

# Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings

---

Volume 15 *Seoul, South Korea*

Article 47

---

2015

## Dynamic Styling For Thematic Mapping

Simon Moncrieff

*Department of Spatial Sciences, Curtin University*

Elizabeth-Kate Gulland

*Cooperative Research Centre for Spatial Information (CRCSI)*

Follow this and additional works at: <https://scholarworks.umass.edu/foss4g>



Part of the [Geography Commons](#)

---

### Recommended Citation

Moncrieff, Simon and Gulland, Elizabeth-Kate (2015) "Dynamic Styling For Thematic Mapping," *Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings*: Vol. 15 , Article 47.

DOI: <https://doi.org/10.7275/R57942WT>

Available at: <https://scholarworks.umass.edu/foss4g/vol15/iss1/47>

This Paper is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings by an authorized editor of ScholarWorks@UMass Amherst. For more information, please contact [scholarworks@library.umass.edu](mailto:scholarworks@library.umass.edu).

## DYNAMIC STYLING FOR THEMATIC MAPPING

Simon Moncrieff<sup>1</sup> and Elizabeth-Kate Gulland<sup>2</sup>

Department of Spatial Sciences, Curtin University  
Kent Street, Bentley WA 6102, Australia  
and Cooperative Research Centre for Spatial Information (CRCSI)

<sup>1</sup>Email: s.moncrieff@curtin.edu.au

<sup>2</sup>Email: e.gulland@curtin.edu.au

### ABSTRACT

*Current web standards have facilitated the online production and publication of thematic maps as a useful aid to interpretation of spatial data and decision making. Patterns within the raw data can be highlighted with careful styling choices, which can be defined for online maps using tools such as Styled Layer Descriptor (SLD) XML schema. Dynamic generation of maps and map styles extends their use beyond static publication and into exploration of data which may require multiple styles and visualisations for the same set of data.*

*This paper explores the application of thematic styling options to online data, including mapping services such as Open Geospatial Consortium (OGC)-compliant Web Mapping and Web Feature Services. In order to be relevant for both user-specified and automated styling, a prototype online service was developed to explore the generation of styling schema when given data records plus the required output data type and styling parameters. Style choices were applied on-the-fly and to inform the styling characteristics of non-spatial visualisations.*

*A stand-alone web service to produce styling definitions requires a mechanism, such as a RESTful interface, to specify its own capabilities, accept style parameters, and produce schema. The experiments in this paper are an investigation into the requirements and possibilities for such a system. Styles were applied using point and polygon feature data as well as spatially-contextual records (for example, data that includes postal codes or suburb names but no geographical feature definitions). Functionality was demonstrated by accessing it from an online geovisualisation and analysis system. This exploration was carried out as a proof of concept for generation of a map styling web service that could be used to implement automated or manual design choices.*

### 1 INTRODUCTION

In information visualisation, graphical representations are used to explore data and look for meaningful patterns (Mazza, 2009). Thematic maps such as choropleths are a popular visualisation of spatial data (Rautenbach *et al.*, 2012; Yau, 2011). Whilst choropleths, which represent an attribute value spatially by shading regions to show the value's relative intensity, are often published in (potentially static) reports to support decision-making, the interactivity enabled by a slippy map client presents an opportunity to extend thematic mapping to a more dynamic representation of spatial information. In both static and dynamic visualisations, choices made about how to present information visually - the styling - influence the effectiveness of the graphic for communicating information.

This paper explores dynamic styling of data to produce thematic maps as images. Styling parameters are processed on-the-fly, a process that is suitable for online spatial visualisation and analysis systems, as seen for example in Moncrieff *et al.* (2014) and Moore *et al.* (2013), as well as online mapping applications. Different visualisation styles are tested over data records in order to explore possible capabilities to build into an independent dynamic styling agent.

Thematic styling affects interpretation by specifying how to display spatial data, including features such as colour, border style and rules for different map scales. Standard formats such as the Open Geospatial Consortium (OGC) Styled Layer Descriptor (SLD)

(OGC, 2007), part of the OGC Web Mapping Service (WMS) standard, can specify symbols according to attribute values to produce thematic maps from different data sources. A variety of open source desktop applications and servers, including QGIS, GeoServer<sup>1</sup> and MapServer, implement SLD or an alternative styling schema format. Mapping services that comply with standards such as WMS or OGC Web Feature Services (WFS), whilst affording communication across processes and machines, have a limited ability to automatically adapt styling to best represent their contents. For example, GeoServer combines an SLD file with a WMS layer's data to produce and present a map image using pre-set styles, or it can provide WFS layer data to be displayed in flexibly by processing styling instructions on a client machine. Some orchestrated services use a script or service to produce more dynamic style definitions but these are specialised instructions designed for specific circumstances (Rautenbach *et al.*, 2012). Difficulties faced in the development of a more general dynamic styler include the wide variety of choices and the at times complex relationship between style and data. In this paper, choices that would be useful for a styling service - including possibilities for combining dynamic and server-side styling - are investigated from the perspective of an online spatial visual analytics engine.

In the proposed method, valid style instructions can be produced for a variety of spatial data and visualisations. A styling request must specify the data to be mapped as lists or vectors of text-based records and can also specify styling options including colour schema and the method for classifying data features. The fundamental styling was then extended to explore additional methods for styling and presenting information and data visually. This caters for experimentation with different styles over a set of data in order to develop a more informative map. It also allows styling decisions to be made separately from the specific data records or processes they are being applied to. Styling information is used to produce visualisation images in a similar manner to a WMS layer, but with more flexibility for styling options and no need for manual specification of complex styling schema syntax. Where a style parameter is not specified, default values can be used depending upon other input parameters. For example, a diverging colour scheme such as red-to-blue could be used as a default for a data style to highlight values above and below a pivot value (such as mean temperature).

In this paper, section 2 discusses map styling choices and previous research into styling in automatic mapping systems. In section 3, the design of the new styling service is described along with a case study implementation. Results from testing the case study are discussed in section 4, and final conclusions are outlined in section 5.

## 2 BACKGROUND

The way in which data is presented has a profound effect on the way it is interpreted. Although this paper focusses on styling possibilities for digital thematic maps, display options are equally important in printed maps and non-spatial data visualisations.

Many elements are needed to create a meaningful map. In a thematic map, design decisions about the type of map, such as choropleth or point symbols; classification method and number of classes; and colour scheme influence perceptions about its message and purpose. Mapping applications make it easier to create maps but do not offer guidance on choices such as colour scheme, which can result in confusing maps (Harrower and Brewer, 2003). Design issues such as these affect automated map production as much as manually created maps. Possibilities for an automated system include predefining styles for each (or

---

<sup>1</sup> GeoServer <http://geoserver.org/about/>

all) datasets on offer or applying manual or calculated style options to the map data as required.

