Beolvasás, mentés

Térinformatika R-ben

2023.11.07.

Section 1

Bevezetés

Bemutatkozás

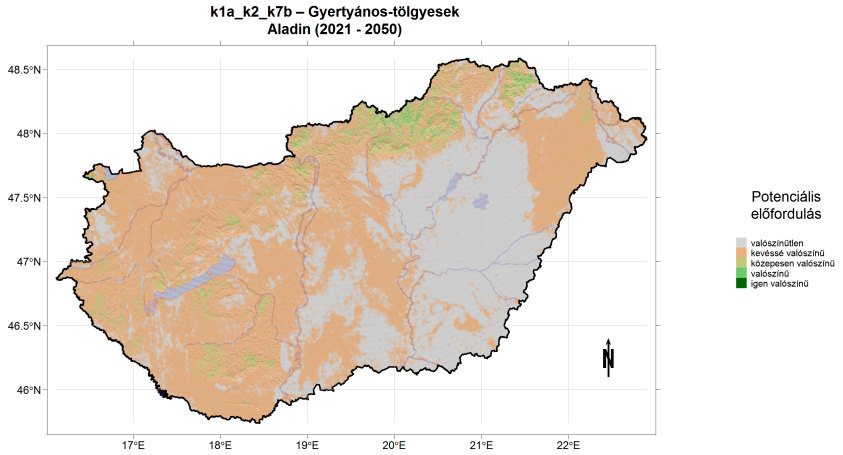
végzettségek: tájépítészmérnök + programtervező informatikus

tudományos munkatárs Vácrátóton

- prediktív elterjedési modellek
- éghajlatváltozás várható hatása a fajok/vegetációtípusok elterjedésére

egyetemi adjunktus az ELTE TTK-n

- programozási alapismeretek (Python)
- térinformatika
- térinformatikai programozás (szkripteszközök fejlesztése ArcMaphez és QGIShez, Leaflet, Google Earth Engine)



Bemutatkozás

tegeződjünk!
rólatok:

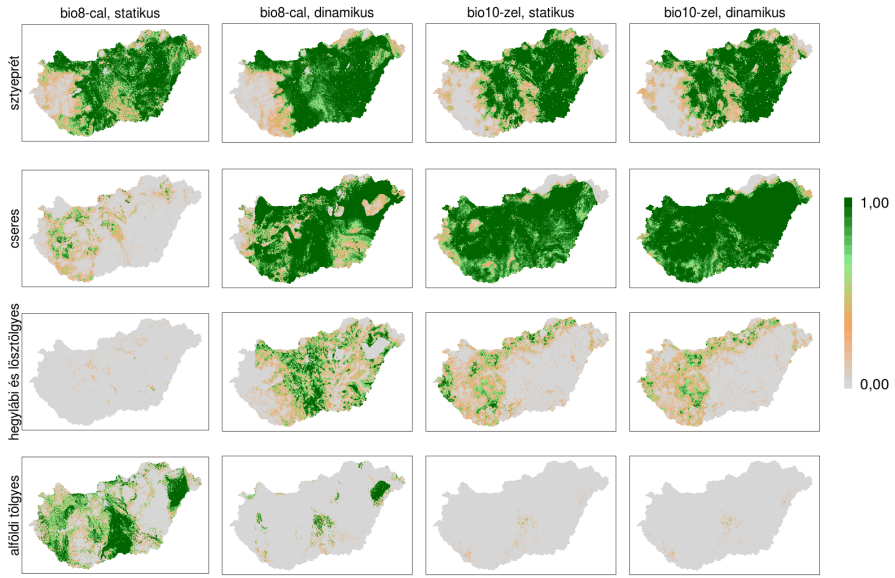
- R-t mennyire ismered, mire használod?
- programozási tapasztalatok (pl. Python)
- mit vársz a tárgytól?
- miért vetted fel a tárgyat?

Tárgy célja

A munkám során három gyakori esetben kell R-es térinformatikai csomagokat használnom:

- valódi térinformatikai feladatok (pl. éghajlati adatok leskálázása),
- hagyományos statisztikai elemzések térben allokált adatsorokon,
- térképi ábrázolás.

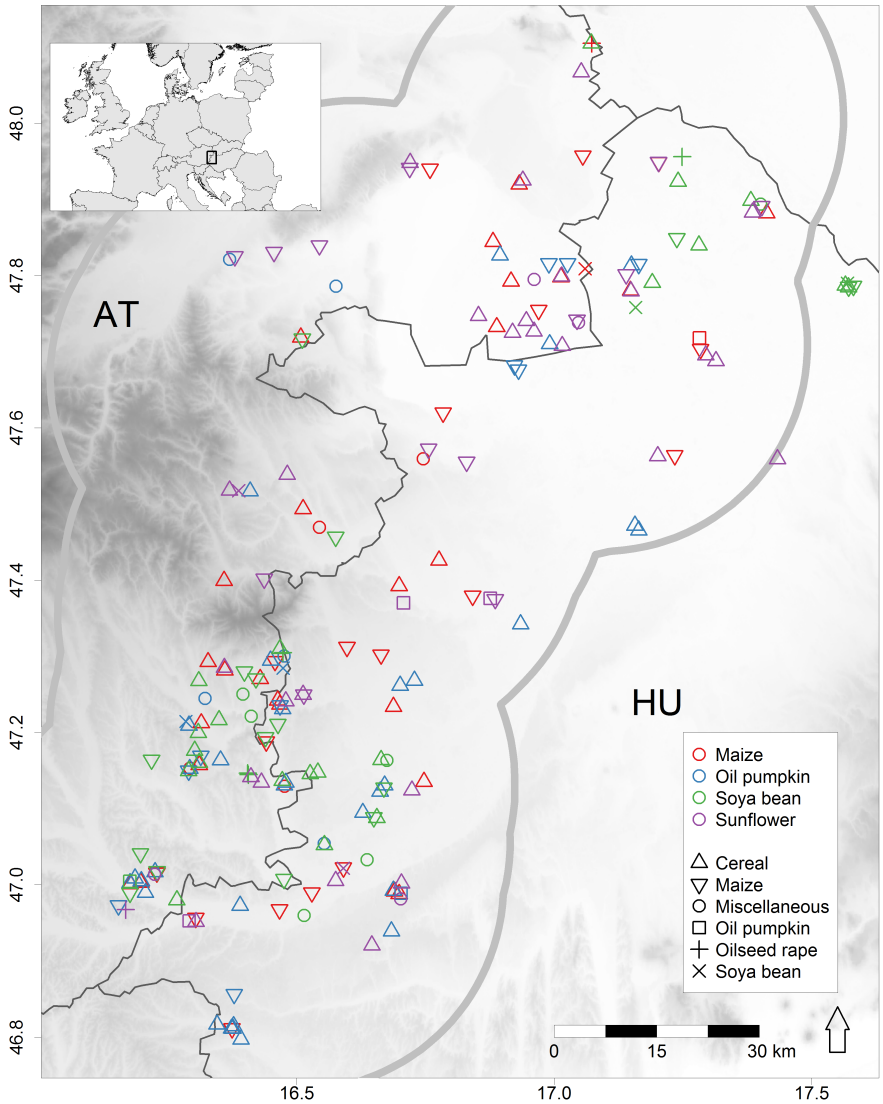
Amit lehet, R-ben valósítok meg (kb. mindent lehet...).



A kurzusról

érintett témakörök:

- bevezetés, csomagok, alapok
- vektoros adatok kezelése
- geometria és tulajdonságok megjelenítése
- vetületek, transzformációk
- egy- és kétváltozós geometriai műveletek
- raszteres adatok kezelése
- vektorok és raszterek kapcsolata
- (geostatisztikai alapok)



A kurzusról

Gyakorlati jellegű tudás, de nincs elég idő a begyakorlásra...
Hibrid megoldás:

- nagyobb anyagot előadásszerűen bemutatok
- melynek során kivetítek R-kódokat (+ az eredményüket)
- ezek a kiadott szkriptek segítségével követhetőek (de ez nem feltétlenül szükséges)

Begyakorlás:

- az előadásba viszonylag sok gyakorló feladatot tűzdeltem
- ezek egy részét átugorjuk (házi feladat), egy részével megpróbálkozunk
- a feladatok megoldása a diasoron és a kiadott szkriptekben is megtalálható
- aki követi a szkriptet, futtassa le minden esetben a feladatmegoldásokat is!

És természetesen kérdés esetén később is keressetek bátran!

Section 2

Irodalmak

Irodalmak

Roger S. Bivand, Edzer J. Pebesma, Virgilio Gómez-Rubio: **Applied Spatial Data Analysis with R** (link)
Barry Rowlingson: **Geospatial Data in R And Beyond!** (link)
Francisco Rodriguez-Sanchez: **Spatial data in R: Using R as a GIS** (link)
Robin Lovelace, James Cheshire, Rachel Oldroyd és mtsai.: **Introduction to visualising spatial data in R** (link)

Irodalmak

Jérôme Guélat: **Spatial data manipulation** (link)
Bede-Fazekas Ákos: **Térinformatika az sp, raster, rgdal és rgeos csomagokkal** (link)
useR! International R User 2017 Conference. R Spatial Talks (link) – prezentációk hanganyaggal
Roger Bivand: **Analysis of Spatial Data** (link)
Robert Hijmans: **Spatial Data Analysis and Modeling with R** (link)

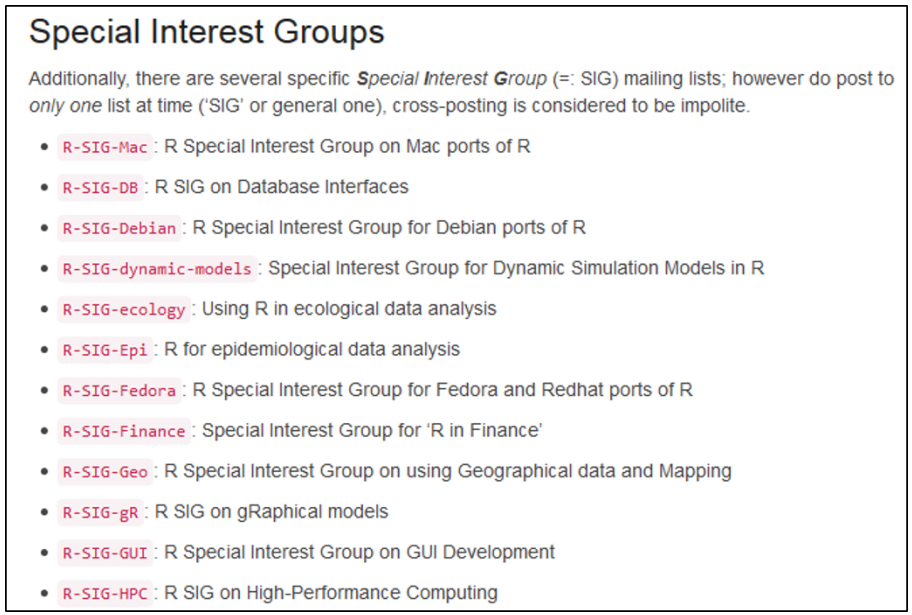
Irodalmak

r-sig-geo levelezőlista

- Special Interest Groups (SIG)
- angol nyelvű
- segítőkész közösség, viszonylag gyors válaszok
- nagyobb pörgés, mint az r-sig-ecology levelezőlistán

CRAN Task View: Analysis of Spatial Data link

- nagyon sok (ha nem is az összes) térinformatikai R-csomag tematikus bontásban
- hasznos, ha a térinformatikán belül valami szűkebb területre kell fókuszálnunk



Special Interest Groups

Additionally, there are several specific ***S**pecial **I**nterest **G**roup* (=: SIG) mailing lists; however do post to *only one* list at time ('SIG' or general one), cross-posting is considered to be impolite.

