heterogeneity.R

user

2020-03-20

```r
################################################################################
## Course: Machine Learning for Economists and Business Analysts
## Topic: Effect Heterogeneity
################################################################################

rm(list = ls())
set.seed(100239)

#getwd()
#setwd("")

#  Load Packages
library("fBasics")
```

```
## Warning: package 'fBasics' was built under R version 3.6.1

## Loading required package: timeDate

## Loading required package: timeSeries

## Warning: package 'timeSeries' was built under R version 3.6.1
```

```r
library("glmnet")
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-18
```

```r
library("AER")
```

```
## Warning: package 'AER' was built under R version 3.6.2

## Loading required package: car

## Warning: package 'car' was built under R version 3.6.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 3.6.1

##
## Attaching package: 'car'

## The following object is masked from 'package:fBasics':
##
##     densityPlot

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'
```

```
## The following object is masked from 'package:timeSeries':
##
##     time<-

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: survival
```

```r
library("grf")
```

```
## Warning: package 'grf' was built under R version 3.6.1
```

```r
library("hdm")
```

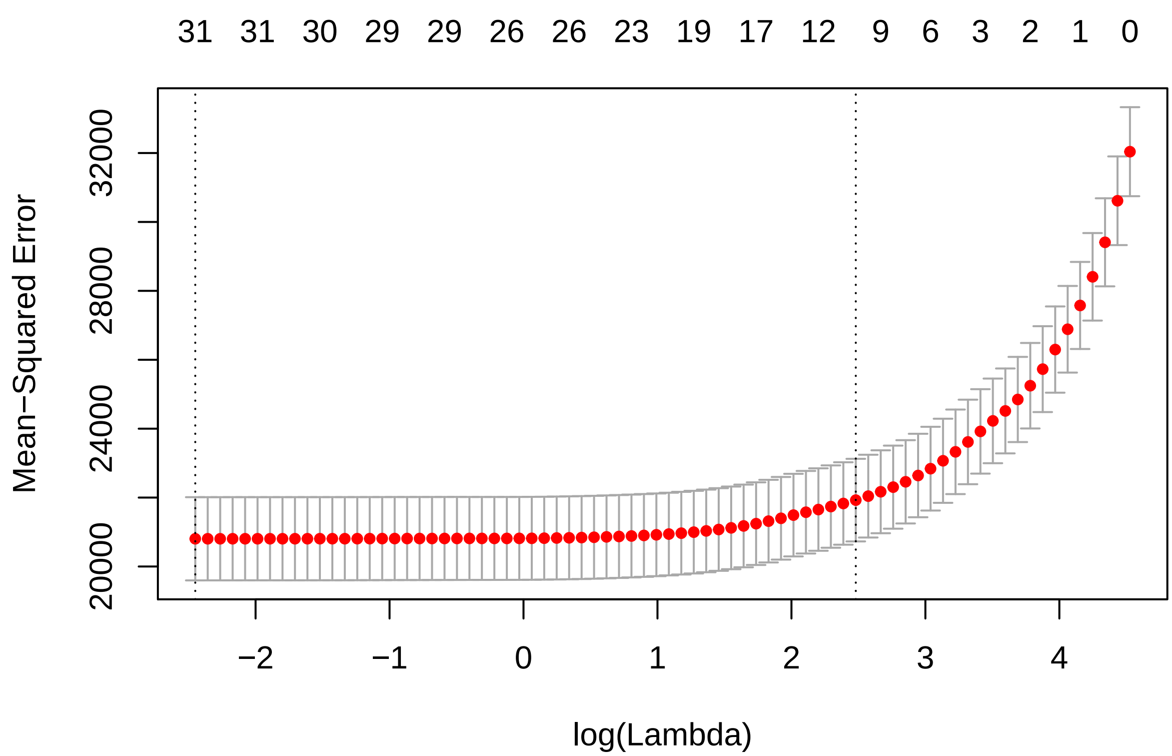
```
## Warning: package 'hdm' was built under R version 3.6.3
```

```r
library("lmtest")
library("sandwich")
library("tidyverse")
```

