Lab 5

Attila Lazar

11.11.2020

data

We load the dataset and split it in train and test sets

```
load("data/dat.RData")
data <- d
set.seed(1234)
n <- nrow(data)
train <- sample(1:n, round(n*2/3))
test <- (1:n) [-train]
```

First We establish a baseline by building a RS model and calculate MSE with the test data

```
rtmean <- function(x,trim = 0) {
  x <- sort(x)
  v <- x[1:floor(length(x)*(1-trim))]
  mean(v)
}

mse <- function(y.true,y.pred, trim=0){
  return(rtmean((y.true - y.pred)^2, trim =trim))
}


model1 <- lm(y~., data, subset=train)

rtmean((data[test, 'y'] - predict(model1, data[test,]))^2)
```

```
## Warning in predict.lm(model1, data[test, ]): prediction from a rank-
## deficient fit may be misleading
```

```
## [1] 8.781102
```

```
mse.base <- rtmean((data[test, 'y'] - predict(model1, data[test,]))^2, trim=0.1)
```

```
## Warning in predict.lm(model1, data[test, ]): prediction from a rank-
## deficient fit may be misleading
```

```
mse.base
```

```
## [1] 0.6878118
```

1. Ridge Regression

a)

We estimate the Ridge model with lambdas from 1 to 100.

```r
library(MASS)
lambda <- seq(1, 100, 0.1)
model2 <- lm.ridge(y~., data, subset=train, lambda = lambda)
```

The minimum GVC:

```r
min(model2$GCV)
```

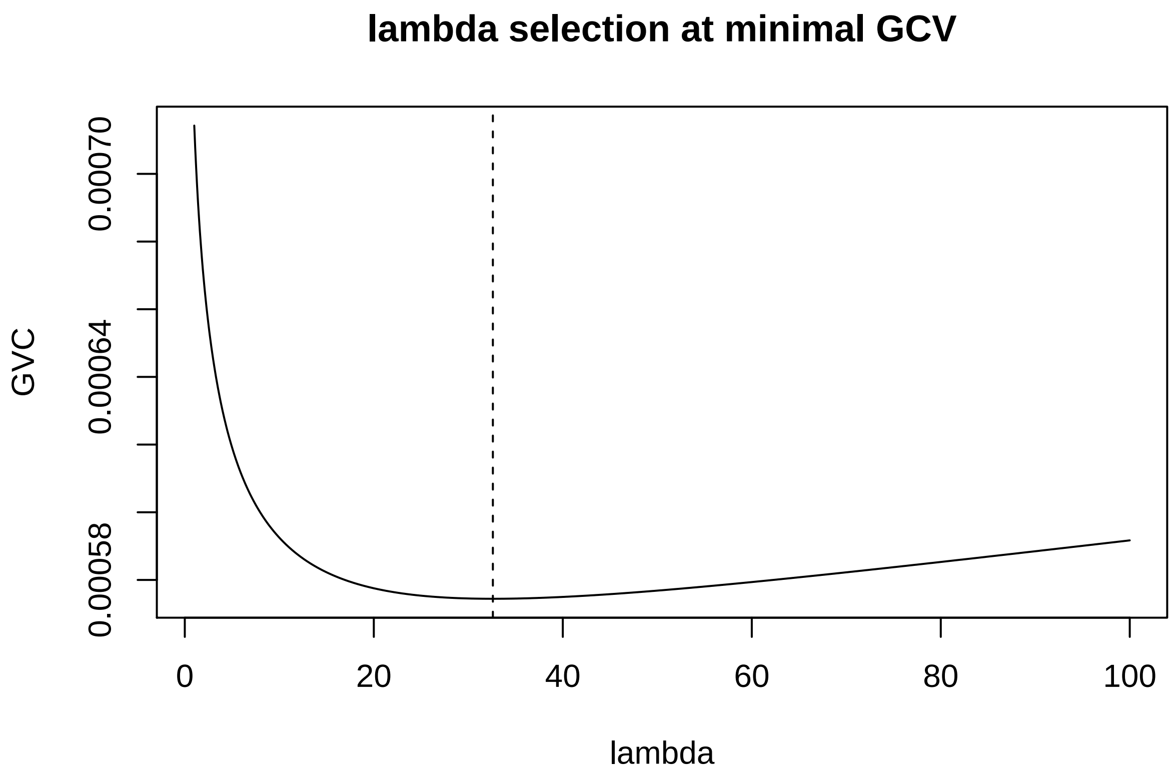
```
## [1] 0.000574434
```

with the lambda parameter:

```r
model2$lambda[which.min(model2$GCV)]
```

```
## [1] 32.6
```

```r
lambda.opt <- model2$lambda[which.min(model2$GCV)]
plot(lambda, model2$GCV, type='l', main="lambda selection at minimal GCV", ylab="GVC")
abline(v=lambda.opt, lty=2)
```



b)

