Analysis

Arthur Starodynov, Ekaterina Hofrenning, Lauren Lazaro

2024-03-24

```
library(tidyverse)
library(caret)
library(mgcv)
library(earth)
library(leaps)
```

Load in the data and explore the data.

```
load("data/recovery.Rdata")
head(dat)
```

```
##   id age gender race smoking height weight  bmi hypertension diabetes SBP LDL
## 1  1  56      0    1       2  170.2   78.7 27.2            0        0 120  97
## 2  2  70      1    1       1  169.6   73.1 25.4            1        0 134 112
## 3  3  57      1    1       0  168.4   77.4 27.3            1        0 131  88
## 4  4  53      0    1       0  166.7   76.1 27.4            0        0 115  87
## 5  5  59      1    1       2  173.6   70.2 23.3            0        0 127 118
## 6  6  60      1    3       1  162.8   75.1 28.4            0        0 129 104
##   vaccine severity study recovery_time
## 1       0        0     A            31
## 2       0        0     A            44
## 3       1        0     A            29
## 4       0        1     A            47
## 5       1        0     A            40
## 6       0        0     A            34
```

```
set.seed(2024)
```

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

#dfSummary(data[,-1])
```

```
data <- dat %>%
  select(-id)   # removing the id variable from the data
set.seed(2024)
```

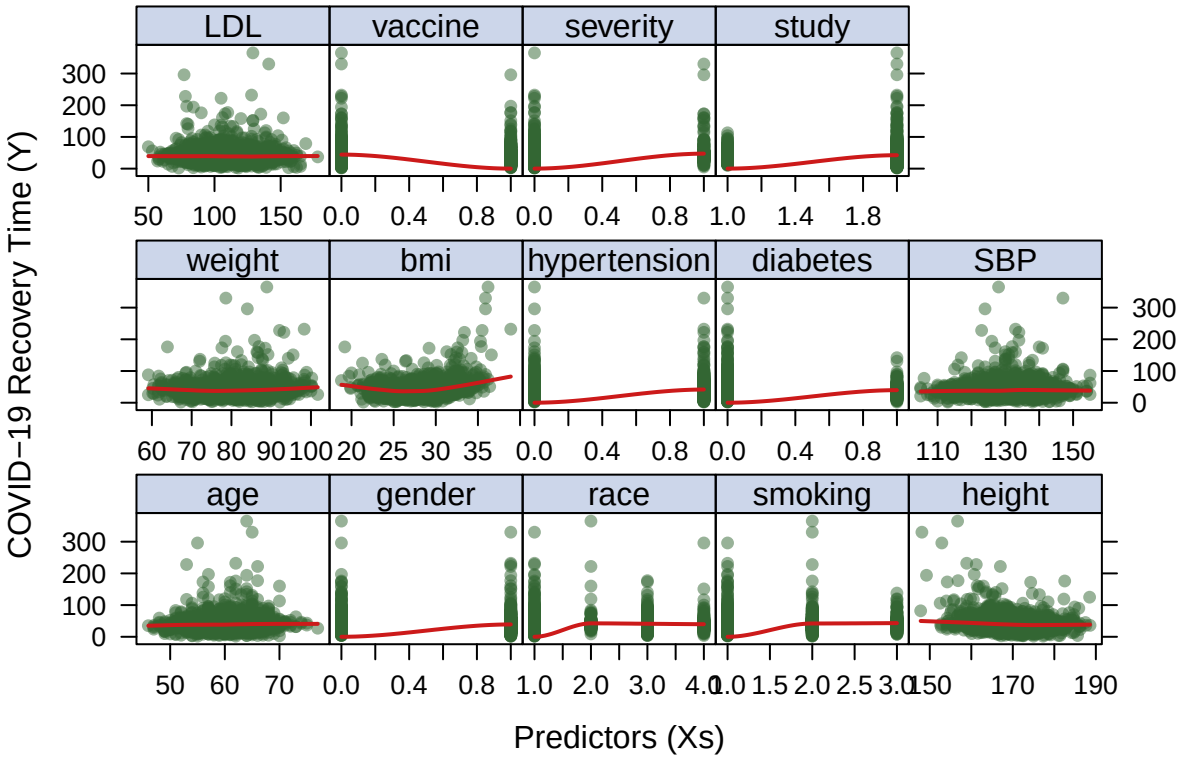
```r
tRows <- createDataPartition(dat$recovery_time, p = 0.7, list = FALSE)
# training data
data_train <- data[tRows, ]
x <- model.matrix(recovery_time~.,data)[tRows,-1]
y <- data$recovery_time[tRows]

#Test data
data_test <- data[-tRows, ]
x2 <- model.matrix(recovery_time~.,data)[-tRows,-1]
y2 <- data$recovery_time[-tRows]
```

Exploring the data set :

```r
data_train_viz <- data_train %>%
  mutate(study = case_when( # turn study (character variable) into a numeric variable
    study == "A" ~ 1,
    study == "B" ~ 2,
    study == "C" ~ 3))
non_numeric_cols <- sapply(data_train_viz, function(x) !is.numeric(x))
# Convert non-numeric columns to numeric
data_train_viz[, non_numeric_cols] <- lapply(data_train_viz[, non_numeric_cols], as.numeric)
```

```r
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 = data_train_viz[ ,1:14],
            y = data_train_viz[ ,15],
            plot = "scatter",
            span = .5,
            labels = c("Predictors (Xs)", "COVID-19 Recovery Time (Y)"),
            main = "Figure 1. Lattice Plot",
            type = c("p", "smooth"))
```

Figure 1. Lattice Plot



Training models

Training Various models to see which will perform the best.

LGM:

```
set.seed(2024)
ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 5) # Using the best rule
linear_model <- train(recovery_time ~ .,
                data = data_train,
                method = "lm",
                trControl = ctrl)
summary(linear_model)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.256 -11.308  -0.031   8.772 248.452
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.327e+03  1.235e+02 -18.849  < 2e-16 ***
## age           2.290e-01  1.131e-01   2.026 0.042941 *
## gender       -3.220e+00  8.879e-01  -3.627 0.000294 ***
## race2         4.179e+00  2.097e+00   1.992 0.046451 *
```

