P8106_group2recovery_primaryanalysis

Yimin Chen (yc4195), Yang Yi (yy3307), Qingyue Zhuo (qz2493)

Contents

Import and data manipulation **1**

Model training **3**

Least squares . . . 3
Ridge . . . 4
LASSO . . . 5
PCR & PLS tune . . . 7
PCR . . . 8
PLS . . . 9
GAM . . . 10
MARS . . . 11
Regression tree . . . 13
Random Forest . . . 15
Boosting . . . 16

Model selection **17**

Import and data manipulation

```
# Load recovery.RData environment
load("./recovery.Rdata")

dat %>% na.omit()

# dat1 draw a random sample of 2000 participants Uni:3307
set.seed(3307)

dat1 = dat[sample(1:10000, 2000),]

dat1 =
  dat1[, -1] %>%
  mutate(
```

```
    gender = as.factor(gender),
    race = as.factor(race),
    smoking = as.factor(smoking),
    hypertension = as.factor(hypertension),
    diabetes = as.factor(diabetes),
    vaccine = as.factor(vaccine),
    severity = as.factor(severity),
    study = as.factor(
      case_when(study == "A" ~ 1, study == "B" ~ 2, study == "C" ~ 3)
      )
    )

# dat2 draw a random sample of 2000 participants Uni:2493
set.seed(2493)

dat2 = dat[sample(1:10000, 2000),]

dat2 =
  dat2[, -1] %>%
  mutate(
    gender = as.factor(gender),
    race = as.factor(race),
    smoking = as.factor(smoking),
    hypertension = as.factor(hypertension),
    diabetes = as.factor(diabetes),
    vaccine = as.factor(vaccine),
    severity = as.factor(severity),
    study = as.factor(
      case_when(study == "A" ~ 1, study == "B" ~ 2, study == "C" ~ 3)
      )
    )

# Merged dataset with unique observation
covid_dat = rbind(dat1, dat2) %>%
  unique()

covid_dat2 = model.matrix(recovery_time ~ ., covid_dat)[, -1]

# Partition dataset into two parts: training data (70%) and test data (30%)
rowTrain = createDataPartition(y = covid_dat$recovery_time, p = 0.7, list = FALSE)

trainData = covid_dat[rowTrain, ]
testData = covid_dat[-rowTrain, ]

# matrix of predictors
x1 = covid_dat2[rowTrain,]
# vector of response
y1 = covid_dat$recovery_time[rowTrain]
# matrix of predictors
x2 = covid_dat2[-rowTrain,]
# vector of response
y2 = covid_dat$recovery_time[-rowTrain]
```

```
ctrl1 = trainControl(method = "repeatedcv", number = 10, repeats = 5)
```

Model training

Least squares

```
num_cores <- detectCores()
cl <- makePSOCKcluster(num_cores)
registerDoParallel(cl)

set.seed(2)
ls.fit = train(x1, y1,
               method = "lm",
               trControl = ctrl1)
summary(ls.fit)
```

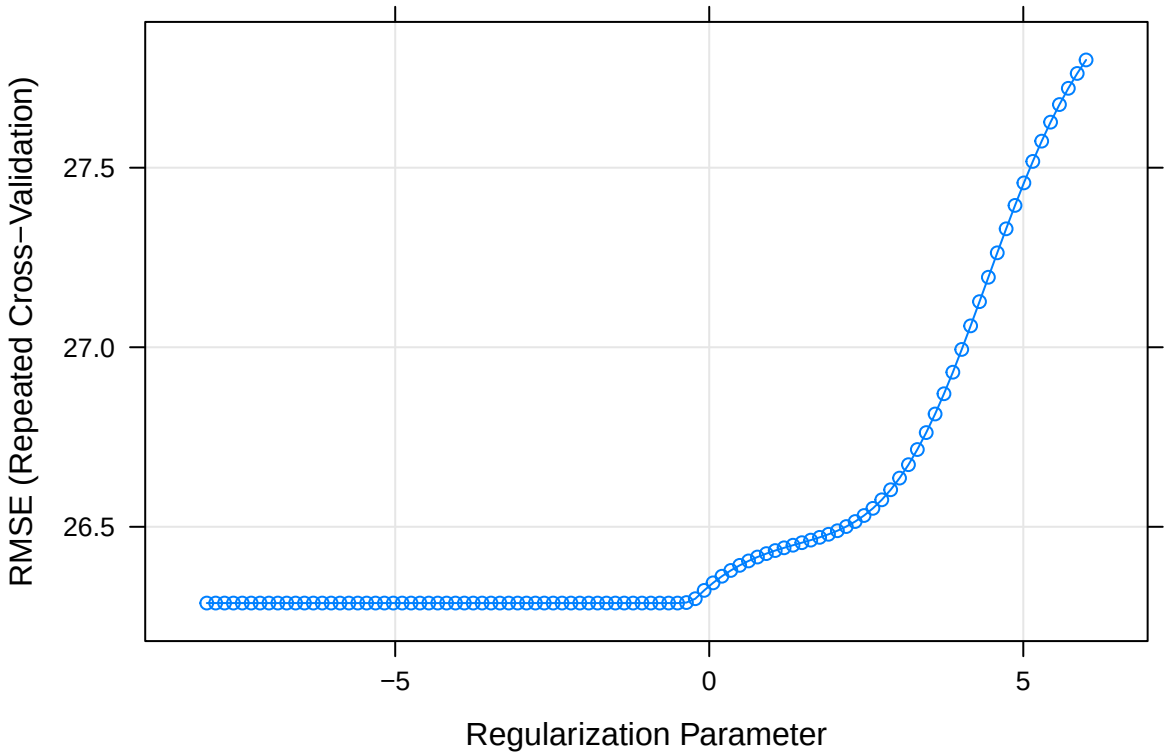
```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -82.641 -13.896  -2.076   9.749 296.936
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.552e+03  1.439e+02 -17.741  < 2e-16 ***
## age            2.949e-01  1.290e-01   2.287 0.022287 *
## gender1       -4.369e+00  1.010e+00  -4.325 1.59e-05 ***
## race2          9.035e-01  2.319e+00   0.390 0.696813
## race3         -8.613e-01  1.282e+00  -0.672 0.501793
## race4         -1.815e+00  1.723e+00  -1.053 0.292229
## smoking1       4.939e+00  1.124e+00   4.396 1.15e-05 ***
## smoking2       8.248e+00  1.719e+00   4.797 1.71e-06 ***
## height         1.487e+01  8.450e-01  17.601  < 2e-16 ***
## weight        -1.612e+01  8.933e-01 -18.040  < 2e-16 ***
## bmi            4.846e+01  2.554e+00  18.971  < 2e-16 ***
## hypertension1  4.381e+00  1.669e+00   2.625 0.008722 **
## diabetes1     -1.153e+00  1.383e+00  -0.834 0.404553
## SBP           -1.726e-02  1.108e-01  -0.156 0.876159
## LDL           -4.415e-02  2.656e-02  -1.663 0.096524 .
## vaccine1      -7.114e+00  1.028e+00  -6.921 5.67e-12 ***
## severity1      7.924e+00  1.627e+00   4.871 1.18e-06 ***
## study2         4.817e+00  1.328e+00   3.627 0.000292 ***
## study3         3.098e-01  1.599e+00   0.194 0.846409
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.21 on 2494 degrees of freedom
## Multiple R-squared:  0.2308, Adjusted R-squared:  0.2252
```

```
## F-statistic: 41.56 on 18 and 2494 DF,  p-value: < 2.2e-16
```

```
stopCluster(cl)
registerDoSEQ()
```

