

Lesson 3: Choropleth Mapping and tmap Methods

Dr. Kam Tin Seong

Assoc. Professor of Information Systems

**School of Computing and Information Systems,
Singapore Management University**

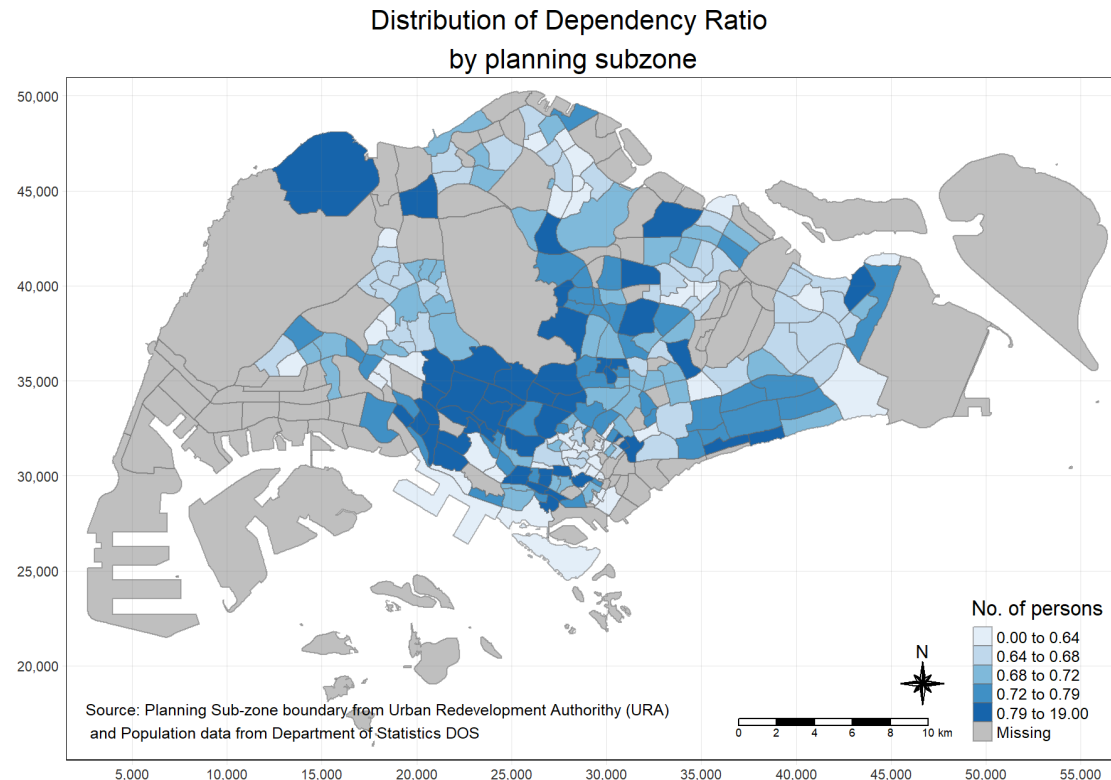
2019/11/25 (updated: 2021-08-29)

Content

- Choropleth Mapping with in R
 - Type of choropleth map
 - Choosing number of classes
 - Data classification methods
 - Selecting colour scheme
- tmap Methods
 - tmap framewok
 - tmap elements
 - tmap layers
 - tmap layout
 - tmap style
 - tmap facet

Choropleth Map

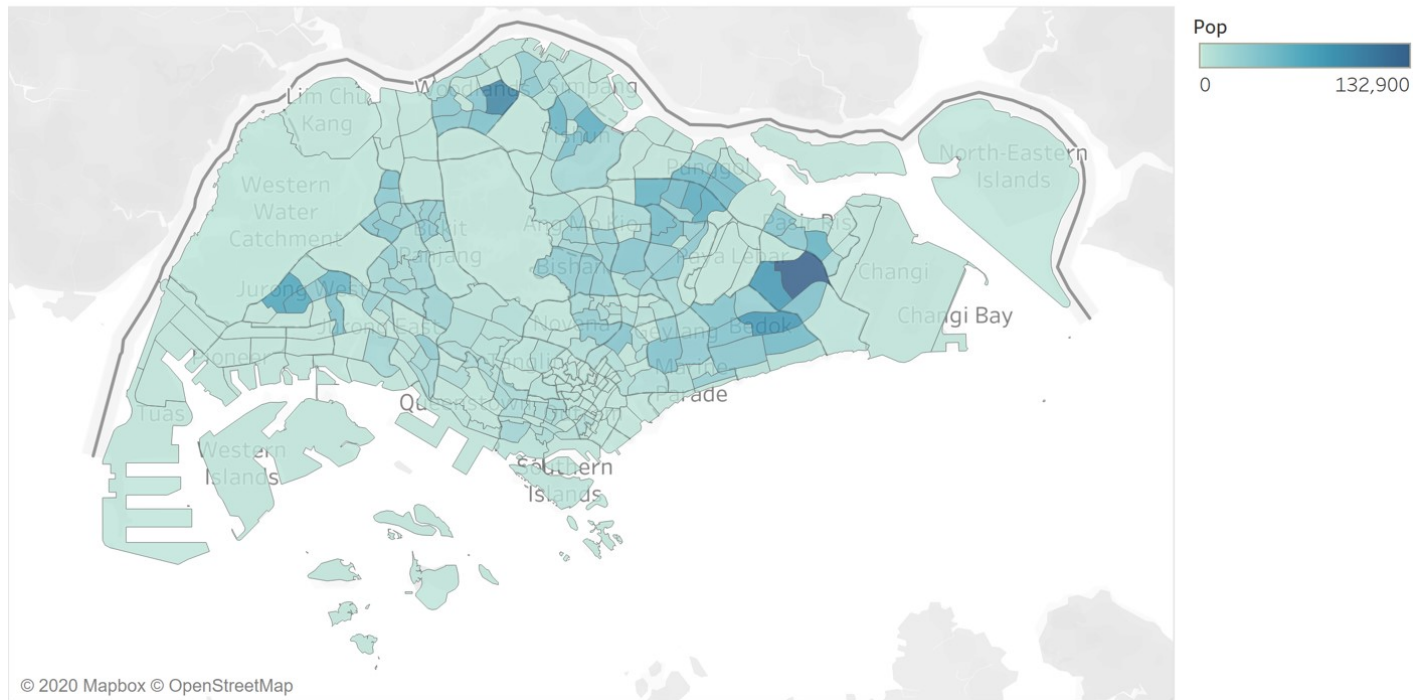
A choropleth map is a type of thematic map in which areas are shaded or patterned in proportion to a statistical variable that represents an aggregate summary of a geographic characteristic within each area, such as population or per-capita income.



Classified choropleth map

- Choropleth maps can be either classified or unclassified.
- A classed choropleth map combines areal units into a smaller number of groups. Interval levels may vary, but typically 4 to 7 are used in a map. There is different classification techniques used to divide up the intervals.

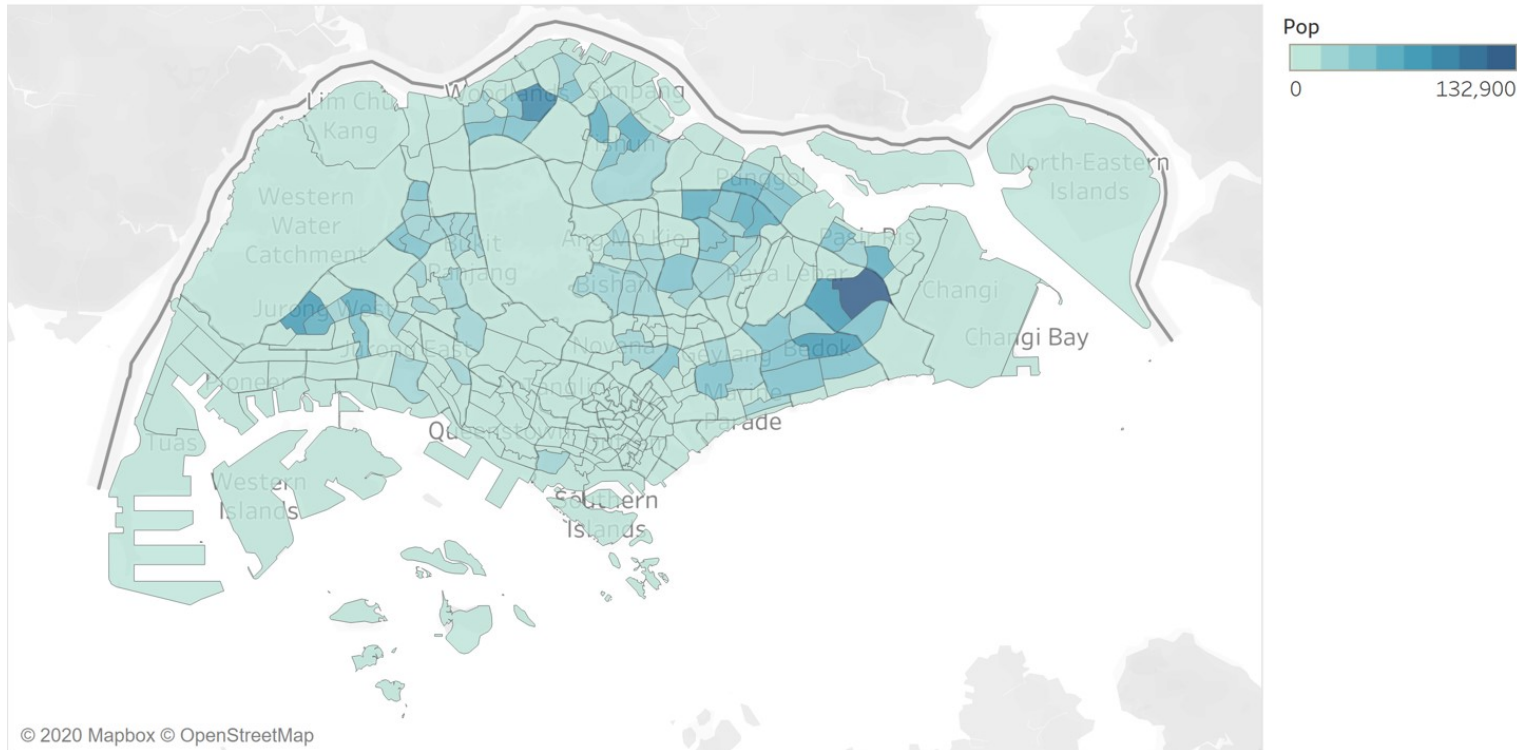
Distribution population by planning subzone, 2019



Unclassified choropleth map

- Unclassed choropleth maps are similar to classed choropleth maps; however, unclassified choropleth maps do not have an averaged statistic towards each particular colour.

