

## IV. NetCDF állományok adatainak kezelése

### TARTALOM:

1. [Állomány beolvasása](#)
2. [Adatok kijelölése](#)
  - 2.1. [Adott időpontra és rácspontra vonatkozóan](#)
  - 2.2. [Adott rácspontra vonatkozóan](#)
  - 2.3. [Adott időpontra vonatkozóan](#)
3. [Felhasznált irodalom és csomagok](#)

### 1. ÁLLOMÁNY BEOLVASÁSA:

A meteorológiában gyakran tároljuk az adatokat kettőnél több dimenziós tömbökben, hiszen nemcsak időben, hanem térben rendelkezésünkre álló adatokat is vizsgálunk.

#### **Olvassuk be az előző fejezetben létrehozott `proba.nc` állomány adatait<sup>1</sup>!**

Adjuk meg az állomány elérési útját a következőképpen!

```
file="meghajtóazonosító:\\mappanév\\almappanév\\proba.nc"
```

Nyissuk meg az `ncdf4` csomagot!

```
library(ncdf4)
```

#### **Olvassuk be a NetCDF állományt az `nc_open` függvénnyel és tároljuk objektumként!**

```
proba <- nc_open(file)
```

#### **Jelenítsük meg a NetCDF állomány alapvető információit!**

```
proba
```

```
File C:\Users\user\Documents\proba.nc (NC_FORMAT_CLASSIC):  
  1 variables (excluding dimension variables):  
    double var3d[elsodim,masodikdim,harmadikdim]  
      units: units  
      _FillValue: 15.6007  
      long_name: a változo hosszu neve  
  3 dimensions:  
    elsodim Size:31  
      units: degrees  
      long_name: földrajzi szélesség  
    masodikdim Size:144  
      units: degrees  
      long_name: földrajzi hosszúság  
    harmadikdim Size:2  
      units: day  
      long_name: a mintavételezés időpontja
```

Emlékeztetőül: A `proba ncdf4` típusú objektum, amely a háromdimenziós tömb adatait és legfőbb tulajdonságait tartalmazza. Az állomány elérési útját követő sor alapján egy,

---

<sup>1</sup> Az állomány [innen](#) tölthető le.

mértékegység nélküli, háromdimenziós változó áll rendelkezésünkre double típusú számformátumban, amely var3d néven hivatkozható. A három dimenzió elsodim, masodikdim és harmadikdim néven hivatkozható. A dimenziók a hosszú neveik alapján a földrajzi szélesség [fok], a földrajzi hosszúság [fok] és a mintavételezés időpontja [nap], amelyekből rendre 31, 144 és 2 darab áll rendelkezésre. A hiányzó érték 15,6007. A dimenziókból következően a változó éppen  $31 \times 144 \times 2 = 8928$  értéket tartalmaz.

**A dimenziók és a változók értékeinek tárolása objektumokként:**

```
dim1 <- ncvar_get(proba, varid="elsodim", start=1, count=31)
dim2 <- ncvar_get(proba, varid="masodikdim", start=1, count=144)
dim3 <- ncvar_get(proba, varid="harmadikdim", start=1, count=2)
valtozo <- ncvar_get(proba, varid="var3d", start=c(1,1,1),
count=c(31,144,2))
```

**Próbaképpen jelenítsük meg a dimenziók indexeit!**

```
dim1
15.0 17.5 20.0 22.5 25.0 27.5 30.0 32.5 35.0 37.5 40.0 42.5 45.0 47.5 50.0
52.5 55.0 57.5 60.0 62.5 65.0 67.5 70.0 72.5 75.0 77.5 80.0 82.5 85.0 87.5
90
```

```
dim2
-180.0 -177.5 -175.0 -172.5 -170.0 -167.5 -165.0 -162.5 -160.0 -157.5
-155.0 -152.5 -150.0 -147.5 -145.0 -142.5 -140.0 -137.5 -135.0 -132.5
-130.0 -127.5 -125.0 -122.5 -120.0 -117.5 -115.0 -112.5 -110.0 -107.5
-105.0 -102.5 -100.0 -97.5 -95.0 -92.5 -90.0 -87.5 -85.0 -82.5
-80.0 -77.5 -75.0 -72.5 -70.0 -67.5 -65.0 -62.5 -60.0 -57.5
-55.0 -52.5 -50.0 -47.5 -45.0 -42.5 -40.0 -37.5 -35.0 -32.5
-30.0 -27.5 -25.0 -22.5 -20.0 -17.5 -15.0 -12.5 -10.0 -7.5
-5.0 -2.5 0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
20.0 22.5 25.0 27.5 30.0 32.5 35.0 37.5 40.0 42.5
45.0 47.5 50.0 52.5 55.0 57.5 60.0 62.5 65.0 67.5
70.0 72.5 75.0 77.5 80.0 82.5 85.0 87.5 90.0 92.5
95.0 97.5 100.0 102.5 105.0 107.5 110.0 112.5 115.0 117.5
120.0 122.5 125.0 127.5 130.0 132.5 135.0 137.5 140.0 142.5
145.0 147.5 150.0 152.5 155.0 157.5 160.0 162.5 165.0 167.5
170.0 172.5 175.0 177.5
```

```
dim3
1 2
```

