Česká verze stránek není od roku 2013 aktualizována. Aktuální verzi této stránky najdete v její anglické verzi zde.

Numerická klasifikace

Hierarchická klasifikace

hclust

Počítá vlastní klasifikaci, a vyžaduje dvě hlavní informace (argumenty `d` a `method`): matici distancí mezi vzorky (viz kapitola Ekologická podobnost), a název shlukovacího algoritmu (*cluster algorithm*). Shlukovací algoritmy jsou v principu tří hlavních typů: *single linkage* (metoda jednospojná), *complete linkage* (metoda všespojná) a kompromisní *average linkage*. V ekologii se nejčastěji používá třetí typ, pod kterým se skrývá například populární Wardova metoda nebo beta flexible. Metoda `single linkage` data většinou silně řetězí, `complete linkage` naopak často vytváří téměř "hrábě", což je situace, kdy se jednotlivé shluky spojí až v maximální vzdálenosti na horní hraně dendrogramu. Naopak Wardova metoda je oblíbená právě proto, že dělá pěkné kompaktní klastry - pozor ale na to, že Wardovu metodu bychom neměli kombinovat s distancemi, které nejsou striktně metrické, což je například Bray-Curtis distance. Beta flexible je zase hojně používaná proto, že nastavením koeficientu *beta* mohu ovlivnit vlastní řetězení [1].

Jako `veg.data` jsem v příkladech použil soubor `vltava.spe`, jako `env.data` soubor `vltava.env` - viz Lesní vegetace údolí Vltavy

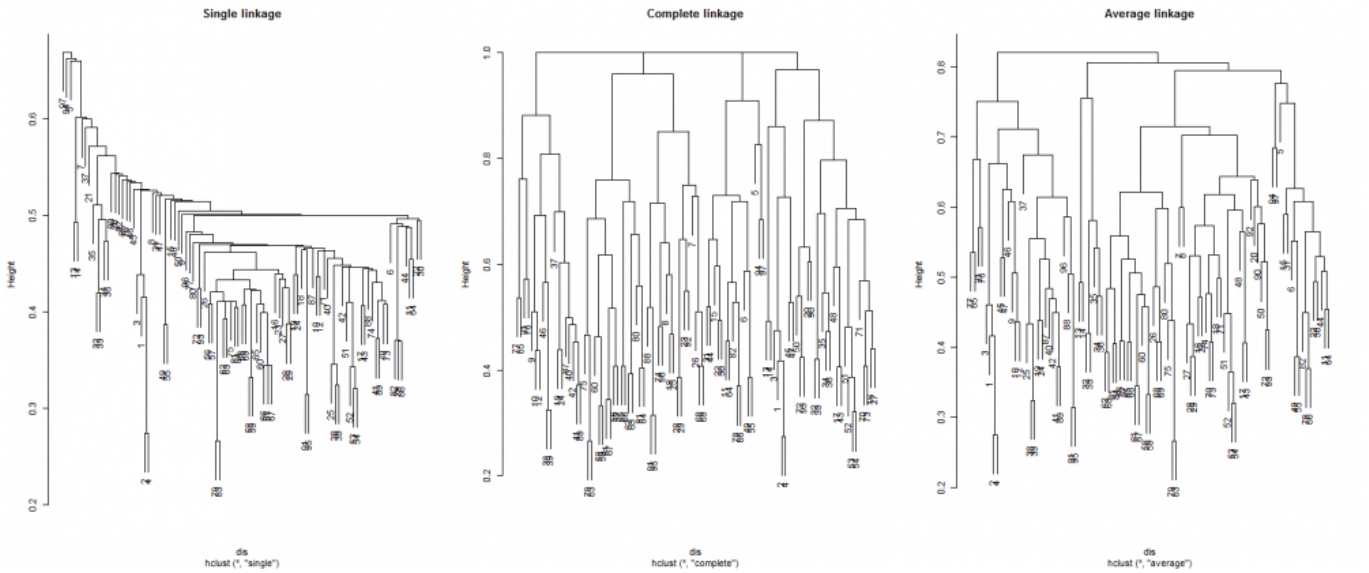
```
library (vegan)
dis <- vegdist (sqrt (veg.data), method = 'bray') # předpokládám, že
veg.data obsahují procentické pokryvnosti, proto ta odmocninová transformace
cluster.single <- hclust (d = dis, method = 'single')
cluster.complete <- hclust (dis, 'complete')
cluster.average <- hclust (dis, 'average')
```

