# Data Science II Midterm Project Analysis

Camille Okonkwo

## Contents

```r
library(tidymodels)
library(splines)
library(caret)
library(glmnet)
library(table1)
library(kableExtra)
library(summarytools)
library(corrplot)
library(cowplot)
```

# 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: a matrix of predictors and a vector of response.

```
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

## Descriptive Statistics Table

```
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%)\         IIIIIIIIIIIII \          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%)\         IIIIIIIIIIII \           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%)\     IIIIIIIIIIIIIII \        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   \ \ \ \ \ \ \ \ \ \ :\   3000
##      [numeric]        min < med < max:\                               \ \ \ \ \ \ \ \ : : .\   (100
##                       28 < 110 < 178\                                 \ \ \ \ \ \ \ \ : : :\
##                       IQR (CV) : 27 (0.2)                             \ \ \ \ \ \ . : : : .\
##                                                                       \ \ \ \ . : : : : : : .
##
## 12   vaccine\         Min  : 0\                 0 : 1212 (40.4%)\     IIIIIIII \               3000
##      [integer]        Mean : 0.6\               1 : 1788 (59.6%)      IIIIIIIIIII              (100
##                       Max  : 1
##
## 13   severity\        Min  : 0\                 0 : 2679 (89.3%)\     IIIIIIIIIIIIIIIII \      3000
##      [integer]        Mean : 0.1\               1 :  321 (10.7%)      II                       (100
##                       Max  : 1
##
## 14   study\           1\. A\                    2000 (66.7%)\         IIIIIIIIIIII \           3000
##      [character]      2\. B                     1000 (33.3%)          IIIIII                   (100
##
## 15   recovery_time\   Mean (sd) : 42.2 (23.2)\ 140 distinct values   : :\                     3000
##      [numeric]        min < med < max:\                               : :\                     (100
##                       2 < 39 < 365\                                   : :\
##                       IQR (CV) : 18 (0.5)                             : :\
##                                                                       : : .
## ----------------------------------------------------------------------------------------------
```

```r
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 + `Bo
                            data = dat_ds,
                            overall = "Total",
                            caption = "Descriptive Statistics")

ds = t1kable(descriptive_table)
```

```
ds
```

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.

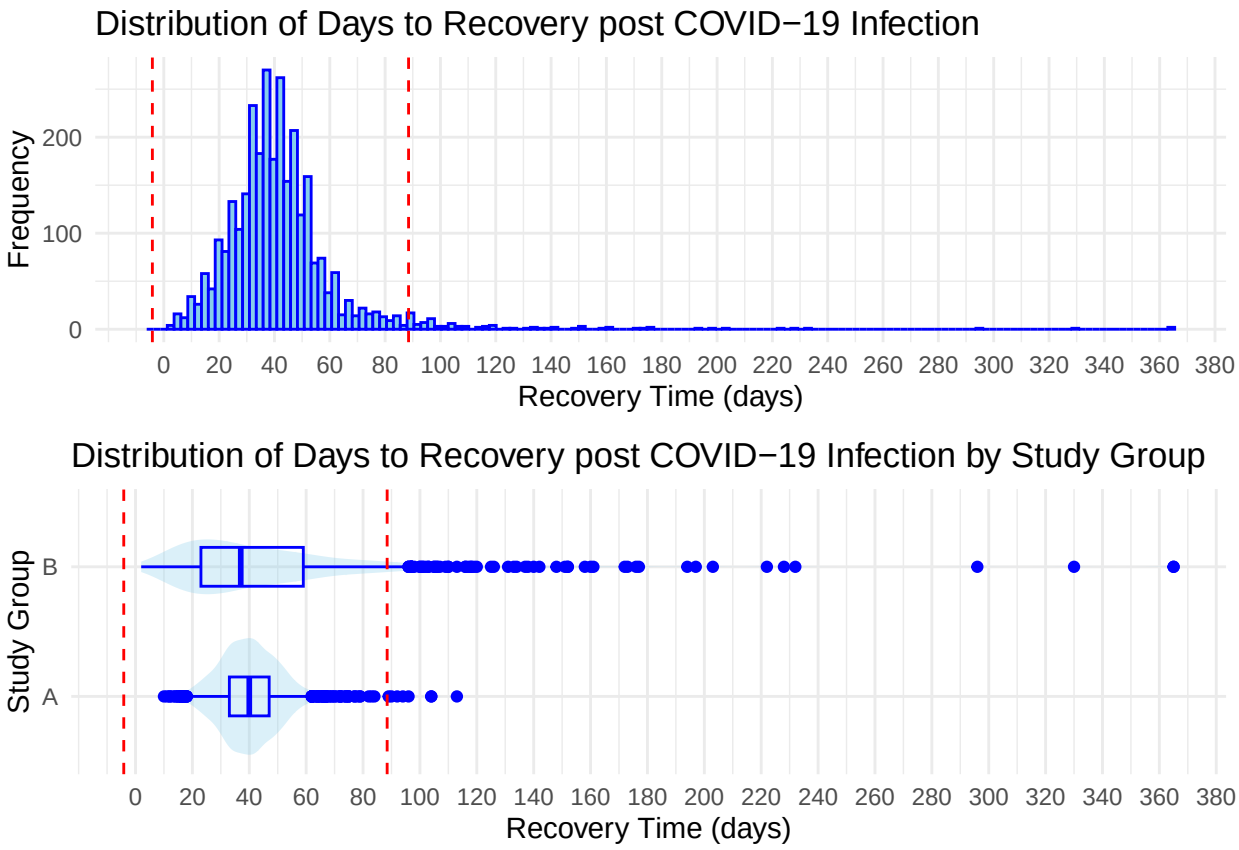
## Response Variable Exploration

```
# Calculate mean and standard deviation
mean_value = mean(dat$recovery_time)
sd_value = sd(dat$recovery_time)

# Define upper and lower bounds
outlier_coeff = 2
outlier_high = mean_value + outlier_coeff * sd_value
outlier_low = mean_value - outlier_coeff * sd_value

recovery_outlier =
  dat |>
  filter(recovery_time >= outlier_low & recovery_time <= outlier_high)

# recovery_time boxplot
boxplot_recovery =
  dat |>
  ggplot(aes(x = recovery_time, y = study)) +
  geom_violin(fill = "skyblue", alpha = 0.3, color= NA) +
  geom_boxplot(fill = NA, color = "blue",
               width = 0.3, coef = outlier_coeff/2) +
  geom_vline(xintercept = c(outlier_low, outlier_high),
             color = "red",linetype = "dashed", size = .5) +
  labs(title = "Distribution of Days to Recovery post COVID-19 Infection by Study Group",
       x = "Recovery Time (days)", y = "Study Group") +
  theme_minimal() +
  scale_x_continuous(
    breaks = seq(0, 400, by = 20),
    labels = seq(0, 400, by = 20)
  )

# recovery_time histogram
histogram_recovery =
  dat |>
  ggplot(aes(x = recovery_time)) +
  geom_histogram(bins = 150, fill = "skyblue", color = "blue") +
  geom_vline(xintercept = c(outlier_low, outlier_high),
             color = "red", linetype = "dashed", size = .5) +
  labs(title = "Distribution of Days to Recovery post COVID-19 Infection",
       x = "Recovery Time (days)", y = "Frequency") +
  theme_minimal() +
  scale_x_continuous(
    breaks = seq(0, 400, by = 20),
    labels = seq(0, 400, by = 20)
  )
```

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] |

```
combined_recovery =
  plot_grid(histogram_recovery, boxplot_recovery, ncol = 1)

# Show the combined plot
print(combined_recovery)
```



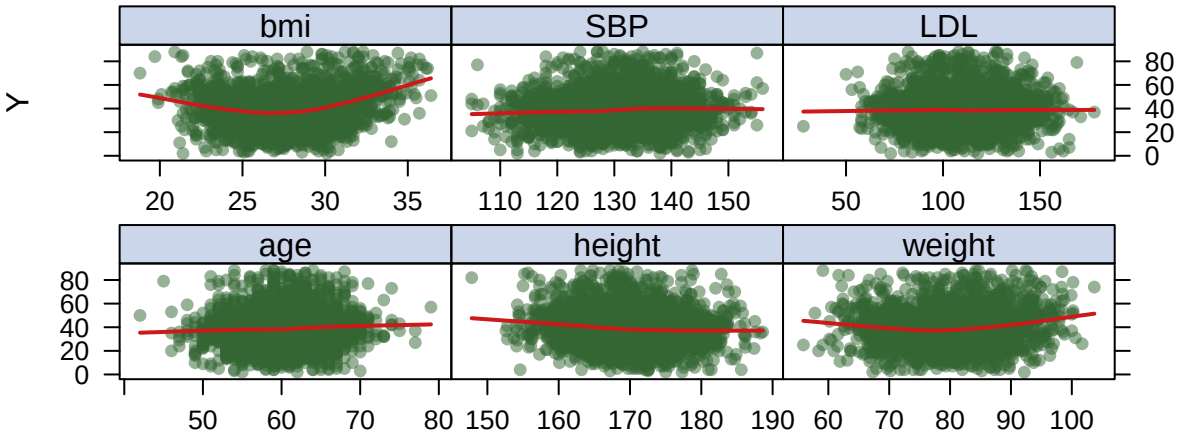

Using a cut-off based on the standard deviation ± 2 times the mean, there are a total 92 outliers (approximately 3% of the observations). These outliers will be excluded from future analysis. Specifically, among the outliers, 84 belong to the study group B population.

