Machine Learning
Master in Big Data Management
Lecture 3

Giuseppe Ragusa
gragusa@luiss.it

Date
LUISS – Business school

Features Selection and Shrinkage Methods

The Bias-Variance trade off

Last time we have seen that you may want to maximize the predictive power of a model by finding a good compromise between its flexibility and simplicity.

- The higher is the flexibility of the model, the lower the bias, but the higher the variance.
- The more a model is simpler, the lower the variance but the higher the bias.

During this class we will show one of the possible model to find the variables that maximize the out-of-sample prediction error.

Linear Model: Review

$$y_i = \beta_0 + \beta_1 x_{1,i} + ... + \beta_k x_{k,i} + \epsilon_i$$

In a more compact way:

$$\hat{Y} = X\beta$$

The OLS estimators solve:

$$\hat{\beta}^{OLS} = \underset{\hat{\beta}}{\text{argmin}} \sum_{i=1}^{n} (y_i - X_i\beta)^2$$

Linear Model: Review

Taking the linear projection of Y onto X:

$$\frac{dRSS(\beta)}{\beta} = 0 \rightarrow \beta^{\hat{O}LS} = (X'X)^{-1}X'Y$$

The general formula of the standard error:

$$Var(\beta^{\hat{O}LS}) = (X^T X)^{-1}(\sum_{i=1}^{n} \epsilon_i^2 x_i^T x_i)(X^T X)^{-1} \qquad (-4)$$

And under the assumption of homoskedasticity:

$$Var(\hat{\beta}^{OLS}) = \sigma^2(X'X)^{-1}$$

What happen if some explanatory variables are highly correlated each other?

Ridge Regression

To avoid singularity problems of the (X'X) matrix you can add a ridge to the matrix:

$$\hat{\beta}^{RR} = (X'X + \lambda I)X'Y$$

Where

$$I = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

and $\lambda > 0$

Note: for any $\lambda > 0$ (X'X) is not singular, why?

Ridge Regression

This is equivalent to solve

$$\min_{\hat{\beta}} \sum_{i=1}^{n} (y_i - X_i\beta)^2$$

subject to

$$\sum_{j=1}^{p} \beta_j^2 < C$$

with $C > 0$

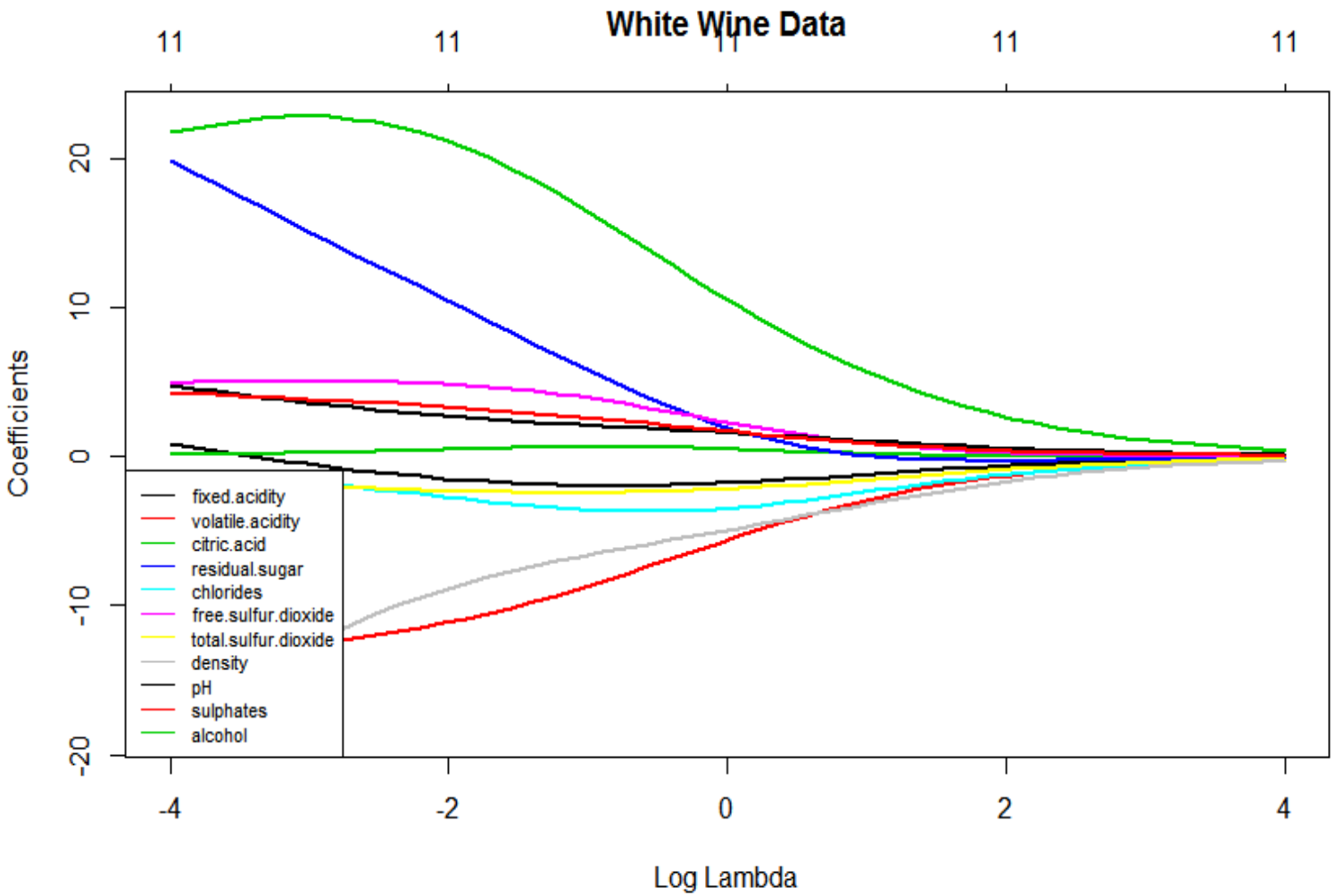
Ridge Regression

Which is equivalent to...

$$\hat{\beta}^{RR} = \underset{\hat{\beta}}{\text{argmin}} \sum_{i=1}^{n}(y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- If $\lambda \to \infty$ then $\hat{\beta}^{RR} \to 0$
- If $\lambda = 0$ then $\hat{\beta}^{RR} = \hat{\beta}^{OLS}$

Wine Dataset: Example



Ridge Regression

Ridge Regression was born to avoid problems of non-singularity. But in addition, it also reduces the variance of the parameters.

Recalling the formula of the variance under homoskedastic error for ridge regression now:

$$Var(\hat{\beta}^{RR}) = \sigma^2(X'X + \lambda I)^{-1}$$

So the higher λ the lower the variance.

But what about the bias?

Ridge Regression

If the gain in terms of variance is higher then the loss in terms of bias, then we should expect an higher predictive power of the model.

We can choose among a set of models with different values of lambda

But...

- What λ represents?
- How do I choose λ?

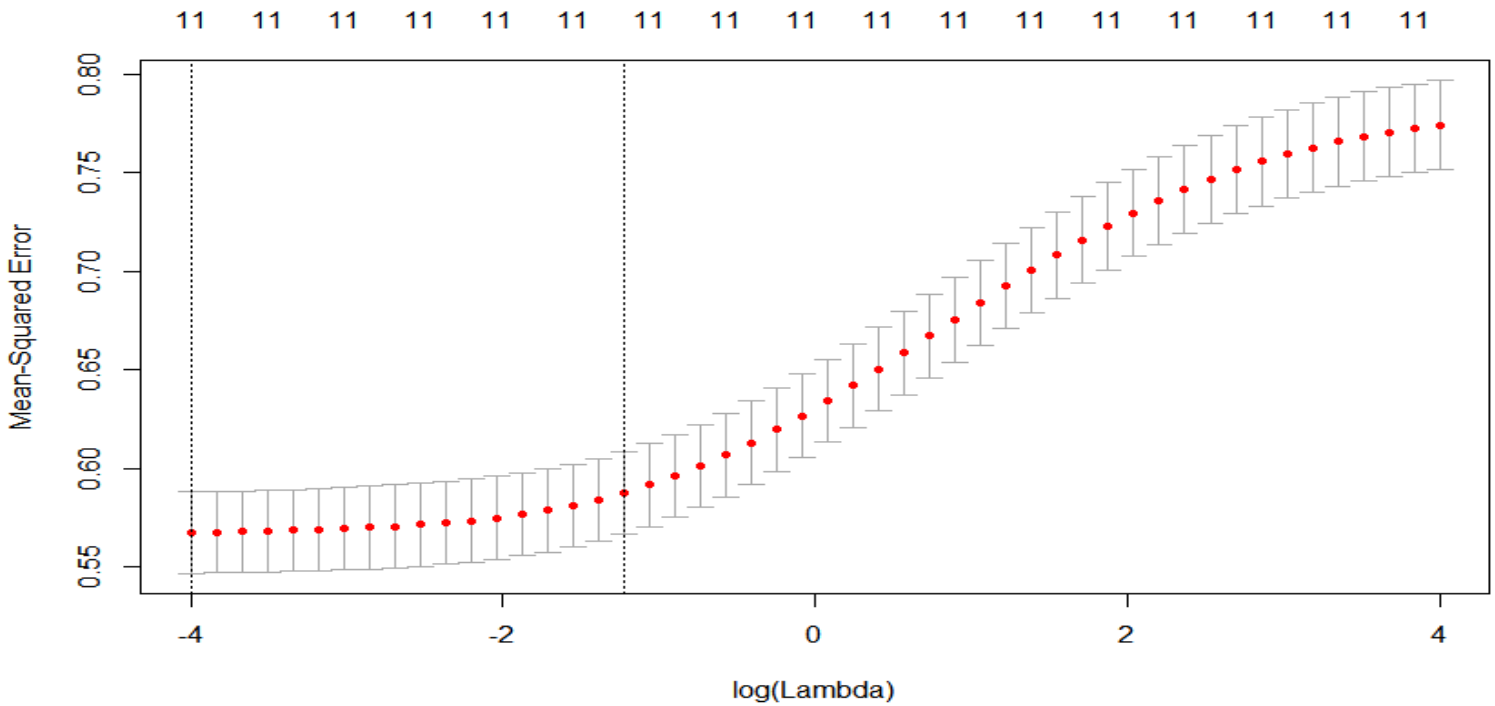
How do we choose lambda?

