

# 인과추론과 실무 : RCT와 실험플랫폼

가짜연구소 인과추론팀

발표자 : LG CNS 김성수

# 인과 추론 Overview Recap

책 67p. “무작위 배정으로 독립성을 확보하기”라는 제목을 설정한 이유

처치 배정 매커니즘 : 어떤 단위가 처치를 받을지, 어떤 단위가 대조군이 될지를 결정하는 과정

- 무시할 수 있는 할당 메커니즘(**Ignorable** Assignment Mechanism) : 처치가 잠재적 결과(Potential Outcome)와 독립
- 혼동되지 않는 할당 메커니즘(**Unconfounded** Assignment Mechanism) : 처치가 잠재적 결과(Potential Outcome)와 독립
- 처치와 결과 사이의 독립성을 이야기하는 것이 아닌 처치와 잠재적 결과가 독립
- 무작위 배정을 수행하면 처치와 잠재적 결과와 독립 -> 처치와 잠재적 결과의 두 집단이 비교 가능(**Exchangeability**)

$$y^0, y^1 \perp T | X.$$

# 인과 추론 Overview Recap

67p. “무작위 배정으로 독립성을 확보하기”라는 제목을 설정한 이유

처치 배정 매커니즘 : 어떤 단위가 처치를 받을지, 어떤 단위가 대조군이 될지를 결정하는 과정

- 무시할 수 있는 할당 메커니즘(**Ignorable Assignment Mechanism**) : 처치가 잠재적 결과(Potential Outcome)와 독립
- 혼동되지 않는 할당 메커니즘(**Unconfounded Assignment Mechanism**) : 처치가 잠재적 결과(Potential Outcome)와 독립
- 처치와 결과 사이의 독립성을 이야기하는 것이 아닌 처치와 잠재적 결과가 독립
- 무작위 배정을 수행하면 처치와 잠재적 결과와 독립 -> 처치와 잠재적 결과의 두 집단이 비교 가능(**Exchangeability**)

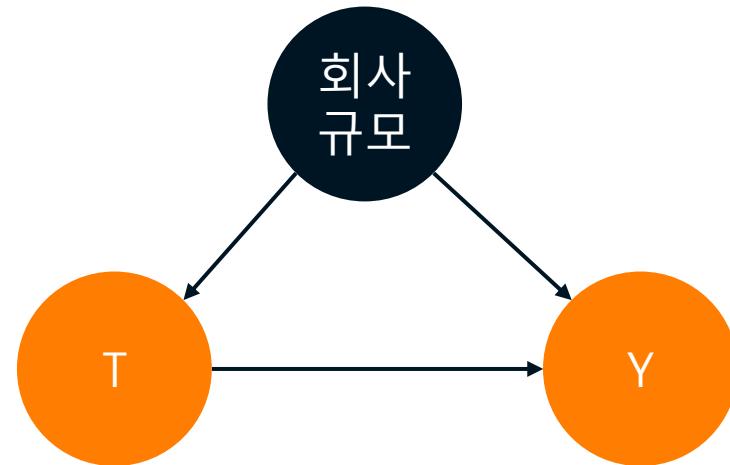
$$y^0, y^1 \perp T | X.$$

-> 처치의 랜덤을 통해 잠재적 결과가 처치로 인해 미리 예측되거나 영향을 받지 않도록 함으로써, 두 집단간 시스템적인 차이가 없이 **비교 가능함**

-> Week2에서 A/B테스트를 할 수 없을 때, 처치 배정 매커니즘을 밝히라고 한 이유

# 인과 추론 Overview Recap

Unignorable Assignment / Unconfounded Assignment Mechanism - 이미 배운 사례로



Unignorable = Unconfounded =  
Selection bias .....

# 인과 추론 Overview Recap

Unignorable Assignment / Unconfounded Assignment Mechanism – Rubin (**Perfect Doctor 사례**)

의사가 수술(1)이나 약물(0)을 처방

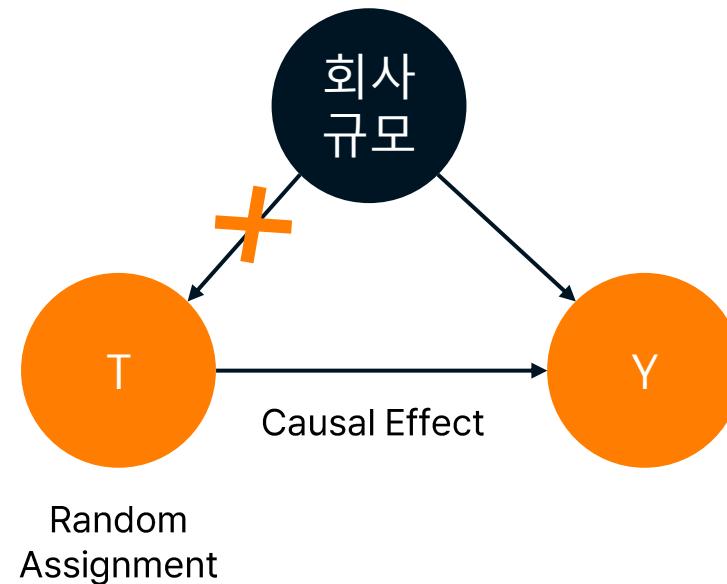
| unit              | $Y_i(0)$ | $Y_i(1)$ | $Y_i(1)-Y_i(0)$ |
|-------------------|----------|----------|-----------------|
| patient #1        | 1        | 7        | 6               |
| <b>patient #2</b> | <b>6</b> | <b>5</b> | <b>-1</b>       |
| patient #3        | 1        | 5        | 4               |
| <b>patient #4</b> | <b>8</b> | <b>7</b> | <b>-1</b>       |
| Average           | 4        | 6        | 2               |

| unit         | $T_i$ | $Y_{iobs}$ |
|--------------|-------|------------|
| patient #1   | 1     | 7          |
| patient #2   | 0     | 6          |
| patient #3   | 1     | 5          |
| patient #4   | 0     | 8          |
| Average Drug |       | 7          |
| Average Surg |       | 6          |

- 의사는 환자의 잠재적 결과에 대해 알고 있기 때문에 각 환자에게 가장 효과적인 방법을 할당 (**고객의 특성에 따라 처치 –처치가 무작위가 아님**)
- 환자 #2와 #4는 처치를 받지 않았을 때 기대할 수 있는 결과가 이미 좋았기 때문에 약물 처치를 받음
- 치료 할당 확률이 0과 1 사이(**positivity**) = 특정 조건(환자 상태)에 따라서 할당이 이루어졌기 때문에 **처치 확률이 정확히 0이나 1**
- 처치 할당 매커니즘이 **잠재적인 결과에 의존**하므로 무시할 수 없으며, 혼란 존재!
- 이 상황에서 관찰된 평균 차이(Average Drug – Average Surg)는 인과 효과인가?

## 무작위 실험(Randomized experiment)

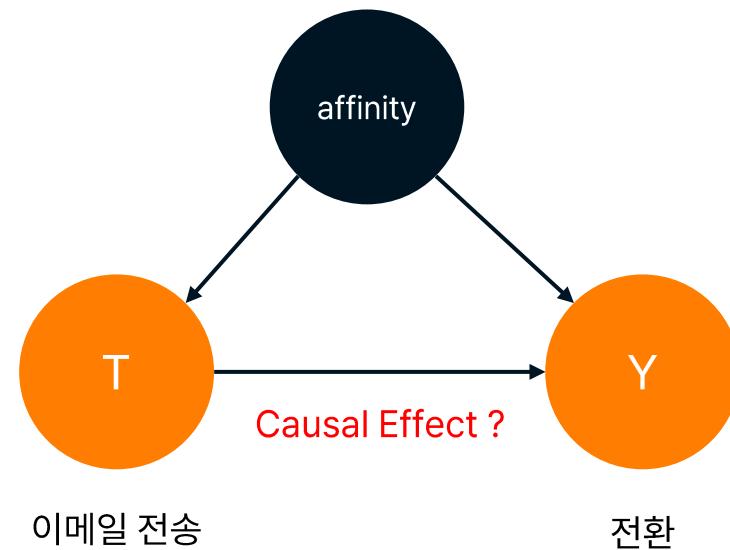
- 무작위 배정을 수행하면 처치와 잠재적 결과와 독립 -> 처치와 잠재적 결과의 두 집단이 비교 가능 (**Exchangeability**)
- 처치 배정 매커니즘이 무작위 = 처치효과를 식별하는 적합한 조건
- 실험군과 대조군의 잠재적 결과의 기댓값은 **처치의 효과를 제외한다면 똑같음**
- **두 그룹의 잠재적 효과의 결과 차이는 처치에 의한 것**



# 무작위 실험(Randomized experiment)

**인과 문제: 교차 판매 이메일이 멤버십 전환에 얼마나 효과적인가?**

- 마케팅팀은 우수 고객에게 이메일을 전송하여, 이메일 전환의 효과를 확인
  - 이메일을 받은 고객은 이메일을 받지 않았더라도, 다른 고객보다 더 많이 전환
  - 처치 할당은 무시할 수 없고, 혼돈이 있으며 비교 불가능함. 심지어 우수 고객이 아닌 사람은 처치 할당의 확률이 0

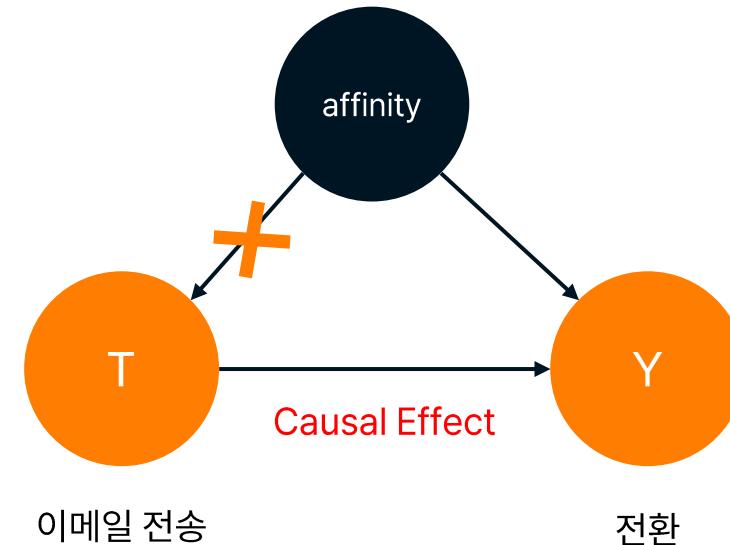


$$E[ \text{Conversion}_0 \mid \text{Email} = 1 ] > E[ \text{Conversion}_0 \mid \text{Email} = 0 ].$$

# 무작위 실험(Randomized experiment)

인과 문제: 교차 판매 이메일이 멤버십 전환에 얼마나 효과적인가?

- 무작위 실험을 수행하여 **이메일을 랜덤하게 전송**
- Affinity와 같은 고객의 특성이 처치 할당에 영향을 주지 못함
- 처치 할당은 무시할 수 있고, 혼돈이 없으며 비교 가능함. 처치 할당의 확률이 0과 1사이



$$E[\text{Conversion} \mid \text{Email} = 1] > E[\text{Conversion} \mid \text{Email} = 0]$$

# 무작위 실험(Randomized experiment)

Stable Unit Treatment Value Assumption (SUTVA)

- **Consistency:** An unverifiable assumption requiring a subject's potential outcome under the observed exposure value is indeed their observed outcome

Causal effect = **Potential outcome with treatment** – Potential outcome without treatment

= **Observed outcome with treatment** – Potential outcome without treatment

Obesity(x) -> Health Status(y)

-> Obesity는 서로 다른 원인에 의해 발생한 Treatment (e.g., 같은 비만이지만 수준에 따라 단계가 다름 -> 서로 다른 Observed outcome)

-> **Potential outcome**을 명확하게 정의할 수 있는 **Treatment**를 디자인하는 것이 중요

- **No interference :** Potential outcomes for any given individual do not depend on the exposure status of another individual  
예: 대조군에 배정받아 쿠폰을 받지 못함 -> 배아파서 나의 앱 활용 패턴에 영향을 미치게 된다면 interference가 발생  
**실험군과 대조군 내의 사용자들을 고립(isolate)시키는 것이 중요 -> Interference**

# 무작위 실험(Randomized experiment)

## 표본의 수

A/B 테스트 상황에서 전통적으로 표본크기가  $N > 300$ 이면, 표본의 합이나 평균의 분포가 정규분포일까?

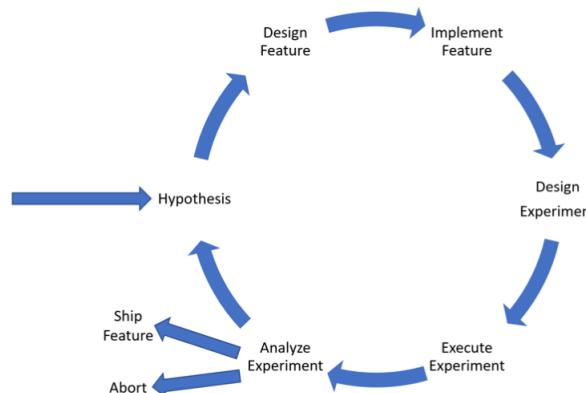
- (데이터의 특징) 데이터의 왜도나 첨도가 높은 경우, 표준 정규 분포로의 근사가 완벽하지 않을 수 있음
  - 데이터 비대칭 상황에서 엄밀한 표본크기 결정과 정규 분포 근사를 위해  $n > 100 s^2$ 를 경험적인 규칙으로 제안 ( $s$ 는 왜도)
- Kohavi et al. (2014)는 유저당 수익(revenue per user) 지표가 17.9의 왜도를 가져, 30,000개의 표본 크기가 필요하다 주장
  - **Capping**  $f(x)=\min(x,c)$  으로 왜도를 5로 줄이면, 2,500개의 표본 크기가 필요
- 베르누이 분포와 같이 0과 1로 데이터가 구성된 경우도 왜도가 높을 수 있음 -> Capping 불가
  - 라플라스 스무딩으로 인위적으로 시행(0과 1)을 추가하여 새롭게 평균을 계산
  - 불균형이 심한 데이터에 대해 보다 정확한 신뢰 구간 제공

$$\tilde{p} = \frac{n_1 + 1}{n + 2}$$

## 2. 무작위 실험의 난제 : 온라인 통제실험을 중심으로

## 온라인 통제 실험(Online Controlled experiment) –A/B 테스트

- 온라인 통제 실험은 소프트웨어 시스템 향상을 위한 표준으로 소프트웨어의 새로운 기능을 평가하기 위해 실험
  - (0) 가설 및 실험 디자인 – Sample Size(how long?), OEC, Guardrail Metric, Data Quality etc..
  - (1) 실험 할당 – 고유 식별자에 기반하여, 사용자를 그룹으로 나누고, 제품 변형(variant) 하나로 할당
  - (2) 실험 실행 – 사용자의 행동 데이터를 Logging
  - (3) 실험 로그 처리 - Log 데이터를 저장소로 업로드하여 처리 (e.g., 서버 로그와 실험 metadata 결합)
  - (4) 실험 분석 - variant로 영향을 받은 사용자로 필터링 처리된 로그를 분석하는 단계
  - (5) Ship or Abort – 실험한 Feature를 반영? or 실험 Reproduce? or Abort ?
- 설계 및 해석을 통해 발생하는 품질 문제나 편향으로 인해 Product에 해를 끼칠 수 있는 결론으로 이어질 수 있음
- 간과된 세부 사항과 예상치 못한 문제로 인해 무결성이 훼손  
-> 신뢰할 수 있는 실험의 중요성



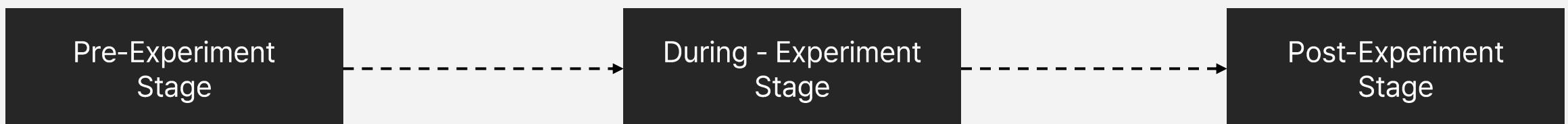
**Figure 1.** The experimentation lifecycle.

## 온라인 통제 실험(Online Controlled experiment)

- 소프트웨어 아키텍처 + 도메인 지식 + 실험 무결성 위한 통계 기법 + 데이터 엔지니어링
- 인과추론을 위한 통계와 함께 실험을 위한 아키텍처 이해가 필요

### Experiment Platform

(1) Random Assignment (2) Release (3) Logging for Experiments



- (1) Hypothesis and Select Metric of Interest
- (2) Imbalance minimization
  - A/A test
- (3) Sample Mismatch
- (4) Randomized & Analysis Unit
- (5) Sample Size
- (6) Triggering

- (1) Monitoring and Alert
  - Circuit Breaker
- (2) Segment – Date, User
  - Novelty & Seasonal & Primary Effect

- (1) Shipping
  - OEC vs Guardrail
- (2) Report
  - Confidence Intervals
  - Treatment Effect
  - P-values

# Engineering Challenges: 소프트웨어 아키텍처

실험의 핵심 요소는 적절한 소프트웨어 아키텍처

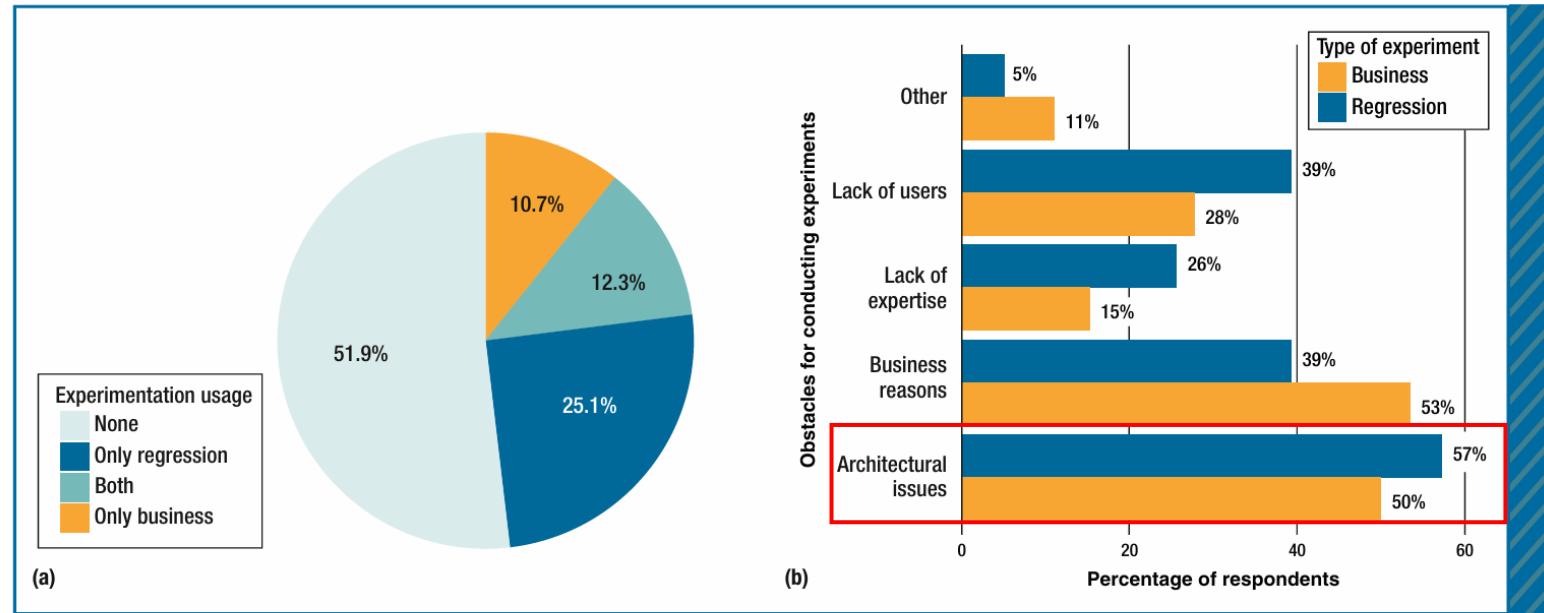
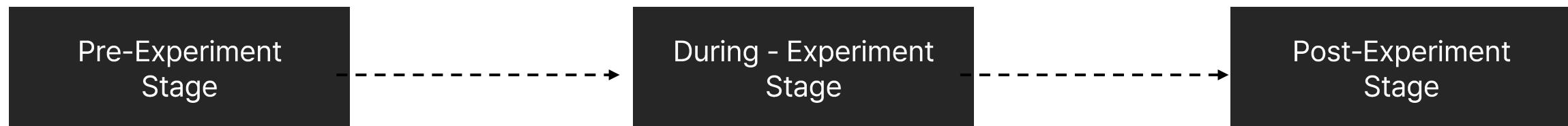


FIGURE 1. Some survey results. (a) Usage of experimentation practices. (b) Obstacles to conducting experiments.

