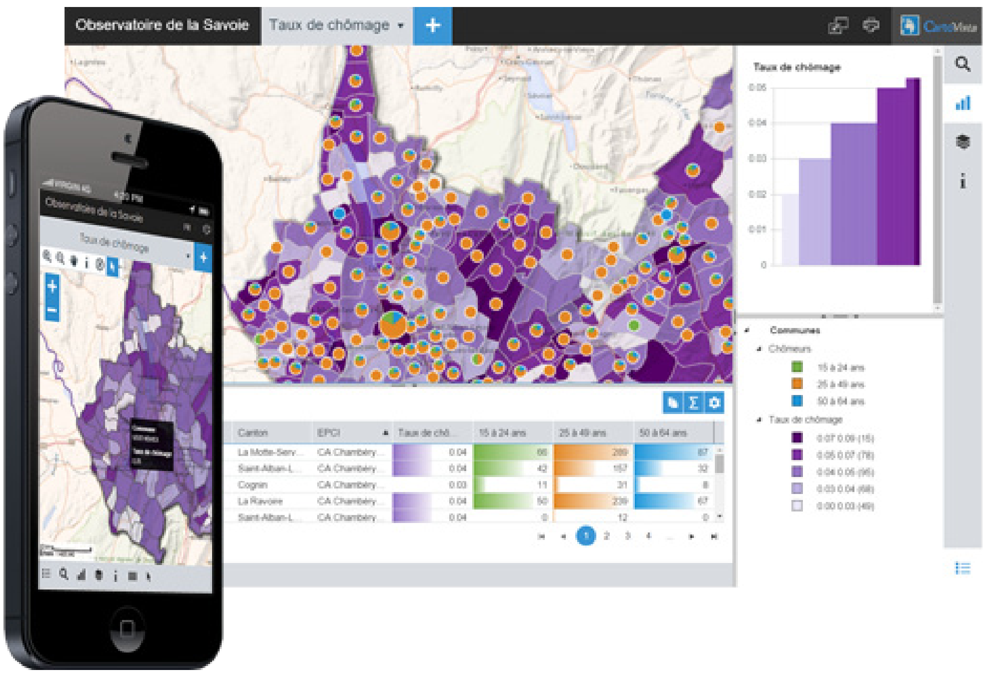
DBx GEOMATICS



Leveraging the HTML5 Canvas/Javascript for web and mobile maps with CartoVista

Dany Bouchard, DBx GEOMATICS inc.

ABSTRACT. Developing cross-browser mapping applications is a challenge that requires good design and a sound understanding of browser and mobile application capabilities. The HTML5 Canvas and Javascript open a new world of possibilities for creating interactive map content. However if you wish to interact with a large number of features, your map rendering has to be designed with care! The CartoVista 5 engine has been developed with advanced canvas rendering techniques, for both vector and raster data. With the proliferation of smartphones and tablets, HTML5 is an opportunity to deliver the richness of an immersive mapping experience

Keywords: HTML5, Interative Maps, Canvas, Rendering, Vector, Raster, Mobile, Apple iOS, Google Android, CartoVista

Table of Contents

Contents

Table of Figures

1. Introduction

HTML5 includes an exciting new element—the CANVAS element. This element has a lot of uses, but in order to use it you need to learn some JavaScript as well as HTML and sometimes CSS. This makes the CANVAS element a bit daunting for many cartographers, and in fact, most cartographers will probably ignore the CANVAS element until there are reliable tools to create CANVAS animations and games without knowing JavaScript at all.

The HTML5 CANVAS element can be used for a lot of things that previously you had to use an embedded application like Flash to generate:

- Dynamic graphics
- Online and offline maps
- Animations
- Interactive video and audio
- Games

In fact, the main reason cartographers and developers are starting to use the CANVAS element is because of how easy it is to turn a plain web page into a dynamic web application and then convert that application into a mobile application for use on smartphones and tablets.

1.1. Why not use Flash?

You may be thinking that we can already do dynamic graphics with Adobe Flash, but there are two major differences between CANVAS and Flash:

First, the CANVAS element is embedded right in the HTML. The scripts that draw on it are right in the HTML (or a linked external file) as well. This means that CANVAS element is a part of the document object model or DOM.

Flash is an embedded external file. It uses either the EMBED or the OBJECT element to display, and cannot interact as easily with the other HTML elements. Because the CANVAS element is part of the DOM, it can interact with the DOM in many ways. For example, you might create a map that changes when some other part of the page is interacted with—such as a chart or dropdown list.

Secondly, the CANVAS element is supported natively by most web browsers.

In order for your end users to use Flash, their browser must have the plugin. It used to be that every browser had the plugin installed, but that is no longer the case—many people are removing the plugin because of difficulties and the fact that it's not available at all on operating systems like iOS (iPhone and iPad).

1.2. Canvas Support in Today's Browsers

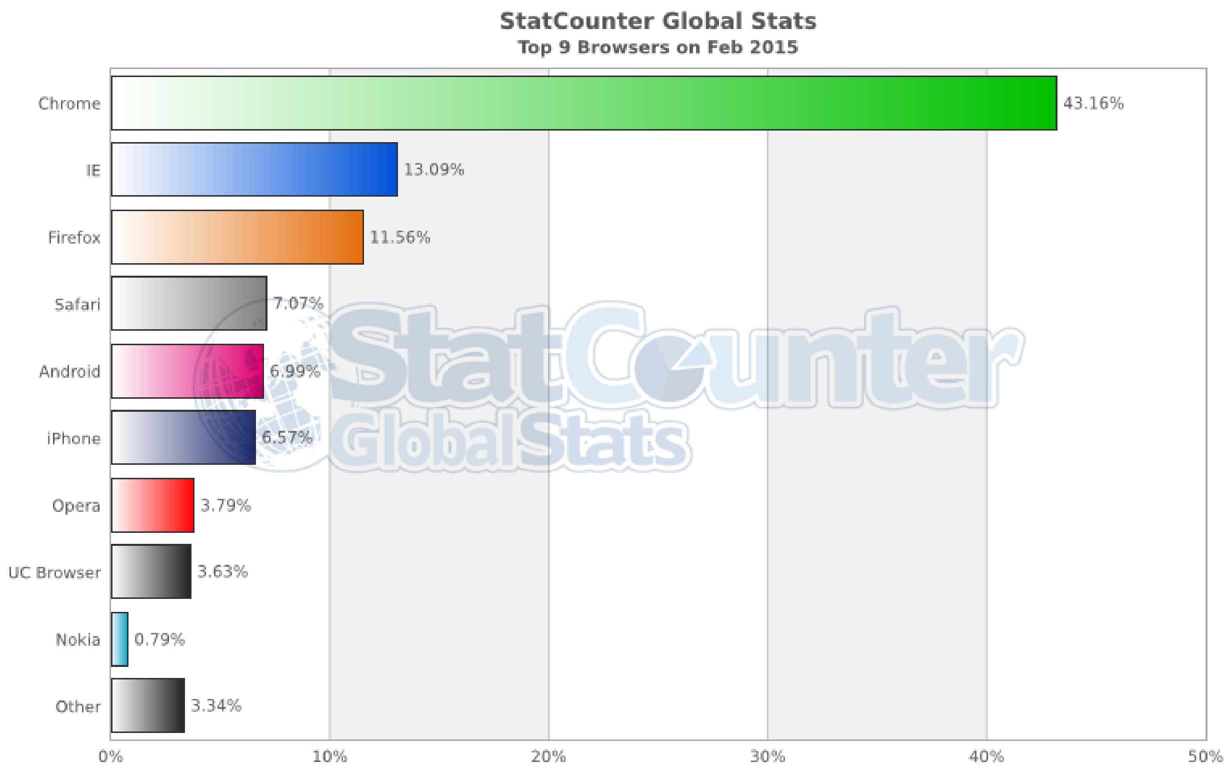


Figure 1: Browser Share as of February 2015 (Statcounter)

