

III. NetCDF állományok készítése és beolvasása

TARTALOM:

1. [Bevezetés](#)
2. [NetCDF állomány létrehozatala](#)
3. [NetCDF állomány beolvasása, adatainak kezelése](#)
4. [Felhasznált irodalom és csomagok](#)

1. BEVEZETÉS:

A meteorológiában gyakran használunk kettőnél több dimenziós tömböket.

Egydimenziós esetben például egy helyszínre állnak rendelkezésre eltérő időpontbeli adatok, míg kétdimenziós esetben különböző rácspontok adatai érhetők el egy időpontban. Az előbbinél az idő az egyetlen dimenzió, utóbbinál pedig a földrajzi szélesség és a földrajzi hosszúság alkotja a két dimenziót. Előfordulhat azonban, hogy rácspontonként egy-egy idősor áll rendelkezésünkre. Az ilyen adatstruktúra háromdimenziós tömbben tárolható, amelyben a dimenziók: az idő, a földrajzi szélesség és a földrajzi hosszúság.

Négydimenziós adatstruktúrára példaként említhetők különböző magassági szintekre, rácspontokra és időpontokra elérhető geopotenciális magasság adatok. Ekkor a négy dimenzió: az idő, a földrajzi szélesség, a földrajzi hosszúság és a vertikális szint. (Négydimenziós tömbként érhetők el például a Kapcsolt modelleket összehasonlító projekt 5. fázisa¹ általános cirkulációs modelljeinek eredményei.)

Az előbbi példában egy változóra, a geopotenciális magasságra állt rendelkezésre adat. Előfordulhat azonban, hogy ugyanolyan dimenziókkal más állapotváltozókra, például a hőmérsékletre is elérhetők adatok, amelyeket ugyanebben az állományban tárolnak. Ebben az esetben az adatstruktúra tehát négydimenziós tömbben tartalmaz két változót.

A többdimenziós tömböket a meteorológiában gyakran NetCDF típusú állományban tárolják, amely az R-ben az `ncdf4` package alkalmazásával szerkeszthető.

¹ Coupled Model Intercomparison Project Phase 5, CMIP5

2. NETCDF ÁLLOMÁNY LÉTREHOZATALA:

Telepítsük az ncd4 R kiegészítő csomagot, majd nyissuk meg!

```
install.packages("ncdf4")  
library(ncdf4)
```

Hozunk létre példaképpen egy 31×144×2 tömböt, amelyet töltünk fel 8928 darab normális eloszlású, mértékegység nélküli adattal! Az eloszlás várható értéke legyen 15, a szórása pedig 1. A normális eloszlású adatsort tekintjük a mintának.

```
dataset <- array(rnorm(8928, m=15, s=1), dim=c(31,144,2))
```

A dataset tömb két darab 31 sorú és 144 oszlopú, normális eloszlású számokat tartalmazó táblázatból áll. Az első táblázat például az alábbi módon jeleníthető meg:

```
dataset[, , 1]
```

Hozzuk létre a dimenziókbeli indexek értékeit és tároljuk ezeket objektumokban!

Legyen az első dimenzió a földrajzi szélesség 15° és 90° között 2,5° fokenként és a második dimenzió a földrajzi hosszúság -180° és 177,5° között 2,5° fokenként. Az első dimenzió (sorok) tehát 1-től 31-ig, a második dimenzió (oszlopok) 1-től 144-ig természetes számokkal indexelt. Két táblázatban található az adatok, azaz a harmadik dimenzió 1 vagy 2 indexértéket vehet fel.

```
elsodim <- seq(15, 90, 2.5)  
masodikdim <- seq(-180, 177.5, 2.5)  
harmadikdim <- c(1, 2)
```

A dimenziók tulajdonságait az ncdim_def függvénnyel adhatjuk meg. Szintaxisa:

```
ncdim_def(name="a dimenzió rövid neve", units="mértékegység",  
vals=a dimenzió indexeinek számsorozata, longname="a dimenzió  
hosszú neve")
```

Megjegyzés: A NetCDF állomány beolvasásakor a name-re fogunk hivatkozni az ncvar_get függvényben.

Az első és a második dimenzió jelentse a földrajzi koordinátákat, így a mértékegységük legyen fok. A harmadik dimenzió utaljon a mintavételezés időpontjára, a mértékegysége legyen nap.

