Data Science II Midterm Project Analysis

Camille Okonkwo

Contents

Background **3**

Data **3**

Data Preparation **3**

Exploratory analysis and data visualization **4**
Descriptive Statistics Table . . . 4
Feature Plot . . . 6
Correlation Matrix of training data . . . 8

Test and Train Data Preparation **9**

Model Fitting in `caret` **11**
Linear Model . . . 11
KNN . . . 12
Ridge Regression . . . 12
Lasso . . . 14
Elastic Net . . . 15
PCR . . . 16
PLS . . . 17
GAM . . . 18
MARS . . . 21

Model Comparison **23**

```
library(tidymodels)
library(splines)
library(caret)
library(glmnet)
library(table1)
library(kableExtra)
```

Background

To gain a better understanding of the factors that predict recovery time from COVID-19 illness, a study was designed to combine three existing cohort studies that have been tracking participants for several years. The study collects recovery information through questionnaires and medical records, and leverages existing data on personal characteristics prior to the pandemic. The ultimate goal is to develop a prediction model for recovery time and identify important risk factors for long recovery time.

Data

The dataset in `recovery.RData` includes data from 3000 participants.

Here is a description of each variable:

- ID (`id`): Participant ID
- Gender (`gender`): 1 = Male, 0 = Female
- Race/ethnicity (`race`): 1 = White, 2 = Asian, 3 = Black, 4 = Hispanic
- Smoking (`smoking`): Smoking status; 0 = Never smoked, 1 = Former smoker, 2 = Current smoker
- Height (`height`): Height (in centimeters)
- Weight (`weight`): Weight (in kilograms)
- BMI (`bmi`): Body Mass Index; BMI = weight (in kilograms) / height (in meters) squared
- Hypertension (`hypertension`): 0 = No, 1 = Yes
- Diabetes (`diabetes`): 0 = No, 1 = Yes
- Systolic blood pressure (`SBP`): Systolic blood pressure (in mm/Hg)
- LDL cholesterol (`LDL`): LDL (low-density lipoprotein) cholesterol (in mg/dL)
- Vaccination status at the time of infection (`vaccine`): 0 = Not vaccinated, 1 = Vaccinated
- Severity of COVID-19 infection (`severity`): 0 = Not severe, 1= Severe
- Study (`study`): The study (A/B) that the participant belongs to
- Time to recovery (`recovery_time`): Time from COVID-19 infection to recovery in days

Data Preparation

Partition the dataset into two parts: training data (80%) and test data (20%) with `tidymodels`.

```
load("data/recovery.RData")

dat = dat |>
  select(-id)

# matrix of predictors & vector of response for data set exploration
x.dat = model.matrix(recovery_time ~., dat)[, -1]
y.dat = dat$recovery_time
```

Exploratory analysis and data visualization

```
dat_ds <- dat |>
  mutate(across(.fns = as.factor)) |>
  rename_with(~str_to_title(.x), everything()) |>
  mutate(
    Age = as.numeric(Age),
    Gender = factor(Gender,
                    levels = c(0, 1),
                    labels = c("Female", "Male")),
    `Race/Ethnicity` = factor(Race,
                              levels = c(1, 2, 3, 4),
                              labels = c("White", "Asian", "Black", "Hispanic")),
    `Smoking status` = factor(Smoking,
                              levels = c(0, 1, 2),
                              labels = c("Never smoked", "Former smoker", "Current smoker")),
    Height = as.numeric(Height),
    Weight = as.numeric(Weight),
    `Body Mass Index` = as.numeric(Bmi),
    Hypertension = factor(Hypertension,
                          levels = c(0, 1),
                          labels = c("No", "Yes")),
    Diabetes = factor(Diabetes,
                      levels = c(0, 1),
                      labels = c("No", "Yes")),
    `Systolic Blood Pressure` = as.numeric(Sbp),
    `Low-density lipoprotein cholesterol` = as.numeric(Ldl),
    `Vaccination status at the time of infection` = factor(Vaccine,
                                                           levels = c(0, 1),
                                                           labels = c("Not vaccinated", "Vaccinated")),
    `Severity of COVID-19 infection` = factor(Severity,
                                              levels = c(0, 1),
                                              labels = c("Not severe", "Severe")),
    `Time from COVID-19 infection to recovery` = as.numeric(Recovery_time),
    Study = factor(Study,
                   levels = c("A", "B"),
                   labels = c("Study A", "Study B"))
    )
```

Descriptive Statistics Table

```
library(summarytools)

st_options(plain.ascii = FALSE,
           style = "rmarkdown",
           dfSummary.silent = TRUE,
           footnote = NA,
           subtitle.emphasis = FALSE)

dfSummary(dat)
```

```
## ### Data Frame Summary
## **dat**
```

```
## **Dimensions:** 3000 x 15
## **Duplicates:** 0
##
## ---------------------------------------------------------------------------------------------------------
## No   Variable         Stats / Values              Freqs (% of Valid)    Graph                    Vali
## ---- ---------------- --------------------------- --------------------- ------------------------ ----
## 1    age\             Mean (sd) : 60.2 (4.5)\     34 distinct values    \ \ \ \ \ \ : .\         3000
##      [numeric]        min < med < max:\                                 \ \ \ \ \ \ : :\         (100
##                       42 < 60 < 79\                                     \ \ \ \ \ \ : :\
##                       IQR (CV) : 6 (0.1)                                \ \ \ \ . : : .\
##                                                                         \ \ \ \ : : : :
##
## 2    gender\          Min  : 0\                   0 : 1544 (51.5%)\     IIIIIIIIII \             3000
##      [integer]        Mean : 0.5\                 1 : 1456 (48.5%)      IIIIIIIII                (100
##                       Max  : 1
##
## 3    race\            1\. 1\                      1967 (65.6%)\         IIIIIIIIIIII \           3000
##      [factor]         2\. 2\                      158 ( 5.3%)\          I \                      (100
##                       3\. 3\                      604 (20.1%)\          IIII \
##                       4\. 4                       271 ( 9.0%)           I
##
## 4    smoking\         1\. 0\                      1822 (60.7%)\         IIIIIIIIIII \            3000
##      [factor]         2\. 1\                      859 (28.6%)\          IIIII \                  (100
##                       3\. 2                       319 (10.6%)           II
##
## 5    height\          Mean (sd) : 169.9 (6)\      313 distinct values   \ \ \ \ \ \ \ \ : :\    3000
##      [numeric]        min < med < max:\                                 \ \ \ \ \ \ \ \ : :\    (100
##                       147.8 < 169.9 < 188.6\                            \ \ \ \ \ \ . : : .\
##                       IQR (CV) : 7.9 (0)                                \ \ \ \ \ \ : : : :\
##                                                                         \ \ \ \ . : : : : : .
##
## 6    weight\          Mean (sd) : 80 (7.1)\       364 distinct values   \ \ \ \ \ \ \ \ : .\    3000
##      [numeric]        min < med < max:\                                 \ \ \ \ \ \ \ \ : :\    (100
##                       55.9 < 79.8 < 103.7\                              \ \ \ \ \ \ : : : :\
##                       IQR (CV) : 9.6 (0.1)                              \ \ \ \ . : : : : : .\
##                                                                         \ \ . : : : : : : : .
##
## 7    bmi\             Mean (sd) : 27.8 (2.8)\     163 distinct values   \ \ \ \ \ \ . :\        3000
##      [numeric]        min < med < max:\                                 \ \ \ \ \ \ : : :\      (100
##                       18.8 < 27.6 < 38.9\                               \ \ \ \ \ \ : : :\
##                       IQR (CV) : 3.7 (0.1)                              \ \ \ \ : : : : :\
##                                                                         \ \ . : : : : : : .
##
## 8    hypertension\    Min  : 0\                   0 : 1508 (50.3%)\     IIIIIIIIII \             3000
##      [numeric]        Mean : 0.5\                 1 : 1492 (49.7%)      IIIIIIIII                (100
##                       Max  : 1
##
## 9    diabetes\        Min  : 0\                   0 : 2537 (84.6%)\     IIIIIIIIIIIIIIII \       3000
##      [integer]        Mean : 0.2\                 1 :  463 (15.4%)      III                      (100
##                       Max  : 1
##
## 10   SBP\             Mean (sd) : 130.5 (8)\      52 distinct values    \ \ \ \ \ \ \ \ : .\    3000
##      [numeric]        min < med < max:\                                 \ \ \ \ \ \ \ \ : : .\  (100
##                       105 < 130 < 156\                                  \ \ \ \ \ \ : : : :\
```

```
##                       IQR (CV) : 11 (0.1)                              \ \ \ \ . : : : : : .\
##                                                                        \ \ . : : : : : : : .
##
## 11   LDL\             Mean (sd) : 110.5 (19.8)\   114 distinct values   \ \ \ \ \ \ \ \ \ \ :\   300
##      [numeric]        min < med < max:\                                \ \ \ \ \ \ \ \ : : .\   (10
##                       28 < 110 < 178\                                  \ \ \ \ \ \ \ \ : : :\
##                       IQR (CV) : 27 (0.2)                              \ \ \ \ \ \ . : : : .\
##                                                                        \ \ \ \ . : : : : : : .
##
## 12   vaccine\         Min  : 0\                   0 : 1212 (40.4%)\    IIIIIIII \               300
##      [integer]        Mean : 0.6\                 1 : 1788 (59.6%)     IIIIIIIIIII              (10
##                       Max  : 1
##
## 13   severity\        Min  : 0\                   0 : 2679 (89.3%)\    IIIIIIIIIIIIIIIII \      300
##      [integer]        Mean : 0.1\                 1 :  321 (10.7%)     II                       (10
##                       Max  : 1
##
## 14   study\           1\. A\                      2000 (66.7%)\        IIIIIIIIIIII \           300
##      [character]      2\. B                       1000 (33.3%)         IIIIII                   (10
##
## 15   recovery_time\   Mean (sd) : 42.2 (23.2)\    140 distinct values  : :\                     300
##      [numeric]        min < med < max:\                                : :\                     (10
##                       2 < 39 < 365\                                    : :\
##                       IQR (CV) : 18 (0.5)                              : :\
##                                                                        : : .
## ---------------------------------------------------------------------------------------------------
```

```
library(table1)
library(kableExtra)

units(dat_ds$Height) <- "cm"
units(dat_ds$Weight) <- "kg"
units(dat_ds$`Body Mass Index`) <- "kg/m^2"
units(dat_ds$`Systolic Blood Pressure`) <- "mm/Hg"
units(dat_ds$`Low-density lipoprotein cholesterol`) <- "mg/dL"
units(dat_ds$`Time from COVID-19 infection to recovery`) <- "days"

descriptive_table <- table1(~ Age + Gender + `Race/Ethnicity` + `Smoking status` + Height + Weight + `B
                            data = dat_ds,
                            overall = "Total",
                            caption = "Descriptive Statistics")

t1kable(descriptive_table)
```

