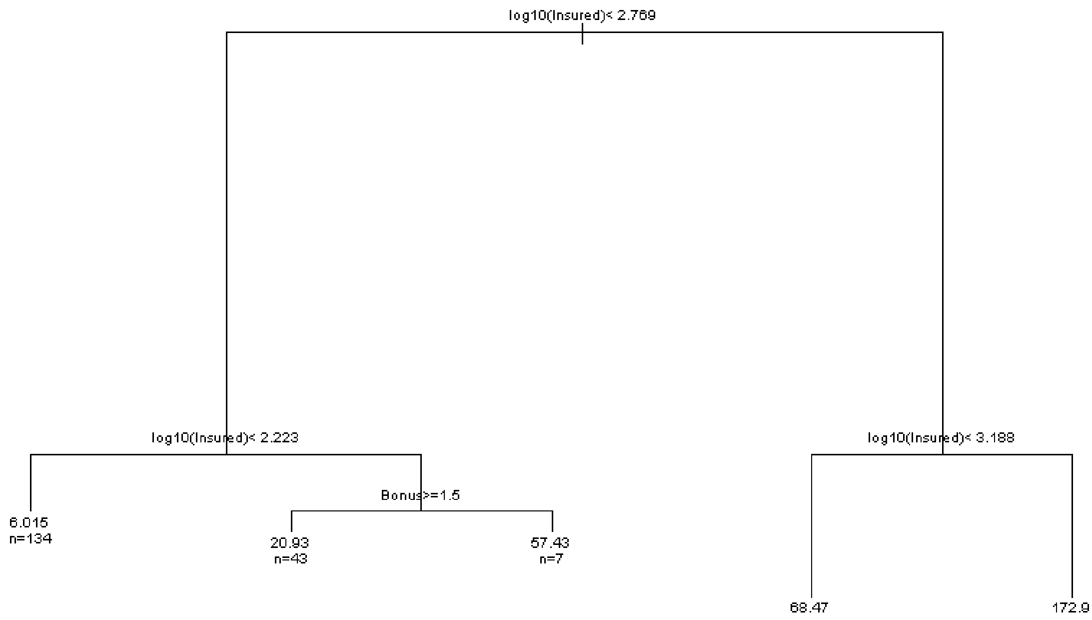
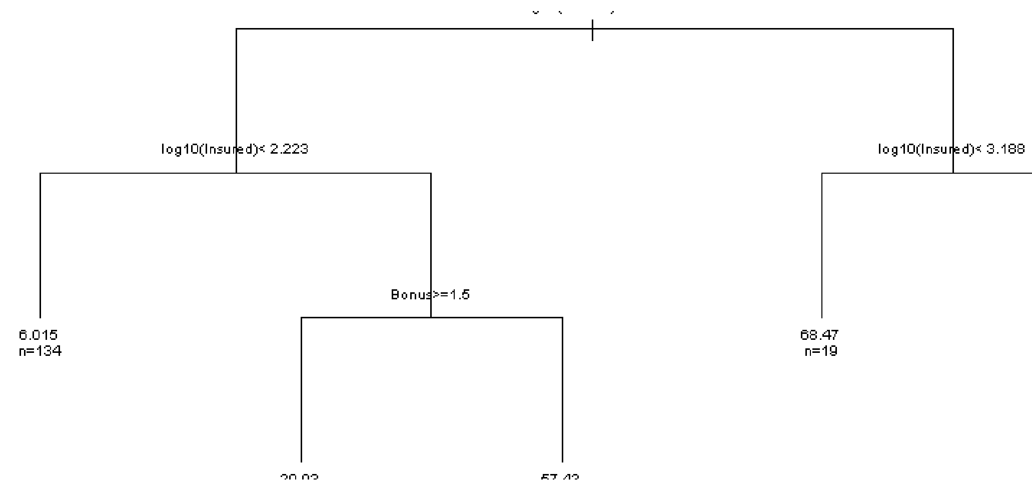
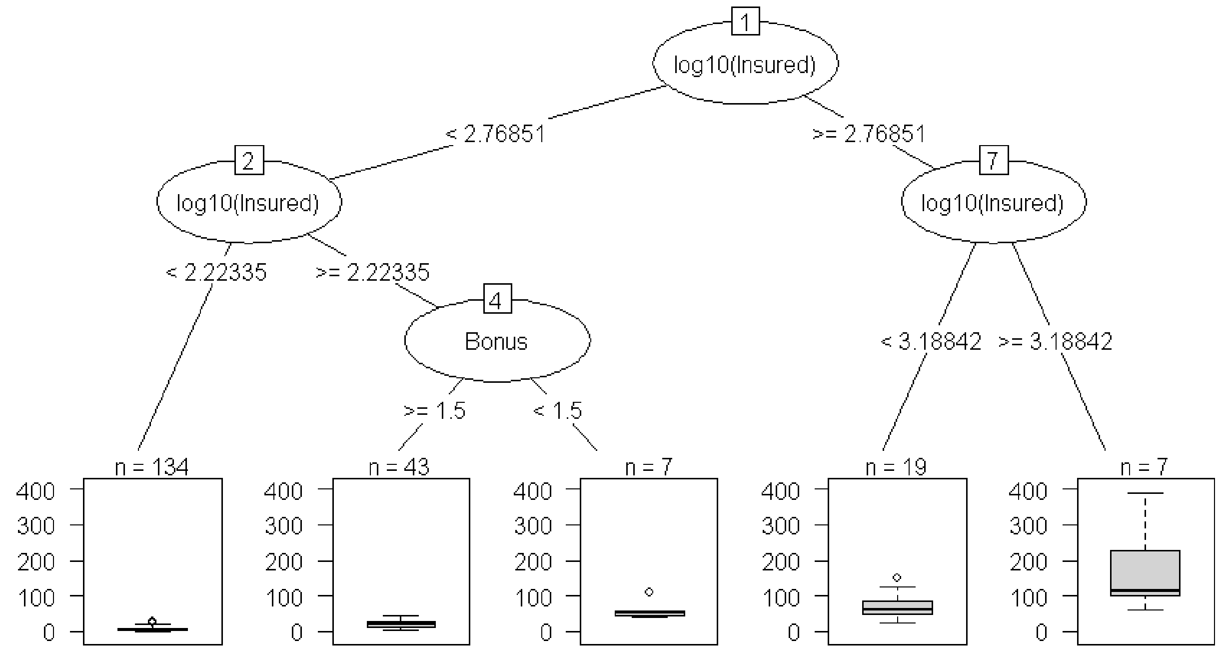
```
library("rpart")
Cl.part <- rpart(Claims ~ fmake + fkilo + log10(Insured)+Bonus,
                 data=mot.sgm)

plot(Cl.part)
```



```
plot(Cl.part,uniform=T)
text(Cl.part, use.n=TRUE,cex=0.6)
```



