1. Targeted Web Analytics

- I'm going to talk about the tools I use, the process, and finally a few case studies of analytics work my library has done. Some of it has resulted in decisions or follow-up work, some of it we're still working on.

2. Tools

- Google Analytics: you probably already have this installed on many of your sites.

- Google Analytics Event tracking: While regular google analytics only lets you track page and session level statistics, event tracking lets you track user interaction with your site in much more detail. You apply an event method to nearly any kind of user interaction - clicking on links, hovering, typing - anything that you can identify with a javascript event can be tracked.

There are four main points of data that can be associated with each event that's tracked: a category, an action, a label, and a value. If you're familiar with bounce rates, events do count as site interactions, so if a user visits a page and then does something with an event attached to it, that visit will not have a bounce. There is one final piece of data you can attach to events that will tell it not to count it as an interaction for the bounce rate calculation.

So, a news site might want to track how its videos are being watched, and an online store might want to track the value of the products that are put into carts but later removed. But, you can really use events to track all sorts of different things - it's very flexible.

There are at least two ways you can add event tracking to your site. One, you can add inline javascript events to individual links. This is time consuming, and wouldn't work for vended search systems like Primo. Two, and this is how we do it for all of our event tracking, not just dynamic sites, you can use javascript to apply events across the board to links (or other interaction points) matching certain html classes or IDs.

- Then, jQuery has been my best friend. It makes it so much easier to traverse through the HTML, picking up elements as you need them, and doing stuff with them.

- Lastly, GreaseMonkey, a Firefox browser extension, lets you test javascript locally on your own computer, rather than having to actually call it from your website. This makes initial testing and proof-of-concept work much easier. Once I had something working, and had it put up on our test server, I used the Chrome developer tools (and especially the console) to test and debug.
- So, to put this altogether, this is what the code for tracking clicks on the online access, details, request via ill, request via docdel, related articles and times cited tabs in our discovery system looks like. The green commented out lines are the lines I added for testing and debugging, the _gaq.push line is what actually sends the event to google analytics, and there's a variety of jQuery selectors and methods in there.

3. Process

- 1. Formulate question. This is obviously super important, because the results you get are only going to be as good as the question you start with. Make it specific, and make sure it's something that can be answered by the data you can get. Questions like, which are the most important Primo Central Index collections to have activated can't be answered, but questions like, how much use does the related articles feature get can. Be careful, though, because the only answer you're getting is data, and it doesn't necessarily answer the question WHY. The answer to the related articles feature might be that it doesn't get much use, but to answer why it doesn't get much use will probably require further qualitative research.

- 2. Get data. In my experience, this has typically been the easiest part. The tools I mentioned earlier are what I've used to get this part of the process done as quickly as possible.

- 3. Make decisions. This has been the hardest part. When it's a one or two person project, it's been relatively easy to formulate a question, gather some data, and move forward after a reasonable amount of time. When we first started using event tracking, we applied it to nearly all the links in our discovery system, and after about six months, had enough data that we were comfortable removing the Reviews and Tags tab on the records. We corroborated the event tracking data with data from Primo, which showed that, in addition to very few people clicking on the tab, only one or two people had actually logged in recently to leave a review or add a tag. Some of the questions, though, are ones we know we won't be able to do anything about right away, or are ones that we'll need more follow-up data to make decisions on.

Ok, now onto a few projects!

4. Basic questions

- These are ones that can be answered by the basic google analytics tracking code, without any extra work. The data can help answer high level questions, like what devices are patrons are using - ie are they using their mobile devices for research, and when are they using our services - ie when is a good time for us to do maintenance that requires a downtime or other disruption?
- So, when are the least busy hours? Ideally, maintenance work would require no
downtime, but some of it does. I'm working on an authentication project for a vended
system for which we do not have a test server, so all testing has to be done (carefully!)
on our production site, so knowing when we can inconvenience the fewest patrons is
ideal. I thought, based on my own habits, I knew when would be a good time for
testing, but I'm glad I looked at our data, cause I was wrong.

- Data gathering: like I said, we just use the basic google analytics data, and look at
either pageviews or sessions for specific months, days and times when usage is very
low.

- Data results: I thought usage would be low Monday morning and Friday evening, but
it's not! Usage is definitely lowest between 2 and 4 in the morning, with weekdays
before 8 and Saturday nights and Sunday mornings being much lower than daytime
hours

- decision: we target weekdays before 8, saturday evenings and sunday mornings to
do work, and some of it is done from home to accommodate these hours.

