

This is an continuous record of the technical challenges faced in building the project webpages, especially in the development of the visualization of data and the design of the QNRs and how solutions were found.

The purpose of this StudyBlog post is to aid the researcher in recalling how the tech-stuff achieves the outcomes it does; to bring to the attention of anyone who is interested, some excellent web-based solutions for presenting and animating webpage content, particularly data, in ways that can help in working out what it means; and to enable a log to be maintained of technical solutions that can be incorporated into the final project write-up.

1. Using javascript to create animated bullet charts

Given the development of the radar plots using a proprietary chart-plotting app (CHARTS.JS – SEE 3, BELOW), a solution was required to present the additionally collected data from the QNR, that is, the respondent’s score from the Academic Behavioural Confidence (ABC) section and their corresponding Dyslexia Index (DI).

Both of these scores are a mean average result generated from the complete results from each section. The mean average ABC score is unweighted whereas the mean average DI is a weighted mean based on data generated from the preliminary enquiry ‘DIMENSIONS OF DYSLEXIA’.

To visualize the data from the DIMENSIONS OF DYSLEXIA enquiry I created a static image showing a horizontal bar chart where each of the bars represented each of the 18 dimensions. The thumbnail shown here is representative but is linked to the project webpage that displays the full-sized image. This is a good display as it is easy to understand because it presents the data in an ‘at-a-glance’, comparative way with all dimensions proportionally drawn relative to the pseudo-size of the dataset.



The chart style lends itself ideally to the presentation of the two, principal QNR-generated scores of ABC and DI, however the prospect of creating a graduated-background, horizontal bar-chart for each of these data points for each respondent to the QNR didn't appear welcoming so a more efficient method to create something that would do the job was sought.

A GoogleSearch using the keywords 'horizontal bar chart' returned several useful results not the least of which indicated that a more appropriate category name for charts of this style is 'bullet charts' – a term I had previously been unfamiliar with and is a term that appears to have been coined by Stephen Few, who calls himself a 'visual business intelligence consultant' who's consultancy '[Perceptual Edge](#)' is based in California. His paper on Bullet Graph Design Specification is [here](#).

A further GoogleSearch to find HTML5 or javascript solutions for creating bullet charts returned many scripts written for these, most part of commercial chart-drawing packages however some excellent open-source solutions were also revealed.

I settled on an easy-to-modify script on the JSFiddle web resource created by the Fiddle Author, J L Briggs – I can find little information about him although this is typical of fiddle authors. The javascript solution he has created is straightforward and proved to be easy to modify to suit my purposes. His original jsfiddle is [here](#), also copied to my own jsfiddle library [here](#).

I tested this javascript solution by using it to recreate the static image above which demonstrated the flexibility of the script although in this case, the original image I created using the Adobe Fireworks app gets my vote as being the more accessible visualization because it is probably the easier of the two for more rapidly gaining an overview of the data it is presenting. View the animated JSFiddle version that I built and included on page 4 of the report on the DIMENSIONS OF DYSLEXIA enquiry [here](#).

2. Setting up ToolTips for the correlation display:

For the Dimensions of Dyslexia enquiry respondents to the QNR were required to judge the prevalence of each of the 18 dimensions of dyslexia listed in the QNR in their interactions with students with dyslexia.

It was felt that looking for correlations between the dimensions may reveal interesting relationships in the data. Correlations were calculated between the mean average % prevalence for each dimension from the 30 responses to the QNR that were received. Excel was used to calculate the Product-Moment correlation (R) and these were represented in a matrix of values of R.

But this matrix chart was quite hard to get to grips with not the least because it contained a good deal of data. 151 correlations were calculated representing a correlation coefficient for every possible pair of dimensions. Visual icons were used to represent each dimension with a graphic that symbolized that dimension so that the meaning of each dimension would be clear, rollover tooltips were added to the HTML code for the icons.



This was a new skill and researching how to make this happen revealed a simple string of HTML that would create the wording for the tooltip. The code included an 'a href="#" and a new attribute: 'title="..." ' in which the wording for the tooltip was inserted. So the HTML string for the complete tooltip could be, for example:

```
<a href="#" title="show evidence of poor short-term or working memory" class="tooltip">
```

Conventional CSS styling was also applied to the tooltip (class="..") and the CSS for this was written into the header of the webpage's code.

The only issue was that although this code string worked and enabled the tooltip wording to be styled, it didn't displace the standard tooltip wording that browsers displayed when the 'title' attribute was included. So a means to switch this off had to be found.

The solution was to wrap the HTML for the image inside a span tag and to give this span tag its own 'title' that was an empty string: .

Browsers then processed (parsed?) the span tag, hence providing no browser-generated tooltip but still recognized the 'title' attribute attached to the <a> which contained the intended tooltip wording.