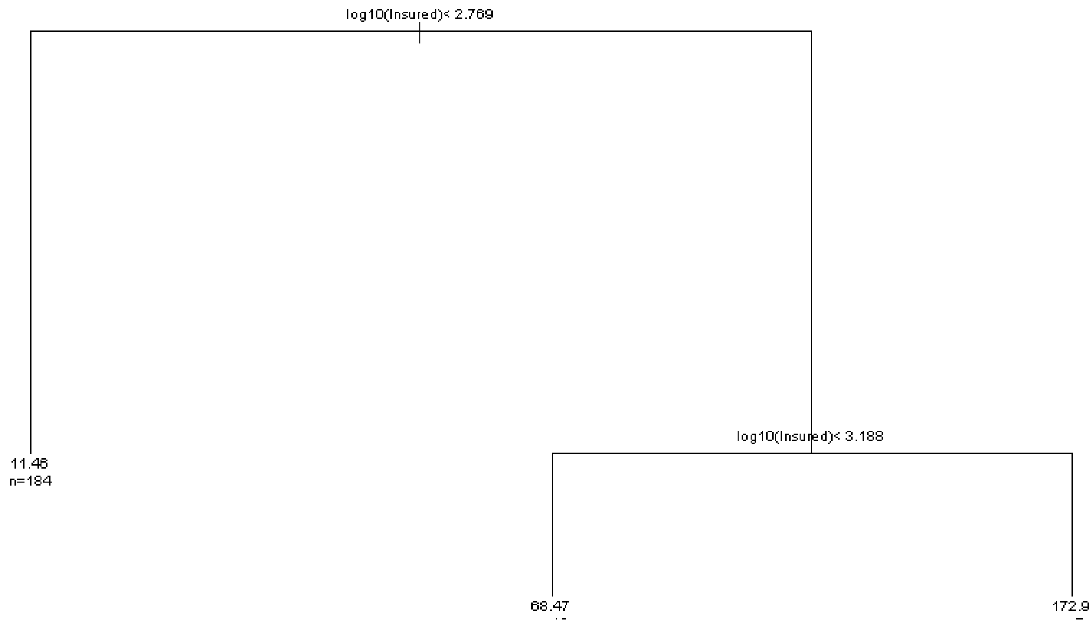

```
library("partykit")
plot(as.party(Cl.part),tp_args=list(id=F))
```



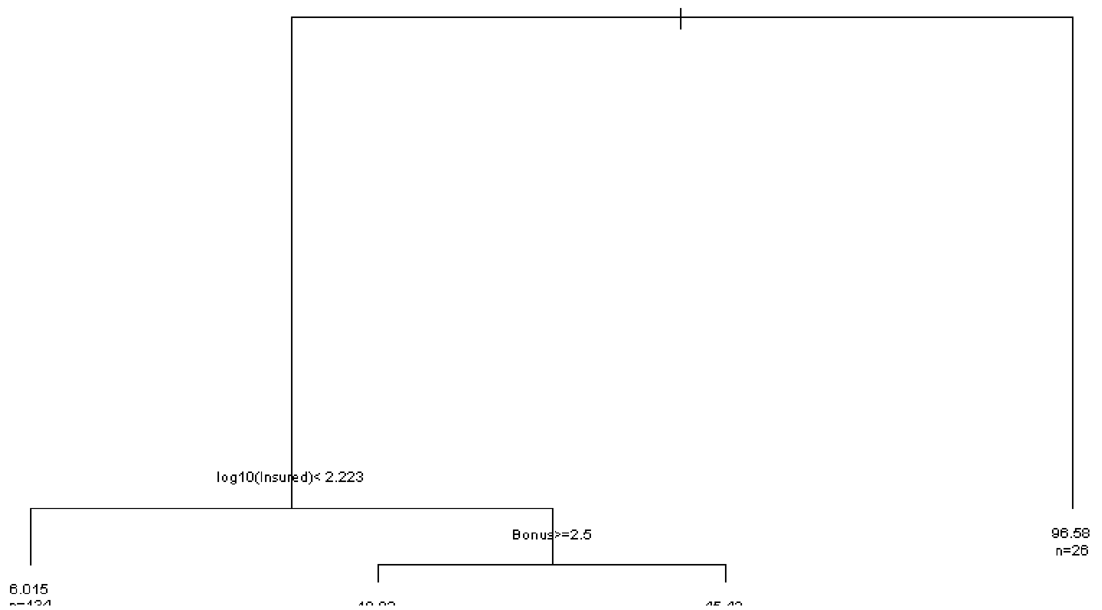
```
print(Cl.part)

n= 210

node), split, n, deviance, yval
      * denotes terminal node

 1) root 210 351789.000  21.995240
   2) log10(Insured)< 2.768512 184  34291.650  11.456520
     4) log10(Insured)< 2.22335 134   4233.970   6.014925 *
     5) log10(Insured)>=2.22335 50  15455.920  26.040000
      10) Bonus>=1.5 43   4190.791  20.930230 *
      11) Bonus< 1.5 7   3245.714  57.428570 *
   3) log10(Insured)>=2.768512 26 152438.300  96.576920
     6) log10(Insured)< 3.188424 19  16432.740  68.473680 *
     7) log10(Insured)>=3.188424 7  80268.860 172.857100 *

summary(Cl.part)

Call:
rpart(formula = Claims ~ fmake + fkilo + log10(Insured) + Bonus,
    data = mot.sgm)
  n= 210

          CP nsplit rel error    xerror      xstd
1 0.46919886      0 1.0000000 1.0187960 0.4281900
2 0.15843802      1 0.5308011 0.9639999 0.4142921
3 0.04150716      2 0.3723631 0.6107336 0.2846787
4 0.02279609      3 0.3308560 0.6044336 0.2847447
5 0.01000000      4 0.3080599 0.5593928 0.2522457

Node number 1: 210 observations,    complexity param=0.4691989
  mean=21.99524, MSE=1675.186
  left son=2 (184 obs) right son=3 (26 obs)
  Primary splits:
      log10(Insured) < 2.768512 to the left,  improve=0.46919890, (0 missing)
      fmake          splits as  RL-LLL-L-,    improve=0.19546080, (0 missing)
      Bonus          < 6.5      to the left,  improve=0.15790030, (0 missing)
      fkilo          splits as  RRRLL,        improve=0.08666632, (0 missing)
  Surrogate splits:
      Bonus < 6.5        to the left,  agree=0.886, adj=0.077, (0 split)

printcp(Cl.part)

Regression tree:
rpart(formula = Claims ~ fmake + fkilo + log10(Insured) + Bonus,
    data = mot.sgm)

Variables actually used in tree construction:
[1] Bonus          log10(Insured)

Root node error: 351789/210 = 1675.2

n= 210

        CP nsplit rel error  xerror    xstd
1 0.469199      0   1.00000 1.01880 0.42819
2 0.158438      1   0.53080 0.96400 0.41429
3 0.041507      2   0.37236 0.61073 0.28468
4 0.022796      3   0.33086 0.60443 0.28474
5 0.010000      4   0.30806 0.55939 0.25225
```