5. Navigation questions

- these are questions we've used relatively simple event tracking for. A couple lines of
code grab all the links in different sections of our website, and then apply labels to the
links just based on the link name

- are the quicklinks being used enough to justify being kept there? This box is
designed to have the links we think most people will need on a regular basis, or links
we want to make it easy for people to find

- and here are the links

- to gather the data, we have apply tracking to everything in this box, and use the link
text as the label

- results, over the spring semester: our reserves are used a bunch. ILL, and My library
accounts are used a decent amount. The interdisciplinary databases we've chosen to
highlight are all in the top 10, so that's a good indication that these were good choices,
and subjects A-Z and refworks make sense too. The other dozen or so links not on the
top ten all have fewer than 1000 clicks over the semester (out of nearly 500,000 visits
to our homepage), and most have fewer than 300. The one with the lowest use that
was present for the entire semester only had around 60.
- we haven't made any decisions about links to change, but now that we have the data, when it comes time to look at this, it'll be much easier to justify keeping or removing certain links

6. Features use questions

- the questions we've looked at so far are mostly about tools like our discovery system and our link resolver, and this is actually where we started with event tracking, since we wanted more data on how the facets in our discovery tool were being used (they didn't get as much use as we thought they would...)

- a more recent project has been to look at the text input boxes on our link resolver, where patrons could specify a year, volume, issue and page number for journals. I didn't think it was likely to get much use, but I wanted to make sure of that.

- this is what the link resolver menu looked like

- so, I set up an event that would fire when any of the boxes were clicked on

- the boxes got used more than I thought, but it was still only about 4% of sessions, and at least of quarter of that was from staff usage. I could see which was staff by looking at the page URLs, and many of them included the SFX source ID RapidILL, which could only be sent by staff using the RapidILL system

- solution: I hid the boxes under a little down arrow, notified staff about the change so that they would know where to look for them, added a neat little "what's new/changed" highlight tool called intro.js for the first month of the new semester, and added hover text to the arrow

- this is what it looks like now

7. Content use questions

- these questions are little bit more difficult to answer, because our discovery system doesn't include a consistent source identifier in the main part of the displayed record

- but, we thought we'd give it a try anyway, and set out to find out more about which collections we've activated in the Primo Central Index are being used

- so, we set up the event tracking to track each time a link in a record was clicked on, with the label being the link text and the action being the collection. Record interactions include clicks on the online access tab, the details tab, the export links, the
button to add the record to the e-shelf, and pretty much all the other links associated with a record

- results: nearly 60% of record interactions are with local catalog records, either our own or from our partner libraries. Included in that are HathiTrust records that we loaded locally. The next biggest chunk is Gale's Cengage collection, which we activated in order to add more newspaper content to the discovery record pool. Web of Science and JSTOR are the next two biggest sources, but they only come in at 4 and 2% of record interactions. Surprisingly, even we look at just online access links, 40% are for local records, 16% are for Gale records, 7% for Web of Science and 4% for JSTOR.

- we haven't made any decisions based on this data yet, and before we look at deactivating certain collections, we'll likely look at the data with a larger group of people in the library and follow up with some qualitative research into why our discovery tool is being used like this. It is really, really interesting to me how much our local records are being used though, especially given that about 2/3 of the usage of Primo is in the articles+book scope.

8. Support questions

- for these, we're just using basic analytics for now.

- we set up the EZproxy needhost.htm page to redirect to a page on our website, and so the ga code that's applied to our webpage templates applies to it as well. When the EZproxy page redirects, it sends along the page requested, the error type and the referrer, so that we have more information about what the error was and where the patron encountered it.

- there have been quite a few hits on this page, and we've been able to correct some of the major problems as they've come up

- but, going forward, we'd rather have some sort of auto-notification, so one of my colleagues is building a script that will auto-report these errors, now that we know it's such a big problem

9. Other questions we've looked at, but which I didn't have time to cover today include the following

10. Thanks!
Tools mentioned:

- Firebug (for Firefox): http://getfirebug.com/
- Intro.js (what’s changed highlight thingy): http://usablica.github.io/intro.js/, and example on our test SFX server: http://link.library.nd.edu/in4qv
- Reveal.js (presentation framework): http://lab.hakim.se/reveal-js/#
- Get in touch with me for our analytics js or for the EZproxy redirect stuff!
lajamie@nd.edu