Partitioning treatment trajectories

Marjolein Fokkema

We have three clinician-reported response variables that were assessed over time: CGI1 and CGI2 (Clinical Global Impression) and GAF (General Assessment of Functioning). We will partition the datasets based on a linear-mixed effects model (LMM) using function `lmertree`, and based on a (non-linear) generalized additive model (GAM).

First, we load libraries and data:

```
## Load libraries
library("foreign")
library("glmertree")
library("gamtree")

## Read data
CGI1 <- read.spss("PP5_ITT_manova_v10_observed_MLwiN_CGI1_ED20120314.sav",
                  to.data.frame = TRUE)
CGI2 <- read.spss("PP5_ITT_manova_v10_observed_MLwiN_CGI2_ED20120314.sav",
                   to.data.frame = TRUE)
GAF <- read.spss("PP5_ITT_manova_v10_observed_MLwiN_GAF_ED20120314.sav",
                  to.data.frame = TRUE)
```

CGI1 analyses

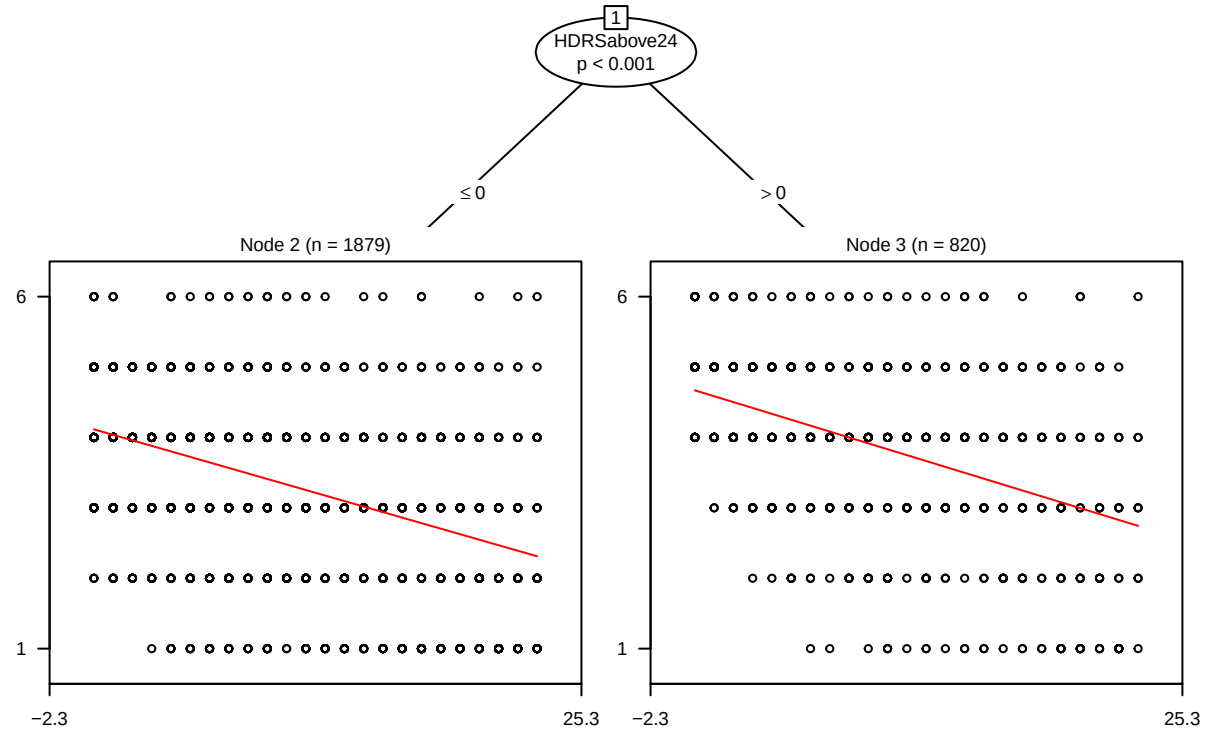
```
CGI1[CGI1 == -9999] <- NA
CGI1$dem1[CGI1$dem1 == "-9999"] <- NA
CGI1 <- CGI1[!is.na(CGI1$CGI1), ]
#sapply(CGI1, function(x) table(is.na(x)))
##
## cluster id: "nummer"
## time indicator: "time"
## response: "CGI1"
##
## additional random effect: "loc", "ptherapeut"
## I have excluded loc, because there are only three levels, and including loc
## as a random effect often yields convergence and/or singularity warnings.
## Beause ptherapeut is nested within loc, random effect of ptherapeut already
## accounts for loc.
##
## treatment indicators: "RAND2"
## partitioning variables (patient): "leeftijd", "sexe", "dem1", "dem2", "dem3",
## "dem4", "dem5", "dem6", "dem6A", "dem7", "dem8", "dem9", "dem10", "midy13b",
## "HDRSabove24"
##
## veel missings: "dem11", "dem12", "dem13", "dem13A", "dem13B", "midy13a", "ps",
## "dimtot", "tic21", "mide25", "mide26b", "mide29"
```

```
##
## partitioning variables (psychotherapeut): "ptherapeut", "pt_fctcat",
## "pt_fctcat2", "pt_sekse", "pt_leeftijd", "pt_ervaring", "pt_completer",
## "pt_zittingen", "pt_reden", "pt_eind", "ftherapeut"
##
## Erg veel missings: "ft_completer", "ft_reden", "ft_eind", "ftadditie_code",
## "ftadditie_momt",  "ftadditie_start"
##
vars <- c("nummer", "time", "CGI1", "loc", "ptherapeut", "RAND2", "leeftijd",
          "sexe", "dem1", "dem2", "dem3", "dem4", "dem5", "dem6", "dem6A",
          "dem7", "dem8", "dem9", "dem10", "midy13b", "HDRSabove24")
CGI1 <- CGI1[ , vars]
CGI1 <- CGI1[complete.cases(CGI1),]
levels(CGI1$time) <- gsub("cgi1_", "", levels(CGI1$time))
CGI1$time <- as.numeric(as.character(CGI1$time)) - 1
CGI1[ , sapply(CGI1, is.factor)] <- lapply(CGI1[ , sapply(CGI1, is.factor)], factor)
CGI1$nummer <- factor(CGI1$nummer)
#summary(CGI1)
```

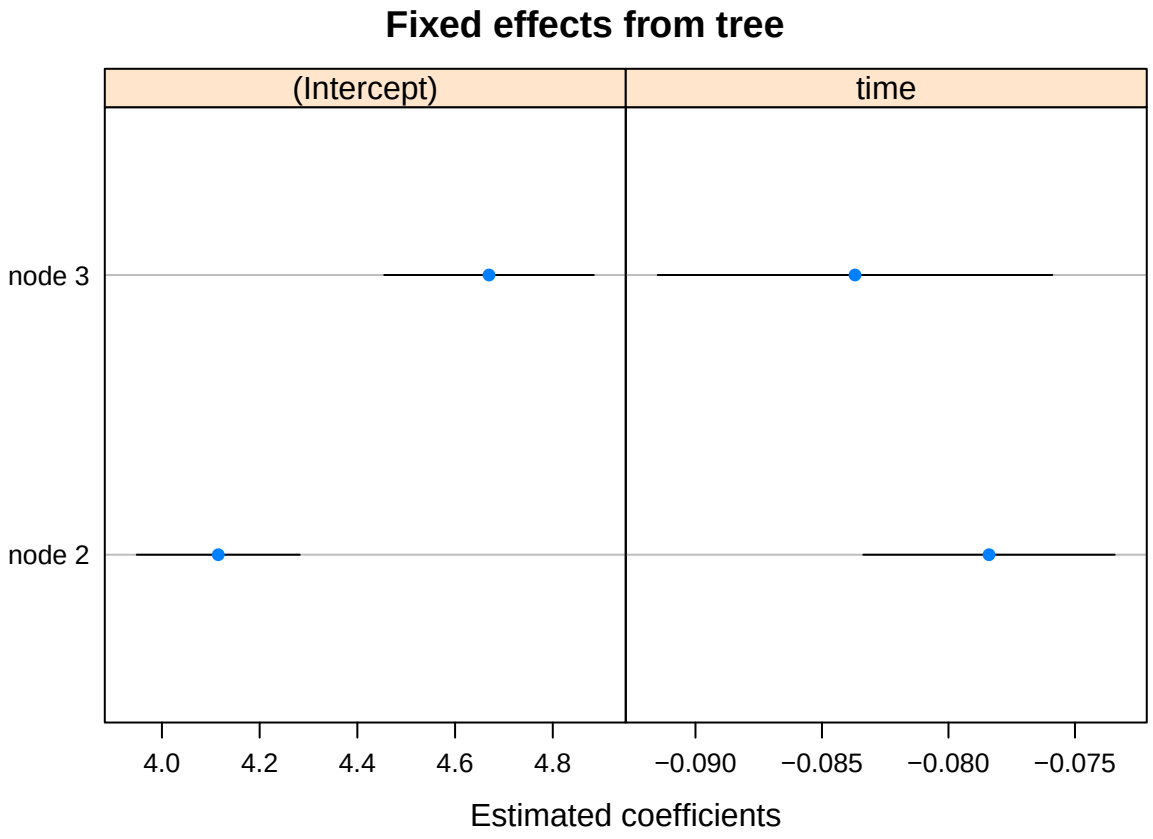
Fit the trees:

```
ltCGI1 <- lmertree(CGI1 ~ time | (1 | ptherapeut/nummer) |
                     RAND2 + leeftijd + sexe + dem1 + dem2 + dem3 + dem4 +
                     dem5 + dem6 + dem6A + dem7 + dem8 + dem9 + dem10 +
                     midy13b + HDRSabove24, data = CGI1, cluster = nummer)
save(ltCGI1, file = "ltCGI1")
```

```
load("ltCGI1")
plot(ltCGI1, which = "tree", fitted = "marginal", gp = gpar(cex = .6))
```




```
plot(ltCGI1, which = "tree.coef")
```

