

General Article

## A Review of Google Data Visualization Tools

Jheel Doshi<sup>A\*</sup>, Ishan Goradia<sup>A</sup> and Dharmeshkumar Mistry<sup>A</sup>

<sup>A</sup>Computer Department, DJSCOE, Vile Parle (W), Mumbai – 400056, India

Accepted 05 Sept 2014, Available online 01 Oct 2014, Vol.4, No.5 (Oct 2014)

### Abstract

The ever expanding information overload has caused a data glut. To navigate through that much amount of data we need to start using our visual senses more. That is, Visualizing information, so that we can see the patterns that matter and focus on the data that is important. In this field, Google has already made striding leaps by not only developing Google Charts, but also introducing the option of directly creating data visualizations on Google Maps using Fusion Tables and Google Spreadsheets. Using these tools, Google has effectively captured the market and has also made it very easy for beginners with minimum coding experience to create good data visualizations.

**Keywords:** Google Charts, Google Maps, Google Visualization API, Data visualization tools, Fusion Tables

### 1. Introduction

Google Visualization provides a perfect way to visualize data dynamically for research, presentation or embedding the charts on the website. Google gives you a data source that is, Google spreadsheets, it also lets you perform queries on that data using the Google Visualization API query language and finally lets you display it on the web using the Google charts libraries. It's a free resource and doesn't require a licence. The API has everything from bar charts to complex hierarchical tree maps and even QR codes. It also includes built-in animation and user interaction controls. (Ying Zhu, 2012). There are different categories of charts Google provides an option to create:

- 1) Data Table Object: A database which uses JavaScript. It queries just like SQL. It is sortable, filterable & can be easily managed. An Example of a simple data Table object has been denoted in Figure 1.
- 2) Image Charts.- Use the Google Chart API. The Google Live Chart Playground is also an excellent tool again for static Images. One can play with parameters and see the changes in the generated image - an easy way to fine-tune the url for your graph. The Chart API provides a simple way to create image charts of various kinds by sending a formatted URL that includes both the data and chart configuration options to a Google server. The Chart API includes a closed set of charts & its datasets are limited to the size of a URL (roughly 2K). The part of tools on the google website for static images has been deprecated, so the Google Chart Tools site currently only offers tools for dynamic charts.
- 3) Interactive Charts.- Use the Google Visualization API. They are far more flexible, because they can trigger events which we can use to interact with other elements in the page.

Task	Hours
Work	11
Eat	1
Commute	2
Watch TV	3

**Fig.1** A Simple Data Table object denoting hours spent

Google Maps API uses a set of JavaScript classes that let users customize and embed Google Maps in their web pages. It can be used to visualize data points related to Geo locations or Distance Matrix. Users can also make queries which are of a spatial nature by using Google Fusion Tables along with the Map API. Google Fusion Tables. One can visualize their data immediately after uploading it. A map is created if we detect a location column for example a column with street addresses, or a pincode.

One will probably find the right visualization for their needs with Google visualization Tools unless one is in need of extreme customisation. As a beginner in data visualization skills, it is a great tool to know how to use.

This review paper focuses on Google Visualization API, Google Maps API and Google Fusion Tables, describing in-depth the various features of Google Visualization API and giving an application where it is used in the healthcare sector.

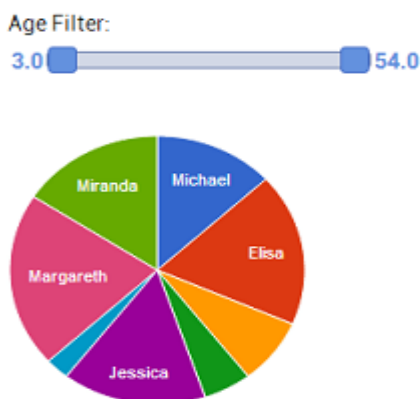
### 2. Google Visualization API

To include interactive charts in web pages, you have to use a different API: the Google Visualization API. Google visualization API is used to create interactive charts and other data visualization aids for online use. It uses its own

\*Corresponding author: **Jheel Doshi**

query language, has event triggers, dashboards. The different types of interactivity are shown in figure 2 and figure 3 wherein the data is visualized in the form of Pie Chart. Google Charts are based on HTML5 and scalable Vector-Graphics and hence they are robust and work on all browsers supporting Canvas, SVG & VML, with no plugins required. It has cross platform compatibility. Interactive uses a JavaScript, but there are libraries present that enable Flash to connect with JavaScript code. The ways to instantiate and draw Google Charts:

**Chart.draw()** : which is the most basic way . It involves loading the required libraries, then loading the data points that have to be visualized and choose the chart options to customize the data visualization. Then instantiate the chart class by passing an element to the webpage code by using a <div> in the website code is enough to add the Google chart to the website. Consequently register for any events or call required methods. Finally call **chart.draw()** . This requires all libraries to be explicitly loaded.  
**ChartWrapper** : Is a convenience class that removes all the drawbacks associated with the previous method by loading all the appropriate chart libraries and simplifies the querying process . Pass in the container element ID, & the **getElementById** is called for user. **DrawChart** : It is a global static method that wraps a **ChartWrapper** which was discussed above and it is much shorter than it .But **DrawChart** doesn't eliminate the drawbacks of **ChartWrapper** .



**Fig.2** A pie chart with a numberRange filter

Gender Selection:

Name	Gender	Age	Donuts eaten
Michael	Male	12	5
Elisa	Female	20	7
Robert	Male	7	3
John	Male	54	2
Jessica	Female	22	6
Aaron	Male	3	1
Margareth	Female	42	8
Miranda	Female	33	6

**Fig.3** The DataTable with drop down controls for gender

### 2.1 Data Manipulation in Google Visualization API

A query string can be attached to a data source request in the following two ways: by placing the query string within the JavaScript code on the website, or by setting the query string as a parameter in the data source URL. If the request does not require a query string, the default behavior for a data source is to return all rows and columns using its default row/column order and formatting. This can be changed by including a query string in the request to the data source. The Google Visualization API Query Language syntax is very similar to SQL syntax. Even though, it is a subset of SQL , it has a few different features which are unique to it, for example Pivot. Google charts also has static methods to perform SQL-like operations on **DataTable** objects & charts , for instance **groupBy()** on a column value or performing Inner or Left joins. Also it contains modifiers and aggregate functions(Average , Maximum), scalar functions and a tool for arithmetic operations.

The Google Visualization API provides formatters that are used to re-format the data in a chart. These formatters change the value of the specified column in all rows. Formatters only work on one column at a time; to make changes on multiple columns, one has to apply a formatter on each column that you want to change. Some of the formatters are:

**ArrowFormat** : Adds an up or down arrow, indicating whether the cell value is above or below a specified value.

**BarFormat** : Adds a colored bar, the direction and color of which indicates whether the cell value is above or below a specified value.

**ColorFormat** :Colors a cell according to whether the values fall within a specified range.

**DateFormat** : Formats a Date or DateTime value in several different ways, including January 1, 2009, 1/1/09 and Jan 1, 2009.

**NumberFormat** : Formats various aspects of numeric values.

### 2.2 Chart Editor for real time interactivity

The **ChartEditor** class is used to open an in-page dialog box that enables a user to customize data visualizations in real time. A toolbar element to any visualization chart to enable the user to export the underlying data into a CSV file or an HTML table. It can also be used to provide code to embed visualization into an arbitrary web page or gadget. This improves human interactivity of charts to a great degree.

Dashboards are a toolkit provided to organize and manage multiple charts that share the same underlying or related data. By using the APIs described in this page, user can be freed from the burden of coordinating all the charts that are part of a dashboard. It instances receive a **DataTable** containing the data to take care of drawing and distributing the data to all the charts that are part of the dashboard. An Analytics Dashboard module is also included. The Dashboard creates an analytics page with time frame filters and some defaults site analytics charts, with the tools to add new charts and control the charts visibility.

Controls are UI widgets (category pickers, range sliders & Auto-completers) which a user interacts with in order to derive the data managed by a dashboard and the charts that are used by it . They collect data and decide how much and which part of that data should be made available to the charts using it just like pipes and valves in a plumbing system.

For events for increased human interactivity ,usually the graph will react in a certain way when clicked (showing a tool tip or animating), but a very useful feature is that they can trigger events and one can register call backs to perform any action related to that event. These trigger events can be selecting a particular pie slice , mouse Over & Out.

Google charts fire events that are triggered by user interaction can be internal: for example, a timer issuing an event every 30 seconds. Every chart defines and has its own events .There is one event that any selectable chart should fire: the select event. However, the behavior and meaning of this event is defined by each chart. It is important to note that the chart events are separate and distinct from the standard DOM events. Most charts are rendered asynchronously; all Google charts throw a ready event after user calls draw() on it, indicating that the chart is rendered, and ready to return properties or handle further method calls. Charts should throw an error event when they encounter an error, to enable its graceful handling . The event handler is passed a description of the error and custom event properties specific to each chart. Filters are graphical elements that enables the user to interactively select which data is displayed and it is very similar to Controls.

