

데이터 기획 및 분석 실무

‘배달 인프라레벨’ 사례 소개

IT/커머스 분야 8년차 데이터분석가



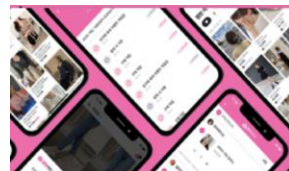
패션/리테일 A사

전략기획/데이터기획
이커머스 CRM/데이터팀



오픈마켓 B사

Business Analyst
(전략사업, 신사업)



패션플랫폼 C사

Product Analyst
(검색, 주문/결제, 셀러)



배달플랫폼 D사

Product Analyst
(배달서비스, 배달품질)

데이터분석가= 컨설턴트/전략가 (#학자, 연구원)

데이터를 분석하여 조직의 **문제를 해결**하는(혹은 방향성을 제시하는) 역할 (문제정의→액션플랜)

- Silo의 CSO로서 데이터를 통해 적극적으로 비즈니스 전략을 제안해주실 수 있는 분이 필요해요.
- Product Owner와 함께 서비스 및 전사 비즈니스의 방향을 결정하는 중요한 전략가로서의 역할을 수행해요.

도메인(Domain knowledge)

문제가 발생하는 대상/영역

① 서비스도메인

② 역할/조직/기능

서비스 도메인의 세분화 (IT서비스 중심)

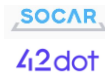
커머스



미디어컨텐츠



모빌리티



...

금융



게임



여행/숙박



서비스 도메인의 세분화 (IT서비스 중심)

비즈니스모델이
다르다

의류브랜드
VS
오픈마켓
VS
배달플랫폼

유저의특성이
다르다

2030패션플랫폼
VS
4050패션플랫폼

2030남성플랫폼
VS
2030여성플랫폼

컨텐츠의특성이
다르다

신선식품
VS
패션아이템



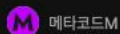
해결해야 하는 문제 정의가 달라진다

(+분석접근방향, 분석방법론, 데이터전처리...)

데이터분석가에게 중요한 문제해결능력을 더 알고싶다면?



데이터분석가 취업의 모든것, 이 영상 하나면 됩니다 | 카카오, G마켓 출신 현직자
조회수 1만회 · 8개월 전



메타코드M

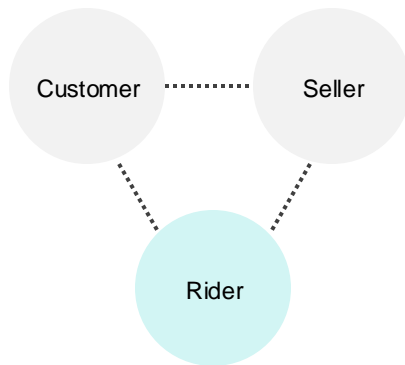
데이터분석 #취업 #메타코드m 데이터분석가 취업에 관한 모든 정보, 이 영상 하나면 됩니다 1시간 Full 버전 바로 시청하기 ...

배달플랫폼은 어떤 문제가 발생할까?

커머스 플랫폼
(오픈마켓, 버티컬플랫폼)

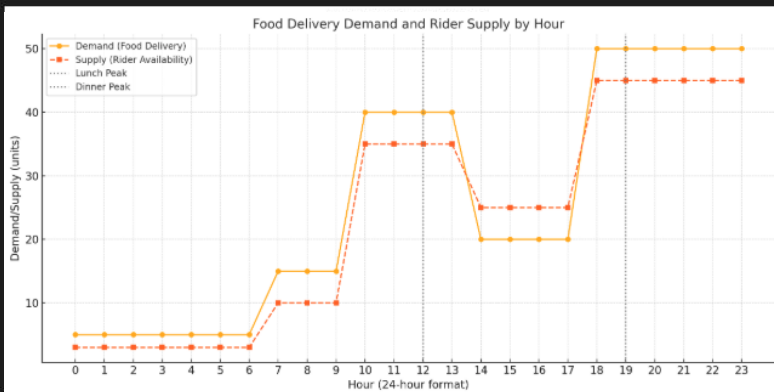


배달 플랫폼
(퀵커머스)



주문 수요 ≠ 라이더 공급(인프라)

일반적으로 시간대별 음식배달 수요와 배달 라이더 공급 수요를 그림으로 그려줘



위 그래프는 시간대별 음식배달 수요와 배달 라이더 공급을 나타낸 것입니다.