Ridge

```
num_cores <- detectCores()
cl <- makePSOCKcluster(num_cores)
registerDoParallel(cl)

set.seed(2)
ridge.fit = train(x1, y1,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = 0,
                  lambda = exp(seq(6, -8, length=100))),
                  trControl = ctrl1
                  )

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



```
ridge.fit$bestTune
```

```
##    alpha    lambda
## 54     0 0.6034751
```

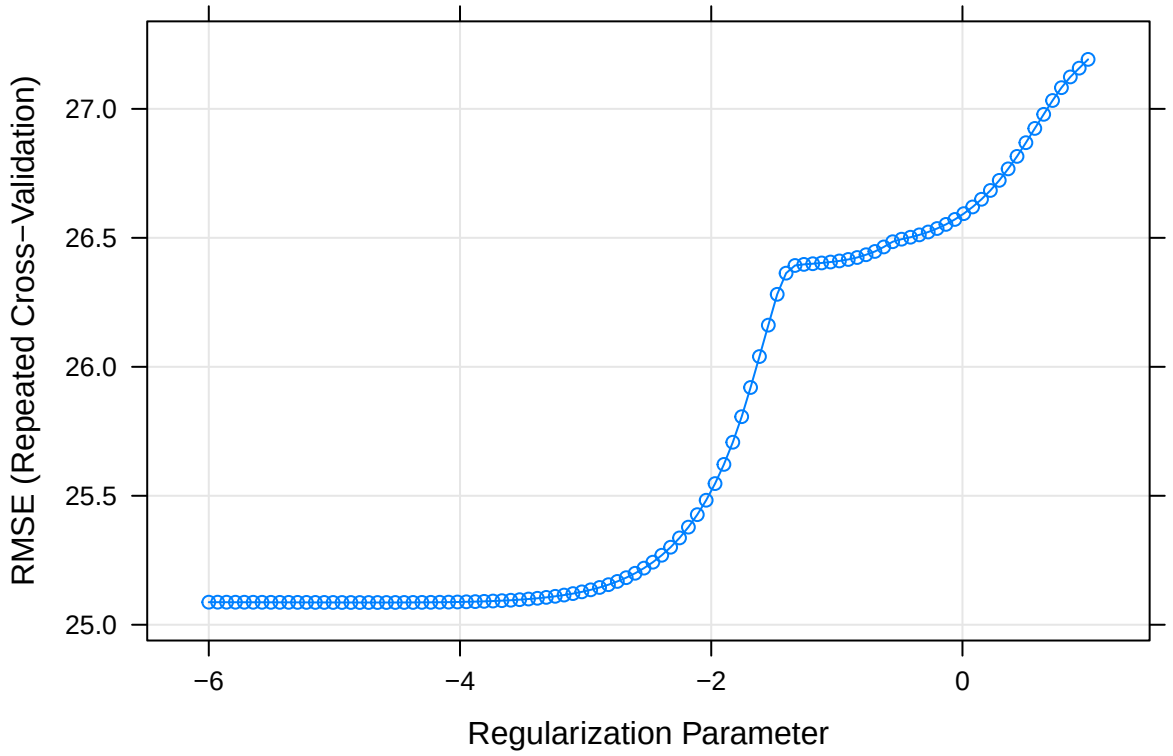
```
coef(ridge.fit$finalModel, s = ridge.fit$bestTune$lambda)
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)   -130.25302438
## age              0.27419110
## gender1         -4.07723482
## race2            1.37722089
## race3           -1.38073639
## race4           -1.99383241
## smoking1         4.76704041
## smoking2         7.58280696
## height           0.56438955
## weight          -0.95396678
## bmi              5.05146831
## hypertension1    3.81103518
## diabetes1       -1.49523236
## SBP              0.01393568
## LDL             -0.05005315
## vaccine1        -7.49149095
## severity1        7.35954976
## study2           5.06812978
## study3           0.41054465
```

```
stopCluster(cl)
registerDoSEQ()
```

LASSO

```
num_cores <- detectCores()
cl <- makePSOCKcluster(num_cores)
registerDoParallel(cl)

set.seed(2)
lasso.fit = train(x1, y1,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = 1,
                  lambda = exp(seq(1, -6, length=100))),
                  trControl = ctrl1
                  )

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




```
lasso.fit$bestTune
```

```
##    alpha      lambda
## 20     1 0.009499029
```

```
coef(lasso.fit$finalModel, lasso.fit$bestTune$lambda)
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)    -2.418158e+03
## age             2.900228e-01
## gender1        -4.338506e+00
## race2           8.964188e-01
## race3          -8.685445e-01
## race4          -1.790989e+00
## smoking1        4.915293e+00
## smoking2        8.184005e+00
## height          1.407858e+01
## weight         -1.527494e+01
## bmi             4.605112e+01
## hypertension1   4.273392e+00
## diabetes1      -1.148526e+00
## SBP            -9.935254e-03
## LDL            -4.412200e-02
## vaccine1       -7.125371e+00
```

```
## severity1      7.873188e+00
## study2         4.791477e+00
## study3         2.643080e-01
```

```
stopCluster(cl)
registerDoSEQ()
```

