

M6120 – 3. CVIČENÍ : M6120cv03 (*Základy grafiky v R*)

A. Úrovně grafiky:

High-level – je vhodná pro tvorbu jednoduchých grafů, patří sem funkce jako `plot\verb, hist, dotplot` atd.

Low-level – umožňuje mnohem více nastavení jednotlivých prvků grafu, dokreslování různých objektů do grafu (kruhy, text, body, matematické křivky atd.)

Interaktivní grafika – dovoluje interaktivně přidávat data do grafu a zpětně informace extrahovat.

Pomocí příkazu `demo(graphics)` (popř. `demo(image)`) podívejte se na pár příkladů.

V případě, že jsou tyto grafy nedostačující, existuje celá řada dalších knihoven, které obsahují velké množství dalších typů grafů (např. balíčky `grid`, `lattice`, `iplots`, `misc3D`, `rgl`, `scatterplot3d`). Existuje celá řada balíčků pro práci s mapami, např. `sp`, `maps`, `maptools`, `rgdal`, `RgoogleMaps` atd.

B. Na adrese

http://www.math.muni.cz/~kolacek/vyuka/vypsyst/navod_R.pdf.

máte k dispozici materiál, ve kterém si prostudujte především 8. kapitolu.

C. Příklady:

PŘÍKLAD 1

Popis dat je v souboru `toxic.txt`, samotná data jsou uložena v souboru `toxic.dat`.

Nejprve načteme popisný soubor pomocí příkazu `readLines`. Díky tomu, že je příkaz v závorkách, ihned se zobrazí obsah proměnné `popis`.

```
> fileTxt <- paste(data.library, "toxic.txt", sep = "")
> con <- file(fileTxt)
> (popis <- readLines(con))

[1] "Toxic Chemical Production Data"
[2] "======"
[3] "Artificial data record the volume of a toxic chemical "
[4] "that is produced as a by-product in a certain industrial"
[5] "manufacturing process."
[6] ""
[7] "The file toxic.dat contains these variables:"
[8] ""
[9] "VOL      The volume of toxic by-product produced (in litres)"
[10] "TEMP     The temperature of the manufacturing process (in°C)"
[11] "CAT      The weight of catalyst (in kg)"
[12] "METHOD   The method used to produce the chemical (qualitative)"

> close(con)
```

Nyní načteme datový soubor `toxic.dat` pomocí příkazu `read.table`. Příkazem `str` vypíšeme strukturu datového rámce a nakonec i jeho obsah:

```
> fileDat <- paste(data.library, "toxic.dat", sep = "")
> data <- read.table(fileDat, header = TRUE)
> str(data)
```

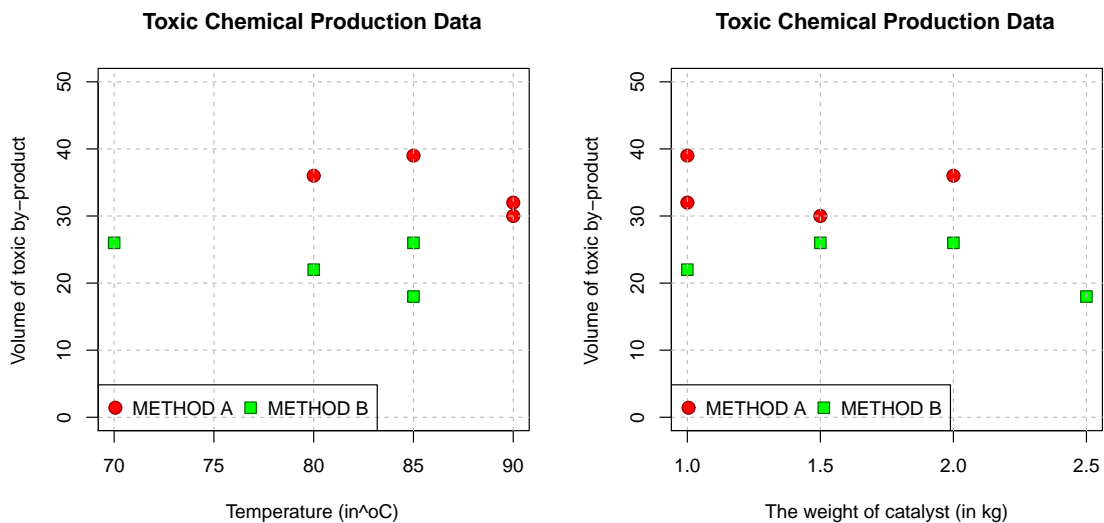
```
'data.frame':      8 obs. of  4 variables:
 $ VOL   : int  30 39 26 36 22 18 32 26
 $ TEMP  : int  90 85 70 80 80 85 90 85
 $ CAT   : num  1.5 1 1.5 2 1 2.5 1 2
 $ METHOD: Factor w/ 2 levels "A","B": 1 1 2 1 2 2 1 2
```

```
> data
```

```
  VOL TEMP CAT METHOD
1  30   90 1.5     A
2  39   85 1.0     A
3  26   70 1.5     B
4  36   80 2.0     A
5  22   80 1.0     B
6  18   85 2.5     B
7  32   90 1.0     A
8  26   85 2.0     B
```

