

## Elastic Stack 을 활용한 Data Dashboard 만들기

Week 1 - Data를 시각화해보자



Fast Campus

내용	페이지
강의소개	3
Elastic Stack 소개	7
Elasticsearch	
특징	20
용어 정리	26
Elastic Stack Workflow	33
Kibana 소개	36
Index 등록	44
데이터 탐색	57
Visualize 맛보기	79
Aggregation	
Bucket Aggregation	87
Metric Aggregation	96
Visualize 안내	105
Visualize 실전	
Markdown	126
Metric	129
Coordinate Map	150
Region Map	154
Tag Cloud	159
Pie Chart	166

강의가 끝나면 data가 주어지면 dashboard를 구축하고 needs에 맞게 운영할 수 있다.

그러므로

- 모든 가능 100% 마스터는 하지 않을 거고
- dashboard 구축 및 운영을 위한 전반적인 내용 학습과
- 문제가 생길 시 troubleshoot



하는 방법을 중심으로 배운다.

단,

- 검색엔진으로서 Elasticsearch
- Elasticsearch Architecture
- (고급) query 및 query 최적화



등은 다루지 않을 것이다.

강의자료

Github

LINK



FAQ

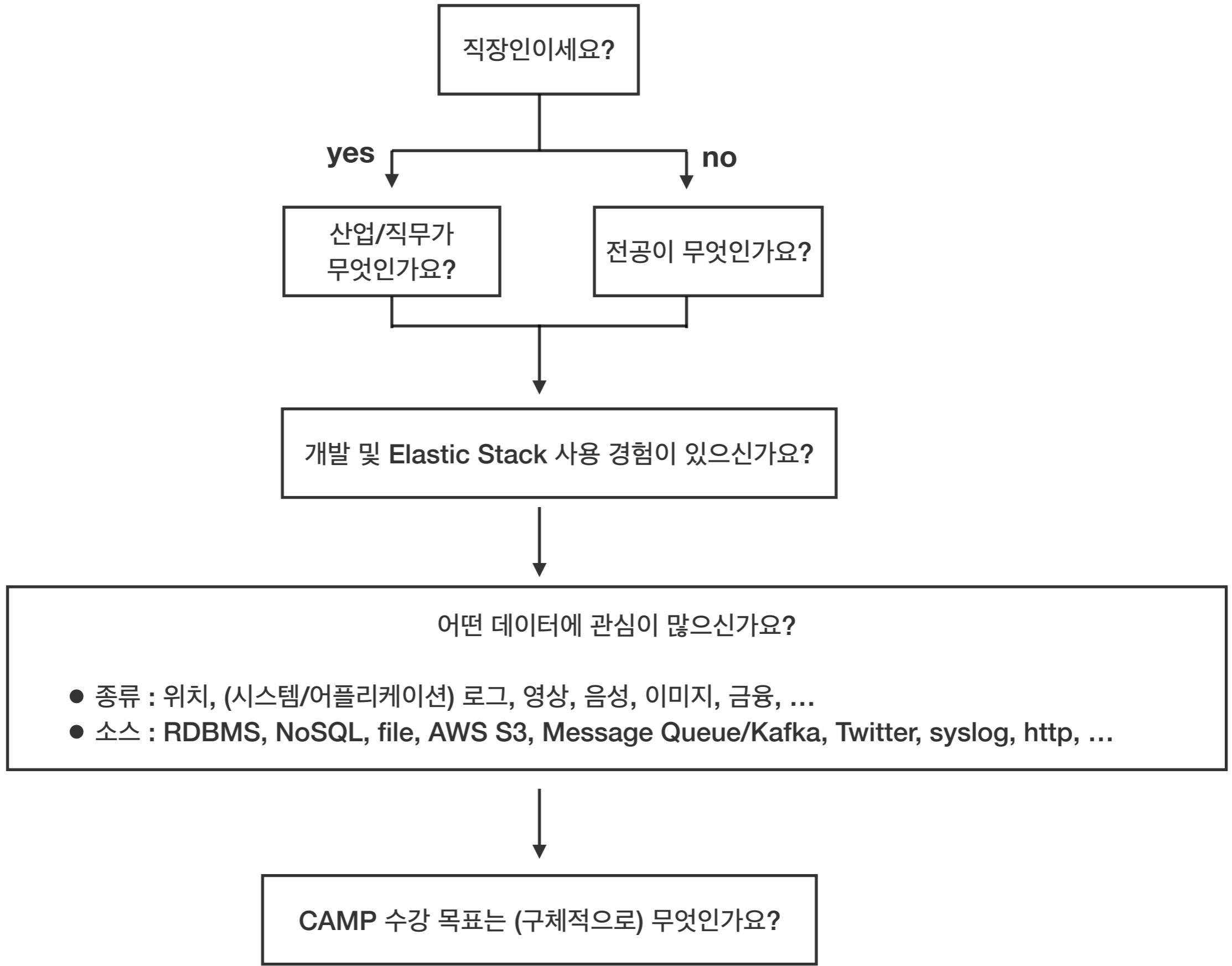
지난 기수 수강생들 질문 정리 🏰 (open/closed)

Questions

- Kakao Open Chat (#elastic5) 🏰
- [패스트캠퍼스] Elastic Stack을 활용한 Data Dashboard 만들기 CAMP 🏰

Online Sources

- Elastic Stack and Product Documentation 🏰
- Discuss the Elastic Stack 🏰
- Facebook Elasticsearch Korea Group 🏰
- Stack Overflow 🏰



개요

ELK Stack?  
Elastic Stack?  
Elasticsearch?  
Elastic?

ELKB Stack?

Elastic Stack이 무엇인지 간략히 살펴보자

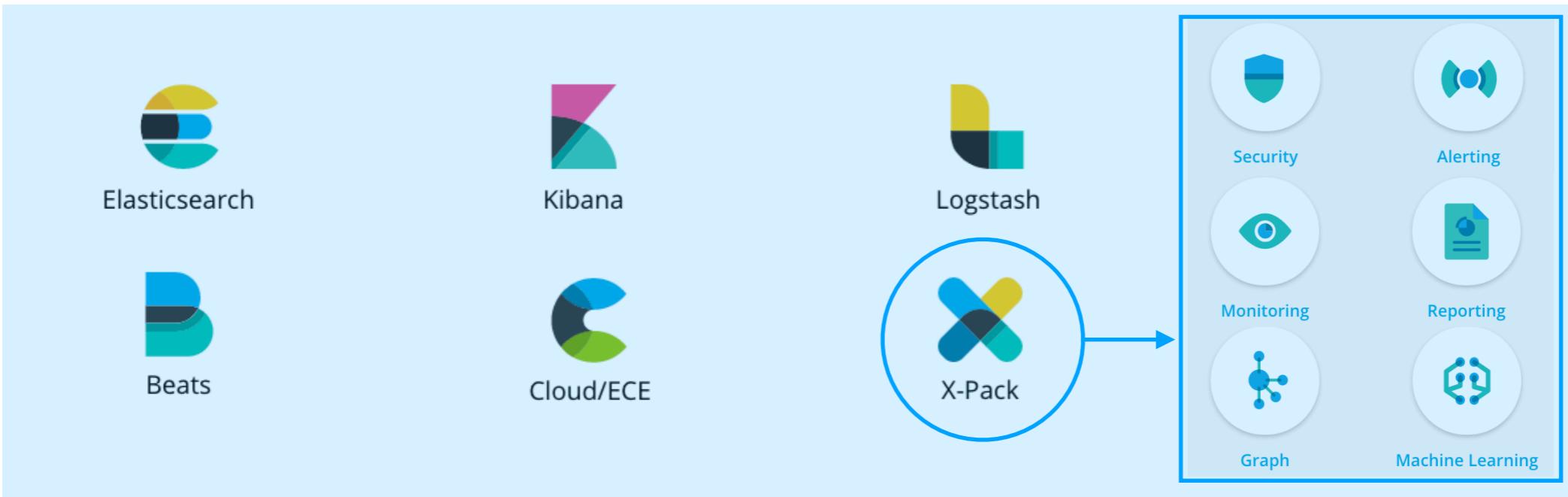
# Elastic Stack

Stack	Description	Symbol	Link
 Elasticsearch	데이터 검색, 분석, 저장	<b>E</b>	
 Logstash	데이터 수집, 변환, 전송	<b>L</b>	
 Kibana	데이터 시각화	<b>K</b>	
 Beats	데이터 수집 및 전송	<b>B</b>	

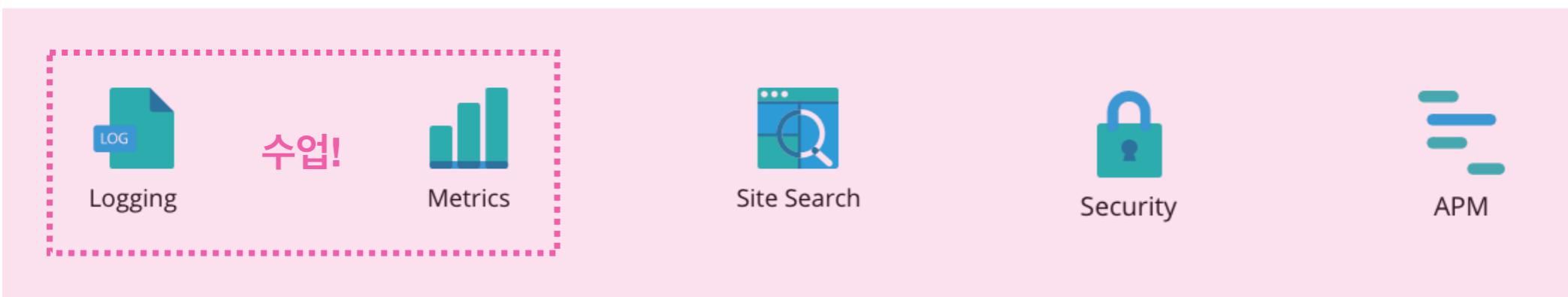
Elastic Stack으로 무얼 할 수 있을까?

= Elastic Stack을 왜 배울까?

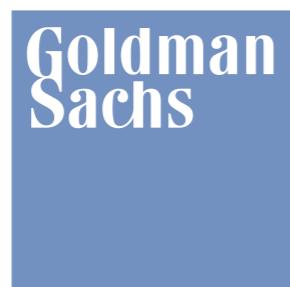
## Products



## Solutions



**Elastic Stack을 실제 Production에서 사용중인 회사는 있을까? **

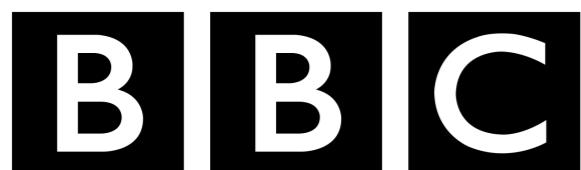


# NAVER

궁금하면 로고 클릭



WIKIPEDIA  
The Free Encyclopedia



**해결하려는 문제만큼 Elastic Stack을 어떻게 사용하는지도 회사마다 다양하다**

## Event prediction and forecasting

- 오늘 3시 A지역에서 몇 건 정도의 Uber 요청이 나올까?
- A 지역에서 B 지역까지 간다면 몇 분이나 걸릴까?

## Engineering Standards

- high availability (HA)
- low latency
- scalability
- operation friendliness

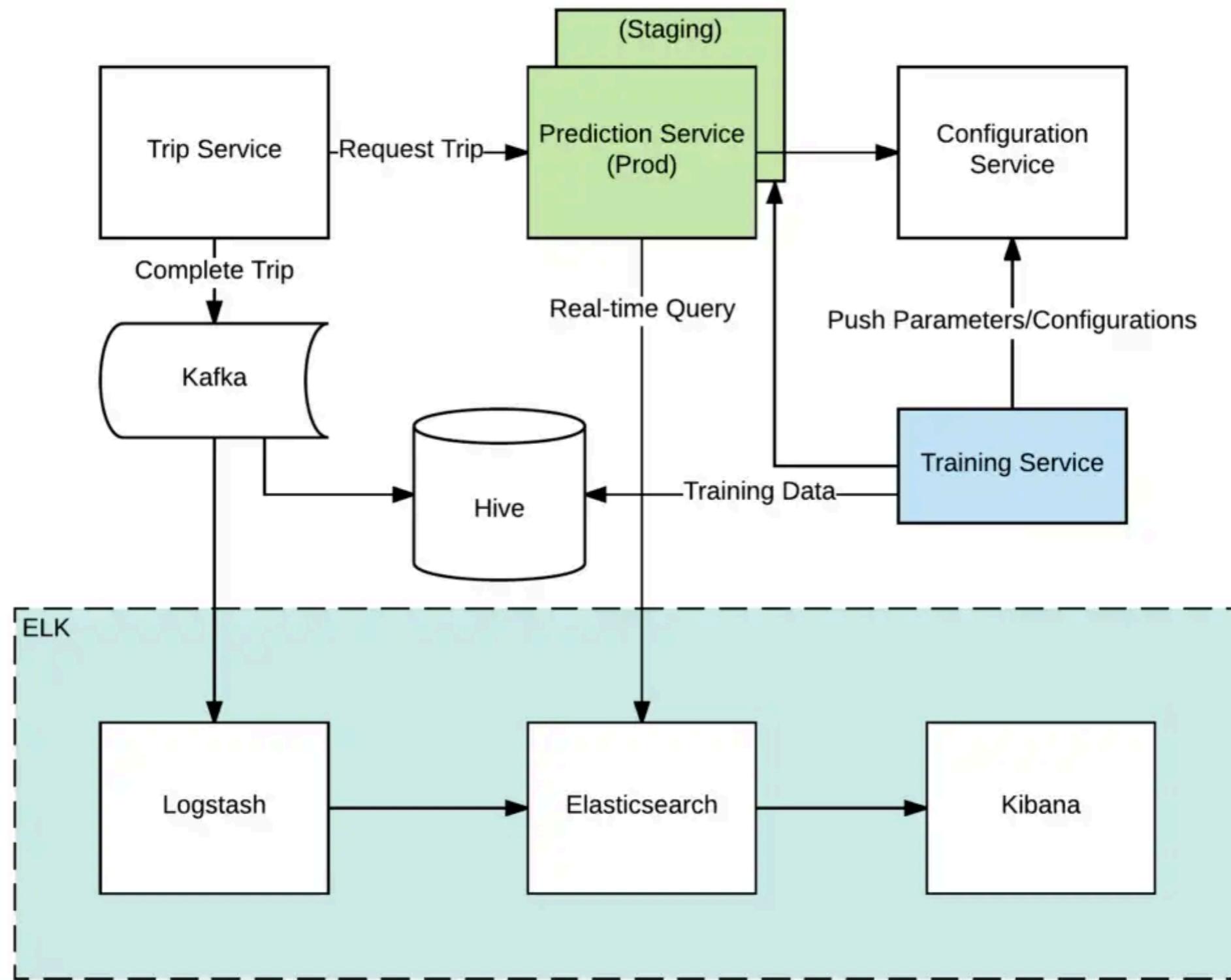
## Algorithm (k-nearest neighbors algorithm, KNN)

- finds k-nearest neighbors (similar historic trips over a period of time)
- performs a regression on them to create a prediction

## Algorithm-related technical challenges

- robust store/search engine able to deal with thousands of queries per second (QPS)
- geospatial query support to assist with filtering  $k$ -candidates.

## Uber - System Architecture



### 주요 이슈 (시간순)

- POI 위치의 정확한 인식
- 로그 분석 및 시각화
- elasticsearch cluster 효율 극대화
- 메뉴 검색 + (고객 문의 검색)



### 대응

- elasticsearch 도입 후 모든 POI에 대해 정확한 주소와 좌표 색인
- logstash, kibana, x-pack 도입
- 관리형 서비스인 elastic cloud enterprise 도입
- elasticsearch full-text search 이용 (+계획중)

## Grab - At a Glance

- Rides per day : 3.5 million
- Clusters : 10
- Nodes : 50
- Documents : 10 billion
- Daily Ingest Rate : 50 million
- Queries per day : 300 million



Elasticsearch



Kibana



Logstash



Cloud/ECE



X-Pack

# Content Warning

The information presented in this chapter is for your interest. You are not required to understand and remember all the details. Elasticsearch. The options that are discussed are for advanced users only.

왜 Elasticsearch?

Read the section to gain a taste for how things work, and to know where the information is in case you need to refer to it in the future, but don't be overwhelmed by the details.

## Near Realtime (NRT)

데이터 색인 (=Indexing) 후 약 1초 (=Near Realtime) 후부터 검색 결과에 반영된다 ( $\neq$  처리 시간)

심화

**REFRESH :** 기본적으로 1초마다 실행하기에 Near Realtime



segment에 존재하는 데이터는 검색 가능

## Fast

(기본적으로) 모든 Field에 대해 Indexing 처리하므로 검색 처리 시간이 짧다

### 심화

```
PUT my_index
{
  "mappings": {
    "_doc": {
      "properties": {
        "user_id": {
          "type": "keyword"
        },
        "last_updated": {
          "type": "date"
        },
        "session_data": {
          "index": false,
          "type": "keyword"
        }
      }
    }
  }
}
```

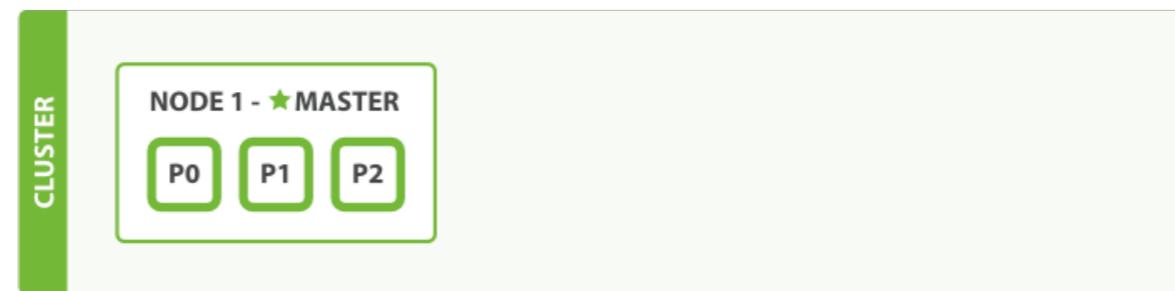


이처럼 강제로 설정하지 않는 이상 기본적으로 모든 field를 Indexing한다

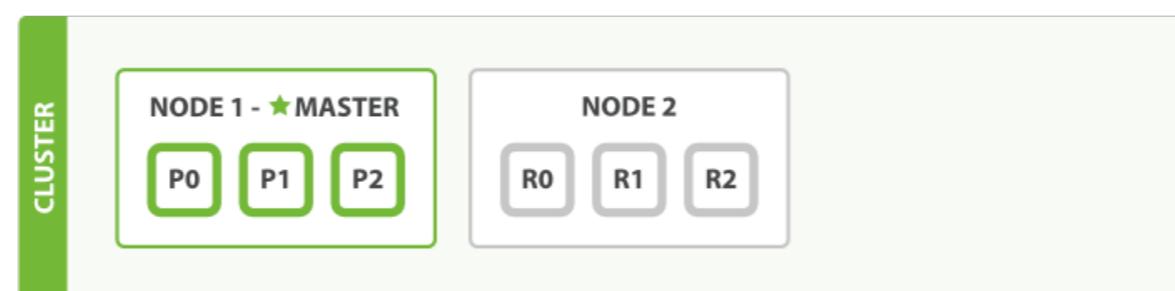
## (Horizontal) Scalability

운영 중인 elasticsearch cluster에 간단한 설정을 통해 elasticsearch node 추가 가능

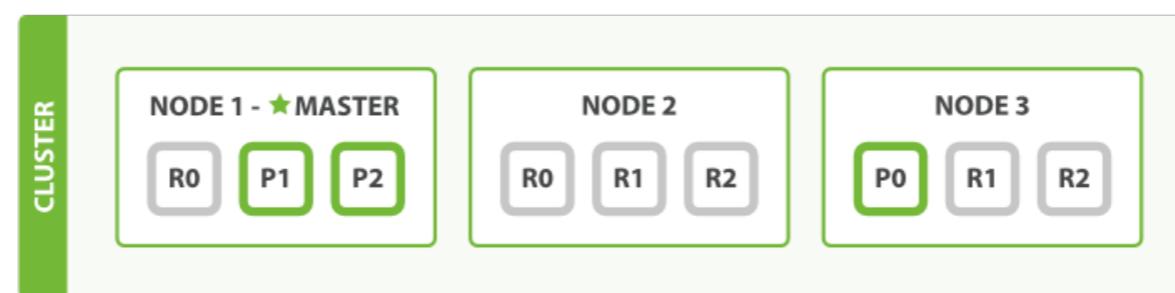
심화



Node (≒ 서버) 1개 추가



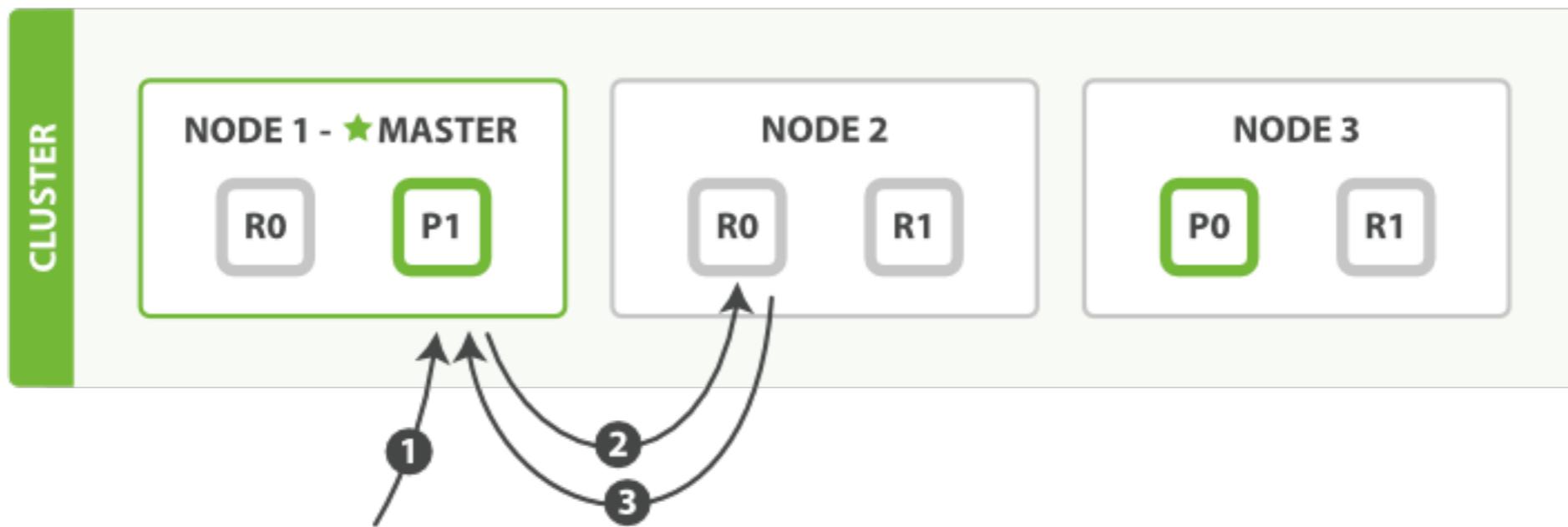
Node (≒ 서버) 1개 추가



# Distributed Operations

Index (데이터)를 shards(조각)로 세분화하여 여러 operations 성능 향상

심화

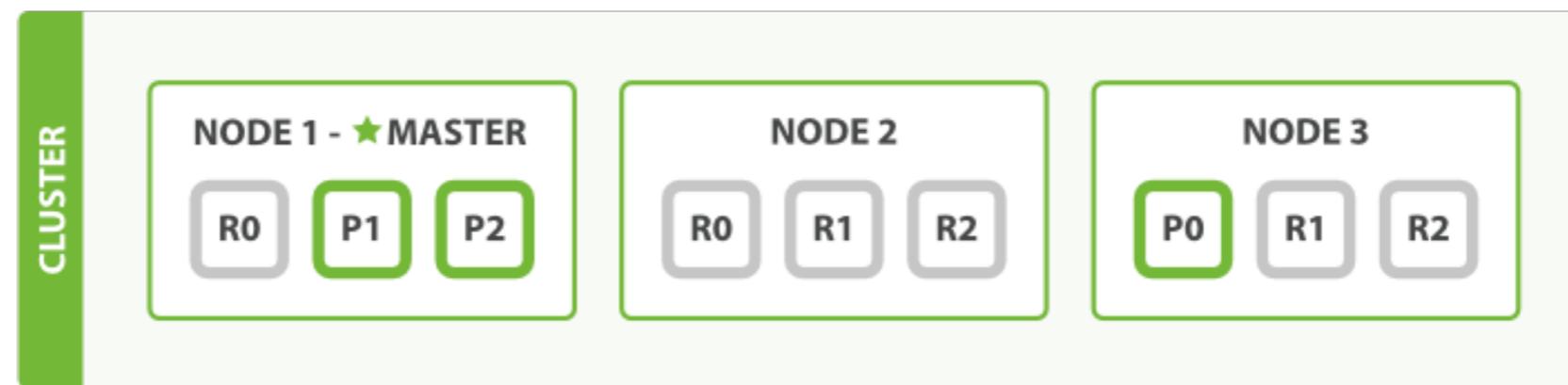


- ① User가 NODE1에 검색 request 전송
- ② NODE1은 NODE2에 request 전달
- ③ NODE2는 request 처리 후 NODE1에 결과 전송
- ④ NODE1는 User에게 결과 전송

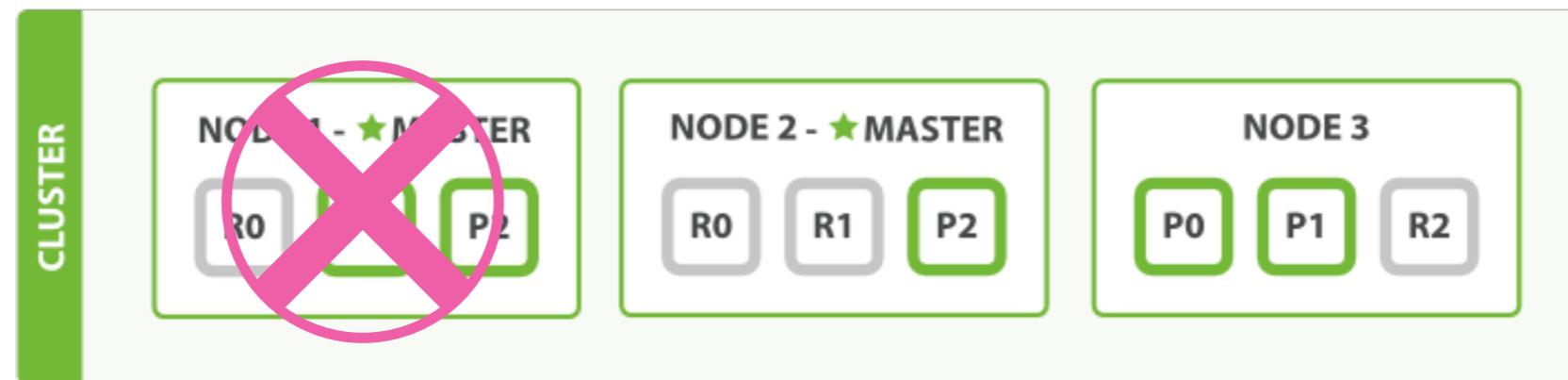
## Fault Tolerance

(Replica Shards를 설정을 통해) 특정 Node (≒서버)가 다운되어도 데이터 유실 없이 운영할 수 있다

심화



Node1이 다운되어도 Node2 & Node3 데이터로 백업 가능



용어 정리

RDBMS	<i>Elasticsearch</i>	Excel
Database	<i>Index</i>	Excel File
Table	<i>Type</i>	Sheet
Row	<i>Document</i>	Row
Column	<i>Field</i>	Column
Schema	<i>Mapping</i>	

- 위의 비교는 어디까지나 이해를 돋기 위함 목적일 뿐 정확히 일치하지는 않는다
- 6.0.0 이후에는 Index 1개에 Type 1개가 되어 사실상 폐지  (7.0 이후 완전 폐지)
- 최소한 Index, Document, Field, Mapping 은 제대로 알고 넘어가자!

## RDBMS

### Database



```
mysql> use Workbook1
Database changed
mysql> select * from Sheet1;
+-----+-----+-----+
| date | product | quantity | sales |
+-----+-----+-----+
| 2018-01-01 | Onepiece | 2 | 39000 |
| 2018-01-01 | Cardigan | 1 | 37000 |
| 2018-01-01 | Knit | 3 | 69000 |
| 2018-01-01 | Jeans | 1 | 78000 |
| 2018-01-01 | T-Shirt | 1 | 89000 |
| 2018-01-01 | Pants | 1 | 55000 |
| 2018-01-01 | Knit | 3 | 69000 |
| 2018-01-01 | Jeans | 1 | 78000 |
| 2018-01-01 | Coat | 1 | 149000 |
+-----+-----+-----+
```

## Elasticsearch

### Index



```
1 [ {
2   "took": 0,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 9,
12    "max_score": 1,
13    "hits": [
14      {
15        "_index": "workbook1",
16        "_type": "sheet1",
17        "_id": "5",
18        "_score": 1,
19        "_source": {
20          "date": "2018-01-01",
21          "product": "T-Shirt",
22          "quantity": 5,
23          "sales": 89000
24        }
25      },
26    ]
27  }
28 }
```

## Excel

### Excel File



	A	B	C	D	E	F	G	H
1								
2		date	product	quantity	sales			
3		01/01/2018	Onepice	2	39,000			
4		01/01/2018	Cardigan	1	37,000			
5		01/01/2018	Knit	3	69,000			
6		01/01/2018	Jeans	1	78,000			
7		01/01/2018	T-Shirt	5	89,000			
8		01/01/2018	Pants	1	55,000			
9		01/01/2018	Knit	3	69,000			
10		01/01/2018	Jeans	1	78,000			
11		01/01/2018	Coat	1	149,000			
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								

## RDBMS

### Table

```
mysql> use Workbook1
Database changed
mysql> select * from Sheet1;
+-----+-----+-----+-----+
| date | product | quantity | sales |
+-----+-----+-----+-----+
| 2018-01-01 | Onepiece | 2 | 39000 |
| 2018-01-01 | Cardigan | 1 | 37000 |
| 2018-01-01 | Knit | 3 | 69000 |
| 2018-01-01 | Jeans | 1 | 78000 |
| 2018-01-01 | T-Shirt | 1 | 89000 |
| 2018-01-01 | Pants | 1 | 55000 |
| 2018-01-01 | Knit | 3 | 69000 |
| 2018-01-01 | Jeans | 1 | 78000 |
| 2018-01-01 | Coat | 1 | 149000 |
+-----+-----+-----+-----+
```

## Elasticsearch

### Type

```
1 [{}]
2   "took": 0,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 9,
12    "max_score": 1,
13    "hits": [
14      {
15        "_index": "workbook1",
16        "_type": "sheet1", 
17        "_id": "5", 
18        "_score": 1,
19        "_source": {
20          "date": "2018-01-01",
21          "product": "T-Shirt",
22          "quantity": 5,
23          "sales": 89000
24        }
25      },
26    ]
27  }
```

## Excel

### Sheet

Workbook1					
	A	B	C	D	
1					
2		date	product	quantity	sales
3		01/01/2018	Onepice	2	39,000
4		01/01/2018	Cardigan	1	37,000
5		01/01/2018	Knit	3	69,000
6		01/01/2018	Jeans	1	78,000
7		01/01/2018	T-Shirt	5	89,000
8		01/01/2018	Pants	1	55,000
9		01/01/2018	Knit	3	69,000
10		01/01/2018	Jeans	1	78,000
11		01/01/2018	Coat	1	149,000
12					
13					
14					
15					
16					
17					

## RDBMS

### Row

```
mysql> use Workbook1
Database changed
mysql> select * from Sheet1;
+-----+-----+-----+
| date | product | quantity | sales |
+-----+-----+-----+
| 2018-01-01 | Onepiece | 2 | 39000 |
| 2018-01-01 | Cardigan | 1 | 37000 |
| 2018-01-01 | Knit | 3 | 69000 |
| 2018-01-01 | Jeans | 1 | 78000 |
| 2018-01-01 | T-Shirt | 1 | 89000 | T-Shirt
| 2018-01-01 | Pants | 1 | 55000 |
| 2018-01-01 | Knit | 3 | 69000 | Knit
| 2018-01-01 | Jeans | 1 | 78000 |
| 2018-01-01 | Coat | 1 | 149000 |
+-----+-----+-----+
```

## Elasticsearch

### Document

```
1 [{}]
2   "took": 0,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 9,
12    "max_score": 1,
13    "hits": [
14      {
15        "_index": "workbook1",
16        "_type": "sheet1",
17        "_id": "5",
18        "_score": 1,
19        "_source": {
20          "date": "2018-01-01",
21          "product": "T-Shirt",
22          "quantity": 5,
23          "sales": 89000
24        }
25      },
26    ]
27  }
```

## Excel

### Row

Workbook1								
	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								

## RDBMS

### Column

```
mysql> use Workbook1
Database changed
mysql> select * from Sheet1;
+-----+-----+-----+
| date | product | quantity | sales |
+-----+-----+-----+
| 2018-01-01 | Onepiece | 2 | 39000 |
| 2018-01-01 | Cardigan | 1 | 37000 |
| 2018-01-01 | Knit | 3 | 69000 |
| 2018-01-01 | Jeans | 1 | 78000 |
| 2018-01-01 | T-Shirt | 1 | 89000 |
| 2018-01-01 | Pants | 1 | 55000 |
| 2018-01-01 | Knit | 3 | 69000 |
| 2018-01-01 | Jeans | 1 | 78000 |
| 2018-01-01 | Coat | 1 | 149000 |
+-----+-----+-----+
```

## Elasticsearch

### Field

```
1 [{}]
2   "took": 0,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 9,
12    "max_score": 1,
13    "hits": [
14      {
15        "_index": "workbook1",
16        "_type": "sheet1",
17        "_id": "5",
18        "_score": 1,
19        "_source": {
20          "date": "2018-01-01",
21          "product": "T-Shirt",
22          "quantity": 5,
23          "sales": 89000
24        }
25      },
26    ]
27  }
```