1. Find the model with smallest mean squared error
2. Find the model with highest in-sample predictive powe
3. Find the model with highest R squared
4. Find the model with lowest out-of-sample mean squared error
5. As usual... We do not care

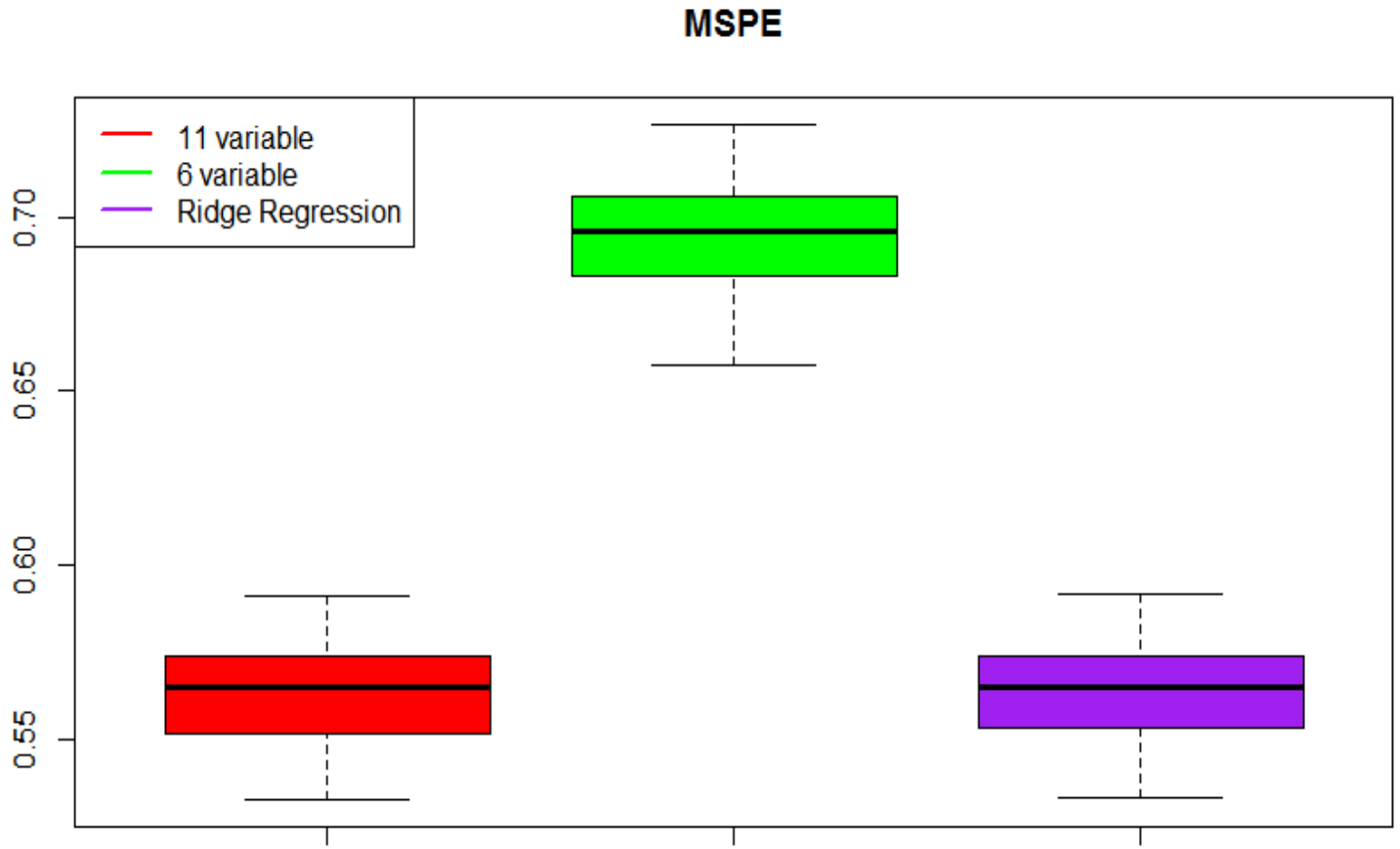
Cross Validation!

```r
## Cross validation

library(glmnet)
lambdas.rr <- exp(seq(-4, 4, length=50))
k <- 10
rr.cv <- cv.glmnet(x=as.matrix(xm), y=y, lambda=lambdas.rr, nfolds=k, alpha=0)
fit <- glmnet(x=as.matrix(xm), y=y, lambda=rr.cv$lambda.min, alpha=0)



plot(rr.cv)
```

Wine Data: Results



Wine Data: Results



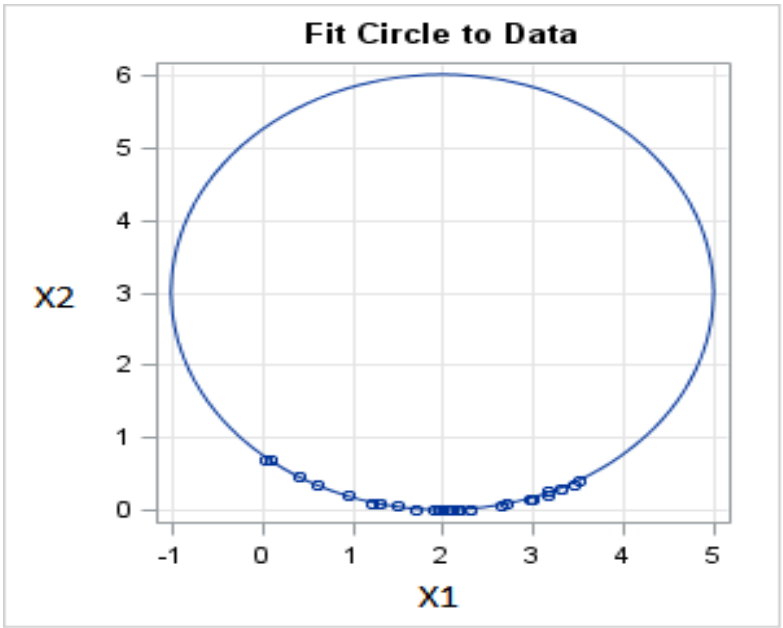
But still... What does lambda mean?

Ridge regression **shrink** parameters to zero, but we still have the same number of variables, so nothing change?... Isn't it?

How many **effective parameters** are we estimating?

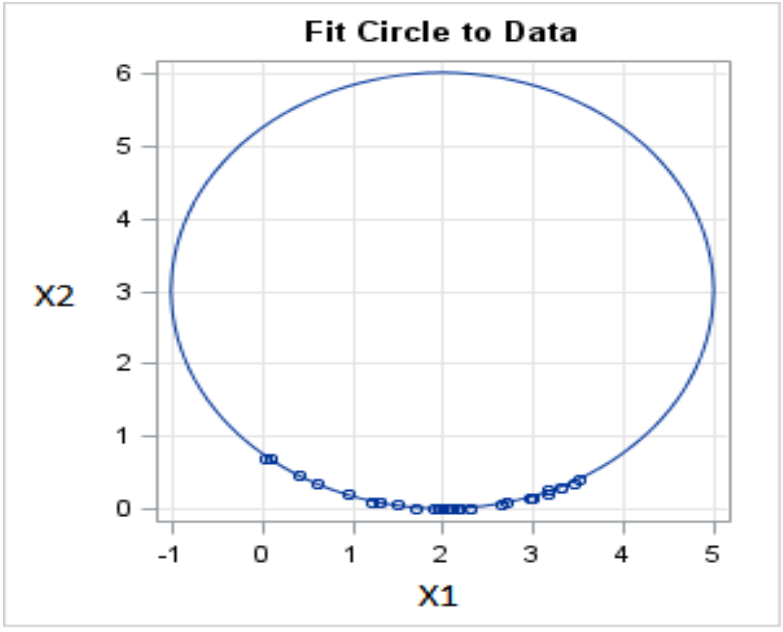
Digression: two dimensional case

How many parameter do you have to choose if you constrain your variables to be on the circle boundary?



