Exercise 4

Nikolaus Czernin

```r
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library("knitr")
# install.packages("glmnet")
library("glmnet")
```

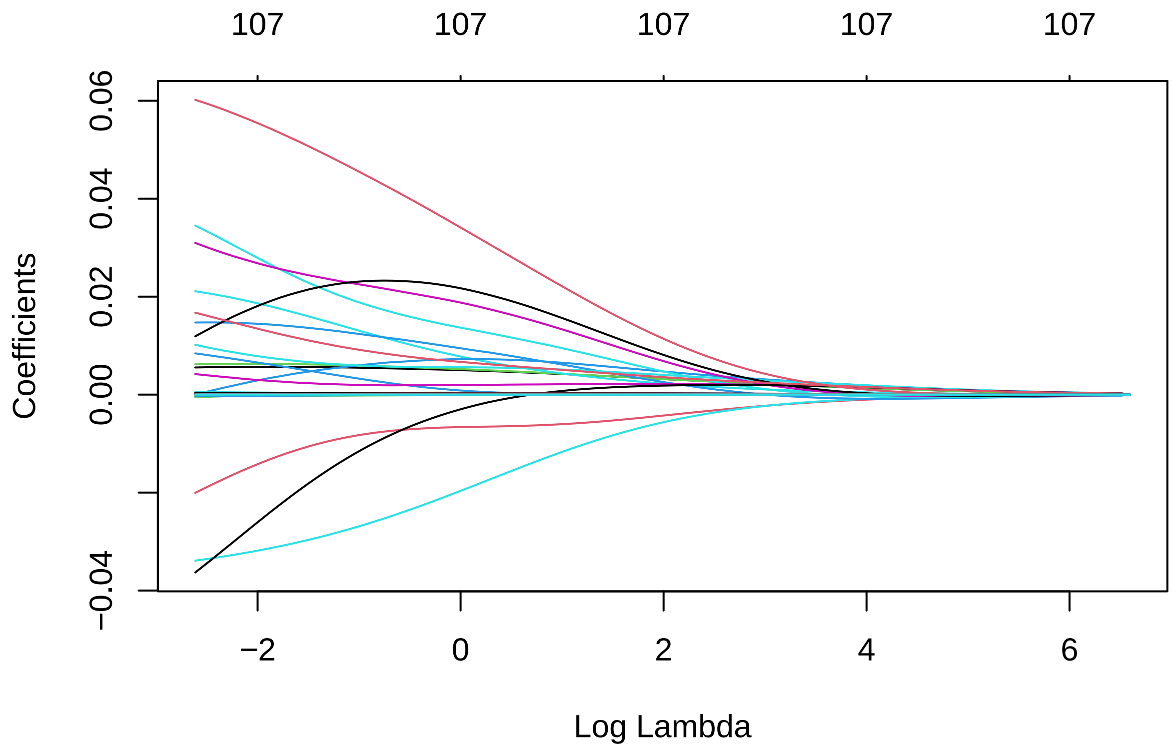
```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-8
```

```r
set.seed(11721138)
load("building.RData")
# df %>% head()
N <- nrow(df)
train_ids <- sample(1:N, (N %/% 3) * 2)
train <- df[train_ids, ]
test <- df[-train_ids, ]
dim(df)
```

```
## [1] 372 108
```

Ridge Regression

```r
ridge <- glmnet(train %>% select(-y), train$y, alpha=0)
ridge %>% plot(xvar="lambda")
```

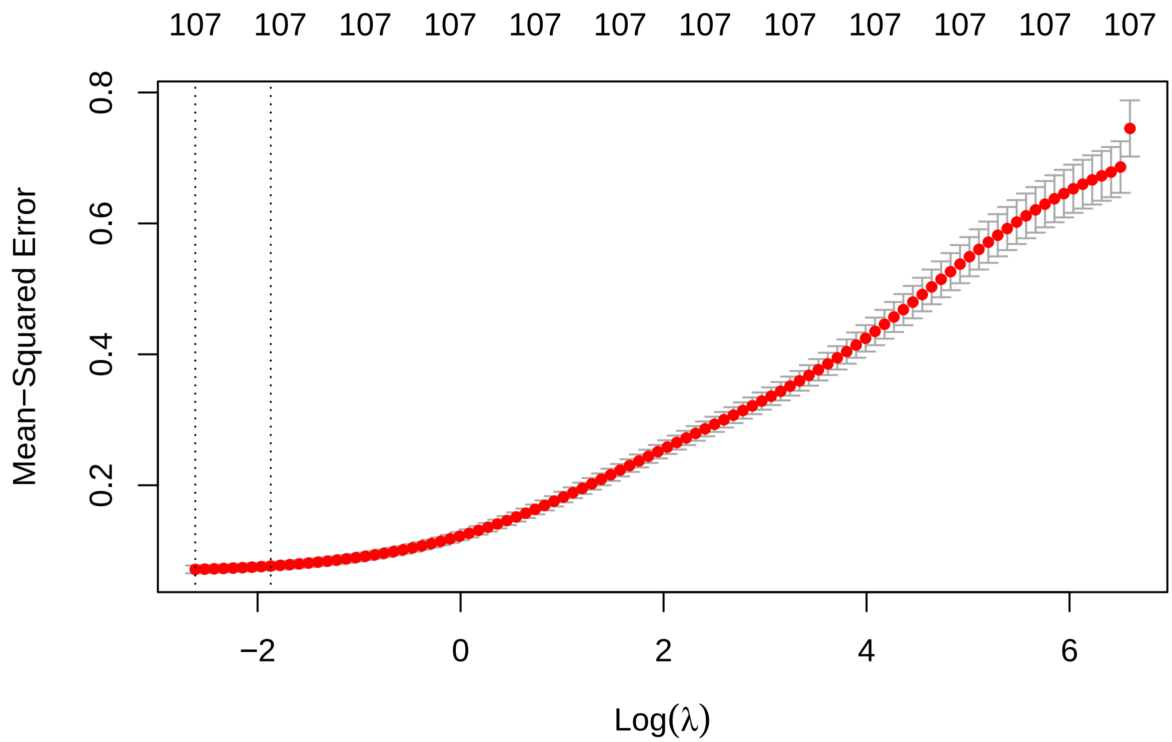


The plot shows how much the regularization parameter Lambda keeps the size of the coefficients of the model in check. The larger Lambda gets, the more the coefficients approach 0. It does not shrink any of them all the way down to zero, as can be seen by the number on the upper x axis, indicating the used number of variables.
The x-axis shows the log of the lambda values, i.e. exponentially rising numbers to allow trying for different magnitudes of lambda. The lambda values of the x-ticks are for example = {0.135, 1, 7.389, 54.598, 403.429}. I presume that the package generates the lamdba value range from top to bottom, getting the lambda value at which all coefficients are almost 0 and a set number of value below. According to the documentation of the package, it generates 100 different lambda values by default.