```
## race3        -7.288e-01  1.132e+00  -0.644 0.519609
## race4        -4.931e-01  1.562e+00  -0.316 0.752287
## smoking1      2.409e+00  1.007e+00   2.393 0.016815 *
## smoking2      3.094e+00  1.470e+00   2.105 0.035408 *
## height        1.360e+01  7.263e-01  18.721  < 2e-16 ***
## weight       -1.482e+01  7.659e-01 -19.356  < 2e-16 ***
## bmi           4.430e+01  2.199e+00  20.147  < 2e-16 ***
## hypertension  1.917e+00  1.457e+00   1.316 0.188388
## diabetes     -2.118e+00  1.220e+00  -1.737 0.082585 .
## SBP           6.747e-02  9.462e-02   0.713 0.475889
## LDL          -6.237e-02  2.354e-02  -2.650 0.008120 **
## vaccine      -6.935e+00  9.061e-01  -7.654 2.96e-14 ***
## severity      8.689e+00  1.424e+00   6.102 1.24e-09 ***
## studyB        5.139e+00  9.427e-01   5.452 5.58e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.29 on 2084 degrees of freedom
## Multiple R-squared:  0.2817, Adjusted R-squared:  0.2758
## F-statistic: 48.07 on 17 and 2084 DF,  p-value: < 2.2e-16
```

Finding the RMSE:

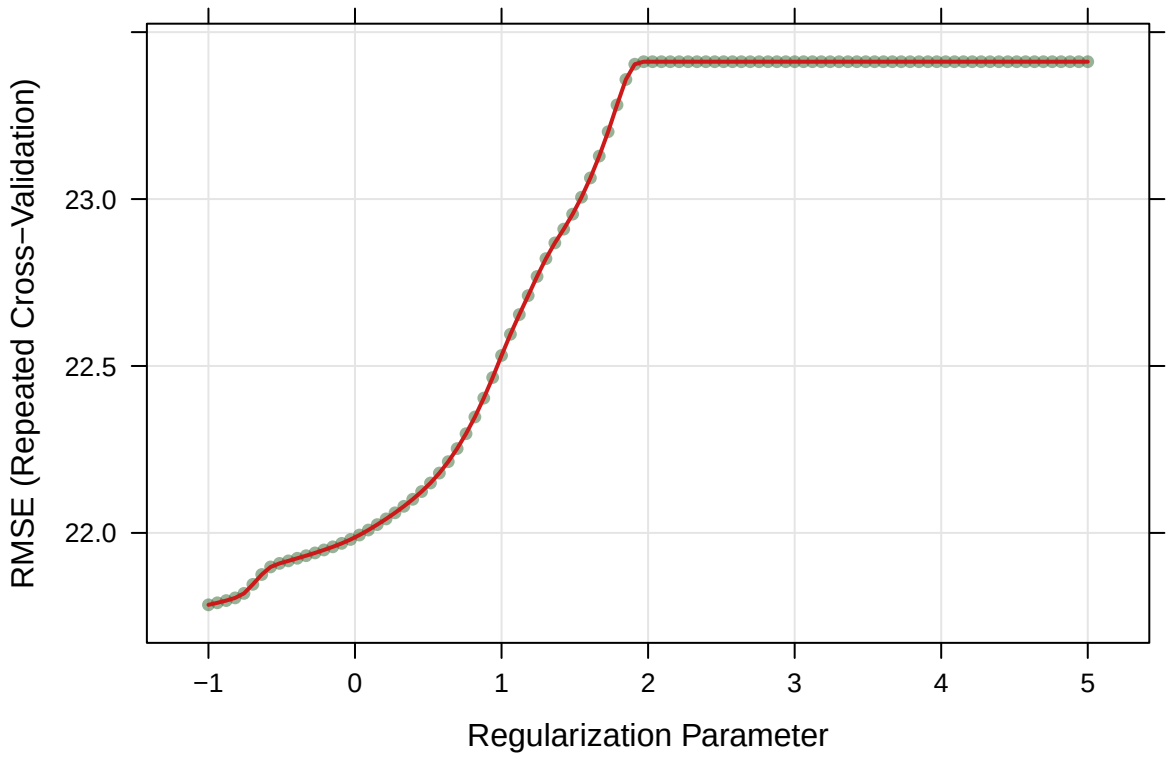
```
linear_pred <- predict(linear_model, newdata = data_test)
linear_rmse <- sqrt(mean((linear_pred - data_test$recovery_time)^2))
linear_rmse
```

```
## [1] 19.85811
```

We can see that the RMSE is 19.858 for the Generalized linear model.

Lasso Model

```
set.seed(2024)
lasso_model <- train(x, y,
                     method = "glmnet",
                     tuneGrid = expand.grid(alpha = 1,
                                            lambda = exp(seq(-1, 5, length = 100))),
                     trControl = ctrl)

