

Introdução à análise geoespacial com R

8 Produção de mapas

Maurício H. Vancine

Milton C. Ribeiro

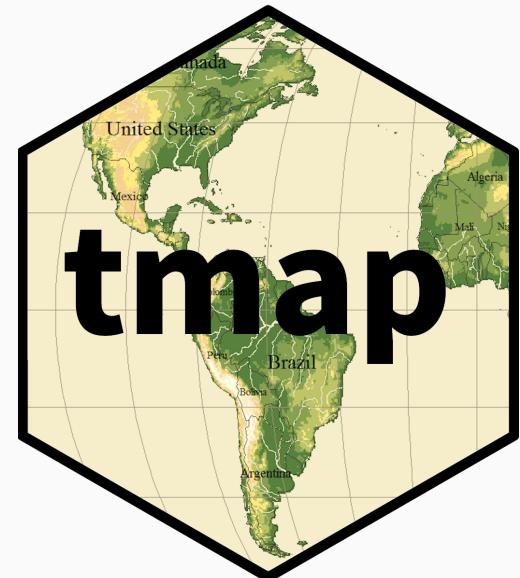
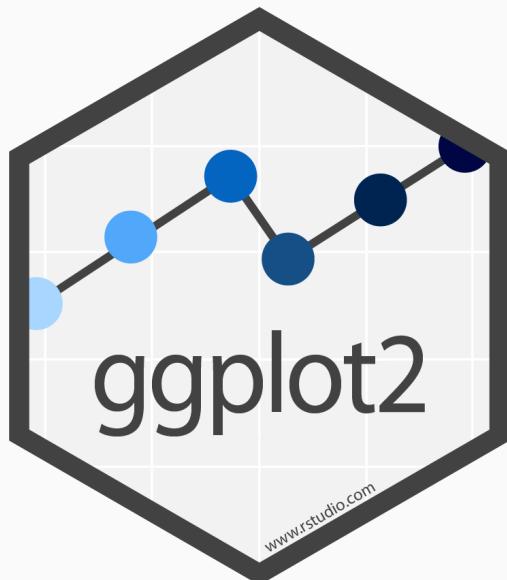
23/10/2020



8 Produção de mapas

Conteúdo

1. Elementos de um mapa
2. Pacotes para produção de mapas
3. Pacote ggplot2
4. Pacote tmap
5. Mapas vetoriais
6. Mapas matriciais
7. Mapas estáticos
8. Mapas animados
9. Mapas interativos
10. Exportar mapas



8 Produção de mapas

Script

```
08_script_intro_geocomp_r.R
```

8.1 Elementos de um mapa

Elementos

- Título
- Mapa principal
- Mapa secundário
- Legenda
- Coordenadas
- Orientação (Norte)
- Barra de escala
- SRC
- Fontes



8.2 Pacotes para produção de mapas

Pacotes

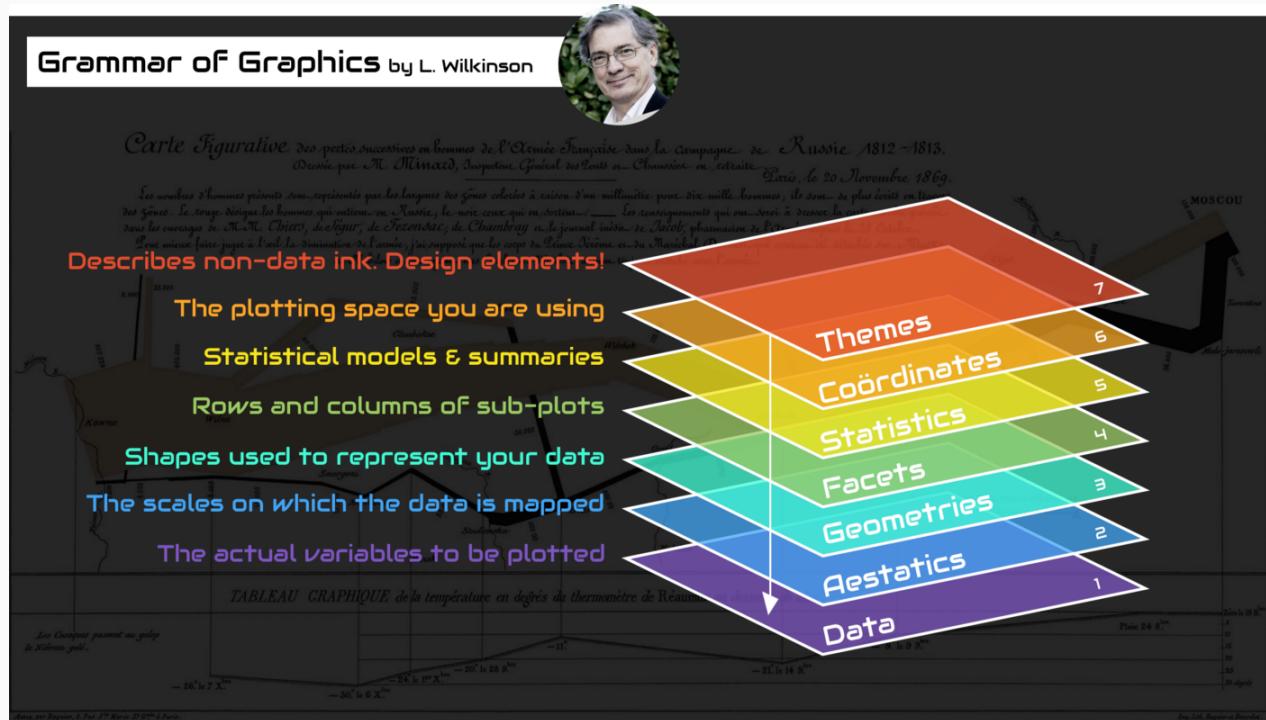
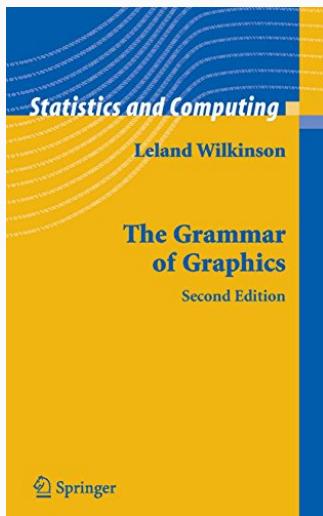
Principais pacotes para produção de mapas no R

| | |
|--------------------|---|
| <u>tmap</u> | Mapas temáticos |
| <u>ggplot2</u> | Cria mapas eleganter usando a Grática de Gráficos |
| <u>ggspatial</u> | Estrutura de dados espaciais para ggplot2 |
| <u>cartography</u> | Cartografia temática |
| <u>googleway</u> | Acessa Google Maps APIs para recuperar dados e traçar mapas |
| <u>leaflet</u> | Cria Web Maps interativos with Leaflet |
| <u>mapview</u> | Visualização interativa de dados espaciais em R |
| <u>plotly</u> | Crie gráficos interativos da Web por meio de ‘plotly.js’ |
| <u>rasterVis</u> | Métodos de visualização para dados raster |

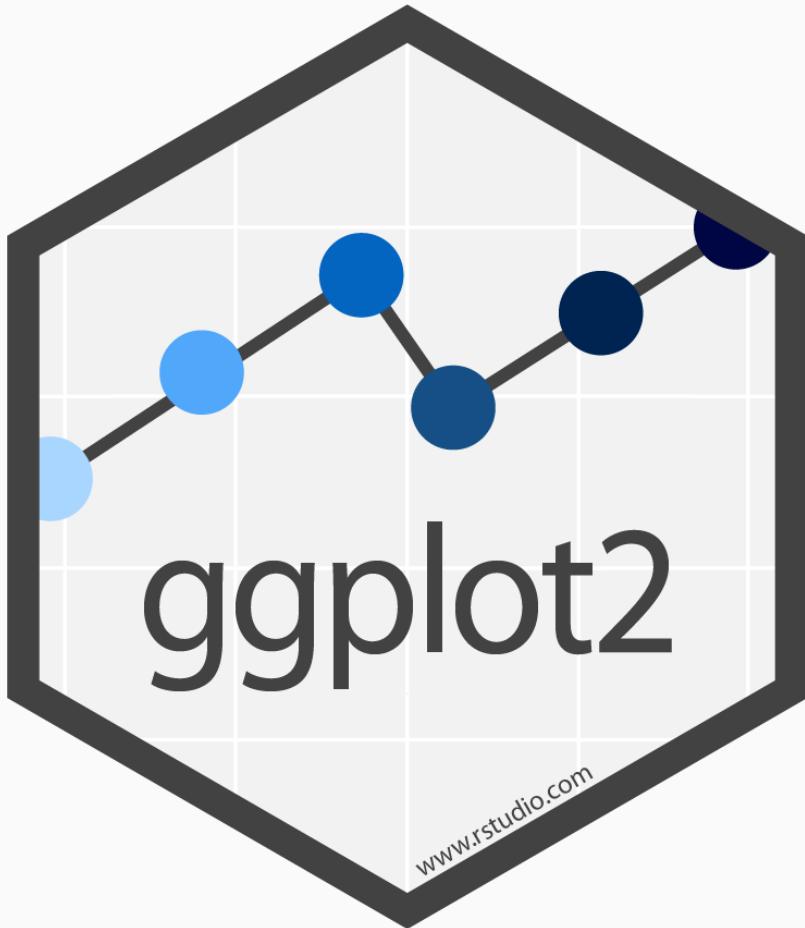
8.2 Pacotes para produção de mapas

The Grammar of Graphics (1999)

Os pacotes `ggplot2` e `tmap` implementam uma adaptação da **gramática dos gráficos** em "layers"



Fonte: <https://medium.com/tdebeus/think-about-the-grammar-of-graphics-when-improving-your-graphs-18e3744d8d18>



8.3 Pacote ggplot2

ggplot2

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

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

Estrutura:

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

Fonte: [@allison_horst](#)

