Worcester Polytechnic Institute

CS 480X Data Visualization

Expanding Understanding Of Graphical Perception : An Extension To Theory, Experimentation, and Application to the Development of Novel Graphical Methods

Authored by:
Dembski, Clayton
Fischler, Mikayla

Advised by:
Professor Lane Harrison

Abstract

This study expands on the classic *Graphical Perception* study done by Cleveland and McGill. Using a bar chart as validation with the original paper, this study uses Lollipop Charts, Tree Charts, and Circular bar charts to examine the accuracy by which people can interpret visual encoded data. 200 data points of each graph was collected. For each graph, a 95% bootstrapped confidence interval was constructed, using 800 resamplings.

The bar chart was most accurate, with a mean log loss of 1.98. The lollipop chart was second, with a mean log loss of 2.15. The circular bar chart was third at 2.42, and the treemap was fourth with 2.83. The site with the study can be found here: https://cabincabin.github.io/03-Experiment/web/

Methodology

The construction of the experiment can be sectioned into two different pieces: a front end, where the users see the graphs for interpretation, and a backend database where the users response, along with the true graph values, are stored for processing.

The page for the experiment is shown below in figure 1. It consists of a title, the trial number out of the total necessary trials in the top right, and the form input at the bottom. The user enters a value from 1 to 100 and press enter or click to submit, which stores the data in a database in a remote server, and generates the next chart. Charts are randomly generated and randomly ordered, but are always 10 of each type and A will never be larger than B.

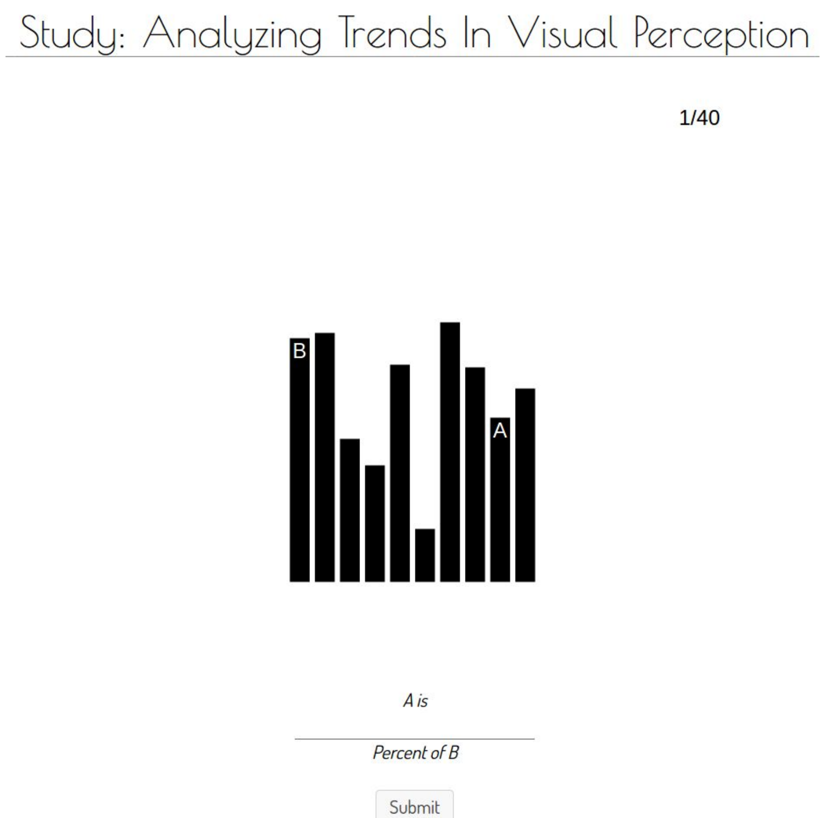

Figure 1: User Interface For Running Experiments

Front End

For this project, four charts were used, a bar chart, a circular bar chart, a treemap, and a lollipop chart. We expected most users to have seen a bar chart before. The height is directly representative of the value of the bar. In addition, this was the baseline set up by the *Graphical Perception* study done by Cleveland and McGill. The lollipop chart is represented in almost the same way as the bar chart, except that the majority of the bar is made up of a skinny, less intrusive line, and the bar ends at the top of a circle which brings attention to the end value of the chart. This type of chart is meant to reduce visual clutter. The circular bar chart is another form of bar chart. The end height, or in this case radius, is still the value read by the participant, however the chart is radially aligned, starting from another section on a shared circle, instead of linearly aligned. Finally we used a treemap. In a treemap, each area is directly proportional to the value that the data point represents. For the tree map, users needed to find ratios of areas, instead of lines.

