140.644 - Statistical Machine Learning

Final Project

Martin Skarzynski

March 18, 2018

Part 1: A prediction model for the age in months at time of examination, RIDAGEEX

```
load("nhanes2003-2004.Rda")

#Explore the dataset
dim(nhanes2003_2004)
```

```
## [1] 10122   813
```

```
#head(nhanes2003_2004)
#names(nhanes2003_2004)
#library(purrr, help)
#map(nhanes2003_2004, class)
library(dplyr, help)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#glimpse(nhanes2003_2004)

#Remove all rows with NAs in the outcome variable column
condition <- !is.na(nhanes2003_2004$RIDAGEEX)
length(condition)
```

```
## [1] 10122
```

```
nhanes2003_2004 <- nhanes2003_2004[condition,]
#Next, remove all variables with NAs
nhanes2003_2004 <- nhanes2003_2004[,colMeans(is.na(nhanes2003_2004))<=0.00]
# nhanes2003_2004 <- nhanes2003_2004[rowMeans(is.na(nhanes2003_2004))<=0.00,]

#Double check that removeing NAs worked
sum(!is.na(nhanes2003_2004))
```

```
## [1] 226320
```

```
sum(is.na(nhanes2003_2004))
```

```
## [1] 0
```

```r
dim(nhanes2003_2004)
```

```
## [1] 9430   24
```

```r
#glimpse(nhanes2003_2004)

# Remove factors with fewer than two levels
#str(nhanes2003_2004)
n_lev <- sapply(nhanes2003_2004, nlevels)
#head(n_lev)
nhanes2003_2004 <- nhanes2003_2004[ , n_lev>=2]

which(names(nhanes2003_2004)=="RIDAGEEX")
```

```
## [1] 9
```

```r
#names(nhanes2003_2004)

#Turn all variables to numeric for now
nhanes2003_2004 <- sapply( nhanes2003_2004, as.numeric )
nhanes2003_2004 <- as.data.frame( nhanes2003_2004 )

"RIDAGEEX" %in% names(nhanes2003_2004)
```

```
## [1] TRUE
```

```r
## nhanes2003_2004$RIDAGEEX <- as.numeric(nhanes2003_2004$RIDAGEEX)

set.seed(20180318)

nhanes <-
  nhanes2003_2004 %>%
  rowwise() %>%
  mutate(splt = sample(
    c("train", "test"),
    1,
    replace = TRUE,
    prob = c(0.75, 0.25) # Set weights for each group here
  ))
#head(nhanes)

train <- nhanes %>%
    filter(splt == "train") %>%
    select(-SEQN, -splt)

test <- nhanes %>%
    filter(splt == "test") %>%
    select(-SEQN, -splt)

index_train <- which(nhanes$splt=="train")
index_test <- which(nhanes$splt=="test")
dim(nhanes[index_train,-c(1,24)])
```

```
## [1] 7056   22
```

```r
all(nhanes[index_train,-c(1,24)]==train)
```

```
## [1] TRUE
```

```r
dim(train)
```

```
## [1] 7056   22
```

```r
#names(nhanes)

lin <- lm(formula = RIDAGEEX ~ .,
          data = train,
          na.action = na.omit) #NAs were already removed
```

When λ is zero, we should obtain the same results with linear, ridge and lasso regression.

```r
library(glmnet, help)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-13
```

```r
#convert response variable to vector
y_train <- train$RIDAGEEX
mod_mat <- model.matrix(object = RIDAGEEX ~ ., data = train)
#mod_mat %>% head

rid_cv <- cv.glmnet(mod_mat, y_train, alpha = 0)
rid_lam <- rid_cv$lambda.min
rid_mod <- glmnet(x = mod_mat,
                  y = y_train,
                  alpha = 0,
                  lambda = rid_lam)

las_cv <- cv.glmnet(mod_mat, y_train, alpha = 1)
las_lam <- las_cv$lambda.min
las_mod <- glmnet(x = mod_mat,
                  y = y_train,
                  alpha = 1,
                  lambda = las_lam)

#make predictions
pred_lm <- predict(object = lin, newdata = test)
```

```
## Warning in predict.lm(object = lin, newdata = test): prediction from a
## rank-deficient fit may be misleading
```

```r
mean((pred_lm - test$RIDAGEEX)^2)
```

```
## [1] 2779.574
```

```r
test_mat <- model.matrix(RIDAGEEX ~ ., data = test)

rid_pred <- predict(rid_mod, s = rid_lam, newx = test_mat)
mean((rid_pred - test$RIDAGEEX)^2)
```

```
## [1] 3967.504
```

```r
las_pred <- predict(las_mod, s = las_lam, newx = test_mat)
mean((las_pred - test$RIDAGEEX)^2)
```

```
## [1] 2787.77
```

```r
coef(lin)
```

```
##   (Intercept)      BMDSTATS      PEASCST1      RIDSTATR      RIDEXMON
##  3.210740e+01 -2.225614e-01  2.969798e+00            NA  1.221872e+00
##      RIAGENDR      RIDAGEYR      RIDAGEMN      RIDRETH1      RIDRETH2
## -1.097913e-01  1.651047e-01  9.629433e-01 -9.632279e-01  1.166802e+00
##       DMDBORN      DMDCITZN      DMDHHSIZ       SIALANG      SIAPROXY
##  2.319925e-01 -8.185409e+00 -7.772925e-01  7.090095e+00  2.096446e+00
##      SIAINTRP      WTINT2YR      WTMEC2YR       SDMVPSU      SDMVSTRA
## -1.072850e+01  7.094412e-04 -4.627857e-04  1.053173e-01 -1.423998e-01
##      DR1DRSTZ      DR2DRSTZ
## -3.693276e+00  1.648239e+00
```

```r
predict(rid_mod, s = rid_lam, exact = T, type = 'coefficients')
```

```
## 23 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept)  80.144337253
## (Intercept)   .
## BMDSTATS      1.186328299
## PEASCST1      3.010562140
## RIDSTATR      .
## RIDEXMON      2.055698820
## RIAGENDR     -0.239904524
## RIDAGEYR      1.359788106
## RIDAGEMN      0.787370334
## RIDRETH1     -1.544006037
## RIDRETH2      0.824958488
## DMDBORN       1.307483464
## DMDCITZN    -13.547197571
## DMDHHSIZ     -4.052524492
## SIALANG      11.954839886
## SIAPROXY      8.938828888
## SIAINTRP    -17.312545231
## WTINT2YR     -0.001575854
## WTMEC2YR     -0.001138606
## SDMVPSU       0.654504185
## SDMVSTRA     -0.192694936
## DR1DRSTZ     -0.800713161
## DR2DRSTZ      0.660650015
```

```r
predict(las_mod, s = las_lam, exact = T, type = 'coefficients')
```

```
## 23 x 1 sparse Matrix of class "dgCMatrix"
##                      1
## (Intercept) 13.5902482
## (Intercept)  .
## BMDSTATS     .
## PEASCST1     .
## RIDSTATR     .
## RIDEXMON     .
```

```
## RIAGENDR     .
## RIDAGEYR     0.1332356
## RIDAGEMN     0.9606375
## RIDRETH1     .
## RIDRETH2     .
## DMDBORN      .
## DMDCITZN     .
## DMDHHSIZ     .
## SIALANG      .
## SIAPROXY     .
## SIAINTRP     .
## WTINT2YR     .
## WTMEC2YR     .
## SDMVPSU      .
## SDMVSTRA     .
## DR1DRSTZ     .
## DR2DRSTZ     .
```

