Homework 8

2023-03-30

Homework 8

Modeling wastewater PH among regulated facilities in Chile

In this homework we are going to use PH data sampled from the wastewater of regulated facilites in Chile. The PH variable is stored in avg.value in the chilePH.csv dataset. There are 13 other possible predictors of PH listed in the comments below.

```
library(tidyverse)
library(glmnet)
library(caret)
chilePH <- read_csv("chilePHdata.csv")
options(scipen = 99, digits = 2)

## Remove missing values
(chilePH1 <- na.omit(chilePH))
```

```
## # A tibble: 13,173 x 16
##     ...1 period avg.val measur~1 disch~2 on.time all.p~3 meets~4 resam~5 resam~6
##    <dbl>  <dbl>   <dbl> <chr>    <chr>   <chr>   <chr>   <chr>   <chr>   <chr>
##  1     1 201603    7.75 AU       SI      SI      SI      SI      NO      Entreg~
##  2     2 201604    8.14 AU       SI      SI      SI      SI      NO      Entreg~
##  3     3 201605    7.97 AU       SI      SI      SI      SI      NO APL~ Entreg~
##  4     4 201606    7.65 AU       SI      SI      SI      SI      NO      Entreg~
##  5     5 201609    7.18 AU       SI      NO      SI      SI      NO APL~ Entreg~
##  6     6 201610    7.10 AU       SI      SI      SI      SI      NO APL~ Entreg~
##  7     7 201611    7.16 AU       SI      SI      SI      SI      NO APL~ Entreg~
##  8     8 201612    7.20 AU       SI      SI      SI      SI      NO APL~ Entreg~
##  9     9 201701    7.8  AC       SI      SI      SI      SI      NO APL~ NO APL~
## 10    10 201702    7.9  AC       SI      SI      SI      SI      NO APL~ NO APL~
## # ... with 13,163 more rows, 6 more variables: num.params.month <dbl>,
## #   num.riles.infrac <dbl>, num.noriles.infrac <dbl>, num.complaints <dbl>,
## #   num.permits <dbl>, plant.ID <dbl>, and abbreviated variable names
## #   1: measurement.type, 2: discharge, 3: all.params, 4: meets.frequency,
## #   5: resamples, 6: resample.report
```

