

Adatok importálása R-ben

Jeszenszky Péter
Debreceni Egyetem, Informatikai Kar
jeszenszky.peter@inf.unideb.hu

Utolsó módosítás: 2019. június 11.



Telepítendő csomagok

- A példák kipróbálásához telepítendő csomagok:
 - `reshape2`
 - `splitstackshape`
 - `tidyverse`
 - `data.table`
 - `R.utils` (a `data.table` csomag használja `gz` és `bz2` állományok olvasásához)
 - `XML`

Objektumok tárolása állományokban

- Az R által a memóriában kezelt objektumokat állományokba lehet menteni.
 - A `save()` függvény az argumentumként adott objektumokat menti.
 - A `save.image()` függvény valamennyi a memóriában tárolt objektumot menti.
- Az így elmentett állományokat a `load()` függvénnyel lehet a memóriába betölteni.
- A tárolás alapértelmezésben az R saját bináris formátumában történik.
 - Az `ascii = TRUE` argumentum megadásával szöveges állományba történik a mentés.

URL-ek használata

- Az állományokat beolvasó számos függvény, mint például a `read.table()` vagy a `data.table` csomag `fread()` függvénye, állományelérési útvonalak helyett URL-eket is elfogad argumentumként.
- A `download.file()` függvény állományok letöltésére szolgál.
 - Ha nem elegendőek a függvény lehetőségei, használjuk a `httr` csomagot.

Tömörített állományok (1)

- Tömörített állományok az állományrendszerből történő beolvasását teszik lehetővé a `base` csomag `bzfile()`, `gzfile()`, `xzfile()`, és `unz()` függvényei.
- További információk: `?connections`

Tömörített állományok (2)

- Példa:

```
read.csv(bzfile("file.csv.bz2"))  
read.csv(gzfile("file.csv.gz"))  
read.csv(xzfile("file.csv.xz"))  
read.csv(unz("file.zip", "file.csv"))
```

a fentiekkel ekvivalensek az alábbiak is:

```
read.csv("file.csv.bz2")  
read.csv("file.csv.gz")  
read.csv("file.csv.xz")
```

Tömörített állományok (3)

- A `data.table` csomag `fread()` függvénye URL-ről is be tud olvasni `gz` és `bz2` tömörített állományt.
 - Ehhez az `R.utils` csomag rendelkezésre állása szükséges.
 - Példa:

```
fread("http://example.com/file.csv.gz")  
fread("http://example.com/file.csv.bz2")
```

Formátumok

- CSV (TSV)
- MS Excel
- JSON
- XML

CSV (1)

- A CSV (TSV) állományok jelentik a legbiztonságosabb módját a más szoftverekkel állományok révén történő kommunikációnak.
 - Ilyen állományokat minden szoftver tud írni és olvasni.

CSV (2)

- Táblázatos adatok szöveges állományokból történő beolvasására szolgálnak az alábbi függvények:
 - `read.table()`
 - `read.csv()` (mezőelválasztó: `" , "`, decimális elválasztó: `" . "`)
 - `read.csv2()` (mezőelválasztó: `" ; "`, decimális elválasztó: `" , "`)
 - `read.delim()` (mezőelválasztó: `"\t"`, decimális elválasztó: `" . "`)
 - `read.delim2()` (mezőelválasztó: `"\t"`, decimális elválasztó: `" , "`)
- A függvények egy adatkeretet adnak vissza.
- A `read.csv()`, `read.csv2()`, `read.delim()` és `read.delim2()` függvények a `read.table()` függvényt hívják meg.
- További információk: `?read.table`

CSV (3)

- Táblázatos adatok (adatkeretek és mátrixok) szöveges állományokba történő kiírására szolgálnak az alábbi függvények:
 - `write.table()`
 - `write.csv()` (mezőelválasztó: `" , "`, decimális elválasztó: `" . "`)
 - `write.csv2()` (mezőelválasztó: `" ; "`, decimális elválasztó: `" , "`)
- A `write.csv()` és `write.csv2()` függvények a `write.table()` függvényt hívják meg.
- További információk: `?write.table`

CSV (4)

- Példa:
 - Az adatok forrása: *United States Geological Survey*
 - *Earthquake Hazards Program*
- <https://earthquake.usgs.gov/earthquakes/feed/v1.0/csv.php>

```
library(lubridate) # az ymd_hms( ) függvény használatához szükséges

X <- read.csv("https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary",
  row.names = "id",
  as.is = c("time", "updated"))

X <- transform(X, time = ymd_hms(time), updated = ymd_hms(updated))
str(X)
summary(X)
```

CSV (5)

