depression

2023-06-25

Data

```
data("DepressionDemo", package = "glmertree")
```

TreeS

They have already used the CART algorithm on this data set, but I will just include it here so that we can look at the RSS and RMSE values for fun.

```
De_Train <- DepressionDemo[1:60,]
cp_vals = 10^seq(-5, 5, length = 100)
colnames(De_Train) <- make.names(colnames(De_Train))
control = trainControl("repeatedcv", number = 10, repeats=10)

set.seed(2022)
De_Tree <- train(data=De_Train, depression ~ treatment + age + duration + anxiety,  meth
od="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.
```

```
De_Best_Tree <- rpart(depression ~ treatment + age + duration + anxiety, data=De_Train,
cp=De_Tree$bestTune)
```

```
data <- head(DepressionDemo$depression,10)
pre <- predict(De_Best_Tree, newdata = DepressionDemo[1:10,])
sum((pre-data)^2)
```

```
## [1] 46.2016
```

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

```
## [1] 2.149456
```

```
dLM <- lmertree(depression ~ treatment | (1|cluster) |
                    age + duration + anxiety, data = DepressionDemo)
pre <- predict(dLM, newdata = DepressionDemo[1:10,])
sum((pre-data)^2)
```

```
## [1] 38.8395
```

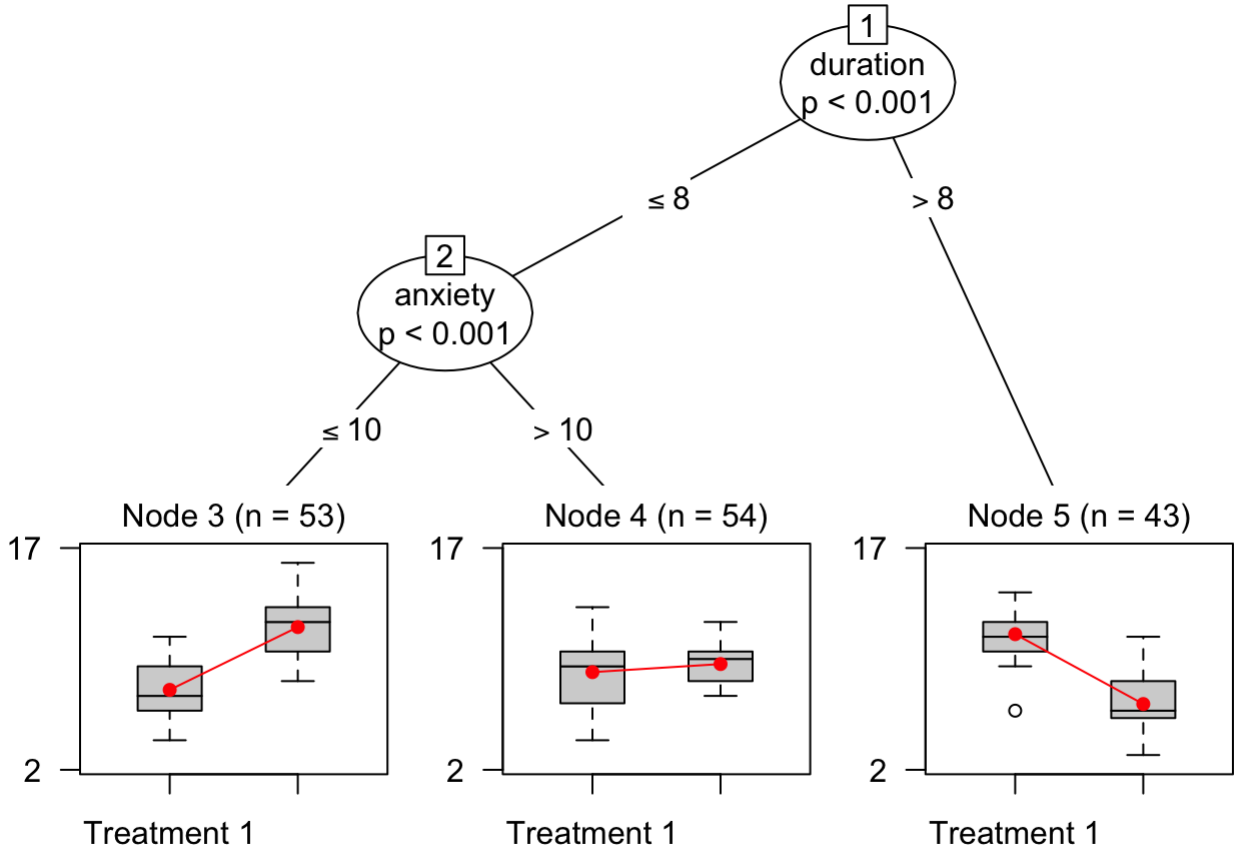
```
rmse(data,pre)
```

```
## [1] 1.970774
```

