Final Primary Analysis

Tianshu Liu, Lincole Jiang, Jiong Ma

Contents

```
library(tidyverse)
library(summarytools)
library(corrplot)
library(caret)
library(vip)
library(rpart.plot)
library(ranger)
```

1 Model Training

1.1 Primary Analysis

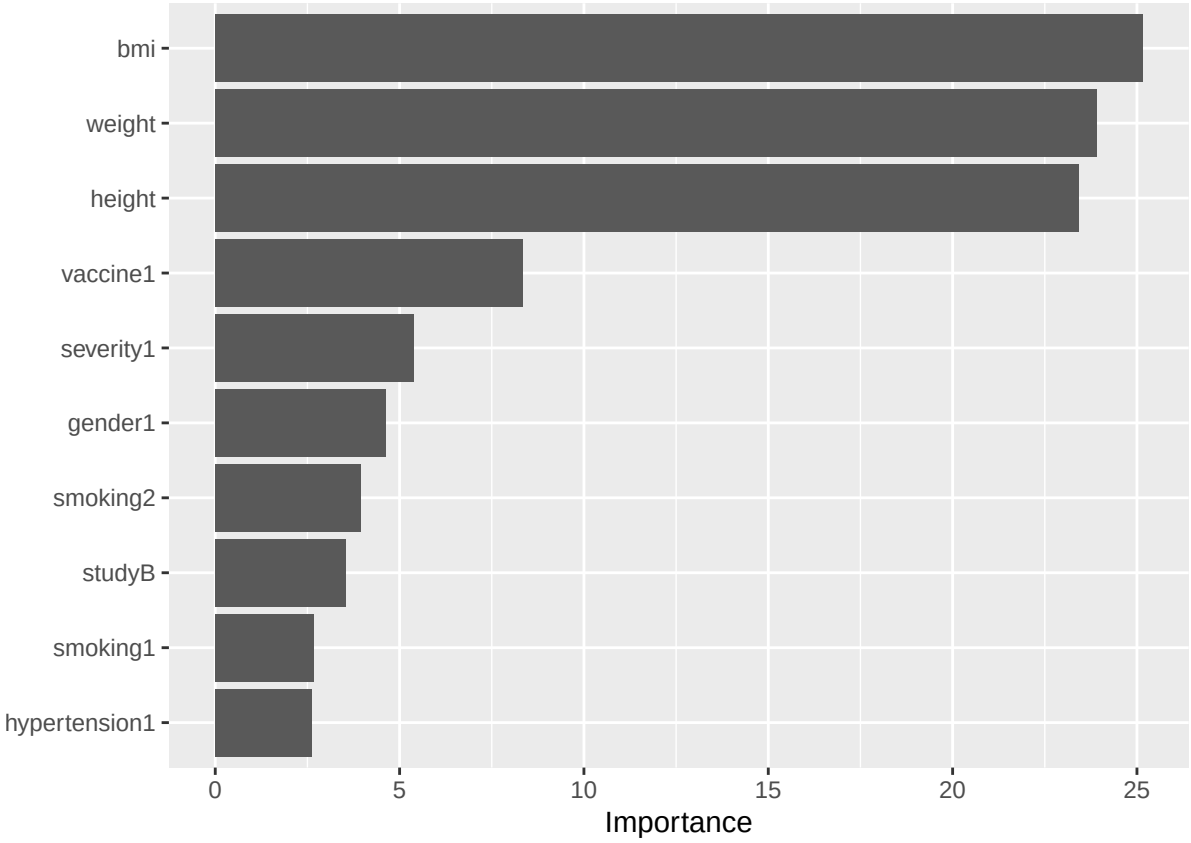
```
ctrl1 <- trainControl(method = "cv", number = 5)
```

1.1.1 Linear Model

```
set.seed(1)

lm.fit <- train(train.x, train.y,
                method = "lm",
                trControl = ctrl1)

coef(lm.fit$finalModel)
```

```
##   (Intercept)           age       gender1         race2         race3
## -3.190120e+03  1.163953e-01 -4.443893e+00  2.189010e+00 -6.599719e-01
##         race4      smoking1      smoking2        height        weight
## -1.156806e+00  2.905693e+00  6.427376e+00  1.866280e+01 -2.014323e+01
##           bmi hypertension1     diabetes1           SBP           LDL
##  6.056969e+01  4.165589e+00 -1.152370e+00 -7.863399e-02 -4.215262e-02
##      vaccine1     severity1        studyB        studyC
## -8.133542e+00  8.747096e+00  4.368587e+00 -6.869681e-01
```

```
vip(lm.fit$finalModel)
```



1.1.2 LASSO

```
set.seed(1)
lasso.fit <- train(train.x, train.y,
                   method = "glmnet",
                   tuneGrid = expand.grid(
                     alpha = 1,
                     lambda = exp(seq(0, -7, length=100))),
                   trControl = ctrl1)

lasso.fit$bestTune
```

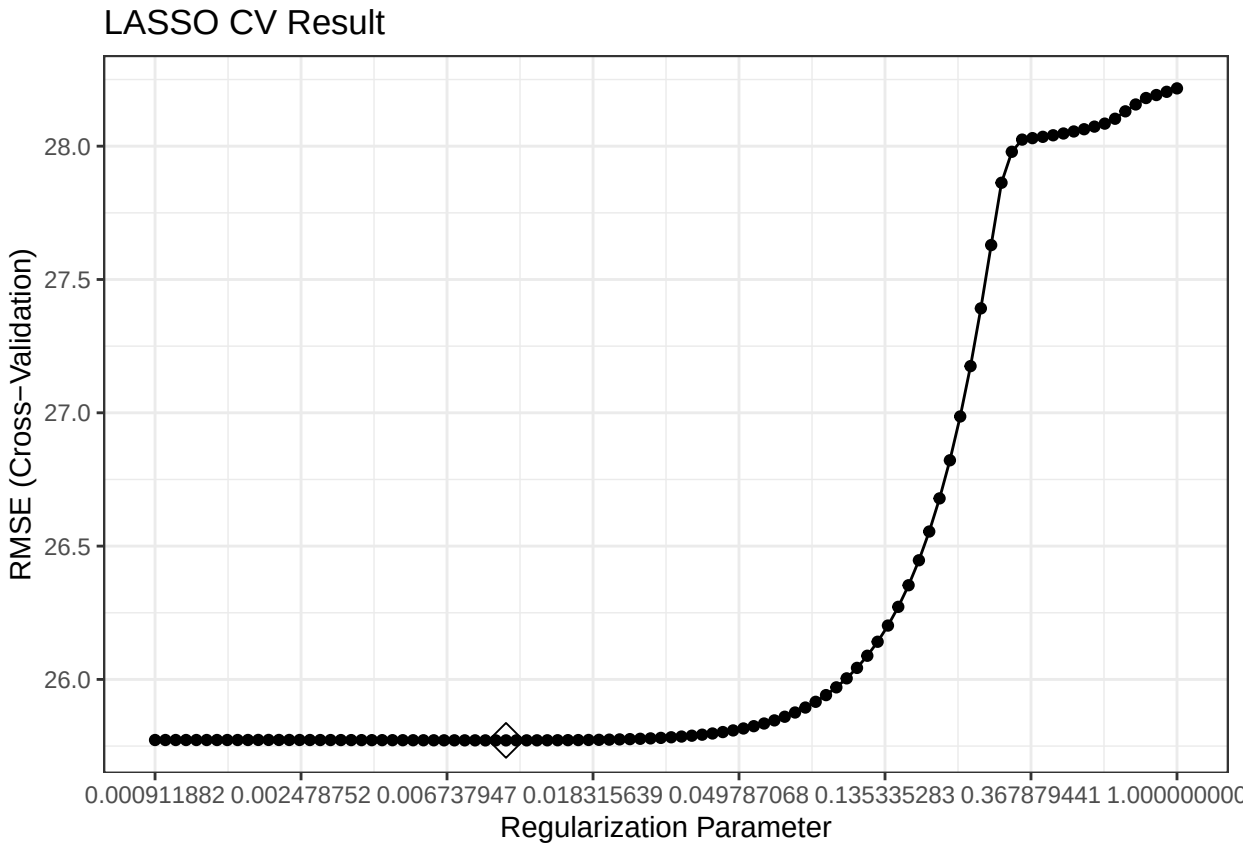
```
##    alpha     lambda
## 35     1 0.01009253
```

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

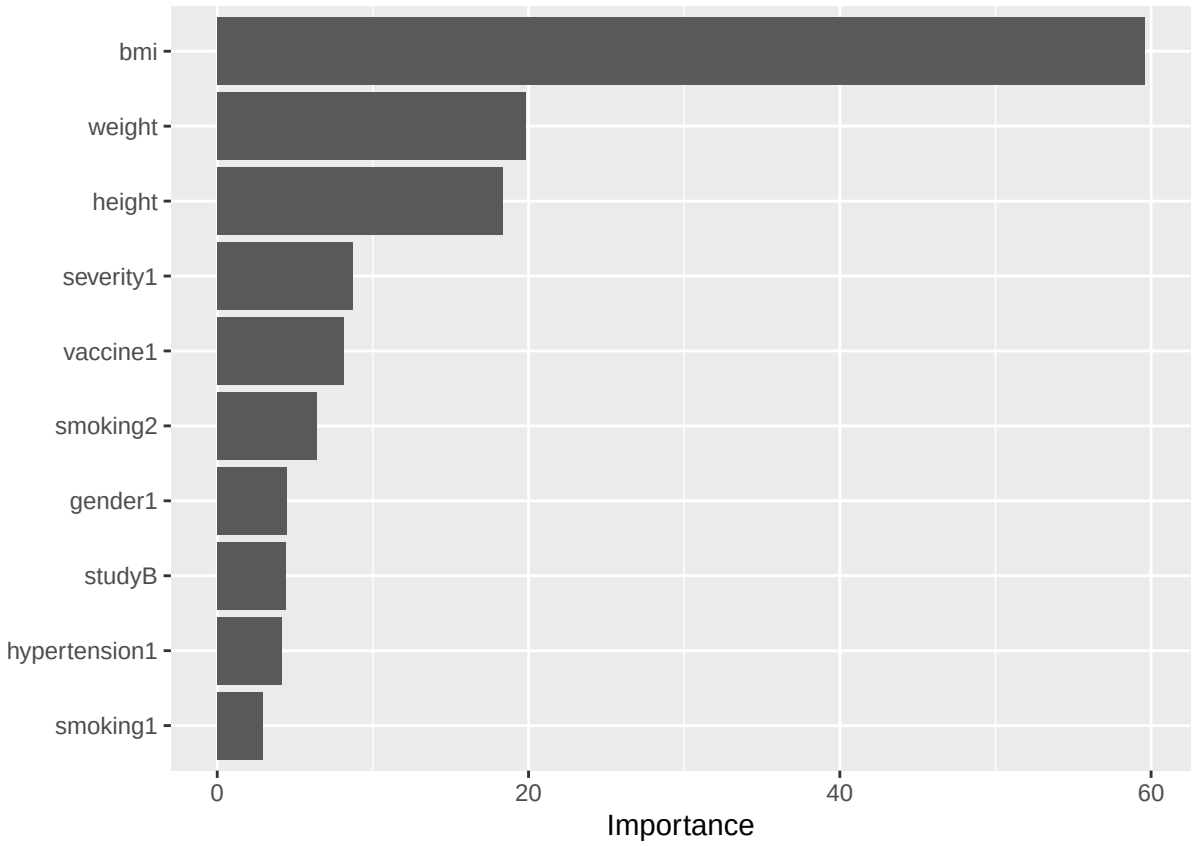
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)   -3.051355e+03
## age            1.115262e-01
## gender1       -4.427839e+00
## race2          2.176007e+00
## race3         -6.665776e-01
## race4         -1.116651e+00
## smoking1       2.878063e+00
## smoking2       6.336796e+00
```

```
## height          1.783946e+01
## weight         -1.927206e+01
## bmi             5.808023e+01
## hypertension1   4.072672e+00
## diabetes1      -1.157773e+00
## SBP            -7.180642e-02
## LDL            -4.176661e-02
## vaccine1       -8.156859e+00
## severity1       8.688928e+00
## studyB          4.363270e+00
## studyC         -6.562541e-01
```

```
ggplot(lasso.fit, highlight = TRUE) +
  labs(title="LASSO CV Result") +
  scale_x_continuous(trans='log',n.breaks = 10) +
  theme_bw()
```




