Understanding User Behavior Through Log Data and Analysis

Susan Dumais, Robin Jeffries, Daniel M. Russell, Diane Tang, and Jaime Teevan

Overview of Log Data and Analysis in HCI

Behavioral logs are traces of human behavior seen through the lenses of sensors that capture and record user activity. They include behavior ranging from low-level keystrokes to rich audio and video recordings. Traces of behavior have been gathered in psychology studies since the 1930s (Skinner, 1938), and with the advent of computer-based applications it became common practice to capture a variety of interaction behaviors and save them to log files for later analysis. In recent years, the rise of centralized, web-based computing has made it possible to capture human interactions with web services on a scale previously unimaginable. Large-scale log data has enabled HCI researchers to observe how information diffuses through social networks in near real-time during crisis situations (Starbird & Palen, 2010), characterize how people revisit web pages over time (Adar, Teevan, & Dumais, 2008), and compare how different interfaces for supporting email organization influence initial uptake and sustained use (Dumais, Cutrell, Cadiz, Jancke, Sarin, & Robbins, 2003; Rodden & Leggett, 2010).

In this chapter we provide an overview of behavioral log use in HCI. We highlight what can be learned from logs that capture people's interactions with existing computer systems and from experiments that compare new, alternative systems. We describe how to design and analyze web experiments, and how to collect, clean and use log data responsibly. The goal of this chapter is to enable the reader to design log studies and to understand results from log studies that they read about.

S. Dumais (✉) • J. Teevan
Microsoft Research One Microsoft Way, Redmond, WA 98005, USA
e-mail: sdumais@microsoft.com; teevan@microsoft.com

R. Jeffries • D.M. Russell • D. Tang
Google, Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA
e-mail: robin@jeffries.org; drussell@google.com; diane@google.com

Table 1 Different types of user data in HCI research

	Observational	Experimental
Lab Studies		
Controlled interpretation of behavior with detailed instrumentation	In-lab behavior observations	In-lab controlled tasks, comparison of systems
Field Studies		
In the wild, ability to probe for detail	Ethnography, case studies, panels (e.g., Nielsen)	Clinical trials and field tests
Log Studies		
In the wild, little explicit feedback but lots of implicit signals	Logs from a single system	A/B testing of alternative systems or algorithms

What Are Behavioral Logs?

In HCI research behavioral logs arise from the activities recorded when people interact with computer systems and services. Behaviors of interest can include: low-level actions such as the keystrokes used when interacting with a productivity application; the content viewed in web browsers or e-readers; the search queries and result clicks captured by web search engines; browsing patterns and purchases on e-commerce sites; content generated and shared via social media; the history of edits to wikis or other documents; detailed traces of eye gaze position when playing a computer game; physiological responses when driving; etc.

An important characteristic of log data is that it captures actual user behavior and not recalled behaviors or subjective impressions of interactions. While logs can be captured in laboratory settings, they are increasingly captured at a much larger scale in situ as people interact with applications, systems, and services. Behavioral observations can be collected on a client machine or on remote servers. Client-side logging can be included in operating systems, applications such as browsers or e-readers, or special purpose logging software or hardware. Server-side logging is commonly used by service providers such as web search engines, e-commerce sites, or online courses. Some behavioral logs are publically available (e.g., Wikipedia content and edit history, Twitter posts, Facebook public feeds, Flickr photos, and Pinterest collections), but many are private, available only to individuals or service providers.

What Can We Learn from Behavioral Logs?

To understand what HCI researchers and practitioners can learn from behavioral logs, it is useful to compare them with other types of data. This book summarizes many different HCI methods for understanding behavior and improving design. In Table 1 we highlight a simplified view of a few approaches, described in more detail in other chapters, which are useful to contrast with log studies. The two dimensions represented in the table are: (1) whether the studies are observational or experimental, and (2) the naturalness, depth and scale of the resulting data.

Lab Studies represent the most controlled approach. In lab studies participants are brought into the laboratory and asked to perform certain tasks of interest. Demographic and other data can easily be collected about participants. The lab setting affords control of variables that are not of interest and enables detailed instrumentation of novel systems that could not be easily deployed more broadly. Researchers can learn a good deal about participants and their motivations in this way, but the observed behavior happens in a controlled and artificial setting and may not be representative of behavior that would be observed “in the wild.” For example, a person may invest more time to complete a task in the lab than they might otherwise to please the investigator (Dell, Vaidyanathan, Medhi, Cutrell, & Thies, 2012). In addition, laboratory studies are often expensive in terms of the time required to collect the data which limits the number of different people and systems that can be studied. Lab studies can be either observational, examining interactions with a specific system of interest, or experimental, comparing two or more variables or systems.

Field Studies collect data from participants in their natural environments conducting their own activities, and, commonly, periodically ask them for additional information. Data collected in this manner tends to be less artificial than in lab studies but also less controlled. As with lab studies, demographic and other data can be collected about participants, but the researcher may still interfere with people’s interactions by asking them about what they are doing. Field studies can be observational (e.g., where TV watching behavior is recorded for Nielsen panelists) or experimental (e.g., in clinical trials where new medical treatments are compared with a control).

Log Studies collect the most natural observations of people as they use systems in whatever ways they typically do, uninfluenced by experimenters or observers. As the amount of log data that can be collected increases, log studies include many different kinds of people, from all over the world, doing many different kinds of tasks. However, because of the way log data is gathered, much less is known about the people being observed, their intentions or goals, or the contexts in which the observed behaviors occur. Observational log studies allow researchers to form an abstract picture of behavior with an existing system, whereas experimental log studies enable comparisons of two or more systems.

Log studies are a valuable complement to other kinds of studies for several reasons. They represent traces of naturalistic human behavior uninfluenced by observers. Because log data provide a portrait of uncensored behavior, they give a more complete, accurate picture of all behaviors, including ones people are unlikely to talk about or remember accurately. Early analyses of web search logs, for example, found that searches for porn were common and were associated with different interaction patterns than other types of search (Silverstein, Henzinger, Marais, & Moricz, 1998). Similarly (Teevan, Adar, Jones, & Potts, 2007) observed that many queries were repeated, a behavior that would probably not be seen in the lab.

Logs also have the benefit of being easy to capture at scale. While laboratory and field studies typically include tens or hundreds of people, log studies can easily include data from tens or hundreds of millions of people. This large sample size means that even small differences that exist between populations can be observed. In particular, large-scale logs give a good picture of unusual but important behavior that is hard to capture in smaller studies. For example, if fewer than 1 in 100 people click on banner advertisements, a lot of effort would be required to collect a reliable number of such clicks in a laboratory setting. In contrast, the behavior can be significantly and reliably understood in web browser logs where there are millions of clicks on advertisements. As another example, student behavior logged during massively online courses can provide detailed insight into individual learning strategies and how they relate to educational success.

In spite of the benefits, logs have disadvantages, including non-random sampling (people must choose to use the system), uncontrolled tasks, and the absence of annotations to indicate motivations, success, or satisfaction. Logs provide a good deal of information about *what* people are doing but much less about *why* they are doing so and whether they are satisfied. This must be taken into account in analyses and complemented by other techniques to provide a more complete understanding of behavior.

