

Mobile Data Visualisation

Group 3

Miran Levar, Johanna Pirker, Bernd Prünster and Werner Sturm

706.057 Information Visualisation SS 2014
Graz University of Technology

21 May 2014

Abstract

The survey explores the state of mobile data visualisations by examining eight popular visualisation libraries that render data on websites using HTML5 and JavaScript. Each library has its features reviewed, especially how it renders data - whether the visualisations are responsive, their usability on mobile devices, and whether their rendering engines support large datasets. Each library also has its user- and development experiences evaluated. For comparison purposes the libraries are in addition to the the showcase examples also used to visualise two datasets - the ACM Computing Classification System hierarchy and the results of the 2013 Austrian elections in Graz. Results of the library examinations are aggregated in a comparison table, along with the average marks the surveyors have given each of them.

Additionally the survey looks into the problems and best practices of designing responsive visualisations.

Keywords: data visualisation, mobile data visualisation, responsive design, JavaScript, HTML5

Contents

1	Introduction	1
1.1	Study Setting and Method	1
2	Visualisation Libraries	3
2.1	Chart.js	4
2.2	ChartJS	7
2.3	Data Driven Documents (D3)	11
2.4	Highcharts	14
2.5	Dojo	16
2.6	Kendo UI	18
2.7	Google Charts	21
2.8	Sencha Touch Charts	25
2.9	Other Libraries	27
3	Responsive Data Visualisation	29
3.1	Responsive Web Design	29
3.2	Responsive Data Visualisation Problems	29
3.3	Responsive Data Visualisation Best Practices	30
3.3.1	Variable Level of Detail	31
3.3.2	Compound Overview-Detail Visualisation	32
3.3.3	Modular Visualisations	32
3.3.4	Responding to the Client	33
3.3.5	Unresponsive Data	33
4	Concluding Remarks	35
4.1	Evaluation Results	36
	Bibliography	43

Chapter 1

Introduction

When choosing a technology for implementing a mobile application there are essentially two possibilities: using platform native technologies or employing web technologies. While applications based on native technologies integrate better, relying on web technologies makes it possible to use a single code-base on all targeted platforms. The same is true when it comes to data visualisation frameworks. Furthermore HTML and JavaScript are on the rise even on the desktop with Microsoft treating them as first-class citizens for developing native Windows 8 applications [Dewey, 2012]. For this reason only libraries employing web technologies were considered for this survey.

1.1 Study Setting and Method

This study focuses on comparing the visualisation libraries using pre-defined quality criteria. A list of criteria was defined as a basis for the comparison and the evaluation. These criteria include objective aspects such as rendering techniques, mobile input support, quality of tutorials and documentation and supported visualisation types, and also subjective elements such as user experience, programmer experience, visual appeal, mobile experience, and versatility. For the evaluation of the subjective aspects a Likert-scale between 1 and 5 (5 being best) was used and the 4 authors of this papers acted as evaluators of each library. To test and compare the visual outcomes and the programming experience of each library two predefined test datasets were used. The first dataset contains Graz election data from 2013 from <http://www.data.gv.at/> [Bundeskanzleramt, 2014], consisting of party names and vote numbers for each district in Graz (see Figure 1.1). The data should be represented with chart types such column charts, line charts, area charts, or pie charts. The data was downloaded as a csv file and converted to a json file. The second dataset contains the hierarchical data of the ACM classification system represented in a XML structure, which can be represented with charts such as tree maps or trees (see Figure 1.2). This dataset is retrieved from [ACM, 1998].

```

"1": {
  "bezname": "Innere Stadt",
  "ptlang": "Sozialdemokratische Partei Oesterreichs",
  "ptname": "SPOe",
  "sprengelanzahl": "4",
  "beznr": "0100",
  "stimmen": "200",
  "gesamt": "1590",
  "unguel": "18",
  "wkurz": "NR2013",
  "listenplatz": "1",
  "gueltig": "1572"
},

```

Figure 1.1: Example of a data entry of the Graz 2013 election data from [Bundeskanzleramt, 2014]

The ACM Computing Classification System (1998)

- A. General Literature
 - A.0 GENERAL
 - *Biographies/autobiographies*
 - *Conference proceedings*
 - *General literary works (e.g., fiction, plays)*
 - A.1 INTRODUCTORY AND SURVEY
 - A.2 REFERENCE (e.g., dictionaries, encyclopedias, glossaries)
 - A.m MISCELLANEOUS
- B. Hardware
 - B.0 GENERAL
 - B.1 CONTROL STRUCTURES AND MICROPROGRAMMING (D.3.2)
 - B.1.0 General
 - B.1.1 Control Design Styles
 - *Hardwired control* [**]
 - *Microprogrammed logic arrays* [**]
 - *Writable control store* [**]
 - B.1.2 Control Structure Performance Analysis and Design Aids
 - *Automatic synthesis* [**]
 - *Formal models* [**]
 - *Simulation* [**]
 - B.1.3 Control Structure Reliability, Testing, and Fault-Tolerance [**] (B.8)

Figure 1.2: Sample of the ACM classification dataset from [ACM, 1998]

Chapter 2

Visualisation Libraries

2.1 Chart.js

Chart.js is a lightweight open-source library for drawing object-oriented client side graphs.¹ It is developed under the MIT license and does not depend on any other library.

Website: <http://www.chartjs.org/>

Technology: Charts.js is completely implemented in JavaScript, conforms completely to the HTML5 standard and it uses the HTML5 canvas element to draw graphs.

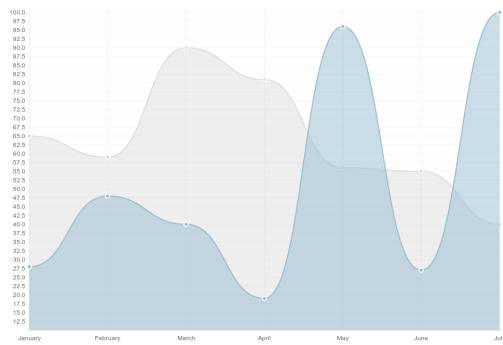
Platform support: Basically, this library is compatible with all modern and major mobile web browsers which support the HTML5 canvas element. It is recommended to use the polyfill *ExplorerCanvas*² for Microsoft Internet Explorer 8 and below.

Features: The library offers smooth animations to fade in graphs. Event handling (for example the *onclick* event) is not supported by this library. None of the generated charts adapt to different display sizes, so such additional functionality has to be implemented by developers.

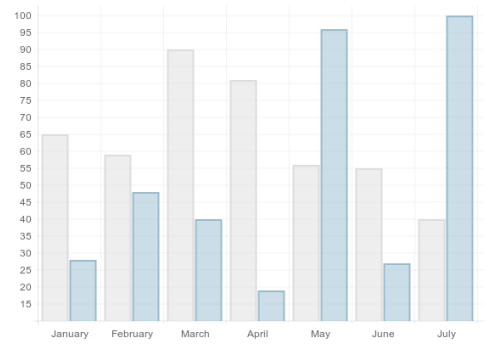
Experience: It is a very simple charting library which is easy to use. If just basic charts are needed it is a good choice, but if more features such as event handling are needed, considering a more advanced library is recommended.

¹All information about the technical details of this library was taken from [Downie, 2013] (if not specified otherwise)

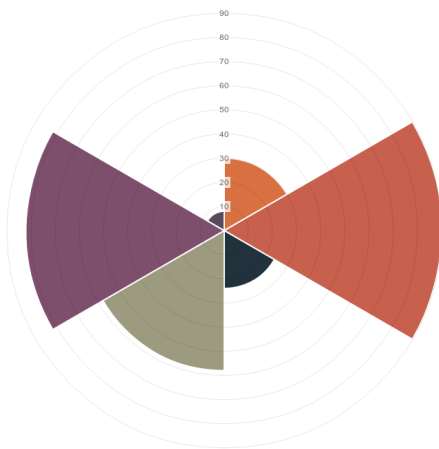
²<https://code.google.com/p/explorercanvas/>



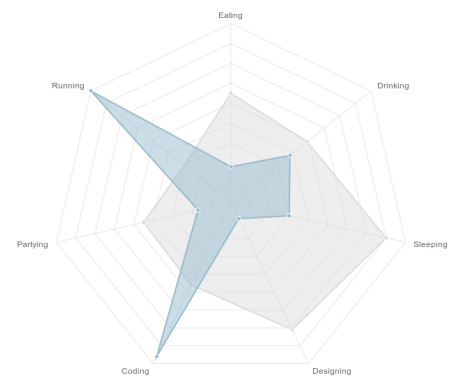
(a) Line chart



(b) Bar chart

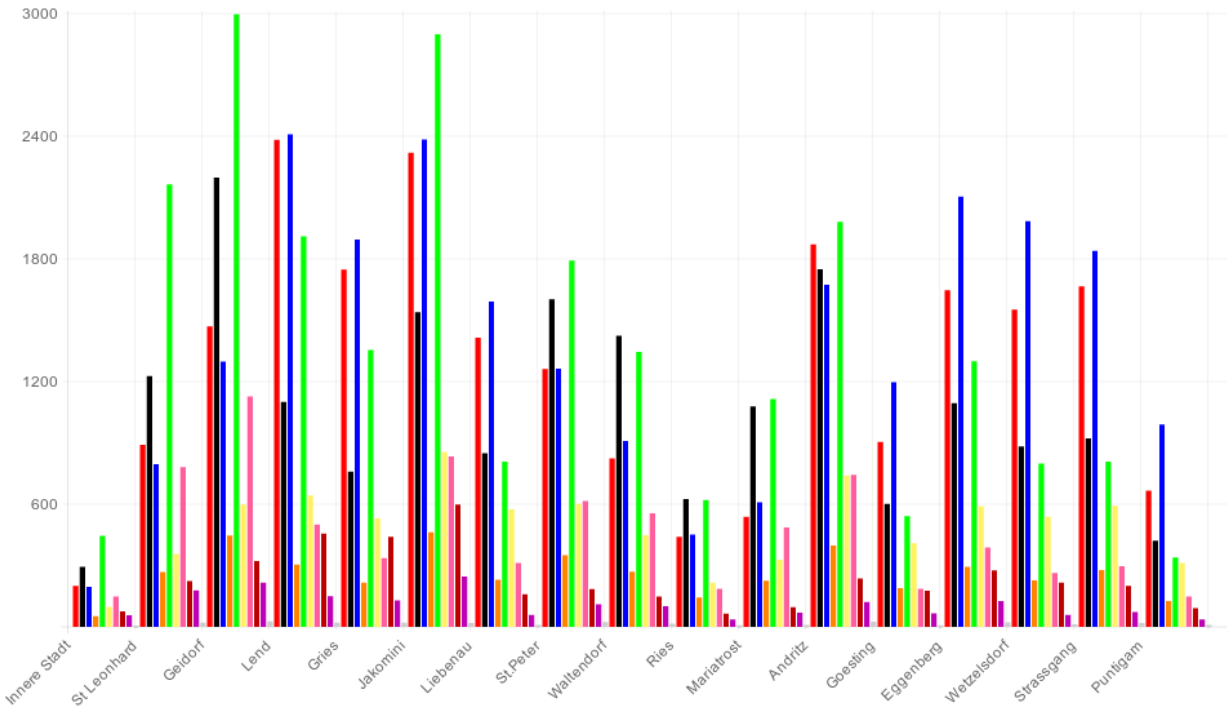


(c) Polar chart

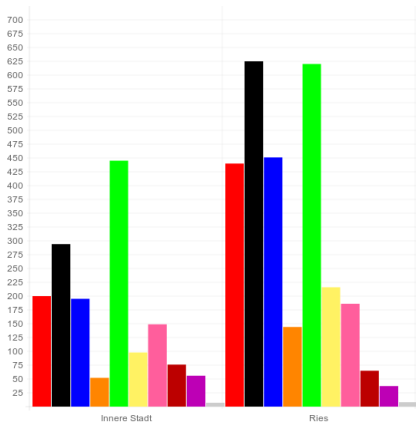


(d) Radar chart

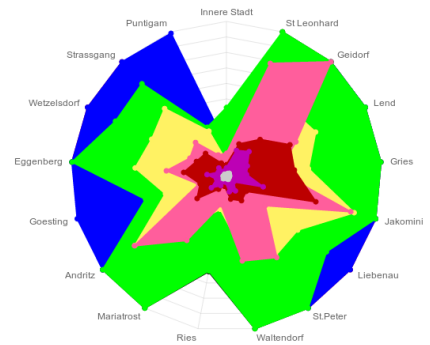
Figure 2.1: List of supported chart types from [Downie, 2013]



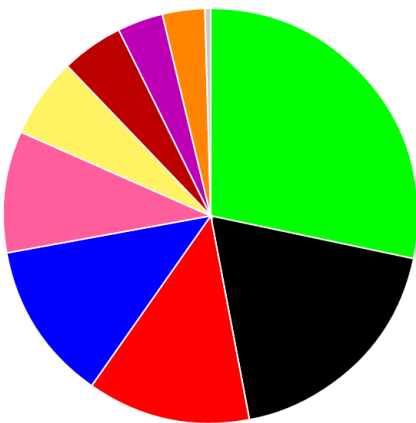
(a) Bar chart depicting votes of all districts



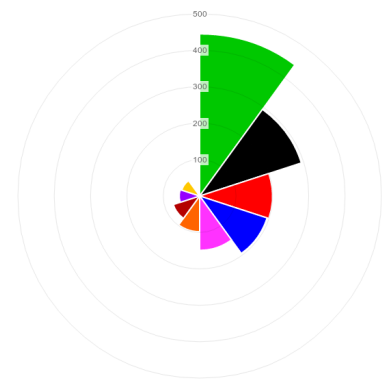
(b) Bar chart comparing two districts.



(c) Radar chart depicting votes of all districts.



(d) Pie chart depicting votes of district "Innere Stadt".



(e) Radar chart depicting votes of district "Innere Stadt".

Figure 2.2: Charts depicting data of election in Graz 2013.

