R nyelv – bevezetés, változók, struktúrák

Vadon Viktória

2024/2025/I. félév

Tartalomjegyzék

1. Véletlen input generálása

- r, mint random (véletlen) + eloszlás R által ismert neve, pl. rnorm(n), vagy runif(n), rexp(n), ...
- eloszlások listája: súgó `?stats::distributions`
- kötelező argumentum: n darabszám (pozitív egész) – n hosszú vektort ad vissza
  – alternatív argumentum: vekt vector, akkor darabszám n = length(vekt)

- eloszlástól függően kötelező vagy opcionális további argumentum(ok)ként az eloszlás paramétere(i)
- pl. rnorm() alapért. standard normális, 0 várható érték, 1 szórás; runif() 0 és 1 közt (folytonos) egyenletes eloszlás, exponenciális lambda=1, stb.
- vagy néha egyszerűbb vektorműveletekkel transzformálni, pl. 2*rnorm(n)+3 – 3-as várható érték, 2-es szórás
- egyéb transzformációk is végezhetők vele, pl. nincs diszkrét egyenletes eloszlás – ezt folytonos egyenletesből tudjuk generálni kerekítéssel, alsó- vagy felsőegészrész függvénnyel: round(), floor(), ceiling()

2. 2D ábrázolás alapjai

2.1. Alapparancs: plot

- alap parancs: plot(x,y)
- ld. súgó `?graphics::plot`, `?graphics::plot.default`
  - kell: x, y numeric vektorok, n = length(x) = length(y).
  - értelmezés: n pont, (x(i),y(i)) koordinátákkal
  - mindig új ábrát kezd! – előző ábrához hozzáfűzés: ld. későbbi parancsok
  - alapértelmezés: pontfelhő / különálló pontok
  - plot(y) : x koordinátának 1:length(y)-t veszi.
  - plot(x,y,...) – ... helyére egyéb grafikus paraméterek, opcionális argumentumok
- type=
  - ábrázolás típusa, mint pontfelhő, töröttvonal, stb.
  - alapértelmezett `type="p"`, pontok
  - `type="l"` l = kis L, line: töröttvonal – vigyázzunk, mert nem balról jobbra, hanem index szerint növekvő rendben köti össze a pontokat!
  - `type="b"` b = both: töröttvonal és pontok – gyakori használat: függvénygörbe markerekkel
  - egyéb elfogadott értékek és hatásuk: ld. súgó, `?base::plot.default`
- ábrázolási tartomány:
  - xlim=c(min,max), ylim=c(min,max)
  - opcionális, magától is az ábrázolt pontokhoz igazít!
  - akkor érdemes kézzel, ha van később hozzáadott tartalom, ami kilógna
- asp= : aspect ratio, „képarány”; y/x tengelyskála-arány.
- címkék
  - main= : ábra felirata
  - xlab= , ylab=: x és y tengely felirata
- logaritmikus skála
  - log="x", log="y" vagy log="xy": x, y vagy mindkét tengelyen logaritmikus skálázásra kapcsol – alapért.: lineáris skála; következő plot() paranccsal

visszaáll az alapért.

2.2. Grafikus paraméterek

- ezek használhatók a plot() paranccsal is, de a lenti, ábrát kiegészítő parancsokkal is.
- grafikus paraméterek súgó `?graphics::par`
- pch= : plotting character, avagy marker
  - több formában is megadható, pl. `pch="o"`, vagy `pch=5` – pozitív egészekhez előre definiált karakterek társulnak, elég hosszú a definiált lista (majd ciklikusan ismétel)
  - használható character vagy numeric vector is – ciklikusan ismétlődve párosítja a pontokkal
  - vagy, pch=factor esetén, ha a factor valamilyen kategorizálást jelent, a különböző kategóriák különböző markert kapnak
- col= : szín
  - több mint 600 elérhető szín, colors() paranccsal listázhatók
  - alapszínek angol nevükkel, pl. `col="red"`
  - ez is használható col=3 formában – előre definiált 8 alapszín ismétlődik ciklikusan
  - szintén használható vector-ral vagy factor-ral is!
- lty= : line type, vonalstílus: pontozott, szaggatott, stb.
  - 0-7 számokhoz előre definiált vonalstílusok
  - apaért. 0 : folytonos vonal
- lwd= : line width, vonalvastagság
  - pozitív szám, de tört is lehet; alapért: 1

