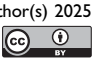
Special Issue: Data/Code Products from the GISRUK Community

Urban Analytics and City Science

EPB: Urban Analytics and City Science
2025, Vol. 0(0) 1–7



DOI: 10.1177/23998083251374737
journals.sagepub.com/home/epb


gridmappr: An R package for creating small multiple gridmap layouts

Roger Beecham[1], Martijn Tennekes[2] and Jo Wood[3]

Abstract

We present *gridmappr*, an R package that automates the process of generating gridmaps – small multiple data graphics of regular size, laid out with an approximate geographic arrangement. Given a set of real geographic point locations, *gridmappr* allocates points to a regularly sized grid of stated row–column dimensions. This allocation is constrained such that the distance between points in real and grid space is minimised and with a parameter that affects how compactly points are allocated to the regular grid. For geographies with features such as large bodies of water, fixed spacers can be introduced – reserved cells that cannot be allocated points. Layout examples are presented using different parameterisations, and code for generating a family of information-rich glyphmap and origin-destination maps is demonstrated using standard ggplot2.

Keywords

geovisualisation, gridmaps, cartograms, origin-destination, grammar of graphics

Introduction

Gridmaps, sometimes called tilemaps, are maps where spatial units form grid cells of regular size, allocated into an approximate spatial arrangement. The advantage is that complex multivariate structure, rather than single values, can be depicted (Beecham et al., 2020; Beecham and Slingsby 2019) and tested (Beecham et al., 2021; Wood et al., 2011) within geographic context. Many gridmaps are generated manually, the LondonSquared layout of London boroughs (After the Flood 2019) or those made available via the *geofacet* package (Hafen 2024). For automatic allocation of spatial units to grid cells, various constraints might be considered, such as *compactness* and *alignment*, and the preservation of *distance*, *topology* and *shape* (Meulemans et al., 2017).

[1]School of Geography, University of Leeds, UK
[2]Statistics Netherlands, The Hague, Netherlands
[3]City University of London, UK

Corresponding author:
Roger Beecham, School of Geography, University of Leeds, Leeds LS29JT, UK.
Email: r.j.beecham@leeds.ac.uk

Data Availability Statement included at the end of the article

gridmappr is an R implementation of Jo Wood's Observable notebook on *Grid map allocation* (Wood 2021) that uses linear programming, via the *ompr* R package, for handling constraints and deriving solutions. The package provides functions that allocate geographic points to a set of candidate gridmap cells such that the total of squared distances between geographic and grid locations is minimised. Each point is allocated to one grid cell only, and a cell in the grid can contain no more than one geographic point. A *compactness* parameter determines whether points should be positioned from the centre or edges of the grid, and fixed spacers – reserved cells that cannot be allocated points – can be introduced to preserve breaks between non-contiguous spatial units. *geogrid* (Bailey 2023) is an alternative R package supporting automatic grid allocation. This package uses gradient descent to approximate a grid directly from a spatial polygon file and the Hungarian algorithm for efficiently calculating assignments, with a single constraint of distance-distortion. *gridmappr* offers more control over the constraints in returned layouts than does *geogrid*, but with greater computational overhead. We intend in future versions of *gridmappr* to explore simulated annealing to efficiently optimise on a wider set of constraints, as do Meulemans et al. (2017).

In this short paper, we demonstrate how to parameterise *gridmappr*'s functions to generate different layout solutions and how information-rich glyphmap and origin-destination map data graphics can be created from those layouts using standard ggplot2.

Defining layouts with *gridmappr*

The main allocation function in *gridmappr* is points_to_grid(). For a given set of geographic point locations, this returns two-dimensional grid cell positions (row and column identifiers). The function is parameterised with:

- pts: A tibble of geographic points (*x,y*) to be allocated to a grid.
- nrow: Maximum number of rows in grid.
- ncol: Maximum number of columns in grid.
- compactness: Optional parameter between 0 and 1 where 0 allocates towards edges, 0.5 preserves scaled geographic location and 1 allocates towards centre of grid.
- spacers: Optional list of grid cell locations defining grid location of fixed spacers which cannot be allocated points. Coordinates are in (row, column) order with the origin (1,1) in the bottom-left. The default is an empty list.

