BodyFat Random Forest and CART

jdt

2/16/2021

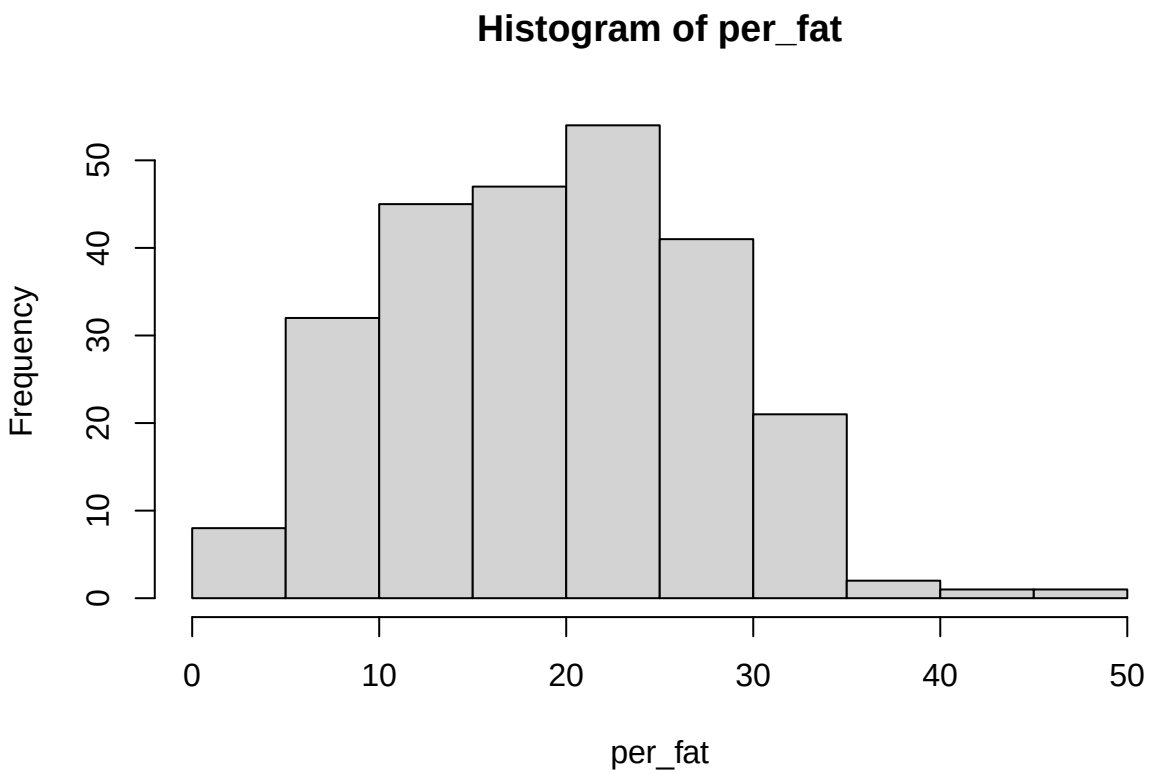
```
  # clear the environment and set seed
rm(list = ls())
set.seed(123)
```

Read Body Fat Data

```
library(foreign)
# this data came from SASHELP.BWEIGHT
bfat = read.csv('bodyfat.csv', header = TRUE, fileEncoding = 'UTF-8-BOM')
```

Describe Dependent Variable

```
per_fat = bfat$per_fat              #per_fat is continuous
hist(per_fat)
```




```
summary(per_fat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   12.47   19.20   19.15   25.30   47.50
```

```r
# Create Binary variable for Crime
# 24 was my choice
high_fat = factor((per_fat > 24))   #high_fat is discrete
```

```r
options("repos" = c(CRAN = "https://cran.rstudio.com"))

if (!require("randomForest")) install.packages("randomForest", dep=TRUE)
if (!require("dplyr")) install.packages("dplyr", dep=TRUE)
if (!require("ggpubr")) install.packages("ggpubr", dep=TRUE)
if (!require("lme4")) install.packages("lme4", dep=TRUE)
if (!require("rpart")) install.packages("rpart", dep=TRUE)
if (!require("rpart.plot")) install.packages("rpart.plot", dep=TRUE)

if (!require("partykit")) install.packages("partykit", dep=TRUE)
```

