An Algorithm For Spatial Mapping Using a Hexagon Tilegram, With Application to Australian Maps

by Stephanie Kobakian, Dianne Cook

Abstract This algorithm creates a hexagon to represent each of the spatial polygons provided. It allocates these hexagon in a manner that preserves the spatial relationship of the areas. It showcases spatial distributions, by emphasising the small geograhical regions that are often diffcult to locate on large geographic maps.

Introduction

Spatial distrbutions have utilised alternative representations of geography for many years. In modern times interactivity and animation have begun to play a larger role, as alternative representations have been popularised by online news sites, with a focus on public consumption. Applications are increasingly widespread, especially in the areas of disease mapping, and election results.

Motivation

Spatial distributions of people and human related statistics can be misrepresented in geographic maps. Populations congregate around major cities and vertical living is increasingly common. Population statistics often requires dividing people into smaller, measureable areas. Government bodies such as the Australian Bureau of Statistics, or intentionally segregated organisations, like the Australian Electoral Comission hold the responsibility of the division. Australia is an example of strong urbanisation, the rural areas are often sparsely populated in comparison to the urban centres. To divide the population equally, the square meterage of the geographic areas differ dramatically in size.

Alternative mapping methods allow increased understanding of the spatial distribution of a variable across the population. Alternative maps allow the focus to be placed on the distribution of the statistics between the groups of areas.

Extremely small geographic areas are lost on large geographic maps, small inner areas of Sydney or Melbourne are not easily compared at a high level. Cartograms place importance on the statistic of interest, allowing distorted map space on the display to represent differences in the statistic. These maps rely on the statistic of interest to determine the layout, and for Australia, often fail to preserve a recognisable view. Tilegrams allow preservation of spatial relationships as they decrease the emphasis on the amount of geograhic area considered to be interesting. These maps focus on the relationship between neighbours, in considering aggregated statistics of heterogeneous regions.

Our case is an extension of these concepts. Extending the tilegram to Australian applications required preserving the spatial relationships. It emphasises the capital cities as population hubs, and recognises the populations distributed across the large rural areas.

Algorithm

Our solution operates on a set of polygons. There are parameters used in the process that may be provided, or will be automatically derived. All necessary functions are exported, with a main function `create_hexmap` used to step through these automatically.

Parameters

The `create_hexmap` function requires several parameters, if they are not provided, the information will be derived from the polygon set used.

The following must be provided to `create_hexmap`:

parameter	description
shp	an Rdata object containing the polygon information
shp_path	character string location of the shape file that contains the polygon information
sf_id	name of a unique column that distinguishes areas
capital_cities	a data frame of reference locations used to allocate hexagons

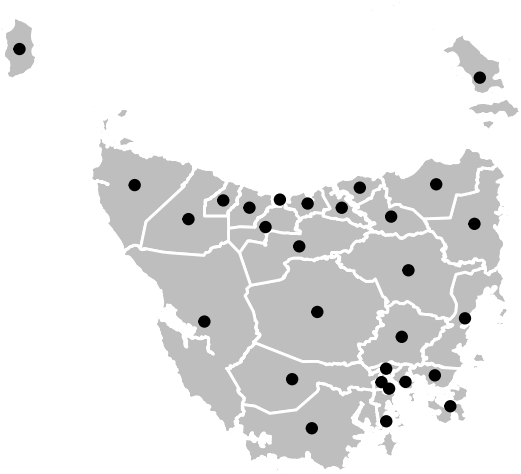


Figure 1: lga Local Government Areas and derived centroids.

The polygon set of Local Government Areas in Tasmania in 2016 is stored in the sugarbag package data as `tas_lga`. A single column of the data set must be used as a unique identifier of areas. In this case, the unique LGA codes associated with each LGA have been used, their names could have also been used.

The centre of the population hubs are used to distribute areas around the closest captial city.

The following parameters will be determined within `create_hexmap` if not provided. They are created throughout the following example:

parameter	description
buffer_dist	a float value for distance in degrees to extend beyond the - geometry provided
hex_size	a float value in degrees for the diameter of the hexagons
hex_filter	amount of hexagons around centroid to consider for allocation
width	the angle used to filter the grid points around a centroid

When utilising individual `sugarbag` functions, we recommend the following approach:

Polygon Information

Begin with a set of polygons. The set should be provided as an `sf` object, this is a data frame containing a `geometry` column. The `read_shape` function can assist in creating this object. We will use the Loca Goverment Areas of Tasmania, sourced from the Australian Bureau of Statistics.

