A/B테스트 실무 정리하기

Chapter 1. 데이터 분석 직군에 대한 기본 이해

▼ **데이터 조직의 역할은 'Decision Science' 과 'Product Science' 이고, 성과는 '부가가치'의 창출이다.**

- `Decision Science`
  1. **Data Driven Decision** → 지금 하는 일을 더 잘할 수 없을까?

     > 과거 데이터를 기반으로 최적화된 의사결정을 돕는 분석을 하는 것

     - 인과추론의 꽃인 **A/B테스트가 대표적**이다.
       - 할 수 없는 것: 혁신이 필요하거나 / 새로운 기능을 만들거나
         - Why? : 과거 데이터가 없다 / 비교대상이 없다
       - 적용이 어려운 것: 가격 결정에 대한 의사결정
         - Why? : 가격에 대한 A/B 테스트는 윤리적 문제, 시장 왜곡, 고객 간 상호작용, 데이터 노이즈, 수익성 손실, 복잡한 가격 구조 등의 이유를 고려해야함
     - 전/후 비교 방법이나 각종 시계열 지표를 통한 추측은 통제되지 않았기 때문에 결과가 편향(bias)되거나 명확히 통제하여 측정할 수 없는 점이 존재함
  2. **Data Informed Decision** → 새로운 서비스를 어떻게 만들어볼까?

     > 가지고 있는 데이터를 참고해서 새로운 아이디어나 서비스를 만들기 위한 분석을 하는 것

     - 서비스/기능을 런칭하는 것이 타당할지를 결정하기 위함
       - 탐색적 데이터 분석(EDA), 통계적 분석 등의 다양한 분석 방법으로 아이디어를 얻기
       - 과거의 사례나 다른 경쟁사의 사례를 분석하거나, 비즈니스를 모델링하며 예측해보기
- `Product Science`

  > 데이터를 기반으로 UX를 개선하거나 서비스의 프로세스를 최적화하는 것

  1. **UX 개선**의 관점
     - Decision Science를 통해 UX를 개선할 수 있는 '근거'를 마련할 수 있음
       - How? : A/B테스트를 통해 기존 기능 개선안에 대한 사용자 반응을 추론할 수 있음
     - 실제 개선을 위한 기획/디자인/개발 단계에서 파생되는 협업체의 갈증 해소도 중요
       - What? : 이 서비스 퍼널이 다른 곳에도 유효할까요? 로그는 어떻게 심을까요?
  2. **프로세스 최적화**의 관점
     - 최근은 머신러닝 알고리즘을 통해 서비스 경험을 크게 개선하는 것이 일반적
       - How? : 글 작성이 어려울 때 판매 게시글 AI 글쓰기로 자동화시켜주기
     - 개인화 가능한 추천 알고리즘 등을 통해 프로세스를 최적화함으로써 UX를 개선
       - How? : 헤메기 전에 나에게 딱 맞는 보험 상품이나 대출을 추천해주기

▼ **데이터 업무의 흐름은 데이터 수집 → ETL로 웨어하우스 적재 → ELT로 분석용 데이터 생성 → 분석과 모델링이다.**

1. **서비스에서 발생하는 직접 데이터와 외부 업체 등을 통해 발생하는 간접 데이터를 수집**

   > 주로 백엔드 엔지니어가 직접/간접 데이터를 수집하고, 데이터 웨어하우스에 연동 가능하도록 작업을 진행

   - **직접 데이터 (1st Party Data)**
     - **자바스크립트**를 사용해 웹사이트에서 발생하는 사용자 활동(버튼 클릭, 동영상 시청 등)을 추적
     - **쿠키**를 통해 사용자의 방문 이력 및 행동 데이터를 저장
     - 서버에 저장된 기록인 **로그 파일**로, IP 주소나 방문 시간을 기록하고
     - **세션 추적**을 통해 사용자가 사이트에서 어떤 경로로 이동하는지 파악

- **간접 데이터 (3nd Party Data)**
  - 외부 업체나 플랫폼(광고 네트워크, 소셜 미디어 등)에서 수집된 사용자 데이터를 가져오는 방식
  - **해당 업체 API**나 **태그 관리자** 등을 통해 연동해 수집
    - Appsflyer: 앱 설치, 사용자 세그먼트, 광고 캠페인 효과 추적
    - Google Ads: 광고 네트워크에서 수집된 클릭 데이터, 광고 조회수, 사용자 행동 분석

2. **앞서 수집한 직접/간접 데이터를 ETL 프로세스를 통해 데이터 웨어하우스에 적재**

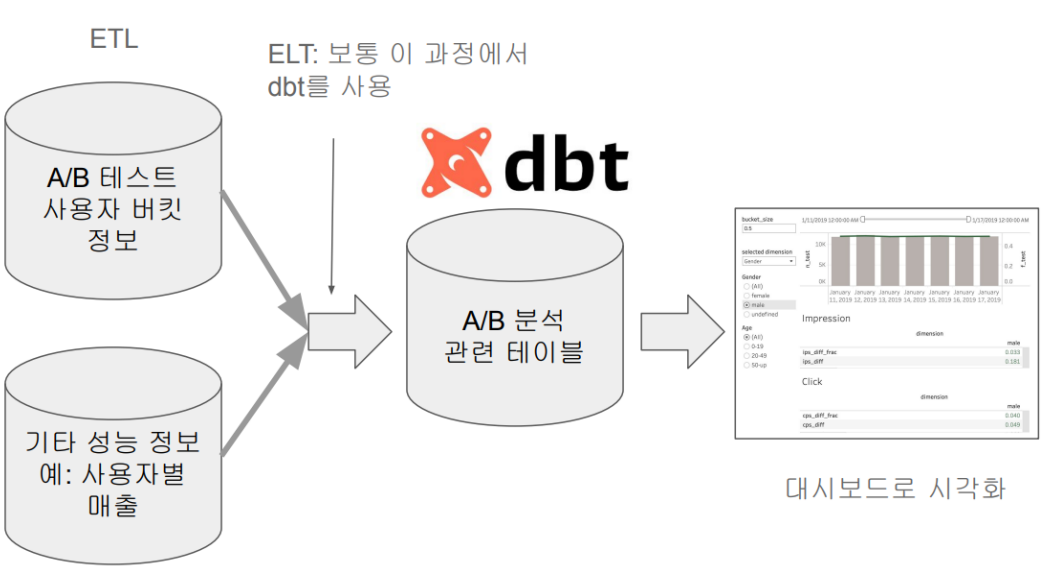
> 주로 데이터 엔지니어가, 수집한 대용량의 직접/간접 데이터를 빅데이터 처리 엔진을 활용해 정제하고 Airflow와 같은 스케줄링 도구로 정해진 시간마다 데이터 웨어하우스에 적재하도록 작업을 진행

- **ETL 프로세스**란? → 데이터를 추출(Extract), 변환(Transform), 적재(Load)하는 과정을 요약한 단어
  - **Airflow**라는 데이터 워크플로우 관리 도구로 Python, SQL등으로 이뤄진 코드를 ETL프로세스에 맞게 구현하여 스케줄링 함
- **대용량 데이터를 어떻게 전처리하고 정제하지?** → 하둡에서 파생된 '빅데이터 분산처리 엔진'을 이용
  - **Apache Spark**와 같은 빅데이터 분산 처리 엔진으로 빠르게 처리하는 코드를 구현
  - 초기에 Hadoop이라는 분산처리 시스템에서 맵리듀스 알고리즘으로 데이터를 분산처리함
- **데이터 웨어하우스**는 뭐지?
  - 대량의 데이터를 통합하고 분석하기 위한 중앙 저장소로, 여러 소스(웹사이트, CRM, 서드파티 등)에서 수집된 데이터를 한곳에 모아, 분석 및 비즈니스 인사이트 도출에 사용
  - Amazon Redshift, BigQuery, Snowflake, Databricks와 같은 것이 대표적 데이터 웨어하우스 플랫폼

3. 데이터 웨어하우스에 적재된 데이터를 가지고 DBT와 같은 ELT 프로세스 구현 도구를 통해 분석에 사용하기 쉽고 의미가 담긴 테이블을 생성

> 주로 데이터 분석가가, 데이터 웨어하우스의 데이터를 비즈니스나 서비스의 의미가 담긴 2차 분석용 데이터를 새롭게 정제하고 적재하는 작업을 진행

- **ELT 프로세스**란? → ETL프로세스에서 추출 및 적재를 먼저한 뒤, 마지막에 변환을 하는 차이가 있음
  - Why? : 분석가가 데이터 웨어하우스에서 원시 데이터를 추출하여 적재한 뒤, 분석에 적합한 테이블을 생성하기 위해 다양한 조작/변환을 하는 작업 구조이기 때문
- **DBT**란? → ELT의 'T'를 맡고 있는 데이터 변환 작업 관리 도구로, 'Data Build Tool'의 준말
  1. 원시 데이터 적재: 데이터 웨어하우스에 추출된 원시 데이터를 적재
  2. 변환 작업: DBT를 통해 SQL 쿼리를 사용해 원시 데이터를 변환(클렌징, 집계 등)하여 분석용 테이블을 생성
  3. 모델 관리: 데이터 모델을 관리하고 문서화하여, 팀 간 협업이 용이
  4. 자동화: 데이터 변환 파이프라인을 자동화하고, 스케줄링을 통해 정기적인 업데이트를 지원



