

가상 데이터 기반 A/B 테스트 분석

버킷 크기, Impression, click, purchase, 매출액까지
비교해보자

Contents

1. 3강 통계 리뷰
2. 가상 A/B 테스트 데이터 소개
3. Two-Sample t-test 리뷰와 실습
4. Impression/Click/Purchase/Amount 비교
5. A/B 테스트 분석은 어떻게 구현이 되나?
6. 숙제



3강 통계 리뷰

3강에서 이야기했던 AB 테스트 관련 통계를 정리해보자

두 샘플 간의 데이터 비교: Two Sample T-Test

- Two Sample t-test의 결과는 결국 z-score
- https://www.statsdirect.co.uk/help/parametric_methods/utt.htm

Assuming equal variances, the test statistic is calculated as:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s^2 \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}}$$

$$s^2 = \frac{\sum_{i=1}^{n_1} (x_i - \bar{x}_1)^2 + \sum_{j=1}^{n_2} (x_j - \bar{x}_2)^2}{n_1 + n_2 - 2}$$

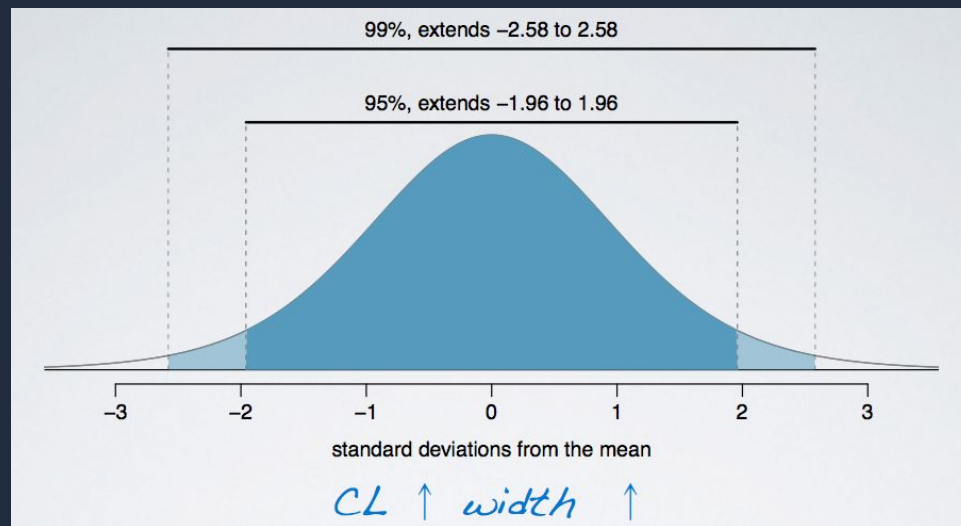
- where \bar{x}_1 and \bar{x}_2 are the sample means, s^2 is the pooled sample variance, n_1 and n_2 are the sample sizes

- The variance can be summarized as

$$\begin{aligned} s^2 &= \frac{\sum x_i^2}{n} - \frac{2\bar{x} \sum x_i}{n} + \frac{\sum x^2}{n} \\ &= \frac{\sum x_i^2}{n} - 2\bar{x} \cdot \bar{x} + \bar{x}^2 \\ &= \frac{\sum x_i^2}{n} - 2\bar{x}^2 + \bar{x}^2 \\ &= \frac{\sum x_i^2}{n} - \bar{x}^2 \end{aligned}$$

95% 신뢰도 이외의 신뢰도도 사용하느냐?

- 온라인 회사에서 테스트는 대부분 95%
- 신뢰도를 높이는 것(95% → 99%) 보다는 샘플크기를 늘리는 것이 더 좋음
- z-score가 1.95로 계산이 된다면 어떻게 하나?



95% 신뢰도로 가면 대부분 동일하다고 나오지 않느냐?

- ~~가설~~ 차이가 있고 샘플 크기가 충분하면 다르다고 나옴 (경험상)
 - 단 테스트를 얼마나 돌려야한다는 점에서는
 - 결과가 나쁜 경우의 임계치를 정하고
 - 통계적으로 의미있는 결과가 나올만큼 샘플이 커질 때까지 기다려야함
 - Don't do **data peeking**

이상 데이터 처리는 보통 어떻게 하는가?