[*] <https://ggplot2.tidyverse.org/>

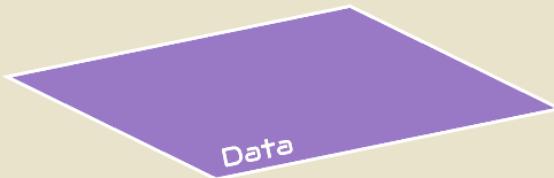


8.3 Pacote ggplot2

ggplot2

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("ggplot2")
35 # Then activate ggplot2 package
36 library(ggplot2)
37
38 # Create new variable for plot only x and y axis, ('data' and 'aesthetics' layer)
39 plot <- ggplot(data=new.data, aes(x=Genre, y=Gross...$Gross))
40
41 # Create new variable with geometries layer.
42 p <- plot + geom_boxplot(outlier.colour = NA) # removes boxplot layer outliers.
43
44 # Change axis and title if needed.
45
46 xlab("Genre") +
47 ylab("Gross X US") +
48 ggtitle("Domestic Gross % by Genre")
49
50 # Have your plot visually attractive and readable with the 'theme' function. (Theme layer)
51 q <- theme_cbo(element.text.size = 12,
52 axis.text = element.text.size = 12),
53 legend.title = element.text.size = 12),
54 legend.text = element.text.size = 12),
55 panel.grid.major.size = 1.5, # 'hjust' will center your text
56 panel.background = element_rect(fill = "#E6E3C5"))
57
```

Fonte: <https://medium.com/tdebeus/think-about-the-grammar-of-graphics-when-improving-your-graphs-18e3744d8d18>

8.3 Pacote ggplot2

Data Visualization Cheatsheet

Data Visualization with ggplot2 :: CHEAT SHEET

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and **geoms**—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(mapping = aes(< MAPPINGS >),  
    stat = <STAT>, position = <POSITION>) +  
  <COORDINATE_FUNCTION> +  
  <FACET_FUNCTION> +  
  <SCALE_FUNCTION> +  
  <THEME_FUNCTION>
```

ggplot(data = mpg, aes(x = cyl, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

```
  aesthetic mappings | data | geom  
  plot(x = cyl, y = hwy, data = mpg, geom = "point")  
Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.
```

last_plot() Returns the last plot

```
  ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory.  
  Matches file type to file extension.
```

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

```
a <- ggplot(economics, aes(date, unemployed))  
b <- ggplot(seals, aes(z = long, y = lat))  
  
a + geom_blank()  
(Useful for expanding limits)  
  
b + geom_curve(aes(yend = lat + 1,  
  xend = long + 1, curvature = -2)) x, yend, y, start,  
alpha, angle, color, curvature, linetype, size  
  
a + geom_path(linewidth = "butt", linejoin = "round",  
  x, y, alpha, color, group, linetype, size  
  
a + geom_polygon(aes(group = group)) x, y, alpha, color, fill, group, linetype, size  
  
b + geom_rect(aes(xmin = long, ymin = lat + 1), xmax =  
  long + 1, ymax = lat + 1) x, y, alpha, xmin, ymax,  
ymin, alpha, color, fill, group, linetype, size  
  
a + geom_ribbon(aes(ymin = unemployed - 900,  
  ymax = unemployed + 900)) x, y, alpha, ymin, ymax,  
alpha, color, fill, group, linetype, size
```

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size
b + geom_abline(aes(intercept = 0, slope = 1))
b + geom_hline(aes(intercept = 1))
b + geom_vline(aes(intercept = long))

b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:1155, radius = 1))

ONE VARIABLE continuous

```
c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)  
  
c + geom_area(stat = "bin") x, y, alpha, fill, group,  
linetype, size  
c + geom_density(kernel = "gaussian")  
  x, y, alpha, color, fill, group, linetype, size, weight  
  
c + geom_dotplot() x, y, alpha, color, fill,  
linetype, size  
c + geom_freqpoly() x, y, alpha, color, group,  
linetype, size  
c + geom_histogram(binwidth = 5) x, y, alpha,  
color, fill, linetype, size, weight  
c2 + geom_qq(aes(sample = hwy)) x, y, alpha,  
color, fill, linetype, size, weight
```

discrete

```
d <- ggplot(mpg, aes(fct))  
d + geom_bar()
```

```
x, alpha, color, fill, linetype, size, weight
```

TWO VARIABLES

```
continuous x, continuous y  
continuous x, discrete y  
discrete x, continuous y  
discrete x, discrete y
```

```
f <- ggplot(mpg, aes(class, hwy))  
  
f + geom_boxplot() x, y, lower, middle, upper,  
ymin, ymax, alpha, color, fill, group, linetype,  
shape, size, weight  
f + geom_dotplot(binaxis = "y", stackdir =  
  "center") x, y, alpha, color, fill, group  
f + geom_violin(scale = "area") x, y, alpha, color,  
fill, group, linetype, size, weight
```

discrete x , discrete y

```
g <- ggplot(diamonds, aes(cut, color))  
  
g + geom_count() x, y, alpha, color, fill, shape,  
size, stroke
```

THREE VARIABLES

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))  
l <- ggplot(seals, aes(long, lat))  
l + geom_contour(aes(z ~ z)) x, y, z, alpha, colour, group, linetype,  
size, weight  
  
l + geom_raster(aes(fill = z), hijust = 0.5, vjust = 0.5,  
  interpolate = FALSE) x, y, alpha, fill  
l + geom_tile(aes(fill = z)) x, y, alpha, color, fill,  
linetype, size, width
```

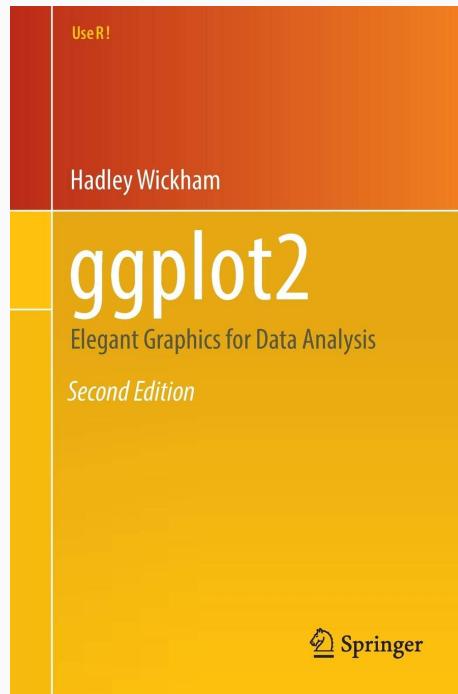


RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at <http://ggplot2.tidyverse.org> • ggplot2 2.1.0 • Updated: 2016-11

8.3 Pacote ggplot2

Livros

ggplot2 (2009, 2016)

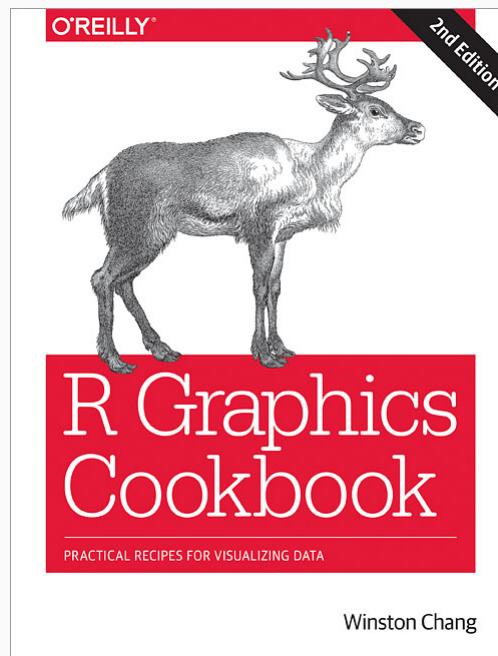


[*] <https://ggplot2-book.org/>

8.3 Pacote ggplot2

Livros

R Graphics Cookbook (2018)

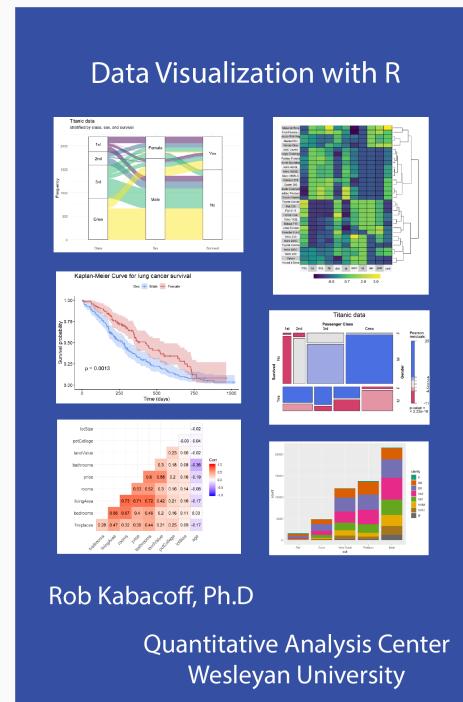


[*] <http://www.cookbook-r.com/Graphs/>

8.3 Pacote ggplot2

Livros

Data Visualization with R (2018)

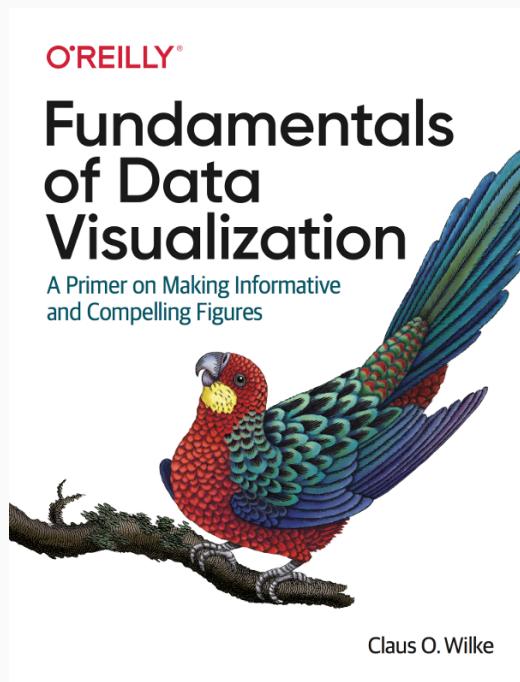


[*] <https://rkabacoff.github.io/datavis/>

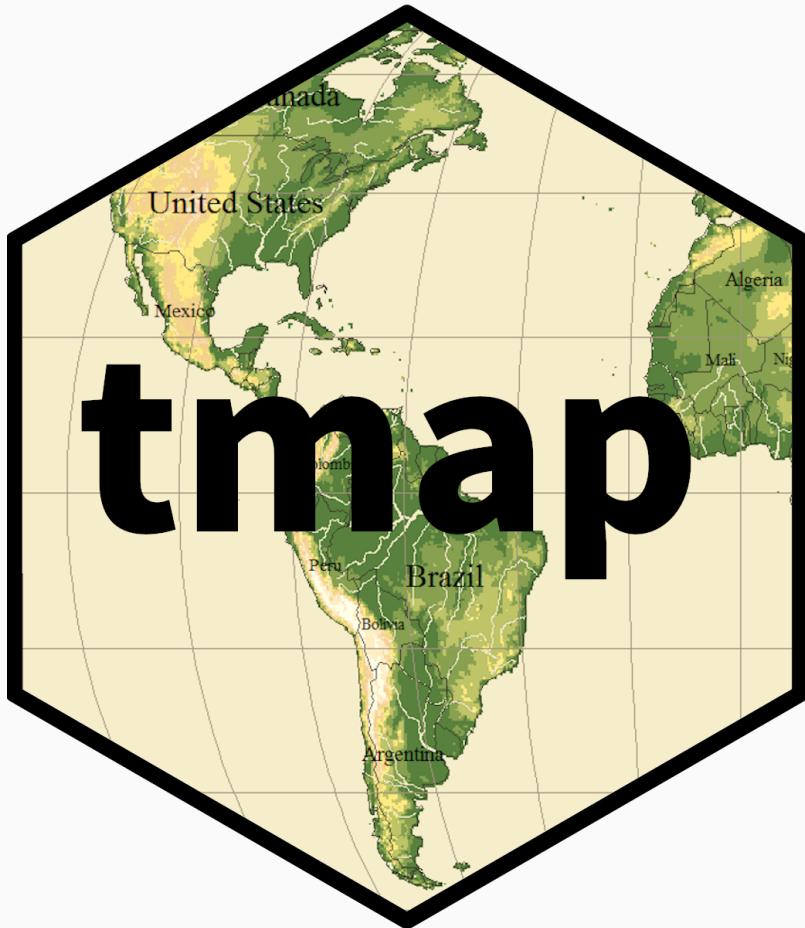
8.3 Pacote ggplot2

Livros

Fundamentals of Data Visualization (2019)



[*] <https://serialmentor.com/dataviz/>



8.4 Pacote tmap

tmap

Artigo

- Tennekes, Martijn. "[tmap: Thematic Maps in R.](#)" Journal of Statistical Software 84.06 (2018): 1-39.



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.

8.4 Pacote tmap

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( ... )  
tmap_arrange( ... )  
tm_mode( ... )
```