`alpha` is the parameter that defines whether we are doing Ridge or Lasso regression, where 0 is Ridge and 1 is Lasso. It is not binary though, but a floating point number $\in [0,1]$. It defines the nature of the penalty applied to coefficient size. The penalty's formula is $(1-\alpha)/2||\beta||_2^2 + \alpha||\beta||_1$.

Cross Validation

```r
ridge.cv <- cv.glmnet(train %>% select(-y) %>% as.matrix(), train$y, alpha=0, nfolds=10)
ridge.cv %>% plot()
```



The plot above shows the MSE results (red dots) of the 10-fold cross validation on different lambda values and also the standard errors (vertical intervals surrounding).
There are two dashed lines, where the leftmost one marks the lambda value with the globally minimal MSE, which is 0.073, and the one on the right is the largest lambda value where the MSE is not significantly worse than the next-smaller lambda's MSE. This way of selecting an optimal lambda values is called the one-standard error rule. The lambda value here is 0.154

All in all, we still always stick with 107 coefficients, as none of them are reduced all the way down to 0.

We can get the coefficients and the lambda value at the right dashed line now:

```
# get the coefficients using the 1-standard error rule
coef(ridge.cv,s="lambda.1se")
```

```
## 108 x 1 sparse Matrix of class "dgCMatrix"
##                                s1
## (Intercept)          2.172428e+00
## START.YEAR           5.674151e-03
## START.QUARTER       -1.314820e-02
## COMPLETION.YEAR      6.263716e-03
## COMPLETION.QUARTER   6.124613e-03
## PhysFin1            -3.133869e-02
## PhysFin2             1.422555e-05
## PhysFin3            -8.619841e-05
## PhysFin4             1.907416e-05
## PhysFin5            -4.899865e-05
## PhysFin6             2.590497e-04
```

```
## PhysFin7             1.799705e-02
## PhysFin8             2.682560e-04
## Econ1                8.518456e-06
## Econ2                6.808321e-05
## Econ3                2.367297e-04
## Econ4                3.425190e-03
## Econ5                1.135425e-08
## Econ6                9.147665e-07
## Econ7                1.524036e-05
## Econ8                1.307993e-04
## Econ9                5.042320e-07
## Econ10               1.429969e-02
## Econ11               2.050152e-06
## Econ12               5.126611e-06
## Econ13               9.102279e-09
## Econ14               1.469277e-05
## Econ15               2.628539e-04
## Econ16               9.826436e-05
## Econ17               5.449309e-06
## Econ18               4.267601e-07
## Econ19               3.788238e-08
## Econ1.lag1           1.423630e-05
## Econ2.lag1           1.451734e-05
## Econ3.lag1           1.643102e-05
## Econ4.lag1           7.468939e-03
## Econ5.lag1           1.377683e-08
## Econ6.lag1          -2.245175e-06
## Econ7.lag1          -8.290790e-05
## Econ8.lag1           1.954401e-04
## Econ9.lag1           6.016001e-07
## Econ10.lag1          2.651849e-02
## Econ11.lag1         -5.228841e-06
## Econ12.lag1          5.840783e-06
## Econ13.lag1          5.366923e-06
## Econ14.lag1          9.546540e-06
## Econ15.lag1          2.089869e-04
## Econ16.lag1          9.769342e-05
## Econ17.lag1          2.896918e-06
## Econ18.lag1          4.441327e-07
## Econ19.lag1          1.844229e-08
## Econ1.lag2          -6.870043e-06
## Econ2.lag2           3.127551e-05
## Econ3.lag2          -8.682174e-05
## Econ4.lag2           2.787634e-03
## Econ5.lag2           1.476923e-08
## Econ6.lag2          -6.090719e-06
## Econ7.lag2          -2.618183e-04
## Econ8.lag2           1.771550e-04
## Econ9.lag2           1.403271e-06
## Econ10.lag2          2.610561e-02
## Econ11.lag2          5.811010e-06
## Econ12.lag2          2.653549e-06
## Econ13.lag2          1.400416e-06
## Econ14.lag2          1.448822e-05
```