# Engineering Challenges: 소프트웨어 아키텍처

## 실험의 핵심 요소는 적절한 소프트웨어 아키텍처

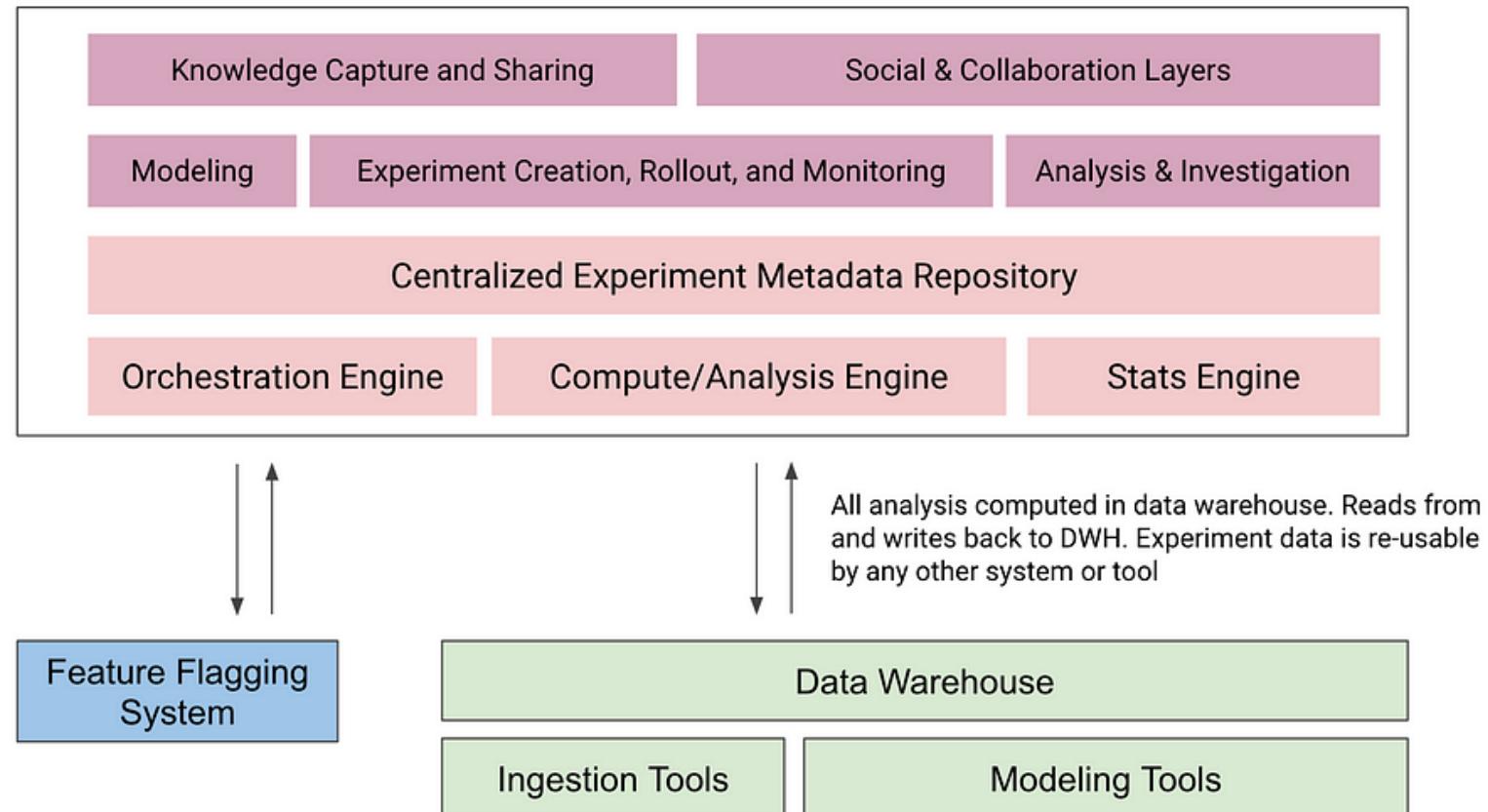


- |                                   |   |  |
|-----------------------------------|---|--|
| 1. 사용자군 설정이 무작위로 이루어지지 않음<br>>최소화 | 2. 사용자 로깅의 일부가 누락되거나 오류 발생<br>3. 여러 팀에서 수행하는 실험에 상호 간섭 발생<br>4. 실험이 사용자 지표에 악영향을 미칠 수 있음<br>>간섭 및 실험에 대한 모니터링 & 셧다운 | 5. 의사결정의 기준 지표 정의가 되지 않음<br>6. 분석 결과에 대한 유의성 정의가 되지 않음<br>7. 실험을 너무 일찍 중단하고 의사 결정을 내림<br>>실험 결과에 대한 대시보드 |
|-----------------------------------|---|--|

- End-to-End System  
전체 실험 워크플로우를 위한 통합 플랫폼  
Release, Metrics 계산, Analysis, Logging 등 실험 전반적인 워크플로우에 대한 자동화
- 실험 생성 및 모니터링 서비스  
PO 및 디자이너가 실험을 직접 생성 및 모니터링 할 수 있는 관리자 페이지, 실험 결과에 대한 대시보드
- DevOps  
플랫폼 및 실험에 대한 이상에 대한 지표 모니터링, Shutdown 및 경고 시스템, 디버깅, 롤백, Release

# Engineering Challenges: Overview

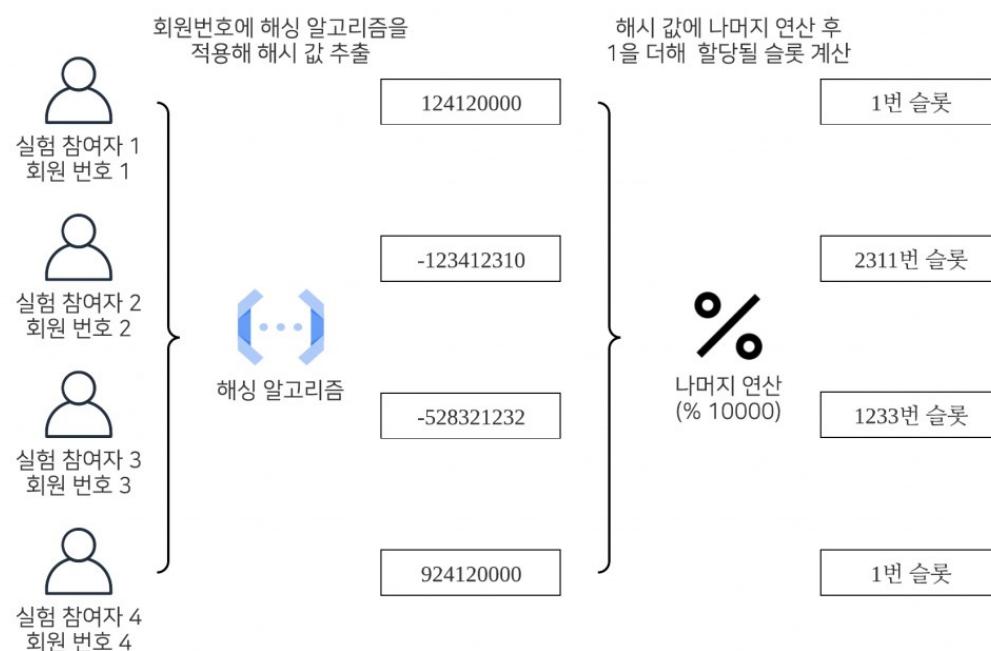
실험의 핵심 요소는 적절한 소프트웨어 아키텍처



# Engineering Challenges: Assignment

Feature Variant에 대해 트래픽을 분배: Hash Function

- Randomize 요청에 따라 \***Deterministic**하게 응답 관리자가 설정한 비율대로 올바르게 배정
- Murmur Hash3, SHA1(Facebook PlanOut), SHA256, MD5
- 여기서 Bucket(Slot)은 물리적으로 존재하는 저장 공간이 아니며, 논리적으로 존재하는 개념



```
def getHash(self, appended_unit=None):
    if 'full_salt' in self.args:
        full_salt = self.getArgString('full_salt') + '.' # do typechecking
    else:
        full_salt = '%s.%s%' % (
            self.mapper.experiment_salt,
            self.getArgString('salt'),
            self.mapper.salt_sep)

    unit_str = '.'.join(map(str, self.getUnit(appended_unit)))
    hash_str = '%s%s' % (full_salt, unit_str)
    if not isinstance(hash_str, six.binary_type):
        hash_str = hash_str.encode("ascii")
    return int(hashlib.sha1(hash_str).hexdigest()[:15], 16)
```

\*Deterministic : 동일한 항목에 대해 항상 동일한 결과를 반환, 시드가 변경되지 않은 한 같은 사용자는 항상 동일한 정수로 맵핑

Source) 우아한형제들, 실험과 기능플래그를 위한 실험플랫폼 구축하기

# Engineering Challenges: Assignment

Assignment 샘플 모집단 목록을 재사용 한다면? Carryover Effect (Kohavi et al. 2012)

- 동일한 사용자 그룹이 여러 테스트에서 지속적으로 변경 사항을 경험할 때 발생
- 버킷 할당에 사용되는 일관된 시드 때문에 버킷은 이전 A/B 테스트를 "기억"하여 후속 테스트 결과에 영향
- 테스트 중인 새로운 기능에 사용자의 학습이 필요한 경우 기존 처치 그룹 구성원은 더 빨리 적응하여 다른 그룹의 사용자보다 우위를 점할 수 있음

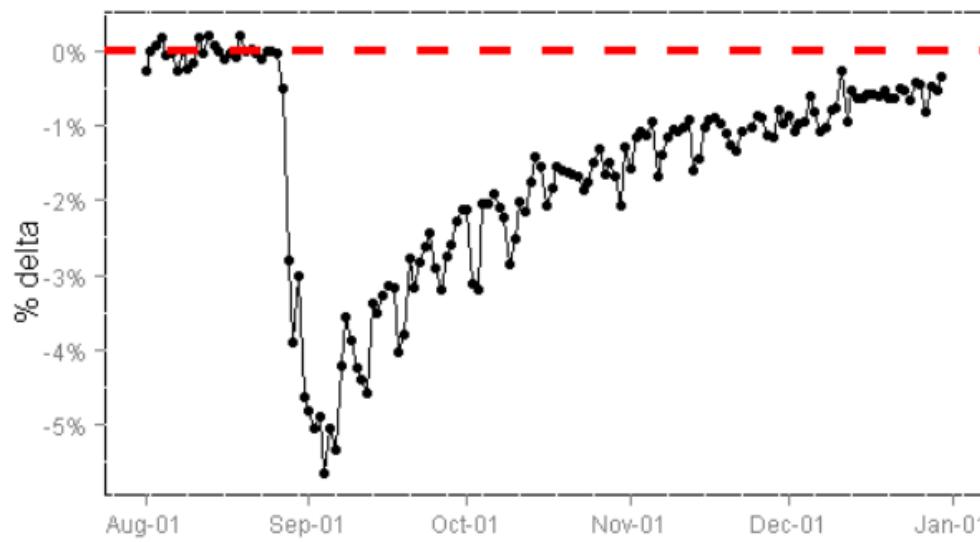
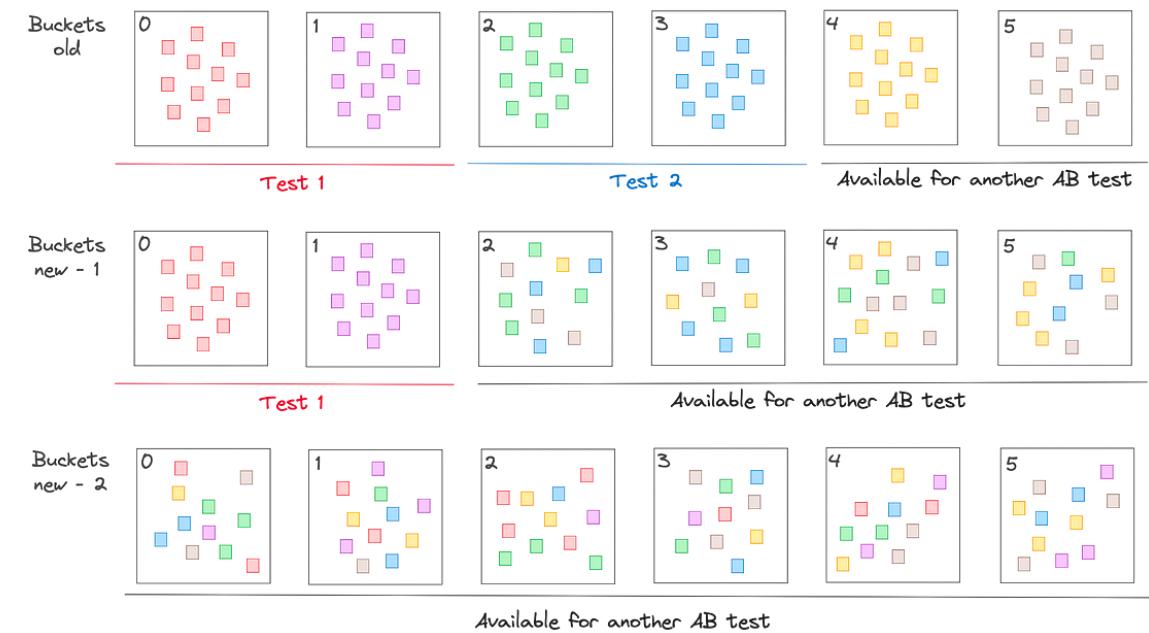


Figure 9: Long Lasting (3 Months) Carryover Effects



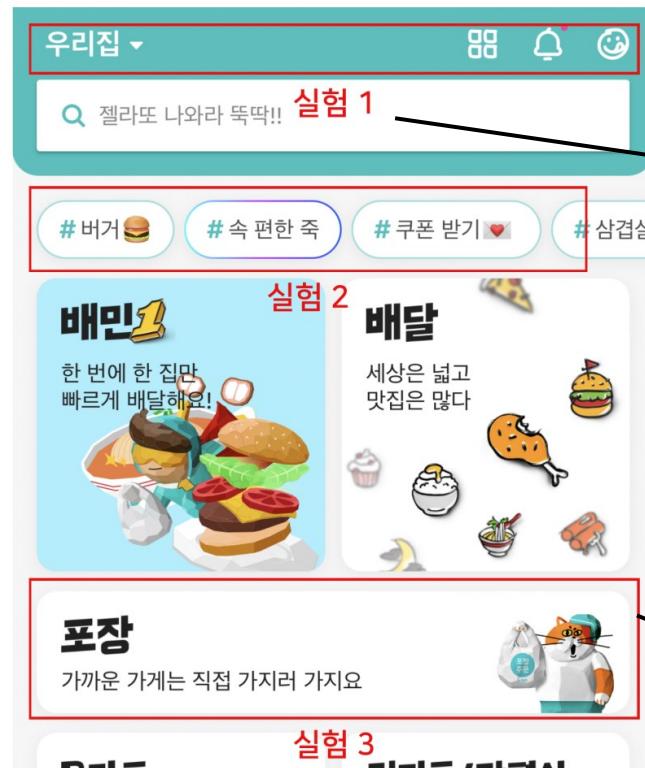
Source) On AB tests and Carryover Effect

Source) Kohavi et al., Trustworthy Online Controlled Experiments: Five Puzzling Outcomes Explained, 2012

# Engineering Challenges: Assignment

배달의민족 사례: 실험간 충돌을 방지하기 위해서 각각의 실험이 가지는 슬롯의 범위 조정

실험 간에 서로 영향을 미치지 않고, 각 실험의 독립성을 보장



**Two-level bucket system (Kohavi et al. 2012)**  
**A/A 이후 2<sup>nd</sup> Level 실험 그룹만 새로운 Hash로 Re-randomized (P-value <0.2)**



# Engineering Challenges: Release

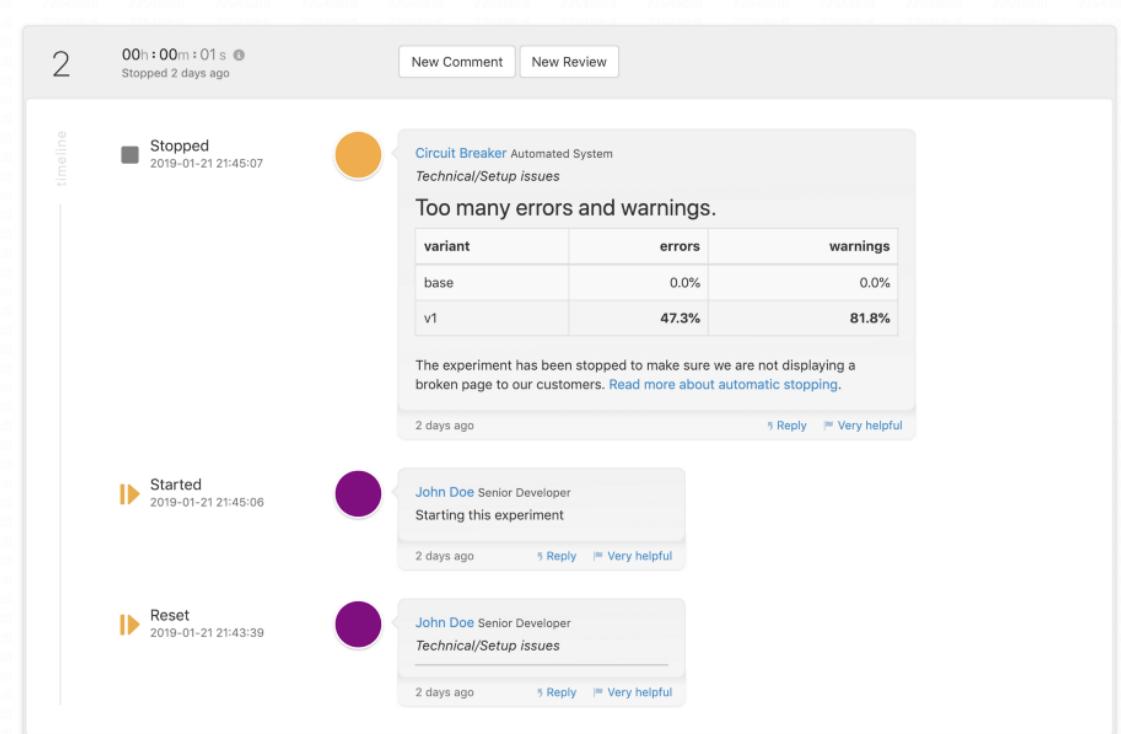
## Feature Flag(Toggle)

