final_project

wz2631 rz2614 jn2855

2023-04-30

Data preparation

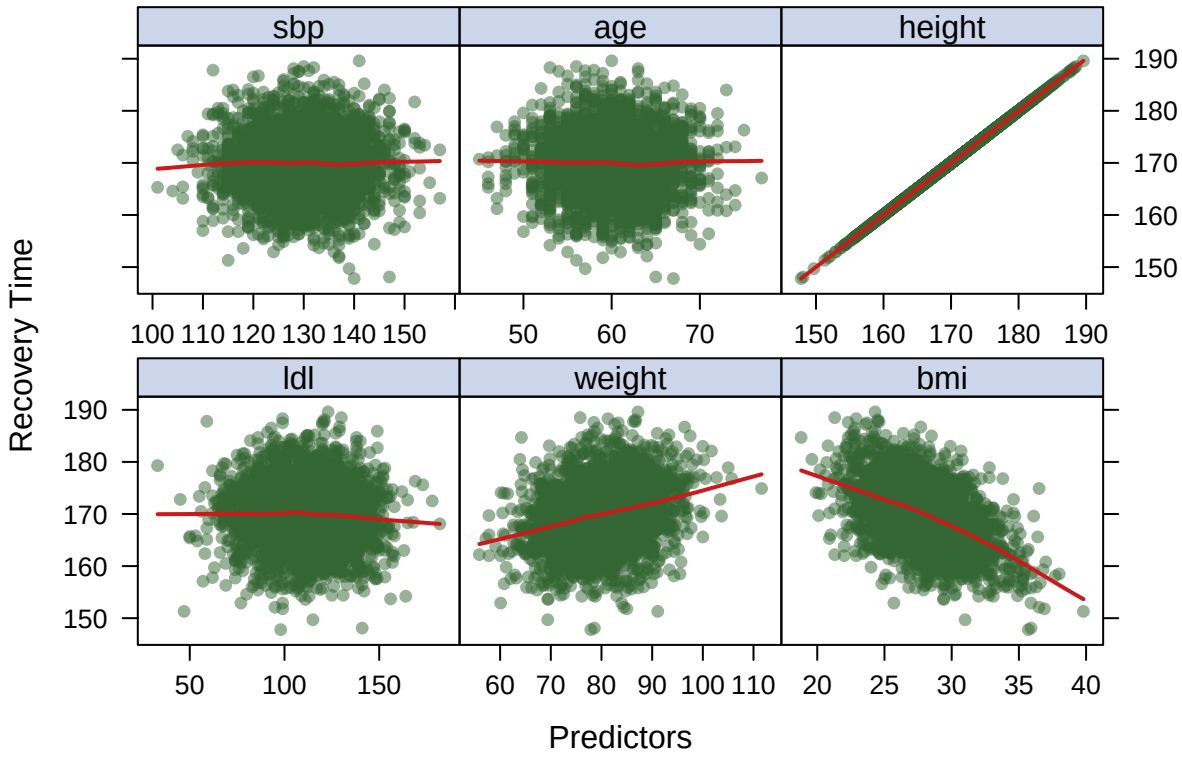
```
# draw 2 random samples of 2000 participants
load("./recovery.Rdata")
set.seed(2631)
dat1 <- dat[sample(1:10000, 2000),] %>%
  janitor::clean_names() %>%
  na.omit()
set.seed(2855)
dat2 <- dat[sample(1:10000, 2000),] %>%
  janitor::clean_names() %>%
  na.omit()
dat <- rbind.fill(dat1, dat2) %>%
   dplyr::select(-id) %>%
  unique() %>% mutate( gender=fct_recode(factor(gender),male='1',female='0'),
    race=fct_recode(factor(race),white='1',asian='2',black='3',hispanic='4'),
    smoking=fct_recode(factor(smoking),never='0',former='1',current='2'),
    hypertension=factor(hypertension),
    diabetes=factor(diabetes),
    vaccine=factor(vaccine),
    severity=factor(severity),
    study=factor(study)
  )
```

```
#data partition
set.seed(2023)
train_index=createDataPartition(y = dat$recovery_time,
                                p = 0.8,
                                list = FALSE)
train_dat=dat[train_index,]
test_dat=dat[-train_index,]
#training data
train_data=dat[train_index,]
x1 = model.matrix(recovery_time~., data=dat)[train_index,-1]
y1=dat$recovery_time[train_index]
#testing data
test_data=dat[-train_index,]
x2=model.matrix(recovery_time~., data=dat)[-train_index,-1]
y2=dat$recovery_time[-train_index]
```

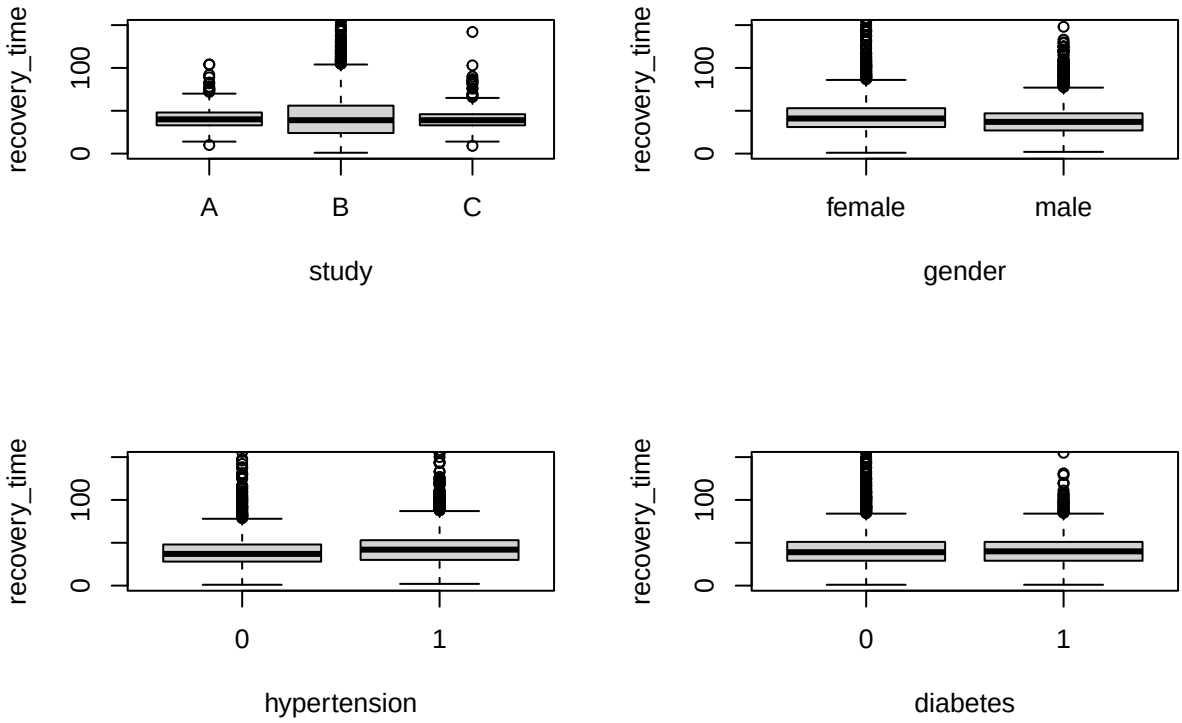
```r
#exploratory analysis and data visualization
visualization = train_dat %>%
  mutate(study=case_when(
    study == "A" ~ 1,
    study == "B" ~ 2,
    study == "C" ~ 3
  )) %>%
  dplyr::select(ldl,weight,bmi,sbp,age,height)
non_numeric= sapply(visualization, function(x) !is.numeric(x))
visualization[, non_numeric] = lapply(visualization[, non_numeric], as.numeric)
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 = visualization[ ,1:6],
            y = visualization[ ,6],
            plot = "scatter",
            span = .5,
            labels = c("Predictors", "Recovery Time"),
            main = "Figure 1. the relationship between predictors and recovery time",
            type = c("p", "smooth"))
```

