Table of Contents

Česká verze stránek není od roku 2013 aktualizována.

Klasifikační a regresní stromy

Metoda klasifikačních a regresních stromů (*Classification and regression trees, CART*) je obdobou mnohonásobné regrese - vysvětlovaná proměnná je jen jedna, vysvětlujících je několik. Výhodou této metody (oproti regrese) je méně požadavků na kvalitu vysvětlujících proměnných. V ekologii se metoda používá především pro explorativní analýzu - pro popis vztahů mezi vysvětlovanou a vysvětlujícími proměnnými. Pokud je **vysvětlovaná** proměnná kvantitativní, počítá se **regresní strom**, pokud je kvalitativní, jde o **klasifikační strom**.

rpart (library rpart)

Vypočteme regresní strom závislosti počtu druhů na faktorech prostředí (použijeme datový soubor Vltava a jako faktory prostředí použijeme vypočtené Ellenbergovy indikační hodnoty[1)]):

```
vltava.spe <- read.delim
('http://www.davidzeleny.net/anadat-r/data-download/vltava-spe.txt',
row.names = 1)
vltava.env <- read.delim
('http://www.davidzeleny.net/anadat-r/data-download/vltava-env.txt')
env <- vltava.env[, 20:25]
```

Počet druhů vypočteme pomocí funkce `specnumber` z knihovny `vegan`:

```
library (vegan)
S <- specnumber (vltava.spe)
```

Následuje vlastní model pro výpočet regresního stromu, pro který je použita funkce `rpart` ze stejnojmenné knihovny. Pokud chcete získat stejný strom jako v této úloze, použijte funkci `set.seed` a stejnou numerickou hodnotu, která nastaví výchozí hodnotu pro generátor pseudonáhodných čísel na vašem počítači.

```
library (rpart)
# set.seed (1234)   # pokud chcete získat stejný strom jako v této úloze,
odkomentujte tuto funkci
tree.1 <- rpart (S ~ ., env)

tree.1
```

Funkce vrací textovou podobu regresního stromu s čísly jednotlivých uzlů a terminálních lístků (ty jsou označeny hvězdičkou):

