Lasso estimation in multiple linear regression

Universitat Politècnica de Catalunya

Andreu Meca, Jan Leyva, Geraldo Gariza, Victor Duque

2/28/2021

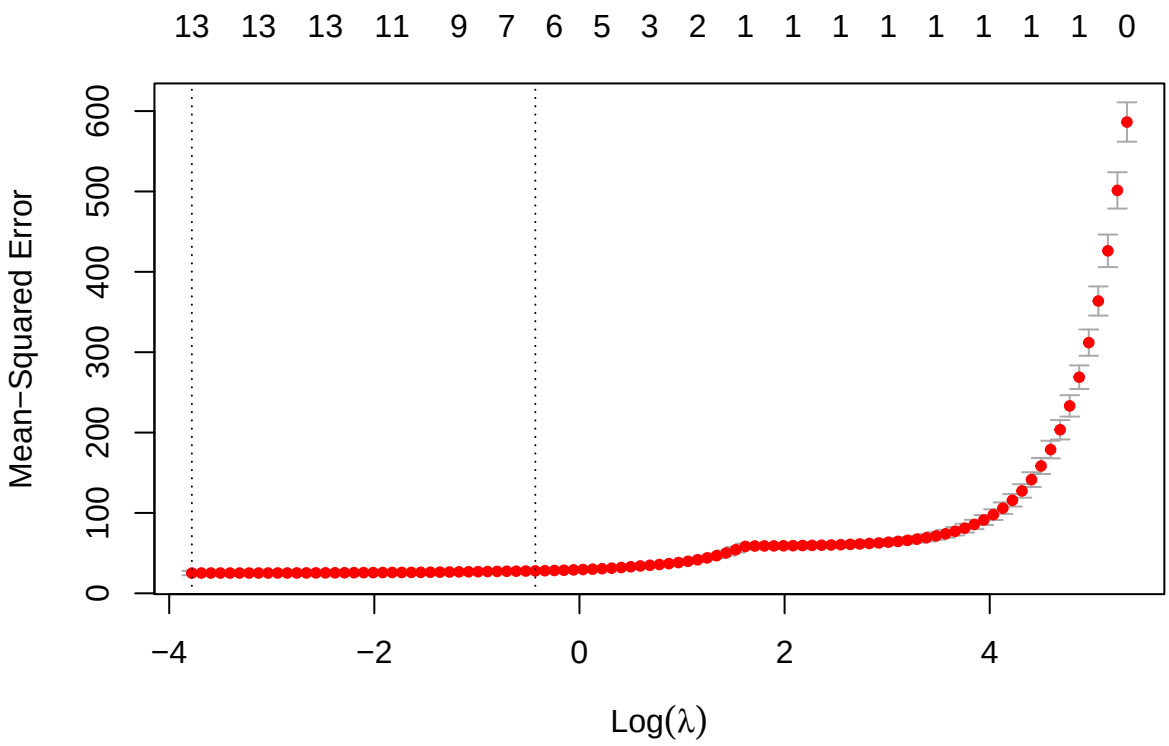
---

1. Lasso for the Boston Housing data

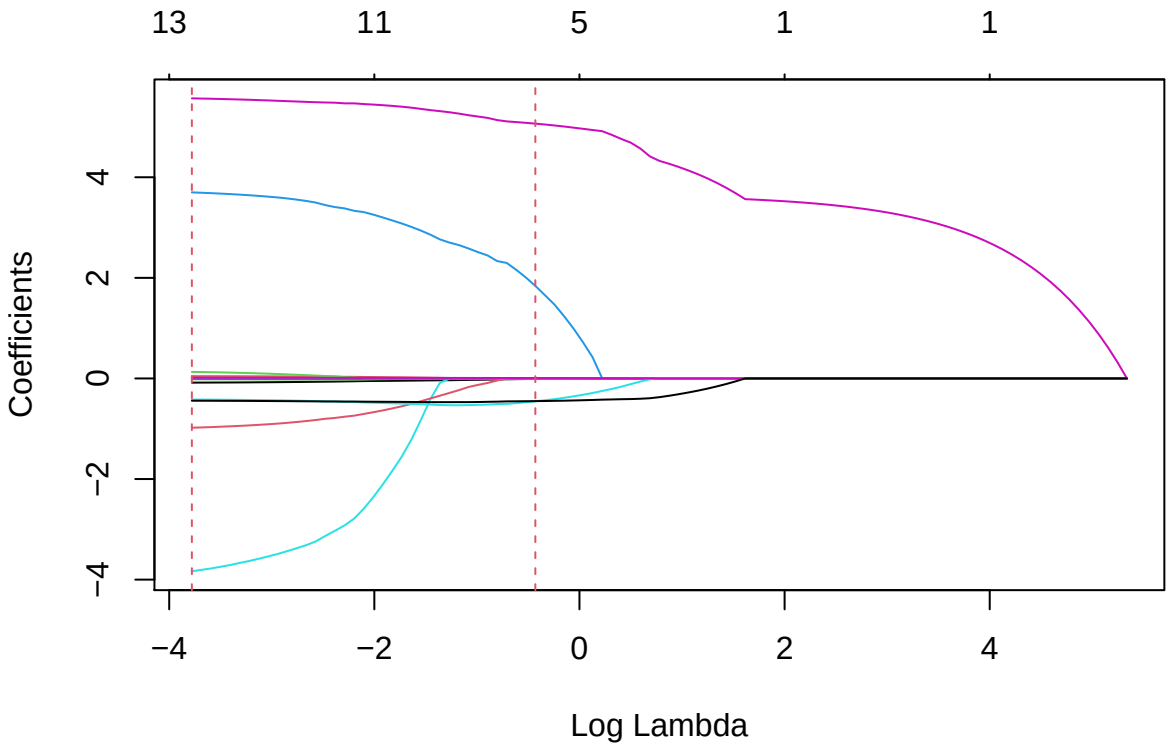
1.1. For the Boston House-price corrected dataset use Lasso estimation (in glmnet) to t the regression model where the response is CMEDV (the corrected version of MEDV) and the explanatory variables are the remaining 13 variables in the previous list. Try to provide an interpretation to the estimated model.

```
lasso.1 <- glmnet(X, Y, standardize=TRUE, intercept=FALSE)
cv.lasso.1 <- cv.glmnet(X,Y, standardize=TRUE, intercept=FALSE,nfolds=10)

plot(cv.lasso.1)
```




```
plot(lasso.1, xvar="lambda")
abline(v=log(cv.lasso.1$lambda.min),col=2,lty=2)
abline(v=log(cv.lasso.1$lambda.1se),col=2,lty=2)
```



First graphic:

As it can be seen, the log(λ) minimum is near -4 (first vertical doted line). Moreover the 1se (one standard error) λ has a log(λ) almos 0 (negative). Hence there is a broad number of able lok(λ). Using the 1se log(λ) bestow a difference of 13 - 6 = 7 coefficients. Hence using the almost 0 log(λ) allows for a easier to interpret model while being significantly equivalent.

Second graphics:

This graphic allows to determine which variables enables for a better model keeping the prediction capacity while being elastic enough. In the minimum log(λ) there are three non 0 coeficients (CHAS, NOX, RM). As the log(λ) value increases until matching the 1se log(λ) one of the coefficients (the one with negative value, NOX) goes to 0. Therefore there are only 2 coefficients left with a non 0 value. It can be observed that soon after the 1se, one of the coefficients (CHAS) goes to 0 too. This can be a signal that this coefficient is relevant enough for the correct fitting of the model. So maybe without it, and having only the RM to model, it would deviate too much. The coefficient of greater value (RM) changes very little througout the different log(λ) values. From a less formal perspective we can see a logic in this correlation since RM (number of rooms) is a trusty indicator of a house price. While other factors such as the river proximity (CHAS) or the nitrogen monoxide on air (NOX) do not seem that relevant.

```
print(coef(lasso.1,s=cv.lasso.1$lambda.min))
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                        1
```

```
## (Intercept)  .
## CRIM        -0.083193877
## ZN           0.045611958
## INDUS       -0.017186834
## CHAS         3.699740772
## NOX         -3.833098974
## RM           5.568764288
## AGE         -0.005770689
## DIS         -0.979937196
## RAD          0.130455377
## TAX         -0.007293136
## PTRATIO     -0.419429691
## B            0.013504092
## LSTAT       -0.442899630
```