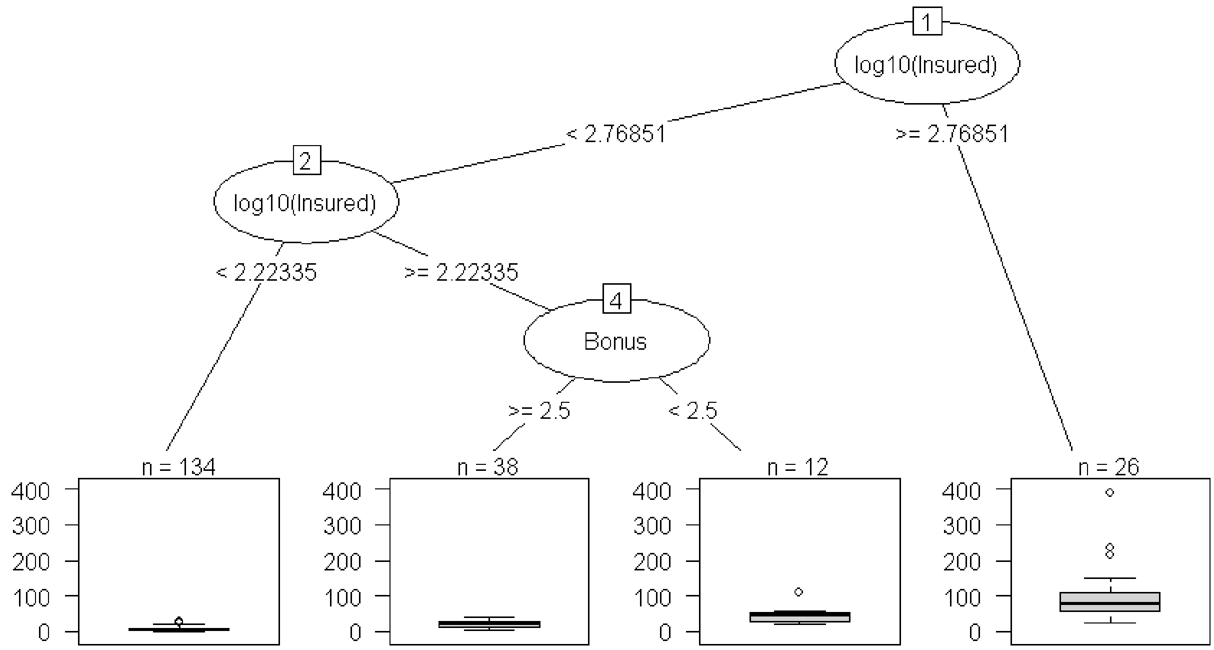
xerror – PRESS statistics (buduje się model bez i-tei obserwacji, prognozuje się y_i a bierze się różnicę kwadratową między y_i a prognozą -> suma po i

```
Cl.part1 <- prune(Cl.part,cp=0.05)
plot(Cl.part1)
text(Cl.part1, use.n=TRUE,cex=0.6)
```