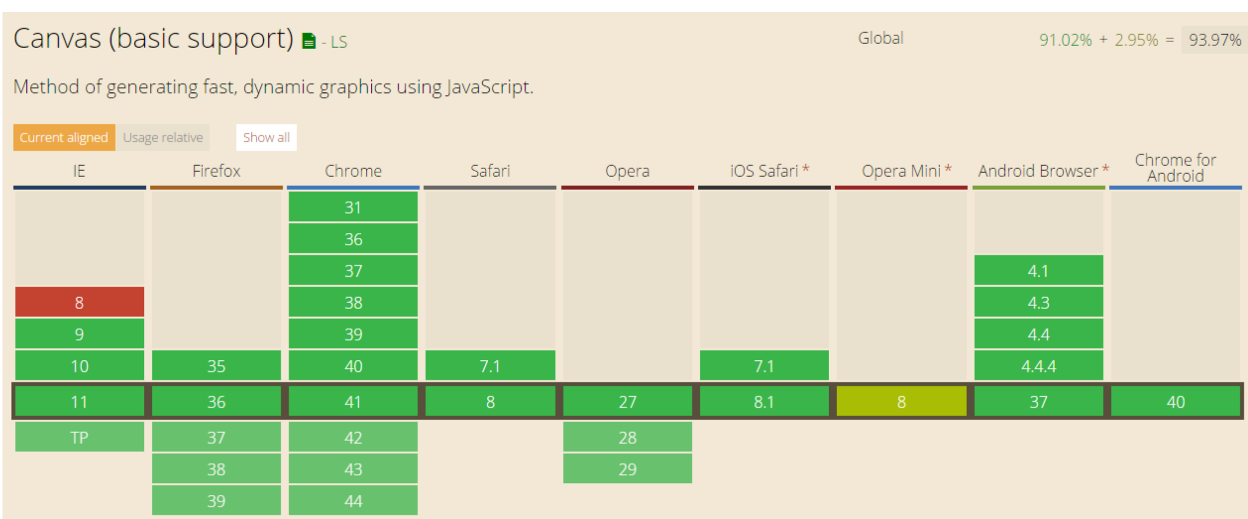


Figure 2: Basic Canvas Support in Modern Web Browsers

1.3. Basic Canvas Example

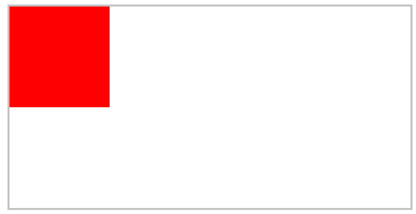
```
<html>
<body>

<canvas id="myCanvas" width="200"
height="100" style="border:1px solid
#c3c3c3;">
Your browser does not support the
HTML5 canvas tag.
</canvas>

<script>
var c =
document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.fillStyle = "#FF0000";
ctx.fillRect(0,0,150,75);
</script>

</body>
</html>
```

Result



1.4. The Spatial Rendering Challenge

Making your map rendering device-friendly and responsive is quite challenging when you take into consideration the following factors.

1.5. Performance

Smartphones and tablets have fewer hardware resources than desktop computers today—less RAM, CPU power, GPU resources, etc. In particular, this is an important factor to consider when working with complex vector map features and content (e.g. a large layer of polygons).

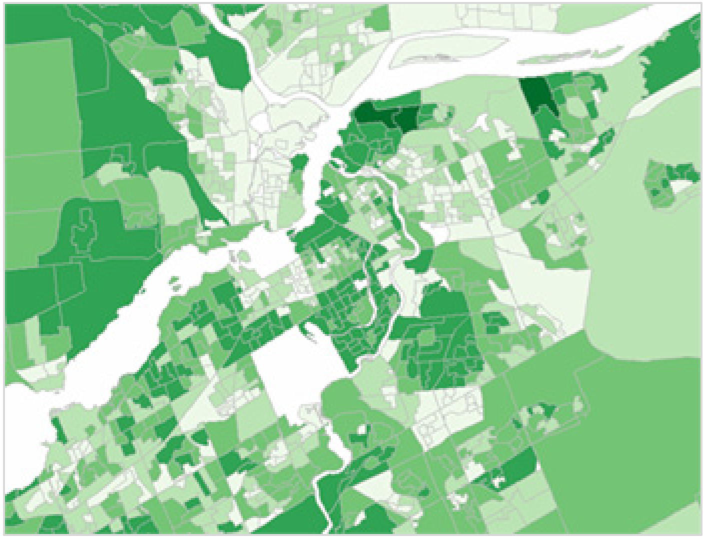


Figure 3: Complex Polygon Layer (Demographic Data)

1.6. Screen size

As the mobile applications continue to proliferate and reach more devices, developers need to adopt techniques for authoring with multiple screen sizes and resolutions in mind.

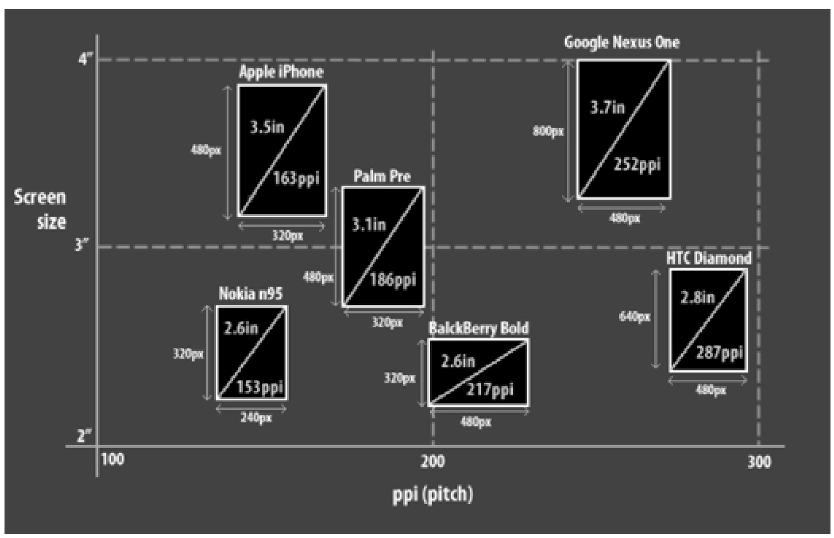


Figure 4: Different screen sizes for common mobile devices (From Adobe Systems)

1.7. Interaction - Mobile Map Navigation and Basic User Controls

Today's smartphones and tablets are becoming predominantly touch-screen devices for any native applications.

In addition to the web interface, the application UI should then anticipate touch-based input and shouldn't assume the presence of a physical keyboard or mouse.

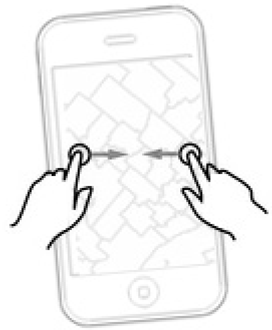
This is a key concept for mapping applications to enable simple and straightforward map navigation.

Figure 5: Mobile device interaction: touch-based input

Navigating the map with gestures on smartphones

The following map navigation methods are common and standard on almost all smart phone devices on the market.

One-Finger Drag		MAP ACTION: Recenter This common gesture used also for media content (images, etc.) allows to recenter the content view by a simple drag. To drag, users place a finger on the screen and move it in the desired direction without lifting it from the screen.
Two-Finger Zoom-In		MAP ACTION: Zoom in This common gesture used also for media content (images, etc.) allows zooming in the content. This gesture is executed by moving two fingers farther apart on the device (pinch open).

Two-Finger Zoom-Out		MAP ACTION: Zoom out This common gesture used also for media content (images, etc.) allows zooming out of the content. This gesture is executed by moving two fingers closer together on the device (pinch close).
One-Finger Double Tap		MAP ACTION: Zoom in This user gesture is commonly used to zoom in of content or an image. A double tap consists of two quick taps. On the map, one finger double tap allows zooming-in by factor of 2.
Two-Finger Tap		MAP ACTION: Zoom out This user gesture is commonly used to zoom out of content or an image. On the map, two-finger tap allows zooming-out by factor of 2.
Two-Finger Double Tap		MAP ACTION: Zoom out This user gesture is commonly used to zoom out of content or an image quickly. A double tap consists of two quick taps. On the map, two-finger double tap allows zooming-out by factor of 4.

One-Finger Tap		MAP ACTION: Get information on map feature (Attribute Info) A user gesture used to display an information bubble, magnify content under the finger, or to perform specific actions in built-in applications and features. To touch and hold, users touch the screen, leaving their finger motionless until the information is displayed or the action occurs.

2. Advanced HTML5 Rendering: Interactive Canvas Experience

2.1. Introduction – Optimizations required!

Several techniques in HTML5 / Javascript can be used by cartographers and developers to author content that will render properly on the web or any device, regardless of its screen resolution and pixel density. Javascript and HTML5 technologies offer a very rich set of functionality to help developer design their maps and UI graphics with ease.

However, in order to deliver a nice mapping experience while maintaining high graphic quality, the map content has to be optimized.