2.2 ChartJS

ChartJS is an advanced library for drawing various graphs on heterogeneous devices such as desktop computers, tablets and mobile phones.³ It is free for non-commercial usage and it is developed by DevExpress⁴.

Website: <http://chartjs.devexpress.com/>

Technology: ChartJS is implemented in JavaScript and it conforms completely to the HTML5 standard. It depends on popular third-party libraries like *JQuery*⁵, *Knockout*⁶ and *Globalize*⁷. SVG is used to draw graphs.

Platform support: ChartJS is compatible with all modern and major mobile web browsers which support SVG.

Features: The library supports plenty of different chart types, range selectors, and even vector graphic based maps to realise various geographic visualisations. User interactions like scrolling, zooming, click handling, legend click, hover, and selection handling are supported as well. Charts can be combined (for example, a line chart and a bar chart) to visualise additional information in a single graph. In addition to that many charts are responsive to different display sizes.

Experience: This library works very well out-of-the-box. All charts can be adjusted to different needs and detailed information for developers is available on the website. The most noteworthy feature is the high responsiveness of many charts. Developers do not need to extend these charts to make them adjust to different display sizes.

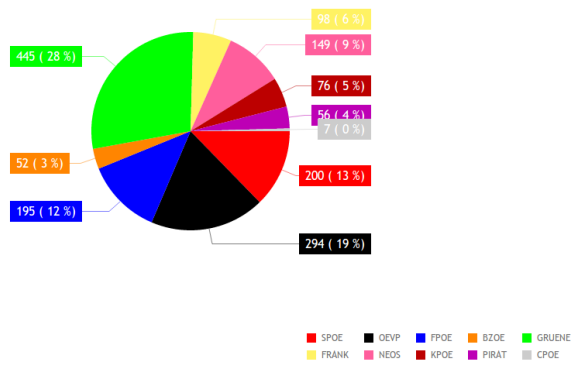
³All information about the technical details of this library was taken from [DevExpress, 2013] (if not specified otherwise)

⁴www.devexpress.com

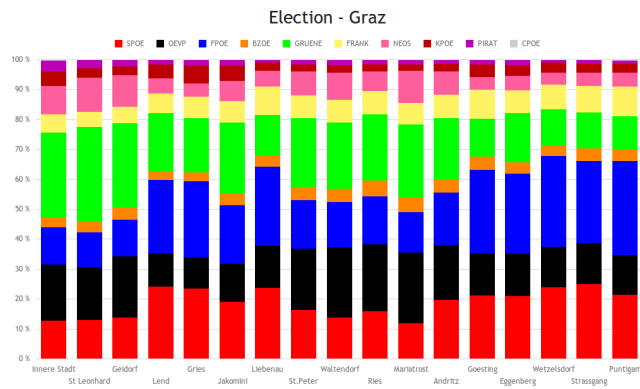
⁵<http://jquery.com/>

⁶<http://knockoutjs.com/>

⁷<https://github.com/jquery/globalize>

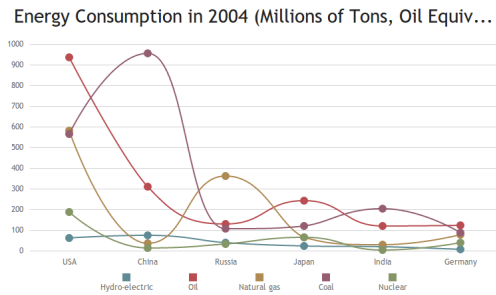


(a) Pie chart

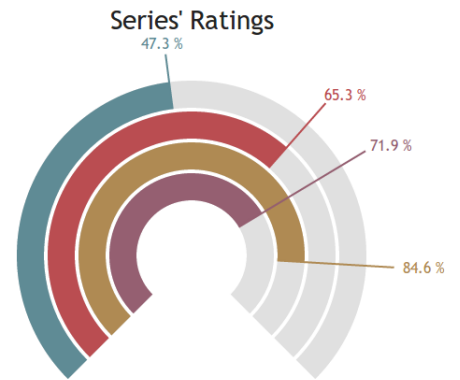


(b) Stacked bar chart

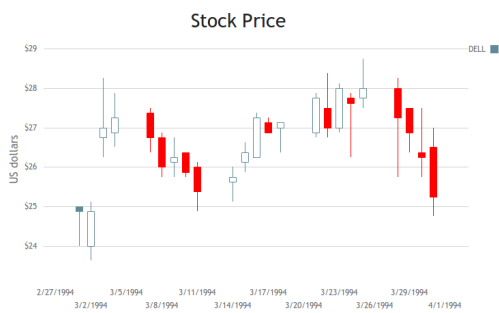
Figure 2.3: Pie chart and stacked bar chart depicting election data.



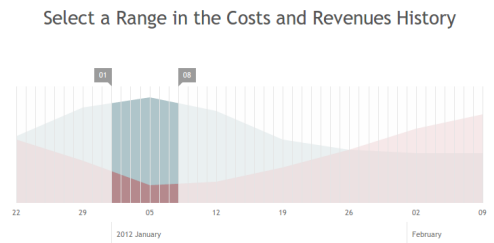
(a) Cross Hair - Line Chart



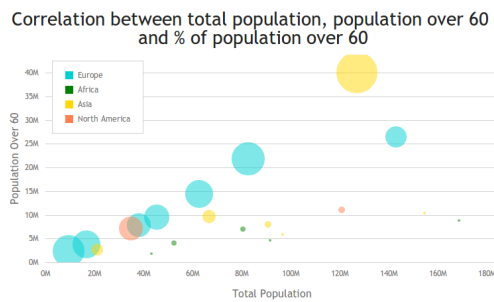
(b) Bar Gauge



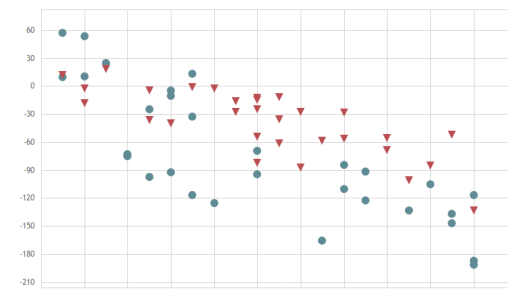
(c) Candle Stick



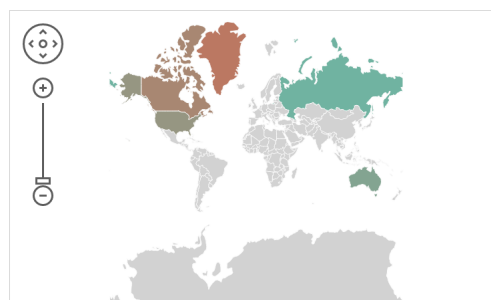
(d) Range Selector



(e) Bubble Chart

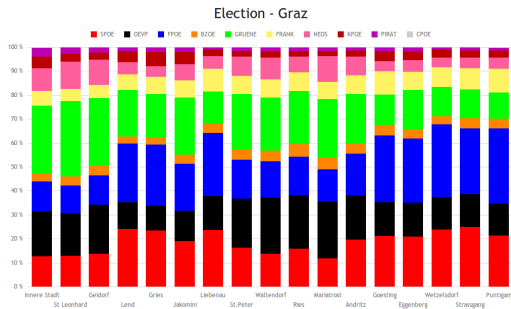


(f) Scatter Plot

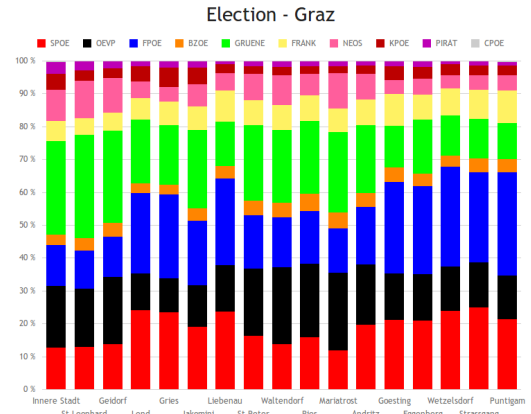


(g) Vector Map

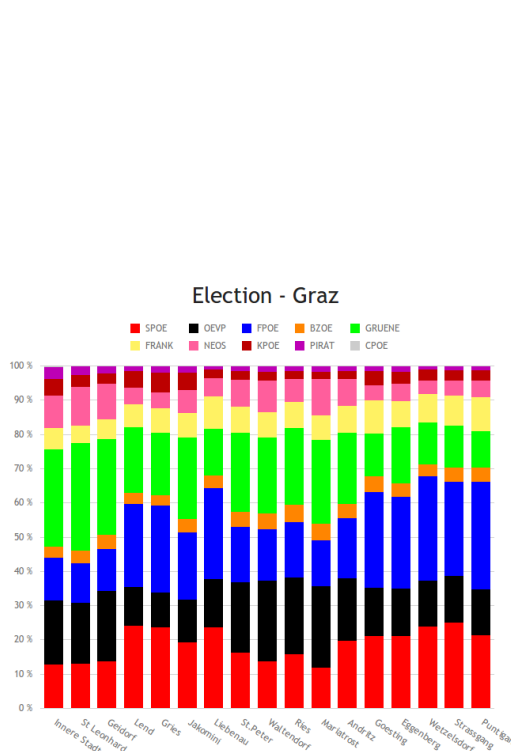
Figure 2.4: Examples of chart types supported by ChartJS from [Downie, 2013].



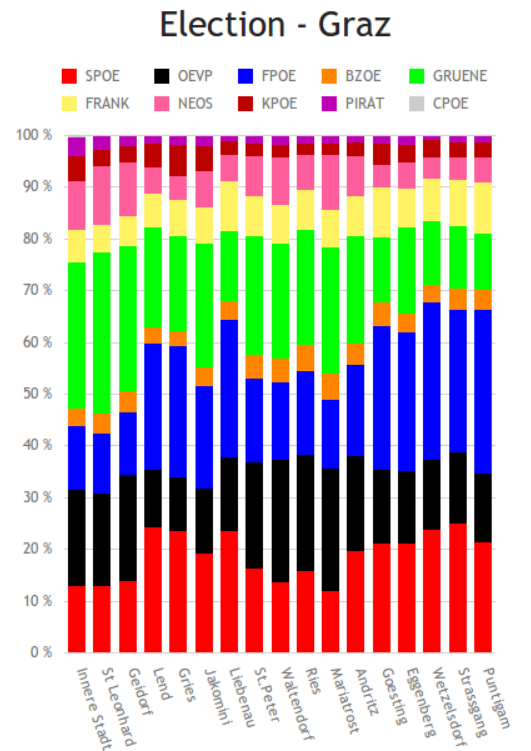
(a) Stacked bar chart.



(b) Labels are closer to each other.



(c) Legend is organised in two rows and labels are reoriented to save space.



(d) Labels are oriented almost vertically.

Figure 2.5: Charts adjust to different screen sizes dynamically (responsive).

2.3 Data Driven Documents (D3)

Data-Driven Documents

As the successor to the *Protovis*⁸ framework, *D3* is a generic JavaScript library available under a BSD licence to manipulate documents based on data.⁹ Its data-driven approach to DOM manipulation allows developers to realise complex visualisations.

Website: <http://d3js.org/>

Technology: D3 uses HTML5, SVG and CSS3 to draw graphics. It dynamically generates and styles SVG objects in the DOM tree. There exists a proof-of-concept implementation which also use canvas to improve performance by [Bostock, 2011b](#).

Platform support: This library is compatible with all modern and major mobile web browsers which support SVG in DOM.

Features: D3 is known for smooth animations and visually complex data visualisations. Due to its generic data-driven approach is it very flexible. There already exist more than 1900 examples on the community's website¹⁰. D3 offers interfaces for defining transitions, styles and mappings between data and its visualisation. Unfortunately, most of available demos are not responsive and they do not adjust to different display sizes automatically but it is possible to fix this issue by using CSS when the visualisation does not use fixed pixel sizes.

Experience: Most of available demos are very easy to use but even slight modifications need a deep insight into library. So if an example fits specific demands, D3 a good choice to visualise data in a modern way.

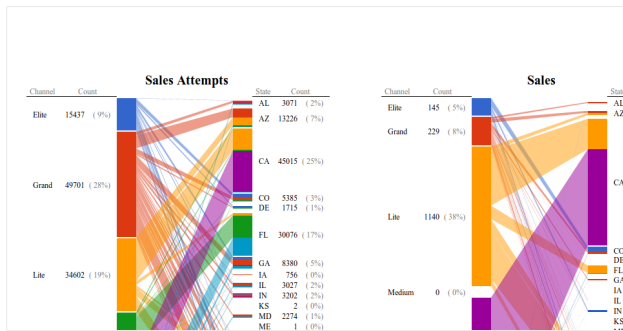
Capabilities of D3 enable developers to create sophisticated visualisations but a significant drawback of D3 are performance issues when using larger data sets. The reason for its declining performance is the usage of SVG objects. D3 manipulates SVG objects to generate animations which results in a high amount of DOM operations.