- 일반적으로 **outlier** 분석과 그를 기반으로한 필터링이 필요함
- 이는 A/B 테스트마다 임의적으로 실행하기 보다는 모든 A/B 테스트에 적용하는 것이 필요
 - Bot 감지와 제거
 - Whale user 감지와 제거
 - Normalization도 필요 (click과 impression 예)
- 예외적인 필터링이 필요한 경우 A/B 테스트 분석시 분명하게 언급이 되어야함

가상 A/B 테스트 데이터 소개

A/B 테스트 시스템은 런타임 시스템과 분석 시스템 두 개로
구성되는데 이에 대해 살펴보자

테이블 소개

- Production DB에 저장되는 정보들을 Data Warehouse로 적재했다고 가정
- `raw_data.user_event`
 - 사용자/날짜/아이템별로 `impression`이 있는 경우 그 정보를 기록하고 `impression`으로부터 클릭, 구매, 구매시 금액을 기록. 실제 환경에서는 이런 `aggregate` 정보를 로그 파일등의 소스 (하나 이상의 소스가 될 수도 있음)로부터 만들어내는 프로세스가 필요함
- `raw_data.user_variant`
 - 사용자가 소속한 AB test variant를 기록한 파일 (control vs. test)
- `raw_data.user_metadata`
 - 사용자에게 관한 메타 정보가 기록된 파일 (성별, 나이 등등)

raw_data.user_event

```
CREATE TABLE raw_data.user_event (
```

```
  user_id int,
```

```
  timestamp timestamp,
```

```
  item_id int,
```

```
  clicked int,
```

```
  purchased int,
```

```
  paidamount int
```

```
);
```

사용자별/날짜별/아이템별

impression/clicked/purchase/paidamount 요약

보통 이런 형태의 테이블 적재는 데이터 엔지니어들이 이벤트 데이터를 가지고 하는 경우가 많음

raw_data.user_variant

```
CREATE TABLE raw_data.user_variant (  
    user_id int,  
    variant_id varchar(32) -- control vs. test  
);
```

- 보통은 **experiment**와 **variant** 테이블이 별도로 존재함
- 그리고 위의 테이블에도 언제 **variant_id**로 소속되었는지 타임스탬프 필드가 존재하는 것이 일반적
- 이 테이블은 보통 프로덕션에 있는 데이터베이스에서 가져오는 것이 일반적

raw_data.user_metadata

```
CREATE TABLE raw_data.user_metadata (  
  user_id int,  
  age varchar(16),  
  gender varchar(16)  
);
```

- 사용자별 메타정보:
 - 이를 이용해 다양한 각도에서 **AB** 테스트 결과를 분석해볼 수 있음

요약 테이블 만들어보기

```
CREATE TABLE analytics.variant_daily_sessions AS
SELECT
  variant_id,
  user_id,
  datestamp,
  count(distinct item_id) num_of_items, -- 총 impression
  sum(clicked) num_of_clicks,           -- 총 click
  sum(purchased) num_of_purchases,      -- 총 purchase
  sum(paidamount) revenue               -- 총 revenue
FROM raw_data.user_event ue
JOIN raw_data.user_variant uv ON ue.user_id = uv.user_id
GROUP by 1, 2, 3;
```

Variant별, 사용자별, 날짜별로
통계정보를 만들어주는 ELT
테이블

즉 사용자/날짜별로 요약해서
사용

- 사용자별 메타정보:
user_metadata를 조인하면
다양한 각도에서 AB 테스트
결과를 분석해볼 수 있음

데이터 살펴보기

- 가상 AB 테스트 데이터 살펴보기

데이터 살펴보기

- t-score를 2가지 방법으로 계산해봄
 - `scipy.stats.ttest_ind`

`ttest_ind` 함수를 사용해서 두 그룹의 값들을 비교

이 함수는 t-score (사실상 z-score)와 p value를 계산해서 리턴해줌

```
tscore, pvalue = stats.ttest_ind(b, a)
```

```
print(tscore, pvalue)
```

- 직접 계산해봄
- t-score를 나중에 Tableau 안에서 직접 계산해봐야함
 - 선택 정보 (날짜 혹은 사용자 그룹)에 따라 동적으로 재계산해야함

Two-Sample t-test 리뷰와 실습

impressions/clicks/revenue와 같은 지표에 대해 A와 B,
2개의 집단을 비교하는 경우


Two Sample T-Test 공식 요약 (1)

- \bar{x}_1 과 \bar{x}_2 는 각 집단의 평균
- n_1 과 n_2 는 각 집단의 크기
- s_1 과 s_2 는 각 집단의 표준편차

$$d = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Two Sample T-Test 공식 요약 (2)

