

## 10 Věrohodnost – Normální rozdělení

### Dataset 6: 03-paired-means-clavicle2.txt

- Datový soubor obsahuje osteometrické údaje o délkách klíčních kostí (*clavicula*). Data pochází z anglického souboru dokumentovaných skeletů (Parsons, 1916). V souboru se nachází délky klíčních kostí na pravé a levé straně těla v párovém uspořádání. Jednotlivé kosti bez druhostranné kosti nebyly do souboru zařazeny.
- Přehled proměnných v datasetu:
  - id ... ID jedince;
  - sex ... pohlaví jedince (*m* - muž, *f* - žena);
  - length.L ... délka klíční kosti z levé strany (v mm);
  - length.R ... délka klíční kosti z pravé strany (v mm).

### Příklad 10.1. Maximálně věrohodné odhady parametrů $\mu$ a $\sigma^2$ v normálním modelu

Načtěte datový soubor 03-paired-means-clavicle2.txt. Nechtě náhodná proměnná  $X$  popisuje délku klíční kosti z levé strany u mužů. Za předpokladu, že náhodná veličina  $X$  pochází z normálního rozdělení, tj.  $X \sim N(\mu, \sigma^2)$ :

1. Odvod'te
  - a. tvar jádra věrohodnostní funkce  $L((\mu, \sigma^2)^T | x)$  normálního modelu;
  - b. tvar jádra logaritmu věrohodnostní funkce  $\ell((\mu, \sigma^2)^T | x)$  normálního modelu;
  - c. skóre funkci pro parametr  $\mu$ ;
  - d. skóre funkci pro parametr  $\sigma^2$ ;
  - e. tvar Fisherovy informační matice.
2. Dosazením do vzorců stanovte přesnou hodnotu maximálně věrohodného odhadu parametrů  $\mu$  a  $\sigma^2$ , tj.  $\hat{\mu}$  a  $\hat{\sigma}^2$ .
3. Pomocí maximalizace logaritmu věrohodnostní funkce  $\ell((\mu, \sigma^2)^T | x)$  normálního modelu nalezněte maximálně věrohodný odhad parametrů  $\mu$  a  $\sigma^2$ . Maximalizaci proved'te
  - a. pomocí funkce `optim()`;
  - b. pomocí vlastnoručně naprogramované dvourozměrné Newton-Raphsonovy metody `NRnorm()`;
  - c. pomocí vlastnoručně naprogramované Broydenovy metody `BMnorm()`.
4. Vykreslete (a) vrstevnicový diagram; (b) 3D-diagram logaritmu dvourozměrné věrohodnostní funkce normálního modelu spolu s maximálně věrohodnými odhady parametrů  $\mu$  a  $\sigma^2$  odhadnutými pomocí všech tří funkcí. K vykreslení použijte (a) funkce `image()` a `contour()`, (b) funkci `persp()`.