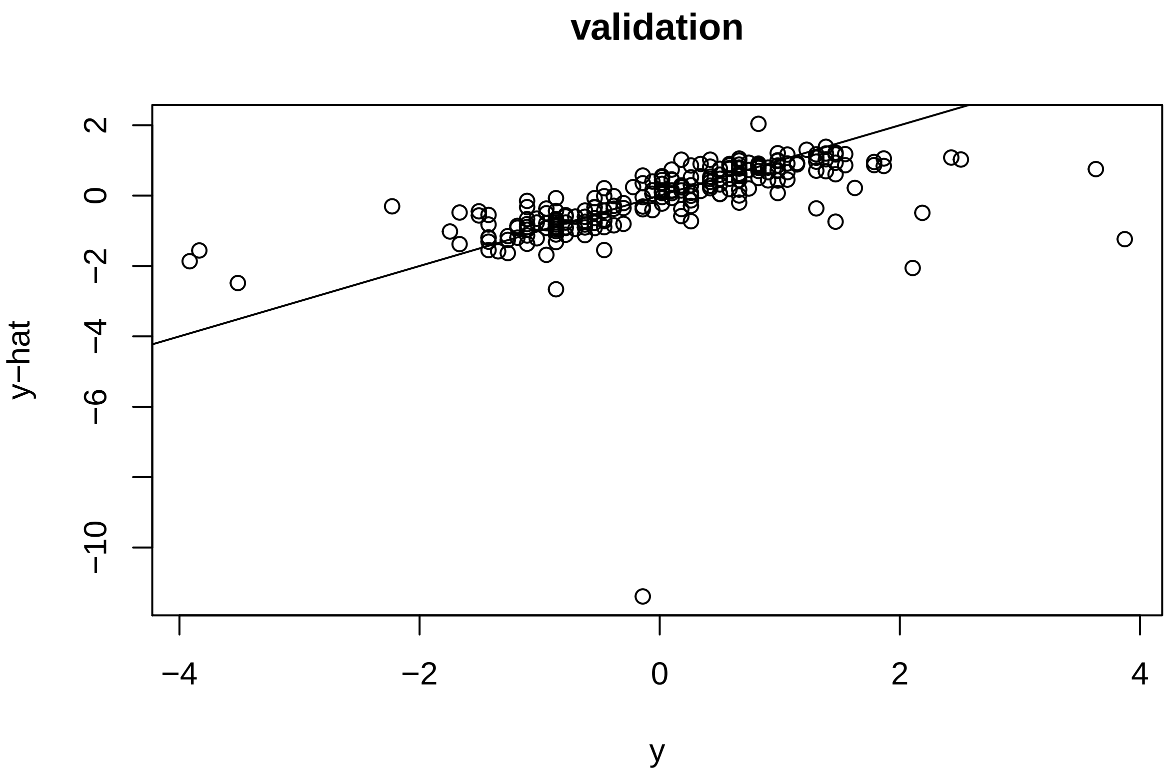
We again estimate the Ridge model with the optimal lambda parameter

```r
model2.sel <- lm.ridge(y~., data, subset=train, lambda = lambda.opt)
coef.model2.sel <- coef(model2.sel)
```

c)

We use the model from b) to predict values from the test dataset

```r
y.pred <- as.matrix(cbind(rep(1,length(test)), data[test,-1])) %*% coef.model2.sel
plot(data[test,'y'], y.pred, xlab='y' ,ylab='y-hat', main="validation")
abline(c(0,1))
```



There is one outlier, We compute the trimmed MSE

```r
mse1 <- mse(data[test, 'y'], y.pred, trim=0.1)
sprintf("%f < %f", mse1, mse.base)
```

