KMI/ZZD – získávání znalostí z dat

Sumarizace a vizualizace dat
demonstrace v R

Jan Konečný

22. února 2012

Dataset Algae

Na tomto datasetu si vyzkoušíme předzpracování dat, a explorační datovou analýzu (později se k tomuto datasetu ještě vrátíme).

Z hlediska standardů DM jde o velmi malý problém:
Předvídání frekvence výskytu několika škodlivých řas ve vzorcích vody.

Popis problému a cíle

- vzorky vody byly sbírány v různých evropských řekách v různých časech během zhruba jednoho roku.
- v každém vzorku byly měřeny různé chemické vlastnosti, a frekvence výskytu sedmi škodlivých řas.
- Navíc je každý vzorek charakterizován podmínkami sběru: roční období, velikost řeky, rychlost řeky.

Hlavní motivace: chemické monitorování je levné a snadno automatizovatelné. Naopak biologická analýza vzorku zahrnuje mikroskopické vyšetření, je k ní potřeba trénovaná lidská síla a jde o pomalý a náročný proces. Získání modelu, který by na založený na chemických vlastnostech, by napomohlo vytvořit levný a automatický systém pro monitorování výskytu škodlivých řas.

Dalším cílem je poskytnout lepší pochopení faktorů, které ovlivňují výskyt těchto řas.

Popis dat

Máme dataset, ve kterém je 200 pozorování.

Každé pozorování obsahuje informaci o 11 proměnných:
Tři z nich jsou kategorické: roční období, kdy byl vzorek odebrán, velikost řeky a rychlost řeky.

Zbylých 8 proměnných jsou hodnoty různých chemických parametrů:

- maximální hodnota pH
- minimální hodnota O_2
- průměrná hodnota Cl
- průměrná hodnota NO_3^-
- průměrná hodnota NH_4^+
- průměr PO_4^{3-}
- průměr všech PO_4
- průměr chlorofylu

S každým pozorováním je spojeno 7 čísel reprezentujících frekvenci výskytu 7mi druhů řas.

Načtení datasetu Algae do R

Snadná cesta: je zabudovaný v balíku DMwR

```
> library(DMwR)
```

Nesnadná cesta: Stáhněte si dataset v txt:

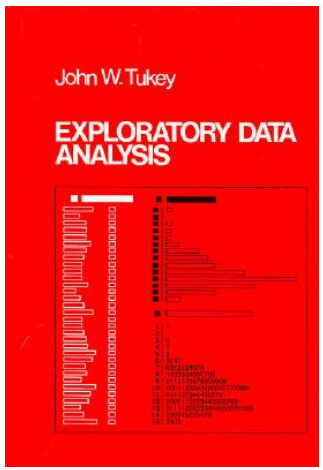
`http://www.liaad.up.pt/~ltorgo/DataMiningWithR/DataSets/Analysis.txt`

a načtěte ho:

```
> algae <- read.table('Analysis.txt',
+                     header=F, dec='.',
+                     col.names=c('season','size','speed',
+                                 'mxPH','mnO2','Cl','NO3',
+                                 'NH4','oPO4','PO4','Chla',
+                                 'a1','a2','a3','a4',
+                                 'a5','a6','a7'),
+                     na.strings=c('XXXXXXX'))
```

Sumarizace a vizualizace dat

Explorační analýza dat – je přístup k analyzování datasetů, který využívá sumarizaci hlavních charakteristik v snadno pochopitelné podobě, často s vizuálními grafy.



Tukey J. W.,
Exploratory Data Analysis,
Addison-Wesley, 1977

ftk

Sumarizace dat

pro vektor čísel

```
> v1 <- rnorm(10)
> summary(v1)
    Min.  1st Qu.   Median     Mean 3rd Qu.     Max.
-1.9760  -1.0030  -0.2004  -0.3350   0.3999   0.7872
```

pro vektor faktorů

```
> v2 <- sample(factor(c("small","medium","large")),
+              10, replace=T)
> summary(v2)
 large  medium   small
     6       1       3
```

Je-li funkce aplikována na datový rámec, je aplikována na každý jeho sloupec:

```
> summary(algae)
```

