Responsive Visualisation

Keith Andrews
ISDS, Graz University of Technology
Graz, Austria
kandrews@tugraz.at

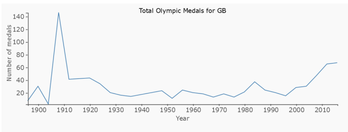
MobileVis '18 Workshop at CHI 2018, April 21, 2018, Montreal, QC, Canada.
https://mobilevis.github.io/

Abstract
Responsive web design allows web pages and web apps to be assembled from flexible components, which can adapt to the constraints and opportunities of the display device. This paper looks at how the principles of responsive web design can be applied to five commonly used web-based data visualisations: line chart, bar chart, parallel coordinates, scatterplot, and choropleth map. Interactive examples are provided.

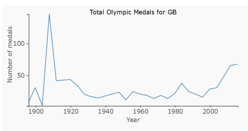
Author Keywords
Responsive web design; data visualisation; information visualisation; D3.

ACM Classification Keywords
H.5.2 [User Interfaces]: Graphical user interfaces (GUI); H.3.5 [Online Information Services]: Web-based services

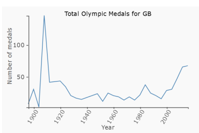
Introduction
Responsive web design [15] has rapidly become the predominant way to design web pages and applications. Web design today *is* responsive web design. In responsive web design, rather than designing and maintaining separate web sites for different types of device, a single design under a single URL adapts (responds) to the characteristics of the end user's device. The various chunks of content such as blocks of text, tables, images, videos, ads, as well as



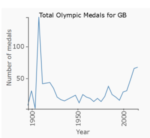
(a) 70 em.



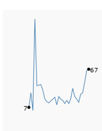
(b) 50 em.



(c) 40 em.



(d) 30 em.



(e) 20 em.

Figure 1: A responsive line chart. Tick marks are thinned out and tick labels are rotated. Finally, the chart becomes a sparkline.

charts, graphics, and visualisations must be able to adapt both to the available screen space and to other device capabilities such as touch interaction, motion events, ambient light, and so forth.

Responsive Rather Than Simply Scalable

I believe a distinction should be made between scalable visualisations and responsive visualisations. *Scalable* visualisations freely scale to fit the available screen space, but their basic appearance remains the same. Scaling can be implemented in many ways, both with or without skewing, for example by using relative units (rems or ems) rather than absolute units (such as pixels) or by intercepting resize events and reacting accordingly.

Responsive visualisations, on the other hand, go further:

- *Responsive Layout*: The visualisation is freely scalable between breakpoints, but changes its appearance or form at specific breakpoints. For example, a bar chart might switch from vertical to horizontal bars at smaller viewport widths.
- *Responsive Display Density*: The visualisation takes account of the display density, for example by sampling (important) data points for lower resolution displays.
- *Responsive Interaction*: the visualisation provides selective support for a variety of input modalities, such as touch (tap, swipe, pinch zoom), keyboard, mouse, motion events, etc.

In essence, a responsive visualisation contains logic within itself, so that it can adapt to specific display constraints and opportunities. It is more than simply scaling to fit the available space. Web-based visualisations can be made responsive using JavaScript and/or the ability to embed CSS3 media queries inside SVG.

Hinderman's book [9] is titled "Building Responsive Data Visualization for the Web", but essentially describes how to make visualisations scalable using D3 [6]. Koerner [10] covers similar ground, but also considers interactive selection and touch events.

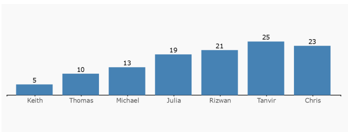
Examples of Responsive Visualisations

To illustrate responsive visualisations, several examples are provided. Since it is difficult to convey the interactive nature of responsive visualisations in a static format such as this paper, the reader is invited to watch the video clips and try out the online examples at the project web site [2]. The first four examples are implemented in JavaScript with D3 v4 [6], the choropleth map is implemented with Leaflet [14] and SVG and uses map tiles from basemap.at.

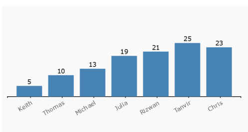
Responsive Line Chart

A line chart can adapt to its display width in a number of ways. First, tick marks and labels on the x axis can be thinned out. Tick labels can be rotated or abbreviated if less space is available. At the narrowest widths, the axes can be removed entirely, producing a sparkline [22, pages 46–63]. Figure 1 shows the same responsive line chart at widths of 70, 50, 40, 30, and 20 em.