- 수요(Demand): 점심(12시)과 저녁(19시)에 수요가 급증하며, 이때 피크를 기록합니다. 새벽 시간대에는 수요가 낮습니다.
- 공급(Supply): 배달 라이더 공급은 수요를 따라가려 하지만 약간의 시간차가 있을 수 있습니다. 점심과 저녁 시간대에 역시 높은 공급을 보이지만, 수요 대비 약간 부족하거나 타이밍이 어긋나는 경우가 있을 수 있습니다.

이 그래프는 일반적인 경향성을 시각화한 것이며, 실제 상황은 지역이나 요일에 따라 다를 수 있습니다. [↗]

1.주문 수요에 영향을 미치는 요인

2.주문 수요에 영향을 미치는 요인

3.주문수요가 라이더 공급을 초과한다면?
배달품질(배달시간)에 영향

라이더 공급(인프라)을 증대시키기 위한 방향

라이더 인프라(주문 처리건수) = ① 라이더수 x ② (효율) 라이더당 처리건 수

- ① 라이더 공급(수) 자체를 증대시키는 방향
- ② 라이더의 1명당 효율(시간당 처리건수)을 증대시키는 방향

라이더 공급이 주문수요 대비 부족한 경우 아래 방안을 고민해줘

- 1) 라이더 공급 자체를 증대시키기 위한 방법
- 2) 라이더를 효율을 증대시키는 방법

1) 실시간 라이더 공급 증대

- 긴급 보너스 제공: 특정 시간대나 지역에서 라이더 부족 시 즉시 추가 보너스 공지.
- 실시간 라이더 호출: 비활성 상태 라이더에게 푸시 알림으로 배달 요청.
- 근처 대체 인력 활용: 근처 파트타임 라이더나 협력 플랫폼에 즉시 요청.

2) 실시간 효율 증대

- 실시간 경로 최적화: 교통 상황과 주문 밀집도를 분석해 최적 경로 제공.
- 피크 시간대 집중 배치: 수요 급증 지역에 실시간 라이더 재배치.
- 배달 클러스터링: 동일 지역 주문을 실시간으로 묶어 한 라이더에게 배정.
- 문제 해결 지원: 라이더의 위치/배달 문제를 즉시 해결할 수 있는 24시간 지원팀 운영.

분석사례 소개 : 배달 인프라레벨 ([링크](#))

현재 **배달인프라상황**(주문수요 대비 라이더 공급이 얼마나 많이 제공되고 있는지)을
실시간으로 제공해주는 관제 지표

실시간 배달 상황이 궁금해! 배달 인프라 레벨 구축 경험담

Jul.09.2024 천은정



Data PM

여러분은 오늘 컨디션을 표현해보는다면 어떻게 하시겠어요? 개인마다 답이 다를 수 있죠. 좋다, 나쁘다를 비롯하여 1~100에 대입하여 표현할 수도 있고요. 배달상황도 마찬가지예요. 지금 비가 오는데 배달상황이 어떨지? 이처럼 주관적인 배달상황을 측정 가능한 지표로 새롭게 정의한 "배달 인프라 레벨" 구축 경험담을 풀어볼게요 😊

분석사례 소개 : 배달 인프라레벨 ([링크](#))

늘 그렇듯, 사용자 인터뷰를 시작으로 문제정의 하기

딜리버리서비스팀에서는 배달 도메인을 담당하고 있고, 주 사용자로는 라이더 및 배달상황 운영을 하는 운영자가 있어요. 운영자는 실시간 배달상황을 모니터링하며, 상황이 좋지 않을 때 다양한 설정을 통해 안정적인 상황을 만들기 위해 힘써주고 있어요. 그래야 고객님의 주문하신 음식을 더 빠르게 배달할 수 있거든요.

문제점 1. 모니터링을 위해 운영자가 확인하고 판단해야하는 지표가 많다.

배달상황에 영향을 주는 요소들이 많기 때문에 운영자는 실시간으로 다양한 지표를 파악하고 있었어요. 아래 보이는 8개의 지표를 보고 종합적인 판단을 해야하는데, 특정 몇 개의 지표만 악화되었을 때 전반적인 수준을 운영자 감에 의해 판단해야 했어요. 여러 지표를 적절히 통합하여 객관적으로 해석하기 어려움이 있었던 거죠.

