

# Using MapWeaver to Make Tiled and Woven Maps of Multivariate Thematic Data

David O'Sullivan

University of Auckland / Geospatial Stuff  
david.osullivan@auckland.ac.nz

Luke Bergmann

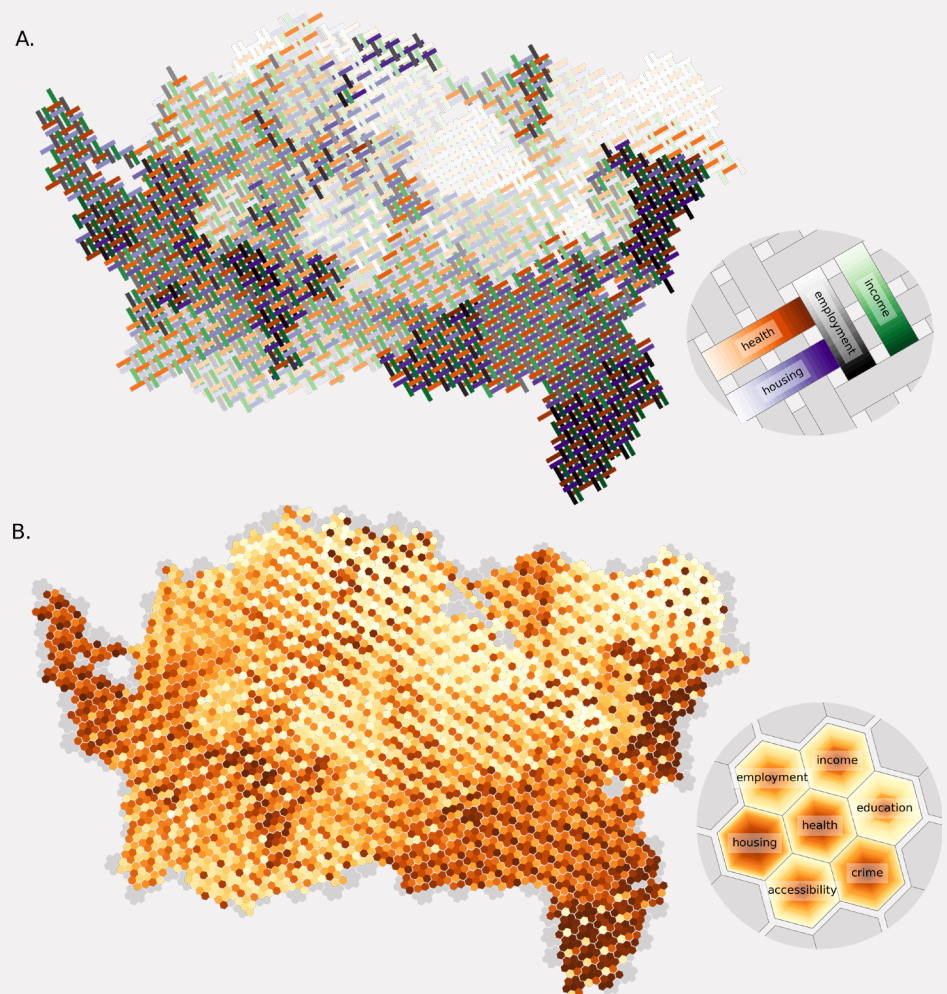
University of British Columbia  
luke.bergmann@gmail.com

## INTRODUCTION

MAPPING HIGH-DIMENSIONAL area unit data, so that multivariate patterns across geographical space can be explored and understood, is never easy. In this article, we explain how to use the *MapWeaver* web app to make tiled and woven maps of such multivariate data, as one approach to the problem.

Examples of such maps are shown in Figures 1 and 2. Figure 1A shows four components of an index of deprivation (Exeter et al. 2017) using a “basketweave” pattern. Figure 1B shows the same data set, extended to include a further three indices, and employing a “7-colouring” of hexagonal tiles. In this map, all seven indices are shown on an identical colour ramp, drawing attention to areas where one or more of the indices is out of step with others. Figure 2 shows the technique applied to categorical data and displays anthropogenic biomes (Ellis et al. 2021) at six moments in time from 1760 to 2010 CE, for a section of the northeast of the North American continent.

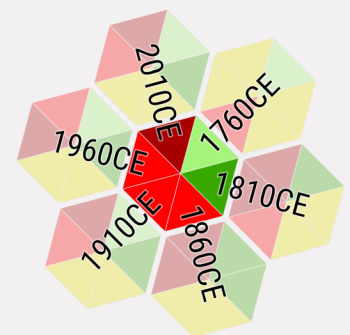
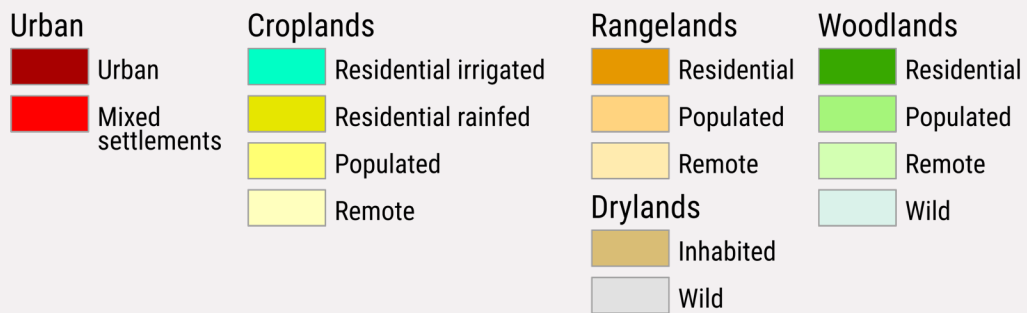
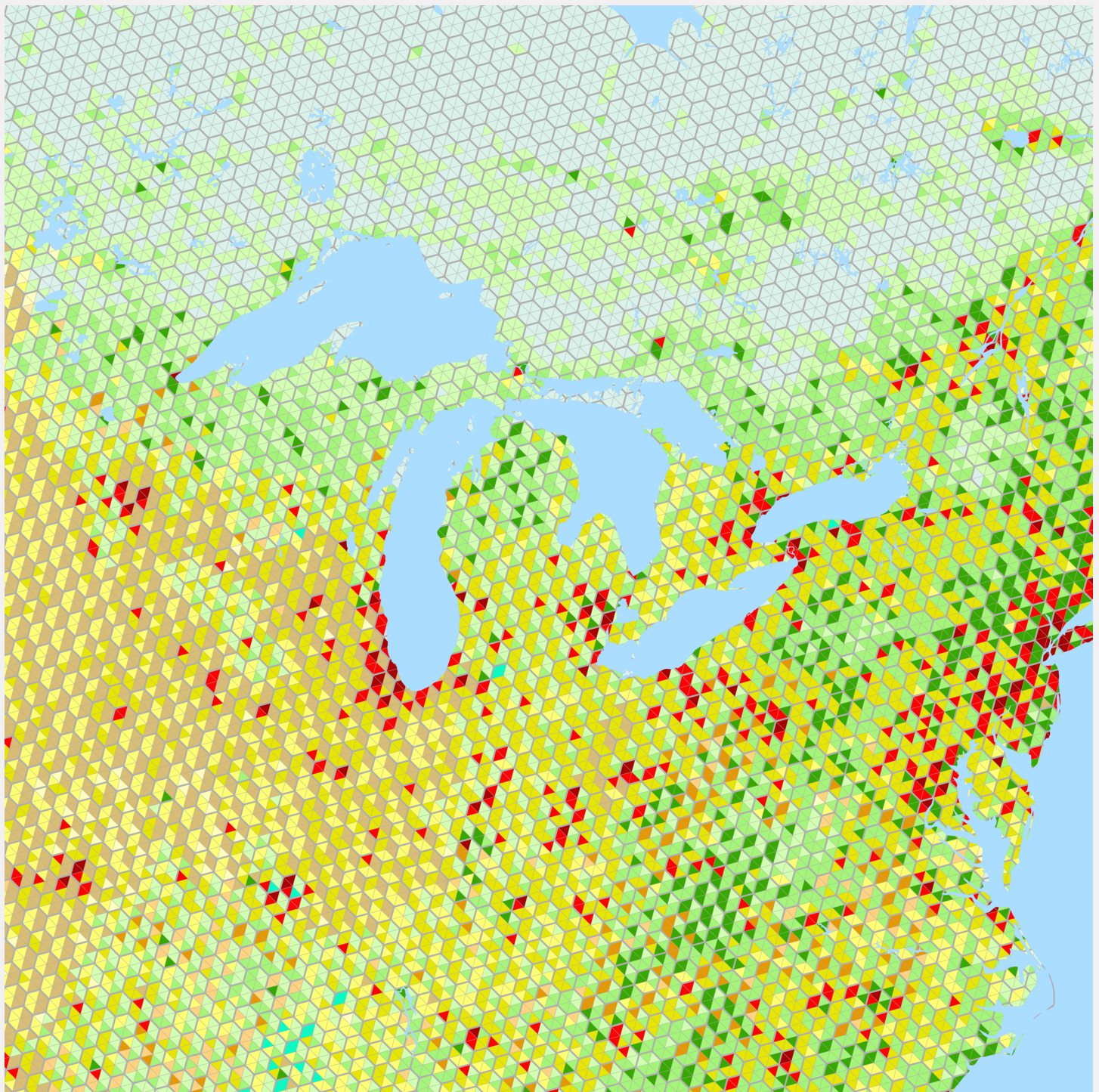
With small to medium numbers of variables—up to about twelve for some patterns—this approach offers a way to visualize complex geographical data that cannot otherwise be easily



