Introduction to classification and regression trees, random forests and model-based recursive partitioning in R

Day 2: Stability and tree ensembles

Exercise 1: Glaucoma data revisited (1)

a) Use the `stabletree()` function from the `stablelearner` package to assess stability of the tree you fitted yesterday. Do not forget to set the random seed, as the `stabletree()` function uses bootstrap samples to assess stability.

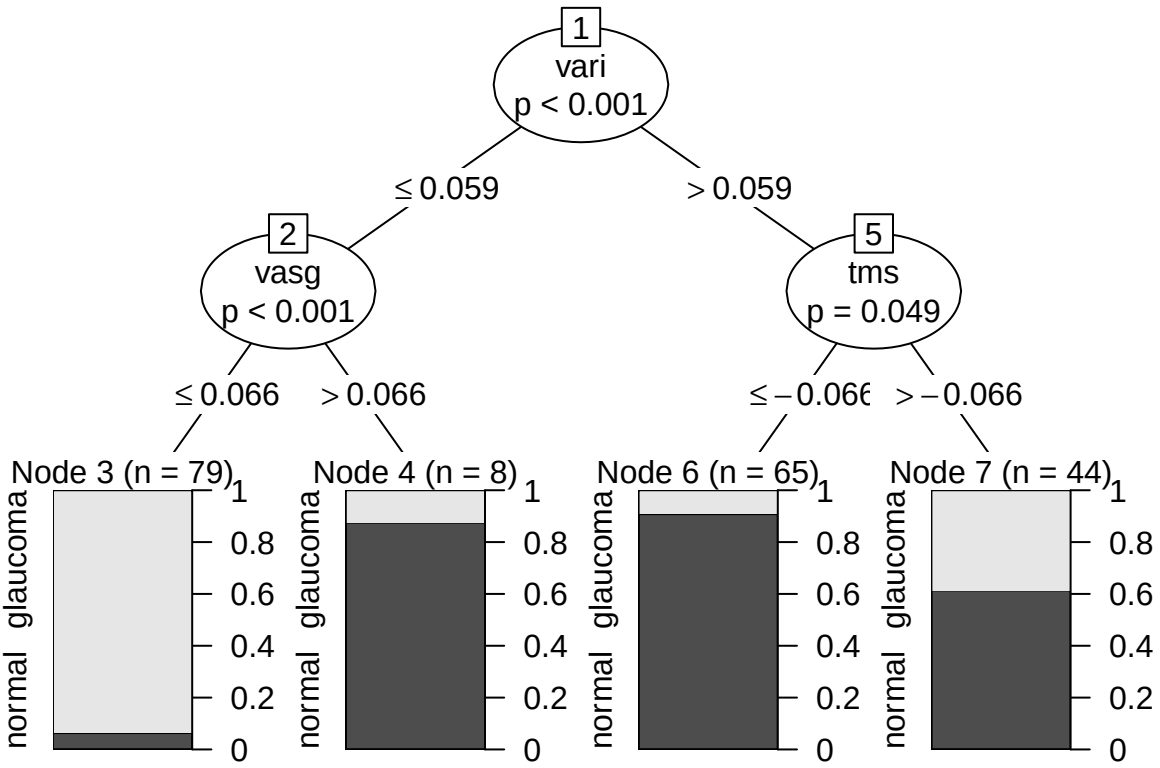
b) Is the first variable in the tree grown in Exercise 4 often selected? And how about the variables that appear further down the tree?

c) Check whether there is multicollinearity in the data: Could that explain that the variables appearing in the tree are not selected in each of the bootstrap samples?

```
library(partykit)
```

```
## Loading required package: grid
```

```
data("GlaucomaM", package = "TH.data")
ct <- ctree(Class ~ ., data = GlaucomaM)
plot(ct)
```