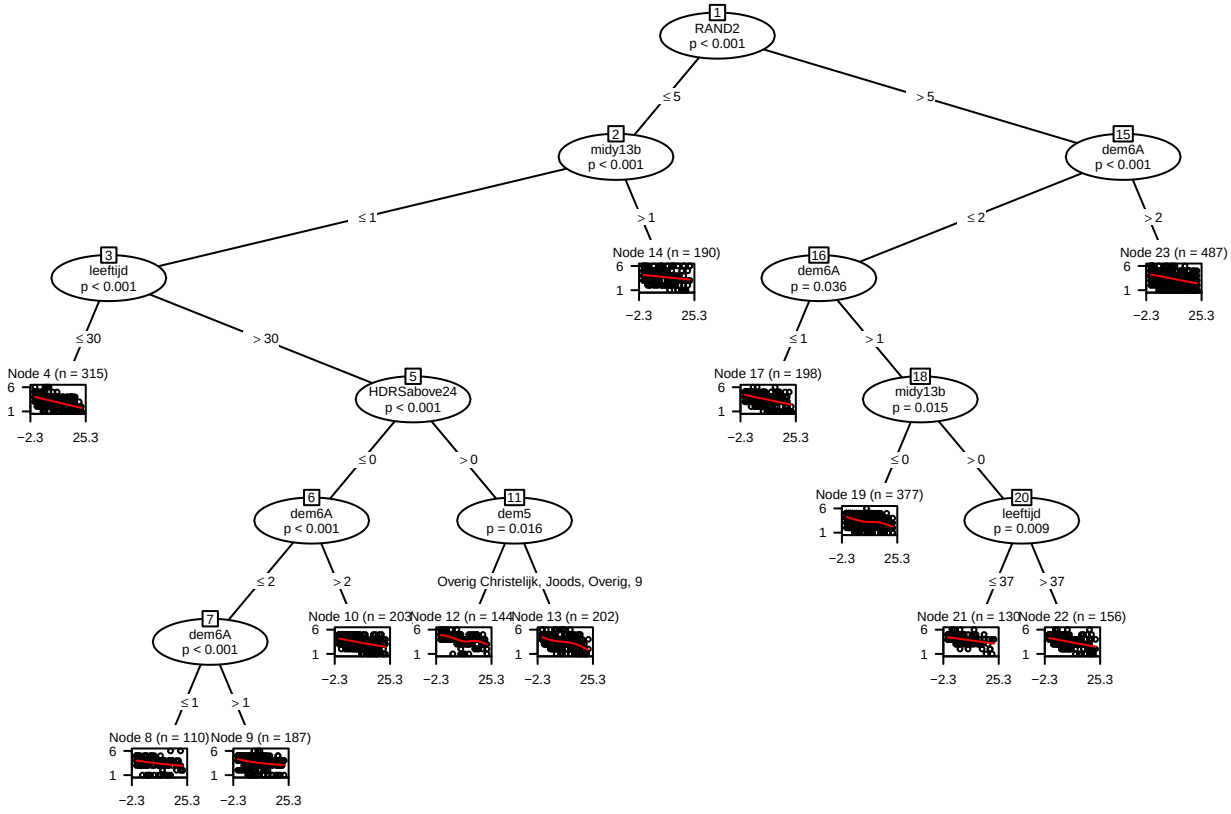



```
plot(ltCGI1, which = "random")
VarCorr(ltCGI1)
```

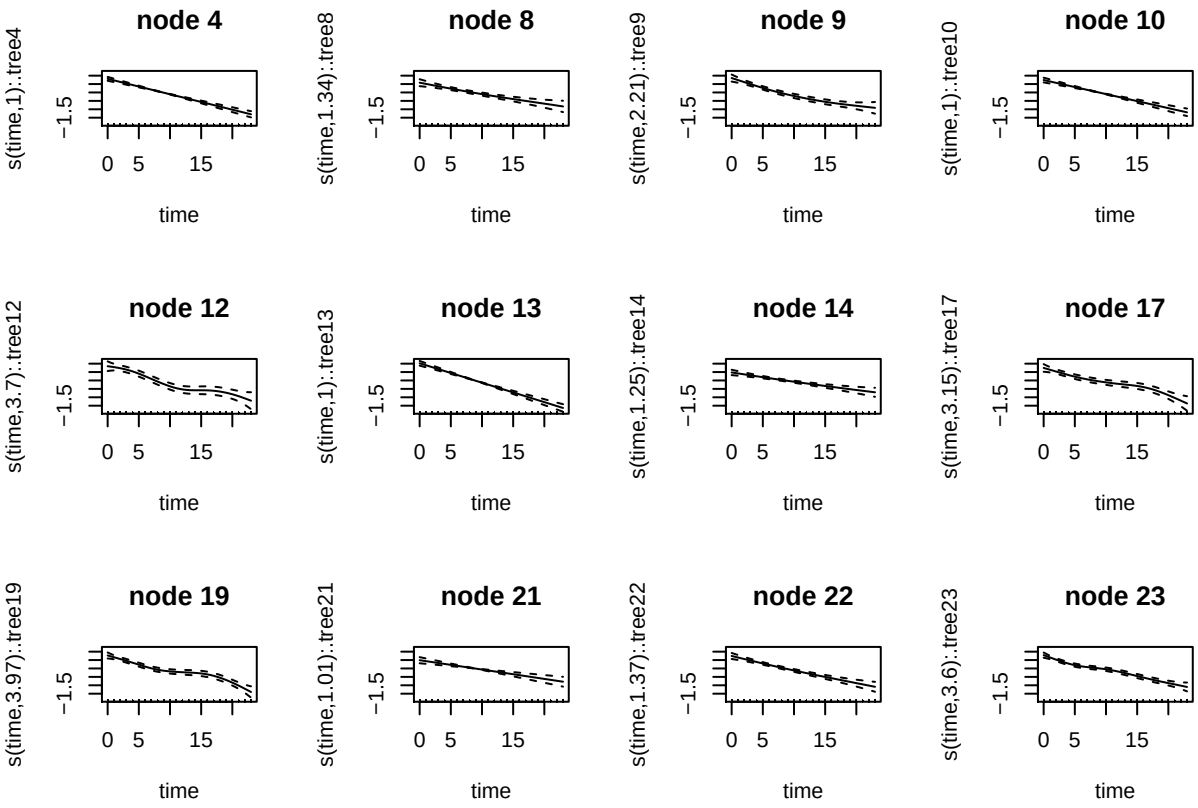
```
##  Groups             Name        Std.Dev.
##  nummer:ptherapeut (Intercept) 0.68827
##  ptherapeut        (Intercept) 0.46472
##  Residual                      0.70333
```

```
gtCGI1.1 <- gamtree(CGI1 ~ s(time) | s(ptherapeut, bs = "re") +
                  s(nummer, bs = "re") | RAND2 + leeftijd + sexe + dem1 + dem2 +
                  dem3 + dem4 + dem5 + dem6 + dem6A + dem7 + dem8 + dem9 +
                  dem10 + midy13b + HDRSabove24,
                data = CGI1, cluster = CGI1$nummer, verbose = FALSE)
save(gtCGI1.1, file = "gtCGI1.1")
```

```
load("gtCGI1.1")
plot(gtCGI1.1, which = "tree", treeplot_ctrl = list(gp = gpar(cex=.4)))
```