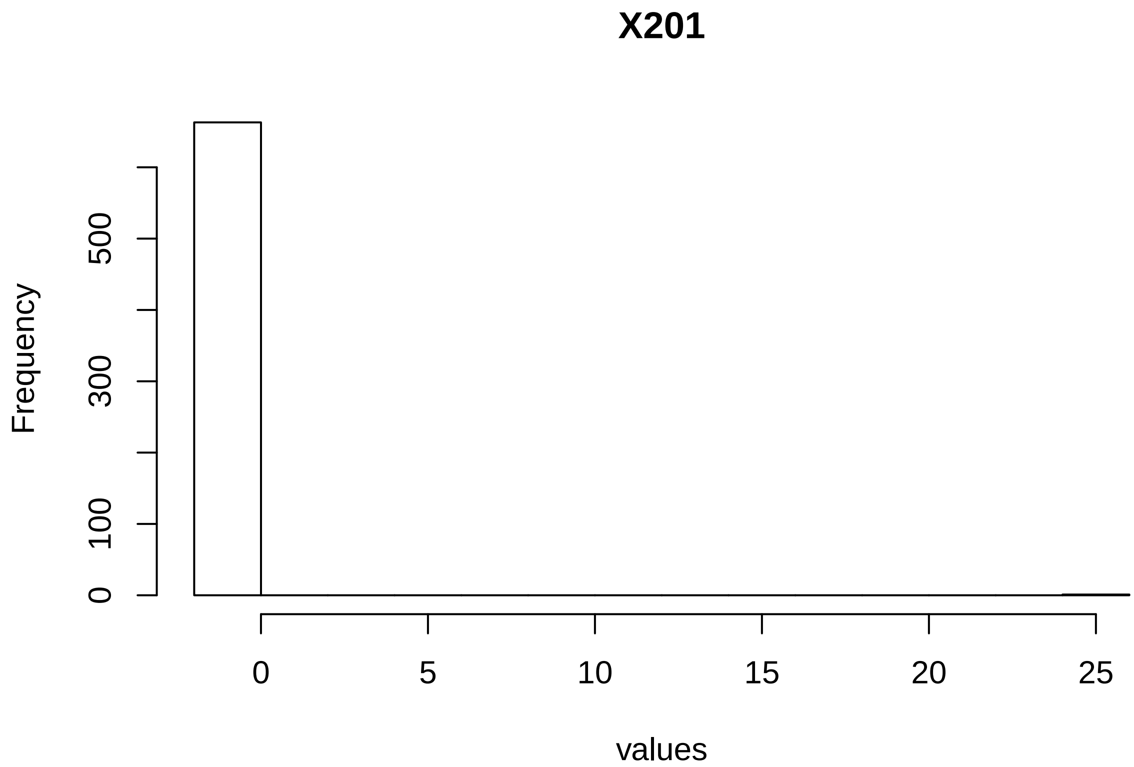
```
## [1] "0.132832 < 0.687812"
```

We see that the MSE is significantly better then our baseline

d)

We use MAD to find variables very little variance

```r
vars <-  which(apply(d,2,mad) < 0.001)
for (x in seq_along(vars)) {
  hist(data[,vars[x]], main=names(vars)[x], xlab="values")
}
```



Since this variable is nearly always 0 we exclude this variable from the data. Then we again estimate the Ridge model using our cleaned dataset

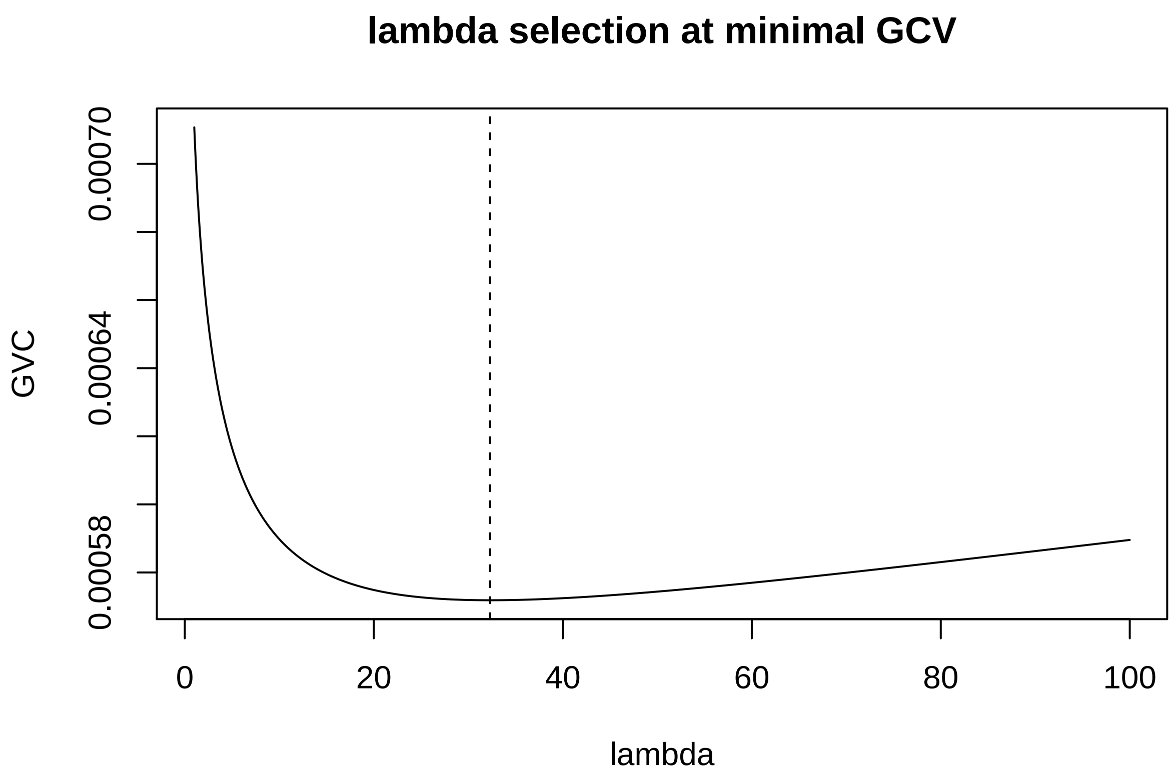
```
data.clean <- data[,(!names(data) %in% c('X201'))]
model3 <- lm.ridge(y~., data.clean, subset=train, lambda = lambda)
```

