

Welcome to **instats**

The Session Will Begin Shortly

START

Spatial Data Analysis and Visualization in R

Session 24: Extensibility of tmap

instats

Extending tmap: Overview

`tmap` can be extended in **three directions**:

1. **Modes**: how maps are rendered (e.g. static, interactive, 3D)
2. **Layer types**: kinds of layers (e.g. polygons, lines)
3. **Spatial object support**: object classes (e.g. `sf` objects of `SpatRaster` from `terra`)

Current extension packages

| Package | Extends Mode | New Layers | New Object Types |
|----------------|--------------|-----------------------|------------------|
| tmap.mapgl | ✓ | ✓ (tm_polygons_3d) | ✗ |
| tmap.cartogram | ✗ | ✓ (tm_cartogram()) | ✗ |
| tmap.glyphs | ✗ | ✓ (tm_glyph()) | ✗ |
| tmap.networks | ✗ | ✓ (tm_network()) | ✓ (ifnetwork) |

Modes

Table: tmap Modes and Backends

| Mode | Description | R Package | R Backend | JS Library |
|-------------------------|---------------------|--------------------------|---------------------------|----------------|
| <code>plot</code> | Static graphics | tmap | grid | <i>(none)</i> |
| <code>view</code> | Interactive web map | tmap | leaflet and family | Leaflet |
| <code>mapbox</code> | Interactive web map | tmap.mapgl | mapgl | Mapbox GL JS |
| <code>maplibre</code> | Interactive web map | tmap.mapgl | mapgl | MapLibre GL JS |
| <code>highcharts</code> | Interactive web map | <i>to be implemented</i> | <code>highcharter</code> | Highcharts |