- s_1 은 각 값의 제곱의 합을 알면 쉽게 계산 가능.
- 이를 뒤에서 Tableau에서 z-score 계산할 때 사용

$$\begin{aligned} S_1^2 &= \frac{\sum_{i=1}^n (x_i - \bar{x}_1)^2}{n_1} \\ &= \frac{\sum_{i=1}^n (x_i^2 - 2x_i \bar{x}_1 + \bar{x}_1^2)}{n_1} \\ &= \frac{\sum_{i=1}^n x_i^2}{n_1} - 2\bar{x}_1 \cdot \bar{x}_1 + \bar{x}_1^2 \\ &= \frac{\sum_{i=1}^n x_i^2}{n_1} - \bar{x}_1^2 \end{aligned}$$


실습

- analytics.variant_daily_sessions에서 매출액 리스트를 가지고 앞서 t-test 수행
- Two-Sample t-test Colab 링크
 - scipy.stats.ttest_ind을 사용하여 t-score (z-score)와 p-value를 계산
 - 앞서 공식으로 매뉴얼하게 계산
- Two-Sample t-test Colab 링크
 - pyspark으로 계산

Impression/Click/Purchase/Amount 비교

Two-Sample T-Test 코드를 제대로 설명해보자

Two Sample T-Test 계산을 아래 지표에 반복

- Impression
- Click
- Purchase
- Paidamount

A와 B별로 위의 평균 값을 보여주고

B쪽 값의 경우 t-score값을 바탕으로 컬러코딩

	A	B
Impressions	109	105
Clicks	15	14
Purchase	1.6	2.0
Paidamount	110	120

Two Sample t-test 요약 - “paidamount” 대상 수동 계산

Bucket “B”

- **n_b**: 세션의 수
- **sum_b**: 매출액의 합
- **mean_b**: 매출액 평균, $\text{sum_b}/\text{n_b}$
- **sum2_b**: 매출액 제곱의 합
- **square_b**: 매출액 제곱 평균, $\text{sum2_b}/\text{n_b}$
- **var_b**: 매출액 분산 (variance)
 - $\text{square_b} - \text{mean_b} * \text{mean_b}$

Bucket “A”

- **n_a**: 세션의 수
- **sum_a**: 매출액 합
- **mean_a**: 매출액 평균, $\text{sum_a}/\text{n_a}$
- **sum2_a**: 매출액 제곱의 합
- **square_a**: 매출액 제곱 평균, $\text{sum2_a}/\text{n_a}$
- **var_a**: 매출액 분산 (variance)
 - $\text{square_a} - \text{mean_a} * \text{mean_a}$

t-score 계산

- $\text{t-score} = (\text{mean_b} - \text{mean_a}) / \text{math.sqrt}(\text{var_a}/\text{n_a} + \text{var_b}/\text{n_b})$
- t-score의 값이 1.96과 -1.96 사이인지 확인 (양측 검정)
 - t-score 값은 z-score 값임

Two Sample t-test 요약 - 파이썬 모듈 사용

- Test와 Control의 raw value들을 각기 얻어온다
- 이를 `scipy`의 `stats.ttest_ind`의 인자로 지정하여 t-score와 p-value를 받아낸다

Two Sample t-test 실습

- Impression/Click/Purchased 데이터에 Two-Sample t-test 수행해보기

A/B 테스트 분석은 어떻게 구현이 되나?

OLAP Cube와 대시보드 사용: Tableau

동적으로 SQL을 생성하여 사용: Looker

AB 테스트 분석 시각화 대시보드 요구 조건

- AB 테스트별로 다음 분석이 가능해야 한다
 - AB 테스트 전체 기간에 걸쳐 키 지표가 비교 가능해야 한다
 - 일별로 키 지표의 비교가 가능해야 한다 (trend)
 - 키 지표의 경우 통계적으로 유의미한지 무의미한지 표시가 되어야 한다 (Color coding)
 - 트래픽(사용자) 메타 데이터가 있다면 이를 바탕으로 필터링이 가능해야 한다
 - 성별
 - 나이
 - 지역
 - 신규 사용자 vs. 기존 사용자
 - Acquisition channel
 - 위 정보를 통해 새 기능의 부분적인 론치가 가능할 수 있다

여기서 어려운 점은?