PCR & PLS tune

```
# Find ncomp for PCR
set.seed(2)
pcr.mod = pcr(recovery_time ~ .,
              data = trainData,
              scale = TRUE,
              validation = "CV")

summary(pcr.mod)
```

```
## Data:    X dimension: 2513 18
##  Y dimension: 2513 1
## Fit method: svdpc
## Number of components considered: 18
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           28.64    28.61    27.84    27.76     27.7    27.62    27.63
## adjCV        28.64    28.61    27.83    27.75     27.7    27.61    27.63
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       27.65    27.63    27.64     27.19     27.20     27.18     27.08
## adjCV    27.64    27.63    27.65     27.18     27.19     27.17     27.06
##        14 comps  15 comps  16 comps  17 comps  18 comps
## CV        27.07     27.06     27.02     27.02     25.51
## adjCV     27.05     27.05     27.00     27.00     25.49
##
## TRAINING: % variance explained
##                1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X               12.417   22.241   31.356   38.386   45.193   51.676   57.683
## recovery_time    0.281    6.484    6.889    7.304    8.107    8.113    8.136
##                8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## X               63.405   69.077     74.56     79.78     84.55     88.89
## recovery_time    8.249    8.348     11.14     11.19     11.36     12.17
##                14 comps  15 comps  16 comps  17 comps  18 comps
## X                 93.06     96.89     98.93     99.99    100.00
## recovery_time     12.27     12.37     12.63     12.67     23.08
```

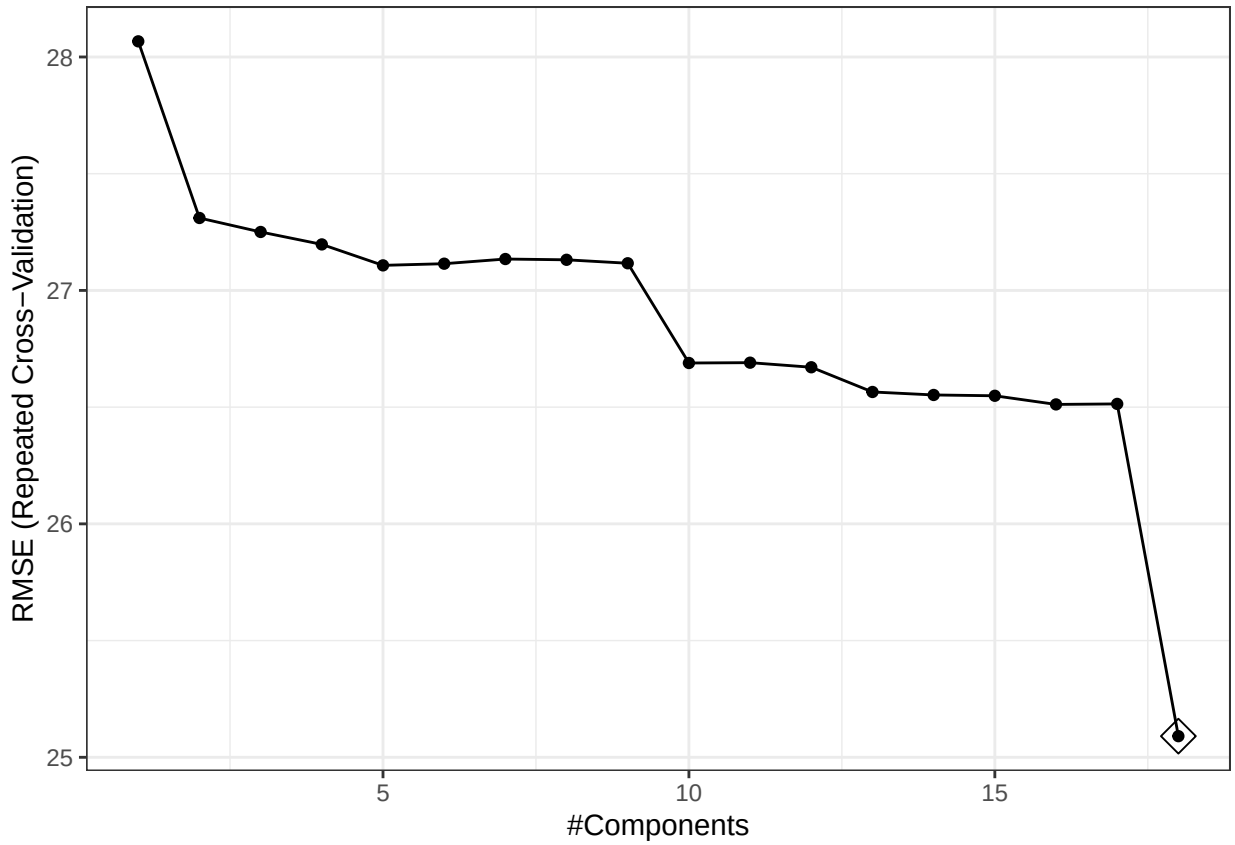
```
pls.mod = plsr(recovery_time ~ .,
          data = trainData,
          scale = TRUE,
          validation = "CV")

summary(pls.mod)
```

```
## Data:    X dimension: 2513 18
##  Y dimension: 2513 1
## Fit method: kernelpls
## Number of components considered: 18
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           28.64    27.14    27.02    27.01    26.99    26.96    26.88
## adjCV        28.64    27.13    27.00    26.99    26.97    26.95    26.87
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       26.62    26.12    25.70     25.50     25.49     25.50     25.49
## adjCV    26.63    26.12    25.67     25.48     25.47     25.48     25.47
##        14 comps  15 comps  16 comps  17 comps  18 comps
## CV        25.50     25.50     25.50     25.50     25.50
## adjCV     25.47     25.47     25.47     25.47     25.47
##
## TRAINING: % variance explained
##                 1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X                 9.101     16.1    25.94    31.30    36.15    41.88    46.78
## recovery_time    11.594     12.6    12.72    12.88    13.11    13.77    15.97
##                 8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## X                 50.39    52.65     57.02     62.14     67.31     72.58
## recovery_time     19.02    22.07     22.99     23.07     23.08     23.08
##                 14 comps  15 comps  16 comps  17 comps  18 comps
## X                  77.97     83.13     88.78     94.31    100.00
## recovery_time      23.08     23.08     23.08     23.08     23.08
```

The range of number of components considered is 1:18 for both PCR and PLS.