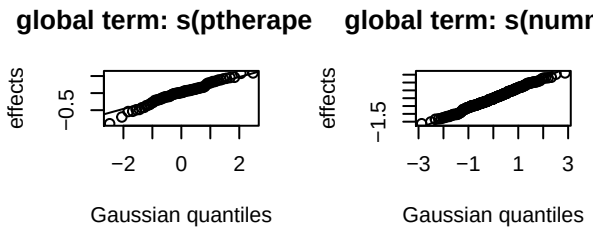



```
plot(gtCGI1.1, which = "nodes")
```




```
length(gtCGI1.1$tree)
```

```
## [1] 23
```




```
gtCGI1.2 <- gamtree(CGI1 ~ s(time) | s(ptherapeut, bs = "re") +
                    s(nummer, bs = "re") | RAND2 + leeftijd + sexe + dem1 + dem2 +
                    dem3 + dem4 + dem5 + dem6 + dem6A + dem7 + dem8 + dem9 +
                    dem10 + midy13b + HDRSabove24,
                  data = CGI1, globalstart = TRUE, verbose = FALSE)
save(gtCGI1.2, file = "gtCGI1.2")
```

```
load("gtCGI1.2")
#plot(gtCGI1.2, which = "tree", treeplot_ctrl = list(gp = gpar(cex=.6)))
#plot(gtCGI1.2, which = "nodes")
length(gtCGI1.2$tree)
```

```
## [1] 35
```

```
gtCGI1.3 <- gamtree(CGI1 ~ s(time) | s(ptherapeut, bs = "re") +
                    s(nummer, bs = "re") | RAND2 + leeftijd + sexe + dem1 + dem2 +
                    dem3 + dem4 + dem5 + dem6 + dem6A + dem7 + dem8 + dem9 +
                    dem10 + midy13b + HDRSabove24,
                  data = CGI1, cluster = CGI1$nummer, globalstart = TRUE)
save(gtCGI1.3, file = "gtCGI1.3")
```

```
load("gtCGI1.3")
#plot(gtCGI1.3, which = "tree", treeplot_ctrl = list(gp = gpar(cex=.6)))
#plot(gtCGI1.3, which = "nodes")
length(gtCGI1.3$tree)
```

```
## [1] 23
```

LMM trees yield the lowest number of splits. GAM trees yield a higher number of splits when initializing estimation with the (globally speicifed) random effects and not using the `cluster` argument. For GAM trees, initialization does not matter when the `cluster` argument is employed.

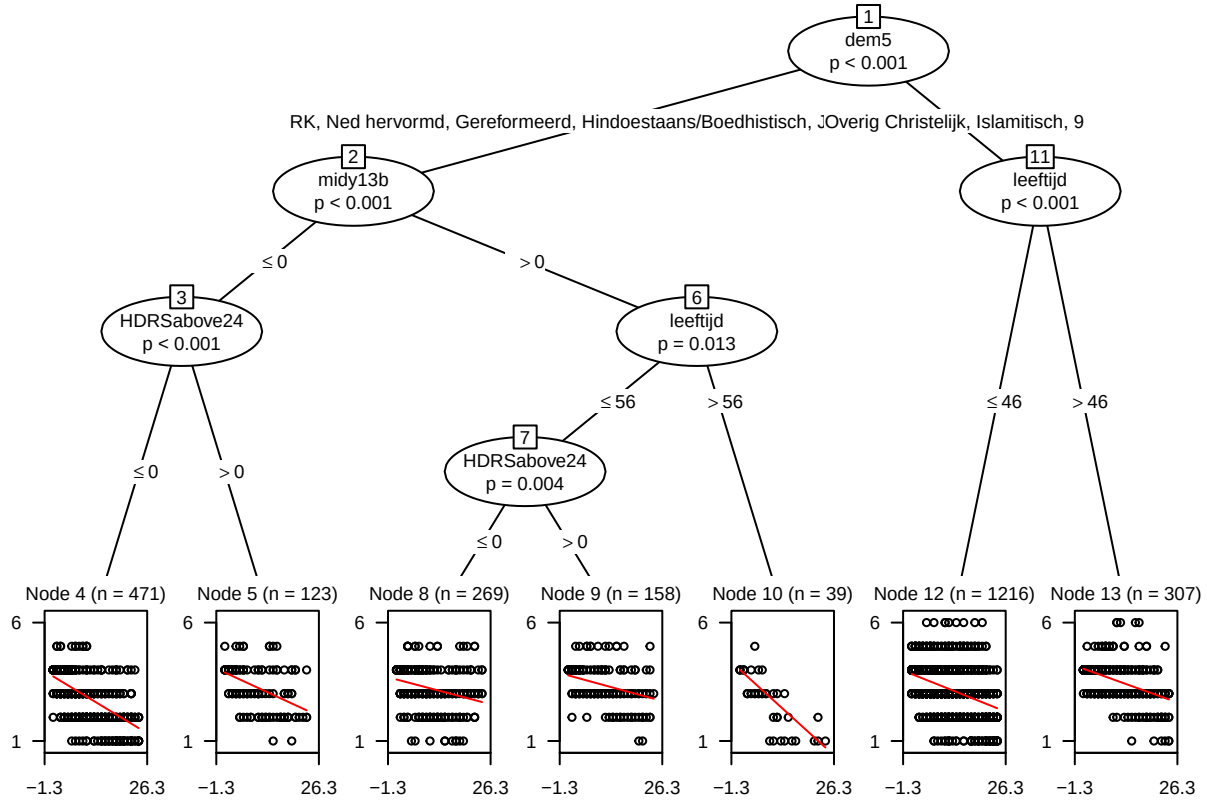
CGI2 analyses

```
CGI2[CGI2 == -9999] <- NA
CGI2$dem1[CGI2$dem1 == "-9999"] <- NA
CGI2 <- CGI2[!is.na(CGI2$CGI2), ]
#sapply(CGI2, function(x) table(is.na(x)))
vars <- c("nummer", "time", "CGI2", "loc", "ptherapeut", "RAND2", "leeftijd",
          "sexe", "dem1", "dem2", "dem3", "dem4", "dem5", "dem6", "dem6A",
          "dem7", "dem8", "dem9", "dem10", "midy13b", "HDRSabove24")
CGI2 <- CGI2[ , vars]
CGI2 <- CGI2[complete.cases(CGI2),]
levels(CGI2$time) <- gsub("cgi2_", "", levels(CGI2$time))
CGI2$time <- as.numeric(as.character(CGI2$time))
CGI2[ , sapply(CGI2, is.factor)] <- lapply(CGI2[ , sapply(CGI2, is.factor)], factor)
CGI2$nummer <- factor(CGI2$nummer)
#summary(CGI2)
```

We fit the trees:

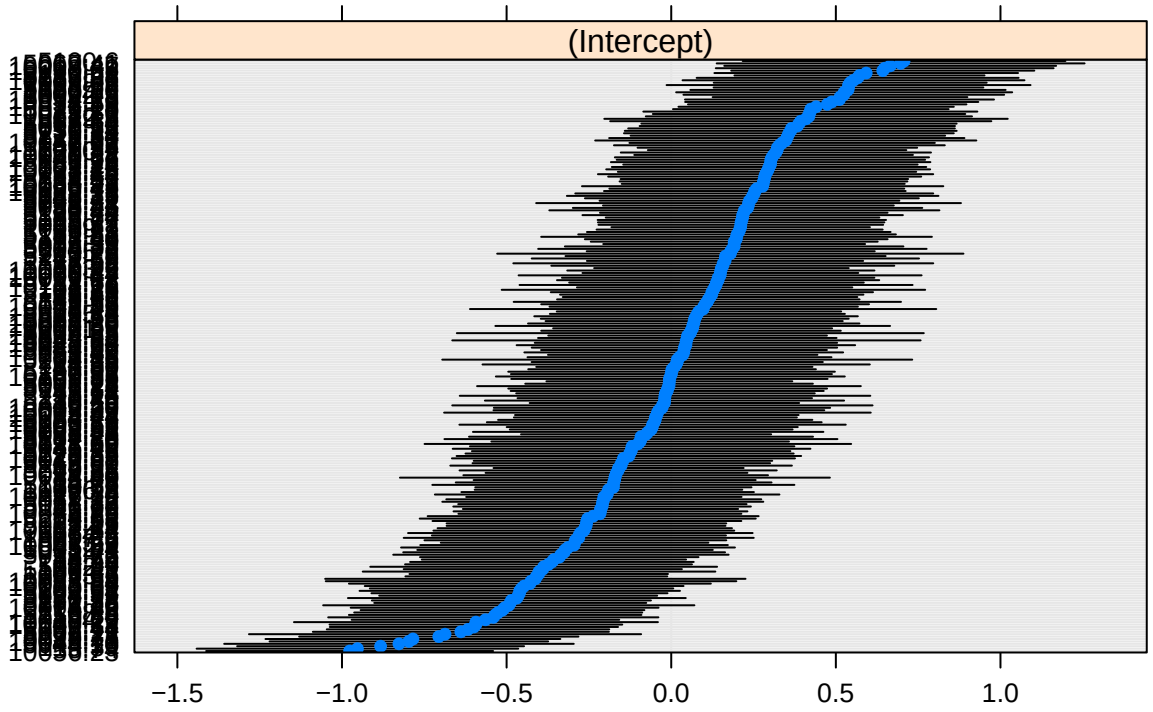
```
ltCGI2 <- lmertree(CGI2 ~ time | (1 | ptherapeut/nummer) |
                     RAND2 + leeftijd + sexe + dem1 + dem2 + dem3 + dem4 +
                     dem5 + dem6 + dem6A + dem7 + dem8 + dem9 + dem10 +
                     midy13b + HDRSabove24,
                   data = CGI2, cluster = nummer)
save(ltCGI2, file = "ltCGI2")
```

```
load("ltCGI2")
plot(ltCGI2, which = "tree", fitted = "marginal", gp = gpar(cex=.6))
```

