

Workshop Speech Prosody. Part I: Introduction to Mixed-Effects Trees

Marjolein Fokkema

Preparations

Make sure you have a recent version of **R** installed (version 4.4 or higher). The code examples *may* also run with earlier **R** versions, but this is not guaranteed. You can check the current version by typing `R.Version()$version.string`.

The latest **R** version can be downloaded from <https://cran.r-project.org/mirrors.html>. First pick a mirror from the country you are currently in. After selecting a mirror, you'll find links to download and install **R** for different operating systems.

You are strongly advised to use a code editor such as R Studio or Visual Studio Code.

Several packages need to be installed and loaded. All but one can be downloaded from the default CRAN servers by clicking “Packages”, “Install” in R Studio and typing the packages names, or by typing the following in the command line:

```
install.packages("partykit", "glmertree", "mgcv", "merDeriv",  
                 "splines", "lme4", "Formula", "devtools")
```

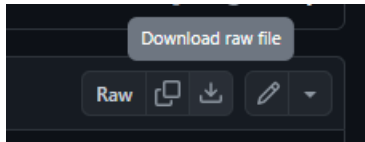
Finally, one package that is needed for the workshop is not yet available from the default CRAN servers (but will be in the future). It is currently available from GitHub (<https://github.com/marjoleinF/gamtree>) and can be installed as follows:

```
library("devtools")  
install_github("r-forge/partykit/pkg/glmertree")  
install_github("marjoleinF/gamtree")
```

The packages can then be loaded as follows:

```
library("partykit")  
library("glmertree")  
library("mgcv")  
library("merDeriv")  
library("splines")  
library("lme4")  
library("Formula")  
library("gamtree")
```

All materials for the workshop (scripts, data, slides) can be found on <https://github.com/marjoleinF/Speech-prosody-workshop-trees/>. All datasets are available as .rda files, and can be downloaded by clicking on the name of the dataset, and downloading the ‘raw file’ by clicking on this button in the top-right of the screen:



Example dataset 1: Filled pauses

For the first examples we repeat analyses from Gardner et al. (2021), who analysed a subset of the Switchboard Corpus of American English (> 6,000 speaking turns from 285 transcripts of 34 speakers). They show “that the presence of variable contexts does not positively correlate with [...] production difficulty, namely filled pauses (um and uh) and unfilled pauses (speech planning time).”

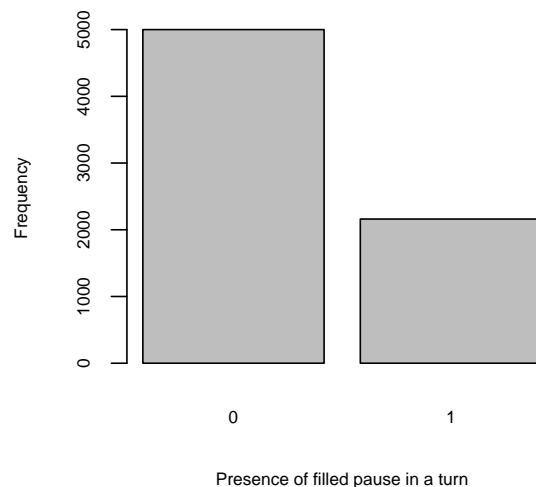
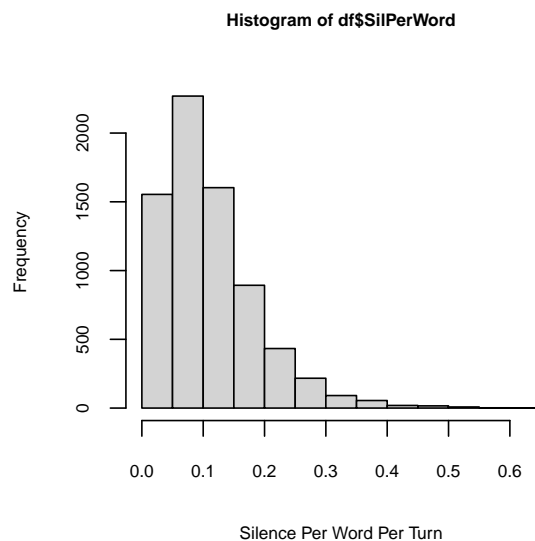
Models for a binary outcome (presence versus absence of filled pauses), as well as a continuous outcome (length of silence per word) were fitted. In both analyses several control predictors were included, a-priori known to influence disfluency: overall turn length, speech rate and content complexity. The main interest of the analyses was in the effect of

