

Introdução ao uso de dados geoespaciais no R

9 Visualização de dados geoespaciais

Maurício H. Vancine

Milton C. Ribeiro

UNESP - Rio Claro

Laboratório de Ecologia Espacial e Conservação (LEEC)

25/10/2021-05/11/2021

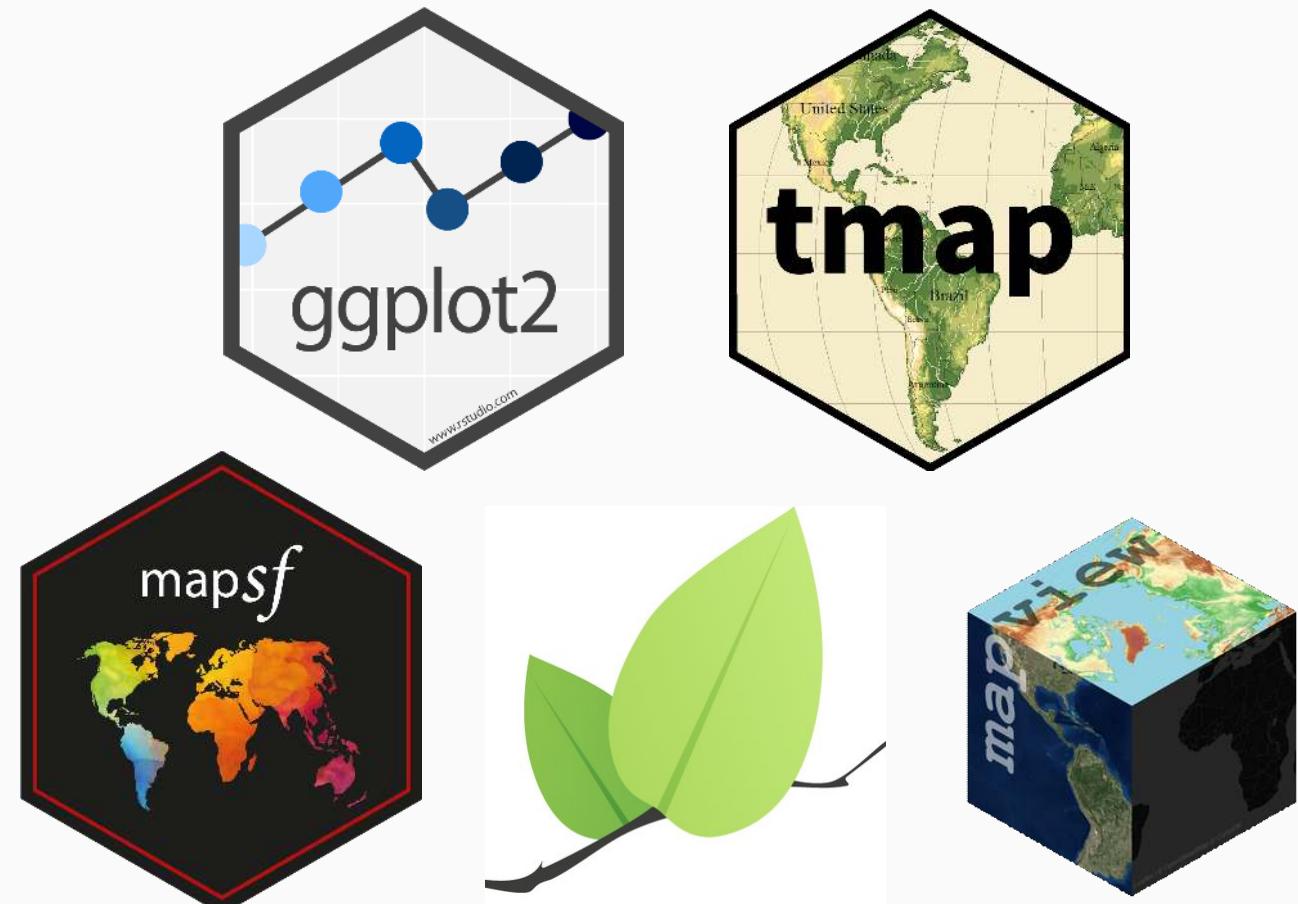


9 Visualização de dados geoespaciais

Conteúdo

1. Elementos de um mapa
2. Pacotes para produção de mapas
3. Mapas estáticos
4. Mapas animados
5. Mapas interativos

[ggplot2](#), [tmap](#), [maps](#), [leaflet](#), [mapview](#)



9 Visualização de dados geoespaciais

Script

```
09_script_intro_geoespacial_r.R
```

1. Elementos de um mapa

Elementos

- Mapa principal
- Mapa secundário
- Título
- Legenda
- Barra de escala
- Indicador de orientação (Norte)
- Grade de coordenadas
- Descrição do Sistema de Referência de Coordenadas
- Informações de origem
- Elementos auxiliares



2. Pacotes para produção de mapas

Principais pacotes

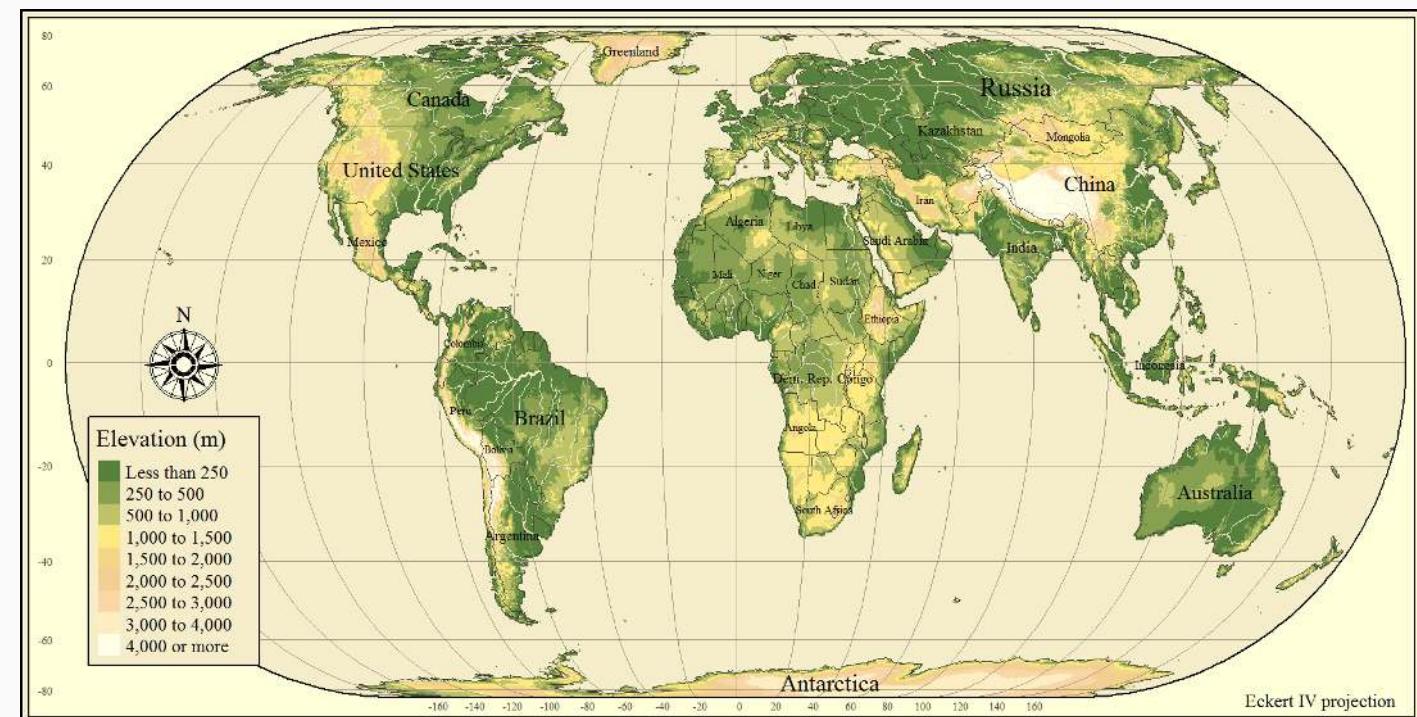
- [ggplot2](#): cria mapas elegantes usando a Gramática de Gráficos
- [ggspatial](#): estrutura de dados espaciais para ggplot2
- [ggmap](#): visualização espacial com ggplot2
- [tmap](#): cria mapas temáticos
- [maps](#): cria mapas temáticos com dados vetoriais
- [linemap](#): cria mapas de linhas
- [cartography](#): cartografia temática
- [cartogram](#): métodos de visualização para cartogramas
- [googleway](#): acessa Google Maps APIs para recuperar dados e traçar mapas
- [rasterVis](#): métodos de visualização para dados raster
- [mapview](#): visualização interativa de dados espaciais em R
- [leaflet](#): cria Web Maps interativos com Leaflet
- [plotly](#): crie gráficos interativos da Web por meio de 'plotly.js'
- [mapedit](#): edição interativa de dados geoespaciais em R

3. Mapas estáticos

Mapas estáticos são mapas simples e fixos para visualização de dados

Principais

1. `plot()` (vários pacotes)
2. `ggplot2`
3. `ggmap`
4. `tmap`
5. `maps`
6. `linemap`
7. `cartography`
8. `cartogram`
9. `googleway`
10. `rasterVis`



3. Mapas estáticos

Função `plot()`

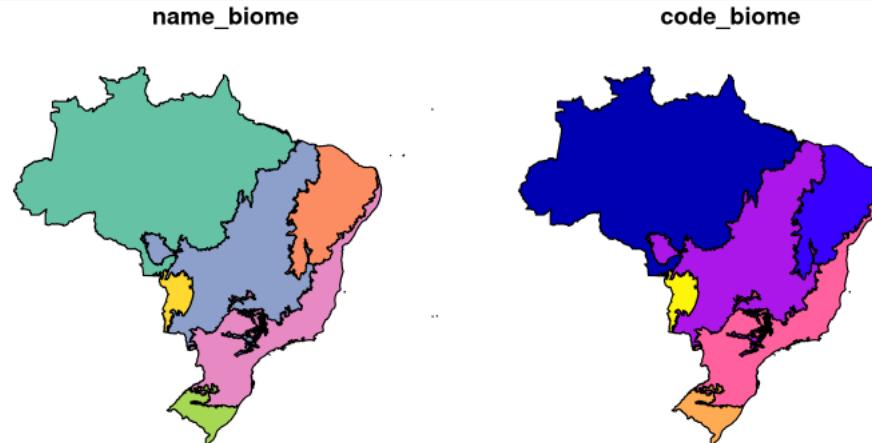
```
# biomas
biomas ← geobr::read_biomes(showProgress = FALSE) %>%
  dplyr::filter(name_biome ≠ "Sistema Costeiro")
biomas

## Simple feature collection with 6 features and 3 fields
## Geometry type: MULTIPOLYGON
## Dimension: XY
## Bounding box: xmin: -73.98304 ymin: -33.75115 xmax: -28.84785 ymax: 5.269581
## Geodetic CRS: SIRGAS 2000
##          name_biome code_biome year           geom
## 1      Amazônia         1 2019 MULTIPOLYGON (((-44.08515 - ...
## 2     Caatinga         2 2019 MULTIPOLYGON ((((-41.7408 -2 ...
## 3      Cerrado         3 2019 MULTIPOLYGON ((((-43.39009 - ...
## 4 Mata Atlântica       4 2019 MULTIPOLYGON ((((-48.70814 - ...
## 5       Pampa          5 2019 MULTIPOLYGON ((((-52.82472 - ...
## 6    Pantanal         6 2019 MULTIPOLYGON ((((-57.75946 - ...
```

3. Mapas estáticos

Função `plot()`

```
# plot  
plot(biomas)
```

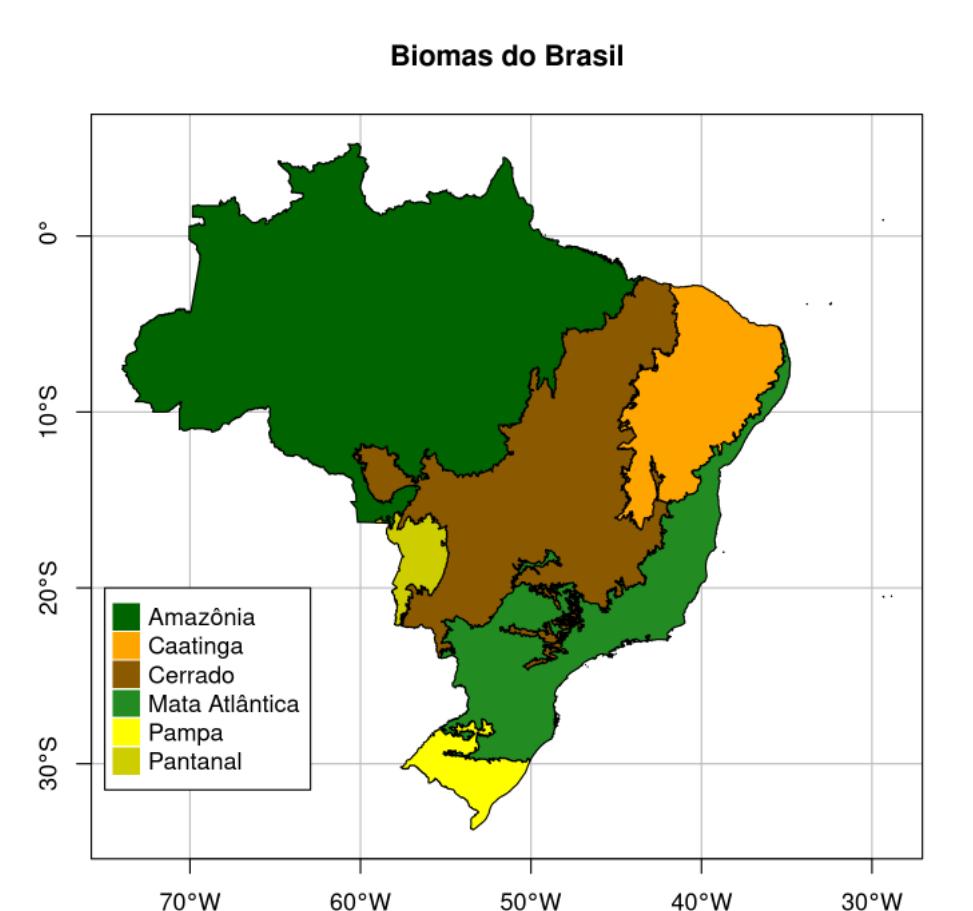


3. Mapas estáticos

Função `plot()`

```
# plot
plot(biomas$geom,
      col = c("darkgreen", "orange", "orange4",
             "forestgreen", "yellow", "yellow3"),
      main = "Biomas do Brasil", axes = TRUE,
      graticule = TRUE)

# legend
legend(x = -75, y = -20, pch = 15, cex = 1,
       pt.cex = 2.5, legend = biomas$name_biome,
       col = c("darkgreen", "orange", "orange4",
              "forestgreen", "yellow", "yellow3"))
```



3. Mapas estáticos

Função `plot()`

```
# rio claro
rc_2020 ← geobr::read_municipality(code_muni = 3543907, year = 2020, showProgress = FALSE) %>%
  sf::st_transform(crs = 4326)
rc_2020

## Simple feature collection with 1 feature and 7 fields
## Geometry type: MULTIPOLYGON
## Dimension: XY
## Bounding box: xmin: -47.76521 ymin: -22.55203 xmax: -47.46188 ymax: -22.24368
## Geodetic CRS: WGS 84
##   code_muni name_muni code_state abbrev_state name_state code_region name_region      geom
## 493    3543907 Rio Claro        35           SP    São Paulo         3     Sudeste MULTIPOLYGON (((-47.46875 - ...
```

3. Mapas estáticos

Função `plot()`

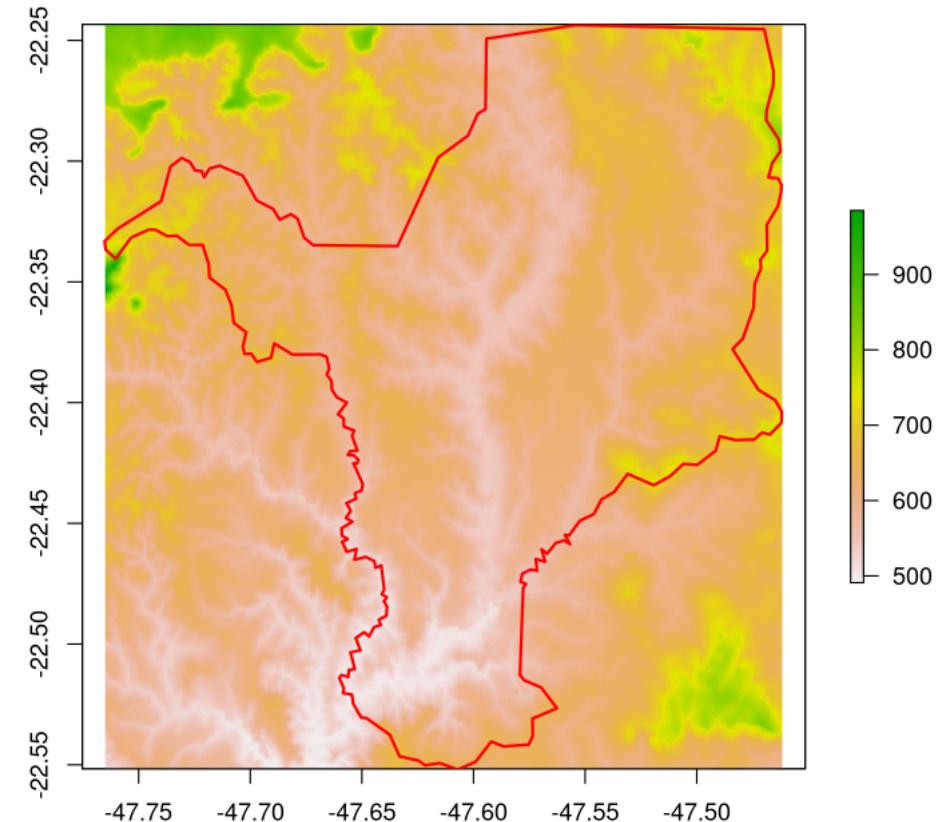
```
# importar raster
dem_rc ← raster::raster(here::here("03_dados", "raster", "srtm_27_17.tif")) %>%
  raster::crop(rc_2020)
dem_rc
```

```
## class      : RasterLayer
## dimensions : 370, 364, 134680  (nrow, ncol, ncell)
## resolution : 0.0008333333, 0.0008333333  (x, y)
## extent     : -47.765, -47.46167, -22.55167, -22.24333  (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs
## source     : memory
## names      : srtm_27_17
## values     : 491, 985  (min, max)
```

3. Mapas estáticos

Função `plot()`

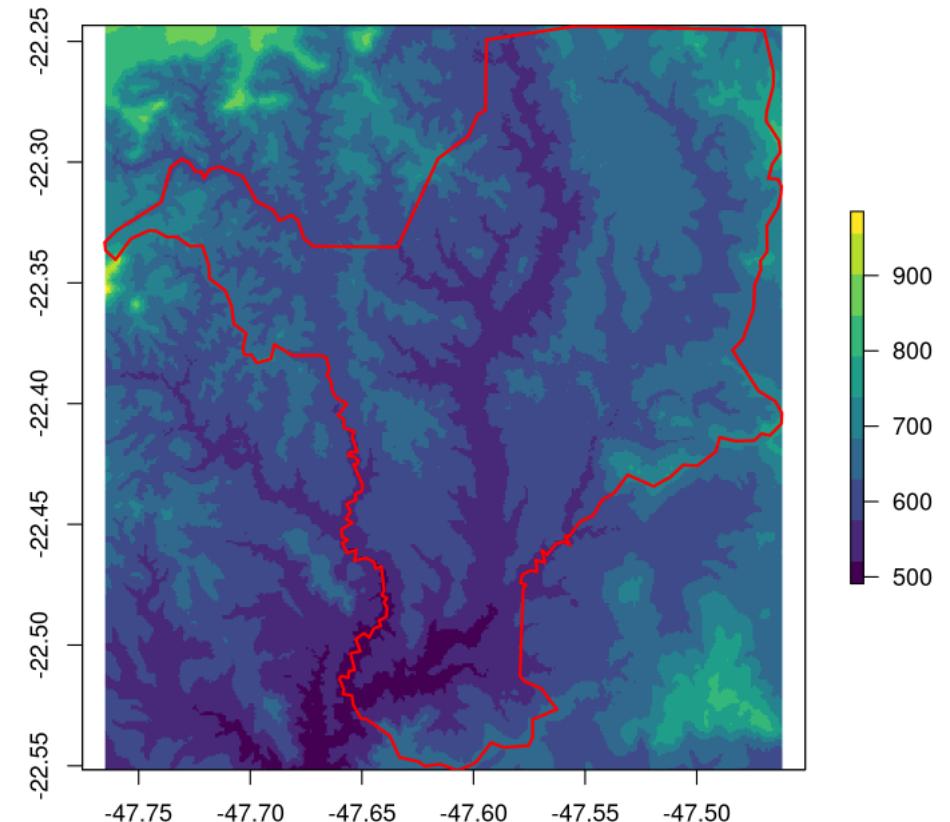
```
# plot  
plot(dem_rc)  
plot(rc_2020$geom, col = NA, border = "red", lwd = 2)
```



3. Mapas estáticos

Função `plot()`

```
# plot  
plot(dem_rc, col = viridis::viridis(10))  
plot(rc_2020$geom, col = NA, border = "red", lwd = 2)
```



3. Mapas estáticos

Função `plot()`

```
# listar arquivos
files <- dir(path = here::here("03_dados", "raster"), pattern = "wc", full.names = TRUE) %>%
  grep(".tif", ., value = TRUE)

# importar
bioclim <- raster::stack(files)

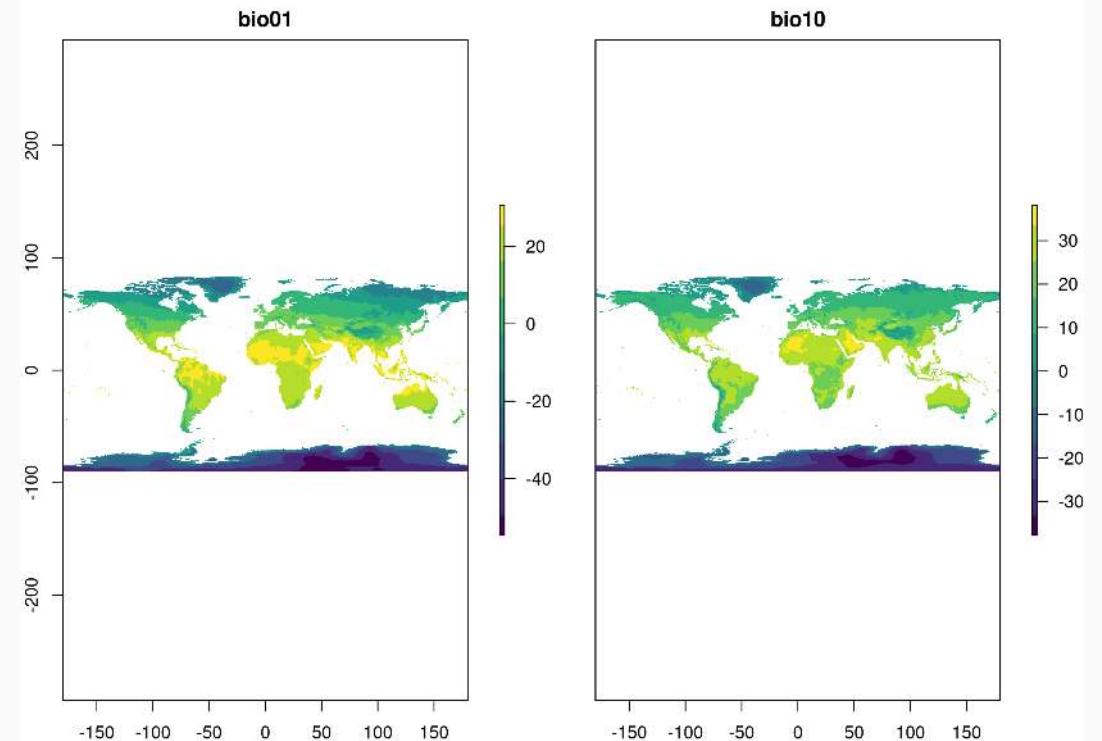
# names
names(bioclim) <- c("bio01", paste0("bio", 10:19), paste0("bio0", 2:9))
bioclim
```

```
## class      : RasterStack
## dimensions : 1080, 2160, 2332800, 19  (nrow, ncol, ncell, nlayers)
## resolution : 0.1666667, 0.1666667  (x, y)
## extent     : -180, 180, -90, 90  (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs
## names      : bio01,      bio10,      bio11,      bio12,      bio13,      bio14,      bio15,      bio16
## min values : -54.724354, -37.781418, -66.311249,  0.000000,  0.000000,  0.000000,  0.000000,  0.000000
## max values :  30.98764,   38.21617,   29.15299, 11191.00000, 2381.00000, 484.00000, 229.00169, 5284.00000
```