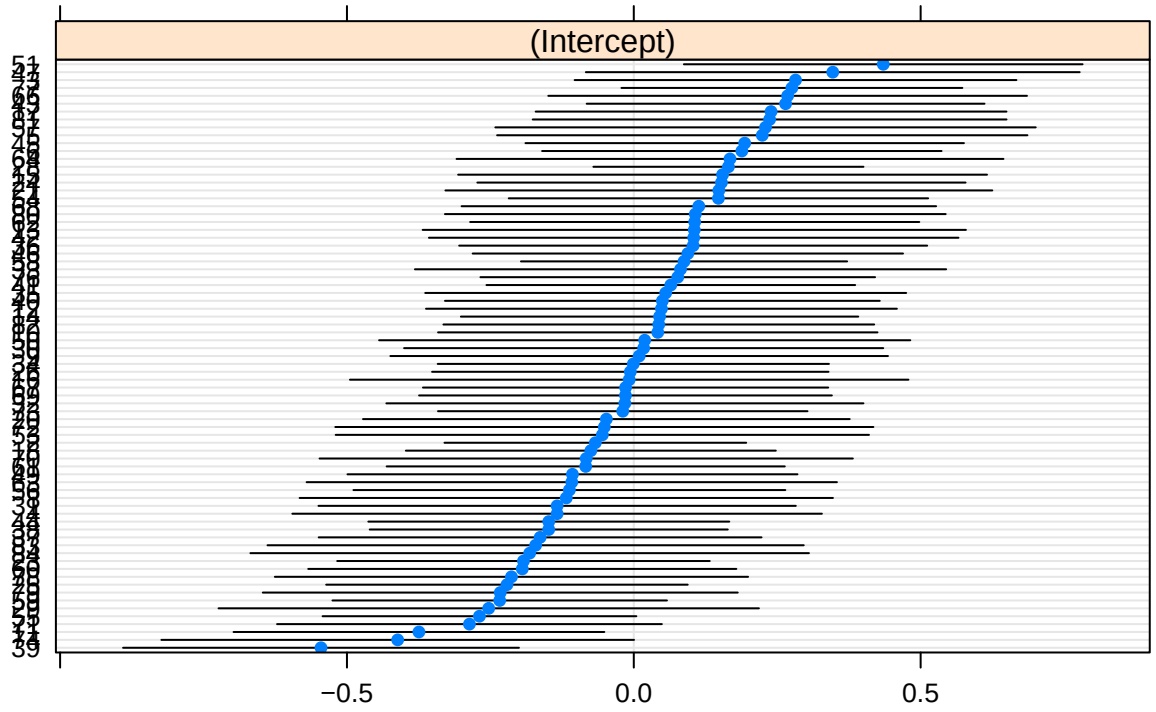



```
plot(ltCGI2, which = "ranef")
```

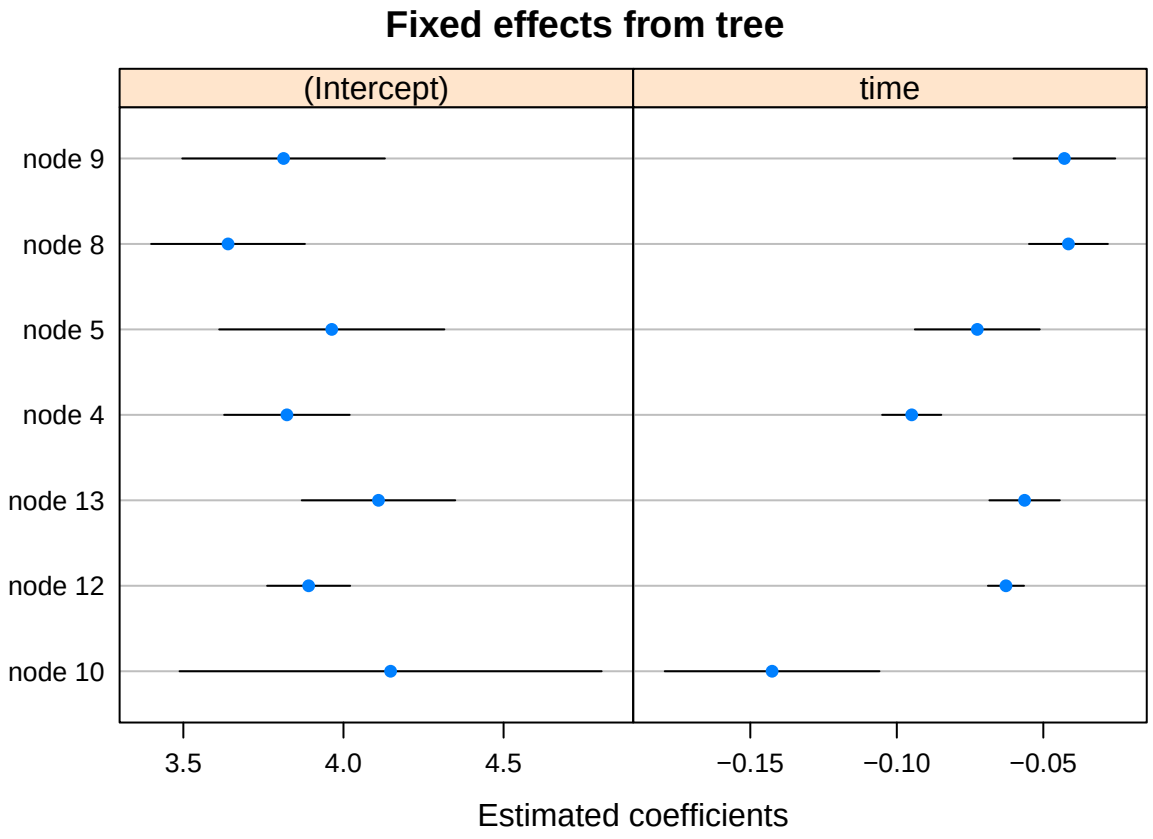
```
## $`nummer:ptherapeut`
```




```
##
## $ptherapeut
```




```
plot(ltCGI2, which = "tree.coef")
```