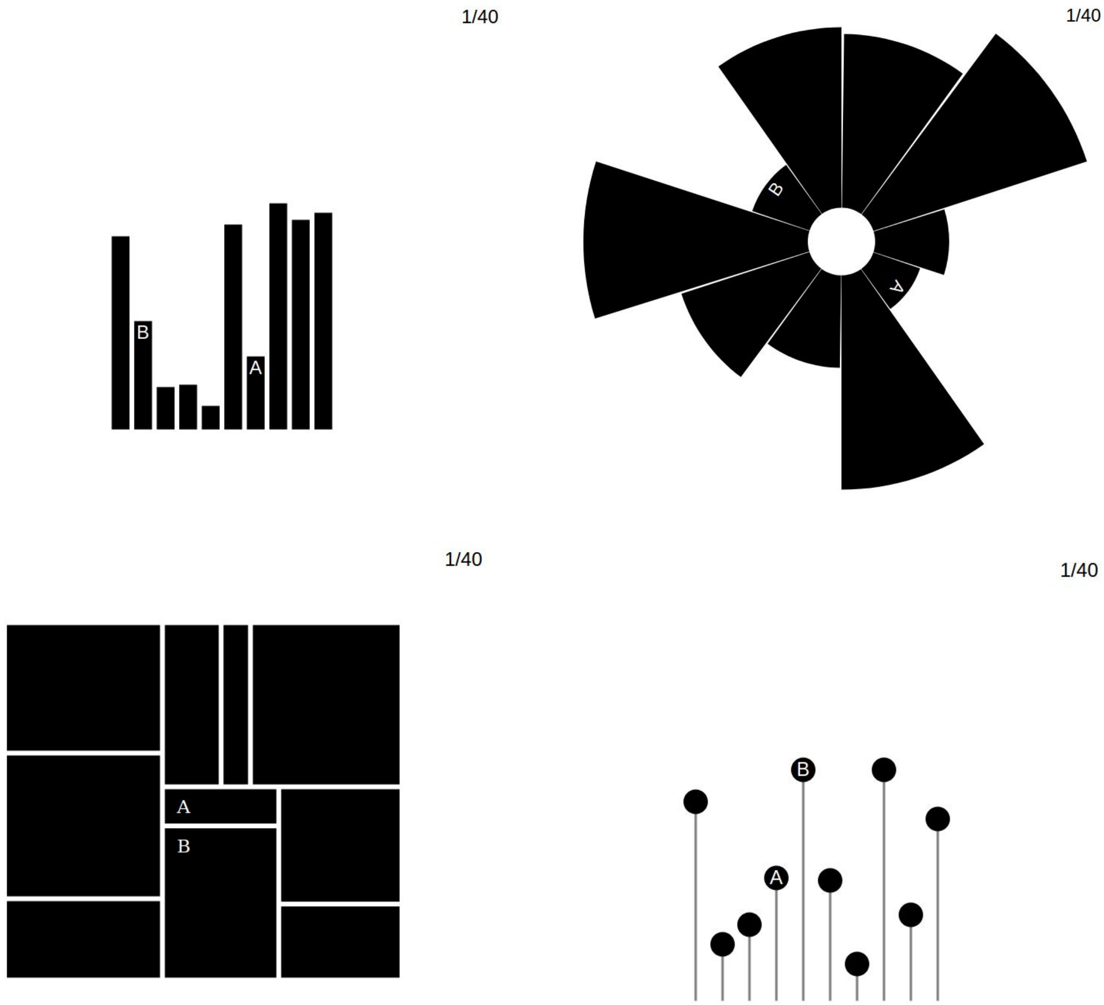

Figure 2: Chart Types. From Left To Right, Top to Bottom - Bar Chart, Circular Bar Chart, Treemap, Lollipop Chart

The standard requirements for our study required a minimum of 200 data points for each chart, and to use a minimum of at least 10 participants. The problem with this was that, by using only 10 participants, as we had 4 charts, each participant would have to complete 80 trials. While each trial only took a few seconds, and completing all trials would take less than 10 minutes, the sheer number seemed to be overwhelming to most participants. We then decided to reduce the total number of trials, from 80 to 40, and expand the total number of participants from 10 to 20. By reducing the number of trials to a, still large, but not overwhelming number, we lowered the barrier of entry for our participants. In this way, more people were willing to participate in the trials, and each person had much less exposure fatigue.