```
# removing recovery_time outliers
dat2 =
  dat |>
  filter(recovery_time >= outlier_low & recovery_time <= outlier_high)

x.dat2 = model.matrix(recovery_time ~., dat2)[, -1]
y.dat2 = dat2$recovery_time
```

## Feature Plot

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

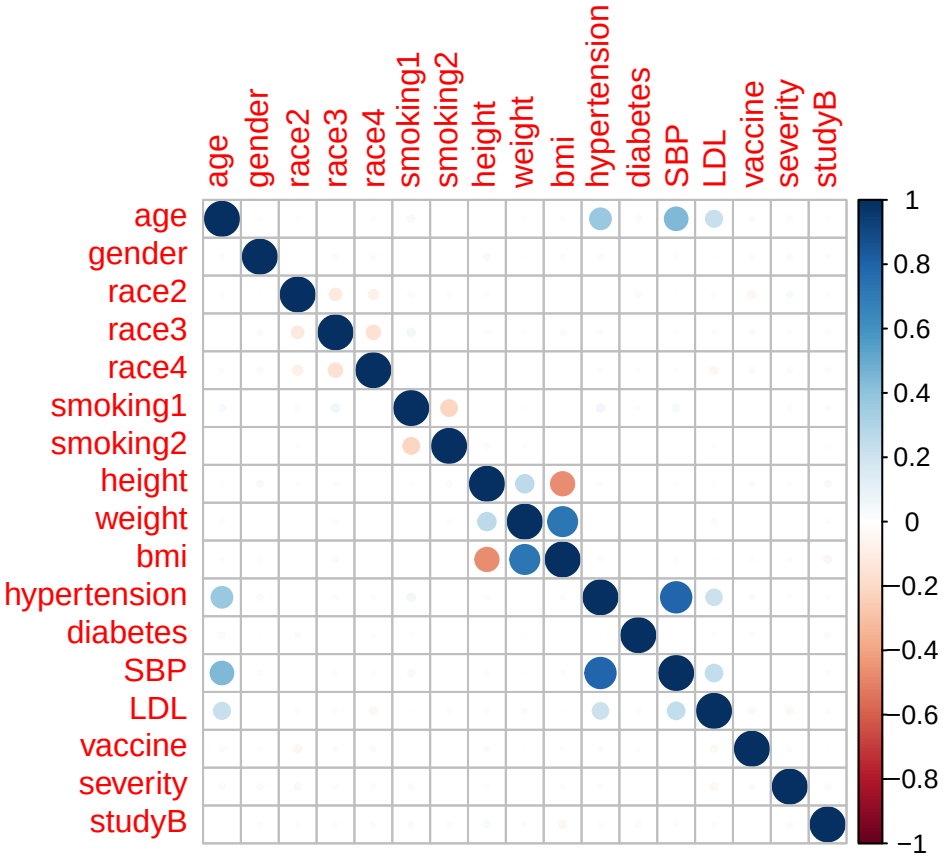
```
trellis.par.set(theme1)

featurePlot(
  x.dat2[, -c(2, 3, 4, 5, 6, 7, 11, 12, 15, 16, 17) ],
  y.dat2,
  plot = "scatter",
  labels = c("", "Y"),
  type = c("p", "smooth"),
  layout = c(3, 3))
```




From the feature plot of the continuous variables, there appears to be no strong linear correlations with our response variable `recovery_time`. `bmi` and `weight` however show a potential non-linear relationship. A GAM or MARS model may be best (**Do we want to do a transformation?**)

## Correlation Matrix

```
corrplot(cor(x.dat2), method = "circle", type = "full")
```




The correlation matrix between predictors indicates multicollinearity between `bmi` and `weight`, `sbp` and `hypertension`, and potentially `bmi` and `height`.

# Model Training in caret

## 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
head(x)
```

```
##   age gender race2 race3 race4 smoking1 smoking2 height weight  bmi
## 1  61      0     0     0     0        0        1  164.5   78.9 29.1
## 2  52      1     0     0     0        1        0  176.0   76.0 24.5
## 3  60      1     0     1     0        1        0  167.4   72.6 25.9
## 4  58      0     0     0     0        0        0  172.6   96.2 32.3
## 5  56      0     0     0     0        1        0  177.9   85.0 26.9
## 6  56      0     0     0     1        0        1  162.4   81.6 30.9
##   hypertension diabetes SBP LDL vaccine severity studyB
## 1            1        0 136  97       1        0      0
## 2            0        0 115 107       0        1      0
## 3            1        0 145 145       1        0      1
## 4            0        0 123  98       0        0      0
## 5            1        0 131  83       0        0      0
## 6            0        0 114 111       0        1      0
```

```
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

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

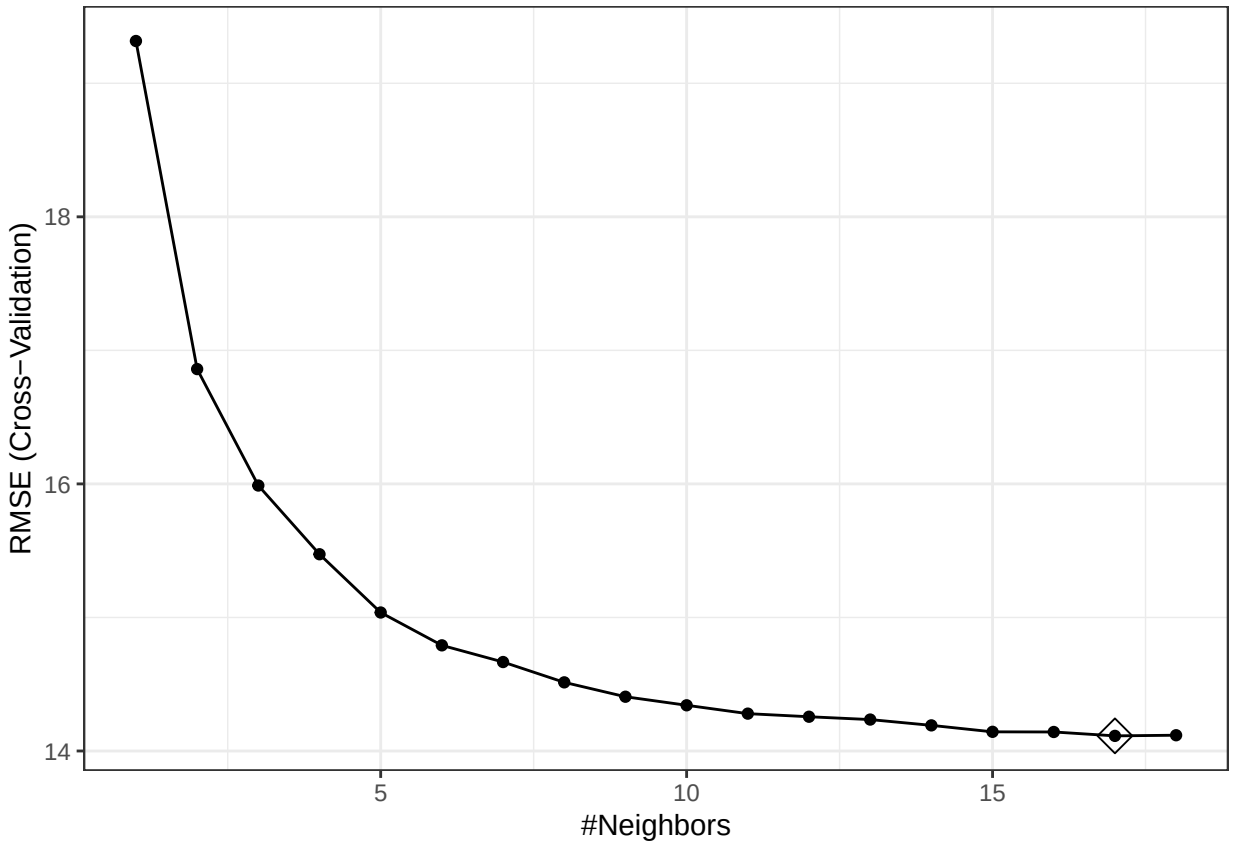
## Linear Model