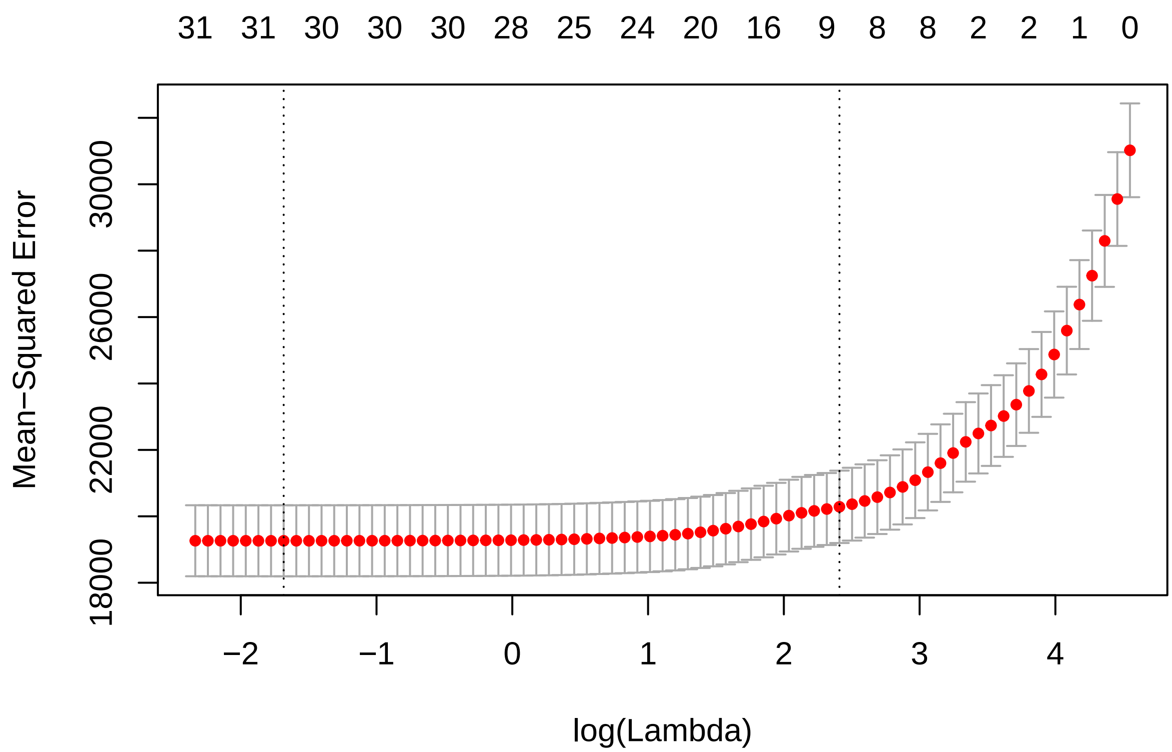
```
## -- Attaching packages ----------------------------------------------------------------- tidyverse 1.

## v ggplot2 3.2.0     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.2
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------- tidyverse_conflict
## x purrr::accumulate() masks foreach::accumulate()
## x tidyr::expand()     masks Matrix::expand()
## x dplyr::filter()     masks timeSeries::filter(), stats::filter()
## x dplyr::lag()        masks timeSeries::lag(), stats::lag()
## x dplyr::recode()     masks car::recode()
## x purrr::some()       masks car::some()
## x purrr::when()       masks foreach::when()
```

```r
# Load data
df <- read.csv("job_corps.csv",header=TRUE, sep=",")

###########################################
### Exercise 3: Double Machine Learning ###
###########################################

######################
## Data Preparation ##
######################
set.seed(123456789)

# Generate variable with the rows in training data
size <- floor(0.5 * nrow(df))
set_A <- sample(seq_len(nrow(df)), size = size)
set_B <- seq_len(nrow(df))[-set_A]

##  Generate Variables
# Outcome
```

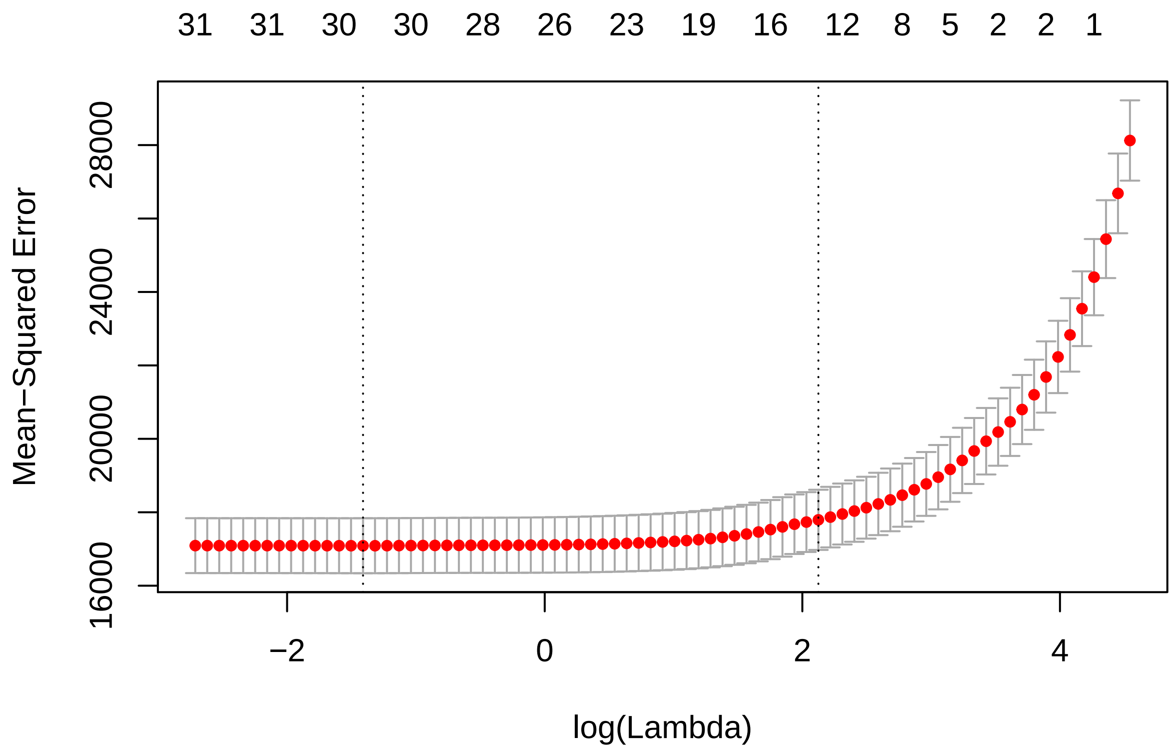
```r
earnings <- as.matrix(df[,1])

# Treatment
treat = 2 #Select treatment 2= offer to participate, 3 = actual participation
treat <- as.matrix(df[,treat])

# Covariates
covariates <- as.matrix(df[,c(4:ncol(df))])

