Fitting GLMM trees using R package glmertree

This manual illustrates how GLMM trees can be fitted using **R** package **glmertree** through several data-analytic examples. The package can be installed as follows:

```
install.packages("glmertree")
```

After installation, the package can be loaded as follows:

```
library("glmertree")
```

Dataset

In the examples, we will use an artificial dataset, generated so as to mimic the results for the unadjusted treatment outcome. This outcome reflects SDQ mental-health difficulty scores assessed approximately 4-8 months after baseline assessment, corrected for the baseline assessment, and then standardized. Thus, higher values indicate poorer outcomes.

Specifically, we generated 3256 observations and 18 uncorrelated predictor variables: three sociodemographic variables, nine case characteristics and six impact indicators. These variables were generated according to the univariate statistics reported in Table 1 of the main paper. Furthermore, the node-specific means for the outcome variable were generated according to the partition and values presented in Figure 1 of the main paper. Higher values of the outcome variable (`outcome`) indicate poorer treatment outcomes. Finally, a random intercept with respect to 13 different service providers was generated, following a normal distribution with mean zero and variance set so as to yield an intra-class correlation of about .06.

The artificial dataset is included as supplementary material in the file "MHserv_data.txt". After this file is placed in the current working directory, it can be loaded as follows:

```
MHserv_data <- read.table("MHserv_data.txt",
                          colClasses = c(ethnicity = "factor",
                                         cluster_id = "factor"))
```

We used the `colClasses` argument, so as to specify `ethnicity` and `cluster_id` to be of class `factor`, to prevent these variables being interpreted as numeric.

To inspect the data, we request the number of rows and columns:

```
dim(MHserv_data)
```

```
## [1] 3256   20
```

Next, we request a summary of the variables in the dataset:

```
summary(MHserv_data)
```

```
##       age           gender     ethnicity emotional  autism     conduct
##  Min.   : 4.00   female:1551   1:2109    no :1468   no :2971   no :2623
##  1st Qu.: 8.90   male  :1705   2: 168    yes:1788   yes: 285   yes: 633
##  Median :11.40                 3: 237
##  Mean   :11.28                 4: 206
##  3rd Qu.:13.60                 5: 155
##  Max.   :18.00                 6: 381
##
##  eating     self_harm  hyperactivity  SEN       other_problems
##  no :3098   no :2589   no :2916      no :2997   no :2864
##  yes: 158   yes: 667   yes: 340      yes: 259   yes: 392
##
##
##
##
##
##  infrequent_characteristics problem_duration    distress
##  no :2996                   Min.   :1.000    Min.   :1.000
##  yes: 260                   1st Qu.:1.000    1st Qu.:1.000
##                             Median :2.000    Median :1.000
##                             Mean   :1.903    Mean   :1.674
##                             3rd Qu.:2.000    3rd Qu.:2.000
##                             Max.   :5.000    Max.   :3.000
##
##     homelife        friends        classroom        leisure
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :1.000   Median :1.000   Median :1.000   Median :1.000
##  Mean   :1.583   Mean   :1.448   Mean   :1.543   Mean   :1.411
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :3.000   Max.   :3.000   Max.   :3.000   Max.   :3.000
##
##    cluster_id      outcome
##  5      : 274   Min.   :-1.29000
##  13     : 264   1st Qu.:-0.38000
##  9      : 264   Median :-0.10000
##  10     : 263   Mean   :-0.08178
##  12     : 259   3rd Qu.: 0.20000
##  1      : 258   Max.   : 1.56000
##  (Other):1674
```

The results show that `age` is a numerical variable, and `gender` and `ethnicity` are categorical variables (factors). Variables `emotional` through `infrequent_characteristics` are binary coded case characteristics. Variables `problem_duration` through `leisure` are numerically coded indicators for the severity of mental-health problems: duration (rated on a 5-point scale ranging from absent to > 1 year), overall distress, and impairment on home life, friendships, classroom performance, and leisure activities (all rated on a 3-point scale ranging from little or no severity to high severity). Finally, `cluster_id` is an indicator for mental-health service provider, and `outcome` is the treatment outcome to be predicted.