plot(lasso_model, xTrans = log)
```




```
tuning_param <- lasso_model$bestTune
coef(lasso_model$finalModel, lasso_model$bestTune$lambda)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)  -25.43825107
## age            0.19265246
## gender        -1.93903073
## race2          3.10474378
## race3          .
## race4          .
## smoking1       0.94047609
## smoking2       1.41274621
## height         .
## weight        -0.40581611
## bmi            2.90669760
## hypertension   0.23950557
## diabetes      -1.66338038
## SBP            0.10053663
## LDL           -0.03119877
## vaccine       -6.45367070
## severity       7.96723577
## studyB         5.11509535
```

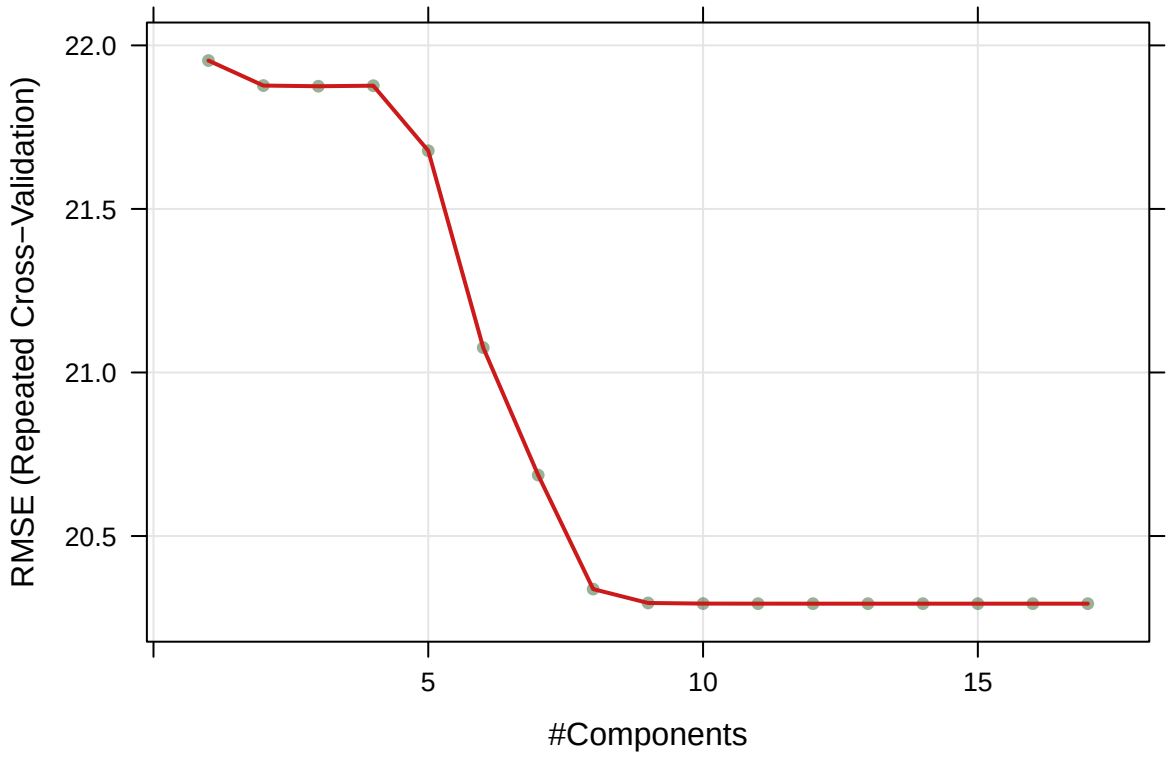
Finding RMSE of the Lasso model:

```
lasso_pred <- predict(lasso_model, newdata = x2)
lasso_rmse <- sqrt(mean((lasso_pred - data_test$recovery_time)^2))
lasso_rmse
```

```
## [1] 20.61116
```

PLS model

```
set.seed(2024)
pls_model <- train(x, y,
                   method = "pls",
                   tuneGrid = data.frame(ncomp = 1:17),
                   trControl = ctrl,
                   preProcess = c("center", "scale"))
print(plot(pls_model))
```



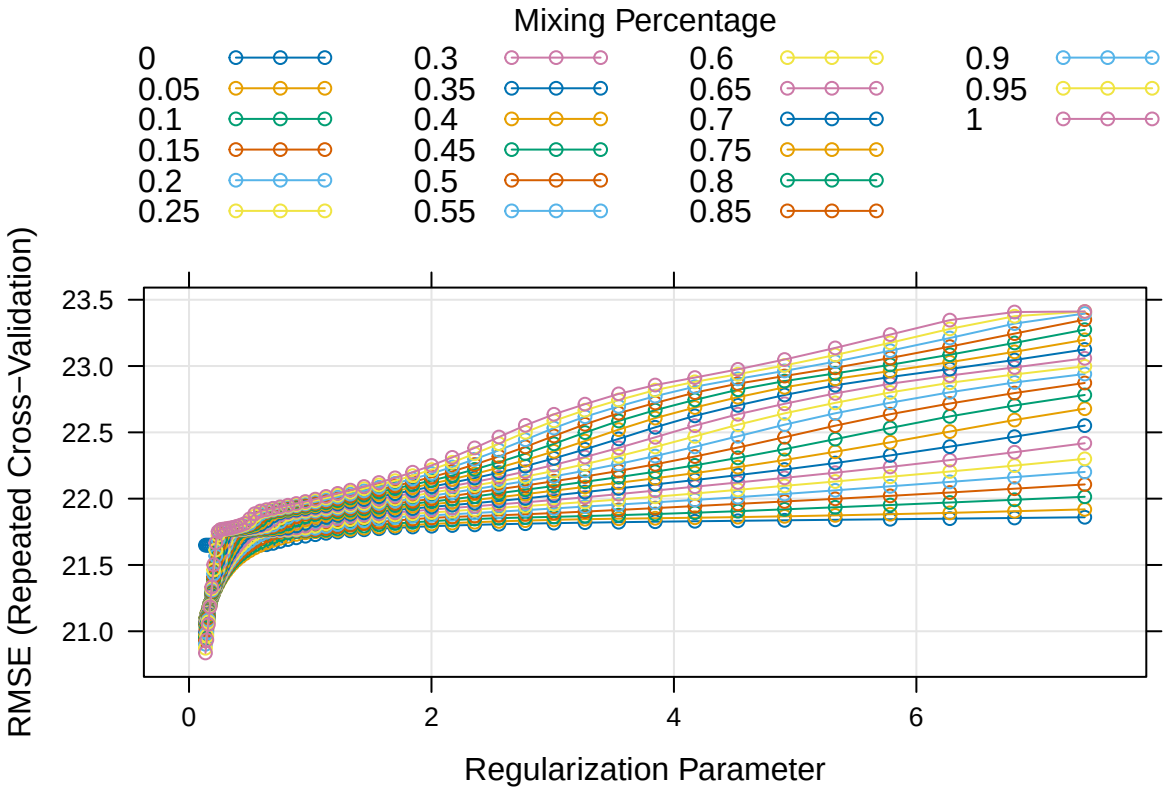
Finding the RMSE

```
pls_pred <- predict(pls_model, newdata = x2)
pls_rmse <- sqrt(mean((pls_pred - data_test$recovery_time)^2))
pls_rmse
```

```
## [1] 19.85812
```

Elastic net model

```
set.seed(2024)
enet_model <- train(x, y,
                    method = "glmnet",
                    tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                           lambda = exp(seq(2, -2, length = 50))),
                    trControl = ctrl)
print(plot(enet_model))
```



Finding the RMSE