3. Mapas estáticos

Função `plot()`

```
# plot  
plot(bioclim[[1:2]], col = viridis::viridis(10))
```

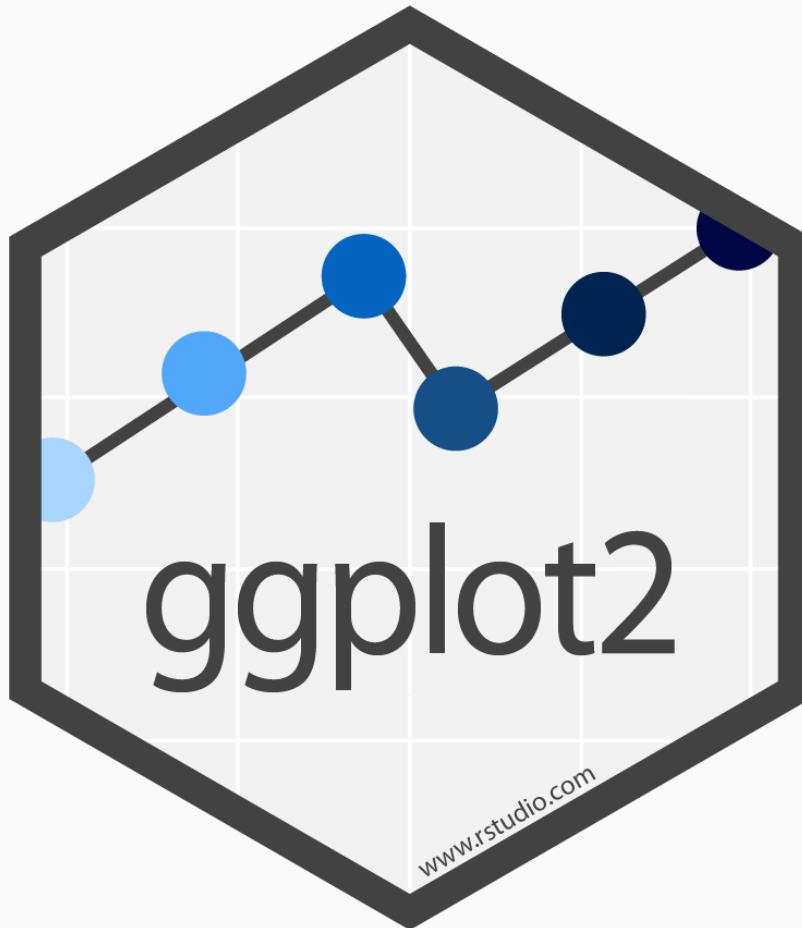


3. Mapas estáticos

Função `plot()`

```
# diretorio
dir.create(here::here("03_dados", "mapas"))

# exportar mapa
png(filename = here::here("03_dados", "mapas", "mapa_biomias.png"),
     width = 20, height = 20, units = "cm", res = 300)
plot(biomias$geom,
      col = c("darkgreen", "orange", "orange4", "forestgreen", "yellow", "yellow3"),
      main = "Biomas do Brasil", axes = TRUE, graticule = TRUE)
legend(x = -75, y = -20, pch = 15, cex = 1, pt.cex = 2.5, legend = biomias$name_biome,
       col = c("darkgreen", "orange", "orange4", "forestgreen", "yellow", "yellow3"))
dev.off()
```



[ggplot2](#)

3. Mapas estáticos

ggplot2, ggspatial e ggmap

Integrado ao **tidyverse**, possui uma sintaxe própria

Principal pacote para **visualização de dados** no R

Estrutura:

```
ggplot( ... ) +  
aes( ... ) +  
geom_( ... ) +  
stats_( ... ) +  
coord_( ... ) +  
facet_( ... ) +  
theme_( ... )
```

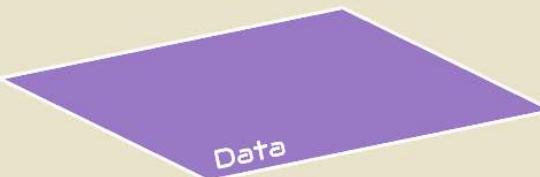
[ggplot2, @allison_horst](#)



3. Mapas estáticos

ggplot2, ggspatial e ggmap

Grammar of Graphics



xy, 3902, 29, 9,
4756, x, 72, 633,
647, 617, 827, 3,
1, 21, 45, tyu, 6,
987, 457, 283, 8,
4, 5, 671, 34, 67,
x, 981, hu, 89, 5

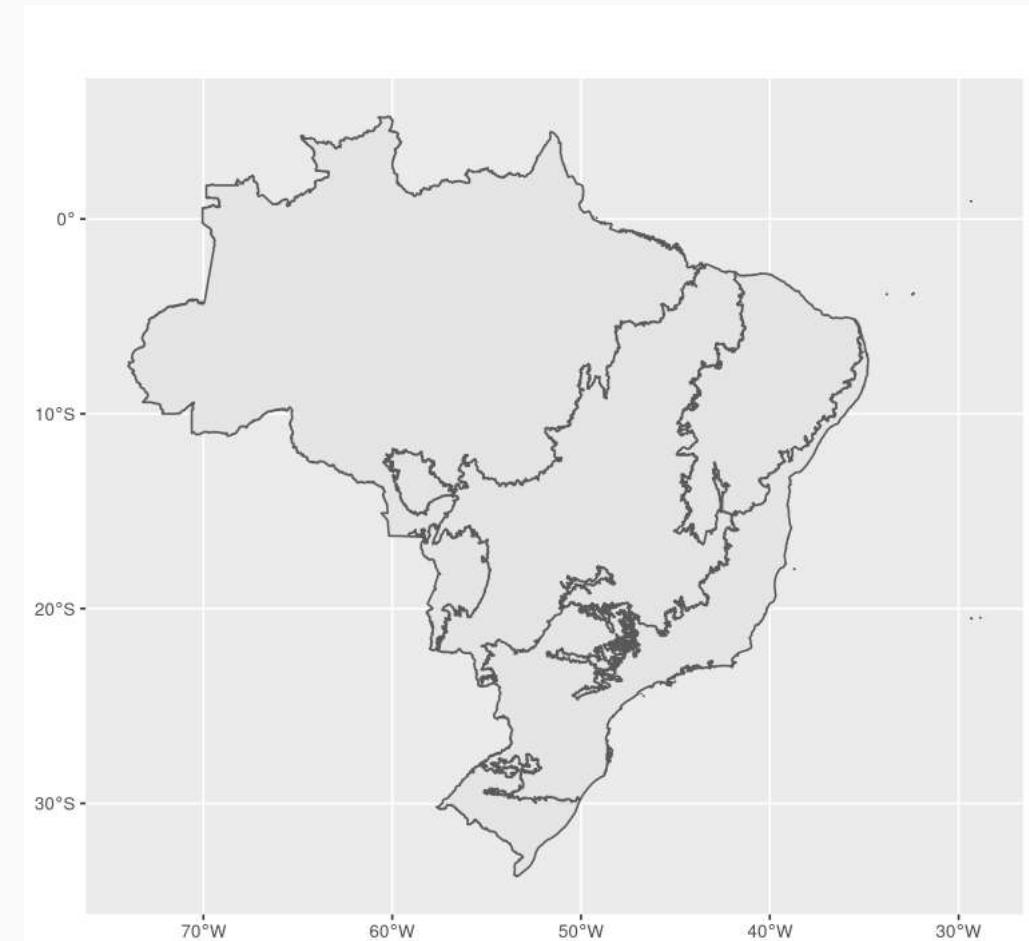
```
32: #----- ----- Visualising your boxplot -----
33:
34: # Before plotting (if not installed) install.packages("gridExtra")
35: # Then activate ggplot2 package
36: library(ggplot2)
37:
38: # Create new variable for plot only x and y axis. ('data' and 'statistics' layer)
39: plot <- ggplot(data=new.data, aes(x=Genre, y=Gross..05))
40:
41: # Create new variable with geom/statistics layer.
42: q <- plot + geom_boxplot(fill="Studio", size=Budget..$all.)
43: q + theme_minimal() # removes grid lines and adds a border around data points.
44:   + theme(panel.border = element_rect(colour = "black")) # with the border color of black.
45:   + geom_boxplot(alpha=0.7, outlier.color = NA) # places the boxplot on the data points
46:   + theme(panel.outline = element_line(size = 1, colour = "black"))
47:
48: # Change axis and title if needed.
49: q + xlab("Genre") +
50:   ylab("Gross N US") +
51:   ggtitle("Domestic Gross N by Genre")
52:
53: # Make your plot visually attractive and readable with the 'theme' function. (Theme layer)
54: q + theme(axis.title = element_text(colour = "#00aaff", size = 14),
55:           axis.text = element_text(size = 12),
56:           legend.title = element_text(size = 12),
57:           legend.text = element_text(size = 10),
58:           plot.title = element_text(size = 14, hjust = 0.5), # 'hjust' will center your text.
59:           panel.background = element_rect(fill = "#E0E0E0"))
```

[Think About the Grammar of Graphics When Improving Your Graphs](#)

3. Mapas estáticos

ggplot2 e ggspatial

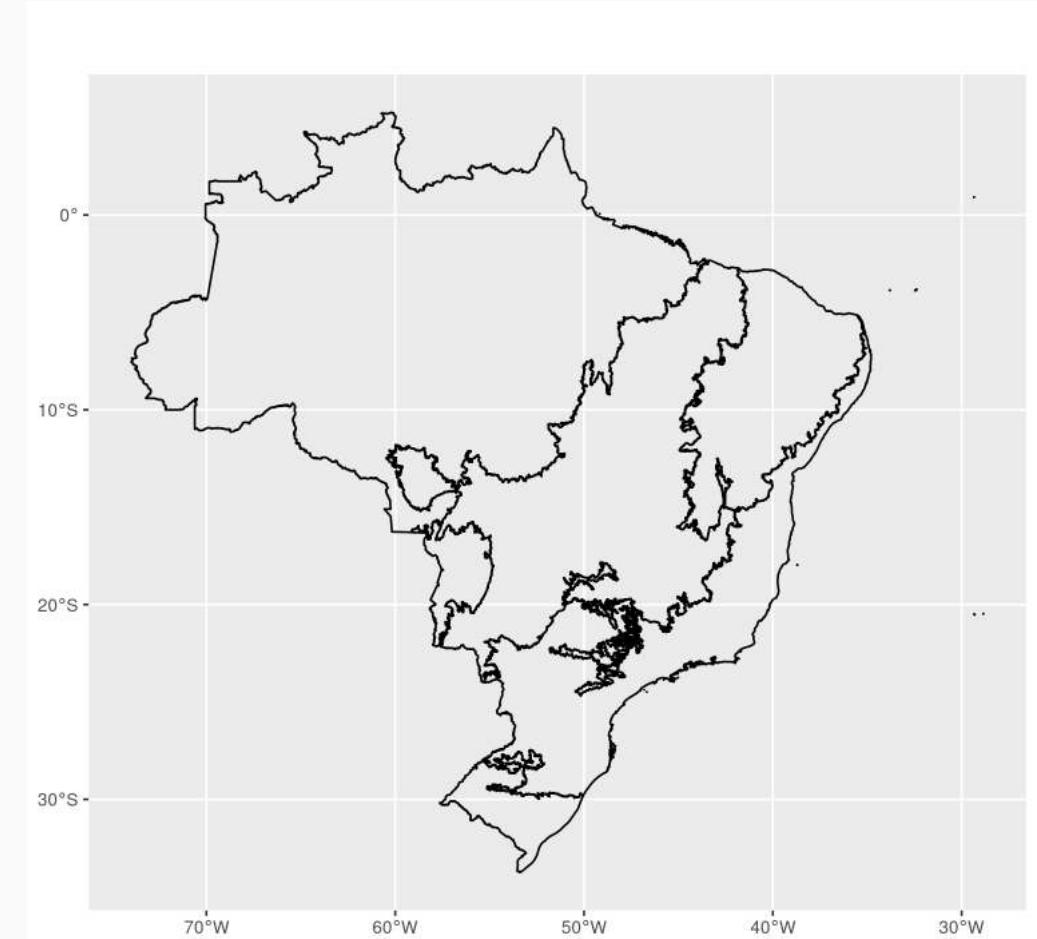
```
# dados  
ggplot() +  
  geom_sf(data = biomas)
```



3. Mapas estáticos

ggplot2 e ggspatial

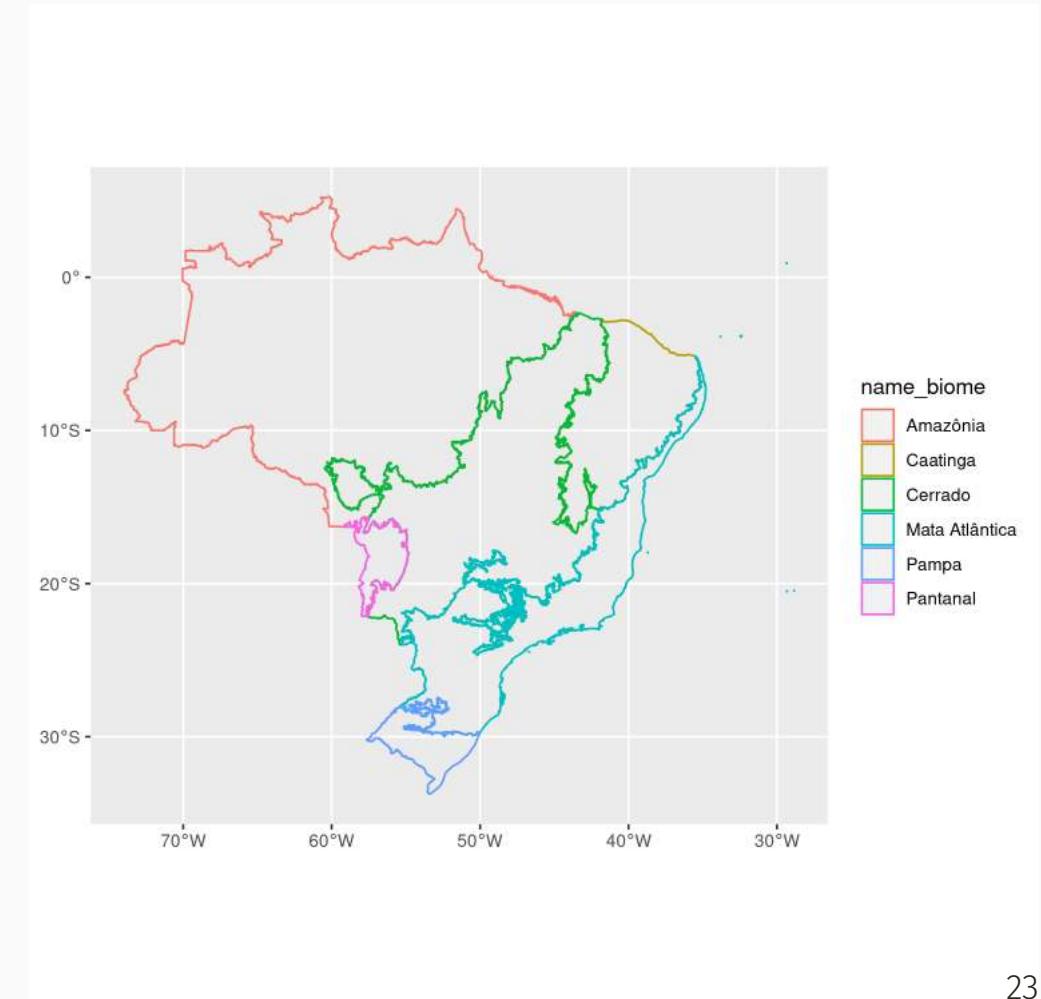
```
# cor e preenchimento  
ggplot() +  
  geom_sf(data = biomas, color = "black", fill = NA)
```



3. Mapas estáticos

ggplot2 e ggspatial

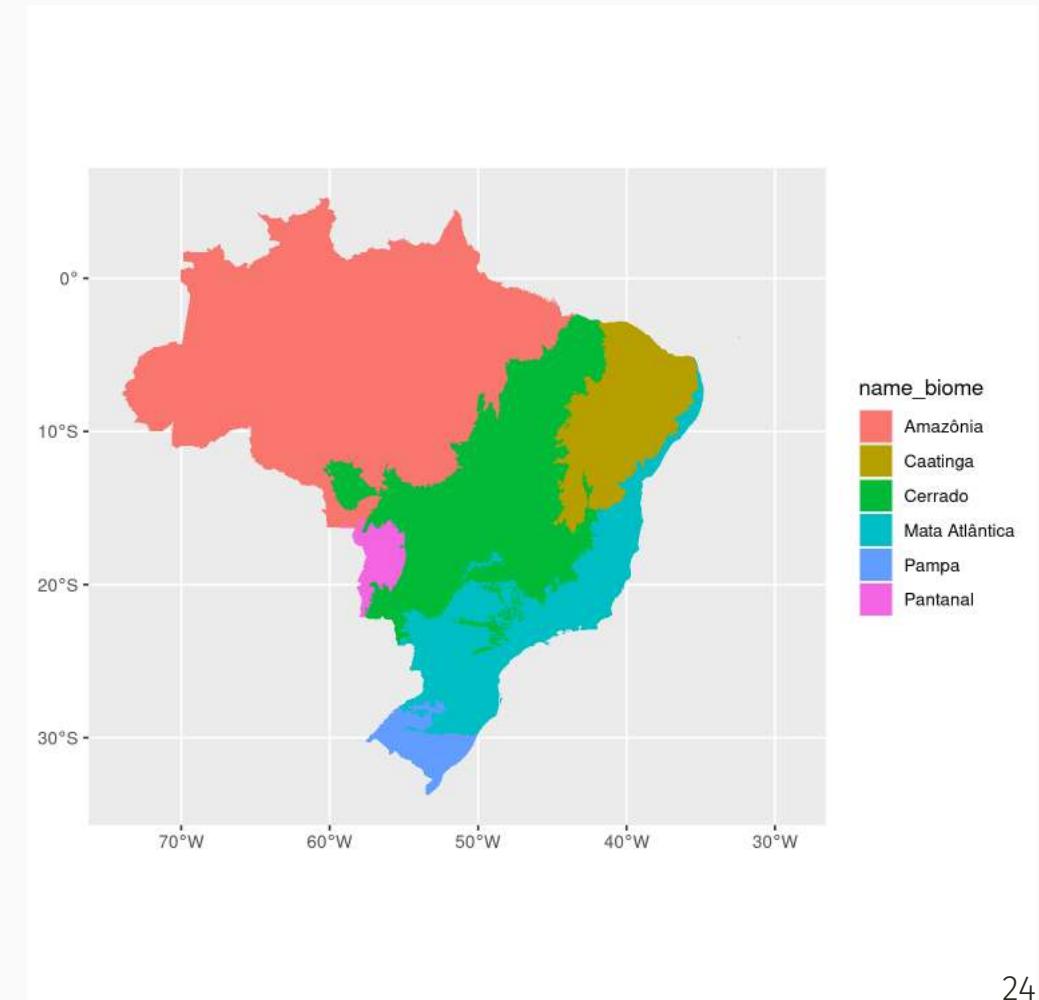
```
# cor e preenchimento  
ggplot() +  
  geom_sf(data = biomas,  
          aes(color = name_biome),  
          fill = NA)
```



3. Mapas estáticos

ggplot2 e ggspatial

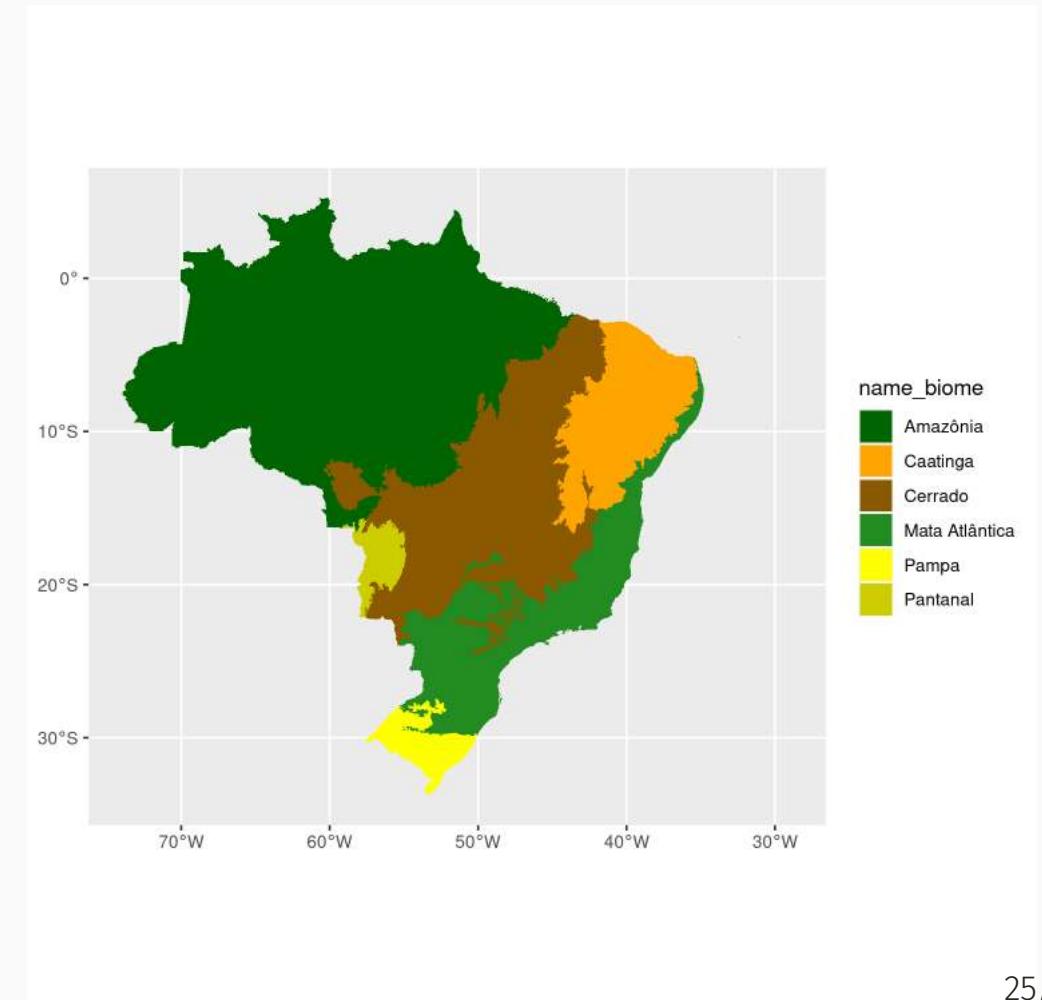
```
# cor e preenchimento  
ggplot() +  
  geom_sf(data = biomas,  
          aes(fill = name_biome),  
          color = NA)
```



