

Tvorba geometrie

αγεωμετρητος μηδεις εισιτω

# Agenda

- 1) Hierarchie objektů: `sf`, `sfc`, `sfg`
- 2) Tvorba vlastních objektů
- 3) Prostorová informace  
(vzdálenost) jako vstup pro  
statistické zpracování

# Hierarchie objektů

- `sf` objekt kombinuje prostor a data
- `sfc` objekt je čistá prostorová složka / může mít více prvků
- `sfg` objekt je 1 prvek z `sfc`

# Konstrukce a destrukce

- sfc ze sf

```
st_geometry()
```

- sfc z sfg

```
st_sf( .. , crs )
```

- sfg z sfc

```
[[ ]](subset)
```

- sf z sfc

```
st_as_sf()
```

- data.frame z sf

```
st_drop_geometry()
```

# Hierarchie objektů

*sf*

```
Simple feature collection with 14 features and 3 fields
geometry type:  GEOMETRY
dimension:      XY
bbox:           xmin: 12.09066 ymin: 48.5549 xmax: 18.85927 ymax: 51.05508
geographic CRS: WGS 84
```

First 10 features:

	KOD_KRAJ	KOD_CZNUTS3	NAZ_CZNUTS3	geom
1	3026	CZ020	Středočeský kraj	POLYGON ((14.93205 49.54945 ...
2	3034	CZ031	Jihočeský kraj	POLYGON ((14.72344 48.67577 ...
3	3042	CZ032	Plzeňský kraj	POLYGON ((13.55091 48.96773 ...
4	3051	CZ041	Karlovarský kraj	POLYGON ((12.55076 49.9054, ...
5	3069	CZ042	Ústecký kraj	POLYGON ((13.40684 50.08968 ...
6	3077	CZ051	Liberecký kraj	POLYGON ((15.58642 50.51438 ...
7	3085	CZ052	Královéhradecký kraj	POLYGON ((16.13918 50.06282 ...
8	3093	CZ053	Pardubický kraj	POLYGON ((16.8041 49.59882, ...
9	3107	CZ063	Kraj Vysočina	POLYGON ((15.58682 48.94701 ...
10	3115	CZ064	Jihomoravský kraj	POLYGON ((17.1504 49.13337, ...