Jazyk R má k dispozici celou řadu grafů, pomocí kterých lze data znázornit. Vytvoříme nejprve dva grafy vedle sebe (viz parametr `mfrow=c(1,2)`), a to vždy pomocí *high-level* příkazu `plot`. Kromě toho použijeme *low-level* příkazy `points`, `grid` a `legend`.

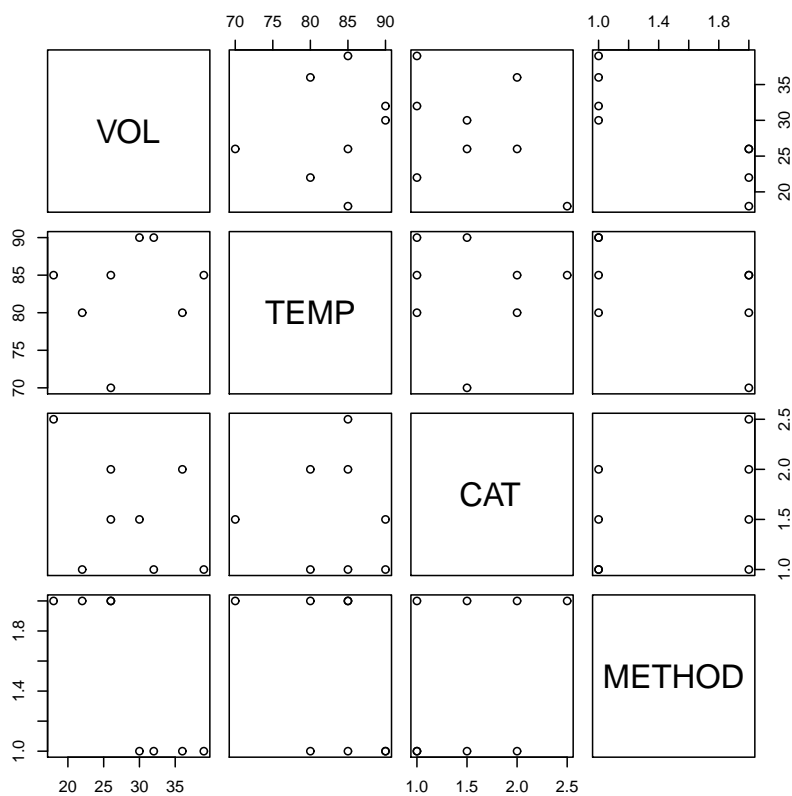
```
> LA <- data$METHOD == "A"
> LB <- data$METHOD == "B"
> par(mfrow = c(1, 2))
> plot(data$VOL ~ data$TEMP, type = "n", ylim = c(0, 50), xlab = "Temperature (in ^oC)",
       ylab = "Volume of toxic by-product", main = "Toxic Chemical Production Data")
> points(data$TEMP[LA], data$VOL[LA], pch = 21, col = "darkred", bg = "red",
        cex = 1.5)
> points(data$TEMP[LB], data$VOL[LB], pch = 22, col = "darkgreen", bg = "green",
        cex = 1.5)
> grid(nx = NULL, ny = NULL, col = "grey", lty = 2)
> legend("bottomleft", paste("METHOD", c("A", "B")), col = c("darkred",
        "darkgreen"), pch = 21:22, ncol = 2, cex = 1, pt.bg = c("red", "green"),
        pt.cex = 1.5)
> plot(data$VOL ~ data$CAT, type = "n", ylim = c(0, 50), xlab = "The weight of catalyst (in kg)",
       ylab = "Volume of toxic by-product", main = popis[1])
> points(data$CAT[LA], data$VOL[LA], pch = 21, col = "darkred", bg = "red",
        cex = 1.5)
> points(data$CAT[LB], data$VOL[LB], pch = 22, col = "darkgreen", bg = "green",
        cex = 1.5)
> grid(nx = NULL, ny = NULL, col = "grey", lty = 2)
> legend("bottomleft", paste("METHOD", c("A", "B")), col = c("darkred",
        "darkgreen"), pch = 21:22, ncol = 2, cex = 1, pt.bg = c("red", "green"),
        pt.cex = 1.5)
```