In the remainder of this chapter, we describe in more detail large-scale log studies, providing examples of observational (section "Observational Log Studies: Understanding Behavior Through Log Analysis") and experimental (section "Experimental Log Studies: Comparing Alternative Systems Through Log Analysis") approaches, and discussing how to collect, clean (section "Collecting, Cleaning, and Using Log Data"), and share log (section "Using Log Data Responsibly") data. While publically available log data such as Wikipedia edit history or Twitter posts can support interesting observational analyses, there are fewer reports of large-scale experimental design and analysis, so we cover this aspect in particular detail. Most of the examples that we present in this chapter come from our own experiences in collecting and analyzing large-scale behavioral logs from web search engines, but the methods are more broadly applicable.

Observational Log Studies: Understanding Behavior Through Log Analysis

Most analyses of log data collected through observational studies provide a descriptive overview of human behavior. Simply observing behavior at scale provides insights about how people interact with existing systems and services, often revealing surprises. For example, an early analysis of more than a million web searches found that queries were short, averaging only 2.35 terms, and that over 80 % of all queries did not include advanced operators (Silverstein et al., 1998). Although these findings are consistent with what we expect for web search today, they were initially quite surprising because they differed from previous observations of search conducted in other contexts, such as in libraries. Librarians tended to issue queries that were much longer and included many more advanced operators. Another important observation from early web search engine logs was that query frequency was not uniform. Some queries were asked frequently (the 25 most common queries in the Silverstein et al. study accounted for 1.5 % of all queries), while others occurred very infrequently (64 % of the queries occurred only once). For additional results from observational studies of web search logs, we recommend (Jansen, 2006; Silverstein et al., 1998; Spink, Ozmutlu, Ozmutlu, & Jansen, 2002).

When analyzing log data, researchers extract a variety of metrics. *Metrics* are measurable quantities that matter to the users or system stakeholders. Metrics can emerge directly from the data, such as, in the case of search, query length or frequency. However, other metrics can be computed that allow researchers to infer information that is not directly represented in the raw data. For example, researchers have developed behavioral proxies for search success based on clicking and dwelling behavior (Fox, Karnawat, Mydland, Dumais, & White, 2005). While inferences based on such analyses can be noisy and imperfect, the large scale of log data can overcome distortions due to randomness. Noise due to factors that are not systematically related to the phenomenon of interest tends to even out with a large number of observations.

In order to place descriptive metrics in context, it is necessary to compare them to similar metrics for different contexts. For this reason, the process of learning about user behavior from log data typically involves partitioning the data into meaningful subsets, called *partitions*, and comparing across the different partitions. There are many different ways behavioral log data can be partitioned, including by language (Ghorab, Leveling, Zhou, Jones, & Wade, 2009), geography (Efthimiadis, 2008), device (Baeza-Yates, Dupret, & Velasco, 2007), time (Beitzel, Jensen, Chowdhury, Grossman, & Frieder, 2004), and user (Kotov, Bennett, White, Dumais, & Teevan, 2011; Teevan et al., 2007). Log data can also be partitioned by system variant, where behavior is observed under two different system conditions, such as when comparing two different user interfaces. How to successfully partition, collect, and analyze experimental data from system variants is discussed in section "Experimental Log Studies: Comparing Alternative Systems Through Log Analysis."

Two common ways to partition log data are by time and by user. Partitioning *by time* is interesting because log data often contains significant temporal features, such as periodicities (including consistent daily, weekly, and yearly patterns) and spikes in behavior during important events. It is often possible to get an up-to-the-minute picture of how people are behaving with a system from log data by comparing past and current behavior. For example, researchers can accurately predict the strength of seasonal flus based on search engine log data with a lag of only 1 day (Ginsberg, Mohebbi, Patel, Brammer, Smolinski, & Brilliant, 2009). In contrast, the Center for Disease Control and Prevention (CDC) typically reports such information with a 1–2 week lag. Care must be taken when partitioning log data by time because logs contain observations from many different time zones. This is discussed in greater detail in section "Collecting, Cleaning, and Using Log Data."

It is also interesting to partition log data *by user characteristics*. For example, researchers have looked at how advanced search users compare with novices (White & Morris, 2007), and how domain experts use different vocabulary, resources, and strategies than people who do not know about a domain (White, Dumais, & Teevan, 2009). In addition to comparing across users, it is also possible to look for patterns of behavior within an individual. In this way, researchers have discovered that people often return to common topics when they search (Kotov et al., 2011) and even repeat the same query over and over again (Teevan et al., 2007; Tyler & Teevan, 2010). One challenge with partitioning data by user is that it can be hard to accurately identify a user from log traces, and we discuss common ways this is done in section "Collecting, Cleaning, and Using Log Data."

Ideally the log partitions will be similar in all aspects other than what is being studied, to control for other factors potentially influencing the observed differences. It is useful to run a sanity check across partitions to confirm that metrics that should be consistent actually are. For example, White et al. (2009) examined differences in search strategies and outcomes for domain experts as compared to novices. Within the domain of interest, experts used different vocabulary, visited different sites, used different strategies, and achieved higher success. But, outside their domain of expertise, there were no differences in search performance, indicating that differences were isolated to the variable of interest and not to more general differences between the groups.

Although many useful things can be learned from observational log analysis, there are drawbacks to the approach. For one, rarely is there much information about the people who generate the data; not their age, gender, or even whether events observed at different times or on different machines are from the same person. Even less is known about user motivations. Logs cannot tell us people's intent, success, experience, attention, or beliefs about what is (or is not) happening. For example, when a person leaves a search page without clicking anything, it could be because they could not find what they were looking for, or it could be because the content presented on the result page was sufficient to satisfy their information need. As another example, click entropy, or the variation in what people click following a query, is often used as a proxy for how ambiguous a query is. However, while variation in clicks can arise due to variation in intent, it can also be caused by changing results or an ambiguous need that requires synthesis across multiple pages to meet (Teevan, Dumais, & Liebling, 2008). We also do not know when people are confused. For example, a person may inadvertently switch from general web search to image search, yet still believe they are searching the entire web when they issue a query.

Additionally, when a system uses log data to drive its own performance, people may have ulterior (often adversarial) motives to create artificial traces. For example, if a search engine boosts results that are consistently clicked on for a query towards the top of the ranked list, then spammers may game the system by repeatedly clicking a result solely for the purpose of boosting its rank (Fetterly, Manasse, & Najork, 2004).

One way to mitigate such limitations is by controlling for as many external factors as possible. In the case of query ambiguity, for example, appropriate metrics will consider not just click entropy but also the entropy of the results returned and the average number of clicks per user. This kind of deep analysis is best done by examining a sample of the user traces directly (often by a hand examination of a representative sample) and not just computing metrics over them, so the researchers can be sure that important insights have not been lost in the averaging of millions of data points. We describe more about how to look at and clean data is discussed in section "Collecting, Cleaning, and Using Log Data."

Another way to better understand what is going on within log data is to supplement the logs by capturing context that may not be directly represented. Capturing as much contextual data as possible up front, including the version of the system being interacted with and what users of the system actually see, can be critical to understanding log data or comparing data collected at different points in time. Additionally, log data can be enhanced with insight about what the user is doing via field trials or critical incident studies. For example, Broder (2002) conducted a critical incident study in which users were occasionally interrupted with a pop up window asking the motivation behind the query they had just issued. This led to the classification of queries as navigational (i.e., targeted at getting to a particular web page), informational (i.e., intended to find information that is present in one or more pages), or transactional (i.e., intended to perform a web-mediated activity).