We can load and inspect the data as follows:

```
load("gardner.rda")
summary(df) ## results omitted for space considerations
sapply(df, class) ## results omitted for space considerations
```

The continuous and binary outcomes are as follows:

```
par(mfrow = c(1, 2))
hist(df$SilPerWord, xlab = "Silence Per Word Per Turn", cex = .7, cex.lab = .7,
     cex.axis = .7, cex.main = .7)
plot(df$NFP.bi, xlab = "Presence of filled pause in a turn", ylab = "Frequency",
     cex = .7, cex.lab = .7, cex.axis = .7, cex.main = .7)
```



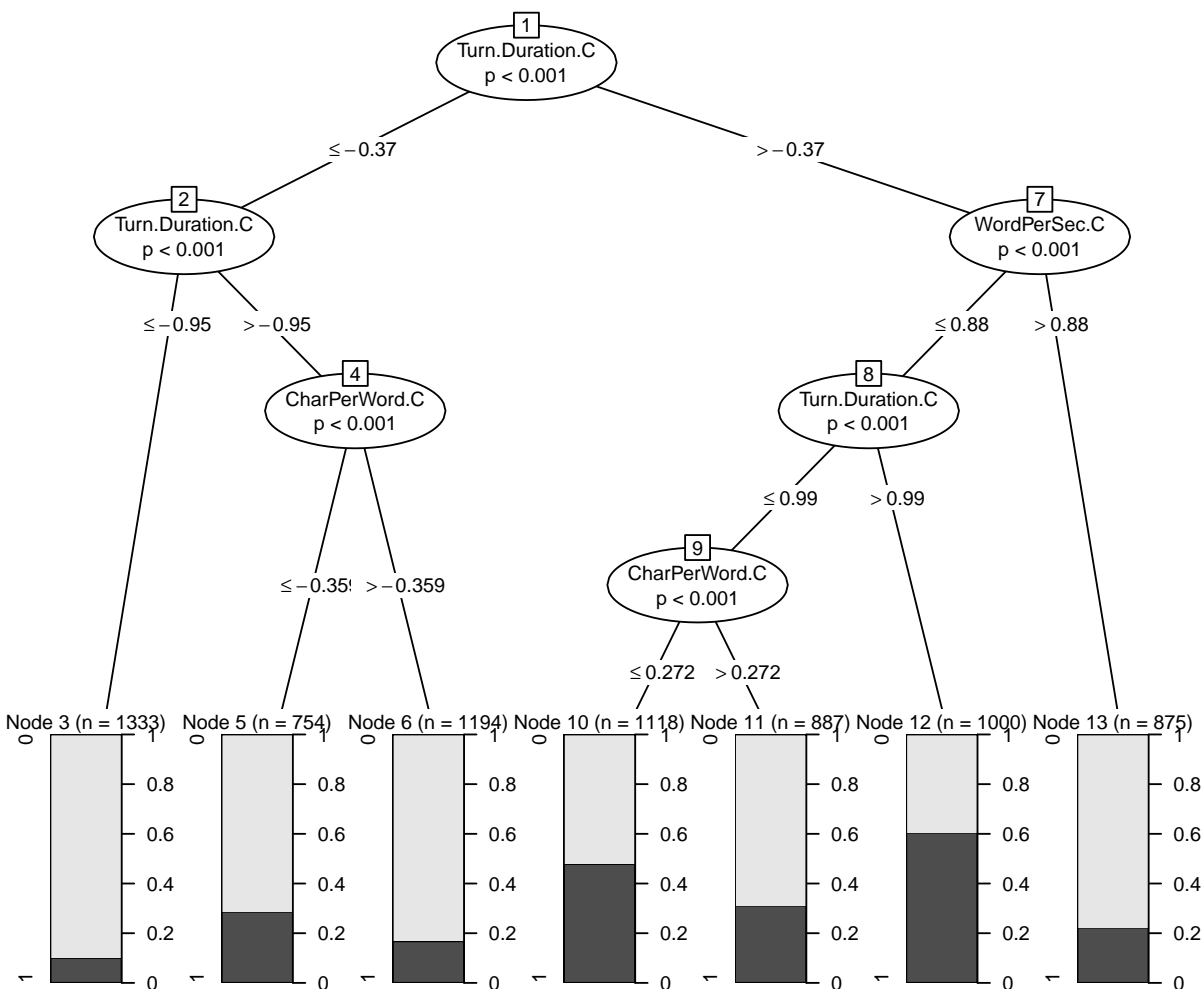
Fitting a GLMM tree to a binary outcome

```
## Tree fitting will take some time, verbose = TRUE allows to monitor progress
gt1 <- glmertree(NFP.bi ~ 1 | (1 | Speaker_Number) | NVarbs.bi + Turn.Duration.C +
  CharPerWord.C + WordPerSec.C, minsize = 750,
  data = df, verbose = TRUE, family = "binomial")
```

```
## 'log Lik.' -3639.952 (df=8)
## 'log Lik.' -3636.764 (df=8)
## 'log Lik.' -3636.764 (df=8)
```