## Excel

### Column

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								

## RDBMS

### Schema

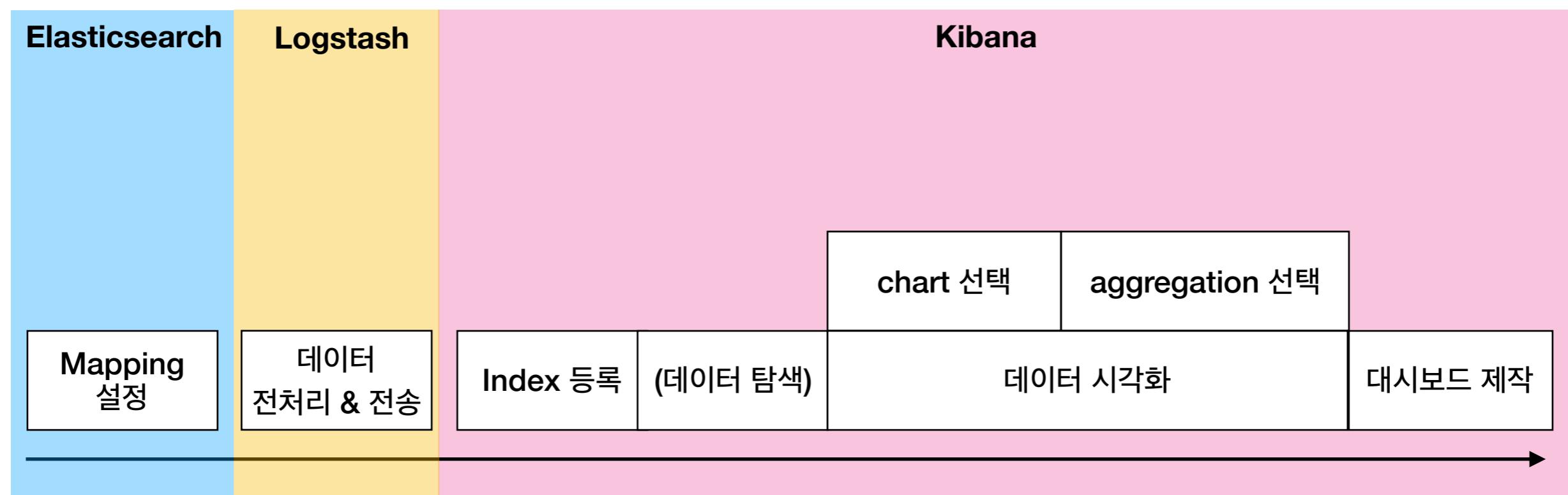
```
mysql> CREATE TABLE Sheet1 (
    -> date DATE,
    -> product VARCHAR(32),
    -> quantity INT(100),
    -> sales INT(100)
    -> );
```

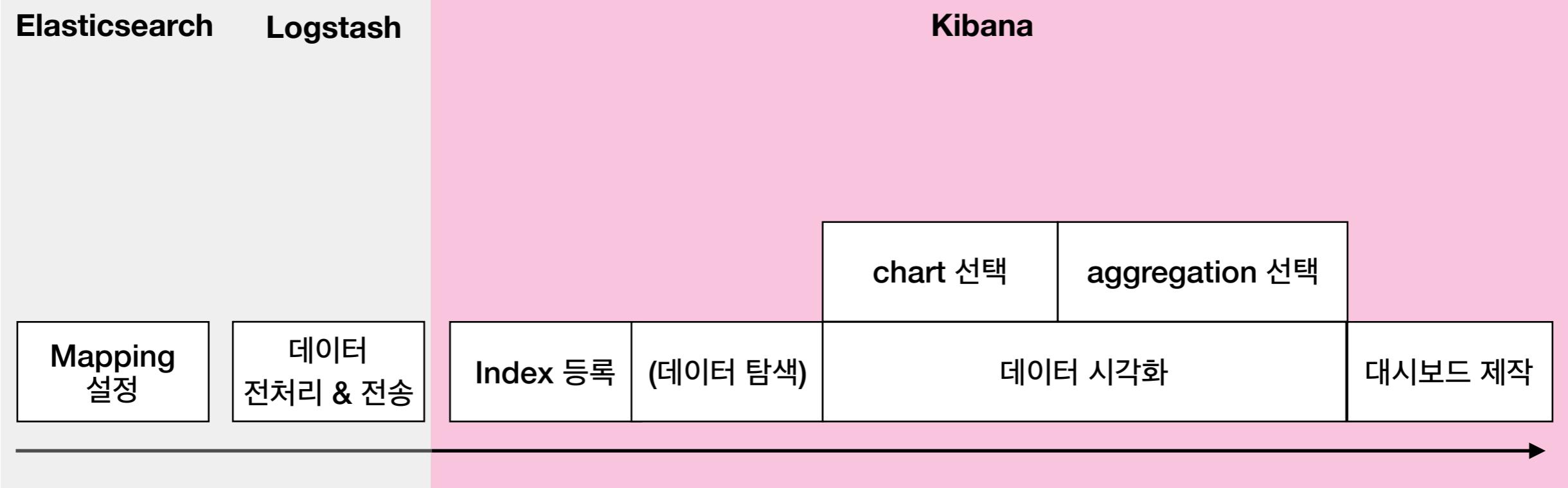
## Elasticsearch

### Mapping

```
PUT workbook1
{
  "mappings": {
    "sheet1": {
      "properties": {
        "date": {
          "type": "date"
        },
        "product": {
          "type": "keyword"
        },
        "quantity": {
          "type": "integer"
        },
        "sales": {
          "type": "integer"
        }
      }
    }
  }
}
```

## Elastic Stack Workflow





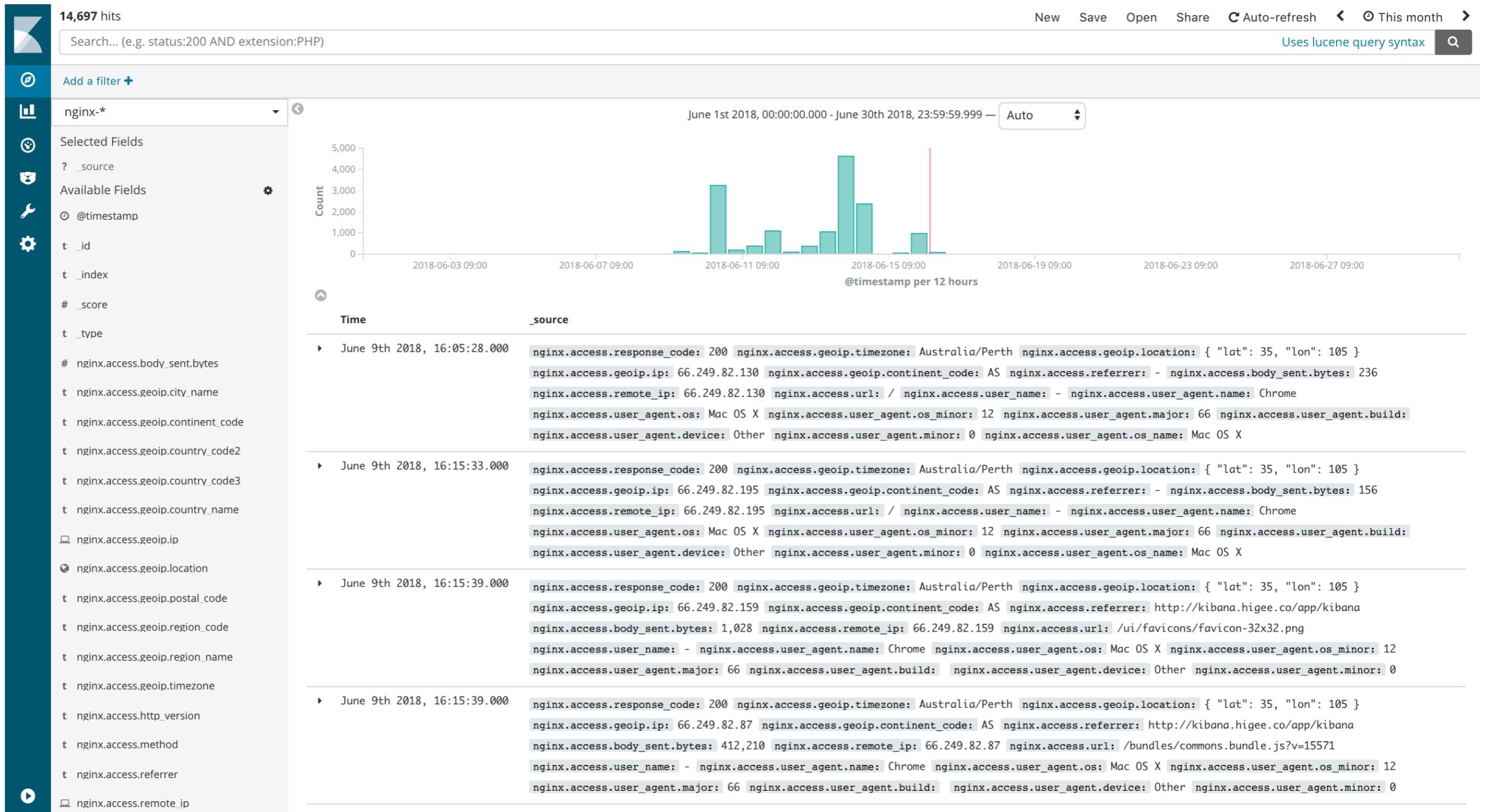
## Kibana 화면 소개

우선 Kibana에 접속해 보자

# Discover

데이터 검색 및 필터 등을 이용한 간단한 EDA 작업

클릭



손가락

# Visualize

Dashboard에 배치할 Visualization 생성

클릭



Visualize / New

Select visualization type

Search visualization types...

Basic Charts

- Area
- Heat Map
- Horizontal Bar
- Line
- Pie
- Vertical Bar

Data

- Data Table
- Gauge
- Goal
- Metric

Maps

- Coordinate Map
- Region Map

Time Series

- Timelion
- Visual Builder

# Dashboard

Visualize에서 생성한 Visualization을 이용한 Dashboard 생성

클릭

링크

날짜

검색

필터

설정

시작

Dashboard / nginx

Full screen Share Clone Edit Auto-refresh < This month >

Uses lucene query syntax

[nginx] markdown

Nginx Access Log Dashboard

```
66.249.82.131 - - [13/Jun/2018:17:01:02 +0000] "GET /ui/favicons/favicon-16x16.png HTTP/1.1" 304 0 "http://kibana.higee.co/app/kibana" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.87 Safari/537.36" 118.221.38.242 - - [13/Jun/2018:17:01:14 +0000] "POST /api/console/proxy?path=_aliases&method=GET HTTP/1.1" 200 107 "http://kibana.higee.co/app/kibana" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.87 Safari/537.36" 118.221.38.242 - - [13/Jun/2018:17:01:14 +0000] "POST /api/console/proxy?path=_mapping&method=GET HTTP/1.1" 200 974 "http://kibana.higee.co/app/kibana" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.87 Safari/537.36" 66.249.82.131 - - [13/Jun/2018:17:01:16 +0000] "GET /ui/favicons/favicon-32x32.png HTTP/1.1" 304 0 "http://kibana.higee.co/app/kibana" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.87 Safari/537.36" 66.249.82.129 - - [13/Jun/2018:17:01:17 +0000] "GET /ui/favicons/favicon-16x16.png HTTP/1.1" 304 0 "http://kibana.higee.co/app/kibana" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.87 Safari/537.36"
```

[nginx] search

Time	nginx.access.url	nginx.access.response_code	nginx.access.body_sent.bytes
▶ June 10th 2018, 22:49:58.000	/bundles/vendors.bundle.js?v=16627	200	2,018,434
▶ June 9th 2018, 16:28:52.000	/bundles/kibana.bundle.js?v=15571	200	1,662,257
▶ June 9th 2018, 16:15:41.000	/bundles/kibana.bundle.js?v=15571	200	1,662,123
▶ June 9th 2018, 16:15:41.000	/bundles/kibana.bundle.js?v=15571	200	1,662,046
▶ June 9th 2018, 16:15:41.000	/bundles/kibana.bundle.js?v=15571	200	1,661,797

[nginx] heat map

시간대

요일

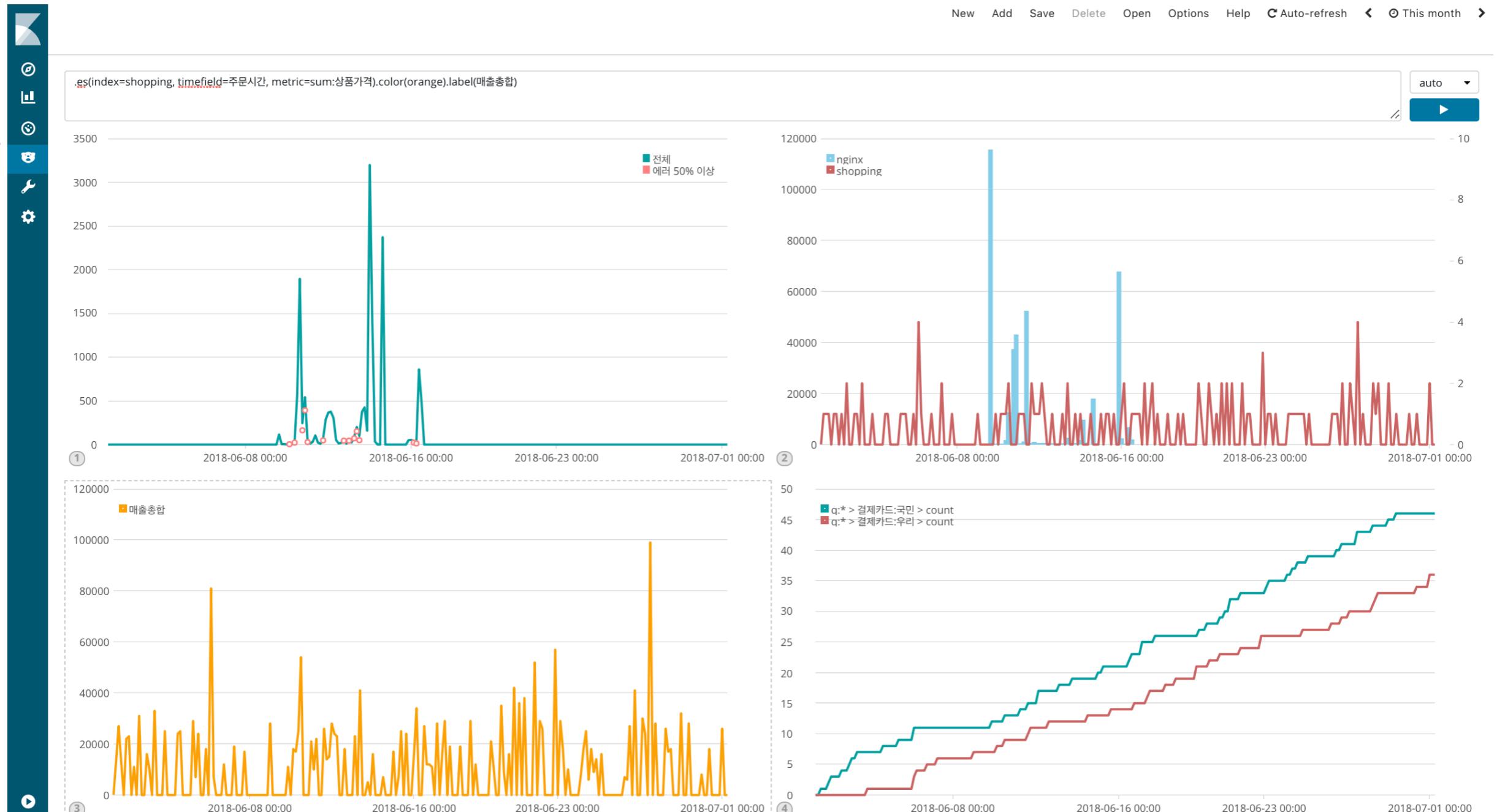
[nginx] tag-cloud

Mac OS X  
Windows 10 Other iOS  
Windows 7

14,754 req

# Timelion

## Visualization의 하나인 (시계열에 특화된) Timelion 생성



클릭  
▶

# Dev Tools

## Elasticsearch REST API를 위한 UI

클릭



The screenshot shows the Elasticsearch Dev Tools interface with the 'Console' tab selected. On the left, there's a sidebar with various icons: a Kibana logo, a magnifying glass, a bar chart, a clock, a gear, and a wrench. A pink hand icon points to the wrench icon. Another pink hand icon points to the right.

The main area has two panes. The left pane displays a complex Elasticsearch search query (highlighted in blue) with line numbers from 1 to 51. The right pane shows the resulting JSON response with line numbers from 1 to 53. The results include document details like '\_index', '\_type', '\_id', '\_score', and '\_source' fields, along with various metadata and field values.

```
1 GET shopping/_search
2 {
3   "query": {
4     "bool": {
5       "should": [
6         {
7           "bool": {
8             "must": [
9               {
10                  "range": {
11                    "고객나이": {
12                      "gte": 27,
13                      "lte": 35
14                    }
15                  }
16                },
17                {
18                  "wildcard": {
19                    "고객주소_시도": {
20                      "value": "경??"
21                    }
22                  }
23                }
24              ],
25            },
26            {
27              "bool": {
28                "must": [
29                  {
30                    "term": {
31                      "결제카드": {
32                        "value": "시티"
33                      }
34                    }
35                  ],
36                  "must_not": [
37                    {
38                      "prefix": {
39                        "결제카드": "하나"
40                      }
41                    }
42                  ]
43                }
44              }
45            ],
46            "minimum_should_match": 1
47          }
48        }
49      }
50    }
51 }
```

```
1 {
2   "took": 4,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 520,
12    "max_score": 5.0557966,
13    "hits": [
14      {
15        "_index": "shopping",
16        "_type": "shopping",
17        "_id": "VpJd_WMBByNsCKuKnHrYc",
18        "_score": 5.0557966,
19        "_source": {
20          "접수번호": 4144,
21          "주문시간": "2017-05-10T09:25:53",
22          "수령시간": "2017-05-12T15:06:53",
23          "예약여부": "일반",
24          "배송메모": "상품 이상",
25          "고객ip": "38.157.29.35",
26          "고객성별": "여성",
27          "고객나이": 33,
28          "물건좌표": "36.777264394697816, 127.82310387941406",
29          "고객주소_시도": "경기도",
30          "구매사이트": "옥션",
31          "판매자평점": 3,
32          "상품분류": "니트",
33          "상품가격": 20000,
34          "상품개수": 7,
35          "결제카드": "시티"
36        }
37      },
38      {
39        "_index": "shopping",
40        "_type": "shopping",
41        "_id": "kpJU_WMBByNsCKuKnVp0u",
42        "_score": 3.2064722,
43        "_source": {
44          "접수번호": 758,
45          "주문시간": "2018-06-21T19:57:34",
46          "수령시간": "2018-06-23T22:08:34",
47          "예약여부": "일반",
48          "배송메모": "관리실에 맡김",
49          "고객ip": "6.231.54.4",
50          "고객성별": "남성",
51          "고객나이": 41,
52          "물건좌표": "36.11621638917374, 127.66787855345112",
53          "고객주소_시도": "서울특별시",
          "고객성별": "남성"
        }
54      }
    ]
  }
}
```

# Management

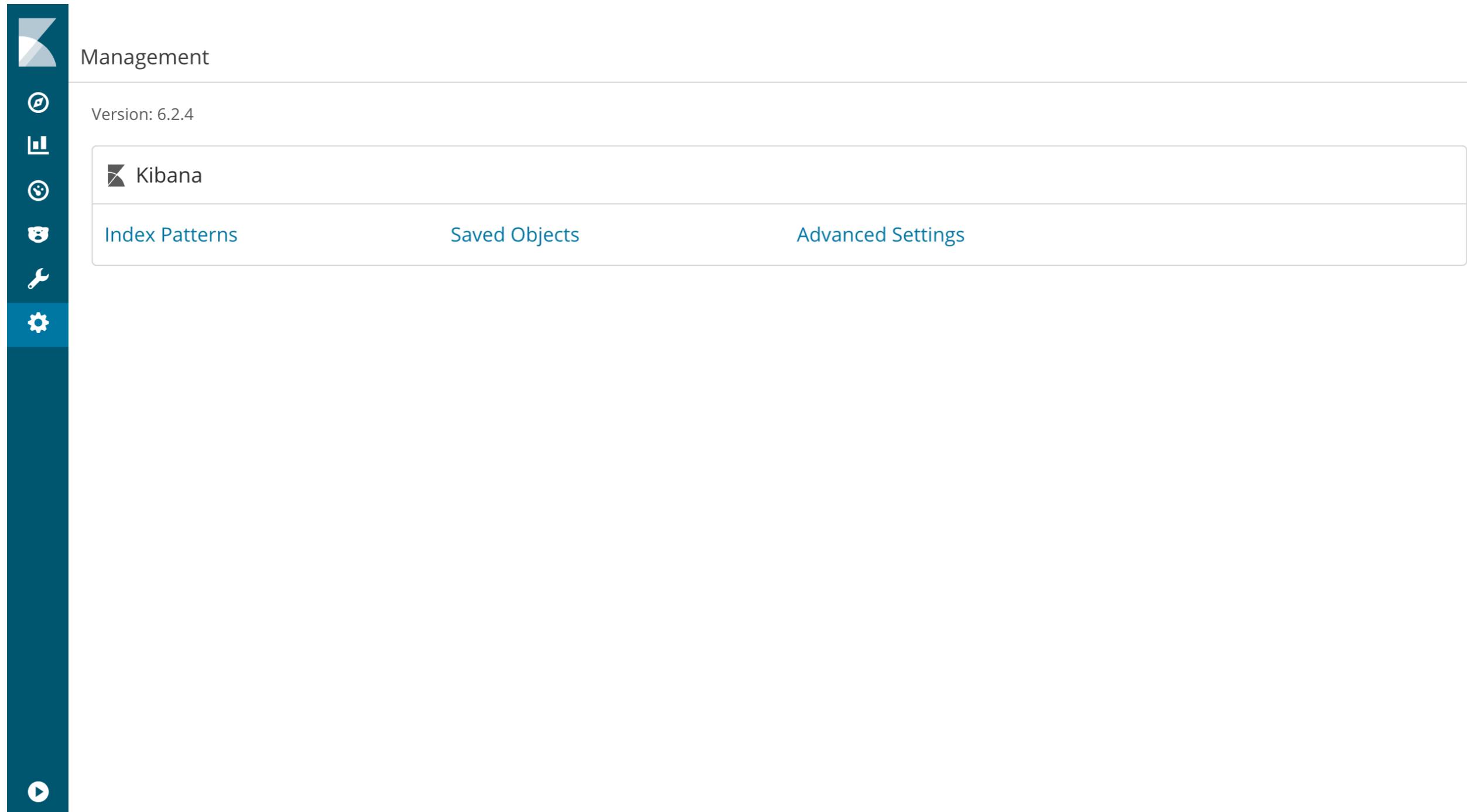
## Kibana 설정 수정

Management

Version: 6.2.4

 Kibana

[Index Patterns](#)   [Saved Objects](#)   [Advanced Settings](#)

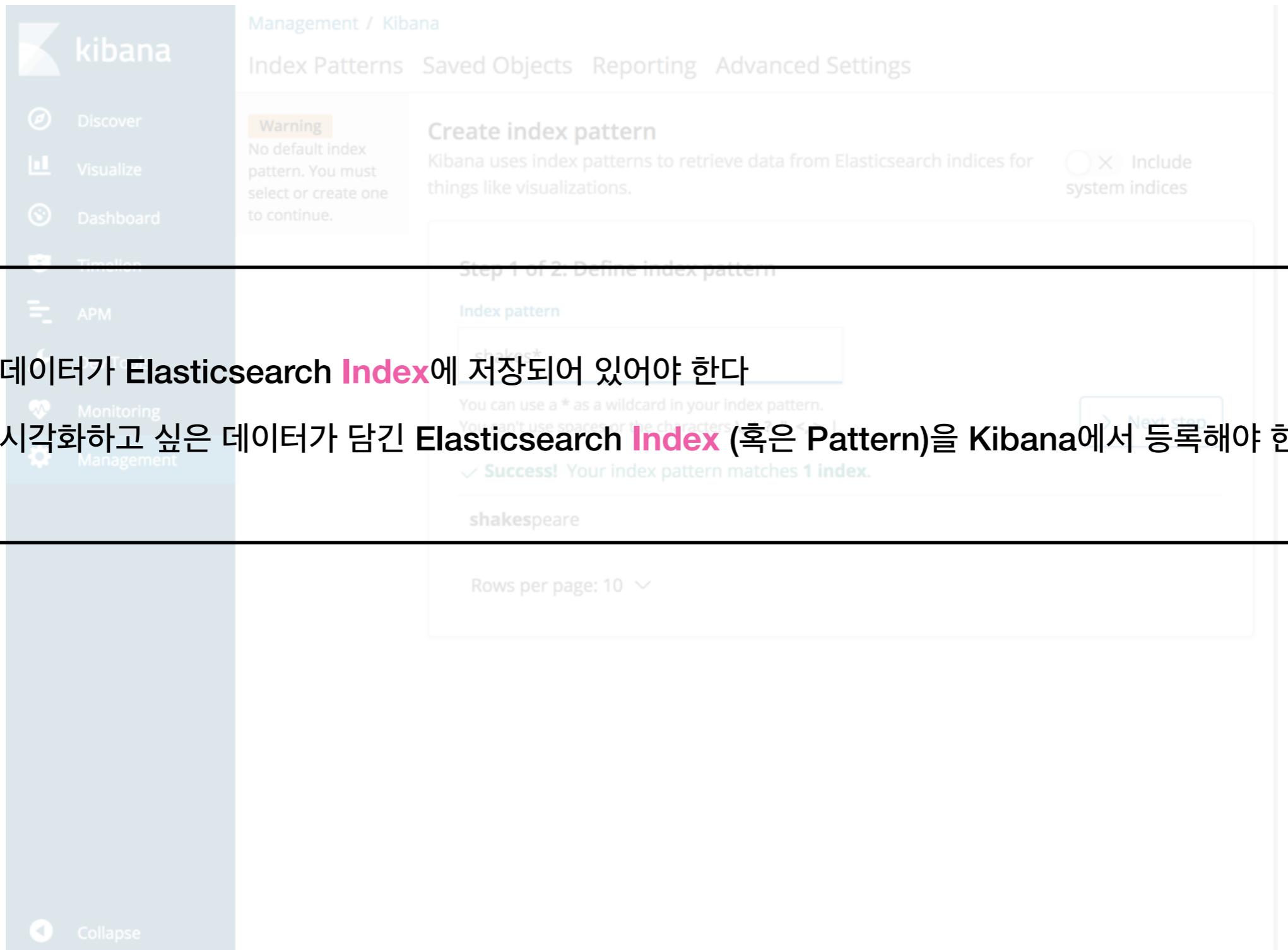


**Index 등록**

## Management 페이지로 이동하자

The screenshot shows the Elasticsearch Management interface. On the left is a vertical sidebar with icons for various management sections: Management (selected), Index Patterns, Saved Objects, Advanced Settings, Visualize, Discover, Settings, and Help. The main area displays the 'Management' section, which includes the version information 'Version: 6.2.4' and a 'Kibana' section. Below these are three tabs: 'Index Patterns' (selected), 'Saved Objects', and 'Advanced Settings'. A pink hand icon points to the 'Management' icon in the sidebar.

## Kibana 시작화의 전제 조건



The screenshot shows the Kibana management interface. On the left sidebar, there are several options: Discover, Visualize, Dashboard, Timeline, APM, Monitoring, and Management. The Management option is currently selected. At the top, there are tabs for Index Patterns, Saved Objects, Reporting, and Advanced Settings. Below these, a warning message states: "Warning: No default index pattern. You must select or create one to continue." To the right of the warning, there is a checkbox labeled "Include system indices" with a checked status. The main content area is titled "Create index pattern" and contains the sub-section "Step 1 of 2: Define index pattern". It features an input field for the index pattern name, which has "shakespeare" typed into it. Below the input field, there is explanatory text: "You can use a \* as a wildcard in your index pattern. You can't use spaces or the characters < > | !". A success message at the bottom of the input field says: "Success! Your index pattern matches 1 index." A "Next step" button is located to the right of the input field.

- 데이터가 Elasticsearch **Index**에 저장되어 있어야 한다
- 시각화하고 싶은 데이터가 담긴 Elasticsearch **Index** (혹은 Pattern)을 Kibana에서 등록해야 한다

## Index Patterns 등록 1단계 - Elasticsearch Index (Patterns) 등록

1. Kibana -> Management -> Index Patterns -> Create Index Pattern
2. 시각화 하고 싶은 Index (Patterns) 입력, 예) Index 이름 : shopping
3. Next step 클릭

### Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

Include system indices

#### Step 1 of 2: Define index pattern

Index pattern

shopping

You can use a \* as a wildcard in your index pattern.

You can't use empty spaces or the characters \, /, ?, ", <, >, |.

> Next step

✓ Success! Your index pattern matches 1 index.

올바르게 입력하면 왼쪽처럼 나온다

shopping

## **Index Pattern 등록 2단계 - 해당 Index에서 기준 시간으로 사용할 Time Field 선택**

1. Timer Filter field name 클릭

2. 표시되는 Date Field 중에서 기준 시간으로 사용할 Field 선택

-> **Index Pattern 등록 완료 후 여기서 선택한 Time Field를 기준으로 데이터 정렬 및 필터**

-> 단, Time Filter가 필요 없는 Index의 경우 “I don't want to use the Time Filter”를 선택

3. Create index pattern 클릭

### Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

#### Step 2 of 2: Configure settings

You've defined **shopping** as your index pattern. Now you can specify some settings before we create it.

Time Filter field name

Refresh

수령시간  
 주문시간

I don't want to use the Time Filter

.....> 2. 목적에 맞게 1개 선택

> Show advanced options

Back

Create index pattern

3. 선택

## 방금 배운 내용을 직접 해보자

\* Elasticsearch Index는 실습용으로 사전에 생성

호호	elasticsearch index	time field	kibana index pattern
1	{id}_2018.08.12	주문시간	ex) higee_2018.08.12
2	{id}_2018.08.13	주문시간	ex) higee_2018.08.13
3	{id}_2018.08.14		ex) higee_2018.08.14



- id
  - 캠퍼 수강시 등록한 이메일 주소 (도메인 제거)
  - higee@fastcampus.com → higee

## Time Field를 선택한 경우 - higee 2018.08.12, higee 2018.08.13

(바로 다음에 배울) Discover에서 보면 아래와 같이 나온다

The screenshot shows the Elasticsearch Discover interface with the following details:

- Top Bar:** Shows "10 hits", a search bar with placeholder "(e.g. status:200 AND extension:PHR)", and various navigation buttons like New, Save, Open, Share, Auto-refresh, and a date range selector set to "This week". A pink hand icon points to the date range selector.
- Left Sidebar:** Lists "Selected Fields" and "Available Fields". The selected field is "higee\_2018.08.12". A pink hand icon points to the "Available Fields" section.
- Central Area:**
  - A histogram titled "Count" showing the number of documents per day from August 12th to August 18th. A specific bar for August 12th is highlighted with a pink border and a callout box indicating "Count 10 주문시간 per day 2018-08-12".
  - A table of search results for "Time" (August 12th) and "\_source". Each result row contains a timestamp, a list of document details, and a "score" value. A pink hand icon points to the first result row.

## Time Field를 선택하지 않은 경우 - higee 2018.08.14

(바로 다음에 배울) Discover에서 보면 아래와 같이 나온다

10 hits

New Save Open Share

Search... (e.g. status:200 AND extension:PHP)

Uses lucene query syntax

Add a filter +

Selected Fields

? \_source

Available Fields

t \_id

t \_index

# \_score

t \_type

t 결제카드

□ 고객ip

# 고객나이

t 고객성별

t 고객주소\_시도

t 구매사이트

⌚ 물건좌표

t 배송메모

# 상품가격

# 상품개수

t 상품분류

⌚ 수령시간

t 예약여부

# 접수번호

⌚ 주문시간

higee\_2018.08.14

\_source

- 접수번호: 7 주문시간: August 14th 2018, 22:51:38.000 수령시간: August 19th 2018, 20:44:38.000 예약여부: 일반 배송메모: 관리실에 맡김 고객ip: 232.39.144.64 고객성별: 여성 고객나이: 4 물건좌표: 36.3545882287262, 127.69302768088149 고객주소\_시도: 서울특별시 구매사이트: 11번가 판매자평점: 4 상품분류: 가디건 상품가격: 18,000 상품개수: 1 결제카드: 국민 \_id: 6qq3LWUBME B1QWBFB0BU \_type: higee\_2018.08.14 \_index: higee\_2018.08.14 \_score: -
- 접수번호: 8 주문시간: August 14th 2018, 22:50:56.000 수령시간: August 16th 2018, 20:32:56.000 예약여부: 예약 배송메모: 주소 오류 고객ip: 196.27.135.161 고객성별: 남성 고객나이: 26 물건좌표: 36.36295508892756, 127.25952572750205 고객주소\_시도: 전라남도 구매사이트: g마켓 판매자평점: 3 상품분류: 코트 상품가격: 7,000 상품개수: 1 결제카드: 우리 \_id: 66q3LWUBMEB1QWBFB0Bc \_type: higee\_2018.08.14 \_index: higee\_2018.08.14 \_score: -
- 접수번호: 6 주문시간: August 14th 2018, 22:41:16.000 수령시간: August 19th 2018, 07:18:16.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 248.108.137.152 고객성별: 여성 고객나이: 33 물건좌표: 35.834332272376514, 126.08112936142254 고객주소\_시도: 충청북도 구매사이트: 11번가 판매자평점: 3 상품분류: 스웨터 상품가격: 9,000 상품개수: 7 결제카드: 하나 \_id: 6aq3LWUBMEB1QWBFB0BL \_type: higee\_2018.08.14 \_index: higee\_2018.08.14 \_score: -
- 접수번호: 5 주문시간: August 14th 2018, 20:00:38.000 수령시간: August 15th 2018, 21:13:38.000 예약여부: 일반 배송메모: 부재중 고객ip: 119.231.92.238 고객성별: 남성 고객나이: 58 물건좌표: 35.4349595430256, 127.77210918373116 고객주소\_시도: 세종특별자치시 구매사이트: 11번가 판매자평점: 3 상품분류: 셔츠 상품가격: 23,000 상품개수: 1 결제카드: 우리 \_id: 6Kq3LWUBMEB1QWBFB0BD \_type: higee\_2018.08.14 \_index: higee\_2018.08.14 \_score: -
- 접수번호: 4 주문시간: August 14th 2018, 19:56:32.000 수령시간: August 15th 2018, 16:12:32.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 32.227.140.43 고객성별: 여성 고객나이: 30 물건좌표: 36.06009629486569, 128.08612427860692 고객주소\_시도: 충청남도 구매사이트: 옥션 판매자평점: 1 상품분류: 티셔츠 상품가격: 7,000 상품개수: 1 결제카드: 우리 \_id: 56q3LWUBMEB1QWBFB0A8 \_type: higee\_2018.08.14 \_index: higee\_2018.08.14 \_score: -
- 접수번호: 10 주문시간: August 14th 2018, 16:34:47.000 수령시간: August 15th 2018, 11:44:47.000 예약여부: 일반 배송메모: 환불 요청 고객ip: 131.154.247.184 고객성별: 남성 고객나이: 6 물건좌표: 36.923985211864995, 128.71955943571606 고객주소\_시도: 경상남도 구매사이트: g마켓 판매자평점: 5 상품분류: 남방 상품가격: 11,000 상품개수: 1 결제카드: 우리 \_id: 7aq3LWUBMEB1QWBFB0Br \_type: higee\_2018.08.14 \_index: higee\_2018.08.14 \_score: -
- 접수번호: 3 주문시간: August 14th 2018, 15:33:38.000 수령시간: August 16th 2018, 00:47:38.000 예약여부: 예약 배송메모: 상품 이상 고객ip: 183.196.225.165 고객성별: 여성 고객나이: 36 물건좌표: 35.32958090825842, 127.51780035179722 고객주소\_시도: 세종특별자치시 구매사이트: 위메프 판매자평점: 2 상품분류: 셔츠 상품가격: 15,000 상품개수: 1 결제카드: 우리 \_id: 5qq3LWUBME B1QWBFB0Aw \_type: higee\_2018.08.14 \_index: higee\_2018.08.14 \_score: -
- 접수번호: 2 주문시간: August 14th 2018, 08:16:12.000 수령시간: August 17th 2018, 19:34:12.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 150.207.141.224 고객성별: 여성 고객나이: 40 물건좌표: 35.500921728484236, 127.44515567736283 고객주소\_시도: 서울특별시 구매사이트: 11번가 판매자평점: 1 상포부르 셔츠 상포가격: 21,000 상포개수: 1 결제카드: 구미 \_id: 5qq3LWUBMEF1

## Index Patterns - Wildcard 사용

### Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.