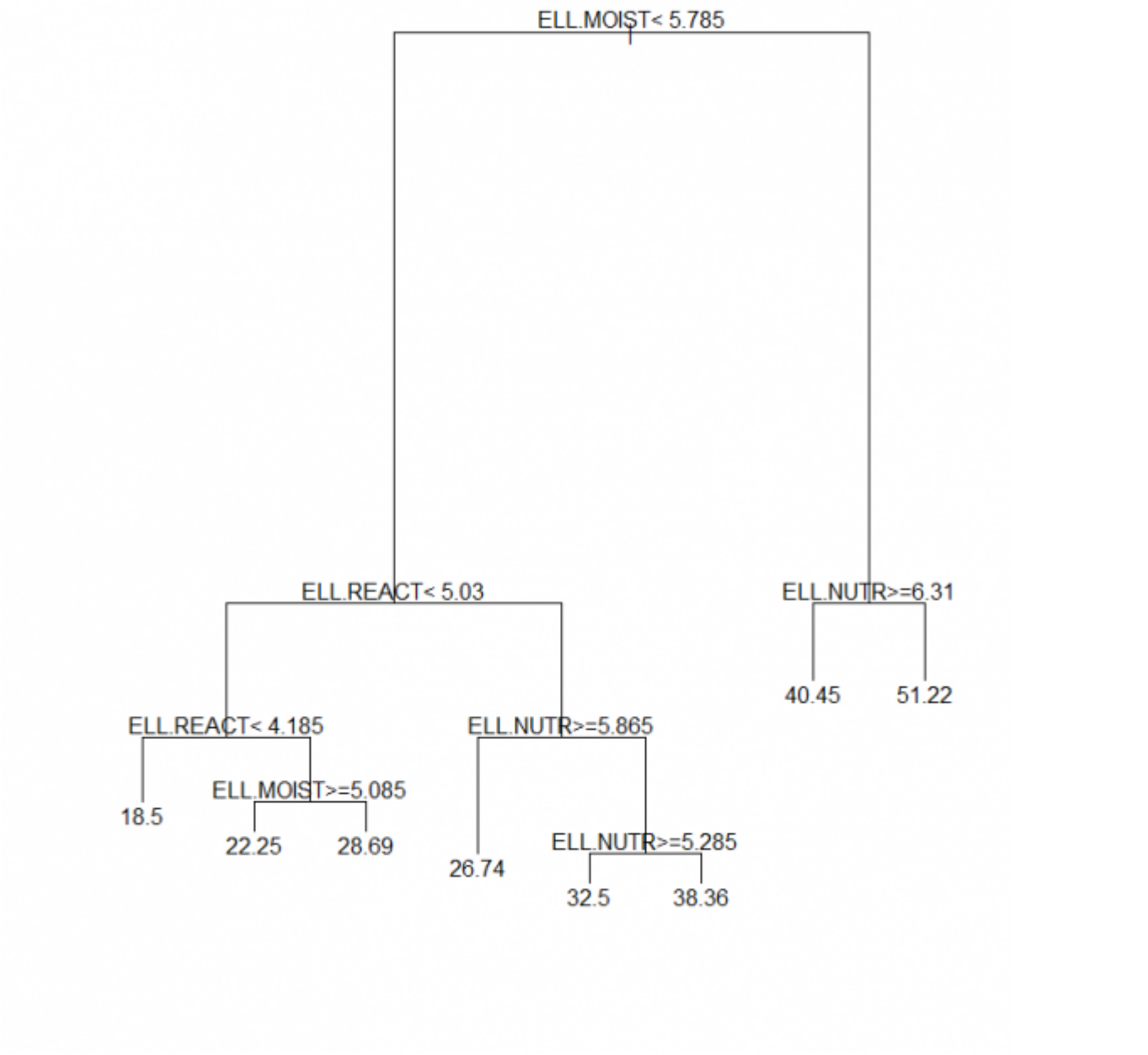
```
n= 97
```

```
node), split, n, deviance, yval
      * denotes terminal node

 1) root 97 11009.0100 32.22680
   2) ELL.MOIST< 5.785 77  4824.8050 28.83117
     4) ELL.REACT< 5.03 32  1427.9690 24.53125
       8) ELL.REACT< 4.185 8   104.0000 18.50000 *
       9) ELL.REACT>=4.185 24   935.9583 26.54167
        18) ELL.MOIST>=5.085 8    95.5000 22.25000 *
        19) ELL.MOIST< 5.085 16   619.4375 28.68750 *
     5) ELL.REACT>=5.03 45  2384.4440 31.88889
      10) ELL.NUTR>=5.865 19   369.6842 26.73684 *
      11) ELL.NUTR< 5.865 26  1141.8850 35.65385
        22) ELL.NUTR>=5.285 12   395.0000 32.50000 *
        23) ELL.NUTR< 5.285 14   525.2143 38.35714 *
   3) ELL.MOIST>=5.785 20  1878.2000 45.30000
     6) ELL.NUTR>=6.31 11   676.7273 40.45455 *
     7) ELL.NUTR< 6.31 9   627.5556 51.22222 *
```