#########################
## Nuisance Parameters ##
#########################

###############################################################################

##  Conditional Potential Earnings under Non-Treatment
p = 1 # 1 for LASSO, 0 for Ridge
set.seed(100237)

## Using Sample A to Predict Sample B
# Potential Earnings under Non-Treatment
lasso_y0_A <- cv.glmnet(covariates[c(set_A,treat==0),], earnings[c(set_A,treat==0)],
                              alpha=p, type.measure = 'mse')
plot(lasso_y0_A)
```



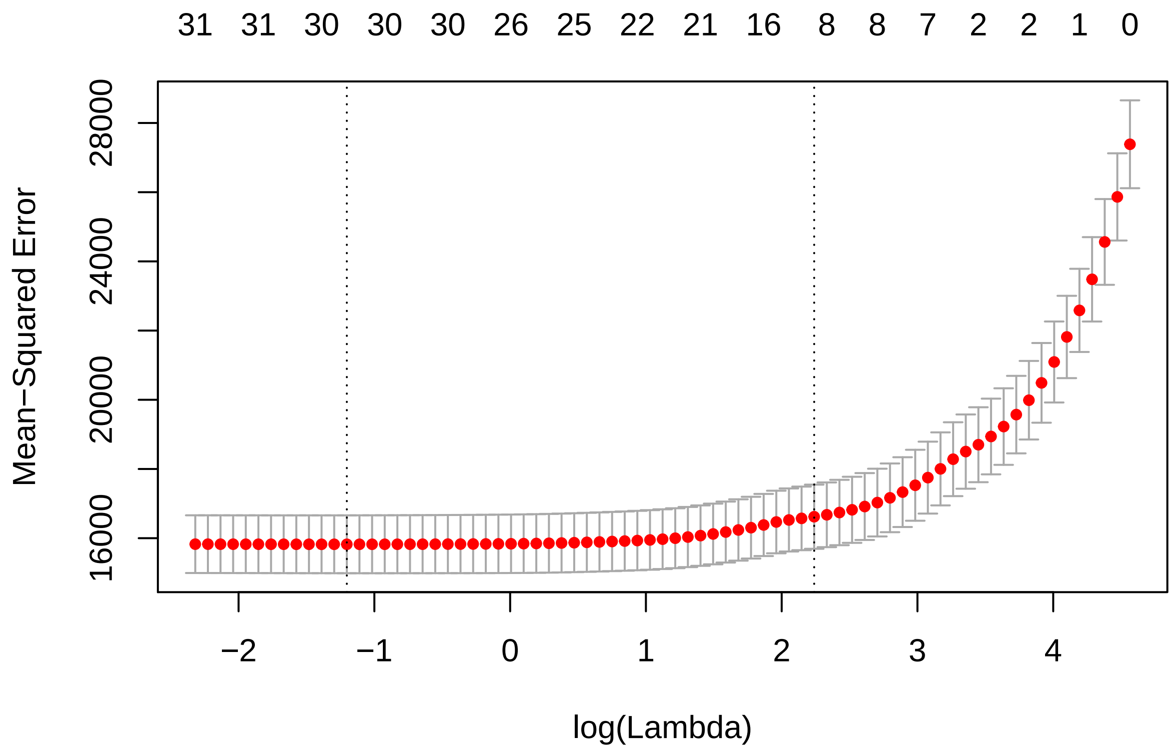
```
fit_y0_A <- glmnet(covariates[c(set_A,treat==0),], earnings[c(set_A,treat==0)]
                        ,lambda = lasso_y0_A$lambda.min)
y0hat_B <- predict(fit_y0_A, covariates)

## Using Sample B to Predict Sample A
# Potential Earnings under Non-Treatment
lasso_y0_B <- cv.glmnet(covariates[c(set_B,treat==0),], earnings[c(set_B,treat==0)],
                              alpha=p, type.measure = 'mse')
plot(lasso_y0_B)
```



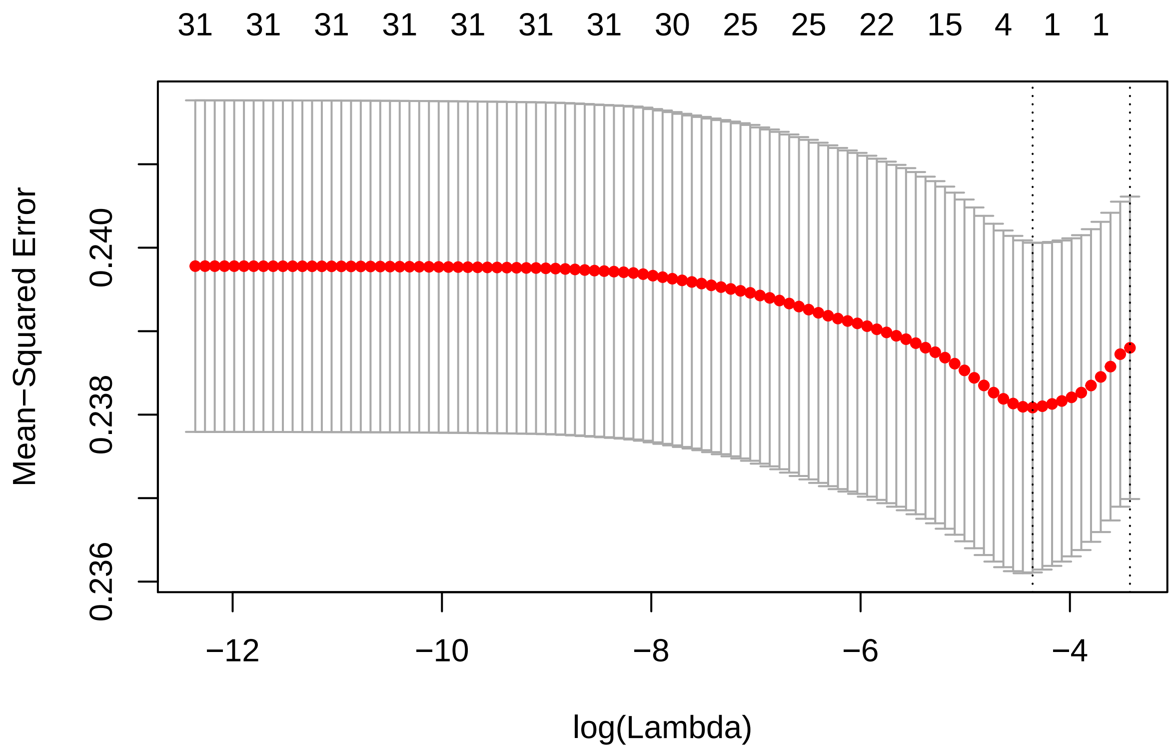

```
fit_y0_B <- glmnet(covariates[c(set_B,treat==0),], earnings[c(set_B,treat==0)]
                        ,lambda = lasso_y0_B$lambda.min)
y0hat_A <- predict(fit_y0_B, covariates)

###############################################################################

##  Conditional Potential Earnings under Treatment
p = 1 # 1 for LASSO, 0 for Ridge
set.seed(100237)

