

Trends at a Glance: A Management Dashboard of Library Statistics

Emily Morton-Owens
and Karen L. Hanson

ABSTRACT

Systems librarians at an academic medical library created a management data dashboard. Charts were designed using best practices for data visualization and dashboard layout, and include metrics on gatecount, website visits, instant message reference chats, circulation, and interlibrary loan volume and turnaround time. Several charts draw on EZproxy log data that has been analyzed and linked to other databases to reveal use by different academic departments and user roles (such as faculty or student). Most charts are bar charts and include a linear regression trend line. The implementation uses Perl scripts to retrieve data from eight different sources and add it to a MySQL data warehouse, from which PHP/JavaScript webpages use Google Chart Tools to create the dashboard charts.

INTRODUCTION

New York University Health Sciences Libraries (NYUHSL) had adopted a number of systems that were either open-source, home-grown, or that offered APIs of one sort or another. Examples include Drupal, Google Analytics, and a home-grown interlibrary loan (ILL) system. Systems librarians decided to capitalize on the availability of this data by designing a system that would give library management a single, continuously self-updating point of access to monitor a variety of metrics. Previously this kind of information had been assembled annually for surveys like AAHSL and ARL.¹ The layout and scope of the dashboard was influenced by Google Analytics and a beta dashboard project at Brown.²

The dashboard enables closer scrutiny of trends in library use, ideally resulting in a more agile response to problems and opportunities. It allows decisions and trade-offs to be based on concrete data rather than impressions, and it documents the library's service to its user community, which is important in a challenging budget climate.

Although the end product builds on a long list of technologies—especially Perl, MySQL, PHP, JavaScript, and Google Chart Tools—the design of the project is lightweight and simple, and the number of lines of code required to power it is remarkably small. Further, the design is modular. This means that NYUHSL could offer customized versions for staff in different roles, restricting the display to show only data that is relevant to the individual's work.

Because most libraries have a unique combination of technologies in place to handle functions like circulation, reference questions, circulation, and so forth, a one-size-fits-all software package that

Emily Morton-Owens (emily.morton.owens@gmail.com) was Web Services Librarian and **Karen Hanson** (Karen.Hanson@med.nyu.edu) is Knowledge Systems Librarian, New York University Health Sciences Libraries, New York.

could be used by any library may not be feasible. Instead, this lightweight and modular approach could be re-created relatively easily to fit local circumstances and needs.

Visual Design Principles

In designing the dashboard, we tried to use some best practices for data visualization and assembling charts into a dashboard.

The best-known authority on data visualization, Edward Tufte, states “above all else, show the data.”³ In part, this means minimizing distractions, such as unnecessary gridlines and playful graphics. Ideally, every dot of ink on the page would represent data.

He also emphasizes truthful proportions, meaning the chart should be proportional to the actual measurements.⁴ A chart should display data from zero to the highest quantity, not arbitrarily starting the measurements at a higher number, because that distorts the proportions between the part and the whole. A chart also should not use graphics that differ in width as well as length, because that causes the *area* of the graphic to increase incorrectly, as opposed to simply the *length* increasing. Pie charts are popular chart types that have serious problems in this respect despite their popularity; they require users to judge the relative area of the slices, which is difficult to do accurately.⁵ Generally, it is better to use a bar chart with different length bars whose proportions users can judge better.

Color should also be used judiciously. Some designers use too many colors for artistic effect, which creates a “visual puzzle”⁶ as the user wonders whether the colors carry meaning. Some colors stand out more than others and should be used with caution. For example, red is often associated with something urgent or negative, so it should only be used in appropriate contexts. Duller, less saturated colors are more appropriate for many data visualizations.

A contrasting style is exemplified by Nigel Holmes, who designs charts and infographics with playful visual elements. A recent study compared the participants’ reactions to Holmes’ work with plain charts of the same data.⁷ There was no significant difference in comprehension or short-term memorability; however, the researchers found that the embellished charts were more memorable over the long term, as well as more enjoyable to look at. That said, Holmes’ style is most appropriate for charts that are trying to drive home a certain interpretation. In the case of the dashboard, we did not want to make any specific point, nor did we have any way of knowing in advance what the data would reveal, so we used Tufte’s principles in our design.

A comparable authority on dashboard design is Stephen Few. A dashboard combines multiple data displays in a single point of access. As in the most familiar example, a car dashboard, it usually has to do with controlling or monitoring something without taking your focus from the main task.⁸

A dashboard should be simple and visual, not requiring the user to tune out extraneous information or interpret novel chart concepts. The goal is not to offer a lookup table of precise values. The user should be able to get the idea without reading too much text or having to think

too hard about what the graph represents. Thinking again of a car, its speedometer does not offer a historical analysis of speed variation because this is too much data to process while the car is moving. Similarly, the dashboard should ideally fit on one screen so that it can be taken in at a glance. If this is not possible, at least all of the individual charts should be presented intact, without scrolling or being cramped in ways that distort the data.