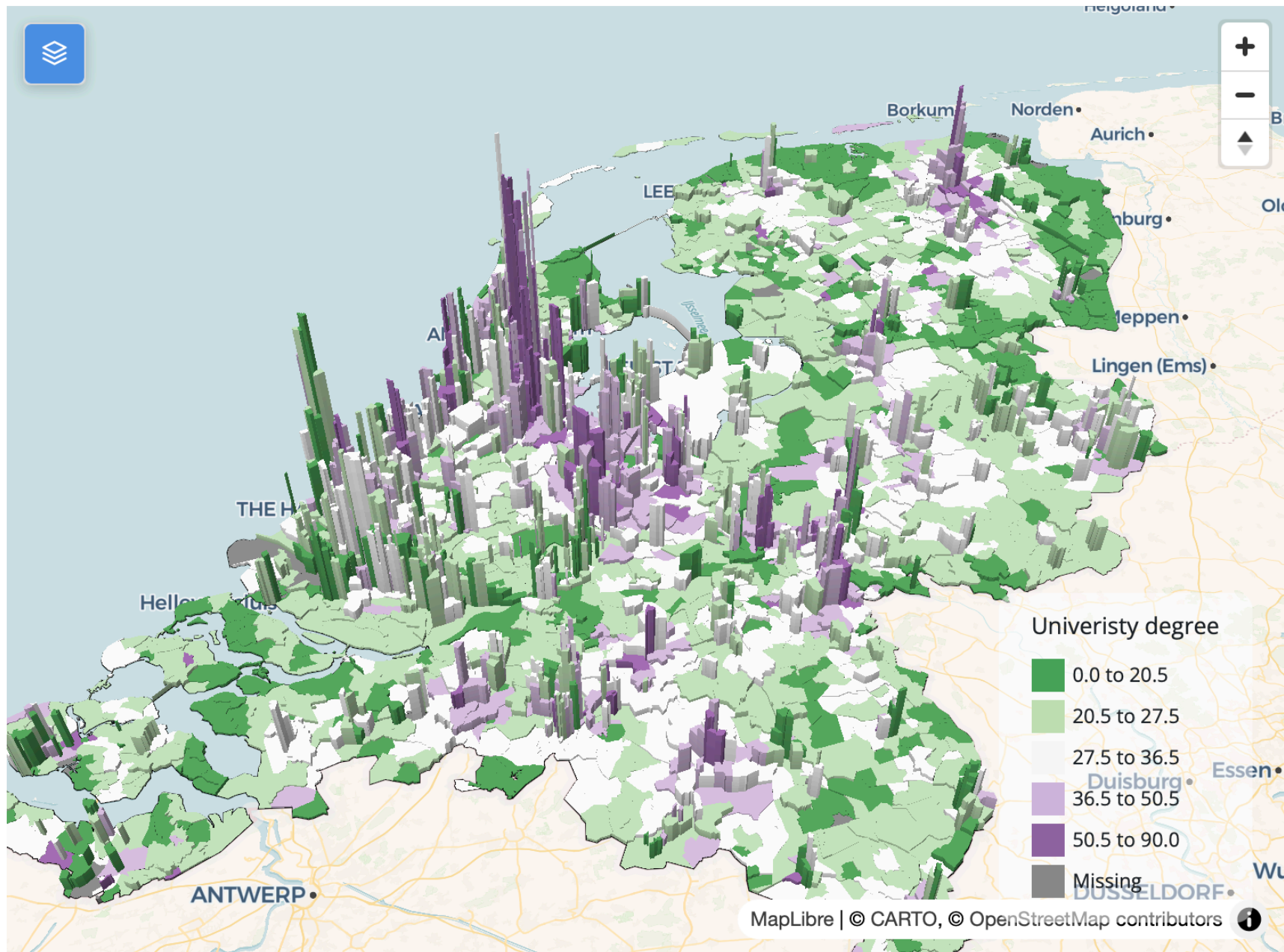
tmap.mapgl: two new modes

- **tmap.mapgl** uses **Mapbox GL JS** and **MapLibre GL JS** via the R package **mapgl**
- Adds the layer **tm_polygons_3d()**

```
tmap_mode("maplibre")
#> i tmap mode set to "maplibre".

NLD_dist$pop_dens = NLD_dist$population / NLD_dist$area

tm_shape(NLD_dist) +
  tm_polygons_3d(height = "pop_dens",
    fill = "edu_appl_sci",
    fill.scale = tm_scale_intervals(style = "kmeans", values = "-pu_gn"),
    fill.legend = tm_legend("Univeristy degree")) +
tm_maplibre(pitch = 45)
```



Cartograms with `tmap.cartogram`

Visualize distorted shapes based on variable values.

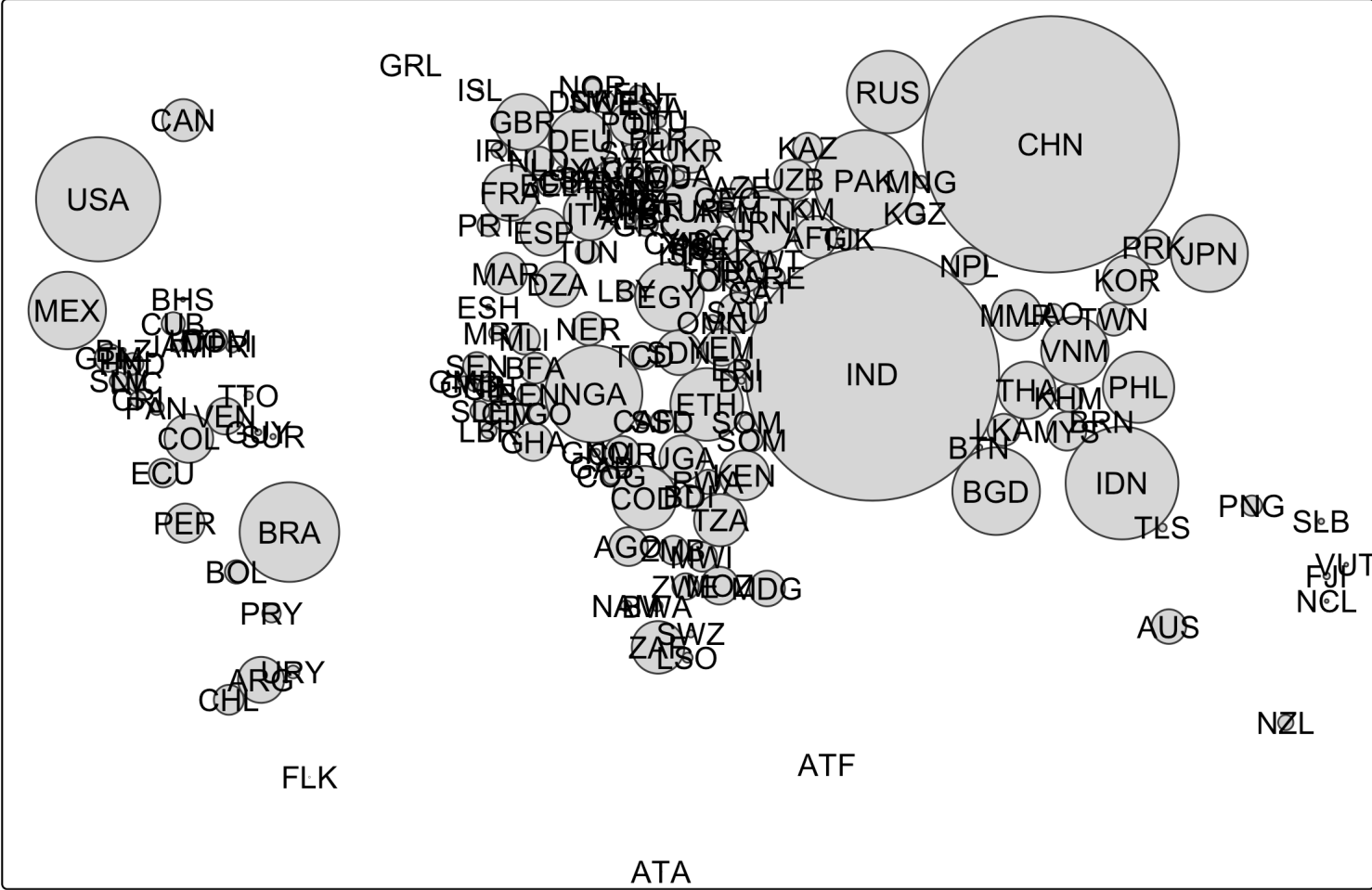
- Layer: `tm_cartogram()`
- Uses the **cartogram** package internally

```
library(tmap)
library(tmap.cartogram)

tm_shape(World, crs = "auto") +
  tm_cartogram_dorling("pop_est") +
  tm_text("iso_a3")
```