**Figure 1.** (A) Four deprivation indices in a basket weave pattern. (B) Seven indices of deprivation in a 7-colouring of hexagons. Both maps cover the centre city area of Auckland, New Zealand. Legends show which elements in the tiling correspond to various aspects of deprivation and the associated colour ramp, from lighter colours for less deprived to darker colours for more deprived.



© by the author(s). This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/4.0>.



**Figure 2.** Anthropogenic biomes mapped at six different moments in time in the northeast of the North American continent. This data and the associated colour categories are from Ellis et al. (2021).

communicated in a single map. This method can accommodate even larger numbers of variables (as many as twenty at the time of writing, and potentially more if you use the associated Python `weavingspace` module<sup>1</sup>). As that number grows, the tiled or woven patterns become more

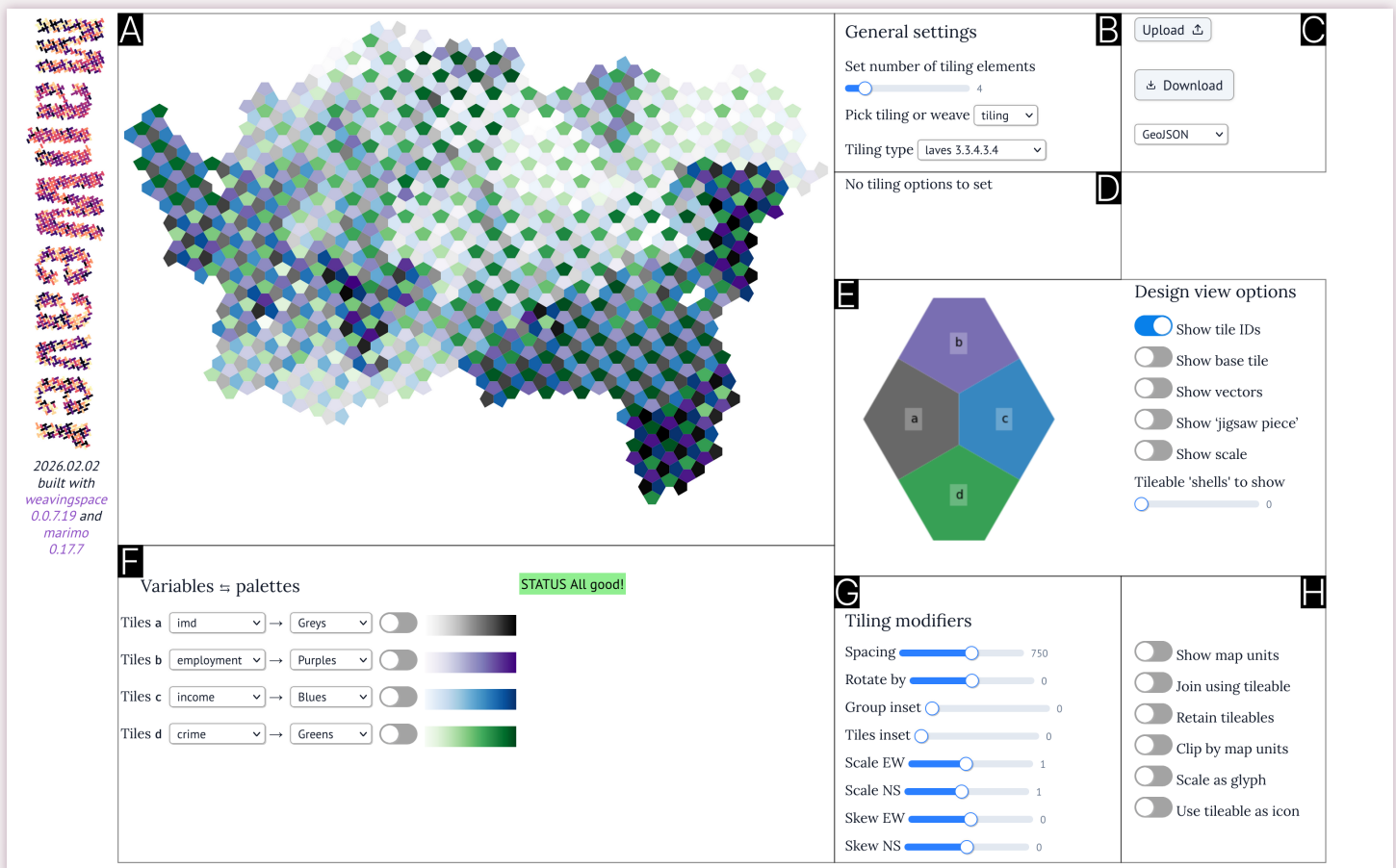
like textures, giving a more impressionistic view of complex geographical patterns. This approach brings thematic mapping into closer conversation with more holistic geographical approaches to the landscape.