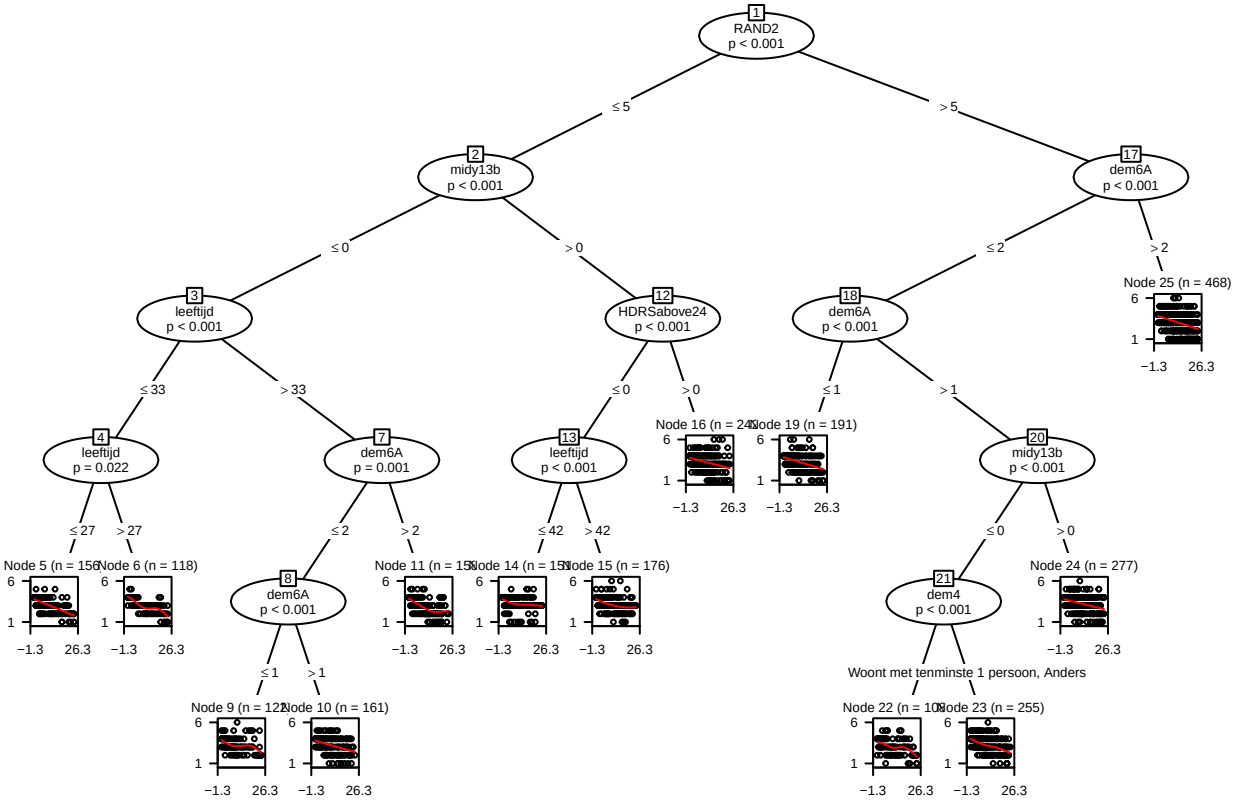



```
VarCorr(ltCGI2)
```

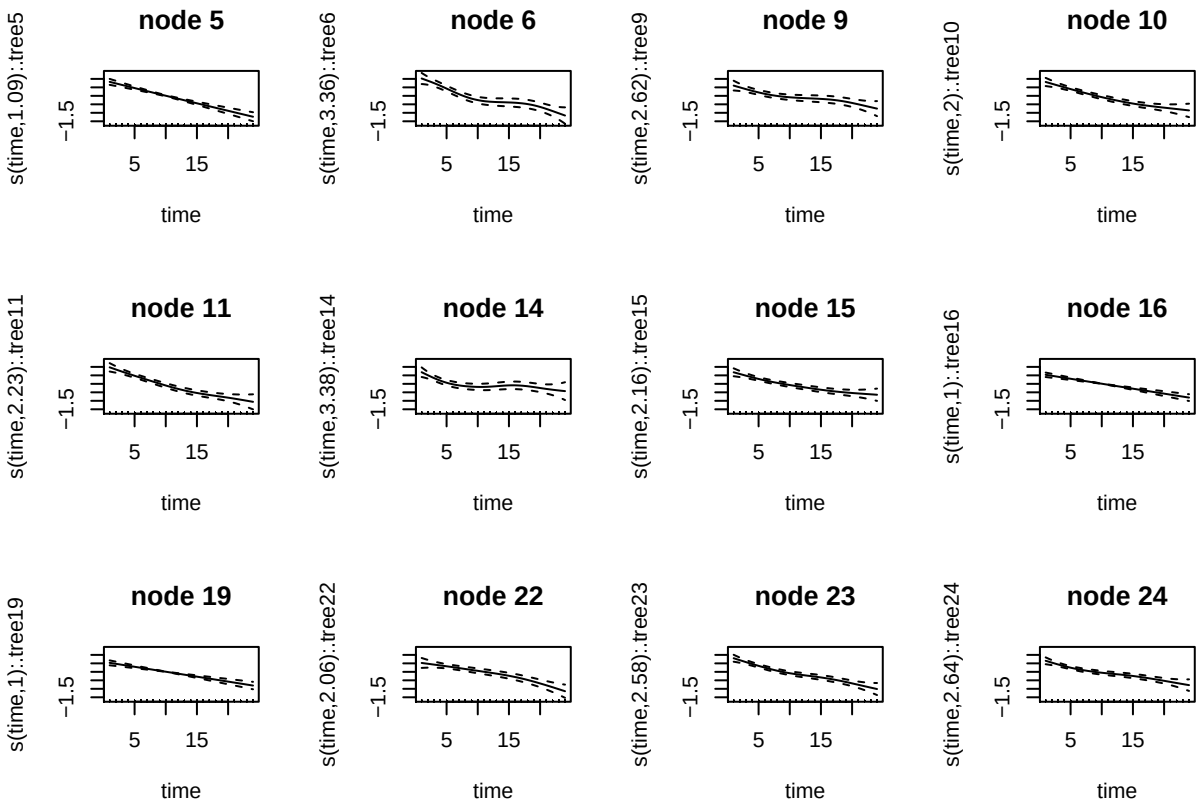
```
##  Groups             Name        Std.Dev.
##  nummer:ptherapeut (Intercept) 0.42069
##  ptherapeut        (Intercept) 0.27468
##  Residual                      0.69497
```

```
gtCGI2.1 <- gamtree(CGI2 ~ s(time) | s(ptherapeut, bs = "re") +
                      s(nummer, bs = "re") | RAND2 + leeftijd + sexe + dem1 + dem2 +
                      dem3 + dem4 + dem5 + dem6 + dem6A + dem7 + dem8 + dem9 +
                      dem10 + midy13b + HDRSabove24,
                    data = CGI2, cluster = CGI2$nummer, verbose = FALSE)
save(gtCGI2.1, file = "gtCGI2.1")
```

```
load("gtCGI2.1")
plot(gtCGI2.1, which = "tree", treeplot_ctrl = list(gp = gpar(cex=.4)))
```

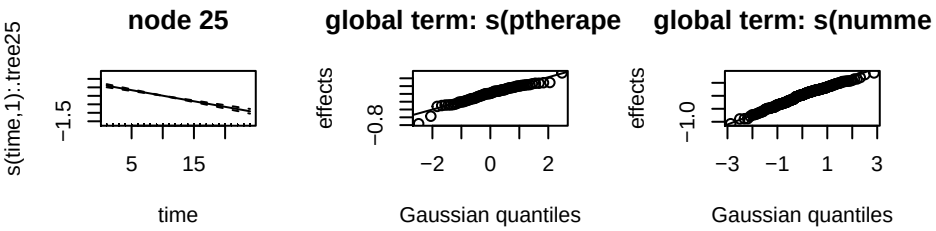



```
plot(gtCGI2.1, which = "nodes")
```




```
length(gtCGI2.1$tree)
```

```
## [1] 25
```




```
gtCGI2.2 <- gamtree(CGI2 ~ s(time) | s(ptherapeut, bs = "re") +
                      s(nummer, bs = "re") | RAND2 + leeftijd + sexe + dem1 + dem2 +
                      dem3 + dem4 + dem5 + dem6 + dem6A + dem7 + dem8 + dem9 +
                      dem10 + midy13b + HDRSabove24,
                    data = CGI2, globalstart = TRUE, verbose = FALSE)
save(gtCGI2.2, file = "gtCGI2.2")
```

```
load("gtCGI2.2")
#plot(gtCGI2.2, which = "tree", treeplot_ctrl = list(gp = gpar(cex=.6)))
#plot(gtCGI2.2, which = "nodes")
length(gtCGI2.2$tree)
```

```
## [1] 33
```

```
gtCGI2.3 <- gamtree(CGI2 ~ s(time) | s(ptherapeut, bs = "re") +
                      s(nummer, bs = "re") | RAND2 + leeftijd + sexe + dem1 + dem2 +
                      dem3 + dem4 + dem5 + dem6 + dem6A + dem7 + dem8 + dem9 +
                      dem10 + midy13b + HDRSabove24,
                    data = CGI2, cluster = CGI2$nummer, globalstart = TRUE,
                    verbose = FALSE)
save(gtCGI2.3, file = "gtCGI2.3")
```

```
load("gtCGI2.3")
#plot(gtCGI2.3, which = "tree", treeplot_ctrl = list(gp = gpar(cex=.6)))
#plot(gtCGI2.3, which = "nodes")
length(gtCGI2.3$tree)
```

```
## [1] 25
```

LMM trees yield the lowest number of splits. GAM trees yield a higher number of splits when initializing estimation with the (globally speicifed) random effects and not using the `cluster` argument. For GAM trees, initialization does not matter when the `cluster` argument is employed.

