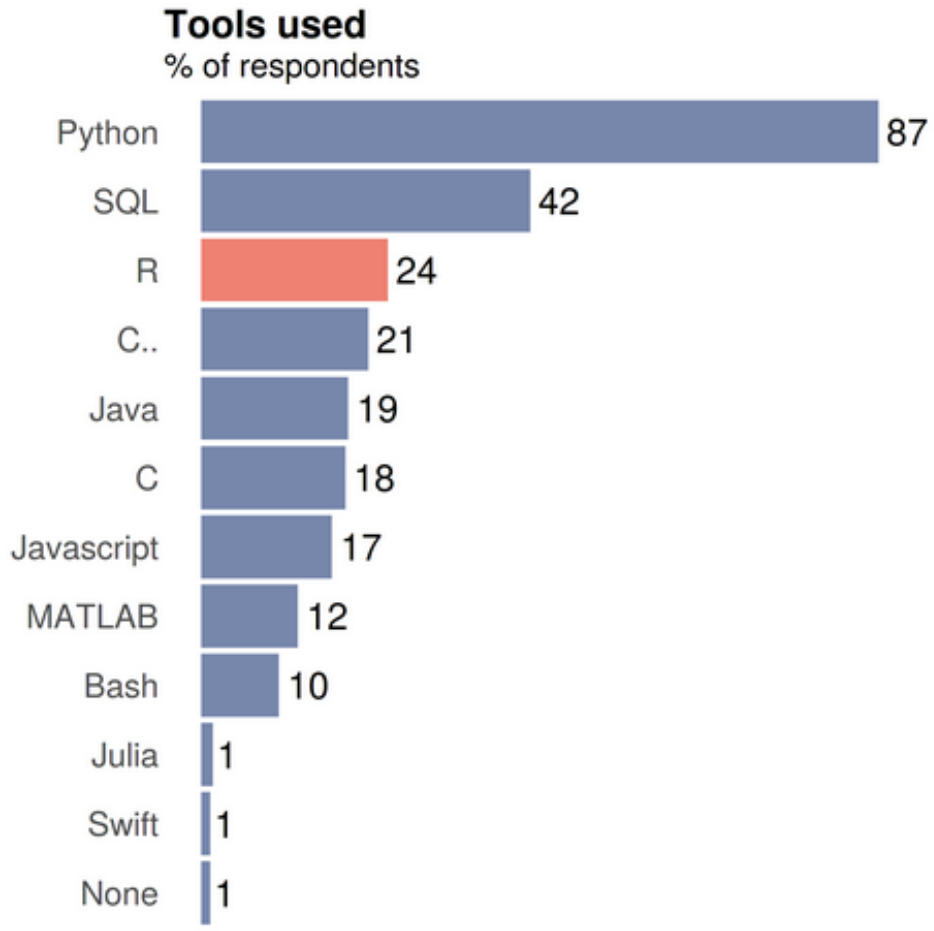
Why R

Francisco Rodríguez-Sánchez

https://frodriguezsanchez.net

@frod_san

R: dominant language in data science

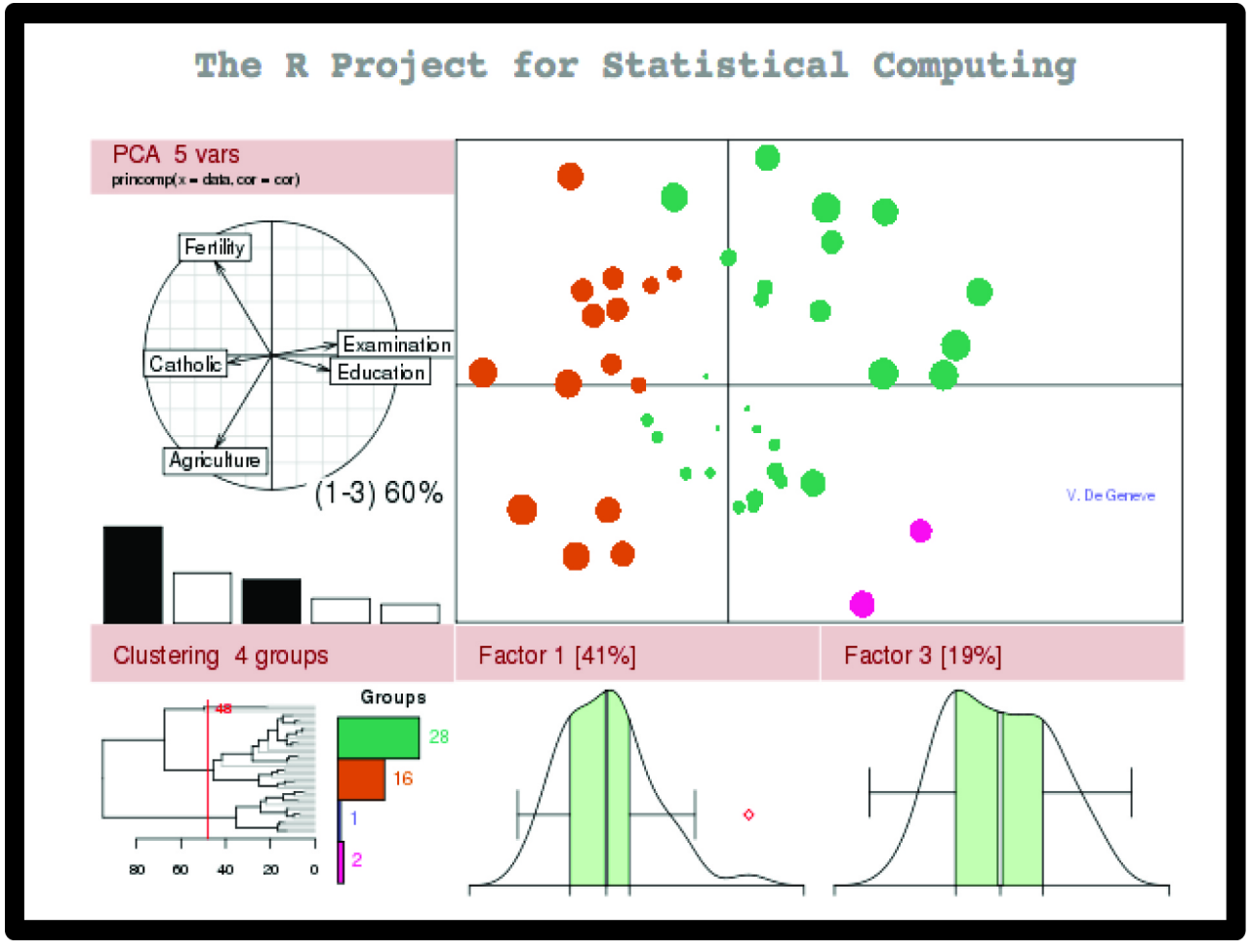


- Data manipulation
- Data visualisation
- Modelling

Why R

- Free, open-source, cross-platform
- Not just stats package, but a programming language
- Can do many things beyond stats (e.g. scrape web data, GIS, etc)
- +20,000 packages extending functionality
- Flexible, powerful
- Can easily connect to other languages (e.g. Python, C++)
- High-quality graphics
- Large, helpful community (forums, StackOverflow, Twitter)

R: Not only for stats



Some cool things you can do with R

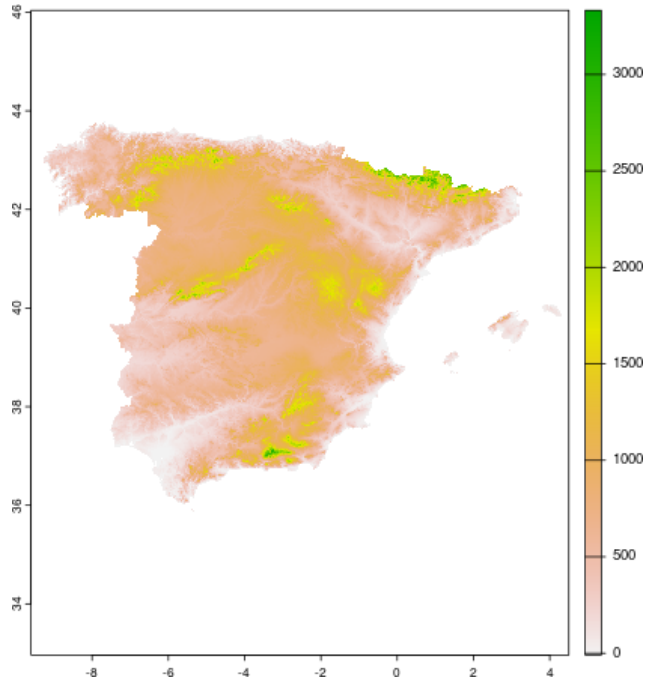
(besides cutting-edge stats)

Where is Rio de Janeiro?

```
library(tmaptools)
rio <- geocode_OSM("Rio de Janeiro", as.sf = TRUE)

library(ggplot2)
ggplot() +
  borders() +
  geom_sf(data = rio, size = 4)
```

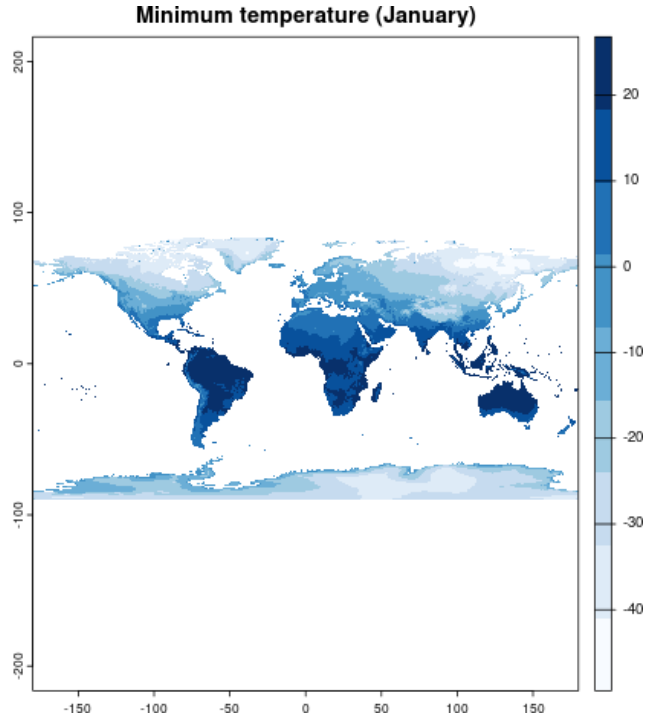
Get elevation anywhere in the world

```
library(geodata)
elev.sp <- elevation_30s(country = "Spain", path = "tmp")
plot(elev.sp)
```



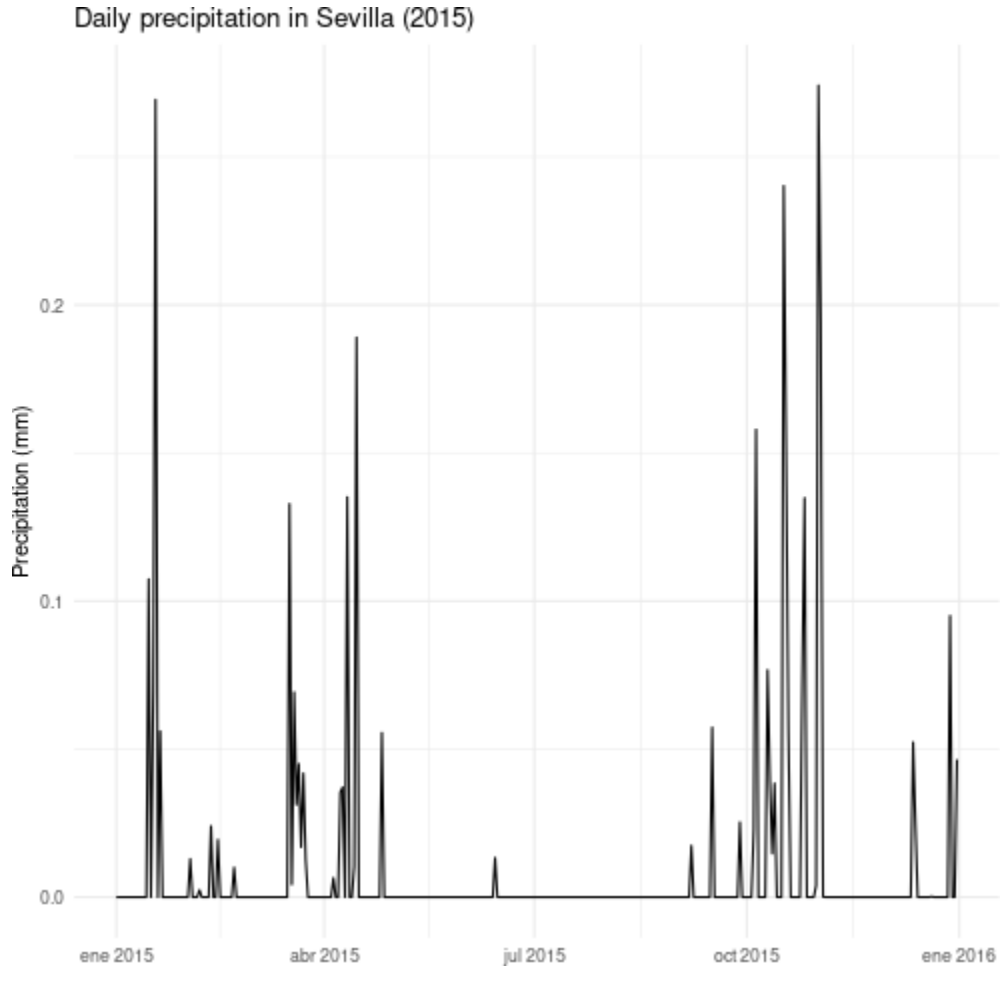
Get climate data from the web

```
tmin <- worldclim_global(var = "tmin", res = 10, path = "tmp")
plot(tmin, 1, col = brewer.pal(9, "Blues"))
```



Get climate data from the web

```
sevilla.coords <- geocode_OSM("Sevilla, Spain",
                              as.sf = TRUE)

sevilla.precip.2015 <- get_daily_climate(
  coords = sevilla.coords,
  climatic_var = "Prcp",
  period = 2015)

ggplot(sevilla.precip.2015) +
  geom_line(aes(x = as.Date(date), y = Prcp/100)) +
  labs(x = "", y = "Precipitation (mm)",
       title = "Daily precipitation in Sevilla (2015)") +
  theme_minimal()
```

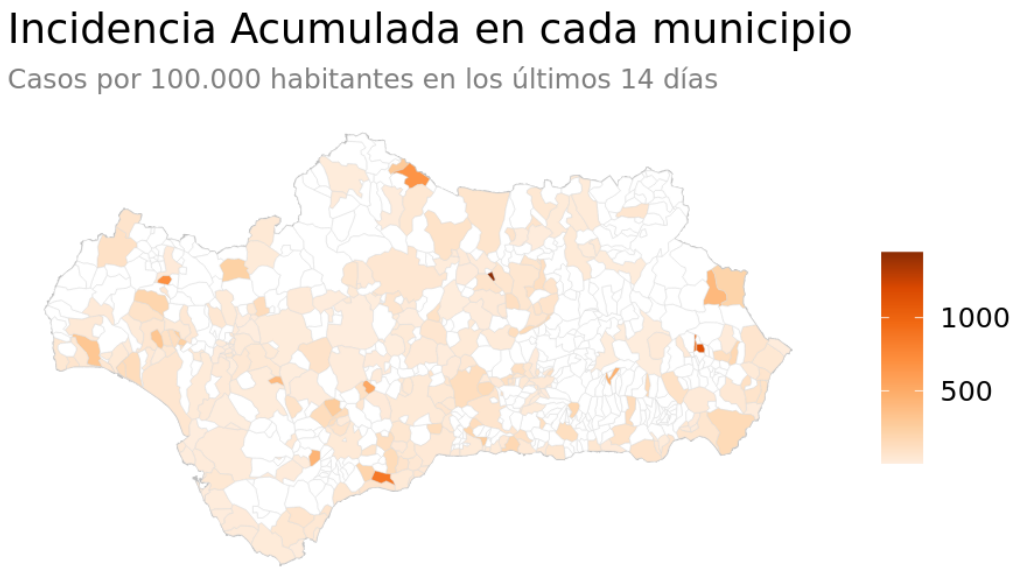


Create websites, slides, reports, books

Run dynamic website

Datos COVID-19 Andalucía: evolución y situación actual

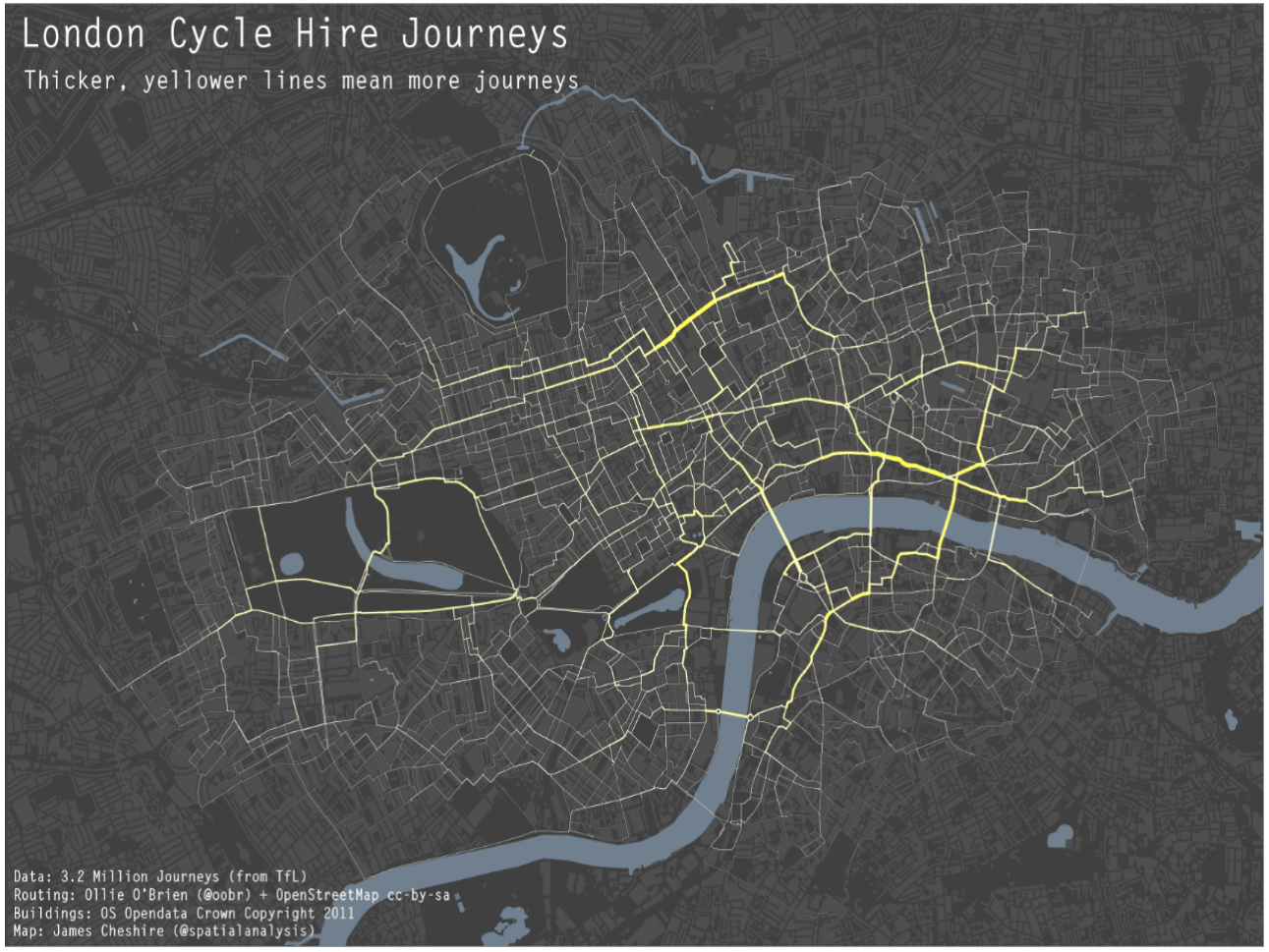
Datos diarios de casos, ingresos y defunciones por rango de edad, municipio, provincia y comunidad autónoma.



https://tiny.cc/COVID19-Andalucia

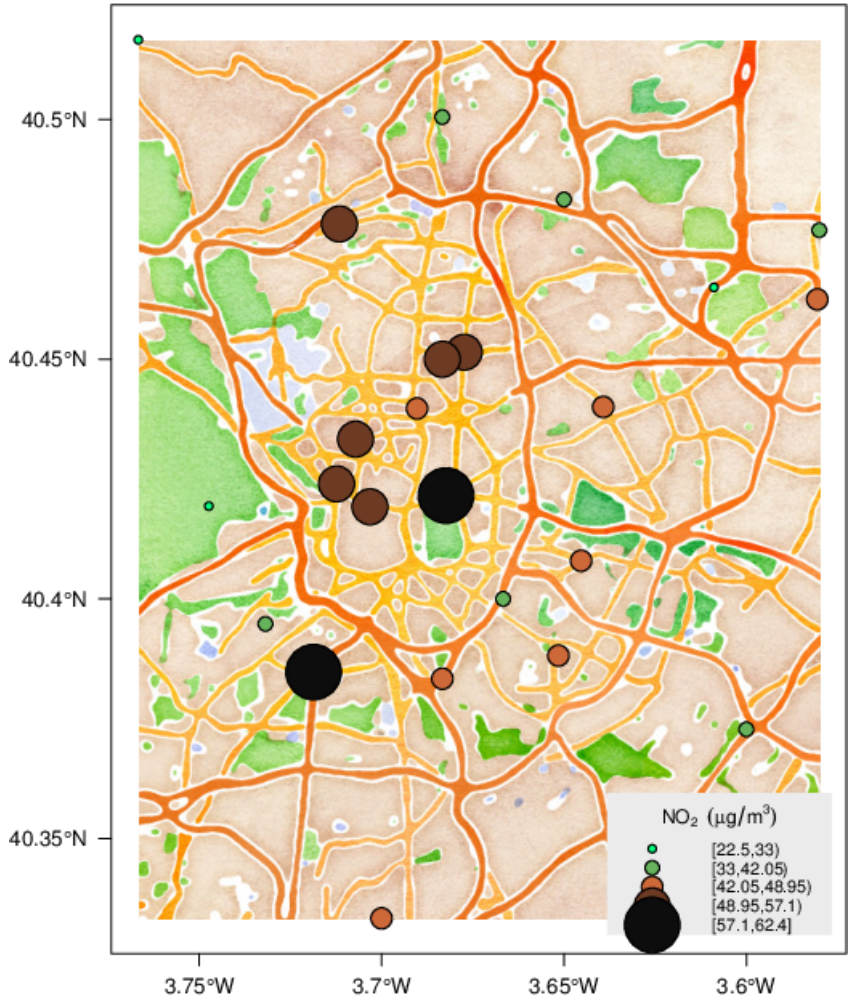
R can make beautiful maps

Made in R



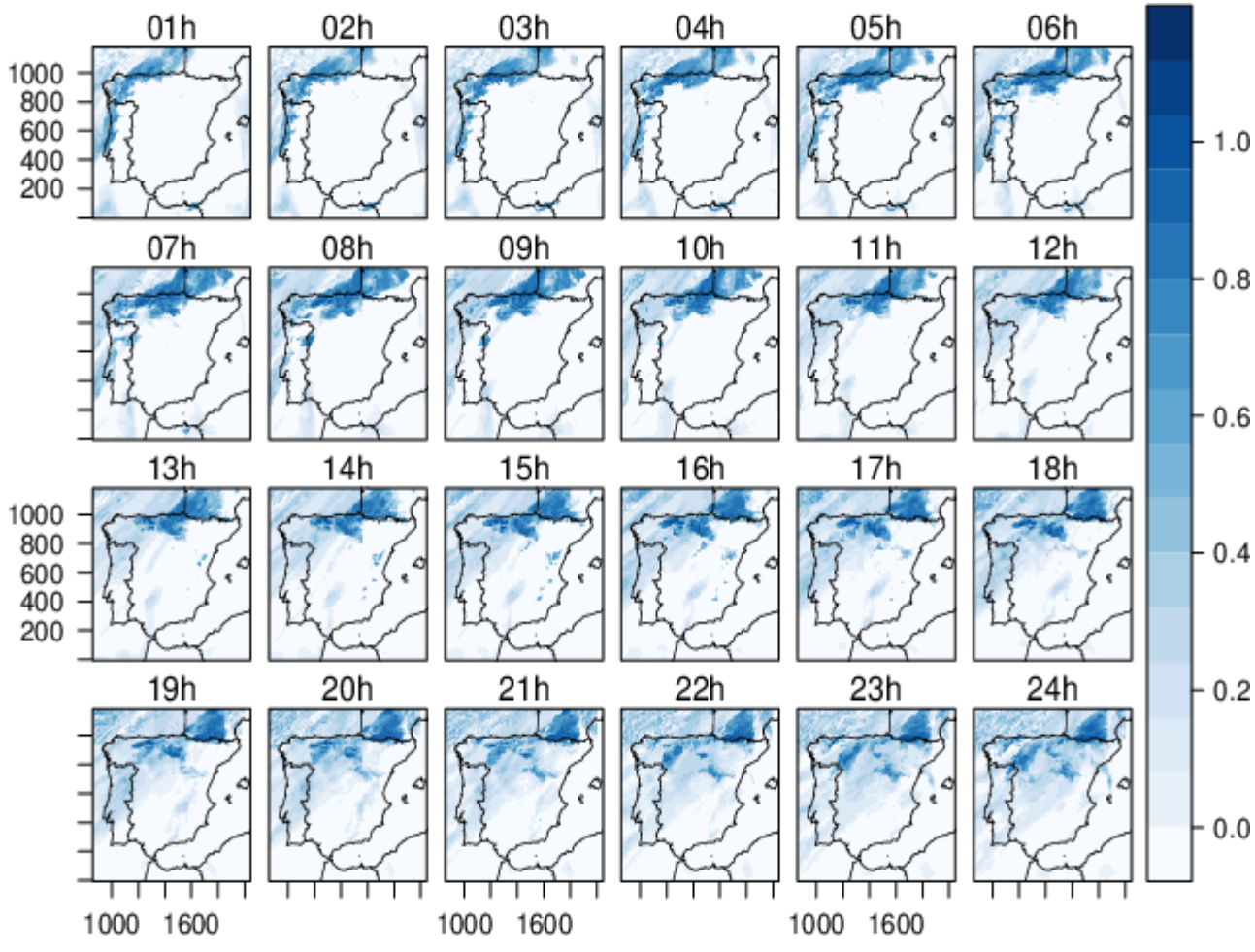
http://spatial.ly/2012/02/great-maps-ggplot2/

Made in R



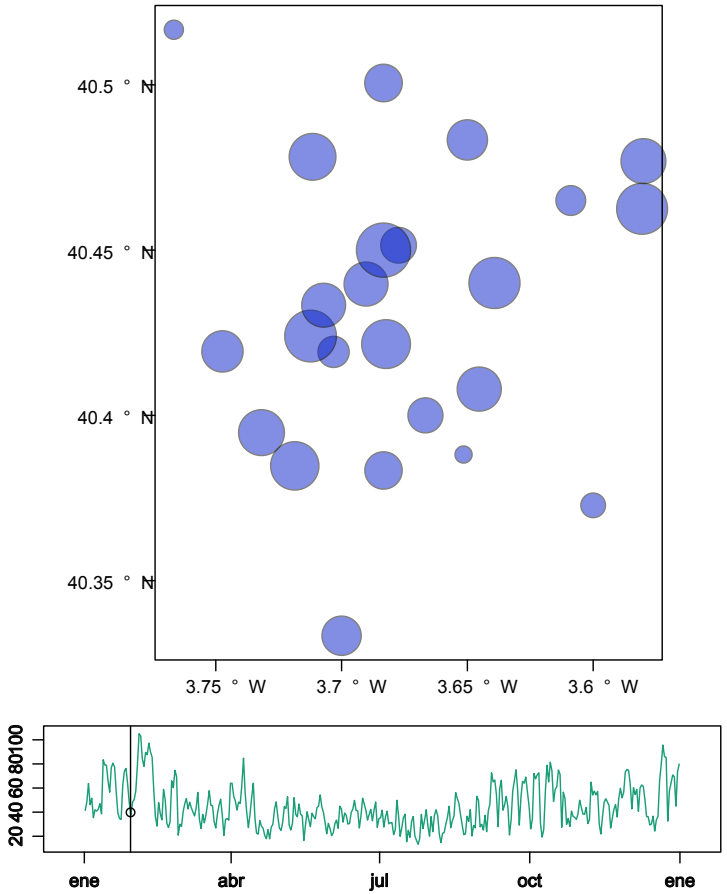
http://oscarperpinan.github.io/spacetime-vis/

Made in R



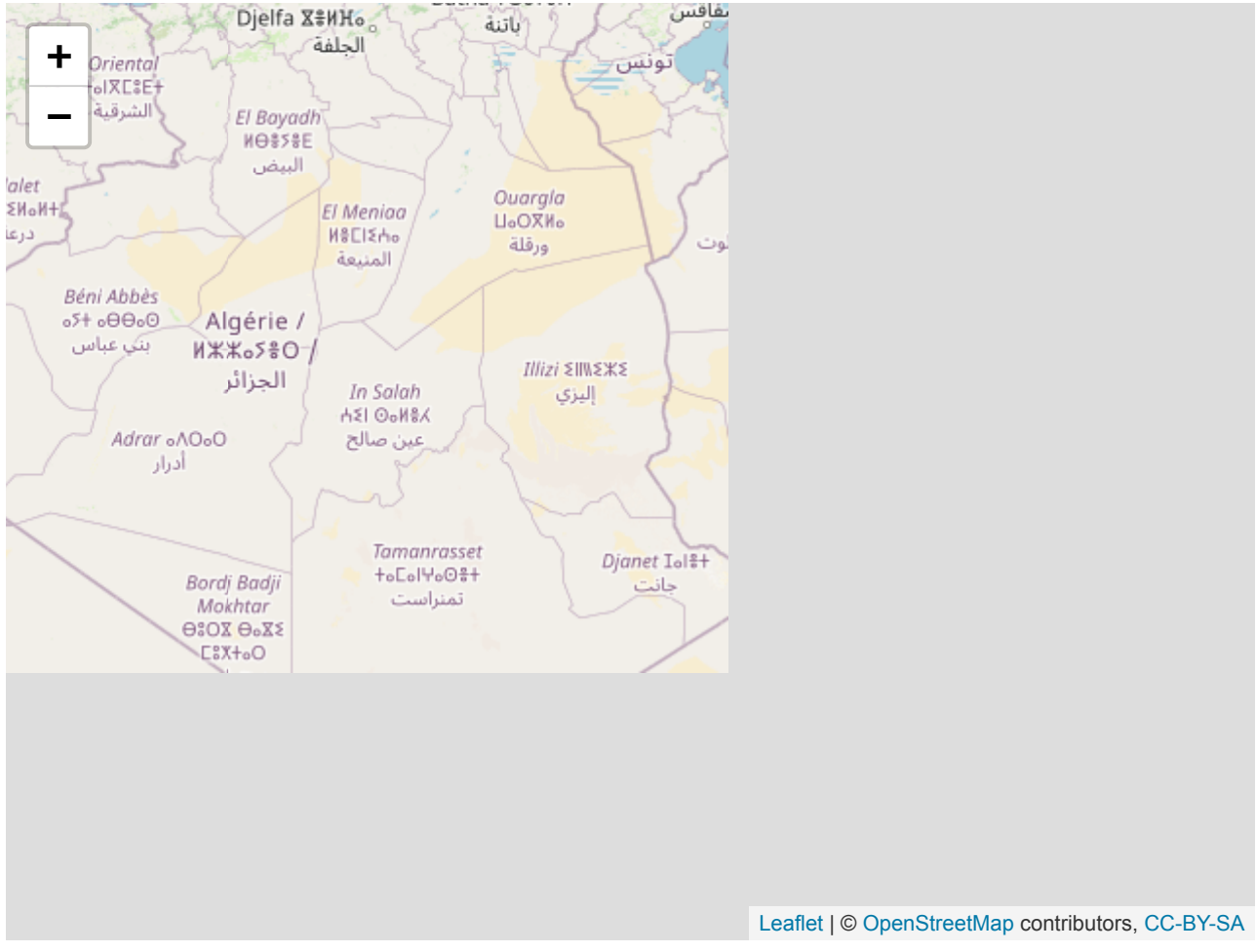
http://oscarperpinan.github.io/spacetime-vis/

Made in R



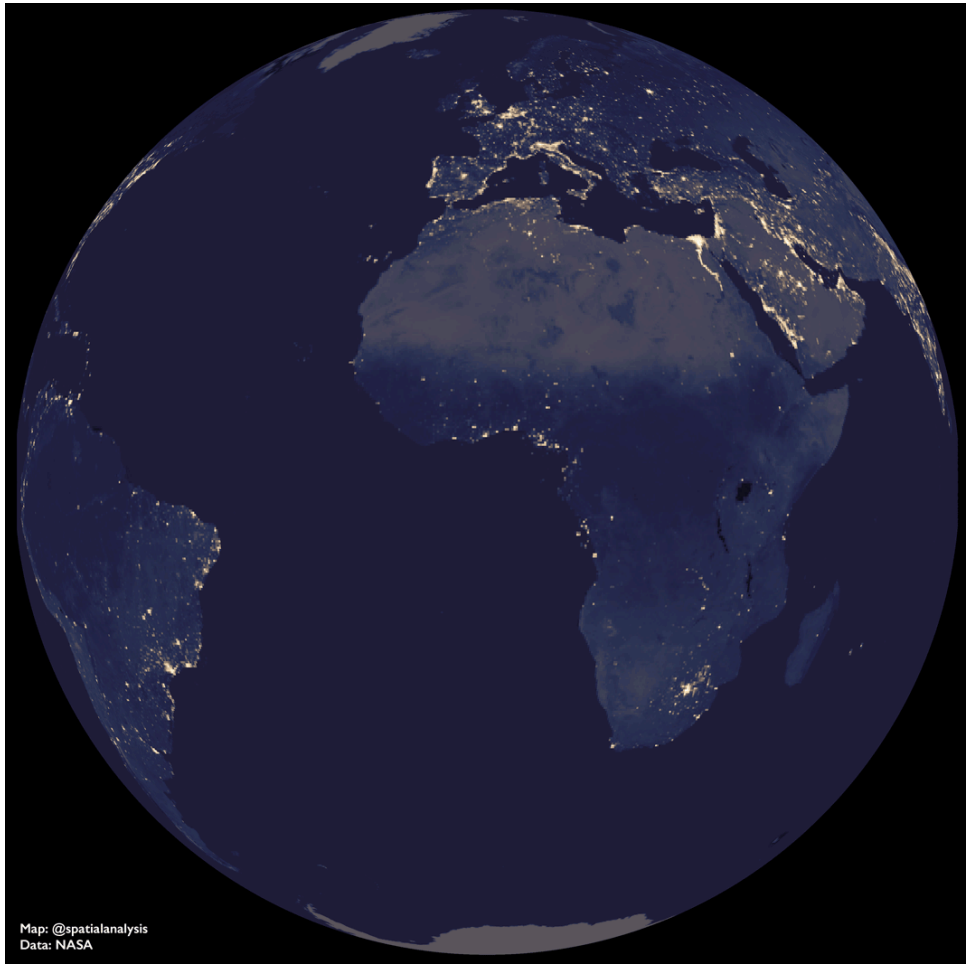
http://oscarperpinan.github.io/spacetime-vis/

Made in R



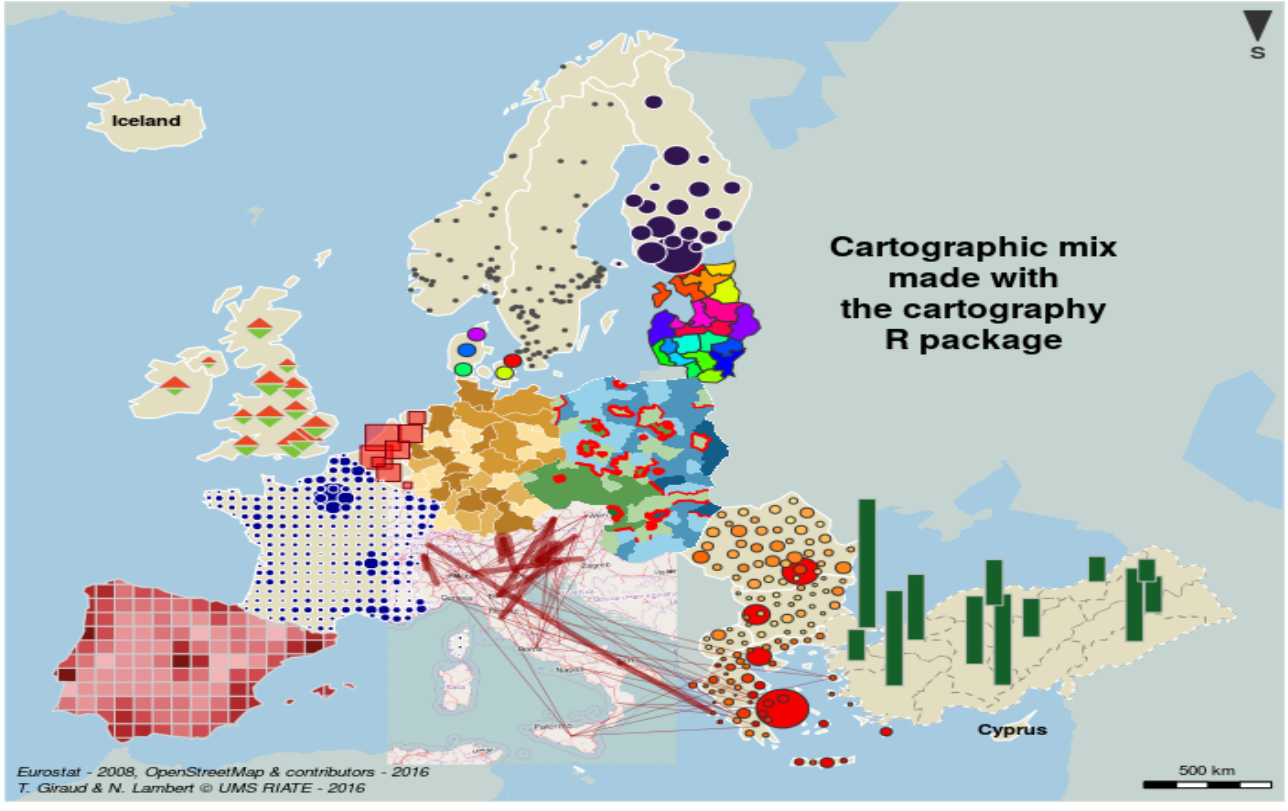
https://rstudio.github.io/leaflet/

Made in R



http://spatial.ly/2017/05/spinning-globes-with-r/

Made in R



https://cran.r-project.org/package=cartography

Made in R

https://cran.r-project.org/package=tmap

Made in R

https://github.com/tylermorganwall/rayshader

Made in R

R can make beautiful maps

And beautiful stats too

END

Slides and source code available at https://github.com/Pakillo

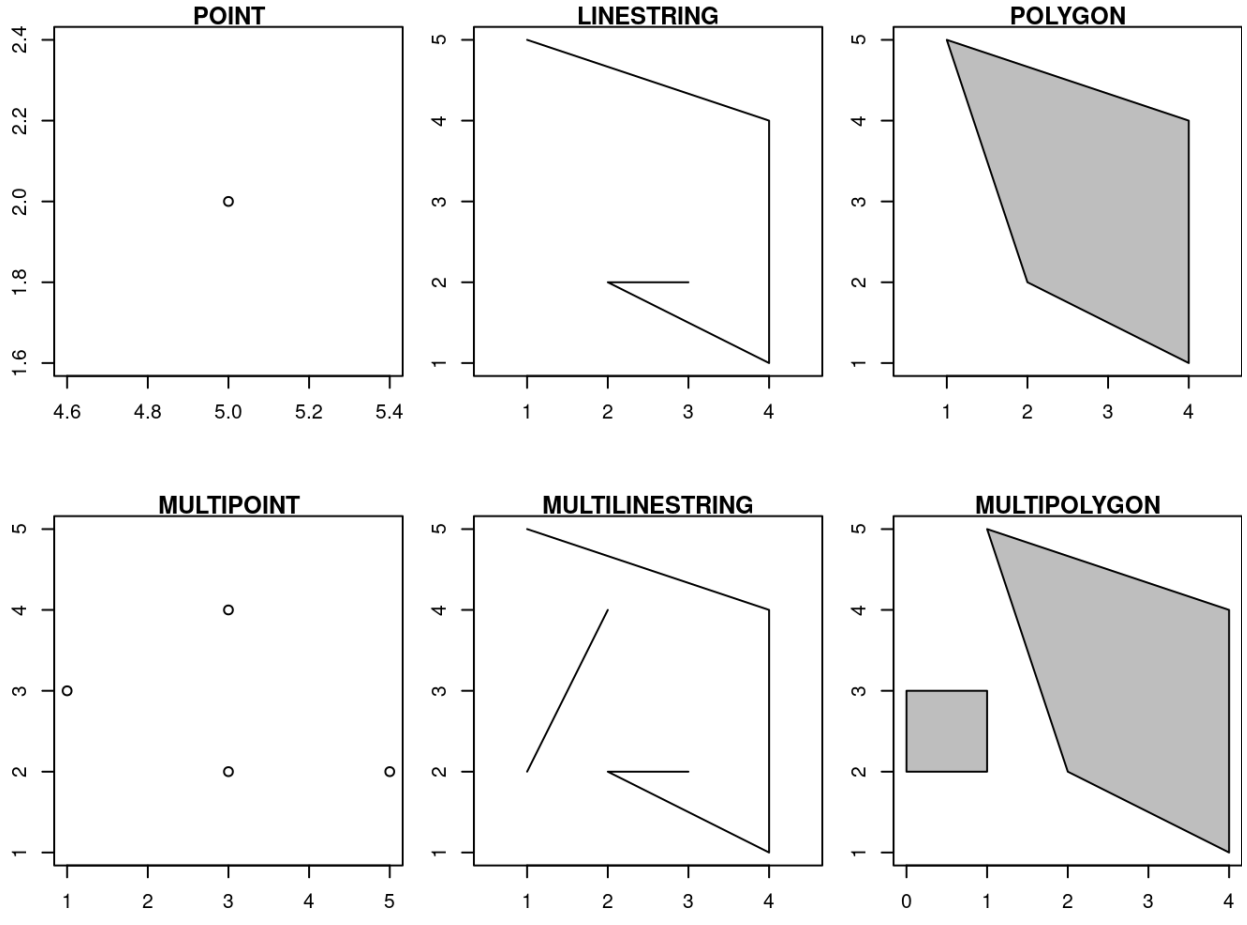
Spatial vector data (sf)

Francisco Rodríguez-Sánchez

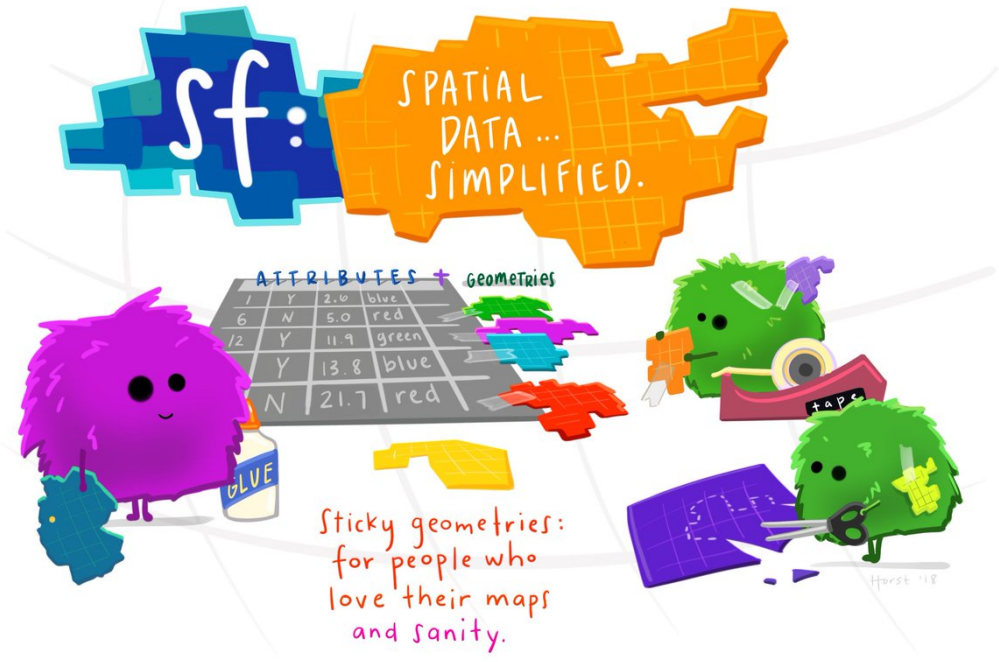
https://frodriguezsanchez.net

@frod_san

Vectorial spatial data (simple features)



The sf (simple features) package



https://r-spatial.github.io/sf/

Read COVID data

```
library(dplyr)
covid <- readr::read_csv("data/covid.csv")
muni.coords <- readr::read_csv("data/coords_towns.csv")
covid <- left_join(covid, muni.coords, by = "Municipio")
covid
```

```
# A tibble: 778 × 7
   Provincia Municipio                Poblacion Casos Fallecidos      x       y
   <chr>     <chr>                        <dbl> <dbl>      <dbl>  <dbl>   <dbl>
 1 Cádiz     Algeciras                   123078 11945        264 279223 4001392
 2 Cádiz     Barbate                      22556  2369         27 237228 4009013
 3 Cádiz     Tarifa                       18183  1542         30 265096 3988986
 4 Cádiz     San Roque                    31571  3000         50 285306 4010004
 5 Cádiz     Barrios (Los)                23777  2164         80 276026 4007210
 6 Cádiz     Conil de la Frontera         22775  2243         10 222693 4019422
 7 Cádiz     Castellar de la Frontera      3057   300          4 282718 4018348
 8 Cádiz     Benalup-Casas Viejas          6986   664          6 247412 4025926
 9 Málaga    Manilva                      16439  1540         22 298249 4027913
10 Cádiz     Vejer de la Frontera         12622  1051         10 233423 4015856
# … with 768 more rows
```

Making a data frame *spatial*

```
library(sf)
covid.sf <- st_as_sf(covid)
```

```
Error in st_sf(x, ..., agr = agr, sf_column_name = sf_column_name): no simple features geometry column pr
```

Making a data frame *spatial*

```
covid.sf <- st_as_sf(covid, coords = c("x", "y"))
covid.sf
```

```
Simple feature collection with 778 features and 5 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 104973 ymin: 3988986 xmax: 610819 ymax: 4273703
CRS:           NA
# A tibble: 778 × 6
   Provincia Municipio            Pobla…¹ Casos Falle…² geometry
   <chr>     <chr>                 <dbl> <dbl>   <dbl> <POINT>
 1 Cádiz     Algeciras            123078 11945     264 (279223 4001392)
 2 Cádiz     Barbate               22556  2369      27 (237228 4009013)
 3 Cádiz     Tarifa                18183  1542      30 (265096 3988986)
 4 Cádiz     San Roque             31571  3000      50 (285306 4010004)
 5 Cádiz     Barrios (Los)         23777  2164      80 (276026 4007210)
 6 Cádiz     Conil de la Fronte…   22775  2243      10 (222693 4019422)
 7 Cádiz     Castellar de la Fr…    3057   300       4 (282718 4018348)
 8 Cádiz     Benalup-Casas Viej…    6986   664       6 (247412 4025926)
 9 Málaga    Manilva               16439  1540      22 (298249 4027913)
10 Cádiz     Vejer de la Fronte…   12622  1051      10 (233423 4015856)
# … with 768 more rows, and abbreviated variable names ¹Poblacion, ²Fallecidos
```

Must specify Coordinate Reference System (CRS)

```
covid.sf
```

```
Simple feature collection with 778 features and 5 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 104973 ymin: 3988986 xmax: 610819 ymax: 4273703
CRS:           NA
# A tibble: 778 × 6
   Provincia Municipio           Pobla…¹ Casos Falle…²          geometry
   <chr>     <chr>                 <dbl> <dbl>   <dbl>       <POINT>
 1 Cádiz     Algeciras            123078 11945     264 (279223 4001392)
 2 Cádiz     Barbate               22556  2369      27 (237228 4009013)
 3 Cádiz     Tarifa                18183  1542      30 (265096 3988986)
 4 Cádiz     San Roque             31571  3000      50 (285306 4010004)
 5 Cádiz     Barrios (Los)         23777  2164      80 (276026 4007210)
 6 Cádiz     Conil de la Fronte…   22775  2243      10 (222693 4019422)
 7 Cádiz     Castellar de la Fr…    3057   300       4 (282718 4018348)
 8 Cádiz     Benalup-Casas Viej…    6986   664       6 (247412 4025926)
 9 Málaga    Manilva               16439  1540      22 (298249 4027913)
10 Cádiz     Vejer de la Fronte…   12622  1051      10 (233423 4015856)
# … with 768 more rows, and abbreviated variable names ¹Poblacion, ²Fallecidos
```