## OVERVIEW OF THE APPROACH AND OF THE APP

YOU CAN FIND THE current version of MapWeaver at [geospatialstuff.com/mapweaver/app](https://geospatialstuff.com/mapweaver/app). The app takes a little while to load, during which time a brief explanation is displayed in a splash screen. Links in the splash screen will take you to pages with additional information about the underlying mapping approach. The app interface is shown in Figure 3, with different areas annotated and described in more detail below. While annotated in a left-to-right,

top-to-bottom order on the figure, they are described below in the order in which they are likely to be used in practice.

- **Map view [A]** provides a preview of what a map made with currently chosen settings will look like. If the map is downloaded as a graphical file this is *exactly* what you will get. If you instead opt to download a



**Figure 3.** The MapWeaver web app, with each area of the interface marked: (A) map view, (B) tileable unit setup, (C) upload/download options, (D) tile or weave type specific options, (E) tileable unit design view and options, (F) variable and palettes selection, (G) modifiers controlling the tileable unit, and (H) further options for how tiling overlay should be applied.

1. See [github.com/DOSull/weavingspace](https://github.com/DOSull/weavingspace).

geospatial data file this shows more or less what you will get, subject to differences in the details of how you assign colour ramps to variables or other changes you might make in other software.

- **Upload/download options [C]** allow users to upload their own data set (in GeoJSON format) and download results as an image, or as a geospatial data file (geopackage or GeoJSON).
- The tileable unit **Design view [E]** and options panel displays the current tileable unit (explained below) with any options or modifiers as specified by settings chosen in **B**, **D**, and **G**, and also allows selection of a number of options for how the tileable unit is displayed.
- **General settings [B]** allow you to choose how many variables from the input data you wish to map, and, contingent on that choice, offers a range of tiling or woven pattern choices.
- **Tiling or weave options [D]** are offered depending on the chosen tile or weave type in **B**. Additional options relevant to that choice will be available here (no such options are available for the tiling shown in Figure 3).
- **Tiling modifiers [G]** provides for a number of changes to the tileable unit, such as rotation, scaling or skewing in the east-west and north-south directions,<sup>2</sup> and inseting of the tileable unit and/or its constituent tiles.
- **Variables and palettes [F]** enables configuration of which elements in the tileable unit are to represent which variables in the data, and which colour ramps are to be applied to them.
- **Options for tiling overlay [H]** control how the GIS overlay operation to create the tiled map should be applied.

## THE MAP DESIGN PROCESS

---

LIKE ANY OTHER DESIGN PROCESS, CREATING A MAP using MapWeaver is iterative, but broadly the sequence of decisions is to: (1) prepare and upload your data; (2) choose a tiled or woven pattern to work with; (3) adjust the specific and general options associated with that pattern; and, finally (4) make decisions about the colours to be used. The specifics of each of these are set out below. If you want to simply learn the process and get a feel for the possibilities, MapWeaver launches with a default dataset (based on Exeter et al. 2017) containing 30 numeric variables mapped for around 150 areas.

### 1. PREPARE AND UPLOAD YOUR DATA

Once you have a sense of what the app can do, you will want to upload your own data. The app only accepts data in GeoJSON format. MapWeaver does not enforce any restrictions on chosen map projections, so you'll get away with any valid map projection—it's your design after all!

Officially, GeoJSON only supports WGS84 (EPSG 4326) latitude-longitude projected data, but in practice GIS platforms allow projection metadata to be added to GeoJSON data. If your data has projection metadata that MapWeaver recognizes, MapWeaver will treat it as such and all dimensions in the app settings will be denominated in the units of that projection. Otherwise, your geometry data will be treated as if it were expressed in latitude and longitude (WGS84) and MapWeaver will internally reproject the data to Web Mercator (EPSG 3857) for the purpose of giving you mapping parameters with units in meters. Transformation to EPSG 3857 is not an ideal choice for final cartographic products, but is a convenient conformal projection with global coverage that is reasonably applicable to areas of limited regional extent. With these considerations in mind, it is recommended that you prepare your data in the final coordinate system you intend to use.

When your data are loaded, the map view (**A**) will update and show a default map applying the so-called Cairo

---

2. Strictly speaking, the scaling and skewing is horizontal and vertical with respect to the map projection.

tiling.<sup>3</sup> At this point, we recommend that you increase the **Spacing** setting in the tileable unit modifiers area (**G**) to its highest value, or a value towards the high end of the available range. This will ensure more rapid updates of the map view (**A**) and tileable unit displays (**E**) while you explore the options and tweak settings on the tileable unit before fine tuning the spacing for the final map.

## 2. CHOOSING A TILED OR WOVEN PATTERN

Central to this method of mapping is the concept of a *tileable unit*.<sup>4</sup> This is the repeating set of tiles which, when tessellated across the map, yield a tiled or woven overall pattern. The current state of the tileable unit based on selected options is shown in the tileable unit design view area (**E**) of the interface. Controls alongside this view allow you to show various aspects of the tileable unit. The **tile IDs** are labels associated with each tile and used to assign variables in your data to specific tiles in the final map pattern. The **base tile**, **vectors**, and **'jigsaw piece'** options may help you understand how the tileable units will fit together to produce a final tiled map. The vectors show the directions and distances which the tileable unit will be copied in to produce the map pattern. You may also optionally switch on a **scale** to see the size (in units of the projected coordinate system) of the tileable unit. Finally, in this part of the interface, you can request to see additional **'shells'** of the tiling—neighbouring iterations of the tileable unit—to get a better sense of the pattern in detail. Of course, at all times the final overall pattern will appear in the map view area (**A**).

MapWeaver offers a wide array of tiled and woven patterns.<sup>5</sup> The single most important determinant of which patterns are available to you is the number of variables you wish to represent in your map. Setting the **number of tiling elements** in the general settings area (**B**) will change the patterns available to you. Under these same settings, you also need to pick either a tiling or a weave pattern. Based on these choices, different options will be available in a **Tiling type** or **Weave type** drop-down list. These have been arranged in what we consider to be an

approximate order of likely usefulness, but we suggest you examine all the options before picking one or two to focus your attention on. It is not difficult to change this choice later, but it can easily become a distraction if you regularly jump around among the options.

### *How Many Variables?*

How many variables you choose to represent is entirely up to you. The options available in the MapWeaver app can accommodate any number between two and twenty, except for seventeen. If you simply *must* map seventeen variables, or even some number greater than twenty, then greater flexibility is available using the Python **weaving-space** module (see “Digging Deeper” below).<sup>6</sup>

### *Tiled or Woven?*

There is no reason to favour either one of the tiled or woven map patterns over the other. We, as developers of this approach, started out favouring woven patterns because that was the genesis of the method, but as we have added options for tiled patterns, we have come to appreciate the potential of both.

Tiled patterns offer some interesting opportunities for further manipulation that don't apply to woven patterns. For example, a tiling consisting only of convex polygons, particularly if the tileable unit itself is also a convex polygon (usually a hexagon or square), can yield what are in effect statistical glyphs by rescaling the elements. These can be placed one per map area as “icons” as an alternative mapping method by switching on the option **Use tileable as icon** in the further options area (**H**).

On the other hand, woven maps can be manipulated to create variations on a cross-hatched map, and they are particularly good at enabling map readers to see connections and relationships between areas widely scattered across the mapped area. Weave patterns are specified by a “strands code” included in the pattern's name in the drop-down list of patterns. For example, “basket ab|cd” specifies a basket weave pattern with two strands in each direction (either

3. Also known as Laves tiling (3<sup>2</sup>.4.3.4), this is the best tiling in the opinion of at least one of the authors. No correspondence will be entered into on this matter, nor will we consider what “best” means in this context.