Include system indices

#### Step 1 of 2: Define index pattern

Index pattern

shopping

You can use a \* as a wildcard in your index pattern.  
You can't use empty spaces or the characters \, /, ?, ", <, >, |.

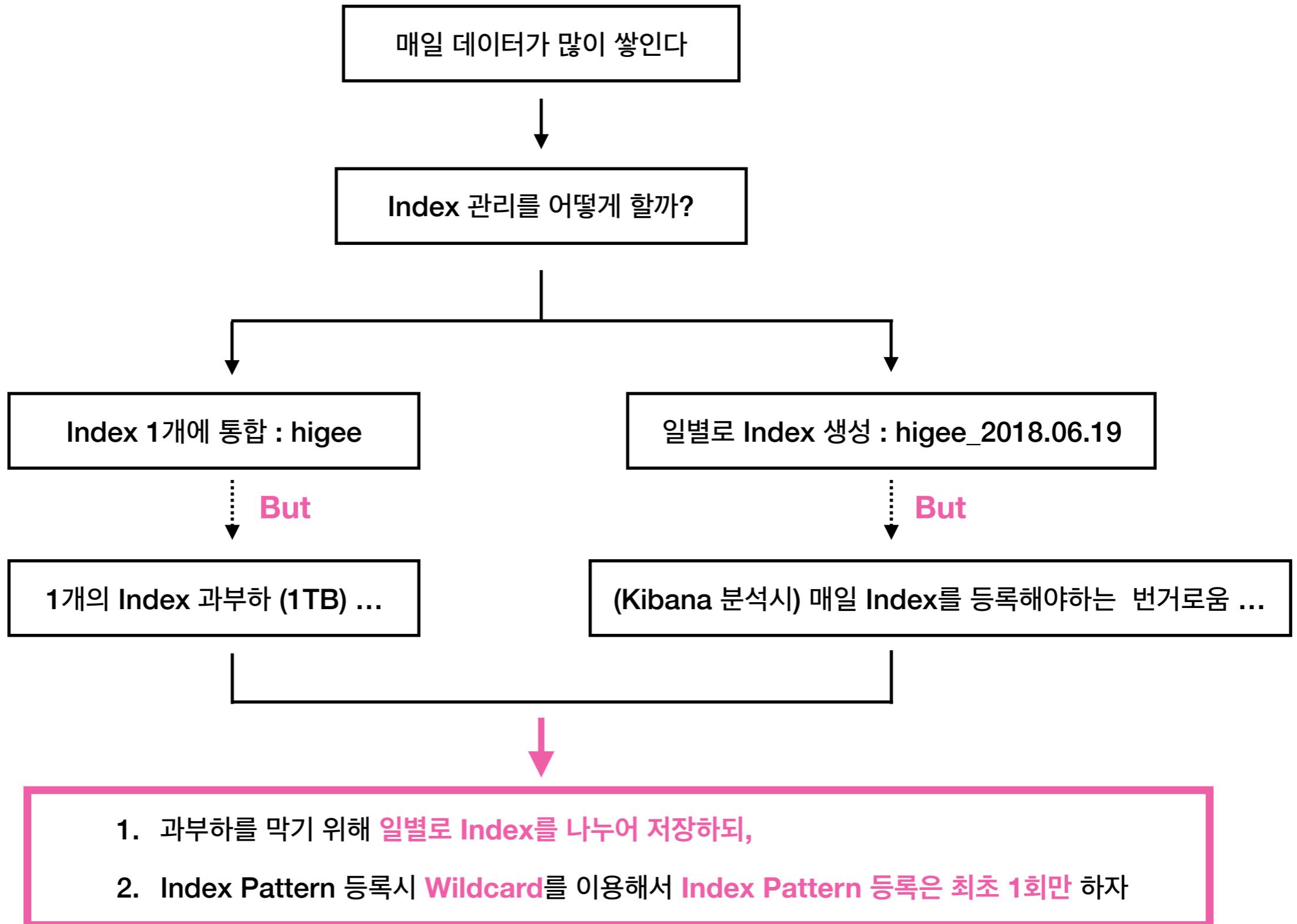
✓ Success! Your index pattern matches 1 index.

shopping

**Index Pattern을 등록하는데  
Wildcard(\*)가 왜 필요할까?**

> Next step

## Index Patterns - Wildcard 사용



## Index Patterns - Wildcard 사용

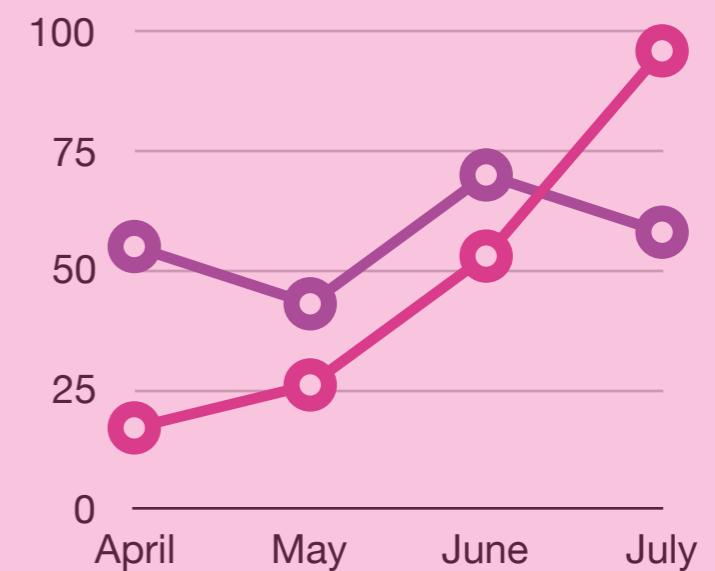
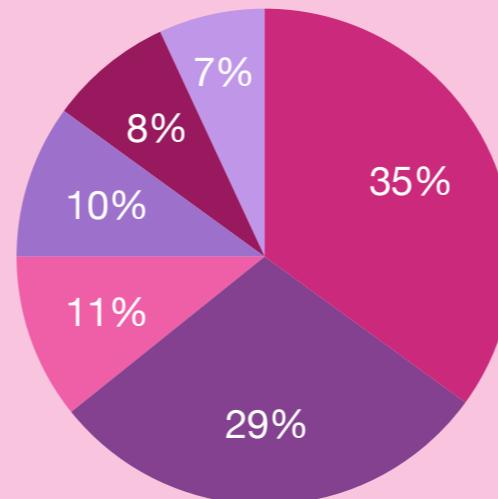
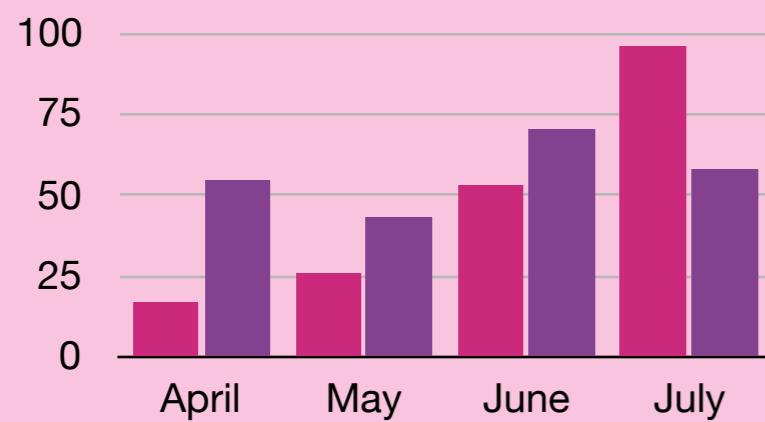
higee\_\*

데이터 저장은 분산, 검색 및 시각화는 통합

higee\_2018.08.12

higee\_2018.08.13

### Kibana



## 방금 배운 내용을 직접 해보자

번호	<i>elasticsearch index</i>	time field	<i>kibana index pattern</i>
1	{id}_2018.08.12 {id}_2018.08.13 {id}_2018.08.14	주문시간	ex) higee_*

## Time Field를 선택한 경우 - higee 2018.08.12, higee 2018.08.13

(바로 다음에 배울) Discover에서 보면 아래와 같이 나온다



30 hits

New Save Open Share C Auto-refresh < > This week

Search... (e.g. status:200 AND extension:PHP)

Add a filter +

Selected Fields: higee\_\*

Available Fields: t\_id, t\_index, #\_score, t\_type, t\_결제카드, #\_고객ip, #\_고객나이, t\_고객성별, t\_고객주소\_시도, t\_구매사이트, #\_물건좌표, t\_배송메모, #\_상품가격, #\_상품개수, t\_상품분류, #\_수령시간, t\_예약여부, #\_접수번호, #\_주문시간

Count: August 12th 2018, 00:00:00.000 - August 18th 2018, 23:59:59.999 — Daily

August 12th 2018, 00:00:00.000 - August 18th 2018, 23:59:59.999 — Daily

Count: 10  
2018-08-12 2018-08-12 2018-08-13 2018-08-14 2018-08-15 2018-08-16 2018-08-17 2018-08-18  
주문시간 per day

Time: source

- ▶ August 14th 2018, 22:51:38.000 접수번호: 7 주문시간: August 14th 2018, 22:51:38.000 수령시간: August 19th 2018, 20:44:38.000 예약여부: 일반 배송메모: 관리실에 맡김 고객ip: 232.39.144.64 고객성별: 여성 고객나이: 45 물건좌표: 36.3545882287262, 127.69302768088149 고객주소\_시도: 서울특별시 구매사이트: 11번가 판매자평점: 4 상품분류: 가디건 상품가격: 18,000 상품개수: 1 결제카드: 국민 \_id: 6qq3LWUBMEB1QWBFb0BU \_type: higee\_2018.08.14 \_index: higee\_2018.08.14 \_score: -
- ▶ August 14th 2018, 22:50:56.000 접수번호: 8 주문시간: August 14th 2018, 22:50:56.000 수령시간: August 16th 2018, 20:32:56.000 예약여부: 예약 배송메모: 주소 오류 고객ip: 196.27.135.161 고객성별: 남성 고객나이: 26 물건좌표: 36.36295508892756, 127.25952572750205 고객주소\_시도: 전라남도 구매사이트: g마켓 판매자평점: 3 상품분류: 코트 상품가격: 7,000 상품개수: 1 결제카드: 우리 \_id: 66q3LWUBMEB1QWBFb0Bc \_type: higee\_2018.08.14 \_index: higee\_2018.08.14 \_score: -
- ▶ August 14th 2018, 22:41:16.000 접수번호: 6 주문시간: August 14th 2018, 22:41:16.000 수령시간: August 19th 2018, 07:18:16.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 248.10.8.137.152 고객성별: 여성 고객나이: 33 물건좌표: 35.834332272376514, 126.08112936142254 고객주소\_시도: 충청북도 구매사이트: 11번가 판매자평점: 3 상품분류: 스웨터 상품가격: 9,000 상품개수: 7 결제카드: 하나 \_id: 6aq3LWUBMEB1QWBFb0BL \_type: higee\_2018.08.14 \_index: higee\_2018.08.14 \_score: -
- ▶ August 14th 2018, 20:00:38.000 접수번호: 5 주문시간: August 14th 2018, 20:00:38.000 수령시간: August 15th 2018, 21:13:38.000 예약여부: 일반 배송메모: 부재중 고객ip: 119.231.9.2.238 고객성별: 남성 고객나이: 58 물건좌표: 35.4349595430256, 127.77210918373116 고객주소\_시도: 세종특별자치시 구매사이트: 11번가 판매자평점: 3 상품분류: 셔츠 상품가격: 23,000 상품개수: 1 결제카드: 우리 \_id: 6Kq3LWUBMEB1QWBFb0BD \_type: higee\_2018.08.14 \_index: higee\_2018.08.14 \_score: -
- ▶ August 14th 2018, 19:56:32.000 접수번호: 4 주문시간: August 14th 2018, 19:56:32.000 수령시간: August 15th 2018, 16:12:32.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 32.227.

데이터 탐색

## Discover 페이지로 이동하자

2,417 hits

Search... (e.g. status:200 AND extension:PHP)

New Save Open Share Auto-refresh Today < Today >

Add a filter +

nginx-\*

Selected Fields

- ? \_source
- Available Fields
- @timestamp
- t \_id
- t \_index
- # \_score
- t \_type
- # nginx.access.body\_sent.bytes
- t nginx.access.geoip.city\_name
- t nginx.access.geoip.continent ... **add**
- t nginx.access.geoip.country\_code
- t nginx.access.geoip.country\_na...
- t nginx.access.geoip.ip
- nginx.access.geoip.location
- t nginx.access.geoip.region\_code
- t nginx.access.geoip.region\_na...
- t nginx.access.geoip.timezone
- t nginx.access.http\_version
- t nginx.access.method
- t nginx.access.referrer

August 12th 2018, 00:00:00.000 - August 12th 2018, 23:59:59.999 — Hourly

Count

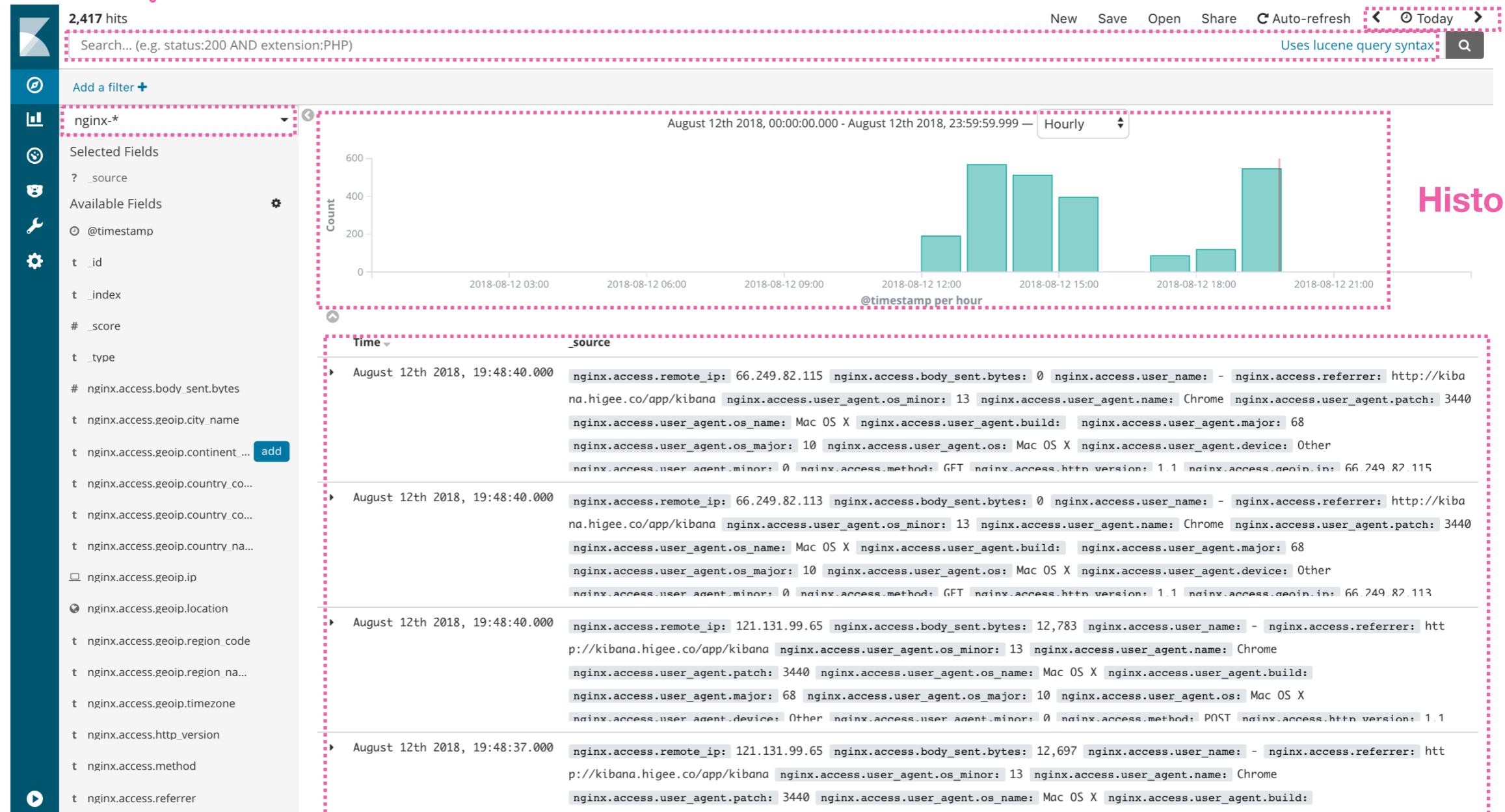
Time

\_source

Time	_source
August 12th 2018, 19:48:40.000	nginx.access.remote_ip: 66.249.82.115 nginx.access.body_sent.bytes: 0 nginx.access.user_name: - nginx.access.referrer: http://kibana.higee.co/app/kibana nginx.access.user_agent.os_minor: 13 nginx.access.user_agent.name: Chrome nginx.access.user_agent.patch: 3440 nginx.access.user_agent.os_name: Mac OS X nginx.access.user_agent.build: nginx.access.user_agent.major: 68 nginx.access.user_agent.os_major: 10 nginx.access.user_agent.os: Mac OS X nginx.access.user_agent.device: Other nginx.access.user_agent.minor: 0 nginx.access.method: GET nginx.access.http_version: 1.1 nginx.access.geoip.ip: 66.249.82.115
August 12th 2018, 19:48:40.000	nginx.access.remote_ip: 66.249.82.113 nginx.access.body_sent.bytes: 0 nginx.access.user_name: - nginx.access.referrer: http://kibana.higee.co/app/kibana nginx.access.user_agent.os_minor: 13 nginx.access.user_agent.name: Chrome nginx.access.user_agent.patch: 3440 nginx.access.user_agent.os_name: Mac OS X nginx.access.user_agent.build: nginx.access.user_agent.major: 68 nginx.access.user_agent.os_major: 10 nginx.access.user_agent.os: Mac OS X nginx.access.user_agent.device: Other nginx.access.user_agent.minor: 0 nginx.access.method: GET nginx.access.http_version: 1.1 nginx.access.geoip.ip: 66.249.82.113
August 12th 2018, 19:48:40.000	nginx.access.remote_ip: 121.131.99.65 nginx.access.body_sent.bytes: 12,783 nginx.access.user_name: - nginx.access.referrer: http://kibana.higee.co/app/kibana nginx.access.user_agent.os_minor: 13 nginx.access.user_agent.name: Chrome nginx.access.user_agent.patch: 3440 nginx.access.user_agent.os_name: Mac OS X nginx.access.user_agent.build: nginx.access.user_agent.major: 68 nginx.access.user_agent.os_major: 10 nginx.access.user_agent.os: Mac OS X nginx.access.user_agent.device: Other nginx.access.user_agent.minor: 0 nginx.access.method: POST nginx.access.http_version: 1.1
August 12th 2018, 19:48:37.000	nginx.access.remote_ip: 121.131.99.65 nginx.access.body_sent.bytes: 12,697 nginx.access.user_name: - nginx.access.referrer: http://kibana.higee.co/app/kibana nginx.access.user_agent.os_minor: 13 nginx.access.user_agent.name: Chrome nginx.access.user_agent.patch: 3440 nginx.access.user_agent.os_name: Mac OS X nginx.access.user_agent.build:

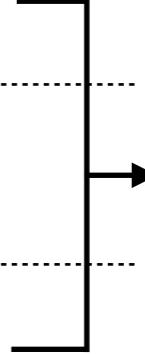
# 용어를 살펴보자

Query Bar

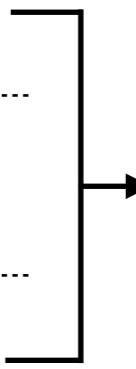
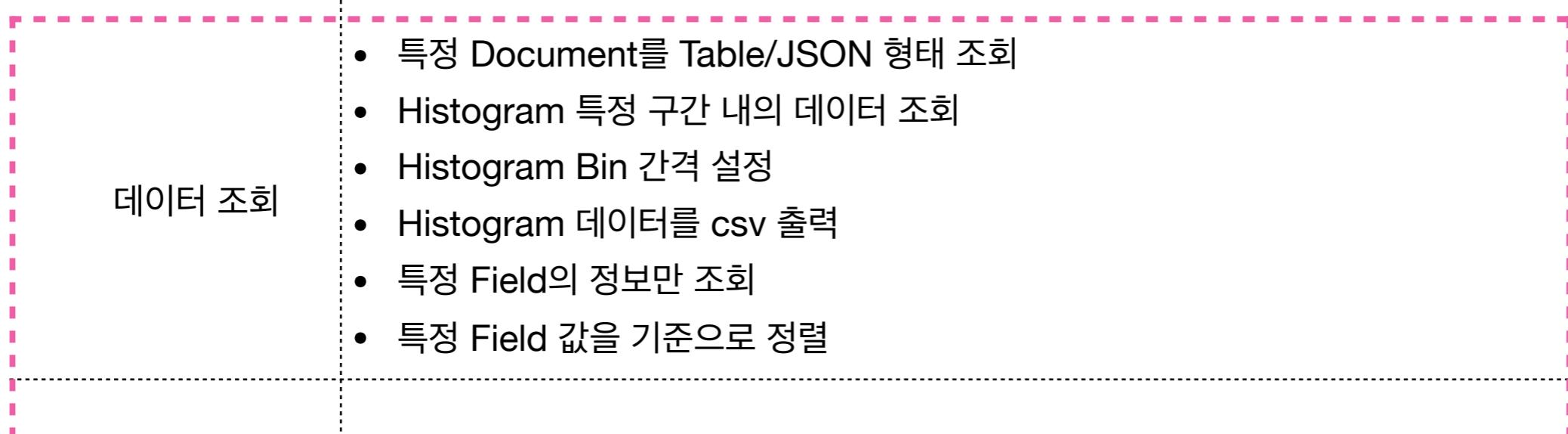


Document Table

## Discover에서는 어떤 작업을 할 수 있을까?

주요 기능	세부 기능	
데이터 검색	(복잡한) 조건을 만족하는 데이터 선별적 탐색 가능	
데이터 검색 저장	검색한 결과를 저장하여 Visualize에서 사용	 “검색 및 필터” 학습 후
데이터 필터링	(간단한) 특정 조건을 만족하는 데이터 선별적 탐색 가능	
데이터 조회	<ul style="list-style-type: none"><li>특정 Document를 Table/JSON 형태 조회</li><li>Histogram 특정 구간 내의 데이터 조회</li><li>Histogram Bin 간격 설정</li><li>Histogram 데이터를 csv 출력</li><li>특정 Field의 정보만 조회</li><li>특정 Field 값을 기준으로 정렬</li></ul>	
데이터 통계	<ul style="list-style-type: none"><li>(선택한 Time Range 내의) Documents 개수 확인</li><li>특정 Field Value의 분포 확인 (주로 Categorical Data, 상위 500개)</li></ul>	

## Discover에서는 어떤 작업을 할 수 있을까?

주요 기능	세부 기능	
데이터 검색	(복잡한) 조건을 만족하는 데이터 선별적 탐색 가능	
데이터 검색 저장	검색한 결과를 저장하여 Visualize에서 사용	 “검색 및 필터” 학습 후
데이터 필터링	(간단한) 특정 조건을 만족하는 데이터 선별적 탐색 가능	
데이터 조회	<ul style="list-style-type: none"><li>특정 Document를 Table/JSON 형태 조회</li><li>Histogram 특정 구간 내의 데이터 조회</li><li>Histogram Bin 간격 설정</li><li>Histogram 데이터를 csv 출력</li><li>특정 Field의 정보만 조회</li><li>특정 Field 값을 기준으로 정렬</li></ul> 	 하나씩 보자
데이터 통계	<ul style="list-style-type: none"><li>(선택한 Time Range 내의) Documents 개수 확인</li><li>특정 Field Value의 분포 확인 (주로 Categorical Data, 상위 500개)</li></ul>	

## 데이터 조회 - 특정 Document를 Table/JSON 형태 조회

109 hits

New Save Open Share Auto-refresh < ⏪ ⏩ Month to date >

Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax

Add a filter +

Selected Fields: shopping

Available Fields: \_source, t\_id, t\_type, 고객ip, t\_연령대, t\_index, #\_score, t\_결제카드, #\_고객나이, t\_고객성별, t\_고객주소\_시도, t\_구매사이트, 물건좌표, t\_배송메모, #\_배송소요시간, #\_상품가격, #\_상품개수, t\_상품분류, ⏪ 수령시간

Time: August 1st 2018, 00:00:00.000 - August 12th 2018, 19:53:40.216 — Auto

Count: 주문시간 per 3 hours

**클릭** (highlighted in pink)

Time	_source
▶ August 12th 2018, 19:22:18.000	접수번호: 4,419 주문시간: August 12th 2018, 19:22:18.000 수령시간: August 16th 2018, 17:03:18.000 예약여부: 일반 배송메모: 부재중 고객ip: 190.7 1.170.58 고객성별: 여성 고객나이: 29 물건좌표: 35.21234732520931, 127.71795589785694 고객주소_시도: 서울특별시 구매사이트: 11번가 판매자평점: 2 상품분류: 셔츠 상품가격: 20,000 상품개수: 1 결제카드: 신한 _id: 8KomLGUBMEB1QWBFYbX8 _type: shopping _index: shopping _score: - 주문시간_시간 대: 19 연령대: 20~30대 주문시간_요일: SUNDAY 배송소요시간: 93
▶ August 12th 2018, 17:56:54.000	접수번호: 3,611 주문시간: August 12th 2018, 17:56:54.000 수령시간: August 14th 2018, 18:34:54.000 예약여부: 일반 배송메모: 부재중 고객ip: 1.20 2.26.66 고객성별: 여성 고객나이: 59 물건좌표: 35.25961656520813, 127.59964526333675 고객주소_시도: 세종특별자치시 구매사이트: GS샵 판매자평점: 1 상품분류: 스웨터 상품가격: 13,000 상품개수: 7 결제카드: 하나 _id: 35Xtm2QByNsCKuKnnwfU _type: shopping _index: shopping _score: - 주문시간_시간 대: 17 연령대: 40대 이상 주문시간_요일: SUNDAY 배송소요시간: 48
▶ August 12th 2018, 12:36:07.000	접수번호: 7,110 주문시간: August 12th 2018, 12:36:07.000 수령시간: August 14th 2018, 20:54:07.000 예약여부: 일반 배송메모: 부재중 고객ip: 125.1 79.199.93 고객성별: 남성 고객나이: 26 물건좌표: 36.23845867441393, 128.49313157159364 고객주소_시도: 경기도 구매사이트: g마켓 판매자평점: 1 상품분류: 수트 상품가격: 24,000 상품개수: 1 결제카드: 우리 _id: c6omLGUBMEB1QWBFW8DF _type: shopping _index: shopping _score: - 주문시간_시간 대: 12 연령대: 20~30대 주문시간_요일: SUNDAY 배송소요시간: 56
▶ August 12th 2018, 12:34:43.000	접수번호: 8,650 주문시간: August 12th 2018, 12:34:43.000 수령시간: August 14th 2018, 06:41:43.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 69.77.194.10 고객성별: 남성 고객나이: 47 물건좌표: 36.74282248611463, 127.17094688049092 고객주소_시도: 강원도 구매사이트: 옥션 판매자평점: 4 상품분류: 수트 상품가격: 25,000 상품개수: 1 결제카드: 신한 _id: d6omLGUBMEB1QWBF_cYW _type: shopping _index: shopping _score: - 주문시간_시간대: 12 연령대: 40대 이상 주문시간_요일: SUNDAY 배송소요시간: 42

## 데이터 조회 - 특정 Document를 Table 형태 조회

109 hits      New Save Open Share Auto-refresh < ⏪ ⏩ Month to date >

Search... (e.g. status:200 AND extension:PHP)      Uses lucene query syntax

Add a filter +

shopping

Selected Fields

? \_source

Available Fields

Popular

t \_id

t \_type

□ 고객ip

t 연령대

t \_index

# \_score

t 결제카드

# 고객나이

t 고객성별

t 고객주소\_시도

t 구매사이트

물건좌표

t 배송메모

# 배송소요시간

# 상품가격

# 상품개수

t 상품분류

수령시간

▲ 이전페이지 ▾ 다음페이지

August 1st 2018, 00:00:00.000 - August 12th 2018, 19:53:40.216 — Auto

Count

주문시간 per 3 hours

Time ▾

\_source

August 12th 2018, 19:22:18.000

접수번호: 4,419 주문시간: August 12th 2018, 19:22:18.000 수령시간: August 16th 2018, 17:03:18.000 예약여부: 일반 배송메모: 부재중 고객ip: 190.71.170.58 고객성별: 여성 고객나이: 29 물건좌표: 35.21234732520931, 127.71795589785694 고객주소\_시도: 서울특별시 구매사이트: 11번가 판매자평점: 2 상품분류: 셔츠 상품가격: 20,000 상품개수: 1 결제카드: 신한 \_id: 8KomLGUBMEB1QWBFYbX8 \_type: shopping \_index: shopping \_score: - 주문시간\_시간 대: 19 연령대: 20~30대 주문시간\_요일: SUNDAY 배송소요시간: 93

클릭

Table JSON

View surrounding documents View single document

t _id	8KomLGUBMEB1QWBFYbX8	
t _index	shopping	
# _score	-	
t _type	shopping	
t 결제카드	신한	
□ 고객ip	190.71.170.58	
# 고객나이	29	
t 고객성별	여성	
t 고객주소_시도	서울특별시	
t 구매사이트	11번가	
물건좌표	35.21234732520931, 127.71795589785694	

## 데이터 조회 - 특정 Document를 JSON 형태 조회

109 hits

New Save Open Share Auto-refresh < ⏪ ⏩ Month to date >

Search... (e.g. status:200 AND extension:PHP)

Uses lucene query syntax

Add a filter +

shopping

Selected Fields

? \_source

Available Fields

Popular

t \_id

t \_type

□ 고객ip

t 연령대

t \_index

# \_score

t 결제카드

# 고객나이

t 고객성별

t 고객주소\_시도

t 구매사이트

⌚ 물건좌표

t 배송메모

# 배송소요시간

# 상품가격

# 상품개수

t 상품분류

⌚ 수령시간

⌚ 일정선택

August 1st 2018, 00:00:00.000 - August 12th 2018, 19:53:40.216 — Auto

Count

주문시간 per 3 hours

Time ▾

\_source

August 12th 2018, 19:22:18.000

접수번호: 4,419 주문시간: August 12th 2018, 19:22:18.000 수령시간: August 16th 2018, 17:03:18.000 예약여부: 일반 배송메모: 부재중 고객ip: 190.71.170.58 고객성별: 여성 고객나이: 29 물건좌표: 35.21234732520931, 127.71795589785694 고객주소\_시도: 서울특별시 구매사이트: 11번가 판매자평점: 2 상품분류: 셔츠 상품가격: 20,000 상품개수: 1 결제카드: 신한 \_id: 8KomLGUBMEB1QWBFYbX8 \_type: shopping \_index: shopping \_score: - 주문시간\_시간 대: 19 연령대: 20~30대 주문시간\_요일: SUNDAY 배송소요시간: 93