plot

Nakreslí výsledný dendrogram. Pokud budete hledat nápovědu k této funkci, vězte, že ve skutečnosti se funkce jmenuje `plot.hclust` - pokud chci kreslit objekty, které vznikout pomocí funkce `hclust`, těmto objektům se přiřadí třída (*class*) `hclust`. Funkce `plot` se nejdříve podívá, do jaké třídy patří objekt, který chci nakreslit (my se na to můžeme podívat také, např: `class (cluster.average)`), a vlastní kreslení pak svěří funkci, která je na danou třídu specializovaná (`plot.hclust`). Proto musíte zadat `?plot.hclust`, protože nápověda k `?plot` vás dovede jen k velmi obecné funkci `plot`.

```
plot (cluster.average)
# pokud chci nakreslit všechny tři dendrogramy do jednoho obrázku:
```

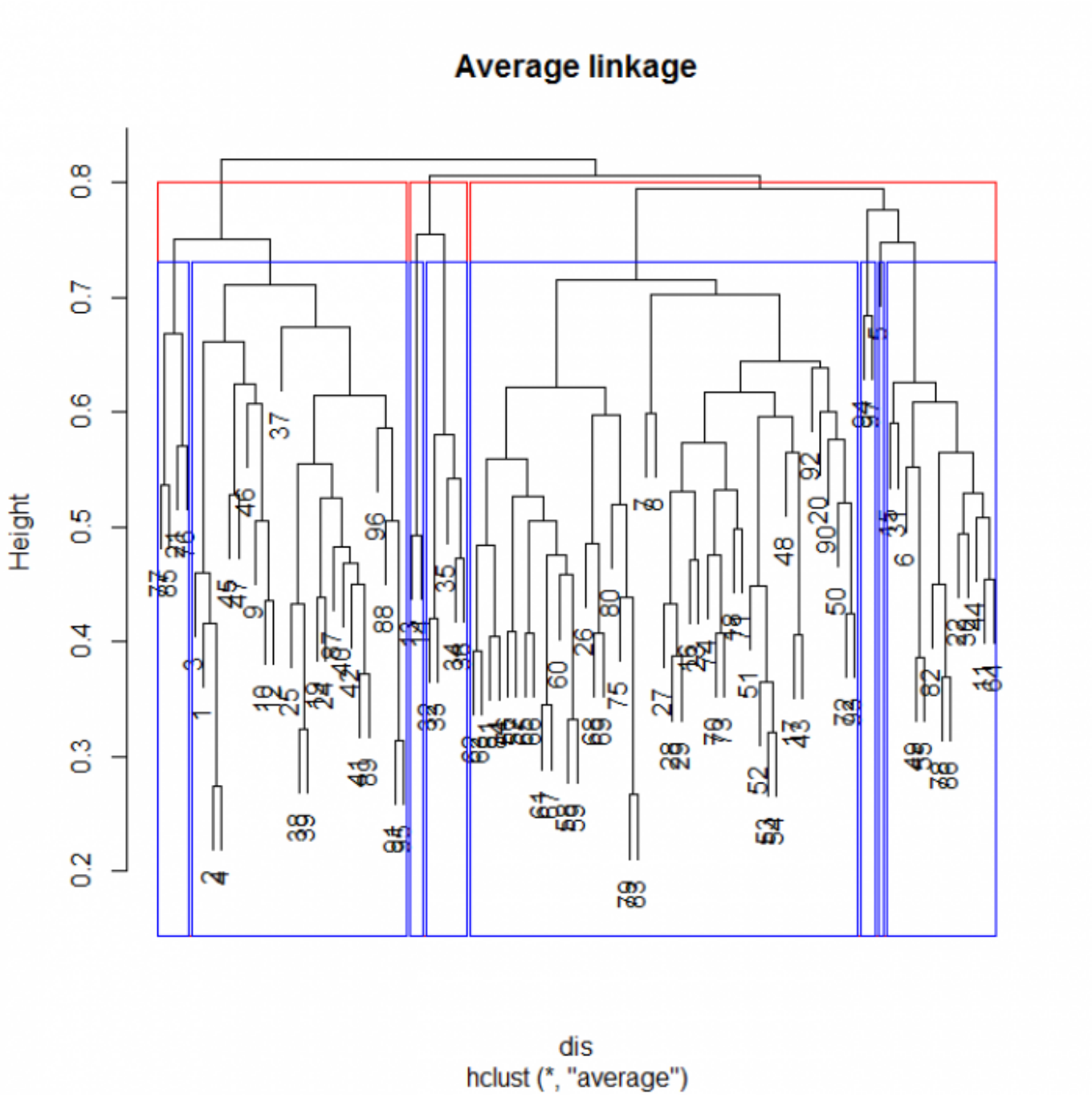
```
par (mfrow = c (1,3))
plot (cluster.single, main = 'Single linkage')
plot (cluster.complete, main = 'Complete linkage')
plot (cluster.average, main = 'Average linkage')
```