## Using Sample A to Predict Sample B
# Potential Earnings under Treatment
lasso_y1_A <- cv.glmnet(covariates[c(set_A,treat==1),], earnings[c(set_A,treat==1)],
                        alpha=p, type.measure = 'mse')
plot(lasso_y1_A)
```



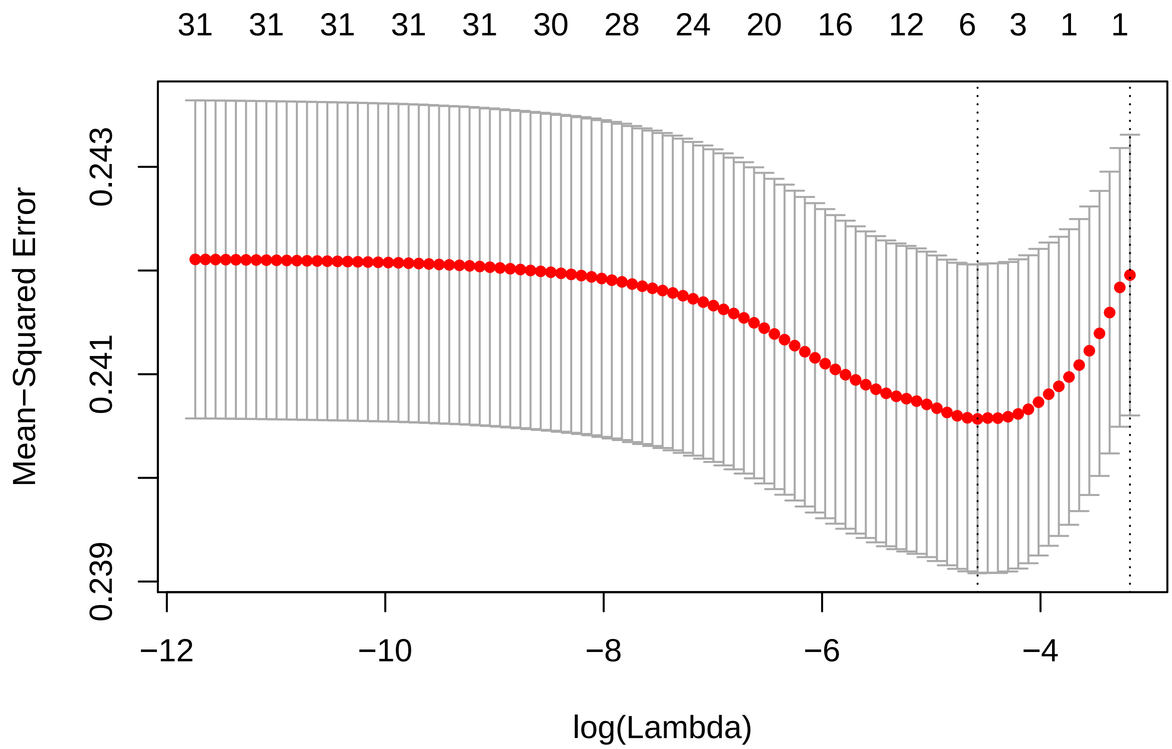

```
fit_y1_A <- glmnet(covariates[c(set_A,treat==1),], earnings[c(set_A,treat==1)]
                   ,lambda = lasso_y1_A$lambda.min)
y1hat_B <- predict(fit_y1_A, covariates)

## Using Sample B to Predict Sample A
# Potential Earnings under Treatment
lasso_y1_B <- cv.glmnet(covariates[c(set_B,treat==1),], earnings[c(set_B,treat==1)],
                        alpha=p, type.measure = 'mse')
plot(lasso_y1_B)
```




```r
fit_y1_B <- glmnet(covariates[c(set_B,treat==1),], earnings[c(set_B,treat==1)]
                   ,lambda = lasso_y1_B$lambda.min)
y1hat_A <- predict(fit_y1_B, covariates, type = 'response')


###############################################################################

##  Propensity Score
p = 1 # 1 for LASSO, 0 for Ridge
set.seed(100236)

# Using Sample A to Predict Sample B
lasso_p_A <- cv.glmnet(covariates[set_A,], treat[set_A], alpha=p, type.measure = 'mse')
plot(lasso_p_A)
```




```r
fit_p_A <- glmnet(covariates[set_A,], treat[set_A],lambda = lasso_p_A$lambda.min)
pscore_B <- predict(fit_p_A, covariates)

# Using Sample B to Predict Sample A
lasso_p_B <- cv.glmnet(covariates[set_B,], treat[set_B,], alpha=p, type.measure = 'mse')