Setting the Coordinate Reference System (CRS)

Search EPSG, e.g. at https://spatialreference.org/

For UTM 30N datum ETR89, EPSG = 25830

```
covid.sf <- st_set_crs(covid.sf, value = 25830)
covid.sf
```

```
Simple feature collection with 778 features and 5 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 104973 ymin: 3988986 xmax: 610819 ymax: 4273703
Projected CRS: ETRS89 / UTM zone 30N
# A tibble: 778 × 6
   Provincia Municipio           Pobla…¹ Casos Falle…²         geometry
 * <chr>     <chr>                 <dbl> <dbl>   <dbl>      <POINT [m]>
 1 Cádiz     Algeciras            123078 11945     264 (279223 4001392)
 2 Cádiz     Barbate               22556  2369      27 (237228 4009013)
 3 Cádiz     Tarifa                18183  1542      30 (265096 3988986)
 4 Cádiz     San Roque             31571  3000      50 (285306 4010004)
 5 Cádiz     Barrios (Los)         23777  2164      80 (276026 4007210)
 6 Cádiz     Conil de la Fronte…   22775  2243      10 (222693 4019422)
 7 Cádiz     Castellar de la Fr…    3057   300       4 (282718 4018348)
 8 Cádiz     Benalup-Casas Viej…    6986   664       6 (247412 4025926)
```

Common CRS I use

- **4326** (lonlat, datum WGS84)
- **3035** (Lambert Azimuthal Equal Area, datum ETRS89)
- **25830** (UTM 30N, datum ETRS89)
- **23030** (UTM 30N, datum ED50)

Change projection

From UTM 30N (EPSG = 25830) to lonlat (EPSG = 4326)

```
covid.geo <- st_transform(covid.sf, crs = 4326)
covid.geo
```

```
Simple feature collection with 778 features and 5 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: -7.466321 ymin: 36.01708 xmax: -1.747812 ymax: 38.59634
Geodetic CRS:  WGS 84
# A tibble: 778 × 6
   Provincia Municipio           Pobla…¹ Casos Falle…²             geometry
 * <chr>     <chr>                 <dbl> <dbl>   <dbl>          <POINT [°]>
 1 Cádiz     Algeciras            123078 11945     264 (-5.453473 36.13213)
 2 Cádiz     Barbate               22556  2369      27 (-5.922205 36.19029)
 3 Cádiz     Tarifa                18183  1542      30 (-5.606631 36.01708)
 4 Cádiz     San Roque             31571  3000      50 (-5.388281 36.21107)
 5 Cádiz     Barrios (Los)         23777  2164      80  (-5.490629 36.1838)
 6 Cádiz     Conil de la Fronte…   22775  2243      10 (-6.087324 36.27994)
 7 Cádiz     Castellar de la Fr…    3057   300       4 (-5.419364 36.28565)
 8 Cádiz     Benalup-Casas Viej…    6986   664       6 (-5.814541 36.34529)
 9 Málaga    Manilva               16439  1540      22 (-5.249033 36.37519)
10 Cádiz     Vejer de la Fronte…   12622  1051      10 (-5.966796 36.25086)
```

Retrieve coordinates

```
st_coordinates(covid.geo)
```

```
            X        Y
1   -5.453473 36.13213
2   -5.922205 36.19029
3   -5.606631 36.01708
4   -5.388281 36.21107
5   -5.490629 36.18380
6   -6.087324 36.27994
7   -5.419364 36.28565
8   -5.814541 36.34529
9   -5.249033 36.37519
10  -5.966796 36.25086
11  -6.152125 36.41611
12  -5.271542 36.44469
13  -6.204396 36.45958
14  -5.156160 36.42545
15  -5.450567 36.43299
16  -4.905783 36.51512
17  -5.934385 36.46466
18  -5.865098 36.52304
19  -5.319097 36.51911
```

Manipulating sf objects

sf are data frames with 'geometry' column

```
covid.sf
```

```
Simple feature collection with 778 features and 5 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 104973 ymin: 3988986 xmax: 610819 ymax: 4273703
Projected CRS: ETRS89 / UTM zone 30N
# A tibble: 778 × 6
   Provincia Municipio           Pobla…¹ Casos Falle…²         geometry
 * <chr>     <chr>                 <dbl> <dbl>   <dbl>      <POINT [m]>
 1 Cádiz     Algeciras            123078 11945     264 (279223 4001392)
 2 Cádiz     Barbate               22556  2369      27 (237228 4009013)
 3 Cádiz     Tarifa                18183  1542      30 (265096 3988986)
 4 Cádiz     San Roque             31571  3000      50 (285306 4010004)
 5 Cádiz     Barrios (Los)         23777  2164      80 (276026 4007210)
 6 Cádiz     Conil de la Fronte…   22775  2243      10 (222693 4019422)
 7 Cádiz     Castellar de la Fr…    3057   300       4 (282718 4018348)
 8 Cádiz     Benalup-Casas Viej…    6986   664       6 (247412 4025926)
 9 Málaga    Manilva               16439  1540      22 (298249 4027913)
10 Cádiz     Vejer de la Fronte…   12622  1051      10 (233423 4015856)
# … with 768 more rows, and abbreviated variable names ¹Poblacion, ²Fallecidos
```

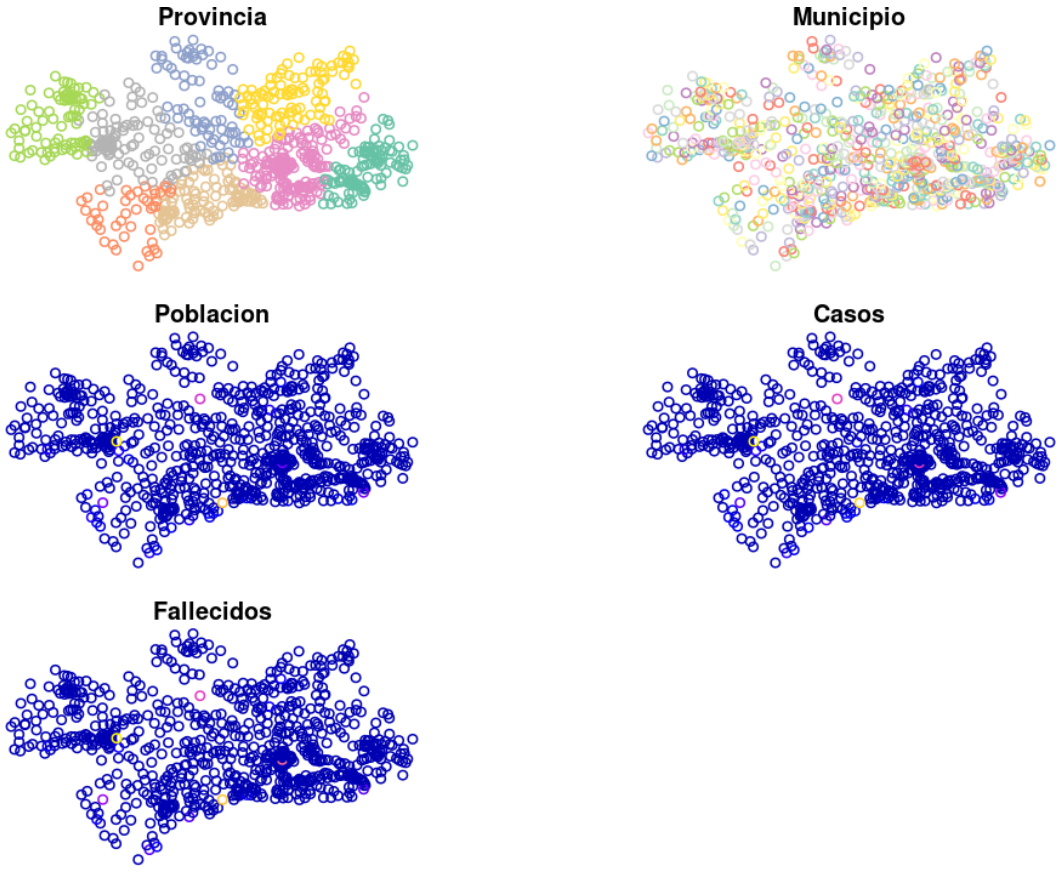
Manipulating sf objects with dplyr

- Filter rows from Huelva
- Filter rows from Huelva and Sevilla
- Filter towns in Granada with > 10 deaths
- Get average number of cases in each province

Mapping sf objects

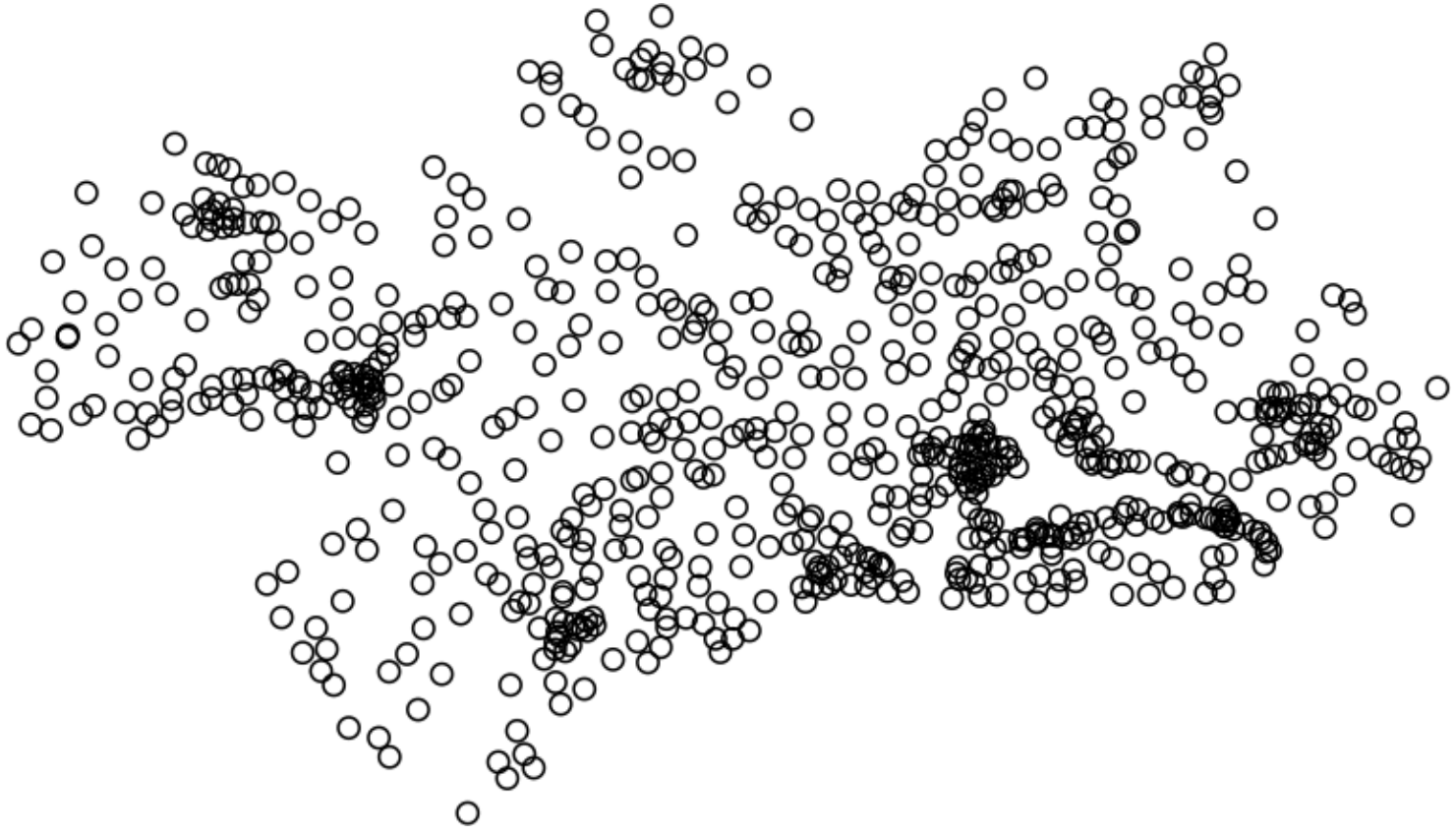
Basic mapping (sf::plot)

```
plot(covid.sf)
```



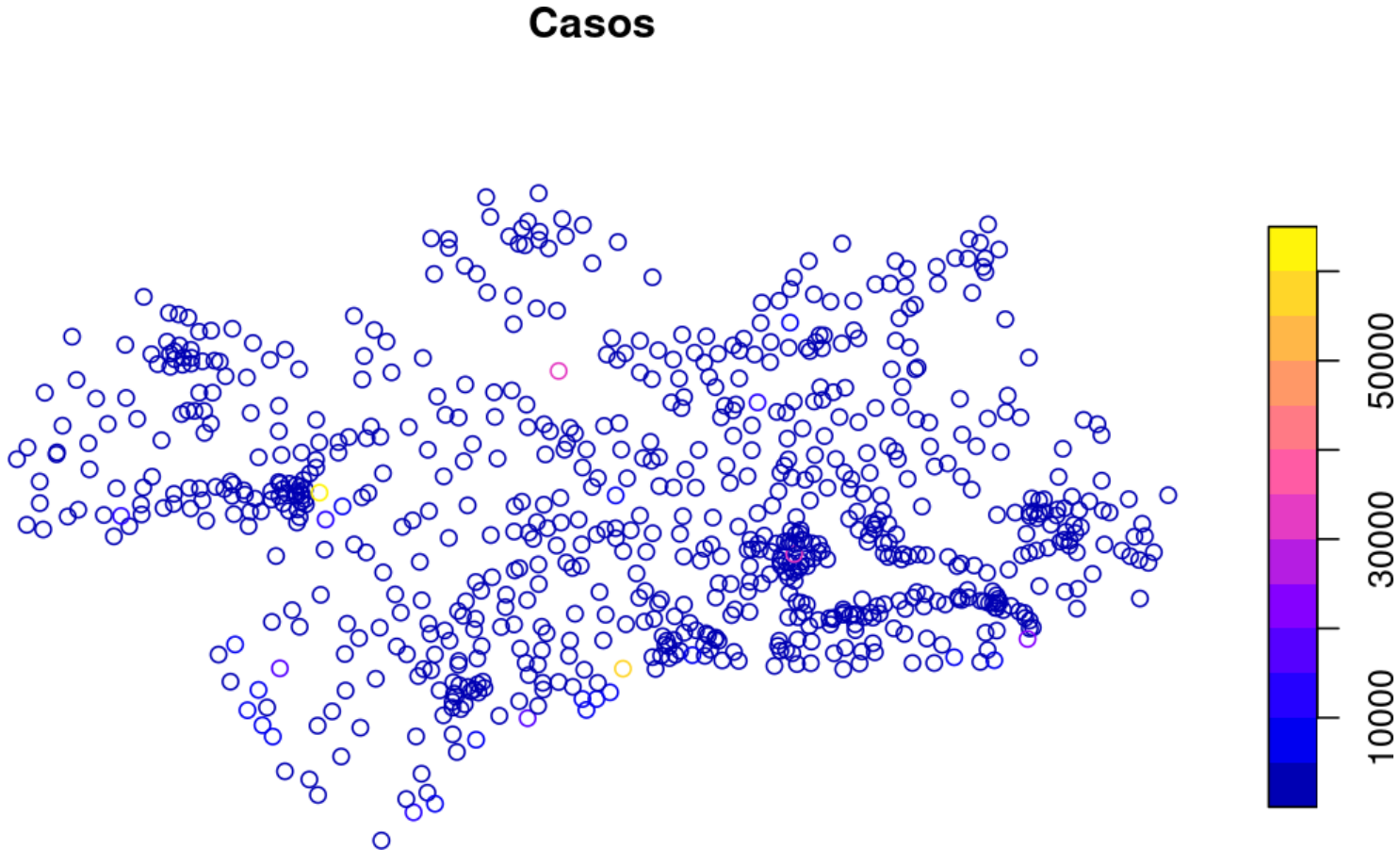
Basic mapping (sf::plot)

```
plot(covid.sf[0])
```



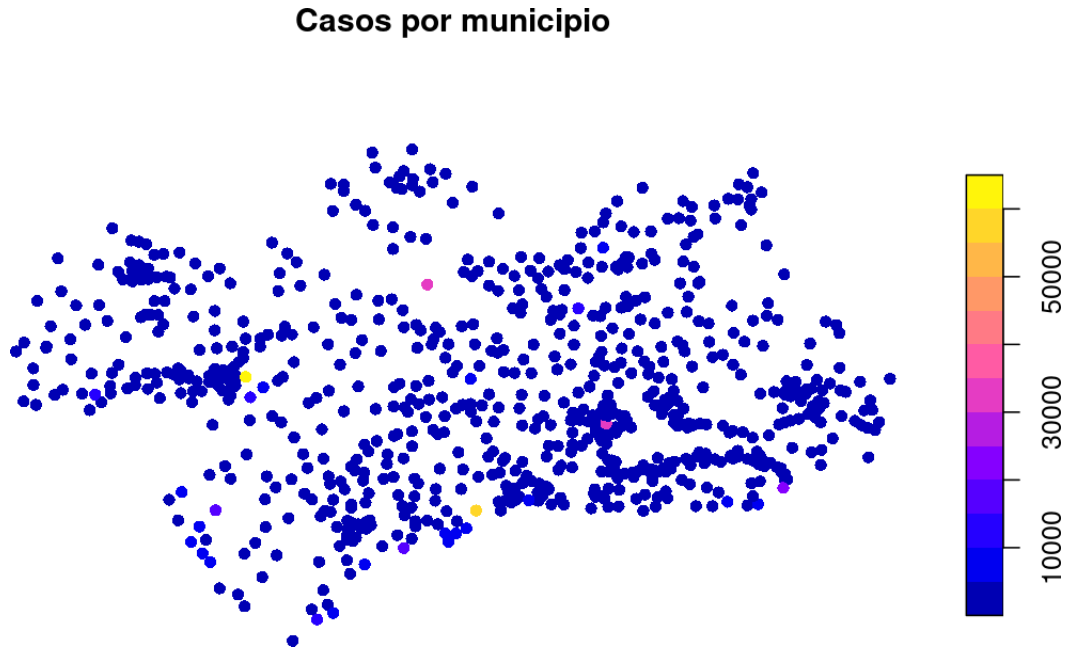
Basic mapping (sf::plot)

```
plot(covid.sf["Casos"])
```



Basic mapping (sf::plot)

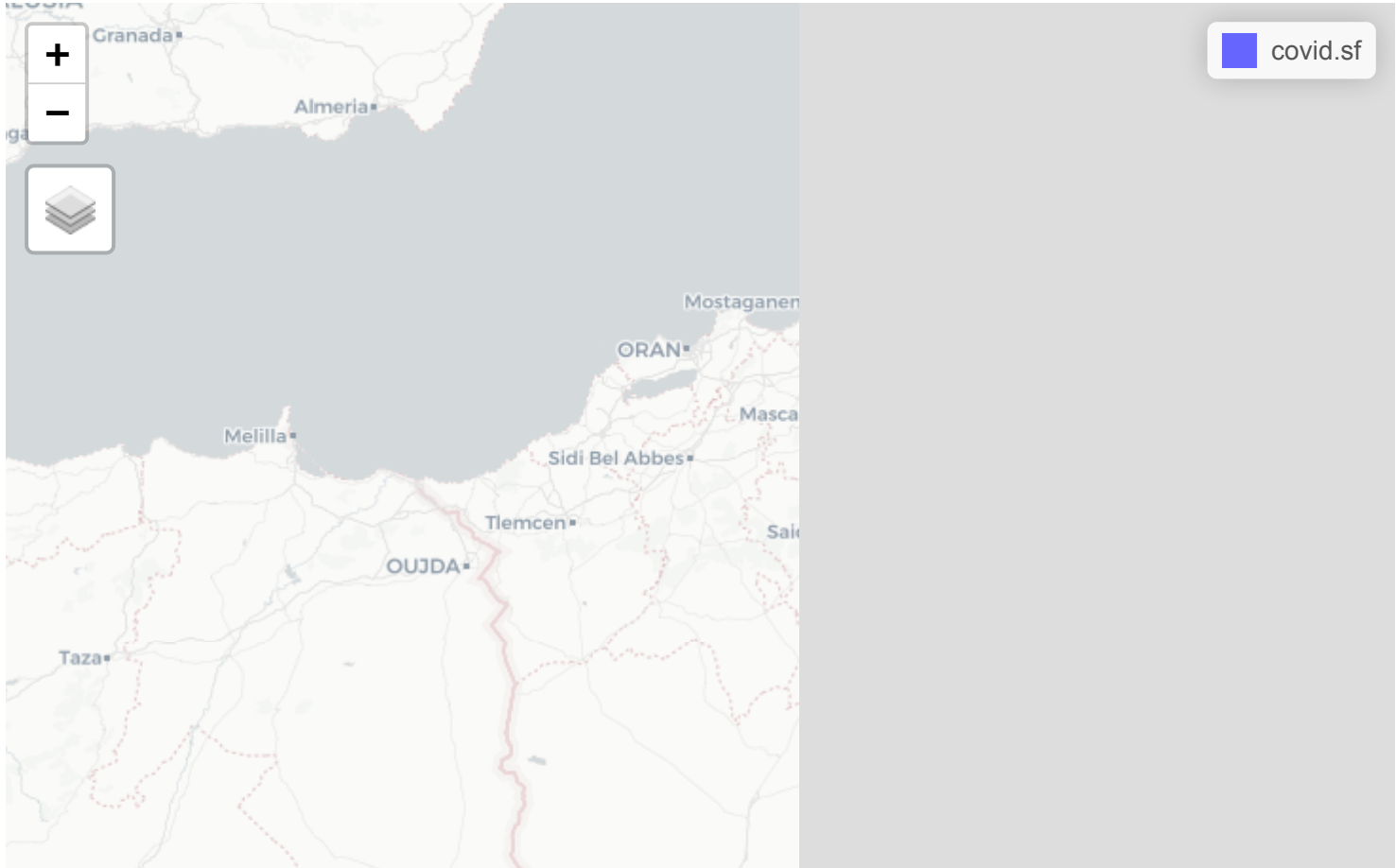
```
plot(covid.sf["Casos"],
     main = "Casos por municipio",
     pch = 16)
```



Quick interactive (leaflet) maps

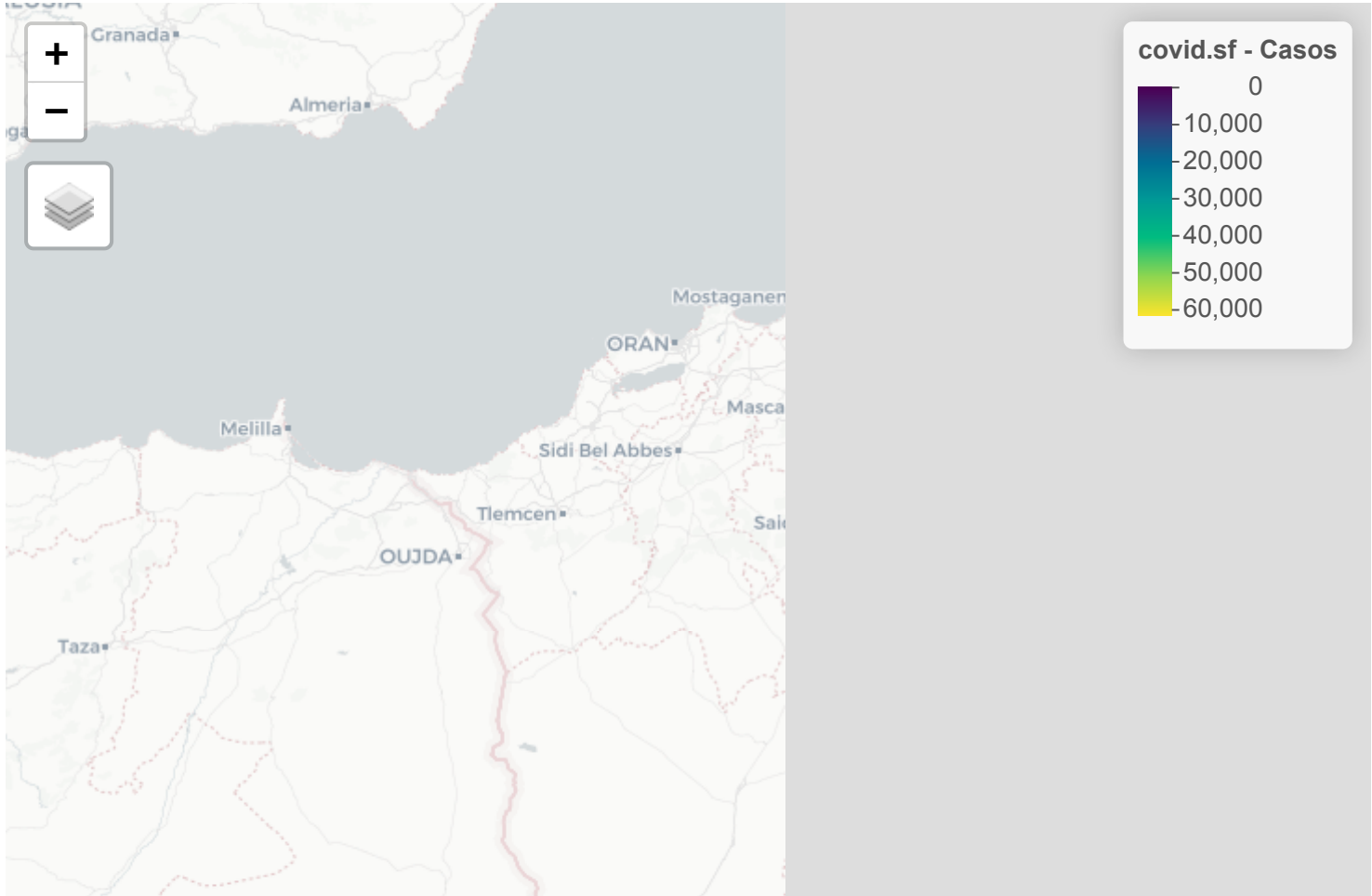
Quick interactive maps with mapview

```
library(mapview)
mapview(covid.sf)
```



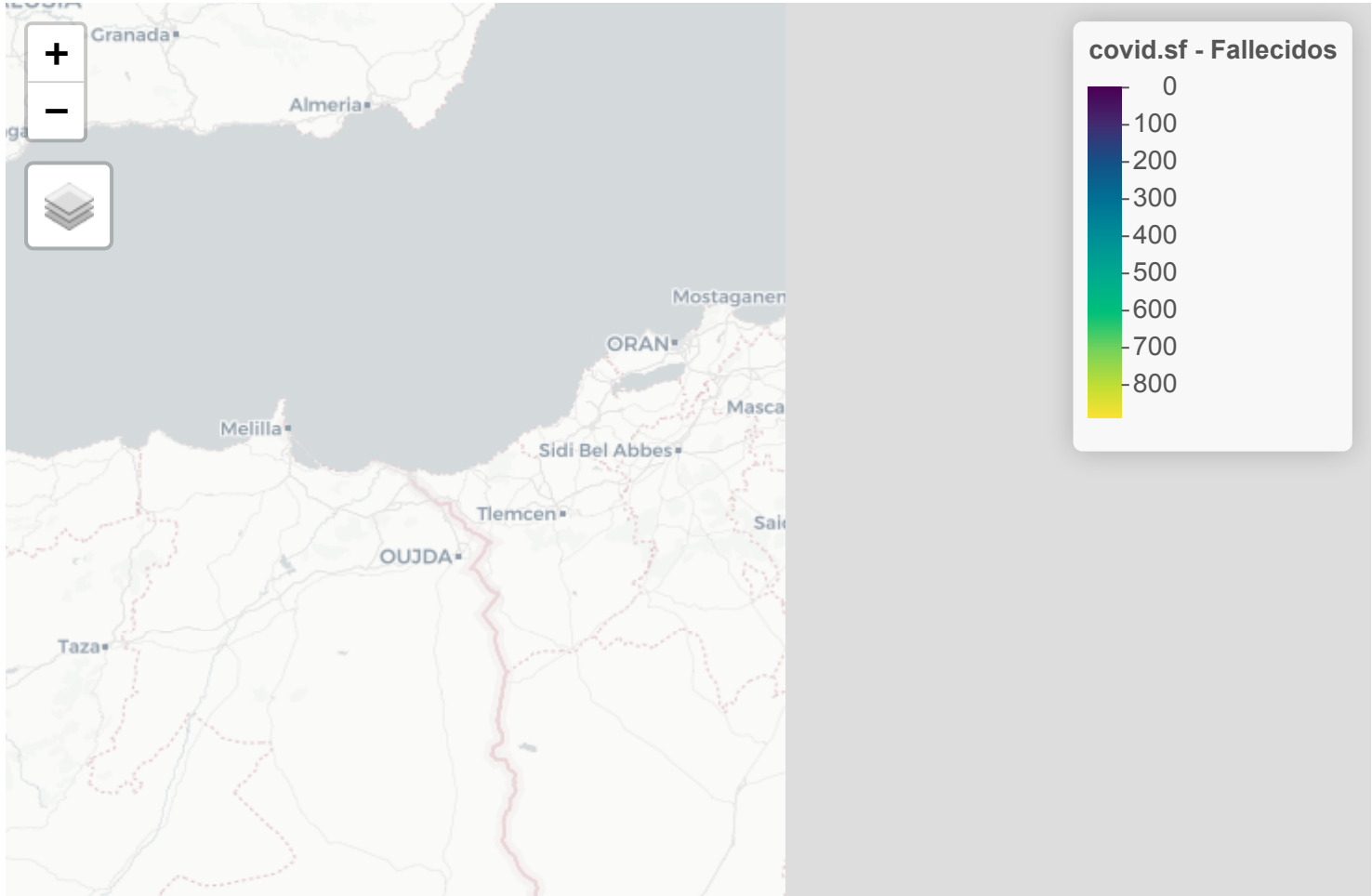
Quick interactive maps with mapview

```
mapview(covid.sf, zcol = "Casos")
```



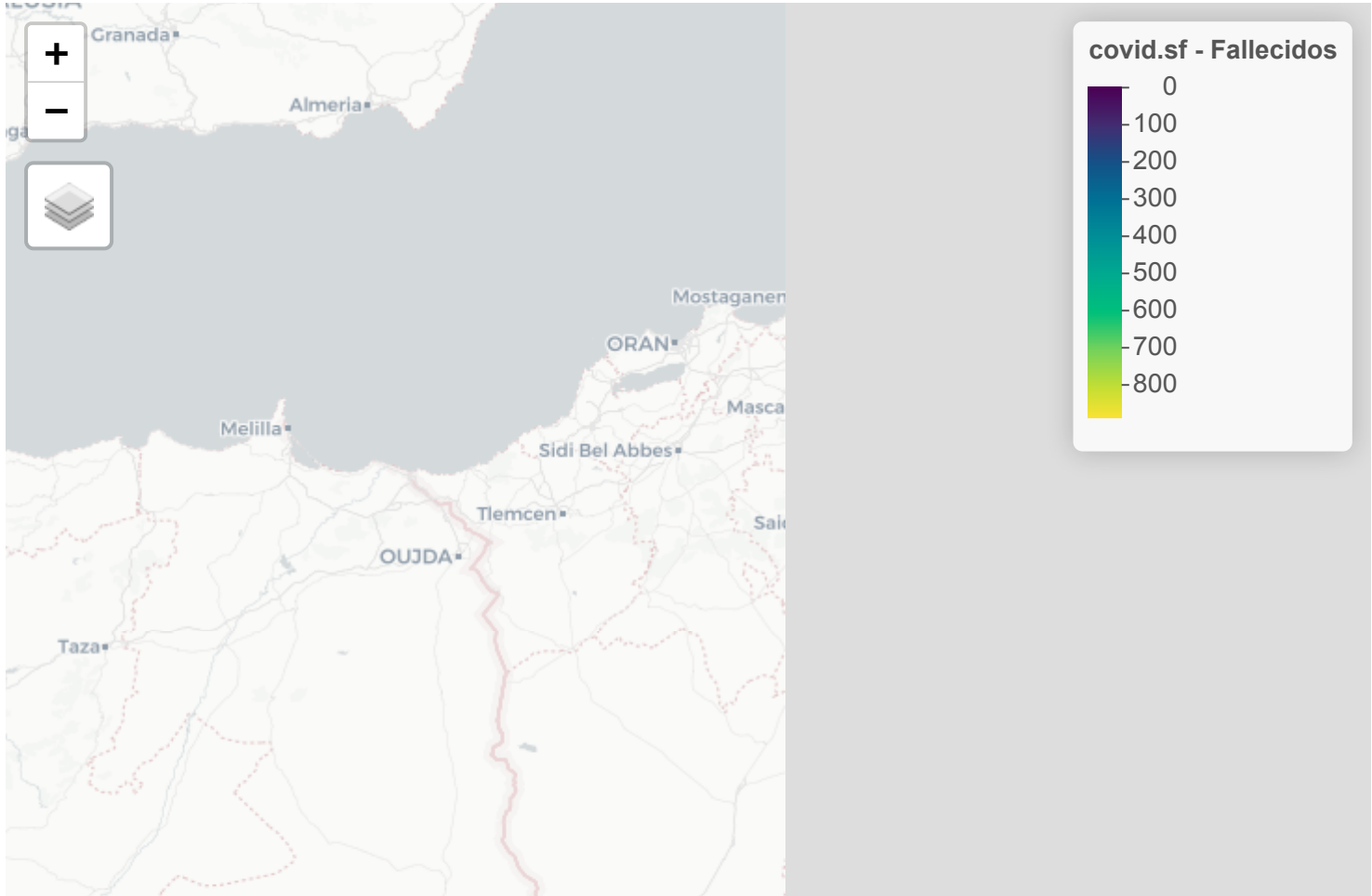
Quick interactive maps with mapview

```
mapview(covid.sf, zcol = "Fallecidos")
```



Quick interactive maps with mapview

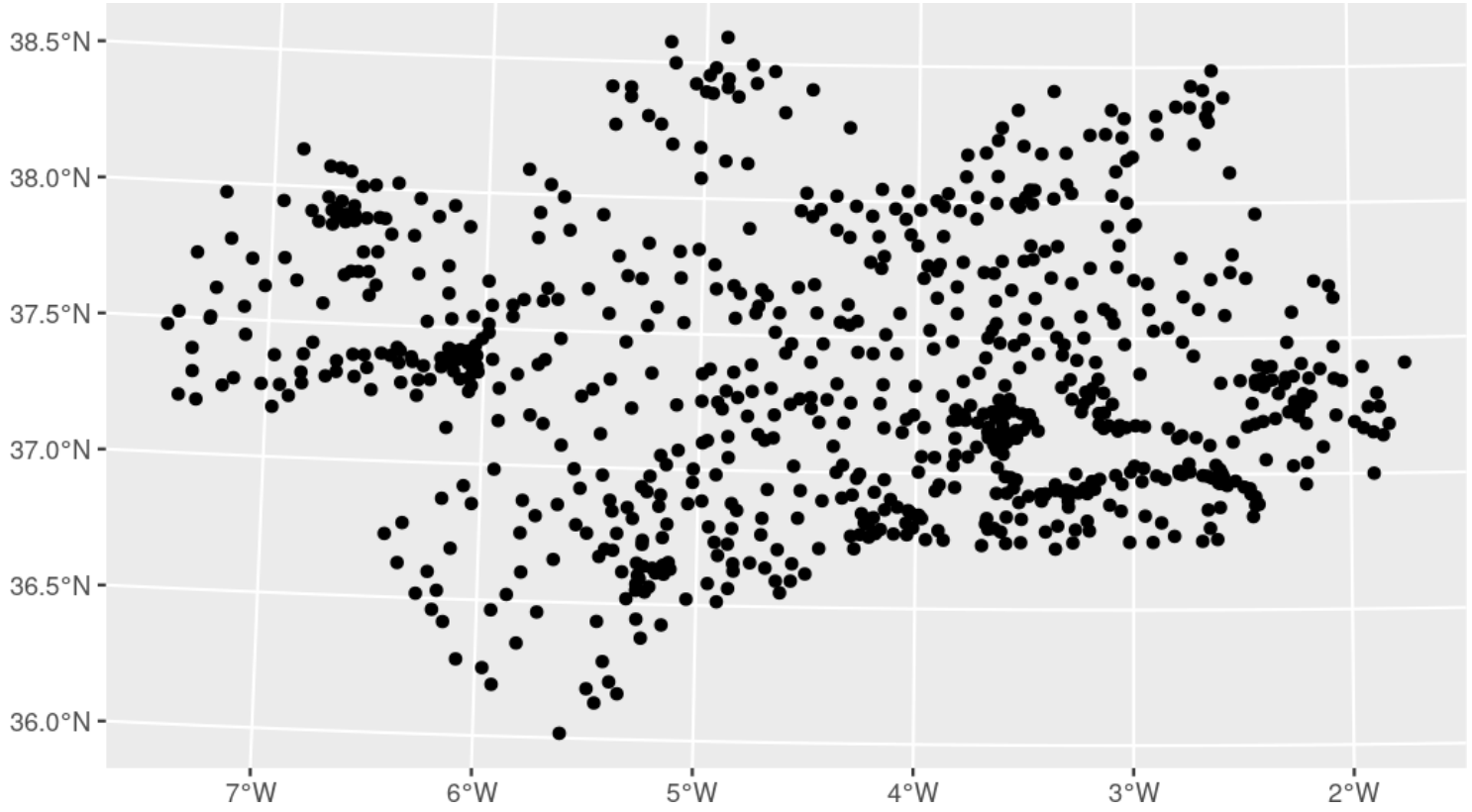
```
mapview(covid.sf, zcol = "Fallecidos", cex = "Fallecidos")
```



Mapping with ggplot2

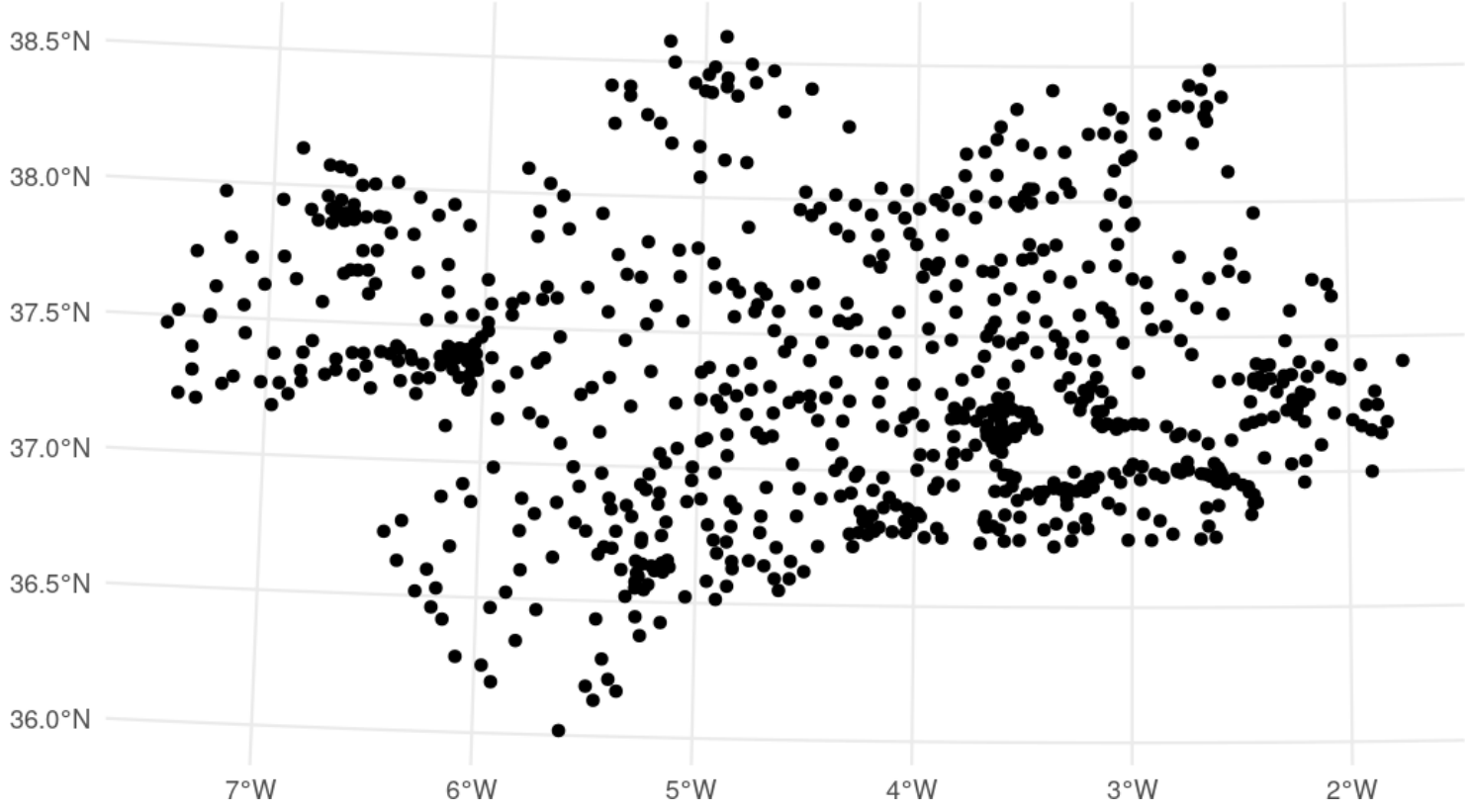
Mapping with ggplot2

```
library(ggplot2)

ggplot(covid.sf) +
  geom_sf()
```



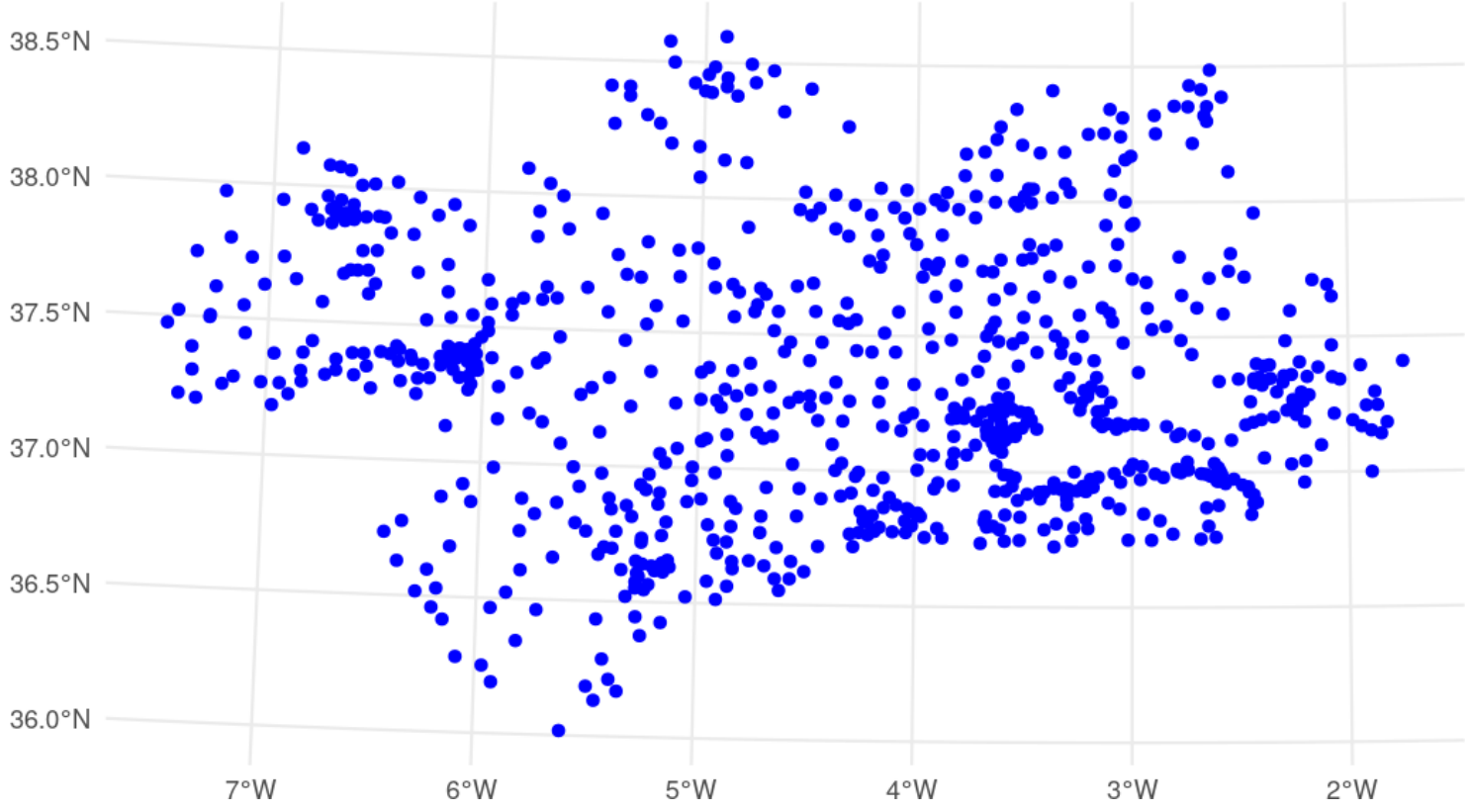
Mapping with ggplot2

```
ggplot(covid.sf) +
  geom_sf() +
  theme_minimal()
```



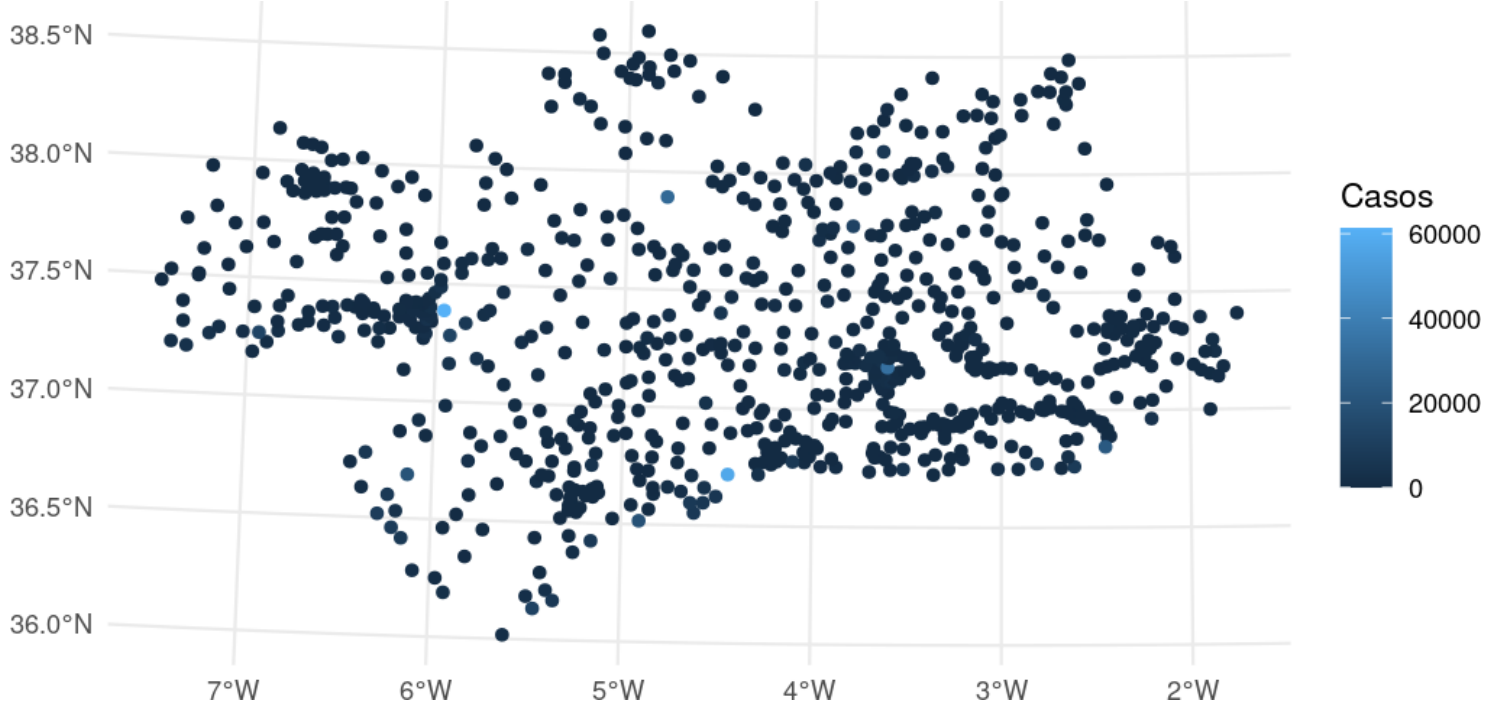
Mapping with ggplot2

```
ggplot(covid.sf) +
  geom_sf(colour = "blue") +
  theme_minimal()
```