Digression: two dimensional case

How many parameter do you have to choose if you constrain your variables to be on the circle boundary?



Only one: the angle of the circle!

Effective degrees of freedom

Effective degrees of freedom are the number of parameters to be estimated. In case of OLS regression this number is equal to p, where p is the number of variables plus one (the intercept).

In general:

$$\hat{Y} = SY$$

with S the regression matrix which does not depend on Y, Y are the actual values of the output variables and $\hat{Y}$ the predictions.

Effective degrees of freedom

The general result:

$$EDF = \frac{\sum_{i=1}^{n} cov(\hat{y}_i, y_i)}{\sigma^2}.$$

But it can be shown that

$$EDF = trace(S) = \sum_{i=1}^{n} S_{i,i}$$

Effective Degrees of Freedom

For linear regression:

$$S = X(X'X)^{-1}X'$$

For ridge regression:

$$S = X(X'X + \lambda I)^{-1}X'$$

What is the trace?

Effective Degrees of Freedom

Using the Singular Value Decomposition

$$X = UDV'$$

with

U'U = V'V = I and

$$D = \begin{bmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & 0 & \dots & d_j \end{bmatrix}$$

$$EDF = \sum_1^j \frac{d_j}{d_j + \lambda}$$

Alternative method: Lasso

In a compact form:

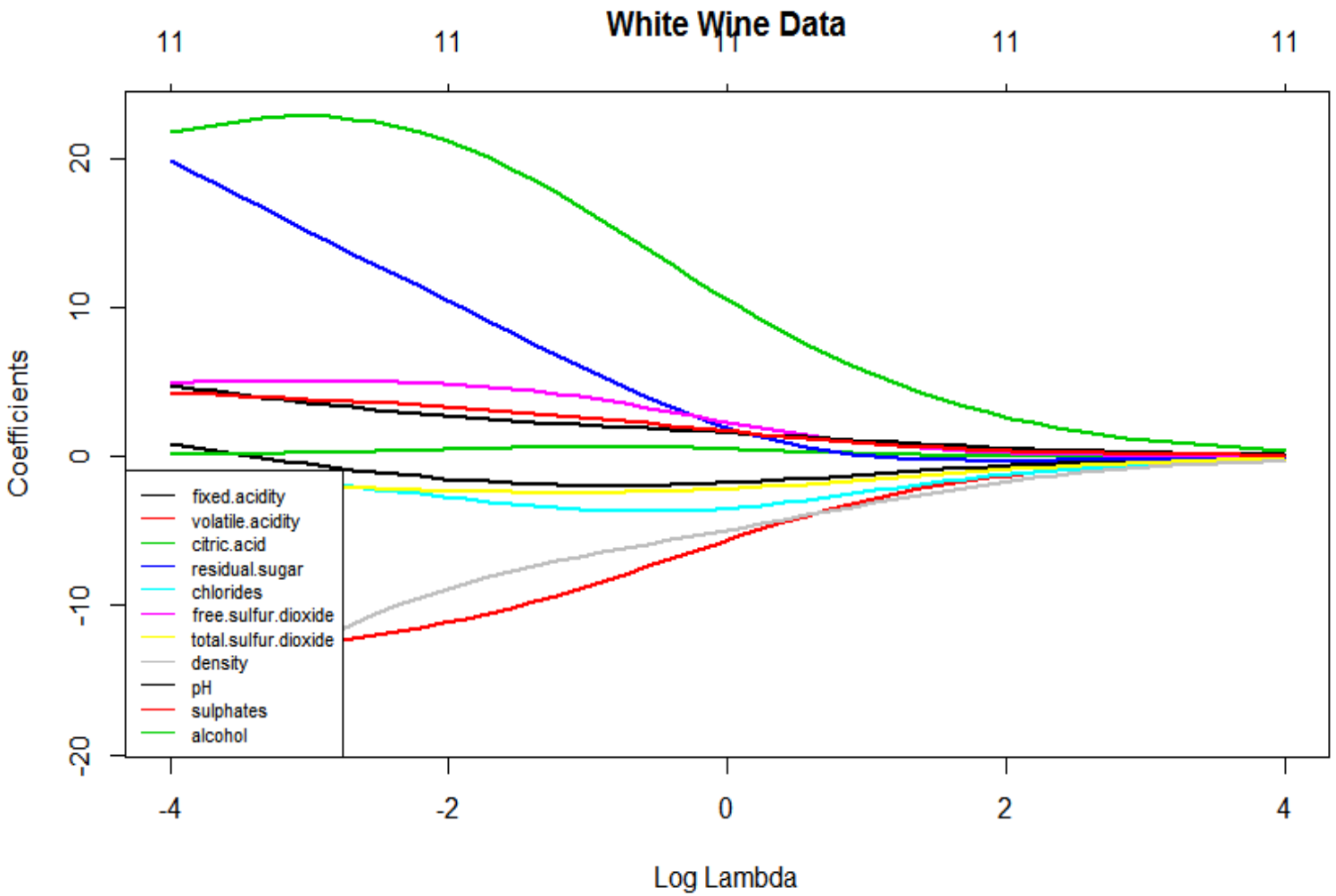
$$\beta^{Ridge} = \underset{\beta}{\text{argmin}}(Y - X\beta)'(Y - X\beta) + \lambda\beta^2$$

But an alternative method may be...

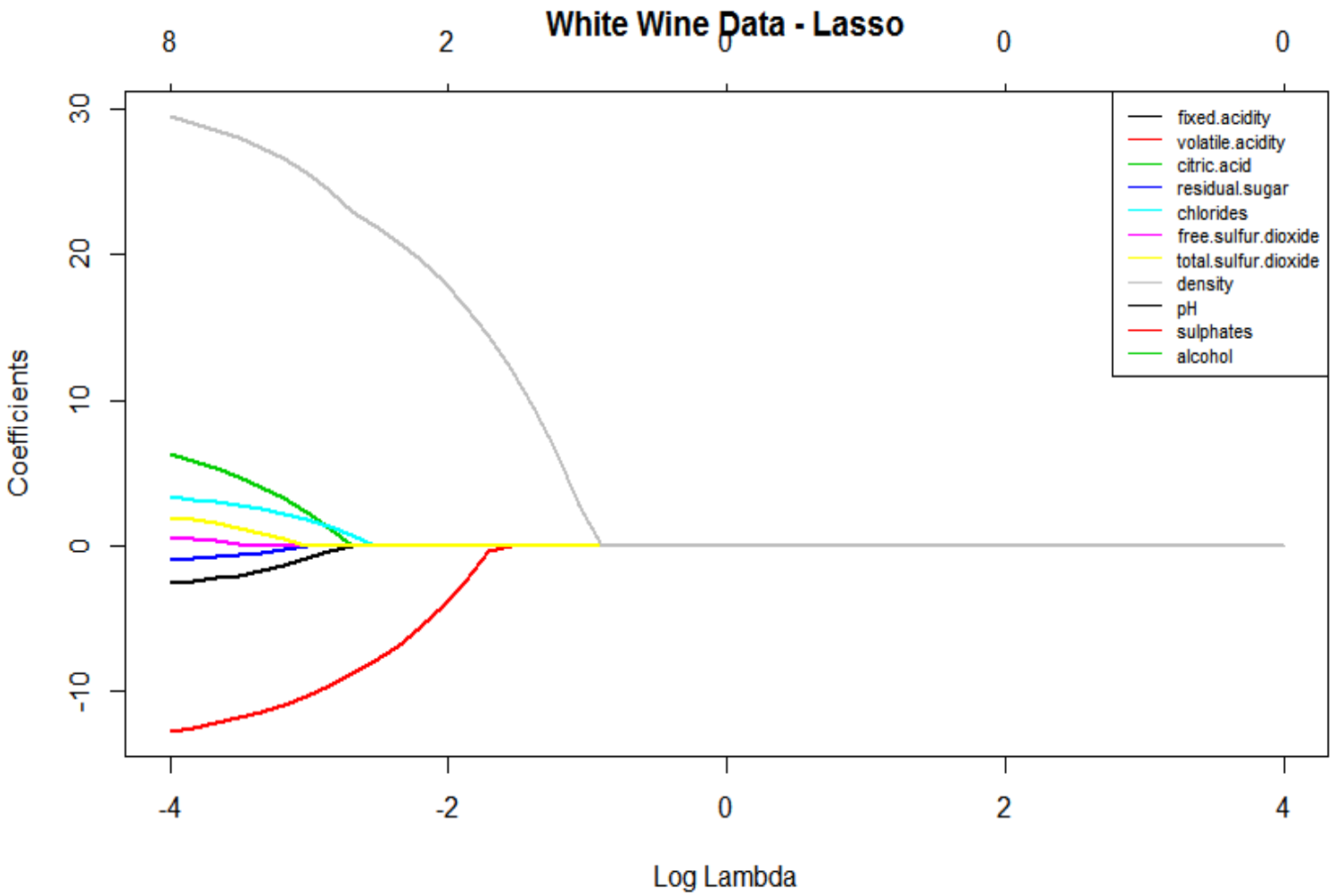
$$\beta^{Lasso} = \underset{\beta}{\text{argmin}}(Y - X\beta)'(Y - X\beta) + \lambda|\beta|$$

So...Which one?