plot(lasso_p_B)
```



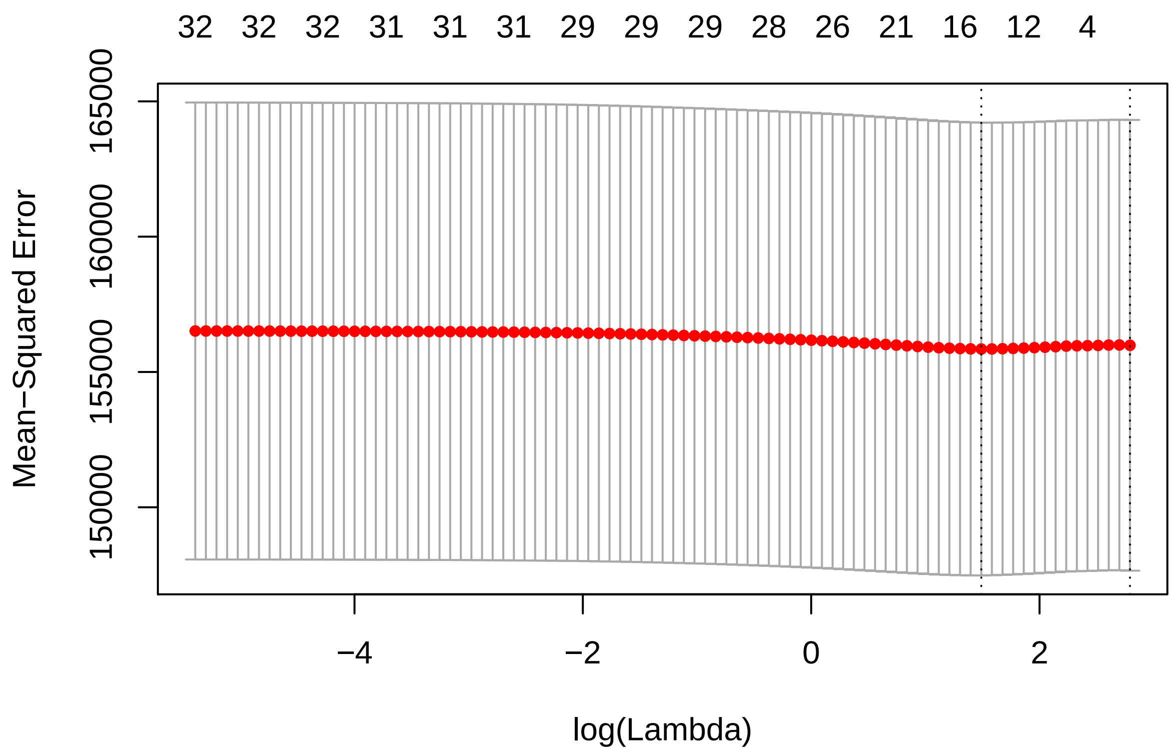

```
fit_p_B <- glmnet(covariates[set_B,], treat[set_B,],lambda = lasso_p_B$lambda.min)
pscore_A <- predict(fit_p_B, covariates)

####################################
## Average Treatment Effect (ATE) ##
####################################

## Efficient Score
# Generate Modified Outcome in each sample
Y_star <- matrix(NA,nrow=nrow(df),ncol=1)
Y_star[set_A] <- invisible(y1hat_A[set_A] -y0hat_A[set_A]
          + treat[set_A]*(earnings[set_A]-y1hat_A[set_A])/pscore_A[set_A]
          - (1-treat[set_A])*(earnings[set_A]-y0hat_A[set_A])/(1-pscore_A[set_A]))
Y_star[set_B] <- invisible(y1hat_B[set_B] -y0hat_B[set_B]
          + treat[set_B]*(earnings[set_B]-y1hat_B[set_B])/pscore_B[set_B]
          - (1-treat[set_B])*(earnings[set_B]-y0hat_B[set_B])/(1-pscore_B[set_B]))

# Average Treatment Effect (ATE)
ATE <- round(mean(Y_star), digits=2)
se_ATE <- round(sd(Y_star)/sqrt(nrow(df)), digits=2)

print(paste0("Average Treatment Effect (ATE): ", ATE))
```

```
## [1] "Average Treatment Effect (ATE): 15.27"
```

```r
print(paste0("Standard Error for ATE: ", se_ATE))
```

```
## [1] "Standard Error for ATE: 3.74"
```

```r
##########################
## Effect Heterogeneity ##
##########################

set.seed(100237)

## Predict Effect Heterogeneity
lasso_A <- cv.glmnet(covariates[set_A,], Y_star[set_A],