- 주문 건수
- 배차대기 건수
- 평균 배달 소요시간
- 평균 배차 소요시간
- 운행 라이더 수
- 주문취소율
- 배달취소율
- 고객안내시간 준수율

분석사례 소개 : 배달 인프라레벨 ([링크](#))

문제점 2. 실시간으로 변하는 지역 상황을 즉각 인지하는 데 한계가 있다.

운영자는 지역마다 빠르게 변하는 상황을 주기적으로 체크하며, 운영하는 전체 지역을 확인해야하는 상황이었어요. 언제 어떻게 바뀔지 모르는데 배달상황이 좋지 않은 경우라면, 안정적인 상황으로 만들기 위한 각종 운영 조치도 이뤄져야 하는데 효율적이지 않은 상황이었죠. [모니터링 > 운영 설정 > 모니터링 > 운영 설정 해제]와 같이 반복적인 업무가 모두 수기로 진행되고 있었어요.

문제점 3. 지표의 기준이 일관되지 않고 상황마다 바뀐다.

또 위 지표들은 외부적인 변수(날씨, 월드컵 등과 같은 주문수 급증)에 따라 달라지는데, 전일/전주 기준으로 비교하므로 조회 시점마다 모니터링 비교군이 다르다 보니, 지표에 대한 정확한 기준이 없어 경험적 기준에 의존하여 실시간 배달상황을 판단하고 있었어요.

분석사례 소개 : 배달 인프라레벨 ([링크](#))

시시각각 달라지는 배달상황을 측정가능한 지표로!

위 방향을 바탕으로 새로운 지표를 만들게 되었고, 이 지표는 “배달인프라 레벨”로 재탄생했어요.

지역별 주문수 대비 라이더 수가 적정한지를 판단하여, 이를 Level 1~9로 정의했어요. 누가봐도 지금 송파구는 배달인프라 레벨이 2야 = 배달상황이 안정적이구나 또는 지금 송파구는 배달인프라 레벨이 8이야 = 인프라 상황이 심각하구나를 알 수 있죠.

배달인프라 레벨	레벨 그룹
1	양호
2	
3	
4	주의
5	
6	경고
7	
8	위험
9	

배달인프라 레벨을 만들기 위해서는 현재 레벨을 판단할 수 있는 기반 데이터와 이 데이터로 어떻게 상황을 판단할 것인지 기준을 잡는 게 중요했어요. 레벨별 상황이 의미하는 바는 결국, 우리가 만든 배달 인프라 레벨이 정말 현재의 배달상황을 잘 표현하는가?와 같은 뜻이니까요. 그래서 데이터과학자분들과 함께 협업이 필요했고, 과제 초반부터 PM, 데이터과학자, 개발자가 함께 과제에 참여했어요.

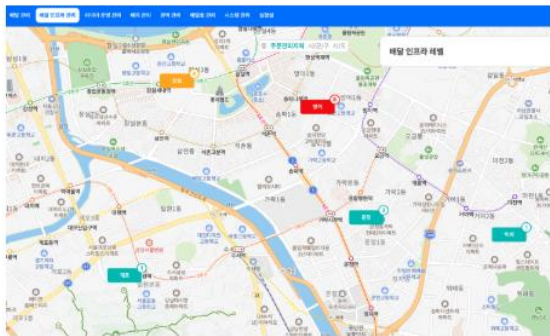
분석사례 소개 : 배달 인프라레벨 ([링크](#))

구분	작업내용	담당자	6월		7월				8월				9월				
			3w	4w	1w	2w	3w	4w	1w	2w	3w	4w	5w	1w	2w	3w	4w
기획	<ul style="list-style-type: none">기획 및 방향탐색	@이민준(기획/데이터팀)															
데이터	<ul style="list-style-type: none">과거 배달 품질 관련 지표 분석, 배달 인프라 지표 개발 방안 검토	@이지익(데이터/데이터팀)															
	<ul style="list-style-type: none">지표 산출식 설계 및 고도화																
	<ul style="list-style-type: none">배달인프라 개발 구간 정의																
	<ul style="list-style-type: none">지표 모니터링, 산출식 보완																
개발	<ul style="list-style-type: none">시스템 구조 설계, 작업 계획	@이신준(엔지니어링팀) @이우연(엔지니어링팀)															
	<ul style="list-style-type: none">라이더 관제변수 집계																
	<ul style="list-style-type: none">배달 관제변수 집계																
	<ul style="list-style-type: none">배달인프라 지표, 개발 계산																
검증	<ul style="list-style-type: none">데이터 제공<ul style="list-style-type: none">API, 이연표, 매번치																
	<ul style="list-style-type: none">실물테스트																
운영자랑 제공	<ul style="list-style-type: none">현재 프로세스 및 수식 검증																
	<ul style="list-style-type: none">대시보드 및 알림 제공																