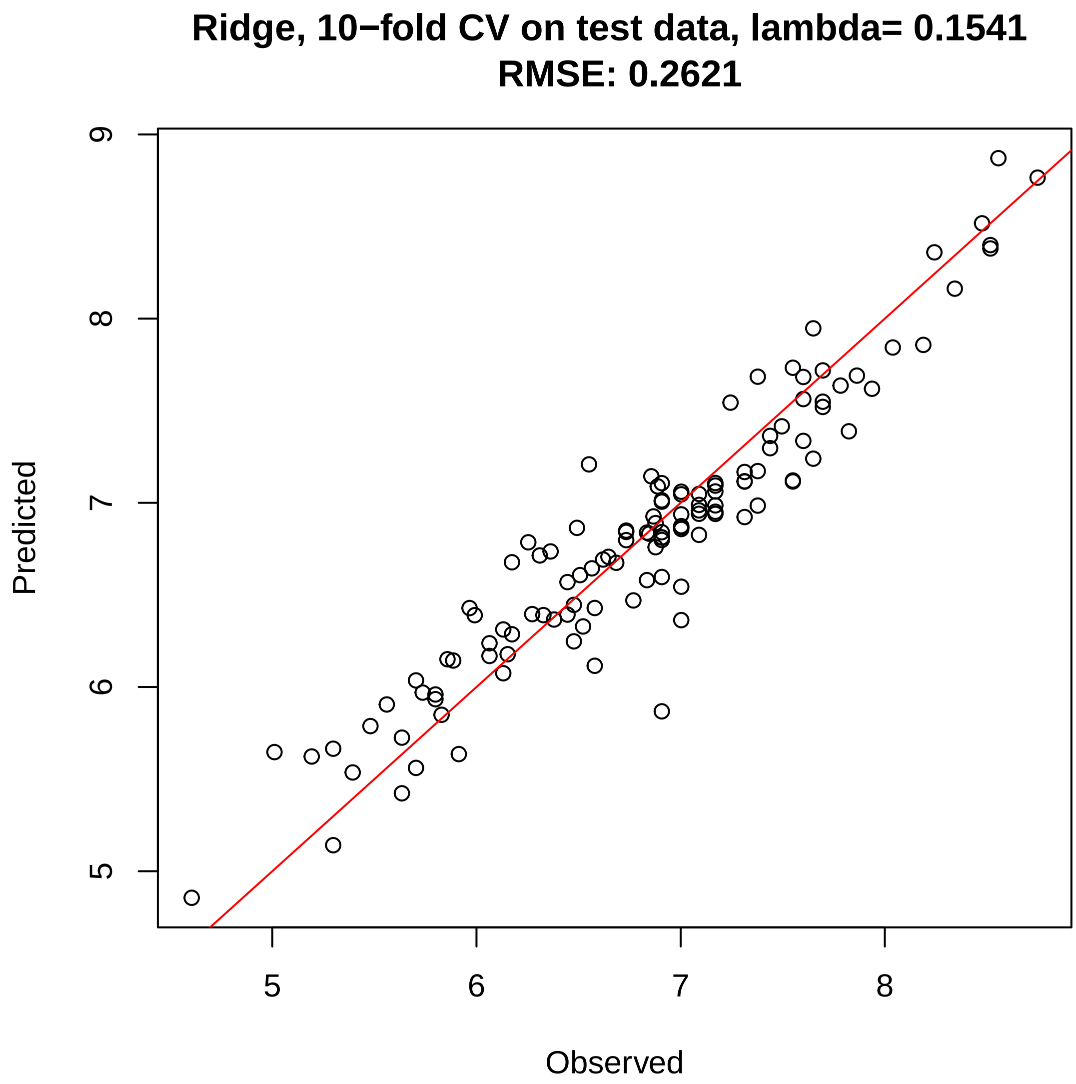
```
## Econ15.lag2          1.768869e-04
## Econ16.lag2          1.434788e-04
## Econ17.lag2         -1.469593e-06
## Econ18.lag2          1.020628e-06
## Econ19.lag2          8.213704e-09
## Econ1.lag3           2.696270e-05
## Econ2.lag3           9.981341e-05
## Econ3.lag3           1.733380e-05
## Econ4.lag3          -2.391428e-02
## Econ5.lag3           8.411039e-09
## Econ6.lag3          -3.147912e-06
## Econ7.lag3          -2.352807e-04
## Econ8.lag3           4.004694e-04
## Econ9.lag3           7.456226e-07
## Econ10.lag3          1.911997e-02
## Econ11.lag3          1.508244e-05
## Econ12.lag3          8.908567e-06
## Econ13.lag3          1.876527e-06
## Econ14.lag3          1.981595e-05
## Econ15.lag3          2.439123e-04
## Econ16.lag3          1.512279e-04
## Econ17.lag3         -2.022896e-06
## Econ18.lag3         -1.328181e-07
## Econ19.lag3          1.695960e-08
## Econ1.lag4           1.596308e-05
## Econ2.lag4           1.735986e-04
## Econ3.lag4           3.373272e-04
## Econ4.lag4           1.280583e-02
## Econ5.lag4           3.892952e-09
## Econ6.lag4           2.340699e-06
## Econ7.lag4           3.976534e-06
## Econ8.lag4          -1.403358e-05
## Econ9.lag4          -2.915527e-08
## Econ10.lag4          5.428056e-02
## Econ11.lag4          4.191224e-06
## Econ12.lag4          1.811547e-05
## Econ13.lag4          3.438890e-06
## Econ14.lag4          1.429084e-05
## Econ15.lag4          3.258896e-04
## Econ16.lag4          1.385515e-04
## Econ17.lag4          3.106905e-06
## Econ18.lag4          4.462396e-07
## Econ19.lag4          1.758737e-08
```

The coefficients are all very small, but never zero.

Testing the model

```r
rmse <- function(y, yhat, r=4){
  sqrt(mean((y-yhat)^2)) %>% round(4)
}
# ridge.cv
```

```r
ridge.yhat <- predict(ridge.cv, newx=test %>% select(-y) %>% as.matrix(),s="lambda.1se")
ridge.rmse <- rmse(test$y, ridge.yhat)

plot(test$y, ridge.yhat, main=paste("Ridge, 10-fold CV on test data, lambda=",  ridge.cv$lambda.1se %>%
     xlab="Observed", ylab="Predicted")

abline(coef = c(0,1), col="red")
```



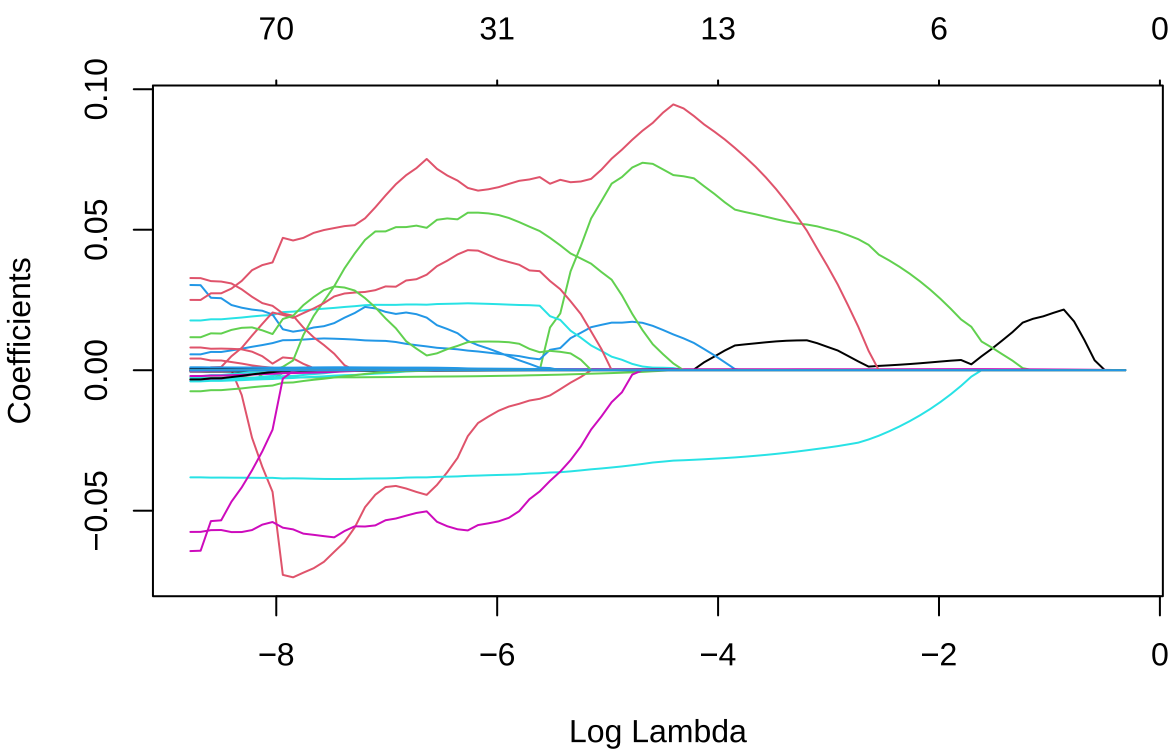

```r
data.frame(
  model = c("Full linear model",  "Significant coefficients only",  "PCR",  "PLS",  "Ridge"),
  RMSE = c(0.540,0.265,0.287,0.284,ridge.rmse %>% round(3))
) %>% kable()
```

model	RMSE
Full linear model	0.540
Significant coefficients only	0.265
PCR	0.287
PLS	0.284
Ridge	0.262

So far, Ridge regression can outperform the cross validation model where we picked only the coefficients that were singificant in the full linear model, and even PCR and PLS.