## 2. ADATOK KIJEJELŐLÉSE:

### 2.1. Adott időpontra és rácspontra vonatkozóan:

Keressük ki a változó Budapest földrajzi koordinátáira és a második időpontra vonatkozó értékét! Budapest földrajzi koordinátái körülbelül é. sz.  $47^\circ$  és k. h.  $19^\circ$ , amelyekhez legközelebb eső elemek a dim1 objektum 14. és a dim2 objektum 81. indexű elemei. A változó objektum első és második dimenziójaként ezek adandóak meg, a harmadik dimenzió indexe pedig 2.

```
dim1[14]
```

```
[1] 47.5
```

```
dim2[81]
```

```
[1] 20
```

```
dim3[2]
```

```
[1] 2
```

A keresett adat:

```
valtozo[14,81,2]
```

### 2.2. Adott rácspontra vonatkozóan:

*Jelöljük ki adott rácspontra az összes időlépcső adatát, azaz egy idősor adatait!*

A `proba.nc` állományban egynél több időpontra tárolunk adatokat, ezért például Budapest földrajzi koordinátáira idősort az alábbi módon tudunk megjeleníteni:

```
valtozo[14,81,]
```

vagy:

```
valtozo[14,81,1:2]
```

(Az idősorok ábrázolásának lehetőségei a 2. fejezetben találhatók.)

### 2.3. Adott időpontra vonatkozóan:

*Jelöljük ki és jelenítsük meg adott időpontra az összes rácspont adatát, azaz egy mező adatait!*

Ebben az esetben egy időpontra,  $144 \times 31 = 4464$  rácspontra kérjük le az adatokat, azaz például a második időpont esetén az alábbi adatmezőt jelenítjük meg:

```
valtozo[, , 2]
```

vagy:

```
valtozo[1:31, 1:144, 2]
```

*Jelenítsük meg az adatokat térképen!*

A táblázatba rendezett adatok bal felső (jobb alsó) eleme kerüljön az ábra bal felső (jobb alsó) sarkába.

### **Az ábrázoláshoz szükséges R kiegészítő csomagok:**

Országhatárok ábrára illesztéséhez:

```
library(maps)
library(maptools)
```

Az `image.plot` függvény alkalmazásához:

```
library(fields)
```

Színskála illesztéséhez:

```
library(RColorBrewer)
```

**Az ábrázoláshoz az `image.plot` függvényt alkalmazzuk**, ugyanis ezzel az ábraterületet két részre tudjuk osztani: a tényleges ábrára és a színskálára.

Próbaképpen ábrázoljuk az adatokat formázatlan színskálával!

A 2. *fejezet*ben megadott módon ábrázoljuk az adatokat az `image.plot` függvénnyel.

```
image.plot(dim2[1:144], dim1[1:31], t(valtozo[31:1,1:144,2]))
```

### **Adjunk az ábrához országhatárokat!**

```
data(wrld_simpl)
plot(wrld_simpl, add = TRUE)
```

### **Hozzuk létre az ábrát kék-piros átmenetű színskálával!**

Például 11 és 19 értékek között 8 intervallumba essenek az adatok. Az adatsor minimális és maximális értékeit a `min` és a `max` függvénnyel határozhatjuk meg.

```
min(valtozo[31:1,1:144,2])
max(valtozo[31:1,1:144,2])
```

Az osztásközök felvétele:

```
brks <- seq(11,19,1)
```

Az `image.plot` függvény argumentumának első elemeként az x tengelyt, a második elemeként az y tengelyt, a harmadik elemeként pedig a `valtozo` értékeit adtuk meg.

Az adatok olyan struktúrában találhatóak a `valtozo` objektumban, hogy a sorok jelentik a földrajzi szélességeket, az oszlopok pedig a földrajzi hosszúságokat. Emiatt transzponálnunk kell az adatsort. (Bővebb információért lásd a 2. *fejezetet*!)

Tiltsuk le az alapértelmezett tengelyeket az `xaxt="n"` és `yaxt="n"` paraméterekkel és az alapértelmezett tengelycímkéket az `ann=FALSE` paraméterrel.

```
image.plot(dim2[1:144], dim1[1:31], t(valtozo[31:1,1:144,2]),
col=rev(colorRampPalette(brewer.pal(8, "RdBu"))(8)),
breaks=brks, lab.breaks=brks, xaxt="n", yaxt="n", ann=FALSE)
```

A beépített `RdBu` színskálát használtuk, 8 alapértelmezett színnel. A színek sorrendjét meg kellett cserélnünk, mivel alapértelmezetten a pirosas árnyalatok jelentik az alacsonyabb hőmérsékletet. (Az adatokat úgy ábrázoltuk, hogy a 0 érték a színskálán középre essen.)