4. 데이터 수집 → 데이터 웨어하우스 적재 → ELT과정을 통한 분석용 데이터를 가지고 지표를 정의하고 시각화하며 데이터를 분석

> 이 과정에 있는 데이터를 다룰 줄 알고, 데이터 안에 숨겨진 의미를 잘 이해할 수 있으며, 이를 통해 만들어지는 지표나 비즈니스 데이터에 대한 이해가 높을수록 데이터 리터러시가 높은 조직

- 앞서, 데이터 분석 조직의 가장 큰 범주인 Decision Science, Product Science를 하기 위해서는 이 단계 이전의 작업들이 모두 완료되어야 하며, 데이터 품질에서 신뢰도가 있어야 분석이 의미가 있어짐
- 분석가가 분석용 데이터를 어떻게 만드느냐에 따라 3차 가공 및 처리가 필요한 경우가 있고, 그대로 시각화를 위해 쉬운 집계만 할수도 있음

    - Tip1 : 데이터 웨어하우스 플랫폼에서 SQL, Python, R 등의 언어로 데이터 핸들링이 가능하도록 잘 마련해놓고 있어서 별도의 툴이나 플랫폼을 사용할 필요 없음
    - Tip2 : Tableau, Power BI, Superset, Looker Studio 등 시각화 툴에서도 단순한 수준의 집계를 지원하며, 시각화 툴마다 직접 SQL을 통해 디테일한 조작이 가능하거나 불가능한 경우가 있음
    - Tip3 : 현재 모든 대시보드/시각화 툴은 내부 로직에서 SQL 쿼리문을 날려서 재집계하는 방식을 거치고 있으므로 SQL을 잘할수록 시각화되는 데이터 집계의 품질에도 관여할 수 있음
5. 신뢰할만한 데이터를 갖고 다양하게 분석하여 나온 Insight를 의사결정에 반영하거나 데이터 과학에 적용해서 개인화된 사용자 경험 개선 및 최적화를 진행함
    - 00 서비스 런칭에 대한 파급효과와 예상 매출 분석
    - 00 기능에 대한 UX/UI 개선에 대한 A/B테스트
    - VIP유저와 일반 유저의 00제품군 결제 동향과 사용자 경험 차이 분석
    - 추천 알고리즘의 성능 개선폭에 따른 UX 개선효과 비교
    - 기타 등등...

Chapter 2. Data Driven Decision의 꽃, A/B테스트 개론

▼ **A/B테스트가 무엇일까?** → 객관적으로 사용자 그룹을 비교하는 과학적 방법

> 실제 사용자를 대상으로, 새로 만든 기능을 보여주고, 기존 기능과 비교하는 테스트

- 의료 분야에서 주로 사용하던 무작위 통제 실험을 온라인으로 진행하는 것
    - 동일한 담배 상품에 대한 특정 기간 동안 흡연자와 비흡연자의 사망률에 대해...
- 반드시 가설이 필요하며, 다수의 Variant(변인)을 비교하는 실험
    - 하나의 대조군(Control)과 하나 이상의 실험군(Treatment)를 설정함
    - 모든 Variant가 표본 불균형이 되지 않도록 **표본 크기가 50:50으로 설정**되어야 함
- A/B테스트의 직관적인 실험 흐름
    1. 테스트에 공통으로 해당하는 수준의 전체 표본을 수집한다.
    2. 표본 안에서 기존 대상인 대조군과 비교 대상인 실험군을 50:50으로 공평하게 나눈다.
    3. 실험 결과를 해석/분석할 측정 지표와 실험 기간 등을 설정한다.
    4. 실험을 진행해서 측정 지표에 차이가 있는지 확인하고, 이를 통계적으로 유의미한지 검정한다.

▼ **A/B테스트를 하는 이유가 뭘까?** → '객관성' 확보와 '위험' 최소화

- 비즈니스 관련 지표가 개선되는지 '객관적'으로 측정하기 위함
    - 가설을 기반으로 사용자를 나누어 객관적으로 새로운 기능이나 변경을 측정/비교하는 방식이기 때문이다.
- 다양한 왜곡(Bias)이나 위험(Risk)을 최소화하기 위함
    - 이미 기능이나 서비스를 런칭한 뒤 이전 상황과 비교하는 것은 객관적이지 않음
    - 아무리 사용자 설문 등이 좋아도 실제 사용자 반응에 대해선 명확히 알 수 없음
    - 머신러닝 모델의 경우 테스트 데이터로만 모델 성능 개선을 측정하는 것에 한계가 있음
    - 처음에 작은 퍼센트의 사용자들에게만 새 기능을 노출시켜보고 문제가 없다면 퍼센트를 증가시키면서 서비스에 미칠 부정 영향을 최소화할 수 있음

▼ **A/B테스트를 하기 위해 반드시 필요한 조건** → '명확하고 구체적인 가설 설정'

1. 가설이 없는 A/B테스트는 불가능하다.
    - A/B테스트는 우리가 궁금해했던 가설을 실험하고 검증하는 것이기 때문
    - A/B테스트는 통계학의 가설검정을 확인하는 방법을 기반하므로 가설이 반드시 존재해야함
    - 가설이란 것의 구체적인 예시
        - 새로운 기능 추천방식이 기존보다 매출을 증대시키는가?
        - 상품 선택부터 결제까지의 Step을 줄이면 결제 전환율이 더 증가하는가?
        - 결제 전 본인인증 기능을 도입하면 어뷰저가 감소하는가? 또는 사용자 이탈률에 영향이 없는가?
2. 가설은 반드시 명확하고 구체적이어야 한다.
    - 어떻게 실험의 성공과 실패를 결정할 것인가?
        - **측정 지표에 대한 첨예하고 밀도있는 설계가 필요**하다. 어떻게, 무엇을, 어떤 방식으로!
    - 가설을 명확하지 않게 설정하면 안된다. 실험 결과에 따라 말을 바꾸며 결과론적으로 해석하기 시작한다.

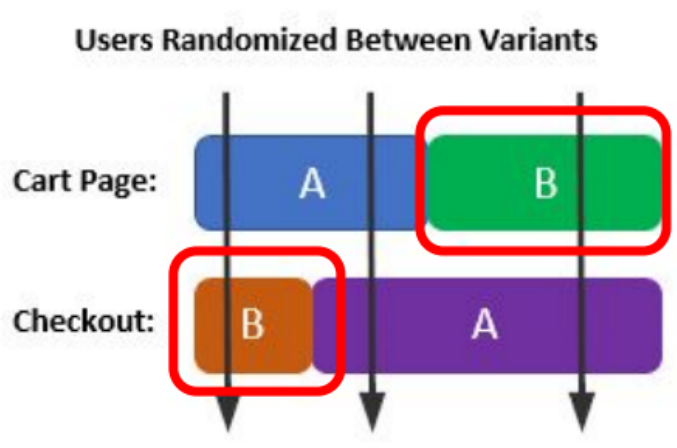
    - 상품 결제 퍼널의 단계를 줄이면 당연히 매출이 올라가지 않을까?
    - → 오히려 가격이 비싼 물건은 단계가 많을수록 매출이 증가했고, 가격이 저렴한 물건은 단계가 적을수록 매출이 증가한다는 레슨런을 배움
    - → 분석가가 디테일하게 후속분석을 하지 않는 경우, 또는 후속분석의 결과도 확연하게 실패한 실험임에도 불구하고 실험을 제안한 사람이 가설대로 결과가 나오지 않는 경우 좋은 쪽으로 결과를 끼워맞추려 함

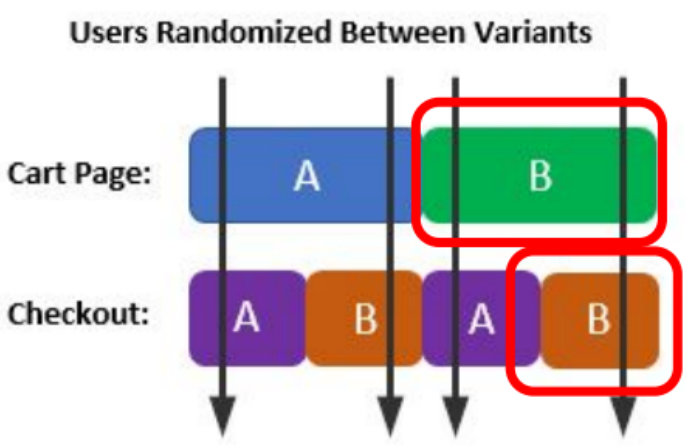
▼ **A/B테스트를 활용하는 것이 맞을지 고려하는 방법들** → 다중 실험 동시 진행과 하면 안되는 경우들

- 서로 다른 A/B테스트를 같은 기간에 여러 개 진행하는 경우를 고려해야 한다.

> 한 사용자가 다수의 A/B테스트에 영향을 받을 수 있음을 꼭 알아두자.