Hexagon grid

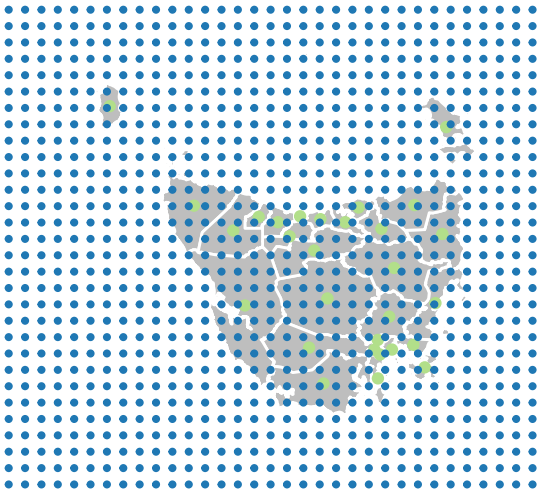
Each new hexagon will need to tesselate, a grid is created to ensure a close point is found, that will allow all areas to tesselate, this is inspired by tilegrams. The grid of possible hexagon centroids can be made using the `create_grid` function. The grid creation takes several steps. It requires the centroids, the hexagon size and the buffer distance.

Step 1: Expand the grid

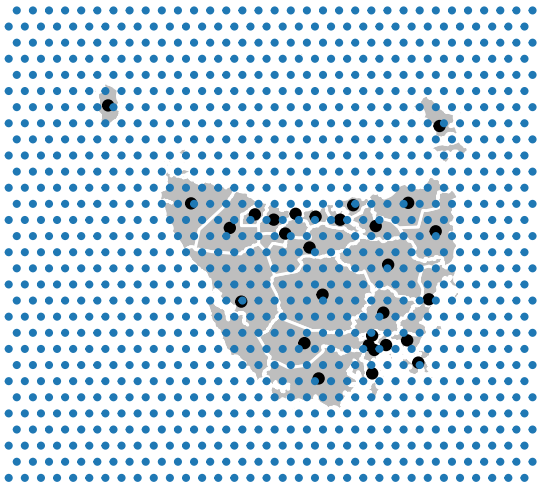
One sequence is created for longitude columns, and another for latitude rows.

The sequences begin at the minimum longitude or latitude, minus the buffer distance. Equally spaced intervals the size of the hexagons are created up to the maximum longitude or latitude, plus the buffer distance.

An individual point is created from all intersections of the longitude columns and latitude row sequences.



A square grid will not facilitate tessellated hexagons. Every second latitude row of points will be shifted right by half of the hexagon size.



Each point is now given an ID.

Step 2: Rolling windows

Not all of the grid points will be used, especially if islands impact the overall space covered. To filter the grid for appropriate points for allocation, the `create_buffer` function is called from `create_grid`. It finds the amount of columns and rows needed to best capture the set of centroids.

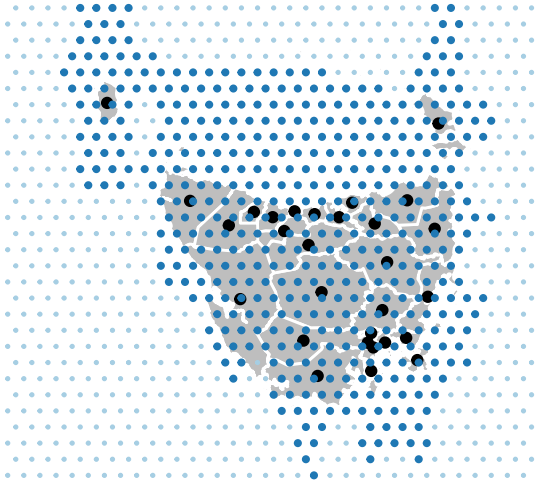
The rows and columns are divided into 20 groups. The amount of rows in each latitude group and the amount of columns in each longitude group are then used as the width of manual rolling windows. This will tailor the available points to the areas most likely to be used. It helps reduce amount of time taken.

The specific rows and columns in the rolling windows are defined.The first rolling window function finds the minimum and maximum centroid values for the sliding groups of longitude columns and the groups of latitude rows.

The second rolling window function finds the average of the rolling minimum and rolling maximum centroid values, for the longitude columns and latitude rows.

Step 3: Filtering the grid

Only the grid points between the rolling average of the minimum and maximum centroid values are kept, for each row and column of the grid.



Centroid to focal point distance

For each polygon centroid in the set, the distance to each of the focal points provided is recorded. The closest focal point name, the distance to the polygon centroid, and the angle from focal point to polygon centroid will be added to the polygon's row, in the polygon data set. To minimise time taken for this step, only Tasmania's capital city Hobart has been provided.

The distance between the polygon centroid and it's closest focal point data set is used to order the data set for allocation. The points are arranged in ascending order, from the centroid closest to any of the focal points, to the furthest.

Allocation of centroids

The distance around a centroid to consider for possible hexagon locations is determined by the hex_filter. It multiplies the amount given by hex_filter, by the size of the hexagons to find the distance.