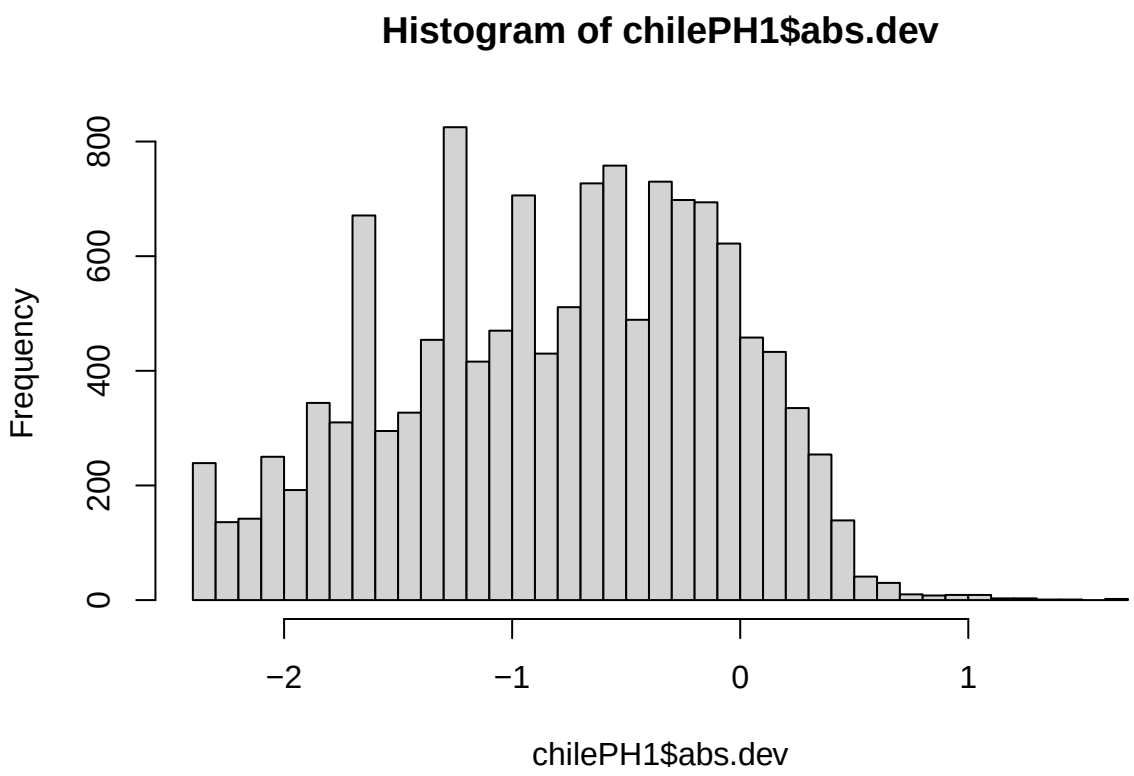
```
dim(chilePH1)
```

```
## [1] 13173    16
```

```
# Remove bad PH values
chilePH1 <- chilePH1 %>%
  filter(avg.val <= 14, avg.val >= 0)
dim(chilePH1)
```

```
## [1] 13172    16
```

```
# Calculate absolute deviation
chilePH1 <- chilePH1 %>%
  mutate(abs.dev = log(abs(avg.val - 7) + 0.1))
hist(chilePH1$abs.dev, breaks = "FD")
```



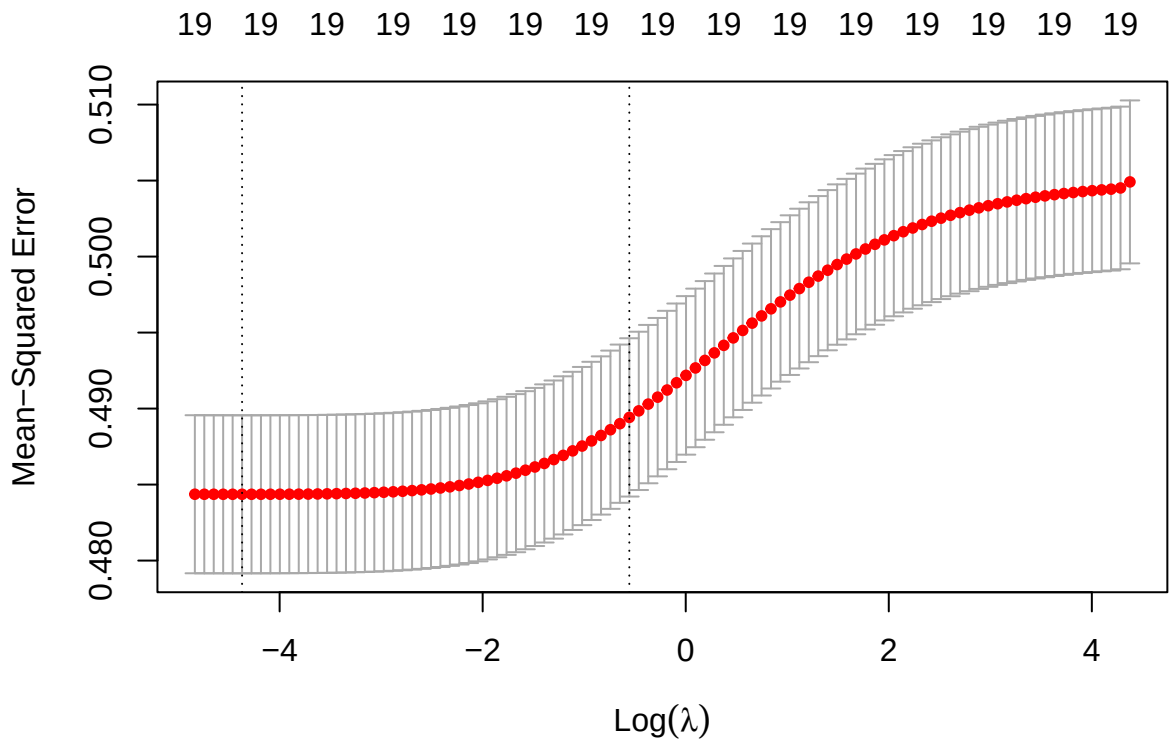

```
# Get regressor matrix
X <- chilePH1 %>%
  select(-abs.dev, -avg.val, -period, -discharge, -plant.ID)
X <- model.matrix( ~ ., data = X)[, -1]
y <- chilePH1 %>%
  select(abs.dev)
y <- unlist(y)


# period: Time period (yyyy-mm)
# avg.val: Average Ph value
# measurement.type: AC (continuous measurement), AU (lab sample)
# discharge: Report discharge (SI, NO, NA)
```

```
# on.time: Emissions reported on time (NO/No APLICA/SI)
# all.params: Reported all parameters required by regulation (SI/NO/NO APLICA)
# meets.frequency: Firm  meets the reporting frequency required by regulation (SI/NO/NO APLICA)
# resamples: Is a resampling report provided (NO/SI/NO APLICA)
# resample.report: Does the resampling report have the required paramters (Entrega parametros exigidos
# num.params.month: Number of parameters required to report each month
# num.riles.infrac: Number of previous water environmental infractions
# num.noriles.infrac: Number of previous non-water environmental infractions
# num.noriles.infrac: Number of citizen complaints related to the firm
# num.permits: Number of environmental permits
```