Distribution population by planning subzone, 2019



Choosing an appropriate number of classes

- Sturges' formula

$$k = 1 + 3.32 * \log n$$

n = number of values

k = number of classes

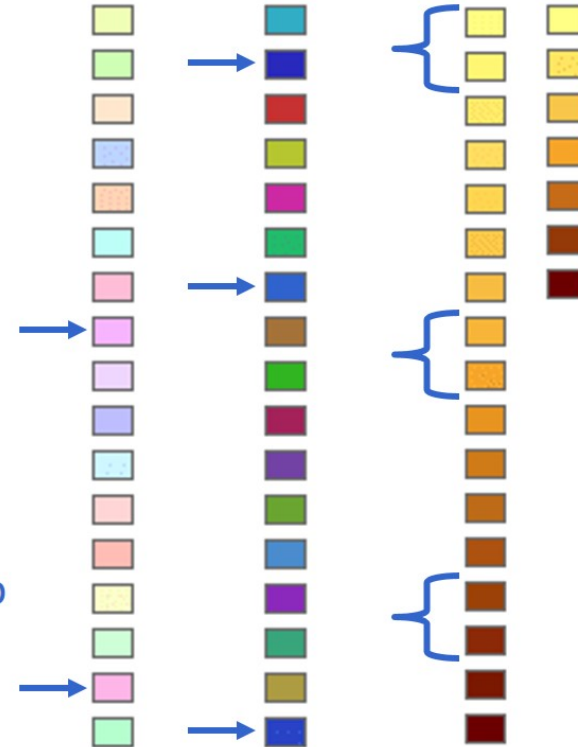
If $n = 36$

$$k = 1 + 3.32 * \log n$$

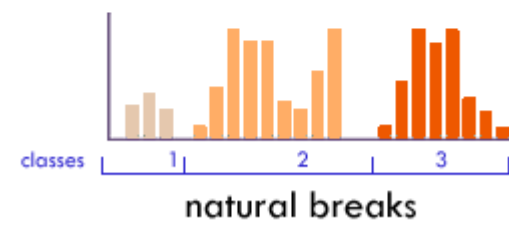
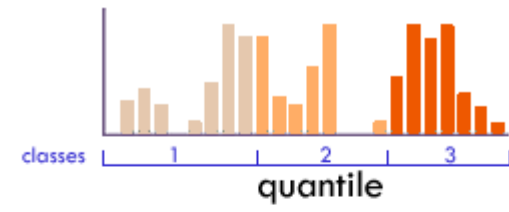
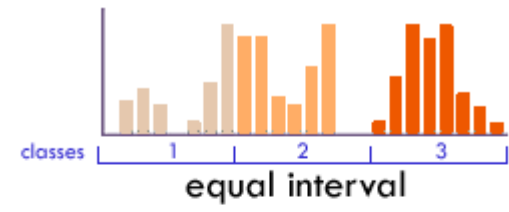
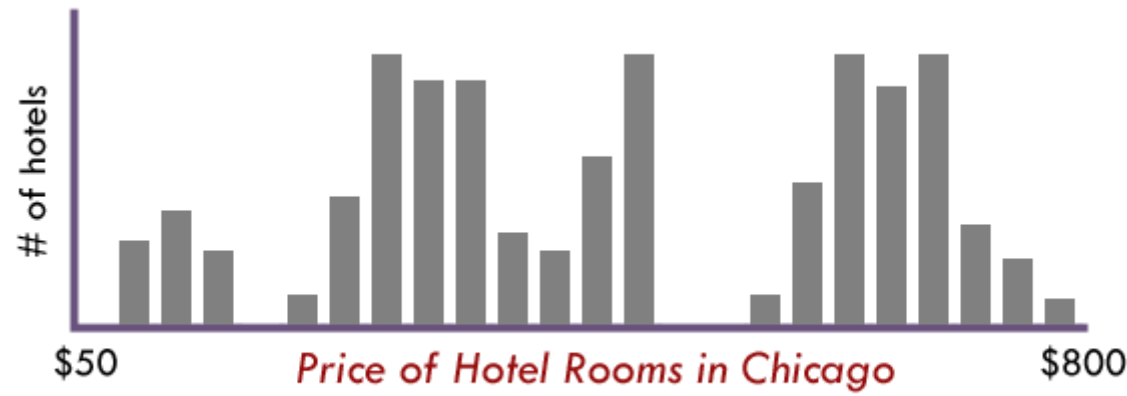
$k = 6$ approx.

Example

- Number of intervals?
 - Not less than 4
 - To avoid an overly generalized map
 - Not more than
 - 12 colors
 - 7 or 8 shades of the same color



Data classification



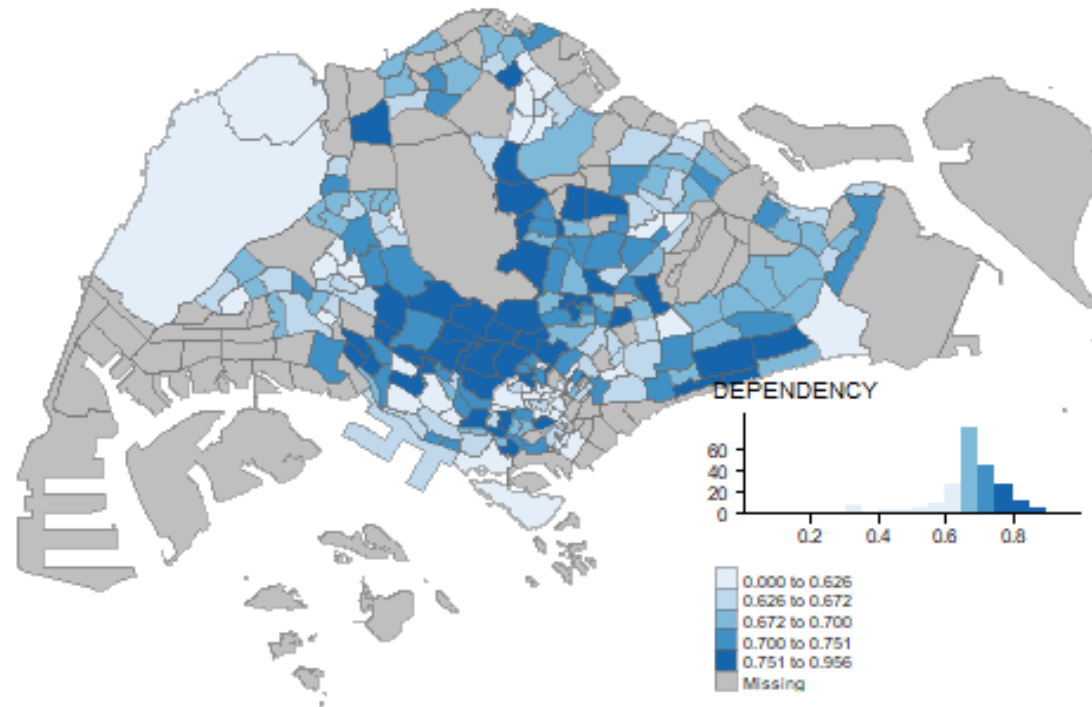
Methods of choosing classes

- Based on the nature of the distribution
 - quantile, equal interval, natural breaks, standard deviations, defined interval
- Arbitrary
 - Can be based on round numbers.
 - Examples: Grouping according to age or census housing categories
 - Can result in empty categories

Data classification method: Quantile

- Same number of features per class.

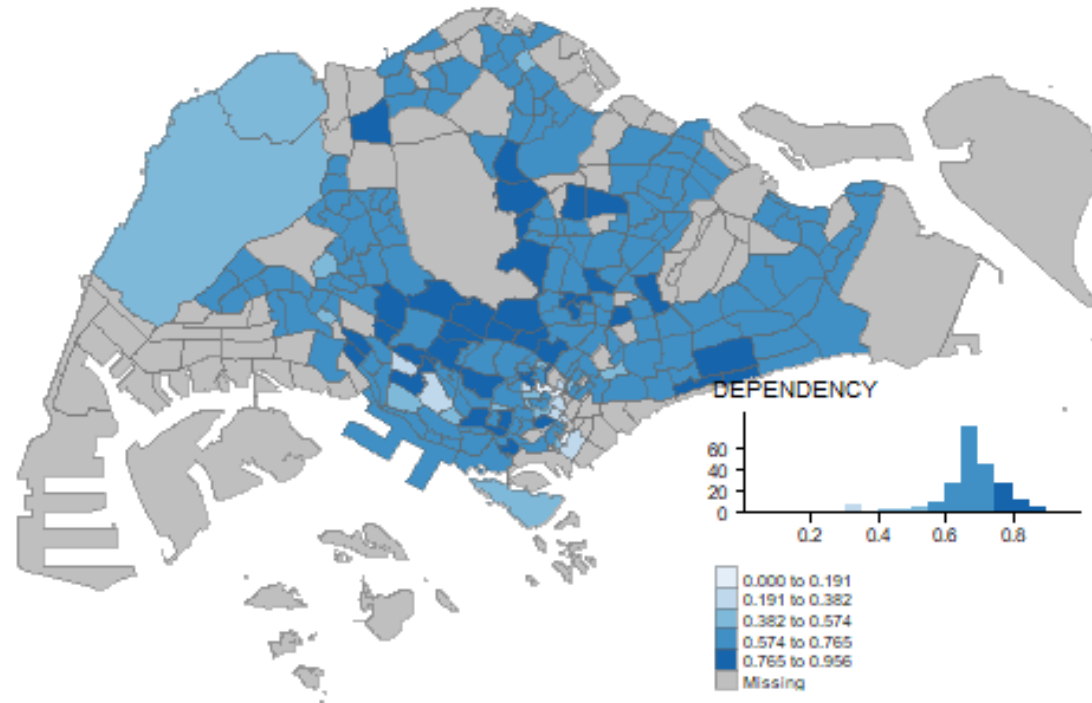
Distribution of Dependency Ratio by planning subzone
(quantile classification)