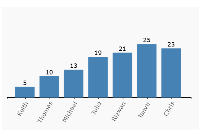
If a very large number of data points are available for a line chart, the data itself can be downsampled to an appropriate density (taking care to preserve important landmark data points). Le Bek [13] presents a crude way to downsample line chart data once it has reached the client, which reduces the number of drawing operations required. However, it makes more sense in general to downsample such data server-side before shipping it to the browser, much like the way responsive images are now handled.



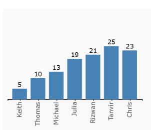
(a) 70 em.



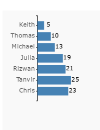
(b) 50 em.



(c) 40 em.



(d) 30 em.



(e) 20 em.

Figure 2: A responsive bar chart. First, labels are rotated. Finally, the chart is flipped by 90°.

Responsive Bar Chart

Bar charts (assuming vertical bars for the moment) can be made freely scalable by scaling the width of the individual bars between a maximum and minimum value. Bar charts can be made responsive in a similar way to line charts. Labels on the x axis can be rotated or abbreviated. A more radical approach, at the narrowest widths, is to flip (rotate) the bar chart by 90°, so that vertical bars become horizontal bars, utilising vertical space to accommodate however many bars are present. Figure 2 shows the same responsive bar chart at widths of 70, 50, 40, 30, and 20 em.

Responsive Parallel Coordinates

A parallel coordinates visualisation is usually displayed with the data dimensions spaced as vertical parallel lines. Fluid scaling can be achieved by smoothly decreasing the separation between the dimensions and possibly rotating the dimension labels. To be responsive, at narrower widths, it might be preferable to selectively display a subset of the available dimensions. This is shown in Figures 3 and 4. As soon as one or more dimensions have been omitted from the display, a button “Dimensions” is shown, so that the user can override the default choice of which dimensions to display and which to hide. A more radical approach at narrower widths would be to flip the visualisation by 90°, so that dimensions are drawn as horizontal parallel lines and the visualisation extends vertically.

Responsive Scatterplot

Scatterplots are particularly difficult to make responsive, because of issues of occlusion of data points when display space is limited. Sampling or aggregation can be used to reduce the number of points to be displayed, but the original data points are no longer reflected. Filtering and zooming (pinch zooming on touch devices) are essential for meaningful interaction. Additionally, disambiguation methods

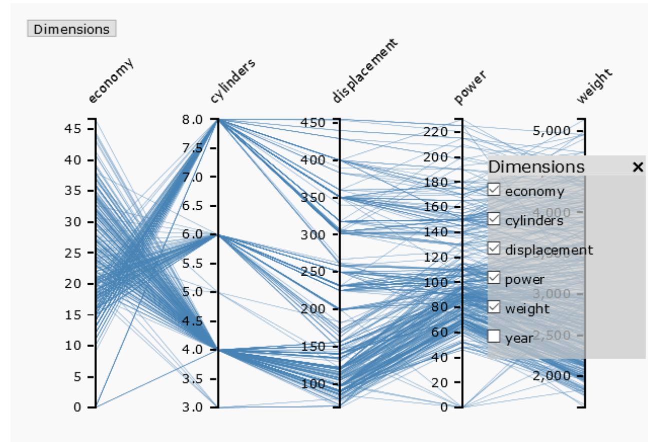
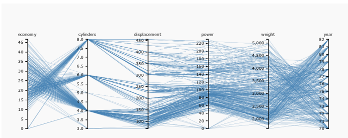


Figure 3: Responsive parallel coordinates with a dimensions chooser. At narrower widths, only a subset of dimensions is shown, under user control.

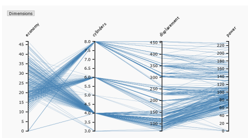
such as fisheye distortion, Cartesian distortion, and temporary displacement can be extremely useful. Figure 5 shows a responsive scatterplot with a fisheye lens at 70 em. The x (y) axis can be swiped left or right (up or down) to select the dimension to display. Figure 6 shows the same responsive scatterplot with a Cartesian lens at 30 em.

Responsive Choropleth Maps

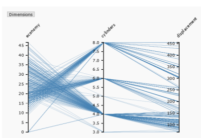
Responsive choropleth maps should support smooth zooming and touch interaction (pinch zoom, etc.) on touch-enabled devices. On larger displays, higher resolution (a higher sampling density of points) along the polygon borders is desirable. On smaller displays, lower border resolution might well suffice and is faster to draw. A set of polygon borders at different resolutions can be prepared in advance (server-side) and the most appropriate chosen dynamically to suit the current display space and resolution, similar to the way



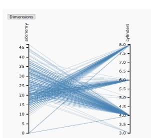
(a) 70 em.



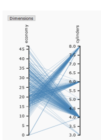
(b) 50 em.



(c) 40 em.



(d) 30 em.



(e) 20 em.

