Regression Tree - Framingham Heart Study data

Sangsoo Park (D)

April 12, 2014

1 What is Regression Tree?

Regression tree analysis has been used to understand underlying structures of data to predict a continuous outcome variable. The analysis divides data into sub-groups using binary branches, which indicates partioning of the outcome variable based predictor variables. There are two benefits using the analysis. The first one is that results from the analysis are simple to understand. Another one is that we don't need to assume linearlity relationship between ouctcome and predictor variables that linear regression model has. In the analysis, splits are determined by minimum residual sum of squares.That is, splits can be selected by minimization of within group sum of squares during maximization of between group sum of squares (ANOVA). This process is repeated using meaningful predictor variables that can greatly improve accuracy of the tree (least squares) until a minimum size of the end sub-groups is reached or no longer improvements in the accuracy of adding splits are showen.Thus, we need to have decision criteria for not only stopping criteria of the analysis but also having proper results without overfitting data. More details will be explained in the decision section.

2 Methods

2.1 Data filtering

Five categorical and six continuous variables were selected because of consistency between the other group members. The 11 variables were extracted from the original dataset and the new dataset (frm2) was used for regression tree analysis.

```
require(ggplot2)

## Loading required package:  ggplot2

require(RCurl)

## Loading required package:  RCurl
## Loading required package:  bitops

data <- getURL("https://raw.githubusercontent.com/arvindram12/Final-Project/master/frmgham2.csv",
    ssl.verifypeer = 0L, followlocation = 1L)
writeLines(data, "framingham.csv")
frm <- read.csv("framingham.csv")
frm$RANDID <- as.factor(frm$RANDID)
frm1 <- frm[which(!duplicated(frm$RANDID)), ]
frm2 <- data.frame(as.factor(frm1$SEX), as.factor(frm1$DIABETES), as.factor(frm1$BPMEDS),
    as.factor(frm1$CURSMOKE), as.factor(frm1$educ), frm1$AGE, frm1$TOTCHO, frm1$BMI,
    frm1$GLUCOSE, frm1$HEARTRTE, frm1$SYSBP)
names(frm2) <- c("SEX", "DIABETES", "BPMEDS", "CURSMOKE", "educ", "AGE", "TOTCHOL",
    "BMI", "GLUCOSE", "HEARTRTE", "SYSBP")
```

2.2 Regression tree analysis

First, 10 variables were used to predict systolic blood pressure (SYSBP). The numbers at the terminal are mean systolic blood pressures of the number of observations falling in each terminal.To start, the initial cp value was set to 0.005, which might result in overfitting of the data. There were missing values in the outcome and predictor variables. If the outcome variable has missing values, all observations from the predictors were removed. However, outcome variable values were kept when one or more predictor variables have missing values.