3. Mapas estáticos

ggplot2 e ggspatial

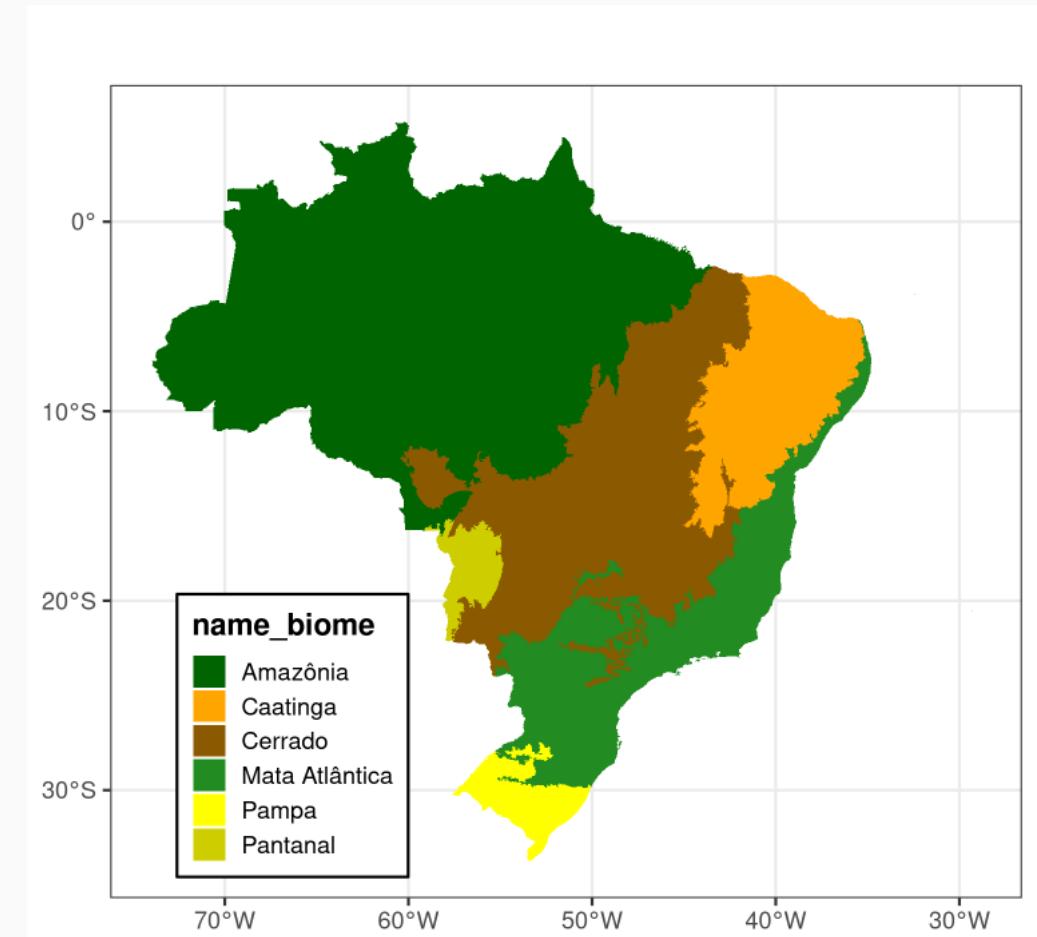
```
# definir cor
ggplot() +
  geom_sf(data = biomas, aes(fill = name_biome), color = "black")
  scale_fill_manual(values = c("darkgreen", "orange",
                               "forestgreen", "yellow",
                               "brown", "olivedrab"))
```



3. Mapas estáticos

ggplot2 e ggspatial

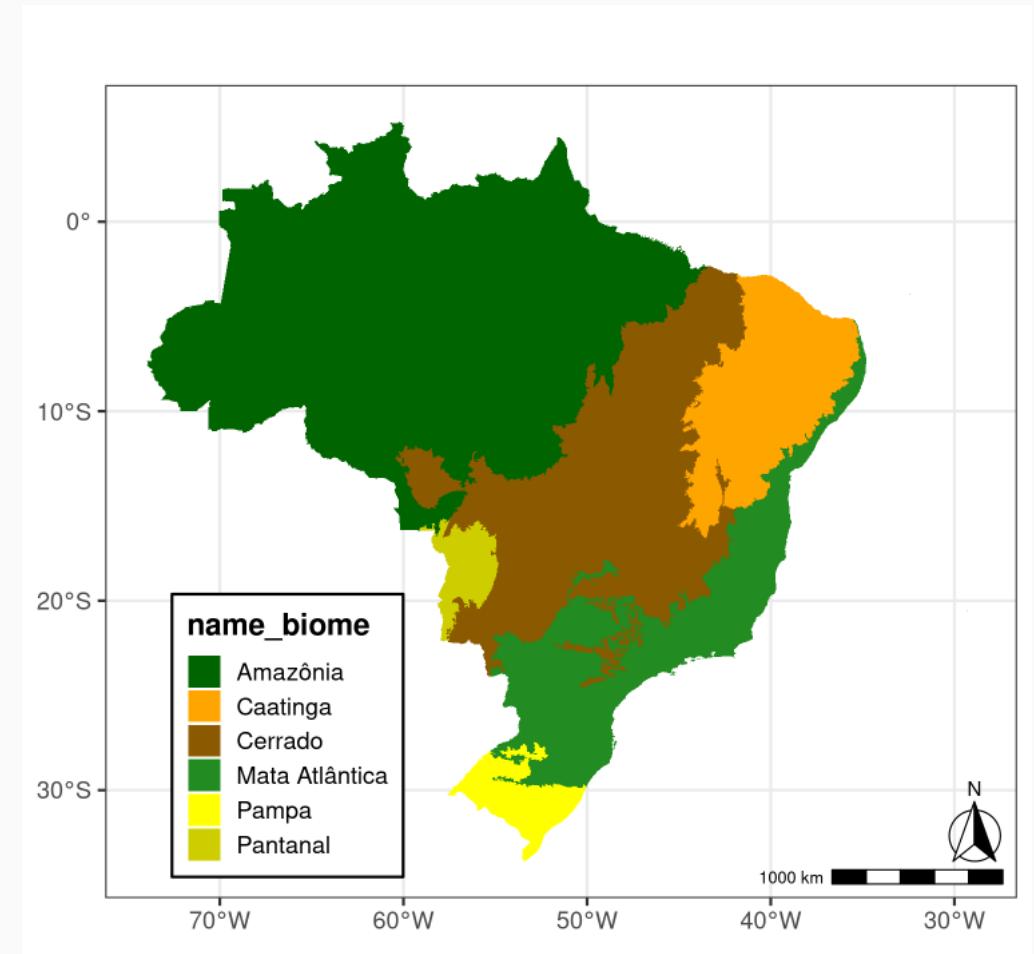
```
# tema
ggplot() +
  geom_sf(data = biomas, aes(fill = name_biome), color = "black") +
  scale_fill_manual(values = c("darkgreen", "orange",
                               "forestgreen", "yellow",
                               "#8B4513", "#FFFF00"))
  theme_bw(base_size = 15) +
  theme(legend.title = element_text(size = 15, face = "bold"),
        legend.position = c(.2, .2),
        legend.background = element_rect(colour = "black", fill = "white"))
```



3. Mapas estáticos

ggplot2 e ggspatial

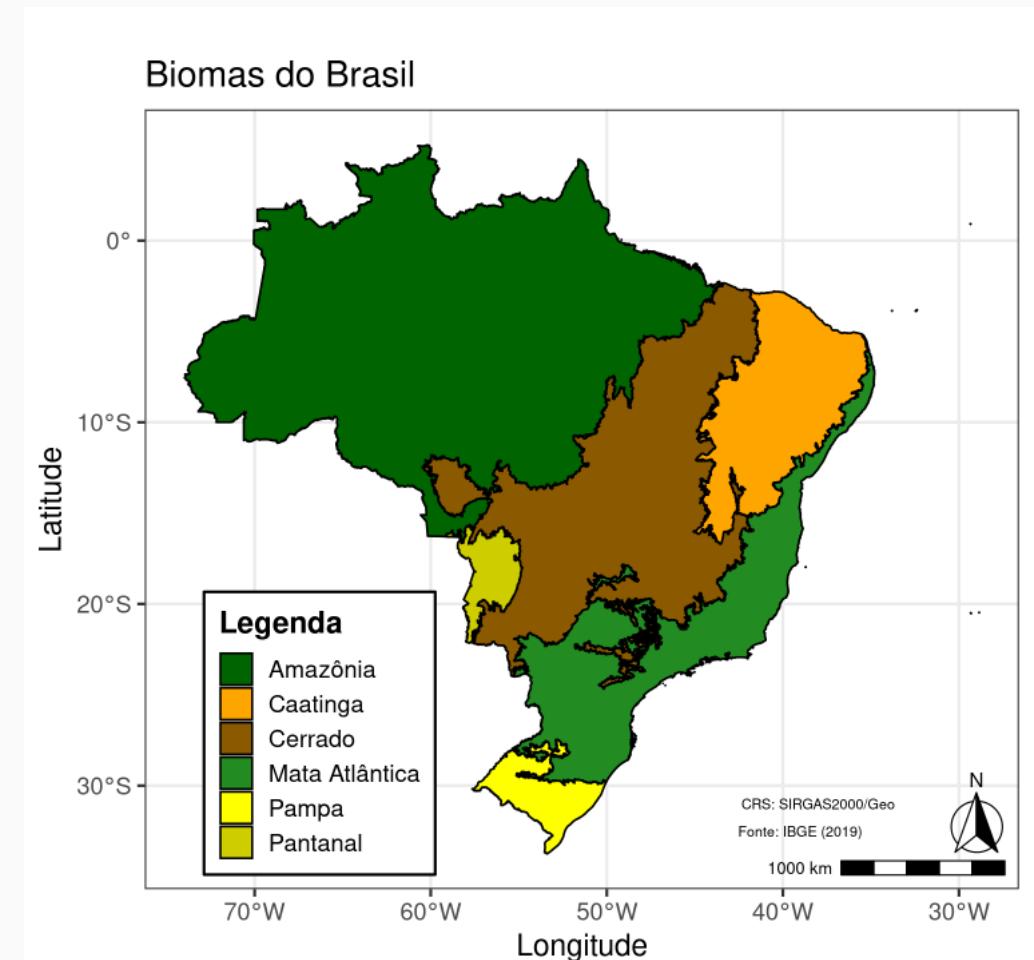
```
# barra de escala e norte
ggplot() +
  geom_sf(data = biomas, aes(fill = name_biome), color = "black") +
  scale_fill_manual(values = c("darkgreen", "orange",
                               "forestgreen", "yellow",
                               "brown", "darkblue"))
  theme_bw(base_size = 15) +
  theme(legend.title = element_text(size = 15, face = "bold"),
        legend.position = c(.2, .2),
        legend.background = element_rect(colour = "black", fill = "white"))
  annotation_scale(location = "br") +
  annotation_north_arrow(location = "br", which_north = "true",
                         pad_x = unit(0, "cm"), pad_y = unit(0, "cm"),
                         style = north_arrow_fancy_online)
```



3. Mapas estáticos

ggplot2 e ggspatial

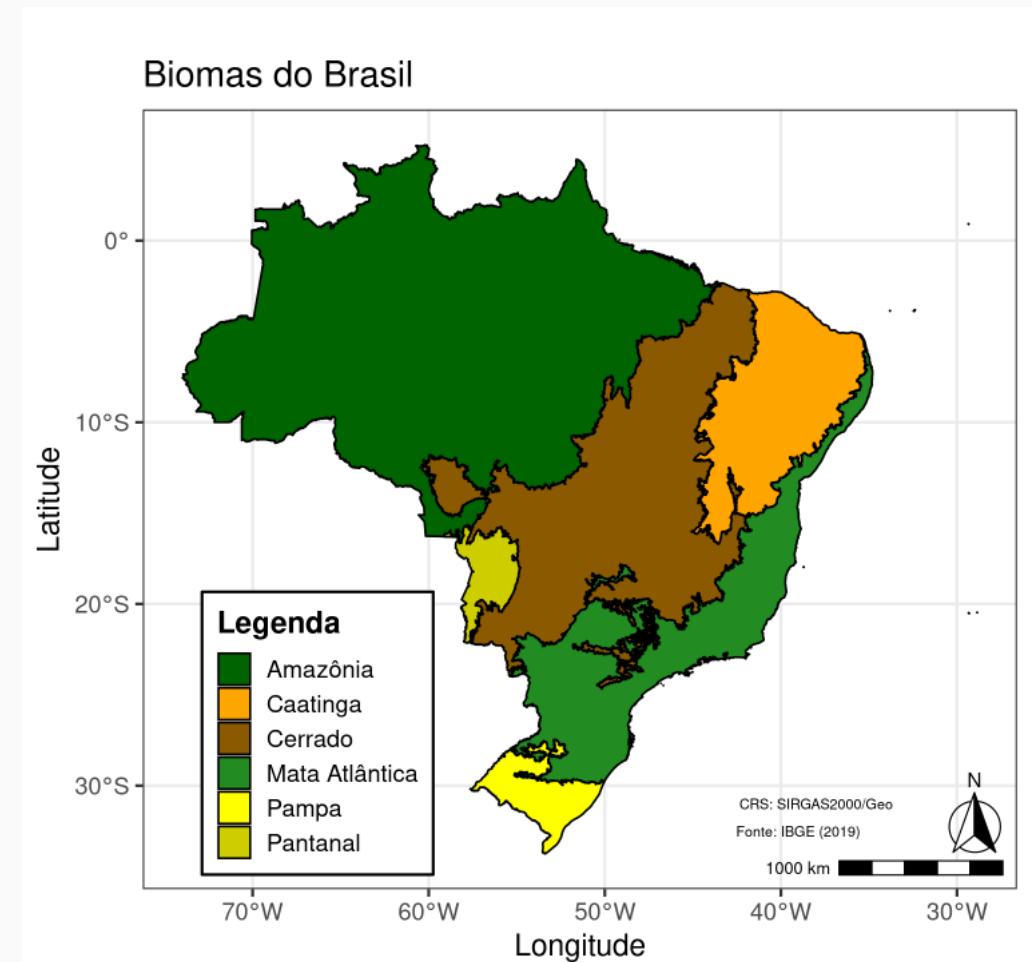
```
# nomes e anotações
ggplot(data = biomas) +
  aes(fill = name_biome) +
  geom_sf(color = "black") +
  scale_fill_manual(values = c("darkgreen", "orange",
                               "forestgreen", "yellow",
                               "brown"))
  theme_bw(base_size = 15) +
  theme(legend.title = element_text(size = 15, face = "bold"),
        legend.position = c(.2, .2),
        legend.background = element_rect(colour = "black", fill = "white"))
  annotation_scale(location = "br") +
  annotation_north_arrow(location = "br", which_north = "true",
                         pad_x = unit(0, "cm"), pad_y = unit(0, "cm"),
                         style = north_arrow_fancy_onion())
  annotate(geom = "text", label = "CRS: SIRGAS2000/Geo")
  annotate(geom = "text", label = "Fonte: IBGE (2019)")
  labs(title = "Biomas do Brasil", fill = "Legenda",
```



3. Mapas estáticos

ggplot2 e ggspatial

```
# atribuicao
map_biomas_ggplot2 <- ggplot(data = biomas) +
  aes(fill = name_biome) +
  geom_sf(color = "black") +
  scale_fill_manual(values = c("darkgreen", "orange",
                               "forestgreen", "yellow",
                               "brown"))
  theme_bw(base_size = 15) +
  theme(legend.title = element_text(size = 15, face = "bold"),
        legend.position = c(.2, .2),
        legend.background = element_rect(colour = "black", fill = "white"))
  annotation_scale(location = "br") +
  annotation_north_arrow(location = "br", which_north = "true",
                         pad_x = unit(0, "cm"), pad_y = 0,
                         style = north_arrow_fancy_on)
  annotate(geom = "text", label = "CRS: SIRGAS2000/Geo")
  annotate(geom = "text", label = "Fonte: IBGE (2019)")
  labs(title = "Biomas do Brasil", fill = "Legenda",
       map_biomas_ggplot2
```



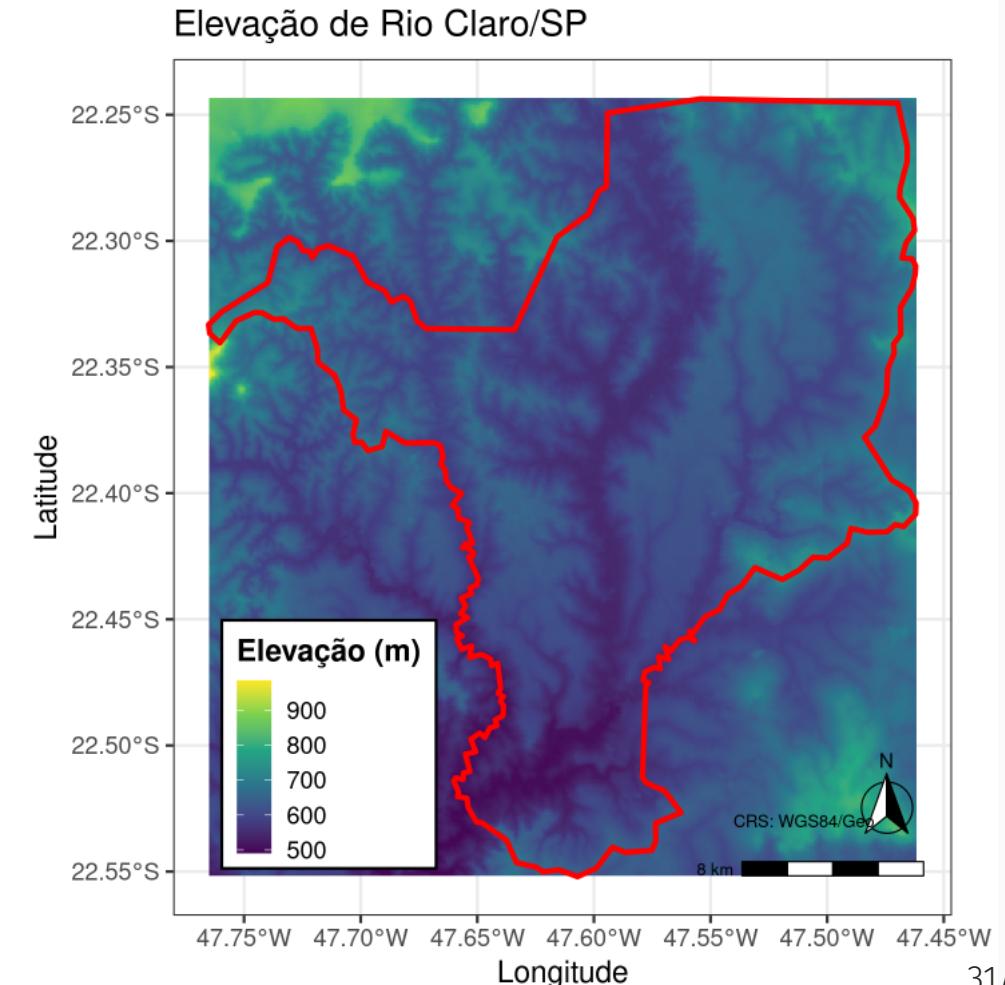
3. Mapas estáticos

ggplot2 e ggspatial

3. Mapas estáticos

ggplot2 e ggspatial

```
# plot
map_dem_rc_ggplot2 ← ggplot() +
  geom_raster(data = dem_rc_da, aes(x = x, y = y, fill = Elevation)) +
  geom_sf(data = rc_2020, col = "red", fill = NA, size = 1) +
  scale_fill_viridis_c() +
  coord_sf() +
  theme_bw(base_size = 15) +
  theme(legend.title = element_text(size = 15, face = "bold"),
        legend.position = c(.2, .2),
        legend.background = element_rect(colour = "black", fill = "white")) +
  annotation_scale(location = "br",
                    pad_x = unit(.5, "cm"), pad_y = unit(.5, "cm")) +
  annotation_north_arrow(location = "br", which_north = "true",
                         pad_x = unit(.4, "cm"), pad_y = unit(.4, "cm"),
                         style = north_arrow_fancy_open()) +
  annotate(geom = "text", label = "CRS: WGS84/Geo", x = -180, y = 900) +
  labs(title = "Elevação de Rio Claro/SP", fill = "Elevação (m)") +
  map_dem_rc_ggplot2
```



3. Mapas estáticos

ggplot2 e ggspatial

```
# exportar
ggsave(
  filename = here::here("03_dados", "mapas", "mapa_biomas_ggplot2.png"),
  plot = map_biomas_ggplot2,
  width = 20,
  height = 20,
  units = "cm",
  dpi = 300
)
```

3. Mapas estáticos

ggplot2 e ggspatial

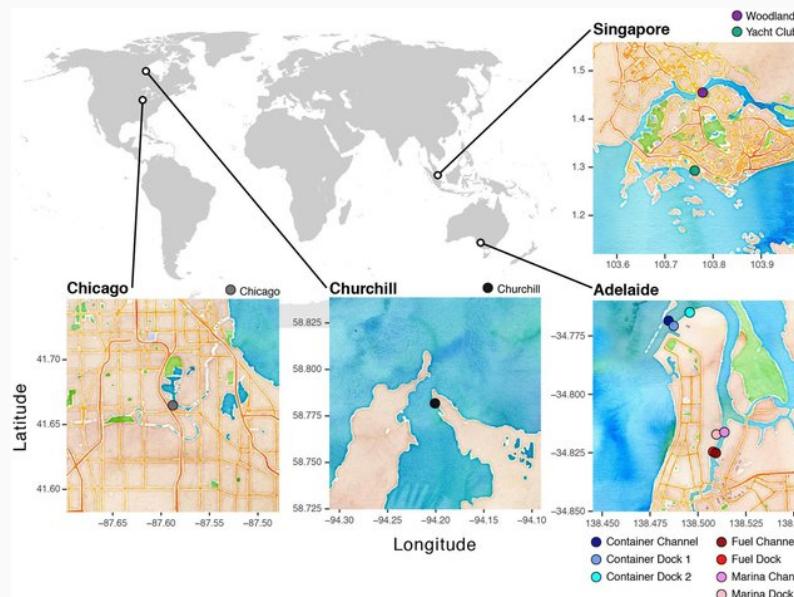
```
# exportar
ggsave(
  filename = here::here("03_dados", "mapas", "mapa_dem_rc_rc_ggplot2.png"),
  plot = map_dem_rc_ggplot2,
  width = 20,
  height = 20,
  units = "cm",
  dpi = 300
)
```

3. Mapas estáticos

ggmap

```
# pacote
# install.packages("ggmap")
library(ggmap)

# limite
brasil <- c(left = -80, bottom = -35, right = -30, top = 5)
rio_claro <- c(left = -47.67, bottom = -22.48, right = -47.46, top = -22.32)
```