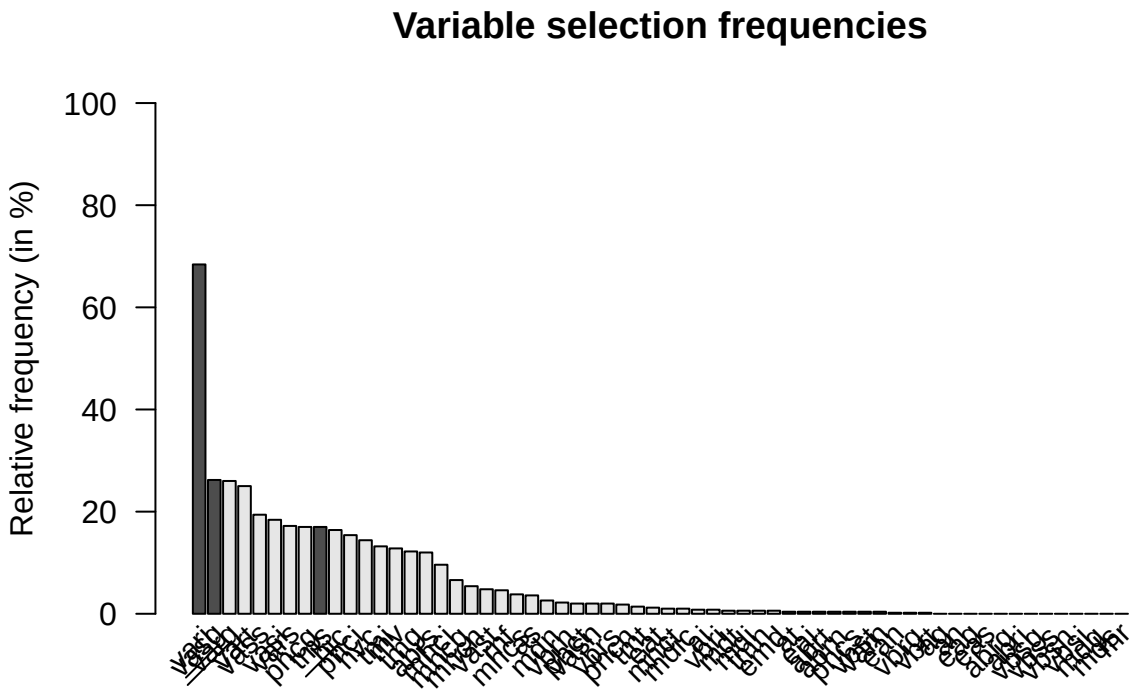

```
library("stablelearner")

set.seed(42)
```

```r
ct_stab <- stabletree(ct)
summary(ct_stab)
```

```
##
## Call:
## ctree(formula = Class ~ ., data = GlaucomaM)
##
## Sampler:
## B = 500
## Method = Bootstrap sampling
##
## Variable selection overview:
##
##       freq *  mean *
## vari 0.684 1 0.722 1
## vasg 0.262 1 0.262 1
## varg 0.260 0 0.260 0
## vart 0.250 0 0.250 0
## vass 0.194 0 0.196 0
## vasi 0.184 0 0.190 0
## vars 0.172 0 0.174 0
## phcg 0.170 0 0.172 0
## tms  0.170 1 0.170 1
## hic  0.164 0 0.166 0
## phci 0.154 0 0.156 0
## hvc  0.144 0 0.144 0
## tmi  0.132 0 0.132 0
## mv   0.128 0 0.132 0
## tmg  0.122 0 0.130 0
## abrs 0.120 0 0.120 0
## mhci 0.096 0 0.096 0
## mhcg 0.066 0 0.066 0
## mhcn 0.054 0 0.054 0
## vast 0.048 0 0.050 0
## rnf  0.046 0 0.046 0
## mhcs 0.038 0 0.038 0
## as   0.036 0 0.036 0
## mdn  0.026 0 0.026 0
## vbrn 0.022 0 0.022 0
## phct 0.020 0 0.020 0
## vasn 0.020 0 0.020 0
## vbrs 0.020 0 0.020 0
## phcn 0.018 0 0.018 0
## tmt  0.014 0 0.014 0
## eat  0.012 0 0.012 0
## mhct 0.010 0 0.010 0
## mdic 0.010 0 0.010 0
## ai   0.008 0 0.008 0
## vbri 0.008 0 0.008 0
## mdt  0.006 0 0.006 0
## mdi  0.006 0 0.006 0
## tmn  0.006 0 0.006 0
## emd  0.006 0 0.006 0
## at   0.004 0 0.004 0
```

```