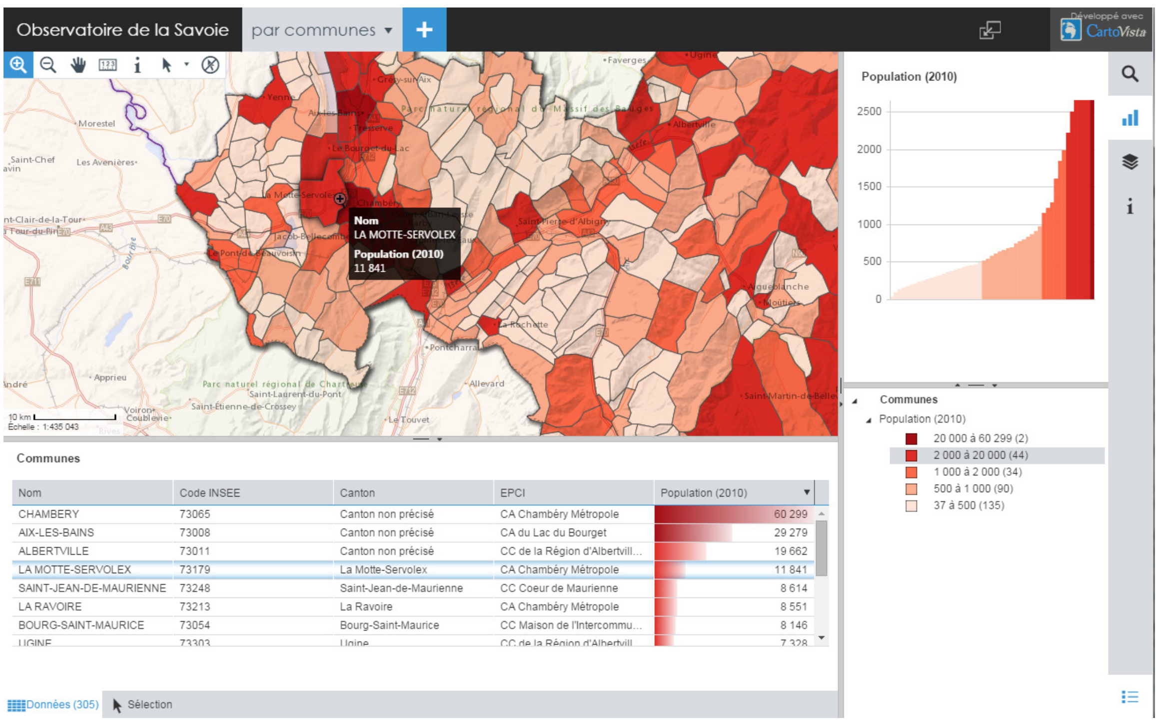


Figure 1: *The CartoVista Interface in HTML5 in a Web Browser– Statistical data*

Figure 2: *The CartoVista Interface in HTML5 on a mobile device browser (Safari)– Statistical data*

2.2. Map Content and Performance

Achieving good performance in web maps is best reached by selecting a solid fundamental approach to each aspect of how the map data is stored and most importantly rendered.

2.3. The separation of subject (business) and base map data

As map data can be very complex by nature, a clear separation of the subject data from the base map is the first step to plan for getting decent performance.

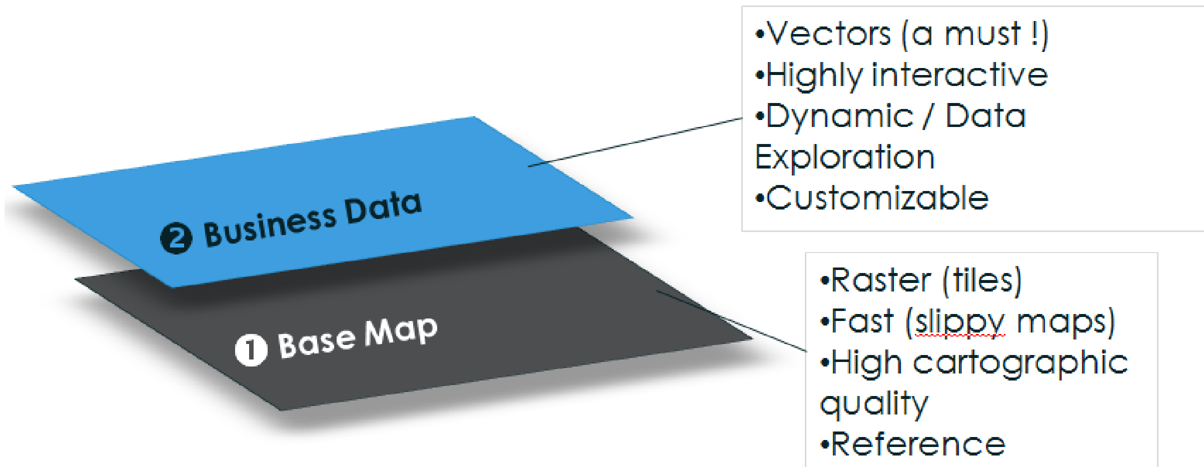


Figure 6: Base Map / Business Data Architecture

Tiled-map deployments have proven to be a very popular method to develop mapping functionality on the web, GoogleMaps being one the most popular example of this approach. This method of organizing the base map data is also ideal for web and mobile applications because tiling offers many tangible benefits:

- Seamless map navigation
- Screen resolution independence
- Low resources consumption
- Unlimited amount of details

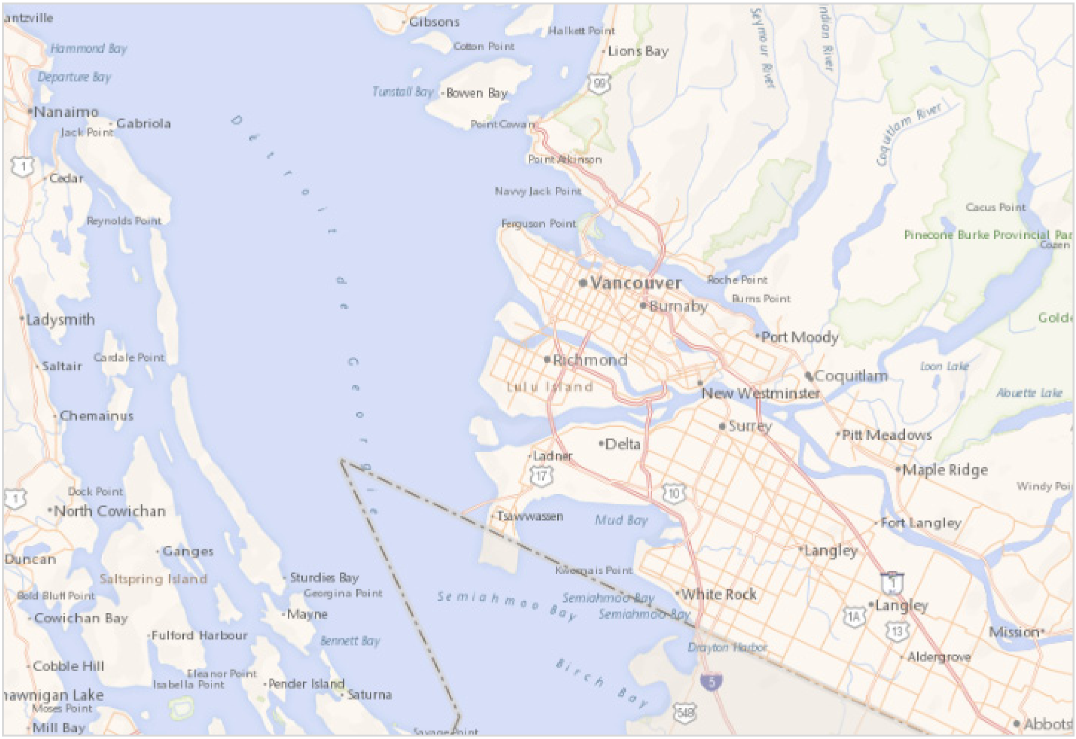


Figure 7: Tiled Basemap – Lambert Conformal Conic (Vancouver, Canada)

2.4. Overlay of layers of interest / Data binding capabilities

Setting up the base map with tiles allows focusing properly on the layers of interest for the end user. This is accomplished by overlaying one or more vector layers of interest on top of the map tiles base map. The layers of interest are defined with the required interactivity and potential thematic analysis options. The separation of the map from the business data creates a dynamic interface that avoids data duplication and opens the door to all sorts of data binding capabilities.

2.5. Vector Data - JavaScript Object Notation (JSON)

In order to get decent performance, the map vector content should be optimized for fast loading and smooth navigation in the web browser. Generalizing the map data using standard simplication algorithms (like Douglas-Peucker) is highly desirable for complex map layers.

JSON is the format of choice for storing and transporting the layer and feature data.

JavaScript Object Notation (JSON) is lightweight, language independent and self-describing.

A polygon feature in JSON would be expressed like the following:


```
{"scaleUpFactor":0,"type":"FeatureCollection","proj":"+proj=lcc +lat_1=49 +lat_2=77
"f":[{"t":"Feature","g":{"t":"Polygon","c":[[2451038,987048,18297,1032,3529,-72,6017
```


Figure 8: Polygon Feature Stored in JSON

The feature coordinates are stored in the array item "c". Note that the coordinates for the first node are expressed with the full number while all other coordinates in the ring are relative. This signifiantly reduces the size of the output and speeds up the parsing process in the web browser.

When working with optimized JSON objects, the canvas rendering is fast (panning, zooming, loading of a theme, etc.), at most times, rendering is completed in less than 1 second.

2.6. Canvas Rendering - Blend Modes

When working with polygon layers, it is desirable to show the underlying tiled base map at the same time as the polygon information, in a consistent manner. Layer blend modes are very useful to allow to "mix" the map layers properly. The following figure shows a polygon layer overlayed on top of the basemap with a "multiply" blend mode. Multiply blend mode multiplies the numbers for each pixel of the top layer with the corresponding pixel for the bottom layer.