  - 테스트하는 표본이 섞이면서 영역 간 상호작용(interactions)이 발생할 수 있다.
    - Why? : 이러한 상호작용을 간과하고 있는 상태에서 실험끼리 간섭되는 경우가 많아지면 그만큼 통제가 되지 않은 왜곡된 결과를 받게 되는데, 이를 가지고 성공/실패로 간주할 수 있기 때문
  - 이런 사용자의 다양한 테스트 노출로 인한 간섭을 방지하기 위해 미리 사용자를 잘 분리시켜서 버킷팅(Bucketing)하는 곳도 있다. (e.g. 넷플릭스)
  - 그래서 하나의 개선에 하나의 테스트를 하는 것이 가장 안전하지만, 이럴 경우 빠르게 서비스 런칭이나 개선을 하기 어렵게 된다.
  - 보통은 테스트들이 서로 독립적이라 가정하고 다수의 A/B테스트를 동시 실행하는 것이 일반적이다.
  - 이런 상호작용을 줄이는 대안 (왼쪽은 interaction 발생, 오른쪽은 그에 대한 해법)





- A/B테스트를 사용하면 안되는 경우 Check List!
  1. 테스트를 하기 위한 데이터가 없는 경우
     - 실험을 위해 필요한 데이터가 없는데 어떻게 테스트를 하나요?
     - 기존에 가지고 있던 데이터만으로 어떻게든 실험하고 측정하려다가 잘못된 지표 설정이나 가설 변경 등이 이루어지면서 엉망인 A/B테스트를 하게 됨
     - 이렇게 되는 이유는 보통 업무 리소스 부족, 인력 부족, 기술 부족 등의 이유로 데이터 설계 및 로깅, 기획/디자인 요소에 반영하기 오래걸리거나 힘들기 때문
     - “측정하지 않으면 개선할 수 없다” → 필요하다면 아무리 사소해도, 힘들어도 수집해야함!
  2. 어떤 버그를 수정하고나서 그 임팩트를 측정하려는 경우
     - 망해가는 회사에서 많이 하고있는 경우로, 조직, 부서간 성과 욕심 때문에 일부러 하기도 함
     - 버그는 당연히 없어져야 할 일이므로 자화자찬 금지, 사이드 이펙트는 로그 분석으로도 충분히 확인할 수 있는 일임
  3. 아직 구체화가 되지 않은 아이디어를 가지고 테스트를 하려는 경우
     - 문제에 대한 가설을 명확하고 구체적으로 자주 쓰는 습관이 필요하다.
     - 여러 사람이 모인 곳에서 지속적으로 가설을 리뷰하고, 수평적으로 Reject할 수 있는 환경이 필요하다.
  4. 가설도 없이 굉장히 Random하고 다양한 아이디어를 한꺼번에 테스트해보려는 경우
     - 이 버튼의 문구를 개선하고, 결제 페이지 Step도 줄여보고 새로운 랜딩페이지도 도입하는걸 해보면 어때요?
     - 이탈 원인이 뭔지를 찾고 싶어요. 일단 A, B, C, D안 네 개에 대해 차이가 있는지 테스트하면 안돼요?
  5. 비교 대상이 없는 완전히 새로운 기능을 테스트하려는 경우
     - 기존에는 하나만 팔 수 있는 버튼이 있었는데, ‘여러 개 팔기’ 버튼을 도입해보려고 해요.
     - 기존에는 없던 ‘소환수’ 키우기 시스템을 도입해보려고 해요.
     - 제휴 업체의 보험상품을 추천하는 배너를 새로 도입해보려고 해요.

▼ **A/B테스트는 Agile하게 이루어져야 하는 이유** → 비즈니스는 ‘속도’가 중요

> 비즈니스에서는 ‘속도’가 가장 중요하며, 현재까지 Agile 방법론이 가장 효율적이다.

- Agile (애자일) 이란?

    - 가설 → 디자인 → 개발 → 검증 → 배포 → 리뷰의 과정을 하나의 Cycle로 반복하는 것
    - Cycle이 빨라지면서 실험 시간이 감소 → 더 많은 기능의 실험과 런칭이 가능함
        - 그러므로 A/B테스트는 실험 전체 프로세스를 최적화 및 자동화하는 것이 매우 중요하다.
- A/B테스트 결과를 잘 해석하고, 밀도 높게 분석하는 것은 당연히 중요하다.
    - 이 실험에서 대조군/실험군의 성별, 연령, OS에 따른 여파는 어떠할까?
    - 전월/전년 대비해서 동기간에 얼만큼 성장한걸까?
    - 이 테스트를 반영(Rollout)하게 되면 얼만큼 더 벌 수 있는걸까?
    - 추가적으로 더 레슨런(Lesson & Run)하기 위해 뭘 분석하면 좋을까?
- 하지만, 회사에서는 '속도'가 가장 중요하다.
    - 분석에 몇 달이나 몇 주가 걸린다면? → 시장 선점의 문제, 긴급한 개선에서의 시급성 등
    - 특히 머신러닝과 관련된 실험이라면 Agile 방법론으로 속도를 단축하는 것이 가능하다.

Chapter 3. A/B테스트 방법론에 대한 이해

▼ **A/B테스트 전체 프로세스에 대한 이해** : 가설 → 승인 → 구현 → 배포 → 모니터링 → 분석의 반복

1. 명확하고 구체적인 가설을 설정하고 문서화한다.
2. 중요 팀원 및 이해관계자들과 가설이 승인될때까지 논의한다.
3. 승인된 내용을 토대로 필요한 것들을 구현하고 QA를 진행한다.
4. 프로덕션 환경에서 배포한다. (Rollout)
5. 실험을 계속 리뷰하면서 이슈를 파악하고 실험 진행/중단에 대한 의사결정을 진행한다.
6. 최종 실험결과를 해석/분석하고 얻은 레슨런을 통해 위 Cycle을 계속 반복한다.

▼ **A/B테스트 프로세스의 각 단계별 메뉴얼**

1. 가설 설정에 대한 Tips → One Pager로 작성하라!

    ▼ One Pager에 들어가야할 기본적인 내용

    - Why?에 대한 내용
    - 예상되는 Impact (지표 설정)
    - 이 실험에 대한 신뢰 수준에 대해 서술
    - 현재 구현 가능한 프로토타입에 대한 서술 (Minimum Viable Product)
    - 이 실험을 통해 잠재적으로 발생할 수 있는 문제들에 대한 서술
    - 이 실험에 대한 거버넌스를 가지는 Owner

    ▼ One Pager 작성법에 대한 예시 (Udemy에서의 사례)

This is to make sure that an experiment is really needed and that the experiment is designed correctly. Every experiment requires time and effort. Sometimes we can't experiment with certain things (due to a small sample size or a brand-new thing where there is nothing to compare).

Motivation for Experiment

- What customer/business pain point is being addressed?
- What is the hypothesis?
- What data (qualitative / quantitative) do we have to support the hypothesis?
- What's the target audience, please include everybody who would be impacted by this experiment.

Metrics

- What are you trying to measure? In other words, what does the success look like?
- What is the primary metric for the experiment?
- Any secondary metrics?
- What's the estimated impact of the experiment? (How many people will get impacted, best-case, mid-case, and worst-case impact scenarios)

Description of Experiment

- What is the change? Please give as much detail as possible
- What is the very first implementation? MVP

What is the test plan?

- What audience do you want to launch the experiment for? (target audience, desktop users, US-only, logged-in user or logged-out user or everyone , etc.)
- How is the audience being split between test and control?
- What is the desired split? (50/50?) Will there be a ramp-up period?
- What's the sample size needed?
- How long does the experiment run?
- Who will monitor the experiment?
- What dashboard/tools will be used?

Results & Iterations

- What are the high-level experiment results?
- What are the details of the experiment results? (Slice the results by context, location, gender, age, etc.)
- What are the key learnings?
- What are the next steps?
    - Stop, re-launch with some change, increase bucket size, or full launch?
- If it's an iteration, please add the iteration hypothesis as well.

▼ '당근마켓'에서의 실험 설계 문서에 대한 예시

- 실험하게 된 배경(문제점)을 서술하고 PRD(제품요구사항) 문서를 첨부
- 실험을 통해 얻고자 하는 Key Result (가설 설정)
- 실험에서 측정할 지표 설정 (Metrics)
    - 핵심지표 - 실험의 성공/실패에 가장 중요하게 작용하는 지표
    - 보조지표 - 실험에서 보고 싶은 부가적인 긍정/부정 영향에 대해 측정하고자 하는 지표
    - 공통 보조지표 - 전사적인 차원에서 중요하게 보고 있는 KPI에 대한 지표
    - 가드레일 지표 - 실험을 통해 Risk가 발생하게 되는지 확인하기 위한 지표
- 실험 표본의 크기와 실험 기간에 대한 서술
- 앞서 정의한 실험지표의 등락에 따른 의사결정표 서술

지표 종류	Winner	의사결정
핵심 지표1	실험군 Win	Rollout
	대조군 Win	Rollback
가드레일 지표3	실험군 Win	Rollback
…	…	…

- 실험플랫폼의 테스트 환경과 프로덕션 환경에 대한 링크나 Spec
- 실험에 필요한 이벤트 로그에 대한 Tech Spec

2. 실험 설계 문서(가설)에 대해 잘 논의하는 Tips → 늘 가설을 만들어두고, 정기 미팅을 가져라!