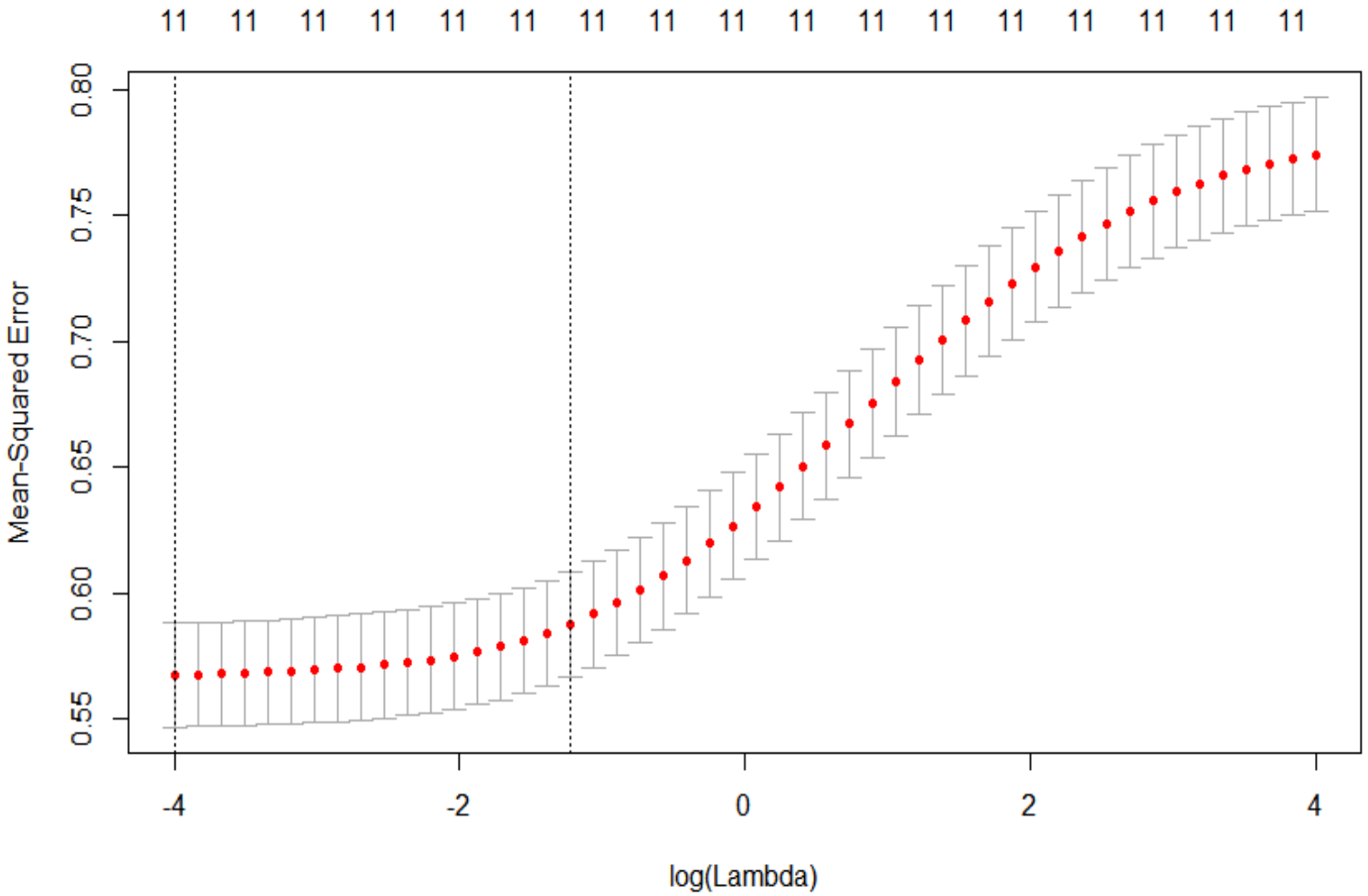
Wine Dataset: Shrinkage with Ridge



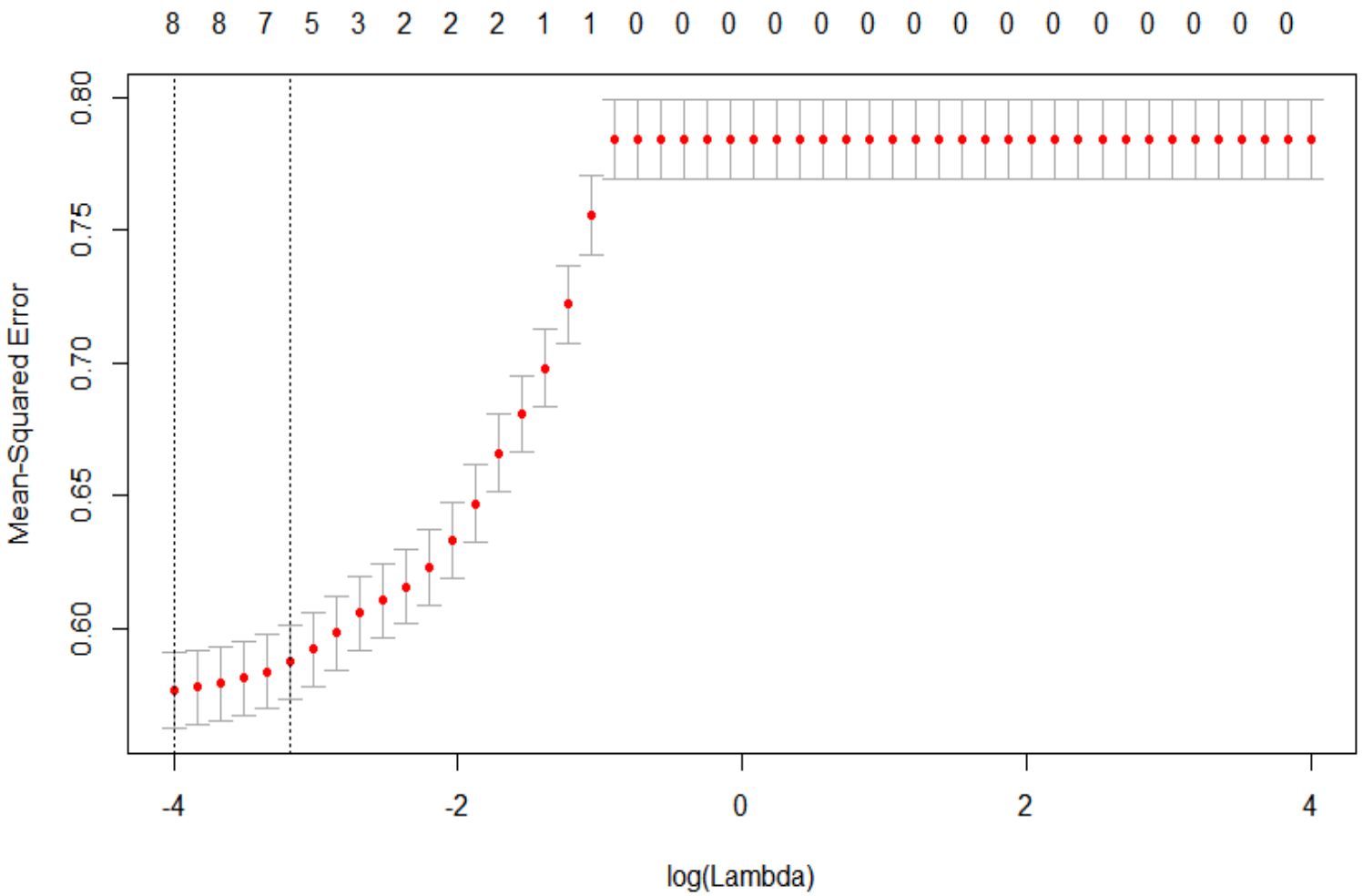
Wine Dataset: Shrinkage with Lasso



Wine Dataset: MSPE with Ridge



Wine Dataset: MSPE with Lasso



Summary: Difference

- Lasso shrink some coefficients to zero after a certain treshold for lambda (features selection).
- Ridge never shrinks coefficients completely to zero.