```
enet_pred <- predict(enet_model, newdata = x2)
enet_rmse <- sqrt(mean((enet_pred - data_test$recovery_time)^2))
enet_rmse
```

```
## [1] 19.95906
```

MARS model

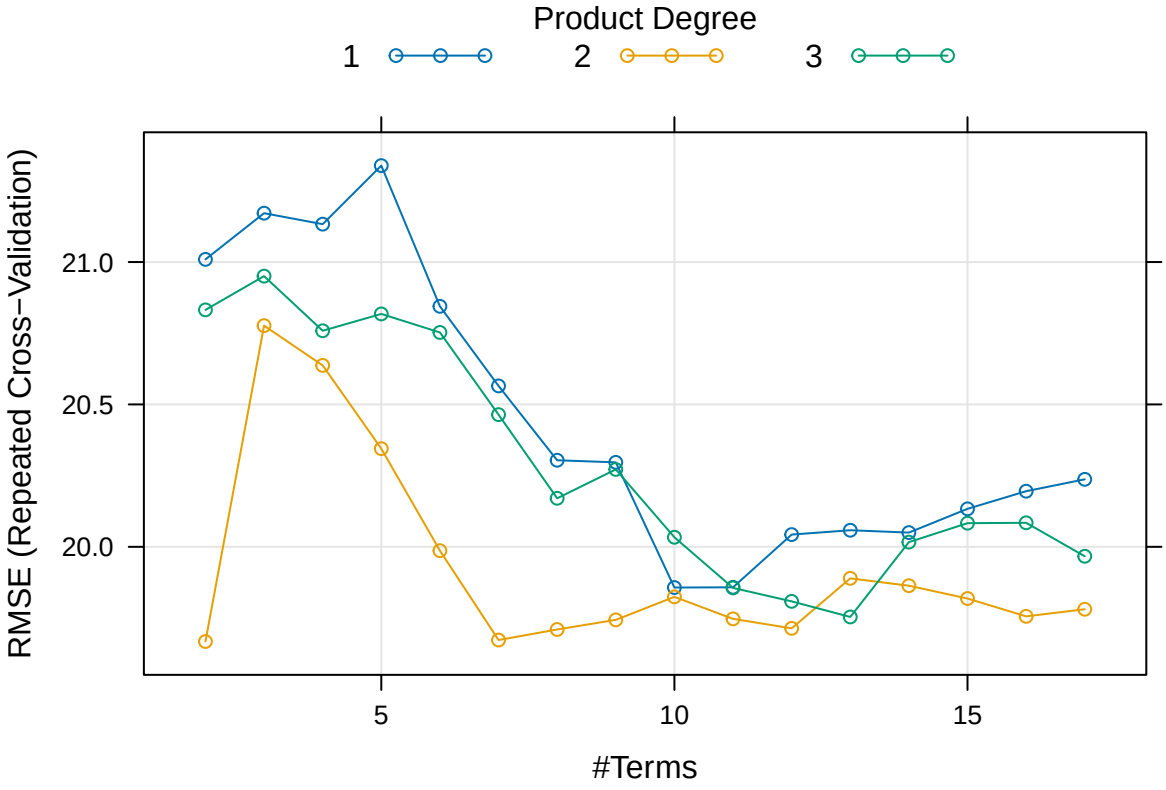
Resampling the data by encoding dummy variables to be able to use a MARS model.

```
set.seed(2024)
df_dummies <- data.frame(model.matrix(~ . - 1, data = dat[, c("gender", "race", "smoking", "hypertensio
                         age = dat$age,
                         height = dat$height,
                         weight = dat$weight,
                         bmi = dat$bmi,
                         SBP = dat$SBP,
                         LDL = dat$LDL,
                         recovery_time = dat$recovery_time)

data_mars <- df_dummies

#training
data_train_mars <- data_mars[tRows, ]
mars_x <- model.matrix(recovery_time~.,data_mars)[tRows,-1]
mars_y<- data_mars$recovery_time[tRows]

# test
```

```
data_test_mars <- data_mars[-tRows, ]
## matrix of predictors
mars_x2 <- model.matrix(recovery_time~.,data_mars)[-tRows,-1]
## vector of response
mars_y2 <- data_mars$recovery_time[-tRows]
```

Using newly created data to train a MARS model.

```
set.seed(2024)
mars_grid <- expand.grid(degree = 1:3,
                         nprune = 2:17)

mars_model <- train(mars_x, mars_y,
                    method = "earth",
                    tuneGrid = mars_grid,
                    trControl = ctrl)
print(plot(mars_model))
```



Find the RMSE

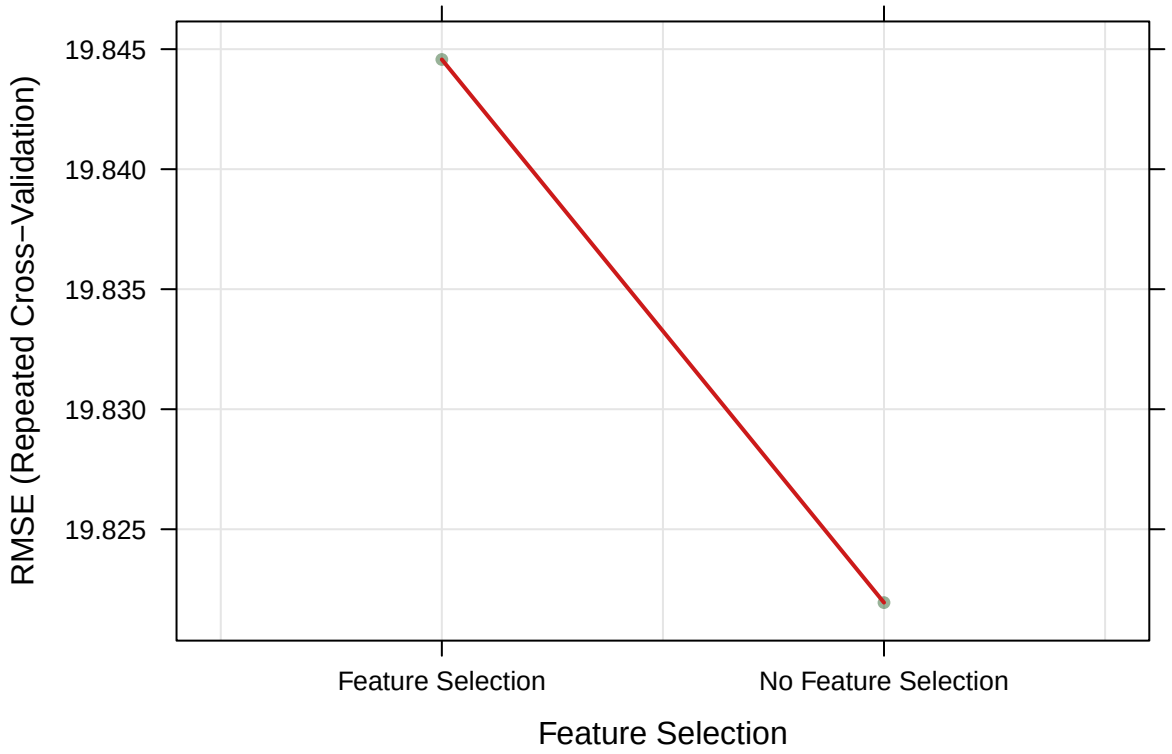
```
mars_pred <- predict(mars_model, newdata = mars_x2)
mars_rmse <- sqrt(mean((mars_pred - data_test_mars$recovery_time)^2))
mars_rmse
```