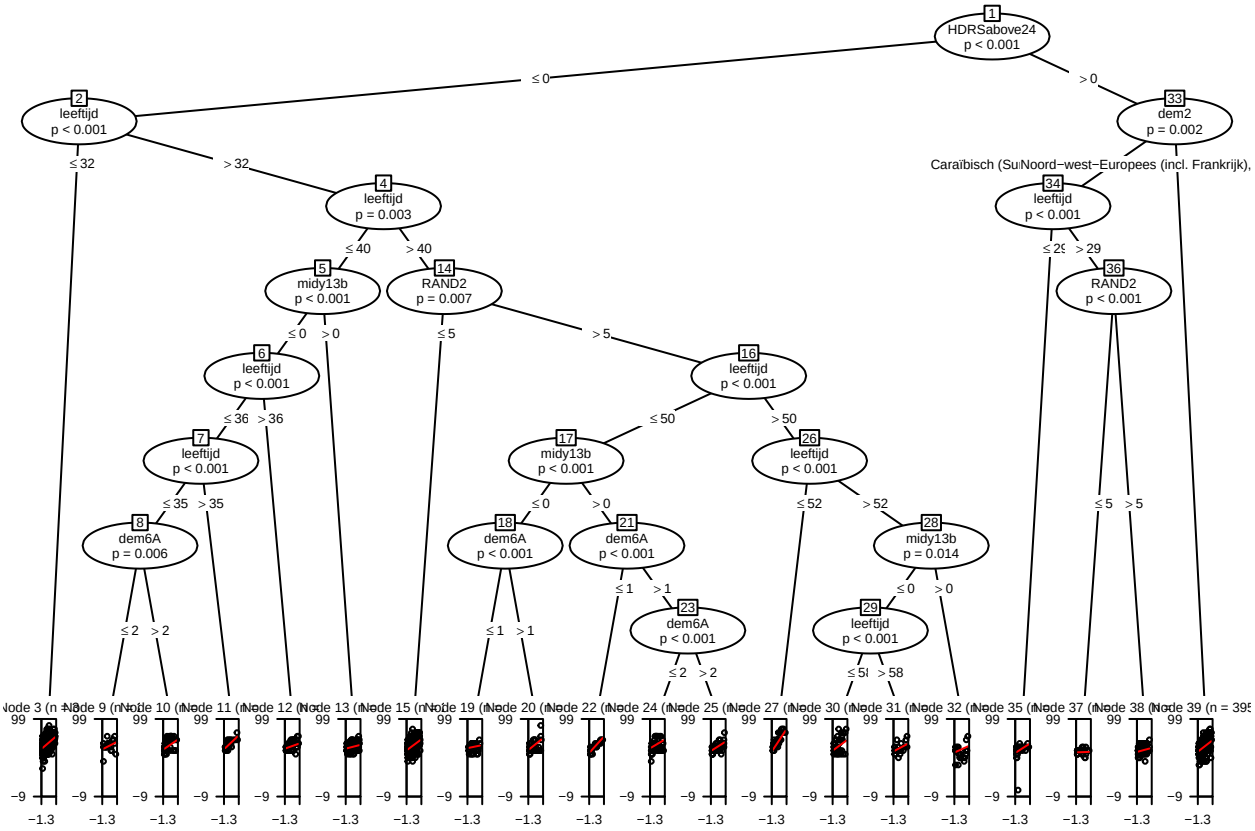
GAF analyses

```
GAF[GAF == -9999] <- NA
GAF$dem1[GAF$dem1 == "-9999"] <- NA
GAF <- GAF[!is.na(GAF$GAF), ]
#sapply(GAF, function(x) table(is.na(x)))
vars <- c("nummer", "time", "GAF", "loc", "ptherapeut", "RAND2", "leeftijd",
          "sexe", "dem1", "dem2", "dem3", "dem4", "dem5", "dem6", "dem6A",
          "dem7", "dem8", "dem9", "dem10", "midy13b", "HDRSabove24")
GAF <- GAF[ , vars]
GAF <- GAF[complete.cases(GAF),]
levels(GAF$time) <- gsub("gaf_", "", levels(GAF$time))
GAF$time <- as.numeric(as.character(GAF$time))
GAF[ , sapply(GAF, is.factor)] <- lapply(GAF[ , sapply(GAF, is.factor)], factor)
GAF$nummer <- factor(GAF$nummer)
#summary(GAF)
```

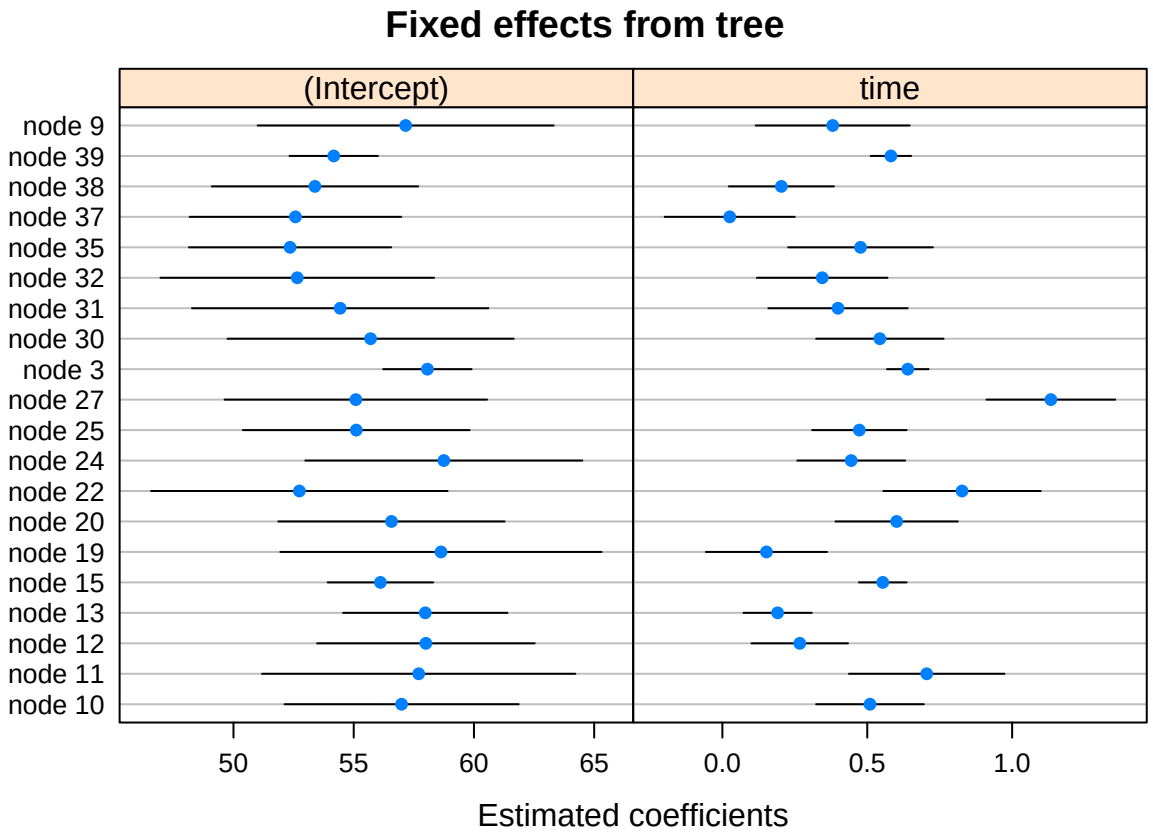
We fit the trees:

```
ltGAF <- lmertree(GAF ~ time | (1 | ptherapeut/nummer) |
                    RAND2 + leeftijd + sexe + dem1 + dem2 + dem3 + dem4 +
                    dem5 + dem6 + dem6A + dem7 + dem8 + dem9 + dem10 +
                    midy13b + HDRSabove24,
                  data = GAF, cluster = nummer)
save(ltGAF, file = "ltGAF")
```

```
load("ltGAF")
plot(ltGAF, which = "tree", fitted = "marginal", gp = gpar(cex=.4))
```

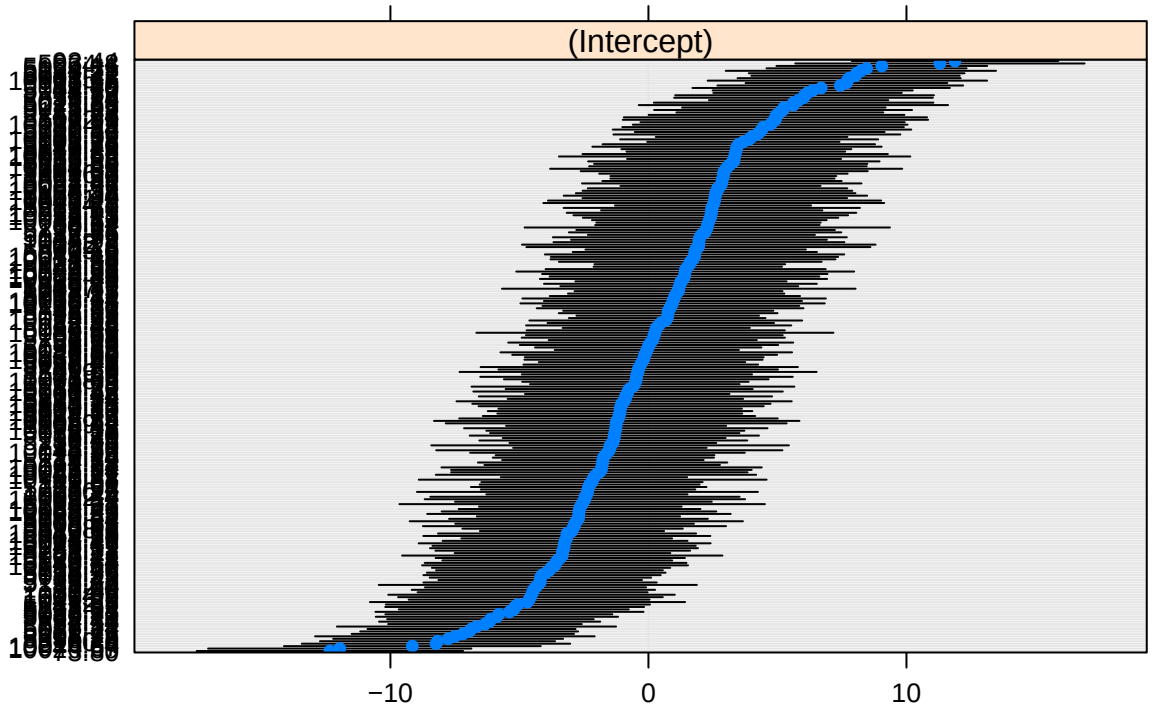



```
plot(ltGAF, which = "tree.coef")
```

