Grafika

Csima Judit

BME, VIK,
Számítástudományi és Információelméleti Tanszék

2014. március 13. és 20.

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`
- `ggplot2`: ötvözi a fenti két package előnyeit, alapparancs: `qplot` és `ggplot`

Milyen grafika kell?

Fő különbség base és lattice között:

- base: apránként rakjuk össze, amit akarunk, külön parancsokkal állítjuk be az ábra paramétereit, az ábra a képernyőre kerül, innen lementhető képként
- lattice: egy függvényt írunk, ami tartalmaz minden beállítást, amit szeretnénk, az eredmény egy R objektum

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?
- Melyiket szeretjük használni, milyen típusú ábrát akarunk

Base grafika alaptulajdonságai

- 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)
}
)
```

ggplot2 általános elvek

- `qplot()` függvénynél egy csomó paraméter értéke be van állítva, ezzel lehet ábrázolni
- de ha ennél rafináltabbat akarunk: `ggplot()`
- lehet apránként összerakni az ábrát (mint base-nél)
- könnyen lehet vele egy factor szerint szétvágott adathalmazról sok kis ábrát egymás mellé készíteni (mint lattice-nál)

`qplot()`

- olyan, mint `plot` volt
- legtöbb dologról maga gondoskodik (margóméret, betűméret, tengelyek felirata, stb.)
- ha nem tetszik, amit csinál, akkor `ggplot()`
- ezzel lehet pontdiagrammot és hisztogrammot is

qplot()

- mindenképp meg kell mondani, hogy mi a két tengely változója és mi a data frame, ahonnan jönnek
- ezután további dolgok megadhatók, pl. xlab, ylab, main, illetve
  - aesthetics: pont színe, formája, mérete
  - geoms: pontok összekötése, illesztések

facets

- ha egy faktor típusú változó szerint szétbontva akarjuk ugyanazt ábrázolni a különböző csoportokban
- hisztogrammra is működik
- exploratory elemzésnél is hasznos