- 코드 내에서 분기하여 진행하는 코드 테스트 - 특정 집단의 사용자에게 각 집단의 Flag 값을 받음으로 A/B 테스팅 수행
- 특정 사용자(버킷)에게만 Code를 On하도록 제어  
운영 용이: 처치 Variant가 기술적 문제를 유발하거나 주요 가드레일 메트릭의 감소를 일으킬 때 즉각 중단 가능(기능 Off)

```

1 if isEnabled('fastCheckout', $user)
2 # code block containing new feature
3 else
4 # code block containing old functionality
5 end

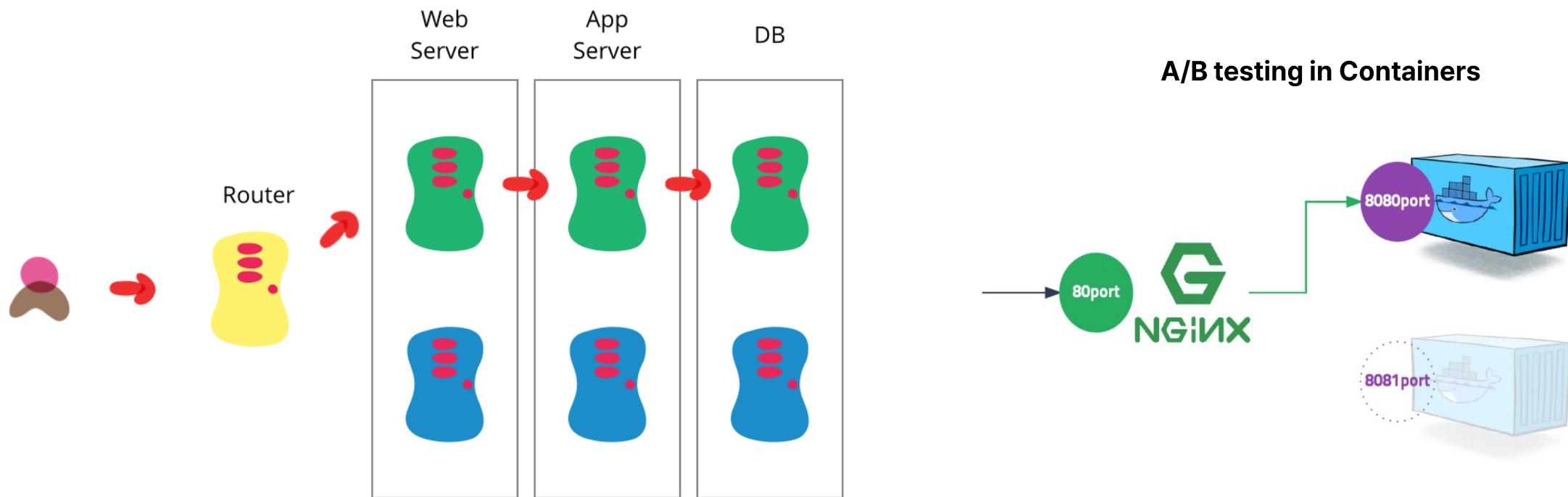
```



# Engineering Challenges: Release

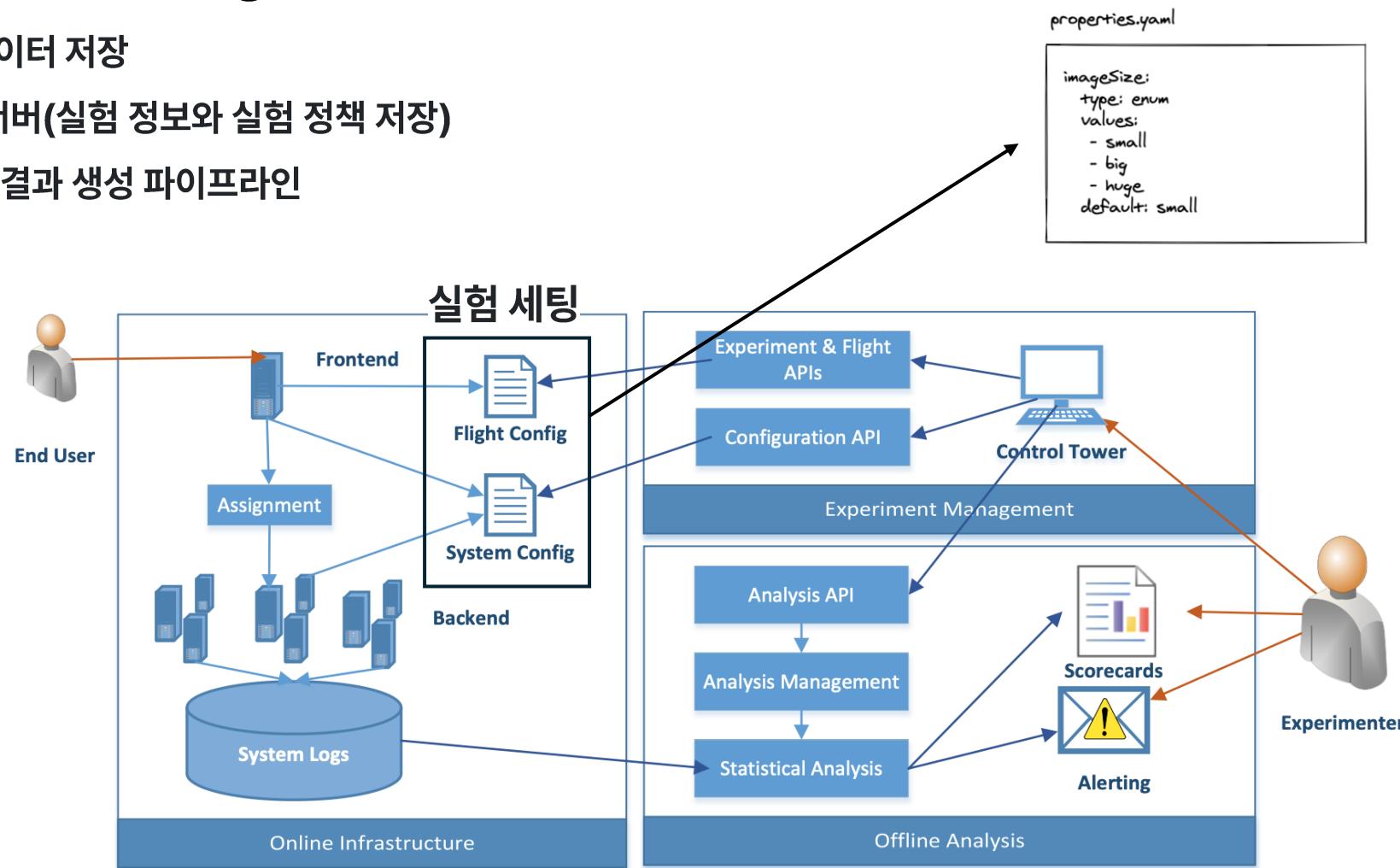
## 라우팅: 병렬의 인스턴스 운영 및 사용자 라우팅

- Blue-Green 배포 : 두 가지 버전(Blue & Green)을 통해 하나는 운영하고 하나는 다른 버전으로 활용
- 서비스 인스턴스를 병렬로 운영하여, A/B 테스트 수행 - 각각의 환경을 다른 사용자 그룹에 대한 테스트로 할당
- 인스턴스 관리와 트래핑 라우팅을 효율적으로 하기 위해 도커 및 Kubernetes의 기술이 필요할 수 있음



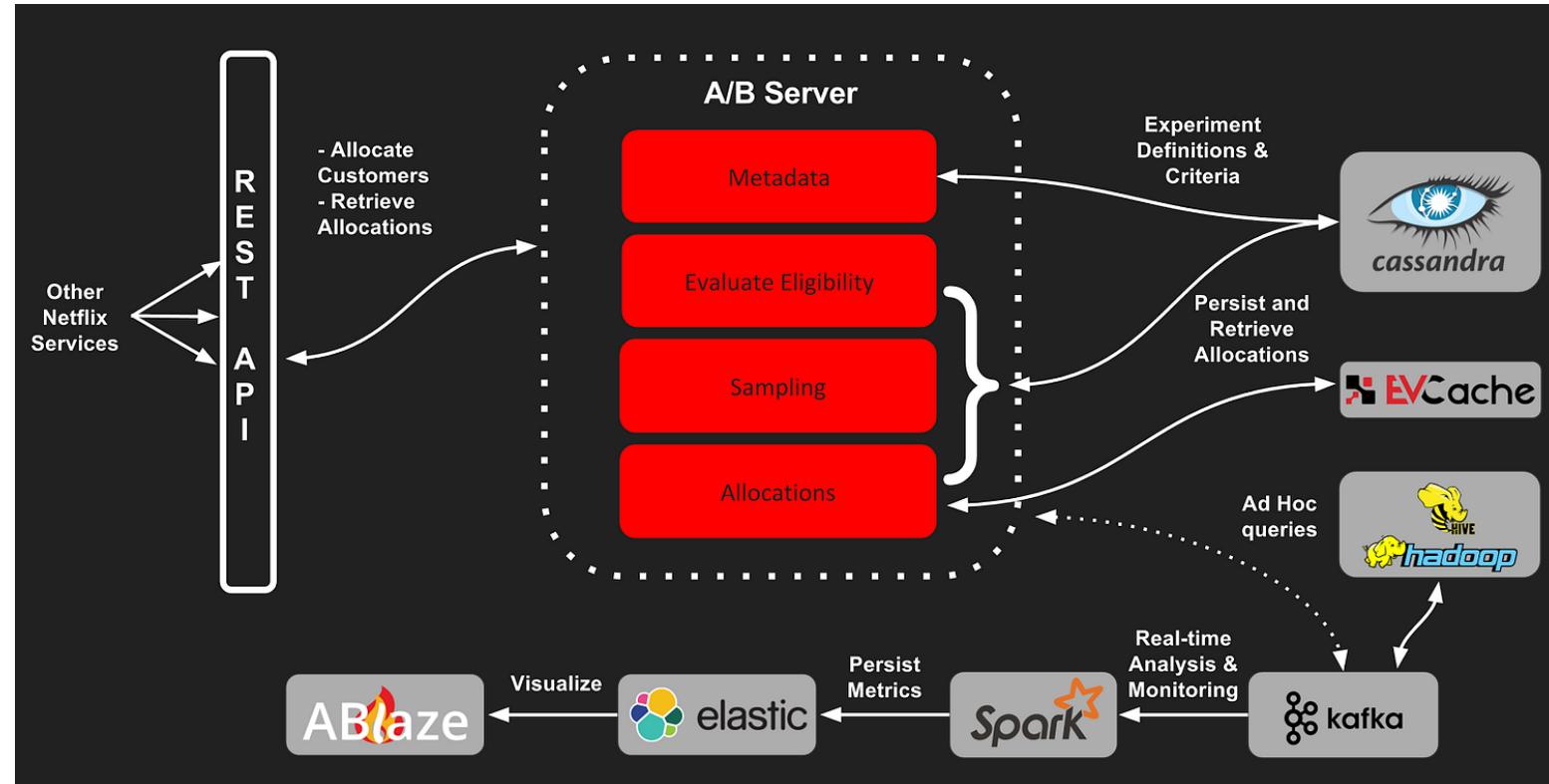
# Engineering Challenges: Architecture

1. 사용자 로그 및 메타 데이터 저장
2. 실험 관리와 Config 서버(실험 정보와 실험 정책 저장)
3. 지표 계산 및 대시보드 결과 생성 파이프라인



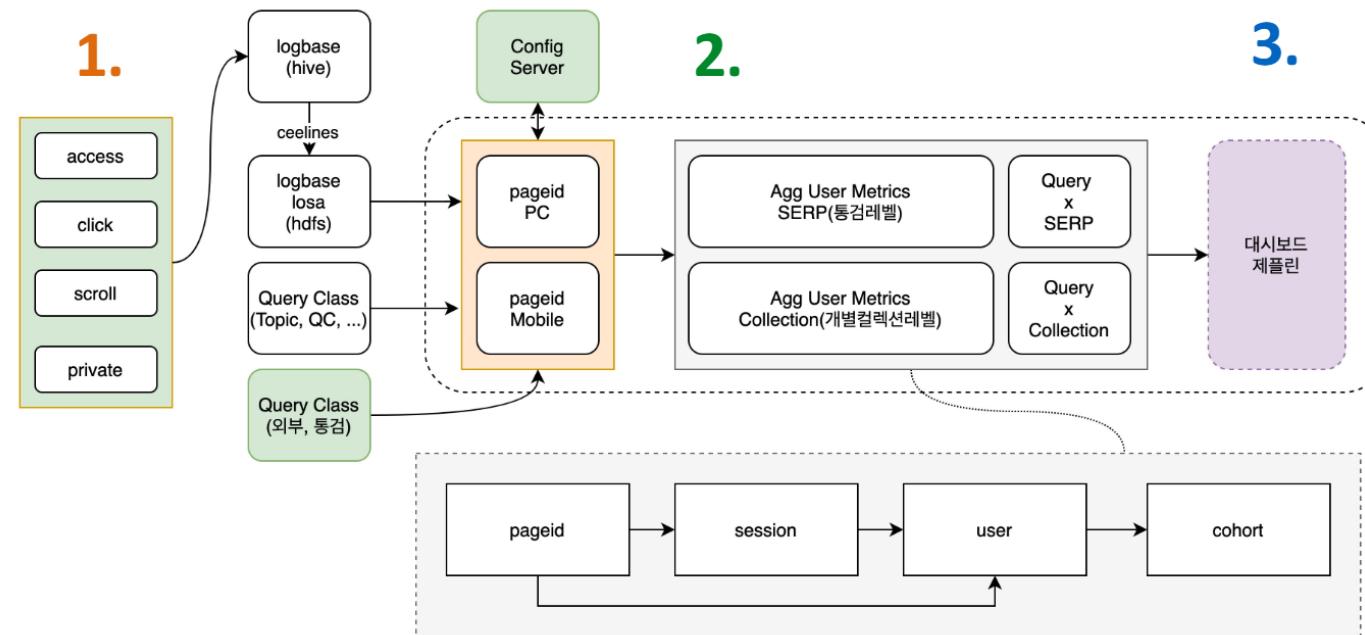
# Engineering Challenges: Architecture

1. 사용자 로그 및 메타 데이터 저장
2. 실험 관리와 Config 서버(실험 정보와 실험 정책 저장)
3. 지표 계산 및 대시보드 결과 생성 파이프라인

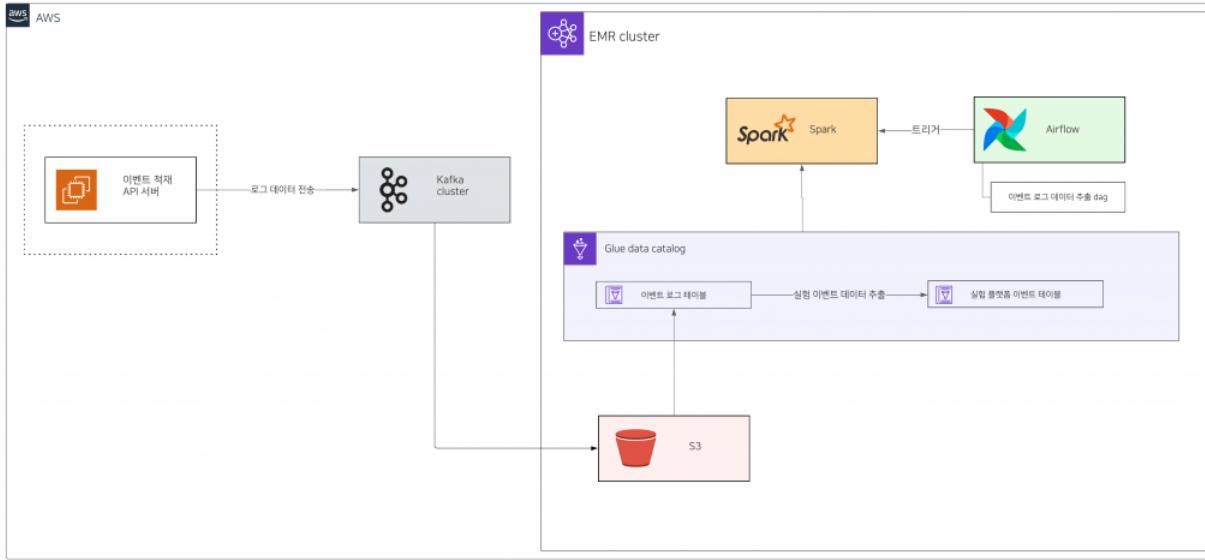


# Engineering Challenges: Architecture

1. 사용자 로그 및 메타 데이터 저장
2. 실험 관리와 정보를 저장하는 Config 서버 + 정제 작업
3. 지표 계산 및 대시보드 결과 생성 파이프라인



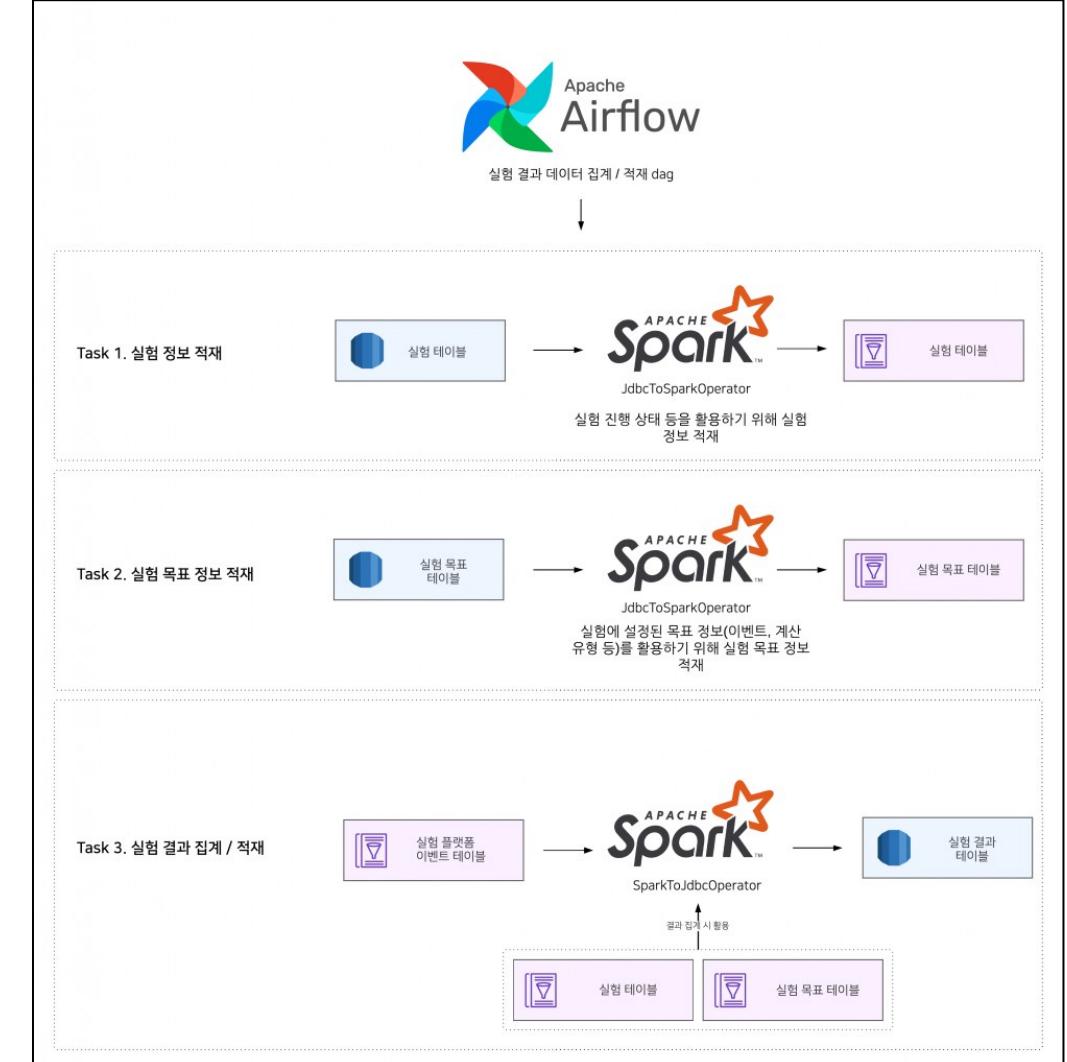
# Engineering Challenges: Architecture



1. 사용자 로그 및 메타 데이터 저장

2. 실험 관리와 Config 서버(실험 정보와 실험 정책 저장)

3. 지표 계산 및 대시보드 결과 생성 파이프라인



# Summary

## Assignment : Hash Function

- Randomize 요청에 따라 Deterministic하게 응답 관리자가 설정한 비율대로 올바르게 배정

## Release :

- Feature Toggles + Release Strategy (Blue & Green, Canary)

## Architecture:

- 사용자 로그 및 메타 데이터 저장
- 실험 관리와 Config 서버(실험 정보와 실험 정책 저장)
- 지표 계산 및 대시보드 결과 생성 파이프라인

# Analytics Challenges

| Essential                               | Pitfall                                |
|---|--|
| A/A testing                             | Overall Evaluation Criteria (OEC)      |
| Statistical significance tests          | Primacy and newness                    |
| Statistical power and confidence        | Page speed/latency                     |
| Randomisation and Sample Ratio Mismatch | Day of week effects                    |
| User group assignments                  | Beacon (loss)                          |
| Data aggregation                        | Browser differences                    |
|   | Carry over effects                     |
|   | Robots                                 |
|   | Device time                            |
|   | Browser redirects                      |
|   | Error checks                           |
|   | Unplanned differences between variants |

Table 2.1: Factors affecting trustworthiness and soundness of A/B tests divided per category.

