Seven Rules of Thumb for Web Site Experimenters

Ron Kohavi	Alex Deng	Roger Longbotham	Ya Xu
Microsoft	Microsoft	SW Jiaotong University	LinkedIn
One Microsoft Way	One Microsoft Way	No. 111, Sec.1, N. Erhuan	2029 Stierlin Court
Redmond, WA 98052	Redmond, WA 98052	Rd. Chengdu, China 610031	Mountain View, CA 94043
ronnyk@live.com	alexdeng@microsoft.com	rogerppm@hotmail.com	xulongya@gmail.com

ABSTRACT

Web site owners, from small web sites to the largest properties that include Amazon, Facebook, Google, LinkedIn, Microsoft, and Yahoo, attempt to improve their web sites, optimizing for criteria ranging from repeat usage, time on site, to revenue. Having been involved in running thousands of controlled experiments at Amazon, Booking.com, LinkedIn, and multiple Microsoft properties, we share seven rules of thumb for experimenters, which we have generalized from these experiments and their results. These are principles that we believe have broad applicability in web optimization and analytics outside of controlled experiments, yet they are not provably correct, and in some cases exceptions are known.

To support these rules of thumb, we share multiple real examples, most being shared in a public paper for the first time. Some rules of thumb have previously been stated, such as "speed matters," but we describe the assumptions in the experimental design and share additional experiments that improved our understanding of where speed matters more: certain areas of the web page are more critical.

This paper serves two goals. First, it can guide experimenters with rules of thumb that can help them optimize their sites. Second, it provides the KDD community with new research challenges on the applicability, exceptions, and extensions to these, one of the goals for KDD's industrial track.

Categories and Subject Descriptors
G.3 Probability and Statistics/Experimental Design: controlled experiments; randomized experiments; A/B testing.

General Terms
Measurement; Design, Experimentation

Keywords
Controlled experiments; A/B testing; Online experiments

1. INTRODUCTION
Web site owners, from small web sites to the largest properties, attempt to improve their sites. Sophisticated sites use controlled experiments (e.g. A/B tests) to evaluate their changes, including Amazon [1], eBay, Etsy [2], Facebook [3], Google [4], Groupon, Intuit [5], LinkedIn [6], Microsoft [7], Netflix [8], Shop Direct [9], Yahoo, and Zynga [10].

Our experience in optimizing web sites comes from having worked on optimizing different sites, including Amazon, Booking.com, LinkedIn, and multiple Microsoft properties. Bing and LinkedIn, in particular, run hundreds of concurrent experiments at any point in time [6; 11]. Given the wide range and thousands of experiments, we have been involved in, we share useful "rules of thumb." These rules of thumb are supported by experiments, but they are sometimes known to have exceptions (we note known ones ourselves). The rule of 72 is a good example of a useful rule of thumb in the financial world. It states that you can divide 72 by the percent interest rate to determine the approximate amount of number of years it would take to double one's money in an investment. While it is very useful for the common interest range of 4% to 12%, it is known to be less accurate outside that range.

While these rules of thumb were generalized from controlled experiments, they are likely applicable in web optimization and analytics, including sites that do not run controlled experiments. However, sites that make changes without controlled evaluations will not be able to accurately assess the impact of the change.

Our contributions in this paper include:

1. Useful Rules of Thumb for web-site experimenters. We note that these are **emerging**, in the sense that we expect new research to refine their applicability and find exceptions. The value/payoff from utilizing controlled experiments is highly significant and was previously discussed in *Online Controlled Experiments at Large Scale* [11].
2. Refinement of prior rules of thumb. Observations like "speed matters" were previously stated by others [12; 13] and by us [14], but we describe the assumptions in the experimental design and share additional experiments that improved our understanding of where speed matters more: certain areas of the web page are more critical. Likewise, a perennial question is how many users are needed to run controlled experiments; we refine prior guidance of "thousands of users" [11].
3. Real examples of controlled experiments, most of which are being shared in a public paper for the first time. At Amazon, Bing, and LinkedIn, controlled experiments are used as part of the product development process [7; 11]. Many companies who are not yet using controlled experiments, can benefit from additional examples to handle the cultural challenges associated with a new product development paradigm [7; 15]. Companies already using controlled experiments can benefit from the insights shared.

The paper is organized as follows. Section 2 provides a brief introduction to controlled experiments and explains the data sources and the KDD process used in the examples. Section 3 is the heart of the paper with the rules of thumb, followed by conclusions in Section 4.

2. Controlled Experiments, the Data, and KDD Process

In the online controlled experiment we discuss here, users are randomly split between the variants (e.g., two different versions of the site) in a persistent manner (a user receives the same experience in multiple visits). Their interactions with the site are instrumented (e.g., page views, clicks) and key metrics computed (e.g., clickthrough-rates, sessions/user, revenue/user). Statistical tests are used to analyze the resulting metrics. If the delta between the metric values for Treatment and Control is statistically significant, we conclude with high probability that the change we introduced caused the observed effect on the metric. See *Controlled experiments on the web: survey and practical guide* [16] for details.

We have been involved in a lot of controlled experiments whose results were initially incorrect, and it took significant effort to understand why and correct them. Many pitfalls were documented [17; 18]. Given our emphasis on trust, we want to highlight a few things about the data and KDD process used in the online examples we present:

1. The **data sources** for the examples are the actual web sites discussed. There was no artificially generated data or data generated from a model; the examples are based on actual real user interactions after bot removal [16].
2. The **user samples** used in the examples were all uniformly randomly selected from the triggered population (e.g., an experiment that requires users to click a link to observe a difference limits to that population) [16]. User identification depends on the web site, and is cookie-based or login-based.
3. **Sample sizes** for the experiments are at least in the hundreds of thousands of users, with most experiments involving millions of users (numbers are shared in the specific examples) after bot removal, providing statistical power to detect small differences with high statistical significance.
4. Results noted were **statistically significant** with p-value$<$ 0.05, and usually much lower. Surprising results (in Rule #1) were replicated at least once more, so the combined p-value, based on Fisher's Combined Probability Test (or meta-level analysis) [19] has a much lower p-value.
5. We have personal experience with each example, which was vetted by at least one of the authors and checked against common pitfalls. Each experiment ran for at least a week, the proportions assigned to the variants were stable over the experimentation period (to avoid Simpson's paradox), and the sample ratios matched the expected ratios [17].

3. RULES of THUMB

We now state the seven rules of thumb. The first three are related to impact of changes on key metrics: small changes *can* have a big impact; changes rarely have a big positive impact; and your attempts to replicate stellar results reported by others will likely not be as successful (your mileage will vary). The latter four rules of thumb are independent with no specific order; each is a very useful generalization based on multiple experiments.