Fitting a mixed-effects regression tree for a continuous outcome variable

In this first example, we predict the continuous outcome variable using all potential partitioning variables. Furthermore, we estimate a random intercept with respect to the indicator for mental-health service provider (i.e., the `cluster_id` variable). We employ the `lmertree()` function, which requires the user to specify a `formula` and `data` argument, respectively:

```
lmm_tree <- lmertree(outcome ~ 1 | cluster_id | age + gender + ethnicity +
                       autism + conduct + emotional + autism + conduct +
                       eating + self_harm + hyperactivity + SEN +
                       other_problems + infrequent_characteristics +
                       problem_duration + distress + homelife + friends +
                       classroom + leisure, data = MHserv_data)
```

With the first argument (`formula`), we specified the model to be fitted to the data. This formula consists of a left- and right-hand part: The left-hand part, preceding the tilde symbol, specifies the outcome variable (`outcome`). The right-hand part consists of three subparts, separated by vertical bars: The first part specifies the predictor variable(s) of the linear model, the second part specifies the random effects and the third part specifies the potential partitioning variables.

We did not specify any predictor variables for the linear model, because we are interested in predicting values of the treatment outcome only. We therefore specified an intercept-only model (`~ 1`). We specified a single variable in the random-effects part, resulting in estimation of a random intercept with respect to the cluster indicator. By default, if only a single variable name is supplied for the random-effects part, it is assumed a random intercept should be estimated with respect to that variable.

Specifying more complex random-effects structures

We can also specify more complex random effects structures. Specification of the random effects is performed as is customary with package **lme4**. Thus, if we would have wanted to estimate a random slope of gender, which may be correlated with the random intercept, we could have specified the following model formula:

```
outcome ~ 1 | (1 + gender | cluster_id) | gender + ethnicity +
  autism + conduct + emotional + autism + conduct + eating + self_harm +
  hyperactivity + SEN + other_problems + infrequent_characteristics +
  problem_duration + distress + homelife + friends + classroom + leisure
```

If we would have wanted to estimate a random slope of gender, which should not be correlated with the random intercept, we could have specified the following model formula:

```
outcome ~ 1 | (1 | cluster_id) + (0 + gender | cluster_id) | age + gender +
  ethnicity + autism + conduct + emotional + autism + conduct + eating +
  self_harm + hyperactivity + SEN + other_problems +
```

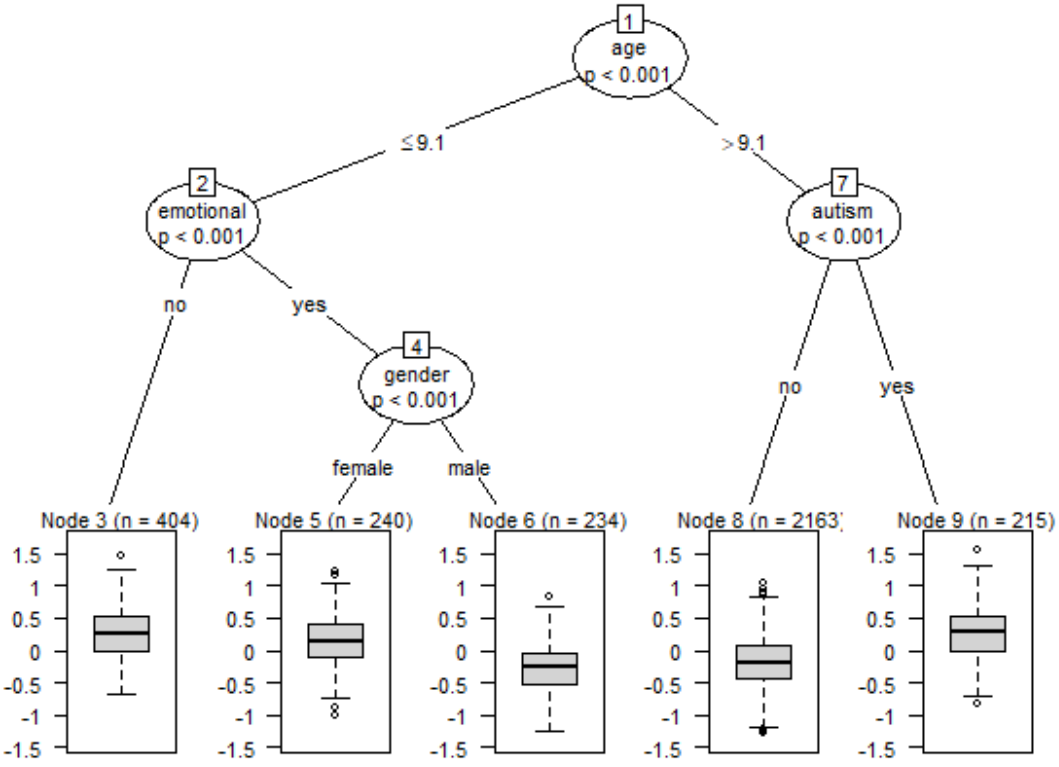
```
  infrequent_characteristics + problem_duration + distress + homelife +
  friends + classroom + leisure
```