- `R-SIG-Mac` : R Special Interest Group on Mac ports of R
- `R-SIG-DB` : R SIG on Database Interfaces
- `R-SIG-Debian` : R Special Interest Group for Debian ports of R
- `R-SIG-dynamic-models` : Special Interest Group for Dynamic Simulation Models in R
- `R-SIG-ecology` : Using R in ecological data analysis
- `R-SIG-Epi` : R for epidemiological data analysis
- `R-SIG-Fedora` : R Special Interest Group for Fedora and Redhat ports of R
- `R-SIG-Finance` : Special Interest Group for 'R in Finance'
- `R-SIG-Geo` : R Special Interest Group on using Geographical data and Mapping
- `R-SIG-gR` : R SIG on gRaphical models
- `R-SIG-GUI` : R Special Interest Group on GUI Development
- `R-SIG-HPC` : R SIG on High-Performance Computing

Section 3

Térinformatikai csomagok

Alapvető térinformatikai csomagok

vektor vs. raszter

`sp` – vektorok

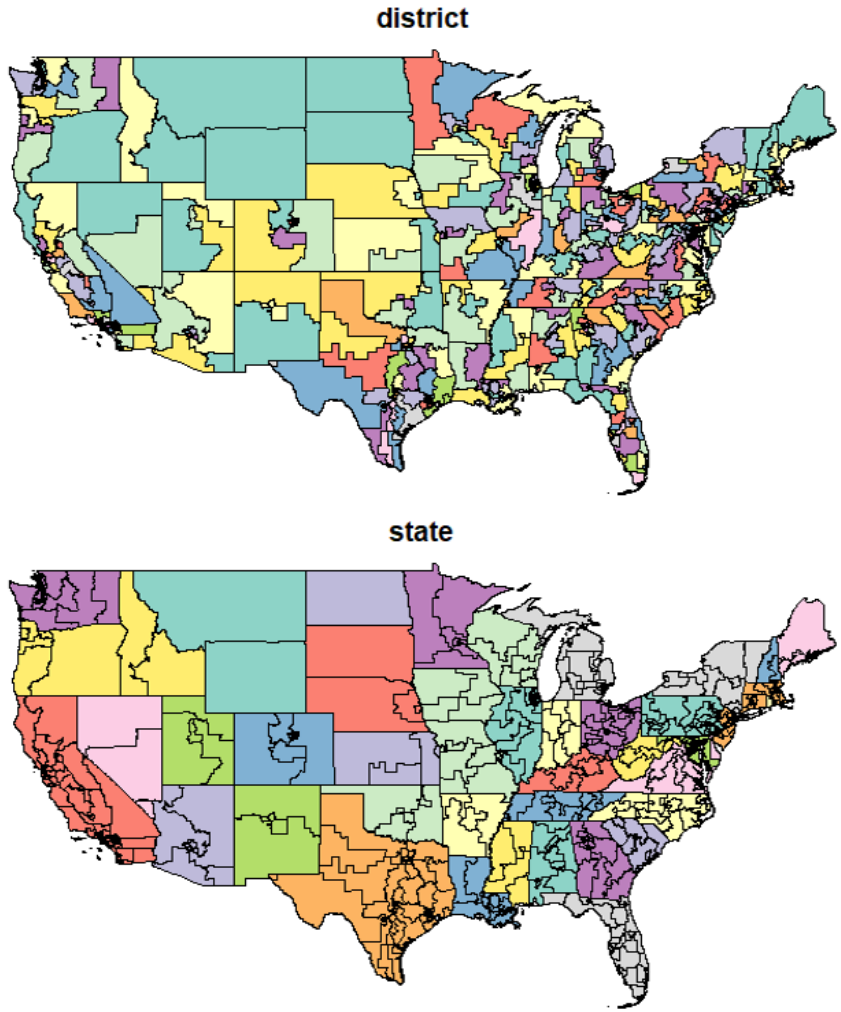
- "Spatial"
- régi
- de széles körben használt
- kompatibilis

`sf` – vektorok

- "Simple Features"
- újabb, néhány éve fejlesztették
- nagyon kényelmes (`ggplot`)
- hosszú távon átveszi az `sp` csomag szerepét

A következőkben az `sf`-et fogjuk használni.

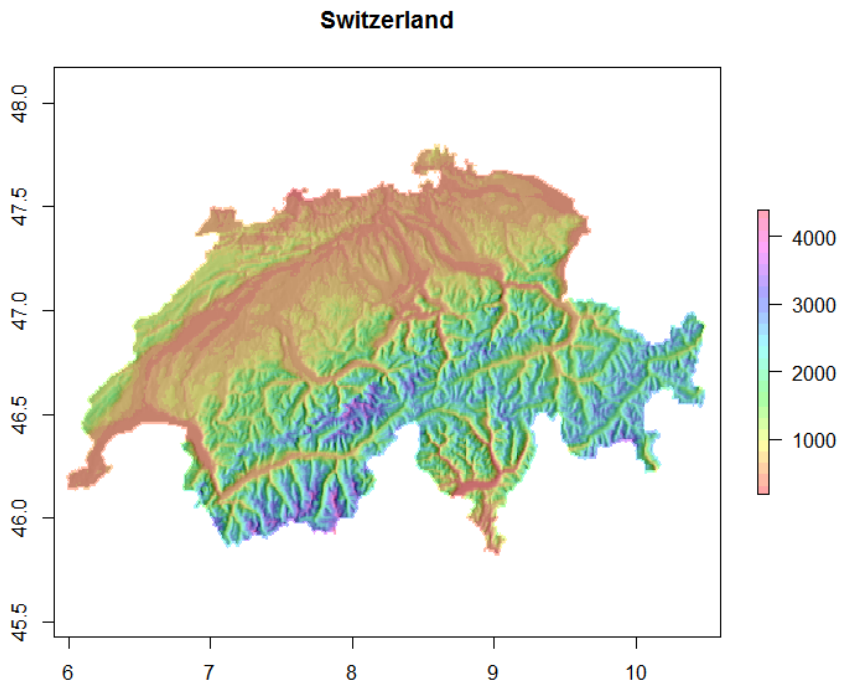
A két csomag között kényelmes az átjárás.



Alapvető térinformatikai csomagok

`raster` – raszterek

- régi csomag
- sok függvény
- egy ember fejlesztette
- rossz dokumentáció
- de egyeduralkodó



Alapvető térinformatikai csomagok

`terra` – raszterek (+vektorok)

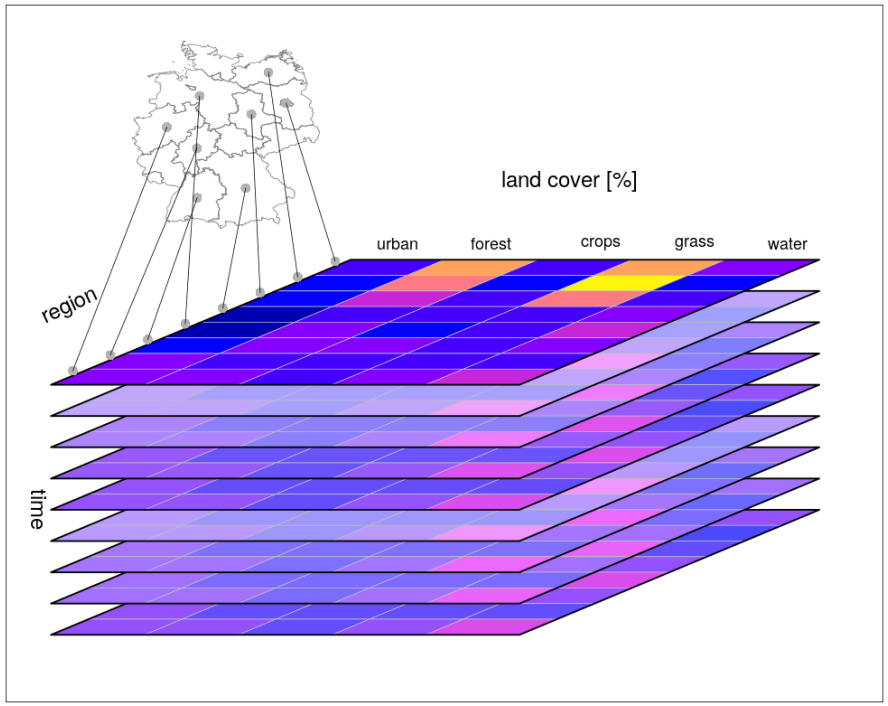
- új csomag, még aktív fejlesztés alatt
- csapatban fejlesztik
- célja, hogy átvegye a `raster` csomag szerepét
- szándékosan nagyon hasonló a logikája, hasonlók a függvénynevek (könnyen át lehessen térni erre a csomagra)
- gyorsabb és egy fokkal jobban dokumentált, mint a `raster`

Ebben a félévben a `raster`-t fogom bemutatni.
A két csomag között kényelmes az átjárás.

Alapvető térinformatikai csomagok

`stars` – többdimenziós tér-idő tömbök (adatkockák)

- **S**patio**t**emporal **Ar**ray**s**
- elvben vektoros adatot is kezel, de leginkább raszterekre lett kifejlesztve
- lehet vele egyszerű rasztereket is kezelni, de ágyúval verébre...
- elsősorban akkor van értelme, ha a két földrajzi dimenzión túl legalább egy további dimenziónk is van (pl. idő)
- friss csomag, még aktív fejlesztés alatt
- közeli rokona az `sf` csomagnak, hasonló logika, szoros kapcsolat



Alapvető térinformatikai csomagok

A különböző vektoros (`sp`, `sf`, `terra`) és raszteres (`raster`, `terra`, `stars`) csomagok saját adattípusokat (osztályokat) használnak.
Ezek között viszonylag fájdalommentes az átjárás: (link).

