
 mean((predict(tem\_reg[[i]], newdata = dt\_test)-dt\_test$MEDV)^2),  
 formu)  
   
 }  
   
 temp\_res = do.call(rbind.data.frame, temp\_res)  
 names(temp\_res) = c("nvar", "var\_include", "tr\_mse", "ts\_mse", "formula")  
 temp\_res$var\_include = as.character(temp\_res$var\_include)  
 temp\_res$formula = as.character(temp\_res$formula)  
 varinclude = c(varinclude, as.character(temp\_res[which.min(temp\_res$ts\_mse), 2]))  
 varexlude = varexlude[!(varexlude %in% varinclude)]  
   
 fi\_reg[[k]] = tem\_reg[[which.min(temp\_res$ts\_mse)]]  
 fi\_res[[k]] = c(temp\_res[which.min(temp\_res$ts\_mse),])  
  
 k=k+1  
}   
   
fi\_res = do.call(rbind.data.frame, fi\_res)  
names(fi\_res) = c("nvar", "var\_include", "tr\_mse", "ts\_mse", "formula")  
forward\_res = fi\_res  
  
plot(fi\_res$nvar, fi\_res$tr\_mse, type = "l", xlab = "Number of features", lwd = 2, col = 2,  
 ylab = "Training MSE")  
title("Training MSE for all subsets")



plot(fi\_res$nvar, fi\_res$ts\_mse, type = "l", xlab = "Number of features", lwd = 2, col = 2,  
 ylab = "Test MSE")  
title("Test MSE for all subsets")



print(fi\_res)

print(fi\_res[which.min(fi\_res$ts\_mse),])

aa = which.min(fi\_res$ts\_mse)  
res\_131 = fi\_reg[[aa]]  
len1 = length(res\_131$coefficients)  
print(paste("Forward Best Subset Model :", fi\_res[aa,]$formula, ": y^hat = ", round(res\_131$coefficients[1], 3), " + " ,  
 paste(round(res\_131$coefficients[2:len1], 3),   
 names(res\_131$coefficients)[2:len1], sep = " \* ", collapse = " + "),  
 " with training MSE = ", round(fi\_res[aa,3],3),   
 " and validation MSE = ", round(fi\_res[aa,4],3)))

## nvar var\_include tr\_mse ts\_mse  
## 2 0 intercept 83.80701 102.22659  
## 21 1 LSTAT 42.62158 23.92312  
## 3 2 PTRATIO 37.65518 16.76650  
## 4 3 CHAS 36.43344 16.55229  
## 5 4 ZN 36.42852 16.61204  
## 6 5 INDUS 36.41458 16.87186  
## 7 6 TAX 36.38863 17.15157  
## 8 7 NOX 36.37764 17.36296  
## 9 8 DIS 30.84108 17.31292  
## 10 9 CRIM 30.83273 16.81341  
## 11 10 AGE 30.51697 17.53081  
## 12 11 B 30.28257 19.59739  
## 13 12 RAD 28.17532 18.49887  
## 14 13 RM 22.30523 37.89378  
## formula  
## 2 MEDV ~ 1  
## 21 MEDV ~ 1+LSTAT  
## 3 MEDV ~ 1+LSTAT+PTRATIO  
## 4 MEDV ~ 1+LSTAT+PTRATIO+CHAS  
## 5 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN  
## 6 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS  
## 7 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX  
## 8 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX  
## 9 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS  
## 10 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS+CRIM  
## 11 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS+CRIM+AGE  
## 12 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS+CRIM+AGE+B  
## 13 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS+CRIM+AGE+B+RAD  
## 14 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS+CRIM+AGE+B+RAD+RM

## nvar var\_include tr\_mse ts\_mse formula  
## 4 3 CHAS 36.43344 16.55229 MEDV ~ 1+LSTAT+PTRATIO+CHAS

## [1] "Forward Best Subset Model : MEDV ~ 1+LSTAT+PTRATIO+CHAS : y^hat = 51.786 + -0.853 \* LSTAT + -1.011 \* PTRATIO + 3.923 \* CHAS with training MSE = 36.433 and validation MSE = 16.552"

Using forward search, based on validation MSE, the best model includes three variables (LSTAT, PTRATIO, CHAS).