4. Note that whereas we refer throughout this article to “tileable units,” on the app interface the term “tileable” is used for brevity.

5. This notebook [github.com/DOSull/weavingspace/blob/main/examples/all-the-tiles.ipynb](https://github.com/DOSull/weavingspace/blob/main/examples/all-the-tiles.ipynb) provides an overview of the available tiling options.

6. If you are exploring highly multivariate maps using this method, we'd love to hear from you and would be happy to advise on the possibilities and challenges. Such challenges often motivate new approaches.

side of the ‘|’ symbol) with element identifiers “a” through “d” corresponding to the choices made in the variable and palettes section (F). A “-” symbol in the strands code indicates a strand to be skipped in the pattern, which can be used to make it easier to identify which element is which in the pattern. This is particularly important for twill weave patterns, where it may otherwise be impossible to distinguish elements in the resulting overall pattern (see Figure 4).

### 3. TWEAK SETTINGS OF THE CHOSEN PATTERN

The effects of the controls in the **Tiling modifiers** section (G) of the interface are set out below.

#### Spacing

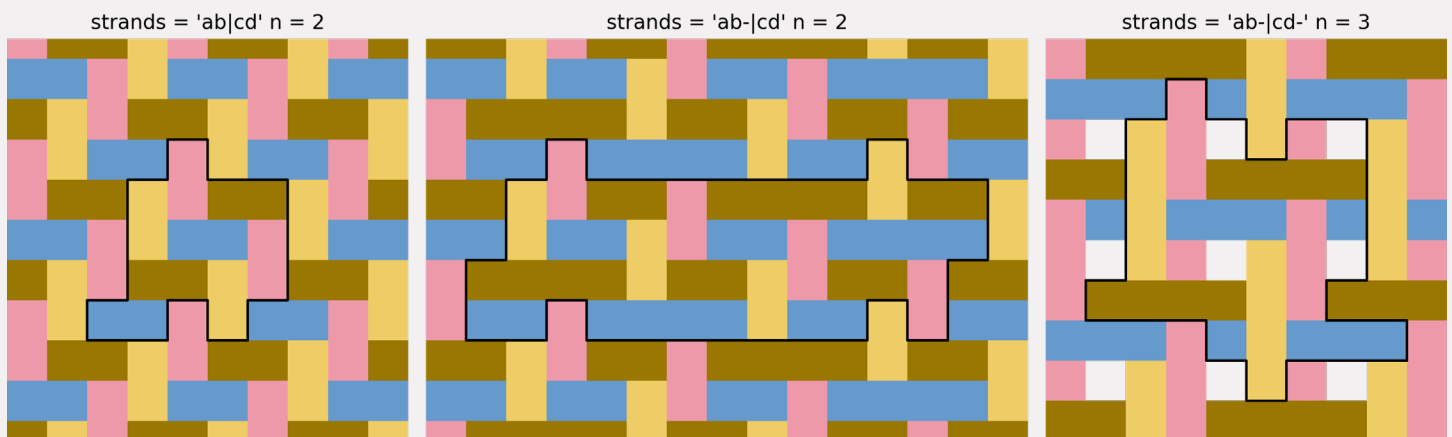
The spacing setting controls the size of the tileable unit expressed in units of your data’s map projection. The interpretation of this setting differs between tilings and weaves. In the former case, it is the distance across the repeating tileable unit. In weave patterns, it is the distance between strands in the pattern, so that the repeat distance of the tileable unit will depend on how many strands are in the pattern. We recommend keeping the spacing setting relatively high while you experiment with other details of your design. This will allow the app to update the map view more quickly in the early stages of the process.

One heuristic for choosing the spacing setting is to have it reflect the typical size of the enumeration unit polygons in the dataset. If polygons are typically 500 metres across, then a spacing setting in a tiling pattern might be chosen

around that value. This would be a compromise between information density and readability. In a weave pattern of (say) four strands (two in each direction) the relevant spacing setting would be around 250 metres. Alternatively, if you absolutely must represent all data attributes in all the mapped area polygons, then the dimensions of the smallest polygons will dictate the maximum spacing you would use. If the size of polygons varies widely, then a compromise setting may mean that some data attributes in the smallest polygons are not represented by any tile in the final map. This is a general difficulty of choropleth mapping when very small areas are mapped alongside much larger ones (e.g., San Francisco County in a choropleth map of California), and is not a weakness particular to tile and weave pattern mapping. Other cartographic considerations may also come into play in making such choices, such as the extent of the final map, its resolution, and its typical viewing distance. Options related to how tiling elements are rendered when they cross data polygon boundaries are considered below in “Further Options” (e.g., **Join using tileable**).

#### Rotation

The **Rotate by** setting changes the orientation of the tileable unit and of the resulting tiled pattern. Experiment with this setting to find an orientation of the pattern that works well in combination with the geographical orientation and alignments in the mapped area. We have found that it is often good to avoid horizontal or vertical pattern alignment when the base tileable unit is square. For example, biaxial weave patterns may be more visually appealing when rotated by anywhere from 15° to 75° from the



**Figure 4.** Three four-attribute twill weaves differentiated by strands code. The first of these yields elements in the two directions that are only distinguishable by their colouring, not by location in the pattern. The other two patterns, while based on a similar weave, allow distinct elements to be identified from their location in the pattern. The repeating tileable unit in each pattern is outlined in black.

starting orientation. For a hexagon based tileable unit, it may be important to avoid 30° and 60° rotations (but note that the starting orientation of some tileable units, such as some hexagon and square colourings, may not require adjustment in this way).

### Scaling and Skewing

Tileable units can be stretched or shrunk in the east-west and north-south directions using the **Scale EW** and **Scale NS** settings. Any stretching or shrinking is applied to the tileable unit *after* rotation (see Figure 5).