In Figure 1, candidate gridmap layout solutions are shown for France's 96 départements. For this example, a single grid is used with row–column dimensions 13 × 12. Were a larger gird to be used, the layout would more closely approximate to the real geography of France (assuming a compactness of c.0.5). The trade-off with an increasing grid-size, however, is gaps and whitespace between spatial units and reduced data density – the graphic space on which data can be encoded. The three layouts in the bottom row of Figure 1 are created left-to-right with compactness scores of 0, 0.6 and 1. Spacers are introduced in the middle layout to ensure that départements in Corsica are non-contiguous with mainland France. We also annotate layout candidates with displacement vectors and provide an extract of the data frame returned by points_to_grid(), to help with understanding the data structure of gridmap allocations. Code and helper functions for plotting these displacement vectors are demonstrated in the package documentation (Beecham 2025).

Once a gridmap solution is arrived at, a polygon file representing the grid and corresponding cell positions (row and column IDs) is generated with the function make_grid(), which comes with *gridmappr*. This function, demonstrated in Figure 2, takes a simple features (*sf*) (Pebesma 2018) data frame containing polygons with 'real' geography and returns an *sf* object representing the grid,

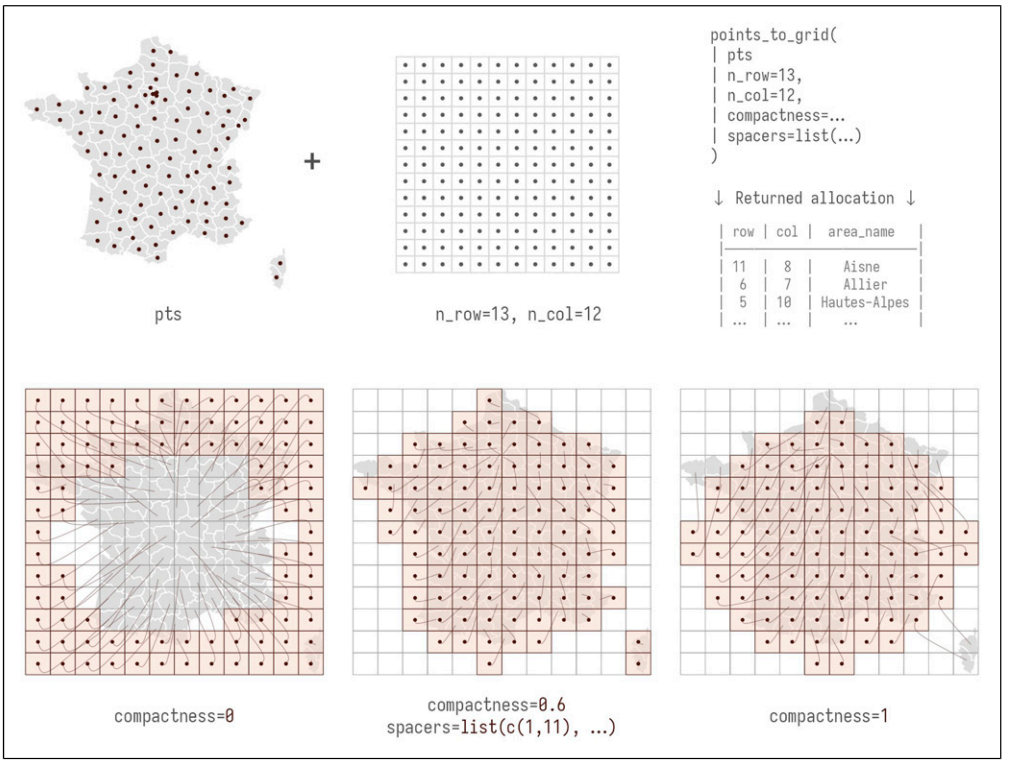


Figure 1. Candidate gridmap layouts and displacement vectors for France's 96 départements.

with variables identifying column and row IDs and geographic centroids of grid squares. The gridded object can be joined on a gridmap solution returned from points_to_grid() and cells plotted with standard ggplot2: geom_sf() for drawing cell outlines, and in Figure 2 proportional symbols placed at the centroids of grid cells with ggplot2's geom_point() function.

