A/B TESTING CHEAT SHEET

EMMA ding

EMMA ding

emmading.com
info@datainterviewpro.com

A/B Testing Cheat Sheet

This comprehensive guide serves as a quick reference for various concepts, steps, and techniques in A/B tests, which include:

1. **Selecting metrics for experimentation**
2. **Selecting randomization units**
3. **Choosing a target population**
4. **Computing sample size**
5. **Determine test duration**
6. **Analyzing results**
7. **Alternatives to A/B tests**

Table of Contents

▼ Selecting Metrics for Experimentation

💡 **Experiment Metrics Criteria**

- **Measurable** within the experiment timeframe.
- **Attributable** to the change in the product/feature.
- **Sensitive** enough to detect changes that matter in a **timely** fashion.

▼ Success Metrics (goal metrics, true north metrics)

- A single or a very small set of metrics that capture the ultimate success you are striving towards
- Ensure success metrics are **simple** and **stable**
  - **Simple**: easily understood and broadly accepted by stakeholders
  - **Stable**: should not be necessary to update goal metrics every time you launch a new feature

▼ Driver Metrics (signpost metrics, surrogate metrics, indirect or predictive metrics)

- Shorter-term, faster-moving, and more sensitive than goal metrics
- Reflects hypotheses on the drivers of success and indicates we are moving in the right direction to move the goal metrics
- Actionable
- Resistant to gaming
- How to generate driver metrics:
  - Business goals: growth, engagement, revenue
  - HEART: happiness, engagement, adoption, retention, and task Success
  - AARRR: acquisition, activation, retention, referral, and revenue
  - User funnel

💡 Minimize to 5 key metrics (success and driver metrics) as a rough rule of thumb. When dealing with a lot of metrics, OEC (Overall Evaluation Criterion), a combination of multiple key metrics, can be used. Devising an OEC makes the tradeoffs explicit and makes the exact definition of success clear. The OEC can be the weighted sum of normalized metrics (each normalized to a predefined range, say 0-1).

▼ Guardrail Metrics (counter metrics)

- Organizational Guardrails
  - Ensures we move towards success with the right balance and without violating important constraints

    - E.g. Website/App performance, latency: wait times for pages to load, error logs: number of error messages, client crashes: number of crashes per user, business goals, revenue: revenue per user and total revenue, and engagement (e.g. time spent per user, DAU, and page views per user)
- Trust-related guardrails
    - Assess the trustworthiness and internal validity of experiment results
    - E.g. the Sample Ratio Mismatch (SRM) guardrail and cache hit ratio to be the same among Control and Treatment.

Selecting Randomization Units

▼ General Considerations

1. **Consistent user experience** (video)
    - For changes visible to users, we should use a user ID or a cookie as the randomization unit.
    - For changes invisible to users, e.g., change in latency, it depends on what we want to measure. A user ID or a cookie are still good options if we want to see what happens over time.
2. **Variability** (video)
    - If the randomization unit is the same as the unit of analysis, the empirically computed variability is similar to the analytically computed variability.
    - If the randomization unit is coarser than the unit of analysis, e.g., the randomization unit is the user and we wish to analyze the click-through rate (the unit of analysis is a page view), the variability of the metric will be much higher. This is because the independence assumption is invalid as we are dividing groups of correlated units, which increases the variability.
    - Example (video), the graph mentioned in the example is Figure 4 in this paper.
3. **Ethical considerations** (video)
    - May face security and confidentiality issues when using identifiable randomization units.

> 🔑 User-level randomization is the most common in practice because:
>
> - It ensures a **consistent user experience**.
> - It allows for **long-term measurements**, such as user retention and users' learning effect.

▼ Different Choices of Randomization Units

▼ **Account-Based (User-Based):** Every single account is a randomization unit. Users are identifiable via signing to the website or via cookies.

- Pros: Stable across time and platforms

- Cons: Identifiable

▼ **Cookie-Based:** A pseudonymous user ID, specific to a browser and a device.

- Pros: Anonymous.
- Cons: Users can clear cookies. Not persistent across platforms, changes when users switch browser or device platforms.

▼ **Session-Based (or Page View-Based):** Every user session is a randomization unit. A session starts when a user logs in and ends when a user logs out or after 30 min of inactivity.

- Pros: Finer level of granularity creates more units, and the test will have more power to detect smaller changes.
- Cons: May lead to inconsistent user experience, so it's appropriate when changes are not visible to the user

▼ **IP-Based:** Every IP address is a randomization unit. Every device in every network is assigned a unique IP.

- Pros: Maybe the only option for certain experiments, e.g., testing latency using one hosting service versus another
- Cons: Changes when users change places, creating an inconsistent experience. Many users may share the same IP address. Therefore, not recommended unless it's the only option.