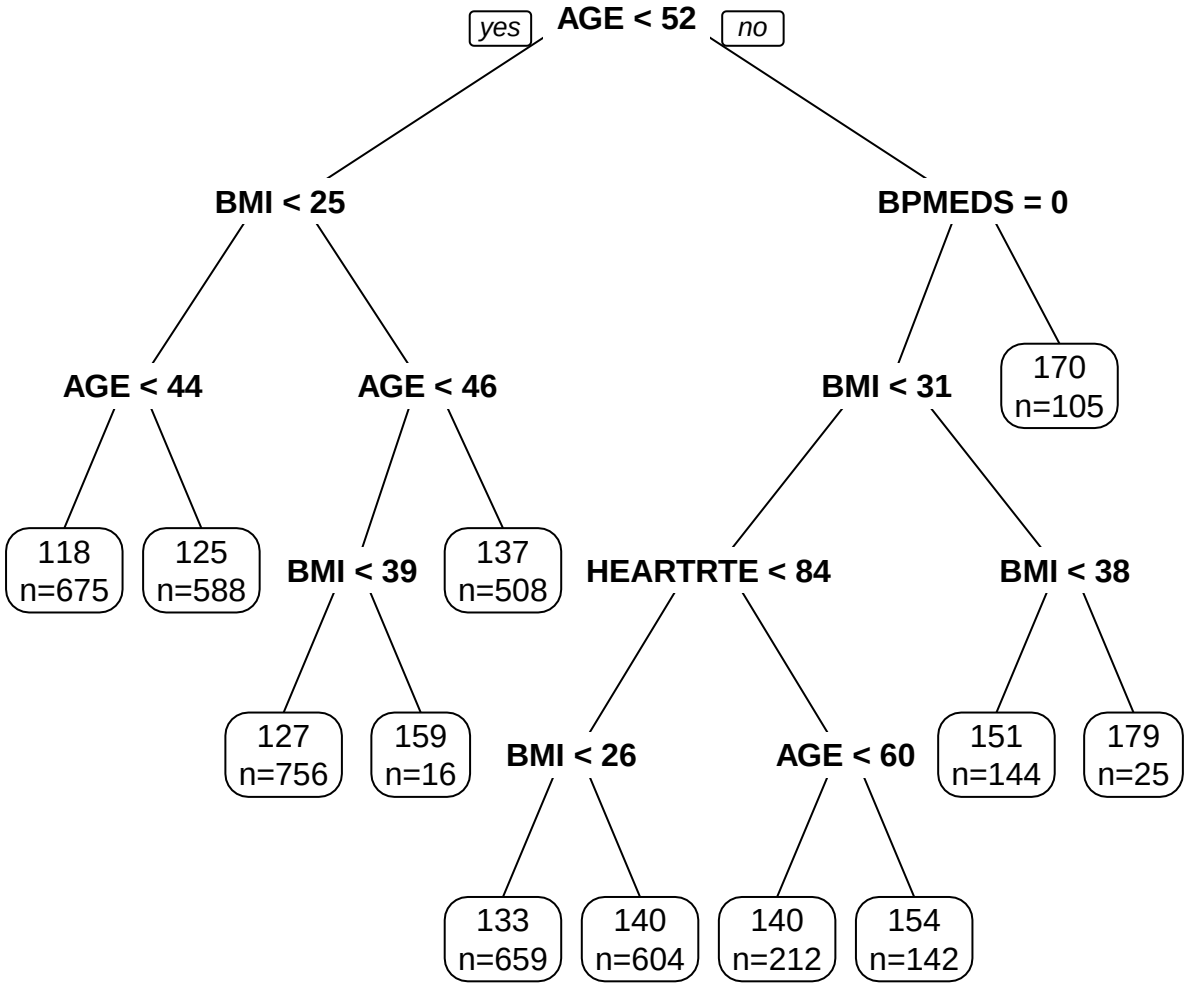
```r
require(rpart)
```

```
## Loading required package:  rpart
## Warning:  package 'rpart' was built under R version 3.0.3
```

```r
require(rpart.plot)
```

```
## Loading required package:  rpart.plot
## Warning:  package 'rpart.plot' was built under R version 3.0.3
```

```r
# 'anova' = when outcome variable is continuous with cp=0.005 na.action =
# default (explained above)
fit <- rpart(SYSBP ~ ., data = frm2, method = "anova", cp = 0.005)

# Visualize tree
rpart.plot(fit, digits = 2, extra = 1)
```