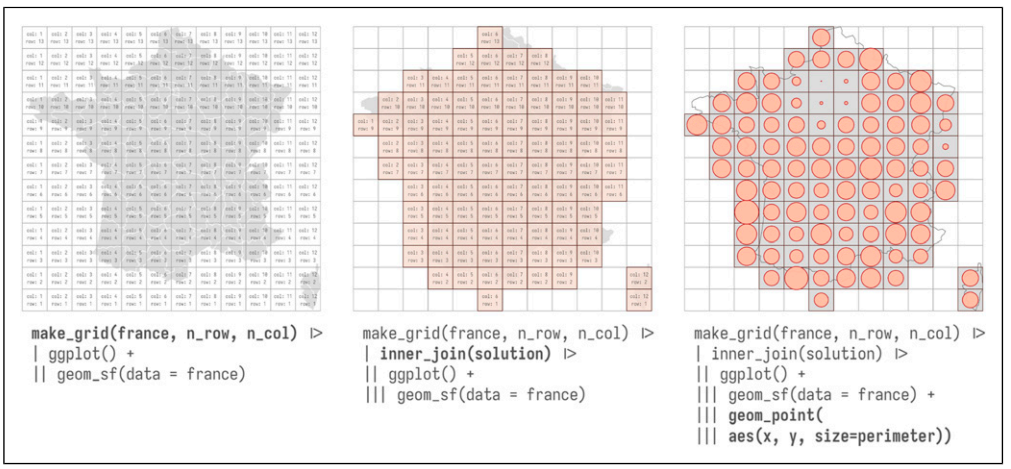


Figure 2. Generating spatial objects from gridmap layouts and plotting via ggplot2.

Building glyphmaps with *gridmappr* and ggplot2

The purpose of gridmaps is to enable visual analysis of multivariate structure within geographic context. The sorts of data-rich graphics envisioned, sometimes called glyphmaps, are made possible by the use of distorted spatial layouts, where geographic precision is relaxed in favour of regularly sized grid cells. When generating glyphmap graphics, it is therefore useful to have full access to ggplot2's charting idioms, not just the sorts of point or line primitives that appear in Figure 2.

In Figure 3, ggplot2's column chart idiom (geom_col()) is used to compare inverse distances between each département in France and its nearest 20 neighbours. We can think of this as an origin-destination dataset describing the inverse distance into each *destination* département from its neighbours (*origins*). Note also that these are rank-size charts in that bars are ordered on the *x*-axis according to proximity: the longer the bar, the shorter the distance between a département and its neighbour. In the left column, we highlight Paris and Corse-du-Sud. The shape of these distributions can be intuitively understood: Corse-du-Sud has few nearby départements other than an immediate neighbour on the island of Corsica, while Paris has several nearby départements, as expected given its population density. These 'rank-size' charts are plotted side-by-side for comparison by introducing a conditioning (facet) variable on the destination département (d_dep). To design glyphmaps, it is helpful to think of the spatial units that form gridmaps in the same way – as categorical conditioning variables. In the main portion of the figure, we show a full gridmap of bar charts laid out with an approximate geographic arrangement by supplying the *gridmappr*-generated column and row IDs of destination départements to facet_grid(). The only update to the bar chart code is that we supply row and column identifiers to facet_grid(), with a slight hack on the row variable (*-row*) as

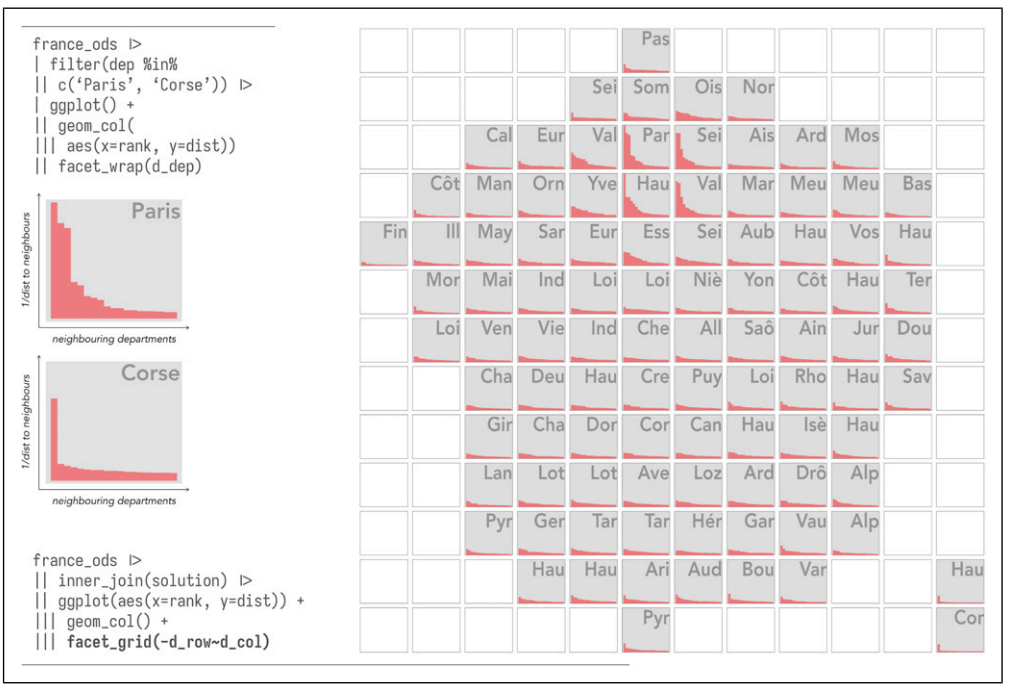


Figure 3. Standard ggplot2 column chart with gridmap layout applied via faceting on *gridmappr* row and column IDs.