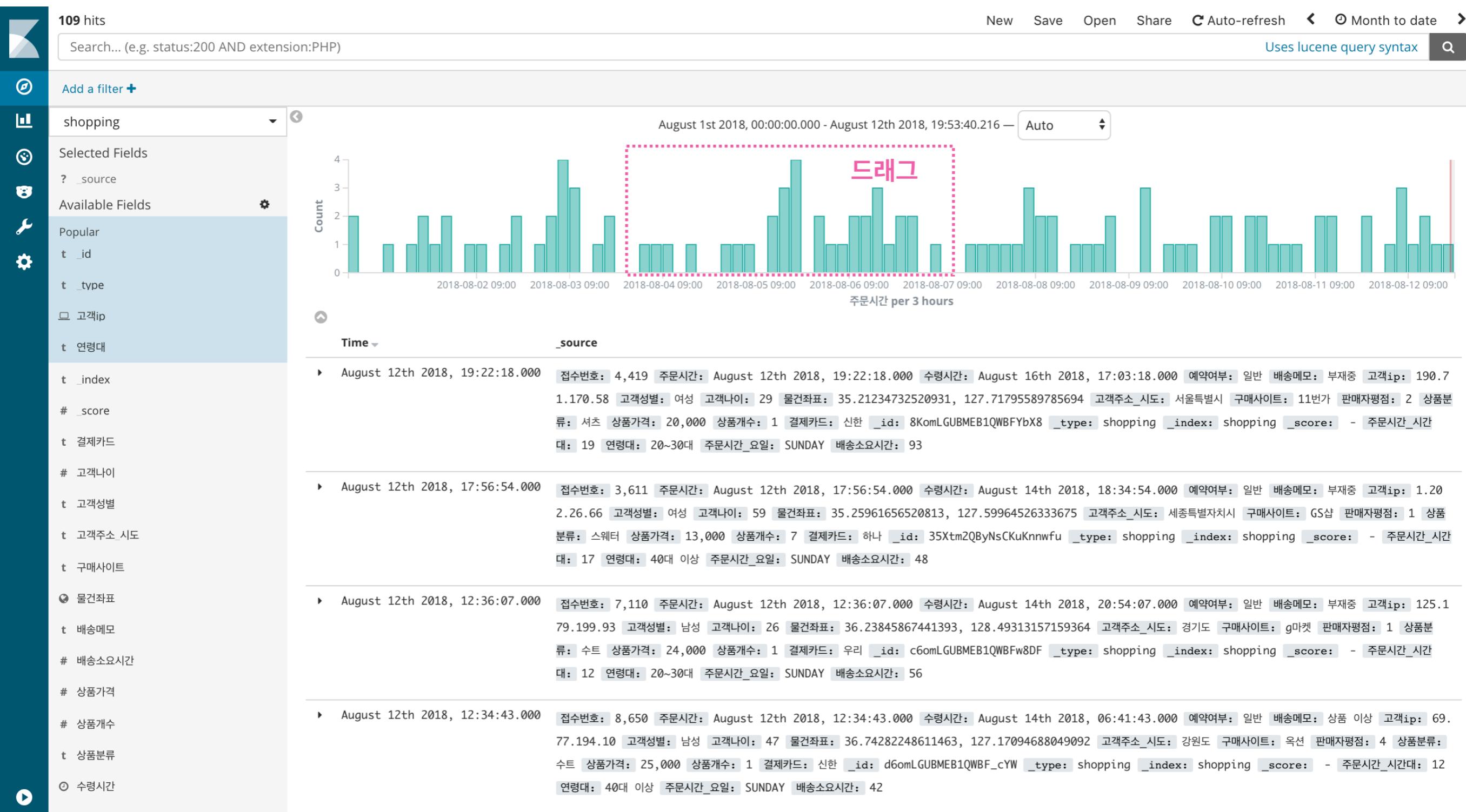
클릭

Table JSON

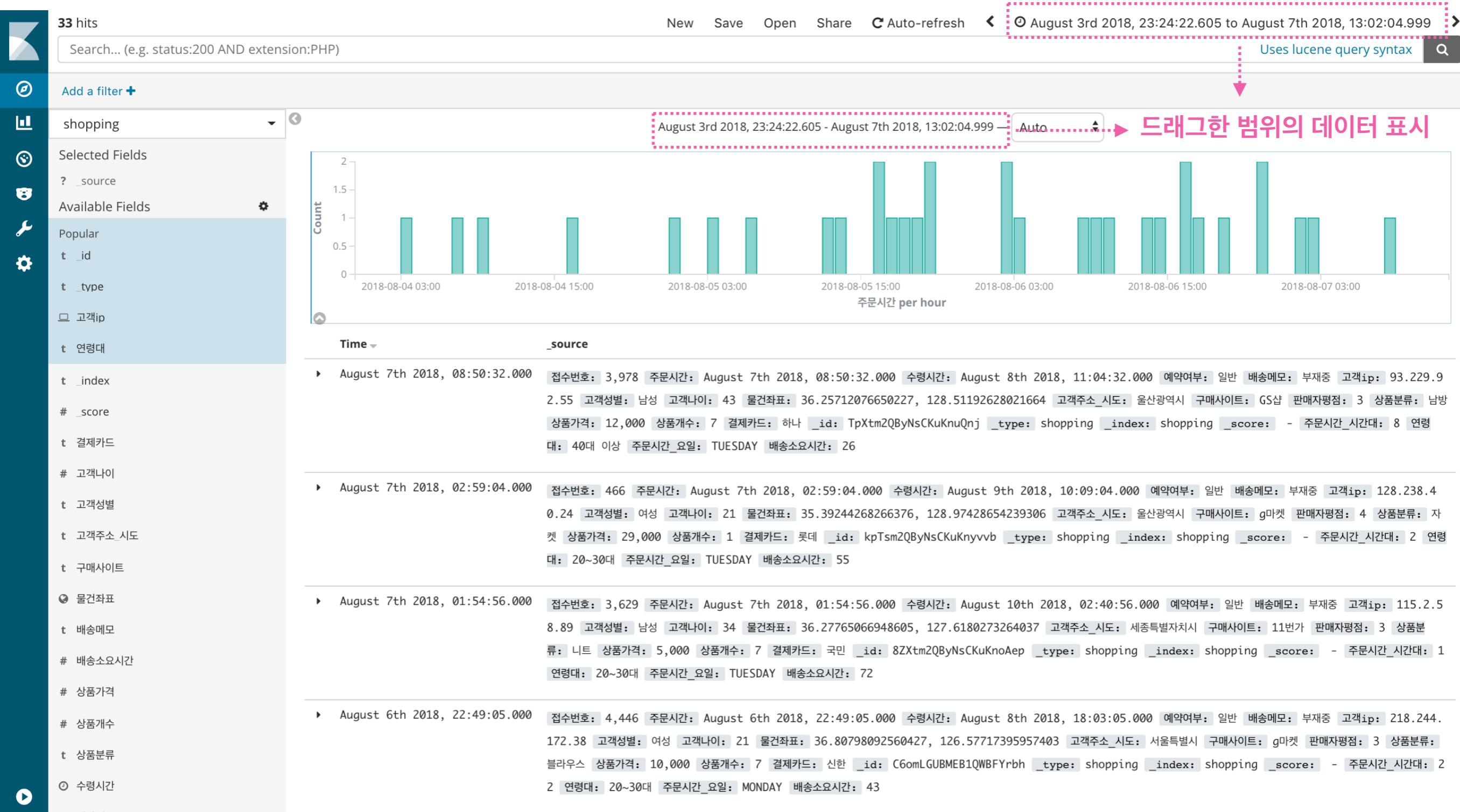
View surrounding documents View single document

```
1 ~ {  
2   "_index": "shopping",  
3   "_type": "shopping",  
4   "_id": "8KomLGUBMEB1QWBFYbX8",  
5   "_version": 1,  
6   "_score": null,  
7   "_source": {  
8     "접수번호": 4419,  
9     "주문시간": "2018-08-12T10:22:18",  
10    "수령시간": "2018-08-16T08:03:18",  
11    "예약여부": "일반",  
12    "배송메모": "부재중",  
13    "고객ip": "190.71.170.58",  
14    "고객성별": "여성",  
15    "고객나이": 29,  
16    "물건좌표": "35.21234732520931, 127.71795589785694",  
17    "고객주소_시도": "서울특별시",  
18    "구매사이트": "11번가",  
19    "판매자평점": 2,  
20    "상품분류": "셔츠"}
```

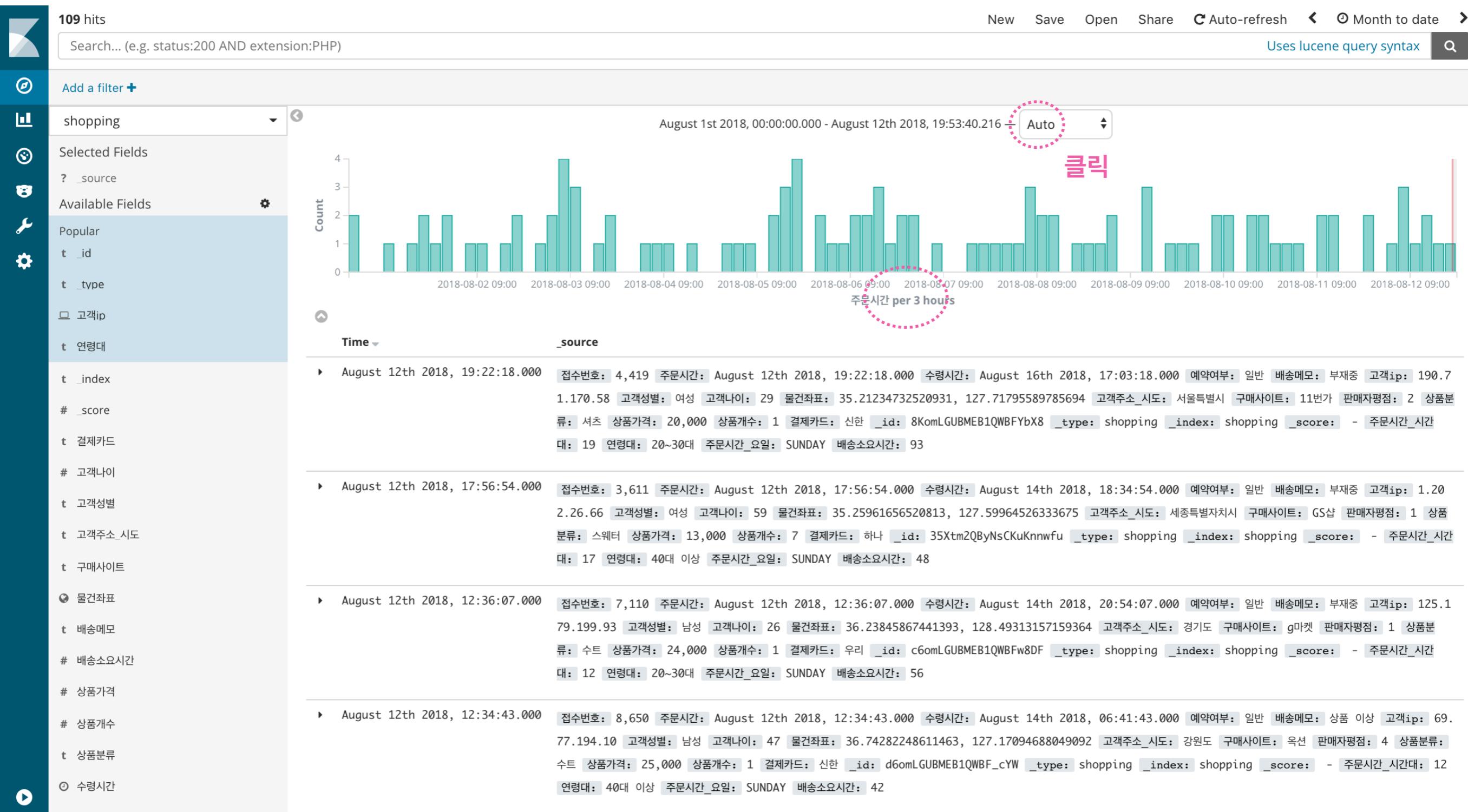
## 데이터 조회 - Histogram 특정 구간 내의 데이터 조회



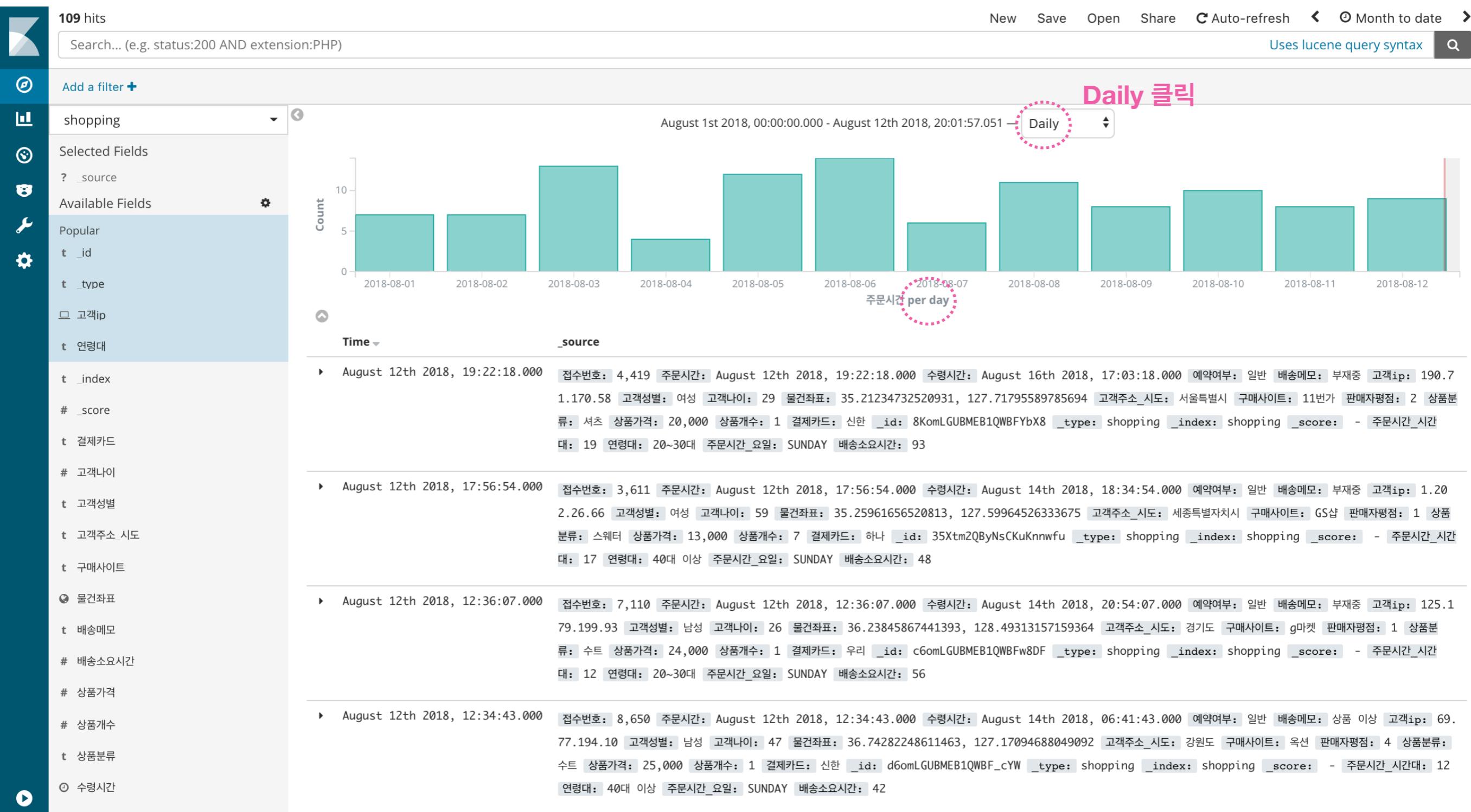
# 데이터 조회 - Histogram 특정 구간 내의 데이터 조회



## 데이터 조회 - Histogram 간격 설정



## 데이터 조회 - Histogram 간격 설정



# 데이터 조회 - Histogram 데이터를 csv 출력

109 hits

New Save Open Share Auto-refresh < ⏪ ⏩ Month to date >

Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax

Add a filter +

shopping

Selected Fields  
? \_source

Available Fields  
Popular  
t \_id  
t \_type  
고객ip  
t 연령대

Time ▾

Count

August 1st 2018, 00:00:00.000 - August 12th 2018, 19:53:40.216 — Auto

주문시간 per 3 hours

1. 클릭

\_source

▶ August 12th 2018, 19:22:18.000 접수번호: 4,419 주문시간: August 12th 2018, 19:22:18.000 수령시간: August 16th 2018, 17:03:18.000 예약여부: 일반 배송메모: 부재중 고객ip: 190.7 1.170.58 고객성별: 여성 고객나이: 29 물건좌표: 35.21234732520931, 127.71795589785694 고객주소\_시도: 서울특별시 구매사이트: 11번가 판매자평점: 2 상품분류: 셔츠 상품가격: 20,000 상품개수: 1 결제카드: 신한 \_id: 8KomLGUBMEB1QWBFYbX8 \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 19 연령대: 20~30대 주문시간\_요일: SUNDAY 배송소요시간: 93

▶ August 12th 2018, 17:56:54.000 접수번호: 3,611 주문시간: August 12th 2018, 17:56:54.000 수령시간: August 14th 2018, 18:34:54.000 예약여부: 일반 배송메모: 부재중 고객ip: 1.20 2.26.66 고객성별: 여성 고객나이: 59 물건좌표: 35.25961656520813, 127.59964526333675 고객주소\_시도: 세종특별자치시 구매사이트: GS샵 판매자평점: 1 상품분류: 스웨터 상품가격: 13,000 상품개수: 7 결제카드: 하나 \_id: 35Xtm2QByNsCKuKnnwfU \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 17 연령대: 40대 이상 주문시간\_요일: SUNDAY 배송소요시간: 48

▶ August 12th 2018, 12:36:07.000 접수번호: 7,110 주문시간: August 12th 2018, 12:36:07.000 수령시간: August 14th 2018, 20:54:07.000 예약여부: 일반 배송메모: 부재중 고객ip: 125.1 79.199.93 고객성별: 남성 고객나이: 26 물건좌표: 36.23845867441393, 128.49313157159364 고객주소\_시도: 경기도 구매사이트: g마켓 판매자평점: 1 상품분류: 수트 상품가격: 24,000 상품개수: 1 결제카드: 우리 \_id: c6omLGUBMEB1QWBFW8DF \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 12 연령대: 20~30대 주문시간\_요일: SUNDAY 배송소요시간: 56

▶ August 12th 2018, 12:34:43.000 접수번호: 8,650 주문시간: August 12th 2018, 12:34:43.000 수령시간: August 14th 2018, 06:41:43.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 69.77.194.10 고객성별: 남성 고객나이: 47 물건좌표: 36.74282248611463, 127.17094688049092 고객주소\_시도: 강원도 구매사이트: 옥션 판매자평점: 4 상품분류: 수트 상품가격: 25,000 상품개수: 1 결제카드: 신한 \_id: d6omLGUBMEB1QWBF\_cYW \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 12 연령대: 40대 이상 주문시간\_요일: SUNDAY 배송소요시간: 42

◀ 이전 다음 ▶

# 데이터 조회 - Histogram 데이터를 csv 출력

109 hits      New Save Open Share Auto-refresh < ⏪ ⏩ Month to date >

Search... (e.g. status:200 AND extension:PHP)      Uses lucene query syntax

Add a filter +

shopping

Selected Fields

? \_source

Available Fields

Popular

t \_id

t \_type

고객ip

t 연령대

t \_index

# \_score

t 결제카드

# 고객나이

t 고객성별

t 고객주소\_시도

t 구매사이트

물건좌표

t 배송메모

# 배송소요시간

# 상품가격

# 상품개수

t 상품분류

수령시간

August 1st 2018, 00:00:00.000 - August 12th 2018, 20:01:57.051 — Daily

Table Request Response Statistics

2018-08-09      8

2018-08-10      10

Export: Raw Formatted

2. 최하단까지 스크롤 다운

3. 클릭

Time ↓      \_source

August 12th 2018, 19:22:18.000      접수번호: 4,419 주문시간: August 12th 2018, 19:22:18.000 수령시간: August 16th 2018, 17:03:18.000 예약여부: 일반 배송메모: 부재중 고객ip: 190.7 1.170.58 고객성별: 여성 고객나이: 29 물건좌표: 35.21234732520931, 127.71795589785694 고객주소\_시도: 서울특별시 구매사이트: 11번가 판매자평점: 2 상품분류: 셔츠 상품가격: 20,000 상품개수: 1 결제카드: 신한 \_id: 8KomLGUBMEB1QWBFYbX8 \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 19 연령대: 20~30대 주문시간\_요일: SUNDAY 배송소요시간: 93

August 12th 2018, 17:56:54.000      접수번호: 3,611 주문시간: August 12th 2018, 17:56:54.000 수령시간: August 14th 2018, 18:34:54.000 예약여부: 일반 배송메모: 부재중 고객ip: 1.20 2.26.66 고객성별: 여성 고객나이: 59 물건좌표: 35.25961656520813, 127.59964526333675 고객주소\_시도: 세종특별자치시 구매사이트: GS샵 판매자평점: 1 상품분류: 스웨터 상품가격: 13,000 상품개수: 7 결제카드: 하나 \_id: 35Xtm2QByNsCKuKnnwfU \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 17 연령대: 40대 이상 주문시간\_요일: SUNDAY 배송소요시간: 48

August 12th 2018, 12:36:07.000      접수번호: 7,110 주문시간: August 12th 2018, 12:36:07.000 수령시간: August 14th 2018, 20:54:07.000 예약여부: 일반 배송메모: 부재중 고객ip: 125.1 79.199.93 고객성별: 남성 고객나이: 26 물건좌표: 36.23845867441393, 128.49313157159364 고객주소\_시도: 경기도 구매사이트: g마켓 판매자평점: 1 상품분류: 수트 상품가격: 24,000 상품개수: 1 결제카드: 우리 \_id: c6omLGUBMEB1QWBFw8DF \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 12 연령대: 20~30대 주문시간\_요일: SUNDAY 배송소요시간: 56

August 12th 2018, 12:34:43.000      접수번호: 8,650 주문시간: August 12th 2018, 12:34:43.000 수령시간: August 14th 2018, 06:41:43.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 69. 77.194.10 고객성별: 남성 고객나이: 47 물건좌표: 36.74282248611463, 127.17094688049092 고객주소\_시도: 강원도 구매사이트: 옥션 판매자평점: 4 상품분류: 수트 상품가격: 25,000 상품개수: 1 결제카드: 신한 \_id: d6omLGUBMEB1QWBF\_cYW \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 12 연령대: 40대 이상 주문시간\_요일: SUNDAY 배송소요시간: 42

Page Size 10

## 데이터 조회 - Histogram 데이터를 csv 출력

109 hits

New Save Open Share Auto-refresh < ⏪ ⏩ Month to date >

Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax

Add a filter +

shopping

Selected Fields

? \_source

Available Fields

Popular

t \_id

t \_type

고객ip

t 연령대

t \_index

# \_score

t 결제카드

# 고객나이

t 고객성별

t 고객주소\_시도

t 구매사이트

물건좌표

t 배송메모

# 배송소요시간

# 상품가격

# 상품개수

t 상품분류

August 1st 2018, 00:00:00.000 - August 12th 2018, 20:01:57.051 — Daily

Table Request Response Statistics

2018-08-09 8

2018-08-10 10

Export: Raw Formatted

1 2 »

Page Size 10

Time	_source
▶ August 12th 2018, 19:22:18.000	접수번호: 4,419 주문시간: August 12th 2018, 19:22:18.000 수령시간: August 16th 2018, 17:03:18.000 예약여부: 일반 배송메모: 부재중 고객ip: 190.7 1.170.58 고객성별: 여성 고객나이: 29 물건좌표: 35.21234732520931, 127.71795589785694 고객주소_시도: 서울특별시 구매사이트: 11번가 판매자평점: 2 상품분류: 셔츠 상품가격: 20,000 상품개수: 1 결제카드: 신한 _id: 8KomLGUBMEB1QWBFYbX8 _type: shopping _index: shopping _score: - 주문시간_시간 대: 19 연령대: 20~30대 주문시간_요일: SUNDAY 배송소요시간: 93
▶ August 12th 2018, 17:56:54.000	접수번호: 3,611 주문시간: August 12th 2018, 17:56:54.000 수령시간: August 14th 2018, 18:34:54.000 예약여부: 일반 배송메모: 부재중 고객ip: 1.20 2.26.66 고객성별: 여성 고객나이: 59 물건좌표: 35.25961656520813, 127.59964526333675 고객주소_시도: 세종특별자치시 구매사이트: GS샵 판매자평점: 1 상품분류: 스웨터 상품가격: 13,000 상품개수: 7 결제카드: 하나 _id: 35Xtm2QByNsCKuKnnwfU _type: shopping _index: shopping _score: - 주문시간_시간 대: 17 연령대: 40대 이상 주문시간_요일: SUNDAY 배송소요시간: 48
▶ August 12th 2018, 12:36:07.000	접수번호: 7,110 주문시간: August 12th 2018, 12:36:07.000 수령시간: August 14th 2018, 20:54:07.000 예약여부: 일반 배송메모: 부재중 고객ip: 125.1 79.199.93 고객성별: 남성 고객나이: 26 물건좌표: 36.23845867441393, 128.49313157159364 고객주소_시도: 경기도 구매사이트: g마켓 판매자평점: 1 상품분류: 수트 상품가격: 24,000 상품개수: 1 결제카드: 우리 _id: c6omLGUBMEB1QWBFW8DF _type: shopping _index: shopping _score: - 주문시간_시간 대: 12 연령대: 20~30대 주문시간_요일: SUNDAY 배송소요시간: 56
▶ August 12th 2018, 12:34:43.000	접수번호: 8,650 주문시간: August 12th 2018, 12:34:43.000 수령시간: August 14th 2018, 06:41:43.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 69. 77.194.10 고객성별: 남성 고객나이: 47 물건좌표: 36.74282248611463, 127.17094688049092 고객주소_시도: 강원도 구매사이트: 옥션 판매자평점: 4 상품분류: 스트 상품가격: 25,000 상품개수: 1 결제카드: 시하 _id: d6omLGLRMR10WRFcYW _type: shopping _index: shopping _score: - 주문시간_시간 대: 12 연령대: 20~30대 주문시간_요일: SUNDAY 배송소요시간: 56

New Saved Search.csv ... Show All X

다운로드 완료

## 데이터 조회 - 특정 Field의 정보만 조회

109 hits

New Save Open Share Auto-refresh < ⏪ ⏩ Month to date >

Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax

Add a filter +

Selected Fields

Available Fields ⚙️

Popular  
t \_id  
t \_type  
고객ip  
t 연령대  
t \_index  
# \_score  
t 결제카드  
# 고객나이  
t 고객성별  
t 고객주소\_시도  
t 구매사이트  
# 물건좌표  
t 배송메모  
# 배송소요시간  
# 상품가격  
# 상품개수  
t 상품분류  
# 수령시간

Count

August 1st 2018, 00:00:00.000 - August 12th 2018, 20:01:57.051 — Daily

주문시간 per day

Time ▾ source

▶ August 12th 2018, 19:22:18.000 접수번호: 4,419 주문시간: August 12th 2018, 19:22:18.000 수령시간: August 16th 2018, 17:03:18.000 예약여부: 일반 배송메모: 부재중 고객ip: 190.7 1.170.58 고객성별: 여성 고객나이: 29 물건좌표: 35.21234732520931, 127.71795589785694 고객주소\_시도: 서울특별시 구매사이트: 11번가 판매자평점: 2 상품분류: 셔츠 상품가격: 20,000 상품개수: 1 결제카드: 신한 \_id: 8KomLGUBMEB1QWBFYbX8 \_type: shopping \_index: shopping \_score: - 주문시간\_시간 대: 19 연령대: 20~30대 주문시간\_요일: SUNDAY 배송소요시간: 93

▶ August 12th 2018, 17:56:54.000 접수번호: 3,611 주문시간: August 12th 2018, 17:56:54.000 수령시간: August 14th 2018, 18:34:54.000 예약여부: 일반 배송메모: 부재중 고객ip: 1.20 2.26.66 고객성별: 여성 고객나이: 59 물건좌표: 35.25961656520813, 127.59964526333675 고객주소\_시도: 세종특별자치시 구매사이트: GS샵 판매자평점: 1 상품분류: 스웨터 상품가격: 13,000 상품개수: 7 결제카드: 하나 \_id: 35Xtm2QByNsCKuKnnwfU \_type: shopping \_index: shopping \_score: - 주문시간\_시간 대: 17 연령대: 40대 이상 주문시간\_요일: SUNDAY 배송소요시간: 48

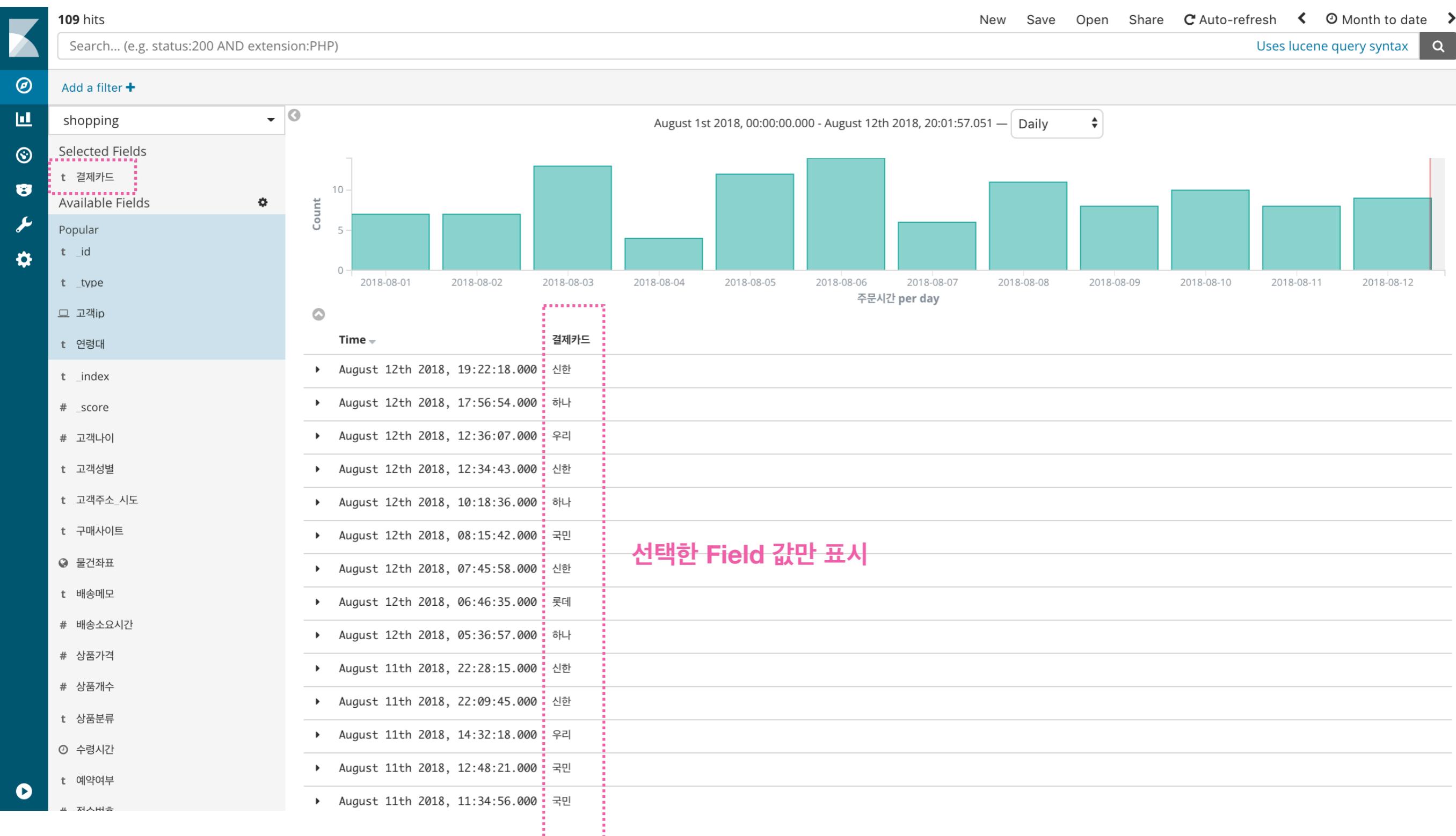
▶ August 12th 2018, 12:36:07.000 접수번호: 7,110 주문시간: August 12th 2018, 12:36:07.000 수령시간: August 14th 2018, 20:54:07.000 예약여부: 일반 배송메모: 부재중 고객ip: 125.1 79.199.93 고객성별: 남성 고객나이: 26 물건좌표: 36.23845867441393, 128.49313157159364 고객주소\_시도: 경기도 구매사이트: g마켓 판매자평점: 1 상품분류: 수트 상품가격: 24,000 상품개수: 1 결제카드: 우리 \_id: c6omLGUBMEB1QWBFw8DF \_type: shopping \_index: shopping \_score: - 주문시간\_시간 대: 12 연령대: 20~30대 주문시간\_요일: SUNDAY 배송소요시간: 56

▶ August 12th 2018, 12:34:43.000 접수번호: 8,650 주문시간: August 12th 2018, 12:34:43.000 수령시간: August 14th 2018, 06:41:43.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 69. 77.194.10 고객성별: 남성 고객나이: 47 물건좌표: 36.74282248611463, 127.17094688049092 고객주소\_시도: 강원도 구매사이트: 옥션 판매자평점: 4 상품분류: 수트 상품가격: 25,000 상품개수: 1 결제카드: 신한 \_id: d6omLGUBMEB1QWBF\_cYW \_type: shopping \_index: shopping \_score: - 주문시간\_시간 대: 12 연령대: 40대 이상 주문시간\_요일: SUNDAY 배송소요시간: 42