```
ggsave("./figure/lasso_cv.jpeg", dpi = 500)

vip(lasso.fit$finalModel)
```



1.1.3 Ridge

```
set.seed(1)
ridge.fit <- train(train.x, train.y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                                          lambda = exp(seq(1, -5, length=100))),
                   trControl = ctrl1)

ridge.fit$bestTune
```

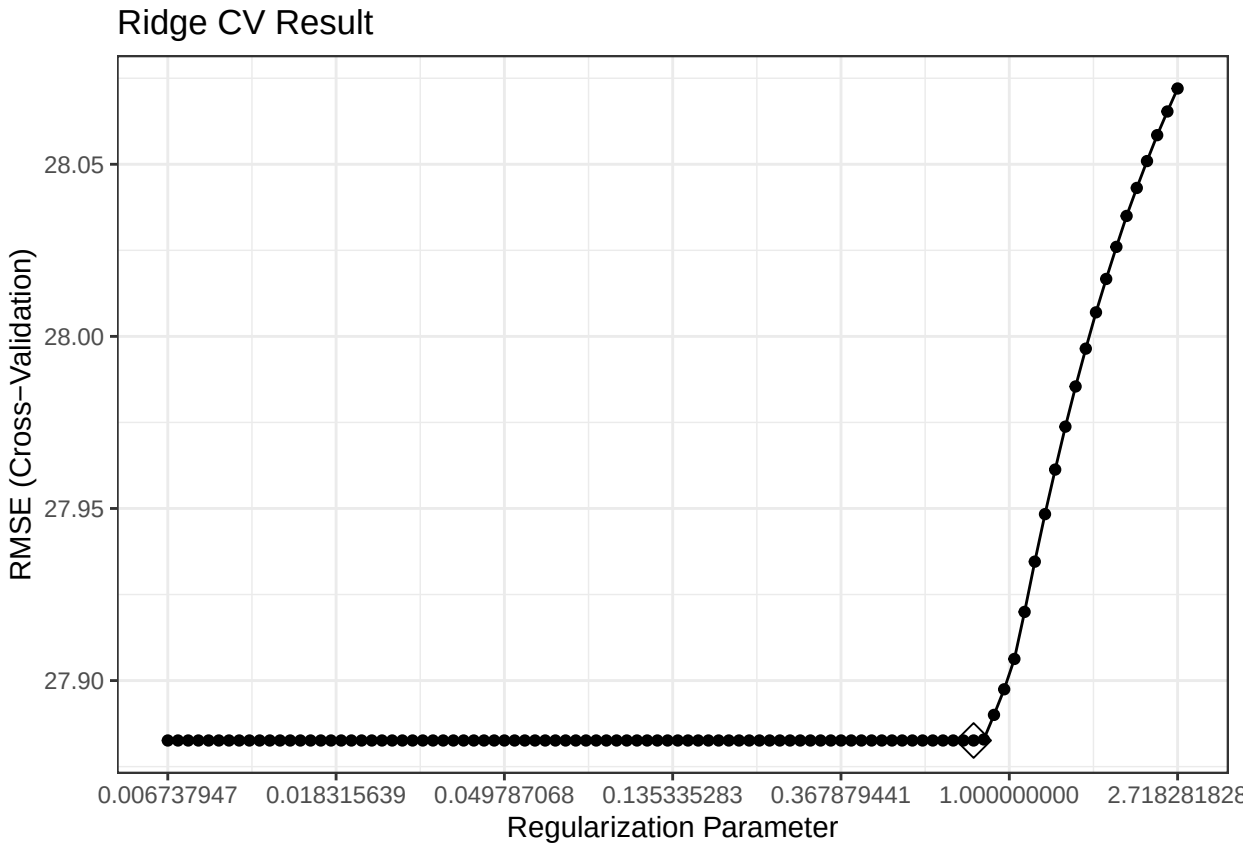
```
##    alpha    lambda
## 80     0 0.8088666
```

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

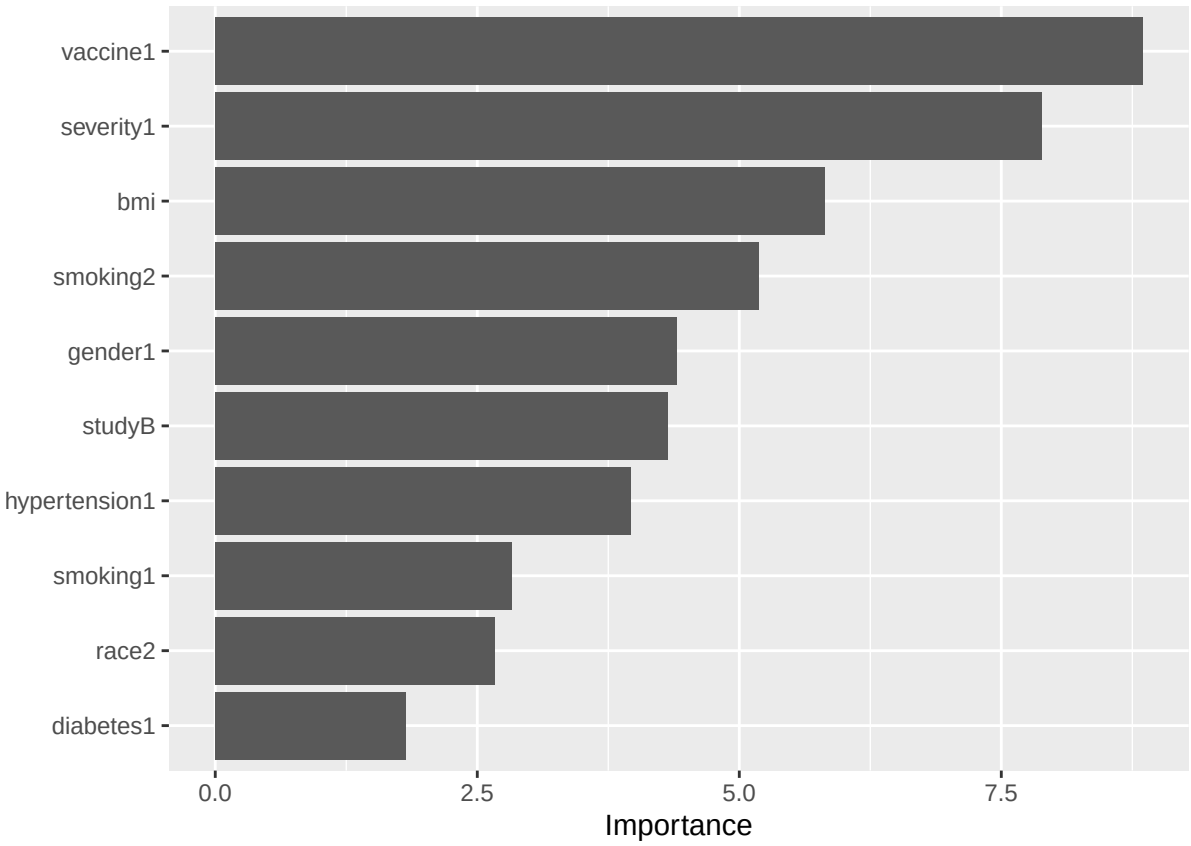
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)   -131.33806374
## age              0.09731228
## gender1         -4.40320528
## race2            2.66527141
## race3           -1.32710400
## race4           -1.12570977
## smoking1         2.82624366
## smoking2         5.18400128
## height           0.60404463
```

```
## weight          -1.01341715
## bmi              5.81922510
## hypertension1    3.96367066
## diabetes1       -1.81677375
## SBP             -0.06303616
## LDL             -0.04440780
## vaccine1        -8.84608080
## severity1        7.88676978
## studyB           4.32156225
## studyC          -0.51357417
```

```
ggplot(ridge.fit,highlight = TRUE) +
  scale_x_continuous(trans='log', n.breaks = 6) +
  labs(title="Ridge CV Result") +
  theme_bw()
```




```
ggsave("./figure/ridge_cv.jpeg", dpi = 500)

vip(ridge.fit$finalModel)
```



1.1.4 Elastic Net

```
set.seed(1)
enet.fit <- train(train.x, train.y,
                  method = "glmnet",
                  tuneGrid = expand.grid(
                    alpha = seq(0, 1, length = 11),
                    lambda = exp(seq(0, -8, length = 50))),
                  trControl = ctrl1)

enet.fit$bestTune
```

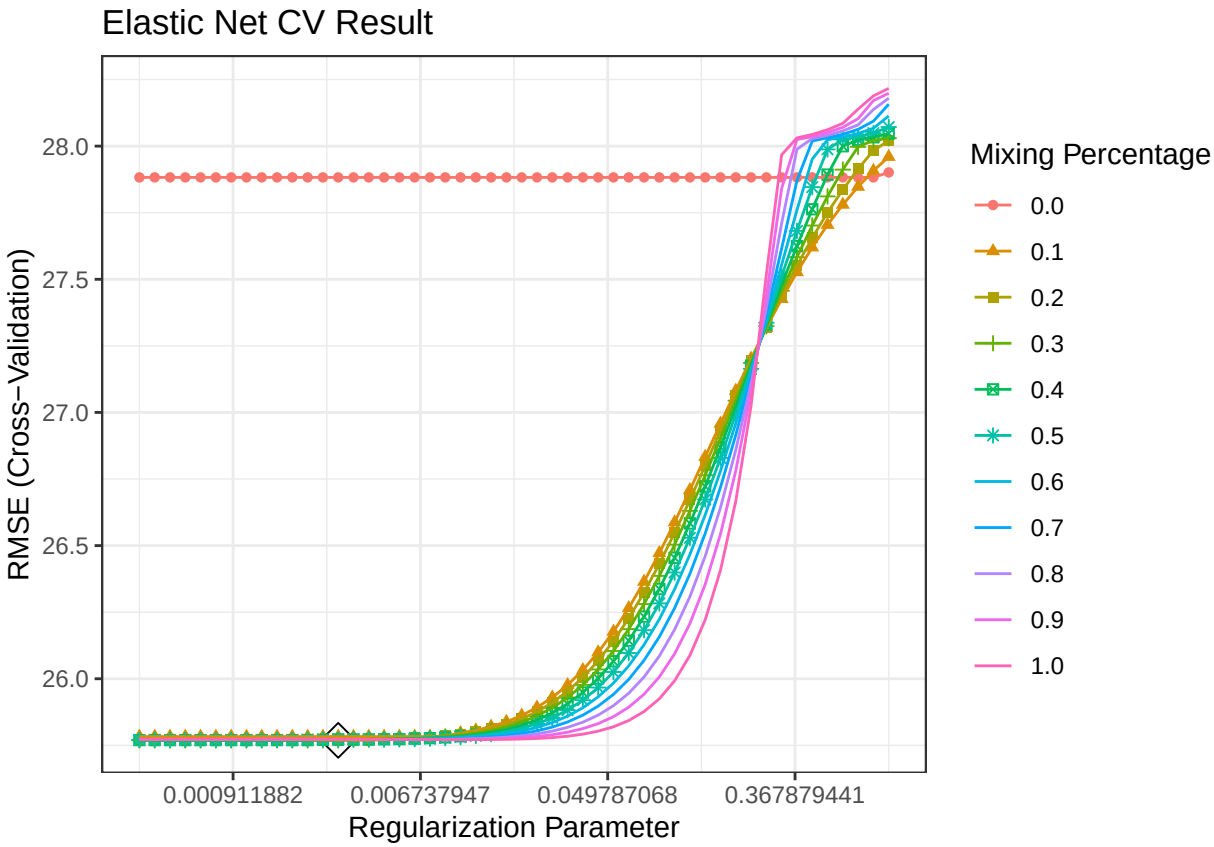
```
##     alpha      lambda
## 164   0.3 0.002801638
```

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

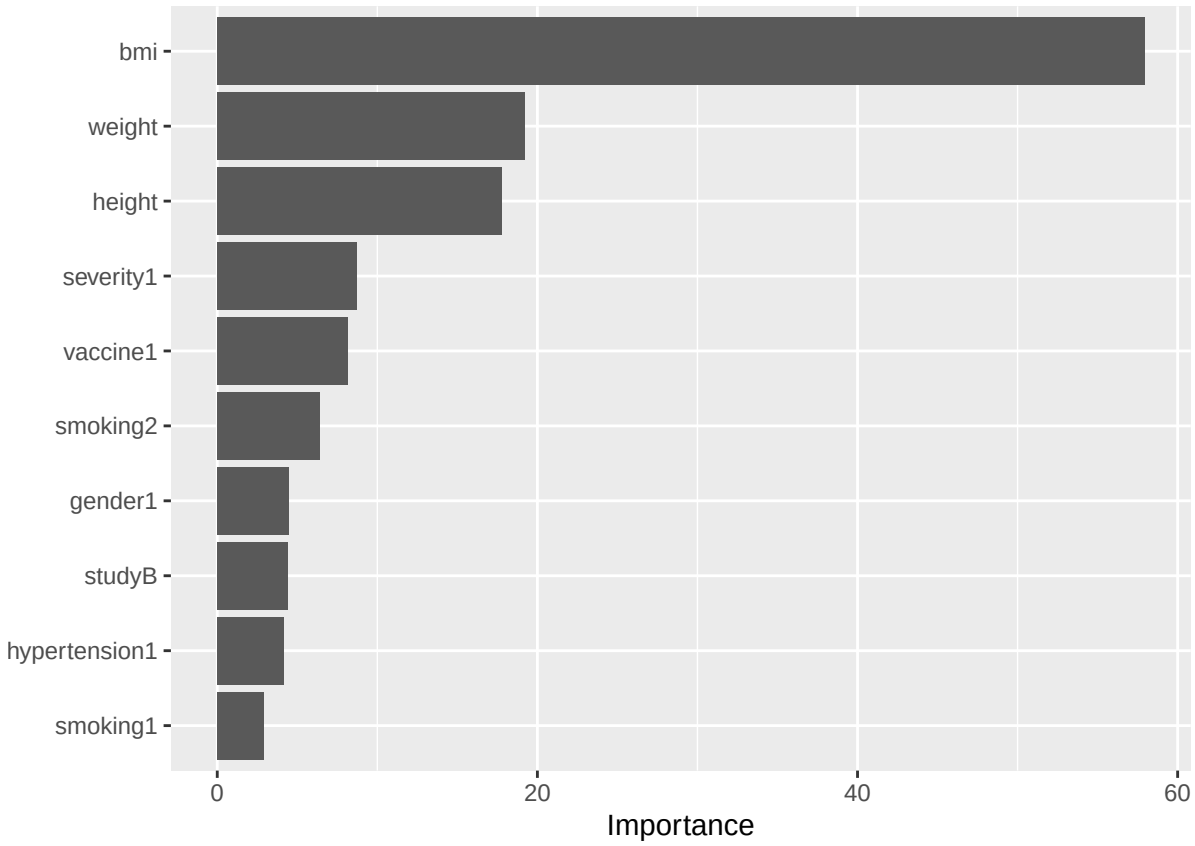
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                               s1
## (Intercept)   -3.041934e+03
## age            1.153916e-01
## gender1       -4.445993e+00
## race2          2.210482e+00
## race3         -6.913699e-01
## race4         -1.153389e+00
## smoking1       2.905138e+00
## smoking2       6.373116e+00
```

