

Az R használata

Jeszenszky Péter

Debreceni Egyetem, Informatikai Kar

jeszenszky.peter@inf.unideb.hu

Utolsó módosítás: 2019. június 3.



Futtatás

- Minden egyes feladathoz célszerű egy külön alkönyvtárat létrehozni és abban elindítani a programot.
 - Kilépéskor automatikusan elmenthető ide a felhasználó munkaterülete.
 - Így nem vesz el az elvégzett munka, később bármikor ott folytatható, ahol kilépéskor abbahagytuk.
- A tipikus futtatás Linux környezetben:

```
$ mkdir work      # munkakönyvtár létrehozása  
$ cd work        # belépés a munkakönyvtárba  
$ R               # az R indítása
```

Inicializálás az R indításakor (1)

- Az inicializálás főbb lépései:
 1. Felhasználói profil betöltése.
 2. A előző munkamenet végén elmentett munkaterület betöltése.
 3. A korábbi munkamenetekből származó parancs előzmények betöltése.
- Az inicializálás menetének részletes leírása megjeleníthető a `?Startup` paranccsal.

Inicializálás az R indításakor (2)

- Felhasználói profil betöltése:
 - A felhasználói profilt tartalmazó állományt alapértelmezésben az aktuális könyvtárban keresi a program `.Rprofile` néven, ha ott nincs ilyen, akkor a felhasználó HOME könyvtárában.
 - A felhasználói profil a `source()` függvénnnyel kerül beolvasásra és végrehajtásra.
 - R kódot tartalmaz, tipikusan beállításokat.

Inicializálás az R indításakor (3)

- Felhasználói profil betöltése (folytatás):
 - Példa felhasználói profilra (Linux):

```
options(browser="/usr/bin/firefox")
options(editor="/usr/bin/gedit")
options(digits=5, help.try.all.packages=TRUE, help_type="html")
.First <- function() {
  cat("\nWelcome, ", Sys.getenv("LOGNAME"), "!\n", sep="")
}
```

Inicializálás az R indításakor (4)

- A előző munkamenet végén elmentett munkaterület betöltése:
 - Ha az aktuális könyvtár tartalmaz egy `.RData` nevű állományt, akkor ennek beolvasása a `load( )` függvénnnyel.
 - Az állomány tipikusan egy korábbi R munkamenet végén, a kilépésnél automatikusan kerül létrehozásra és a felhasználó által a munkamenet során létrehozott objektumokat tárolja.

Inicializálás az R indításakor (5)

- A korábbi munkamenetekből származó parancs előzmények betöltése:
 - Az előzményeket, azaz egy korábbi munkamenet során végrehajtott parancsokat tartalmazó állomány betöltése, ha van ilyen.
 - Alapértelmezésben az aktuális könyvtárban keresi a program az állományt `.Rhistory` néven.

Kilépés (1)

- Kilépéshez használjuk a `quit()` vagy `q()` függvényhívásokat.
 - A platformtól és a grafikus felhasználói interfésztől is függhet, hogy pontosan mi történik egy munkamenet befejeződésekor.
- Használjuk a `?quit` parancsot részletes leírás megjelenítéséhez.

Kilépés (2)

- A program rá fog kérdezni, hogy elmentse-e a felhasználó munkaterületét, mely a munkamenet során a felhasználó által létrehozott objektumokat tárolja.
 - Igen válasz esetén a `save.image()` függvénnnyel az aktuális könyvtárba `.RData` néven kerül elmentésre a munkaterület.
 - Az `.RData` állományt tartalmazó könyvtárból elindítva újra a programot automatikusan beolvasásra kerül az inicializálás során az állomány, így ott folytathatjuk a munkát, ahol az előző kilépésnél abbahagytuk.

Interaktív parancsvégrehajtás (1)

- Vegyük figyelembe, hogy az R megkülönbözteti a kis- és nagybetűket!
- Egy sorba akár több parancsot is gépelhetünk ' ; ' karakterekkel elválasztva.
- Parancs gépelése közben a parancsot alkotó tokenek között elhelyezhető új sor karakter, így a parancs több sorba tördelhető.
 - A végrehajtás a teljes parancs beolvasása után történik csak meg.

Interaktív parancsvégrehajtás (2)

- A konzolon begépelte parancsok maximális hossza 4095 byte lehet (nem karakter!).
- A legtöbb platformon a lefelé és felfelé nyíl billentyűkkel lehet a korábban végrehajtott parancsokat visszahívni.
- Rendelkezésre állhat automatikus parancskiegészítés funkció is, melyet a TAB billentyűvel lehet elérni.