The label above the entry field originally stated Ratio of A/B (Value From 1 to 100). During the first round of experimentation, however, the participant did not realize that they needed to represent the fraction as a percentage (IE Multiply by 100), due to the language. Because of this, the entry box was updated to say:

```
A is
___
percent of B
```

In this way, users would know to represent their value as a percent, and know not to add a % symbol after their integer number.

The entry space and submit button auto focused, meaning, instead of clicking back and forth between the entry percentage and the submit button, users only needed to hit enter to submit their observed value, and then could immediately type in their next observed value, without needing to reselect the box.

When the page for the study was opened, a list of 40 values were generated, representing 10 of each of the graphs. When a new graph was needed (On opening the page, and after each submission), this list was randomly sampled, and the element was removed, ensuring that a participant, over 40 trials, would do exactly 10 of each graph. On graph creation, 10 values were randomly generated. Each of these values were from 10 to 100. This was to confirm that the ratio of A:B would never be infinite, and that each visualization of the datapoint would be large enough to be clearly labeled.

On submission of a single value, the front end would then store the user's id, generated on entering the web page, the type of chart, the trial number, the actual value of the ratio of A:B, stored as a percentage, and the observed value given by the participant. This was then posted to a hosted server.

Back End

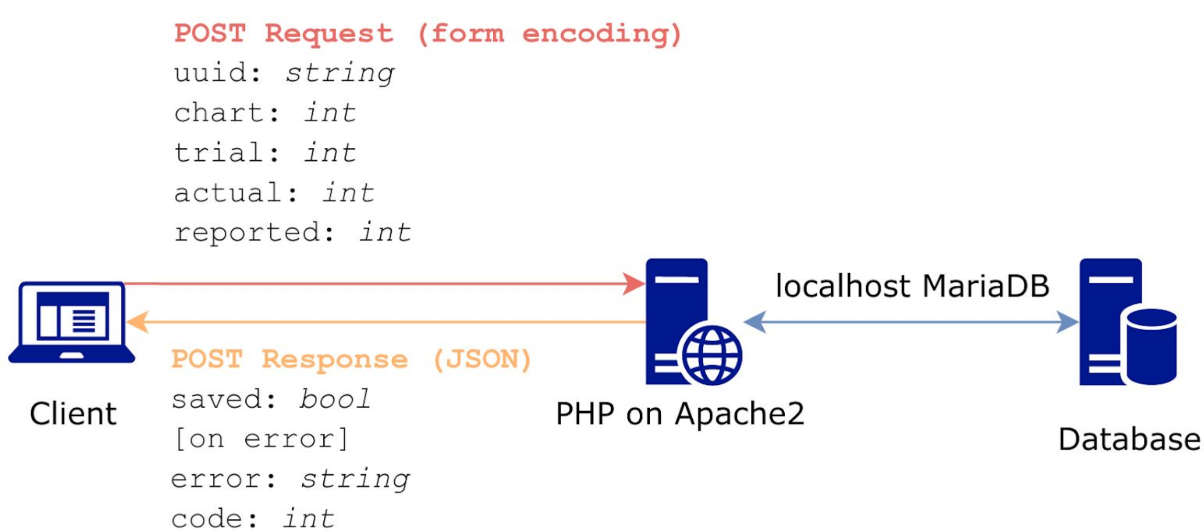

Figure 3: Reporting Server Setup

To streamline the experimental process, we set up a MariaDB database and Apache2 web server on a personal server to automate response recording. Each time the user submits a value for a particular chart, a POST request containing the relevant information is sent to the server which is handled by a short PHP script, which validates the input then records it to the database. It returns an HTTP response code indicating success or the type of failure along with JSON encoded status information. The JSON status includes whether or not the data was successfully recorded, and if false, it includes a string indicating the error type along with an error code indicating which specific operation failed. This was primarily for debugging but would also allow us to quickly fix any issues that arose during the experiment.

To fetch the data from the database, a PHP report script goes through the database and generates a CSV representing the data, which includes replacing NULLs with NAs, which was necessary as we had to nullify a couple values due to instructional misunderstanding.

Results And Discussion

The study was made up of 20 people. Each person answered 40 individual questions: A is what percentage of B. Of these 40 questions, 10 were the standard bar charts, 10 were lollipop charts, 10 were circular bar charts, and 10 were treemaps. 200 points of data were collected for each of the 4 charts, each made up of the participant id, the trial number, the type of visualization, the true graph percentage, and the user entered estimation.