## eai  0.004 0 0.004 0
## abrt 0.004 0 0.004 0
## abrn 0.004 0 0.004 0
## phcs 0.004 0 0.004 0
## vbst 0.004 0 0.004 0
## varn 0.004 0 0.004 0
## ean  0.002 0 0.002 0
## vbrg 0.002 0 0.002 0
## vbrt 0.002 0 0.002 0
## ag   0.000 0 0.000 0
## an   0.000 0 0.000 0
## eag  0.000 0 0.000 0
## eas  0.000 0 0.000 0
## abrg 0.000 0 0.000 0
## abri 0.000 0 0.000 0
## vbsg 0.000 0 0.000 0
## vbss 0.000 0 0.000 0
## vbsn 0.000 0 0.000 0
## vbsi 0.000 0 0.000 0
## mdg  0.000 0 0.000 0
## mds  0.000 0 0.000 0
## mr   0.000 0 0.000 0
## (* = original tree)
```

```
barplot(ct_stab)
```



```
cor(GlaucomaM[,c("vari", "vasg", "varg", "vart", "vass", "vasi", "vars", "phcg")])
```

```
##            vari       vasg       varg       vart       vass       vasi
## vari  1.0000000  0.4711082  0.9276727  0.6121546  0.3281612  0.5348386
## vasg  0.4711082  1.0000000  0.6181722  0.5601590  0.9196471  0.8230907
## varg  0.9276727  0.6181722  1.0000000  0.7021653  0.4978143  0.5806854
## vart  0.6121546  0.5601590  0.7021653  1.0000000  0.5509784  0.5425974
## vass  0.3281612  0.9196471  0.4978143  0.5509784  1.0000000  0.6434028
## vasi  0.5348386  0.8230907  0.5806854  0.5425974  0.6434028  1.0000000
## vars  0.8013808  0.6025811  0.9311459  0.7353698  0.5734093  0.5288761
## phcg -0.5631943 -0.2728333 -0.5466596 -0.3827756 -0.1519453 -0.3363697
##            vars       phcg
## vari  0.8013808 -0.5631943
## vasg  0.6025811 -0.2728333
## varg  0.9311459 -0.5466596
## vart  0.7353698 -0.3827756
## vass  0.5734093 -0.1519453
## vasi  0.5288761 -0.3363697
## vars  1.0000000 -0.4746608
## phcg -0.4746608  1.0000000
```