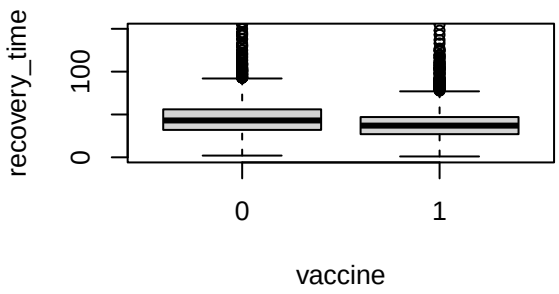
Figure 1. the relationship between predictors and recovery time

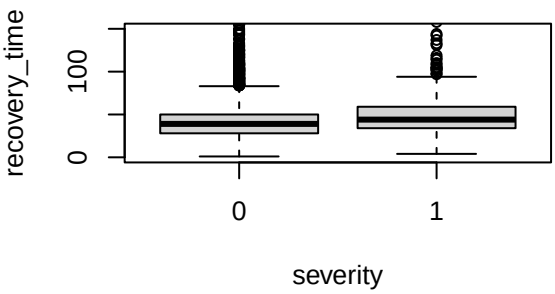


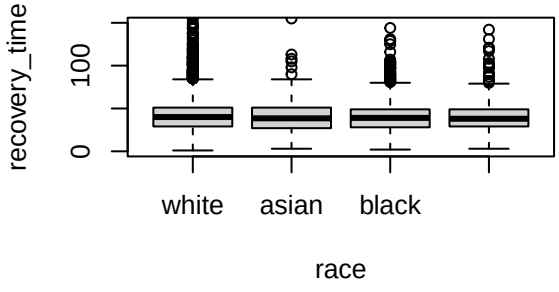
```
par(mfrow=c(2,2))
boxplot(recovery_time~study, data=dat, xlab="study", ylim=c(0,150))
boxplot(recovery_time~gender, data=dat, xlab="gender", ylim=c(0,150))
boxplot(recovery_time~hypertension, data=dat, xlab="hypertension", ylim=c(0,150))
boxplot(recovery_time~diabetes, data=dat, xlab="diabetes", ylim=c(0,150))
```

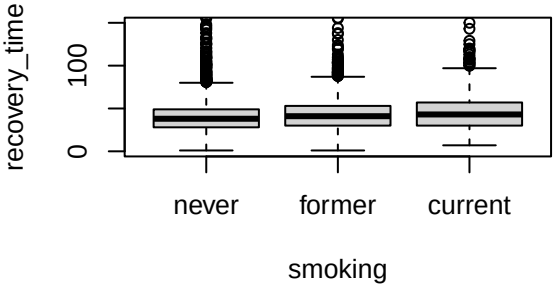



```
boxplot(recovery_time~vaccine, data=dat, xlab="vaccine", ylim=c(0,150))
boxplot(recovery_time~severity, data=dat, xlab="severity", ylim=c(0,150))
boxplot(recovery_time~race, data=dat, xlab="race", ylim=c(0,150))
boxplot(recovery_time~smoking, data=dat, xlab="smoking", ylim=c(0,150))
```