#Perform Regression

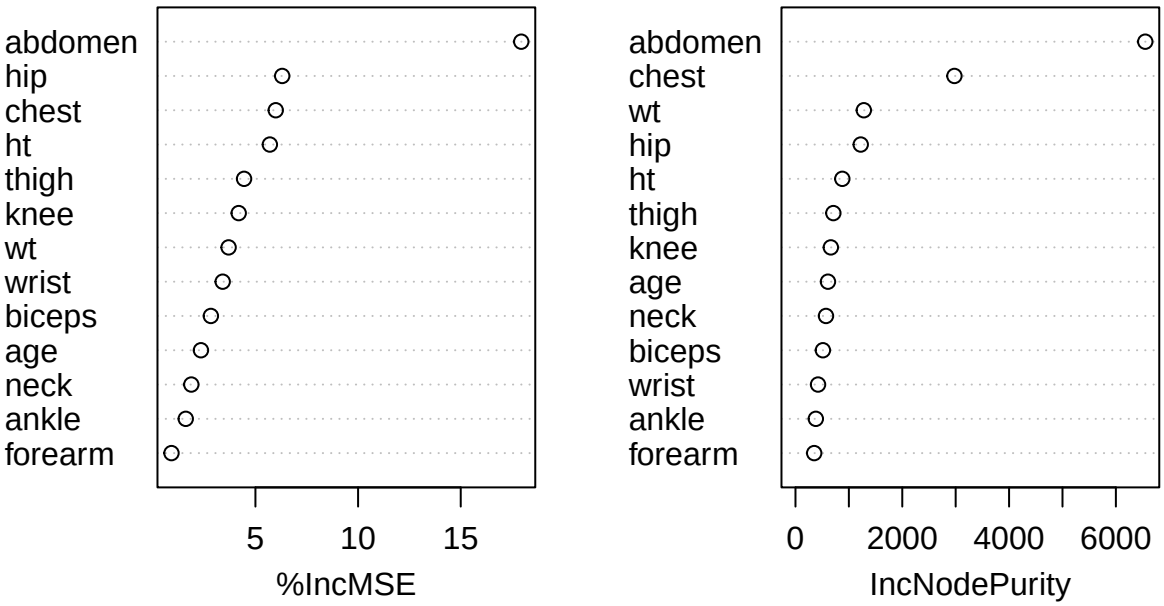
Random Forest

```r
library(randomForest)
bfat.rf <- randomForest(per_fat ~ . - density,data=bfat,mtry=5,
              importance=TRUE, ntree=100,
              na.action=na.omit)

print(bfat.rf)
```

```
##
## Call:
##  randomForest(formula = per_fat ~ . - density, data = bfat, mtry = 5,      importance = TRUE, ntree =
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 22.60518
##                     % Var explained: 67.59
```

```r
varImpPlot(bfat.rf)
```



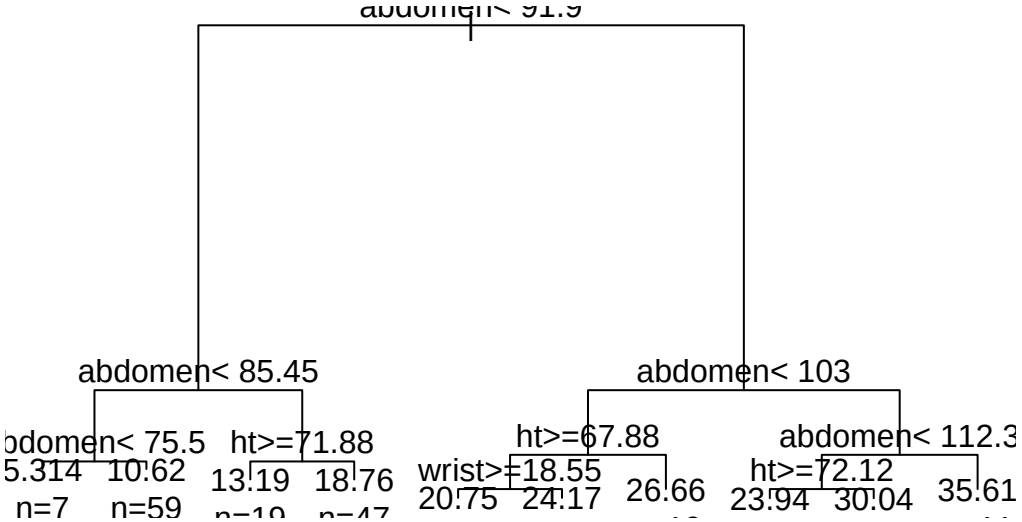
CART with rpart

```
library(rpart)
bfat.tr = rpart(per_fat ~ . - density, data=bfat)
# Output for Tree
print(bfat.tr)
```

