Collection Data Visualization: Seeing the Forest Through the Treemap

Geoffrey P. Timms
Mercer University, timmsgp@gmail.com

Jeremy M. Brown
Mercer University, brown_jm@mercer.edu

Geoffrey P. Timms and Jeremy M. Brown, "Collection Data Visualization: Seeing the Forest Through the Treemap" (2014). *Proceedings of the Charleston Library Conference.*
http://dx.doi.org/10.5703/1288284315636

Collection Data Visualization: Seeing the Forest Through the Treemap

Geoffrey P. Timms, Systems Librarian, Mercer University Libraries
Jeremy M. Brown, Associate Director for Technical Services and Systems, Mercer University Libraries

Abstract

Collection management is one of the more complicated responsibilities in librarianship. In this task, the librarian must simultaneously synthesize the needs, desires, and aspirations of the institution, departments, and individuals. While much of this is elusive qualitative data that may not yield a definitive answer, we also have increasingly accessible hard data from our integrated library systems (ILSs) that we can synthesize to complement it. In the latest generations of ILSs, this information is readily available to use for statistical analysis and visualization. When it comes to our increasingly limited materials budgets, it is important to make sure that we make the best decisions possible, thus it is advantageous to analyze all the data at our disposal. We introduce a web application that produces live statistics from the ILS. The system uses data points, including collection use and metrics, which describe a collection (e.g., age, quantity). This system goes beyond traditional charts and graphs by employing several visualization techniques that lend a unique perspective to these data points. The particular techniques allow collection managers to visualize multiple data points simultaneously and reveal data correlations that might not otherwise be obvious.

Manually analyzing data from the Integrated Library System (ILS) is a challenging task. As we move from broad data points deeper into the classification system to study particular subsets of the collection, the numbers seem incessant and the data is hard to translate into an understanding of the collection's characteristics and use. This is where data visualization techniques can be applied to graphically describe data in a meaningful way, conducive to human interpretation. When data is presented graphically the absolute numbers, while used to generate the graph, are not the focus of the output. The emphasis of data represented graphically is identifying trends and comparing data points. For the purpose of comparing subsections of our library collection, we focused on two techniques for representing data graphically: treemaps and cartograms.

A treemap is a constrained proportional graphical representation of data. Perhaps the most-recognized treemap is the squarified treemap where a square represents the data in its entirety and individual subsets of data are proportionally represented by scaled rectangles contained therein and sorted from top left to bottom right in diminishing size order. In addition, a second tier or subset of data can be proportionally represented within each of those rectangles as another series of scaled rectangles. In theory this could continue ad infinitum but in reality, it is hard to derive meaning from more than two tiers of data presented simultaneously. A second data point can be represented using varying shades of color. An example of a treemap representing votes cast by county, state, and recipient (Obama [blue] vs. Romney [red]) in the US Presidential Elections of 2012, is demonstrated in Figure 1.

A cartogram is constructed on similar visual principles to a treemap insomuch as data can be represented graphically and a second data point can be represented with shades of color. The primary distinction between a cartogram and a treemap is that a cartogram can be any shape and may be distorted in its entirety to represent data proportionally. This is not necessarily always the case, however, as the overall shape of the image can be maintained by using noncontiguous graphics to represent the first tier of data within the confines of the overall image. An example of a contiguous area cartogram is a map of the United States, with each county rescaled in proportion to its population. Colors refer to the results of the 2012 U.S. presidential election, as demonstrated in Figure 2.

