P8106_group2recovery_primaryanalysis

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

Contents

Import and data manipulation

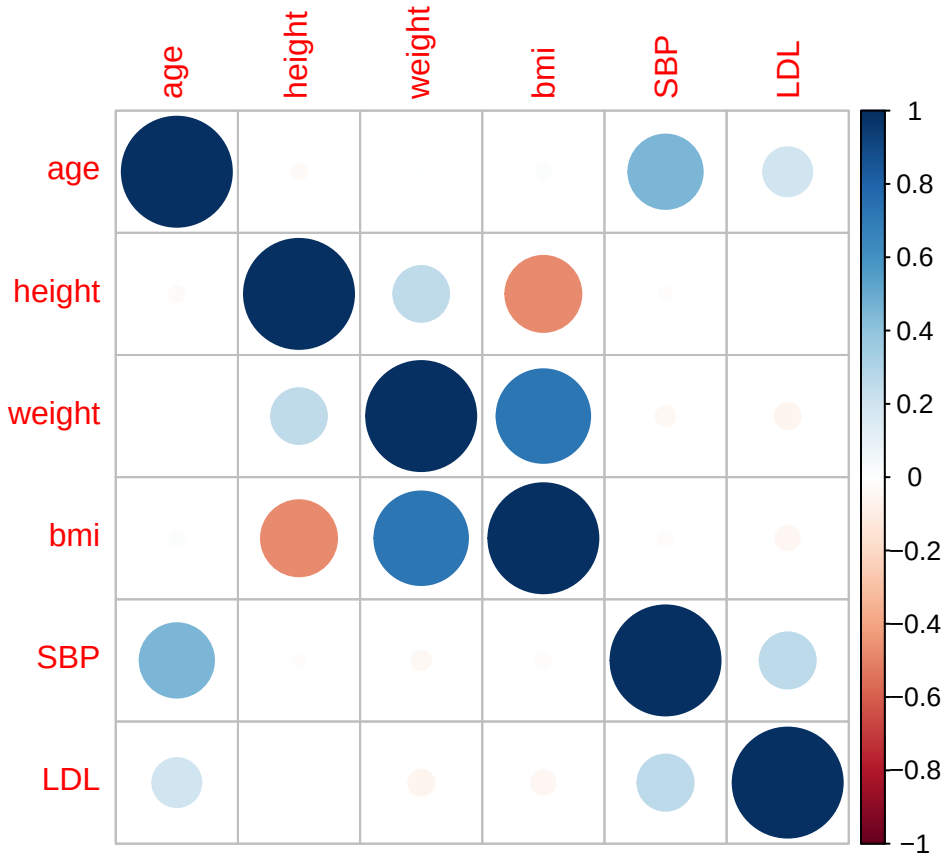
```
# Load recovery.RData environment
load("./recovery.Rdata")

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, ]

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

Data visualization

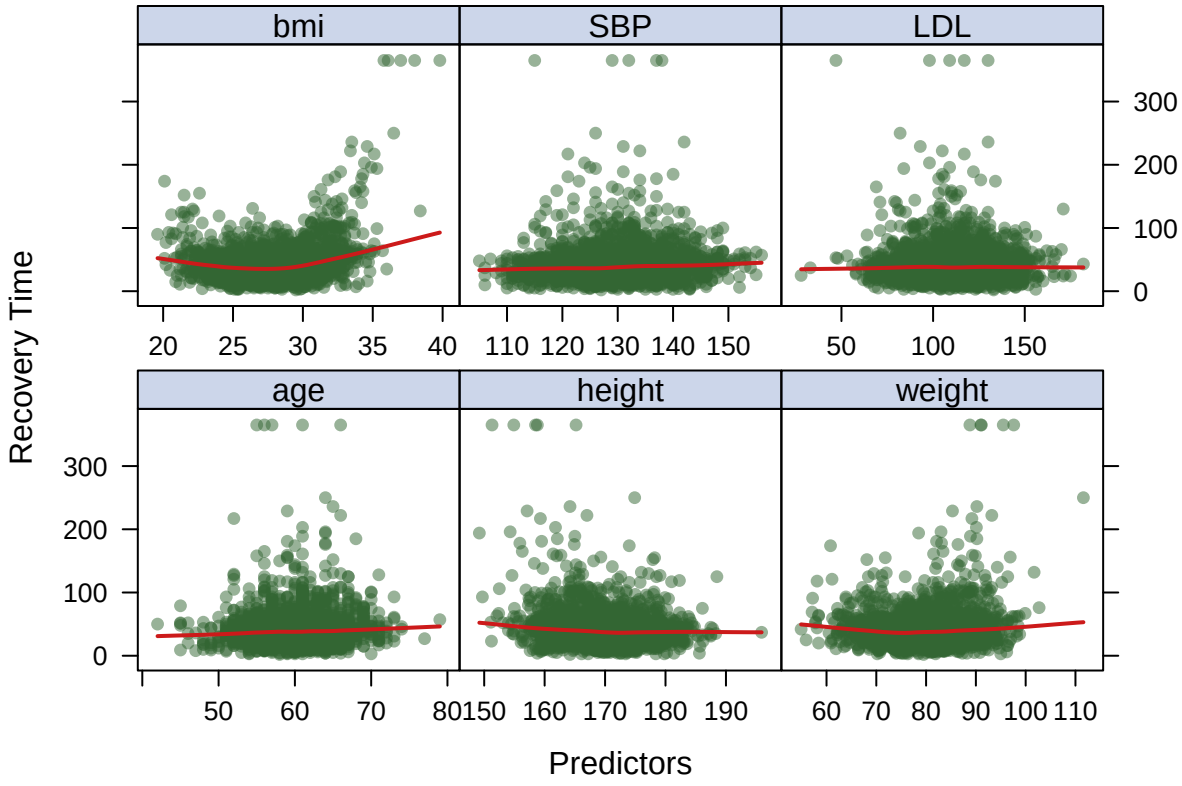
Correlation plot