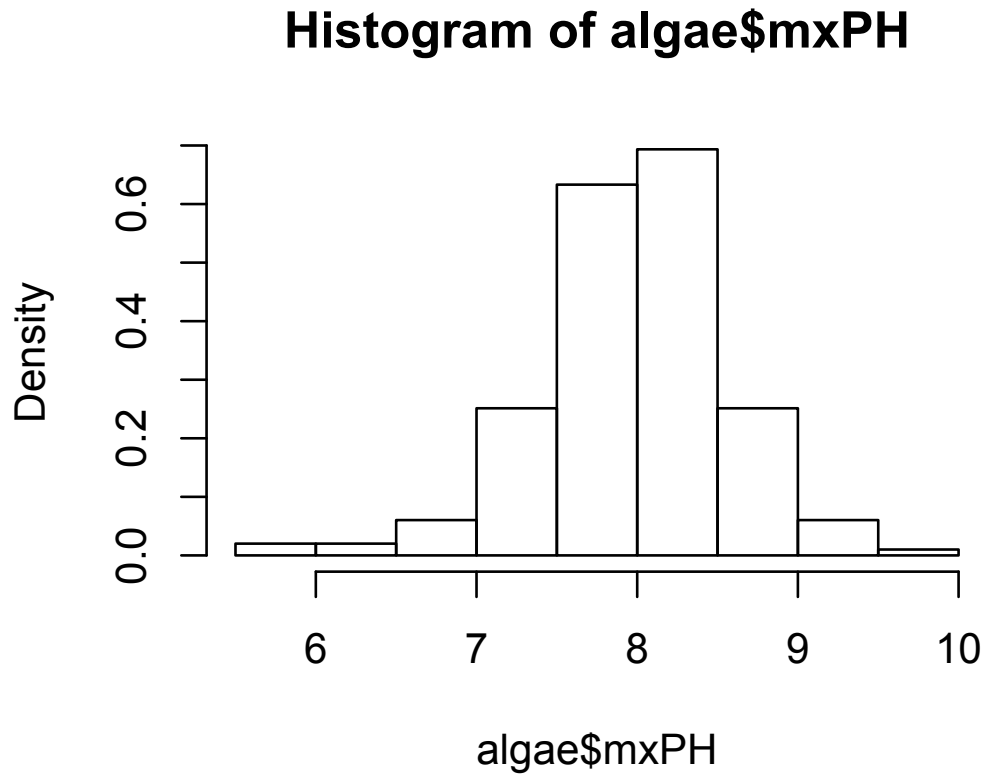
Viz také

```
> library(Hmisc)
> describe(algae)
```

. . . obrázek obvykle řekne víc:
Histogram číselného vektoru.

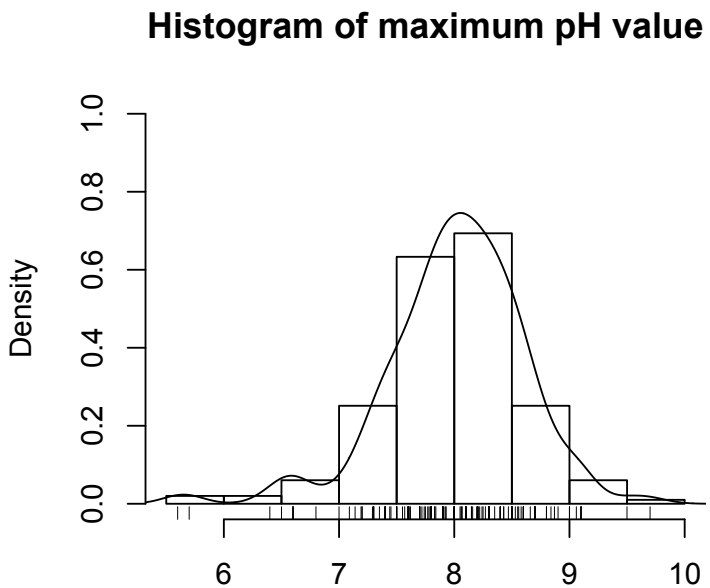
```
> hist(algae$mxPH, prob = T)
```

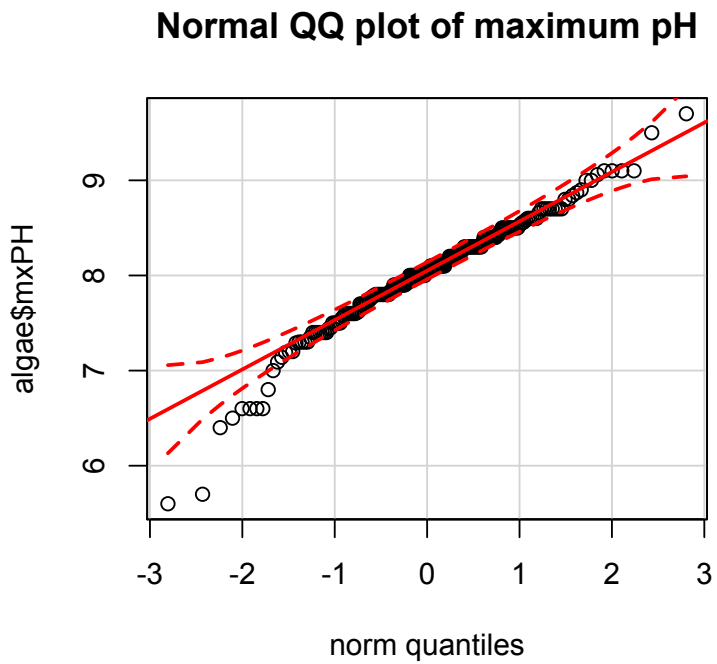
Parametr prob = T znamená, že pro každý interval dostaneme pravděpodobnost; s defaultním prob = F bychom dostali četnosti.