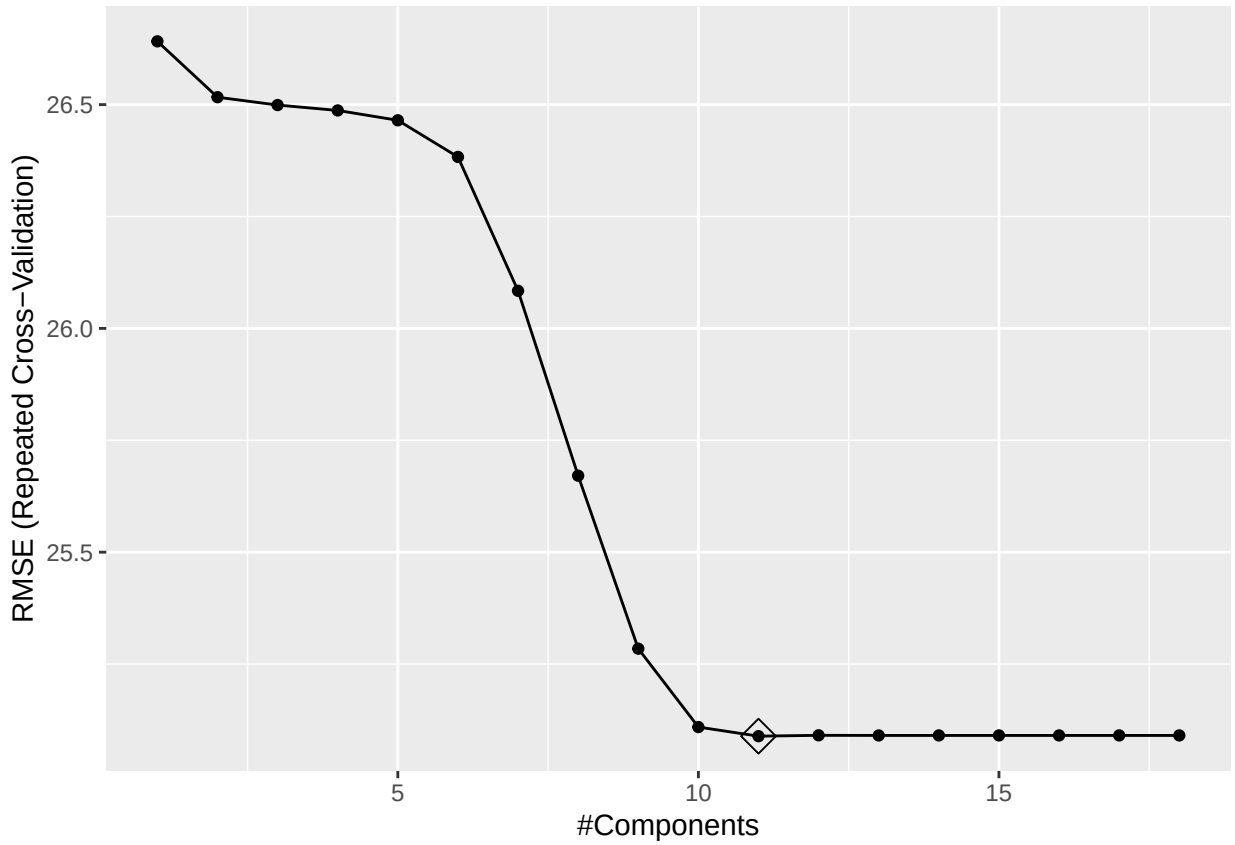
PCR

```
# PCR
set.seed(2)
ctrl2 = trainControl(method = "repeatedcv",
                     number = 10,
                     repeats = 5,
                     selectionFunction = "best")

pcr.fit = train(x1, y1,
                method = "pcr",
                tuneGrid = data.frame(ncomp = 1:18),
                trControl = ctrl2,
                preProcess = c("center", "scale"))

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



PLS

```
set.seed(2)

pls.fit = train(x1, y1,
                method = "pls",
                tuneGrid = data.frame(ncomp = 1:18),
                trControl = ctrl2,
                preProcess = c("center", "scale"))

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



GAM

```
num_cores <- detectCores()
cl <- makePSOCKcluster(num_cores)
registerDoParallel(cl)

set.seed(2)
gam.fit = train(x = covid_dat2[rowTrain,],
                y = covid_dat$recovery_time[rowTrain],
                method = "gam",
                tuneGrid = data.frame(method = "GCV.Cp", select = c(TRUE,FALSE)),
                trControl = ctrl1)

gam.fit$bestTune
```

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

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

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

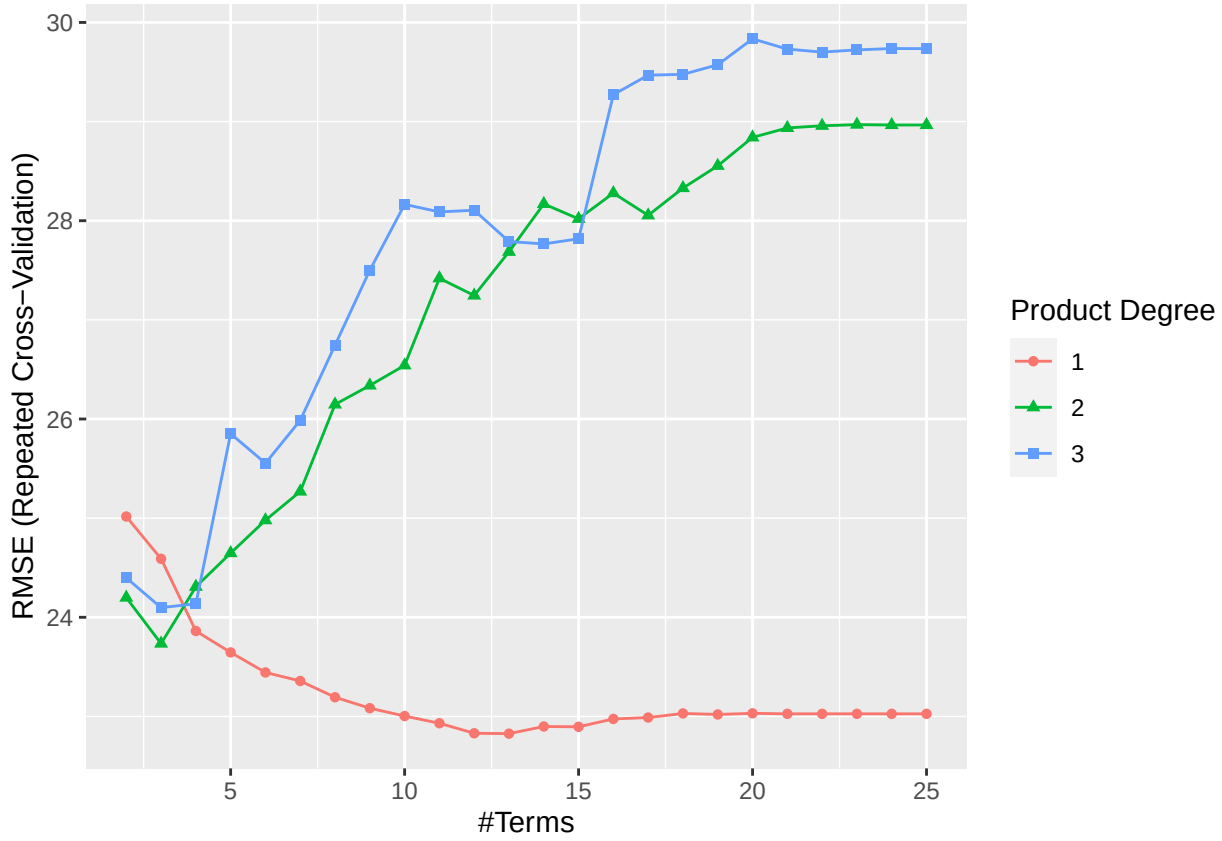
```
##
## Formula:
## .outcome ~ gender1 + race2 + race3 + race4 + smoking1 + smoking2 +
##     hypertension1 + diabetes1 + vaccine1 + severity1 + study2 +
##     study3 + s(age) + s(SBP) + s(LDL) + s(bmi) + s(height) +
##     s(weight)
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   42.70432    1.41505  30.179  < 2e-16 ***
## gender1       -4.95243    0.88819  -5.576 2.73e-08 ***
## race2          0.97140    2.04346   0.475  0.63456
## race3         -0.08402    1.12806  -0.074  0.94064
## race4         -1.71615    1.51650  -1.132  0.25789
## smoking1       4.64567    0.98845   4.700 2.74e-06 ***
## smoking2       7.63539    1.51209   5.050 4.75e-07 ***
## hypertension1  4.05767    0.93489   4.340 1.48e-05 ***
## diabetes1     -1.43590    1.21458  -1.182  0.23723
## vaccine1      -7.49163    0.90263  -8.300  < 2e-16 ***
## severity1      9.08541    1.42832   6.361 2.38e-10 ***
## study2         4.24808    1.16745   3.639  0.00028 ***
## study3         0.08235    1.40632   0.059  0.95331
## ---
## 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)    7.651e-01      9   0.349  0.0428 *
## s(SBP)    4.354e-07      9   0.000  0.8475
## s(LDL)    6.113e-06      9   0.000  0.4023
## s(bmi)    7.872e+00      9 122.480  <2e-16 ***
## s(height) 4.491e+00      9   0.509  0.3127
## s(weight) 4.297e-01      9   0.036  0.2825
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.403   Deviance explained = 40.9%
## GCV = 495.04  Scale est. = 489.8     n = 2513
```

```
stopCluster(cl)
registerDoSEQ()
```