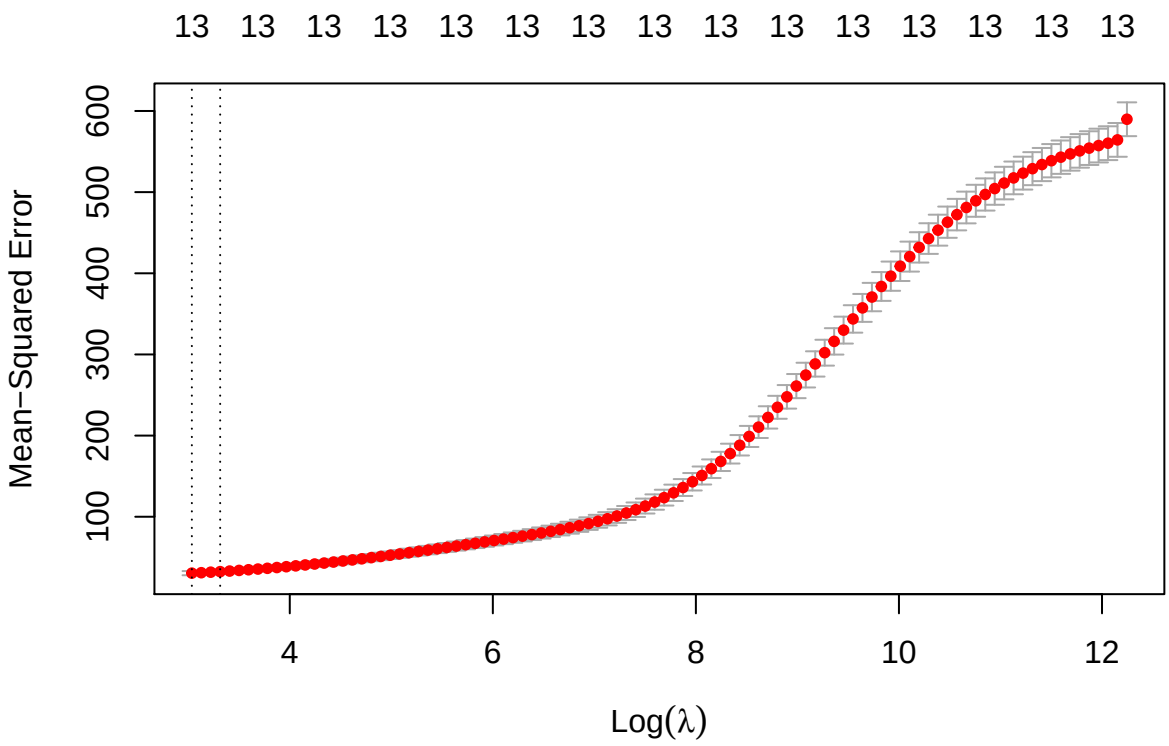
```
print(coef(lasso.1,s=cv.lasso.1$lambda.1se))
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  .
## CRIM        -0.008620712
## ZN           .
## INDUS        .
## CHAS         1.840673229
## NOX          .
## RM           5.065578638
## AGE          .
## DIS          .
## RAD          .
## TAX          .
## PTRATIO     -0.453948326
## B            0.007862924
## LSTAT       -0.450721912
```