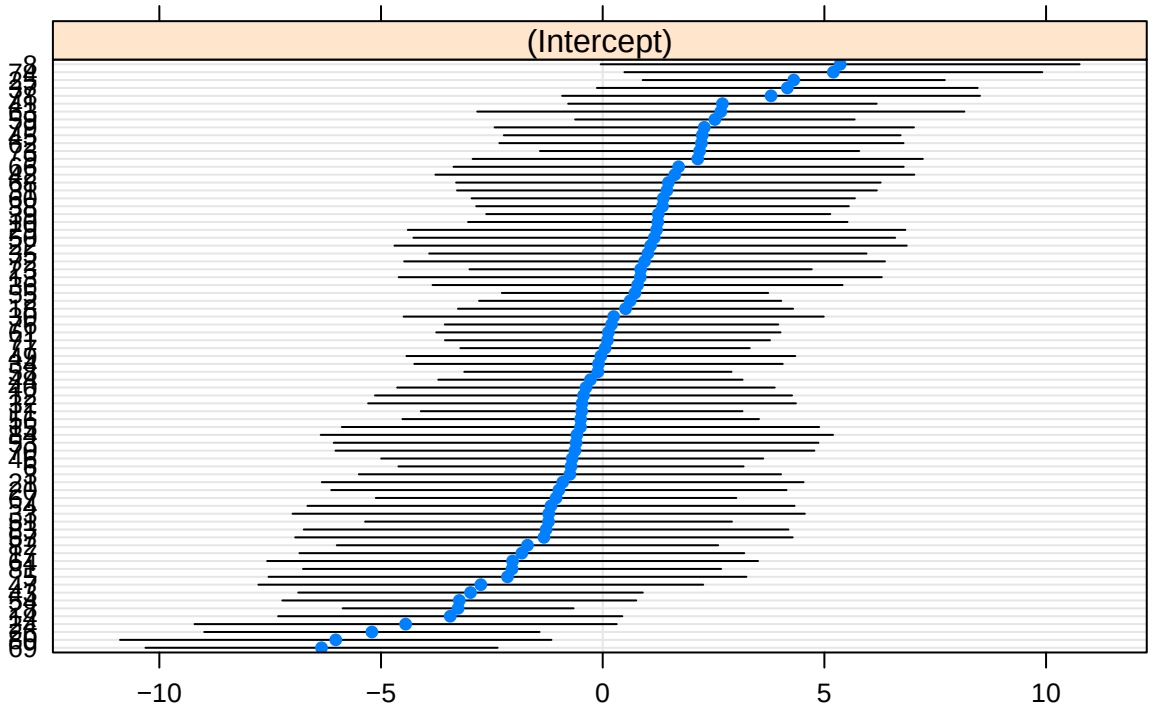



```
plot(ltGAF, which = "ranef")
```

```
## $`nummer:ptherapeut`
```




```
##
## $ptherapeut
```




```
VarCorr(ltGAF)
```

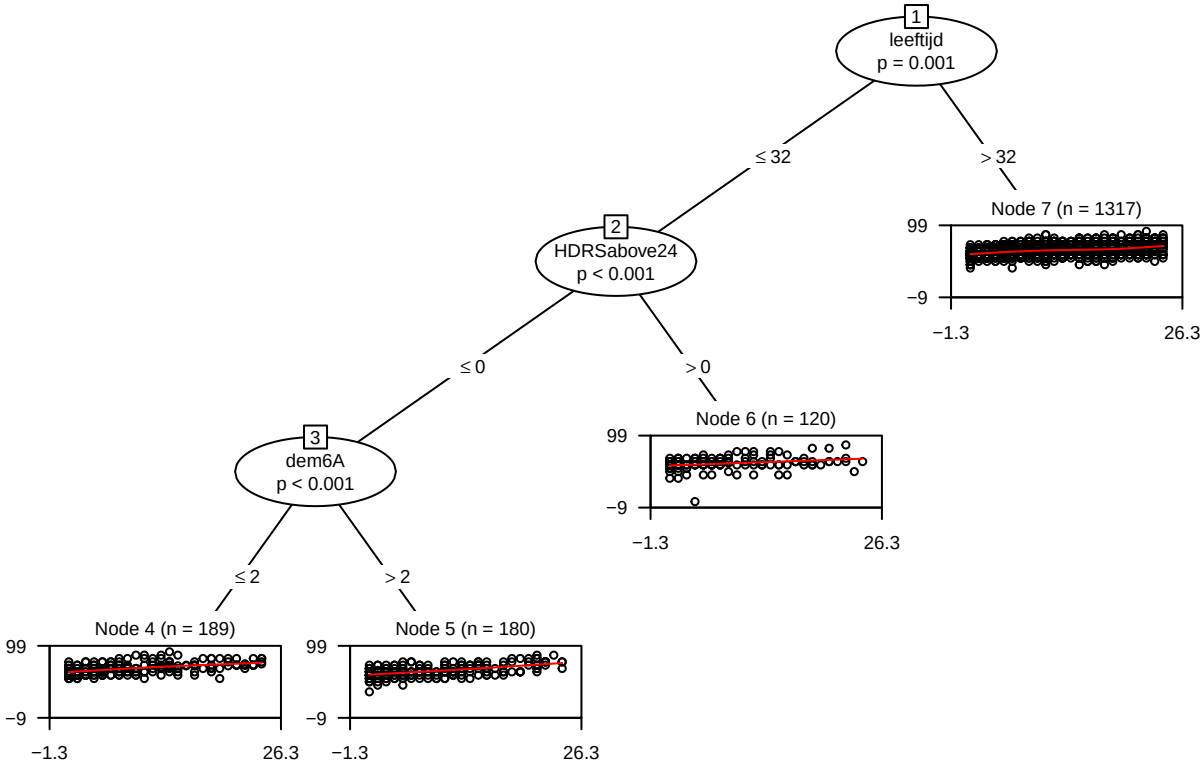
```
##  Groups             Name        Std.Dev.
##  nummer:ptherapeut (Intercept) 4.8042
##  ptherapeut        (Intercept) 3.3077
##  Residual                      4.3733
```

```
length(ltGAF$tree)
```