```
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
## -43.252  -8.769  -0.081   7.770  57.163
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -921.70615   88.95873 -10.361  < 2e-16 ***
## age             0.09001    0.07143   1.260 0.207714
## gender         -2.34238    0.56854  -4.120 3.92e-05 ***
## race2           0.15622    1.26267   0.124 0.901544
## race3          -0.67548    0.72746  -0.929 0.353221
## race4          -1.60376    1.02078  -1.571 0.116294
## smoking1        2.43690    0.64370   3.786 0.000157 ***
## smoking2        2.74070    0.94085   2.913 0.003614 **
## height          5.54049    0.52181  10.618  < 2e-16 ***
## weight         -5.98870    0.55353 -10.819  < 2e-16 ***
## bmi            18.06359    1.59283  11.341  < 2e-16 ***
## hypertension    2.71329    0.94396   2.874 0.004086 **
## diabetes       -0.80078    0.78196  -1.024 0.305908
## SBP            -0.02290    0.06145  -0.373 0.709407
## LDL            -0.02633    0.01503  -1.752 0.079845 .
## vaccine        -4.61566    0.58134  -7.940 3.14e-15 ***
## severity        3.26083    0.94284   3.459 0.000553 ***
## studyB         -1.42762    0.61463  -2.323 0.020280 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.67 on 2308 degrees of freedom
## Multiple R-squared:  0.1303, Adjusted R-squared:  0.1238
## F-statistic: 20.33 on 17 and 2308 DF,  p-value: < 2.2e-16
```

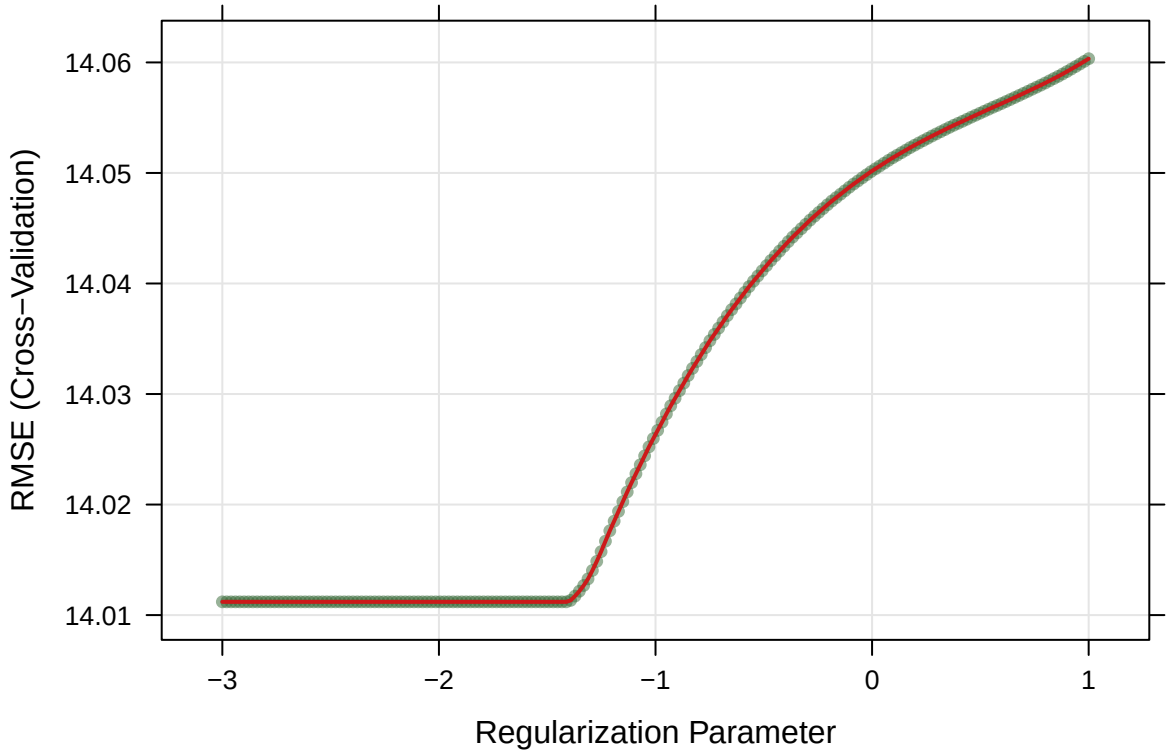
## KNN

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

knn.fit <- train(x, y,
                 method = "knn",
                 trControl = ctrl,
                 tuneGrid = expand.grid(k = seq(from = 1, to = 18, 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
## 80     0 0.2436408
```

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

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)  -41.110841702
## age            0.085355232
## gender        -2.106552923
## race2          0.350339165
## race3         -0.662534557
## race4         -1.849407841
## smoking1       2.319391839
## smoking2       2.721960592
## height         0.346873433
## weight        -0.469146660
## bmi            2.166457412
## hypertension   2.459339794
## diabetes      -0.778031077
## SBP           -0.006287343
## LDL           -0.027702596
## vaccine       -4.482501524
## severity       3.086484155
## studyB        -1.616103346
```

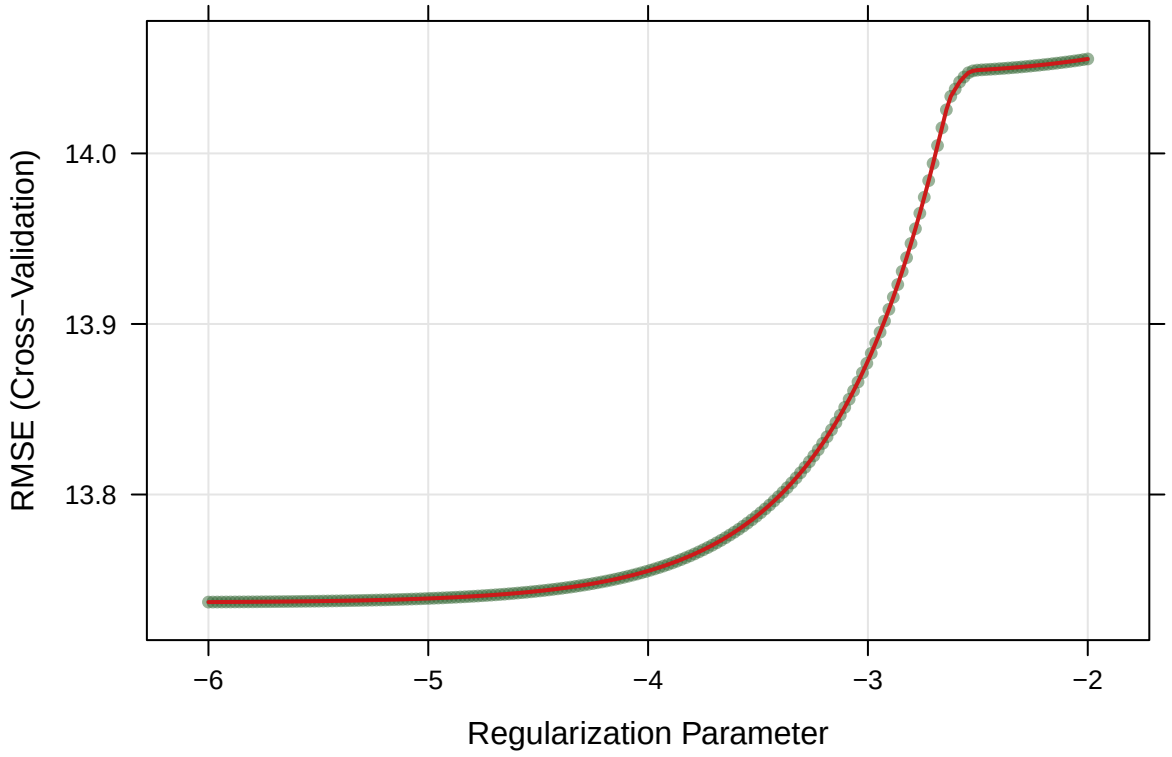
```
ridge.pred <- predict(ridge.fit, newdata = model.matrix(recovery_time ~ ., testing_data)[,-1])
```

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