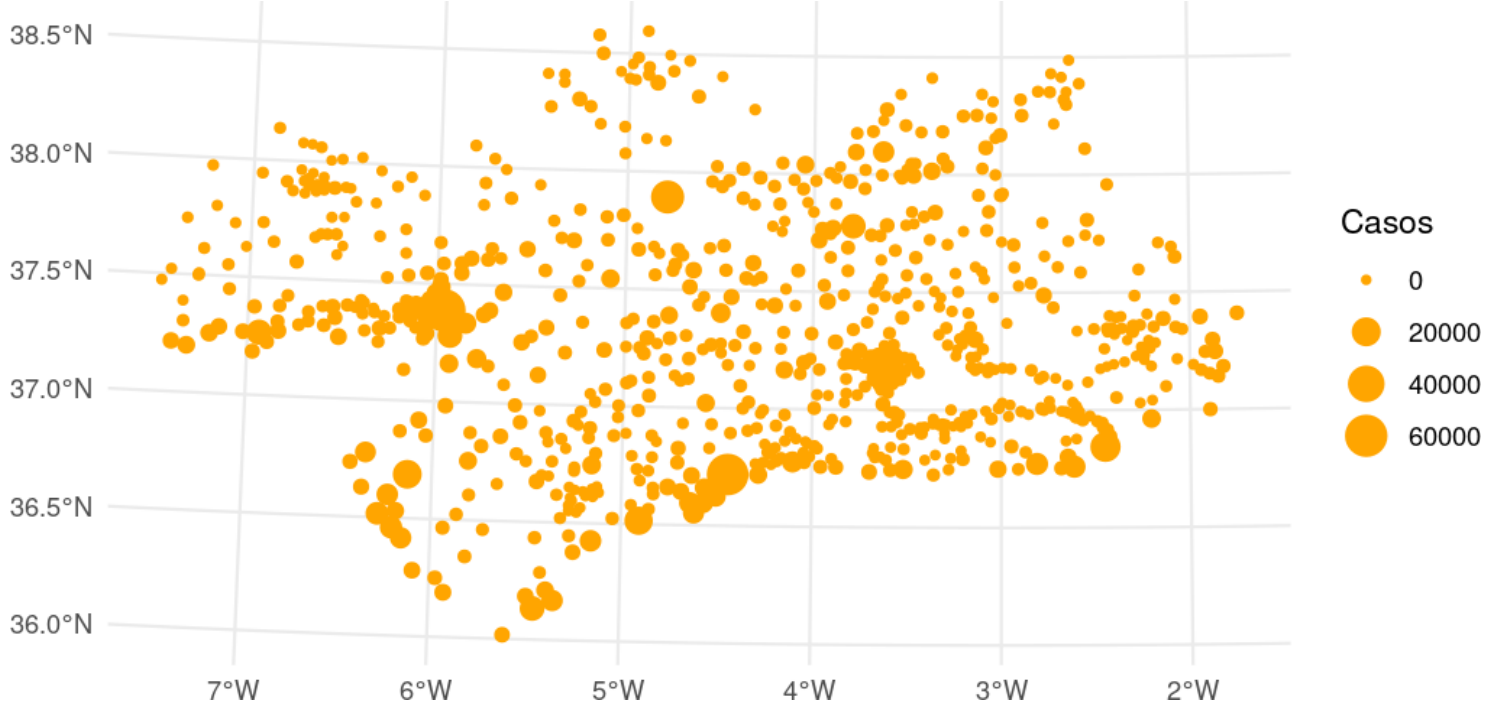
Mapping with ggplot2

```
ggplot(covid.sf) +
  geom_sf(aes(colour = Casos)) +
  theme_minimal()
```



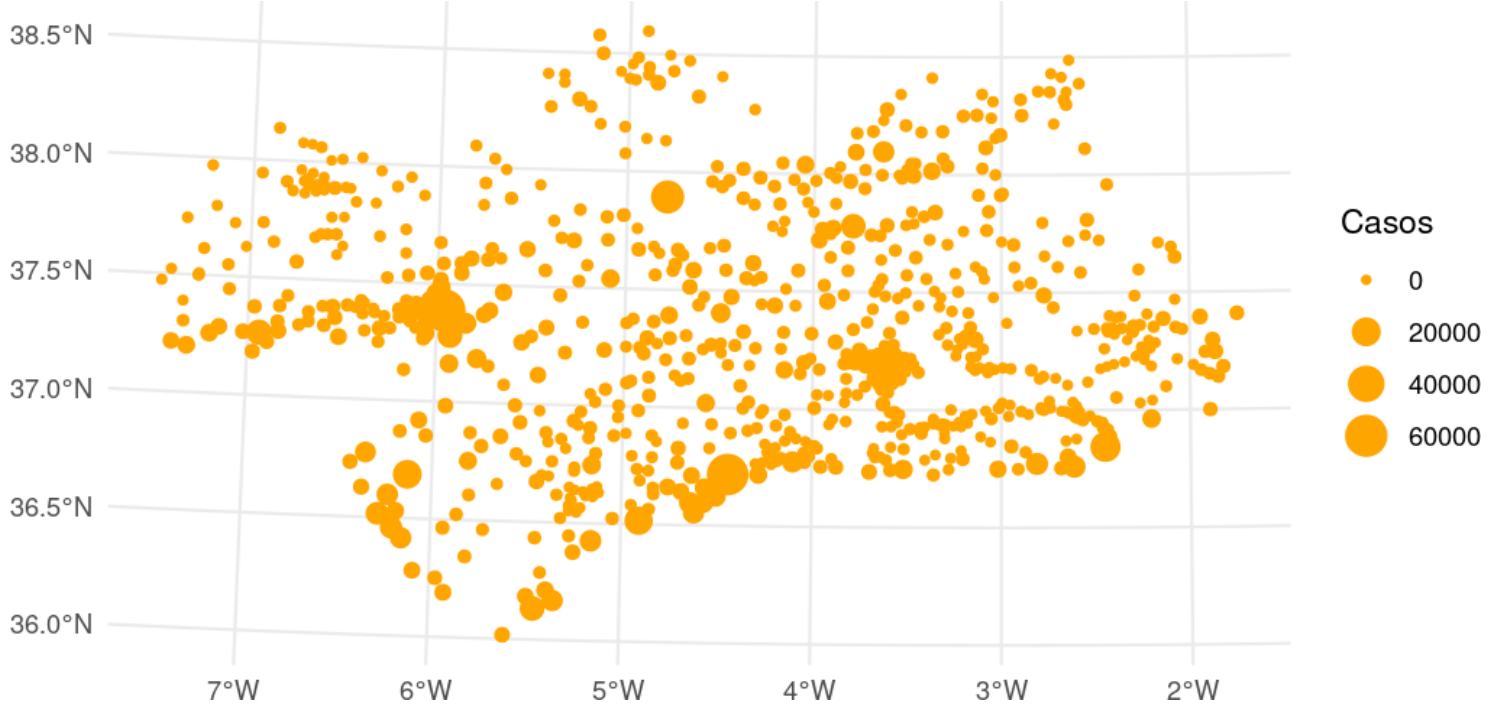
Mapping with ggplot2

```
ggplot(covid.sf) +
  geom_sf(aes(size = Casos), colour = "orange") +
  theme_minimal()
```



Mapping with ggplot2

```
ggplot(covid.sf) +
  geom_sf(aes(size = Casos), colour = "orange") +
  theme_minimal()
```



Mapping with ggplot2

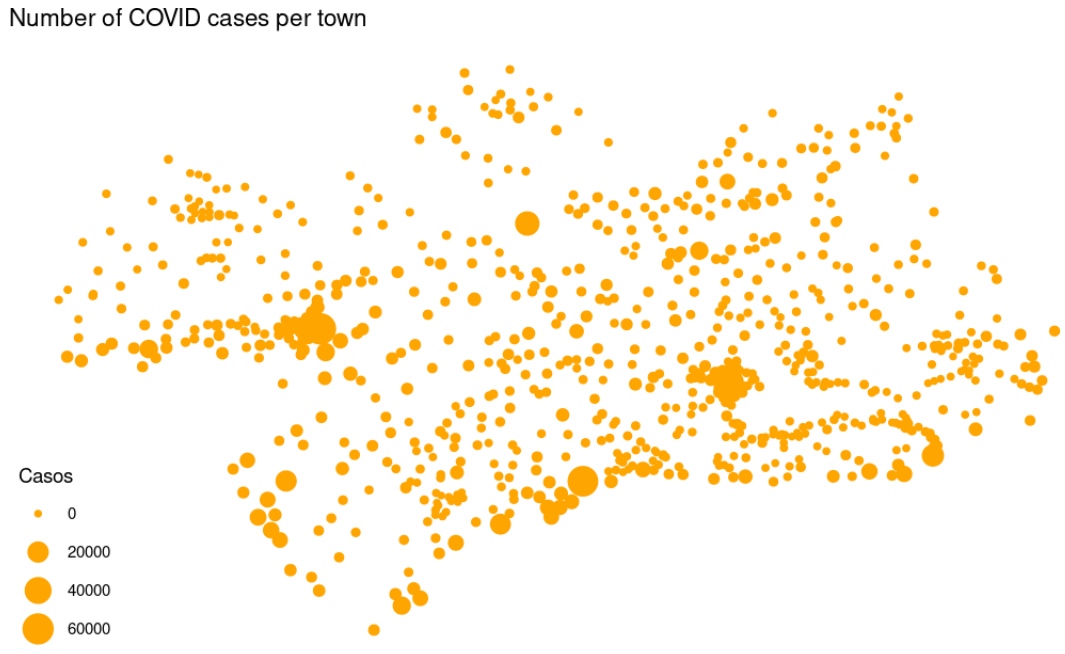
```
ggplot(covid.sf) +
  geom_sf(aes(size = Casos), colour = "orange") +
  cowplot::theme_map()
```

Mapping with ggplot2

```
ggplot(covid.sf) +
  geom_sf(aes(size = Casos), colour = "orange") +
  ggthemes::theme_map()
```

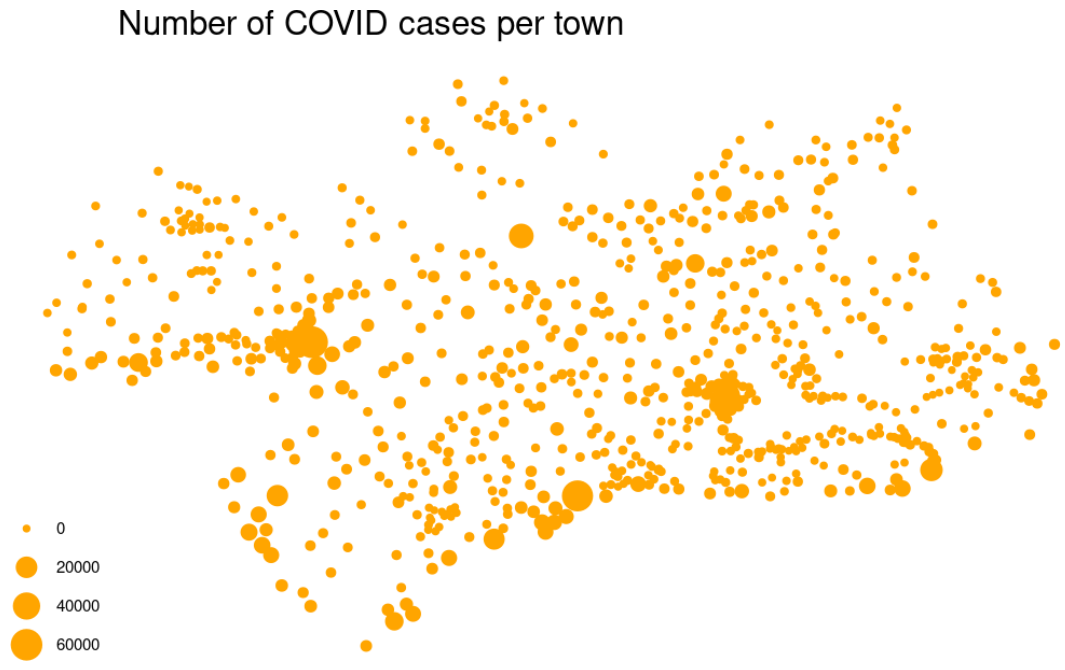
Mapping with ggplot2

```
ggplot(covid.sf) +
  geom_sf(aes(size = Casos), colour = "orange") +
  ggthemes::theme_map() +
  labs(title = "Number of COVID cases per town")
```



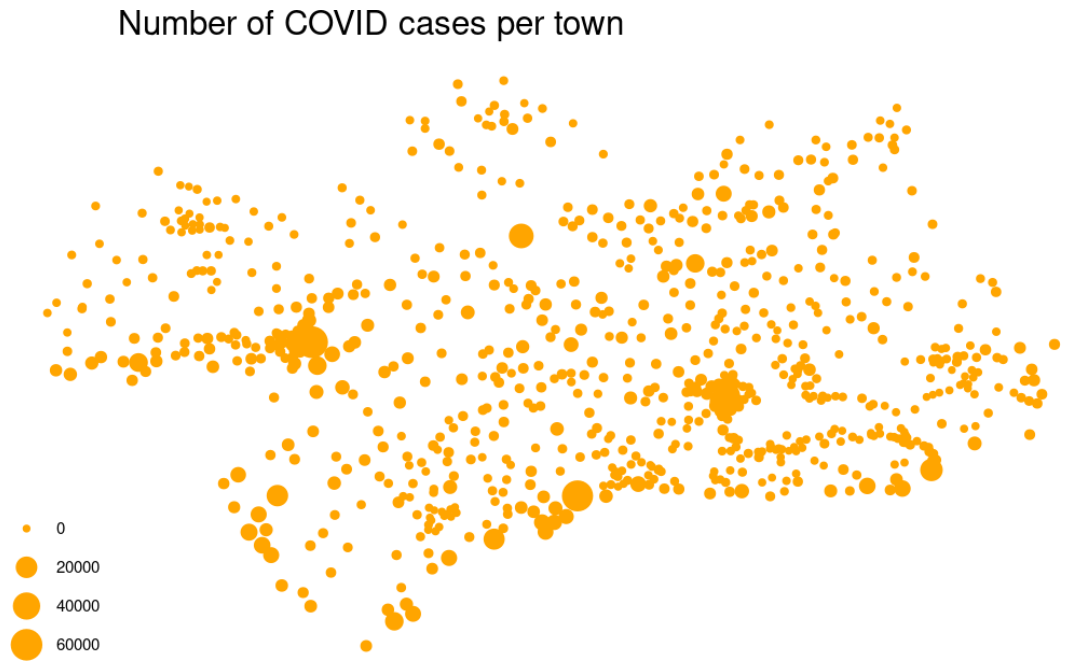
Mapping with ggplot2

```
ggplot(covid.sf) +
  geom_sf(aes(size = Casos), colour = "orange") +
  ggthemes::theme_map() +
  labs(title = "Number of COVID cases per town") +
  theme(plot.title = element_text(size = 15, hjust = 0.2),
        legend.title = element_blank())
```



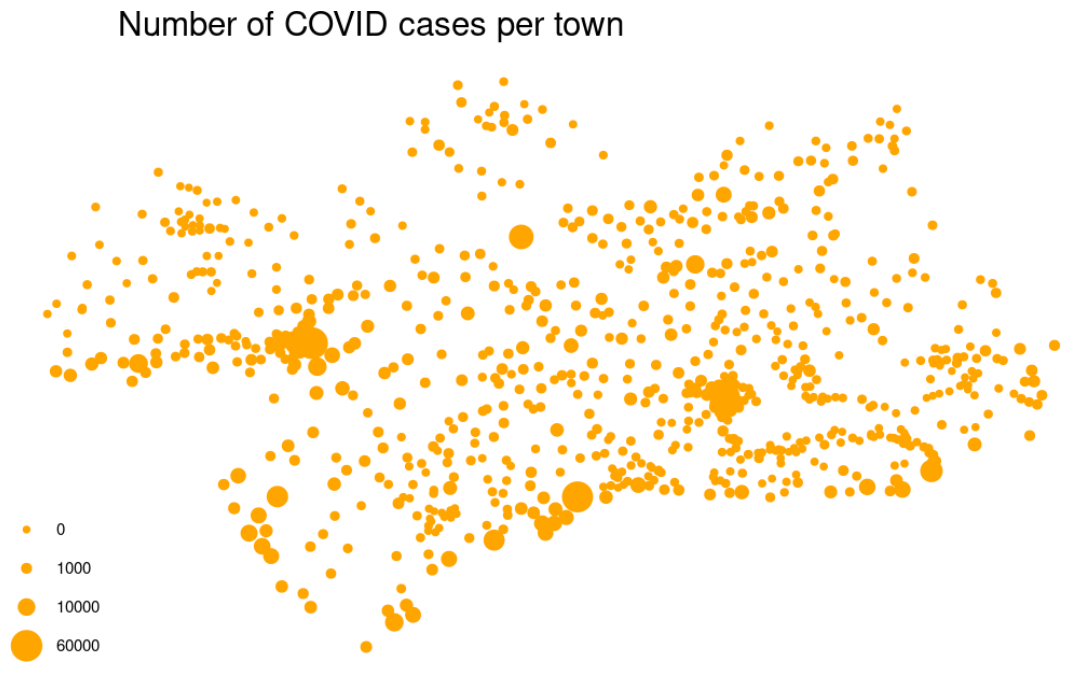
Mapping with ggplot2

```
ggplot(covid.sf) +
  geom_sf(aes(size = Casos), colour = "orange") +
  ggthemes::theme_map() +
  labs(title = "Number of COVID cases per town") +
  theme(plot.title = element_text(size = 15, hjust = 0.2),
        legend.title = element_blank())
```



Mapping with ggplot2

```
ggplot(covid.sf) +
  geom_sf(aes(size = Casos), colour = "orange") +
  ggthemes::theme_map() +
  labs(title = "Number of COVID cases per town") +
  theme(plot.title = element_text(size = 15, hjust = 0.2),
        legend.title = element_blank()) +
  scale_size_continuous(breaks = c(0, 1000, 10000, 60000))
```

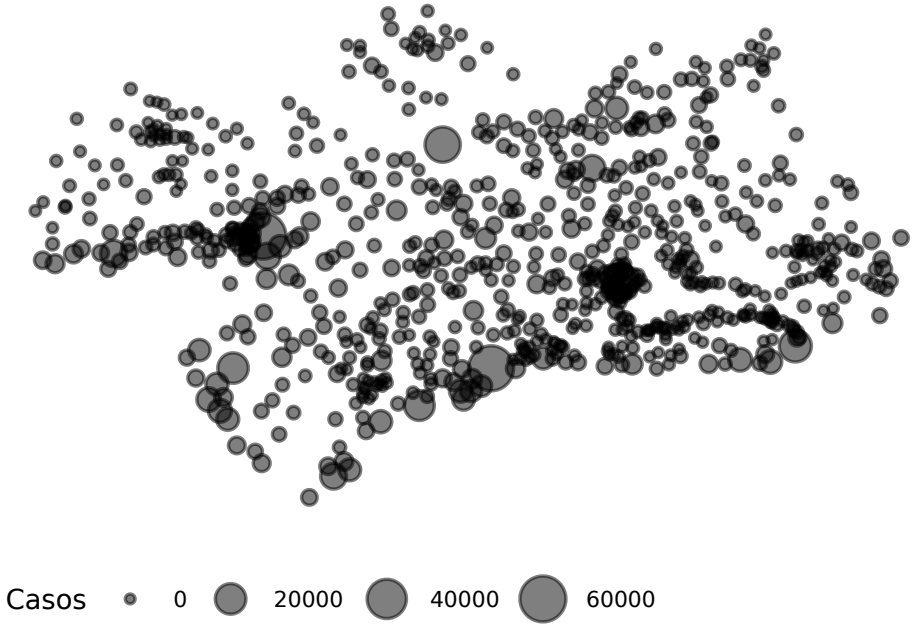


Saving map

```
mymap <- ggplot(covid.sf) +
  geom_sf(aes(size = Casos), colour = "orange") +
  ggthemes::theme_map() +
  labs(title = "Number of COVID cases per town") +
  theme(plot.title = element_text(size = 15, hjust = 0.2),
        legend.title = element_blank())

ggsave(filename = "mymap.pdf", plot = mymap,
       width = 10, height = 7, units = "cm")
```

Interactive map

```
library(ggiraph)

ggobj <- ggplot(covid.sf) +
  geom_sf_interactive(aes(size = Casos,
                          tooltip = Municipio
                      alpha = 0.5) +
  ggthemes::theme_map(base_size = 8) +
  theme(legend.position = "bottom")

girafe(ggobj = ggobj, width_svg = 4, height_s
```



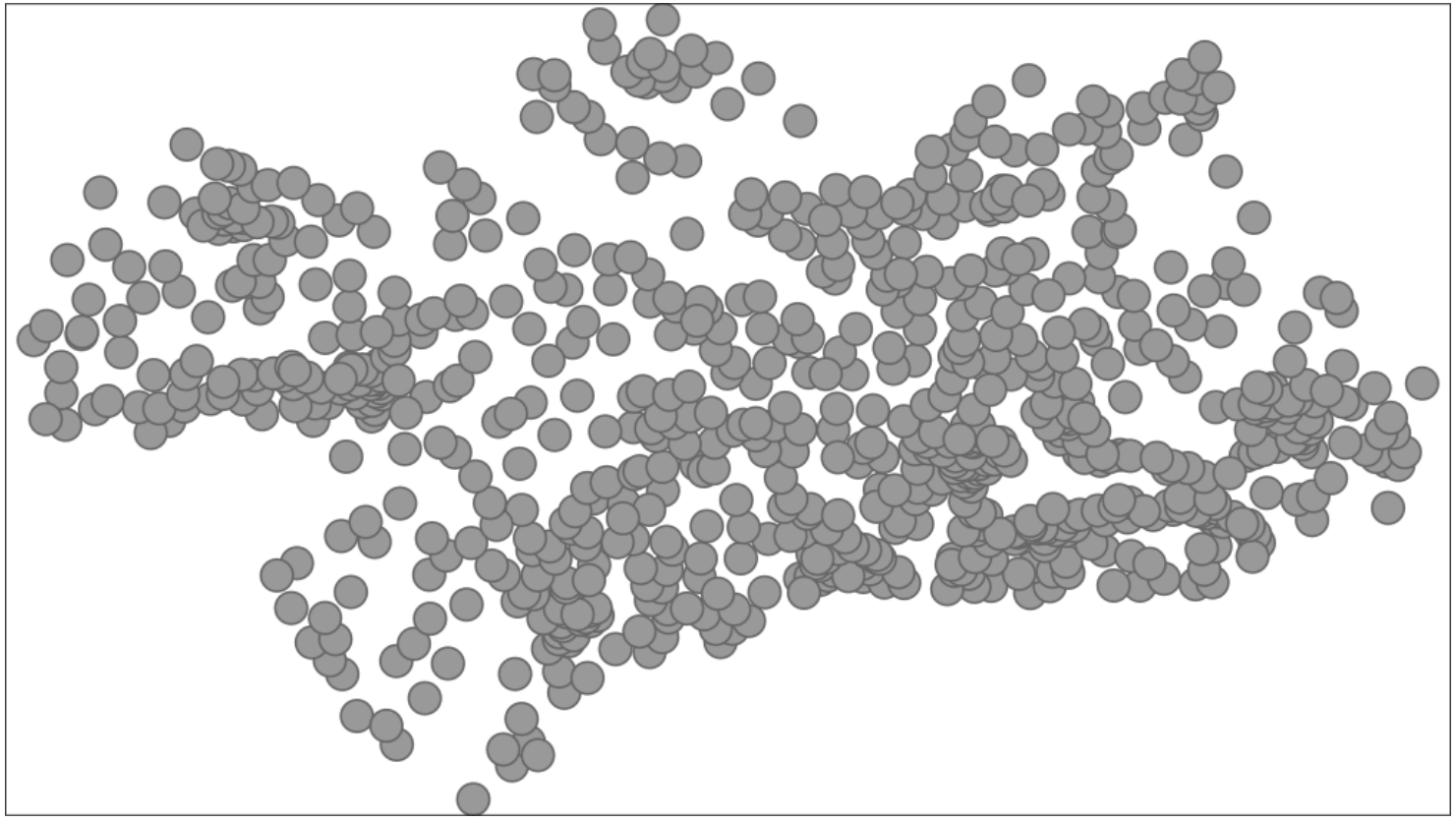
Can use all ggplot2 power

- Add many layers
- Label points (geom_text, ggrepel…)
- Facetted maps
- Combine maps (cowplot, patchwork…)
- Animate maps (gganimate)
- etc

Mapping with tmap

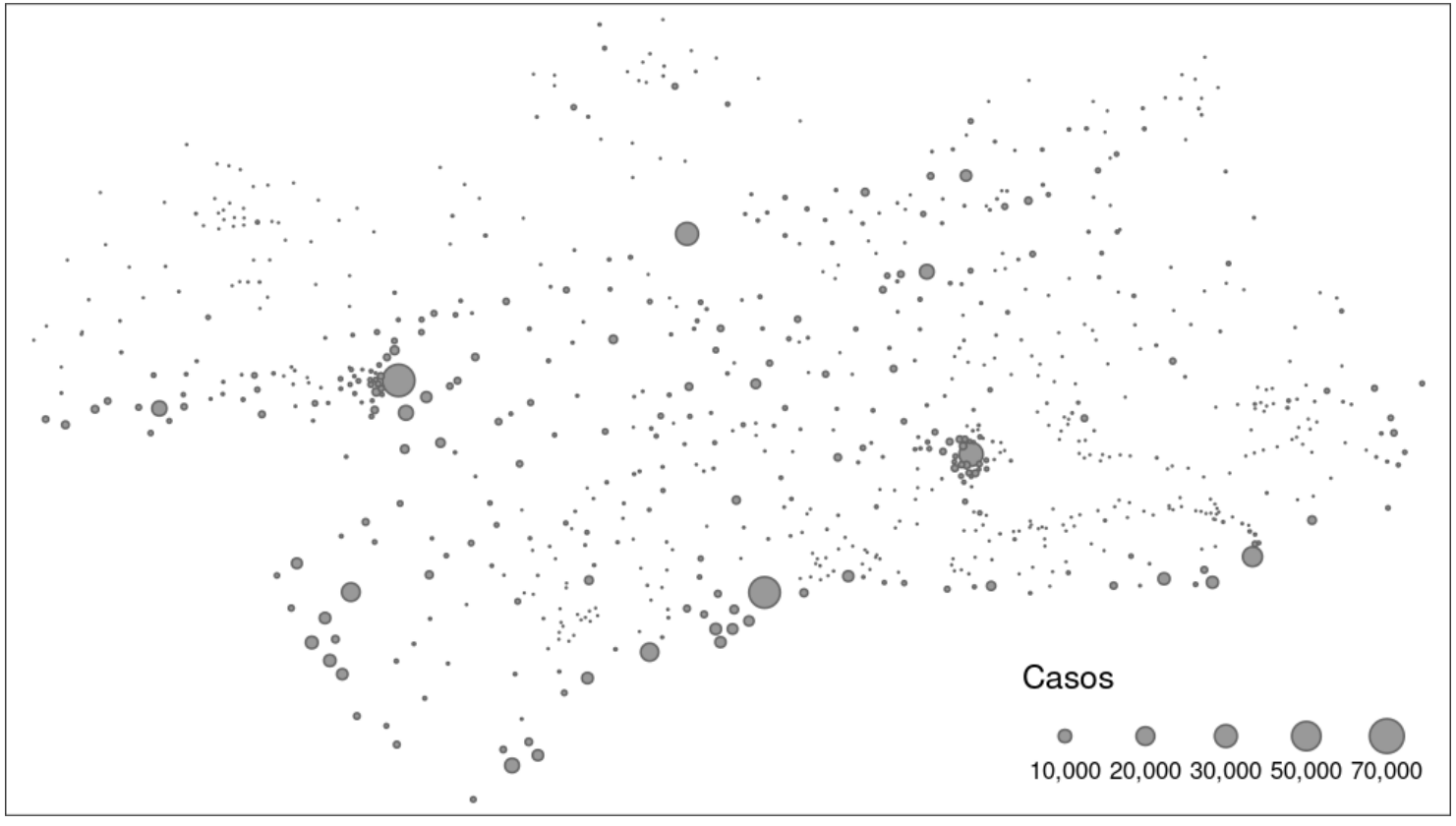
Mapping with 'tmap' package

```
library(tmap)

tm_shape(covid.sf) +    # specify spatial object
  tm_symbols()          # map points
```



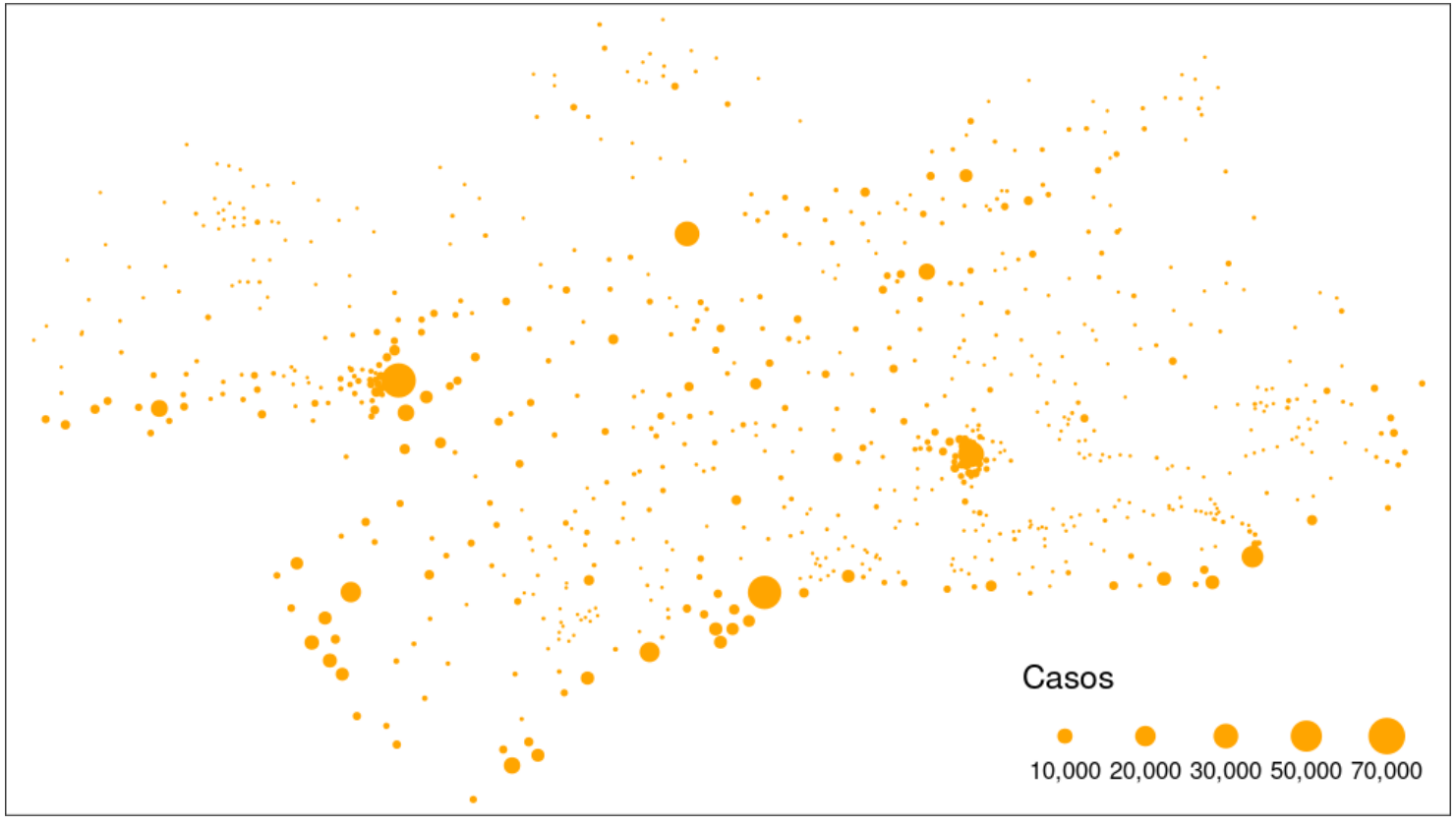
Mapping with 'tmap' package

```
tm_shape(covid.sf) +
  tm_symbols(size = "Casos")
```



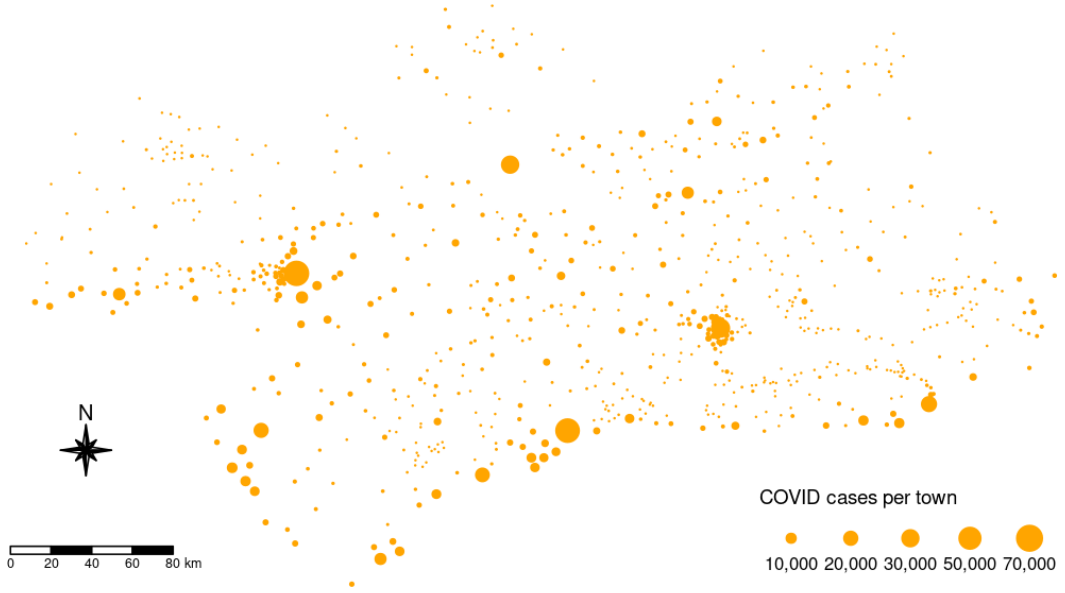
Mapping with 'tmap' package

```
tm_shape(covid.sf) +
  tm_symbols(size = "Casos",
             col = "orange", border.col = "orange")
```



Mapping with 'tmap' package

```
tm_shape(covid.sf) +
  tm_symbols(size = "Casos",
             col = "orange", border.col = "orange",
             title.size = "COVID cases per town") +
  tm_layout(frame = FALSE, legend.title.size = 0.9) +
  tm_compass(type = "8star", position = c(0.04, 0.2), size = 2) +
  tm_scale_bar(width = 0.15, position = c(0, 0.03))
```

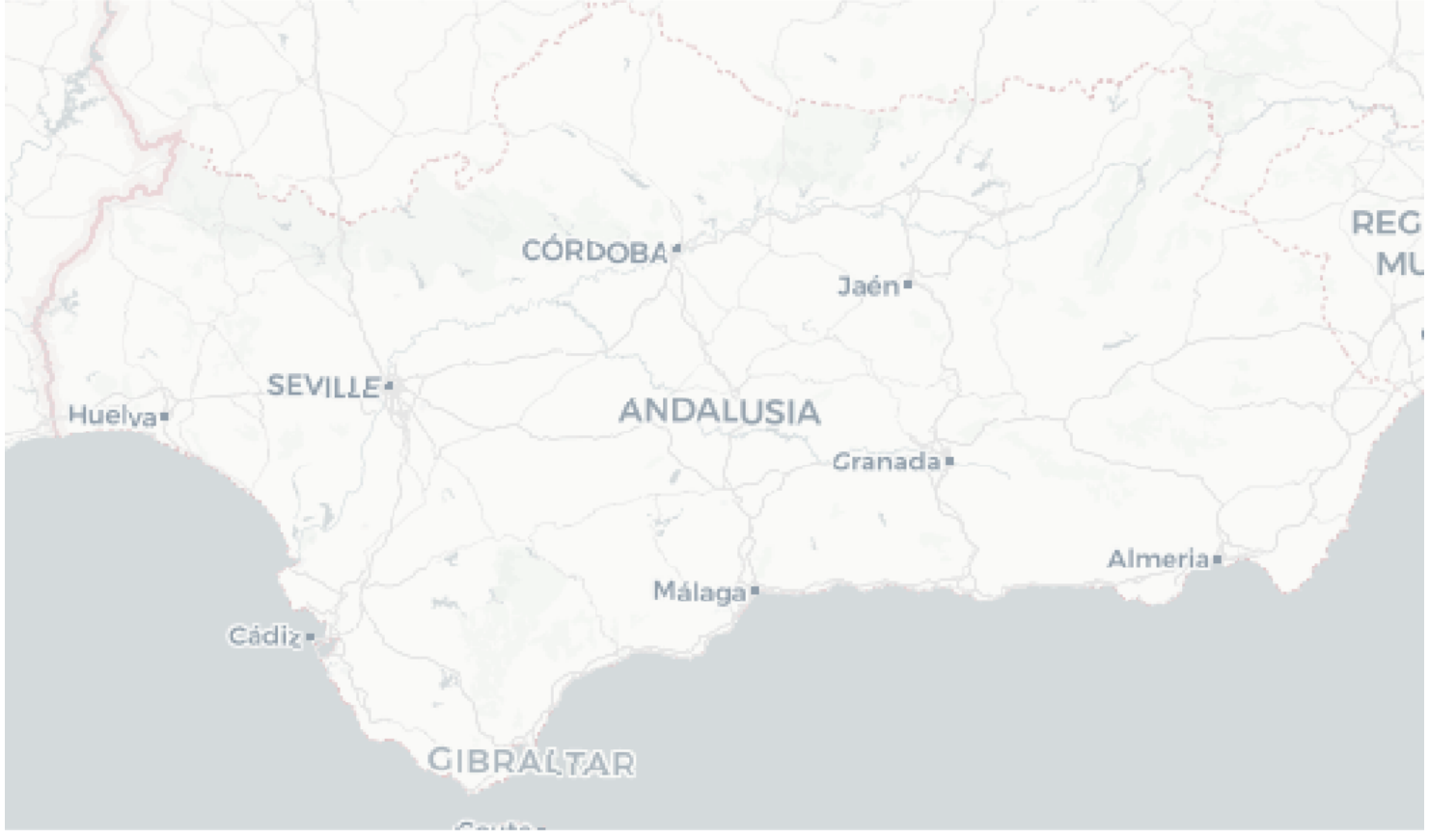


Adding basemap

```
library(maptiles)
library(terra)

bmap <- get_tiles(covid.sf, zoom = 7, crop = TRUE)
plot(bmap)
```

Adding basemap

```
bmap <- get_tiles(covid.sf, provider = "CartoDB.Positron", zoom = 7, crop = TRUE)
plot(bmap)
```



Adding basemap

```
bmap <- get_tiles(covid.sf, provider = "Esri.WorldShadedRelief", zoom = 7, crop = TRUE)
plot(bmap)
```

Adding basemap

```
bmap <- get_tiles(covid.sf, provider = "Esri.WorldImagery", zoom = 7, crop = TRUE)
plot(bmap)
```

Adding basemap

```
bmap <- get_tiles(covid.sf, provider = "Stamen.Watercolor", zoom = 7, crop = TRUE)
plot(bmap)
```

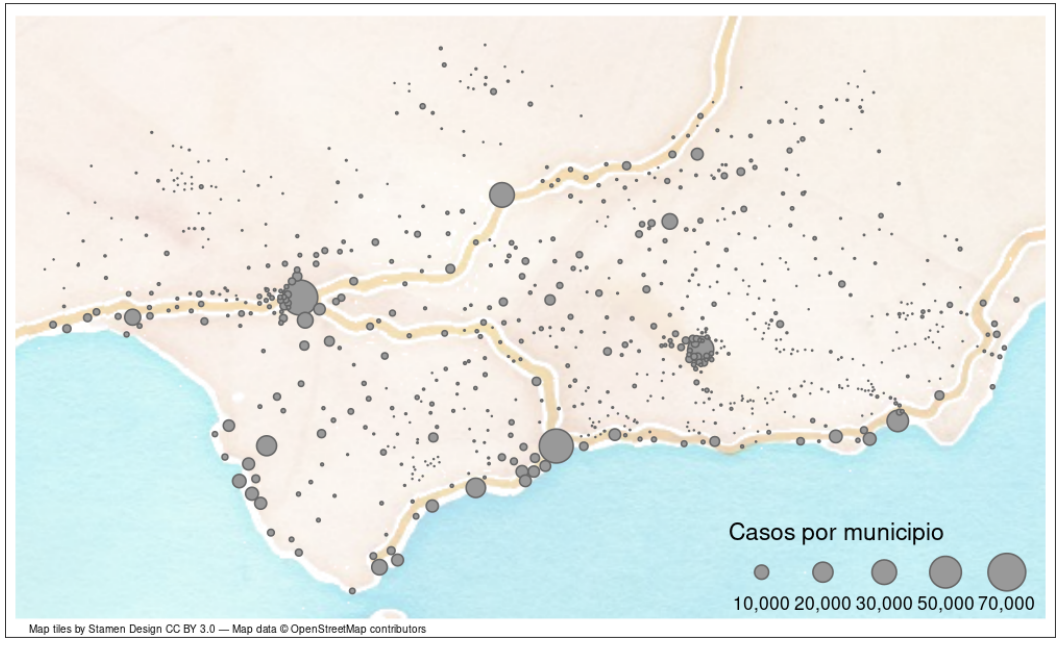
Adding basemap: give credit!

```
credits <- get_credit("Stamen.Watercolor")
credits
```

```
[1] "Map tiles by Stamen Design CC BY 3.0 — Map data © OpenStreetMap contributors"
```

Adding basemap

```
tm_shape(bmap) +
  tm_rgb(alpha = 0.3) +
  tm_shape(covid.sf) +
  tm_symbols(size = "Casos", scale = 1.5, title.size = "Casos por municipio") +
  tm_layout(legend.position = c("right", "bottom"),
            inner.margins = c(0.03, 0.01, 0.02, 0.01)) +
  tm_credits(get_credit("Stamen.Watercolor"), size = 0.4, position = c(0.02, 0))
```

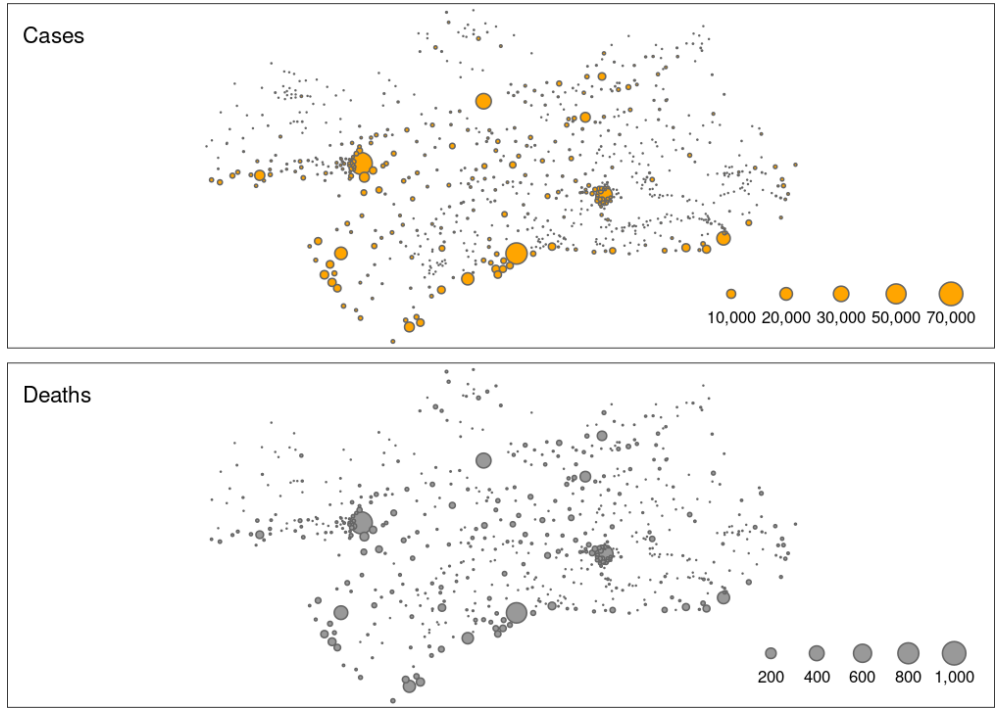


Quick zoom + add text labels

```
tm_shape(bmap, bbox = "provincia de Cádiz") +
  tm_rgb(alpha = 0.3) +
  tm_shape(covid.sf) +
  tm_symbols(size = "Casos", scale = 2,
             title.size = "Casos por municipio",
             sizes.legend = c(1000, 10000, 20000)) +
  tm_text(text = "Municipio", size = "Casos",
          legend.size.show = FALSE,
          auto.placement = FALSE, remove.overlap = TRUE, s
          xmod = 1, ymod = 0.7) +
  tm_layout(legend.position = c("left", "bottom"))
```

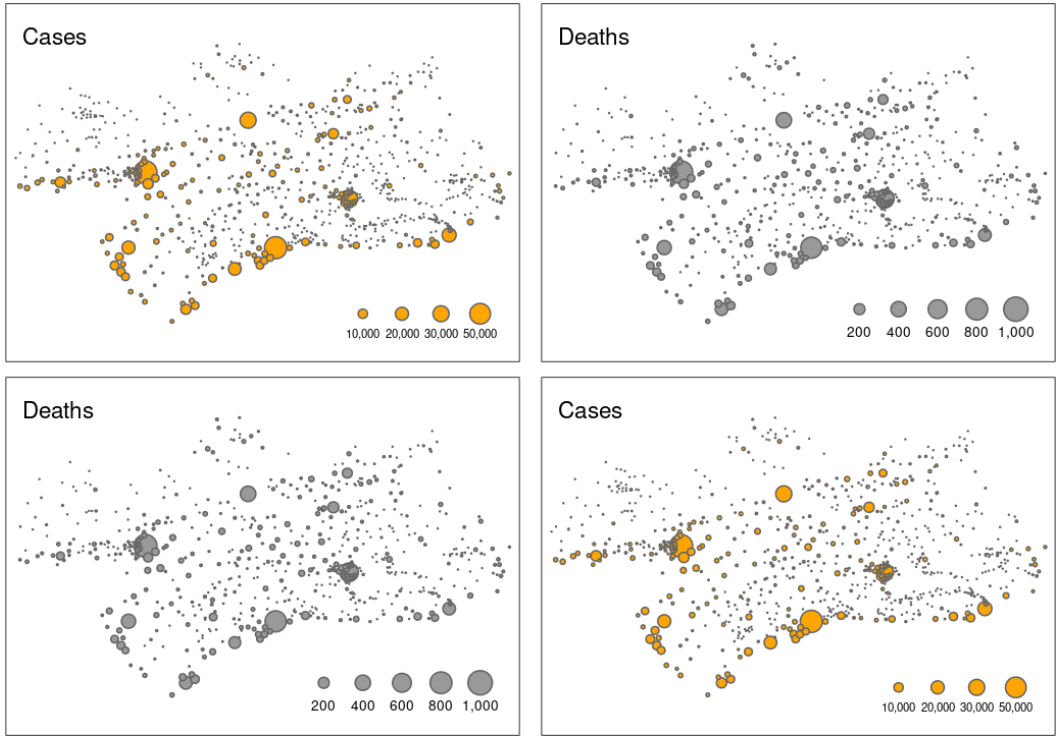
Arrange several maps

```
map.cases <- tm_shape(covid.sf) +
  tm_symbols(size = "Casos", col = "orange",
  tm_layout(title = "Cases", title.size = 0.9


map.deaths <- tm_shape(covid.sf) +
  tm_symbols(size = "Fallecidos", title.size
  tm_layout(title = "Deaths", title.size = 0.

tmap_arrange(map.cases, map.deaths)
```



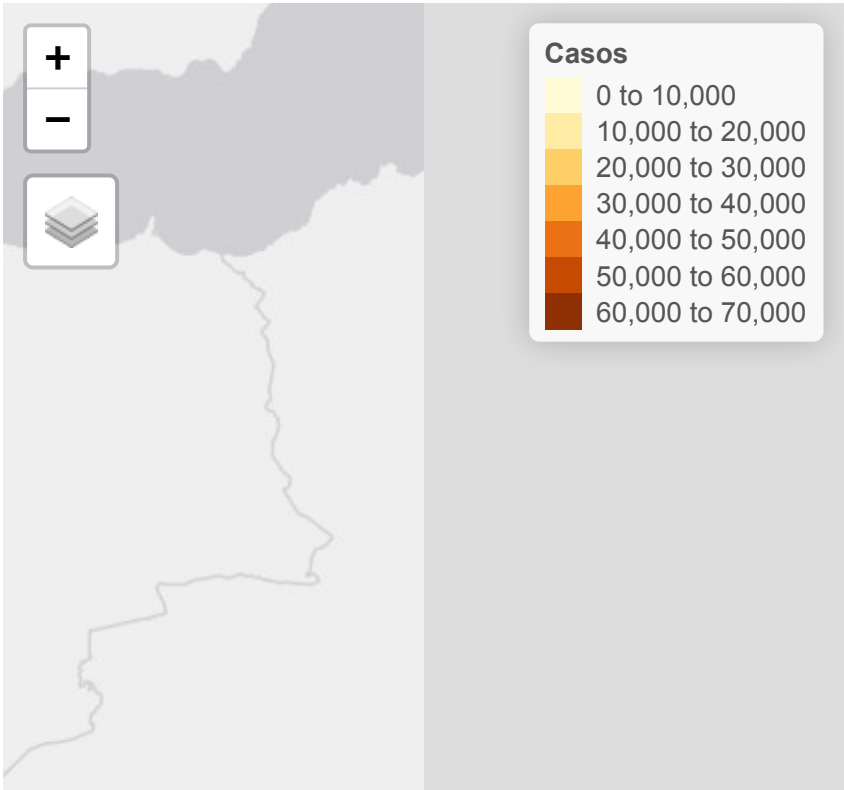
Arrange several maps

```
tmap_arrange(map.cases, map.deaths, map.deaths, map.cases,
             nrow = 2, ncol = 2)
```