## 2.1 Styles in Mapping Systems

Thematic styling definitions can be used directly to transform data to meet specific interpretation needs. Take, for example, *rendering transformation* functions in GeoServer. A function can be embedded in a SLD file so that raw spatial data is transformed and then styled to give a meaningful, visual result. An example is `gs:Contour`, which can restyle a raster image as a contour map with specified contour line values.

Thematic styling files have been applied to multi-part systems that coordinate several spatial services. One example is a multipublishing environment (Kettunen *et al.*, 2012), which uses style rules to cater for different types of display devices, such as touch screen, mobile phone or print map. However, as with many online systems, the SLD files used are predefined and static.

Another application of thematic styling is on mobile mapping devices. In Weiser and Zipf (2007), a tool was developed to produce SLD for an OGC Web Map Service (WMS) compliant output. Styling information was based on their original maps, produced on proprietary desktop software (Esri ArcMap). A user can change styling options via the tool and the resulting SLD information is saved as an external file. This file could then be accessed by a WMS client or used in a local application on a mobile device.

The importance of easily-interpretable visuals is paramount in spatial decision support systems (SDSS). People making high-level decisions about issues, such as service accessibility, need access to images that correctly highlight the current state of a region. Visualisation decisions such as using a network distance rather than a straight-line distance can make an enormous difference to the practicality of results. Burdziej (2012) incorporated dynamic SLD production into an SDSS for accessibility information, producing an SLD on demand by a PHP script. This enhanced the system by creating maps suited to the users and avoided the necessity of pre-storing an unreasonably large number of SLD files to allow for the wide range of options.

Iosifescu-Enescu *et al.* (2010) developed a 'Map and Diagram Service' that extended WMS and SLD standards and incorporated style information to produce a map image on-the-fly via a 'getMap' operation. It also defined a 'getStyle' operation to retrieve styling information about a layer in a map visualisation. They found that styling formats in existing OGC standards were suited for topographic maps, but less appropriate for thematic maps. They added a new 'Diagram Symbolizer' to the Symbol Encoding part of the SLD standard to allow for symbols such as pie charts and classification schemes like 'quantile'. The service was tested within an SDSS based on sensor readings. The same team also showed that styling information to be sent to a mapping service could be updated by users via graphic user interfaces (Iosifescu *et al.*, 2013).

## 3 APPROACH

A wide variety of styling options was trialled to test the visualisation possibilities for a dynamic styling service. In particular, options for attributes at point locations or within regions, including classification schemes and buffer overlays, were examined.

A number of factors need to be considered in order to generate style definitions in a flexible manner. A styling web service needs to respond to a number of different input types for spatial and other data content. It also must be able to adapt to different ways of interpreting the input data vectors, including a variety of spatial analysis algorithms. Finally,

it must be able to return output and styles in a specified format, such as SLD, WFS and/or WMS. For usability of the experimental styling service, it is also important that the features it supports - including available styling methods and output formats - are discoverable. This could be achieved through parameters to a RESTful interface, such as a GetCapabilities request, as defined in the OGC Web Processing Service (WPS) standard for returning the service's capabilities.

### 3.1 Architecture

A proof of concept web service was developed using the Django<sup>2</sup> web framework, which is suited to generating the output of such a web service (see Figure 1). The web service employs established map classification techniques, implemented through APIs provided by the PySAL<sup>3</sup> spatial analysis python library. It applies thematic colour schemes identified by ColorBrewer<sup>4</sup> and derived using Matplotlib<sup>5</sup> colour schemes to encapsulate a higher number of colour gradients within each colour space. As a proof of concept, the style descriptors were encapsulated within Mapnik<sup>6</sup>, although the principles in a dynamic styler design can be applied equally to alternative style schema formats such as SLD.

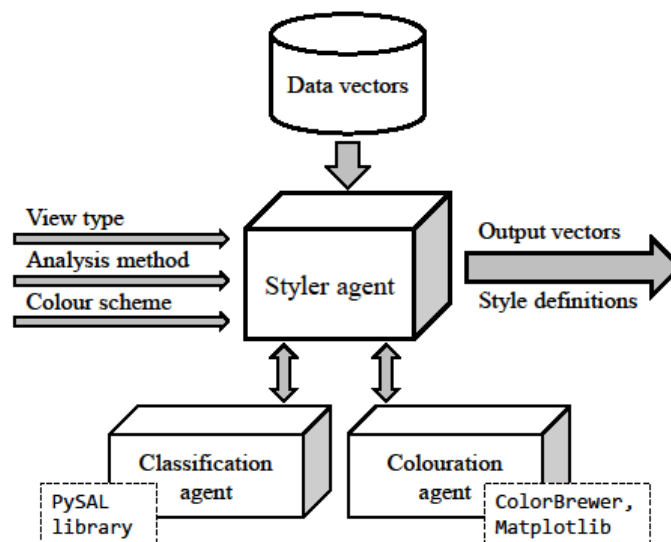


Figure 1: Process of dynamic styling web service

### 3.2 System inputs

System inputs include the data to style, either using a text based format or a link to the data to be styled (for example, a WFS dataset), along with any required styling options. A final parameter for the service is the output format required, for example an image or a WFS incorporating a colour attribute. In the case of an image output, functionality enabling a style descriptor input was also included. Separate agents were designed to interpret map classification analysis algorithms and colour schemes to be called upon from the styler.

<sup>2</sup> Django <https://www.djangoproject.com/>

<sup>3</sup> PySAL Python library <https://pysal.readthedocs.org/en/latest/>

<sup>4</sup> ColorBrewer <http://colorbrewer2.org/>

<sup>5</sup> Matplotlib <http://matplotlib.org/>

<sup>6</sup> Mapnik <http://mapnik.org/>

### 3.2.1 Data inputs

The styler accepts geometry vectors of point or polygon features with attributes, or a list of records, each containing an attribute list of key-value pairs. While styling parameters can be specified through a web service REST API, incorporating meta-data directly into the input data was also explored.

### 3.2.2 Style parameters

Up to three styling parameters can be specified to determine the generation of the thematic map: 1) analysis method; 2) view type; and 3) colour scheme. Default values are assigned for style choices, depending upon the type of input vectors.

The *analysis method* available depends upon the input data vector. First, the attribute(s) to map must be specified. In implementing the polygon region mapping algorithm, only univariate data is considered with respect to styling. However, point data can be visualised in multiple dimensions, as discussed in section 3.3. Other options include size of feature buffers (if any) and the method for classifying features. Once data classes have been assigned, the colour scheme chosen is also important for conveying meaning.

