musicians data set

2023-06-20

Data sets

```
musicians <- read.csv("https://raw.githubusercontent.com/proback/BeyondMLR/master/data/m
usicdata.csv")
musicians <- musicians %>%
  mutate(students = ifelse(audience=="Student(s)",1,0),
         juried = ifelse(audience=="Juried Recital",1,0),
         public = ifelse(audience=="Public Performance",1,0),
         solo = ifelse(perform_type=="Solo",1,0),
         memory1 = ifelse(memory=="Memory",1,0),
         female = ifelse(gender=="Female",1,0),
         vocal = ifelse(instrument=="voice",1,0),
         orch = ifelse(instrument=="orchestral instrument",1,0))

head(musicians)
```

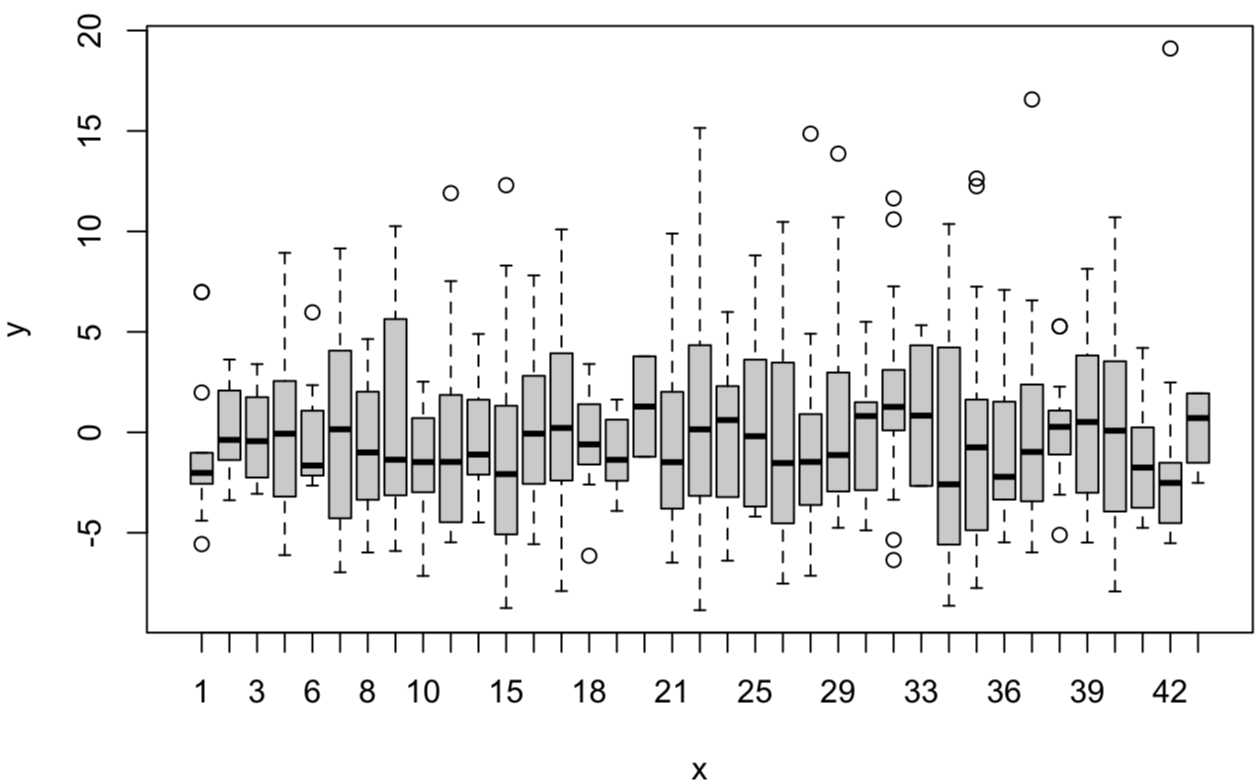
```
##   X id diary previous   perform_type      memory           audience pa na age
## 1 1  1     1        0           Solo Unspecified         Instructor 40 11  18
## 2 2  1     2        1 Large Ensemble      Memory Public Performance 33 19  18
## 3 3  1     3        2 Large Ensemble      Memory Public Performance 49 14  18
## 4 4  1     4        3           Solo      Memory Public Performance 41 19  18
## 5 5  1     5        4           Solo      Memory         Student(s) 31 10  18
## 6 6  1     6        5           Solo      Memory         Student(s) 33 13  18
##   gender instrument years_study mpqab mpqsr mpqpem mpqnem mpqcon students
## 1 Female      voice           3    16     7     52     16     30        0
## 2 Female      voice           3    16     7     52     16     30        0
## 3 Female      voice           3    16     7     52     16     30        0
## 4 Female      voice           3    16     7     52     16     30        0
## 5 Female      voice           3    16     7     52     16     30        1
## 6 Female      voice           3    16     7     52     16     30        1
##   juried public solo memory1 female vocal orch
## 1      0      0    1       0      1     1    0
## 2      0      1    0       1      1     1    0
## 3      0      1    0       1      1     1    0
## 4      0      1    1       1      1     1    0
## 5      0      0    1       1      1     1    0
## 6      0      0    1       1      1     1    0
```