Obrázek 1: Dva bodové grafy vedle sebe pomocí příkazu plot.

Všimněte si, jaký graf dostaneme, použijeme-li jednoduchý příkaz

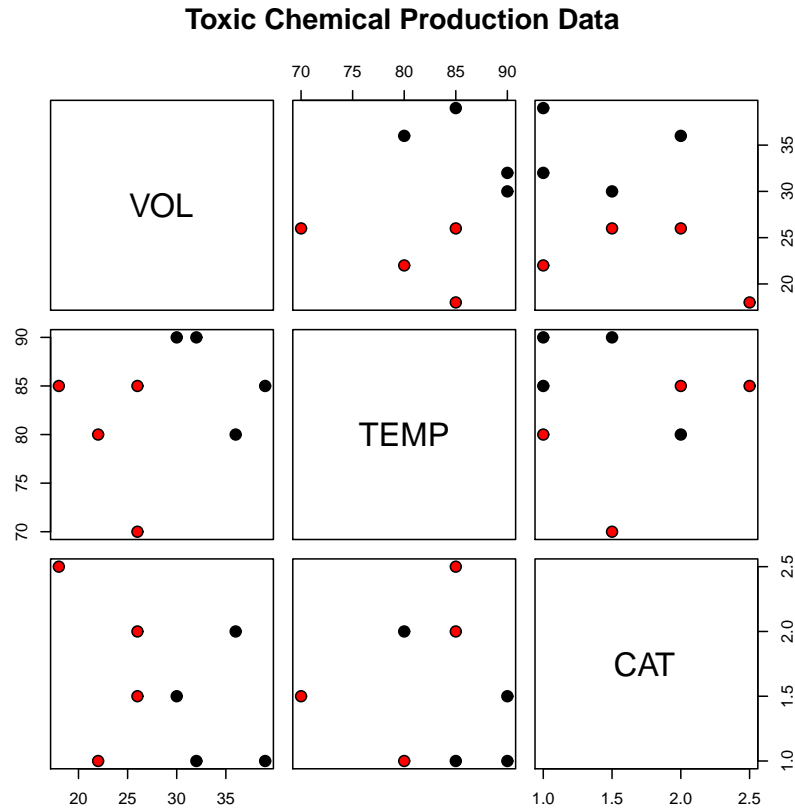
```
> plot(data)
```



Obrázek 2: Maticové bodové grafy pomocí příkazu plot.

Další možností je vykreslit maticový bodový graf pomocí příkazu `pairs`. Díky volbě `pch=21` (vyplněný znak) a především s využitím `bg=as.numeric(data$METHOD)` se nám podařilo barevně odlišit obě dvě metody.