Hodnoty proměnné mxPH se zjevně v normálním rozložení s hodnotami uspořádanými okolo průměru.

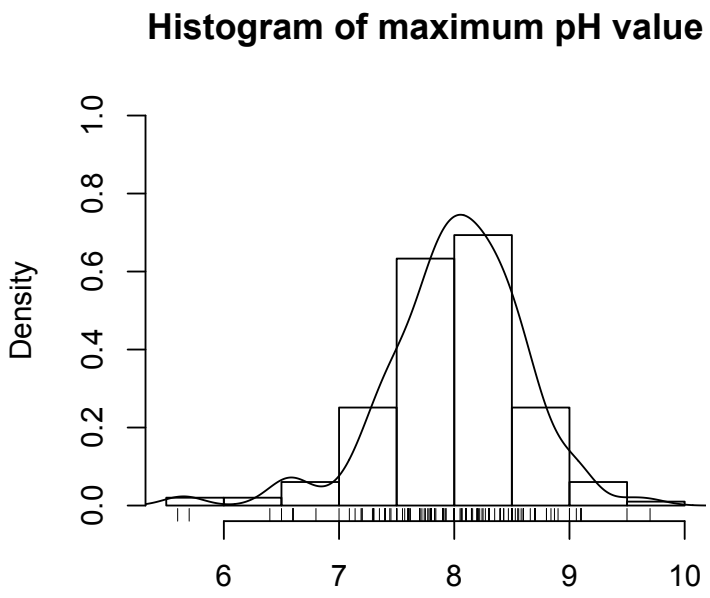
```
> library(car)
> par(mfrow=c(1,2))
> hist(algae$mxPH, prob=T, xlab='',
+    main='Histogram of maximum pH value',ylim=0:1)
> lines(density(algae$mxPH,na.rm=T))
> rug(jitter(algae$mxPH))
> qqPlot(algae$mxPH,
+           main='Normal QQ plot of maximum pH')
> par(mfrow=c(1,1))
```

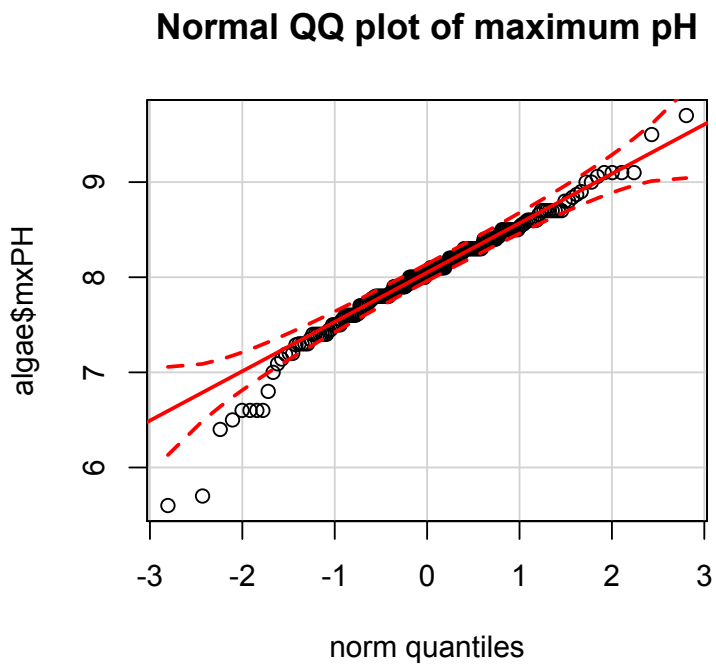




```
1  > library(car)
2  > par(mfrow=c(1,2))
3  > hist(algae$mxPH, prob=T, xlab='',
4  +    main='Histogram of maximum pH value',ylim=0:1)
5  > lines(density(algae$mxPH,na.rm=T))
6  > rug(jitter(algae$mxPH))
7  > qqPlot(algae$mxPH,
8  +           main='Normal QQ plot of maximum pH')
9  > par(mfrow=c(1,1))
```

1 – nahrajeme balík car.
2 – par() slouží k nastavení parametrů grafiky R; v tomto případě říkáme, že výstupní okno má mít 1 řádek a dva sloupce.
3,4 – kreslíme první graf: histogram proměnné mxPH; tentokrát říkáme, že popisek u x má být prázdný, nastavujeme název grafu a určujeme limity pro y.
5 – přidáme hladkou verzi histogramu (odhad hustoty)
6 – přidá reálné hodnoty na osu x
7,8 – QQ-plot, porovnání hodnot vzhledem k teoretickým kvantilům normálního rozdělení.
9 – nastavení parametru grafiky na původní hodnotu.