Trees

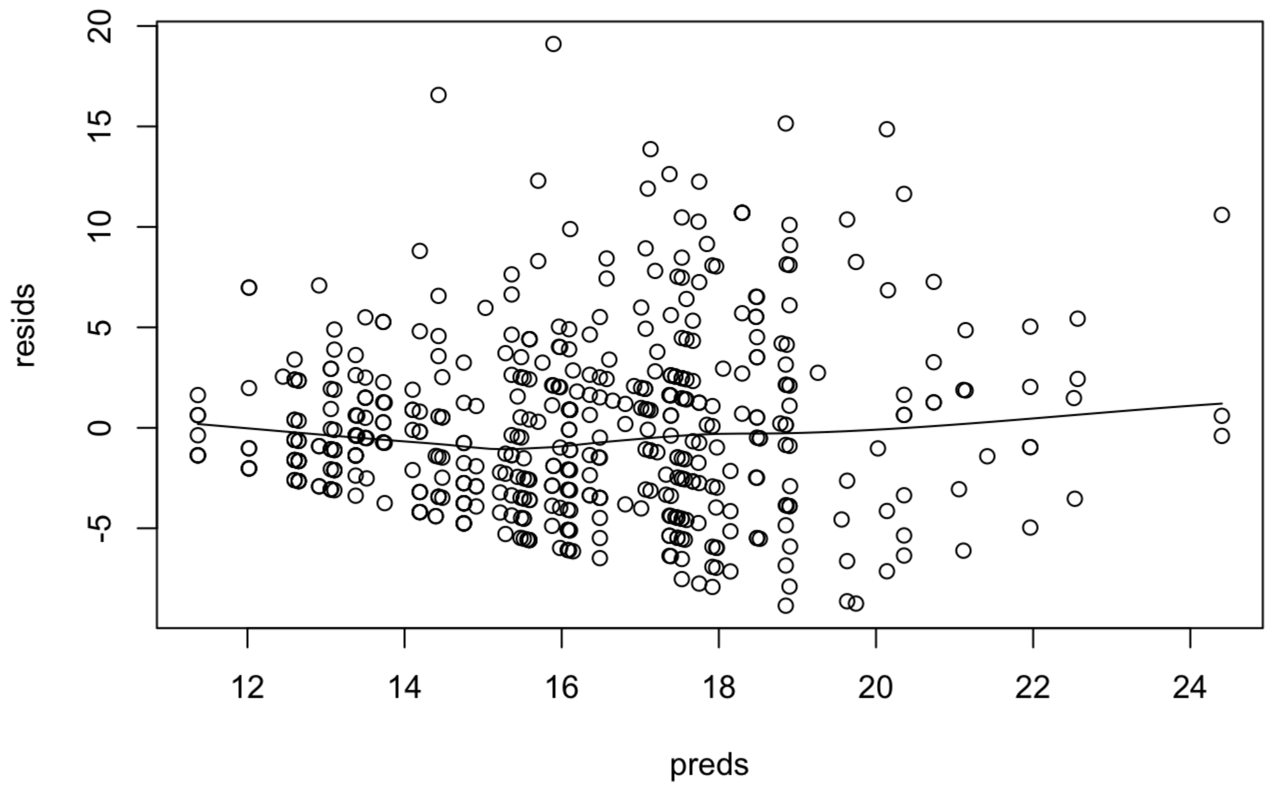
```
musicLM <- lmertree(na ~ 1 | id | previous + students + juried +
    public + solo + mpqpem + mpqab + orch + mpqnem +
    mpqnem:solo, data = musicians, cluster = id)
width(musicLM$tree)
```

```
## [1] 8
```

```
resids <- residuals(musicLM)
preds <- predict(musicLM)
plot(factor(musicians$id), resids)
```




```
scatter.smooth(preds, resids)
```




```
fligner.test(resids ~ musicians$id)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  resids by musicians$id
## Fligner-Killeen:med chi-squared = 61.467, df = 36, p-value = 0.005138
```

```
bartlett.test(resids ~ musicians$id)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  resids by musicians$id
## Bartlett's K-squared = 98.273, df = 36, p-value = 1.099e-07
```