Even when the log data is very rich, researchers should not rely solely on logs to understand user behavior. Converging methods can help confirm and provide insight into what is learned from log data. Methods that complement log analysis include usability studies, eye tracking studies (see Eye Tracking in HCI: A Brief Introduction, this volume), field studies, diary studies, retrospective analysis (see Looking Back: Retrospective Study Methods for HCI, this volume), and surveys (see Survey Research in CHI, this volume). For example, Teevan and Hehmeyer (2013) analyzed the logs of a popular enterprise communication system that infers and projects availability state of users. They found that people were significantly more likely to answer the phone when their status indicated that they were busy, but were unable to tell from the logs why this was the case. By conducting a complementary survey study they discovered that busy people perceived incoming phone calls as particularly important because they knew the caller chose to call as opposed to email. The survey helped to explain the rich, real-world picture provided by the log files, and gave a view into communication behavior that would not have been possible otherwise.

In addition to inspiring complementary studies and suggesting interesting areas for further research, the results of observational log studies make it possible to design computer systems that support people's actual behavior rather than their presumed behavior. For example, search engines were designed to cache search result pages because query log analysis revealed that only a handful of unique queries represent a significant portion of search engine traffic (Silverstein et al., 1998).

Log data can also be used to test hypotheses that researchers develop about user behavior. Lau and Horvitz (1999) used log data to learn a probabilistic model of how users refine their queries over the course of a session, and then evaluated how well they could predict the next action in a sequence using log data. Likewise, Kotov et al. (2011) used log data to learn a predictive model of whether web searchers were likely to return to their current search task sometime in the future. To avoid over-fitting, models should be learned from a subset of log data and evaluated on data that is similar in characteristic to (but not the same as) the data used to construct the model.

All of the studies described above rely on logs recorded from existing systems. As such, the findings are limited to understanding existing interactions. It is impossible to learn from observational log analysis how people might interact with a different system. For example, people may want to use facets to navigate web search results, filtering results by how recent the content is or the quality of the web site. But search logs cannot reveal this because people do not currently have the option to use facets. Similarly, people might want to search for old posts on Twitter, but this behavior is rarely observed (Teevan, Ramage, & Morris, 2011). This could be because old posts are not interesting, or it could be an artifact of the fact that the Twitter search interfaces currently only returns the most recent tweets. Observational log analysis can only reveal what people do with the tools they have. A richer way to test hypotheses that allows researchers to explore new interaction paradigms is to vary the system users interact with in an experimental framework, and compare how the logged behavior differs across system variants. This is discussed in detail in the next section.

Experimental Log Studies: Comparing Alternative Systems Through Log Analysis

To understand how people react to different user experiences, a typical approach is to run an in situ experiment designed to compare behavior across different system variants. Web experiments are commonly used to understand and improve a variety of services (Kohavi, Longbotham, Sommerfield, & Henne, 2009; Tang, Agarwal, O'Brien, & Meyer, 2010) and may be colloquially called *A/B tests* (meaning comparing system A with system B) or *bucket tests* (because users are "bucketed" into different user experiences, from the term "hash bucket").

Log-based experiments are often the only way to evaluate small changes to a system (e.g., a change in font; a change in the text label on a button or in a status message) and can also be useful to evaluate larger changes (e.g., complete redesign of a site's page layout or a change in the workflow to complete a task). The nature of an experiment and its analysis differ as a function of the change being studied. Large changes tend to produce noisier data and thus require more data points to get a statistically reliable signal. For small changes, the analyst will usually be able to identify a small number of metrics that should improve for the design change to be considered a success. In the case of a major change, the analyst may have metrics specific to the goals of the particular experiment (e.g., how many people completed the task flow; how much time people spent on the newly redesigned site), but may also need to look at a broad suite of metrics, particularly since a priori hypotheses about how behavior will change are harder to arrive at. Even if tools that automatically compute typically useful metrics are available [e.g., Google Analytics (Google, 2012)], the researcher may need to conduct a custom analysis for a large change. If the goal of a redesign is to move clicks from one category of UI element to a different one (e.g., changing from clicks on elements that move one forward to clicks on elements that invite deeper exploration), then a custom analysis that looks at exactly those clicks will be needed.

Sometimes it feels that doing a formal experiment to compare two approaches is unnecessary because the answer is obvious (typically that the new design is "better"). Experience says otherwise: those of us accustomed to doing log analysis have many times seen the outcome of an experiment differ from the supposedly obvious answer. For example, changing a text button's font to make it more visible and more likely to be clicked may backfire if instead it makes the button seem not to be a clickable object. Moreover, even when the intended outcome does occur, there can be unanticipated side effects. For example, people may use additional information available in one of the conditions, such as automatic spell correction in search, but are slowed down because they feel compelled to click through to the results for their misspelled query. Side effects are typically not easily predicted, but may be important enough to overpower the positive aspects of a change. Without an experiment and examination of a broad set of metrics, a team may not discover tradeoffs inherent in a proposed change.

The basics of log data collection are covered in section "Data Collection." We next cover how to design an experiment that will give valid results (meaning if it were repeated by a different set of experimenters, similar results would be expected) and the basics and common pitfalls of analyzing experiments.

Definitions

In order to talk about experiments we need to have a common language. Here are a few important definitions that are important in understanding server-based web experiments:

Request: A user action or user request for information. For web applications, a new web page (or change in the current page) is requested via a set of parameters (this might be a query, a submit button at the end of a form, etc.). The result of the request is typically the unit of analysis that the researcher is interested in (e.g., a query, an email message displayed, a program being debugged).

Cookies: A way of identifying a specific session in a browser. We typically equate a cookie with a user, but this is an important source of bias in log studies as described in more detail in section "Collecting, Cleaning, and Using Log Data."

Diversion: How traffic is selected to be in a particular experimental condition. It might be random at the request level; it might be by user id, by cookie, or as a hybrid approach, by cookie-day (all requests from a given cookie are either in or out of the experiment each day). Typically experiments for user experience changes are user-id or cookie based.

Triggering: Even if a request or cookie is in the experiment, the experimental change may not occur for all requests. For example, if the change is to show the current weather at the user's location on weather-related queries, this will only trigger on the small fraction of queries that are weather related; all conditions in the experiment will provide identical experiences for other queries. On the other hand, for an experiment that changes the size of the logo shown in the upper left of all pages, all requests diverted into non-control conditions will trigger the change.

Experiment Design

A web-based experimental condition diverts some subset of the incoming requests to an alternate processing path and potentially changes what is shown to the user. A control condition diverts some subset of incoming requests, but does not change what is shown to the user. For example, in an experiment that explores the effect of the size of the logo in the upper left corner of the page, the control would leave the logo unchanged, one experimental condition would increase the size of the logo, and another condition would decrease the size of the logo.