⁸<http://mbostock.github.io/protovis/>

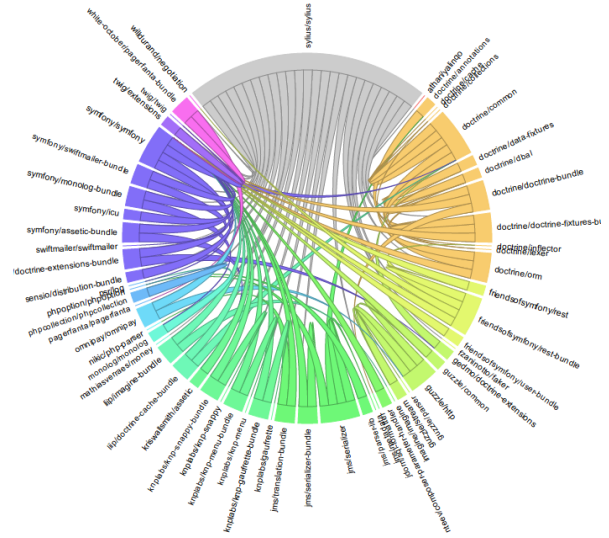
⁹All information about the technical details of this library was taken from [[Bostock, 2011a](#)] (if not specified otherwise)

¹⁰<https://github.com/mbostock/d3/wiki/Gallery>

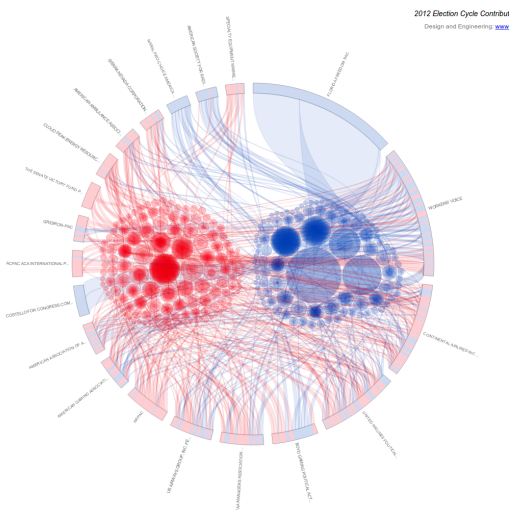
BiPartite



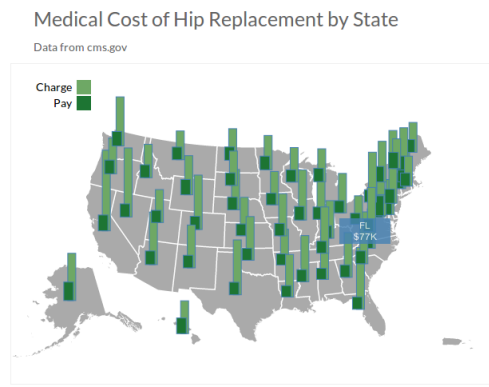
(a) BiPartite



(b) Dependency Wheel



(a) Political influence



(b) Medical cost

Figure 2.6: Examples of visually complex visualisations realised with D3 from [Bostock, Mike et al., 2014]

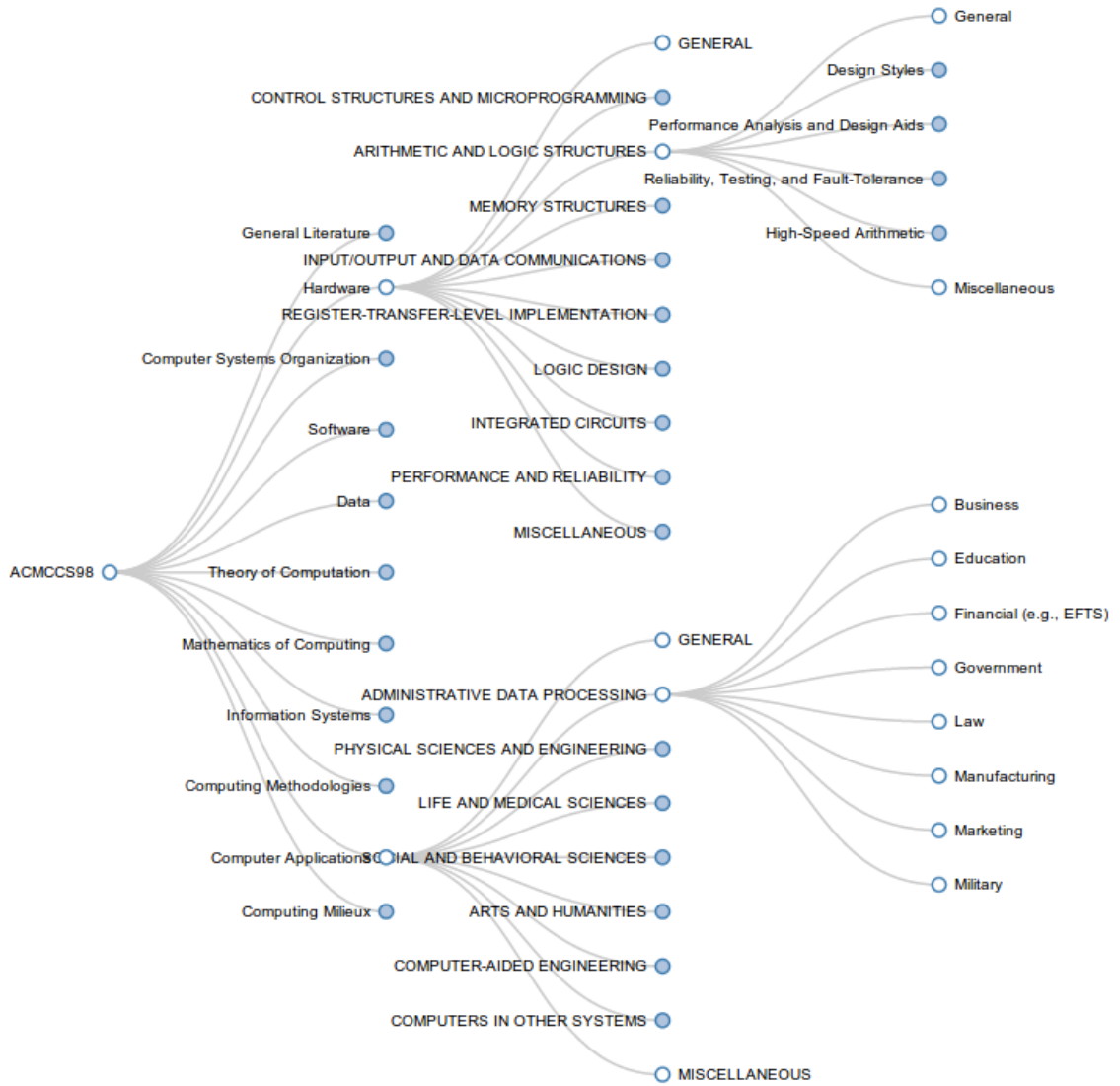


Figure 2.7:
Interactive visualisation of ACM hierarchy.

2.4 Highcharts

Highcharts is a JavaScript charting library developed by Highsoft AS.¹¹ Its intended use is purely client-side dynamic visualisation of data inside web pages (and web applications). The Highcharts library is split up into three separate modules, each focusing on different visualisation types. They are not capable of visualising hierarchical data.

Highcharts is intended to render a wide variety of charts, at the time of this writing these include: line charts, spline charts, area charts, bar charts, pie charts, scatter plots, angular gauges, box plots, and polar charts.

Highstock is aimed at creating time-line charts having features specific for this purpose such as a date picker and zooming and panning options well adapted for time-line data.

Highmaps targets solely map-based visualisations and therefore includes features such as mouse-wheel-zooming and navigation buttons and other elements that are commonly associated with map-based visualisations. At the time of this writing Highmaps is still in beta.

Website: <http://highcharts.com>

Technology: Highcharts uses JavaScript to render to SVG. Highcharts does not require any browser plugins such as Adobe Flash, but rather relies on modern web technologies. However, supporting libraries (one of: *jQuery*, *MooTools*¹², *Prototype*¹³) are required.

Platform Support: The library itself, as well as the charts created using it are cross-browser compatible, supporting also smartphones as well as Microsoft Internet Explorer down to version 6.

Experience: Rendered charts can be directly printed and exported to PDF, SVG, PNG, and JPEG and the dynamic charts adapt to the viewport size. However, this dynamic scaling is accomplished through absolute coordinates and JavaScript functions that recalculate them upon changing viewport size. This results in both mobile- and desktop-friendly charts featuring touch-friendly controls (not relying on hover elements) and compound overview-detail charts which are all the more useful on small-screen devices. The input data has to be provided in JSON format or an URL pointing to a resource that is transmitted as JSON-encoded data.

The Highcharts website provides more than one hundred examples on how to use the library. This along with an extensive API documentation that is again supported by example code snippets results in a gentle learning curve for new developers and makes Highcharts easy-to-use in general.

When using combined bar charts, stacked bar charts, or other visualisation types that use systematic grouping of data points the format of the input data is counter-intuitive. One would imagine that when using a two-dimensional array to model such a data set, that each group (each stacked bar) would be contained in one array. However, the exact opposite is the case. Figure 2.9 shows a stacked bar chart of the tabular data set used for this survey, while Listing 2.1 illustrates the format of the data.

Another issue is the way colours are assigned to categories (slices of a pie chart, columns of a bar chart): Colours need to be specified globally and cannot be specified as part of the input data. If more categories are present than colours specified, colours are reused in the order they were defined. This property in conjunction with the counter-intuitive format used for stacked and grouped bar charts renders these specific chart types prone to programming and data preprocessing errors and generally not very easy-to-use. Since hierarchical data cannot be visualised using Highcharts, the creation of a visualisation of the hierarchical data set was not attempted.

¹¹All information about the technical details of this library was taken from [Highsoft AS, 2009] (if not specified otherwise)

¹²<http://mootools.net/>

¹³<http://prototypejs.org/>

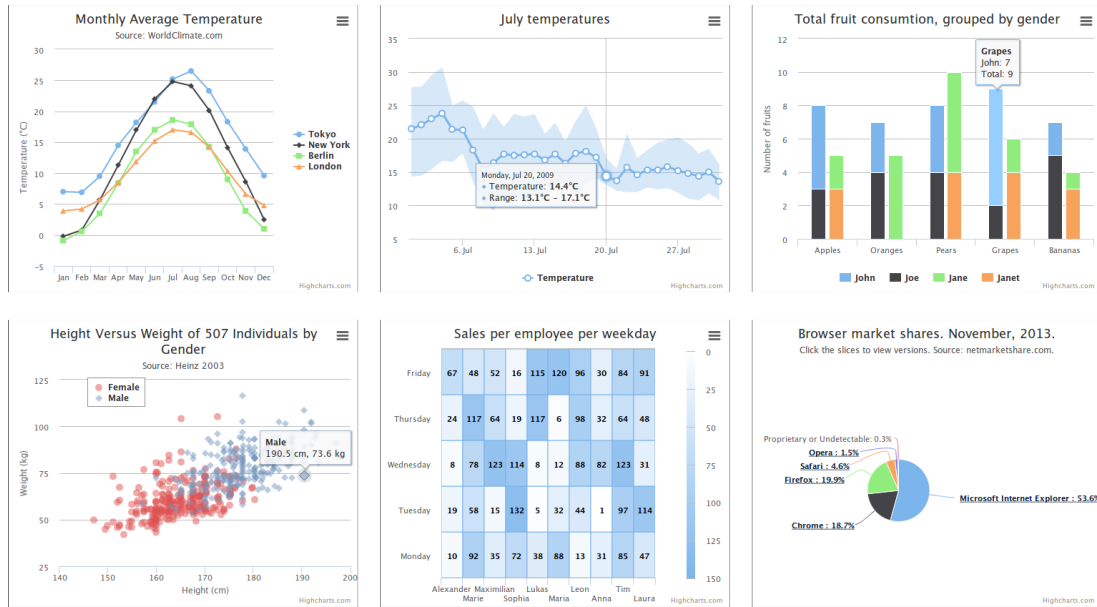


Figure 2.8: Examples of some of the supported chart types from [Highsoft AS, 2009]

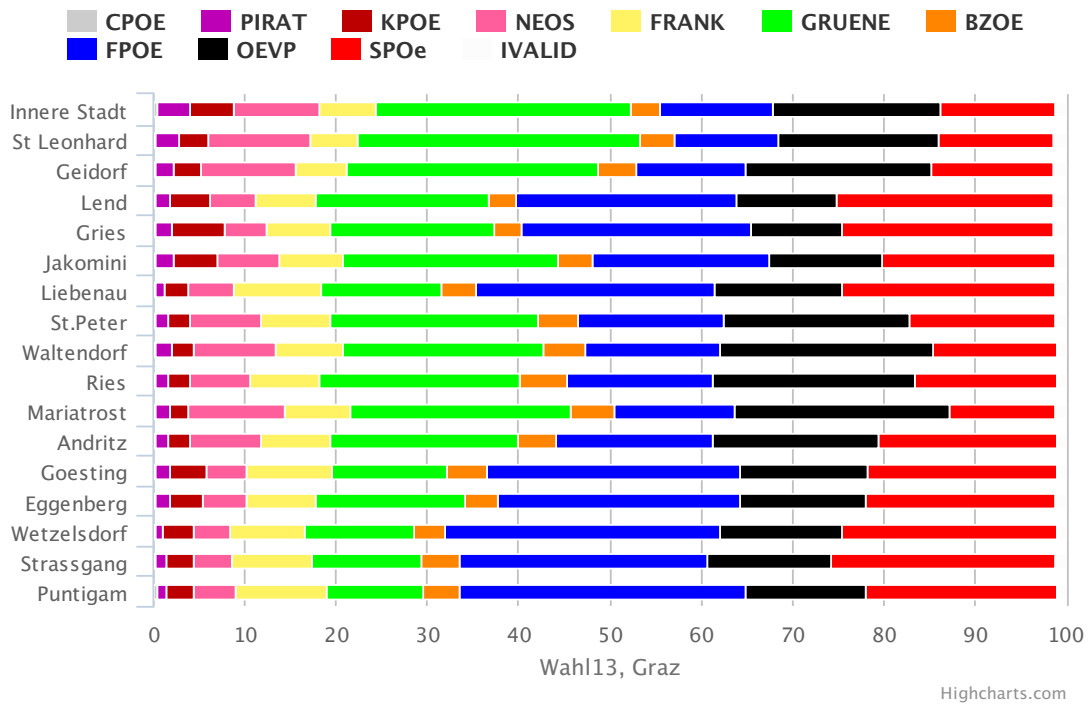


Figure 2.9: Stacked bar chart of the 2013 Graz election data

```

1  [{ "name": "INVALID", "data": [ 18, 94, 141, 134, 100, 154, 74, 99, 66, 28, 54, 105, 39,
2  101, 71, 82, 29 ] },
3  { "name": "SPOe", "data": [ 200, 891, 1470, 2383, 1748, 2320, 1414, 1262, 824, 440,
538, 1871, 904, 1646, 1552, 1665, 667 ] },
4  { "name": "OEVP", "data": [ 294, 1227, 2198, 1101, 759, 1540, 849, 1603, 1424, 625,
1079, 1750, 601, 1094, 882, 922, 421 ] }, ...
    
```

Listing 2.1: The input data for the stacked-bar chart shown in Figure 2.9. The data is grouped by party, not by district.