The sample error for each point was calculated using the log loss formula in *Graphical Perception* and perfect observations were given a value of 0:

$$Loss = log_2(|Observed - Actual| + 1/8)$$

The bootstrapped confidence interval for each chart type was calculated with 800 resamples from the initial table.

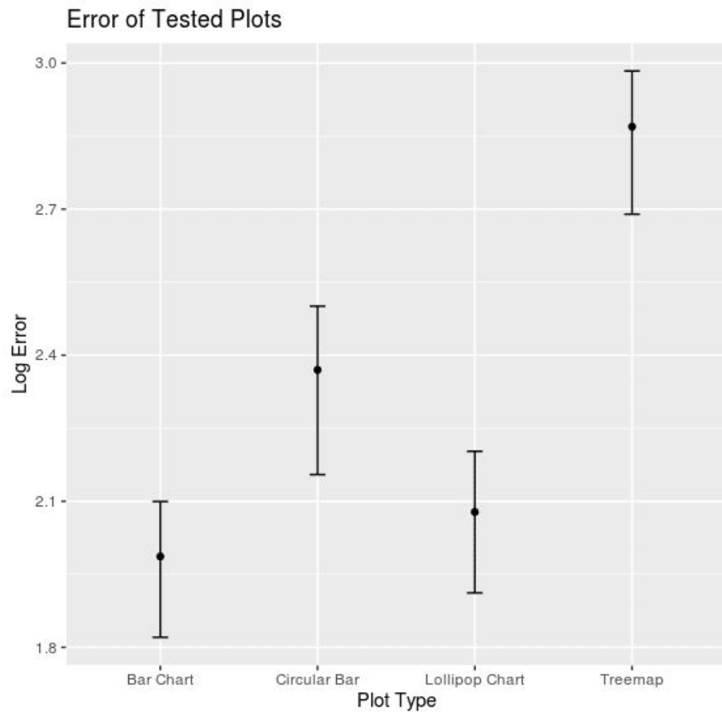

Figure 4: 95% Bootstrapped Confidence Interval of Log Error

The bar chart most effectively displays the data for interpretation, followed closely by the lollipop chart, then the circular bar chart, and finally, by a large margin, the treemap. One sees that, for each of the confidence intervals, the error margins are left skewed: skewed towards lower error.

A simple explanation for the data skew lies directly in the encoding of the error. As the scale is logarithmic, to have a proportionally higher data point on the graph to be sampled, the difference between the observed and actual values must grow exponentially.

For example, to get from 0 to 1, the diff must be

|O-A| =1.875

log2(|O-A|+ 1/8)=1

From 1 to 2

|O-A| = 3.875

log2(|O-A|+ 1/8)=2

From 2 to 3

|O-A| = 7.875

log2(|O-A|+ 1/8)=3

Looking at the same bootstrapped dataset without a log scale, just the absolute difference of the observed and actual values, we see that the confidence interval is instead slightly left skewed: the subjects tended to have a lot of tightly packed lower error guesses, with some far spread, very erroneous guesses.

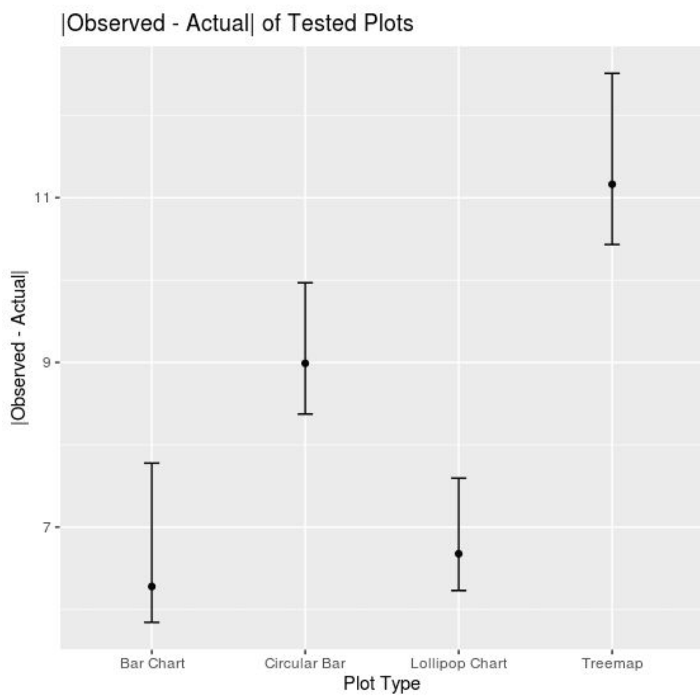

Figure 5: Difference Chart - Chart of Observed Minus Actual Difference

Looking at this Difference chart, we can also get a sense of how far off the standard guess was. The bootstrapped means were: Bar - 6.279621, Circular - 8.989336, Lollipop - 6.677536, Treemap - 11.16299.