3. Mapas estáticos

ggmap

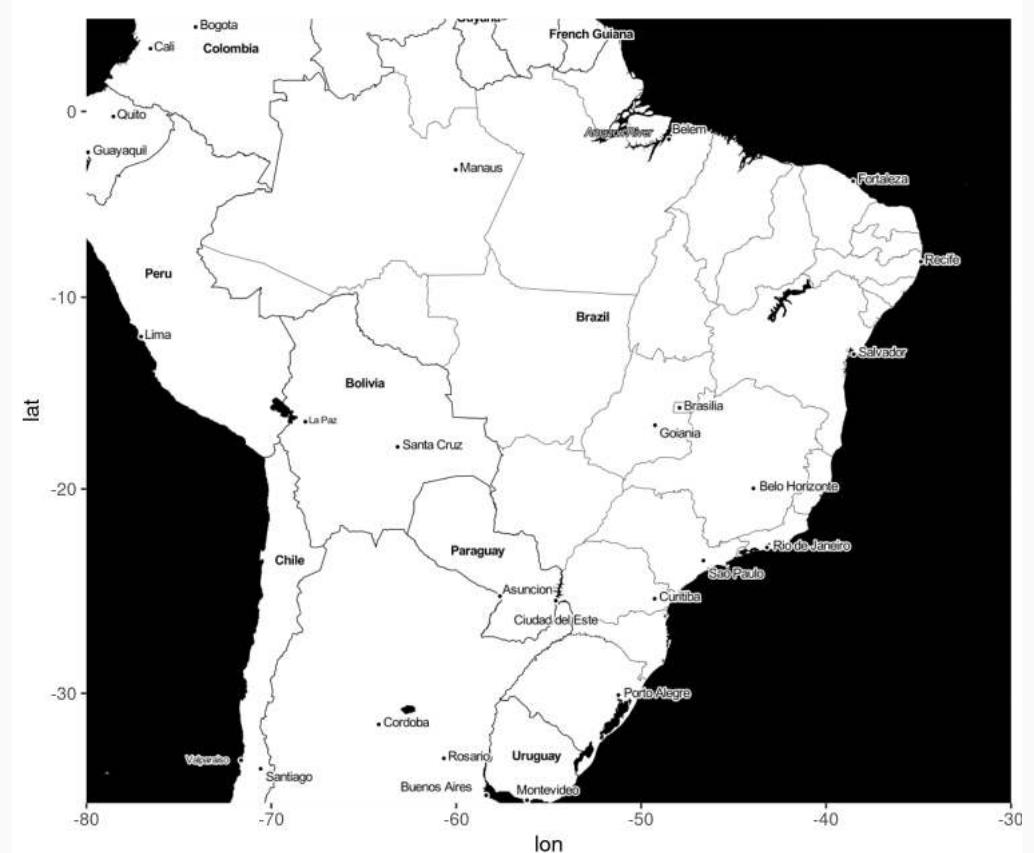
```
# plot  
get_stamenmap(bbox = brasil,  
              maptype = "terrain",  
              zoom = 5) %>%  
  
ggmap()
```



3. Mapas estáticos

ggmap

```
# plot  
get_stamenmap(bbox = brasil,  
              maptype = "toner",  
              zoom = 5) %>%  
  ggmap()
```



3. Mapas estáticos

ggmap

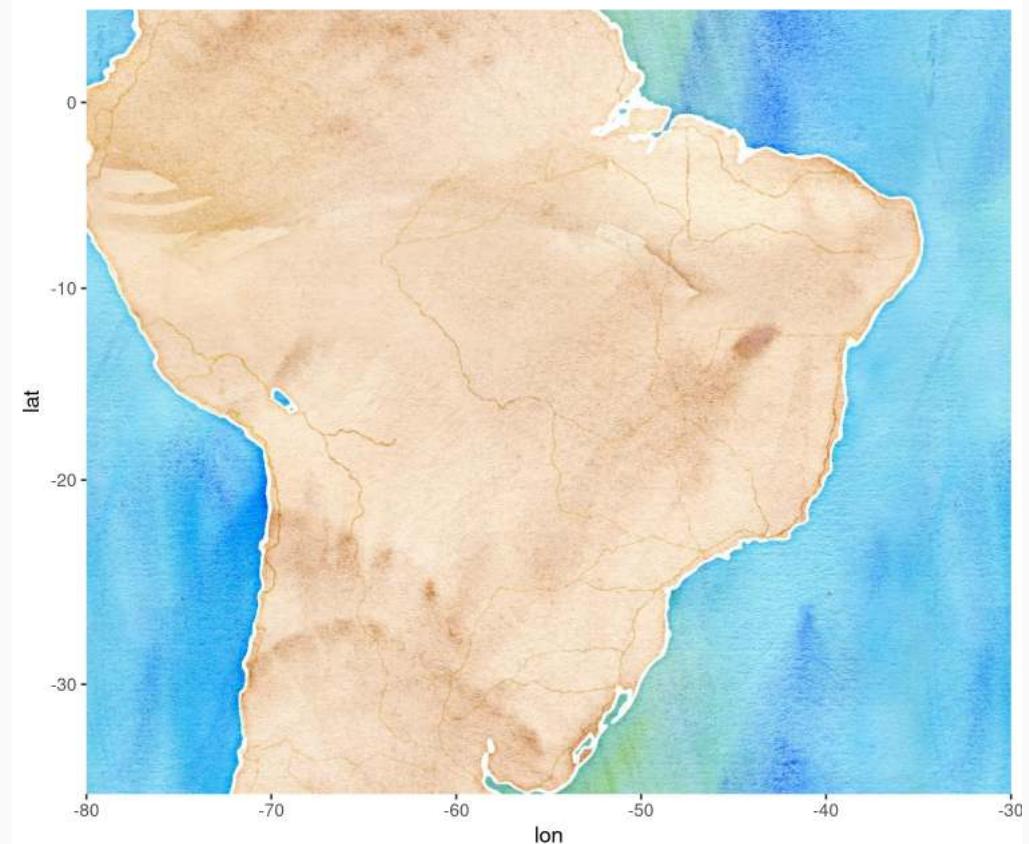
```
# plot  
get_stamenmap(bbox = brasil,  
              maptype = "toner-lite",  
              zoom = 5) %>%  
  
ggmap()
```



3. Mapas estáticos

ggmap

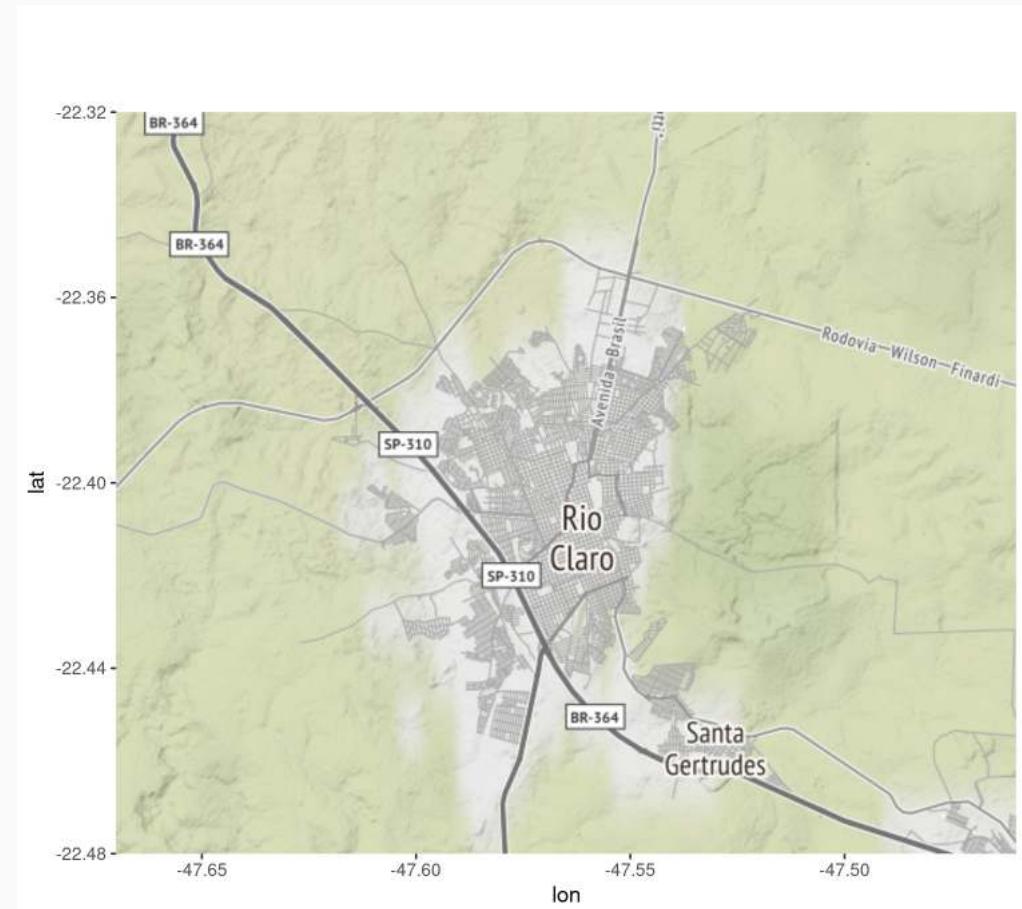
```
# plot  
get_stamenmap(bbox = brasil,  
              maptype = "watercolor",  
              zoom = 5) %>%  
  ggmap()
```

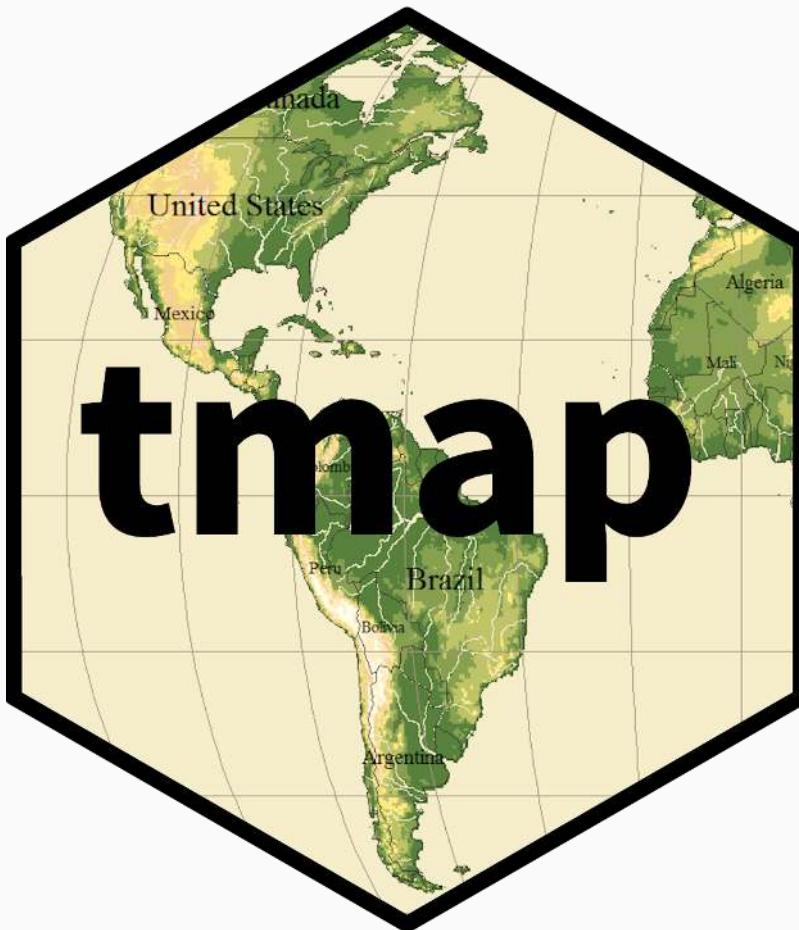


3. Mapas estáticos

ggmap

```
# plot  
get_stamenmap(bbox = rio_claro,  
               maptype = "terrain",  
               zoom = 12) %>%  
  ggmap()
```





[tmap](#)

3. Mapas estáticos

tmap

Artigo



Journal of Statistical Software
April 2018, Volume 84, Issue 6. doi: 10.18637/jss.v084.i06

tmap: Thematic Maps in R

Martijn Tennekes
Statistics Netherlands

Abstract

Thematic maps show spatial distributions. The theme refers to the phenomena that is shown, which is often demographical, social, cultural, or economic. The best known thematic map type is the choropleth, in which regions are colored according to the distribution of a data variable. The R package **tmap** offers a coherent plotting system for thematic maps that is based on the *layered grammar of graphics*. Thematic maps are created by stacking layers, where per layer, data can be mapped to one or more aesthetics. It is also possible to generate small multiples. Thematic maps can be further embellished by configuring the map layout and by adding map attributes, such as a scale bar and a compass. Besides plotting thematic maps on the graphics device, they can also be made interactive as an HTML widget. In addition, the R package **tmaptools** contains several convenient functions for reading and processing spatial data.

Keywords: thematic maps, spatial data, R.

[Tennekes \(2018\)](#)

3. Mapas estáticos

tmap

Sintaxe baseada no **ggplot2** e na **Gramática dos Gráficos**

Dois modos: **plot** (estáticos) e **visualização** (interativos)

Estrutura:

```
tm_shape( ... ) +  
tm_*( ... ) +  
tm_facets( ... ) +  
tm_layout( ... )
```

```
tm_style( ... )  
tmapArrange( ... )  
tm_mode( ... )
```

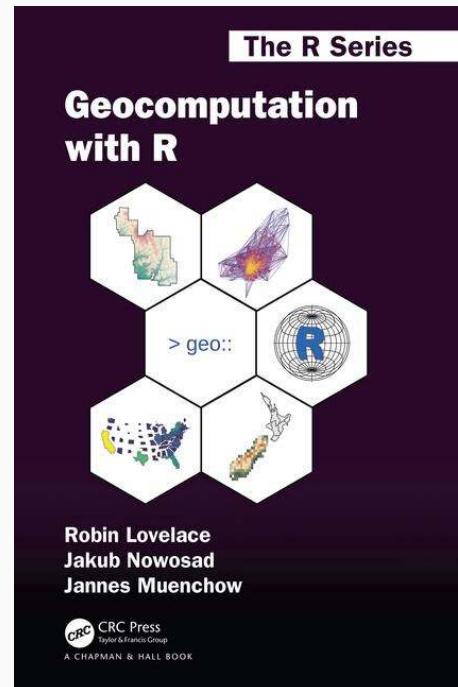
[tmap](#)



3. Mapas estáticos

tmap

Geocomputation with R (2019)



[Lovelace et al. \(2019\)](#)

3. Mapas estáticos

tmap

Elegant and informative maps with tmap

Elegant and informative maps with tmap

Martijn Tennekes, Jakub Nowosad

2021-09-05

Welcome

This is the online home of *Elegant and informative maps with tmap*, a work-in-progress book on geospatial data visualization with the R-package `tmap`.

How to contribute?

We encourage contributions on any part of the book, including:

- improvements to the text, e.g., clarifying unclear sentences, fixing typos
- changes to the code
- suggestions on content (see [the project's issue tracker](#))

Additional information

 This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](#).

This version of the book was built on GH Actions on 2021-09-05.

[Tennekes & Nowosad \(inacabado\)](#)

3. Mapas estáticos

tmap

	(Spatial) data	Layers (geometry, mapping, and scaling)	Small multiples	Layout
ggplot2	ggplot(...) +	geom_...(...) + scale_...(...) +	facet_wrap(...) +	theme(...)
tmap	tm_shape(...) +	tm_...(...) +	tm_facets(...) +	tm_layout(...)

Implemented: tm_polygons, tm_symbols, tm_lines, tm_raster, tm_text, tm_fill, tm_borders, tm_bubbles, tm_squares, tm_dots, tm_rgb, tm_markers, tm_iso

ggplot2

Layered Grammar of Graphics

- Defaults
 - Data
 - Aesthetics

- Layers
 - Data
 - Aesthetics
 - Geometry
 - Statistics
 - Position

Scales

Coordinates

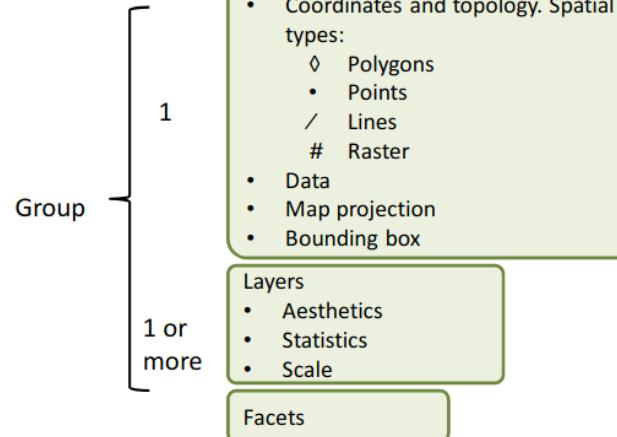
Facets

tmap

Layered Grammar of Thematic Maps

Shape

- Coordinates and topology. Spatial types:
 - ◊ Polygons
 - Points
 - / Lines
 - # Raster
- Data
- Map projection
- Bounding box



3. Mapas estáticos

tmap

```
# polígonos  
tm_shape(biomas) +  
  tm_polygons()
```



3. Mapas estáticos

tmap

```
# bordas  
tm_shape(biomas) +  
  tm_borders()
```



3. Mapas estáticos

tmap

```
# preenchimento  
tm_shape(biomas) +  
  tm_fill()
```



3. Mapas estáticos

tmap

```
# cores  
tm_shape(biomas) +  
  tm_fill(col = "name_biome", title = "Legenda")
```



3. Mapas estáticos

tmap

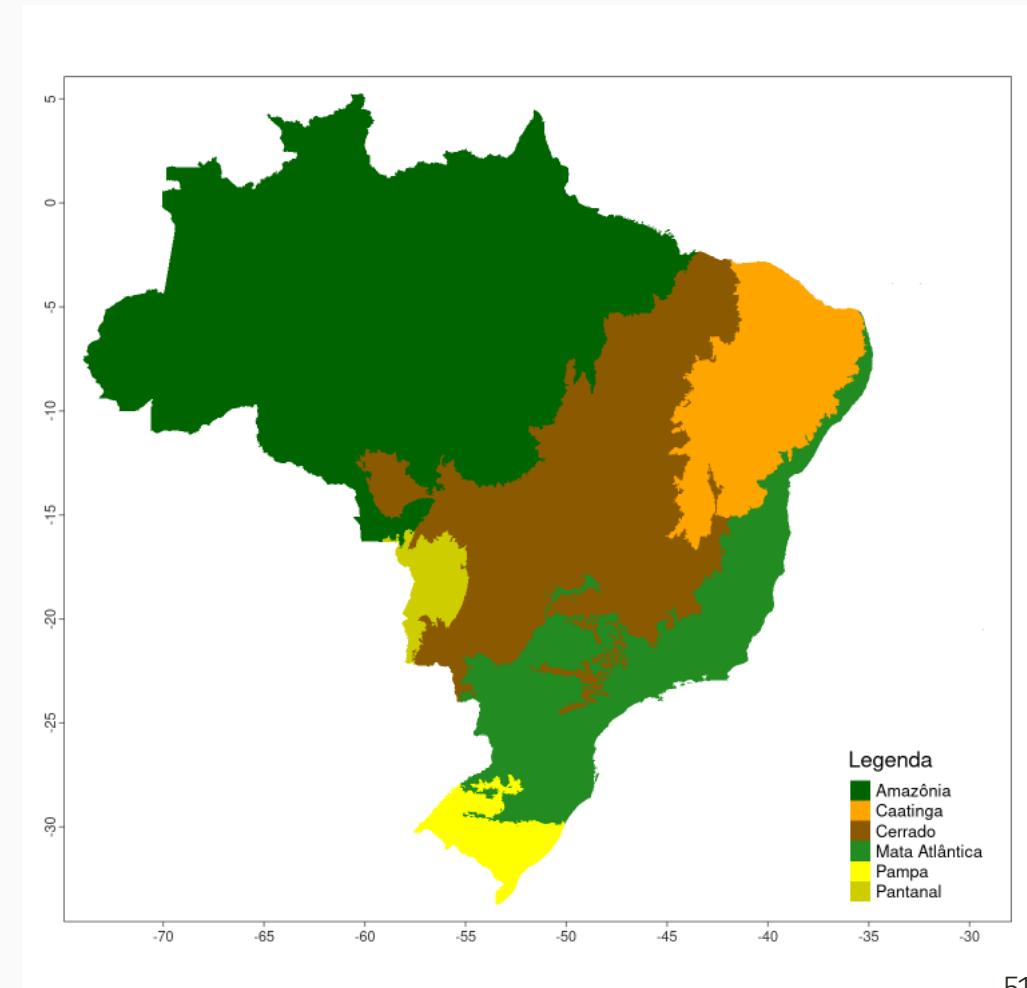
```
# definir cores  
tm_shape(biomas) +  
  tm_fill(col = "name_biome",  
          pal = c("darkgreen", "orange", "orange4",  
          title = "Legenda")
```



3. Mapas estáticos

tmap

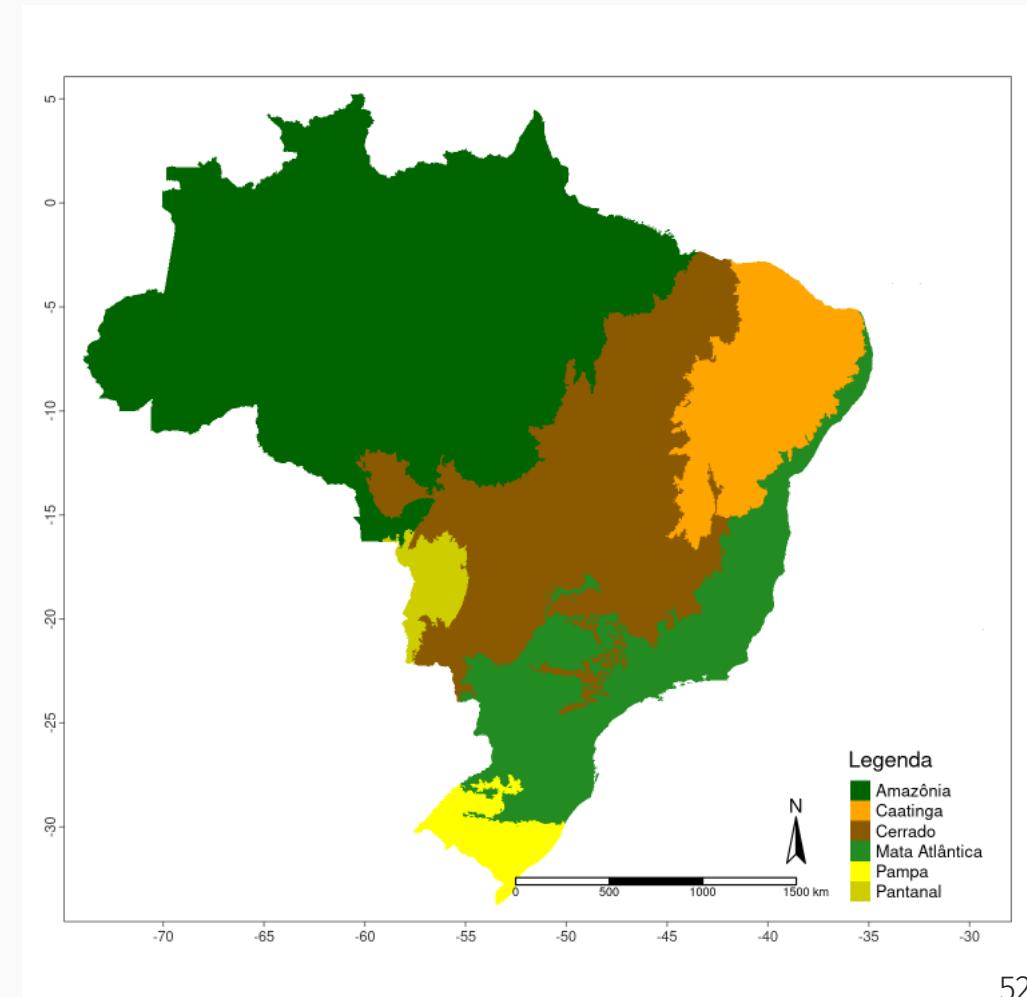
```
# grid coordinates
tm_shape(biomas) +
  tm_fill(col = "name_biome",
         pal = c("darkgreen", "orange", "orange4",
                title = "Legenda") +
  tm_grid(lines = FALSE,
          labels.format = list(big.mark = " "),
          labels.rot = c(0, 90))
```