```
## height          1.778761e+01
## weight         -1.921664e+01
## bmi             5.792147e+01
## hypertension1   4.164562e+00
## diabetes1      -1.183733e+00
## SBP            -7.825297e-02
## LDL            -4.228081e-02
## vaccine1       -8.179141e+00
## severity1       8.715312e+00
## studyB          4.374718e+00
## studyC         -6.704444e-01
```

```
ggplot(enet.fit, highlight = TRUE) +
  scale_x_continuous(trans='log', n.breaks = 6) +
  labs(title ="Elastic Net CV Result") +
  theme_bw()
```

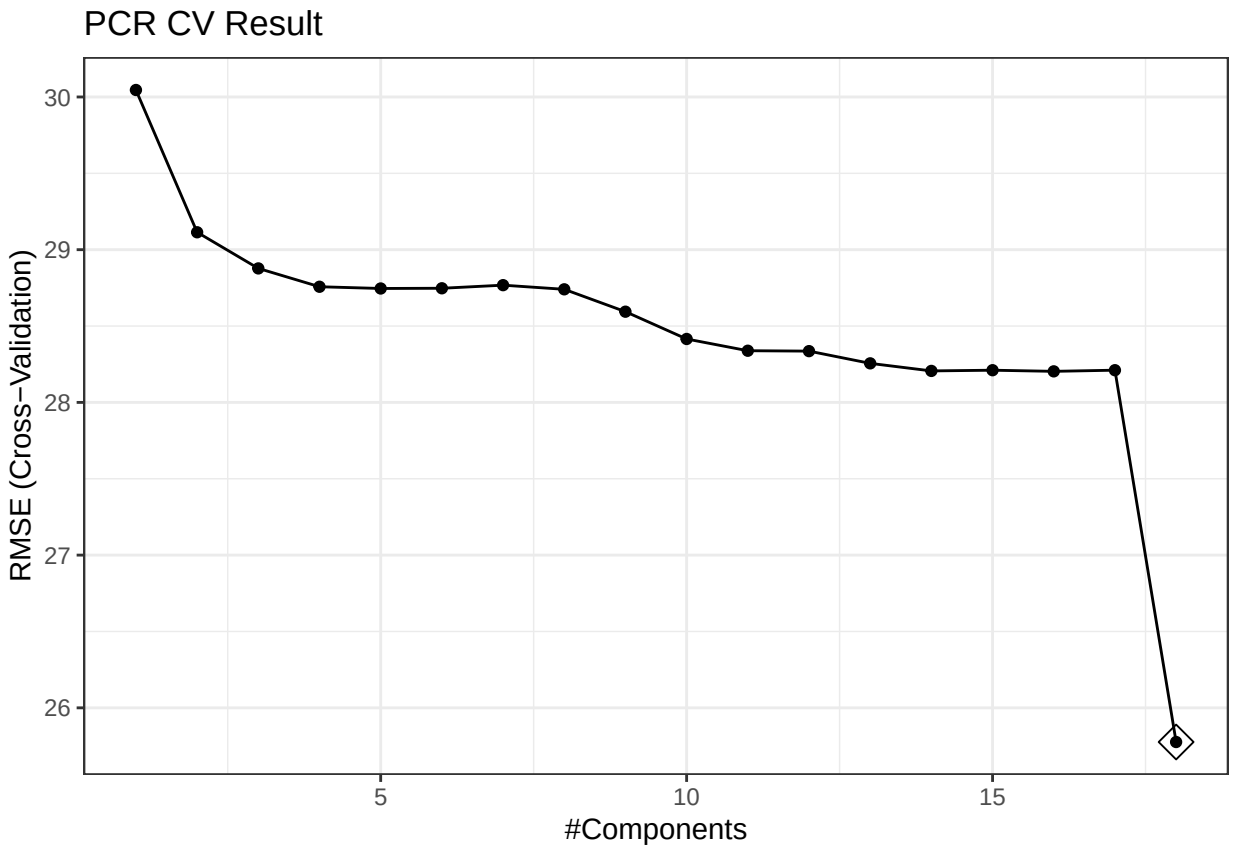



```
ggsave("./figure/enet_cv.jpeg", dpi = 500)

vip(enet.fit$finalModel)
```



1.1.5 Principal components regression (PCR)

```
set.seed(1)
pcr.fit <- train(train.x,
                 train.y,
                 method = "pcr",
                 tuneGrid  = data.frame(ncomp = 1:ncol(train.x)),
                 trControl = ctrl1,
                 preProcess = c("center", "scale"))
ggplot(pcr.fit, highlight = TRUE) +
  labs(title  ="PCR CV Result") +
  theme_bw()
```




```
ggsave("./figure/pcr_cv.jpeg", dpi = 500)

pcr.fit$bestTune
```

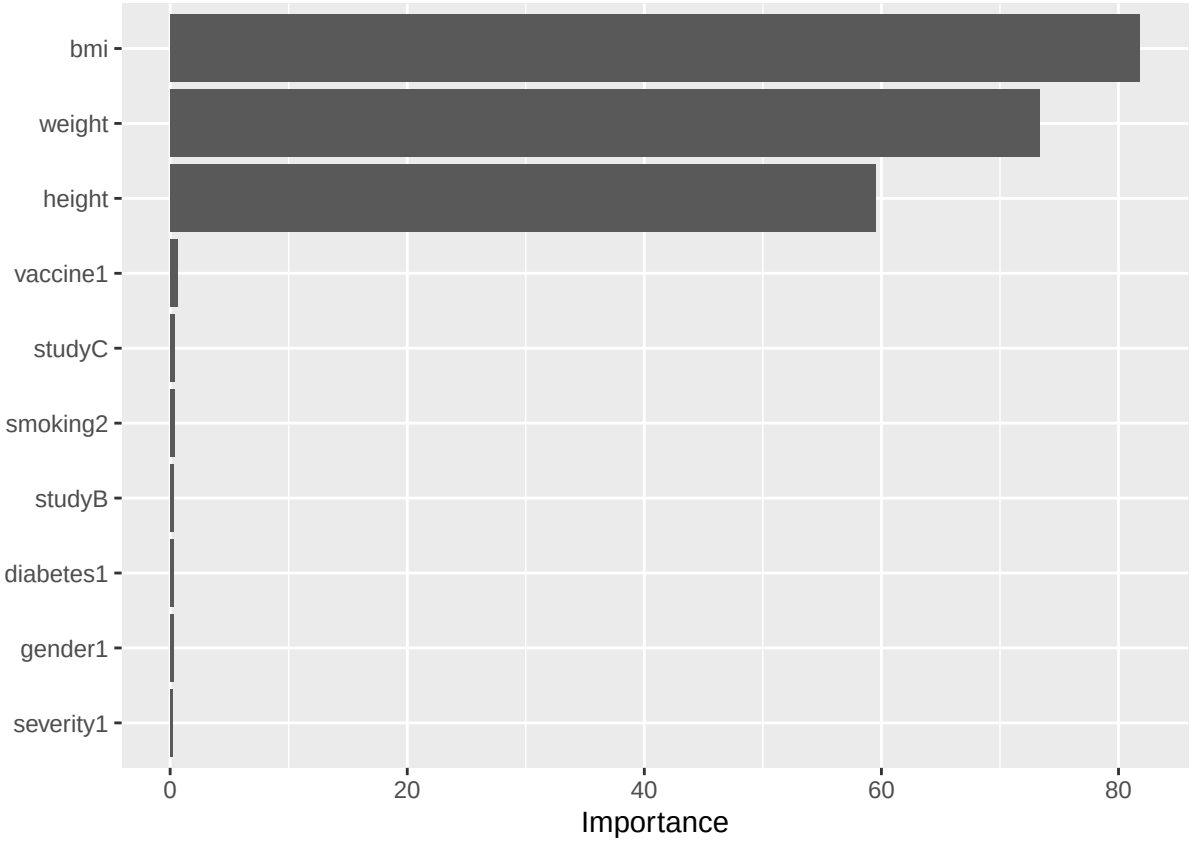
```
##    ncomp
## 18    18
```

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

```
## , , 18 comps
##
##                    .outcome
## age               0.5252538
## gender1          -2.2221586
## race2             0.4563464
## race3            -0.2619635
## race4            -0.3476329
## smoking1          1.3205684
## smoking2          1.9344423
## height          112.6936931
## weight         -141.0001175
## bmi             165.1518985
## hypertension1     2.0811234
## diabetes1        -0.4188178
## SBP              -0.6356938
## LDL              -0.8376686
## vaccine1         -4.0025673
## severity1         2.5879846
```

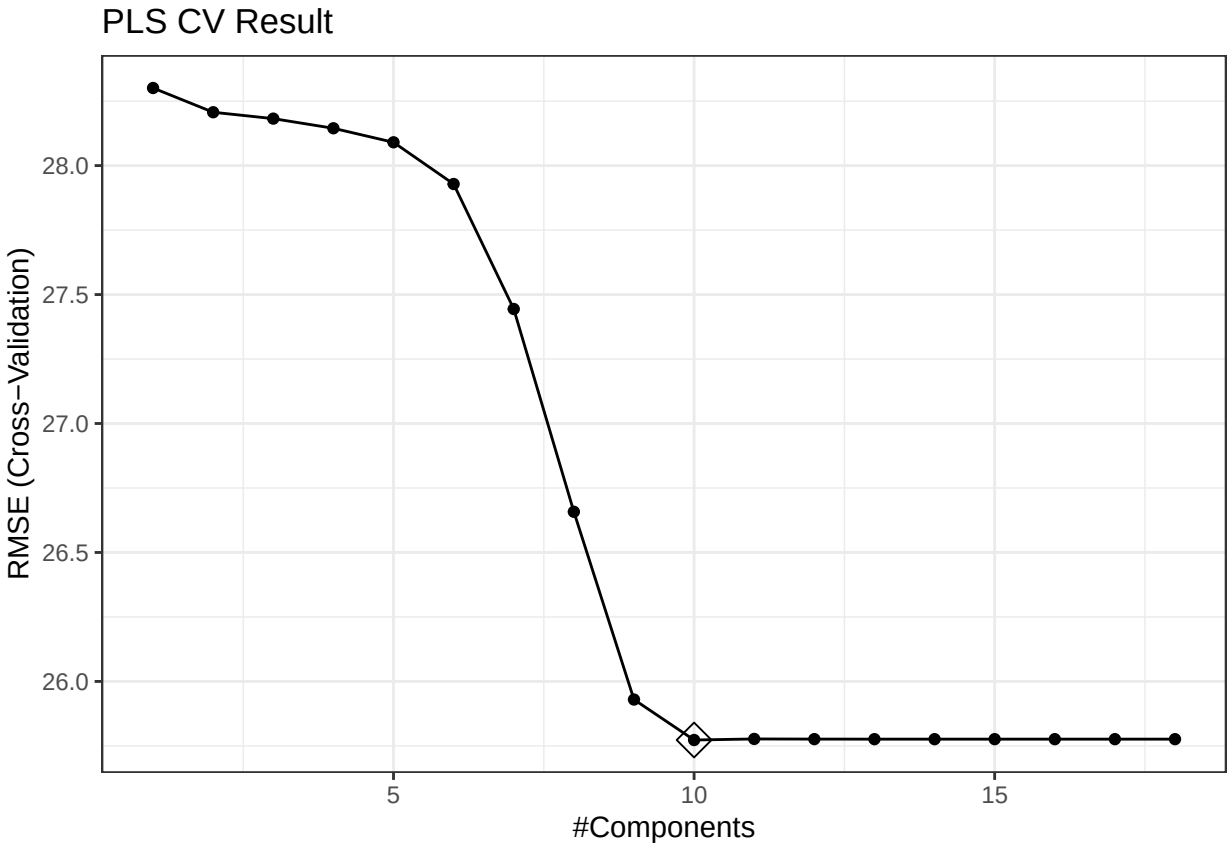
```
## studyB            2.1374000
## studyC           -0.2730416
```

```
vip(pcr.fit$finalModel)
```



1.1.6 Partial Least Squares (PLS)

```
set.seed(1)
pls.fit <- train(train.x,
                 train.y,
                 method = "pls",
                 tuneGrid = data.frame(ncomp = 1:ncol(train.x)),
                 trControl = ctrl1,
                 preProcess = c("center", "scale"))
ggplot(pls.fit, highlight = TRUE) +
  labs(title  ="PLS CV Result") +
  theme_bw()
```