위 일정표를 보면, 데이터 분석과 개발이 병렬로 진행되는데요. [분석 > 개발 > 검증]의 구간을 짧게짧게 계획해서 여러번 다듬어 정확도를 높이려고 했어요. 데이터과학자는 지표 산출 과정 그리고 이 지표와 수식을 계속 보완하며 고도화하는 과정에 많은 시간을 쏟았어요. 개발자는 지표를 만들기 위한 기반 데이터를 쌓아야 했는데 하루 100만건 가량의 주문, 배달과 15만명 가량의 라이더 정보를 실시간으로 시스템 부하 없이 집계하는 부분에 집중했어요. PM인 저는 지표를 활용하게 될 운영자 및 유관부서와 과제방향 및 세부 정책을 맞추고, 로드맵을 세우는 부분에 집중했어요. 배달 인프라 레벨을 구축한 이후 어떻게 활용하는게 좋을지?에 초점을 맞췄죠.

분석사례 소개 : 배달 인프라 레벨 (링크)

이렇게, 반복되는 검증 과정을 겪고 약 3개월동안 만든 배달 인프라 레벨을 실시간 대시보드와 경고 단계에 있는 지역들을 Slack 알람으로 인지할 수 있는 기능으로 운영자(사용자)에게 오픈하게 됩니다.



운영자 어드민



배달 인프라 레벨 대시보드

서비스에 녹아들기, 글을 마치며

	운영 현황	모니터링 소요 시간	운영 설정 소요 시간	운영 설정 주기
배달 인프라 레벨 도입 전	4-5 명	10분	5-20분	1시간에 2회 설정
배달 인프라 레벨 도입 후	2-3 명	3분	1-5분	10분에 1회 설정

배달 인프라 레벨 도입 전/후 성과

배달 인프라 레벨은 기존 모니터링에 투입되는 인원을 최대 5명 → 3명으로, 모니터링에 소요되는 시간을 10분 → 3분으로 단축하는 성과가 있었어요. 모니터링~운영 설정 소요 시간으로 본다면 최대 30분 → 5분 내 설정하여 조치를 빠르게 할 수 있다는 측면에서도 개선효과가 있었죠. 이후 인프라 레벨을 활용하여 수립된 운영 정책 기준에 맞춰 할증, 배달반경 조정 등 수기로 설정했던 기능을 자동화하는 작업이 후속과제로 이루어져 수기업무에 투입되는 시간을 5.6시간(운영시간 대비 약 30%) → 0시간 으로 절감하여 운영 효율을 높이는데 큰 효과가 있었어요. 현재는 특정 배달 인프라 레벨이 되었을 때 고객/업주 대상으로 배달상황을 안내하는 기능도 적용되어 있고요.

그동안 사용자 화면, 기능 등과 같은 눈에 보이는 어떤 결과물을 만들었다면, 배달 인프라 레벨은 운영상황을 판단할 수 있는 기준점을 만들고, 객관적으로 상황을 표현할 수 있는 지표로 새롭게 정의하여 의미 있었던 프로젝트라고 생각해요. 기준점을 만들고 나니 다양한 조직과 배달상황을 이야기할 때 동일한 기준으로 이해하고 소통할 수 있었고, 이 기준점을 토대로 운영 설정 기능, 정책 등에도 활용되고 있어 과제를 시작으로 많은 산출물도 얻었으니깐요. 회의 중에 "인프라 레벨 5인 상황에서" 라고 누군가 이야기하면, 가슴이 웅장해집니다 🍷

현재의 배달 인프라 레벨은 실시간 배달수, 라이더수를 기반으로 만들어졌지만, 앞으로는 외부 환경 요소(계절, 실시간 날씨, 교통상황 등)에도 더 기민하게 판단하여 고도화할 계획이에요. (배달 인프라 레벨 Ver2 가보자고!!)

비슷한 사례로 고민하시는 분들께 도움이 되길 바라며, 앞으로도 다양한 문제를 해결하며 경험한 내용을 공유 할 날을 기대하며 글 마치도록 하겠습니다.