Lasso Regression

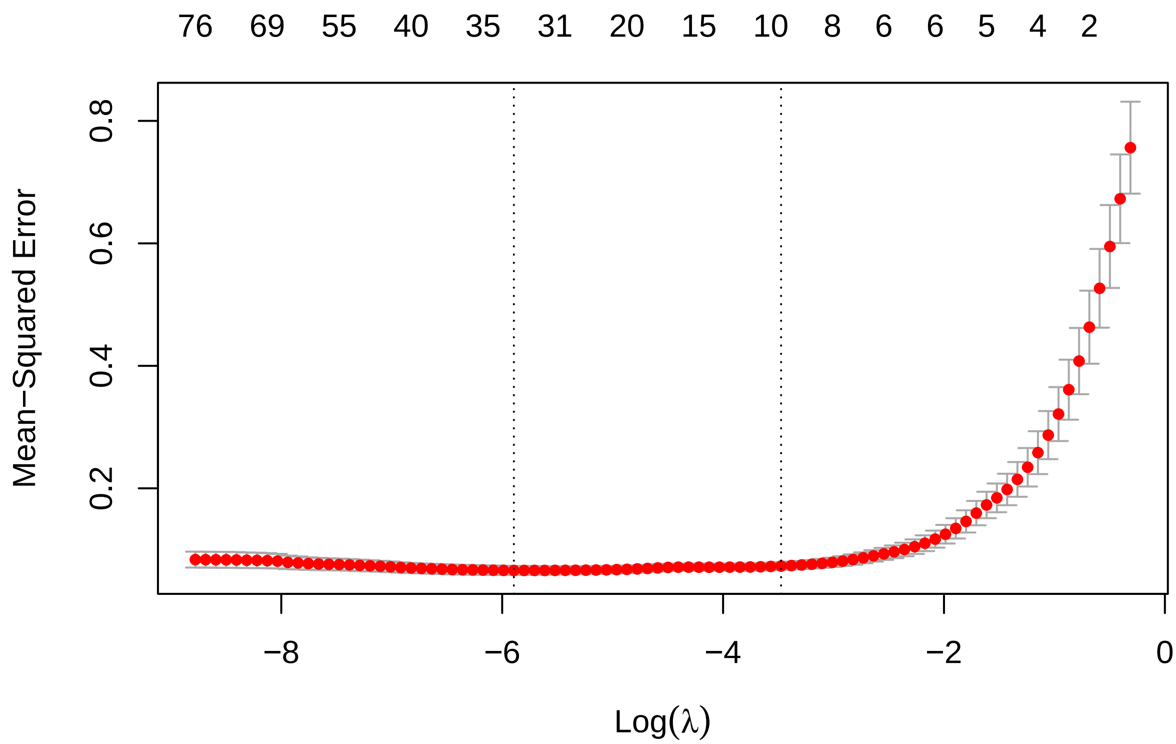
```
lasso <- glmnet(train %>% select(-y), train$y, alpha=1)
lasso %>% plot(xvar="lambda")
```



In Lasso regression, where the parameter alpha is 1, even at similar lambda values a lot of the coefficients are set to 0, effectively causing the model to do variable selection. Even at a lambda value as low as3.35×10^{-4} we only end up with 56 non-zero coefficients, as oppose to 107 like in Ridge regression.

Cross Validation

```
lasso.cv <- cv.glmnet(train %>% select(-y) %>% as.matrix(), train$y, alpha=1, nfolds=10)
lasso.cv %>% plot()
```



In Lasso regression we can pick way higher lambda values than before. Whereas before the 1-standard error rule made up pick lambda=0.073, now we get an optimal value of lambda=0.031, here there are 11 variables unequal to zero (incl an intercept). The minimal MSE lambda value is now 0.073.