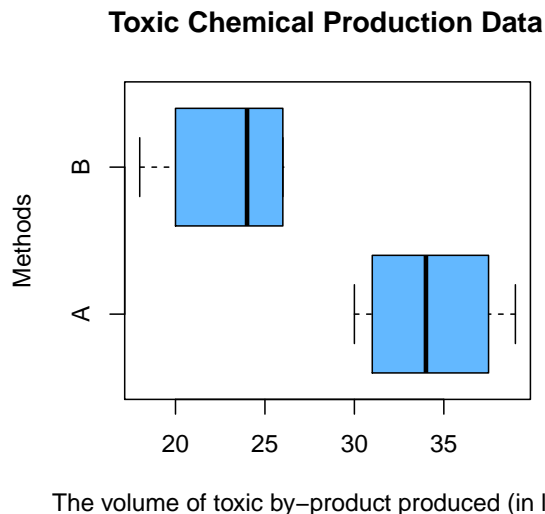
```
> pairs(VOL ~ TEMP + CAT, data = data, pch = 21, bg = as.numeric(data$METHOD),
      cex = 1.5, main = popis[1])
```



Obrázek 3: Maticové bodové grafy pomocí příkazu `pairs`.

Příkaz `plot` lze použít i k vykreslení krabicových grafů, a to díky faktu, že jedna z uvažovaných proměnných je typu faktor.

```
> plot(data$VOL ~ data$METHOD, main = popis[1], horizontal = TRUE, ylab = substr(popis[9],
  9, nchar(popis[9])), xlab = "Methods", col = "steelblue1")
```



Obrázek 4: Krabičkový graf pro VOL dle metod pomocí příkazu `plot`.

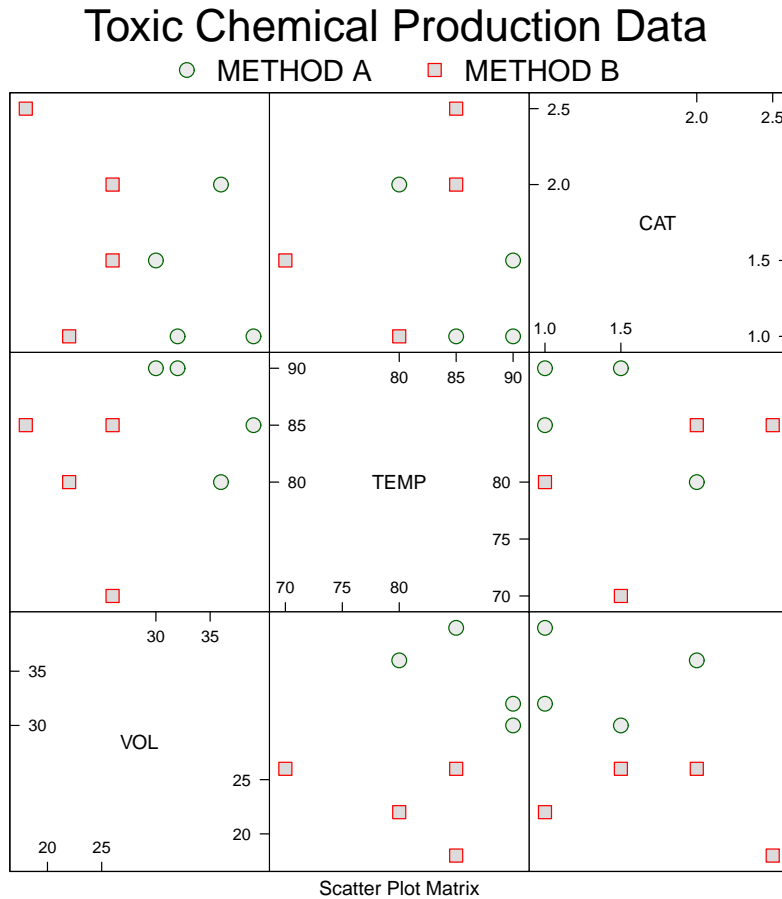
PŘÍKLAD 2

Zůstaneme u datového souboru z předešlého příkladu. Ukážeme si vykreslení bodového grafu pomocí speciálních grafů z knihovny `lattice`, která obsahuje velmi sofistikované grafy vhodné pro práci s datovými soubory, které obsahují vedle spojitých proměnných také celou řadu kategoriálních proměnných.