gridmappr's origin (*min-row, min-col*) is the bottom-left cell in the grid, whereas for facet_grid() the origin is the top-left.

Origin-destination maps with *gridmappr* and ggplot2

We can extend the idea of faceting *gridmappr* layouts to generate full origin-destination maps (Wood et al., 2010). Although they initially require some cognitive effort, OD maps overcome many of the difficulties from which standard origin-destination flow analysis techniques suffer. They use a map-within-map layout to encode origin-destination data within geographic context. In the example that appears in Figure 4, the distances between départements are represented not using bar charts, but as choropleth maps.

Previously, we referred to the two highlighted départements – Paris and Corse-du-Sud – as destinations. In an OD map, these destinations are the large, reference cells of the map-within-map layout and the smaller cells are origins, in this case coloured according to the straight-line distance between those origins and the reference cell. Again, the ggplot2 code is only minimally updated, replacing geom_col() with geom_sf(), for drawing the origin cell boundaries and now filled according to distance (fill=dist) – darker colours for longer distances. An important subtlety is that the origin-destination dataset is joined on a geometry file describing the outlines of the gridmap using row and column IDs of the 'origin' départements: the smaller cells of the map-within-map layout. The plot is therefore a faceted set of gridmap polygons. Given this approach, it would be straightforward to join on the original, true geometry of French départements so that the smaller origin maps are of a more familiar shape, aiding interpretation.

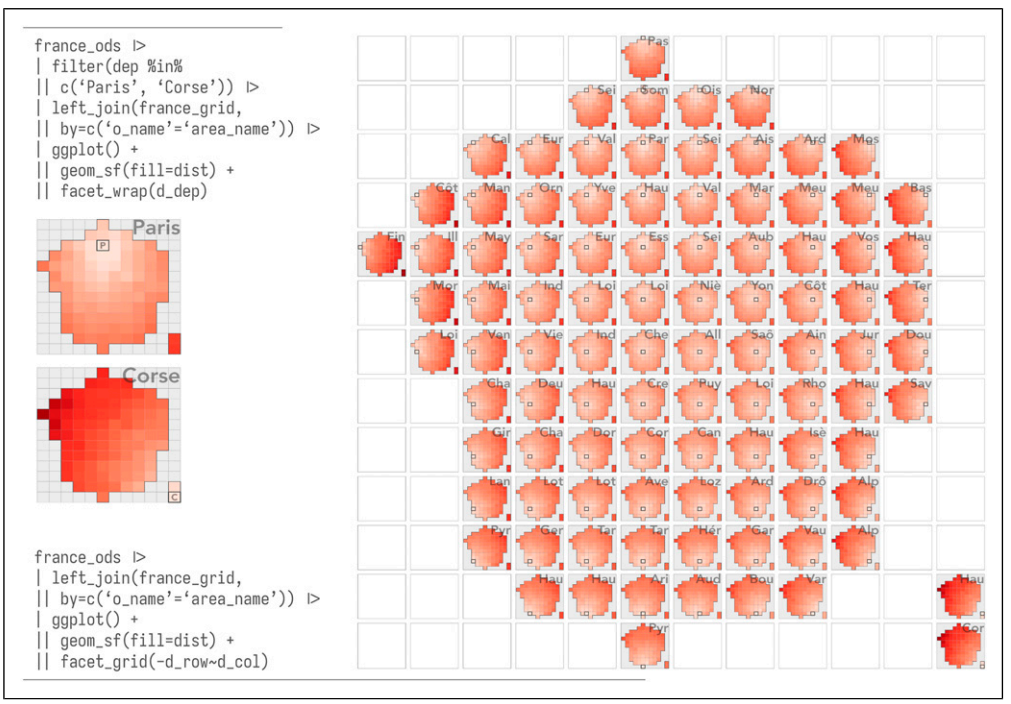


Figure 4. Full origin-destination maps in ggplot2 – data frames with spatial geometries are faceted into *gridmappr* row and column layouts.