The MSEs I obtained are really low, unfortunately, looking at the coefficients it appears that a few of the variables are driving the models. Upon further inspection, `RIDAGEMN` and `RIDAGEYR` is the age in months and years, respectively. I will rerun the models after removing these variables.

```
train2 <- train %>%
    select(-RIDAGEMN, -RIDAGEYR)

test2 <- test %>%
    select(-RIDAGEMN, -RIDAGEYR)

lin2 <- lm(formula = RIDAGEEX ~ .,
          data = train2,
          na.action = na.omit) #NAs were already removed

#convert response variable to vector
y_train2 <- train2$RIDAGEEX
mod_mat2 <- model.matrix(object = RIDAGEEX ~ ., data = train2)
#mod_mat2 %>% head

rid_cv2 <- cv.glmnet(mod_mat2, y_train2, alpha = 0)
rid_lam2 <- rid_cv2$lambda.min
rid_mod2 <- glmnet(x = mod_mat2,
                  y = y_train2,
                  alpha = 0,
                  lambda = rid_lam2)

las_cv2 <- cv.glmnet(mod_mat2, y_train2, alpha = 1)
las_lam2 <- las_cv2$lambda.min
las_mod2 <- glmnet(x = mod_mat2,
                  y = y_train2,
                  alpha = 1,
                  lambda = las_lam2)

#make predictions
pred_lm2 <- predict(object = lin2, newdata = test2)
```

```
## Warning in predict.lm(object = lin2, newdata = test2): prediction from a
```

```
## rank-deficient fit may be misleading
```

```r
mean((pred_lm2 - test2$RIDAGEEX)^2)
```

```
## [1] 67086.84
```

```r
test_mat2 <- model.matrix(RIDAGEEX ~ ., data = test2)

rid_pred2 <- predict(rid_mod2, s = rid_lam2, newx = test_mat2)
mean((rid_pred2 - test2$RIDAGEEX)^2)
```

```
## [1] 67170.32
```

```r
las_pred2 <- predict(las_mod2, s = las_lam2, newx = test_mat2)
mean((las_pred2 - test2$RIDAGEEX)^2)
```

```
## [1] 67092.88
```

```r
coef(lin2)
```

```
##   (Intercept)      BMDSTATS      PEASCST1      RIDSTATR      RIDEXMON
##  644.59400668   14.70610782   16.73483133            NA   13.06993171
##      RIAGENDR      RIDRETH1      RIDRETH2       DMDBORN      DMDCITZN
##    6.56668458   -7.92154819  -10.88664541   43.10342921 -120.17079744
##      DMDHHSIZ       SIALANG      SIAPROXY      SIAINTRP      WTINT2YR
##  -47.43690898   92.46659665  106.06749533  -96.62026334   -0.01568087
##      WTMEC2YR       SDMVPSU      SDMVSTRA      DR1DRSTZ      DR2DRSTZ
##   -0.02159239    0.16133156   -0.83021491   12.12515026   -9.09910450
```

```r
predict(rid_mod2, s = rid_lam2, exact = T, type = 'coefficients')
```

```
## 21 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  634.55798922
## (Intercept)     .
## BMDSTATS      14.76046516
## PEASCST1      16.31810247
## RIDSTATR        .
## RIDEXMON      12.06962470
## RIAGENDR       6.37345939
## RIDRETH1      -7.73782378
## RIDRETH2     -10.43609440
## DMDBORN       38.12304130
## DMDCITZN    -106.13660962
## DMDHHSIZ     -45.76005969
## SIALANG       83.21932075
## SIAPROXY     103.97270981
## SIAINTRP     -92.53809987
## WTINT2YR      -0.01770495
## WTMEC2YR      -0.01981538
## SDMVPSU        0.34838774
## SDMVSTRA      -0.88199384
## DR1DRSTZ      10.87664069
## DR2DRSTZ      -8.27157724
```

```r
predict(las_mod2, s = las_lam2, exact = T, type = 'coefficients')
```

```
## 21 x 1 sparse Matrix of class "dgCMatrix"
```

```
##                           1
## (Intercept)  637.63512973
## (Intercept)    .
## BMDSTATS      14.43356411
## PEASCST1      15.97926172
## RIDSTATR       .
## RIDEXMON      12.29523129
## RIAGENDR       5.91603651
## RIDRETH1      -7.40621496
## RIDRETH2     -10.24691339
## DMDBORN       40.73254601
## DMDCITZN    -114.86491424
## DMDHHSIZ     -47.34453189
## SIALANG       90.00730520
## SIAPROXY     105.84308716
## SIAINTRP     -93.24053506
## WTINT2YR      -0.01565267
## WTMEC2YR      -0.02138917
## SDMVPSU        .
## SDMVSTRA      -0.76740007
## DR1DRSTZ      10.77998070
## DR2DRSTZ      -8.08346165
```

```
library(randomForest)
```

```
## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
rf_mod <- randomForest(RIDAGEEX ~ ., data = train, mtry = ncol(train)-2, ntree = 500)
dim(train)
```

```
## [1] 7056   22
```

```
rf_pred <- predict(rf_mod, newdata = test)
length(rf_pred)
```

```
## [1] 2374
```

```
length(test$RIDAGEEX)
```

```
## [1] 2374
```

```
mean((rf_pred - test$RIDAGEEX)^2)
```

```
## [1] 924.8965
```

```
rf_mod2 <- randomForest(RIDAGEEX ~ ., data = train2, mtry = ncol(train2)-2, ntree = 500)
dim(train)
```

```
## [1] 7056   22
```

```
rf_pred2 <- predict(rf_mod2, newdata = test2)
length(rf_pred2)
```

```
## [1] 2374
```