                        alpha=p, type.measure = 'mse')
plot(lasso_A)
```

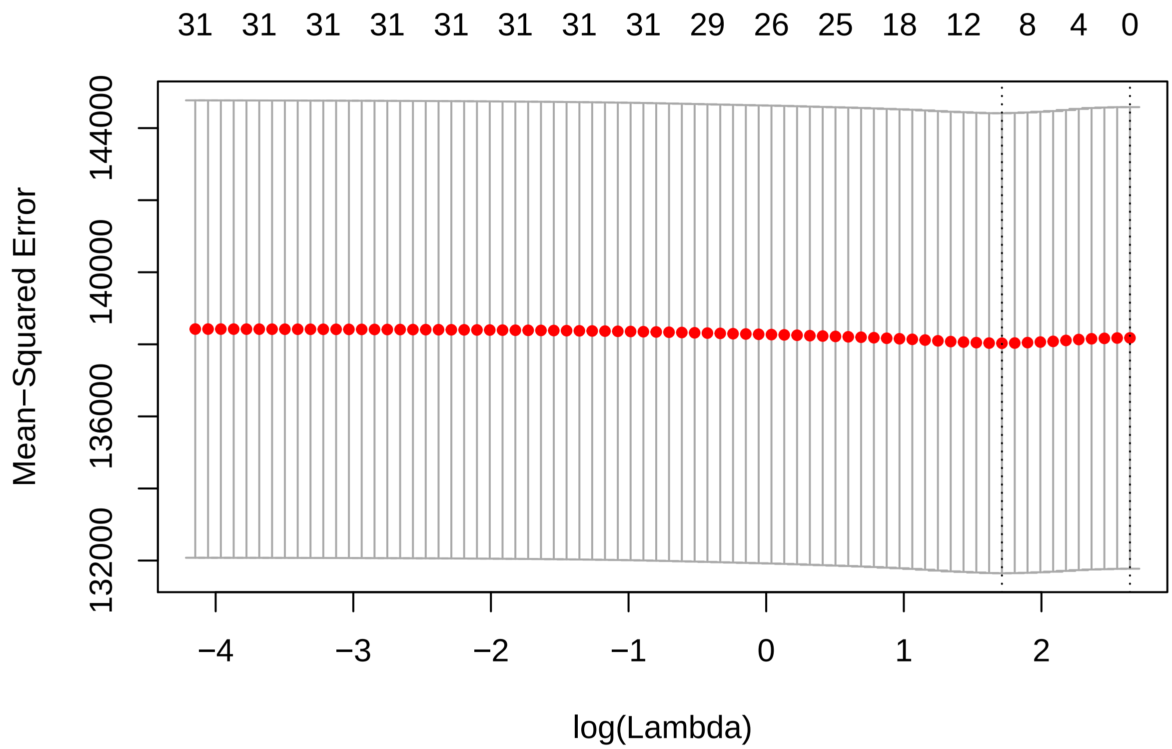



```r
fit_A <- glmnet(covariates[set_A,], Y_star[set_A] ,lambda = lasso_A$lambda.min)
coef(fit_A)
```

```
## 34 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept)  23.52742930
## female         .
## age_1          .
## age_2          .
## age_3        12.52033180
## ed0_6        -5.82078980
## ed6_12         .
```

```
## hs_ged        .
## white         .
## black         .
## hisp        -16.25774632
## oth_eth       .
## haschld       .
## livespou     21.09230073
## everwork      .
## yr_work       .
## currjob       .
## job0_3      -19.42816527
## job3_9        0.00933076
## job9_12      -9.55264268
## earn1         .
## earn2         .
## earn3        30.62522730
## earn4        -8.13622557
## badhlth     -17.28160068
## welf_kid      .
## got_fs        .
## publich       .
## got_afdc      9.80739646
## harduse       2.15854929
## potuse       -6.14731521
## evarrst       .
## pmsa        -14.75654089
## msa           .
```

```r
# Extrapolate to sample B
het_B <- predict(fit_A, covariates)

## Predict Effect Heterogeneity
lasso_B <- cv.glmnet(covariates[set_B,], Y_star[set_B],
                   alpha=p, type.measure = 'mse')
plot(lasso_B)
```

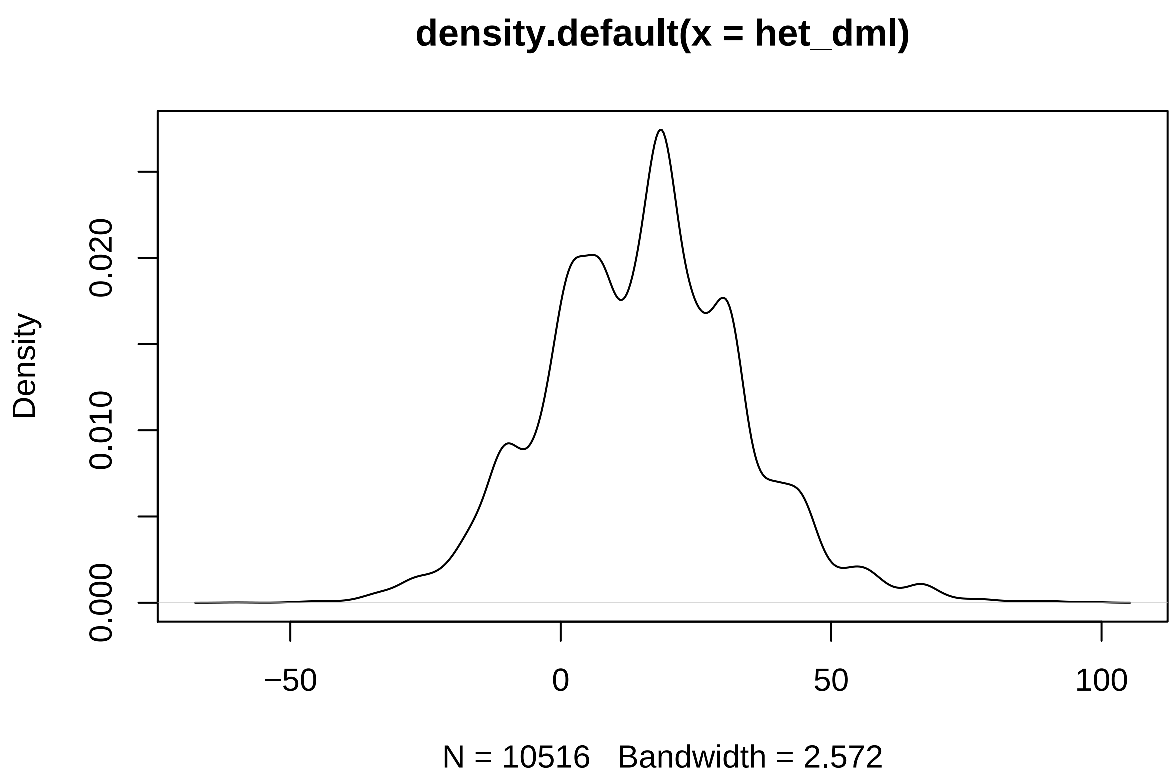



```r
fit_B <- glmnet(covariates[set_B,], Y_star[set_B],lambda = lasso_B$lambda.min)
coef(fit_B)
```