http://dx.doi.org/10.5703/1288284315636

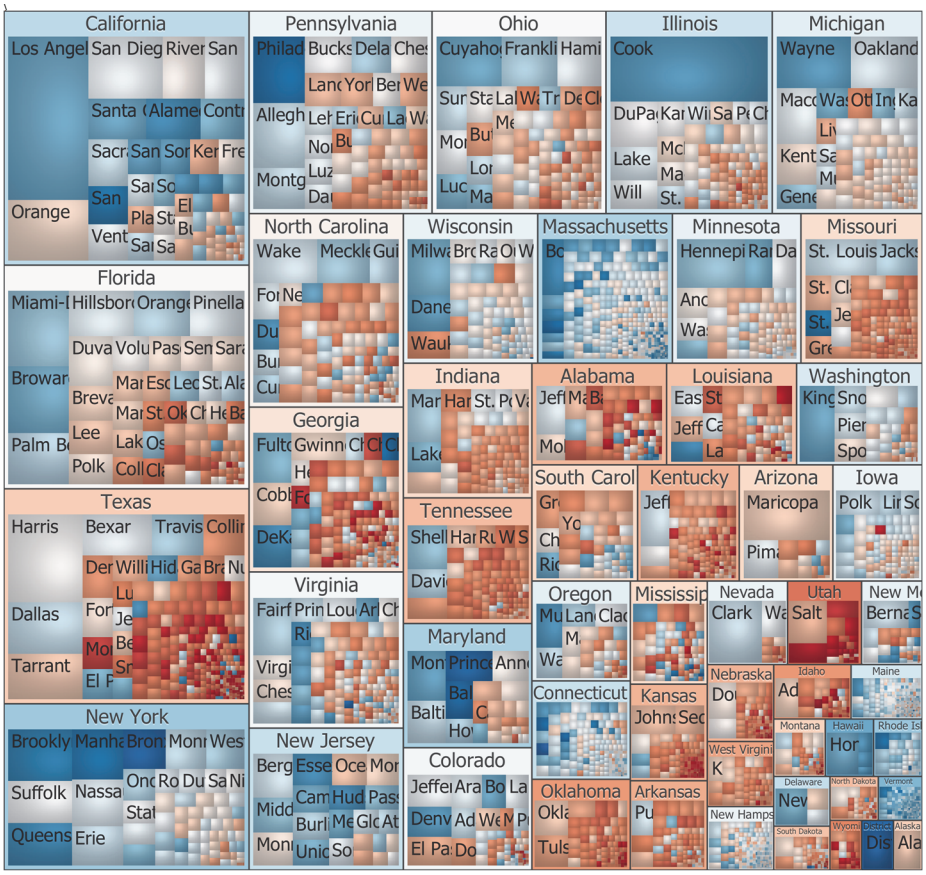

Figure 1. Treemap of votes by county, state, and locally predominant recipient (Obama blue, Romney red in the US Presidential Elections of 2012). Reprinted from Treemapping, In *Wikipedia*, 2012, Retrieved October 28, 2014, from http://en.wikipedia.org/wiki/Treemapping. Copyright 2012 by Luc Girardin. Reprinted with permission. CC BY 3.0.

Figure 2. Cartogram of the United States, with counties sized according to their population and colors demonstrating the percentage of votes cast for Obama (blue) and Romney (red) in the 2012 U.S. presidential election. In *Maps of the 2012 US Presidential Election Results,* 2012, Retrieved October 28, 2014, from http://www-personal.umich.edu/~mejn/election/2012/. Copyright 2012 by Mark Newman. Reprinted with permission. CC BY 2.0.

Application Design

Mercer University Libraries' Systems and Technical Services Unit operates Innovative Interfaces Sierra ILS. With our recent upgrade to Sierra, we gained the opportunity to query elements of the database to access data about the collection. This can be performed in dynamically, using scripted procedures initiated from within a web interface. We undertook to develop a web application which would provide the user with the opportunity to choose two data points describing the collection and its use, and which would then query the database and generate either a squarified treemap or a hybrid cartogram to visually represent the data.

The web application is created in Python with a lightweight underlying CherryPy web framework. The squarified treemap coordinates are generated from the raw data using Uri Laserson's Squarify Python library (https://github.com/laserson/squarify) and both the treemap and cartogram are drawn and color-enhanced using Python Imaging Library (http://www.pythonware.com/products/pil/). The treemap is a squarified graphic sorted in diminishing order by primary data point size, while the cartogram simulates ranges of library shelves and maintains sorting in call number order.

As an academic library, our collection is organized using Library of Congress (LC) call numbers. In order to facilitate convenient visual interpretation we only present one tier of data at a time, starting with the primary call numbers A, B, C, etc. We enable the user to proceed to the next tier of call number data by clicking on the graphic. Thus, by selecting P, for example, a new graphic will be drawn presenting the data for call number P and its subsets P, PA, PB, etc. The user can continue further into the third tier, PA for example, where the data will be subdivided into call numbers 0-999, 1000-1999, 2000-2999, and so on. These can be further subdivided into 100s and 10s.