The *view type* relates to symbolisation changes as the map window changes, such as zooming in or out. The options are global, local, and static. Global applies a style across the entire dataset; if the map view is focussed on a small subregion, it will still use the global values to define the range of the colour bins used. In a local view, symbols will be adjusted to the range of values in currently visible features within a map extent, or within a given bounding box search area. Static mode is an extension of local; a local view will be applied to an initial map extent and this styling can then be maintained during any subsequent changes to the extent, for example if zoom or pan actions are used within a slippy map. This method is appropriate for analysis of regions such as a metropolitan area in which a large number of geographic regions may be defined, for example city blocks. See Figures 2 and 3 for a comparison between local and global styling methods.

*Colour scheme* can be selected from available schemes as described in Section 3.1.

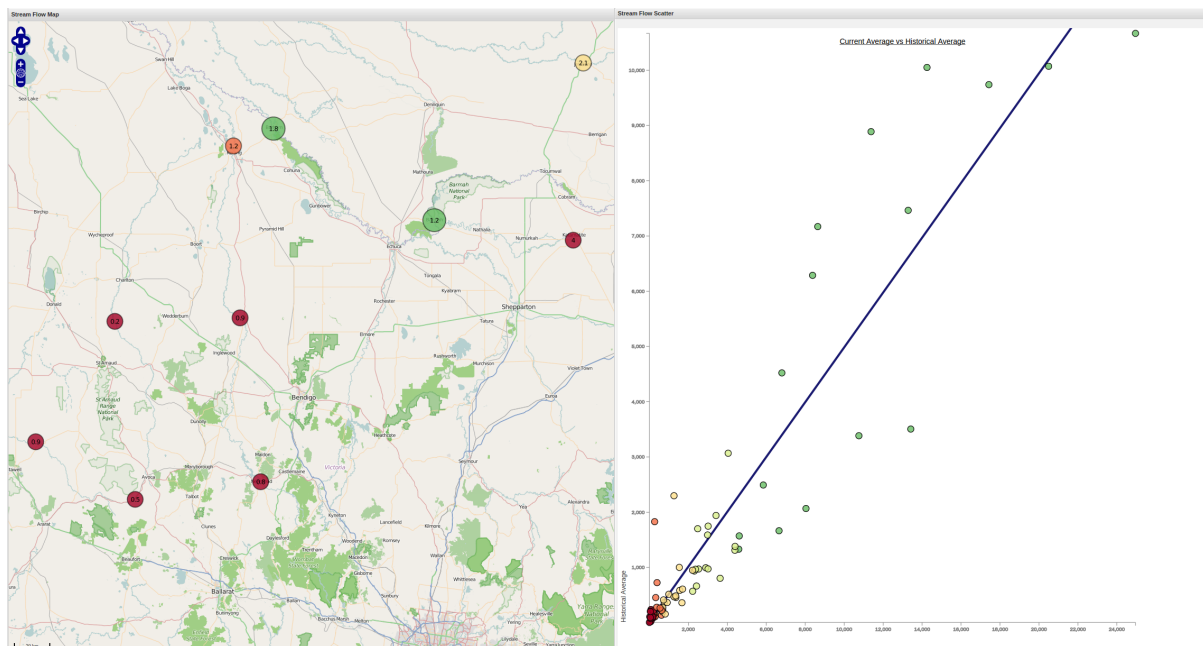
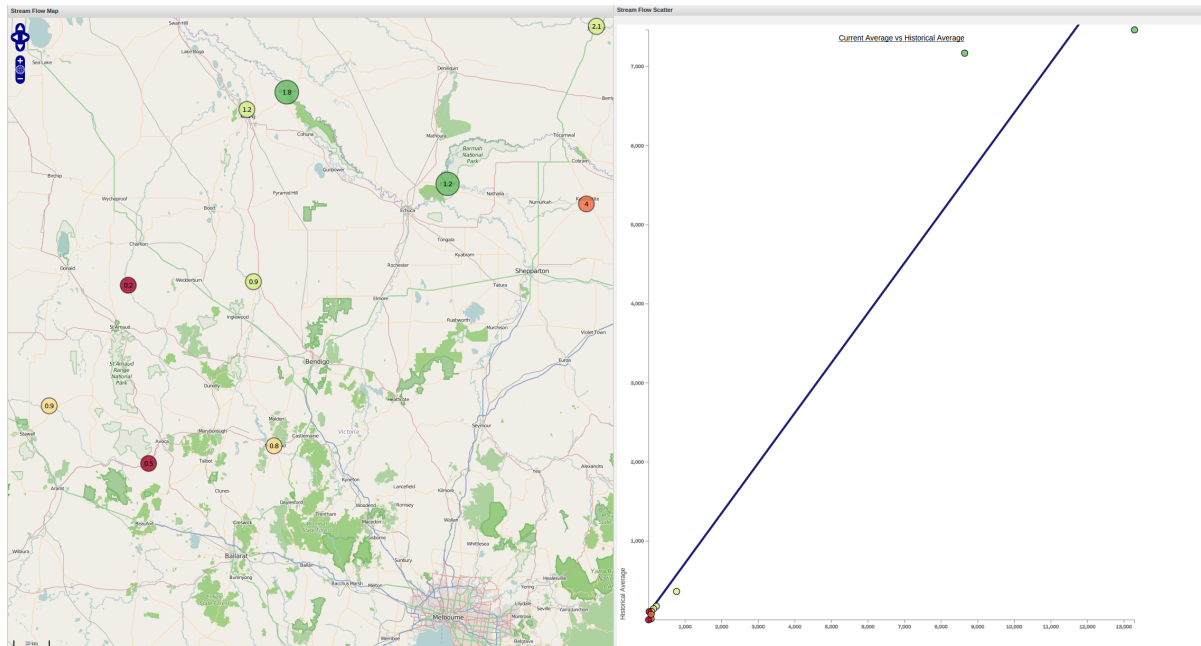


Figure 2: Legend Extent: Map classification by global extent (entire dataset).



**Figure 3: Legend Extent: Map classification by local extent (dataset represented in map extent).**

### 3.3 Thematic Mapping

To visually represent thematic data, two types of information were considered: spatial data, and spatially contextual data. The latter is data that has a spatial attribute imposed on the data, such as population counts by postcode. In this example, a postcode can be linked to a spatial region for calculation and display of population density. The styler can produce styles for region data, point data and non-spatial data. It is possible to show more than a single dimension in a thematic map; for regions, represented by polygons, this can be achieved using a bivariate choropleth and for point data, it is possible to show multiple attributes for a single point visualisation by manipulating variables such as the  $x$  and  $y$  radii of the point.