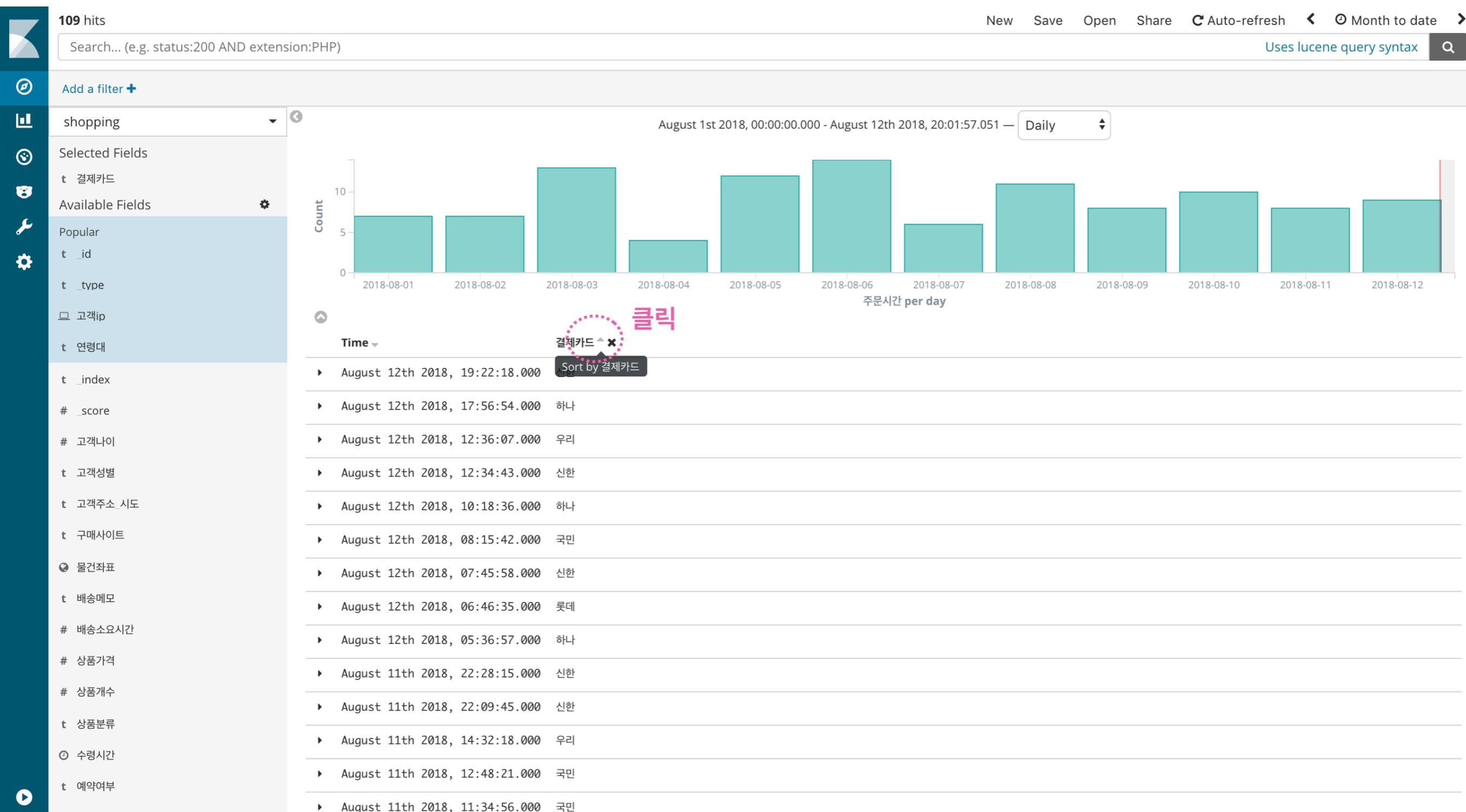
클릭

add

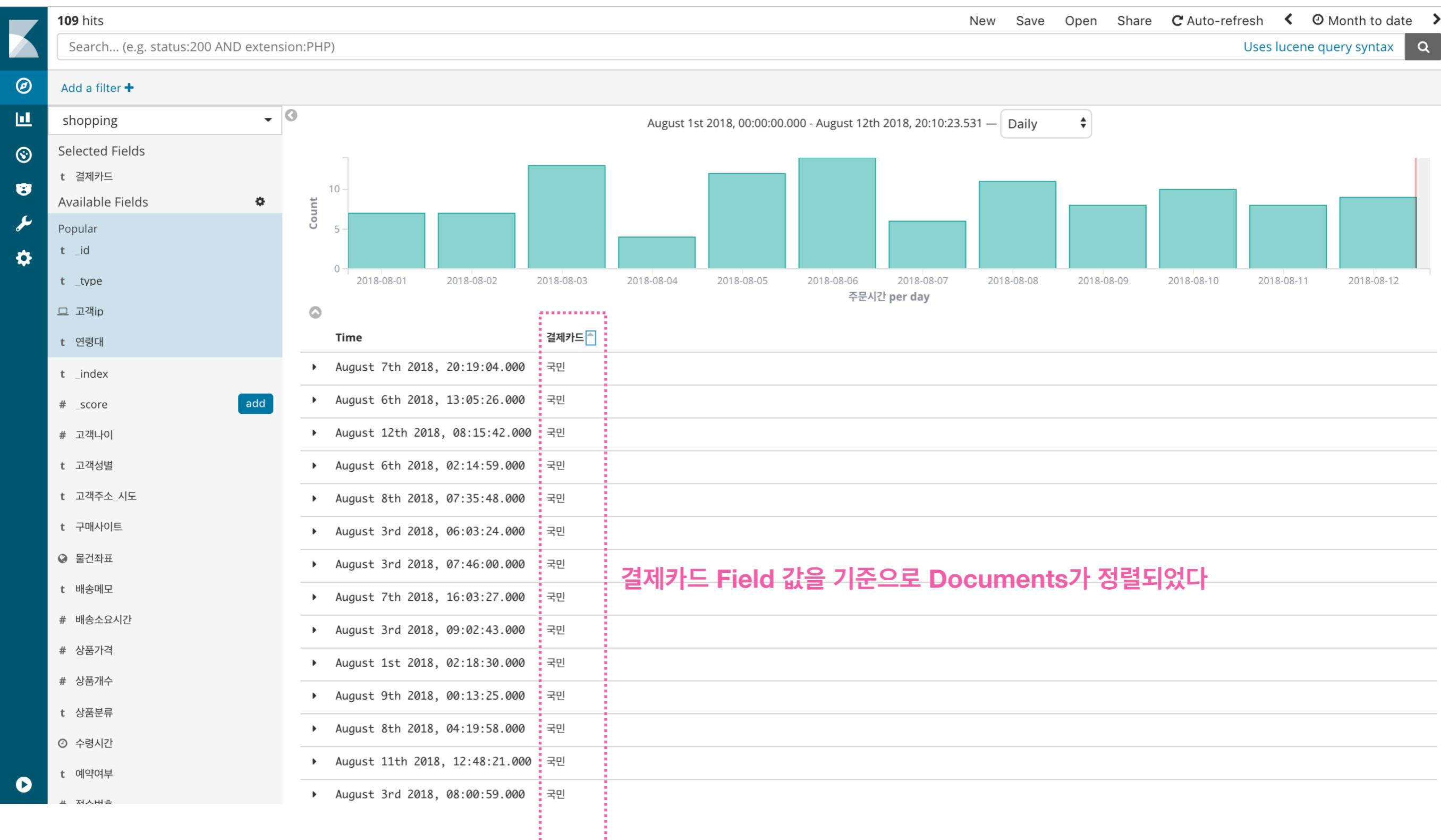
## 데이터 조회 - 특정 Field의 정보만 조회



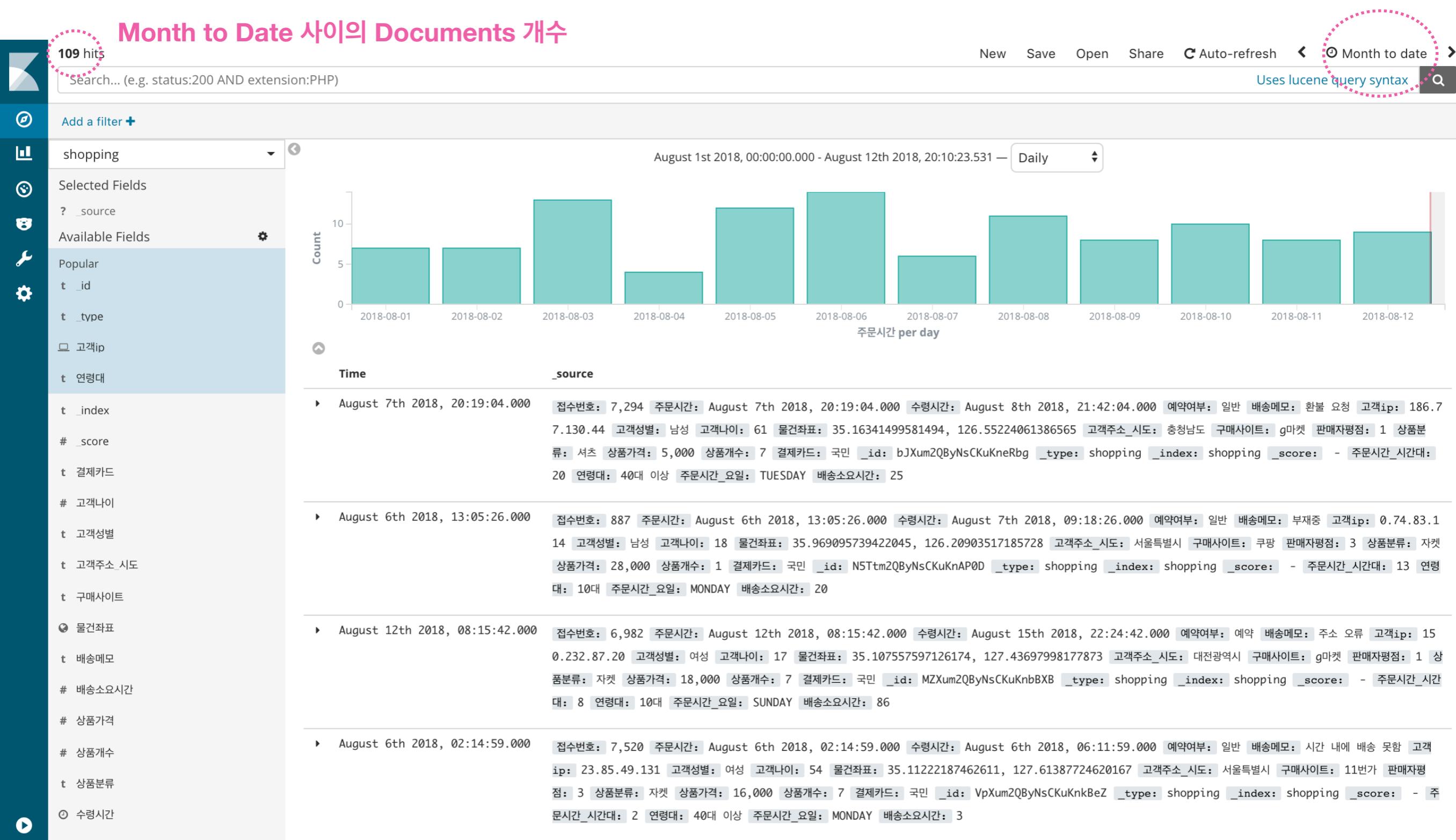
## 데이터 조회 - 특정 Field 값을 기준으로 정렬



## 데이터 조회 - 특정 Field 값을 기준으로 정렬



## 데이터 통계 - (선택한 Time Range 내의) Documents 개수 확인



## 데이터 통계 - 특정 Field Value의 분포 확인 (상위 500개)

109 hits

New Save Open Share Auto-refresh < ⏪ ⏩ Month to date >

Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax

Add a filter +

shopping

Selected Fields

- ? \_source
- Available Fields
  - Popular
  - t \_id
  - t \_type
  - 고객ip
  - t 연령대
  - t \_index
  - # \_score
  - t 결제카드
  - # 고객나이
  - t 고객성별
  - t 고객주소\_시도
  - t 구매사이트
  - 물건좌표
  - t 배송메모
  - # 배송소요시간
  - # 상품가격
  - # 상품개수
  - t 상품분류
  - 수령시간

Count

August 1st 2018, 00:00:00.000 - August 12th 2018, 20:10:23.531 — Daily

Time	_source
August 7th 2018, 20:19:04.000	접수번호: 7,294 주문시간: August 7th 2018, 20:19:04.000 수령시간: August 8th 2018, 21:42:04.000 예약여부: 일반 배송메모: 환불 요청 고객ip: 186.7 7.130.44 고객성별: 남성 고객나이: 61 물건좌표: 35.16341499581494, 126.55224061386565 고객주소_시도: 충청남도 구매사이트: g마켓 판매자평점: 1 상품분류: 셔츠 상품가격: 5,000 상품개수: 7 결제카드: 국민 _id: bJXum2QByNsCKuKneRbg _type: shopping _index: shopping _score: - 주문시간_시간대: 20 연령대: 40대 이상 주문시간_요일: TUESDAY 배송소요시간: 25
August 6th 2018, 13:05:26.000	접수번호: 887 주문시간: August 6th 2018, 13:05:26.000 수령시간: August 7th 2018, 09:18:26.000 예약여부: 일반 배송메모: 부재중 고객ip: 0.74.83.1 14 고객성별: 남성 고객나이: 18 물건좌표: 35.969095739422045, 126.20903517185728 고객주소_시도: 서울특별시 구매사이트: 쿠팡 판매자평점: 3 상품분류: 자켓 상품가격: 28,000 상품개수: 1 결제카드: 국민 _id: N5Ttm2QByNsCKuKnAP0D _type: shopping _index: shopping _score: - 주문시간_시간대: 13 연령대: 10대 주문시간_요일: MONDAY 배송소요시간: 20
August 12th 2018, 08:15:42.000	접수번호: 6,982 주문시간: August 12th 2018, 08:15:42.000 수령시간: August 15th 2018, 22:24:42.000 예약여부: 예약 배송메모: 주소 오류 고객ip: 15 0.232.87.20 고객성별: 여성 고객나이: 17 물건좌표: 35.107557597126174, 127.43697998177873 고객주소_시도: 대전광역시 구매사이트: g마켓 판매자평점: 1 상품분류: 자켓 상품가격: 18,000 상품개수: 7 결제카드: 국민 _id: MZXum2QByNsCKuKnbBXB _type: shopping _index: shopping _score: - 주문시간_시간대: 8 연령대: 10대 주문시간_요일: SUNDAY 배송소요시간: 86
August 6th 2018, 02:14:59.000	접수번호: 7,520 주문시간: August 6th 2018, 02:14:59.000 수령시간: August 6th 2018, 06:11:59.000 예약여부: 일반 배송메모: 시간 내에 배송 못함 고객ip: 23.85.49.131 고객성별: 여성 고객나이: 54 물건좌표: 35.11222187462611, 127.61387724620167 고객주소_시도: 서울특별시 구매사이트: 11번가 판매자평점: 3 상품분류: 자켓 상품가격: 16,000 상품개수: 7 결제카드: 국민 _id: VpXum2QByNsCKuKnkBeZ _type: shopping _index: shopping _score: - 주문시간_시간대: 2 연령대: 40대 이상 주문시간_요일: MONDAY 배송소요시간: 3

**클릭** add

## 데이터 통계 - 특정 Field Value의 분포 확인 (상위 500개)

109 hits

New Save Open Share Auto-refresh Month to date Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax

Add a filter +

Selected Fields

Available Fields

Popular

t\_id

t\_type

고객ip

t\_연령대

t\_index

#\_score

t\_결제카드

Top 5 values in 109 / 109 records

국민	34.9%
우리	19.3%
하나	18.3%
신한	14.7%
롯데	8.3%

선택한 Field 값의 분포를 보여준다

Visualize

Count August 1st 2018, 00:00:00.000 - August 12th 2018, 20:10:23.531 — Daily

Time \_source

August 7th 2018, 20:19:04.000 접수번호: 7,294 주문시간: August 7th 2018, 20:19:04.000 수령시간: August 8th 2018, 21:42:04.000 예약여부: 일반 배송메모: 환불 요청 고객ip: 186.7 7.130.44 고객성별: 남성 고객나이: 61 물건좌표: 35.16341499581494, 126.55224061386565 고객주소\_시도: 충청남도 구매사이트: g마켓 판매자평점: 1 상품분류: 셔츠 상품가격: 5,000 상품개수: 7 결제카드: 국민 \_id: bJXum2QByNsCKuKneRbg \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 20 연령대: 40대 이상 주문시간\_요일: TUESDAY 배송소요시간: 25

August 6th 2018, 13:05:26.000 접수번호: 887 주문시간: August 6th 2018, 13:05:26.000 수령시간: August 7th 2018, 09:18:26.000 예약여부: 일반 배송메모: 부재중 고객ip: 0.74.83.1 14 고객성별: 남성 고객나이: 18 물건좌표: 35.969095739422045, 126.20903517185728 고객주소\_시도: 서울특별시 구매사이트: 쿠팡 판매자평점: 3 상품분류: 자켓 상품가격: 28,000 상품개수: 1 결제카드: 국민 \_id: N5Ttm2QByNsCKuKnAP0D \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 13 연령대: 10대 주문시간\_요일: MONDAY 배송소요시간: 20

August 12th 2018, 08:15:42.000 접수번호: 6,982 주문시간: August 12th 2018, 08:15:42.000 수령시간: August 15th 2018, 22:24:42.000 예약여부: 예약 배송메모: 주소 오류 고객ip: 15 0.232.87.20 고객성별: 여성 고객나이: 17 물건좌표: 35.107557597126174, 127.43697998177873 고객주소\_시도: 대전광역시 구매사이트: g마켓 판매자평점: 1 상품분류: 자켓 상품가격: 18,000 상품개수: 7 결제카드: 국민 \_id: MZXum2QByNsCKuKnBxB \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 8 연령대: 10대 주문시간\_요일: SUNDAY 배송소요시간: 86

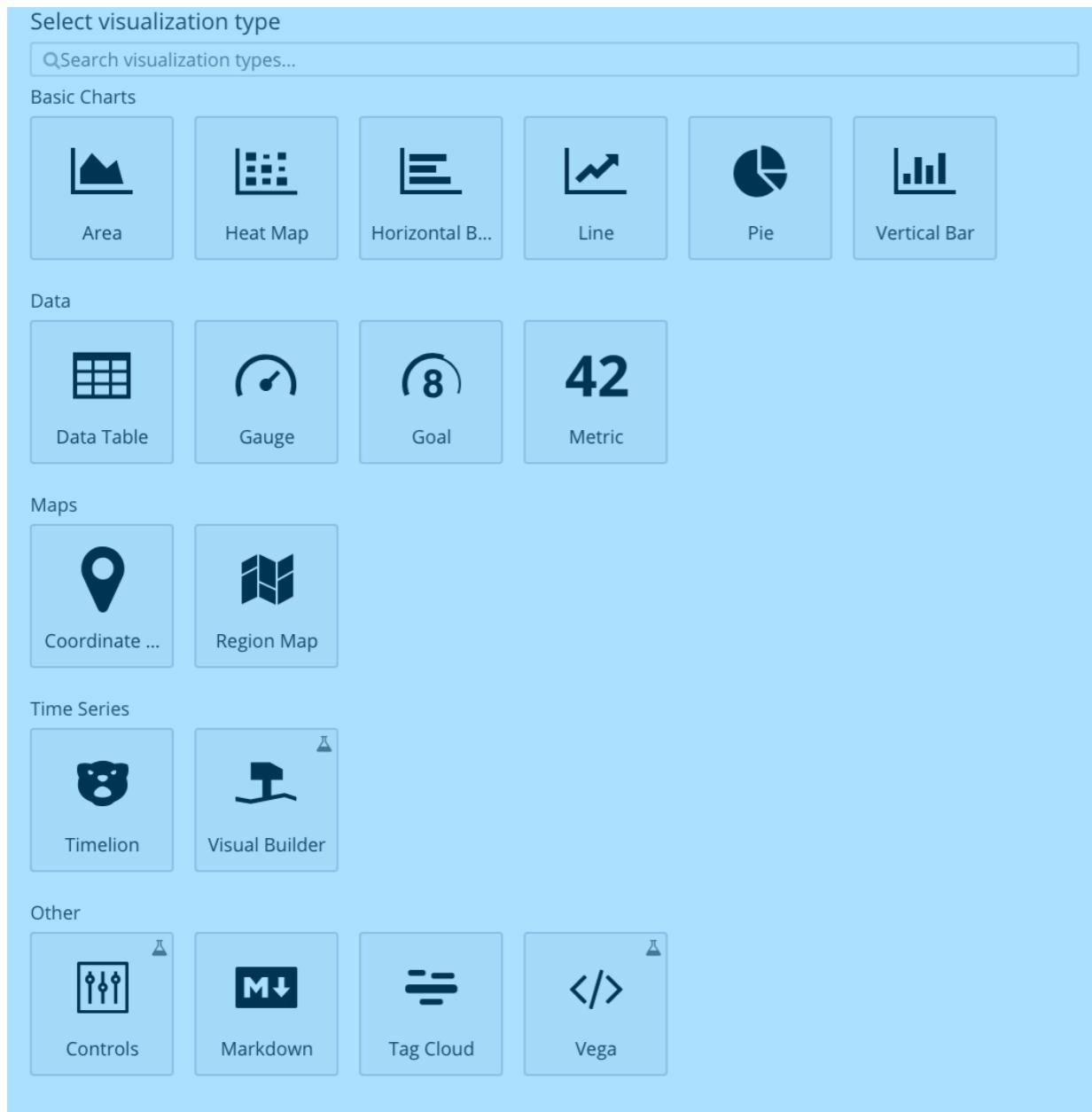
August 6th 2018, 02:14:59.000 접수번호: 7,520 주문시간: August 6th 2018, 02:14:59.000 수령시간: August 6th 2018, 06:11:59.000 예약여부: 일반 배송메모: 시간 내에 배송 못함 고객ip: 23.85.49.131 고객성별: 여성 고객나이: 54 물건좌표: 35.11222187462611, 127.61387724620167 고객주소\_시도: 서울특별시 구매사이트: 11번가 판매자평점: 3 상품분류: 자켓 상품가격: 16,000 상품개수: 7 결제카드: 국민 \_id: VpXum2QByNsCKuKnkBeZ \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 2 연령대: 40대 이상 주문시간\_요일: MONDAY 배송소요시간: 3

상위 500개를 어떻게 sort 할 지에 따라서 결과가 바뀌는 것 주의!

데이터 시각화

## 어떤 시각화를 할 수 있을까?

공식



비공식

network

cohort

dendrogram

:

Kibana Visualize는 어렵나?



그럴 수 있다. 그렇다면 왜?



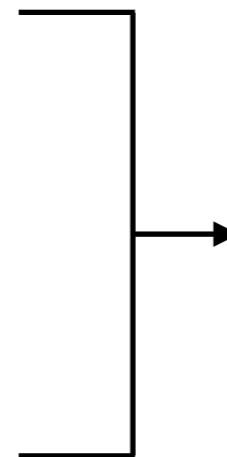
용어가 너무 낯설어서



눈 딱 감고 맛보기로 1개만 따라해보자

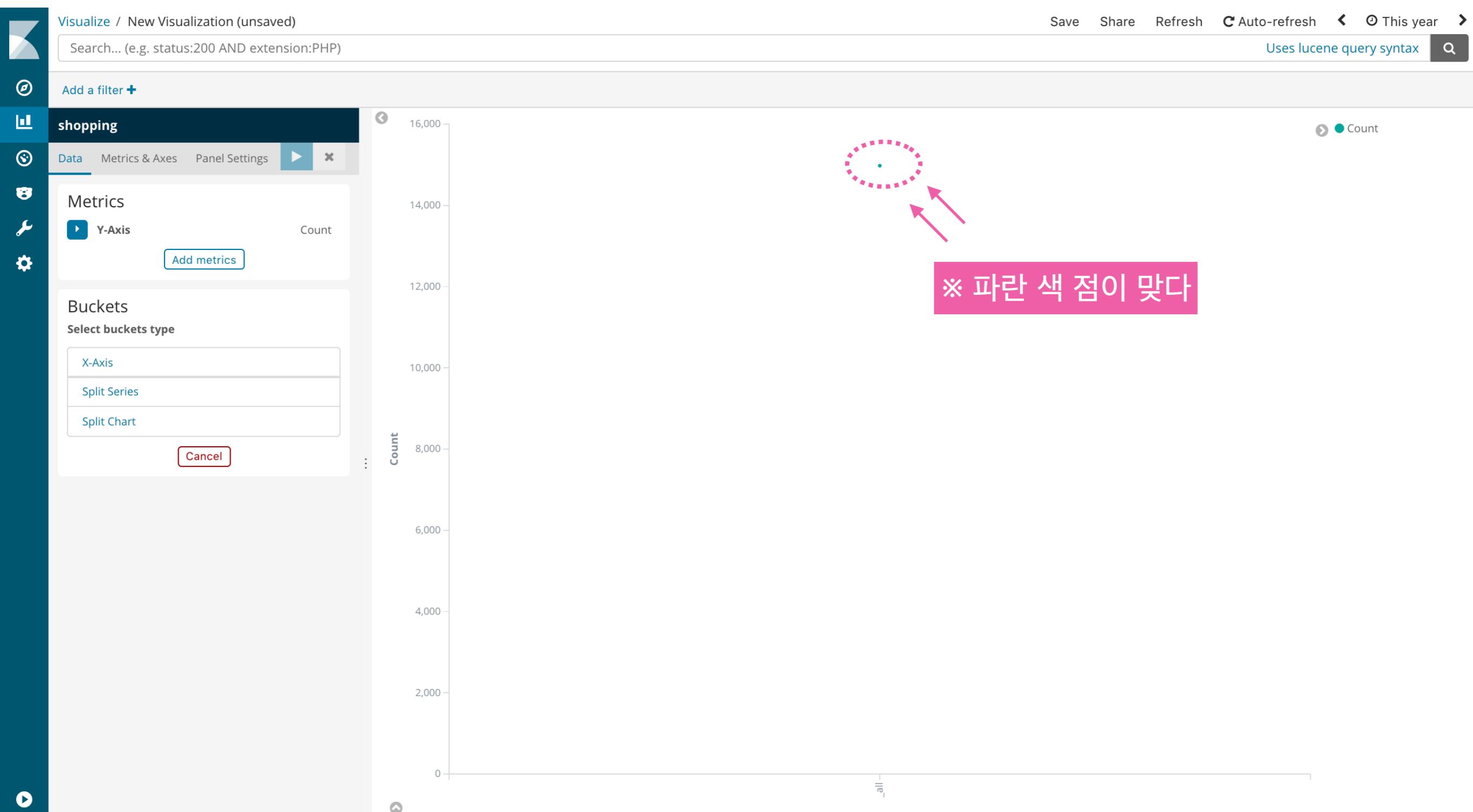
## Visualize 과정

- Kibana 접속
- Visualize 선택
- Create new visualization 선택
- Select visualization type 선택 - 예) Line Chart
- From a New Search, Select Index - 예) shopping



이 과정을 잘 기억하자

## 다음과 같은 화면이 나온다



## 아직 아무 것도 안한거처럼 보이지만 이미 하나의 Visualization을 생성했다

- shopping index에 있는 데이터 중에서
- *This year* 기간에 해당하는 데이터만 선별해서
- documents의 개수를 count 한 후
- y축에 표시해라



이번 페이지의 목적은 “익숙해지기” 이니 **metrics**와 **buckets** 등을 이것저것 클릭해보자.  
그리고 어떤 내용을 알아야 Visualize를 자유자재로 사용할 수 있을지 정도만 생각해보자

## Aggregation!?

shopping

Data Metrics & Axes Panel Settings ⏪ ✕

Metrics

Y-Axis

Aggregation

Count

Custom Label

Advanced

Add metrics

This screenshot shows the 'Data' panel of a Kibana visualization titled 'shopping'. It includes tabs for 'Data', 'Metrics & Axes', and 'Panel Settings'. Under 'Metrics', there's a dropdown for 'Y-Axis' set to 'Count'. Below it is a 'Custom Label' input field and an 'Add metrics' button. A pink box highlights the 'Count' dropdown.

shopping

Data Metrics & Axes Panel Settings ⏪ ✕

Metrics

Y-Axis Count

Add metrics

Buckets

X-Axis

Aggregation

Select an aggregation

Add sub-buckets

This screenshot shows the 'Data' panel with 'Metrics & Axes' selected. It has sections for 'Metrics' (Y-axis set to 'Count') and 'Buckets' (X-axis set to '주문시간 per week'). Under 'Metrics', there's an 'Add metrics' button. Under 'Buckets', there's an 'Add sub-buckets' button. A pink box highlights the 'Select an aggregation' dropdown in the 'Aggregation' section.

shopping

Data Metrics & Axes Panel Settings ⏪ ✕

Metrics

Y-Axis Count

Add metrics

Buckets

X-Axis 주문시간 per week

Split Series

Sub Aggregation

Select an aggregation

Add sub-buckets

This screenshot shows the 'Data' panel with 'Metrics & Axes' selected. It includes sections for 'Metrics' (Y-axis set to 'Count'), 'Buckets' (X-axis set to '주문시간 per week'), and 'Sub Aggregation'. The 'Sub Aggregation' section contains a 'Select an aggregation' dropdown and an 'Add sub-buckets' button. A pink box highlights the 'Select an aggregation' dropdown in the 'Sub Aggregation' section.

Kibana 시각화를 제대로 하려면 **Aggregation**을 이해해야 한다!

## 시각화 = Aggregation

전국 학생들의 지역별 평균 키를 막대 그래프로 시각화 있다고 하자



Kibana Frame

①      ②      ③

전국 학생들의 지역별 평균 키를 막대 그래프로 시각화 있다고 하자

- ① Bucket Aggregation (Terms)
- ② Metric Aggregation (Average)
- ③ Visualization Type (Vertical Bar)

**Aggregation - Bucket**

## Bucket = Group

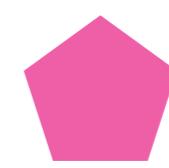
예를 들어 다음과 같은 도형이 있다고 하자



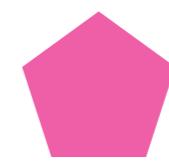
위의 도형을 여러 그룹으로 나눠야 한다면 어떻게 할 수 있을까?

어떤 방법을 택하든 가장 먼저 하는 작업은 기준을 정하는 것이다

내각의 합



색



**Bucket Aggregation** = 데이터를 일정한 기준으로 나누어 여러 Bucket으로 나누는 Aggregation

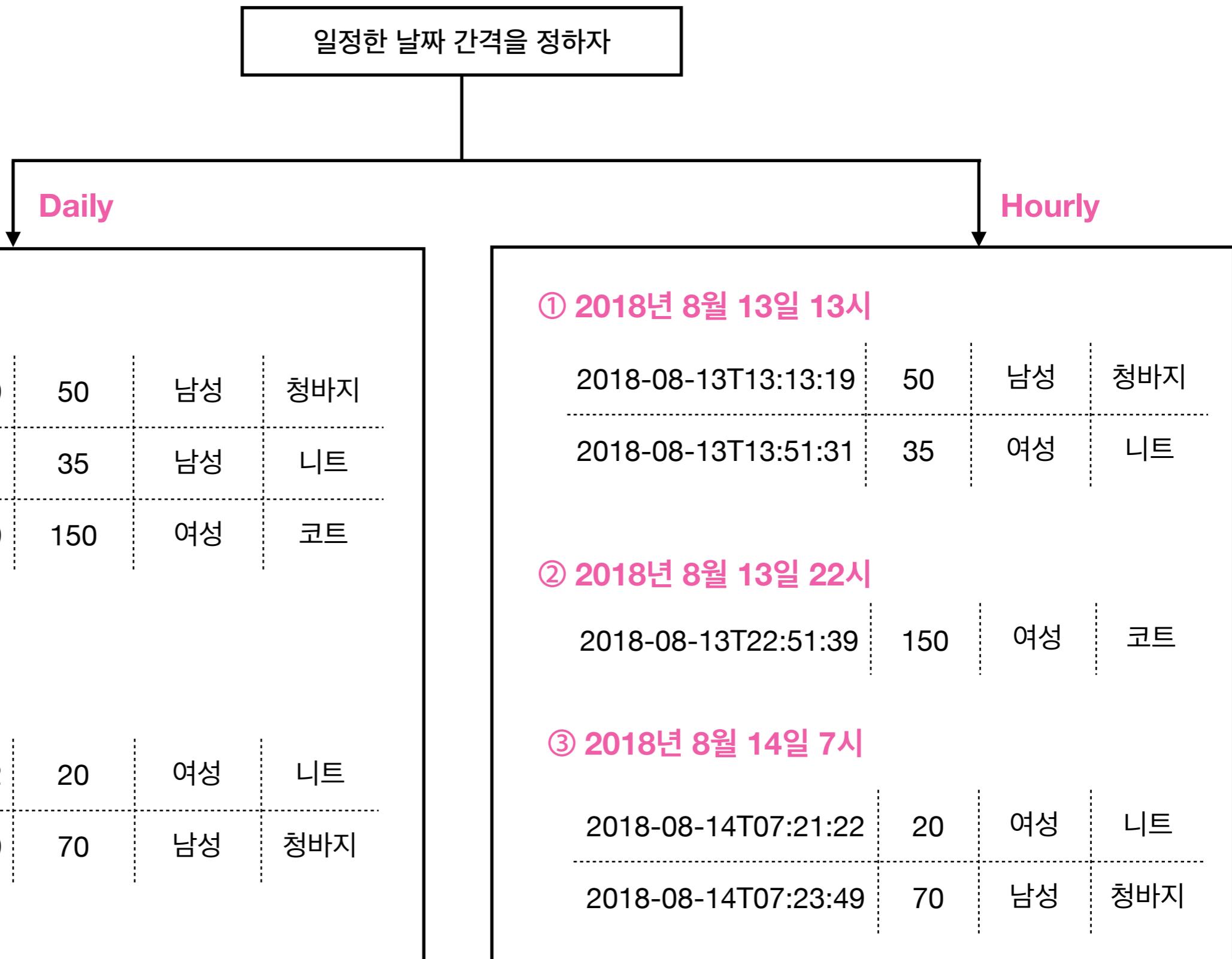
## Kibana가 지원하는 Bucket Aggregation

종류	적용 가능 Type	기준	예시
Date Histogram	Date	일정한 간격의 날짜/시간	월별, 주별, 일별, 시간별
Date Range	Date	일정하지 않은 간격의 날짜/시간	작년, 최근 석 달, 저번 주, 오늘
Histogram	Number	일정한 간격의 값	100~200, 200~300, 300~400
Range	Number	일정하지 않은 간격의 값	10~50, 150~200, 500~100
Terms	All	(카테고리 Field) 값	남성/여성, 서울/경기도/강원도
Significant Terms	String	(전체 대비) Foreground에서 interesting/unusual한 값	전체 대비 “서울”에서 interesting/unusual 한 상품분류
Filters	All	직접 입력	서울, 20대, 쿠팡
Geo Hash	Geo Point	geo point 간의 거리	거리가 가까운 상점
IPv4 Range	IP	IP 주소의 범위	0.0.0.0 ~ 127.255.255.255

아래와 같은 데이터는 어떤 기준으로 Bucket을 생성할 수 있을까?

시간	가격	성별	분류
2018-08-13T13:13:19	50	남성	청바지
2018-08-13T13:51:31	35	남성	니트
2018-08-13T22:51:39	150	여성	코트
2018-08-14T07:21:22	20	여성	니트
2018-08-14T07:23:49	70	남성	청바지

## Date Histogram Aggregation으로 Bucket을 생성하자



## Terms Aggregation으로 Bucket을 생성하자

(Categorical) Field를 정하자

성별

분류

① 성별 = 남성

2018-08-13T13:13:19	50	남성	청바지
2018-08-13T13:51:31	35	남성	니트
2018-08-14T07:23:49	70	남성	청바지

② 성별 = 여성

2018-08-14T07:21:22	20	여성	니트
2018-08-13T22:51:39	150	여성	코트

① 분류 = 청바지

2018-08-13T13:13:19	50	남성	청바지
2018-08-14T07:23:49	70	남성	청바지

② 분류 = 코트

2018-08-13T22:51:39	150	여성	코트
---------------------	-----	----	----

③ 분류 = 니트

2018-08-14T07:21:22	20	여성	니트
2018-08-13T13:51:31	35	여성	니트

## Significant Terms Aggregation은 언제 사용할까?

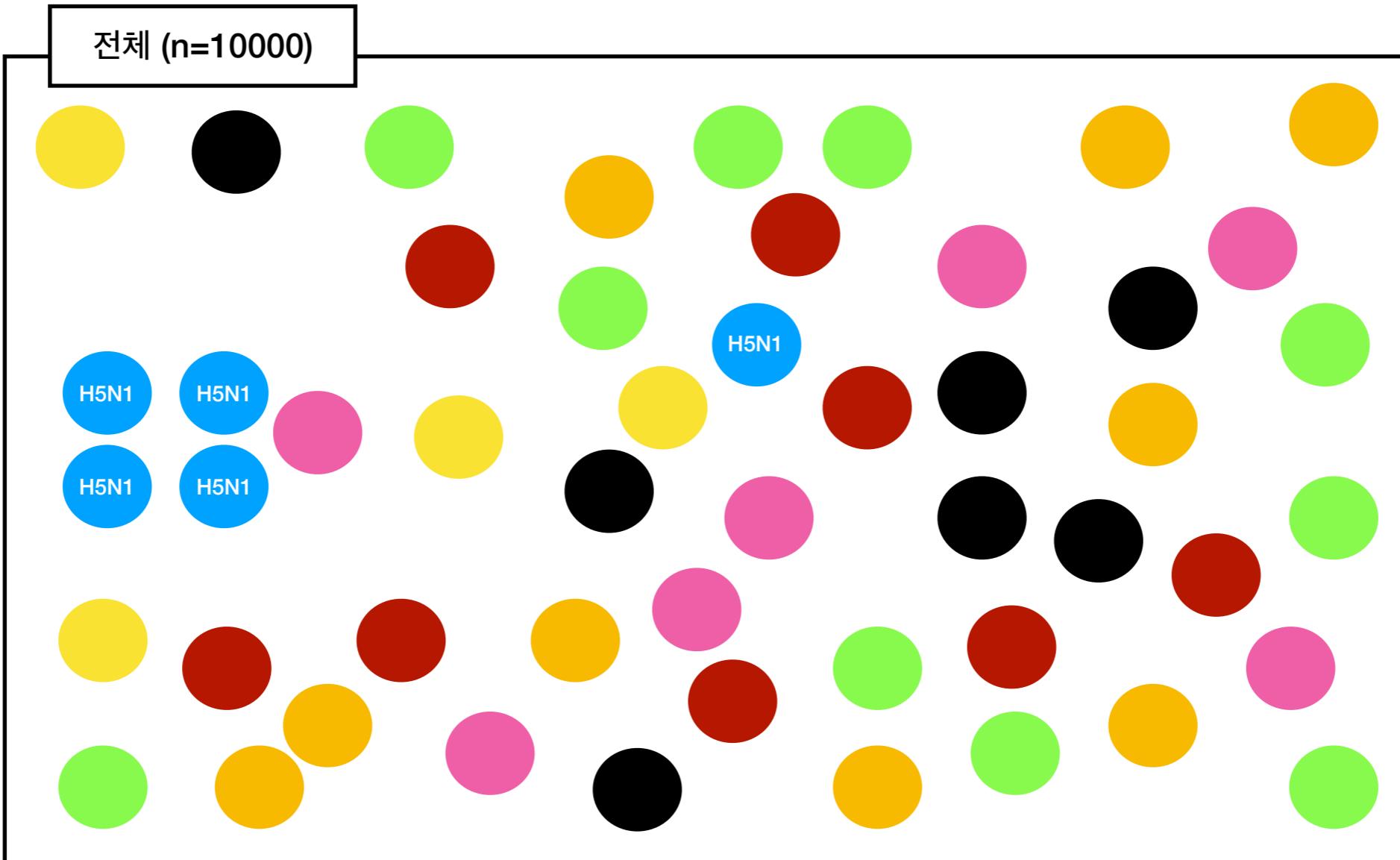
- Significant Terms Aggregation도 Terms Aggregation의 일종이다
- 다만 직접 기준을 설정하는 Terms Aggregation과 달리 Significant Terms Aggregation은 모호해 보인다
- 공식 문서에도 다음과 같이 나와 있다.