Brilliant solution

3. The trials and tests to get the chart applications to work:

Once collected, the primary research data will be displayed in graphs, charts and diagrams because these aid understanding of the analysis. Graphs and charts present an

overview of data and enable a focus to be directed either selectively or collectively making an appraisal of either specifics, or to gain an holistic view of larger sections of the project possible.

In the MSc Pilot Study, Locus of Control Profiles were drawn by hand in a graphics application (Adobe Fireworks) which took ages. These profile diagrams were a key aid in presenting and exploring the data but were difficult to create as the basic chart required 5 axes, one represent each of the 5 constructs that the data had collected to create the LoC Profiles: learned helplessness; anxiety, regulation and motivation; self-esteem; self-efficacy; affective process.

Essentially, these profiles were merely 'radar plots' but at the time of this earlier, research project, HTML functionality was not available to create these diagrams for inclusion into webpages. Standard data analysis applications such as MS Excel and IBM SPSS were available, and indeed these were both used to number-crunch the data collected, these application were also not capable of creating 5-axes radar plots. However, for this current research project, HTML5 functionality has enabled a variety of chart-plotting applications to be developed, many of them free. After reviewing a few, two applications were chosen:

The HTML5 application [Charts.js](#) was chosen for the [radial \(radar\) charts](#) tested in the preliminary enquiry 'DIMENSIONS OF DYSLEXIA'. The [documentation](#) is clear albeit quite technical and required an understanding of HTML protocols, especially CSS syntax and options, so that the examples included in the catalogue could be formatted as required.

For the [scatter diagrams](#) in 'DIMENSIONS OF DYSLEXIA' the different applications [HighCharts](#) was used as Charts.js didn't provide this functionality. Again, an element of HTML/CSS coding ability has been required to make the scatter diagrams work and display in the desired format.

4. Getting the HTML sliders to work:

I read a good deal about Likert Scales, especially on the disagreements and controversies relating to how many data points each item response measure should include, whether this has an impact on the data it collects, whether a neutral point needs to be included and so forth. Digging deeper revealed an 'expert' in the field who appears to have contributed a good deal to this debate, he being James Carifio, currently Professor of Education, Psychology, Research and Statistics. An intriguing paper: 'TEN COMMON MISUNDERSTANDINGS, MISCONCEPTIONS, PERSISTENT MYTHS AND URBAN LEGENDS ABOUT LIKERT SCALES..' (available in the project repository [here](#)) cleared my thinking about using Likert scales a great deal. I contacted Prof

Carifio by e-mail to ask his views on my intention to use a continuously variable scale as the Likert response format in my research QNR and I was greatly encouraged by his reply which indicated that, in his view at least, 'USING PERCENTAGES IS THE WAY TO GO ... AND GIVE YOU THE CAPABILITIES OF MAKING REASONABLE RATIO LIKE STATEMENTS ABOUT YOUR RESULTS'.

Plenty of examples and development work in creating browser-based sliders is available at [CodePen](#) which calls itself 'A PLAYGROUND FOR THE FRONT END WEB' which means that it's a development sandbox for budding HTML5 authors and javascript gurus. In particular, I was taken by the range of sliders collected by Ana Tudor [here](#). These sliders are modifications to the browser-based built-in sliders that respond to the HTML (form) input type: 'range', which appears to be one of the new developments available in HTML5, now widely adopted.

Later versions of most web browsers are able to respond to this input type as it is built in to their functionality but there appears to be no cross-browser standardization relating to the way browsers render the input 'range' when displaying the webpage that contains them. I didn't discover this until well in to the development process for including this exciting way to record data onto my QNR forms.

Plenty more digging around on the web revealed quite an interest in developing these range sliders in HTML5, not the least is an interesting blog-cum-webresource by Dudley Storey, '[the new code](#)' where he dedicates a very useful blog-post to the topic [here](#).

Storey claims that the input 'range' is particularly useful for (web-based) forms where three conditions exist:

- a wide range of numbers is equally available to the user;
- a known upper and lower limit to the range can be set;
- the user is expected to "play" with their input, adjusting it frequently;

which exactly matched my design rationale for providing a continuous scale Likert response format for the Likert scales in my QNRs.

I learnt a good deal more from the [HTML5 Tutorial](#) article on input type: range, which is also an excellent resource for finding out more about the new functionality that is incorporated into HTML5.

5. Using HTML drag-and-drop functionality for the dyslexia definitions survey:

Summary coming soon

6. Server-based MySQL Gantt Chart creator DHTMLXGANTT for the Gantt Chart project overview:

Summary coming soon

7. Using SMOOTHDIVSCROLL

Summary coming soon