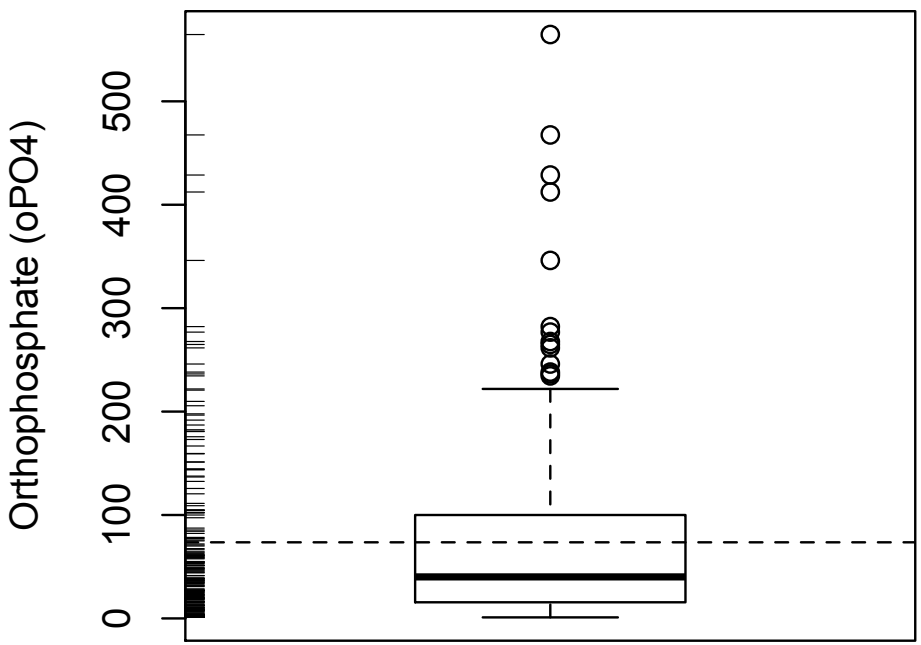




Funkce qqPlot kreslí i 95% interval konfidence (červené čárkované hranice).
Jak můžeme vidět, několik nízkých hodnot nám narušuje předpoklad normálního rozložení.

Jiný příklad ukazující tento způsob vyšetřování dat
(tentokrát pro proměnnou oPO4)

```
1 > boxplot(algae$oPO4,
2 +         ylab = "Orthophosphate (oPO4)")
3 > rug(jitter(algae$oPO4), side = 2)
4 > abline(h = mean(algae$oPO4, na.rm = T), lty = 2)
```

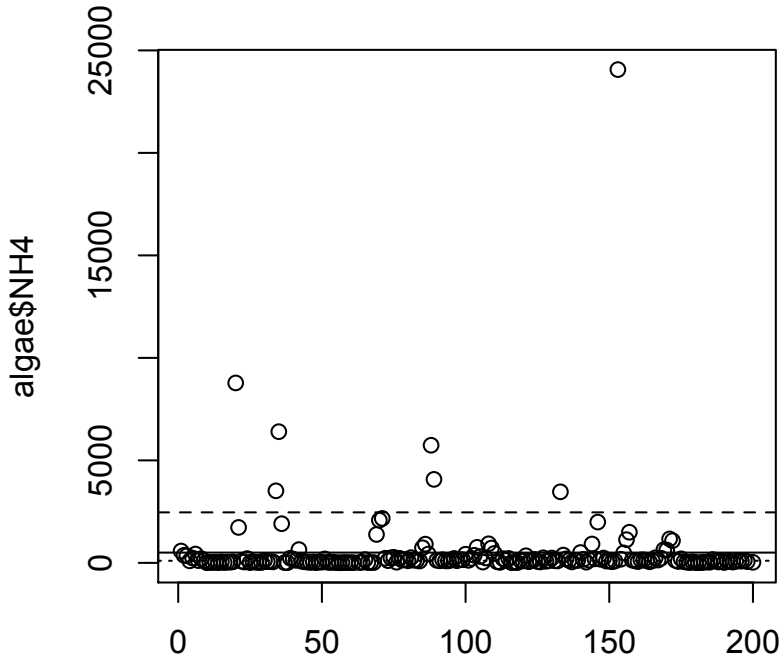


1,2 – vykreslí box plot
2 – vykreslí reálné hodnoty u osy y
3 – vykreslí čárkovaně průměr.

box představuje medián a první a třetí kvartil (Q_1 a Q_3).
IQR je vzdálenost mezi kvartily $Q_3 - Q_1$,
anténky označují interval $(Q_1 - 1.5 \cdot IDQ, Q_1 + 1.5 \cdot IDQ)$,
přesněji, vyznačují krajní hodnoty, které jsou pořád v tomto intervalu.

Body, které jsou mimo 'anténky' jsou považovány za odlehlé body. Někdy můžeme chtít prošetřit tyto „divné“ hodnoty. Ukážeme si to pro hodnoty proměnné NH4:

```
> plot(algae$NH4, xlab = "")
> abline(h = mean(algae$NH4, na.rm = T), lty = 1)
> abline(h = mean(algae$NH4, na.rm = T) +
+                 sd(algae$NH4, na.rm = T), lty = 2)
> abline(h = median(algae$NH4, na.rm = T), lty = 3)
> identify(algae$NH4)
[1]  20 153
```



identify () umožňuje identifikovat (klikáním) body v grafu (ukončuje se kliknutím pravého tlačítka). Výsledkem je vektor čísel řádků.

