Reproducible maps for everyone

**Keywords:** Maps, spatial data, visualisation, software

# Introduction

One map may say more than a thousand words. At least, a good map is easier and faster to read than plain text that describes the content of the map. It is therefore not surprising that maps play an important role in official statistics. Statistical researchers use maps to analyse data, from raw data to estimates, and use them to share their findings. Maps are also a perfect medium to communicate statistical output to the general public, since the vast majority of people are familiar with maps.

We focus on maps that display statistics. These maps are also known as thematic maps, where the theme is referred to as the subject of the statistics. Note that official statistics are often produced on the level of administrative regions, e.g. municipalities or square grid cells. The general aim of national statistical institutes is to produce statistics as detailed as possible regarding both the time and space dimensions. For instance, a monthly unemployment index per municipality is preferred over a yearly unemployment number for the whole country. That is where maps come into play.

The choropleth, the bubble map, and the raster map are the most popular thematic map types in official statistics, since they are very suitable to display data on administrative region level. The choropleth is a map where polygons (i.e. administrative regions) are coloured according to a quantitative variable, e.g. population per squared kilometre, or the percentage of smokers. In a bubble map, bubbles are drawn where the area size represents a quantitative variable. Bubble maps are particularly popular to display absolute numbers per administrative regions, e.g. population counts. Raster maps are used to display gridded data, where each squared grid cell is coloured according to a quantitative variable.

Thematic map types that require methodological challenges are the dot map, cartogram and the flow map. In a dot map, one dot is placed per data unit, e.g. the location of a person or an event [1]. A cartogram is a choropleth where the polygons are distorted such that the area sizes correspond to a quantitative variable. The flow map displays flow data by directed arrows, which are often bundled to prevent occlusion.

There are many different approaches and tools for making maps. Before the computer era, cartographers used pencil and paper to make maps. For obvious reasons, we leave this approach outside of the scope this paper. As for making maps with computers, there are two main approaches: by using specialized desktop geographic information system (GIS) software and by using general purpose programming languages. In this paper, we advocate the latter. In particular, we illustrate how easy it has become to make maps with R, in particular using the packages, dplyr [2], sf [3], and tmap [4].

# Creating maps with R

Maps can be created with desktop geographic information system (GIS) software, such as ArcGIS [5], QGIS [6], and GRASS [7]. These tools are very powerful, perhaps a little too powerful for statistical researchers. The abundance of methods and tools available and shown in the menus and toolbars may back off people who just want to make a map for a publication of regional statistics. Moreover, reproducing a map is less straightforward, especially for novice users. Maps are often made with a series of mouse clicks. This is not a problem per se, but this approach becomes tedious when maps have to be remade, which often occurs in daily practice, e.g. when the estimates have changed or for statistics that are published on a periodic basis.

In recent years, R has become accessible to a larger audience. Although R is programming language, you don’t need a master in computer science to use R for daily data analysis and visualisation. Computer scientists often regard R as a mediocre programming language, since it is slow, inefficient, and its syntax is often inconsistent. To cope with these shortcomings, many packages have been written that extend and improve the language. The result is an immense ecosystem of packages, from which several are so good that they are widely considered as new standards.

The so-called tidyverse bundle of R packages is a good example of a popular set of tools that are sufficient for general data manipulation and visualisation. One of these packages is called dplyr, which handles all basic operations on tabular data, such as filtering, merging and aggregating.

A decade ago, spatial data analysis and visualisation could be done with R, but it was not as easy as with GIS software, and moreover, there were many limitations regarding processing and visualising spatial data. A couple of years ago, there were two breakthroughs that made R a mature and solid tool for spatial data analysis and visualisation. First of all, the sf package has made spatial data processing much easier. Before that, you needed to know a couple of packages, each with its own syntax and peculiarities. The sf package is an all-in-one package where the dplyr methods can be used effortlessly. The second breakthrough is the use of interactive data visualisations in R [8]. These interactive visualisations, which all use state-of-the-art Javascript libraries under the hood, can also be used and embedded in dashboard applications with the R package shiny [9].

These recent developments have led to a couple of R packages that can be used to create maps in an intuitive and flexible way. In this paper, we will illustrate one of them: tmap. This package offers a flexible, layer-based, and easy to use approach to create maps. Maps can be exported into static images for journal and book publications. It is also possible to create interactive maps, and embed them in dashboard tools and websites.