Rule #1: Small Changes can have a Big Impact to Key Metrics

Anyone who has been involved in a live site knows that small changes can have a big *negative* impact on key metrics. A small JavaScript error can render checkout impossible and small bugs corrupting memory in unmanaged code can cause servers to crash. Our focus here is therefore on *positive* differences to key metrics,

and the good news is that there are many such examples. Bryan Eisenberg wrote that removing the coupon code at checkout increased conversion rate by 1,000 percent at Doctor Footcare [20]. Jared Spool wrote that removing the registration requirement at checkout was worth $300 million a year to a large retailer [21].

While we have not seen such dramatic relative differences in experiments we have been involved in personally, we have seen dramatic improvements from small changes with surprisingly high Return-On-Investment (ROI).

We also want to highlight that we are discussing *sustained* impact, not a flash in the pan, or features exhibiting strong novelty/newness effects [16]. An example of something that we are *not* looking for is one told in *Yes!: 50 Scientifically proven ways to be Persuasive* [22]. In that book, the authors discuss how Colleen Szot authored a television program that shattered a nearly twenty-year sales record for a home-shopping channel. Szot changed three words to a standard infomercial line that caused a huge increase in the number of people who purchased her product: instead of the all-too-familiar "Operators are waiting, please call now," it was "If operators are busy, please call again." The authors explain that this is social proof: viewers think "If the phone lines are busy, then other people like me who are also watching this infomercial are calling, too."

Ploys, such as the above, will have a short shelf life if users recognize that it is used regularly. In a controlled experiment, the analysis will show an effect that quickly diminishes, which is why we recommend running experiments for two weeks and looking for such effects. In practice, novelty and primacy effects are uncommon [11; 18]. The situations where we observe them are in recommendation systems, where either the diversity itself causes a short-term effect, or when the Treatment utilizes a finite resource. For instance, when the algorithm for the People You May Know is changed at LinkedIn, it introduces a one-time diversity, which causes the new algorithm to evaluate better at the beginning (more clicks). Moreover, even if the algorithm is recommending better results, there is only a finite pool of people that one knows. After one connects to the top recommendations, the effect of the new algorithm dies down.

Example: Opening Links in new Tabs. A series of three experiments ran over time.

In Aug 2008, MSN UK ran an experiment with over 900,000 users, whereby the link to Hotmail opened in a new Tab (or new window for older browsers). We previously reported [7] that this trivial change (a single line of code) increased MSN users' engagement, as measured by clicks/user on the home page, by 8.9% for the triggered users (those who clicked on the Hotmail link).

In June 2010, we replicated the experiment on a larger population of 2.7M users on MSN in the US, and results were similar. This is also an example of an experiment that had novelty effects: on the first day the change deployed to all users, 20% of feedback messages were about this feature, most negative. In week two, the percentage went down to 4%, then 2% during the third and fourth week. The improvements to key metrics were sustained over time.

In April 2011, MSN in the US ran a very large experiment, with over 12M users, which opened the search results in a new tab/window, and engagement as measured by clicks per user increased by a whopping 5%. This was one of the best features that MSN has ever implemented in terms of increasing user engagement, and it was a trivial coding change.
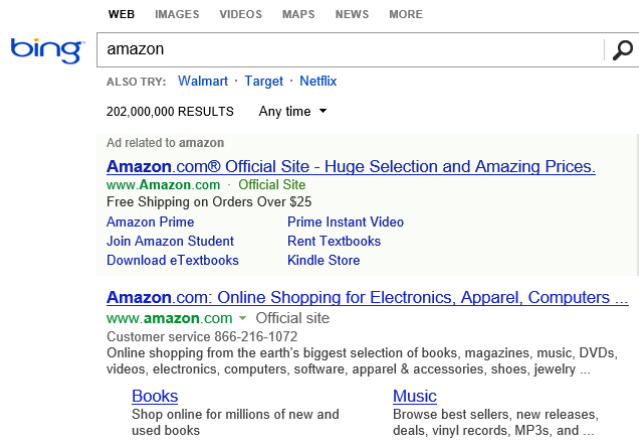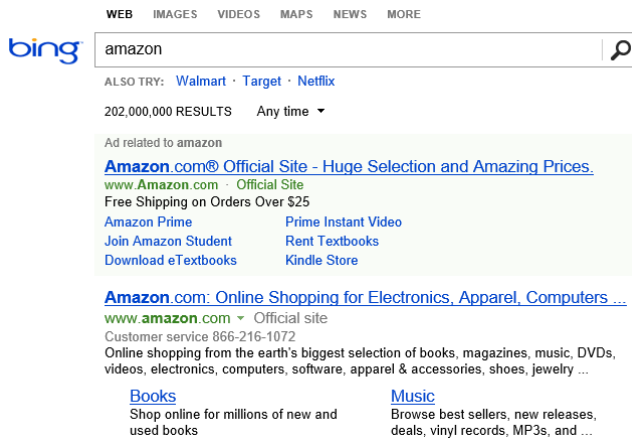
Figure 1: Font color experiment. Can you tell the difference?

All the major search engines are experimenting with opening links in new tabs/windows, but the results appear less beneficial for search engine result pages (SERPs).

Example: Font Colors. In 2013, Bing ran a set of experiments on font colors. The winning variant is shown on the right in Figure 1. To highlight the differences, the three color changes that were made are shown in this paragraph.

The cost of making such a change? Trivial: all it takes is changing several colors in the Cascading Style Sheet file (CSS). The results showed that users were more successful at completing tasks (the exact definition of success is proprietary), their time-to-success improved, and monetization improved to the tune of over $10M annually. Because such surprising results are usually viewed (rightly-so) with skepticism, this initial experiment was replicated with a much larger sample of 32 million users, and the results held.

Example: Right Offer at the Right Time. At Amazon back in 2004, the home page was split into slots, and content for the slots was tested automatically so that better content improving key metrics would be displayed more [1]. Amazon's credit-card offer was winning the top slot, which was surprising because it had very low clickthrough-rate. The reason it won was that the offer was very profitable, so despite low clickthrough-rate, the expected value was very high. But is this really the right place to offer it? No! The offer was moved to the shopping cart one sees after adding an item with some simple math shown below, highlighting the savings relative to the items in the shopping cart. Since users adding an item to the shopping cart have clear purchase intent, this offer comes at the right time.

The controlled experiment showed that this simple change was worth tens of millions of dollars in profit annually.