2.5 Dojo

Dojo Toolkit, developed by the Dojo Foundation, is an open-source JavaScript library that aims to ease development of cross-platform, JavaScript/Ajax-based applications and web sites¹⁴. The data visualisation part of the tool-kit is *Dojo GFX*, a high level graphics API. The tool-kit is dual-licensed with the Modified BSD License (BSD) or the Academic Free License version 2.1.

Website: dojotoolkit.org

Technology: Dojo GFX supports a range of rendering technologies out of the box: SVG, VML, Canvas, Silverlight, and SVGWeb. It selects the best graphics engine based on the browser capabilities. When a web-page is loaded GFX dynamically calculates the dimensions of the available surface and then renders the graphics. This means that the visualisations are not responsive out of the box and require a redraw after dimensions of the container change.

Platform support: The library supports all major desktop and mobile browsers, including legacy support ranging back to Internet Explorer 6.

Features: The library supports the common two-dimensional chart types ranging from bar and pie to candlestick, spider, and area charts (some example can be seen on Figure 2.11). Additionally it includes gauges and geographic visualisations. The latter is of special note because the library supports inclusion of OpenLayers¹⁵ map data and also its own maps built with vector graphics.

All supported visualisations are built with the GFX library and can because of that be heavily modified. Developers can also use the underlying library to construct their own visualisations.

Experience: From the developers perspective using the Dojo toolkit can be frustrating. There are few examples of visualisations, because of that progress is made by exploring the at times poor documentation. Simple changes like switching colours have relatively straightforward solutions but the discovery of these solutions can be time consuming. The lack of a built-in solution for responsiveness means that the visualisations require changes to perform as is expected of modern ones.

The user experience of viewing and interacting with visualisations will mostly depend on the skill of the developer as the out-of-the-box solutions can neither be considered eye-catching nor impressive. Consider Figure 2.10 as an example of both ends of the skill gap.

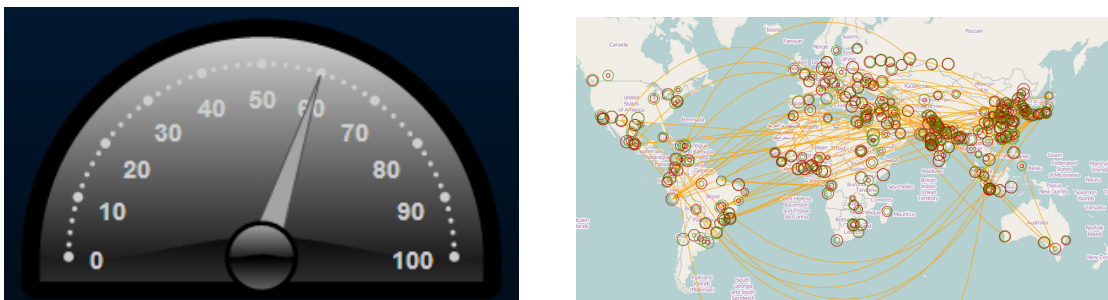


Figure 2.10: Example of a poor and a reasonably good visualisation with the Dojo toolkit (examples selected from [The Dojo Foundation, 2004]).

¹⁴All information about the technical details of this library was taken from [The Dojo Foundation, 2004]

¹⁵openlayers.org

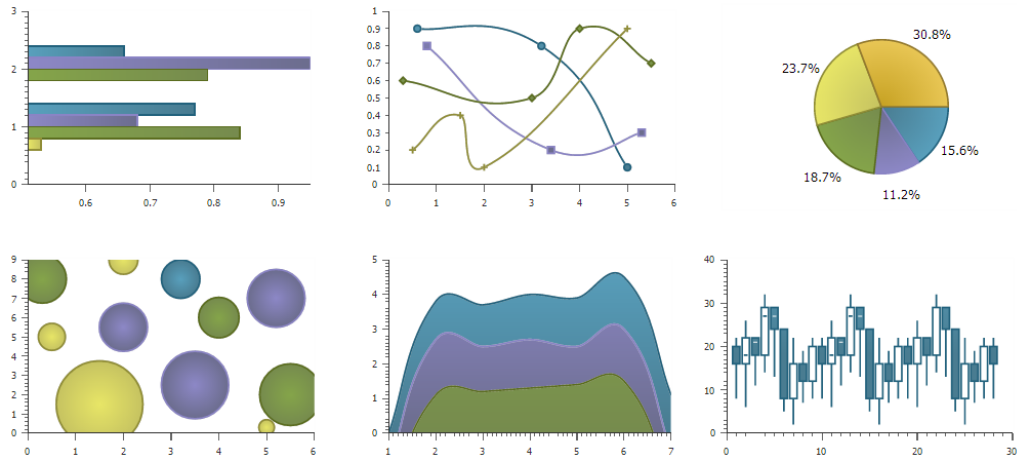


Figure 2.11: Examples of some of the supported chart types (images taken off [The Dojo Foundation, 2004]).

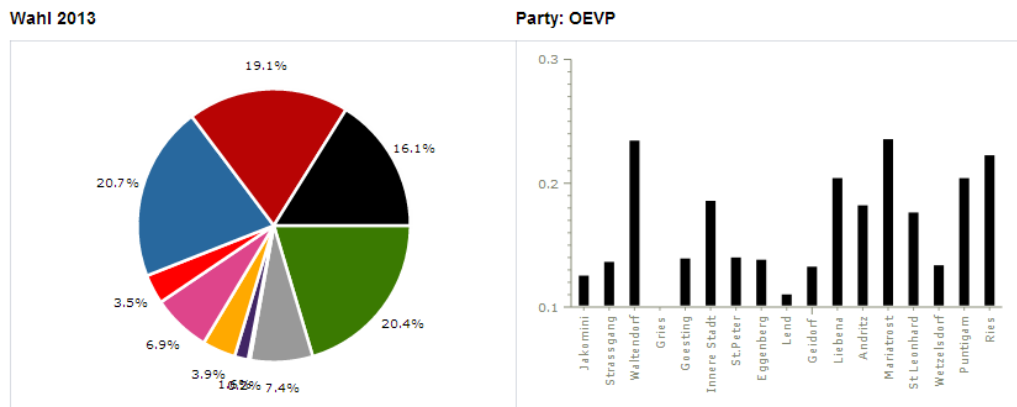


Figure 2.12: Example visualisation of the 2013 Graz election data.

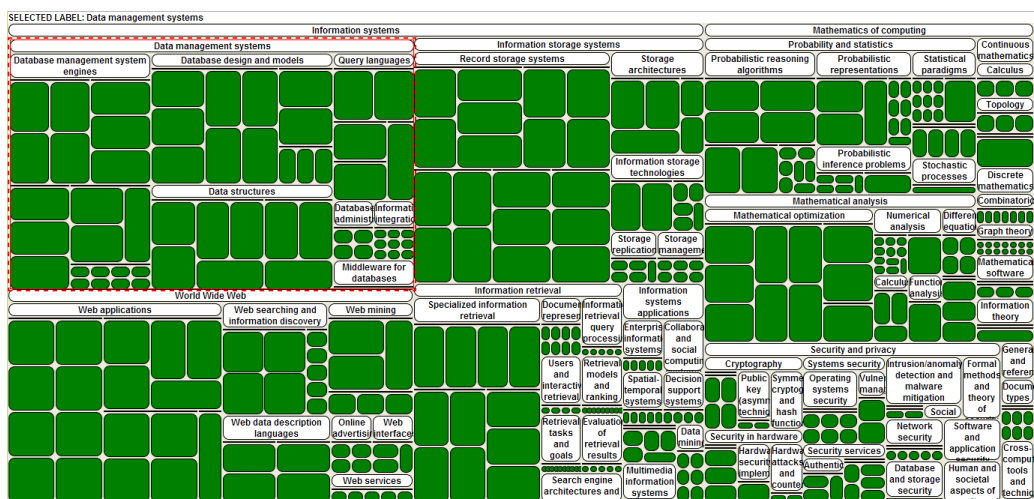


Figure 2.13: Example of a hierarchical visualisation of the ACM hierarchic data with the experimental tree-map visualisation.

2.6 Kendo UI

Kendo UI, developed by Telerik¹⁶, is a commercial JavaScript and HTML5 framework that provides developers with widgets and features for rapid development across different browsers and platforms¹⁷. Besides the payable full developer selection of widgets and features Telerik also offers a subset of them under the open source Apache v2 license - *Kendo UI Core*. The data visualisation portion of Kendo UI is *Kendo DataViz* and none of its components are included in the open source Core.

Website: www.telerik.com/kendo-ui

Technology: Kendo UI is based on *jQuery* and renders graphics using SVG, VML, and Canvas, the default is SVG. When a web-page is loaded Kendo dynamically calculates the dimensions of the rendering surface and then renders the graphics. This means that despite advertising responsiveness the *DataViz* visualisations are not actually responsive out of the box, that only applies to user interface components.

Browser support: The library supports all major desktop and mobile browsers, including legacy support ranging back to Microsoft Internet Explorer 7.

Features: Kendo DataViz supports the common chart types, examples of which can be seen on Figure 2.15. It also supports gauges and geographic visualisation (example can be seen at Figure 2.14). The map widget supports *Open Street Map* data, *GeoJSON*, and more.

All visualisations support touch-based control on mobile devices.

Experience: Developers should enjoy using Kendo UI. Demonstrations are provided for all supported visualisations and they cover most use-cases developers should encounter, in addition non-stop support support is provided for paying customers. The documentation is comprehensive and reasonably well structured, it made writing small tweaks to demonstrational examples easy. Despite everything being polished the library is quite limited. It allows for basic visualisations but is not really meant to be expanded into very complex and dynamic ones. The aim of the library can be seen in one of the examples Telerik provides, a dashboard for displaying stock data (see Figure 2.16).

The user experience should be very positive, because of their simplicity the visualisation work well - they seem both professional and still eye catching.

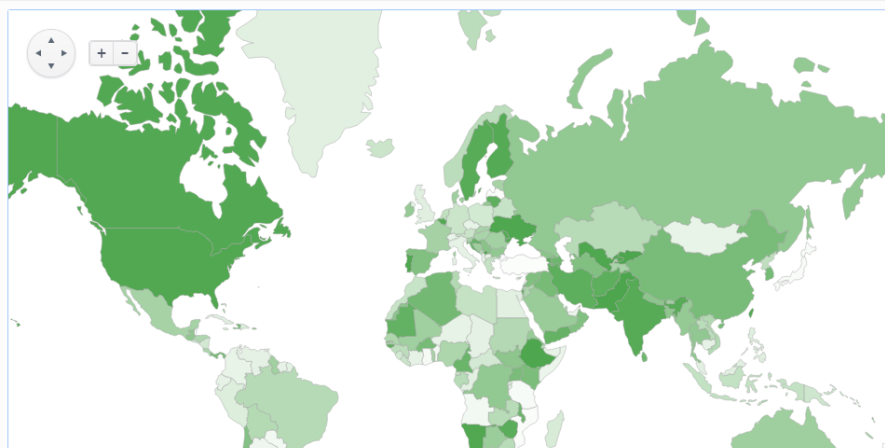


Figure 2.14: Example geographical visualisation (taken off www.telerik.com/kendo-ui).

¹⁶www.telerik.com

¹⁷All information about the technical details of this library was taken from [Telerik, 2014]

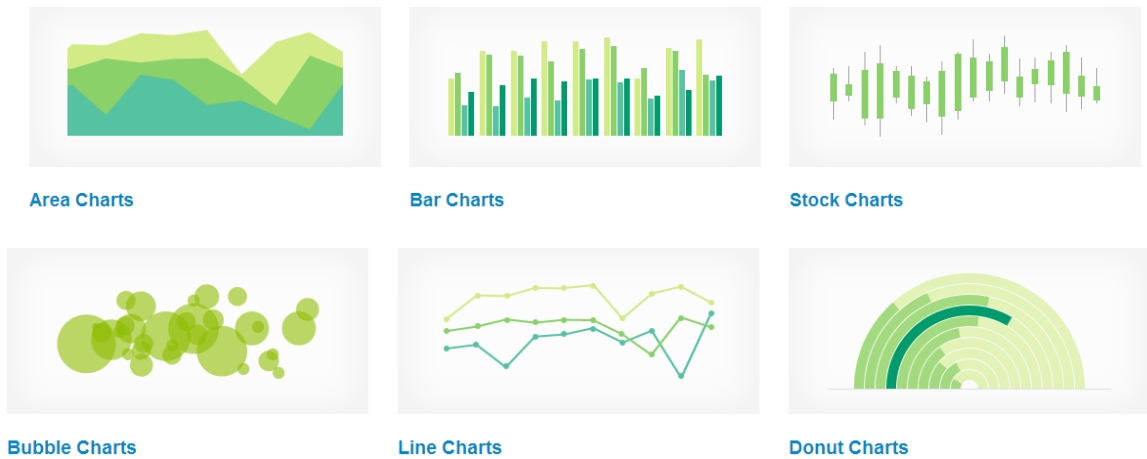


Figure 2.15: Examples of some of the supported chart types (taken off [Telerik, 2014]).