Part 1: Ridge regression

Use ridge regression (`cv.glmnet`) to predict the absolute deviation of PH from 7 based on any transformations you used in Homework 6. What is the optimal lambda selected using cross-validation? How does it change the estimates relative to ordinary least squares (OLS)?

```
model = cv.glmnet(x=X, y=y, alpha=0)
plot(model)
```




```
cv_mse_ridge <- model$cvm[model$lambda == model$lambda.min]
cv_mse_ridge
```

```
## [1] 0.48
```

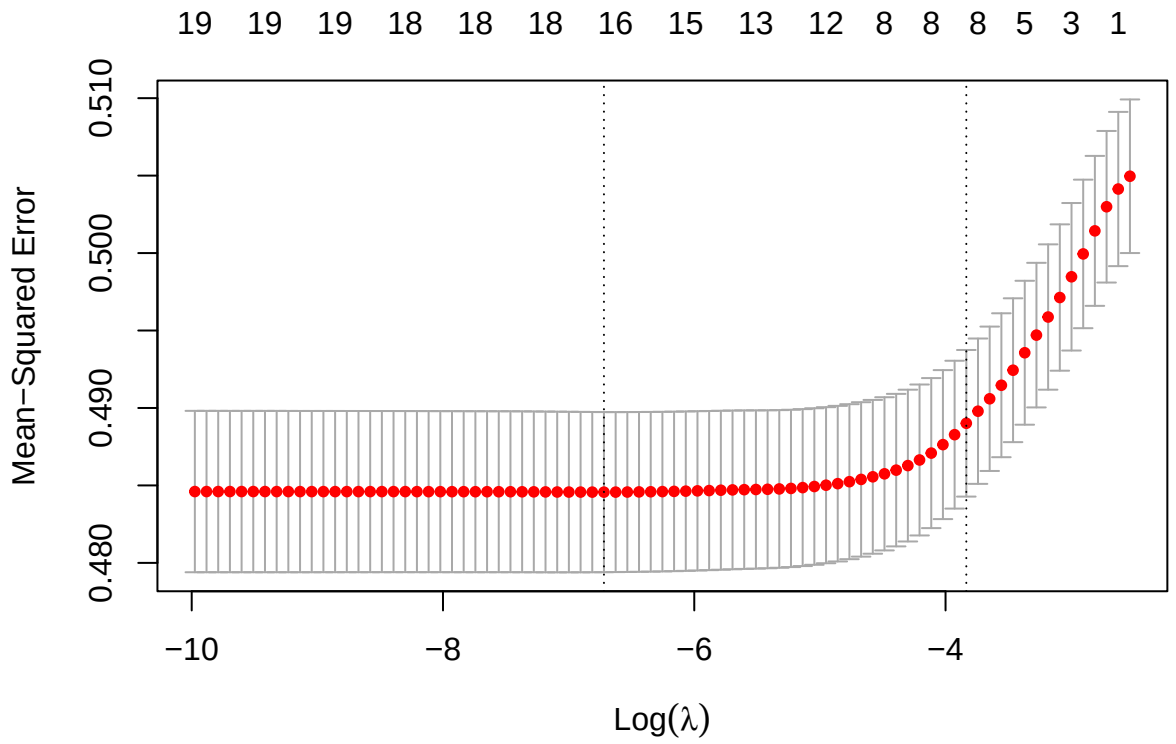
```
opt_lambda <- model$lambda.min
cat("Optimal Lambda:", opt_lambda)
```

```
## Optimal Lambda: 0.013
```

Part 2: Lasso regression

Repeat part 1 for the Lasso. Are any variables removed from the model?

```
model1 = cv.glmnet(x=X, y=y, alpha=1.0)
plot(model1)
```




```
coef(model1)
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                                    s1
## (Intercept)                -0.493900
## ...1                       -0.000013
## measurement.typeAU          0.037919
## on.timeNO APLICA            .
## on.timeSI                  -0.064319
## all.paramsNO APLICA         .
## all.paramsSI               -0.014605
```

```
## meets.frequencyNO APLICA  .
## meets.frequencySI        -0.043170
## resamplesNO APLICA       -0.136761
## resamplesSI               .
## resample.reportNO         .
## resample.reportNO APLICA -0.003588
## resample.reportNULL       .
## resample.reportSI         .
## num.params.month          .
## num.riles.infrac          .
## num.noriles.infrac        .
## num.complaints            .
## num.permits               0.006114
```

```
cv_mse_lasso <- model1$cvm[model$lambda == model1$lambda.min]
cv_mse_lasso
```

```
## numeric(0)
```

```
opt_lambda1 <- model1$lambda.min
cat("Optimal Lambda:", opt_lambda1)
```

```
## Optimal Lambda: 0.0012
```