We provide the user multiple data points by which to evaluate and describe the collection. Each of these can be represented either by size or shade of color on the graphic. The data points available are:

- Collection size (item count).
- Last year's total use.
- Last year's average use.
- Year to date total use.
- Year to date average use.
- Average publication year (relative age indicator).
- Renewal total.
- Renewal average.
- Total usage (all years).
- Average total use (all years).
- Average number of uses per item.
- Percentage of items with no circulations.
- Percentage of items with at least one circulation.
- Percentage of items with more than one circulation.

Data Interpretation

Figures 3 and 4 demonstrate the treemap and cartogram respectively, presenting the same two select data points on the graphics. In this example, *collection size* is chosen as the primary data point and *percentage of collection circulated at least once* as the secondary data point.

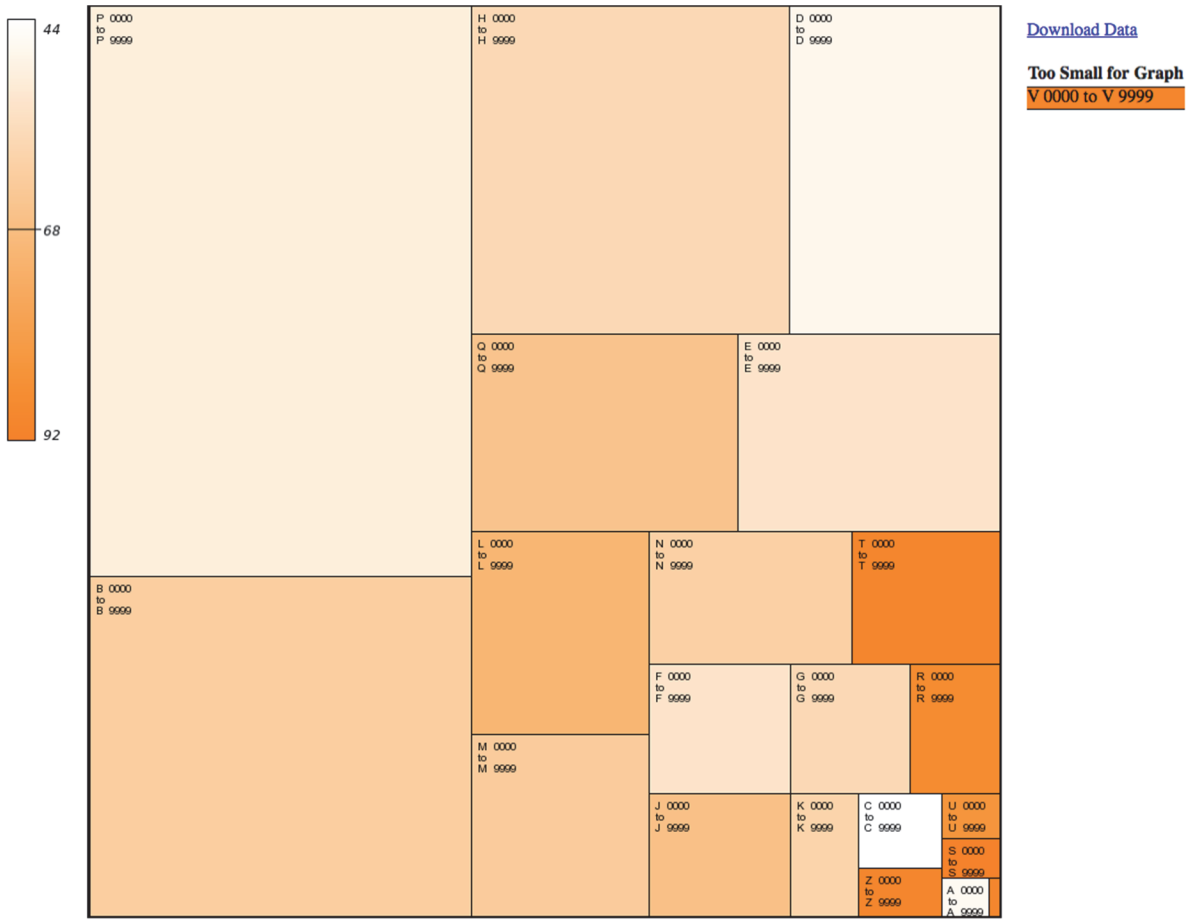

Figure 3. Treemap with size representing collection size (item count) and color representing percentage of items circulated at least once.