Pokud se chceme podívat přímo na ta data:

```
> plot(algae$NH4, xlab = "")
> clicked.lines <- identify(algae$NH4)
> algae[clicked.lines, ]
```

Samozřejmě to můžeme udělat i bez grafiky

```
> algae[algae$NH4 > 19000, ]
```

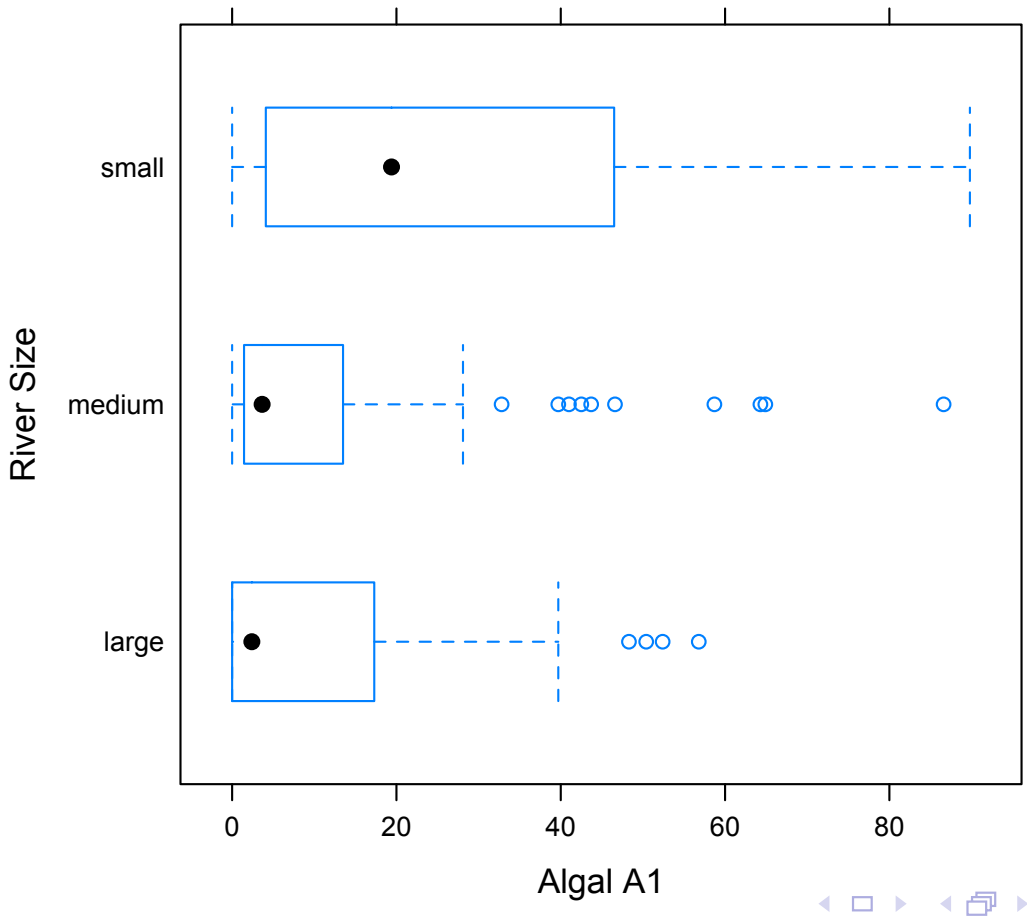
Možná je podezřelé, že ve výsledku dostaneme i řádky, které mají v NH4 hodnotu NA.