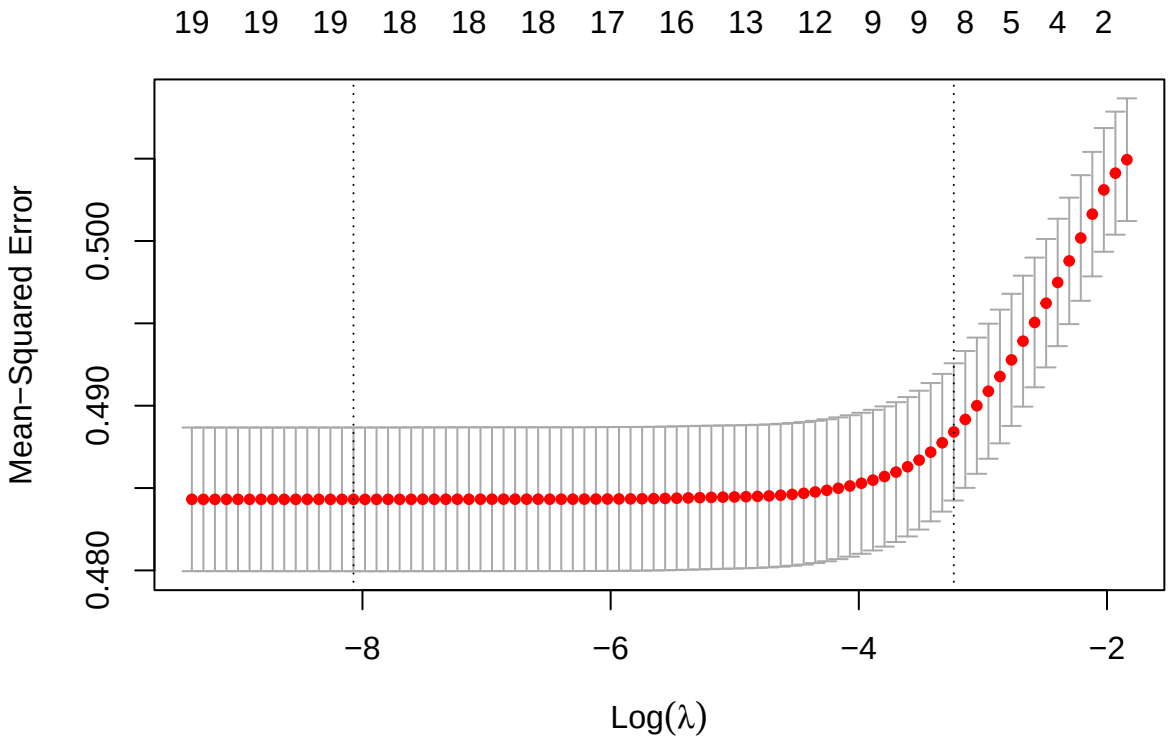
Yes 11 variables have been removed from the model.
Part 3: Elastic net

In ridge we use `alpha=0` for ridge and `alpha=1` for lasso.

Elasticnet refers to blending the ridge and lasso loss, and can take any value between 0.0 and 1.0.

Predict PH using the elastic net with $\alpha = 1/2$. How does the result compare to ridge or Lasso?

```
model2 = cv.glmnet(x=X, y=y, alpha=0.5)
plot(model2)
```




```
cv_mse_elas <- model2$cvm[model2$lambda == model2$lambda.min]
cv_mse_elas
```

```
## [1] 0.48
```

```
opt_lambda2 <- model2$lambda.min
cat("Optimal Lambda:", opt_lambda2)
```

```
## Optimal Lambda: 0.00031
```

The regularization in ridge model seems to be the highest in Ridge model. This means that the overfitting in ridge would be the least. ## Part 4: Hyperparameter selection using cross-validation

Use cross-validation to select the hyper parameters for Ridge, Lasso, and Elastic Net (both lambda and alpha). You will find the `caret` package useful here.

```
ctrl <- trainControl(method = "cv", number = 10)

rdg <- train(x = X, y = y, method = "ridge", tuneLength = 10, trControl = ctrl)
ridge$bestTune

las <- train(x = X, y = y, method = "lasso", tuneLength = 10, trControl = ctrl)
lasso$bestTune

elas <- train(x = X, y = y, method = "glmnet", tuneLength = 10, trControl = ctrl)
elas$bestTune
```

Part 5: Test set comparison

Instead of using cross-validation twice as in Part 6, set aside 20% of the data as a test set. Use the remaining data to select the hyperparameters of each model, then make predictions on the test set.

```
chilePHn = chilePH1 %>%
  select(-abs.dev, -avg.val, -period, -discharge, -plant.ID, -on.time)
chilePHn$y = y

train <- head(chilePHn, 0.8*length(y))
test <- tail(chilePHn, 0.2*length(y))


fmodel = train(y~., data=train ,method = "glmnet")

pred_y <- predict(fmodel, test)

print(fmodel)

mean(test$y-pred_y)
```