Vykreslíme nyní bodový graf pomocí příkazu `splom` z knihovny `lattice`. Prostudujte si tento příkaz pomocí nápovědy, tj. například pomocí příkazu `?splom`.

Chceme-li u těchto sofistikovaných grafů přidat legendu (díky specifikacím uvedených u parametru `key`), je třeba ošetřit, aby se tiskly stejné symboly ve stejných barvách jak v legendě, tak v jednotlivých panelech (viz grafická systémová proměnná `superpose.symbol`).

```
> library(lattice)
> trellis.par.set(col.whitebg())
> super.sym <- trellis.par.get("superpose.symbol")
> super.sym$pch <- c(21, 22)
> super.sym$cex <- rep(1.5, length(super.sym$pch))
> graf <- splom(~data[1:3], groups = METHOD, data = data, pch = c(21,
  22), cex = 1.5, bg = c(1, 2), panel = panel.superpose, as.table = T,
  key = list(title = popis[1], columns = 2, cex = 1.5, points = Rows(super.sym,
    1:2), text = list(paste("METHOD", levels(data$METHOD))))))
> print(graf)
```



Obrázek 5: Maticové bodové grafy pomocí příkazu `splom` z knihovny `lattice`.

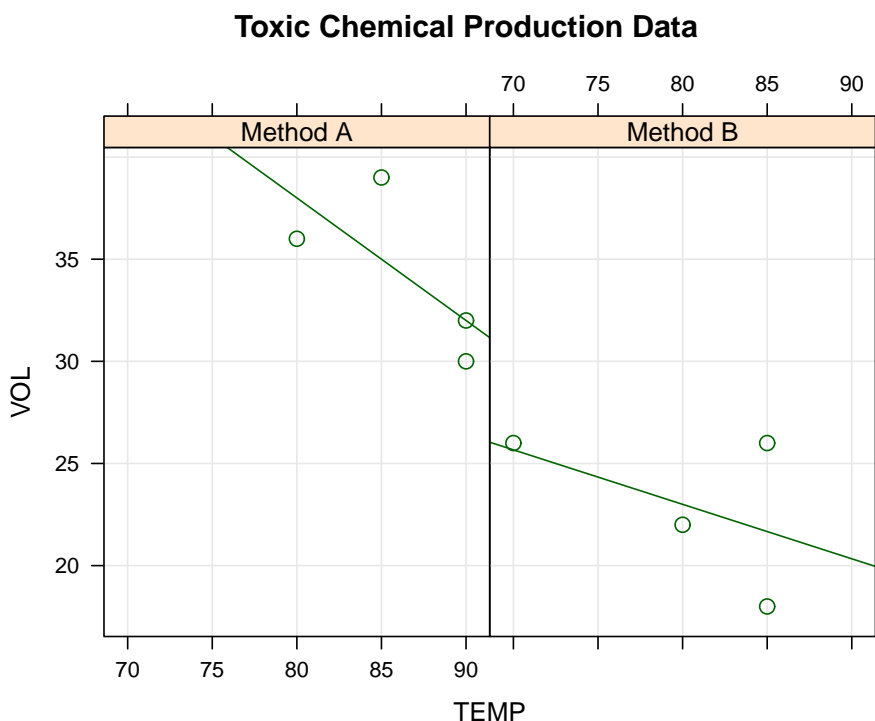
Velmi frekventovaným typem grafu z knihovny `lattice` je graf vytvořený příkazem `xyplot`. Ukážeme jeho různé využití.