Strom můžeme nakreslit takto:

```
plot (tree.1)
text (tree.1)
```

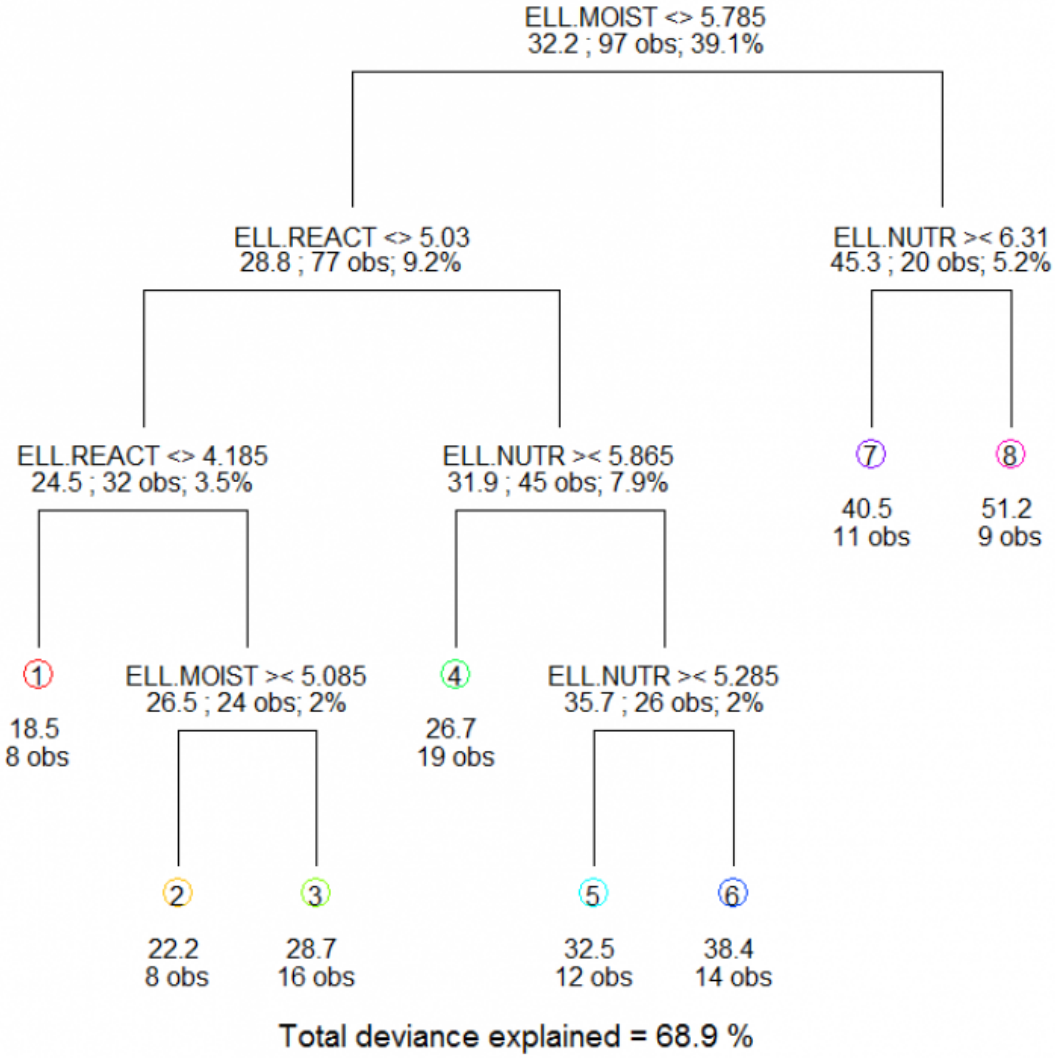


draw.tree (library maptree)

Lepší grafické zobrazení regresních a klasifikačních stromů nabízí funkce `draw.tree` z knihovny `maptree`:

```
library (maptree)
draw.tree (tree.1, nodeinfo = T, digits = 1)
```

Argument `nodeinfo` ovlivňuje množství informací, které se v obrázku vykreslí - konkrétně ke každému uzlu přidá detailní informaci o množství vysvětlené variability. Argument `digits` ovlivňuje počet desetinných míst, který se zobrazuje u jednotlivých hodnot závislé proměnné (počet druhů v tomto případě) - pokud ho nezadáte, zobrazí se průměrné počty druhů v nodech s přesností na nesmysleně vysoký počet desetinných míst).