# Example

The following example shows the code required to create a choropleth. Note that there are two odd operators that are typical for R. First, the left arrow is the R symbol for assignments (other programming languages use the = symbol for that). Second, the so-called pipeline operator %>% is used to chain actions.

# step 1: load required packages

library(dplyr)

library(sf)

library(tmap)

# step 2: read data

shp <- read\_sf("municipalities\_2018.shp")

dat <- read.csv("health\_survey\_2018.csv")

# step 3: process data

shp <- shp %>%

left\_join(dat, by = "MUNI\_CODE") %>%

mutate(perc\_smokers = smokers \* 100)

# step 4: create a map

tm\_basemap("Esri.WorldTopoMap") +

tm\_shape(shp) +

tm\_polygons("perc", palette = "Reds", title = "Smokers (in %)")

The first code chunks loads the required libraries. The data is read in the second code chunk. Typically, when creating a map, two datasets are required: the spatial data which contains the coordinates of the spatial points, lines or polygons, and the statistical output, which is data on the level of spatial units (e.g. municipalities). In this code chunk they are named shp and dat.

The third code chunk processes these data. It merges them by the common key MUNI\_CODE and calculates the percentage of smokers simply by multiplying the fraction of smokers times 100. The obtained shp object is a tabular data object where the rows correspond to spatial units, and the columns with data variables, with one special column in which spatial coordinates are contained.

The fourth code chunk is where the magic happens for the visualisation part. In the first line, a basemap is configured which serves as a background layer in the interactive map. In the code lines two and three, the data layer is added. The layers are stacked with the + operator. Line two specifies the spatial object and line three specifies which type of spatial geometry is plotted. Instead of tm\_polygons, one can also choose tm\_bubbles in order to create a bubble map.

# Conclusions

This paper describes the advantages of R over desktop GIS software and illustrates to power of R with a short example. Not only the code is short, but it is reproducible, which makes it an excellent choice for production processes.

Recently, R has become a powerful tool for spatial data analysis and visualisation. The set of spatial functions for processing spatial data is more than sufficient for most statistical researchers, and for those who still need specific algorithms from GIS software such as GRASS, there are R interfaces available [10].

Also for spatial data visualisation, R is very useful for statistical researchers and data scientists [4]. Maps can be made really quickly in R, with just a few code lines, which is ideal for data exploration. When more polished maps are required, e.g. for publication, R offers many functionalities, for instance custom style themes.

Regarding interactive maps, R is especially powerful since just a few lines of code are required. However, when more sophisticated maps are required, e.g. with dynamic legends and interactive storytelling blocks, other tools or languages may be more suitable. The strength of R regarding interactive data visualisations is exactly what statistical researchers and data analysts typically need: the ability to explore spatial data, and to share and publish the results.

# References

[1] M. Tennekes, Aggregation of dots. Paper for the Seminar Methods for Big Data in Official Statistics, 4-5 October, Heerlen, The Netherlands (2018).

[2] H. Wickham, R. François, L. Henry and K. Müller (2018). dplyr: A Grammar of Data Manipulation. R package version 0.7.5. <https://CRAN.R-project.org/package=dplyr>

[3] E. Pebesma (2018). sf: Simple Features for R. R package version 0.6-3. <https://CRAN.R-project.org/package=sf>

[4] M. Tennekes, tmap: Thematic Maps in R, Journal of Statistical Software 84(6) (2018), 1-39.

[5] ESRI (2016). ArcGIS Desktop: Release 10.4. Environmental Systems Research Institute, Redlands.

[6] QGIS Development Team (2016). QGIS Geographic Information System. Open Source Geospatial Foundation. URL <http://qgis.osgeo.org>

[7] GRASS Development Team (2016). Geographic Resources Analysis Support System (GRASS GIS) Software, Version 7.0. Open Source Geospatial Foundation. URL <http://grass.osgeo.org/>

[8] R. Vaidyanathan, Y. Xie, J.J. Allaire, J. Cheng and K. Russell (2018). htmlwidgets: HTML Widgets for R. R package version 1.2. <https://CRAN.R-project.org/package=htmlwidgets>

[9] W. Chang, J. Cheng, J.J. Allaire, Y. Xie and J. McPherson (2018). shiny: Web Application Framework for R. R package version 1.1.0. <https://CRAN.R-project.org/package=shiny>

[10] R. Lovelace, J. Nowosad, J. Muenchow (2018) Geocomputation with R, Forthcoming in CRS Press. URL <https://geocompr.robinlovelace.net/>