```
## [1] 18.05067
```

GAM Model

```
set.seed(2024)
```

```r
GAM_model <- train(x, y,
                   method = "gam",
                   trControl = ctrl,
                   control = gam.control(maxit = 150)) # <- adjusted for maxit failure

print(plot(GAM_model))
```




```r
summary(GAM_model$finalModel)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender + race3 + race4 + smoking1 + smoking2 + hypertension +
##     diabetes + vaccine + severity + studyB + s(age) + s(SBP) +
##     s(LDL) + s(bmi) + s(height) + s(weight)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   44.1392     1.1752  37.560  < 2e-16 ***
## gender        -3.5459     0.8381  -4.231 2.43e-05 ***
## race3         -0.6689     1.0599  -0.631  0.52803
## race4         -1.0546     1.4686  -0.718  0.47276
## smoking1       2.6229     0.9504   2.760  0.00583 **
## smoking2       3.6774     1.3880   2.649  0.00813 **
## hypertension   1.9348     1.4111   1.371  0.17048
## diabetes      -2.0844     1.1518  -1.810  0.07050 .
## vaccine       -7.0273     0.8545  -8.224 3.43e-16 ***
## severity       9.3538     1.3413   6.974 4.13e-12 ***
```

```
## studyB         4.6053     0.8908   5.170 2.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F  p-value
## s(age)    1.000  1.000  1.677 0.195457
## s(SBP)    1.478  1.813  0.786 0.543422
## s(LDL)    1.219  1.408  3.834 0.027315 *
## s(bmi)    7.199  8.167 57.524  < 2e-16 ***
## s(height) 6.908  7.956  9.740  < 2e-16 ***
## s(weight) 1.670  2.194  6.925 0.000453 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.358   Deviance explained = 36.7%
## GCV = 370.41  Scale est. = 365.04    n = 2102
```

Finding the RMSE:

```
GAM_pred <- predict(GAM_model, newdata = x2)
GAM_rmse <- sqrt(mean((GAM_pred - data_test$recovery_time)^2))
GAM_rmse
```

```
## [1] 18.48635
```

Model Selection

For choosing the best model, we assessed the RMSE and checked which model had the lowest RMSE. This indicates that the model is the best performing due to a low prediction error. Models with high RMSE has high prediction error meaning they are worse models.

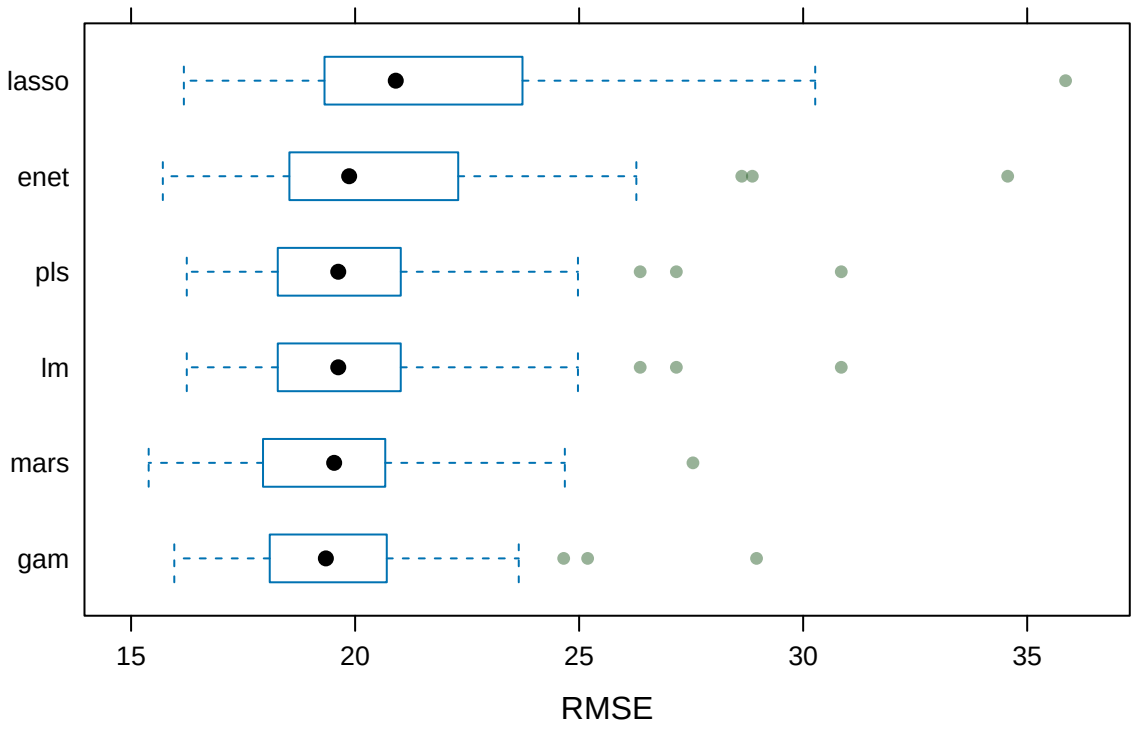
```
set.seed(2024)

resample_data <- resamples(list(
  lm = linear_model,
  lasso = lasso_model,
  enet = enet_model,
  pls = pls_model,
  gam = GAM_model,
  mars = mars_model
  ))

summary(resample_data)
```

```
##
## Call:
## summary.resamples(object = resample_data)
##
## Models: lm, lasso, enet, pls, gam, mars
## Number of resamples: 50
##
## MAE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lm    12.01315 13.18428 13.42575 13.55674 13.93071 15.25681    0
```