Data classification method: Equal interval

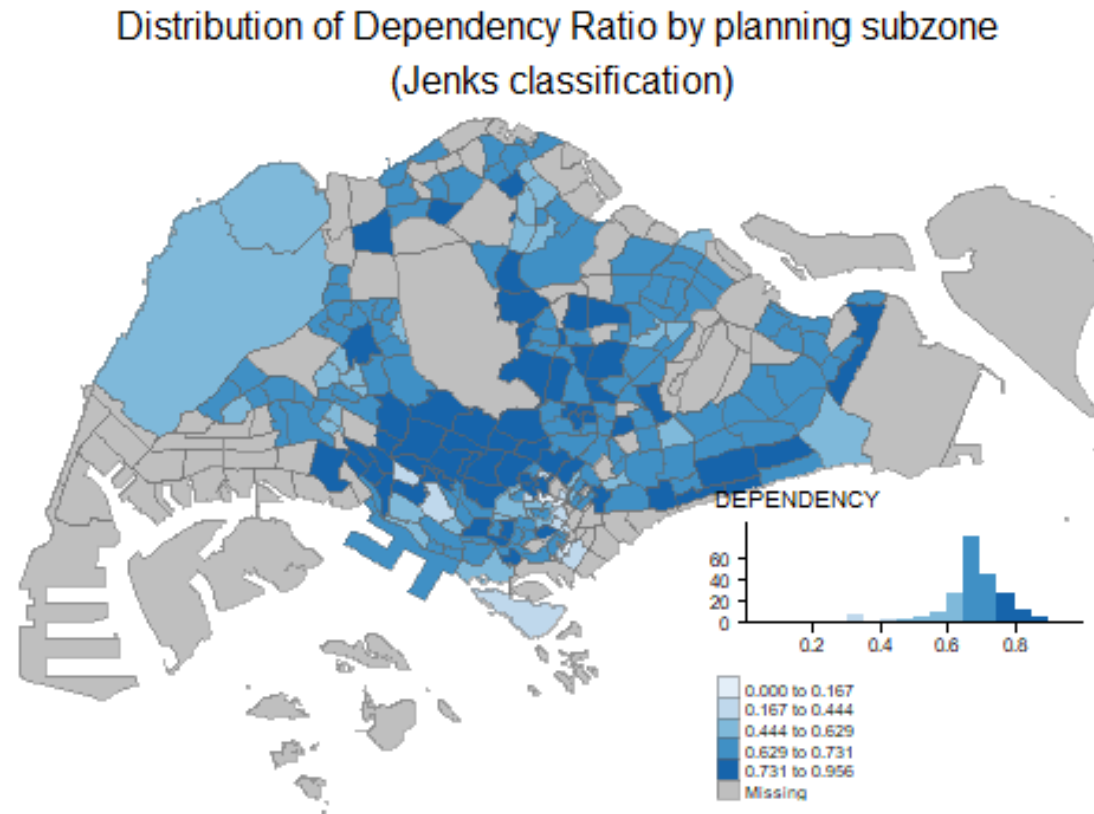
- Divides the range of attribute values into equally sized classes.
- CAUTION: Avoid equal interval if your data are skewed to one end or if you have one or two really large outlier values.

Distribution of Dependency Ratio by planning subzone
(Equal interval classification)



Data classification method: Jenks (also known as Natural breaks)

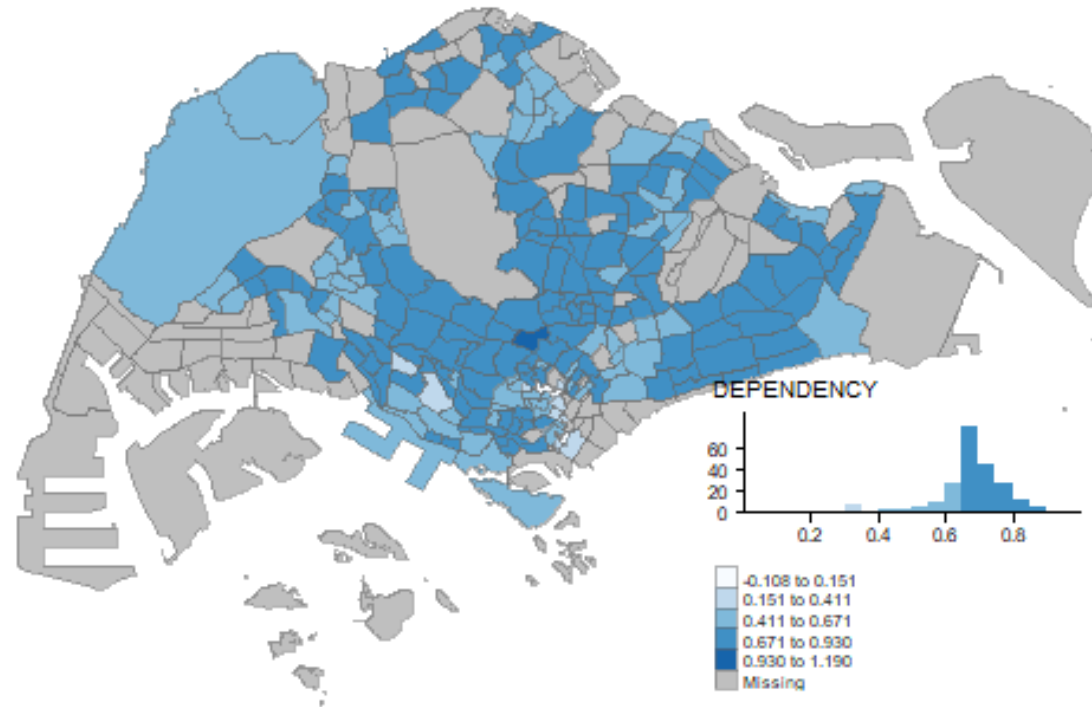
- Default Jenk's statistical optimization
- Finds natural groupings in the data



Data classification: Standard deviation

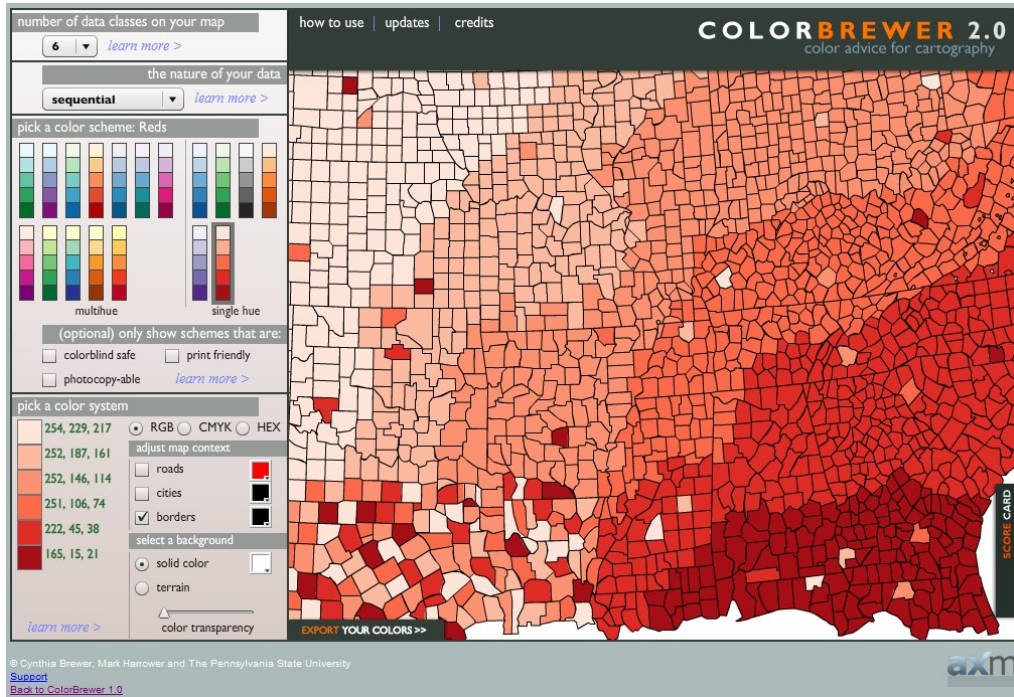
- A measure of dispersion.
- Use if the distribution approximates a normal distribution (bell-shaped curve)

Distribution of Dependency Ratio by planning subzone
(sd classification)

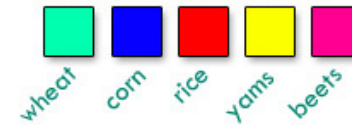


Colour scheme

ColorBrewer

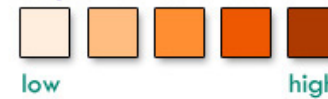


Nominal Color Scheme



different hues that keep lightness and saturation constant should be used for **nominal data** (i.e., un-orderable categories, not numerical data).

Sequential Color Scheme



any sequence that is **dominated by changes in lightness** can be used with orderable (rankable) categories (low/med/high) or with numerical data.