Comparison

We'll be using the first 10 rows of the data set and calculate the RSS of each model.

```
data <- head(musicians$na,10)
```

This is one of our final models in 455:

```
modelB <- lmer(na ~ previous + students + juried +
    public + solo + mpqpem + mpqab + orch + mpqnem +
    mpqnem:solo + (1 | id), data = musicians, REML=TRUE)
```

I purposefully copied these variables and pasted them to be potential partitioning variables.

```
pre <- predict(musicLM, newdata = musicians[1:10,])
sum((pre-data)^2)
```

```
## [1] 154.718
```

```
rmse(data,pre)
```

```
## [1] 3.933421
```

```
pre <- predict(modelB, newdata = musicians[1:10,])
sum((pre-data)^2)
```

```
## [1] 107.4193
```

```
rmse(data,pre)
```

```
## [1] 3.277489
```

This would make me believe that modelB is better.

I decided to use these variables in the model part of the tree instead of using them as partitioning variables just to see what would happen (I'm not expecting much).

```
musicLM2 <- lmertree(na ~ previous + students + juried +
    public + solo + mpqpem + mpqab + orch + mpqnem +
    mpqnem:solo | id, data = musicians, cluster = id)
```

```
## Warning in formula.Formula(ff, lhs = 1L, rhs = c(1L, 3L)): subscript out of
## bounds, not all 'rhs' available
```

```
pre <- predict(musicLM2, newdata = musicians[1:10,])
sum((pre-data)^2)
```

```
## [1] 136.4342
```

```
rmse(data,pre)
```

```
## [1] 3.693701
```

```
musicLM3 <- lmertree(na ~ previous + students + juried +
    public + solo + mpqpem + mpqab + orch + mpqnem +
    mpqnem:solo | id | 1, data = musicians, cluster = id)
```

```
## Error in `[.data.frame`(z, , i) : undefined columns selected
## Error in `[.data.frame`(z, , i) : undefined columns selected
```

```
pre <- predict(musicLM3, newdata = musicians[1:10,])
sum((pre-data)^2)
```

```
## [1] 107.4193
```

```
rmse(data,pre)
```

```
## [1] 3.277489
```

The second error makes perfect sense, but I don't know what the first one means. The first model's RSS and RMSE are both better than my initial tree, though.

I guess at least we can trust that the model-based part of the algorithm really works the same as a regular lmer model.

This is where I wonder if a CART would get an RSS as good as a GLMM tree

Good 'ol training to find the optimal parameter value, followed by a tree:

```
Mus_Train <- musicians[1:300,]
cp_vals = 10^seq(-5, 5, length = 100)
colnames(Mus_Train) <- make.names(colnames(Mus_Train))
control = trainControl("repeatedcv", number = 10, repeats=10)

set.seed(2022)
Mus_Tree <- train(data=Mus_Train, na ~ previous + students + juried +
    public + solo + mpqpem + mpqab + orch + mpqnem +
    mpqnem:solo,  method="rpart", trControl=control,tuneGrid=expand.grid(cp=cp_vals))
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
Mus_Best_Tree <- rpart(na ~ previous + students + juried +
    public + solo + mpqpem + mpqab + orch + mpqnem, data=Mus_Train, cp=Mus_Tree$bestTun
e)
```

```
pre <- predict(Mus_Best_Tree, newdata = musicians[1:10,])
sum((pre-data)^2)
```

```
## [1] 113.1796
```