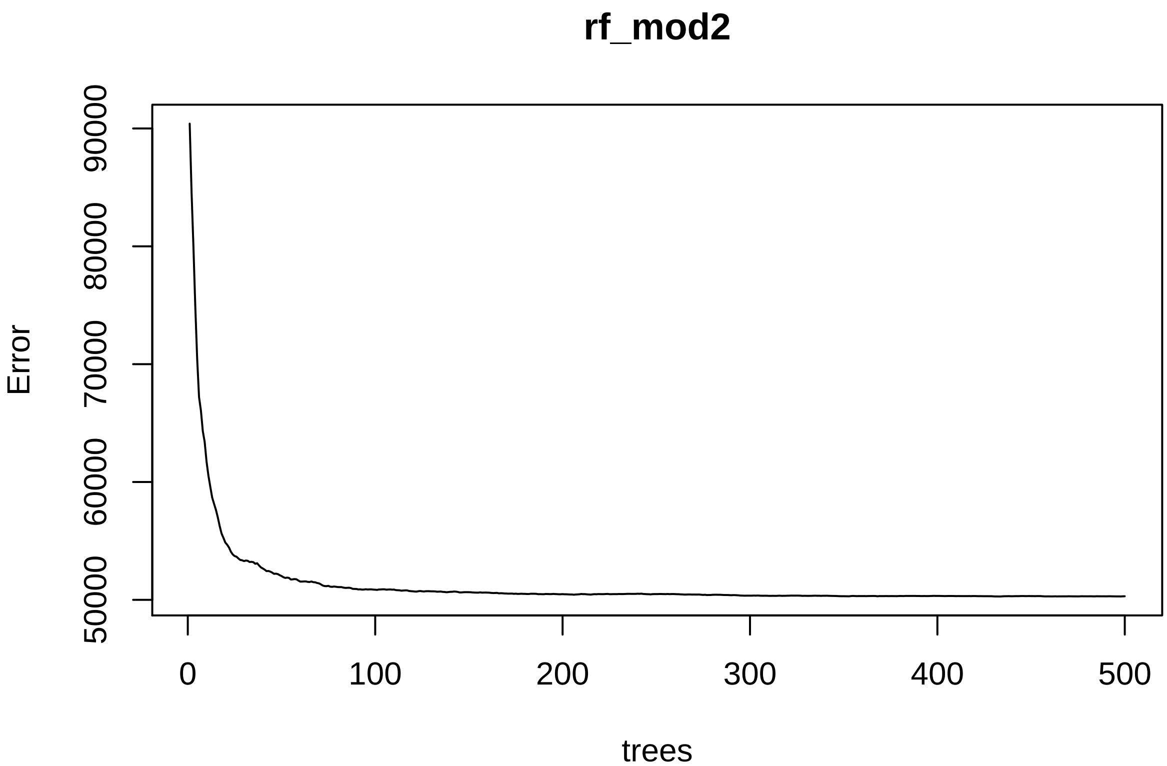
```
length(test2$RIDAGEEX)
```

```
## [1] 2374
```

```
mean((rf_pred2 - test2$RIDAGEEX)^2)
```

```
## [1] 50638.12
```

```
plot(rf_mod2)
```




```
#boosting
library(gbm)
```

```
## Loading required package: survival
```

```
## Loading required package: lattice
```

```
## Loading required package: splines
```

```
## Loading required package: parallel
```

```
## Loaded gbm 2.1.3
```

```
lambdas <- 10^seq(-10, -0.2, by = 0.1)
train_err <- rep(NA, length(lambdas))
for (i in 1:length(lambdas)) {
    boost <- gbm(RIDAGEEX ~ .,
                         data = train2,
                         distribution = "gaussian",
```

```r
                          n.trees = 500,
                          shrinkage = lambdas[i])
    pred_train <- predict(boost, train2, n.trees = 500)
    train_err[i] <- mean((pred_train - train$RIDAGEEX)^2)
}
```

```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.
```

```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.
```

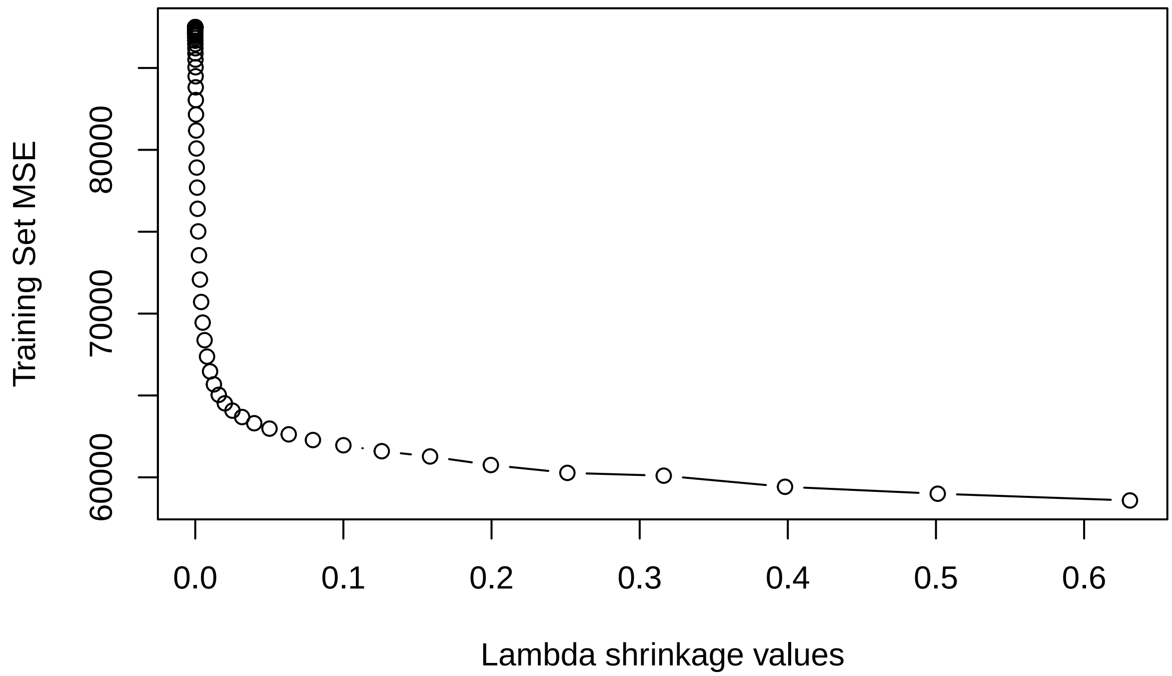
```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.
```

```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.
```

```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.
```

```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.
```

```r
plot(lambdas,
     train_err,
     type = "b",
     xlab = "Lambda shrinkage values",
     ylab = "Training Set MSE")
```