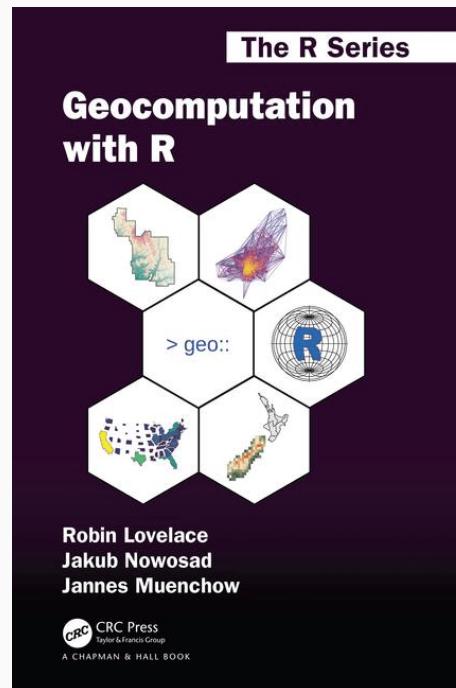


[*] <https://mtennekes.github.io/tmap/>

8.4 Pacote tmap

Livro

Geocomputation with R (2019)



[*] <https://geocompr.robinlovelace.net/adv-map.html>

Mapas vetoriais

8.5 Mapas vetoriais

Limite de Rio Claro/SP

```
# rio claro
rc_2019 ← geobr::read_municipality(code_muni = 3543907, year = 2019)
rc_2019
plot(rc_2019$geom, col = "gray", main = NA, axes = TRUE, graticule = TRUE)
```

8.5 Mapas vetoriais

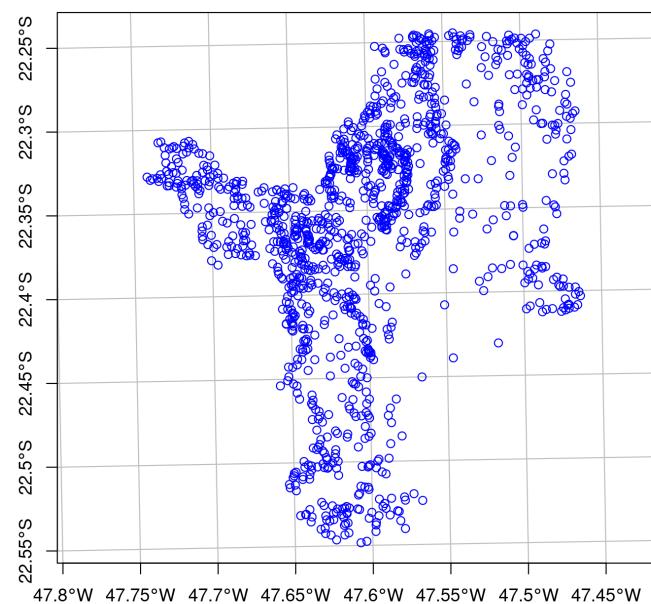
Limite de Rio Claro/SP

```
# rio claro utm
rc_2019_utm ← geobr::read_municipality(code_muni = 3543907, year = 2019) %>% sf::
rc_2019_utm
plot(rc_2019_utm$geom, col = "gray", main = NA, axes = TRUE, graticule = TRUE)
```

8.5 Mapas vetoriais

Pontos - nascentes

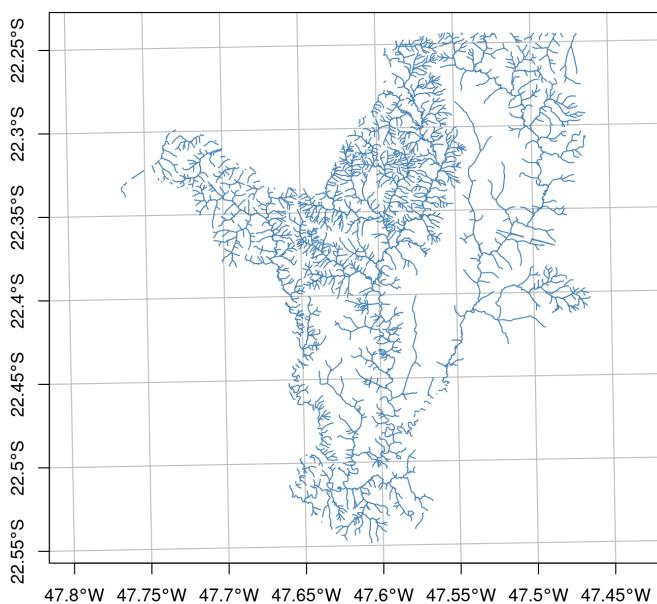
```
# import points
rc_spr ← sf::st_read(here::here("03_dados", "vetor", "SP_3543907_NASCENTES.shp"),
rc_spr
plot(rc_spr$geometry, col = "blue", main = NA, axes = TRUE, graticule = TRUE)
```



8.5 Mapas vetoriais

Linhas - rios

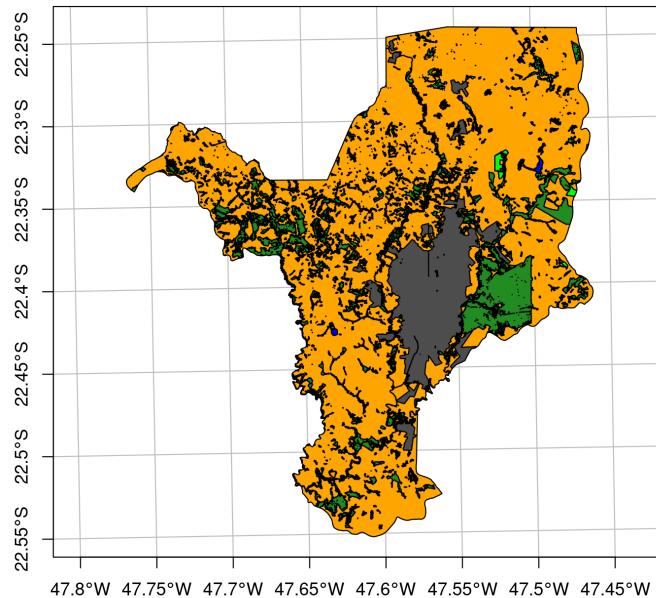
```
# import lines
rc_riv ← sf::st_read(here::here("03_dados", "vetor", "SP_3543907_RIOS_SIMPLES.shp"))
rc_riv
plot(rc_riv$geometry, col = "steelblue", main = NA, axes = TRUE, graticule = TRUE)
```

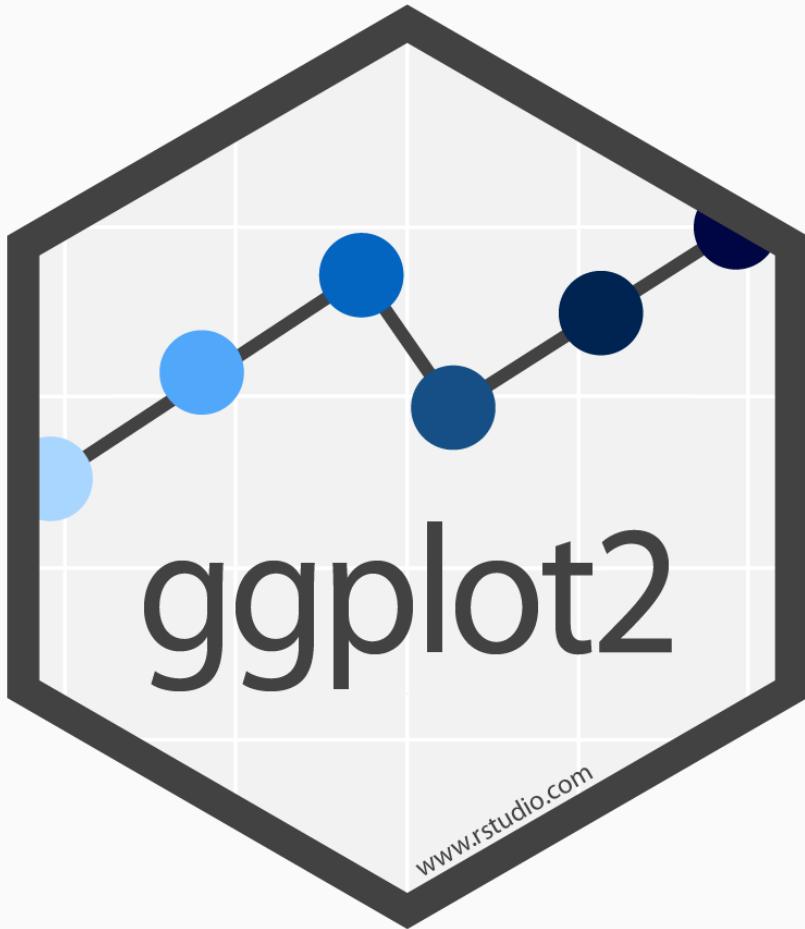


8.5 Mapas vetoriais

Polígonos - cobertura da terra

```
# import polygons
rc_use ← sf::st_read(here::here("03_dados", "vetor", "SP_3543907_US0.shp"), quiet)
rc_use
plot(rc_use$geometry, col = c("blue", "orange", "gray30", "forestgreen", "green"),
```