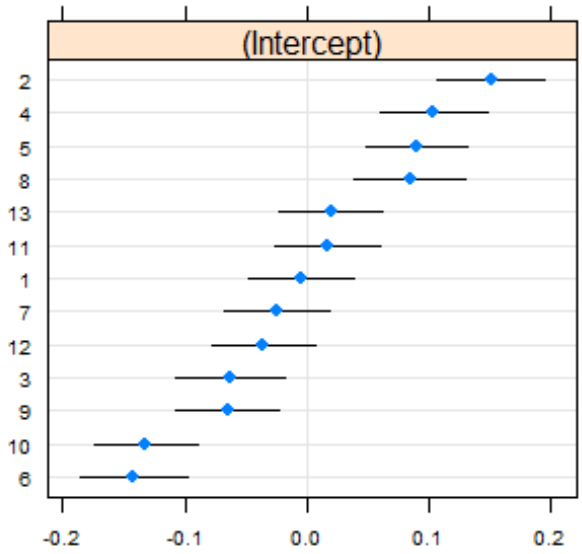
Note that the round brackets are needed here to protect the vertical bars in the formulation of the random effects. The above two formulas are just for illustration, so we will not fit it to the data. Note also that, depending on the context and the variables in the model, it may or may not be sensible to specify a random slope with respect to a predictor variable, which is also a potential partitioning variable. With the above formulas, we specified a model where the average treatment outcome (the intercept) as well as the effect of gender can vary over service providers. We also specified gender as a potential partitioning variable, so it can additionally be used to define subgroups.

Plotting and inspecting the fitted model

Using the `plot` method, we can plot the tree and random effects:

```
plot(lmm_tree)
```





The inner nodes of the tree reveal the partitioning variable selected for splitting, with the p value of the variable's parameter stability test, which quantifies the association between partitioning and outcome variables. The labels of the edges below the node reveal the splitting values.

The terminal node contain boxplots of the observed values of the outcome variable in that node. Note that part of the observed variation within terminal nodes has been explained by the random intercept term in the model. The tree indicates that, on average, respondents aged ≤ 9.1 with emotional problems and male gender, as well as respondents aged > 9.1 without autism, have lower values of the outcome variable (i.e., better outcomes) than the other subgroups.

In every inner node of the plotted tree, the splitting variable and corresponding p-value from the parameter stability test is reported. To control for multiple testing, the p-values are Bonferroni corrected, by default. The significance level α equals .05 by default, but a different value, say for example .01, can be specified by including `alpha = .01` in the call to function `lmertree()`.

By default, the predicted random effects are also plotted. They reveal a rather symmetric distribution of the random intercepts, Also, on average, respondents from cluster 2 have somewhat higher values of the outcome variable (indicating poorer outcomes) and respondents from cluster 6 have somewhat lower values for the outcome variables (indicating better outcomes).

To obtain numerical results, we can employ the `print` method:

```
print(lmm_tree)
```

```
## Linear mixed model tree
##
## Model formula:
## outcome ~ 1 | age + gender + ethnicity + autism + conduct + emotional +
##     autism + conduct + eating + self_harm + hyperactivity + SEN +
##     other_problems + infrequent_characteristics + problem_duration +
##     distress + homelife + friends + classroom + leisure
##
## Fitted party:
## [1] root
## |   [2] age <= 9.1
## |   |   [3] emotional in no: n = 404
## |   |       (Intercept)
## |   |         0.2564449
## |   |   [4] emotional in yes
## |   |   |   [5] gender in female: n = 240
## |   |   |       (Intercept)
## |   |   |         0.1365093
## |   |   |   [6] gender in male: n = 234
## |   |   |       (Intercept)
## |   |   |        -0.2551774
## |   [7] age > 9.1
## |   |   [8] autism in no: n = 2163
## |   |       (Intercept)
## |   |        -0.1840601
## |   |   [9] autism in yes: n = 215
## |   |       (Intercept)
## |   |         0.2693084
##
## Number of inner nodes:    4
## Number of terminal nodes: 5
## Number of parameters per node: 1
## Objective function (residual sum of squares): 421.0794
##
## Random effects:
## $cluster_id
##     (Intercept)
## 1  -0.004791212
## 10 -0.132258830
## 11  0.017028204
## 12 -0.035721453
## 13  0.019848115
## 2   0.151193027
## 3  -0.062784981
## 4   0.103771255
## 5   0.089618840
## 6  -0.141694526
## 7  -0.024497751
## 8   0.084793316
## 9  -0.064504004
```

```
##
## with conditional variances for "cluster_id"
```