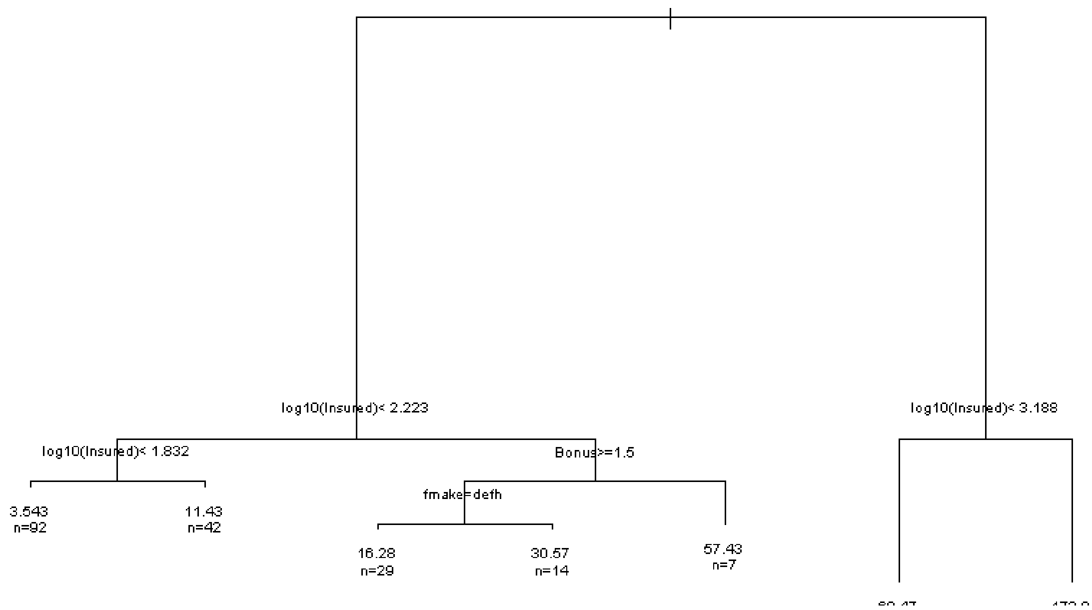
```
Cl.part30 <- rpart(Claims ~ fmake + fkilo + log10(Insured)+Bonus,
                 control = rpart.control(minsplit=30),
                 data=mot.sgm)
plot(Cl.part30)
text(Cl.part30, use.n=TRUE,cex=0.6)
```





```
print(Cl.part30)

n= 210

node), split, n, deviance, yval
      * denotes terminal node

 1) root 210 351789.000 21.995240
   2) log10(Insured)< 2.768512 184  34291.650 11.456520
     4) log10(Insured)< 2.22335 134   4233.970  6.014925 *
     5) log10(Insured)>=2.22335 50  15455.920 26.040000
      10) Bonus>=2.5 38   3480.763 19.921050 *
      11) Bonus< 2.5 12   6046.917 45.416670 *
   3) log10(Insured)>=2.768512 26 152438.300 96.576920 *
```

```
Cl.part004 <- rpart(Claims ~ fmake + fkilo + log10(Insured)+Bonus,
                    control = rpart.control(cp=0.04),
                    data=mot.sgm)
plot(Cl.part004)
text(Cl.part004, use.n=TRUE,cex=0.6)
```



```
print(Cl.part004)
```

```
n= 210

node), split, n, deviance, yval
      * denotes terminal node

 1) root 210 351789.0000  21.995240
   2) log10(Insured)< 2.768512 184   34291.6500  11.456520
     4) log10(Insured)< 2.22335 134    4233.9700   6.014925
       8) log10(Insured)< 1.8315 92     984.8261   3.543478 *
       9) log10(Insured)>=1.8315 42    1456.2860  11.428570 *
     5) log10(Insured)>=2.22335 50   15455.9200  26.040000
      10) Bonus>=1.5 43    4190.7910  20.930230
        20) fmake=4,5,6,8 29    1273.7930  16.275860 *
        21) fmake=1,2 14     987.4286  30.571430 *
      11) Bonus< 1.5 7    3245.7140  57.428570 *
   3) log10(Insured)>=2.768512 26 152438.3000  96.576920
     6) log10(Insured)< 3.188424 19  16432.7400  68.473680 *
     7) log10(Insured)>=3.188424 7  80268.8600 172.857100 *
```

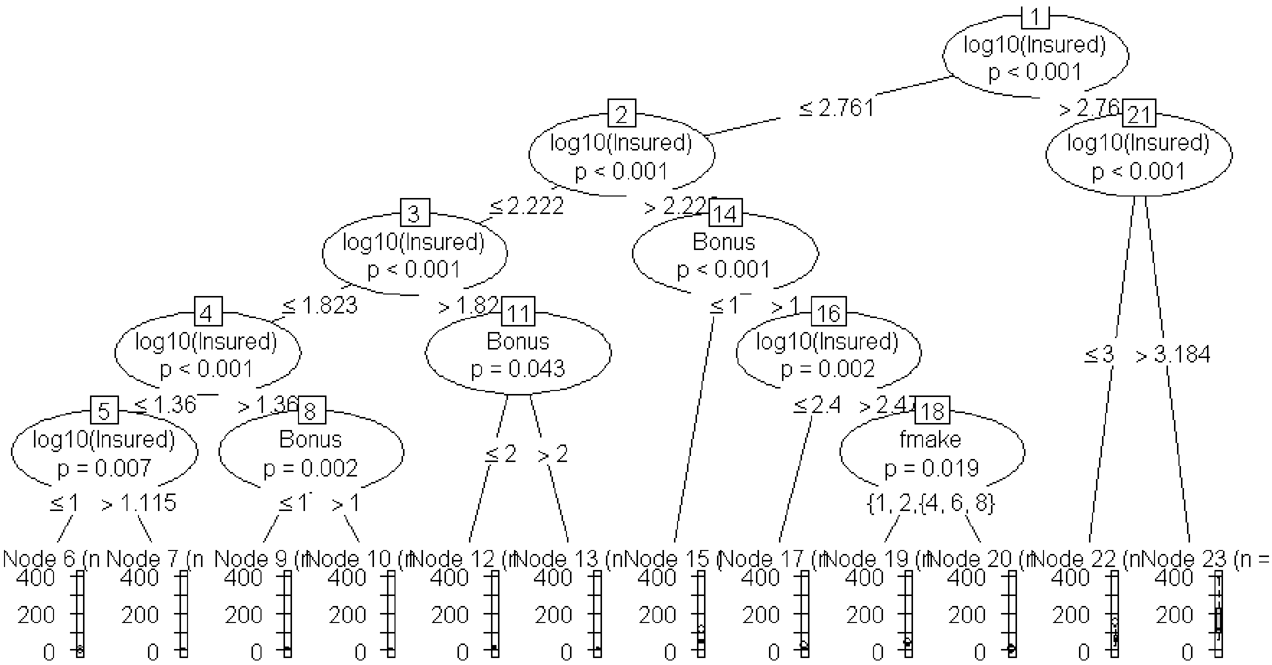
```
Cl.part004$cptable
```

```
           CP nsplit rel error    xerror      xstd
1 0.469198864      0 1.0000000 1.0073344 0.4232063
2 0.158438021      1 0.5308011 0.7102673 0.2739999
3 0.041507160      2 0.3723631 0.5497413 0.2641215
4 0.022796094      3 0.3308560 0.4829506 0.1993378
5 0.005485018      4 0.3080599 0.4592409 0.2000840
6 0.005096403      5 0.3025748 0.4482159 0.1996284
7 0.004000000      6 0.2974784 0.4430436 0.1996484
```