There are no missing values in the dataset. The distribution of the demographic variables `age`, `gender`, `race` are about the same between treatment groups. Mean `height`, `weight`, `BMI`, `SBP` and `LDL` variables are also similarly distributed between groups. There are more people who are vaccinated than not vaccinated in study group A and B, and also there are more participants who are reported to have not severe COVID-19 infections. `recovery_time` mean and SD is higher for Study B. There is also a larger interval range.

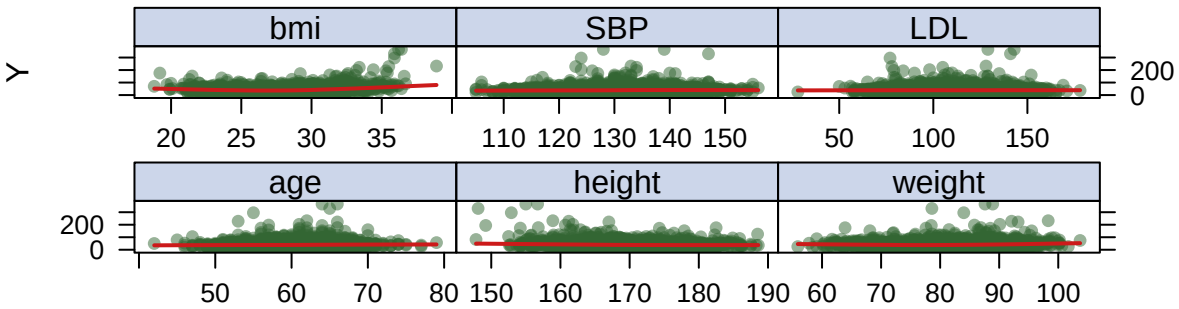
Feature Plot

```
theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(.2, .4, .2, .5)
```

Table 1: Descriptive Statistics

	Study A	Study B	Total
	(N=2000)	(N=1000)	(N=3000)
Age			
Mean (SD)	17.2 (4.52)	17.2 (4.38)	17.2 (4.47)
Median [Min, Max]	17.0 [1.00, 34.0]	17.0 [2.00, 33.0]	17.0 [1.00, 34.0]
Gender			
Female	1036 (51.8%)	508 (50.8%)	1544 (51.5%)
Male	964 (48.2%)	492 (49.2%)	1456 (48.5%)
Race/Ethnicity			
White	1312 (65.6%)	655 (65.5%)	1967 (65.6%)
Asian	108 (5.4%)	50 (5.0%)	158 (5.3%)
Black	408 (20.4%)	196 (19.6%)	604 (20.1%)
Hispanic	172 (8.6%)	99 (9.9%)	271 (9.0%)
Smoking status			
Never smoked	1225 (61.3%)	597 (59.7%)	1822 (60.7%)
Former smoker	557 (27.9%)	302 (30.2%)	859 (28.6%)
Current smoker	218 (10.9%)	101 (10.1%)	319 (10.6%)
Height (cm)			
Mean (SD)	160 (58.8)	161 (59.1)	160 (58.9)
Median [Min, Max]	160 [1.00, 313]	161 [2.00, 312]	160 [1.00, 313]
Weight (kg)			
Mean (SD)	181 (70.0)	182 (70.5)	182 (70.2)
Median [Min, Max]	178 [1.00, 364]	182 [3.00, 358]	180 [1.00, 364]
Body Mass Index (kg/m^2)			
Mean (SD)	77.6 (27.5)	77.6 (28.3)	77.6 (27.8)
Median [Min, Max]	77.0 [1.00, 162]	76.0 [2.00, 163]	76.5 [1.00, 163]
Hypertension			
No	998 (49.9%)	510 (51.0%)	1508 (50.3%)
Yes	1002 (50.1%)	490 (49.0%)	1492 (49.7%)
Diabetes			
No	1678 (83.9%)	859 (85.9%)	2537 (84.6%)
Yes	322 (16.1%)	141 (14.1%)	463 (15.4%)
Systolic Blood Pressure (mm/Hg)			
Mean (SD)	26.6 (8.02)	26.3 (7.88)	26.5 (7.97)
Median [Min, Max]	27.0 [1.00, 52.0]	26.0 [1.00, 51.0]	26.0 [1.00, 52.0]
Low-density lipoprotein cholesterol (mg/dL)			
Mean (SD)	58.3 (19.7)	58.7 (19.7)	58.4 (19.7)
Median [Min, Max]	58.0 [1.00, 114]	58.0 [3.00, 112]	58.0 [1.00, 114]
Vaccination status at the time of infection			
Not vaccinated	797 (39.9%)	415 (41.5%)	1212 (40.4%)
Vaccinated	1203 (60.2%)	585 (58.5%)	1788 (59.6%)
Severity of COVID-19 infection			
Not severe	1785 (89.3%)	894 (89.4%)	2679 (89.3%)
Severe	215 (10.8%)	106 (10.6%)	321 (10.7%)
Time from COVID-19 infection to recovery (days)			
Mean (SD)	39.4 (11.1)	42.8 (28.1)	40.5 (18.7)
Median [Min, Max]	39.0 [9.00, 107]	36.0 [1.00, 140]	38.0 [1.00, 140]

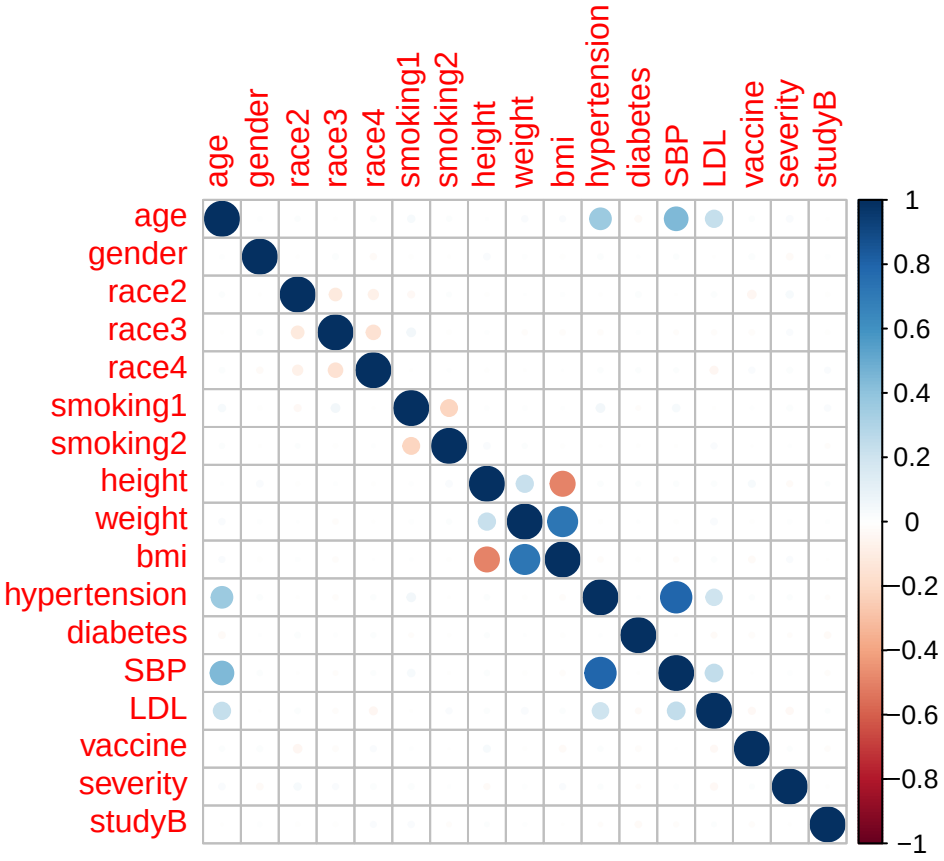
```
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <- rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(.0, .2, .6, .2)
trellis.par.set(theme1)

featurePlot(
  x.dat[, -c(2, 3, 4, 5, 6, 7, 11, 12, 15, 16, 17) ],
  y.dat,
  plot = "scatter",
  labels = c("", "Y"),
  type = c("p", "smooth"),
  layout = c(3, 3))
```



Seems to be mostly linear, with some outliers. I will try and remove some to get a more optimal fit.