▼ 누구와 미팅을 가져야할까?

1. 미팅에 반드시 필요한 사람들
    - 실험의 Owner / 백엔드+프론트엔드 엔지니어 / 데이터 엔지니어 / 데이터 분석가 / 서비스기획자(PM) / UXUI 디자이너
    - Owner가 One Pager를 만들고, 각 직군이나 책임자의 승인을 받는 구조

2. 실험에 유형에 따라 참석이 필요한 사람들
    - 크게는 제품, 작게는 특정 기능에 대해 담당하고 있는 유관부서의 리더나 책임자
    - 전반적인 KPI에 대해 의사결정권을 가지고 있는 경영진/상위권자 (KPI에 큰 영향인 실험의 경우)
    - +@) 'UX Researcher'는 Data driven인 A/B테스트보다 Data informed 단계나 Product Science에서 전략기획/선행기획에 필요한 직군이므로 필참이 아니거나 실험의 Owner를 맡는 것이 적절함

▼ 어떻게 정기적으로 미팅 Cycle을 만들까?

- 매주마다 설정한 가설에 대해 논의하고 수정할 수 있도록 정기 미팅을 설정
    - 진행할 A/B테스트에 대한 가설이 없는 경우라도 미팅은 계속 잡혀있어야 한다.
    - 멱살을 잡고 이끌어갈 미팅 Owner가 반드시 필요하다.
- One Pager의 내용 중 부족한 것을 수정/보완하여 모두에게 승인을 받기 위한 목적
    - 실험 설계 → 실험 시작 → 실험 모니터링 → 결과 분석 → 개선 모두 투명하게 Sync해야함
- 실험을 주간 모니터링하며 중단할지 계속할지 결정하거나 레슨런하기 위한 목적
    - 제안한 사람이 본인이 제안한 실험이 release 되는 것에 관심이 많고 간섭이 많아서, 시니어 분석가가 보통 오너쉽을 갖고 진행함
- 배포 및 모니터링, 분석결과 공유, 레슨런의 단계까지도 계속 정기 미팅을 통하여 보고가 되어야함
    - 빠르게 모니터링하고 결과를 직관적으로 공유할 수 있는 대시보드나 자동화가 필요하다.

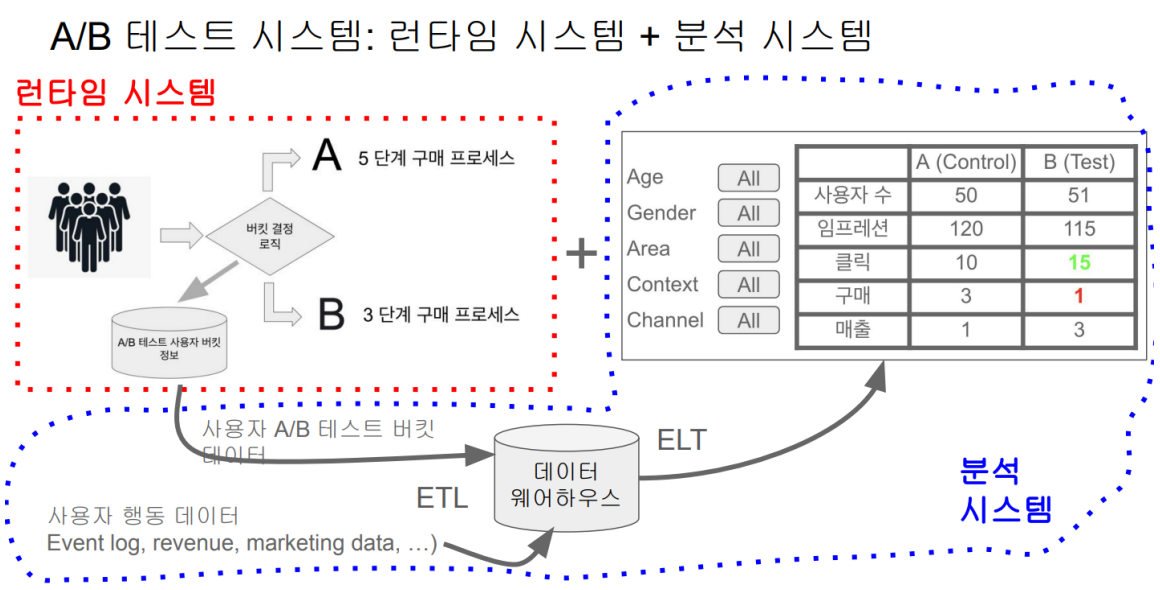
▼ 어떻게 리뷰 미팅을 수평적으로 유지할까?

- 데이터 Ownership이 있는 데이터 직군이 Reject하는 것을 두려워하지 않아야 함
    - A/B테스트는 성과를 홍보하기 위한 것이 아닌 의사결정을 지원하는 것임을 기억하자.
    - 데이터 직군의 의무는 실험이 '객관적'으로 정확하게 측정되고 분석되는 것에 Focus해야함
- 특정 부서나 특정 의사결정권자의 이해관계나 사정 때문에 졸속 승인이 되거나 반려되서는 안됨
    - 이런 경우, 정기 미팅은 아무런 의미가 없다.
- 미팅에 참석하지 않는 사람들도 결과를 투명하게 볼 수 있고 의견을 제시할 수 있도록 문서가 공개되어야함
    - 일부 내부 구성원의 의견 합치를 하려하지 말고, 객관적이고 투명한 결과를 통해 의견을 나누고 배우는 것이 필요함

3. 실험을 배포할 때 위험을 줄이는 Tips → 사용자 노출 트래픽을 조절하라!
    - **스모크 테스트 (Smoke Test)**
        - 테스트에 앞서 실험의 중요 기능들이 잘 작동하는지 확인하기 위한 빌드업 테스트
        - 매우 작은 %의 사용자(트래픽)를 노출 시켜보고 A/B테스트 세팅에 문제가 있는지 확인 (1% 미만)
    - Rollout 후 최종 실험 런칭까지의 트래픽 조절하기
        - 스모크 테스트에서 문제가 발생되지 않는 경우 점차적으로 트래픽 %를 늘려보며 위험도 테스트
            1. 초기 단계 (첫 주차) → 5~10%로 트래픽을 확장
            2. 중간 단계 (2~3주) → 25% ~ 50%로 트래픽을 확장
            3. 최종 단계 → 100%의 트래픽으로 런칭하여 실험을 진행

▼ **A/B테스트 실전 방법론** : 런타임 시스템과 분석 시스템을 설계하여 실시간으로 모니터링하기

> A/B테스트의 환경(시스템)은 '런타임 시스템'과 '분석 시스템' 2가지로 나뉘어져 있다.



1. **런타임 시스템 설계**
    - 주로 백엔드 엔지니어가 담당하지만, 분석가가 런타임 시스템 전 과정에서 검수해야함

    ▼ 자주 하게 되는 테스트는 템플릿화가 가능함을 인지하자.

    - **통계적으로 표본 추출과 관련된 방법을 자동화하는 것이 중요함**
        - **SRM** (Sample Ratio Mismatch, 표본이 50:50으로 잘 나뉘는지 확인하는 방법)
        - **MDE** (Minimum Detectable Effect, 실험 결과가 유의미하기 위한 최소 표본 크기를 확인하는 방법
    - 지표 이외에 실험과 관련된 변수도 유사한 실험에 동일한 default 값을 적용하기
        - CTR이 핵심지표인 기능 개선 실험은 유의수준 0.05, 표본 크기는 3,000명 이상일 때 효과가 있었음
        - 그러면 이와 유사한 실험에서는 매번 이 설정값들을 고려하지 않도록 default 설정 가능
    - '당근마켓'의 지표 생성 템플릿화 예시

        ▼ 모니터링 지표나 실험 지표를 템플릿하여 적재하기

        ▼ 트래킹할 유저가 00한 행동과 결과물을 Jinja Template를 통해 SQL로 정의

```
user_action: |
    SELECT
        local_date,
        user_id,
        category_id,
        event_id,
        price
    FROM
        dataset_{{country_code}}.impression_home_feed
    WHERE
        screen_name = 'fleamarket_article'
        AND DATE(created_at, '{{timezone}}') BETWEEN '{{start_date}}' AND '{{end_date}}'

metric_calculation: |
    SUM(price) / COUNT(DISTINCT user_id)
```

▼ 지표가 계산되는 Spec을 정의

```
avg_sum_of_fleamarket_price_per_users:
    supported_countries : ['kr', 'jp']
    time_grain: ['daily', 'weekly']
    dimensions: ['category_id']
```

▼ Airflow 상에서 위의 yaml 템플릿을 읽어 Python 코드와 추가적인 집계 SQL 템플릿으로 아래와 같이 실제 지표 loop로 계산 및 적재

```
WITH user_actions AS (
        SELECT
            local_date,
            user_id,
            category_id,
            event_id,
            price
    FROM
        dataset_kr.impression_home_feed
    WHERE
        screen_name = 'fleamarket_article'
        AND created_at BETWEEN '2024-08-01' AND '2024-08-07'
),

rollup_by_dimensions AS (
    SELECT
        local_date,
        category_id,
        SUM(price) / COUNT(DISTINCT user_id) AS metric_value
    FROM
        user_actions
```

```
    GROUP BY ROLLUP (
        local_date,
        category_id
    )

SELECT
    local_date,
    CURRENT_DATE('Asia/Seoul') AS updated_at,
    'asdf1234qwerty-0121' AS airflow_id,
    'avg_sum_of_fleamarket_price_per_users' AS metric_name,
    'kr' AS country_code,
    'weekly' AS time_grain,
    category_id,
    metric_value
FROM
    rollup_by_dimensions
```

▼ A/B 테스트의 Configuration을 잘 설정하자.