A dashboard should present data dimensions that are dynamic. The user will refer to the dashboard frequently, so presenting data that does not change over time only takes up space. Better yet, the data should be presented alongside a benchmark or goal. A benchmark may be a historical value for the same metric or perhaps a competitor's value. A goal is an intended future value that may or may not ever have been reached. Either way, including this alternate value gives context for whether the current performance is desirable. This is essential for making the dashboard into a decision-making tool.

Nils Rasmussen et al. discuss three levels of dashboards: strategic, tactical (related to progress on a specific project), and operational (related to everyday, department-level processes).⁹ So far, NYUHSL's dashboard is primarily operational, monitoring whether ordinary work is proceeding as planned. Later in this paper we will discuss ways to make the dashboard better suited to supporting strategic initiatives.

System Architecture

The dashboard architecture consists of three main parts: importer scripts that get data from diverse sources, a data warehouse, and PHP/JavaScript scripts that display the data. The data warehouse is a simple MySQL database; the term "warehouse" refers to the fact that it contains a stripped-down, simplified version of the data that is appropriate for analysis rather than operations. Our approach to handling the data is an ETL (extract, transform, load) routine. Data are *extracted* from different sources, *transformed* in various ways, and *loaded* into the data warehouse. Our data transformations include reducing granularity and enriching the data using details drawn from other datasets, such as the institutional list of IP ranges and their corresponding departments. Data rarely change once in the warehouse because they represent historical measurements, not open transactions.¹⁰

There is an importer script customized for each data source. The data sources differ in format and location. For example, Google Analytics is a remote data source with a unique Data Export API, the ILL data are in a local MySQL database, and LibraryH3lp has remote CSV log files. The scripts run automatically via a cron job at 2a.m. and retrieves data for the previous day. That time was chosen to ensure all other nightly cron jobs that affect the databases are complete before the dashboard imports start. Each uses custom code for its data source and creates a series of MySQL INSERT queries to put the needed data fields in the MySQL data warehouse. For example, a script might pull the dates when an ILL request was placed and filled, but not the title of the requested item.

A carefully thought-out data model simplifies the creation of reports. The data structure should aim to support future expansion. In the data warehouse, information that was previously formatted and stored in very inconsistent ways is brought together uniformly. There is one table for each kind of data with consistent field names for dates, services, and so forth, and others that combine related data in useful ways.

The dashboard display consists of a number of widgets, one for each chart. Each chart is created with a mixture of PHP and JavaScript. Google Chart Tools interprets lines of JavaScript to draw an attractive, proportional chart. We do not want to hardcode the values in this JavaScript, of course, because the charts should be dynamic. Therefore we use PHP to query the data warehouse and a statement for each line of results to “write” a line of the data in JavaScript.

```
11 function drawLHdatechart() {
12     var LHdata = new google.visualization.DataTable();
13     LHdata.addColumn('string', 'Date'); // 0 = date
14     LHdata.addColumn('number', 'Chats'); // 1 = # of chats
15     LHdata.addColumn('number', 'Trend'); // 2= whether this metric is going up or down
16
17     <?php
18     echo "LHdata.addRows(" . (mysql_num_rows($records)) . ");";
19     while ($row = mysql_fetch_array($records)) {
20         echo "LHdata.setValue(" . $icount . ", 0, '" . $row['month'] . '/' . $row['day'] . "');" . " ";
21         echo "LHdata.setValue(" . $icount . ", 1, '" . $row['NumChats'] . "');" . " ";
22         $icount++;
23     }
```

Figure 1. PHP is used to read from the database and generate rows of data as server-side JavaScript.

Each PHP/JavaScript file created through this process is embedded in a master PHP page. This master page controls the order and layout of the individual widgets using the PHP *include* feature to add each chart file to the page plus a CSS stylesheet to determine the spacing of the charts. Finally, because all the queries take a relatively long time to run, the page is cached and refreshes itself the first time the page is opened each day. The dashboard can be refreshed manually if the database or code is modified and someone wants to see the results immediately.

Many of the dashboard’s charts include a linear regression trend line. This feature is not provided by Google Charts and must be inserted into the widget’s code manually. The formula can be found online.¹¹ The sums and sums of squares are totted up as the code loops through each line of data, and these totals are used to calculate the slope and intercept. In our twenty-six-week displays, we never want to include the twenty-sixth week of data because that is the present (partial) week. The linear regression line takes the form $y = mx + b$. We can use that formula along with the slope and intercept values to calculate y-values for week zero and the next-to-last week (week twenty-five). Those two points are plotted and the trend line is drawn between them. The color of the line depends on its slope (greater or less than zero). Depending on whether we want that chart’s metric to go up or down, the line is green for the desirable direction and red for the undesirable direction.

Details on Individual Systems

Gatecount

Most of NYUHSL's five locations have electronic gates to track the number of patrons who visit. Formerly these statistics were kept in a Microsoft Excel spreadsheet, but now there is a simple web form into which staff can enter the gate reading twice daily. The data goes directly into the data warehouse, and the a.m. and p.m. counts are automatically summed. There is some error-checking to prevent incorrect numbers being entered, which varies depending on whether that location's gate is the kind that provides a continuously increasing count or is reset each day.