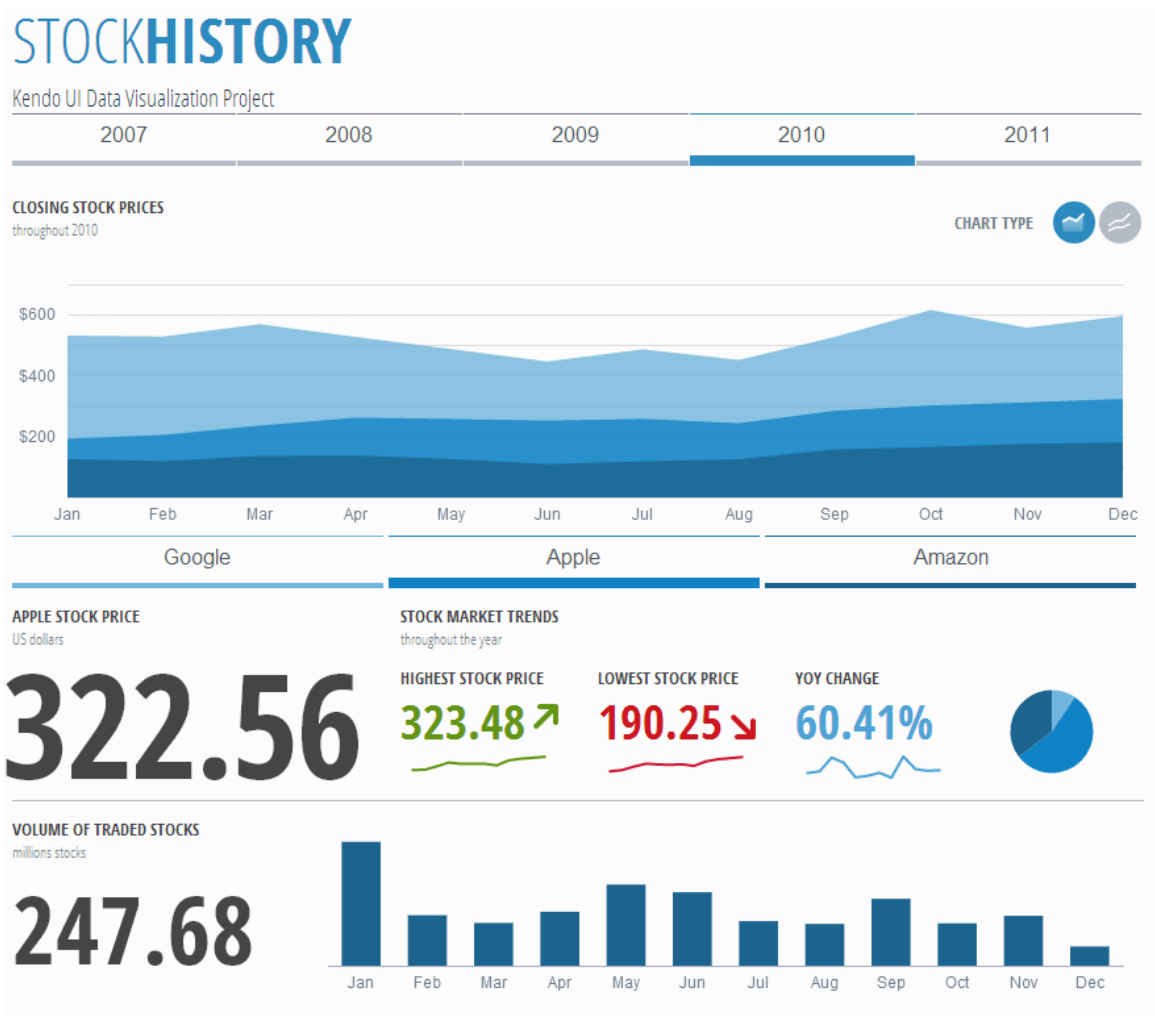


Figure 2.16: An example of how the developers see the library used well (example from [Telerik, 2014]).

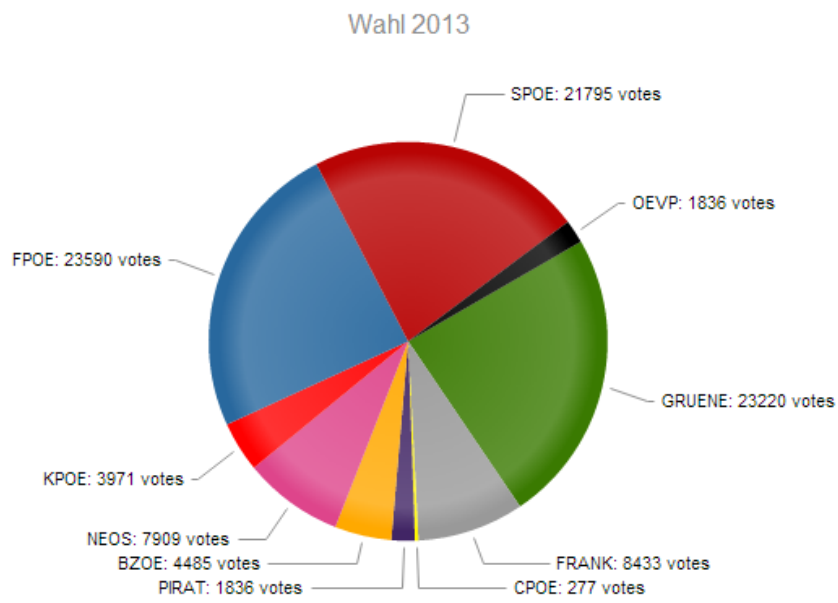


Figure 2.17: Example visualisation of the 2013 Graz election data with a pie chart.

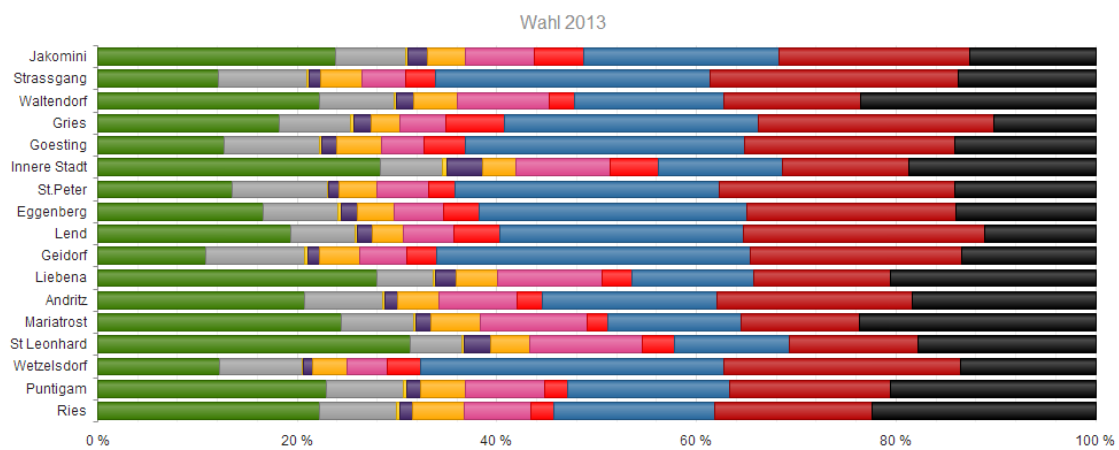


Figure 2.18: Example visualisation of the 2013 Graz election data with a stacked bar chart.

2.7 Google Charts

Google Charts, developed by Google¹⁸, is a library to create and embed interactive visualisations in webpages. It supports many different chart types and visualisations based on HTML5/SVG technology. It is free to use under the Creative Commons 3.0 License. The Google Chart code, however, may not be downloaded and hosted locally, that limits the usage of the visualisations to online usage.

Website: developers.google.com/chart/

Technology: Google Charts API uses HTML5 and SVG technologies to support cross-browser compatibility as well as the usage on different platform (such as mobile devices).

Platform support: Google Charts supports all different browsers (even older Microsoft Internet Explorer version with VML) and can be used on all mobile devices.

Features: The library is easy and intuitive to use and supports a variety of different chart types and visualisations such as bar charts, geo charts, tree-maps, org charts, gauges, timelines, heat maps, piles of money, or term clouds (see Figures 2.19 and 2.20). It also allows for different interactions with the charts. The current Google Charts version is not responsive. By adding a 100% width to the element and forcing the chart to reload when the window is resized charts can be made responsive. A nice feature of Google Charts is the playground which enables users to test and adapt their graph with an interactive interface.

Experience: Google Charts is a powerful tool for developers since it is very easy and intuitive to use and comes with developer tools such as a developer playground (see Figure 2.21), an elaborate documentation, and a vivid community. Also, it is easy to add new interactions and customise the charts. A large drawback is Google's terms of service which do not permit downloading and hosting the library code locally. This makes Google Charts only available to online use and makes the user highly dependent on the Google service. Since the charts are highly interactive and simple, the user experience is very positive. The visualisation of the election dataset (see Figures 2.22 and 2.23) was very straight forward and customisations such as individual colouring was experienced as very intuitive. Visualising the ACM hierarchy dataset in a tree-based chart (see Figure 2.24) was also very straight-forward. However, the standard implementation of a tree map supports going down the tree by clicking onto the category, but does not support going up again.

¹⁸All information about the technical details of this library was taken from [Google, 2012]

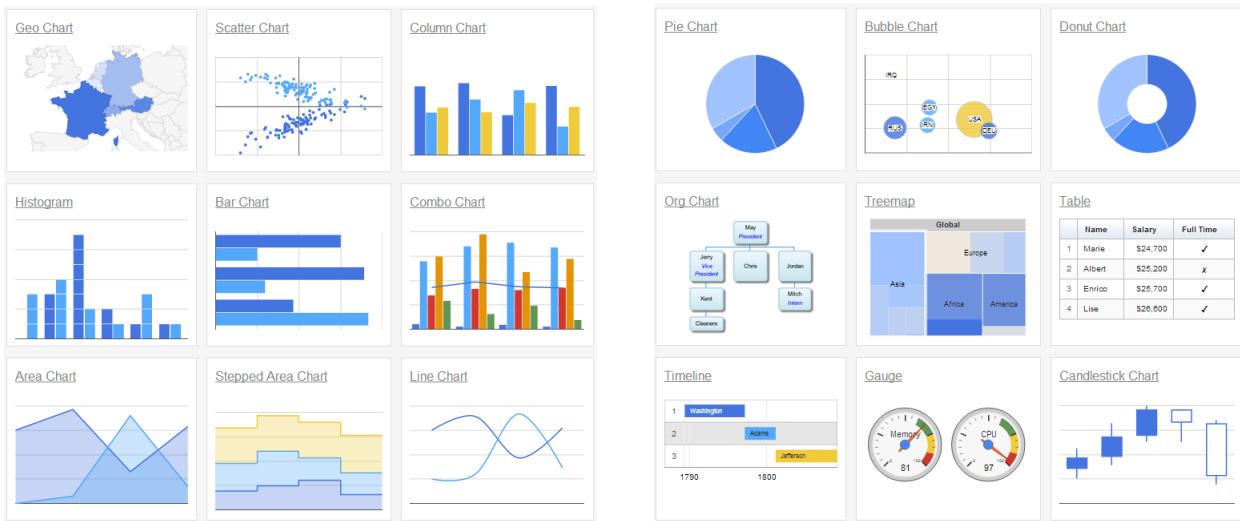


Figure 2.19: Examples of some of the supported chart types from [Google, 2012].

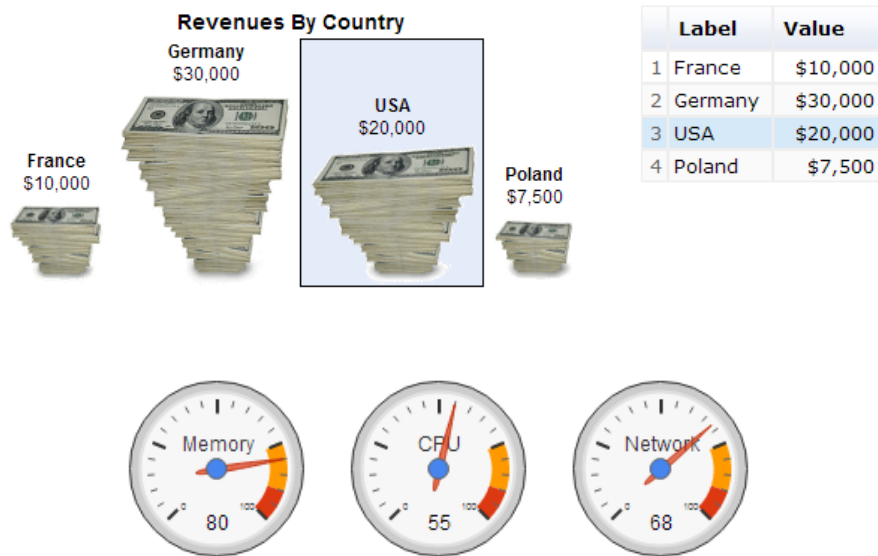


Figure 2.20: Examples of some of the supported chart types such as gauge charts or individual visualisations from [Google, 2012].

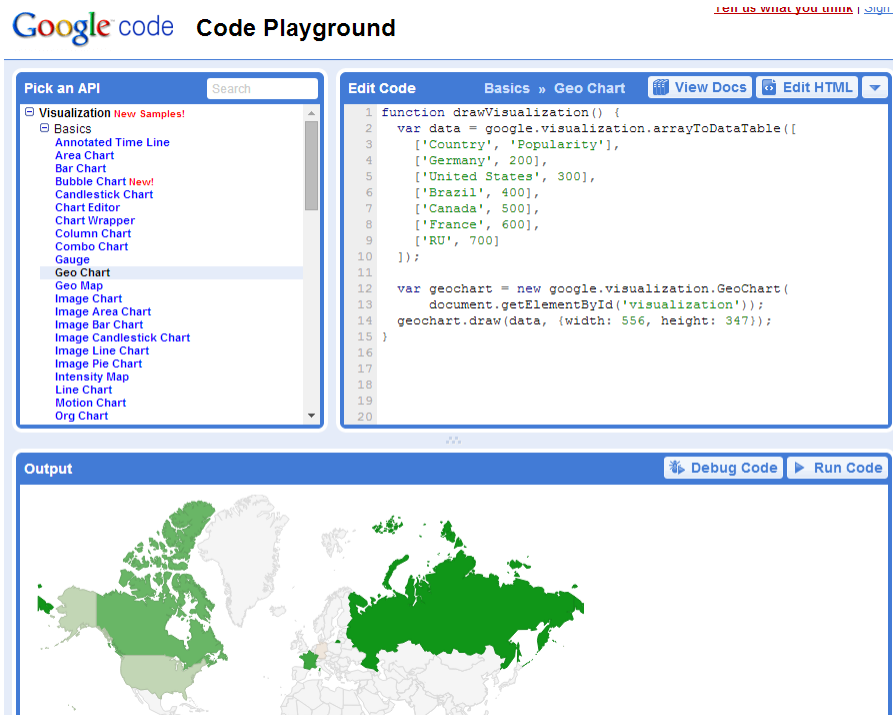


Figure 2.21: Google playground to test chart implementations from [Google, 2012].