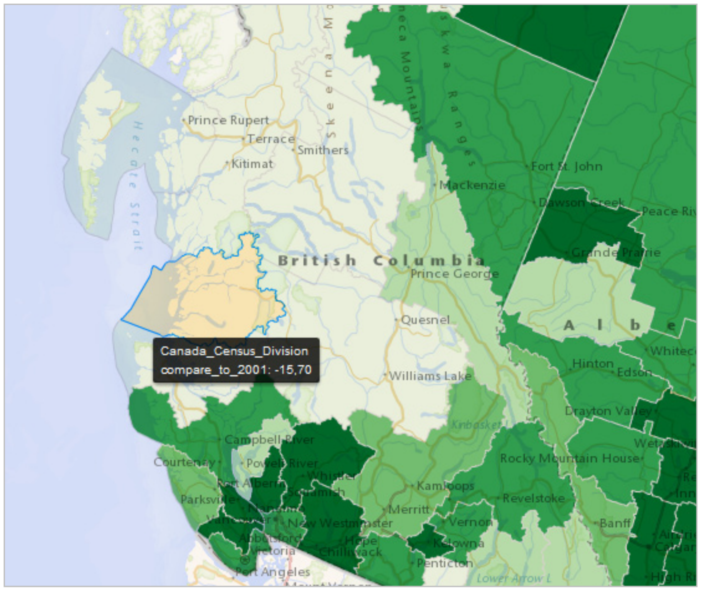
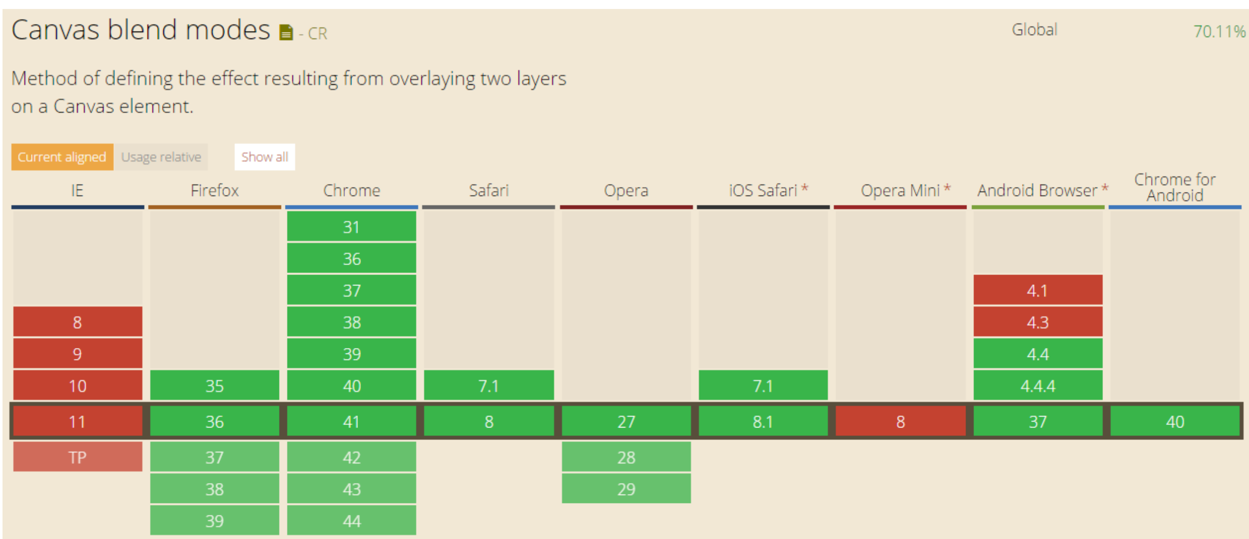


Figure 9: Tiled base map and overlay of polygons (Choropleth Analysis) – Multiply Blend mode (Green theme)

Blend mode is well supported in all modern web browsers, except for Internet Explorer.



Canvas blend modes - CR — Global 70.11%

Method of defining the effect resulting from overlaying two layers on a Canvas element.

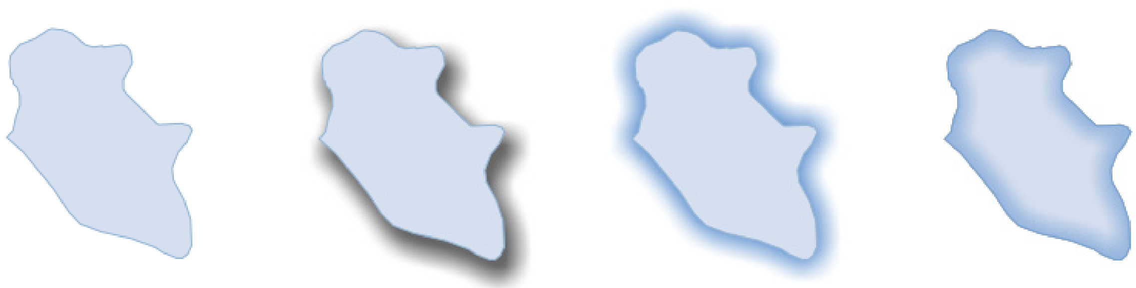
Current aligned | Usage relative | Show all

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		31						
		36						
		37					4.1	
8		38					4.3	
9		39					4.4	
10	35	40	7.1		7.1		4.4.4	
11	36	41	8	27	8.1	8	37	40
TP	37	42		28				
	38	43		29				
	39	44						

Figure 10: Canvas Blend Mode Support

2.7. Canvas Vector Rendering – Filter Effects

The HTML5 Canvas includes great support to enhance the rendering of vector map data. In CartoVista, the following effects are supported:



No Effect Drop Shadow Outer Glow Inner Glow

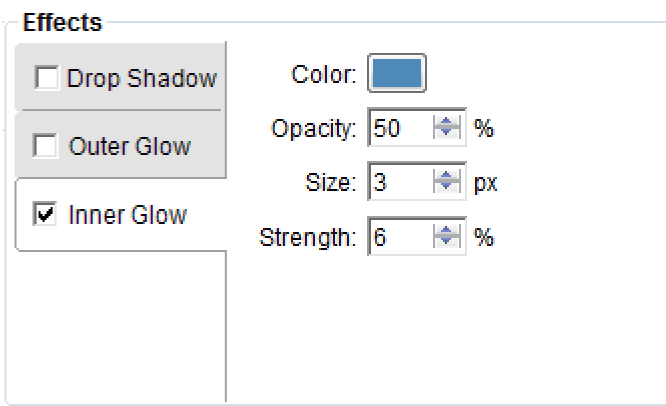


Figure 11: Inner Glow Filter Effect Settings

Each effect option includes several parameters to control its appearance on the layer such as:

- Color
- Opacity
- Size
- Strength

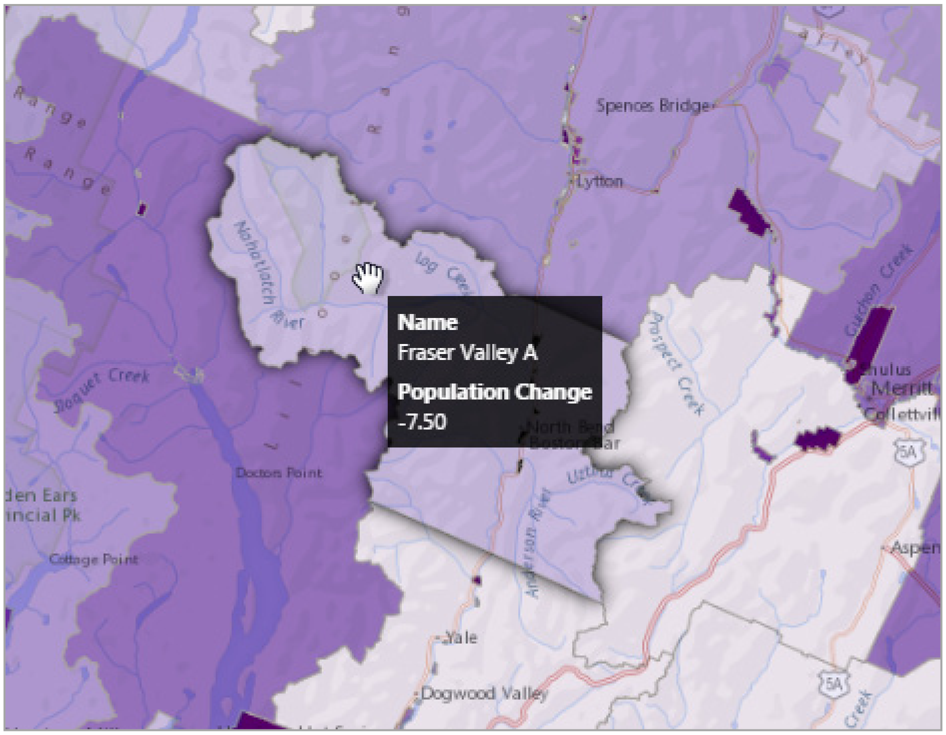


Figure 12: Use of a drop shadow effect when mousing over a polygon to enhance the rendering