Example: Anti-malware. Ads are a lucrative business, and "freeware" installed by users often contains malware that pollutes pages with ads. For example, Figure 2 shows what a resulting page from Bing looked like to a user with malware, where multiple ads (highlighted in red) were added to the page. Users often do not even realize that it is not the site they are on that is showing so many ads,

but rather malware they inadvertently installed. This experiment was not trivial to code, but it was relatively simple: overriding the basic routines that modify the DOM (Document Object Model) and limiting who could modify the page. The experiment ran for 3.8 million triggered users, who had 3rd party code modifying the DOM, and the changes were blocked for users in the Treatment. The results showed improvements to all of Bing's key metrics, including the North-star metric Sessions/user, i.e., users came more often. In addition, users were more successful at reaching results, were quicker to succeed, and annual revenue improved by several million dollars. Page load time, a key metric discussed later in Rule #4 on speed, improved by hundreds of milliseconds for the triggered pages.

At Bing, two other small changes, which are confidential, took days to develop, and each increased ad revenues by about $100 million annually. Microsoft's Oct 2013 quarterly announcement noted that "Search advertising revenue grew 47% driven by an increase in revenue per search and volume." These two changes are responsible for a significant portion of that growth.

Given the above examples, one might think that the organization should focus on many small changes, but as the next rule shows, this is not the case. While breakthroughs due to small changes happen, they are very rare and surprising: at Bing, perhaps one in 500 experiments meets the bar of such high ROI and replicable positive impact. We also do not claim that these results will replicate to other domains, a point we make below, but rather that these and other easy-to-run experiments may be worth trying, in case they lead to a breakthrough.

The risk of focusing on small changes is Incrementalism: an organization should have a portfolio of small changes that potentially have high ROI, but also some big bets for the Big Hairy Audacious Goals [23].

Rule #2: Changes Rarely have a Big Positive Impact to Key Metrics

As Al Pacino says in the movie Any Given Sunday, winning is done inch by inch. For web sites like Bing, where thousands of experiments are being run annually, most fail, and those that succeed improve key metrics by 0.1% to 1.0%, once diluted to overall impact. While small changes with big positive impact discussed in Rule #1 do happen, they are the exception.

Two key points are important to highlight:

Figure 2: SERP with malware ads highlighted in red

1. Key metrics are not some specific feature metric, as those are easy to improve, but an all-up organizational metric, such as Sessions/user [18] and Time-to-success [24].
   For example, when building a feature, it is easy to significantly increase clicks to that feature (a feature metric) by highlighting it, or making it larger, but improving the overall page clickthrough-rate, or the overall experience is what really matters. Many times all the feature is doing is shifting clicks around and cannibalizing other areas of the page.
2. Metrics should be diluted by their segment size. It is much easier to improve metrics for a small segment. For example, a team can improve key metrics for weather-related queries on Bing, or purchases of TVs on Amazon by introducing a good comparison tool. However, a 10% improvement to key metrics must then be diluted to the overall impact, which takes into account the segment size. That 10% improvement to a 1% segment has an overall impact of approximately 0.1% (approximate because if the segment metrics are different than the average, the impact will be different).

The implication of this rule of thumb is significant because of occurrences of false positives. It is important to distinguish between two types of false positives:

1. Those that are expected from the Statistics. Because we run thousands of experiments a year, a false positive rate of 0.05 implies hundreds of false positive results for a given metric, and this is exacerbated if multiple uncorrelated metrics are used. For metrics like Sessions/user, even large sites like Bing do not have sufficient traffic to improve the sensitivity and result in very low p-values [18].
2. Those that are due to a bad design, data anomalies, or bugs, such as instrumentation errors.

---

[1] Ioannidis uses R as the ratio of true relationships to no relationships, so $\pi = R/(R + 1)$ and our formula is equivalent to his PPV, or Positive Predictive Value.

Results with borderline statistically significant results should be viewed as tentative and rerun to replicate the results [11]. This can be formalized using Bayes Rule [25; 26][1]. If the probability of a true positive effect is low, i.e., most ideas fail to move key metrics in a positive direction, then the probability of a true effect when the p-value is close to 0.05 is still low. Formally, if α is the statistical significance level (usually 0.05) and β is the type-II error level (normally 0.2 for 80% power), π is the prior probability that the alternative hypothesis is true, and we denote by TP a True Positive and by SS a Statistically Significant result, then we have

$$P(TP|SS) = P(SS|TP) * \frac{P(TP)}{P(SS)} = \frac{(1-\beta)\pi}{(1-\beta)\pi + \alpha(1-\pi)}$$

Using $\alpha = 0.05, \beta = 0.20$, if we have a prior probability of success of 1/3, which is what we reported is the average across multiple experiments at Microsoft [7], then the posterior probability for a true positive result given a statistically significant experiment is 89%. However, if breakthrough results noted in Rule #1 are one in 500, then the posterior probability drops to 3.1%.

One interesting corollary to this rule of thumb is that following taillights is easier than innovating in isolation. Features that we have seen introduced by statistically-savvy companies have a higher chance of having positive impact for us. If our success rate on ideas at Bing is about 10-20%, in line with other search engines, the success rate of experiments from the set of features that the competition has tested and deployed to all users is higher. This observation is symmetric: other search engines tend to test and deploy positive changes that Bing introduces too.

One of the more interesting generalizations we have made over time is not to trust results that are too good to be true. Human reaction is naturally different to results in different directions. We are inclined to resist and question negative results to our great new feature that is being tried, so we drill deeper to find the cause. However, when the effect is positive, the inclination is to celebrate rather than drill deeper and look for anomalies. When results are exceptionally strong, we learned to call out Twyman's law [27]:

Any figure that looks interesting or different is usually wrong!

Twyman's law can be explained using Bayes Rule. We have been running thousands of experiments and know that breakthrough results are rare. For example, few experiments improve our North-star metric Sessions/user significantly. Let's assume that the distribution we see in experiments is Normal, centered on 0, with a standard-deviation of 0.25%. If an experiment shows +2.0% improvement to Sessions/user, we will call out Twyman, pointing out that 2.0% is "extremely interesting" but also eight standard-deviations from the mean, and thus has a probability of 1e-15 excluding other factors. Even with a statistically significant result, the prior is so strong against this result, that we avoid any celebration and start working on finding the bug, which is usually of the second false positive type described above (e.g., an instrumentation error). Twyman's law is regularly applied to proofs that $P = NP$. No modern editor will celebrate such a submission; instead, they will send it to a reviewer to find the bug, attaching a template that says "with regards to your proof that $P = NP$, the first major error is on page x."

Example: Office Online Surrogate Metric. Cook et al. [17] reported an interesting experiment ran by Microsoft Office Online. The team tested a redesign of a page with a strong call-to-action button. The key metric the team wanted to test is the actual purchases, or purchases-per-user. However, tracking the actual purchases required hooking to the billing system, which was hard at the time. So the team decided to use "clicks on revenue generating links" assuming clicks * conversion-rate = revenue, where the conversion-rate is from click to purchase.