The design of an experiment begins with a set of questions to be answered. In HCI experiments, the question will be a variant of, "Will this user experience be better for users (in some way that needs to be defined and quantified) than another experience (which might be the existing experience)?" "Better" may be defined, for example, as getting results faster, or it may be defined as being more fun, resulting in people spending more time on the site. A log experiment only makes sense with specific, concrete research questions. For a small change, this is likely to be easy; for example, if a UI manipulation changes the font of the text on the page, the associated hypotheses will most likely have to do with engagement with the page. But with a larger change, such as changing the location and wording of links on the homepage of a site, it is harder to attribute changes in behavior to specific changes made. For example, one result of the page layout change might be that more people click on the "Contact Us" link. First, are more clicks on "Contact Us" good? Second, if the primary metrics for this experiment are site-wide—e.g., how much time do people spend on the site? Do they end up engaging with the site in a way that represents the core purpose of the site?—how does the improved access to the Contact Us page contribute to that metric, and is it a positive or negative contribution?

Questions such as these still needs to be turned into testable hypotheses about the qualitative and quantitative behavioral changes expected. For example, the experimenter may hypothesize that the condition with a large red button reading "sign up" will result in significantly more signups. There may be several testable hypotheses that relate to different metrics. Alternatively, if the goal is to explore a space of design choices (e.g., whether a button 1, 2, or 3 in. tall results in the most signups), there may not be specific predictions beyond the expectation that behavior will differ across the conditions.

Testable hypotheses will lead to a set of conditions to test. A condition may or may not be “user visible” (one might experiment with different delays in delivering the page, which changes the user experience but not in a visible way), or even “user noticed” (a change in the order of search results on a page will not be noticed unless a user can see results from more than one condition in the experiment).

A special condition is one designated as the *control*. The control is the “baseline” user experience to which the “new, improved” experience is being compared. Typical controls in web experiments include the existing treatment of a feature, or to not show a feature shown in another condition. If there are two novel treatments, neither can be considered the control—there needs to be a third condition to serve as the control. In some experiments, there may be multiple controls.

In an ideal world one would test a set of parameters (the simplest case being one two-level parameter: feature X is “on” or “off”) and all the factorial combinations of them. In practice, only a subset of the possible combinations is tested. This may be:

A practical constraint: if the parameters are the background color of the page and the text color, setting both to the same value produces an unreadable page,

A logical constraint: neither a large nor a small image makes sense for the condition where no image is shown, or

A resource constraint: there may not be enough traffic to make statistically meaningful comparisons among a large number of different conditions unless the experiment is run for an unrealistically long time.

Because of this log experiments are seldom factorial experiments and are typically analyzed as a set of pairwise comparisons with the appropriate control.

We have covered the aspects of experimental design that are determined by the study’s research questions; next we discuss aspects that influence the analysis phase or have pragmatic significance.

Making Conditions Comparable

A good deal of experiment analysis depends on having a control condition that is directly comparable to the experimental conditions. However, this is often more complex than imagined. There are a variety of ways that comparisons between conditions go wrong.

It is important to run all conditions in the experiment concurrently, if possible. It may be tempting to run the control and the experimental conditions at different times, because the traffic to the site being experimented on is low, and it is less work to change the user experience for the entire site than it is to randomly divert traffic to several different versions of the site. However, all sorts of things can happen that make those time periods different: a major event that relates to the site (e.g., a sports event for a sport related site); a significant shopping period, like Valentine’s Day; a situation that causes people to be on-line more (or less). If the site attracts visitors from many different countries, the experimenter may not even be aware of all the relevant holidays or important news events.

Users should be assigned to conditions by some random process, not by, for example, an opt-in process. Opt-in can be useful for getting feedback on how well a particular user experience works, but this approach does not constitute a valid experiment. Participants in the experimental condition will be people who are early adopters, interested in technology, and perhaps need the feature being provided. The control will be full of people who do not like change, could care less about the latest technology and might find the new feature gets in the way of what they want to get done. These two groups will likely behave differently even with identical user interfaces, and we cannot accurately attribute the differences we see to the new user experience. In addition, the experiment is only measuring the reactions of a fraction of the intended population.

There are, of course, many other ways the various conditions may differ subtly. All conditions should be diverting on people from the same countries, speaking the same language, on the same types of devices (e.g., tablets are different from desktops). Similarly, all requests need to be assigned to conditions by the same method: request (every incoming request is randomly and independently assigned to a condition), cookie (all requests from a single cookie are in a particular condition throughout the entire experiment), or cookie-day (within a day, requests from a cookie are in a specific condition, but each day which condition the cookie is in is randomly selected). The decision about which method to use for assignment is a tradeoff among the need for consistency in user experience, possible changes due to learning, and independence of observations (e.g., those based on cookies or cookie-day assignments are not independent).

The idea of *counterfactuals* is important in designing and analyzing log experiments. Often the experimental change in user experience does not occur for every request. As described earlier, if the feature is to show the current weather whenever the user searches for "weather," people in the experimental condition will only see the weather information for a small fraction of searches. The counterfactuals are the requests where the person *would* have seen weather information if they had been in the experimental condition, but did not, because they were in the control. Thus it is very important to mark the counterfactual events in the control logs. This enables the analyst to easily identify the comparable subset of requests. Otherwise, any effects on the small subset of searches which change will be diluted in the larger body of unchanged behavior.

Experiment Sizing

Experiment sizing is about determining the power the experiment requires in order to detect differences of interest (Huck, 2011). A power calculation (Wikipedia: Power) is the number of observations needed to see a statistically reliable difference, assuming one exists. This determines the minimum number of data points needed in each condition of the experiment. One reason experimenters want to control for size is that log experiments are typically run with real-world customers and carry the risk of a negative user experience, so efficiently discovering whether the experience is positive or negative is important.

When sizing log experiments it is important to recognize that the kinds of changes we typically want to measure (e.g., a 5 % increase in the number of visitors who book a hotel room, a 3 % increase in the number of visitors who sign up for the newsletter or sign the petition) require large numbers of observations. This is in part because the changes are often small. But the larger issue is that log data is extremely noisy. It is an amalgam of people doing what appear to be the "same" actions, but with very different tasks and intents behind them. And in many cases, they come from multiple countries, with different languages, different cultural assumptions, etc. Thus, we often need tens of thousands of observations or more to get significant results, sometimes a lot more.

Given the need for very large numbers of observations to get statistically reliable results, it can be frustrating to run an experiment and discover that while the control and experiment conditions differ by an amount that would be considered practically meaningful, those differences are not statistically meaningful.

To estimate the number of observations needed to detect differences that are "interesting," the analyst needs to determine:

– The metric(s) of interest,
– For each metric, the minimum effect size change that the experiment should be able to detect statistically, e.g., a 2 % change in click-through rate,
– For each metric, the standard error.

Each of these is explained further below.

Deciding what effect size matters can be a challenge until an analyst or group has carried out enough experiments to know what level of change is practically important. But a larger problem is how to estimate the standard error, especially for metrics that are a ratio of two quantities (e.g., CTR (click-through-rate)—the number of clicks/number of queries). The most common problem arises when the unit of analysis is different than the experimental unit. For example, for CTR, the unit of analysis is a query, but for cookie-based experiments (as most user experience experiments will be), the experimental unit is a cookie (a sequence of queries) and we cannot assume that queries from the same cookie are independent. There are a variety of ways to calculate the standard error when the observations are not independent—two common ones are the delta method (Wikipedia: Delta method) and using "uniformity trials" (Tang et al., 2010).