A példában: A dimenziók rövid és hosszú nevének, mértékegységének megadása, objektumokként való tárolása:

```
dimx <- ncdim_def(name="elsodim", units="degrees",  
vals=elsodim, longname="földrajzi szélesség")  
  
dimy <- ncdim_def(name="masodikdim", units="degrees",  
vals=masodikdim, longname="földrajzi hosszúság")  
  
dimz <- ncdim_def(name="harmadikdim", units="day",  
vals=harmadikdim, longname="a mintavételezés időpontja")
```

Hiányzó értékek kezelése:

Meg kell határoznunk, hogy a NetCDF állományban mivel jelöljük a hiányzó értékeket. A példában: Jelölje a hiányzó értékeket 15,60070 és ezt mentjük az mv objektumba.

```
mv <- 15.60070
```

A változó definiálása:

A változó rövid és hosszú neve, mértékegysége szintén az ncvar_def függvénnyel adható meg, továbbá felsorolandók a dimenziók a dim=list() függvényben, illetve definiálandó a missval paraméterrel a hiányzó érték. A prec paraméterrel a változó pontossága adható meg, a lehetőségek: "short", "integer", "float", "double", "char".

A példában: A változó rövid neve legyen var3d, vagyis adjuk meg ezt name paraméterként idézőjelek között. A változó értékei mértékegység nélküliek, ezért a units paraméternél szerepeljen a units kifejezés szintén idézőjelek között. A dimenziók a korábban definiált dimx, dimy, dimz objektumok, a hiányzó érték pedig mv. Dupla pontosságú lebegőpontos számot állítsunk be a prec paraméternél double-t választva, amely 15 tizedes jegy kezelésére alkalmas. Az előbb megadott tulajdonságokat tároljuk a var3d objektumban.

```
var3d <- ncvar_def(name="var3d", units="units",  
dim=list(dimx,dimy,dimz), missval=mv, longname="a változó  
hosszu neve", prec="double")
```

Megjegyzés: A var3d-re hivatkozva fogjuk megjeleníteni a dataset tömb adatait az ncvar_get függvénnyel.

A NetCDF típusú objektum és állomány létrehozatala:

Az nc_create függvény szintaxisa:

```
nc_create(filename="a létrehozandó nc kiterjesztésű állomány  
neve", vars=list(első változó, ..., n. változó))
```

Megjegyzés: Ha egynél több változónk lenne, akkor a list függvényben vesszővel elválasztva sorolhatnánk fel ezeket. A verbose=TRUE paraméterrel kiírathatók a képernyőre a NetCDF fájl készítésének információi. Alapértelmezetten ezek nem jelennek meg.

A példában: Az ncdf4 típusú objektum neve legyen nc, az állomány neve pedig proba.nc!

```
nc <- nc_create(filename="proba.nc", vars=list(var3d))
```

A függvénnyel nemcsak az nc nevű ncdf4 típusú objektum készült el, hanem a proba.nc nevű állomány is a munkavégzés mappájában. Utóbbi még adatokkal való feltöltésre vár.

A NetCDF típusú állomány feltöltése adatokkal:

A feltöltés az ncvar_put függvénnyel végezhető, amelynek szintaxisa:

```
ncvar_put(az ncdf4 típusú objektum neve, varid=a változó  
tulajdonságait tartalmazó objektum neve, vals=a változót  
tartalmazó objektum, start=c(szám1,szám2,...),  
count=c(szám1,szám2,...))
```

A függvény argumentumának első helyére írandó az `nc_create` függvénnyel létrehozott `ncdf4` típusú objektum neve. A változó értékei az `ncvar_put` függvény `vals` paraméterénél adandók meg, pontosan feltüntetve az adatokat tartalmazó tömb dimenzióit. A `start` és a `count` paraméterrel szabályozható, hogy a változó mely értékei kerüljenek kiírásra, hiszen tetszőleges értékek figyelmen kívül hagyhatók.