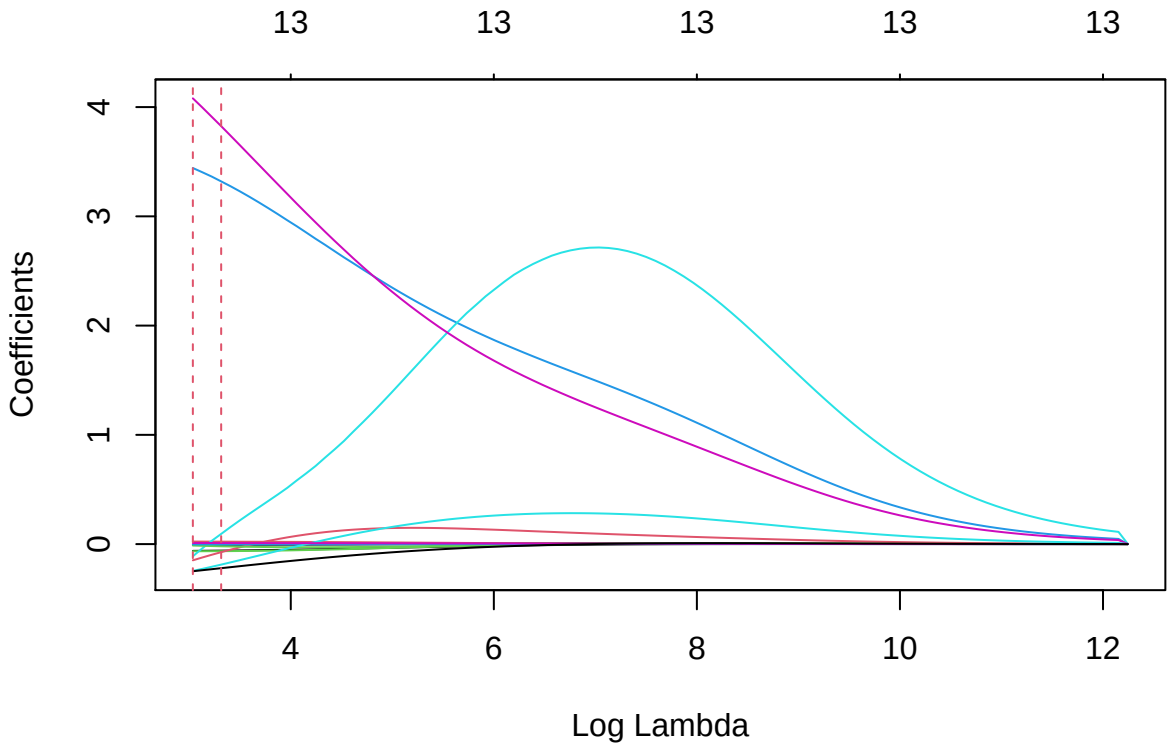
1.2. Use glmnet to fit the previous model using ridge regression. Compare the 10-fold cross validation results from function cv.glmnet with those you obtained in the previous practice with your own functions.

```
ridge.1 <- glmnet(X, Y, standardize=TRUE, intercept=FALSE, alpha = 0)
# specify alpha = 0 for use Ridge regression not LASSO
cv.ridge.1 <- cv.glmnet(X,Y, standardize=TRUE, intercept=FALSE, nfolds=10, alpha = 0)

plot(cv.ridge.1)
```




```
plot(ridge.1,xvar="lambda")
abline(v=log(cv.ridge.1$lambda.min),col=2,lty=2)
abline(v=log(cv.ridge.1$lambda.1se),col=2,lty=2)
```




```
print(coef(ridge.1,s=cv.ridge.1$lambda.min))
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  .
## CRIM        -0.061969763
## ZN           0.023101849
## INDUS       -0.065339831
## CHAS         3.442213784
## NOX         -0.111578552
## RM           4.079909748
## AGE         -0.006025971
## DIS         -0.148493382
## RAD         -0.013204683
## TAX         -0.002411107
## PTRATIO     -0.245417203
## B            0.010108453
## LSTAT       -0.248394565
```

```
print(coef(ridge.1,s=cv.ridge.1$lambda.1se))
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  .
```

```
## CRIM        -0.059521477
## ZN           0.022339737
## INDUS       -0.064453635
## CHAS         3.319762923
## NOX          0.093303219
## RM           3.825441781
## AGE         -0.005879167
## DIS         -0.073602902
## RAD         -0.019901814
## TAX         -0.002287582
## PTRATIO     -0.187715408
## B            0.009768768
## LSTAT       -0.220381692
```

Regression with our previous function:

```
lambda.max <- 1e5
n.lambdas <- 25
lambda.v <- exp(seq(0,log(lambda.max+1),length=n.lambdas))-1

ridge_regression_k_fold_CV(X, Y, lambda.v, k = 10)[[5]]
```

```
## [1] "Value of best lambda: 5.81293"
```

We have obtained different Betas values with glmnet() and our function. Also the optimum lambda is in different position, we reach the minimum betas with the fifth lambda, however with glmnet() it reach at third position. Even though, the values of beta are different we see the same sign between both approach.

To conclude, the difference in values could be because of the consistency of the glmnet packet.

