Machine Learning and Forecasting Models

Part 1: Shrinkage

Gabriel Vasconcelos

The GLMNET package

```
r   install.packages("glmnet")
```

```
library(glmnet)
?glmnet
```

Main Arguments:

- x and y: design matrix and dependent variable,
- alpha: 1 for LASSO, 0 for Ridge, $(0, 1)$ for elastic-net,
- penalty.factor: Penalty for (adaLASSO),
- nlambda: Number of λs to be teste in the regularization.

Example (Lags form an autorregressive model)

$$y_t = -0.5y_{t-1} + 0.25y_{t-3} - 0.125y_{t-5} + 0.0625y_{t-7} + \varepsilon_t, \quad \varepsilon_t \sim N(0,1)$$

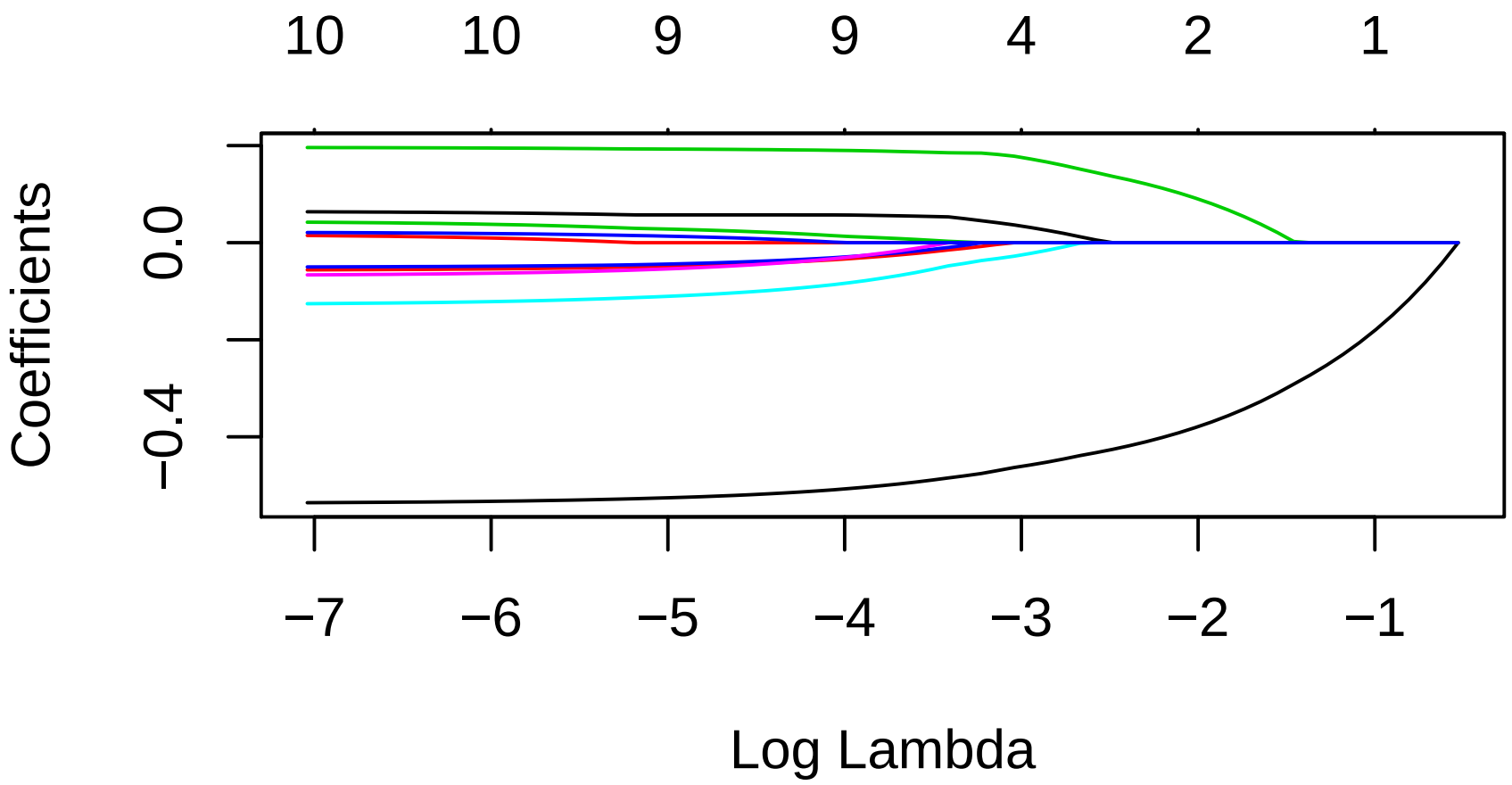
```
T=1000
lags = c(1,3,5,7)
set.seed(1)
e = rnorm(T)
y = rep(NA,T)
y[1:max(lags)] = 0
b = c(-0.5)^(1:length(lags))

for(i in (max(lags)+1):T){
  y[i] = sum(b*y[i-lags])+e[i]
}
y = y[-c(1:max(lags))]
```