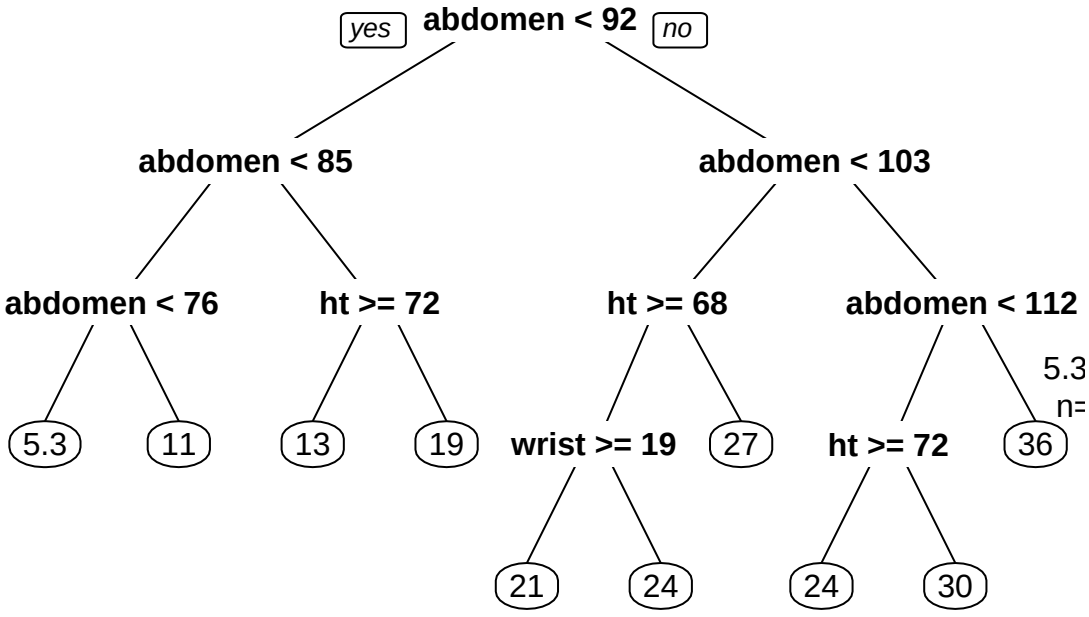
```
## n= 252
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 252 17578.99000 19.150790
##    2) abdomen< 91.9 132  4698.25500 13.606060
##      4) abdomen< 85.45 66  1303.62400 10.054550
##        8) abdomen< 75.5 7   113.54860  5.314286 *
##        9) abdomen>=75.5 59  1014.12300 10.616950 *
##      5) abdomen>=85.45 66  1729.68100 17.157580
##       10) ht>=71.875 19   407.33790 13.189470 *
##       11) ht< 71.875 47   902.23110 18.761700 *
##    3) abdomen>=91.9 120  4358.48000 25.250000
##      6) abdomen< 103 81  1752.42000 22.788890
##       12) ht>=67.875 71  1394.53500 22.243660
##         24) wrist>=18.55 40   718.49970 20.747500 *
##         25) wrist< 18.55 31   470.95940 24.174190 *
##       13) ht< 67.875 10   186.92400 26.660000 *
##      7) abdomen>=103 39  1096.45200 30.361540
##       14) abdomen< 112.3 28   413.60000 28.300000
##         28) ht>=72.125 8    89.39875 23.937500 *
```

```
##         29) ht< 72.125 20   111.04950 30.045000 *
##       15) abdomen>=112.3 11   260.94910 35.609090 *
```

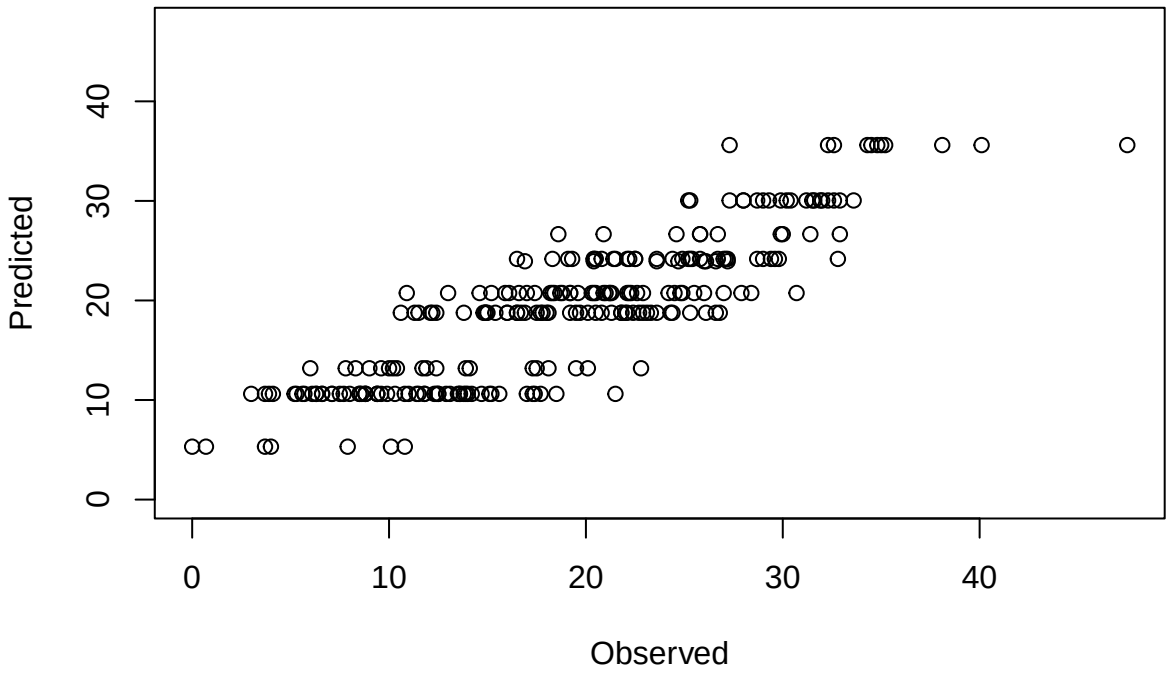
```
#plotting using rpart plot
plot(bfat.tr)
text(bfat.tr, use.n=T)
```




```
#plotting using rpart prp
library(rpart.plot)
prp(bfat.tr)
text(bfat.tr, use.n=T)
```