The primary data point, represented by the relative size of the boxes in the treemap and of the shelf sections (which may span more than one shelf range) in the cartogram, demonstrates the relative size of each call number range in the context of the total collection size. The treemap presents the call number ranges in order of diminishing item count, emphasizing the largest call number range at the top left of the treemap with the largest box and the smallest call number range at the bottom right with the smallest box. The cartogram maintains the alphabetical call number order with the section size varying in situ.

The secondary data point, represented by the shade, demonstrates the percentage of items circulated at least once in each call number range and ranges from white to a deep orange with increasing percentage value. This is true regardless of the data type. In this example, the shade represents the percentage of items circulated at least once. Therefore, white represents the lowest percentage of items circulated at least once and full-intensity orange represents the highest percentage of items circulated at least once, for the collection assessed.

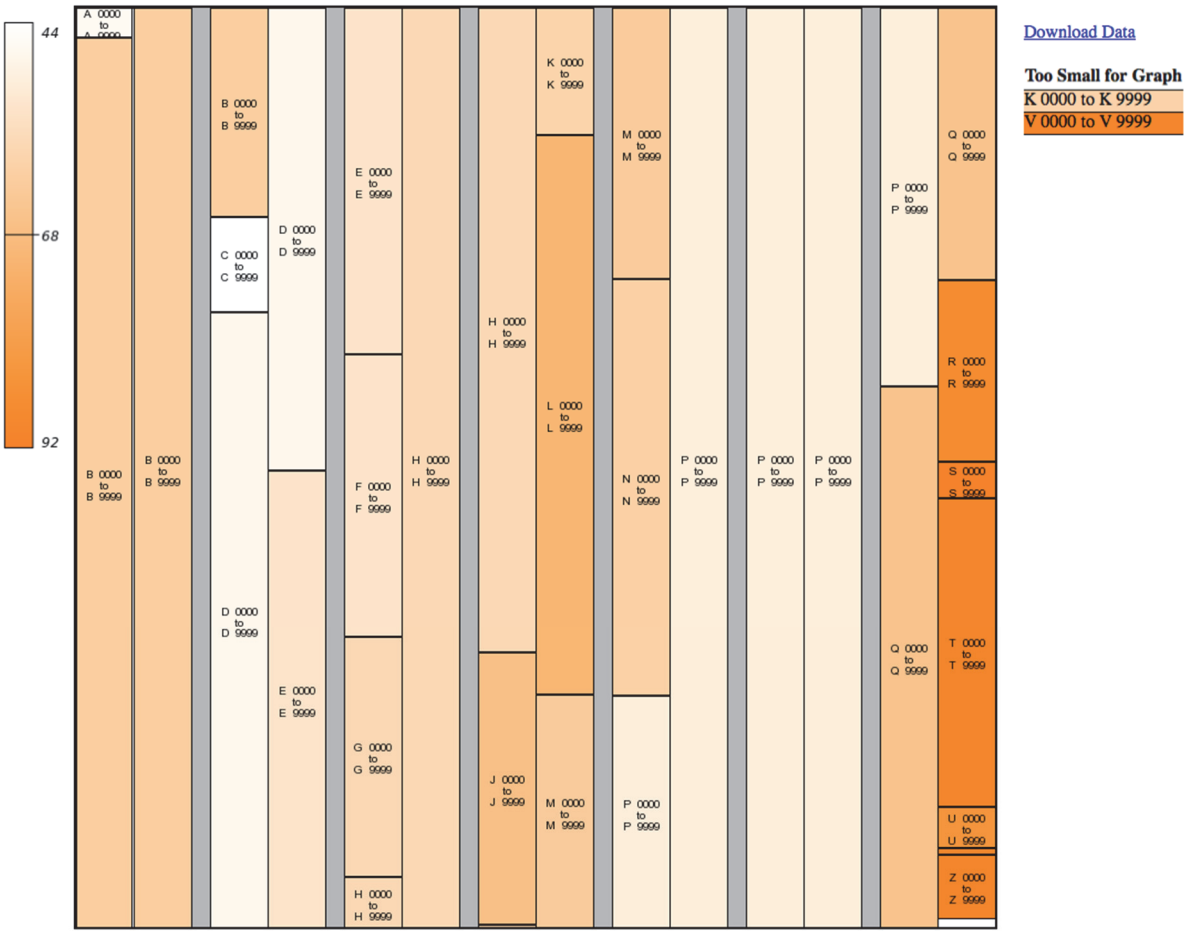

Figure 4. Cartogram with size representing collection size (item count) and color representing percentage of items circulated at least once.