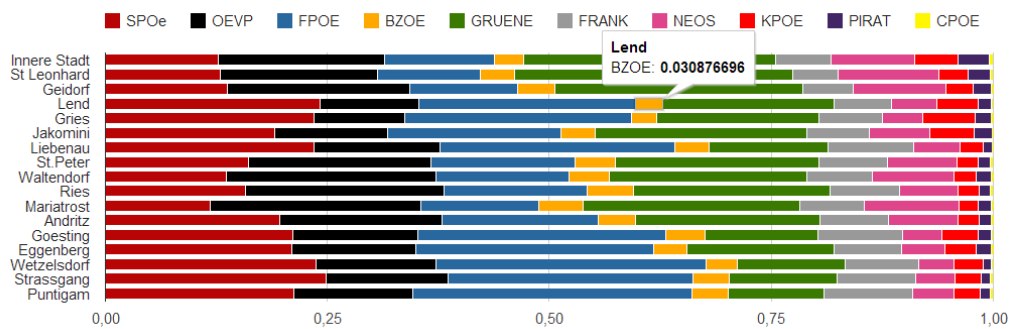


Figure 2.22: Stacked bar chart visualising the 2013 Graz election data.

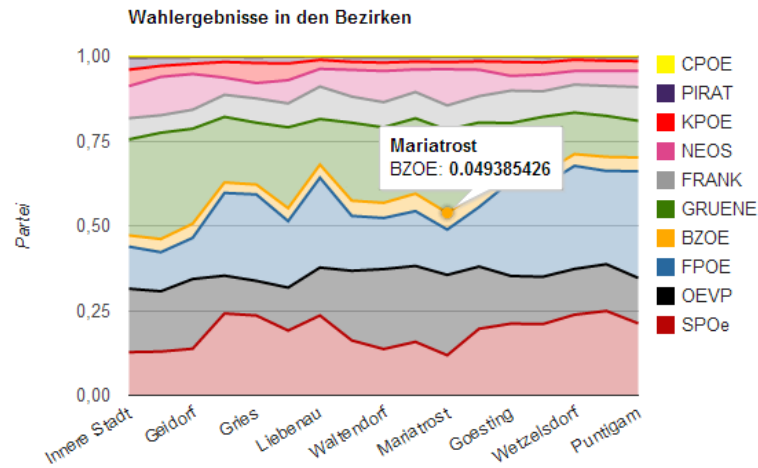


Figure 2.23: Area chart visualising the 2013 Graz election data.

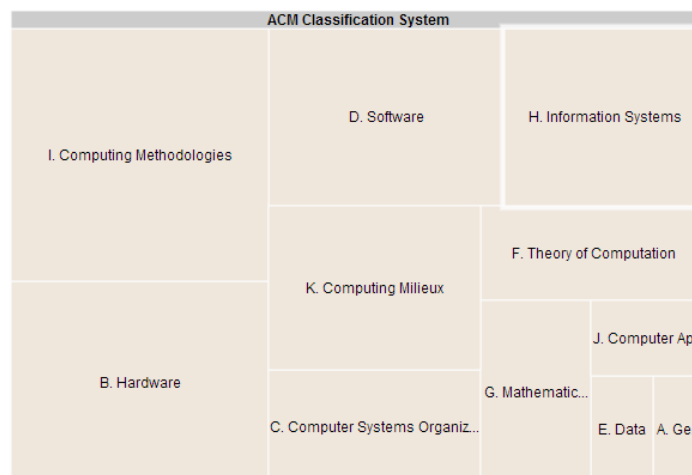


Figure 2.24: Example of a hierarchical visualisation of the ACM hierarchic data using the experimental tree-map visualisation.

2.8 Sencha Touch Charts

Sencha Touch Charts, a product developed by Sencha ¹⁹, is a HTML5-based library, especially designed for mobile devices. The actual version is part of the Sencha Touch Complete development kit (995 USD / 1 Developer Pack by May, 2014). It offers a range of visually appealing charts which are interactive and can utilise device-specific input such as swipes or rotations.

Website: dev.sencha.com/develop/touch-charts-rc

Technology: The charts are built in HTML5 and rendered in either in Canvas or SVG (depending on the platform). The charts are highly responsive and even react to the rotation of the used device.

Platform support: Sencha Touch Charts is especially designed for mobile devices, so some of the defined gestures to interact with the charts (e.g. swipes or rotations) might not work in the browser.

Features: Sencha Touch Charts supports basic chart types such as stock, area, bar, column, gauge, line, pie, radar and scatter chart (see Figures 2.25 or 2.26). Charts to represent tree structures are not supported.

Experience: Compared to other libraries such as Google Charts, Sencha Touch Charts is rather complicated to use. There are some examples and tutorials on the web, but the support section is not very helpful. This makes even simple changes such as colour changes time consuming. For the user the charts are very intuitive to use, they offer several visualisations, and also the device specific interactions are usually well designed. The standard styles are designed in line with mobile interface guidelines and the charts can be described as clearly arranged and visually appealing. The visualisation of the election data was not very intuitive. The visualisation functions are found in a separate JavaScript file, colour changes are made in a separate .css. Some interactions come from scratch (e.g. rotating the radar chart shown in Figure 2.27), other interactions (e.g. zooming in the area chart shown in Figure 2.28) have to be integrated manually. Since a tree representation is not supported by Sencha Touch Charts a visualisation of the ACM's hierarchical data is not possible.

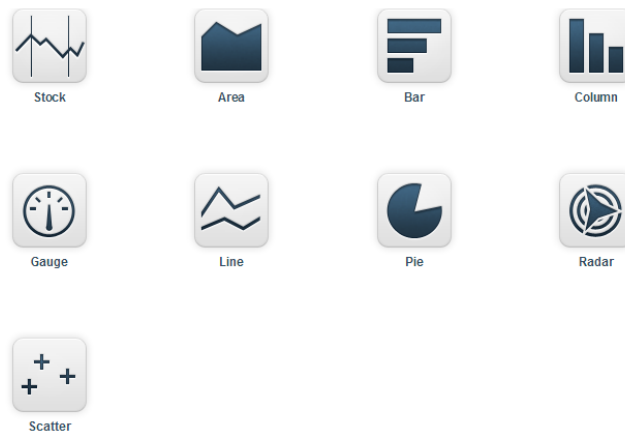


Figure 2.25: Supported chart types retrieved from [Sencha, 2011]

¹⁹All information about the technical details of this library was taken from [Sencha, 2011]

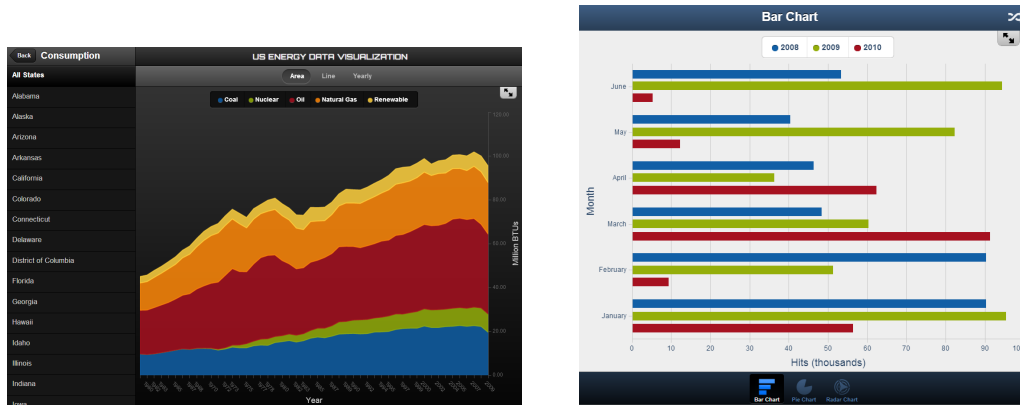


Figure 2.26: Sencha on mobile devices from [Sencha, 2011].

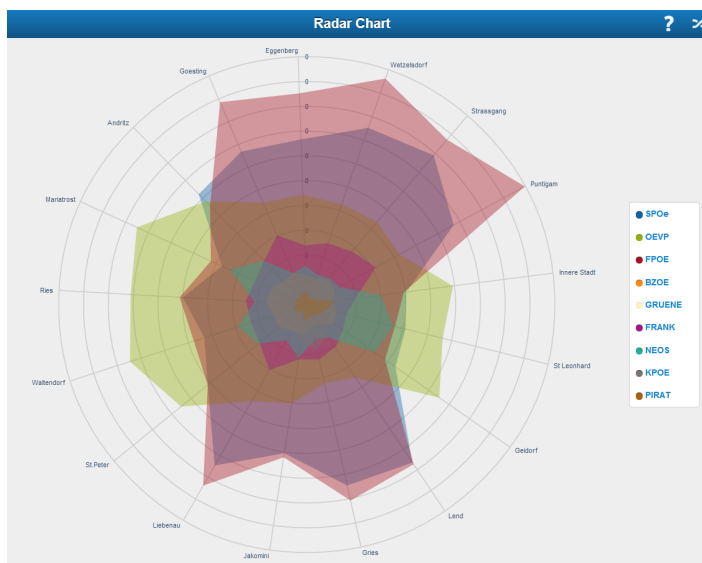


Figure 2.27: Example visualisation of the 2013 Graz election data with an radar chart.

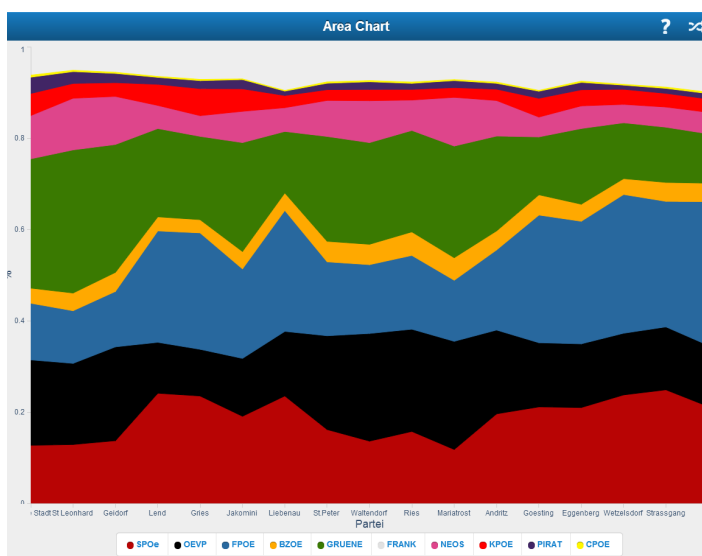


Figure 2.28: Example visualisation of the 2013 Graz election data with an area chart.

2.9 Other Libraries

There are many libraries which support especially mobile development but are often too restrictive (e.g. support only one platform), commercial or not documented. In this section some of the libraries are covered, which are in particular designed for mobile applications but were not tested in detail.

Android Charts: Android Charts is a tool to create Java charts for Android.

Website: www.artfulbits.com/products/android/aicharts.aspx

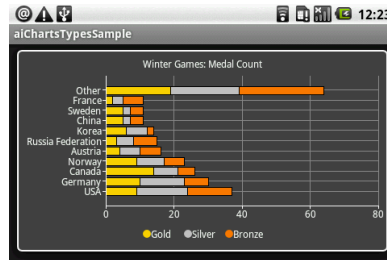


Figure 2.29: Bar Chart with Android Charts from [Android Charts, 2014]

Core Plot: Core Plot is a plotting framework which is specialised on 2D visualisations for OS X and iOS. The project is available on Github as open source software.

Website: github.com/core-plot



Figure 2.30: OHLC plot with coer plot on an iPad from [Skroch, 2014]

Zing: Zing is a commercial JavaScript chart library which uses HTML5 to support different devices. It is highly interactive and visually appealing. A license for a single domain is 249 USD.

Website: www.zingchart.com/



Figure 2.31: Dashboard visualisations with Zing from from [Zing Chart, 2014]

Chapter 3

Responsive Data Visualisation

Mobile devices have become ubiquitous. In fact [Cisco Systems, 2014] predict that the number of mobile devices will exceed the total world population by the end of 2014. This trend makes it continuously more important to support a wide range of clients when it comes to designing interfaces. While for some time it was customary to develop dedicated smartphone versions of web sites (more often than not with less features than the original version) responsive web design is now becoming increasingly popular [Kadlec, 2012]. However responsive data visualisation is still in its early stages.

3.1 Responsive Web Design

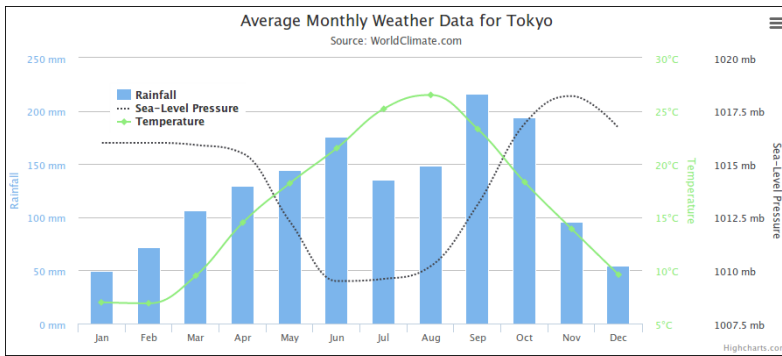
The key idea of responsive web design is to design web sites in a manner that allows for an optimal viewing experience regardless of the client used. The most prominent aspect of this paradigm is a layout that adapts itself dynamically to the client's screen size. For making websites containing text, images and interactive elements responsive, there are paradigms follow and technical standards and frameworks available that aid in doing so. [Marcotte, 2011] When it comes to data visualisation however, there are no common standards or methodologies that allow for a general solution to this problem. Since there are no standards and no established paradigms little to no resources are available on this subject.