*sfc*

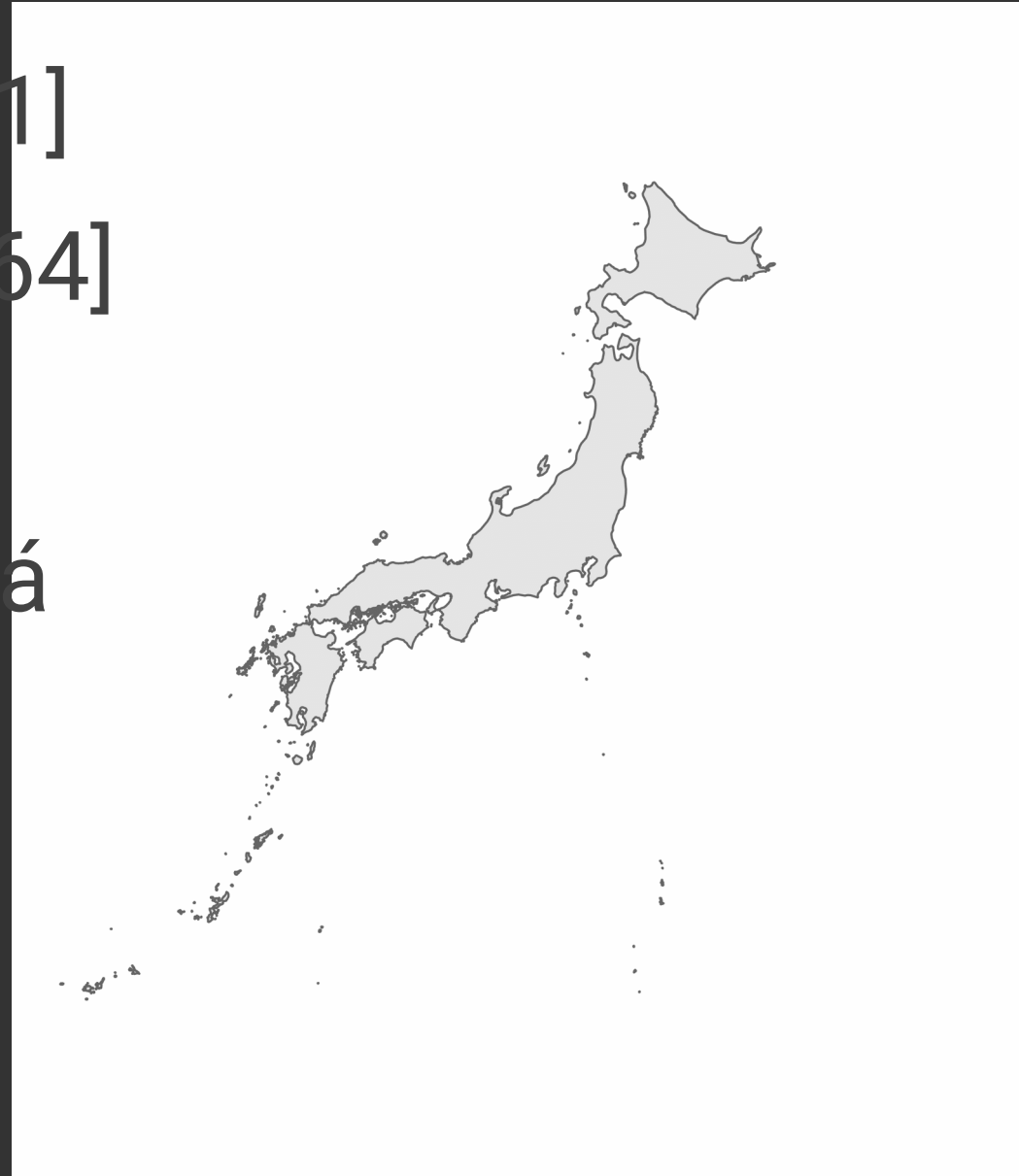
*sfg*

# Řešený příklad

**3-hierarchie.R**

# Co je celek, co je část?

- Japonské císařství [1]
- Polygony ostrovů [364]
- Prefektury [47]
- Technika jednoduchá
- Kontext klíčový :)



# Co je celek, co je část?

- Sloučení částí v celek

- `sf::st_union()` logické řešení
- `dplyr::summarise()` nečekaně elegantní v rámci pipeline / agregace

- Rozdělení celku v části

- `sf::st_cast(x, to = ...)` kde to je podle typu geometrie (multi >> single)



# Objekt typu buffer

- Okolí bodu (čáry, polygonu) vymezené vzdáleností
- Pro správné fungování vyžaduje projekci do plochy (inž. Křovák)
- `sf::st_buffer()`

# Objekt typu convex hull

- Vymezí konvexní \* prostor daný argumentem (body, polygony)
- Dobré pro vymezení "nadmnožiny"
- `sf::st_convex_hull()`
- Alternativa – konkáv v konvex:  
`concaveman::concaveman()`

# Centroid z polygonu

- Středový bod = centroid
- Více algoritmů (díry?)
- `sf::st_centroid()`
- Praktická redukce polygonů na body / odstraní plochu, zjednoduší vzdálenosti + relace více entit

# Voroného polygon z bodů

- Polygon vymezený vzdáleností od bodu (daného vstupem)
- `sf::st_voronoi()`
- Praktická transformace bodů na polygony (když se nehodí buffer, a potřebuju plochu)