Summary: Difference

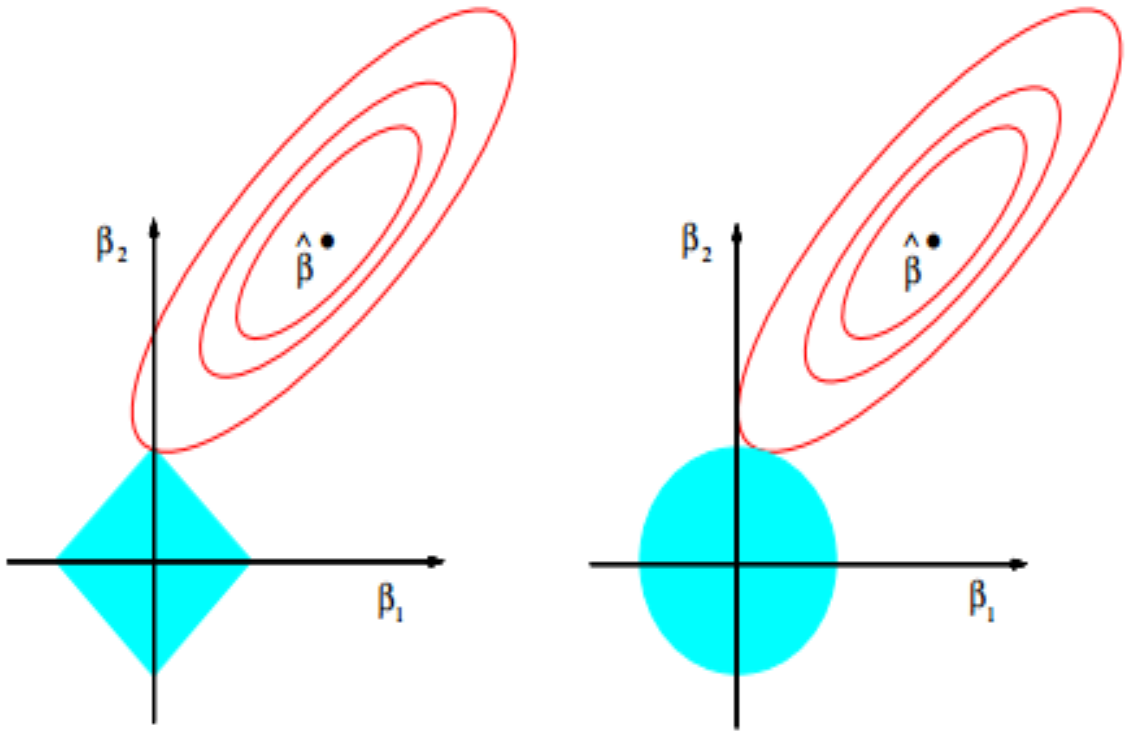


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Lasso: features Selection

- Lasso picks at most as many covariates as the number of observations, if available (no overidentified models).
- When covariates are highly correlated Lasso will tend to pick just one of them.

Could you explain why?

Between the two: Elastic Net

You might prefer to weight the squared and linear penalty:

$$\beta^{ElNet} = \underset{\beta}{\text{argmin}}(Y - X\beta)'(Y - X\beta) + \lambda(\alpha|\beta| + \frac{(1-\alpha)}{2}\beta^2)$$

With $\alpha \in [0, 1]$

Why do we take $(1 - \alpha)/2$?

Shrinkage methods in practice

The idea of shrinkage methods is to find the value of lambda that maximize the **out of sample** prediction error. To compute the MSPE we use **cross validation**.

In R we will use the package glmnet , availble on CRAN.

An example: white wine data

We are still interested in predicting what is the expected quality of a wine controlling for chemical attributes.

Previously we showed the result of the analysis, here we provide further details on some issues to be careful about.

The data set is available on the web page.

Prepare the analysis

```
wine.white <- read.table("./data/wine.white.txt")
library(glmnet)
lambdas.rr <- exp(seq(-4, 4, length=50))
```

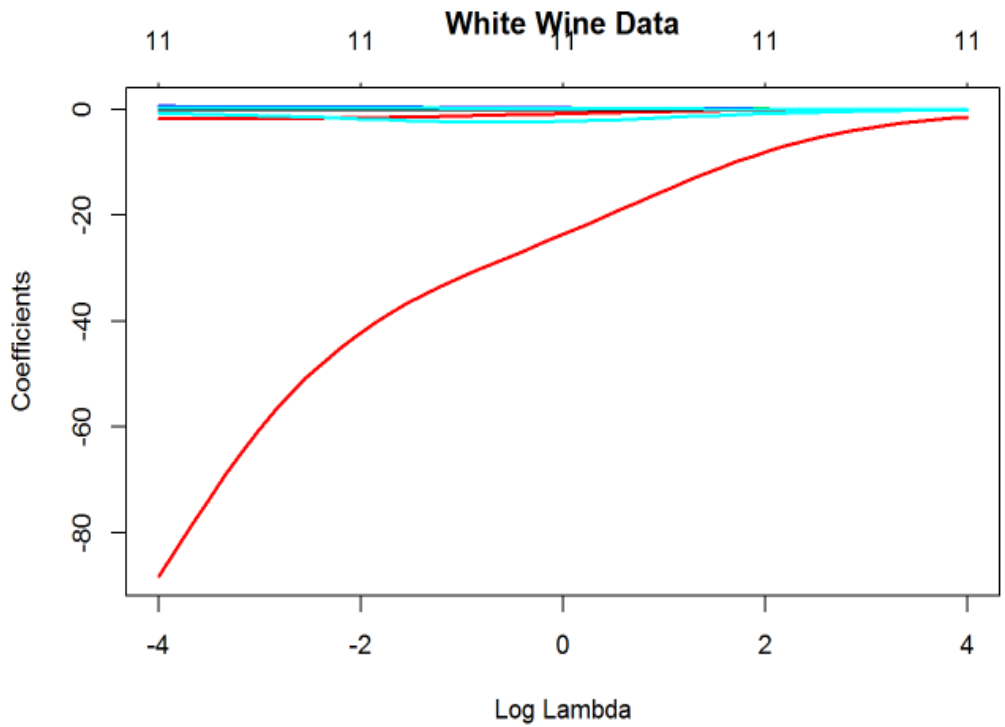
Ridge Regression: example

Build a ridge regression for each value of lambda

```
y <- wine.white$quality

## Construct the covariate matrix

xm <- wine.white[,-12]
fit <- glmnet(x=as.matrix(xm), y=y, lambda=lambdas.rr, alpha=0)


## Make a plot of coefficients

plot(fit, xvar='lambda', lwd=2, main= "White Wine Data")
```

Note: for lasso set alpha $= 1$, while for elastic net set alpha $=$ c, where c is any value you want between 0 and 1.

Ridge Regression: example



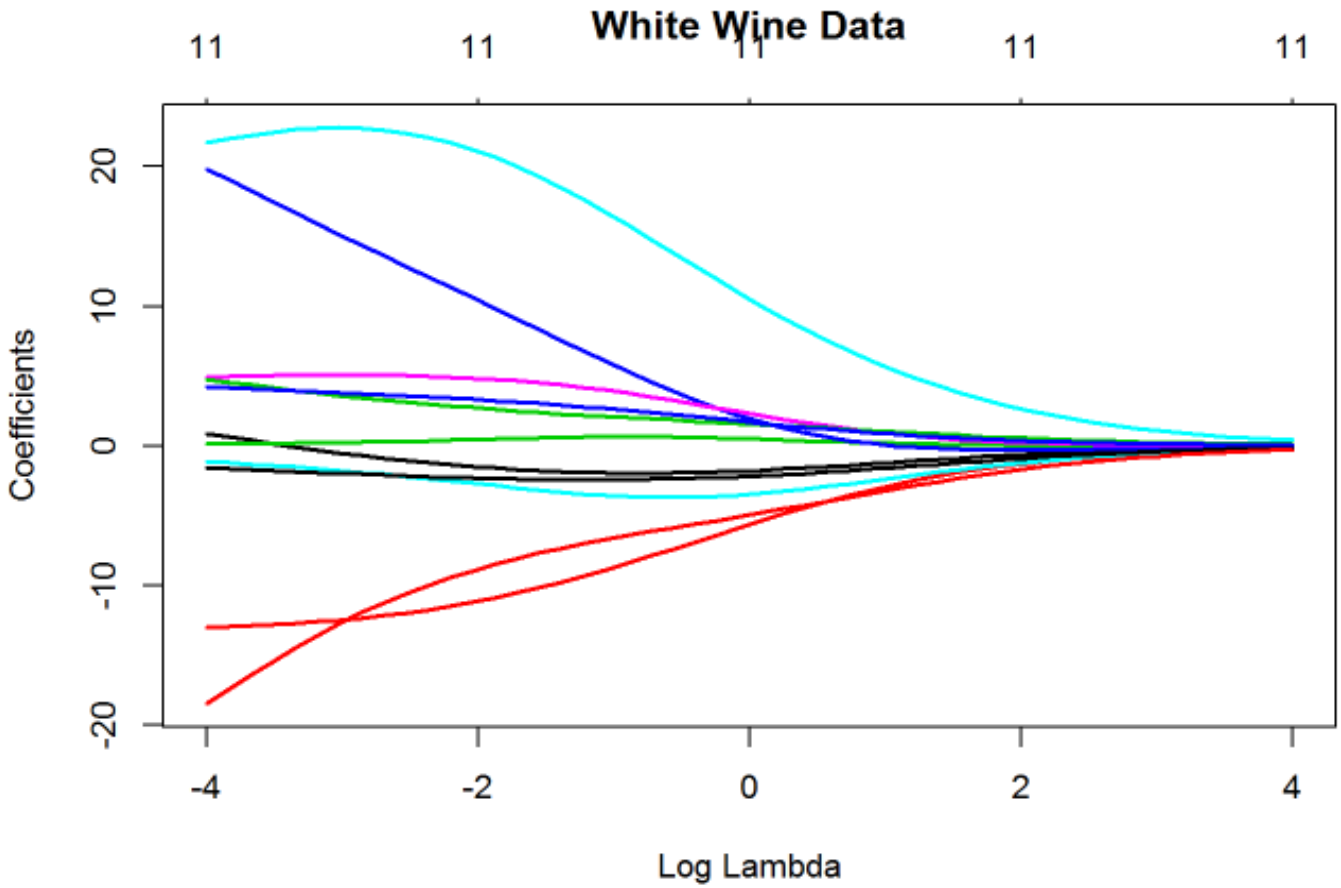
Something wrong?