Now I will produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.

```
test_err <- rep(NA, length(lambdas))
for (i in 1:length(lambdas)) {
    boost <- gbm(RIDAGEEX ~ .,
                         data = train2,
                         distribution = "gaussian",
                         n.trees = 500,
                         shrinkage = lambdas[i])
    pred_test <- predict(boost, test2, n.trees = 500)
    test_err[i] <- mean((pred_test - test2$RIDAGEEX)^2)
}
```

```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
```

```
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
```

```
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
```

```
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
```

```
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
```

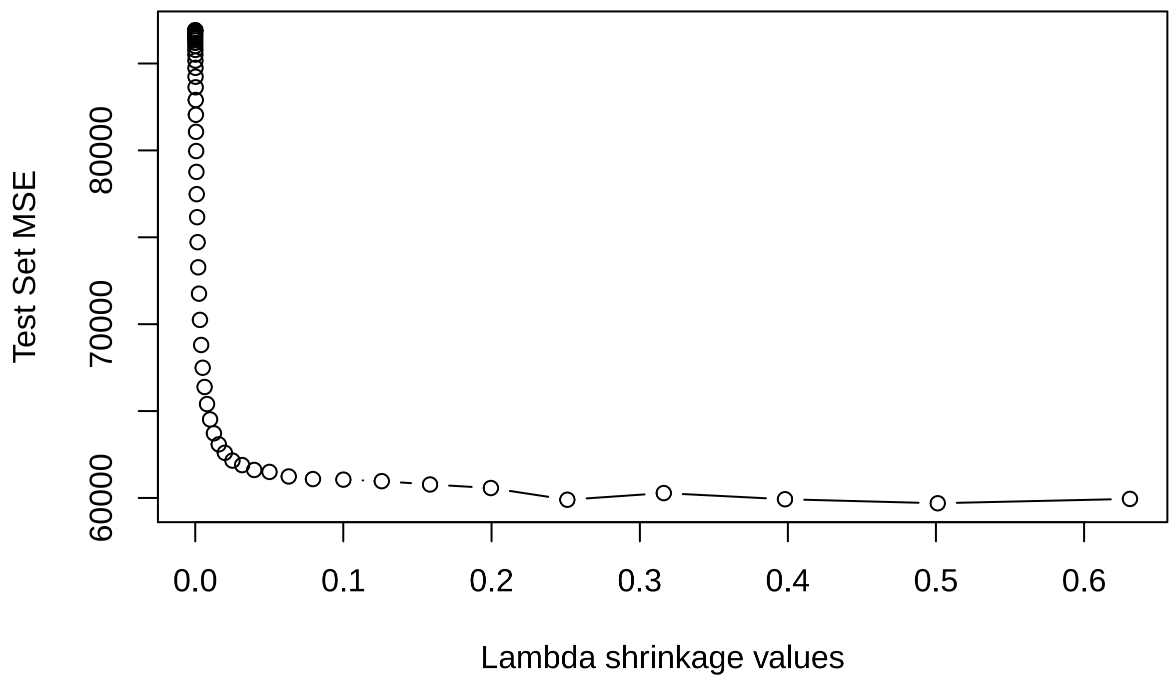
```
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
```

```
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.
```

```
plot(lambdas,
     test_err,
     type = "b",
     xlab = "Lambda shrinkage values",
     ylab = "Test Set MSE")
which.min(test_err)
```

```
## [1] 98
```

```
abline(h = which.min(test_err), lty = 2, col = "red")
```




```
lambdas[which.min(test_err)]
```