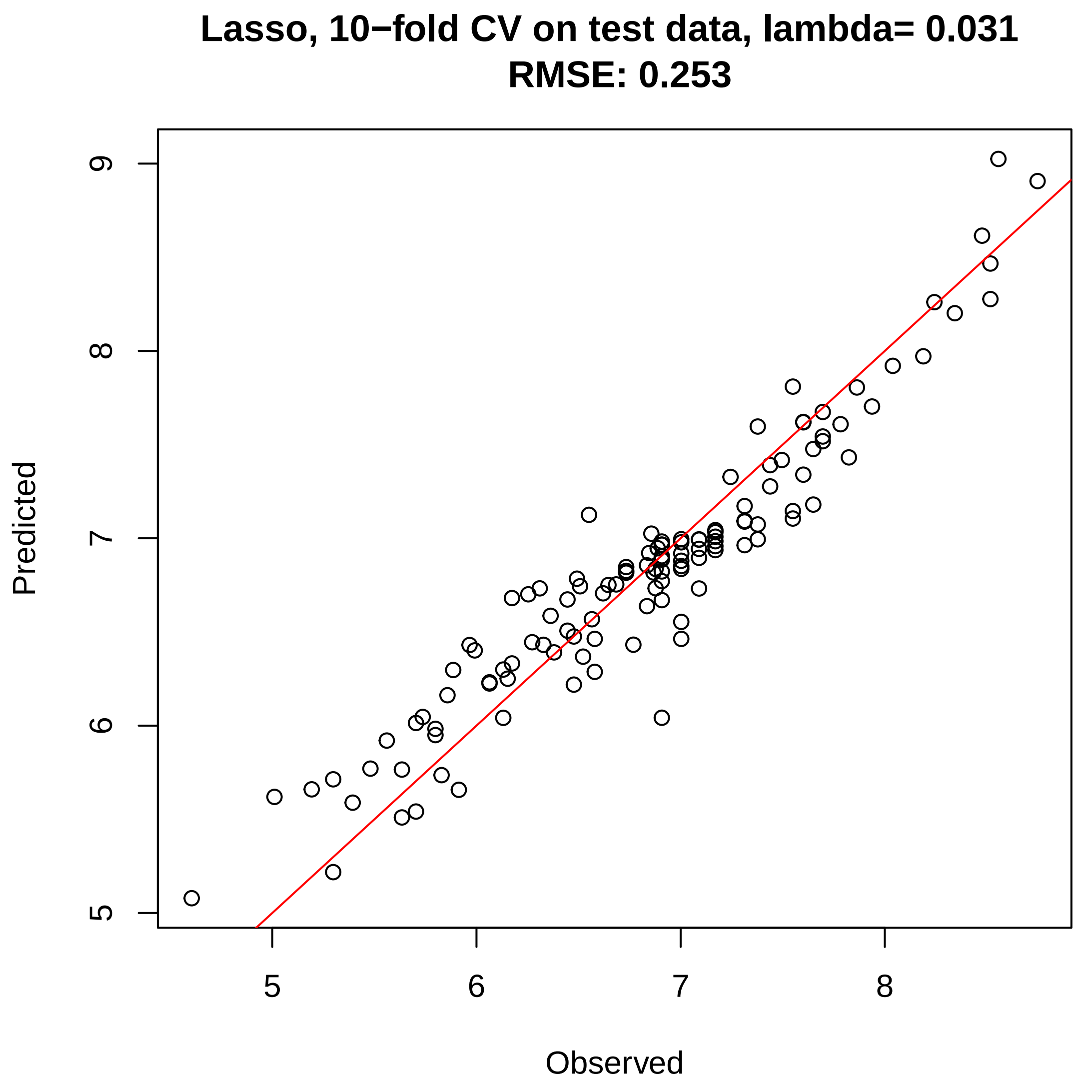
```
# get the coefficients using the 1-standard error rule
coef(lasso.cv,s="lambda.1se")
```

```
## 108 x 1 sparse Matrix of class "dgCMatrix"
##                               s1
## (Intercept)        -1.431092e-01
## START.YEAR          1.025716e-02
## START.QUARTER       .
## COMPLETION.YEAR     5.374455e-02
## COMPLETION.QUARTER  .
## PhysFin1           -2.976887e-02
## PhysFin2            .
## PhysFin3            .
## PhysFin4            .
## PhysFin5            .
## PhysFin6            9.813203e-05
```

```
## PhysFin7          .
## PhysFin8          3.440695e-04
## Econ1             .
## Econ2             .
## Econ3             .
## Econ4             .
## Econ5             .
## Econ6             .
## Econ7             .
## Econ8             .
## Econ9             .
## Econ10            .
## Econ11            .
## Econ12            .
## Econ13            .
## Econ14            .
## Econ15            .
## Econ16            .
## Econ17            .
## Econ18            .
## Econ19            .
## Econ1.lag1        .
## Econ2.lag1        .
## Econ3.lag1        .
## Econ4.lag1        .
## Econ5.lag1        .
## Econ6.lag1        .
## Econ7.lag1        .
## Econ8.lag1        7.146399e-05
## Econ9.lag1        .
## Econ10.lag1       .
## Econ11.lag1       .
## Econ12.lag1       .
## Econ13.lag1       .
## Econ14.lag1       .
## Econ15.lag1       .
## Econ16.lag1       .
## Econ17.lag1       .
## Econ18.lag1       .
## Econ19.lag1       .
## Econ1.lag2        .
## Econ2.lag2        .
## Econ3.lag2        .
## Econ4.lag2        .
## Econ5.lag2        .
## Econ6.lag2        .
## Econ7.lag2        .
## Econ8.lag2        1.458874e-04
## Econ9.lag2        8.598864e-07
## Econ10.lag2       .
## Econ11.lag2       .
## Econ12.lag2       .
## Econ13.lag2       .
## Econ14.lag2       .
```

```
## Econ15.lag2         .
## Econ16.lag2         .
## Econ17.lag2         .
## Econ18.lag2         .
## Econ19.lag2         .
## Econ1.lag3          .
## Econ2.lag3          .
## Econ3.lag3          .
## Econ4.lag3          .
## Econ5.lag3          .
## Econ6.lag3          .
## Econ7.lag3          .
## Econ8.lag3          .
## Econ9.lag3          .
## Econ10.lag3         .
## Econ11.lag3         .
## Econ12.lag3         .
## Econ13.lag3         .
## Econ14.lag3         8.287425e-05
## Econ15.lag3         .
## Econ16.lag3         .
## Econ17.lag3         .
## Econ18.lag3         .
## Econ19.lag3         .
## Econ1.lag4          .
## Econ2.lag4          .
## Econ3.lag4          .
## Econ4.lag4          .
## Econ5.lag4          .
## Econ6.lag4          .
## Econ7.lag4          .
## Econ8.lag4          .
## Econ9.lag4          .
## Econ10.lag4         6.445550e-02
## Econ11.lag4         .
## Econ12.lag4         .
## Econ13.lag4         .
## Econ14.lag4         .
## Econ15.lag4         .
## Econ16.lag4         .
## Econ17.lag4         .
## Econ18.lag4         .
## Econ19.lag4         .
```

```
lasso.yhat <- predict(lasso.cv, newx=test %>% select(-y) %>% as.matrix(),s="lambda.1se")
lasso.rmse <- rmse(test$y, lasso.yhat)

plot(test$y, lasso.yhat, main=paste("Lasso, 10-fold CV on test data, lambda=",  lasso.cv$lambda.1se %>%
     xlab="Observed", ylab="Predicted")

abline(coef = c(0,1), col="red")
```