MARS

```
num_cores <- detectCores()
cl <- makePSOCKcluster(num_cores)
registerDoParallel(cl)

set.seed(2)

mars_grid = expand.grid(degree = 1:3,
                        nprune = 2:25)
```

```
mars.fit = train(x = covid_dat2[rowTrain,],
                 y = covid_dat$recovery_time[rowTrain],
                 method = "earth",
                 tuneGrid = mars_grid,
                 trControl = ctrl1)

ggplot(mars.fit)
```




```
mars.fit$bestTune
```

```
##    nprune degree
## 12     13      1
```

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

```
## Call: earth(x=matrix[2513,18], y=c(35,40,35,28,1...), keepxy=TRUE, degree=1,
##             nprune=13)
##
##               coefficients
## (Intercept)     -41.526084
## gender1          -4.942170
## smoking1          4.616207
## smoking2          7.657797
```

```
## hypertension1     4.535753
## vaccine1         -7.436460
## severity1         9.131057
## study2            4.155071
## h(bmi-23.9)      10.051484
## h(bmi-29.5)       7.357588
## h(31.7-bmi)       9.948860
## h(bmi-34.5)      35.195751
##
## Selected 12 of 20 terms, and 8 of 18 predictors (nprune=13)
## Termination condition: RSq changed by less than 0.001 at 20 terms
## Importance: bmi, vaccine1, severity1, gender1, hypertension1, study2, ...
## Number of terms at each degree of interaction: 1 11 (additive model)
## GCV 500.9977    RSS 1236067    GRSq 0.3893063    RSq 0.3999563
```