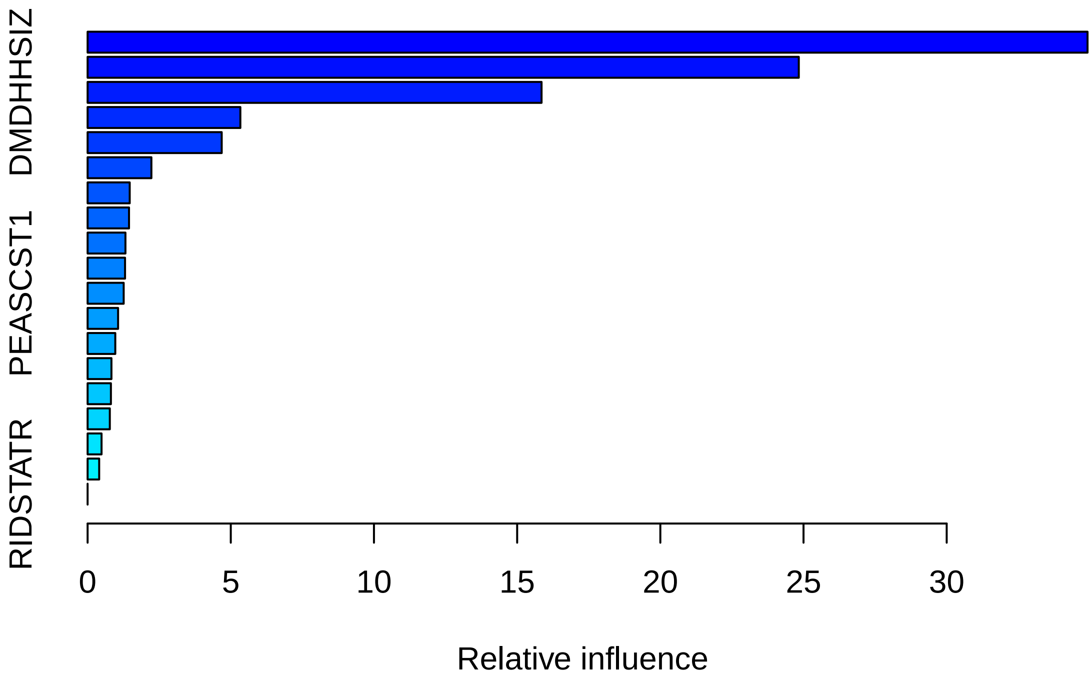
```
## [1] 0.5011872
```

The minimum test MSE is 98, which was obtained with λ of 0.501. The test MSE for boosting is (98).

```
library(gbm)
boost <- gbm(RIDAGEEX ~ .,
                     data = train2,
                     distribution = "gaussian",
                     n.trees = 1000,
                     shrinkage = lambdas[which.min(test_err)])
```

```
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 3: RIDSTATR has no variation.
```

```r
summary(boost)
```




```
##                 var    rel.inf
## WTINT2YR WTINT2YR 34.9185137
## WTMEC2YR WTMEC2YR 24.8350773
## DMDHHSIZ DMDHHSIZ 15.8527183
## SDMVSTRA SDMVSTRA  5.3338751
## SIAPROXY SIAPROXY  4.6830020
## RIDRETH1 RIDRETH1  2.2273256
## SIAINTRP SIAINTRP  1.4706002
## RIDRETH2 RIDRETH2  1.4463935
## DR1DRSTZ DR1DRSTZ  1.3213961
## SDMVPSU   SDMVPSU  1.3074629
## PEASCST1 PEASCST1  1.2595944
## DMDBORN   DMDBORN  1.0646520
## BMDSTATS BMDSTATS  0.9665995
## SIALANG   SIALANG  0.8323435
## DR2DRSTZ DR2DRSTZ  0.8138287
## DMDCITZN DMDCITZN  0.7756094
## RIAGENDR RIAGENDR  0.4869990
## RIDEXMON RIDEXMON  0.4040086
## RIDSTATR RIDSTATR  0.0000000
```