```
corr_dat = covid_dat[rowTrain,] %>%
  dplyr::select('age', 'height', 'weight', 'bmi', 'SBP', 'LDL')
corrplot(cor(corr_dat), method = "circle", type = "full")
```

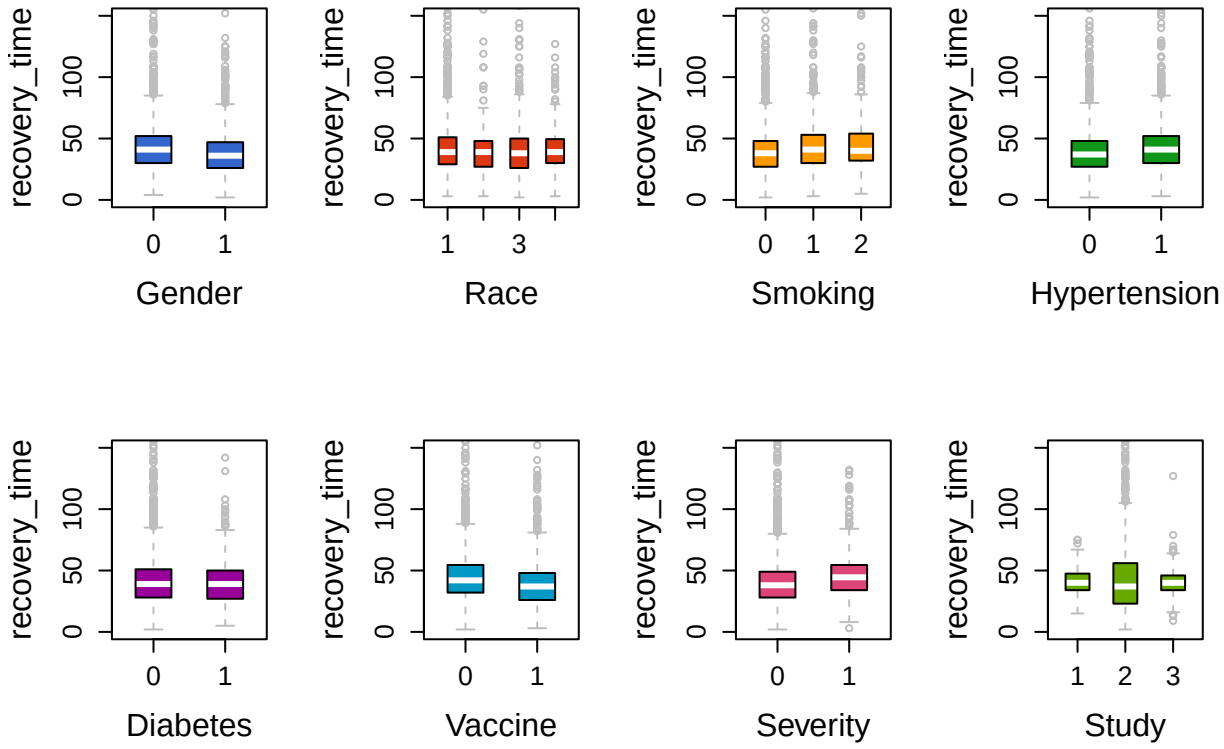


Feature plot

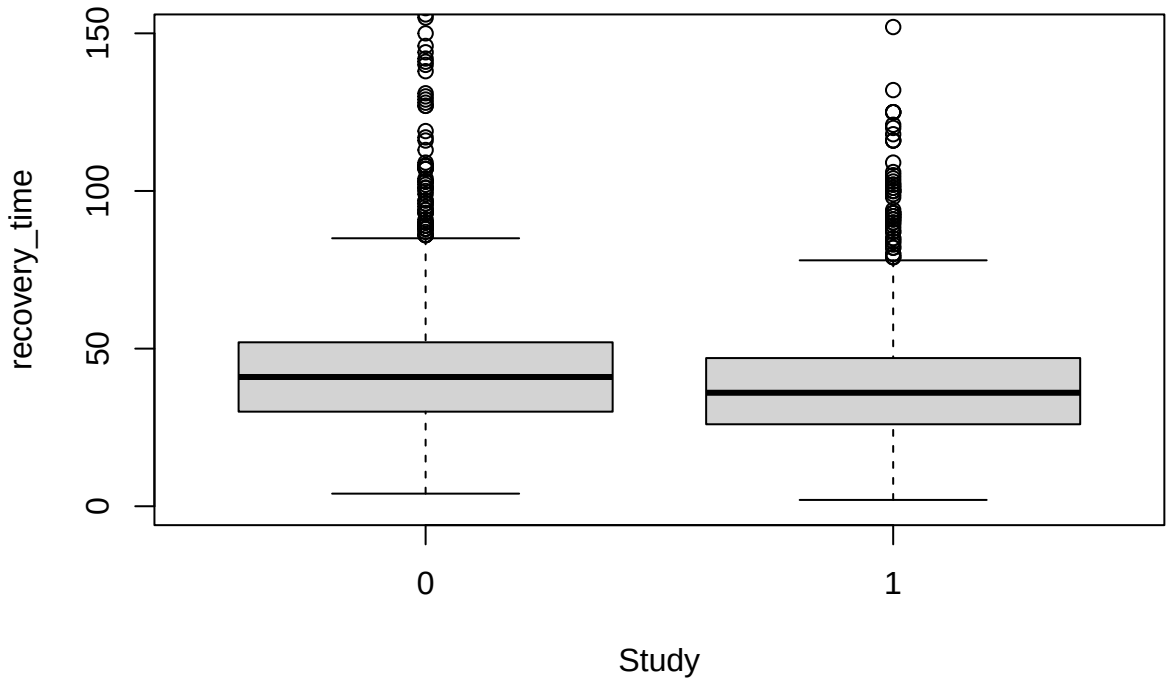
```
vis_trdat = trainData %>%
  dplyr::select('age', 'height', 'weight', 'bmi', 'SBP', 'LDL', 'recovery_time')

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 = vis_trdat[, 1:6],
            y = vis_trdat[, 7],
            plot = "scatter",
            span = 0.5,
            labels = c("Predictors", "Recovery Time"),
            type = c("p", "smooth"))
```


bmi
SBP
LDL
age
height
weight
Recovery Time
Predictors
0
100
200
300
20
25
30
35
40
110
120
130
140
150
50
100
150
50
60
70
80
150
160
170
180
190
60
70
80
90
100
110

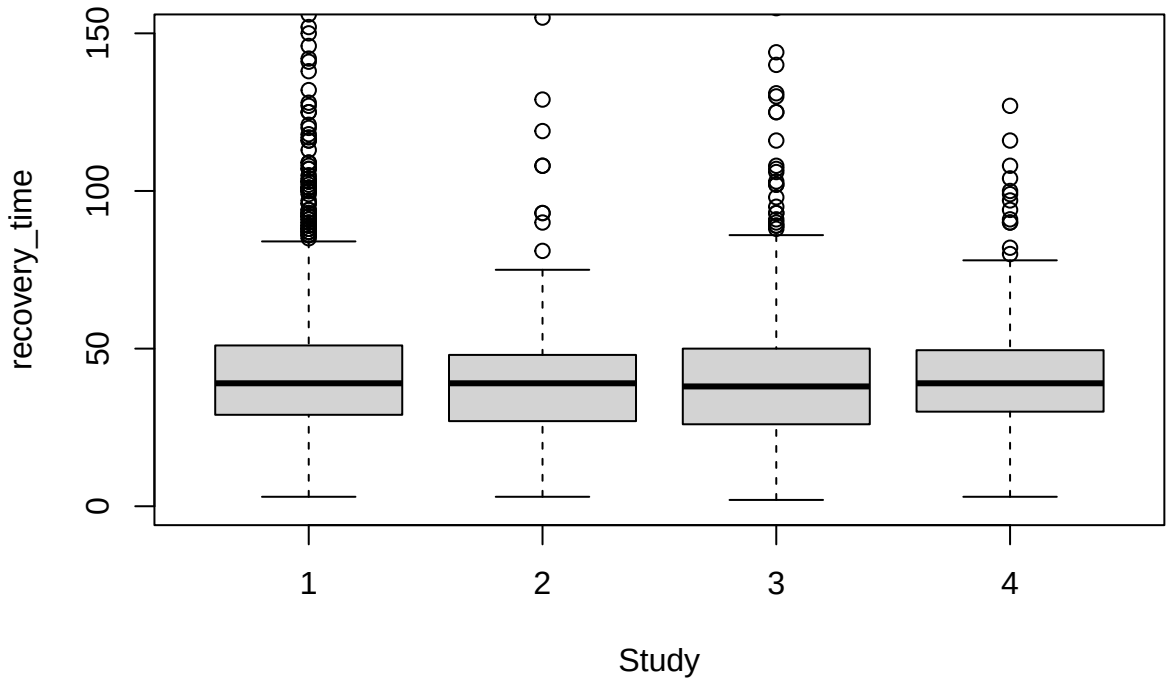
Boxplot



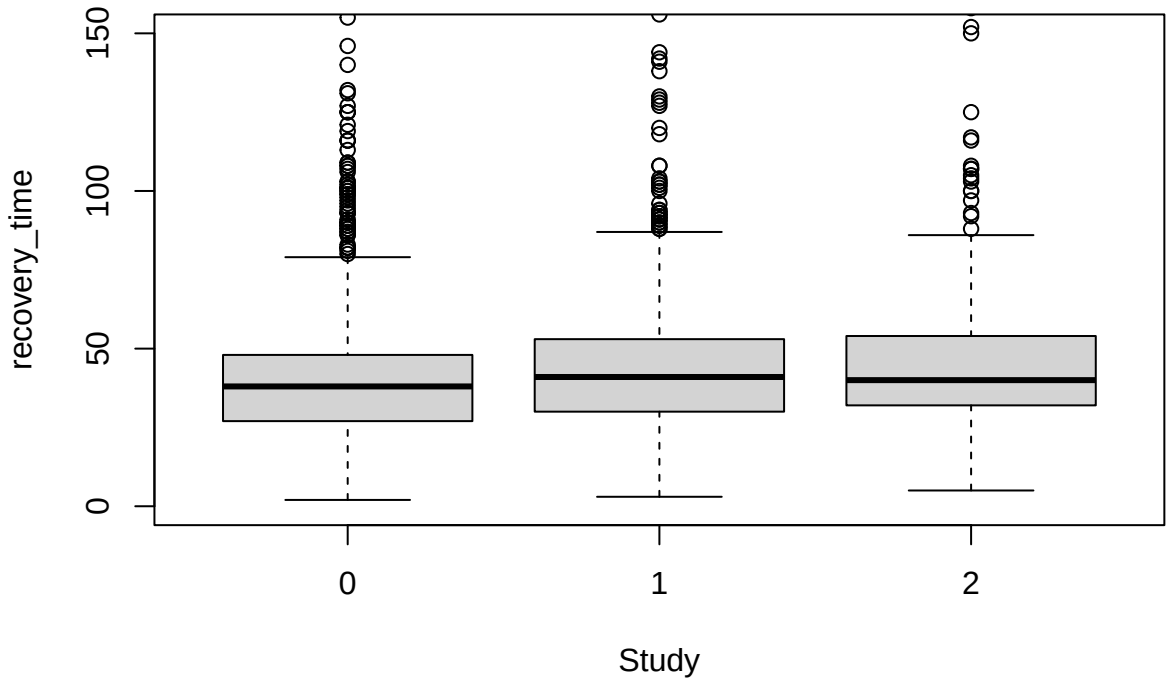