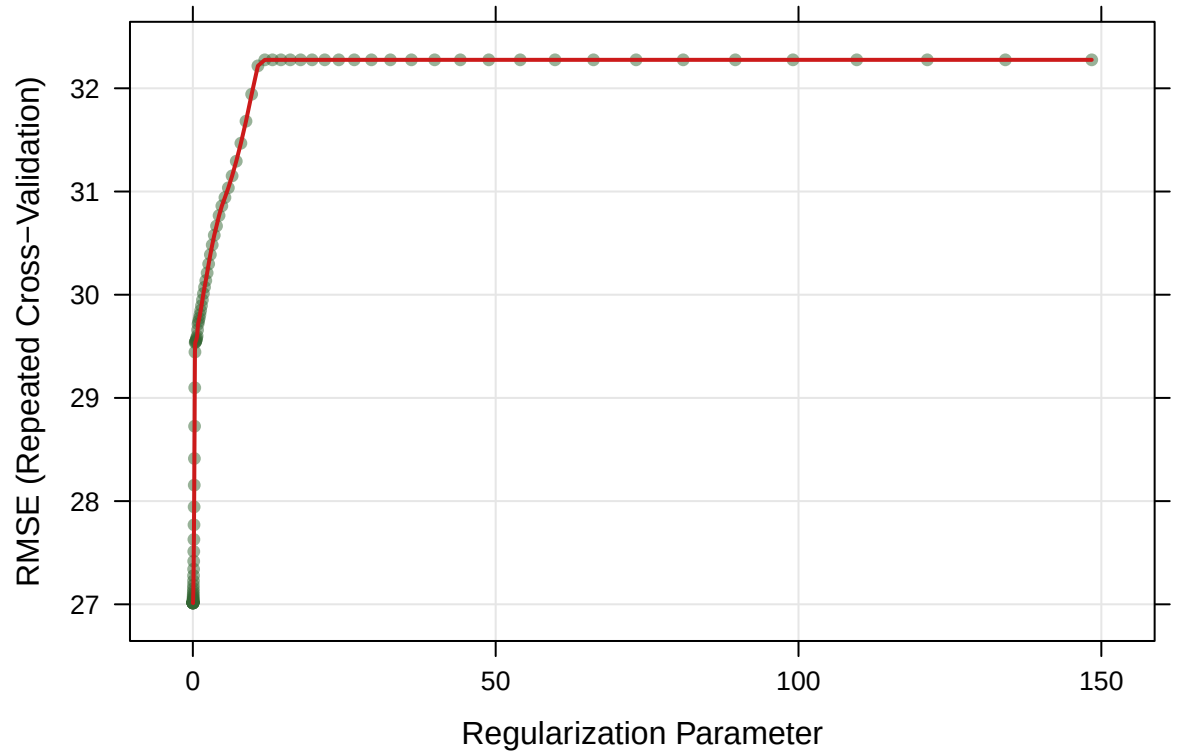



```r
#linear model
set.seed(2023)
ctrl=trainControl(method = "repeatedcv", number =10, repeats = 5)
linear = train(recovery_time ~ age + gender + race + smoking + height +
                        weight + bmi + hypertension + diabetes + sbp + ldl +
                        vaccine + severity + study,
               data = train_dat,
               method = "lm",
               trControl = ctrl)
summary(linear$finalModel)
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -85.895 -14.897  -1.583  11.054 250.397
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.398e+03  1.372e+02 -24.774  < 2e-16 ***
## age             1.790e-01  1.272e-01   1.406   0.1597
## gendermale     -5.601e+00  1.008e+00  -5.556 3.02e-08 ***
## raceasian      -2.674e-01  2.354e+00  -0.114   0.9096
## raceblack      -2.943e+00  1.272e+00  -2.314   0.0207 *
## racehispanic   -1.030e+00  1.734e+00  -0.594   0.5525
```

```
## smokingformer   5.233e+00  1.135e+00   4.611 4.19e-06 ***
## smokingcurrent  7.340e+00  1.696e+00   4.328 1.56e-05 ***
## height          1.973e+01  8.058e-01  24.479  < 2e-16 ***
## weight         -2.134e+01  8.486e-01 -25.140  < 2e-16 ***
## bmi             6.424e+01  2.427e+00  26.468  < 2e-16 ***
## hypertension1   2.655e+00  1.682e+00   1.578   0.1146
## diabetes1       9.470e-01  1.373e+00   0.690   0.4904
## sbp             5.394e-02  1.096e-01   0.492   0.6226
## ldl            -4.210e-02  2.686e-02  -1.567   0.1171
## vaccine1       -8.254e+00  1.028e+00  -8.025 1.46e-15 ***
## severity1       8.467e+00  1.630e+00   5.195 2.19e-07 ***
## studyB          6.723e+00  1.308e+00   5.139 2.95e-07 ***
## studyC          3.407e-01  1.595e+00   0.214   0.8309
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.97 on 2859 degrees of freedom
## Multiple R-squared:  0.3235, Adjusted R-squared:  0.3192
## F-statistic: 75.94 on 18 and 2859 DF,  p-value: < 2.2e-16
RMSE
## function (pred, obs, na.rm = FALSE)
## sqrt(mean((pred - obs)^2, na.rm = na.rm))
## <bytecode: 0x7feaa63f8ca0>
## <environment: namespace:caret>
test_pred1=predict(linear,newdata = test_dat)
rmse1=sqrt(mean((test_pred1-test_dat$recovery_time)**2))
rmse1
## [1] 23.74624
```