From the results above, it appears that WTINT2YR and WTMEC2YR are the most important and second most important variables, respectively. # Part 2: For participants 50 years and older, build a prediction model for the final mortality status, mortstat

```r
load("nhanes2003-2004.Rda")
#head(nhanes2003_2004)
#Turn all variables to numeric
nhanes2003_2004 <- sapply( nhanes2003_2004, as.numeric )
nhanes2003_2004 <- as.data.frame( nhanes2003_2004 )
#Remove all rows with NAs in the outcome variable column
```

```r
nhanes <- nhanes2003_2004 %>%
    filter(RIDAGEYR > 50) %>%
    filter(!is.na(mortstat))

#Next, remove all variables with NAs
nhanes <- nhanes[,colMeans(is.na(nhanes))<=0.00]
# nhanes2003_2004 <- nhanes2003_2004[rowMeans(is.na(nhanes2003_2004))<=0.00,]

#Double check that removeing NAs worked
sum(!is.na(nhanes))
```

```
## [1] 137268
```

```r
sum(is.na(nhanes))
```

```
## [1] 0
```

```r
dim(nhanes)
```

```
## [1] 2214   62
```

```r
glimpse(nhanes)
```

```
## Observations: 2,214
## Variables: 62
## $ SEQN     <dbl> 5, 8, 11, 16, 17, 29, 32, 33, 35, 46, 47, 50, 51, 54,...
## $ BPQ010   <dbl> 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ BPQ060   <dbl> 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1,...
## $ SDDSRVYR <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ RIDSTATR <dbl> 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ RIAGENDR <dbl> 1, 1, 1, 2, 2, 2, 2, 1, 2, 1, 1, 2, 2, 1, 1, 1, 2, 2,...
## $ RIDAGEYR <dbl> 52, 61, 83, 52, 71, 54, 85, 85, 84, 67, 52, 56, 72, 7...
## $ RIDRETH1 <dbl> 3, 4, 3, 4, 3, 4, 3, 3, 3, 3, 3, 3, 1, 4, 3, 1, 2, 3,...
## $ RIDRETH2 <dbl> 1, 2, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 3, 2, 1, 3, 5, 1,...
## $ DMQMILIT <dbl> 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 1,...
## $ DMDBORN  <dbl> 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 2, 1, 1,...
## $ DMDCITZN <dbl> 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1,...
## $ DMDEDUC2 <dbl> 3, 3, 4, 2, 3, 3, 2, 1, 3, 3, 3, 4, 1, 1, 2, 1, 3, 1,...
## $ DMDEDUC  <dbl> 2, 2, 3, 1, 2, 2, 1, 1, 2, 2, 2, 3, 1, 1, 1, 1, 2, 1,...
## $ DMDHHSIZ <dbl> 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 2, 2, 4, 2, 1,...
## $ DMDHRGND <dbl> 1, 1, 2, 1, 2, 2, 2, 1, 2, 1, 1, 2, 2, 1, 1, 2, 2, 2,...
## $ DMDHRAGE <dbl> 40, 48, 67, 45, 57, 42, 70, 70, 69, 54, 40, 44, 58, 3...
## $ SIALANG  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1,...
## $ SIAPROXY <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ SIAINTRP <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ WTINT2YR <dbl> 3505, 243, 550, 1049, 1976, 881, 626, 582, 455, 2529,...
## $ WTMEC2YR <dbl> 5064, 405, 953, 1622, 1, 1332, 1116, 986, 846, 4009, ...
## $ SDMVPSU  <dbl> 2, 2, 2, 1, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2,...
## $ SDMVSTRA <dbl> 3, 5, 5, 10, 11, 13, 10, 11, 2, 4, 2, 11, 8, 13, 5, 1...
## $ DIQ010   <dbl> 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ DIQ050   <dbl> 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ DIQ090   <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ DIQ100   <dbl> 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2,...
## $ DIQ120   <dbl> 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 2, 1,...
## $ DIQ140   <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 1, 1,...
## $ HSAQUEX  <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
```

```
## $ MCQ010   <dbl> 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1,...
## $ MCQ053   <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ MCQ092   <dbl> 2, 2, 1, 2, 1, 2, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2,...
## $ MCQ140   <dbl> 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2,...
## $ MCQ160A  <dbl> 2, 1, 1, 2, 1, 2, 2, 2, 1, 1, 1, 2, 1, 1, 2, 2, 1, 1,...
## $ MCQ160B  <dbl> 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ MCQ160C  <dbl> 2, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ MCQ160D  <dbl> 2, 2, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ MCQ160E  <dbl> 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ MCQ160F  <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ MCQ160G  <dbl> 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ MCQ160J  <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 1, 2, 1,...
## $ MCQ160K  <dbl> 2, 2, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1,...
## $ MCQ160L  <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2,...