Állományokban tárolt R kód végrehajtása

- Állományokban tárolt R kód beolvasásához és végrehajtásához használjuk a `source( file )` függvényhívást.
 - Nem csupán állománynév adható meg, hanem akár URI is (`http`, `https`, `ftp` és `file` URI-k támogatottak).
 - Ha az R kód szintaktikai hibá(ka)t tartalmaz, akkor egyáltalán nem történik végrehajtás.
 - A végrehajtás során nem jelenik meg a kiértékelt kifejezések értéke, ezért a kódban megfelelően gondoskodjunk a megjelenítésről.
 - Ha megadjuk a `chdir=TRUE` argumentumot, akkor állomány elérési útvonal használata esetén a végrehajtás során ideiglenesen az állományt tartalmazó könyvtár lesz az aktuális könyvtár.
 - Van néhány további hasznos argumentum, ezeket lásd a dokumentációban.

Dokumentáció (1)

- Az Rd („*R documentation*”) az R saját dokumentációs formátuma.
 - Egy egyszerű, a LaTeX-hez hasonló jelölőnyelv.
 - Az ebben megírt dokumentációt több különböző formátumba lehet alakítani (LaTeX, HTML, sima szöveg).
 - Az R disztribúció több mint 1300 ilyen dokumentációs állományt tartalmaz.
 - Az Rd állományok feldolgozásával kapcsolatos dokumentáció megjelenítéséhez használjuk a `?Rdconv` parancsot.

Dokumentáció (2)

- A csomagok dokumentációját ugyan tipikusan Rd formátumban írják, de tetszőleges egyéb dokumentációs formátum is használható.
 - Ajánlott a platformfüggetlen PDF formátum használata.
- Egy speciális dokumentációs forma a vignetta.
 - A vignetták olyan PDF vagy HTML dokumentumok, melyek szöveget és R kódot vegyesen tartalmazó sima szöveges állományokból kerülnek létrehozásra.
 - Vignetta motorok állítják elő őket, az alapértelmezésben használt motor az Sweave.

Segítség

- A dokumentáció eléréséhez, kereséséhez az alábbi függvényeket használhatjuk:
 - `help()`
 - `help.start()`
 - `help.search()`
 - `browseVignettes()`
 - `vignette()`
 - `RsiteSearch()`
- További információk: <https://www.r-project.org/help.html>

A help függvény (1)

- A sűgó oldalak `help(téma)` függvényhívásokkal érhető el.
 - A *téma* argumentum lehet azonosító és karakterlánc is.
 - Operátorokat, függvényekhez és vezérlési szerkezetekhez kapcsolódó kulcsszavakat (`break`, `else`, `for`, `function`, `if`, `in`, `next`, `repeat`, `while`) karakterláncként kell megadni.
 - Vannak további argumentumok is, ezeket lásd a dokumentációban.

A help függvény (2)

- A `help_type` argumentumban lehet megadni a dokumentáció megjelenítési formátumát.
 - A lehetséges értékek `"text"`, `"html"` és `"pdf"`.
 - Az utóbbi esetén egy PDF állomány jön létre az állományrendszerben, melynek előállításához szükséges a `pdflatex` parancs rendelkezésre állása.

A help függvény (3)

- A témához tartozó sugó oldalak keresése alapértelmezésben a keresési útvonalban lévő (betöltött) csomagokban történik.
 - Megadható a `package` argumentum értékeként egy csomagnév, mint azonosító vagy egy csomagneveket tartalmazó karakter vektor.
 - Így az adott csomag(ok)ra korlátozható a keresés.

A help függvény (4)

- Ha megadjuk a `try.all.packages=TRUE` argumentumot, de nem adjuk meg a keresést az adott csomagokra korlátozó `package` argumentumot, akkor az összes telepített és rendelkezésre álló csomagban történik keresés, ha a keresési útvonalban lévő csomagoknál sikertelen volt.
 - Az így megtalált súgó oldalak nem kerülnek megjelenítésre, csupán az jelenik meg, hogy mely csomagokban talált a program a témához súgó oldalakat.

A help függvény (5)

- A `help(téma)` függvényhívás helyett használható a `?téma` rövid alak.
- Hasonlóan használható a `help(téma, csomag)` függvényhívás helyett a `?csomag::téma` rövid alak.
 - Ilyenkor `csomag` csak karakterlánc vagy azonosító lehet.