Sizing is also impacted by the triggering rate, i.e., the fraction of the traffic that the experimental change actually impacts (see definition in section "Definitions"). If a particular experimental change impacts only 5 % of the traffic, then it will take 20 times as long to see an effect than if the change happens on all the traffic. Table 2 shows the effect that different triggering rates have on the number of observations (queries) needed to see an effect. Column 5 shows that more queries are required for lower triggering rates, and Column 7 shows that if counterfactuals are not logged

Table 2 The number of observations needed to achieve a given standard error as a function of the triggering rate (the fraction of traffic the experimental change impacts) and the effect size (the size of a change that would be meaningful in the experiment)

Metric standard error	Trigger rate (%)	Effect size on affected traffic (%)	Needed queries (affected)	Queries needed in expt. (counterfactuals logged)	Effect size if no counterfactuals (measured on all traffic)	Queries needed in expt. (no counterfactuals logged)
5	1	10	52,500	5,250,000	0.1 % (10 % * 1 %)	525,000,000
5	5	10	52,500	1,050,000	0.5 % (10 % * 5 %)	21,000,000
5	20	10	52,500	262,500	2 % (10 % * 20 %)	1,312,500
5	50	10	52,500	105,000	5 % (10 % * 50 %)	210,000

The required number of observations grows inversely with triggering rate and effect size

even more queries are required since experimental differences are diluted by all the observations that are the same across conditions. In practice, it helps to have some historical information to make an educated guess about what the triggered fraction will be in calculating the number of (diverted) observations needed for the given power level.

Interpreting Results

Once the experiment has been designed, users have been exposed to the experimental variants, and information has been logged about their behavior, the researcher needs to perform the analyses that will answer the original experimental questions.

Sanity Checks: The analyst's first task is to make sure that the data makes sense. An initial step is to calculate means, standard deviations, and confidence intervals for the metrics of interest. This may be done via a dashboard, such as in Google Analytics, via a script written in a general purpose language such as Python, or in a special purpose language specifically optimized for log analysis [e.g., map-reduce languages, such as Hadoop (Wikipedia: Hadoop)]. It is particularly important to look at overall traffic in all the conditions, which should be the same with random assignment. If there are differences in overall traffic across conditions, be sure to rule out the many artifacts (including bugs in logging) before assuming that an observed difference is a real effect. It is also important to break down the data in as many ways as possible—by browser, by country, etc. It may be that some differences are actually caused by a small subset of the population instead of the experimental manipulation.

Interpreting the metrics: In log analyses, it is standard practice to use confidence intervals (Huck, 2011) rather than analysis of variance significance testing, because the conditions are not organized into a factorial design with interaction terms, and because a confidence interval gives useful information about the size of the effect and its practical significance that is not as easily visible in a significance table.

It is conventional to use a 95 % confidence interval in comparing each of the experimental conditions to the control.

As described earlier, it is common to consider many different metrics, e.g., click-through rate (on results, ads, whole page), time to first click, and time on page. Given a large number of metrics, some of those supposedly significant differences are spurious (about 1 in 20, to be precise). How does one decide which to trust? Look for converging evidence; are there other metrics that ought to increase/decrease when this one does, and do they move in the appropriate direction? Might a logging error account for the effect? Is this a difference seen before in other experiments? How this is done depends on the domain and on previous experience—for example, in search, clicks per ad-shown and clicks per page are very likely to be correlated, but these click metrics are unlikely to be correlated with conversions (when someone purchased something on a page that they got to from an ad). Recognizing which changes may be artifactual comes from experience, and is somewhat of an art. It is most important to have converging evidence for the metrics that will directly impact decision-making. An important source of artifacts to be aware of are those that lead to Simpson's Paradox, which arises when ratios that have different denominators are compared (Wikipedia: Simpson's Paradox). These situations are very common in log experiments, and all experimenters should be aware of them, how to identify them, and how they change the analysis.

Now that the believable significant metrics have been identified, what is the broader interpretation of the results? Generally, there will be agreement of whether a metric's "good" direction is an increase (number of clicks) or a decrease (latency, time to click). Most likely some metrics will move in the "good" direction and some in the "bad" direction. If this is not a logging error, it is a tradeoff. People may be clicking more (because the experimental UI gives them more things to click on), but it takes them longer (because there are more things to decide among). Is the net effect positive or negative? This is always a judgment call, but a broad set of metrics may lead to a more holistic answer. For example, look for a measure of the total interaction with the site, rather than just the clicks on an individual UI element. Otherwise, go back to the original goals of the study. If the goal is to get people to the best possible information/outcome, then what they click on matters most. If the goal is to get them there quickly, then latency matters (possibly at the cost of more clicks). If the goal is to maximize the number of people who find useful information (or sign up or are able to send a message), then the fraction of people who click through to an "end result" page is the most critical measure. Sometimes it is not possible to make a single statement about what constitutes "good," there are a classic set of tradeoffs to make among speed, efficiency, usability, and design consistency. It is the analyst's job to decide and justify the decision to stakeholders (or conference paper reviewers).

Practical Significance: While we have been emphasizing the importance of statistical significance, it is also important to take practical significance into account. The experiment may show a statistically reliable difference when, say, the number of "undo" commands goes from 0.1 to 0.12 %, but that small a number might not have any practical significance (it, of course, depends on the application and what people use undo for). Do not get so blinded by the statistics (which are easy to compute) that the practical importance (which can be harder to determine) gets lost from the discussion.

Whenever possible, it is useful to have an understanding of the range of values the metrics being used take in a steady state—this is obviously not possible when just starting out with a product or prototype that has not previously been exposed to users. But knowing what typical values are and how hard it is to move those values (based on previous experiments) will help assess meaningful changes in the metrics.

So far in this chapter we have taken the existence of logs for granted. In the next two sections we consider how to collect log data, focusing on three important practical challenges: data collection, data cleaning, and using data responsibly.

Collecting, Cleaning, and Using Log Data

A typical log for a web service requires infrastructure that creates data (at the server end) about users' interactions with the service, recording all relevant information needed for later analysis. If the infrastructure also supports experiments, it must enable different users, chosen according to some appropriately random scheme, to be shown pages associated with different conditions in the experiment.

This recording infrastructure might be web server logs (Ogbuji, 2009) or logs created by special analytics packages such as Google Analytics (Google, 2012). There will be ways to configure the log-recording program to record parameters specific to the experiment or the activities the researcher is interested in (Brown, 2012).

Logs can also be created with code at the client end, but this requires users to download some sort of logging program and for that program to send data back to the server (Capra, 2011; Fox et al., 2005). Both client and server logging are lossy, but each type tends to lose different kinds of data: the server may not receive full information about aborted or timed-out operations; the client will not be aware of data that is not reported to the application that contains the logging code. Both the kind of data loss that is acceptable and the challenges of users needing to install a software plugin are important considerations for deciding between server- and client-side logging.

Data Collection

Consider a simple example of how to understand the success and strategies of searchers using a web search engine. At a minimum, a useful log would capture what queries people issue, which (if any) search results they click, and a timestamp for each of these actions. An ideal log would additionally allow the experimenter to reconstruct exactly what the user saw at the moment of behavior. But at a minimum, a web service should log:

- The time an event happened,
- The user session it was part of (often via a cookie),
- The experiment condition, if any,
- The event type and any associated parameters (e.g., on login page, user selected "create new account").