#### 3.3.1 Regions

To convert a data input vector into a choropleth map, a number of factors were considered including: how should the data be classified for display? What colours should be used?

Inputs to the data styler include the number of classification bins to separate data into, and the map classification scheme technique to use. For example, an input of 10 bins and a quantile scheme would produce deciles. More specifically, the styler produces a set of values representing the decile class for each polygon and styling information defining how to display it as a choropleth map.

A number of PySAL map classifications were integrated into the service including Jenks natural breaks, quantiles, and standard deviations. The service uses thematic colour schemes identified by ColorBrewer, and derived using Matplotlib colour schemes. The PySAL library includes a wide range of spatial analysis methods which can be used to extend the capabilities of the styler in the future.

#### 3.3.2 Points

Three methods of styling point data were examined: a thematic visual representation, a contextual representation, and a relative size point representation. A number of parameters can be used to alter the presentation of point data. For example, the attribute(s) used to

determine the radius (both in the x and y directions) can be specified along with the presence, or absence, of a label and the content of the label, and the attribute used to determine the fill colour of the point. For the first two methods the point size is determined using the minimum and maximum values of the attribute feature space to scale the radius of each point from a minimum set radius to a maximum radius. The visual representation emphasises the attribute over which the thematic styling was determined, by default, the radius is determined using the same attribute ( $x=y$ ). The contextual representation uses multi-variate data to incorporate further context into the thematic map, in this case, a label and the radius (x, y) are included, and can correspond to different variables. The relative size point representation incorporates a radius determined in metres, providing a spatial context.

Once the colour space and filter bins have been determined, layer styling options that can be applied to the thematic visualisation include the colour of the boundary of the polygons, and the layer alpha. To define styles for point data, PySAL, ColorBrewer and Matplotlib are used to select appropriate colour schemes in a similar manner to that described for polygon data. Circle sizes of individual point symbols can also be scaled according to the range (minimum to maximum) of the attribute of interest in the input dataset.

Multivariate mapping extends thematic mapping techniques to two or more attributes and also has a range of options. For instance, using scaled ellipses as described earlier to compare two attributes. Style information on a set of points in Mapnik can be displayed by altering colour, radius x, radius y and contextual information such as text. The styler can reinforce a message by associating the same attribute with symbol colour and size, or assist in comparisons by using different attributes for each display variable. Adding complexity to the symbology increases the amount of information that can be portrayed, but at a potential increased cognitive cost to the user who needs to interpret it (Mazza, 2009).

### 3.4 Extended Styling

To increase the flexibility of the styling service, a number of different methods were explored to extend the base styling, detailed in Section 3.2.2. While these methods will be detailed in Section 4, two such extensions are discussed in detail in this section.

#### 3.4.1 Meta-layers

Meta layers are layers of ancillary data that can be viewed concurrently to the thematic map to provide context. Consequently, maps that incorporate meta-layers can also accept more than one set of data as input. For an example of a meta-layer, consider public transportation coverage data (how many households are within walking distance of each bus stop) that may use a point vector to represent bus stops and their coverage values. But a meta-layer used to calculate this result - in this case house locations - can also be included to add meaning to the final result. The styler can combine these two sets of data inputs into a single map output.

#### 3.4.2 Beyond spatial

Thematic styles can also be applied to non-spatial visualisations. For example, a plot can be styled so that points on the plot are colour-matched to regions shown on a map, with the region name or postcode acting as a common property. Further, cartographic styling techniques can be used to thematically present non-spatial data. Colour schemes generated in this manner, for both spatial and non-spatial data, can be extended to be applied to other visualisations, such as scatter plots. The method can also be applied to visualisations of multivariate data, to highlight a relationship between the styling attribute and the remaining attributes; examples of such visualisations include a scatter plot matrix and parallel



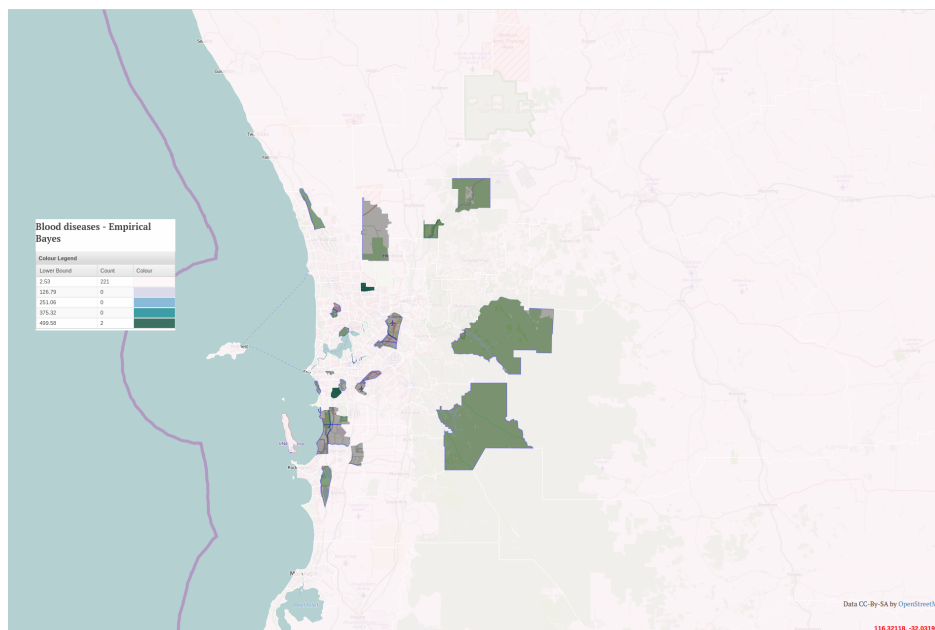
coordinate plots. This method is implemented by appending a colour attribute to a given feature space, for example incorporating a colour attribute into a JSON array, with D3<sup>7</sup>.

## 4 EXPERIMENTAL RESULTS

Styling options were tested on spatial data from a number of sources, to examine the extent of styling available through the service. The results in this section represent a subset of the styling methods that were implemented and investigated, and are presented to demonstrate the differences in interpretability between applying different styling methods to different data layers. All base layer map images are sourced from Open Street Maps<sup>8</sup>.