plotcp a rsq.rpart (library rpart)

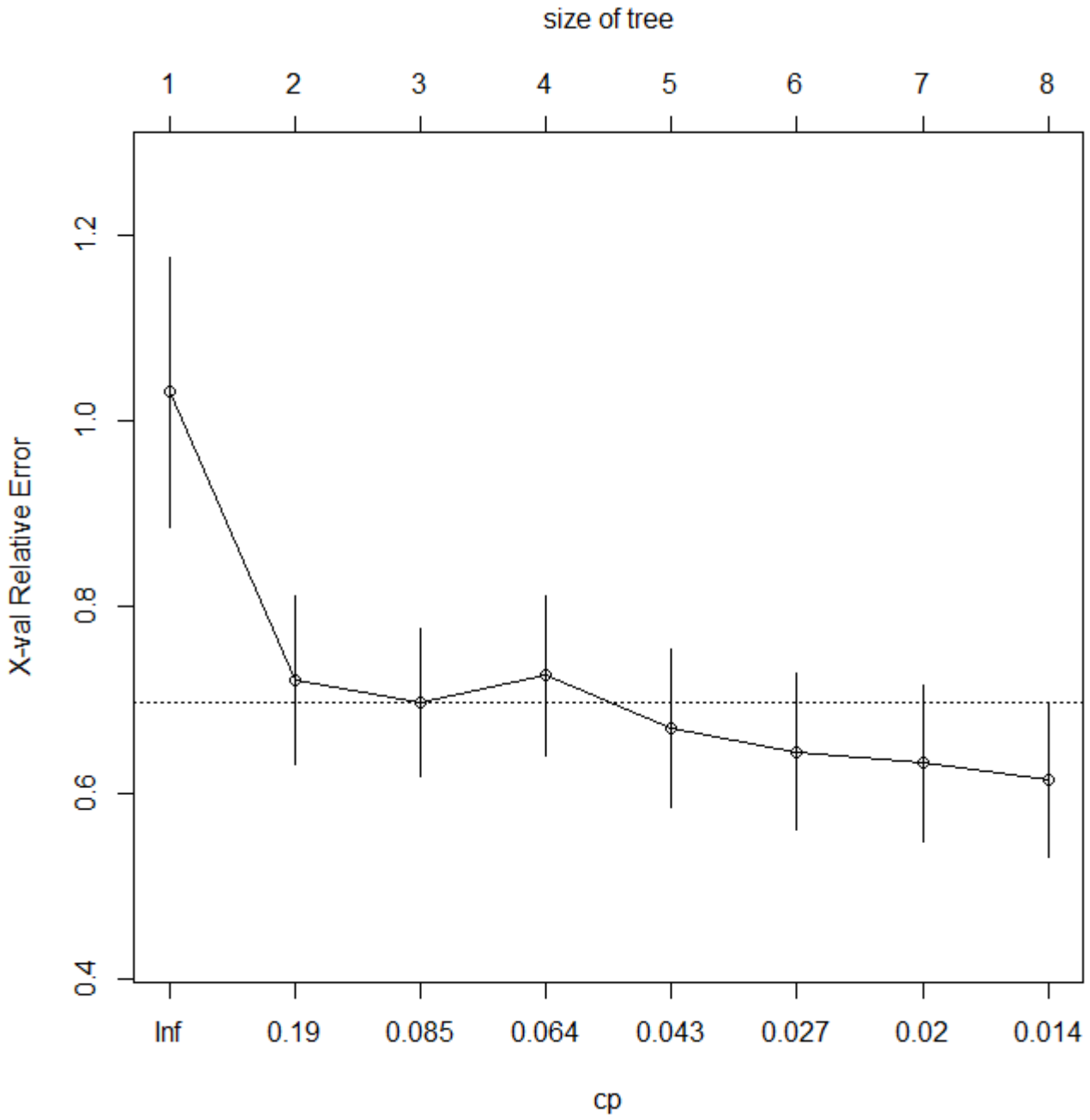
Algoritmus klasifikačních a regresních stromů má tendenci vytvářet komplikované struktury - stromy s mnoha koncovými větvemi. Je proto vhodné stromy nějakým způsobem zjednodušit. Míra zjednodušení bude záležet na tom, co od stromu očekávám. Pokud mi jde o zobecnitelný popis situace, pak je na místě strom prořezat pořádně, aby se dobře interpretoval a zároveň aby v něm byla určitá míra zobecnění. Pokud je naopak účelem vytvořit strom pro predikci z nových dat, může být složitější, aby dosáhl přesnějších výsledků.