```
bp_gender = boxplot(recovery_time ~ gender, data = trainData, xlab = "Study", ylim = c(0, 150))
```



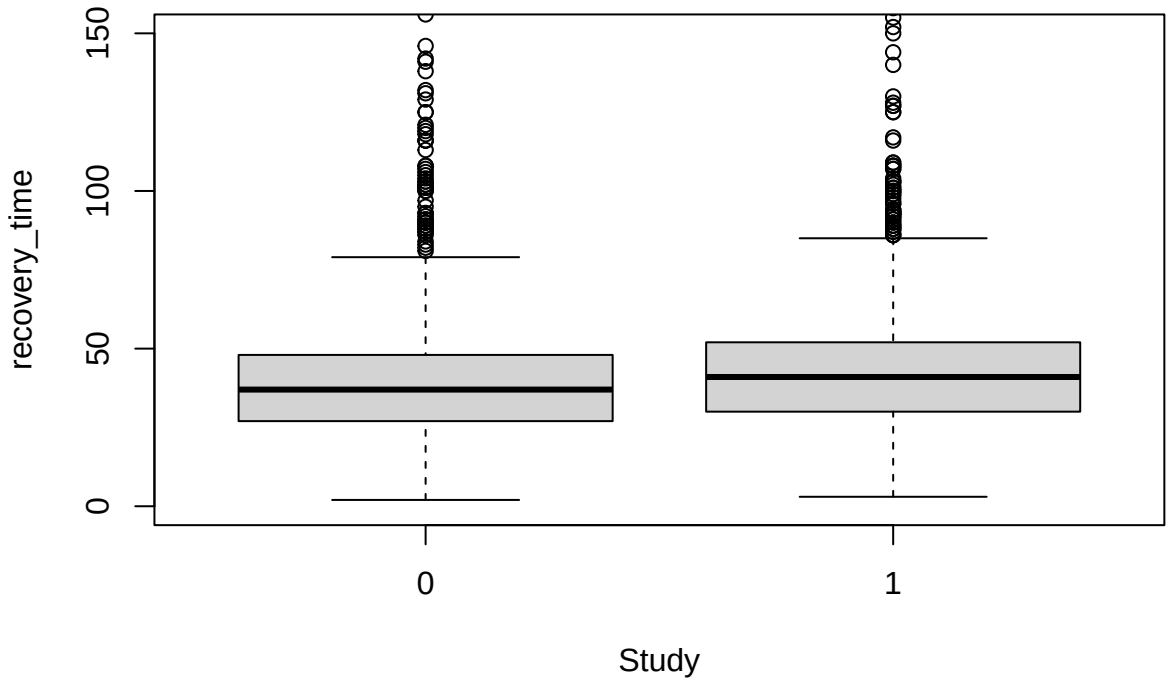

```
bp_race = boxplot(recovery_time ~ race, data = trainData, xlab = "Study", ylim = c(0, 150))
```



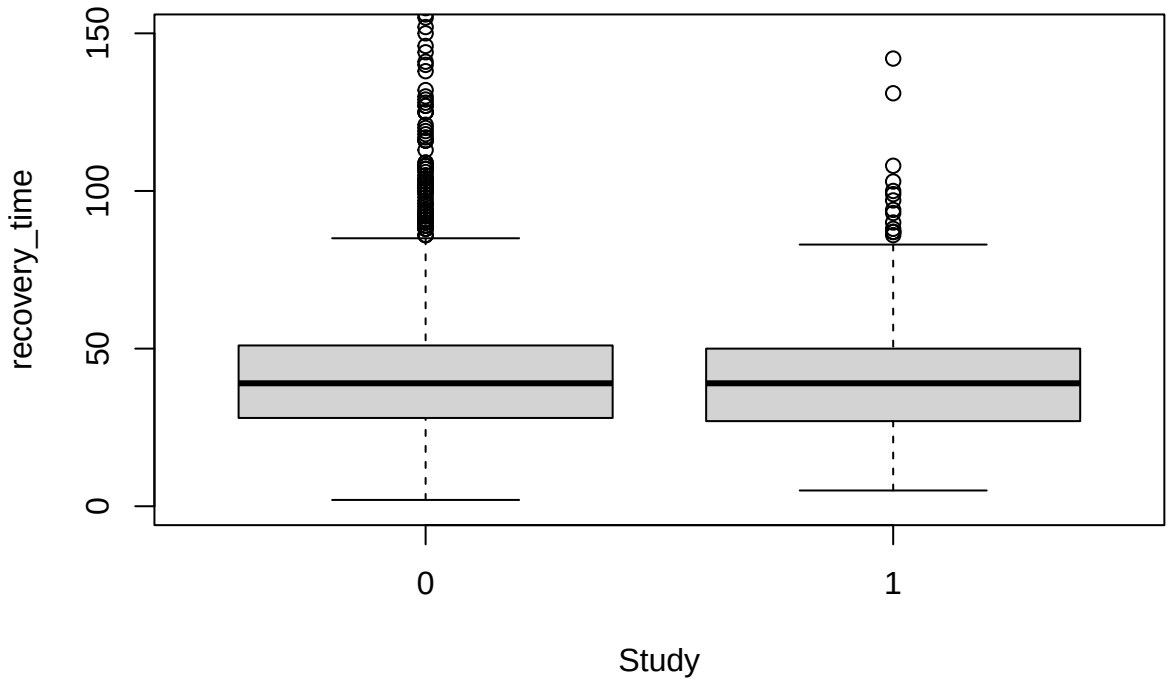