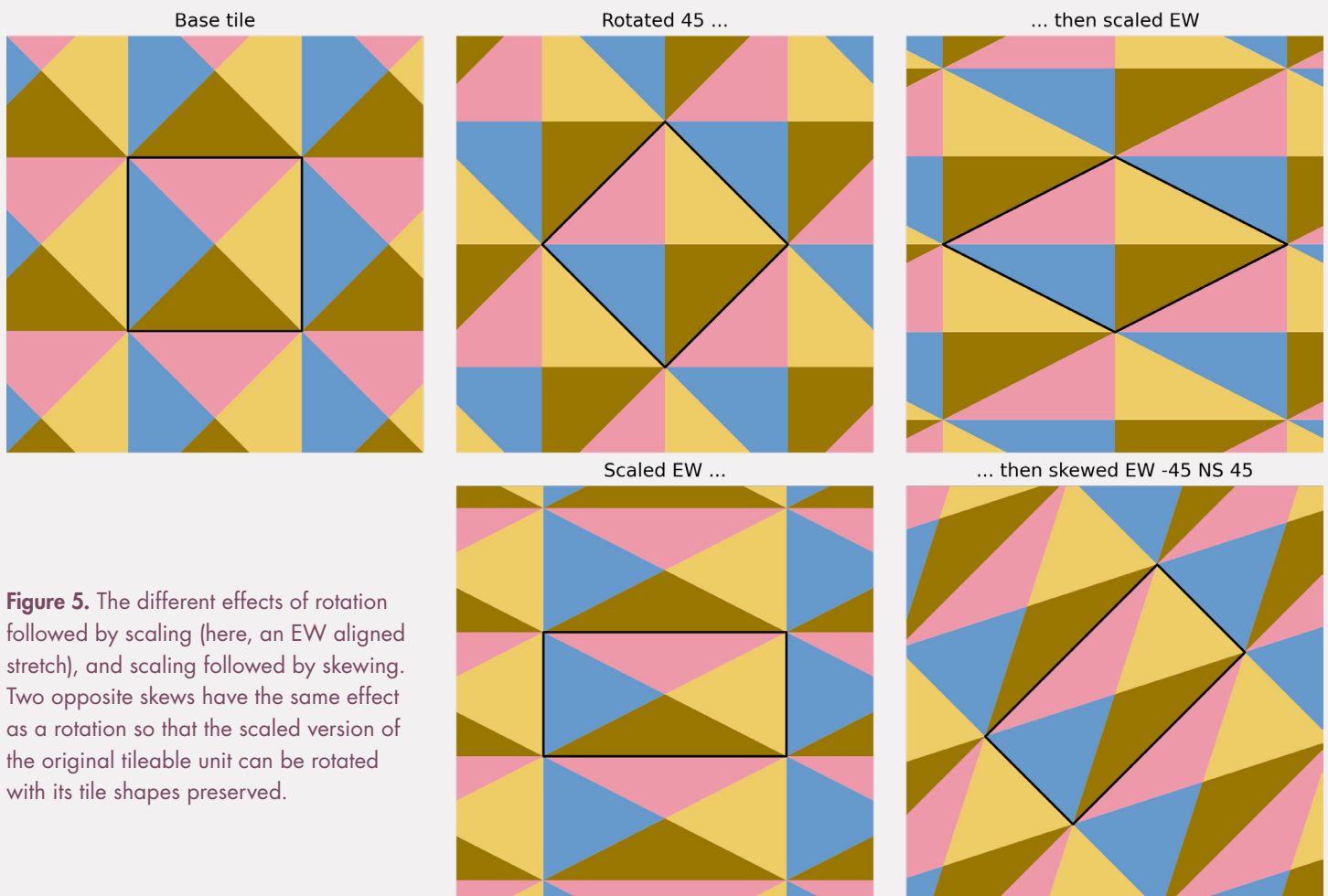
Tileable units may also have an affine skew transformation applied in the east-west and north-south directions using the **Skew EW** and **Skew NS** settings. Again, any skew is applied *after* rotation and scaling.

It is useful to know that a north-south skew combined with an equal *opposite* east-west skew produces a combined effect equivalent to rotation by the angle of the north-south skew. This means you can produce rotated patterns based

on scaled tileable units that preserve the shape you would get if that scaling were applied to the unrotated tiling. If you try to do this using rotation and scaling, you may get effects you do not expect, because the app applies the rotation first, then the scaling, and the resulting tileable unit is shown in the top row of Figure 5. If the desired outcome is the tileable unit in the bottom row of Figure 5, then use scaling followed by skewing to achieve the desired rotation. To apply arbitrary combinations of rotation, scaling, and skewing to tileable units in any order, you should use the Python `weavingspace` module.

### Insetting

You can *inset* tileable units in one of two ways, using either the **Group inset** and **Tiles inset** settings, or scaling combined with the **Scale as glyph** option (**H**). Insets are an important design feature of tiled maps both visually and practically, as they are one way to help readers to tell which element is which when a tiling is applied across a whole map.



**Figure 5.** The different effects of rotation followed by scaling (here, an EW aligned stretch), and scaling followed by skewing. Two opposite skews have the same effect as a rotation so that the scaled version of the original tileable unit can be rotated with its tile shapes preserved.

The most straightforward of these settings is the **Tiles inset**, which reduces the size of tiles in a tileable unit by the specified percentage of their spacing. The reduction in size is uniform regardless of the aspect ratio of a tile (it is implemented in software using a negative buffer operation). The effect of this setting is seen in Figure 6A. This can be especially useful when applied to weave pattern tileable units where it can enhance the woven appearance of the overall pattern.

While inseting individual tiles achieves a nice visual effect, it may not help with distinguishing which tile element is which in the overall pattern. Inseting the whole tileable unit using the **Group inset** setting can help with this, as seen in Figure 6B, where the position of each tile within the repeating tileable unit is clear. This setting can be combined with the **Tiles inset** setting as in Figure 6C. The **Group inset** setting is unavailable for weave patterns, as carrying out this operation on woven patterns would break strands and introduce discontinuities in the woven appearance.

A side effect of group inseting is that differences in the size and shape of elements may arise. The hexagonal tiles central to each tileable unit in Figures 6B and 6C are the

only tiles in these patterns that have retained their original shape, even if their size is slightly reduced in Figure 6C. Other tiles in each tileable unit are inset differently along edges external to the tileable unit than along interior edges, reducing their size relative to the central tile, and also introducing an asymmetry. In most cases these effects will be barely noticeable, but if a group inset without these side effects is desired, applying **Scale EW** and **Scale NS** settings less than 1, combined with turning on the **Scale as glyph** option, as in Figure 6D, is recommended. This approach works well on tileable units which are symmetrical about their centre and have a unit aspect ratio, but it can be tricky to select the right scale settings in many other cases, especially if rotation and skew are also in effect.

### PATTERN SPECIFIC OPTIONS

Many of the chosen tile and weave unit choices allow additional options explained below. These options appear at the bottom of the **General Settings** area when available.

#### Hexagon and Square Slice Offsets

These tileable units are generated by cutting a base hexagon or square into slices, starting from its centre and

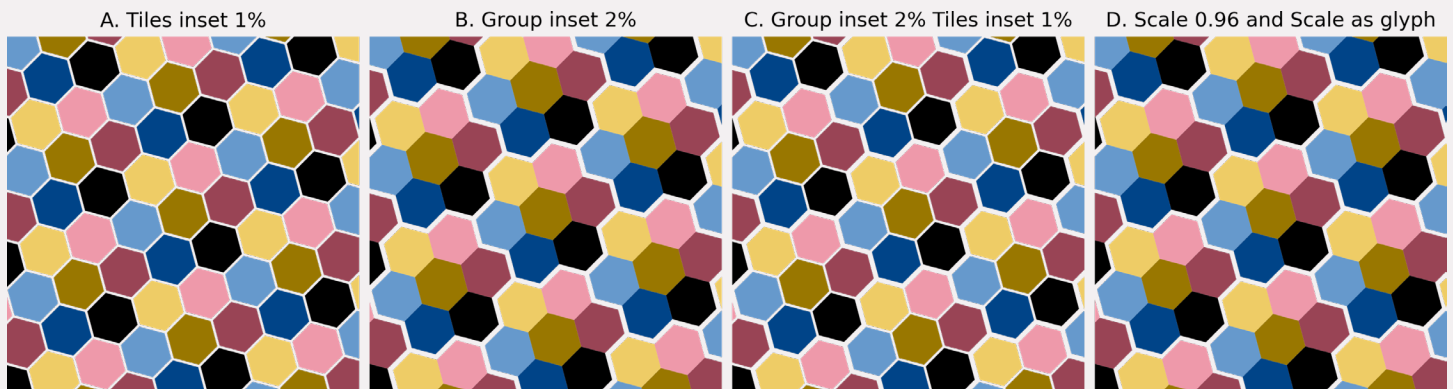


Figure 6. The effect of various inseting options.