2.3. Ábra kiegészítése

- a plot() mindig új ábrát kezd; ezek a parancsok az előző ábrába illesztenek be:
- points(x,y): kb. plot(x,y,type="p")
- lines(x,y): kb. plot(x,y,,type="l")
- de, tulajdonképpen ők is bármilyen type= értékkel használhatók!
- speciális: abline(), egyenes ábrázolására
  - abline(a,b): a, b valós számok, a+bx formájú egyenes; vagy abline(v), 2-dimenziós vektor, értelmezés v(1)=a, v(2)=b
  - abline(h=y): vízszintes, y magasságban – itt y lehet vektor, akkor több vízszintes vonal.
  - abline(v=x): függőleges, x koordinátá(k)ban – x is lehet vektor.

3. Függvényábrázolás

3.1. Függvény definíció

- függvény definíciója: függvény objektummal
- függvény objektum létrehozása:
- pl. atfogo <- function(a,b) sqrt(a^2+b^2)
- használat: atfogo(3,4) – eredmény: 5
- általában: fuggveny.neve <- function (argumentumok) {utasítás 1; utasítás 2; visszaadott érték/formula}
  - többlépéses számítási utasítás a fenti formában: kapcsos zárójelekkel összefogva, pontosvesszővel elválasztva. használhatók benne helyi változók! az utolsóként kiszámított érték, vagy meghívott (belső) változó a visszaadott érték
- ezután használat/kiértékelés adott pontban: fuggveny.neve(argumentumok)
- vector input
  - ha valamelyik inputra adott hosszú vector formájában számítunk (pl. mert egy eloszlás paramétervektora), használható v[i], stb. a kiszámításban!
  - de figyeljünk, hogyan definiáltuk: ha úgy vector az input, hogy egyszerre több pontban szeretnénk kiszámítani a függvény értékét, úgy működik-e, ahogy várjuk? ehhez jó(?) hír: sok beépített függvény, mint abs(), sin(), stb. koordinátánként hat.
- egy függvény lehet helyileg használt, névtelen függvény is:
  - pl. ha csak egyszer ki akarjuk értékelni: function(a,b) {sqrt(a^2+b^2) } (3,4)
  - vagy ha egy beépített parancs/függvény/metódus egy függvényt vár inputként: function(a,b) { sqrt(a^2+b^2) }

3.2. 2D függvényábrázolás

- mivel a plot csak töröttvonalat tud, vegyünk fel alappontokat a kívánt (min,max) ábrázolási tartományban, kellően sűrűn, egy x vektorba, pl. seq(min,max,by=...) segítségével
- számítsuk ki ezekben az alappontokban z függvényértékeket egy y vektorba
  - lehet f függvény definiálásával (ld. függvény objektum) és kiértékelésével – ha sikerült úgy definiálni az f függvényt, hogy koordinátánként hat egy vector-ra
  - vagy szimplán kiszámítjuk a formulát vektorműveletekkel, beépített függvényekkel, stb.
- ábrázolás: plot(x,y,type="l")

3.3. 3D függvényábrázolás

3.3.1. Kétváltozós függvény kiszámítása

- cél: z = f(x,y) felület ábrázolása
- legyen x vektor egy felbontása az (xmin,xmax) tartománynak, y vektor egy felbontása az (ymin,ymax) tartománynak
- kiértékelés: z <- outer(x,y,f) paranccsal
  - f a kiértékelni kívánt kétváltozós függvény, akár itt névtelen függvényként megadva, vagy ha korábban definiáltuk, itt elég a neve
  - z egy mátrix, z[i,j] az (x[i],y[j]) pontban számolt függvényérték

3.3.2. Ábrázolási módok

- image(x,y,z)
  - „hőtérkép”, 2D
  - szintek változtatásához színpaletta generálás is... ld. súgó és hivatkozásai.
- contour(x,y,z)
  - szintvonalak, 2D
  - contour(x,y,z,nlevels=...) – szintvonalak száma – közeli megengedett értékre kerekít
  - contour(x,y,z,add=TRUE) – nem kezd új plot-ot, az előző ábrába rajzol – így kombinálható image-el, „domborzati térkép”
- persp(x,y,z)
  - „3D” rácsháló
  - statikus, nem forgatható kép
  - de irányszög megadható: persp(x,y,z,theta=,phi=) – theta az irány („polár koord az x-y síkban”), phi a magasság (x-y síkkal bezárt szög), alapért theta=0, phi=15