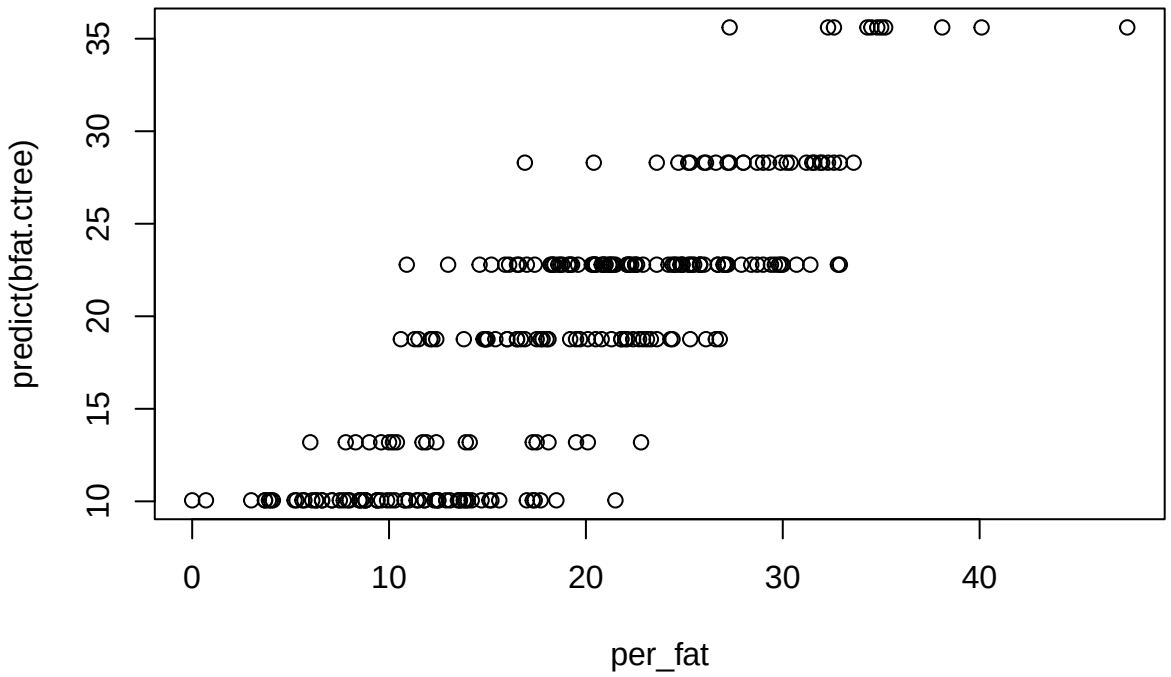



```
bfat_pred = predict(bfat.tr)
# plot predicted
xlim = range(per_fat)
plot(bfat_pred ~ per_fat, data=bfat, xlab="Observed",
     ylab="Predicted", ylim=xlim, xlim=xlim)
```



CART with party

```
bfat.ctree = ctree(per_fat ~ . - density, data=bfat)
# Output for Tree
plot(per_fat, predict(bfat.ctree))
```