▼ **Device-Based:** Every device ID is a randomization unit.

- Pros: Immutable Id associated with a specific device.
- Cons: Identifiable. Only available for mobile devices

▼ Randomization Unit vs. Unit of Analysis

The general recommendation for the randomization unit is the **same as (or coarser than)** the unit of analysis.

- **It works if the randomization unit is coarser than the unit of analysis.**
  - e.g., the randomization unit is the user and we analyze the click-through rate (the unit of analysis is a page view).
  - The caveat is that in this case, we need to pay attention to the variability of the unit of analysis as explained earlier.
- **It does not work if the randomization unit is finer than the unit of analysis.**
  - e.g., the randomization unit is a page view and we analyze user-level metrics.
  - This is because the user's experience is likely to include a mix of variants (i.e., some in Control and some in Treatment), and computing user-level metrics will not be meaningful.

▼ Choosing a Target Population

- Do we want to target a specific population / all the users?
- Consider geographic region, platform (mobile vs tablet vs laptop), device type, user demographics (age, gender, country, etc), usage or engagement level (analyze the user journey, example), etc
  - E.g. A new feature only available for users in a particular geographic region → Only need to select users in that region

> 💡 Be careful if you select users based on usage and your treatment affects usage. This violates the stable unit treatment value assumption.

▼ Computing Sample Size

Two-sampled t-tests are the most common statistical significance tests used. Suppose Y is a metric of interest.

- H_0: mean(Y^t) = mean(Y^c) (no treatment effect)
- H_α: mean(Y^t) $\neq$ mean(Y^c) (there is a treatment effect)

The required sample size depends on 4 things:

▼ Significance level: α (common choice is .05):

- α = Type I Error = when the null hypothesis is actually true, rejecting the null hypothesis = incorrectly rejecting the null hypothesis = False Positive
- **Significance level:** The probability that we reject H_0 even when the treatment has no effect. The probability of committing a Type I error (α).

▼ Statistical power: $1 - P(\beta)$ (common choice is .8):

- β = Type II Error = when the alternative hypothesis is true, failing to reject the null hypothesis = incorrectly accepting the null hypothesis = False Negative
- **Statistical power**: The probability that we reject H_0 when the treatment indeed has an effect. This measures how sensitive the experiment is. If power is too low, we can't detect true effects; if it's unrealistically high (.99), we may never finish the experiment.

▼ Variances:

- Because samples are independent, $Var(\Delta) = Var(\bar{Y}^t) + Var(\bar{Y}^c)$ where Δ is the difference between the Treatment average and the Control average. Variances are often estimated either from historical data or from A/A tests.

▼ Minimally detectable effect (MDE) δ (a.k.a. practical significance):

- The smallest difference that matters in practice

📖 **Sample Size Formula**

$$n = \frac{(\sigma_t^2+\sigma_c^2)(z_{1-\alpha/2}+z_{1-\beta})^2}{\delta^2}$$

Typically, we choose $\alpha = 0.05$, then $z_{1-\alpha/2} = z_{.975} \approx 1.96$; and $\beta = 0.2$, then $z_{1-\beta} = z_{.80} \approx 0.84$. Assuming Treatment and Control are of equal size, the required sample size for each variant is about:

$$n = \frac{16\sigma^2}{\delta^2}$$

σ^2: Sample variance of the difference between the Treatment and the Control. Estimated from historical data. (For ratio metrics, the maximum variance is 0.25.)

δ: Practical significance (Minimum detectable effect), determined among multiple stakeholders.

▼ Determine Test Duration

📖 **Test Duration**

$$\frac{\text{Sample Size}}{\text{Randomization Units/Day}}$$

Pitfalls:

- Avoid a duration of less than a week.
- A longer test gathers more data and is almost always better.

Ramping: trade-off among **speed, quality, risk (SQR)**

- **mitigate risk** (0-5%): Start with team members, company employees, loyal users, etc. in fear of bugs or other risks — these people tend to be more forgiving.
- **maximum power ramp** (MPR, 5-50%): Measure treatment effect.
- **post-MPR** (optional): Ensure infra can withstand the change.
- **long-term holdout** (optional): Be aware of opportunity costs and ethics because those users won't enjoy new features for a while.

Analyzing Results

▼ Sanity Checks

▼ Guardrail Metrics

▼ **Trust-related** guardrail metrics help us ensure our assumptions regarding the data are not violated.