# Analytics Challenges

인과추론 실무자의 Lifecycle – Design 편

## OEC(Overall Evaluation Criterion) : 실험의 목표를 달성했는지 측정하기 위한 정량적인 지표

- 특정 부서의 이해관계만을 반영하는 것이 아니라, 실험에 관련된 모두가 합의된 지표
- OEC는 일반적으로 2차 지표를 사용함 -> 매출/세션, 클릭/노출
- 지표의 선정은 이후 분산 추정에도 영향을 미칠 수 있음 (Delta Method)
- 전사의 목표와 맞닿아 있어야 하며, 단순히 매출과 같은 후행지표를 설정하면 안됨 (린스타트업의 지표 관련 부분 참고)

## 가드레일 지표(Guardrail Metric) : 가정을 위반하는 것을 방지하기 위해 모니터링하는 지표

- 실험 결과의 신뢰도를 평가할 수 있는 신뢰도 가드레일 지표, 비즈니스를 보호할 수 있는 조직 가드레일 지표

## Before the A/B Test – 모든 가설이 테스트할 가치가 있는 것은 아님 (정답이 있는 것이 아니기 때문에, 조직의 합의가 중요)

- 고객의 어떤 문제를 해결하려고 하는지? 혹은 어떤 혜택을 주려고 하는지?
- 해당 기능이 고객이 얻는 가치를 향상시키는가? 그렇다면 고객이 얻는 가치가 매출 증대로 이어지는가?
- 목표와 맞닿은 OEC & 가드레일 지표 설정 -> 실험을 제품에 적용할지에 대한 A/B Test Ship의 기준
- OEC의 개선이 가드레일 지표에 영향을 주는 것에 대해 어떻게 절충할 것인가? -> 가드레일 지표와 OEC의 가중치 설정**

# Analytics Challenges

인과추론 실무자의 Lifecycle – Design 편

## 테스트에 필요한 유저와 얼마나 길게 테스트를 해야 하는가?

- 테스트 수행 기간을 정하기 위해서는 테스트를 위한 샘플 사이즈를 알아야 하며, 요구되는 파라미터는 다음과 같음

- Type II 에러율  $\beta$  또는 Power ( $1-\beta$ ) -> 업계 표준 0.8
- 유의 수준  $\alpha$  -> 업계 표준 0.05
- 최소 검출 가능 효과 (Minimum detectable effect, MDE)

$$\text{Sample size approximately equal} = \frac{16 * \sigma^2}{\delta^2} \quad \begin{array}{l} \xrightarrow{\text{sample variance}} \\ \xrightarrow{\text{Difference between treatment and control}} \end{array}$$

[Trustworthy Online Controlled Experiments](#) by Ron Kohavi, Diane Tang and Ya Xu

- 샘플 사이즈와 파라미터간 관계

- $\sigma$ 가 크면 샘플이 더 필요하고,  $\delta$ 가 크면 적은 샘플로도 충분함
- $\sigma$ 는 존재하는 데이터로 계산할 수 있으나  **$\delta$ 는 미리 계산할 수 없음**
- 특정 신뢰 수준과 검정력을 가정한 샘플 크기를 계산을 위해 MDE가 필요함

# Analytics Challenges

인과추론 실무자의 Lifecycle – Design 편

## Minimum Detectable Effect

- 실무에서 효과를 거둘 수 있는 최소 수치를 의미함
  - 실험에서 수익률 0.1% 상승을 Relative MDE로 설정할 수는 있지만, 실제로는 여러 이해관계자들과 논의를 통해서 정해야 함
- 
- 샘플 사이즈를 알게 되었다면 샘플 사이즈를 사용자 수로 나눠 테스트 수행 기간을 구할 수 있음
    - 일반적으로 2주 정도는 테스트하길 권장하지만 데이터를 더 수집할 수 있다면 길 수록 좋음
    - 일주일 미만인 경우에는 주간 패턴을 포착하기 위해 최소 7일 동안 실험

# Analytics Challenges

A/B 테스트에서 무작위화 단위가 사용자인 경우, 각 사용자를 처치 그룹 A와 컨트롤 그룹 B로 할당  
-> 독립성 확보 But, 무작위화와 독립성 가정이 모든 상황에서 완벽하게 유지가 되지 않음

## 1. 상관 관계가 있는 데이터: 사용자들은 서로 독립적이지 않을 수 있음

예를 들어, 같은 가정이나 기관, 지리적 위치에 속한 사용자들은 비슷한 행동 패턴이나 선호도를 보일 수 있음  
이러한 상황에서는 **사용자들 간의 상호작용이나 외부 요인으로 인해 결과에 영향을 줄 수 있음**

## 2. 무작위화의 한계: 모든 사용자를 완벽하게 무작위로 그룹에 할당한다 하더라도, 각 그룹의 구성이 완전히 균형을 이루는 것은 아님

예를 들어, 성별, 연령, 사용 패턴 등의 측면에서 불균형이 발생할 수 있습니다. 이러한 불균형은 연구 결과의 해석을 왜곡할 수 있음

## 3. 외부 타당성의 문제: A/B 테스트 결과가 특정 사용자 그룹에 대해서만 유효할 수 있음

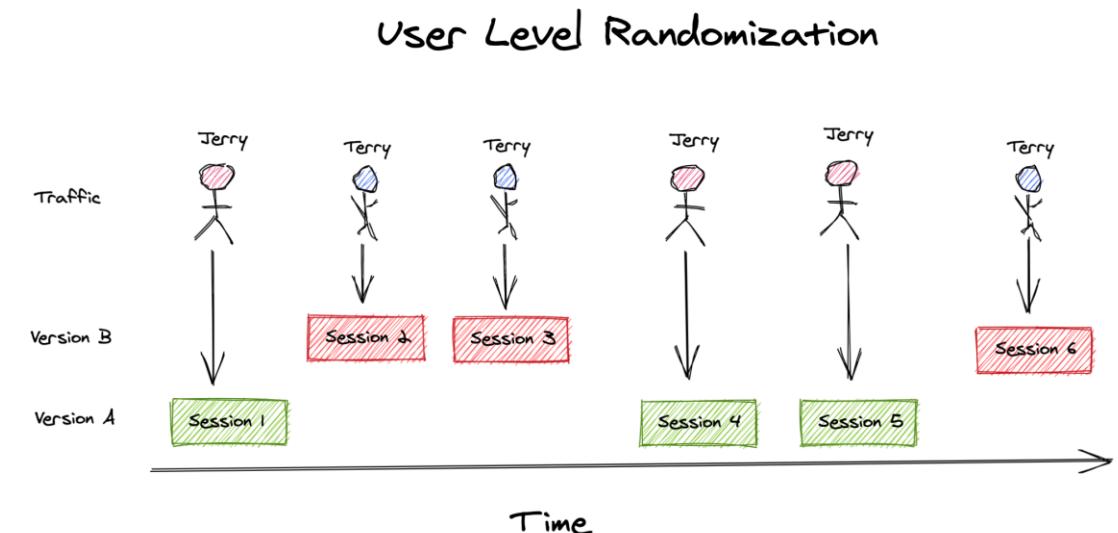
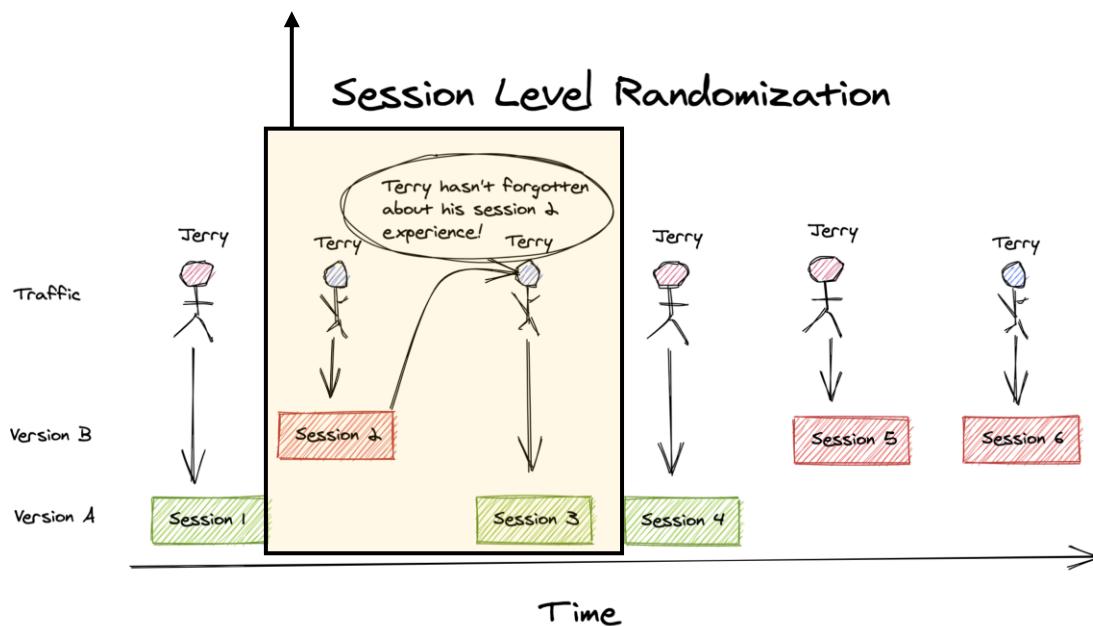
예를 들어, 올해 실험한 Feature의 효과가 내년에도 동일하게 작동하는가? 외부의 생태계 변화, 계절성에 따라서 가설 검증의 결과가 달라질 수 있음

# Analytics Challenges: Randomization Unit

무작위 단위(Randomization Unit) : 무작위로 할당되는 단위 (Session vs User)

고려해야 하는 가정: Independence of randomization units -> User Level Randomization

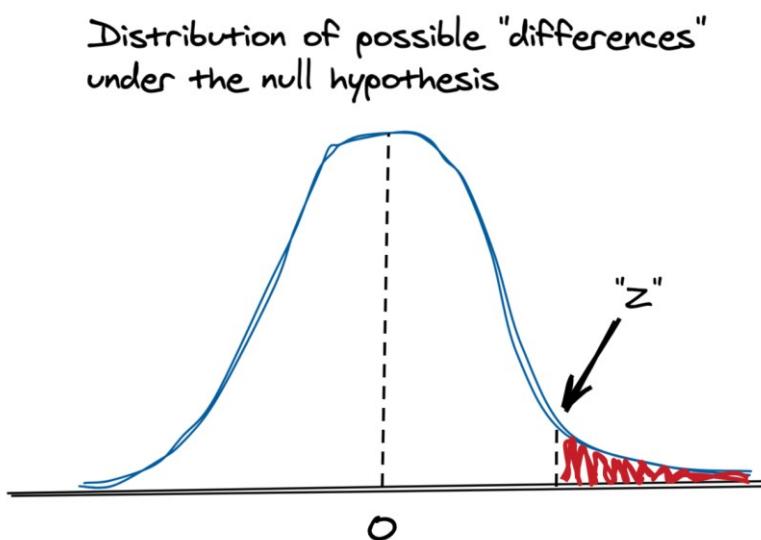
Independence? 이월 효과(carryover effect)



# Analytics Challenges: Randomization Unit

**무작위 단위(Randomization Unit) != 분석 단위(Analysis Unit : 분석에 사용되는 지표의 단위)의 경우**

- 무작위 단위 내의 분석 단위들이 서로 독립적이지 않음 -> 사용자들의 PV이 사용 습관이나 선호도에 따라 비슷한 경향을 보일 수 있음  
예시: CTR = Sum(Click) / Sum(PV) : 무작위 단위는 유저이지만, 분석 단위인 PV는 서로 독립적이지 않음  
-> These measurements **are not independent** and therefore our **variance estimate is biased** -> 잘못된 의사결정
- 유저 단위로 Randomization Unit을 설정하고, 분석 단위도 유저 단위로 통일 (Click/ PV per visitor)
- 우리 회사 주요 지표는 CTR인데요? – CTR 분산 추정 방식 참고: [Delta Method](#)



표본 분산을 계산하는 접근 방식에서는 샘플이 i.i.d(독립적으로 동일하게 분포됨)이거나 적어도 상관 관계가 없다고 가정

$$Z = \frac{(\hat{p}_T - \hat{p}_C) - (p_T - p_C)}{\sqrt{Var(p_T - p_C)}} = \frac{\hat{p}_T - \hat{p}_C}{\sqrt{Var(\hat{p}_T) + Var(\hat{p}_C)}} = \frac{\hat{p}_T - \hat{p}_C}{\sqrt{\hat{p}(1 - \hat{p})(\frac{1}{n_T} + \frac{1}{n_C})}}$$

where,

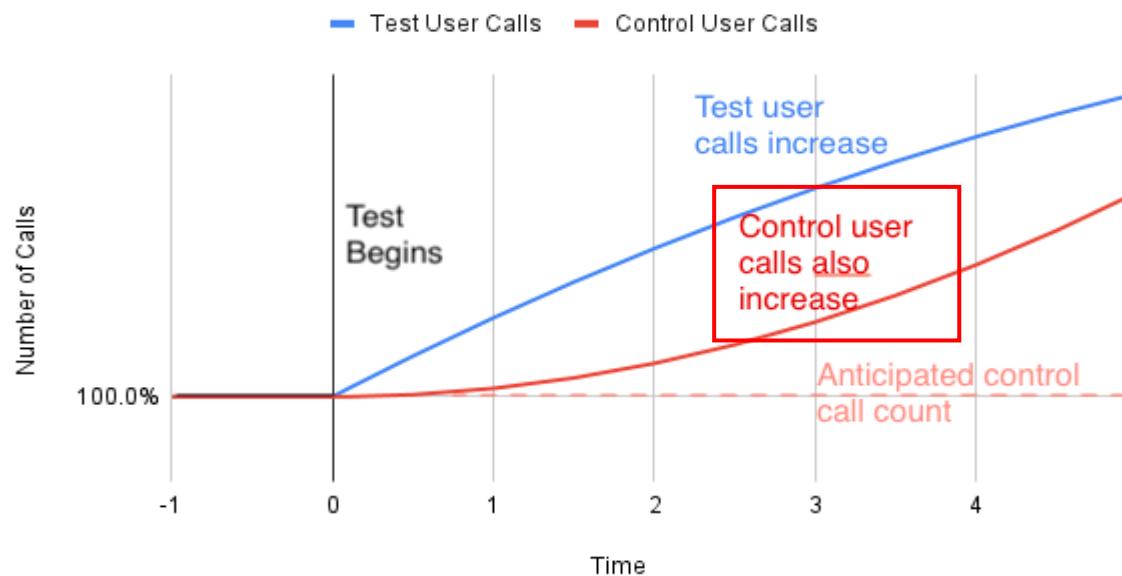
- $\hat{p}_T$  and  $\hat{p}_C$  are our observed treatment and control session conversion rates
  - $p_T$  and  $p_C$  are the true treatment & control session conversion rates
  - $p_T - p_C$  is 0 under the null hypothesis
- $n_T$  and  $n_C$  are the number of sessions in treatment and control
- $\hat{p}$  is the pooled session conversion rates (i.e. total number of conversions / total number of sessions, from both groups)

## Analytics Challenges: Interference

- 사용자가 상관관계가 있는 두 개 이상의 실험에 참여할 때, 처치 그룹이 통제 그룹의 행동에 영향을 미침 (SUTVA)
- 사용자들이 동시에 여러 실험에 참여하여, 결과에 왜곡을 줄 수 있기 때문에 상호작용을 감지하고 경고
- P2P, 통화, 채팅 등에서 실험을 진행할 때 이러한 Interference로 인한 Network effect가 존재할 수 있음

실험: 통화가 갑자기 끊긴 경우 테스트 사용자에게 재다이얼 버튼이 제공

Dropped-Call Redial Experiment: Number of Calls



Control 그룹의  
Number of Call도 증가

서로 아는 사람 ㅎㅎ

Treatment



통화 중  
->재다이얼

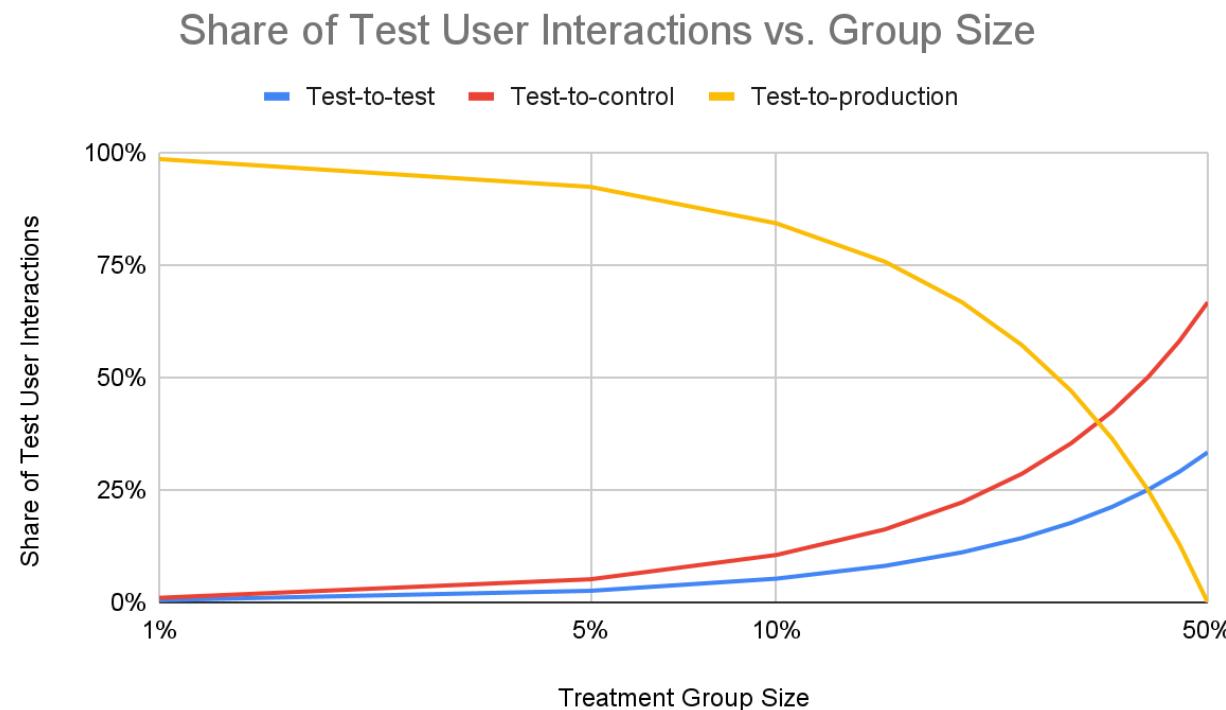
Control



Interference  
-> Treatment의 행동이 Control에 영향

## Analytics Challenges: Interference

- 실험의 규모가 커질수록, 처치 그룹의 숫자가 커질 수록 실험 참여자 상호간에 영향을 주는 Indirect, direct interference가 커짐
- A well-designed experiment must not only be capable of introducing an effect, but that **effect must also be contained in order to be measurable**. What we really need to do is **isolate the test and control users, enabling us to better compare test interactions against control interactions**.