Rozhodnutí o komplexnosti stromu bývá založené na tzv. krosvalidaci, křížovém zhodnocení. Datový soubor se rozdělí např. na 10 skupin vzorků, náhodně se vybere 9 a na nich se vytvoří nový strom. Nepoužité vzorky z desáté skupiny se pak použijí k predikci a vyhodnotí se kvalita této predikce. Pak

se odřízne poslední větvička, znovu se vyhodnotí kvalita predikce, a v prořezávání se pokračuje až na pařez. Toto se zopakuje pro všechny kombinace 9 podskupin vzorků.

Následující funkce slouží k zobrazení výsledků krosvalidace daného (tu provádí už funkce `rpart`). Funkce `plotcp` kreslí vztah mezi složitostí stromu (vodorovná osa) a hodnoty chyb v predikci modelem pro danou velikost (svislá osa). Křivka na začátku strmě klesá, tzn. že složitost modelu je vyvážena lepší predikcí. Jakmile začne růst, znamená to že složitější model jde na úkor kvality predikce. Jedno z možných pravidel při rozhodnutí o velikosti stromu říká vybrat takové dělení, které se jako první vyskytuje pod horizontální čárou.

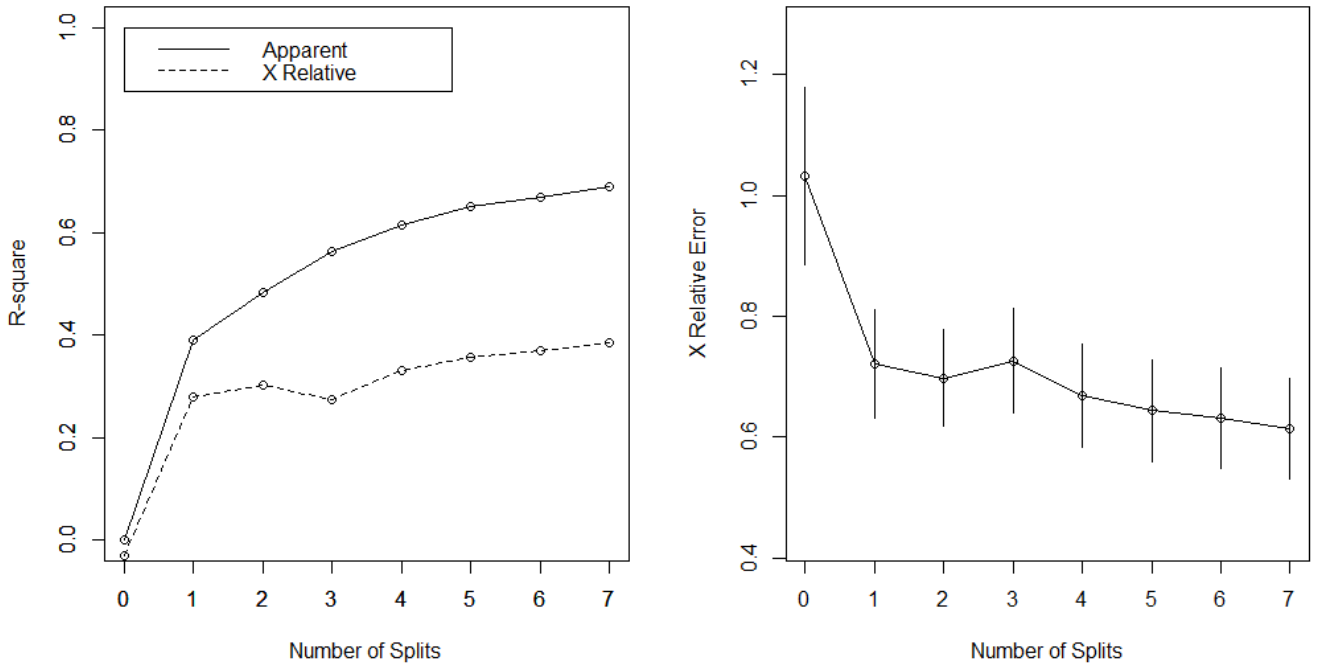
```
plotcp (tree.1)
```



Podobný obrázek zobrazuje i funkce `rsq.rpart` - ta kreslí obrázky dva za sebou, je ale dobré si je zobrazit vvedle sebe, proto funkce `par` s argumentem `mfrow`. Druhý obrázek je identický s předchozím výstupem funkce plotcp, ale bez dalších informací. První obrázek ukazuje, jak stoupá koeficient determinace (vysvětlená vvariabilita) modelu s jeho složitostí. Plná čára (apparent) ukazuje

zjevnou vysvětlenou variabilitu, tedy tu kkterou vysvětlí strom o dané velikosti na všech datech. Přerušovaná čára ukazuje na krosvalidovaný odhad koeficientu determinace pro nová data. V našem případě roste kvalita předpovědi modelem na nových datech (z krosvalidace) do velikosti stromu 2, pak opět klesne.

```
par (mfrow = c(1,2))
rsq.rpart (tree.1)
```