Correlation Matrix of training data

```
library(corrplot)
corrplot(cor(x.dat), method = "circle", type = "full")
```



There may be multicollinearity between some predictors, so when fitting models I may consider a GAM or MARS.

```
outlier_coeff = 2

# Calculate upper and lower bounds for outliers
outlier_high = mean(dat$recovery_time) + outlier_coeff/2 * sd(dat$recovery_time)
outlier_low = mean(dat$recovery_time) - outlier_coeff/2 * sd(dat$recovery_time)

# Remove outliers based on the calculated bounds
dat2 = dat %>%
  filter(recovery_time <= outlier_high & recovery_time >= outlier_low)
```

Test and Train Data Preparation

```
set.seed(2)

# create a random split of 80% training and 20% test data
data_split <- initial_split(data = dat2, prop = 0.8)

# partitioned datasets
training_data = training(data_split)
testing_data = testing(data_split)

# training data
x <- model.matrix(recovery_time ~ ., training_data)[, -1] # matrix of predictors
```

```r
head(x)
```

```
##   age gender race2 race3 race4 smoking1 smoking2 height weight  bmi
## 1  62      0     0     0     0        0        0  178.2   82.0 25.8
## 2  58      0     0     0     0        0        0  178.0   78.3 24.7
## 3  53      0     0     0     1        0        0  167.3   70.4 25.2
## 4  53      1     1     0     0        0        0  164.5   89.1 32.9
## 5  54      1     0     0     1        0        0  171.5   87.3 29.7
## 6  65      1     0     0     0        0        0  171.4   78.0 26.5
##   hypertension diabetes SBP LDL vaccine severity studyB
## 1            0        1 126 125       0        0      0
## 2            0        0 126 142       0        0      0
## 3            0        0 119  63       1        0      0
## 4            1        0 136 106       0        0      0
## 5            1        0 139 112       0        0      0
## 6            1        0 137 115       0        0      0
```

```r
y <- training_data$recovery_time # vector of response

# testing data
x2 <- model.matrix(recovery_time ~ .,testing_data)[, -1] # matrix of predictors
y2 <- testing_data$recovery_time # vector of response
```

Model Fitting in caret

```
# setting a 10-fold cross-validation
ctrl <- trainControl(method = "cv",
                     number = 10,
                     selectionFunction = "best")
```

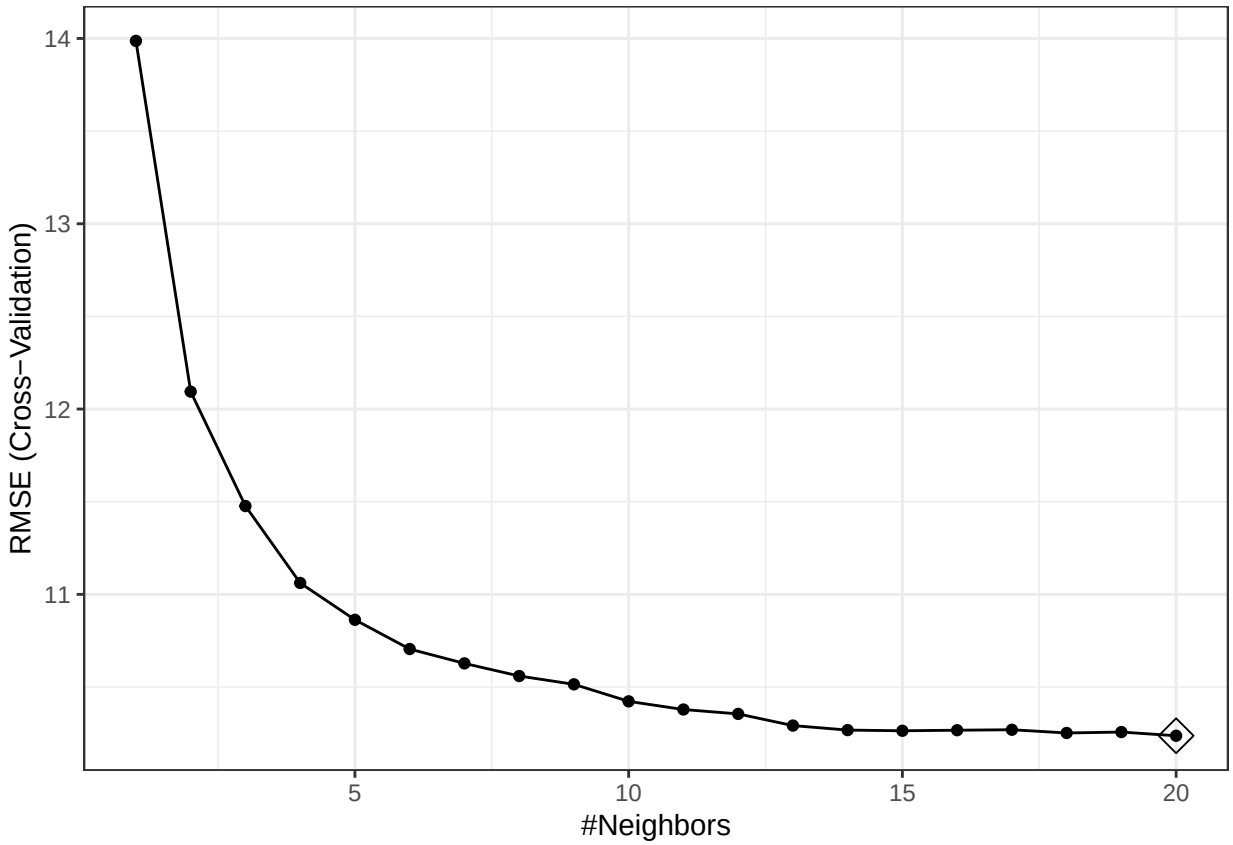
Linear Model

```
set.seed(2)

# fit a linear model
lm.fit <- train(x, y,
                method = "lm",
                trControl = ctrl)

summary(lm.fit)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.005  -7.147  -0.198   6.877  28.856
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -380.71182   71.56444  -5.320 1.15e-07 ***
## age             0.04603    0.05466   0.842 0.399853
## gender         -1.79538    0.43510  -4.126 3.84e-05 ***
## race2           0.54624    0.95865   0.570 0.568880
## race3           0.24300    0.55872   0.435 0.663667
## race4          -1.19334    0.78295  -1.524 0.127627
## smoking1        1.22064    0.49615   2.460 0.013969 *
## smoking2        2.52437    0.72127   3.500 0.000476 ***
## height          2.44068    0.41956   5.817 6.94e-09 ***
## weight         -2.70134    0.44582  -6.059 1.63e-09 ***
## bmi             8.12174    1.28625   6.314 3.33e-10 ***
## hypertension    2.41632    0.71903   3.361 0.000792 ***
## diabetes        0.04836    0.59768   0.081 0.935521
## SBP            -0.01613    0.04684  -0.344 0.730640
## LDL            -0.02665    0.01161  -2.297 0.021743 *
## vaccine        -3.64355    0.44552  -8.178 5.04e-16 ***
## severity        2.25396    0.70096   3.216 0.001323 **
## studyB         -1.25700    0.51058  -2.462 0.013903 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.794 on 2018 degrees of freedom
## Multiple R-squared:  0.09656,    Adjusted R-squared:  0.08894
## F-statistic: 12.69 on 17 and 2018 DF,  p-value: < 2.2e-16
```

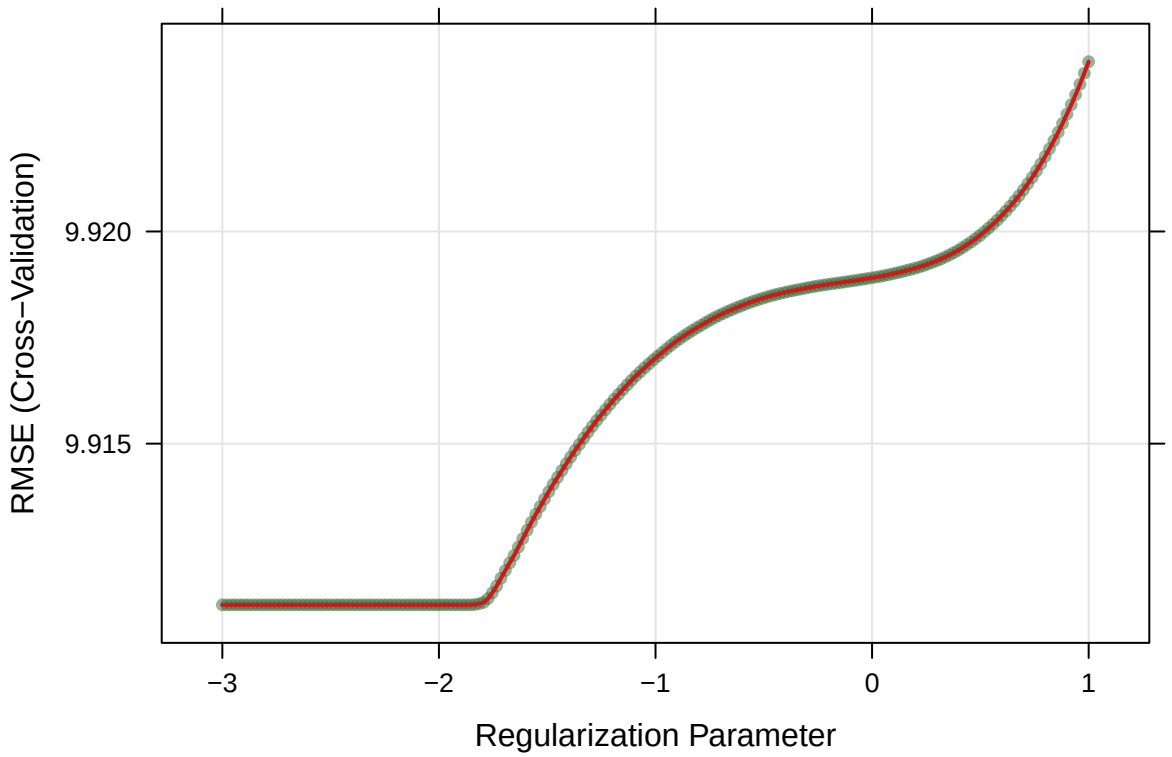
KNN

```
# knn using `caret`
set.seed(2)

knn.fit <- train(x, y,
                 method = "knn",
                 trControl = ctrl,
                 tuneGrid = expand.grid(k = seq(from = 1, to = 20, by = 1)))

ggplot(knn.fit, highlight = TRUE) + theme_bw()
```