# Analytics Challenges: Interference

## Randomized unit 변경 - 네트워크 클러스터 (Network Cluster) 생성

- 그룹 외부의 사용자보다 그룹 내부의 사용자들과 더 상호작용할 수 있는 클러스터를 생성
- 클러스터에 따라 사용자를 무작위로 처리하여 동일한 클러스터의 모든 사용자가 동일한 치료 그룹에 속하도록 처리
- 특정 커뮤니티(군집) 전체가 처리(treatment) 그룹이거나 대조(control) 그룹에 속하는 실험 설계

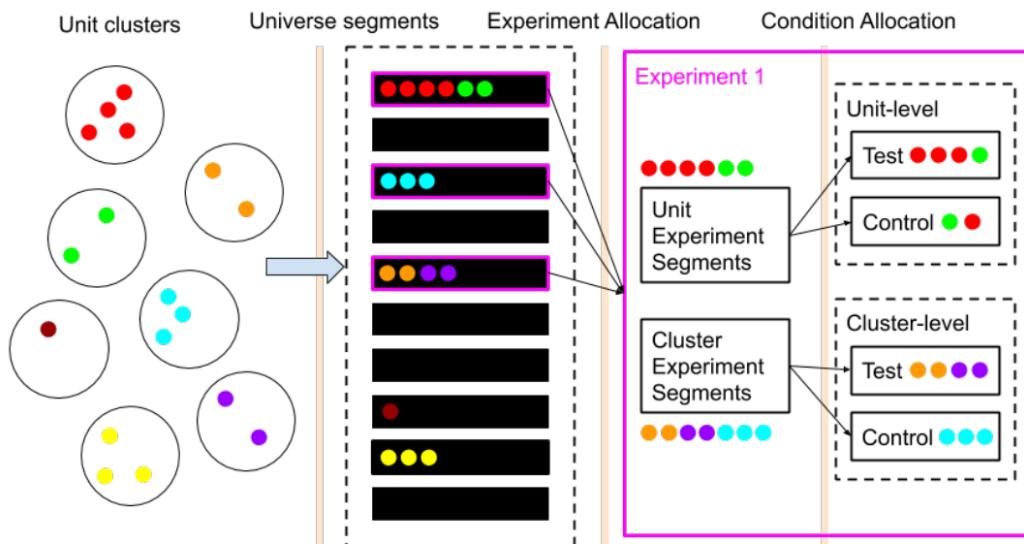


Figure 1: Visualization of the network experiment randomization process.

### 1) 클러스터링 생성 (좋은 클러스터란 무엇인가에 대한 내부 기준 필요)

Restreaming Linear Deterministic Greedy(Nishimura & Ugander, 2013)

2) A : Unit 단위 무작위 방식 B : 클러스터 기준 분할

3) A(Unit level)와 B(Cluster level)에 대한 효과 차이를 통해 SUTVA 여부 체크( $\Delta = \mu_{BR} - \mu_{CBR}$ )

A: Difference in means B: \*Horvitz-Thompson Estimator

| Statistic   | Experiment 1  |                |  | Experiment 2  |                |  |
|---|---------------|----------------|--|---------------|----------------|--|
|   | pre-treatment | post-treatment | post-treatment<br>(Y = Y <sub>t</sub> - Y <sub>t-1</sub> ) | pre-treatment | post-treatment | post-treatment<br>(Y = Y <sub>t</sub> - Y <sub>t-1</sub> ) |
| BR Treatment Effect ( $\hat{\mu}_{br}$ )              | -0.0261       | 0.0432         | 0.0559   | 0.0230        | 0.2338         | 0.2108   |
| CBR Treatment Effect ( $\hat{\mu}_{cbr}$ )            | 0.0638        | 0.1653         | 0.0771   | 0.2733        | 0.8123         | 0.5390   |
| Delta ( $\Delta = \hat{\mu}_{br} - \hat{\mu}_{cbr}$ ) | -0.0899       | -0.1221        | -0.0211  | -0.2504       | -0.5785        | -0.3281  |
| BR standard deviation ( $\hat{\sigma}_{br}$ )         | 0.0096        | 0.0098         | 0.0050   | 0.3269        | 0.3414         | 0.2911   |
| CBR standard deviation ( $\hat{\sigma}_{cbr}$ )       | 0.0805        | 0.0848         | 0.0260   | 0.9332        | 0.9966         | 0.5613   |
| Delta standard deviation ( $\hat{\sigma}$ )           | 0.0811        | 0.0856         | 0.0265   | 0.9367        | 1.0000         | 0.5712   |
| p-value (2-tailed)                                    | <b>0.2670</b> | <b>0.1530</b>  | <b>0.4246</b>  | <b>0.5753</b> | <b>0.2560</b>  | <b>0.0483</b>  |

Population: 36% of all LinkedIn users [Bernoulli: 20%, Cluster-based: 16%]

– Time period: 4 weeks , Number of clusters: k = 10,000 , Outcome: feed engagement

# Analytics Challenges: Interference (참고)

## Hypothesis Test

$H_0$ : SUTVA Holds

$$E_{\mathbf{W}, \mathbf{Z}} [\hat{\mu}_{cbr} - \hat{\mu}_{cr}] = 0$$

$$\text{var}_{\mathbf{W}, \mathbf{Z}} [\hat{\mu}_{cr} - \hat{\mu}_{cbr}] \leq E_{\mathbf{W}, \mathbf{Z}} [\hat{\sigma}^2]$$

Reject the null when:

$$\frac{|\hat{\mu}_{cr} - \hat{\mu}_{cbr}|}{\sqrt{\hat{\sigma}^2}} \geq \frac{1}{\sqrt{\alpha}}$$

Type I error is no greater than  $\alpha$

## Choosing the Number of Clusters

Understanding the Type II error

Assuming an interference model

$$Y_i = \beta_0 + \beta_1 Z_i + \beta_2 \rho_i + \epsilon_i$$

$\rho_i$  : fraction of treated friends

$$E [\hat{\mu}_{cbr} - \hat{\mu}_{cr}] \approx \rho \cdot \beta_2$$

$\rho$  : average fraction of a unit's neighbors contained in the cluster

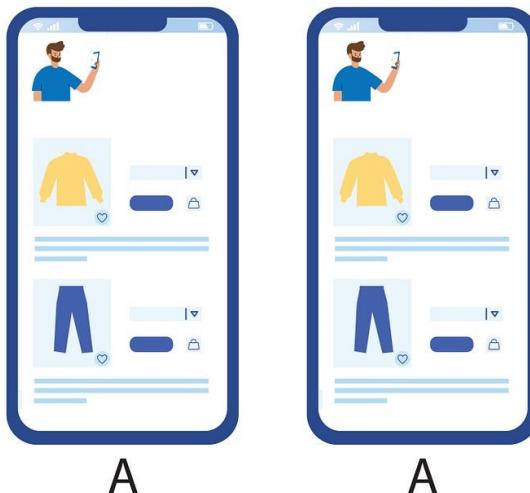
Choose number of clusters  $M$  and clustering  $C$  such that

$$\max_{M,C} \frac{\rho}{\sqrt{\hat{\sigma}_C^2}}$$

# Analytics Challenges: A/A Testing (Randomization Check)

## 실험의 조건: 처치가 잠재적 결과와 독립 – 비교가능한 두 집단

- 두 그룹 간에 관심 지표 및 공변량에서 차이가 없어야 함 (불균형 테스트)
  - (1) A/A 테스트가 필요한 이유: 실험 이전에 대조 버킷과 컨트롤 버킷 사이의 메트릭에 차이가 있는 경우  
A/B 테스트 중에 나타난 메트릭 차이가 고유한 것인지 아니면 처리로 인해 발생한 것인지 알 수 없음
  - (2) A/A 테스트가 필요한 이유: Skewed metrics (e.g., 시청시간, 채널당 채팅 메시지 수) -> T-test 잘 작동?  
-> Skewed metrics은 높은 분산 & Outliers가 발생시키는 잠재적인 문제 발견
  - (3) A/A 테스트가 필요한 이유: 할당의 오류가 존재하는가?



We inspect the **treatment and control consumers before the Experiment start date**, when they were all receiving recommendations from the same production recommender system

Wang, Yuyan, Long Tao, and Xian Xing Zhang. "Recommending for a multi-sided marketplace: A multi-objective hierarchical approach." Available at SSRN 4602954 (2023)

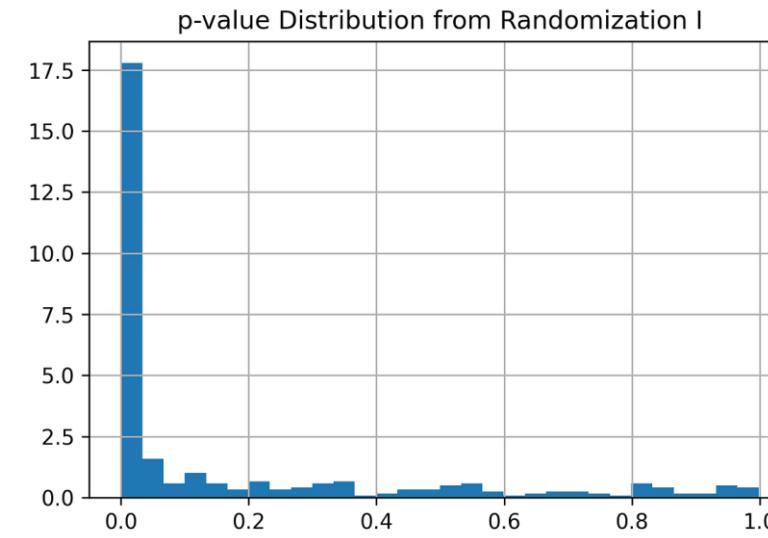
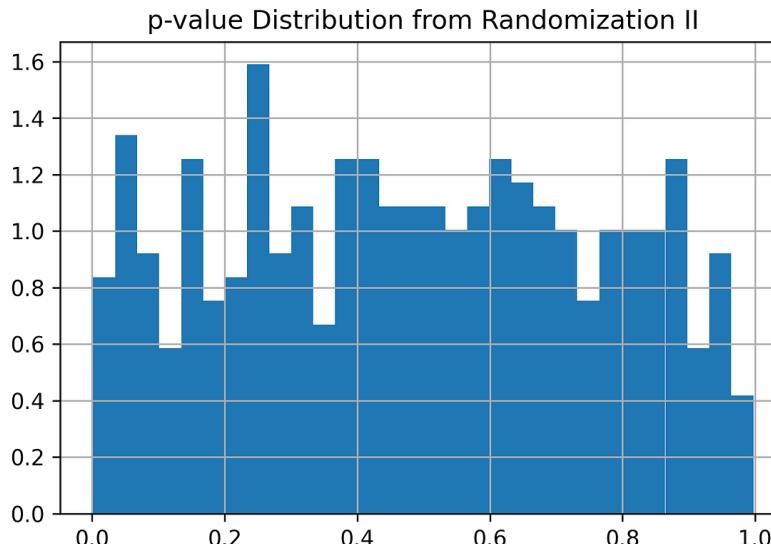
# Analytics Challenges: A/A Testing (Randomization Check)

## 부트스트랩된 A/A 테스트에서 p값 분포를 생성

- A/A t검정에서 결과 p값 분포는 균일-> 즉, p값이 0.05 미만인 경우는 5% 정도 발생
- p값 분포가 균일하지 않으면 테스트 방법론에 결함이 있음을 나타내며 가정을 위반
- 분포가 균일해야 하는 이유는 귀무가설이 참일 때 표본에서 차이를 발견할 확률을 나타내는 p-값의 정의 때문
- 분포가 정말로 균일한지, Kolmogorov-Smirnov test를 수행해야 할 수도 있음

$H_0: \text{average metric in group A} = \text{average metric in group B}$

$H_1: \text{average metric in group A} \neq \text{average metric in group B}$



# Analytics Challenges: A/A Testing 실무 예시

인과 문제: Uber의 추천 알고리즘 A와 B가 전환율, 소비자당 주문 수, 검색률 등에 미치는 효과 측정

목표: 실험 시작일 28일 전에 처리군과 통제군 간의 주요 지표 차이에 대한 p-값을 계산

(1) 처리 그룹과 컨트롤 그룹 소비자는 동일한 알고리즘에 의해 생성된 추천을 받게 됨

| Metric              | Conversion rate | Basket value per order | Retention rate | Orders per consumer | Search rate |
|---------------------|-----------------|------------------------|----------------|---------------------|-------------|
| A/A testing p-value | 0.326           | 0.452                  | 0.947          | 0.853               | 0.286       |

Table 10 p-values for the A/A testing on key business metrics.

(2) 플랫폼에서 다양한 소비자 행동을 포괄하는 472개의 지표를 수집하고, 472개의 지표 차이에 대한 p-값을 계산

(3) 처리 그룹과 컨트롤 그룹 소비자는 통계적으로 유의미하게 다르지 않다는 귀무가설 하에서, p-값은 균등 분포

(4) 472개의 p-값의 분포에 대해 KS-검정 실시했으며, 균등 분포를 따른다는 귀무가설을 기각할 수 없음

(5) 무작위 배정이 유효함을 시사

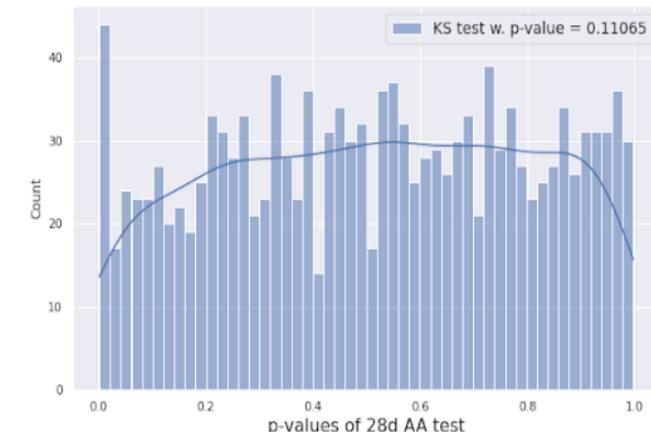
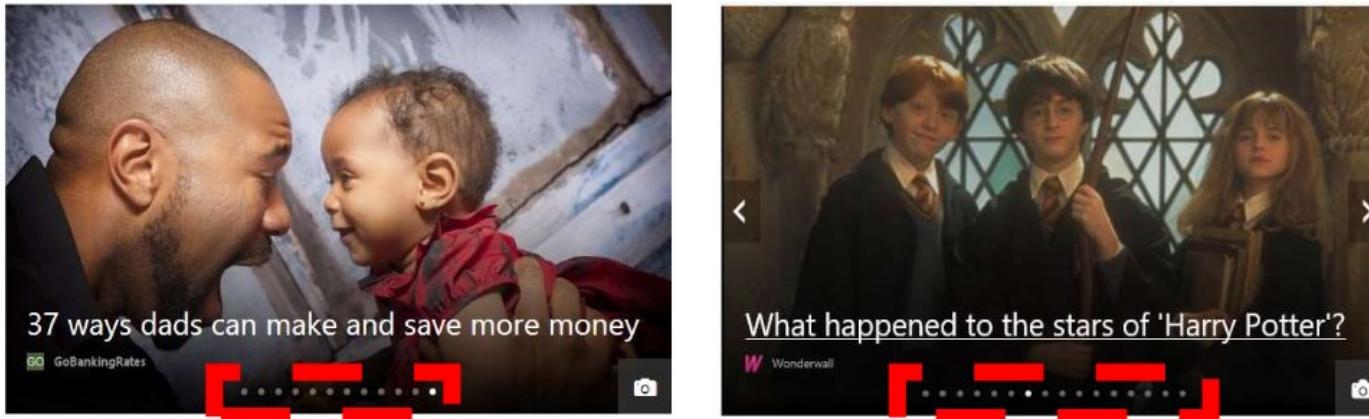


Figure 10 Histogram of the p-values for the 472 metric differences for A/A test. Kolmogorov-Smirnov (KS) test which compares the empirical distribution of the p-values against the uniform distribution on [0,1] has p-value of 0.11, which fails to reject the null hypotheses that these metrics are not statistically significantly different during the A/A testing period.

## Analytics Challenges: Sample Ratio Mismatch (SRM)

- A와 B에서 계산된 사용자 비율과 실험이 시작되기 전에 구성된 비율(예: 50/50 분할) 사이에 통계적으로 유의미한 차이
- SRM 문제는 RCT의 가장 핵심적인 가정인 처치배정 매커니즘이 랜덤이 아니라는 것을 의미함 (Selection Bias)



**Figure 1. The MSN Carousel Experiment.**

- A product team at MSN increased the number of rotating cards on the carousel from 12 to 16
- 해당 실험에서 Carousel 카드와의 상호작용이 크게 감소
- 16 카드 Carousel에 노출된 처치 그룹에서 설정 및 실험 설정에서 예상된 것보다 분석에 포함된 사용자가 적었음
- A/B의 분할에서 이미 편차가 존재하였음 -> Sample Ratio Mismatch(SRM)
- SRM은 Microsoft에서 지난 1년 동안 수행된 실험에서 약 6%를 차지할 만큼 일반적인 Data Quality 문제

# Analytics Challenges: Sample Ratio Mismatch (SRM)

## (1) 할당으로 인한 SRM

> 실험 할당 서비스는 해시 함수를 통해 천 개의 버킷으로 무작위화, But 50/50 테스트가 49.9/50 설정

> 동일한 사용자의 반복 할당은 결정적(**Deterministic**)

> 실험 간에 interference가 없어야 함

많은 실험에 따라 이러한 현상이 발생할 확률이 높아짐 예를 들어, 365개의 다른 버킷을 사용하면, 약 23개의 실험만으로도 적어도 하나의 실험 쌍이 버킷을 공유할 확률이 50% -> 시간이 지남에 따라 사용되는 버킷 목록을 변경하여 해결

## (2) 실험 실행 SRM

> 실험이 제품의 성능을 저하시킬 경우, 사용자가 로그를 발생시키기 전에 프로덕트를 꺼버림 or 잘 못된 트리거링

> 실험을 통해 제품의 성능(앱 로딩 화면)이 저하되었을 때 사용자가 프로덕트를 꺼버려 데이터 로깅 불가

## (3) 로그 처리 SRM

> 처치 그룹의 참여도가 높은 사용자가 컴퓨터 볏으로 분류되어 실험 분석에서 제거..