Exercise 2: Bootstrapping vs. subsampling

Below, the code for generating one of the datasets I used in the presentation sheets is provided. Using this data and the `stabletree()` function from the `stablelearner` package, evaluate the difference between subsampling and bootstrap sampling in terms of the selection with which noise variables are selected:

```
set.seed(12)
x1 <- rnorm(250, mean = 7, sd = 4)
x2 <- rnorm(250, mean = 14, sd = 4)
y_cont <- 8 + 5*(x1 > 7 & x2 > 13) + rnorm(250, 0, 1.5)
x3 <- rnorm(250)
ex_data <- data.frame(x1, x2, x3, y_cont)
```

a) Fit a ctree which predicts `y_cont` from `x1`, `x2` and `x3`.

b) Use the `stabletree()` function to assess the tree's stability using bootstrap sampling (the default).

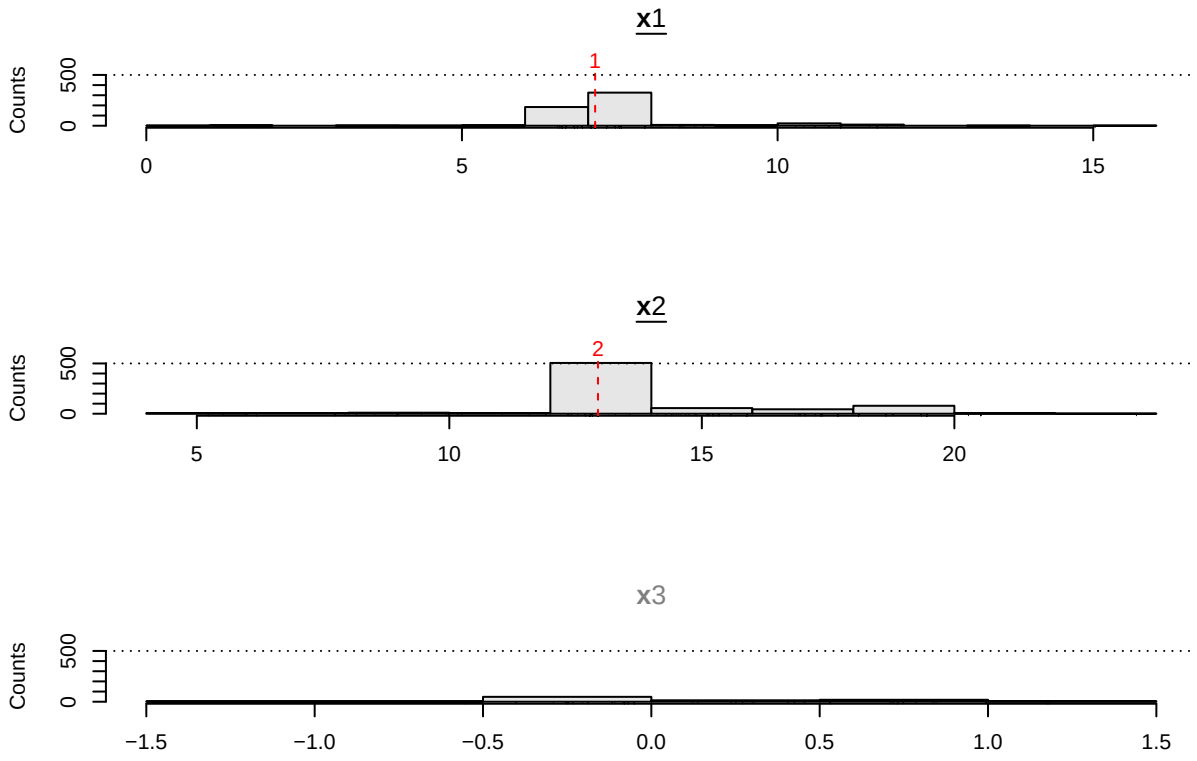
c) Assess the tree's stability using subsamples instead of bootstrap samples. (Hint: Use `?stabletree` and `?bootstrap` to see how you can change the sampler used).

d) Compare the selection frequencies you obtained with both sampling methods.