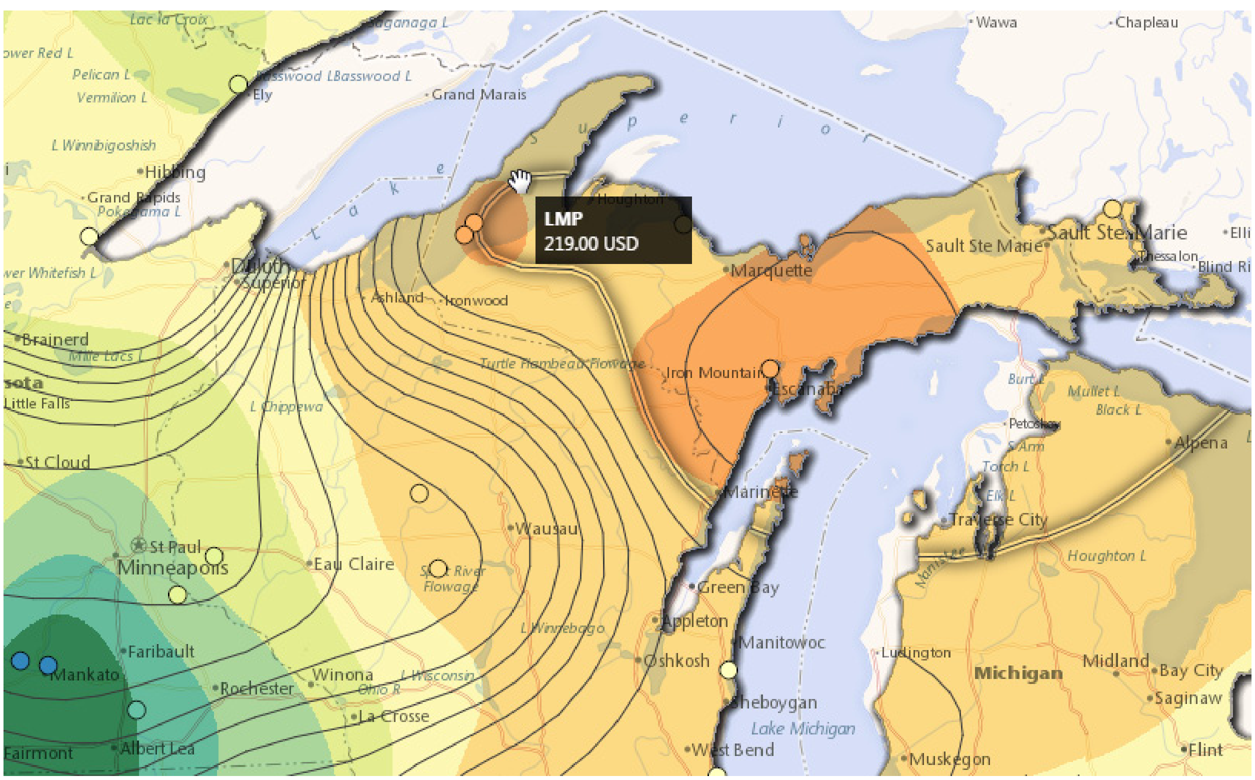


Figure 13: Use of a drop shadow effect on a layer of polygons and rollover on lines (contours) to enhance the rendering

2.8. Canvas Vector Rendering – Uses for Thematic Mapping

The use of the HTML5 Canvas opens the possibility of creating advanced thematic analysis, combining multiple indicators (bi-variate analysis, etc.) on layers of points, lines and polygons.

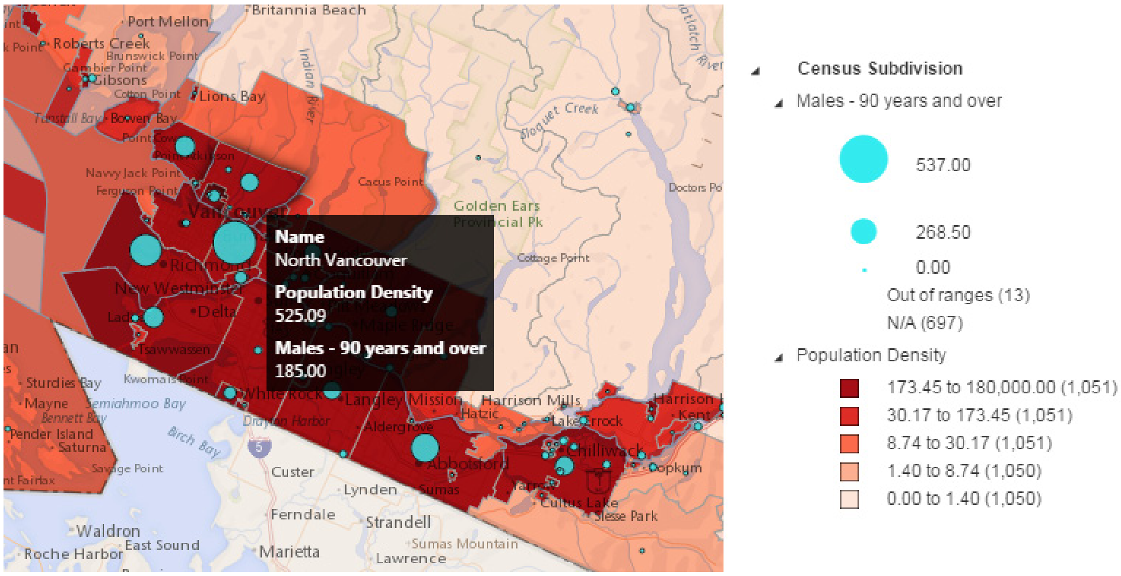


Figure 14: Background and overlay (graduated symbols) thematic analysis in CartoVista

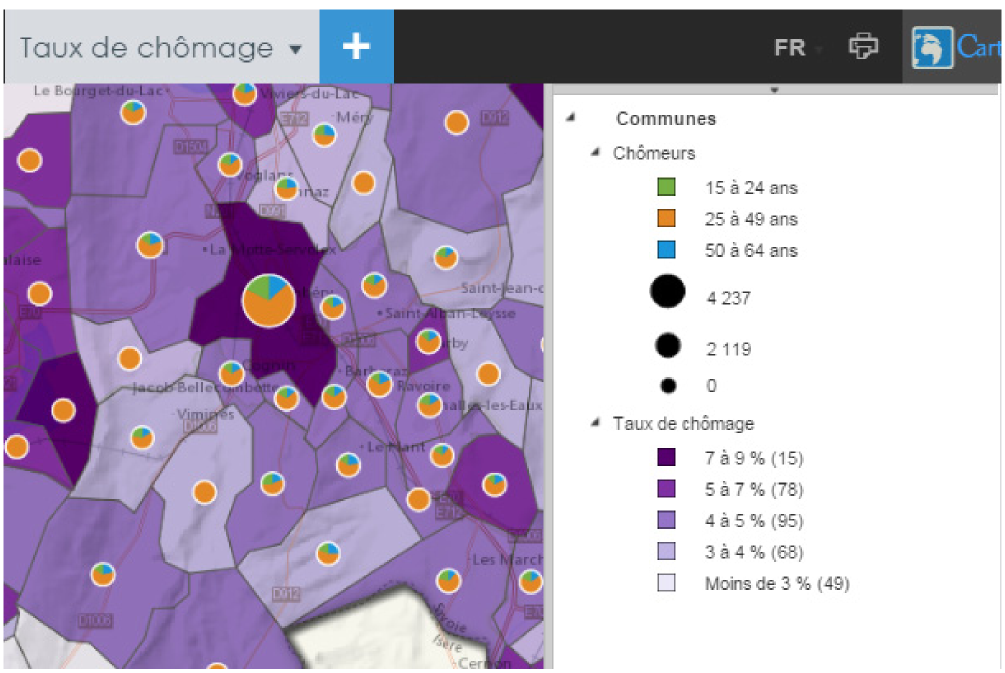


Figure 15: Background and overlay (pie chart) thematic analysis in CartoVista

Several thematic analyses can be integrated in a simple treeview for efficient data presentation