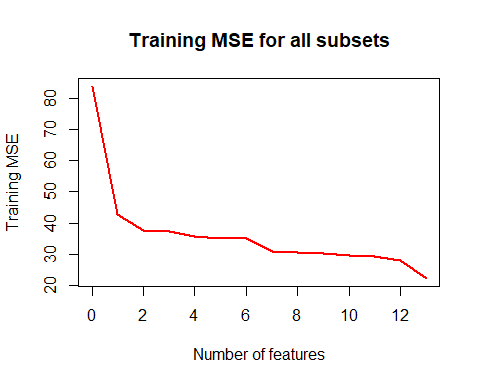
### 

### (d)

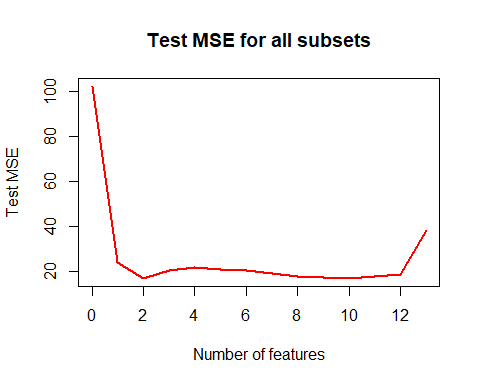
#### (ii)

dt\_train = dt[1:400,]  
dt\_test = dt[401:dim(dt)[1],]  
varn = c("CRIM", "ZN", "INDUS", "CHAS", "NOX", "RM",  
 "AGE", "DIS", "RAD", "TAX", "PTRATIO", "B" , "LSTAT")  
  
varinclude = varn  
varexlude = c()  
fi\_res = list()  
fi\_reg = list()  
k=2  
  
formu = paste("MEDV ~ 1", paste(varinclude, collapse = "+"), sep = "+")   
tem\_reg = lm(as.formula(formu), data = dt\_train)  
temp\_res = list(13, "None",   
 mean(tem\_reg$residuals^2),   
 mean((predict(tem\_reg, newdata = dt\_test)-dt\_test$MEDV)^2),  
 formu)  
  
fi\_reg[[1]] = tem\_reg  
fi\_res[[1]] = c(temp\_res)  
  
while (!is.null(varinclude) & length(varinclude)>0) {  
 tem\_reg = list()  
 temp\_res = list()  
   
 for (i in c(1:length(varinclude))){  
  
 if (is.null(varinclude) | length(varinclude)==1){  
 formu = "MEDV ~ 1"   
 } else {  
 formu = paste("MEDV ~ 1", paste(varinclude[-i], collapse = "+"), sep = "+")   
 }  
   
 tem\_reg[[i]] = lm(as.formula(formu), data = dt\_train)  
 temp\_res[[i]] = list(length(varinclude)-1, varinclude[i],   
 mean(tem\_reg[[i]]$residuals^2),   
 mean((predict(tem\_reg[[i]], newdata = dt\_test)-

dt\_test$MEDV)^2), formu)  
 }  
   
 temp\_res = do.call(rbind.data.frame, temp\_res)  
 names(temp\_res) = c("nvar", "var\_exclude", "tr\_mse", "ts\_mse", "formula")  
 temp\_res$var\_exclude = as.character(temp\_res$var\_exclude)  
 temp\_res$formula = as.character(temp\_res$formula)  
 varexlude = c(varexlude, as.character(temp\_res[which.min(temp\_res$ts\_mse), 2]))  
 varinclude = varinclude[!(varinclude %in% varexlude)]  
   
 fi\_reg[[k]] = tem\_reg[[which.min(temp\_res$ts\_mse)]]  
 fi\_res[[k]] = c(temp\_res[which.min(temp\_res$ts\_mse),])  
   
