2

Running and Analyzing Experiments

An End-to-End Example

> The fewer the facts, the stronger the opinion
> – *Arnold Glasow*

In Chapter 1, we reviewed what controlled experiments are and the importance of getting real data for decision making rather than relying on intuition. The example in this chapter explores the basic principles of designing, running, and analyzing an experiment. These principles apply to wherever software is deployed, including web servers and browsers, desktop applications, mobile applications, game consoles, assistants, and more. To keep it simple and concrete, we focus on a website optimization example. In Chapter 12, we highlight the differences when running experiments for *thick clients*, such as native desktop and mobile apps.

Setting up the Example

Our concrete example is a fictional online commerce site that sells widgets. There are a wide range of changes we can test: introducing a new feature, a change to the user interface (UI), a back-end change, and so on.

In our example, the marketing department wants to increase sales by sending promotional emails that include a coupon code for discounts on the widgets. This change is a potential business model change, as the company has not previously offered coupons. However, an employee at the company recently read about Dr. Footcare losing significant revenue after adding a coupon code (Kohavi, Longbottom et al. 2009, section 2.1) and also read that *removing* coupon codes is a positive pattern on GoodUI.org (Linowski 2018). Given these external data, there is concern that adding the coupon code field to checkout will degrade revenue, even if there are no coupons, that is, just the

fact of users seeing this field will slow them down, and cause them to search for codes, or even abandon.

We want to evaluate the impact of simply adding a coupon code field. We can use a fake door or painted door approach (Lee 2013) – the analogy is that we build a fake door or paint it on a wall and see how many people try to open it. In this case, we implement the trivial change of adding a coupon code field to the checkout page. We do not implement a true coupon code system, as there are no codes available. Whatever the user enters, the system says: "Invalid Coupon Code." Our goal is simply to assess the impact on revenue by having this coupon code field and evaluate the concern that it will distract people from checking out. As this is a simple change, we will test two UI implementations. It is common to test several Treatments simultaneously to evaluate an idea versus an implementation. In this case, the idea is adding coupon code, while the implementation is a specific UI change.

This simple A/B test is a critical step in assessing the feasibility of the new business model.

When translating this proposed UI change into a hypothesis, it is useful to think about the online shopping process as a funnel, shown in Figure 2.1. A customer starts at the home page, browses through a few widgets, adds a widget to the cart, starts the purchase process, and finally completes a purchase. Of course, the idea of a funnel is simplistic; customers rarely complete the steps in a consistently linear fashion. There is a lot of back-and-forth swirl between states as well as repeat visitors who skip intermediate steps. However, this simple model is useful in thinking through experiment design and analysis, as experiments commonly target improving a particular step in the funnel (McClure 2007).

For our experiment, we are adding a coupon code field to the checkout page, and we are testing two different UIs, as shown in Figure 2.2, and would like to evaluate the impact (if any) on revenue. Our hypothesis is: "Adding a coupon code field to the checkout page will degrade revenue."

To measure the impact of the change, we need to define goal metrics, or success metrics. When we have just one, we can use that metric directly as our *OEC* (see Chapter 7). One obvious choice for this experiment might be revenue. Note that even though we want to increase overall revenue, we do not recommend using the sum of revenue itself, as it depends on the number of users in each variant. Even if the variants are allocated with equal traffic, the actual number of users may vary due to chance. We recommend that key metrics be normalized by the actual sample sizes, making *revenue-per-user* a good OEC.