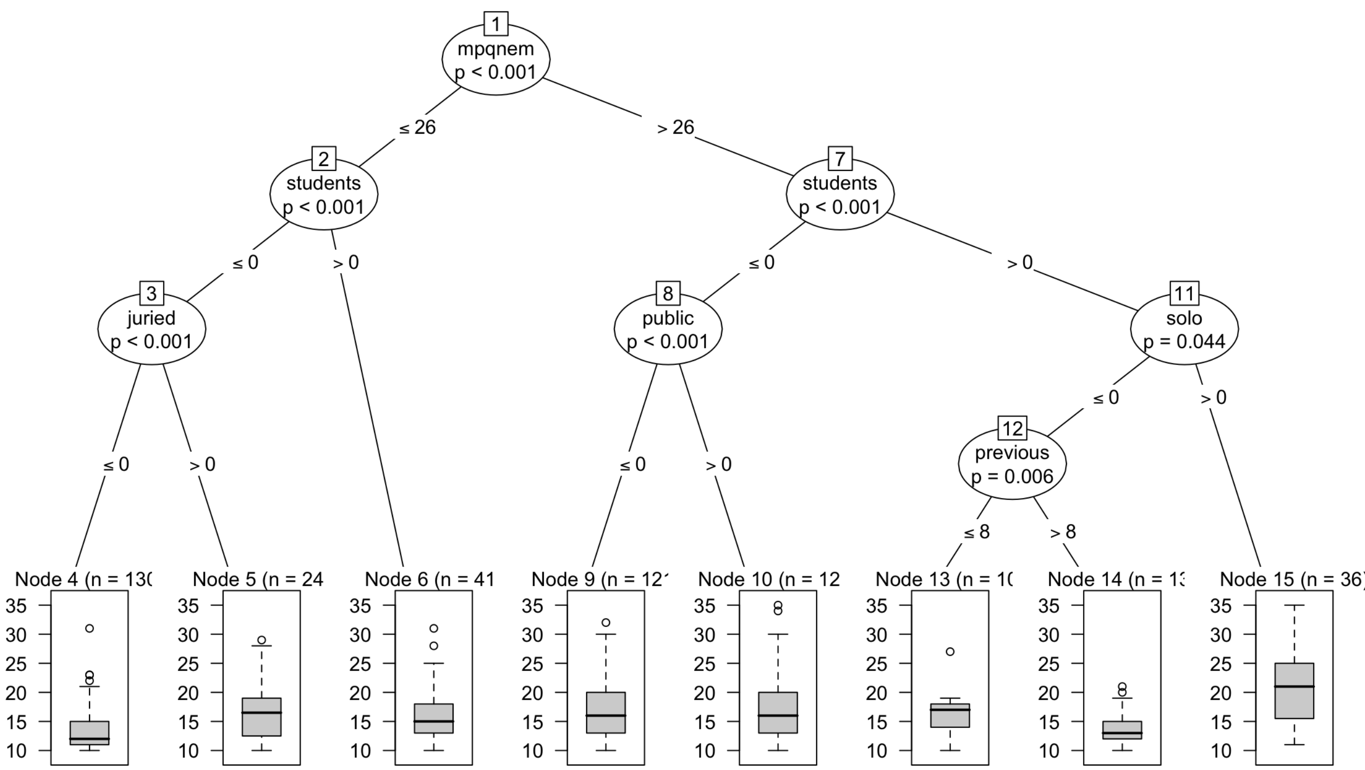
```
rmse(data,pre)
```

```
## [1] 3.364218
```