 k=k+1  
}   
  
fi\_res = do.call(rbind.data.frame, fi\_res)  
names(fi\_res) = c("nvar", "var\_include", "tr\_mse", "ts\_mse", "formula")  
backward\_res = fi\_res  
  
plot(fi\_res$nvar, fi\_res$tr\_mse, type = "l", xlab = "Number of features", lwd = 2, col = 2,  
 ylab = "Training MSE")  
title("Training MSE for all subsets")



plot(fi\_res$nvar, fi\_res$ts\_mse, type = "l", xlab = "Number of features", lwd = 2, col = 2,  
 ylab = "Test MSE")  
title("Test MSE for all subsets")



print(fi\_res)

print(fi\_res[which.min(fi\_res$ts\_mse),])

aa = which.min(fi\_res$ts\_mse)  
res\_131 = fi\_reg[[aa]]  
len1 = length(res\_131$coefficients)  
print(paste("For subset model ", fi\_res[aa,]$formula, ": y^hat = ", round(res\_131$coefficients[1], 3), " + " ,  
 paste(round(res\_131$coefficients[2:len1], 3),   
 names(res\_131$coefficients)[2:len1], sep = " \* ", collapse = " + "),  
 " with training MSE = ", round(fi\_res[aa,3],3),   
 " and validation MSE = ", round(fi\_res[aa,4],3)))

## nvar var\_include tr\_mse ts\_mse  
## 2 13 None 22.30523 37.89378  
## 21 12 RM 28.17532 18.49887  
## 3 11 TAX 29.25133 17.51549  
## 4 10 AGE 29.58767 16.93445  
## 5 9 ZN 30.43122 17.26423  
## 6 8 CRIM 30.72554 17.80309  
## 7 7 INDUS 30.90940 18.75269  
## 8 6 DIS 34.93004 20.21552  
## 9 5 NOX 34.94184 20.86509  
## 10 4 CHAS 35.82823 21.82590  
## 11 3 RAD 37.32106 20.31659  
## 12 2 B 37.65518 16.76650  
## 13 1 PTRATIO 42.62158 23.92312  
## 14 0 LSTAT 83.80701 102.22659  
## formula  
## 2 MEDV ~ 1+CRIM+ZN+INDUS+CHAS+NOX+RM+AGE+DIS+RAD+TAX+PTRATIO+B+LSTAT  
## 21 MEDV ~ 1+CRIM+ZN+INDUS+CHAS+NOX+AGE+DIS+RAD+TAX+PTRATIO+B+LSTAT  
## 3 MEDV ~ 1+CRIM+ZN+INDUS+CHAS+NOX+AGE+DIS+RAD+PTRATIO+B+LSTAT  
## 4 MEDV ~ 1+CRIM+ZN+INDUS+CHAS+NOX+DIS+RAD+PTRATIO+B+LSTAT  
## 5 MEDV ~ 1+CRIM+INDUS+CHAS+NOX+DIS+RAD+PTRATIO+B+LSTAT  
## 6 MEDV ~ 1+INDUS+CHAS+NOX+DIS+RAD+PTRATIO+B+LSTAT  
## 7 MEDV ~ 1+CHAS+NOX+DIS+RAD+PTRATIO+B+LSTAT  
## 8 MEDV ~ 1+CHAS+NOX+RAD+PTRATIO+B+LSTAT  
## 9 MEDV ~ 1+CHAS+RAD+PTRATIO+B+LSTAT  
## 10 MEDV ~ 1+RAD+PTRATIO+B+LSTAT  
## 11 MEDV ~ 1+PTRATIO+B+LSTAT  
## 12 MEDV ~ 1+PTRATIO+LSTAT  
## 13 MEDV ~ 1+LSTAT  
## 14 MEDV ~ 1

## nvar var\_include tr\_mse ts\_mse formula  
## 12 2 B 37.65518 16.7665 MEDV ~ 1+PTRATIO+LSTAT

## [1] "For subset model MEDV ~ 1+PTRATIO+LSTAT : y^hat = 52.79 + -1.05 \* PTRATIO + -0.85 \* LSTAT with training MSE = 37.655 and validation MSE = 16.766"

Using backward search, based on validation MSE, the best model includes only two variables (LSTAT, PTRATIO).