We can plot the tree as follows:

```
plot(gt1, which = "tree", gp = gpar(cex=.7))
```



With `gp = gpar(cex=.7)` we specified the size of plotting labels and symbols, in order to make the tree easier to read (see `?gpar` for more info). The plot shows observed proportions of filled pauses in the terminal nodes. We see that only the control variables have been picked up to predict presence of a filled pause.

If we want to obtain the node-specific fixed-effects estimates from each terminal node, we can type:

```
fixef(gt1)
```

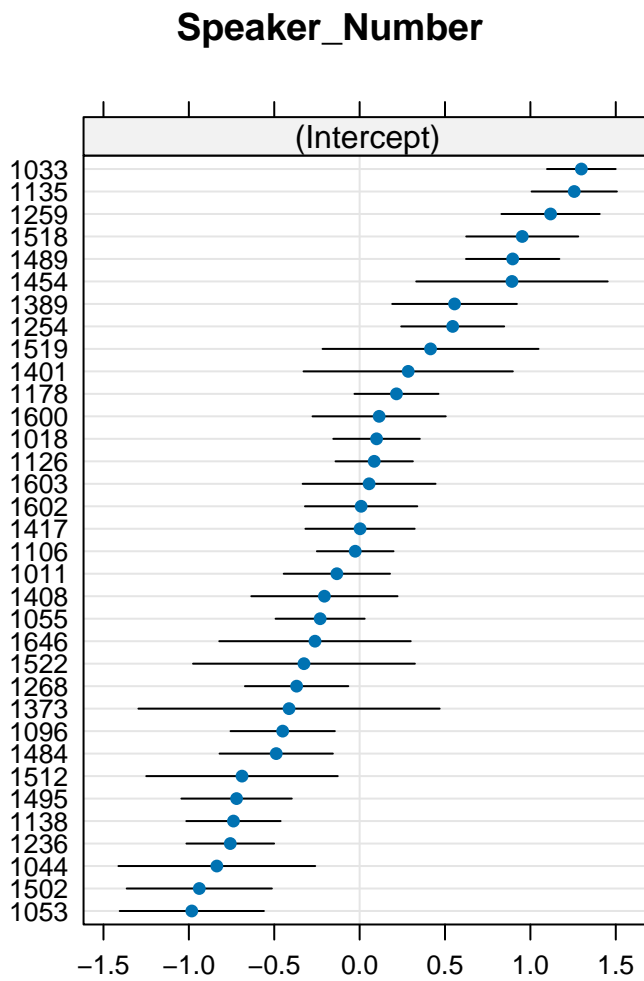
```
##      (Intercept)
## 3      -2.4054977
## 5      -1.1114517
## 6      -1.7355811
## 10     -0.1797613
## 11     -0.8547658
## 12      0.3968831
## 13     -1.1937829
```

The estimated coefficients represent the log-odds of a filled pause.