Now our optimal lambda ist

```
lambda.opt <- model3$lambda[which.min(model3$GCV)]
lambda.opt
```

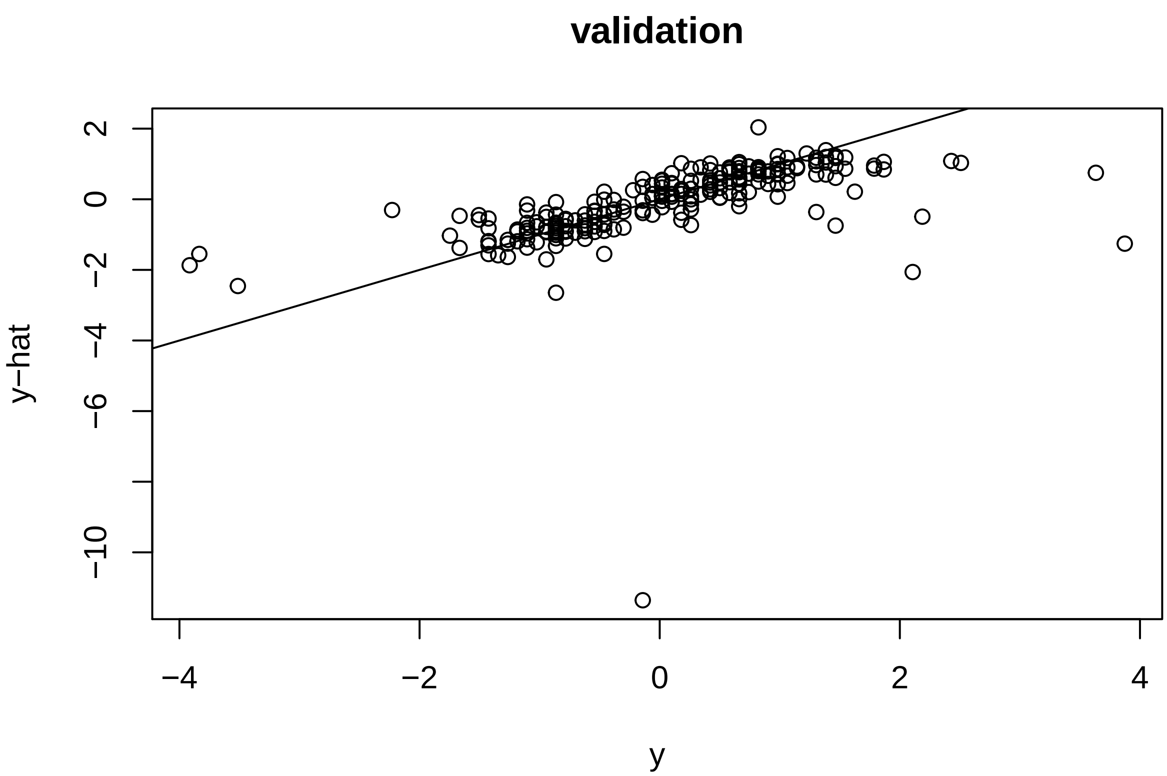
```
## [1] 32.3
```

```
plot(lambda, model3$GCV, type='l', main="lambda selection at minimal GCV", ylab="GVC")
abline(v=lambda.opt, lty=2)
```