3. Mapas estáticos

tmap

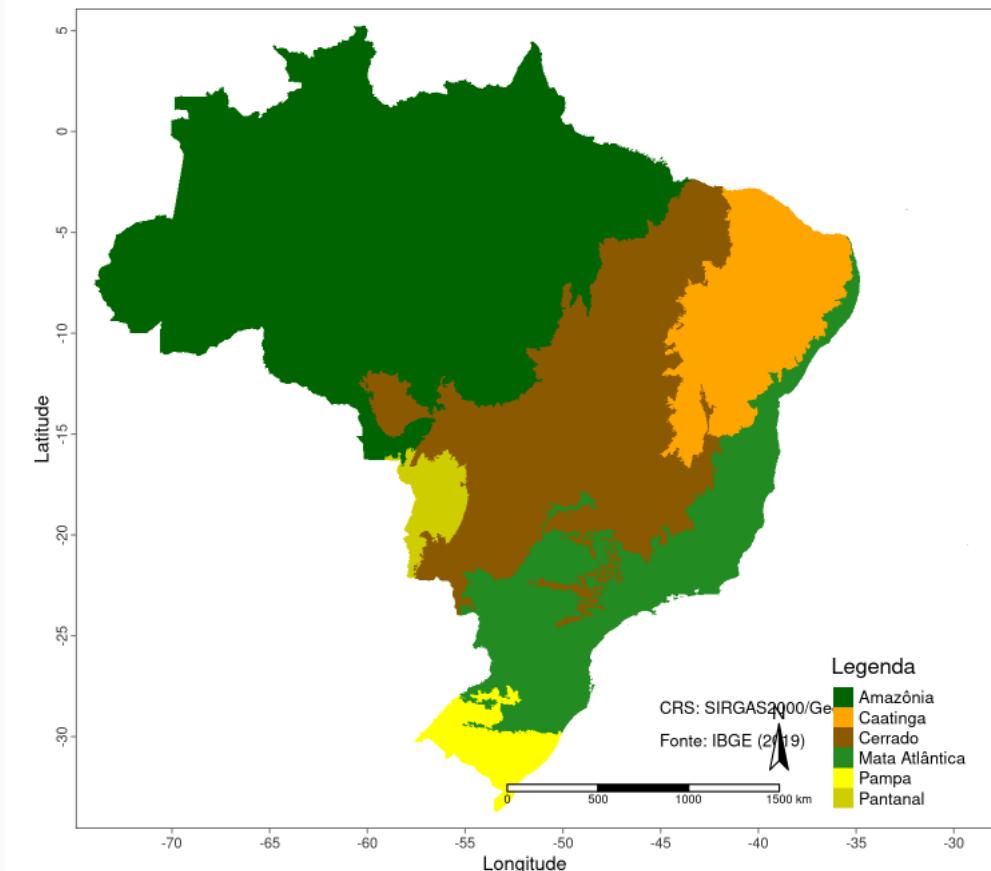
```
# barra de escala e norte
tm_shape(biomas) +
  tm_fill(col = "name_biome",
         pal = c("darkgreen", "orange", "orange4",
                title = "Legenda") +
  tm_grid(lines = FALSE,
          labels.format = list(big.mark = " "),
          labels.rot = c(0, 90)) +
  tm_compass() +
  tm_scale_bar()
```



3. Mapas estáticos

tmap

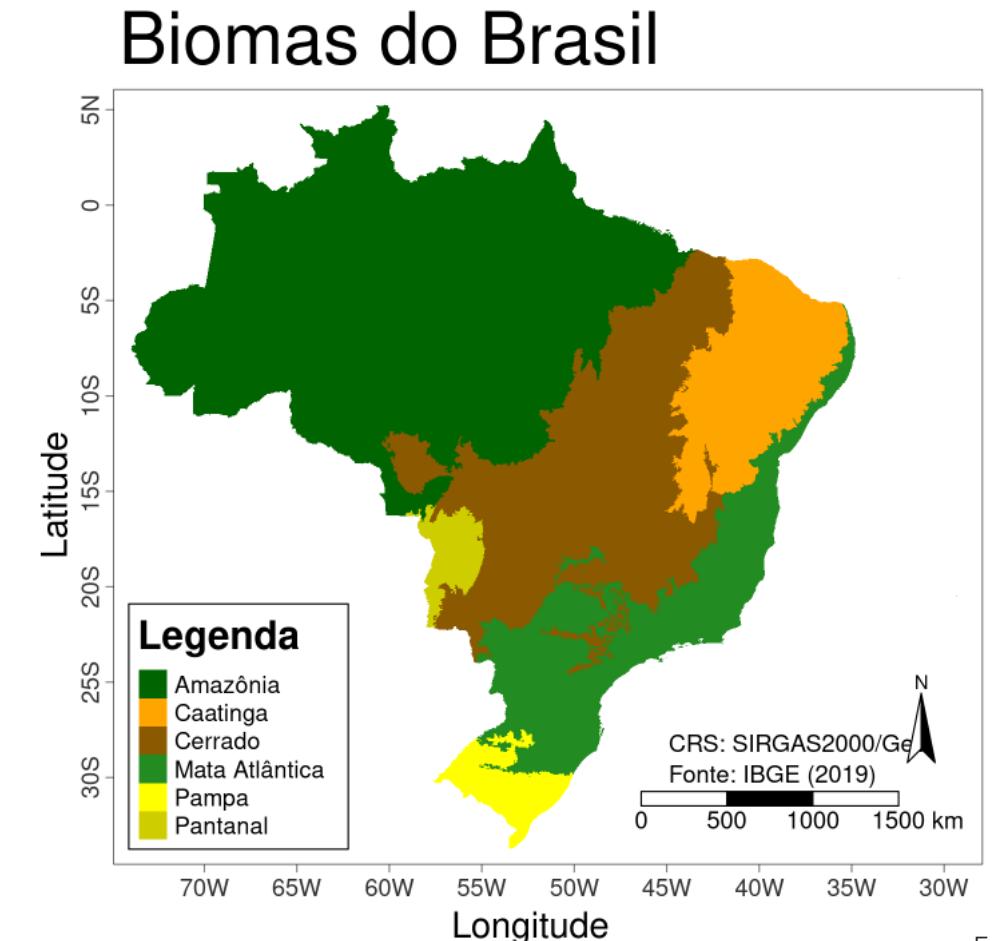
```
# nomes
tm_shape(biomas) +
  tm_fill(col = "name_biome",
         pal = c("darkgreen", "orange", "orange4",
                title = "Legenda") +
  tm_grid(lines = FALSE,
          labels.format = list(big.mark = " "),
          labels.rot = c(0, 90)) +
  tm_compass() +
  tm_scale_bar() +
  tm_xlab("Longitude") +
  tm_ylab("Latitude") +
  tm_credits("CRS: SIRGAS2000/Geo", position = c(.63,
  tm_credits("Fonte: IBGE (2019)", position = c(.63,
```



3. Mapas estáticos

tmap

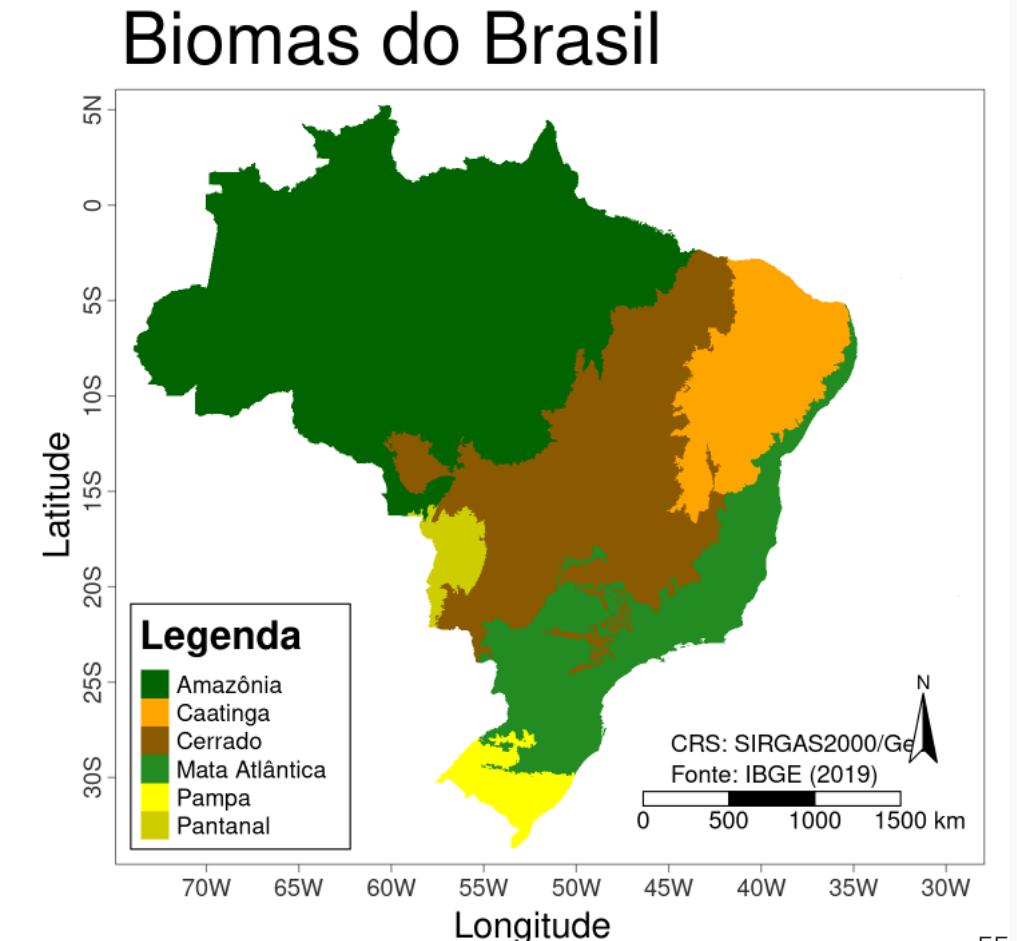
```
# titulos
tm_shape(biomas) +
  tm_fill(col = "name_biome",
          pal = c("darkgreen", "orange", "orange4",
          title = "Legenda") +
  tm_graticules(lines = FALSE,
                 labels.format = list(big.mark = ""),
                 labels.rot = c(0, 90),
                 labels.size = 1) +
  tm_compass(size = 3) +
  tm_scale_bar(size = 1) +
  tm_xlab("Longitude", size = 1.5) +
  tm_ylab("Latitude", size = 1.5) +
  tm_credits("CRS: SIRGAS2000/Geo", position = c(.63,
  tm_credits("Fonte: IBGE (2019)", position = c(.63,
  tm_layout(main.title = "Biomas do Brasil",
            main.title.position = c(.1, .95),
            main.title.size = 3,
            title.fontface = "bold",
            legend.frame = TRUE,
            legend.position = c("left", "bottom"),
```



3. Mapas estáticos

tmap

```
# atribuicao
map_biomas_tmap ← tm_shape(biomas) +
  tm_fill(col = "name_biome",
          pal = c("darkgreen", "orange", "orange4",
                 title = "Legenda") +
  tm_graticules(lines = FALSE,
                 labels.format = list(big.mark = ""),
                 labels.rot = c(0, 90),
                 labels.size = 1) +
  tm_compass(size = 3) +
  tm_scale_bar(size = 1) +
  tm_xlab("Longitude", size = 1.5) +
  tm_ylab("Latitude", size = 1.5) +
  tm_credits("CRS: SIRGAS2000/Geo", position = c(.63,
  tm_credits("Fonte: IBGE (2019)", position = c(.63,
  tm_layout(main.title = "Biomas do Brasil",
            main.title.position = c(.1, .95),
            main.title.size = 3,
            title.fontface = "bold",
            legend.frame = TRUE,
            legend.position = c("left", "bottom"),
```

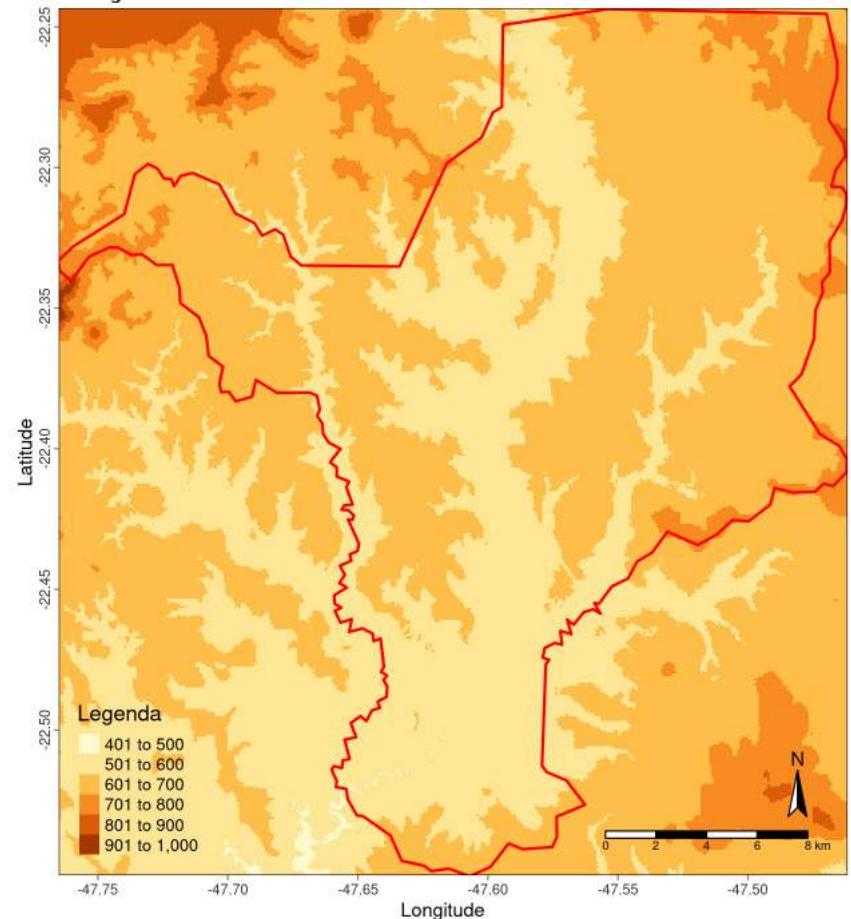


3. Mapas estáticos

tmap

```
# plot
map_dem_rc_tmap ← tm_shape(dem_rc) +
  tm_raster(title = "Legenda") +
  tm_shape(rc_2020) +
  tm_borders(col = "red", lwd = 2) +
  tm_grid(lines = FALSE,
    labels.format = list(big.mark = " "),
    labels.rot = c(0, 90)) +
  tm_compass() +
  tm_scale_bar() +
  tm_xlab("Longitude") +
  tm_ylab("Latitude") +
  tm_layout(legend.position = c("left", "bottom"),
    main.title = "Elevação Rio Claro/SP")
map_dem_rc_tmap
```

Elevação Rio Claro/SP

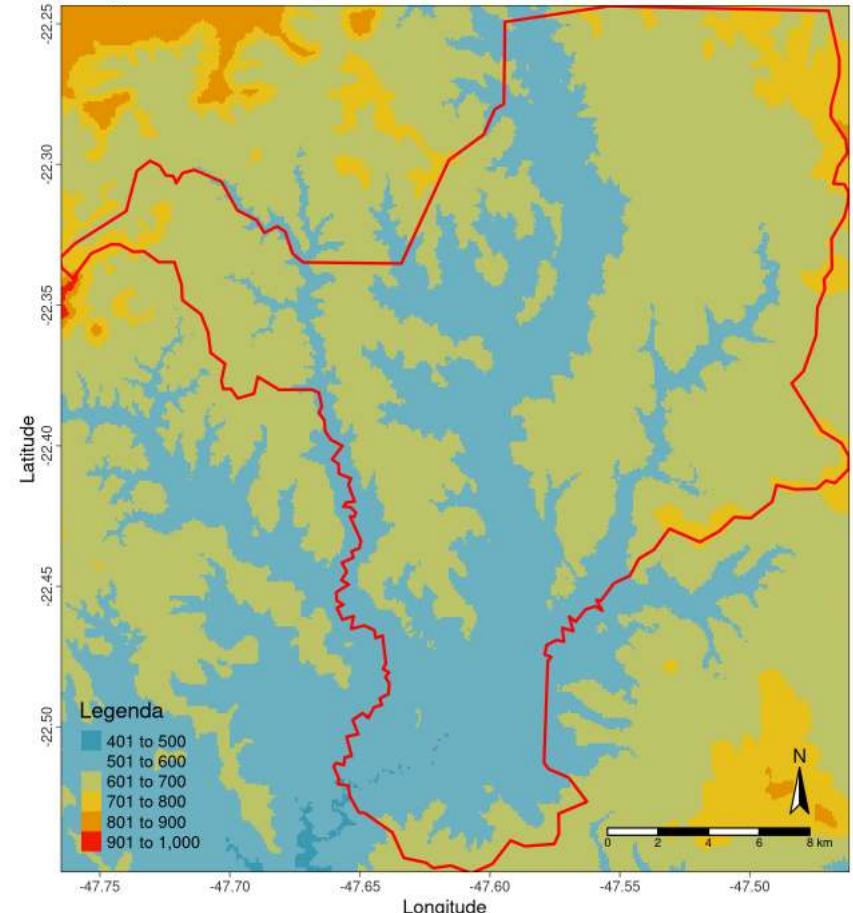


3. Mapas estáticos

tmap

```
# plot
map_dem_rc_tmap ← tm_shape(dem_rc) +
  tm_raster(pal = wesanderson::wes_palette("Zissou1"))
  tm_shape(rc_2020) +
  tm_borders(col = "red", lwd = 2) +
  tm_grid(lines = FALSE,
    labels.format = list(big.mark = " "),
    labels.rot = c(0, 90)) +
  tm_compass() +
  tm_scale_bar() +
  tm_xlab("Longitude") + tm_ylab("Latitude") +
  tm_layout(legend.position = c("left", "bottom"),
    main.title = "Elevação Rio Claro/SP")
map_dem_rc_tmap
```

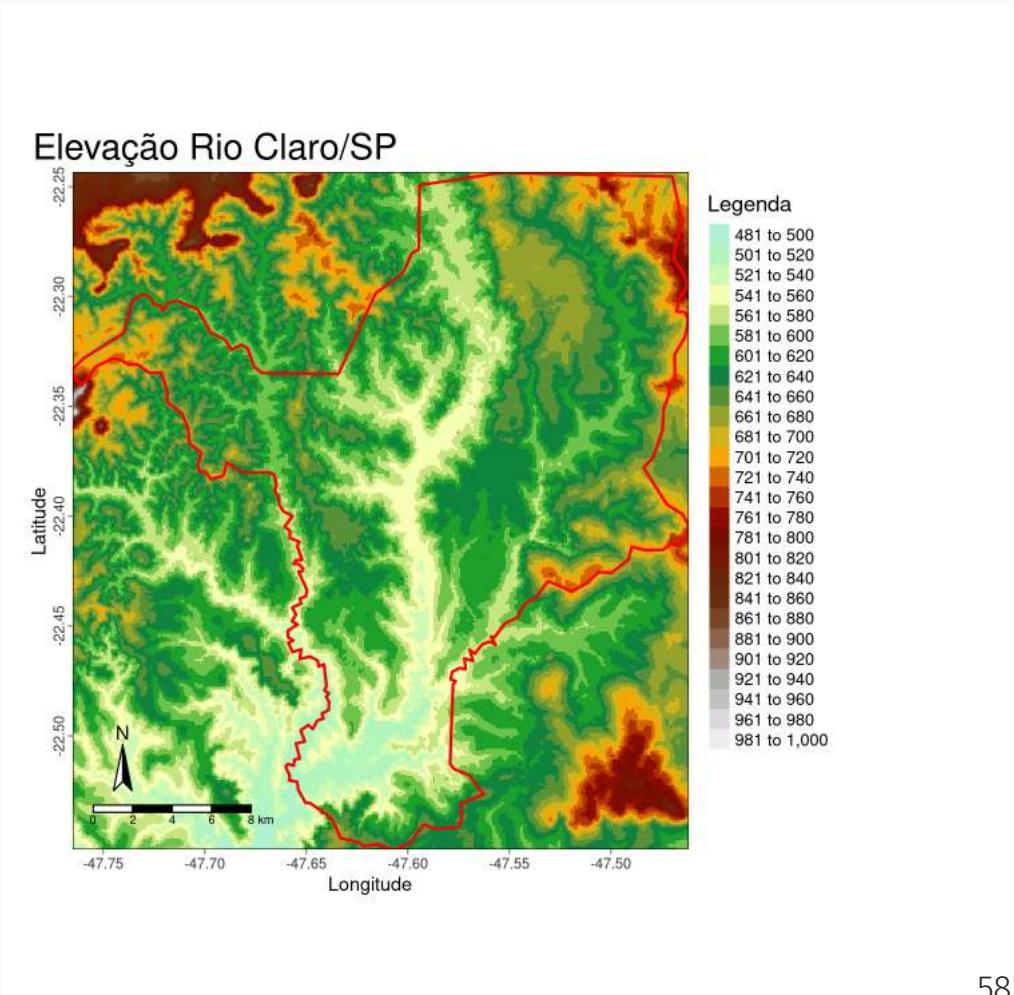
Elevação Rio Claro/SP



3. Mapas estáticos

tmap

```
# plot
map_dem_rc_tmap ← tm_shape(dem_rc) +
  tm_raster(pal = cptcity::cpt(pal = "gmt_GMT_dem4"))
  tm_shape(rc_2020) +
  tm_borders(col = "red", lwd = 2) +
  tm_grid(lines = FALSE,
    labels.format = list(big.mark = " "),
    labels.rot = c(0, 90)) +
  tm_compass(position = c("left", "bottom")) +
  tm_scale_bar(position = c("left", "bottom")) +
  tm_xlab("Longitude") +
  tm_ylab("Latitude") +
  tm_layout(legend.outside = TRUE,
    main.title = "Elevação Rio Claro/SP")
map_dem_rc_tmap
```



3. Mapas estáticos

tmap

```
# exportar
tmap::tmap_save(tm = map_biomas_tmap,
                 filename = here::here("03_dados", "mapas", "mapa_biomas_tmap.png"),
                 width = 20,
                 height = 20,
                 units = "cm",
                 dpi = 300)
```

3. Mapas estáticos

tmap

```
# exportar
tmap::tmap_save(tm = map_dem_rc_tmap,
                 filename = here::here("03_dados", "mapas", "mapa_dem_rc_tmap.png"),
                 width = 20,
                 height = 20,
                 units = "cm",
                 dpi = 300)
```



[mapsf](#)

