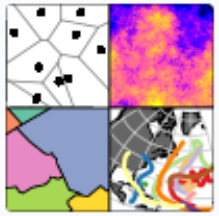
Les progrès de r-spatial

Etienne Racine

Search or jump to… / Pull requests Issues Marketplace Explore



r-spatial

For packages raster, terra, dismo & geosphere visit the rspatial github organisation (mind the missing '-') or click the link below

https://github.com/rspatial

Repositories 31 | People 12 | Teams 2 | Projects 1 | Settings

Pinned repositories

Customize pinned repositories

mapview
Interactive viewing of spatial data in R
R ★232 49

sf
Simple Features for R
R ★510 130

stars
Spatiotemporal Arrays, Raster and Vector Data Cubes
R ★228 27

mapedit
Interactive editing of spatial data in R
R ★126 16

r-spatial.org
r-spatial.org blog sources
HTML ★26 18

discuss
a discussion repository: raise issues, or contribute!
★18


R consortium

Depuis la création de `sf`...

- Compatibilité améliorée
- Gain de vitesse
- Adopté par plusieurs autres packages

Intégration avec ggplot

```
library(tidyverse)
library(sf)

storms <-
  system.file("shape/storms_xyz_feat
  read_sf() %>%
  st_set_crs(4326)

storms %>%
  arrange(-st_length(geometry)) %>%
  slice(1:12) %>%
  ggplot() +
  geom_sf(data = storms %>%
            select(-Track),
          color = "lightgrey") +
  geom_sf(color = "blue") +
  facet_wrap(~Track)
```

Mapview

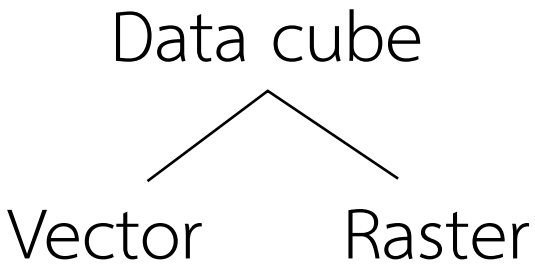
```
library(mapview)

storms %>%
  mutate(length = st_length(geometry)) %>%
  mapview()
```

Postgis

```
library(DBI)
con <- dbConnect(RPostgres::Postgres())

resto <- read_sf(con, "restaurant")

resto <- tbl(con, "restaurant")

resto %>%
  filter(st_area(geometry) > 3e3) %>%
  summarise(tot_area = geometry %>% st_union() %>% st_area() / 1e6)

# A tibble: 1 x 1
  tot_area
     <dbl>
1   30002.
```

Stars

Spatiotemporal Arrays, Raster and Vector Data Cubes



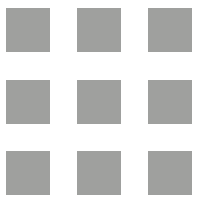
- Cube **raster**: x et y prennent une dimension spatiale
- Cubes données **vectorielles**: où un ensemble de géométries de caractéristiques (points, lignes, polygones) forment les valeurs d'au moins une dimension

Data Cubes

les cubes de données sont des matrices où des valeurs sont données pour chaque combinaison des valeurs de dimension.

Exemples:

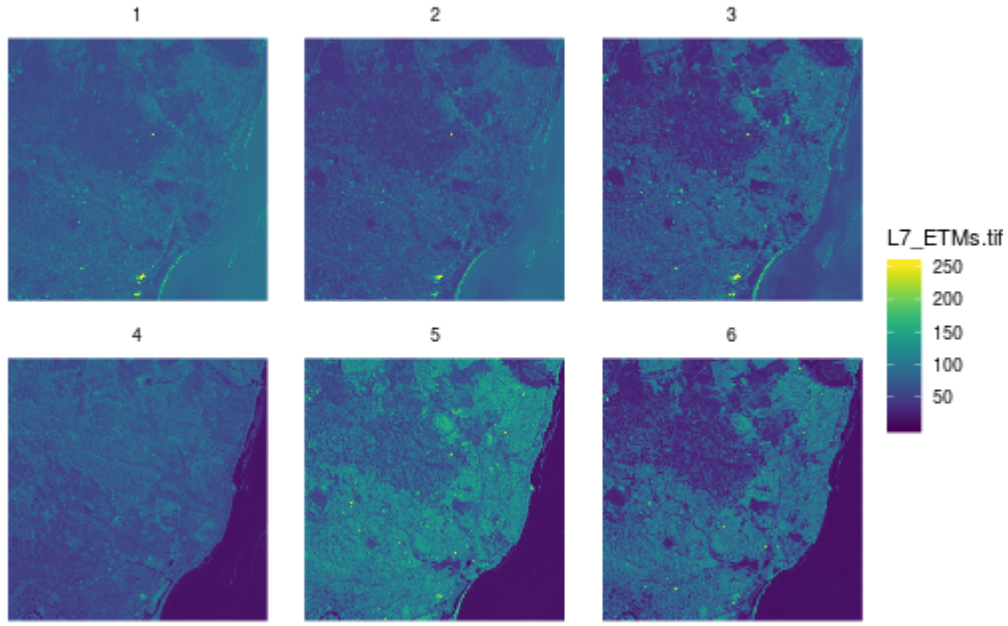
- ventes par produit, magasin et semaine;
- population par sexe, classe d'âge, région et recensement;
- température par coordonnée x, y, z et temps;
- série d'images satellites multi-bande;
- prévision par heure de prévision, heure et emplacement (x, y, z);
- mesures répétées d'un lidar.