We repeat the prediction on the test data

```
model3.sel <- lm.ridge(y~., data.clean, subset=train, lambda = lambda.opt)
coef.model3.sel <- coef(model3.sel)
y.pred <- as.matrix(cbind(rep(1,length(test)), data.clean[test,-1])) %*% coef.model3.sel
plot(data.clean[test,'y'], y.pred, xlab='y' ,ylab='y-hat', main="validation")
abline(c(0,1))
```



As we see the results did not improve a lot, we still have a big outlier which spoils the results

```
mse3 <- mse(data.clean[test, 'y'], y.pred, trim=0.1)
sprintf("%f > %f", mse3, mse1)
```

```
## [1] "0.133000 > 0.132832"
```

In fact out calculated MSE is slightly worse then before cleaning the dataset

2. Lasso Regression

a)

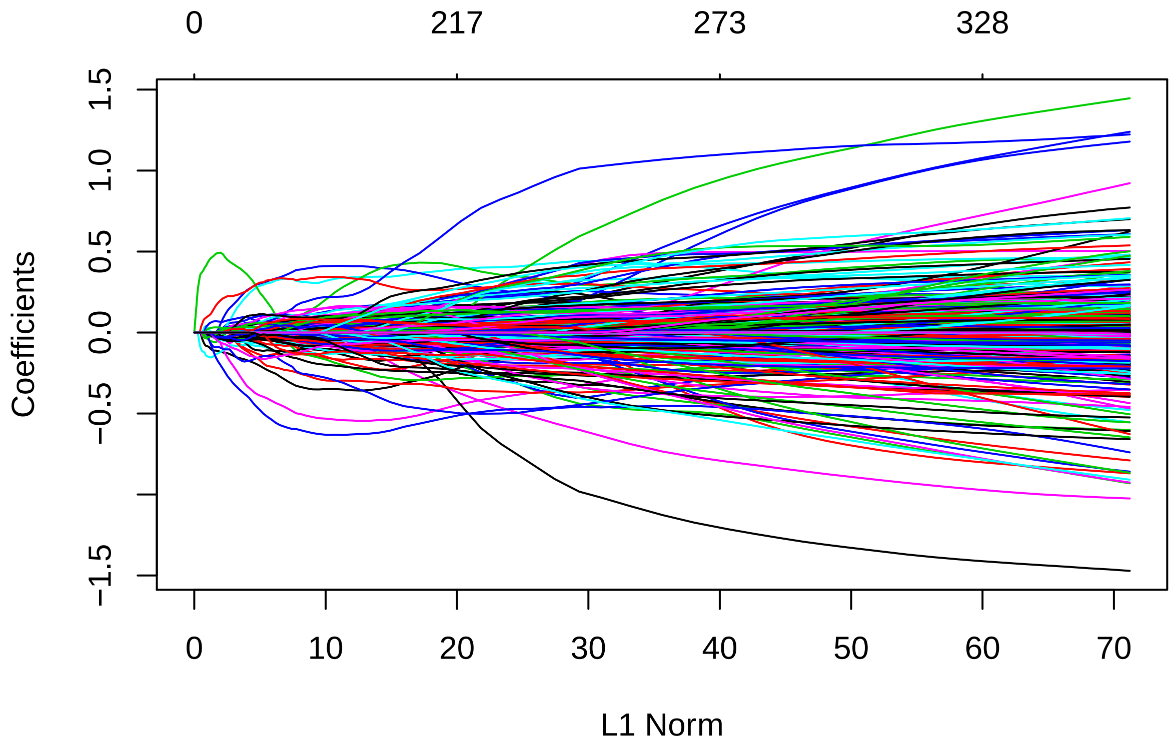