The 5 Google Chart filters are:

**ChartRangeFilter:** A slider with two thumbs superimposed onto a chart, to select a range of values from the continuous axis of the chart.

**StringFilter :** A simple text input field that lets you filter data via string matching. It updates after every keypress .

**CategoryFilter:** A picker to choose 1 or more values between a set of Pre defined values .

**DateRangeFilter :** A dual-valued slider for selecting ranges of dates.

**NumberRangeFilter :** Similar to DateRange Slider but for numeric values. Shown in figure 2.

### 2.3 Google Charts Gallery

The Charts Gallery consists of interactive charts that can be customized to the required data visualization needs .Out of which many are Pannable and Zoomable . Also changing from one chart object to another just requires the name changed at one place. All chart types are created with data using the DataTable class.. The DataTable can be used for sorting, modifying, and filtering data . Data tables can in turn be populated using data from any of the data sources mentioned earlier in the paper. So trying out different charts to find the right one is also an easy process. Google Charts supports 17 basic types of charts which are given below:

Pie Chart, Geo Chart, Bar Chart, Scatter Chart, Histogram, Combo Chart, Area Chart, Stepped Area Chart, Line Chart, Bubble Chart, Donut Chart, Org Chart, TreeMap Chart, Table , Timeline, Gauge, Candlestick.

Most of the Charts resemble their names and examples of two of them are given in Fig.4 & Fig.5.



Fig.4 A Gauge Chart to show resource distribution.

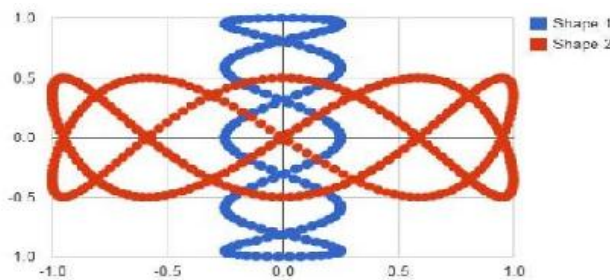


Fig.5 A Scatter chart depicting two function shapes.

Some of the Google community sourced chart types are really popular. Out of which two of these have been discussed below:

**Annotated Timeline:** is Part of the Google Visualization graphing API. An annotated timeline is an interactive time series line chart with the option to add details and annotations. You can display one or more lines on the chart. Example of Annotated timeline of an investment company over the years is given in Fig.6 .

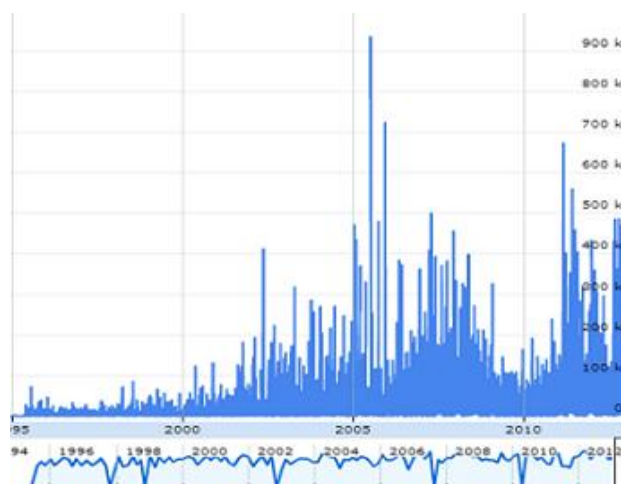


Fig.6 An annotated timeline.

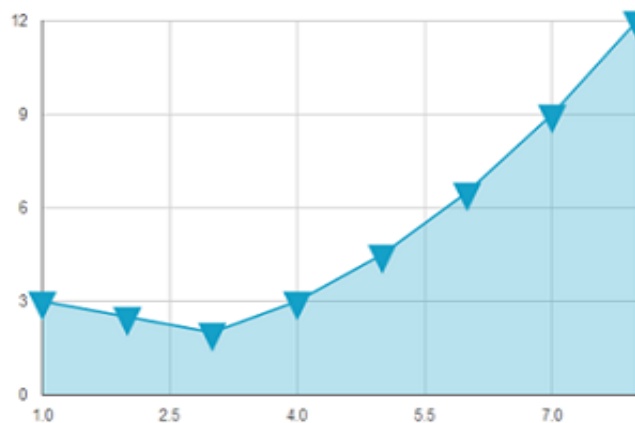
**Heat Maps :** They are a useful way to visualize matrices of data. Scientists often use green-black-red heat maps to visualize gene expression data from microarrays. This visualization supports both three color heat maps (ex: green to black to red) and two color heatmaps.