### 4.1 Analysis Method

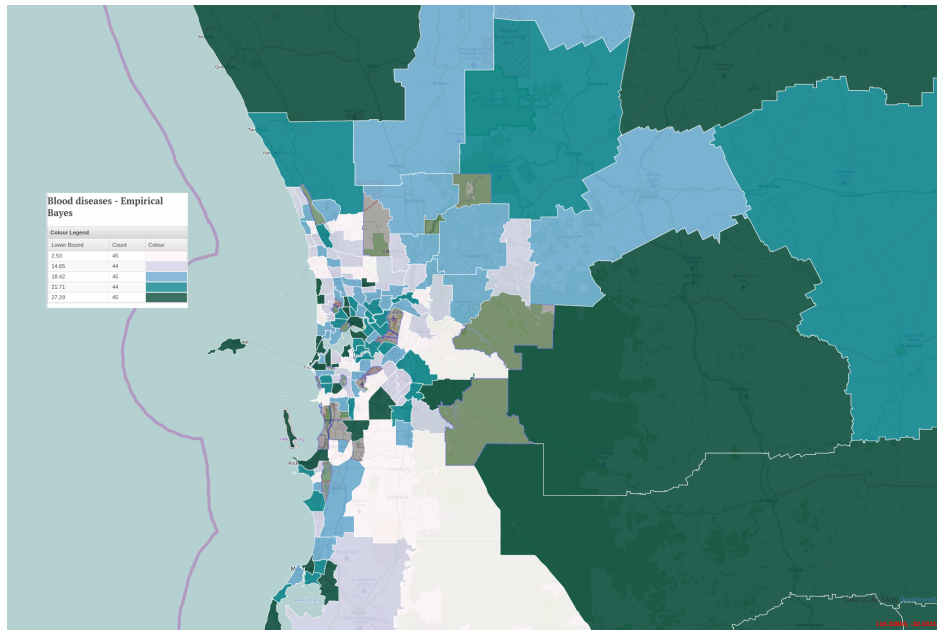
As well as the view types, as shown in Figures 2 and 3, different map classification algorithms were also explored, in particular, the efficacy of different approaches with respect to the underlying data being visualised and the subsequent impact on the interpretation of the data. Figures 4 and 5 show the same data, with the same number of bins (five), but styled differently; equal intervals was applied in Figure 4, while Figure 5 was generated using quantiles. Both maps show the smoothed rate of blood borne diseases derived from synthetic data designed to realistically simulate hospitalisations for Western Australia, with greyed out areas representing no results for the area. These figures demonstrate the importance of the map classification to the interpretation of the data; in this case, an outlier skews the map classification in the case of Figure 4, limiting a visual interpretation of the underlying disease rate.



**Figure 4: Map classification of blood borne diseases using equal interval, with 5 intervals.**

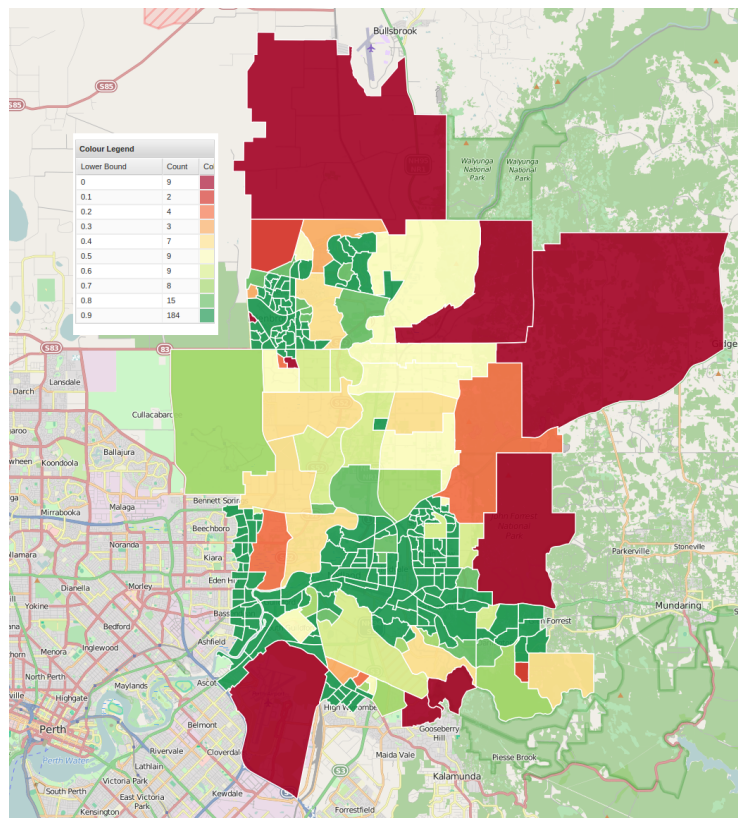
<sup>7</sup> Data driven documents <http://d3js.org>

<sup>8</sup> OpenStreetMap <http://www.openstreetmap.org/copyright>



**Figure 5: Map classification of blood borne diseases using quintiles; that is, quintiles with 5 intervals.**

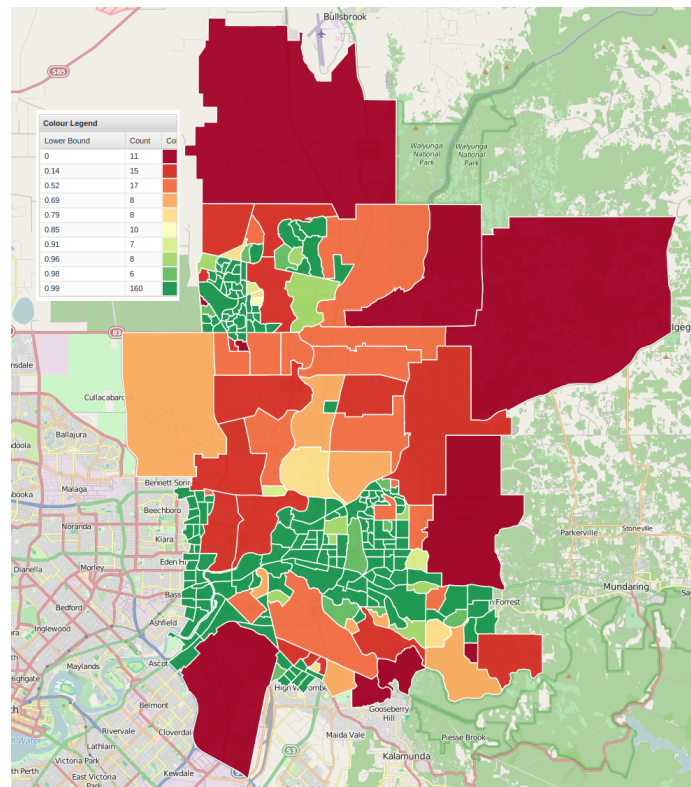
However, there are situations where equal intervals map classification is suitable. Figure 6 displays a probability calculated for each geographic feature displayed, using equal intervals with 10 intervals has the effect of partitioning the probability space represented into deciles. Additionally, the legend represents a coarse histogram of the distribution of the probabilities.