A példában: A függvény argumentumának első és második helyén az `nc` és a `var3d` objektum adandó meg. A változó értékeit a `dataset` objektum adatai jelentik, ezért ez állítandó be a `vals` paraméternél. Az összes adatot tárolni szeretnénk, amit a `start` és a `count` paraméterrel állítunk be. A `start` paraméterrel jelöljük ki a kezdőindexeket, a `count` paraméterrel pedig azt, hogy ettől az indextől hányadik indexig kerüljön sor az adatok kiírására, esetünkben 1-től 31-ig, 1-től 144-ig és 1-től 2-ig.

```
ncvar_put(nc,      varid=var3d,      vals=dataset[1:31,1:144,1:2],  
start=c(1,1,1), count=c(31,144,2))
```

Megjegyzés: Ha minden adatot kiírunk, akkor a `count` paraméterhez `(-1,-1,-1)`-et is írhatunk 31, 144 és 2 helyett. A `start` és a `count` paraméterek elhagyásával a változó(k) minden értéke kiírásra kerül.

A NetCDF állomány lezárása az `nc_close` függvénnyel:

Az `ncdf4` objektum lezárásával válik hozzáférhetővé a munkavégzés mappájában létrehozott NetCDF állomány.

```
nc_close(nc)
```

3. NETCDF ÁLLOMÁNY BEOLVASÁSA, ADATAINAK KEZELÉSE:

Olvassuk be az előbb létrehozott proba.nc NetCDF állomány adatait!

Adjuk meg az állomány elérési útját a következőképpen!

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

Ha új R-t nyitottunk, akkor nyissuk meg az ncd4 csomagot!

```
library(ncdf4)
```

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

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

Megjegyzés: Ha átállítottuk volna a setwd függvénnyel a munkavégzés mappáját arra, amelyben a proba.nc állományt található, akkor a proba <- nc_open("proba.nc") utasítással is megnyithattuk volna az állományt.

A NetCDF állomány az nc_open függvénybeli write=TRUE paraméterrel nyitható meg írásra. Alapértelmezetten csak olvasásra nyitjuk meg a fájlt. A verbose=TRUE paraméterrel kiírathatók a képernyőre a megnyitás információi. Alapértelmezetten ezek nem jelennek meg.

Ellenőrizzük a proba objektum típusát!

```
class(proba)
[1] "ncdf4"
```

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 valtozo 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
```

A proba ncd4 típusú objektum, amely a háromdimenziós tömb adatait és legfőbb tulajdonságait tartalmazza. Az állomány elérési útját követő sor alapján egy, 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 mentése:

Előbbi utasításokkal a dimenziók és a változók számszerű értékei még nem állnak rendelkezésre. Ezek a változók és a dimenziók rövid neveinek hivatkozásával menthetők ki az `ncvar_get` függvénnyel, amelynek szintaxisa:

```
ncvar_get(az ncdf4 típusú objektum neve, varid="a változó vagy  
dimenzió rövid neve", start=c(szám1,szám2,...),  
count=c(szám1,szám2,...))
```

A példában: Mentsük ki a dimenziók és a változók értékeit objektumokba!

```
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))
```

A `start` és a `count` paraméterek jelentése az `ncvar_put` függvénybelivel azonos.

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
```

A `dim1`, `dim2`, `dim3` és `valtozo` objektumok tartalmának azonosnak kell lennie a NetCDF állomány készítése során létrehozott `elsodim`, `masodikdim`, `harmadikdim` és `dataset` objektumok adataival.

Fontos! A változók objektumként való tárolása helyigényes. Egy 3,1-3,3 MHz processzorú, 6 GB RAM-mal rendelkező számítógépen körülbelül 1 GB-os NetCDF állományból már nem

menthető ki a változó objektumként teljes egészében, a számítógép jelentős lassulása nélkül. Ilyen esetben a változónak csak az a része olvasandó be, amellyel éppen dolgozni szeretnénk.

Jelöljük ki a változó adott időpontra és rácspontra vonatkozó értékét!

Például keressük ki a változó Budapest földrajzi koordinátáira és az első időpontra vonatkozó értékét! Budapest földrajzi koordinátái körülbelül é. sz. 47° és k. h. 19° , 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 értéke pedig 1.

```
dim1[14]
```

```
[1] 47.5
```

```
dim2[81]
```

```
[1] 20
```

```
dim3[1]
```

```
[1] 1
```

A keresett adat:

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

4. 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>