- Sample Ratio Mismatch (SRM). For the study population, we want 50% in the treatment and 50% in the control. If our study population was 1,000 with 800 in the treatment and 200 in the control, obviously something is wrong. We will have to perform a hypothesis test if this is not something we can easily judge.
- Cache hit ratio to be the same among Control and Treatment.
- Test statistics follow the assumed distribution
  - When the sample size is big enough, by the central limit theorem (CLT), the sampling distribution of $\mu_t - \mu_c$ should be normally distributed.
  - Normality test

▼ **Organizational-related** guardrail metrics are used to ensure that the performance of the organization is following the standard we expect.

- Website/App performance
- Latency: wait times for pages to load.
- Error Logs: number of error messages.
- Client Crashes: crashes per user.
- Business goals
- Revenue: revenue per user and total revenue.
- Engagement: e.g., time spent per user, daily active users (DAU), and page views per user.

▼ How To Do Sanity Checks

- **A/A test:** Sanity check of the A/B testing system. Run **before** the system is used in the application
- **Z-test or T-test**: Both tests can be used to compare proportions or group means and test for significant differences between them.
  - Example: checking sample sizes between groups using Z-test (video)
    - **Note:** In step 1, the "standard deviation" is the standard deviation of the sampling distribution for the proportion, or standard error (SE). SE should be used in computations instead of SD.
- **Chi-Squared Test**
  - Example: checking for the SRM. Using the Chi-squared test as a goodness of fit test (Fairness of Die in the Wiki page), it is analogous to testing if the treatment/control assignment mechanism is a fair game (should be 50/50).

▼ What If Sanity Checks Fail?

- Stop and assess. Ask what went wrong and how we can address it.
- These failures should be a priority concern before moving on to analyzing the data. Is this just a one-time issue or if it will persist or become worse over time? These are supposed to be invariant metrics; we do not want these to differ between groups.

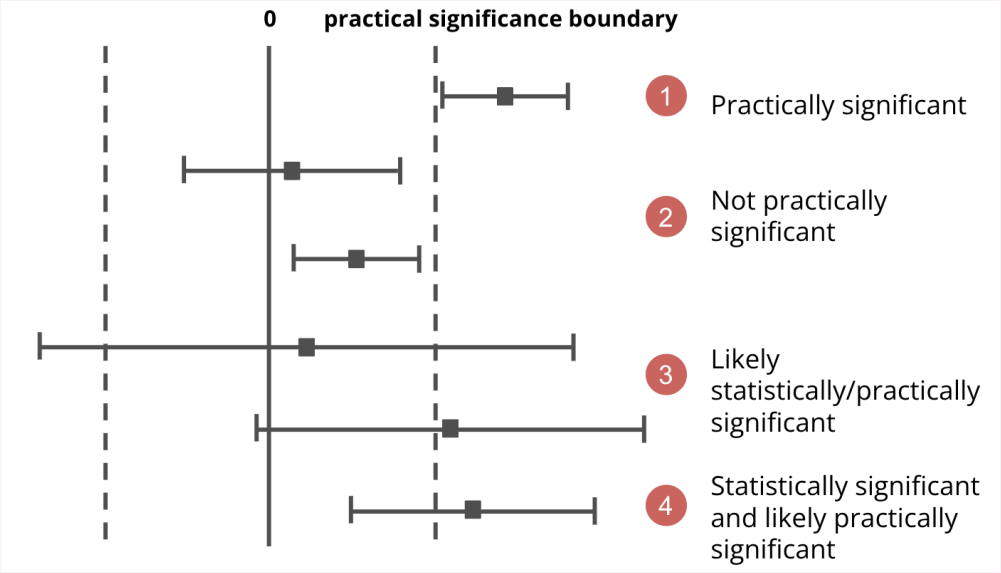
- We can also rerun the experiment.
- How to debug SRM (video)

▼ Hypothesis Tests

- If test statistics follow or can be approximated by normal or t-distributions, use the Z-test or t-test.
  - Z-test or t-test
    - Decide one-tailed or two-tailed tests
    - Compute the mean
    - Compute either pooled or unpooled variance
  - P-value
    - Definition: If H_0 is true, what's the probability of seeing an outcome (e.g., a t-statistic) *at least* this extreme?
    - How to use: If the p-value is below your threshold of significance (typically 0.05), then you can reject the null hypothesis.
    - Misconception: It's **not** the probability of H_a being true.
  - Assumptions
    - **Normality**: When the sample size is big enough, by the central limit theorem (CLT), the sampling distribution of the difference in the means between the two groups should be normally distributed.
      - If the sample isn't large enough for the sampling distribution to be normal
        - **solution #1**: cap values if data is highly skewed
        - **solution #2**: use bootstrapping to calculate statistics
    - **Independence**: Each observation of the dependent variable is independent of other observations.
- Otherwise, use non-parametric tests.

▼ Statistical and Practical Significance

📌 The graph shows 4 patterns of A/B testing results in terms of statistical and practical significance.