rect.hclust

Rozdělí dendrogram do daného počtu skupin pomocí obdélníků (alternativné může dendrogram rozdělit ve specifikované úrovni).

```
plot (cluster.average, main = 'Average linkage')
rect.hclust (cluster.average, k = 3)
rect.hclust (cluster.average, k = 8, border = 'blue') # argumentem border se dá ovlivnit barva obdélníku
```



cutree

Vrací vektor, který obsahuje informaci o přiřazení jednotlivých vzorků ke klastrům. Definovat se dá buď počet klastrů, na které se mají vzorky rozdělit (argument k), nebo výška, ve které se má dendrogram říznout (argument h).

```
klastry <- cutree (cluster.average, k = 5)
klastry
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54
 1  1  1  1  2  2  3  2  1  1  2  1  4  1  2  2  2  2  1  2  1  2  2  1  1
2  2  2  2  2  2  2  2  2  2  2  1  1  1  1  1  1  2  1  2  1  1  2  2  2  2
```

```
2  2  2
55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97
 2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  1  1  2  1  2  2  1  1  2  2
2  2  2  2  2  2  2  2  2  2  2  2  2  2  5  2  2  5
```

Alternativní funkce pro hierarchickou klasifikaci

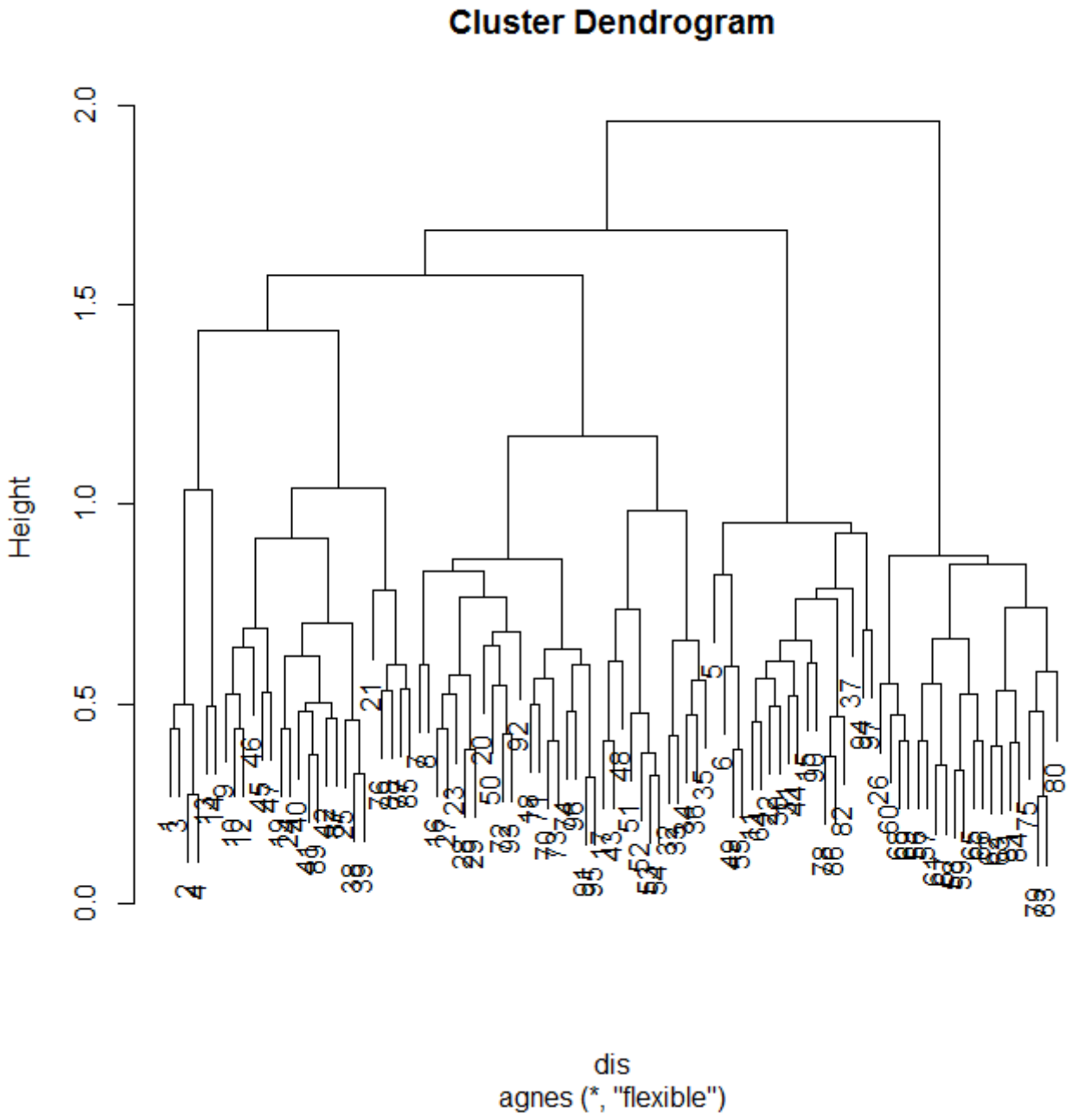
agnes (knihovna cluster)