Ridge Regression

```
# ridge using `caret`
set.seed(2)

ridge.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                                          lambda = exp(seq(1, -3, length=200))),
                   trControl = ctrl)

plot(ridge.fit, xTrans = log)
```




```
ridge.fit$bestTune
```

```
##    alpha   lambda
## 58     0 0.156567
```

```
# coefficients in the final model
coef(ridge.fit$finalModel, s = ridge.fit$bestTune$lambda)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept)  13.946746030
## age           0.040006500
## gender       -1.718809077
## race2         0.699025921
## race3         0.282581005
## race4        -1.225942285
## smoking1      1.192067800
## smoking2      2.473157695
## height        0.113564170
## weight       -0.223789489
## bmi           0.969222436
## hypertension  2.254588204
## diabetes      0.109136659
## SBP          -0.006327395
## LDL          -0.025291861
## vaccine      -3.511548656
## severity      2.164547963
## studyB       -1.316785639
```

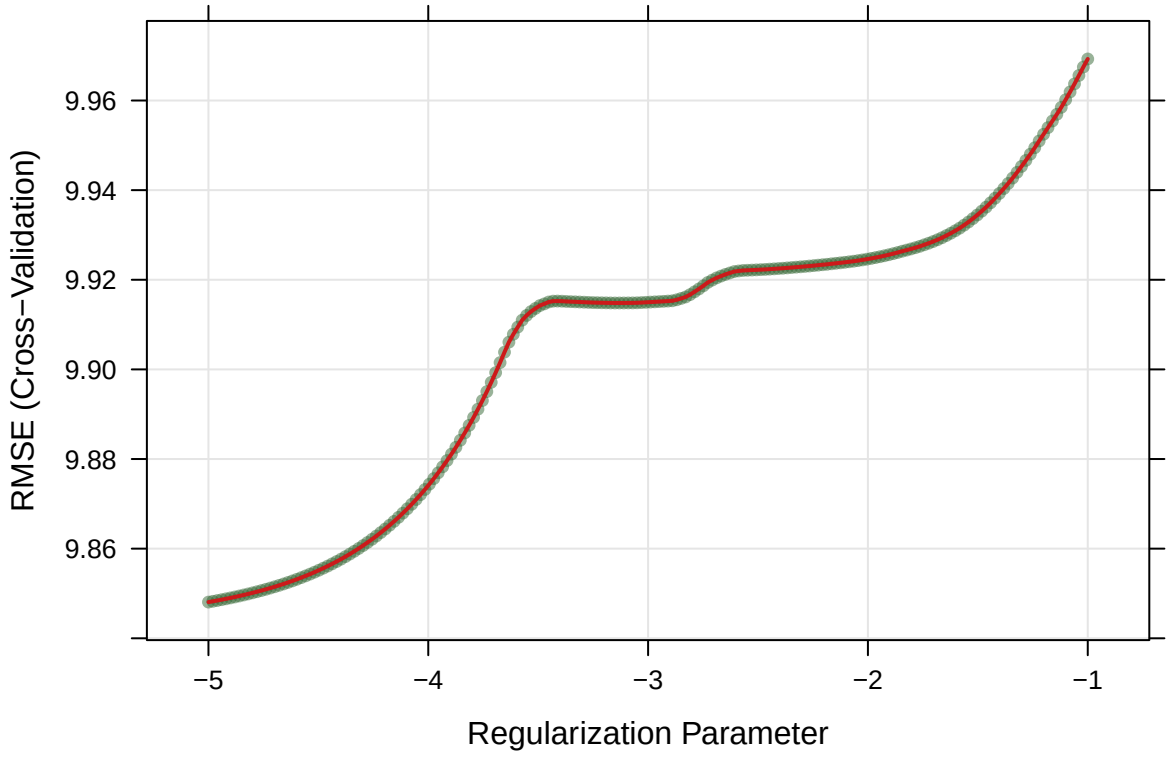
```
ridge.pred <- predict(ridge.fit, newdata = model.matrix(recovery_time ~ ., testing_data)[,-1])
```

```r
# test error
mean((ridge.pred - testing_data[, "recovery_time"])^2)
```

```
## [1] 105.2438
```

Lasso

```r
set.seed(2)

# lasso using caret
lasso.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(-1, -5, length=200))),
                   trControl = ctrl)

plot(lasso.fit, xTrans = log)
```




```r
lasso.fit$bestTune
```

```
##   alpha      lambda
## 1     1 0.006737947
```

```r
# coefficients in the final model
coef(lasso.fit$finalModel, lasso.fit$bestTune$lambda)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)  -274.21117808
## age             0.04185032
## gender         -1.76875935
```

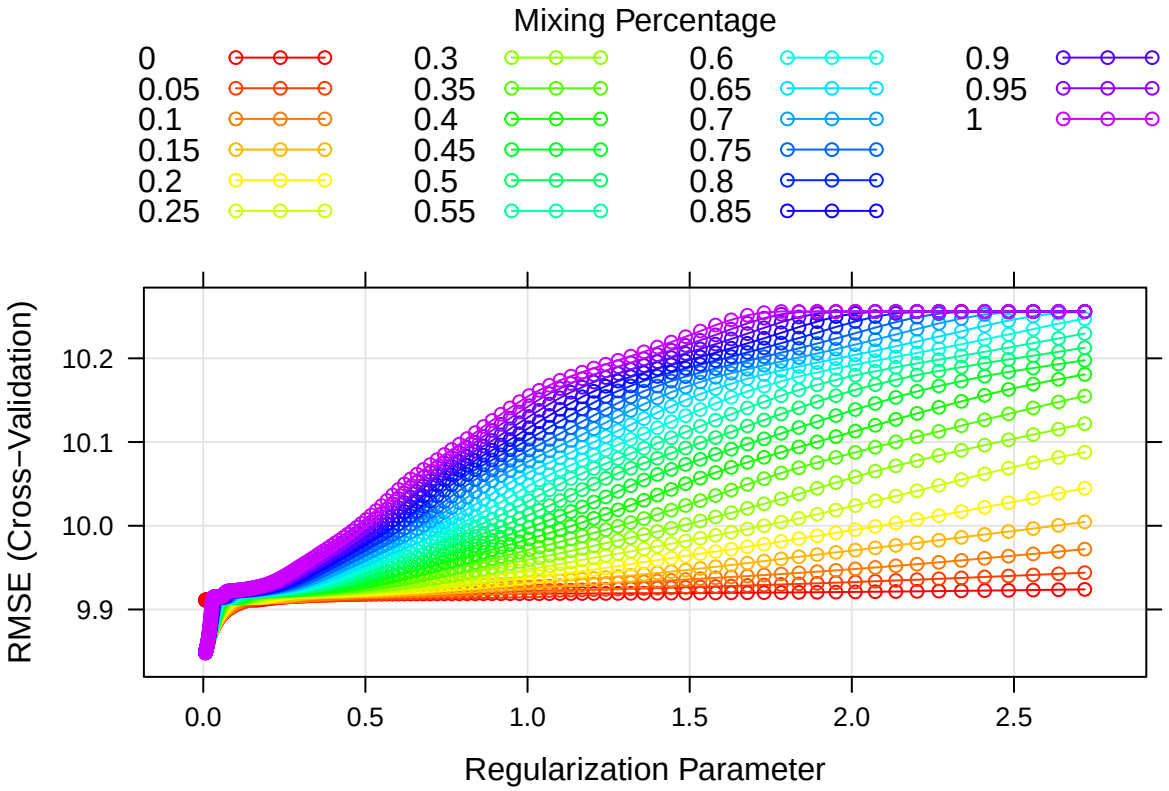
```
## race2           0.55780952
## race3           0.23775836
## race4          -1.18661239
## smoking1        1.20168311
## smoking2        2.49562590
## height          1.81095118
## weight         -2.03143326
## bmi             6.18907370
## hypertension    2.33630344
## diabetes        0.04696404
## SBP            -0.01033364
## LDL            -0.02601451
## vaccine        -3.61006147
## severity        2.22121938
## studyB         -1.26407283
```