*In all these cases the terms being selected are not simply the most popular terms in a set.  
They are the terms that have undergone a significant change in popularity measured between a foreground and background set.*

- 위의 3가지 포인트를 중심으로 다음의 예시를 살펴보자

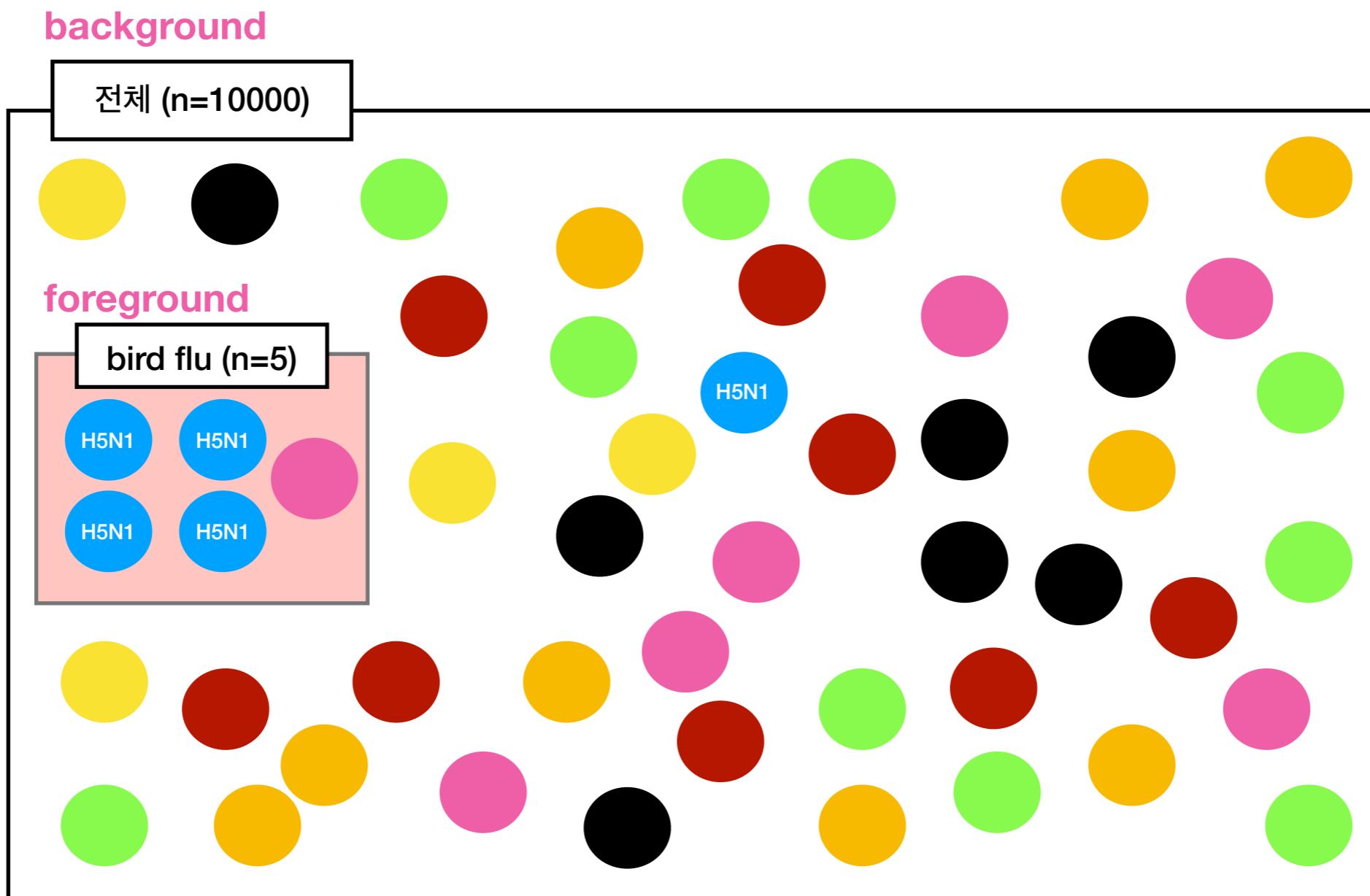
## Significant Terms Aggregation은 언제 사용할까? 🤔

background



## Significant Terms Aggregation은 언제 사용할까?

bird flu를 검색했을 때 H5N1를 return



- 전체에서 H5N1 비중 :  $1/10000 \rightarrow 0.0001\%$
- bird ful에서 H5N1 비중 :  $4/5 \rightarrow 80\%$

significant change

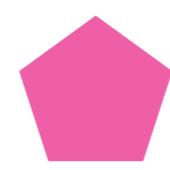
## Aggregation - Metrics

## Metrics = 수치화

Bucket은 일종의 Grouping 작업이라고 했다.

다만 Grouping만으로는 유의미한 결과를 얻을 수 없다

색



위의 결과를 해석해보면 “**파란색, 녹색, 자주색**” 이다. 유의미한가?

	파란색	녹색	자주색
개수	2	1	1

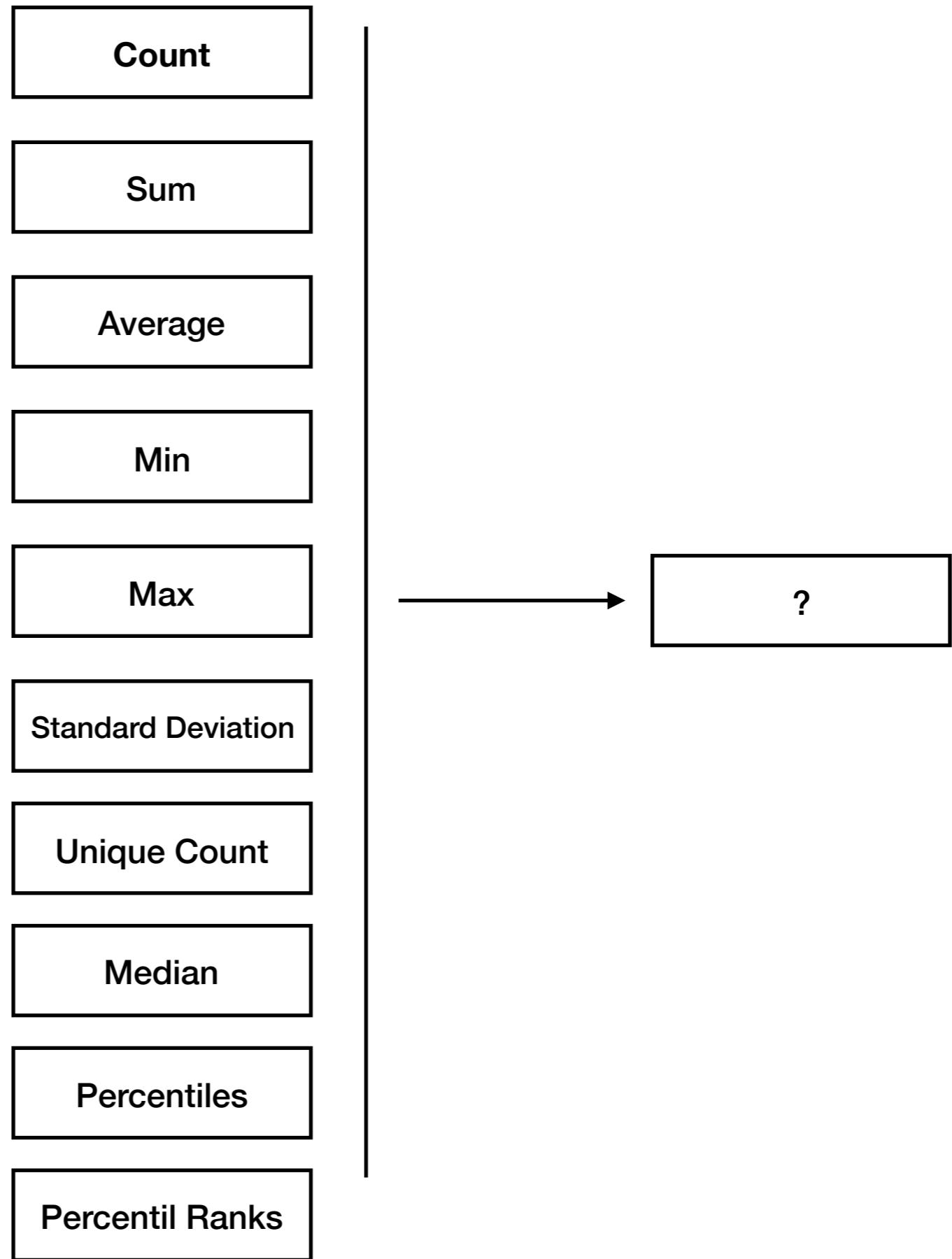
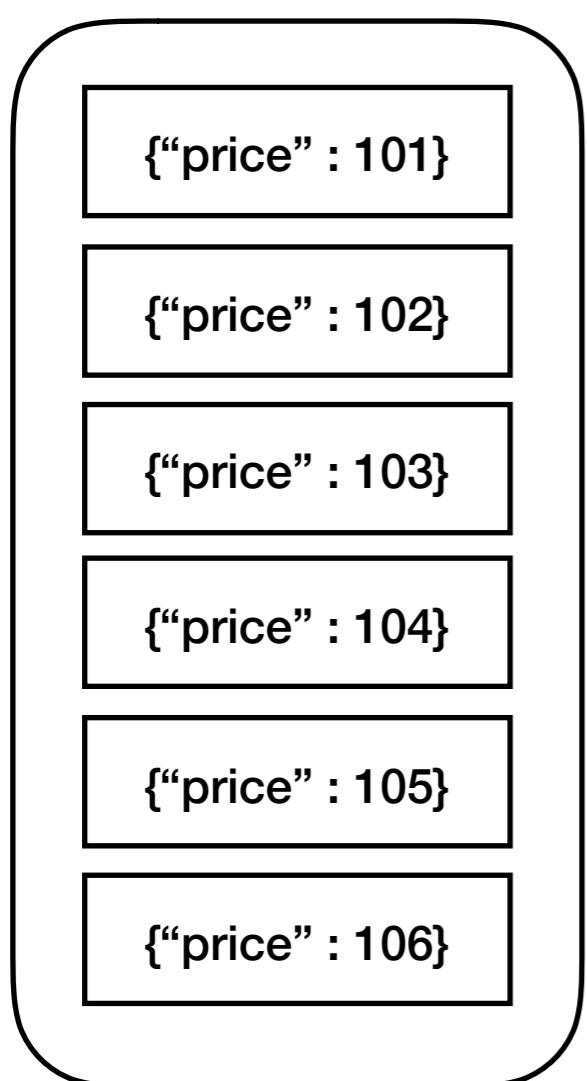
다시 해석해보면 “**파란색 2개, 녹색 1개, 자주색 1개**”이다. 유의미한가?

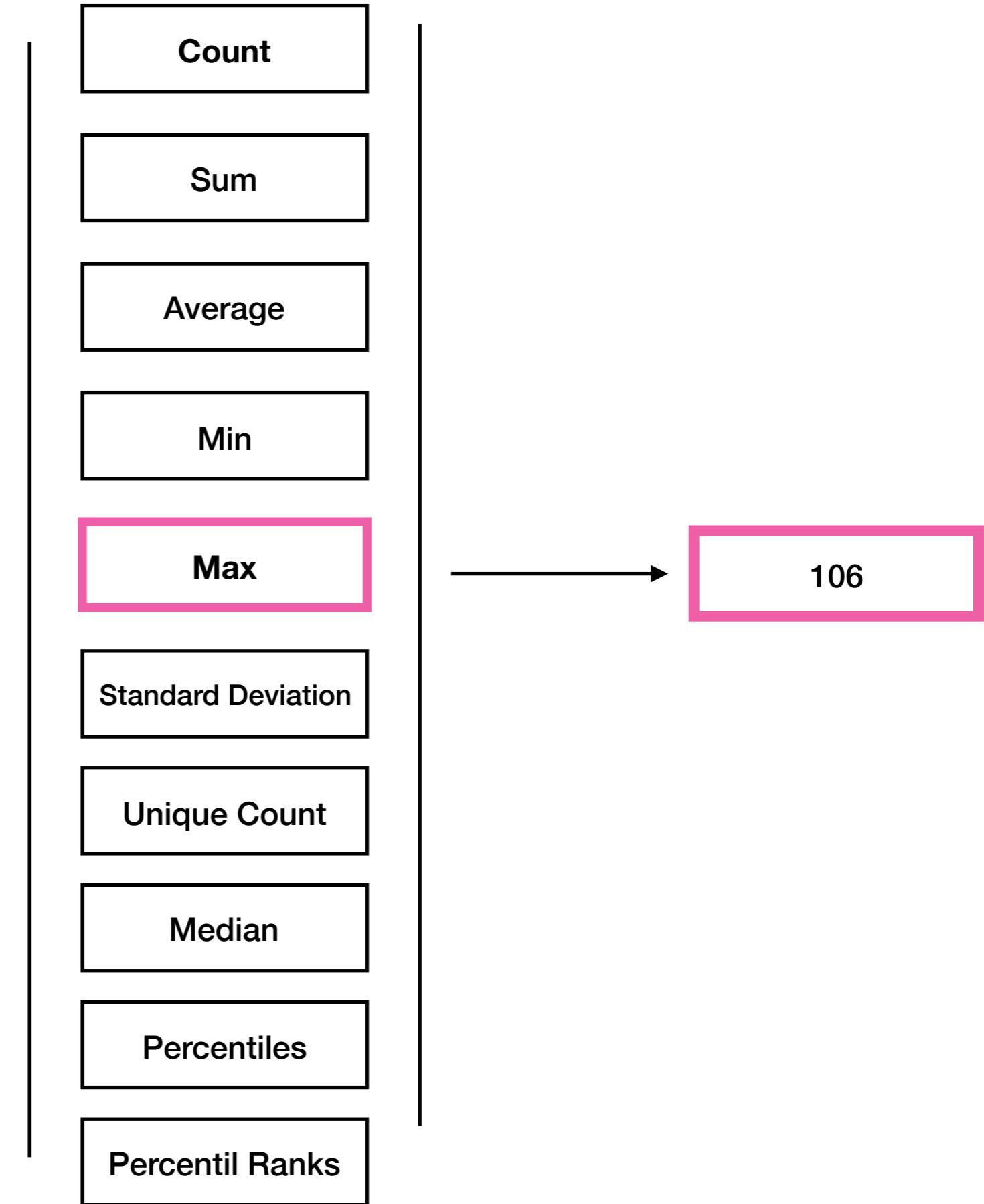
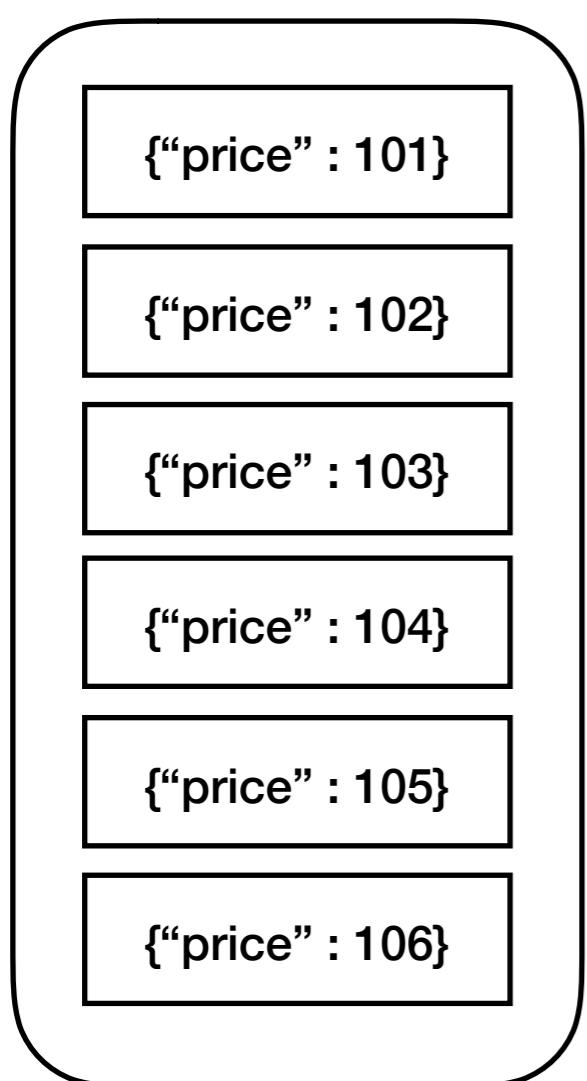
**Metrics Aggregation** = (Bucket 내의 Documents 단위로) 특정 연산을 수행하는 Aggregation

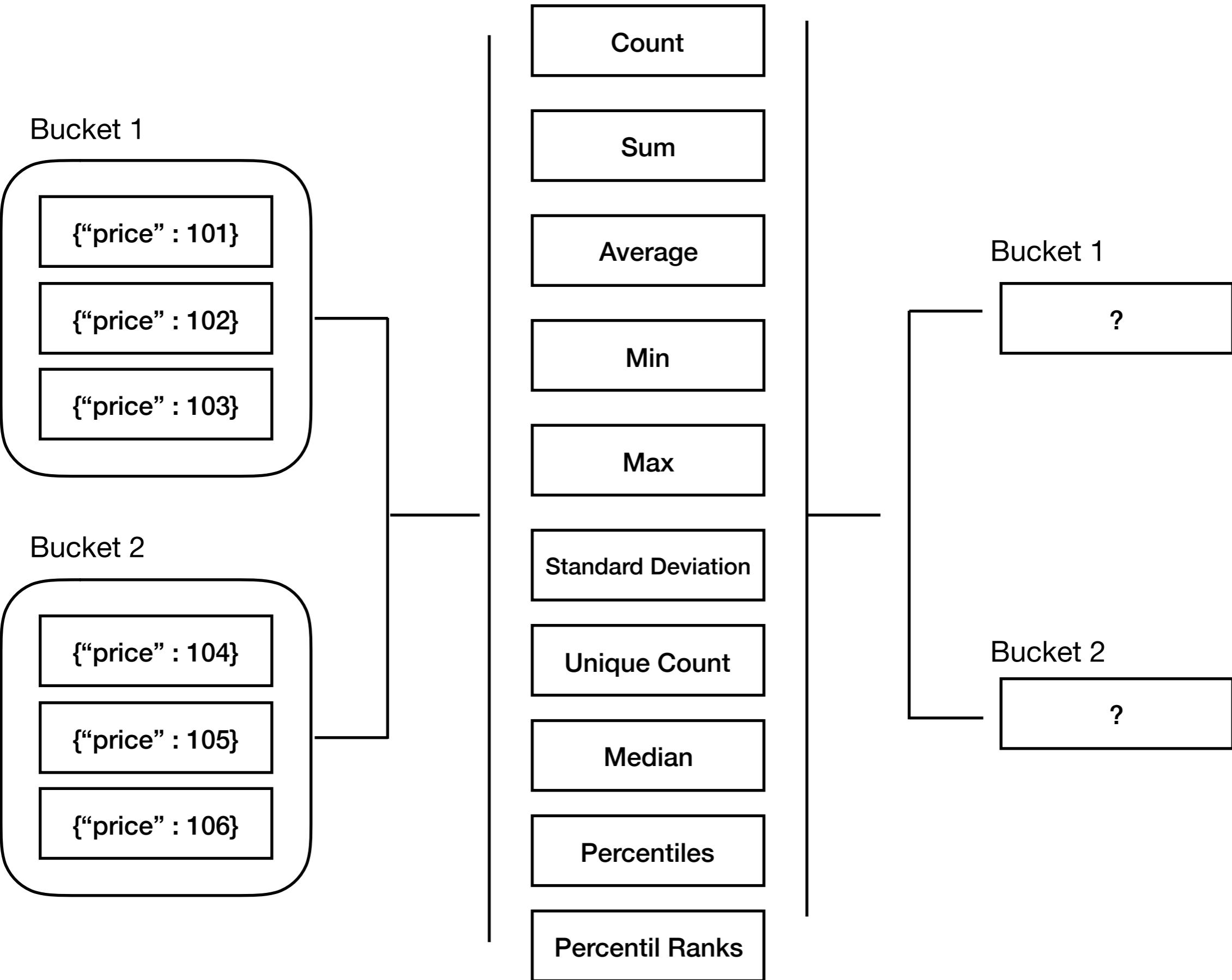
## Kibana가 지원하는 Metrics Aggregation

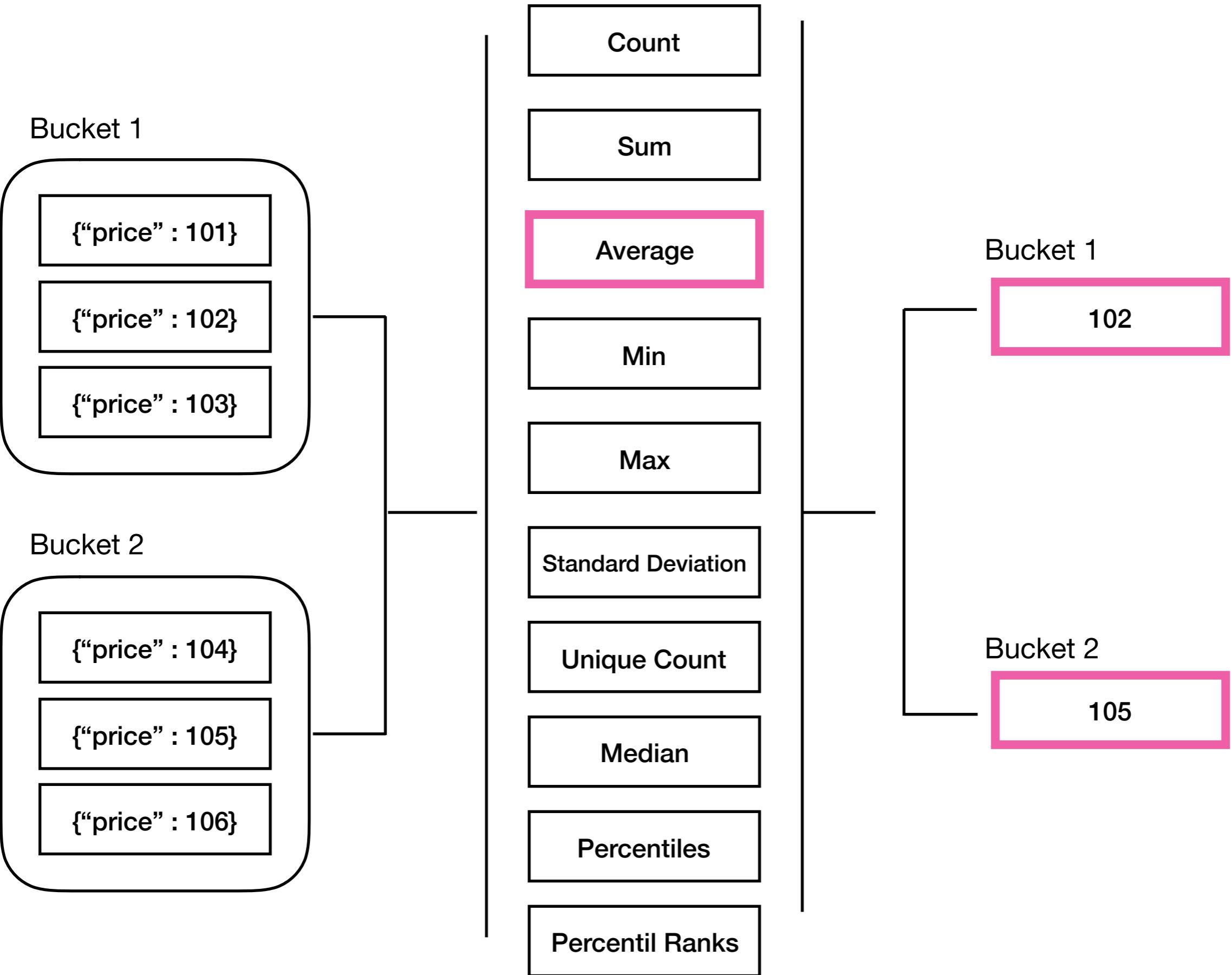
종류	적용 가능 Type	상세
Value Count	All	(Bucket 내) Document의 개수 계산
Avg	Number	(Bucket 내) Document의 특정 Field Values의 평균 계산
Sum	Number	(Bucket 내) Document의 특정 Field Values의 합 계산
Min/Max	Number	(Bucket 내) Document의 특정 Field Values의 최소/최대 계산
Extended Stats	Number	(Bucket 내) Document의 특정 Field Values의 기초 통계값 계산
Cardinality	Number	(Bucket 내) Document의 특정 Field Values의 고유한 개수 계산
Percentiles	Number	(Bucket 내) Document의 특정 Field Values의 백분위수 계산
Percentiles Ranks	Number	(Bucket 내) Document의 특정 Field Value의 백분위 계산
Top Hits	All	(Bucket 내) 특정 조건을 만족하는 Documents의 특정 Field Values의 Agg 반환

- 
- Number Field : Concat, Sum, Min, Max, Count
  - 기타 Field : Concat









{“번호” : 1, “날짜” : “10-01”, “역” : 강남 }

{“번호” : 2, “날짜” : “10-11”, “역” : 신사 }

{“번호” : 3, “날짜” : “10-30”, “역” : 역삼 }

{“번호” : 4, “날짜” : “10-10”, “역” : 송내 }

{“번호” : 5, “날짜” : “10-05”, “역” : 선릉 }

{“번호” : 6, “날짜” : “10-06”, “역” : 언주 }

{“번호” : 7, “날짜” : “10-07”, “역” : 잠원 }

{“번호” : 8, “날짜” : “10-08”, “역” : 시청 }

## Top Hits Aggregation

날짜가	빠른	데이터 3개의	번호	합을 구하세요
번호가	작은	데이터 2개의	역명	모두 나열하세요
역명이	빠른	데이터 2개의	번호	평균을 구하세요

{“번호” : 1, “날짜” : “10-01”, “역” : 강남 }

{“번호” : 2, “날짜” : “10-11”, “역” : 신사 }

{“번호” : 3, “날짜” : “10-30”, “역” : 역삼 }

{“번호” : 4, “날짜” : “10-10”, “역” : 송내 }

{“번호” : 5, “날짜” : “10-05”, “역” : 선릉 }

{“번호” : 6, “날짜” : “10-06”, “역” : 언주 }

{“번호” : 7, “날짜” : “10-07”, “역” : 잠원 }

{“번호” : 8, “날짜” : “10-08”, “역” : 시청 }

### Top Hits Aggregation



날짜가 빠른 데이터 3개의 번호 합을 구하세요 → 12

**Aggregation도 해봤으니 이제 Visualization도 바로 할 수 있을까?**

# Data Table

Visualize / New Visualization (unsaved)

Save Share Refresh Auto-refresh This year  Uses lucene query syntax 

Add a filter +

**shopping**

Data Options  

**Metrics**  Metric Count 

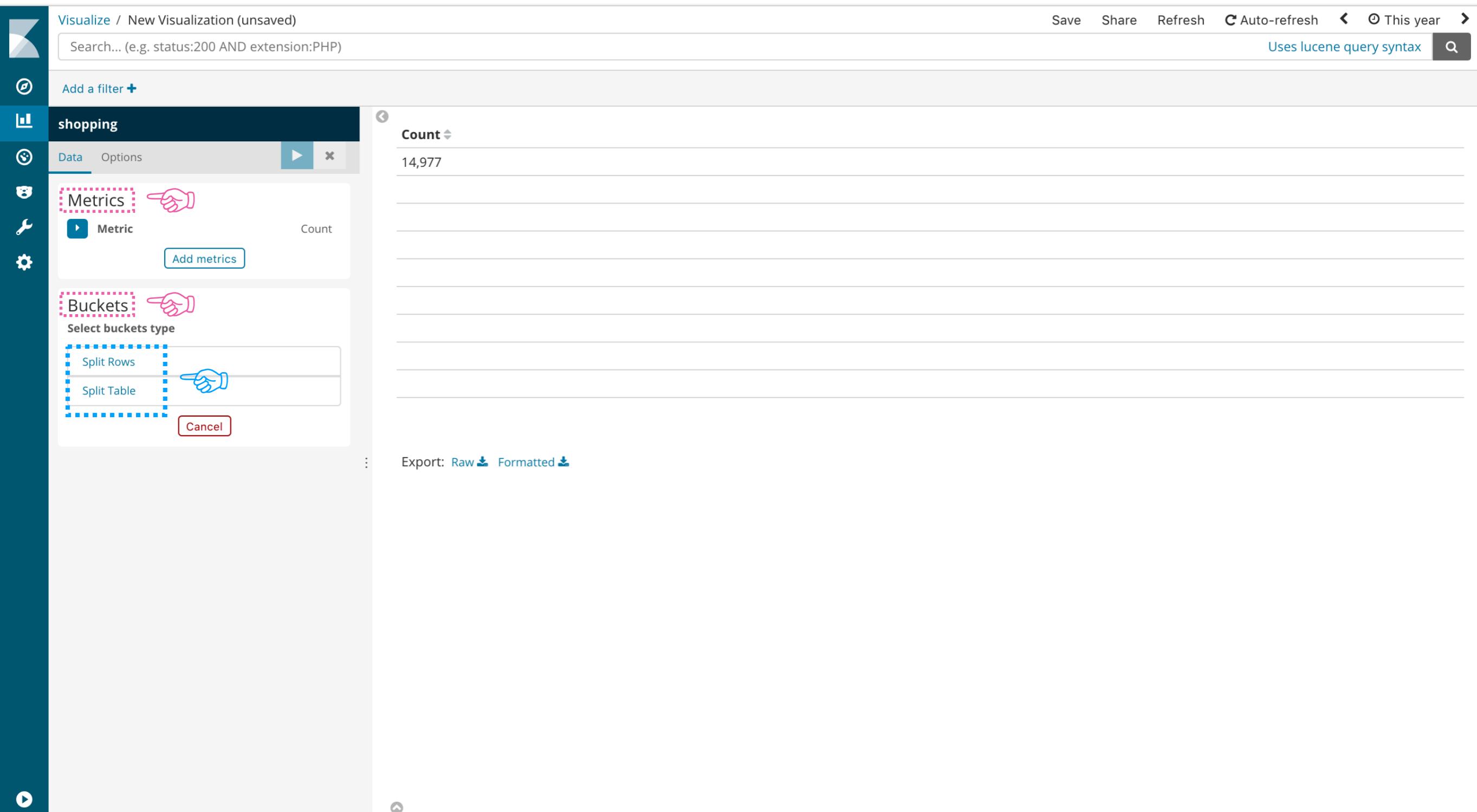
**Buckets**  Select buckets type 

Split Rows  Split Table

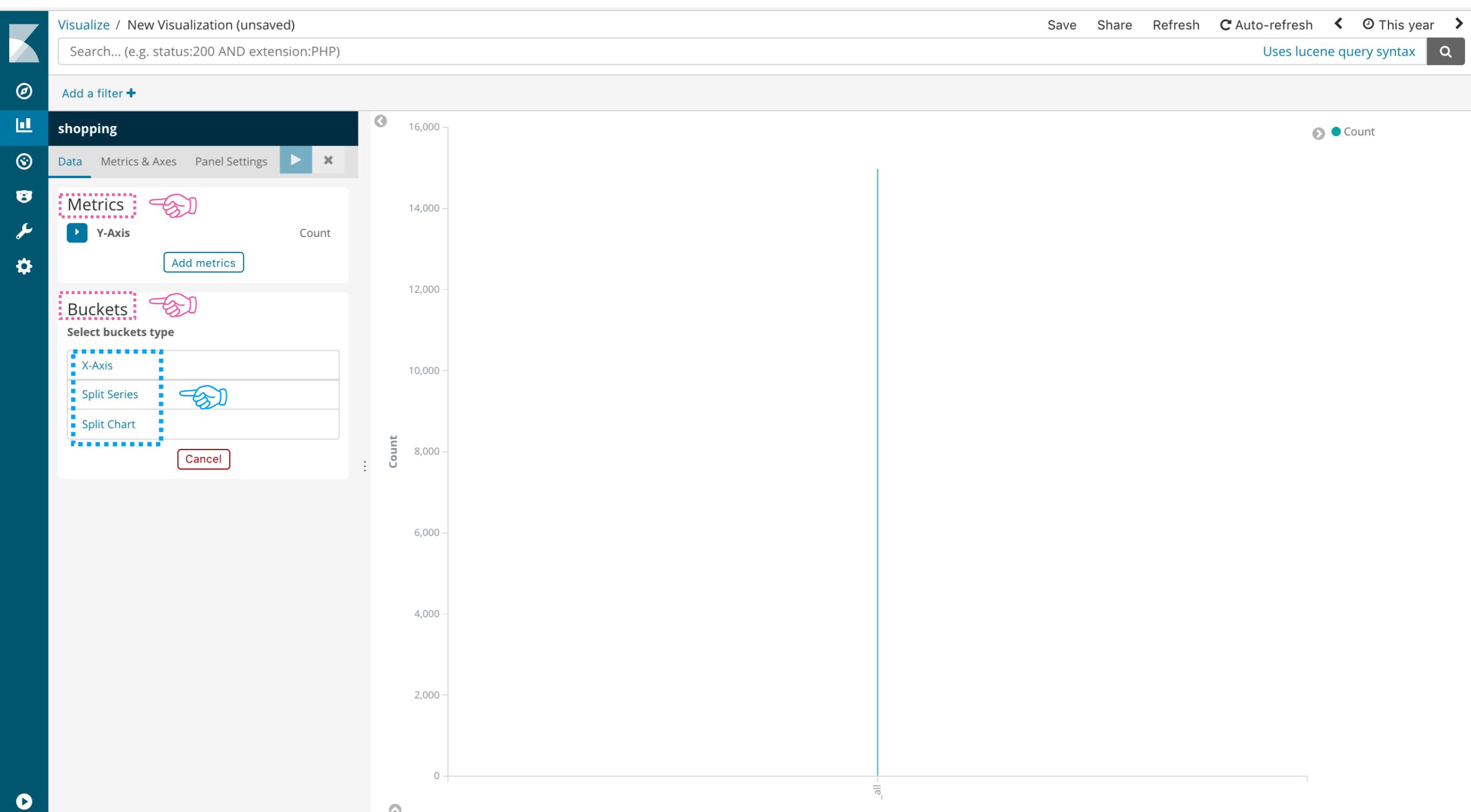
Count 

14,977

⋮ Export: Raw  Formatted 



# Area



metrics, buckets, x-axis, split series, split chart, split rows, split table ...