Example (Lags form an autorregressive model)

```
Y = embed(y,11)
y = Y[,1]
x = Y[,-1]

lasso = glmnet(x,y)
plot(lasso,xvar="lambda")
```



Example (Lags form an autorregressive model)

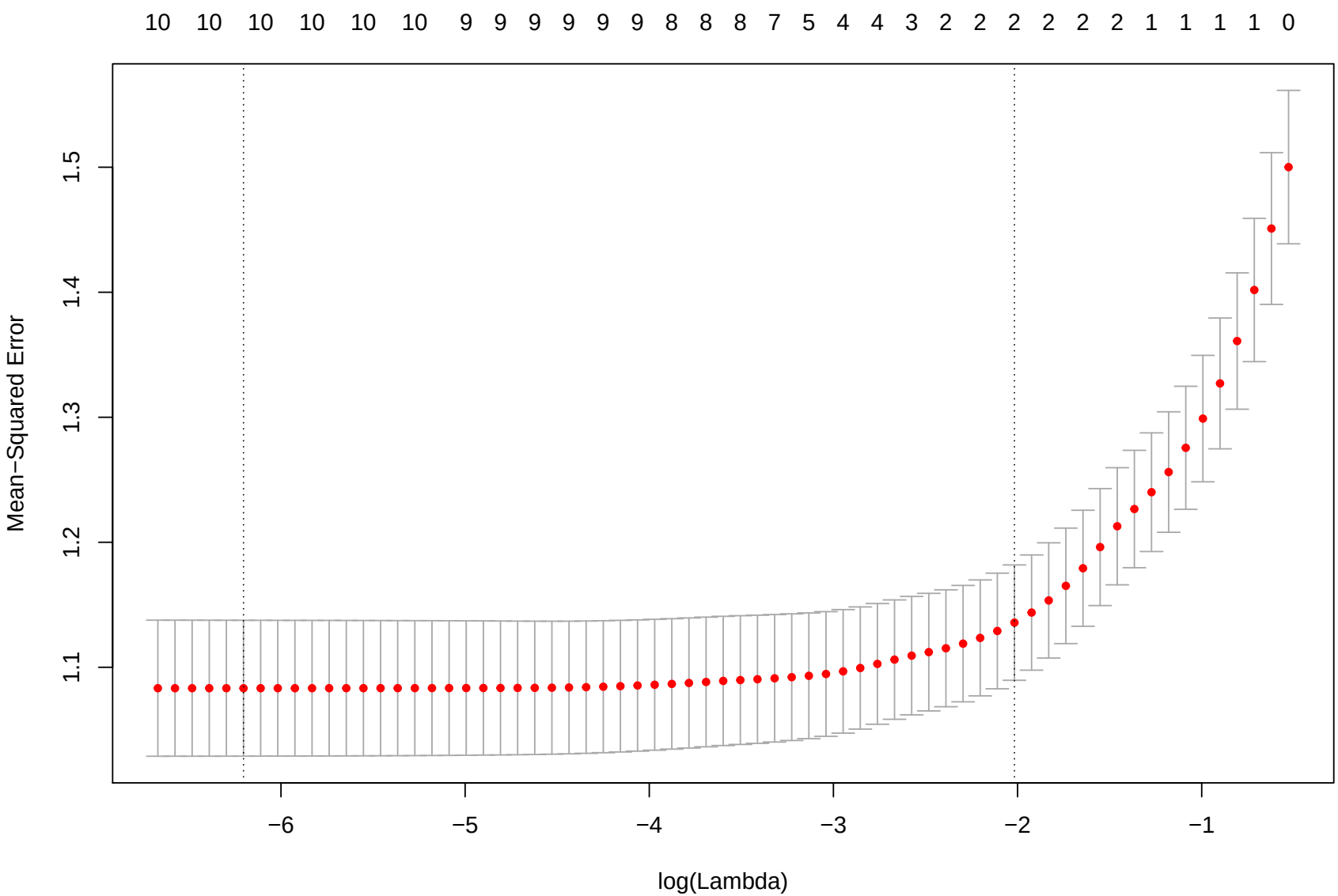
Estimate and select the best λ by cross-validation in one step:

```
lasso_cv = cv.glmnet(x,y)
coef(lasso_cv)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept) -0.01273935
## V1          -0.38106490
## V2           .
## V3           0.09198978
## V4           .
## V5           .
## V6           .
## V7           .
## V8           .
## V9           .
## V10          .
```