To their surprise, there was a 64% reduction in clicks per user. This shocking result made people look deeper into the data. It turns out that the assumption of a stable conversion rate from click to purchase was flawed. The Treatment page, which showed the price of the product, attracted fewer clicks, but those users were better qualified and had a much higher conversion-rate.

Example: More Clicks from a Slower Page. JavaScript code was added to Bing's search result page. This additional script normally slows things, so one expected to see a small negative impact on key metrics measuring user engagement such as clicks-per-user. However, the results showed the opposite: users clicked more [18]**!** In spite of the positive movement, we followed Twyman's law and solved the puzzle. Click tracking is based on web beacons and some browsers eliminate the call when the user is navigating away from the page [28]**.** The additional JavaScript had a side effect of improving click tracking fidelity, not actual user clicks.

Example: Bing Edge. Over a period of several months in 2013, Bing switched its Content Delivery Network (CDN) from Akamai to its own Bing Edge. The Traffic ramp-up to Bing's Edge occurred together with many other improvements Bing deployed during this period. Several teams reported that key metrics improved over time: the Bing Home page clickthrough-rate was improving, features were used more, and our abandonment rates were coming down. It turns out that these improvements were related to click tracking fidelity: Bing's Edge improved not just page performance, but also click tracking fidelity. To quantify the impact, we ran an experiment where we replaced the beacon-based click tracking with redirects, a technique used in tracking ad clicks that has negligible click loss, but introduces a slowdown per click. The results showed that the click loss rate for some browsers dropped by more than 60%! A large portion of the gains over time were actually an artifact of improved click tracking.

Example: MSN Searches to Bing. The auto-suggest feature shows a drop-down box with possible completions and variants below a search box, as the user is typing. An experiment at MSN attempted to improve this feature with a new and better algorithm (feature teams are always able to explain a-priori why the new feature is going to be better before the experiment, but are often disappointed by the results). The experiment was a huge success with the number of searches on Bing referred from MSN dramatically improving. Given this rule of thumb, we investigated more deeply and it turns out the new code was effectively issuing two searches when users selected one of the auto-suggested options (one was always disconnected by the browser as only one SERP was displayed).

Although the explanations of many positive results may not be as exciting as if the improvements were real, our goal is to find true user impact, and Twyman's law has improved our fundamental understanding in multiple cases.

Rule #3: Your Mileage WILL Vary

There are many documented examples of successes using controlled experiments. For example, Anne Holland's "Which Test Won?" site (http://whichtestwon.com) has hundreds of case studies of A/B tests, and a new case is added about every week.

While these are great idea generators, there are several problems with such case studies

1. The quality varies. In these studies, someone at some company reported the result of an A/B experiment. Was it peer reviewed? Was it properly run? Were there outliers? Was the p-value low (we've seen tests published where the p-value was > 0.05, which is normally considered not statistically significant). There are pitfalls we have warned about [17] and many experiments do not properly check for issues.

2. What works in one domain may not work in another. For example, Neil Patel [29] recommends using the word "free" in ads and offering "30-day free trial" instead of "30-day money back guarantee." These may work for certain products and audience, but we suspect that results will vary by domain. Joshua Porter [30] reported that "Red Beats Green" for the call-to-action button "Get Started Now" on a web site. Since we do not see a lot of sites with red call-to-action buttons, we believe this is not a general result that replicates well.

3. Novelty and Primacy effects. As discussed previously, we are looking for sustained improvements, and many experiments were not run long enough to check for such effects.

4. Misinterpretation of result. Effects are often attributed to a specific factor, or the underlying reason is not understood. Below are two examples; the first is one of the earliest documented controlled experiments.

Historical Example: Lack of medical knowledge about Vitamin C. Scurvy is a disease that results from vitamin C deficiency. It killed over 100,000 people in the 16th-18th centuries, mostly sailors who went out for long-distance voyages and stayed at sea longer than perishable fruits could be stored. In 1747, Dr. James Lind noticed lack of scurvy in Mediterranean ships and gave some sailors oranges and lemons (Treatment), and others ate regular diet (Control). The experiment was very successful, but Dr. Lind did not understand the reason. At the Royal Naval Hospital in England, he treated scurvy patients with concentrated lemon juice called "rob." He concentrated the lemon juice by heating it, thus destroying the vitamin C. He lost faith in the remedy and became increasingly reliant on bloodletting. In 1793, a formal trial was done and lemon juice became part of the daily rations throughout the navy; Scurvy was quickly eliminated and British sailors are called Limeys to this day.

Example: **Speed vs. more results.** In a Web 2.0 talk by Marissa Mayer, then at Google, she described an experiment where Google increased the number of search results on the SERP from ten to thirty [31]. Traffic and revenue from Google searchers in the experimental group dropped by 20%. Her explanation? The page took half a second more to generate. Performance is a critical factor, but we suspect it only accounts for a small percentage of the loss. Here are three reasons:

a. Slowdown experiments that ran at Bing [11] isolated just the performance factor. Numbers showed that a 250msec delay at the server impacts revenue at about 1.5% and clickthrough-rate by 0.25%. While this is a massive impact, 500msec would impact revenue about 3% not 20%, and clickthrough-rate would drop by 0.50%, not 20% (assuming a linear approximation is reasonable). Earlier tests at Bing

[32] had similar click impact and smaller revenue impact with delays of up to two seconds.

b. Jake Brutlag from Google blogged about an experiment [12] showing that slowing down the search results page by 100 to 400 milliseconds has a measurable impact on the number of searches per user, which declined 0.2% to 0.6%, very much in line with our experiment, but far from the results reported in Marissa Mayer's talk.

c. An experiment ran at Bing, where 20 results were shown instead of 10. The revenue loss was nullified by adding another mainline ad (which slowed the page a bit more). We believe the ratio of ads to algorithmic results plays a more important role than performance.

We are skeptical of many amazing results of A/B tests reported in the literature. When reviewing results of experiments, ask yourself what trust level to apply, and remember that even if the idea worked for the specific site, it may not work as well for another. One of the best things we can do is to report replications of prior experiments (successful or not). This is how science works best.

Rule #4: Speed Matters a LOT

Web site developers that evaluate features using controlled experiments quickly realize web site performance, or speed, is critical **[13; 14; 33]**. Even a slight delay to the page performance may impact key metrics in the Treatment.