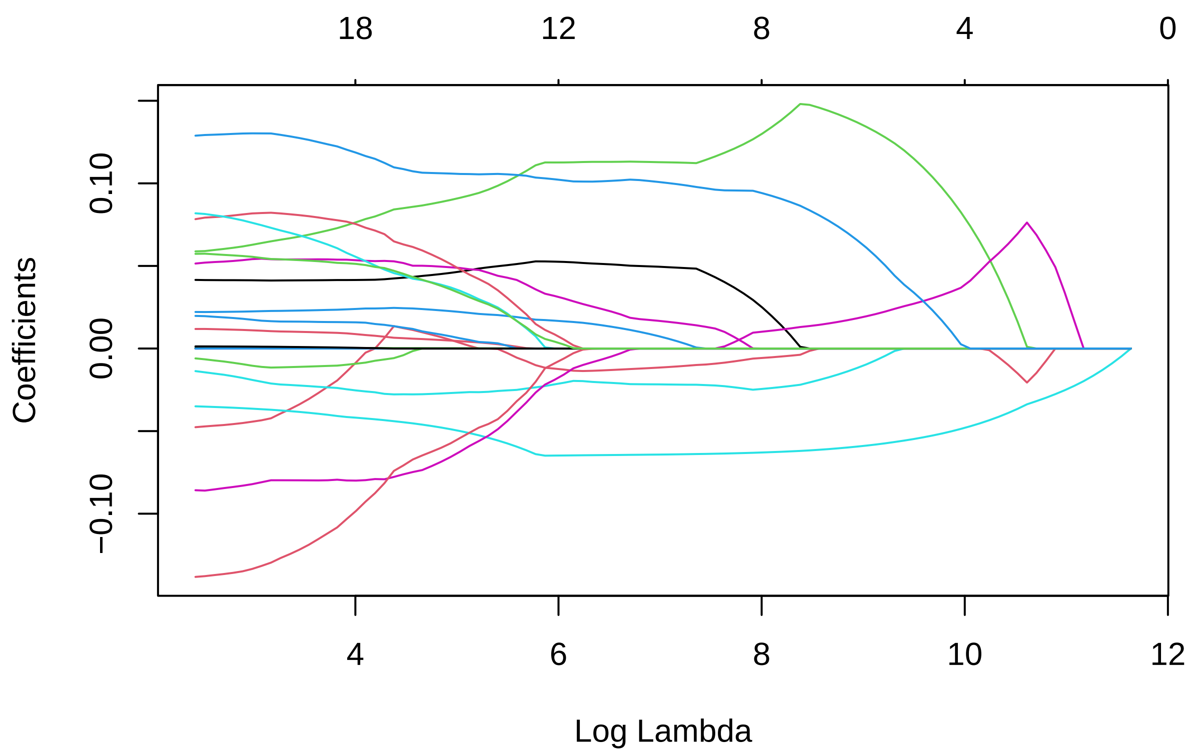

```
data.frame(
  model = c("Full linear model",  "Significant coefficients only",  "PCR",  "PLS",  "Ridge", "Lasso"),
  RMSE = c(0.540,0.265,0.287,0.284,ridge.rmse %>% round(3), lasso.rmse %>% round(3))
) %>% kable()
```

model	RMSE
Full linear model	0.540
Significant coefficients only	0.265
PCR	0.287
PLS	0.284
Ridge	0.262
Lasso	0.253

Lasso regression performed even a little better on the test dataset than Ridge Regression.

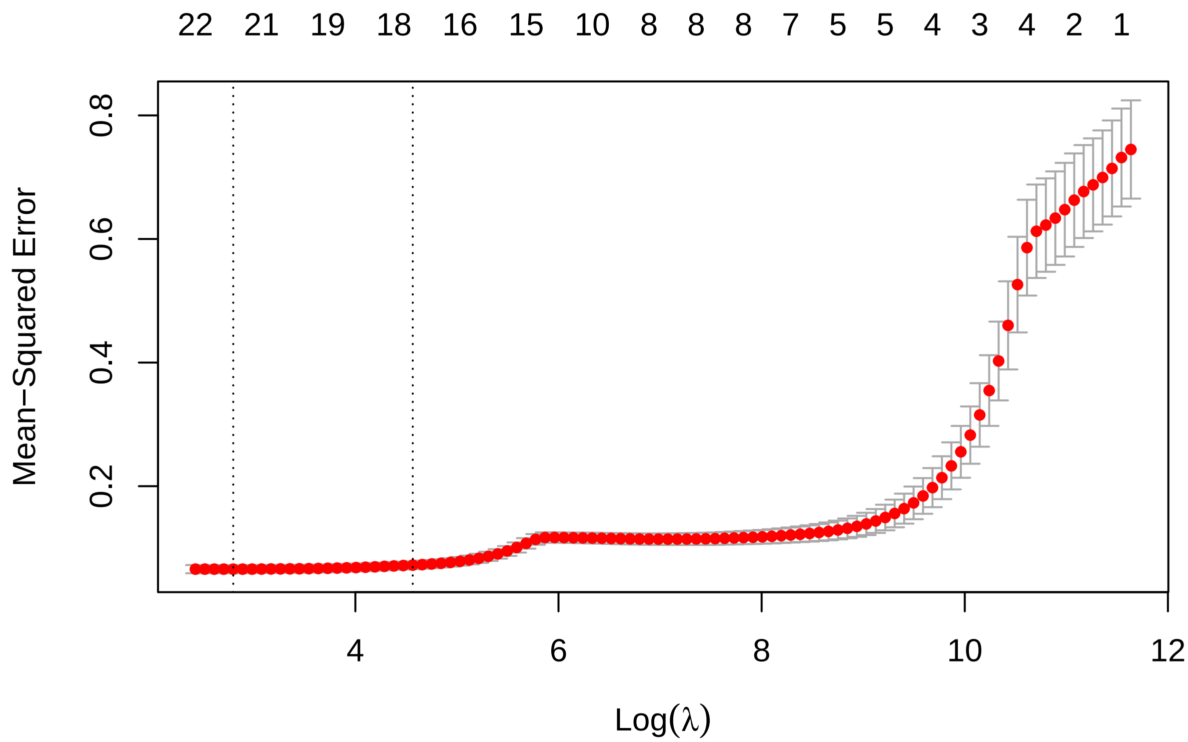
Adaptive Lasso

```
ridge.coeffs <- coef(ridge.cv,s="lambda.1se")
alasso <- glmnet(train %>% select(-y) %>% as.matrix(), train$y,
                 penalty.factor = 1 / abs(ridge.coeffs[-1]))
plot(alasso, xvar="lambda")
```



When using the ridge coefficients as penalty, the resulting model also does variable selection, reducing many of the coefficients down to zero. This model is designed to assign importance weights to strong predictors, in order not to exclude them in Lasso regression, which could prune strong predictors.

```
alasso.cv <- cv.glmnet(train %>% select(-y) %>% as.matrix(), train$y, penalty.factor = 1 / abs(ridge.co
alasso.cv %>% plot()
```



With adaptive lasso, the new minimum MSE lambda value is 16.4047235 and the optimal lambda value is 96.0828069, both of which are way higher lambda values than in the previous 2 models. In the optimal case, we select 18, more than before.