```
ggsave("./figure/pls_cv.jpeg", dpi = 500)

pls.fit$bestTune
```

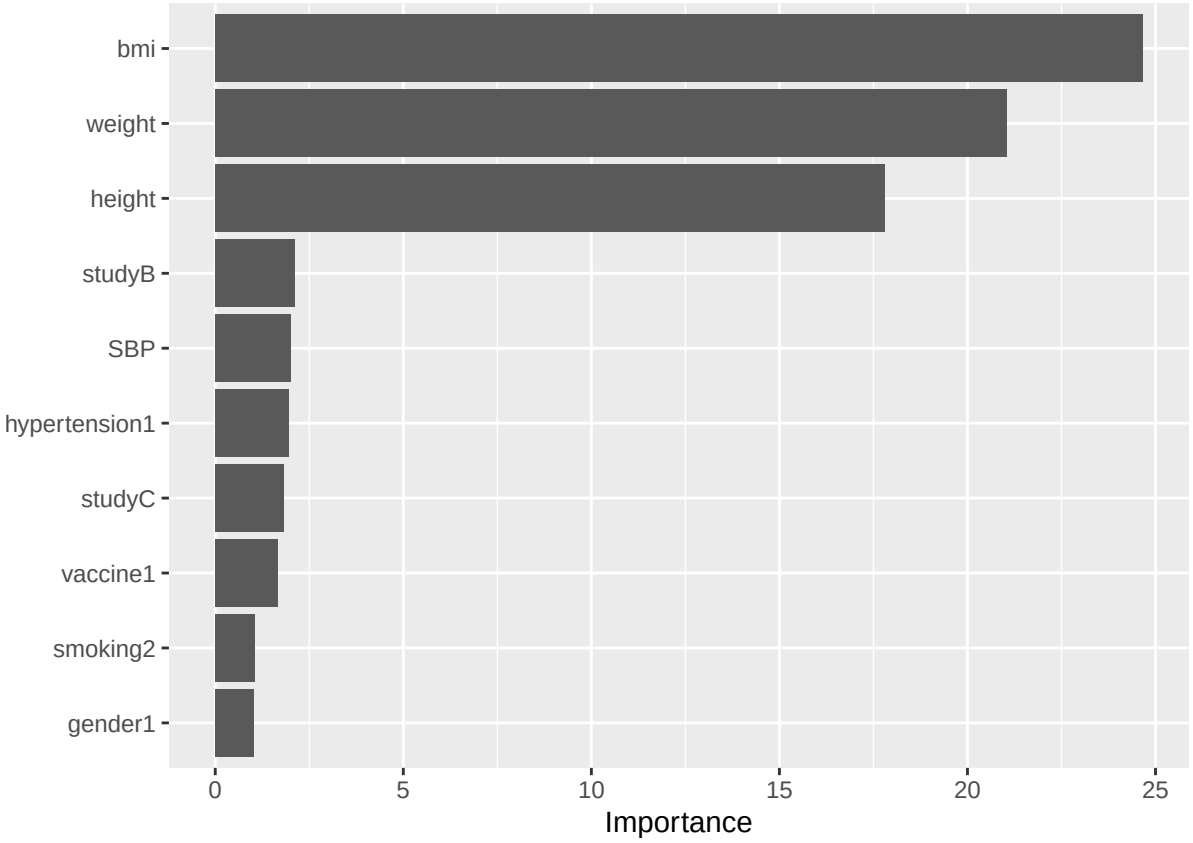
```
##    ncomp
## 10    10
```

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

```
## , , 10 comps
##
##                    .outcome
## age               0.2557655
## gender1          -2.2288535
## race2             0.4448161
## race3            -0.1161982
## race4            -0.4146309
## smoking1          1.3626057
## smoking2          1.8793373
## height          112.5738962
## weight         -140.9034395
## bmi             165.0235495
## hypertension1     2.1927025
## diabetes1        -0.4588377
## SBP              -0.6092501
## LDL              -0.7129796
## vaccine1         -4.0284909
## severity1         2.5664367
```

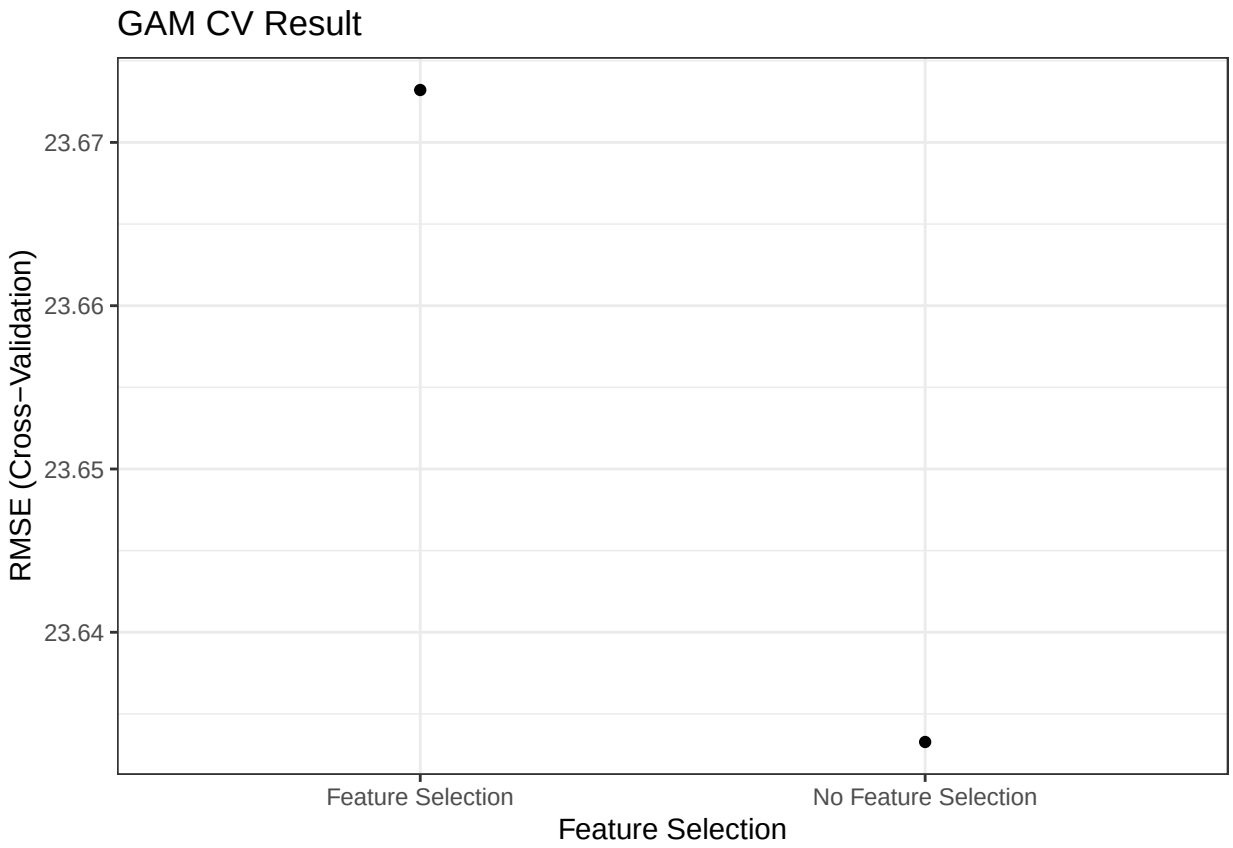
```
## studyB            2.1234056
## studyC           -0.2961257
```

```
vip(pls.fit$finalModel)
```



1.1.7 Generalized Additive Model (GAM)

```
set.seed(1)
gam.fit <- train(train.x,
                 train.y,
                 method = "gam",
                 tuneGrid = data.frame(select = c(TRUE, FALSE),
                                       method = "GCV.Cp"),
                 trControl = ctrl1)


ggplot(gam.fit) +
  labs(title = "GAM CV Result") +
  theme_bw()
```



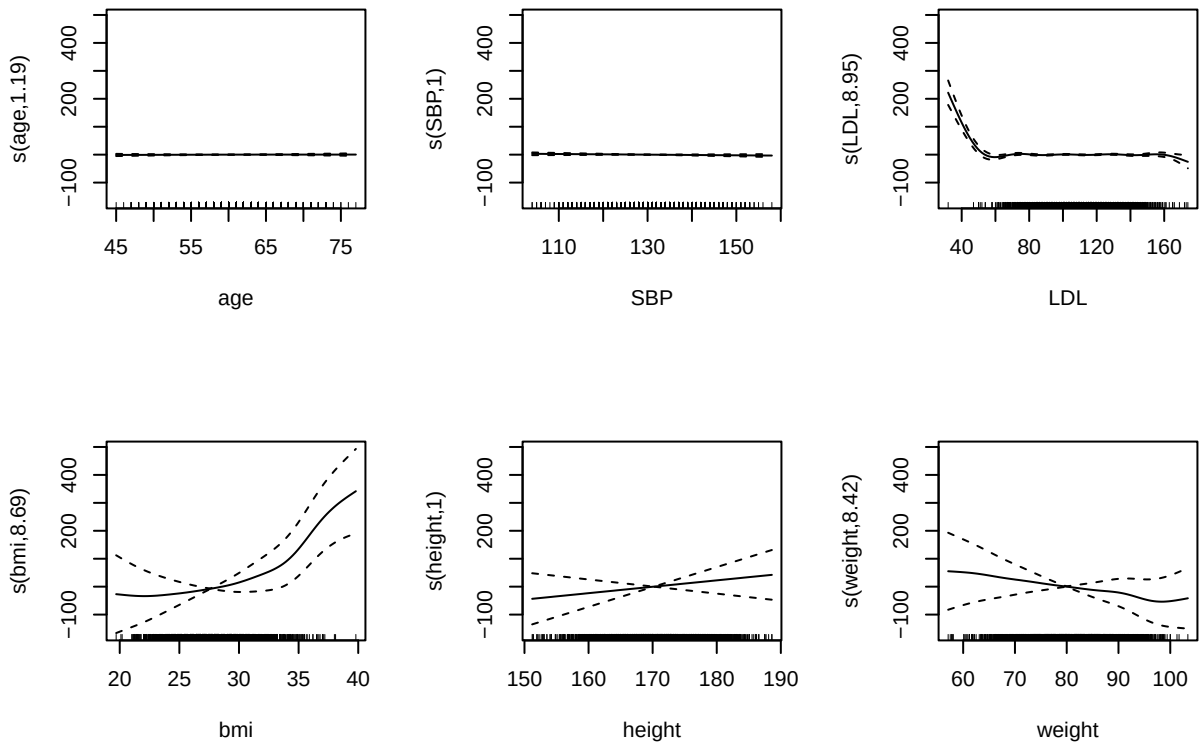

```
ggsave("./figure/gam_cv.jpeg", dpi = 500)

gam.fit$bestTune
```

```
##   select method
## 1  FALSE GCV.Cp
```

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

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender1 + race3 + race4 + smoking1 + smoking2 + hypertension1 +
##     diabetes1 + vaccine1 + severity1 + studyB + studyC + s(age) +
##     s(SBP) + s(LDL) + s(bmi) + s(height) + s(weight)
##
## Estimated degrees of freedom:
## 1.19 1.00 8.95 8.69 1.00 8.42  total = 41.24
##
## GCV score: 524.5768
```

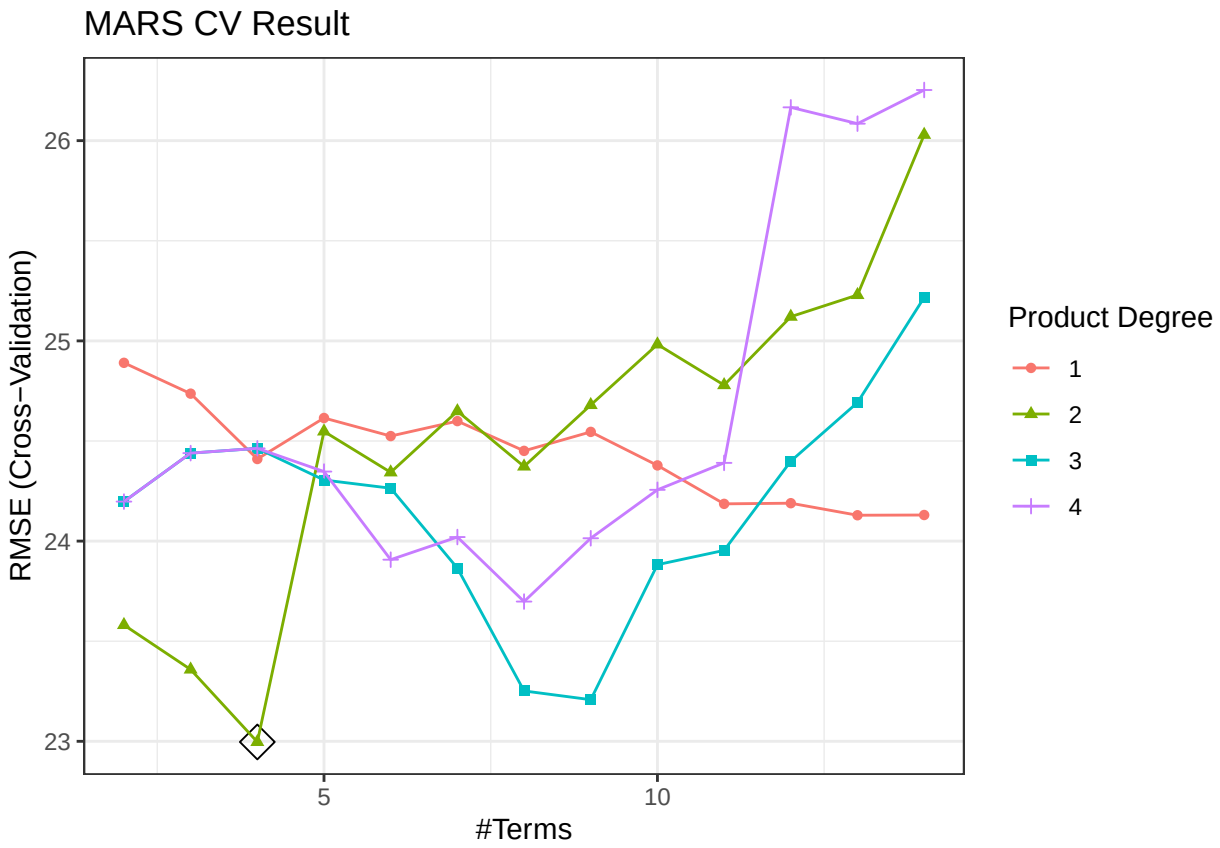
```
par(mfrow=c(2, 3))
plot(gam.fit$finalModel)
```




```
par(mfrow=c(1, 1))
```