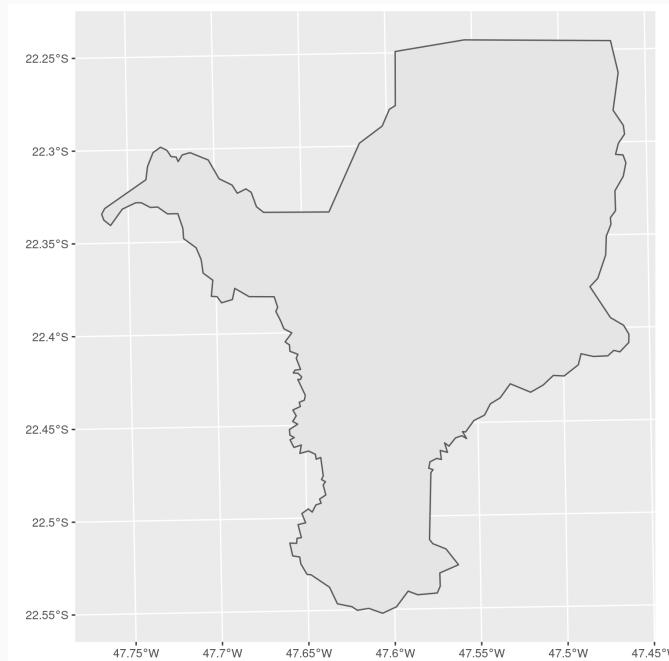


8.5 Mapas vetoriais

ggplot2

Mapa do limite de Rio Claro/SP

```
# rio claro limit
ggplot() +
  geom_sf(data = rc_2019_utm)
```

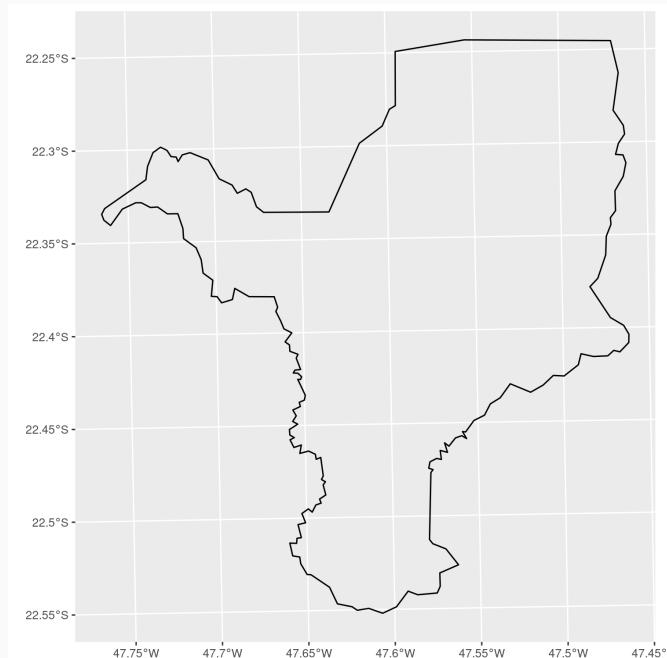


8.5 Mapas vetoriais

ggplot2

Mapa do limite de Rio Claro/SP - cor e preenchimento

```
# rio claro limit color and fill  
ggplot() +  
  geom_sf(data = rc_2019_utm, color = "black", fill = NA)
```

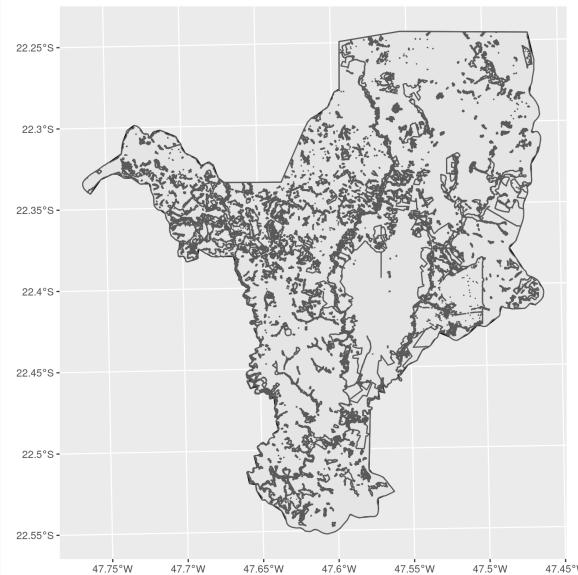


8.5 Mapas vetoriais

ggplot2

Limite e cobertura da terra

```
# rio claro limit fill + land use
ggplot() +
  geom_sf(data = rc_2019_utm, color = "black", fill = NA) +
  geom_sf(data = rc_use)
```



8.5 Mapas vetoriais

ggplot2

Limite e cobertura da terra com cores

```
# rio claro limit fill + land use with colors
ggplot() +
  geom_sf(data = rc_2019_utm, color = "black", fill = NA) +
  geom_sf(data = rc_use, aes(fill = CLASSE_USO), color = NA)
```

8.5 Mapas vetoriais

ggplot2

Limite e cobertura da terra com cores (inverter)

```
# land use with colors + rio claro limit fill
ggplot() +
  geom_sf(data = rc_use, aes(fill = CLASSE_USO), color = NA) +
  geom_sf(data = rc_2019_utm, color = "black", fill = NA)
```

8.5 Mapas vetoriais

ggplot2

Definir cores para as classes

```
# land use and choose colors + rio claro limit fill
ggplot() +
  geom_sf(data = rc_use, aes(fill = CLASSE_USO), color = NA) +
  geom_sf(data = rc_2019_utm, color = "black", fill = NA) +
  scale_fill_manual(values = c("blue", "orange", "gray30", "forestgreen", "green"))
```

8.5 Mapas vetoriais

ggplot2

Mudar o CRS (Sistema de Coordenadas)

```
# land use and choose colors + rio claro limit fill + coords
ggplot() +
  geom_sf(data = rc_use, aes(fill = CLASSE_USO), color = NA) +
  geom_sf(data = rc_2019_utm, color = "black", fill = NA) +
  scale_fill_manual(values = c("blue", "orange", "gray30", "forestgreen", "green"))
  coord_sf(datum = 31983)
```

8.5 Mapas vetoriais

ggplot2

Mudar o tema

```
# land use and choose colors + rio claro limit fill + coords + themes
ggplot() +
  geom_sf(data = rc_use, aes(fill = CLASSE_USO), color = NA) +
  geom_sf(data = rc_2019_utm, color = "black", fill = NA) +
  scale_fill_manual(values = c("blue", "orange", "gray30", "forestgreen", "green"))
  coord_sf(datum = 31983) +
  theme_bw()
```

8.5 Mapas vetoriais

ggplot2

Adicionar barra de escala e norte

```
# land use and choose colors + rio claro limit fill + coords + themes + scalebar +
ggplot() +
  geom_sf(data = rc_use, aes(fill = CLASSE_USO), color = NA) +
  geom_sf(data = rc_2019_utm, color = "black", fill = NA) +
  scale_fill_manual(values = c("blue", "orange", "gray30", "forestgreen", "green"))
  coord_sf(datum = 31983) +
  theme_bw() +
  annotation_scale(location = "br", width_hint = .3) +
  annotation_north_arrow(location = "br", which_north = "true",
                         pad_x = unit(0, "cm"), pad_y = unit(.7, "cm"),
                         style = north_arrow_fancy_orienteering)
```

8.5 Mapas vetoriais

ggplot2

Adicionar barra de escala e norte

8.5 Mapas vetoriais

ggplot2

Add título e nome dos eixos, ângulo das coords e atribuir

```
# land use and choose colors + rio claro limit fill + coords + themes + scalebar +
map_use_gg ← ggplot() +
  geom_sf(data = rc_use, aes(fill = CLASSE_USO), color = NA) +
  geom_sf(data = rc_2019_utm, color = "black", fill = NA) +
  scale_fill_manual(values = c("blue", "orange", "gray30", "forestgreen", "green"))
  coord_sf(datum = 31983) +
  theme_bw() +
  annotation_scale(location = "br", width_hint = .3) +
  annotation_north_arrow(location = "br", which_north = "true",
                         pad_x = unit(0, "cm"), pad_y = unit(.7, "cm"),
                         style = north_arrow_fancy_orienteering) +
  labs(x = "Longitude", y = "Latitude", title = "Cobertura da terra Rio Claro/SP (" +
    theme(legend.position = c(.18,.18),
          legend.box.background = element_rect(colour = "black"),
          axis.text.y = element_text(angle = 90, hjust = .5))
map_use_gg
```