```
## [1] 173.9842
```

### Lasso

```
set.seed(2)

# lasso using caret
lasso.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(-2, -6, length=200))),
                   trControl = ctrl)

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




```
lasso.fit$bestTune
```

```
##   alpha      lambda
## 1     1 0.002478752
```

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

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)  -869.53072339
## age             0.08885791
## gender         -2.32570530
```

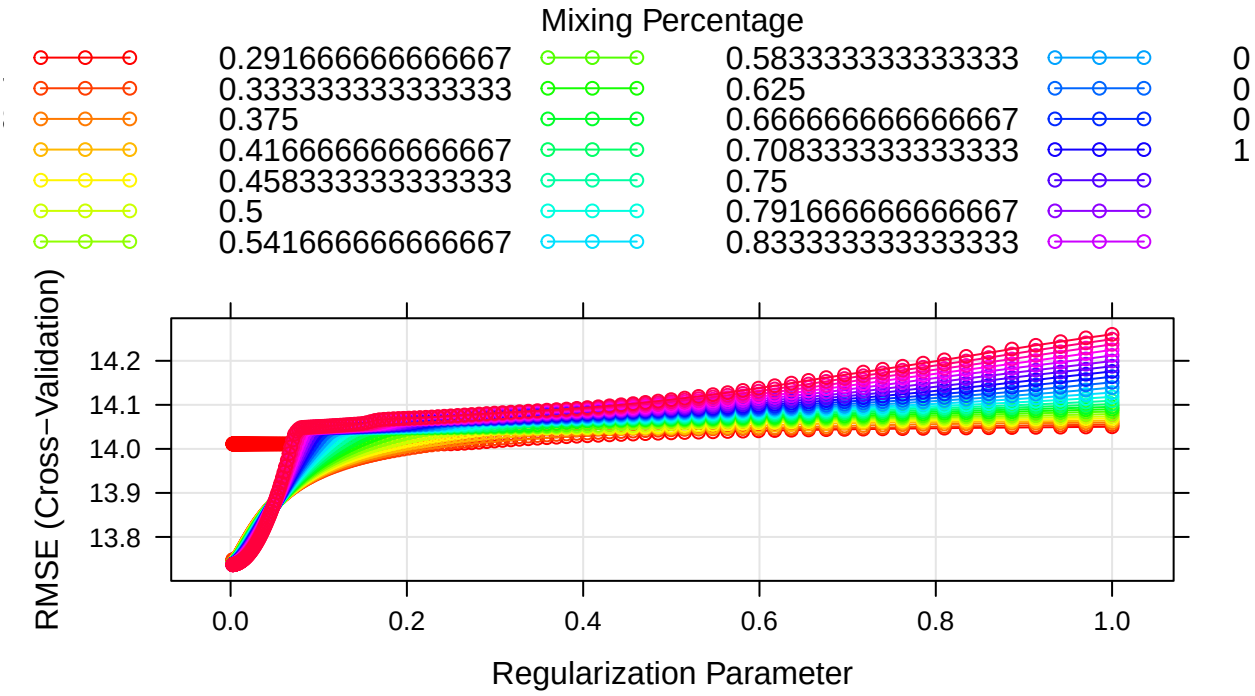
```
## race2           0.15926040
## race3          -0.66871775
## race4          -1.60977732
## smoking1        2.42619513
## smoking2        2.73387753
## height          5.23182712
## weight         -5.66083508
## bmi            17.12056779
## hypertension    2.68178958
## diabetes       -0.79365994
## SBP            -0.02062032
## LDL            -0.02630786
## vaccine        -4.60689656
## severity        3.24636259
## studyB         -1.43493314
```

## Elastic Net

```
set.seed(2)

# elastic net using caret
enet.fit <- train(x, y,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = seq(0, 1, length =
                                                       25),
                                         lambda = exp(seq(0, -6, length=200))),
                  trControl = ctrl)

enet.fit$bestTune
```

```
##      alpha      lambda
## 4801     1 0.002478752
```

```
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)  -869.53072339
## age             0.08885791
## gender         -2.32570530
## race2           0.15926040
## race3          -0.66871775
## race4          -1.60977732
## smoking1        2.42619513
## smoking2        2.73387753
## height          5.23182712
## weight         -5.66083508
## bmi            17.12056779
## hypertension    2.68178958
## diabetes       -0.79365994
## SBP            -0.02062032
## LDL            -0.02630786
## vaccine        -4.60689656
## severity        3.24636259
## studyB         -1.43493314
```

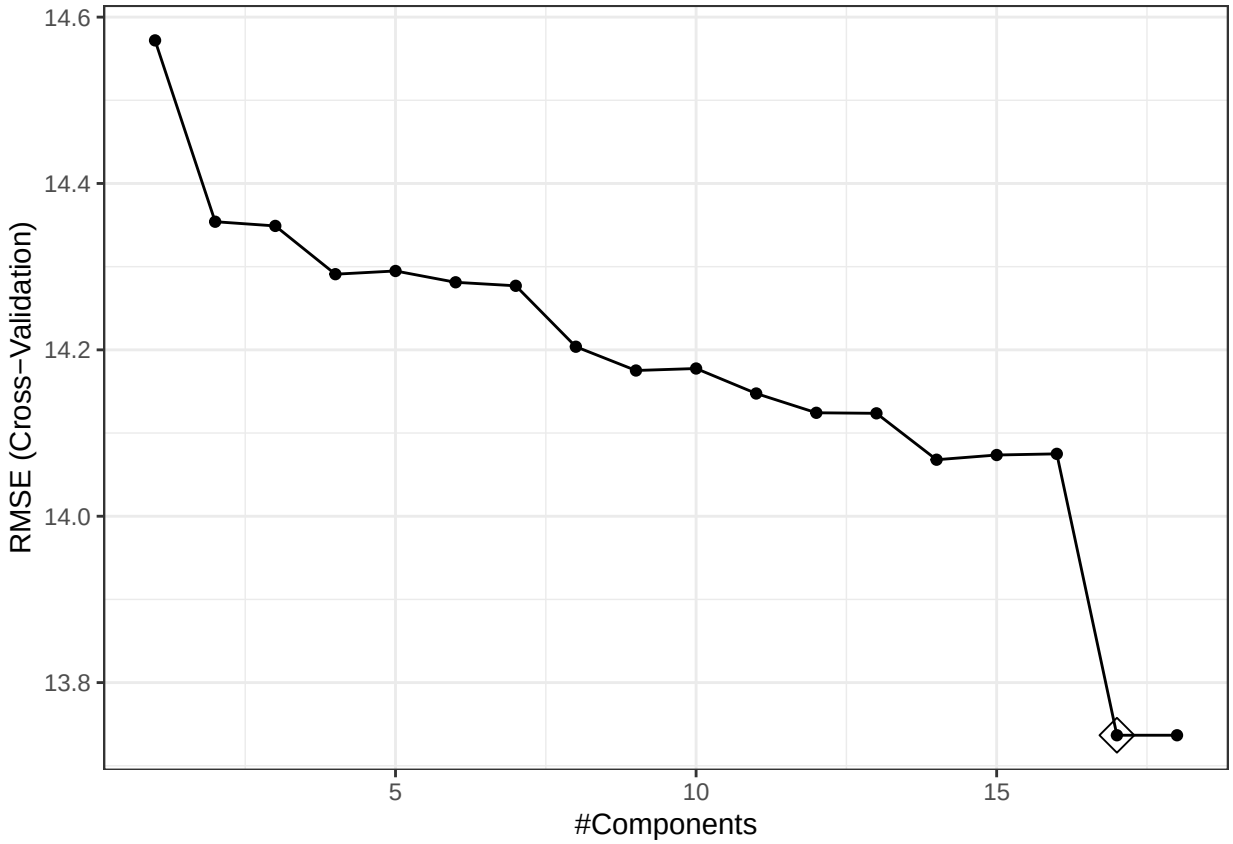
## PCR