FROM/TO	sf	sp	terra
sf		*as(x, "Spatial")*	*vect()*
sp	*st_as_sf()*		*vect()*
terra	*st_as_sf()*	*as(x, "Spatial")*	

FROM/TO	raster	terra	stars
raster		*rast()*	*st_as_stars()*
terra	*raster()*		*st_as_stars()*
stars	*raster()*	*as(x, "SpatRaster")*	

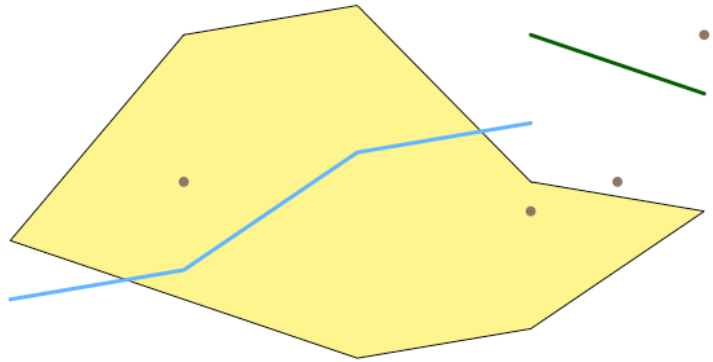
Alapvető térinformatikai csomagok

`rgdal`

- beolvasás és mentés
- GDAL (Geospatial Data Abstraction Library) szükséges hozzá
- más, akár kattintgatós térinformatikai programok (pl. QGIS) is támaszkodnak a GDAL-ra
- már nem fejlesztik

`rgeos`

- geometriai műveletek, topológia



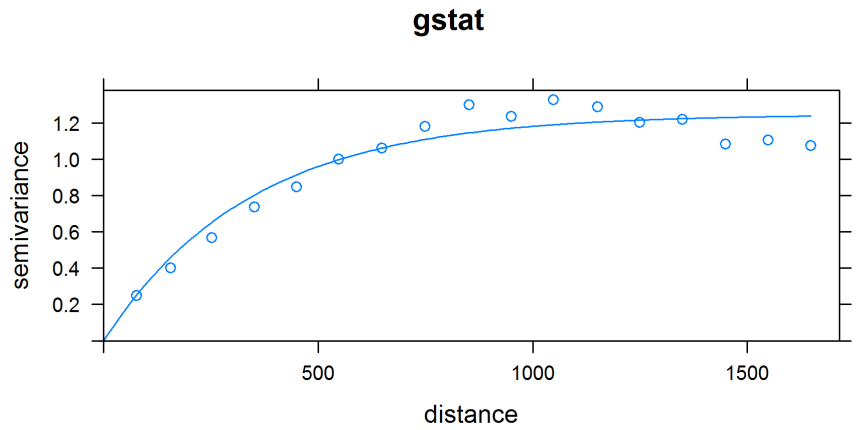
További térinformatikai csomagok

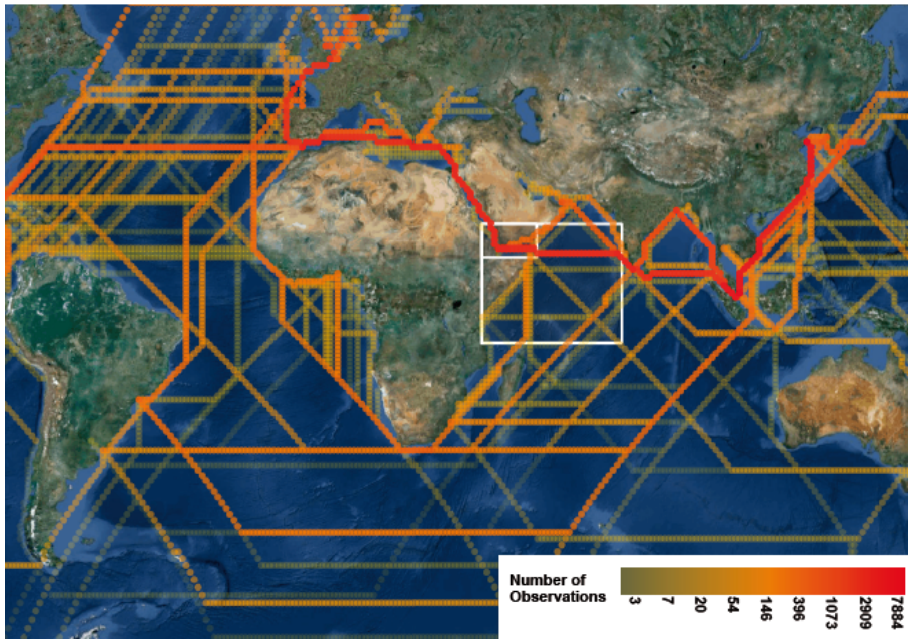
`gdistance`, `geosphere`
- terület-/távolságszámítások