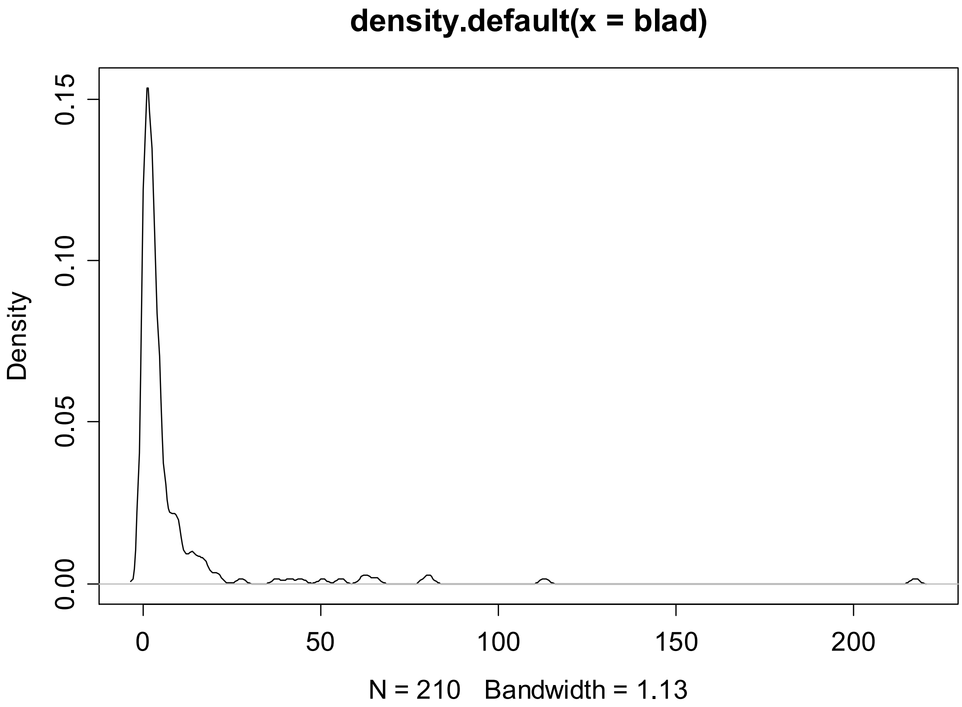
```
Cl.partxv <- rpart(Claims ~ fmake + fkilo + log10(Insured)+Bonus,
                   control = rpart.control(xval=50),
                   data=mot.sgm)
```

jak wyjściowy podział

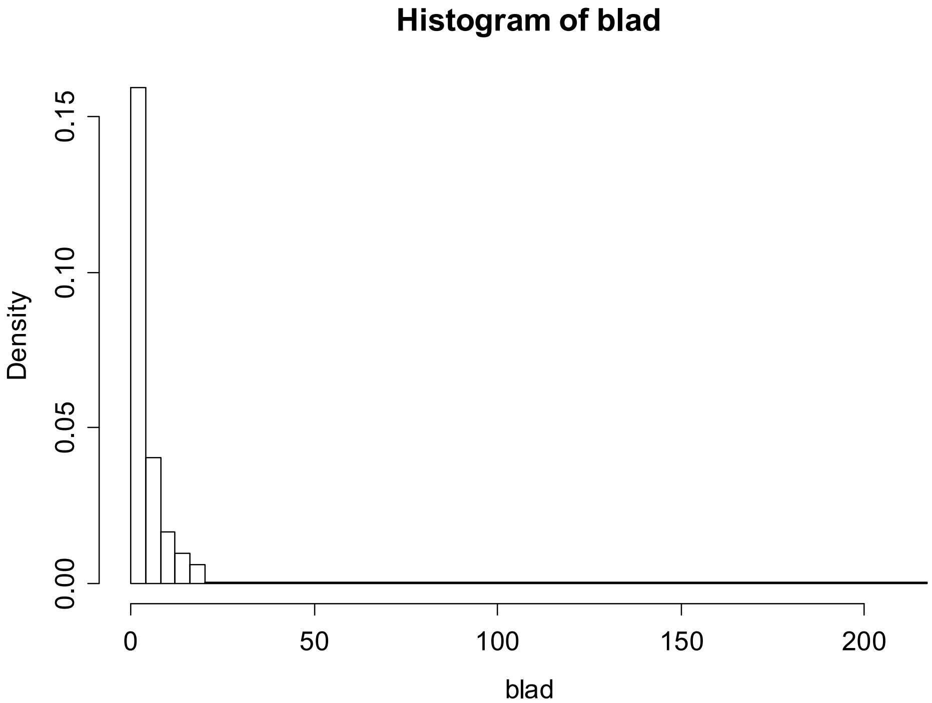
```
library("party")
Cl.ctree <- ctree(Claims ~ fmake + fkilo + log10(Insured)+Bonus,
                  data=mot.sgm)
plot(Cl.ctree)
```




```
pr.ctree <- predict(Cl.ctree)
blad <- abs(mot.sgm$Claims-pr.ctree)
plot(density(blad))
```




```
hist(blad,fr=F,breaks=c(0,(1:5)*4,max(blad)))
```




```
blad.f <-cut(blad,include.lowest =T, breaks=c(0,(1:5)*4,max(blad)))
print(table(blad.f)/2.1,digits=3)

blad.f
   [0,4]    (4,8]   (8,12]  (12,16]  (16,20] (20,217] 
   63.81    16.19     6.67     3.81     2.38     7.14 
```

```
data("GlaucomaM",package = "ipred")
str(GlaucomaM)
```

```
'data.frame':   196 obs. of  63 variables:
 $ ag   : num  2.22 2.68 1.98 1.75 2.99 ...
 $ at   : num  0.354 0.475 0.343 0.269 0.599 0.483 0.355 0.312 0.365 0.53 ...
 $ as   : num  0.58 0.672 0.508 0.476 0.686 0.763 0.601 0.463 0.57 0.834 ...
…
$ mv    : num  0.035 0.022 0.029 0.023 0.034 0.028 0.012 0.023 0.029 0.022 ...
 $ Class: Factor w/ 2 levels "glaucoma","normal": 2 2 2 2 2 2 2 2 2 2 ...
```