The data are presented in a stacked bar chart, summed for the week. The user can hover over the stacked bars to see numbers for each location, but the height of the stacked bar and the trend line represent the total visits for all locations together.

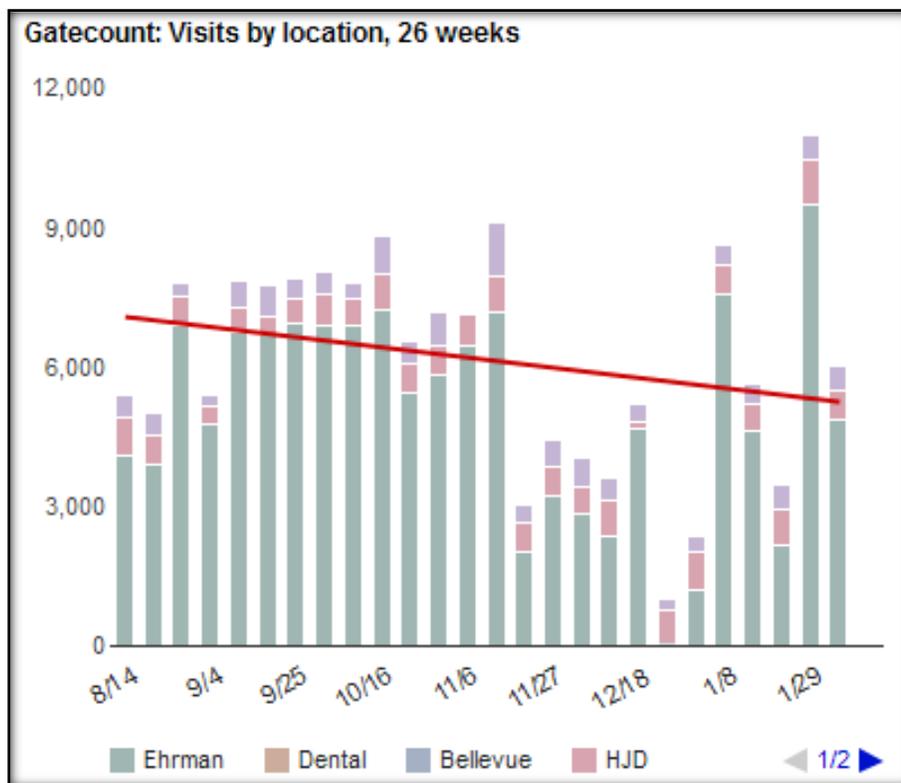


Figure 2. Stacked Bar Chart with Trendline Showing Visits per Week to Physical Library Branches over a Twenty-Six-Week Period

Ticketing

NYUHSL manages online user requests with a simple ticketing system that integrates with Drupal. There are four types of tickets, two of which involve helping users and two of which involve reporting problems. The "helpful" services are general reference questions and literature search requests. The "trouble" services are computer problems and e-resource problems. These two pairs

each have their own stacked bar chart because, ideally, the number of “helpful” tickets would go up while the number of “trouble” tickets would go down. Each chart has a trend line, color-coded for the direction that is desirable in that case.