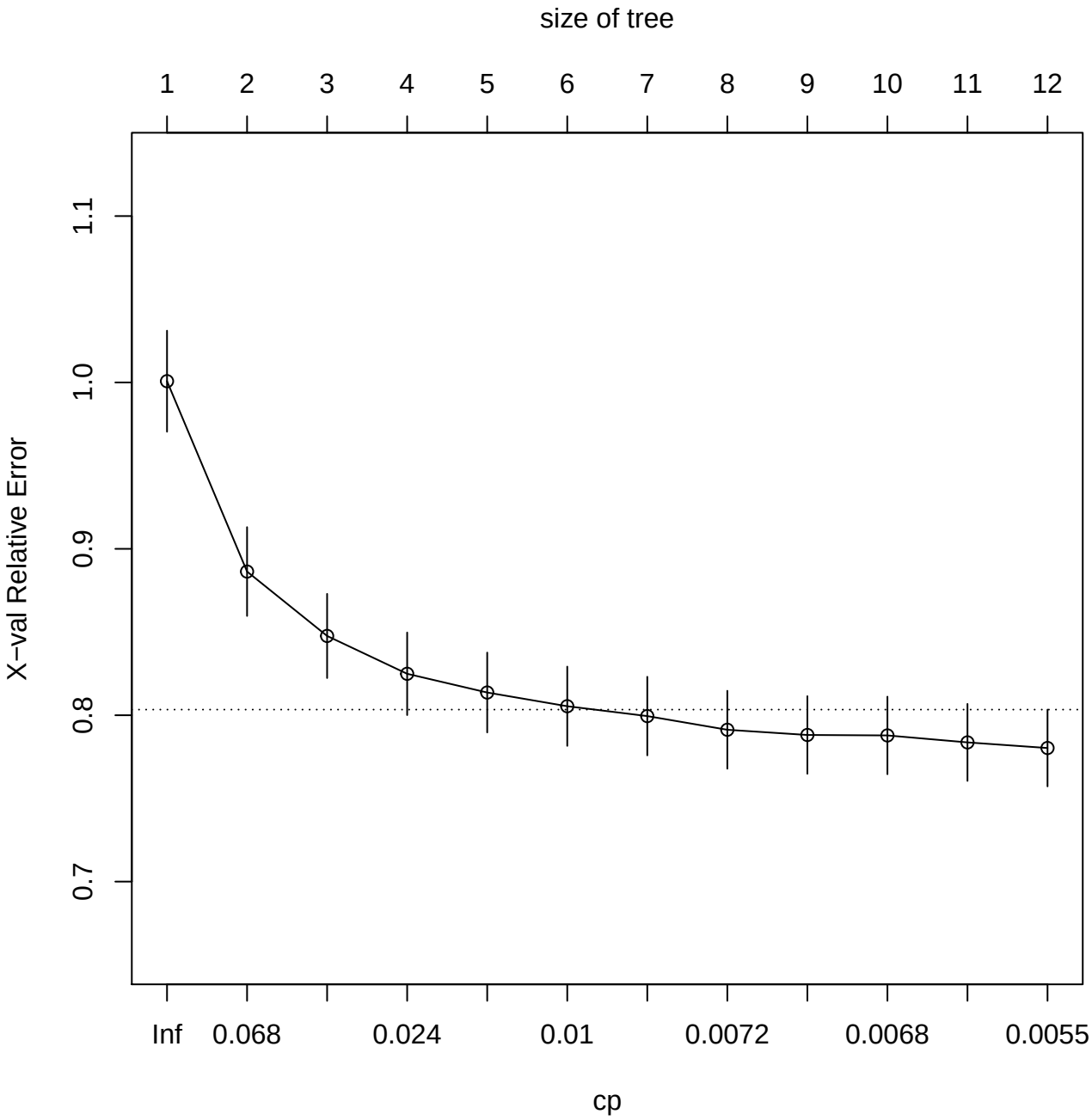
2.3 Validation of the result

Keep adding more splits results in increase in R-squared even though the added splits could just result from adding more splits. This is what we saw in the multiple regression model. To avoid this overfitting problem, we need decision criteria on the appropriate number of splits.

The cp indicates complexity parameter that allows us to decide the number of meaningful splits (cost of adding one additional split).To be specific, higher cp values lead to smaller number of splits. we can decide the number of splits where the overall R-squared value is not increased more by increase in cp values.