`stars`: cubes de données raster et vectorielles

- Les cubes ont des dimensions:
  - régulières (décalage, résolution)
  - valeurs (e.g., dates ou des géométries)
  - unités de mesure, systèmes de référence de référence (PROJ)
- lire et écrire tout format pris en charge par GDAL
- peut lire (directement) netcdf
- des supports distribué (stars_proxy), et
- rendra possible le traitement sur le cloud

```
library(stars)
library(viridis)
library(ggplot2)

x <- system.file("tif/L7_ETMs.tif", package = "stars") %>%
  read_stars()
ggplot() +
  geom_stars(data = x) +
  coord_equal() +
  facet_wrap(~band) +
  scale_fill_viridis()
```

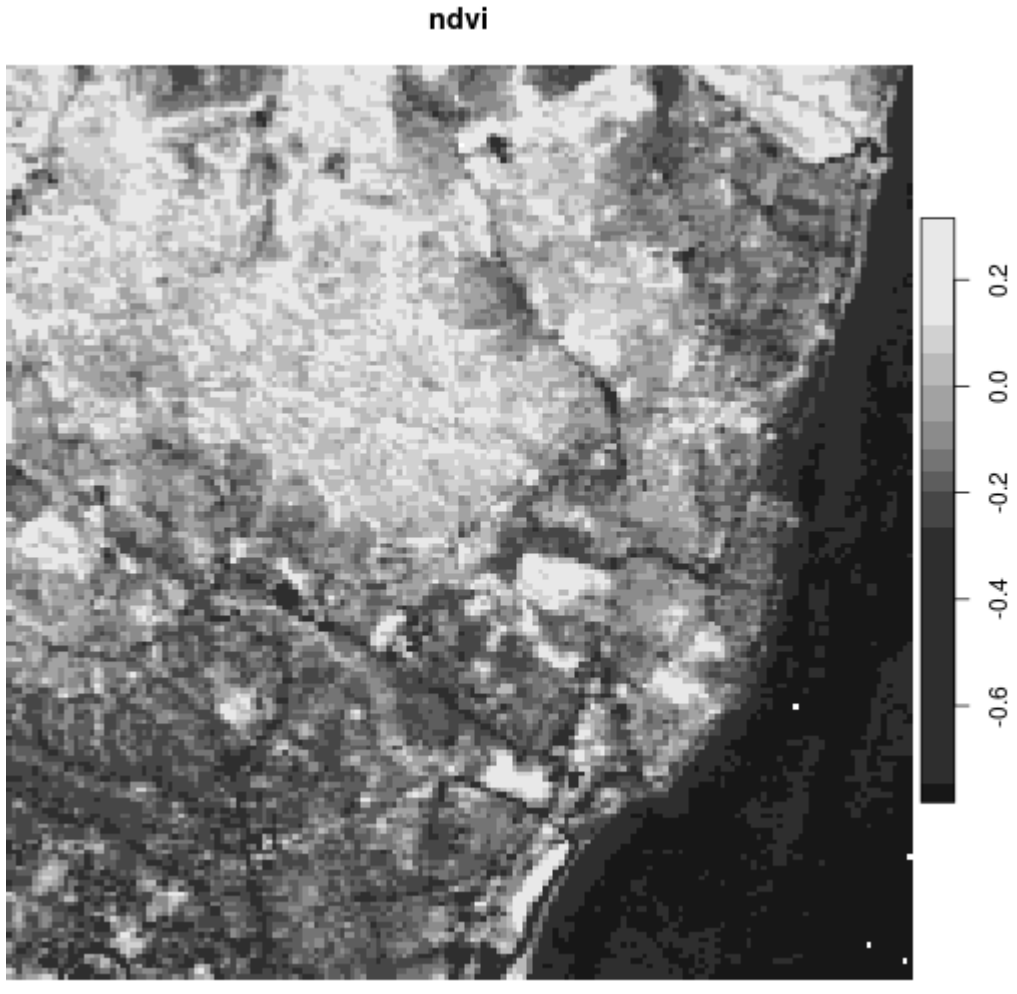

1
2
3
4
5
6
L7_ETMs.tif
250
200
150
100
50

Calcul à la demande

```
x <- system.file("tif/L7_ETMs.tif", package = "stars") %>%
  read_stars(proxy = TRUE)
ndvi <- function(x) (x[4]-x[1])/(x[4] + x[1])
system.time(x_ndvi <- st_apply(x, c("x", "y"), ndvi))
```

```
##    user  system elapsed
##   0.001   0.000   0.001
```

```
# les instructions sont réordonnées:
# reads (downsampled), run `ndvi`, plot
plot(x_ndvi)
```


ndvi
0.2
0.0
-0.2
-0.4
-0.6

Conclusion

- `sf` pour les vecteurs
- `stars` pour les `data cube`
- `stars` calcul hors mémoire, lazy et sur le cloud
- Permet l'analyse **interractive** (avec `mapview`)
- Spatial Data Science upcoming book: https://www.r-spatial.org/book/

https://edzer.github.io/rstudio_conf/2019/index.html