Diverging Color Scheme



any numerical data that can be divided meaningful at a **mid-point** (e.g., national average, zero) can use a diverging scheme: the data are split in two around the lightest, middle color/class.

Mapping packages in R

Selected popular mapping packages

CRAN Task View: Analysis of Spatial Data

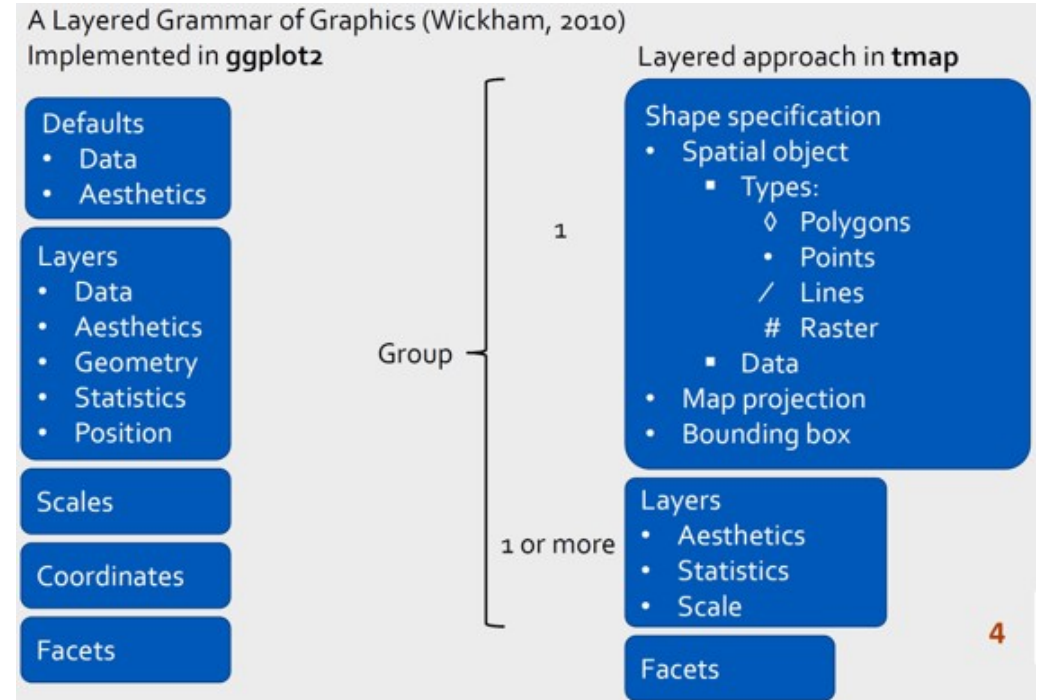
- tmap
- mapsf
- leaflet
- ggplot2. Read [Chapter 6: Maps](#) of 'ggplot2: Elegant Graphics for Data Analysis' for more detail.
- ggmap
- quickmapr
- mapview

Other packages

- RColorBrewer
- classInt

Introducing *tmap*

- *tmap* is a R package specially designed for creating thematic maps using the principles of the **Grammar of Graphics**.
- It offers a flexible, layer-based, and easy to use approach to create thematic maps, such as choropleths and proportional symbol maps.
- It supports two modes: *plot*(static maps) and *view* (interactive maps).
- It provides shiny integration(with *tmapOutput* and *renderTmap*).



Shape objects

- *tmap* supports **simple features** from the new *sf* package.
- It also supports the class *Spatial* and *Raster*, respectively from the *sp* and the *raster* package. The supported subclasses are:

	Without data	With data
Polygons	SpatialPolygons	SpatialPolygonsDataFrame
Points	SpatialPoints	SpatialPointsDataFrame
Lines	SpatialLines	SpatialLinesDataFrame
Raster	SpatialGrid	SpatialGridDataFrame
Raster	SpatialPixels	SpatialPixelsDataFrame
Raster		RasterLayer
Raster		RasterBrick
Raster		RasterStack

```
## Reading layer `MP14_SUBZONE_WEB_PL' from data source
##   `D:\tskam\IS415\lesson\Lesson03\data\geospatial' using driver `ESRI Shapefile'
## Simple feature collection with 323 features and 15 fields
## Geometry type: MULTIPOLYGON
## Dimension:      XY
```


Plotting functions of *tmap*

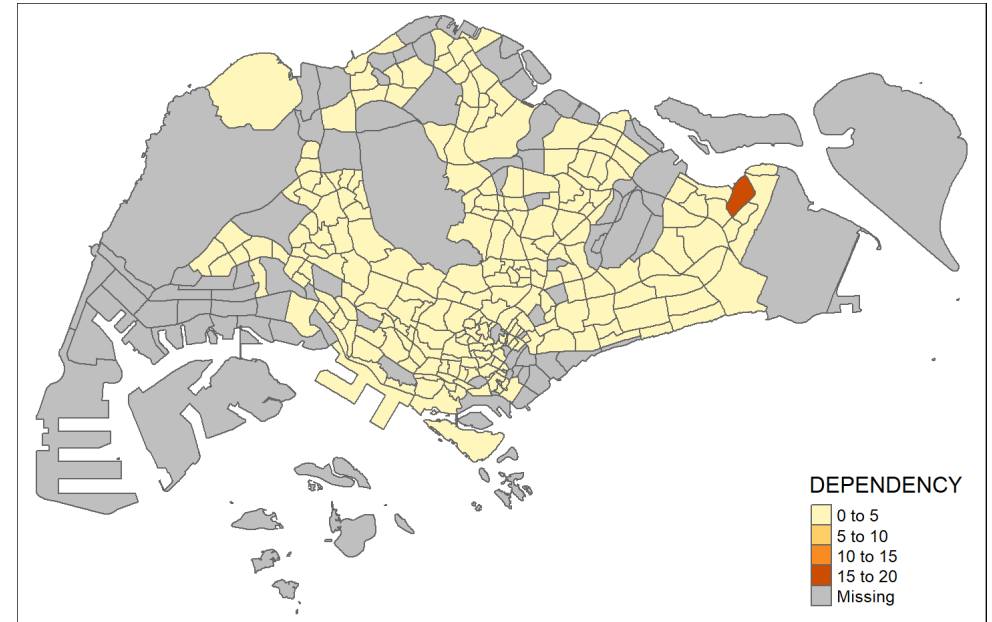
Two approaches can be used to prepare thematic map using *tmap*, they are:

- Plotting a thematic map quickly by using `qtm()`.
- Plotting highly customisable thematic map by using *tmap* elements.

Plotting a choropleth map quickly by using *qtm()*

The easiest and quickest to draw a choropleth map using **tmap** is using *qtm()*. It is concise and provides a good default visualisation in many cases.

- *tmap_mode()* with "plot" option is used to produce a static map. For interactive mode, "view" option should be used.
- *fill* argument is used to map the attribute (i.e. DEPENDENCY)



*tm*map elements

tm_shape()

- The first element to start with is tm_shape(), which specifies the shape object.

Function	Argument	Description	Spatial types
tm_shape	shp	shape object	polygons
			points
			lines
			raster cells
	projection	projection code	
	bbox	bounding box	

tmap elements

Base layers

- Next, one, or a combination of the following drawing layers should be specified:

Drawing layer	Description	Aesthetics
Base layer		
<code>tm_polygons</code>	Draw polygons	col
<code>tm_symbols</code>	Draws symbols	size, col, shape
<code>tm_lines</code>	Draws polylines	col, lwd
<code>tm_raster</code>	Draws a raster	col
<code>tm_text</code>	Add text labels	text, size, col

tmap elements

Base layers

- Each of these functions specifies the geometry, mapping, and scaling component of the LGTM.
- An aesthetic can take a constant value, a data variable name, or a vector consisting of values or variable names.
- If a data variable is provided, the scale is automatically configured according to the values of this variable, but can be adjusted with several arguments. For instance, the main scaling arguments for a color aesthetic are color palette, the preferred number of classes, and a style to create classes.
- Also, for each aesthetic, except for the text labels, a legend is automatically created.
- If a vector of variable names is provided, small multiples are created, which will be explained further below.

*tm*map elements

Derived layers

- The supported derived layers are as follows:

Derived layer		
<code>tm_fill</code>	Fills the polygons	see <code>tm_polygons</code>
<code>tm_borders</code>	Draws polygon borders	none
<code>tm_bubbles</code>	Draws bubbles	see <code>tm_symbols</code>
<code>tm_squares</code>	Draws squares	see <code>tm_symbols</code>
<code>tm_dots</code>	Draws dots	see <code>tm_symbols</code>
<code>tm_markers</code>	Draws markers	see <code>tm_symbols</code> and <code>tm_text</code>
<code>tm_iso</code>	Draws iso/contour lines	see <code>tm_lines</code> and <code>tm_text</code>

tmap elements

Derived layers

- Each aesthetic can take a constant value or a data variable name. For instance, `tm_fill(col="blue")` colors all polygons blue, while `tm_fill(col="var1")`, where "var1" is the name of a data variable in the shape object, creates a choropleth.

Drawing a base map

The basic building block of **tmap** is `tm_shape()` followed by one or more layer elements such as `tm_fill()` and `tm_polygons()`.

Be warned: The "+" sign should be place at the end of a code line and not at the front of a code line.