Zahrnuje 6 klastrovacích algoritmů, z nichž některé nejsou obsažené ve funkci `hclust`. Je to například metoda `flexible`, známá také jako *beta flexible*, která se často používá na ekologická data a jejíž výhoda spočívá v tom, že nastavením argumentu beta je možné změnit míru, s jakou se výsledný dendrogram řetězí (pokud je beta ~ +1, pak je řetězení maximální, naopak při beta = -1 je výsledek obdobný metodě complete linkage). V eRku to ale není tak jednoduché - ve funkci `agnes` je při nastavení `method = "flexible"` vyžadován další argument `par.method`, který vyžaduje nastavení jednoho až čtyř parametrů. Pro detaily viz nápovědu k vlastní funkci `?agnes`. Nejjednodušší je přiřadit argumentu `par.method` pouze jednu hodnotu, tzv. alpha, přičemž platí že beta = 1-2*alpha. Pokud vím, jakou hodnotu beta chci nastavit, hodnotu alpha pak vypočtu jako alpha = (1-beta)/2. Pokud tedy chci spočíst beta flexible s beta = -0.25 (která optimálně reprezentuje reálné distance mezi vzorky), pak vypočtu alpha = (1-(-0.25))-2 = 1.25/2 = 0.625. V praxi to bude vypadat takto:

```
library (cluster)
cluster.flexible <- agnes (x = dis, method = 'flexible', par.method = 0.625)
```

Pokud chci dendrogram nakreslit, použiji funkci `plot`. V případě objektů z knihovny `cluster` však funkce `plot` funguje trochu jinak než u objektů vytvořených funkcí `hclust`: kreslí obrázků hned několik, přičemž před nakreslením dalšího obrázku čeká na potvrzení, které provedete kliknutím myší do obrázku. Více podrobností k tomu získáte v nápovědě (v případě kreslení objektu, který je výsledkem funkce `agnes`, stačí napsat `?plot.agnes`). Pokud se tomu chcete vyhnout, stačí převést objekt na třídu `hclust` a pak jej nakreslit. To ostatně budete muset udělat v případě, že byste chtěli použít další kreslící funkce, jako např. `rect.hclust`, které v obrázcích vytvořených funkcí `plot.agnes` nefungují (nebo fungují špatně):

```
cluster.flexible.hclust <- as.hclust (cluster.flexible)
plot (cluster.flexible.hclust)
```



Nehierarchická klasifikace

kmeans

Funkce pro nehierarchickou klasifikaci pomocí metody k průměrů. Nehierarchické metody jsou pomíjené, i když ekologicky dávají zajímavé výsledky (to u až ale tak bejvá). Do funkce vstupuje distanční matice a počet klastrů, na který chceme soubor rozdělit.

```
cluster.kmeans <- kmeans (dist, centers = 5)
cluster.kmeans$cluster
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54
```