To můžeme napravit:

```
> algae[!is.na(algae$NH4) & algae$NH4 > 19000,]
```

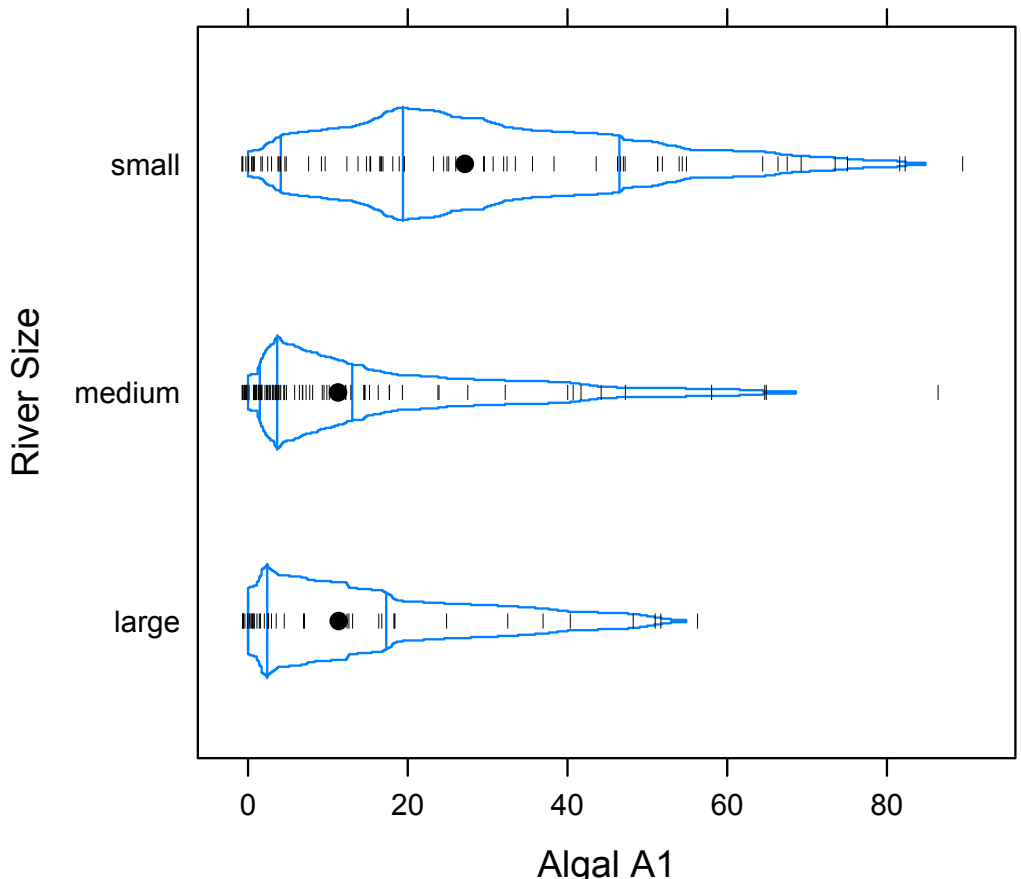
Pár příkladů jiného typu inspekce dat.
Tentokrát nás bude zajímat závislost dvou proměnných – řekněme *a1* v závislosti na *size*. Budeme potřebovat balík lattice

```
> library(lattice)
> bwplot(size ~ a1, data=algae,
+        ylab='River Size',xlab='Algal A1')
```



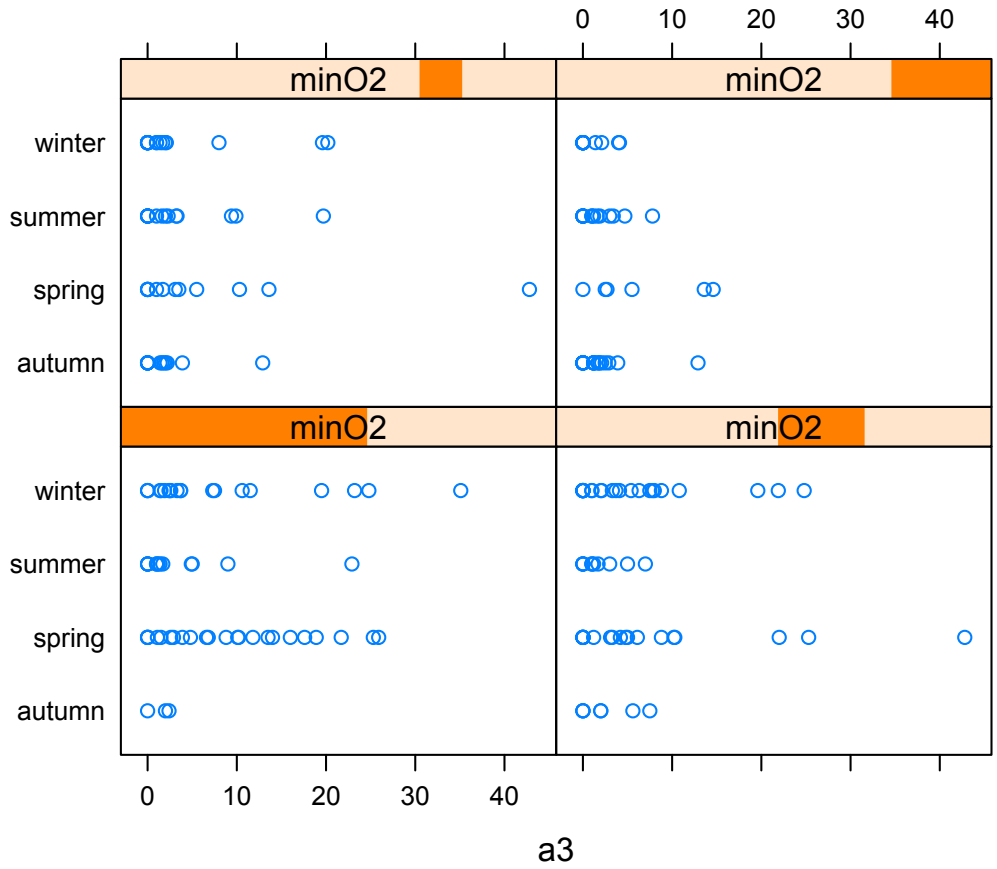
Zajímavou alternativu, která nám dává více informací o rozložení proměnné, poskytuje balík Hmisc:

```
> library(Hmisc)
> bwplot(size ~ a1, data=algae,panel=panel.bpplot,
+        probs=seq(.01,.49,by=.01), datadensity=TRUE,
+        ylab='River Size',xlab='Algal A1')
```