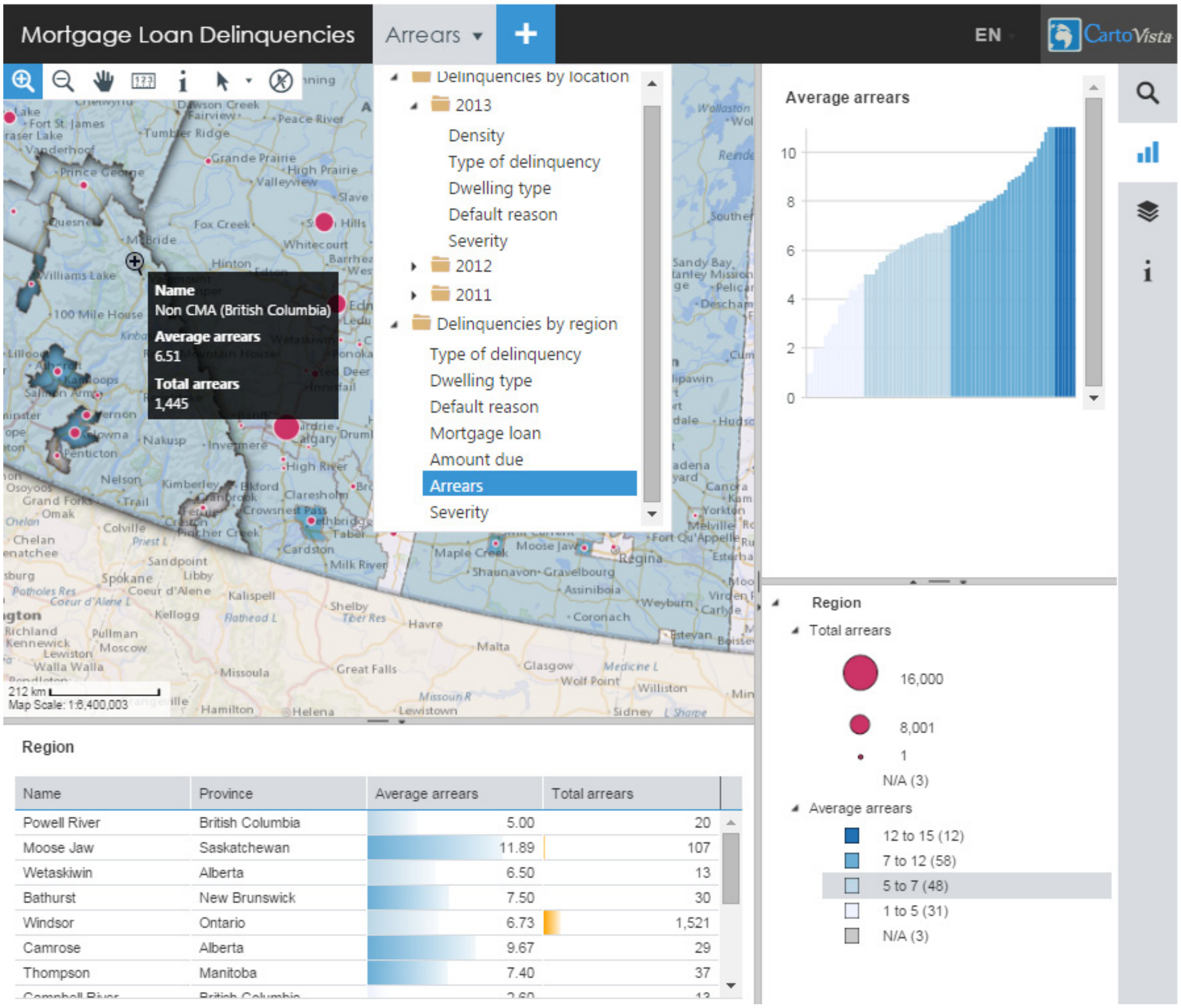


Figure 16: Presentation of map themes in the CartoVista HTML5 Viewer (Mortgage Loan Delinquencies)

The vector rendering provide the capabilities to change the thematic analysis parameters directly in the map.

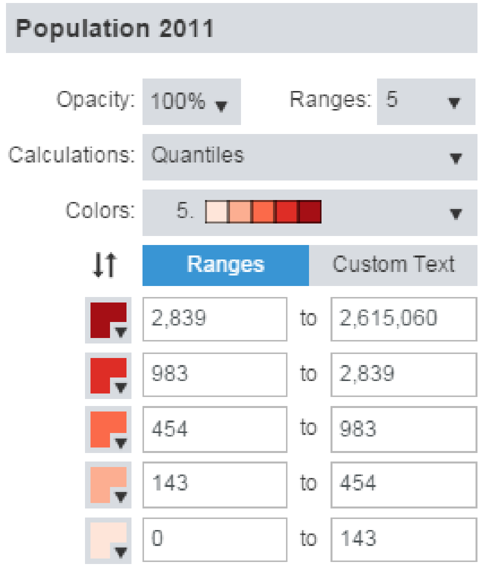


Figure 17: Thematic Analysis Configuration (choropleth) in the CartoVista HTML5 Viewer

2.9. Raster Interactivity

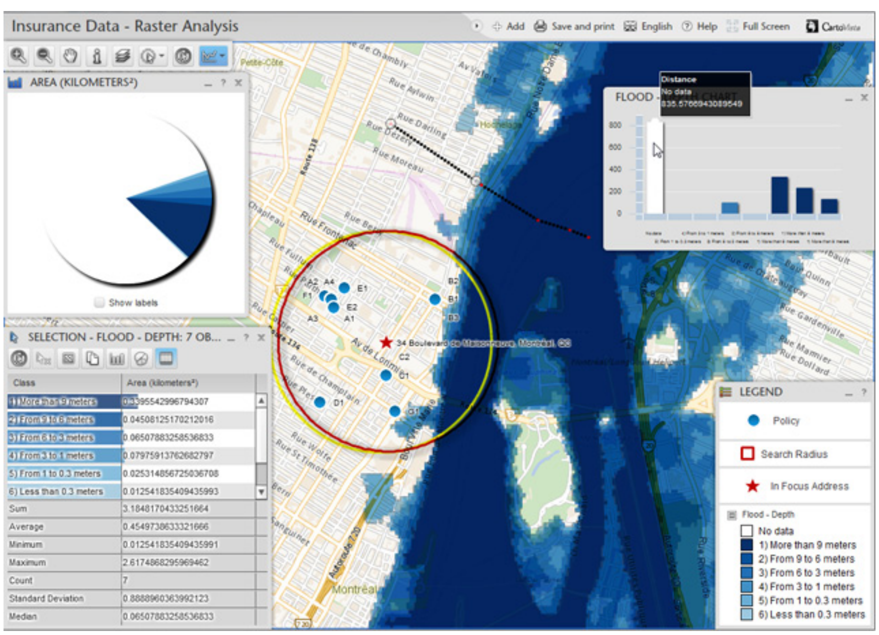


Figure 18: Classified Grid – Raster Interactivity on tile layer (flood risk layer)

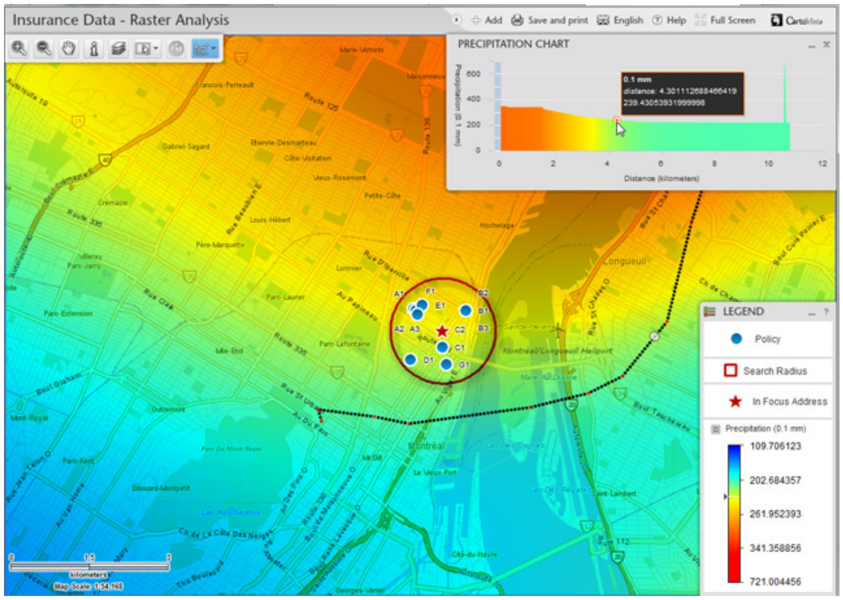


Figure 19: Numeric Grid – Raster Interactivity on tile layer (precipitation layer)

2.10. Canvas Interactivity – Feature Information

The HTML5 Canvas offers the capacity cretate fully interactive layers with support for datatips and user friendly spatial data query tools (info, selection, etc.)