Interactive maps

```
tmap_mode("view")

tm_shape(covid.sf) +
  tm_symbols(size = "Casos", col = "Casos", id = "Municipio",
             popup.vars = c("Municipio", "Casos"))
```



Saving tmap objects

```
tmap_save(mymap, "mymap.png")  # raster image

tmap_save(mymap, "mymap.svg")  # vector

tmap_save(mymap, "mymap.html")  # html (interactive, animated...)
```

Your turn

Reading spatial vector data

sf can read multiple spatial data types

```
library(sf)
sort(st_drivers()$name)
```

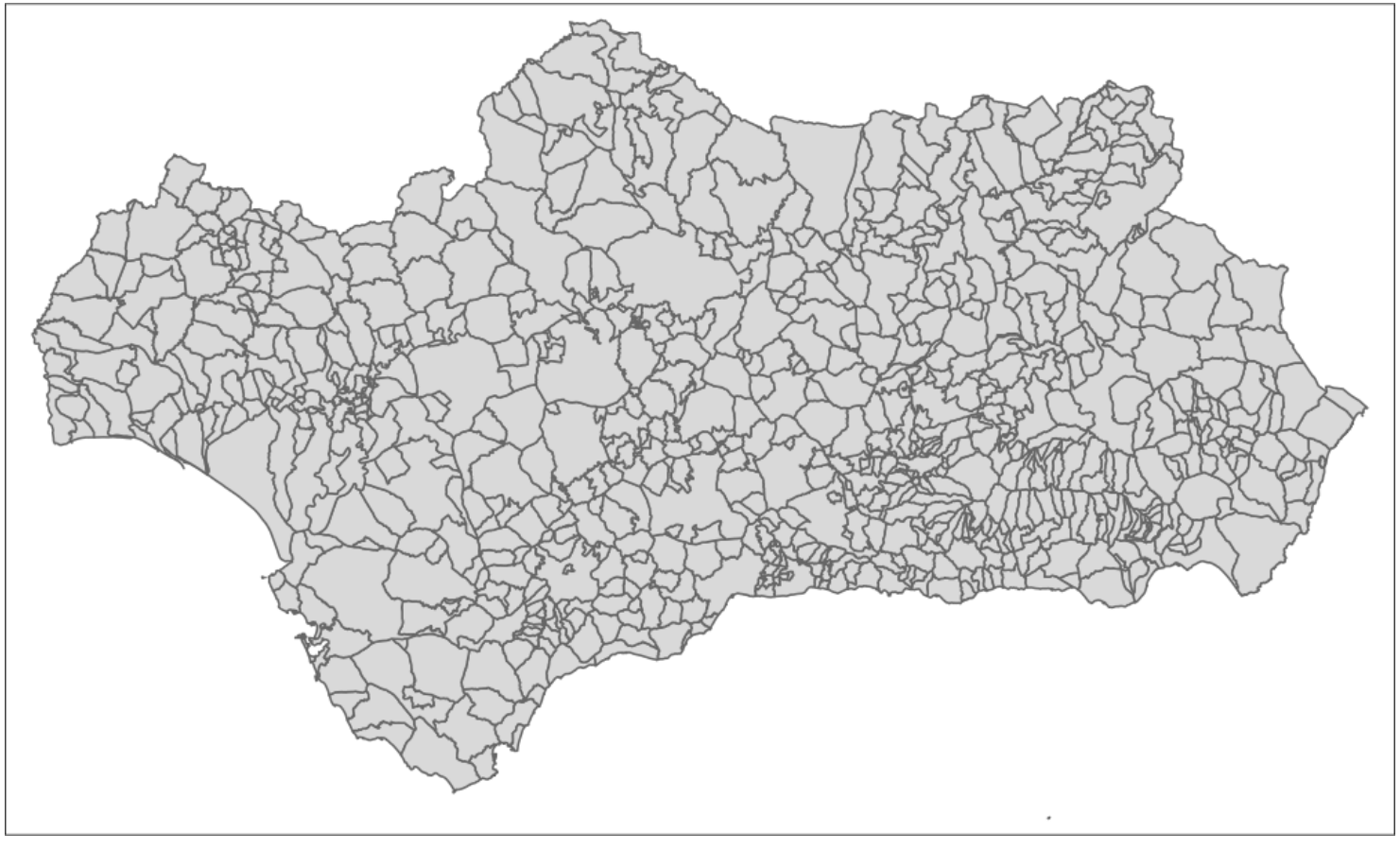
```
 [1] "AeronavFAA"     "AmigoCloud"     "ARCGEN"         "AVCBin"
 [5] "AVCE00"         "BNA"            "CAD"            "Carto"
 [9] "Cloudant"       "CouchDB"        "CSV"            "CSW"
[13] "DGN"            "DXF"            "EDIGEO"         "EEDA"
[17] "ElasticSearch"  "ESRI Shapefile" "ESRIJSON"       "Geoconcept"
[21] "GeoJSON"        "GeoJSONSeq"     "Geomedia"       "GeoRSS"
[25] "GFT"            "GML"            "GMLAS"          "GPKG"
[29] "GPSBabel"       "GPSTrackMaker"  "GPX"            "HTF"
[33] "HTTP"           "Idrisi"         "Interlis 1"     "Interlis 2"
[37] "JML"            "JP2OpenJPEG"    "KML"            "LIBKML"
[41] "MapInfo File"   "MBTiles"        "Memory"         "MSSQLSpatial"
[45] "MVT"            "MySQL"          "NAS"            "netCDF"
[49] "NGW"            "ODBC"           "ODS"            "OGR_DODS"
[53] "OGR_GMT"        "OGR_OGDI"       "OGR_PDS"        "OGR_SDTS"
[57] "OGR_VRT"        "OpenAir"        "OpenFileGDB"    "OSM"
[61] "PCIDSK"         "PDF"            "PDS4"           "PGDUMP"
[65] "PGeo"           "PLSCENES"       "PostgreSQL"     "REC"
[69] "S57"            "SEGUKOOA"       "SEGY"           "Selafin"
[73] "SOSI"           "SQLite"         "SUA"            "SVG"
[77] "SXF"            "TIGER"          "TopoJSON"       "UK .NTF"
[81] "VDV"            "VFK"            "Walk"           "WAsP"
[85] "WFS"            "WFS3"           "XLS"            "XLSX"
[89] "XPlane"
```

Read polygon layer (municipalities)

```
munis <- st_read("data/municipios.gpkg")
```

```
Reading layer `municipios' from data source
  `/home/frs/Dropbox/Rcode/myRcode/courses_talks/course-using-R-as-GIS/data/municipios.gpkg'
  using driver `GPKG'
Simple feature collection with 778 features and 4 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 100401.7 ymin: 3977033 xmax: 621287.7 ymax: 4288703
Projected CRS: ETRS89 / UTM zone 30N
```

```
tmap_mode("plot")

tm_shape(munis) +
  tm_polygons()
```



Simplifying polygon layer

```
object.size(munis)
```

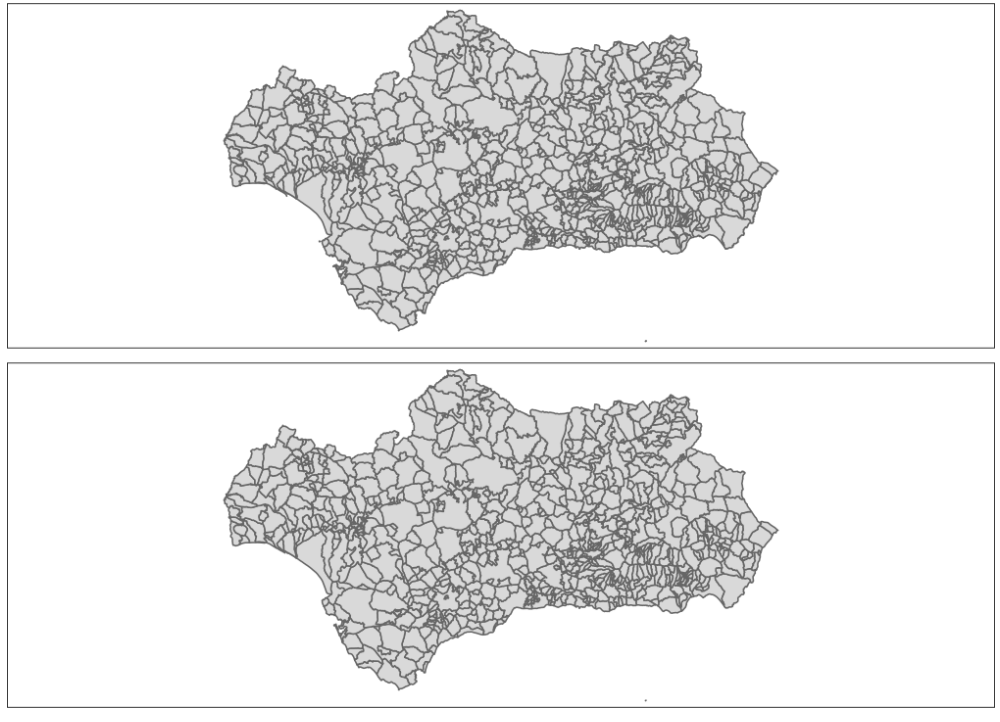
```
7910624 bytes
```

```
muni.lite <- tmaptools::simplify_shape(munis, keep.units = TRUE)
object.size(muni.lite)
```

```
1539952 bytes
```

Original vs Simplified layer

```
orig <- tm_shape(munis) +
  tm_polygons()

simple <- tm_shape(muni.lite) +
  tm_polygons()

tmap_arrange(orig, simple)
```



Save simplified layer

```
sf::st_write(muni.lite, "data/muni.lite.gpkg")
```

To save as shapefile:

```
sf::st_write(muni.lite, "data/muni.lite.shp")
```

Let's join polygon layer (town boundaries)
with point layer containing COVID data
for each town

To join a spatial (sf) object with plain dataframe, use normal join

```
covid.plain <- covid.sf %>%
  st_drop_geometry() %>%
  rename(MUNICIPIO = Municipio)

left_join(muni.lite, covid.plain)
```

```
Simple feature collection with 778 features and 8 fields
Geometry type: GEOMETRY
Dimension:     XY
Bounding box:  xmin: 100402.7 ymin: 3977033 xmax: 621287.7 ymax: 4288703
Projected CRS: ETRS89 / UTM zone 30N
First 10 features:
   COD_MUN                MUNICIPIO PROVINCIA COD_ENT Provincia Poblacion Casos
1    11004                Algeciras     Cádiz     d06     Cádiz    123078 11945
2    11007                  Barbate     Cádiz     d06     Cádiz     22556  2369
3    11035                   Tarifa     Cádiz     d06     Cádiz     18183  1542
4    11033                San Roque     Cádiz     d06     Cádiz     31571  3000
5    11008            Barrios (Los)     Cádiz     d06     Cádiz     23777  2164
6    11014     Conil de la Frontera     Cádiz     d06     Cádiz     22775  2243
7    11013 Castellar de la Frontera     Cádiz     d06     Cádiz      3057   300
8    11901     Benalup-Casas Viejas     Cádiz     d06     Cádiz      6986   664
9    29068                  Manilva    Málaga     d06    Málaga     16439  1540
10   11039     Vejer de la Frontera     Cádiz     d06     Cádiz     12622  1051
   Fallecidos                             geom
1         264 MULTIPOLYGON (((275250.5 39...
2          27 MULTIPOLYGON (((249488.7 40...
3          30 MULTIPOLYGON (((257731.7 40...
4          50 MULTIPOLYGON (((297785.7 40...
5          80 MULTIPOLYGON (((281015.8 40...
```

'muni.lite' and 'covid.sf' are both sf objects

```
head(muni.lite)
```

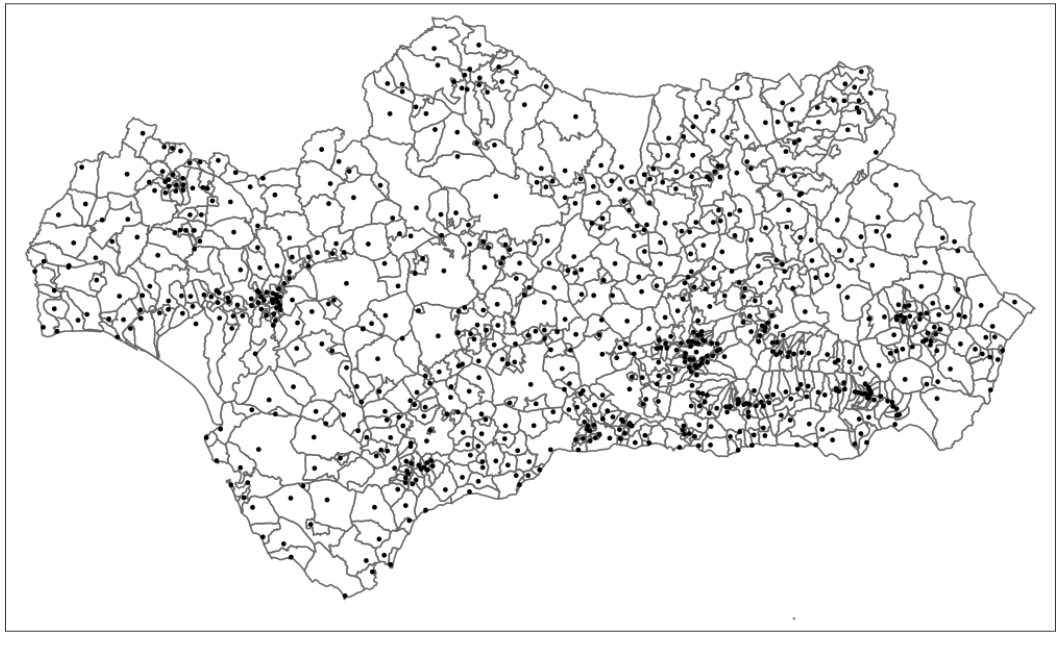
```
Simple feature collection with 6 features and 4 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 216157.5 ymin: 3987198 xmax: 297785.7 ymax: 4026730
Projected CRS: ETRS89 / UTM zone 30N
  COD_MUN             MUNICIPIO PROVINCIA COD_ENT                            geom
1   11004             Algeciras     Cádiz     d06 MULTIPOLYGON (((275250.5 39...
2   11007               Barbate     Cádiz     d06 MULTIPOLYGON (((249488.7 40...
3   11035                Tarifa     Cádiz     d06 MULTIPOLYGON (((257731.7 40...
4   11033             San Roque     Cádiz     d06 MULTIPOLYGON (((297785.7 40...
5   11008         Barrios (Los)     Cádiz     d06 MULTIPOLYGON (((281015.8 40...
6   11014 Conil de la Frontera     Cádiz     d06 MULTIPOLYGON (((230475.7 40...
```

```
head(covid.sf)
```

```
Simple feature collection with 6 features and 5 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 222693 ymin: 3988986 xmax: 285306 ymax: 4019422
Projected CRS: ETRS89 / UTM zone 30N
# A tibble: 6 × 6
  Provincia Municipio            Pobla…¹ Casos Falle…²         geometry
  <chr>     <chr>                  <dbl> <dbl>   <dbl>      <POINT [m]>
1 Cádiz     Algeciras             123078 11945     264 (279223 4001392)
2 Cádiz     Barbate                22556  2369      27 (237228 4009013)
3 Cádiz     Tarifa                 18183  1542      30 (265096 3988986)
```

Plot both objects

```
tmap_mode("plot")

tm_shape(muni.lite) +
  tm_polygons(col = "white") +
  tm_shape(covid.sf) +
  tm_dots()
```



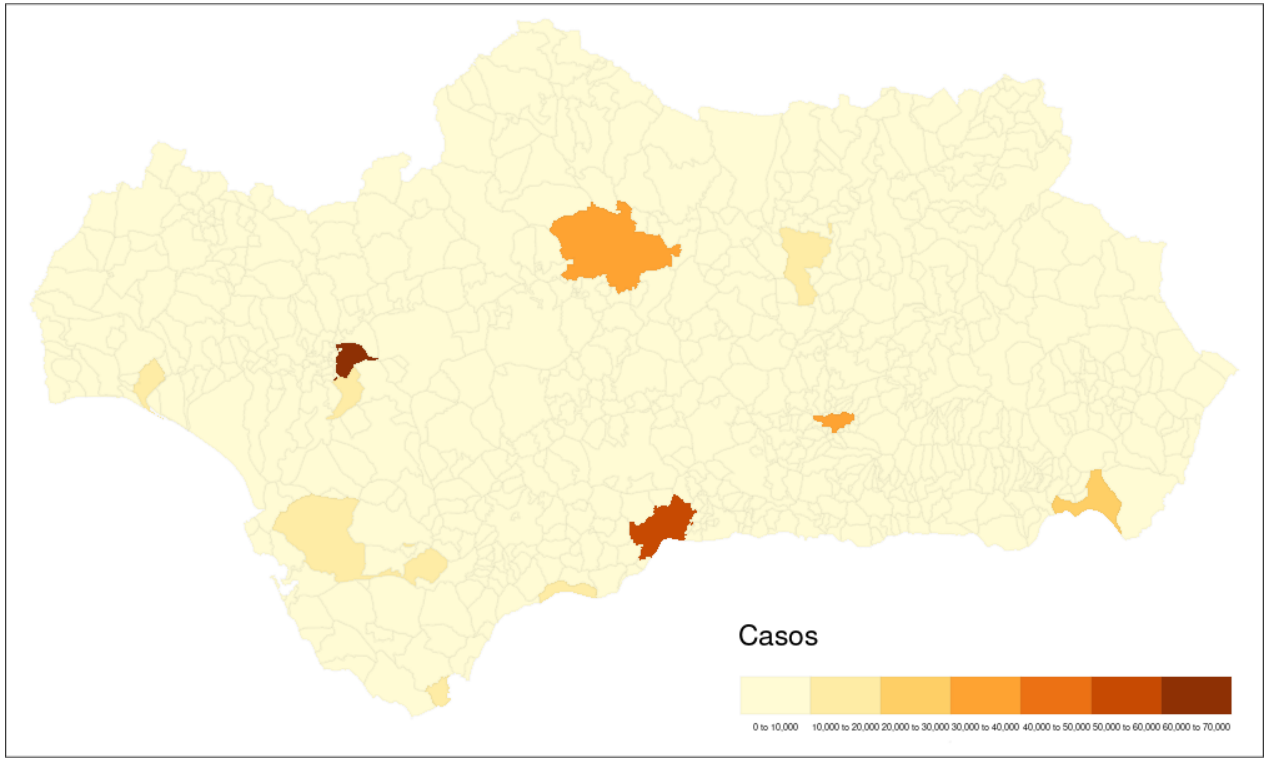
Spatial join: join data that intersects spatially

```
covid.town <- st_join(muni.lite, covid.sf)
```

```
Simple feature collection with 778 features and 9 fields
Geometry type: GEOMETRY
Dimension:     XY
Bounding box:  xmin: 100402.7 ymin: 3977033 xmax: 621287.7 ymax: 4288703
Projected CRS: ETRS89 / UTM zone 30N
First 10 features:
   COD_MUN                MUNICIPIO PROVINCIA COD_ENT Provincia
1    11004                Algeciras     Cádiz     d06     Cádiz
2    11007                  Barbate     Cádiz     d06     Cádiz
3    11035                   Tarifa     Cádiz     d06     Cádiz
4    11033                San Roque     Cádiz     d06     Cádiz
5    11008            Barrios (Los)     Cádiz     d06     Cádiz
6    11014     Conil de la Frontera     Cádiz     d06     Cádiz
7    11013 Castellar de la Frontera     Cádiz     d06     Cádiz
8    11901     Benalup-Casas Viejas     Cádiz     d06     Cádiz
9    29068                  Manilva    Málaga     d06    Málaga
10   11039     Vejer de la Frontera     Cádiz     d06     Cádiz
                  Municipio Poblacion Casos Fallecidos
1                 Algeciras    123078 11945        264
2                   Barbate     22556  2369         27
3                    Tarifa     18183  1542         30
4                 San Roque     31571  3000         50
5             Barrios (Los)     23777  2164         80
6      Conil de la Frontera     22775  2243         10
7  Castellar de la Frontera      3057   300          4
8      Benalup-Casas Viejas      6986   664          6
9                    Manilva     16439  1540         22
10     Vejer de la Frontera     12622  1051         10
                            geom
1  MULTIPOLYGON (((275250.5 39...
2  MULTIPOLYGON (((249488.7 40...
3  MULTIPOLYGON (((257731.7 40...
4  MULTIPOLYGON (((297785.7 40...
5  MULTIPOLYGON (((281015.8 40...
6  MULTIPOLYGON (((230475.7 40...
7  MULTIPOLYGON (((288527.6 40...
8  MULTIPOLYGON (((248920.1 40...
9  MULTIPOLYGON (((300755.7 40...
```

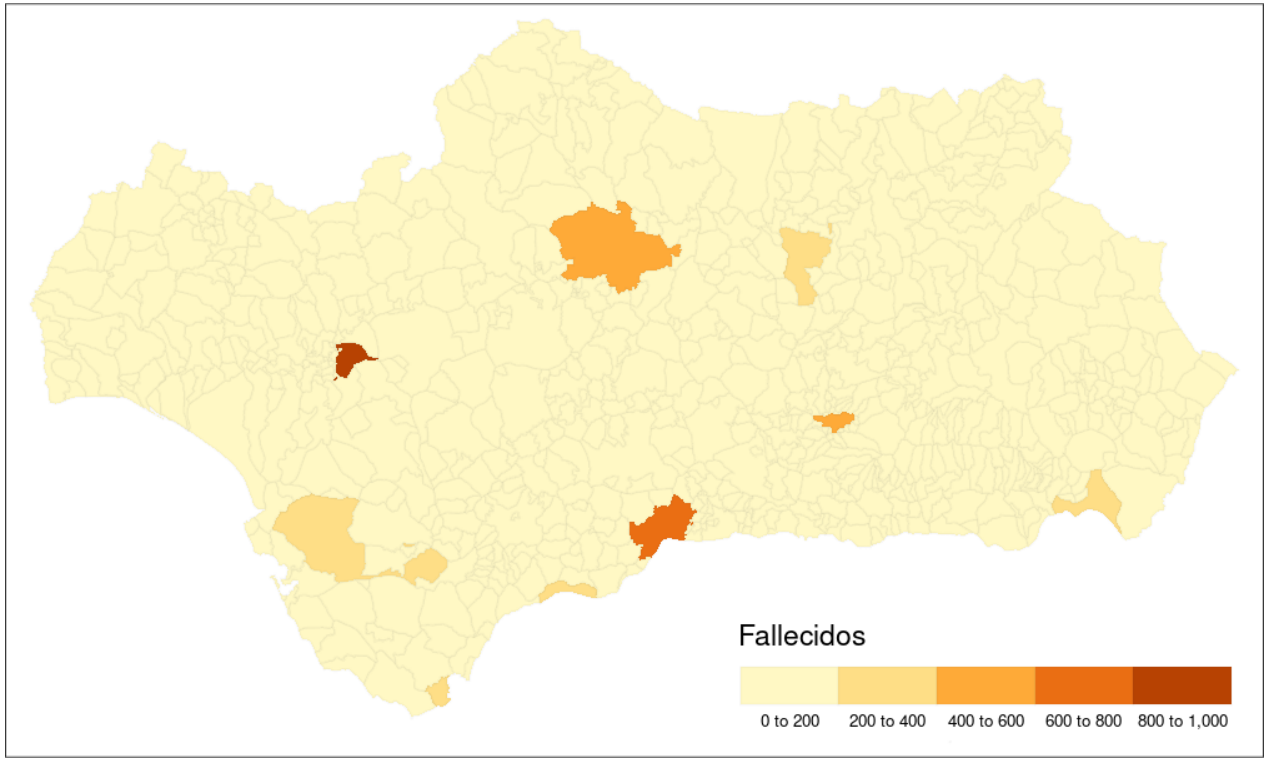
Map of the joined data

```
tm_shape(covid.town) +
  tm_polygons(col = "Casos", border.alpha = 0.05, legend.is.portrait = FALSE)
```



Map of the joined data

```
tm_shape(covid.town) +
  tm_polygons(col = "Fallecidos", border.alpha = 0.05, legend.is.portrait = FALSE)
```



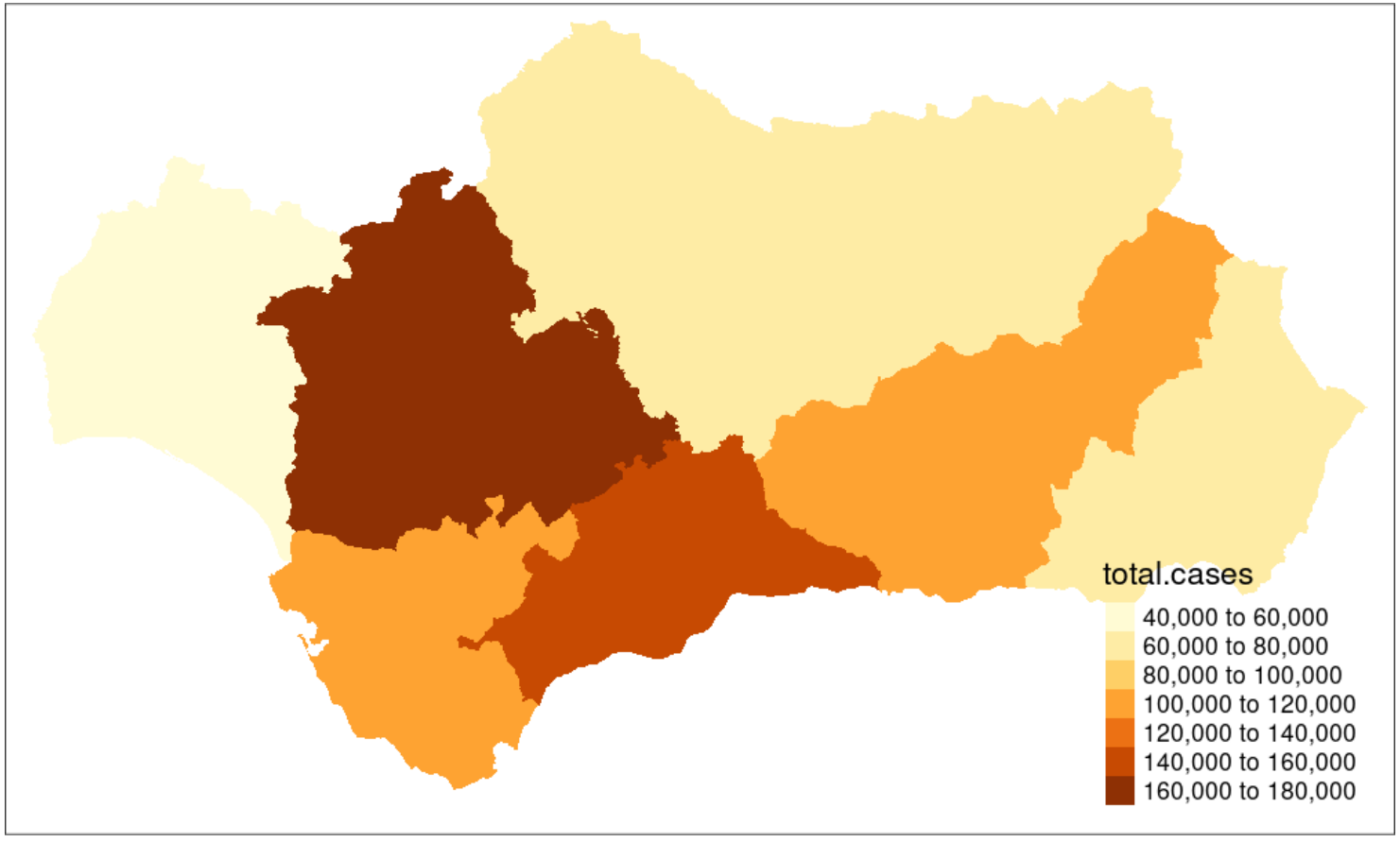
Number of cases per province

```
cases.province <- covid.town %>%
  group_by(PROVINCIA) %>%
  summarise(total.cases = sum(Casos, na.rm = TRUE))
cases.province
```

```
Simple feature collection with 8 features and 2 fields
Geometry type: GEOMETRY
Dimension:     XY
Bounding box:  xmin: 100402.7 ymin: 3977033 xmax: 621287.7 ymax: 4288703
Projected CRS: ETRS89 / UTM zone 30N
# A tibble: 8 × 3
  PROVINCIA total.cases                                                geom
  <chr>           <dbl>                                      <GEOMETRY [m]>
1 Almería         71015 MULTIPOLYGON (((497200.8 3977463, 496827 3977033, 49666…
2 Cádiz          111338 POLYGON ((281383.7 4005784, 280887.8 4005281, 280460.8 …
3 Córdoba         72573 MULTIPOLYGON (((367733.7 4122381, 367873.7 4121546, 367…
4 Granada        109313 POLYGON ((468710.7 4064932, 468198.4 4064130, 467599.4 …
5 Huelva          46321 POLYGON ((199053.7 4092965, 199721.6 4092060, 200461.6 …
6 Jaén            64770 POLYGON ((428507.9 4145982, 429047.9 4145468, 428788.9 …
7 Málaga         156953 POLYGON ((300755.7 4027668, 300058.7 4025889, 299585.7 …
8 Sevilla        175681 POLYGON ((237751.7 4082485, 236729.7 4082650, 235996.7 …
```

Number of cases per province

```
tm_shape(cases.province) +
  tm_fill(col = "total.cases")
```



Your turn

- Join polygon with point layer
- Map deaths per town (polygons)
- Map deaths per province (polygons)

Spatial operations with sf objects

Get borders of Andalucia

st_union combines geometries (dissolve polygons)

```
andal <- st_union(muni.lite)
andal
```

```
Geometry set for 1 feature
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 100402.7 ymin: 3977033 xmax: 621287.7 ymax: 4288703
Projected CRS: ETRS89 / UTM zone 30N
```

Get borders of Andalucia

```
tm_shape(andal) +
  tm_borders()
```

Get borders of Andalucia

```
andal <- muni.lite %>%
  summarise()
andal
```

```
Simple feature collection with 1 feature and 0 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 100402.7 ymin: 3977033 xmax: 621287.7 ymax: 4288703
Projected CRS: ETRS89 / UTM zone 30N
                              geom
1 MULTIPOLYGON (((497200.8 39...
```

Get borders of Andalucia

```
tm_shape(andal) +
  tm_borders() +
  tm_compass()
```

Is Grazalema town within Sevilla province?

```
grazalema <- muni.lite %>%
  filter(MUNICIPIO == "Grazalema")
grazalema
```

```
Simple feature collection with 1 feature and 4 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: 276808.4 ymin: 4066496 xmax: 295547.5 ymax: 4077397
Projected CRS: ETRS89 / UTM zone 30N
  COD_MUN MUNICIPIO PROVINCIA COD_ENT                            geom
1   11019 Grazalema     Cádiz     d06 MULTIPOLYGON (((295390.6 40...
```

Is Grazalema town within Sevilla province?

```
sevilla <- cases.province %>%
  filter(PROVINCIA == "Sevilla")
sevilla
```

```
Simple feature collection with 1 feature and 2 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: 187708.5 ymin: 4081268 xmax: 353380.2 ymax: 4231238
Projected CRS: ETRS89 / UTM zone 30N
# A tibble: 1 × 3
  PROVINCIA total.cases                                                geom
  <chr>            <dbl>                                  <POLYGON [m]>
1 Sevilla         175681 ((237751.7 4082485, 236729.7 4082650, 235996.7 4082466,…
```

Is Grazalema town within Sevilla province?

```
st_within(grazalema, sevilla, sparse = FALSE)
```

```
      [,1]
[1,] FALSE
```

Does Grazalema share border with Sevilla province?

```
st_touches(grazalema, sevilla, sparse = FALSE)
```

```
      [,1]
[1,] FALSE
```

Is Grazalema within 10 km of Sevilla province?

```
st_is_within_distance(grazalema, sevilla, dist = 10000, sparse = FALSE)
```

```
      [,1]
[1,] FALSE
```

Is Grazalema within 50 km of Sevilla province?

```
st_is_within_distance(grazalema, sevilla, dist = 50000, sparse = FALSE)
```

```
     [,1]
[1,] TRUE
```

Distance between Grazalema and Sevilla province border

```
st_distance(grazalema, sevilla)
```

```
Units: [m]
          [,1]
[1,] 11324.91
```

Distance between Grazalema and Sevilla town

```
sevilla.town <- covid.sf %>%
  filter(Municipio == "Sevilla (capital)")
sevilla.town
```

```
Simple feature collection with 1 feature and 5 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: 237881 ymin: 4141973 xmax: 237881 ymax: 4141973
Projected CRS: ETRS89 / UTM zone 30N
# A tibble: 1 × 6
  Provincia Municipio         Poblacion Casos Fallec…¹          geometry
  <chr>     <chr>                 <dbl> <dbl>    <dbl>       <POINT [m]>
1 Sevilla   Sevilla (capital)    691395 61273      886 (237881 4141973)
# … with abbreviated variable name ¹Fallecidos
```

Distance between Grazalema and Sevilla town

```
st_distance(grazalema, sevilla.town)
```

```
Units: [m]
          [,1]
[1,] 77441.29
```

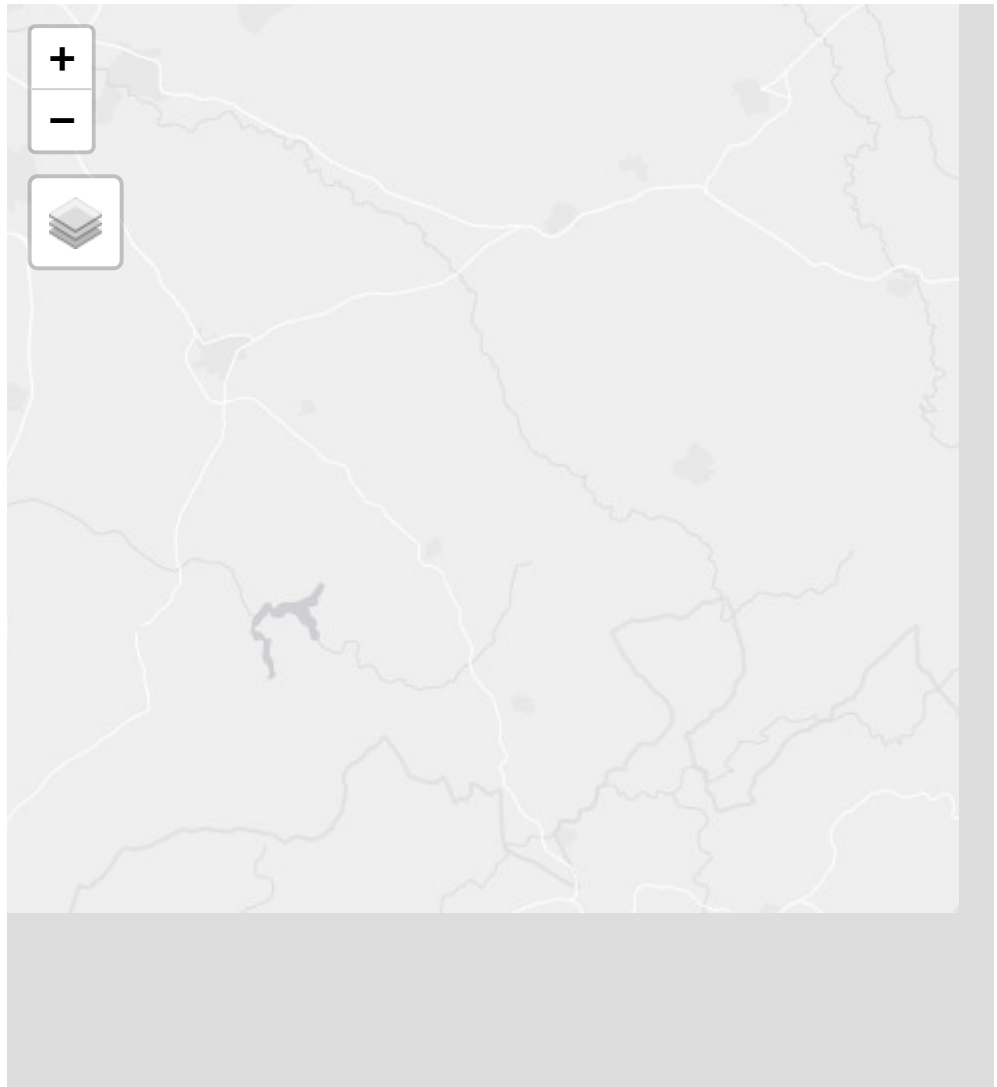
Define metropolitan area around Sevilla

```
sev.muni <- muni.lite %>%
  filter(MUNICIPIO == "Sevilla (capital)")

sev.metro <- st_buffer(sev.muni, dist = 20000)
sev.metro
```

```
Simple feature collection with 1 feature and 4 fields
Geometry type: POLYGON
Dimension:     XY
Bounding box:  xmin: 211192.6 ymin: 4112522 xmax: 270515.3 ymax: 4169206
Projected CRS: ETRS89 / UTM zone 30N
  COD_MUN         MUNICIPIO PROVINCIA COD_ENT                            geom
1   41091 Sevilla (capital)   Sevilla     d06 POLYGON ((212311.1 4139709,...
```

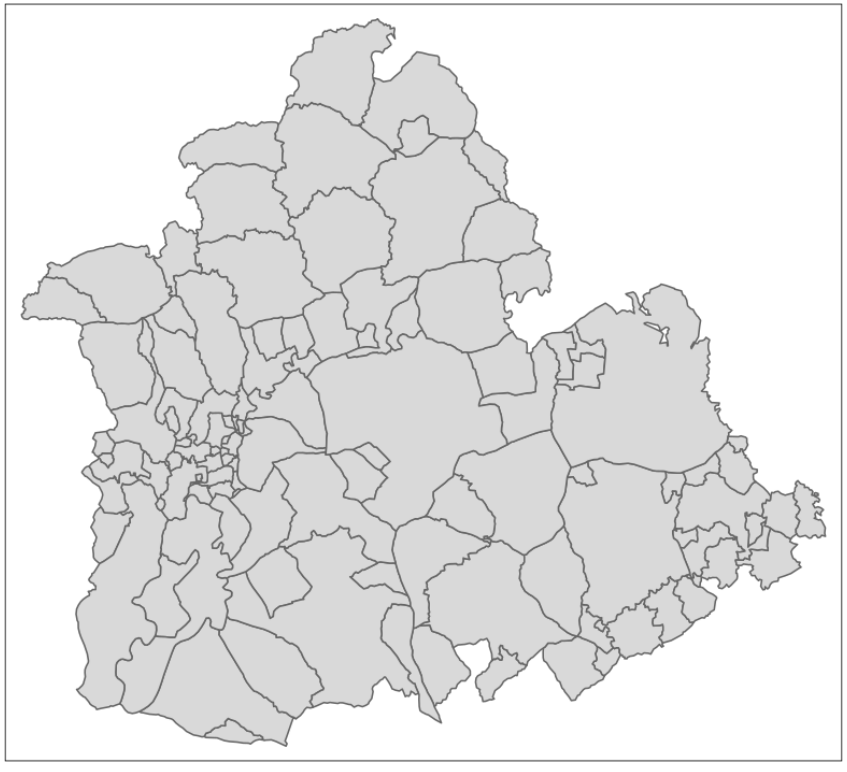
Define metropolitan area around Sevilla

```
tmap_mode("view")

tm_shape(sev.metro) +
  tm_fill() +
  tm_shape(sev.muni) +
  tm_fill(col = "black")

tmap_mode("plot")
```



Find which towns intersect metropolitan area

```
sev.munis <- muni.lite %>%
  filter(PROVINCIA == "Sevilla")

tm_shape(sev.munis) +
  tm_polygons()
```



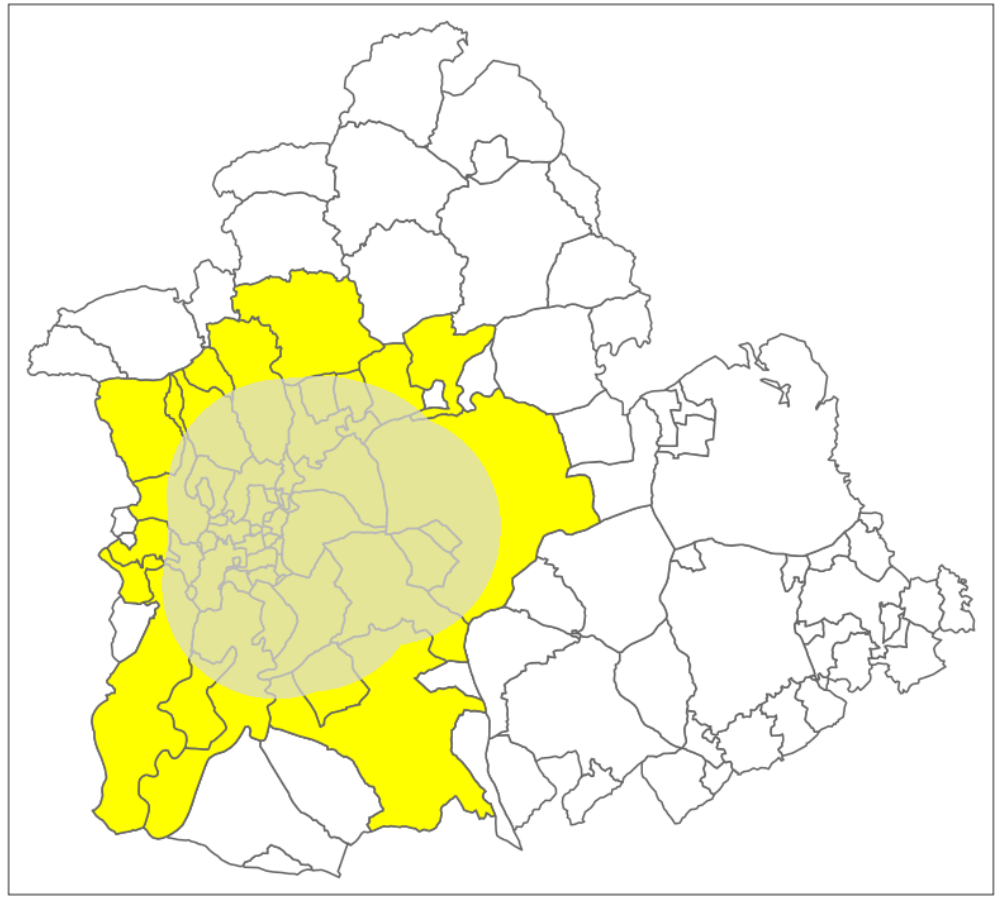
Find which towns intersect metropolitan area

```
intersect.metro <- st_join(sev.munis, sev.metro, left = FALSE)
intersect.metro
```

```
Simple feature collection with 49 features and 8 fields
Geometry type: GEOMETRY
Dimension:     XY
Bounding box:  xmin: 198938.3 ymin: 4088004 xmax: 287791.5 ymax: 4187978
Projected CRS: ETRS89 / UTM zone 30N
First 10 features:
   COD_MUN.x                  MUNICIPIO.x PROVINCIA.x COD_ENT.x COD_MUN.y
14     41902                   Isla Mayor     Sevilla       d06     41091
15     41069 Palacios y Villafranca (Los)     Sevilla       d06     41091
21     41095                       Utrera     Sevilla       d06     41091
26     41079          Puebla del Río (La)     Sevilla       d06     41091
27     41034                Coria del Río     Sevilla       d06     41091
29     41010                   Almensilla     Sevilla       d06     41091
31     41070            Palomares del Río     Sevilla       d06     41091
32     41012                   Aznalcázar     Sevilla       d06     41091
35     41044                       Gelves     Sevilla       d06     41091
36     41075                        Pilas     Sevilla       d06     41091
         MUNICIPIO.y PROVINCIA.y COD_ENT.y                           geom
14 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((215176.6 41...
15 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((233926.2 41...
21 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((260985.4 41...
26 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((226413.6 41...
27 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((230498.2 41...
29 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((226055.7 41...
31 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((231061.2 41...
32 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((215888.7 41...
```

Find which towns intersect metropolitan area

```
tm_shape(sev.munis) +
  tm_polygons(col = "white") +
  tm_shape(intersect.metro) +
  tm_polygons(col = "yellow") +
  tm_shape(sev.metro) +
  tm_fill(alpha = 0.7)
```



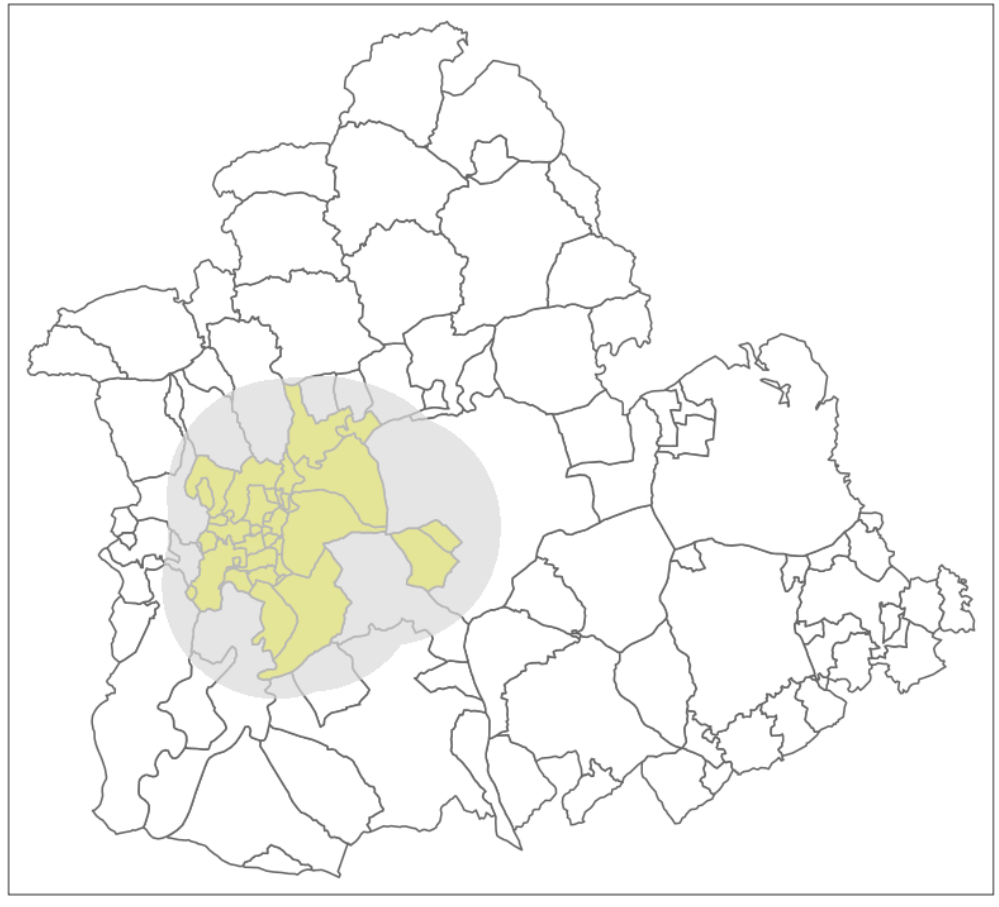
Find which towns fall within metropolitan area

```
within.metro <- st_join(sev.munis, sev.metro,
                        join = st_within, left = FALSE)
within.metro
```

```
Simple feature collection with 29 features and 8 fields
Geometry type: GEOMETRY
Dimension:     XY
Bounding box:  xmin: 215192.1 ymin: 4115884 xmax: 263682.4 ymax: 4167872
Projected CRS: ETRS89 / UTM zone 30N
First 10 features:
   COD_MUN.x              MUNICIPIO.x PROVINCIA.x COD_ENT.x COD_MUN.y
27     41034            Coria del Río     Sevilla       d06     41091
29     41010               Almensilla     Sevilla       d06     41091
31     41070        Palomares del Río     Sevilla       d06     41091
35     41044                   Gelves     Sevilla       d06     41091
37     41059     Mairena del Aljarafe     Sevilla       d06     41091
40     41016 Bollullos de la Mitación     Sevilla       d06     41091
41     41086 San Juan de Aznalfarache     Sevilla       d06     41091
42     41093                  Tomares     Sevilla       d06     41091
46     41017                 Bormujos     Sevilla       d06     41091
48     41094                  Umbrete     Sevilla       d06     41091
         MUNICIPIO.y PROVINCIA.y COD_ENT.y                           geom
27 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((230498.2 41...
29 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((226055.7 41...
31 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((231061.2 41...
35 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((224523.8 41...
37 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((231321.9 41...
40 Sevilla (capital)     Sevilla       d06 POLYGON ((223818.4 4139661,...
41 Sevilla (capital)     Sevilla       d06 MULTIPOLYGON (((232258.6 41...
```