### 

### (d)

#### (iii)

print(forward\_res)

## nvar var\_include tr\_mse ts\_mse  
## 2 0 intercept 83.80701 102.22659  
## 21 1 LSTAT 42.62158 23.92312  
## 3 2 PTRATIO 37.65518 16.76650  
## 4 3 CHAS 36.43344 16.55229  
## 5 4 ZN 36.42852 16.61204  
## 6 5 INDUS 36.41458 16.87186  
## 7 6 TAX 36.38863 17.15157  
## 8 7 NOX 36.37764 17.36296  
## 9 8 DIS 30.84108 17.31292  
## 10 9 CRIM 30.83273 16.81341  
## 11 10 AGE 30.51697 17.53081  
## 12 11 B 30.28257 19.59739  
## 13 12 RAD 28.17532 18.49887  
## 14 13 RM 22.30523 37.89378  
## formula  
## 2 MEDV ~ 1  
## 21 MEDV ~ 1+LSTAT  
## 3 MEDV ~ 1+LSTAT+PTRATIO  
## 4 MEDV ~ 1+LSTAT+PTRATIO+CHAS  
## 5 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN  
## 6 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS  
## 7 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX  
## 8 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX  
## 9 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS  
## 10 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS+CRIM  
## 11 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS+CRIM+AGE  
## 12 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS+CRIM+AGE+B  
## 13 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS+CRIM+AGE+B+RAD  
## 14 MEDV ~ 1+LSTAT+PTRATIO+CHAS+ZN+INDUS+TAX+NOX+DIS+CRIM+AGE+B+RAD+RM

print(backward\_res)

## nvar var\_include tr\_mse ts\_mse  
## 2 13 None 22.30523 37.89378  
## 21 12 RM 28.17532 18.49887  
## 3 11 TAX 29.25133 17.51549  
## 4 10 AGE 29.58767 16.93445  
## 5 9 ZN 30.43122 17.26423  
## 6 8 CRIM 30.72554 17.80309  
## 7 7 INDUS 30.90940 18.75269  
## 8 6 DIS 34.93004 20.21552  
## 9 5 NOX 34.94184 20.86509  
## 10 4 CHAS 35.82823 21.82590  
## 11 3 RAD 37.32106 20.31659  
## 12 2 B 37.65518 16.76650  
## 13 1 PTRATIO 42.62158 23.92312  
## 14 0 LSTAT 83.80701 102.22659  
## formula  
## 2 MEDV ~ 1+CRIM+ZN+INDUS+CHAS+NOX+RM+AGE+DIS+RAD+TAX+PTRATIO+B+LSTAT  
## 21 MEDV ~ 1+CRIM+ZN+INDUS+CHAS+NOX+AGE+DIS+RAD+TAX+PTRATIO+B+LSTAT  
## 3 MEDV ~ 1+CRIM+ZN+INDUS+CHAS+NOX+AGE+DIS+RAD+PTRATIO+B+LSTAT  
## 4 MEDV ~ 1+CRIM+ZN+INDUS+CHAS+NOX+DIS+RAD+PTRATIO+B+LSTAT  
## 5 MEDV ~ 1+CRIM+INDUS+CHAS+NOX+DIS+RAD+PTRATIO+B+LSTAT  
## 6 MEDV ~ 1+INDUS+CHAS+NOX+DIS+RAD+PTRATIO+B+LSTAT  
## 7 MEDV ~ 1+CHAS+NOX+DIS+RAD+PTRATIO+B+LSTAT  
## 8 MEDV ~ 1+CHAS+NOX+RAD+PTRATIO+B+LSTAT  
## 9 MEDV ~ 1+CHAS+RAD+PTRATIO+B+LSTAT  
## 10 MEDV ~ 1+RAD+PTRATIO+B+LSTAT  
## 11 MEDV ~ 1+PTRATIO+B+LSTAT  
## 12 MEDV ~ 1+PTRATIO+LSTAT  
## 13 MEDV ~ 1+LSTAT  
## 14 MEDV ~ 1