The best way to quantify the impact of performance is to isolate just that factor using a slowdown experiment, i.e., add a delay. Figure 3 shows a graph of depicting a common relationship between time (performance) and a metric of interest (e.g., clickthrough-rate per page, success rate per session, or revenue per user). Typically, the faster the site, the better (higher in this example) the metric value. By slowing the Treatment relative to Control, you can measure the impact on the metric of interest. There are a few key points about such an experiment

1. The slowdown quantifies the impact on the metric of interest at the point today, shown by the dotted vertical line in Figure 3. If the site performance changes (e.g., site is faster), or the audience changes (e.g., more international users) the impact may be different.
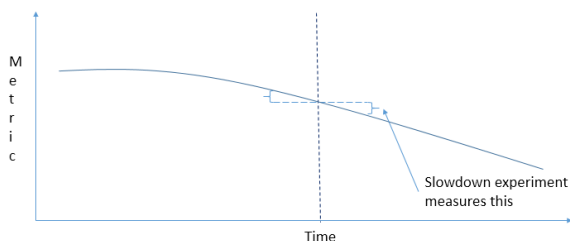

Figure 3: Typical relationship between performance (time) and a metric of interest

2. The experiment measures the impact of a slowdown. This is very useful when trying to assess the value of a feature whose first implementation is inefficient: say it moves metric M by X% and also slows the site by T%. Using the slowdown experiment, we can estimate the impact of the slowdown on metric M, and thus adjust the impact of the feature to X'% (assuming additivity), thus answering the question of its impact *if* it were implemented efficiently.

3. We can assess the impact to key metrics if the site were faster, helping us evaluate ROI (Return-On-Investment) of such

efforts. Using a linear approximation (1st-order Taylor expansion), we can assume that the impact of the metric is similar in both directions (slowdown and speedup). As shown in Figure 3, we assume that the vertical delta on the right is similar to that on the left. By running slowdown experiments with different slowdown amounts, we have confirmed that a linear approximation is very reasonable for Bing.

How important is performance? Critical. At Amazon, 100msec slowdown decreased sales by 1% as shared by Greg Linden [34 p. 10]. A talk by speakers from Bing and Google [32] showed the significant impact of performance on key metrics.

Example: Server slowdown experiment. A slowdown experiment at Bing [11] slowed 10% of users by 100msec (milliseconds) and another 10% by 250msec for two weeks. The results of this controlled experiment showed that every 100msec speedup improves revenue by 0.6%. The following phrasing resonated extremely well in our organization (based on translating the above to profit): *an engineer that improves server performance by 10msec (that's 1/30 of the speed that our eyes blink) more than pays for his fully-loaded annual costs*. Every millisecond counts.

The above experiments slowed down the server's response, thus slowing down all elements of the page. It is natural to assume that some areas of the page are more important. For example, users cannot tell that elements "below the fold" (i.e., below what's visible in the current window) [35] have not been loaded yet without scrolling. Are there some elements can could be shown late, with little user impact? The following controlled experiment shows that this is indeed the case.

Example: Performance of the right pane is less critical. At Bing, some elements on the right pane (called the snapshot) are loaded late (technically, after the `window.onload` event). A recent slowdown controlled experiment was run, similar to the one described above, delaying when the right pane elements were shown by 250 milliseconds. If there was an impact on key metrics, it was not detectible, despite the experiment size of almost 20 million users.

Page Load Time (PLT) is often used to measure performance using the `window.onload` to mark the end of the useful browser activity. However, this metric has severe deficiencies with modern web pages. As Steve Souders showed [36], an Amazon page can render in 2.0 seconds above the fold, but the `window.onload` event fires at 5.2 seconds. Schurman [32] reported that being able to progressively render a page, so the header shows up early, helps. The opposite is also true with Gmail as a good example: the `window.onload` fires at 3.3 seconds, at which point only the progress bar is visible, and the above-the-fold content shows at 4.8 seconds.

There are other metrics commonly measured, such as time to first result (e.g. time to first tweet on Twitter, first algorithmic result on a SERP), but the term "Perceived performance" is often used to denote the intuitive idea that users start to interpret the page once enough of it is showing. The concept of perceived performance is easier to state abstractly than measure in practice, and `perception.ready()` isn't on any browser's roadmap [36]. Multiple proposals have been developed to estimate perceived performance, including

1. Above the Fold Time (AFT) [37], which measure the time until pixels above the fold have been painted. Implementations need to use heuristics to handle videos, animated GIFs, rotating galleries, and other dynamic content that changes the

page above the fold. Thresholds may be set for "percent of pixels painted" to avoid trivial elements of little consequence from prolonging the measured time.

2. Speed Index [38] is a generalization of AFT, which averages the time at which visible elements on the page are displayed. This does not suffer from trivial elements showing late, but still suffers from dynamic content changing above-the-fold.

3. Page Phase Time and User-Ready Time [39]. Page Phase Page Time requires identifying which rendering phase satisfies perceived performance, and phases are determined by pixel changing velocity. User-Ready time measures the time until essential elements of the page (defined for each context) are ready to use.

New W3C timing interfaces are being made available in newer HTML standards, which provide access to finer-grained events and may help understand performance issues better. The above experiments are all on desktop, and there is a lot to learn for mobile.

At Bing, we use multiple performance metrics for diagnostics, but our key time-related metric is Time-To-Success (TTS) [24], which side-steps the measurement issues. For a search engine, our goal is to allow users to complete a task faster. For clickable elements, a user clicking faster on a result from which they do not come back for at least 30 seconds is considered a successful click. TTS as a metric captures perceived performance well: if it improves, then important areas of the pages are rendering faster so that users can interpret the page and click faster. This relatively simple metric does not suffer from heuristics needed for many performance metrics. It is highly robust to changes, and very sensitive. Its main deficiency is that it only works for clickable elements. For queries where the SERP has the answer (e.g., for "time" query), users can be satisfied and abandon the page without clicking.

Rule #5: Reducing Abandonment is Hard, Shifting Clicks is Easy

A key metric that Bing measures in controlled experiment is abandonment rate on the SERP (Search Engine Results Page): the percentage of users who never click on any link. Increasing user engagement, or reducing abandonment, is considered positive, but it is a difficult metric to move. Most experiments show that there can be significant shifts in clicks from one area of the page to another, but abandonment rate rarely moves or moves very little. Below we share several examples of experiments where significant changes were made, yet abandonment rate did not change statistically significantly.

Example: **Related Searches in right column.** Some related searches were removed from the right column on Bing's SERP for an experiment with over 10 million users. If a user searches for "data mining" Bing will normally show related searches, such as "Examples of Data Mining," "Advantages of Data Mining," "Definition of Data Mining," "Data Mining Companies," "Data Mining Software," etc. These can help users modify their query (e.g., refine it) and help them be more successful. In the experiment, clicks shifted to other areas of the page, but abandonment rate did not change statistically significantly (p-value 0.64).