Methods supported: `"dorling"`, `"noncont"`, `"cont"`.

Cartograms with `tmap.cartogram`



Glyph Maps with `tmap.glyphs`

Combine multiple variables into small graphs per feature.

- Layer: `tm_donut()` and `tm_flower()`

```
library(tmap.glyphs)

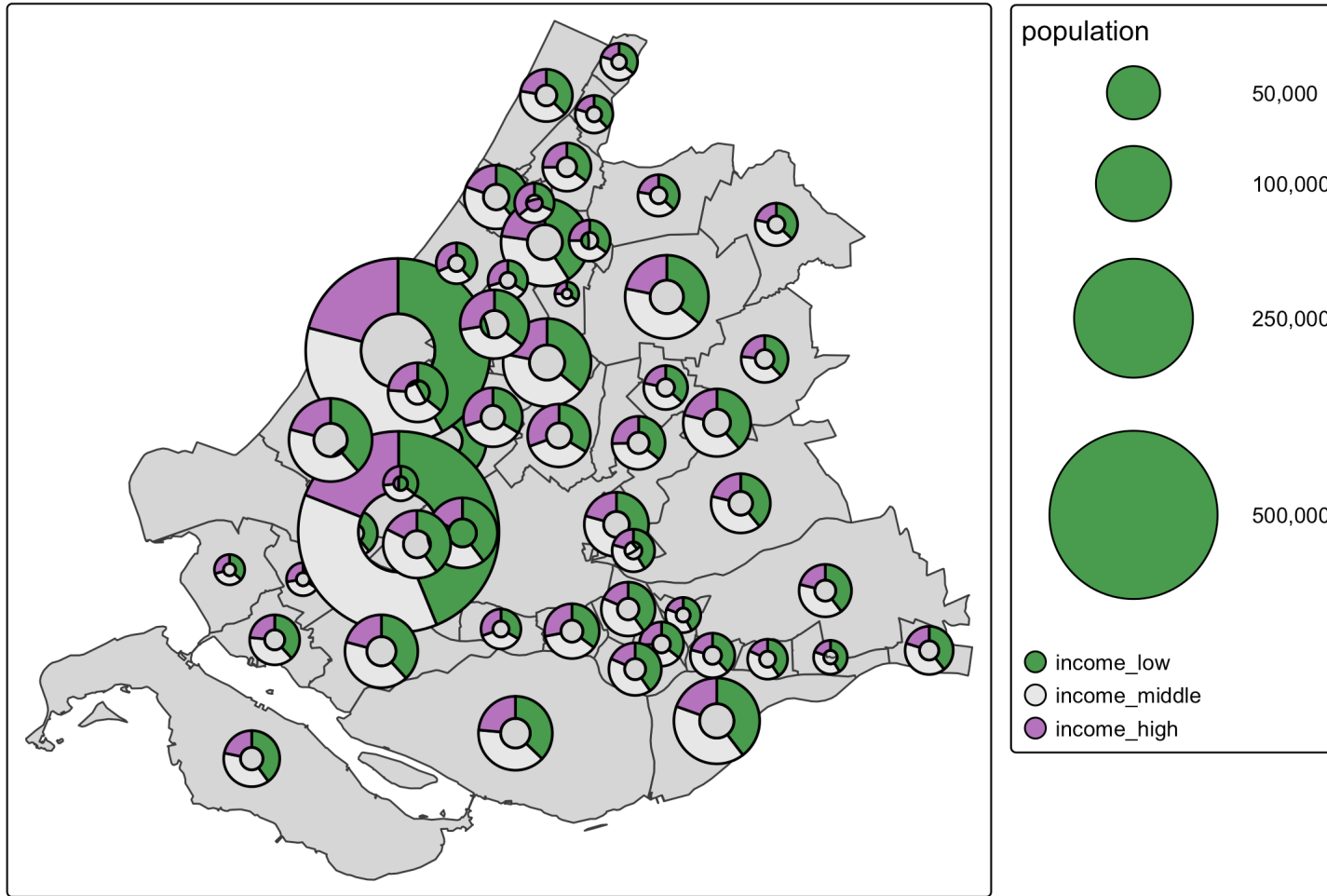
ZH_muni = NLD_muni[NLD_muni$province == "Zuid-Holland", ]

ZH_muni$income_middle = 100 - ZH_muni$income_high - ZH_muni$income_low

tm_shape(ZH_muni) +
  tm_polygons() +
  tm_donuts(parts = tm_vars(c("income_low", "income_middle", "income_high"), multivariate = TRUE),
    fill.scale = tm_scale_categorical(values = "-pu_gn_div"),
    size = "population",
    lwd = 1,
    size.scale = tm_scale_continuous(ticks = c(50000, 100000, 250000, 500000)),
    options = opt_tm_donuts(fill_hole = FALSE))
```