Find which towns fall within metropolitan area

```
tm_shape(sev.munis) +
  tm_polygons(col = "white") +
  tm_shape(within.metro) +
  tm_polygons(col = "yellow") +
  tm_shape(sev.metro) +
  tm_fill(alpha = 0.7)
```



Your turn

Meteorological stations

- Download shp of stations from https://www.miteco.gob.es/es/cartografia-y-sig/ide/descargas/otros/default.aspx
- Map towns with a station within their borders (`st_join`)
- Map number of stations per province (`summarise`)
- Find nearest meteorological station for 'Algar' town (`st_distance`/`st_nearest_point`)

Parks & Natural areas

- Download https://www.juntadeandalucia.es/institutodeestadisticaycartografia/DERA/g07.htm
- Calculate & Map area of parks for each town (`st_join`)

Distance to highways

- Download data from https://www.juntadeandalucia.es/institutodeestadisticaycartografia/DERA/g09.htm
- Find distance from one/all towns to nearest highway
- Plot facetted histogram of distances per province

Airports

- Download data from https://www.juntadeandalucia.es/institutodeestadisticaycartografia/DERA/g09.htm
- Map towns within buffer of 10 km around airports (`st_buffer` & `st_join`)
- Calculate area of each town included within the buffer (`st_area`)

Campings within flooding areas

- Download flooding risk layers from https://www.miteco.gob.es/es/cartografia-y-sig/ide/descargas/agua/ARPSI.aspx
- Download campings (e.g. from https://portalrediam.cica.es/geonetwork/srv/eng/catalog.search#/metadata/e6f9aac9-eadb-4cdf-888d-a04c68909f34)
- Map flooding risks & campings

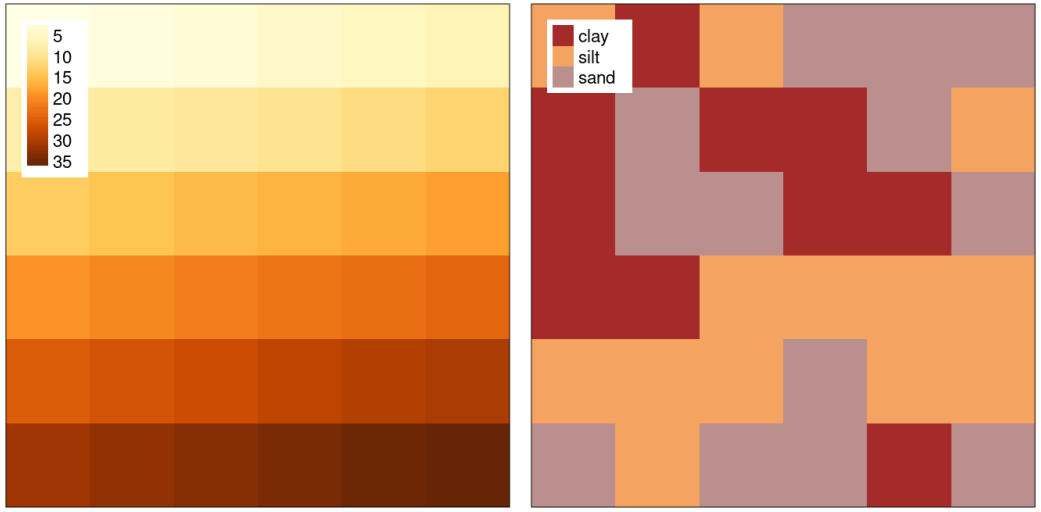
Spatial raster data with 'terra' package

Francisco Rodríguez-Sánchez

https://frodriguezsanchez.net

@frod_san

Raster data

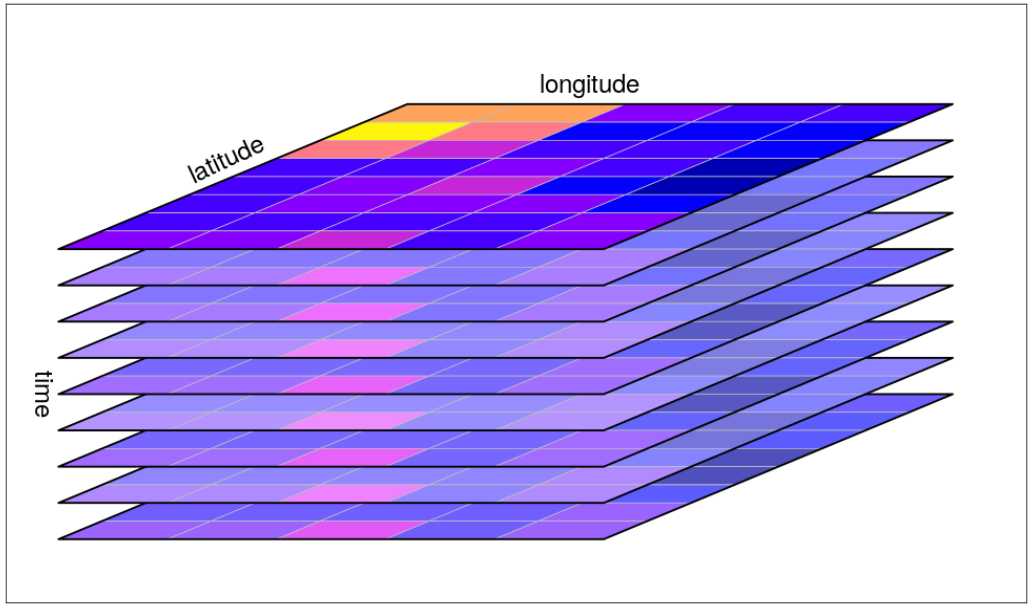


Source: Geocomputation with R

terra & stars packages

https://rspatial.github.io/terra/index.html

https://r-spatial.github.io/stars/



Read raster data

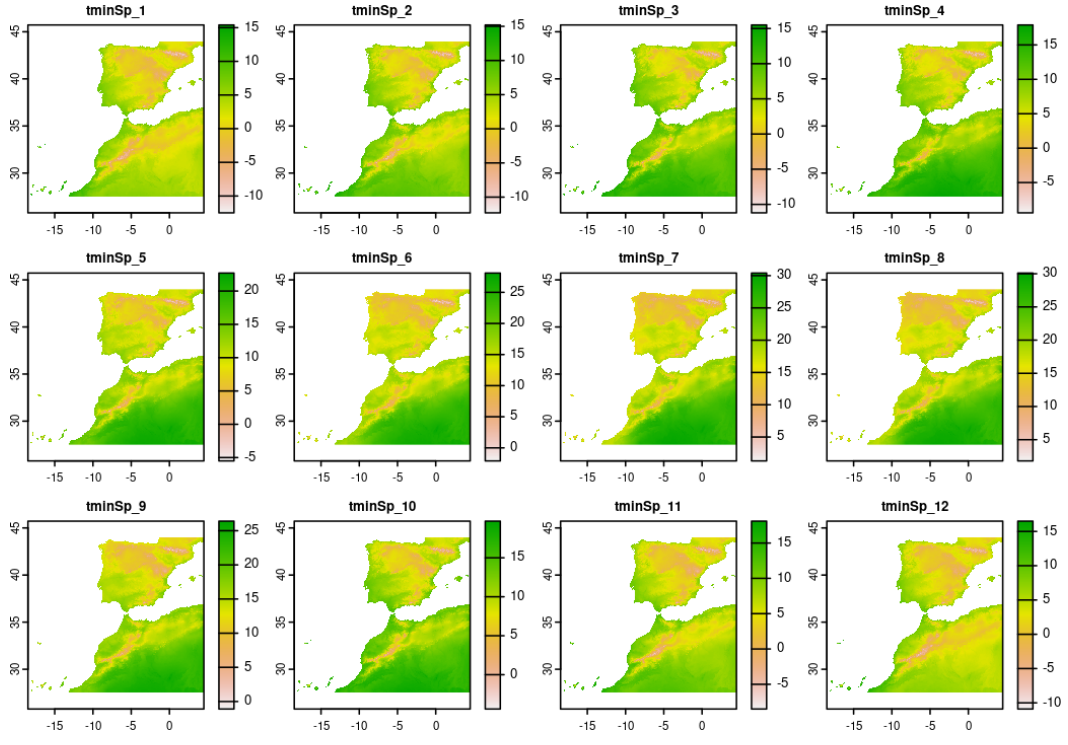
```
library(terra)
tmin.sp <- rast("data/tminSp.tif")
tmin.sp
```

```
class       : SpatRaster
dimensions  : 1980, 2760, 12  (nrow, ncol, nlyr)
resolution  : 0.008333333, 0.008333333  (x, y)
extent      : -18.5, 4.5, 27.5, 44  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (EPSG:4326)
source      : tminSp.tif
names       : tminSp_1, tminSp_2, tminSp_3, tminSp_4, tminSp_5, tminSp_6, ...
min values  :    -12.6,    -12.5,    -11.4,     -9.6,     -5.7,     -2.2, ...
max values  :     17.4,     17.2,     17.2,     18.0,     22.8,     28.2, ...
```

Plotting rasters with terra

Plot raster (multiple layers)

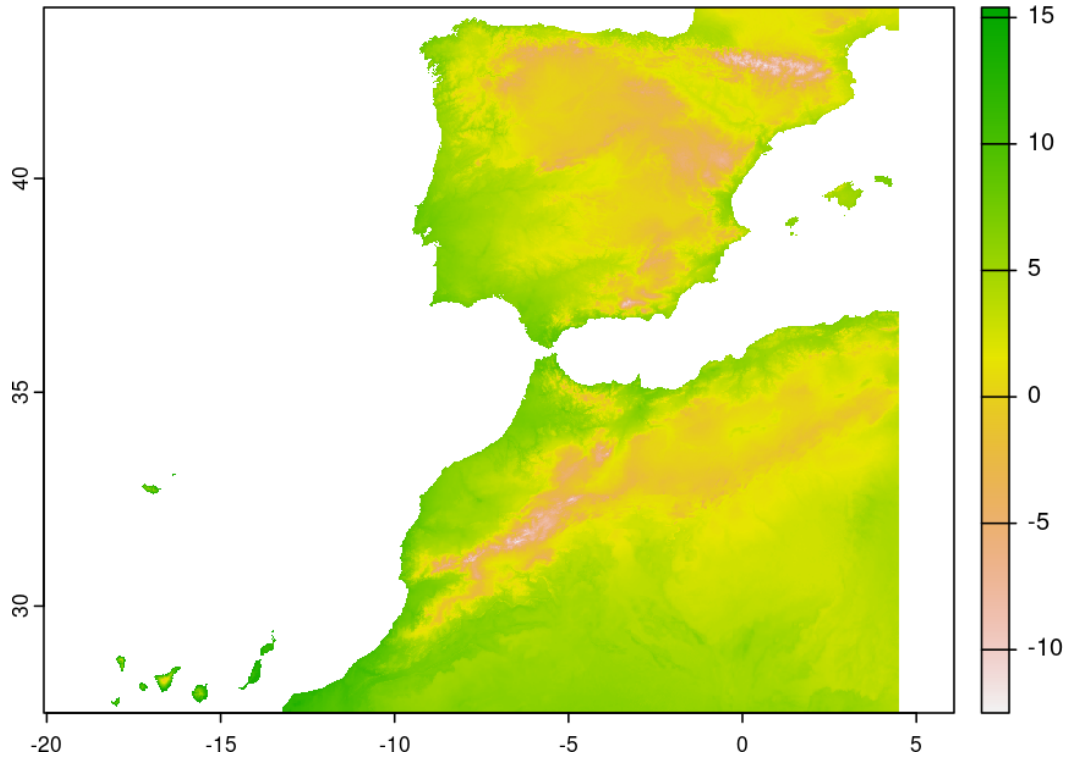
```
plot(tmin.sp)
```



Plot raster (single layer)

Tmin January

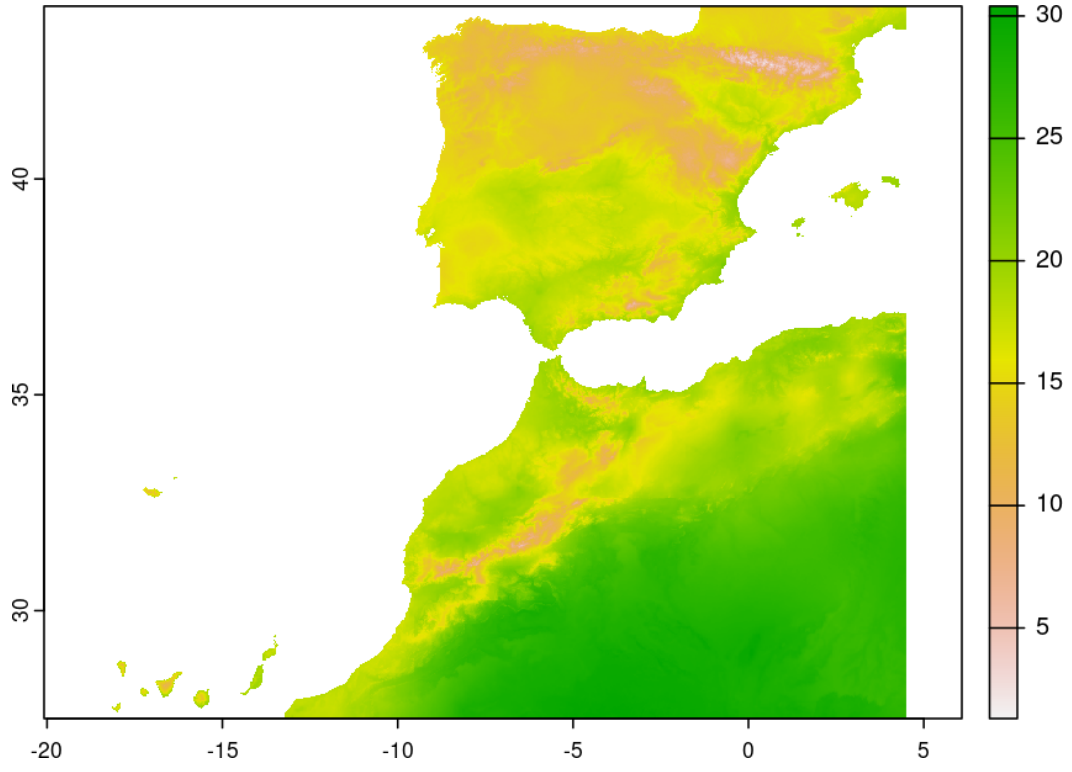
```
plot(tmin.sp, y = 1)
```



Plot raster (single layer)

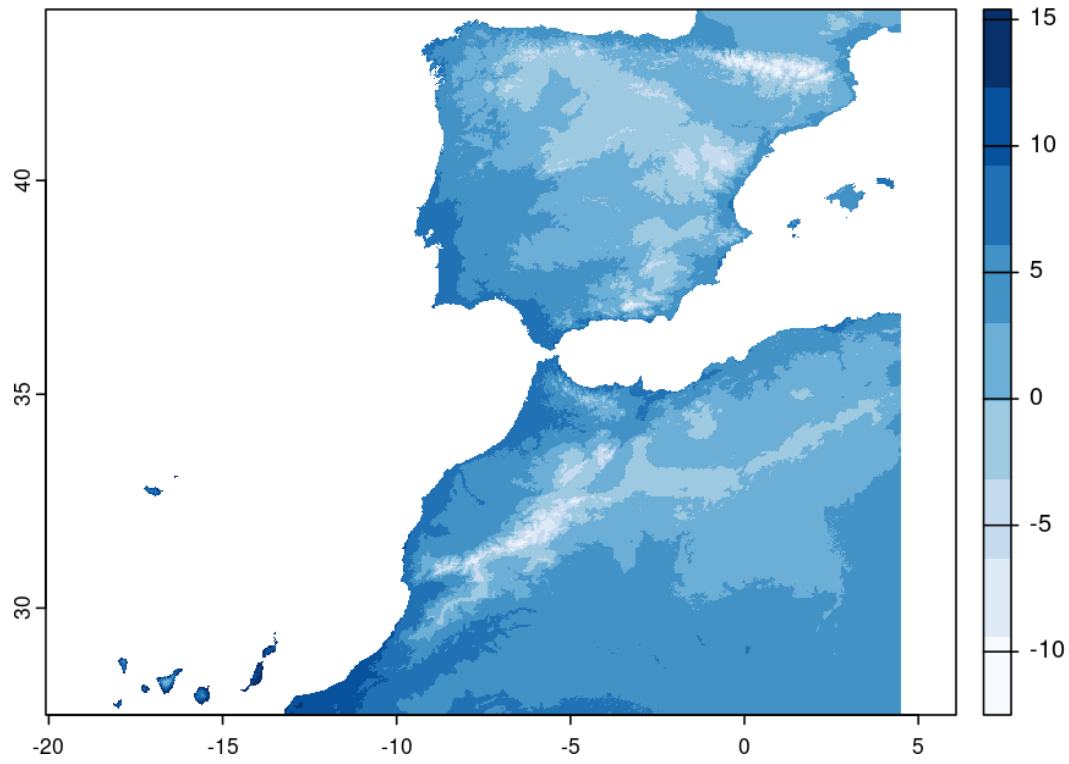
Tmin July

```
plot(tmin.sp, y = 7)
```



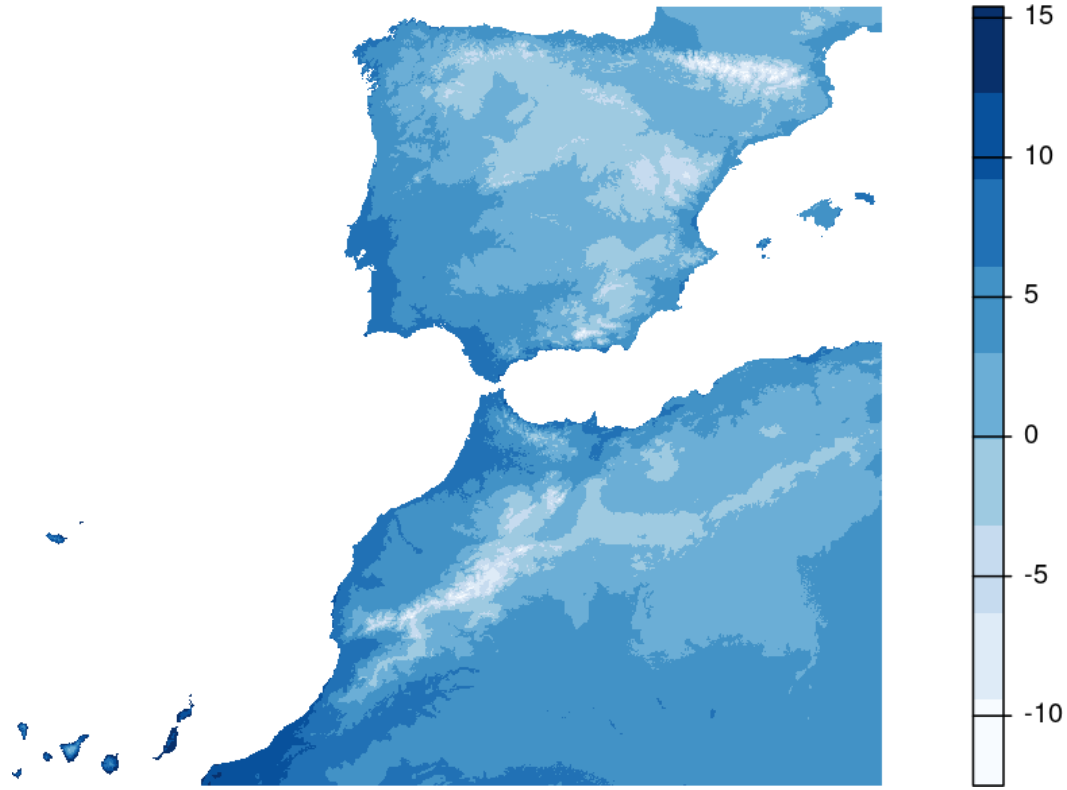
Set colours

```
library(RColorBrewer)
plot(tmin.sp, y = 1,
     col = brewer.pal(9, "Blues"))
```



Remove axes?

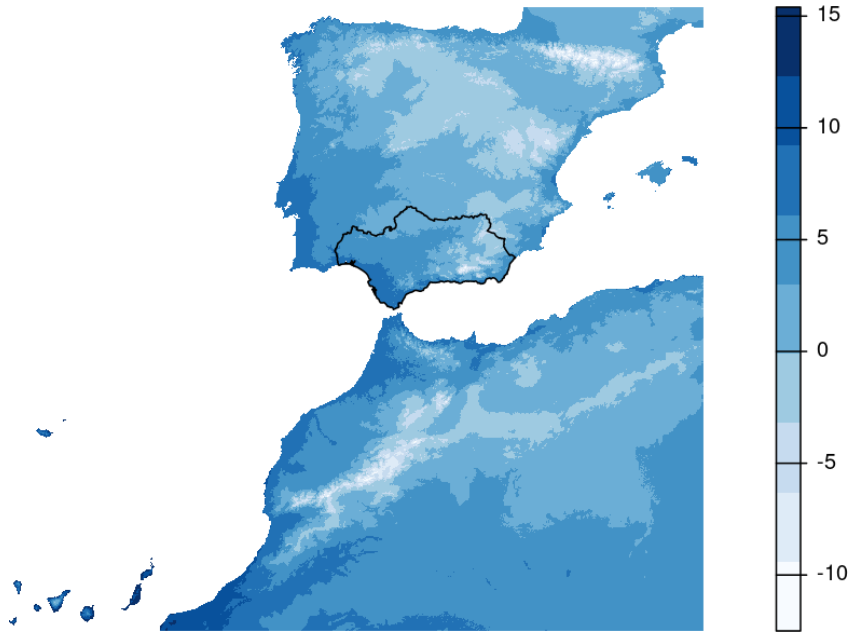
```
plot(tmin.sp, y = 1,
     col = RColorBrewer::brewer.pal(9, "Blues"),
     axes = FALSE)
```



Adding vector shapes

Loading sf, transforming to vect

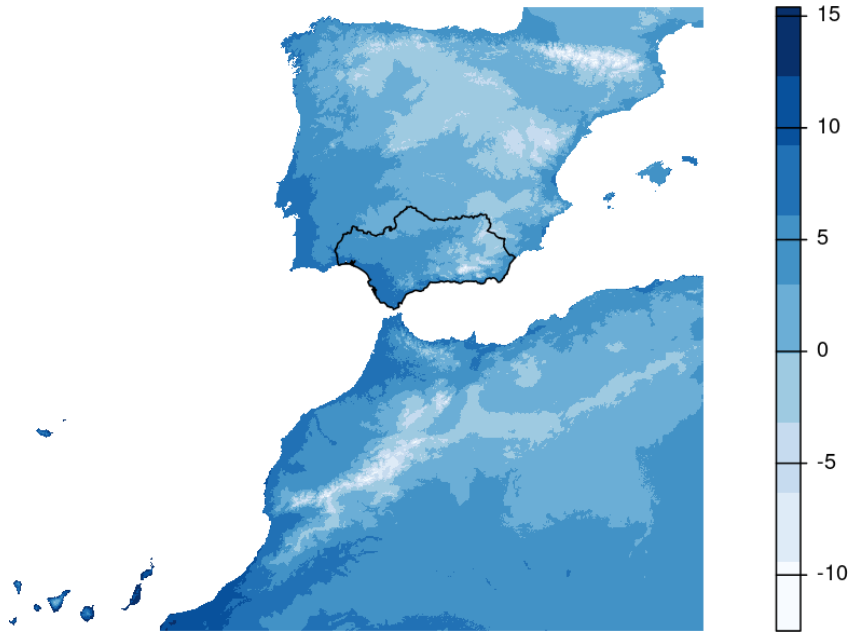
```
andal.sf <- sf::st_read("data/Andalucia_contorno.gpkg", quiet = TRUE)
plot(tmin.sp, y = 1, axes = FALSE,
     col = RColorBrewer::brewer.pal(9, "Blues"))
plot(vect(andal.sf), add = TRUE)
```



Adding vector shapes

Reading vect directly

```
andal <- vect("data/Andalucia_contorno.gpkg")
plot(tmin.sp, y = 1, axes = FALSE,
     col = RColorBrewer::brewer.pal(9, "Blues"))
plot(andal, add = TRUE)
```



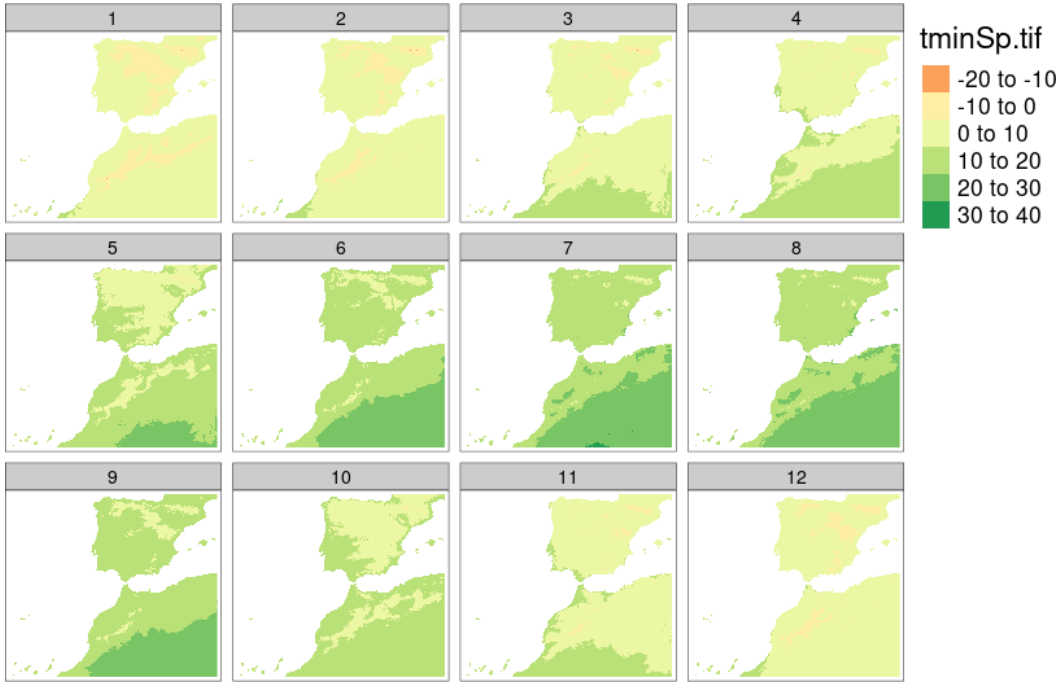
Save these plots

```
pdf("mymap.pdf")

plot(tmin.sp, y = 1, col = brewer.pal(9, "Blues"))

dev.off()
```

Plotting rasters with tmap

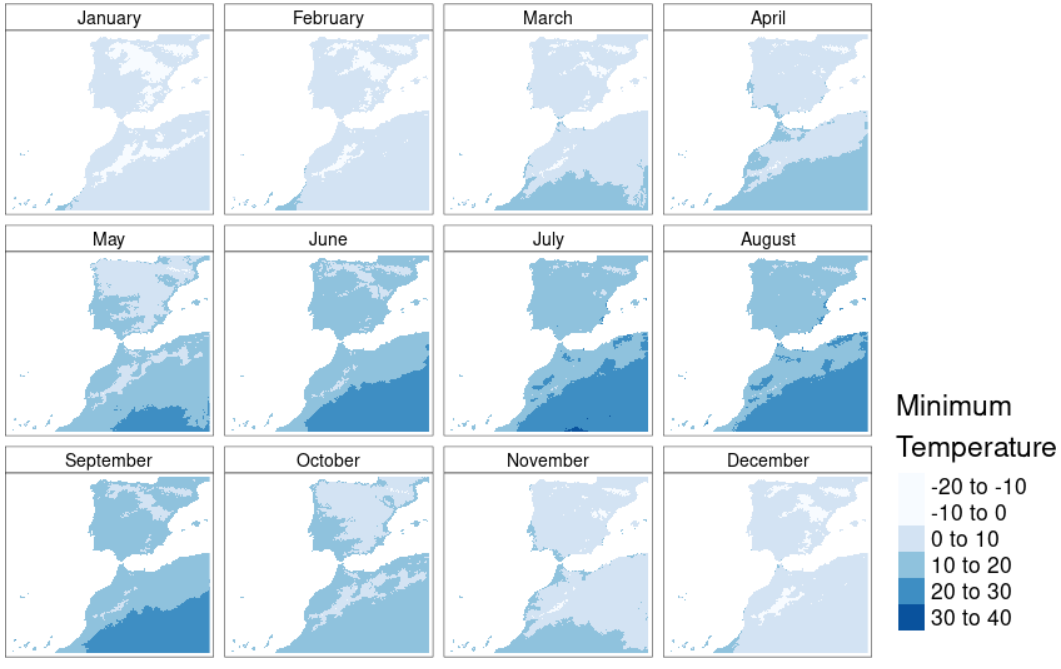
Plotting rasters with tmap

```
library(tmap)
tm_shape(tmin.sp) +
  tm_raster()
```



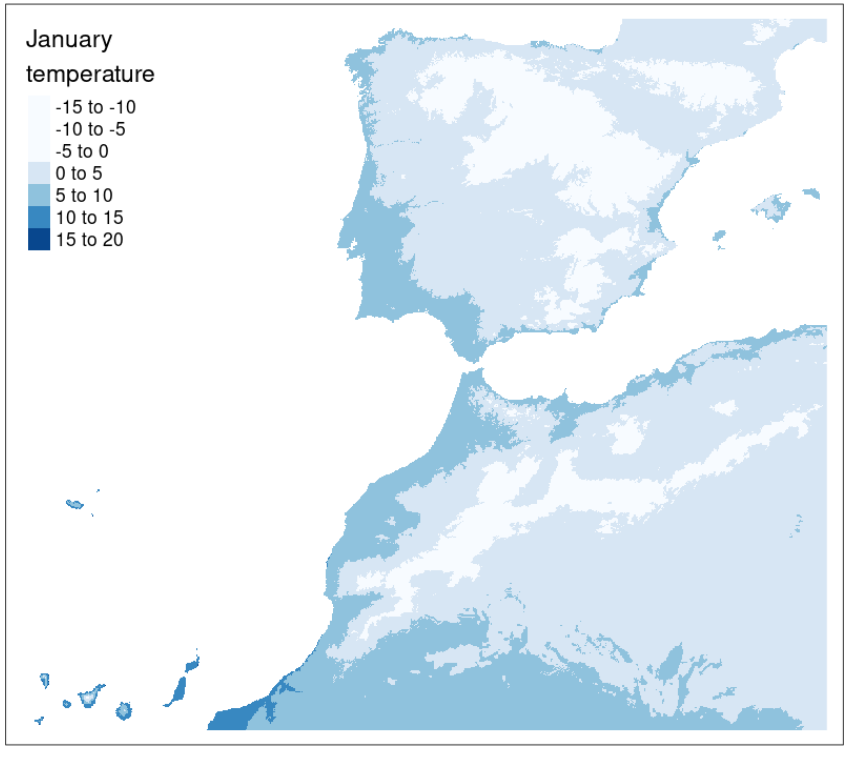
Plotting rasters with tmap

```
tm_shape(tmin.sp) +
  tm_raster(palette = "Blues",
            title = "Minimum\nTemperature") +
  tm_layout(legend.position = c(0.05, 0.03),
            panel.label.bg.color = "white",
            panel.labels = month.name)
```



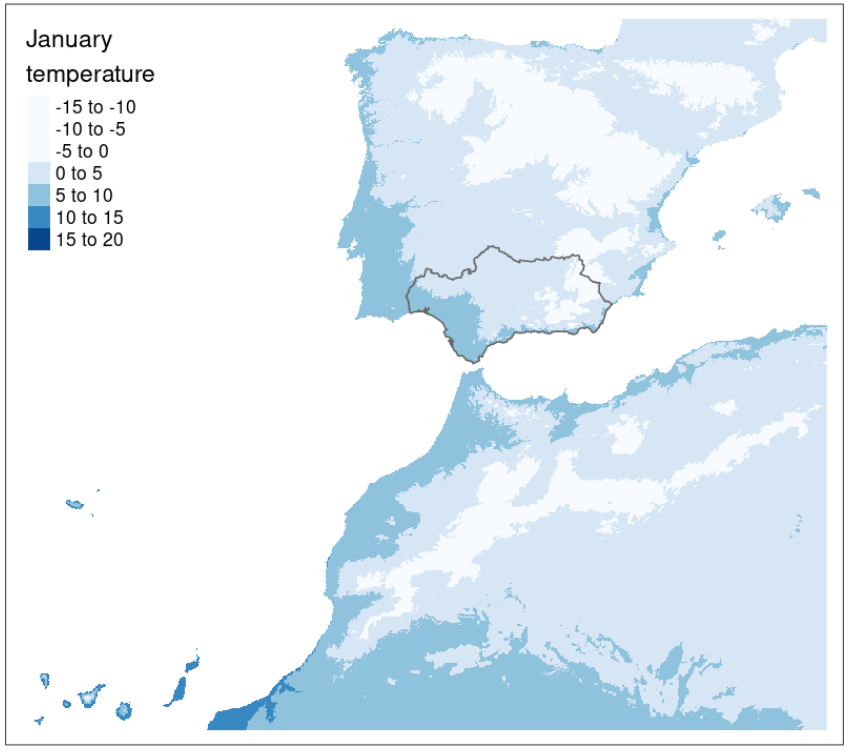
Plotting rasters with tmap

```
tmin.jan <- rast("data/tmin.jan.tif")
tm_shape(tmin.jan) +
  tm_raster(palette = "Blues", title = "January\ntemperature")
```



Adding vector shapes

```
tm_shape(tmin.jan) +
  tm_raster(palette = "Blues", title = "January\ntemperature") +
  tm_shape(andal.sf) +
  tm_borders()
```



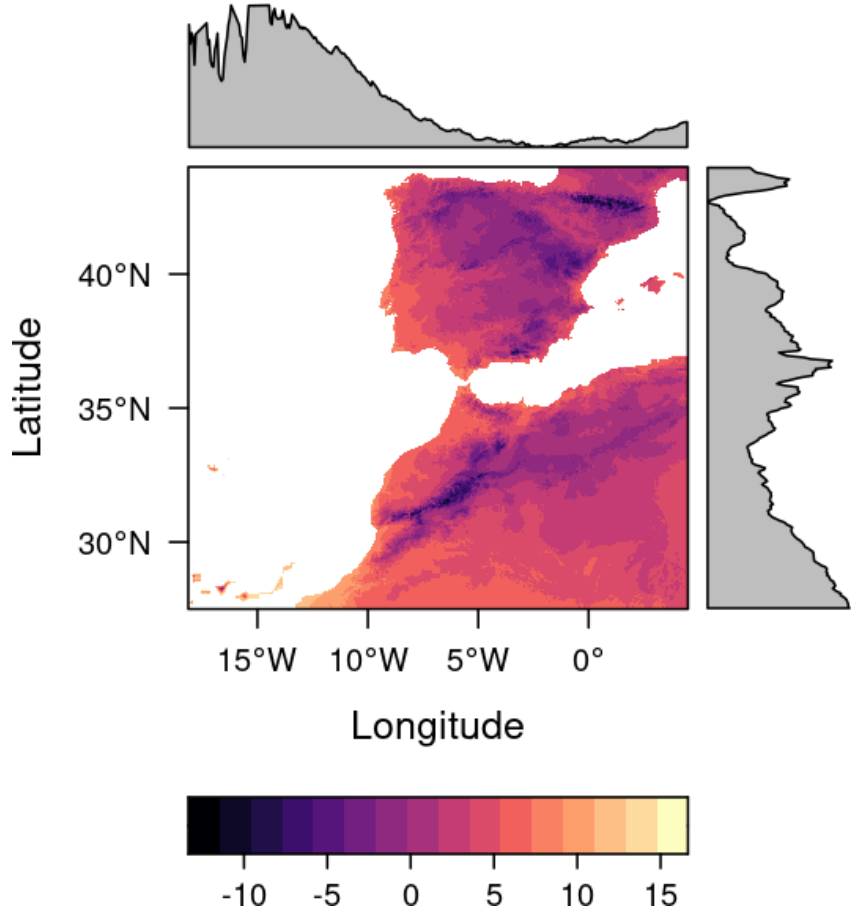
Saving tmap figures

```
map <- tm_shape(tmin.jan) +
  tm_raster(palette = "Blues", title = "January\ntemperature")

tmap_save(map, "mytmap.png")
```

Plotting rasters with rasterVis

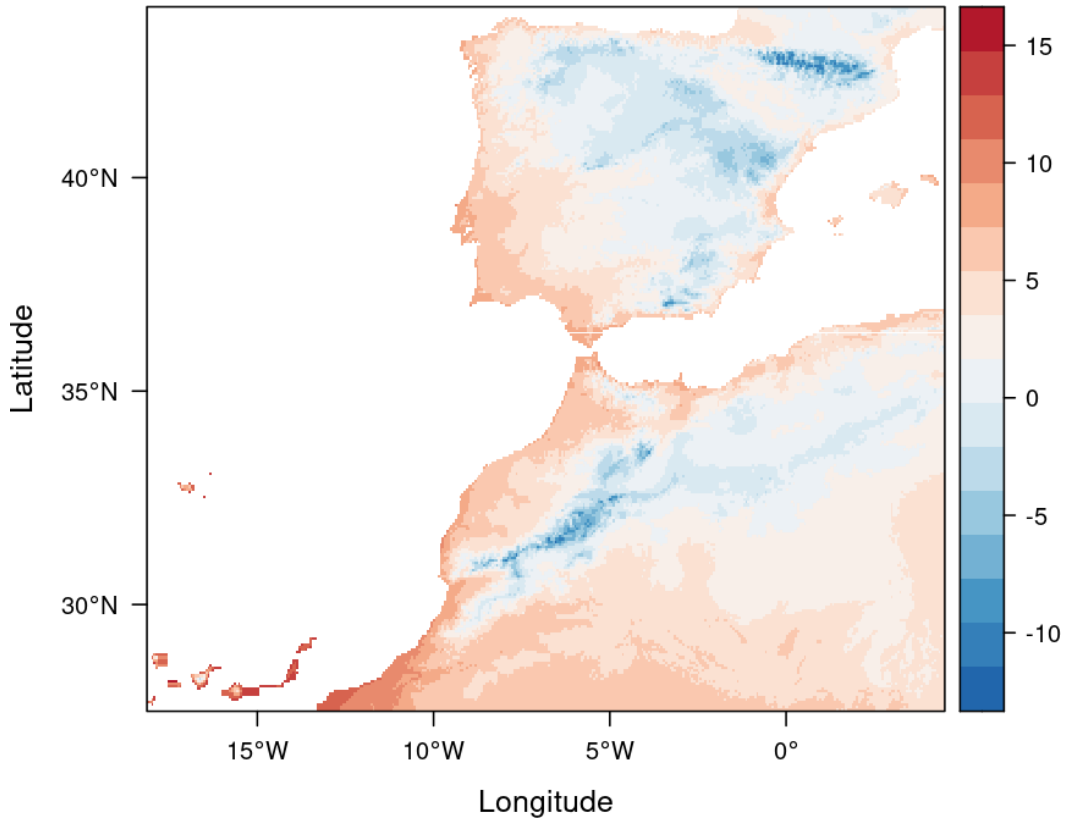
Plotting rasters with rasterVis

```
library(rasterVis)
levelplot(tmin.jan)
```



Plotting rasters with rasterVis

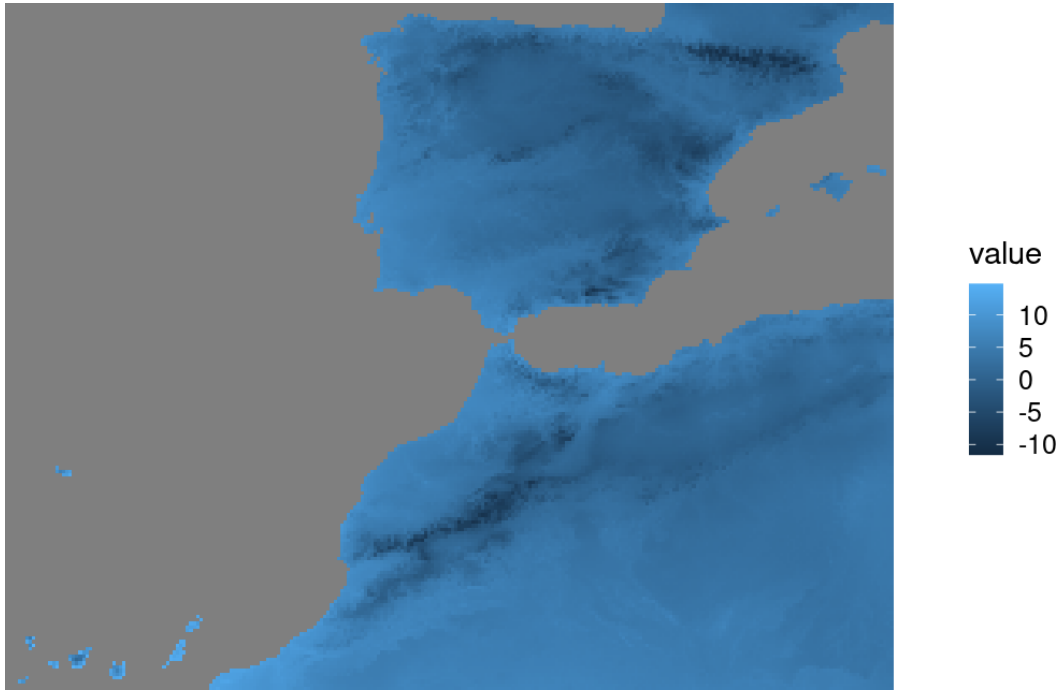
```
levelplot(tmin.jan, margin = FALSE, par.settings = BuRdTheme())
```



Plotting rasters with rasterVis & ggplot2

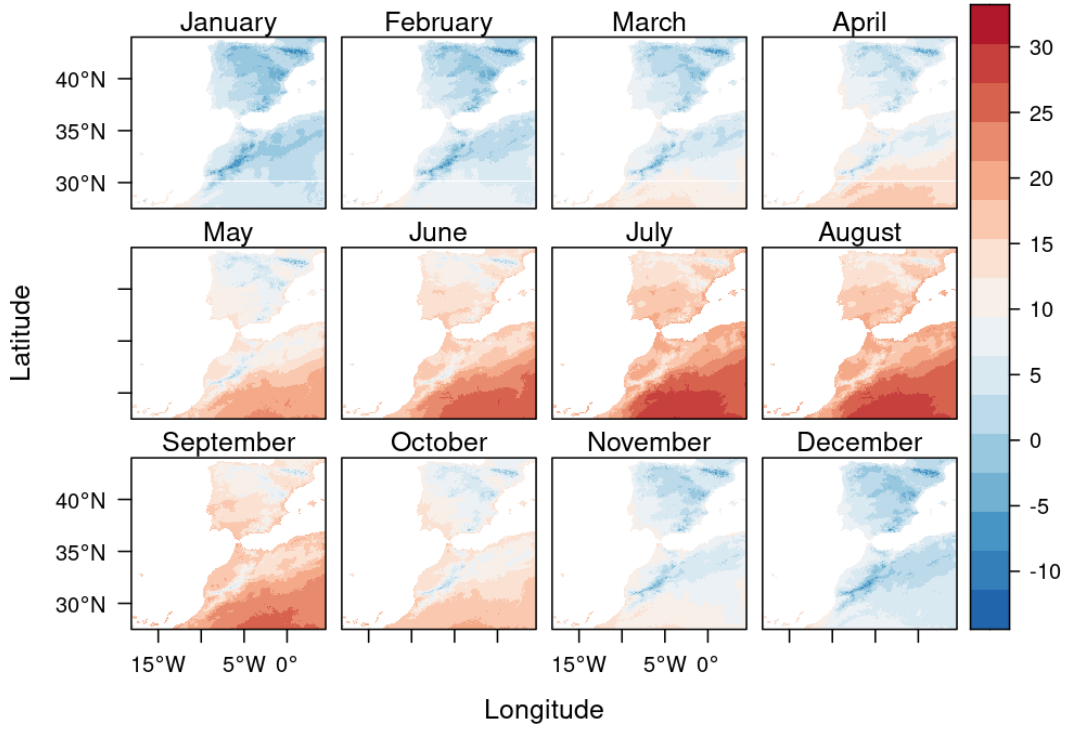
Note `gplot` rather than `ggplot`

```
library(ggplot2)
gplot(tmin.jan) +
  geom_tile(aes(fill = value)) +
  cowplot::theme_map()
```



Plotting rasters with rasterVis

```
names(tmin.sp) <- month.name
levelplot(tmin.sp, par.settings = BuRdTheme())
```



Your turn

Make a map of minimum temperature for July

Raster algebra

Download Tmax

```
library(geodata)
tmax.sp <- worldclim_country(country = "Spain", var = "tmax", path = "data/")
tmax.sp
```

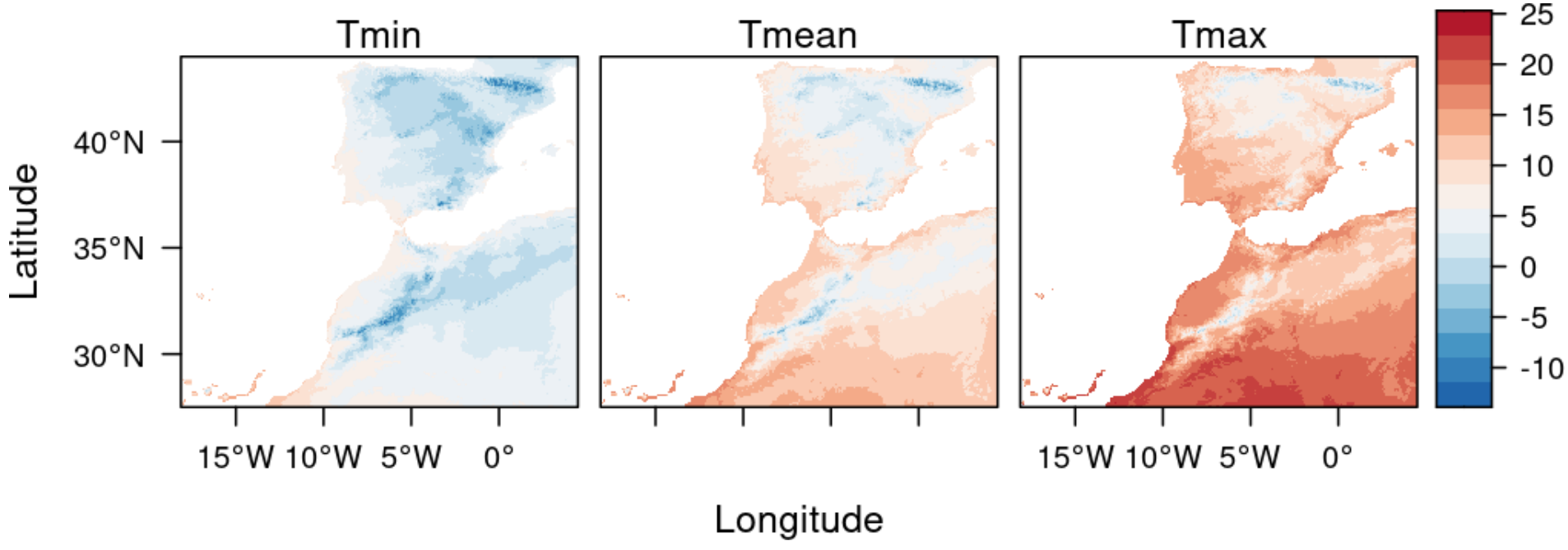
```
class       : SpatRaster
dimensions  : 1980, 2760, 12  (nrow, ncol, nlyr)
resolution  : 0.008333333, 0.008333333  (x, y)
extent      : -18.5, 4.5, 27.5, 44  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (EPSG:4326)
source      : ESP_wc2.1_30s_tmax.tif
names       : ESP_w~max_1, ESP_w~max_2, ESP_w~max_3, ESP_w~max_4, ESP_w~max_5, ESP_w~max_6, ...
min values  :        -6.9,        -6.3,        -4.0,        -2.3,         2.9,         8.7, ...
max values  :        23.1,        25.3,        29.7,        34.2,        38.7,        44.5, ...
```

Calculate Tmean for January

```
tmin.jan <- subset(tmin.sp, 1)

tmax.jan <- subset(tmax.sp, 1)

tmean.jan <- mean(tmin.jan, tmax.jan)
```

Combine and map

```
tjan <- c(tmin.jan, tmean.jan, tmax.jan)
names(tjan) <- c("Tmin", "Tmean", "Tmax")
levelplot(tjan, layout = c(3, 1), par.settings = BuRdTheme())
```