`gstat`, `geoR`
- geostatisztika

`spatial`, `spatstat`
- pontmintázat elemzése

`spcosa`
- mintavételezés





További térinformatikai csomagok

`spdep`

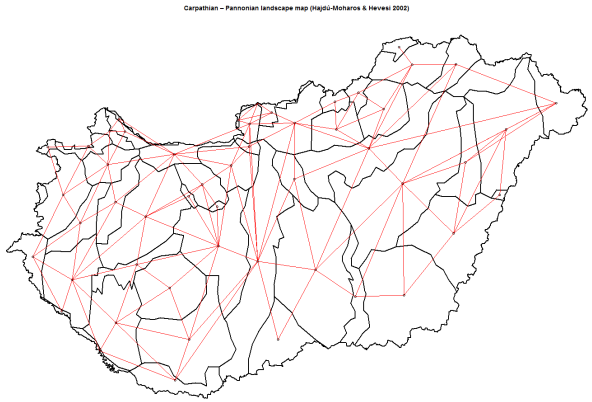
- térbeli összekötöttség

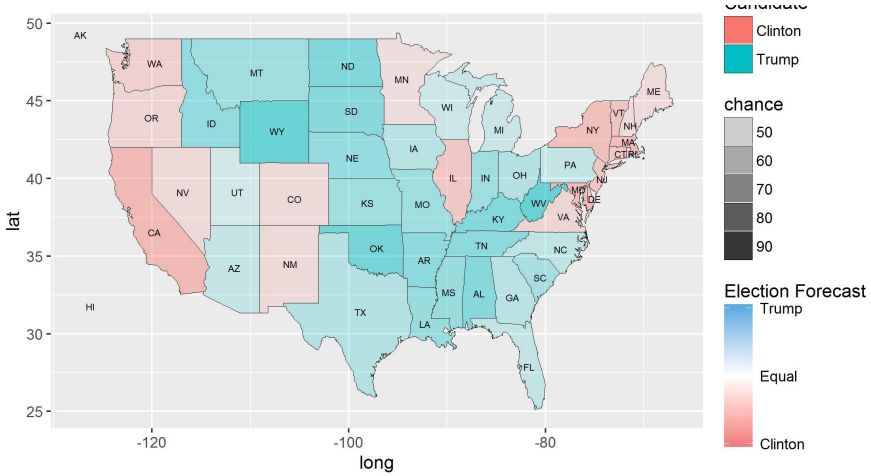
`maps`, `maptools`, `shapefiles`

- további adatbeolvasós csomagok

`spatial.tools`

- további hasznos függvények





További térinformatikai csomagok – a `raster` csomag kiegészítői

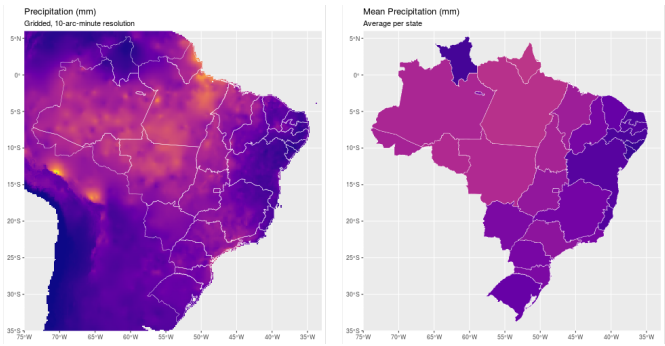
`Velox`
- gyors rasztermű̈veletek

`exactextractr`
- gyors és pontos cellaértékkivonás poligonok alól

`rasterB`
- memóriahatékony rasztermű̈veletek

`fasterize`
- memóriahatékony poligonraszterizálás

`ncdf4`, `RNetCDF`
- NetCDF-fájlok kezelése



További térinformatikai csomagok – vizualizáció

`rasterVis`

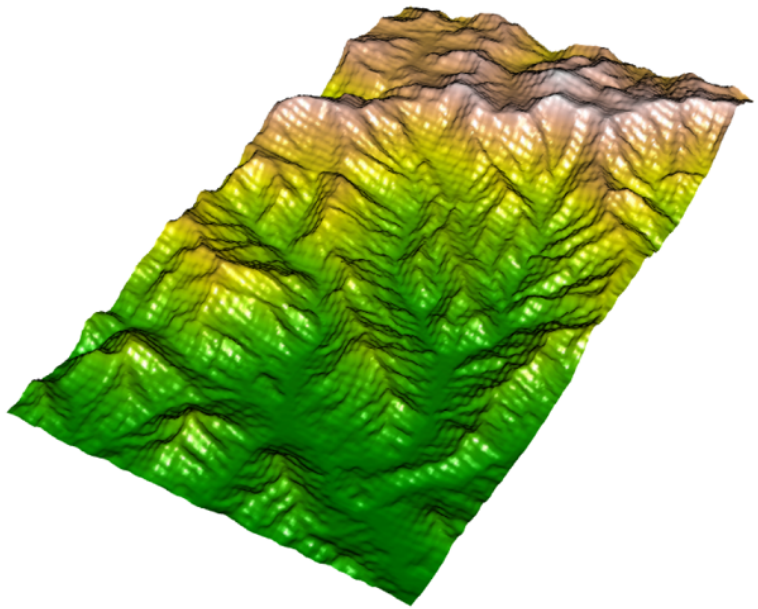
- raszterekhez

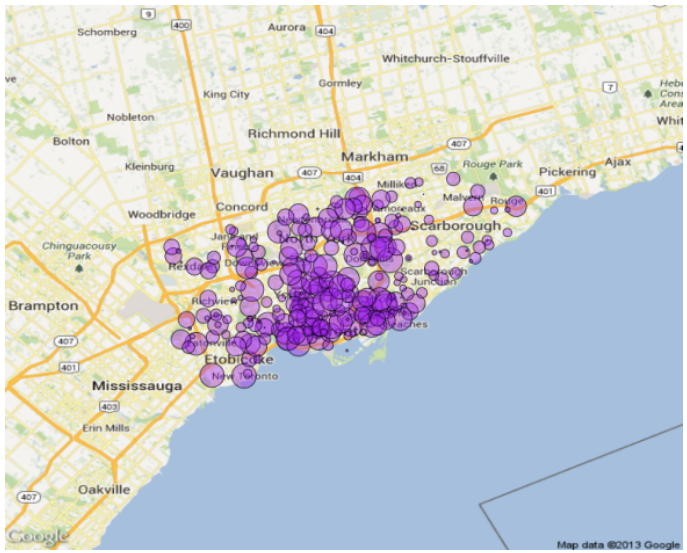
`rworldmap`, `rnaturalearth`

- világtérkép

`RgoogleMaps`, `ggmap`, `ggsn`, `plotGoogleMaps`

- alaptérképek





További térinformatikai csomagok – vizualizáció

`tmap`

- ggplot-jellegű megjelenítés

`quickmapr`, `mapview`, `leafletR`

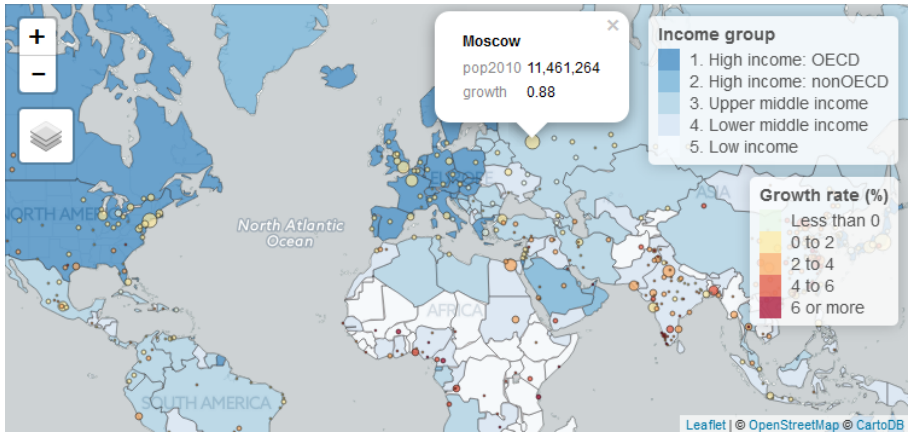
- interaktív megjelenítés

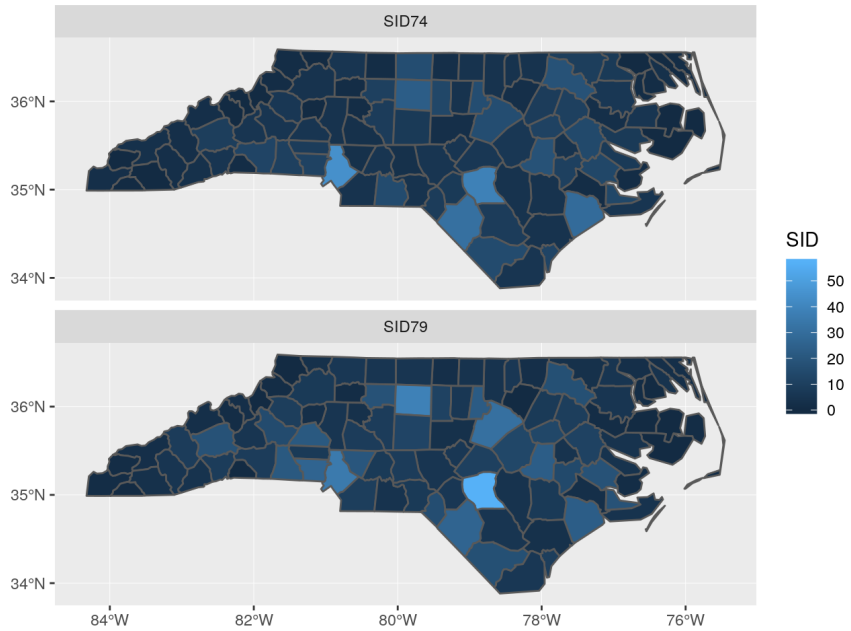
`mapmisc`

- gyors, egyszerű megjelenítés

`cartography`, `mapsf`, `prettymapr`

- különböző kartográfiai eszközök





További térinformatikai csomagok

és még sok más...

A felsorolt csomagok egy része nincs fent a CRAN-on vagy már nem fejlesztik.
Elég dinamikusan változik a paletta, de a CRAN Task View: Analysis of Spatial Data oldal naprakész.

Section 4

Egyszerű alakzatok (Simple Features)

Egyszerű alakzatokról általában

feature

- a valós világ térbeli tárgyainak (jelenségeinek) digitális reprezentációja, absztrakciója
- vektorok

Simple Geospatial Features Access

- röviden: **Simple Features**
- ~egyszerű térbeli alakzatok kezelése

ISO-szabvány

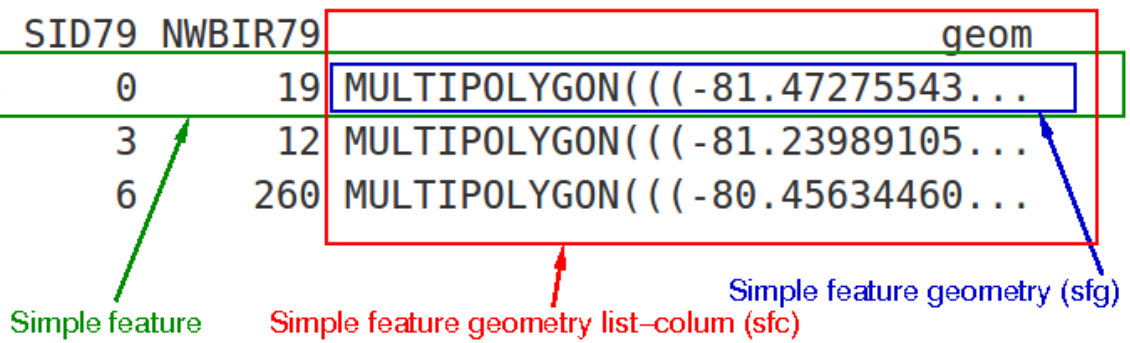
- térinformatikai (2D-s) alakzatok tárolására és kezelésére
- 2004

Egyszerű alakzatokról általában

táblázatos adatszerkezet

- sor: egyszerű alakzat
- oszlop: jellemző
- geometria is egy jellemző → egy újabb oszlopban
- geometriai oszlop neve általában: “geometry”

```
## Simple feature collection with 100 features and 6 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## epsg (SRID):    4267
## proj4string:    +proj=longlat +datum=NAD27 +no_defs
## precision:      double (default; no precision model)
## First 3 features:
##   BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79                       geom
## 1  1091     1       10  1364     0      19 MULTIPOLYGON(((-81.47275543...
## 2   487     0       10   542     3      12 MULTIPOLYGON(((-81.23989105...
## 3  3188     5      208  3616     6     260 MULTIPOLYGON(((-80.45634460...
```



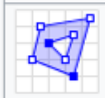
Egyszerű alakzatokról általában

geometria
- tárolása a “well-known text” (WKT) jelölőnyelvvel

17 geometriatípus; a fontosabbak:
- **Point**, MultiPoint
- **LineString**, **MultiLineString**
- Polygon, **MultiPolygon**
- GeometryCollection

dimenziók
- 2D: **XY**
- 3D/4D: **XYZ**, XYM, XYZM
- Z: tszf. magasság
- M: valaminek (idő/hiba/vonal menti távolság) a mértéke

Geometry primitives (2D)

Type	Examples	
Point		`POINT (30 10)`
LineString		`LINESTRING (30 10, 10 30, 40 40)`
Polygon		`POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))`
		`POLYGON ((35 10, 45 45, 15 40, 10 20, 35 10), (20 30, 35 35, 30 20, 20 30))`

Multipart geometries (2D)

Type	Examples	
MultiPoint		`MULTIPOINT ((10 40), (40 30), (20 20), (30 10))`
		`MULTIPOINT (10 40, 40 30, 20 20, 30 10)`
MultiLineString		`MULTILINESTRING ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))`
MultiPolygon		`MULTIPOLYGON (((30 20, 45 40, 10 40, 30 20)), ((15 5, 40 10, 10 20, 5 10, 15 5)))`
		`MULTIPOLYGON (((40 40, 20 45, 45 30, 40 40)), ((20 35, 10 30, 10 10, 30 5, 45 20, 20 35), (30 20, 20 15, 20 25, 30 20)))`

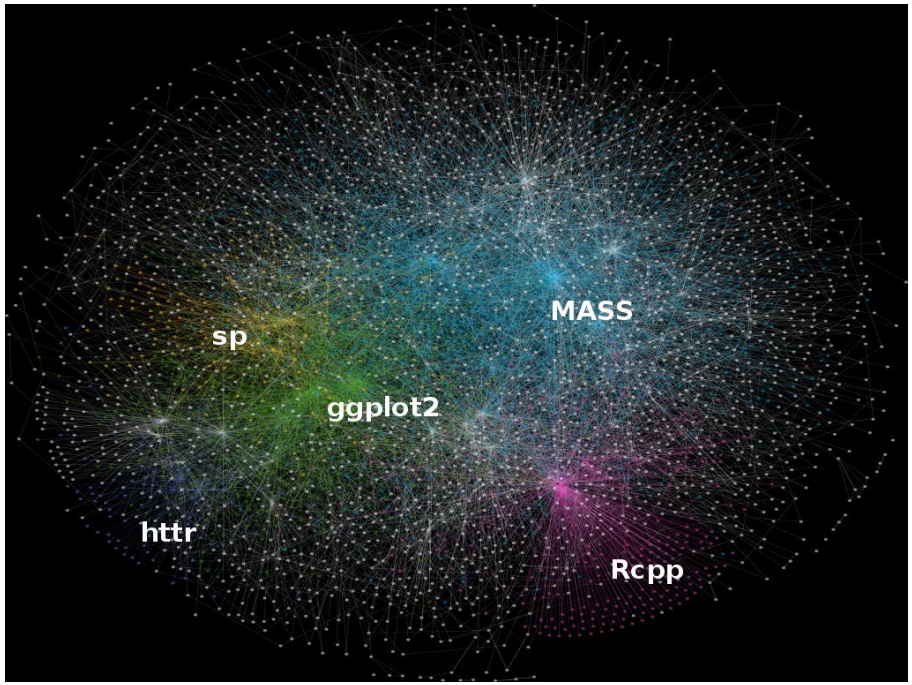
Egyszerű alakzatok R-ben

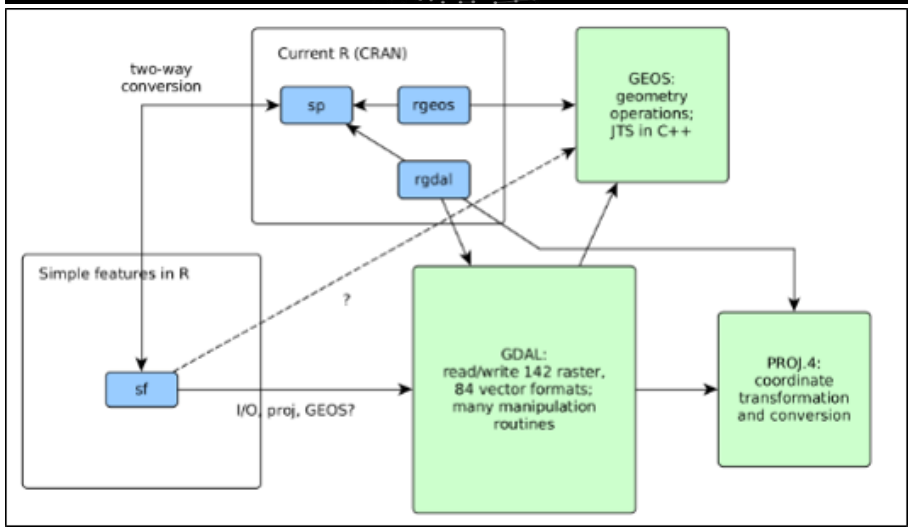
sf csomag

- Edzer Pebesma
- 2017-től elérhető
- gyakorlatilag a korábbi sp csomagot leváltotta

funkcionalitás

- sp + rgdal + rgeos egyben
- függvények megfeleltetése: github.com/r-spatial/sf/wiki/Migrating