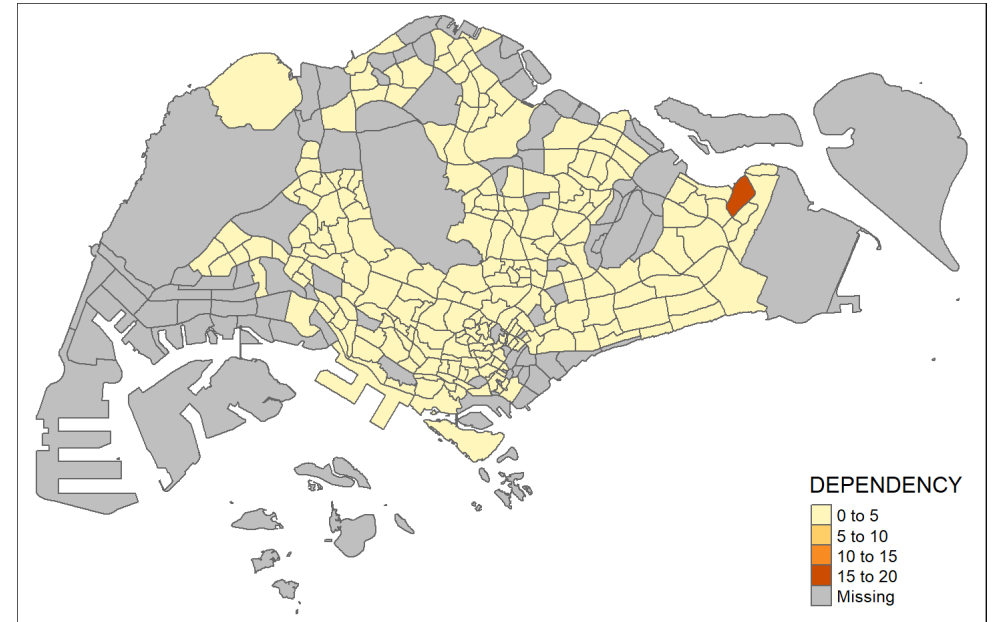


Drawing a choropleth map using *tm_polygons()*

To draw a choropleth map showing the geographical distribution of a selected variable by planning subzone, we just need to assign the target variable such as *DEPENDENCY* to *tm_polygons()*.

Things to learn from *tm_polygons()*:

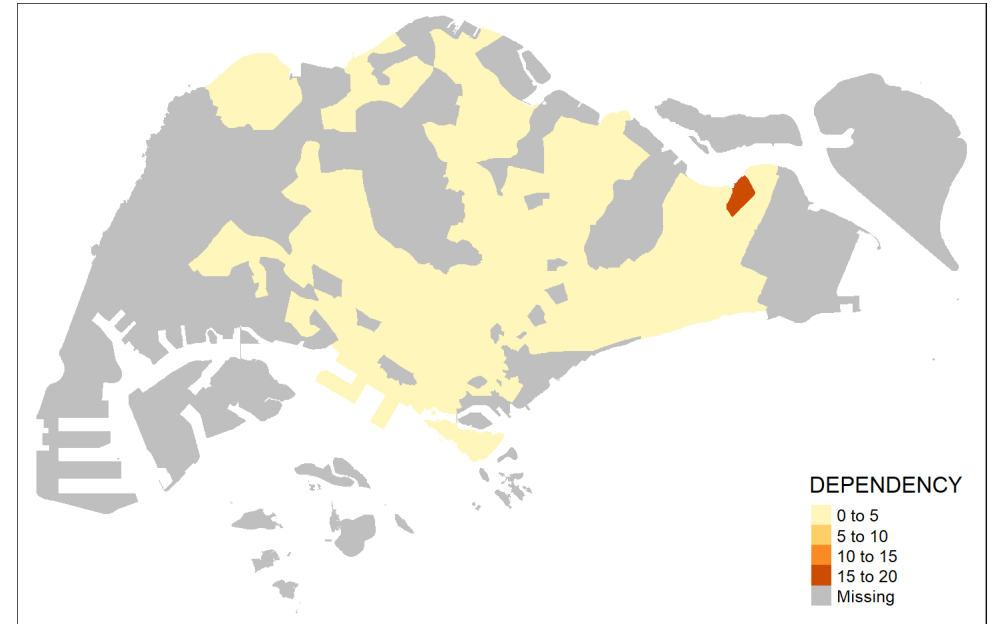
- By default, 5 bins will be used.
- The default data classification method used is called "pretty".
- The default colour scheme used is "YlOrRd" of ColorBrewer. You will learn more about the color palette later.
- By default, Missing value will be shaded in gray.



Drawing a choropleth map using *tm_fill()* and *tm_border()**

Actually, *tm_polygons()* is a wrapper of *tm_fill()* and *tm_border()*. *tm_fill()* shades the polygons by using the default colour scheme and *tm_borders()* adds the borders of the shapefile onto the choropleth map.

- Notice that the planning subzones are shared according to the respective dependency values

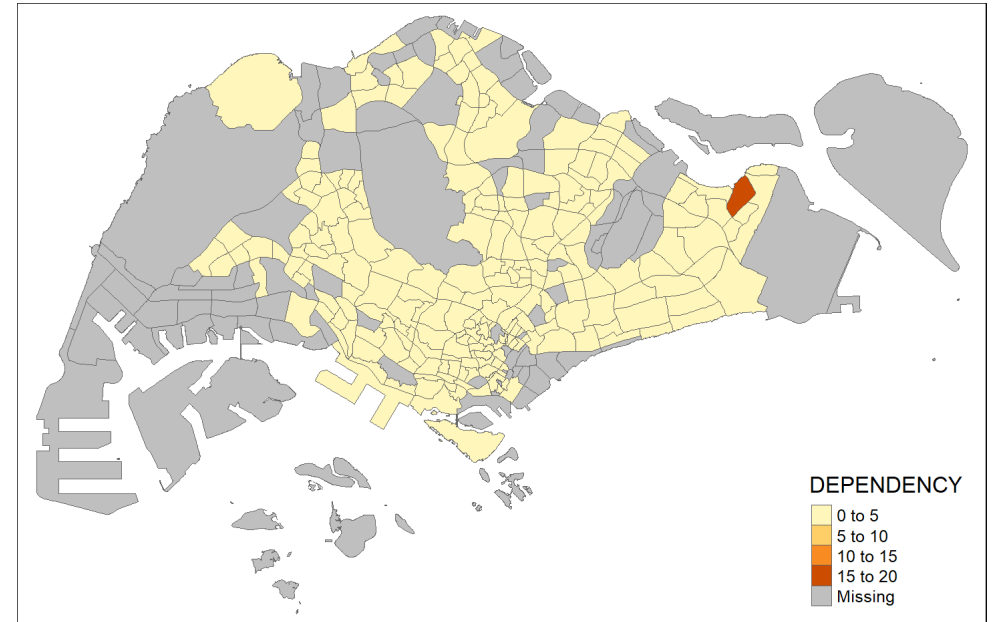


Drawing a choropleth map using `tm_border()`*

To add the boundary of the planning subzones, `tm_borders` will be used.

Notice that light-gray border lines have been added on the choropleth map.

- *lwd* = border line width. The default is 1,
- *alpha* = transparency number between 0 (totally transparent) and 1 (not transparent). By default, the alpha value of the col is used (normally 1),
- *col* = border colour, and
- *lty* = border line type. The default is "solid".



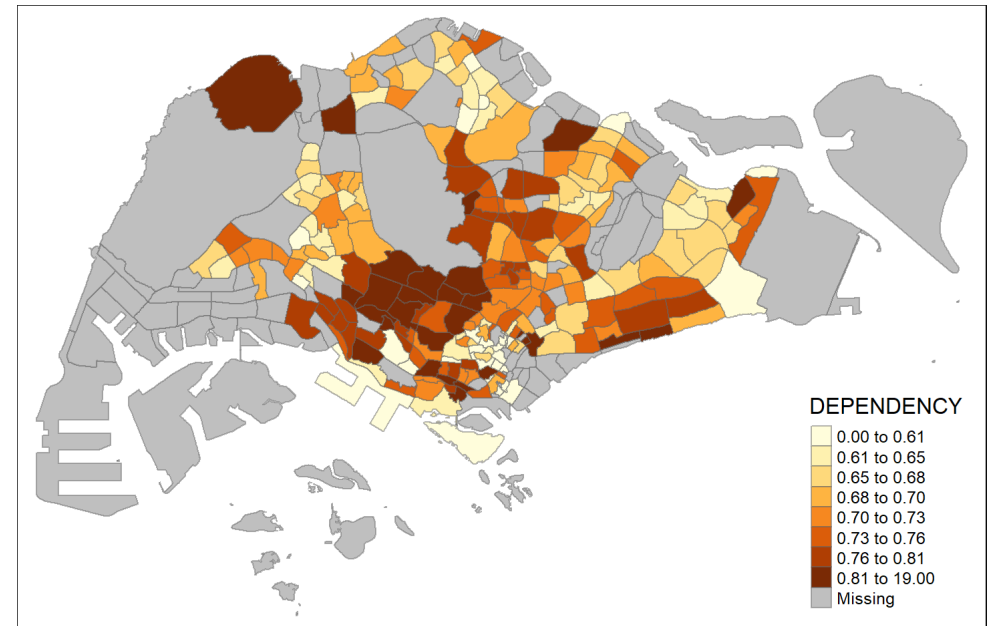
Data classification methods of tmap

Most choropleth maps employ some method of data classification. The point of classification is to take a large number of observations and group them into data ranges or classes.

tmap provides a total ten data classification methods, namely: *fixed*, *sd*, *equal*, *pretty* (default), *quantile*, *kmeans*, *hclust*, *bclust*, *fisher*, and *jenks*.

To define a data classification method, the *style* argument of *tm_fill()* or *tm_polygons()* will be used.

The choropleth map on the right shows a quantile data classification with 8 classes are used.



Comparing Quantile and Equal Interval

Choropleth maps on the left and right below show *quantile* and *equal* data classification methods are used.