1.1.8 Multivariate Adaptive Regression Splines (MARS)

```
mars.grid <- expand.grid(degree = 1:4,
                         nprune = 2:14)
set.seed(1)
mars.fit <- train(train.x,
                  train.y,
                  method = "earth",
                  tuneGrid = mars.grid,
                  trControl = ctrl1)

ggplot(mars.fit, highlight = TRUE)+
  labs(title  ="MARS CV Result") +
  theme_bw()
```




```
ggsave("./figure/mars_cv.jpeg", dpi = 500)

mars.fit$bestTune
```

```
##    nprune degree
## 16      4      2
```

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

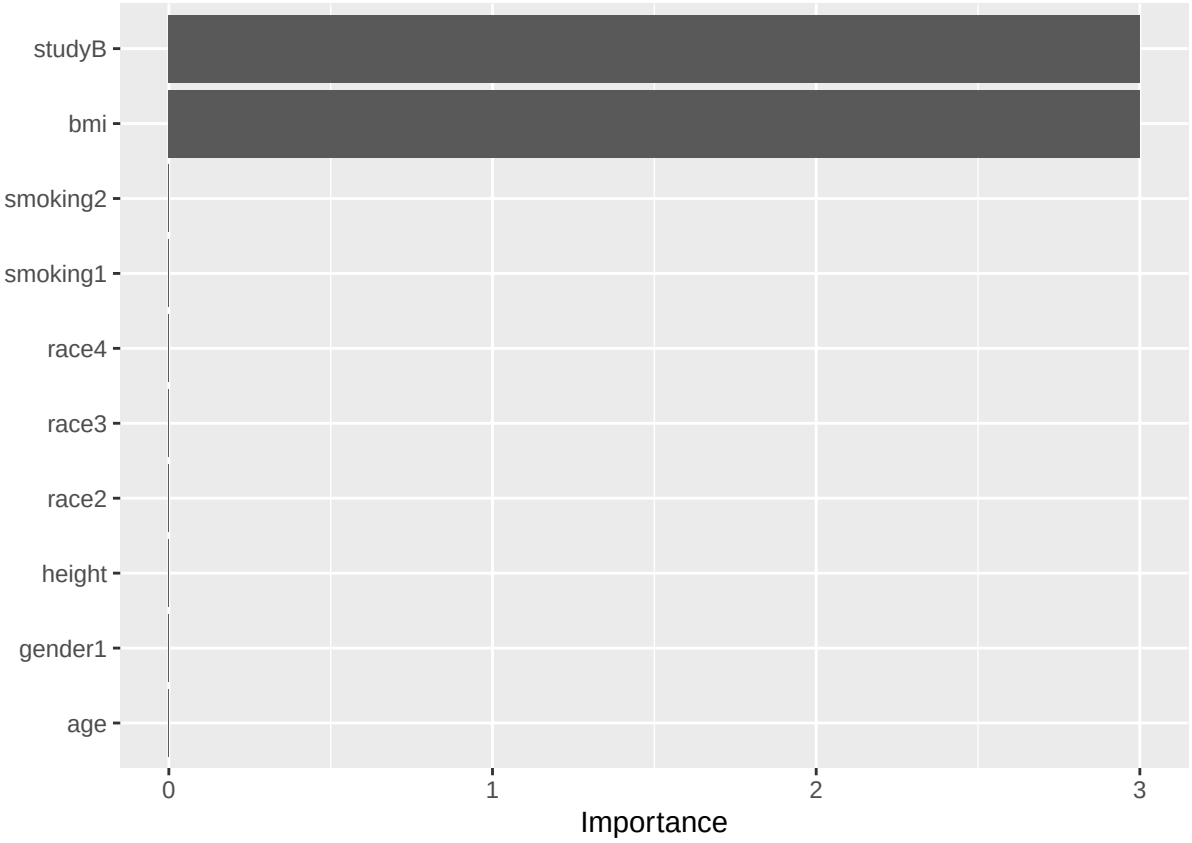
```
##          (Intercept)          h(31.7-bmi) h(bmi-31.7) * studyB
##            14.402672            3.762816           34.486367
##          h(bmi-26.8)
##             6.780526
```

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

```
## Call: earth(x=matrix[2900,18], y=c(40,34,31,50,3...), keepxy=TRUE, degree=2,
##             nprune=4)
##
##                       coefficients
## (Intercept)              14.402672
## h(bmi-26.8)               6.780526
## h(31.7-bmi)               3.762816
## h(bmi-31.7) * studyB     34.486367
##
## Selected 4 of 25 terms, and 2 of 18 predictors (nprune=4)
## Termination condition: Reached nk 37
## Importance: bmi, studyB, age-unused, gender1-unused, race2-unused, ...
## Number of terms at each degree of interaction: 1 2 1
```

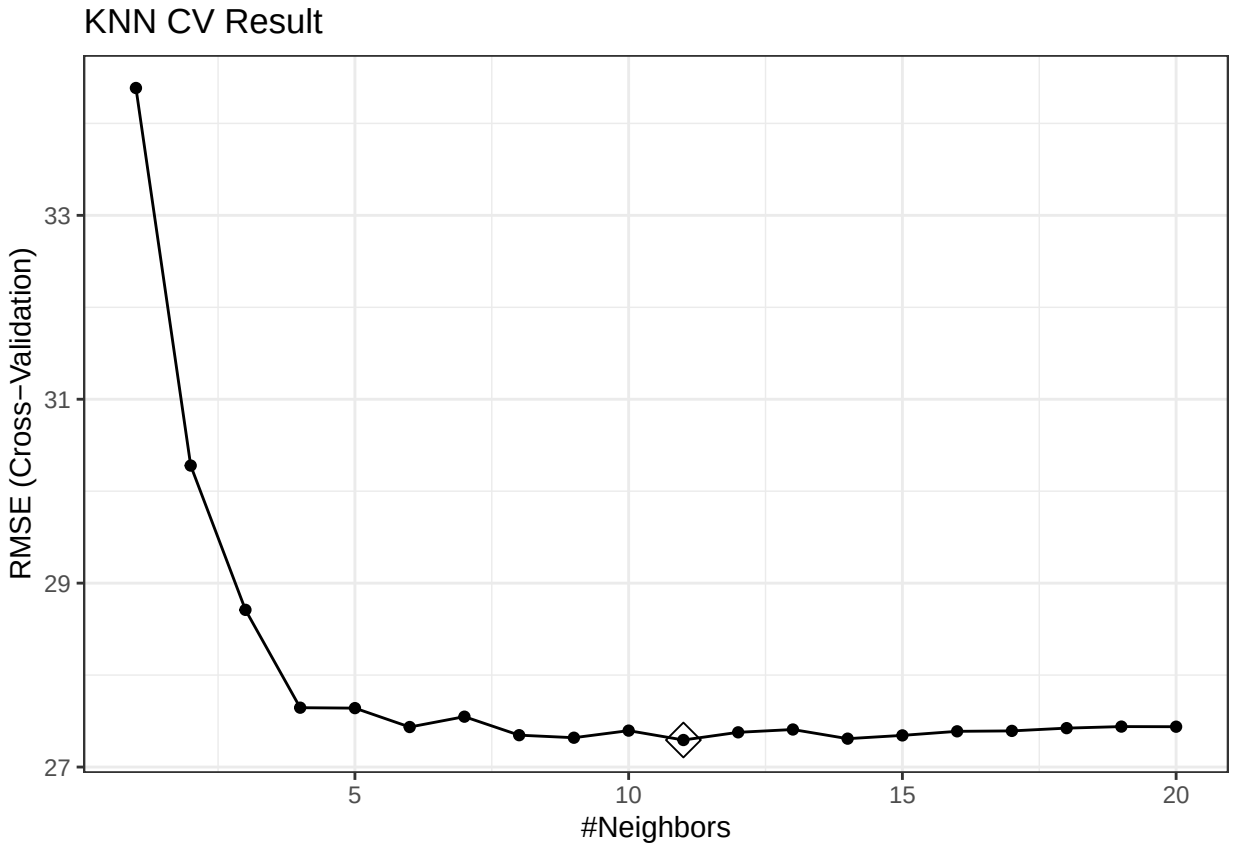
```
## GCV 505.0777    RSS 1456152    GRSq 0.4574307    RSq 0.4602344
```

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



1.1.9 K-Nearest Neighbour (KNN)

```
set.seed(1)
knn.fit <- train(train.x,
                 train.y,
                 tuneGrid  = data.frame(k = 1:20),
                 method = "knn",
                 trControl = ctrl1)

ggplot(knn.fit, highlight = TRUE) +
  labs(title  ="KNN CV Result") +
  theme_bw()
```




```
ggsave("./figure/knn_cv.jpeg", dpi = 500)

knn.fit$bestTune
```

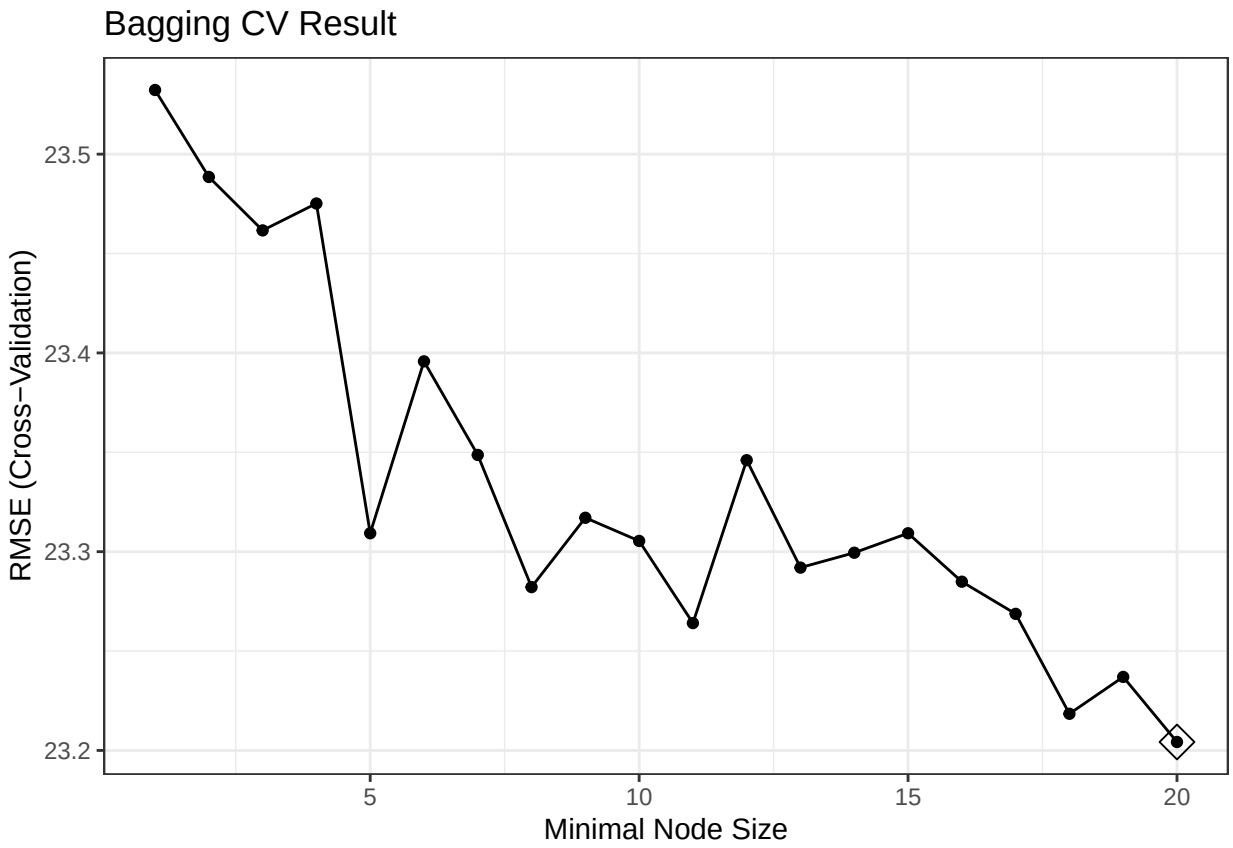
```
##     k
## 11 11
```

1.1.10 Bagging

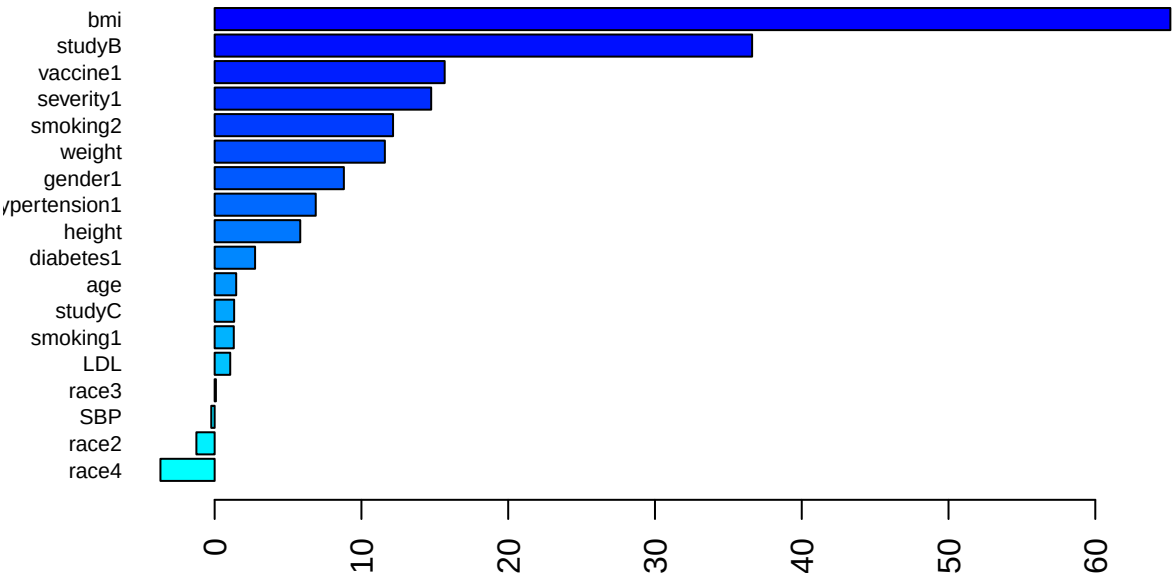
```
bag.grid <- expand.grid(mtry = ncol(train.x),
                        splitrule = "variance",
                        min.node.size = 1:20)

set.seed(1)
bag.fit <- train(train.x,
                 train.y,
                 method = "ranger",
                 tuneGrid = bag.grid,
                 trControl = ctrl1)

bag.fit$bestTune
```

```
##    mtry splitrule min.node.size
## 20   18  variance            20
```

```
ggplot(bag.fit, highlight = TRUE) +
  labs(title = "Bagging CV Result") +
  theme_bw()
```




```
ggsave("./figure/bagging_cv.jpeg", dpi = 500)

bag.final.per <- ranger(recovery_time ~ .,
                        data = train.dat.matrix,
                        mtry = ncol(train.x),
                       splitrule = "variance",
                       min.node.size = bag.fit$bestTune[[3]],
                       importance = "permutation",
                       scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(bag.final.per),
             decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(ncol(train.x)))
```