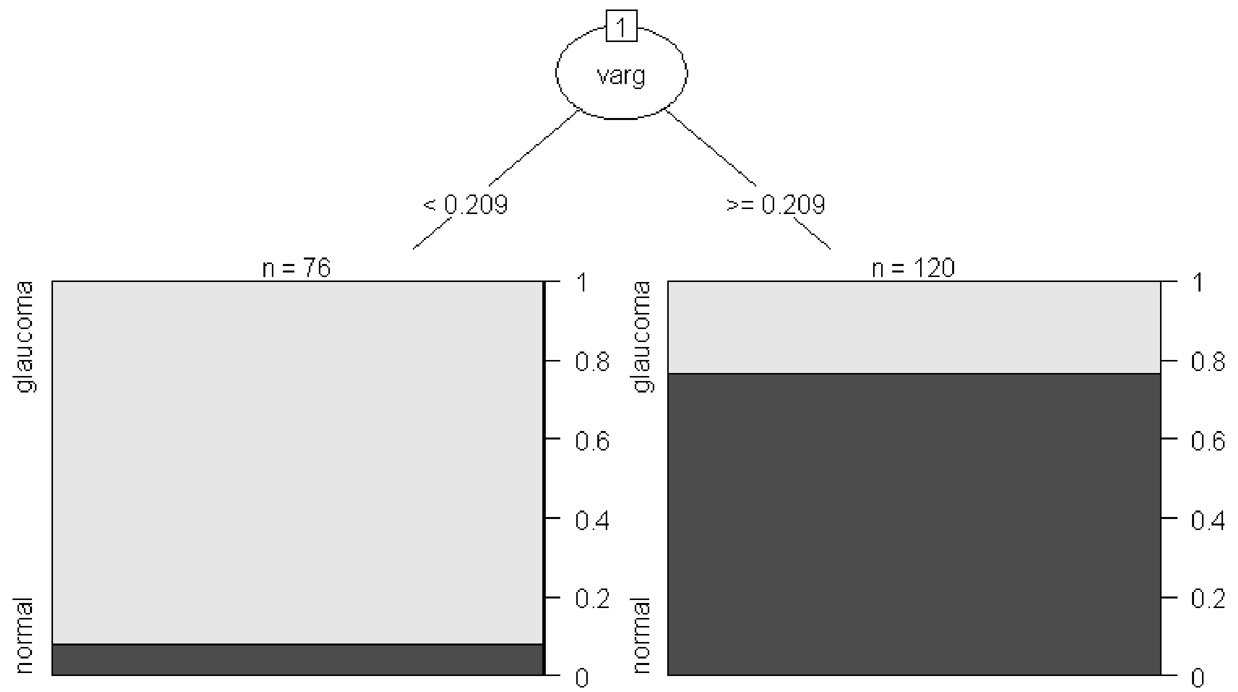
```
glaucoma.t <- rpart(Class ~ .,data=GlaucomaM,control = rpart.control(xval=100))
```

xval=100 100 razy 10 krotna kroswalidacja (normalnie 10 razy)

```
glaucoma.t$cptable
```

```
          CP nsplit rel error    xerror       xstd
1 0.65306122      0 1.0000000 1.6020408 0.05703324
2 0.07142857      1 0.3469388 0.3877551 0.05647630
3 0.01360544      2 0.2755102 0.3877551 0.05647630
4 0.01000000      5 0.2346939 0.4795918 0.06099412
```

```
 (opt <- which.min(glaucoma.t$cptable[,"xerror"]))
2
(cp.opt <- glaucoma.t$cptable[opt,"CP"])
0.07142857
plot(as.party(glaucoma.tpr),tp_args=list(id=F))
```



varg - volume of optic nerve above reference global stabilność drzewa

```
nsplitopt <- vector(mode="integer",length=25)
 for (i in 1:length(nsplitopt)) {
     cp <-rpart(Class ~ .,data=GlaucomaM)$cptable
      nsplitopt[i] <- cp[which.min(cp[,"xerror"]),"nsplit"]
    }
table(nsplitopt)

nsplitopt
 1  2  5
15  5  5
```

W 15 powtórzeniach 1 węzeł, w 5 – 2 w 5 – 5

bagging = bootstrap aggregating

```
trees <- vector(mode="list",length=25)
n <- nrow(GlaucomaM)
boot <- rmultinom(length(trees),n,rep(1,n)/n)
```

25 razy rozkład wielomianowy z pstwem 1/n

```
mod <- rpart(Class ~ .,data=GlaucomaM, control =rpart.control(xval=0))
 for(i in 1:length(trees)) trees[[i]] <- update(mod,weights=boot[,i])
```

xval=0 bez pruning
weights z takim pstwem losuje się elementy do modelu
update(mod,weights=boot[,i]) uaktualnij model, dokładając opcję weights

```
#zmienne użyte w korzeniu
table(sapply(trees, function(x) as.character(x$frame$var[1])))
tmi varg vari varn vars
   1   10   11    1    2
```

