Příklad II: Strom typu CHAID
Cvičení v programu R.

```
> titanic<-read.delim("clipboard",row.names=1)
> summary(titanic)
```

Knihovna pro výpočet stromu CHAID není ve standardní nabídce instalace R, je však možné ji nainstalovat pomocí příkazu:

```
> install.packages("CHAID", repos="http://R-Forge.R-project.org")
```

Současný algoritmus umožňuje použití pouze kategoriálních a ordinálních proměnných. Spojitá data musí být před použitím převedena na ordinální.

Skript obsahuje dvě části. První je pro nastavení parametrů algoritmu *chaid_control*. Zde je uvedeno defaultní nastavení:

```
> chaid_control(alpha2 = 0.05, alpha3 = -1, alpha4 = 0.05,
              minsplit = 20, minbucket = 7, minprob = 0.01)
```

Parametr *alpha2* určuje hladinu významnosti použitou pro slučování kategorií prediktoru (krok 3), *alpha3* – pokud je zadána kladná hodnota < 1 je hladina významnosti použita také pro rozdělení již dříve sloučených kategorií prediktoru (krok 4), jinak je tento krok vypuštěn (defaultní nastavení), *alpha4* určuje hladinu mezní významnosti pro adjustovanou hodnotu prediktoru (krok 6), *minsplit* definuje minimální počet pozorování, při němž již nedochází k dalšímu rozdělení uzlu, *minbucket* je minimální počet pozorování v potenciálním terminálním uzlu a *minprob* udává minimální frekvenci pozorování v terminálních uzlech.

```
> chaid(formula, data, subset, weights, na.action = na.omit,
      control = chaid_control())
```

Funkce *chaid* spouští výpočet stromu a lze u ní nastavit další parametry týkající se souboru jako je: *subset* pro definování testovacího souboru (je-li k dispozici), *weights* pro nastavení vah u jednotlivých pozorování, parametr *na.action*, který určuje, jak bude naloženo s nevyplněnými hodnotami (defaultní nastavení je odstranění řádků s prázdnou hodnotou z výpočtu) a *control*, který definuje parametry algoritmu popsané výše.

```
> library("CHAID")
> set.seed(123)
> ctrl <- chaid_control(minsplit = 200, minprob = 0.1, alpha2 = 0.05,
alpha3 = -1, alpha4 = 0.05)
> chaidTitanic <- chaid(survival ~ class+age+gender, data = Titanic,
control = ctrl)
```

Výsledky v textové podobě zobrazíme pomocí funkce *print*. Hodnota α pro sloučení i výsledné testování sloučených kategorií byla nastavena na 0,05. S touto hladinou významnosti byly porovnávány *p* hodnoty kontingenčních tabulek přežití versus věk, pohlaví a třída.

```
> print(chaidTitanic)
```

Výsledkem je, podobně jako v případě stromu typu CART, hierarchická textová podoba stromu, kdy je pro každý uzel uvedena hodnota kategorie prediktoru. Pro terminální uzel je

dále zobrazena kategorie závisle proměnné, počet pozorování v uzlu a klasifikační chyba. Výsledná hodnota terminálního uzlu (přežil/nepřežil) je určena jako převládající kategorie závisle proměnné v tomto uzlu a klasifikační chyba je procento chybně klasifikovaných pozorování.

```
> plot(chaidTitanic,cex=0.6)
```

Příklad III: Strom typu PRIM

Cvičení v programu R.

Podíváme se na stejný příklad, jaký byl použit pro regresní stromy: závislost koncentrace ozónu (ppb) na teplotě (stupně Fahrenheita), rychlosti větru (míle/h) a intenzitě slunečního záření (cal/cm^2). Soubor obsahuje 111 měření.

Načteme knihovnu *lattice*, která obsahuje výše popsaný datový soubor se jménem *environmental* a knihovnu *prim* s funkcemi pro výpočet:

```
> library(prim)
> library(lattice)
> data(environmental)
```

Do proměnné *Y* uložíme závisle proměnnou koncentraci ozónu, do proměnné *X* prediktory teplotu a rychlost větru:

```
>  y <-environmental[,1]
>  x <-environmental[,3:4]

> summary(environmental)
```

U funkce *prim* lze nastavit různé parametry: *peel.alpha* a *paste.alpha* určují podíl pozorování, o které se okno bude zmenšovat, respektive zvětšovat; *mass.min* je minimální podíl pozorování z celkového souboru (defaultně 0,05). Za diskriminační hladinu proměnné *Y* je používán průměr, pokud je parametr *threshold.type* roven 1, je hledáno okno s hodnotami závisle proměnné >= než průměr, při nastavení na nulu hledáme hodnoty <= průměru. Nastavením *threshold.type=0* můžeme zvolit rozsah hodnot závisle proměnné.