```
bp_smoking = boxplot(recovery_time ~ smoking, data = trainData, xlab = "Study", ylim = c(0, 150))
```



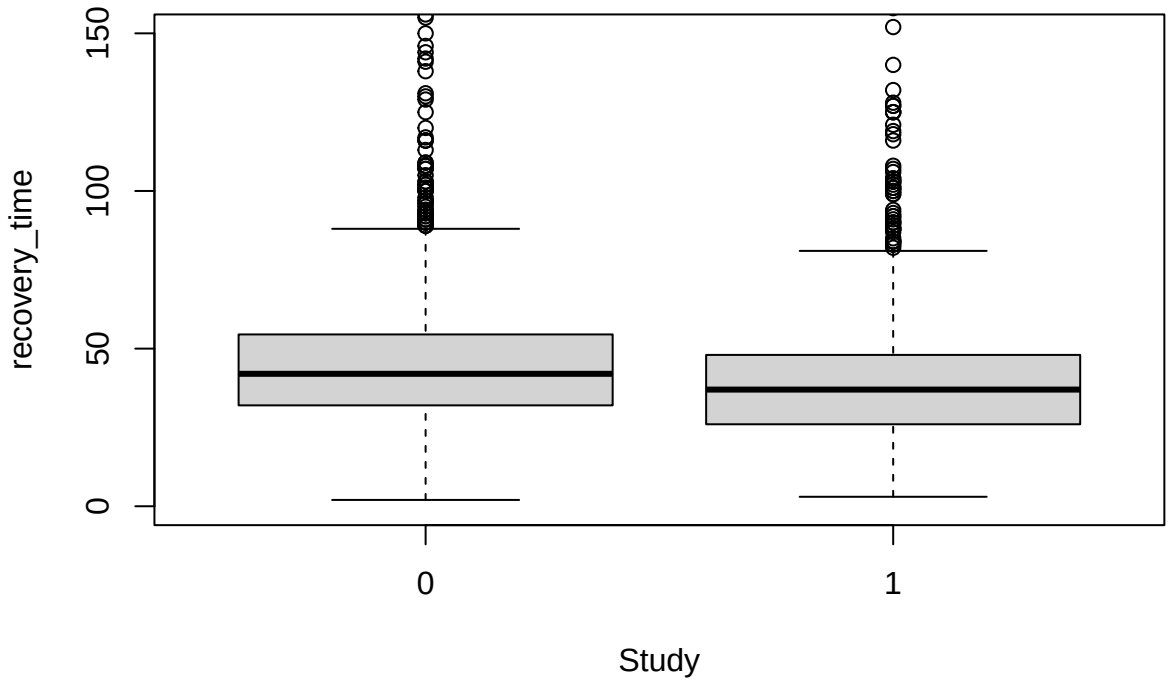

```
bp_hypertension = boxplot(recovery_time ~ hypertension, data = trainData, xlab = "Study", ylim = c(0, 15
```



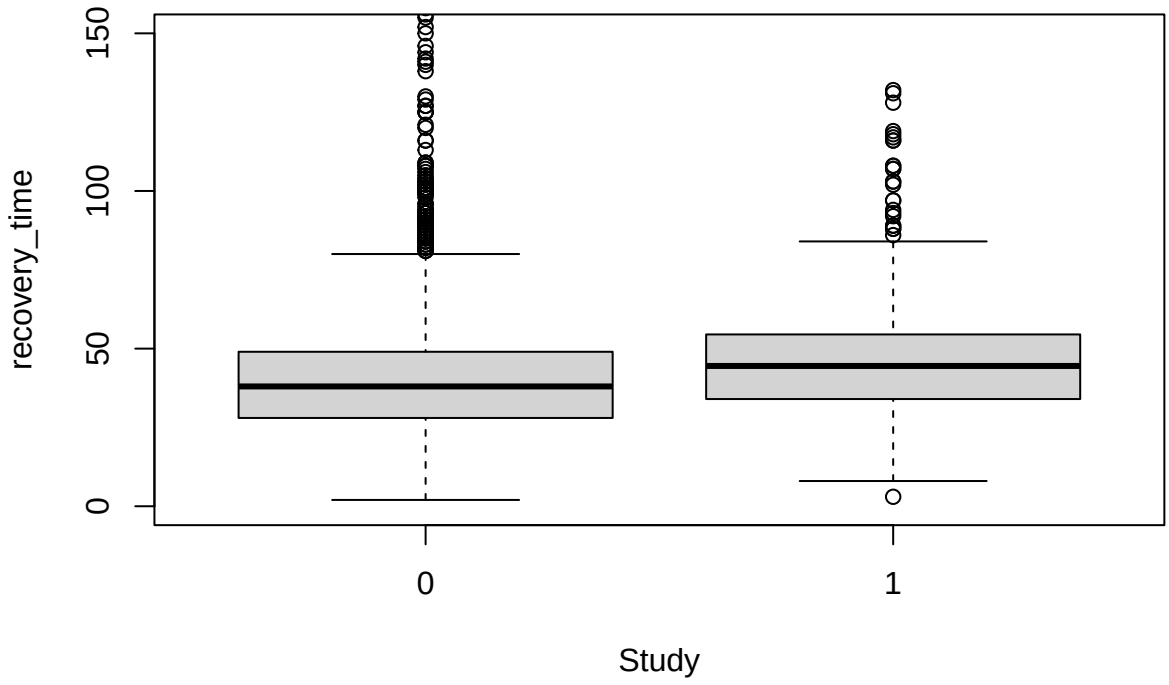