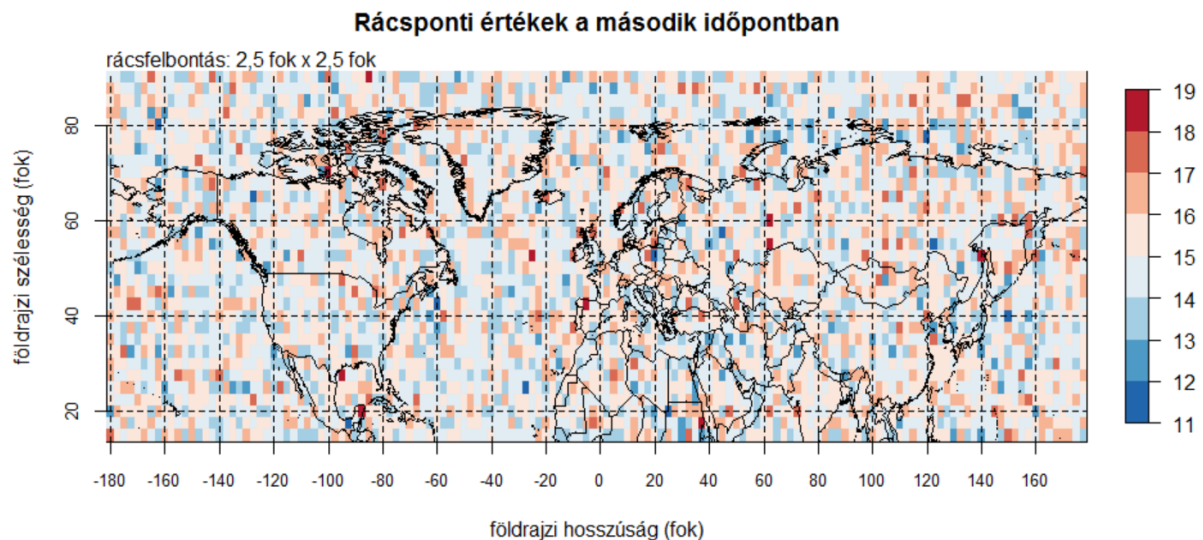
Vegyünk fel 20 földrajzi fokonként szaggatott vonallal jelölt szélességi és hosszúsági köröket!  
`abline(h=seq(20,90,20), v=seq(-180,179,20), lty=2)`

Adjunk az ábrához diagramcímet, képaláírást (balra igazítva az `adj=0` paraméterrel), tengelycímkéket és tengelyfeliratokat (utóbbi mérete legyen az alapértelmezett 0,8-szerese), majd formázzuk a tengelyeket!

```
title(main="Rácsponti értékek a második időpontban")
mtext("rácsfelbontás: 2,5 fok x 2,5 fok", adj=0)
title(xlab="földrajzi hosszúság (fok)", ylab="földrajzi
szélesség (fok)")
axis(1, at=seq(-180,179,20), labels=seq(-180,179,20),
cex.axis=0.8)
axis(2, at=seq(20,90,20), labels=seq(20,90,20), cex.axis=0.8)
```

Végül adjunk az ábrához országhatárokat!

```
data(wrld_simpl)
plot(wrld_simpl, add = TRUE)
```



### 3. FELHASZNÁLT IRODALOM ÉS CSOMAGOK:

R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

A <https://www.rdocumentation.org/> oldalon elérhető leírások a függvények paramétereiről.

David Pierce (2017). ncdf4: Interface to Unidata netCDF (Version 4 or Earlier) Format Data Files. R package version 1.16. <https://CRAN.R-project.org/package=ncdf4>

Roger Bivand and Nicholas Lewin-Koh (2017). mapproj: Tools for Reading and Handling Spatial Objects. R package version 0.9-2. <https://CRAN.R-project.org/package=mapproj>

Original S code by Richard A. Becker, Allan R. Wilks. R version by Ray Brownrigg. Enhancements by Thomas P Minka and Alex Deckmyn. (2017). maps: Draw Geographical Maps. R package version 3.2.0. <https://CRAN.R-project.org/package=maps>

Douglas Nychka, Reinhard Furrer, John Paige and Stephan Sain (2017). “fields: Tools for spatial data.” doi: 10.5065/D6W957CT (URL: <http://doi.org/10.5065/D6W957CT>), R package version 9.6, <URL: [www.image.ucar.edu/~nychka/Fields](http://www.image.ucar.edu/~nychka/Fields)>.

Erich Neuwirth (2014). RColorBrewer: ColorBrewer Palettes. R package version 1.1-2. <https://CRAN.R-project.org/package=RColorBrewer>