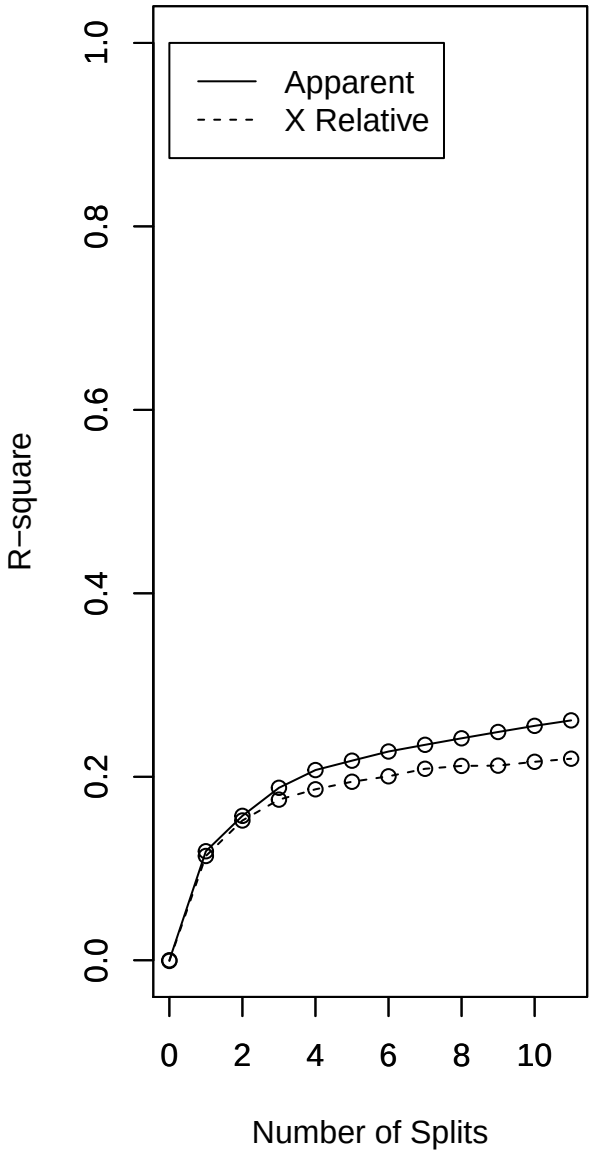
The minimum number of splits was defined by changes in X-val relative error as a function of cp values. After the size of tree is larger than 6, the cp values remain constant and there were no further improvements in the X-val relative error. Thus, six splits were decided here.