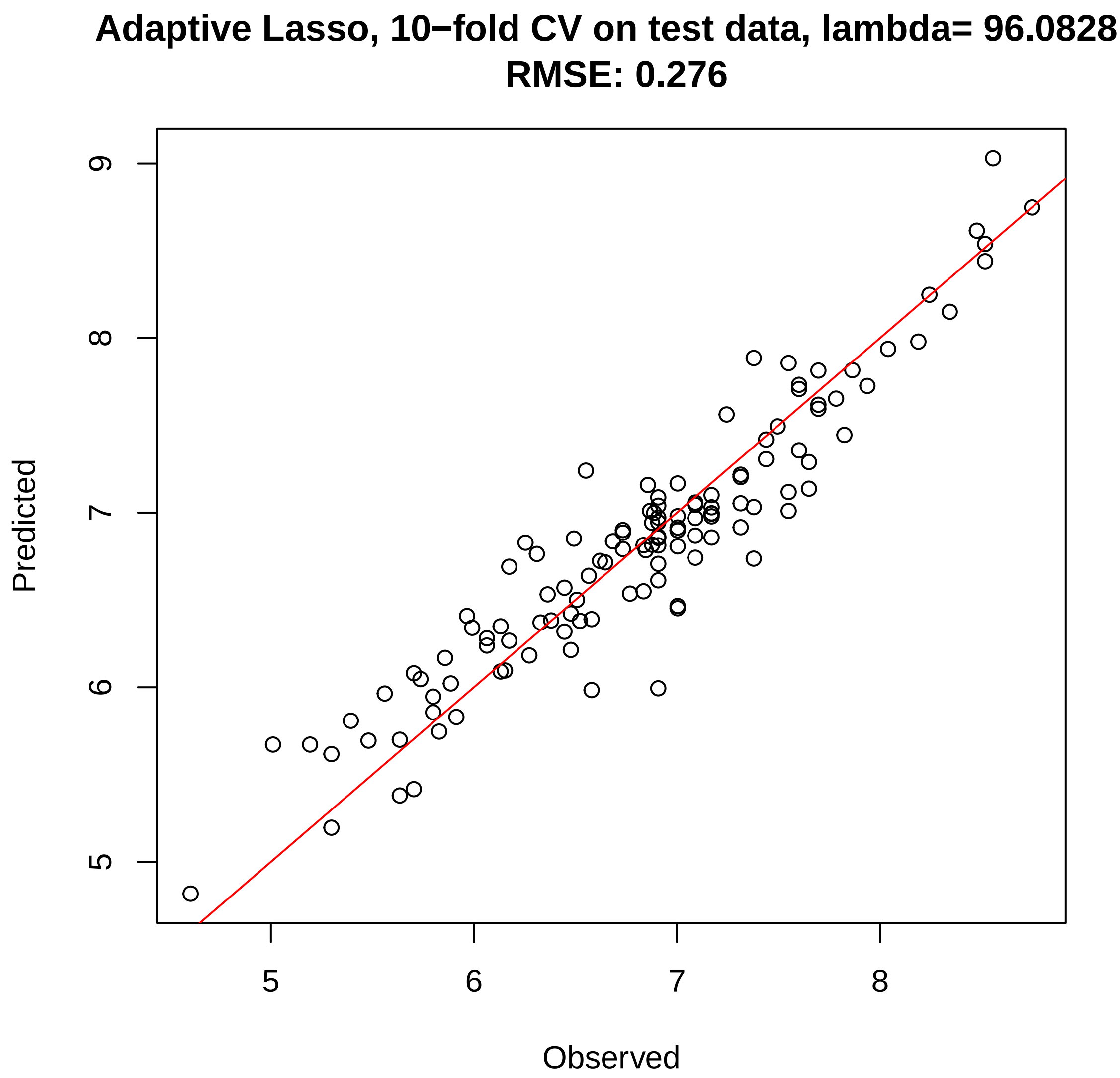
```
# get the coefficients using the 1-standard error rule
coef(alasso.cv,s="lambda.1se")
```

```
## 108 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)        -5.4876528463
## START.YEAR          0.0433325832
## START.QUARTER       0.0112356922
## COMPLETION.YEAR     0.0858004972
## COMPLETION.QUARTER  0.0242469115
## PhysFin1           -0.0453002210
## PhysFin2            .
## PhysFin3            .
## PhysFin4            .
## PhysFin5            .
## PhysFin6            .
## PhysFin7            0.0059634731
## PhysFin8            0.0002514425
## Econ1               .
## Econ2               .
## Econ3               .
## Econ4               0.0118671527
```

```
## Econ5             .
## Econ6             .
## Econ7             .
## Econ8             .
## Econ9             .
## Econ10           -0.0277452430
## Econ11            .
## Econ12            .
## Econ13            .
## Econ14            .
## Econ15            .
## Econ16            .
## Econ17            .
## Econ18            .
## Econ19            .
## Econ1.lag1        .
## Econ2.lag1        .
## Econ3.lag1        .
## Econ4.lag1        0.0501263396
## Econ5.lag1        .
## Econ6.lag1        .
## Econ7.lag1        .
## Econ8.lag1        .
## Econ9.lag1        .
## Econ10.lag1       0.0614914114
## Econ11.lag1       .
## Econ12.lag1       .
## Econ13.lag1       .
## Econ14.lag1       .
## Econ15.lag1       .
## Econ16.lag1       .
## Econ17.lag1       .
## Econ18.lag1       .
## Econ19.lag1       .
## Econ1.lag2        .
## Econ2.lag2        .
## Econ3.lag2        .
## Econ4.lag2       -0.0015280368
## Econ5.lag2        .
## Econ6.lag2        .
## Econ7.lag2        .
## Econ8.lag2        .
## Econ9.lag2        .
## Econ10.lag2       0.0422727540
## Econ11.lag2       .
## Econ12.lag2       .
## Econ13.lag2       .
## Econ14.lag2       .
## Econ15.lag2       .
## Econ16.lag2       .
## Econ17.lag2       .
## Econ18.lag2       .
## Econ19.lag2       .
## Econ1.lag3        .
```

```
## Econ2.lag3          .
## Econ3.lag3          .
## Econ4.lag3         -0.0747674715
## Econ5.lag3          .
## Econ6.lag3          .
## Econ7.lag3          .
## Econ8.lag3          .
## Econ9.lag3          .
## Econ10.lag3        -0.0671832859
## Econ11.lag3         .
## Econ12.lag3         .
## Econ13.lag3         .
## Econ14.lag3         .
## Econ15.lag3         .
## Econ16.lag3         .
## Econ17.lag3         .
## Econ18.lag3         .
## Econ19.lag3         .
## Econ1.lag4          .
## Econ2.lag4          .
## Econ3.lag4          .
## Econ4.lag4          0.0433173592
## Econ5.lag4          .
## Econ6.lag4          .
## Econ7.lag4          .
## Econ8.lag4          .
## Econ9.lag4          .
## Econ10.lag4         0.1073044605
## Econ11.lag4         .
## Econ12.lag4         .
## Econ13.lag4         .
## Econ14.lag4         .
## Econ15.lag4         .
## Econ16.lag4         .
## Econ17.lag4         .
## Econ18.lag4         .
## Econ19.lag4         .
```

```r
alasso.yhat <- predict(alasso.cv, newx=test %>% select(-y) %>% as.matrix(),s="lambda.1se")
alasso.rmse <- rmse(test$y, alasso.yhat)

plot(test$y, alasso.yhat, main=paste("Adaptive Lasso, 10-fold CV on test data, lambda=",  alasso.cv$lamb
     xlab="Observed", ylab="Predicted")

abline(coef = c(0,1), col="red")
```




```
data.frame(
  model = c("Full linear model",  "Significant coefficients only",  "PCR",  "PLS",  "Ridge", "Lasso", "A
  RMSE = c(0.540,0.265,0.287,0.284,ridge.rmse %>% round(3), lasso.rmse %>% round(3), alasso.rmse %>% rou
) %>% kable()
```