Egyszerű alakzatok R-ben

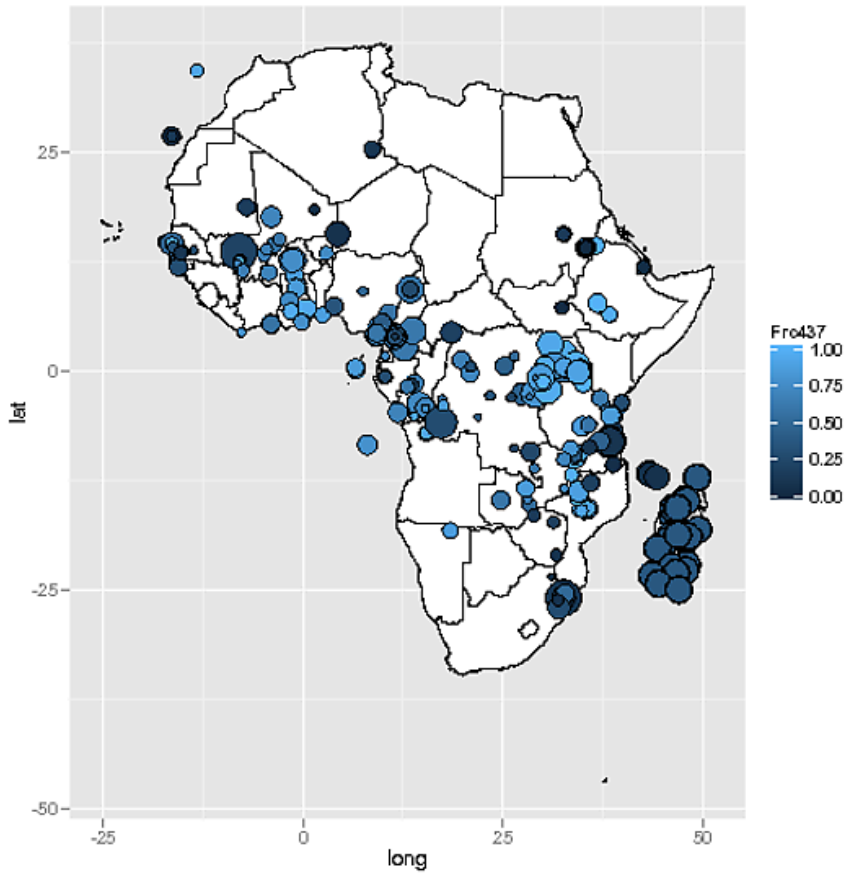
adattípus (osztály):

- sf és data.frame
- data.frame → sok művelet egyszerűen végezhető (pl. szűkítés)
- de a geometriai oszlop bezavarhat...

függvények egységes prefixszel

```
st_*
```

Valaha tér-idő (**s**patio**t**emporal) adatok kezelésére kezdték kifejleszteni a Simple Features-szabványt.
Az időbeli dimenzió azóta elhalt, de a t betű megmaradt...



Egyszerű alakzatok R-ben

tiszta adatok (tidy data)

- tiszta adatokat kezelő függvények használhatóak
- csőszintaxis (pipe, %>%, |>) használatát segítő függvények

```
st_set_*
```

Több függvényből van hagyományos (pl. `st_crs()`, `st_geometry()`) és csőszintaxist segítő verzió (pl. `st_set_crs()`, `st_set_geometry()`) is.

Section 5

Beolvasás és mentés

Térinformatikai adatfájl beolvasása

a kezelt adatformátumok lekérdezése:

```
st_drivers()
```

táblázatot ad vissza
`write` oszlop: tud-e írni ebben a formátumban

Térinformatikai adatfájl beolvasása

```
library(sf)
ismert_adatformatumok <- st_drivers()
ismert_adatformatumok$name
```

```
 [1] "PCIDSK"          "netCDF"          "PDS4"
 [4] "VICAR"           "JP2OpenJPEG"     "PDF"
 [7] "MBTiles"         "BAG"             "EEDA"
[10] "OGCAPI"          "ESRI Shapefile"  "MapInfo File"
[13] "UK .NTF"         "LVBAG"           "OGR_SDTS"
[16] "S57"             "DGN"             "OGR_VRT"
[19] "Memory"          "CSV"             "GML"
[22] "GPX"             "KML"             "GeoJSON"
[25] "GeoJSONSeq"      "ESRIJSON"        "TopoJSON"
[28] "OGR_GMT"         "GPKG"            "SQLite"
[31] "ODBC"            "WAsP"            "PGeo"
[34] "MSSQLSpatial"    "PostgreSQL"      "MySQL"
[37] "OpenFileGDB"
```

Térinformatikai adatfájl beolvasása

```
ismert_adatformatumok[ismert_adatformatumok$write,
 c("long_name", "is_vector")]
```

```
                        long_name is_vector
1           PCIDSK Database File      TRUE
2     Network Common Data Format      TRUE
3  NASA Planetary Data System 4      TRUE
4                MIPL VICAR file      TRUE
5                 Geospatial PDF      TRUE
6                        MBTiles      TRUE
7     Bathymetry Attributed Grid      TRUE
8                 ESRI Shapefile      TRUE
9                   MapInfo File      TRUE
10                IHO S-57 (ENC)      TRUE
```

1. feladat (házi)

- Kérd le azon térinformatikai formátumok hosszú nevét és írhatóságát, amelyek vektoros adat tárolására alkalmasak.

1. feladat (házi) – megoldás

```
ismert_adatformatumok[ismert_adatformatumok$is_vector,
 c("long_name", "write")]
```

```
                    long_name write
1       PCIDSK Database File  TRUE
2 Network Common Data Format  TRUE
3             Geospatial PDF  TRUE
4                    MBTiles  TRUE
5 Bathymetry Attributed Grid  TRUE
6      Earth Engine Data API FALSE
7                      OGCAPI FALSE
8             ESRI Shapefile  TRUE
```

Térinformatikai adatfájl beolvasása

`st_read(dsn, layer, quiet = FALSE)`

- `dsn`: **d**ata **s**ource **n**ame, mappa vagy fájl – formátumfüggő
- `layer`: fájl vagy rétegnév – formátumfüggő
- `quiet`: csöndben, információk képernyőre írása nélkül olvassa be a fájlt?, alapértelmezetten `FALSE`, vagyis szószátyár
- `dsn` + `layer` helyett lehet csak `dsn`, ekkor a kiterjesztésből megtippeli a formátumot

Térinformatikai adatfájl beolvasása

```
zolyomi <- st_read(dsn = getwd(), layer = "zolyomi", quiet
 = FALSE)
```

```
Reading layer `zolyomi' from data source
        `S:\Oktatás, tanulás\Oktatás, tanulás
         2023_24_1_ősz\Térinformatika R-ben (TTK
         BDI)\előadás\adatok'
  using driver `ESRI Shapefile'
Simple feature collection with 908 features and 3 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box: xmin: 427259.1 ymin: 42504.55 xmax: 937694.3
 ymax: 362053.3
Projected CRS: Hotine_Oblique_Mercator_Azimuth_Center
```

Térinformatikai adatfájl beolvasása

```
class(zolyomi)
```

```
[1] "sf"         "data.frame"
```

Térinformatikai adatfájl beolvasása

Fontos, áttekintő információk képernyőre írása a `print()` függvénnyel (vagy interaktív módban egyszerűen a változónévvel).
Legfontosabb jellemzők:

- sorok (features) és nem geometriai oszlopok (mezők, fields) száma
- geometriatípus
- dimenziók
- befoglaló doboz (bbox, **b**ounding **box**)
- vetület
- első néhány sor

```
print(zolyomi)
zolyomi
```

Térinformatikai adatfájl beolvasása

```
Simple feature collection with 908 features and 3 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box: xmin: 427259.1 ymin: 42504.55 xmax: 937694.3
 ymax: 362053.3
Projected CRS: Hotine_Oblique_Mercator_Azimuth_Center
First 10 features:
   azonosito tipus_id
1          1       17
2          2       18
3          3       18
4          4       21
5          5       19
6          6       20
7          7       20
8          8       18
9          9       19
10        10       20
                                  tipus_nev
```

Térinformatikai adatfájl beolvasása

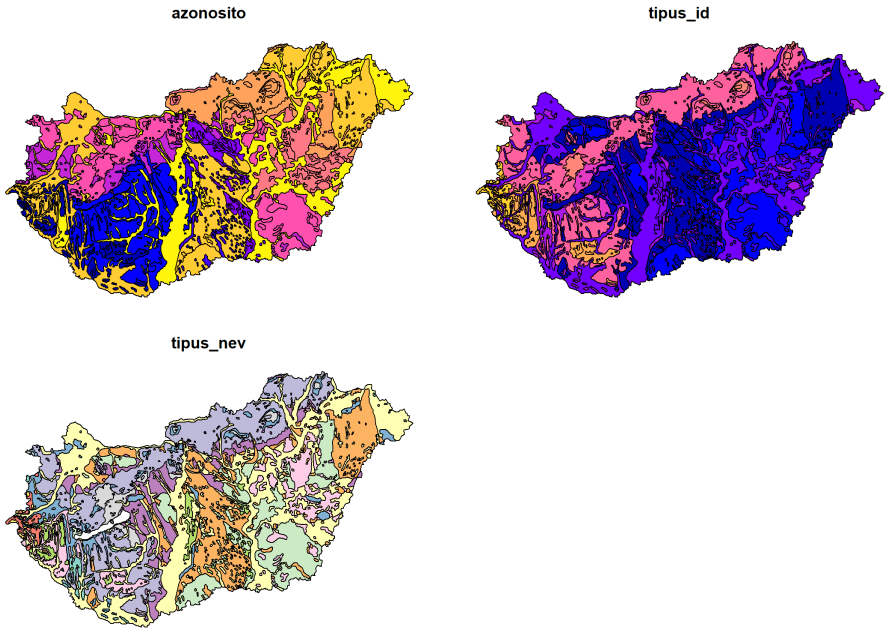
```
str(zolyomi)
```

```
Classes 'sf' and 'data.frame':  908 obs. of  4 variables:
 $ azonosito: int  1 2 3 4 5 6 7 8 9 10 ...
 $ tipus_id : int  17 18 18 21 19 20 20 18 19 20 ...
    $ tipus_nev: chr "illir jellegu bukkosok" "illir
     jellegu gyertyanos-tolgyesek" "illir jellegu
     gyertyanos-tolgyesek" "meszkerulo lomberdo es
     jegenyefenyves lucos" ...
    $ geometry :sfc_MULTIPOLYGON of length 908; first
     list element: List of 1
  ..$ :List of 1
        .. ..$ : num [1:41, 1:2] 461336 461369
         461307 461387 461222 ...
  ..- attr(*, "class")= chr [1:3] "XY" "MULTIPOLYGON" "sfg"
 - attr(*, "sf_column")= chr "geometry"
    - attr(*, "agr")= Factor w/ 3 levels
     "constant","aggregate",..: NA NA NA
        ..- attr(*, "names")= chr [1:3]
         "azonosito" "tipus_id" "tipus_nev"
```

Térinformatikai adatfájl beolvasása

Ábrázolni a `plot()` függvénnyel tudjuk (részletek hamarosan…).
Tulajdonságonként/mezőnként egy színes ábrát kapunk.

```
plot(zolyomi)
```



Térinformatikai adatfájl beolvasása

```
kozeptajak <- st_read(dsn = getwd(), layer = "kozeptajak",
 quiet = TRUE)
kef_halo <- st_read(dsn = "kef_halo.shp", quiet = TRUE)
```

2. feladat (órai)

- Töltsd be csendben (visszajelzés nélkül) az utak.shp-t és a repterek.shp-t, az egyiknél csak a `dsn`, a másiknál a `dsn` és `layer` paramétereket add meg.
- Írasd ki a képernyőre a legfőbb jellemzőiket, tartalmukat.
- Megegyezik a kettőnek az osztálya? Használd a kérdés megválaszolásához az `identical()` függvényt, amely a két bemeneti paramétere teljes egyezőségét vizsgálja ellentétben az == operátorral, amely elemenkénti egyezőséget vizsgál.