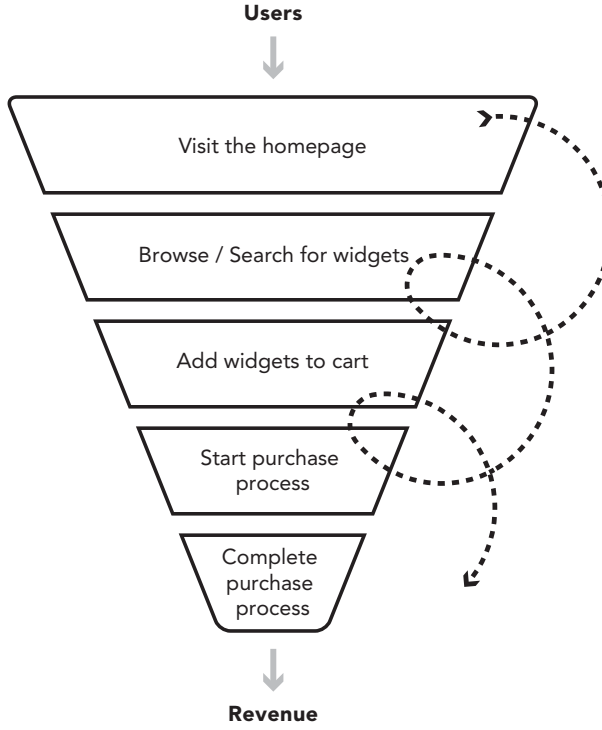

Figure 2.1 A user online shopping funnel. Users may not progress linearly through a funnel, but instead skip, repeat or go back-and-forth between steps

The next critical question is to determine which users to consider in the denominator of the revenue-per-user metric:

- **All users who visited the site.** This is valid; however, it is noisy because it includes users who never initiated checkout, where the change was made. We know that users who never initiated checkout could not be impacted by our change. Excluding these users will result in a more sensitive A/B test (see Chapter 20).
- **Only users who complete the purchase process.** This choice is incorrect, as it assumes that the change will impact the amount purchased, not the percentage of users who complete the purchase. If more users purchase, revenue-per-user may drop even though total revenue increases.
- **Only users who start the purchase process.** This is the best choice, given where the change is in the funnel. We include all potentially affected users, but no unaffected users (users who never start checking out) who dilute our results.

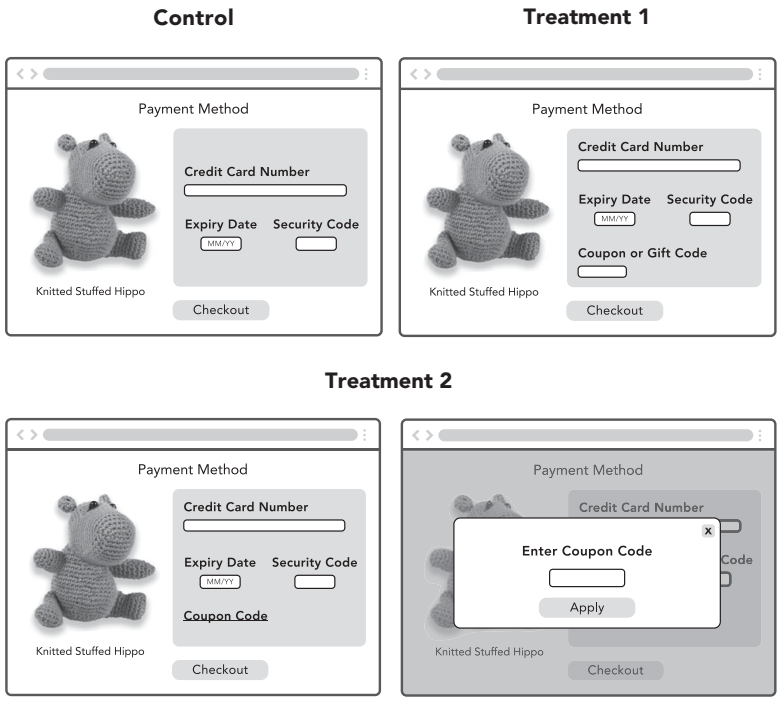

Figure 2.2 (1) Control: the old checkout page. (2) Treatment one: coupon or gift code field below credit card information (3) Treatment two: coupon or gift code as a popup

Our more refined hypothesis becomes "Adding a coupon code field to the checkout page will degrade revenue-per-user for users who start the purchase process."

Hypothesis Testing: Establishing Statistical Significance

Before we can design, run, or analyze our experiment, let us go over a few foundational concepts relating to statistical hypothesis testing.

First, we characterize the metric by understanding the baseline *mean* value and the *standard error* of the mean, in other words, how variable the estimate of our metric will be. We need to know the variability to properly size our experiment and calculate statistical significance during analysis. For most metrics we measure the mean, but we can also choose other summary statistics, such as percentiles. The sensitivity, or ability to detect statistically significant

differences, improves with lower standard errors of the mean. This can typically be achieved by allocating more traffic to the variants or running the experiment longer because the number of users typically grows over time. The latter, however, may not be as effective after the first couple of weeks as unique user growth is sub-linear due to repeat users while some metrics themselves have a "growing" variance over time (Kohavi et al. 2012).