model	RMSE
Full linear model	0.540
Significant coefficients only	0.265
PCR	0.287
PLS	0.284
Ridge	0.262
Lasso	0.253
Adaptive Lasso	0.276

Adaptive Lasso in my case could not quite outperform Lasso or Ridge.

Comparing the coefficients of Lasso and adaptive Lasso

```r
data.frame(
  Lasso = coef(lasso.cv,s="lambda.1se")[,1],
  Adaptive.Lasso = coef(alasso.cv,s="lambda.1se")[,1]
) %>%
  mutate_all(function(x)ifelse(x==0, NA, x))
```

```
##                              Lasso Adaptive.Lasso
## (Intercept)          -1.431092e-01  -5.4876528463
## START.YEAR            1.025716e-02   0.0433325832
## START.QUARTER                   NA   0.0112356922
## COMPLETION.YEAR       5.374455e-02   0.0858004972
## COMPLETION.QUARTER              NA   0.0242469115
## PhysFin1             -2.976887e-02  -0.0453002210
## PhysFin2                        NA             NA
## PhysFin3                        NA             NA
## PhysFin4                        NA             NA
## PhysFin5                        NA             NA
## PhysFin6              9.813203e-05             NA
## PhysFin7                        NA   0.0059634731
## PhysFin8              3.440695e-04   0.0002514425
## Econ1                           NA             NA
## Econ2                           NA             NA
## Econ3                           NA             NA
## Econ4                           NA   0.0118671527
## Econ5                           NA             NA
## Econ6                           NA             NA
## Econ7                           NA             NA
## Econ8                           NA             NA
## Econ9                           NA             NA
## Econ10                          NA  -0.0277452430
## Econ11                          NA             NA
## Econ12                          NA             NA
## Econ13                          NA             NA
## Econ14                          NA             NA
## Econ15                          NA             NA
## Econ16                          NA             NA
## Econ17                          NA             NA
## Econ18                          NA             NA
## Econ19                          NA             NA
## Econ1.lag1                      NA             NA
## Econ2.lag1                      NA             NA
## Econ3.lag1                      NA             NA
## Econ4.lag1                      NA   0.0501263396
## Econ5.lag1                      NA             NA
## Econ6.lag1                      NA             NA
## Econ7.lag1                      NA             NA
## Econ8.lag1            7.146399e-05             NA
## Econ9.lag1                      NA             NA
## Econ10.lag1                     NA   0.0614914114
## Econ11.lag1                     NA             NA
## Econ12.lag1                     NA             NA
```

```
## Econ13.lag1                 NA            NA
## Econ14.lag1                 NA            NA
## Econ15.lag1                 NA            NA
## Econ16.lag1                 NA            NA
## Econ17.lag1                 NA            NA
## Econ18.lag1                 NA            NA
## Econ19.lag1                 NA            NA
## Econ1.lag2                  NA            NA
## Econ2.lag2                  NA            NA
## Econ3.lag2                  NA            NA
## Econ4.lag2                  NA -0.0015280368
## Econ5.lag2                  NA            NA
## Econ6.lag2                  NA            NA
## Econ7.lag2                  NA            NA
## Econ8.lag2        1.458874e-04            NA
## Econ9.lag2        8.598864e-07            NA
## Econ10.lag2                 NA  0.0422727540
## Econ11.lag2                 NA            NA
## Econ12.lag2                 NA            NA
## Econ13.lag2                 NA            NA
## Econ14.lag2                 NA            NA
## Econ15.lag2                 NA            NA
## Econ16.lag2                 NA            NA
## Econ17.lag2                 NA            NA
## Econ18.lag2                 NA            NA
## Econ19.lag2                 NA            NA
## Econ1.lag3                  NA            NA
## Econ2.lag3                  NA            NA
## Econ3.lag3                  NA            NA
## Econ4.lag3                  NA -0.0747674715
## Econ5.lag3                  NA            NA
## Econ6.lag3                  NA            NA
## Econ7.lag3                  NA            NA
## Econ8.lag3                  NA            NA
## Econ9.lag3                  NA            NA
## Econ10.lag3                 NA -0.0671832859
## Econ11.lag3                 NA            NA
## Econ12.lag3                 NA            NA
## Econ13.lag3                 NA            NA
## Econ14.lag3       8.287425e-05            NA
## Econ15.lag3                 NA            NA
## Econ16.lag3                 NA            NA
## Econ17.lag3                 NA            NA
## Econ18.lag3                 NA            NA
## Econ19.lag3                 NA            NA
## Econ1.lag4                  NA            NA
## Econ2.lag4                  NA            NA
## Econ3.lag4                  NA            NA
## Econ4.lag4                  NA  0.0433173592
## Econ5.lag4                  NA            NA
## Econ6.lag4                  NA            NA
## Econ7.lag4                  NA            NA
## Econ8.lag4                  NA            NA
## Econ9.lag4                  NA            NA
```

```
## Econ10.lag4         6.445550e-02   0.1073044605
## Econ11.lag4                   NA             NA
## Econ12.lag4                   NA             NA
## Econ13.lag4                   NA             NA
## Econ14.lag4                   NA             NA
## Econ15.lag4                   NA             NA
## Econ16.lag4                   NA             NA
## Econ17.lag4                   NA             NA
## Econ18.lag4                   NA             NA
## Econ19.lag4                   NA             NA
```

While adaptive Lasso regression is known not to be generally a better performer than Ridge and Lasso, it has “Oracle” properties, in that it is somehow able to very well find out what variables are good predictors of the data.