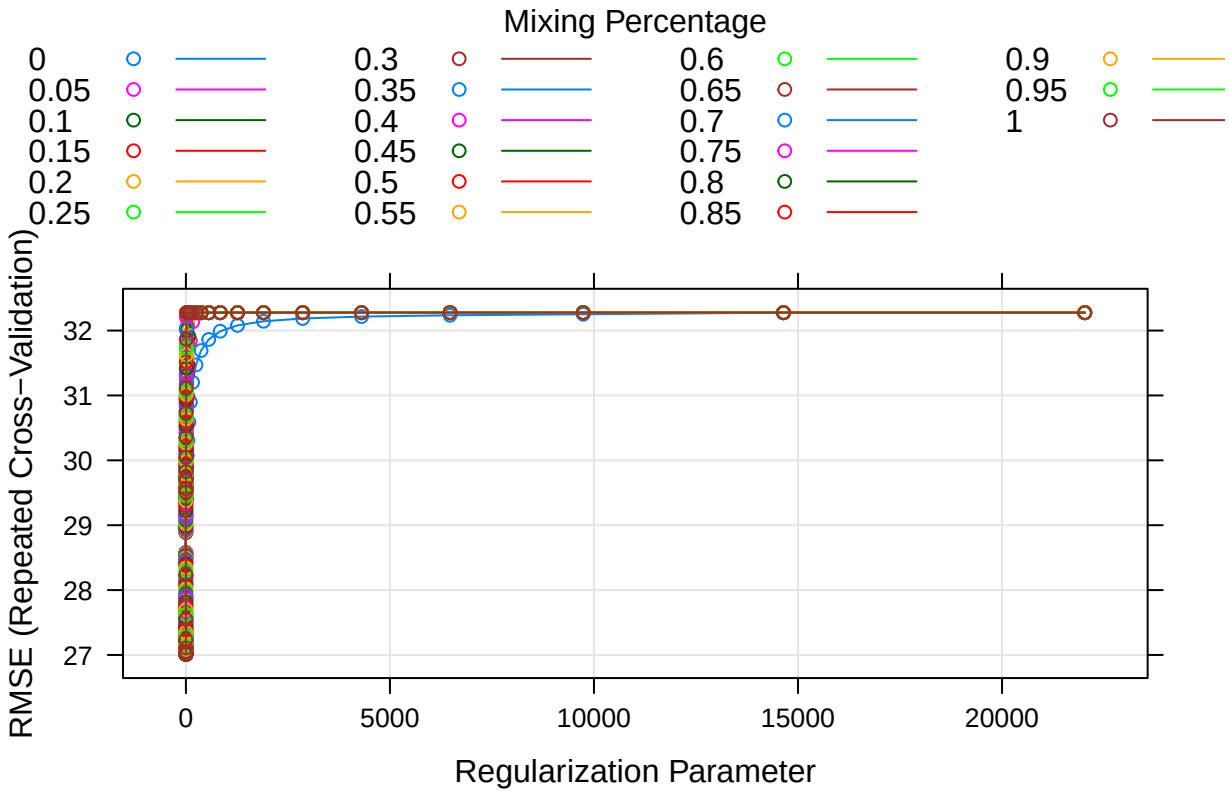
```
#lasso
set.seed(2023)
ctrl=trainControl(method = "repeatedcv", number =10, repeats = 5)
lasso=train(x1,y1,
            method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(-5, 5, length = 100))),
                   trControl = ctrl)
coef(lasso$finalModel, lasso$bestTune$lambda)
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)    -3.296528e+03
## age             1.741806e-01
## gendermale     -5.573616e+00
## raceasian      -2.728152e-01
## raceblack      -2.937140e+00
## racehispanic   -9.971317e-01
## smokingformer   5.221471e+00
## smokingcurrent  7.329564e+00
## height          1.912639e+01
## weight         -2.070218e+01
## bmi             6.242957e+01
## hypertension1   2.623883e+00
## diabetes1       8.784877e-01
```

```
## sbp              5.597457e-02
## ldl             -4.154013e-02
## vaccine1        -8.265120e+00
## severity1        8.465667e+00
## studyB           6.702389e+00
## studyC           2.989155e-01
lasso$bestTunetest_pred2=predict(lasso,newdata=x2)
pred_lasso=predict(lasso, newx = x2, s = lasso$lambda.min)
rmse_lasso= sqrt(mean((pred_lasso-y2)**2))
rmse_lasso
## [1] 31.86435
coef=coef(lasso, s = lasso$lambda.min)
n.pred=sum(coef[-1] != 0)
n.pred
## [1] 0
plot(lasso)
```




```
#elastic net
set.seed(2023)
elastic_net=train(x1, y1,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                         lambda = exp(seq(10, -10, length = 50))),
                  trControl = ctrl)
elastic_net$bestTune
```

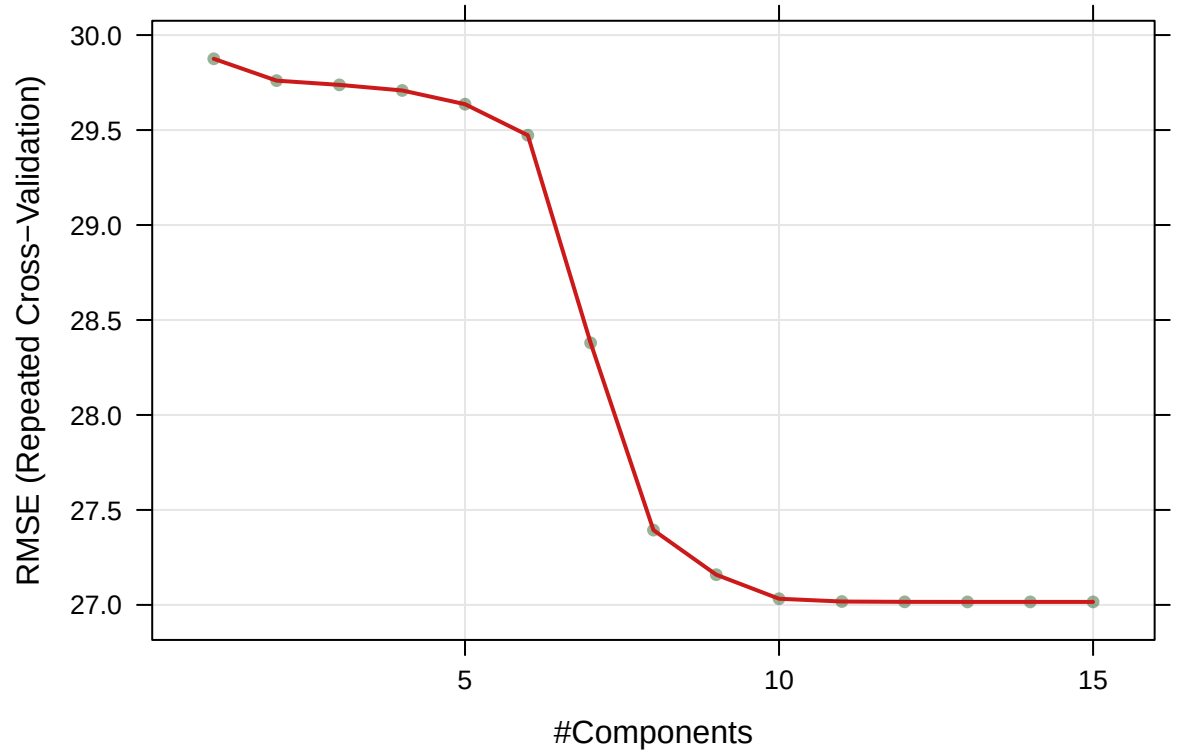
```
##     alpha      lambda
## 410   0.4 0.001788227
test_pred_elastic=predict(elastic_net, newdata = x2)
rmse_elastic=sqrt(mean((test_pred_elastic - test_dat$recovery_time)**2))
rmse_elastic
## [1] 23.69623
plot(elastic_net)
```




```
#pls
set.seed(2023)
pls=train(x1, y1,
          method = "pls",
          tuneGrid = data.frame(ncomp = 1:15), # CHECK THIS
          trControl = ctrl,
          preProcess = c("center", "scale"))

summary(pls$finalModel)
## Data:    X dimension: 2878 18
##  Y dimension: 2878 1
## Fit method: oscorespls
## Number of components considered: 14
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X           9.381    17.17    26.42    32.11    37.50    43.75    47.00
## .outcome   15.787    16.77    16.96    17.18    17.64    18.83    26.21
```

