Grafika

Csima Judit

BME, VIK,
Számítástudományi és Információelméleti Tanszék

2013. március 14. és 21.

Véletlen minta előállítása

Többféle eloszlásra beépített véletlen-minta generálás:

- `rnorm(db, mean, dev)`

```
> rnorm(5, 1,2)
[1] 4.024425 1.165931 2.134442 -1.049097 1.646013
```

- `rbinom(db, kiserletek, sikervaloszinusege)`

```
> rbinom(10, 3, 0.6)
[1] 2 1 2 3 2 3 2 2 0 2
```

- `rpois(db, lambda)`

```
> rpois(10,2)
[1] 3 1 4 1 1 1 2 3 4 3
```

Véletlen minta összefoglalása

- ```
  > x <- rnorm(5, 1,2)
  > x
  [1] 3.64800 2.23127 3.18333 1.61320 0.77968
  > summary(x)
     Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
   0.7797    1.6130   2.2310   2.2910    3.1830   3.6480
  ```
- ```
  > y <- rbinom(10,2, 0.4)
  > y
  [1] 0 0 2 0 1 1 0 1 1 0
  > z <- as.factor(y)
  > summary(z)
  0 1 2
  5 4 1
  ```

Factors

- factor: egy lehetséges típusa az objektumoknak
- kategória típusú adatokra használatos
- szintjei vannak
- ```
  > y <- factor(c(1, 1, 2, 1, 2, 2))
  > y
  [1] 1 1 2 1 2 2
  Levels: 1 2
  > table(y)
  y
  1 2
  3 3
  ```
  ```

Grafika az R-ben

Van néhány alapvető package az ábrázolásra:

- `graphics`: ez az alap (base) grafikai csomag, legfontosabb függvények: `plot, hist, boxplot`
- `lattice`: Trellis típusú objektumokat lehet vele csinálni, független a graphics csomagtól, más filozófia, fő parancsok: `xyplot, bwplot, levelplot`
- vannak még más rafinált ábrázolási eszközök is

Milyen grafika kell?

Mit használjunk: base-t vagy lattice-t? Más a filozófia:

- Base: apránként rakjuk össze, amit akarunk, külön parancsokkal állítjuk be az ábra paramétereit
- Lattice: egy függvényt írunk, ami tartalmaz minden beállítást, amit szeretnénk

Mi alapján döntjük el, hogy milyen ábrát csinálunk és mivel?

- Csak egyszer akarunk ránézni az ábrára a képernyőn vagy prezentációban ill. cikkben akarjuk használni?
- Sok adatunk van? Milyen típusúak?

Base grafika alaptulajdonságai

- Általában ezt használjuk, 2D-s ábrák lesznek belőle
- (az oprendszertől függő helyen) megjeleníti az ábrát a képernyőn
- rstudioban a jobb alsó sarokban új ablakban
- fő parancsok plot(x,y) és hist(x,y)

Paraméterek a base grafikában

- rengeteg paraméter, beállítható az ábra címe, a tengelyek nevei, a margók, stb.
- a paraméterek egy része globálisan állítható be : minden ábrára érvényes lesz ezután
- a globális beállítások egy része felülírható az egyes `plot` hívásokból
- vannak csak a `plot`-ban beállítható paraméterek is

A base grafika paraméterei

A `par` függvénnyel lehet beállítani a globális értékeket, lokálisan a `plot` függvény belsejében adhatunk meg paramétereket

A legfontosabb globális paraméterek:

- `pch`: milyen szimbólummal jelöljük a pontokat az ábrán (default a karika)
- `col`: a pontok színe milyen legyen? (lehet szám, színnév, stb., a `colors` fv. megadja a lehetőségeket)
- `las`: a tengelyeken levő számok milyen irányúak legyenek
- `mar`: a margó mérete

Még paraméterek:

Globálisak:

- `mfrow`: hány ábrát jelenítsen megy soronként, oszloponként (soronkénti feltöltés)
- `mfcol`: hány ábrát jelenítsen megy soronként, oszloponként (oszloponkénti feltöltés)

Lokális:

`xlab, ylab`: a tengelyek nevei

Default értékek lekérdezése, help

- `help(par)`: milyen grafikus paraméterek vannak, milyen lehetséges értékekkel
- par("paraméter neve"): kiírja a default értéket

```
> par("pch")
[1] 1
```

- az ábrázoló fvek (`plot, hist`), stb. help-jeiből

Ábrázoló függvények

- `plot, hist, boxplot`: pont-diagramm, hisztogram és box-plot rajzolása
- `text`: adott helyre szöveget illeszt be az ábrába pont-diagrammon
- `title`: ábra címe (ezt lehet a `plot` híváson belülről is a main paraméter beállításával)
- `points`: további pontokat ad az ábrához

Kiírás más formátumba

- sokféle formátumba lehet konvertálni a kapott ábrát
- módszer:
  - elkészítjük az ábrát a képernyőre
  - más formátumba konvertáljuk
- átkonvertálás egyszerűen: rstudioban Export as Image vagy as PDF
- vannak függvények az átírásra pl. `dev.copy2pdf`
- help ehhez: `?Devices`, `?dev.copy`

Lattice grafika

- nincs benne a base R-ben, le kell tölteni hozzá a lattice csomagot: `library(lattice)`
- máshogy működik, mint a base grafika:
  - base grafika közvetlenül a képernyőre készítette el az ábrát, amit el lehetett menteni
  - lattice grafika egy trellis típusú objektumot hoz létre, amit külön paranccsal lehet megjeleníteni a képernyőn (kivéve a command line-os mód, mert akkor rögtön kirakja képernyőre)
  - help: `> package ?lattice`

Alapvető grafikai függvények a lattice-ben

- `xyplot`: pontdiagrammhoz
- `bwplot`: box-plothoz
- `histogram`: hisztogrammhoz
- vannak még mások is, megnézhető így: `?xyplot`

Formula az első argumentumban

Lattice függvények általában y ∼ x | f * g típusú formulát kapnak első argumentumként, ennek jelentése:

- a ∼ jeltől jobbra az y tengely, balra az x tengely változója áll
- a | jel után áll az a változó, ami (mint faktor) szerint szétvágja az adatokat és külön jeleníti meg az egyes ábrákat, egymás mellett (ez a rész hiányozhat is)
- f*g azt jelenti, hogy két faktort is megadhatunk és akkor az összes lehetséges kombinációjukra lesz egy-egy ábra

További argumentumok beállítása, panel függvény

- ha nincs f*g rész (egy ábra készül): hasonóan, mint base-ben volt: `xlab, ylab, main, pch, col stb.`
- ha több ábra készül: `panel` függvénnyel:

```
xyplot(y ~ x | f,
         panel = function(x, y, ...)  {
                 panel.xyplot(x, y, ...)
                 panel.lmline(x, y, col = 2)
}
)
```