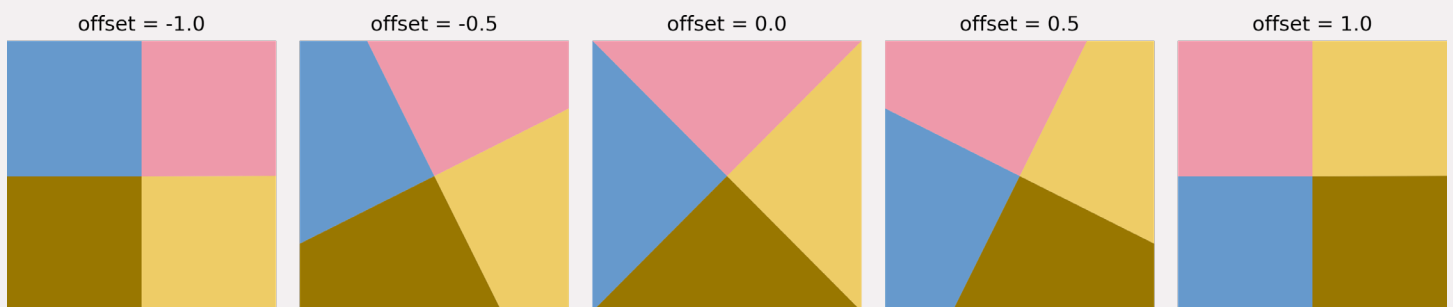
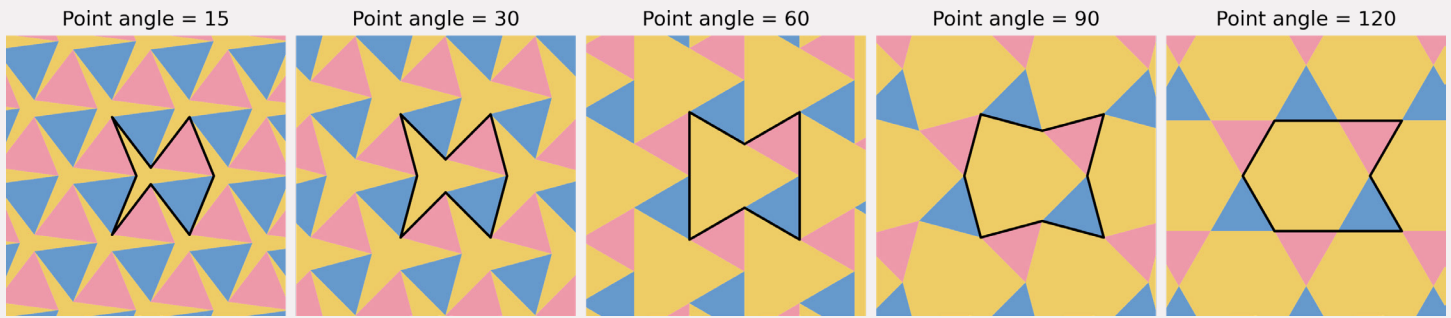


Figure 7. Tileable units produced by varying the Offset setting on a four-slice tiling of the square.



**Figure 8.** The effect of changing the Point angle of some star tilings can be unexpectedly dramatic. The repeating tileable unit composed of three tiles is outlined in black in each panel.

cutting to locations equidistant along its perimeter, to give equal-area tiles. In addition to the number of slices, an **Offset** between -1 and 1 is required. The effect of this setting is shown in Figure 7. Setting **Offset** to 0 means the first slice will start at a corner with successive cuts equidistant around the perimeter. The range of allowed values from -1 to +1 produces a range of different tileable units as shown.

### Star Tiling Point Angles

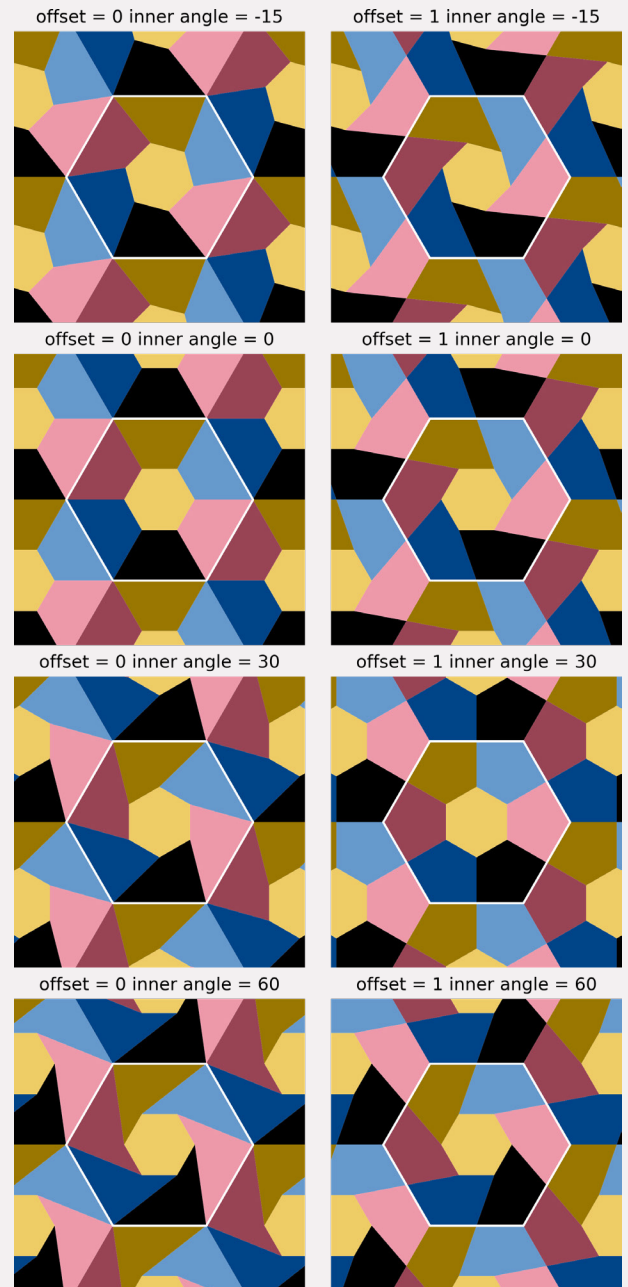
Some star-based tilings designated “star1” in the **Tiling type** drop-down have an associated **Point angle** setting (found under **Set tiling options**), which defines the interior angle of any tiling of this kind’s star polygon. Note that “star2” tilings do not allow this option. This setting is straightforward in its effect but can dramatically change the appearance of a tiling in ways that may not be obvious (see Figure 8). Increasing the point angle will change the size of the star tiles and, as a result, affect the overall scale of the tileable unit, regardless of its nominal spacing. The spacing is approximately correct when the point angle is at its default value of 30°.