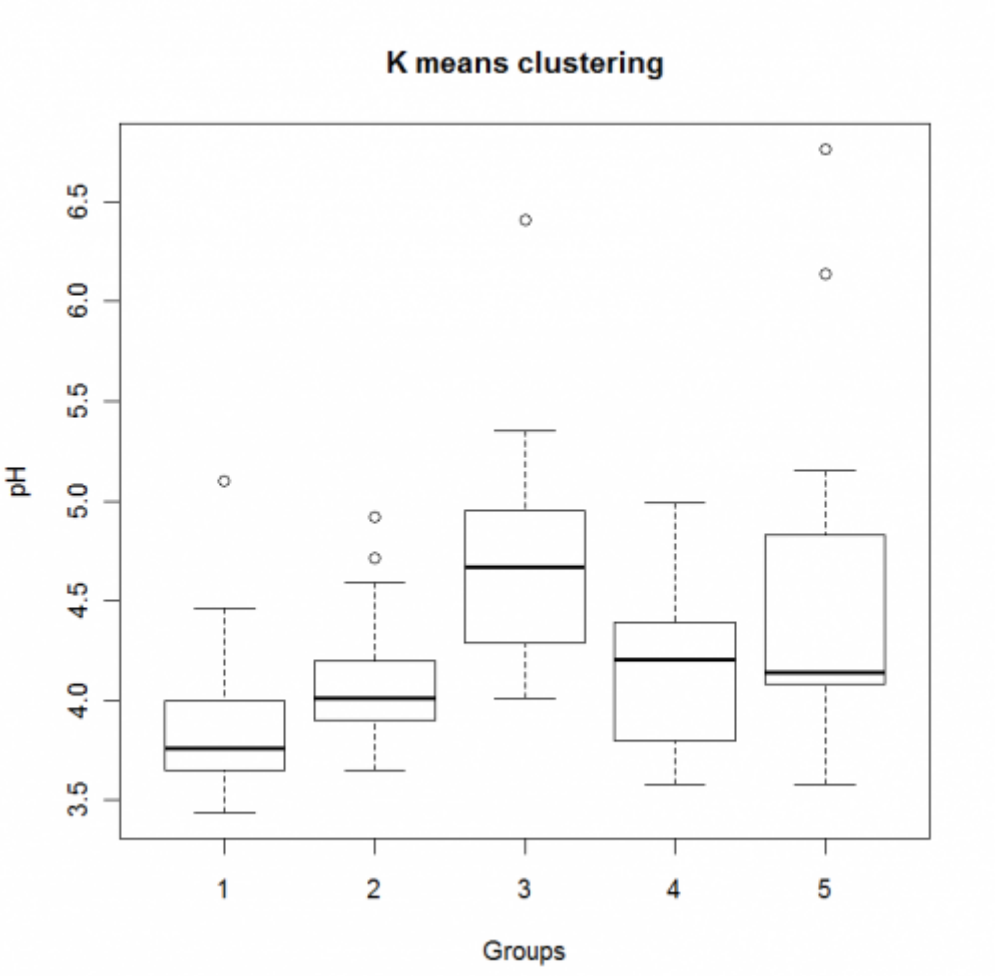
```
 1  1  5  1  3  3  5  4  1  1  3  1  5  5  3  4  4  4  1  5  1  3  4  1  1
2  4  4  4  3  3  5  5  5  5  5  1  1  1  1  1  1  4  3  1  1  1  5  3  4  4
4  4  2
55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97
 3  2  2  2  2  2  2  2  2  3  2  2  2  2  2  4  4  4  4  4  2  1  1  3  2
2  2  3  2  2  1  3  1  1  1  3  4  4  4  3  4  5  5
```

Další možnosti zobrazení

boxplot

Klasifikaci většinou používáme při ekologické interpretaci vzniklých klastrů pomocí externích proměnných prostředí. K tomu může sloužit třeba boxplot:

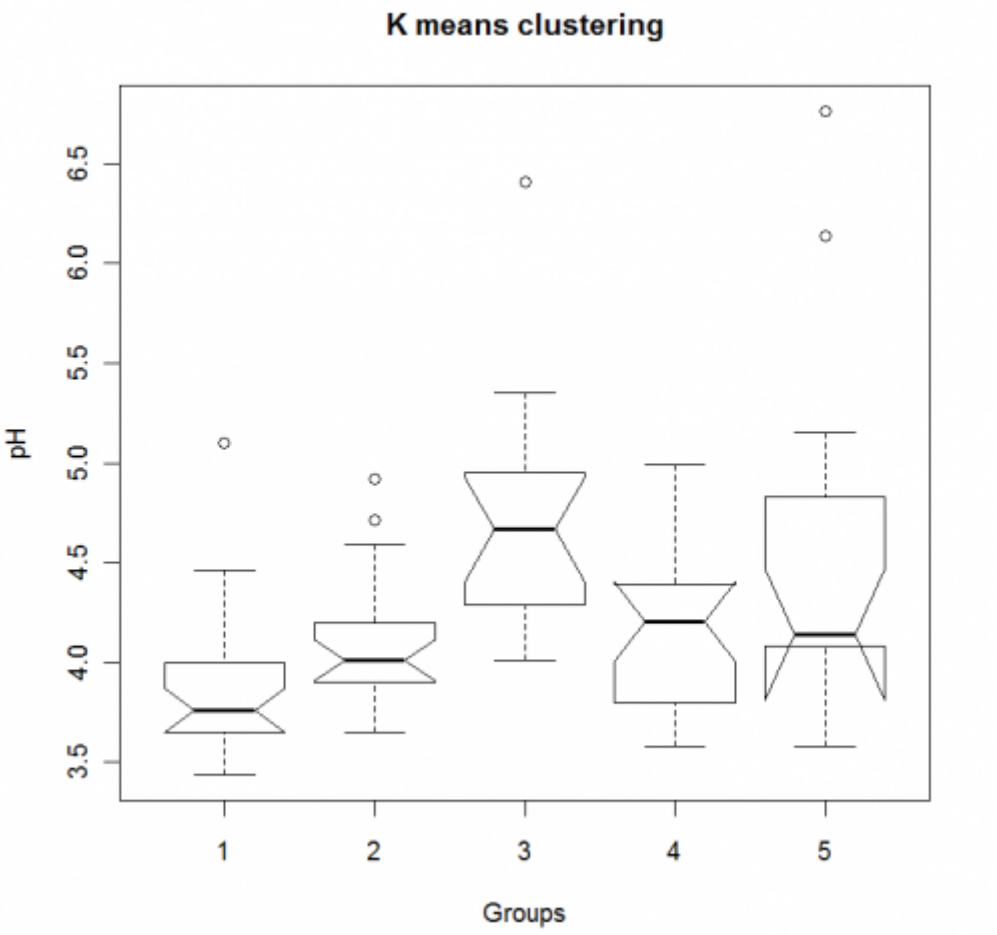
```
env.data <- vltava.env$pH.H  # obsahuje pH půdy jednotlivých ploch (měřené ve vodním roztoku, proto to H)
boxplot (env.data ~ cluster.kmeans$cluster, main = 'K means clustering', xlab = 'Groups', ylab = 'pH')
```



Ten samý boxplot můžeme ještě “vyšperkovat” přidáním tzv. zářezů - pokud se dvě krabice svými

zářezy nepřekrývají, indikuje to signifikantní rozdíl mezi jejich průměry.

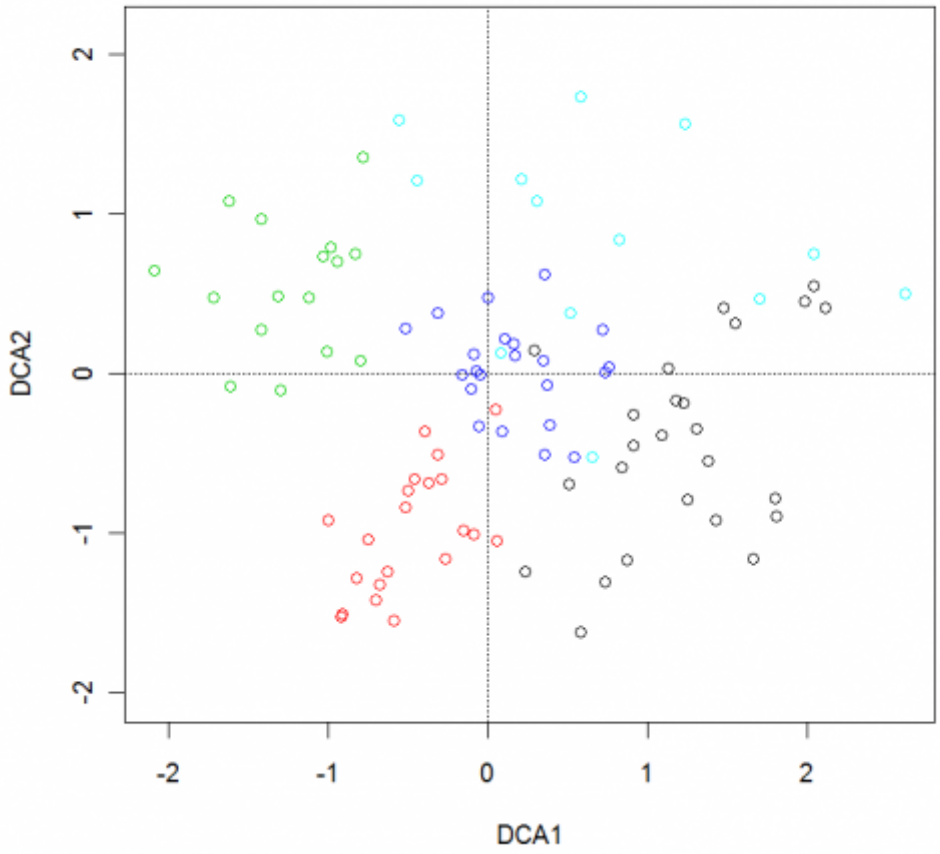
```
boxplot (env.data ~ cluster.kmeans$cluster, main = 'K means clustering',
xlab = 'Groups', ylab = 'pH', notch = T)
```