Example (Lags form an autorregressive model)

```
plot(lasso_cv)
```



HDeconometrics (Selecting using BIC)

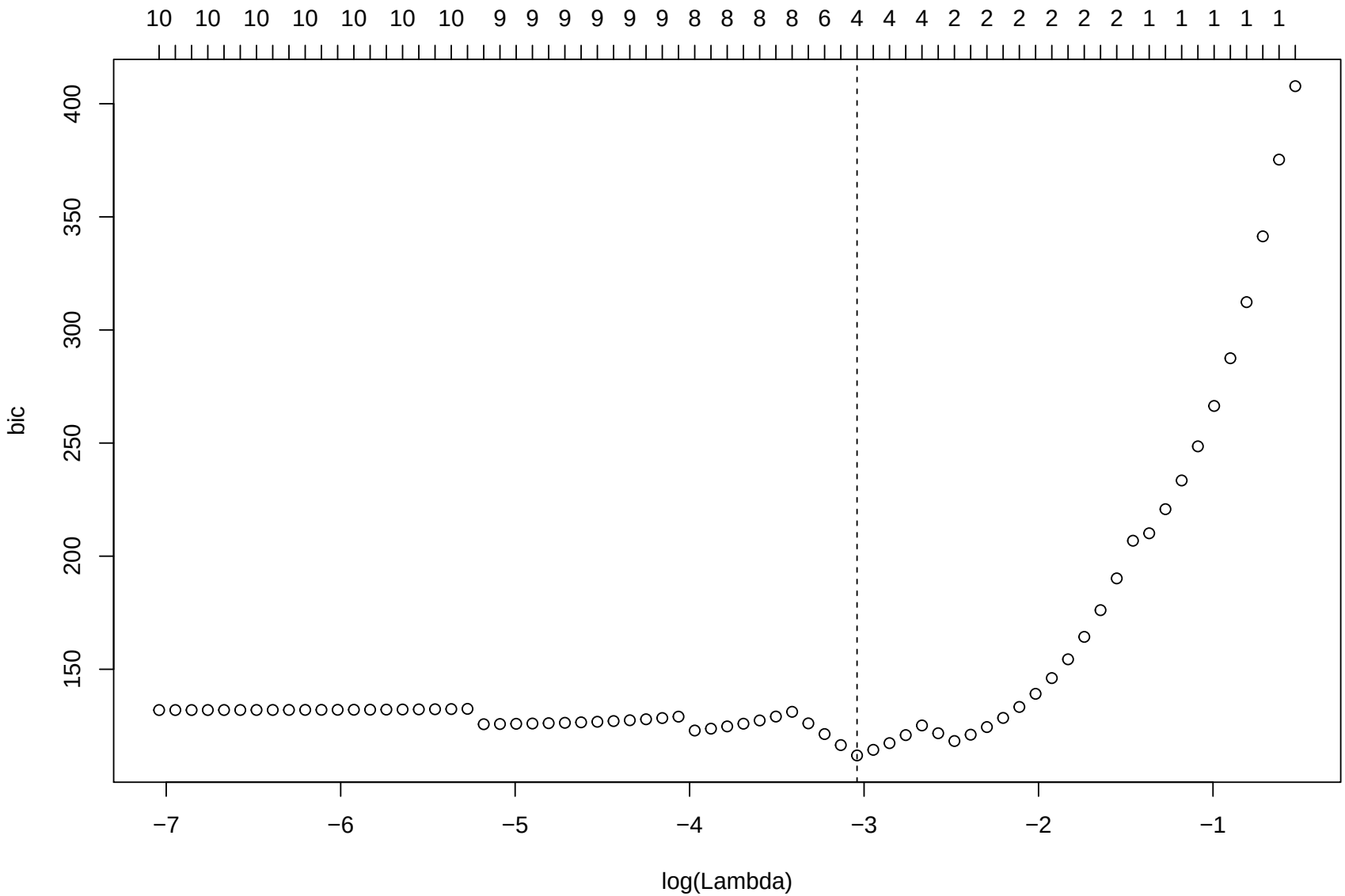
```
library(devtools) #install.packages("devtools")
install_github("gabrielrvsc/HDeconometrics")
```

```
library(HDeconometrics)
lasso_bic = ic.glmnet(x,y)
round(coef(lasso_bic),4)
```

```
## (Intercept)          V1          V2          V3          V4          V5
##     -0.0129     -0.4632      0.0000      0.1782      0.0000     -0.0278
##          V6          V7          V8          V9         V10
##      0.0000      0.0365      0.0000      0.0000      0.0000
```