8.5 Mapas vetoriais

ggplot2

Add título e nome dos eixos, ângulo das coords e atribuir

8.5 Mapas vetoriais

ggplot2

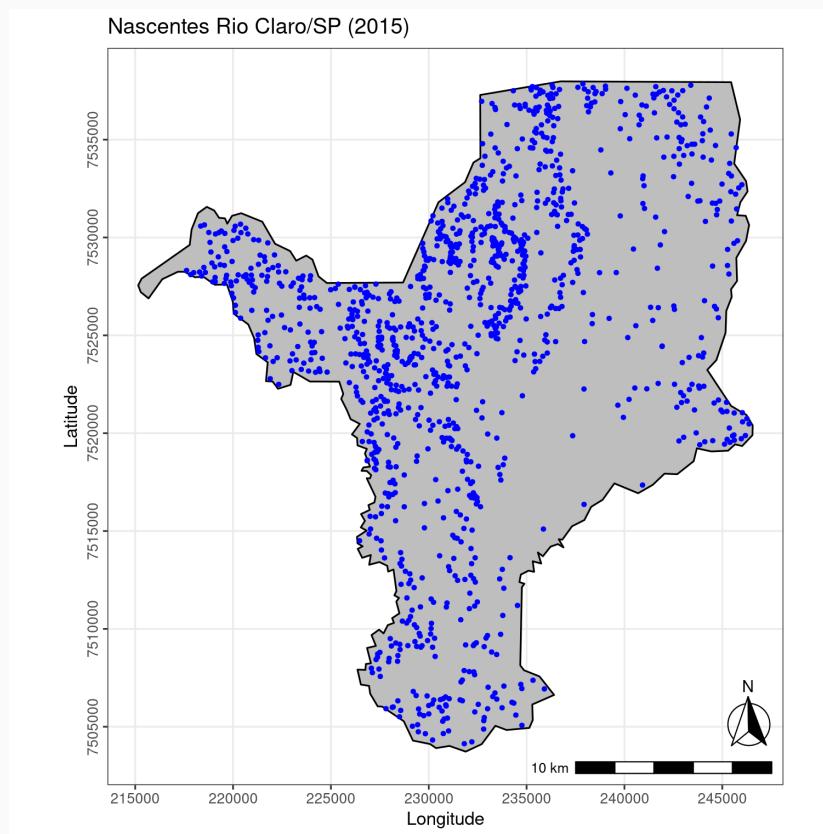
Nascentes

```
# springs
ggplot() +
  geom_sf(data = rc_2019_utm, color = "black", fill = "gray") +
  geom_sf(data = rc_spr, shape = 20, color = "blue") +
  coord_sf(datum = 31983) +
  theme_bw() +
  annotation_scale(location = "br", width_hint = .3) +
  annotation_north_arrow(location = "br", which_north = "true",
                         pad_x = unit(0, "cm"), pad_y = unit(.7, "cm"),
                         style = north_arrow_fancy_orienteering) +
  labs(x = "Longitude", y = "Latitude", title = "Nascentes Rio Claro/SP (2015)") +
  theme(legend.position = c(.18,.18),
        legend.box.background = element_rect(colour = "black"),
        axis.text.y = element_text(angle = 90, hjust = .5))
```

8.5 Mapas vetoriais

ggplot2

Nascentes



8.5 Mapas vetoriais

ggplot2

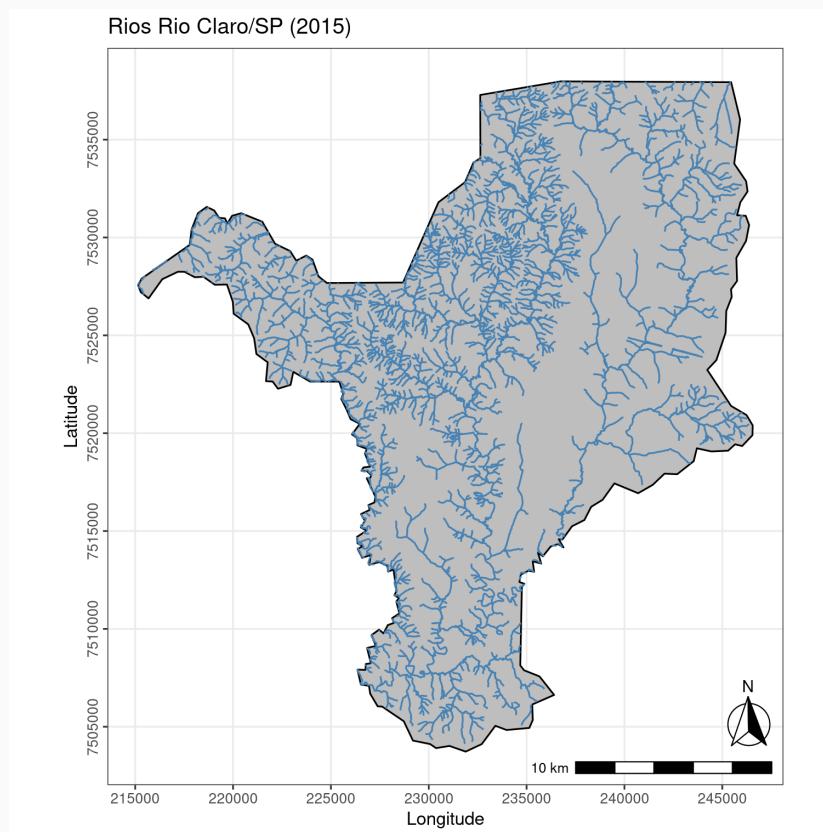
Rios

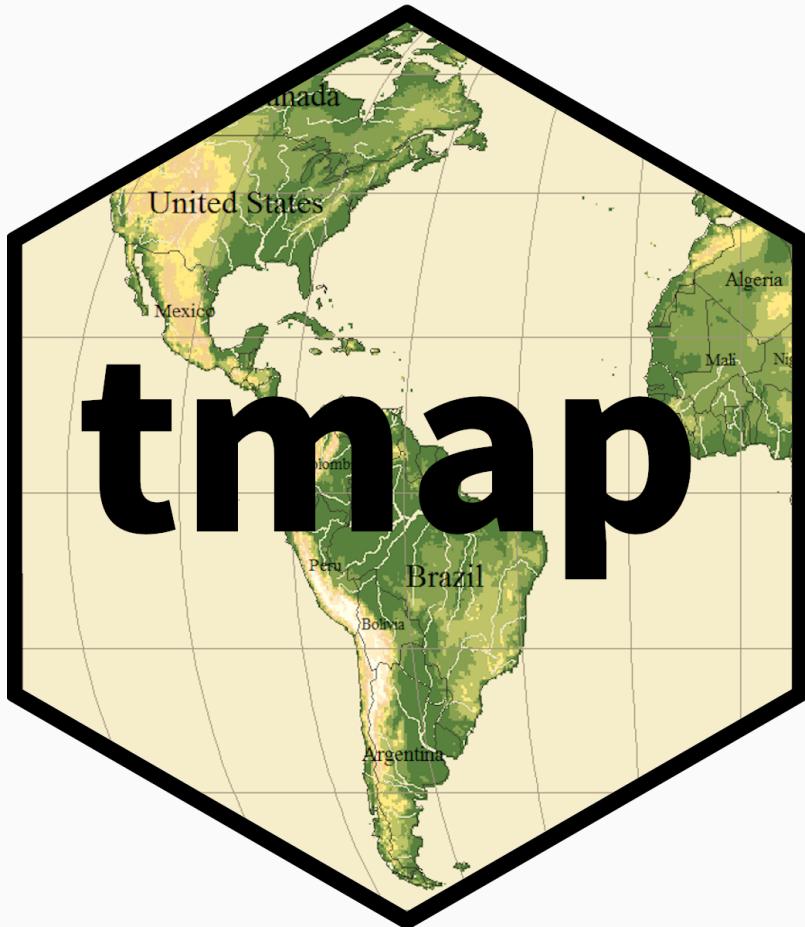
```
# rivers
ggplot() +
  geom_sf(data = rc_2019_utm, color = "black", fill = "gray") +
  geom_sf(data = rc_riv, shape = 20, color = "steelblue") +
  coord_sf(datum = 31983) +
  theme_bw() +
  annotation_scale(location = "br", width_hint = .3) +
  annotation_north_arrow(location = "br", which_north = "true",
                         pad_x = unit(0, "cm"), pad_y = unit(.7, "cm"),
                         style = north_arrow_fancy_orienteering) +
  labs(x = "Longitude", y = "Latitude", title = "Rios Rio Claro/SP (2015)") +
  theme(legend.position = c(.18,.18),
        legend.box.background = element_rect(colour = "black"),
        axis.text.y = element_text(angle = 90, hjust = .5))
```

8.5 Mapas vetoriais

ggplot2

Rios





8.5 Mapas vetoriais

ggplot2 e 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

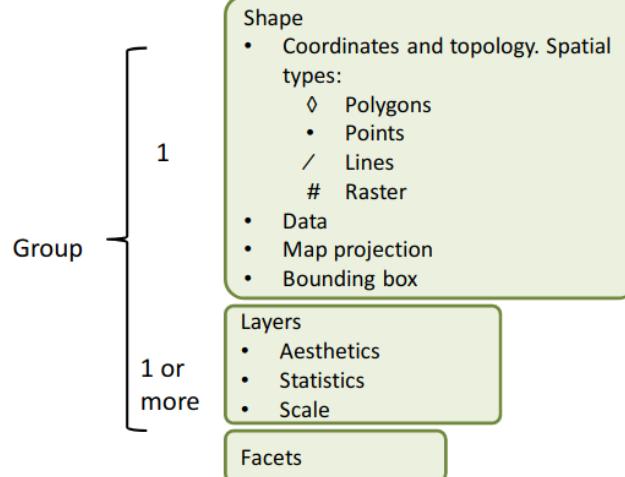
- Layers
 - Data
 - Aesthetics
 - Geometry
 - Statistics
 - Position

Scales

Coordinates

Facets

tmap Layered Grammar of Thematic Maps



8.5 Mapas vetoriais

tmap

Mapa do limite de Rio Claro/SP

```
# rio claro limit  
tm_shape(rc_2019_utm) +  
  tm_polygons()
```

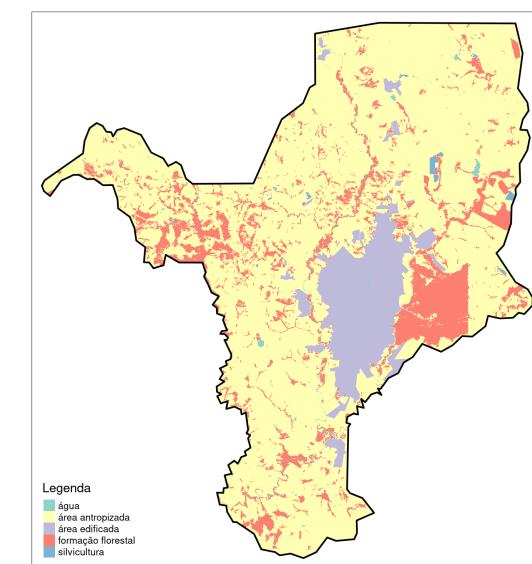


8.5 Mapas vetoriais

tmap

Limite e cobertura da terra

```
# rio claro limit fill + land use
tm_shape(rc_use) +
  tm_fill(col = "CLASSE_USO", title = "Legenda") +
  tm_shape(rc_2019_utm) +
  tm_borders(lwd = 2, col = "black")
```