4. Egyéb, speciális ábrázolások

4.1. Valószínűségi eloszlások függvényei

- pnorm(x), pexp(x), stb., p + eloszlás neve: eloszlásfüggvény, $F(x) = \mathbb{P}(X \leq x)$[1]
- dnorm(x), dbinom(x), stb: súly- vagy sűrűségfüggvény
  - diszkrét eloszlásra, pl. dbinom() súlyfüggvény: $p(x) = \mathbb{P}(X = x)$
  - folytonos eloszlásra, pl. dnorm() sűrűségfüggvény $f(x) = F'(x)$
- qnorm(q), stb.: kvantilisek
  - mit jelent a kvantilis, **ha egy adatsor van előttünk:**
  - rendezzük növekvő rendbe. – sort()

[1] az R-ben nyugat-európai konvenció szerint így definiált az eloszlásfüggvény, jobbról folytonosan! mi magyarok az orosz, balról folytonos $F(x) = \mathbb{P}(X < x)$ konvenciót vettük át.

  - korábbról ismert, nevezetes: medián, a rendezett adatsor közepe – más néven 0.5-kvantilis, vagy 50-percentilis (adatsor 50%-ánál van)
  - általánosan: q-kvantilis: az adatsor q hányadánál lévő érték, pl. 0.25-kvantilis (25-percentilis) a rendezett sor 1/4-énél lévő érték, avagy, a minta negyede kisebb nála.
  - mit jelent a kvantilis, **elméleti valószínűségi változóról beszélve:**
  - q-kvantilis: az az x szám, aminél a valószínűségi változó q eséllyel kisebb; azaz $F(x) = \mathbb{P}(X \leq x) = q$ megoldása, ha most $q \in [0, 1]$ adott és x az ismeretlen
  - fentiekből: $q = F^{-1}(x)$, gondolhatunk a kvantilis függvényre úgy, mint az eloszlásfüggvény inverze.
  - pl. medián, 0.5-kvantilis, 50-percentilis: mi az a szám, aminél a valószínűségi változó 0.5 (50%) eséllyel kisebb?

4.2. Több ábra egyben

- tegyük fel, hogy adott egy pontok nevű 3 oszlopos data.frame, x, y oszlopai a pontok koordinátái, halmaz oszlopa egy factor változó, azt jelöli, hogy melyik pont melyik halmazba tartozik.
- plot(pontok): egy táblázatban minden lehetséges változópárt ábrázol
  - sorokat, oszlopokat is változónként indexel; adott ábrában sorindex változó az y tengelyre, oszlopindex az x tengelyre.
  - itt factor változókat is as.numeric()-kel számként kezel.
- `coplot(pontok$y ~ pontok$x | pontok$halmaz)`
  - halmaz változó értékei szerint „külön” koordináta-rendszerben ábrázolja a pontokat, x-y koordinátáik szerint; felette jelzi a halmaz változó szintjeit
  - általánosan: `coplot(y ~ x | factor)`, vagy `coplot(y ~ x | factor1 * factor2)` – y-t ábrázolja x függvényében (azaz y kerül az y tengelyre, x az x tengelyre), factor(ok) szerinti bontásban – 2 factor esetén táblázatba rendezve, felette és mellette a factor-ok szintjei

4.3. Eloszlás vizsgálata

- **Hisztogram**
- fakt factor típusú változóra plot(fakt) hisztogram jellegű, gyakorisági ábra
- x vector típusú, folytonos változóra hist(x)
  - automatikusan intervallumokra bont – elég jó a heurisztikája, nem túl sok, nem túl kevés intervallum, „kerek” számok mentén vágva; átlátható eredmény
  - kézi intervallumok: hist(x,breaks=), ahol
  - breaks=n intervallumok számát adja meg (közeli „megengedett” értékre kerekít, továbbra is „kerek” számok mentén vág)

