Maps in R

ESTP Use of R in Official Statistics

- *Creating maps*
- graphics in R

Spatial data

Data often has a spatial component:

- differences in income between municipalities/neighborhoods
- population density
- companies, high tech, agricultural
- road density
- mobility, migration
- land use etc.

You are often interested in (the relevance of) areas or locations.

GIS

- software for geographical data is called GIS.
- *Geographical Information Systems*
- Broad umbrella term for spatial analysis, processing and * cartography *.
- Cartography is only part of this.

Spatial data

Spatial or spatial data has different forms:

Vector data (`sf`)

- *point* data: location with (measured) values for that location. E.g. locations of theft, fire.
- *areal*/*polygon* data: boundaries of areas/regions, municipalities , police regions etc. with associated data
- *line* data: roads/rivers with associated data.

Raster data (`raster`)

- Grid data: the data is a collection of rectangles/pixels with data
- Eg: satellite photos, infrared, drone data.

CRS: coordinate reference system

- All cards use a CRS (coordinate system)
- In the Netherlands this is often the National Triangle System (rd, EPSG: 28992)

The earth is (approximately) a sphere, but a map is flat:

- Every card is distorted (is not a perfect image).
- that is why there are many ways to smash (a piece of) the earth.
- in Geo data you occasionally have to transform from one CRS to another CRS.
- sounds simple, but the details are not (for example, earth pollen moving)

Google/Bing maps (WGS84, Webmercator)

- Due to the rise of GPS, lat/lon coordination is often used (sphere coordinates). (WGS84)
- Many systems also use this convention, so sometimes you have to transform data from this form to your desired CRS.

sf (*simple features*)

- point, polygon, line data.
- Supports union, intersections etc.
- crs transformations
- join with data (`dplyr`)
- writing and reading of many GIS vector formats
- (used geos, gdal, proj)

grid

- multi-layer raster data
- writing and reading many GIS raster formats
- can handle very large frames (by leaving data on disk)
- help functions for converting vector -> raster (and back).

sf

Data is a “data.frame” with an additional “geometry” column.

read/write

- `st_read` read (many) different formats: geojson, geopackage and shp (ESRI)
- `st_write` write to different formats

Manipulate

- ‘create, modify, association, intersection etc. of” geometry ”

Calculate

- buffer query, interpolation, etc

Maps

Different applications:

- Topography: showing infrastructure
- Satellite: show
- Cartography: display of (statistical) information (Dutch: Bos Atlas!)
- etc.

Thematic cards

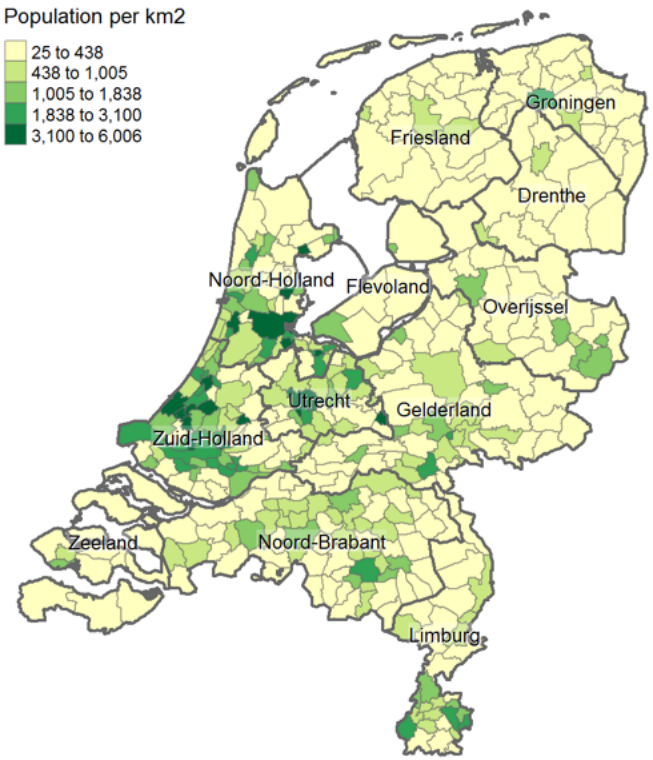
A ** thematic map ** is a visualization in which statistical information (theme) with a geographical component is shown.

- Cartography: communication (*popular!*)
- Analytical purposes: insight, is there a spatial pattern?
- This is typically an application that you will have as a Data Scientist.