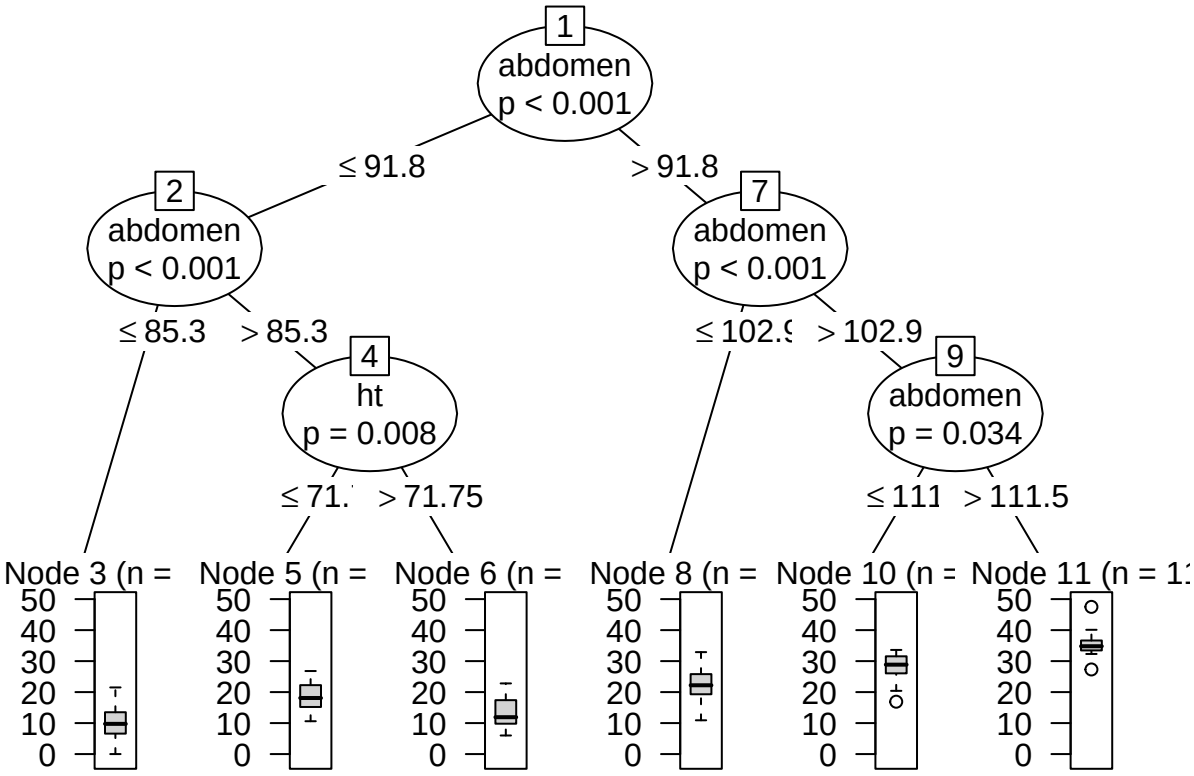



```
print(bfat.ctree)
```

```
##
## Model formula:
## per_fat ~ age + wt + ht + neck + chest + abdomen + hip + thigh +
##     knee + ankle + biceps + forearm + wrist
```

```
##
## Fitted party:
## [1] root
## |   [2] abdomen <= 91.8
## |   |   [3] abdomen <= 85.3: 10.055 (n = 66, err = 1303.6)
## |   |   [4] abdomen > 85.3
## |   |   |   [5] ht <= 71.75: 18.762 (n = 47, err = 902.2)
## |   |   |   [6] ht > 71.75: 13.189 (n = 19, err = 407.3)
## |   [7] abdomen > 91.8
## |   |   [8] abdomen <= 102.9: 22.789 (n = 81, err = 1752.4)
## |   |   [9] abdomen > 102.9
## |   |   |   [10] abdomen <= 111.5: 28.300 (n = 28, err = 413.6)
## |   |   |   [11] abdomen > 111.5: 35.609 (n = 11, err = 260.9)
##
## Number of inner nodes:    5
## Number of terminal nodes: 6
```

```
plot(bfat.ctree)
```




```
#plot(bfat.ctree, type="simple")
```