## $ MCQ160M  <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ MCQ220   <dbl> 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2,...
## $ MCQ245A  <dbl> 1, 1, 2, 1, 2, 1, 2, 2, 2, 2, 1, 1, 1, 2, 2, 2, 1, 2,...
## $ MCQ250A  <dbl> 2, 2, 2, 2, 1, 1, 2, 2, 2, 3, 1, 2, 2, 2, 1, 2, 3, 1,...
## $ MCQ250B  <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 1, 2, 2, 2, 2, 3, 2,...
## $ MCQ250C  <dbl> 2, 2, 1, 1, 2, 2, 2, 2, 2, 3, 1, 2, 2, 2, 2, 2, 3, 2,...
## $ MCQ250E  <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 3, 1, 2, 2, 1, 2, 2, 2, 3, 2,...
## $ MCQ250F  <dbl> 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 1,...
## $ MCQ250G  <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2,...
## $ MCQ265   <dbl> 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 2,...
## $ SSQ011   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1,...
## $ SSQ051   <dbl> 1, 1, 1, 2, 2, 1, 1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 1, 2,...
## $ SSQ061   <dbl> 15, 2, 3, 20, 20, 24, 3, 11, 24, 5, 12, 2, 1, 22, 20,...
## $ WHQ030   <dbl> 1, 2, 3, 3, 1, 1, 3, 3, 1, 1, 1, 1, 3, 1, 3, 1, 1, 1,...
## $ WHQ040   <dbl> 2, 1, 3, 3, 2, 2, 3, 3, 2, 2, 2, 2, 3, 2, 3, 2, 2, 2,...
## $ WHQ090   <dbl> 2, 2, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
## $ mortstat <dbl> 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,...
```

```r
# Remove factors with fewer than two levels
str(nhanes)
```

```
## 'data.frame':    2214 obs. of  62 variables:
##  $ SEQN    : num  5 8 11 16 17 29 32 33 35 46 ...
##  $ BPQ010  : num  2 1 1 1 1 1 1 1 1 1 ...
##  $ BPQ060  : num  1 1 1 1 1 1 2 1 1 1 ...
##  $ SDDSRVYR: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ RIDSTATR: num  2 2 2 2 1 2 2 2 2 2 ...
##  $ RIAGENDR: num  1 1 1 2 2 2 2 1 2 1 ...
##  $ RIDAGEYR: num  52 61 83 52 71 54 85 85 84 67 ...
##  $ RIDRETH1: num  3 4 3 4 3 4 3 3 3 3 ...
##  $ RIDRETH2: num  1 2 1 2 1 2 1 1 1 1 ...
##  $ DMQMILIT: num  2 2 1 2 2 2 2 1 2 2 ...
##  $ DMDBORN : num  1 1 1 1 3 1 1 1 1 1 ...
##  $ DMDCITZN: num  1 1 1 1 2 1 1 1 1 1 ...
##  $ DMDEDUC2: num  3 3 4 2 3 3 2 1 3 3 ...
##  $ DMDEDUC : num  2 2 3 1 2 2 1 1 2 2 ...
##  $ DMDHHSIZ: num  2 2 2 2 1 1 1 1 1 2 ...
##  $ DMDHRGND: num  1 1 2 1 2 2 2 1 2 1 ...
##  $ DMDHRAGE: num  40 48 67 45 57 42 70 70 69 54 ...
##  $ SIALANG : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ SIAPROXY: num  2 2 2 2 2 2 2 2 2 2 ...
```

```
##  $ SIAINTRP: num  2 2 2 2 2 2 2 2 2 2 ...
##  $ WTINT2YR: num  3505 243 550 1049 1976 ...
##  $ WTMEC2YR: num  5064 405 953 1622 1 ...
##  $ SDMVPSU : num  2 2 2 1 2 1 1 2 2 2 ...
##  $ SDMVSTRA: num  3 5 5 10 11 13 10 11 2 4 ...
##  $ DIQ010  : num  2 2 2 2 1 2 2 2 2 2 ...
##  $ DIQ050  : num  2 2 2 2 1 2 2 2 2 2 ...
##  $ DIQ090  : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ DIQ100  : num  2 2 2 2 1 2 2 2 2 2 ...
##  $ DIQ120  : num  2 2 2 2 1 2 2 2 2 2 ...
##  $ DIQ140  : num  2 2 2 2 2 2 2 2 2 1 ...
##  $ HSAQUEX : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ MCQ010  : num  2 2 1 1 1 2 2 2 2 2 ...
##  $ MCQ053  : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ MCQ092  : num  2 2 1 2 1 2 2 1 2 1 ...
##  $ MCQ140  : num  2 2 1 2 2 2 2 1 2 2 ...
##  $ MCQ160A : num  2 1 1 2 1 2 2 2 1 1 ...
##  $ MCQ160B : num  2 2 2 2 1 2 2 2 2 1 ...
##  $ MCQ160C : num  2 2 1 2 1 2 2 2 2 1 ...
##  $ MCQ160D : num  2 2 2 2 1 2 2 2 2 1 ...
##  $ MCQ160E : num  2 2 1 2 2 2 2 2 2 1 ...
##  $ MCQ160F : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ MCQ160G : num  2 2 2 2 1 2 2 2 2 2 ...
##  $ MCQ160J : num  2 2 2 2 2 2 2 2 1 1 ...
##  $ MCQ160K : num  2 2 2 1 1 1 2 2 1 2 ...
##  $ MCQ160L : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ MCQ160M : num  2 2 2 2 2 2 2 2 1 1 ...
##  $ MCQ220  : num  2 2 1 2 2 1 2 2 2 2 ...
##  $ MCQ245A : num  1 1 2 1 2 1 2 2 2 2 ...
##  $ MCQ250A : num  2 2 2 2 1 1 2 2 2 3 ...
##  $ MCQ250B : num  2 2 2 2 2 2 2 2 2 3 ...
##  $ MCQ250C : num  2 2 1 1 2 2 2 2 2 3 ...
##  $ MCQ250E : num  2 2 2 2 2 2 2 2 3 1 ...
##  $ MCQ250F : num  2 1 2 1 2 2 2 2 2 2 ...
##  $ MCQ250G : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ MCQ265  : num  4 2 2 2 2 2 2 2 2 2 ...
##  $ SSQ011  : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ SSQ051  : num  1 1 1 2 2 1 1 1 2 2 ...
##  $ SSQ061  : num  15 2 3 20 20 24 3 11 24 5 ...
##  $ WHQ030  : num  1 2 3 3 1 1 3 3 1 1 ...
##  $ WHQ040  : num  2 1 3 3 2 2 3 3 2 2 ...
##  $ WHQ090  : num  2 2 2 2 1 1 2 2 1 2 ...
##  $ mortstat: num  0 0 1 0 1 0 1 1 0 0 ...
```

```r
unique(nhanes$SDDSRVYR)
```

