

VI. Lista típusú objektumok

Előfordulhat, hogy egy objektumban kezelni kívánunk különböző dimenziójú vagy típusú adatokat. Tegyük fel, hogy eredményül n darab, különböző sor- és oszlopszámú mátrixot kapunk. Ezeket csak akkor tudjuk egy tömbbe (`array`) menteni, ha alapul vesszük a legnagyobb mátrixot és az összes többiben feltöltjük a hiányzó elemeket, például NA értékkel. Így azonban feleslegesen nagy tömböt tárolunk, továbbá nincs garancia arra, hogy később nem fordul elő a korábbiaknál nagyobb, tárolandó mátrix.

Az R-ben ilyen esetekben célszerű lista (`list`) típusú objektumokat használni. Ezekben azonos típusú, eltérő dimenziójú, illetve eltérő típusú adatok is tárolhatók. Például több NetCDF állomány metaadatait is lista típusú objektumban tudjuk tárolni.

A lista részlistákból épül fel, amelyek két szögletes zárójelpárral hivatkozhatók.

Tekintsünk egy példát!

Hozzunk létre két, négy és három elemű részlistákból álló lista típusú objektumot!

- Az első részlista legyen karakterlánc és álljon a következő elemekből: "alma", "korte".
- A második részlista álljon az 1, 2, 3, 4 természetes számokból.
- A harmadik részlista pedig legyen logikai értékek sorozata: TRUE, FALSE, TRUE

Lista a `list` függvénnyel hozható létre a következőképpen:

```
lista <- list(list("alma", "korte"),
             list(1, 2, 3, 4),
             list(TRUE, FALSE, TRUE))
```

Hivatkozás a második részlistára és annak első elemére:

```
lista[[2]]
[[1]]
[1] 1
:
[[4]]
[1] 4
lista[[2]][1] # vagy
lista[[2]][[1]]
```

A részlisták azonban karakterként kerülnek tárolásra, ezért ahhoz, hogy számolni tudjunk a második részlista értékeivel, használnunk kell az `unlist` függvényt.

Az alábbi utasítás tehát hibát ad:

```
lista[[2]][1] + lista[[2]][2]
Error in lista[[2]][1] + lista[[2]][2] :
  non-numeric argument to binary operator
```

Helyes eredményt az alábbi utasítás ad:

```
unlist(lista[[2]][1]) + unlist(lista[[2]][2])
[1] 3
```

Ha a második részlistát a `list(1:4)` kóddal hoztuk létre, akkor a részlista egy vektor. Ennek komponensei a lista részlistájának részlistájaként hivatkozhatók, amelynek típusa integer, vagyis műveletek elvégzéséhez nincs szükség az `unlist` függvényre. Például az első komponens:

```
lista[[2]][[1]][1]
[1] 1
```

Megjegyzés: Ha a listát `for` ciklusban töltjük fel értékekkel, akkor a ciklus előtt létre kell hoznunk a feltöltendő üres listát:

```
lista <- list()
for (...) {
  kifejezés }
```