- 선택된 필터에 따라 **z-score** 계산이 이뤄져야 한다는 점
 - 지표, 날짜, 데모그래픽 조건
- 먼저 선택된 필터에 맞춰 **raw data** 수집이 이뤄져야함
 - 아니면 모든 가능한 조합에 대해 미리 수집을 해놓고 필터 선택에 따라 지표들을 **aggregate**
 - 이는 어떤 대시보드를 사용하느냐에 따라 다름

어느 대시보드도 A와 B 양쪽의 매출액 리스트를 읽어와서 **z-score**를 계산하지 않음

모든 대시보드도 A와 B 양쪽에서 아래 3개 정보를 각각 가지고와서 **z-score**를 계산

- 샘플 수
- 매출의 합
- 매출 제공의 합

이를 읽어오는 방법은 크게 두 가지

- 모든 필터 조합에 대해 미리 계산 (Ex: Tableau)
- 동적으로 SQL을 실행해서 계산 (Ex: Looker)

A/B 테스트 분석은 어떻게 구현이 되나?

OLAP Cube란?

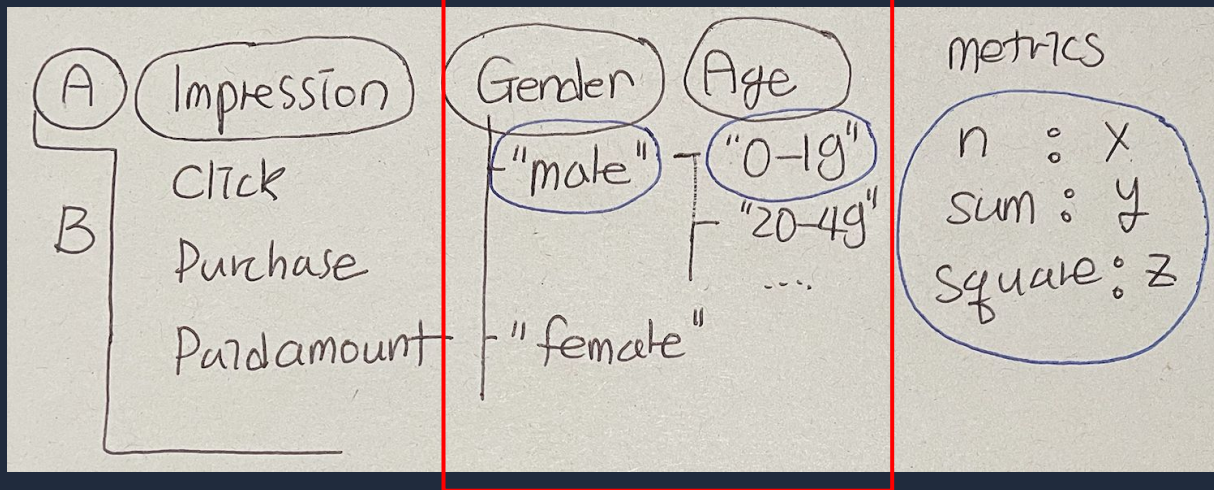
- Tableau를 사용한다면 미리 모든 조합에 대해 데이터를 수집
 - A와 B별로 impression/click/purchase/paidamount에 대해
 - 가능한 모든 date, age, gender 조합에 대해 아래를 미리 계산
 - 샘플 수
 - 매출의 합
 - 매출 제공의 합
 - 그걸 바탕으로 t-score 계산을 수행
 - 장점: 속도가 빠름 (데이터를 매번 읽어올 필요가 없음)
 - 단점: 필터가 변경될 때마다 데이터 수집 방법도 바뀌어야함
- 이렇게 미리 모든 조합에 대해 계산된 데이터를 OLAP Cube라고 부름

A/B 테스트 분석은 어떻게 구현이 되나?

OLAP Cube 예제

Variant Category

Dimensions

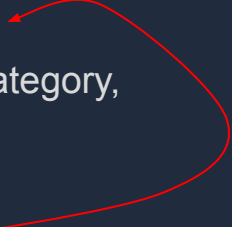


timestamp를 단순히 하기 위해 넣지 않음

A/B 테스트 분석은 어떻게 구현이 되나?

OLAP Cube 생성 SQL: Impression 예

```
SELECT
  variant_id,
  'impression' category,
  datestamp,
  age,
  gender,
  count(1) n, -- number of sessions
  sum(num_of_items) sum,
  sum(num_of_items*num_of_items) sum2 -- square
FROM keeyong.analytics_user_daily
GROUP BY 1, 2, 3, 4, 5
```



뒤에서 dbt로 만들어볼 써머리
테이블



4강 속제

오늘 속제들에 대해 알아보자

오늘 코드 모두 따라해보기

- 가상 AB 테스트 데이터 살펴보기와 Two-Sample t-test 이해하기
- Impression/Click/Purchased 데이터에 Two-Sample t-test 수행해보기

4강 QA

4강 관련 질문들에 대해 이야기해보자

다음 시간에는 **dbt**로 분석 요약 테이블을 만드는 방법에
대해 배워보자