```
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] 177.0229
```

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

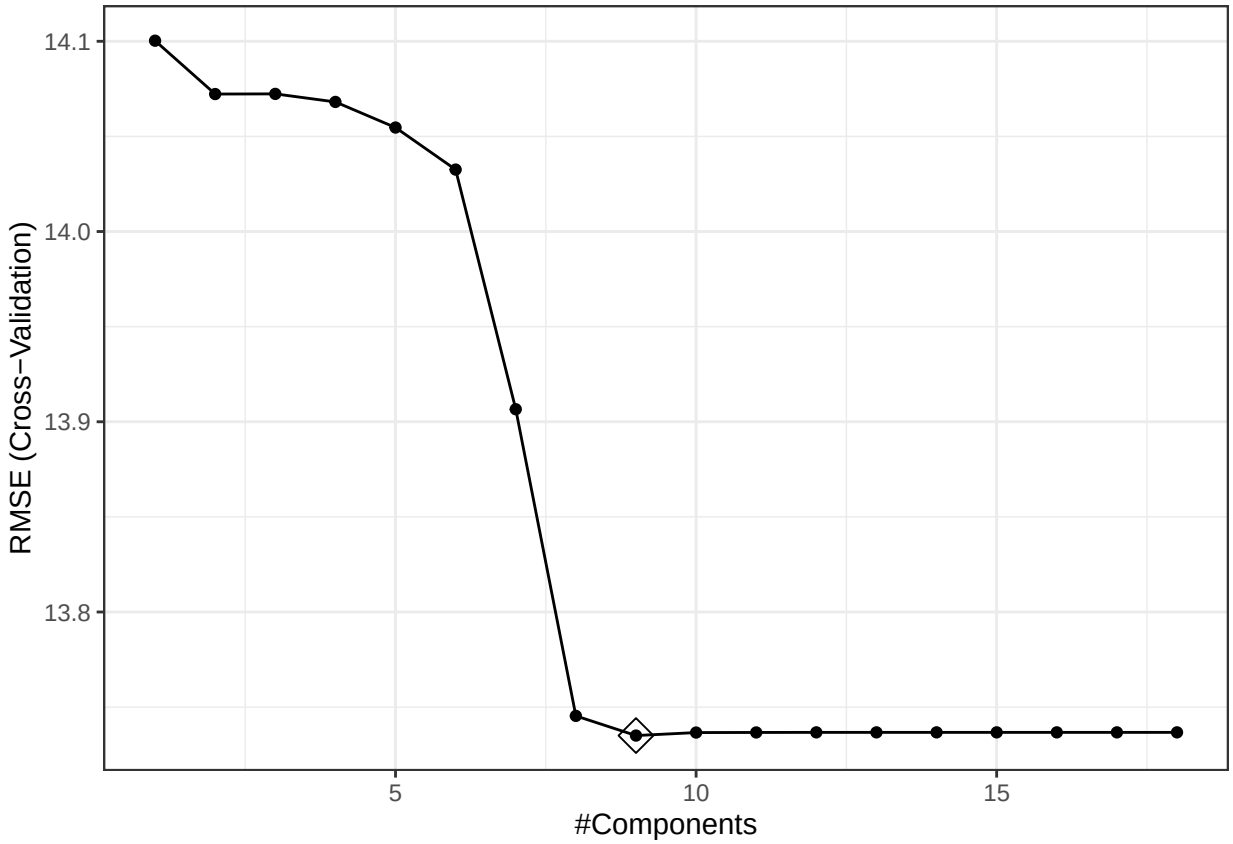



## PLS

```
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] 177.0472
```

```
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
```

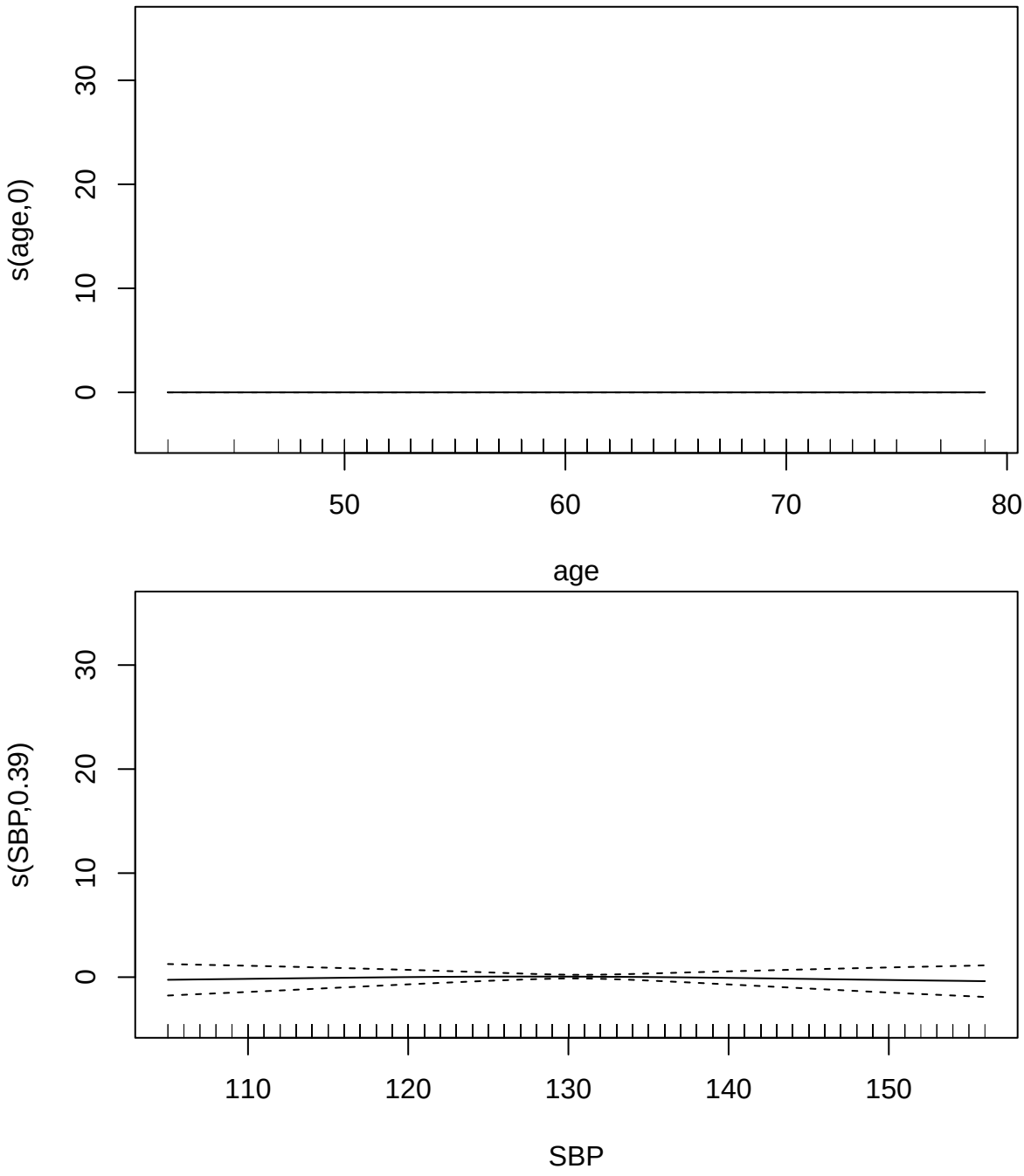
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender + race2 + race3 + race4 + smoking1 + smoking2 +
##     hypertension + diabetes + vaccine + severity + studyB + s(age) +
##     s(SBP) + s(LDL) + s(bmi) + s(height) + s(weight)
```