3. Mapas estáticos

mapsf

```
# pacote  
# install.packages("mapsf")  
library(mapsf)  
  
# sintaxe  
mf_map(x, var, type)
```

map type	type	data	polygon	point	line
Base Map	"base"	geometry			
Proportional Symbols	"prop"	stock			
Typology	"typo"	category			
Choropleth	"choro"	ratio			
Graduated Symbols	"grad"	stock			

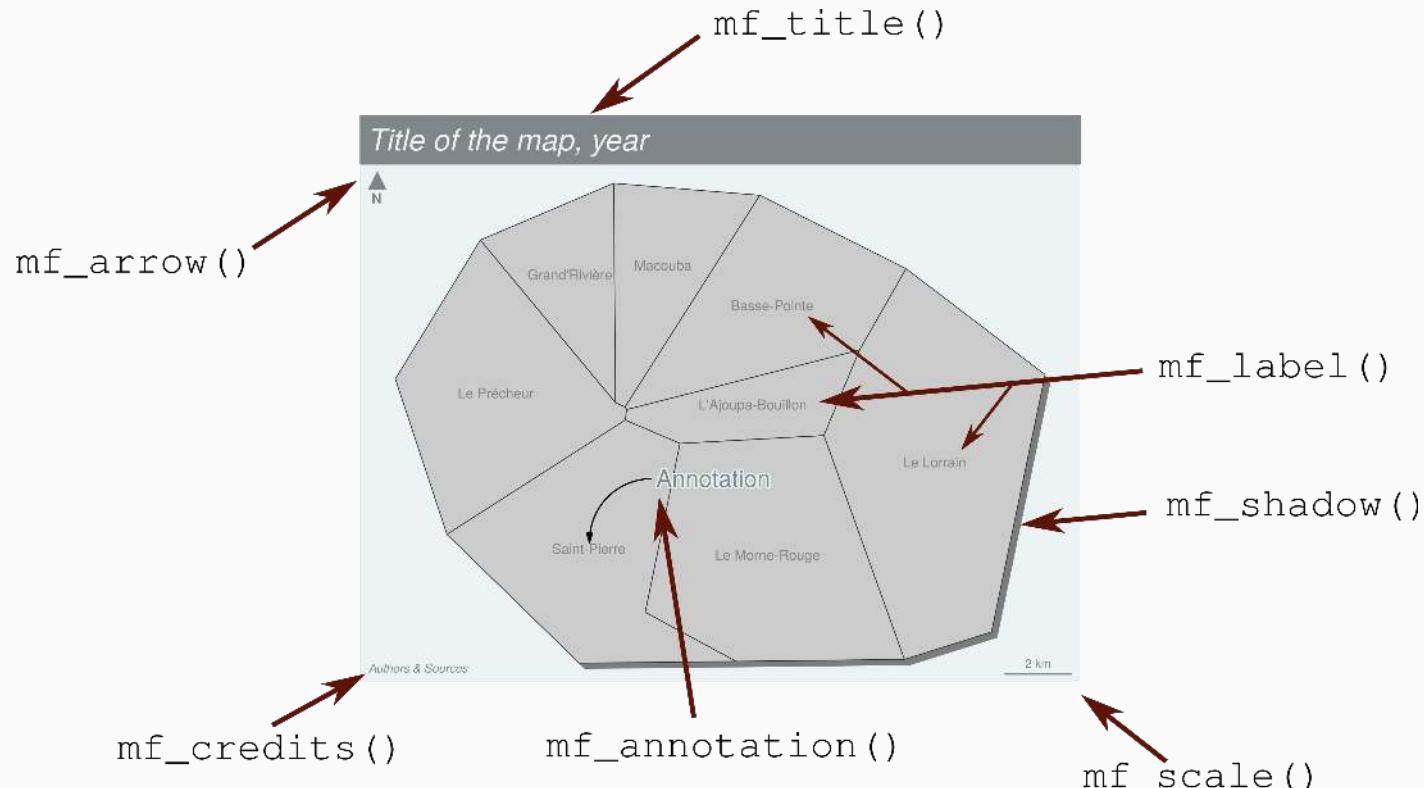
mapsf

Symbols	"symb"	category		
Proportional Symbols + Typo	"prop_typo"	stock & category		
Proportional Symbols + Choro	"prop_choro"	stock & ratio		
Symbols + Choro	"symb_choro"	category & ratio		

3. Mapas estáticos

mapsf

Layout



mapsf

3. Mapas estáticos

mapsf

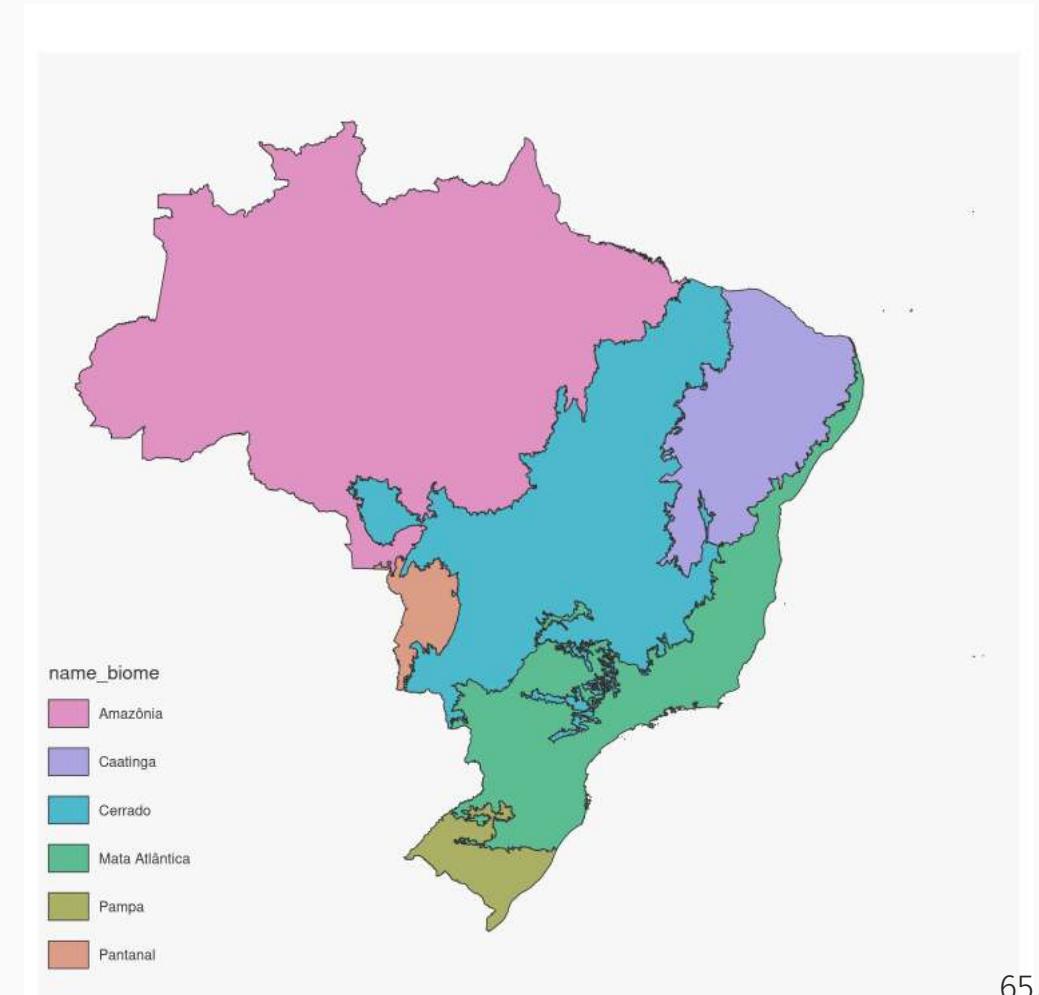
```
# plot  
mf_map(x = biomas)
```



3. Mapas estáticos

mapsf

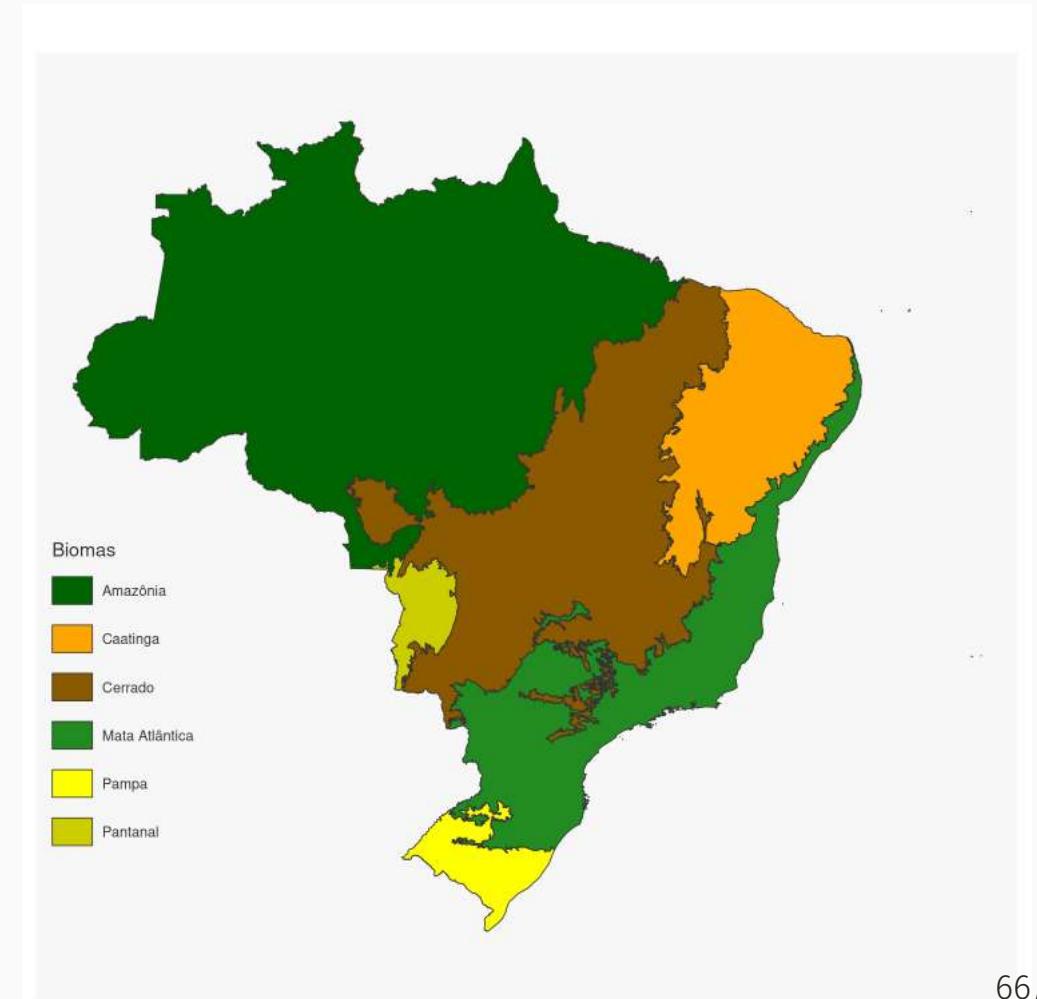
```
# plot  
mf_map(x = biomas, var = "name_biome", type = "typo")
```



3. Mapas estáticos

mapsf

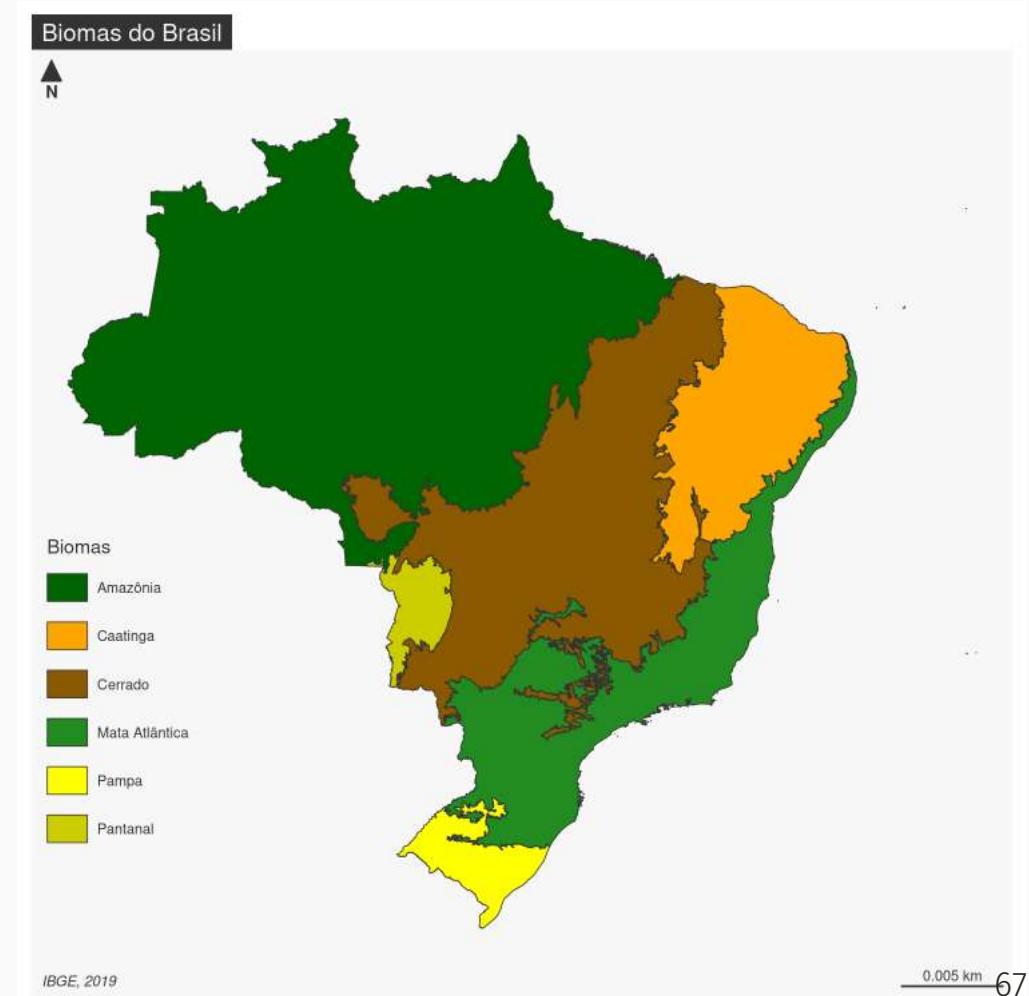
```
# plot
mf_map(x = biomas, var = "name_biome", type = "typo"
       pal = c("darkgreen", "orange", "orange4",
              "forestgreen", "yellow", "yellow3"),
       leg_title = "Biomas",
       leg_pos = c(-75, -15))
```



3. Mapas estáticos

mapsf

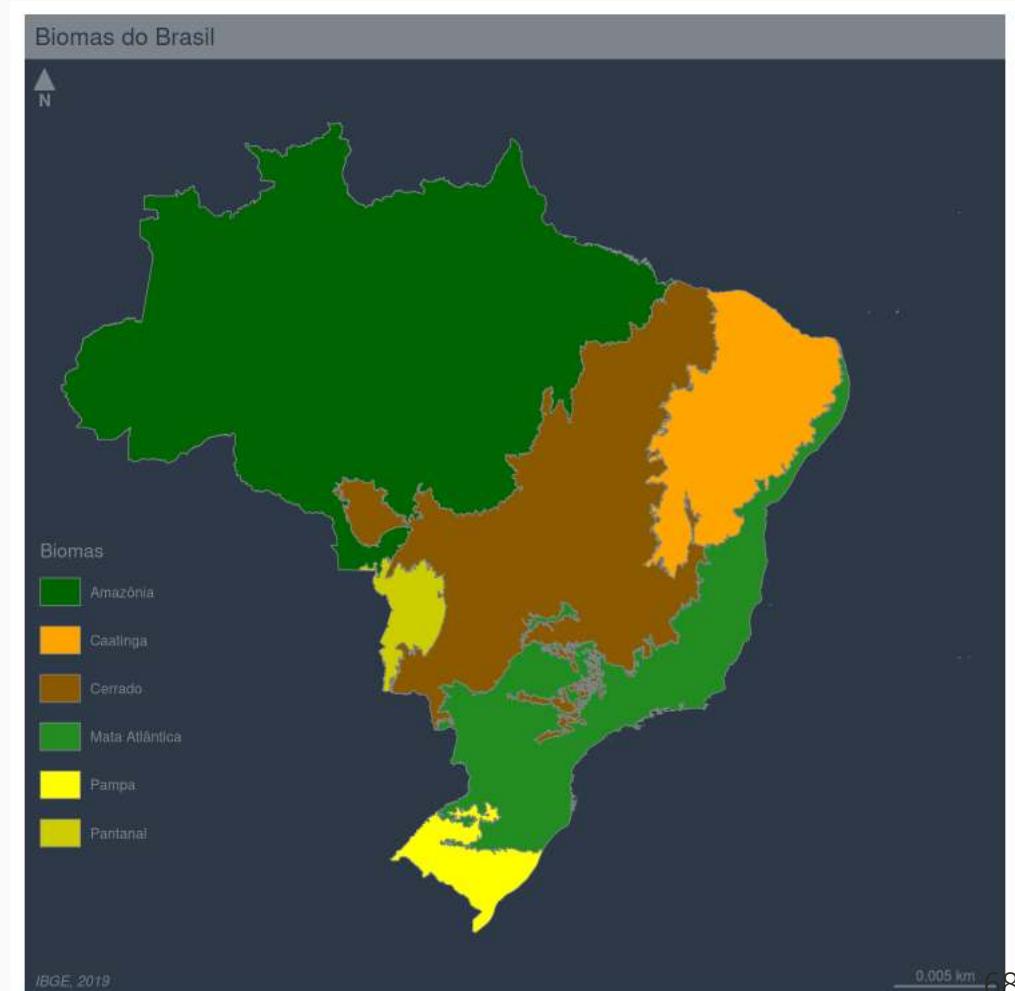
```
# plot
mf_map(x = biomas, var = "name_biome", type = "typo"
       pal = c("darkgreen", "orange", "orange4",
              "forestgreen", "yellow", "yellow3"),
       leg_title = "Biomas",
       leg_pos = c(-75, -15))
mf_title("Biomas do Brasil")
mf_credits("IBGE, 2019")
mf_scale()
mf_arrow()
```



3. Mapas estáticos

mapsf

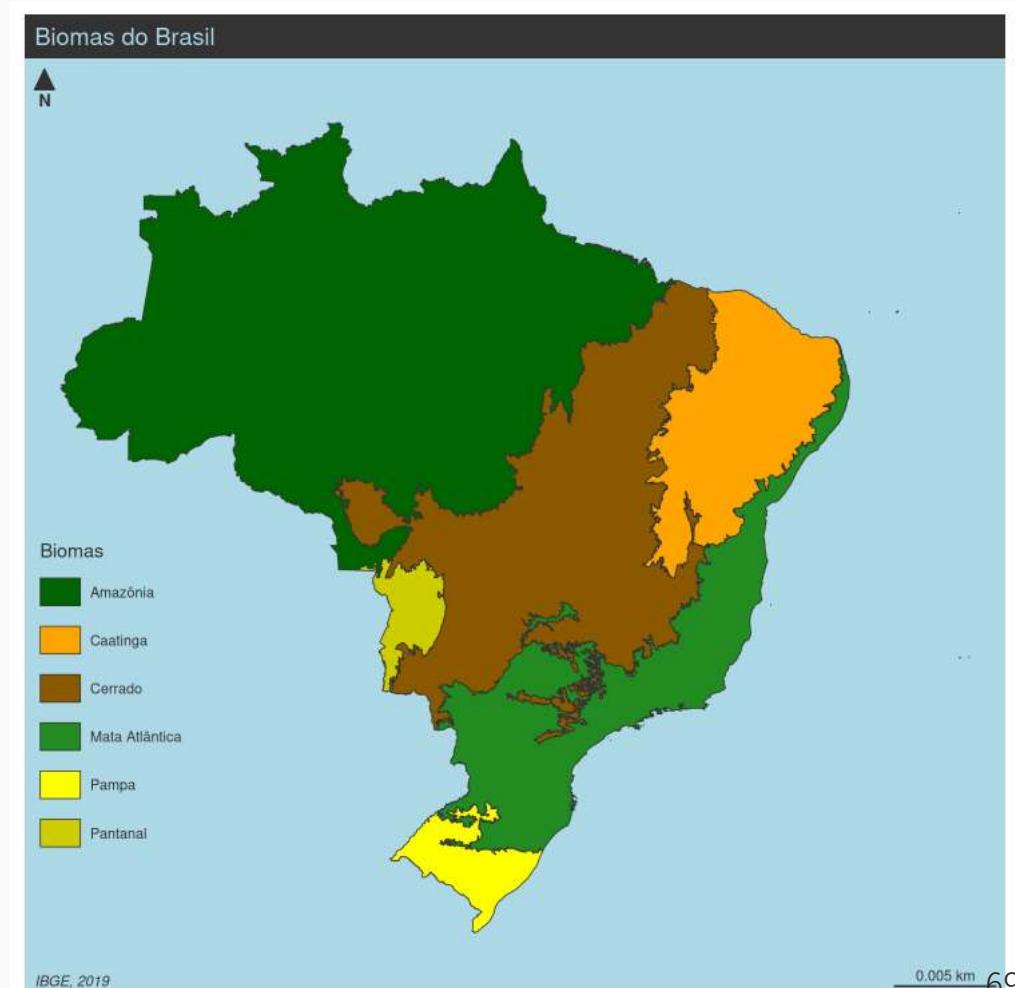
```
# plot
mf_init(x = biomas, theme = "dark")
mf_map(x = biomas, var = "name_biome", type = "typo"
       pal = c("darkgreen", "orange", "orange4",
              "forestgreen", "yellow", "yellow3"),
       leg_title = "Biomias",
       leg_pos = c(-75, -15),
       add = TRUE)
mf_title("Biomas do Brasil")
mf_credits("IBGE, 2019")
mf_scale()
mf_arrow()
```