Sapply w wyniku daje wektor lub macierz

```
pstwo <- matrix(0,nrow=n,ncol=length(trees))
for (i in 1: length(trees)){
    pstwo[,i] <- predict(trees[[i]],newdata = GlaucomaM)[,1]
    pstwo[boot[,i]>0,i] <-NA
   }
```

pstwo[boot[,i]>0,i] <-NA wyłączamy te wiersze, które posłużyły do zbudowania modelu

```
srd <- rowMeans(pstwo,na.rm=T)

prognoza <-factor(ifelse(srd>0.5,"JASKRA","NORMALNE"))
(progTab <-table(prognoza,GlaucomaM$Class))

prognoza   glaucoma normal
  JASKRA         78     12
  NORMALNE       20     86
czulosc <- round(progTab[1,1]/colSums(progTab)[1]*100)
specyficznosc <- round(progTab[2,2]/colSums(progTab)[2]*100)
cat("czułość=", czulosc," specyficzność=" , specyficznosc)
```

czulosc poprawna klasyfikacja choroby
specyficzność poprawna klasyfikacja zdrowia

```
czułość= 80  specyficzność= 88
```

LASY LOSOWE

Learning algorithm

Each tree is constructed using the following algorithm:

1. Let the number of training cases be *N*, and the number of variables in the classifier be *M*.
2. We are told the number *m* of input variables to be used to determine the decision at a node of the tree; *m* should be much less than *M*.
3. Choose a training set for this tree by choosing *n* times with replacement from all *N* available training cases (i.e. take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.
4. For each node of the tree, randomly choose *m* variables on which to base the decision at that node. Calculate the best split based on these *m* variables in the training set.
5. Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

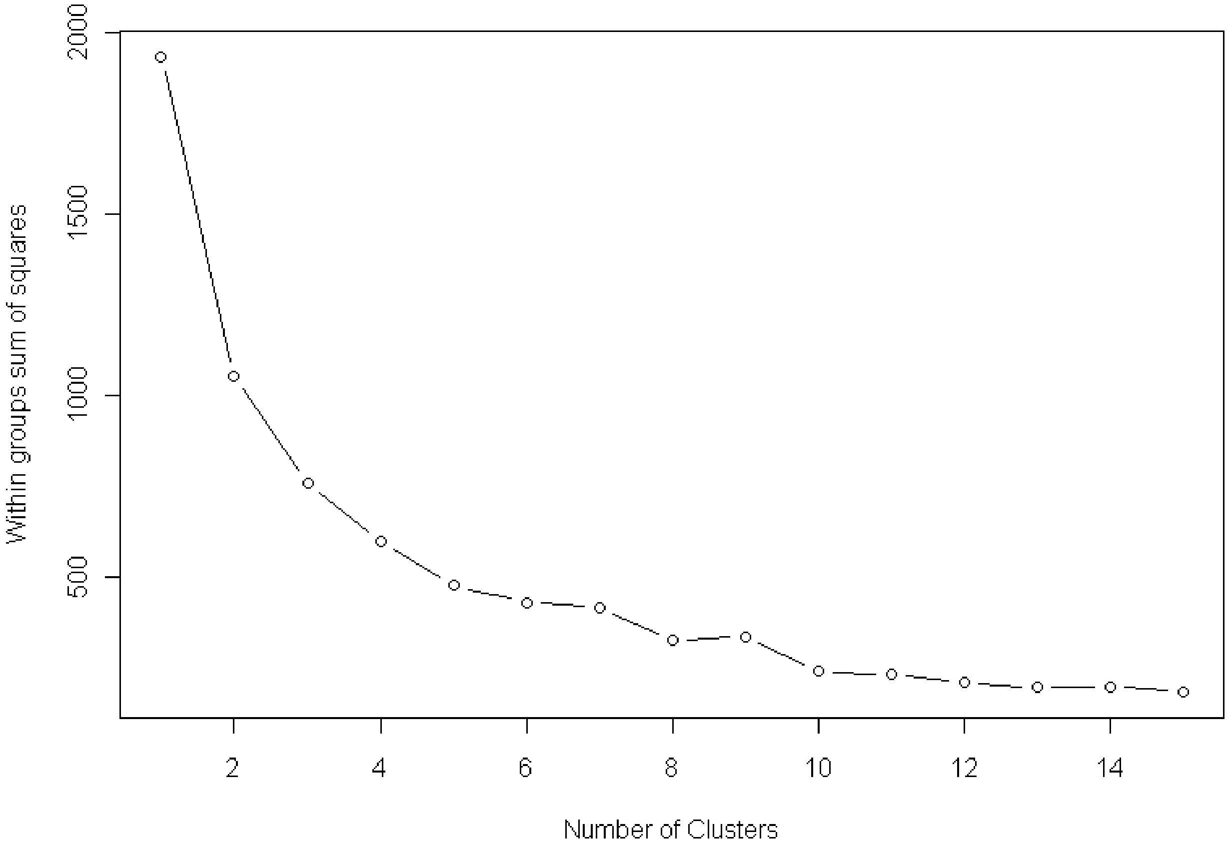
For prediction a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the average vote of all trees is reported as random forest prediction

```
las <- randomForest(Class ~ .,data=GlaucomaM)
ntree=500
mtry=if (!is.null(y) && !is.factor(y))
             max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x)))
sampsize = if (replace) nrow(x) else ceiling(.632*nrow(x))


(t.las <- table(predict(las),GlaucomaM$Class))

           glaucoma normal
  glaucoma       82     10
  normal         16     88

czulosc <- round(t.las[1,1]/colSums(t.las)[1]*100)
specyficznosc <- round(t.las[2,2]/colSums(t.las)[2]*100)
cat("czułość=", czulosc," specyficzność=" , specyficznosc)

czułość= 84  specyficzność= 90

print(las)

Call:
 randomForest(formula = Class ~ ., data = GlaucomaM)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 7

        OOB estimate of  error rate: 13.27%
Confusion matrix:
         glaucoma normal class.error
glaucoma       82     16   0.1632653
normal         10     88   0.1020408
```