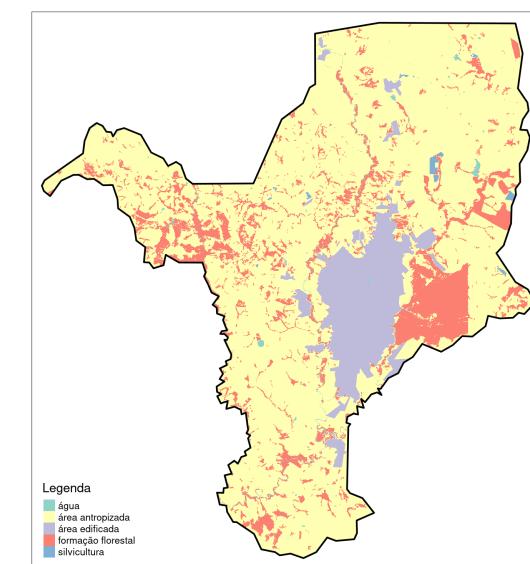


8.5 Mapas vetoriais

tmap

Limite e cobertura da terra com cores

```
# rio claro limit fill + land use with colors
tm_shape(rc_use) +
  tm_fill(col = "CLASSE_USO", title = "Legenda") +
  tm_shape(rc_2019_utm) +
  tm_borders(lwd = 2, col = "black")
```



8.5 Mapas vetoriais

tmap

Definir cores para as classes

```
# land use and choose colors + rio claro limit fill
tm_shape(rc_use) +
  tm_fill(col = "CLASSE_USO",
          pal = c("blue", "orange", "gray30", "forestgreen", "green"), title = "Le
tm_shape(rc_2019_utm) +
  tm_borders(lwd = 2, col = "black")
```

8.5 Mapas vetoriais

tmap

Add coordenadas

```
# land use and choose colors + rio claro limit fill + coords
tm_shape(rc_use) +
  tm_fill(col = "CLASSE_USO",
          pal = c("blue", "orange", "gray30", "forestgreen", "green"), title = "Le
tm_shape(rc_2019_utm) +
  tm_borders(lwd = 2, col = "black") +
  tm_grid(lines = FALSE, labels.format = list(big.mark = "")), labels.rot = c(0, 90
```

8.5 Mapas vetoriais

tmap

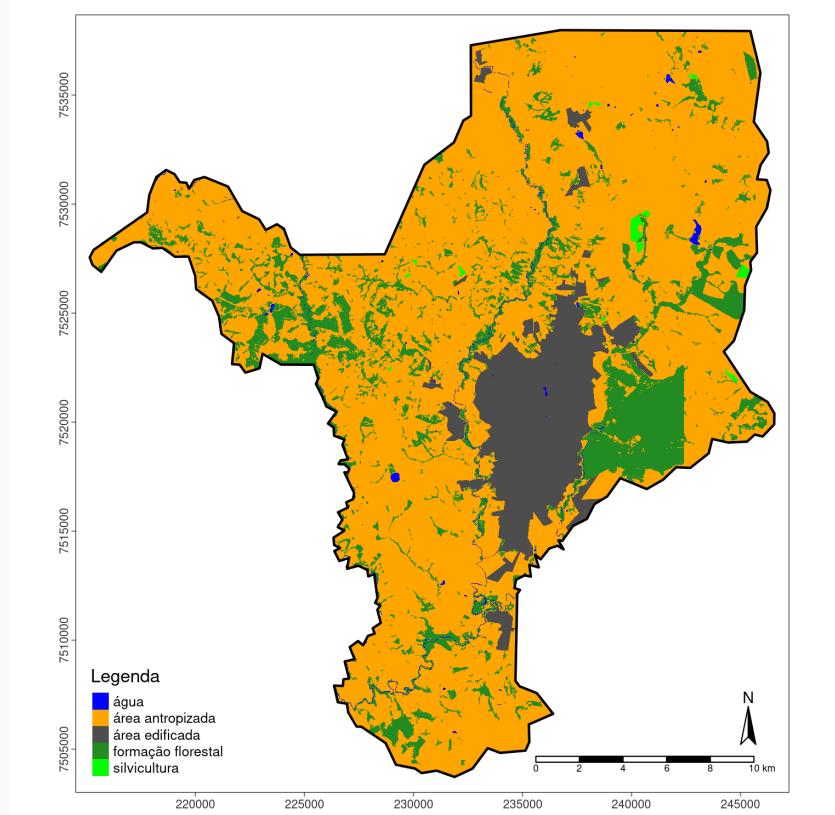
Adicionar barra de escala e norte

```
# land use and choose colors + rio claro limit fill + coords + themes + scalebar +
tm_shape(rc_use) +
  tm_fill(col = "CLASSE_USO",
          pal = c("blue", "orange", "gray30", "forestgreen", "green"), title = "Le
tm_shape(rc_2019_utm) +
  tm_borders(lwd = 2, col = "black") +
  tm_grid(lines = FALSE, labels.format = list(big.mark = "")), labels.rot = c(0, 90
tm_compass() +
  tm_scale_bar()
```

8.5 Mapas vetoriais

tmap

Adicionar barra de escala e norte



8.5 Mapas vetoriais

tmap

Add título, nome dos eixos e atribuir a um objeto

```
# land use and choose colors + rio claro limit fill + coords + themes + scalebar +
map_use_tmap ← tm_shape(rc_use) +
  tm_fill(col = "CLASSE_USO",
          pal = c("blue", "orange", "gray30", "forestgreen", "green"), title = "Le
tm_shape(rc_2019_utm) +
  tm_borders(lwd = 2, col = "black") +
  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(main.title = "Cobertura da terra Rio Claro/SP (2015)") )
map_use_tmap
```

8.5 Mapas vetoriais

tmap

Add título, nome dos eixos e atribuir a um objeto

8.5 Mapas vetoriais

tmap

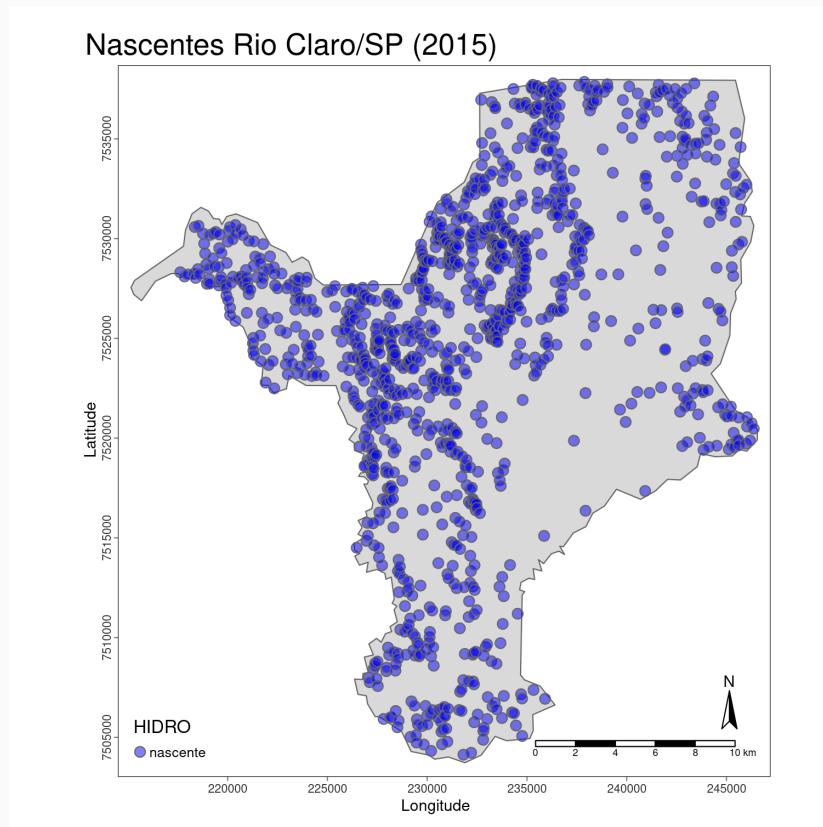
Nascentes

```
# springs
tm_shape(rc_2019_utm) +
  tm_polygons() +
  tm_shape(rc_spr) +
  tm_bubbles(col = "HIDRO", pal = "blue",
             size = .2, alpha = .5) +
  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(main.title = "Nascentes Rio Claro/SP (2015)")
```

8.5 Mapas vetoriais

tmap

Nascentes



8.5 Mapas vetoriais

tmap

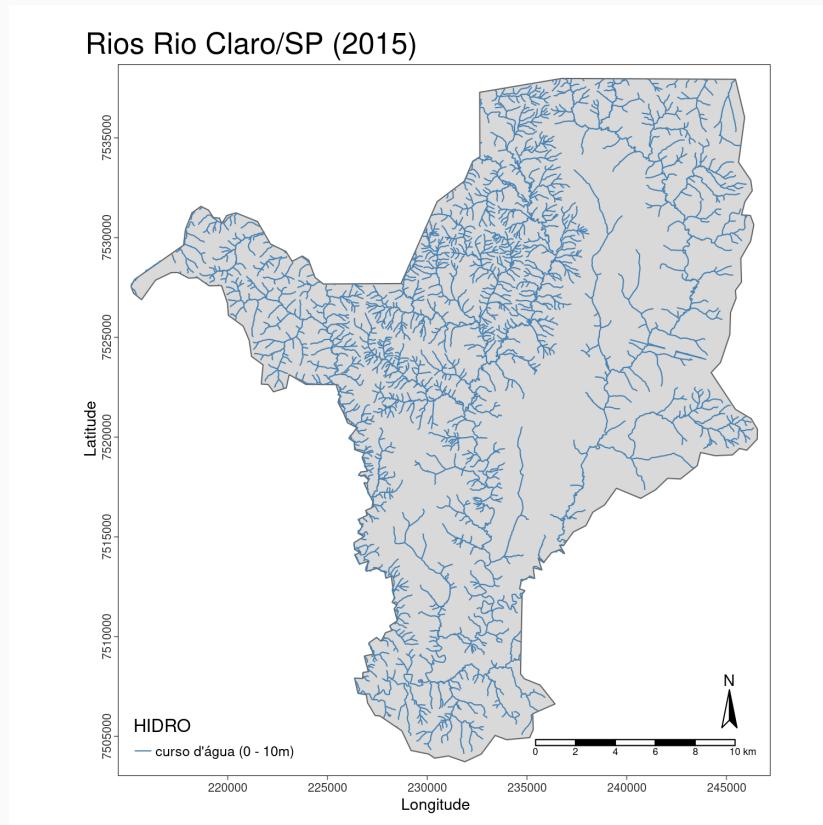
Rios

```
# rivers
tm_shape(rc_2019_utm) +
  tm_polygons() +
  tm_shape(rc_riv) +
  tm_lines(col = "HIDRO", pal = "steelblue") +
  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(main.title = "Rios Rio Claro/SP (2015)")
```