```
## 34 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept)  19.2502851
## female       -1.5098977
## age_1          .
## age_2       -11.2265841
## age_3        13.1617072
## ed0_6          .
## ed6_12         .
## hs_ged         .
## white         0.9417883
## black          .
## hisp        -17.5812696
## oth_eth     -12.5355559
## haschld        .
## livespou       .
## everwork       .
## yr_work       6.8533344
## currjob        .
## job0_3        0.7697507
## job3_9         .
## job9_12      -8.6377619
## earn1          .
```

```
## earn2          4.6619298
## earn3          .
## earn4          .
## badhlth        .
## welf_kid       .
## got_fs         .
## publich        .
## got_afdc       .
## harduse        .
## potuse       -17.4413881
## evarrst        .
## pmsa           .
## msa            .
```

```r
# Extrapolate to sample B
het_A <- predict(fit_B, covariates)

het_dml <- matrix(NA, nrow = nrow(df),ncol =1)
het_dml[set_A] <- het_A[set_A]
het_dml[set_B] <- het_B[set_B]

# Kernel Density Plot
d_dml <- density(het_dml)
plot(d_dml)
```




```r
##################
## Post DML-OLS ##
```

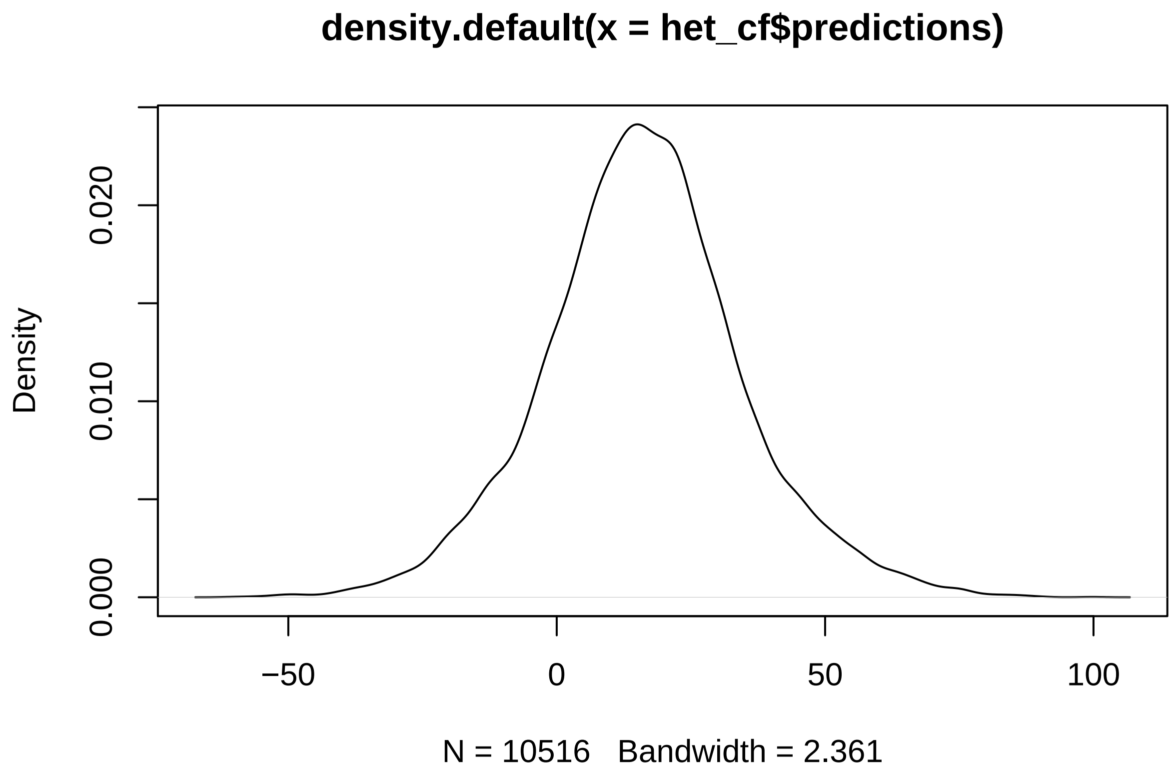
```r
##################

# Multivariate OLS
ols <- lm(Y_star ~ ., data = df[,-c(1,2,3,6,11)])
summary(ols)
```

```
##
## Call:
## lm(formula = Y_star ~ ., data = df[, -c(1, 2, 3, 6, 11)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5938.7  -226.8    -9.4   229.4  2966.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25.4478    16.4056   1.551  0.12089