Similarly, if the shade is chosen to represent percentage of items uncirculated then white represents the lowest percentage of items uncirculated and full-intensity orange represents the highest percentage of items uncirculated for the collection assessed. Even though the two concepts described have a completely inverse relationship, the color is determined by percentage value increasing, from white to orange. It is, therefore, very important to maintain awareness of what is actually being represented by the graphic.

The gradient scale to the left of the image identifies the lowest and highest values as well as the median value to assist with interpretation. The values may be absolute or percentages depending upon the secondary data point selected. In either case the lowest value, represented by white, is not zero unless the lowest absolute or percentage value represented is actually zero. Similarly, if percentages are represented, the highest percentage represented by intense orange is not always 100% unless the highest value represented is actually 100%. It will always represent the highest absolute or percentage observed in the data. We see from the gradient scale that, of the call number ranges assessed, the lowest percentage of items circulated at least once is 44% and the highest is 92%.

Figure 5. Treemap with size representing percentage of items circulated at least once and color representing collection size (item count).

A particularly interesting perspective on the size and use of the collection is to reverse the primary and secondary data points, as seen in Figures 5 and 6. In doing so, size represents the percentage of items circulated at least once and color represents collection size. No new information is presented compared to Figures 3 and 4, but the different perspective on the same information is notable.

To assist with interpretation of the visual information, we provide a link to download the raw data used to create the graphic as an Excel spreadsheet. This adds context to the analysis of the graphics. The data used to generate Figures 3-6 is shown in Table 1.

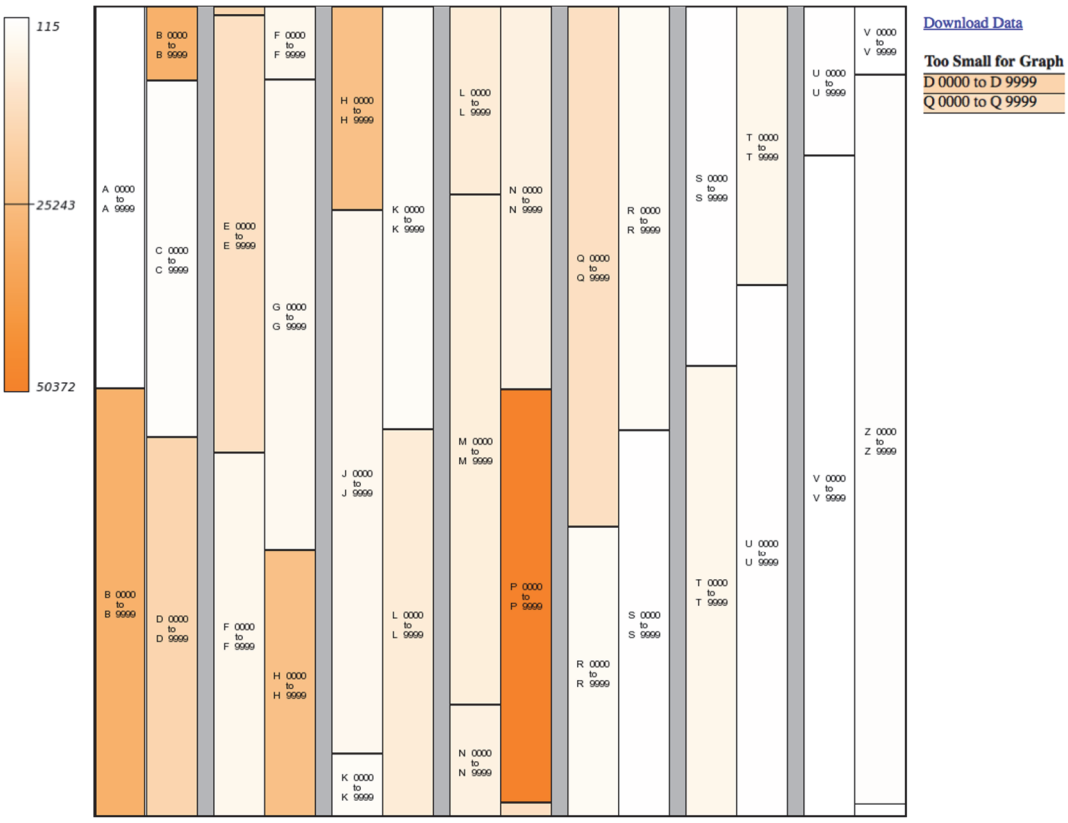

Figure 6. Cartogram with size representing percentage of items circulated at least once and color representing collection size (item count).