> 데이터가 처리되는 방식, 즉 로그가 다양한 수준에서 요약 통계로 변환되는 방식에서 발생 (집계, 필터링, 조인)

서로 다른 데이터 파이프라인을 활용하여 원인을 탐지, 분석에서 제외되는 로그 모니터링

# Analytics Challenges: Sample Ratio Mismatch (SRM)

SRM은 Microsoft에서 지난 1년 동안 수행된 실험에서 약 6%를 차지할 만큼 일반적인 Data Quality 문제

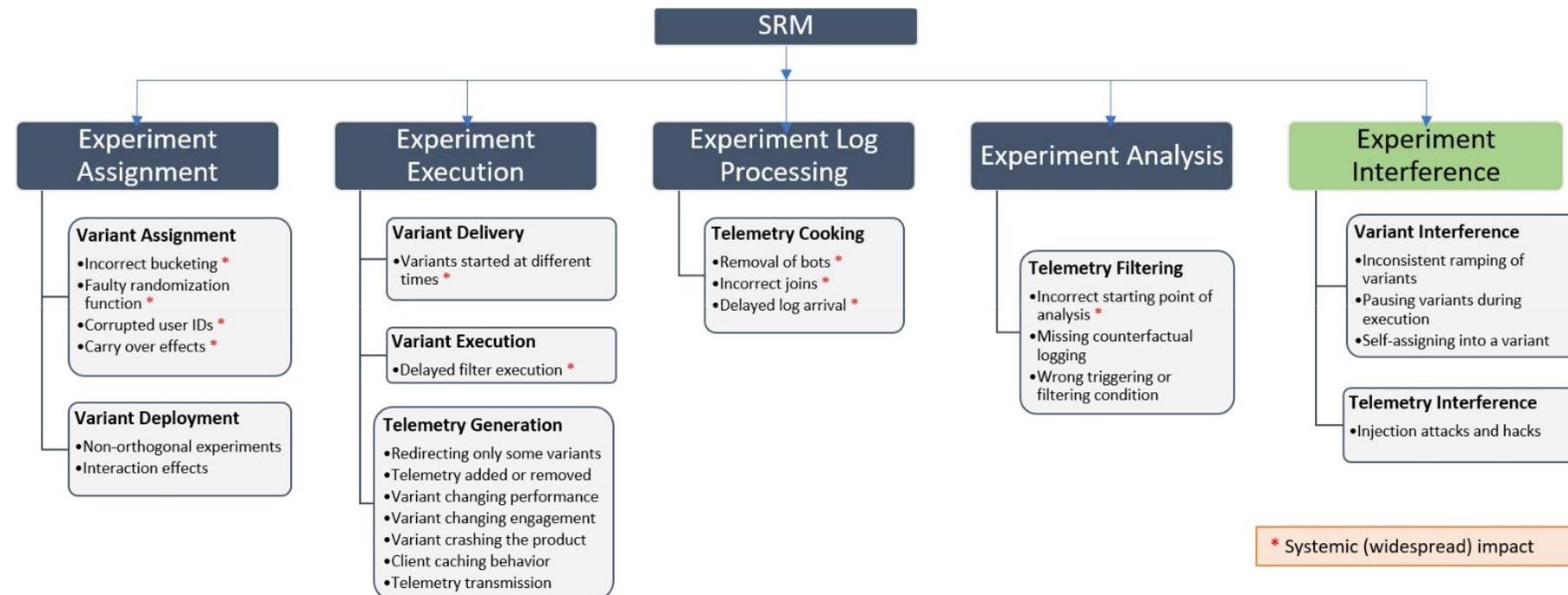
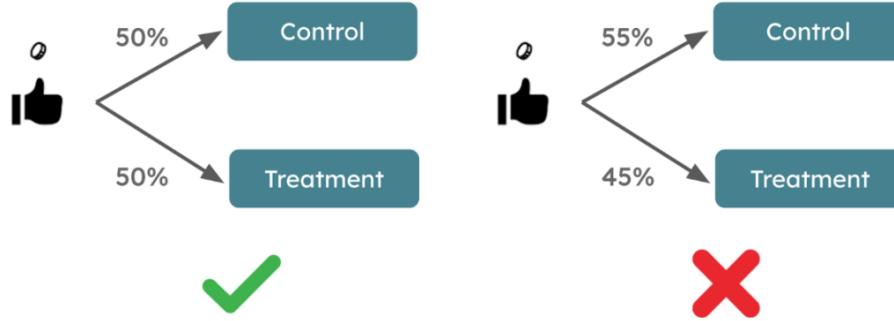


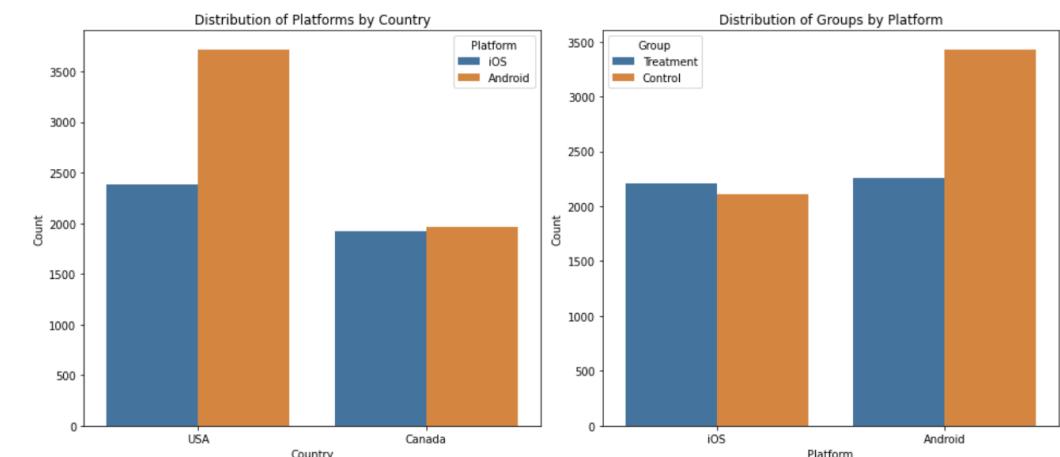
Figure 5. Taxonomy of common SRM types and root-causes.

# Analytics Challenges: Sample Ratio Mismatch (SRM)

기본적인 접근은 세그먼트 하위 모집단에 대해 카이제곱 테스트를 통해서, 비율의 불균형을 파악 -> 일반적인 방법



*Figure 1: If we have two groups that are expected to have a distribution of 50/50, we expect the SRM check would pass if that 50/50 split is indeed observed. **We should be concerned, however, if there is instead a split of 55/45.***

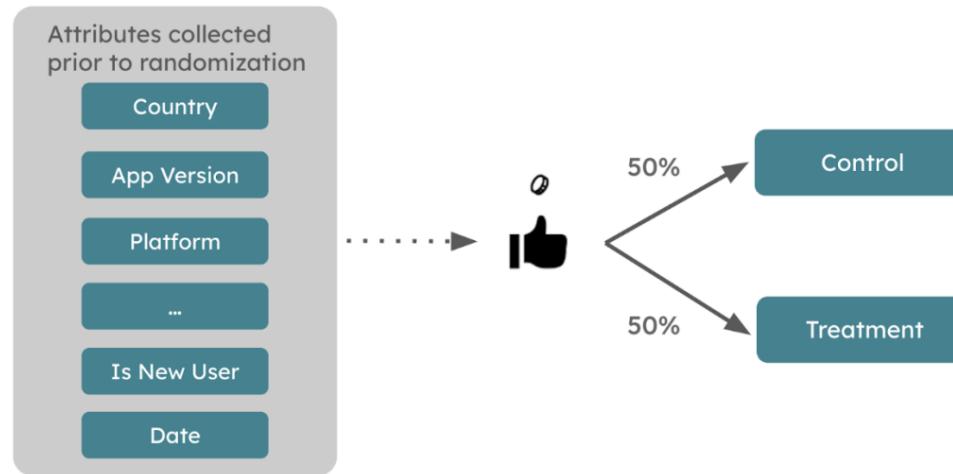


- 불균형을 초래하는 **근본 문제는 Android**
- 미국에는 다른 국가보다 Android 사용자가 더 많기 때문에 문제가 사용자 국가에 있다고 잘못 가정할 수 있음

# Analytics Challenges: Sample Ratio Mismatch (SRM)

가장 중요한 것은 SRM의 근본 원인을 파악하는 것이지만, 카이제곱 검정은 직교화 효과(orthogonal effect)가 없음

> Regression을 활용하여 SRM 발생 원인을 찾고 제어



- Country: USA, Canada, Australia
- Platform: Web, iOS, Android

We will test our approach with three scenarios:

- 1. No imbalance
- 2. Imbalance due to country=Australia
- 3. Imbalance due to country=Australia OR platform=Web

| scenario   | attribute | pvalue   |
|------------|-----------|----------|
| scenario_1 | platform  | 0.357744 |
| scenario_1 | country   | 0.362216 |
| scenario_2 | platform  | 0.233428 |
| scenario_2 | country   | 0.000000 |
| scenario_3 | platform  | 0.000000 |
| scenario_3 | country   | 0.000000 |

- 무작위화 이전에 수집된 속성은 처치 할당에 영향을 미치지 않아야 함 (Selection bias)
- 변수가 처치 할당과 관련되어 있는지 선형회귀를 통해 확인

- Treatment에 영향을 주는 Covariates를 회귀하여, 처치 할당에 영향을 주는 변수파악 및 보정 (Endogenous Choice 상황에서 Matching의 방식과 유사)
  - (1)  $\text{is\_treatment} \sim \text{Covariates}(X)$   
- Platform과 Country가 모두 처치할당을 예측
  - (2)  $\text{metric\_outcome} \sim \text{is\_treatment} + \text{country} + \text{platform}$  (통제)

## Analytics Challenges: 지표의 Variance와 민감도

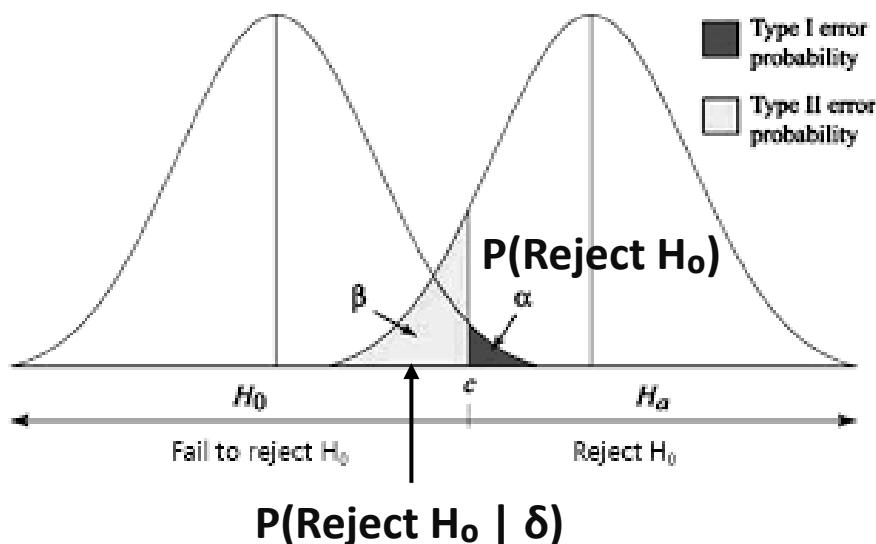
2%의 전환율을 가진 회사에서 Relative MDE 1%에 해당하는 작은 효과를 올바르게 감지하기 위해 1,200만명의 유저가 필요  
**민감도(Sensitivity):** 어떤 측정치에 대해 유의미한 효과(significant effect)를 감지하는 능력

$$P(\text{Reject } H_0) = \int P(\text{Reject } H_0 | \delta) dP(\delta) \quad \xrightarrow{\text{Treatment Effect}}$$

(1) 통계적 검정력 (고정된 ATE  $\delta$ 가 주어졌을 때 영가설을 기각할 확률) (2)  $\delta$ 의 분포

->  $n$ 을 늘려 통계적 검정력을 향상시키거나, 지표의 분산을 낮출 수 있는 추정기로 변경

->  $\delta$  자체의 문제일 수도 있음



$$\text{Power} := 1 - \beta = P(H_0 \text{ 기각} | H_0 \text{ 거짓}) = P\left(\frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}} > z_\alpha | \mu = \mu_1\right)$$

$$= P\left(\frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}} - \frac{\mu}{\sigma/\sqrt{n}} > z_\alpha - \frac{\mu}{\sigma/\sqrt{n}} | \mu = \mu_1\right)$$

$$= P\left(\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} > z_\alpha - \frac{\mu - \mu_0}{\sigma/\sqrt{n}} | \mu = \mu_1\right)$$

$$= P(Z > z_\alpha - \frac{\delta}{\sigma/\sqrt{n}})$$

$$= \int_{z_\alpha - \frac{\delta}{\sigma/\sqrt{n}}}^{\infty} f_Z(z) dz$$

효과 크기 ( $\delta$ ):  $\delta = \mu_t - \mu_c$   
 표준 오차 ( $\sigma$ )

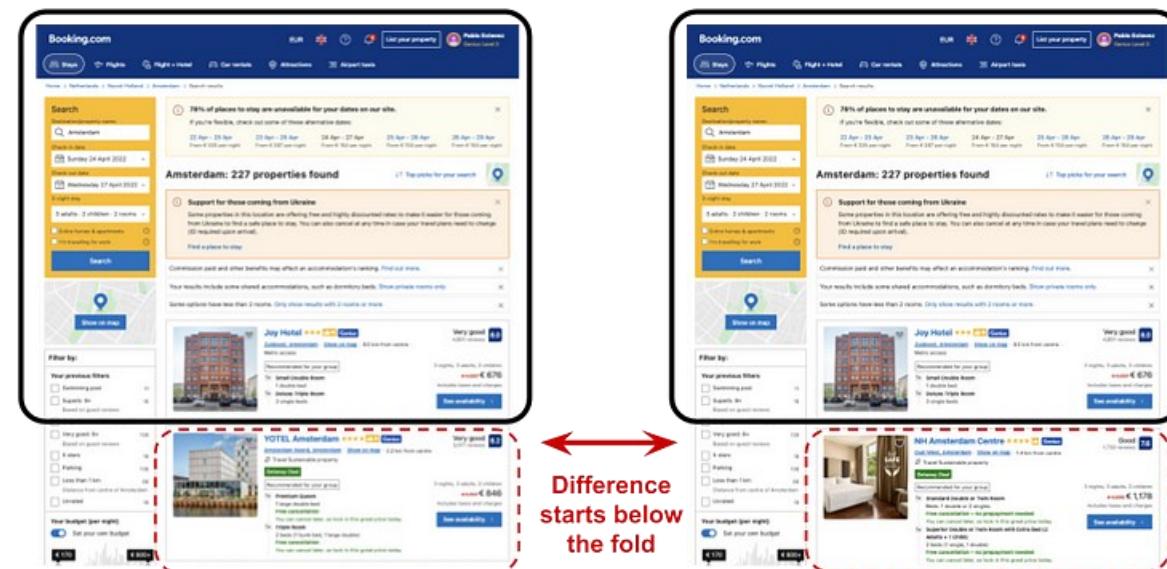
유의 수준 ( $\alpha$ ): 0.05 업계 표준  
 표본 크기 ( $n$ )

1.  $n$ 을 늘리면, 검정력 증가
2. 지표의  $\sigma$ (표준편차)를 줄이면 검정력 증가
3.  $\delta$  자체가 0에 가까움 -> 검정력이 낮아짐

-> 더 작은 효과도 통계적으로 유의미하게 검출

# Analytics Challenges: Conditional Triggering

- A/B 테스트의 맥락에서 트리거는 일반적으로 사용자가 테스트의 특정 variant에 포함되도록 하는 조건 예를 들어, 결제 페이지 혹은 상세 페이지에 접속한 사용자를 트리거 - 특정 기능과 상호작용한 사용자
- (아래 그림) 하단 페이지에 도달한 사용자를 처리 그룹과 컨트롤 그룹으로 할당할 수 있지만, 스크롤을 내려야 Variant를 볼 수 있음
  - > 스크롤을 충분히 내린 사용자만이 treatable -> 일부 처리 그룹의 사용자는 처리할 수 없음
  - > 스크롤 하지 않는 사용자를 실험에 포함 시키면 오버트래킹 현상 (**ATE가 다 0이야!** -> 데이터 분산 증가! -> 통계적 유의성을 낮춤)
- We could also conduct “trigger analysis”, which achieves the same **result by ignoring in the analysis phase those users who did not get treated** (Deng and Hu, 2015)



## Analytics Challenges: Conditional Triggering

- 사용자가 결제 페이지를 방문하는 순간 트리거! 반면에, 사용자가 결제 페이지를 방문하지 않으면 사용자가 트리거되지 않음
- 처치와 컨트롤 그룹에 대한 할당이 완전히 무작위인 경우 트리거링 비율이 처치그룹과 컨트롤 그룹 전체에서 동일할 것으로 예상

|                    | Treatment | Control |
|--------------------|-----------|---------|
| Trigger-Complement | T0        | C0      |
| Triggered          | T1        | C1      |

트리거링 비율이 낮을 경우 일반적으로 전체 분석 ( $T = T_0 + T_1$ 과  $C = C_0 + C_1$  비교)  
트리거된 분석( $T_1$ 과  $C_1$  비교)을 수행하는 것이 좋음

> 전체 사용자 집단을 분석에 사용하면 결과가 희석(diluted) (Kohavi, 2009)

당근마켓 사례>

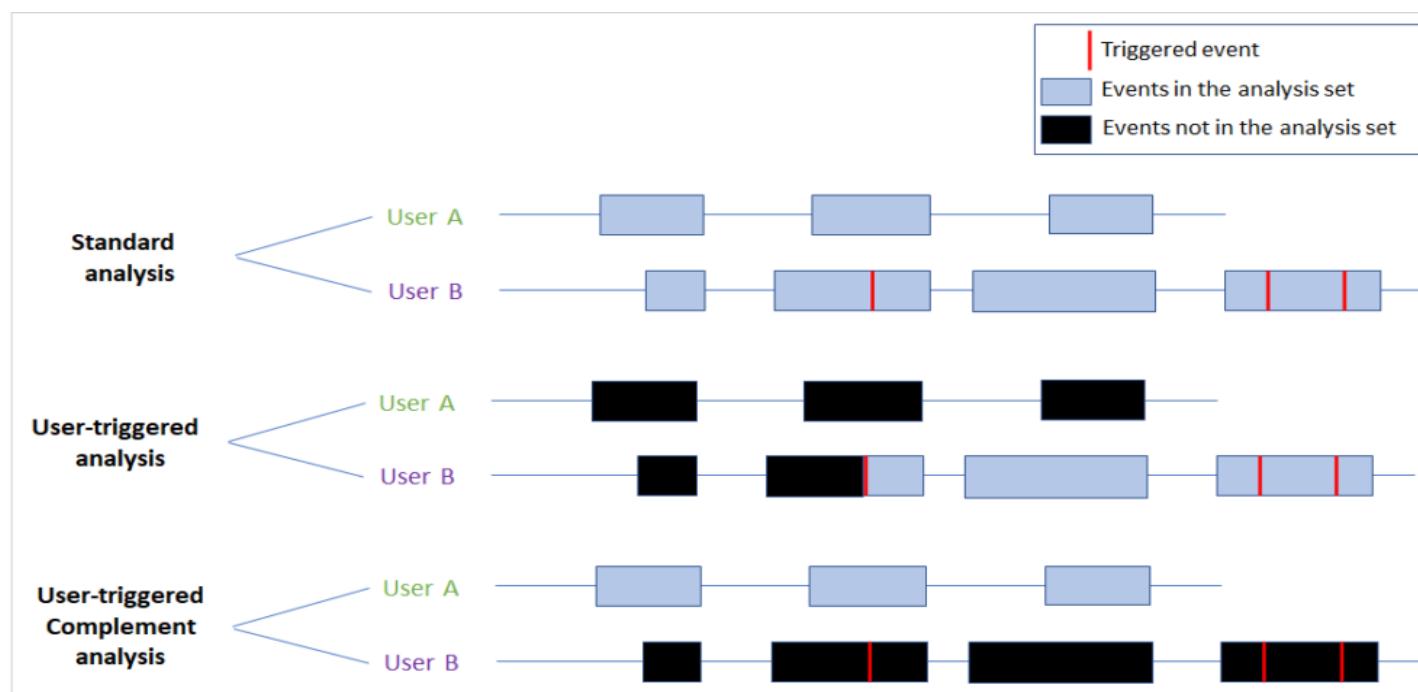
만약 전체 당근마켓 방문 사용자 수가 1000만 명이고 동네생활 방문 사용자가 200만 명이고 진행한 실험이 어떠한 지표에 0.1만큼 개선했다고 해볼게요.  
만약에 실험 결과를 구할 때 동네생활 방문 사용자만 대상으로 분석하지 않고 전체에 대해서 분석한다면(안하는 사람이 많으면), 0.1의 개선은 0.02의 개선으로 축소되어서 나타날 거예요. 이렇게 되면 차이가 있음에도 불구하고 없다고 판단할 가능성이 커져요. 당근마켓같이 여러 서비스가 공존하는 서비스에서는 이렇게 실질적으로 실험에 참여한 사용자들 대상으로만 결과 분석을 하는 것이 중요해요.