We can also plot and print the random effects:

```
plot(gt1, which = "ranef")
```

```
## $Speaker_Number
```



```
VarCorr(gt1)
```

```
## Groups          Name          Std.Dev.  
## Speaker_Number (Intercept) 0.6716
```

```
coef(gt1)
```

```
## (Intercept)  
## 3 -2.4054977  
## 5 -1.1114517  
## 6 -1.7355811  
## 10 -0.1797613  
## 11 -0.8547658  
## 12 0.3968831  
## 13 -1.1937829
```

There seems to be considerable variation between different speakers, which is accounted for by the random intercept term.

Different models, different specifications

Continuous outcome

For modeling the continuous outcome (silence per word, log transformed by the original authors because of skewness), we can use function `lmertree` instead of `glmertree`:

```
lt <- lmertree(SilPerWord.log ~ 1 | (1|Audiofile) | NVarbs.bi + Turn.Duration.C +  
              CharPerWord.C + WordPerSec.C, minsize = 750, data = df,  
              verbose = TRUE)
```

```
## 'log Lik.' -6141.884 (df=9)  
## 'log Lik.' -6152.339 (df=8)  
## 'log Lik.' -6152.339 (df=8)
```

The resulting model can be inspected by using `print`, `plot` etc., as before with `glmertree`.

More partitioning variables

We could also specify a much larger number of potential partitioning variables. Because use of the dot (.) should be avoided to specify ‘all remaining variables’, we use a trick to specify a large number of predictors with similar names:

```
## Fit a tree with many more predictors (binary ones) added  
V.bi <- c(paste0("V0", 1:9, ".bi"), paste0("V", 10:20, ".bi"))  
V.bi
```

```
## [1] "V01.bi" "V02.bi" "V03.bi" "V04.bi" "V05.bi" "V06.bi" "V07.bi" "V08.bi"  
## [9] "V09.bi" "V10.bi" "V11.bi" "V12.bi" "V13.bi" "V14.bi" "V15.bi" "V16.bi"  
## [17] "V17.bi" "V18.bi" "V19.bi" "V20.bi"
```