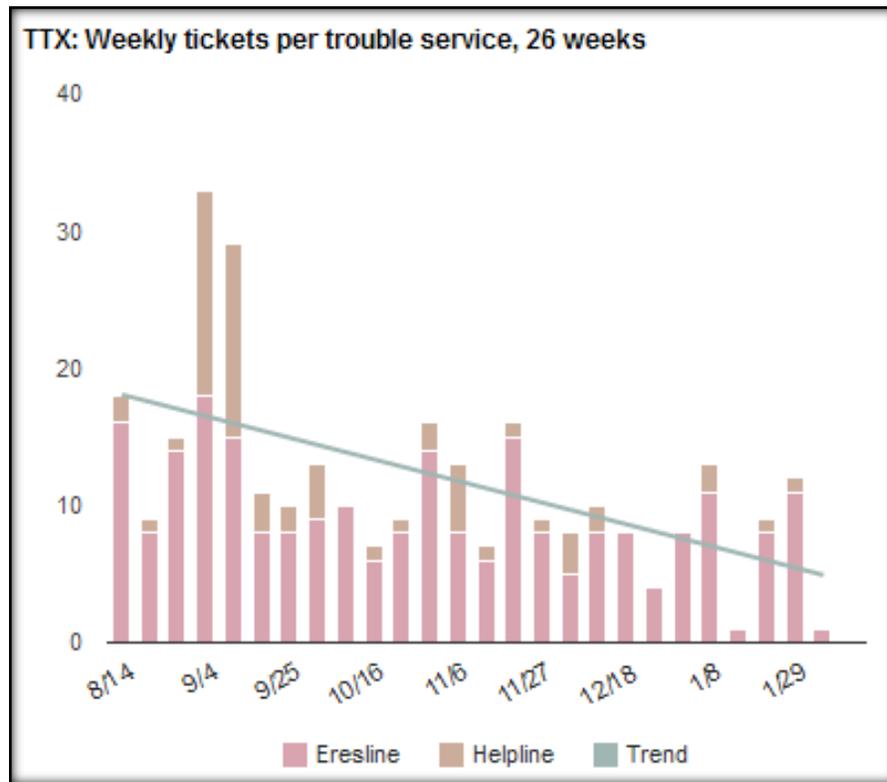


Figure 3. Stacked Bar Chart with Trendline Showing Trouble Tickets by Type

The script that imports this information into the data warehouse simply does so from another local MySQL database. It only fetches the date and the type of request, not the actual question or response. It also inserts a record into the user transactions table, which will be discussed in the section on user data.

Drupal

NYUHSL’s Drupal site allows librarians directly to contribute content like subject guides and blog posts.¹² The dashboard tracks the number of edits contributed by users (excluding the web services librarian and the web manager, who would otherwise swamp the results). This is done with a simple COUNT query on the *node_revisions* table in the Drupal database. Because no other processing is needed and caching ensures the query will be done at most once per day, this is the only widget that pulls data directly from the original database at the time the chart is drawn.

Koha

Koha is an open-source OPAC system. At NYUHSL, Koha’s database is in MySQL. Each night the importer script copies “issues” data from Koha’s *statistics* table. This supports the creation of a

stacked bar chart showing the number of item checkouts each week, with each bar divided according to the type of item borrowed (e.g., book or laptop). As with other charts, a color-coded trend line was added to show the change in the number of item checkouts.

Google Analytics

The dashboard relies on the Google Analytics PHP Interface (GAPI) to retrieve data using the Google Analytics Data Export API.¹³ Nothing is stored in the data warehouse and there is no importer script. The first widget gets and displays weekly total visits for all NYUHSL websites, the main NYUHSL website, and visits from mobile devices. A trend line is calculated from the “all sites” count.

The second widget retrieves a list of the top “outbound click” events for the past thirty days and returns them as URLs. A regular expression is used to remove any EZproxy prefix, and the remaining URL is matched against our electronic resources database to get the title. Thus, for example, the widget displays “Web of Knowledge” instead of “http://ezproxy.med.nyu.edu/login?url=http://apps.isiknowledge.com/.” A future improvement to this display would require a new table in the data warehouse and importer script to store historic outbound click results. This data would support comparison of the current list with past results to identify click destinations that are trending up or down.

Outbound links from Drupal sites, 30 days - Top 200	
Resource	Number of outbound clicks
1 PubMed	3199
2 Medline (PubMed)	3090
3 NYULMC Email	2162
4 NYU Home	2095
5 Medline (OVID)	1794
6 UpToDate	1756
7 The Link	1167
8 Access Medicine	632
9 New England journal of medicine (NEJM)	625
10 MD Consult	599
11 Micromedex	582
12 USMLEasy	531

Figure 4. Most Popular Links Clicked On to Leave the Library’s Website in a Thirty-Day Period

Libraryh3lp

LibraryH3lp is a Jabber-based IM product that allows librarians to jointly manage a queue of reference queries. It offers CSV-formatted log files that a Perl script can access using “curl,” a command-line tool that mimics a web browser’s login, cookies, and file requests. The CSV log is downloaded via curl, processed with Perl’s Text::CSV module, and the data are then inserted into the warehouse.

The first LibraryH3lp widget counts the number of chats handled by each librarian over the past ninety days. The second widget tracks the number of chats for the past twenty-six weeks and includes a trend line.

