Minimumi vsot odklonov in razmerij

A. Blejec

23. april 2008

1 Vpliv velikih vrednosti na vsote odklonov

Za prikaz lastnosti vsote absolutnih odklonov od a

$$\sum |x - a|$$

vsote kvadratnih odklonov od a

$$\sum (x - a)^2$$

in produkta razmerij x in a

$$\prod e^{log(x/a)^2}$$

sem pripravil funkcijo `showMinDeviation`. Funkcija je v razdelku 2.

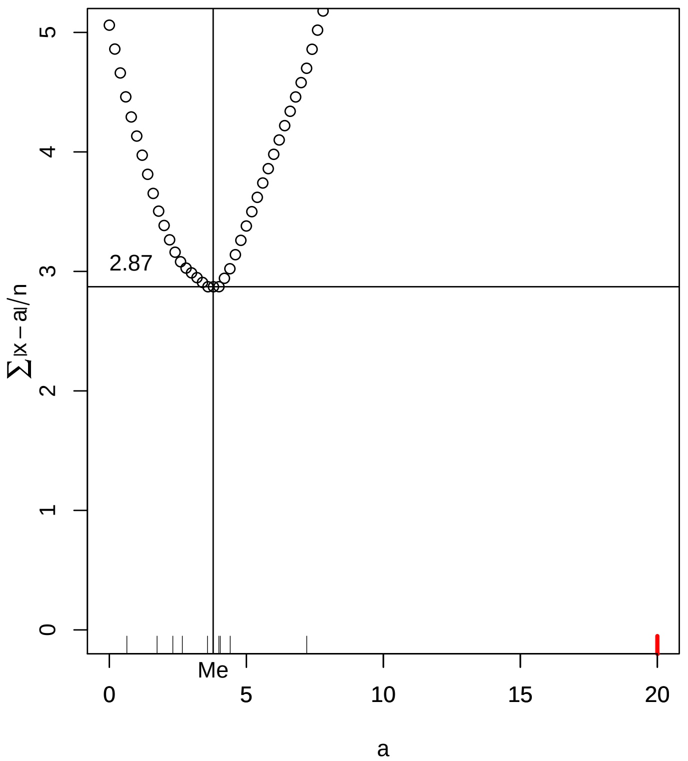
Funkcija prikaže vrednosti odklonov za različne vrenosti a, pri čemer se ena od vrednosti medpodatki spreminja od zelo majhne do zelo velike (argument `Xs=20`). Pri tem opazimo, da je minimum dosežen pri mediani, povprečju (aritmetični sredini) in pri geometrijski sredini.

Mediana se z velikimi vrednostmi nič ne spreminja, in ohranja minimum absolutnih odklonov.

Iz praktičnuh razlogov so na slikah prikazana povprečja odklonov, kar lege ekstremov seveda nič ne spremeni. Na vsaki sliki je uporabljen drug vzorec podatkov.

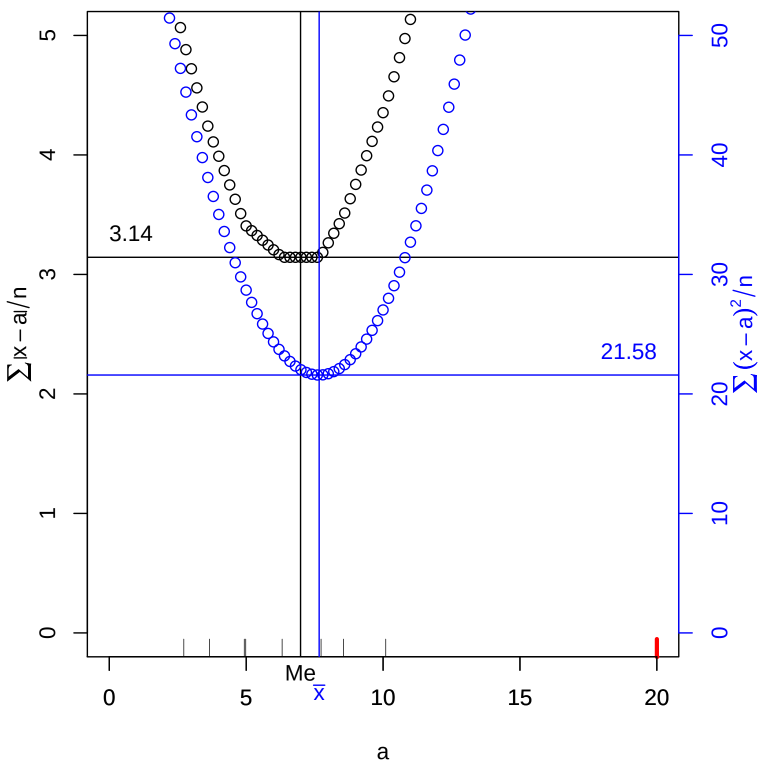
Minimum vsote absolutnih odklonov

```
> showMinDeviation(what = c("median"), Xs = 20)
```