Ridge Regression: example

Remember that the penalty on beta may depend on the unit of measure of the input variable. So you need first to standardize all the covariates!

```
# Standardize
p <- ncol(xm)
n <- nrow(xm)
xmn <- scale(scale(xm), center=FALSE, scale=rep(sqrt(n-1), p))

fit <- glmnet(x=as.matrix(xmn), y=y, lambda=lambdas.rr, alpha=0)
plot(fit, xvar='lambda', lwd=2, main= "White Wine Data")
```

Ridge Regression



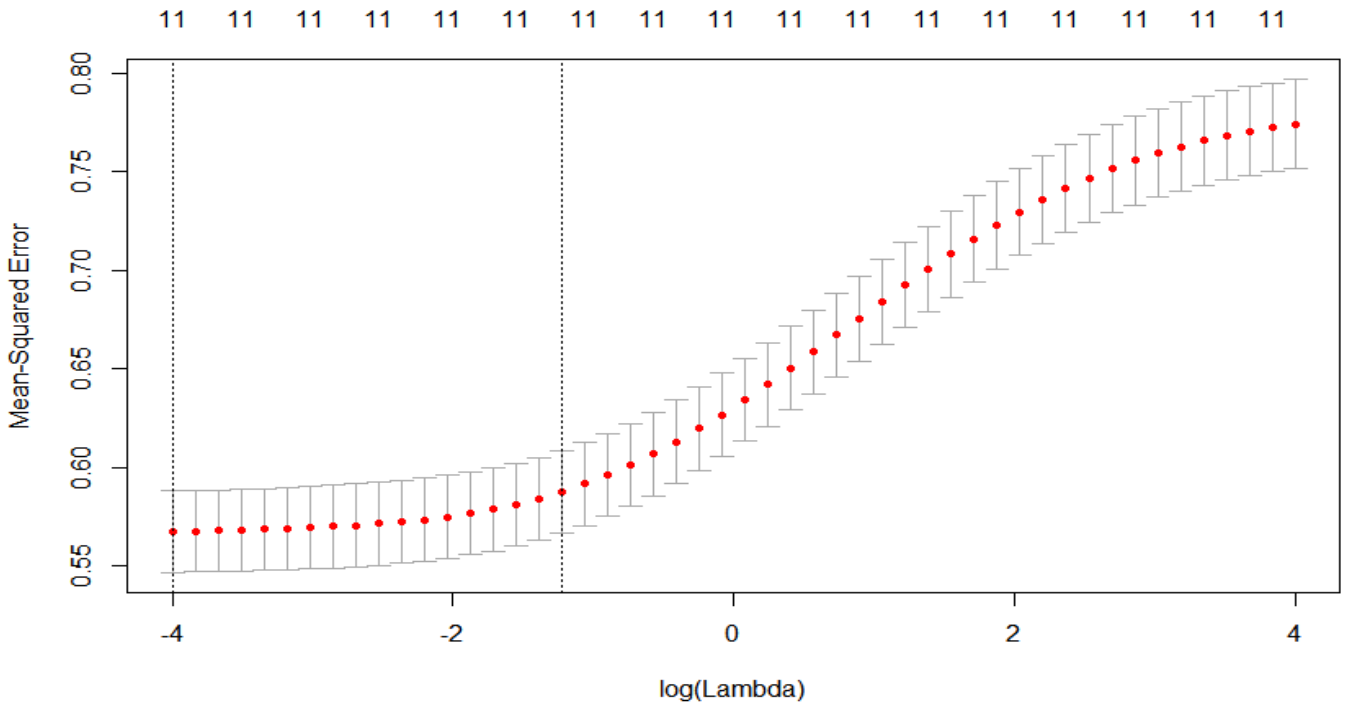
Find the best lambda with CV

Do cross validation and fit the value with the smallest lambda.

```
## Cross validation

library(glmnet)
lambdas.rr <- exp(seq(-4, 4, length=50))
k <- 10
rr.cv <- cv.glmnet(x=as.matrix(xmn), y=y, lambda=lambdas.rr, nfolds=k, alpha=0)
fit <- glmnet(x=as.matrix(xmn), y=y, lambda=rr.cv$lambda.min, alpha=0)



plot(rr.cv)
```

Note: an alternative way is to pick rr.cv$lambda.1se
Could you explain why?

Find the best lambda with CV



Exercise

We are interested in predicting the level of alchol consumption during the weekend for students, controlling for many social and academic indicators. Some of them are the average grades for three years, the income of the family, the age, etc. In total we have 32 variables, but we want to find just the ones most correlated with alchol consumption.

Which model would you use?

Exercise

Do the following steps:

1. Download the student text file
2. Explore the variables and construct two different linear models you might think are meaningful
3. Report the interpretation of your coefficients

The dependent variable in the regression is Walc.

Exercise

Now we want to use our new powerful knowledge to find the coefficients.

1. Construct a sequence of number of size 50, from 10^{-4} to 10^{4}
2. Standardize the covariates
3. Open in the library glmnet
4. Use cross validation to choose the best lambda
5. Construct a lasso with the lambda with minimum error

Hint: for the first part of the exercise you might want to use the code provided (either on the website or on the next slide).

Remember to set alpha $= 1$.

Exercise: Code to use

```
## Expand the multivel factors to 0/1 dummies

xm <- model.matrix(~. ,data=student[,-27])[,-1]
y <- student[,27]

## Use this functions to standardize

standard_for_dummy <- function (k){ if (length(k[!duplicated(k)])==2)
{ return(1)}
return(sd(k))                                                    }

sd.tr <- apply(xm, 2, standard_for_dummy)

mu_for_dummy <- function (k){ if (length(k[!duplicated(k)])==2)
{ return(0.5)}
mean(k)       }

mu.tr <-apply(xm, 2, mu_for_dummy)

## New covariate matrix

xmn <- scale(xm, center = mu.tr, scale=sd.tr)

## Set your lambda

lambdas.rr <- exp(seq(-4, 4, length=50))
```

Exercise

Finally:

1. Cross Validate also the linear models using the skeleton for Cross Validation provided
2. Assess which model is the best
3. Do you think this is the correct way to compare the three models?

Optional:

1. Use fit$beta to see the coefficients actually used by the best lasso.
2. Question: do you think it is a proper way to assess causality between variables? Explain why yes or no.

Exercise: Solution 1st part

```
reg1 <- lm(Walc~school, data=student)
reg2 <- lm(Walc~ school + age + health + school*health , data=student)

summary(reg1)
```

```
##
## Call:
## lm(formula = Walc ~ school, data = student)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5217 -1.2607 -0.2607  0.7393  2.7393
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.26074    0.06888  32.821   <2e-16 ***
## schoolMS     0.26099    0.20184   1.293    0.197
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.287 on 393 degrees of freedom
## Multiple R-squared:  0.004236,   Adjusted R-squared:  0.001703
## F-statistic: 1.672 on 1 and 393 DF,  p-value: 0.1968
```

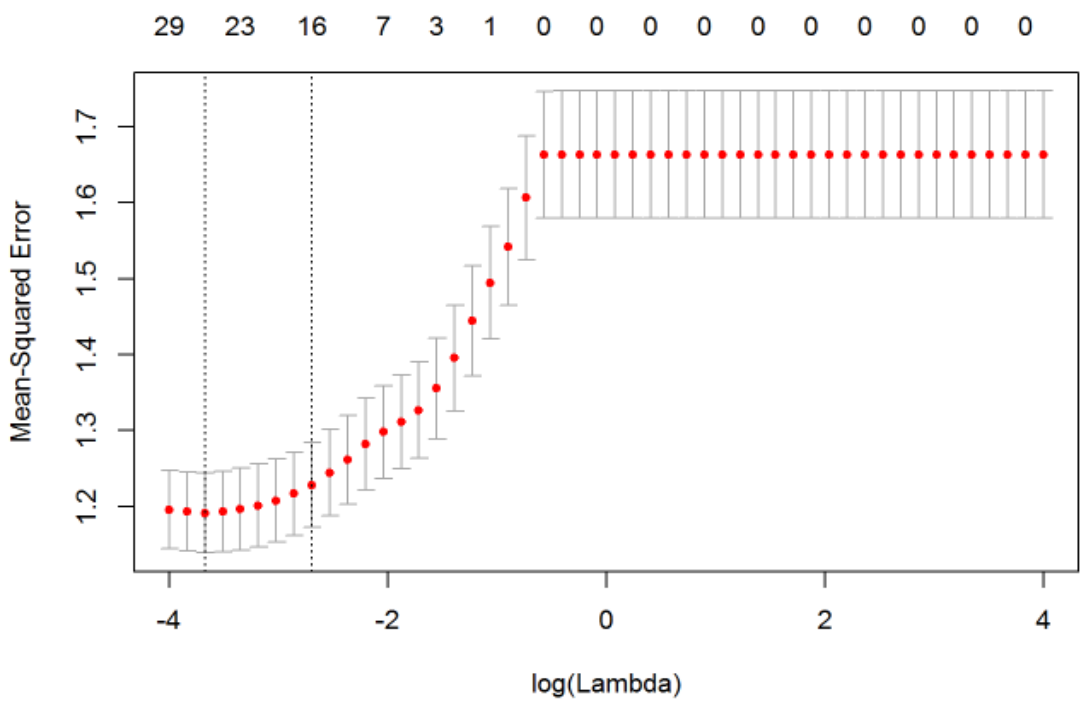
Exercise: Solution 1st part

```
summary(reg2)
```

```
##
## Call:
## lm(formula = Walc ~ school + age + health + school * health,
##     data = student)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8027 -1.2238 -0.2779  0.8379  3.1672
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.001606   0.938437  -0.002   0.9986
## schoolMS         0.225423   0.520777   0.433   0.6654
## age              0.115782   0.054790   2.113   0.0352 *
## health           0.097726   0.049720   1.966   0.0501 .
## schoolMS:health -0.035411   0.140793  -0.252   0.8016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.279 on 390 degrees of freedom
## Multiple R-squared:  0.02451,    Adjusted R-squared:  0.0145
## F-statistic: 2.449 on 4 and 390 DF,  p-value: 0.04577
```