Challenges

At the beginning of this project, we encountered some interesting obstacles. Once we had adequately described the problem we were attempting to help solve, and what kinds of visualizations we wanted to implement, we prepared to tackle those.

Call Number	Percent with 1+ Circulations	Count
A	47	438
B	62	30178
C	44	1434
D	48	16115
E	54	12016
F	54	4231
G	58	3576
H	58	24162
J	67	4048
K	60	1978
L	71	8361
M	63	7545
N	61	6241
P	51	50372
Q	66	12162
R	88	2724
S	92	546
T	90	4611
U	84	627
V	90	115
Z	90	965

Table 1. Raw data showing percentage of items circulated at least once and collection size by call number.

Data acquisition was of primary importance. While Sierra can be queried directly, the volume of data being queried is large and structured in such a way that that any data query would take just under two minutes to run. At that point, we would begin to be able to conduct calculations upon that data to produce visualizations, and that is an unacceptable amount of time. To mitigate this issue, we created a PostgreSQL database on another server and provided sufficient indexes to optimize the queries used in the application. As a result, the query run time was reduced to a fraction of a second. The data stored in Postgres is refreshed periodically by querying Sierra using a system cron job. This still takes nearly two minutes to run, but with transactions, a user can continue to work on old data while the data is being refreshed. The application, therefore, does not query Sierra in real time but utilizes data that is sufficiently current for our purposes.

With proportional representation of data on the graphic and call numbers written within each section to identify the range covered, it was inevitable that some sections would be too small in which to list the call numbers. We elected to provide a table to the side of the graphic to list those sections. This was particularly important, as each section of the graphic is clickable to enable the user to delve deeper into the data. We made the items listed in the table clickable also, so that the user could be sure to access the desired data.

Larsen's Squarify Python library did not account for zero-value data, which certainly occurs within the data points utilized. Some subsections of the collection, for example, did not see any use. This resulted in calculation errors in generating the data used to draw the treemap. We adapted the script to extract and exclude data that would result in a zero-size square, presenting it in the table along with sections where the insufficiently sized sections were represented.

Future Enhancements

Opportunities exist to optimize data analysis performance further. When we added the last three data columns: percent of items with no circulation, items with circulations, and items with more than one circulation, we encountered a problem. The original query merely aggregated data, either by summation or average, and it ran in under a half second. Producing data for each of the final three items requires each to run as separate queries. Each of those would require approximately a half second to run, and when run over all of the rows in the original query, we added seconds on to the original run time. We mitigated much of the performance issues by running it on more robust hardware, but further work on the query or postprocessing available data might speed that up to run in under a second.

Although we provide two different visualizations and a Microsoft Excel export option, many users might prefer other visualization methods. Some users will prefer something like a bar graph, or perhaps pie charts. Some users might prefer an in-browser tabular data display as well. All of these visualization methods are implementable, given sufficient time.

Our query is specific to the Innovative Sierra database schema. However, we designed the program to make it easy to adapt to other integrated library systems. The data extraction is compartmentalized in its own Python object, and each charting object's data needs are minimal and uniform. That is, each graphed call number range is represented by a data tuple with three elements: the call number range, data for the first element, and data for the second element. This is passed to the data objects as a list of tuples.

LoC call number ranges representing sub-disciplines are comprehensive and specific. We would like to map the call number ranges to their subject descriptors and use that structure to define the data divisions displayed in the graphics by discipline and sub-discipline. The task of making the structure of call numbers and descriptors programmatically traversable, however, is of significant magnitude.

Conclusion

Data visualization does lend a unique perspective to the analysis of a library collection. It is

important to note that context is the filter through which visual data must be interpreted and, therefore, data visualization alone does not paint the entire picture. It serves best to highlight areas of a collection where further investigation might be prudent in order to ascertain whether a change in collection development strategy is needed. Pairing visual (relative) data with raw data will help the librarian determine a more complete picture of the nature and use of the collection. With modest investment of time and skills, a data visualization tool can be developed to serve the needs of librarians who address collection development and analysis.