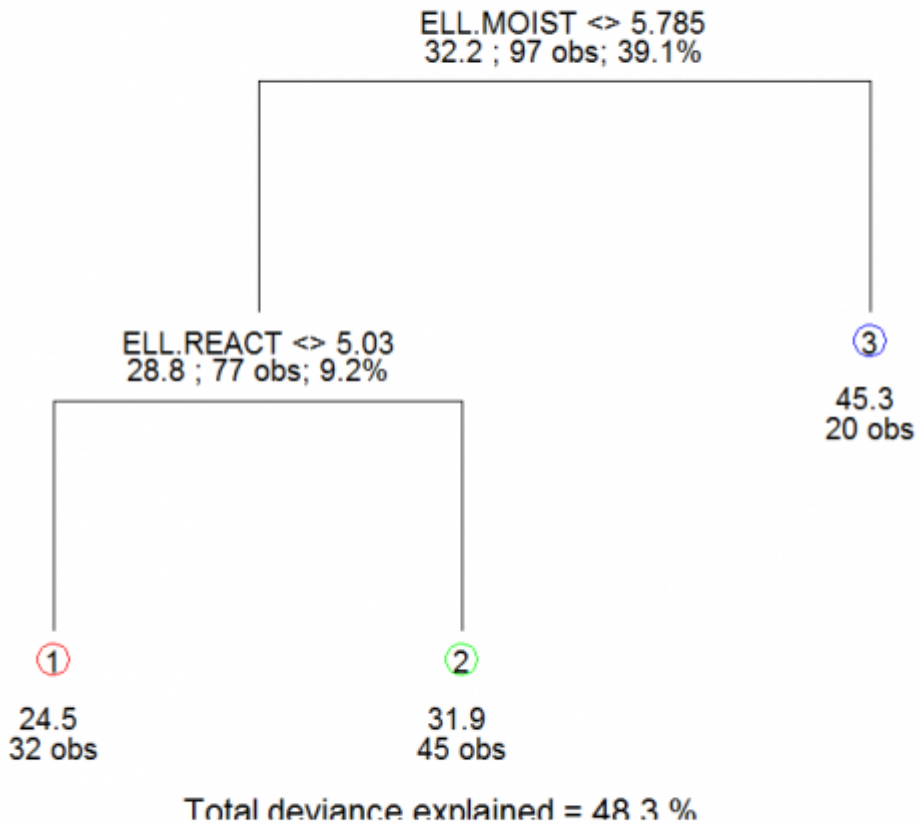


Interpretace výsledků krosvalidace není jednoznačná - vypadá to, že bychom mohli vybrat jak velmi jednoduchý strom s dvěmi větvemi, tak i celý strom se sedmi větvemi. Výsledek se navíc v každé analýze bude měnit - zkuste analýzu příkazem `rpart` několikrát zopakovat, pokaždé dostanete mírně jiný výsledek (je to způsobeno právě krosvalidací, resp. tím, že dělení do 10 podskupin je pokaždé provedeno jinak).

prune (library rpart)

Pokud se rozhodneme rozsáhlý strom, který vrátila předchozí analýza, zjednodušit (prořezat) podle výsledků krosvalidace, použijeme k tomu příkaz `prune` z knihovny `rpart`. Argument `cp` v této funkci představuje *copmplexity parameter*, vyjadřující složitost stromu který se má zachovat - najdete ho na horizontální ose v obrázku vygenerovaném funkcí `plotcp`[2)]:

```
tree.2 <- prune (tree.1, cp = 0.085)
draw.tree (tree.2, nodeinfo = T, digits = 1)
```



Rozšiřující informace

Podrobný popis, jak spočítat regresní a klasifikační stromy v R, najdete ve skriptech Petra Šmilauera *Moderní regresní metody* v kapitole 7 (pdf). Velmi detailní rozbor problematiky rozhodovacích stromů s teorií týkající se i dalších metod najdete ve skriptech Kláry Komprdové *Rozhodovací stromy a lesy* (pdf).

Cvičení 1