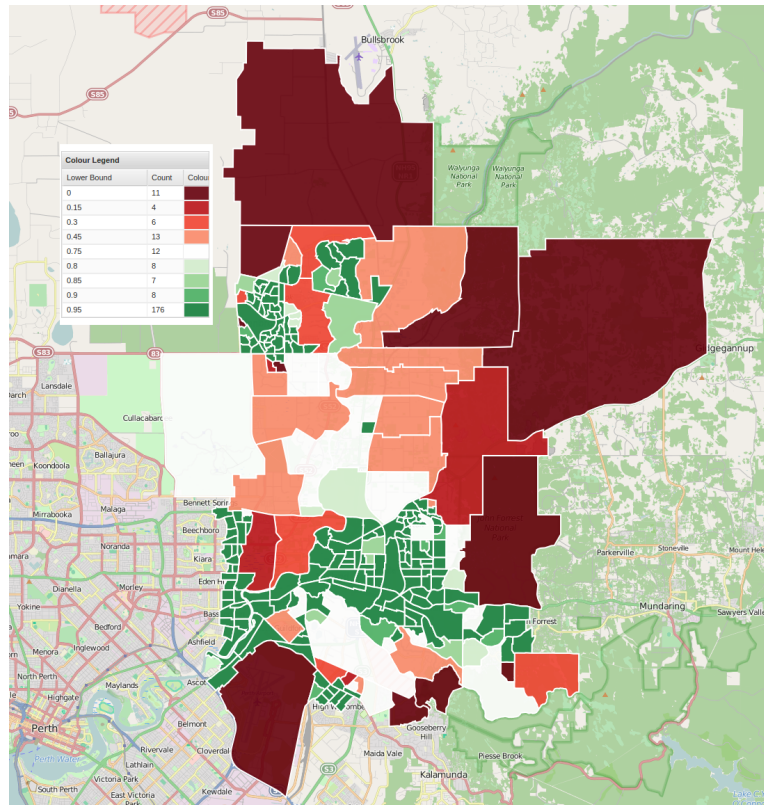
**Figure 6: Map classification of a data set with a range from 0 to 1.**

## 4.2 Colour scheme

Figures 7 and 8 represent the application of two different colour schemes to the same set of data. Figure 7 uses a red, yellow, green colour scheme, delimited using natural breaks. In contrast, Figure 8 uses a diverging scale around a pivot point for the same data set. In this case, 0.75 was selected as a significant value and therefore applied as the pivot point. Consequently the thematic map highlights the values above and below 0.75. This example shows why the method used to present the data should be tied to the purpose of the resulting visualisation.



**Figure 7: Red-green colour scheme over natural breaks.**



**Figure 8: Diverging colour scheme around a pivot point value of 0.75.**

### 4.3 Point data

Figures 9 and 10 show examples of thematic visual styling and contextual representation respectively for data set comprising multivariate point data, in this case the long and short term average of time series data, and the proportion of the two. Quintiles map classification was used in both cases, along with a red, yellow, green colour scheme. Figure 9 shows solely the proportion of the short term to long term averages. Figure 10 shows the same information, with the proportion being included as the label, and the radius determined by the short term average. While a static radius was used for both figures, the minimum radius can be tied to zoom level to dynamically adjust radii of points as a user zooms in.

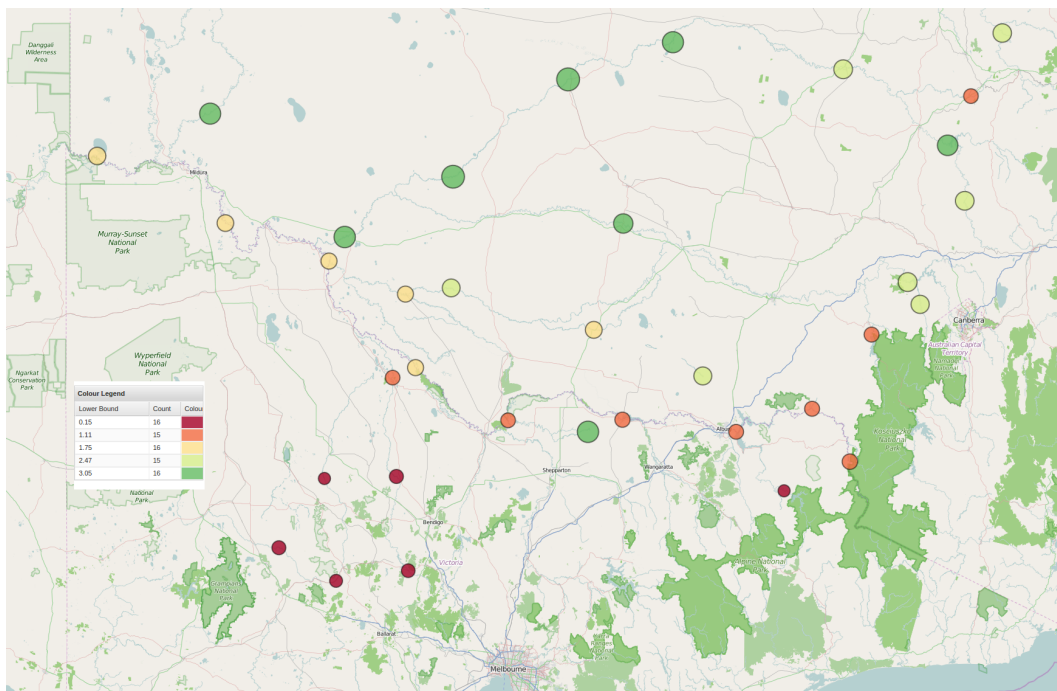


Figure 9: Thematic visual styling: proportion of short to long term averages.