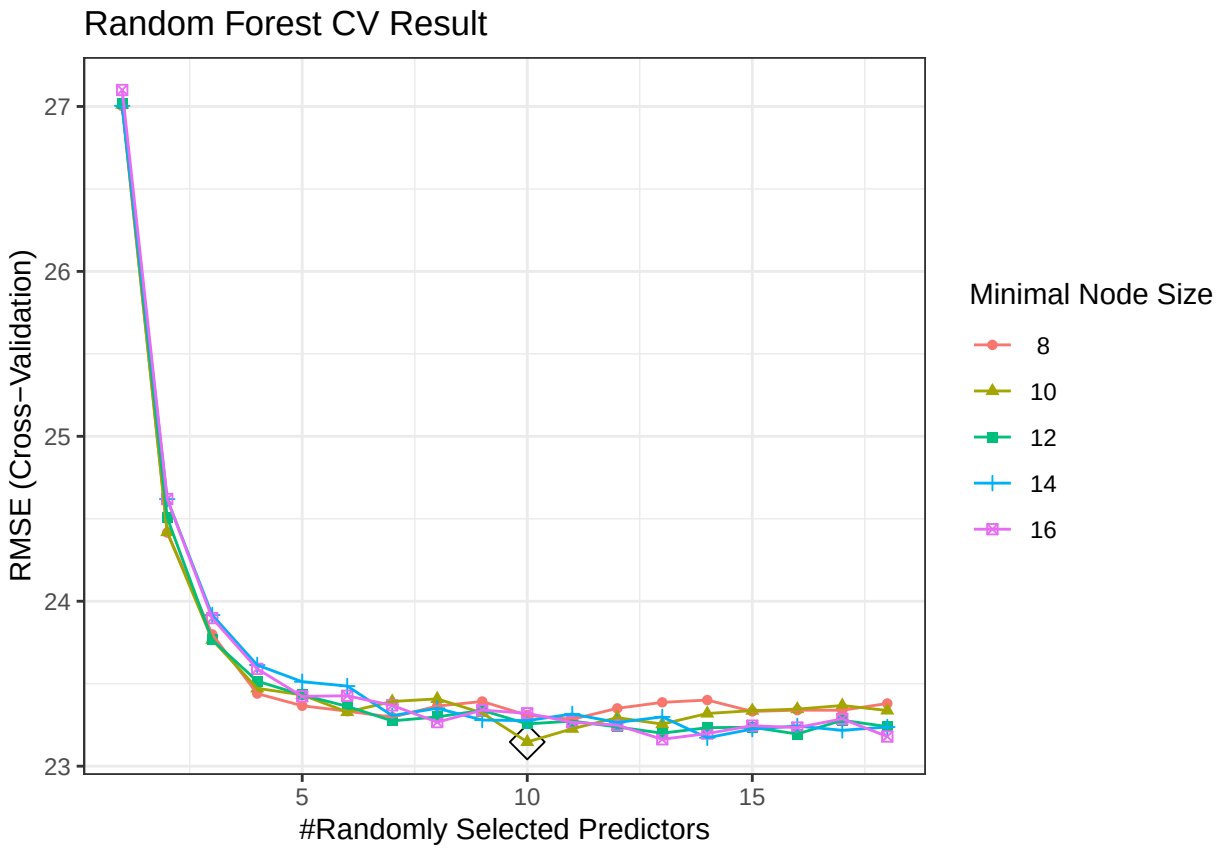

```
# p1 <- pdp::partial(
#   bag.fit,
#   pred.var = "Lot_Area",
#   grid.resolution = 20
#   ) %>%
#   autoplot()
# p2 <- pdp::partial(
#   bag.fit,
#   pred.var = "Lot_Frontage",
#   grid.resolution = 20
#   ) %>%
#   autoplot()
# gridExtra::grid.arrange(p1, p2, nrow = 1)
```

1.1.11 Random Forest

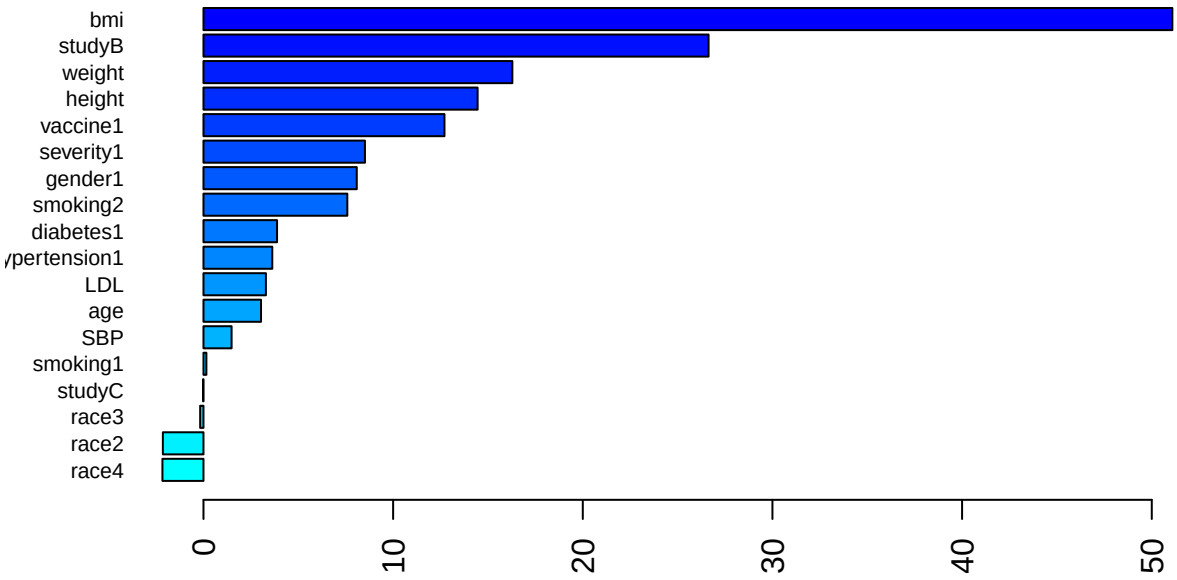
```
rf.grid <- expand.grid(mtry = 1:ncol(train.x),
                       splitrule = "variance",
                       min.node.size = seq(8, 16, by = 2))
set.seed(1)
rf.fit <- train(train.x,
                train.y,
                method = "ranger",
                tuneGrid = rf.grid,
                trControl = ctrl1)

rf.fit$bestTune
```

```
##    mtry splitrule min.node.size
## 47   10  variance            10
```

```
ggplot(rf.fit, highlight = TRUE) +
  labs(title = "Random Forest CV Result") +
  theme_bw()
```




```
ggsave("./figure/rf_cv.jpeg", dpi = 500)

rf.final.per <- ranger(recovery_time ~ .,
                       data = train.dat.matrix,
                       mtry = rf.fit$bestTune[[1]],
                       splitrule = "variance",
                       min.node.size = rf.fit$bestTune[[3]],
                       importance = "permutation",
                       scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(rf.final.per), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(ncol(train.x)))
```