Elastic Net

```
set.seed(2)

# elastic net using caret
enet.fit <- train(x, y,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = seq(0, 1, length =
                                                       21),
                                         lambda = exp(seq(1, -5, length=200))),
                  trControl = ctrl)

enet.fit$bestTune
```

```
##      alpha      lambda
## 4001     1 0.006737947
```

```
myCol <- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))
plot(enet.fit, par.settings = myPar)
```




```
# coefficients in the final model
coef(enet.fit$finalModel, enet.fit$bestTune$lambda)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)  -274.21117808
## age             0.04185032
## gender         -1.76875935
## race2           0.55780952
## race3           0.23775836
## race4          -1.18661239
## smoking1        1.20168311
## smoking2        2.49562590
## height          1.81095118
## weight         -2.03143326
## bmi             6.18907370
## hypertension    2.33630344
## diabetes        0.04696404
## SBP            -0.01033364
## LDL            -0.02601451
## vaccine        -3.61006147
## severity        2.22121938
## studyB         -1.26407283
```

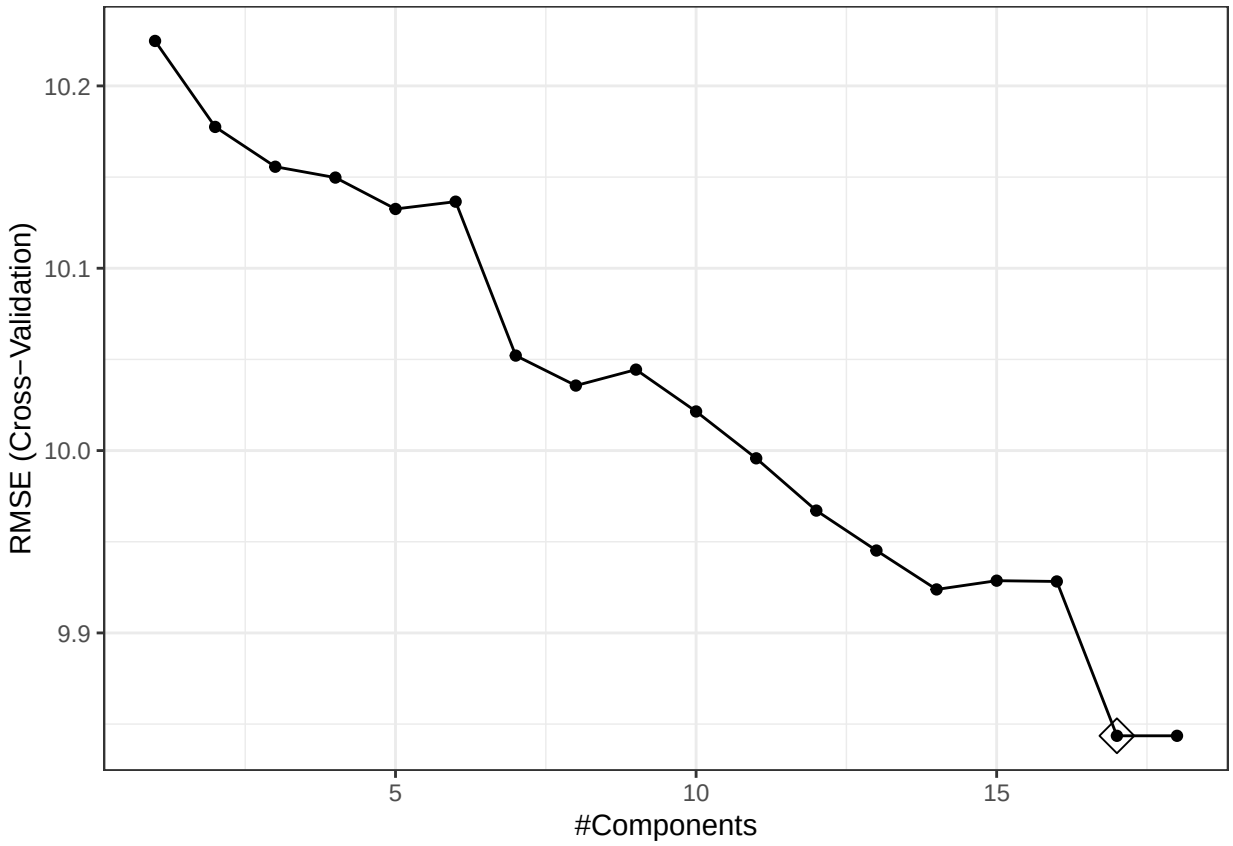
PCR

```
set.seed(2)

# pcr using caret
```

```
pcr.fit <- train(x, y,
                 method = "pcr",
                 tuneGrid = data.frame(ncomp = 1:18),
                 trControl = ctrl,
                 preProcess = c("center", "scale"))

predy2.pcr2 <- predict(pcr.fit, newdata = x2)

mean((y2 - predy2.pcr2)^2)
```

```
## [1] 101.691
```

```
ggplot(pcr.fit, highlight = TRUE) + theme_bw()
```



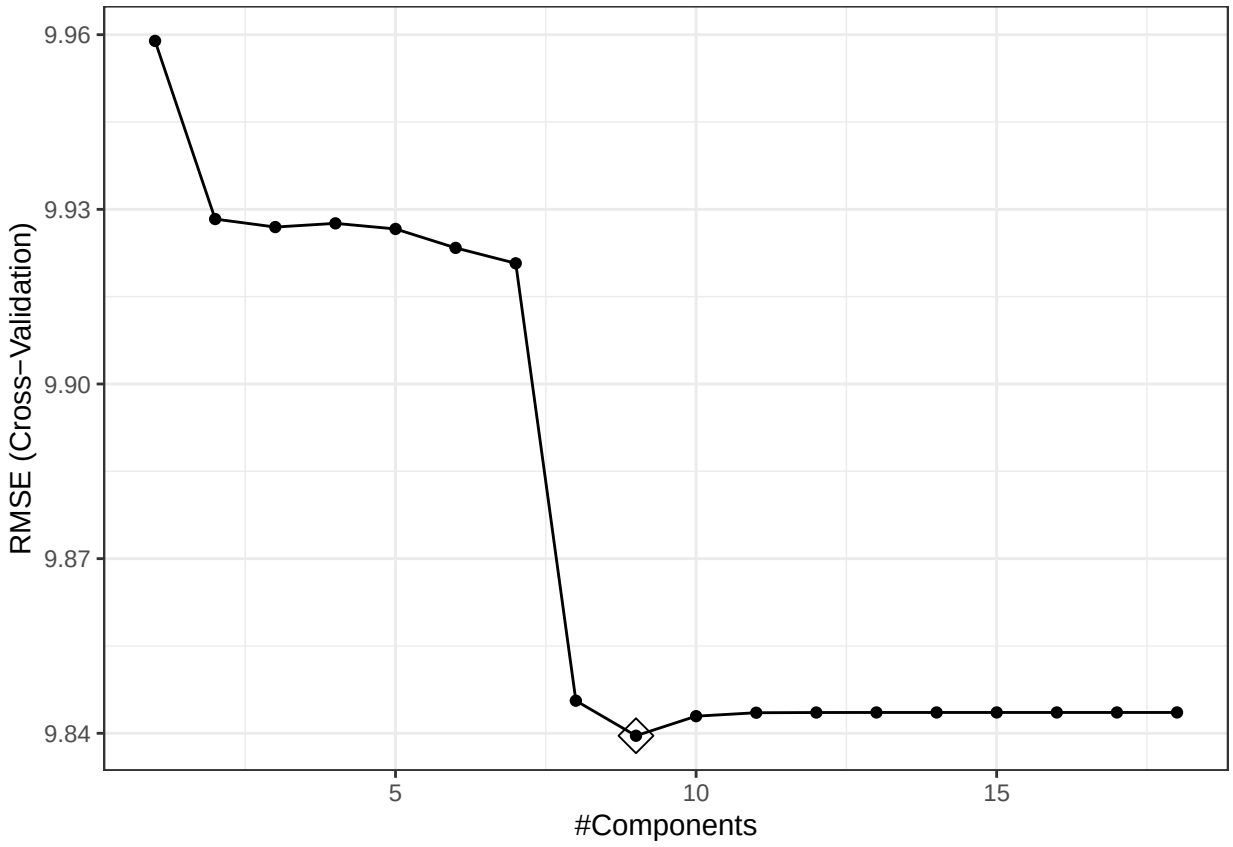
PLS

```
set.seed(2)

# pls using caret
pls.fit <- train(x, y,
                 method = "pls",
                 tuneGrid = data.frame(ncomp = 1:18),
                 trControl = ctrl,
                 preProcess = c("center", "scale"))

predy2.pls2 <- predict(pls.fit, newdata = x2)
```

```
mean((y2 - predy2.pls2)^2)
```

```
## [1] 101.7632
```

```
ggplot(pls.fit, highlight = TRUE) + theme_bw()
```



GAM

```
set.seed(2)

gam.fit <- train(x, y,
                 method = "gam",
                 tuneGrid = data.frame(method = "GCV.Cp",
                                       select = c(TRUE, FALSE)),
                 trControl = ctrl)

gam.fit$bestTune
```