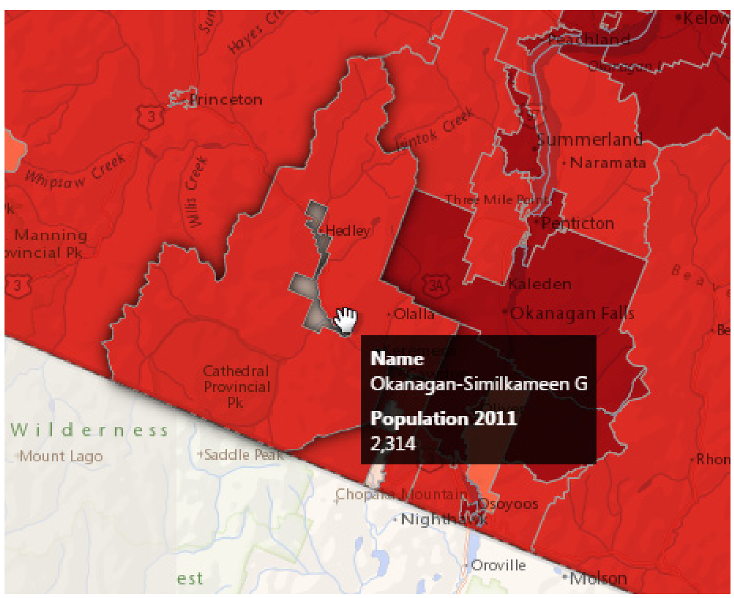


Figure 20: Rollover – Datatip in the CartoVista HTML 5 Viewer

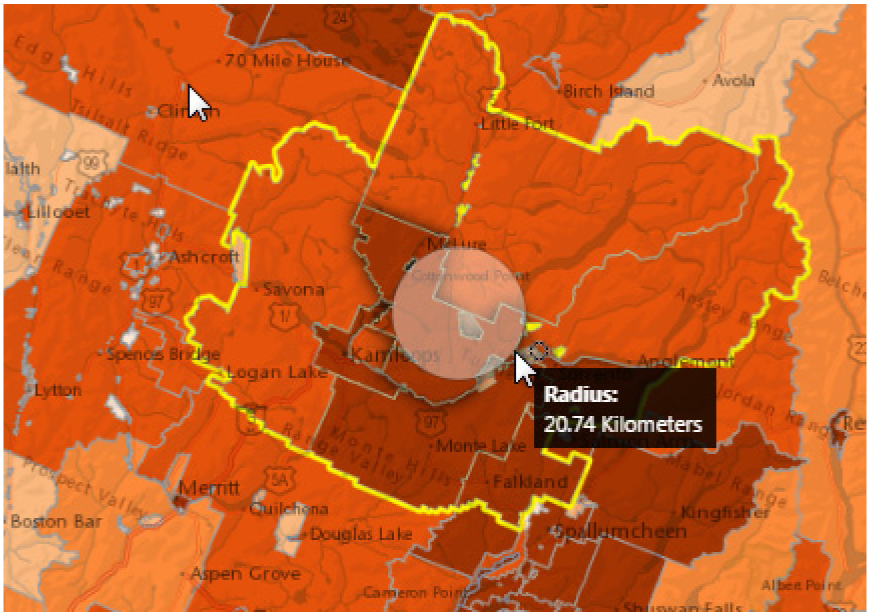


Figure 21: Selection by Radius in the CartoVista HTML 5 Viewer

Census Subdivision

Name
Okanagan-Similkameen D

Population 2011
5,717

Population 2006
5,913

Population Density
6.22

National Population Rank
633

Population Change
-3.30

Private Dwellings Occupied by Usual Residents
2,436

Total Private Dwellings
2,845

Figure 22: Info Panel in the CartoVista HTML 5 Viewer

2.11. Canvas Interactivity – Tables and Charts

HTML5 offers new possibilities for visualizing data with interactive charts (histogram, pie charts) in relationship with the map displayed

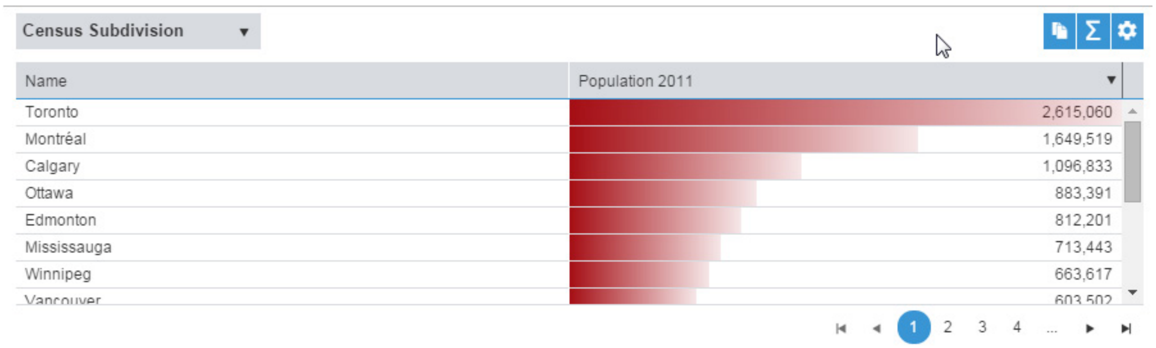


Figure 23: Data table in the CartoVista HTML 5 Viewer

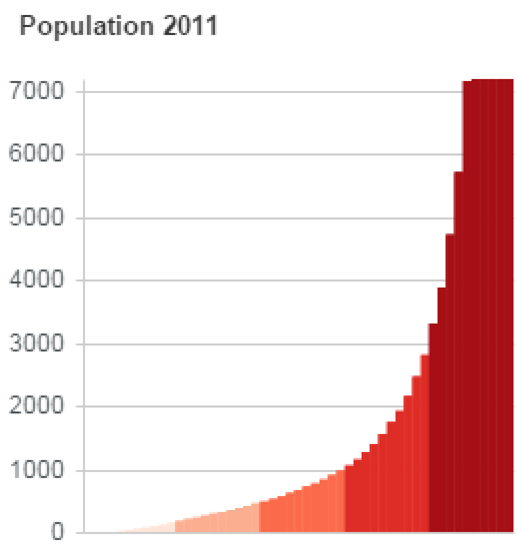


Figure 24: Associated chart view in the CartoVista HTML 5 Viewer

The canvas offers the capacity to propagate events so that components can interact with each other. All sorts of interaction between the map, legend, histogram, data table, layer control, etc. are then possible.

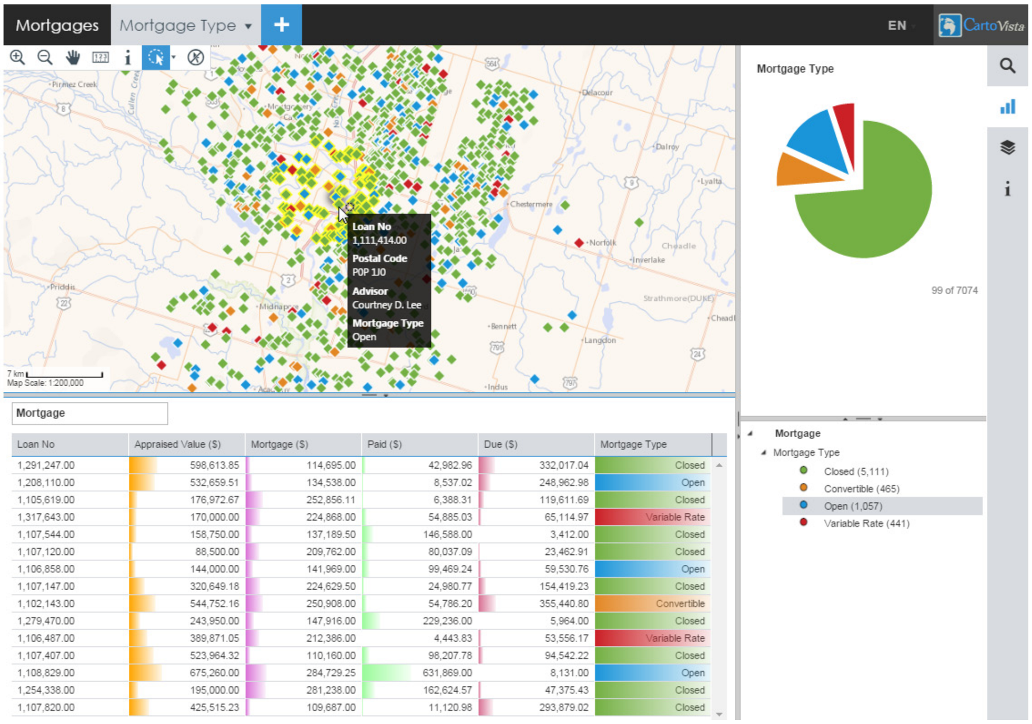


Figure 25: Interaction between the map, legend, pie chart and data table in the CartoVista HTML 5 Viewer

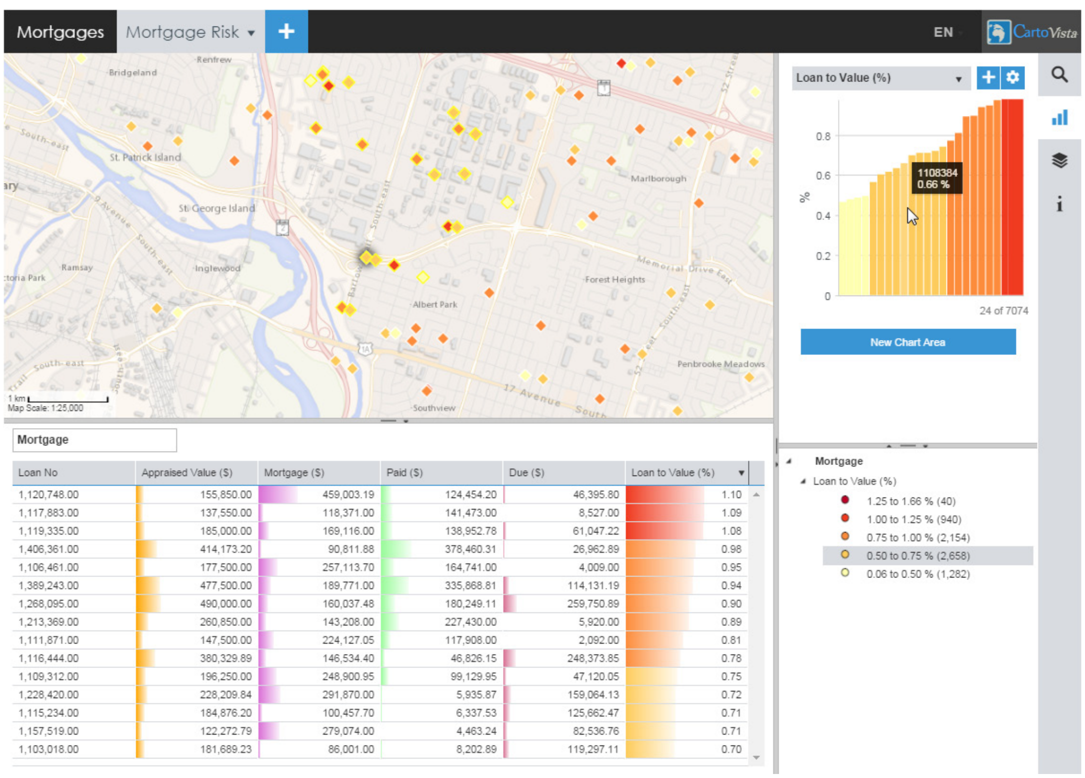


Figure 26: Interaction between the map, legend, histogram and data table in the CartoVista HTML 5 Viewer

2.12. Javascript / Customization and Extensibility

The JS/Canvas/CartoVista SDK combination offers multiple possibilities to customize the user interface with existing HTML5 frameworks (such as KendoUI, ExtJS, etc.)