8.5 Mapas vetoriais

tmap

Rios

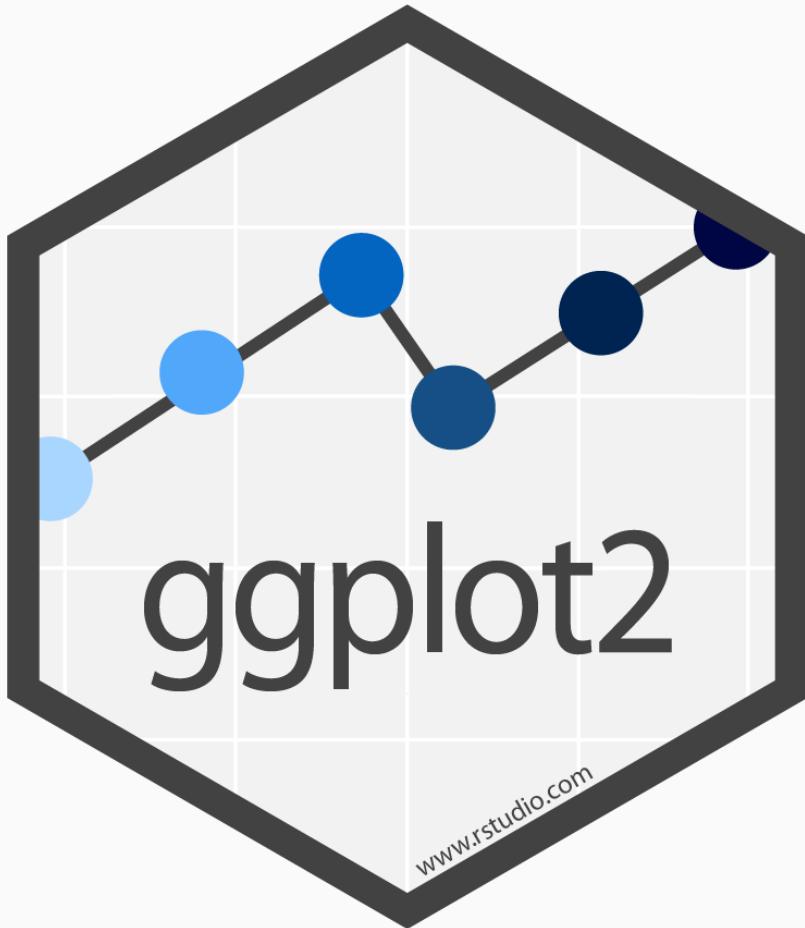


Mapas matriciais

8.6 Mapas matriciais

Elevação

```
# import elevation
elev <- raster::raster(here::here("03_dados", "raster", "srtm_27_17_rc.tif")) %>%
  raster::mask(rc_2019)
elev
plot(elev, col = viridis::viridis(10))
```



8.6 Mapas matriciais

ggplot2

Transformar o raster em um tibble

```
# raster to tibble
da_elev <- raster::rasterToPoints(elev) %>%
  tibble::as_tibble() %>%
  dplyr::rename(elev = srtm_27_17_rc)
head(da_elev)
```

```
## # A tibble: 6 x 3
##       x     y   elev
##   <dbl> <dbl> <dbl>
## 1 -47.6 -22.2   650
## 2 -47.6 -22.2   646
## 3 -47.6 -22.2   644
## 4 -47.6 -22.2   643
## 5 -47.6 -22.2   641
## 6 -47.6 -22.2   688
```

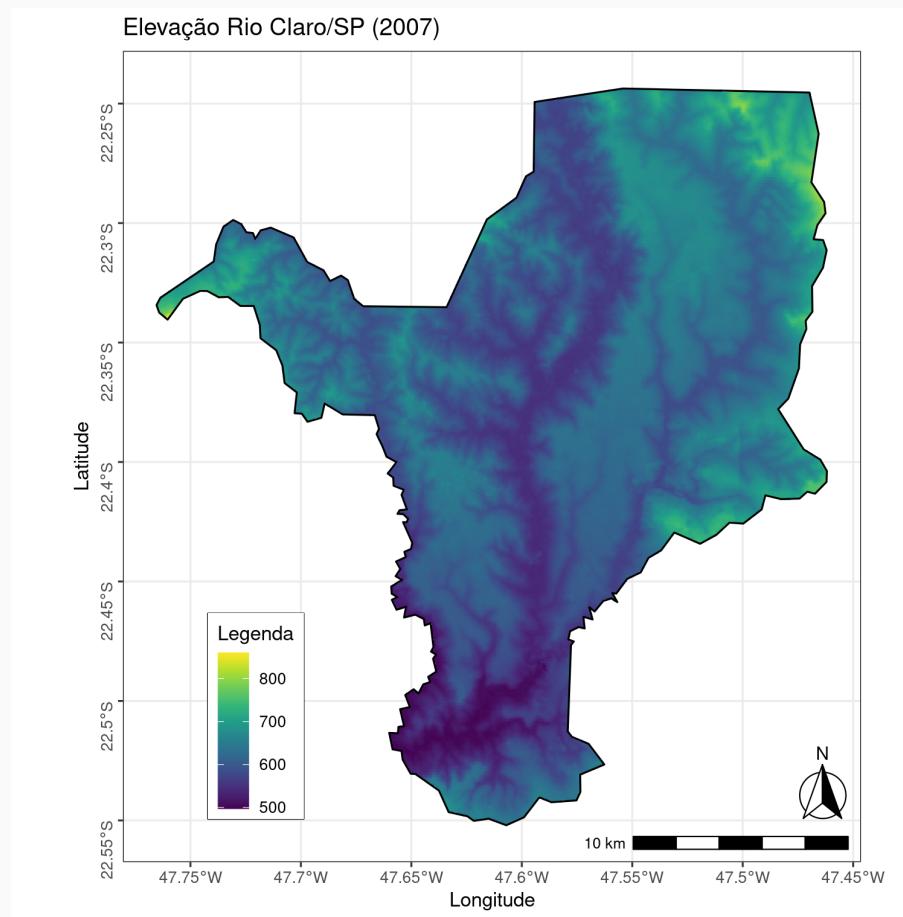
8.6 Mapas matriciais

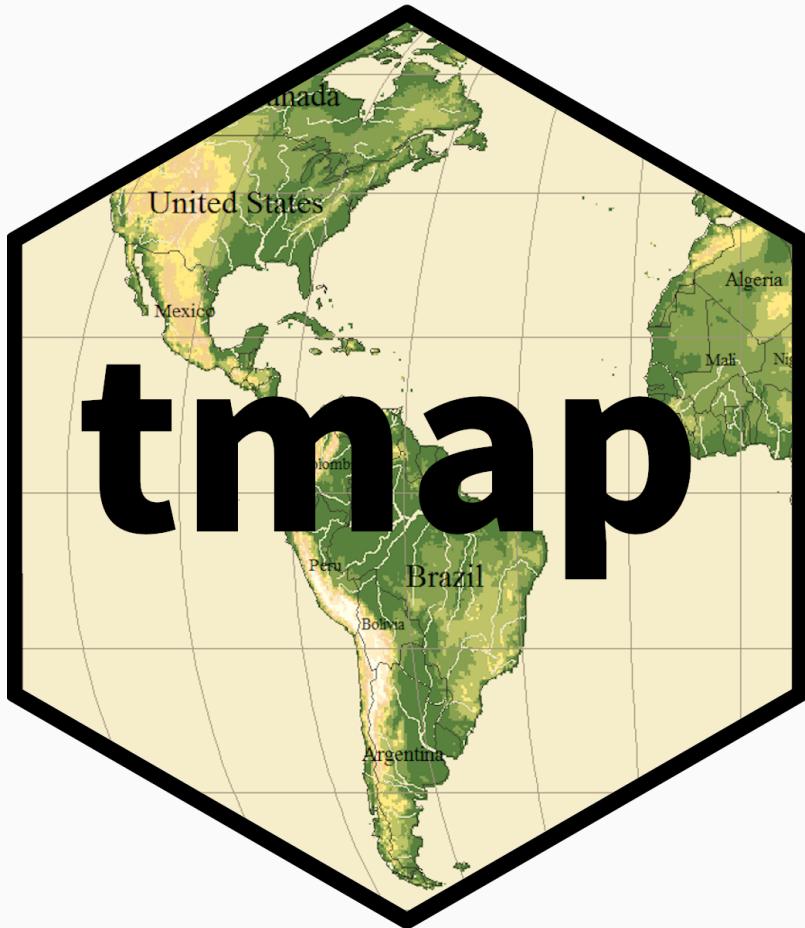
ggplot2

```
# elevation map
map_elev_gg ← ggplot() +
  geom_raster(data = da_elev, aes(x = x, y = y, fill = elev)) +
  geom_sf(data = rc_2019, color = "black", fill = NA) +
  scale_fill_gradientn(colors = viridis::viridis(10)) +
  theme_bw() +
  annotation_scale(location = "br", width_hint = .3) +
  annotation_north_arrow(location = "br", which_north = "true",
                         pad_x = unit(0, "cm"), pad_y = unit(.7, "cm"),
                         style = north_arrow_fancy_orienteering) +
  labs(x = "Longitude", y = "Latitude", title = "Elevação Rio Claro/SP (2007)", fi
theme(legend.position = c(.18,.18),
      legend.box.background = element_rect(colour = "black"),
      axis.text.y = element_text(angle = 90))
map_elev_gg
```