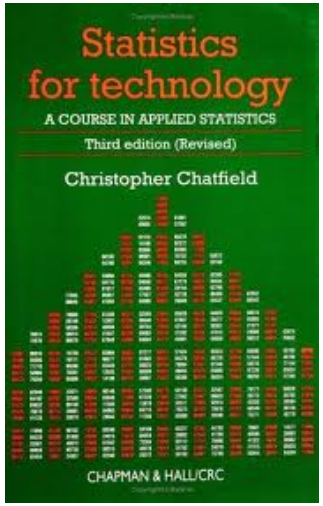


Další možnosti zobrazení

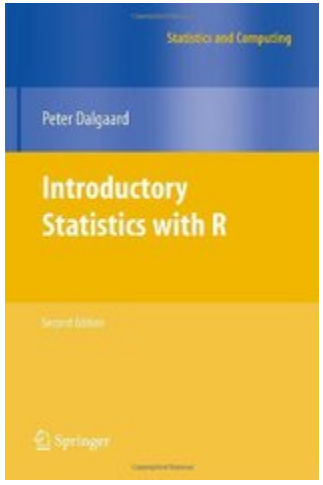
```
> minO2 <- equal.count(na.omit(algae$mnO2),
+                      number=4,overlap=1/5)
> stripplot(season ~ a3|minO2,
+               data=algae[!is.na(algae$mnO2),])
```



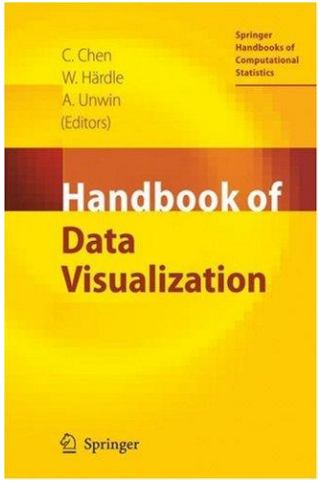
Chcete vědět víc?



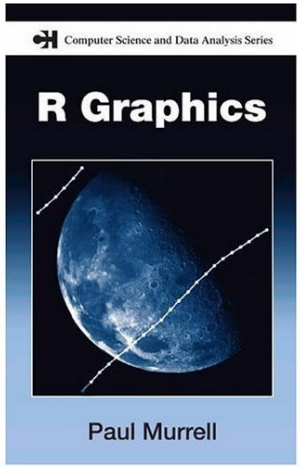
Chatfield C.,
Statistics for Technology,
Springer, 1983



Dalgaard P.,
Introductory Statistics with R,
Springer, 2002



Chen C., Härdle W., Unwin A. (eds.) ,
Handbook of Data Visualization,
Chapman & Hall/CRC, 2006



Murrell R.,
R Graphics,
Chapman & Hall/CRC, 2006

?h , ?

`http://www.stat.auckland.ac.nz/~paul/RGraphics/rgraphics.html`

Neznámé hodnoty

Viděli jsme, že máme v datech chybějící/neznámé hodnoty
Nejčastější řešení:

- odstranit je,
- doplnit je prozkoumáním korelací mezi proměnnými,
- doplnit je prozkoumáním podobností mezi objekty,
- používat metody, které s se nimi umí vypořádat.

Tak se na to podíváme v R.

Odstranění objektů s neznámými hodnotami

Výhoda: je to jednoduché;
Nevýhoda: ztrácíme data; navíc v některých datasetech jsou chybějící hodnoty naprosto běžné (třeba v medicíně).

Předtím, než řádky obsahující neznámé hodnoty odstraníme, je dobré se na ně podívat, nebo je aspoň spočítat

```
> algae[!complete.cases(algae),]
...
...

> nrow(algae[!complete.cases(algae),])
[1] 16
```

Funkce complete.cases() vyprodukuje logický vektor s tolika elementy, kolik je řádků. *n*-tý element je TRUE, pokud *n*-tý řádek neobsahuje NA.

na odstranění těch 16ti řádek stačí

```
> algae <- algae[complete.cases(algae),]
```

nebo

```
> algae <- omit.na(algae)
```

Obnovení původního datasetu:

```
> data(algae)
```

Obvyklejší je mazat jen řádky, které mají příliš velké množství neznámých hodnot, např řádky 62 a 199 mají 6 z 11 proměnných s neznámými hodnotami.
To z nich dělá nepoužitelnou záležitost – doplňování těchto hodnot by nebylo příliš spolehlivé.

```
> algae <- algae[-c(62, 199), ]
```

U velkých datasetů bychom stěží toto prováděli vizuální inspekcí ...

```
> apply(algae, 1, function(x) sum(is.na(x)))
```

Apply je mocný nástroj (funkce vyššího řádu),
doporučuji prozkoumat ?apply.