A help függvény (5)

- Példák a függvény használatára:

```
help("if")  
?"if"  
help()  
help(source)  
?source  
help(package="e1071")  
help(permutations, "e1071")  
?e1071::permutations  
help(svm, try.all.packages=TRUE)
```

A `help.start` függvény

- A `help.start()` függvényhívással érhetjük el a dokumentációt HTML-ben.
- Hatására egy olyan oldal jelenik meg egy böngészőablakban, ahonnan elérhető többek között valamennyi R kézikönyv és egy keresőmotor a dokumentáció kereséséhez.
 - A használt böngésző megadható a `browser` opcióval.

A help.search függvény (1)

- A `help.search(minta)` függvényhívással kereshető a dokumentáció.
 - A *minta* argumentum kötelezően egy karakterlánc.
 - A további argumentumokat lásd a dokumentációban.
 - A keresés alapértelmezésben az összes telepített és rendelkezésre álló csomagban történik.
 - A keresés az dokumentációs oldalak különböző metaadat mezőiben történhet (például állománynév, cím, kulcsszavak).
 - A kis- és nagybetűk alapértelmezésben nincsenek megkülönböztetve a keresésnél.
 - A keresés során fuzzy mintaillesztés vagy reguláris kifejezés illesztés történik.
 - A keresés eredményeként a dokumentációs oldalak neve és címe jelenik meg.

A `help.search` függvény (2)

- A `help.search(minta)` függvényhívás helyett használható a `??minta` rövid alak.
- Hasonlóan használható a `help.search(minta, package=csomag)` függvényhívás helyett a `??csomag::minta` rövid alak.
 - Ilyenkor *csomag* csak karakterlánc vagy azonosító lehet.

A help.search függvény (3)

- Példák a függvény használatára:

```
help.search("normal distribution")  
help.search(keyword="optimize")  
help.search(keyword="IO")  
help.search("^read", fields="name",  
            ignore.case=FALSE)
```

A `help.search` függvény (4)

- Az RStudio-ban az R 3.6.0 verzió használata esetén jelenleg nem működik a `help.search()` függvény.
 - Lásd: <https://github.com/rstudio/rstudio/issues/4819>
 - A hiba kijavításáig az RStudio-ban használjuk a *Help* panel keresőmezőjét a dokumentáció kereséséhez.

A vignette függvény (1)

- A `vignette(téma)` függvényhívással lehet megjeleníteni az adott vignettát.
 - A *téma* argumentum egy karakterlánc lehet, ez a vignetta lesz megjelenítve.
 - A `package` argumentum értéke egy olyan karakter vektor lehet, melyben felsoroljuk azoknak a csomagoknak a nevét, melyekben a vignetták keresése történik.
 - Alapértelmezésben az összes telepített csomagban történik a keresés.
 - A függvényt meghívhatjuk argumentum nélkül is, ekkor a rendelkezésre álló vignettákat listázza.

A vignette függvény (2)

- Példák a függvény használatára:

```
vignette()  
vignette(package="data.table")  
vignette("datatable-intro")  
v <- vignette("datatable-intro")  
edit(v)
```

A browseVignettes függvény

- A `browseVignettes()` függvényhívás a rendelkezésre álló vignetták listáját jeleníti meg egy böngészőablakban.

Az RsiteSearch függvény (1)

- Az `RSiteSearch(x)` függvényhívással a <http://search.r-project.org/> címen működő keresőmotort érhetjük el.
 - Argumentumként egy karakterláncban adhatjuk meg a keresendő kifejezés(ek)e)t.
 - Ha egy több szóból álló kifejezésre akarunk pontosan keresni, akkor azt ' { ' és ' } ' karakterek között adjuk meg a karakterláncban.
 - Vannak további argumentumok, ezeket lásd a dokumentációban.
 - A keresés a nézetekben, súgó oldalakban és vignettákban történhet.
 - A találatok egy böngészőablakban jelennek meg.

Az RsiteSearch függvény (2)

- Példák a függvény használatára:

```
RSiteSearch("eigenvalue or eigenvector")  
RSiteSearch("{hidden Markov model}")
```

Az example függvény

- Az `example(téma)` függvényhívással lehet végrehajtani a `help(téma)` módon elérhető dokumentációs oldalak `Examples`: cím alatt feltüntetett R kódjait.
 - Ezek a függvények használatára adnak példákat.
- Az úgynevezett *dontrun* részek nem kerülnek végrehajtásra, melyeket a dokumentációban "## Not run:" szöveg jelez.
- A kód végrehajtása alapértelmezésben a felhasználó munkaterületén történik, így a végrehajtás során létrehozott objektumok a végrehajtás után is rendelkezésre állnak.
 - Ez felülbírálnak a `local=TRUE` argumentum megadásával.

Feladatok

- Hogyan generálhatók permutációk?
- Hogyan generálható Poisson-eloszlásból származó minta?
- Hogyan végezhető főkomponens-analízis?