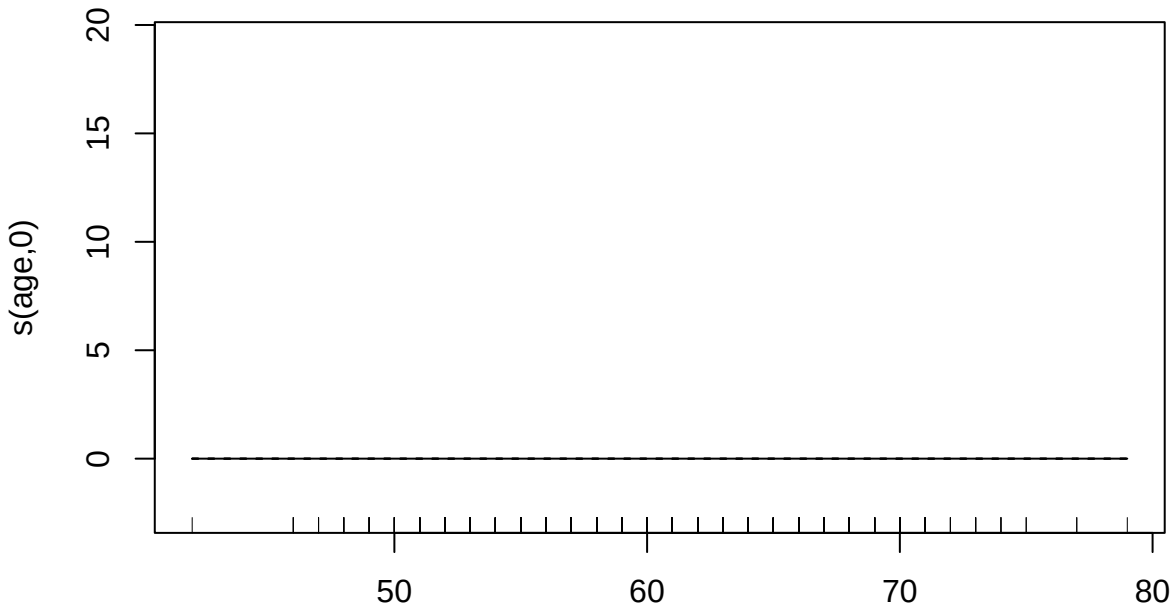
```
##   select method
## 2   TRUE GCV.Cp
```