When we run an experiment, instead of characterizing a metric for a single sample, we instead have multiple samples. Specifically, in controlled experiments, we have one sample for the *Control* and one sample for each *Treatment*. We quantitatively test whether the difference between a pair of Treatment and Control samples is unlikely, given the *Null hypothesis* that the means are the same. If it is unlikely, we reject the Null hypothesis and claim that the difference is statistically significant. Specifically, given revenue-per-user estimates from the Control and Treatment samples, we compute the *p-value* for the difference, which is the probability of observing such difference or more extreme assuming the Null hypothesis is true. We reject the Null hypothesis and conclude that our experiment has an effect (or the result is statistically significant) if the p-value is small enough. But what is small enough?

The scientific standard is to use a p-value less than 0.05, meaning that if there is truly no effect, we can correctly infer there is no effect 95 out of 100 times. Another way to examine whether the difference is statistically significant is by checking whether the *confidence interval* overlaps with zero. A 95% confidence interval is the range that covers the true difference 95% of the time, and for fairly large sample sizes it is usually centered around the observed delta between the Treatment and the Control with an extension of 1.96 standard errors on each side. Figure 2.3 shows the equivalence of the two views.

Statistical power is the probability of detecting a meaningful difference between the variants when there really is one (statistically, reject the null when there is a difference). Practically speaking, you want enough power in your experiment to be able to conclude with high probability whether your experiment has resulted in a change bigger than what you care about. Usually, we get more power when the sample size is larger. It is common practice to design experiments for 80–90% power. Chapter 17 further discusses the statistical details.

While "statistical significance" measures how likely the result you observe or more extreme could have happened by chance assuming the null, not all statistically significant results are practically meaningful. How big of a difference, in this case for revenue-per-user, actually matters to us from a business perspective? In other words, what change is *practically significant*?

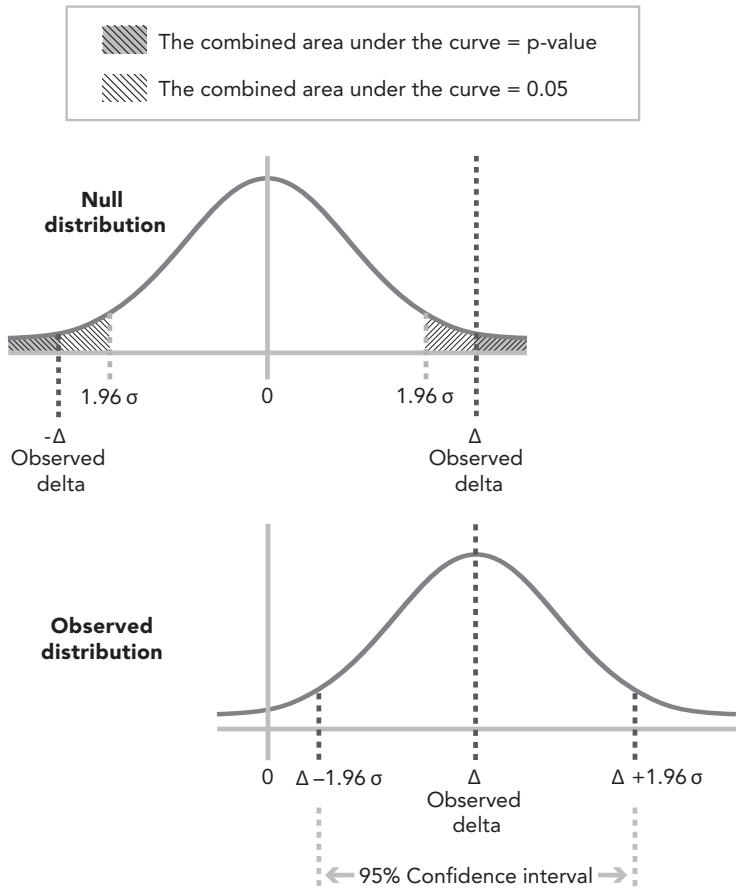

Figure 2.3 Top: Using p-value to assess whether the observed delta is statistically significant. If p-value is less than 0.05, we declare that the difference is statistically significant. Bottom: The equivalent view of using 95% confidence interval $[\Delta - 1.96\,\sigma, \Delta + 1.96\,\sigma]$ to assess statistical significance. If zero lies outside of the confidence interval, we declare significance

Establishing this substantive boundary is important for understanding whether the difference is worth the costs of making the change. If your website generates billions of dollars, like Google and Bing, then a 0.2% change is practically significant. In comparison, a startup may consider even a 2% change too small, because they are looking for changes that improve by 10% or more. For our example, let's state that from a business perspective, a 1% or larger increase in revenue-per-user is a change that matters or is practically significant.