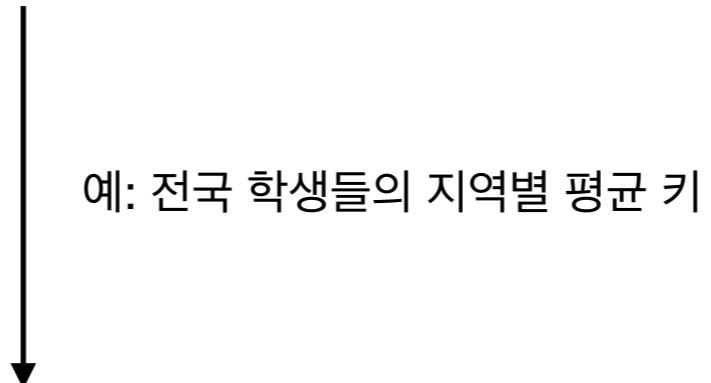
시각화하려는 문제는 명확한데,

어디에 들어가서 어떻게 조작해야되는지 모르겠다

## 1. 큰 틀은 비슷하다

**metrics** : sum, avg, min, max 등 수치 연산을 수행하는 부분

**buckets** : 위의 metrics를 적용할 그룹을 정의하는 부분

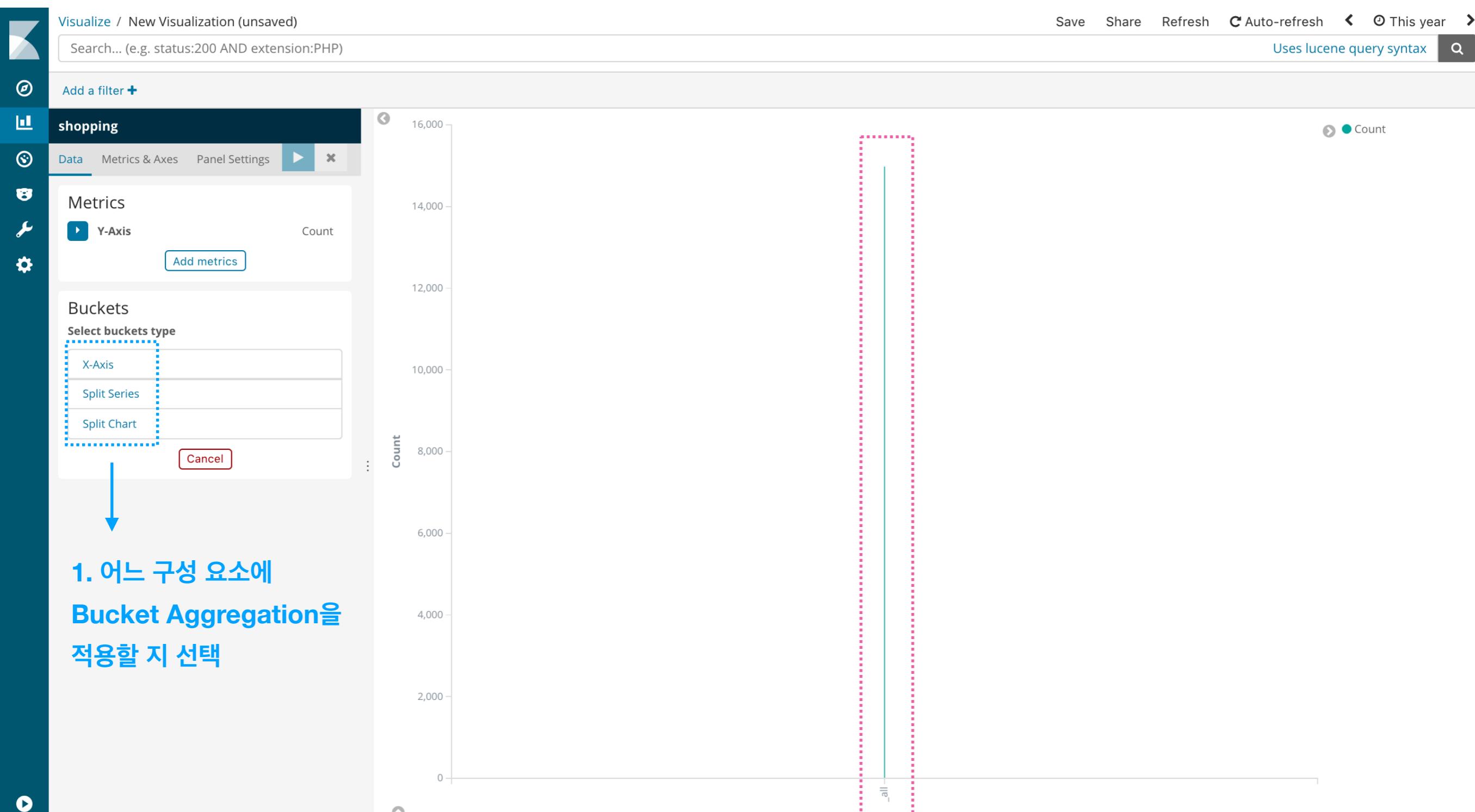


metrics : 키의 평균을 구하는 작업

buckets : 학생들을 지역별로 나누는 작업

## **2. 개별적 구성요소는 Visualization Type마다 상이할 수 있다**

## 대표적인 bucket type 몇 개를 살펴보자 X-Axis Before

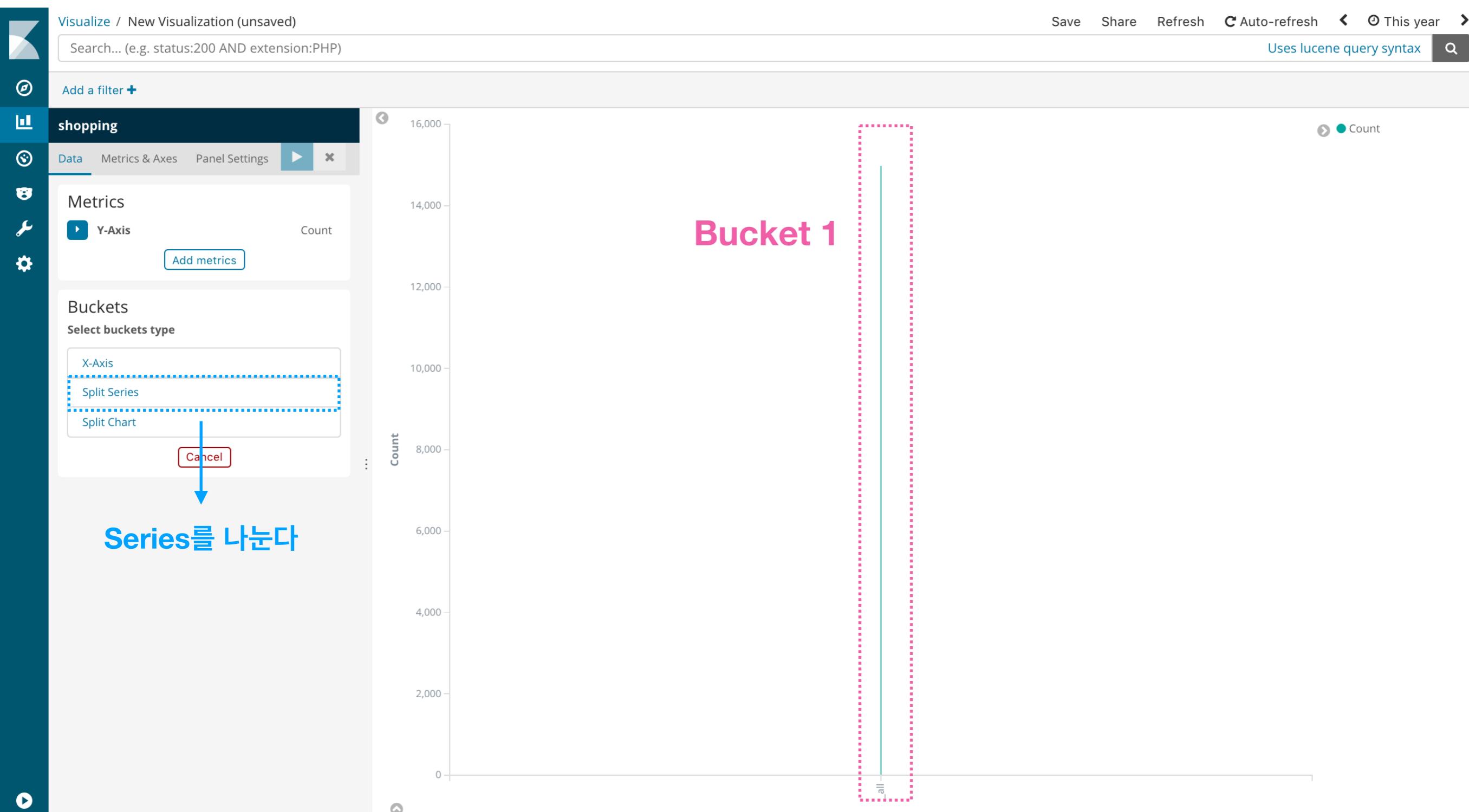


Bucket 1 (\_all)

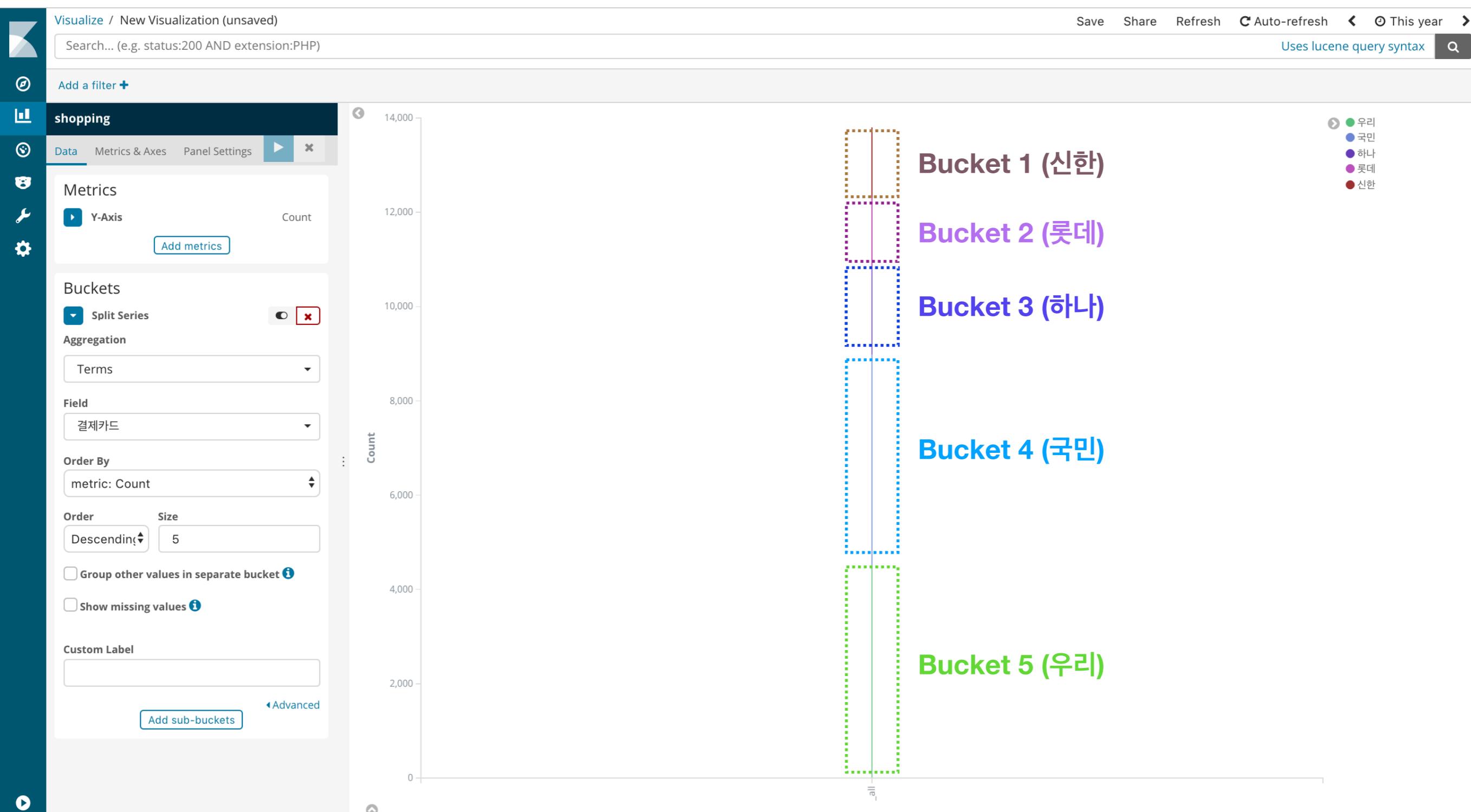
## 대표적인 bucket type 몇 개를 살펴보자 X-Axis After



## 대표적인 bucket type 몇 개를 살펴보자 Split Series Before

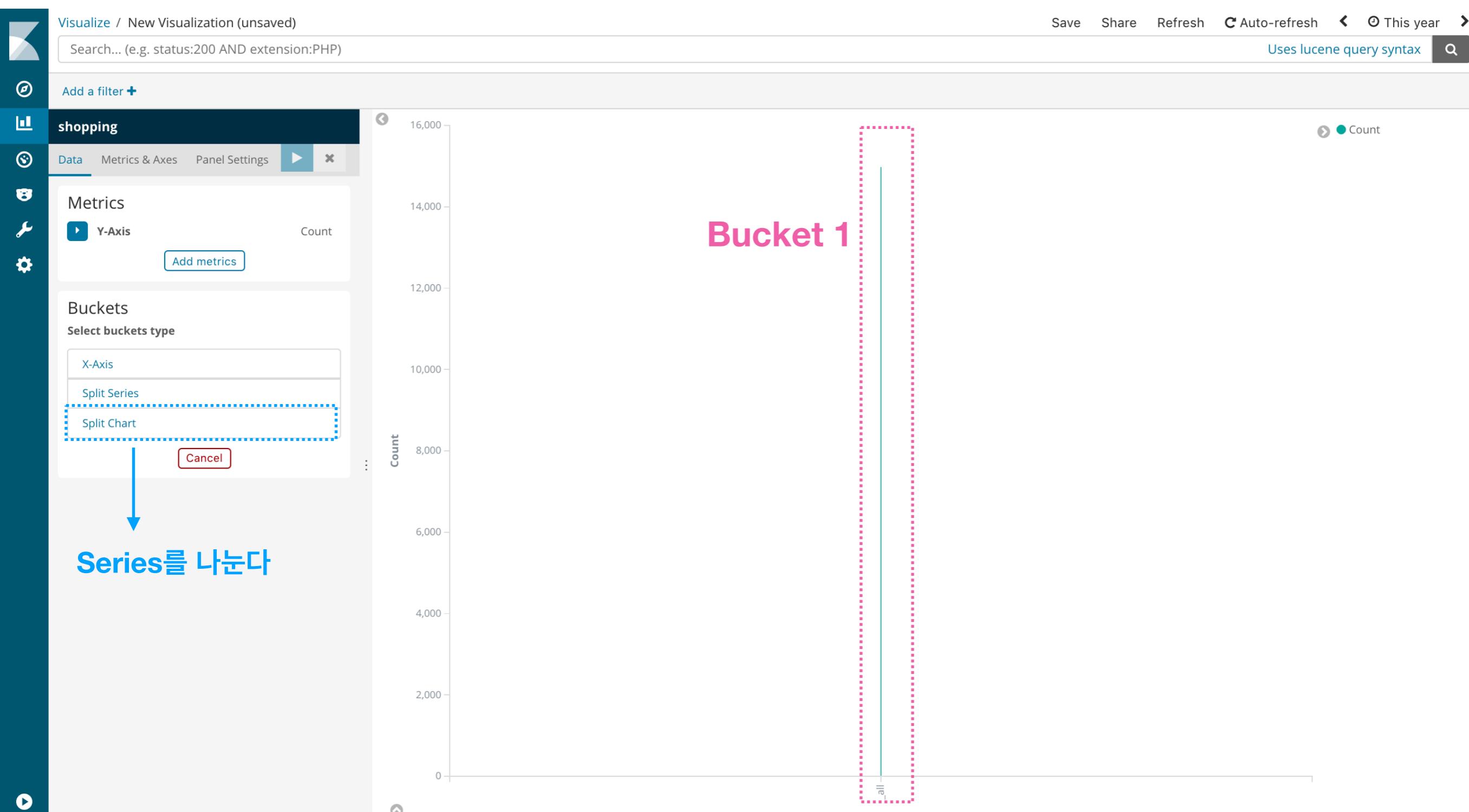


## 대표적인 bucket type 몇 개를 살펴보자 Split Series After



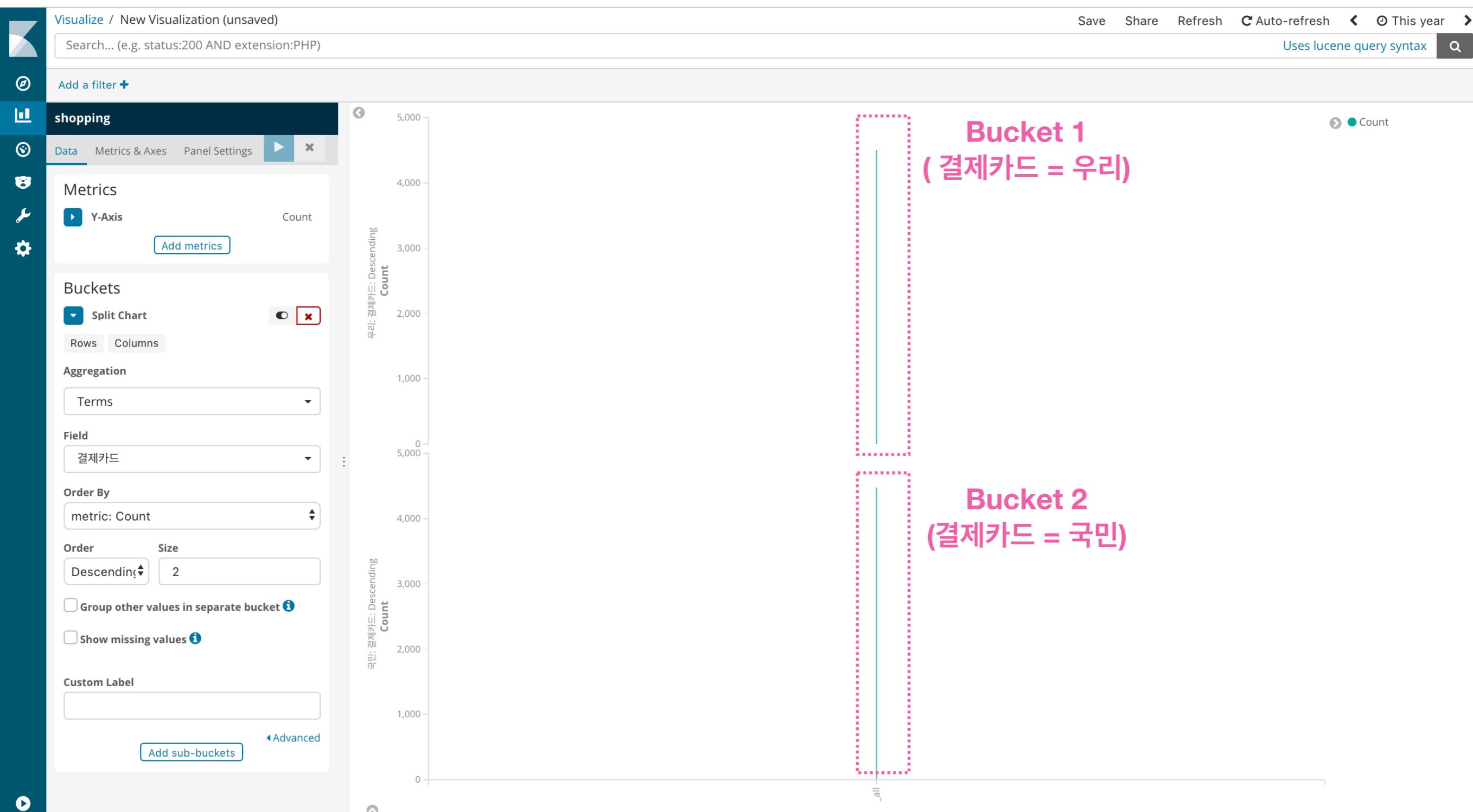
## 대표적인 bucket type 몇 개를 살펴보자 Split Chart

Before



Series를 나눈다

## 대표적인 bucket type 몇 개를 살펴보자 Split Chart After

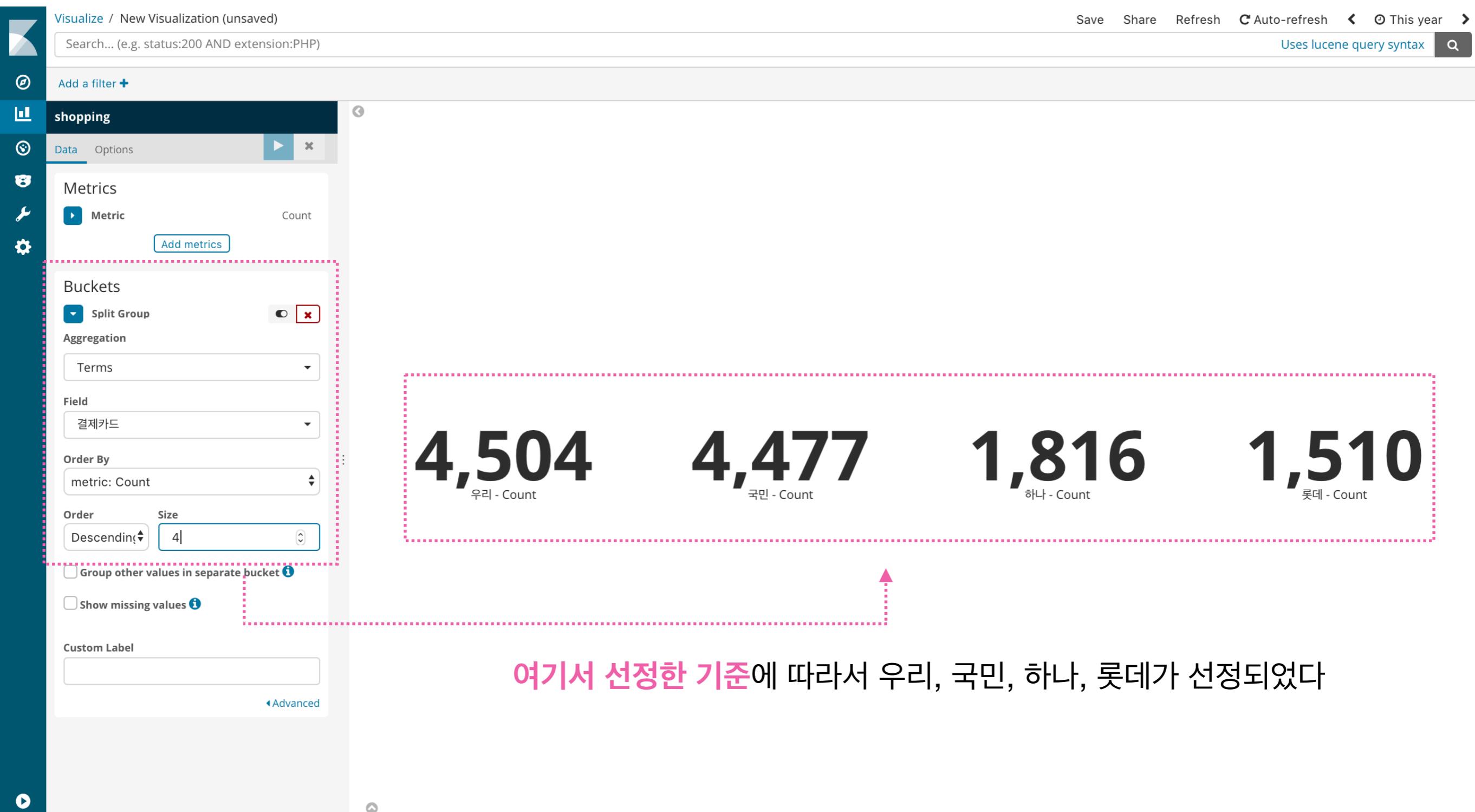


보통은 이 작업의 반복이지만 **Term Aggregation**으로  
Bucket을 나눌 경우 한 단계 더 고려해야한다

- 결제카드 별 사용자 수를 구한다고 하자.
- 모든 결제카드에 대해 구할 수 있지만 특정 4개 카드에 대해서만 본다고 하자.
- 이 때 특정한 카드 4개는 어떻게 선정할까?



이를 위해 Term Aggregation 내에서  
**Bucket을 선정하기 위한 Aggregation**을 수행한다



shopping

Data Options  

Metrics  Metric Count 

Buckets   

Aggregation Terms

Field 결제카드

Order By metric: Count

Order Descending Size 4

Group other values in separate bucket 

Show missing values 

Custom Label



1. 결제카드로 Bucket을 구분해서...
2. Bucket 별 Count를 구하고...
3. Count가 큰 순으로 정렬해서...
4. 상위 4개를 선정해라

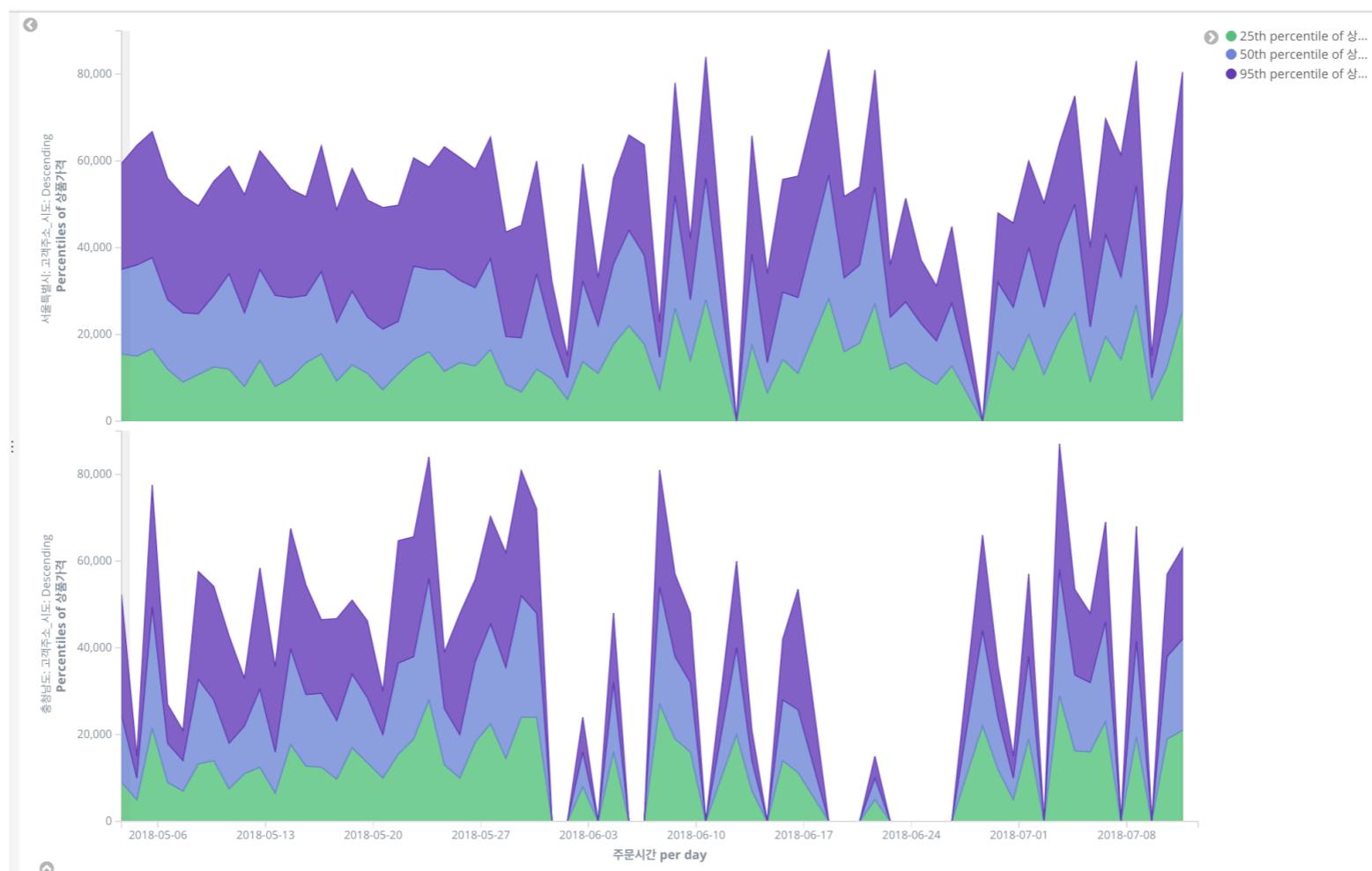
국민	롯데	시티	신한	우리	하나
국민	롯데	시티	신한	우리	하나
4477	1510	720	1490	4504	1816
<b>우리</b>	<b>국민</b>	<b>하나</b>	<b>롯데</b>		
<b>4504</b>	<b>4477</b>	<b>1816</b>	<b>1510</b>		
우리	국민	하나	롯데	신한	시티
4504	4477	1816	1510	1490	720

## 그렇다면 Visualization 문제가 주어지면 어떤 flow로 생각해야 할까?

1. 문제에서 **metrics** 영역과 **buckets** 영역으로 구분한다
2. **metrics**와 **buckets** 내에서 사용할 aggregation을 선택한다
3. term aggregation으로 **bucket**을 나눌 경우 sorting을 위한 aggregation을 정의한다

## 예시를 통해 어떻게 적용하는지 보자

- “상품가격”의 합이 가장 큰
- “고객주소\_시도” 2개의
- “상품가격”의 25백분위수, 50백분위수, 95백분위수를
- “주문시간”을 기준으로 daily로 표시



## 문제에서 metrics 영역과 buckets 영역으로 구분한다

### 문제

• “상품가격”의 합이 가장 큰

• “고객주소\_시도” 2개의

• “상품가격”의 25백분위수, 50백분위수, 95백분위수를

• “주문시간”을 기준으로 daily로 표시해라



Bucket

Meric

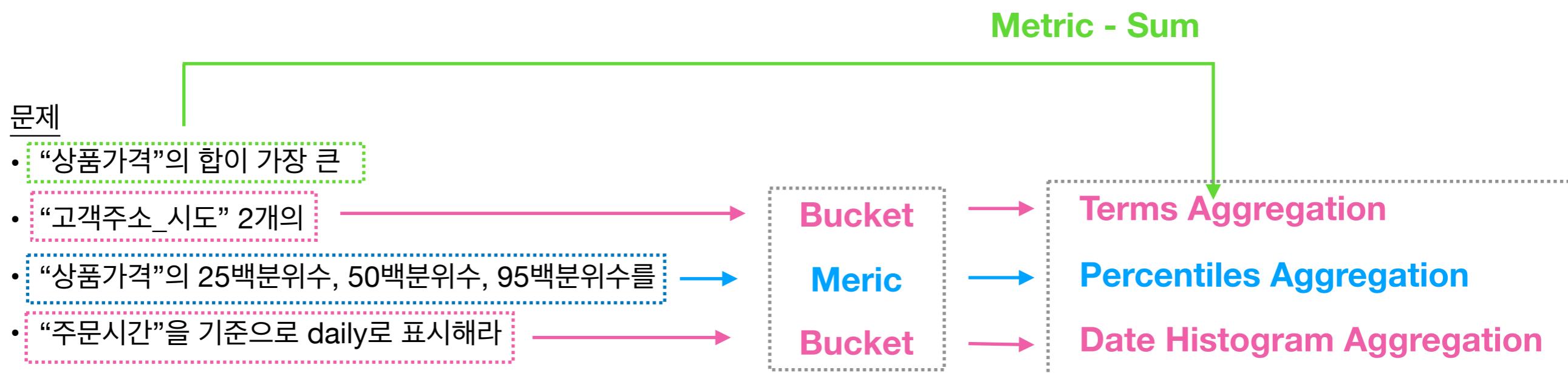
Bucket

## metrics와 buckets 내에서 사용할 aggregation을 선택한다

### 문제

- “상품가격”의 합이 가장 큰  
→ Bucket → Terms Aggregation
- “고객주소\_시도” 2개의  
→ Bucket → Percentiles Aggregation
- “상품가격”의 25백분위수, 50백분위수, 95백분위수를  
→ Metric → Date Histogram Aggregation
- “주문시간”을 기준으로 daily로 표시해라  
→ Bucket → Date Histogram Aggregation

term aggregation으로 bucket을 나눌 경우 sorting을 위한 aggregation을 정의한다



# 이제는 실제로 직접 해보면서 익혀보자



Visualize / New

Select visualization type

Search visualization types...

Basic Charts

- Area
- Heat Map
- Horizontal Bar
- Line
- Pie
- Vertical Bar

Data

- Data Table
- Gauge
- Goal
- Metric

Maps

- Coordinate Map
- Region Map

Time Series

- Timelion
- Visual Builder

# Markdown



- Dashboard 관련 안내 사항 등을 텍스트 형태로 남기고 싶은 경우 이용
- Format은 이름 그대로 Markdown 문법(👑)을 사용해서 지정

## Markdown Object

---

[shopping] markdown

## **Elastic Stack을 활용한 Data Dashboard 만들기 CAMP**

---

- 강의자료
- 강의질문
- Markdown문법

## Markdown Configuration

The screenshot shows a Jupyter Notebook interface with the following elements:

- Left Sidebar:** A vertical toolbar with icons for Visualize, Refresh, Cell, Run, Stop, and Help.
- Title Bar:** "Visualize / [shopping] markdown".
- Toolbar:** Includes a "Font Size (12pt)" slider, a "Help" link, and a "② 실행" (Run) button.
- Code Cell:** Contains the following Markdown content:

```
### Elastic Stack을 활용한 Data Dashboard 만들기 CAMP
---
* [강의자료](https://github.com/higee/elastic)
* [강의질문](https://www.facebook.com/groups/FCElasticStack/)
* [Markdown문법](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)
```
- Bottom Status Bar:** "① 적당한 내용을 입력하자 !"