Exercise: Solution 2nd part

```
k <- 10

lr.cv <- cv.glmnet(x=as.matrix(xmn), y=student[,27], lambda=lambdas.rr, nfolds=k, alpha=1)


fit <- glmnet(x=as.matrix(xmn), y=y, lambda=lr.cv$lambda.min, alpha=1)
plot(lr.cv)
```



Exercise: Solution 3rd part

```
set.seed(123)

  n <- length(y)
  k <- 10

  ii <- sample(rep(1:k, length= n))

  pr.1 <- pr.2 <- rep(NA, length(y))
  for (j in 1:k){
    hold <- (ii == j)
    train <- (ii != j)


  reg1 <- lm(Walc~school, data=student[train,])
  reg2 <- lm(Walc~ school + age + health + school*health , data=student[train,])

   pr.1[hold] <- predict(reg1, newdata=student[hold,])
    pr.2[hold] <- predict(reg2, newdata=student[hold,])

  }

mspe1 <- mean((pr.1-y)^2)
mspe2 <- mean((pr.2-y)^2)

print(cbind(mspe1, mspe2))
```

```
##         mspe1    mspe2
## [1,] 1.655833 1.650803
```

Exercise: Solution 3rd part

So... The MSPE for the linear regressions is 1.65 for both, while for lasso it is 1.2 , correct?

Isnt't it?

Exercise: Solution 3rd part

```
set.seed(123)

  n <- length(y)
  k <- 10

  ii <- sample(rep(1:k, length= n))

  pr.1 <- pr.2 <- pr.l <- rep(NA, length(y))
  for (j in 1:k){
    hold <- (ii == j)
    train <- (ii != j)
    xx.tr <- xmn[train,]
    y.tr <-  y[train]
    xx.te <- xmn[hold,]

  reg1 <- lm(Walc~school, data=student[train,])
  reg2 <- lm(Walc~ school + age + health + school*health , data=student[train,])



    ### find the best ridge on the TRAINING SET!
    lr.cv <- cv.glmnet(x=as.matrix(xx.tr), y=y.tr, lambda=lambdas.rr, nfolds=k, alpha=1)


fit <- glmnet(x=as.matrix(xx.tr), y=y.tr, lambda=lr.cv$lambda.min, alpha=1)

    pr.1[hold] <- predict(reg1, newdata=student[hold,])
    pr.2[hold] <- predict(reg2, newdata=student[hold,])

    pr.l[hold] <- predict(fit, newx=xx.te)
  }

mspe1 <- mean((pr.1-y)^2)
mspe2 <- mean((pr.2-y)^2)
mspe.Lasso <- mean((pr.l-y)^2)
print(cbind(mspe1, mspe2, mspe.Lasso))
```

```
##          mspe1    mspe2 mspe.Lasso
## [1,] 1.655833 1.650803   1.184304
```

Exercise: Solution 3rd part

Almost correct...in this case...but...

Why did we have to cross validate twice the lasso regression?

The right and the wrong way to do cross validation

The choice of lambda is done by cross validating the whole data set and THEN using the minimum lambda for constructing the regression. By doing so the regression becomes:

$$\hat{Y} = S_\lambda Y$$

Where S_λ is chosen looking at in-sample observations.

But to have an unbiased estimator we need always out-of-sample information.

The right and the wrong way to do cross validation

The correct way of doing cross validation is:

- Divide the data set in k folds
- Find the tuning parameters (λ) on the k-1 folds
- Train the model on the k-1 folds
- Compute the MSPE on the remaining fold
- Repeat the process k times and pick the mean of the MSPE

Every time you cross-validate you must cross validate everything, also the way you select the model!

Optional Question: Solution

No, it is not a proper way to find causal relationship among variables. In fact if some variables are highly correlated each other just one or few of these variables will enter in the regression, but this does not mean that they have a causal relationship with the dependent variable.

For example in our case mother education (Meduc) enters in the regression, while father education (Feduc) does not. Probably the variables are highly correlated(same education of husbands) and just one of the two enter in the regression (the other is masked). But still... Also father education may have an effect on alchol consumption of the son!

Exercise: Optional Challenge

We want to predict the waiting time for a bus in Rome at a certain stop. To do so, we have collected data from Atac, the firm who manages the public traffic service between April and May 2016. We have at disposal 193 variables and the dependent variable is travel time between two points of the city. The only objective is to have an accurate out-of-sample predictive power.

Which model would you use? How would you choose the model?

Exercise: Optional Challenge

Open the text file Atacfinal and set the dependent variable the 160th columns (Hint: use y = data[,160], xm = data[,-160]).

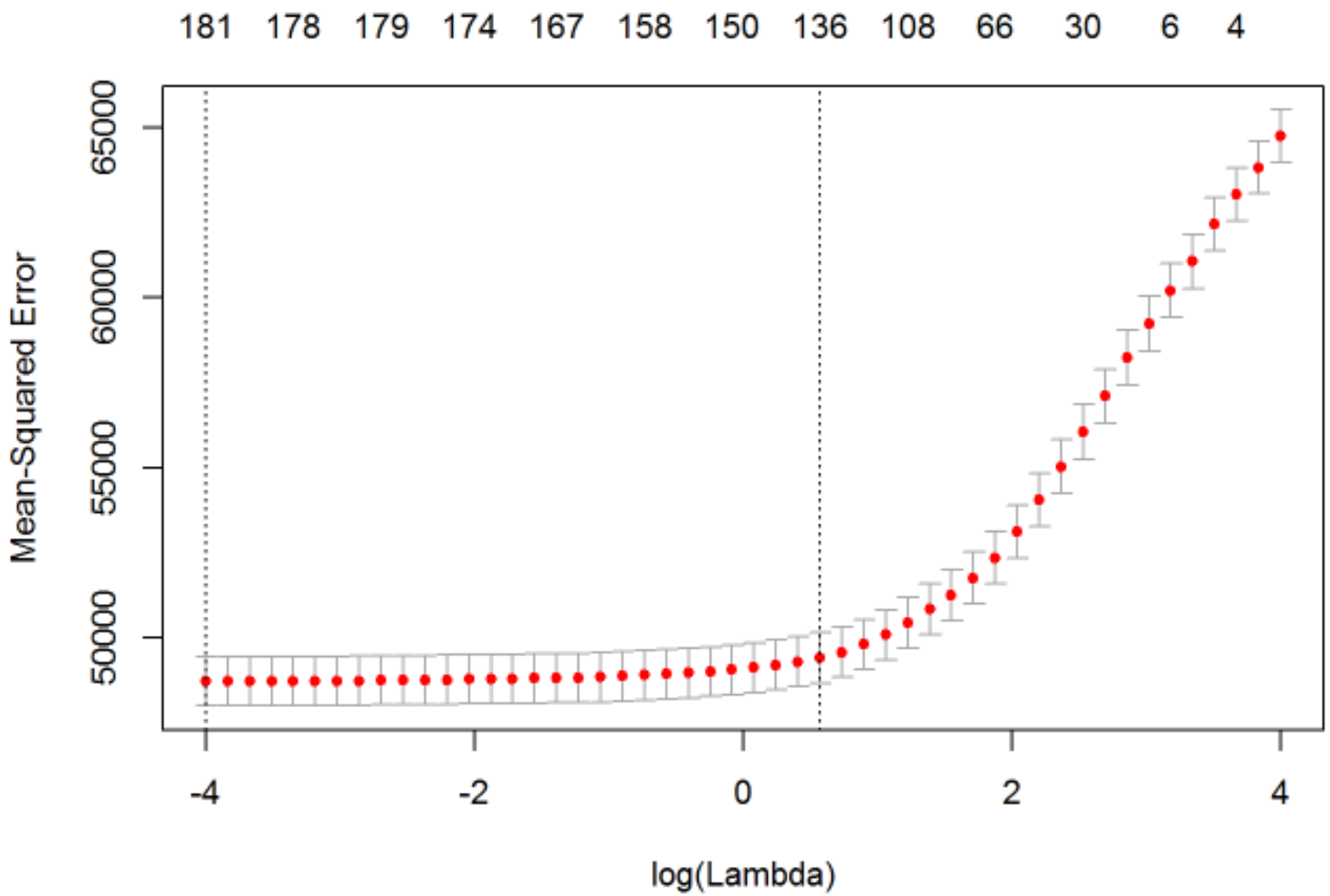
1. Construct a lasso regression and an elastic net regression with alpha = 0.5.
2. Use cross validation to choose the best lambda in each case and compare the values of lambda.
3. Plot the behaviour of the MSPE as a function of lambda for the lasso regression and elastic net regression.
4. Which difference do you notice? Which one do you prefer? How many variables would you use?