Vypočtěte klasifikační strom, ve kterém bude vysvětlovaná proměnná zařazení vegetačních snímků do vegetačního typy, a vysvětlující proměnné budou abundance jednotlivých druhů v druhové matici. Použijte soubor dat z údolí Vltavy. Klasifikace jednotlivých snímku do vegetačních typů je obsažena v proměnné GROUP v datovém rámci vltava.env. Druhová data před použitím odmocněte. Nakreslete

klasifikační strom pomocí funkce `draw.tree`.

Nápověda: použijte funkci `rpart`. Na levé straně rovnice bude proměnná vltava.env$GROUP - pozor ale, proměnnou je třeba zadat jako faktor, tedy obalit funkcí `as.factor`, jinak bude použita jako proměnná kvantitativní a výsledný strom bude regresní, ne klasifikační.

Řešení cvičení

1)
Průměrné Ellenbergovy indikační hodnoty jsou vypočtené ze stejných dat, jako počty druhů, které budou v následující analýze použity jako závislá proměnná. Tím, že vysvětlující (průměrné Ell. hodnoty) i závislá proměnná na sobě nejsou nezávislé dává následující analýza příliš optimistické výsledky - rozumněj příliš vysoké hodnoty vysvětlené variability. Pokud nám jde jen o popis vztahu, pak to nejspíš nevadí. Není ale možné v analýze průměrné EIH míchat s jinými, měřenými faktory prostředí, protože průměrné EIH se právě díky nezávislosti budou tvářit jako lepší proměnné a budou umístěny výše ve stromu. Více k této problematice zde a zde.

2)
jinak škálovanou hodnotu CP najdete také ve výpisu funkce `summary` - pro příkaz `prune` je ale třeba používat hodnoty z grafu