```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```

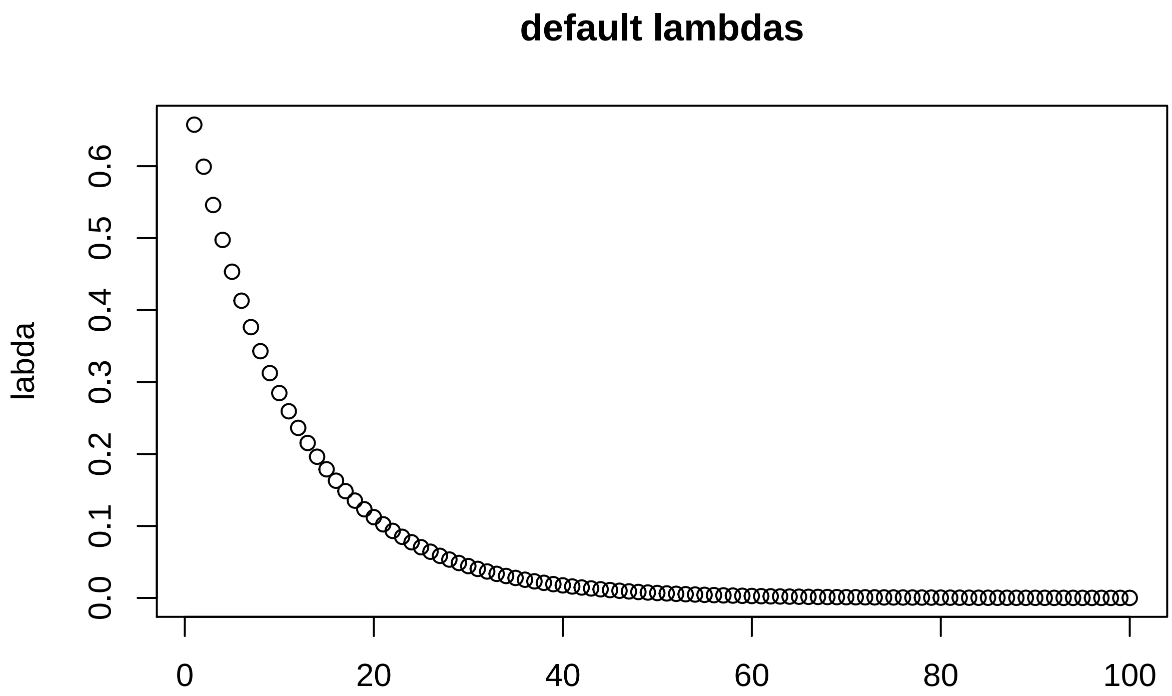
```
model4 <- glmnet(as.matrix(data.clean[train, -1]), data.clean[train, 1])
plot(model4)
```



The plot shows the shrinkage of the regression parameters. From left to right the parameters get smaller, some reaching 0.

We plot the choosen default lambdas. There are in the range of 0-1. The parameter "alpha" lets us combile the ridge and lasso methods. default is set to 1 which means we are using "pure" lasso regression.

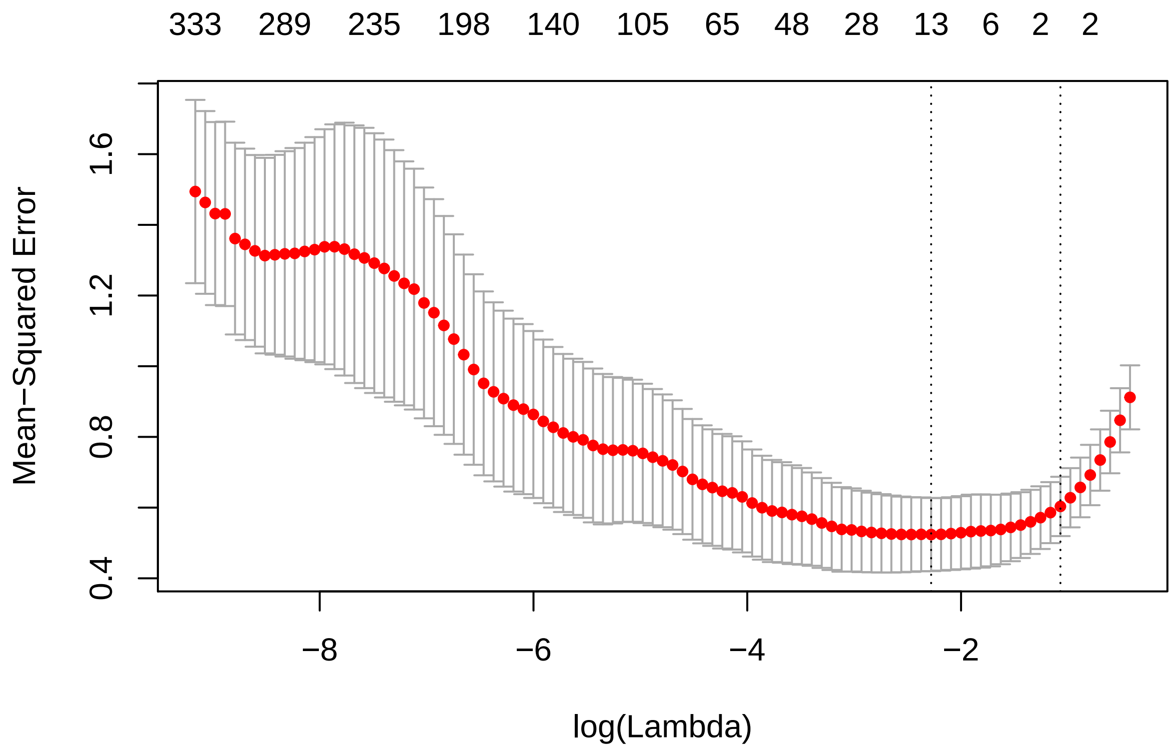
```
plot(model4$lambda, main="default lambdas", xlab="", ylab="labda")
```



b)

We use CV to obtain the optimal choice of lambda