The printed tree shows the node-specific estimated means. These would be the values used for prediction of new observations. The printed random effects show the points estimates plotted above. Alternatively, to obtain separate numerical results for the fixed- and random-effects parts of the fitted model, the `coef` and `fixef` (for the fixed effects) and `ranef` (for the random effects) methods can be employed (results omitted for space considerations):

```
coef(lmm_tree)
fixef(lmm_tree)
ranef(lmm_tree)
```

To obtain the estimated variance of the random effects, we can employ the `varCorr` method:

```
VarCorr(lmm_tree)
```

```
##  Groups     Name        Std.Dev.
##  cluster_id (Intercept) 0.092648
##  Residual               0.360521
```

To compute the intra-class correlation, we can divide the variance of the random intercept by the sum of the residual and random intercept variance:

```
vc <- as.data.frame(VarCorr(lmm_tree))
vc$vcov[1] / sum(vc$vcov)
```

```
## [1] 0.06194886
```

To obtain predicted values, we can employ the `predict` method:

```
predict(lmm_tree, newdata = MHserv_data[1:5, ])
```

```
##           1           2           3           4           5
## -0.24684511  0.11201159 -0.09926682 -0.32575466 -0.20855788
```

When the `newdata` argument is not specified, predictions for the training observations are returned, by default. Random effects can be excluded from the predictions by adding `re.form = NA`. This is useful, for example, when the `newdata` argument is specified, but the new observations are from 'new' clusters, or do not have a cluster indicator, e.g.:

```
predict(lmm_tree, newdata = MHserv_data[1:5, -7], re.form = NA)
```

```
##          1          2          3          4          5
## -0.1840601  0.1365093 -0.1840601 -0.1840601 -0.1840601
```

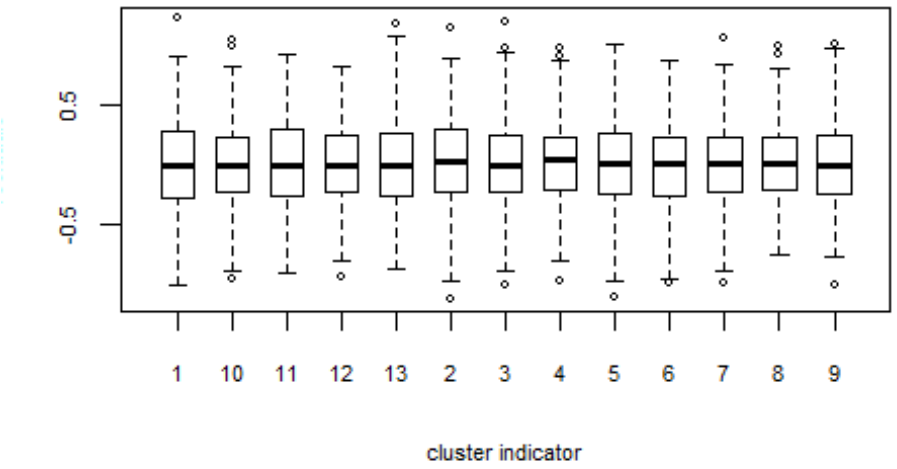
Inspecting residuals

Residuals of the fitted model can be obtained with the `residuals` method. This can be useful for assessing potential misspecification of the model:

```
resids <- residuals(lmm_tree)
preds <- predict(lmm_tree)
```