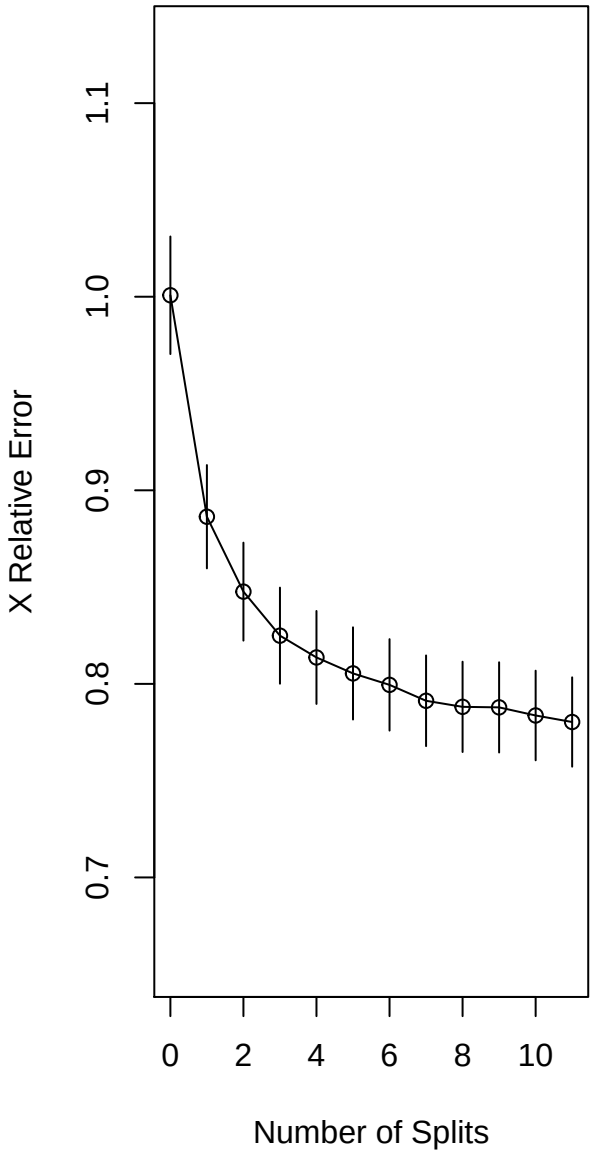
```
# Show results
printcp(fit)

##
## Regression tree:
## rpart(formula = SYSBP ~ ., data = frm2, method = "anova", cp = 0.005)
##
## Variables actually used in tree construction:
## [1] AGE      BMI      BPMEDS   HEARTRTE
##
## Root node error: 2228593/4434 = 503
##
## n= 4434
##
##         CP nsplit rel error xerror  xstd
## 1  0.1188      0      1.00   1.00 0.030
## 2  0.0386      1      0.88   0.89 0.027
## 3  0.0306      2      0.84   0.85 0.025
## 4  0.0193      3      0.81   0.82 0.025
## 5  0.0102      4      0.79   0.81 0.024
## 6  0.0100      5      0.78   0.81 0.024
## 7  0.0073      6      0.77   0.80 0.024
## 8  0.0071      7      0.77   0.79 0.023
## 9  0.0069      8      0.76   0.79 0.023
## 10 0.0067      9      0.75   0.79 0.023
## 11 0.0060     10      0.74   0.78 0.023
## 12 0.0050     11      0.74   0.78 0.023

# Cross-Validation results graphics.off()
plotcp(fit)
```




```
par(mfrow = c(1, 2))
rsq.rpart(fit)

##
## Regression tree:
## rpart(formula = SYSBP ~ ., data = frm2, method = "anova", cp = 0.005)
##
## Variables actually used in tree construction:
## [1] AGE      BMI      BPMEDS   HEARTRTE
##
## Root node error: 2228593/4434 = 503
##
## n= 4434
##
##        CP nsplit rel error xerror  xstd
## 1  0.1188      0      1.00   1.00 0.030
## 2  0.0386      1      0.88   0.89 0.027
```

```
## 3  0.0306      2      0.84   0.85 0.025
## 4  0.0193      3      0.81   0.82 0.025
## 5  0.0102      4      0.79   0.81 0.024
## 6  0.0100      5      0.78   0.81 0.024
## 7  0.0073      6      0.77   0.80 0.024
## 8  0.0071      7      0.77   0.79 0.023
## 9  0.0069      8      0.76   0.79 0.023
## 10 0.0067      9      0.75   0.79 0.023
## 11 0.0060     10      0.74   0.78 0.023
## 12 0.0050     11      0.74   0.78 0.023
```