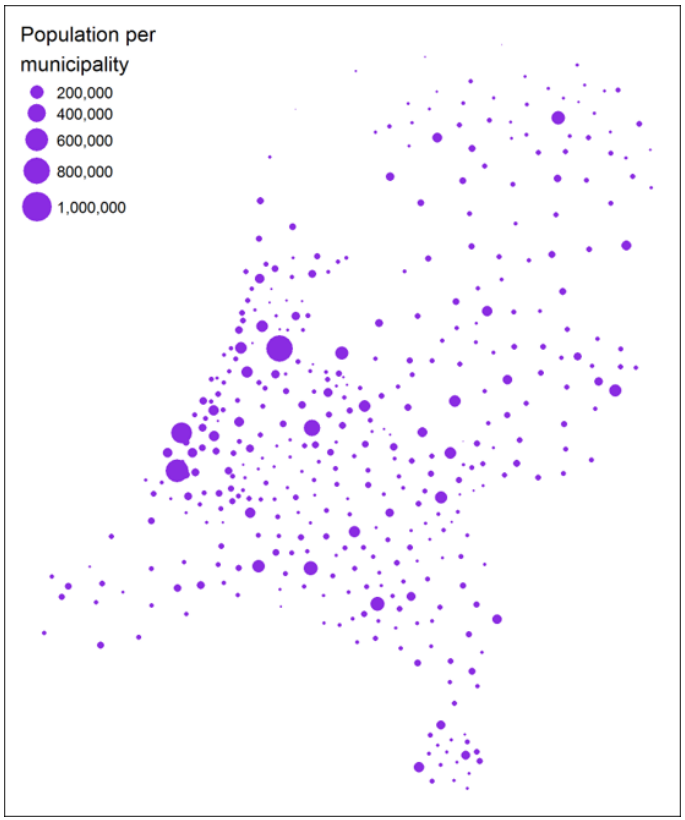
Species

- Choropleet
- Bubble map
- Contour map (isopleet)
- Raster/Density map
- Cartogram

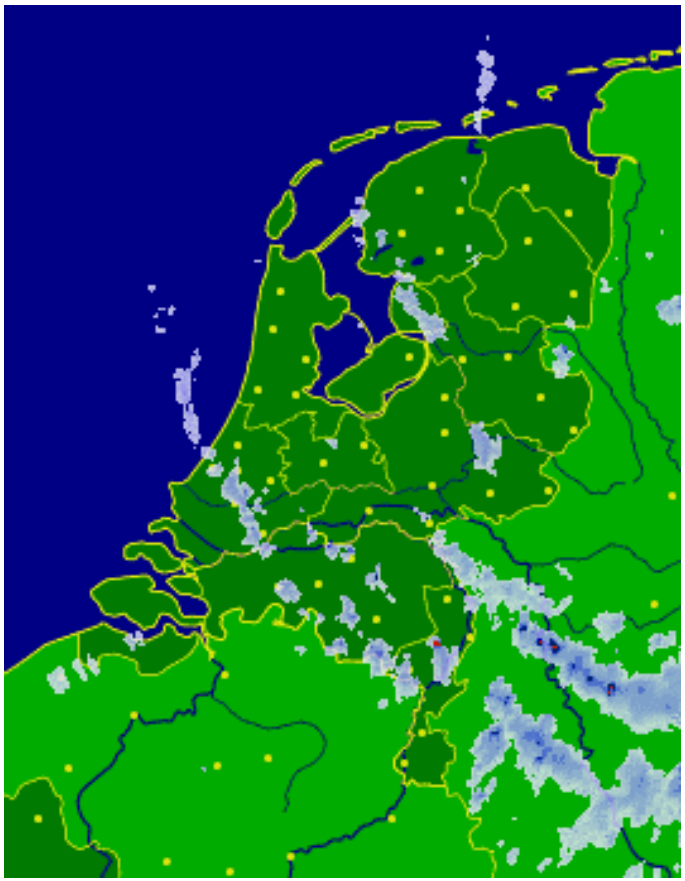
Choropleth


Population per km2
25 to 438
438 to 1,005
1,005 to 1,838
1,838 to 3,100
3,100 to 6,006
Groningen
Friesland
Drenthe
Noord-Holland
Flevoland
Overijssel
Utrecht
Gelderland
Zuid-Holland
Zeeland
Noord-Brabant
Limburg