Useful for multivariate comparisons across space.

Glyph Maps with `tmap.glyphs`



Network Maps with `tmap.networks`

Visualize network data:

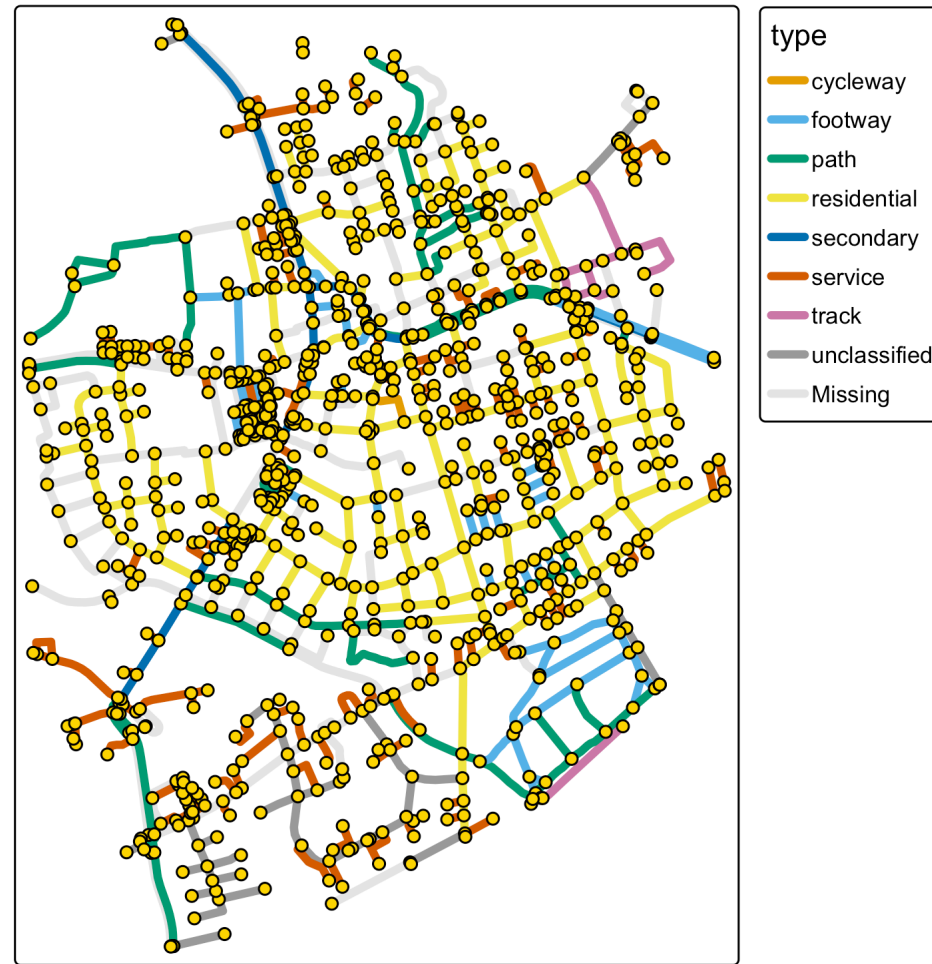
- Works with `igraph` or `sfnetwork`
- Layer: `tm_network()`

```
library(tmap.networks)
library(sfnetworks)

roxel_nw = as_sfnetwork(roxel)

tm_shape(roxel_nw) +
  tm_edges(col = "type", lwd = 4) +
  tm_nodes()
```

Network Maps with `tmap.networks`



Summary: Why Use Extensions?

- Add new visualization types (e.g. glyphs, cartograms)
- Support more data formats (e.g. graphs)
- Experiment with rendering engines (e.g. 3D)

Try one or more of these packages to enhance your maps!

STOP