#Perform Classification

Random Forest

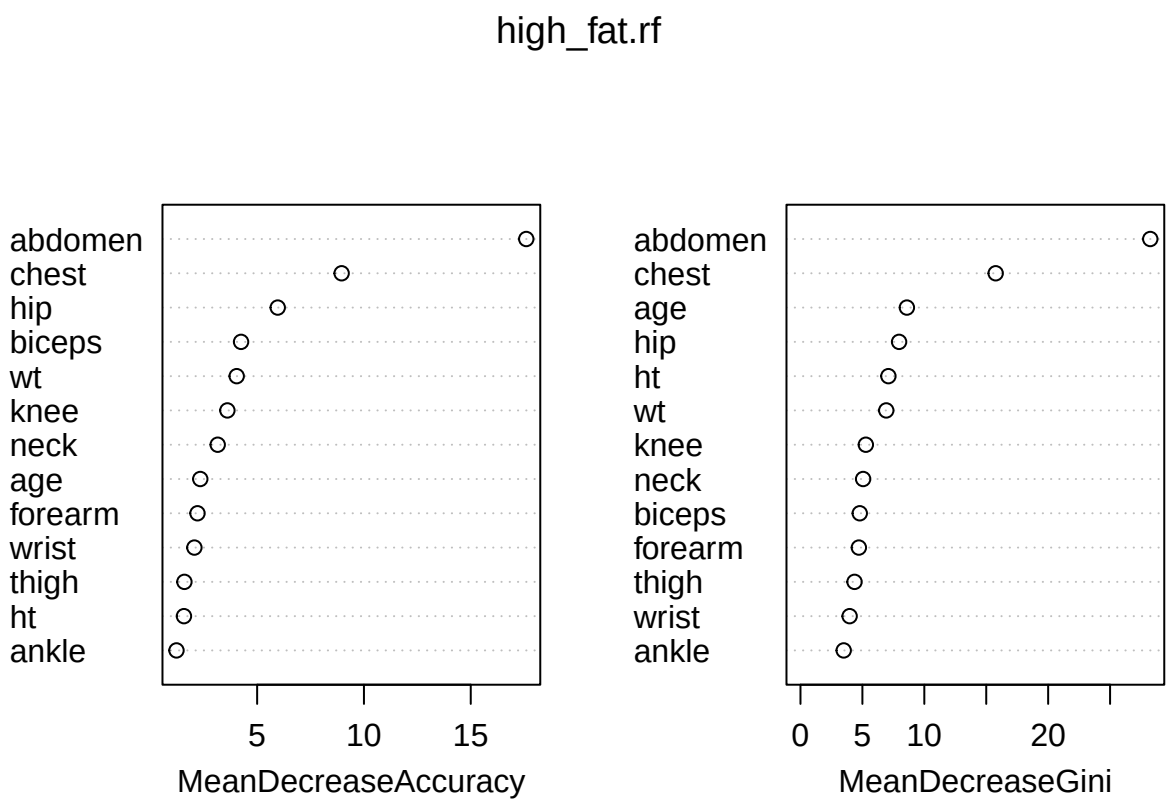
```
high_fat.rf <- randomForest(high_fat ~ .-density - per_fat, data=bfat,mtry=5, importance=TRUE, ntree=10

print(high_fat.rf)
```

```
##
## Call:
```

```
##  randomForest(formula = high_fat ~ . - density - per_fat, data = bfat,      mtry = 5, importance = TR
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 17.86%
## Confusion matrix:
##       FALSE TRUE class.error
## FALSE   157   19   0.1079545
## TRUE     26   50   0.3421053
```

```
varImpPlot(high_fat.rf)
```



CART with rpart

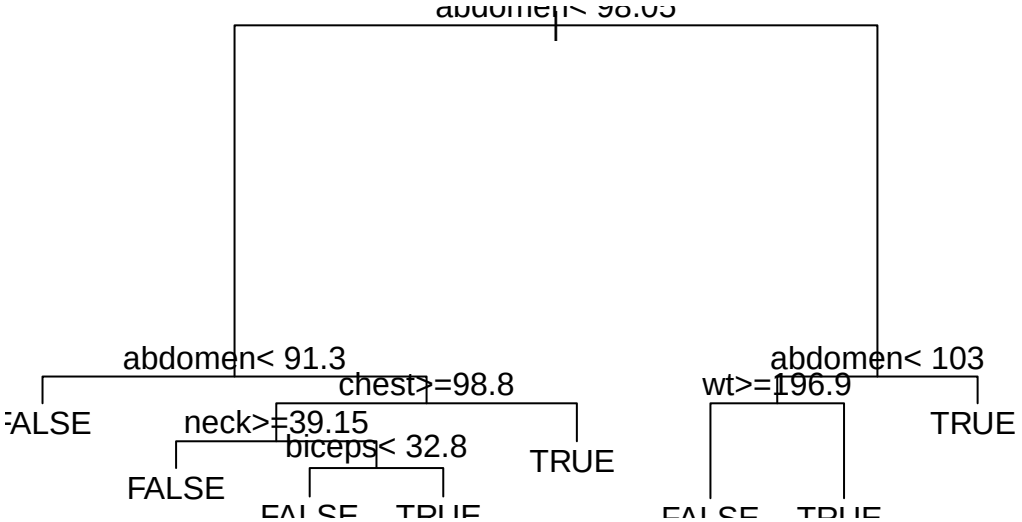
```
high_fat.tr = rpart(high_fat ~ .-density - per_fat, data=bfat)

print(high_fat.tr)
```

```
## n= 252
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 252 76 FALSE (0.69841270 0.30158730)
##    2) abdomen< 98.05 177 20 FALSE (0.88700565 0.11299435)
##      4) abdomen< 91.3 128  5 FALSE (0.96093750 0.03906250) *
##      5) abdomen>=91.3 49 15 FALSE (0.69387755 0.30612245)
##       10) chest>=98.8 39  8 FALSE (0.79487179 0.20512821)
##         20) neck>=39.15 16  0 FALSE (1.00000000 0.00000000) *
```

```
##         21) neck< 39.15 23  8 FALSE (0.65217391 0.34782609)
##           42) biceps< 32.8 16  3 FALSE (0.81250000 0.18750000) *
##           43) biceps>=32.8 7  2 TRUE (0.28571429 0.71428571) *
##       11) chest< 98.8 10  3 TRUE (0.30000000 0.70000000) *
##    3) abdomen>=98.05 75 19 TRUE (0.25333333 0.74666667)
##      6) abdomen< 103 36 16 TRUE (0.44444444 0.55555556)
##       12) wt>=196.875 18  4 FALSE (0.77777778 0.22222222) *
##       13) wt< 196.875 18  2 TRUE (0.11111111 0.88888889) *
##      7) abdomen>=103 39  3 TRUE (0.07692308 0.92307692) *
```