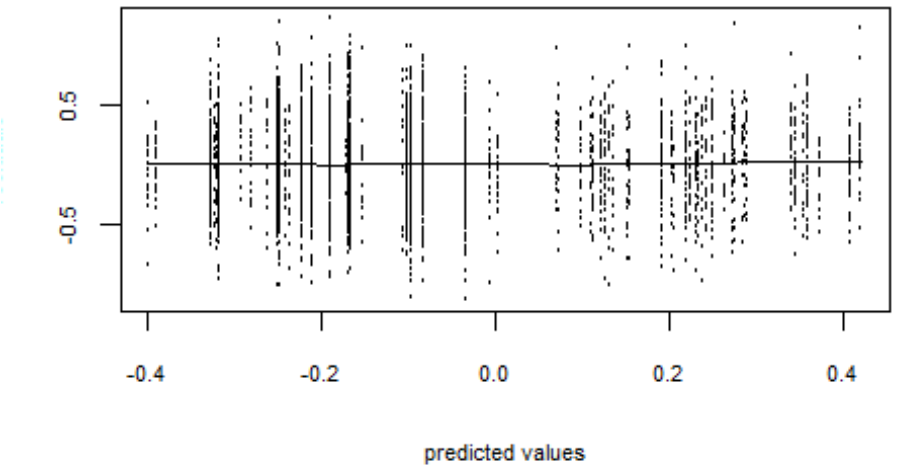
We can plot the residuals against the cluster indicator:

```
plot(factor(MHserv_data$cluster_id), resids, xlab = "cluster indicator",
     ylab = "residuals")
```



The resulting plot indicates similar residual variances across the clusters. We can also plot the residuals against the fitted values to check for possible heteroscedasticity:

```
scatter.smooth(preds, resids, xlab = "predicted values", ylab = "residuals")
```



The resulting plot does not indicate heteroscedasticity (i.e., a pattern of differences in residual variance with respect to the fitted/predicted values). Note that the predicted

values need not be continuously distributed: because of the subgroup-specific predicted values, gaps may occur in the fitted values.

Fitting a tree to binary or count outcomes

For binary and count outcomes, the `glmertree()` function can be used. We create a binary variable from the continuous outcome variable:

```
MHserv_data$outcome_bin <- factor(MHserv_data$outcome > 0)
levels(MHserv_data$outcome_bin) <- c("favorable", "poor")
```

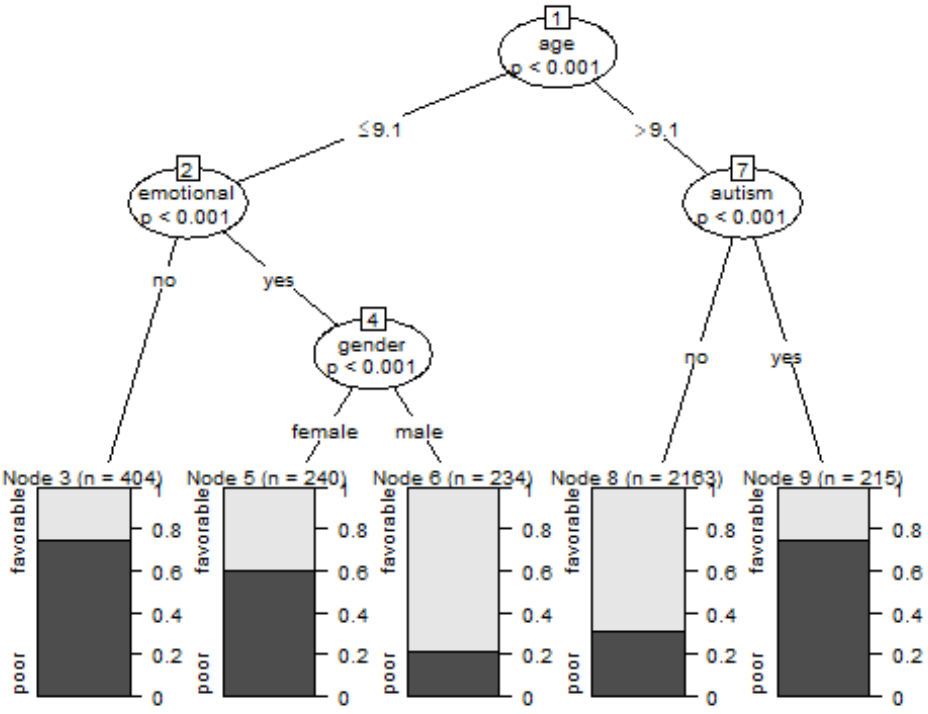
Note that binarizing continuous variables results in a loss of information and is normally not recommended. In this example, we binarized the outcome variable merely to provide an example of predicting a binary outcome.