Also Google offers a Dent option is now available. In this Data Values are displayed at precise points in the

**Table 1** Comparative Study

	Google Chart Tools	Jqplot	gRaphael
Summary	Displays live data on a website	Versatile and Expandabl j query plotting plugin	Stunning chart designs for a website
Browder supported	HTML5 charts using SVG and VML	HTML5 charts using Canvas	HTML5 charts using SVG and VML
Chart & Map types	13 chart types in 2D. Maps available as GeoChart. Some 3D charts are available.	25+ chart types in 2D. Maps and 3D charts not supported.	4 chart types in 2D. Maps and 3D charts are not supported
Licensing	Free for all	GPL & MIT	MIT only
Animation	Yes but turned off by default	Yes	Additional code required
Legend	Yes	No	No
Zooming/Scolling	In some chart designs only	Only Zooming	No
Multiple Axis	Yes	Yes	No
Number formatting	Yes	No	No
Trendlines	Yes	Yes	No
Ability to export chart as jpg or png	No	Partially through Canvas	No
Javascript events exposed for data	Yes	Yes	Not native
Server side API for easy integration	No	No	No

Chart or Graph. In all charts except the Scatter Chart these points have no size that is they are zero sized but in this chart you can control their size and shape depicted in Fig.7 where dents denote rise or fall .



**Fig.7** Dents in a Scatter Chart.

**2.3 Comparative Study of Google Charts and current tools in the market.**

It is shown in table 1.

**3. Google Maps integrated with Google Fusion Tables**

Mapping used to be a really hard task on the web. Then Google Maps were introduced and blew away every preconceived notion of how an online map should operate and look like. Soon after, Google released its Maps API, which allowed any developer to embed maps in their own sites. Since then, the mapping tools have developed a great deal. There are now many choices out there for users looking to embed custom mapping solutions in their own

data visualization project. Google Maps API uses a set of JavaScript classes that let users customize and embed

Google Maps in their web pages . It can be used to visualize data points related to Geo locations or Distance Matrix . Most of the inner features related to data manipulation are similar to Google Visualization API. (Ying Zhu, 2012)

Users can also make queries which are of a spatial nature by using Google Fusion Tables along with the Map API . Google Fusion Tables offers a tool for data users to easily upload, query and share data sets. Users can also create data visualizations easily and share them on the Web. The information provided by a google Map at any instant is an overlay of multiple layers , that is Streets , Satellite and terrain maps .Any query result that is to be displayed on a map is represented as such layers . The front end conveys the query to the backend which creates tiles putting together tiles based on current zoom levels and coordinates and returns this tile as an output .( Jayant Madhavan , et al,2012)

Therefore by adding data points to the Google Fusion Table one can select rows which can be mapped or choose a dataset from Google’s Public Data Set and those to the tables , one can create a data source which can be mapped. Now data source may be statistics or population or most frequented joins or weather conditions of a particular region . Hence one can choose the column , then the type of Map and then the region in the map to showcase data visualizations. In fig 8. population of Indian cities was added . The map table was manipulated to show the most populous cities of India. Thus Data Visualizations can be easily implemented with the help of Google Maps Integrated with Google Fusion Tables. Google Maps API is also well documented. There’s also an active user forum for questions and answers. Google Maps API is a free service but has many technical restrictions. For example, a restriction exists on the number of queries that can be submitted by the user to Google Maps services per day and per second. Users can raise this quota with paid subscriptions, but the vast majority of users rarely reach the default quota which is

25000. Hence for normal users of web can easily use these tools daily but companies require a paid subscription .

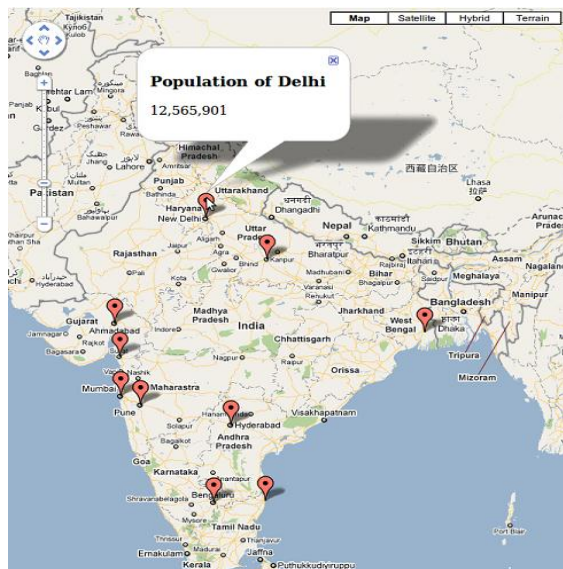


Fig.8 Google Maps API for populous cities in India

3.1 Google Maps API application

The best example of a Visualization chart map using fusion tables is FLU activity tracker. It was a project initiated by Google & CDC . The data sources for the maps and charts were populated using Google search history, public health report and online medical purchases. It lets you track if your city or street or state has a high risk for the common flu. Its picto-graphical representation is given in fig. 9.