Logging a simple query is straightforward. But practitioners need to understand the different challenges that need to be addressed when collecting data for deeper analysis.

Recording accurate and consistent time is often a challenge. Web log files record many different timestamps during a search interaction: the time the query was sent from the client, the time it was received by the server, the time results were returned from the server, and the time results were received on the client. Server data is more robust but includes unknown network latencies. In both cases the researcher needs to normalize times and synchronize times across multiple machines. It is common to divide the log data up into "days," but what counts as a day? Is it all the data from midnight to midnight at some common time reference point or is it all the data from midnight to midnight in the user's local time zone? Is it important to know if people behave differently in the morning than in the evening? Then local time is important. Is it important to know everything that is happening at a given time? Then all the records should be converted to a common time zone.

The language of the interaction is often an important variable in studying behavior from people in multiple countries. If grouping by language is necessary to analyze the data, one has to determine what does "language" mean? Be careful not to confuse the user's country with their language, or the language the UI is presented in with the language of the words people type in their interactions or queries. They will often differ, especially if the experiment runs in countries where people speak multiple languages. Depending on the question of interest, it may be appropriate to partition by the language of the query or UI.

UserIDs are another challenge for identifying distinct users accurately. HTTP cookies, IP addresses, and temporary IDs are broadly applicable and easy to use, but there is a great deal of churn in such IDs (Jupiter Research Corporation, 2005). Further, cookies and temporary IDs are not uniquely associated with individuals—several people can use the same browser instance, and the same person may use multiple devices. A closer correspondence between IDs and individuals can be achieved via logins or client code, but this requires that people sign in or download client code, raising other issues (e.g., of a biased sample). Either way there is a bias in data that is captured, which can have significant impact on the results and is often overlooked by analysts.

In web logs, knowing where a page request, such as a query, came from can be important in understanding unique behavioral patterns. Queries can be generated from many different entry points—from the home page of search engines, the search results page, a browser search box or address bar, an installed toolbar, by clicking on query suggestions or other links, etc. Other applications will also have large numbers of entry points. Metadata of this kind (e.g., about the point of request origin) may be useful in later analysis, but involves additional planning and effort to collect at this stage. Without this kind of contextual information, partitioning and interpreting the data is much harder. Ultimately, all of the data and metadata collected needs to be in service of the overall goals of analysis—this defines what needs to be logged.

Data can also be distorted by exogenous factors that might also need to be considered. The site might become unexpectedly viral, perhaps being picked up by SlashDot, Reddit, or the New York Times. Virality can cause a huge swing in logged behavior. This usually occurs when a blogger shares a link that has an implicit parameter assigning all those visitors to the same condition. Even if the virality does not cause a change in how users are allocated to conditions, the behavior of people who visit out of curiosity is different from that of regular users. And if the onslaught of new users causes the site to slow down or go down, the data is even less realistic. In addition, real world events, such as the death of a major sports figure or a political event can often cause people to interact with a site differently. Again, be vigilant in sanity checking (e.g., look for an unusual number of visitors) and exclude data until things are back to normal.

Data Cleaning

A basic axiom of log analysis is that the raw data cannot be assumed to correctly and completely represent the data being recorded. Validation is really the point of data cleaning: to understand any errors that might have entered into the data and to transform the data in a way that preserves the meaning while removing noise. Although we discuss web log cleaning in this section, it is important to note that these principles apply more broadly to all kinds of log analysis; small datasets often have similar cleaning issues as massive collections. In this section, we discuss the issues and how they can be addressed.

How can logs possibly go wrong? Logs suffer from a variety of data errors and distortions. The common sources of errors we have seen in practice include:

- *Missing Events*: Sometimes client applications make optimizations that (in effect) drop events that should have been recorded. One example of this is the web browser that uses a locally cached copy of a web page to implement a "go back" action. While three pages might be visited, only two events may be logged because the visit to the cached page is not seen on the server.
- *Dropped Data*: As logs grow in size, they are frequently collected and aggregated by programs that may suffer instabilities. Gaps in logs are commonplace, and while easily spotted with visualization software, logs still need to be checked for completeness.

- *Misplaced Semantics*: For a variety of reasons, logs often encode a series of events with short (and sometimes cryptic) tags and data. Without careful, continual curation, the meaning of a log event or its interpretation can be lost. Even more subtle, small changes in the ways logging occurs can change the semantics of the logged data. (For instance, the first version of logging code might measure time-of-event from the first click; while later versions might measure time-of-event from the time the page finishes rendering—a small change that can have a substantial impact.) Since data logging and interpretation often take place at different times and with different teams of people, keeping semantics aligned is an ongoing challenge.

Data transformations: The goal of data-cleaning is to preserve the meaning with respect to an intended analysis. A concomitant lesson is that the data-cleaner *must track all transformations performed on the data*. As data is modified (e.g., removing spurious events, combining duplicates, or eliminating certain kinds of "non-signal" events), the data-cleaner must annotate the metadata for the log file with each transformation performed. Ideally, the entire chain of cleaning transformations should be maintained and tightly associated with progressive copies of the log. Not all data transformations can be reversed, but they should be recreatable from the original data set, given the log of actions taken.

The metadata associated with a dataset is necessary for other analysts to understand what the log files include. The metadata should have enough information so that the "chain of change" can be tracked back from the original file to the one that is used in the final analysis. If metadata about the cleaning of the log file is missing, analysts cannot know the semantics of the data they are analyzing. No matter how complete the record may appear, without the metadata the researcher can never be sure, and confidence in conclusions based on analysis is undermined.

Understanding the structure of the data: In order to clean log data properly, the researcher must understand the meaning of each record, its associated fields, and the interpretation of values. Contextual information about the system that produced the log should be associated with the file directly (e.g., "Logging system 3.2.33.2 recorded this file on 12-3-2012") so that if necessary the specific code that generated the log can be examined to answer questions about the meaning of the record before executing cleaning operations. The potential misinterpretations take many forms, which we illustrate with encoding of missing data and capped data values.

A common data cleaning challenge comes from the practice of encoding missing data with a value of "0" (or worse; "zero" or "−1"). Only with knowledge of what is being captured in the log files, along with the analyst's judgment of the meaning of missing data, can reasonable decisions be made about how to treat such data. Ideally, data logging systems represent missing data as NIL, ø, or some other non-confusable data value. But if the logger does not and uses a value that is potentially valid as a behavioral data value, the analyst will need to distinguish valid "0"s from missing data "0"s (for example), and manually replace the missing data with a non-confusable value.

Capped data values, usually expressing some value on a scale that has an arbitrary max (or min) value, can cause both cleaning and analysis problems. Unless the analyst knows that a particular data value being captured varies between integer values 0…9, validating data as well as making decisions about cleaning are compromised. For example, if the log is capturing data whose value is capped at 9 (because we "know the value can never go higher"), this can lead to an insight when a long string of 9's suddenly appears in the log stream.

Our point is that while the logs might not lie—they represent the values actually written out by the logging system—the interpretation of those values relies on knowledge and expertise about the data being captured on the part of the person doing the interpretation. It is not enough to know that a given action took place at a given time, the analyst also needs to know things like the possible range of values the data may take, whether measurements can be capped, and how missing data is encoded, detected, and interpreted.