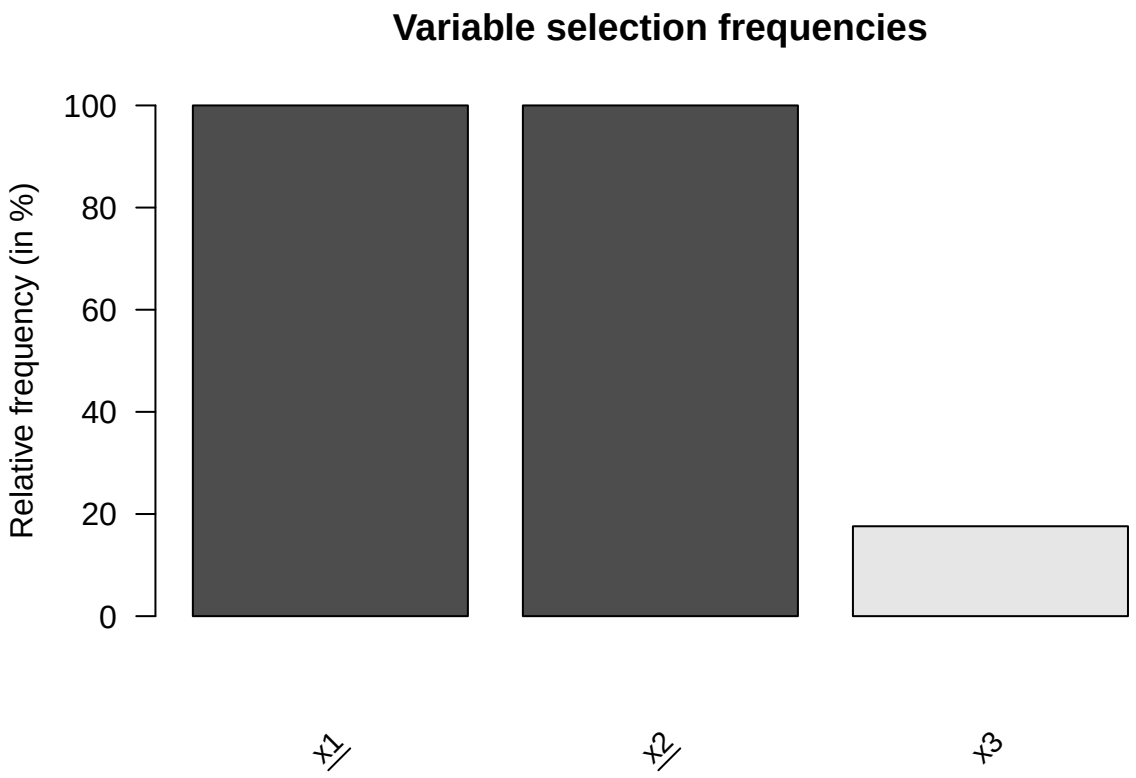
```
library(stablelearner)
set.seed(21)
ct_stab <- stabletree(ctree(y_cont ~ x1 + x2 + x3, data = data.frame(x1, x2, x3, y_cont)))
dev.off() # note: normally not needed
```

```
## null device
##           1
```

```
plot(ct_stab)
```