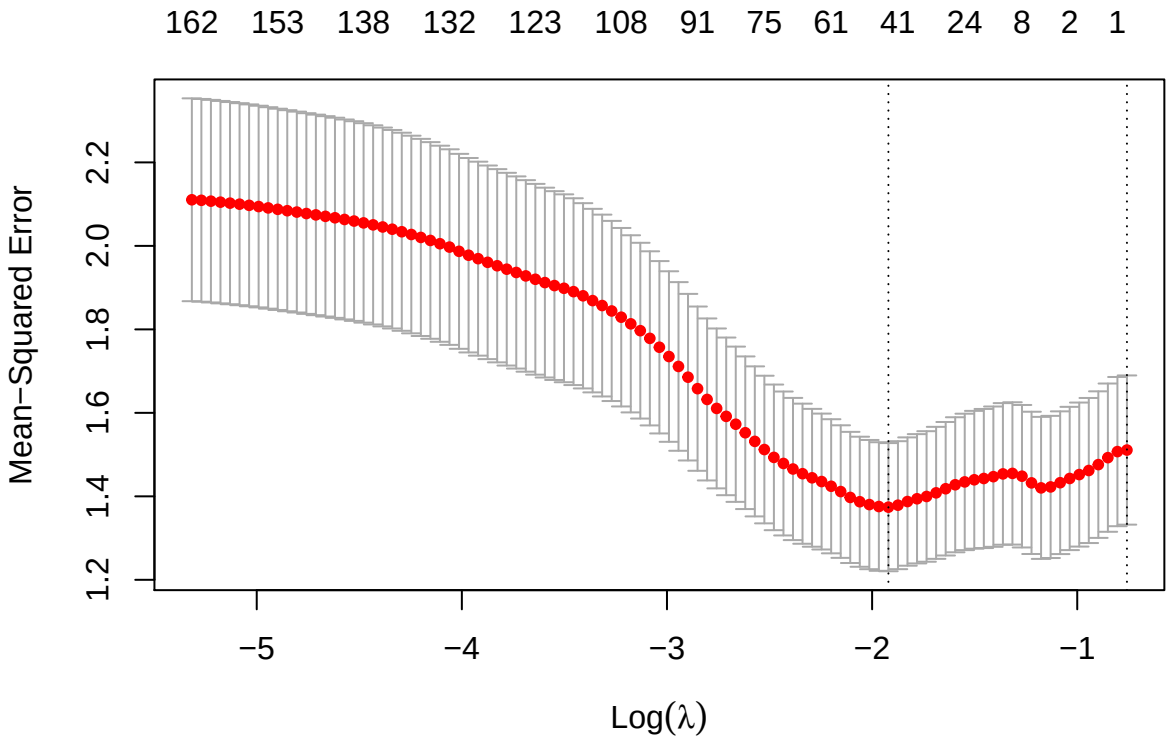
2. A regression model with "p » n"

2.1. Use glmnet and cv.glmnet to obtain the Lasso estimation for regressing log.surv against expr. How many coefficient different from zero are in the Lasso estimator? Illustrate the result with two graphics.

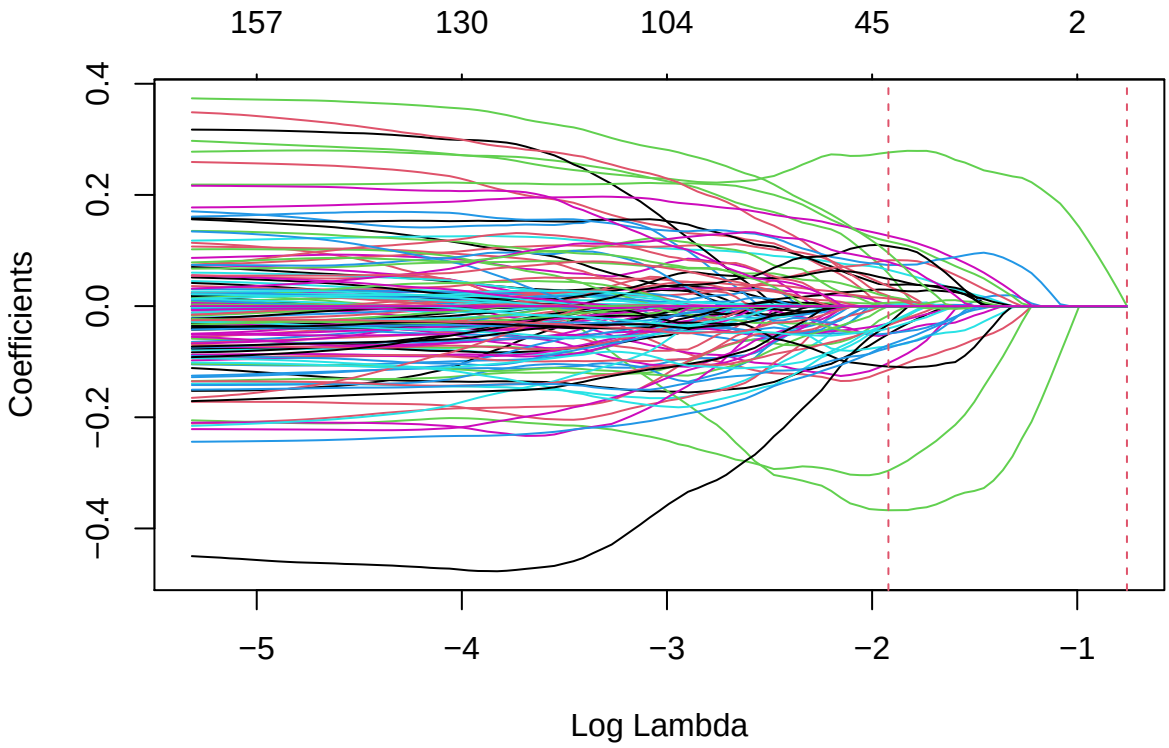
```
express <- read.csv("journal.pbio.0020108.sd012.CSV",header=FALSE)
surv <- read.csv("journal.pbio.0020108.sd013.CSV",header=FALSE)
death <- (surv[,2]==1)
log.surv <- log(surv[death,1]+.05)
expr <- as.matrix(t(express[,death]))

lasso.3 <- glmnet(expr, log.surv, standardize=TRUE, intercept=TRUE)
cv.lasso.3 <- cv.glmnet(expr, log.surv, standardize=TRUE, intercept=TRUE, nfolds = 100)

plot(cv.lasso.3)
```




```
plot(lasso.3, xvar="lambda")
abline(v=log(cv.lasso.3$lambda.min),col=2,lty=2)
abline(v=log(cv.lasso.3$lambda.1se),col=2,lty=2)
```




```
coef(lasso.3, s=cv.lasso.3$lambda.min)[which(coef(lasso.3, s=cv.lasso.3$lambda.min) != 0)]
```

```
##  [1]  0.1072988339 -0.1149960583 -0.0502218070  0.0585753811 -0.1024611181
##  [6] -0.0170115375 -0.0497527144 -0.3672843897 -0.0725618571 -0.0493116919
## [11]  0.0651681576 -0.0312602856 -0.2960888334  0.0384604555  0.0356776877
## [16]  0.0006386386  0.0062755321  0.0026650428  0.2764135506  0.0823479138
## [21]  0.0056830630  0.0012942002  0.1165380506  0.0294877648  0.0183466773
## [26]  0.0171799638  0.1265015285  0.0138736920  0.0737819031  0.0490078599
## [31]  0.1079156359 -0.0338022957 -0.0016548119 -0.0344033261 -0.0312115436
## [36]  0.0794233967 -0.0322655057  0.0954413957 -0.1087980064 -0.0752829473
## [41] -0.0403127953 -0.0303447468 -0.0701284589
```

```
coef(lasso.3, s=cv.lasso.3$lambda.1se)[which(coef(lasso.3, s=cv.lasso.3$lambda.1se) != 0)]
```