CLUSTER

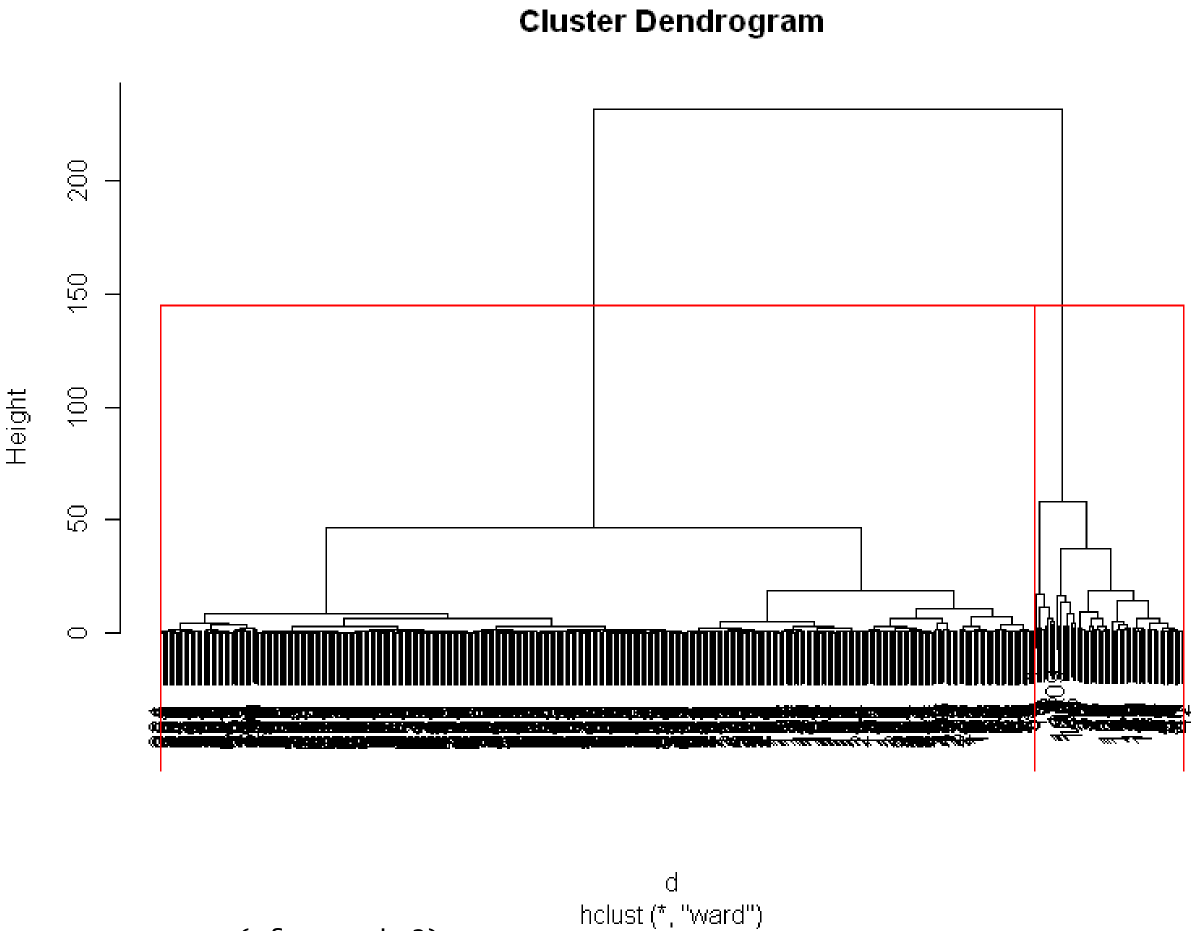
```
f.BIM<-with(Forbes.BIM, cbind(sales,profits,assets,marketvalue))
f.BIM <-na.exclude(f.BIM)
sf.BIM<-scale(f.BIM)
wss <- (nrow(sf.BIM)-1)*sum(apply(sf.BIM,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(sf.BIM,centers=i)$withinss)
```

```
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
```




```
mkm5<-kmeans(sf.BIM,5)
aggregate(sf.BIM,by=list(mkm5$cluster),FUN=median)

  Group.1      sales    profits     assets marketvalue
1       1  2.1857397  1.6670971  3.6724560   1.8594195
2       2  0.9025741  0.6389967  0.8161410   0.6001001
3       3  4.0294719  3.7749899  4.1312765   7.4316966
4       4 -0.4190537 -0.1363862 -0.3764153  -0.3487328
5       5  1.6206722 -3.1603797  2.7209062   0.6809513
m5.cl<-mkm5$cluster
table(m5.cl)
m5.cl
  1   2   3   4   5
 15  47   5 411   6


mkm2<-kmeans(sf.BIM,2)
aggregate(sf.BIM,by=list(mkm2$cluster),FUN=median)
  Group.1      sales    profits     assets marketvalue
1       1 -0.3924575 -0.1363862 -0.3633421  -0.3349235
2       2  2.1444132  1.4143797  2.0413175   1.7937140

m2.cl<-mkm2$cluster
table(m2.cl)
m2.cl
  1   2
446  38
```

```
d <- dist(sf.BIM, method = "euclidean") # odległość
wf.BIM <- hclust(d, method="ward")
plot(wf.BIM)
rect.hclust(wf.BIM, k=2, border="red")
```




```
grupy <- cutree(wf.BIM, k=2)
table(grupy)
grupy
  1   2
 71 413

aggregate(sf.BIM,by=list(grupy),FUN=median)
  Group.1      sales    profits     assets marketvalue
1       1  1.4254968  0.6849453  1.2320964   0.9275142
2       2 -0.4165987 -0.1363862 -0.3728327  -0.3482873
```