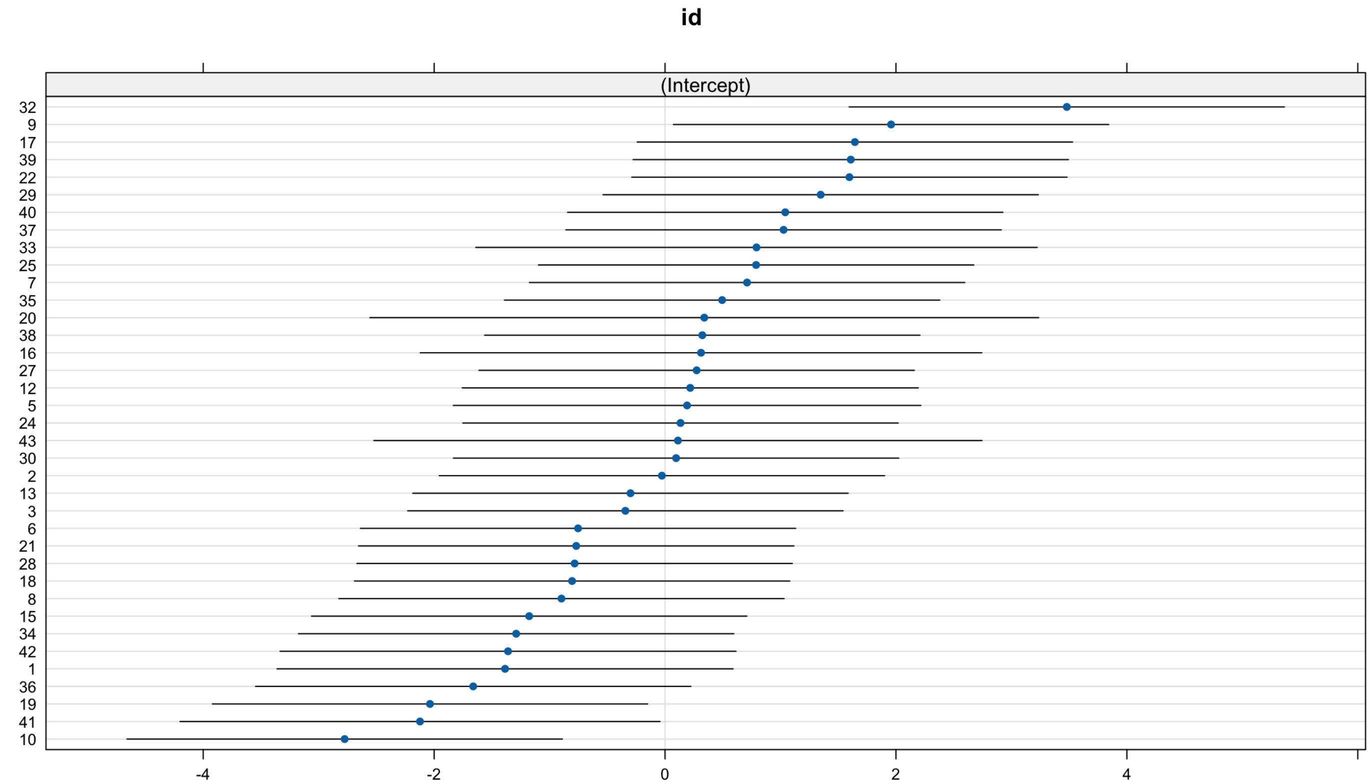
This would make me believe that this algorithm is still really good. But, since I'm not sure if all of my trees are correct (meaning that I don't know if I can take the 107 tree as a win), I don't really have a conclusion after comparing the trees.