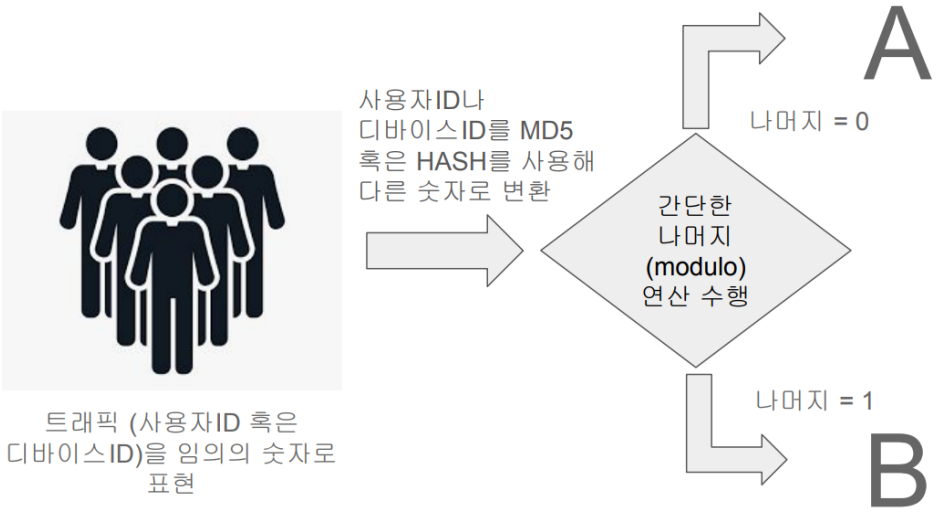
- **버켓팅(Bucketing) :** 실험 대상인 유저를 가져오는 것 (⇒ 대조군/실험군 할당)
- **버켓팅하는 방법을 미리 설계해야 한다.**
  - 미리 사용자를 A/B로 나누는 방법
    - 장점:
      - 이미 가입했거나 로그인한 사용자를 대상으로 하는 경우 가능함
      - 동시 진행되는 실험들에서 발생할 bias를 최소화할 수 있음
    - 단점:
      - 테스트 중 신규 가입한 사람들은 실험에 포함할 수 없음
      - 비가입자나 비로그인 유저에 대한 테스트는 측정할 수 없음
  - 사용자를 실시간으로 A/B테스트 진행 중에 나누는 방법
    - 장점:
      - 일반적으로 사용하고 있는 버켓팅 설계 방법
      - 로그인 여부 등과 관련 없이 적용할 수 있는 방법
    - 단점:
      - 동시 진행되는 실험의 bias가 발생하면 통제할 수 없음
      - 즉, 사용자가 실험 영역간 interaction할 가능성이 높음
- **버켓팅할 파라미터를 잘 설정해야 한다.**
  - user_id → 이미 가입하거나 로그인된 사용자 단위에서 이루어지는 실험인 경우
  - device_id → 아직 가입하지 않고 앱을 둘러보는 사용자 단위 실험인 경우
  - article_id / character_id → 특정 기업의 도메인에 따른 사용자 단위 실험인 경우 (게시글, 게임 캐릭터 등)
- **버켓팅할 크기를 잘 설정해야 한다.**
  - 우리 서비스 전체 traffic의 몇 %까지만 가져올지 결정해야 한다.
  - 주로 분석을 통해 확보할 수 있는 한계가 있는 표본 크기가 있고, 직접 설정할 수 있는 대상인 경우도 있음
    1. 복잡한 비즈니스 필터링에 해당하는 유저를 기본 대상자로 하려면 표본 크기 한계가 발생 가능
    2. 그냥 어떤 시점에 무엇을 클릭한 유저 전체가 충분히 규모가 크다면 그 안에서 %를 설정 가능

▼ 실험 표본을 선정하고, 유저를 버켓팅하여 대조군/실험군으로 할당하자.

1. 실험에 사용할 대상자의 **트리거 시점**을 설계
   - 트리거(trigger)란? → A/B테스트의 대조군/실험군을 가르게 되는 노출 시점
   - 트리거 시점 직전의 유저를 대상으로 하기 위해 선집계를 하게 됨
   - 아래와 같은 트리거 시점의 예시들에 맞게 전체 표본을 가져오는 집계를 실행함
     - 처음 앱을 설치하고 방문한 경우 (버켓팅 파라미터가 보통 device_id)
     - 가입 유저가 앱에서 채팅하기를 클릭한 경우 (e.g. user_id, chat_id)
     - 상품 결제 유저가 재결제하기 위해 결제 수단 선택 버튼을 클릭한 경우 (e.g. user_id, payment_id, click_event_id)

- 설계한 트리거 시점의 유저 표본 수가 필터링이 많은 집계일수록 표본 확보가 어려울 수 있음
    - 이런 경우, MDE를 미리 Check한 뒤 일부 집계 조건을 완화하거나, 실험을 진행하면서 동시에 MDE Check를 지속적으로 진행하며 충분한 표본을 확보

2. 트리거 시점에 해당하는 전체 유저를 **대조군/실험군으로 버켓팅**
    - 어떻게 실험군과 대조군을 50:50으로 잘 나눌 수 있을까??
        1. 버켓팅 파라미터 (user_id 등)을 MD5 또는 Hash 함수를 통해 랜덤한 다른 숫자로 변환한다.
        2. 이렇게 변환된 버켓팅 파라미터를 표본 크기로 나누어 나머지가 0이면 대조군, 1이면 실험군으로 할당한다.
        3. 이를 데이터 웨어하우스나 실험 전용 DB 등에 적재한다.
    - 잠깐, 왜 버켓팅 파라미터에 해싱 같은 변환을 하게 될까?
        - user_id가 무작위 발급되도록 해도 트리거 시점 유저의 user_id가 80%가 홀수, 20%이 짝수라면, 그냥 표본크기로 나눌 때 80:20으로 할당될 수도 있다.
        - 해시함수 등을 통해 다시 한 번 user_id를 무작위 해싱 처리를 해서 50:50으로 맞추기 위해 해싱을 한다.



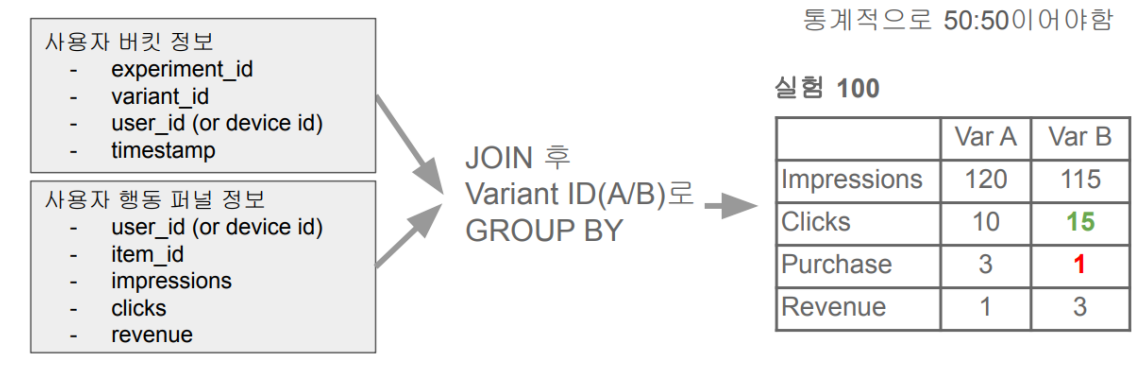
- 그럼 해싱한 상태로 표본 크기로 나눈 나머지에 따라 대조군/실험군 할당하면 끝인가?
    - **'A/A테스트'**를 진행해서 실험군과 대조군이 50:50으로 나뉘었는지 검증해야한다.
- 그럼 A/A 테스트는 어떻게 하는 것일까?
    - 말 그대로, 통계적 가설검정을 One-Sample Test로 진행하는 것
        - A/B 테스트는 2개 그룹에 대한 검정이므로 Two-Sample Test임을 참고하자.
    - **카이제곱 동질성 검정**을 통해 할당해본 대조군/실험군 간 차이가 통계적으로 유의미하지 **'않은'** 결과가 나오면 50:50으로 나뉘었다고 할 수 있다.