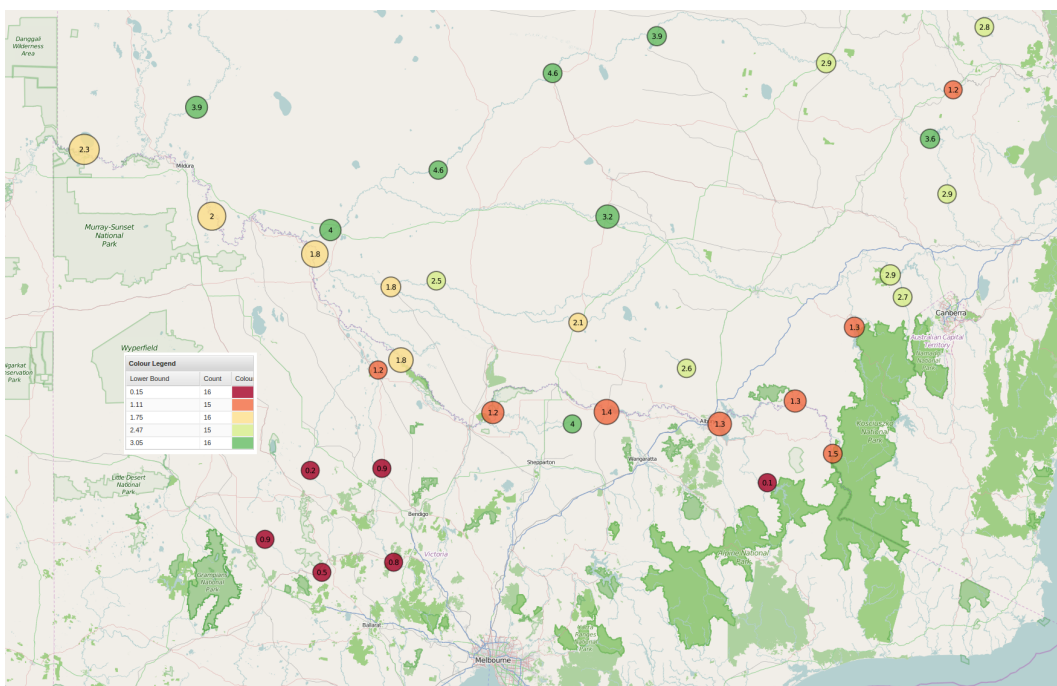


Figure 10: Contextual representation: labelled proportion of short to long term averages.

The styling of points was extended to incorporate a relative size on the map for a point by using a buffer size to determine the radius. For example, Figure 11 shows the results of displaying a set of points, corresponding to a public service, with a 500m buffer as the point radius that can be used to approximate coverage for each service. Due to the large overlap of points visualised in this manner, a high transparency was applied to ensure overlapping points remained visible.

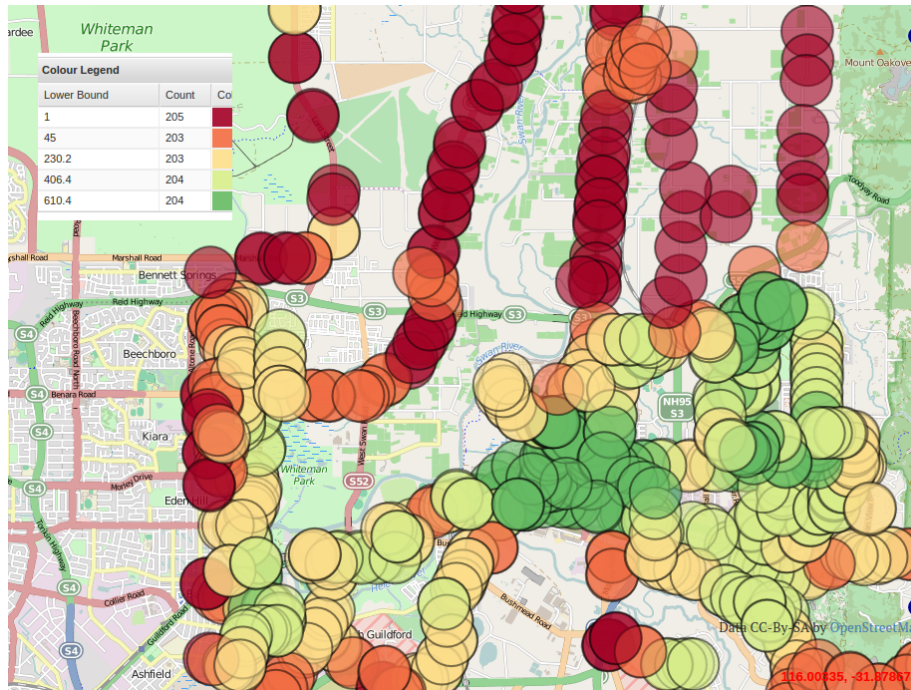


Figure 11: Point data plotted using a 500m radius.

#### 4.4 Non-spatial visualisations

Figure 12 displays an example of applying cartographic styling techniques to non-spatial visualisations. This method inserts the colour scheme into a JSON based spatial format, which can then be interpreted by JavaScript visualisation libraries such as D3 to produce a visualisation combining the colour scheme and a visualisation technique with the aim of embedding the map classification colour scheme within the visualisation.

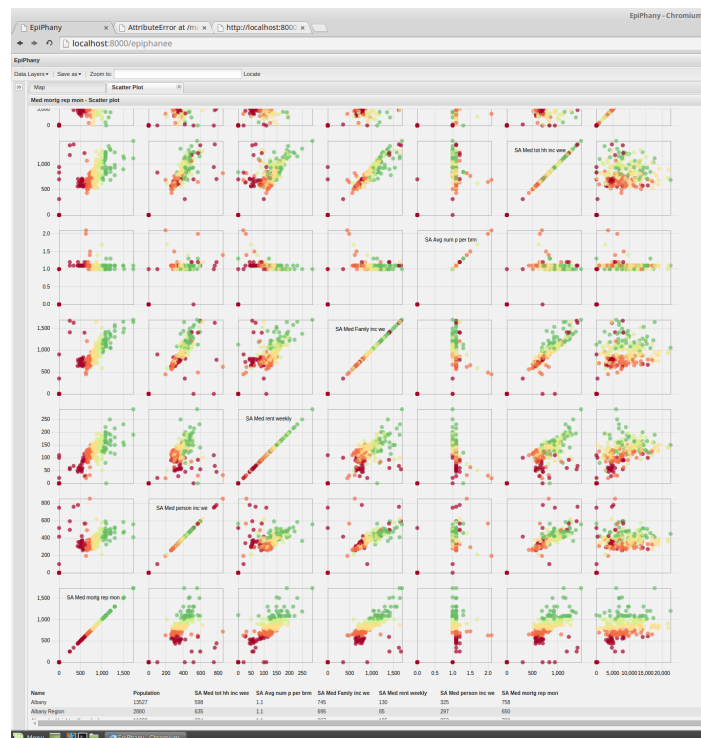


Figure 12: Scatter plot matrix generated using cartographic styling for the plot colours.

## 4.5 Meta-layer

This method refers to incorporating an ancillary layer within a data layer, with both layers being viewed simultaneously so that the meta-layer can provide context. In general, the meta-layer should correspond to underlying data used in the derivation of the data layer. Consequently, when shown with minimal styling, the spatial relationship between the initial data and the subsequently derived data can be seen. Figure 13 shows a grid data visualisation with a bespoke defined style applied that defines a class for each grid; in this case, the classes were derived from the trend associated with each grid. For this dataset, the trend was derived by determining a combined trend associated with points contained within each grid. Consequently, this data is shown as a meta-layer to provide context.