The CartoVista SDK includes a full API documentation to develop custom applications.

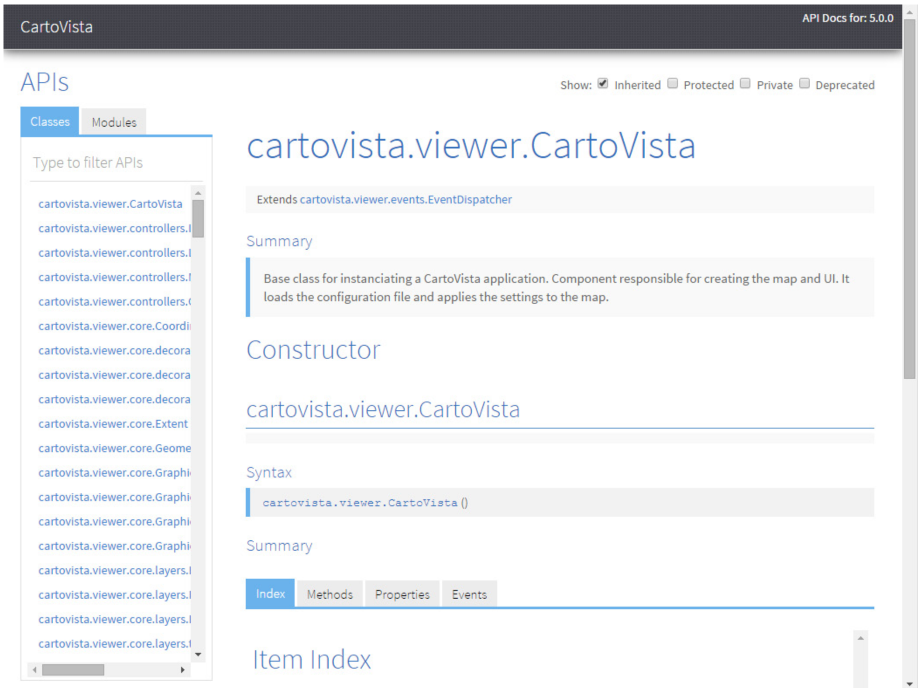


Figure 27: CartoVista SDK Documentation (HTML5 Viewer)

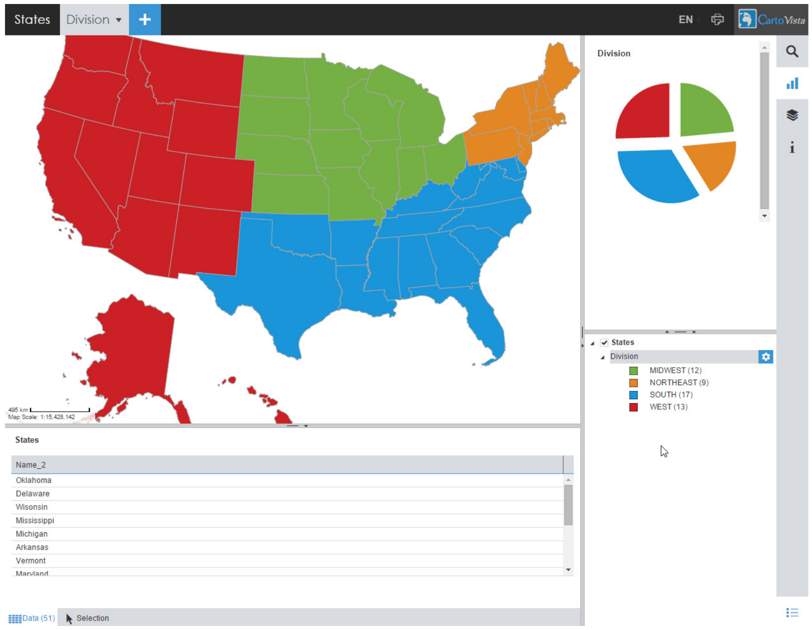


Figure 28: SDK Sample Project to get event information from map events (selection)

Many HTML5 frameworks include rapid application development (RAD) components for the creation of customized applications (with very decent support for UI, inheritance, etc.).

In addition to multimedia applications, dashboard solutions can be built with full interactivity on desktop browsers and mobile devices

Figure 29: SDK Mutimedia Sample Project – Interactive Shoreline Video

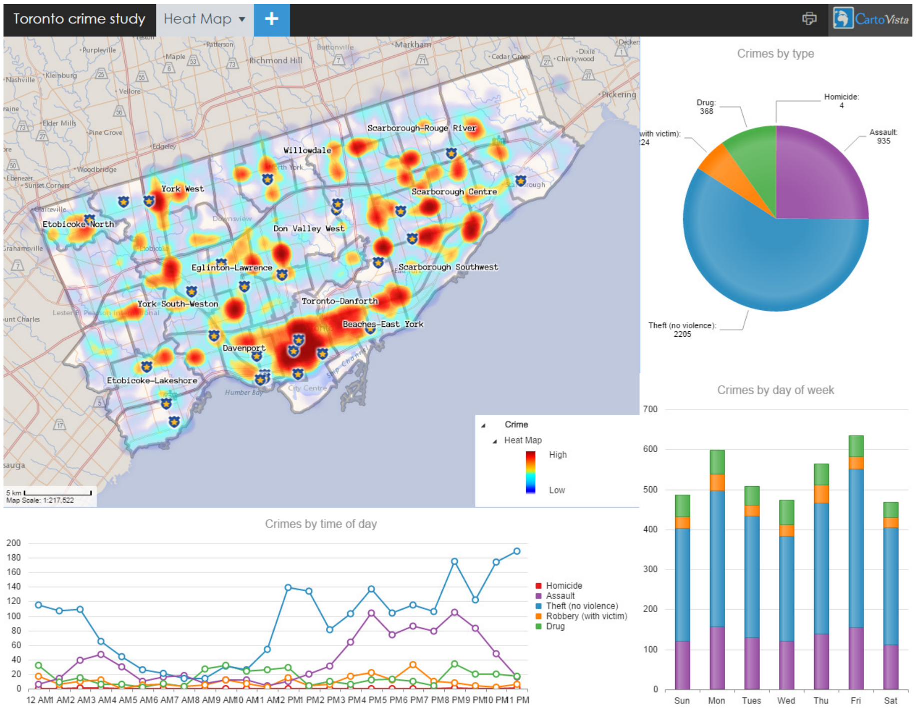


Figure 30 : HTML5 - Crime Analysis Dashboard

3. Conclusion

The presentation introduces the design patterns differences between mobile and browser-based applications while reviewing the challenges to create a seamless, predictable, Canvas-based mapping experience between the end users and the cartographic content.

This paper provides concrete map examples of how the HTML5 Canvas was used in CartoVista for web and mobile maps, combining a rich user experience with high quality rendering:

- Web Interface Vs Mobile Interface
- Vector Data Content Optimizations
- Advanced rendering (Blend Modes, opacity, filter effects, drop shadow, glow effect, etc.)
- HTML5 Data Interactions (Charts, data tables, legends, etc.)

We hope the maps and techniques included in this presentation will help developers and cartographers understand how to build map applications that are better-suited for mobile and tablet devices while leveraging the benefits of today's browsers graphic capabilities.

References

Bouchard D, DBx GEOMATICS inc., CARTOVISTA - ARCHITECTURE & DESIGN, URL: http://www.cartovista.com/solutions.aspx

Bouchard D., DBx GEOMATICS inc., CANADA DEMOGRAPHICS IN 2011, URL: https://www.cartovista.com/cartovista/demos/canada-demographics/

Bouchard D., DBx GEOMATICS inc., DELINQUENCIES & FRAUD ANALYSIS IN THE LAST YEAR, URL: https://www.cartovista.com/cartovista/demos/delinquencies-fraud-analysis/

Bouchard D., DBx GEOMATICS inc., OBSERVATORY OF SAVOIE, FRANCE, URL: https://www.cartovista.com/cartovista/demos/observatory-of-savoie-france/

Bouchard D., DBx GEOMATICS inc., MARKET ANALYSIS IN INSURANCE BY PRODUCT TYPE, SUMMARY OF CLAIMS AND RESULTS, URL: https://www.cartovista.com/cartovista/demos/insurance-market-analysis/

StatCounter, Global Stats, URL: http://gs.statcounter.com/#all-browser-ww-monthly-201502-201502-bar

Alexis Deveria- Can I use, Up-to-date browser support tables for support of front-end web technologies on desktop and mobile web browsers), URL: http://caniuse.com/#feat=canvas