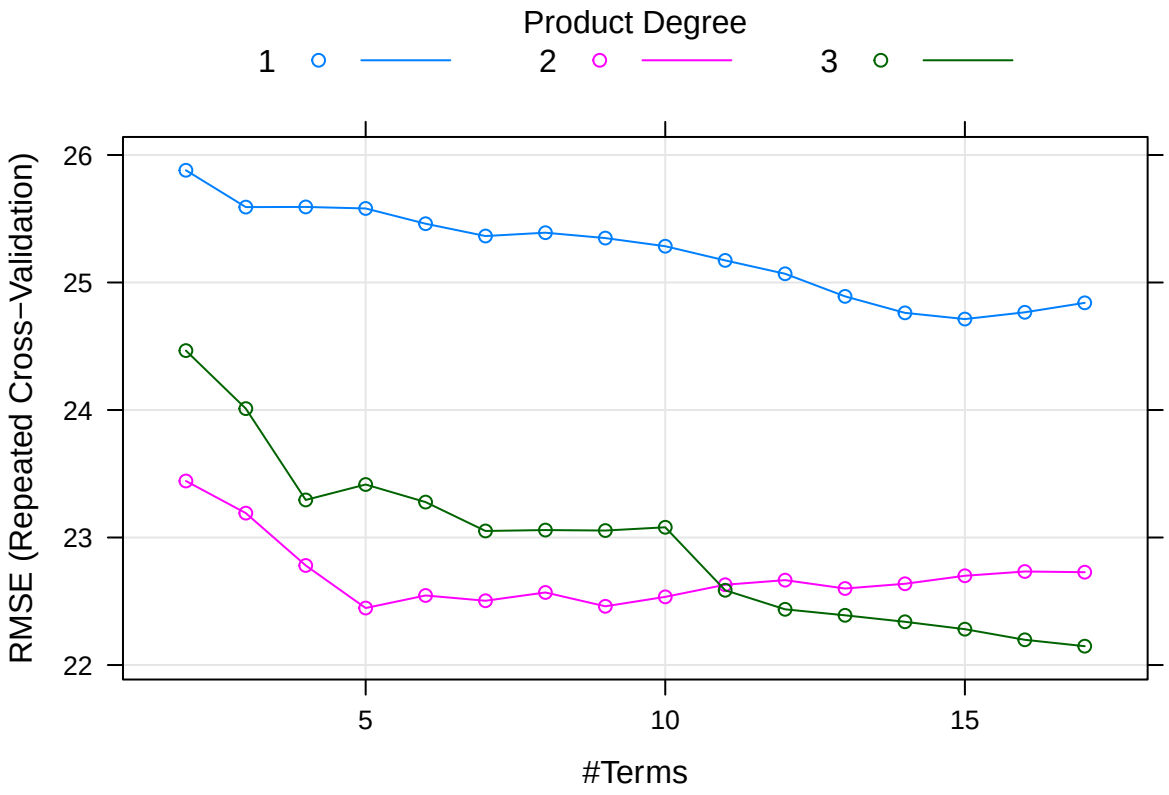
```
##           8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X           50.35    55.19     58.08     63.03     68.03     73.65     78.61
## .outcome    31.89    32.13     32.32     32.35     32.35     32.35     32.35
test_pred_pls=predict(pls, newdata = x2)
rmse_pls=sqrt(mean((test_pred_pls - test_dat$recovery_time)**2))
rmse_pls
## [1] 23.74624
plot(pls)
```




```
#mars
set.seed(2023)
mars_grid = expand.grid(degree = 1:3,
                        nprune = 2:17)
mars = train(x1, y1,
             method = "earth",
             tuneGrid = mars_grid,
             trControl = ctrl)
kable(mars$bestTune,"simple")
```