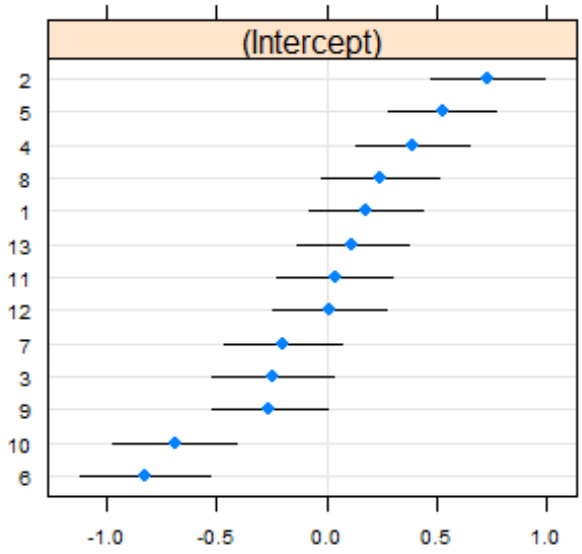
Now we fit a mixed-effects regression tree for a binary outcome using the `glmertree()` function. We need to specify the type of outcome using the `family` argument, which should be set to `binomial` for a binary outcome (for a count response, it should be set to `poisson`):

```
glmm_tree <- glmertree(outcome_bin ~ 1 | cluster_id | age + gender +
                         ethnicity + autism + conduct + emotional +
                         eating + self_harm + hyperactivity + SEN +
                         other_problems + infrequent_characteristics +
                         problem_duration + distress + homelife + friends +
                         classroom + leisure, data = MHserv_data,
                       family = binomial)
```

As above, we can plot the tree using the `plot` method:

```
plot(glmm_tree)
```





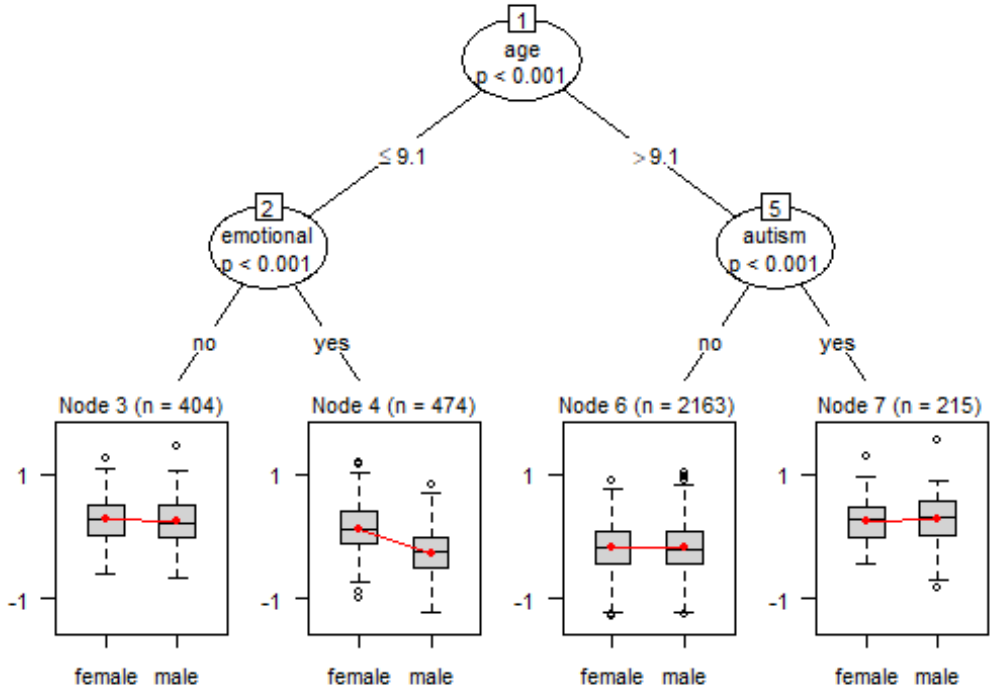
The plotted tree reveals the same structure for the binary outcome as for the continuous outcome. The plot of the random effects also reveals a similar pattern as for the continuous outcome.