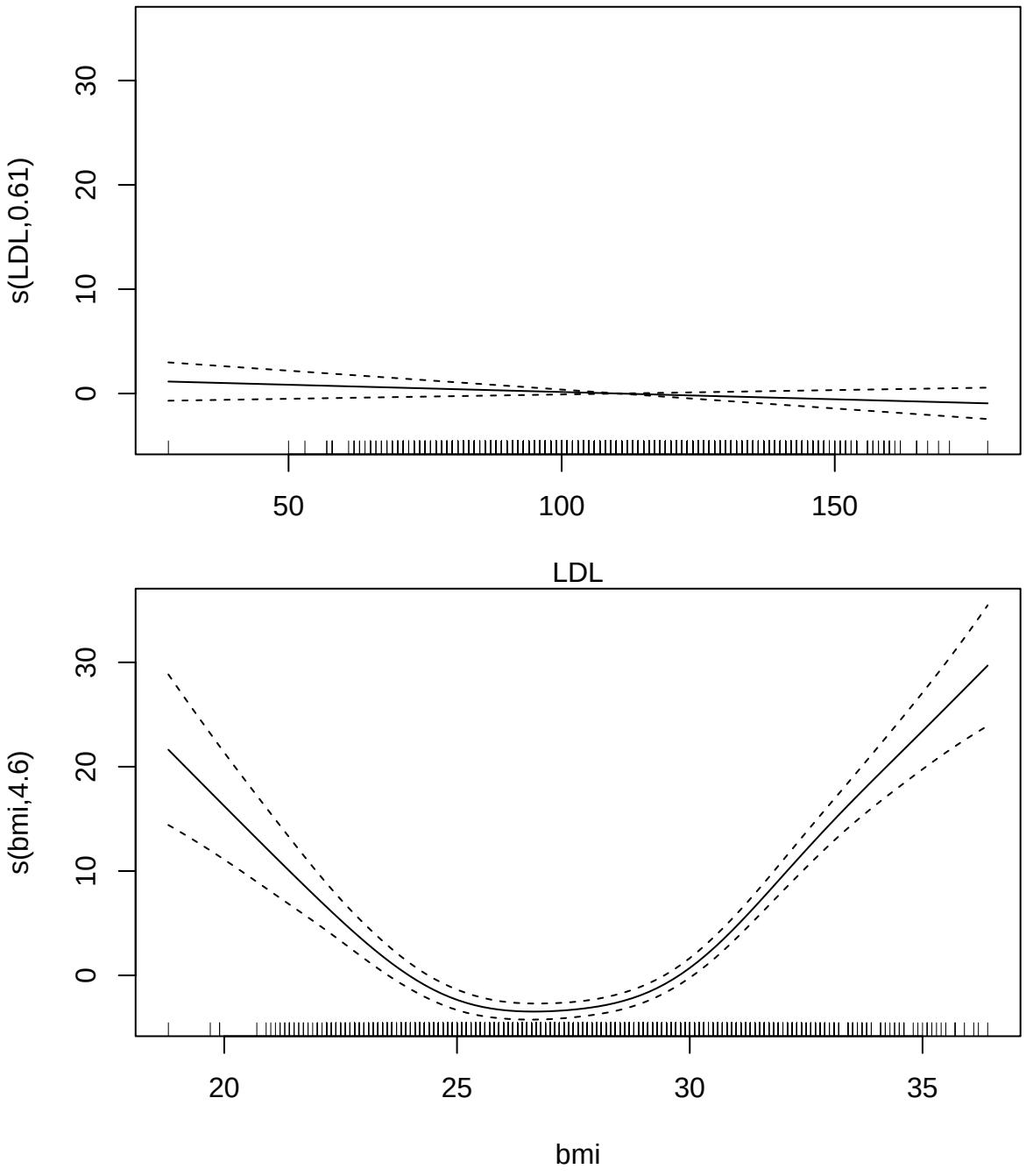
```
##
## Estimated degrees of freedom:
## 0.000 0.391 0.606 4.601 0.000 0.000  total = 17.6
##
## GCV score: 176.5636
```

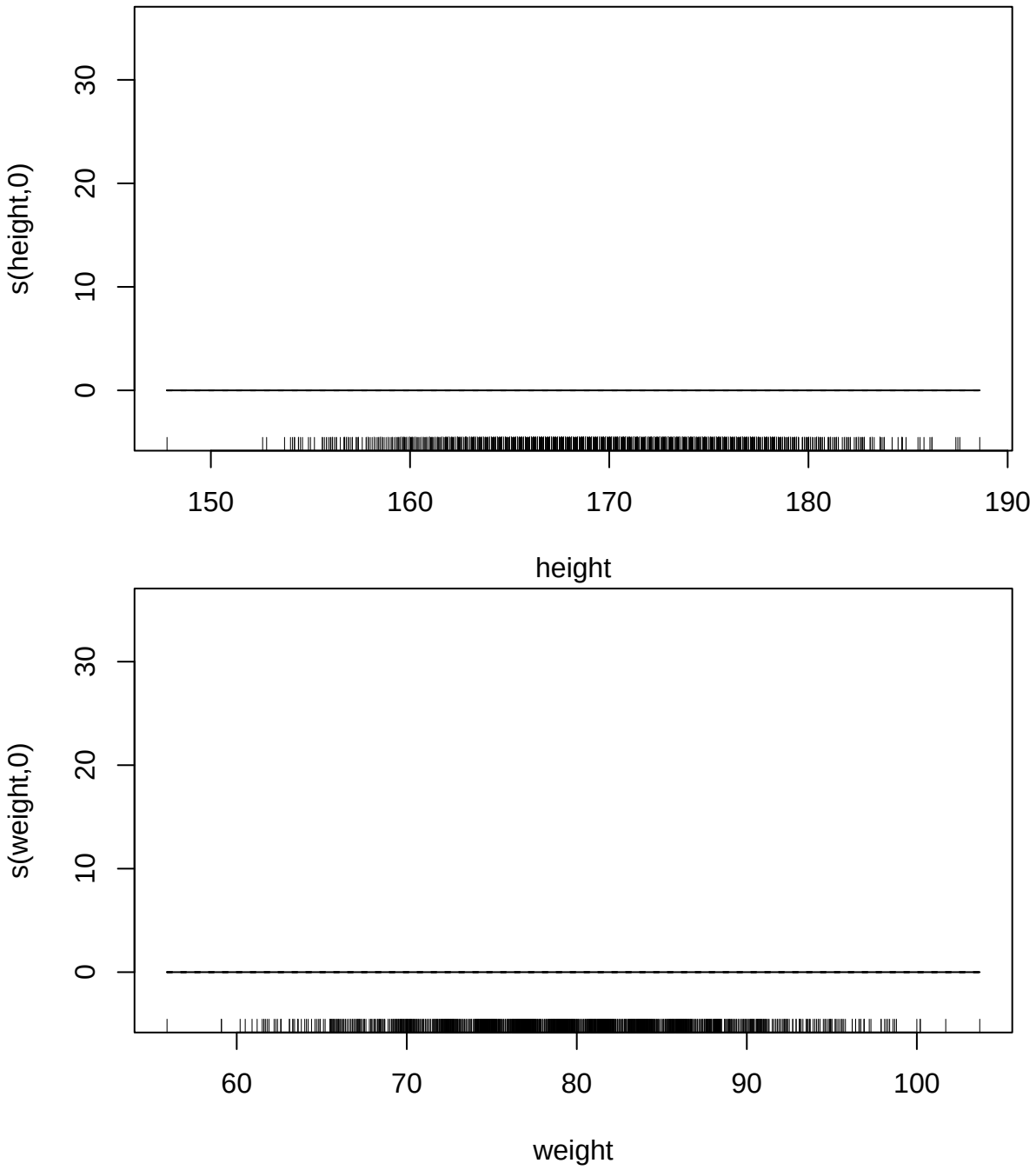
```
coef(gam.fit$finalModel)
```

```
##   (Intercept)        gender         race2         race3         race4
##  4.156512e+01 -2.424904e+00 -2.069929e-01 -4.087208e-01 -1.517799e+00
##      smoking1      smoking2  hypertension      diabetes       vaccine
##  2.532428e+00  3.125378e+00  2.725841e+00 -8.823480e-01 -4.645970e+00
##      severity        studyB      s(age).1      s(age).2      s(age).3
##  3.430045e+00 -1.492737e+00  1.466527e-09 -4.296600e-11 -2.532287e-10
##      s(age).4      s(age).5      s(age).6      s(age).7      s(age).8
## -2.448336e-10  1.968084e-10 -1.819328e-10 -1.800354e-10 -5.896549e-10
##      s(age).9      s(SBP).1      s(SBP).2      s(SBP).3      s(SBP).4
##  4.452399e-11 -4.399891e-02 -1.277177e-02  2.126971e-02 -3.427318e-02
##      s(SBP).5      s(SBP).6      s(SBP).7      s(SBP).8      s(SBP).9
## -1.912220e-02 -3.680040e-02  1.949880e-02  1.926686e-01 -3.918919e-11
##      s(LDL).1      s(LDL).2      s(LDL).3      s(LDL).4      s(LDL).5
##  7.283651e-09  5.561433e-10  2.780708e-09 -2.048260e-09  1.929882e-09
##      s(LDL).6      s(LDL).7      s(LDL).8      s(LDL).9      s(bmi).1
##  1.692079e-09  1.731161e-09  1.540469e-08 -2.751085e-01 -4.900514e+00
##      s(bmi).2      s(bmi).3      s(bmi).4      s(bmi).5      s(bmi).6
##  1.161375e+00  3.643567e+00  3.081315e+00 -2.063386e+00 -3.638837e+00
##      s(bmi).7      s(bmi).8      s(bmi).9   s(height).1   s(height).2
##  2.542167e+00 -1.931156e+01 -1.448275e-10  9.166223e-10  8.452646e-11
##   s(height).3   s(height).4   s(height).5   s(height).6   s(height).7
## -6.468484e-11 -1.030657e-10 -1.844148e-11  6.679943e-11  2.627186e-11
##   s(height).8   s(height).9   s(weight).1   s(weight).2   s(weight).3
##  7.713828e-10 -1.409085e-10  7.547872e-10  2.438624e-11  1.237405e-10
##   s(weight).4   s(weight).5   s(weight).6   s(weight).7   s(weight).8
##  6.971581e-11  3.993648e-11 -4.840477e-11 -1.118272e-11  5.631146e-10
##   s(weight).9
## -2.201982e-09
```