HDeconometrics (Selecting using BIC)

```
plot(lasso_bic)
```



HDeconometrics (Selecting using BIC)

- Same parameters as the glmnet.
- The new parameter “crit” has the BIC as default.
- Standard model is the LASSO.

HDeconometrics (Selecting using BIC)

- Ridge:

```
ridge = ic.glmnet(x,y,alpha=0)
round(coef(ridge),4)
```

```
## (Intercept)          V1          V2          V3          V4          V5
##     -0.0145     -0.5023     -0.0437      0.1846     -0.0555     -0.1121
##          V6          V7          V8          V9         V10
##     -0.0545      0.0616      0.0078      0.0402      0.0211
```

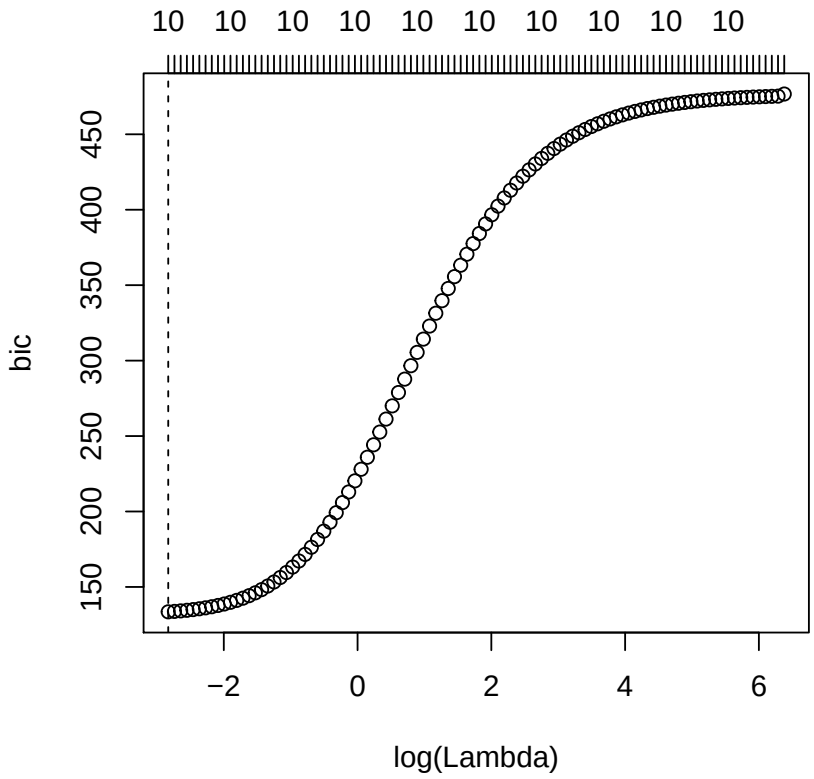
- Elastic-net:

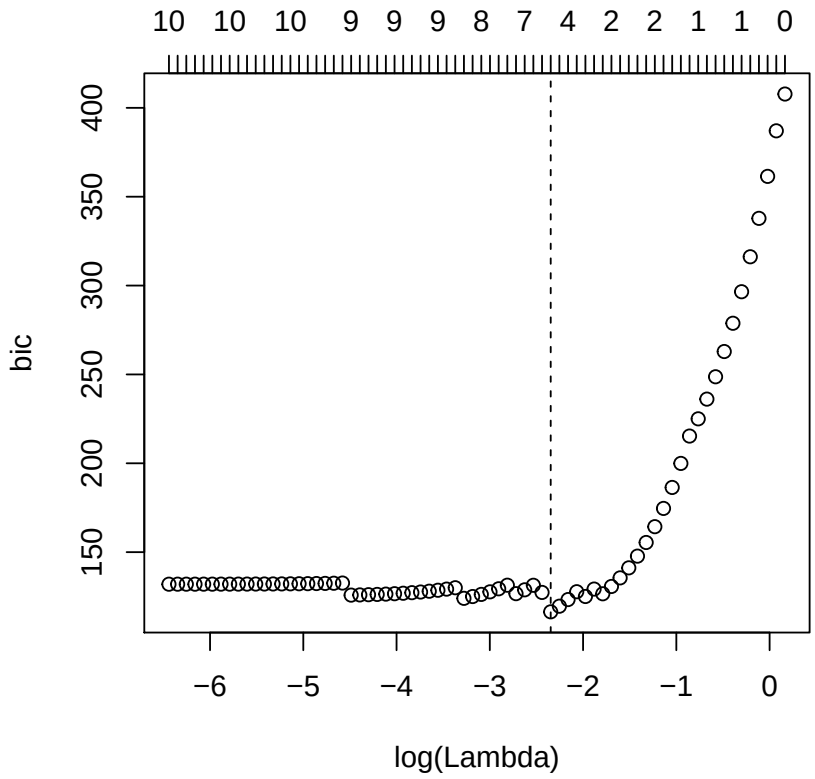
```
elnet = ic.glmnet(x,y,alpha=0.5)
round(coef(elnet),4)
```

```
## (Intercept)          V1          V2          V3          V4          V5
##     -0.0127     -0.4441      0.0000      0.1687      0.0000     -0.0223
##          V6          V7          V8          V9         V10
##      0.0000      0.0333      0.0000      0.0000      0.0000
```

HDeconometrics (Selecting using BIC)

```
par(mfrow=c(1,2))
plot(ridge)
plot(elnet)
```





adaLASSO

- Before estimating the adaLASSO one must calculate the penalties using the betas from the LASSO.

```
LASSO_coef = coef(lasso_bic)[-1]
penalty = abs(LASSO_coef+0.01)^(-1)
adaLASSO = ic.glmnet(x,y, penalty.factor = penalty)
round(LASSO_coef,4)
```

```
##      V1      V2      V3      V4      V5      V6      V7      V8      V9
## -0.4632  0.0000  0.1782  0.0000 -0.0278  0.0000  0.0365  0.0000  0.0000
##     V10
##  0.0000
```

```
round(coef(adaLASSO)[-1],4)
```

```
##      V1      V2      V3      V4      V5      V6      V7      V8      V9
## -0.5128  0.0000  0.2315  0.0000 -0.0664  0.0000  0.0806  0.0000  0.0000
##     V10
##  0.0000
```