Furthermore there is no general rule of thumb that applies to all different visualisation types for any given data sets which reduces clutter on clients with small screens. A visualisation that shows different levels of detail depending on the screen space available has to be designed for each type of data set and each type of visualisation, since simply downscaling the content is not always an option. However, there are also other options for responsive data visualisation apart from zooming.

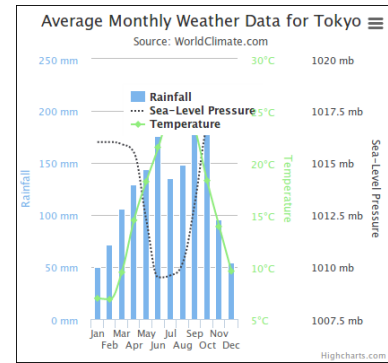
3.2 Responsive Data Visualisation Problems

When implementing a visualisation of the same data set for screens of arbitrary sizes trying to make all features of the desktop version operate in the same manner on every client can be problematic [Florit, 2013]. An easy-to-grasp example of such features are hover elements: On a client equipped with a mouse hovering the cursor poses no problem. On touchscreen devices however, hovering is not possible, since the input device does not allow for it.

Putting all information inside the visualisation container can also be problematic. If, for example, the legend of a chart is embedded inside the chart it might overlap labels or even data points when displayed on a small screen (see Figure 3.1). This can easily be avoided by displaying the legend outside of the visualisation itself. Well-known layouting and styling techniques can be used instead to still communicate the affiliation between the legend and the visualisation. However, visualisations consisting of multiple containers can sometimes cause users do make associations between individual portions of the visualisation depending on how they are

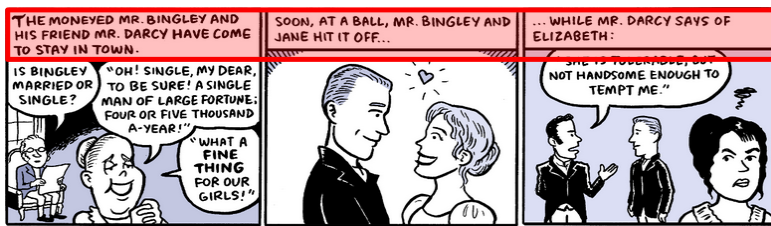


(a) A mixed bar-/line chart viewed on a regular PC.

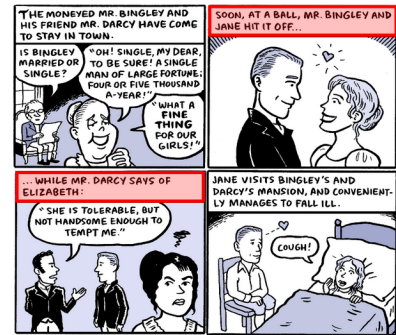


(b) The same chart viewed on a smaller screen

Figure 3.1: The same chart viewed on clients with different screen sizes. (a) shows the chart displayed on a large screen client. (b) illustrates the legend overlapping some data points on smaller screens.



(a) Three pictures seem to be directly related to each other.



(b) The same pictures do not seem that closely related when positioned differently

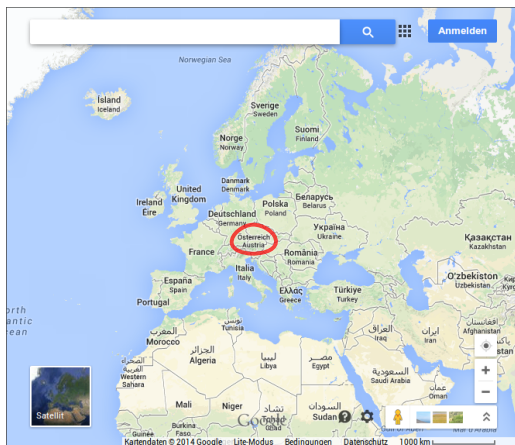
Figure 3.2: This responsive web comic by [Sorensen, 2013] uses text positioned at the same location on all tiles. This creates the impression that all side-by-side pictures are closely related to each other, even though they are not.

positioned. This is especially apparent for comic strip style narrative visualisations. Figure 3.4a illustrates this perceived connectedness, while in Figure 3.4b depicts a separation solely due to different positioning.

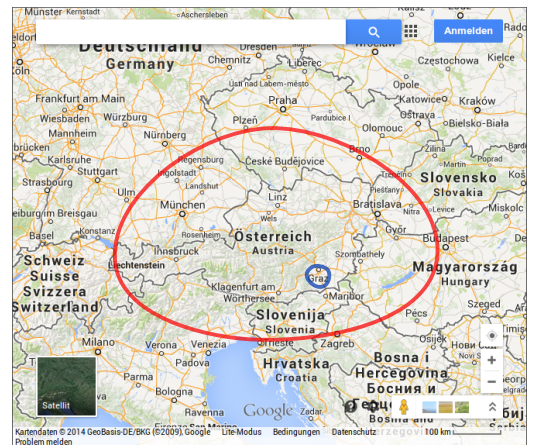
Another problem can arise from the data itself: when data points are close together they can not only be hard to distinguish but even harder to select on small-screen touchscreen devices. This is especially true when multivariate data is represented using glyphs, since glyphs generally do not scale. Simply augmenting the full-size visualisation with zoom controls for touchscreen devices is often not an option as much of the context can be lost when zooming in.

3.3 Responsive Data Visualisation Best Practices

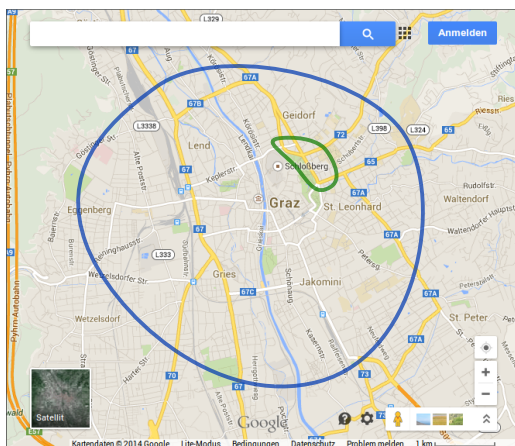
It is important to keep in mind that different devices often imply different use cases and that responsive design about more than screen size [Florit, 2013]. Since the visualisation used depends highly on the data and the intended message that is to be delivered there is no panacea for common problems such as fitting large amounts of data points into very little space. Therefore only basic approaches are presented in this section.



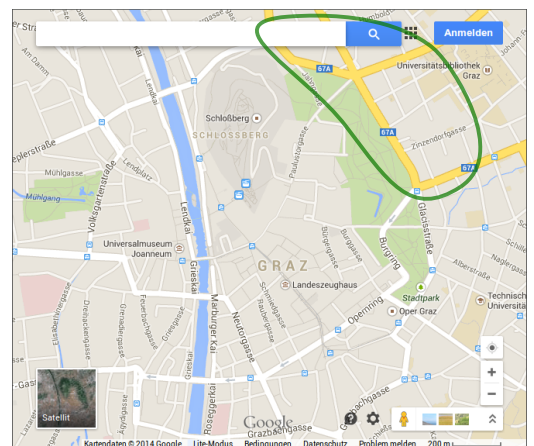
(a) Overview of Europe



(b) View of Austria and neighbouring countries



(c) Map of Graz and surroundings



(d) Centre of Graz

Figure 3.3: Illustration of Google Maps' increasing level of detail depending on the current zoom level with highlighted reference points.

3.3.1 Variable Level of Detail

Some data sets contain information that can be clustered, others may be structured in a way that a partial view or an overview is enough to outline the contents of the data set. Other data sets again contain information to which the intended audience already has a relation to or where the big picture is present in the minds of the viewer, so more details can be shown without the need to display the whole context.

Application in Map-Based Visualisations (GeoVis)

When geo-spatial data is to be visualised, the target audience can be expected to have a general idea of the area that is displayed in the visualisations. This knowledge can be exploited, by providing zoom controls (regardless of client screen size) that display different kinds of labels and borders (country, state, province, city, district) as users zoom. Google Maps¹ is a prime example for this kind of zoom controls and works on any screen size. Users can zoom in to the maximum level even on small screen smartphones since users generally know about the context of the part of the map that is currently shown. Furthermore even if users do not know much about the area displayed continuous zooming makes it easy to keep track of reference points while zooming (see Figure 3.3).

¹<http://maps.google.com>



(a) A time line data visualisation zoomed to a level appropriate for a large screen.



(b) The same dataset zoomed in for viewing on small screen clients

Figure 3.4: The same time line data set on different zoom levels with adaptive axis labelling and an overview pane at the bottom.

Application with Clusterable Data

When trying to make existing variable level of detail visualisations of clusterable data responsive, general responsive web design approaches apply: The main thing to keep in mind is to layout the visualisation contents as space-saving as possible on small screen clients. If additional views are present, these should be put into separate components, so that a responsive layout can be applied to all of the visualisation components individually, making the most out of the available screen space.

3.3.2 Compound Overview-Detail Visualisation

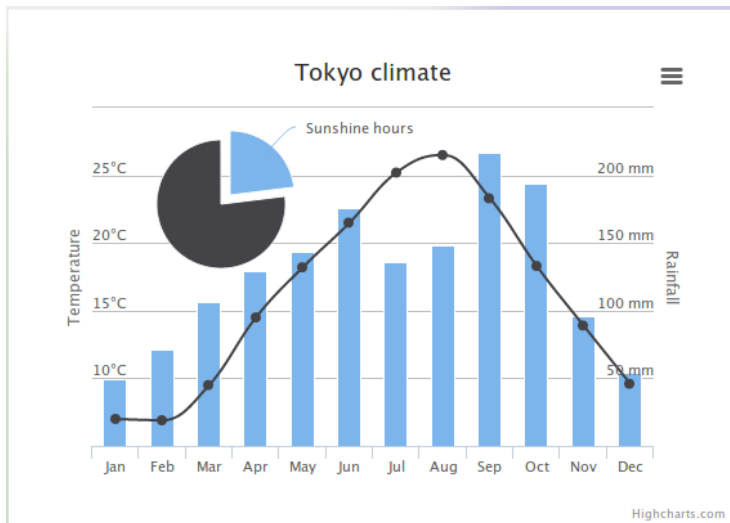
Some visualisations rely on zoom controls to present the information contained in the data at hand in a meaningful way. However zooming generally results in some degree of context loss. This problem is especially present on small screen clients. Any easy way to remedy this problem is employing a two-component overview-detail visualisation.

Application in Time Line Data Visualisation

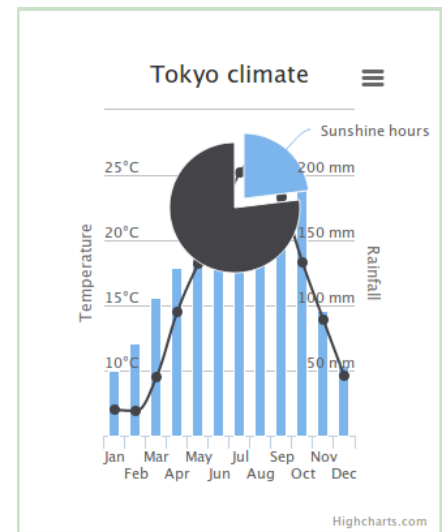
Time line data can span over a large period of time. For large data sets it is generally impossible for humans to process all information at once. Therefore a second visualisation element displaying an overview can be employed to provide context. Adaptive axis labelling, smoothing of data lines and adaptive scaling of the grid can also be applied. Figure 3.4 illustrates adaptive stepping of axis labelling as well as countering context-loss using an overview element displaying the whole data set. This combines variable level of detail for the axis labelling with the overview-detail approach resulting in a pleasant viewing experience on all clients.

3.3.3 Modular Visualisations

Some data sets contain different kinds of information. When implementing a visualisation for such data sets, using the available space efficiently to visualise multiple aspects of the source data inside one container can be suitable. However, on clients with small screens this can lead to overlaps inside a single container even if the visualisation scales accordingly (see Figure 3.5). On the other hand If the visualisation does not adapt its size it forces the user so scroll or zoom in order to be able to view the visualisation as a whole. This can be easily remedied by splitting up tightly-packed visualisations into multiple containers.



(a) The example chart from www.highcharts.com viewed on a regular PC.



(b) The same chart viewed on a smaller screen results in a confusing picture

Figure 3.5: The same data viewed on clients with different screen sizes. (a) shows a combined chart displayed on a large screen client. (b) illustrates one sub-visualisation overlapping the main data points on smaller screens.

3.3.4 Responding to the Client

When it comes to fully harnessing the capabilities of each client the same guidelines apply for both responsive web design and responsive data visualisation. For example, in cases where geolocation data is to be put into context with the current position of the client, built-in sensors can be used on smartphones, while a location picker is displayed on other clients. [Florit, 2013] This can easily be implemented since modern web standards feature APIs for accessing sensor data.

Another important factor is processing power and web browser optimisation: Different hardware, operating systems and web browsers behave differently and one may take significantly longer perform the same tasks as others. This is most apparent when it comes to different rendering techniques. SVG-based visualisations inject elements into a website's DOM tree, leading to complex structures that are kept in-memory, sometimes even after they are not visible any longer. When animated SVG is used to visualise data, low-performance clients are essentially left out, since they cannot process the workload. HTML5 Canvas is more suitable for this task, since this technology is designed for immediate mode graphics and animations. However neither SVG nor Canvas employ hardware acceleration. Therefore WebGL should be used where possible, since it drastically improves rendering performance on supported clients.[Andrews and Wright, 2014]

3.3.5 Unresponsive Data

Even though different data sets require different visualisation approaches and a variety of techniques can be applied to implement responsive data visualisations there are some data sets that cannot be made available on all clients. In such cases filtering data or providing mobile clients with a visualisation that lacks some features has to be considered. Since this approach conflicts with responsive design principles the visualisation choice for a particular data set may also be questioned. In the end however, some data sets and some visualisation techniques that present the information contained in complex data require large amounts of screen estate and complex controls and can therefore not be adaptable for all possible clients.