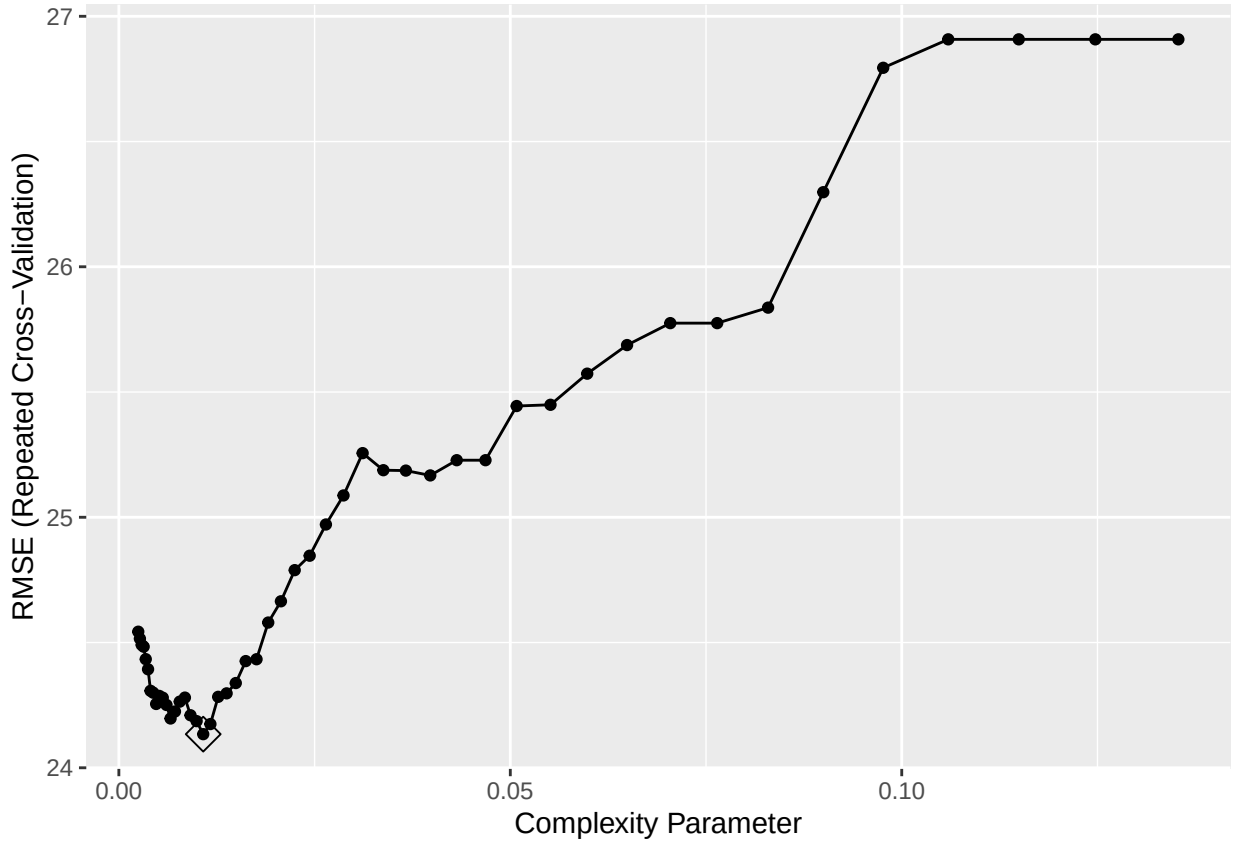
```
stopCluster(cl)
registerDoSEQ()
```

Regression tree

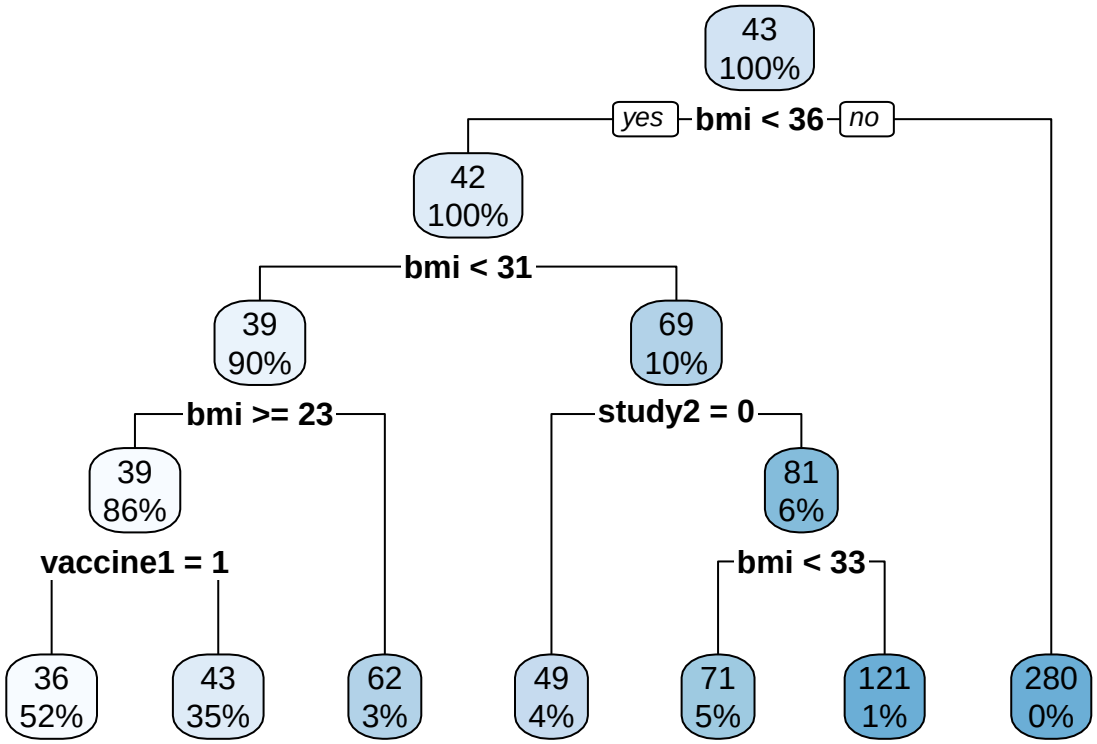
```
num_cores <- detectCores()
cl <- makePSOCKcluster(num_cores)
registerDoParallel(cl)

set.seed(2)

#Build a regression tree on the training data to predict the respons
rpart.fit = train(recovery_time ~ . ,
                  covid_dat[rowTrain,],
                  method = "rpart",
                  tuneGrid = data.frame(cp = exp(seq(-6,-2, length = 50))),
                  trControl =ctrl1)

rpart.fit$bestTune
```

```
##            cp
## 19 0.01077408
```

```
#plot of the tree
ggplot(rpart.fit, highlight = TRUE)
```




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



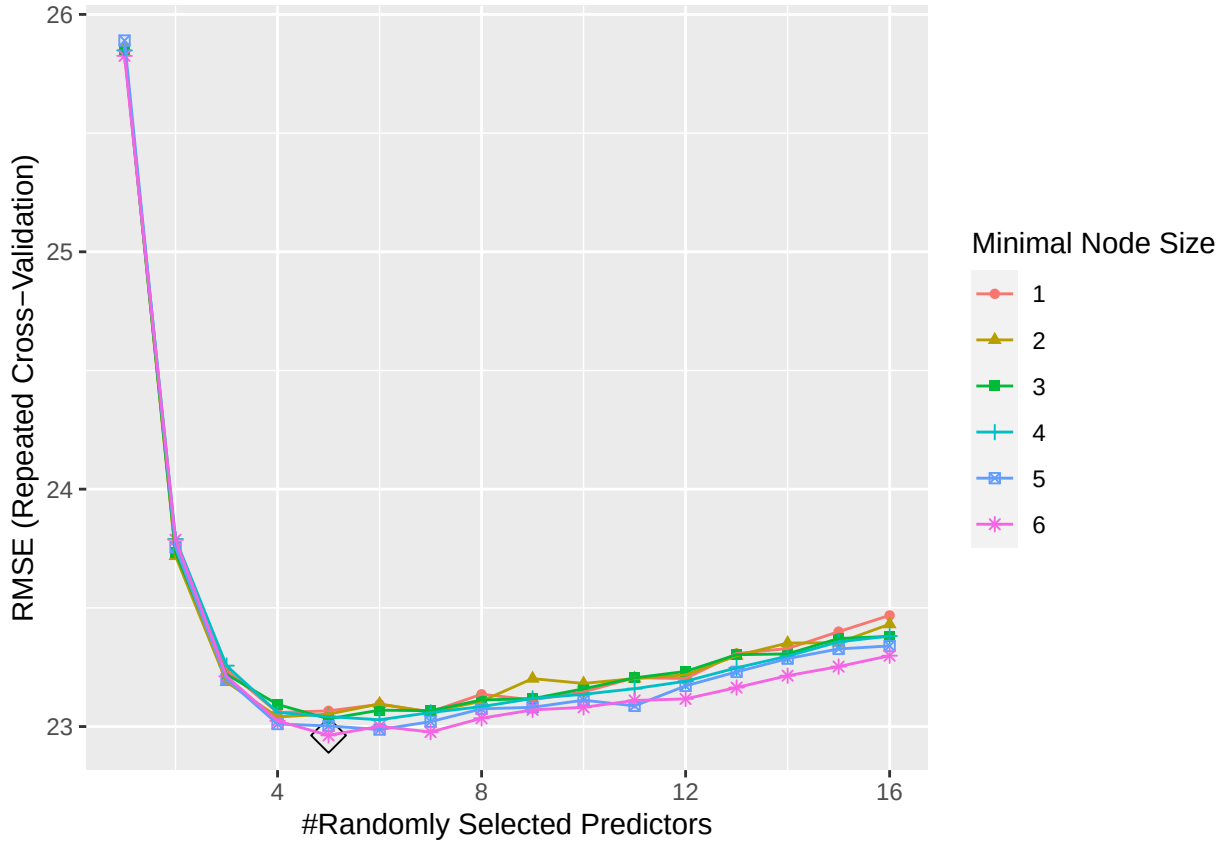

```
stopCluster(cl)
registerDoSEQ()
```