Figure 4: A responsive parallel coordinates visualisation. Dimensions are selectively removed.

responsive images are typically handled. In addition, the thickness of polygon borders can be steered according to the characteristics of the display. Where points of interest (POIs) are overlaid atop the map, grouping icons can be used to reduce clutter, automatically expanding and collapsing as the user zooms in and out. The user should be able to maximise the available display space for the map, by clicking (tapping) to temporarily hide away controls and options. The widely used JavaScript library Leaflet [14] supports layers of vectors atop layers of tiles.

The Styrian Diversity Visualisation (in German “Steirische Vielfalt Visualisiert” or SVV) project [4, 5, 20] provides a graphical web interface built around a choropleth map to a large collection of demographic data concerning the Austrian province of Styria. Figure 7 shows SVV on an iPad Air 2 at a resolution of 2014×1536. Figure 8 shows SVV on a OnePlus 2 smartphone at a resolution of 1080×1920. Upto three info boxes are displayed on top of the map. At narrower widths, a carousel-like arrangement is used with left and right arrows to move between the info boxes. In Figure 8, the user has temporarily hidden the Scenario bar at the top; it is revealed by tapping the double chevron tab.

Related Work

Hinderman [9] describes how to make scalable (but not responsive) visualisations with D3 [6]. Koerner [10] covers similar ground, but goes on to consider interactive selection and touch events. Le Bek [13] discusses aspects of responsive line charts, including a crude way to downsample line chart data once it has reached the client. Examples of responsive bar charts include [17, 18]. Responsive scatterplots are discussed by Florit [8], Kissane [11], and Meeks [16]. Nick Rabinowitz [19] demonstrates a simple responsive scatterplot, which aggregates points into cells based on the point to pixel density ratio. Touch interactions for vi-

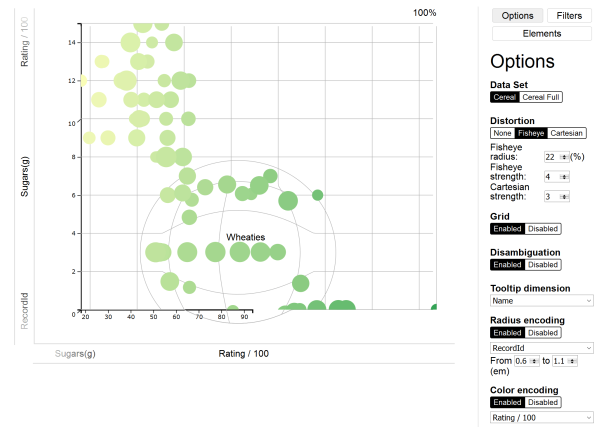


Figure 5: Responsive scatterplot at 70 em. Pinch zooming and panning are available. The fisheye lens can be moved and resized.

Figure 6: Responsive scatterplot at 30 em with a Cartesian lens. The Options panel is repositioned below the plot.

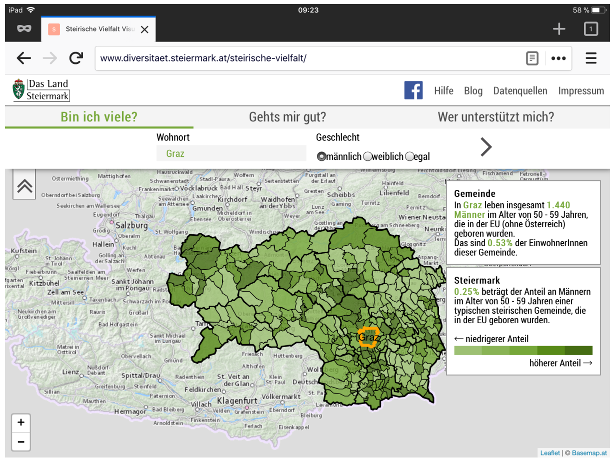


Figure 7: The SVV user interface on an iPad Air 2 at a resolution of 2014×1536.

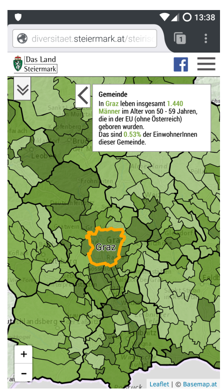


Figure 8: The SVV user interface on a OnePlus 2 smartphone at a resolution of 1080×1920. The double chevron tab conceals the Scenario bar. At narrower widths, upto three info boxes are provided in the carousel.

sualisations are described by Diakopoulos [7]. Tableau's Vizable [21] provides examples of touch interaction for line charts and bar charts.

The Chartist.js library [12] provides simple line charts and bar charts using JavaScript and SVG (without D3). The charts can be made responsive through an elegant mechanism to selectively override base settings with CSS media queries specified in JavaScript.