### Hexagon and Square Dissection Offset and Inner Angle

MapWeaver’s dissection tilings are generated by adding an interior polygon to the base polygon at its centre and rotating it by the **Inner angle** relative to the outer polygon. Corners of the inner polygon are then connected to either the corners of the outer polygon (**Offset 0**), or the centres of the outer polygon edges (**Offset 1**). Examples of such tilings are shown in Figure 9.

### Weave Unit Over-Under Pattern and Strand Width

For weave type tileable units, two configurable settings are available, **Strand width** and **Over-under pattern**. The



**Figure 9.** Dissected hexagon tilings showing the effect of changing the Offset and Inner angle settings.

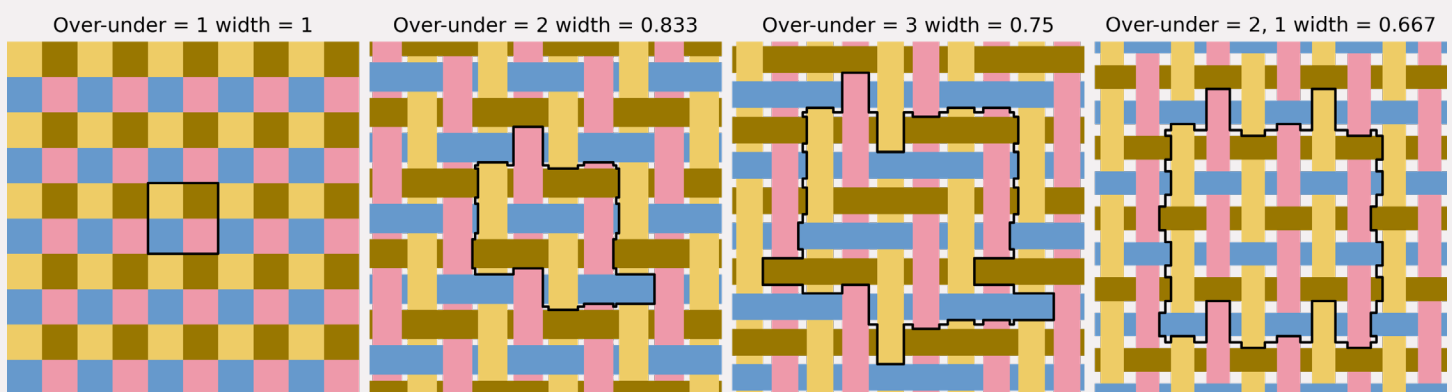
former is simply the width of strands relative to their spacing. As this setting is reduced from 1, which is space-filling, wider gaps open up between strands, giving the appearance of a more open weave. With very low settings the weave pattern will appear more like a cross hatching, although the “lines” are actually long, narrow polygons.

The **Over-under pattern** refers to the number of strands in the other direction that each strand in the pattern passes over, then under. In all cases shown in Figure 10, the base weave is a twill pattern with strands *a* and *b* in one direction, and *c* and *d* in the other. If only a single over-under value is supplied, then strands will pass over that number of opposite strands, then under the same number and repeat. If more than one number is supplied, it is interpreted as a sequence of over, then under steps that the weave pattern will follow. Careful examination of the last panel in Figure 10 will show that each strand running vertically passes over two strands, then under one strand running horizontally. An even number of comma-separated values must be supplied in the **Over-under pattern** box. If an odd number of values is provided, then the sequence will be repeated to yield an even number of values and that will define the weave pattern produced. While complex woven patterns can be produced in this way, long over-under sequences will yield large tileable units, which may be too large for effective mapping outcomes.<sup>7</sup>

## FURTHER OPTIONS

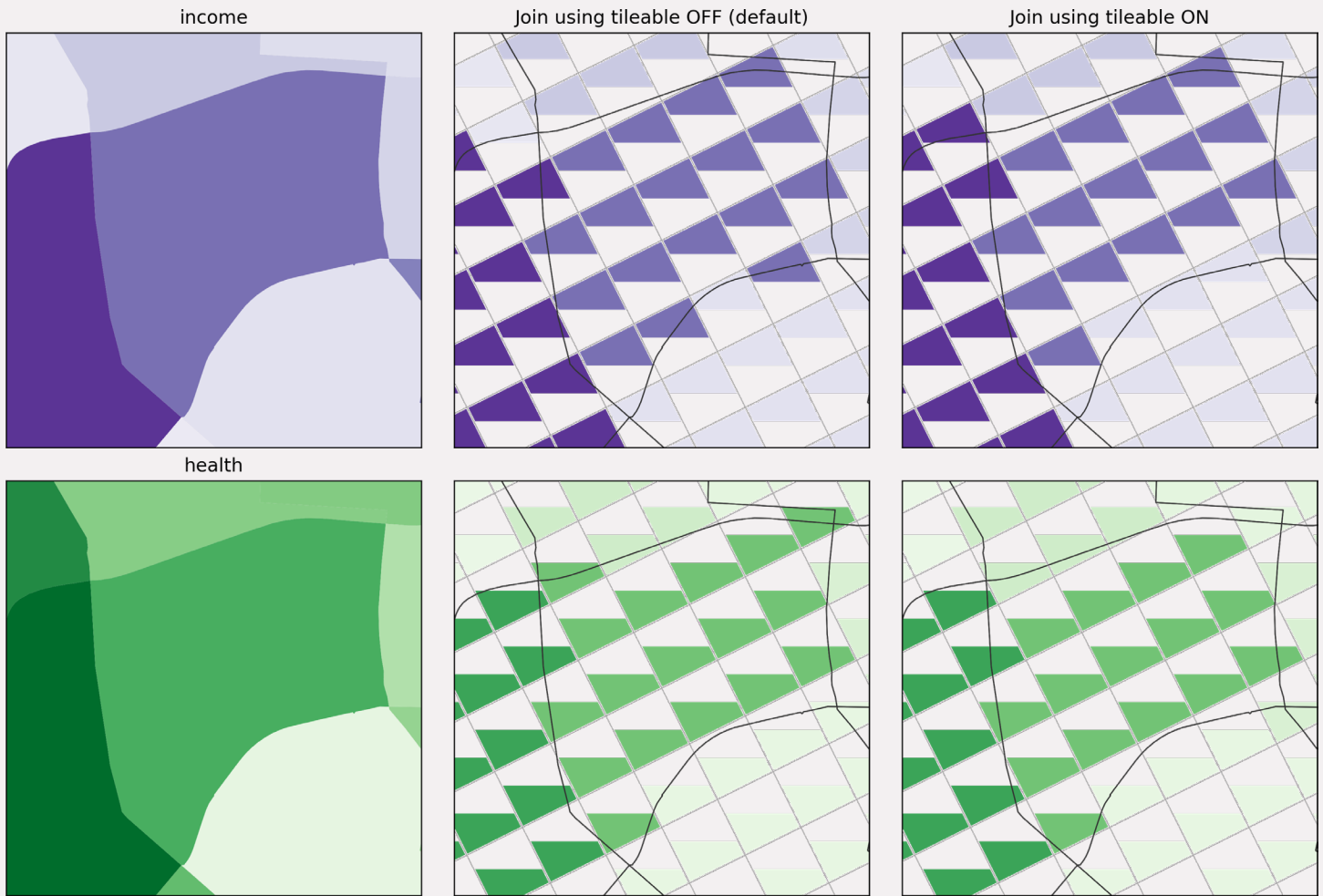
Panel (H) in the MapWeaver interface offers some additional options for control of how the final map is produced:

- When **Join using tileable** is selected, data from map areas are assigned to tiles based on overlap between the tileable unit of which each tile is a member, so that all the elements in any given tileable unit represent data from the same map area (see Figure 11). By default, this option is *not* selected and tiles are assigned data based on the map area that they overlap *individually*, so that tiles in a given tileable unit may represent data from different map areas.
- When **Retain tileables** is selected, all tileable units that overlap any part of the map area are retained in the final map. This may mean that tiles are included in the map that do not overlap any part of the map area, and that the final tiled map extends noticeably beyond the map area. This setting only makes sense when **Join using tileable** is activated. When **Retain tileables** is off, only tiles that overlap the map area are retained.
- Turning **Clip by map units** on clips the final map to the outer boundary of the mapped area. This may result in some tiles in the pattern being reduced to small fragments, but may also help map readers with orientation. Clipping slows down the map refresh, so it is helpful to keep this setting turned off while iterating on your design. The default *off* setting yields maps more consistent with the tiled aesthetic of the method, retaining complete tiles beyond the edges of the map area extent.
- The **Scale as glyphs** setting applies the **Scale EW** and **Scale NS** settings to tileable units independently

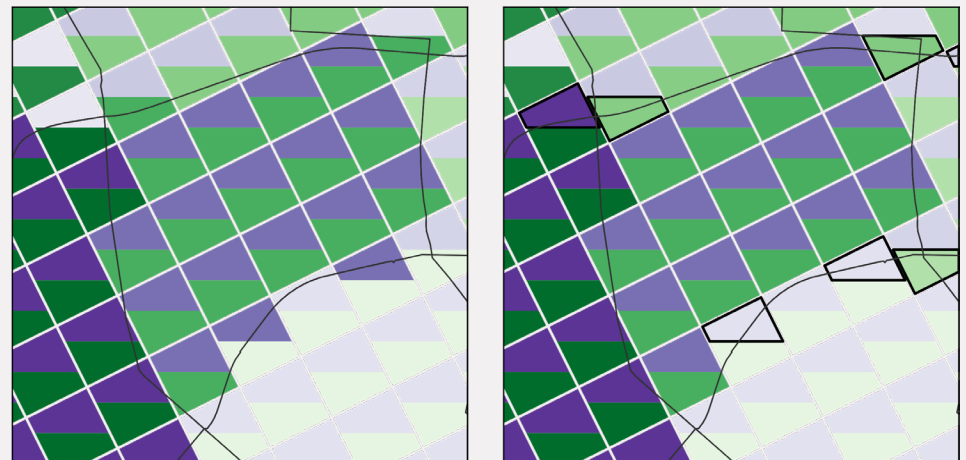


**Figure 10.** The effect of different “Over-under” and “Strand width” settings on a twill weave tileable unit with strands code “ab|cd.”

7. To really get to grips with the possibilities of biaxial weave patterns, we recommend using the `weavingspace` Python module, although the web app does offer a range of built-in options.



**Figure 11.** How **Join using tileable** affects overlay of tiles and source data. When this setting is off, tiles show data from the map area unit that they individually have the greatest overlap with. When set on, tiles show data from the area unit that their “parent” tileable unit has the greatest overlap with. In this example, tiles where this makes a difference are heavily outlined in the lower-right image. With the default setting, tiles in tileable units on the boundary of an area unit may pick up data from neighbouring areas, resulting in tileable units with combinations of attribute values not present in the source data, effectively “blurring” areas together.



from the spacing of the tileable units. This can be used to produce an inset effect without altering the relative size of tiles in each tileable unit, but it is not a viable approach for non-convex tileable units.

- Turning on **Use tileable as icon** treats the tileable unit in your design as an icon and places one per map area rather than creating a tiled map. If you also turn

on the **Show map units** option, how this works will be more apparent.

#### 4. ATTRIBUTES AND COLOUR PALETTES

When a satisfactory tiled or woven (or even tileable unit icon) map has been achieved, the final stage of the design process is to associate tileable unit elements with

data attributes, and data attributes with colour ramps. The **Variables and Palettes selection (F)** section of the interface allows you to make these associations. As noted, it is advisable to keep the spacing of the design large while changing these settings, as changing them quickly can result in the interface and the design getting out of sync with one another, and a larger spacing will make this less likely to happen.

MapWeaver is not a complete mapping platform, so only unclassed visualization of variables is available in the interface, and only numeric attributes in the dataset will be available for selection. If you want to use other classification schemes, to design bespoke colour palettes, or to map categorical variables, you will need to do this in another platform, having first worked on the geometry of the tiles in MapWeaver and then exported the tiled map data for further work elsewhere.

### Colour Design

Our experience designing tiled maps with many variables has taught us that colour choice in such maps is a challenging design problem. We recommend thinking carefully about the relationships among the attributes in your data, and what aspects of those relationships, both statistical and spatial, you wish to emphasize.

## DIGGING DEEPER

---

AS NOTED AT SEVERAL POINTS ABOVE, FINER CONTROL over options in making tiled maps is available using the Python `weavingspace` module. Of particular interest may be the option for creating an indicative legend for your map (as a geospatial data layer). Resources and download of the module are available in the repository at [github.com/DOSull/weavingspace](https://github.com/DOSull/weavingspace). Alternatively, the module may be installed from the Python Package

It may make sense, for example, to assign a number of related variables to the *same* colour palettes, and to assign them to tiles in the design that are close to one another. This has the visual effect of making that group of variables appear like a single combined attribute, but with departures from correlation in the group highlighted as contrasting textures. On the other hand, assigning contrasting colour palettes to all variables may make distinguishing which tiles represent which variables easier.

If your aim is to produce maps that provide an overall sense of the spatial distribution of multiple attributes in your mapped area, then the tools for colour design in MapWeaver may be sufficient. For more control over this aspect of the design, you can use the Python `weavingspace` module, or export your map to a geospatial data format for processing and mapping in other platforms.

### OPTIONS FOR EXPORTING YOUR MAP

You can export your map either as a geospatial data file (GPKG or GeoJSON), as a PNG image, or as an SVG vector file. The geospatial formats will give you the most control over what you can subsequently do with your map. If you just want a quick map to add to a presentation, one of the image export formats might suffice. Those who do much of their cartography in vector graphics software may find the direct export to SVG useful.

Index with the command `pip install weavingspace`. Other efforts to diversify the possibilities of `weavingspace` and MapWeaver are underway. That said, even we are often surprised by what is possible with creative uses of MapWeaver followed by further work within a GIS and/or vector graphics software. We welcome your examples, critiques, comments, and questions.

## REFERENCES

---

Ellis, Erle C., Nicolas Gauthier, Keel Klein Goldewijk, et al. 2021. "People have shaped most of terrestrial nature for at least 12,000 years." *Proceedings of the National Academy of Sciences* 118 (17): e2023483118. <https://doi.org/10.1073/pnas.2023483118>.

Exeter, Daniel John, Jinfeng Zhao, Sue Crengle, Arier Lee, and Michael Browne. 2017. "The New Zealand Indices of Multiple Deprivation (IMD): A new suite of indicators for social and health research in Aotearoa, New Zealand." *PLOS ONE* 12 (8): e0181260. <https://doi.org/10.1371/journal.pone.0181260>.