Csomagok keresése

- Csomagok keresésére szolgál a `pkgsearch` csomag.
 - Példa a csomag használatára:

```
library(pkgsearch)

ps("reshape")
ps()                # az előbbi találatok részletesebb leírása

ps("gaussian")     # az első 10 találat
more()             # a következő 10 találat
more()             # a következő 10 találat
```

Csomagok telepítésének helye (1)

- A `.Library` globális változó értéke egy karakterlánc, mely annak a könyvtárnak az elérési útvonalát tartalmazza, amelyben alapértelmezésben történik a csomagok keresése.
 - Tipikusan rendszergazdai jogosultság birtokában lehet csupán ebbe a könyvtárba csomagokat telepíteni.
- Csomagok telepítése történhet az állományrendszerben ettől eltérő tetszőleges olyan könyvtár(ak)ba, mely(ek)re a felhasználó az ehhez szükséges jogosultságokkal rendelkezik.
 - Megfelelő beállítás szükséges ahhoz, hogy ezekben a könyvtárakban is történjen a csomagok keresése (lásd a `.libPaths()` függvényél leírtakat).

Csomagok telepítésének helye (2)

- A `.libPaths(x)` függvényhívás használható a telepített csomagokat tartalmazó könyvtárak megadására.
 - Az argumentum egy könyvtár elérési útvonalakat tartalmazó karakter vektor lehet, mely azt adja meg, hogy a továbbiakban az alapértelmezett hely mellett mely könyvtárakban történjen még a csomagok keresése.
 - A függvény meghívható argumentum nélkül is, ekkor egy karakter vektorban adja vissza azoknak a könyvtáraknak az elérési útvonalait, melyekben történik a csomagok keresése.

Csomagok telepítése (1)

- Az `install.packages(csomagok)` függvényhívással lehet csomagokat telepíteni.
 - A telepítés történhet tárolókból vagy lokális állományokból.
 - Az argumentum a telepítendő csomagok neveit/állomány elérési útvonalait tartalmazó karakter vektor.
 - Vannak további argumentumok is, ezeket lásd a dokumentációban!
 - Grafikus felhasználói felület rendelkezésre állása esetén a függvényt argumentum nélkül is meghívhatjuk, ekkor egy listából interaktív módon lehet kiválasztani a telepítendő csomagokat.

Csomagok telepítése (2)

- A telepítés alapértelmezésben egy olyan könyvtárba történik, melynek írásához tipikusan rendszergazdai jogosultság szükséges.
 - Ez felülbíráható a `lib` argumentum megadásával, melynek értéke egy könyvtár elérési útvonalat tartalmazó karakterlánc lehet.
 - Szükség esetén a függvény interaktív módon ajánlja fel egy alkalmas célkönyvtár létrehozását a felhasználó HOME könyvtárában.

Csomagok telepítése (3)

- A függvény automatikusan letölti a csomagok telepítéséhez szükséges további csomagokat.
- Az `install.packages()` függvényhívás előtt érdemes az `update.packages()` függvényhívással frissíteni a telepített csomagokat.
 - Ez biztosítja, hogy a szükséges függőségek a legfrissebb verzióban álljanak rendelkezésre.

Csomagok telepítése (4)

- További függvények csomagok kezeléséhez:
 - **`installed.packages()`**: a lokálisan telepített csomagokat szolgáltatja.
 - **`available.packages()`**: az elérhető telepíthető csomagokat szolgáltatja.
 - **`old.packages()`**: azokat a csomagokat szolgáltatja, melyeknek újabb verziója érhető el a lokálisan telepítetttnél.
 - **`new.packages()`**: az elavult és az elérhető, de lokálisan nem telepített csomagokat szolgáltatja.
 - **`update.packages()`**: a lokálisan telepített csomagokat frissíti az elérhető legfrissebb verzióra.

Csomagok telepítése (5)

- Nézetek telepítésére szolgál a `ctv` csomag, melynek főbb függvényei:
 - **`available.views()`**: a rendelkezésre álló nézeteket listázza.
 - **`install.views()`**: az adott nézetbe tartozó csomagokat telepíti.
 - Például az `install.views("MachineLearning")` függvényhívás a gépi tanuláshoz kötődő csomagokat telepíti.