# Analytics Challenges: Conditional Triggering

목적: 사용자 세분화를 통해 측정항목의 민감도를 높이고 처치 그룹의 모집단에서 직접 측정된 작은 변화를 감지

> 트리거 분석은 트리거 조건을 충족하지 못한 사용자를 제외하여 영향을 받은 사용자에 중점

> 트리거된 분석에서는 트리거 조건이 충족된 첫 번째 이벤트 이전에 발생한 모든 사용자 로그를 추가로 제외



표준 분석에는 트리거된 이벤트와 관계없이 모든 사용자가 포함

트리거된 이벤트가 있는 사용자 B와 같은 사용자만 포함되며 처음 트리거된 이벤트 이후의 이벤트만 포함

User-triggered complement 분석은 처치 효과를 감지하지 않아야 함

## Analytics Challenges: Conditional Triggering

**Motivating Example:** Suppose engineers are testing a change made on an e-commerce website's checkout page. **Users in the experiment who never interact with this checkout page are not impacted by the experiment and so their treatment effect is zero. Many such users will increase noise and dilute the treatment effect. Sensitivity may be increased by analyzing only the users who could have been impacted by the experiment; those that were triggered into the analysis.** Although this reduces sample size, the treatment effect among the triggered users is undiluted and therefore higher and easier to detect.

Let  $\Omega$  be the overall user population and  $\Theta \subset \Omega$  the population of users who could be affected by the treatment. A **given user is determined to belong to  $\Theta$  via techniques such as conditional checks or counterfactual logging** (Kohavi et al., 2020; Deng et al., 2023). **If  $\Theta$  comprises only a modest fraction of  $\Omega$ , (i.e.,  $|\Theta|/|\Omega| \leq 0.2$ , for instance), an experiment that samples data from the entire population could be severely under-powered, particularly when effect sizes are small (Kohavi et al., 2009)**

The most common analysis technique is the user-trigger analysis, which incorporates all events beginning with the first event where i triggered. **Such analyses are quite popular as they do not require any assumptions regarding the treatment effect, and are amenable to common user-level metrics.**

However,  $\tau\Theta$  is typically larger than the population-level  $\tau\Omega$  and the corresponding estimator generally has greater variance

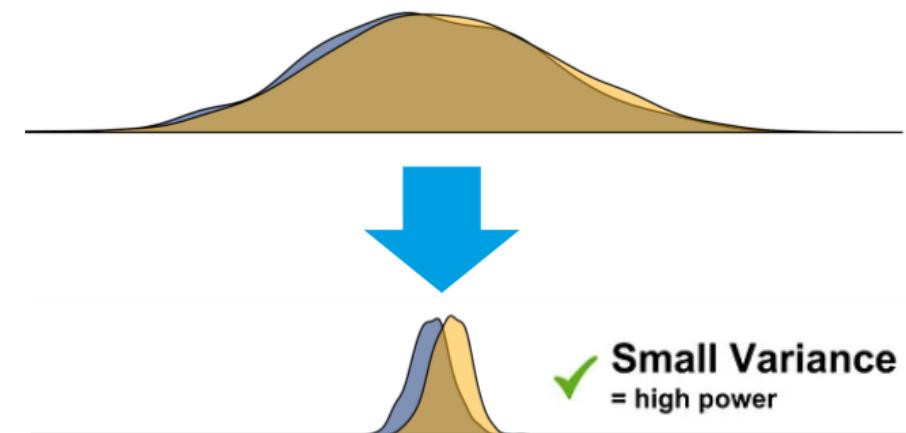
## Analytics Challenges: Variance Reduction

A/B 테스트에서 효율적인 테스트를 위해서 측정값의 Variance를 Reduction -> 검정력 증가

- 지표 자체를 변경(극단값 처리, 비율로 변경)하는 것이 어려운 경우, CUPED기법을 활용하여 실험 결과의 Variance를 줄일 수 있음
  - 특히 극단적인 지표일 경우 Variance가 큼 (숙박 시설 당 일일 예약 건수)
    - 전제: 실험 전 숙박 시설의 일일 평균 예약 건수를 알고 있음
- > 실험 전 데이터의 변동은 실험의 효과와 관련이 없으므로 제거

CUPED-adjusted metric =  
 metric - (covariate - mean(covariate)) x **theta**

- (1) **metric**은 실험 후 측정된 결과값
- (2) **covariate**은 실험 전 측정된 연속형 지표
- (3) **mean(covariate)**은 모든 단위에 대한 공변량의 평균값
- (4) **theta** (세타)는 공변량과 메트릭 사이의 관계를 나타내는 회귀계수  
 $\theta = \text{covariance}(\text{metric}, \text{covariate}) / \text{variance}(\text{covariate})$



**표준: 공변량을 측정하는 지표와 동일한 실험 전 측정항목 활용 (Reduction 효과는 상관관계에 비례)**

예: 측정하는 지표가 숙박 시설 당 일일 예약 건수의 차이라면, Covariate도 동일한 측정항목 활용

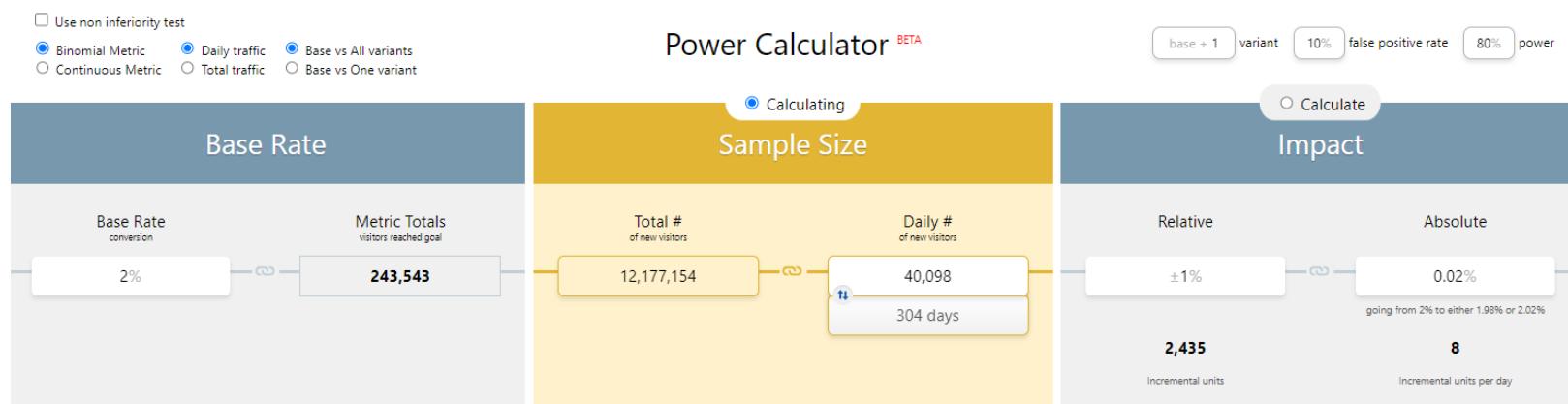
# Analytics Challenges: Variance Reduction

## (1) OCE와 RCT의 차이점

- 전통적인 RCT는 일반적으로 OCE와 다르게 제한적이고 작은 n을 활용하지만, OCE는 수백, 수천의 사용자를 활용하기 때문에 복잡함
- 사용자간의 이질성이 최소화된 RCT와는 다르게, OCE는 다양한 구매 패턴을 가진 사용자가 함께 존재하여 변형이 많음
- 온라인 서비스 및 제품에는 연휴 판매 급증, 계절적 변경, 특별 행사, 소셜 미디어의 영향이 많음
- 처치 효과인지, 노이즈인지 구분하기 어려움

## (2) 실험을 더 오랫동안 진행하고, 더 많은 사용자를 활용하는 것의 한계

- 실험의 민감도는 n의 제곱에 반비례 Delta가 10배 변경되면 n이 100배 변경되어야 함(Deng et al., 2013)
- 작은 효과를 감지하는 것은 어려운 일(Booking.com) -> 높은 분산에 비해서 큰 효과를 가지기 위해서는 더 큰 Effect가 필요
- 일반적으로 전환율이 2%인 전자상거래 웹사이트에서 1%의 상대적 전환율 변화를 감지하려면 1,200만명이 넘는 사용자가 대상으로 필요
- (핵심) 적은 사용자로 더 많은 실험을 할 수 있게 해주며, 이를 통해 동일 사용자로 100개 실험할거, 1000개 할 수 있음**



# Analytics Challenges: Variance Reduction

일반 사용자에게 제공되는 할인의 변화가 주당 평균 예약을 증가시키는 것으로 가정되는지 여부를 테스트하는 실험

|     | user_id | group | bookings-per-week<br>(metric) | bookings-per-week<br>(pre-experiment covariate) | CUPED-adjusted metric                           |
|-----|---------|-------|-------------------------------|---|---|
| 1   | base    |       | 2                             | 2   | $2 - (2 - \text{covariate mean}) \times \theta$ |
| 2   | base    |       | 3                             | 2   | $3 - (2 - \text{covariate mean}) \times \theta$ |
| 3   | variant |       | 5                             | 4   | $5 - (4 - \text{covariate mean}) \times \theta$ |
| 4   | variant |       | 5                             | 6   | $5 - (6 - \text{covariate mean}) \times \theta$ |
| ... | ...     | ...   | ...                           | ...   | ...   |

their bookings-per-week before and after being exposed to the experiment

**Question.** 실험 동안 로그인하지 않은 사용자나 숙소 데이터 없으면? 즉, covariate가 없으면?

> 기존 Metrics으로 활용

|     | exp_id | unit_id | grp | metric | covariate |
|-----|--------|---------|-----|--------|-----------|
| 1   | 1      | base    | 7   | 6      |           |
| 1   | 2      | variant | 12  | NULL   |           |
| 1   | 3      | variant | 11  | 7      |           |
| 2   | 1      | base    | 8   | 10     |           |
| 2   | 2      | base    | 6   | 5      |           |
| 2   | 3      | variant | 9   | NULL   |           |
| ... | ...    | ...     | ... | ...    | ...       |

```

covariate_mean = mean(covariate)
theta = covariance(metric, covariate) / variance(covariate)

for each unit:
    if covariate is missing:
        cuped_metric = metric
    else:
        cuped_metric = metric - (covariate - covariate_mean) x theta
  
```

# Analytics Challenges: Variance Reduction

인과 문제: 새로운 달력 인터페이스를 통해 객실 숙박을 추가함으로써 이익을 얻을 수 있을까?

> 6주간 실험 진행

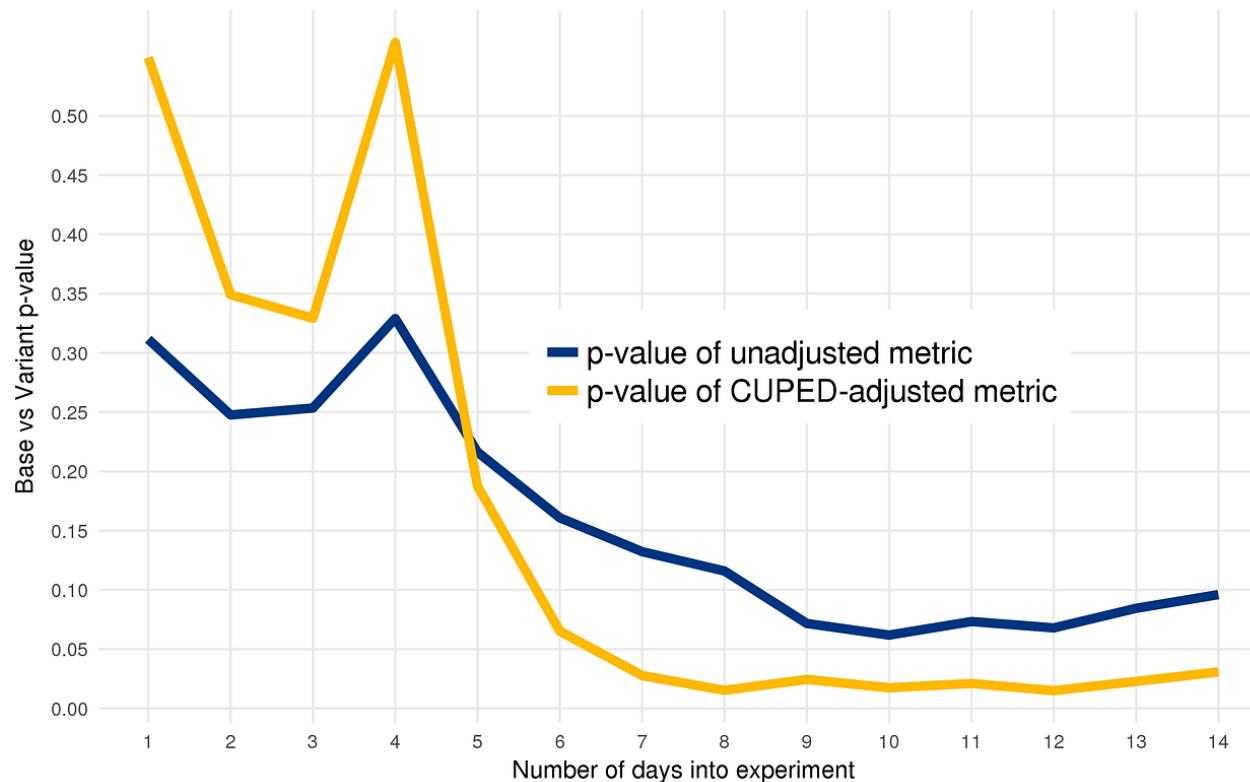
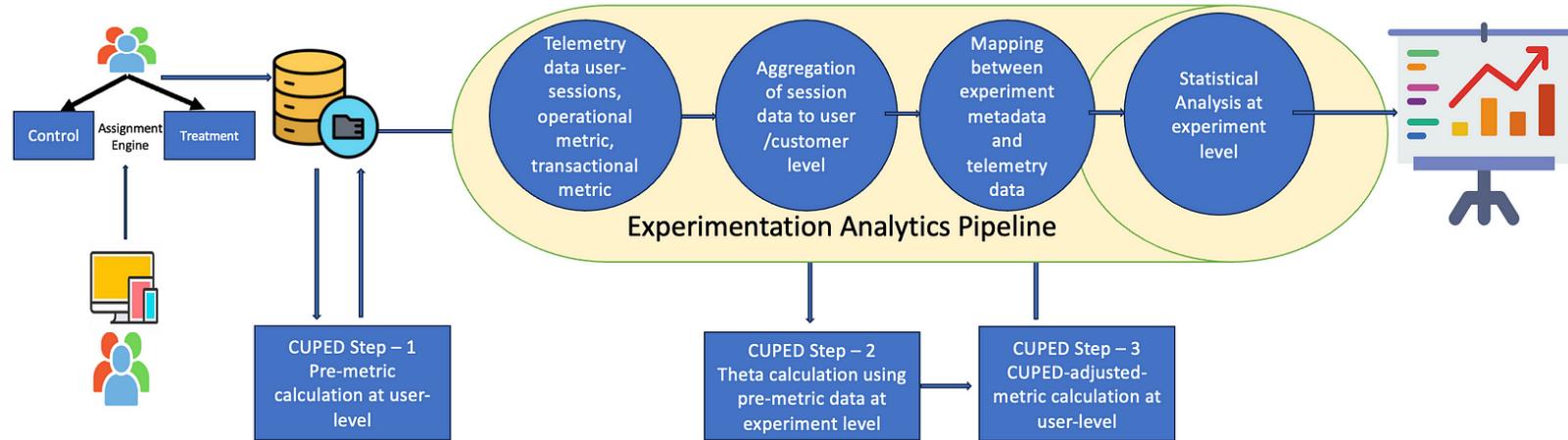


Figure 4. Comparison of experiment results with and without CUPED adjustment for low-traffic experiment

***For this low-traffic experiment, CUPED-adjustment helped detect a clearly significant result sooner and with a smaller sample than with no adjustment***

# Analytics Challenges: Variance Reduction



## 목표 지표: 수익(revenue)

(Step 0) A/B Assignment

(Step 1) 사용자 수준으로 지난 28일간의 수익 합계를 구하기 : Pre\_revenue

(Step 2) 실험 수준으로 Pre\_revenue의 Sum, mean, standard deviation, variance 등 집계(aggregated)

(Step 2-1) 수익과 Pre\_revenue의 공분산을 계산하여 theta 계산

(Step 3) CUPED adjusted revenue 계산

(Step 4) Adjusted된 지표를 바탕으로 통계 분석

# Analytics Challenges: Novelty & Primary effect

새 기능이 도입될 때, 해당 기능이 Primary effect 또는 Novelty Effect를 유발할 수 있는 위험이 존재

## 초두 효과 (Primacy Effect)

- 사용자들이 기존 제품/방식에 익숙하고 변화를 꺼려하여 발생하는 효과

## 신기 효과 (Novelty Effect)

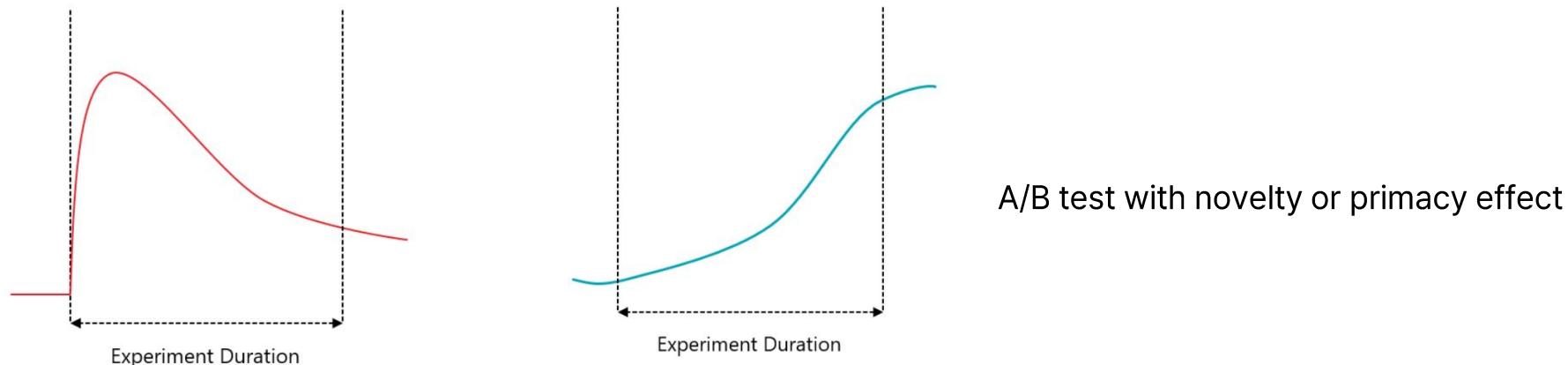
- 변화를 좋아하고 기존 제품/방식보다 새로운 기능을 선호하여 발생하는 효과