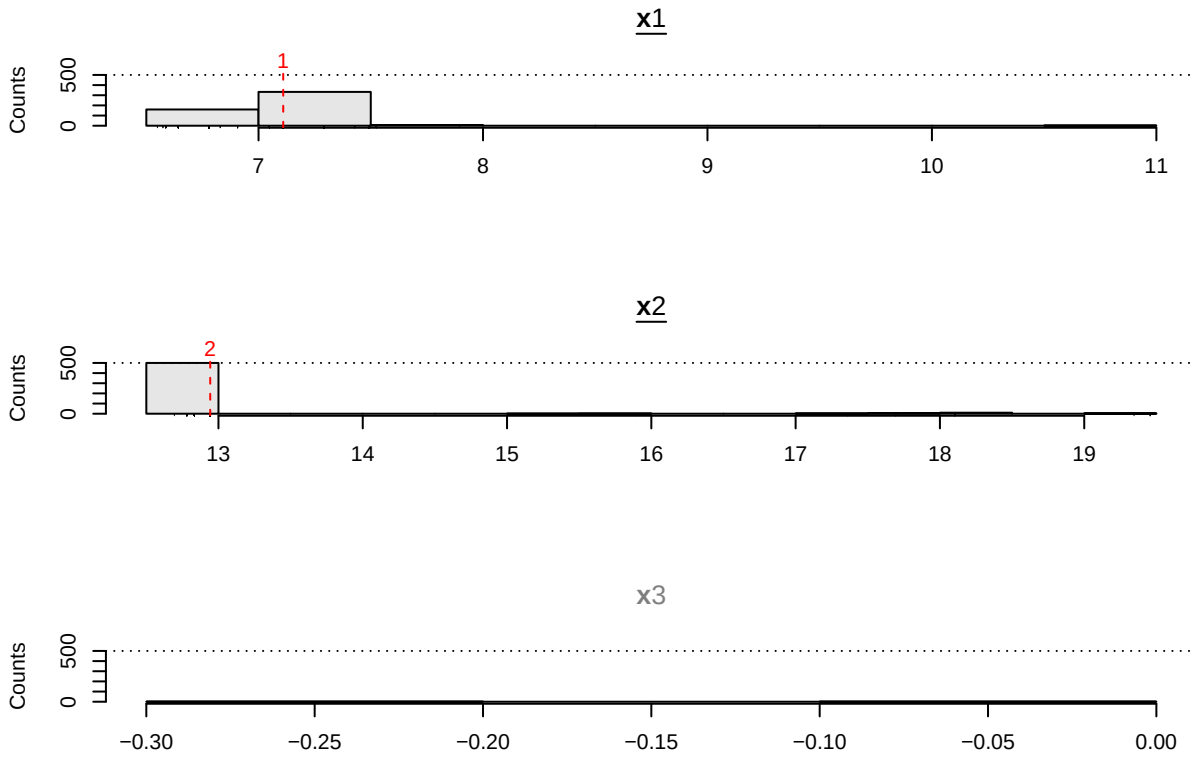

```
barplot(ct_stab)
```



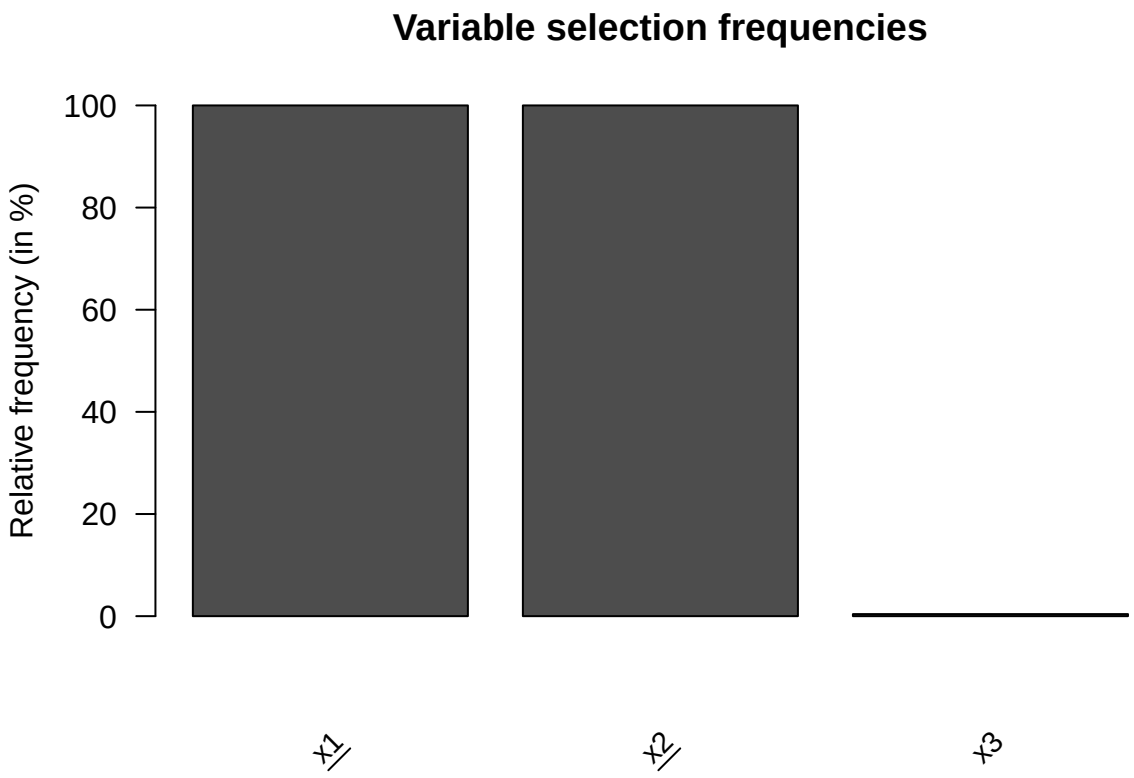

```
dev.off() # note: normally not needed
```

```
## null device
##           1
```

```
set.seed(21)
ct_stab <- stabletree(ctree(y_cont ~ x1 + x2 + x3, data = data.frame(x1, x2, x3, y_cont)), sampler = su
plot(ct_stab)
```




```
barplot(ct_stab)
```




```
dev.off() # note: normally not needed
```

```
## null device
##           1
```