ordiplot

Nakreslí ordinační diagram, ve kterém se dají různým způsobem zviditelnit jednotlivé klastry vzorků. Na tohle dojde v budoucnu, zde jen skript bez vyzvětlení:

```
library (vegan) # pokud jste ještě nenačetli
dca <- decorana (sqrt (veg.data)) # vypočítý DCA
ordiplot (dca, display = 'sites', type = 'n') # nakreslí prázdný ordinační
diagram, ve kterém jsou už naškálované ordinační osy
points (dca, col = cluster.kmeans$cluster)
```



Cvičení 1

1. naimportujte data z třebíčských trávníků (grassland-spe.csv a grassland-env.csv)
2. vypočtěte distanční matici pomocí euklidovské vzdálenosti
3. naklastrujte data pomocí Wardovy metody
4. nakreslete dendrogram a krabice kolem 5 klastrů
5. nakreslete boxplot rozdílů v pH půdy (proměnná pH.H v souboru grassland-env.csv) mezi 5 klastry

Nápověda k funkcím:

1. `read.delim` nebo `read.csv2` nebo `read.table`
2. `vegdist` nebo `dist`
3. `hclust`
4. `plot` a `rect.hclust`
5. `cutree` a `boxplot`

Řešení

Cvičení 2

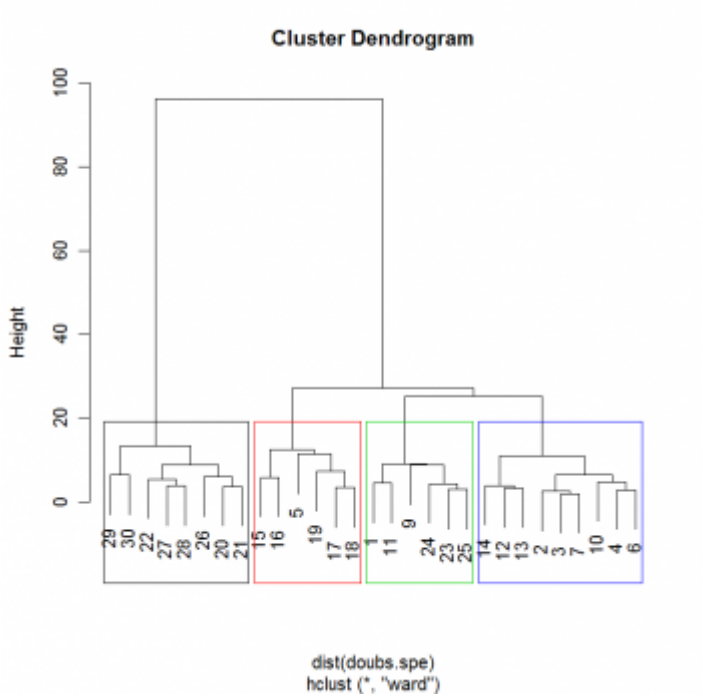
Použijte data o rybích společenstvech v řece Doubs.