# Mřížka / grid

- Klíčový vstup pro modelovací techniky – společný jmenovatel
- V geometrickém CRS pozor – nestejně velká pole (lichoběžníky)
- `sf::st_make_grid()`

# Řešený příklad

**3-tři-školy.R**

# Dělení polygonu

- Průsečík – společná část dvou prostorových objektů
- `sf::st_intersection()`
- Rozdíl – část jednoho polygonu, neobsažená v druhém
- `sf::st_difference()`

# Řešený příklad

**3-praha-brno.R**



# Dělení polygonu

- Rozseknutí polygonu čarou
- Pozor – rozbije datovou složku
- `lwgeom::st_split()`

# Přenos dat při průsečíku

- Pro každý polygon známe plochu
- Předpoklad rovnoměrné hustoty umožňuje použít trojčlenku
- $\text{podíl metriky} = \text{původní metrika} \times \text{podíl plochy průsečíku z původní}$
- `sf::st_interpolate_aw()`

# Příklad: levý a pravý břeh

- Počet obyvatel levého a pravého břehu, když známe obyvatele pražských částí [57]
- Problém:
  - Počet obyvatel částí Prahy ze sčítání lidu
  - Polygony levého a pravého břehu Vltavy
  - Interpolovat data podle plochy
  - Podat zprávu

# Řešený příklad

**3-obyvatele-břehů.R**

# Zjednodušení polygonů



- Výhody: elegantnější tvar, menší

objekt, rychlejší vykreslení

- `rmapshaper::ms_simplify()`

# Řešený příklad

**3-simple-česko.R**

# Externí zdroj dat: HERE

- HERE / dříve Nokia Maps (IT svět)
- Aktivně hledá vývojáře mobilních aplikací (\$\$\$)
- 250 tisíc dotazů měsíčně zdarma
- Vedlejší efekt (kvůli erkařům to nedělají): `{hereR}` nad jejich REST API

# Externí zdroj dat: HERE

- Geocoding `hereR::geocode()`
- Routing `hereR::route()`
- Isochrony `hereR::isoline()`



# Řešený příklad

**3-routing-here.R**

# Příklad: pivo po škole

- U které školy dát sraz na pivo?
  - Matfyz × VŠE × Přfuk?
- Problém:
  - Zaměřit školy, identifikovat hospody
  - Získat isochrony / 10 minut pěšky
  - Spočítat body hospod v polygonu isochron
  - Podat zprávu

# Řešený příklad

**3-tři-školy-here.R**

# Interpretace "vzdálenosti"

- Co znamená vzdálenost?
  - **Spojitou veličinu** – čím víc, tím víc?  
limituje k 20 015 Km ( $6\,371 \times \pi$ )
  - **Kategorickou veličinu** – blízko ano / ne?
- Přístupy k řešení:
  - `sf::st_distance()`
  - `sf::st_intersects(sf::st_buffer(x))`

# Příklad: benzín na dálnici

- Je na benzín v okolí dálnice dražší?
- Problém:
  - Zjistit ceny benzínu, zaměřit pumpy
  - Zaměřit dálnice, definovat okolí
  - Zkombinovat ceny a okolí
  - Standardní statistika :)

# Řešený příklad

**3-benzín.R**

# Příklad: polelány...

- Souvisí počet traktorů registrovaných v obci s plochou ornice?
- Problém:
  - Zjistit počet traktorů z open dat [mdcr.cz](https://mdcr.cz)
  - Zaměřit ornou půdu z Copernikus satelitů
  - Přenést registrace + ornici na obce ČR
  - Standardní statistika :)

# Řešený příklad

**3-cropland-traktory.R**



# Klíčové body

- Hierarchie objektů: `sf` `sfc` `sfg`
- Celek a část / entity a metriky
- Tvorba vlastní geometrie
  - Buffer, grid, Voroného polygony
  - Průsečíky polygonů
- Interpolace hodnot podílem na ploše