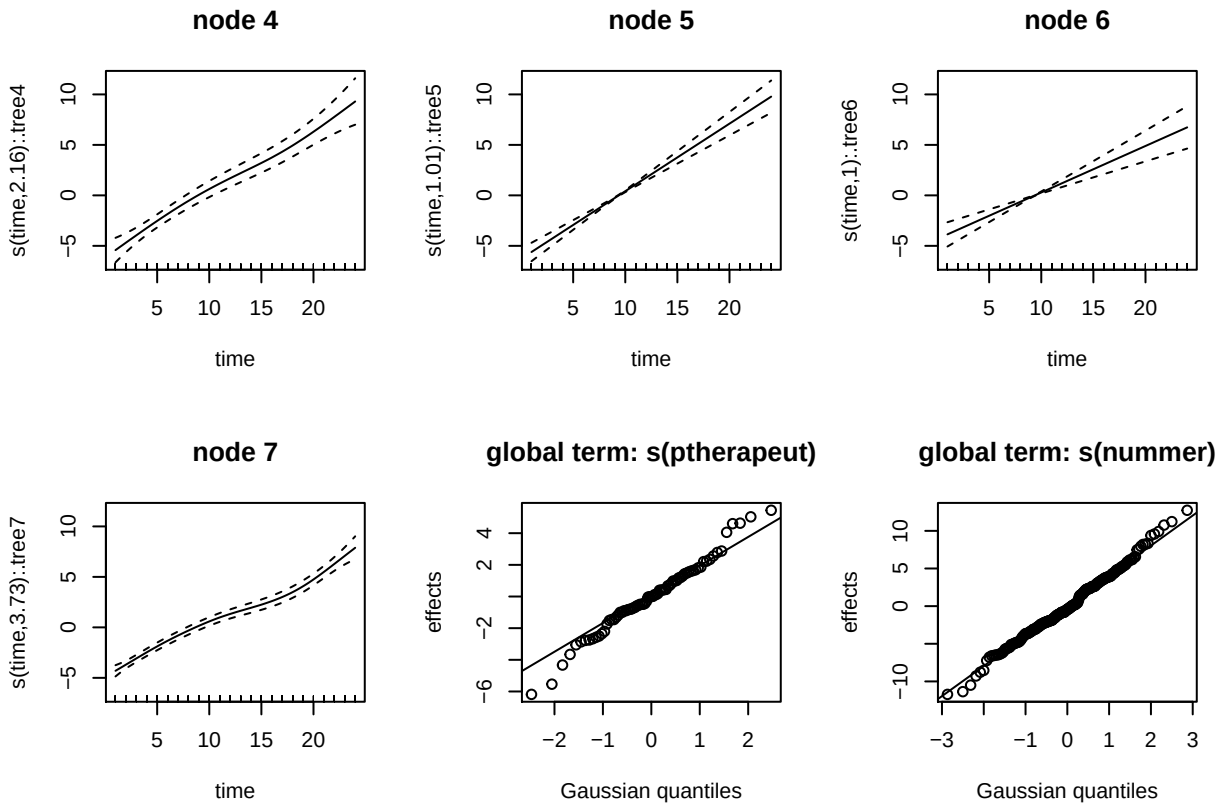
```
## [1] 39
```

```
gtGAF.1 <- gamtree(GAF ~ s(time) | s(ptherapeut, bs = "re") +
                     s(nummer, bs = "re") | RAND2 + leeftijd + sexe + dem1 + dem2 +
                     dem3 + dem4 + dem5 + dem6 + dem6A + dem7 + dem8 + dem9 +
                     dem10 + midy13b + HDRSabove24,
                   data = GAF, cluster = GAF$nummer, verbose = FALSE)
save(gtGAF.1, file = "gtGAF.1")
```

```
load("gtGAF.1")
plot(gtGAF.1, which = "tree", treeplot_ctrl = list(gp = gpar(cex=.6)))
```




```
plot(gtGAF.1, which = "nodes")
```




```
length(gtGAF.1$tree)
```

```
## [1] 7
```

```
gtGAF.2 <- gamtree(GAF ~ s(time) | s(ptherapeut, bs = "re") +
                     s(nummer, bs = "re") | RAND2 + leeftijd + sexe + dem1 + dem2 +
                     dem3 + dem4 + dem5 + dem6 + dem6A + dem7 + dem8 + dem9 +
                     dem10 + midy13b + HDRSabove24,
                   data = GAF, globalstart = TRUE, verbose = FALSE)
save(gtGAF.2, file = "gtGAF.2")
```

```
load("gtGAF.2")
#plot(gtGAF.2, which = "tree", treeplot_ctrl = list(gp = gpar(cex=.6)))
#plot(gtGAF.2, which = "nodes")
length(gtGAF.2$tree)
```

```
## [1] 17
```

```
gtGAF.3 <- gamtree(GAF ~ s(time) | s(ptherapeut, bs = "re") +
                     s(nummer, bs = "re") | RAND2 + leeftijd + sexe + dem1 + dem2 +
                     dem3 + dem4 + dem5 + dem6 + dem6A + dem7 + dem8 + dem9 +
                     dem10 + midy13b + HDRSabove24,
                   data = GAF, cluster = GAF$nummer, globalstart = TRUE,
                   verbose = FALSE)
save(gtGAF.3, file = "gtGAF.3")
```

```
load("gtGAF.3")
#plot(gtGAF.3, which = "tree", treeplot_ctrl = list(gp = gpar(cex=.6)))
```

```
#plot(gtGAF.3, which = "nodes")
length(gtGAF.3$tree)
```

Again: LMM trees yield the lowest number of splits. GAM trees yield a higher number of splits when initializing estimation with the (globally specified) random effects and not using the `cluster` argument. For GAM trees, initialization does not seem to matter when the `cluster` argument is employed.

Compare accuracy and sparsity

```
nfolds <- 10
MSEs <- tree_sizes <- data.frame(CGI1.lt = rep(NA, times = nfolds),
                                 CGI1.gt = rep(NA, times = nfolds),
                                 CGI2.lt = rep(NA, times = nfolds),
                                 CGI2.gt = rep(NA, times = nfolds),
                                 GAF.lt = rep(NA, times = nfolds),
                                 GAF.gt = rep(NA, times = nfolds))
dataset_counter <- 0
for (dataset in list(CGI1, CGI2, GAF)) {

  dataset_counter <- dataset_counter + 1
  print(paste(c("CGI1", "CGI2", "GAF")[dataset_counter]))

  ## Create vector of fold ids
  set.seed(42)
  Nc <- length(unique(dataset$nummer))
  folds <- sample(rep(1:nfolds, each = (Nc/nfolds)+1), size = Nc)
  fold_ids <- folds[as.numeric(dataset$nummer)]

  ## Prepare formulas
  lt_formula <- formula(paste0(names(dataset)[3],
      "~ time | (1 | ptherapeut/nummer) | RAND2 +
      leeftijd + sexe + dem1 + dem2 + dem3 + dem4 +
      dem5 + dem6 + dem6A + dem7 + dem8 + dem9 + dem10 +
      midy13b + HDRSabove24"))
  gt_formula <- formula(paste0(names(dataset)[3],
      "~ s(time) | s(ptherapeut, bs = \"re\") +
      s(nummer, bs = \"re\") | RAND2 + leeftijd + sexe + dem1 + dem2 +
      dem3 + dem4 + dem5 + dem6 + dem6A + dem7 + dem8 + dem9 + dem10 +
      midy13b + HDRSabove24"))

  for (fold in 1:6) {

    print(paste("Fold", fold))

    ## Prepare train and test data
    traindata <- dataset[fold_ids != fold, ]
    testdata <- dataset[fold_ids == fold, ]
    if (dataset_counter == 2L & fold == 7L) {
      testdata <- testdata[-which(testdata$nummer == 5124), ]
      ## otherwise, dem5 has only one observation with the level "Joods"
      ## In CGI2, dem5 is used for splitting, yielding an error
    }
```

```
    ## Fit lmertree and evaluate
    column_id <- (dataset_counter * 2) - 1
    lt <- lmertree(lt_formula, data = traindata, cluster = nummer)
    tree_sizes[fold, column_id] <- length(lt$tree)
    lt_preds <- predict(lt, newdata = testdata, re.form = NA)
    MSEs[fold, column_id] <- mean((testdata[, 3] - lt_preds)^2)

    ## Fit gamtree and evaluate
    column_id <- (dataset_counter * 2)
    gt <- gamtree(gt_formula, data = traindata, cluster = traindata$nummer,
                  verbose = FALSE)
    tree_sizes[fold, column_id] <- length(gt$tree)
    gt_preds <- suppressWarnings(predict(gt, newdata = testdata, type = "terms"))
      ## (get warnings because levels of nummer and/or ptherapeut
      ## were present not in training data, but these are excluded anyway)
    columns_to_exclude <- sapply(c("ptherapeut", "nummer"),
                                 function(x) grep(x, colnames(gt_preds)))
    gt_preds <- rowSums(gt_preds[ , -columns_to_exclude])
    MSEs[fold, column_id] <- mean((testdata[, 3] - gt_preds)^2)

  }
}

save(MSEs, file = "MSEs")
save(tree_sizes, file = "tree_sizes")
```

```
load(file = "MSEs")
load(file = "tree_sizes")
cbind(MSE = sort(sapply(MSEs, mean)),
      SE_MSE = sapply(MSEs, sd)[order(sapply(MSEs, mean))]/sqrt(nrow(MSEs)))
```

```
##                MSE     SE_MSE
## CGI2.lt  0.7950368 0.04499501
## CGI2.gt  0.8374043 0.05832140
## CGI1.gt  1.2549439 0.05508624
## CGI1.lt  1.2636281 0.06279685
## GAF.gt  60.3735799 3.86792857
## GAF.lt  66.5942944 4.73249629
```

LMM trees are more accurate than GAM trees `CGI2` data, GAM trees more accurate on `CGI1` and `GAF` data. Furthermore, higher MSEs also have higher standard error.

```
cbind(M_tree_size = sort(sapply(tree_sizes, mean)),
      SE_tree_size = sort(sapply(tree_sizes, sd))[order(sapply(tree_sizes, mean))] /
        sqrt(nrow(MSEs)))
```

```
##         M_tree_size SE_tree_size
## CGI2.lt         4.8    1.4847372
## GAF.gt         14.6    8.1475286
## CGI1.gt        20.2    1.1333333
## CGI2.gt        24.0    1.6653328
## CGI1.lt        29.4    0.9545214
## GAF.lt         33.6    4.1101500
```

LMM trees are more sparse on `CGI2`, GAM trees more sparse on `CGI1` and `GAF`. GAM trees have larger standard error of tree size on all three datasets. Thus, it appears that sparsest method is also most accurate,

and GAM tree has higher tree size variablity.