3. Mapas estáticos

mapsf

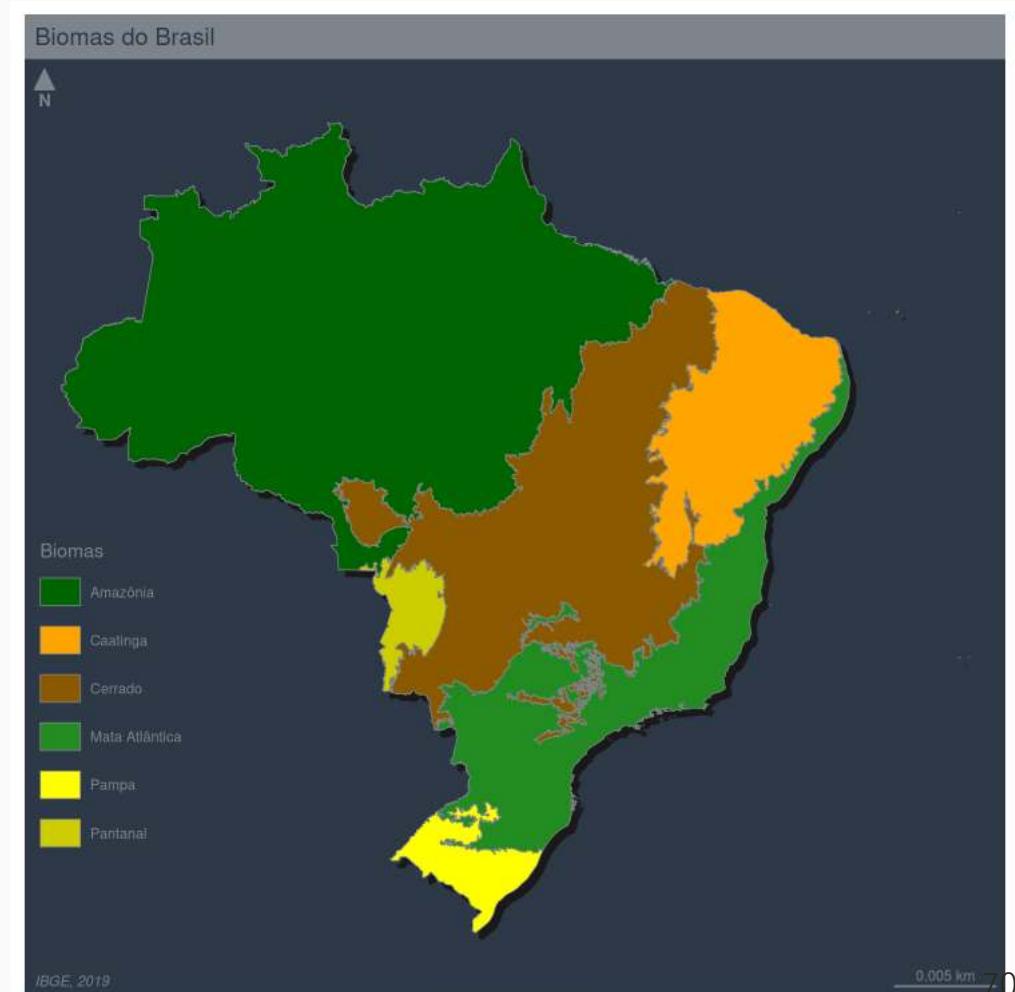
```
# plot
mf_theme(bg = "lightblue", fg = "gray20")
mf_map(x = biomas, var = "name_biome", type = "typo"
       pal = c("darkgreen", "orange", "orange4",
              "forestgreen", "yellow", "yellow3"),
       leg_title = "Biomas",
       leg_pos = c(-75, -15))
mf_title("Biomas do Brasil")
mf_credits("IBGE, 2019")
mf_scale()
mf_arrow()
```



3. Mapas estáticos

mapsf

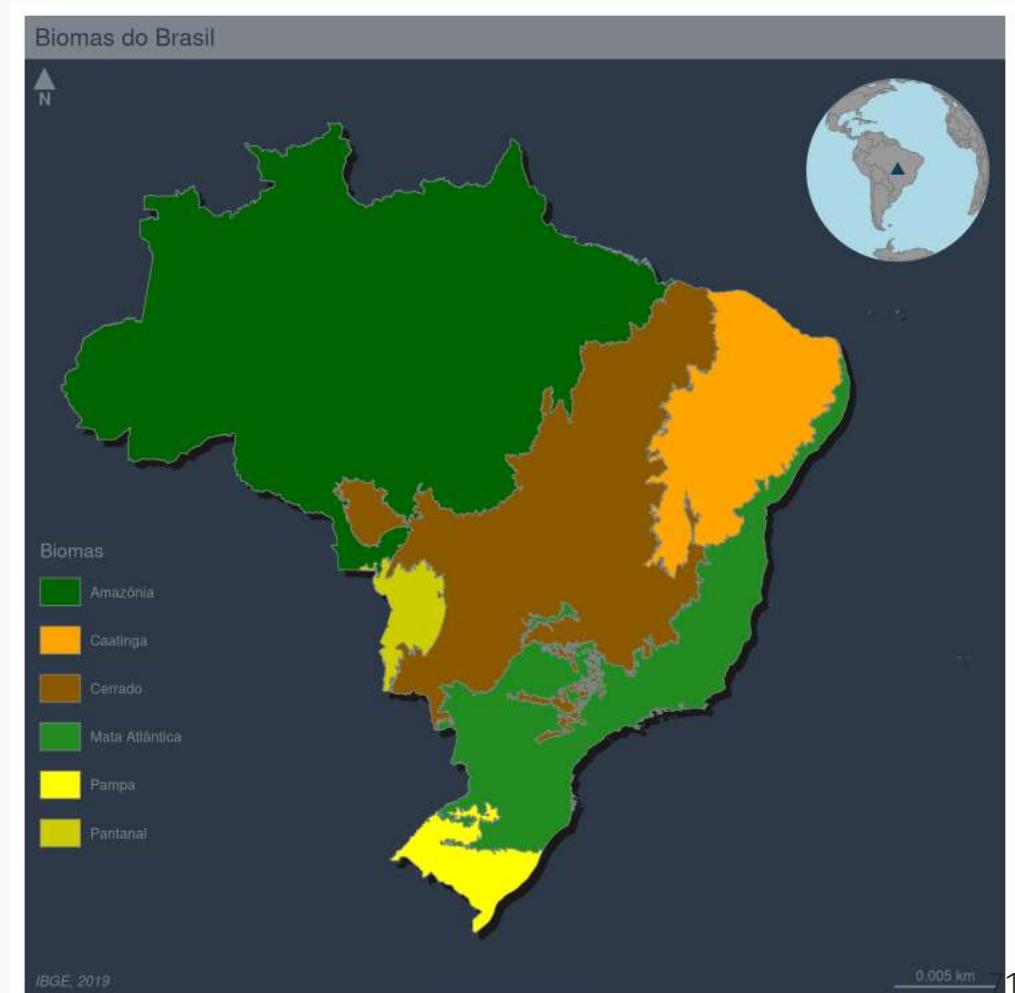
```
# plot
mf_init(x = biomas, theme = "dark")
mf_shadow(x = biomas, col = "gray10", add = TRUE)
mf_map(x = biomas, var = "name_biome", type = "typo"
       pal = c("darkgreen", "orange", "orange4",
              "forestgreen", "yellow", "yellow3"),
       leg_title = "Biomas",
       leg_pos = c(-75, -15),
       add = TRUE)
mf_title("Biomas do Brasil")
mf_credits("IBGE, 2019")
mf_scale()
mf_arrow()
```



3. Mapas estáticos

mapsf

```
# plot
mf_init(x = biomas, theme = "dark")
mf_shadow(x = biomas, col = "gray10", add = TRUE)
mf_map(x = biomas, var = "name_biome", type = "typo"
       pal = c("darkgreen", "orange", "orange4",
              "forestgreen", "yellow", "yellow3"),
       leg_title = "Biomas",
       leg_pos = c(-75, -15),
       add = TRUE)
mf_inset_on(x = "worldmap", pos = "topright")
mf_worldmap(biomas, col = "#0E3F5C")
mf_inset_off()
mf_title("Biomas do Brasil")
mf_credits("IBGE, 2019")
mf_scale()
mf_arrow()
```



3. Mapas estáticos

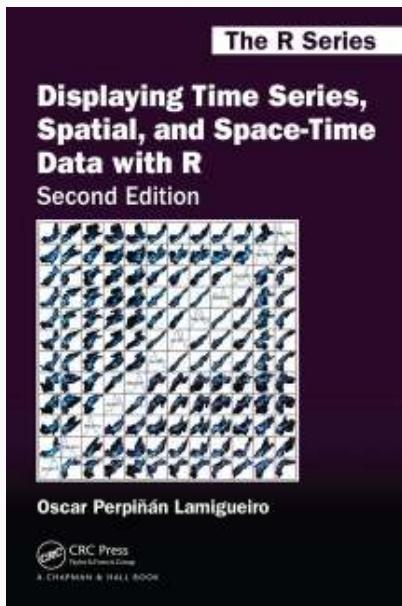
mapsf

```
# export
mf_export(x = biomas,
           filename = here::here("03_dados", "mapas", "mapa_biomass_mapsf.png"),
           wi = 20, he = 20, un = "cm", res = 300)
mf_init(x = biomas, theme = "dark")
mf_shadow(x = biomas, col = "gray10", add = TRUE)
mf_map(x = biomas, var = "name_biome", type = "typo",
       pal = c("darkgreen", "orange", "orange4",
              "forestgreen", "yellow", "yellow3"),
       leg_title = "Biomas",
       leg_pos = c(-70, -17),
       add = TRUE)
mf_inset_on(x = "worldmap", pos = "topright")
mf_worldmap(biomass, col = "#0E3F5C")
mf_inset_off()
mf_title("Biomass do Brasil")
mf_credits("IBGE, 2019")
mf_scale()
mf_arrow()
dev.off()
```

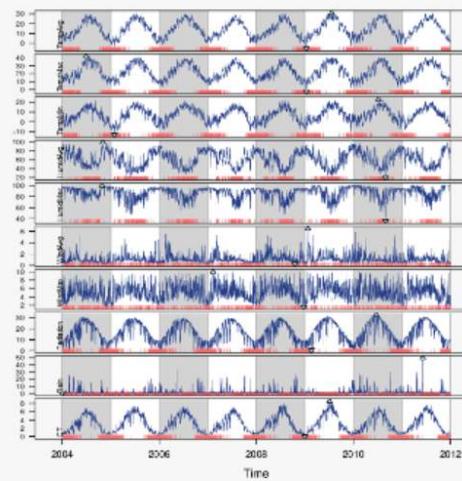
3. Mapas estáticos

rasterVis

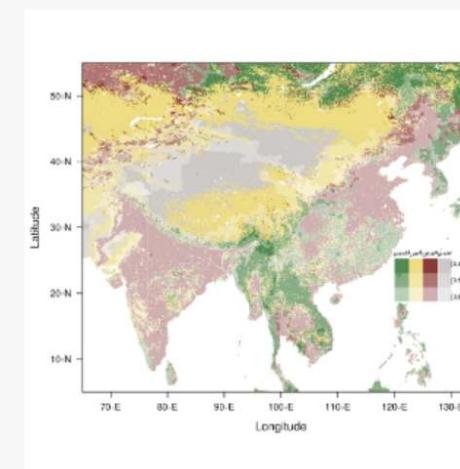
```
# pacote  
# install.packages('rasterVis')  
library(rasterVis)
```



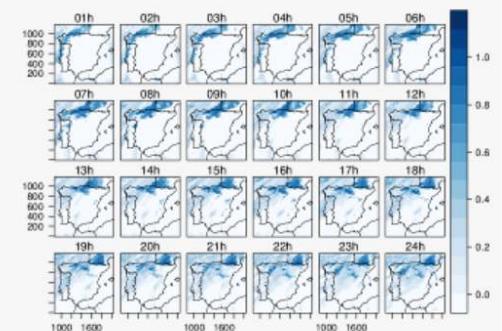
Time Series



Spatial Data



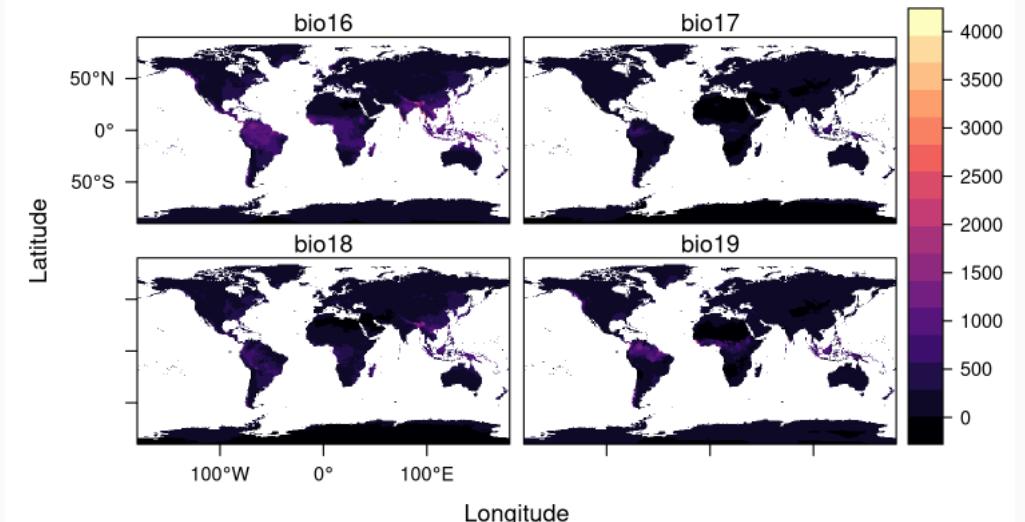
Space-time Data



3. Mapas estáticos

rasterVis

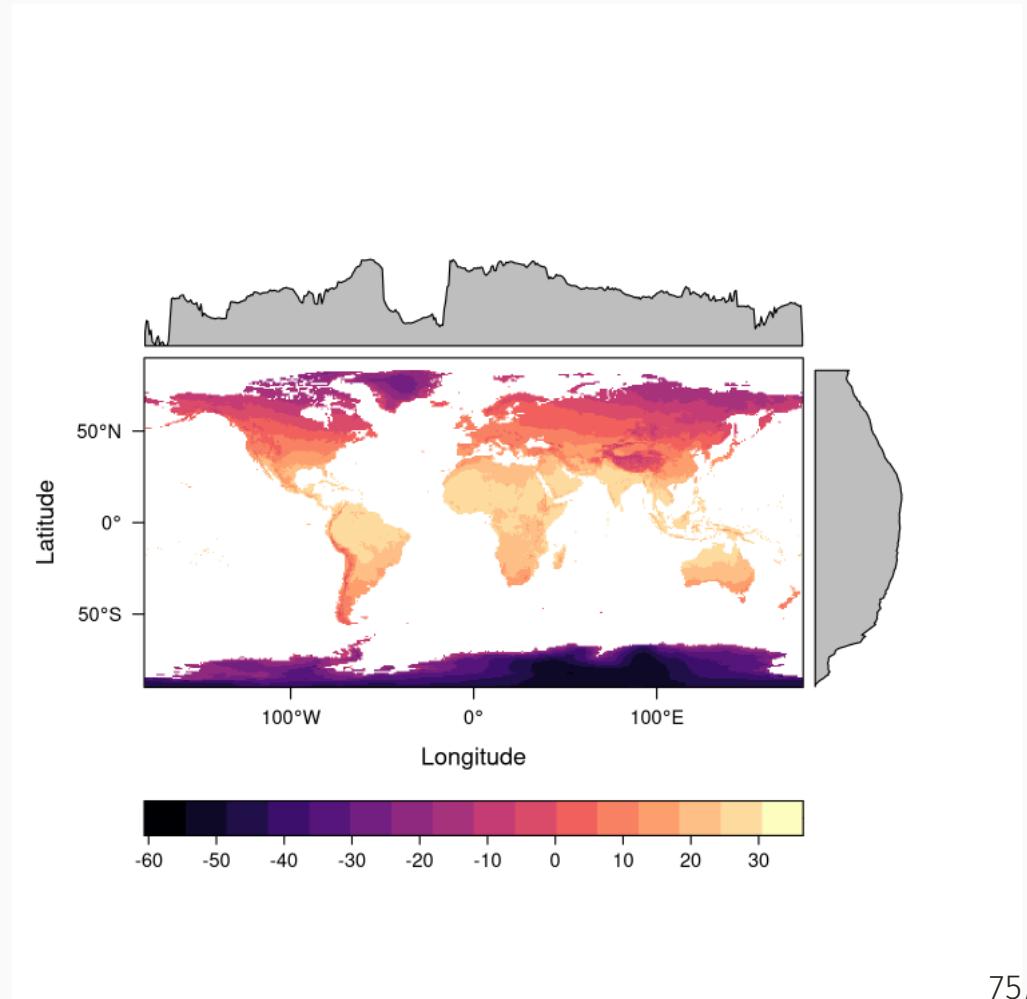
```
# plot  
levelplot(bioclim[[8:11]])
```



3. Mapas estáticos

rasterVis

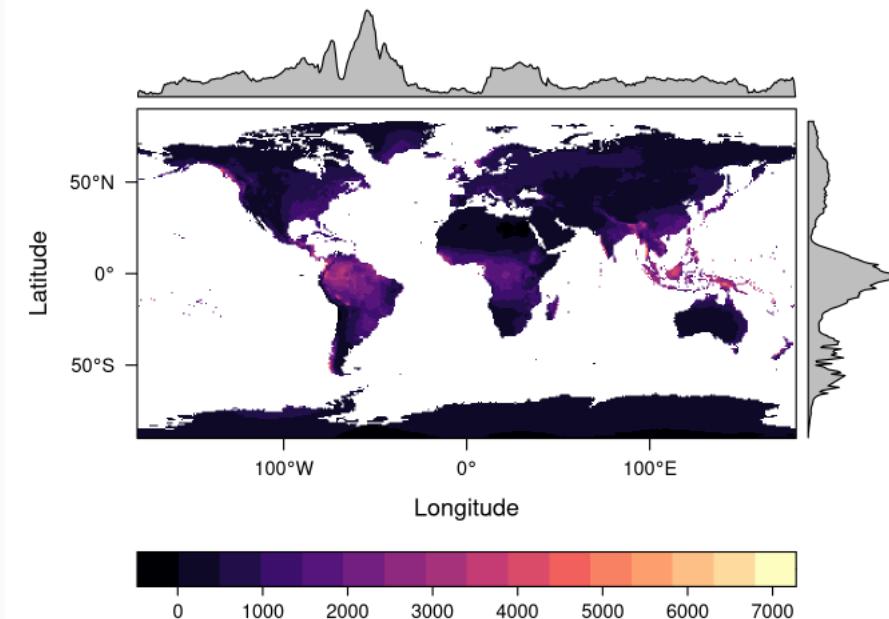
```
# plot  
levelplot(bioclim,  
         layers = 1,  
         margin = list(FUN = "median"))
```



3. Mapas estáticos

rasterVis

```
# plot  
levelplot(bioclim,  
         layers = 4,  
         margin = list(FUN = "median"))
```



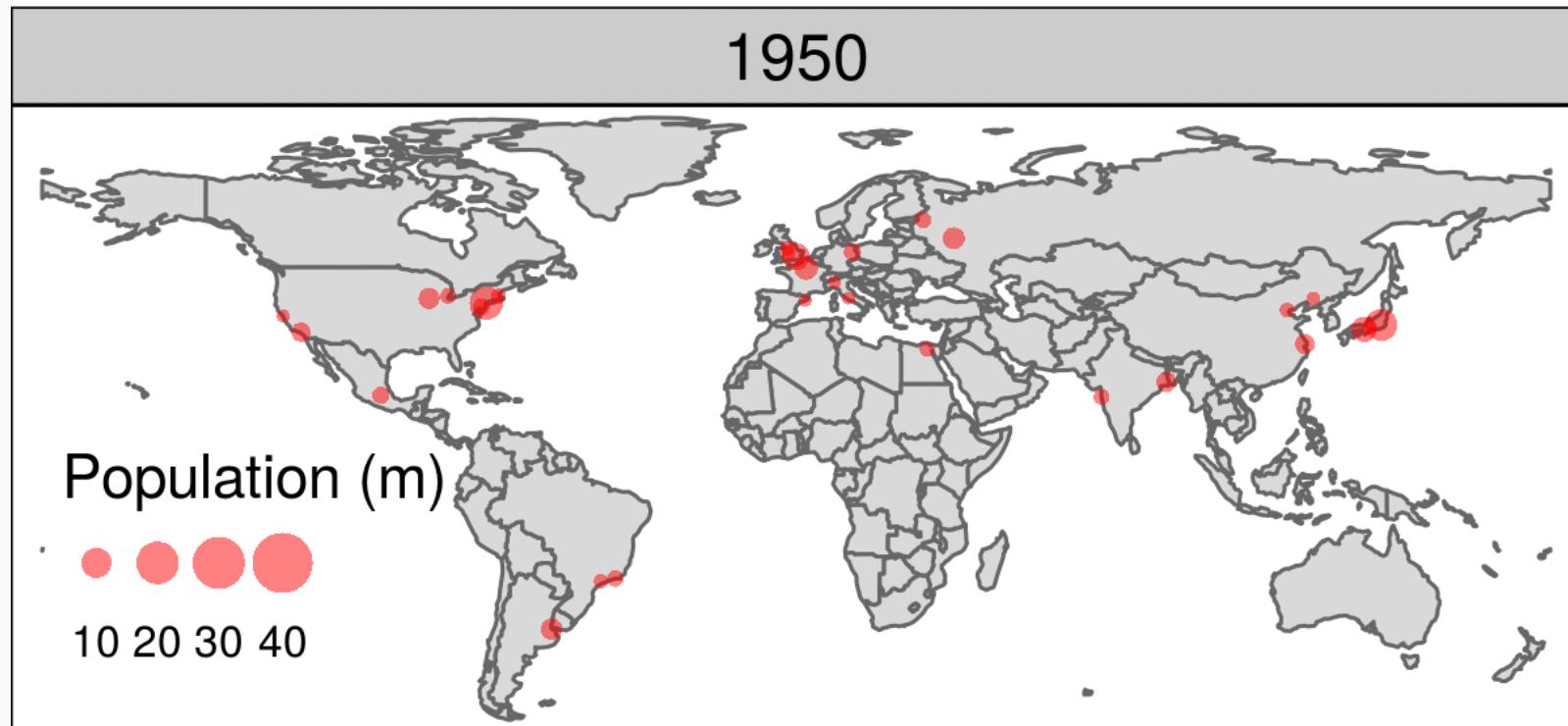
3. Mapas estáticos

rasterVis

```
# export
png(filename = here::here("03_dados", "mapas", "mapa_bioclim_rastervis.png"),
    wi = 20, he = 20, un = "cm", res = 300)
levelplot(bioclim,
    layers = 4,
    margin =
        list(FUN = "median"))
dev.off()
```

4. Mapas animados

Mapas animados são utilizados para entender a mudanças espaciais e/ou no tempo



4. Mapas animados

Mudança dos estados do Brasil ao longo do tempo

```
# dados
br_anos ← NULL
for(i in c(1872, 1900, 1911, 1920, 1933, 1940, 1950, 1960, 1970, 1980, 1991, 2001, 2010, 2020)){
  br_anos ← geobr::read_state(code_state = "all", year = i, showProgress = FALSE) %>%
    sf:::st_geometry() %>%
    sf:::st_as_sf() %>%
    dplyr::mutate(year = i) %>%
    dplyr::bind_rows(br_anos, .)
}
```

4. Mapas animados

Mudança dos estados do Brasil ao longo do tempo

```
# mapa facetado  
map_brasil_tmap ← tm_shape(br_anos) +  
  tm_polygons() +  
  tm_facets(by = "year", nrow = 4)  
map_brasil_tmap
```