### Řešení příkladu 10.1

```
1 data <- read.delim(...) # nacteni dat
2 cla.L <- data[... , ...] # vyber delek leve kl. kosti muzu
3 data <- na.omit(...) # odstraneni NA hodnot
4 n <- ... # delka vektoru cla.L
5 mu <- ... # MLE odhad parametru mu
6 sigma2 <- ... # MLE odhad parametru sigma ^ 2
7 tab <- data.frame(...) # souhrnna tabulka vysledku
8
9 lnormo <- function(theta, x){ # log. veroh. fce rozd. N(mu, sigma ^ 2)
10   n <- ... # delka vektoru x
11   like <- ... # log. fce veroh. rozdeleni N(theta[1], theta[2])
12   return(...) # vystup; promenna like
13 }
14 OPTtheta <- optim(c(145, 100), lnormo, x = cla.L,
15                 control = list(fnscale = -1)) # MLE odhad vektoru
16   # parametru theta = (mu, sigma ^ 2) ^ T
17 max.m <- OPTtheta$par[1] # MLE odhad parametru mu
18 max.s <- OPTtheta$par[2] # MLE odhad parametru sigma ^ 2
```

```
mu sigma2
1 153.6 96.96
```

```
21 lnorm <- function(mu, sigma, x){ # log. veroh. fce rod. N(mu, sigma ^ 2)
22   n <- ... # delka vektoru x
23   like <- ... # logaritmus veroh. fce rozd. N(mu, sigma ^ 2)
24   return(...) # vystup; promenna like
25 }
26
27 snorm <- function(mu, sigma, x){ # vektor skore funkci U
28   n <- # delka vektoru x
29   skore.mu <- ... # skore fce pro par. mu
30   skore.sigma <- ... # skore fce pro par. sigma ^ 2
31   U <- c(..., ...) # vektor U = (S(mu), S(sigma ^ 2)) ^ T
32   return(...) # vystup; vektor U
33 }
34
35 hnorm <- function(mu, sigma, x){ # Fisherova inf. matice I
36   n <- ... # delka vektoru x
37   I11 <- ... # I_{11}
38   I12 <- ... # I_{12} = I_{21}
39   I22 <- ... # I_{22}
40   I <- matrix(...) # Fisherova informaci matice I
41   return(...) # vystup; matice I
42 }
43
44 NMnorm <- function(x01, x02, t){
45   # iniciacni faze funkce; x = (mu, sigma ^ 2) ^ T
46   x <- c(x01, x02) # poc. nastaveni vektoru x v kroku 1
47   f1 <- ... # poc. nastaveni f1 v kroku 1: f1 = 0
48   f2 <- lnorm(...) # poc. nastaveni f2 v kroku 1: f2 = l(x[1], x[2] | t)
49   k <- ... # poc. nastaveni pocitadla k: k = 0
50
51   # telo funkce; proces probiha, dokud podminka while(...) plati
52   while(abs(f1 - f2) > 0.00001 & k < 1000){
53     k <- ... # zvetseni pocitadla o 1
54     U <- snorm(..., t) # vektor skore fci U = (S(x[1]), S(x[2]))^T
55     H <- hnorm(..., t) # FIM v hodnotach mu = x[1] a sigma^2 = x[2]
56     x <- x + solve(H) %*% U # aktualizace vektoru x
57     f1 <- f2 # aktualizace f1
58     f2 <- lnorm(..., t) # aktualizace f2: log. veroh. fce v novem x
59   }
60   return(list(mu = x[1], sigma2 = x[2], like = f2,
61             k = k)) # vystup; seznam o 4 polozkach
62 }
63 NMres <- NMnorm(x01 = ..., x02 = ..., cla.L) # MLE odhad (mu, sigma^2)^T
64 # x01 a x02 je treba zvolit vhodne, napr. x01 = 150 a x02 = 90
65 max.nm <- NMres$... # MLE odhad par. mu
66 max.ns <- NMres$... # MLE odhad par. sigma ^ 2
```

```

67 BMnorm <- function(x10, x11, x20, x21, t){
68   # iniciacni faze funkce; x1 = (mu, sigma ^ 2)^T v kroku 1;
69   # x2 = (mu, sigma ^ 2)^T v kroku 2
70   x1 <- c(x10, x20) # poc. nastaveni vektoru x v kroku 1
71   x2 <- c(x11, x21) # poc. nastaveni vektoru x v kroku 2
72   f1 <- lnorm(x1[1], x1[2], t) # log. veroh. fce v kroku 1
73   f2 <- lnorm(...) # log. veroh. fce v kroku 2
74   k <- ... # poc. nastaveni pocitadla k: k = 0
75   B <- diag(2) # poc. nastaveni matice B (jednotkova m. dimenze = 2x2)
76
77 # telo funkce; proces probiha, dokud podminka while(...) plati
78 while(abs(f1 - f2) > 0.00001 & k < 1000){
79   k <- ... # zvetseni pocitadla o 1
80   U1 <- snorm(x1[1], x1[2], t) # vektor skore fci v kroku 1
81   U2 <- snorm(...) # vektor skore fci v kroku 2
82   B <- B + c((((...) - B %*% (...)) / sum(...) ^ 2)) %*% t(c(...))
83   # aktualizace matice B
84   x <- (x2 - solve(B) %*% U2) # vypocet vektoru x
85   f1 <- ... # aktualizace f1 (nyni f2)
86   f2 <- lnorm(...) # aktualizace f2 (log. veroh. fce ve vektoru x)
87   x1 <- ... # aktualizace x1 (nyni x2)
88   x2 <- ... # aktualizace x2 (nyni x)
89 }
90 return(list(mu = x[1], sigma2 = x[2], like = f2, k = k)) # vystup; seznam o 4 polozkach
91 }
92
93 BMres <- BMnorm(..., t = cla.L) # MLE odhad (mu, sigma ^ 2)^T
94 # x10, x11, x20 a x21 je treba zvolit vhodne, napr. 147, 157, 90 a 100.
95 max.bm <- BMres$mu # MLE odhad par. mu
96 max.bs <- BMres$sigma2 # MLE odhad par. sigma ^ 2