- A `read.table()` függvény nem alkalmas túl nagy adatállományok beolvasásához!
- Példa:
 - Az adatok forrása: *IMDb Datasets*
<https://www.imdb.com/interfaces/>
 - Az alábbi adatállomány mérete tömörítve kb. 100 MB, összesen közel 6 millió sort tartalmaz.

```
tmp <- tempfile(fileext = ".gz")
download.file("https://datasets.imdbws.com/title.basics.tsv.gz", tmp)

R.utils::hsize(file.size(tmp), units = "MB") # "100.8 MB"

X <- read.delim(tmp, header = TRUE, na.strings = "\\N", nrows = 1000)
```

CVS (6)

- Ajánlott csomag nagy adatállományhoz:
`data.table`
 - További információk:
<https://github.com/Rdatatable/data.table/wiki>

CSV (7)

- Példa:
 - Az adatok forrása: *IMDb Datasets*
<https://www.imdb.com/interfaces/>

```
library(data.table)

tmp <- tempfile(fileext = ".gz")
download.file("https://datasets.imdbws.com/title.basics.tsv.gz", tmp)

X <- fread(tmp, sep = "\t", na.strings = "\\N", quote = "",
           verbose = TRUE)

class(X)
dim(X)
summary(X)
```

CVS (8)

- RStudio:
 - CSV (TSV) állományok importálását segítik a **File** → **Import Dataset** → **From Text...** menüpontok.
 - A `util` és a `readr` csomagok függvényei kerülnek meghívásra megfelelően felparaméterezve, melyhez a beállításokat a felhasználó szolgáltatja párbeszédablakokban.

MS Excel (1)

- Ajánlott csomag: `readxl`
 - Automatikusan telepítésre kerül a `tidyverse` csomagokkal együtt.
 - Probléma: nem adható meg URL argumentumként

MS Excel (2)

- Példa:
 - Az adatok forrása: *World Bank Open Data – Population, total*
<https://data.worldbank.org/indicator/sp.pop.totl>

```
library(readxl)

tmp <- tempfile()
download.file("http://api.worldbank.org/v2/en/indicat

X <- read_excel(tmp, skip = 3)
```

MS Excel (3)

- RStudio:
 - MS Excel állományok importálását segíti a **File** → **Import Dataset** → **From Excel...** menüpont.
 - A readxl csomag read_excel() függvénye kerül meghívásra megfelelően felparaméterezve, melyhez a beállításokat a felhasználó szolgáltatja egy párbeszédablakban.

JSON (1)

- Ajánlott csomag: `jsonlite`
 - Automatikusan telepítésre kerül a `tidyverse` csomagokkal együtt.

JSON (2)

- Példa:

```
library(jsonlite)

json <- toJSON(iris)
print(json)
json <- toJSON(iris, pretty = TRUE)
print(json)
write(json, "iris.json")

X <- fromJSON(toJSON(iris))
isTRUE(all.equal(iris, X)) # FALSE
X <- transform(X, Species = factor(Species))
isTRUE(all.equal(iris, X)) # FALSE
identical(iris, X)        # FALSE
```

JSON (3)

- Példa:
 - Az adatok forrása:
<https://github.com/r-hub/cranlogs.app>

```
library(jsonlite)

X <- fromJSON("https://cranlogs.r-pkg.org/top/last-month/10")

str(X)
downloads <- transform(X$downloads,
  downloads = as.integer(downloads))
str(downloads)
summary(downloads)
```

XML (1)

- Javasolt csomag: XML

XML (2)

- Példa:
 - Az adatok forrása: *World Bank Indicators API*
<https://datahelpdesk.worldbank.org/knowledgebase/articles/889392-about-the-indicators-api-documentation>

```
library(XML)

doc <- xmlParse("http://api.worldbank.org/v2/country/")

root <- xmlRoot(doc)
countries <- xmlToDataFrame(doc)
```

XML (3)

- Példa (folytatás):

```
library(magrittr)

xmlName(root)    # a gyökérelem neve
xmlSize(root)    # a gyökérelem gyermekeinek száma
root[[1]]        # a gyökérelem első gyermeke
root[[1]][[2]]   # a gyökérelem első gyermekének második
                  # gyermeke
root[[1]][[2]] %>% xmlValue()

root[[1]] %>% xmlAttrs() # a gyökérelem első
                        # gyermekének attribútumai
root[[1]] %>% xmlAttrs() #> ["id"]
```

XML (4)

- Példa (folytatás):

```
xpathSApply(root, "//wb:iso2Code", xmlValue)
```

További ajánlott olvasnivaló

- *R Data Import/Export*
<https://cran.r-project.org/doc/manuals/R-data.html>
- Karlijn Willems. *This R Data Import Tutorial Is Everything You Need*. November 20th, 2018.
<https://www.datacamp.com/community/tutorials/r-data-import-tutorial>
- Karlijn Willems. *Importing Data Into R – Part Two*. March 18th, 2019.
<https://www.datacamp.com/community/tutorials/importing-data-r-part-two>