과제내 분석가 담당 작업 내용

구분	작업내용
기획	<ul style="list-style-type: none">• 기획 및 방향협의
데이터	<ul style="list-style-type: none">• 과거 배달 품질 관련 지표 분석, 배달 인프라 지표 개발 방안 검토
	<ul style="list-style-type: none">• 지표 산출식 설계 및 고도화
	<ul style="list-style-type: none">• 배달인프라 레벨 구간 정의
개발	<ul style="list-style-type: none">• 지표 모니터링, 산출식 보완
	<ul style="list-style-type: none">• 시스템 구조 설계, 작업 계획
	<ul style="list-style-type: none">• 라이더 관제변수 집계
	<ul style="list-style-type: none">• 배달 관제변수 집계
	<ul style="list-style-type: none">• 배달인프라 지표, 레벨 계산
검증	<ul style="list-style-type: none">• 데이터 제공<ul style="list-style-type: none">• API, 이벤트, 메트릭
	<ul style="list-style-type: none">• 성능테스트• 전체 프로세스 및 수식 검증
운영자향 제공	<ul style="list-style-type: none">• 대시보드 및 알람 제공

지표 산출식 조건

- 기본적으로 배달수&라이더수에 대한 특성을 반영해야한다.

- 배달건수가 증가할수록 배달인프라 악화 → 양의 관계
- 라이더수가 증가할수록 배달인프라 개선 → 음의 관계

- 가능한 단순한 산출식을 구상하여 이해가 쉽도록 한다.

$$\frac{\text{배달건수}}{\text{라이더수}}$$

- 배달품질 관련 지표(배차소요시간 등)와의 상관성이 유의미한 지표여야 한다.

(배달건수 및 라이더수 지표가 실제 배달품질에 어느 정도의 영향이 있는지 파악하기 위한 **파라미터 작업** 필요)

$$\text{배달인프라레벨(배달품질)} = \frac{\text{배달건수} * a}{\text{라이더수} * b}$$

예를들어 라이더 1명이 1시간의 3건의 배달건을 처리한다고 할때

동일한 배달인프라레벨(배달품질)을 유지하기 위해서, 라이더수 1명이 증가하면 배달건수 3명이 증가해야 함

내부 데이터가 아닌 더미 데이터 활용

테스트 또는 모의 작업을 위해 인위적으로 생성된 가상의 데이터

```
python

# Fixing the issue by ensuring all times are datetime objects and durations are timedelta objects
# Ensure all times are datetime objects and durations are timedelta objects

# Re-import necessary libraries after reset
from datetime import datetime, timedelta
import pandas as pd
import numpy as np

# Constants
start_time = datetime(2024, 12, 19, 0, 0) # Start of the day
end_time = datetime(2024, 12, 19, 23, 59) # End of the day

# Generate timestamps for every 10 minutes
timestamps = pd.date_range(start=start_time, end=end_time, freq='10min')

# Dummy Data 1: Orders and Riders Count
np.random.seed(42)
orders_count = np.random.poisson(lam=5, size=len(timestamps))
riders_count = np.random.randint(1, 10, size=len(timestamps))

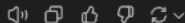
dummy_data1 = pd.DataFrame({
    'Timestamp': timestamps,
    'Orders': orders_count,
    'Riders': riders_count
})
```

하루동안 특정 배달플랫폼에서 발생한 2개의 더미 데이터를 만들어 줘 (총 1000개 샘플)

- 더미데이터1 : 하루내 10분 간격으로 발생한 주문건수 및 라이더수 데이터
- 더미데이터2 : 하루 동안 발생한 주문내역 (배달id, 주문시간, 배달완료시간)



It seems like I can't do more advanced data analysis right now. Please try again later. If you need help generating dummy data or working with specific datasets, let me know how I can assist in a different way! [\[>-\]](#)



모델링 이전 데이터 전처리

- 주문내역 테이블에서 10분단위 배달시간을 계산하여(group by) 관제지표 테이블과 join

주문 및 라이더 stat 테이블 (ord_rider_stat)

시간대	주문건수	라이더수
9:00		
9:10		
9:20		

주문내역 raw 테이블 (ord_raw)