The bar chart had the lowest error. This was to be expected. The data in the bar chart was aligned, and this chart was comfortable to most users. The lollipop chart had the second lowest error. Two explanations can be provided. First, the representation itself is slightly uncommon. Most users have not directly seen it, so it takes them a moment, or a few tries, to understand how to read the data. Second, the tops of the circles for the lollipop charts are rounded. The user must try to approximate a tangent line on an object with curvature, which, as seen in the original paper, is slightly more difficult.

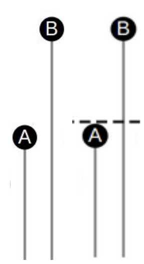

Figure 6: How users Must Interpret The Lollipop Graph

The circular bar graph experienced the next jump in difficulty. This is once again explained by curvature. While each bar is technically aligned, they are aligned radially. Without a marking line indicating what percentage is at what radius, users must follow an arc from A to B to align their data to estimate the percentage. Once this is done, however, the circular bar becomes more linearly aligned and the users may more easily estimate the amount.

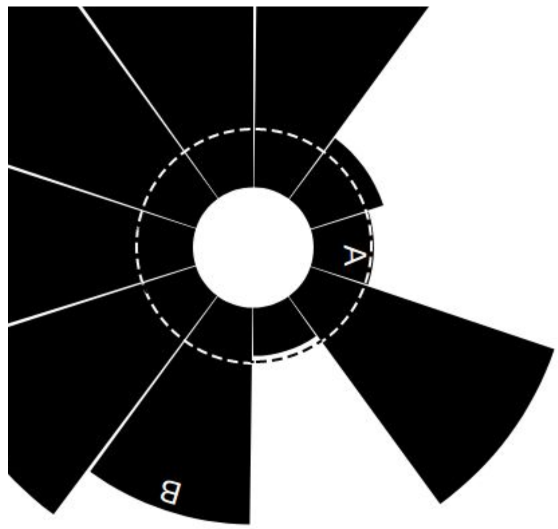

Figure 7: Estimating the Height of A Radially With B

Finally, the treechart was by far the worst. As seen in *Graphical Perception,* humans are significantly worse at measuring systems with two dimensions, such as area. Our results confirm this. It was incredibly difficult to compare rectangle sizes, especially as, most of the time, the sidelengths were not comparable to one another. One thing of particular note were the situations where A appeared to be larger (contain more area) than B, even though this was strictly not possible by the implementation. Users were not physically able to enter over 100%, however it was observed that some users tried to for certain scenarios.

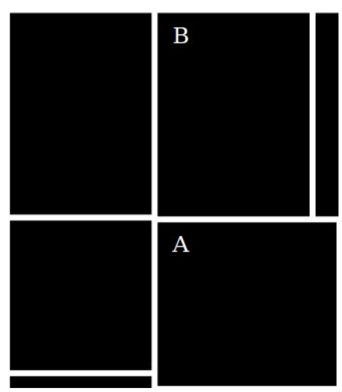

Figure 8: Similar Scenario to One Which Participant Thought A was Larger Than B

The demographic this study was given to was predominantly western college students focused in STEM, many of whom were also computer scientists. In addition, as the students were chosen were from people affiliated with the group, the sampling was not random, leading to another innate bias in the demographic. A novel extension of this study may be to extend the participants of the study either to those who go to an Arts or Liberal Arts college, or to have nonwestern participants with a different set of language skills and innate visual encoding styles.

At the start of the experiment the participant misunderstood the instructions, so the first two values from that user had to be nullified, resulting in the two NA values in our dataset. We made resulting adjustments to the interface to reduce confusion and also changed how we explained the experiment to the subjects, which was successful.

It is worth mentioning that there were some roadblocks to implementing the recording server, most notably the problem with Cross-Origin Resource Sharing (CORS). To allow cross-domain POST requests, some configurations needed to be made and were initially met with failure when trying to implement them on an NGINX server, so a switch was made to Apache. After that switch, the configuration was more straightforward and was achieved. Additionally, a php.ini modification was needed as the MySQL socket the MariaDB server was using was not the default, which meant PHP could not connect even though the address and port were correct. Once resolved, everything else worked as desired.