```
## lasso 11.68470 13.14360 13.58178 13.65305 14.00670 16.14233    0
## enet  11.64707 12.77613 13.24593 13.30991 13.69025 15.49942    0
## pls   12.01308 13.18429 13.42575 13.55674 13.93070 15.25680    0
## gam   11.37981 12.28795 12.91906 12.99219 13.49834 14.52340    0
## mars  11.46416 12.40770 13.04918 13.02956 13.47370 15.26110    0
##
## RMSE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lm    16.24262 18.34594 19.62327 20.29317 21.01705 30.85055    0
## lasso 16.17754 19.31831 20.90705 21.78457 23.72520 35.85732    0
## enet  15.70800 18.53689 19.86654 20.83649 22.29719 34.56517    0
## pls   16.24261 18.34592 19.62328 20.29317 21.01705 30.85054    0
## gam   15.96439 18.12753 19.34727 19.82194 20.70164 28.95957    0
## mars  15.39199 18.03713 19.53205 19.66747 20.66927 27.54021    0
##
## Rsquared
##                Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm    8.905049e-02 0.1802640 0.2591439 0.2617221 0.3401823 0.4626231    0
## lasso 1.426456e-02 0.1158527 0.1460068 0.1482118 0.1843823 0.2711260    0
## enet  7.122077e-02 0.1692736 0.2205285 0.2234573 0.2806290 0.3810741    0
## pls   8.905086e-02 0.1802631 0.2591458 0.2617223 0.3401832 0.4626238    0
## gam   9.561505e-02 0.1879885 0.2928082 0.3091493 0.4603588 0.5413315    0
## mars  4.678139e-05 0.1377927 0.2789584 0.3029493 0.4813938 0.6480672    0
```

```
bwplot(resample_data,
       metric = "RMSE",
       main = "Figure 2. Model Comparison Plot Using RMSE")
```

Figure 2. Model Comparison Plot Using RMSE



Results

```r
summary(mars_model$finalModel)
```

```
## Call: earth(x=matrix[2102,18], y=c(31,47,40,34,3...), keepxy=TRUE, degree=2,
##             nprune=2)
##
##                     coefficients
## (Intercept)            40.14471
## studyB * h(bmi-31)     30.05710
##
## Selected 2 of 26 terms, and 2 of 18 predictors (nprune=2)
## Termination condition: Reached nk 37
## Importance: studyB, bmi, gender-unused, race1-unused, race2-unused, ...
## Number of terms at each degree of interaction: 1 0 1
## GCV 383.1087    RSS 802615    GRSq 0.3266962    RSq 0.3282976
```