Concluding Remarks

Earlier versions of this work were presented as a poster at EuroVis 2017 [3] and a talk at Graphical Web 2016 [1]. The project web site [2] provides online demos and video clips.

Acknowledgements

Miran Levar, Johanna Pirker, Bernd Prünster, Werner Sturm, and Rizwan Mehmood built early versions of the responsive line chart, bar chart, and parallel coordinates examples. Gregor Christandl, Simon Kloiber, Matthias Link, and Lukas Skofitsch built the initial version of the responsive scatterplot example. Aleš Smrdel refactored the code and migrated the examples to D3 v4. The SVV project was carried out in collaboration with the Provincial Government of Styria and FH Joanneum and Thomas Traunmüller developed its user interface.

REFERENCES

1. Keith Andrews. 2016. Responsive Data Visualisation. Graphical Web 2016 Talk. (2 Nov. 2016). `https://youtu.be/cQKzpKfea-E`
2. Keith Andrews. 2018. Responsive Data Visualization. (2018). `http://projects.isds.tugraz.at/respvis/`
3. Keith Andrews and Aleš Smrdel. 2017. Responsive Data Visualisation. In *Proc. Eurographics / VGTC Conference on Visualization (EuroVis 2017 Posters)*.

113–115. DOI: http://dx.doi.org/10.2312/eurp.20171182

4. Keith Andrews, Thomas Traunmüller, Thomas Wolkinger, Eva Goldgruber, Robert Gutounig, and Julian Ausserhofer. 2016. Styrian Diversity Visualisation: Visualising Statistical Open Data with a Lean Web App and Data Server. In *Proc. Eurographics / VGTC Conference on Visualization (EuroVis 2016 Posters)*. DOI: http://dx.doi.org/10.2312/eurp.20161150

5. Keith Andrews, Thomas Traunmüller, Thomas Wolkinger, Robert Gutounig, and Julian Ausserhofer. 2015. Building an Open Data Visualisation Web App using a Data Server: The Styrian Diversity Visualisation Project. In *Proc. 15th International Conference on Knowledge Technologies and Data-Driven Business (I-Know 2015)*. 48:1–48:4. DOI: http://dx.doi.org/10.1145/2809563.2809596 Best Demo Award.

6. Mike Bostock. 2018. D3. (2018). https://d3js.org/

7. Nick Diakopoulos. 2013. Data Visualization for Tablets and Touch Screens. visually blog. (2013). http://blog.visual.ly/data-visualization-for-tablets-and-touch-screens/

8. Gabriel Florit. 2013. On Responsive Design and Data Visualization. Talk at OpenVis Conf 2013. (2013). http://youtu.be/BrmwjVdaxMM

9. Bill Hinderman. 2015. *Building Responsive Data Visualization for the Web*. Wiley.

10. Christoph Körner. 2016. *Learning Responsive Data Visualization*. Packt.

11. Erin Kissane. 2013. The Boston Globe's Gabriel Florit on Responsive Visualizations. Source. (2013). https://source.opennews.org/en-US/articles/boston-globes-gabriel-florit-responsive-visualizat/

12. Gion Kunz. 2017. Chartist.js. (2017). http://gionkunz.github.io/chartist-js/

13. Peter Le Bek. 2014. Building Responsive Visualizations with D3.js. Safari Blog. (2014). https://blog.safaribooksonline.com/2014/02/17/building-responsible-visualizations-d3-js/

14. Leaflet. 2018. Leaflet. (2018). http://leafletjs.com/

15. Ethan Marcotte. 2014. *Responsive Web Design* (2 ed.). A Book Apart. http://abookapart.com/products/responsive-web-design

16. Elijah Meeks. 2014. Responsive Data Visualization. Stanford Digital Humanities Blog. (2014). https://digitalhumanities.stanford.edu/responsive-data-visualization-0

17. Ryan Nagle. 2014. Responsive Charts with D3 and Backbone. Chicago Tribune News Apps Blog. (2014). https://newsapps.wordpress.com/2014/03/07/responsive-charts-with-d3-and-backbone/

18. ORF. 2013. Nationalratswahl 2013. Austrian Broadcasting Corporation (ORF). (2013). http://orf.at/wahlergebnisse/nr13/ergebnisse/

19. Nick Rabinowitz. 2014. Responsive Scatterplot. (2014). http://nrabinowitz.github.io/rdv/?scatterplot

20. SVV. 2018. Steirische Vielfalt Visualisiert. Provincial Government of Styria. (2018). http://diversitaet.steiermark.at/steirische-vielfalt/

21. Tableau. 2017. Vizable. (2017). https://vizable.tableau.com/

22. Edward R. Tufte. 2006. *Beautiful Evidence*. Graphics Press.