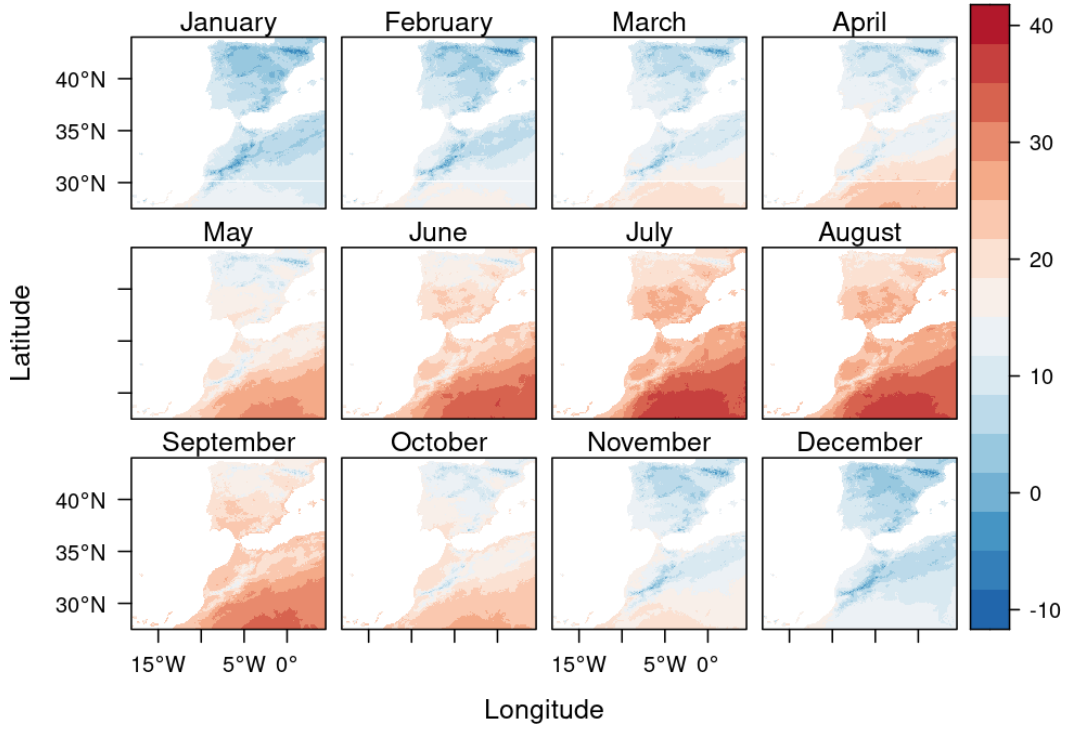
Calculate Tmean for all months

```
tmean.sp <- mean(tmin.sp, tmax.sp)
tmean.sp
```

```
class       : SpatRaster
dimensions  : 1980, 2760, 12  (nrow, ncol, nlyr)
resolution  : 0.008333333, 0.008333333  (x, y)
extent      : -18.5, 4.5, 27.5, 44  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (EPSG:4326)
source      : spat_H5ewbCBDk53UEj7_263065.tif
names       : January, February, March, April,   May,  June, ...
min values  :   -9.75,    -9.40, -7.70, -5.95, -1.40,  3.25, ...
max values  :   19.20,    18.85, 21.65, 26.05, 30.45, 36.00, ...
```

Calculate Tmean for all months

```
names(tmean.sp) <- month.name
levelplot(tmean.sp, par.settings = BuRdTheme())
```



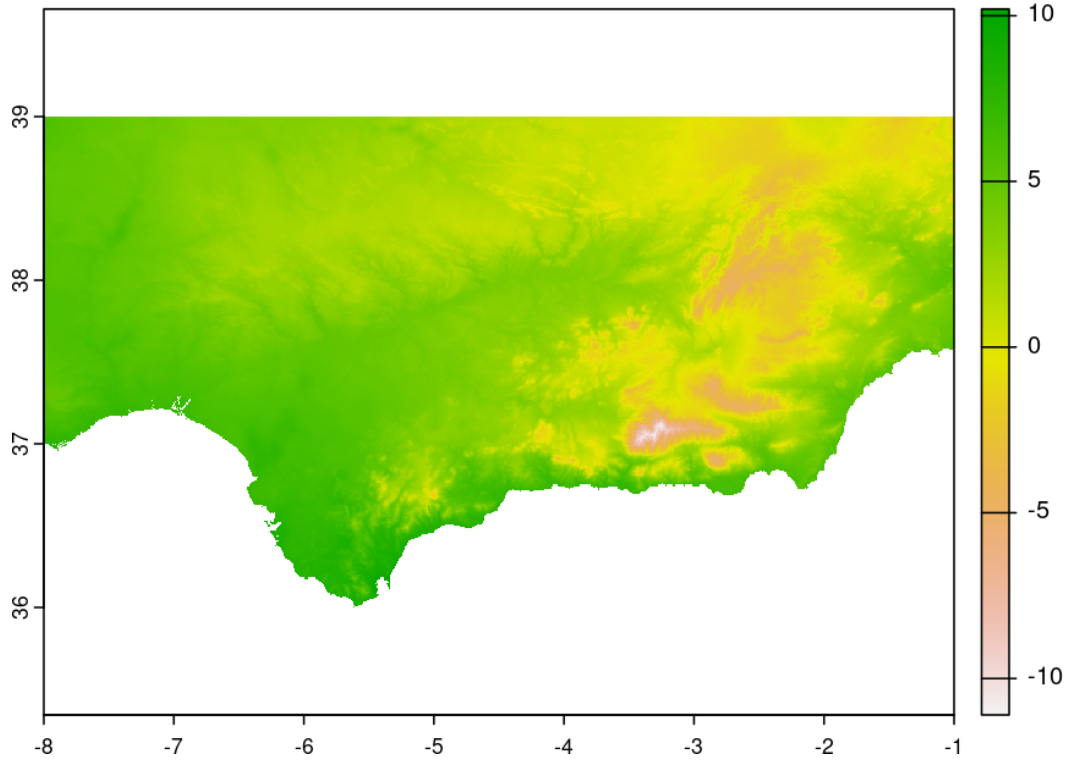
Write raster

```
writeRaster(tmean.sp, "data/tmeanSp.tif")
```

Crop and mask

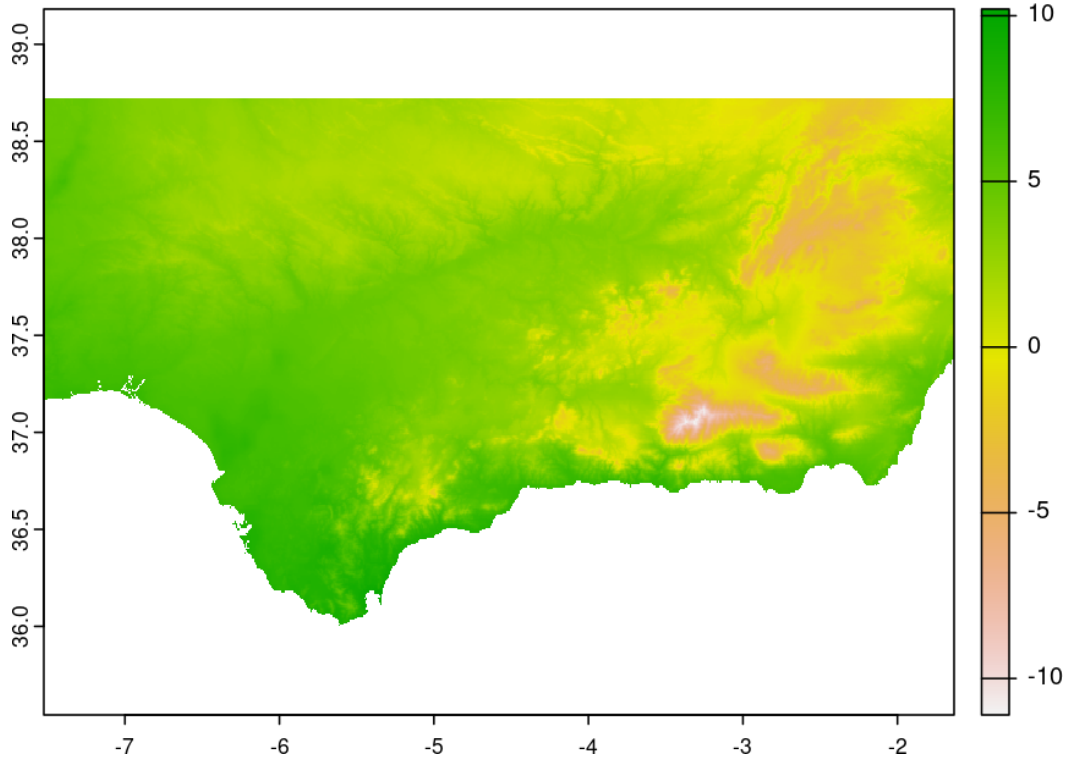
Crop given extent

```
tmin.jan.crop <- crop(tmin.jan, ext(-8, -1, 36, 39))
plot(tmin.jan.crop)
```



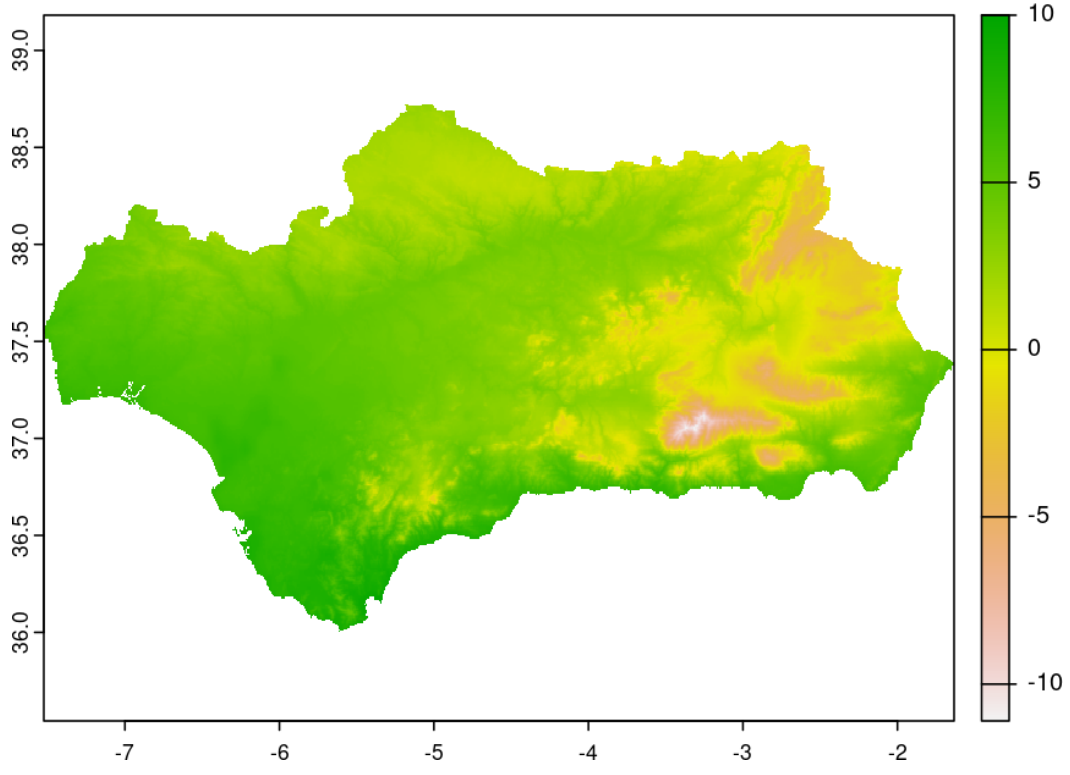
Crop with vector shape

```
library(sf)
and.lim <- vect(andal.sf)
tmin.jan.crop <- crop(tmin.jan, and.lim)
plot(tmin.jan.crop)
```



Mask

```
tmin.jan.crop <- mask(tmin.jan.crop, and.lim)
plot(tmin.jan.crop)
```

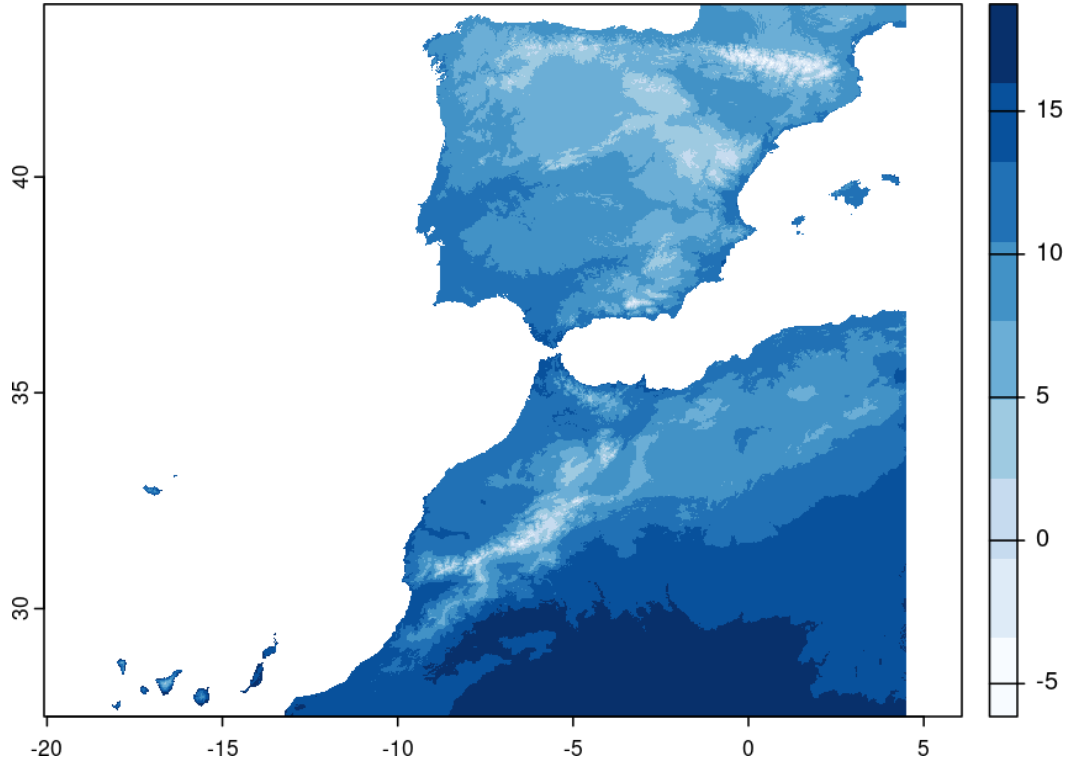


Your turn

Make map of Tmean in July (only Andalucia)

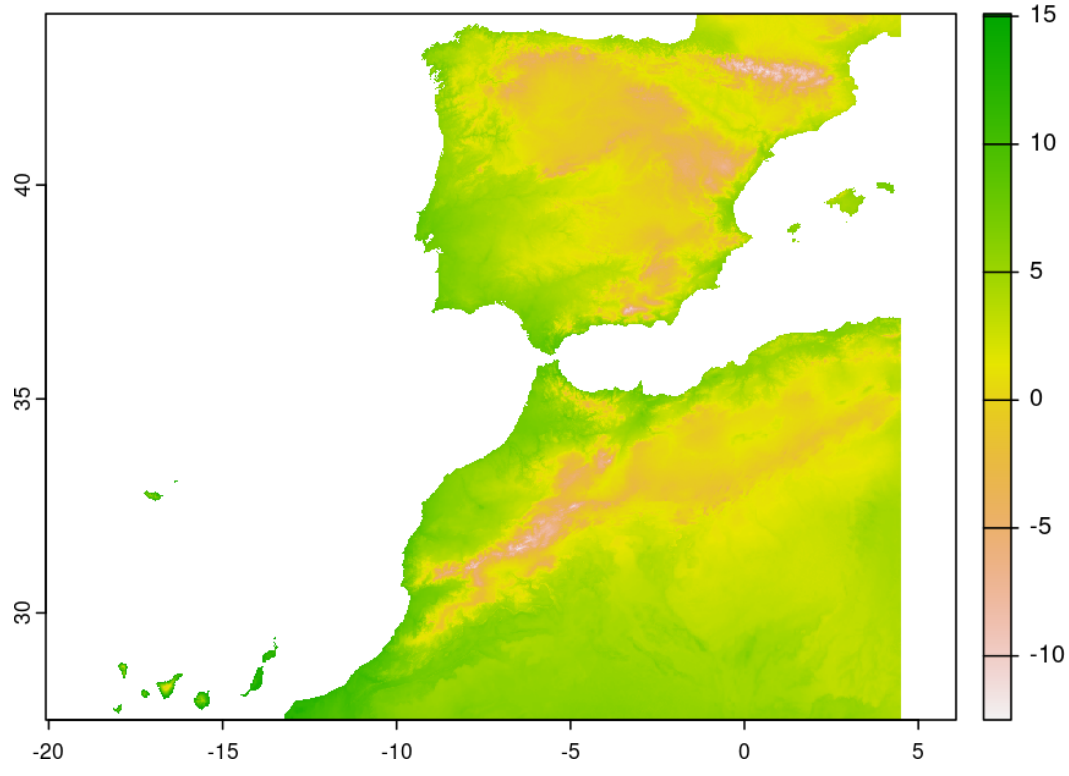
Calculate yearly average of Tmin

```
tmin.year <- mean(tmin.sp)
plot(tmin.year, col = brewer.pal(n = 9, "Blues"))
```



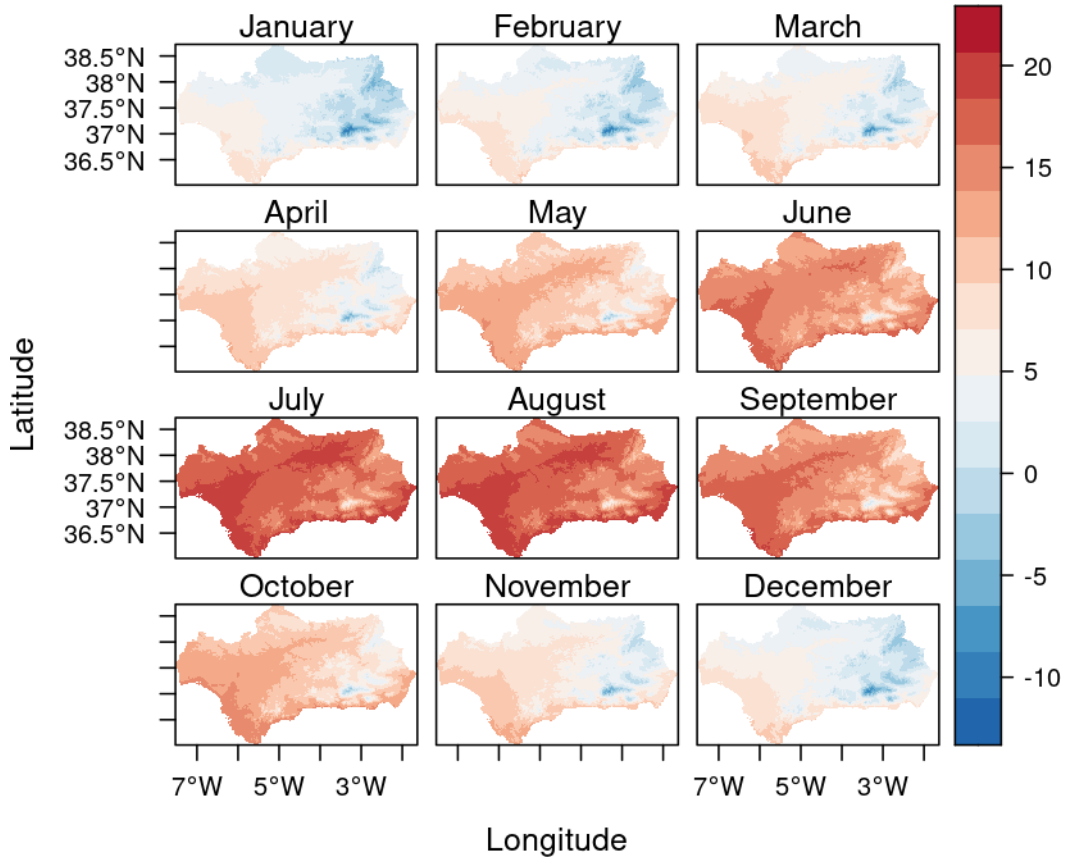
Calculate minimum temperature in winter

```
tmin.winter <- subset(tmin.sp, c("January", "February", "March"))
tmin.winter.avg <- min(tmin.winter)
plot(tmin.winter.avg)
```



Calculate minimum temperature in Andalucia

```
tmin.andal <- crop(tmin.sp, and.lim)
tmin.andal <- mask(tmin.andal, and.lim)
levelplot(tmin.andal, par.settings = BuRdTheme())
```



Get minimum temperature observed in Andalucia per month

```
global(tmin.andal, "min", na.rm = TRUE)
```

```
             min
January    -11.1
February   -11.0
March       -9.4
April       -6.4
May         -3.3
June         1.0
July         5.3
August       4.9
September    0.8
October     -3.7
November    -7.2
December    -9.6
```

Calculate mean minimum temperature in Andalucia per month

```
global(tmin.andal, "mean", na.rm = TRUE)
```

```
               mean
January    3.029599
February   3.967122
March      5.355183
April      7.382273
May       10.484301
June      14.185417
July      17.045692
August    17.052408
September 14.652595
October   10.544086
November   6.714829
December   4.295879
```

Calculate Tmin per province

```
library(dplyr)
andal.muni <- st_read("data/municipios_lite.gpkg", quiet = TRUE)
andal.provs <- andal.muni %>%
  group_by(PROVINCIA) %>%
  summarise()
plot(andal.provs)
```

Calculate Tmin per province

Project to lonlat (EPSG = 4326)

```
andal.provs.geo <- st_transform(andal.provs, crs = 4326)
```

Calculate Tmin per province

Extract raster values per polygon

```
tmin.prov <- extract(tmin.andal, vect(andal.provs.geo))
head(tmin.prov)
```

```
  ID January February March April May June July August September October
1  1    -2.5     -1.7  -0.2   2.2 6.1 10.4 13.8   13.6       9.9     5.0
2  1    -2.5     -1.7  -0.2   2.1 6.1 10.4 13.8   13.5       9.8     5.0
3  1    -2.3     -1.5   0.0   2.4 6.3 10.6 14.0   13.8      10.1     5.2
4  1    -2.2     -1.4   0.1   2.5 6.4 10.7 14.1   13.9      10.2     5.3
5  1    -2.1     -1.3   0.2   2.6 6.5 10.7 14.1   13.9      10.2     5.4
6  1    -2.2     -1.4   0.2   2.5 6.4 10.7 14.1   13.9      10.2     5.4
  November December
1      1.0     -1.4
2      1.0     -1.4
3      1.2     -1.2
4      1.4     -1.1
5      1.4     -1.0
6      1.4     -1.0
```

Calculate Tmin per province

Can apply function on the fly

```
tmin.prov <- extract(tmin.andal, vect(andal.provs.geo),
                     fun = "mean", na.rm = TRUE)
round(tmin.prov, 1)
```

```
  ID January February March April  May June July August September October
1  1     1.7      2.5   4.0   6.1  9.6 13.5 16.6   16.7      13.6     9.3
2  2     6.8      7.6   8.7  10.2 12.6 15.8 18.3   18.6      17.2    13.9
3  3     2.6      3.7   5.2   7.4 10.6 14.3 17.2   17.1      14.8    10.4
4  4    -0.1      0.7   2.2   4.5  7.9 11.9 15.0   14.8      11.7     7.3
5  5     5.1      6.1   7.4   9.1 11.8 15.2 17.7   17.7      16.0    12.6
6  6     1.3      2.3   3.9   6.2  9.8 14.0 17.1   16.8      13.7     9.0
7  7     4.3      5.0   6.1   8.0 10.6 14.1 16.8   17.3      15.2    11.4
8  8     4.7      5.7   7.1   9.0 11.8 15.3 18.0   18.1      16.2    12.3
  November December
1      5.4      2.8
2     10.6      8.2
3      6.4      3.9
4      3.5      1.1
5      8.8      6.4
6      4.8      2.5
7      7.9      5.6
```

Extract minimum temperature for AEMET stations

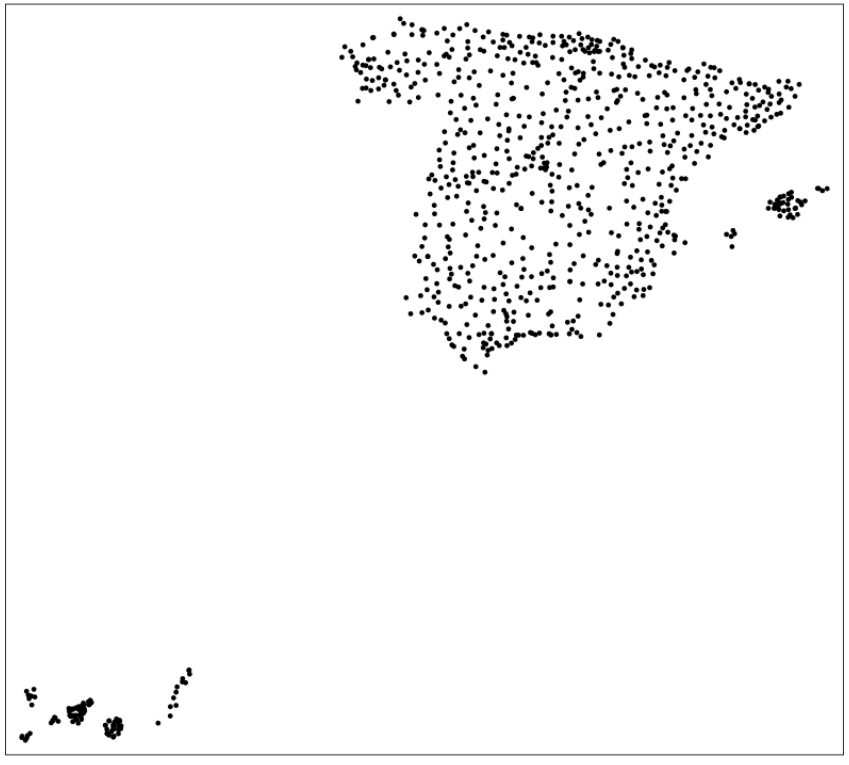
Load AEMET stations

```
aemet <- st_read("data/aemet/Estaciones_Automaticas.shp", quiet = TRUE)
head(aemet)
```

```
Simple feature collection with 6 features and 9 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: -4.575001 ymin: 41.77556 xmax: -1.976663 ymax: 42.56889
Geodetic CRS:  ETRS89
  INDICATIVO                    NOMBRE PROVINCIA ALTITUD COORD_X COORD_Y
1      2276B     VILLAELES DE VALDAVIA  PALENCIA     908  370729 4714145
2      2290Y      PEDROSA DEL PRÍNCIPE    BURGOS     771  401101 4678354
3      2296A    ÓLVEGA-CAMINO VEGAFRÍA     SORIA    1039  585048 4625363
4       2298     PALACIOS DE LA SIERRA    BURGOS    1080  489089 4645313
5      2302N MONTERRUBIO DE LA DEMANDA    BURGOS    1197  490842 4666036
6      2311Y  VILLAMAYOR DE LOS MONTES    BURGOS     882  436519 4661751
   VAR_OBSVER  DATUM        TIPO                    geometry
1   hHVmtTMpP ETRS89 AUTOMATICAS POINT (-4.575001 42.56889)
2   hHVmtTMpP ETRS89 AUTOMATICAS POINT (-4.198895 42.25112)
3   hHVmtTMpP ETRS89 AUTOMATICAS POINT (-1.976663 41.77556)
4 hIHBVmtTMpP ETRS89 AUTOMATICAS POINT (-3.131664 41.95973)
5   hHVmtTMpP ETRS89 AUTOMATICAS POINT (-3.110835 42.14639)
6   hHVmtTMpP ETRS89 AUTOMATICAS  POINT (-3.76778 42.10528)
```

AEMET stations

```
tm_shape(aemet) +
  tm_dots()
```



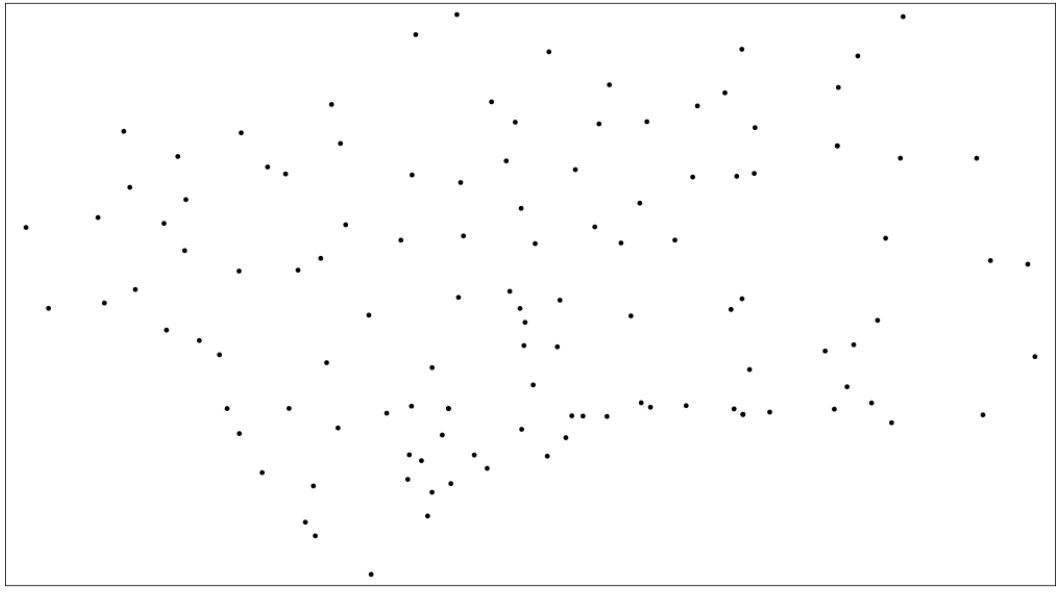
Project to lonlat

```
aemet <- st_transform(aemet, crs = 4326)
head(aemet)
```

```
Simple feature collection with 6 features and 9 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: -4.575001 ymin: 41.77556 xmax: -1.976663 ymax: 42.56889
Geodetic CRS:  WGS 84
  INDICATIVO                    NOMBRE PROVINCIA ALTITUD COORD_X COORD_Y
1      2276B     VILLAELES DE VALDAVIA  PALENCIA     908  370729 4714145
2      2290Y      PEDROSA DEL PRÍNCIPE    BURGOS     771  401101 4678354
3      2296A    ÓLVEGA-CAMINO VEGAFRÍA     SORIA    1039  585048 4625363
4       2298     PALACIOS DE LA SIERRA    BURGOS    1080  489089 4645313
5      2302N MONTERRUBIO DE LA DEMANDA    BURGOS    1197  490842 4666036
6      2311Y  VILLAMAYOR DE LOS MONTES    BURGOS     882  436519 4661751
   VAR_OBSVER  DATUM        TIPO                    geometry
1   hHVmtTMpP ETRS89 AUTOMATICAS POINT (-4.575001 42.56889)
2   hHVmtTMpP ETRS89 AUTOMATICAS POINT (-4.198895 42.25112)
3   hHVmtTMpP ETRS89 AUTOMATICAS POINT (-1.976663 41.77556)
4 hIHBVmtTMpP ETRS89 AUTOMATICAS POINT (-3.131664 41.95973)
5   hHVmtTMpP ETRS89 AUTOMATICAS POINT (-3.110835 42.14639)
6   hHVmtTMpP ETRS89 AUTOMATICAS  POINT (-3.76778 42.10528)
```

Keep only stations in Andalucia

```
aemet.andal <- st_filter(aemet, andal.sf)
tm_shape(aemet.andal) +
  tm_dots()
```



Extract minimum January temperature for AEMET stations

```
aemet.tmin <- extract(tmin.jan, vect(aemet.andal))
aemet.tmin <- bind_cols(aemet.andal, aemet.tmin)
head(aemet.tmin)
```

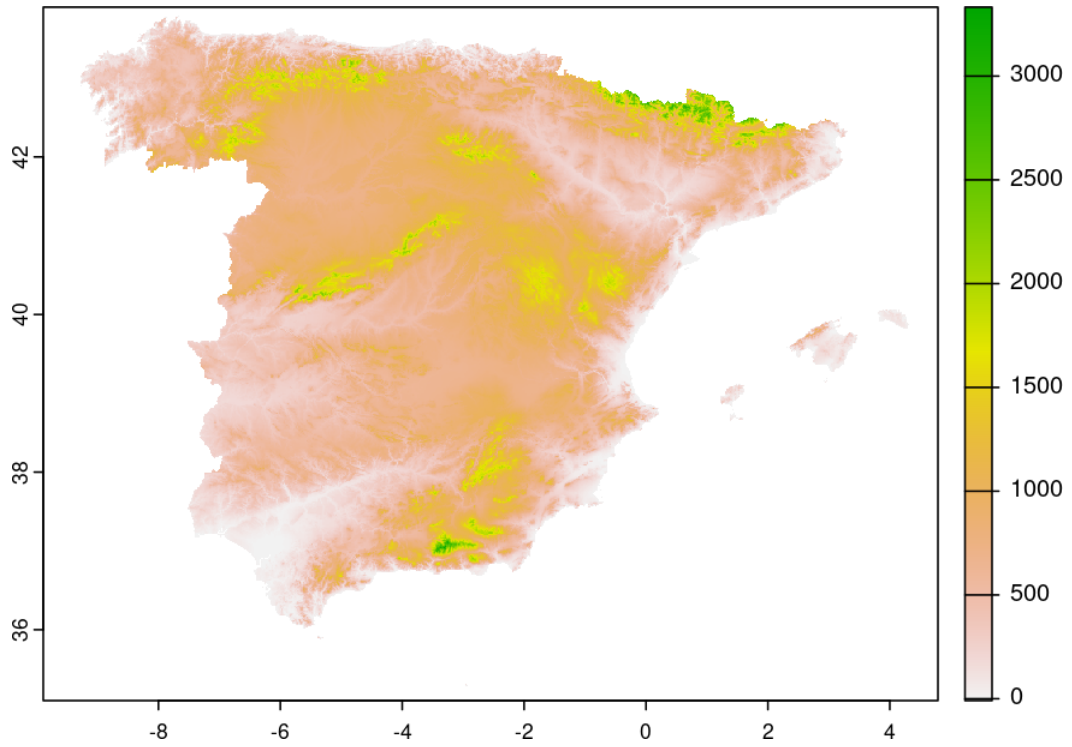
```
Simple feature collection with 6 features and 11 fields
Geometry type: POINT
Dimension:     XY
Bounding box:  xmin: -7.520007 ymin: 37.19476 xmax: -5.121384 ymax: 38.49834
Geodetic CRS:  WGS 84
  INDICATIVO                                NOMBRE PROVINCIA ALTITUD COORD_X
1      4263X                             VALSEQUILLO   CORDOBA     575  294767
2      4267X HINOJOSA DEL DUQUE-DEHESAS CORDOBESAS   CORDOBA     551  315006
3      4541X  EL GRANADO (BOCACHANZA - AUTOMÁTICA)    HUELVA      60  100655
4      4549Y                 AYAMONTE, ISLA CANELA    HUELVA       2  109846
5      4554X                     CARTAYA , PEMARES    HUELVA      15  137615
6      4560Y                 ALAJAR, CABEZO MOLINO    HUELVA     572  176715
  COORD_Y VAR_OBSVER  DATUM        TIPO ID January                    geometry
1 4253863  hHVmtTMpP ETRS89 AUTOMATICAS  1     1.8 POINT (-5.350555 38.40946)
2 4263242 hHBVmtTMpP ETRS89 AUTOMATICAS  2     1.5 POINT (-5.121384 38.49834)
3 4165905   hHmtTMpP ETRS89 AUTOMATICAS  3     6.5 POINT (-7.520007 37.55364)
4 4125534 hHBVmtTMpP ETRS89 AUTOMATICAS  4     7.2 POINT (-7.395001 37.19476)
5 4126915 hHBVmtTMpP ETRS89 AUTOMATICAS  5     6.8 POINT (-7.083612 37.21836)
```

Predict Tmin with elevation across Andalucia

Get elevation raster data

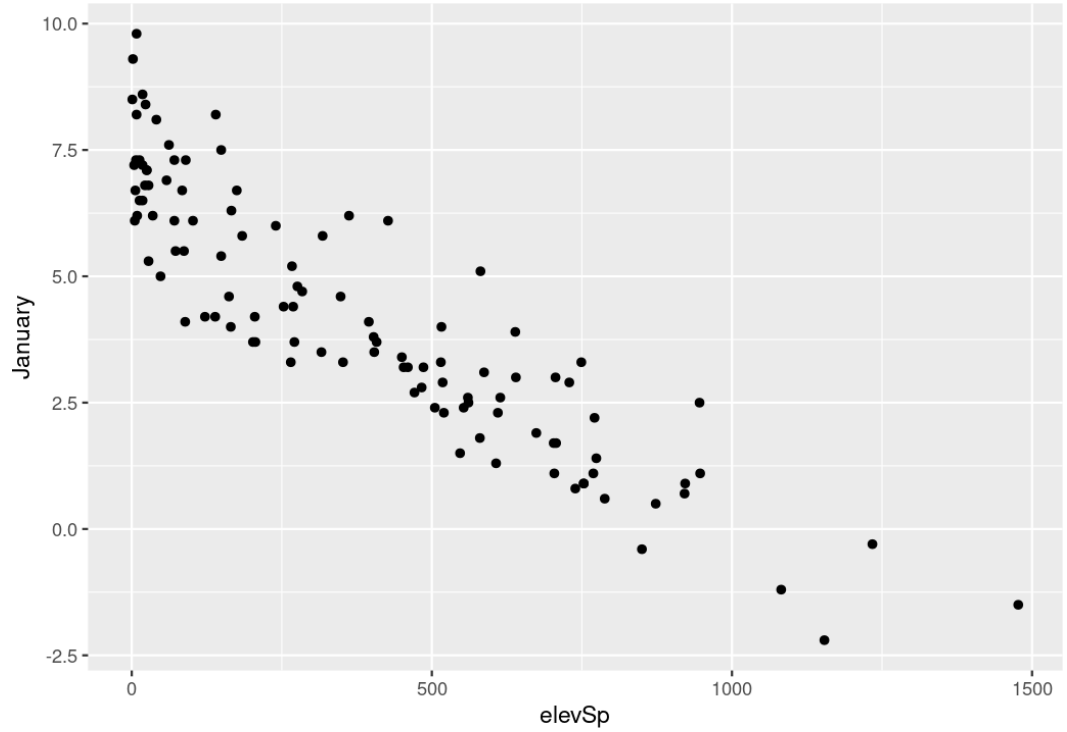
```
library(geodata)
elev <- elevation_30s(country = "Spain", path = "data/")
```

```
plot(elev)
```



Extract elevation for AEMET stations in Andalucia

```
aemet.elev <- extract(elev, vect(aemet.andal))
aemet.tmin.elev <- bind_cols(aemet.tmin, aemet.elev)
ggplot(aemet.tmin.elev) +
  geom_point(aes(elevSp, January))
```



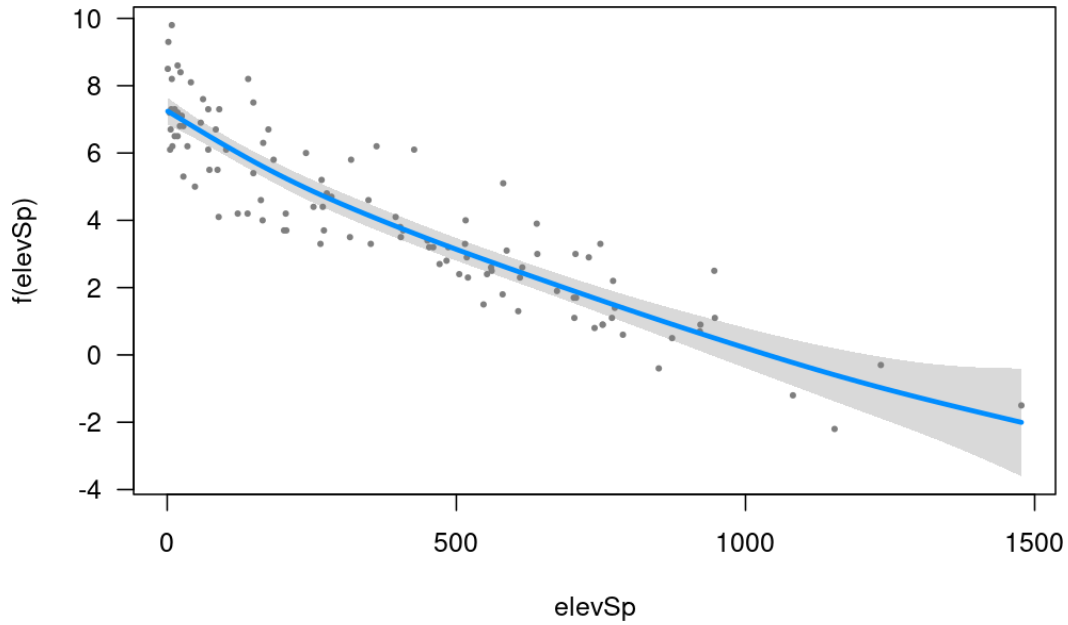
Fit model

```
library(mgcv)
model <- gam(January ~ s(elevSp), data = aemet.tmin.elev)
model
```

```
Family: gaussian
Link function: identity

Formula:
January ~ s(elevSp)

Estimated degrees of freedom:
3.16  total = 4.16

GCV score: 1.178608
```

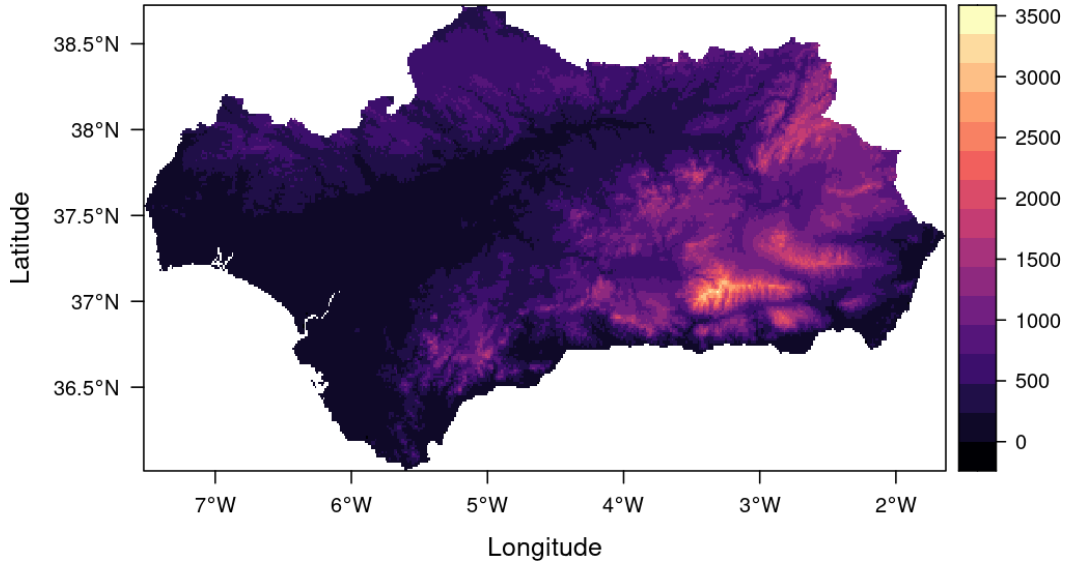
Fit model

```
visreg::visreg(model)
```



Crop & mask elevation map for Andalucia

```
elev.andal <- crop(elev, and.lim)
elev.andal <- mask(elev.andal, and.lim)
levelplot(elev.andal, margin = FALSE)
```



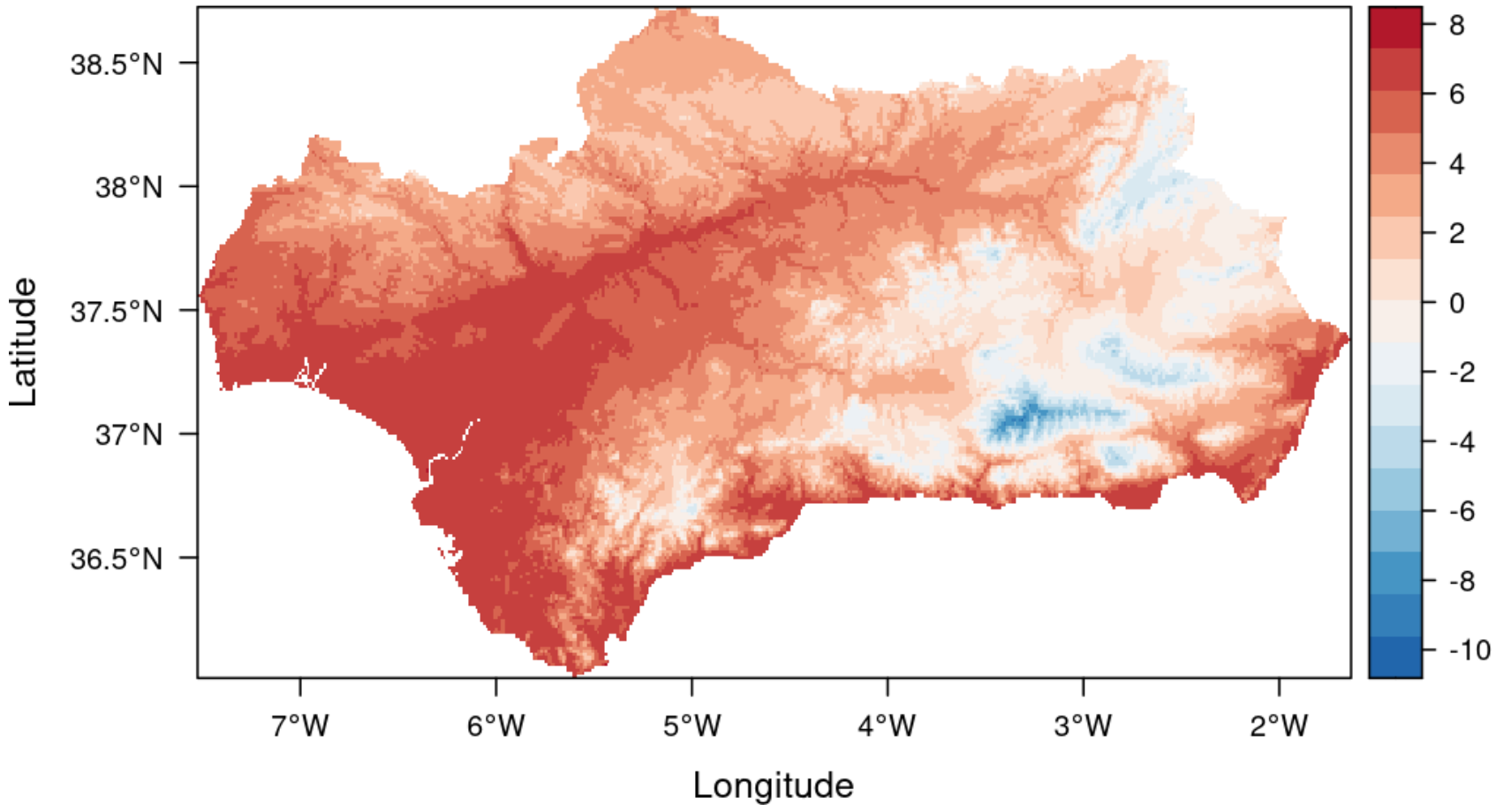
Predict Tmin from elevation across Andalucia

```
tmin.pred <- predict(elev.andal, model)
tmin.pred
```

```
class       : SpatRaster
dimensions  : 327, 707, 1  (nrow, ncol, nlyr)
resolution  : 0.008333333, 0.008333333  (x, y)
extent      : -7.525, -1.633333, 36, 38.725  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (EPSG:4326)
source      : memory
name        :    elevSp
min value   : -9.629749
max value   :  7.313638
```

Predict Tmin from elevation across Andalucia

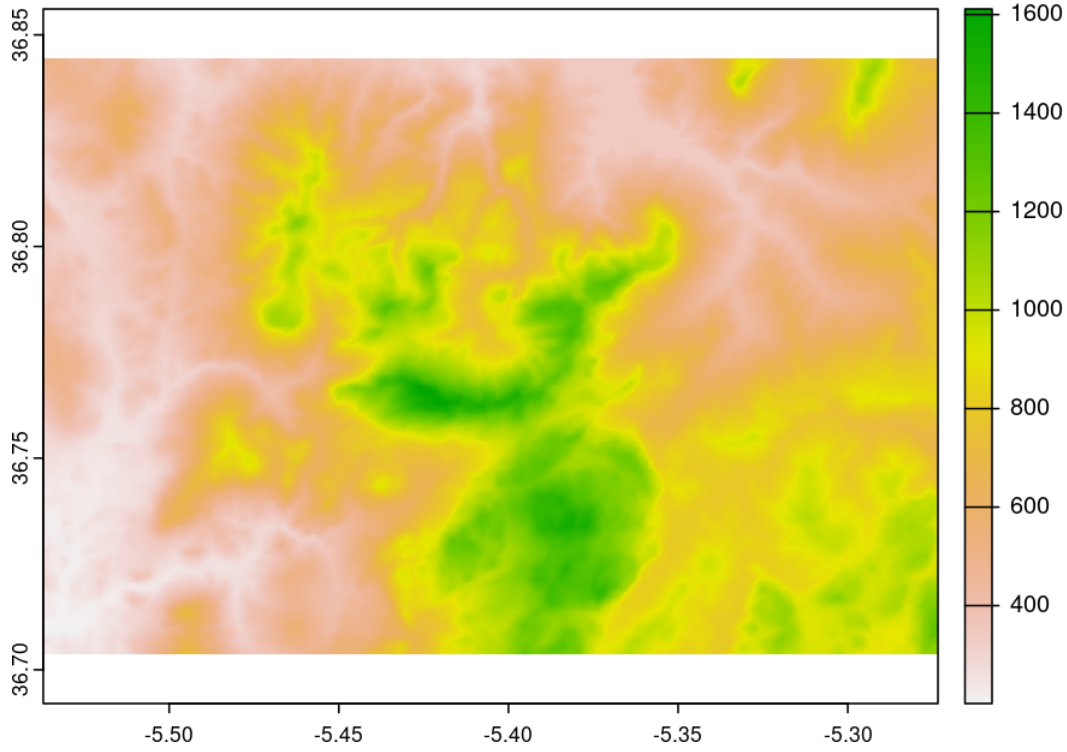
```
levelplot(tmin.pred, margin = FALSE, par.settings = BuRdTheme())
```