```
model4.cv <- cv.glmnet(as.matrix(data.clean[train, -1]), data.clean[train, 1])
plot(model4.cv)
```



We see that the MSE obtained with CV decreases steadily and then increases again.

We get minimal MSE value with lambda

```
model4.cv$lambda.min
```

```
## [1] 0.1023119
```

with the coeffients

```
coef(model4.cv, s="lambda.min") [which(coef(model4.cv, s="lambda.min") != 0),]
```

```
##   (Intercept)           X17           X32           X58           X64
## -0.0215714112  0.0368543991 -0.0776438760 -0.0027086541 -0.0080616902
##           X67          X101          X104          X112          X135
##  0.4049361934 -0.1323710496  0.0011133525  0.0187835226  0.0881929137
##          X191          X232          X270          X282
## -0.0215172942  0.0001447727 -0.0357201570  0.0037778084
```

Probably better to select a lamda where the model has fewer non-zero coefficients

```
model4.cv$lambda.1se
```

```
## [1] 0.3429089
```

with coeffients

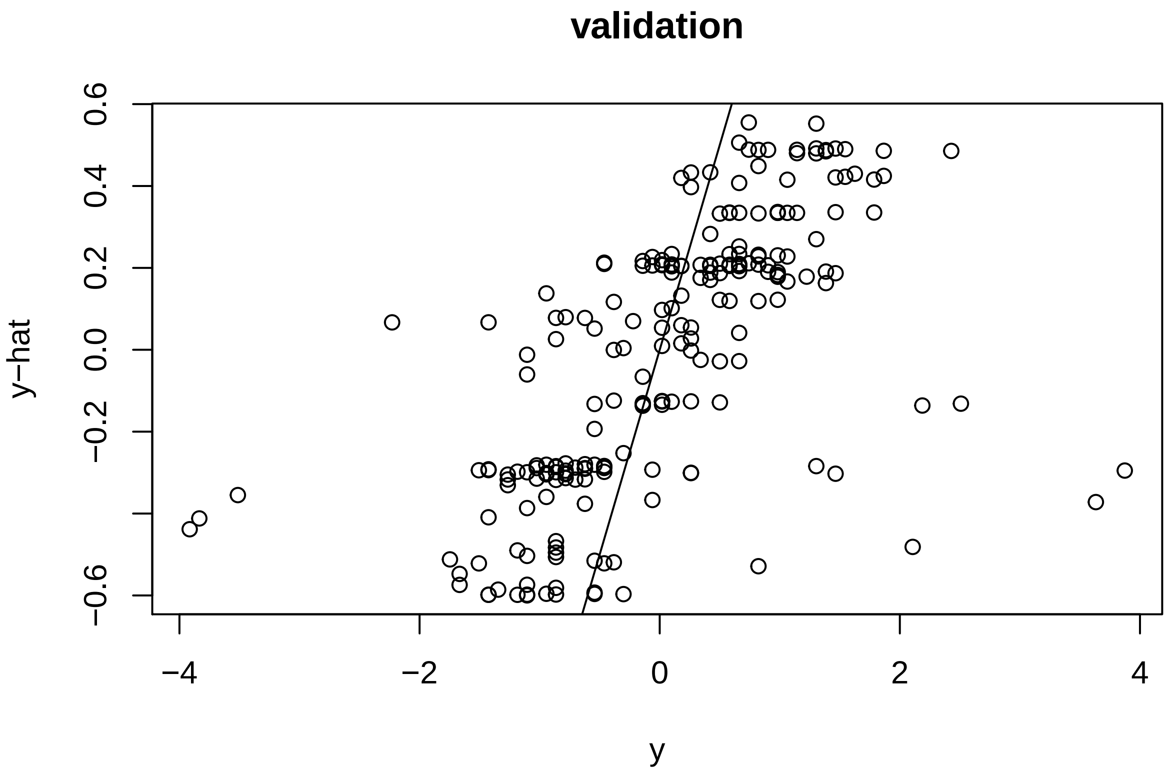
```
coef(model4.cv, s="lambda.1se") [which(coef(model4.cv, s="lambda.1se") != 0),]
```