Designing the Experiment

We are now ready to design our experiment. We have a hypothesis, a practical significance boundary, and we have characterized our metric. We will use this set of decisions to finalize the design:

1. What is the randomization unit?
2. What population of randomization units do we want to target?
3. How large (size) does our experiment need to be?
4. How long do we run the experiment?

For now, let's assume that *users* is our randomization unit. Chapter 14 discusses alternatives, but *users* is by far the most common choice.

Targeting a specific population means that you only want to run the experiment for users with a particular characteristic. For example, you are testing out new text but only have the new text in a few languages; in this case, you could only target users with their interface locale set to those languages. Other common targeting attributes include geographic region, platform, and device type. Our example assumes we are targeting all users.

The size of the experiment (for us, the number of users) has direct impact on the precision of the results. If you want to detect a small change or be more confident in the conclusion, run a larger experiment with more users. Here are some changes we might consider:

- If we use *purchase indicator* (i.e., did the user purchase yes/no, without regard to the purchase amount) instead of using *revenue-per-user* as our OEC, the standard error will be smaller, meaning that we will not need to expose the experiment to as many users to achieve the same sensitivity.
- If we increase our practical significance level, saying that we no longer care about detecting a 1% change, but only bigger changes, we could reduce the sample size because bigger changes are easier to detect.
- If we want to use a lower p-value threshold such as 0.01 to be more certain that a change occurred before we reject the Null hypothesis, we need to increase the sample size.

Here are a few other considerations when deciding experiment size:

- How safe is the experiment? For large changes where you are uncertain about how users might react, you may want to start with a smaller proportion of the users first. This rationale should not impact the choice of the final experiment size but may instead impact the ramp-up tactics (see Chapter 15 for more details).

- Does this experiment need to share traffic with other experiments, and if so, how do you balance traffic requirements? At a high level, if you have other changes to test, you can choose to either run those changes at the same time or sequentially. If you must divide traffic among several simultaneous tests, each test will end up with a smaller amount of traffic. In Chapter 4, we talk about running tests as a single layer or overlapping, and more importantly, how to build a proper infrastructure to scale all experiments.

Another big question is how long to run the experiment. Here are other factors to consider:

- **More users**: In the online experiments, because users trickle into experiments over time, the longer the experiment runs, the more users the experiment gets. This usually results in increased statistical power (exceptions happen if the metric being measured accumulates, e.g., number of sessions, and the variance also increases; see Chapter 18 for details). The user accumulation rate over time is also likely to be sub-linear given that the same user may return: if you have N users on day one, you will have fewer than 2N users after two days since some users visit on both days.
- **Day-of-week effect**: You may have a different population of users on weekends than weekdays. Even the same user may behave differently. It is important to ensure that your experiment captures the weekly cycle. We recommend running experiments for a minimum of one week.
- **Seasonality**: There can be other times when users behave differently that are important to consider, such as holidays. If you have a global user base, US as well as non-US holidays may have an effect. For example, selling gift cards may work well during the Christmas season but not as well during other times of the year. This is called *external validity*; the extent to which the results can be generalized, in this case to other periods of time.
- **Primacy and novelty effects**: There are experiments that tend to have a larger or smaller initial effect that takes time to stabilize. For example, users may try a new flashy button and discover it is not useful, so clicks on the button will decrease over time. On the other hand, features that require adoption take time to build an adopter base.

Our experiment design is now as follows:

1. The randomization unit is a user.
2. We will target all users and analyze those who visit the checkout page.
3. To have 80% power to detect at least a 1% change in revenue-per-user, we will conduct a power analysis to determine size.

4. This translates into running the experiment for a minimum of four days with a 34/33/33% split among Control/Treatment one/Treatment two. We will run the experiment for a full week to ensure that we understand the day-of-week effect, and potentially longer if we detect novelty or primacy effects.