Make elevation map

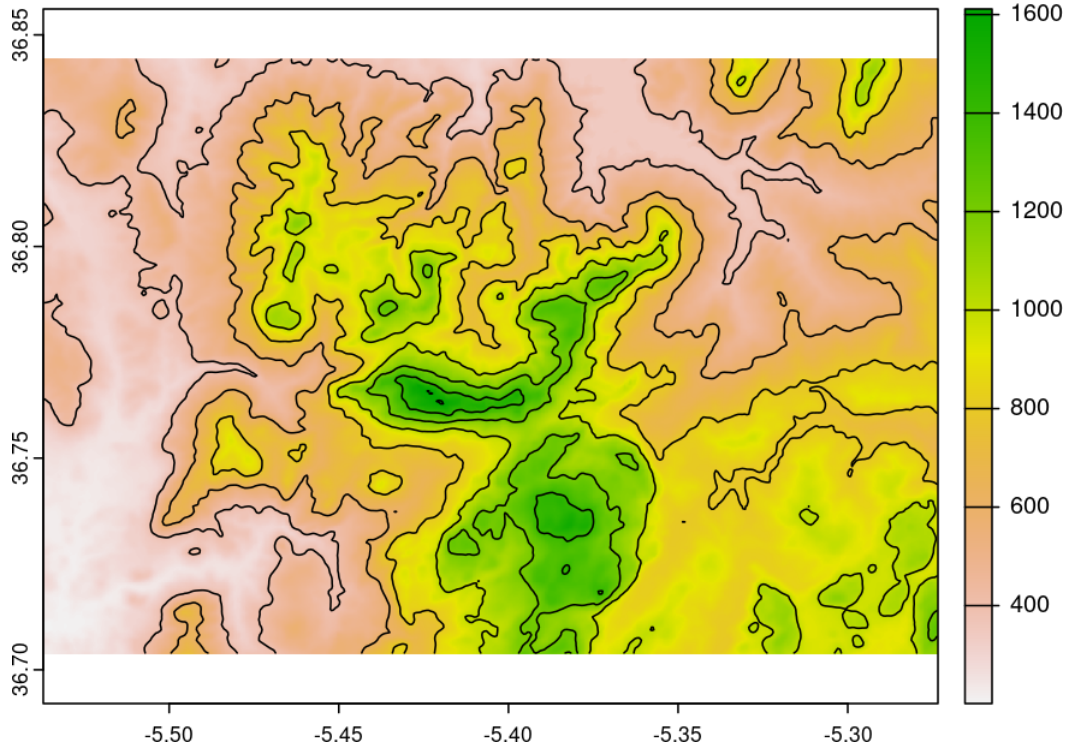
Get Grazalema elevation data

```
grazalema <- andal.muni %>%
  filter(MUNICIPIO == "Grazalema") %>%
  st_transform(crs = 4326)

elev.gra <- elevatr::get_elev_raster(grazalema, z = 13)
elev.gra <- rast(elev.gra)
plot(elev.gra)
```



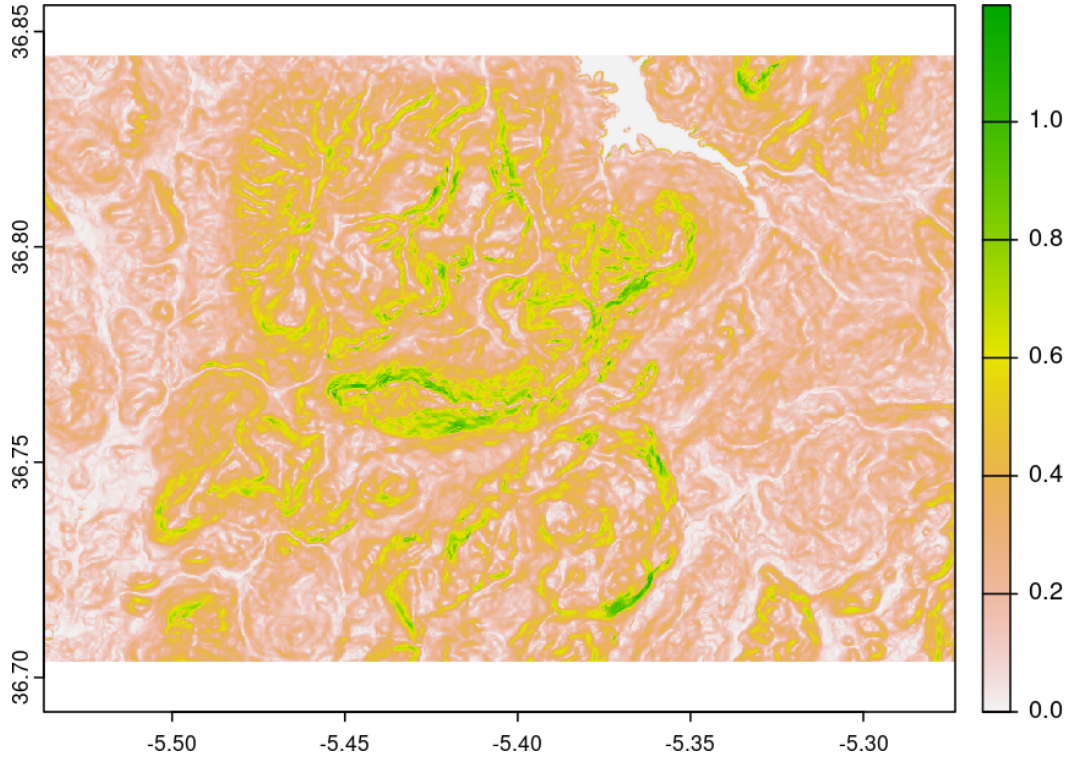
Add contour lines

```
plot(elev.gra)
plot(as.contour(elev.gra), add = TRUE)
```



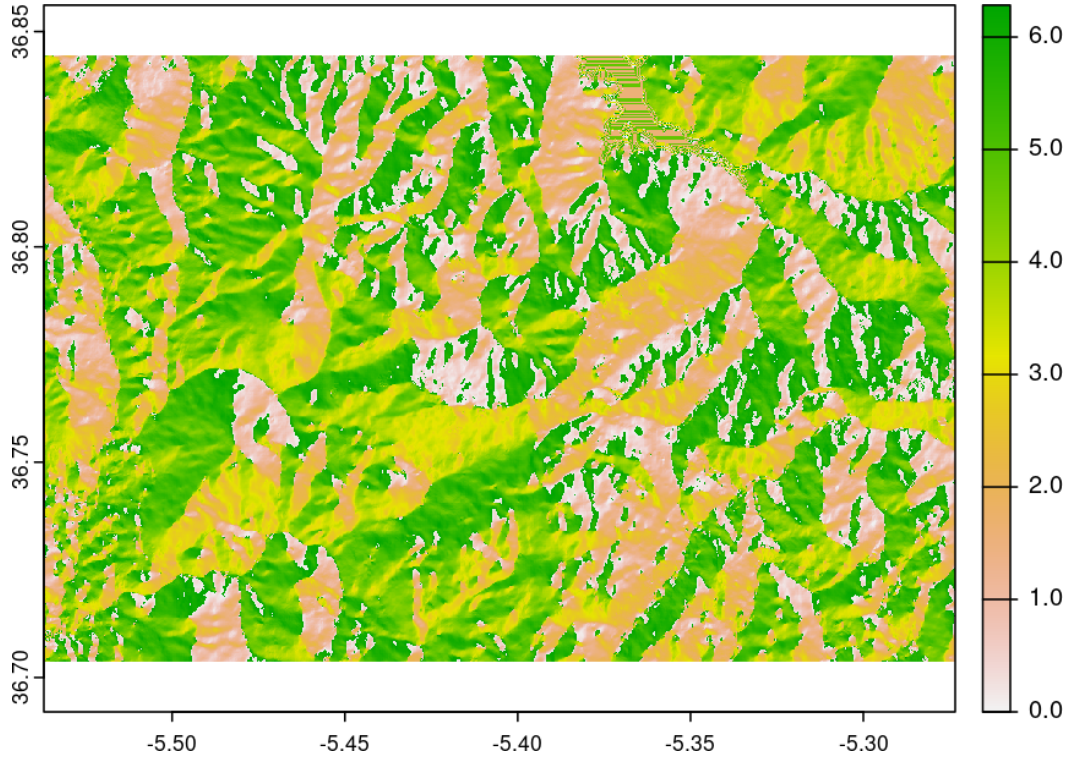
Calculate slopes and aspect for hillshading

```
slopes <- terrain(elev.gra, "slope", unit = "radians")
plot(slopes)
```



Calculate slopes and aspect for hillshading

```
aspect <- terrain(elev.gra, "aspect", unit = "radians")
plot(aspect)
```

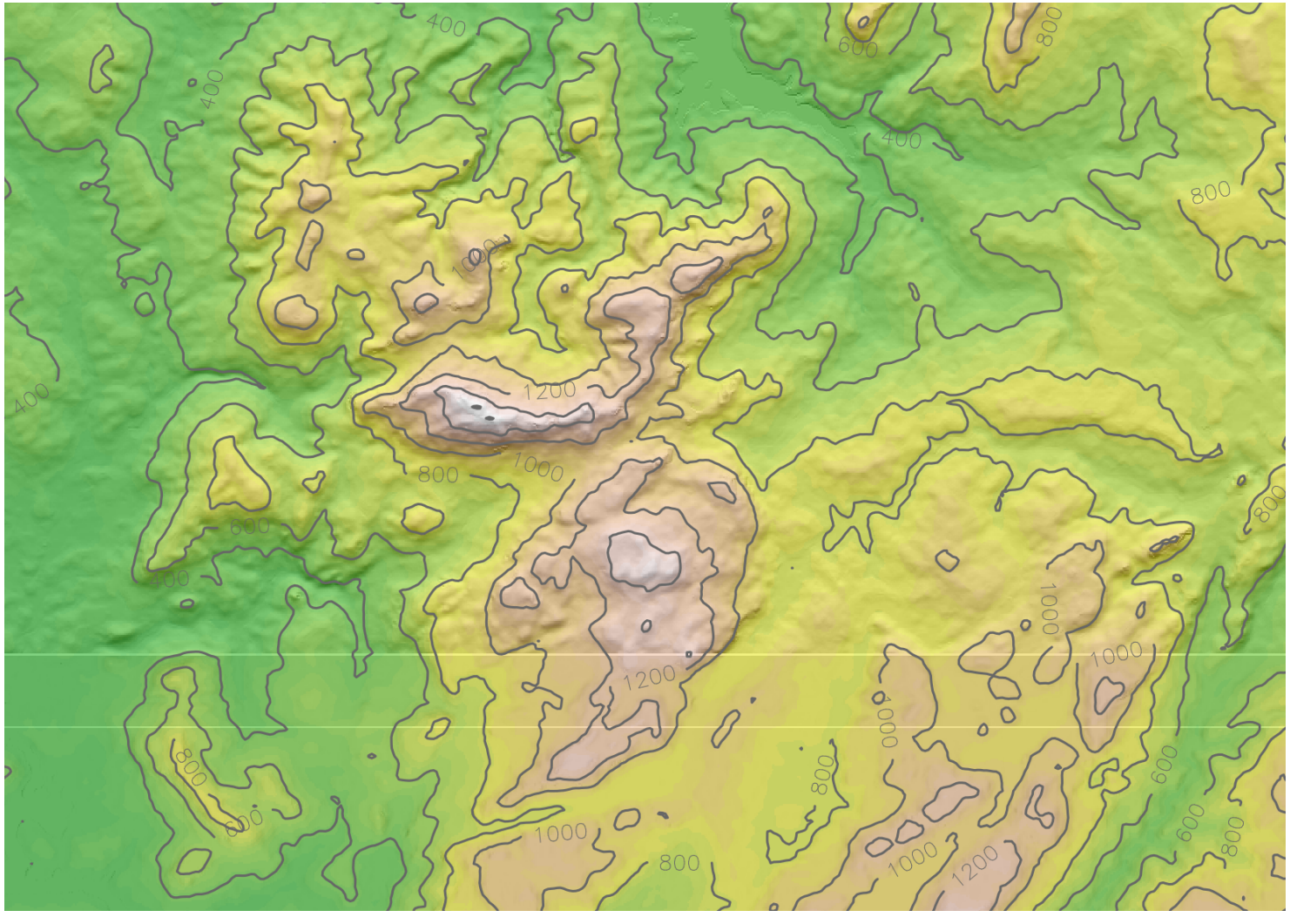


Make elevation map

```
# Plot hillshading as basemap
hs <- shade(slopes, aspect)
plot(hs, col = gray(0:100 / 100), legend = FALSE, axes = FALSE)

# overlay with elevation
plot(elev.gra, col = terrain.colors(25), alpha = 0.5, legend = FALSE, axes = FALSE, add = TRUE)

# add contour lines
contour(elev.gra, col = "grey40", add = TRUE)
```

Source: Geocomputation with R

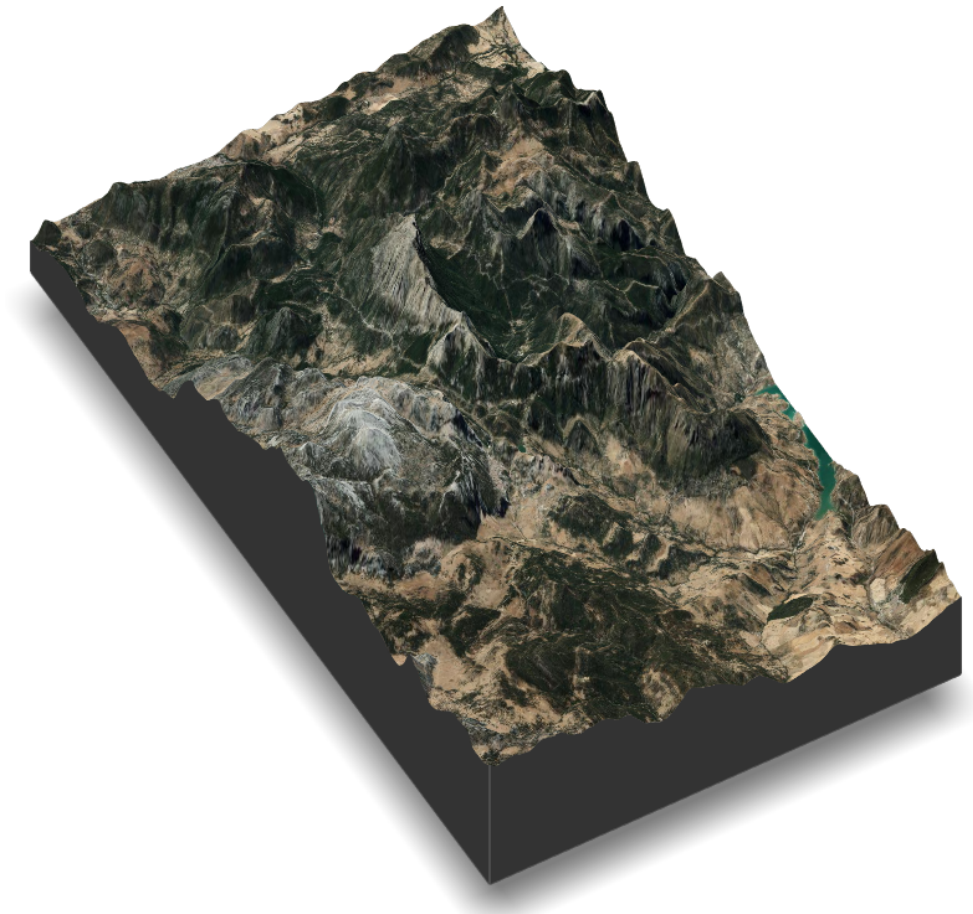
Make elevation map



3-D maps with rayshader and rayvista

3-D maps with rayshader and rayvista

```
library(rayshader)
library(rayvista)

## Grazalema limits (polygon)
grazalema <- andal.muni %>%
  filter(MUNICIPIO == "Grazalema") %>%
  st_transform(crs = 4326)

graz.3D <- plot_3d_vista(req_area = grazalema
```



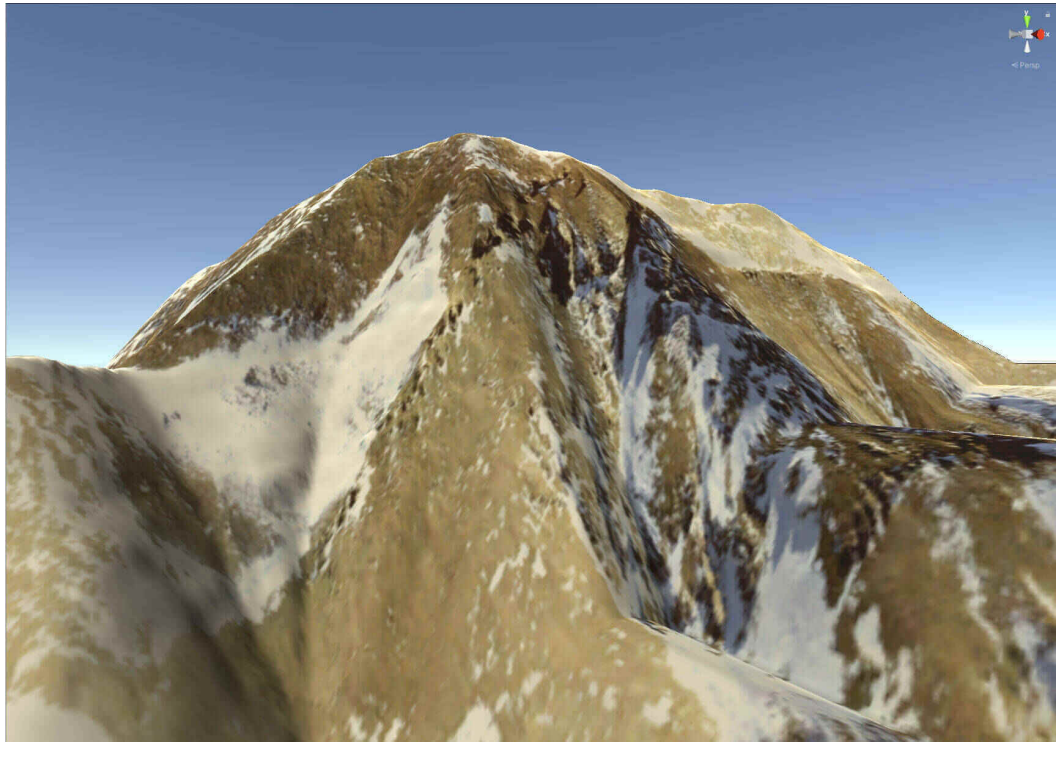
3-D maps with rayshader and rayvista

```
library(rayshader)
library(rayvista)
lapalmaTF <- plot_3d_vista(
  lat = 28.719946, long = -17.867091, radius
  overlay_detail = 13, overlay_alpha = 0.6,
  elevation_detail = 11, show_vista = FALSE)

lapalmaTF$dem_matrix %>%
  height_shade() %>%
  add_shadow(ray_shade(lapalmaTF$dem_matrix,
  add_overlay(., lapalmaTF$texture,rescale_or
  plot_3d(., lapalmaTF$dem_matrix, zscale = 2
          windowsize = 1200, zoom = 0.25, phi

render_snapshot(clear = TRUE)
```

terrainr

Interactive 3D exploration with Unity

https://docs.ropensci.org/terrainr



Project raster

Projecting rasters

There's a `terra::project` function,

but rather convert to vector and project that

Converting raster to points

```
elev.gra <- crop(elev, grazalema)
elev.gra
```

```
class       : SpatRaster
dimensions  : 11, 25, 1  (nrow, ncol, nlyr)
resolution  : 0.008333333, 0.008333333  (x, y)
extent      : -5.5, -5.291667, 36.725, 36.81667  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (EPSG:4326)
source      : memory
name        : elevSp
min value   :    283
max value   :   1499
```

```
elev.pts <- as.points(elev.gra)
elev.pts
```

```
 class       : SpatVector
 geometry    : points
 dimensions  : 275, 1  (geometries, attributes)
 extent      : -5.495833, -5.295833, 36.72917, 36.8125  (xmin, xmax, ymin, ymax)
 coord. ref. : lon/lat WGS 84 (EPSG:4326)
 names       : elevSp
 type        :  <num>
 values      :    434
                  484
                  588
```

Changing points CRS

Using `terra::project`:

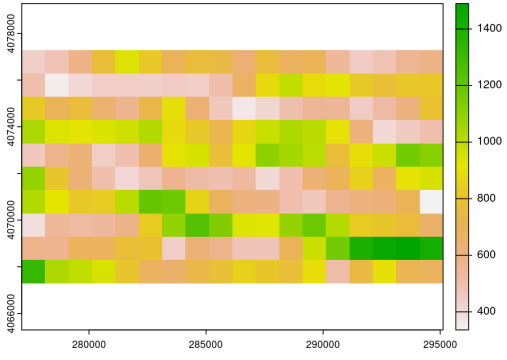
```
elev.pts.utm <- project(elev.pts, "epsg:25830")
```

Using `sf_transform`:

```
elev.pts.utm.sf <- elev.pts %>%
  st_as_sf() %>%
  st_transform(crs = 25830) %>%
  vect()
```

Rasterize points

```
ras <- rast(elev.pts.utm, resolution = 1000,
            vals = elev.pts.utm$elevSp)
ras
```




```
class       : SpatRaster
dimensions  : 10, 18, 1  (nrow, ncol, nlyr)
resolution  : 1000, 1000  (x, y)
extent      : 277125.2, 295125.2, 4067286, 4077286  (xmin, xmax, ymin, ymax)
coord. ref. : ETRS89 / UTM zone 30N (EPSG:25830)
source      : memory
name        : lyr.1
min value   :   336
max value   :  1490
```

Remote sensing

Calculating NDVI from landsat images

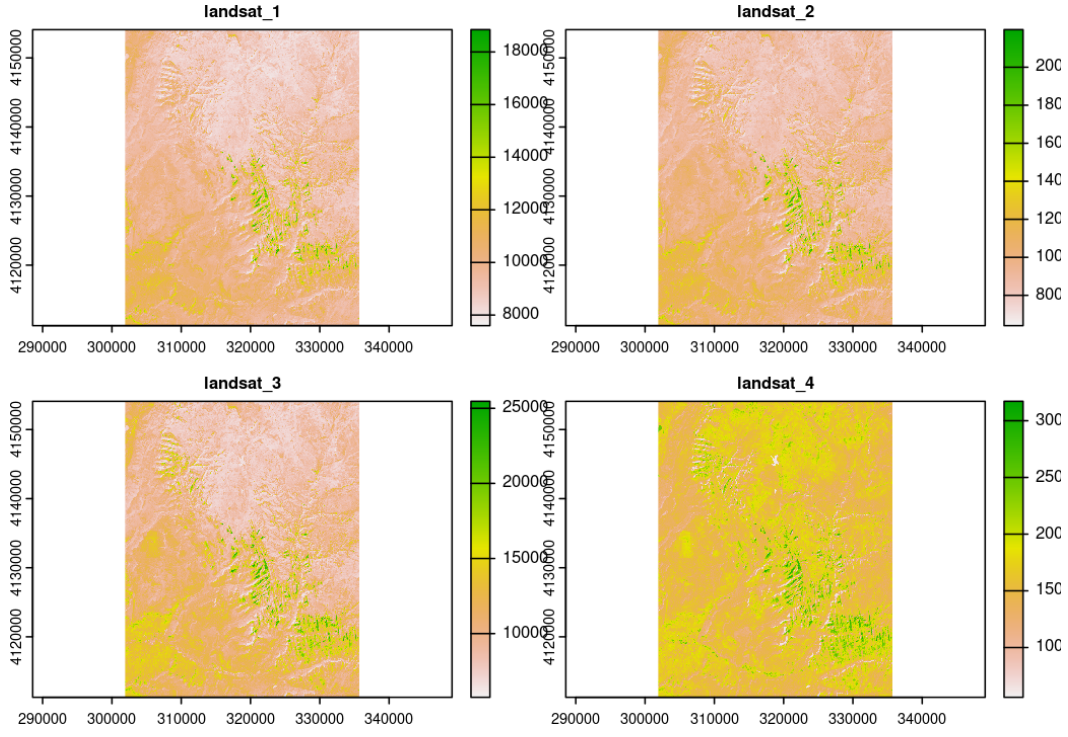
```
landsat <- system.file("raster/landsat.tif", package = "spDataLarge")
landsat <- rast(landsat)
landsat
```

```
class       : SpatRaster
dimensions  : 1428, 1128, 4  (nrow, ncol, nlyr)
resolution  : 30, 30  (x, y)
extent      : 301905, 335745, 4111245, 4154085  (xmin, xmax, ymin, ymax)
coord. ref. : WGS 84 / UTM zone 12N (EPSG:32612)
source      : landsat.tif
names       : landsat_1, landsat_2, landsat_3, landsat_4
min values  :      7550,      6404,      5678,      5252
max values  :     19071,     22051,     25780,     31961
```

Calculating NDVI from landsat images

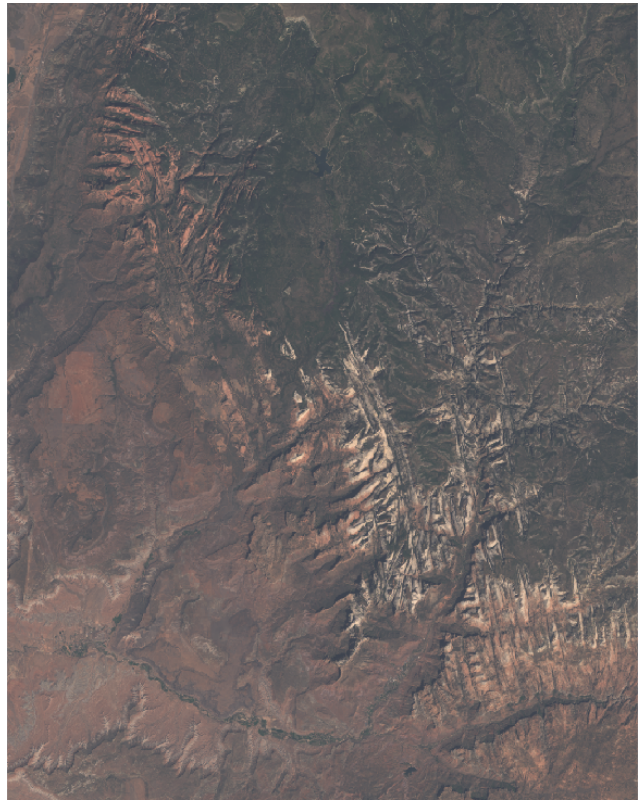
4 bands: blue, green, red, near-infrared

```
plot(landsat)
```



Plotting real-color image

```
plotRGB(landsat, r = 3, g = 2, b = 1)
```



Calculating NDVI from landsat images

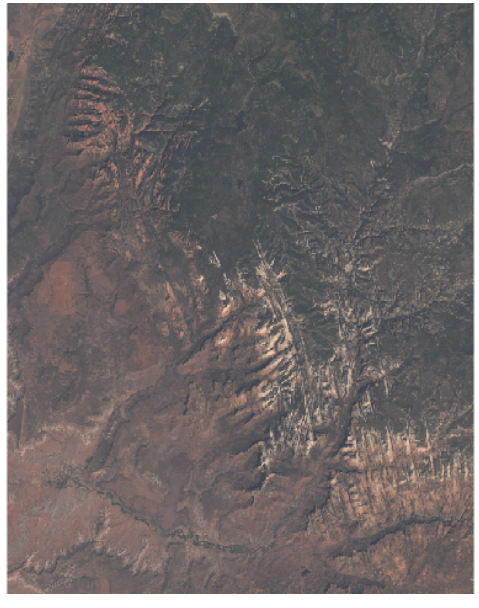
$$NDVI = \frac{NIR - RED}{NIR + RED}$$

```
ndvi <- (landsat[[4]] - landsat[[3]]) / (landsat[[4]] + landsat[[3]])
ndvi
```

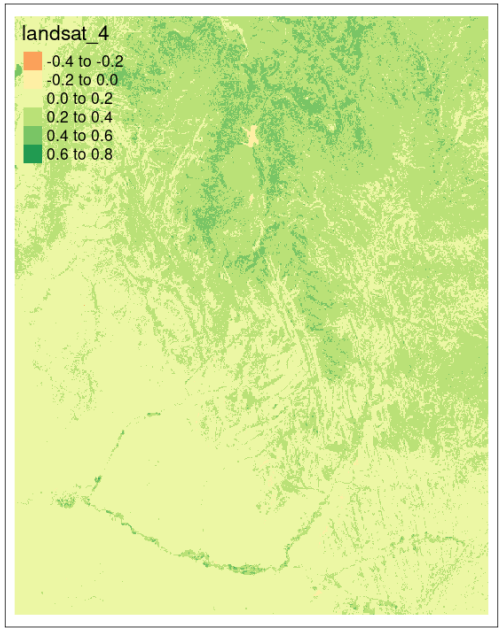
```
class       : SpatRaster
dimensions  : 1428, 1128, 1  (nrow, ncol, nlyr)
resolution  : 30, 30  (x, y)
extent      : 301905, 335745, 4111245, 4154085  (xmin, xmax, ymin, ymax)
coord. ref. : WGS 84 / UTM zone 12N (EPSG:32612)
source      : memory
name        :  landsat_4
min value   : -0.2352531
max value   :  0.6076995
```

Image & NDVI

```
plotRGB(landsat, r = 3, g = 2, b = 1)
```



```
tm_shape(ndvi) + tm_raster()
```



Your turn

Your turn

- Make elevation map
- Find & map 10 highest towns in Andalucia
- Make climatic map
- Find & map 10 towns with coldest temperatures
- Find & map 10 towns with warmest temperatures
- Find & map 10 most rainy towns
- Find & map 10 driest towns

Making interactive leaflet maps with R

Francisco Rodríguez-Sánchez

@frod_san

Mapping point data

Palm trees in Sevilla

X	Y	species	perimeter	height
-5.972411	37.40811	Washingtonia robusta	119	12.0
-5.951808	37.37690	Phoenix canariensis	240	5.0
-5.987472	37.36899	Phoenix dactylifera	131	7.5
-6.006355	37.37571	Phoenix dactylifera	NA	NA
-5.973350	37.41542	Washingtonia filifera	108	8.0
-6.006485	37.39652	Chamaerops humilis	45	2.0

Source: https://opendata.esri.es/datasets/ideSEVILLA::parques-y-jardines-palmera-viario/about

Make interactive map (leaflet)

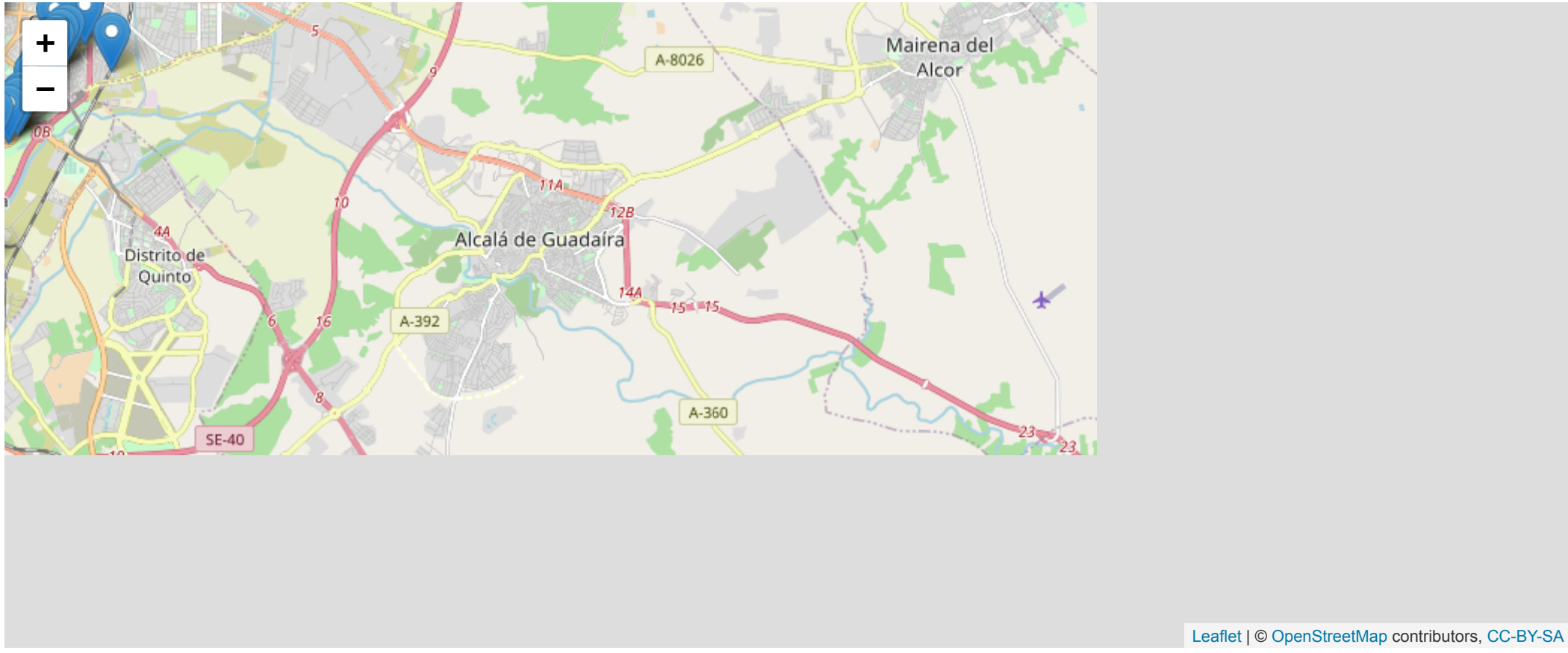
```
library("leaflet")
```

https://rstudio.github.io/leaflet

(Cheatsheet)

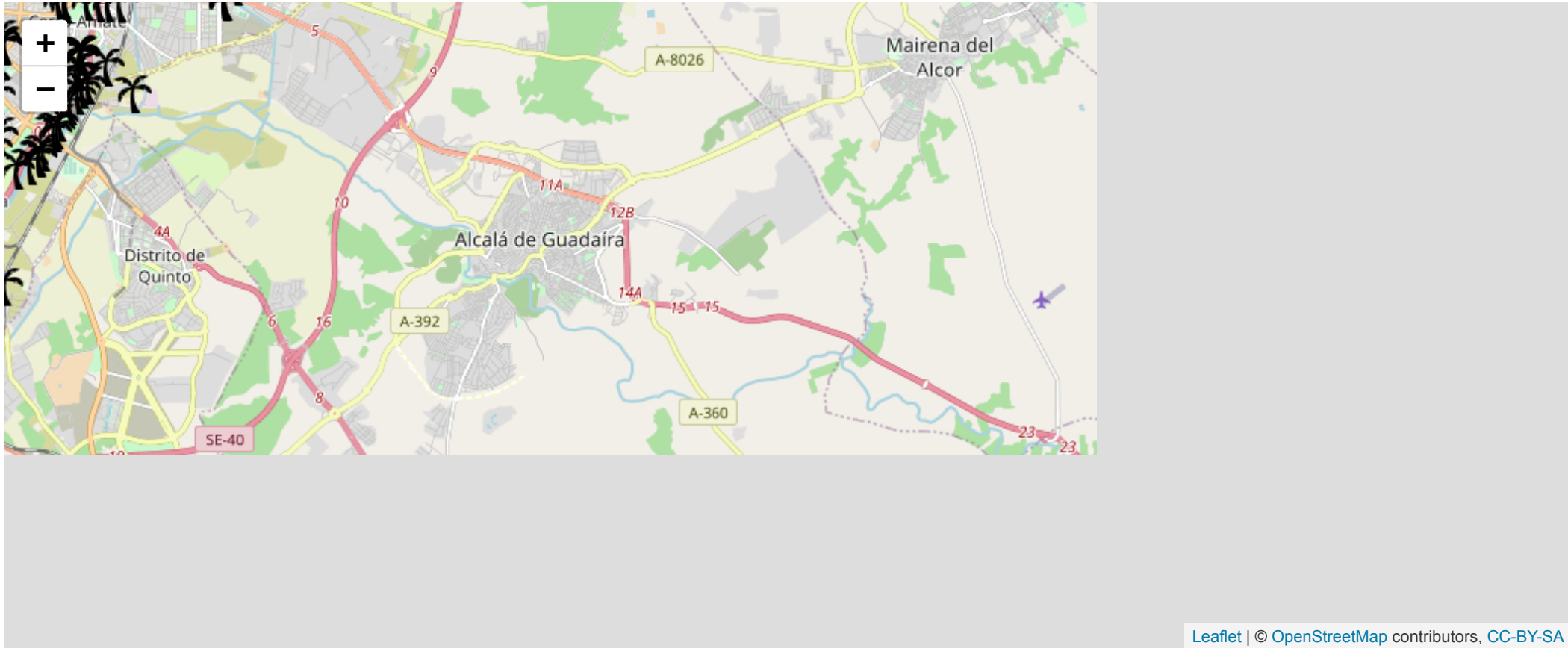
Make interactive map (leaflet)

```
leaflet(palms) %>%
  addTiles() %>%
  addMarkers(lng = ~X, lat = ~Y)
```



Hey those are palm trees!

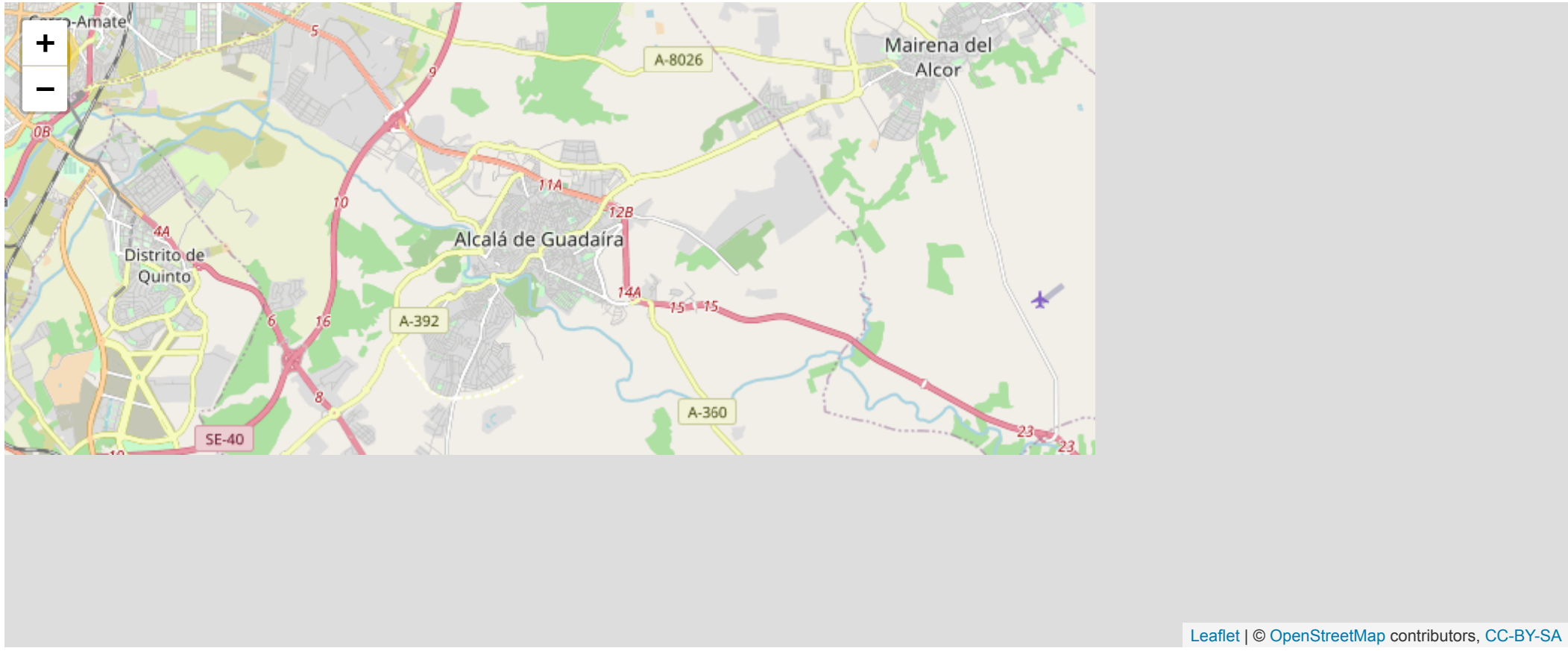
```
palm.icon <- makeIcon("https://img.icons8.com/ios-glyphs/30/000000/palm-tree.png")

leaflet(palms) %>%
  addTiles() %>%
  addMarkers(lng = ~X, lat = ~Y, icon = palm.icon)
```



Palm Tree icon by Icons8

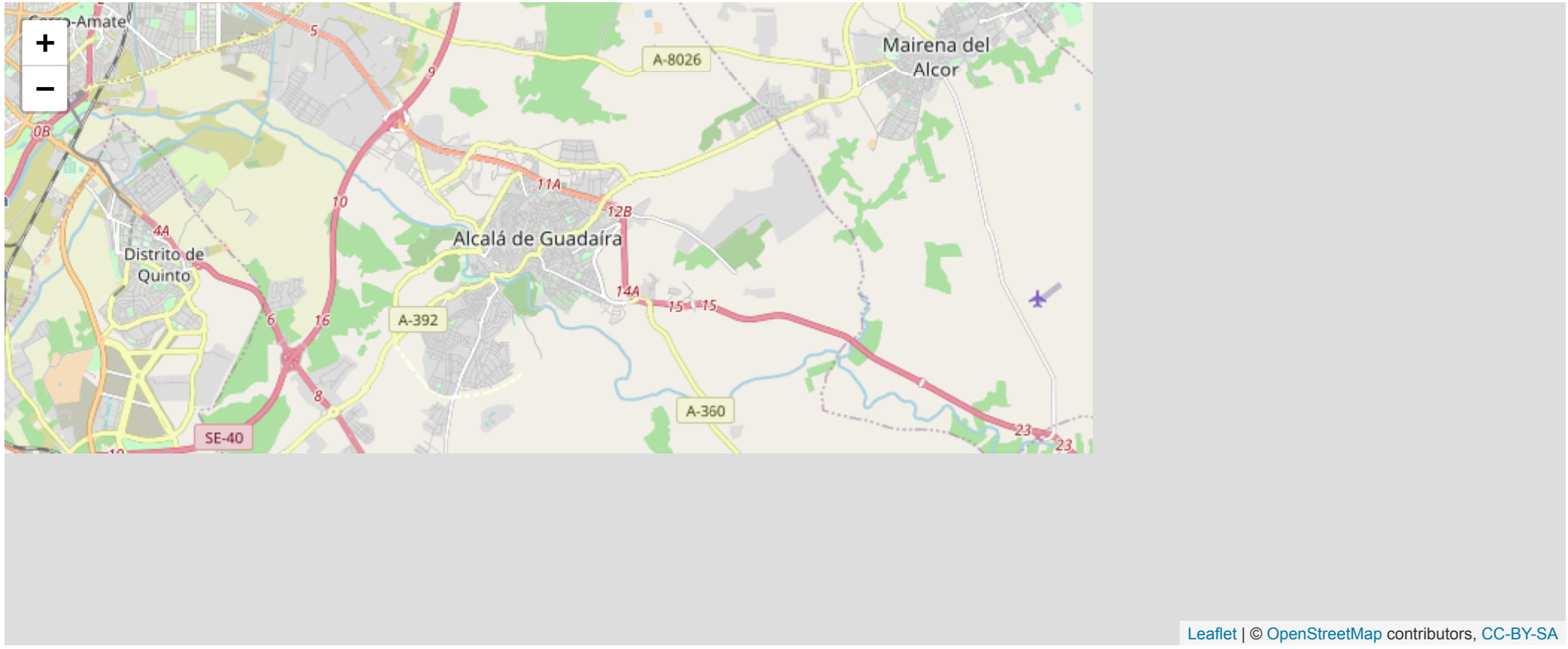
Make point clusters

```
leaflet(palms) %>%
  addTiles() %>%
  addMarkers(lng = ~X, lat = ~Y, icon = palm.icon,
             clusterOptions = markerClusterOptions())
```



Maybe just circles?