Specifying predictor variables of specific interest, or detecting moderators

In the examples above, the partitioning was performed based on an intercept-only model. Additional predictor variables may be specified for the linear model. This may be helpful, for example, if a user is specifically interested in the effects of one (or two) predictor variables, in interaction with potential partitioning variables. For example, we may be specifically interested in the effect of gender on the continuous outcome, and whether the effect of gender differs over subgroups:

```
lmm_tree2 <- lmertree(outcome ~ gender | cluster_id | age +
                        ethnicity + autism + conduct + emotional +
                        eating + self_harm + hyperactivity + SEN +
                        other_problems + infrequent_characteristics +
                        problem_duration + distress + homelife + friends +
                        classroom + leisure, data = MHserv_data)

plot(lmm_tree2, which = "tree")
```



The resulting tree reveals the same subgroups and patterns as the trees we fitted above: Higher (i.e., poorer) outcomes for those with lower age and not presenting with emotional problems; lower (i.e., better) outcomes for those with higher age and not presenting with autism. The strongest effect of gender is observed in the lower age group presenting with emotional problems, where girls show poorer (higher) outcomes than boys. This is similar to the results above; instead of accounting for the effect of gender with an additional split of node four, the effect of gender is accounted for within terminal nodes. In addition, the tree

indicates the effect of gender is much stronger in node 4 than in the other termninal nodes. Thus, the effect of gender is moderated by age and the presence of emotional problems.

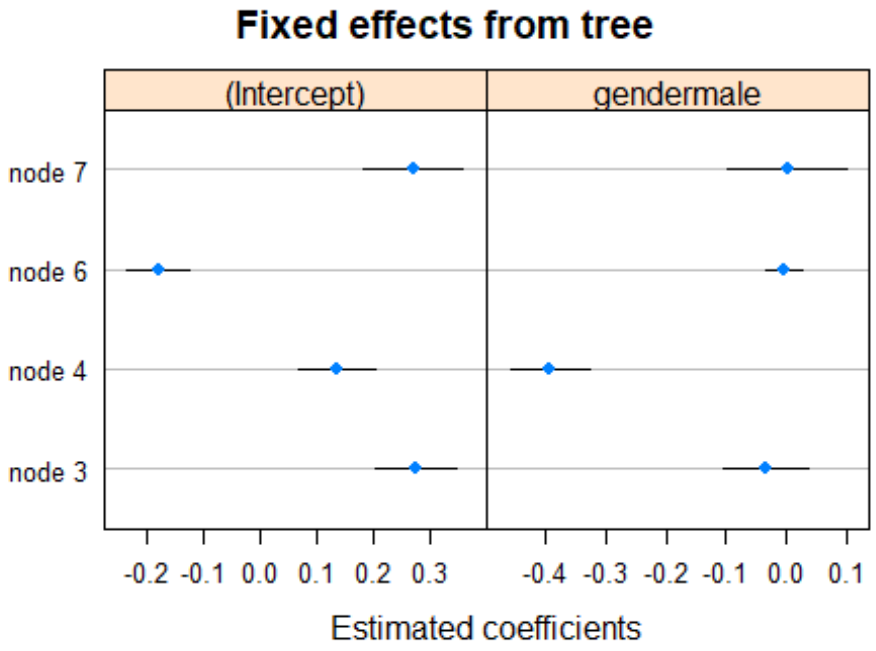
This can also be observed from the estimated fixed-effects coefficients:

```
coef(lmm_tree2)
```

```
##   (Intercept)    gendermale
## 3   0.2746841 -0.0350535436
## 4   0.1364912 -0.3916881607
## 6  -0.1808737 -0.0060421447
## 7   0.2690539  0.0004297227
```

The effect of gender is close to zero in all terminal nodes, but is relatively strong in node 4. We can also plot the estimated effects with error bars, in order to evaluate how strongly the estimated effects differ from zero (note however that the error bars do not account for the searching of the tree structure, and likely underestimate true variability):

```
plot(lmm_tree2, which = "tree.coef")
```



The plot indicated that the effect of gender is not significantly different from zero in nodes 3, 6 and 7. The estimated effect of gender in node 4 appears significant and negative.