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


s(age,0)
0
10
20
30
50
60
70
80
age
s(SBP,0.39)
0
10
20
30
110
120
130
140
150
SBP

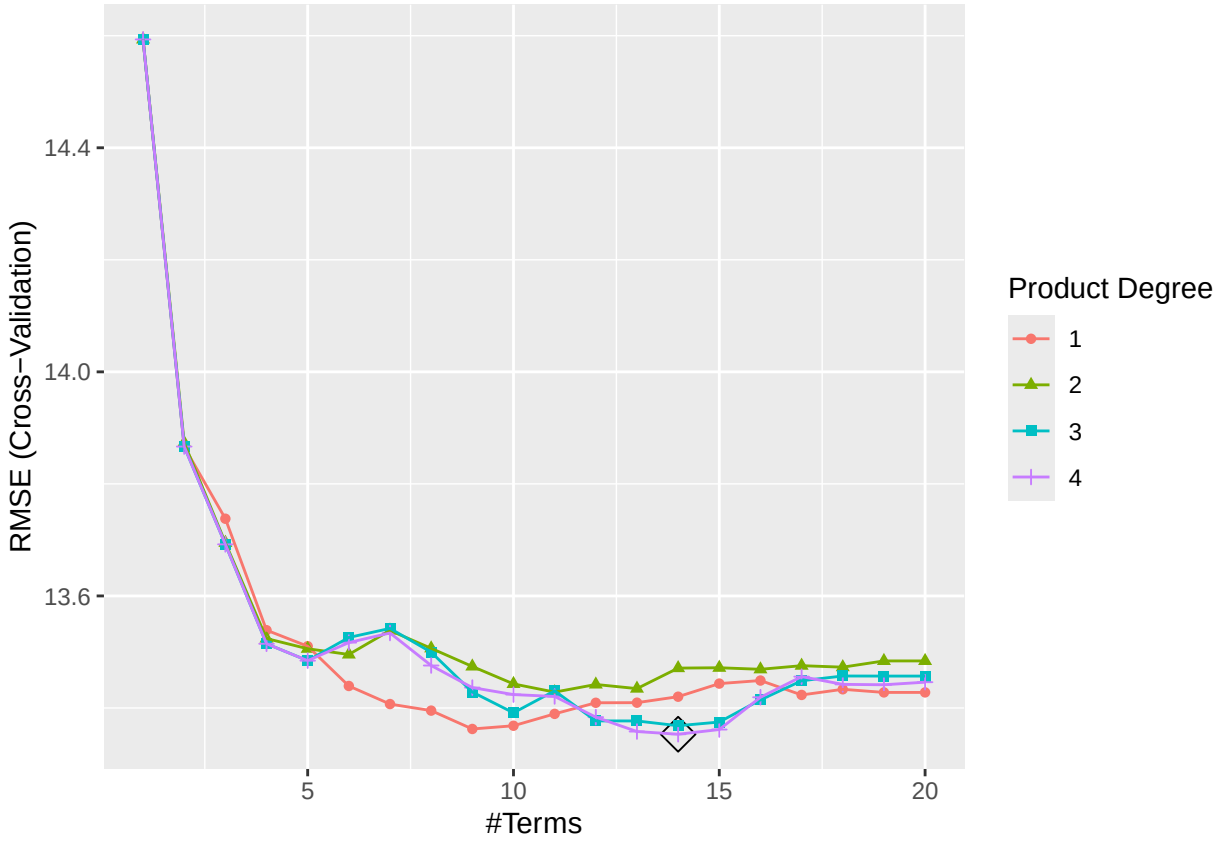

s(LDL,0.61)
0
10
20
30
50
100
150
LDL
s(bmi,4.6)
0
10
20
30
20
25
30
35
bmi




## 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
## 74     14      4
```

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

```
## Selected 14 of 25 terms, and 10 of 17 predictors (nprune=14)
## Termination condition: Reached nk 35
## Importance: bmi, vaccine, SBP, studyB, hypertension, gender, severity, ...
## Number of terms at each degree of interaction: 1 5 4 4
## GCV 173.7019    RSS 392476.4    GRSq 0.1861502    RSq 0.208744
```

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

```
## Call: earth(x=matrix[2326,17], y=c(29,33,14,27,4...), keepxy=TRUE, degree=4,
##             nprune=14)
##
##                                    coefficients
## (Intercept)                          22.2427492
## smoking1                              2.4265710
```

```
## hypertension                              2.7688156
## vaccine                                  -4.5176008
## h(bmi-24)                                 3.4575855
## h(28.4-bmi)                               3.9794908
## smoking2 * vaccine                        4.2232082
## vaccine * severity                        4.7329238
## vaccine * studyB                         -4.0001897
## gender * h(SBP-113)                      -0.1401477
## h(25.3-bmi) * vaccine * studyB            2.5964056
## h(height-170) * h(bmi-28.4) * studyB     -0.6938666
## h(bmi-28.4) * h(SBP-118) * studyB         0.1736301
## h(bmi-28.4) * h(118-SBP) * studyB        -2.4613936
##
## Selected 14 of 25 terms, and 10 of 17 predictors (nprune=14)
## Termination condition: Reached nk 35
## Importance: bmi, vaccine, SBP, studyB, hypertension, gender, severity, ...
## Number of terms at each degree of interaction: 1 5 4 4
## GCV 173.7019    RSS 392476.4    GRSq 0.1861502    RSq 0.208744
```

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