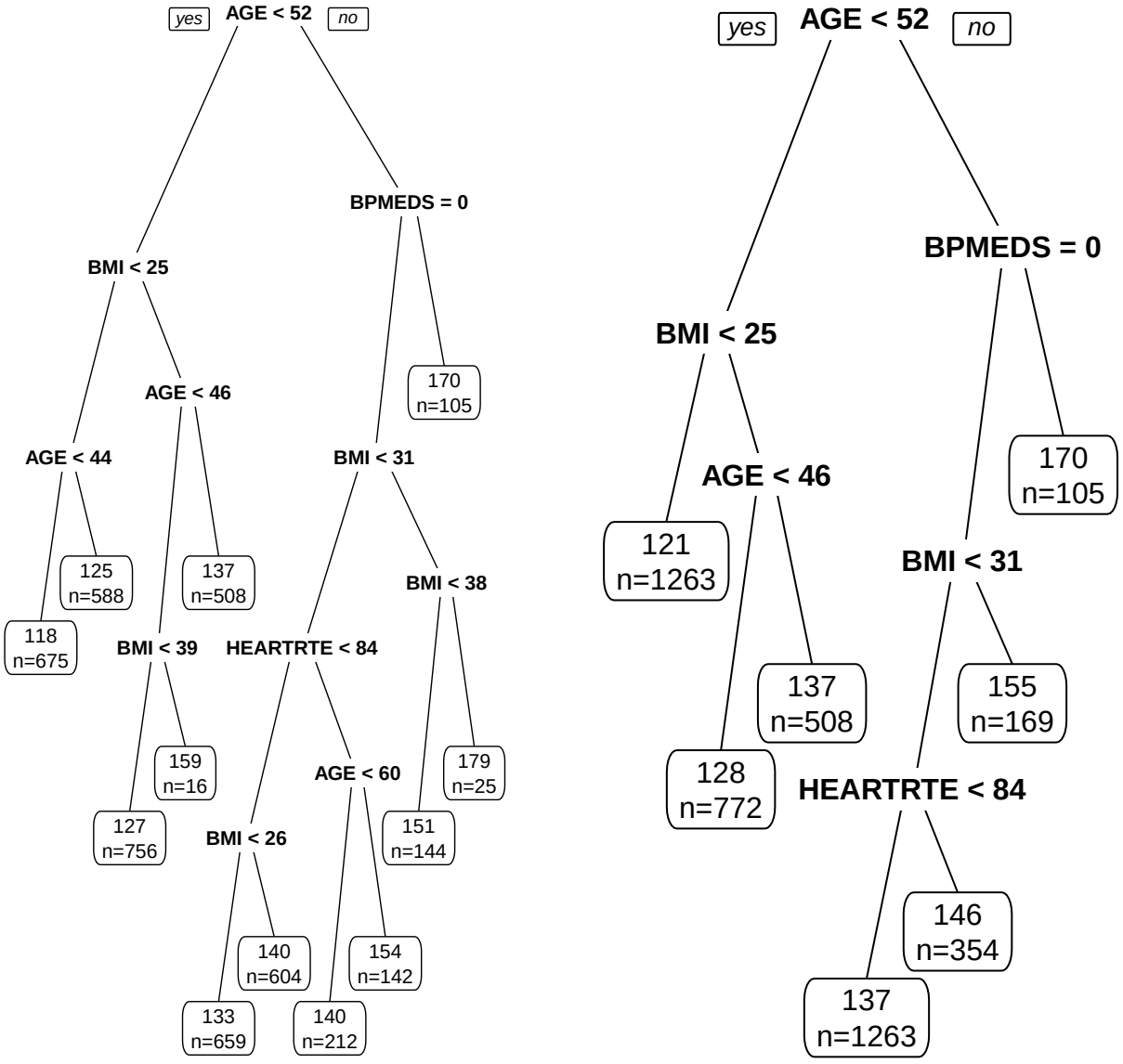



```
# Based on cp value, prune the tree to avoid overfitting from printcp(fit)
pr_fit <- prune(fit, cp = 0.01)
```