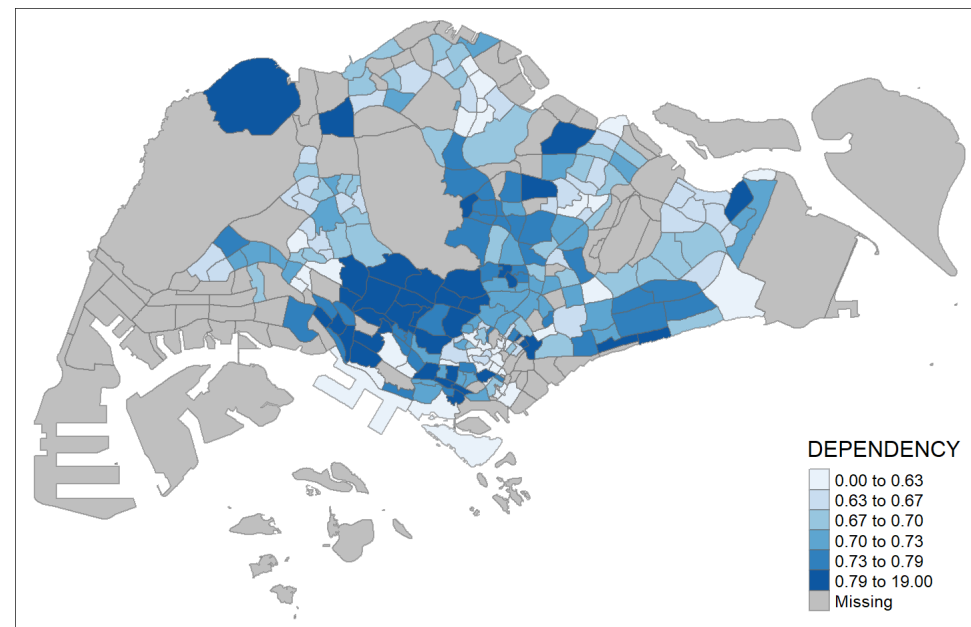
- Notice that the distribution of quantile data classification method are more evenly distributed then equal data classification method.

Colour Scheme

tmap supports colour ramps either defined by the user or a set of predefined colour ramps from the **RColorBrewer** package.

To change the colour, we assign the preferred colour to *palette* argument of *tm_fill()*.

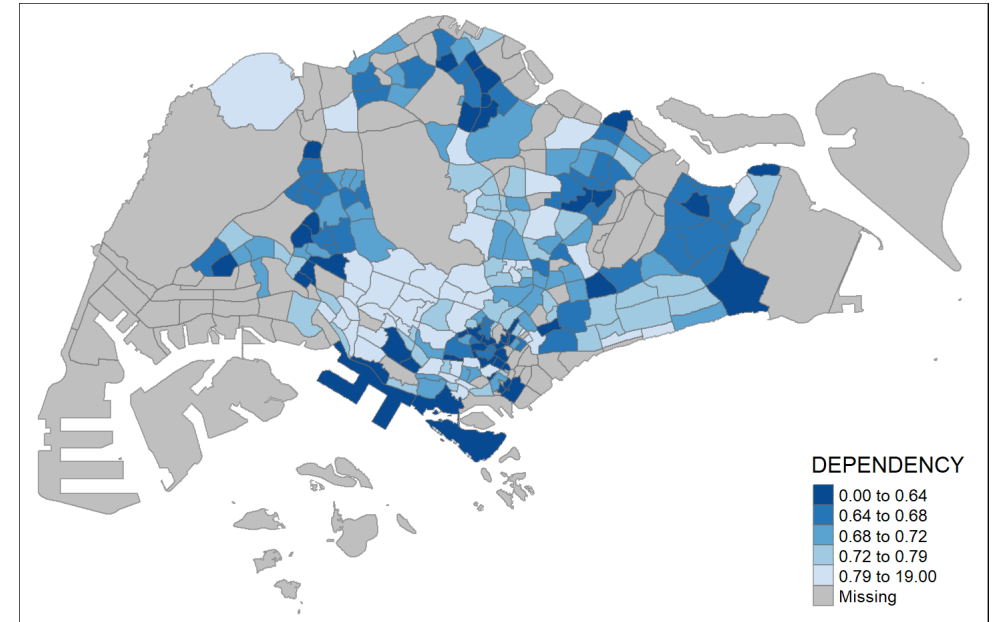
Notice that the choropleth map is shaded in blue.



More about colour

To reverse the colour shading, add a "-" prefix.

Notice that the colour scheme has been reversed.



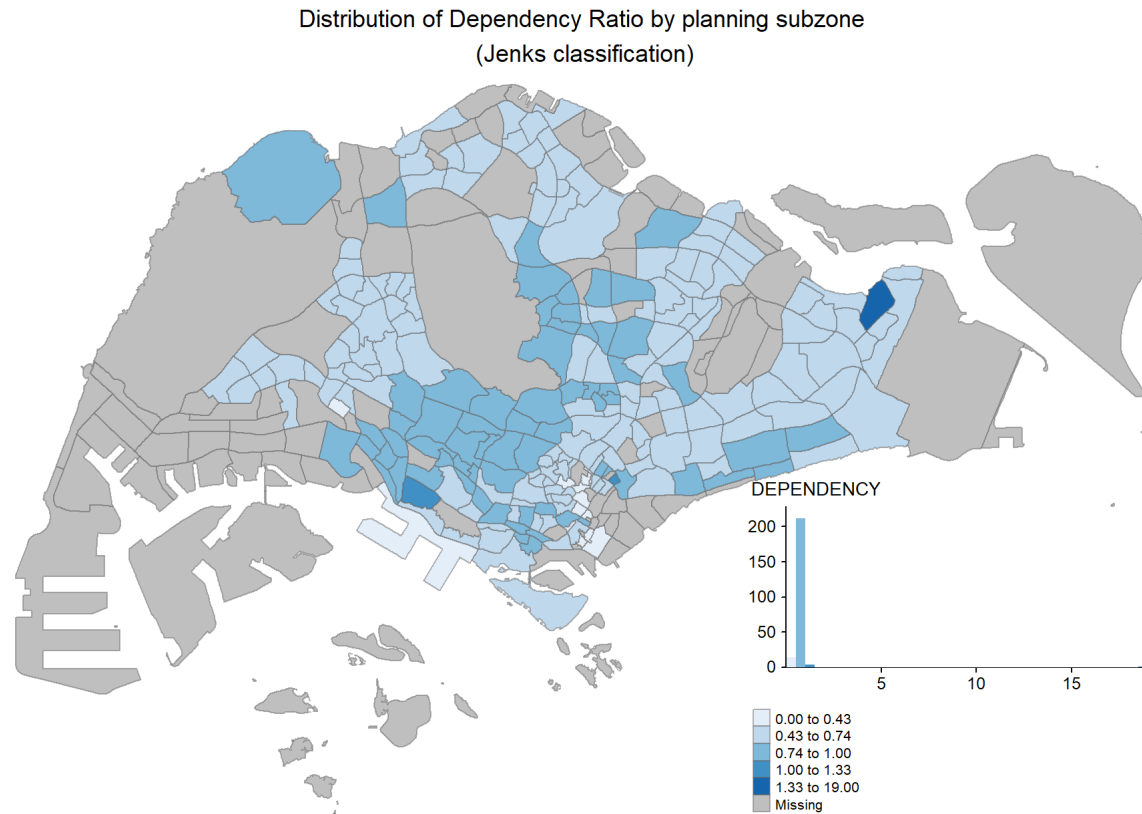
tmap Layouts

Map layout refers to the combination of all map elements into a cohesive map.

- **Map elements** include among others the objects to be mapped, the title, the scale bar, the compass, margins and aspects ratios, while the colour settings and data classification methods covered in the previous section relate to the palette and break-points used to affect how the map looks.

tmap Legend

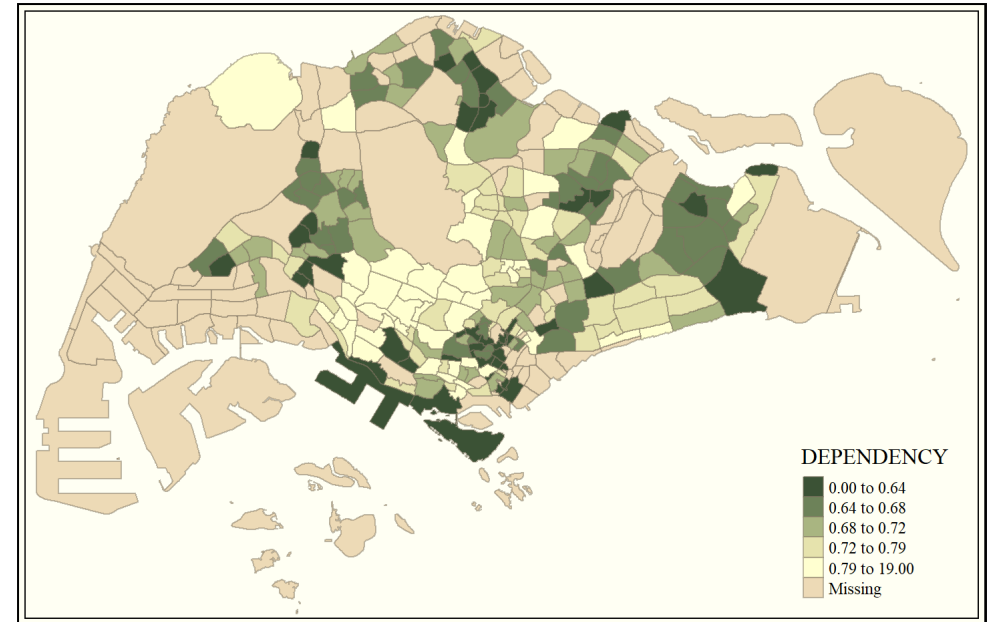
In **tmap**, several *legend* options are provided to change the placement, format and appearance of the legend.