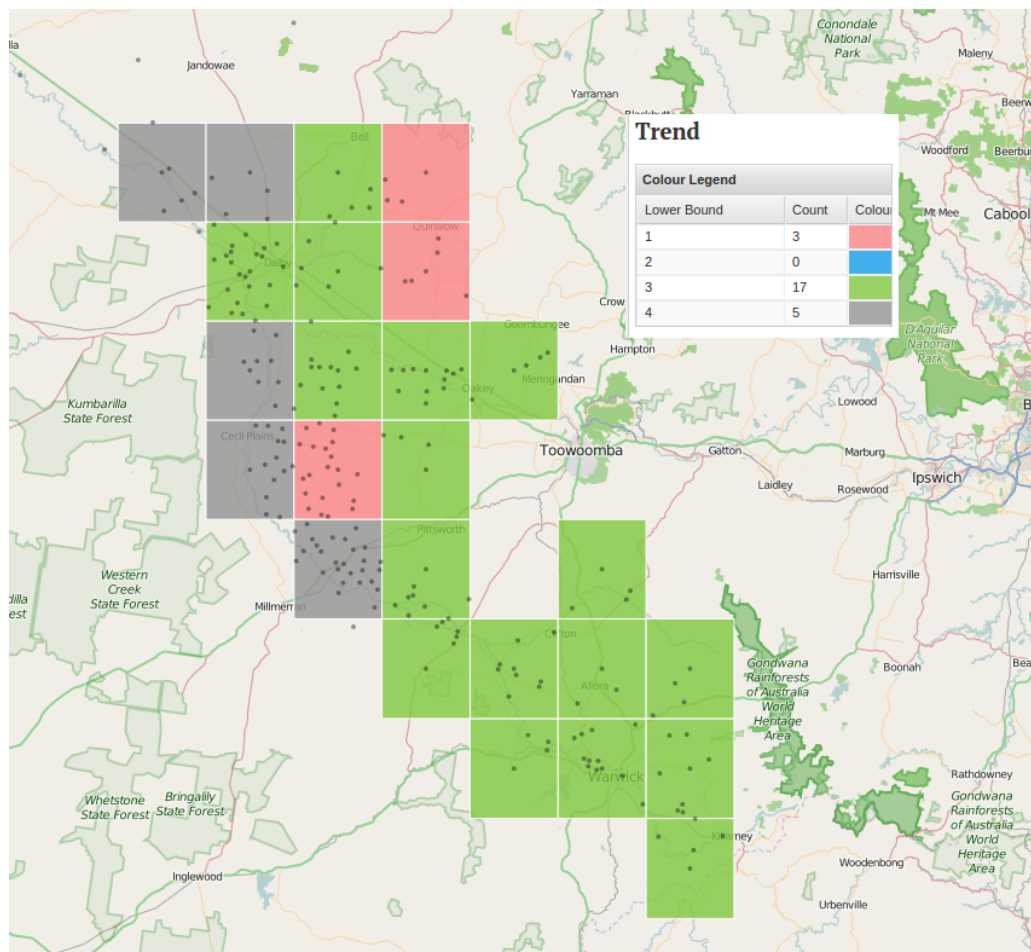


Figure 13: Point data plotted using a 500m radius.

## 5 CONCLUSIONS

No single visualisation can be considered the "most suitable" for all thematic mapping problems. Mapping choices should therefore be made on-the-fly, based on the data itself, rather than predefined and static. This case study has demonstrated how the use of open source tools can be coordinated to produce web map styling information on-the-fly. This will enhance the scope of automated spatial analysis and visualisation systems as demonstrated on the case study using Mapnik. Currently, only point and polygon geometry features can be interpreted by the dynamic styler. However, methods to extend dynamic styling to line visualisations, such as for road or waterway data, will be explored in future work.

## 6 ACKNOWLEDGEMENTS

The Cooperative Research Centre for Spatial Information has supported this work, whose activities were funded by the Australian Commonwealth Cooperative Research Centres Programme.

## 7. REFERENCES

- Burdziej, J., 2012. A web-based spatial decision support system for accessibility analysis - concepts and methods. *Applied Geomatics*, 4(2):75-84.
- Harrower, M. and Brewer, C. A., 2003. Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27-37.
- Iosifescu, I., Iosifescu, C. M., Panchaud, N., Eichenberger, R., Sieber, R., and Hurni, L., 2013. Advances in web service-driven cartography. *International Cartographic Conference*, Dresden Germany.
- Iosifescu-Enescu, I., Hugentobler, M., and Hurni, L., 2010. Web cartography with open standards - a solution to cartographic challenges of environmental management. *Environmental Modelling and Software*, 25(9):988-999.
- Kettunen, P., Sarjakoski, L. T., Ylirisku, S., and Sarjakoski, T., 2012. Web map design for a multipublishing environment based on open APIs. In Peterson, M. P., editor, *Online Maps with APIs and Web Services*, chapter 12:177-193. Springer, Berlin Heidelberg.
- Mazza, R., 2009. *Introduction to Information Visualization*. Springer, London.
- Moncrieff, S., West, G. A. W., Cosford, J., Mullan, N., and Jardine, A., 2014. An open source, server-side framework for analytical web mapping and its application to health. *International Journal of Digital Earth*, 7(4).
- Moore, A., de Oliveira, M., Caminha, C., Furtado, V., Basso, V., and Ayres, L., 2013. Applying geovisual analytics to volunteered crime data. In Moore, A. and Drecki, I., editors, *Lecture Notes in Geoinformation and Cartography: Geospatial Visualisation*, chapter 9,185-209. Springer-Verlag, Berlin.
- Moreland, K., 2009. Diverging color maps for scientific visualization. In *5th International Symposium on Advances in Visual Computing (ISVC '09)*, II:92-103, Las Vegas, USA.
- OGC, 2007. *OpenGIS Styled Layer Descriptor Profile of the Web Map Service Implementation Specification*. Technical report, Open Geospatial Consortium.
- Rautenbach, V., Coetzee, S., Strzelecki, M., and Iwaniak, A., 2012. Results of an evaluation of the orchestration capabilities of the ZOO project and the 52° North framework for an intelligent geoportal. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-4:163-168.
- Weiser, A. and Zipf, A., 2007. A visual editor for OGC SLD files for automating the configuration of WMS and mobile map applications. In Gartner, G., Cartwright, W., and Peterson, M. P., editors, *Location Based Services and TeleCartography*, 265-278. Springer, Berlin Heidelberg.
- Yau, N., 2011. *Visualize This*. Wiley Publishing, Inc, Indianapolis.