Conclusion

gridmappr automates the process of generating gridmaps – maps consisting of grid cells of regular size allocated with an approximate spatial arrangement. We demonstrate how different gridmap layouts can be generated using *gridmappr*'s helper functions and how a family of gridmap data graphics – glyphmaps and OD maps – can be drawn with standard ggplot2 calls. The package currently implements two constraints: *distance* displacement and *compactness*, with the facility to define *spacers* in order to preserve some recognisable geographic context: for example, to separate spatial units that are non-contiguous. These layout constraints are handled via Linear Programming and using the *ompr* R package. *gridmappr* is inspired by Information Visualization Literature and especially Meulemans et al.'s (2017) *small multiples with gaps* work. Future releases of the package will provide options for further constraining layouts using *topology* and *shape* (Meulemans et al., 2017) as well as Van Beusekom et al. (2023)'s extension for novel hybrid *data-spatial* layouts. By demonstrating how data-rich gridmap graphics can be generated within an accessible and widely used declarative visualisation toolkit (ggplot2), we hope to assist others in incorporating novel gridmap designs into their everyday data analysis practice.

ORCID iD

Roger Beecham https://orcid.org/0000-0001-8563-7251

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by grants from EPSRC (EP/Z531273/1).

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Data Availability Statement

gridmappr and the underlying data for the graphics presented in this article is available open source under the license GPL3.0 via github: https://github.com/rogerbeecham/gridmappr. Version v0.2 is described in this article, archived at doi: 10.5281/zenodo.16878384. The boundary data of French départements were originally collected from: Lexman, A. (2018) Carte des départements, *data.gouv.fr*, accessed at: https://www.data.gouv.fr/datasets/carte-des-departements-2-1/, © OpenStreetMap contributors under ODbL license (Beecham, 2025; Lexman, 2018; OpenStreetMap contributors, 2024).

References

After the Flood (2019) LondonSquared. URL. https://github.com/aftertheflood/londonsquared

Bailey J (2023) geogrid. URL. https://github.com/jbaileyh/geogrid.Rpackageversion0.1.2

Beecham R and Slingsby A (2019) Characterising labour market self-containment in London with geographically arranged small multiples. *Environment and Planning A: Economy and Space* 51(6): 1217–1224.

Beecham R, Williams N and Comber L (2020) Regionally-structured explanations behind area-level populism: an update to recent ecological analyses. *PLoS One* 15(3): e0229974.

Beecham R (2025). gridmappr, URL. https://www.roger-beecham.com/gridmappr/

Beecham R, Dykes J, Hama L, et al. (2021) On the use of 'glyphmaps' for analysing Covid-19 reported cases. *ISPRS International Journal of Geo-Information* 10(4): 213.

Hafen R (2024) Geofacet. URL. https://github.com/hafen/geofacet.Rpackageversion0.2.1. https://hafen.github.io/geofacet/

Lexman A (2018) Carte des départements, *data.gouv.fr,* accessed at: https://www.data.gouv.fr/datasets/carte-des-departements-2-1/, © OpenStreetMap contributors under ODbL license.

Meulemans W, Dykes J, Slingsby A, et al. (2017) Small multiples with gaps. *IEEE Transactions on Visualization and Computer Graphics* 23(1): 381–390.

OpenStreetMap contributors (2024) OpenStreetMap. Available at: https://www.openstreetmap.org © OpenStreetMap contributors under Open Database License.

Pebesma E (2018) Simple features for R: standardized support for spatial vector data. *The R Journal* 10(1): 439–446.

van Beusekom N, Meulemans W, Speckmann B, et al. (2023) Data-spatial layouts for grid maps. In: *The 12 International Conference on Geographic Information Science*. Schloss Dagstuhl – LeibnizZentrum, Vol. 277.

Wood J (2021) Grid map allocation. URL. https://observablehq.com/@jwolondon/gridmap-allocation

Wood J, Dykes J and Slingsby A (2010) Visualisation of origins, destinations and flows with OD maps. *The Cartographic Journal* 47(2): 117–129.

Wood J, Badawood D, Dykes J, et al. (2011) BallotMaps: detecting name bias in alphabetically ordered ballot papers. *IEEE Transactions on Visualization and Computer Graphics* 17(12): 2384–2391.

Author biographies

Roger Beecham is Associate Professor of Visual Data Science at School of Geography and Leeds Institute for Data Analytics, University of Leeds.

Martijn Tennekes is a Data Scientist at Statistics Netherlands.

Jo Wood is Professor of Visual Analytics at giCentre, City St George's, University of London.