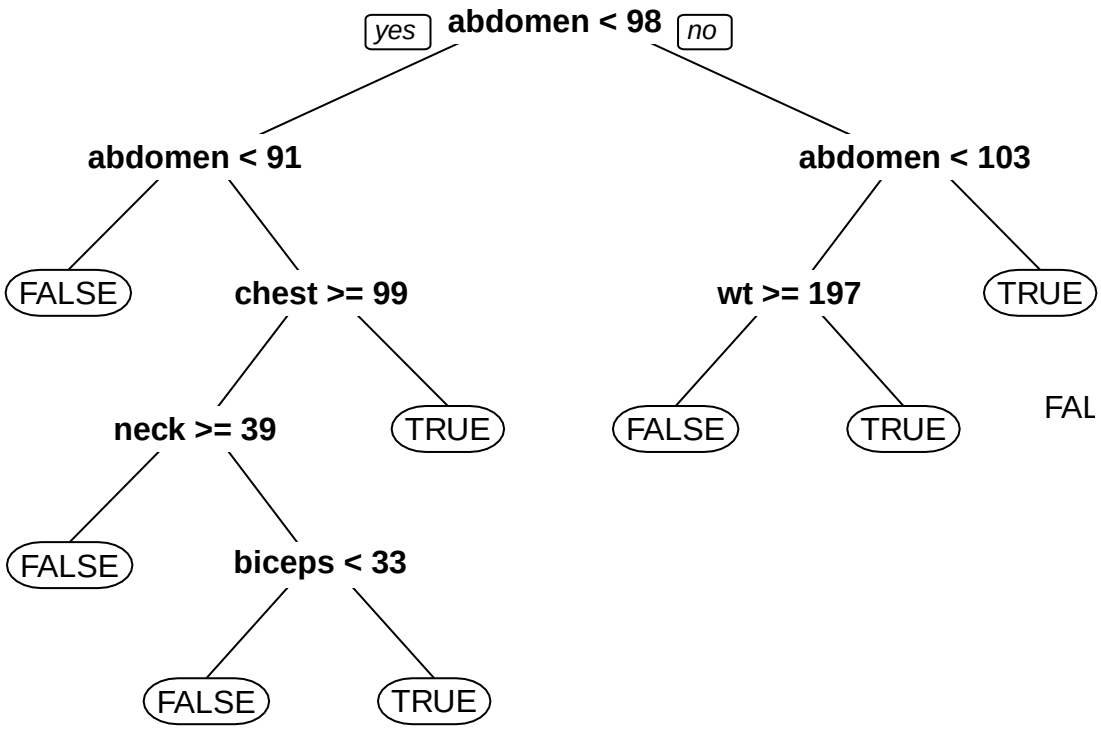
```
plot(high_fat.tr)
text(high_fat.tr)
```




```
print(high_fat.tr)
```