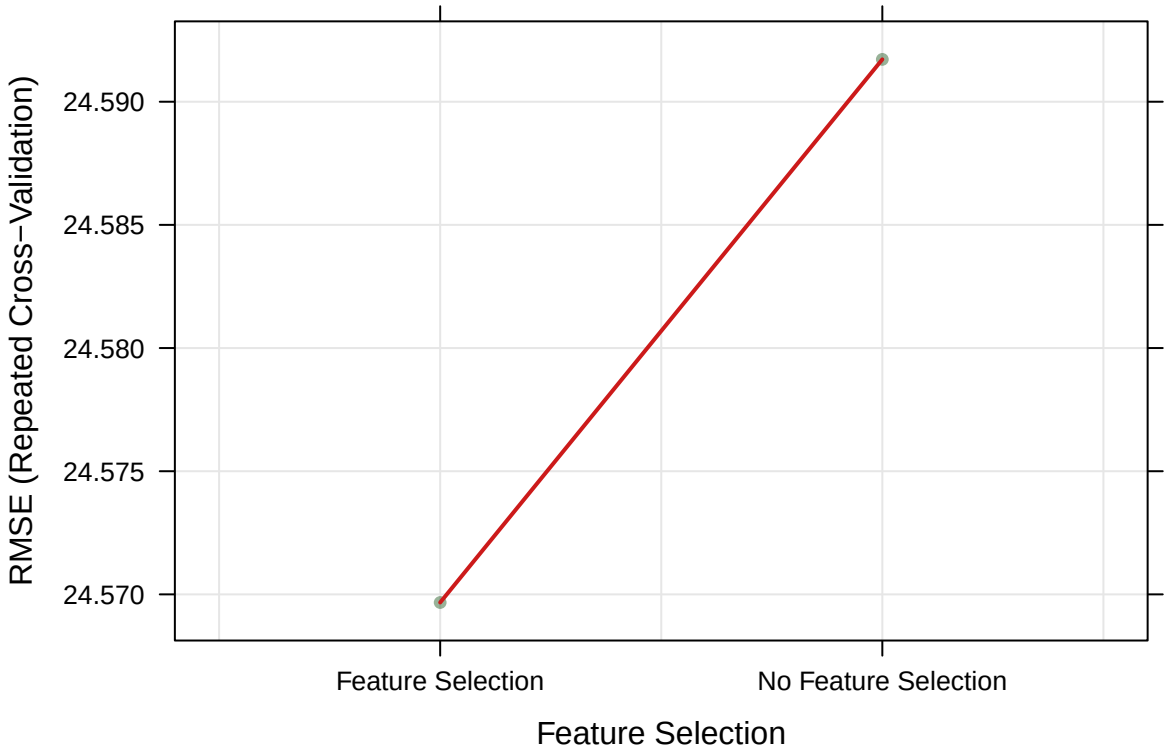
	nprune	degree
48	17	3

```
coef(mars$finalModel)
##                             (Intercept)                             h(bmi-31.4)
##                             -22.0832234                             -16.6988840
##                             h(31.4-bmi)                   h(bmi-31.4) * studyB
##                               8.9172375                              46.2961842
##       h(bmi-31.4) * severity1 * studyB                              h(bmi-24)
##                              23.3320342                               8.5210966
##       h(age-63) * h(bmi-31.4) * studyB                               vaccine1
##                              51.7088130                              -7.4947824
##      h(bmi-31.4) * h(sbp-136) * studyB      h(bmi-31.4) * h(136-sbp) * studyB
##                              -3.0279796                              -1.5779592
## h(height-156.2) * h(bmi-31.4) * studyB       h(age-64) * h(bmi-31.4) * studyB
##                              -0.8211457                             -32.1025357
##  smokingcurrent * h(bmi-31.4) * studyB                 gendermale * h(bmi-24)
##                              17.6476180                              -1.1080574
##                             h(bmi-28.3)                              h(bmi-33)
##                               5.7012625                              17.7953729
##       h(age-61) * h(bmi-31.4) * studyB
##                             -14.1650741
test_pred_mars=predict(mars, newdata = x2)
rmse_mars=sqrt(mean((test_pred_mars - test_dat$recovery_time)**2))
rmse_mars
## [1] 20.27082
summary(mars)
## Call: earth(x=matrix[2878,18], y=c(15,56,42,62,4...), keepxy=TRUE, degree=3,
##             nprune=17)
##
##                                         coefficients
## (Intercept)                              -22.083223
## vaccine1                                  -7.494782
## h(bmi-24)                                  8.521097
## h(bmi-28.3)                                5.701263
## h(31.4-bmi)                                8.917238
## h(bmi-31.4)                              -16.698884
## h(bmi-33)                                 17.795373
## gendermale * h(bmi-24)                    -1.108057
## h(bmi-31.4) * studyB                      46.296184
## smokingcurrent * h(bmi-31.4) * studyB     17.647618
## h(bmi-31.4) * severity1 * studyB          23.332034
## h(age-61) * h(bmi-31.4) * studyB         -14.165074
## h(age-63) * h(bmi-31.4) * studyB          51.708813
## h(age-64) * h(bmi-31.4) * studyB         -32.102536
## h(height-156.2) * h(bmi-31.4) * studyB    -0.821146
## h(bmi-31.4) * h(sbp-136) * studyB         -3.027980
## h(bmi-31.4) * h(136-sbp) * studyB         -1.577959
##
## Selected 17 of 24 terms, and 9 of 18 predictors (nprune=17)
## Termination condition: Reached nk 37
## Importance: bmi, studyB, age, vaccine1, severity1, smokingcurrent, sbp, ...
## Number of terms at each degree of interaction: 1 6 2 8
## GCV 415.8495    RSS 1162958    GRSq 0.6109742    RSq 0.6217165
plot(mars)
```

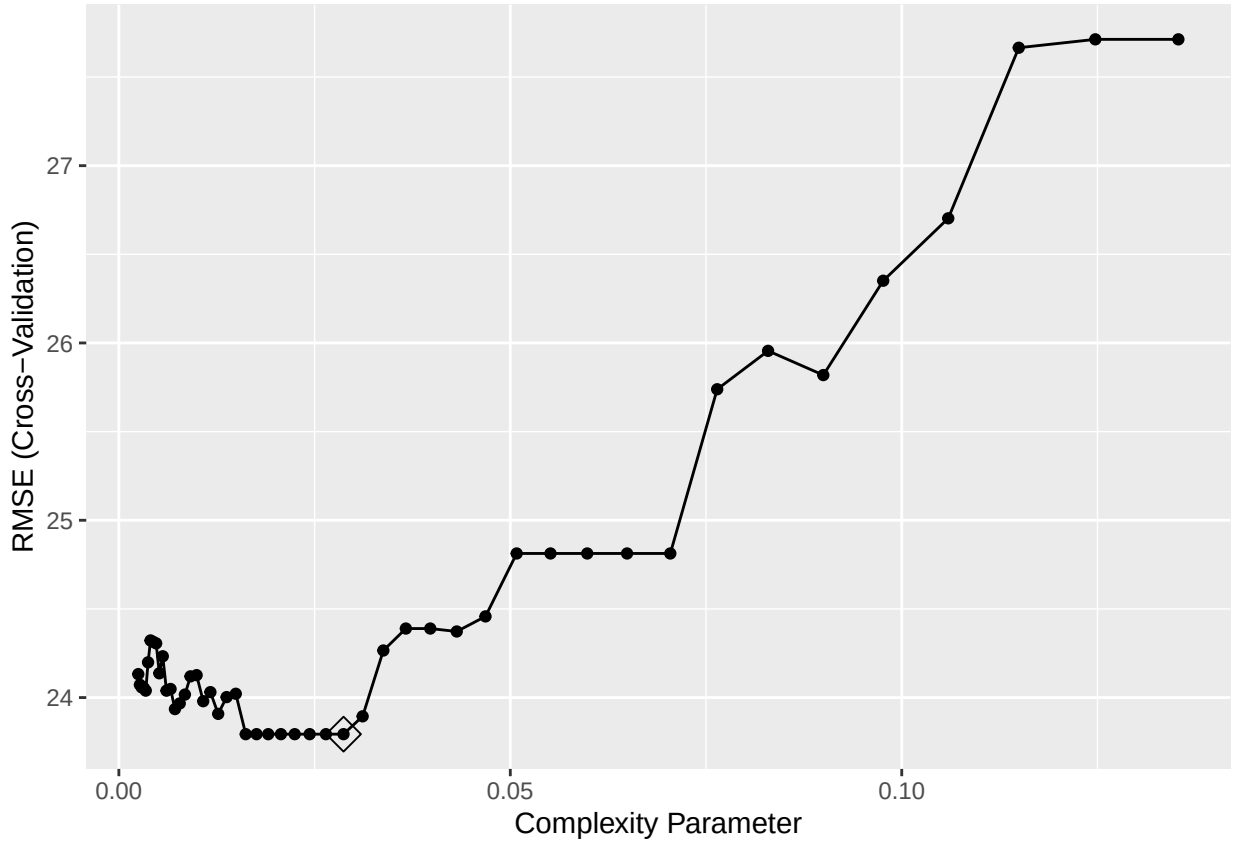



```
#gam
gam = train(x1, y1,
                  method = "gam",
                  trControl = ctrl,
                  control = gam.control(maxit = 200))
summary(gam$finalModel)
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gendermale + raceblack + racehispanic + smokingformer +
##     smokingcurrent + hypertension1 + diabetes1 + vaccine1 + severity1 +
##     studyB + studyC + s(age) + s(sbp) + s(ldl) + s(bmi) + s(height) +
##     s(weight)
##
## Parametric coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      42.4990     1.3926  30.518  < 2e-16 ***
## gendermale       -4.9828     0.8941  -5.573 2.74e-08 ***
## raceblack        -1.8639     1.1186  -1.666   0.0958 .
## racehispanic     -0.4713     1.5248  -0.309   0.7573
## smokingformer     5.2081     1.0034   5.190 2.25e-07 ***
## smokingcurrent    8.1057     1.4999   5.404 7.05e-08 ***
## hypertension1     3.8452     0.8935   4.304 1.74e-05 ***
## diabetes1         1.2776     1.2141   1.052   0.2927
```

```
## vaccine1        -8.2248     0.9097  -9.041  < 2e-16 ***
## severity1        8.3207     1.4403   5.777 8.43e-09 ***
## studyB           7.1837     1.1580   6.204 6.32e-10 ***
## studyC           0.8492     1.4104   0.602   0.5472
## ---
## 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)    4.692e-02      9   0.005    0.316
## s(sbp)    1.455e-08      9   0.000    0.942
## s(ldl)    8.525e-08      9   0.000    0.345
## s(bmi)    6.722e+00      9 137.689  < 2e-16 ***
## s(height) 6.400e+00      9   3.144 4.12e-05 ***
## s(weight) 7.155e+00      9   4.560  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.47   Deviance explained = 47.5%
## GCV = 573.08  Scale est. = 566.65    n = 2878
gam$df.residual
## NULL
test_pred_gam=predict(gam, newdata = x2)
rmse_gam=sqrt(mean((test_pred_gam-test_dat$recovery_time)**2))
rmse_gam
## [1] 20.58043
plot(gam)
```