2. feladat (órai) – megoldás

```
utak <- st_read(dsn = "utak.shp", quiet = TRUE)
repterek <- st_read(dsn = getwd(), layer = "repterek",
 quiet = TRUE)
utak
repterek
identical(class(utak), class(repterek))
```

```
[1] TRUE
```

Térinformatikai adatfájl mentése

`dir.create(path, showWarnings = FALSE, recursive = TRUE)`

- létrehoz egy mappát
- `path`: a mappa relatív vagy abszolút elérési útja
- `showWarnings`: ha már eleve létezett a mappa, dobjon-e figyelmeztetést
- `recursive`: rekurzívan az egész elérési utat létrehozza-e (ha pl. a szülőmappa sem létezik még)

Térinformatikai adatfájl mentése

`st_write(obj, dsn, driver, quiet = FALSE, delete_dsn = FALSE)`

- vektoros térinformatikai adat mentése fájlba (pl. ESRI Shapefile-ként)
- `obj`: a mentendő vektoros térinformatikai adat (`sf` vagy `sfc` típusú változó)
- `dsn`: **d**ata **s**ource **n**ame, kimeneti fájl (vagy mappa - ebben az esetben a `layer` paraméterrel kell megadni a fájlnevet)
- `driver`: a fájltípus szövegként megadva (opcionális, a kiterjesztésből megtippel)
- `driver`: pl. “netCDF”, “ESRI Shapefile”, “KML”, lásd: `st_drivers()$name`
- `quiet`: megkíméljen-e a fölösleges információk képernyőre írásától
- `delete_dsn`: állítsuk `TRUE`-ra, ha automatikus felülírást szeretnénk

Térinformatikai adatfájl mentése

```
dir.create("kimenet", showWarnings = FALSE, recursive =
 TRUE)
```

```
st_write(obj = kef_halo, dsn = "kimenet/kef_halo.geojson",
 driver = "GeoJSON", quiet = TRUE)
st_write(obj = kef_halo, dsn = "kimenet/kef_halo.geojson",
 driver = "GeoJSON", quiet = TRUE)
```

```
Layer kef_halo in dataset kimenet/kef_halo.geojson already
 exists:
use either append=TRUE to append to layer or append=FALSE
 to overwrite layer

Error in eval(expr, envir, enclos): Dataset already exists.
```

Térinformatikai adatfájl mentése

Az előző mentési próbálkozás hibát dobott, hiszen már létezett a fájl.
A következő viszont sikerül, mert a `delete_dsn` paramétert igazra állítjtuk (hasonló érthető el az `append = FALSE`-szal is):

```
st_write(obj = kef_halo, dsn = "kimenet/kef_halo.geojson",
 driver = "GeoJSON", quiet = TRUE, delete_dsn = TRUE)
st_write(obj = kef_halo, dsn = "kimenet/kef_halo.geojson",
 quiet = TRUE, delete_dsn = TRUE)
```

A `driver` paraméter elhagyható, a kiterjesztés alapján egyértelmű a fájlformátum.

3. feladat (házi)

- Mentsd el
  - a `kozeptajak`-at .shp kiterjesztéssel (ESRI shapefile) és
  - a `zolyomi` poligonokat .kml kiterjesztéssel.
- Ha már léteznek a fájlok, írd felül őket!

3. feladat (házi) – megoldás

```
st_write(obj = kozeptajak, dsn = "kimenet/kozeptajak.shp",
 quiet = TRUE, delete_dsn = TRUE)
st_write(obj = zolyomi, dsn = "kimenet/zolyomi.kml", quiet
 = TRUE, delete_dsn = TRUE)
```

Mentés és beolvasás RDatába/-ból

Természetesen hagyományos módon, RData formátumban/-ból is menthetőek/betölthetők az adatok.

`save(..., list, file)`

- `...`: mentendő változók
- `list`: mentendő változók neve szövegvektorként
- `file`: kimeneti fájl helye és neve, tipikusan .RData kiterjesztéssel

`load(file)`

- `file`: beolvasandó fájl helye és neve
- láthatatlan visszatérési értéke a beolvasott változók neve szövegvektorként

Mentés és beolvasás RDatába/-ból

```
save(utak, file = "kimenet/utak.RData")
save(list = c("repterek", "utak", "kozeptajak"), file =
 "kimenet/mindenfele.RData")
```

4. feladat (házi)

- Mentsd el a `zolyomi` és `kef_halo` adatokat a kimenet/tovabbi_poligonok.RData fájlba kétféle módon,
  - a `...` és
  - a `list` paraméterek használatával!

4. feladat (házi) – megoldás

```
save(list = c("zolyomi", "kef_halo"), file =
 "kimenet/tovabbi_poligonok.RData")
save(zolyomi, kef_halo, file =
 "kimenet/tovabbi_poligonok.RData")
```

Mentés és beolvasás RDatába/-ból

- `ls()`: listázza a változókat
- `rm(list)`: törli a megadott nevű változókat

```
rm(list = ls())
load("kimenet/mindenfele.RData")
ls()
```

```
[1] "kozeptajak" "repterek"   "utak"
```

```
load("kimenet/tovabbi_poligonok.RData")
ls()
```

```
[1] "kef_halo"   "kozeptajak" "repterek"   "utak"
[5] "zolyomi"
```

Néha egyszerűbb elkapni a `load()` visszatérési értékét, mint körülményes módon kitalálni, hogy mit olvastunk be.

Mentés és beolvasás RDatába/-ból

```
beolvasott_valtozok <-
 load("kimenet/tovabbi_poligonok.RData")
beolvasott_valtozok
```

```
[1] "zolyomi"  "kef_halo"
```

5. feladat (házi)

- Olvasd be az utak.RDatát úgy, hogy választ kapjál a következő kérdésre:

Betöltődött a munkkörnyezetbe új változó?

- `TRUE` vagy `FALSE` értéket várok eredményként.

5. feladat (házi) – megoldás

```
valtozok <- ls()
load("kimenet/utak.RData")
length(valtozok) == length(ls())
```

```
[1] TRUE
```

Beolvasás tagolt szövegfájlból/táblázatból

```
read.table(file, sep = "", dec = ".", header = FALSE)
```

- CSV és egyéb tagolt szövegfájlok beolvasására alkalmas
- `file`: a bemeneti fájl helye és neve
- `sep`: **sep**arator, tagoló karakter (alapértelmezetten minden üres karakter, pl. szóköz, tabulátor)
- `dec`: **dec**imal, tizedesjel
- `header`: az első sor fejléc-e (avagy adatot tartalmaz)

Beolvasás tagolt szövegfájlból/táblázatból

```
eghajlat <- read.table(file =
 "eghajlat_fix_szelesseg.csv", sep = "", dec = ".", header
 = TRUE)
class(eghajlat)
```

```
[1] "data.frame"
```

```
str(eghajlat)
```

Beolvasás tagolt szövegfájlból/táblázatból

```
'data.frame':   14178 obs. of  11 variables:
 $ eov_y  : num  609024 609342 609660 609977 610295 ...
 $ eov_x  : num  265367 265917 266468 267018 267569 ...
    $ meta_id: int 53073 53077 53081 53084 53088 53091
     53094 53500 53504 53509 ...
 $ P_DJF  : num  114 110 106 107 104 ...
 $ P_MAM  : num  143 138 132 134 131 ...
 $ P_JJA  : num  182 176 171 173 169 ...
 $ P_SON  : num  145 141 136 137 134 ...
 $ T_DJF  : num  -0.105 0.103 0.318 0.243 0.384 ...
 $ T_MAM  : num  10.1 10.5 10.9 10.7 11 ...
 $ T_JJA  : num  19.3 19.7 20.1 19.9 20.2 ...
 $ T_SON  : num  10.2 10.5 10.8 10.7 10.9 ...
```

Beolvasás tagolt szövegfájlból/táblázatból

Természetesen a táblázatunk nem csak CSV-ből, hanem XLS-ből/XLSX-ből is származhat (használjuk ilyenkor az `xlsx` csomagot!). Vagy lehet, hogy eleve egy `data.frame`-et kapunk egy .RData fájlban. A lényeg, hogy legyen két oszlopa a vízszintes és függőleges koordinátákhoz.

`st_as_sf(x, crs, coords)`

- `data.frame`-et átalakít Simple Features-zé (`sf`)
- `x`: az átalakítandó `data.frame`
- `crs`: **c**oordinate **r**eference **s**ystem, vetületi/koordináta-rendszer
- `coords`: kételemű szövegvektor, a koordinátákat tartalmazó oszlopok neve

Beolvasás tagolt szövegfájlból/táblázatból

vetület:

- többféle módon megadható, de a legegyszerűbb az EPSG-azonosítójával (4 vagy 5 jegyű szám)
- Magyarországra EOV (23700)
- Európára LAEA Europe (3035)
- világra WGS-84 (4326)
- később részletesen...

A két koordináta-oszlop sorrendje fontos!