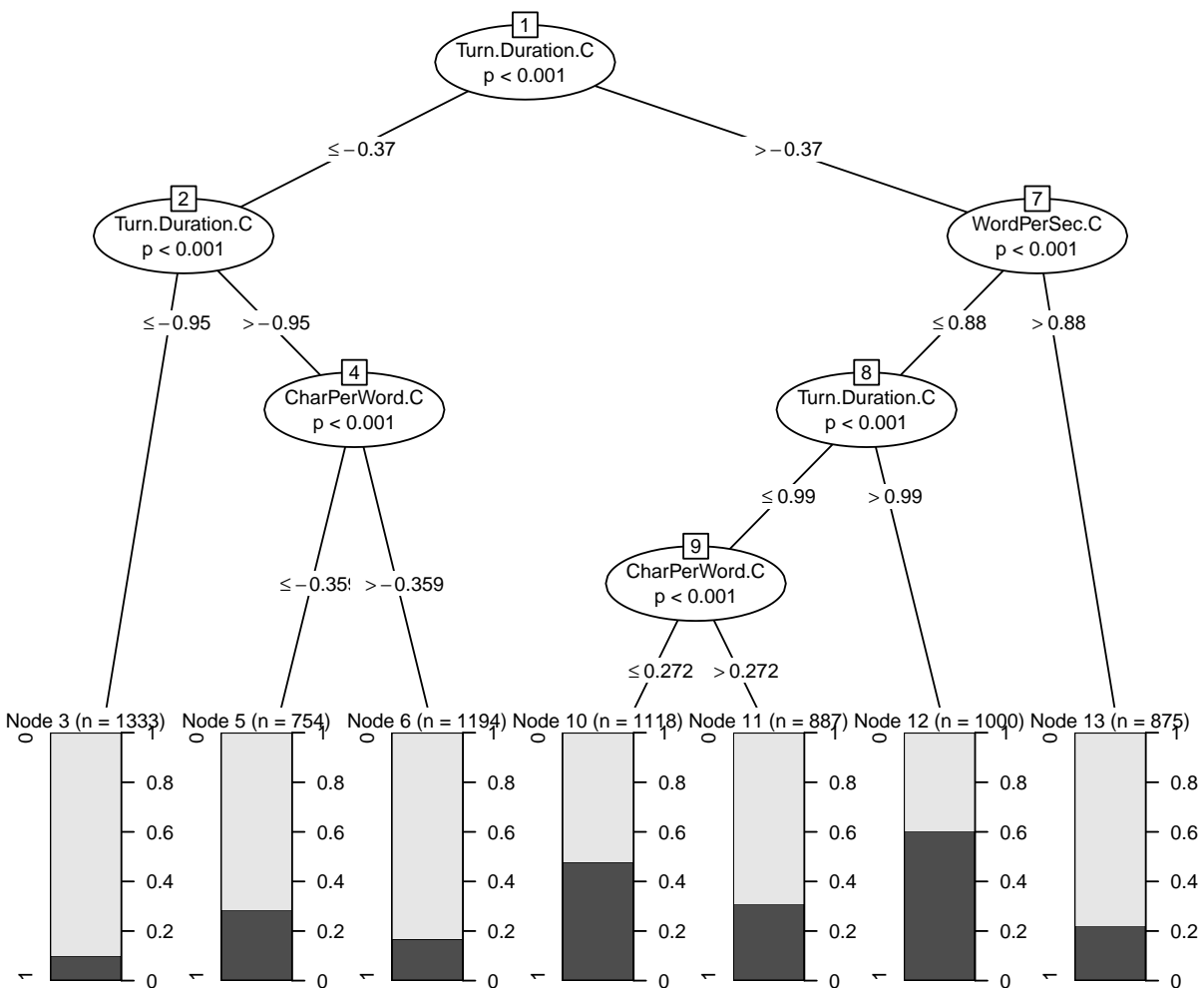
```
ff <- as.Formula(paste("NFP.bi ~ 1 | (1 | Speaker_Number) | NVarbs.bi +
  Turn.Duration.C + CharPerWord.C + WordPerSec.C + ",
  paste0(V.bi, collapse = "+")))
## The error above can safely be ignored
ff
```

```
## NFP.bi ~ 1 | (1 | Speaker_Number) | NVarbs.bi + Turn.Duration.C +
## CharPerWord.C + WordPerSec.C + V01.bi + V02.bi + V03.bi +
## V04.bi + V05.bi + V06.bi + V07.bi + V08.bi + V09.bi + V10.bi +
## V11.bi + V12.bi + V13.bi + V14.bi + V15.bi + V16.bi + V17.bi +
## V18.bi + V19.bi + V20.bi
```

```
gt2 <- glmertree(ff, minsize = 750, data = df, verbose = TRUE, family = "binomial")
```

```
## 'log Lik.' -3639.952 (df=8)
## 'log Lik.' -3636.764 (df=8)
## 'log Lik.' -3636.764 (df=8)
```

```
plot(gt2, which = "tree", gp = gpar(cex = .7))
```



Adding more predictors in this case did not make a difference to the final tree. Note that a Bonferroni correction is applied by default when adding more predictors, so power to detect effects may be reduced.