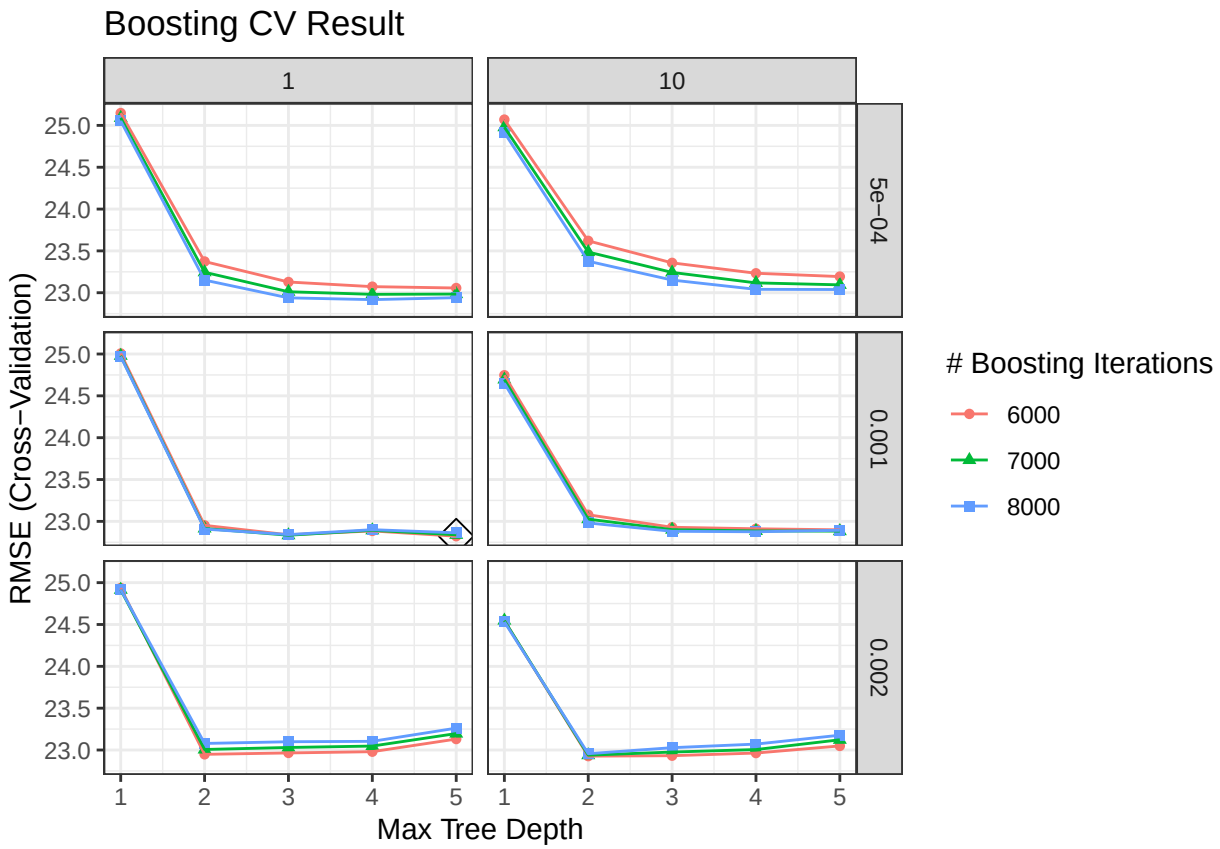


1.1.12 Boosting

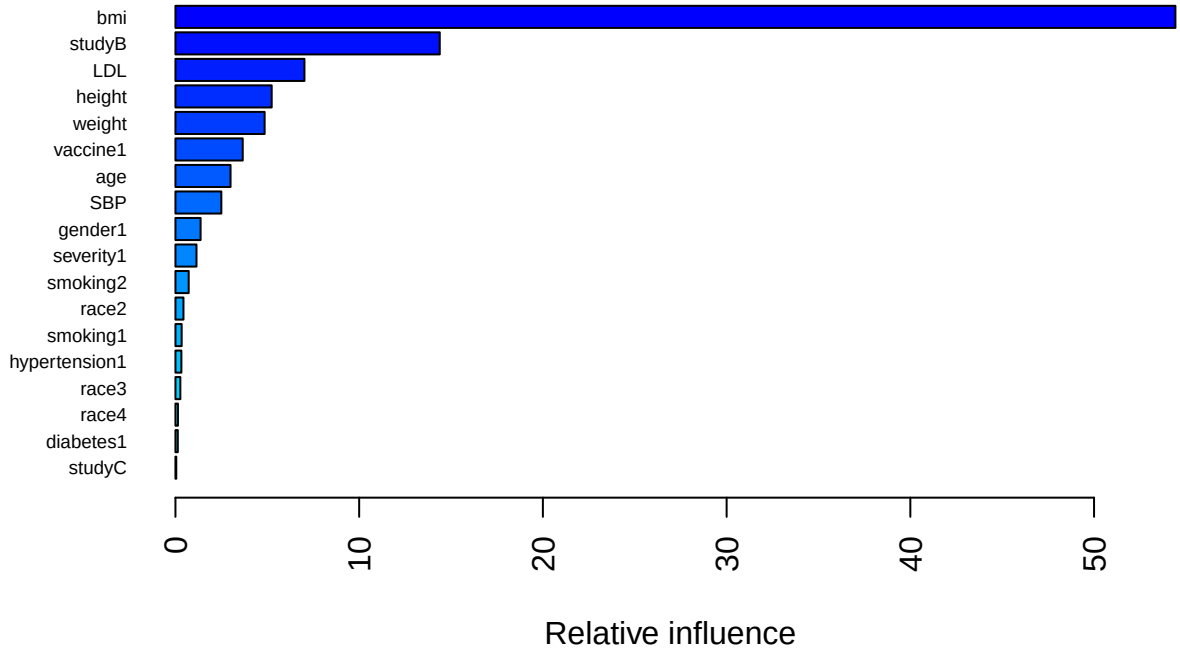
```
set.seed(1)
bst.grid <- expand.grid(n.trees = c(6000, 7000, 8000),
                        interaction.depth = 1:5,
                        shrinkage = c(0.0005, 0.001, 0.002),
                        n.minobsinnode = c(1,10))

bst.fit <- train(train.x,
                 train.y,
                 method = "gbm",
                 tuneGrid = bst.grid,
                 trControl = ctrl1,
                 verbose = FALSE)

bst.fit$bestTune
```

```
##    n.trees interaction.depth shrinkage n.minobsinnode
## 55    6000                 5     0.001              1
```

```
ggplot(bst.fit, highlight = TRUE) +
  labs(title = "Boosting CV Result") +
  theme_bw()
```




```
ggsave("./figure/boosting_cv.jpeg", dpi = 500)

# Variable Importance
summary(bst.fit$finalModel, las = 2, cBars = ncol(train.x), cex.names = 0.6)
```

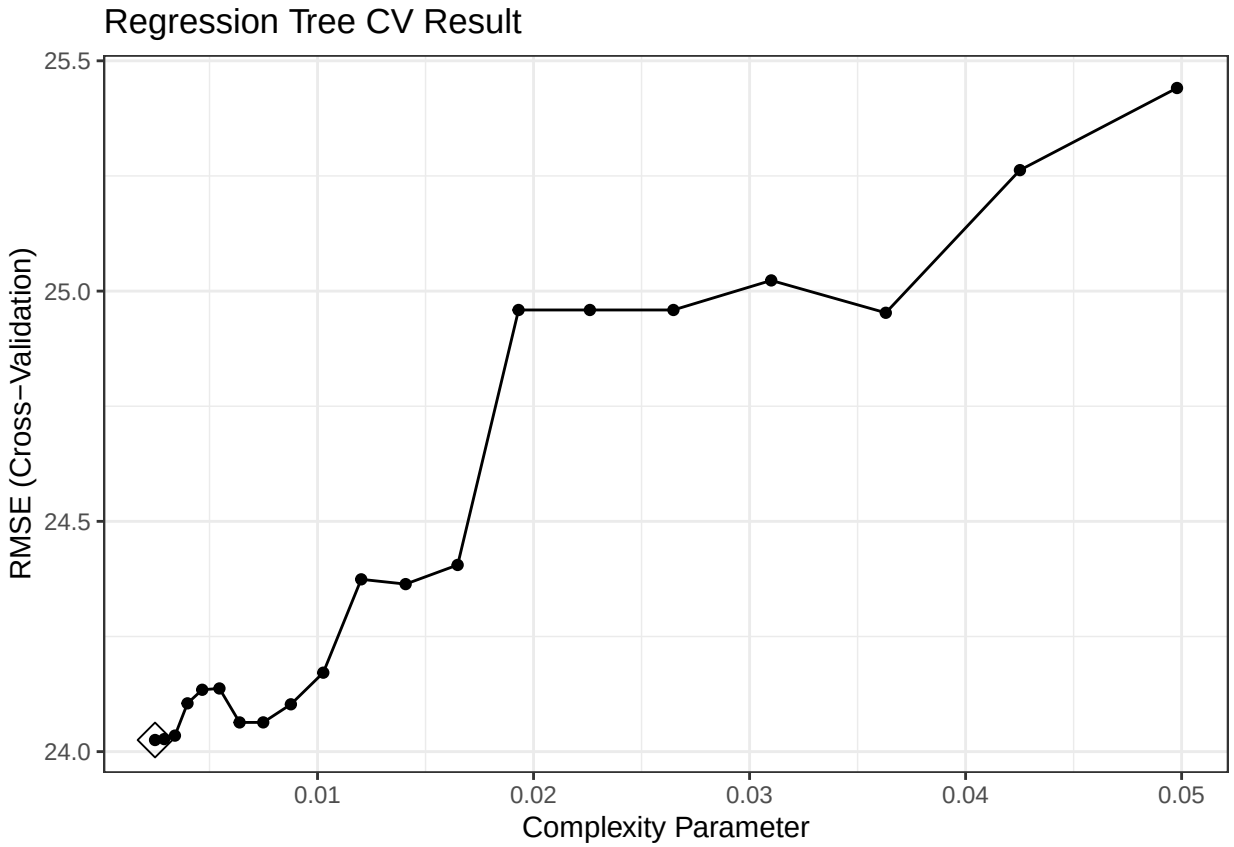



```
##                        var     rel.inf
```

```
## bmi                    bmi 54.42823959
## studyB              studyB 14.38641721
## LDL                    LDL  7.02137704
## height              height  5.23835738
## weight              weight  4.85450164
## vaccine1          vaccine1  3.66265204
## age                    age  2.99850055
## SBP                    SBP  2.49879651
## gender1            gender1  1.36812142
## severity1        severity1  1.14541590
## smoking2          smoking2  0.72209105
## race2                race2  0.43708547
## smoking1          smoking1  0.33759811
## hypertension1 hypertension1  0.32434377
## race3                race3  0.26762636
## race4                race4  0.13594602
## diabetes1        diabetes1  0.12810140
## studyC              studyC  0.04482855
```

1.1.13 Regression Trees

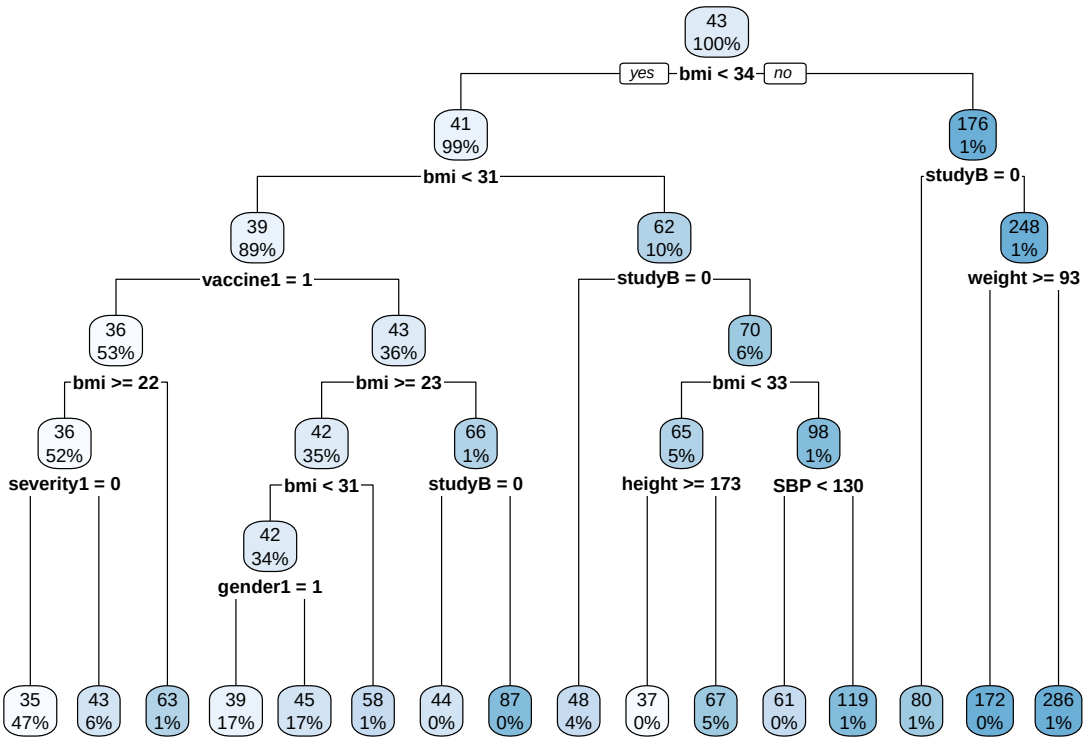
```
rpart.grid <- expand.grid(cp = exp(seq(-6,-3, length = 20)))
set.seed(1)
rpart.fit1 <- train(train.x,
                    train.y,
                    method = "rpart",
                    tuneGrid = rpart.grid,
                    trControl = ctrl1)

ggplot(rpart.fit1, highlight = TRUE) +
  labs(titlem = "Regression Tree CV Result") +
  theme_bw()
```




```
ggsave("./figure/rpart1_cv.jpeg", dpi = 500)

rpart.fit1$bestTune
```

```
##            cp
## 1 0.002478752
```

```
rpart.plot(rpart.fit1$finalModel)
```

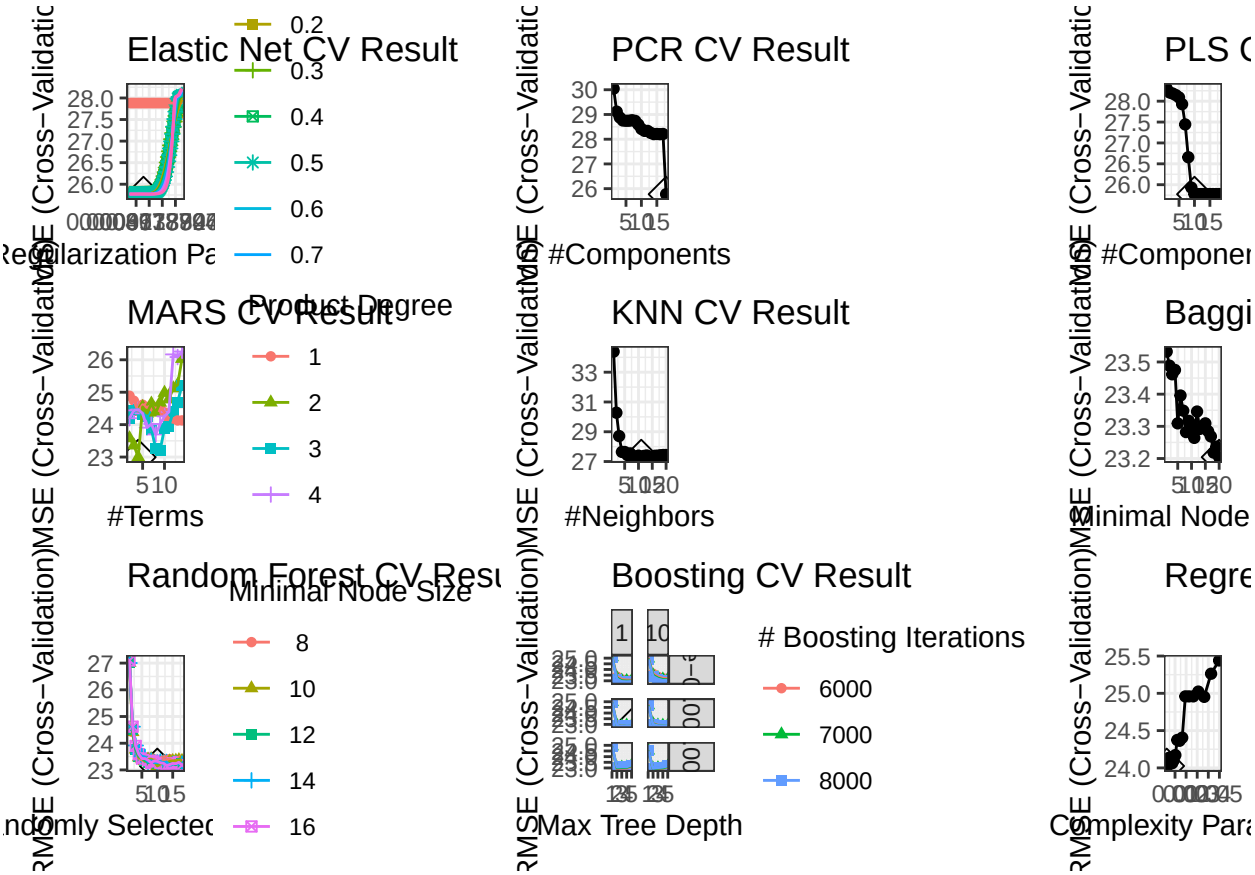



```
jpeg("./figure/rpart1.jpeg", width = 8, height = 6, units="in", res=500)
rpart.plot(rpart.fit1$finalModel)
dev.off()
```

```
## pdf
##   2
```

```
library(patchwork)
lasso <- ggplot(lasso.fit, highlight = TRUE) +
  labs(title="LASSO CV Result") +
  scale_x_continuous(trans='log',n.breaks = 10) +
  theme_bw()

ridge <- ggplot(ridge.fit,highlight = TRUE) +
  scale_x_continuous(trans='log', n.breaks = 6) +
  labs(title="Ridge CV Result") +
  theme_bw()

enet <- ggplot(enet.fit, highlight = TRUE) +
  scale_x_continuous(trans='log', n.breaks = 6) +
  labs(title ="Elastic Net CV Result") +
  theme_bw()

pcr <- ggplot(pcr.fit, highlight = TRUE) +
  labs(title  ="PCR CV Result") +
  theme_bw()

pls <- ggplot(pls.fit, highlight = TRUE) +
  labs(title  ="PLS CV Result") +
  theme_bw()
```

```
gam <- ggplot(gam.fit) +
  labs(title = "GAM CV Result") +
  theme_bw()

mars <- ggplot(mars.fit, highlight = TRUE)+
  labs(title  ="MARS CV Result") +
  theme_bw()

knn <- ggplot(knn.fit, highlight = TRUE) +
  labs(title  ="KNN CV Result") +
  theme_bw()

bagging <- ggplot(bag.fit, highlight = TRUE) +
  labs(title = "Bagging CV Result") +
  theme_bw()

rf <- ggplot(rf.fit, highlight = TRUE) +
  labs(title = "Random Forest CV Result") +
  theme_bw()

boosting <- ggplot(bst.fit, highlight = TRUE) +
  labs(title = "Boosting CV Result") +
  theme_bw()

tree <- ggplot(rpart.fit1, highlight = TRUE) +
  labs(titlem = "Regression Tree CV Result") +
  theme_bw()

p <- wrap_plots(enet, pcr,
           pls,
           mars, knn,
           bagging, rf, boosting, tree,
           ncol = 3)

print(p)
```



