

# Drobtinice<sup>1</sup>

A. Blejec

14. maj 2013

---

<sup>1</sup>... kot bi temu rekel kolega Batagelj :)

# Nekaj tem s konca prejšnjih predavanj

- ▶ I/O
- ▶ kontrolne strukture
- ▶ paketi !admin
- ▶ ML
- ▶ linearni model

# I/O

- ▶ R je glede branja in pisanja precej fleksibilen.
- ▶ Najbolj zanesljive so navadne text datoteke ... *itak*
- ▶ Več o tem je na <http://ablejec.nib.si/R/HowTo-IOS.pdf>

# Kontrolne strukture

Kontrolne strukture omogočajo odločitve (if)  
in ponavljanje izvedbe ukazov (for, while, repeat)

```
if (interactive()) help("if")
```

if

```
if(cond) expr
```

```
if(cond) cons.expr else alt.expr
```

```
x <- 3
```

```
meja <- 5
```

```
if (x < meja) cat(x, "je manj od", meja, "\n")
```

```
3 je manj od 5
```

```
if (x < meja) cat(x, "je manj od", meja, "\n") else cat
```

```
3 je manj od 5
```

Spremenite mejo v 7 in poskusite znova.

## Več kot en ukaz:

Več ukazov, ki sodijo skupaj zapremo z `{ }` v blok

```
x <- 3
meja <- 5
if (x < meja) {
  lbl <- "Vsota: "
  y <- x + meja
} else {
  lbl <- "Produkt: "
  y <- x * meja
}
cat(lbl, y, "\n")
```

**Vsota: 8**

# ifelse

```
ifelse(test, yes, no)
```

```
x <- c(2:-2)
```

```
ifelse(x >= 0, x, NA)
```

```
[1]  2  1  0 NA NA
```

```
sqrt(ifelse(x >= 0, x, NA))
```

```
[1] 1.414214 1.000000 0.000000      NA      NA
```

switch

```
centre <- function(x, type) {  
  switch(type, mean = mean(x), median = median(x),  
    trimmed = mean(x, trim = 0.1))  
}  
x <- rcauchy(10)  
centre(x, "mean")  
[1] -1.532892  
  
centre(x, "median")  
[1] -1.13698  
  
centre(x, "trimmed")  
[1] -1.29566
```



# Zanka for

Uporabljamo, kadar je znano število ponavljanj ali pa seznam vrednosti, za katere se stvar ponovi. V R ponavadi lahko nadomestimo z vektorskim načinom ali pa s katero od `apply` funkcij. Glej paket **plyr**.

```
for (x in 10:13) {  
  y <- x^2  
  print(y)  
}
```

```
[1] 100  
[1] 121  
[1] 144  
[1] 169
```

```
for (lbl in c("a", "e", "i", "o")) cat(paste("šal",  
  lbl, ", ", sep = ""))
```

```
šala, šale, šali, šalo,
```

# Animacija

Kam gre porazdelitev  $t$ , ko naraščajo stopinje prostosti?

```
n <- seq(1, 30, 0.01)
x <- seq(-5, 5, length = 100)
for (df in n) {
  plot(x, dt(x, df), type = "l", lwd = 7, col = 4,
       main = df, ylim = c(0, 0.4))
  lines(x, dnorm(x), type = "l", lwd = 3, col = 2)
}
df
```

## Zanka se vrti do izpolnitve pogoja

`while(cond) expr`

Število poskusov do prvega uspeha, če je  $p = 0.2$

```
set.seed(123)
(x <- as.numeric(runif(20) < 0.2))
[1] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0
i <- 1
while (x[i] != 1) {
  i <- i + 1
}
i
[1] 6
```

Bolj R -jevsko:

```
which(x == 1)
[1] 6 15 18
which(x == 1)[1]
[1] 6
```

repeat

**x**

[1] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0

**i** <- 1

**repeat** {

**if** (x[i] == 1)

**break**

**i** <- i + 1

}

**i**

[1] 6

# Stopnja značilnosti

```
stars <- function(p) {  
  alfa <- c(0.05, 0.01, 0.001, 0)  
  st <- ""  
  i <- 1  
  while (p <= alfa[i]) {  
    st <- paste(st, "*", sep = "")  
    i <- i + 1  
  }  
  if (st == "")  
    st <- "NS"  
  return(st)  
}  
  
p <- 0.002  
cat("p =", p, stars(p), "\n")  
p = 0.002 **  
sapply(c(0.5, 0.05, 0.005, 5e-04), stars)  
[1] "NS"   "*"    "**"   "***"
```

paketi !admin

```
.libPaths()
```

```
.libPaths()
```

```
[1] "D:/RUSER/r/win-library/2.15"
```

```
[2] "C:/Program Files/R/R-2.15.1/library"
```

Npr:

```
.libPaths("H:/R/library")
```

# ML - metoda največjega verjetja

<http://ablejec.nib.si/pub/rps/MetodeS.pdf>

# Metoda največjega verjetja (ML)



# Mediana, aritmetična in geometrijska sredina

<http://bit.ly/XA8qdX>

# linearni model