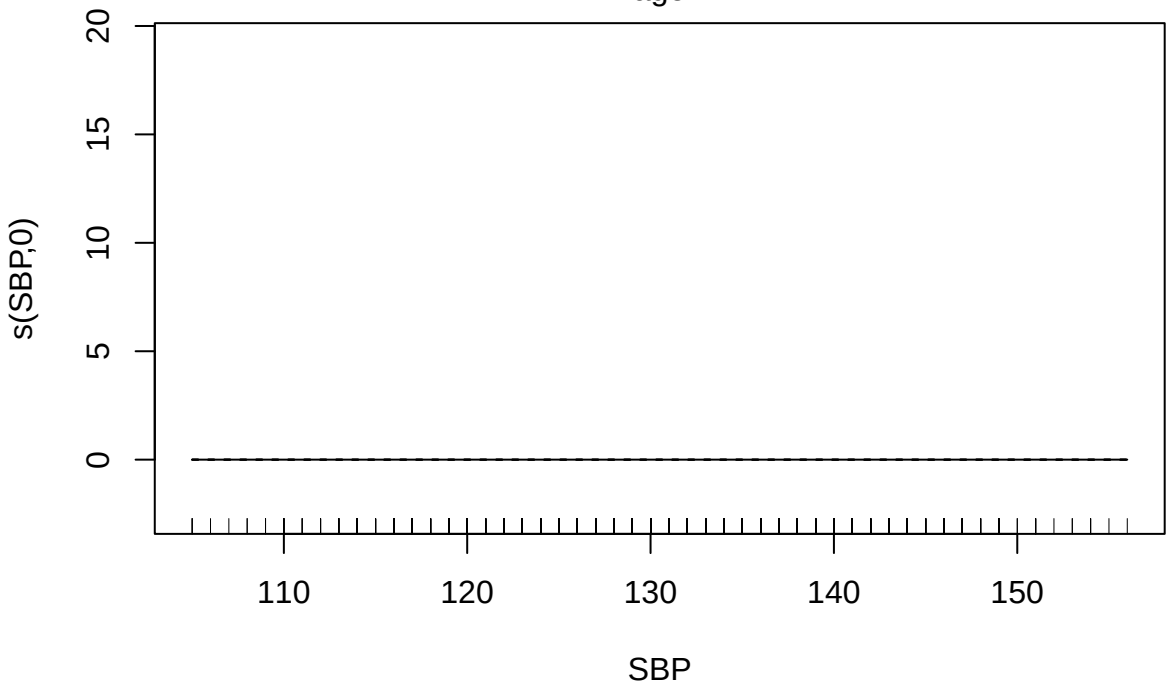
```
gam.fit$finalModel
```

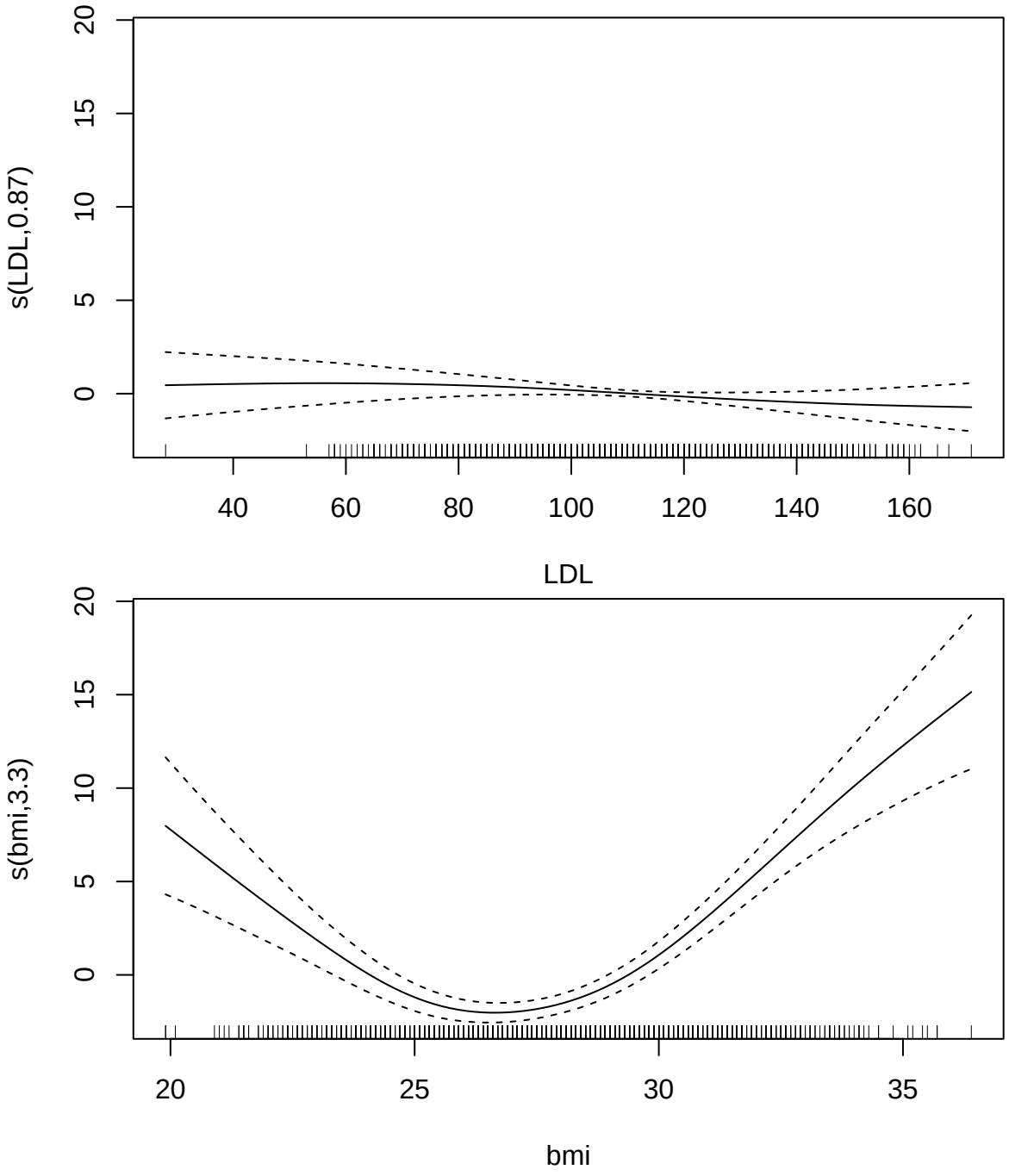
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
```

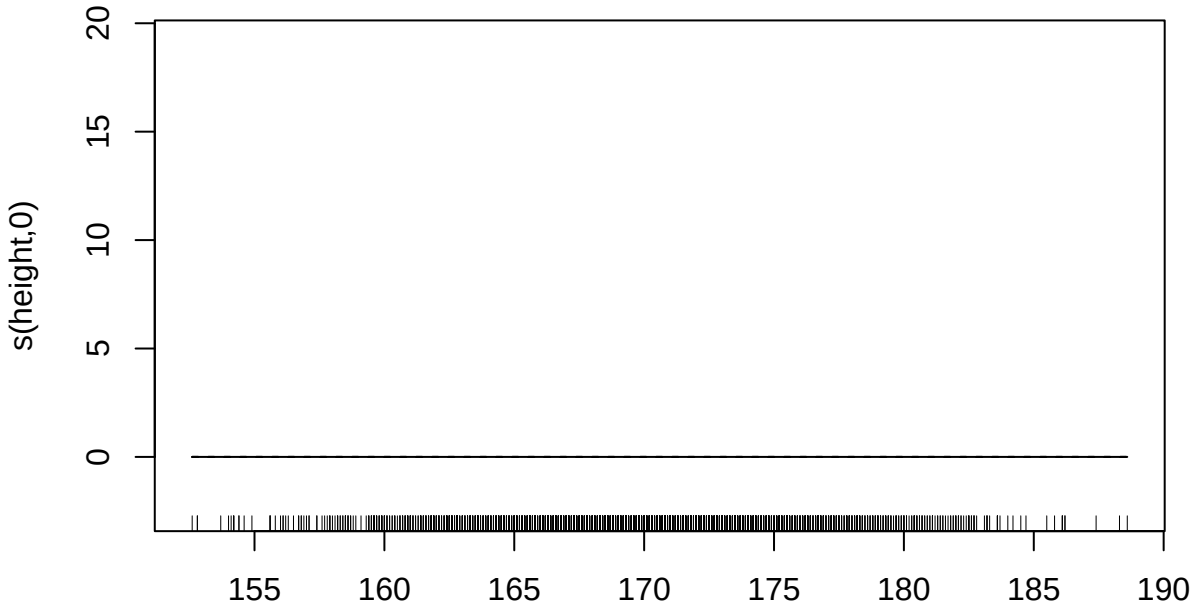
```
## .outcome ~ gender + race2 + race3 + race4 + smoking1 + smoking2 +
##     hypertension + diabetes + vaccine + severity + studyB + s(age) +
##     s(SBP) + s(LDL) + s(bmi) + s(height) + s(weight)
##
## Estimated degrees of freedom:
## 0.000 0.000 0.866 3.295 0.000 1.187  total = 17.35
##
## GCV score: 92.6575
```

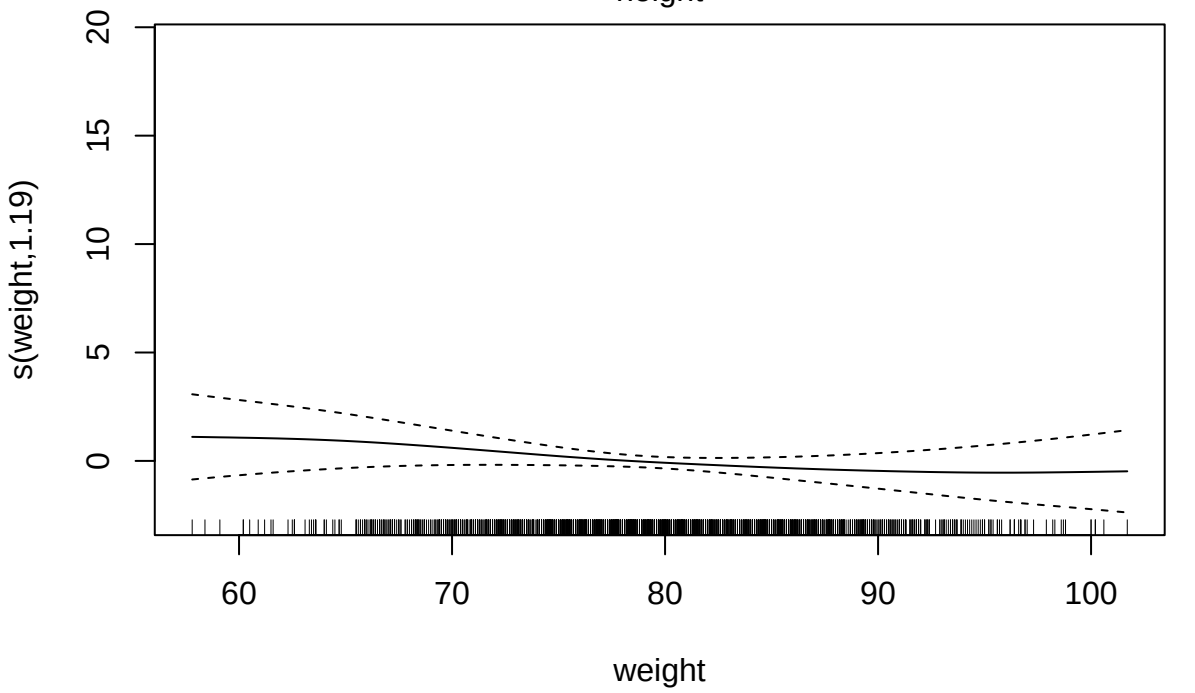
```
plot(gam.fit$finalModel)
```






s(LDL,0.87)
0
5
10
15
20
40
60
80
100
120
140
160
LDL
s(bmi,3.3)
0
5
10
15
20
20
25
30
35
bmi

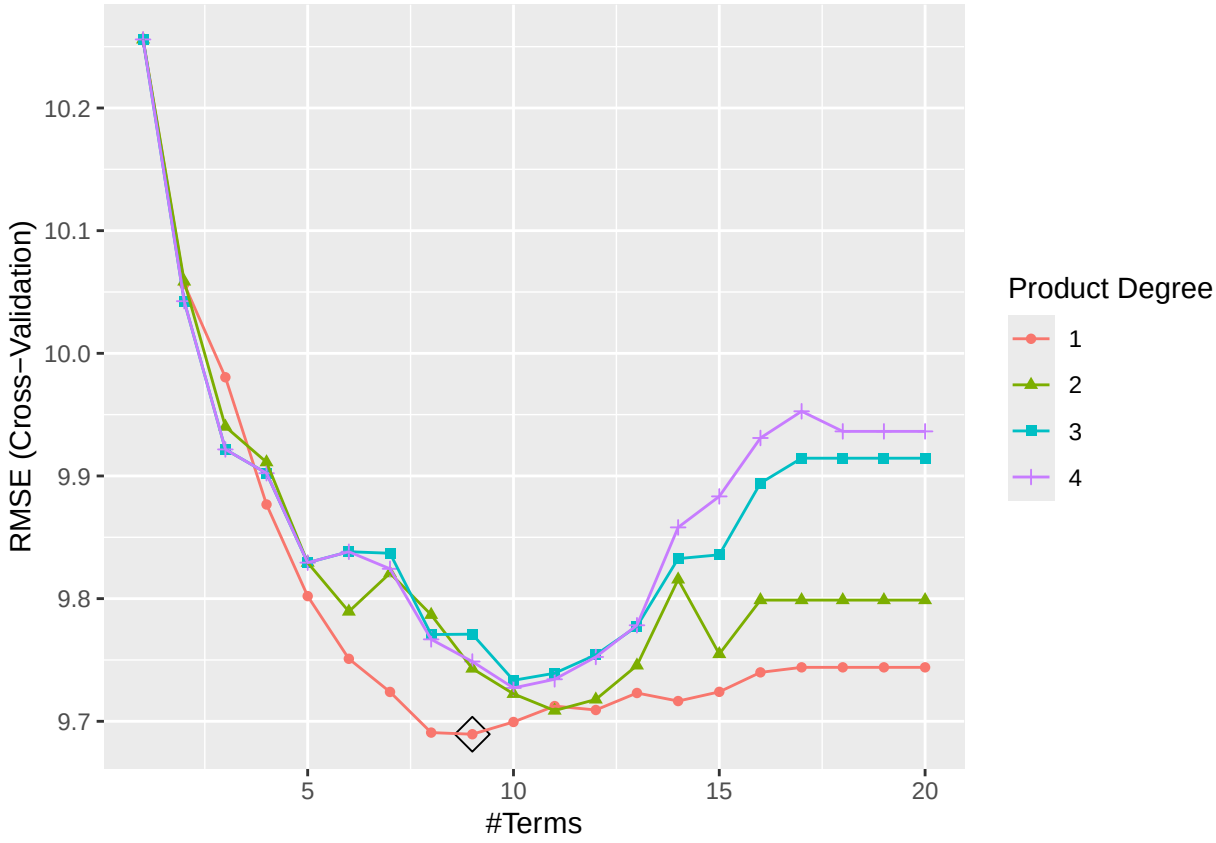




MARS

```
# set grid
mars_grid <- expand.grid(degree = 1:4, nprune = 1:20)

set.seed(2)

# fit a MARS model
mars.fit <- train(x, y,
                  method = "earth",
```

```
                  tuneGrid = mars_grid,
                  trControl = ctrl)
# plot
ggplot(mars.fit, highlight = TRUE)
```




```
# best tuning parameters
mars.fit$bestTune
```

```
##   nprune degree
## 9      9      1
```

```
# regression function
mars.fit$finalModel
```

```
## Selected 9 of 22 terms, and 6 of 17 predictors (nprune=9)
## Termination condition: Reached nk 35
## Importance: bmi, vaccine, hypertension, gender, severity, smoking2, ...
## Number of terms at each degree of interaction: 1 8 (additive model)
## GCV 93.70436    RSS 187609.4    GRSq 0.1103906    RSq 0.1243246
```

```
# report the regression function
summary(mars.fit)
```

```
## Call: earth(x=matrix[2036,17], y=c(45,33,41,50,4...), keepxy=TRUE, degree=1,
##             nprune=9)
##
##              coefficients
## (Intercept)     37.259963
## gender          -1.965793
```

```
## smoking2          2.192786
## hypertension      2.194544
## vaccine          -3.591286
## severity          2.311464
## h(bmi-25.7)       0.986802
## h(27.1-bmi)       1.575856
## h(bmi-29.7)       1.399244
##
## Selected 9 of 22 terms, and 6 of 17 predictors (nprune=9)
## Termination condition: Reached nk 35
## Importance: bmi, vaccine, hypertension, gender, severity, smoking2, ...
## Number of terms at each degree of interaction: 1 8 (additive model)
## GCV 93.70436    RSS 187609.4    GRSq 0.1103906    RSq 0.1243246
```