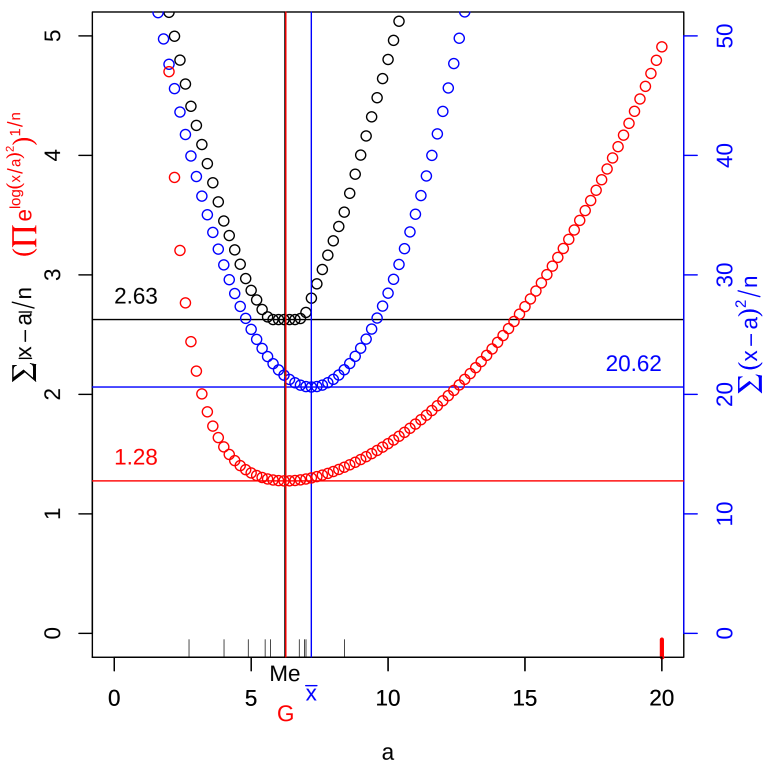
Minimum absolutnih in kvadratnih odklonov

```
> showMinDeviation(what = c("median", "mean"), Xs = 20)
```



Minimumi pri mediani, povprečju in geometrijski sredini:

```
> showMinDeviation(Xs = 20)
```



Za dinamičen prikaz prepišite funkcije iz razdelka 2 v R pri čemer izpustite argument Xs, ki določa kakšne vrednosti naj ima eden od podatkov:

```
> showMinDeviation()
```

2 Funkcije

2.1 gmean
Geometrijska sredina

```
> gmean <- function(x, na.rm = TRUE) {
+     x <- x[!is.na(x)]
+     prod(x)^(1/length(x))
+ }
> gmean(c(1, 10, 100, NA))
```

```
[1] 10
```

2.2 showMinDeviation
Dinamičen prikaz ekstremov vsot odklonov

```
> showMinDeviation <- function(x = rnorm(10, 5, 2), a = seq(0,
+     20, 0.2), Xs = seq(0, max(a), 0.1), xlim = range(a),
+     ylim = c(0, 5), what = c("median", "mean", "gmean"),
+     cols = c("black", "blue", "red")) {
+     n <- length(x)
+     maxa <- max(a)
+     basey <- max(ylim)/15
+     vabs <- Vectorize(FUN = function(x, a) sum(abs(x -
+         a))/length(x), "a")
+     vkv <- Vectorize(FUN = function(x, a) sum((x - a)^2)/length(x),
+         "a")
+     pkvoc <- Vectorize(FUN = function(x, a) prod(exp(log(x/a)^2))^(1/length(
+         "a")
+     par(mar = c(5, 4, 1, 4))
+     for (X in Xs) {
+         x[n] <- X
+         plot(a, a, ylab = "", xlim = xlim, ylim = ylim,
+             type = "n")
+         rug(x)
+         rug(x[n], col = "red", lwd = 3)
+         if (!is.na(pmatch("median", what))) {
+             col <- cols[1]
+             axis(1)
+             mtext(expression(sum(abs(x - a)/n)), 2, 2)
+             points(a, vabs(x, a), col = col)
+             abline(v = median(x), col = col)
+             MINa <- vabs(x, a = median(x))
+             abline(h = MINa, col = col)
+             text(0, MINa + 0.2, round(MINa, 2), xpd = TRUE,
+                 adj = 0, col = col)
+             text(median(x), min(ylim) - basey, "Me",
+                 xpd = TRUE, col = col)
+         }
+         if (!is.na(pmatch("mean", what))) {
+             col <- cols[2]
+             axis(4, col = col, at = axTicks(2), label = axTicks(2) *
```

```
+                  10, col.axis = col)
+              mtext(expression(sum((x - a)^2)/n), 4, 2,
+                  col = col)
+              points(a, vkv(x, a)/10, col = col)
+              abline(v = mean(x), col = col)
+              MINk <- vkv(x, a = mean(x))
+              abline(h = MINk/10, col = col)
+              text(maxa, MINk/10 + 0.2, round(MINk, 2),
+                  xpd = TRUE, adj = 1, col = col)
+              text(mean(x), min(ylim) - basey * 1.5, expression(bar(x)),
+                  xpd = TRUE, col = col)
+          }
+          if (!is.na(pmatch("gmean", what))) {
+              col <- cols[3]
+              points(a, pkvoc(x, a), col = col)
+              abline(v = gmean(x), col = col)
+              MINp <- pkvoc(x, a = gmean(x))
+              abline(h = MINp, col = col)
+              text(gmean(x), min(ylim) - basey * 2, "G",
+                  xpd = TRUE, col = col)
+              text(0, MINp + 0.2, round(MINp, 2), xpd = TRUE,
+                  adj = 0, col = col)
+              mtext(expression((prod(e^{
+                  log(x/a)^2
+              }))^{
+                  1/n
+              }), 2, 2, col = col, adj = 0.8)
+          }
+      }
+ }
> showMinDeviation()
```