Nejprve vytvoříme graf, který pro každou metodu vytvoří samostatný panel. Body (volba "p" u parametru `type`) navíc proložíme regresní přímkou (volba "r" u parametru `type`). Volba "g" u parametru `type` znamená nastavení parametru `grid` (mřížka). Dále si všimněme, že název každého panelu jsme rozšířili o slovo `Method` (viz nastavení parametrů u volby `strip`).

```

> library(lattice)
> trellis.par.set(col.whitebg())
> graf <- xyplot(VOL ~ TEMP | METHOD, data = data, as.table = T,
  type = c("g", "p", "r"), main = popis[1], cex = 1.15,
  strip = strip.custom(factor.levels = paste("Method",
  levels(data$METHOD))))
> print(graf)

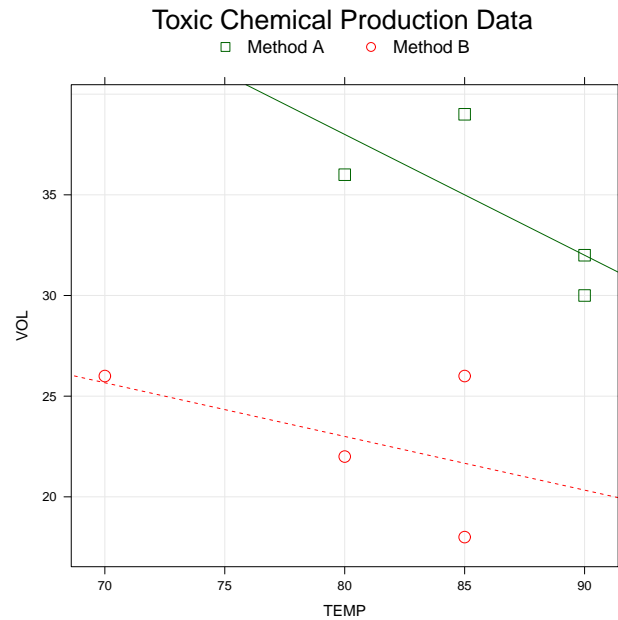
```



Obrázek 6: Dvoupanelový bodový graf (spolu s proloženou regresní přímkou) pomocí příkazu `xyplot` z knihovny `lattice`.

Nyní vytvoříme graf pouze s jedním panelem, ale obě metody rozlišíme graficky, a to tak, že přidáme volbu `group=METHOD`. Body (volba "p" u parametru `type`) opět proložíme regresní přímkou (volba "r" u parametru `type`).

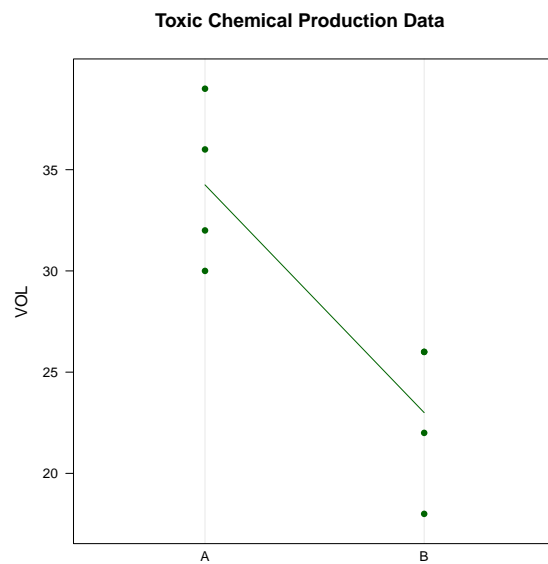
```
> library(lattice)
> trellis.par.set(col.whitebg())
> super.sym <- trellis.par.get("superpose.symbol")
> super.sym$pch <- c(0, 1)
> super.sym$cex <- rep(1.15, length(super.sym$pch))
> graf <- xyplot(VOL ~ TEMP, group = METHOD, data = data,
  type = c("g", "p", "r"), pch = c(0, 1), cex = 1.5,
  bg = c(1, 2), panel = panel.superpose, key = list(title = popis[1],
  columns = 2, cex = 1.15, points = Rows(super.sym,
  1:2), text = list(paste("Method", levels(data$METHOD))))))
> print(graf)
```



Obrázek 7: Jednanelový bodový graf (spolu s proloženou regresní přímkou) pomocí příkazu `xyplot` z knihovny `lattice`.