Allocation of all centroids can now take place using the set of polygon centroids and the hexagon map grid. For each polygon centroid, only the hexagon grid points that have not yet been used can be considered. Allocate each centroid, beginning with the closest centroid to a focal point. This will preserve spatial relationship with the capital, as the inner areas will be allowed closest to the captial, and the areas that are further will be accomodated after.

The filter parameter is used to subset possible grid points to only those surrounding the polygon centroid within the filter distance, smaller distance will increase speed, but can decrease accuracy.

The following example considers one of the Local Government Areas. These steps are repeated for each polygon.

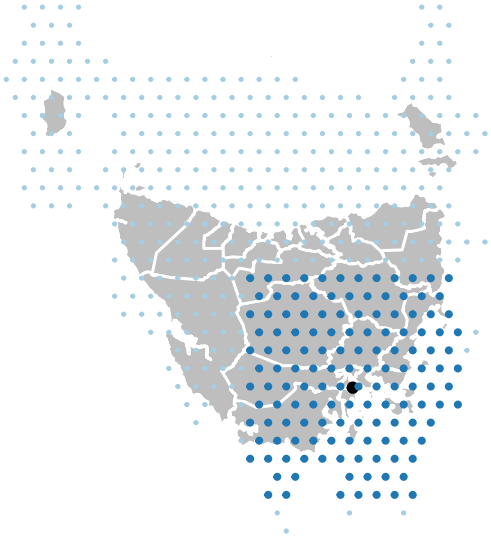
Step 1: Filter the grid for unassigned hexagon points

Keep only the available hexagon points, this will prevent multiple areas being allocated to the same hexagon.

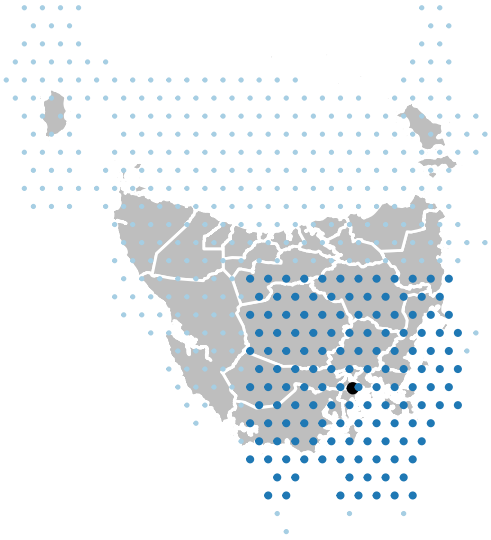
Step 2: Filter the grid points for those closest to the centroid

This will allow only the closest points, that are not assigned, to be considered.

Create a box filter



Create circle filter



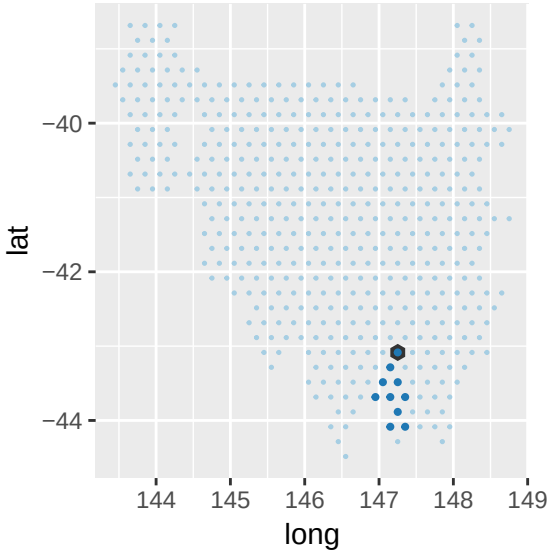
Filter for angle within circle, relates hexagon location to focal point, Sydney.

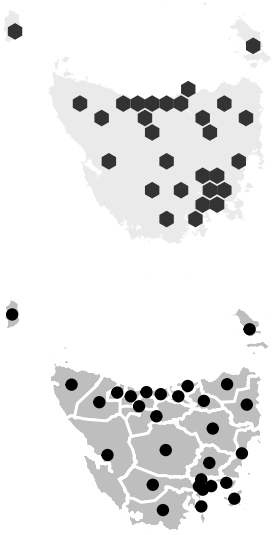
Using the angle between each polygon centroid, and it's the closest focal point, the subset of points

is filtered again. Of these possible points, only those within a specific amount of degrees plus and minus a specific angle range are kept. This angle begins at 30 degrees by default, and may increase if necessary.

If no available hexagon grid point is found within the original filter distance and angle, the distance is expanded, only when a maximum distance is reached will the angle expand to accommodate more possible grid points.

The allocation is returned and combined with the data relating to each polygon.





Applications of algorithm

Examples of Melbourne, lga, Australia At different levels, distance from original centroids, change in area

LGA of all Australia SA2 for lga

Stephanie Kobakian
Queensland Univeristy of Technology

stephanie.kobakian@monash.edu

Dianne Cook
Monash University

dicook@monash.edu