3 Minimum vsote absolutnih odklonov

Iščemo minimum vsote absolutnih odklonov

$$\begin{aligned}\sum |x-a| &= \sum_{x\leq a} |x-a| + \sum_{x>a} |x-a| \\ &= \sum_{x\leq a} (a-x) + \sum_{x>a} (x-a)\end{aligned}$$

Odvajajmo po a in izenačimo z 0:

$$\begin{aligned}\frac{d\sum|x-a|}{da} &= \frac{d\sum_{x\leq a}(a-x)}{da} + \frac{d\sum_{x>a}(x-a)}{da} \\ &= \sum(+1) + \sum(-1) \\ &= n(x\leq a) - n(x>a) \\ &= 0\end{aligned}$$

kjer $n(\cdot)$ označuje število vrednosti, ki ustrezajo pogoju v oklepaju.

Iz tega sledi, da doseže izraz ničlo za vsak a, za katerega je število vrednosti, ki so manjše od a enako številu enot, ki so večje od a:

$$n(x\leq a) = n(x>a)$$

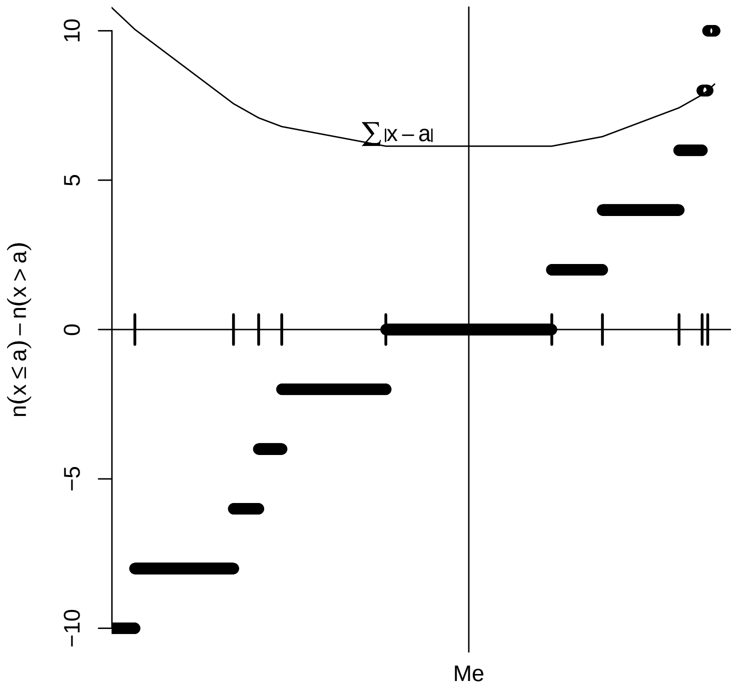
To pa je ravno lastnost mediane.

Odvod je stopničasta funkcija, ki se spremeni ko je a enak kakemu podatku (slika 1). Za a, ki so manjši od mediane je odvod negativen, zato na tem delu z rastočim a vsota odklonov pada. Za a, ki so večji od mediane pa je odvod pozitiven, zato za $a > Me$ vsota odklonov z rastočim a narašča. Torej je ekstrem res minimum.

Zanimivo je, da je za sodo število podatkov $n = 2m$, odvod enak nič za vse vrednosti a, ki so med srednjima podatkoma:

$$x_m < a < x_{m+1} \Rightarrow D(a) = \frac{d\sum|x-a|}{da} = 0$$

Za tiste, ki radi rišemo :) je vse skupaj prikazano na sliki 1, ki kaže podatke (navpične črtice), vrednost odvoda $n(x\leq a) - n(x>a)$ (debele vodoravne črte) in vsoto odklonov (tanka črta), ki je odsekoma linearna. Na sliki imamo 10 podatkov, tako da je dobro viden interval, na katerem je odvod enak nič, vsota absolutnih odklonov pa ekstremno nizka in konstantna.



Slika 1: Odvod vsote absolutnih odklonov.