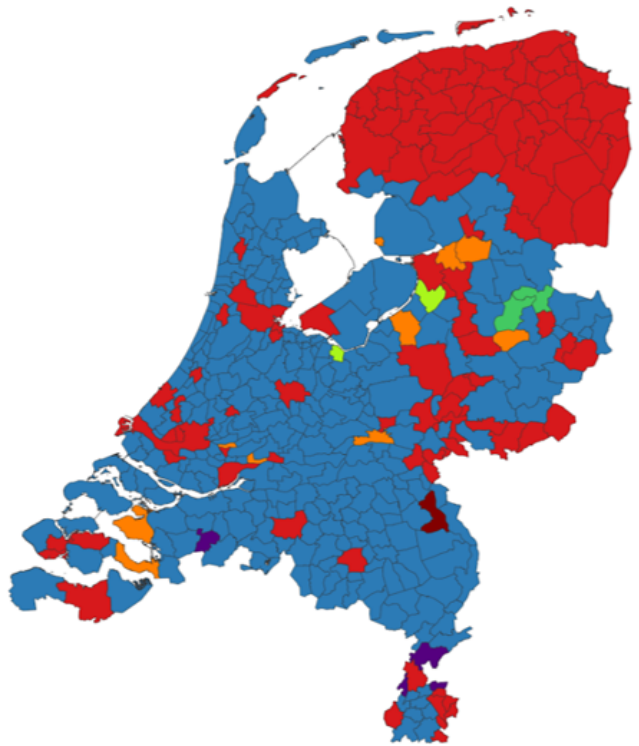
Bubble Chart



Raster data

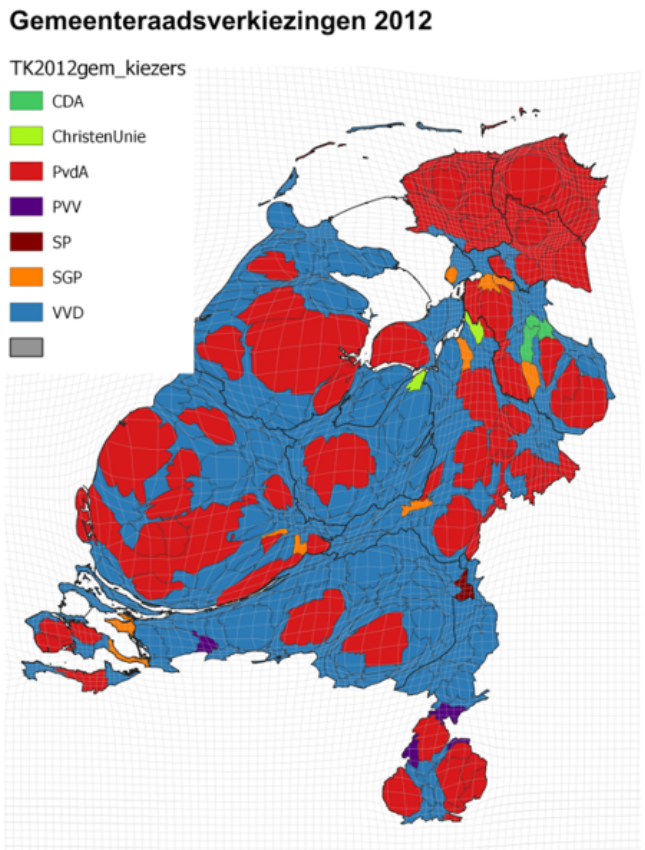


Cartogram



Cartogram

European Commission



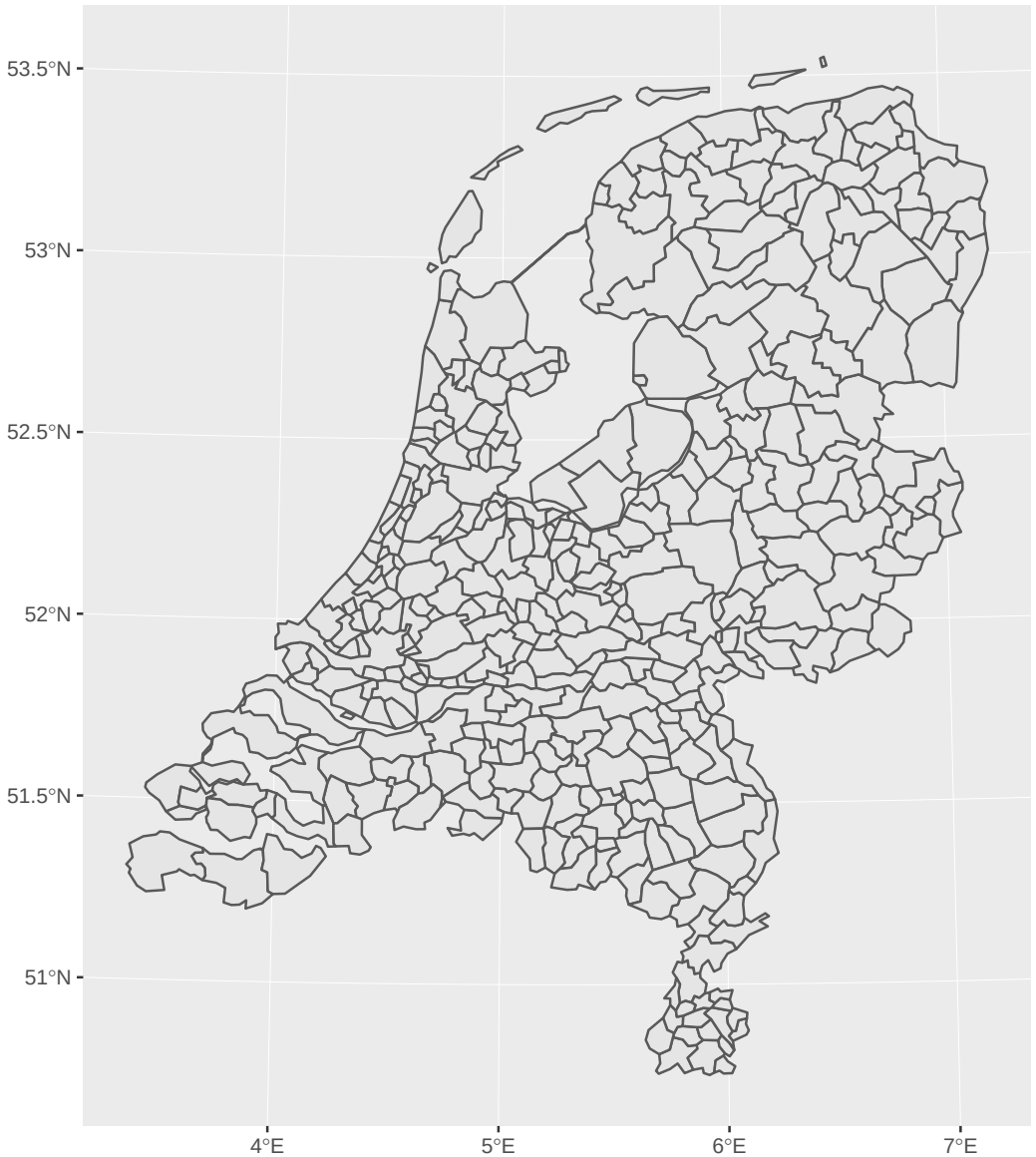
Making maps in R

Different options:

- `tmap`: very extensive options with little code, static + interactive plots
- `ggplot2:geom_sf`, static plots
- `mapview`: interactive plots
- 'leaflet ': interactive plot

ggplot2 :: geom_sf

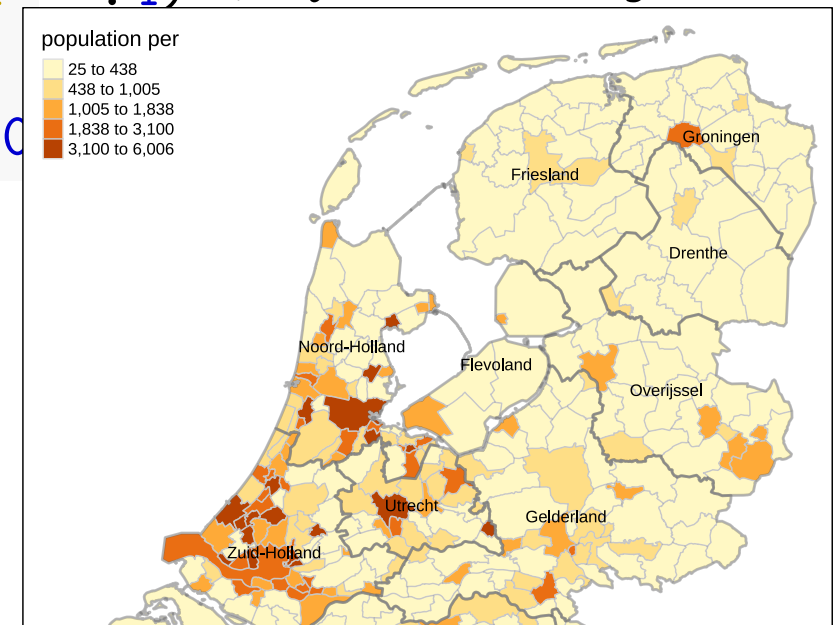

```
library (ggplot2)
ggplot (gm_2017) + geom_sf () + coord_sf ()
```