tmap style

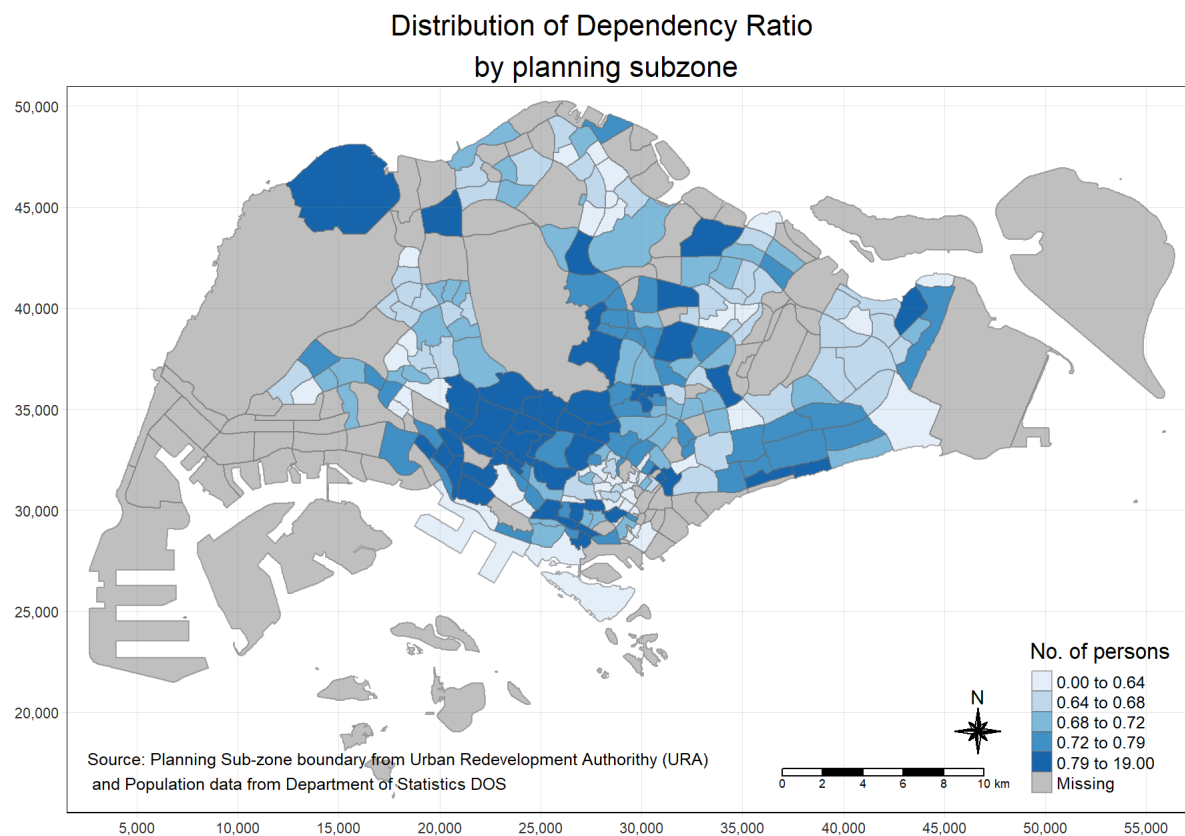
tmap allows a wide variety of layout settings to be changed. They can be called by using *tmap_style()*.

Chloropleth map on the right is plotted by using *classic* style.



Cartographic Furniture

Beside map style, **tmap** also provides arguments to draw other map furniture such as compass, scale bar and grid lines. In the choropleth below, *tm_compass()*, *tm_scale_bar()* and *tm_grid()* are used to add compass, scale bar and grid lines onto the choropleth map.



Drawing Small Multiple Choropleth Maps

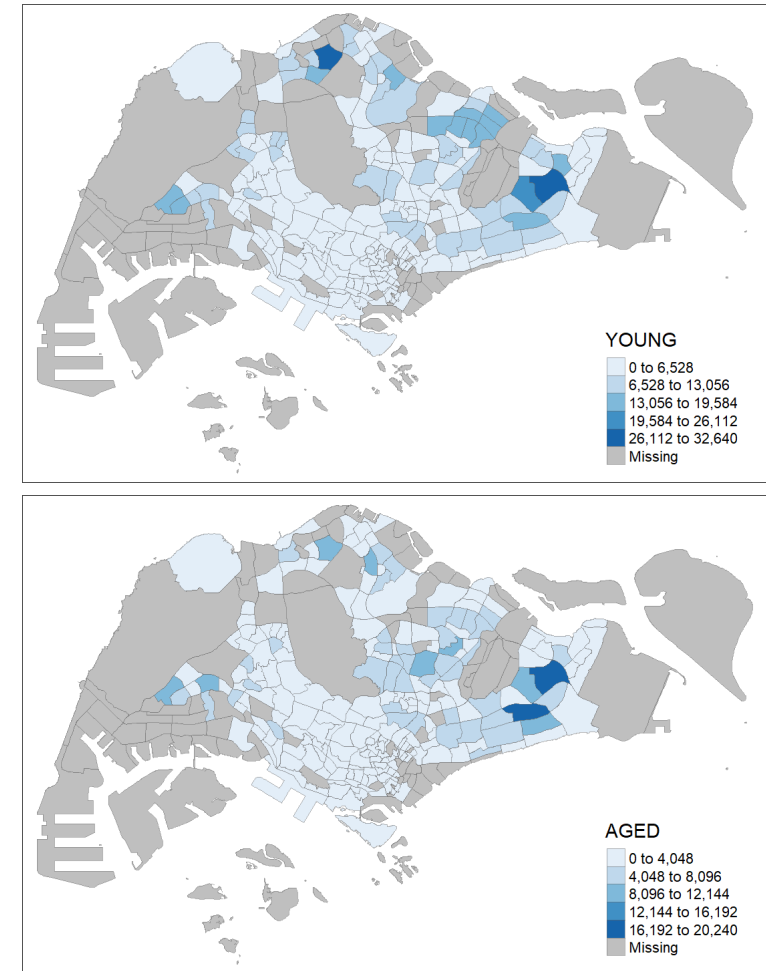
Small multiple maps, also referred to as facet maps, are composed of many maps arranged side-by-side, and sometimes stacked vertically. Small multiple maps enable the visualisation of how spatial relationships change with respect to another variable, such as time.

In **tmap**, small multiple maps can be plotted in three ways:

- by assigning multiple values to at least one of the aesthetic arguments,
- by defining a group-by variable in *tm_facets()*, and
- by creating multiple stand-alone maps with *tmap_arrange()*.

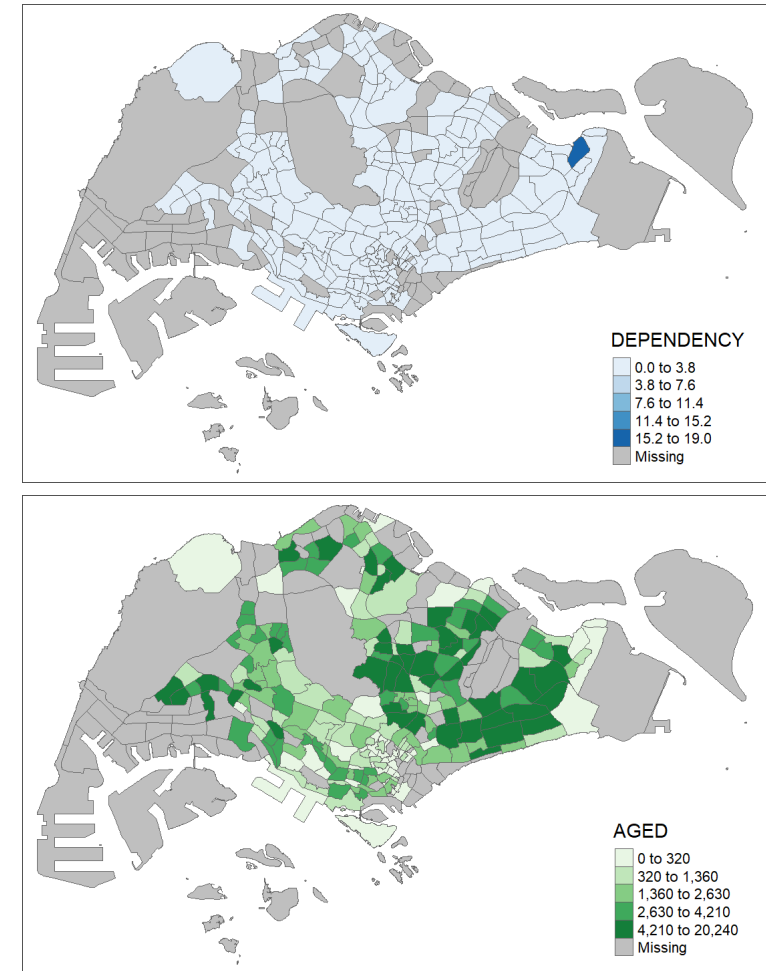
By assigning multiple values to at least one of the aesthetic arguments

In this example, small multiple choropleth maps are created by defining *ncols* in `tm_fill()`