- 사용자 버켓 정보에 반드시 들어가야할 정보들은?
    - 실험 ID,
    - Variant ID (실험군/대조군 할당 정보)
    - Timestamp (기록된 시간)
    - user_id (버켓팅 파라미터)
    - ... (기타 추가하고 싶은 정보)

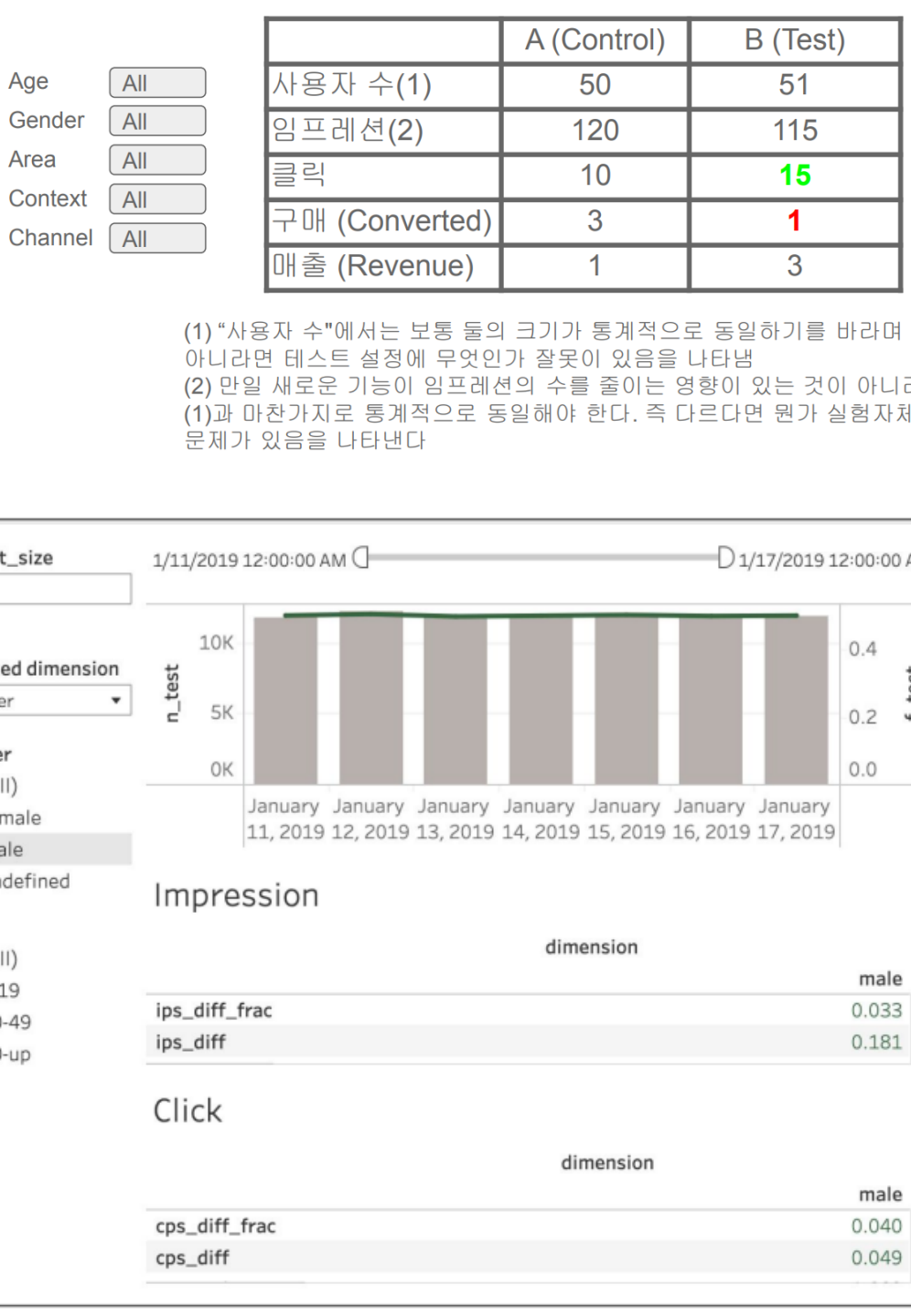
2. **분석 시스템 설계**
    - 앞서 런타임 시스템에서 만든 특정 트리거 사용자의 버켓 정보 데이터를 활용하는 것이다.
    - **사용자 버킷 정보 + 사용자의 행동 정보를 결합(조인)하여 다양한 분석 결과를 내고 모니터링** 해주는 시스템을 구성한다.
        - 사용자의 행동 정보란?
            - '퍼널 데이터', 또는 '이벤트 로그'로 불리는 사용자가 서비스를 이용함에 따라 기록되는 로그 데이터를 의미함
            - `노출`(impression), `클릭`(click), `매출액`(revenue) 같은 단순 Action부터, `X를 한 행동 중 Y를 한 행동` 과 같은 복합 Action을 포함함
            - 특히, 노출과 클릭은 실험 성과에 결정적인 영향을 미치지 않더라도 디버깅하기 좋은 지표이므로 중요하다.
    - 만약 여기에 사용자의 메타 정보 (성별, 연령, OS 등)에 따라 추가적인 Insight를 보고 싶다면 이를 추가할 수 있도록 템플릿화하는 것이 중요하다.
    - 이를 대시보드를 통해 주간 미팅에서 쉽게 이해시키고 리뷰할 수 있도록 준비 과정이 끝나 있어야 하거나 자동화가 되어있어야 가장 좋다.
    - 보통 테스트하는 기능을 백엔드 단에서 flag로 관리하는 것이 일반적이다.
        - 실험 세팅 기능을 추가로 세팅하면 그것이 활성화/비활성화되도록 할 수 있어야 한다.

◦ 회사가 많이 커지면 웹 UI를 통해 이런 부분을 확인할 수 있도록 Admin이 존재하기도 한다.

ID	Name	Page	Variants	Start Date	Active	Owner	Dashboards
100	search_v1	Search	200	2024-08-28	Yes	Simon	Link
			201				
101	recomm_v2	Home	300	2024-09-05	No	Chloe	Link
			302				



대시보드를 통한 A/B 테스트 결과 분석



(1) “사용자 수"에서는 보통 둘의 크기가 통계적으로 동일하기를 바라며 그게 아니라면 테스트 설정에 무엇인가 잘못이 있음을 나타냄
(2) 만일 새로운 기능이 임프레션의 수를 줄이는 영향이 있는 것이 아니라면 (1)과 마찬가지로 통계적으로 동일해야 한다. 즉 다르다면 뭔가 실험자체에 문제가 있음을 나타낸다

3. **A/B테스트 결과 검증하기**
   - 적당한 표본 크기와 대조군/실험군의 동질성을 확인하는 방법 복습!

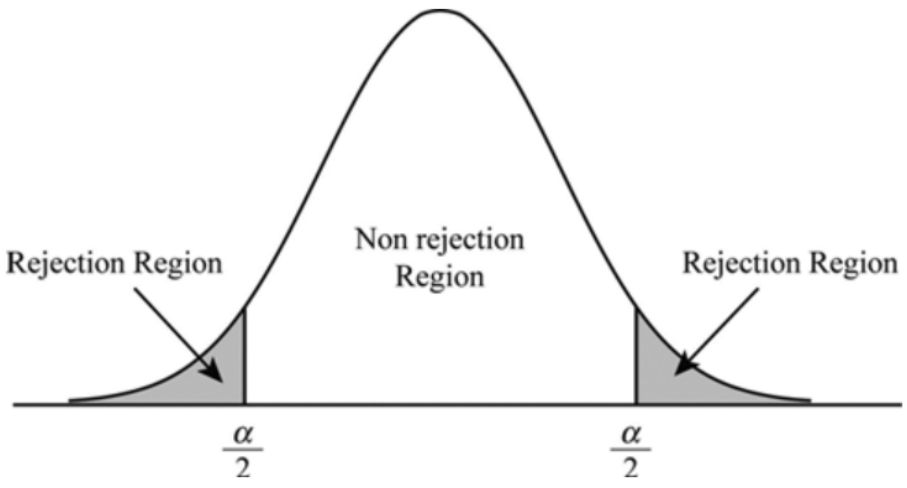
     > 이론적 부분은 학부 1학년 수준의 확률과 통계에 모두 나오며, 이를 잘 이해한다면 SRM과 MDE는 쉽게 이해할 수 있다.