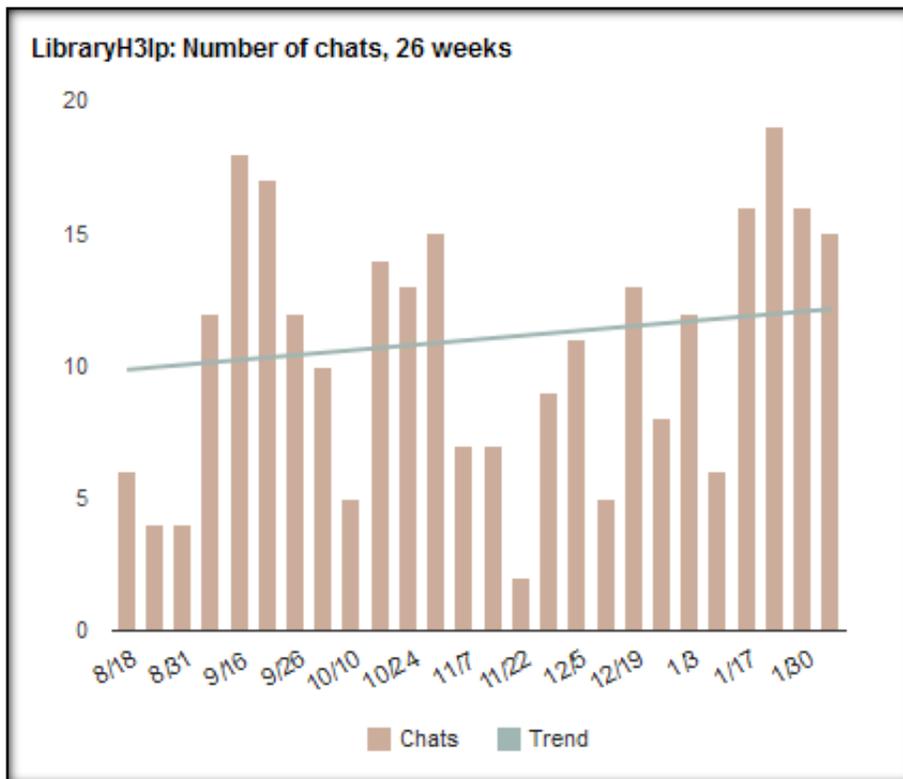


Figure 5. Bar Chart Showing Number of IM Chats per Week over a Twenty-Six-Week Period

Document Delivery Services

The Document Delivery Services (DDS) department fulfills ILL requests. The web application that manages these requests is homegrown, with a database in MySQL. Each night, a script copies the latest requests to the data warehouse. The dashboard uses this data to display a chart of how many requests are made each week and which publications are requested from other libraries most frequently. This data could be used to determine whether there are resources that should be considered for purchase.

The DDS data was also used to demonstrate how data might be used to track service performance. One chart shows the average time it takes to fulfill a document request. Further evaluation is required to determine the usefulness of such a chart for motivating improvement of the service or whether this is perceived as a negative use of the data. Some libraries may find this kind of information useful for streamlining services.

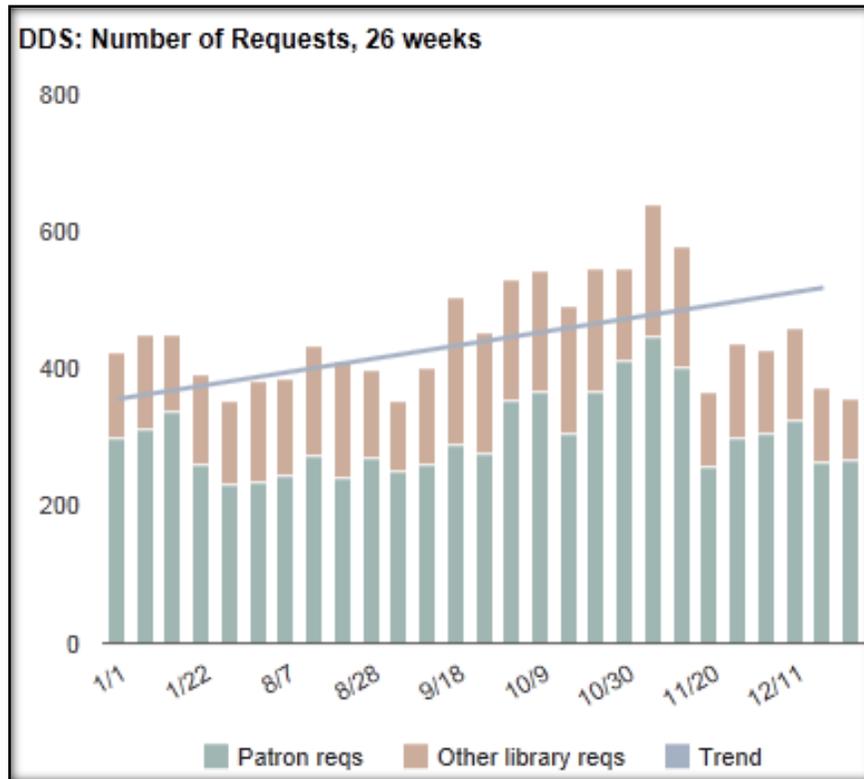


Figure 6. This stacked bar chart shows the number of document delivery requests handled per week. The chart separates patron requests from requests made by other libraries.

EZproxy Data

EZproxy is an OCLC tool for authenticating users who attempt to access the library’s electronic resources. It does not log e-resource use where the user is automatically authenticated using the institutional IP range, but the data are still valuable because it logs a significant amount of use that can support in-depth analysis. Because of the gaps in the data, much of the analysis looks at patterns and relationships in the data rather than absolute values.

Karen Coombs’ article discussing the analysis of EZproxy logs to understand e-resource at the department level provided the initial motivation to switch on the EZproxy log.¹⁴ When logging is enabled, a standard web log file is produced. Here is a sample line from the log:

```
123.45.6.7 amYu0GH5brmUska hansok01 [09/Sep/2011:18:25:23 -0500]
POST http://ovidsp.tx.ovid.com: 80/sp3.3.1a/ovidweb.cgi HTTP/1.1
20020472 http://ovidsp.tx.ovid.com.ezproxy.med.nyu.edu/sp-3.3.1a/ovidweb.cgi
```

Each line in the log contains a user IP address, a unique session ID, the user ID, the date and time of access, the URL requested by the user, the HTTP status code, the number of bytes in the requested file, and the referrer (the page the user clicked on to get to the site).