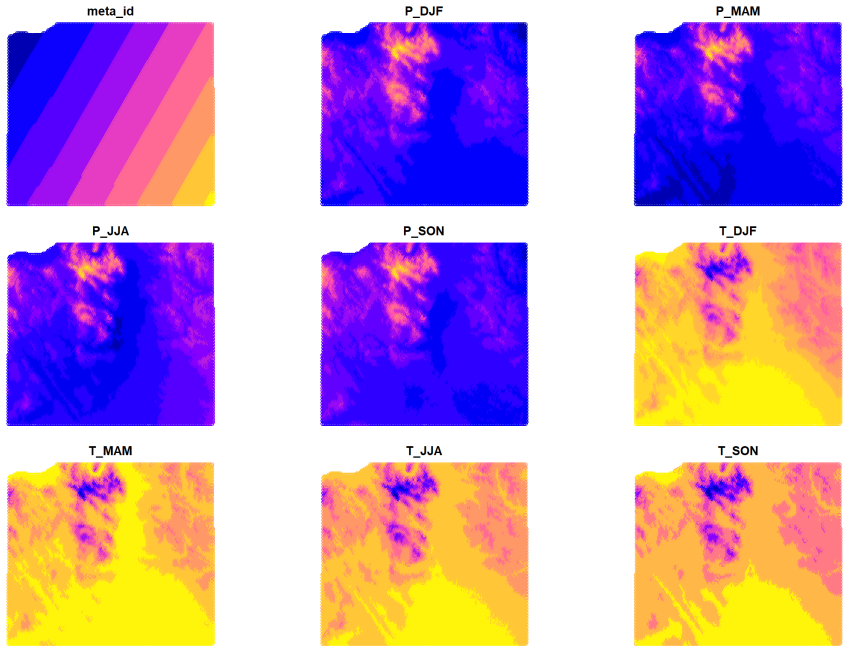
- első a vízszintes (WGS-84 esetén ez általában x/lon/long/longitude, EOV esetén pedig y/eov_y)
- második a függőleges (WGS-84: y/lat/latitude, EOV: x/eov_x)

Beolvasás tagolt szövegfájlból/táblázatból

```
eghajlat <- st_as_sf(x = eghajlat, crs = 23700, coords =
 c("eov_y", "eov_x"))
class(eghajlat)
```

```
[1] "sf"         "data.frame"
```

```
plot(eghajlat)
```



6. feladat (órai)

- Olvasd be az eghajlat_pontosvesszo_tvesszo.csv fájlt, mely
  - pontosvesszővel tagolja az oszlopokat, és
  - tizedesjelként vesszőt alkalmaz.
- Alakítsd át Simple Features-zé.

6. feladat (órai) – megoldás

```
eghajlat <- read.table(file =
 "eghajlat_pontosvesszo_tvesszo.csv", sep = ";", dec =
 ",", header = TRUE)
eghajlat <- st_as_sf(x = eghajlat, crs = 23700, coords =
 c("eov_y", "eov_x"))
```

7. feladat (házi)

- Olvasd be az repterek_vesszo.csv fájlt.
- Alakítsd át Simple Features-zé.

7. feladat (házi) – megoldás

```
repterek <- read.table(file = "repterek_vesszo.csv", sep =
 ",", dec = ".", header = TRUE)
repterek <- st_as_sf(x = repterek, crs = 23700, coords =
 c("eov_y", "eov_x"))
```

Beolvasás tagolt szövegfájlból/táblázatból

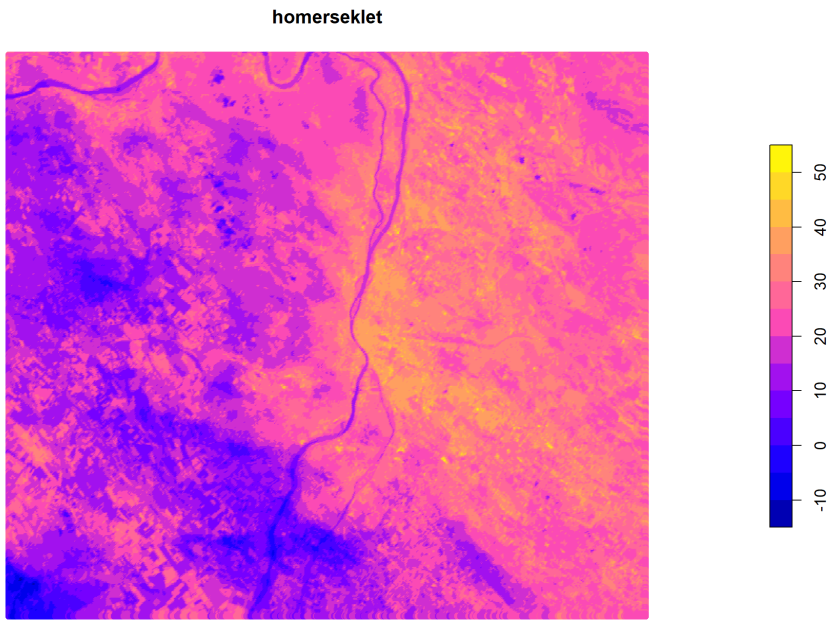
A felszinhomerseklet.csv sűrű rácshálóban tartalmaz felszínhőmérsékleti adatokat.

```
felszinhomerseklet <- read.table(file =
 "felszinhomerseklet.csv", sep = "", dec = ".", header =
 TRUE)
felszinhomerseklet <- st_as_sf(x = felszinhomerseklet, crs
 = 23700, coords = c("eov_y", "eov_x"))
```

Koordináták sorrendjére figyeljünk EOV esetén!

Beolvasás tagolt szövegfájlból/táblázatból

```
plot(felszinhomerseklet)
```



Az ilyen szabályos rácsszerkezetű adatot jellemzően inkább raszterként olvassuk be. Később...

Section 6

Adattábla, geometria, koordináták

Adattábla jellemzői

Oszlopnevek között mindig van egy, ami a geometriát tartalmazza.

- nem feltétlenül az utolsó!
- nem feltétlenül `geometry`-nek hívják
- neve az `sf`-objektum `sf_column` nevű attribútuma

`x` változó `which` nevű attribútumát az `attr(x, which)` függvénnyel kérhetjük le.

Adattábla jellemzői

```
zolyomi <- st_read(dsn = "zolyomi.shp", quiet = TRUE)
```

```
colnames(zolyomi)
```

```
[1] "azonosito" "tipus_id"  "tipus_nev" "geometry"
```

```
names(zolyomi)
```

```
[1] "azonosito" "tipus_id"  "tipus_nev" "geometry"
```

```
attr(x = zolyomi, which = "sf_column")
```

```
[1] "geometry"
```

Adattábla jellemzői

```
nrow(zolyomi)
```

```
[1] 908
```

```
ncol(zolyomi)
```

```
[1] 4
```

8. feladat (házi)

- Olvasd be a folyok.shp-t.
- Három különböző módon kérdezd le az oszlopainak számát!

8. feladat (házi) – megoldás

```
folyok <- st_read(dsn = "folyok.shp", quiet = TRUE)
```

```
dim(folyok)[2]
```

```
[1] 5
```

```
ncol(folyok)
```

```
[1] 5
```

```
length(colnames(folyok))
```

```
[1] 5
```

9. feladat (órai)

- A `zolyomi` adattáblájában hanyadik oszlop tartalmazza a geometriát?

Tipp: használd a `match(x, table)` függvényt!

9. feladat (órai) – megoldás

```
match(x = attr(x = zolyomi, which = "sf_column"), table =
 colnames(zolyomi))
```

```
[1] 4
```

Geometria elhagyása és lekérése

Az `sf` objektumok táblázatos adatokat és geometriát tartalmaznak.

- adatok elhagyása (geometria lekérése) → `sfc` (**S**imple **F**eature geometry list-**c**olumn)
- geometria elhagyása → `data.frame`

```
st_geometry(obj)
st_drop_geometry(x)
```

```
## Simple feature collection with 100 features and 6 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## epsg (SRID):    4267
## proj4string:    +proj=longlat +datum=NAD27 +no_defs
## precision:      double (default; no precision model)
## First 3 features:
##   BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79                       geom
## 1  1091     1      10  1364     0      19 MULTIPOLYGON(((-81.47275543...
## 2   487     0      10   542     3      12 MULTIPOLYGON(((-81.23989105...
## 3  3188     5     208  3616     6     260 MULTIPOLYGON(((-80.45634460...
```

Simple feature

Simple feature geometry list–colum (sfc)

Simple feature geometry (sfg)

Geometria elhagyása

```
class(folyok)
```

```
[1] "sf"         "data.frame"
```

```
folyok_geometria_nelkul <- st_drop_geometry(folyok)
class(folyok_geometria_nelkul)
```

```
[1] "data.frame"
```

Geometria módosítása

Lehetőségünk van a geometria módosítására/felülírására is:

```
st_geometry(x) <- value
st_set_geometry(x, value)
```

- a `value` értéke vagy egy `sfc` (= módosítás), vagy NULL (= elhagyás)
- az `st_set_geometry()` csőszintaxisra alkalmas
- ritkán használjuk (akkor is leginkább a geometria elhagyására), nem mutatom be

10. feladat (házi)

- Olvasd be a zolyomi.RData fájlt.
- Hagyd el a geometriáját.
- Majd kérdezd le, hogy az osztálya `data.frame`-e.
- Az eredmény `TRUE` vagy `FALSE` legyen!

10. feladat (házi) – megoldás

```
load("zolyomi.RData")
```

```
zolyomi <- st_drop_geometry(zolyomi)
identical(x = class(zolyomi), y = "data.frame")
```

```
[1] TRUE
```

Koordináták lekérése

Ha menteni akarjuk táblázatként, érdemes a koordinátákat hozzáfűzni két oszlopban.
Koordináta lekérése táblázatos formában:

```
st_coordinates(x)
```

`POINT` típusú adatnál egyszerű, a többivel most nem foglalkozunk.

Koordináták lekérése

```
load("eghajlat.RData")
```

```
colnames(st_drop_geometry(eghajlat))
```

```
[1] "meta_id" "P_DJF"   "P_MAM"   "P_JJA"   "P_SON"
[6] "T_DJF"   "T_MAM"   "T_JJA"   "T_SON"
```

```
st_coordinates(eghajlat)
```

```
             X        Y
[1,] 609023.8 265366.7
[2,] 609341.6 265917.3
[3,] 609659.5 266467.8
[4,] 609977.4 267018.4
[5,] 610295.2 267569.0
```

Koordináták lekérése

Hozzáfűzéshez a `cbind()` vagy `cbind.data.frame()` függvény használható.

```
head(cbind.data.frame(st_coordinates(eghajlat),
 st_drop_geometry(eghajlat)))
```

```
         X        Y meta_id    P_DJF    P_MAM    P_JJA
1 609023.8 265366.7   53073 114.0949 143.1768 181.8952
2 609341.6 265917.3   53077 109.9018 137.9057 176.4395
3 609659.5 266467.8   53081 105.5848 132.4831 170.8277
4 609977.4 267018.4   53084 107.0700 134.3108 172.7082
5 610295.2 267569.0   53088 104.2061 130.7123 168.9790
6 610613.1 268119.5   53091 102.1007 128.0651 166.2308
```

11. feladat (házi)

- Hozd létre a `repterek_koordinataval` nevű táblázatot, amely a repterek koordinátáit és adatait tartalmazza.
- Mentsd el a korábban létrehozott kimenet nevű mappába, repterek_koordinataval.RData néven.

11. feladat (házi) – megoldás

```
repterek_koordinataval <-
 cbind.data.frame(st_coordinates(repterek),
 st_drop_geometry(repterek))
save(repterek_koordinataval, file =
 "kimenet/repterek_koordinataval.RData")
```

12. feladat (órai)

- Hozd létre a `felszinhomerseklet_koordinataval` nevű táblázatot, amely a felszínhőmérsékleteket, majd utánuk a koordinátáikat tartalmazza.
- Mentsd el a korábban létrehozott kimenet nevű mappába, felszinhomerseklet_koordinataval.RData néven.