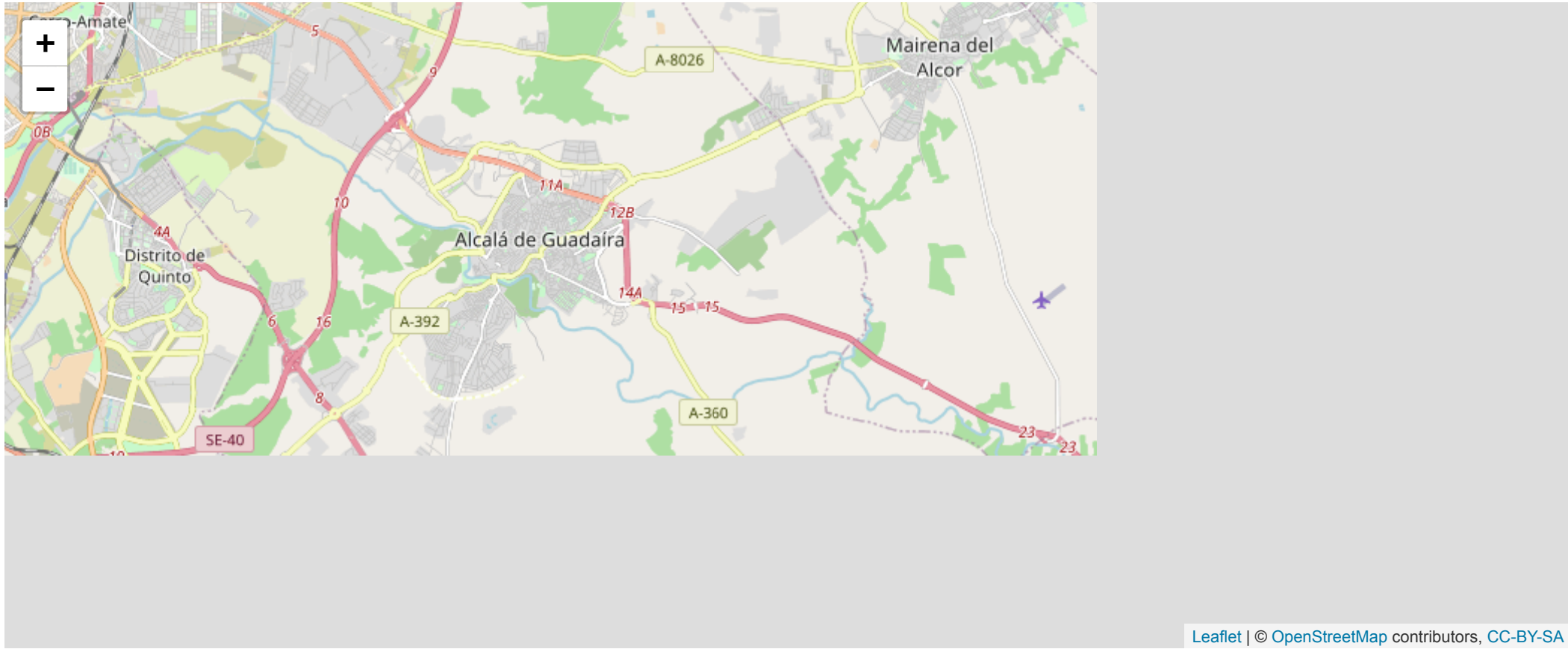
```
leaflet(palms) %>%
  addTiles() %>%
  addCircleMarkers(lng = ~X, lat = ~Y,
                   radius = 5, stroke = FALSE, fillOpacity = 0.7)
```



Changing basemaps

Default tiles: OpenStreetMap

```
leaflet(palms) %>%
  addTiles() %>%
  addCircleMarkers(lng = ~X, lat = ~Y,
                   radius = 5, stroke = FALSE, fillOpacity = 0.7)
```

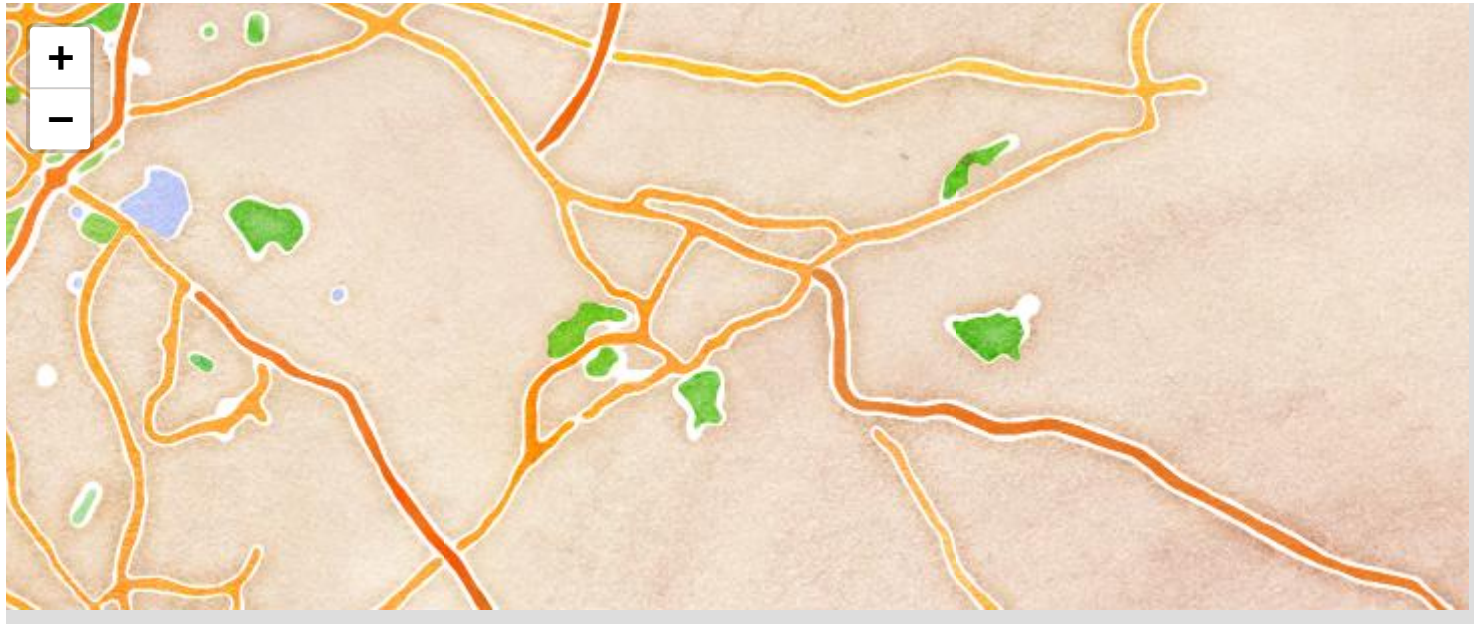


Using other tile providers

```
leaflet(palms) %>%
  addProviderTiles(provider = providers$Esri.WorldImagery) %>%
  addCircleMarkers(lng = ~X, lat = ~Y, radius = 5, stroke = FALSE, fillOpacity = 0.7)
```

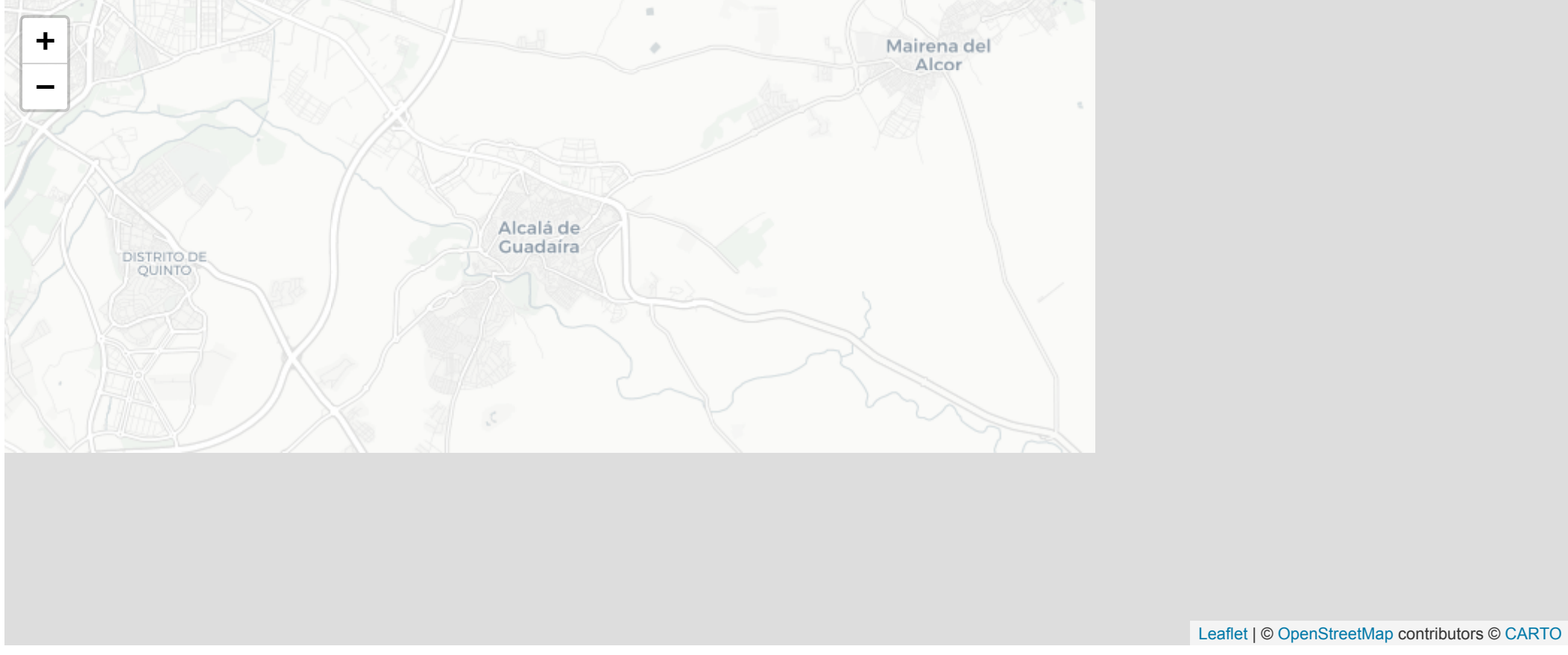
Using other tile providers

```
leaflet(palms) %>%
  addProviderTiles(provider = providers$Stamen.Watercolor) %>%
  addCircleMarkers(lng = ~X, lat = ~Y, radius = 5, stroke = FALSE, fillOpacity = 0.7)
```



Using other tile providers

```
leaflet(palms) %>%
  addProviderTiles(provider = providers$CartoDB.Positron) %>%
  addCircleMarkers(lng = ~X, lat = ~Y, radius = 5, stroke = FALSE, fillOpacity = 0.7)
```



Using WMS tiles

```
leaflet(palms) %>%
  addWMSTiles(baseUrl = "http://www.ign.es/wms-inspire/ign-base",
              layers = "IGNBaseTodo-nofondo") %>%
  addCircleMarkers(lng = ~X, lat = ~Y, radius = 5, stroke = FALSE, fillOpacity = 0.7)
```

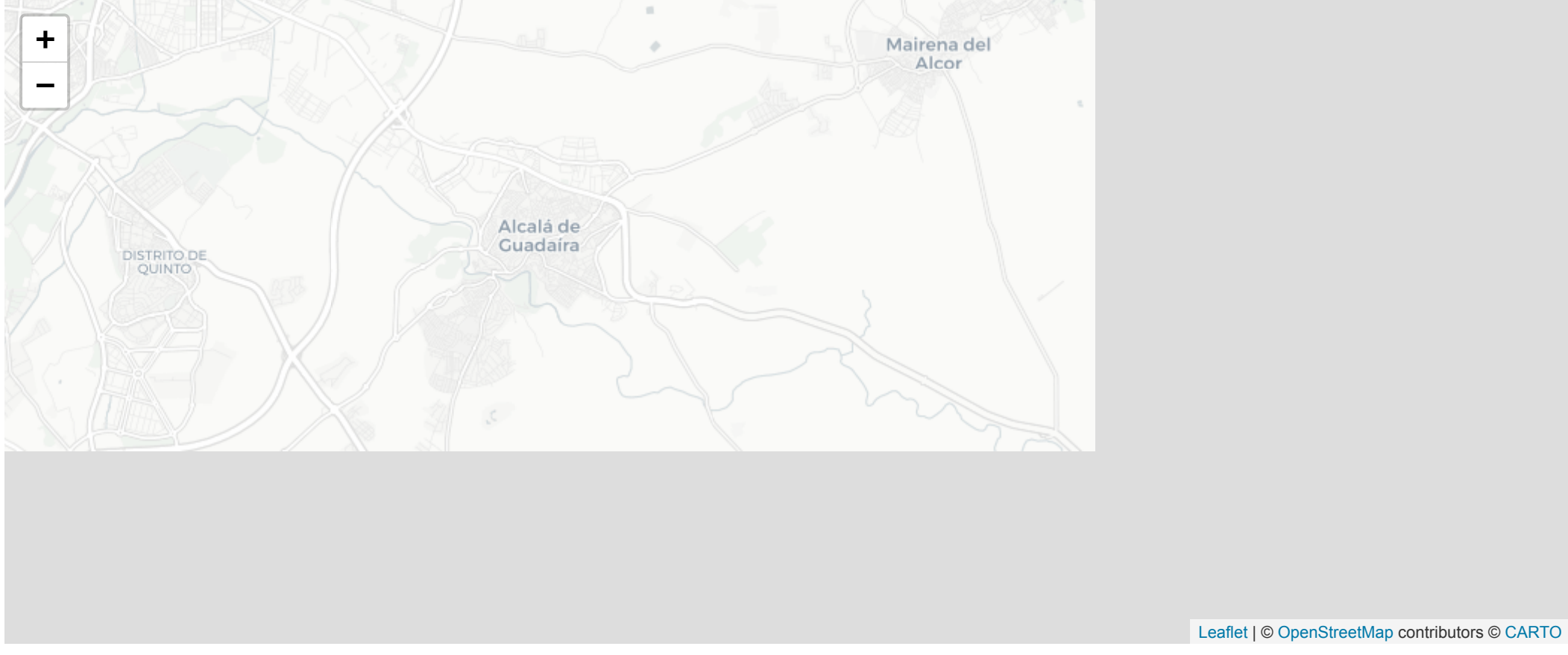
Adding information to points

Point size ~ palm height

```
leaflet(palms) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(lng = ~X, lat = ~Y, stroke = FALSE,
                   radius = ~height)
```

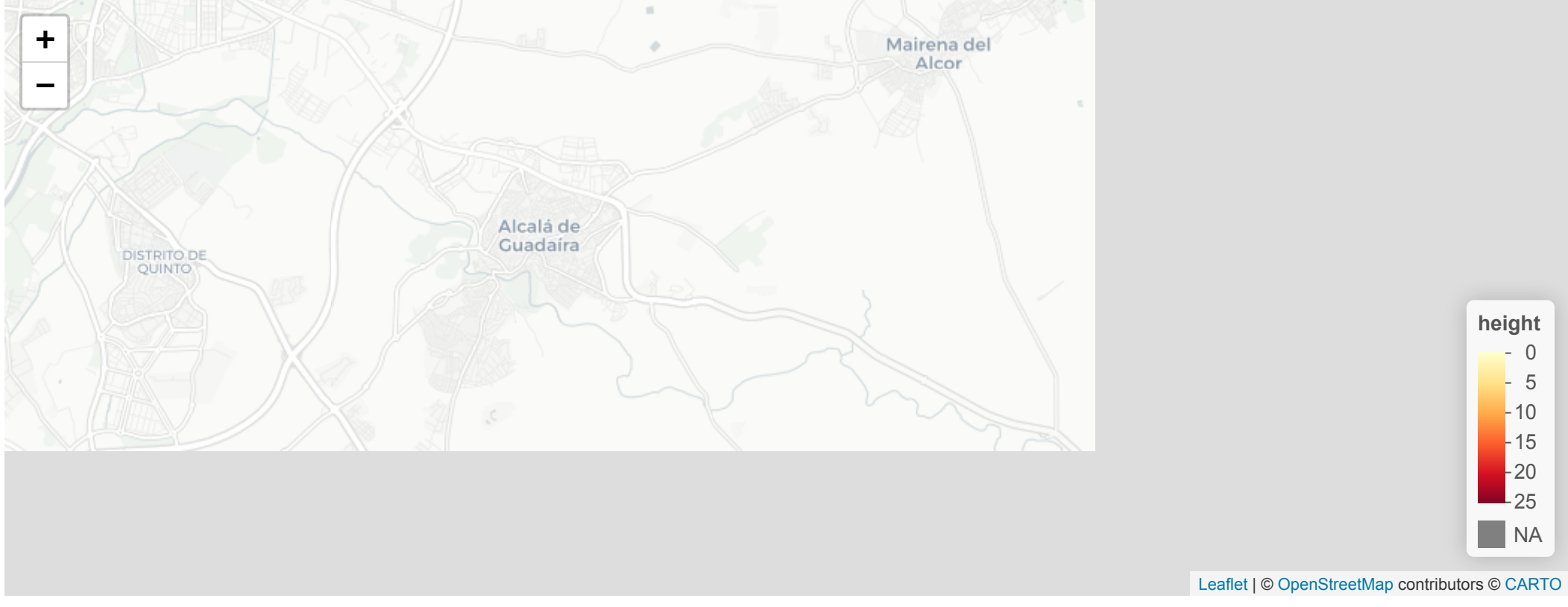
Point colour ~ palm height

```
pal <- colorNumeric(palette = "YlOrRd", domain = palms$height)
leaflet(palms) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(lng = ~X, lat = ~Y,
                   stroke = FALSE, radius = 6, fillOpacity = 0.8,
                   color = ~pal(height))
```



Add legend

```
leaflet(palms, height = '400px') %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(lng = ~X, lat = ~Y, stroke = FALSE, radius = 6,
                   color = ~pal(height), fillOpacity = 0.7) %>%
  addLegend(position = "bottomright",
            pal = pal, values = ~height, opacity = 1)
```

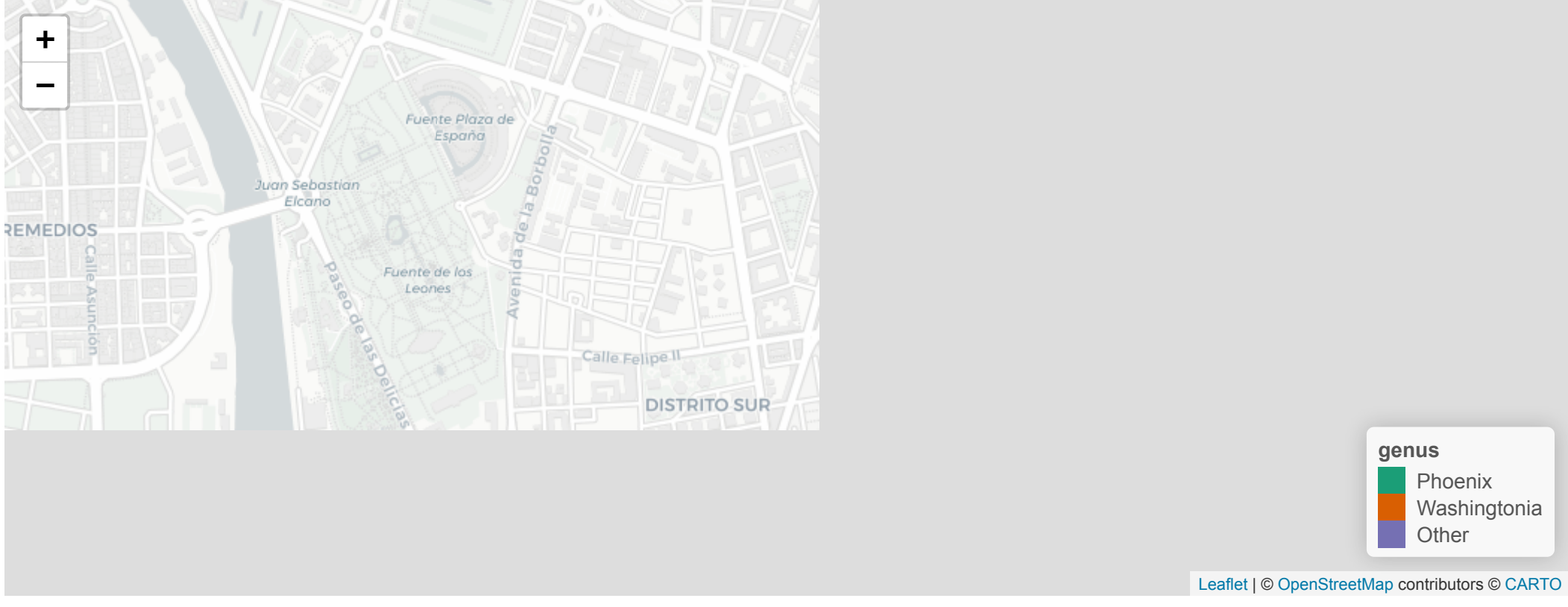


Point colour ~ palm genus (factor)

```
pal.gen <- colorFactor(palette = "Dark2", domain = palms$genus)

leaflet(palms, height = '400px') %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(lng = ~X, lat = ~Y, stroke = FALSE, radius = 6,
                   color = ~pal.gen(genus), fillOpacity = 0.7) %>%
  addLegend(position = "bottomright", pal = pal.gen, values = ~genus, opacity = 1)
```

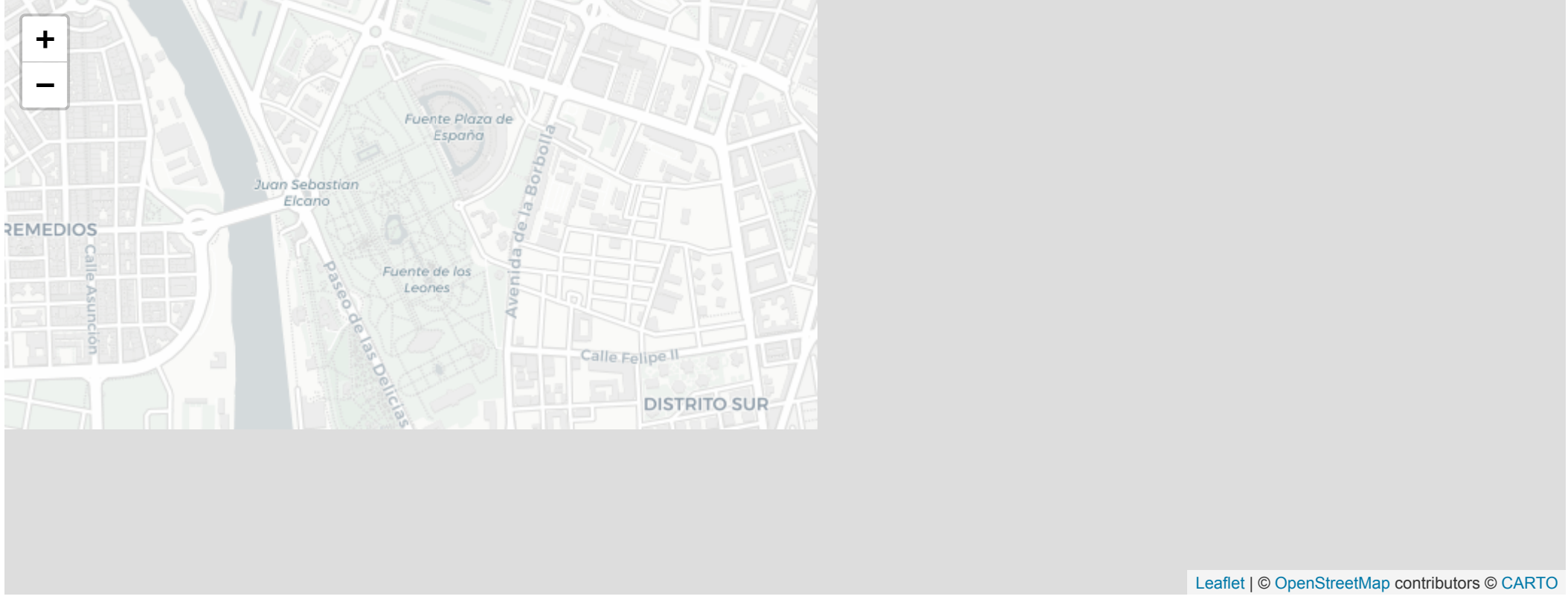
Focus on particular area

```
leaflet(palms, height = '400px') %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(lng = ~X, lat = ~Y, stroke = FALSE, radius = 8,
                   color = ~pal.gen(genus), fillOpacity = 0.7) %>%
  addLegend(position = "bottomright", pal = pal.gen, values = ~genus, opacity = 1) %>%
  setView(lng = -6, lat = 37.38, zoom = 15) # see also fitBounds
```



Adding popups

```
leaflet(palms, height = '400px') %>%
  setView(lng = -6, lat = 37.38, zoom = 15) %>%
  addProviderTiles(provider = providers$CartoDB.Positron) %>%
  addCircleMarkers(lng = ~X, lat = ~Y, stroke = FALSE, radius = 8,
                   color = ~pal.gen(genus), fillOpacity = 0.7,
                   popup = ~species)
```

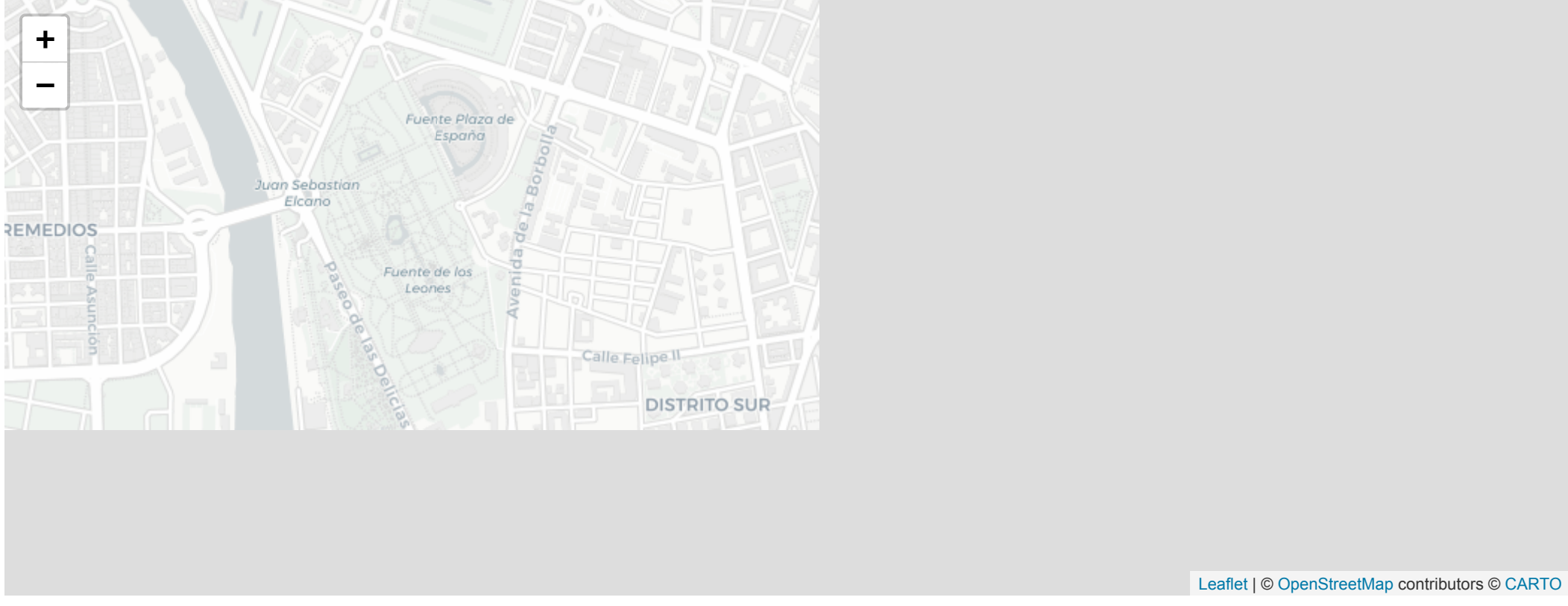


Advanced popups with `leafpop`

https://github.com/r-spatial/leafpop

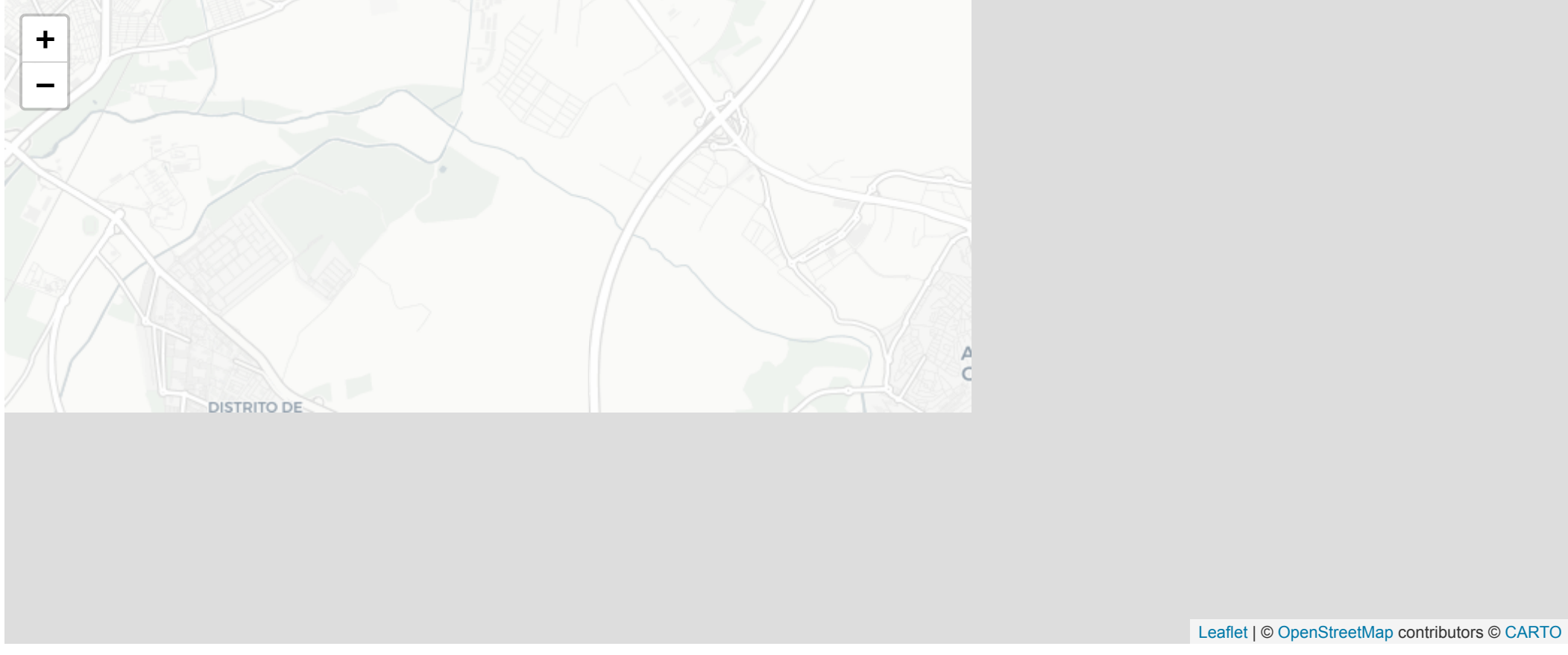
Show tables with extra information

```
leaflet(palms, height = '400px') %>%
  setView(lng = -6, lat = 37.38, zoom = 15) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(lng = ~X, lat = ~Y, stroke = FALSE, radius = 8,
                   color = ~pal.gen(genus), fillOpacity = 0.7,
                   popup = ~leafpop::popupTable(palms,
                                                zcol = c("species", "perimeter", "height"),
                                                row.numbers = FALSE, feature.id = FALSE))
```

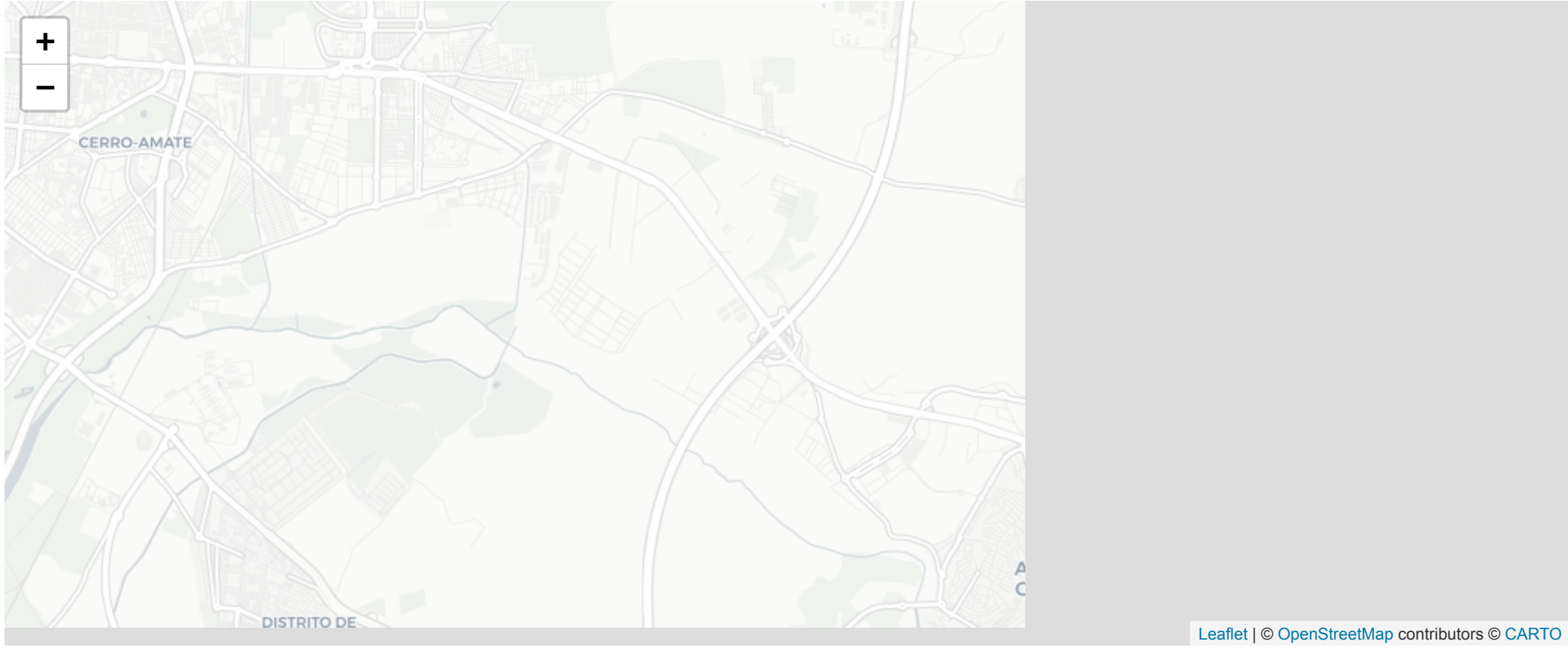


Popups can show images, videos, anything!

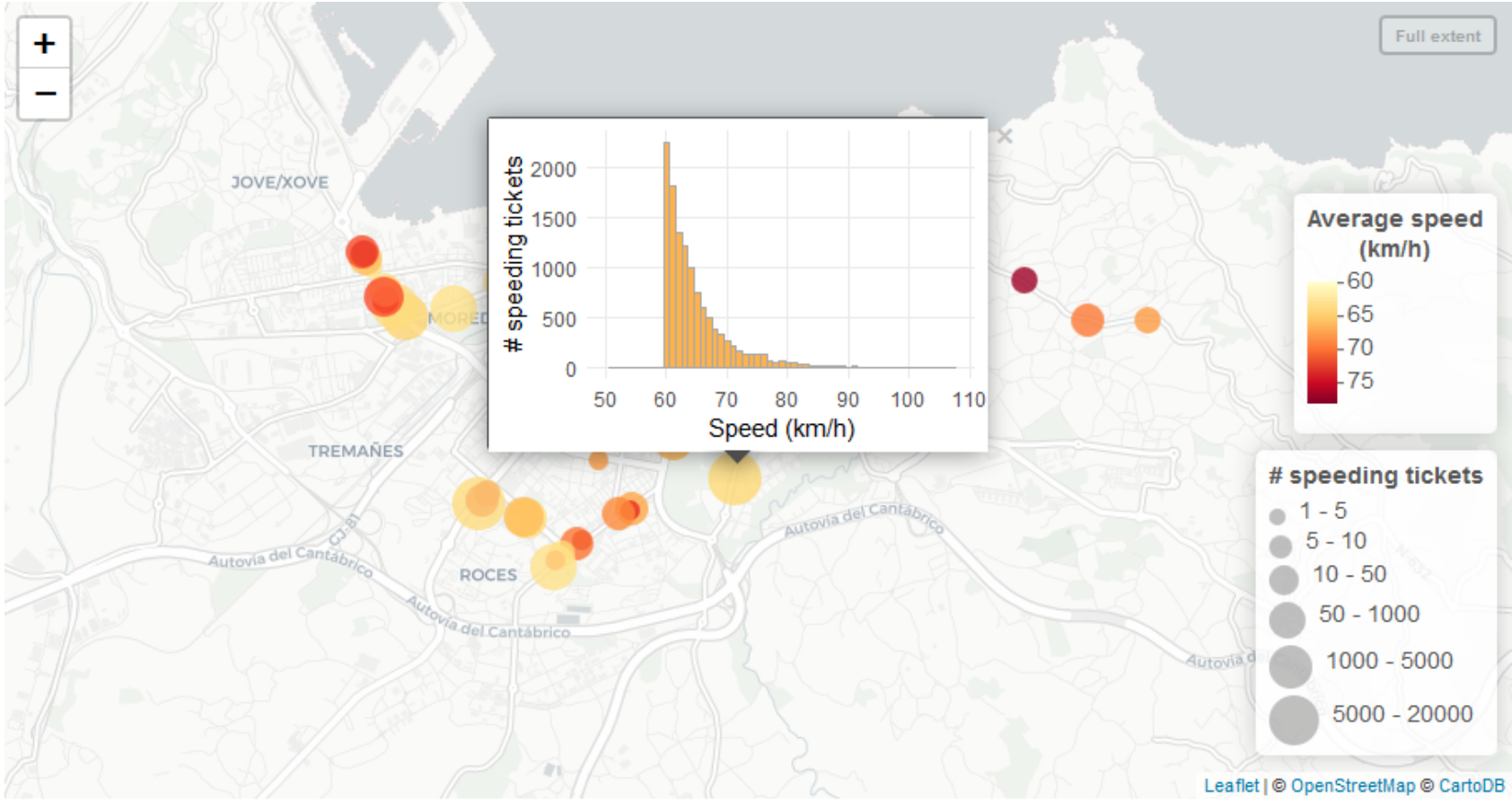
```
leaflet(palms.redux) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(lng = ~X, lat = ~Y, stroke = FALSE, radius = 8,
                   color = ~pal.gen(genus), fillOpacity = 0.7,
                   popup = ~leafpop::popupImage(img = palms.redux$pics,
                                                src = "local", embed = TRUE))
```



Popups can show images, videos, anything!



R graphics as popups



https://pakillo.github.io/R-Asturias-DataViz-Contest/speedmap.html

Mapping several layers

Show/hide layers (data or basemaps)

```
leaflet(height = '400px') %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(data = phoenix, group = "Phoenix",
                   lng = ~X, lat = ~Y, stroke = FALSE, radius = 6,
                   color = ~pal.gen(genus), fillOpacity = 0.7) %>%
  addCircleMarkers(data = washingtonia, group = "Washingtonia",
                   lng = ~X, lat = ~Y, stroke = FALSE, radius = 6,
                   color = ~pal.gen(genus), fillOpacity = 0.7) %>%
  addLayersControl(overlayGroups = c("Phoenix", "Washingtonia"),
                   options = layersControlOptions(collapsed = FALSE))
```

Small multiples of leaflet maps with `leafsync`

https://github.com/r-spatial/leafsync

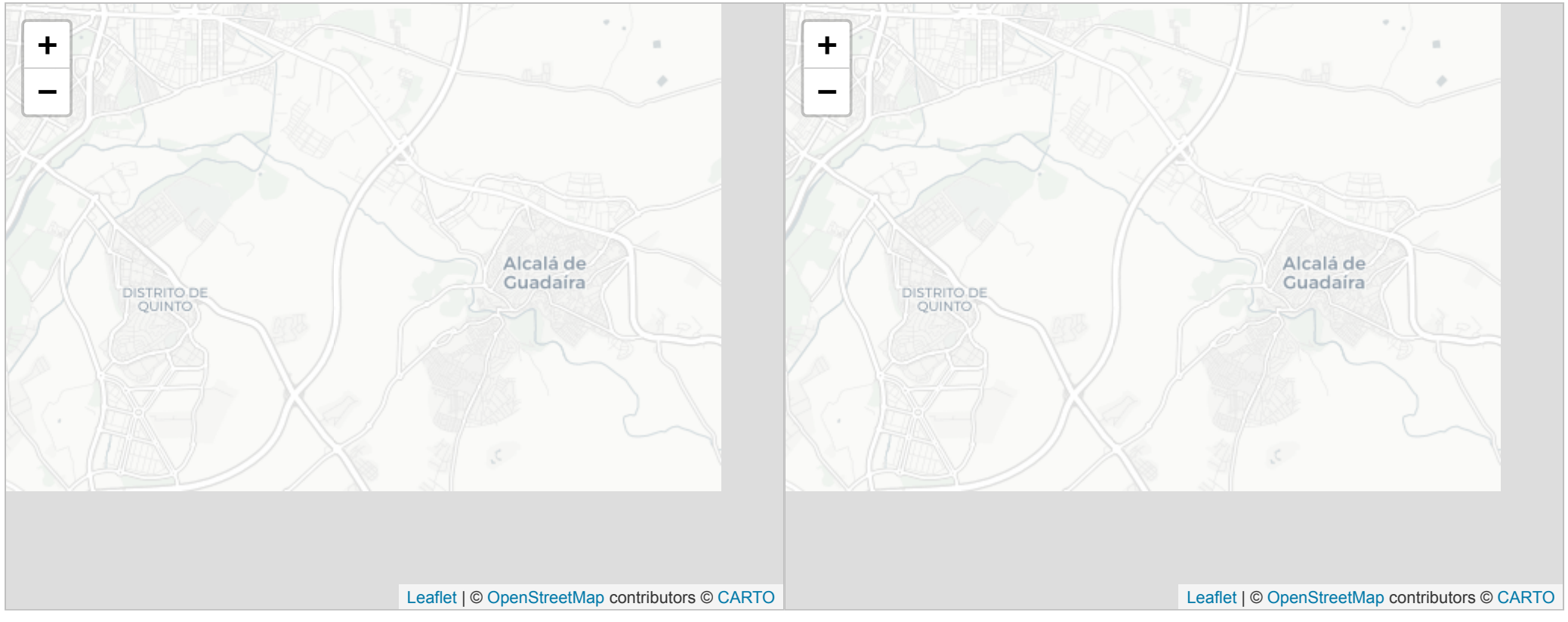
Let's build two leaflet maps

```
phoenix.map <- leaflet() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(data = phoenix,
                   lng = ~X, lat = ~Y, stroke = FALSE, radius = 6,
                   color = ~pal.gen(genus), fillOpacity = 0.7)
```

```
washingtonia.map <- leaflet() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(data = washingtonia,
                   lng = ~X, lat = ~Y, stroke = FALSE, radius = 6,
                   color = ~pal.gen(genus), fillOpacity = 0.7)
```

Synchronised small multiples!

```
leafsync::sync(phoenix.map, washingtonia.map)
```



Mapping other spatial data

Data can be

- Points
- Lines
- Polygons
- Rasters

- Matrix/Dataframe
- sp
- sf
- GeoJSON/TopoJSON
- raster

Sevilla neighbourhoods

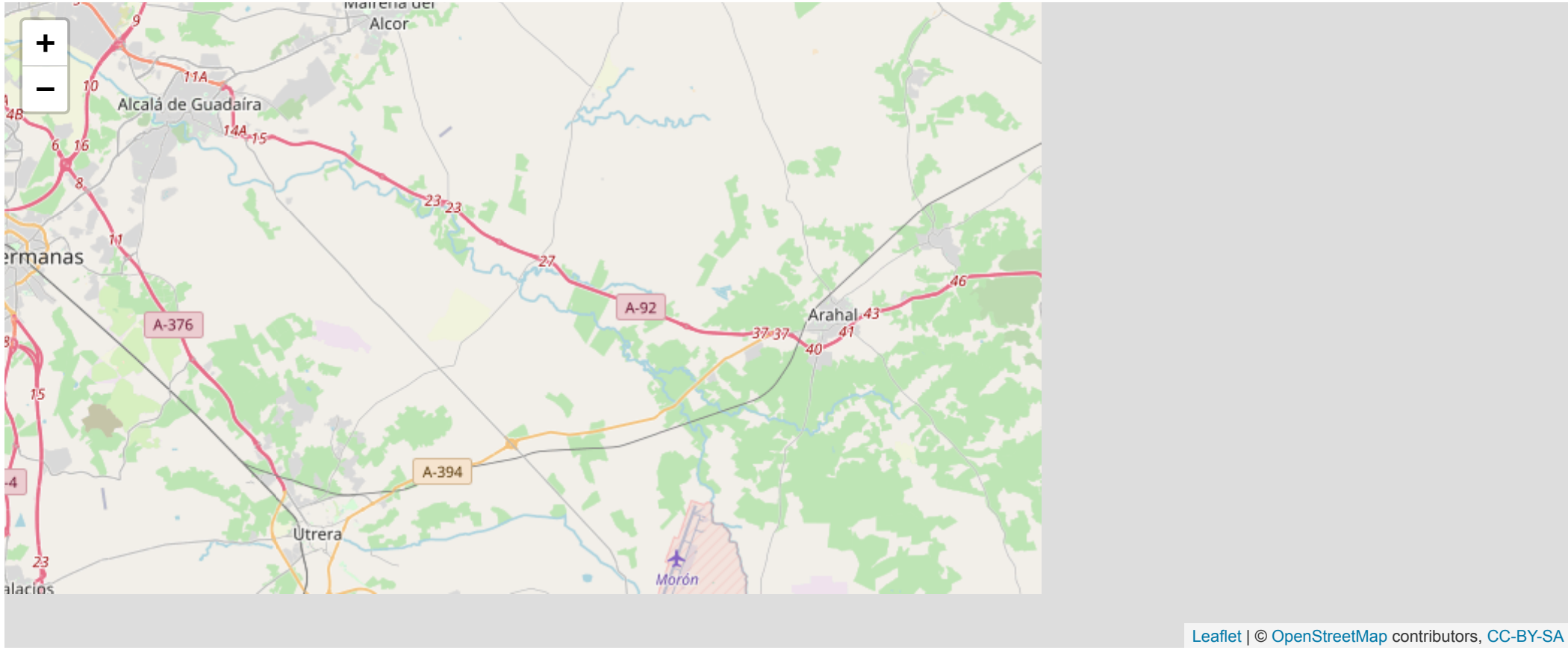
```
barrios <- sf::st_read("data/Barrios.shp")
```

```
## Reading layer `Barrios' from data source
##   `/home/frs/Dropbox/Rcode/myRcode/courses_talks/r-leaflet-maps/data/Barrios.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 109 features and 7 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -6.03342 ymin: 37.30045 xmax: -5.818405 ymax: 37.45294
## Geodetic CRS:  WGS 84

## Simple feature collection with 6 features and 7 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -6.029277 ymin: 37.31342 xmax: -5.818405 ymax: 37.45294
## Geodetic CRS:  WGS 84
##   FID                Barrio DISTRITO                  DISTRITO_N Superf_Ha
## 1   1               TABLADA       11                Los Remedios  1332.013
## 2   2 COLORES, ENTREPARQUES        9 Este - Alcosa - Torreblanca  1331.235
## 3   3         LA BACHILLERA        7                       Norte  1327.843
## 4   4           TORREBLANCA        9 Este - Alcosa - Torreblanca  1335.412
## 5   5            BELLAVISTA       10     Bellavista - La Palmera  1165.637
## 6   6           EL GORDILLO        7                       Norte  1179.351
##    Shape__Are Shape__Len                       geometry
## 1 0.001353248  0.1550361 MULTIPOLYGON (((-6.002653 3...
## 2 0.001353701  0.2112231 MULTIPOLYGON (((-5.876799 3...
## 3 0.001350621  0.2632040 MULTIPOLYGON (((-5.954878 3...
## 4 0.001357710  0.2489455 MULTIPOLYGON (((-5.856267 3...
## 5 0.001184084  0.1850147 MULTIPOLYGON (((-5.953938 3...
## 6 0.001199665  0.1871865 MULTIPOLYGON (((-5.935078 3...
```

Source: http://sevilla-idesevilla.opendata.arcgis.com/datasets/38827fc3eac142149801c2efa2a0bdf9_0

Mapping polygons

```
leaflet(barrios) %>%
  addTiles() %>%
  addPolygons(label = ~Barrio)
```



mapview: quick leaflet maps of any spatial data

```
mapview::mapview(barrios)
```

https://r-spatial.github.io/mapview/

For big datasets, try

https://github.com/r-spatial/leafgl

https://symbolixau.github.io/mapdeck/

There are hundreds of leaflet plugins

(not all implemented in R yet)

Add MiniMap

```
leaflet(barrios) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addPolygons() %>%
  addMiniMap()
```

Measure distances

```
leaflet(barrios) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addPolygons() %>%
  addMeasure()
```

Locate me (using Javascript)

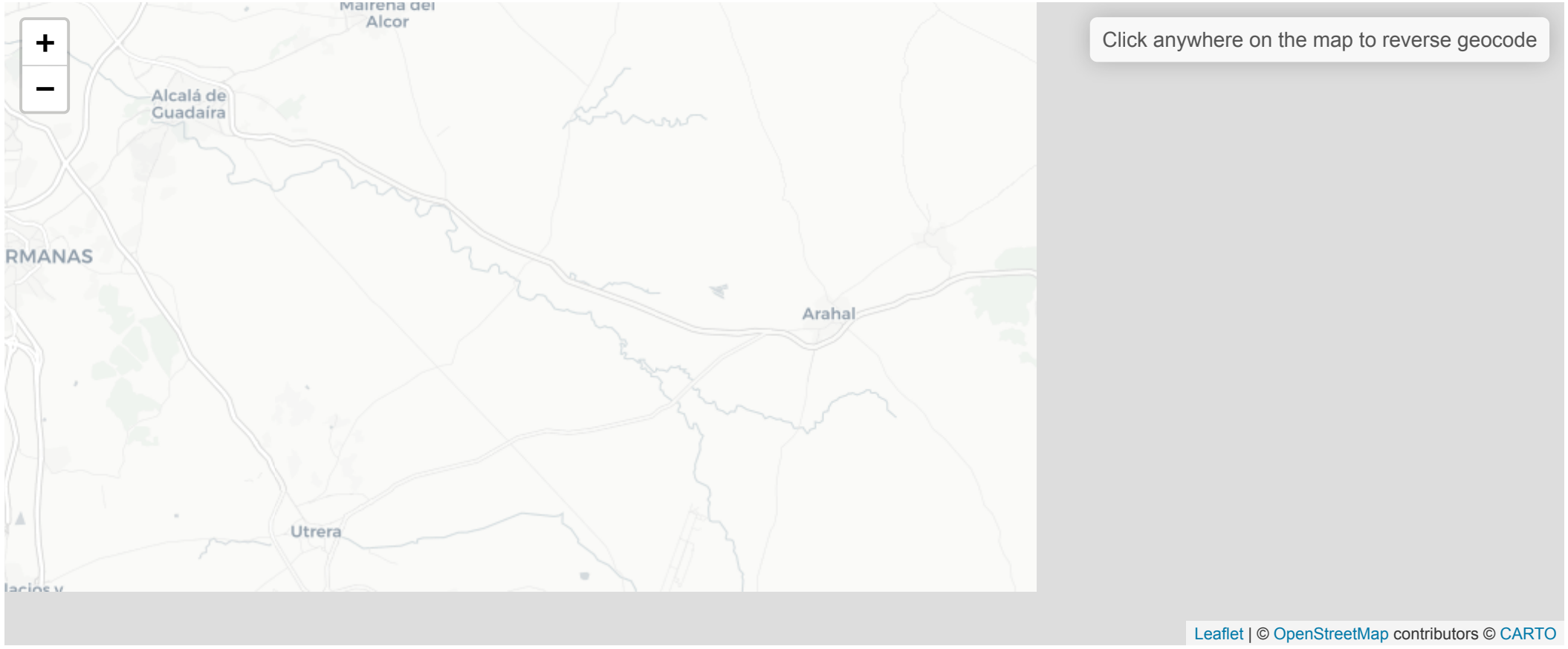
```
leaflet(barrios) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addPolygons() %>%
  addEasyButton(easyButton(
    icon = "fa-crosshairs", title = "Locate Me",
    onClick = JS("function(btn, map){ map.locate({setView: true}); }")))
```

Search places

```
library("leaflet.extras")

leaflet(barrios) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addPolygons() %>%
  addSearchOSM()
```

Add reverse search

```
leaflet(barrios) %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addPolygons() %>%
  addReverseSearchOSM()
```



Animated maps

```
library("leaflet.minicharts")
```

END

Slides and source code available at https://github.com/Pakillo/r-leaflet-maps