```
bp_diabetes = boxplot(recovery_time ~ diabetes, data = trainData, xlab = "Study", ylim = c(0, 150))
```



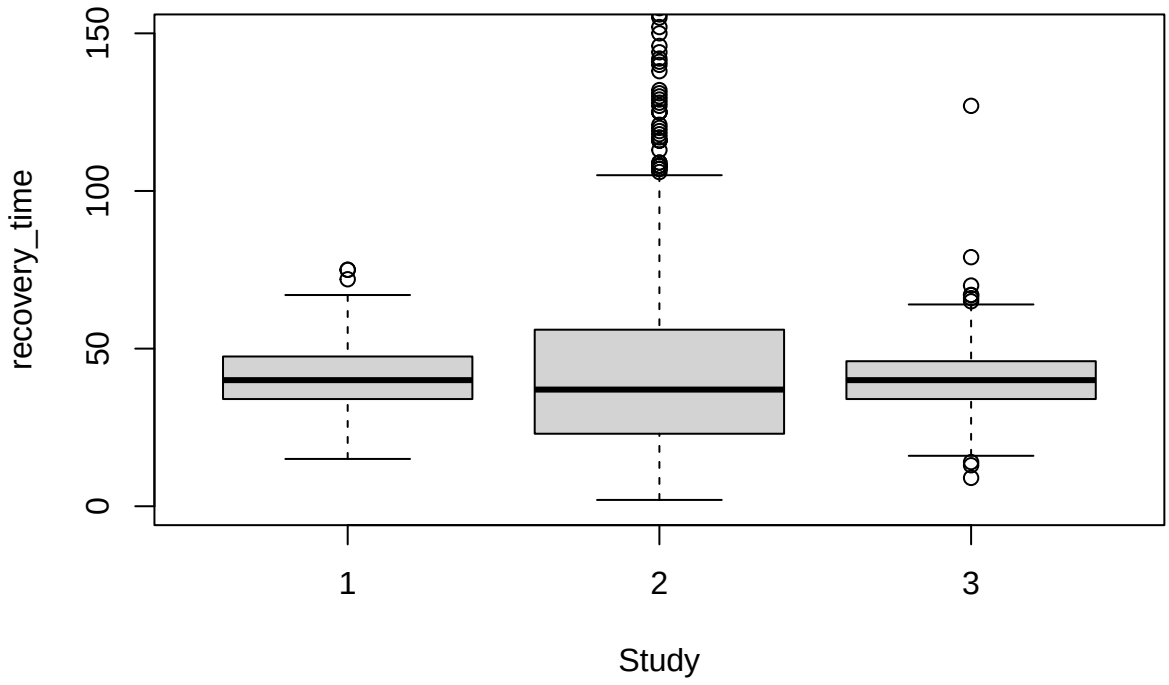

```
bp_vaccine = boxplot(recovery_time ~ vaccine, data = trainData, xlab = "Study", ylim = c(0, 150))
```




```
bp_severity = boxplot(recovery_time ~ severity, data = trainData, xlab = "Study", ylim = c(0, 150))
```




```
bp_study = boxplot(recovery_time ~ study, data = trainData, xlab = "Study", ylim = c(0, 150))
```



Model training

GAM

```
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
```

```
gam.fit$finalModel
```

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

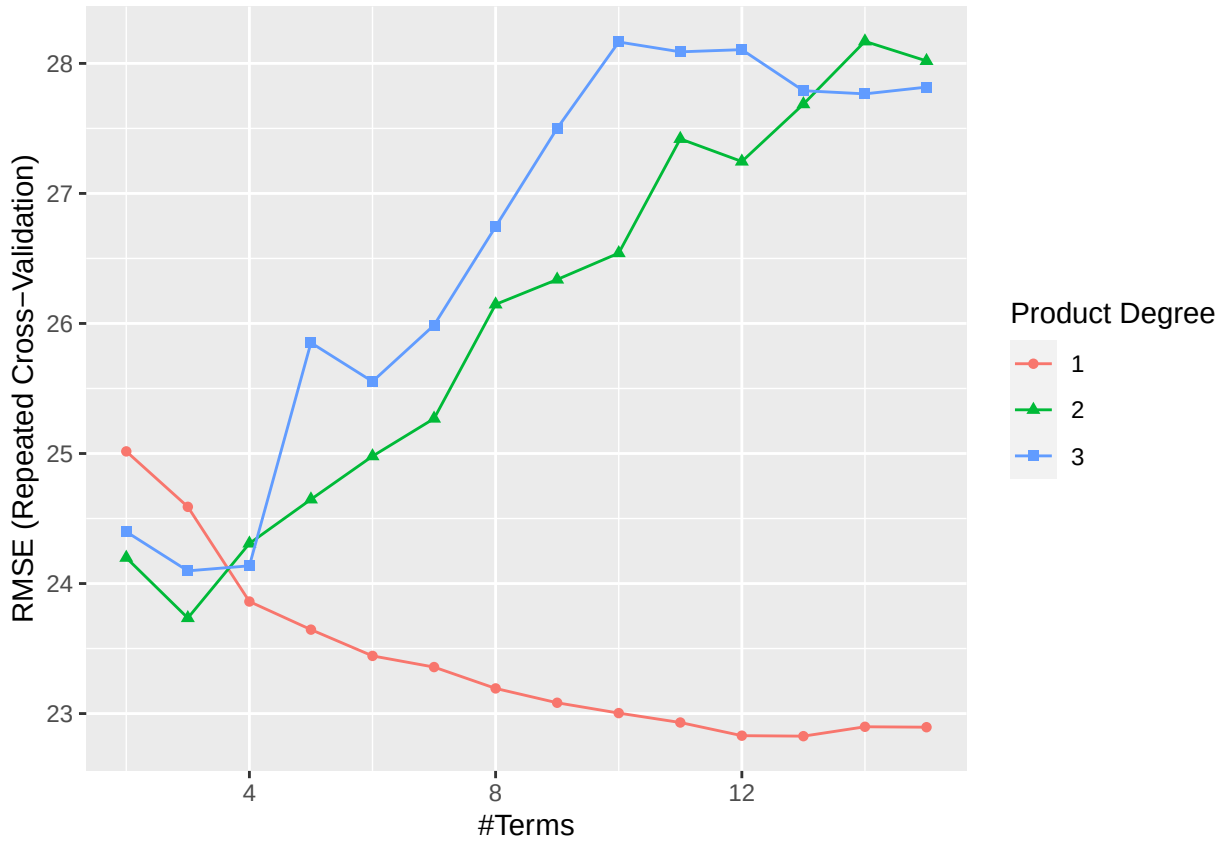
```
## Formula:
## .outcome ~ gender1 + race2 + race3 + race4 + smoking1 + smoking2 +
##     hypertension1 + diabetes1 + vaccine1 + severity1 + study2 +
##     study3 + s(age) + s(SBP) + s(LDL) + s(bmi) + s(height) +
##     s(weight)
##
## Estimated degrees of freedom:
## 0.765 0.000 0.000 7.872 4.491 0.430  total = 26.56
##
## GCV score: 495.0358
```

```
# test error
pred.gam = predict(gam.fit, newdata = covid_dat2[-rowTrain,])
gam.RMSE = mean((pred.gam - covid_dat$recovery_time[-rowTrain])^2)
gam.RMSE
```

```
## [1] 502.8183
```

MARS

```
mars_grid = expand.grid(degree = 1:3,
                        nprune = 2:15)