Example: **Related Searches below bottom ads.** Bing shows related searches inline, and these are allowed to float higher if their clickthrough-rate is better than the algorithmic results below them. In an experiment, with over 5 million users, these were pinned to the bottom of the page, below the bottom ads. The clickthrough-rate on these related searches declined 17%, but the abandonment rate did not change statistically significantly (p-value 0.71)

Example: **Truncating the SERP.** Bing dynamically sizes the SERP, not always showing the classical ten blue links. This change was motivated by the stability of the abandonment rate. For example, here are two experiments.

1. When there is a deep-link block for queries like "ebay," the click-through rate on the top block is over 75%. Showing 10 results for such queries is of little value, and the SERPs for these was truncated to show four algorithmic results in an experiment with over 8 million triggered users (triggered means at least one of their queries showed a page with a deep-links block), the abandonment rate did not change statistically significantly for these pages (p-value 0.92). These pages were therefore faster, and this feature was released.

2. When users navigate from the SERP, but come back either using the browser's back button or by reissuing the query, Bing extends the page and shows more results (14 results). In an experiment with over 3 million triggered users, the page was extended to 20 results and removed related searches. There were significant changes to metrics, including: 1.8% reduction in revenue, 30msec slowdown for page load time, 18% reduction in pagination, but abandonment rate did not change statistically significantly (p-value 0.93). This change was not released.

Example: Ad background color. All major search engines have been experimenting with changing the background color for ads. In a recent experiment with over 10M users, the Treatment color caused a 12% decline in revenue (an annual loss of over $150M if this change were made). Users shifted their clicks from ads to other areas of the page, but abandonment rate did not change statistically significantly (p-value 0.83).

We have observed cases where abandonment improves, such as when we made significant improvements to relevance, and in the Anti-malware flight discussed in Rule #1, but these are uncommon and the movements are smaller than one might expect.

This rule of thumb is extremely important because we have seen many experiments (at Microsoft, Amazon, and reported by others) where a module or widget was added to the page with relatively good click-through rates. The claim is made that new module is clearly good for users because users are clicking. But if the module simply cannibalized other areas of the page, as shown in the examples above, it is only useful if those clicks are better, however "better" is defined for the site (e.g., they lead to higher success or purchases, etc.). Phrased differently: local improvements are easy; global improvements are much harder.

Rule #6: Avoid Complex Designs: Iterate

Good experimental design is vital to getting the best results from experiments. Sir R. A. Fisher once said **[40]** "To consult the statistician after an experiment is finished is often merely to ask him to conduct a post mortem examination. He can perhaps say what the experiment died of." Our experience is that simple designs are best in the online world and given the many pitfalls **[17; 41]**, they are easier to understand, run sanity checks, and thus more trustworthy. A complex design is usually not only unnecessary, but can hide bugs. We share a couple of examples from LinkedIn.

Example: LinkedIn Unified Search. At LinkedIn, a product launch usually involves multiple features/components. One big upgrade to LinkedIn Search launched in 2013 involved improved autocomplete and suggested phrasing, and most importantly, it introduced unified search across different product categories. In the past, search had to take in a facet, whether it is "People", or "Jobs",

or "Companies". With unified search, the search box is smart enough to figure out your query intent and find the relevant results. However, that was not all. Almost every single component on the search landing-page was touched, from the left rail navigation to snippets to the action buttons. The first experiment was run with all changes lumped together and many key metrics tanked. It was a lengthy process to bring back one feature at a time to realize that certain features (removed from final launch), not the unified search, were responsible for bringing down clicks and revenue. After restoring these features, unified search was shown to be positive to the user experience and deployed to everyone.

Example: LinkedIn Contacts. LinkedIn recently introduced the new contacts page that helps people to stay in touch better with their relationships. It was believed to be a great feature for users. However, when the results came back from the experiment, they looked horrifying. The experiment had a very complex design that made it hard to investigate what went wrong. First of all, the experiment was designed to only impact users who were not in a whitelist. To achieve that there was an eligibility check before the experiment was even triggered. Second, depending on whether a user fell into the Treatment or Control, two other experiments would be triggered that may show the new contacts page to that user. The complex design left many possibilities for error and it took days to figure out that the eligibility check was implemented with the following bug: if a user has seen the new feature once, he/she is put on the whitelist that is removed entirely from the experiment! No wonder we saw engagement dropping, as treatment users appeared to churn after one visit!

With offline experiments, where experiments are expensive relative to the design and analysis, it makes sense to make maximum use of the users (experimental units). However, online we have a continuous stream of users and we can use concurrent designs [4; 11] to run hundreds of concurrent experiments, testing one or two variables at a time. While the literature on Multi-Variable Testing (MVT) is rich, and commercial products tout their MVT capabilities, we usually find it more beneficial to run simple uni-variable (e.g., A/B/C/D variant of a feature) or bi-variable designs.

Another important reason for running simple univariable designs is to align with agile software methodologies and building minimum viable products (MVPs) [15]. Instead of building code for a complex MVTs, run an experiment as soon as a key feature is ready. There is always significant learning from exposing new features to users, such as seeing unexpected metrics move, getting verbatim feedback, finding bugs, etc. Complicated MVTs that rely on a lot of new code tend to be invalid because bugs are found in the code for at least one of the variables.

We encourage our engineering teams to deploy new code quickly and use experiments to provide a form of exposure control: start with small 1% treatments, then ramp up if there are no egregious declines in key metrics. With agile methodologies now common, without exposure control provided through controlled experiments, you run the risk of repeating a deployment like the one Knight Capital did, which in Aug 2012 caused a $440 million loss and erased 75% of Knight's equity value.

Rule #7: Have Enough Users

Experimentation methodologies typically rely on the means, which are assumed to be normally distributed. The well-known Central Limit Theorem shows that the mean of a variable has an approximately normal distribution if the sample size is large enough. Applied statistics books will suggest that small numbers usually suffice. For example, one [42] states "In many cases of

practical interest, if $n \geq 30$, the normal approximation will be satisfactory regardless of the shape of the distribution." Because we are looking at statistical significance using the tails of distributions, larger sample sizes are required. Our advice in previous articles [11] is that you need "thousands" of users in an experiment; Neil Patel [29] suggests 10,000 monthly visitors, but the guidance should be refined to the metrics of interest.

Formulas for minimum sample size given the metric's variance and sensitivity (the amount of change one wants to detect) provide one lower bound [16], but these assume that the distribution of the mean is normal. Our experience is that many metrics of interest in online experiments are skewed which may require a higher lower bound before you can assume normality.

Our rule of thumb for the minimum number of independent and identically distributed observations needed for the mean to have a normal distribution is $355 \times s^2$ for each variant, where s is the skewness coefficient of the distribution of the variable X defined as

$$s = \frac{E[X - E(X)]^3}{[Var(X)]^{3/2}}.$$

We recommend the use of this rule when the |skewness| > 1. The following table shows the minimum sample size required under this rule of thumb for a few select metrics from Bing, and the sensitivity (% change detectable at 80% power) such a sample size provides.