Random Forest

```
num_cores <- detectCores()
cl <- makePSOCKcluster(num_cores)
registerDoParallel(cl)

# Using Caret package to build random forest plot
rf.grid = expand.grid(mtry = 1:16,
                      splitrule = "variance",
                      min.node.size = 1:6)

set.seed(2)
rf.fit = train(recovery_time ~ .,
               covid_dat[rowTrain,],
               method = "ranger",
               tuneGrid = rf.grid,
               trControl = ctrl1)

# Best tunning parameter
rf.fit$bestTune
```

```
##    mtry splitrule min.node.size
## 30    5  variance             6
```

```
ggplot(rf.fit, highlight = TRUE)
```




```
stopCluster(cl)
registerDoSEQ()
```

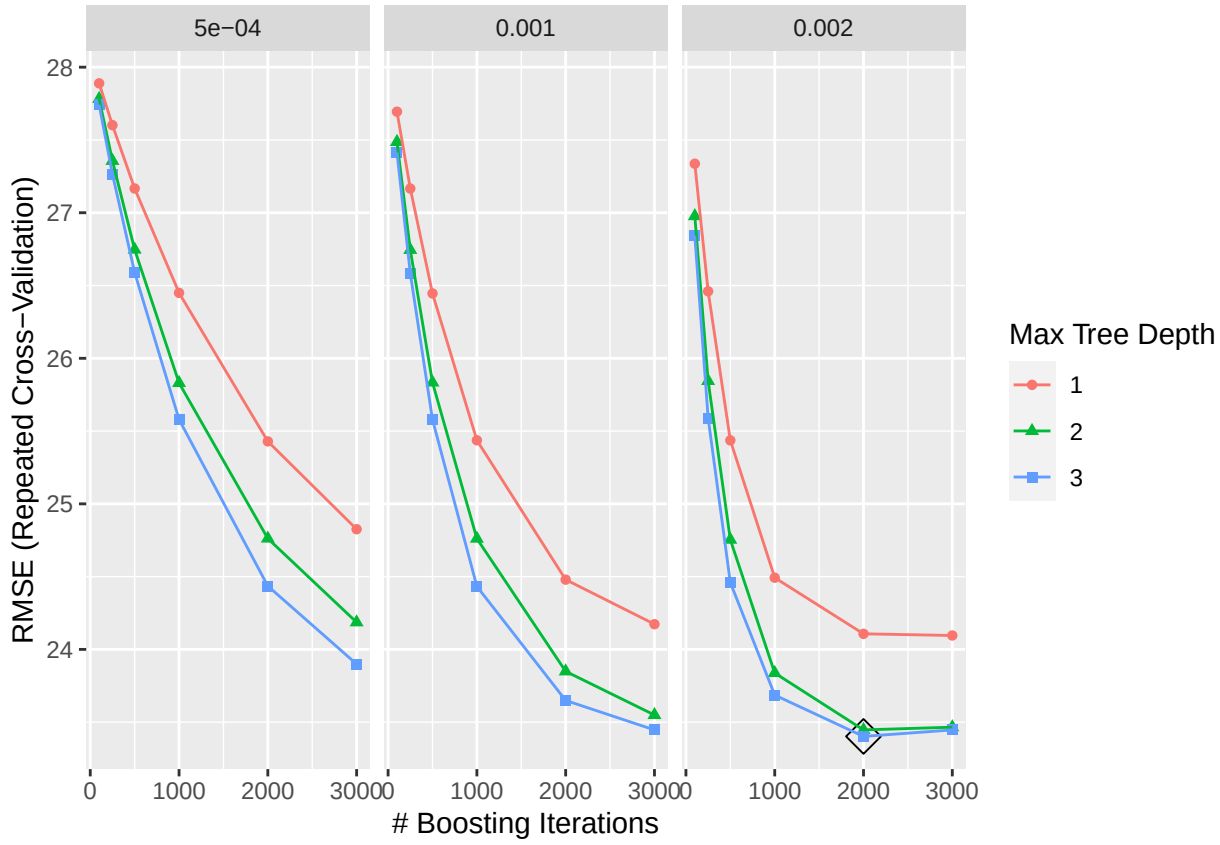
Boosting

```
num_cores <- detectCores()
cl <- makePSOCKcluster(num_cores)
registerDoParallel(cl)

gbm_grid = expand.grid(n.trees = c(100, 250, 500, 1000, 2000, 3000),
                       interaction.depth = 1:3,
                       shrinkage = c(0.0005,0.001,0.002),
                       n.minobsinnode = 1)
set.seed(2)
gbm.fit = train(recovery_time ~ . ,
                covid_dat[rowTrain,],
                tuneGrid = gbm_grid,
                trControl = ctrl1,
```

```
                method = "gbm",
                verbose = FALSE)

gbm.fit$bestTune
```

```
##    n.trees interaction.depth shrinkage n.minobsinnode
## 53    2000                 3     0.002              1
```

```
ggplot(gbm.fit, highlight = TRUE)
```




```
stopCluster(cl)
registerDoSEQ()
```