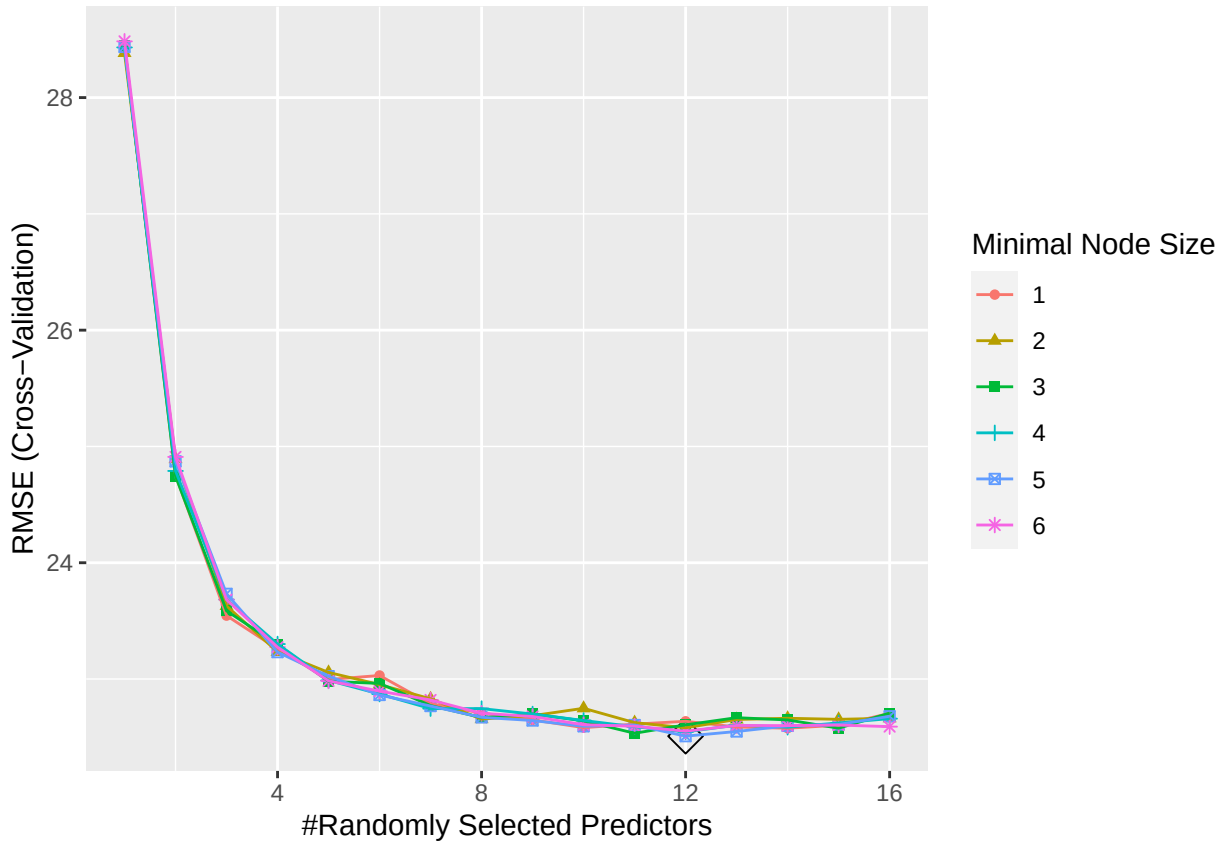

```
#tree
ctrl1=trainControl(method = "cv")
set.seed(2023)
rpart_fit <- train(recovery_time ~., data = dat[train_index,],
                   method = "rpart",
                   tuneGrid = data.frame(cp = exp(seq(-6, -2, length = 50))),
                   trControl = ctrl1)
rpart_fit$bestTune
##            cp
## 31 0.02869534
ggplot(rpart_fit,highlight = TRUE)
```