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

```r
unique(nhanes$mortstat)
```

```
## [1] 0 1
```

```r
"mortstat" %in% names(nhanes)
```

```
## [1] TRUE
```

```r
nhanes2003_2004$RIDAGEEX <- as.numeric(nhanes2003_2004$RIDAGEEX)

set.seed(20180318)

nhanes <-
  nhanes %>%
  rowwise() %>%
  mutate(splt = sample(
    c("train", "test"),
    1,
    replace = TRUE,
    prob = c(0.75, 0.25) # Set weights for each group here
  ))
#head(nhanes)

train <- nhanes %>%
    filter(splt == "train")#%>%
#    select(-SEQN, -splt, -SDDSRVYR)

test <- nhanes %>%
    filter(splt == "test")#%>%
#    select(-SEQN, -splt, -SDDSRVYR)
train <- train[,-c(1,3,ncol(train))]
test <- test[,-c(1,3,ncol(test))]

index_train <- which(nhanes$splt=="train")
index_test <- which(nhanes$splt=="test")

logis <- glm(formula = mortstat ~ .,
             family = binomial,
             data = train,
             na.action = na.omit) #NAs were already removed
```

When λ is zero, we should obtain the same results with linear, ridge and lasso regression.

```r
library(glmnet, help)

#convert response variable to vector
y_train <- train$mortstat
mod_mat <- model.matrix(object = mortstat ~ ., data = train)
#mod_mat %>% head

rid_cv <- cv.glmnet(mod_mat, y_train, alpha = 0)
rid_lam <- rid_cv$lambda.min
rid_mod <- glmnet(x = mod_mat,
                  y = y_train,
                  family = "binomial",
                  alpha = 0,
                  lambda = rid_lam)

las_cv <- cv.glmnet(mod_mat, y_train, alpha = 1)
las_lam <- las_cv$lambda.min
las_mod <- glmnet(x = mod_mat,
                  y = y_train,
```

```r
                  family = "binomial",
                  alpha = 1,
                  lambda = las_lam)

#make predictions
y_test <- test$mortstat
prob_logis <- predict(object = logis, newdata = test, type = "response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```r
pred_logis <- rep(0,length(y_test))
pred_logis[prob_logis>.5]=1
table(pred_logis,y_test)
```

```
##           y_test
## pred_logis   0   1
##          0 325  81
##          1  60  86
```

```r
#prediction accuracy
mean(pred_logis==y_test)*100
```

```
## [1] 74.45652
```

```r
#missclassification rate
(1-mean(pred_logis==y_test))*100
```

```
## [1] 25.54348
```

```r
test_mat <- model.matrix(mortstat ~ ., data = test)

rid_prob <- predict(rid_mod, s = rid_lam, newx = test_mat, type = "response")
rid_pred <- rep(0, length(y_test))
rid_pred[rid_prob>.5]=1
table(rid_pred,y_test)
```

```
##         y_test
## rid_pred   0   1
##        0 339  87
##        1  46  80
```

```r
#prediction accuracy
mean(rid_pred==y_test)*100
```

```
## [1] 75.9058
```

```r
#missclassification rate
(1-mean(rid_pred==y_test))*100
```

```
## [1] 24.0942
```

```r
las_prob <- predict(las_mod, s = las_lam, newx = test_mat, type = "response")
las_pred <- rep(0, length(y_test))
las_pred[las_prob>.5]=1
table(las_pred,y_test)
```

```
##         y_test
## las_pred   0   1
```

```
##        0 337  82
##        1  48  85
```

```r
#prediction accuracy
mean(las_pred==y_test)*100
```

```
## [1] 76.44928
```

```r
#missclassification rate
(1-mean(las_pred==y_test))*100
```

```
## [1] 23.55072
```

```r
exp(coef(logis))
```

```
##  (Intercept)       BPQ010     SDDSRVYR     RIDSTATR     RIAGENDR
## 1.338503e+04 9.914407e-01           NA 4.182160e-01 6.189123e-01
##     RIDAGEYR     RIDRETH1     RIDRETH2     DMQMILIT      DMDBORN
## 1.114684e+00 8.563293e-01 9.264521e-01 8.634467e-01 7.101031e-01
##     DMDCITZN     DMDEDUC2      DMDEDUC     DMDHHSIZ     DMDHRGND
## 1.040911e+00 1.150674e+00 8.766823e-01 8.670911e-01 1.234636e+00
##     DMDHRAGE      SIALANG     SIAPROXY     SIAINTRP     WTINT2YR
## 9.780151e-01 7.504155e-01 3.507126e-01 1.028543e+00 9.997589e-01
##     WTMEC2YR      SDMVPSU     SDMVSTRA       DIQ010       DIQ050
## 9.999094e-01 7.683430e-01 1.004613e+00 7.731992e-01 4.071165e-01
##       DIQ090       DIQ100       DIQ120       DIQ140      HSAQUEX
## 3.347229e-01 1.271025e+00 8.991786e-01 7.080853e-01           NA
##       MCQ010       MCQ053       MCQ092       MCQ140      MCQ160A
## 8.551744e-01 7.069943e-01 6.595456e-01 7.384368e-01 9.375922e-01
##      MCQ160B      MCQ160C      MCQ160D      MCQ160E      MCQ160F
## 5.564638e-01 9.781467e-01 1.117502e+00 8.091150e-01 8.140195e-01
##      MCQ160G      MCQ160J      MCQ160K      MCQ160L      MCQ160M
## 3.709946e-01 9.540182e-01 6.910058e-01 8.340894e-01 1.267628e+00
##       MCQ220      MCQ245A      MCQ250A      MCQ250B      MCQ250C
## 9.035755e-01 1.623616e+00 1.012003e+00 1.014368e+00 9.102191e-01
##      MCQ250E      MCQ250F      MCQ250G       MCQ265       SSQ011
## 1.040810e+00 9.447993e-01 9.712902e-01 8.398450e-01 1.200370e+00
##       SSQ051       SSQ061       WHQ030       WHQ040       WHQ090
## 1.003854e+00 1.013731e+00 1.326810e+00 8.088459e-01 1.386947e+00
```

```r
exp(predict(rid_mod, s = rid_lam, exact = T, type = 'coefficients'))
```

```
## 61 x 1 Matrix of class "dgeMatrix"
##                       1
## (Intercept) 798.2331029
## (Intercept)   1.0000000
## BPQ010        0.9810742
## SDDSRVYR      1.0000000
## RIDSTATR      0.5828499
## RIAGENDR      0.7976910
## RIDAGEYR      1.0527695
## RIDRETH1      0.9449378
## RIDRETH2      0.9302608
## DMQMILIT      0.7975303
## DMDBORN       0.8457229
## DMDCITZN      0.9438924
## DMDEDUC2      1.0108619
```

```
## DMDEDUC       1.0010844
## DMDHHSIZ      0.9435363
## DMDHRGND      1.1481368
## DMDHRAGE      1.0071956
## SIALANG       0.8527956
## SIAPROXY      0.4777184
## SIAINTRP      1.0378410
## WTINT2YR      0.9998252
## WTMEC2YR      0.9998780
## SDMVPSU       0.8655845
## SDMVSTRA      0.9995056
## DIQ010        0.8305113
## DIQ050        0.5706851
## DIQ090        0.4975548
## DIQ100        1.0983771
## DIQ120        0.9331856
## DIQ140        0.8475899
## HSAQUEX       1.0000000
## MCQ010        0.8968256
## MCQ053        0.7290153
## MCQ092        0.7486695
## MCQ140        0.7709688
## MCQ160A       0.9219354
## MCQ160B       0.6075462
## MCQ160C       0.9250233
## MCQ160D       1.0072005
## MCQ160E       0.8560984
## MCQ160F       0.8225180
## MCQ160G       0.4742283
## MCQ160J       1.0411908
## MCQ160K       0.7875970
## MCQ160L       0.9486981
## MCQ160M       1.1266722
## MCQ220        0.8755903
## MCQ245A       1.5586635
## MCQ250A       1.0299147
## MCQ250B       1.0238114
## MCQ250C       0.9229038
## MCQ250E       1.0556966
## MCQ250F       0.9989832
## MCQ250G       1.0022665
## MCQ265        0.9322941
## SSQ011        1.0901229
## SSQ051        0.9924645
## SSQ061        1.0080422
## WHQ030        1.1628070
## WHQ040        0.9650176
## WHQ090        1.2782109
```

```r
exp(predict(las_mod, s = las_lam, exact = T, type = 'coefficients'))
```

```
## 61 x 1 Matrix of class "dgeMatrix"
##                         1
## (Intercept) 299.4966713
## (Intercept)   1.0000000
```

```
## BPQ010         1.0000000
## SDDSRVYR       1.0000000
## RIDSTATR       0.4770661
## RIAGENDR       0.6910605
## RIDAGEYR       1.0981468
## RIDRETH1       0.9964678
## RIDRETH2       0.9994903
## DMQMILIT       0.8647099
## DMDBORN        0.7455959
## DMDCITZN       1.0000000
## DMDEDUC2       1.0156403
## DMDEDUC        1.0000000
## DMDHHSIZ       0.9385592
## DMDHRGND       1.1201018
## DMDHRAGE       0.9942549
## SIALANG        0.8820657
## SIAPROXY       0.4880736
## SIAINTRP       1.0000000
## WTINT2YR       0.9998336
## WTMEC2YR       0.9999115
## SDMVPSU        0.8574278
## SDMVSTRA       1.0000000
## DIQ010         0.8135888
## DIQ050         0.4941549
## DIQ090         0.3954753
## DIQ100         1.0169715
## DIQ120         0.9912958
## DIQ140         0.8126362
## HSAQUEX        1.0000000
## MCQ010         0.9342817
## MCQ053         0.7751238
## MCQ092         0.6977869
## MCQ140         0.7876512
## MCQ160A        0.9924716
## MCQ160B        0.5850207
## MCQ160C        1.0000000
## MCQ160D        1.0000000
## MCQ160E        0.8876824
## MCQ160F        0.8956696
## MCQ160G        0.3983468
## MCQ160J        1.0000000
## MCQ160K        0.7320843
## MCQ160L        0.9577582
## MCQ160M        1.0929697
## MCQ220         0.9534763
## MCQ245A        1.4699759
## MCQ250A        1.0000000
## MCQ250B        1.0000000
## MCQ250C        0.9548581
## MCQ250E        1.0000000
## MCQ250F        1.0000000
## MCQ250G        1.0000000
## MCQ265         0.9157886
## SSQ011         1.0478959
```

```
## SSQ051        1.0000000
## SSQ061        1.0089709
## WHQ030        1.1587417
## WHQ040        0.9731150
## WHQ090        1.2478374
```

```r
#head(train)
library(MASS, help)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
#fit_lda <- lda(mortstat ~ ., data = train)
#fit_lda
#pred_lda <- predict(fit_lda, Weekly_20092010)
#table(pred_lda$class, morstat_20092010)
#
#fit_qda <- qda(mortstat ~ ., data = Weekly, subset = train)
#fit_qda
#pred_qda <- predict(fit_qda, Weekly_20092010)
#table(pred_qda$class, morstat_20092010)
```