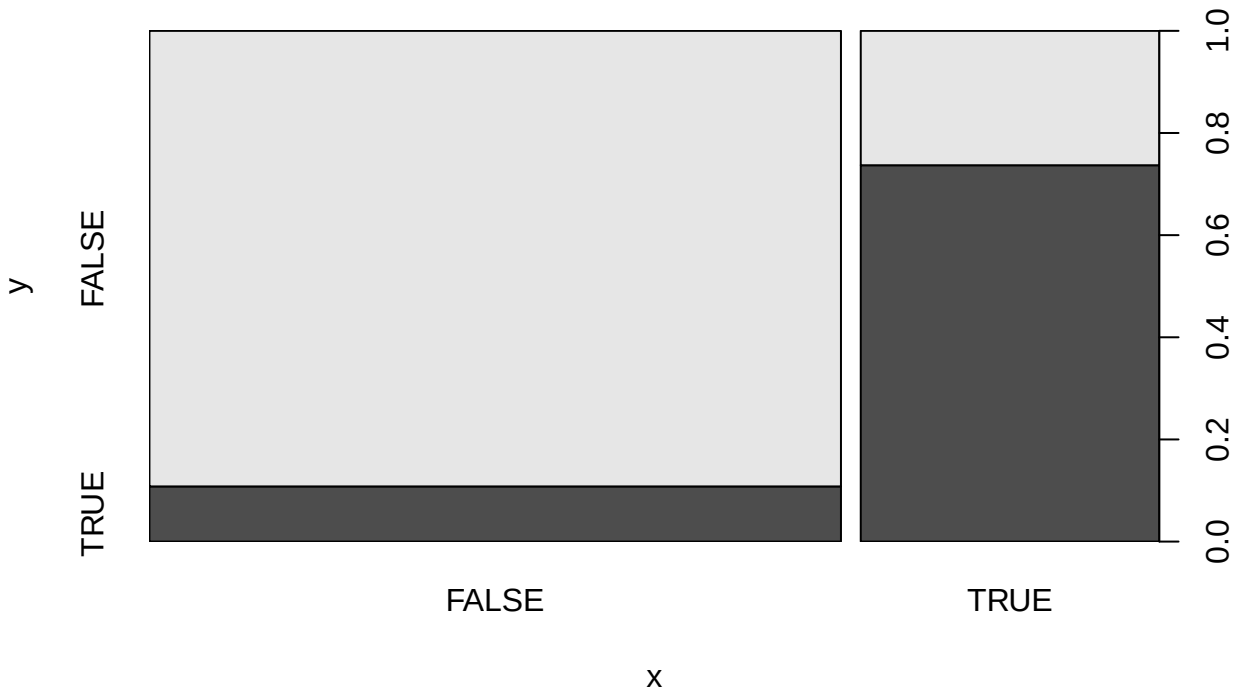
```
## n= 252
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 252 76 FALSE (0.69841270 0.30158730)
##    2) abdomen< 98.05 177 20 FALSE (0.88700565 0.11299435)
##      4) abdomen< 91.3 128  5 FALSE (0.96093750 0.03906250) *
##      5) abdomen>=91.3 49 15 FALSE (0.69387755 0.30612245)
##       10) chest>=98.8 39  8 FALSE (0.79487179 0.20512821)
##         20) neck>=39.15 16  0 FALSE (1.00000000 0.00000000) *
##         21) neck< 39.15 23  8 FALSE (0.65217391 0.34782609)
##           42) biceps< 32.8 16  3 FALSE (0.81250000 0.18750000) *
##           43) biceps>=32.8 7  2 TRUE (0.28571429 0.71428571) *
##       11) chest< 98.8 10  3 TRUE (0.30000000 0.70000000) *
##    3) abdomen>=98.05 75 19 TRUE (0.25333333 0.74666667)
##      6) abdomen< 103 36 16 TRUE (0.44444444 0.55555556)
##       12) wt>=196.875 18  4 FALSE (0.77777778 0.22222222) *
##       13) wt< 196.875 18  2 TRUE (0.11111111 0.88888889) *
##      7) abdomen>=103 39  3 TRUE (0.07692308 0.92307692) *
```

```
prp(high_fat.tr)
text(high_fat.tr)
```



CART with party

```
high_fat.ctree = ctree(high_fat ~ . - density - per_fat, data=bfat)
# Output for Tree
plot(high_fat, predict(high_fat.ctree))
```




```
print(high_fat.ctree)
```

```
##
## Model formula:
```

```
## high_fat ~ age + wt + ht + neck + chest + abdomen + hip + thigh +
##     knee + ankle + biceps + forearm + wrist
##
## Fitted party:
## [1] root
## |   [2] abdomen <= 98
## |   |   [3] abdomen <= 91.1: FALSE (n = 128, err = 3.9%)
## |   |   [4] abdomen > 91.1: FALSE (n = 49, err = 30.6%)
## |   [5] abdomen > 98: TRUE (n = 75, err = 25.3%)
##
## Number of inner nodes:    2
## Number of terminal nodes: 3
```

```
plot(high_fat.ctree)
```