Hint: Remember to standardize covariates!

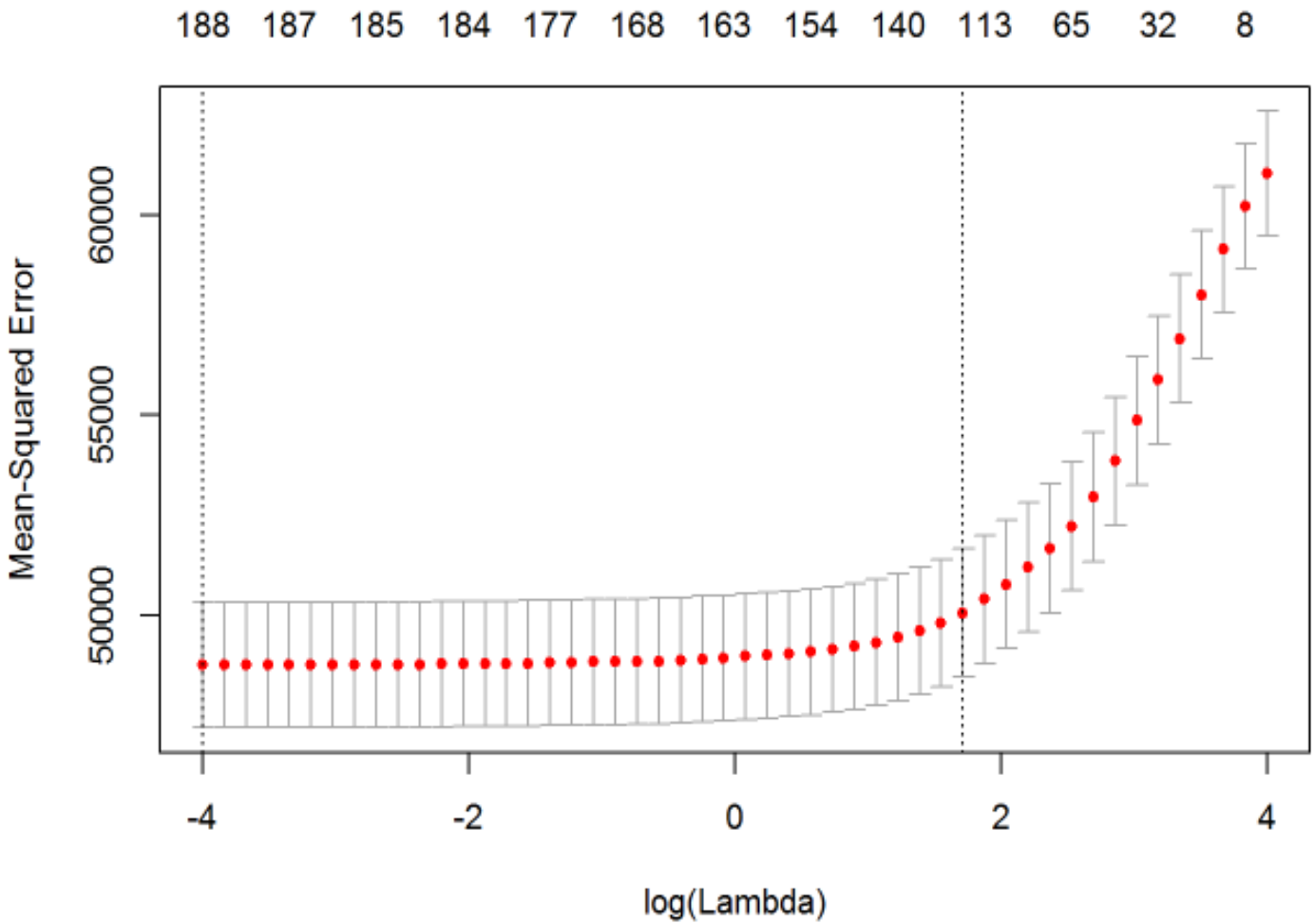
Exercise: Solution optional

```
data <- read.table("./data/Atacfinal.txt")

y <- data[,160]
xm <- data[,-160]

## Standardize covariates


standard_for_dummy <- function (k){ if (length(k[!duplicated(k)])==2)
{ return(1)}
return(sd(k))                                                    }

sd.tr <- apply(xm, 2, standard_for_dummy)

mu_for_dummy <- function (k){ if (length(k[!duplicated(k)])==2)
{ return(0.5)}
mean(k)       }

mu.tr <-apply(xm, 2, mu_for_dummy)

## New covariate matrix


xmn <- scale(xm, center = mu.tr, scale=sd.tr)
```

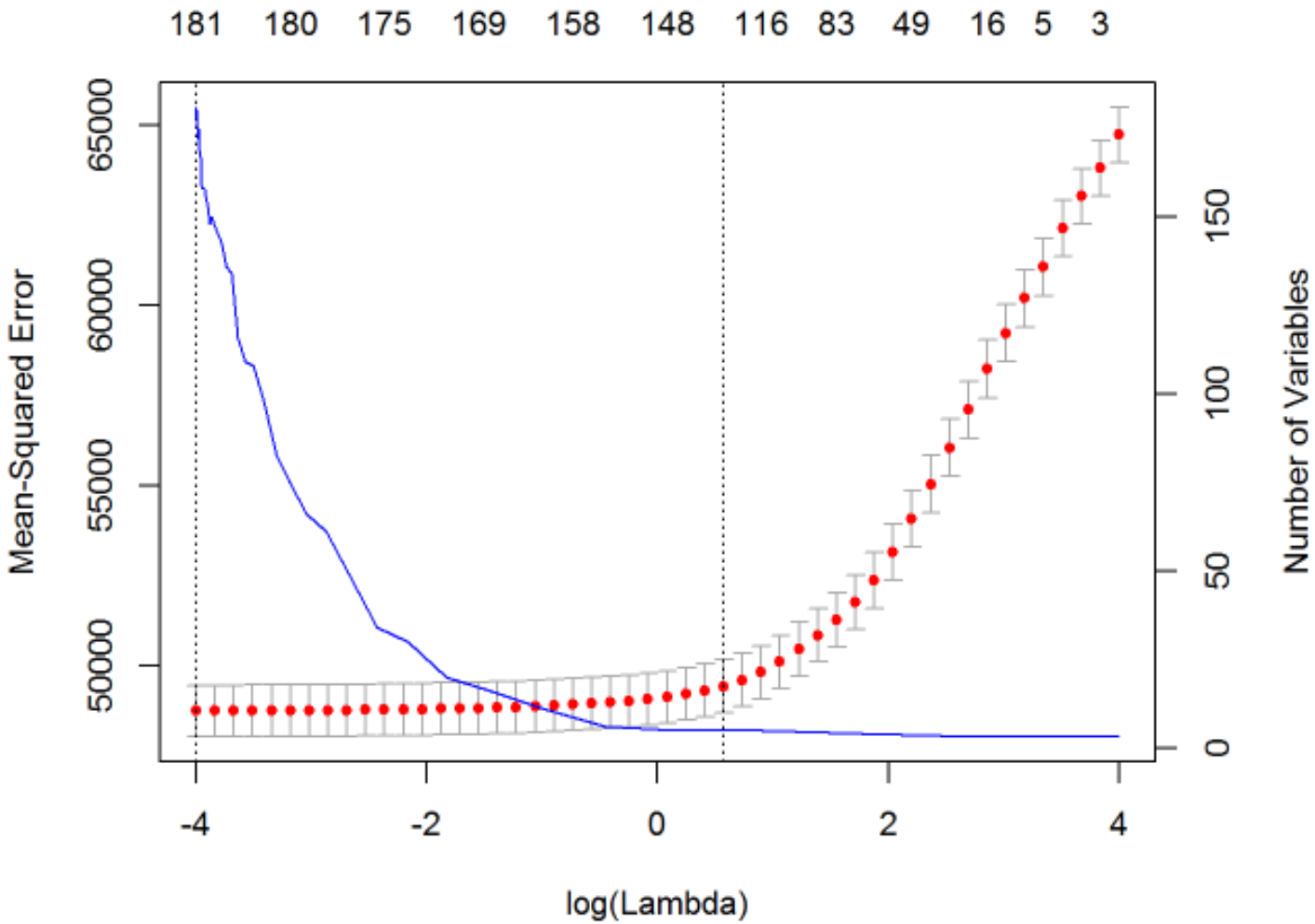
Exercise: Lasso



Exercise: Elastic Net



Exercise: Number of Variables VS Error



Optional Exercise: Solution

The actual error of the algorithm of Atac is 7000. You did better then Atac, congratulations!!