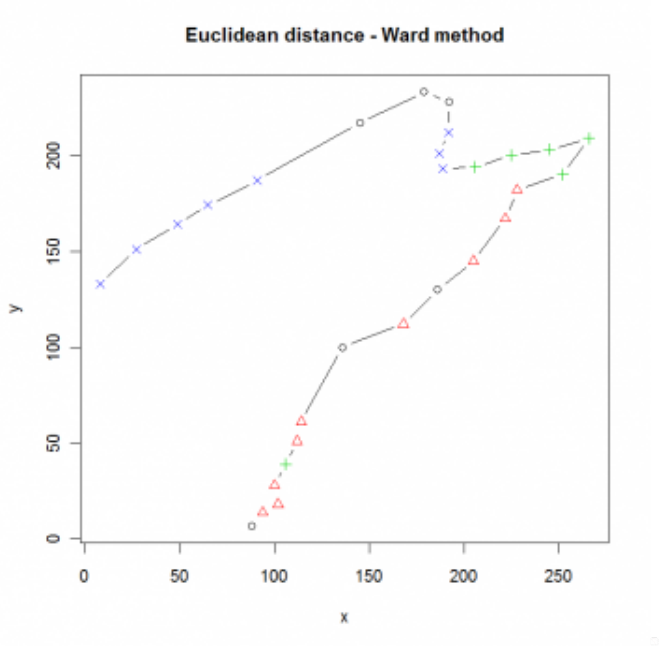
1. Načtěte data do eRka (druhovou matici `DoubsSpe.txt` a matici s geografickými koordinátami `DoubsSpa.txt`, která obsahuje pozici podél x a y geografické osy). Matice v eRku pojmenujte `doubs.spe` a `doubs.spa`.
2. Z obou matic odstraňte prázdný vzorek číslo 8, který neobsahuje žádný druh.
3. Vypočtěte hierarchickou klastrovou analýzu - použijte euklidovskou distanci a Wardovu metodu shlukování.
4. Nakreslete výsledný dendrogram, do kterého dokreslete obdélníky kolem 4 skupin, a to tak, aby každý obdélník měl jinou barvu.
5. Vypočtěte nehierarchickou klastrovou analýzu druhových dat - použijte Bray-Curtis míru nepodobnosti a K-means metodu shlukování pro vytvoření 4 shluků.
6. Nakreslete obrázek prostorového rozmístění jednotlivých vzorků (použijte k tomu datový soubor se souřadnicemi), do kterého zobrazte výsledky hierarchické klasifikace (euklid + Ward) tak, že vzorky z různých klastrů se od sebe budou lišit barvou i typem symbolu.
7. Stejný obrázek nakreslete i pro výsledky nehierarchické klasifikace (Bray-Curtis + K-means).

Nápověda:

1. read.delim
2. `doubs.spe[-8,]`
3. `hclust`, `dist`
4. `plot`, `rect.hclust` (argument `border`, který ovlivňuje barvu obdélníku, musí být vektor čtyř čísel nebo názvů barev)
5. `kmeans`, `vegdist`
6. `cutree`, `plot` s argumenty x a y odpovídajícími souřadnicím v datovém rámci `doubs.spa`. Typ symbolu se mění argumentem `pch`, barva argumentem `col`. Výsledek funkce `cutree` (vektor přirazení jednotlivých vzorků do klastrů) může být použit pro argumenty `pch` a `col`. Argument `type = "b"` ve funkci `plot` způsobí, že jednotlivé body budou spojeny linií.
7. na výsledek funkce `k.means` není možné použít `cutree` - v tomto případě je přiřazení vzorků do klastrů vytvořených `k.means` uloženo ve výsledku této funkce, v proměnné `cluster`.

Řešení





Cvičení 3

Použijte statistická data o evropských zemích (soubor europe.txt[2]), které pocházejí z webových stránek CIA [3].

1. Data naimportujte do eRka (názvy států uložte jako `row.names`) a seznamte se s nimi - podívejte se na význam jednotlivých proměnných. Otázkou je, jestli existují skupiny států s podobnými parametry (ekonomiky, rozlohy, demografie), a které státy do které skupiny patří.
2. Použijte hierarchickou klastrovou analýzu, všespojnou metodu shlukování (“complete linkage”) na matici euklidovských distancí spočtenou na evropských datech. Pozor - původní data je třeba standardizovat po sloupcích, aby byly jednotlivé proměnné na stejné škále!
3. Nakreslete dendrogram s výsledky, a skupiny států rozdělte do rozumného počtu klastrů (např. 4-5), které označte v dendrogramu obdélníky.
4. Na stejných (standardizovaných) datech vypočtěte také PCA, nakreslete ordinační diagram a kolem států ve stejném klastru nakreslete obálky (pomocí funkce `ordihull`).

Řešení

1)

V eRku je tato funkce pod názvem `flexible` zahrnuta ve funkci `agnes` v balíku `cluster`, ale s koeficientem *beta* to zde není tak jednoduché

2)

pozor na to, jakými znaky jsou od sebe odděleny jednotlivé buňky v tabulce!

3)

data jsou převzata z této webové stránky