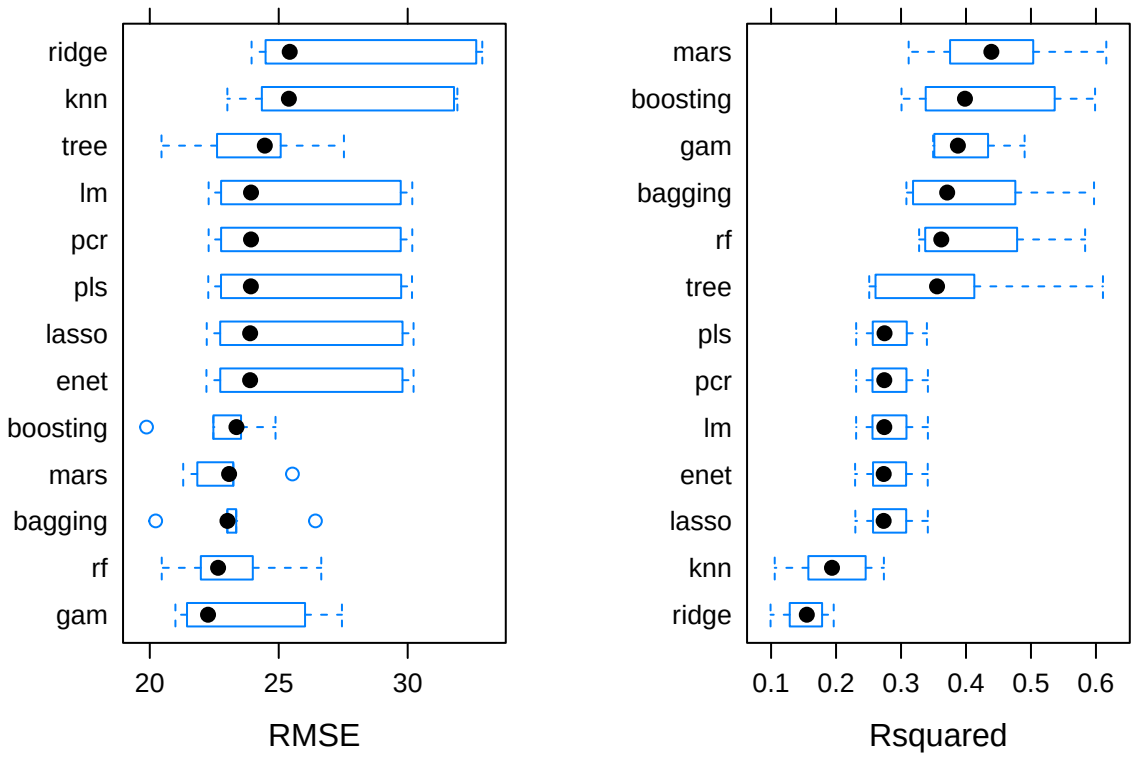
1.2 Model Selection

```
set.seed(1)
resamp <- resamples(list(lm = lm.fit,
                         lasso = lasso.fit,
                         ridge = ridge.fit,
                         enet = enet.fit,
                         pcr = pcr.fit,
                         pls = pls.fit,
                         gam = gam.fit,
                         mars = mars.fit,
                         knn = knn.fit,
                         bagging = bag.fit,
                         rf = rf.fit,
                         boosting = bst.fit,
                         tree = rpart.fit1))

summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: lm, lasso, ridge, enet, pcr, pls, gam, mars, knn, bagging, rf, boosting, tree
## Number of resamples: 5
##
```

```
## MAE
##                Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lm       15.96528 16.05841 16.15746 16.61362 17.35611 17.53083    0
## lasso    15.88262 15.98520 16.07771 16.54081 17.29954 17.45899    0
## ridge    16.20185 16.30253 16.30943 16.81917 17.53434 17.74768    0
## enet     15.87424 15.97981 16.07553 16.53944 17.30292 17.46472    0
## pcr      15.96528 16.05841 16.15746 16.61362 17.35611 17.53083    0
## pls      15.94127 16.05148 16.15418 16.60440 17.34318 17.53190    0
## gam      14.81848 14.86559 14.90671 15.35894 15.79482 16.40911    0
## mars     14.40217 14.82546 15.08888 15.13374 15.15379 16.19842    0
## knn      15.16523 15.33428 15.74108 16.20600 17.29065 17.49878    0
## bagging  14.30573 14.87425 14.93096 15.07427 15.02219 16.23824    0
## rf       14.27247 14.73418 14.87519 15.04845 15.07648 16.28392    0
## boosting 13.89063 14.41200 14.71826 14.67873 14.85924 15.51354    0
## tree     14.38914 14.83369 15.33260 15.27741 15.35360 16.47802    0
##
## RMSE
##                Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lm       22.28192 22.76489 23.92550 25.77618 29.73019 30.17840    0
## lasso    22.20841 22.72983 23.89091 25.77143 29.80066 30.22733    0
## ridge    23.94717 24.48964 25.42540 27.88259 32.65846 32.89230    0
## enet     22.19664 22.73144 23.88994 25.76880 29.79842 30.22759    0
## pcr      22.28192 22.76489 23.92550 25.77618 29.73019 30.17840    0
## pls      22.26788 22.76203 23.91897 25.77286 29.74750 30.16794    0
## gam      20.99325 21.44352 22.25827 23.63328 26.02019 27.45117    0
## mars     21.29599 21.84278 23.07553 22.99675 23.23852 25.53093    0
## knn      23.00812 24.34468 25.39307 27.29397 31.79649 31.92751    0
## bagging  20.22835 23.00556 23.01213 23.20424 23.35161 26.42358    0
## rf       20.46281 21.97878 22.64841 23.14669 23.99432 26.64912    0
## boosting 19.87397 22.46479 23.36056 22.82262 23.53813 24.87563    0
## tree     20.45432 22.60774 24.46022 24.02522 25.07597 27.52782    0
##
## Rsquared
##                 Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm       0.23089680 0.2560430 0.2742497 0.2821842 0.3083763 0.3413553    0
## lasso    0.22968512 0.2569237 0.2732588 0.2818156 0.3079845 0.3412259    0
## ridge    0.09919468 0.1287032 0.1550195 0.1515681 0.1785277 0.1963953    0
## enet     0.22929118 0.2569371 0.2733233 0.2817017 0.3078424 0.3411144    0
## pcr      0.23089680 0.2560430 0.2742497 0.2821842 0.3083763 0.3413553    0
## pls      0.23095996 0.2564685 0.2746187 0.2821505 0.3088601 0.3398454    0
## gam      0.34944854 0.3512306 0.3875093 0.4024771 0.4339766 0.4902202    0
## mars     0.31177294 0.3755147 0.4390225 0.4491446 0.5034847 0.6159280    0
## knn      0.10557822 0.1573365 0.1937424 0.1951755 0.2455872 0.2736333    0
## bagging  0.30825656 0.3184900 0.3710160 0.4141515 0.4759321 0.5970628    0
## rf       0.32782940 0.3371971 0.3619514 0.4177998 0.4786762 0.5833452    0
## boosting 0.30074844 0.3379932 0.3983908 0.4344615 0.5365054 0.5986697    0
## tree     0.25105096 0.2607188 0.3554261 0.3781581 0.4128121 0.6107827    0
```

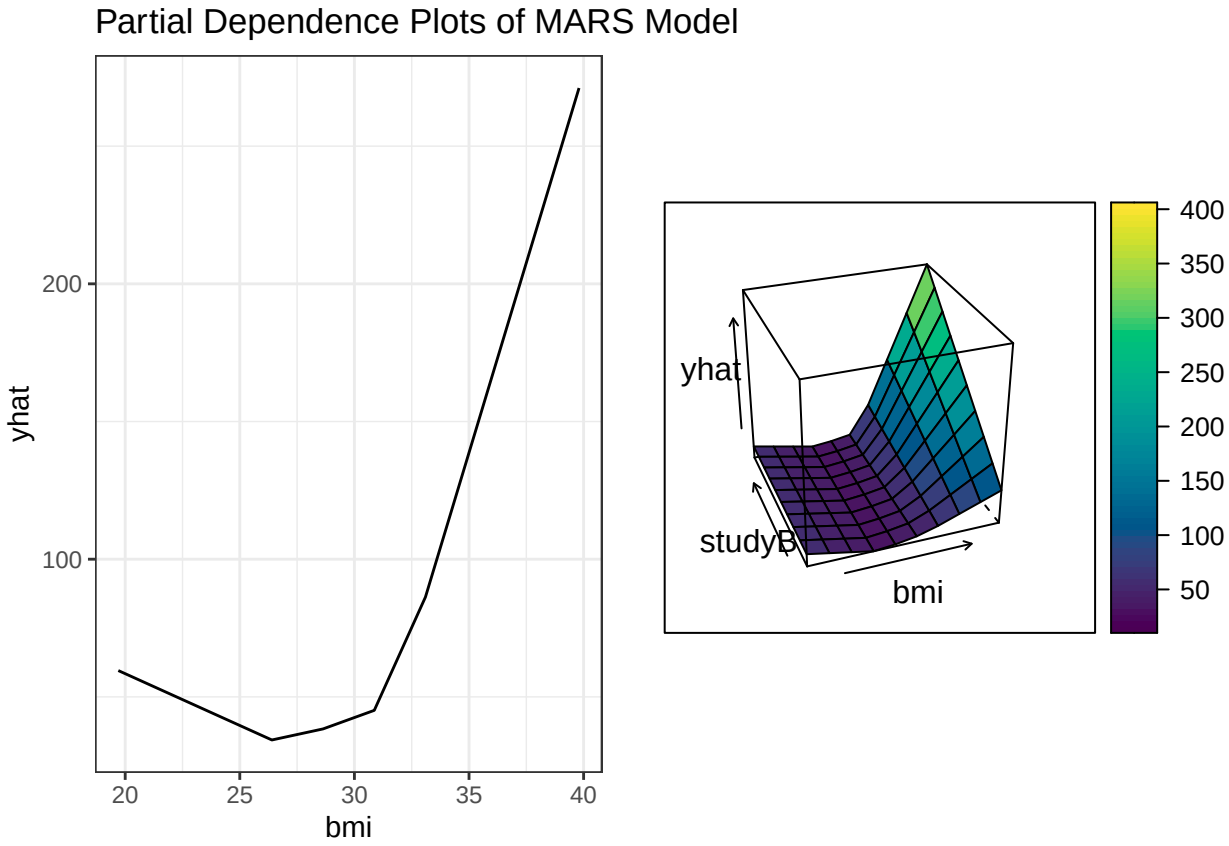
```
p1=bwplot(resamp, metric = "RMSE")
p2=bwplot(resamp, metric = "Rsquared")
grid.arrange(p1, p2 ,ncol=2)
```




```
jpeg("./figure/resample1.jpeg", width = 8, height=6, units="in", res=500)
p1=bwplot(resamp, metric = "RMSE")
p2=bwplot(resamp, metric = "Rsquared")
grid.arrange(p1, p2, ncol=2)
dev.off()
```

```
## pdf
##   2
```

```
p1<- pdp::partial(mars.fit, pred.var = c("bmi"), grid.resolution = 10) %>% autoplot() +
  theme_bw()+
  labs(title = "Partial Dependence Plots of MARS Model")

p2 <-pdp::partial(mars.fit, pred.var = c("bmi", "studyB"),
                  grid.resolution = 10) %>%
      pdp::plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE,
                       screen = list(z = 20, x = -60))

# jpeg("./figure/partial_dependence.jpeg", width = 8, height=6, units="in", res=500)
gridExtra::grid.arrange(p1, p2, ncol = 2)
```




```
# dev.off()

# Important variables
varImp(mars.fit$finalModel)
```

```
##         Overall
## bmi         100
## studyB      100
```

1.3 Training / Testing Error

```
# training error
mars.train.pred = predict(mars.fit, newdata = train.x)
RMSE(train.y, mars.train.pred)
```

```
## [1] 22.40806
```

```
# testing error
mars.pred = predict(mars.fit, newdata = test.x)
RMSE(test.y, mars.pred)
```

```
## [1] 22.59373
```