In general, overpowering an experiment is fine and even recommended, as sometimes we need to examine segments (e.g., geographic region or platform) and to ensure that the experiment has sufficient power to detect changes on several key metrics. For example, we may have enough power to detect revenue impact across all users, but not enough power if we want to look at users in Canada only. Also note that while we have chosen approximately equal sizes for Control and Treatments, if the number of Treatments increases, you may consider increasing the size of the Control to be larger than that of Treatments (see Chapter 18 for more discussion).

Running the Experiment and Getting Data

Now let us run the experiment and gather the necessary data. Here we give you a brief overview of the pieces involved and provide more detail in *Scaling Experimentation: Digging into Variant Assignment* in Chapter 4.

To run an experiment, we need both:

- **Instrumentation** to get logs data on how users are interacting with your site and which experiments those interactions belong to (see Chapter 13).
- **Infrastructure** to be able to run an experiment, ranging from experiment configuration to variant assignment. See Chapter 4 *Experimentation Platform and Culture* for more detail.

Once you have run the experiment and gathered the logs data with the necessary instrumentation, it is time to process the data, compute the summary statistics, and visualize the results (see Chapter 4 and Chapter 16).

Interpreting the Results

We have data from our experiment! Before we look at the revenue-per-user results, let's run some sanity checks to make sure the experiment was run properly.

Table 2.1 *Results on revenue-per-user from the checkout experiment.*

	Revenue-per-user, Treatment	Revenue-per-user, Control	Difference	p-value	Confidence Interval
Treatment One vs. Control	$3.12	$3.21	–$0.09 (–2.8%)	0.0003	[–4.3%, –1.3%]
Treatment Two vs. Control	$2.96	$3.21	–$0.25 (–7.8%)	1.5e-23	[–9.3%, –6.3%]

There are many ways for bugs to creep in that would invalidate the experiment results. To catch them, we'll look at the *guardrail metrics* or *invariants*. These metrics should not change between the Control and Treatment. If they change, any measured differences are likely the result of other changes we made rather than the feature being tested.

There are two types of invariant metrics:

1. Trust-related guardrail metrics, such as expecting the Control and Treatment samples to be sized according to the configuration or that they have the same cache-hit rates.
2. Organizational guardrail metrics, such as latency, which are important to the organization and expected to be an invariant for many experiments. In the checkout experiment, it would be very surprising if latency changed.

If these sanity checks fail, there is probably a problem with the underlying experiment design, infrastructure, or data processing. See Chapter 21 for more information.

After running the sanity checks based on the guardrail metrics, we are ready to look at the results (Table 2.1).

Because the p-value for both Treatments is less than 0.05, we reject the Null hypothesis that Treatment and Control have the same mean.

So, what does this mean? Well, it means that we confirmed the pattern that adding a coupon code to the UI will decrease revenue. If we dig into the numbers further, the results indicate that the decrease is because fewer users complete the purchase process. Thus, any marketing email that sends out coupon codes needs to recoup not just the implementation cost of adding coupon processing and maintenance, but also the negative impact of adding the coupon code in the first place. Since the marketing model estimated a small revenue increase for the targeted users, but the A/B test shows a significant

revenue decrease to all users, the decision is made to scrap the idea of introducing promotion codes. A/B testing with a painted door saved us a large effort!

From Results to Decisions

The goal of running A/B tests is to gather data to drive decision making. A lot of work goes into ensuring that our results are repeatable and trustworthy so that we make the right decision. Let's walk through the decision-making process for a few different cases that could come up.

For each case, we have the results from the experiment, and our goal is to translate the results into a launch/no-launch decision. The reason to stress the decision-making part is because a decision needs to take into consideration both the conclusion from the measurement and the broader context, such as:

- Do you need to make tradeoffs between different metrics? For example, if user engagement goes up, but revenue goes down, should you launch? Another example is if CPU utilization increases, the cost of running your service may outweigh the benefit of the change.
- What is the cost of launching this change? This includes both the:
  - Cost to fully build out the feature before launch. Some features may have been fully built before experimenting. In those cases, the cost of going from 1% to 100% launch is zero. This is not always the case. As in our example, implementing the painted door was cheap, but the cost of implementing a full coupon system is expensive.
  - Cost for ongoing engineering maintenance after launch, since it may be more costly to maintain new code. New code tends to have more bugs and be less well tested for edge cases. If the new code introduces more complexity, it may also add friction and cost to build new changes on top of it.