The EZproxy log data undergoes some significant processing before being inserted into the EZproxy report tables. The main goal of this is to enrich the data with relevant supplemental information while eliminating redundancy. To facilitate this process, the importer script first dumps the entire log into a table and then performs multiple updates on the dataset.

During the first step of processing, the IP addresses are compared to a list of departmental IP ranges maintained by Medical Center IT. If a match is found, the “location accessed” is stored against the log line. Next, the user ID is compared with the institutional people database, retrieving a user type (faculty, staff, or student) and a department, if available (e.g., radiology). One item of significant interest to senior management is the level of use within hospitals. As a medical library, we are interested in the library’s value to patient care. If there is significant use in the hospitals, this could furnish positive evidence about the library’s role in the clinical setting.

Next, the resource URL and the referring address are truncated down to domain names. The links in the log are very specific, showing detailed user activity. Because the library is operating in a medical environment, privacy is a concern and so specific addresses are truncated to a top-level domain (e.g. ovid.com) to suppress any tie to a specific article, e-book, or other specific resource.

Finally, a query is run against the remaining raw data to condense the log down to unique session ID/resource combinations, and this block of data is inserted into a new table. Each user visit to a unique resource in a single session is recorded; for example, if a user visits Lexis Nexis, Ovid Medline, Scopus, and Lexis Nexis again in a single session, three lines will be recorded in the user activity table. A single line in the final EZproxy activity table contains a unique combination of location accessed (e.g., Tisch Hospital), user department (e.g., radiology), user type (e.g., staff), earliest access date/time for that resource (e.g., 9/9/2011 18:25), resource name (e.g., scopus.com), session ID, and referring domain (e.g., hsl.med.nyu.edu).

There is significant repetition in the log. Depending on what filters are set up, every image within a webpage could be a line in the log. The method of condensing the data described previously results in a much smaller and more manageable dataset. For example, on a single day 115,070 rows of were collected in the EZproxy log, but only 2,198 were inserted into the final warehouse table after truncating the URLs and removing redundancy.

In a separate query on the raw data table, a distinct list containing user ID, date, and the word “e-resources” is built and stored in a “user transactions” table. This very basic data are stored so that simple user analysis can be performed (see “User Data” below).

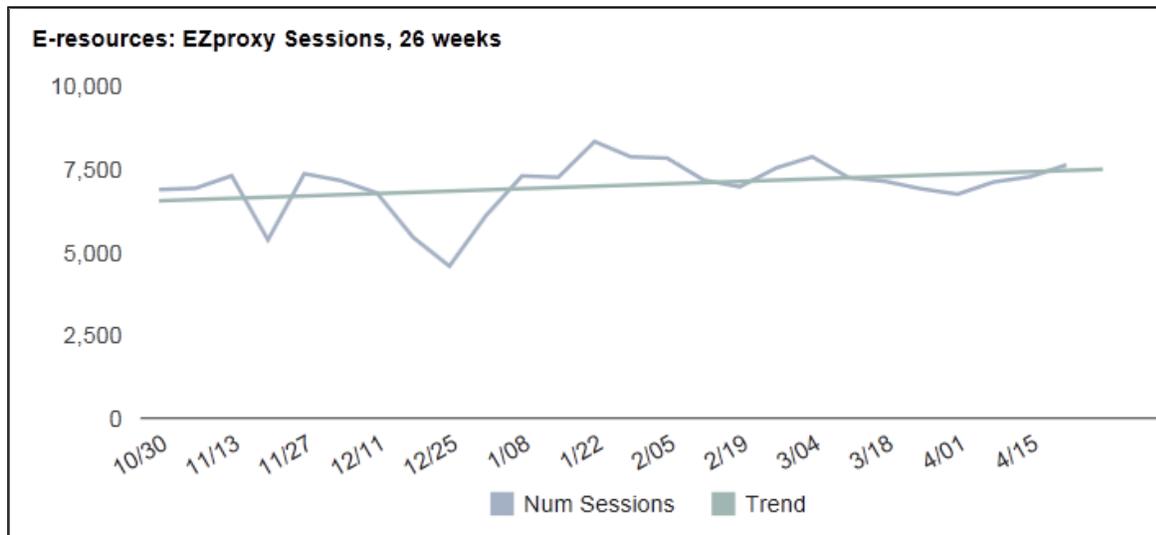


Figure 7. Line Chart Showing Total Number of EZproxy Sessions Captured per Week over a Twenty-Six-Week Period

Once the EZproxy data are transferred to the appropriate tables, the raw data (and thus the most concerning data from a privacy standpoint) is purged from the database.