4. Mapas animados

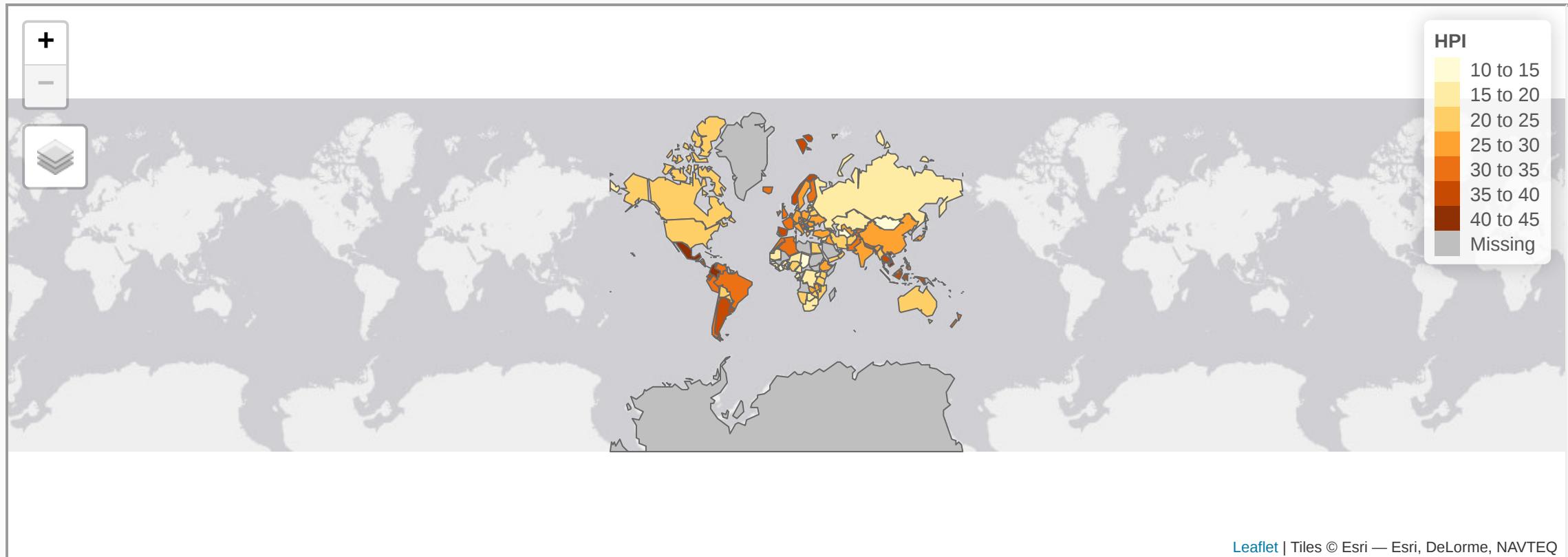
Mudança dos estados do Brasil ao longo do tempo

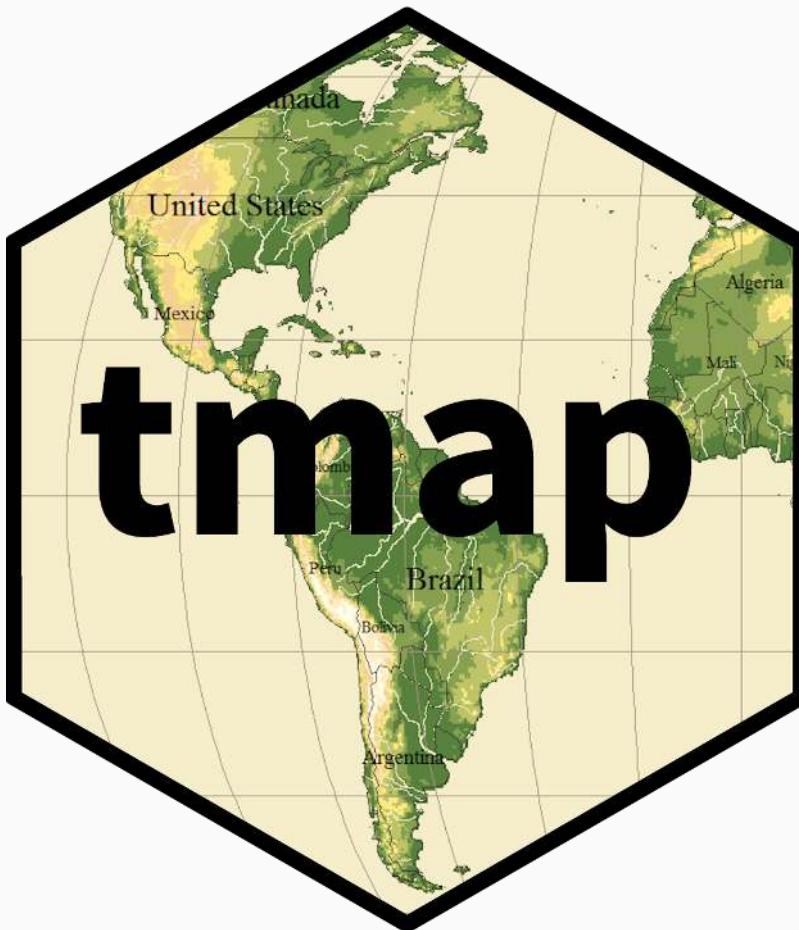
```
# exportar
tmap :: tmap_animation(
  tm = map_brasil_tmap_ani,
  filename = here::here("03_dados",
                        "mapas",
                        "mapa_dem_rc_tmap_ani.gif"),
  delay = 30)
```



5. Mapas interativos

Mapas interativos são utilizados quando precisamos interagir





[tmap](#)

5. Mapas interativos

tmap

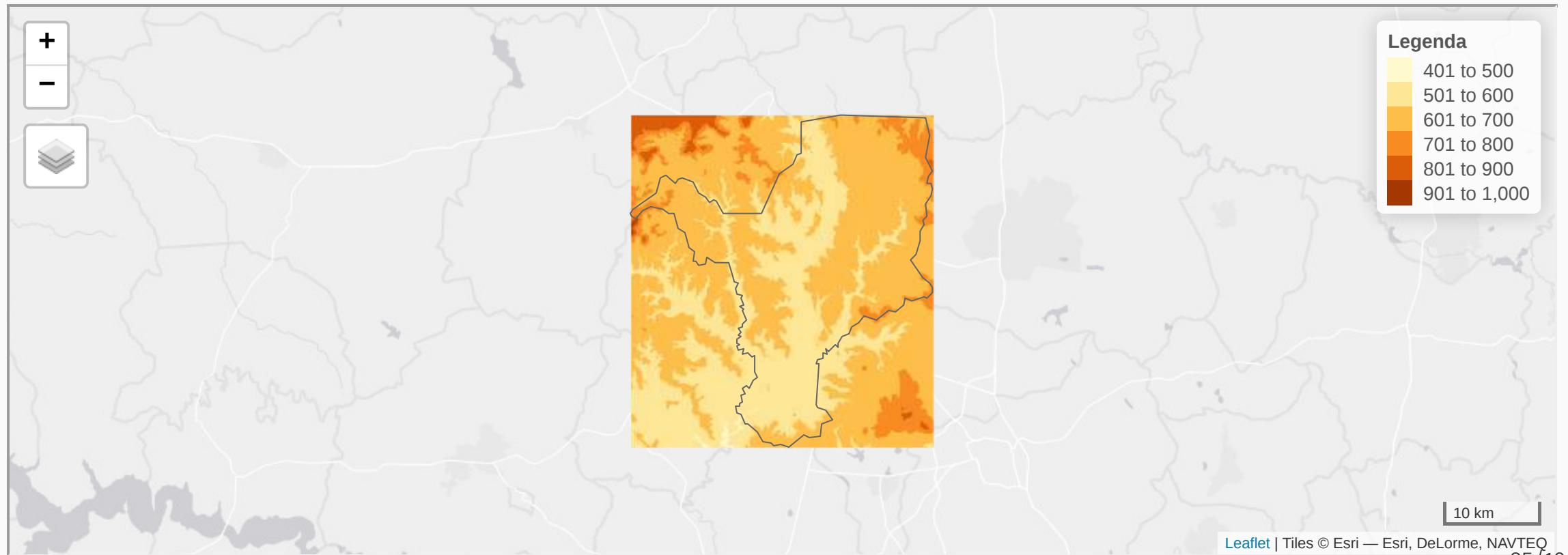
```
# mudar o modo de exibicao do tmap
tmap::tmap_mode(mode = "view")

map_biomas_tmap_int ← map_biomas_tmap
map_biomas_tmap_int
```

5. Mapas interativos

tmap

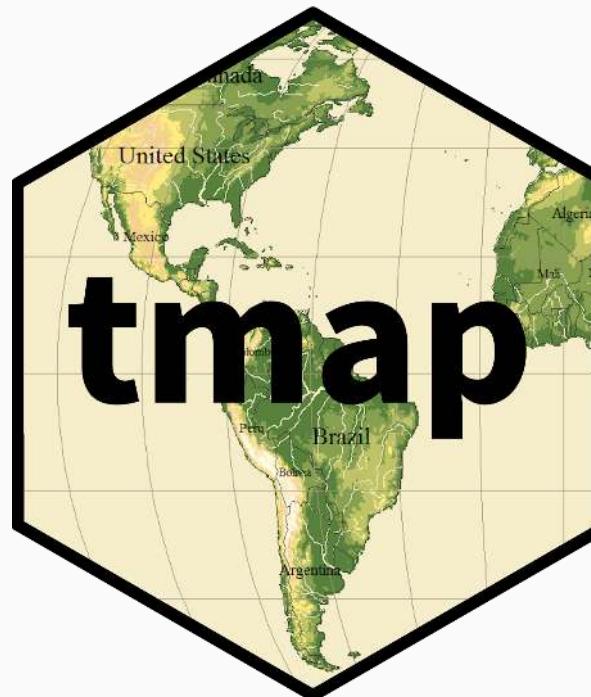
```
map_dem_rc_tmap_int ← map_dem_rc_tmap  
map_dem_rc_tmap_int
```



5. Mapas interativos

tmap

```
# exportar mapa tmap interativo
tmap::tmap_save(tm = map_dem_rc_tmap_int,
                filename = here::here("dados", "mapas", "mapa_dem_rc_tmap_int.html"))
```





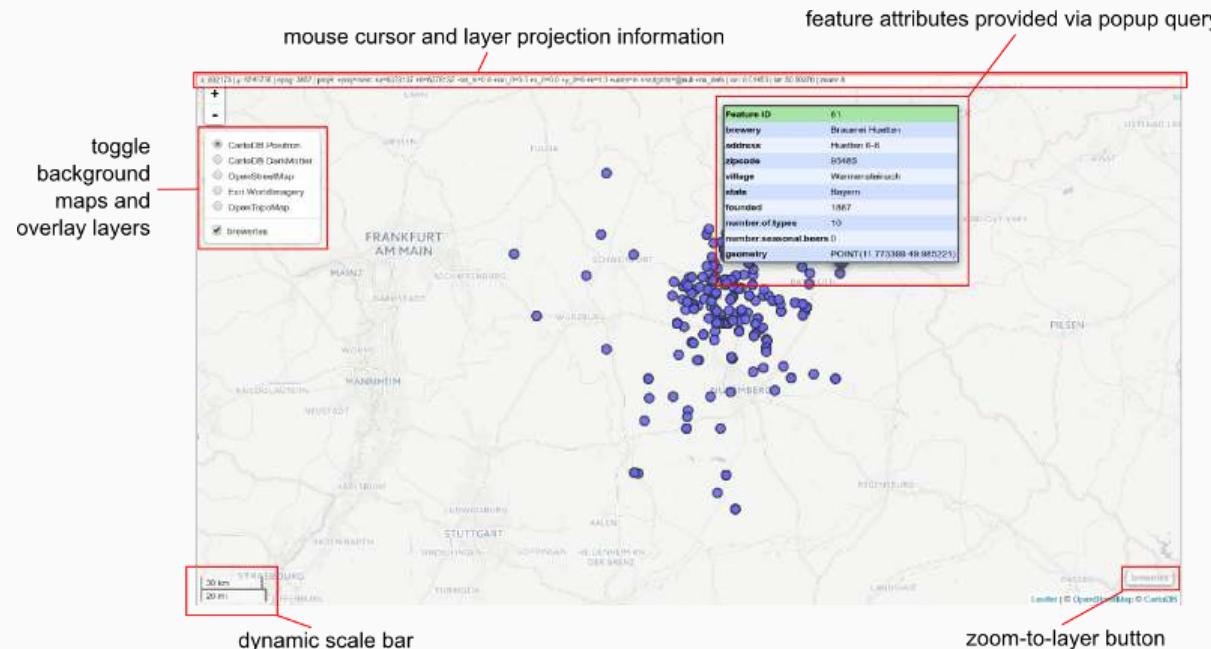
[mapview](#)

5. Mapas interativos

mapview

Display

```
# pacote  
# install.packge("mapview")  
library(mapview)
```



[mapview](#)

5. Mapas interativos

mapview

```
# plot
map_dem_rc_mapview_int ← mapview::mapview(dem_rc, col.regions = viridis::viridis(100))
map_dem_rc_mapview_int
```

5. Mapas interativos

mapview

```
# exportar mapa tmap interativo
mapview::mapshot(x = map_dem_rc_mapview_int,
                  url = here::here("03_dados", "mapas", "mapa_dem_rc_mapview_int.html"))
```





5. Mapas interativos

leaflet

```
# pacote  
# install.packge("leaflet")  
library(leaflet)
```



[leafletjs](#), [leaflet for R](#)

Leaflet for R

Introduction

The Map Widget

Basemaps

Markers

Popups and Labels

Lines and Shapes

GeoJSON and TopoJSON

Raster Images

Shiny Integration

Colors

Legends

Show/Hide Layers

Choropleths

Projections

Additional Features

Introduction

Leaflet is one of the most popular open-source JavaScript libraries for interactive maps. It's used by websites ranging from [The New York Times](#) and [The Washington Post](#) to [GitHub](#) and [Flickr](#), as well as GIS specialists like [OpenStreetMap](#), [Mapbox](#), and [CartoDB](#).

This R package makes it easy to integrate and control Leaflet maps in R.

Features

- Interactive panning/zooming
- Compose maps using arbitrary combinations of:
 - Map tiles
 - Markers
 - Polygons
 - Lines
 - Popups
 - GeoJSON
- Create maps right from the R console or RStudio
- Embed maps in [knitr/R Markdown](#) documents and [Shiny](#) apps
- Easily render spatial objects from the [sp](#) or [sf](#) packages, or data frames with latitude/longitude columns
- Use map bounds and mouse events to drive Shiny logic
- Display maps in non spherical mercator projections
- Augment map features using chosen plugins from [leaflet plugins repository](#)

Installation

To install this R package, run this command at your R prompt:

```
install.packages("leaflet")  
# to install the development version from Github, run  
# devtools::install_github("rstudio/leaflet")
```

5. Mapas interativos

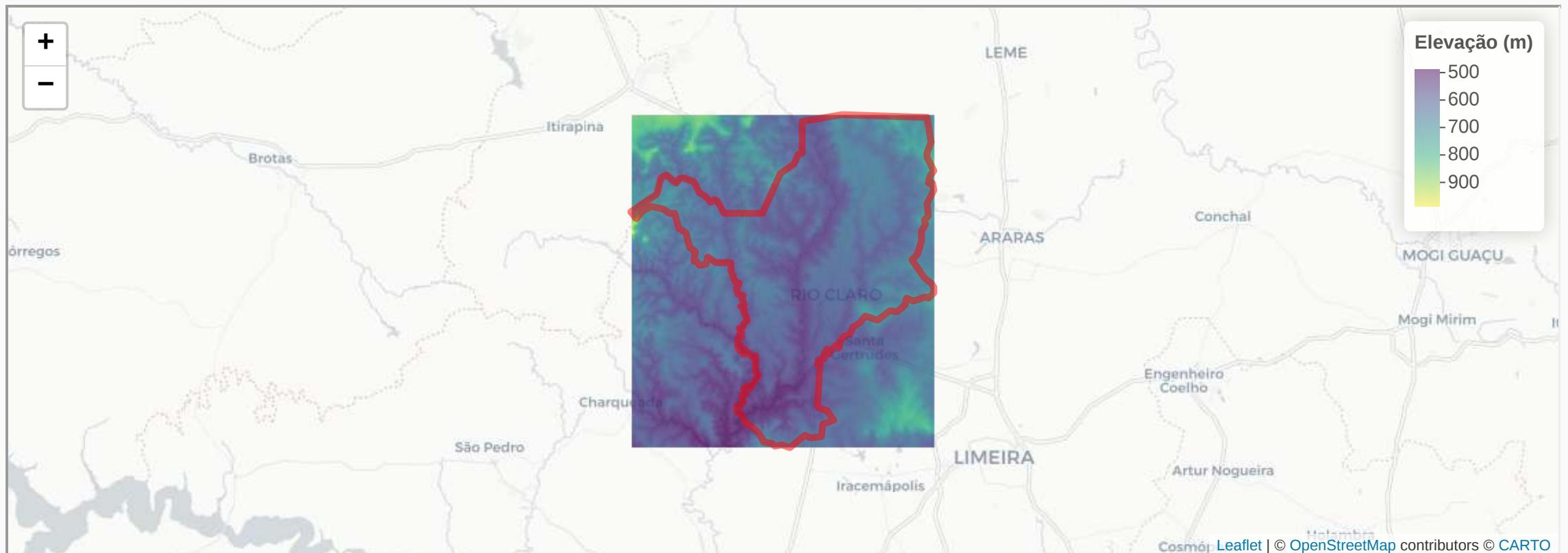
leaflet

```
# paleta de cores
pal ← colorNumeric(viridis::viridis(10), raster::values(dem_rc))

# mapa
map_dem_rc_leaflet_int ← leaflet() %>%
  addProviderTiles("CartoDB.Positron") %>%
  addRasterImage(dem_rc, colors = pal, opacity = .8) %>%
  addLegend(pal = pal, values = raster::values(dem_rc), title = "Elevação (m)") %>%
  addPolygons(data = rc_2020, col = "red", fill = NA)
map_dem_rc_leaflet_int
```

5. Mapas interativos

leaflet



5. Mapas interativos

leaflet

```
# exportar mapa tmap interativo
mapview::mapshot(x = map_dem_rc_leaflet_int,
                  url = here::here("03_dados", "mapas", "mapa_dem_rc_leaflet_int.html"))
```





[plotly](#)

5. Mapas interativos

plotly

```
# pacote  
# install.packge("plotly")  
library(plotly)
```

The screenshot shows two side-by-side views. On the left is an R console window displaying the execution of the provided R code, which results in a blank plot area. On the right is the official Plotly R documentation website. The header includes the Plotly logo, an R icon, and a 'Forum' link. A 'Suggest an edit to this page' button is visible. The main title is 'Plotly R Open Source Graphing Library'. Below it, a paragraph explains the library's purpose and encourages contributions. A callout box promotes deploying AI Dash apps on Kubernetes clusters. The 'Fundamentals' section is highlighted, featuring five sub-sections with corresponding icons: 'The Figure Data Structure', 'Creating and Updating Figures', 'Displaying Figures', 'Exporting Graphs as Static Images', and 'Configuration'.

[plotly](#)

5. Mapas interativos

plotly

```
# plot
map_rc_2020_plotly_int ← plot_geo(rc_2020)
map_rc_2020_plotly_int
```

5. Mapas interativos

plotly

```
# plot
map_rc_2020_plotly_int ← ggplotly(
  ggplot() +
    geom_sf(data = rc_2020) +
    theme_bw(base_size = 15))
map_rc_2020_plotly_int
```

5. Mapas interativos

plotly

```
# plot
map_biomas_plotly_int ← ggplotly(
  ggplot(data = biomas) +
    aes(fill = name_biome) +
    geom_sf(color = "black") +
    scale_fill_manual(values = c("darkgreen", "orange", "orange4",
                                 "forestgreen", "yellow", "yellow3")) +
    theme_bw(base_size = 15) +
    annotate(geom = "text", label = "CRS: SIRGAS2000/Geo", x = -38, y = -31, size = 2.5) +
    annotate(geom = "text", label = "Fonte: IBGE (2019)", x = -39, y = -32.5, size = 2.5) +
    labs(title = "Biomas do Brasil", fill = "Legenda", x = "Longitude", y = "Latitude"))
map_biomas_plotly_int
```

5. Mapas interativos

plotly

```
# exportar mapa tmap interativo
mapview::mapshot(x = map_biomas_plotly_int,
                  url = here::here("03_dados", "mapas", "mapa_dem_rc_leaflet_int.html"))
```



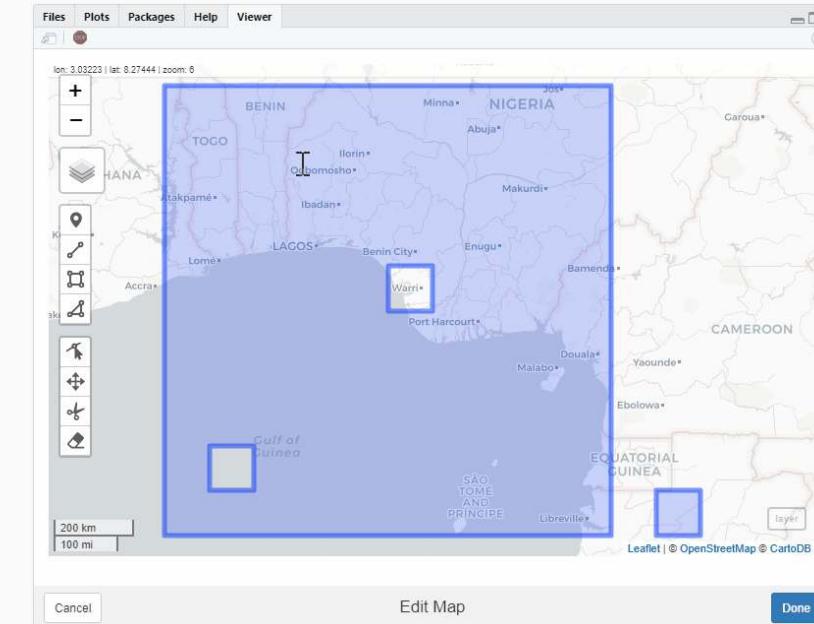
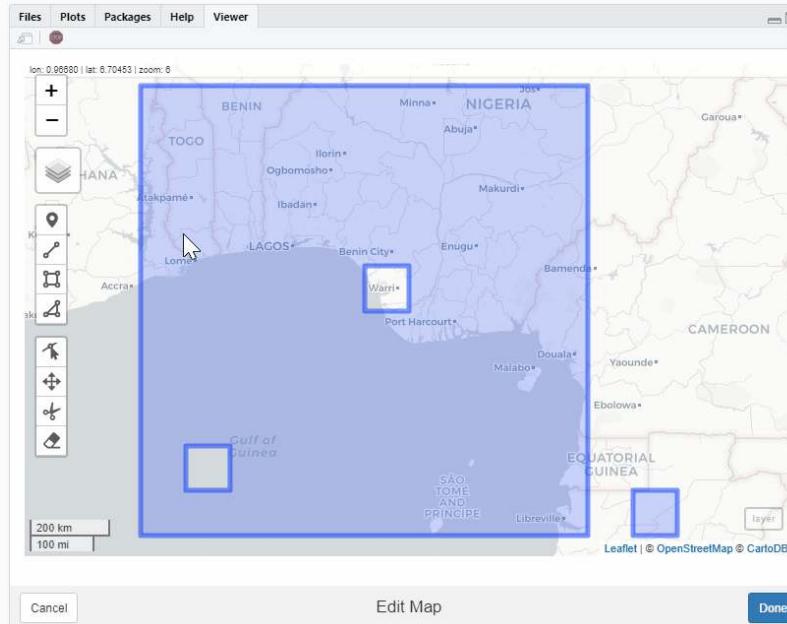
E para finalizar, vamos editar dados vetoriais no R

5. Mapas interativos

mapedit

Edições simples de pontos, linhas e polígonos

```
# pacote  
# install.packages("mapedit")  
library(mapedit)
```



5. Mapas interativos

mapedit

```
# criar
unesp <- mapedit::drawFeatures()
unesp

# editar
unesp_editado <- mapedit::editFeatures(unesp)
unesp_editado

# mapa
mapview::mapview(unesp_editado)
```

Dúvidas?

Muito bem, finalizamos a parte teórico-prática...

Agora vem a parte do projeto, ou seja, onde vocês
trabalham

Gostaria de agradecer todas e todos pela paciência e
confiança 😊

Maurício Vancine

Contatos:

✉ mauricio.vancine@gmail.com

🐦 [@mauriciovancine](https://twitter.com/mauriciovancine)

🐙 [mauriciovancine](https://github.com/mauriciovancine)

🔗 mauriciovancine.github.io



Slides criados via pacote [xaringan](#) e tema [Metropolis](#). Animação dos sapos por [@probzz](#).