    If the cost is high, you must ensure that the expected gain can cover it. In those situations, make sure that your practical significance boundary is high enough to reflect that. Conversely, if the cost is low or even zero, you may choose to launch any change that is positive, in other words, your practical significance boundary is low.
- What is the downside of making wrong decisions? Not all decisions are equal and not all mistakes are equal. There may be no downside of launching a change that has no impact, but the opportunity cost can be high if we forego a change that has impact, and vice versa. For example, you may be testing two possible headline offers on your site, and the offer itself will

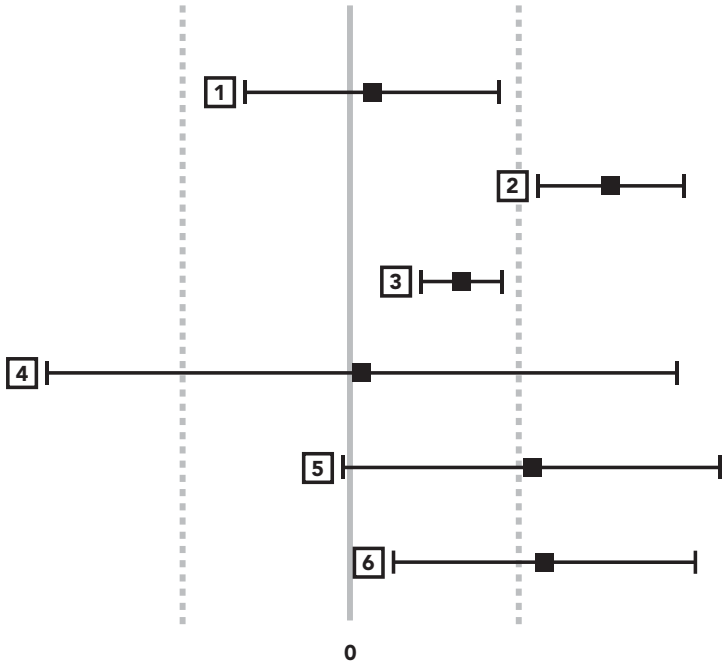

Figure 2.4 Examples for understanding statistical and practical significance when making launch decisions. The practical significance boundary is drawn as two dashed lines. The estimated difference for each example result is the black box, together with its confidence interval

only stay up for a few days. In that case, the downside of making the wrong decision is low because the change has a short lifespan. In this case, you may be willing to lower the bar for both statistical and practical significance.

You need to take these contexts into consideration as you construct your statistical and practical significance thresholds. These thresholds are critical as we move from the results of the experiment to a decision or action. Assuming we have updated the thresholds prior to the start of the experiment to reflect the broader context, let us walk through the examples in Figure 2.4 to illustrate how to use these thresholds to guide our decisions.

1. The result is not statistically significant. It is also clear that there is no practical significance. This leads to an easy conclusion that the change does not do much. You may either decide to iterate or abandon this idea.

2. The result is statistically and practically significant. Again, an easy decision: launch!
3. The result is statistically significant but not practically significant. In this case, you are confident about the magnitude of change, but that magnitude may not be sufficient to outweigh other factors such as cost. This change may not be worth launching.
4. Consider this example neutral, like our first example; however, the confidence intervals are outside of what is practically significant. If you run an experiment and find out it could either increase or decrease revenue by 10%, would you really accept that experiment and say that change is neutral? It's better to say you do not have enough power to draw a strong conclusion, and it is also such that we do not have enough data to make any launch decision. For this result, we recommend running a follow-up test with more units, providing greater statistical power.
5. The result is likely practically significant but not statistically significant. So even though your best guess is that this change has an impact you care about, there is also a good chance that there is no impact at all. From a measurement perspective, the best recommendation would be to repeat this test but with greater power to gain more precision in the result.
6. The result is statistically significant, and likely practically significant. Like 5, it is possible that the change is not practically significant. Thus here, like the prior example, we suggest repeating the test with more power. From a launch/no-launch decision, however, choosing to launch is a reasonable decision.

The key thing to remember is that there will be times you might have to decide even though there may not be clear answer from the results. In those situations, you need to be explicit about what factors you are considering, especially how they would translate into practical and statistical significance boundaries. This will serve as the basis for future decisions versus simply a local decision.