Source: Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing

1. Statistically and practically significant: The result is both statistically ($p < .05$ and 95% CI does not contain 0) and practically significant, so we should obviously launch it. → Launch!

2. Not practically significant:

- **Scenario 1:** The change is neither statistically (95% CI contains 0) nor practically significant (95% CI sits in the middle), so not worth launching. → The change does not do much. Either decide to iterate or abandon this idea.
- **Scenario 2:** Statistically significant (95% CI doesn't contain 0) but not practically significant → if implementing a new algorithm is costly, then it's probably not worth launching; if the cost is low, then it doesn't hurt to launch.

3. Likely statistically/practically significant:

- **Scenario 1:** The 95% CI contains 0 and the CI is outside of what is practically significant. → There is not enough power to draw a strong conclusion and we do not have enough data to make any launch decision. Run a follow-up test with more units, providing greater statistical power.
- **Scenario 2:** Likely practically significant. Even though our best guess (i.e., point estimate) is larger than the practical significance boundary, it's also possible that there is no impact at all. → Repeat this test but with greater power to gain more precision in the result.

Both scenarios suggest our experiment may be underpowered, so we should probably run new experiments with more units if time and resources allow.

4. Statistically significant and likely practically significant. It is possible that the change is not practically significant. → Can repeat the test with more power. However, choosing to launch is a reasonable decision.

▼ Common Problems and Pitfalls

▼ Multiple Testing Problems arise in the two scenarios below.

1. Multiple success metrics (Multiple hypotheses): When the significance level (false positive probability) is 5% for each metric. With N metrics, Pr(at least one metric is false positive) = $1-(1-0.05)^N$ is much greater than 5%.
    - Group metrics into *expected to change*, *not sure*, and *not expected to change*.
    - Set different significance levels for various groups.
2. Post-experiment result segmentation: Multiple hypotheses are squeezed into one experiment. Also a higher chance of false positive results. The overall result can contradict segmented results (Simpson's Paradox).
    - Avoid post-test result segmentation.
    - If post-test result segmentation is desired:
        - Ensure enough randomization units in each segment
        - Ensure sufficient randomization in each segment

▼ Lack of Testing Power

- Causes
    - P-hacking: Stop the experiment earlier than the designed duration when observing the p-value is lower than the threshold value.
    - The experiment ran as designed but there are not enough randomization units.
    - High variance
- Solutions
    - Do not stop the experiment before the design's duration.
    - If there are not enough randomization units
        - If the experiment is still running, we should run the experiment until enough units are collected.
        - If not, we should re-run the experiment
    - Clean data to reduce variance: remove outliers (e.g., capping), log transformation (don't log transform revenue!)
    - Use trigger analysis, i.e., only include impacted units (e.g. conversion rate may be 0.5% when you include users from the top funnel but it may be 50% right before the change). The caveat is when generalizing to all users, true effect could be anywhere between 0 and the observed effect.

▼ **Changes in Users' Behaviors**

- Causes
  - Novelty and primacy effect
  - Seasonality
  - Market change
- Solutions
  - Long-term monitoring

▼ **Network Effects**

- Use isolation methods. Ensure little or no spillover between the control and treatment units
- Cluster-based randomization
  - Randomize based on groups of people who are more likely to interact with fellow group members, rather than outsiders
- Geo-based randomization
  - Limit control/treatment group to a specific location or a city
- Time-based randomization
  - Select a random time and place all users in either control or treatment groups for a short period of time.
- How to detect interference:
  - Monitor during the experiment
  - Long-term monitoring by allowing an experiment to run for at least 3 months or by having a holdback group, namely a small control group that is never given access to a new feature.

▼ Alternatives to A/B tests

▼ **Qualitative Analysis**

- Conduct user experience research: Great for generating hypotheses. Terrible for scaling.
- Focus groups: A bit more scalable but users may fall into groupthink
- Surveys: Responders may not be faithful or representative.
- Human evaluation: Having human raters rate results or label data is useful for debugging, but they may differ from actual users.

▼ **Quantitative Analysis**

- Conduct retrospective analysis by analyzing users' activity logs: Use historical data to understand baselines, metric distributions, form hypotheses, etc.
- Causal inference: interrupted time series (same group go through control and treatment over time), interleaved experiments (results by two rankers are de-duped and mixed together), regression

discontinuity design (compare outcomes for winners vs. near winners — supposedly similar, but end up in different conditions), propensity score matching (units are not randomly assigned — find similar units in different groups to compare), difference in differences (initial values differ — compare changes)

- Requires making many assumptions and incorrect assumptions can lead to a lack of validity.
- Requires a great deal of care to generate reliable results.