Several dashboard charts were created using the streamlined EZproxy data, a simple count of weekly e-resource users, and a table showing resources whose use changed most significantly since the previous month. It was challenging to calculate the significance of the variations in use since resources that went from one session in a month to two sessions were showing the same proportional change as those that increased from one thousand to two thousand sessions. A basic calculation was created to highlight the more significant changes in use.

$$d = (p - q)$$

if $d < 0$ then

$$\text{significance} = d - 8 \times 10^d$$

$$q + 1$$

if $d > 0$ then

$$\text{significance} = d + 8 \times 10^d$$

$$q + 1$$

d = Difference between last month and this month

p = Number of visits last month (8 to 1 days ago)

q = Number of visits previous month (15 to 9 days ago)

This equation serves the basic purpose of identifying unusual changes in e-resource use. For example, one e-resource was shown trending up in use after a librarian taught a course in it.

E-resources: Most Significant Change in Use Over Past Month				
	Vendor name	This mo	Last mo	Change
↑	Science Direct	7527	5898	1629
↑	American College of Chest Physicians Publications	182	60	122
↑	Stat!Ref	321	150	171
↑	Wiley	3907	3154	753
↑	Journal of Neuroscience	610	398	212
↓	Mary Ann Liebert, Inc	15	135	-120
↑	Ebscohost	462	290	172
↑	NCBI	1503	1189	314
↑	The Physiological Society	433	280	153
↑	Science	783	568	215

Figure 8. Table of E-Resources Showing the Most Significant Change in Use over the Last Month Compared to the Previous Month

The EZproxy data has already proven to be a rich source of data. The work so far has only scratched the surface of what the data could show. Only two charts are currently displayed on the dashboard, but the value of this data is more likely to come from one-off customized reports based on specific queries, like tracking use of individual resources over time or looking at variations of use within specific buildings, departments, or user types. There is also a lot that could be done with the referrer addresses. For example, the library has been submitting tips to the newsletter that is delivered by email. The referrer log allows the number of clicks from this source to be measured so that librarians can monitor the success of this marketing technique.

User Data

Each library system includes some user information. Where user information is available in a system, a separate table is populated in the warehouse. As mentioned briefly above, a user ID, a date, and the type of service used (e-resources, DDS, literature search, etc.) is stored. Details of the transaction are not kept here. The user ID can be used to look up basic information about the user such as role (faculty, staff, student) and department.

We should emphasize for clarity that the detailed information about the activity is completely separated from any information about the user so that the data cannot be joined back together.

The most sensitive data, such as the raw EZproxy log data, is purged after the import script has copied the truncated and de-identified data.

Even though the data stored is very basic, information at the granularity of individual users is *never* displayed on the dashboard. The user information is aggregated by user type for further analysis and display. The institutional people database can be used to determine how many people are in each department. A table added to the dashboard shows the number of resource uses and the percentage of each department that used library resources in a six-month period. Some potential uses of this data include identifying possible training needs and measuring the success of library outreach to specific departments. For example, if one department uses the resources very little, this may indicate a training or marketing deficit. It may also be interesting to analyze how the academic success of a department aligns with library resource use. Do the highest intensity users of library resources have greater professional output or higher prestige as a research department, for example? It is unsurprising to find that medical students and librarians are most likely to use library resources. The graduate medical education group is third and includes medical residents (newly qualified doctors on a learning curve). As with the EZproxy data, there are numerous insights to be gained from this data that will help the library make strategic decisions about future services.

Service Use per User Group/Department, (Sept 2011-Feb 2012). [View Data](#)

User Group/Dept name	# in group	# unique users	# uses	Avg uses/user	% of group that used
Medical Students	603	496	14,418	29.07	82.26%
Library	58	45	773	17.18	77.59%
Graduate Medical Education	1,096	556	22,121	39.79	50.73%
Emerg Med	130	55	3,064	55.71	42.31%
Forensic Med	67	27	1,064	39.41	40.30%
Psychology	8	3	18	6.00	37.50%
Env Med	207	73	2,181	29.88	35.27%
EFAA	341	118	4,659	39.48	34.60%
Biochemistry	111	37	946	25.57	33.33%
Microbiology	165	49	1,996	40.73	29.70%

Figure 9. Table Showing the Proportion of Each User Group that has Used at Least One Library Service in a Six-Month Period

RESULTS

The dashboard has been available for almost a year. It requires a password and is only available to NYUHSL's senior management team and librarians who have asked for access.

Feedback on the dashboard has been positive, and librarians have begun to make suggestions to improve its usefulness. One librarian uses the data warehouse for his own reports and will soon provide his queries so that they can be added to the dashboard.

The dashboard has facilitated discoveries about the nature of our users and has identified potential training needs and areas of weakness in outreach. A static dashboard snapshot was recently created for presentation to the dean of the medical school to illustrate the extent and breadth of library use.