Let's look at some trees since we're working with trees.

```
plot(musicLM)
```

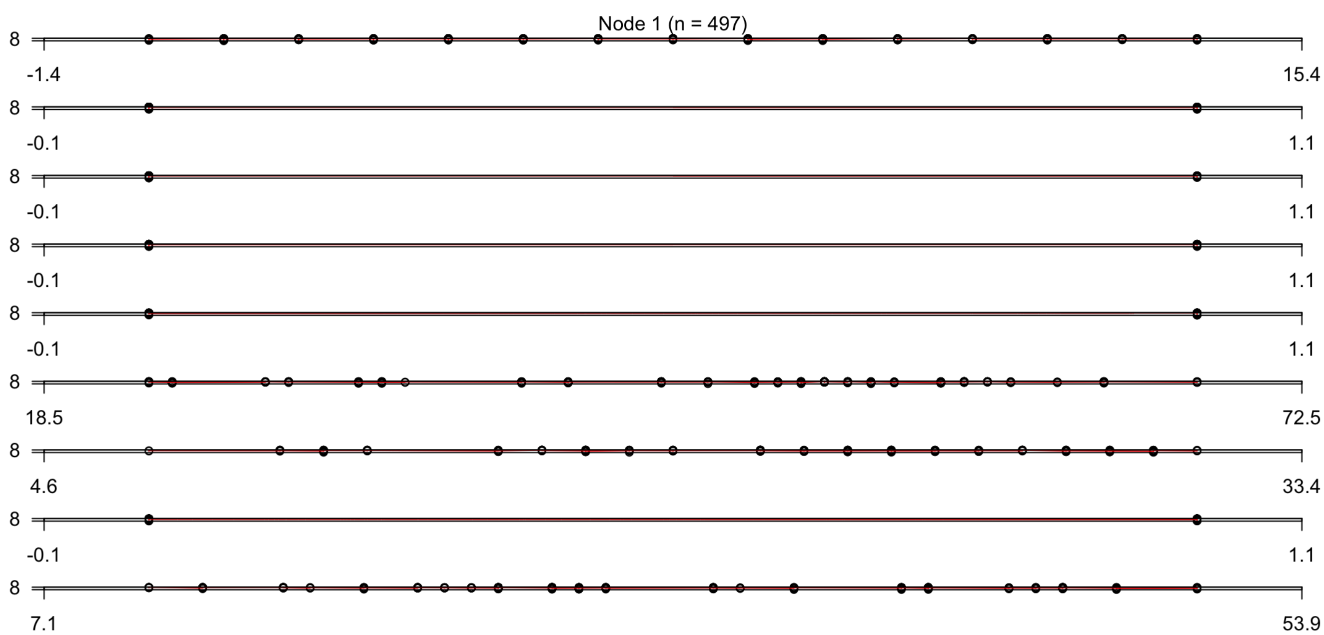



```
## $id
```

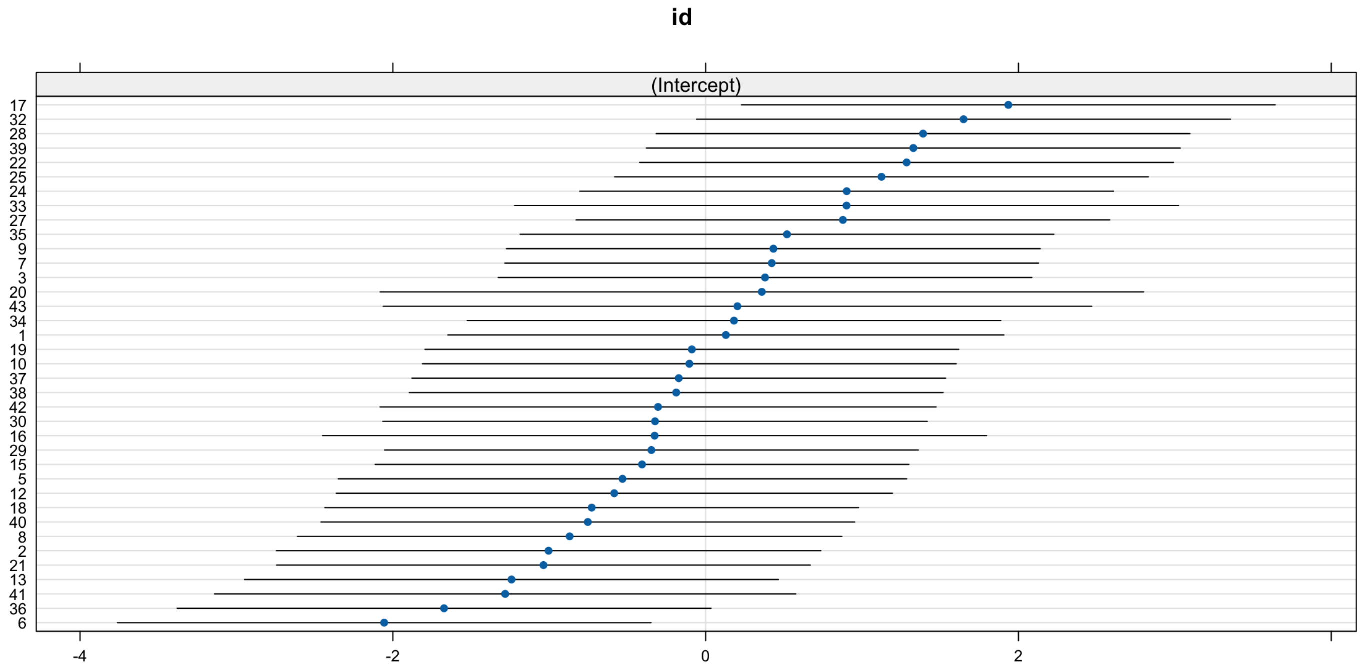


This is the very first tree.

```
plot(musicLM3)
```

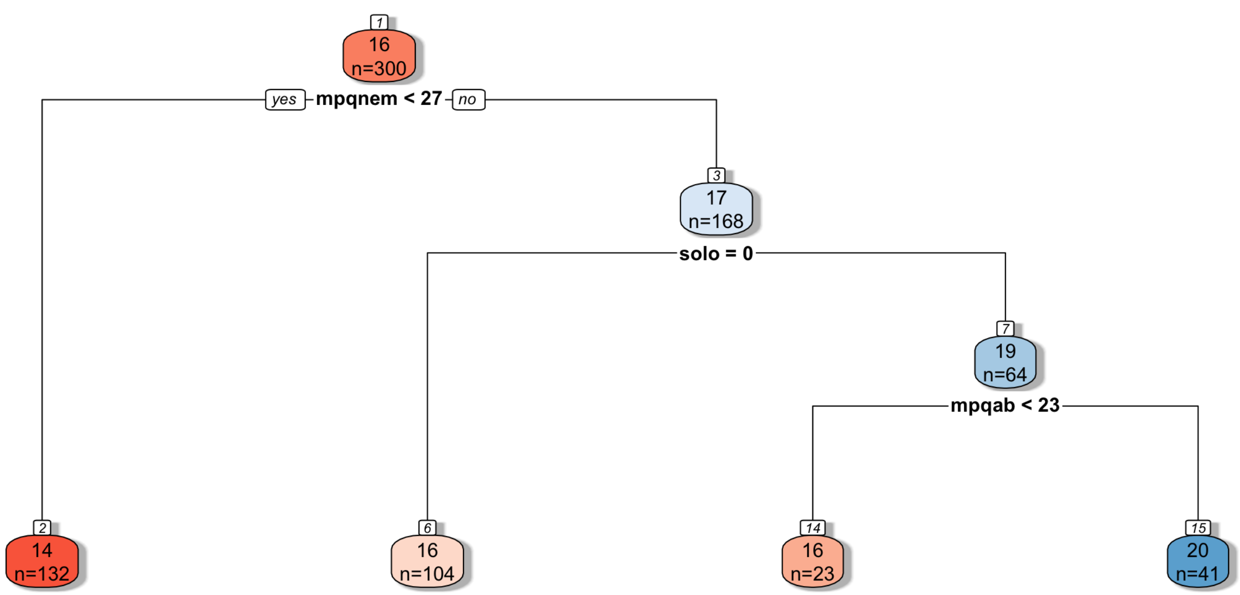



```
## $id
```



This is the GLMM tree that gave us the exact same RSS and RMSE as our linear regression model. I only gave this tree the number 1 as a partitioning variable, so this is expected. I would assume that something like this is not what we would be looking for when we are using a tree algorithm on a data set. But, this is okay for now, since we have a special case on our hands where we already know a really good model.

```
rpart.plot(Mus_Best_Tree, box.palette="RdBu", shadow.col="gray", nn=TRUE, cex=1, extra=
1)
```



This is the CART. If we really, really squint, we can see some similarities with the first GLMM tree.

I realized that the musicLM2 tree is technically not even a tree, because it doesn't have any nodes (partitioning variables).

If we were to use the GLMM tree algorithm to find the relationships between the response and the explanatory variable, I would probably use the first tree where we only fed an intercept as a model to the tree. The results from that tree don't really look great.