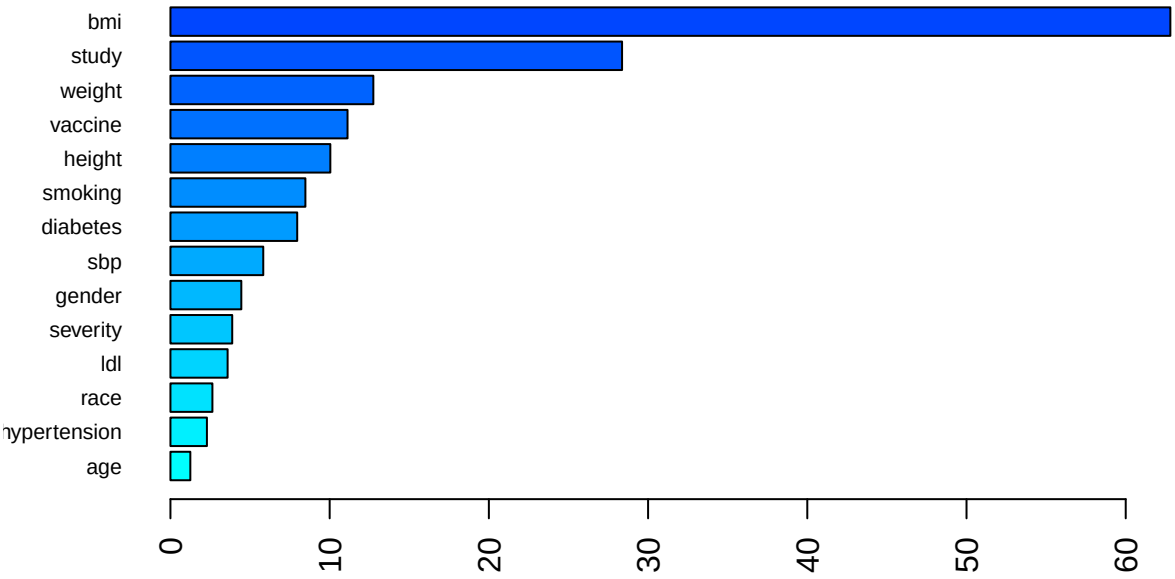

```
test_pred_tree = predict(rpart_fit, newdata = dat[-train_index, ])
rmse_tree = mean((test_pred_tree - dat$recovery_time[-train_index])**2)
rmse_tree
## [1] 410.5504
```

The cp value is 0.0286953.

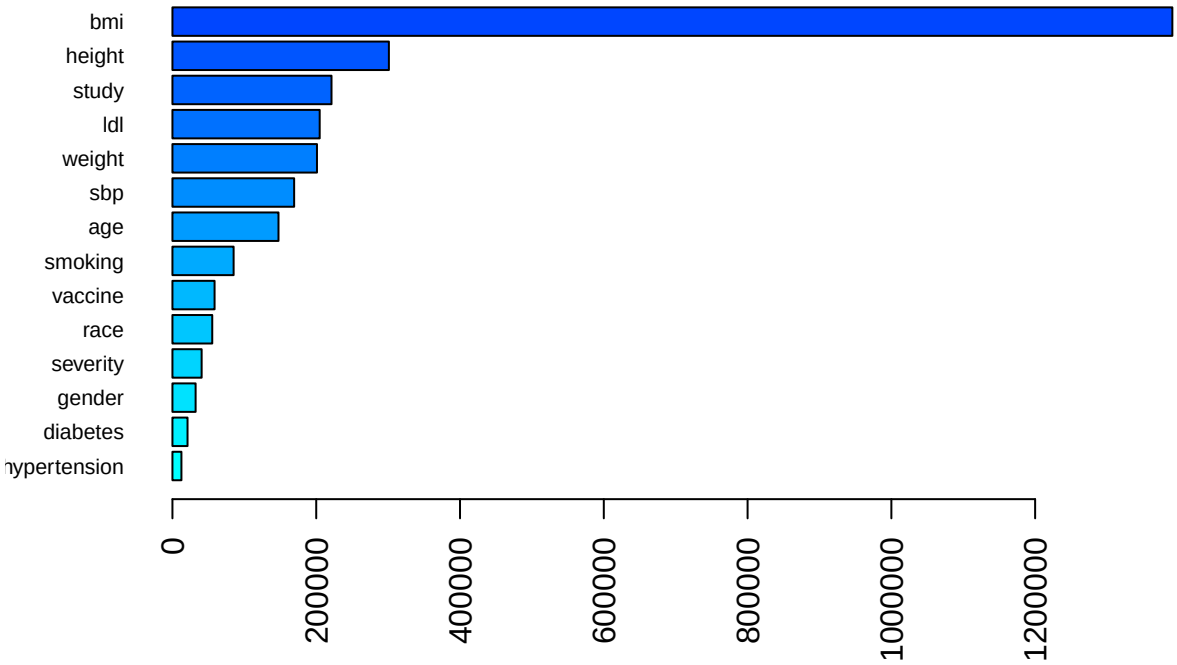
```
#rf
rf_grid=expand.grid(mtry = 1:16,
                        splitrule = "variance",
                        min.node.size = 1:6)
set.seed(2023)
rf_fit=train(recovery_time ~.,
                data = dat[train_index,],
                method = "ranger",
                tuneGrid = rf_grid,
                trControl = ctrl1)
rf_fit$bestTune
##    mtry splitrule min.node.size
## 71   12  variance             5
ggplot(rf_fit,highlight=TRUE)
```




```
set.seed(2023)
rf_perm = ranger(recovery_time ~ . ,
                 train_dat,
                 mtry = rf_fit$bestTune[[1]],
                 splitrule = "variance",
                 min.node.size = rf_fit$bestTune[[3]],
                 importance = "permutation",
                 scale.permutation.importance = TRUE)
barplot(sort(importance(rf_perm), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(19))
```




```
rf_imp = ranger(recovery_time ~ . ,
                train_dat,
                mtry = rf_fit$bestTune[[1]],
                splitrule = "variance",
                min.node.size = rf_fit$bestTune[[3]],
                importance = "impurity")
barplot(sort(importance(rf_imp), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(19))
```


bmi
height
study
ldl
weight
sbp
age
smoking
vaccine
race
severity
gender
diabetes
hypertension
0
200000
400000
600000
800000
1000000
1200000