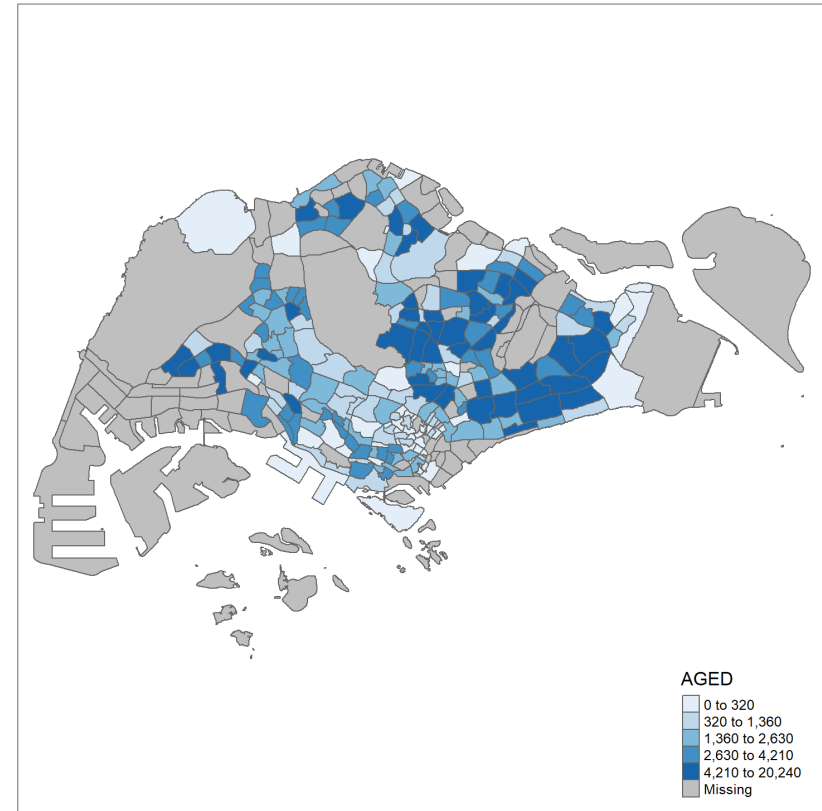
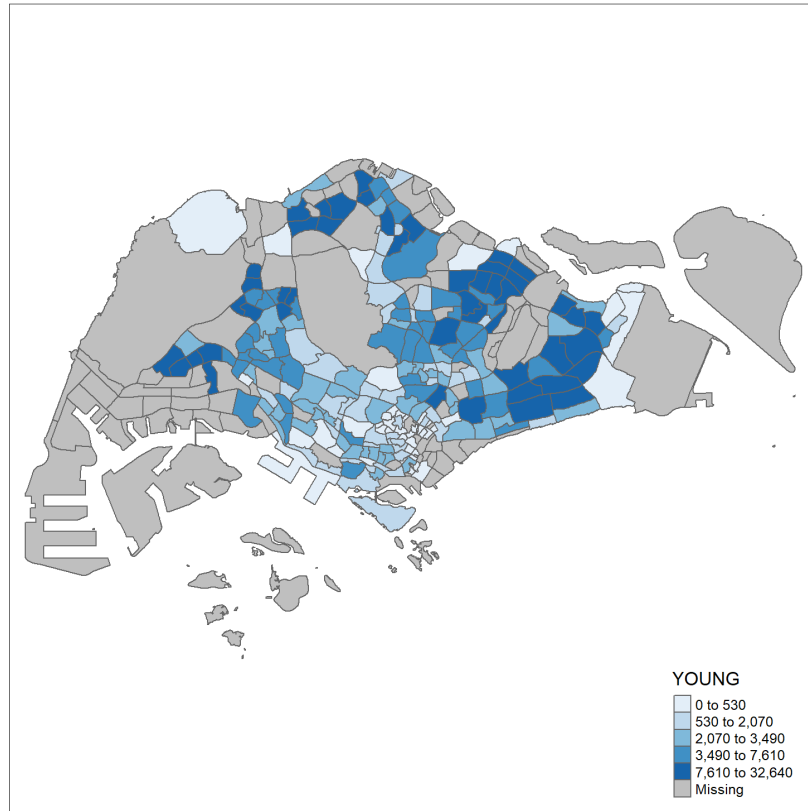
By assigning multiple values to at least one of the aesthetic arguments

In this example, small multiple choropleth maps are created by assigning multiple values to at least one of the aesthetic arguments



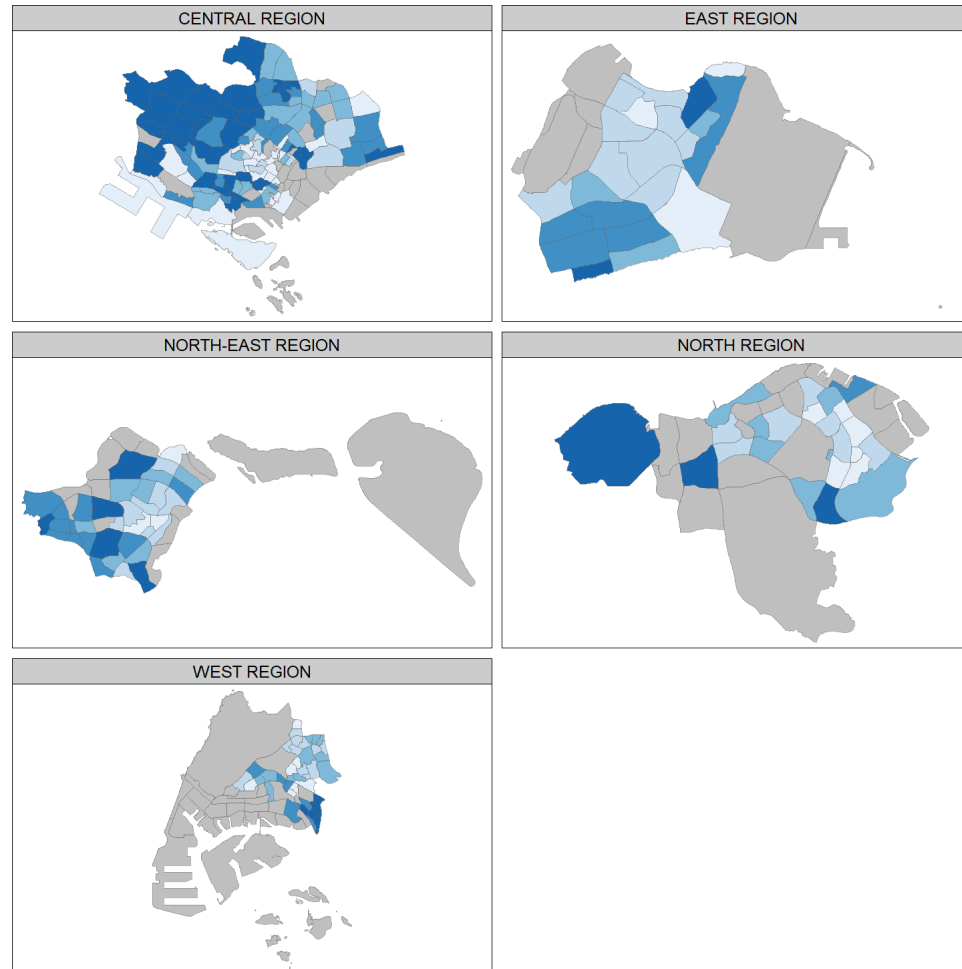
By creating multiple stand-alone maps with `tmap_arrange()`

In this example, multiple small choropleth maps are created by creating multiple stand-alone maps with `tmap_arrange()`.



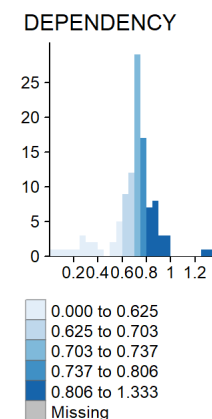
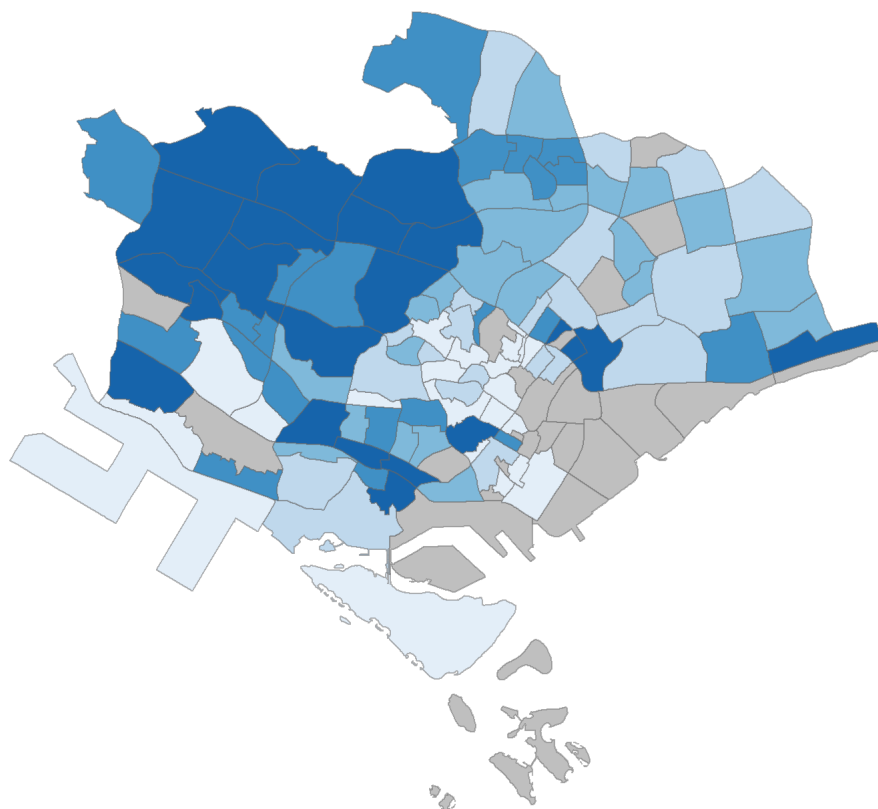
By defining a group-by variable in `tm_facets()`

In this example, multiple small choropleth maps are created by using `tm_facets()`.



Mapping Spatial Object Meeting a Selection Criterion

Instead of creating small multiple choropleth map, you can also use selection function to map spatial objects meeting the selection criterion.



Reference

Principles, Concepts and Methods of Choropleth Maps Design

Core Reading

- Choropleth Maps
- The Basics of Data Classification

Additional Readings

- Choropleth Maps – A Guide to Data Classification
- Bivariate Choropleth
- Value-by-alpha maps
- What to consider when creating choropleth maps
- Choropleth Mapping with Exploratory Data Analysis

References

All About tmap package

- [tmap: Thematic Maps in R](#)
- [Development site](#)
- [tmap User Guide](#)
- [tmap: get started!](#)
- [tmap: changes in version 2.0](#)
- [tmap: creating thematic maps in a flexible way \(useR!2015\)](#)
- [Exploring and presenting maps with tmap \(useR!2017\)](#)