8.6 Mapas matriciais

ggplot2





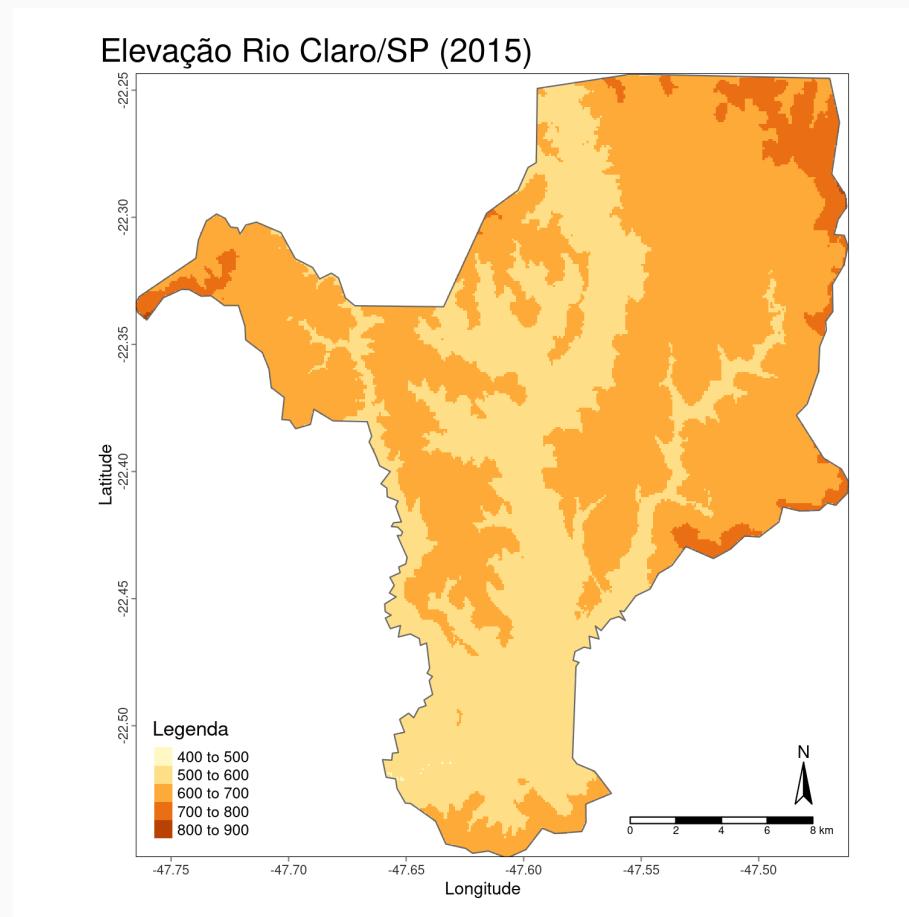
8.6 Mapas matriciais

tmap

```
# elevation map
map_elev_tmap ← tm_shape(elev) +
  tm_raster(title = "Legenda") +
  tm_shape(rc_2019) +
  tm_borders() +
  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 (2015)")
map_elev_tmap
```

8.6 Mapas matriciais

tmap



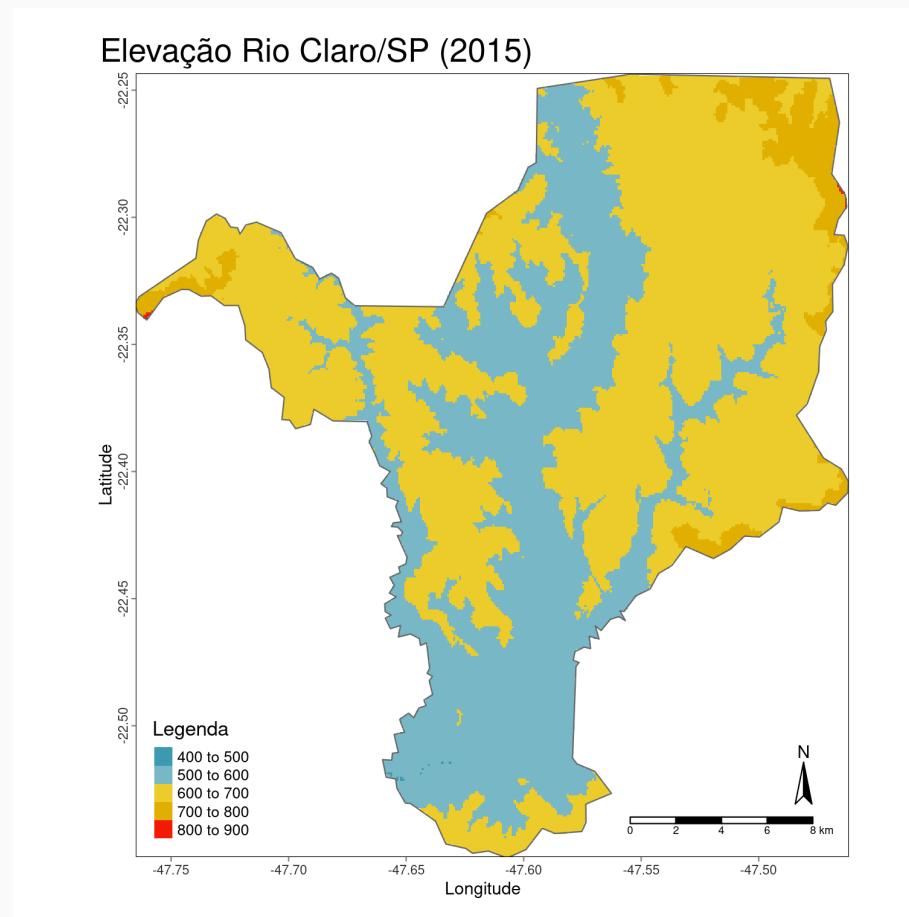
8.6 Mapas matriciais

tmap

```
# elevation map
map_elev_tmap ← tm_shape(elev) +
  tm_raster(pal = wesanderson::wes_palette("Zissou1"), title = "Legenda") +
  tm_shape(rc_2019) +
  tm_borders() +
  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 (2015)")
map_elev_tmap
```

8.6 Mapas matriciais

tmap



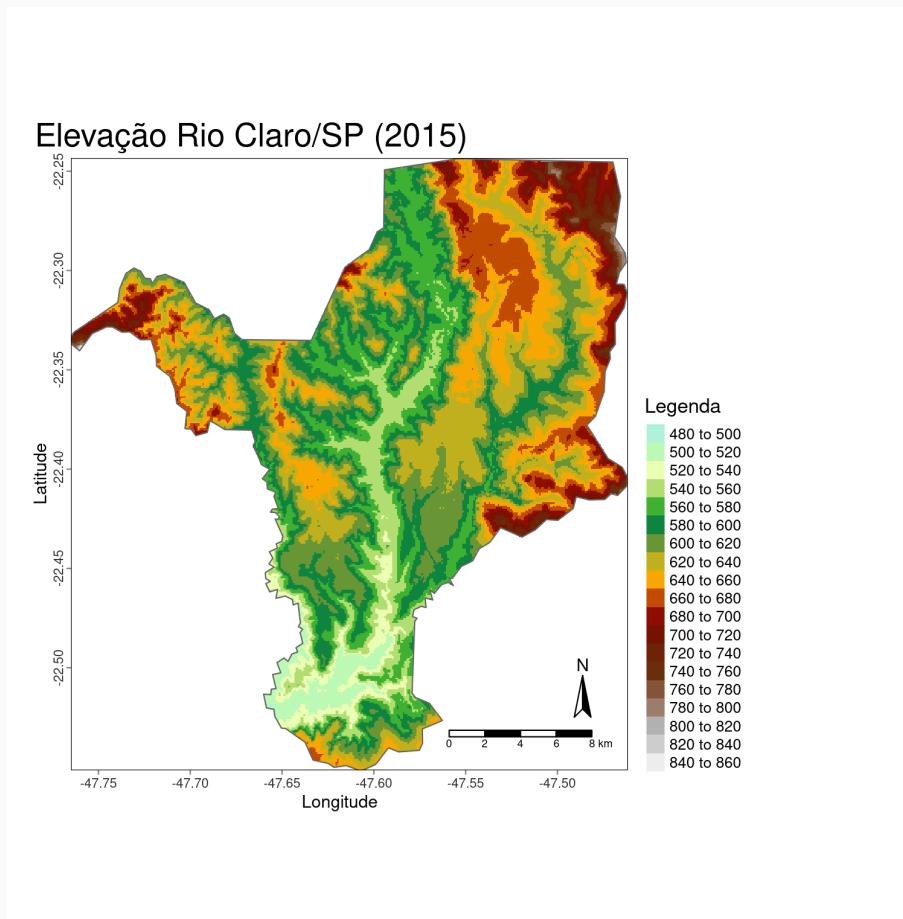
8.6 Mapas matriciais

tmap

```
# elevation map
map_elev_tmap ← tm_shape(elev) +
  tm_raster(pal = cptcity::cpt(pal = "gmt_GMT_dem4"),
             n = 20,
             title = "Legenda") +
  tm_shape(rc_2019) +
  tm_borders() +
  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"),
            legend.outside = TRUE,
            main.title = "Elevação Rio Claro/SP (2015)")
map_elev_tmap
```

8.6 Mapas matriciais

tmap



8.7 Mapas estáticos

Foram os mapas que vimos até o momento

Existe duas formas de mapas não estáticos:
animados e interativos

Mapas animados são utilizados para entender a
mudanças espacial no tempo

Mapas interativos são utilizados quando
precisamos interagir com algo

8.8 Mapas animados

Mudança dos estados do Brasil ao longo do tempo

```
# download
year ← geobr::list_geobr()[3, 3] %>%
  dplyr::pull() %>%
  stringr::str_split(", ", simplify = TRUE) %>%
  as.character()
year

br ← NULL

for(i in year){

  br ← geobr::read_state("all", i) %>%
    dplyr::mutate(year = i) %>%
    dplyr::bind_rows(br, .)

}

br$year %>% table
```

8.8 Mapas animados

Mudança dos estados do Brasil ao longo do tempo

```
# create facet  
br_years ← tm_shape(br) +  
  tm_polygons() +  
  tm_facets(along = "year", free.coords = FALSE)  
  
# export  
tmap_animation(br_years, filename = here::here("03_dados", "mapas", "geo_br_years.
```



8.9 Mapas interativos

Mapas anteriores

8.9 Mapas interativos

Mapas anteriores

8.10 Exportar mapas

ggplot2

Exportar

```
# export
ggsave(map_use_gg,
       filename = "map_rio_claro_land_use_gg.png",
       path = here::here("03_dados", "mapas"),
       width = 20,
       height = 20,
       units = "cm",
       dpi = 300)
```

8.10 Exportar mapas

ggplot2

Exportar

```
# export
ggsave(map_elev_gg,
       filename = "map_rio_claro_elevation_gg.png",
       path = here::here("03_dados", "mapas"),
       width = 20,
       height = 20,
       units = "cm",
       dpi = 300)
```

8.10 Exportar mapas

tmap

Exportar

```
# export
tmap_save(map_use_tmap,
  filename = here::here("03_dados", "mapas", "map_rio_claro_land_use_tmap.
width = 20,
height = 20,
units = "cm",
dpi = 300)
```

8.10 Exportar mapas

tmap

Exportar

```
# export
tmap_save(map_elev_tmap,
  filename = here::here("03_dados", "mapas", "map_rio_claro_elevation_tmap",
  width = 20,
  height = 20,
  units = "cm",
  dpi = 300)
```

Gostaria de agradecer a 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](#)