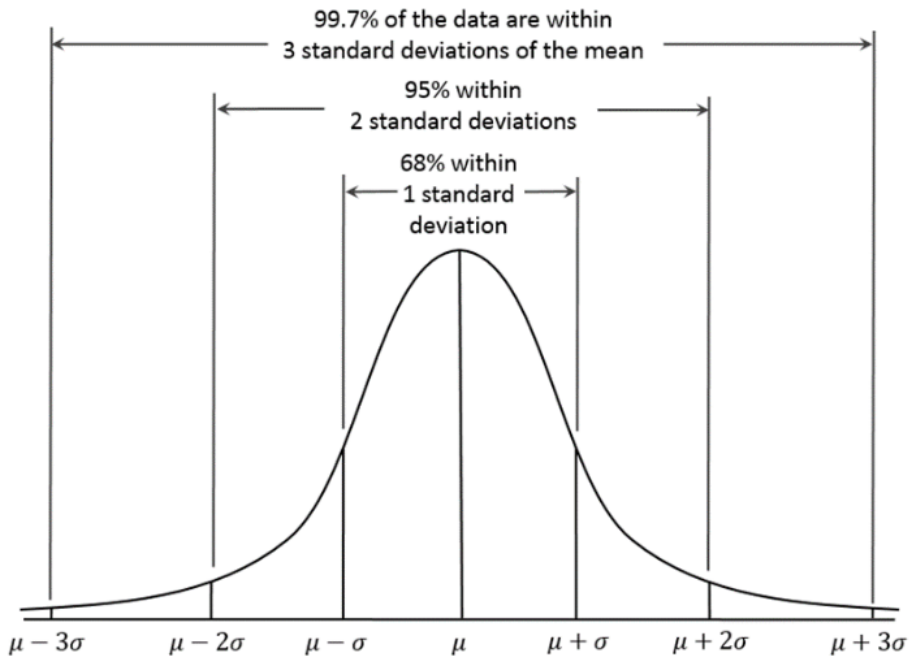
     - **SRM** (Sample Ratio Mismatch, 표본이 50:50으로 잘 나뉘는지 확인하는 방법)
     - **MDE** (Minimum Detectable Effect, 실험 결과가 유의미하기 위한 최소 표본 크기를 확인하는 방법)
   - A와 B간의 차이가 통계적으로 유의미한지 확인하는 방법

> A와 B간에 차이가 없다는 '귀무가설'을 설정하고, 이를 반박하는 '대립가설'을 채택하는지 아닌지 확인하는 과정이 바로 A/B테스트의 통계적 유의성 검증

1. 일반적으로 Two-Sample T-test (T검정) 과 Z-test 라는 가설검정을 이용하며, 두 집단의 평균이나 비율에 대한 검정을 하며, 만약 빈도수에 대한 검정을 하는 경우 카이제곱 검정을 활용한다.
    - (비교 그룹이 3가지 이상인 경우 AB, BC, AC 간 비교를 하거나, ANOVA같은 다른 검정 방법 등을 사용하기도 한다)
2. T든 Z든 둘 다 `'중심극한정리'` 라는 수학적 이론에 의해, 사용자 분포가 어떠하든 표본 수(≥ 30) 가 클수록 `'정규분포'`에 근사한다는 점을 활용한다.
    - T-test와 Z-test의 차이점은? → 일반적인 '종 모양'의 정규분포와 평균이 0이고 표준편차가 1인 '표준정규분포'를 이용하는 차이만 있음

    ▼ 즉, 정규분포의 양 끝단에 발생할 확률 (지극히 발생확률이 낮은)인지 아닌지를 검정하는 원리

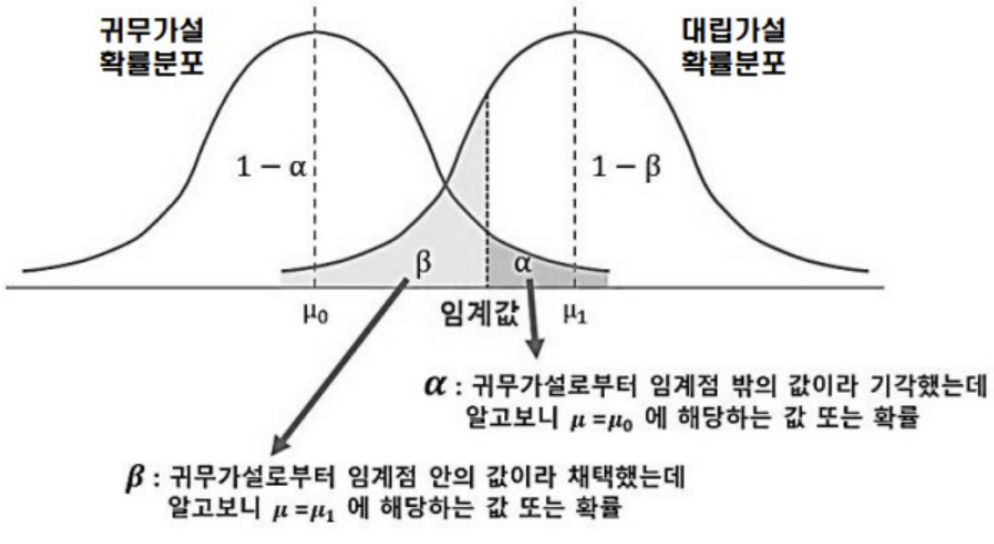




3. 가설을 잘 정의했다면, 그 가설에 의해 A와 B간에 차이가 없다는 '귀무가설'을 설정하고, 이를 반박하는 '대립가설'을 채택하는지 아닌지 확인하는 과정이 바로 A/B테스트의 통계적 유의성 검증이다.
    a. 예를 들어, '결제 페이지 단계를 줄이면 매출 평균이 올라갈 것이다' 라고 가설을 정의했다면, 대조군A와 실험군B간의 실험 결과, 매출 평균이 차이가 없다고 가정을 한다.
    b. 통계적 가설 검정을 하기 위한 '통계량' 계산을 한 뒤, 그것이 T-분포 또는 Z-분포라고 불리는 '정규분포'에서 어느 위치에 있는지 확인한다.
    c. 해당 위치가 우리가 설정한 '유의수준' 이라고 하는 정규분포의 X축의 위치 중 양 끝단에 위치한다면 A와 B가 차이가 없을 확률이 그만큼 드물다는 것으로, 둘 간의 차이가 없다는 귀무가설을 '기각'할 수 있다.
    d. 그럼으로써, A와 B는 통계적으로 유의미한 차이가 있다고 말할 수 있고, 이 결과는 A와 B간의 매출 평균이 차이가 있다고 볼 수 있다.

    - 잠깐, 그럼 p-value는 뭔데?
        - p-value는 설정한 유의수준(정규분포의 x축 값)에 따른 면적값(확률)이다.
        - 즉, 결과적으로 유의수준을 5%로 설정했다는 것은 p-value가 0.05 이하인지 보는 것이다.
    - 검정력과 유의수준에 대한 개념은 뭘까?
        - 1종오류(alpha): 차이가 없는데 둘 간에 차이가 있다고 말하는 상황
            - 이를 측정하기 위한 방법이 유의수준 (p-value를 보는 것)
        - 2종오류(beta): 차이가 있는데 둘 간에 차이가 없다고 말하는 상황
            - 이를 측정하기 위한 방법이 검정력 (Power)

- 일반적으로 1종 오류가 발생하는 것이 더 심각한 경우라고 판단되기 때문에 주로 p-value를 보고 통계적 유의성을 정하는 것이다.
- 1종 오류와 2종 오류는 Trade-off 관계에 있음을 아래 그래프를 보고 판단하자.
- 두 가설에 대한 정규분포를 그려놓고 1종오류와 2종오류가 발생하는지 검증하는게 결국 가설검정이자, A/B테스트의 통계적 유의성을 검정하는 방법이다.



Appendix. 추가로 필요한 지식들 정리해보기

▼ 데이터 처리 방식에서의 OLTP vs OLAP? 그게 뭐지?

- **OLTP (Online transaction processing)**

  > 은행 거래 시스템 같은 대규모 실시간 데이터를 정확하고 신뢰도 높게 처리 및 운영에 필요한 데이터 처리 방식으로, ‘트랜젝션’에 해당하는 데이터를 처리하는 방식

  - ‘트랜젝션’이란?
    - 데이터베이스에서 ‘ACID’ 속성을 만족하도록 수행하는 작업을 말함

    ▼ ‘ACID’ 속성이란?

    - **원자성 (Atomicity)**: 트랜잭션의 모든 작업이 성공적으로 완료되거나, 실패할 경우 트랜잭션의 모든 작업이 취소되어야 합니다.
    - **일관성 (Consistency)**: 트랜잭션이 수행된 후 데이터베이스는 일관된 상태를 유지해야 합니다.
    - **고립성 (Isolation)**: 트랜잭션이 동시에 실행될 때, 각각의 트랜잭션은 서로 영향을 미치지 않아야 합니다.
    - **지속성 (Durability)**: 트랜잭션이 성공적으로 완료되면, 그 결과는 영구적으로 데이터베이스에 저장되어야 합니다.
  - ‘트랜젝션’ 처리에 특화된 데이터 처리 방식으로, 보통 실시간으로 ‘ACID’ 속성을 만족하는 대규모 데이터를 처리해야 하거나, 운영과 관련된 업무에 사용되는 데이터에 많이 쓰이는 방식
  - (예시) 은행 거래 시스템, 전자상거래 예약 시스템, 차량 종류와 위치 정보 기록
- **OLAP (Online Analytical Processing)**

  > 데이터 분석 및 의사결정을 위해 복잡한 쿼리와 분석 작업을 수행할 수 있도록 데이터를 처리하는 방식으로, OLAP로 만들어진 데이터를 ‘OLAP Cube’ 라고도 부름

  - ‘OLAP Cube’ 란?
    - 분석을 위해 사용되는 Tabular Data (행과 열로 이루어져 있는 표)를 겹겹이 쌓아 만든 3차원 이상의 큐브 형태
    - 실제로 데이터가 3차원 큐브처럼 생겼다기보다, 개념적 이해로 접근하면 이해가 쉬움
    - 예를 들어, 시간에 따라, 지역에 따라, 상품에 따라 복합적으로 데이터를 분석하고 싶다면, OLAP 처리 방식을 통해 x축은 시간에 따른, y축은 지역에 따른, z축은 상품에 따른 데이터들을 집계할 수 있음
    - 이는 GROUP BY를 통해 컬럼별로 어떤 수치를 집계하기 위해 컬럼을 추가하는 것과 동일한 로직이며, 이러한 분석을 위한 기준 컬럼 추가하는 것을 ‘dimension을 추가’한다고 함
    - 즉, 결과적으로 OLAP Cube의 dimension이 추가되는 것과 동일한 개념
- **OLTP와 OLAP의 구조와 처리 방식의 핵심 차이점은?**
  - **데이터 구조**:
    - **OLTP 시스템**은 **정규화(Normalization)**된 데이터베이스 구조를 사용
      - 데이터의 중복을 최소화하고 무결성을 유지