set.seed(2)
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
```

```
mars.fit$finalModel
```

```
## 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
```

```
# test error
pred.mars = predict(mars.fit, newdata = covid_dat2[-rowTrain,])
mars.RMSE = mean((pred.mars - covid_dat$recovery_time[-rowTrain])^2)
mars.RMSE
```

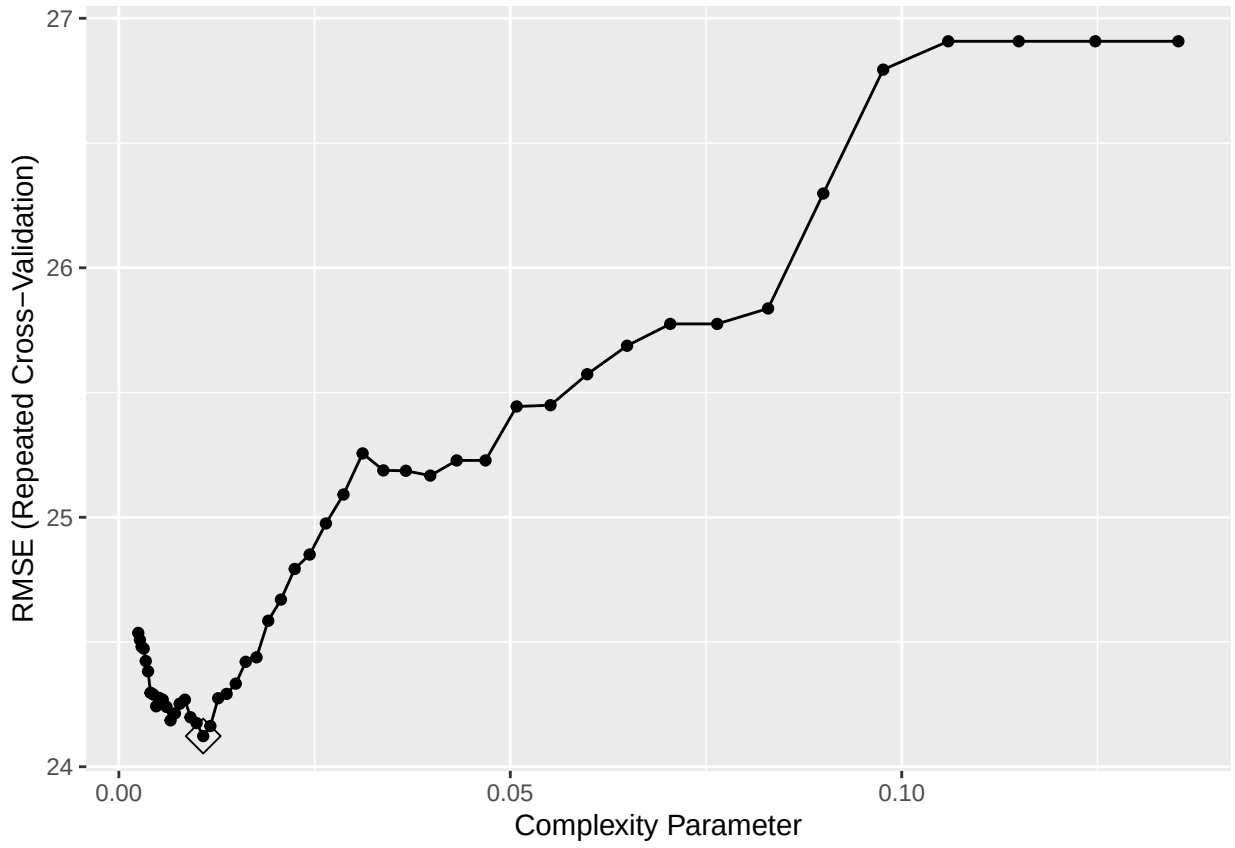
```
## [1] 505.8689
```

Regression tree

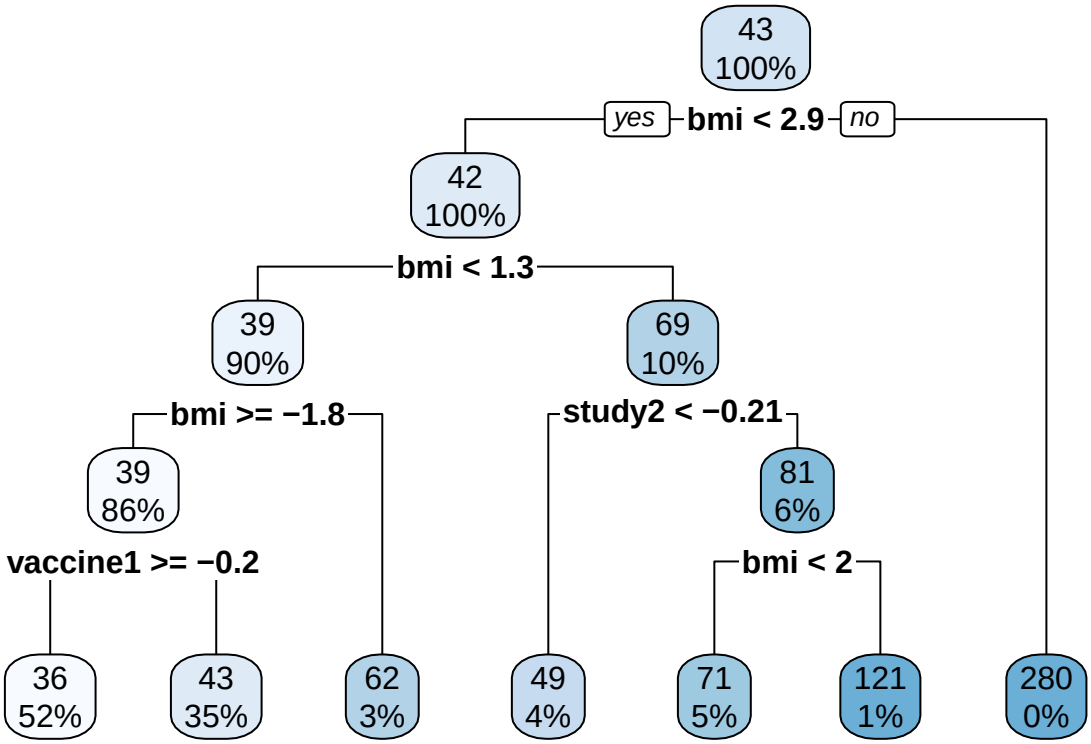
```
set.seed(2)

#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,
                  preProcess = c("center", "scale"))