추가된 Feature가 웹 사이트를 탐색하는 새로운 방법인 경우 사용자들이 이 새로운 방법에 익숙해지는 데 시간이 걸림

-> 초기 결과는 부정적인 사용자 경험을 나타낼 수 있지만, 결과가 시간이 지남에 따라 변할 수 있음

처치 효과가 언제 안정화되는지 확인하려면 (1) 실험을 오래 실행 (2) Novelty & Primary에 영향을 받지 않는 신규 사용자를 실험

-> 신규 사용자는 제품을 처음 사용하기 때문에, 두 효과에 영향을 받지 않음 / 제품 사용 기간에 따라 가중치를 줄 수도 있음



## Analytics Challenges: etc.

아직 다루지 않은 주제 – 추가 업데이트 필요:

- (1) Multiple Comparison : BH Correction
- (2) Long Term Effect
- (3) Sequential Testing
- (4) Quantile Testing

# Analytics 종합: Stats Engine

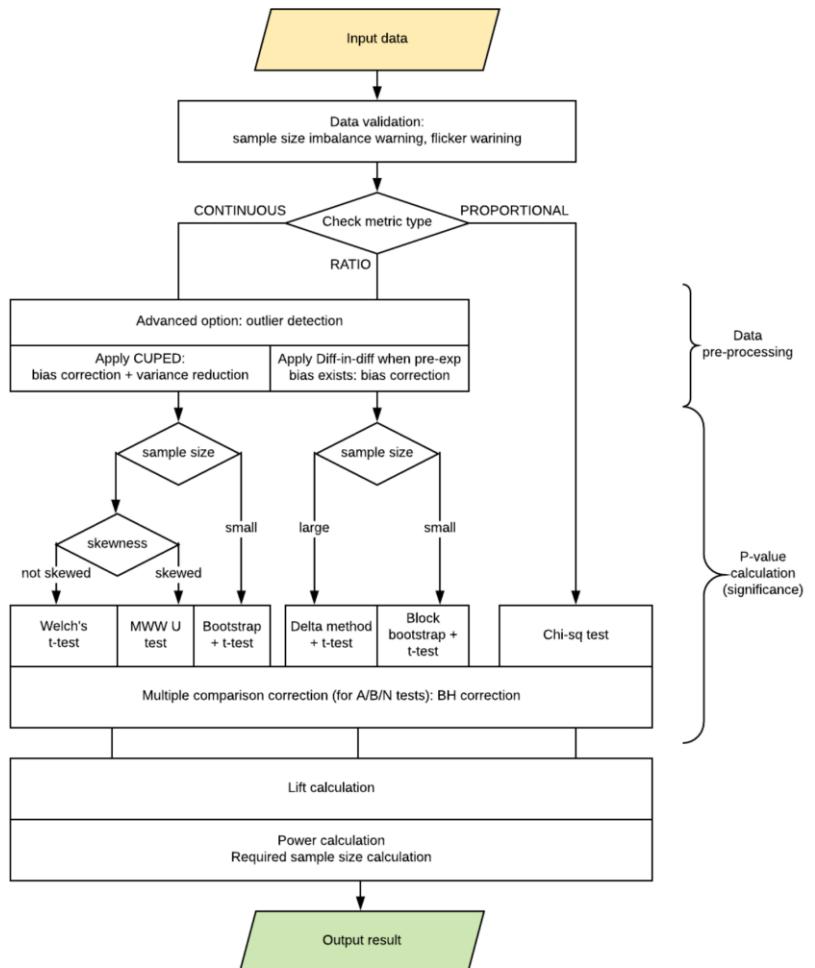


Figure 5: Uber's statistics engine is used for A/B/N experiments and dictated by fixed horizon hypothesis testing methodologies.

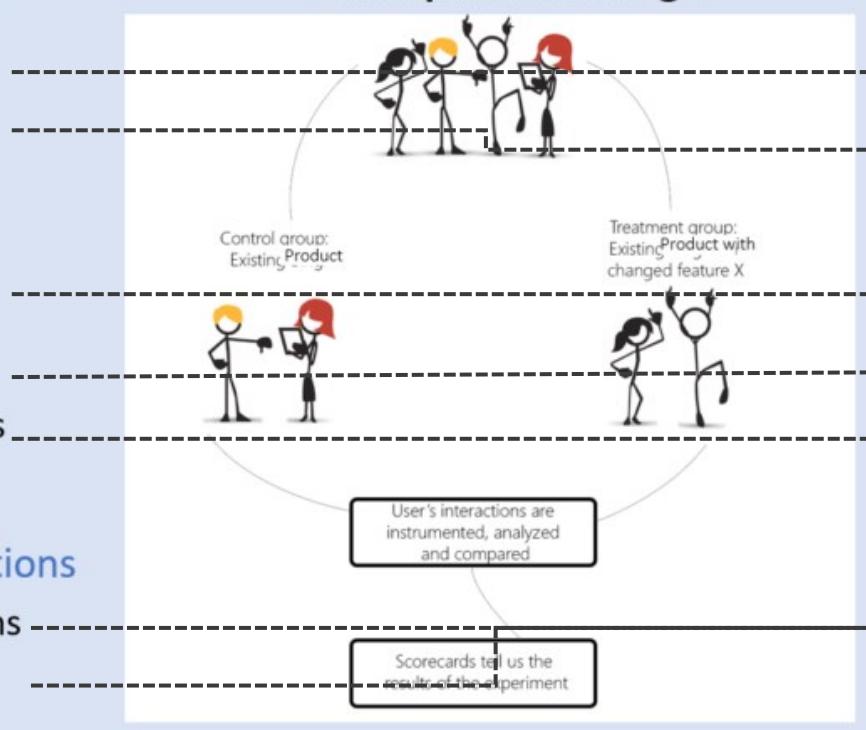
### 3. 실험 패턴 - 마이크로소프트

# Pre-Experiment Stage

## Forming a Hypothesis and Selecting Users

- Formulate your hypothesis and Success Metrics
- Choose the Appropriate Unit of Randomization
- Check and Account for Pre-Experiment Bias

## Trustworthy AB Experimentation: Pre-Experiment Stage



## Pre-Experiment Engineering Design Plan

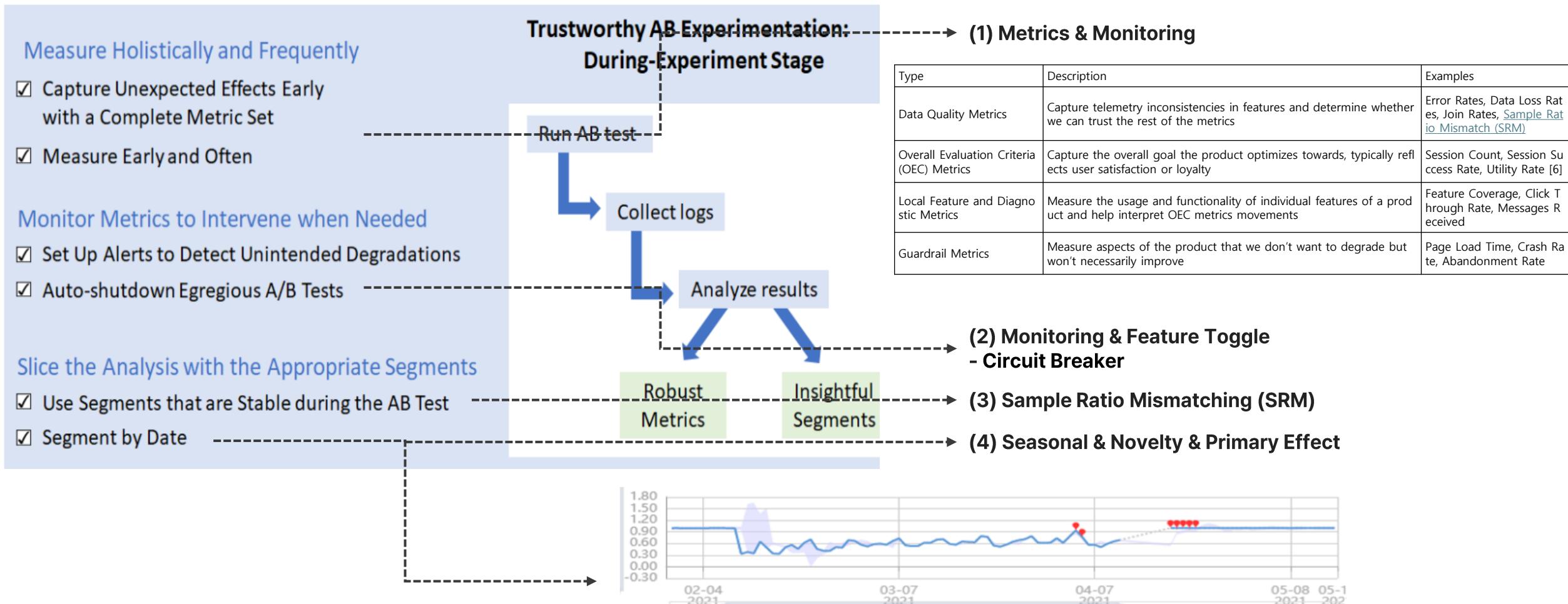
- Set Up Counterfactual Logging
- Have Custom Control and Standard Control
- Review Engineering Design Choices to Avoid Bias

## Pre-Validation by Progressing Through Populations

- Gradual Rollout Across Different User Populations
- Gradual Rollout within a User Population

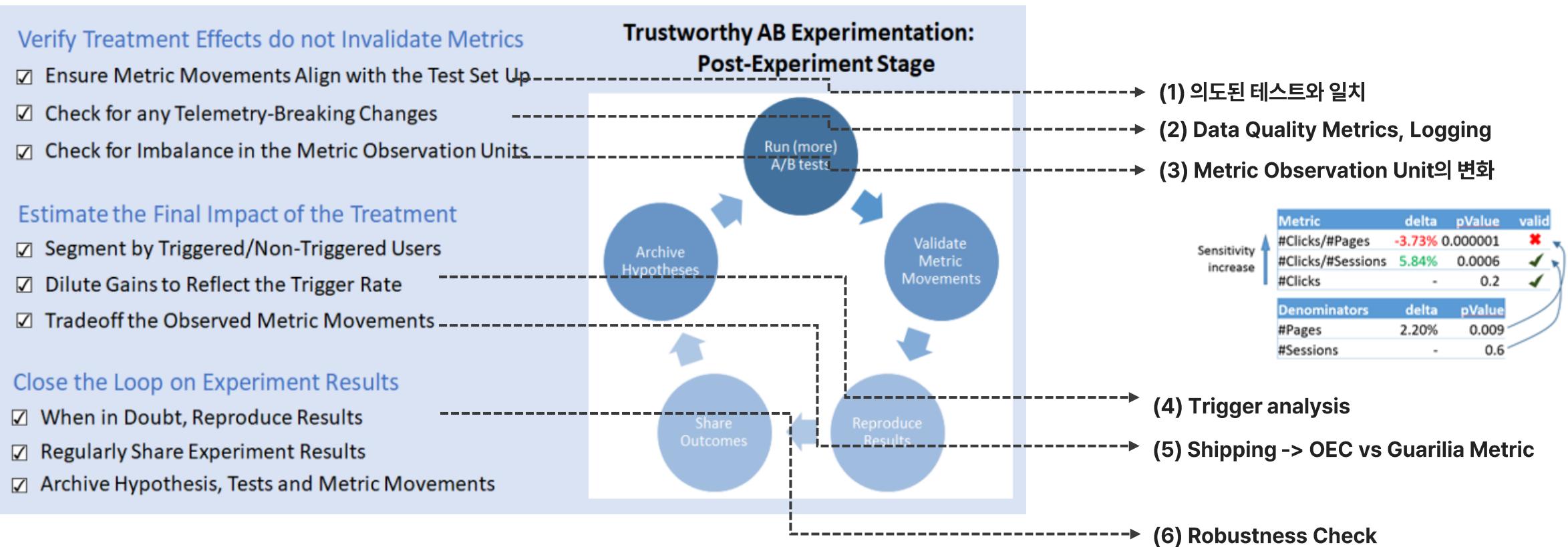
# During-Experiment Stage

실험이 진행되는 동안 인사이트를 얻기 위해 수행할 수 있는 분석과 신뢰할 수 있는 결론을 도출하기 위한 방법



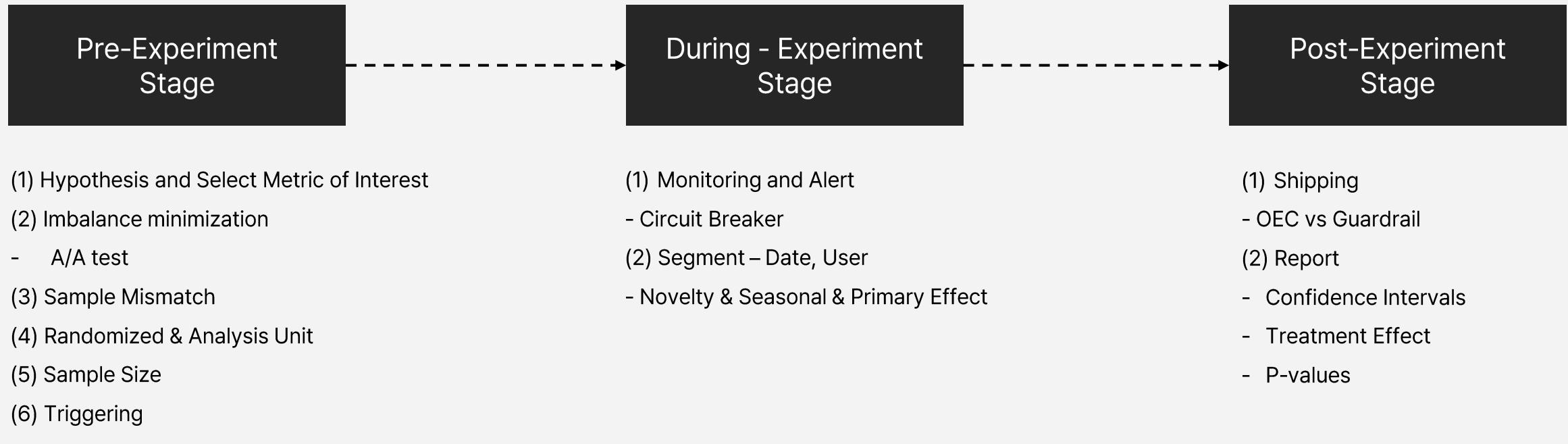
# Post-Experiment Stage

신뢰할 수 있는 결과를 보장하여 실험의 결과를 프로덕트에 반영하는 결정



## Experiment Platform

(1) Random Assignment (2) Release (3) Logging for Experiments





PseudoLab



Causal - Lab

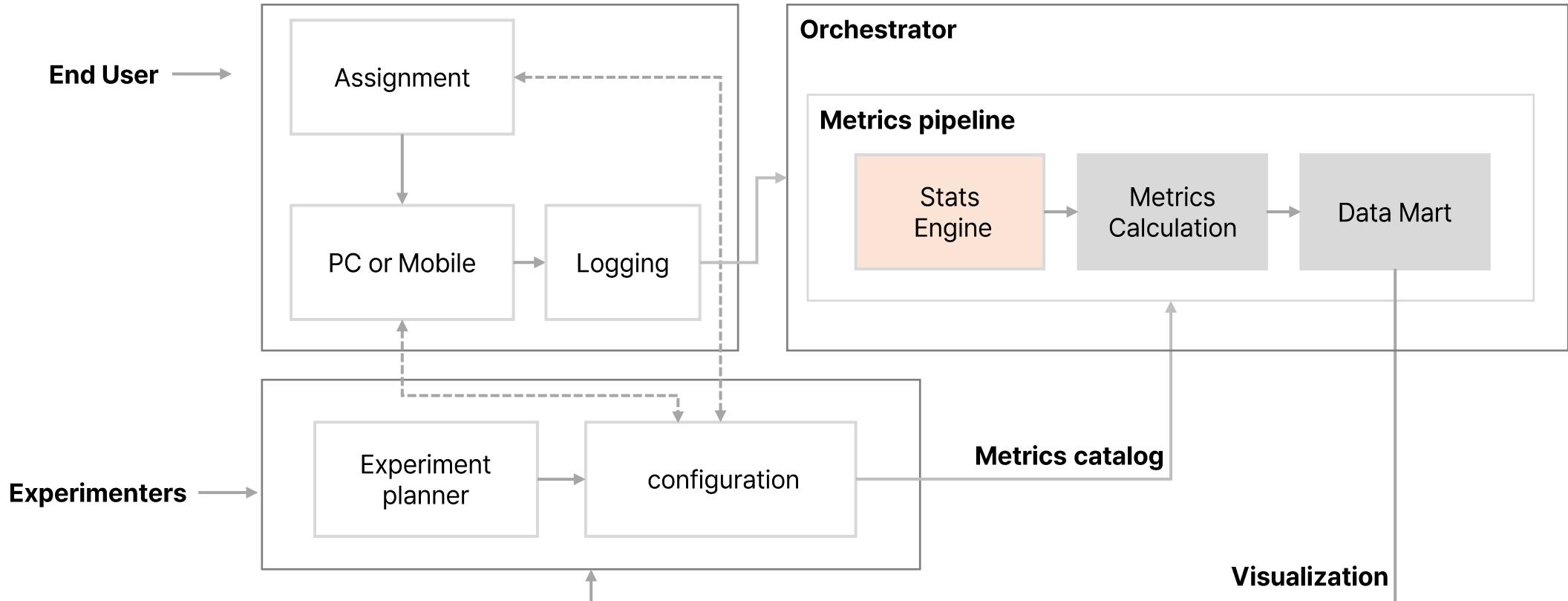
# 4. Wrap-up

# 온라인 실험의 난제와 실험 플랫폼 기능

| Trustworthy principle Name        | Platform feature Name  |
|-----------------------------------|--|
| Hypothesis given upfront          | Power analysis   |
| Correct sample size               | Power analysis   |
|                                   | A/A-test   |
| Reliable experiment users         | Interaction detection<br>Conditional triggering  |
|                                   | SRM-test   |
| Variation assignment monitoring   | Randomization algorithm<br>Accurate peeking  |
| Error detection                   | Missing data detection   |
|                                   | Near real-time alerting<br>Auto-shutdown   |
| Metric monitoring                 | Accurate peeking<br>Novelty and primacy effect detection<br>SRM-test                         |
|                                   | Outlier filtering  |
| High quality data                 | Conditional triggering<br>Missing data detection<br>Heterogeneous treatment effect detection |
|                                   | Alerting   |
| Presentation and feedback         | Ship recommendation<br>Analytic presentation   |
| Use of correct statistical method | Use correct statistical test   |

# 데이터 관점에서의 실험 플랫폼

1. 일관된 할당
2. 로그 적재
3. 데이터 집계, 통계 검정





Causal - Lab

감사합니다