```
##                           (Intercept)                           h(28.4-bmi)
##                            22.2427492                             3.9794908
##                               vaccine                             h(bmi-24)
##                            -4.5176008                             3.4575855
##                          hypertension                   vaccine * severity
##                             2.7688156                             4.7329238
##                       vaccine * studyB                              smoking1
##                            -4.0001897                             2.4265710
##                     smoking2 * vaccine                   gender * h(SBP-113)
##                             4.2232082                            -0.1401477
##    h(bmi-28.4) * h(SBP-118) * studyB    h(bmi-28.4) * h(118-SBP) * studyB
##                             0.1736301                            -2.4613936
##       h(25.3-bmi) * vaccine * studyB h(height-170) * h(bmi-28.4) * studyB
##                             2.5964056                            -0.6938666
```

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

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

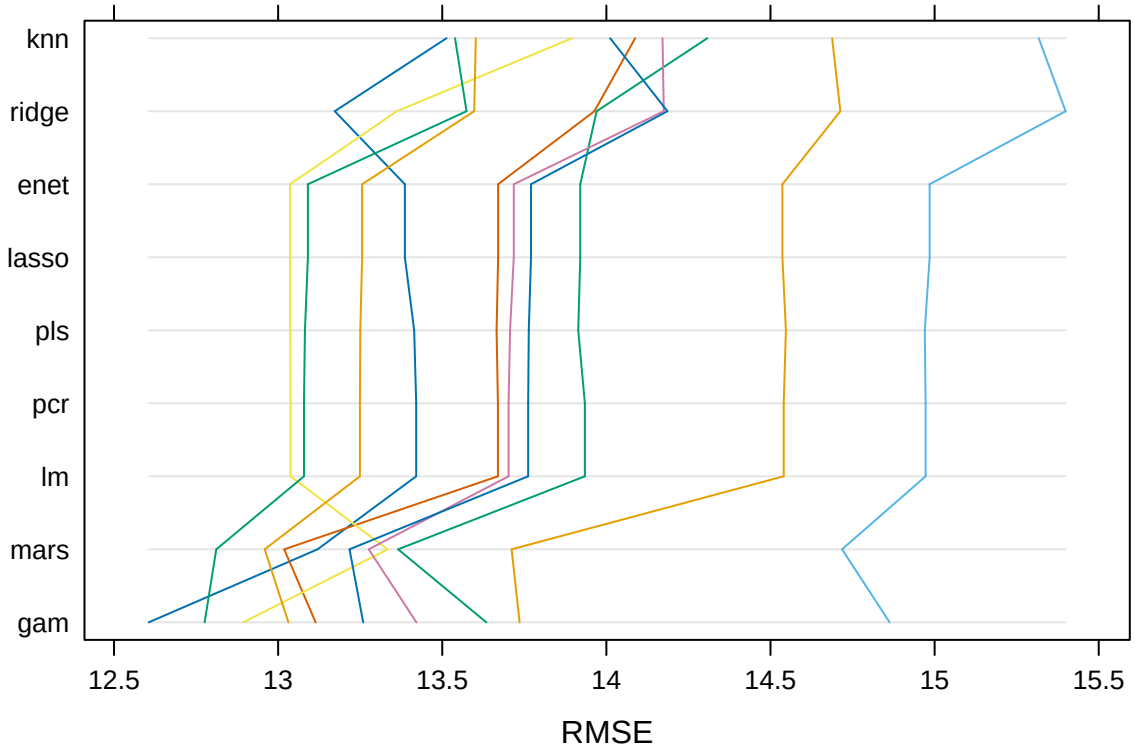
## Model Comparison

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

summary(resamp)
```

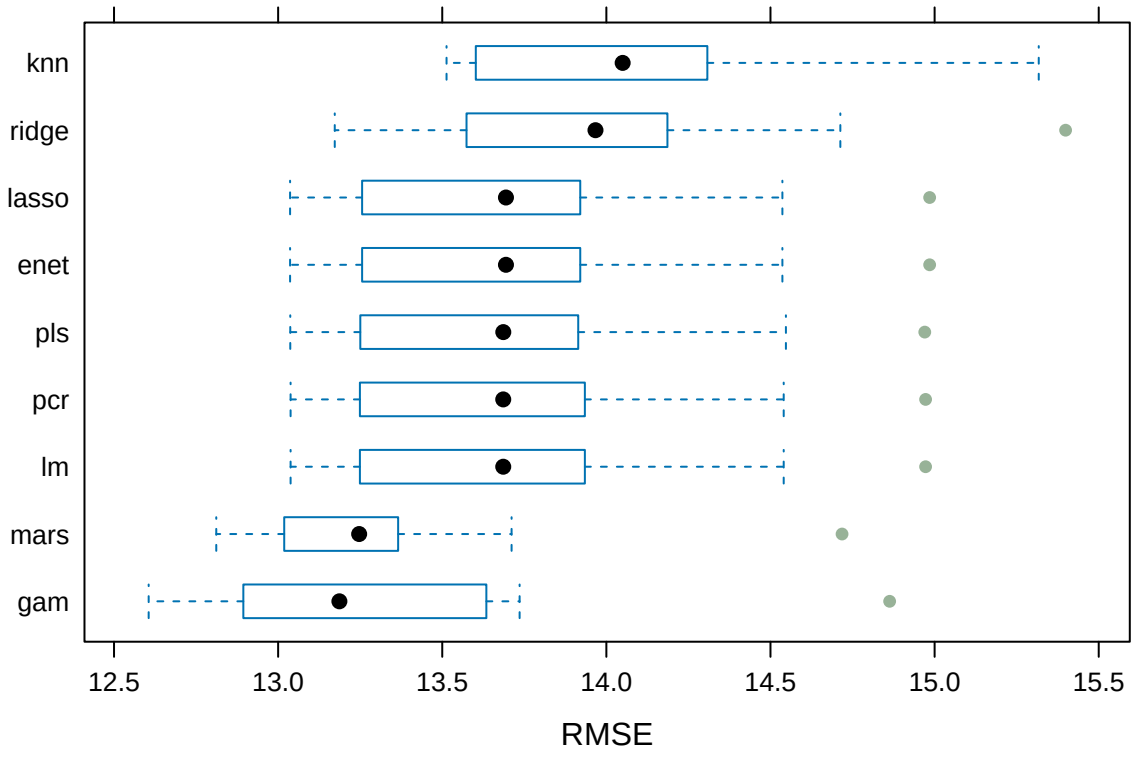
```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: knn, ridge, lasso, enet, pcr, pls, gam, mars, lm
## Number of resamples: 10
```

```
##
## MAE
##            Min.   1st Qu.    Median     Mean  3rd Qu.     Max. NA's
## knn   10.236660 10.621760 10.803936 10.87859 11.13757 11.67292    0
## ridge 10.080176 10.271325 10.588887 10.65027 10.92553 11.34986    0
## lasso  9.702236 10.107696 10.358916 10.46835 10.84601 11.30065    0
## enet   9.702236 10.107696 10.358916 10.46835 10.84601 11.30065    0
## pcr    9.680482 10.112251 10.356188 10.46983 10.86352 11.30953    0
## pls    9.679312 10.115680 10.355105 10.47002 10.85469 11.31507    0
## gam    9.502578  9.831854  9.971805 10.16287 10.50255 11.29174    0
## mars   9.474960  9.873973  9.994118 10.15124 10.44452 11.15498    0
## lm     9.680482 10.112251 10.356188 10.46983 10.86352 11.30953    0
##
## RMSE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## knn   13.51307 13.67554 14.04961 14.11328 14.27353 15.31733    0
## ridge 13.17253 13.57976 13.96685 14.01120 14.18328 15.39914    0
## lasso 13.03641 13.28840 13.69406 13.73692 13.88296 14.98491    0
## enet  13.03641 13.28840 13.69406 13.73692 13.88296 14.98491    0
## pcr   13.03798 13.29199 13.68583 13.73672 13.89133 14.97256    0
## pls   13.03702 13.29137 13.68600 13.73505 13.87660 14.97008    0
## gam   12.60555 12.92835 13.18663 13.33346 13.58112 14.86304    0
## mars  12.81152 13.04393 13.24685 13.35313 13.35759 14.71806    0
## lm    13.03798 13.29199 13.68583 13.73672 13.89133 14.97256    0
##
## Rsquared
##             Min.    1st Qu.     Median       Mean    3rd Qu.      Max. NA's
## knn   0.05121361 0.06160456 0.06707782 0.07199062 0.07619279 0.1063808    0
## ridge 0.04209882 0.06501407 0.07834877 0.08183359 0.10392961 0.1210925    0
## lasso 0.08658456 0.10701959 0.11357611 0.11571147 0.12437460 0.1600097    0
## enet  0.08658456 0.10701959 0.11357611 0.11571147 0.12437460 0.1600097    0
## pcr   0.08376913 0.10809707 0.11452708 0.11601952 0.12553731 0.1597262    0
## pls   0.08440029 0.10848507 0.11472165 0.11623342 0.12505041 0.1598645    0
## gam   0.10640479 0.14539541 0.17728766 0.16842279 0.18625184 0.2205409    0
## mars  0.12334375 0.15603178 0.16467912 0.17022399 0.19381735 0.2184433    0
## lm    0.08376913 0.10809707 0.11452708 0.11601952 0.12553731 0.1597262    0
```

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




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




GAM has lowest mean and median RMSE -> model I pick. GAM is a good choice since it incorporates non-linear terms by adding the smoothing function, as well as linear terms. GAM also performs model selection for us.

## Test Data Simulation