rpart.fit$bestTune
```

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

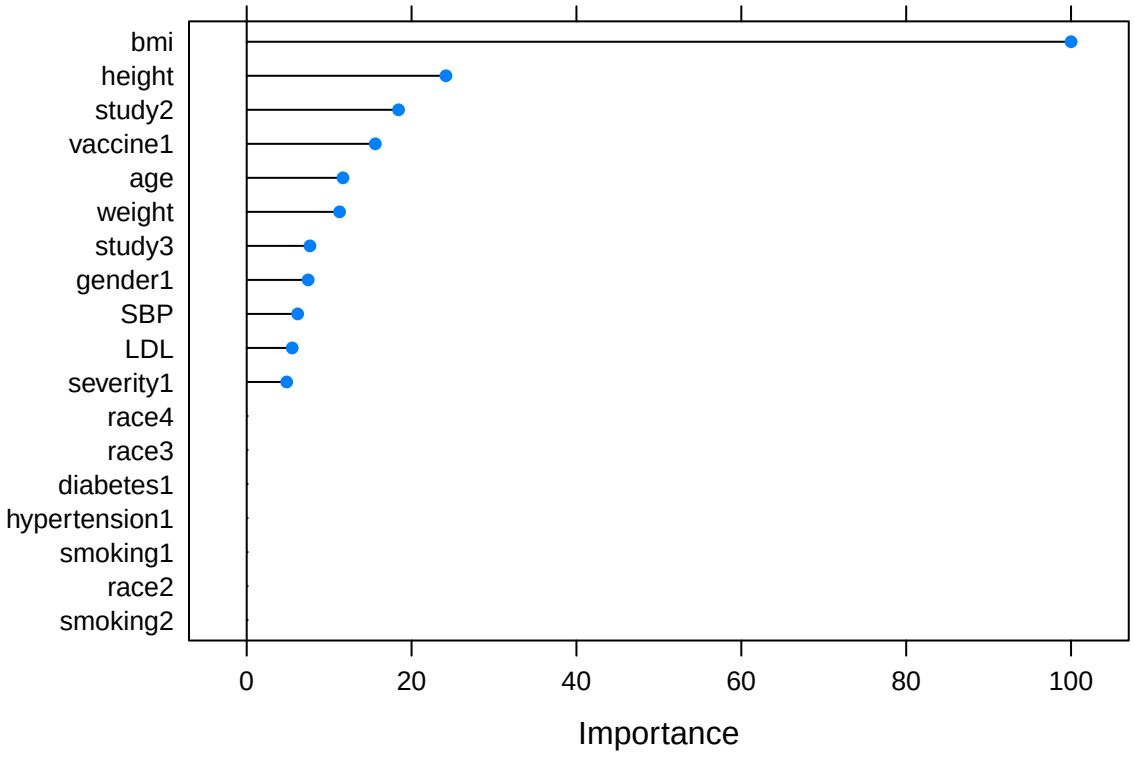
```
#plot of the tree
ggplot(rpart.fit, highlight = TRUE)
```




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



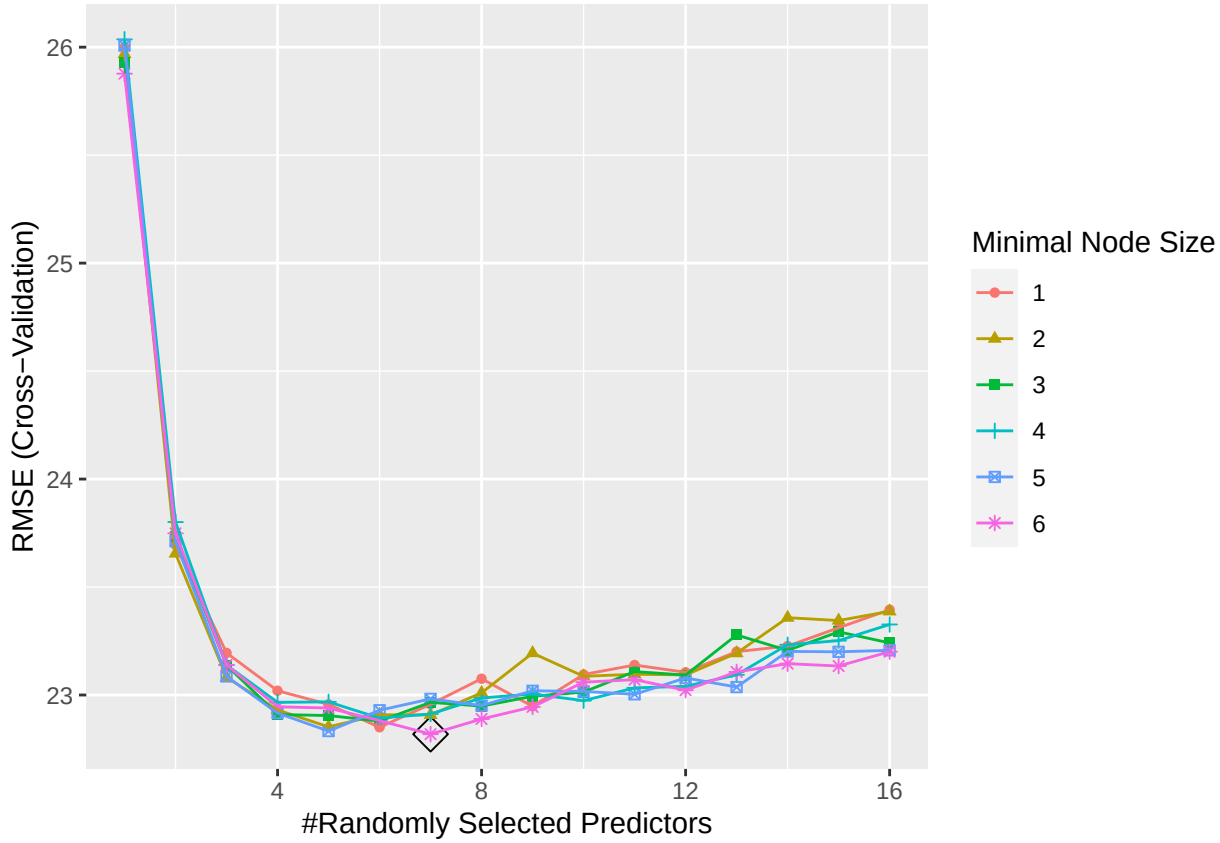

```
# Report the variable importance
plot(varImp(rpart.fit, scale = TRUE))
```




```
## rpart does not have RMSE
```

Random Forest

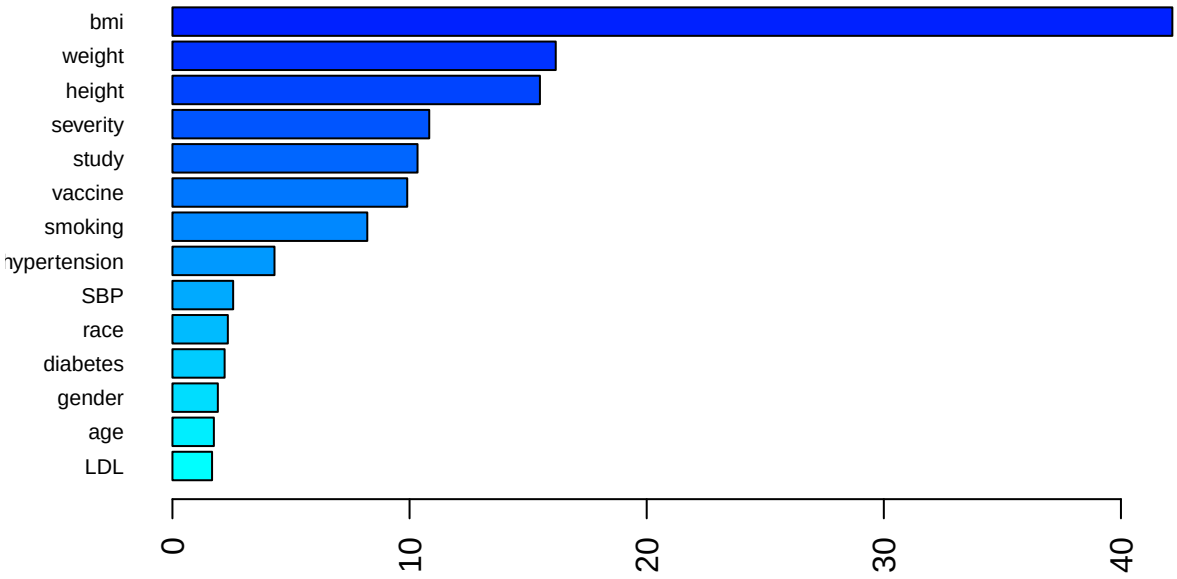
```
ctrlrf = trainControl(method = "cv")

# 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 = ctrlrf)

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




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

```
##    mtry splitrule min.node.size
## 42    7  variance             6
```

```
# Variable importance
rf2.final.per = ranger(recovery_time ~ .,
                covid_dat[rowTrain,],
                mtry = rf.fit$bestTune[[1]],
                splitrule = "variance",
                min.node.size = rf.fit$bestTune[[3]],
                importance = "permutation",
                scale.permutation.importance = TRUE)

barplot(sort(ranger::importance(rf2.final.per), decreasing = FALSE),
        las = 2, horiz = TRUE, cex.names = 0.7,
        col = colorRampPalette(colors = c("cyan","blue"))(16))
```




```
# test error

set.seed(2)
rffit_pred = predict(rf.fit, newdata = covid_dat[-rowTrain,])
rffit.RMSE = RMSE(rffit_pred, covid_dat$recovery_time[-rowTrain])
rffit.RMSE
```

```
## [1] 22.80351
```