```
coef(mars.fit$finalModel)
```

```
##  (Intercept)  h(27.1-bmi)      vaccine hypertension       gender     severity
##   37.2599631    1.5758557   -3.5912857    2.1945438   -1.9657932    2.3114639
##     smoking2  h(bmi-29.7)  h(bmi-25.7)
##    2.1927858    1.3992436    0.9868021
```

```
# test error
pred.mars <- predict(mars.fit, newdata = testing_data)

test.error.mars <- mean((pred.mars - y2)^2)
```

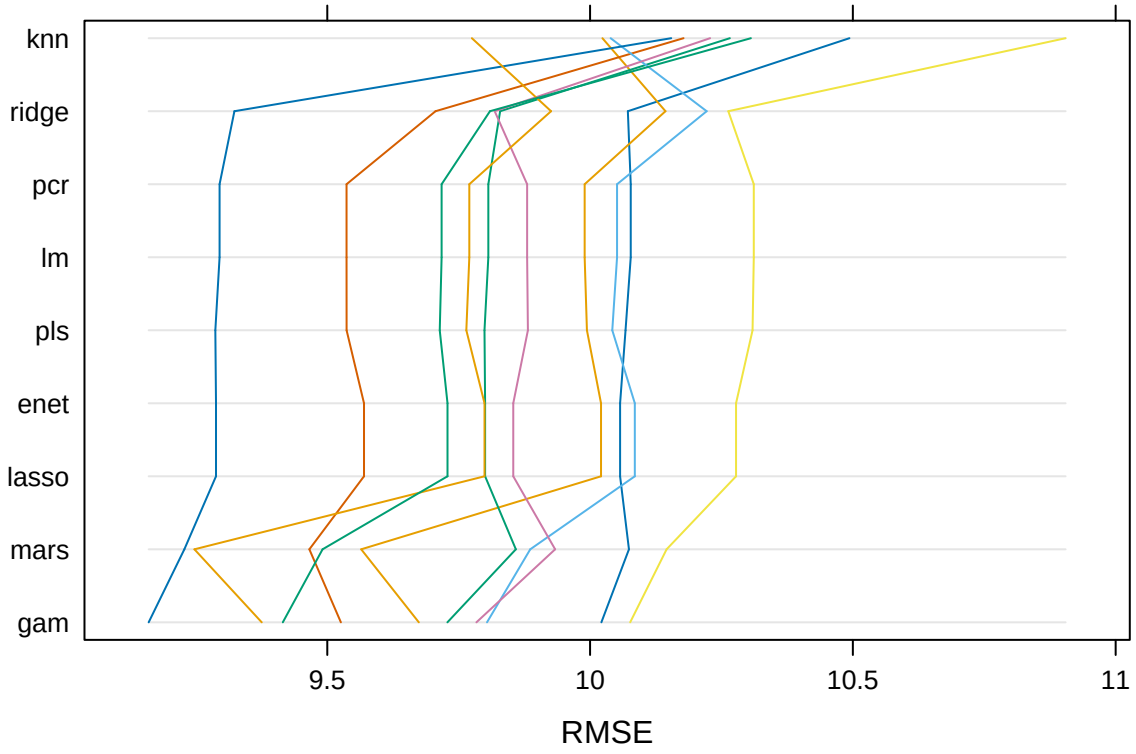
Model Comparison

```
# compare models
resamp <- resamples(list(knn = knn.fit, ridge = ridge.fit, lasso = lasso.fit, enet = enet.fit, pcr = pc

summary(resamp)
```

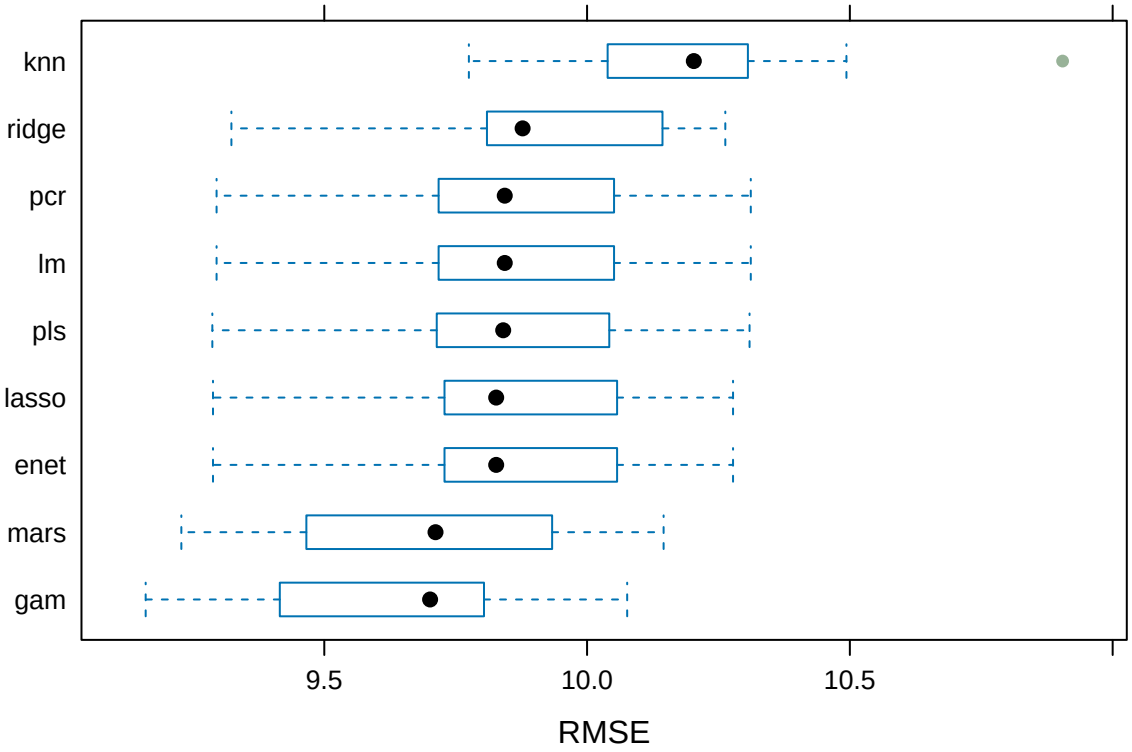
```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: knn, ridge, lasso, enet, pcr, pls, gam, mars, lm
## Number of resamples: 10
##
## MAE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## knn   8.078560 8.238547 8.338241 8.374408 8.423466 9.065122    0
## ridge 7.554839 7.796817 8.028171 8.036024 8.227722 8.534544    0
## lasso 7.490731 7.738740 7.959282 7.976912 8.157811 8.521826    0
## enet  7.490731 7.738740 7.959282 7.976912 8.157811 8.521826    0
## pcr   7.471190 7.732391 7.945312 7.963437 8.138785 8.553821    0
## pls   7.463340 7.729869 7.936804 7.958825 8.137538 8.551126    0
## gam   7.396835 7.714321 7.870555 7.836253 7.981149 8.294511    0
## mars  7.434811 7.654459 7.956438 7.891283 8.044486 8.360594    0
## lm    7.471190 7.732391 7.945312 7.963437 8.138785 8.553821    0
##
## RMSE
##           Min.   1st Qu.    Median      Mean   3rd Qu.     Max. NA's
```

```
## knn   9.775082 10.067897 10.203083 10.236835 10.295972 10.90479    0
## ridge 9.323243  9.811758  9.877341  9.911201 10.125561 10.26300    0
## lasso 9.288226  9.746326  9.827052  9.848079 10.048023 10.27773    0
## enet  9.288226  9.746326  9.827052  9.848079 10.048023 10.27773    0
## pcr   9.295032  9.730692  9.843324  9.843580 10.035902 10.31146    0
## pls   9.286975  9.726609  9.840381  9.839578 10.030158 10.30914    0
## gam   9.160109  9.442958  9.701322  9.656456  9.798819 10.07632    0
## mars  9.228017  9.472152  9.711634  9.689456  9.921687 10.14559    0
## lm    9.295032  9.730692  9.843324  9.843580 10.035902 10.31146    0
##
## Rsquared
##               Min.      1st Qu.      Median        Mean    3rd Qu.       Max. NA's
## knn   0.000876636 0.009157928 0.01945372 0.02633272 0.03838402 0.0656237    0
## ridge 0.032590322 0.047320928 0.07725093 0.06959520 0.08758812 0.1023314    0
## lasso 0.050363974 0.063304846 0.08049866 0.08045844 0.09441546 0.1089420    0
## enet  0.050363974 0.063304846 0.08049866 0.08045844 0.09441546 0.1089420    0
## pcr   0.056315228 0.066906115 0.07619545 0.08163822 0.09510753 0.1123579    0
## pls   0.055737745 0.067973952 0.07622323 0.08231627 0.09592297 0.1123246    0
## gam   0.084929267 0.105907026 0.11054868 0.11621277 0.12693428 0.1584404    0
## mars  0.077602135 0.088553841 0.10969842 0.11135266 0.12754035 0.1626978    0
## lm    0.056315228 0.066906115 0.07619545 0.08163822 0.09510753 0.1123579    0
```

```
parallelplot(resamp, metric = "RMSE")
```




```
bwplot(resamp, metric = "RMSE")
```



MARS has lowest mean and median RMSE -> model I pick