```
## (Intercept)         X67        X101
## -0.02273817  0.29782610 -0.03062193
```

c)

We use the obtained optimal lambda parameter to predinct test data and calculate the MSE

```r
y.pred <- predict(model4.cv, newx=as.matrix(data.clean[test,-1]), s="lambda.1se")
plot(data.clean[test,'y'], y.pred, xlab='y' ,ylab='y-hat', main="validation")
abline(c(0,1))
```



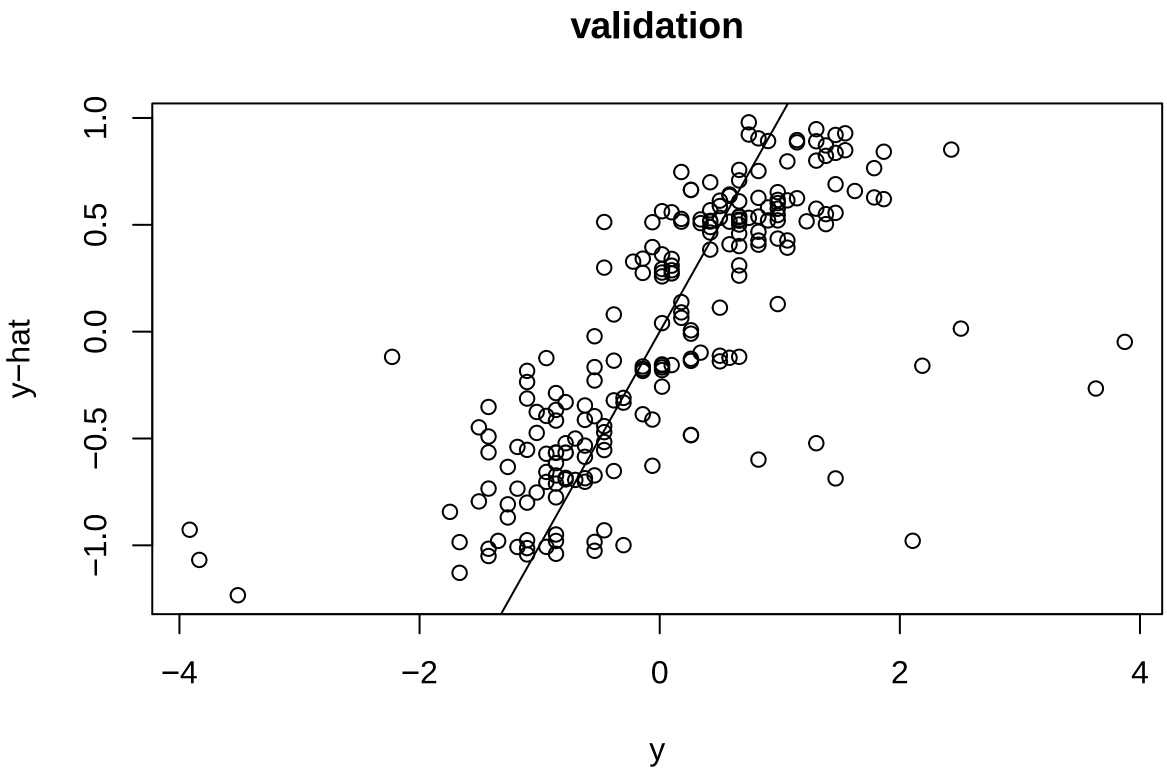

```r
mse4.1 <- mse(data.clean[test, 'y'], y.pred, trim=0.1)
sprintf("%f > %f", mse4.1, mse.base)
```

```
## [1] "0.320809 > 0.687812"
```

The results does not look good, the MSE is worse than our baseline

We try again, this time using the lamdba.min parameter

```r
y.pred <- predict(model4.cv, newx=as.matrix(data.clean[test,-1]), s="lambda.min")
plot(data.clean[test,'y'], y.pred, xlab='y' ,ylab='y-hat', main="validation")
abline(c(0,1))
```




```r
mse4.2 <- mse(data.clean[test, 'y'], y.pred, trim=0.1)
sprintf("%f > %f", mse4.2, mse3)
```

```
## [1] "0.166388 > 0.133000"
```

This model gives us better results but still worse than our model with ridge regression