## female      -10.1023     8.2221  -1.229  0.21922
## age_1         8.3454     9.6815   0.862  0.38871
## age_3        33.6334    10.6629   3.154  0.00161 **
## ed0_6        -8.8541     9.8269  -0.901  0.36761
## ed6_12        6.6268     9.9179   0.668  0.50404
## hs_ged       -4.4720    10.1131  -0.442  0.65836
## black        -6.2643     9.9699  -0.628  0.52981
## hisp        -34.4708    12.1562  -2.836  0.00458 **
## oth_eth     -18.3995    15.9446  -1.154  0.24854
## haschld       1.9428    10.9379   0.178  0.85903
## livespou     23.6093    15.9164   1.483  0.13802
## everwork     -8.8491    12.9009  -0.686  0.49277
## yr_work      28.0670    38.9271   0.721  0.47092
## currjob       0.9235    10.6181   0.087  0.93069
## job0_3       -8.4459    17.7620  -0.476  0.63444
## job3_9       -2.4776    16.6135  -0.149  0.88145
## job9_12     -36.6335    18.0926  -2.025  0.04292 *
## earn1       -19.8919    28.4207  -0.700  0.48400
## earn2        -6.0087    25.6866  -0.234  0.81505
## earn3        18.5540    26.5922   0.698  0.48536
## earn4        -8.6330    29.2573  -0.295  0.76794
## badhlth     -19.7546    11.1167  -1.777  0.07559 .
## welf_kid      2.2484    10.1233   0.222  0.82424
## got_fs       -3.1209     9.4009  -0.332  0.73991
## publich       0.8473     9.5764   0.088  0.92950
## got_afdc     17.6987    10.3180   1.715  0.08632 .
## harduse      13.2195    16.2553   0.813  0.41610
## potuse      -22.4182     9.0346  -2.481  0.01310 *
## evarrst      -6.4151     9.0507  -0.709  0.47847
## pmsa        -13.9848    10.9306  -1.279  0.20078
## msa           1.0269     9.9860   0.103  0.91809
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 382.8 on 10484 degrees of freedom
## Multiple R-squared:  0.005986,   Adjusted R-squared:  0.003046
## F-statistic: 2.036 on 31 and 10484 DF,  p-value: 0.0005847
```

```r
# Robust standard errors
coeftest(ols, vcov = vcovHC(ols, type = "HC1"))
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   25.44779   16.25351  1.5657 0.117454
## female       -10.10230    7.98043 -1.2659 0.205583
## age_1          8.34539    9.83018  0.8490 0.395926
## age_3         33.63337   11.23212  2.9944 0.002756 **
## ed0_6         -8.85410    9.93204 -0.8915 0.372698
## ed6_12         6.62678   10.05294  0.6592 0.509789
## hs_ged        -4.47197   10.53161 -0.4246 0.671120
## black         -6.26432   10.13888 -0.6179 0.536687
## hisp         -34.47076   12.55284 -2.7461 0.006042 **
## oth_eth      -18.39945   17.46574 -1.0535 0.292155
## haschld        1.94279   11.18785  0.1737 0.862143
## livespou      23.60926   16.10231  1.4662 0.142623
## everwork      -8.84908   11.66318 -0.7587 0.448037
## yr_work       28.06696   40.54454  0.6923 0.488796
## currjob        0.92352   11.43150  0.0808 0.935613
## job0_3        -8.44594   19.27477 -0.4382 0.661260
## job3_9        -2.47762   17.46954 -0.1418 0.887221
## job9_12      -36.63353   19.95336 -1.8360 0.066392 .
## earn1        -19.89187   29.41222 -0.6763 0.498857
## earn2         -6.00873   27.13783 -0.2214 0.824774
## earn3         18.55401   28.09955  0.6603 0.509079
## earn4         -8.63303   32.68614 -0.2641 0.791693
## badhlth      -19.75465   11.03919 -1.7895 0.073563 .
## welf_kid       2.24842    9.53789  0.2357 0.813642
## got_fs        -3.12092    9.22048 -0.3385 0.735011
## publich        0.84730    9.02332  0.0939 0.925189
## got_afdc      17.69866    9.66353  1.8315 0.067056 .
## harduse       13.21947   16.52843  0.7998 0.423843
## potuse       -22.41819    8.62404 -2.5995 0.009349 **
## evarrst       -6.41505    9.12383 -0.7031 0.482003
## pmsa         -13.98476   10.95108 -1.2770 0.201623
## msa            1.02693    9.62471  0.1067 0.915031
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
###################
## Causal Forest ##
###################

set.seed(1234567)
cf <- causal_forest(covariates, earnings, treat)
het_cf <- predict(cf,estimate.variance = TRUE)

# Kernel Density Plot
d_cf <- density(het_cf$predictions)
plot(d_cf)
```




```r
cor(het_dml,het_cf$predictions)
```

```
##           [,1]
## [1,] 0.5206994
```

```r
## Inference

# t-Statistics
t_stat <- as.matrix(het_cf$predictions)/ as.matrix(sqrt(het_cf$variance.estimates))

sig_pos <- (t_stat>=1.96)== TRUE
sig_neg <- (t_stat<=-1.96)== TRUE
insig <- (abs(t_stat) <1.96)== TRUE

print(paste0("Share with positive effects: ", round(mean(sig_pos), digits=4)))
```

```
## [1] "Share with positive effects: 0.1503"
```

```r
print(paste0("Share with negative effects: ", round(mean(sig_neg), digits=4)))
```

```
## [1] "Share with negative effects: 0.0035"
```

```r
print(paste0("Share with insignificant effects: ", round(mean(insig), digits=4)))
```

```
## [1] "Share with insignificant effects: 0.8461"
```