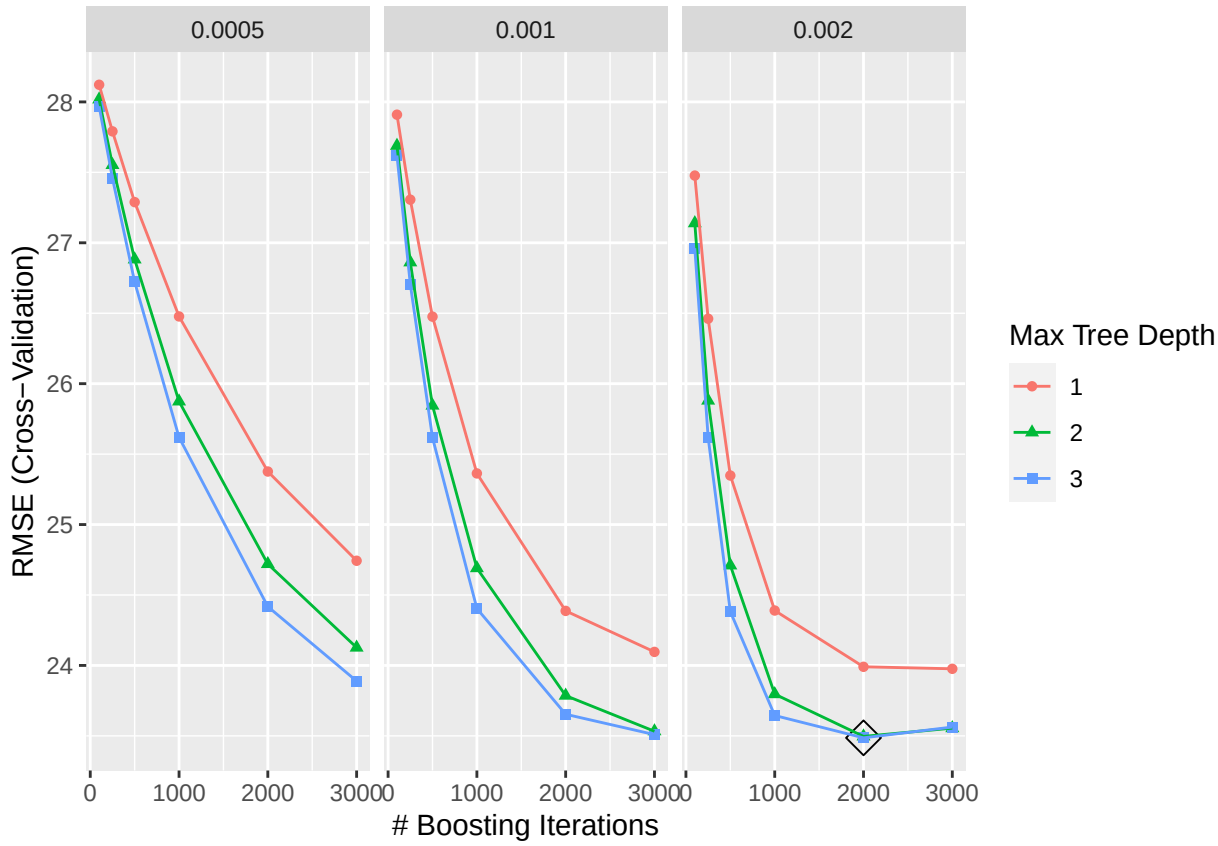
Boosting

```
ctrlboost = trainControl(method = "cv")

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 = ctrlboost,
                method = "gbm",
```

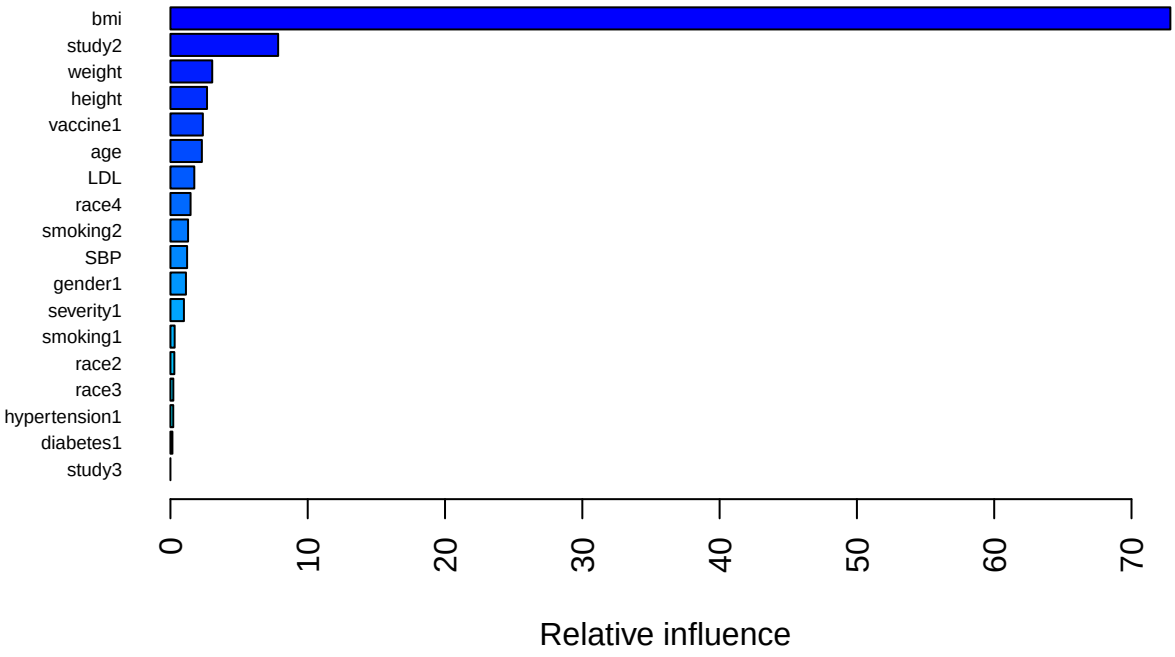
```
                verbose = FALSE)

gbm.fit$bestTune
```

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

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




```
# Report the variable importance
summary(gbm.fit$finalModel, las = 2, cBars = 19, cex.names = 0.6)
```




```
##                                var    rel.inf
## bmi                            bmi 72.8306899
## study2                      study2  7.8446965
## weight                      weight  3.0431561
## height                      height  2.6654281
## vaccine1                  vaccine1  2.3613918
## age                            age  2.2890618
## LDL                            LDL  1.7390874
## race4                        race4  1.4669575
## smoking2                  smoking2  1.2842068
## SBP                            SBP  1.2143366
## gender1                    gender1  1.1307862
## severity1                severity1  0.9822185
## smoking1                  smoking1  0.3068305
## race2                        race2  0.2838777
## race3                        race3  0.2095629
## hypertension1 hypertension1  0.2037763
## diabetes1                diabetes1  0.1439356
## study3                      study3  0.0000000
```

```
#test error
set.seed(2)
pred.gbm = predict(gbm.fit, newdata = covid_dat[-rowTrain,])
gbm.RMSE = RMSE(pred.gbm, covid_dat$recovery_time[-rowTrain])
gbm.RMSE
```

```
## [1] 22.25067
```

Model selection