Csomagok telepítése (6)

- Csomagok telepítése adott tároló(k)ból:
 - A rendelkezésre álló csomagokat listázó, telepítő és frissítő függvényeknek a `repos` argumentumban adható(k) meg a használandó tároló(k) URL-je(i).
 - Használható a `repos` nevű globális opció is.
 - Példa: a Microsoft CRAN tárolójának használata (hasznos lehet tűzfal mögött)

```
install.packages(csomagok,  
  repos = "https://cran.microsoft.com/")
```

```
install.packages(csomagok,  
  repos = "https://cran.microsoft.com/snapshot/éééé-hh-nn")
```

Csomagok betöltése (1)

- Csomagok betöltésére szolgálnak a `library(csomag)` és `require(csomag)` függvényhívások.
 - Az argumentum a csomag nevét megadó azonosító vagy karakterlánc.
 - Az elsőt tipikusan interaktív parancsvégrehajtásnál, a másodikat pedig függvények törzsében használjuk.

Csomagok betöltése (2)

- A `library` függvény:
 - Meghívható argumentum nélkül, ilyenkor a telepített csomagokat listázza.
 - Meghívható `library(help=csomag)` módon, ahol *csomag* a csomag nevét megadó azonosító vagy karakterlánc.
 - Ilyenkor a csomagról szolgáltat információkat (metaadatok, tartalom).
 - A `lib.loc` argumentum értéke egy könyvtár elérési útvonalakat tartalmazó karakter vektor lehet.
 - Az adott könyvtárakban történik a csomag keresése.
 - Lásd a `.libPaths()` függvénynél leírtakat.

Csomagok betöltése (3)

- A `require` függvény:
 - Logikai visszatérési értékű.
 - A visszaadott érték `TRUE` akkor, ha az argumentumként adott csomag lokálisan elérhető, egyébként pedig `FALSE`.
 - Nem telepített csomag megadása a `library()` függvénytől eltérően nem okoz hibát, csak figyelmeztetést.
 - A `lib.loc` argumentum ugyanúgy használható, mint a `library()` függvénynél.

Csomagok leválasztása

- Betöltött csomagok leválasztásához használjuk a `detach(x)` függvényhívást.
 - Az argumentumot `package : név` vagy `"package : név"` módon adhatjuk meg, ahol *név* a leválasztandó csomag neve.

Csomagok használata

- Példa:

```
p <- available.packages()
invisible(edit(p))
install.packages("fortunes", lib="/tmp")
library(help="fortunes", lib.loc="/tmp")
library(fortunes, lib.loc="/tmp")
fortune()
# ...
detach("package:fortunes")
.libPaths()
.libPaths("/tmp")
.libPaths()
library(fortunes)
fortune()
remove.packages("fortunes", "/tmp")
```

Opciók (1)

- Az R globális beállításai.
- A használható opciókat megtaláljuk az `?options` paranccsal megjeleníthető súgó oldalon.
- Opciók kezelésére szolgálnak az `options()` és `getOptions()` függvények.

Opciók (2)

- Az `options( . . . )` függvényhívás segítségével lehet lekérdezni és beállítani az opciókat.
 - A függvény meghívható argumentum nélkül, ekkor az összes beállított opciót adja vissza egy listában.
 - A listában az opciók név szerint ábécé sorrendbe rendezettek.
 - Opciók beállításához tetszőleges számú *név = érték* alakú kifejezést adhatunk meg argumentumként, vagy pedig egy ilyen komponensekből álló listát.
 - *név* az opció nevét megadó azonosító vagy karakterlánc. Tetszőleges név megadható, hatása csak az R által kezeltnek lesz.
 - *érték* az opciót értéket szolgáltató kifejezés. Az R által kezelt opciók értékére ellenőrzés történhet (nem megengedett érték hibát okoz).

Opciók (3)

- A `getOption(opció)` függvényhívás az argumentumként adott nevű opció értékét szolgáltatja.
 - Az argumentum kötelezően egy karakterlánc.
 - Be nem állított opció esetén `NULL` a visszaadott érték.

Opciók (4)

- Példa opciók használatára:

```
options()  
getOption("digits")  
options(digits=getOption("digits") + 1)  
oldOptions <- options()  
options(help.try.all.packages=TRUE, verbose=TRUE)  
options(editor="/usr/bin/nano")  
options(pdfviewer="/usr/bin/evince")  
# ...  
options(oldOptions) # a korábbi beállítások  
                    # visszaállítása
```

Aktuális könyvtár

- Az aktuális könyvtár lekérdezéséhez használjuk a `getwd ( )` függvényhívást.
 - A visszatérési érték az aktuális könyvtár elérési útvonalát tartalmazó karakterlánc, illetve NULL akkor, ha nem áll rendelkezésre az aktuális könyvtár.
- Az aktuális könyvtárat a `setwd (könyvtár)` függvényhívással lehet beállítani.
 - Az argumentum értékeként az elérési útvonalat tartalmazó karakterláncot kell megadni.