The initial dashboard aimed to demonstrate the kinds of library statistics that it is possible to extract and display, but there is much to be done to improve its operational usefulness. A dashboard working group has been established to build on the original proof-of-concept by improving the data model and adding relevant charts. Some charts will be incorporated into the public website as a snapshot of library activity. The dashboard was structured to be adaptable and expandable. The next iteration will support customization of the display for each user. New charts will be added as requested, and charts that are perceived to be less insightful will be removed. For example, one chart shows the number of reference chat requests answered by each librarian in addition to the number of chats handled per week. The usefulness of this chart was questioned when it was observed that the results were merely a reflection of which librarians had the most time at their own desks, allowing them to answer chats. This is an example of how it can be difficult to separate context from numbers. In this instance the individual statistics were only included because the data was available, not because any particular request from management, so these charts may be removed from the dashboard.

NYUHSL is also investigating the Ex Libris tool UStat, which supports analysis of COUNTER (Counting Online Usage of NeTworked Electronic Resources) reports from e-resources vendors. UStat covers some of the larger gaps in the EZproxy log, including journal-level rather than vendor-level analysis, and most importantly, the use statistics for non-EZproxied addresses. A future project will be to see whether there is an automated way to extract use metrics, either from UStat or directly from the vendors to be incorporated into the data warehouse. Preliminary discussion are being held with IT administrators about the possibilities of EZproxying library resource URLs as they pass through the firewall so that the EZproxy log becomes a more complete reflection of use.

An example of a strategic decision based on dashboard data involves NYUHSL's mobile website. Librarians had been considering the question of whether to invest substantial effort in identifying and presenting free apps and mobile websites to complement the library's small collection of licensed mobile content. The chart of website visits on the dashboard surprisingly shows that the number of visits that come from mobile devices is consistently fewer 3 percent, probably because of the relatively modest selection of mobile-optimized website resources. Rather than invest

significant effort in cataloging additional potentially lackluster free resources that would not be seen by a large number of users, the team decided to wait for more headlining subscription-based resources to become available and increase traffic to the mobile site.

It would be worthwhile to add charts to the dashboard that track metrics related to new strategic initiatives requiring librarians to translate strategic ideas into measurable quantities. For example, if the library aspired to make sure users received responses more quickly, charts tracking the response time for various services could be added and grouped together to track progress on this goal.

As data continues to accumulate, it will be possible to extend the timeframe of the charts, for example, making weekly charts into monthly ones. Over time, the data may become more static, requiring more complicated calculations to reveal interesting trends.

CONCLUSIONS

The medical center has a strong ethic of metric-driven decisions, and the dashboard brings the library in line with this initiative. The dashboard allows librarians and management to monitor key library operations from a single, convenient page, with an emphasis on long-term trends rather than day-to-day fluctuations in use. It was put together using freely available tools that should be within the reach of people with moderate programming experience. Assembling the dashboard required background knowledge of the systems in question, was made possible by NYUHSL's use of open-source and homegrown software, and increased the designers' understanding of the data and tools in question.

REFERENCES

¹ Association of Academic Health Sciences Libraries, "Annual Statistics," <http://www.aahsl.org/mc/page.do?sitePageId=84868> (accessed November 7, 2011); Association of Research Libraries, "ARL Statistics," <http://www.arl.org/stats/annualsurveys/arlstats> (accessed November 7, 2011).

² Brown University Library, "dashboard_beta :: dashboard information," <http://library.brown.edu/dashboard/info> (accessed January 5, 2012).

³ Edward R. Tufte, *The Visual Display of Quantitative Information* (Cheshire, CT: Graphics, 2001), 92.

⁴ Ibid., 56.

⁵ Ibid., 178.

⁶ Ibid., 153.

⁷ Scott Bateman et al., "Useful Junk? The Effects of Visual Embellishment on Comprehension and Memorability of Charts," *CHI '10 Proceedings of the 28th International Conference on Human Factors in Computing Systems* (New York, ACM, 2010), doi: 10.1145/1753326.1753716.

-
- ⁸ Stephen Few, *Information Dashboard Design: The Effective Visual Communication of Data* (Beijing: O'Reilly, 2006), 98.
- ⁹ Nils Rasmussen, Claire Y. Chen, and Manish Bansal, *Business Dashboards: A Visual Catalog for Design and Deployment* (Hoboken, NJ: Wiley, 2009), ch. 4.
- ¹⁰ Richard J. Roiger and Michael W. Geatz, *Data Mining: A Tutorial-Based Primer* (Boston: Addison Wesley, 2003), 186.
- ¹¹ One example: Stefan Waner and Steven R. Costenoble, "Fitting Functions to Data: Linear and Exponential Regression," February 2008, http://people.hofstra.edu/Stefan_Waner/RealWorld/calctopic1/regression.html (accessed January 5, 2012).
- ¹² Emily G. Morton-Owens, "Editorial and Technological Workflow Tools to Promote Website Quality," *Information Technology & Libraries* 30, no 3 (September 2011):92–98.
- ¹³ Google, "GAPI—Google Analytics API PHP Interface," <http://code.google.com/p/gapi-google-analytics-php-interface> (accessed January 5, 2012).
- ¹⁴ Karen A. Coombs, "Lessons Learned from Analyzing Library Database Usage Data," *Library HiTech* 23 (2005): 4, 598–609, doi: 10.1108/07378830510636373.