Balík DMwR poskytuje funkci manyNAs

```
> manyNAs(algae, 0.2)
[1] 62 199

> algae <- algae[-manyNAs(algae), ]
```

Doplnění centrální hodnotou

Alternativa k eliminaci řádků, která je ovšem obdobně hloupá.
Jako centrální hodnota může posloužit průměr, medián, nebo modus.
Doplnění průměru

```
> algae[48, "mxPH"] <- mean(algae$mxPH, na.rm = T)
```

Doplnění mediánu

```
> algae[is.na(algae$Chla), "Chla"] <-
+   median(algae$Chla, na.rm = T)
```

Nalezení modu (nejčastější hodnoty)

```
> x <-table(algae$season)
> x

autumn spring summer winter
    40     53     44     61
> names(x)
[1] "autumn" "spring" "summer" "winter"
> names(x)[max(x)==x]
[1] "winter"
```

Doplnění zkoumáním korelací

Korelační matice

```
> cor(algae[, 4:18], use = "complete.obs")
```

Funkce symnum pro kompaktnější zobrazení

```
> symnum(cor(algae[,4:18],use="complete.obs"))
      mP mO Cl NO NH o P Ch a1 a2 a3 a4 a5 a6 a7
mxPH  1
mnO2     1
Cl          1
NO3            1
NH4            ,  1
oPO4     .  .        1
PO4      .  .        * 1
Chla  .                  1
a1          .        .  .    1
.....................................................
a7                                                1
attr(,"legend")
[1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

```
> algae <- algae[-manyNAs(algae), ]
> lm(PO4 ~ oPO4, data = algae)

Call:
lm(formula = PO4 ~ oPO4, data = algae)

Coefficients:
(Intercept)         oPO4
     42.897        1.293
```

Funkce lm() může být použita ke získání lineárních modelů ve tvaru

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n$$

(vrátíme se k ní později v tomto kurzu)

Teď nám říká, že

$$PO4 \approx 42.897 + 1.293 \cdot oPO4$$

Můžeme tedy doplnit hodnotu:

```
> algae[28, "PO4"] <- 42.897 +
+                     1.293 * algae[28, "oPO4"]
```

V tomto příkladě PO4 hodnota chyběla jen jedna.

Pro rozsáhlejší dataset, s více NA hodnotami v této proměnné můžeme udělat následující:

```
> fillPO4 <- function(oP) {
+     if (is.na(oP))
+       return(NA)
+     else return (42.897 + 1.293 * oP)
+ }

> algae[is.na(algae$PO4), "PO4"] <-
+     sapply(algae[is.na(algae$PO4), "oPO4"],
             fillPO4)
```

Doplnění na základě podobnosti

Z přednášky známe k-nearest neighbor

Jako DU naprogramujte sami funkci, která doplní sloupec touto metodou. Vzdálenost měřte jako

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{p} \delta_i(x_i, y_i)},$$

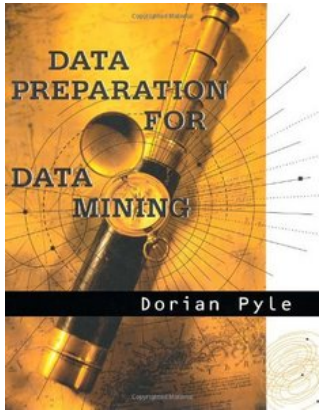
kde

$$\delta_i(v_1, v_2) = \begin{cases} 1 & \text{pokud } i \text{ je nominální a } v_1 \neq v_2. \\ 0 & \text{pokud } i \text{ je nominální a } v_1 = v_2. \\ (v_1 - v_2)^2 & \text{pokud } i \text{ je numerický.} \end{cases}$$

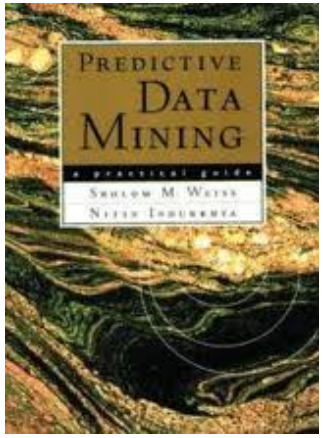
Tyto vzdálenosti jsou počítány až po normalizaci, která je:

$$y_i = \frac{x_i - \hat{x}}{\sigma_x}$$

Chcete vědět víc?



Pyle D.,
Data Preparation for Data Mining,
Morgan Kaufmann, 1999



Weiss S. M., Indurkhya N.,
Predictive Data Mining,
Morgan Kaufmann, 1999