Chapter 4

Concluding Remarks

In this paper several visualisation weblibraries were discussed and evaluated with focus on mobile device support. On the following pages the detailed evaluation results are presented. The qualitative expert evaluation with 4 evaluators (the authors) includes ratings on a Likert scale between 1 (strongly disagree) and 5 (strongly agree) of the aspects (1) user experience, (2) programming experience, (3) visually attractive, (4) mobile device support, (5) flexibility (versatility), (6) performance and (7) an overall rating. Based on these rating Highcharts, Sencha Touch Charts and D3 were evaluated as best choice for visualisation on mobile devices. While Sencha Touch Charts supports several mobile-specific features it lacks on positive programming experience and flexibility. D3 is a very good choice for small datasets, the loading time for large datasets, however, gets too long for mobile devices.

As a optimised solution the use of libraries which support responsive behaviour such as Highcharts is recommended. This makes certain that the visualisation is rendered properly on all different mobile devices such as mobile phones or tables, but also in webpages on different screen resolutions.

Unfortunately at the time this paper was authored, the information and research about responsive visualisations was very limited. For future work the exploitation of this topic is recommended for different chart types with focus on heuristics on how to adapt the visualisations with respect to different content requirements.

4.1 Evaluation Results

The following table summarises the evaluation results and compares the libraries based on different objective and also subjective quality criteria.

Table 4.1: Comparison of the different chart libraries with focus on mobile support

Criteria	Sencha Touch Charts	Charts.js	ChartsJs	Highcharts	Google Charts	D3	Dojo Mobile	Kendo
Summary	charting library especially for mobile devices	Easy, object oriented client side graphs for designers and developers		General-purpose charting library. Large amounts of examples. Extensive API Documentation	charting library for websites, easy to develop and embed			
Company	Sencha		DevExpress	Highsoft AS	Google		the Dojo Foundation	Telerik
First released	2011	2013	unkown	2006	2007	2011	2005	unknown
Charts rendered in	HTML5, SVG, or Cancas	Canvas based charts	SVG	SVG	HTML5, SVG	SVG, (canvas)	SVG, VML, canvas, Silverlight, WebSVG	SVG, VML, canvas
Input Data	JavaScript API	JavaScript API	JavaScript API	JSON	javascript api	javascript api	javascript	javascript
Device Supported	All modern browsers, all mobile devices with HTML5/SVG support	All modern browsers, including IE6+	All modern browsers, including IE8+, Safari 5+, Opera 12+	All Browsers including MSIE6+	All modern Browsers including IE6	All modern browsers which support SVG	All modern browsers	All modern browsers
Special Mobile Input Support	Yes		Yes	not needed			supposedly	Yes
Good Documentation	Average	Average	Yes	Yes	Yes	Yes, most examples	fair	Yes
Good Demos / Examples	Yes	Yes	Yes	Yes	Yes	Yes	no	Yes, excellent

*... rating on a Likert scale from 1 to 5, 5 .. best, Mean (Standard Deviation); **... Overall mean (SD)

Table 4.1: Comparison of the different chart libraries with focus on mobile support

Criteria	Sencha Touch Charts	Charts.js	ChartsJs	Highcharts	Google Charts	D3	Dojo Mobile	Kendo
Licensing	Included in Sencha Touch Complete under a commercial license	Free under MIT license	Free for non-commercial	CC-SA and various paid licences	free CC	BSD	Modified BSD License (BSD) or the Academic Free License version 2.1	Commercial, Apache v2 for a core subset (does not include visualizations)
Support	Forums, paid support			Forums, dedicated StackExchange tag, e-mail, community-voted feature requests, github bug-tracker	Forums		Forums	24h support for paying customers
Responsive	x		x, excellent	x				
Interactive	x		x	x	x	x	x	x
Animations	x	x	x	x	x	x	x	x
Supported Chart Types:								
Column	x		x	x	x	x	x	x
Bar	x	x	x	x	x	x	x	x
Line	x	x	x	x	x	x	x	x
Area	x		x	x	x	x	x	x
Pie / Donut	x	x	x	x	x	x	x	x
Scatter / Bubble	x		x	x	x	x	x	x
Stacked	x		x	x	x	x	x	x
Candestick			x		x	x	x	x
Gauge	x		x	x	x	x	x	x
Pyramid						x		
Gantt						x	x	x
Tree					x	x	x	
Geo map			x	x	x	x	x	x

*... rating on a Likert scale from 1 to 5, 5 .. best, Mean (Standard Deviation); **... Overall mean (SD)

Table 4.1: Comparison of the different chart libraries with focus on mobile support

Criteria	Sencha Touch Charts	Charts.js	ChartsJs	Highcharts	Google Charts	D3	Dojo Mobile	Kendo
Additional charts types	Radar	Radar- and polar chart	RangeSelector, BubbleChart	Radar- and polar-chart, heatmaps, waterfall	Individual visualisations, image visualisations	Almost 2000 examples		
Personal rating user experience*	4 (0)	3.3 (0.6)	4.3 (0.6)	5 (0)	4 (0)	4.7 (0.6)	3 (0)	3.3 (0.6)
Personal rating programmer experience*	3 (0)	3.5 (0.7)	4.3 (0.6)	4 (0)	4.5 (0.7)	3.7 (0.6)	1.7 (0.6)	4.7 (0.6)
Visually attractive charts*	5 (0)	4 (1)	5 (0)	4.3 (0.6)	3.7 (0.6)	5 (0)	3 (0)	4.7 (0.6)
Good for mobile (e.g. touch-friendly)*	5 (0)	2 (0)	3.3 (0.6)	4.3 (0.6)	1.3 (0.6)	2 (0)	2 (0)	2 (0)
Flexibility*	2.7 (1.5)	2 (0)	2.3 (0.6)	3.3 (0.6)	3.7 (1.2)	4.7 (0.6)	4 (0)	2.7 (0.6)
Performance*	4 (0)	3.3 (0.6)	3.3 (0.6)	4 (0)	4 (0)	2.7 (0.6)	4 (0)	4 (0)
Overall Rating*	4 (0)	3 (0)	3.7 (0.6)	4 (0)	3.7 (0.6)	4.3 (0.6)	3.3 (0.6)	3 (0)
Summary Rating**	3,95 (0,89)	3,02 (0,76)	3,76 (0,88)	4,14 (0,50)	3,55 (1,02)	3,86 (1,14)	3,00 (0,90)	3,48 (1,02)

*... rating on a Likert scale from 1 to 5, 5 .. best, Mean (Standard Deviation); **... Overall mean (SD)

Table 4.1: Comparison of the different chart libraries with focus on mobile support

Criteria	Sencha Touch Charts	Charts.js	ChartsJs	Highcharts	Google Charts	D3	Dojo Mobile	Kendo
Comments	Very good solution for mobile devices, unfortunately a commercial product, not very developer friendly	Simple, dependancy free non-responsive graphs.	A very visually pleasing commercial product.	Excellent out-of-the box support for all clients including mobile-friendly components	Excellent for quick implementation of graphs, great documentation, not optimized for mobile devices	Incredible visualization library, with large amounts of examples. It has a high learning curve and performance issues with large amounts of data.	The GFX engine can do anything but with limited examples and support it's up to the developer to create good visualizations.	While limited the library does the supported things well, it's also easy to use with comprehensive examples and documentation.

List of Figures

1.1	Example of a data entry of the Graz 2013 election data from [Bundeskanzleramt, 2014]	2
1.2	Sample of the ACM classification dataset from [ACM, 1998]	2
2.1	Chart.js visualisation examples	5
2.2	Chart.js Polar chart for Election data	6
2.3	Pie chart and stacked bar chart depicting election data.	8
2.4	ChartJS example visualisations	9
2.5	ChartJS Responsive visualisation of election data	10
2.6	Examples of visually complex visualisations realised with D3	12
2.7	Interactive visualisation of ACM hierarchy.	13
2.8	Highcharts example charts	15
2.9	Highcharts stacked bar chart of the 2013 Graz election data	15
2.10	Example of a poor and a reasonably good visualisation with the Dojo toolkit (examples selected from [The Dojo Foundation, 2004]).	16
2.11	Examples of some of the supported chart types (images taken off [The Dojo Foundation, 2004]).	17
2.12	Example visualisation of the 2013 Graz election data.	17
2.13	Example of a hierarchical visualisation of the ACM hierarchic data with the experimental tree-map visualisation.	17
2.14	Example geographical visualisation (taken off www.telerik.com/kendo-ui).	18
2.15	Examples of some of the supported chart types (taken off [Telerik, 2014]).	19
2.16	An example of how the developers see the library used well (example from [Telerik, 2014]).	19
2.17	Example visualisation of the 2013 Graz election data with a pie chart.	20
2.18	Example visualisation of the 2013 Graz election data with a stacked bar chart.	20
2.19	Examples of some of the supported chart types from [Google, 2012].	22
2.20	Examples of some of the supported chart types such as gauge charts or individual visualisations from [Google, 2012].	22
2.21	Google playground to test chart implementations from [Google, 2012].	23
2.22	Stacked bar chart visualising the 2013 Graz election data.	23
2.23	Area chart visualising the 2013 Graz election data.	24
2.24	Example of a hierarchical visualisation of the ACM hierarchic data using the experimental tree-map visualisation.	24
2.25	Supported chart types retrieved from [Sencha, 2011]	25
2.26	Sencha on mobile devices from [Sencha, 2011].	26
2.27	Example visualisation of the 2013 Graz election data with an radar chart.	26
2.28	Example visualisation of the 2013 Graz election data with an area chart.	26
2.29	Bar Chart with Android Charts from [Android Charts, 2014]	27
2.30	OHLC plot with coer plot on an iPad from [Skroch, 2014]	27
2.31	Dashboard visualisations with Zing from from [Zing Chart, 2014]	27
3.1	Responsive data visualisation: scaling problems	30

3.2	Layouting problems through responsive design. Illustrated using the web comic “Pride And Prejudice’ Turns 200: A Cartoon Celebration”	30
3.3	Responsive data visualisation: adaptive level of detail	31
3.4	Responsive data visualisation: adaptive timeline level of detail	32
3.5	Responsive data visualisation: overlapping charts	33

Bibliography

- ACM [1998]. *ACM Classification 1998*. 1998. <http://www.acm.org/about/class/1998> (cited on pages 1, 2).
- Andrews, Keith and Benedict Wright [2014]. “FluidDiagrams: Web-Based Information Visualisation using JavaScript and WebGL”. In: *Eurographics Conference on Visualization (EuroVis) (2014)*. Edited by N. Elmqvist, M. Hlawitschka, and J. Kennedy. Eurographics Conference on Visualization. 2014 (cited on page 33).
- Android Charts [2014]. *Android Charts -v1.7*. 2014. <http://www.artfulbits.com/products/android/aicharts.aspx> (cited on page 27).
- Bostock, Mike et al. [2014]. *D3 Gallery*. 2014. <https://github.com/mbostock/d3/wiki/Gallery> (cited on page 12).
- Bostock, Mike [2011a]. *Data Driven Documents (D3)*. 2011. <http://d3js.org/> (cited on page 11).
- Bostock, Mike [2011b]. *DOM-to-Canvas using D3*. Oct. 2011. <http://bl.ocks.org/mbostock/1276463> (cited on page 11).
- Bundeskanzleramt [2014]. *Open Data Austria*. 2014. <http://www.data.gv.at/> (cited on pages 1, 2).
- Cisco Systems, Inc. [2014]. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018*. Technical report. Cisco Systems, Inc., Feb. 2014 (cited on page 29).
- DevExpress [2013]. *Chart.js*. June 2013. <http://chartjs.devexpress.com/> (cited on page 7).
- Dewey, Ben [2012]. *Getting Started with Windows 8 Apps: A Guide to the Windows Runtime*. O’Reilly Media, 2012. ISBN 9781449321239 (cited on page 1).
- Downie, Nick [2013]. *Chart.js*. 2013. <http://www.chartjs.org/> (cited on pages 4, 5, 9).
- Florit, Gabriel [2013]. “On Responsive Design and Data Visualization”. In: *OpenVis Conf 2013*. The Open Web Data Visualization Conference. Boston, Apr. 2013 (cited on pages 29, 30, 33).
- Google [2012]. *Google Charts*. Apr. 2012. <https://developers.google.com/chart> (cited on pages 21–23).
- Highsoft AS [2009]. *HighCharts*. 2009. <http://www.highcharts.com/> (cited on pages 14, 15).
- Kadlec, T. [2012]. *Implementing Responsive Design: Building sites for an anywhere, everywhere web*. Voices That Matter. Pearson Education, 2012. ISBN 9780132979368 (cited on page 29).
- Marcotte, Ethan [2011]. *Responsive Web Design*. A book apart. A Book Apart, 2011. ISBN 9780984442577 (cited on page 29).
- Sencha, Inc. [2011]. *Sencha Touch Charts v1*. 2011. dev.sencha.com/deploy/touch-charts-rc/ (cited on pages 25, 26).
- Skroch, Eric [2014]. *Core Plot on Github*. 2014. <http://github.com/core-plot> (cited on page 27).

- Sorensen, Jen [2013]. *'Pride And Prejudice' Turns 200: A Cartoon Celebration*. Jan. 2013. <http://www.npr.org/2013/01/27/170253360/pride-and-prejudice-turns-200> (cited on page 30).
- Telerik [2014]. *Kendo UI*. 2014. <http://www.telerik.com/kendo-ui> (cited on pages 18, 19).
- The Dojo Foundation [2004]. *Dojo Toolkit*. 2004. <http://dojotoolkit.org/> (cited on pages 16, 17).
- Zing Chart [2014]. *Zing Chart Website*. 2014. <http://www.zingchart.com/> (cited on page 27).