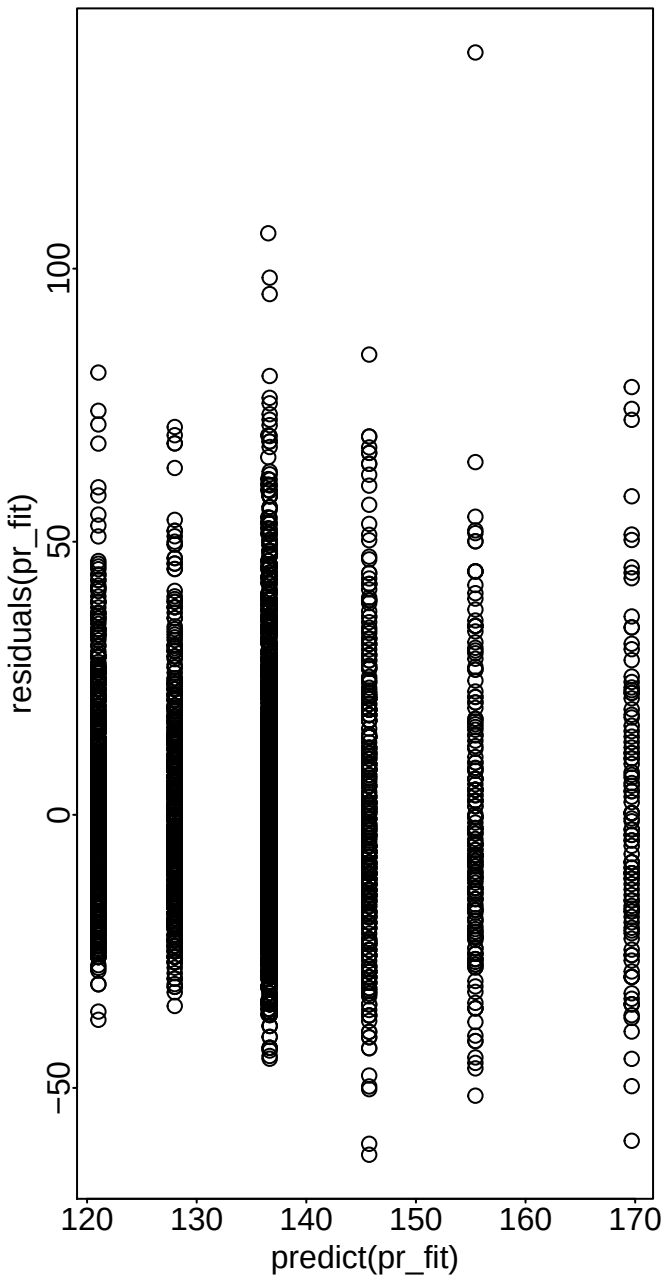
2.4 Comparison

Comparison results from the original tree and the pruned tree.

```
par(mfrow = c(1, 2), oma = c(1, 1, 0, 0), mar = c(2, 2, 1, 0), tcl = -0.1, mgp = c(1,
    0, 0))
rpart.plot(fit, digits = 2, extra = 1)
rpart.plot(pr_fit, digits = 2, extra = 1)
```




```
# Show Summary of the pruned tree summary(pr_fit) Show residuals
summary(residuals(pr_fit))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -62.20  -13.00   -2.66    0.00    9.97  140.00

plot(predict(pr_fit), residuals(pr_fit))
```



2.5 Multiple linear regression

Comparison results from regression tree

```
mlr <- lm(SYSBP ~ AGE + BMI + BPMEDS + HEARTRTE, data = frm2)
summary(mlr)

##
## Call:
## lm(formula = SYSBP ~ AGE + BMI + BPMEDS + HEARTRTE, data = frm2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -66.65 -12.28  -2.22   9.86 127.85
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  29.9361     2.8586    10.5   <2e-16 ***
## AGE           0.8723     0.0333    26.2   <2e-16 ***
## BMI           1.3829     0.0701    19.7   <2e-16 ***
## BPMEDS1      24.2675     1.6093    15.1   <2e-16 ***
## HEARTRTE      0.3011     0.0234    12.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.7 on 4348 degrees of freedom
##   (81 observations deleted due to missingness)
## Multiple R-squared:   0.3,Adjusted R-squared:  0.299
## F-statistic:  466 on 4 and 4348 DF,  p-value: <2e-16
```

```
MSE <- sum(mlr$residuals^2)/4348
MSE
```

```
## [1] 349.1
```