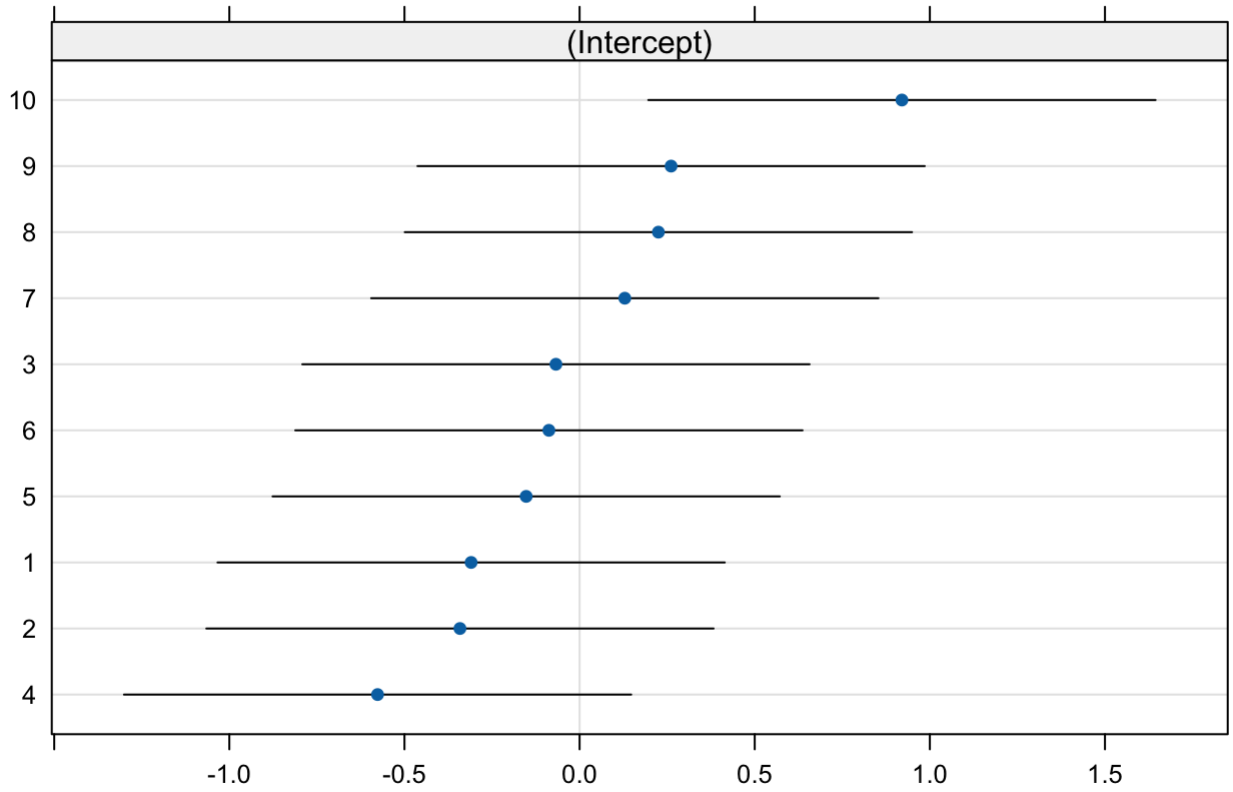
These are definitely not the most exciting numbers since they are just as expected, but it's nice to know that the new and shiny algorithm really is consistently doing better in cases where it's designed to do better.

Again, let's look at trees:

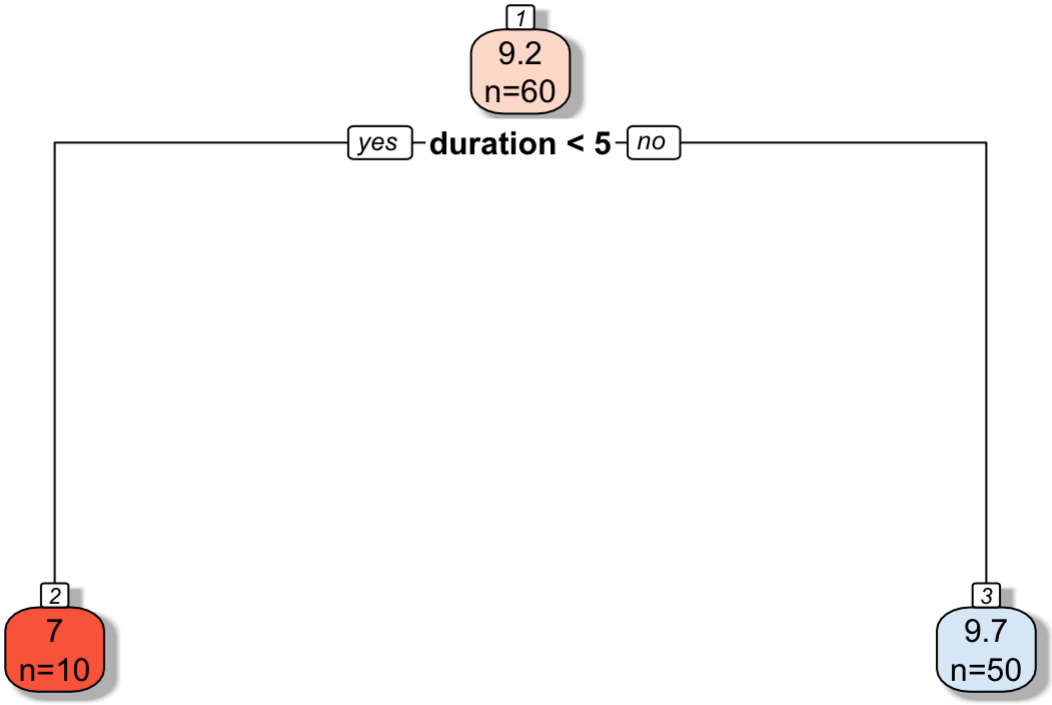
```
plot(dLM)
```




```
## $cluster
```




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



My tree actually didn't even detect the effect of anxiety levels like their tree did.