Outliers: All data sets have an expected range of values, and any actual data set also has outliers that fall below or above the expected range. (Space precludes a detailed discussion of how to handle outliers for statistical analysis purposes, see: Barnett & Lewis, 1994 for details.) How to clean outliers strongly depends on the goals of the analysis and the nature of the data.

Outliers often indicate the range of human possibility, frequently in ways that experimenters (and system designers) do not anticipate. It is not uncommon in log studies to find outliers that are many standard deviations away from the mean. For example, while web search query sessions typically have a mean of around two queries/user-session, the upper end of the distribution can be in the high hundreds. People's behavior is widely variable.

If a system needs to perform correctly over a wide behavioral distribution, then outliers give valuable information about what the boundaries are, and how the system will need to respond. On the other hand, outliers can also often signal underlying exceptional cases in the logging system, the system/application being logged, or spurious signals that have been added to the log stream. Outliers should always be checked for signal integrity. That is, are outlier data points *actual* data, or are they due to some kind of system or logging error along the way? Verifying what causes the outliers to appear will dictate the approach to data cleaning that should be taken.

For a more complete description of data cleaning practices, see (Osborne, 2012).

In summary, it is important for analysts to understand the data they are analyzing. Publications that summarize a log analysis study need to include a careful description of what data cleaning steps were taken, and why they were undertaken. When a log dataset is handed from one researcher to another, data cleaning metadata must be included, along with descriptions of how the cleaning was done (preferably with pointers to the tools and settings used so that reanalysis can be done if needed).

Using Log Data Responsibly

Companies and universities may have data collection, retention, access, and use policies, and it is the researcher's responsibility to seek these out, be aware of their implications, and to make sure they comply with them in the course of carrying out research involving log data. These policies arise from internal business practices, usage agreements with users of a service, and sometimes from government regulations regarding particular types of data or privacy protections. It is fair to say that standards and best practices continue to evolve in an effort to balance privacy concerns with the potential benefits that derive from a richer understanding of user behavior. For additional perspectives on these policies and issues, see Research and Ethics in HCI, this volume.

Additional concerns arise when researchers wish to share data more broadly, often to support academic research. Risks associated with personally identifiable or sensitive information must be addressed before data is shared. Such risks may not be obvious, so researchers must proceed with due caution. Two recent examples serve to highlight the subtlety of risks associated with indirectly identifiable information—in one case involving the use of locations and names; and another case involving linking multiple sources of information.

AOL. On Aug 4, 2006, AOL search data was released to the academic community on. The data consisted of <AnonID, Query text, Query time, Rank of clicked item, URL of clicked item>. Two days later a New York Times story revealed that AnonID 4417749 was Thelma Arnold, a 62 year old woman from Lilburn GA. (Barbaro & Zeller, 2006). Two weeks later two AOL employees were fired and the CTO resigned. How did the anonymized data lead to Thelma Arnold? She issued multiple queries for businesses in Lilburn GA (a town of ~11 k people). She also issued multiple queries for people named Arnold (only 14 people with the name Arnold in Lilburn). A reporter from the NY Times contacted all 14 of these people and Thelma acknowledged to the reporter that the queries were hers. Truly anonymizing log data can be extraordinarily difficult (Wikipedia: AOL Search).

Netflix. On Oct 2, 2006 Netflix announced the Netflix Prize, an open competition for developing new collaborative filtering algorithms for predicting a user's ratings for films, along with a $1 million prize for the first team to beat the current Netflix algorithm by 10 %. The data consisted of: <MovieID, UserID, Rating, Date of Rating> <MovieID, Title, Year>. Care was taken in the data released to the public, including the introduction of random noise. The prize was awarded on Sept 21, 2009, and another competition was announced. Narayanan and Shmatikov (2008) published a paper in which they described how to de-anonymize Netflix IDs in a way that was robust to noise perturbations in the data using background knowledge of reviews in IMDB. On Dec 17, 2009 a suit was filed by one of the people identified in this manner. On Mar 12, 2010 the second Netflix competition was cancelled and no new data released (Wikipedia: Netflix).

When data is shared beyond the set of people bound by the privacy protection policies associated with the original data collection, there may be benefits to the research community, but there are also risks to the privacy of the individuals whose behavior has been logged that are extremely difficult to predict in advance. For this reason, releasing data, even when anonymized, has serious ethical risks.

Summary

In this chapter we have discussed an increasingly important way of knowing how people interact with computer systems: log analysis. As it becomes easier for computer systems to record people's interactions with technology, it is also becoming vital for HCI researchers and practitioners to know how to understand this information. We started the chapter with an eye toward understanding what we can learn about how humans interact with online systems using both observational and experimental log data. Observational log studies enable HCI researchers to summarize patterns of user interactions with existing systems. Experimental log studies enable researchers to compare behaviors in two or more systems.

Large-scale log studies give rise to a wide range of practical problems that need to be addressed to produce reliable results. From setting up the logging system, through experiment design, data collection, data cleaning and interpretation, log analysis rewards careful tracking of what is being done at every step along the way. As we have seen, it is a mistake to think that an incomplete or unreliable logging system will actually reveal deep truths about human behavior. Constant sanity checking and validation of sample sets of log data is essential to developing confidence that what is being logged and interpreted, accurately reflects human use and behaviors.

Although web logs are commonly used to understand how people interact with web services, there are few resources for becoming more expert in the method. Kohavi et al. (2009) and Tang et al. (2010) provide examples of web experimentation in practice and describe the underlying experimental infrastructure and associated analysis tools needed to carry out such experiments. Crook et al. (2009) and Kohavi et al. (2012) present interesting examples of common pitfalls, highlighting that producing reliable results requires ongoing vigilance about every aspect of experimental design, logging, and analysis. There are some available software tools for parsing web server logs and summarizing metrics (Ogbuji, 2009; Google Analytics).

Despite the challenges of dealing with massive data sets, the use of logs to extract useful insights about people's interaction with technology is becoming more common and more useful in HCI research.

Exercises

1. Name all the digital things you can think of to log. What inferences might you draw from each? What would you like to infer but can't?
2. What are the differences between logs and sensor data streams? What analysis method details are the same for both?

References

Adar, E., Teevan, J., & Dumais, S. T. (2008). Large scale analysis of web revisitation patterns. In *Proceedings of CHI 2008* (pp. 1197–1206). New York: ACM.

Baeza-Yates, R., Dupret, G., & Velasco, J. (2007). A study of mobile search queries in Japan. In *Proceedings of WWW 2007 workshop on query log analysis: Social and technical challenges*. New York, NY: ACM.

Barbaro, M. & Zeller, T. (2006). A face is exposed for AOL searcher No. 4417749, *New York Times*, Retrieved on August 9, 2006, from http://www.nytimes.com/2006/08/09/technology/09aol.html?_r=1

Barnett, V., & Lewis, S. (1994). *Outliers in statistical data*. New York, NY: Wiley & Sons.

Beitzel, S. M., Jensen, E. C., Chowdhury, A., Grossman, D. A., & Frieder, O. (2004). Hourly analysis of a very large topically categorized web query log. In *Proceedings of SIGIR 2004* (pp. 321–328). New York, NY: ACM.