12. feladat (órai) – megoldás

```
felszinhomerseklet_koordinataval <-
 cbind.data.frame(st_drop_geometry(felszinhomerseklet),
 st_coordinates(felszinhomerseklet))
save(felszinhomerseklet_koordinataval, file =
 "kimenet/felszinhomerseklet_koordinataval.RData")
```

Geometria lekérése

```
load("zolyomi.RData")
```

```
class(zolyomi)
```

```
[1] "sf"         "data.frame"
```

```
zolyomi_geometria <- st_geometry(zolyomi)
class(zolyomi_geometria)
```

```
[1] "sfc_MULTIPOLYGON" "sfc"
```

A típusa mindig kettős: `sfc` és egy hosszabb valami, ami a geometriatípusra utal (esetünkkben `sfc_MULTIPOLYGON`).

Geometria lekérése

A `print()`-tel (vagy interaktív módban egyszerűen a változónévvel) az `sf`-nél már megismert adatokat kapjuk.
Eltérés:

- nem írja ki a mezők számát (mert nincsenek mezők)
- az első néhány sor helyett az első néhány geometriát írja ki WKT-formátumban

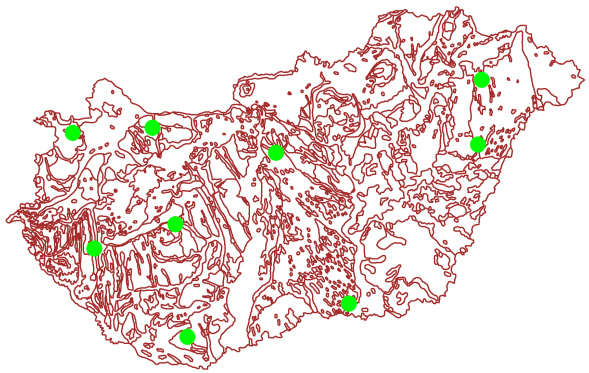
Geometria lekérése

```
Geometry set for 908 features
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box: xmin: 427259.1 ymin: 42504.55 xmax: 937694.3
 ymax: 362053.3
Projected CRS: HD72 / EOV
First 5 geometries:

MULTIPOLYGON (((461336.2 140776.3, 461369 14062...

MULTIPOLYGON (((457096.5 146008.2, 456863.6 145...

MULTIPOLYGON (((451821.4 151222.6, 451913.6 151...

MULTIPOLYGON (((452427.1 152312.6, 451806.1 151...

MULTIPOLYGON (((452856.2 153951.8, 452810.3 153...
```

Geometria lekérése

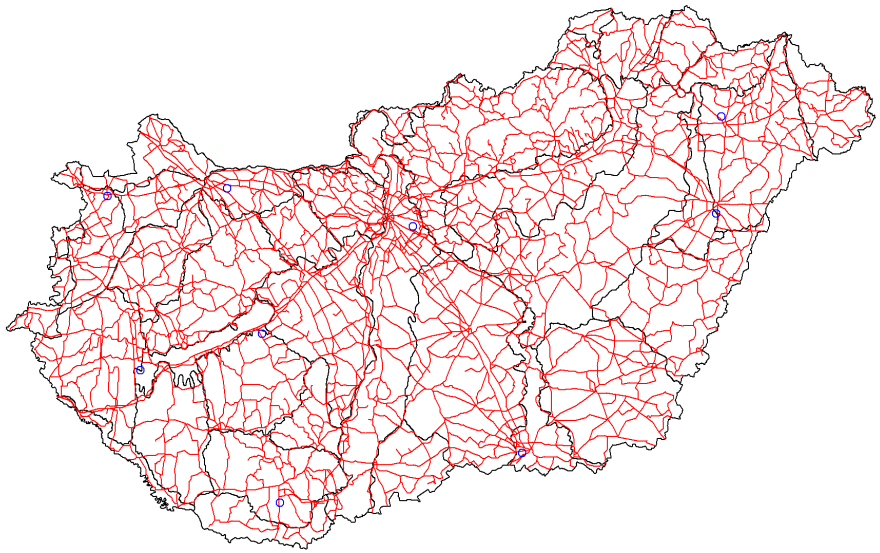
Geometriát ugyanúgy ábrázolhatjuk a `plot()` függvénnyel, csak most nem a mezők alapján színezi, hanem fekete vonalrajzot kapunk.
A színezést/megjelenítést tetszőlegesen felülbírálhatjuk az ismert paraméterekkel (hamarosan...).

```
plot(zolyomi_geometria, border = "brown", col =
 "transparent", lwd = 2)
plot(st_geometry(repterek), col = "green", cex = 3, pch =
 16, add = TRUE)
```



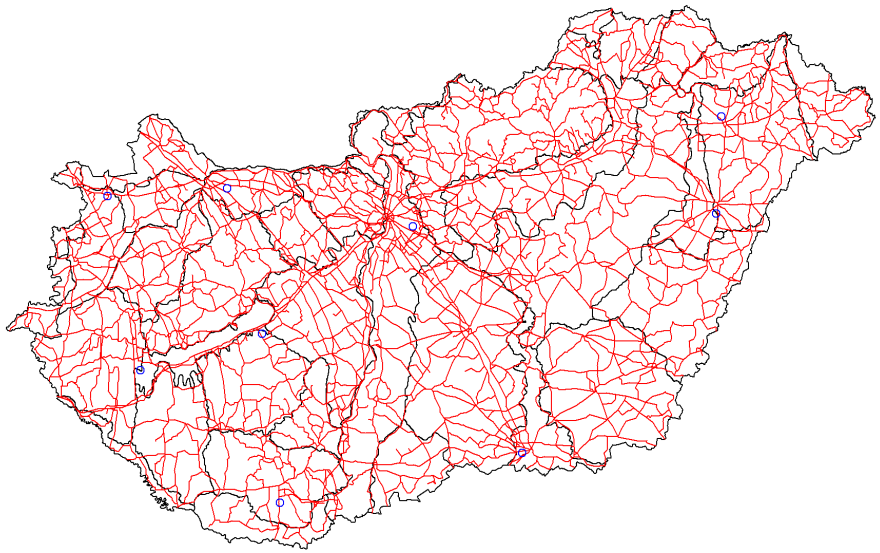
13. feladat (órai)

- Hozz létre három `sfc`-objektumot a középtájak, az utak és a repterek geometriájából.
- Jelenítsd meg ezeket sorban, egymás után úgy, hogy a 2. és 3. `plot()` eredményét a már megnyitott képvászonra illeszted.
- Az utak geometriáját pirossal, míg a repterekét kékkel ábrázold.



13. feladat (órai) – megoldás

```
kozeptajak_geometria <- st_geometry(kozeptajak)
utak_geometria <- st_geometry(utak)
repterek_geometria <- st_geometry(repterek)
plot(kozeptajak_geometria)
plot(utak_geometria, col = "red", add = TRUE)
plot(repterek_geometria, col = "blue", add = TRUE)
```



14. (összefoglaló) feladat (házi)

- Olvasd be az “eghajlat_pontosvesszo_tvesszo.csv” nevű, pontosvesszővel tagolt, tizedesvesszőt alkalmazó, fejlécsorral bíró szövegfájlt.
- Szűrd le az “eov_x”, “eov_y” és, “P_MAM” (tavaszi csapadék) nevű oszlopait.
- Készíts e szűkített táblázatból Simple Features-t a mefelelő vetületet és koordináta-oszlopokat használva.
- Ebből készíts egy, csak a geometriát tartalmazó változót, és mentsd el “tavaszi_csapadek_geometria.RData” néven.
- Jelenítsd meg az utak geometriáját kék színnel, és új rétegként ábrázold rajtuk a tavaszi csapadék geometriáját piros pontjelekkel.
- Megegyeznek-e a szűkített táblázat oszlopnevei annak a táblázatnak az oszlopneveivel, amelyet a Simple Features-ből a geometria elhagyásával nyerünk? Miért?
- Új táblázatba másold ki a tavaszi csapadék koordinátáit.

14. (összefoglaló) feladat (házi) – megoldás

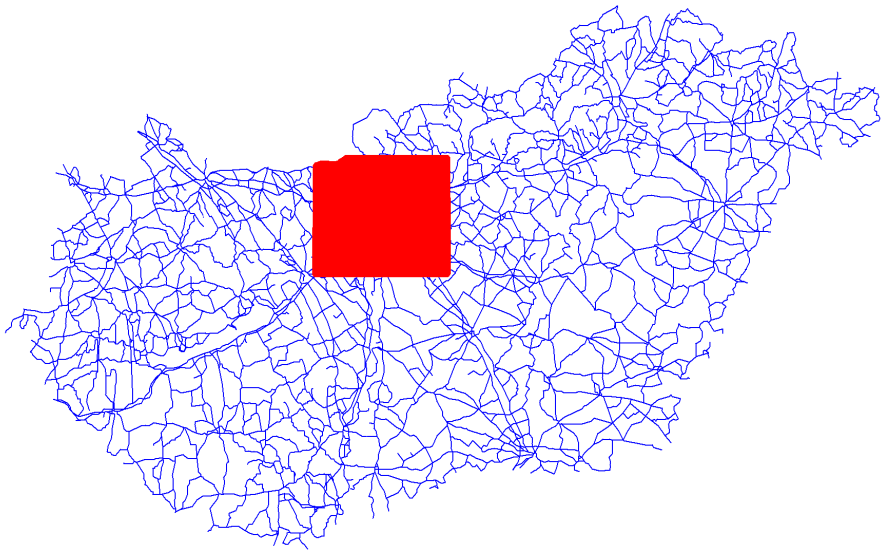
```
tavaszi_csapadek_tablazatkent <- read.table(file =
 "eghajlat_pontosvesszo_tvesszo.csv", sep = ";", dec =
 ",", header = TRUE)
```

```
tavaszi_csapadek_tablazatkent <-
 tavaszi_csapadek_tablazatkent[, c("eov_x", "eov_y",
 "P_MAM")]
tavaszi_csapadek <- st_as_sf(x =
 tavaszi_csapadek_tablazatkent, crs = 23700, coords =
 c("eov_y", "eov_x"))
tavaszi_csapadek_geometria <- st_geometry(tavaszi_csapadek)
```

```
save(tavaszi_csapadek_geometria, file =
 "kimenet/tavaszi_csapadek_geometria.RData")
```

14. (összefoglaló) feladat (házi) – megoldás

```
plot(st_geometry(utak), col = "blue")
plot(tavaszi_csapadek_geometria, col = "red", add = TRUE)
```



14. (összefoglaló) feladat (házi) – megoldás

Nem egyeznek meg, mert a két koordináta-oszlopot nem az adatmezők között tárolja a Simple Features.

```
identical(colnames(tavaszi_csapadek_tablazatkent),
 colnames(st_drop_geometry(tavaszi_csapadek)))
```

```
[1] FALSE
```

```
koordinatak <- st_coordinates(tavaszi_csapadek)
```