```r
summary(GAM_model$finalModel)
```

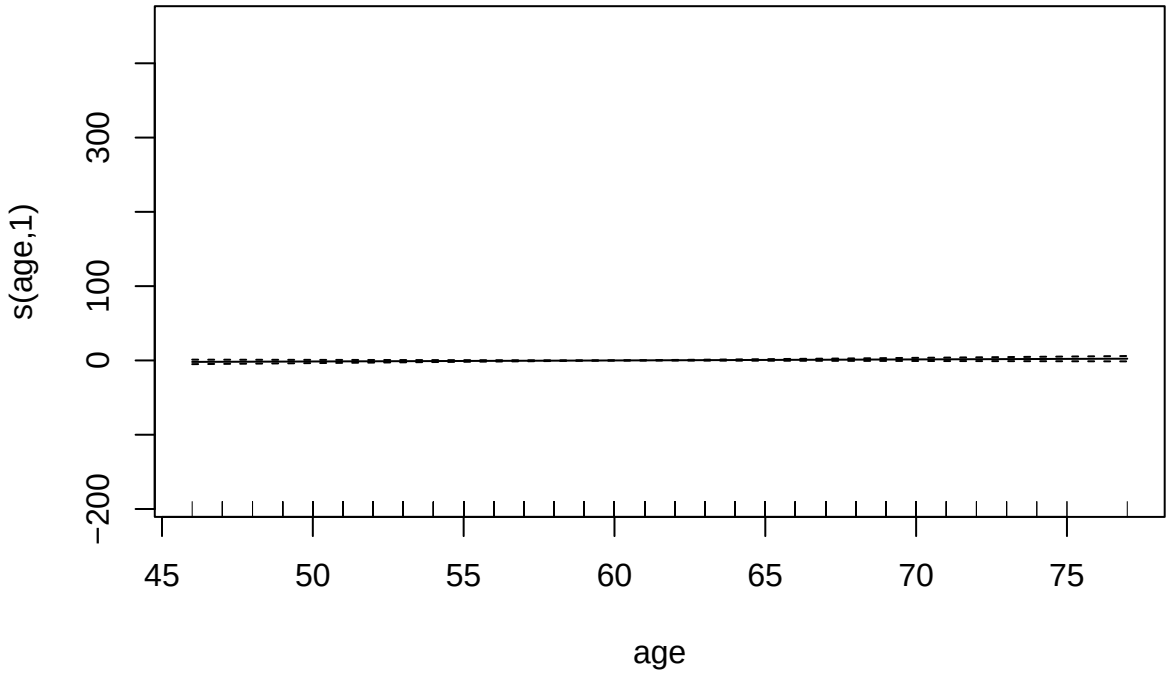
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender + race3 + race4 + smoking1 + smoking2 + hypertension +
##     diabetes + vaccine + severity + studyB + s(age) + s(SBP) +
##     s(LDL) + s(bmi) + s(height) + s(weight)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   44.1392     1.1752  37.560  < 2e-16 ***
## gender        -3.5459     0.8381  -4.231 2.43e-05 ***
## race3         -0.6689     1.0599  -0.631  0.52803
## race4         -1.0546     1.4686  -0.718  0.47276
## smoking1       2.6229     0.9504   2.760  0.00583 **
## smoking2       3.6774     1.3880   2.649  0.00813 **
## hypertension   1.9348     1.4111   1.371  0.17048
## diabetes      -2.0844     1.1518  -1.810  0.07050 .
## vaccine       -7.0273     0.8545  -8.224 3.43e-16 ***
## severity       9.3538     1.3413   6.974 4.13e-12 ***
## studyB         4.6053     0.8908   5.170 2.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F  p-value
## s(age)    1.000  1.000  1.677 0.195457
## s(SBP)    1.478  1.813  0.786 0.543422
## s(LDL)    1.219  1.408  3.834 0.027315 *
## s(bmi)    7.199  8.167 57.524  < 2e-16 ***
## s(height) 6.908  7.956  9.740  < 2e-16 ***
## s(weight) 1.670  2.194  6.925 0.000453 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

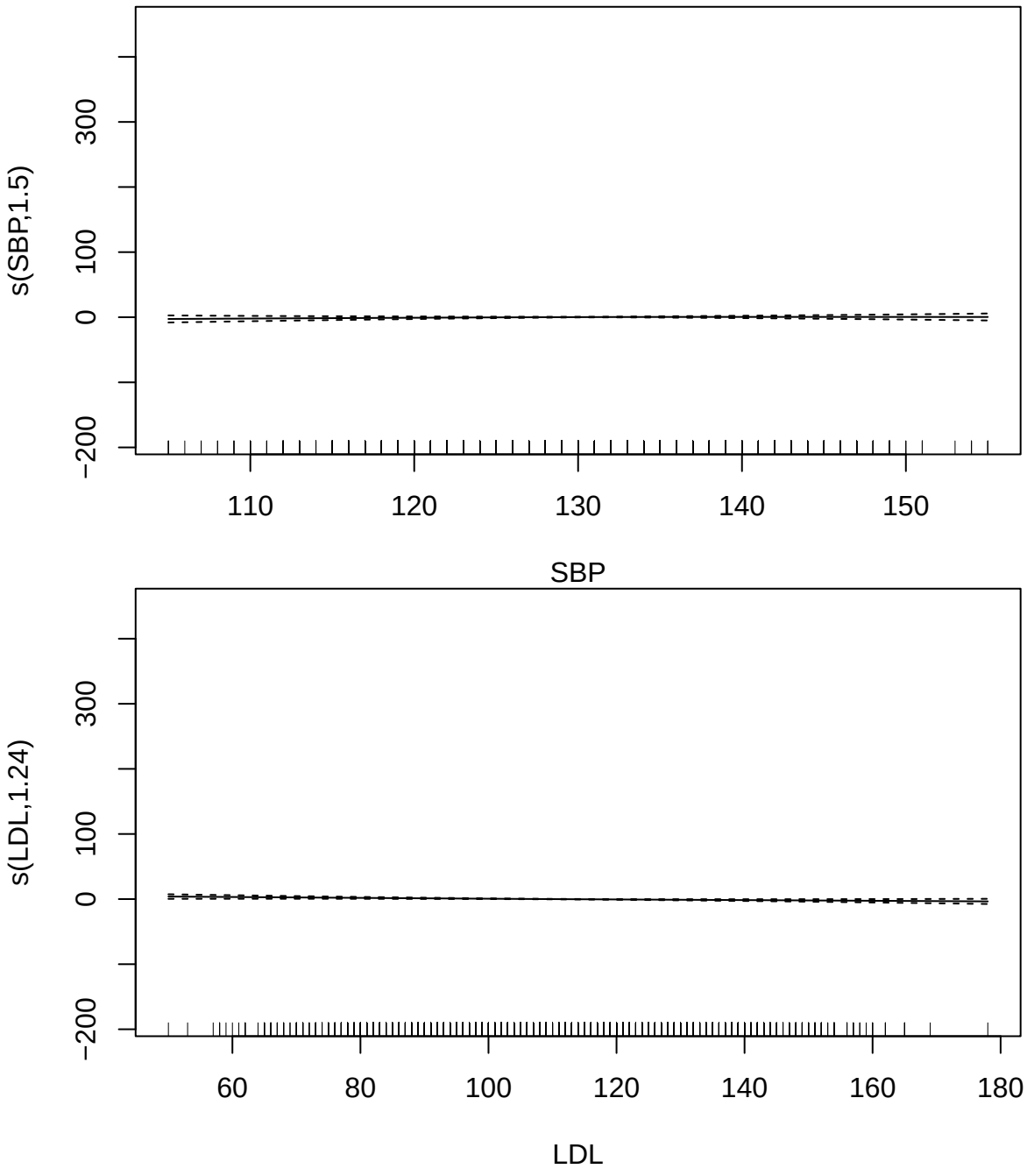