```

Tabulka 1: Odhady parametrů  $\mu$  a  $\sigma^2$  normálního rozdělení

	$\mu$	$\sigma^2$
exaktní výpočet	153.600000	96.960000
funkce optim()	153.601994	96.962806
Newton-Raphsonova metoda	153.600000	96.360928
Broydenova metoda	153.603912	96.461318

```

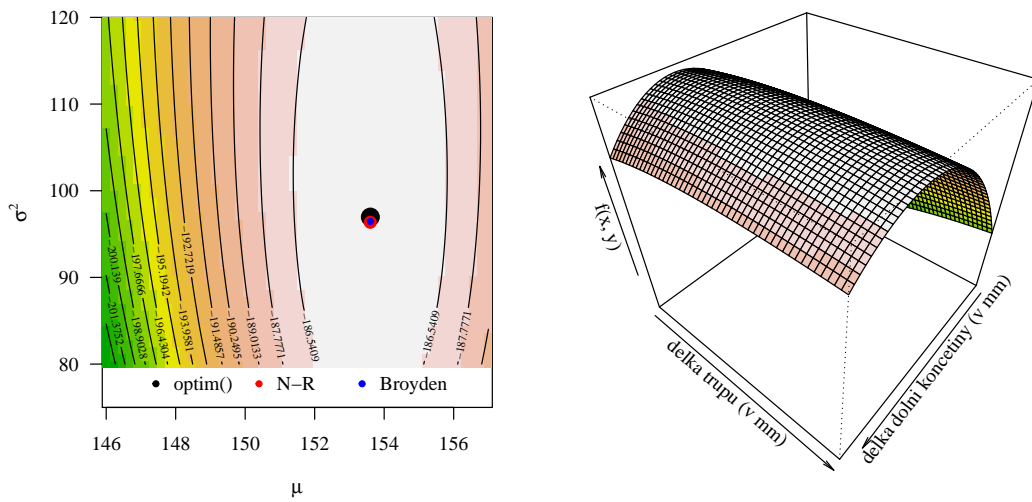
97 N <- 50
98 mu <- seq(...) # posl. bodu osy x; od 146 do 157 o delce N
99 sigma2 <- seq(...) # posl. bodu osy y; od 80 do 120 o delce N
100 M <- matrix(NA, N, N) # prazdna matice dimenze (N x N)
101 for(i in 1:length(mu)){
102   for(j in 1:length(sigma2)){
103     M[i, j] <- lnorm(mu[i], sigma2[j], cla.L)
104   }
105 } # hodnoty log. veroh. fce
106 k <- ... # pocet barev; k = 15
107 par(...) # okraje grafu 4, 4, 2, 2
108 image(..., asp = F, ylim = c(75, 120), col = terrain.colors(...),
109   xlab = expression(...), ylab = expression(...),
110   breaks = seq(min(M), max(M), length = k + 1),
111   las = ..., ) # vrstevnicovy diagram log. veroh. fce
112 contour(..., levels = ..., add = T) # konturovy diagram log. veroh. fce
113 points(...) # cerny bod; MLE odhad (mu, sigma ^ 2); fce optim
114 points(...) # cerveny bod MLE odhad (mu, sigma ^ 2); N-R metoda
115 points(...) # modry bod MLE odhad (mu, sigma ^ 2); Br. metoda
116 legend('bottom', horiz = T, ...) # legenda
117
118 nrz <- nrow(M) # pocet radku matice M
119 ncz <- ncol(M) # pocet sloupcu matice M
120 color <- terrain.colors(...) # paleta k barev
121 stredy <- (M[-1, -1] + M[-1, -ncz] + M[-nrz, -1] + M[-nrz, -ncz]) / 4
122 # matice stredy site
123 stredy.col <- cut(stredy, k) # rozdel rozpeti hodnot do k ekvidistantnich
124 # intervalu a kazde hodnote prirad interval, do ktereho naleza

```

```

125 par(...) # okraje grafu 1, 1, 1, 1
126 persp(..., col = color[stredy.col], ticktype = ..., xlab = ..., ylab = ...,
127       zlab = ..., , phi = 40, theta = 130)) # 3D-diagram log. veroh. fce

```



Obrázek 1: (a) Vrstevnicový diagram (vlevo); (b) 3D-diagram (vpravo) logaritmu dvourozměrné věrohodnostní funkce normálního modelu spolu s maximálně věrohodnými odhady parametrů  $\mu$  a  $\sigma^2$

★