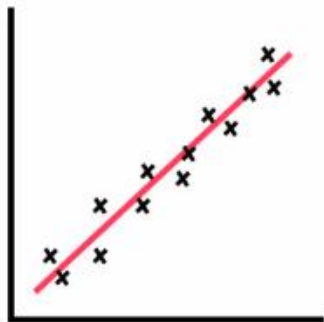
배달id	...	주문시간	배달완료시간
1		9:00	9:30
2		9:02	9:25
3		9:03	9:35



시간대	주문건수	라이더수	평균배달소요시간
9:00			
9:10			
9:20			

상관계수(Correlation)

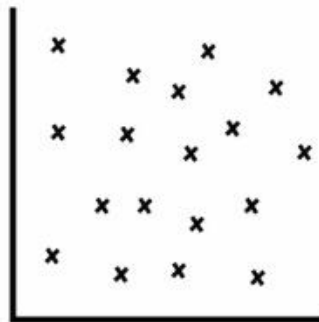
- 상관계수는 두 가지가 얼마나 비슷하게 변하는지를 나타내는 숫자입니다. 1에 가까우면 두 값이 함께 오르거나 내리고, 0에 가까우면 아무 관계가 없고, -1에 가까우면 하나가 오를 때 다른 하나는 내린다는 뜻이에요.



Positive
Correlation



Negative
Correlation



No
Correlation

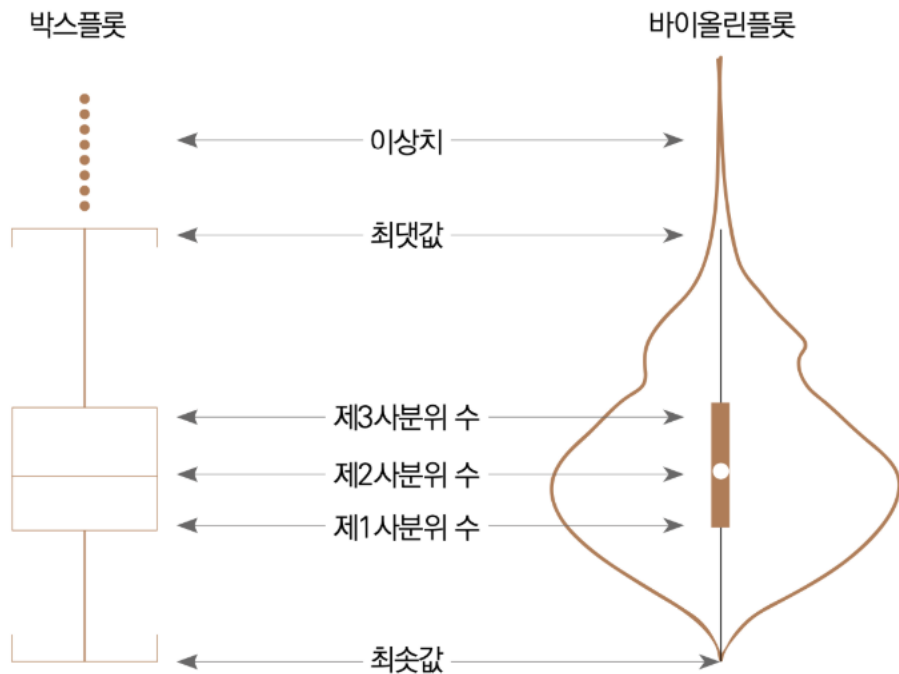
모델링(최적 파라미터 찾기)

시간대	주문건수 * 주문수 parameter	라이더수 * 라이더 parameter	평균배달소요시간
9:00	주문수 * a	라이더수 * b	
9:10	주문수 * a	라이더수 * b	
9:20	주문수 * a	라이더수 * b	
9:30	주문수 * a	라이더수 * b	
9:40	주문수 * a	라이더수 * b	
...	주문수 * a	라이더수 * b	
23:20	주문수 * a	라이더수 * b	
23:30	주문수 * a	라이더수 * b	
23:40	주문수 * a	라이더수 * b	
23:50	주문수 * a	라이더수 * b	

최대한 높은 상관성이
나타나도록 하는
파라미터값 찾기

$$\text{배달인프라레벨} = \frac{\text{배달건수} * a}{\text{라이더수} * b} \longleftrightarrow \text{배달품질}$$

클러스터간 분포를 비교할때 활용하기 좋은 Violin Plot ([링크](#))



클러스터간 분포를 비교할때 활용하기 좋은 Violin Plot ([링크](#))