Dalším zajímavým grafem je tzv. `dotplot`. Volba "a" (*average*) u parametru `type` způsobí, že do grafu se zakreslí úsečka spojující výběrové průměry proměnné VOL obou metod.

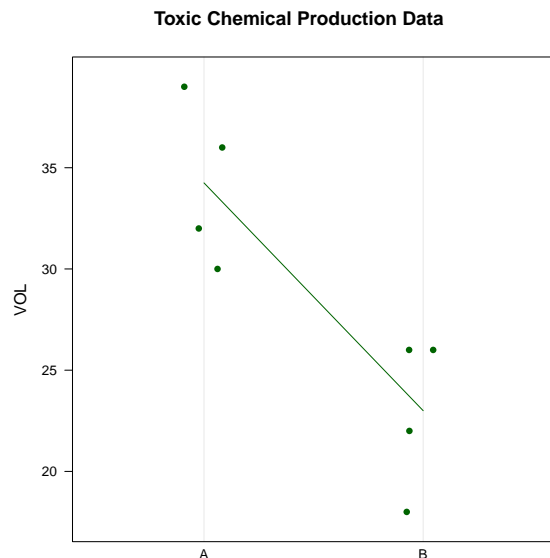
```
> library(lattice)
> graf <- dotplot(VOL ~ METHOD, data = data, type = c("p",
  "a"), main = popis[1])
> print(graf)
```



Obrázek 8: Bodový graf (spolu se spojnicí skupinových průměrů) pomocí příkazu `dotplot` z knihovny `lattice`.

Vzhledem k tomu, že u metody B dva body splývají, provedeme jejich náhodné posunutí (rozmítnutí) pomocí parametru `jitter.x`.

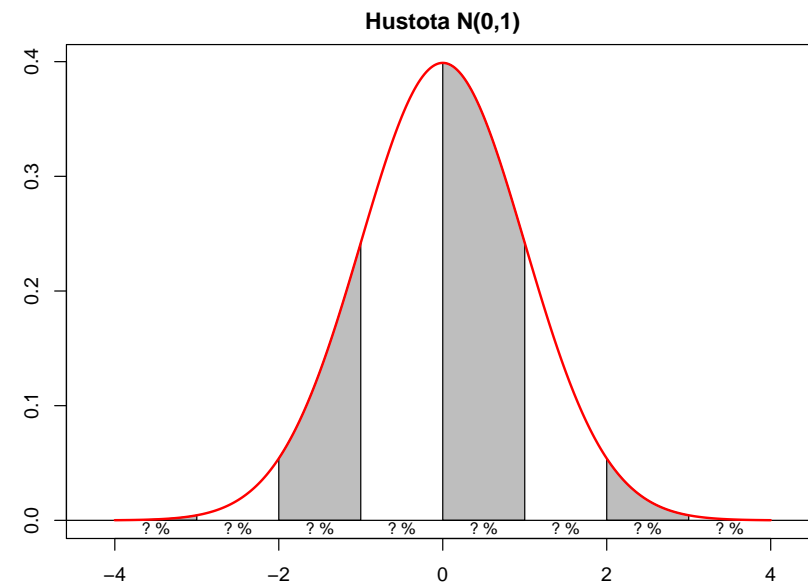
```
> library(lattice)
> graf <- dotplot(VOL ~ METHOD, data = data, type = c("p",
  "a"), main = popis[1], jitter.x = TRUE)
> print(graf)
```



Obrázek 9: Rozmítnutý (*jittered*) bodový graf (spolu se spojnicí skupinových průměrů) pomocí příkazu `dotplot` z knihovny `lattice`.

D. Úkol:

Napište příkazy v jazyku R tak, aby vznikl následující graf. Otazníky nahrad'te skutečnými hodnotami. Návod: text týkající se procent doplňte pomocí příkazu `mtext`.



Obrázek 10: Hustota standardizovaného normálního rozdělení