Metric	|Skewness|	Sample Size	Sensitivity
Revenue/User	18.2	114k	4.4%
Revenue/User (Capped)	5.3	9.7k	10.5%
Sessions/User	3.6	4.70k	5.4%
Time To Success	2.1	1.55k	12.3%

At a commerce site, the skewness for purchases/customer was >10 and for revenue/customer >30. This rule of thumb gives a 95% confidence interval for the mean such that both two tail probabilities (nominally 0.025) are no greater than 0.03 and no less than 0.02. This rule was derived from the work of Boos and Hughes-Oliver [43]. Long tailed distributions are common with web data and can be quite skewed. In the table above, we found Revenue/User had a skewness of 18.2 and therefore 114k users were needed. Figure 4 shows when we only sample 100 and 1,000 users, the distribution of the sample mean is quite skewed and the 95% two-side confidence interval assuming normality would miss the true mean more than 5%. When we increase sample size to 100k, the distribution of sample mean is very close to normal for the range of -2 to 2.
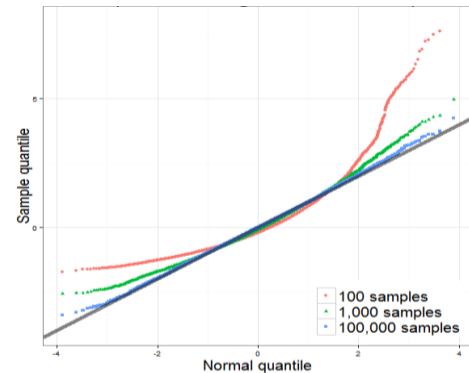

Figure 4: QQ-norm plot for averages of different sample sizes showing convergence to Normal when skewness is 18.2 for Revenue/user. Actual data used

When a metric has a large skewness, it is sometimes possible to transform the metric or cap the values to reduce the skewness so that the average converges to normality faster. After we capped Revenue/User to $10 per user per week, we saw skewness drop from 18 to 5.3 and sensitivity (i.e. power) increased. For the same sample size, Capped Revenue per user can detect a change 30% smaller than Revenue per user.

Our rule of thumb assesses the number of users needed to make the distribution of the mean be well approximated by a normal distribution. If the control and treatment are expected to have the same distribution, there is an important recommendation we can make: ensure that the control and treatment are equally sized. If the split is equally sized (e.g., 50%/50%), then the distribution of the delta will be approximately symmetric (it will be perfectly symmetric with zero skewness under the Null hypothesis) and our rule of thumb does not provide a useful lower bound (we recommended the rule be used when |skewness| > 1). Power calculations will typically provide the lower bound for sample size [16]. For skewed distributions with small samples, one can use bootstrapping techniques [44].

4. Summary

We presented seven rules of thumb for web site experiments, which we have developed based on thousands of online controlled experiments, and supported by examples. The first two show that small changes can have a big positive impact, but that they are rare and most progress is made through many small improvements over time. When results seem too good to be true, apply Twyman's law and investigate deeply before declaring an experiment as a breakthrough; most of the time, you'll find a bug. The third rule warns about claimed results "in the wild," which we learned to be cautious about. Make sure to replicate ideas, as they may not have the same effect (or even a positive effect). The fourth rule is an area we are passionate about: speed. We ran multiple experiments and better understand the relationship between performance and key metrics, showing that server speed is critical; in addition, displaying the key parts of the page faster is more important than others, such as sidebars. Despite our passion, we doubt some extreme results about performance, which we reviewed with the third rule. The fifth rule is an empirical observation that we suspect will be refined over time, but it is surprising how widely it holds: changing abandonment rate is really hard, and most experiments just shift clicks around, so one has to be careful about local optimizations. The sixth rule recommends simpler designs and quicker iterations, which aligns with modern agile software development methodologies. The seventh rule provides a lower bound for the number of users for skewed metrics, which are common in online experiments. Most examples shared here are being shared in a public paper for the first time. They support the rules of thumb, and also strengthen our conviction in the value of experimentation to help guide product development. We hope these rules of thumb will serve the community and will lead to follow-on research that will refine them and provide additional rules of thumb.

ACKNOWLEDGMENTS

References

1. **Kohavi, Ron and Round, Matt.** *Front Line Internet Analytics at Amazon.com.* [ed.] Jim Sterne. Santa Barbara, CA : s.n., 2004. http://ai.stanford.edu/~ronnyk/emetricsAmazon.pdf.

2. **McKinley, Dan.** Design for Continuous Experimentation: Talk and Slides. [Online] Dec 22, 2012. http://mcfunley.com/design-for-continuous-experimentation.

3. **Bakshy, Eytan and Eckles, Dean.** Uncertainty in Online Experiments with Dependent Data: An Evaluation of Bootstrap Methods. *KDD 2013: Proceedings of the 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 2013.

4. **Tang, Diane, et al.** Overlapping Experiment Infrastructure: More, Better, Faster Experimentation. *Proceedings 16th Conference on Knowledge Discovery and Data Mining.* 2010.

5. **Moran, Mike.** Multivariate Testing in Action: Quicken Loan's Regis Hadiaris on multivariate testing. *Biznology Blog by Mike Moran.* [Online] December 2008. www.biznology.com/2008/12/multivariate_testing_in_action/.

6. **Posse, Christian.** Key Lessons Learned Building LinkedIn Online Experimentation Platform. *Slideshare.* [Online] March 20, 2013. http://www.slideshare.net/HiveData/googlecontrolled-experimentationpanelthe-hive.

7. **Kohavi, Ron, Crook, Thomas and Longbotham, Roger.** Online Experimentation at Microsoft. *Third Workshop on Data Mining Case Studies and Practice Prize.* 2009. http://exp-platform.com/expMicrosoft.aspx.

8. **Amatriain, Xavier and Basilico , Justin.** Netflix Recommendations: Beyond the 5 stars. [Online] April 2012. http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html.

9. **McFarland, Colin.** *Experiment!: Website conversion rate optimization with A/B and multivariate testing.* s.l. : New Riders, 2012. 978-0321834607.

10. **Smietana, Brandon.** Zynga: What is Zynga's core competency? *Quora.* [Online] Sept 2010. http://www.quora.com/Zynga/What-is-Zyngas-core-competency/answer/Brandon-Smietana.

11. **Kohavi, Ron, et al.** Online Controlled Experiments at Large Scale. *KDD 2013: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining.* 2013. http://bit.ly/ExPScale.

12. **Brutlag, Jake .** Speed Matters . *Google Research blog.* [Online] June 23, 2009. http://googleresearch.blogspot.com/2009/06/speed-matters.html.