Výsledky výpočtu uložíme do objektu *prim.ozon*:

```
> prim.ozon <- prim(x , y = y, threshold.type = 1)
```

Níže jsou zobrazeny výsledky pro koncentraci ozónu. Sloupeček *box-mean* obsahuje průměrnou hodnotu závisle proměnné v okně a *box-mass* podíl pozorování v okně z celkového souboru. Hvězdička označuje „zbytek“ datového souboru, který již nebyl použit pro rozdělení. Následují pravidla, která definují rozsah okna.

```
> summary(prim.ozon, print.box = TRUE)
```

Zobrazíme výsledný graf a jednotlivá měření. Plná kolečka obsahují hodnoty koncentrace, které jsou vyšší než průměr.

```
> plot(prim.ozon, col = "transparent",cex.axis=1.5, cex=1.5, cex.lab=1.5)
> points(x[y > 42.1, ], pch=16, cex=1.5)
> points(x[y < 42.1, ], cex=1.5)
```

Ke stávajícím prediktorům přidáme ještě intenzitu slunečního záření.

```
>  x <-environmental[,2:4]
> prim.ozon1 <- prim(x , y = y, threshold.type = 1)
> summary(prim.ozon1, print.box = TRUE)
```

Výsledný graf zobrazuje pozorování s vyššími koncentracemi (box1) v prostoru tří prediktorů.

```
> plot(prim.ozon1, cex=1.5, pch=16,cex.axis=1.5, cex.lab=1.5, col='black')
```

Příklad IV: Strom typu MARS

Ukázkový příklad ke cvičení v programu R.

Datový soubor obsahuje údaje o průměru, výšce a objemu 31 pokácených třešní. Průměr je uveden v palcích a byl změřen ve výšce 4-6 stop. Podíváme se na závislost objemu kmene (v krychlových stopách) na jeho průměru při použití lineární regrese (LR) a metody MARS.

```
> trees
> summary(trees)
```

Načteme knihovnu *earth* pro výpočet metody MARS.

```
> library(earth)
```

U funkce *earth* můžeme nastavit parametry pro výpočet:

nk specifikuje maximální počet členů v rovnici (včetně interceptu) při postupném výběru členů před tím, než dojde k jejich zpětnému odstraňování pomocí krosvalidace. Defaultní nastavení je určeno z počtu prediktorů a je vhodné jej ověřit pomocí argumentu *trace* (nastavením na jedna).

Velmi důležitým parametrem je *degree*, který určuje maximální stupeň interakcí v modelu. Defaultně je nastaven na 1, což znamená aditivní model bez interakcí.

Dalším parametrem je *penalty*, který se rovná konstantě *c* použité při krosvalidaci a je nastaven na 3, pokud nejsou zahrnuty interakce (*degree* = 1), nebo na 2 pro rovnici s interakcemi (*degree* > 1).

Další kritéria specifikují nastavení krosvalidace: *nfold* je počet rozdělení na podsoubory a je roven hodnotě *k*. Defaultně se krosvalidace neprovádí (*nfold* = 0), jestliže je *nfold* > 0, nejprve je vytvořen model na všech pozorováních, následně je vytvořeno *k* modelů a R^2 je měřen na *k* testovacích souborech. Výsledná hodnota R^2 z krosvalidace *cv.rsq* je průměrem všech R^2 na testovacích souborech. Lze nastavit i počet opakování krosvalidace parametrem *ncross*.

```
> tresne <- earth(Volume ~ ., data = trees, degree =1, nfold = 1)
```

Příkazem *plotmo* zobrazíme závislosti jednotlivých prediktorů z metody MARS. Hodnoty jsou mediány prediktorů.

```
> plotmo(tresne)
```

Pomocí *summary* zobrazíme výsledky metody v textové podobě.

```
> summary(tresne)
```

Nastavením parametru *trace* = 1 zjistíme postup při vytváření modelu: v jakém pořadí byly vybrány jednotlivé členy a jak se změnil R^2 po zpětném odstranění některých členů.

```
> tresne <- earth(Volume ~ ., data = trees, degree =1, nfold = 1, trace
= 1 )
```

Nyní se podíváme, jak se změní výsledky modelu, pokud budou zahrnuty i interakce proměnných.

```
> tresne1 <- earth(Volume ~ ., data = trees, degree = 2, nfold = 1)
> summary(tresne1)
```

Pomocí funkce *evimp* můžeme spočítat významnost proměnných v modelu na základě jejich příspěvku k vyčerpané variabilitě, přičemž hodnoty jsou standardizovány na 0-100. Parametr *trim* určuje, zda budou zobrazeny výsledky u proměnných, které nebyly vybrány pro žádný podsoubor.

```
> tresne.imp <- evimp(tresne, trim=FALSE, sqrt =TRUE)
> print(tresne.imp)
```

Hodnoty ve sloupečku *nsubsets* označují významnost na základě počtu, kolikrát se proměnná objevila v rovnici při použití různých podsouborů. Ve sloupci *gcv* jsou uvedeny významnosti s použitím GCV kritéria, *rss* je významnost určená použitím celkové residuální sumy čtverců.

Srovnání relativních významností prediktorů zobrazíme v grafu užitím funkce *plot*.

```
> plot(tresne.imp)
```

Další zajímavou funkcí je *predict*, která nám umožní predikci nových pozorování.

```
> predikovane<- predict(tresne)
> trees[1:10,3]
> predikovane[1:10]
> predict(tresne, c(10,80))
```