        - 테이블은 관계형 데이터베이스에서 여러 개의 테이블로 나뉨
        - 트랜잭션을 빠르게 처리할 수 있도록 설계
      - **OLAP 시스템**은 **비정규화(Denormalization)**된 데이터베이스 구조를 사용
        - 데이터 분석이 용이하도록 주로 다차원 데이터 모델을 사용
        - 데이터의 집계 및 분석을 효율적으로 수행할 수 있도록 설계
    - **처리 방식**:
      - **OLTP**는 트랜잭션 단위로 빠른 응답 속도를 요구하며, 데이터의 정확성과 무결성을 중시
      - **OLAP**은 대량의 데이터를 대상으로 복잡한 쿼리와 분석을 수행하고 데이터의 집계 및 시각화 중시

▼ 가설 검정에서 드는 의문점 QnA

▼ Q, 가설검정의 테스트 그룹 수에 따라 검정법이 달라지나?

A. YES. A/B테스트에서는 주로 2개 그룹에 대한 t-test를 사용하고, 표본이 50:50의 비로 나뉘는지 확인하기 위해 단일 그룹에 대한 t-test나 카이제곱 동질성 검정을 활용함

▼ Q. t-test와 z-test 차이를 잘 모르겠는데?

A. z-test는 표준정규분포를 기준으로 검정하며, 모집단의 표준편차를 알고 있을 때 사용하고, t-test는 모집단의 표준편차를 모를 때 정규분포를 가정하고 사용함. 따라서 표준화되지 않은 정규분포를 사용하는 t-test는 표본 수와 자유도에 따라 정규분포의 모양이 달라질 수 있음 (왜도, 첨도 등에 대한)

▼ Q. 그럼 t-test와 z-test 중 무엇을 써야하지?

A. 일반적으로 t-test를 사용함. 왜냐면 실무적 관점에서 모집단의 표준편차를 알기란 사실상 매우 어렵기 때문에 t-test를 사용하되, 표본 크기를 높여서 중심극한정리에 의해 정규분포로 근사함을 가정하고 검정함

▼ Q. 정규분포로 근사하지 않는 경우 비모수적 통계 방법을 사용한다는데?

A1. YES, 하지만 실무에서 비모수 검정을 사용할 일이 거의 없기도 함. 이는 비모수 검정을 사용하게 되는 사례를 알면 이해가 쉬울 것

A2. 비모수 검정을 사용하게 되는 이유는, 표본 크기가 작아서 정규성을 만족하지 못하거나, 표본 크기가 크더라도 정규분포가 아닌 경우 사용할 수 있음. 혹은 순서형 데이터 등에 대해서도 사용할 수 있음. 예를 들어, 극소 표본에 대한 유저 대면 설문 데이터와 같이 표본이 매우 적은 경우 정규성을 만족하지 않아 비모수 검정이 필요할 수 있고, 몇만 개의 표본이 있더라도, 그 표본에 대한 분포가 정규분포를 만족하지 않으면서, 정규분포를 만족할 때까지 표본을 계속 수집할 수 없거나 확정적으로 정규분포가 아닌 서비스 도메인이라면 비모수 검정이 필요함. 구체적으로는 다음 3가지 경우에 비모수 검정을 사용함

1. **작은 표본**: 표본 크기가 작아서 중심극한정리가 적용되지 않을 때 비모수적 기법을 사용
2. **비정규 분포**: 데이터가 정규분포를 따르지 않을 때, 비모수적 기법은 분포 가정이 필요 없어 사용가능
3. **범주형 데이터**: 범주형 데이터나 순서형 데이터에서는 비모수적 기법이 적합할 수 있음

▼ Q. 카이제곱검정이 자주 나오는데, 이게 뭔지와 용례가 궁금한데?

A1. 주로 명목형 데이터의 빈도수에 대한 가설 검정을 하고 싶을 때 사용함. 교차분석의 신뢰도 측정에도 사용하고 표본을 실험 그룹으로 할당할 때도 사용할 수 있고 범용성이 높음

A2. 상황에 따라 3가지 방식으로 문제를 접근하게 되며, 내용은 아래와 같음

- **적합도 검정 (Chi-square goodness of fit test)**: 관찰된 빈도가 기대 빈도와 차이가 있는지 검정
  - 예를 들어, 선호하는 색상 분포가 균등한지 확인할 때 사용
- **독립성 검정 (Chi-square test of independence)**: 두 범주형 변수 간의 독립성을 검정
  - 예를 들어, 성별과 구매 여부 간의 관계를 분석할 때 사용
- **동질성 검정 (Chi-square Test of Homogeneity)**: 서로 다른 집단 간의 분포가 동일한지를 검정
  - 예를 들어, 여러 지역에서의 고객 선호도가 동일한지를 조사할 때, 표본이 50:50으로 나뉘는지 확인할 때 사용 (이 때는 차이가 유의미하지 않아야 50:50으로 잘 나뉜 것, 꼭 카이제곱이 아니더라도 t-test로도 검정 가능)

Q. 그럼 비율 차이나 평균 차이, 빈도수 차이에 각각 다른 검정 방법을 사용해야하나?

A. 그렇다. 하지만 평균과 비율에 대한 것은 모두 t-test를 사용하면 되고, 빈도수에 대한 검정은 카이제곱검정을 사용함. 만약 화면 조회수에 대한 차이를 비교하고 싶다면 카이제곱검정으로도 충분하나, t-test를 사용하고 싶으면 전체 대비 X그룹의 화면 조회 비율을 계산하여 비율차이 검정으로 바꾸면 된다.

Q. 대조군을 포함해서 3개 이상의 그룹에 대해서는 어떻게 검정할까?

A1. t-test를 AB, AC, BC 로 각각 검증하여 대조군과 실험군1~2, 실험군1과 실험군2 간 차이에 대한 검정을 하는 방식을 하거나, ANOVA라는 분산분석 방법을 이용함

A2. ANOVA에 대한 정보는 아래와 같이 정리하였음.

- **분산분석 (ANOVA)**: 세 개 이상의 집단 간의 평균 차이를 검정
  - 예를 들어, 여러 제품의 판매 효과를 비교할 때 사용
  - 각 집단이 모두 정규분포라는 가정이 필요함 (중심극한정리로 만족 가능)

- 각 집단 간의 평균이 차이가 있는지 검정하는 것이지만, 여기서 "AB", "AC" 등의 쌍을 비교하는 것이 아니라, 집단 간의 전반적인 차이를 한 번에 검정하는 것
- 따라서, ANOVA는 집단 간의 평균 차이가 전반적으로 존재하는지 검정하고, 후속 분석을 통해 어떤 집단 간에 구체적인 차이가 있는지를 후속분석을 통해 확인하는 방법이 필요함
- 영향을 미치는 요인의 개수가 1개인지 2개인지 아니면 같은 요인을 여러번 측정할지에 따라 3가지 방법이 있음
  - **일원 분산분석 (One-way ANOVA)**: 한 가지 요인(예: 교육 방법)이 결과에 미치는 영향을 평가
    - 예를 들어, 세 가지 다른 교육 방법이 학생의 성적에 미치는 영향을 비교
  - **이원 분산분석 (Two-way ANOVA)**: 두 가지 요인(예: 교육 방법과 성별)이 결과에 미치는 영향을 평가
    - 교육 방법과 성별에 대한 요인 간 상호작용을 검정할 수 있습니다.
  - **반복 측정 분산분석 (Repeated Measures ANOVA)**: 동일한 집단에서 여러 번 측정을 하여 시간에 따른 변화 등을 분석
    - 예를 들어, 같은 학생의 성적을 여러 학기에 걸쳐 비교

References

▼ 강의자료 아카이빙

A_B+테스트+관련+문제들.pdf

A_B+테스트+분석을+위해+필요한+데이터.pdf

A_B+테스트+시스템+구성과+전체+과정.pdf

A_B+테스트란+무엇인가.pdf

AB Test Proposal Template.pdf

Traffic을+A_B로+나누는+방법+이해하기.pdf

왜+A_B+테스트는+애자일해야+하는가.pdf

유데미+추천엔진+AB+테스트+과정.pdf

전체적인+A_B+테스트+프로세스.pdf

Traffic을+A_B로+나누는+방법+실습하기.pdf

(A_B_테스트)_사용자_버킷팅_테스트.ipynb

A_B+테스트+결과+분석의+어려움.pdf

(AB_테스트)_Central_Limit_Theorem.ipynb

(A_B_테스트)_트래픽_크기_비교.ipynb

(A_B_테스트)_가상_데이터_내용_보기와_Two_Sample_T_Test_실습.ipynb