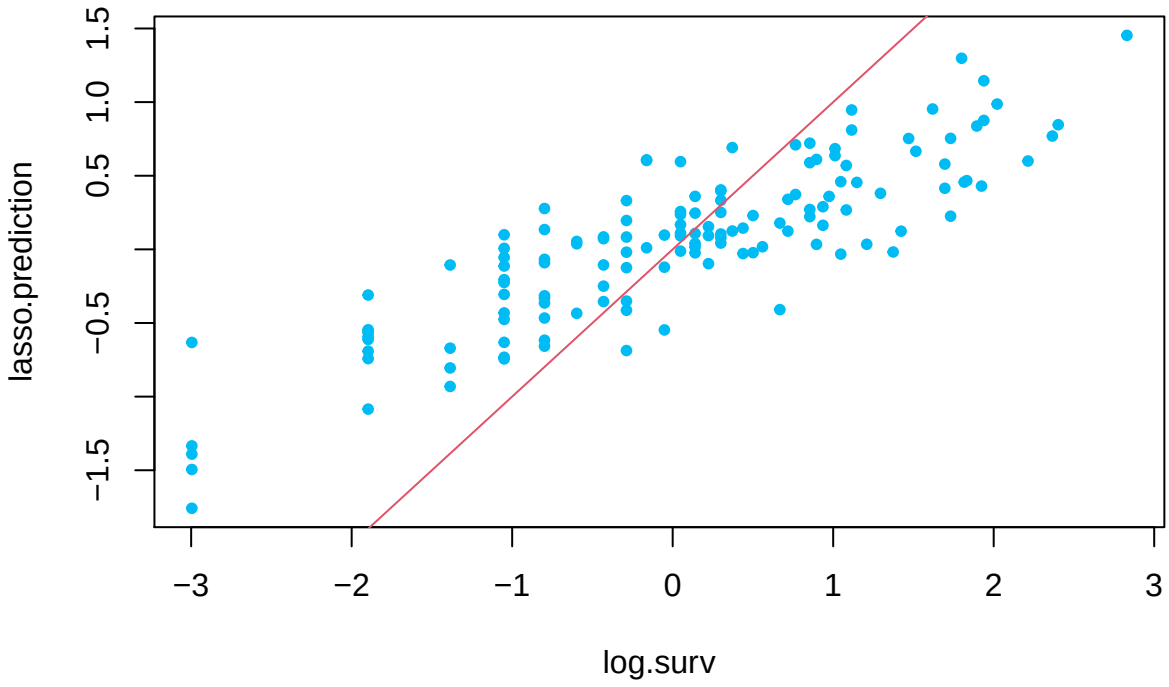
```
## [1] 0.05589672
```

When Lasso regression is estimated can be observed how in lambda min there are 44 coefficients different from zero. But, when it is done by lambda.1se it become zero in coefficients diffents from zero. Then the analysis will be done with the coefficients obtaint by lambda min. If it is not done for this way we can not go further the other regressions (because we only going to have the interception).

2.2. Compute the fitted values with the Lasso estimated model (you can use predict). Plot the observed values for the response variable against the Lasso fitted values.

```
lasso.prediction <- predict(lasso.3, newx = expr, s=cv.lasso.3$lambda.min)

plot(log.surv, lasso.prediction, pch = 20, col = "#00BCF4")
abline(a=0,b=1,col=2)
```



In order to see if the regression fits right the response values the points should fit the red line. As we can see in the plot lasso regression do not do a great job fitting the response values.

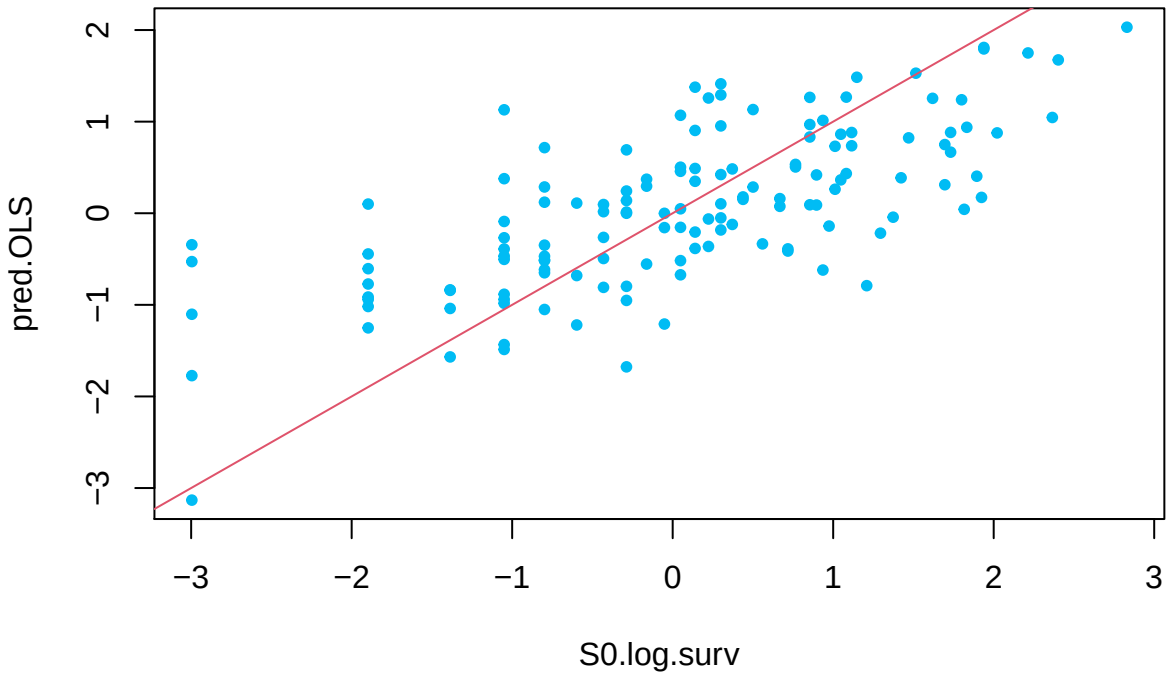
2.3. Consider the set S0 of non-zero estimated Lasso coeffcients. Use OLS to fit a regression model with response log.surv and explanatory variables the columns of expr with indexes in S0. Plot the observed values for the response variable against the OLS fitted values.

```
S0.expr <- expr[,(which(coef(lasso.3, s=cv.lasso.3$lambda.min) != 0))]
S0.log.surv <- log.surv

model.ols <- lm(S0.log.surv ~  (S0.expr))
sum.ols <- summary(model.ols)

# Plot the observed values for the response variable against the OLS fitted values.
pred.OLS <- predict(model.ols, as.data.frame(S0.expr))
```

```
plot(S0.log.surv, pred.OLS, col = "#00BCF4", pch = 20)
abline(a=0,b=1,col=2)
```



In this case OLS do a better job than Lasso regression fitting the response values. The r-squared adjusted shows a 0.2655, it is not to high but better than Lasso.