  - vagy breaks=vekt, ahol vekt vector elemei adják meg az intervallumok kezdő- és végpontjait – le kell, hogy fedje a teljes terjedelmet!
  - **relatív** gyakoriság: hist(x,probability=TRUE) – ez az, amit érdemes empirikus sűrűségfüggvénnyel összehasonlítani.
- empirikus sűrűségfüggvény: density(x)
  - egy komplex objektum – nem megyünk bele mélyen
  - a hisztogramból számolja, simítással; density(...,bw=) bw= bandwidth, egy simítási paraméter – minél nagyobb, annál jobban kisimítja a függvényt.
  - ábrázolása: `plot(density(x),type="l")`, vagy `lines(density(x))`
- empirikus eloszlásfüggvény: ecdf(x)
  - jelentése: egy adott x pontban megmutatja, az adatok mekkora hányada kisebb, mint x
  - képlet: $F_n(x) = \#\{x_i : x_i \leq x\}/n$ [2]
  - szintén komplex objektum
  - pl. lekérhetők belőle (empirikus) kvantilisek quantile(ecdf(x),q) formában; $q \in [0, 1]$, de lehet egyetlen szám, vagy egy vector is, több kvantilis lekérése egyszerre – output percentilis formájában
  - ábrázolás: plot(ecdf(x)) – *jobbról folytonos* lépcsős függvény
- boxplot(x)
  - adatok terjedelmét, (nem-)szimmetriáját hivatott vizuálisan szemléltetni
  - szürke doboz az adatok 50%-át tartalmazza: 0.25-kvantilistől (1. negyed, 1. kvartilis) 0.75-kvantilisig (3. negyed, 3. kvartilis);
  - a vastag vonal a medián: adatok fele kisebb nála
  - a „bajuszok" pedig általában a legkisebb és legnagyobb elemet jelzik – de korlátos, hogy milyen távol kerülhet a doboztól, kiugró értékek, outlier-ök kieshetnek belőle, azokat karikával jelzi.
  - boxplot(x,range=1.5): range paraméter adja meg, milyen távol kerülhet a bajusz. alapértelmezés: 1.5, azaz a bajusz max. 1.5-ször olyan távol lehet, mint a szürke doboz mérete (interquartile range); range=0 esetén nincs korlát.
- plot(factor,vector) hívás esetén: factor szerint szétvágja vector-t, és külön-külön boxplot-on ábrázolja őket
  - pl. ha egy data.frame egyik oszlopa hallgatók magassága vector, másik oszlopa hallgatók neme factor, akkor összehasonlítható a férfiak és nők magassága terjedelem szempontjából.
- qqplot()
  - célja: adathalmaz a sejtett valószínűségi eloszlást követi-e?
  - elv: itt nyilván nem elég átlagot, szórást összehasonlítani. ötlet: ha a teljes eloszlások azonosak, és kellő mennyiségű adatunk van, akkor az összes megfelelő kvantilis közel egyenlők! pl. adatok mediánja közel van az elvi mediánhoz, stb.
  - módszer: x tengelyre elvi eloszlás, val. vált. q-kvantilisei, y tengelyre az ada-

[2] Itt is nyugat-európai, jobbról folytonos konvenció!

tok empirikus q-kvantilisei – megnézzük, kb. illeszkednek-e ezek a pontok a 0-n átmenő, 1 meredekségű egyenesre? – ha egy másik egyenesre illeszkednek, akkor az eloszlás típusa jó, csak az átlag, szórás paraméterek nem.

- használat: pl. qqplot(qexp(q),x) – ahol q: [0,1]-beli vector, amilyen kvantiliseket megnézünk, bele a kívánt eloszlás (itt exponenciális) qexp() kvantilisfüggvényébe; másik argumentum csak az adatok vektora (abból automatikusan számol kvantiliseket)
- **speciális eset:** qqnorm(x) normális (Gauss) eloszlással való összehasonlításra, nem kell kézzel a normális kvantiliseket
- **segédfüggvény:** qqline()
  - qqplot ábrához ad hozzá, egyenest illeszt az ábrázolt pontokra – alapértelmezésben 0.25, 0.75 kvantilisen át – módosítás: probs=c(0.25,0.75)
  - normál eloszlásnál: elég qqline(x)
  - más eloszlásnál: qqline(x,distribution=exp)
  - ha meg akarjuk adni az eloszlás paramétereit is, névtelen függvény objektummal: qqline(x,distribution=function(q) qexp(q,rate=2))