Broder, A. (2002). A taxonomy of web search. *SIGIR Forum, 36*(2), 3–10.

Brown, C. (2012). Split testing with Google analytics experiments. Retrieved on December 16, 2012, from http://webdesign.tutsplus.com/tutorials/applications/split-testing-with-google-analytics-experiments/

Capra, R. (2011). HCI browser: A tool for administration and data collection for studies of web search behavior. In *Proceedings of HCIHCI 2011* (pp. 259–268). New York, NY: Springer.

Crook, T., Frasca, B., Kohavi, R., & Longbotham, R. (2009). Seven pitfalls to avoid when running controlled experiments on the web. In *Proceedings of KDD 2009* (pp. 1105–1114). New York, NY: ACM.

Dell, N., Vaidyanathan, V., Medhi, I., Cutrell, E., & Thies, W. (2012). "Yours is better!": Participant response bias in HCI. In *Proceedings of CHI 2012* (pp. 1321–1330). New York, NY: ACM.

Dumais, S. T., Cutrell, E., Cadiz, J. J., Jancke, G., Sarin, R., & Robbins, D. C. (2003). Stuff I've seen: A system for personal information retrieval and re-use. In *Proceedings of SIGIR 2003* (pp. 72–79). New York, NY: ACM.

Efthimiadis, E. N. (2008). How do Greeks search the web?: A query log analysis study. In *Proceedings iNews 2008* (pp. 81–84). New York, NY: ACM.

Fetterly, D., Manasse, M., & Najork, M. (2004). Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings WebDB 2004* (pp. 1–6). New York, NY: ACM.

Fox, S., Karnawat, K., Mydland, M., Dumais, S. T., & White, T. (2005). Evaluating implicit measures to improve web search. *ACM: Transactions on Information Systems (TOIS), 23*(2), 147–168.

Ghorab, M. R., Leveling, J., Zhou, D., Jones, G. J. F., & Wade, V. (2009). Identifying common user behaviour in multilingual search logs. In *Proceedings of CLEF 2009*, pp. 518–525.

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature, 457*, 1012–1014.

Google. (2012). Google analytics. Retrieved on December 16, 2012, from http://www.google.com/analytics/

Huck, S. (2011). *Reading statistics and research* (6th ed.). Boston, MA: Pearson.

Jansen, B. J. (2006). Search log analysis: What it is, what's been done, how to do it. *Library and Information Science Research, 28*(3), 407–432.

Jupiter Research Corporation. (2005, March 9). *Measuring unique visitors: Addressing the dramatic decline in the accuracy of cookie-based measurement*

Kohavi, R., Deng, A., Frasca, B., Longbotham, R., Walker, T., & Xu, Y. (2012). Trustworthy online controlled experiments: Five puzzling outcomes explained. In *Proceedings of KDD 2012* (pp. 786–794). New York, NY: ACM.

Kohavi, R., Longbotham, R., Sommerfield, D., & Henne, R. M. (2009). Controlled experiments on the web: Survey and practical guide. *Data Mining and Knowledge Discovery, 18*(1), 140–181.

Kotov, A., Bennett, P., White, R. W., Dumais, S. T., & Teevan, J. (2011). Modeling and analysis of cross-session search tasks. In *Proceedings of SIGIR 2011* (pp. 5–14). New York, NY: ACM.

Lau, T., & Horvitz, E. (1999). Patterns of search: Analyzing and modeling web query refinement. In *Proceedings of user modeling 1999* (pp. 119–128). New York, NY: ACM.

Narayanan, A., & Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In *Proceedings of IEEE symposium on security and privacy 2008* (pp. 111–125). Washington, DC: IEEE.

Ogbuji, U. (2009). Working with web server logs. Retrieved on December 16, 2012, fromhttp://www.ibm.com/developerworks/web/library/wa-apachelogs/

Osborne, J. W. (2012). *Best practices in data cleaning: Everything you need to know before and after collecting your data*. Thousand Oak, CA: Sage Publications.

Rodden, K., & Leggett, M. (2010). Best of both worlds: Improving Gmail labels with the affordance of folders. In *Proceedings of CHI 2010* (pp. 4587–4596). New York, NY: ACM.

Silverstein, C., Henzinger, M., Marais, H., & Moricz, M. (1998). *Analysis of a very large web search engine query log*. Technical Report 1998-014. Digital SRC.

Skinner, B. F. (1938). *The behavior of organisms: An experimental analysis*. Oxford, England: Appleton-Century.

Spink, A., Ozmutlu, S., Ozmutlu, H. C., & Jansen, B. J. (2002). U.S. versus European web searching trends. *ACM SIGIR Forum, 36*(2), 32–38.

Starbird, K. & Palen, L. (2010). Pass it on? Retweeting in mass emergencies. In *Proceedings of ISCRAM 2010*, pp. 1–10.

Tang, D., Agarwal, A., O'Brien, D., & Meyer, M. (2010). Overlapping experiment infrastructure: More, better, faster experimentation. In *Proceedings KDD 2010* (pp. 17–26). New York, NY: ACM.

Teevan, J., Adar, E., Jones, R., & Potts, M. (2007). Information re-retrieval: Repeat queries in Yahoo's logs. In *Proceedings of SIGIR 2007* (pp. 151–158). New York, NY: ACM.

Teevan, J., Dumais, S. T., & Liebling, D. J. (2008). To personalize or not to personalize: Modeling queries with variation in user intent. In *Proceedings of SIGIR 2008* (pp. 163–170). New York, NY: ACM.

Teevan, J., & Hehmeyer, A. (2013). Understanding how the projection of availability state impacts the reception of incoming communication. In *Proceedings of CSCW 2013* (pp. 753–758). New York, NY: ACM.

Teevan, J., Ramage, D., & Morris, M. R. (2011). #TwitterSearch: A comparison of microblog search and web search. In *Proceedings of WSDM 2011* (pp. 35–44). New York, NY: ACM.

Tyler, S. K., & Teevan, J. (2010). Large scale query log analysis of re-finding. In *Proceedings of WSDM 2010* (pp. 191–200). New York, NY: ACM.

White, R., Dumais, S. T., & Teevan, J. (2009). Characterizing the influence of domains expertise on web search behavior. In *Proceedings of WSDM 2009* (pp. 132–141). New York, NY: ACM.

White, R., & Morris, D. (2007). Investigating the querying and browsing behavior of advanced search engine users. In *Proceedings of SIGIR 2007* (pp. 255–262). New York, NY: ACM.

Wikipedia: AOL search. Retrieved on December 16, 2012, from http://en.wikipedia.org/wiki/AOL_search_data_scandal

Wikipedia: Delta method. Retrieved on December 16, 2012, from http://en.wikipedia.org/wiki/Delta_method

Wikipedia: Hadoop. Retrieved on December 16, 2012, from http://en.wikipedia.org/wiki/Apache_Hadoop

Wikipedia: Netflix. Retrieved on December 16, 2012, from http://en.wikipedia.org/wiki/Netflix_Prize

Wikipedia: Power. Retrieved on December 16, 2012, from http://en.wikipedia.org/wiki/Statistical_power

Wikipedia: Simpson's Paradox. Retrieved on December 16, 2012, from http://wikipedia.org/Simpsons_Paradox