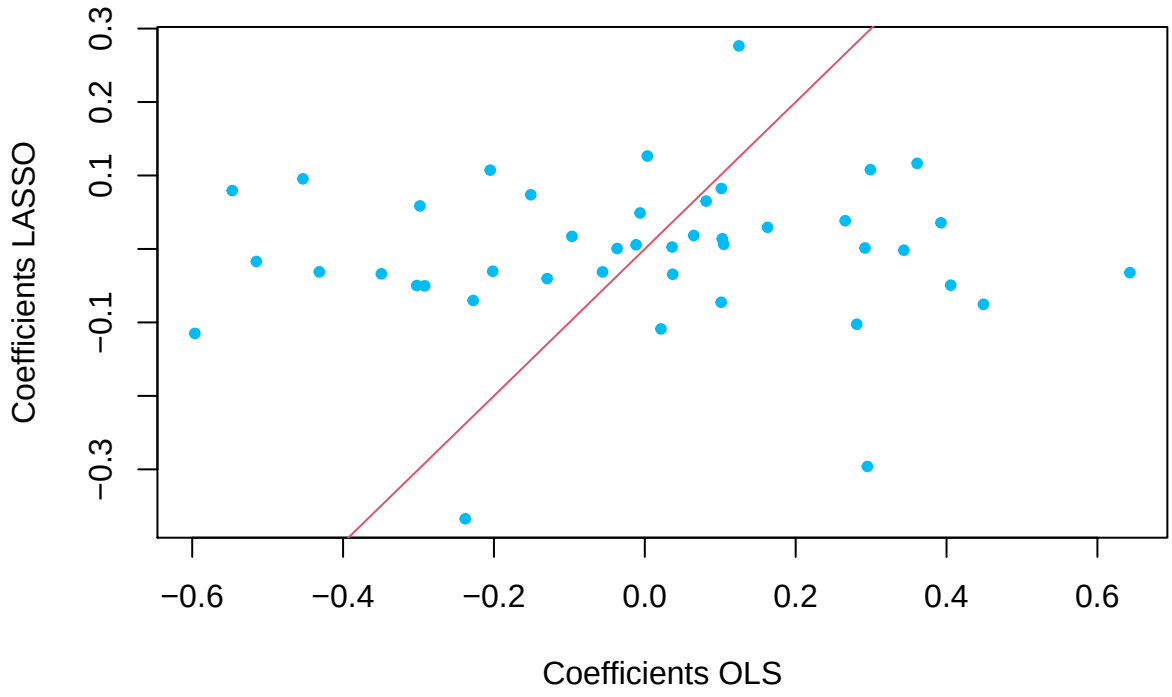
2.4. Compare the OLS and Lasso estimated coefficient. Compare the OLS and Lasso fitted values. Do a plot for that.

```
coef.ols <- sum.ols$coefficients # Coef of OLS

# Coef of Lasso model:
ncoef <- dim(coef.ols)[1]
coef.Lasso <- coef(lasso.3, s=cv.lasso.3$lambda.min)[which(coef(lasso.3, s=cv.lasso.3$lambda.min) != 0)]
cbind(as.data.frame(coef.ols[2:ncoef,1]), as.data.frame(coef.Lasso))[1:6,]
```

```
##          coef.ols[2:ncoef, 1]  coef.Lasso
## S0.expr1           -0.2049196  0.10729883
## S0.expr2           -0.5965239 -0.11499606
## S0.expr3           -0.2917388 -0.05022181
## S0.expr4           -0.2981335  0.05857538
## S0.expr5            0.2809219 -0.10246112
## S0.expr6           -0.5149631 -0.01701154
```

```r
# Do a plot for that.
coef.ols.2<- coef.ols[2:(dim(coef.ols)[1]), 1]
plot(coef.ols.2, coef.Lasso, xlab = "Coefficients OLS",
  ylab = "Coefficients LASSO", col = "#00BCF4", pch = 20)
abline(a=0,b=1,col=2)
```



We can see how OLS and Lasso coefficients estimated are really different. To be similar should fit the points to the red line. Only when one of them are higly possitive or negative the sign of the coefficients are the same.

To sum up, in this case OLS estimation do a better job in order to fit the response variable. But Lasso regression helps to delete some of the variables that in the beginning were on the dataset.