13. **Sullivan, Nicole .** Design Fast Websites. *Slideshare.* [Online] Oct 14, 2008. http://www.slideshare.net/stubbornella/designing-fast-websites-presentation.

14. **Kohavi, Ron, Henne, Randal M and Sommerfield, Dan.** Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO. *The Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007).* August 2007, pp. 959-967. http://www.exp-platform.com/Documents/GuideControlledExperiments.pdf.

15. **Ries, Eric.** *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses .* s.l. : Crown Business, 2011. 978-0307887894.

16. **Kohavi, Ron, et al.** Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery.* February 2009, Vol. 18, 1, pp. 140-181. http://www.exp-platform.com/Pages/hippo_long.aspx.

17. **Crook, Thomas, et al.** Seven Pitfalls to Avoid when Running Controlled Experiments on the Web. [ed.] Peter Flach and Mohammed Zaki. *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* 2009, pp. 1105-1114. http://www.exp-platform.com/Pages/ExPpitfalls.aspx.

18. **Kohavi, Ron, et al.** Trustworthy online controlled experiments: Five puzzling outcomes explained. *Proceedings of the 18th Conference on Knowledge Discovery and Data Mining.* 2012, www.exp-platform.com/Pages/PuzzingOutcomesExplained.aspx.

19. **Wikipedia contributors.** Fisher's method. *Wikipedia.* [Online] Jan 2014. http://en.wikipedia.org/wiki/Fisher%27s_method.

20. **Eisenberg, Bryan.** How to Increase Conversion Rate 1,000 Percent. *ClickZ.* [Online] Feb 28, 2003. http://www.clickz.com/showPage.html?page=1756031.

21. **Spool, Jared.** The $300 Million Button. *USer Interface Engineering.* [Online] 2009. http://www.uie.com/articles/three_hund_million_button/.

22. **Goldstein, Noah J, Martin, Steve J and Cialdini, Robert B.** *Yes!: 50 Scientifically Proven Ways to Be Persuasive.* s.l. : Free Press, 2008. 1416570969.

23. **Collins , Jim and Porras , Jerry I.** *Built to Last: Successful Habits of Visionary Companies.* s.l. : HarperBusiness, 2004. 978-0060566104.

24. **Badam, Kiran.** Looking Beyond Page Load Times – How a relentless focus on Task Completion Times can benefit your users. *Velocity: Web Performance and Operations.* 2013. http://velocityconf.com/velocityny2013/public/schedule/detail/32820.

25. *Why Most Published Research Findings Are False.* **Ioannidis, John P.** 8, 2005, PLoS Medicine, Vol. 2, p. e124. http://www.plosmedicine.org/article/info:doi/10.1371/journal.pmed.0020124.

26. **Wacholder, Sholom, et al.** Assessing the Probability That a Positive Report is False: An Approach for Molecular Epidemiology Studies. *Journal of the National Cancer Institute.* 2004, Vol. 96, 6. http://jnci.oxfordjournals.org/content/96/6/434.long.

27. **Ehrenberg, A. S. C.** The Teaching of Statistics: Corrections and Comments. *Journal of the Royal Statistical Society.* Series A, 1974, Vol. 138, 4.

28. **Ron Kohavi, David Messner,Seth Eliot, Juan Lavista Ferres, Randy Henne, Vignesh Kannappan,Justin Wang.** *Tracking Users' Clicks and Submits: Tradeoffs between User Experience and Data Loss.* Redmond : s.n., 2010.

29. **Patel , Neil .** 11 Obvious A/B Tests You Should Try. *QuickSprout.* [Online] Jan 14, 2013. http://www.quicksprout.com/2013/01/14/11-obvious-ab-tests-you-should-try/.

30. **Porter, Joshua.** The Button Color A/B Test: Red Beats Green. *Hutspot.* [Online] Aug 2, 2011. http://blog.hubspot.com/blog/tabid/6307/bid/20566/The-Button-Color-A-B-Test-Red-Beats-Green.aspx.

31. **Linden, Greg.** Marissa Mayer at Web 2.0 . *Geeking with Greg .* [Online] Nov 9, 2006. http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html.

32. *Performance Related Changes and their User Impact.* **Schurman, Eric and Brutlag, Jake.** s.l. : Velocity 09: Velocity Web Performance and Operations Conference, 2009.

33. **Souders, Steve.** *High Performance Web Sites: Essential Knowledge for Front-End Engineers.* s.l. : O'Reilly Media, 2007. 978-0596529307.

34. **Linden, Greg.** Make Data Useful. [Online] Dec 2006. http://sites.google.com/site/glinden/Home/StanfordDataMining.2006-11-28.ppt.

35. **Wikipedia contributors.** Above the fold. *Wikipedia, The Free Encyclopedia.* [Online] Jan 2014. http://en.wikipedia.org/wiki/Above_the_fold.

36. **Souders, Steve.** Moving beyond window.onload(). *High Performance Web Sites Blog.* [Online] May 13, 2013. http://www.stevesouders.com/blog/2013/05/13/moving-beyond-window-onload/.

37. **Brutlag, Jake, Abrams, Zoe and Meenan, Pat .** Above the Fold Time: Measuring Web Page Performance Visually. *Velocity: Web Performance and Operations Conference.* 2011. http://en.oreilly.com/velocity-mar2011/public/schedule/detail/18692.

38. **Meenan, Patrick.** Speed Index. *WebPagetest.* [Online] April 2012. https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index.

39. **Meenan, Patrick, Feng, Chao (Ray) and Petrovich, Mike .** Going Beyond onload - How Fast Does It Feel? *Velocity: Web Performance and Operations.* 2013. http://velocityconf.com/velocityny2013/public/schedule/detail/31344.

40. **Fisher, Ronald A.** Presidential Address. *Sankhyā: The Indian Journal of Statistics.* 1938, Vol. 4, 1. http://www.jstor.org/stable/40383882.

41. **Kohavi, Ron and Longbotham, Roger.** Unexpected Results in Online Controlled Experiments. *SIGKDD Explorations.* 2010, Vol. 12, 2. http://www.exp-platform.com/Documents/2010-12%20ExPUnexpectedSIGKDD.pdf.

42. **Montgomery, Douglas C.** *Applied Statistics and Probability for Engineers.* 5th. s.l. : John Wiley & Sons, Inc, 2010. 978-0470053041.

43. **Boos, Dennis D and Hughes-Oliver, Jacqueline M.** How Large Does n Have to be for Z and t Intervals? *The American Statistician.* 2000, Vol. 54, 2, pp. 121-128.

44. **Efron, Bradley and Robert J. Tibshirani.** *An Introduction to the Bootstrap.* New York : Chapman & Hall, 1993. 0-412-04231-2.