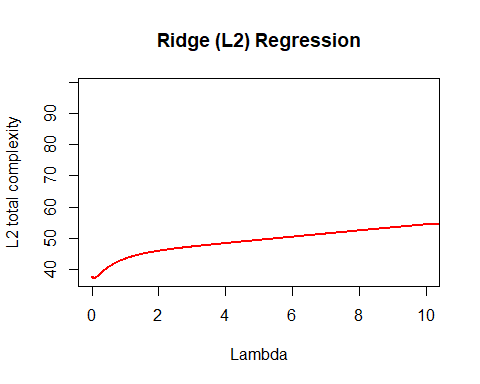
All lower order models are nested in the full model with all features. The best model found from backward search is nested within model found from forward search because forward model includes LSTAT, PTRATIO, CHAS variables and backward model includes only LSTAT, PTRATIO that can be found by considering CHAS coefficient equal zero.

### 

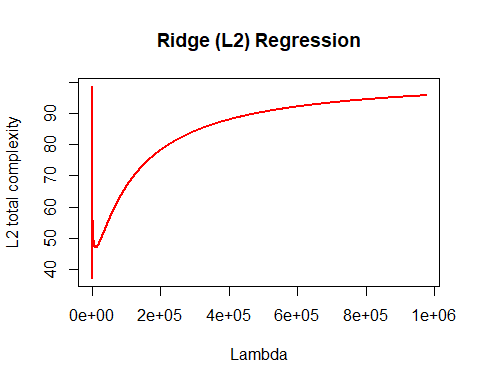
### (d)

#### (iv)

# Ridge  
  
dt\_train = dt[1:400,]  
dt\_test = dt[401:dim(dt)[1],]  
varn = c("CRIM", "ZN", "INDUS", "CHAS", "NOX", "RM",  
 "AGE", "DIS", "RAD", "TAX", "PTRATIO", "B" , "LSTAT")  
  
lambda = 0  
l2\_res = NULL  
  
loop = TRUE  
  
while (loop) {  
 a = glmnet(x = as.matrix(dt\_train[,varn]), y = dt\_train[,"MEDV"], standardize=FALSE,   
 alpha = 0, lambda = lambda)  
 pred = predict(a, as.matrix(dt\_test[,varn]))  
 l2 = mean((dt\_test$MEDV - pred)^2) + lambda \* sum(a$beta^2)  
 l2\_res = rbind(l2\_res, c(a$lambda, l2))  
 if (lambda <10){  
 lambda = lambda + .005  
 }else {  
 lambda = lambda + 100  
 }  
   
 if(sum(a$beta^2) < 1e-5 | lambda > 1500000) loop = FALSE  
}  
  
l2\_res = as.data.frame(l2\_res)  
plot(l2\_res$V1, l2\_res$V2, type = "l", xlab = "Lambda", lwd = 2, col = 2, xlim = c(0, 10),  
 ylab = "L2 total complexity")  
title("Ridge (L2) Regression")



plot(l2\_res$V1, l2\_res$V2, type = "l", xlab = "Lambda", lwd = 2, col = 2,  
 ylab = "L2 total complexity")  
title("Ridge (L2) Regression")



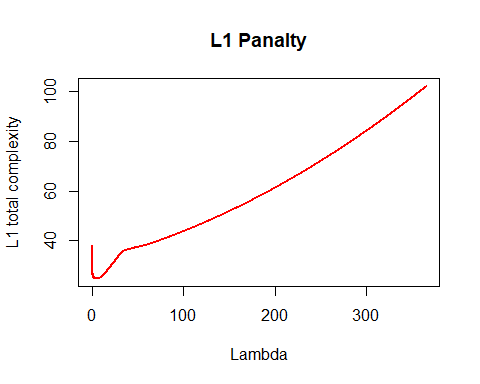
aa = which.min(l2\_res$V2)  
a = glmnet(x = as.matrix(dt\_train[,varn]), y = dt\_train[,"MEDV"], standardize=FALSE,   
 alpha = 0, lambda = l2\_res[aa,1])  
pred = predict(a, as.matrix(dt\_test[,varn]))  
l2 = mean((dt\_test$MEDV - pred)^2)  
  