Forcing a predictor to appear in the node-specific model

If we are specifically interested in the effects of `NVarbs.bi`, we can force it into the model by specifying it as a node-specific predictor:

```
gt3 <- glmertree(NFP.bi ~ NVarbs.bi | (1 | Speaker_Number) | Turn.Duration.C +  
  CharPerWord.C + WordPerSec.C, minsize = 750,  
  data = df, verbose = TRUE, family = "binomial")
```

```
## 'log Lik.' -3628.128 (df=15)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  
## Model failed to converge with max|grad| = 0.00505525 (tol = 0.002, component 1)
```

```
## 'log Lik.' -3626.43 (df=15)
```

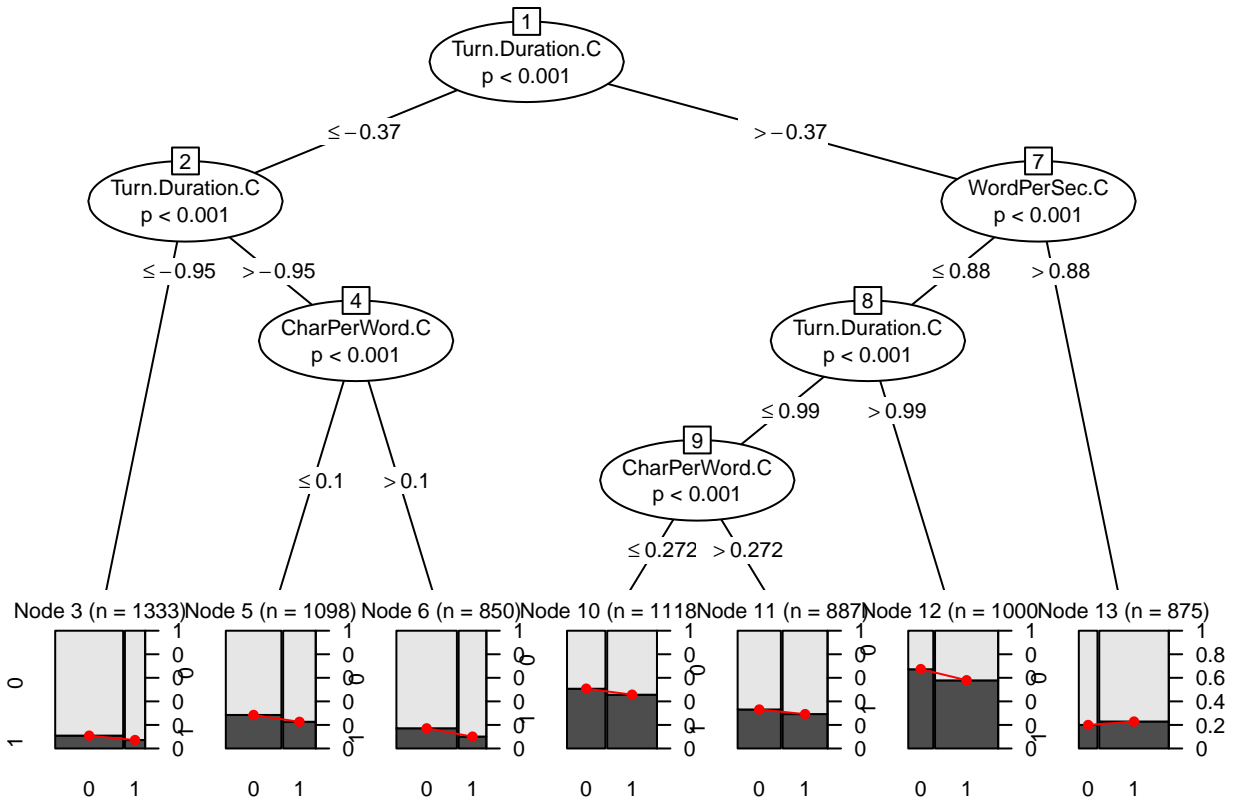
```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :  
## Model failed to converge with max|grad| = 0.00505525 (tol = 0.002, component 1)
```

```
## 'log Lik.' -3626.43 (df=15)
```

We obtained warnings that the model did not converge. Inspecting the tree, this is likely due to `NVarbs.bi` having no effect. In this case, by forcing it into the node-specific model, we have made the model more (perhaps too) complex, without explaining more variance:

```
plot(gt3, which = "tree", gp = gpar(cex = 0.7))
```

```
## Loading required namespace: vcd
```



```
fixef(gt3)
```

```
##      (Intercept) NVarbs.bi1
## 3  -2.32101229 -0.4490296
## 5  -1.10801228 -0.3272581
## 6  -1.68006244 -0.6489687
## 10 -0.08385757 -0.1707005
## 11 -0.79310376 -0.1205371
## 12  0.59991147 -0.2759842
## 13 -1.29173764  0.1241531
```

The tree structure and probabilities in the terminal nodes are very similar as in the previous tree, suggesting no interactions between `NVarbs.bi` and the other predictors.