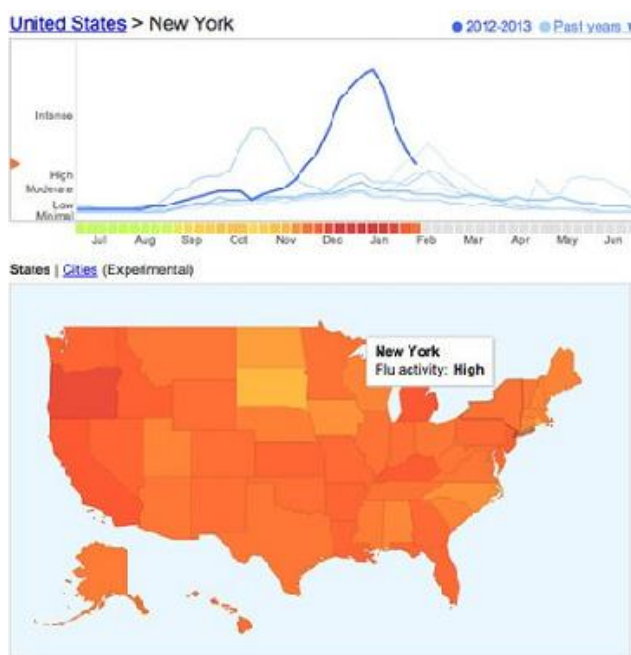


Fig.9 Google Maps Flu Tracker

Conclusions

A problem with Google visualization API is that they are generated on the client side, which creates issues for devices without JavaScript and also for offline use . The users' computers must have access to a library in order to utilise the interactive features of Google Charts. This is because the visualization foundations that charts requires are loaded dynamically before they can be utilised. The terms of service do not allow the downloading of the google.load or google.visualization codes to use offline. Hence to use them offline screenshots of the charts or Maps can be used. Static images didn't have the same issues but they have been deprecated. But it's quite easy to turn this into a server side .NET control that would allow you to just pass the variables to a function and let it render the correct URL for your requirements.

But the best thing about google visualization tools is the reputation associated with Google and also that it is constantly updated and checked for bugs . The most recent update for Google Visualization API was 25th June 2014. The ease of use and the fact that it is free makes this a viable option for all but the most complex charting and graphing requirement. Google Charts & Google Maps connect to data in real time, are free to use , and well documented by Google Developers and Online Google Maps community & hence makes them both a great tool to utilize. (Ying Zhu, 2012)

References

Ying Zhu , (2012), Introducing Google Chart Tools and Google Maps API in Data Visualization Courses *IEEE Computer Graphics and Applications* , 6-9

F Viegas et al., (2007) , Many Eys : A Site for Visualization at Internet Scale, *IEEE Trans. Visualization and Computer Graphics* , vol. 13 ,1121-1128

Hector Gonzalez, Alon Y. Halevy, Christian S. Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley, Warren Shen, Jonathan Goldberg-Kidonhttp,(2010) Google Fusion Tables: Web-Centered Data Management and Collaboration *Sigmond* ,505-516

Jayant Madhavan, Sreeram Balakrishnan, Kathryn Brisbin, Hector Gonzalez, Nitin Gupta, Alon Halevy, Karen Jacqmin-Adams, Heidi Lam, Anno Langen, Hongrae Lee, Rod McChesney, Rebecca Shapley, Warren Shen, (2012),Big Data Story Telling through Interactive Maps , *IEEE Computer Society Technical Committee on Data Engineering* ,10-17

B.B. Bederson, J. Grosjean, and J. Meyer, Toolkit Design for Interactive Structured Graphics, *IEEE Trans. Software Eng.* , vol. 30, no. 8, 2004, pp. 535–546.

C. Stolte, D. Tang, and P. Hanrahan, Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases, *IEEE Trans. Visualization and Computer Graphics*, vol. 8, no. 1, 2002, pp. 52–65.