```
## R-sq.(adj) =  0.358   Deviance explained = 36.7%
## GCV = 370.41  Scale est. = 365.04    n = 2102
```

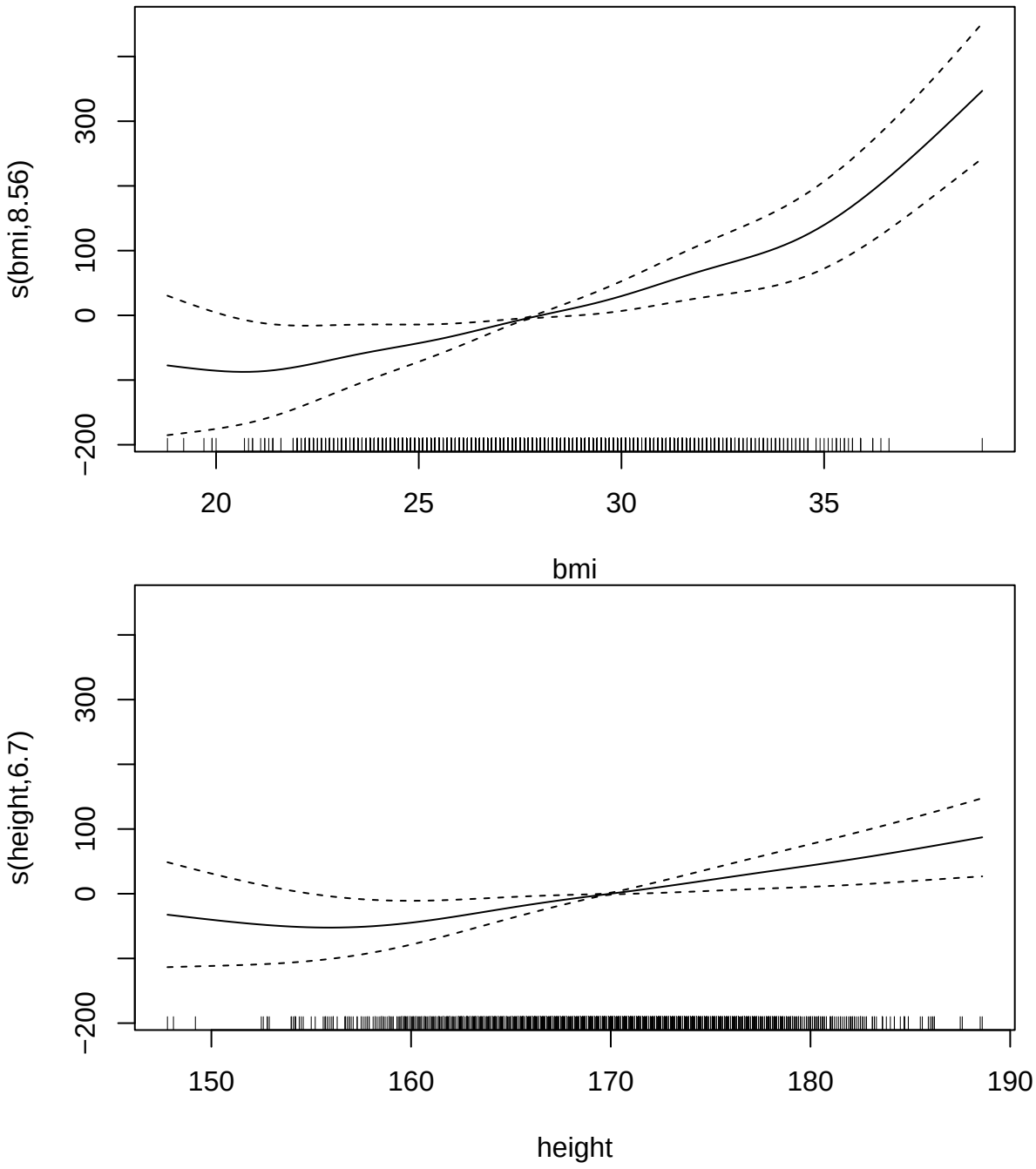
Although we see through the RMSE comparison that the mean and median RMSE of the mars model was the smallest when we compare the number of predictors used for the mars model was 2, we know that this would not be a good and accurate model for future use. Hence, we want to use the GAM model for any further comparison.

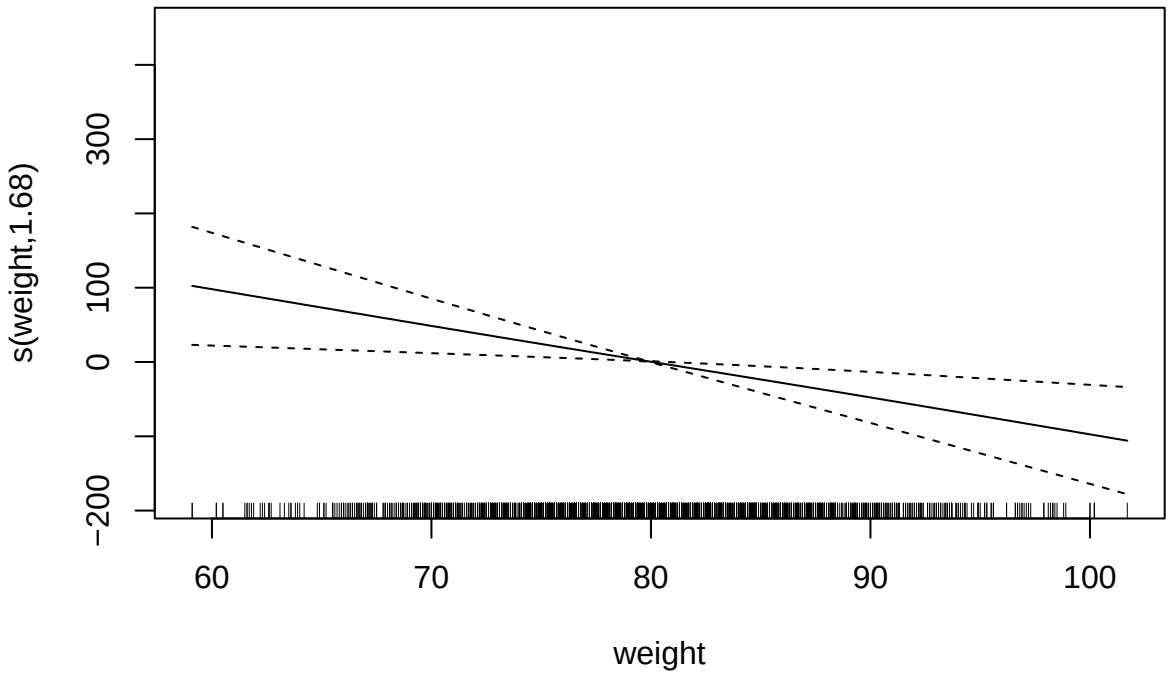
Appendix / GAM plots

```
gam.m1 <- gam(recovery_time ~ gender + race + smoking + hypertension +
    diabetes + vaccine + severity + study + s(age) + s(SBP) +
    s(LDL) + s(bmi) + s(height) + s(weight), data = data_train)

plot(gam.m1)
```




s(SBP,1.5)
300
100
0
−200
110
120
130
140
150
SBP
s(LDL,1.24)
300
100
0
−200
60
80
100
120
140
160
180
LDL


s(bmi,8.56)
300
100
0
−200
20
25
30
35
bmi
s(height,6.7)
300
100
0
−200
150
160
170
180
190
height


s(weight,1.68)
300
100
0
−200
60
70
80
90
100
weight