Specifying more complex random effects

Perhaps we want to have a more complex random-effects specification, e.g., also estimate a random intercept with respect to the audio file that the turns were taken from. In principle, `(g)lmertree` can incorporate any random-effects specification that is supported by `lme4` (e.g., multiple random effects, random slopes, crossed random effects), by including it between the first and second vertical bars:

```
gt4 <- glmertree(NFP.bi ~ 1 | (1|Speaker_Number) + (1|Audiofile) |
  NVarbs.bi + Turn.Duration.C + CharPerWord.C + WordPerSec.C,
  minsize = 750, data = df, verbose = TRUE, family = "binomial")
```



```
## 'log Lik.' -3628.034 (df=9)
## 'log Lik.' -3625.387 (df=9)
## 'log Lik.' -3625.387 (df=9)
```

```
VarCorr(gt4)
```

```
## Groups          Name          Std.Dev.
## Audiofile      (Intercept) 0.36854
## Speaker_Number (Intercept) 0.67251
```

The addition of the second random intercept term did not change any splits in the resulting tree.

Correcting for predictors of a-priori known relevance

Gardner et al. mentioned that they included several control predictors in the analyses. However, we may want to avoid these control variables to appear in the tree, as they do in the previous analyses. We may then want to expand the 'global' part of the model, so as to control for the linear effects of predictors:

```
gt5 <- glmertree(NFP.bi ~ 1 | (1|Speaker_Number) + Turn.Duration.C +
  CharPerWord.C + WordPerSec.C | NVarbs.bi,
  minsize = 750, data = df, verbose = TRUE, family = "binomial")
```

```
## 'log Lik.' -3560.26 (df=6)
## 'log Lik.' -3561.284 (df=5)
## 'log Lik.' -3561.284 (df=5)
```

```
gt5$tree
```

```
## Generalized linear model tree (family: binomial)
##
## Model formula:
## NFP.bi ~ 1 | NVarbs.bi
##
## Fitted party:
## [1] root: n = 7161
##      (Intercept)
##      -1.100646
##
## Number of inner nodes:    0
## Number of terminal nodes: 1
## Number of parameters per node: 1
## Objective function (negative log-likelihood): 3502.425
```

```
fixef(gt5, which = "global")
```

```
## Turn.Duration.C   CharPerWord.C   WordPerSec.C
##      0.9607440      -0.4689586      -0.6029716
```

No splits have been implemented. This confirms our earlier conclusion, that the indicator for variable contexts does not affect the response, over and above the effects of predictors of a-priori known relevance.

References

- Gardner, M. H., Uffing, E., Van Vaec, N., & Szmrecsanyi, B. (2021). Variation isn't that hard: Morphosyntactic choice does not predict production difficulty. *Plos One*, 16(6), e0252602. <https://doi.org/10.1371/journal.pone.0252602>
- Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3), 651-674. <https://doi.org/10.1198/106186006X133933>
- Zeileis, A., Hothorn, T., & Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2), 492-514. <https://doi.org/10.1198/106186008X319331>
- Fokkema, M., Smits, N., Zeileis, A., Hothorn, T., & Kelderman, H. (2018). Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. *Behavior Research Methods*, 50, 2016-2034. <https://doi.org/10.3758/s13428-017-0971-x>
- Schlosser, L., Hothorn, T., & Zeileis, A. (2019). The power of unbiased recursive partitioning: A unifying view of CTree, MOB, and GUIDE. *arXiv preprint arXiv:1906.10179*. <https://doi.org/10.48550/arXiv.1906.10179>