print(paste("Model with lambda = ", round(l2\_res[aa,1],3), ": y^hat = ", round(a$a0, 3),   
 " + " ,  
 paste(round(a$beta, 3),   
 row.names(a$beta), sep = " \* ", collapse = " + "),  
 " with testing MSE = ", round(l2,3)))

## [1] "Model with lambda = 0.065 : y^hat = 21.105 + -0.184 \* CRIM + 0.047 \* ZN + 0.006 \* INDUS + 1.548 \* CHAS + -4.246 \* NOX + 4.849 \* RM + -0.005 \* AGE + -1.159 \* DIS + 0.46 \* RAD + -0.017 \* TAX + -0.696 \* PTRATIO + 0.002 \* B + -0.539 \* LSTAT with testing MSE = 34.081"

print(paste("Model with lambda = ", l2\_res[aa,1], " has lowest validation MSE, ", round(l2\_res[aa,2],3)))

## [1] "Model with lambda = 0.065 has lowest validation MSE, 37.091"

# LASSO  
  
lambda = 0  
l1\_res = NULL  
  
loop = TRUE  
  
while (loop) {  
 a = glmnet(x = as.matrix(dt\_train[,varn]), y = dt\_train[,"MEDV"], standardize=FALSE,   
 alpha = 1, lambda = lambda)  
 pred = predict(a, as.matrix(dt\_test[,varn]))  
 l1 = mean((dt\_test$MEDV - pred)^2) + lambda \* sum(abs(a$beta))  
 l1\_res = rbind(l1\_res, c(a$lambda, l1))  
 lambda = lambda + 0.05  
 if(sum(abs(a$beta)) < 1e-8) loop = FALSE  
}  
  
l1\_res = as.data.frame(l1\_res)  
plot(l1\_res$V1, l1\_res$V2, type = "l", xlab = "Lambda", lwd = 2, col = 2,  
 ylab = "L1 total complexity")  
title("L1 Panalty")



aa = which.min(l1\_res$V2)  
a = glmnet(x = as.matrix(dt\_train[,varn]), y = dt\_train[,"MEDV"], standardize=FALSE,   
 alpha = 1, lambda = l1\_res[aa,1])  
pred = predict(a, as.matrix(dt\_test[,varn]))  
mmse = mean((dt\_test$MEDV - pred)^2)  
  
print(paste("Model with lambda = ", round(l1\_res[aa,1],3), ": y^hat = ", round(a$a0, 3),   
 " + " ,  
 paste(round(a$beta, 5),   
 row.names(a$beta), sep = " \* ", collapse = " + "),  
 " with testing MSE = ", round(mmse,3)))

## [1] "Model with lambda = 5.1 : y^hat = 29.115 + 0 \* CRIM + 0.04001 \* ZN + 0 \* INDUS + 0 \* CHAS + 0 \* NOX + 0 \* RM + 0.05182 \* AGE + 0 \* DIS + 0 \* RAD + -0.00581 \* TAX + 0 \* PTRATIO + 0.00771 \* B + -0.85077 \* LSTAT with testing MSE = 19.898"

print(paste("Model with lambda = ", round(l1\_res[aa,1], 3), " has lowest validation MSE, ", round(mmse,3)))

## [1] "Model with lambda = 5.1 has lowest validation MSE, 19.898"

Using Ridge regression best model found for which includes all variables with validation . On the other hand, best LASSO model found fro with validation which includes only ZN, AGE, TAX, B and LSTAT and it is mach lower that Ridge regression. However, forward search model includes only LSTAT, PTRATIO, CHAS variables with validation and it is the lowest MSE among forward, backward, Ridge and LASSO models.