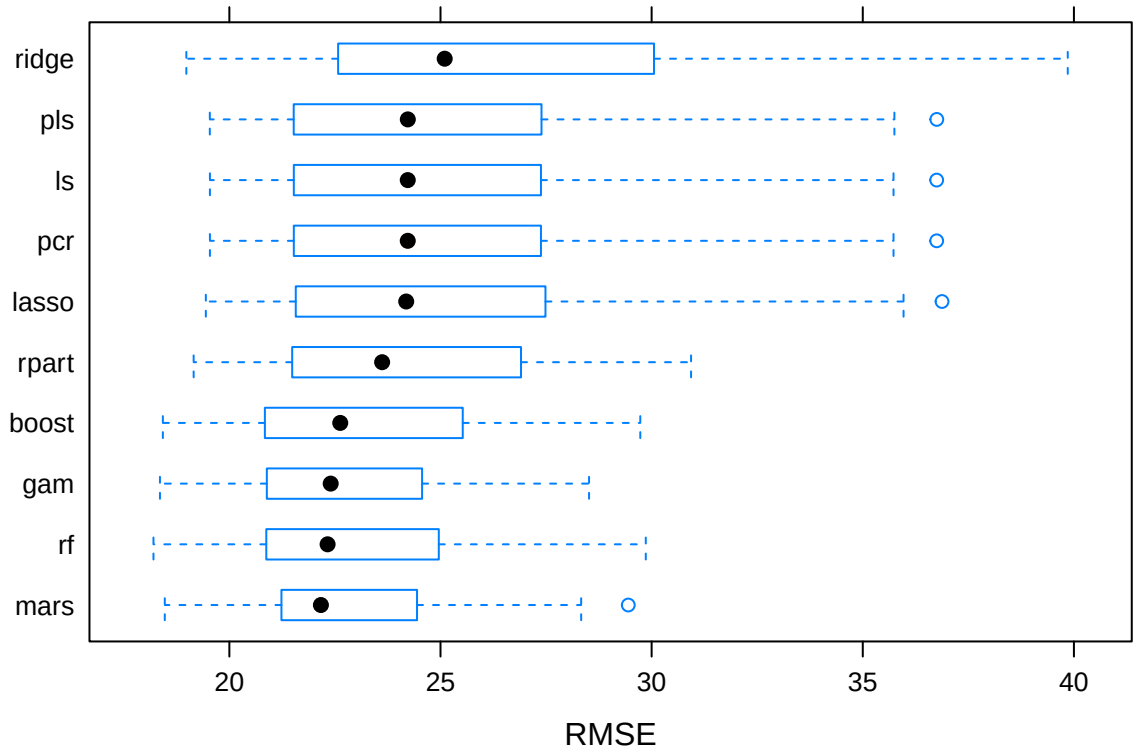
Model selection

```
resamp = resamples(list(ls = ls.fit,
                        ridge = ridge.fit,
                        lasso = lasso.fit,
                        pcr = pcr.fit,
                        pls = pls.fit,
                        gam = gam.fit,
```

```
                         mars = mars.fit,
                         rpart = rpart.fit,
                         rf = rf.fit,
                         boost = gbm.fit))

summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: ls, ridge, lasso, pcr, pls, gam, mars, rpart, rf, boost
## Number of resamples: 50
##
## MAE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## ls    14.64923 15.66708 16.53648 16.41698 17.01352 19.13706    0
## ridge 14.28971 15.62105 16.37738 16.47894 17.03684 19.62712    0
## lasso 14.55464 15.60030 16.47442 16.36440 16.94293 19.10095    0
## pcr   14.64923 15.66708 16.53648 16.41698 17.01352 19.13706    0
## pls   14.64887 15.66889 16.53848 16.41891 17.01693 19.15867    0
## gam   13.72764 14.58034 15.25113 15.31218 15.95440 17.40135    0
## mars  13.73313 14.61445 15.20388 15.23509 15.79857 17.23055    0
## rpart 13.81912 14.75508 15.60355 15.55239 16.30639 17.61039    0
## rf    13.50735 14.38455 14.87164 14.99923 15.62051 17.07063    0
## boost 13.55979 14.25928 15.06306 15.04650 15.61841 17.10484    0
##
## RMSE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## ls    19.53965 21.58444 24.22451 25.09053 27.36552 36.75044    0
## ridge 18.98159 22.60493 25.09909 26.28759 29.81446 39.85442    0
## lasso 19.44279 21.59009 24.18772 25.08628 27.46898 36.87666    0
## pcr   19.53965 21.58444 24.22451 25.09053 27.36552 36.75044    0
## pls   19.53689 21.58001 24.22623 25.08890 27.37554 36.75304    0
## gam   18.35581 20.98344 22.39981 22.82696 24.43688 28.51818    0
## mars  18.47107 21.30049 22.16803 22.82629 24.33608 29.44709    0
## rpart 19.15424 21.54935 23.61790 24.13418 26.86568 30.93387    0
## rf    18.20259 20.88635 22.32617 22.96281 24.83992 29.86165    0
## boost 18.42466 20.97264 22.62550 23.40214 25.43924 29.73134    0
##
## Rsquared
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## ls    0.07061709 0.1559204 0.2038703 0.2160508 0.2666878 0.4437733    0
## ridge 0.03413849 0.0934824 0.1314274 0.1357459 0.1714681 0.2841917    0
## lasso 0.07037256 0.1570906 0.2045110 0.2157923 0.2660000 0.4432293    0
## pcr   0.07061709 0.1559204 0.2038703 0.2160508 0.2666878 0.4437733    0
## pls   0.07059144 0.1558300 0.2046651 0.2161513 0.2672568 0.4439394    0
## gam   0.07942719 0.2172896 0.2971593 0.3480901 0.4974352 0.6569728    0
## mars  0.09689637 0.2206127 0.2995129 0.3442379 0.4819398 0.6649566    0
## rpart 0.02952302 0.1371429 0.2607086 0.2903946 0.4476660 0.6460550    0
## rf    0.07264220 0.2191638 0.2967526 0.3402247 0.4956998 0.6736336    0
## boost 0.02221751 0.1835067 0.2902566 0.3275318 0.4784247 0.6607135    0
```

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



From the summary table above we can observe that MARS model has the best performance (lowest RMSE), which indicates that it appears to be better fitted.

Then make prediction based on MARS model and calculate the test error.

```
mars.fit$bestTune
```

```
##    nprune degree
## 12     13      1
```

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

```
## Call: earth(x=matrix[2513,18], y=c(35,40,35,28,1...), keepxy=TRUE, degree=1,
##             nprune=13)
##
##               coefficients
## (Intercept)     -41.526084
## gender1          -4.942170
## smoking1          4.616207
## smoking2          7.657797
## hypertension1     4.535753
## vaccine1         -7.436460
## severity1         9.131057
```

```
## study2            4.155071
## h(bmi-23.9)      10.051484
## h(bmi-29.5)       7.357588
## h(31.7-bmi)       9.948860
## h(bmi-34.5)      35.195751
##
## Selected 12 of 20 terms, and 8 of 18 predictors (nprune=13)
## Termination condition: RSq changed by less than 0.001 at 20 terms
## Importance: bmi, vaccine1, severity1, gender1, hypertension1, study2, ...
## Number of terms at each degree of interaction: 1 11 (additive model)
## GCV 500.9977    RSS 1236067    GRSq 0.3893063    RSq 0.3999563
```

```
set.seed(2)

mars.pred = predict(mars.fit, newdata = covid_dat2[-rowTrain,])

# Test error
mse.mars = mean((mars.pred - covid_dat$recovery_time[-rowTrain])^2)

mse.mars
```

```
## [1] 505.8689
```

```
rmse.mars = sqrt(mse.mars)
rmse.mars
```

```
## [1] 22.49153
```