53.5°N
53°N
52.5°N
52°N
51.5°N
51°N
4°E
5°E
6°E
7°E

tmap

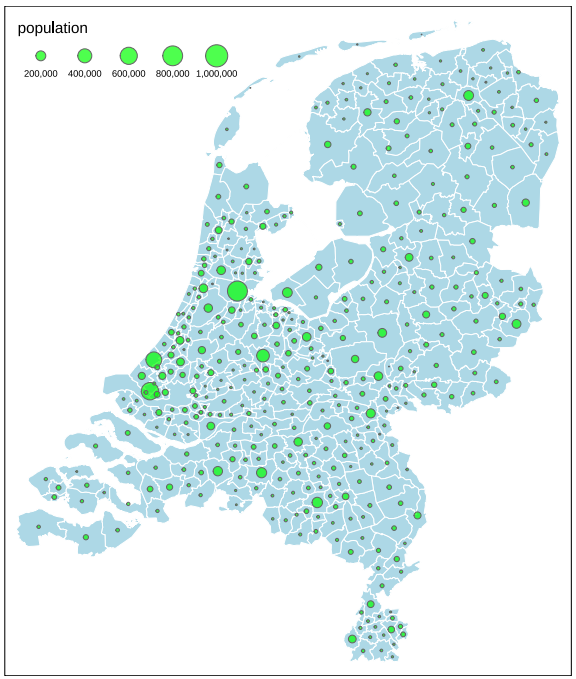
```
tm_shape(NLD_muni) +
  tm_fill( "population"
         , convert2density = TRUE
         , style="kmeans") +
  tm_borders(alpha=0.2) +

  tm_shape(NLD_prov) +
  tm_borders(lwd=2, alpha = .4) +

  tm_text("name", size = 0
```

```
## old-style crs object detected; plea
## old-style crs object detected; plea
## old-style crs object detected; plea
## old-style crs object detected; plea
## old-style crs object detected; plea
## old-style crs object detected; plea
## old-style crs object detected; plea
## old-style crs object detected; plea
```



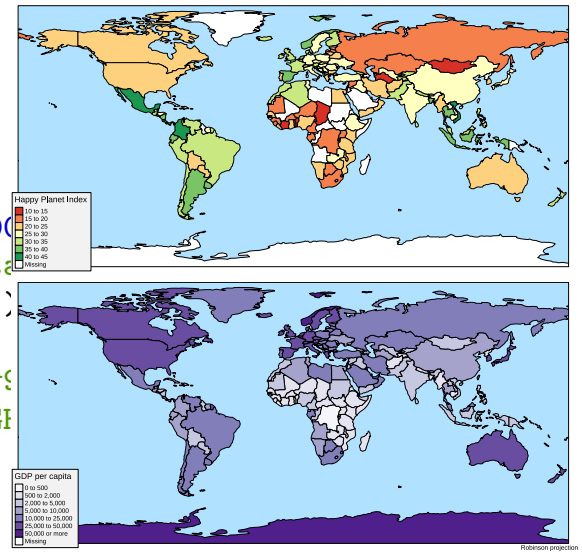
Bubble map

```
tm_shape(NLD_muni) +
  tm_fill("lightblue") +
  tm_borders("white") +
  tm_bubbles("population"
             , col="green"
             , alpha=0.7
             )
```



Example


```
data(World)
tm_shape(World) +
  tm_polygons(c("HPI", "gdp_cap_est"),
              palette = list("RdYlGn", "Purples"),
              style = c("pretty", "fixed"), n = 7,
              breaks = list(NULL, c(0, 500, 2000, 5000, 10000
              title = c("Happy Planet Index", "GDP per capita
  tm_style("natural", earth.boundary = c(-180, -87, 180, 87))
  tm_format("World", inner.margins = 0.02, frame = FALSE) +
  tm_legend(position = c("left", "bottom"), bg.color = "grayS
  tm_credits(c("", "Robinson projection"), position = c("RIGH
```



Geocoding

Finding locations with addresses is called **geocoding**

nlgeocoder


```
library(tmaptools)
library(leaflet)

loc <- geocode_OSM("Turfmarkt, den Haag", as.sf = TRUE)
leaflet(loc) %>%
  addTiles() %>%
  addCircleMarkers(popup = ~paste0(query))
```