# Metric

42

Metric

- KPI 같은 지표를 숫자 형태로 시각화하고 싶을 때 사용

## Metric Object

**14,977**  
Count

해석

- shopping index 중에서
- “주문시간” field 기준 this year의
- documents 개수

## Metric Configuration - Count

The screenshot shows the 'Metric Configuration - Count' dialog box. At the top right, there is a time range selector with arrows and the text 'This year'. Below it, a button labeled '④ 실행' (Execute) is highlighted with a pink border. On the left, a 'Metrics' section has a dropdown menu open, showing 'Count' which is also highlighted with a pink border. In the 'Buckets' section, a dropdown menu for 'Select buckets type' contains the option 'Split Group', which is also highlighted with a pink border. To the right of the dialog box, four numbered steps provide instructions:

- ① Time Range를 This year로 설정
- ② Count aggregation 선택
- ③ bucket aggregation은 고정
- ④ 실행

## Metric Object

**16,880.283**

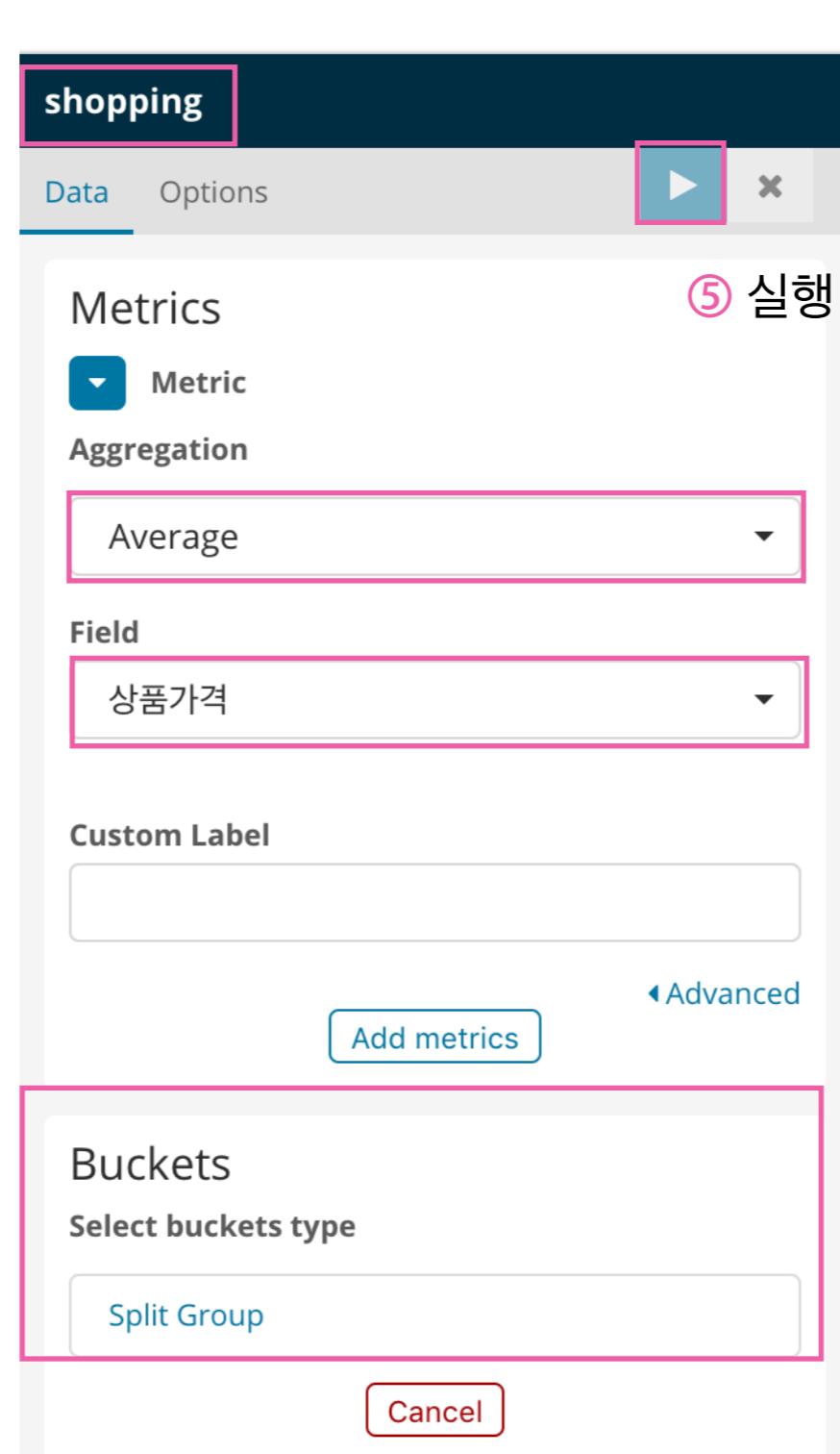
Average 상품가격

해석

- shopping index 중에서
- “주문시간” field 기준 this year documents들의
- “상품가격” field의 평균값

## Metric Configuration - Average

① (Metric 선택 후) shopping index 선택



② Time Range를 This year로 설정

③ Average aggregation 선택

④ Average aggregation을 적용할 Field 선택

⑤ bucket aggregation은 고정

## Metric Object

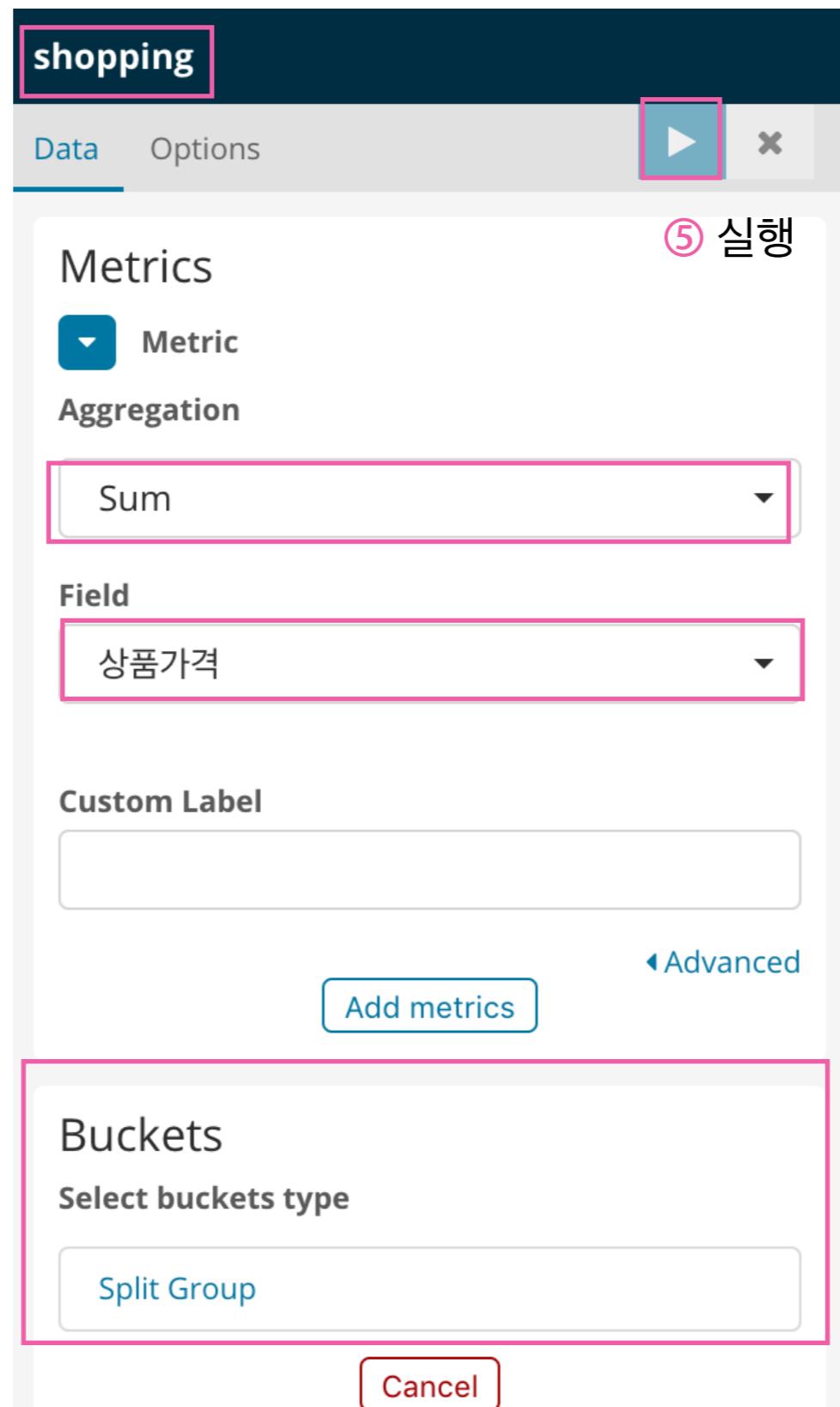
**252,816,000**  
Sum of 상품가격

해석

- shopping index 중에서
- “주문시간” field 기준 this year documents들의
- “상품가격” field의 합

## Metric Configuration - Sum

① (Metric 선택 후) shopping index 선택



① Time Range를 This year로 설정

⑤ 실행

② Sum aggregation 선택

③ Sum aggregation 적용할 Field 선택

④ bucket aggregation은 고정

## Metric Object

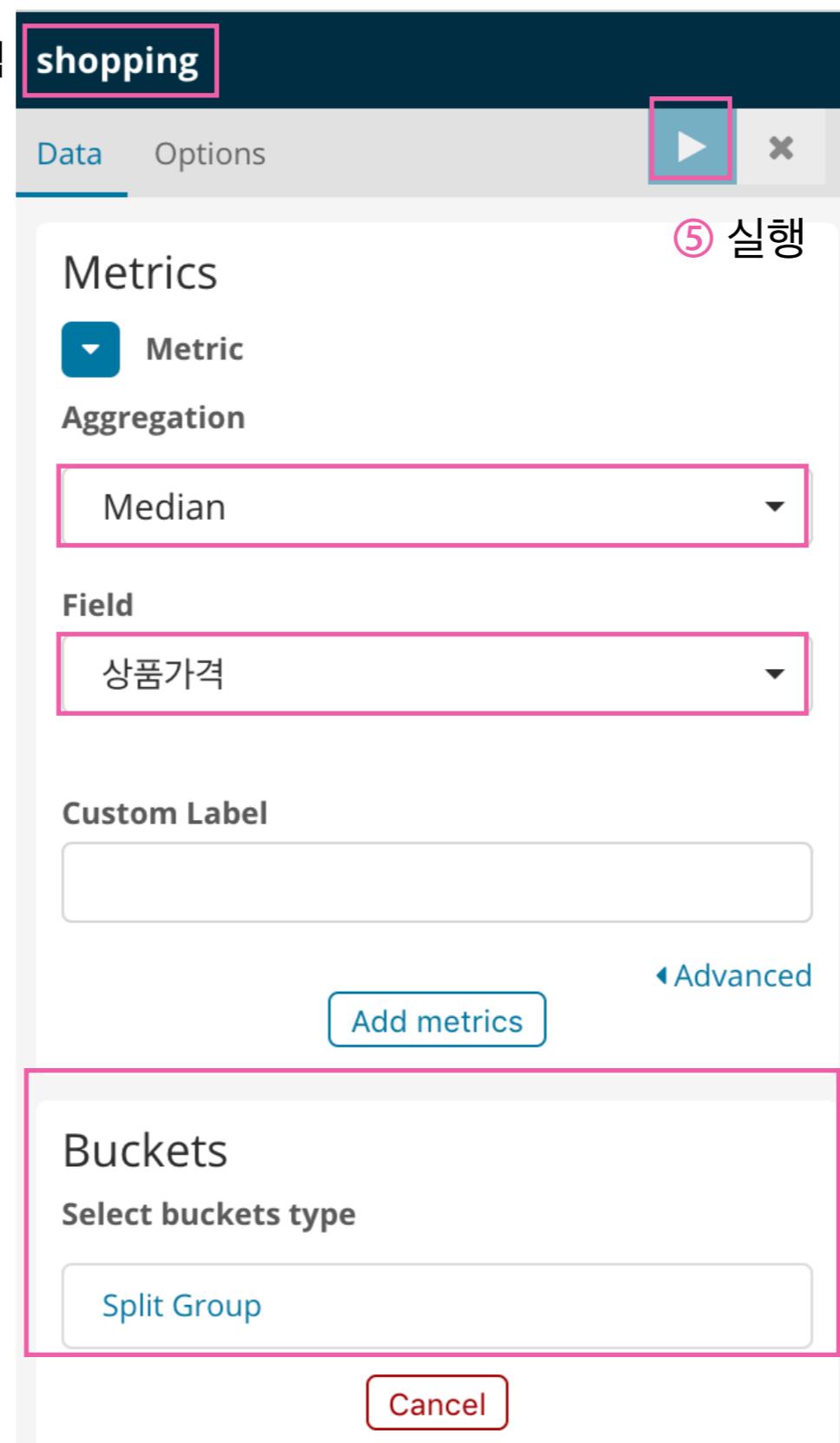
**17,000**  
50th percentile of 상품가격

해석

- shopping index 중에서
- “주문시간” field 기준 this year documents들의
- “상품가격” field의 중위값

## Metric Configuration - Median

① (Metric 선택 후) shopping index 선택



② Time Range를 This year로 설정

③ Median aggregation 적용할 Field 선택

④ bucket aggregation은 고정

## Metric Object

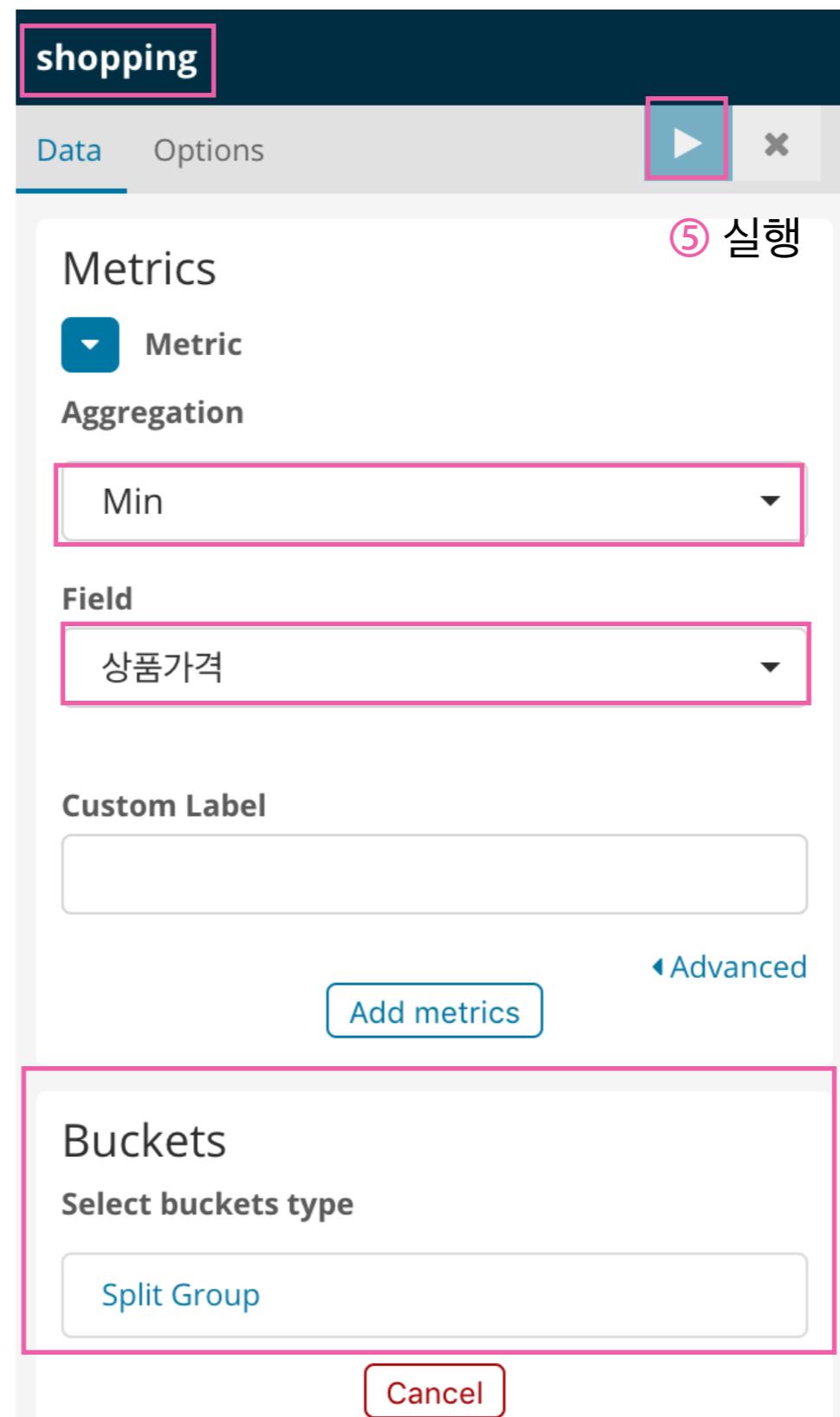
5,000  
Min 상품가격

해석

- shopping index 중에서
- “주문시간” field 기준 this year documents들의
- “상품가격” field의 최소값

## Metric Configuration - Min/Max

① (Metric 선택 후) shopping index 선택



② This year

① Time Range를 This year로 설정

② Min/Max aggregation 선택

③  
④ Min/Max aggregation을  
적용할 Field 선택

④ bucket aggregation은 고정

## Metric Object

# 25

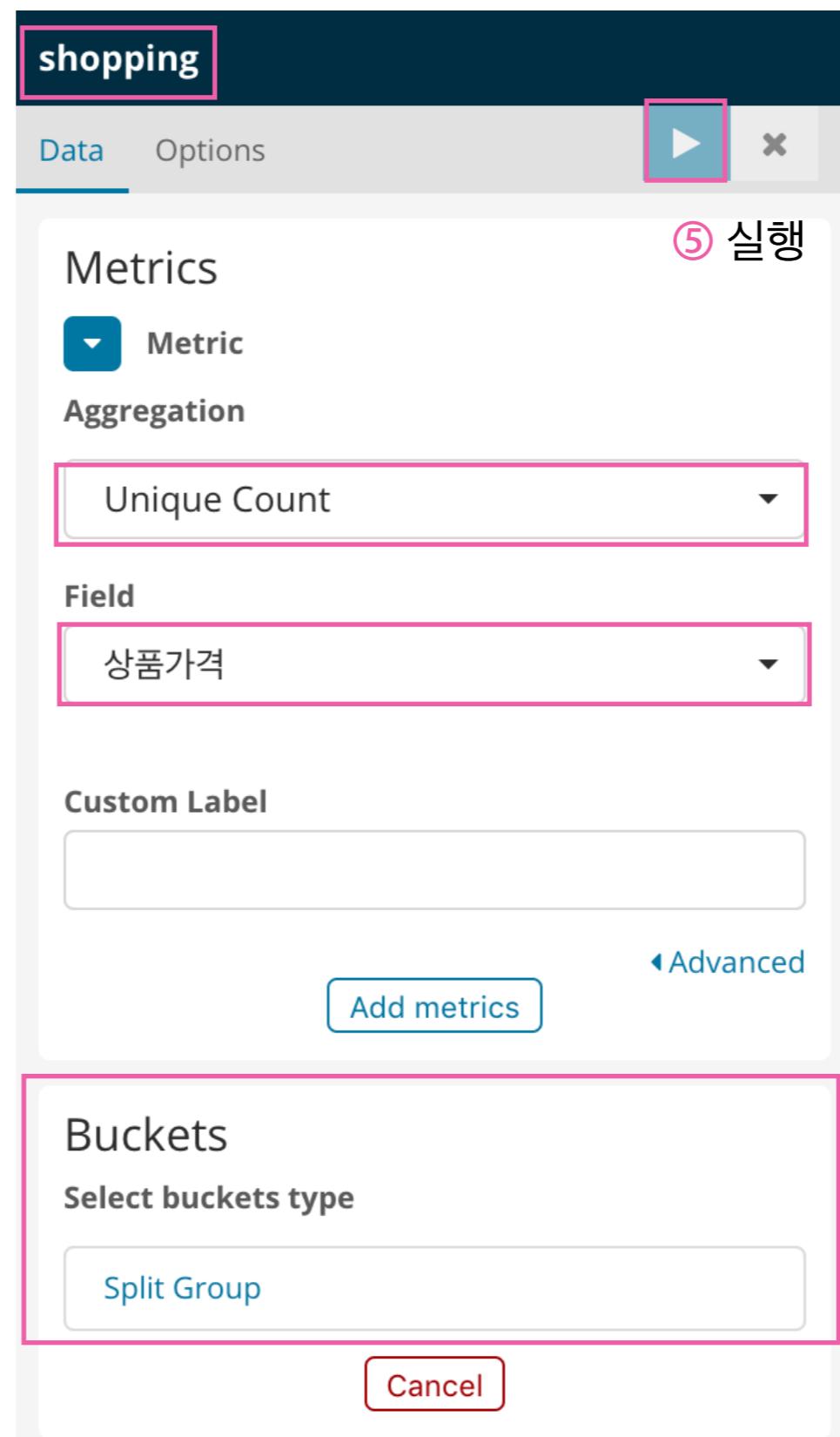
Unique count of 상품가격

해석

- shopping index 중에서
- “주문시간” field 기준 this year documents들의
- “상품가격” field 값의 unique한 개수

## Metric Configuration - Unique Count

① (Metric 선택 후) shopping index 선택



① Time Range를 This year로 설정

② Unique Count aggregation 선택

③ Unique Count aggregation을 적용할 Field 선택

④ bucket aggregation은 고정

## Metric Object

**5,000**

1st percentile of 상품가격

**17,000**

50th percentile of 상품가격

**29,000**

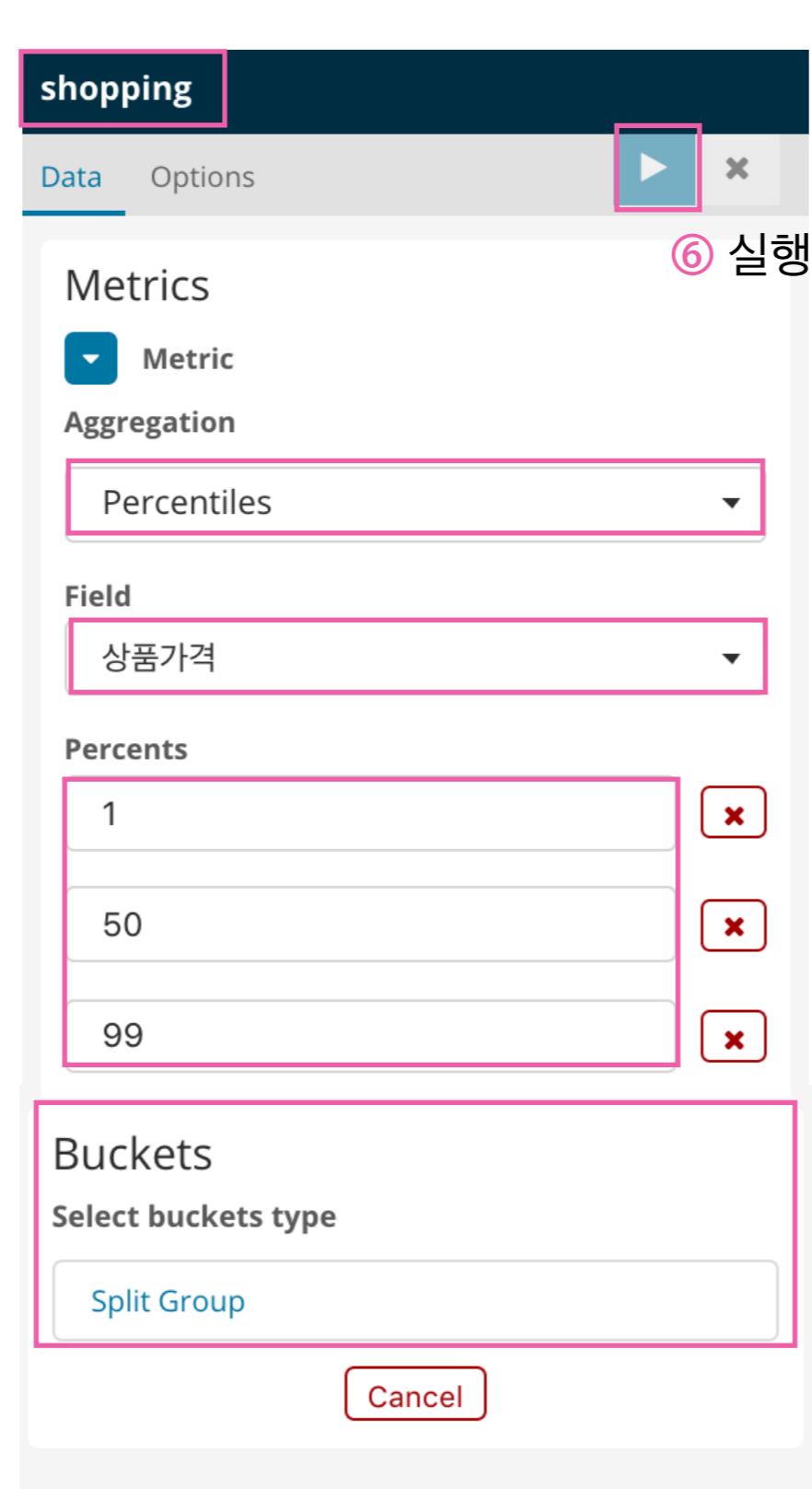
99th percentile of 상품가격

해석

- shopping index 중에서
- “주문시간” field 기준 this year documents들의
- “상품가격” field의 1백분위수, 50백분위수, 99백분위수

## Metric Configuration - Percentiles

① (Metric 선택 후) shopping index 선택



< This year >

① Time Range를 This year로 설정

② Percentiles aggregation 선택

③ Percentiles aggregation을 적용할 Field 선택

④ 백분위수 입력

⑤ bucket aggregation은 고정

## Metric Object

**23.843%**

Percentile rank 10,000 of "상품가격"

**43.901%**

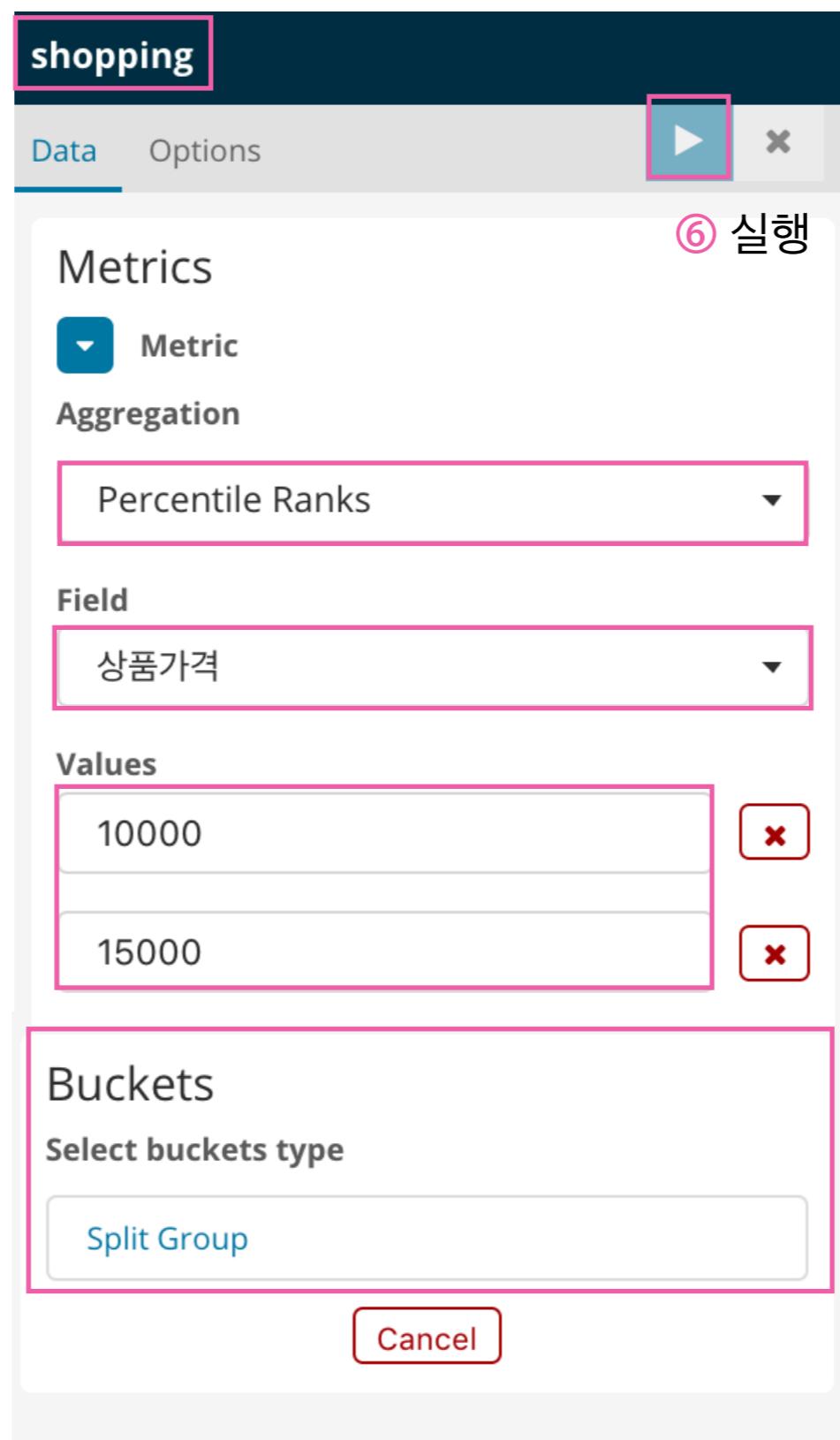
Percentile rank 15,000 of "상품가격"

해석

- **shopping** index 중에서
- “주문시간” field 기준 **this year** documents 중
- “**상품가격**” field 값이 10000, 15000인 데이터의 **백분율**

## Metric Configuration - Percentile Ranks

① (Metric 선택 후) shopping index 선택



④ 백분율을 구하려는 value 입력

① Time Range를 This year로 설정

② Unique Count aggregation 선택

③ Unique Count aggregation을 적용할 Field 선택

⑤ bucket aggregation은 고정

## Metric Object

# 2.9

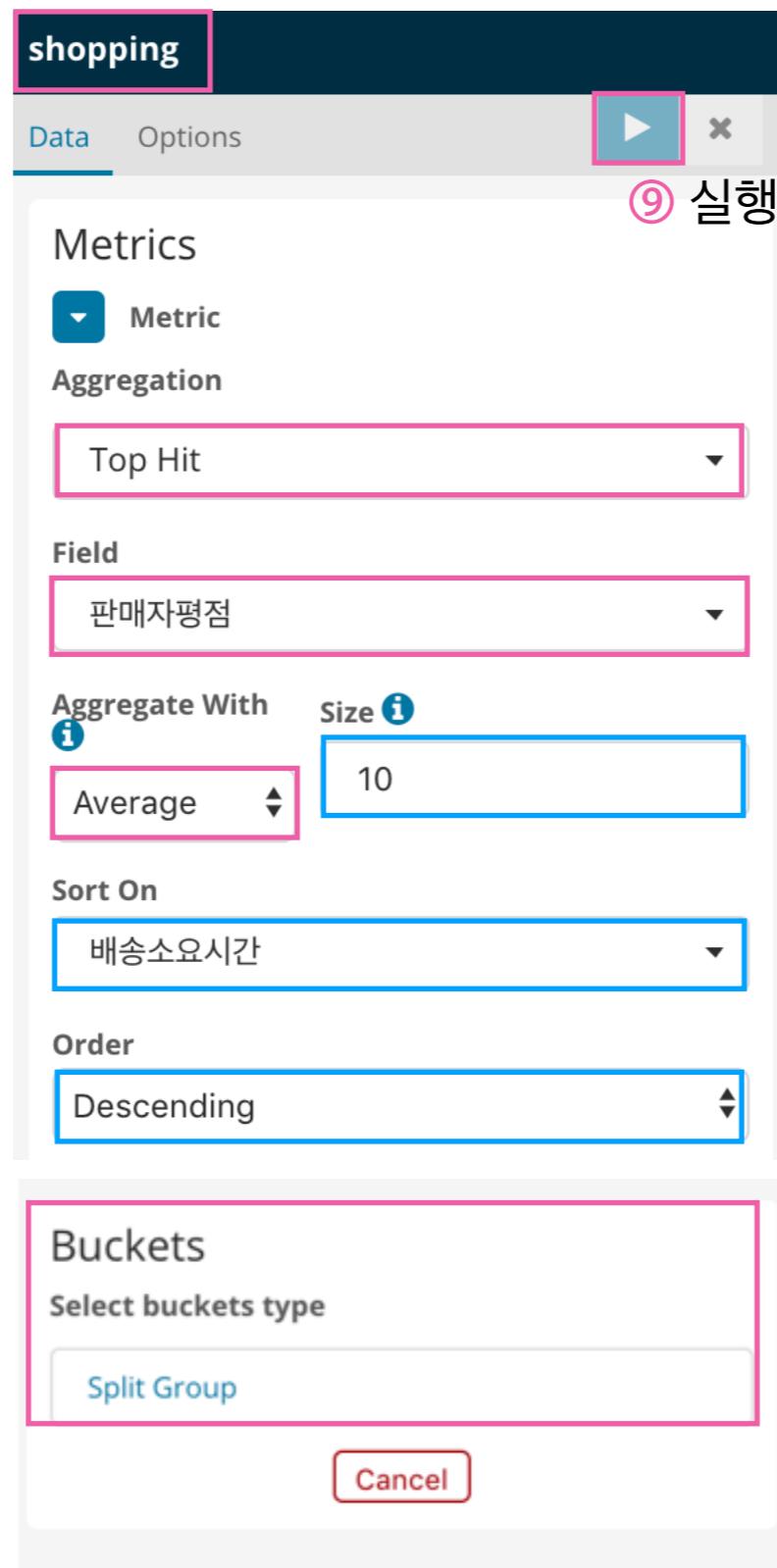
Last 10 판매자평점

해석

- shopping index 중에서
- “주문시간” field 기준 this year documents를
- “배송소요시간” field를 기준으로 내림차순으로 정렬한 뒤
- 상위 10개 Documents의
- “판매자평점” field의 평균값

## Metric Configuration - Top Hit

① (Metric 선택 후) shopping index 선택



① Time Range를 This year로 설정

⑨ 실행

② Top Hit aggregation 선택

⑦

어느 Field에  
Top Hit aggregation을 적용할지 선택

⑥

③~⑤에서 선별한 Documents에  
적용할 Aggregation 선택

⑤ Documents 몇 개를 선택할 건지 입력

③ Documents를 정렬할 기준 Field 선택

④ Documents 정렬 방식 선택 (오름/내림)

⑧ bucket aggregation은 고정

## 예제 1) Metric

**194.491**

50th percentile of nginx.access.body\_sent.bytes

**3,710.9**

95th percentile of nginx.access.body\_sent.bytes

조건

- **nginx-\* index** 중에서
- “@timestamp” field 기준 “**2018년 6월 1일 ~ 2018년 8월 12일**” 사이 documents들의
- “**nginx.access.body\_sent.bytes**” field의 **50백분위수, 95백분위수**

## 예제 2) Metric

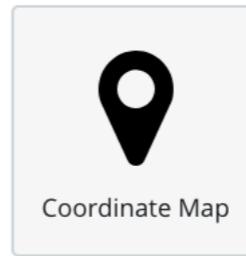
# Windows 10, Mac OS X, iOS

Last 3 nginx.access.user\_agent.os\_name

조건

- nginx-\* index 중에서
- "@timestamp" field 기준 “2018년 6월 1일 ~ 2018년 8월 12일” documents들의
- “nginx.access.body\_sent.bytes” field 값이 가장 큰 documents 3개의
- “nginx.access.user\_agent.os\_name” field 표시

# Coordinate Map



- geo\_point field를 지도에 시각화 할 때 사용 (다른 field 사용 불가)
- zoom 정도에 따라 clustering해서 결과 보여줌
- buckets aggregation은 Geohash aggregation만 지원

## Coordinate Map Object



### 해석

- **nginx-\* index** 중에서
- “@timestamp” field 기준 **this year** documents의
- “**nginx.access.geoip.location**” field에 Geohash aggregation을 적용한 후
- bucket별 (=지도 상 cluster point) documents **개수**

## Coordinates Map Configuration

①

(Coordinate Map 선택 후)  
nginx-\* index 선택

The screenshot shows the configuration interface for a 'Coordinates Map'. At the top, it says 'nginx-\*' with tabs for 'Data' (selected) and 'Options'. A play button icon is highlighted with a pink box, labeled '⑤ 실행' (Execute). In the 'Metrics' section, 'Value' is selected. Below that, under 'Buckets', 'Geo Coordinates' is expanded. In the 'Aggregation' section, 'Geohash' is selected from a dropdown menu. Under 'Field', 'nginx.access.geoip.location' is selected. There are three checked checkboxes at the bottom: 'Change precision on map zoom', 'Place markers off grid (use geocentroid)', and 'Only request data around extent of map'. An information icon (i) is next to the third checkbox. At the bottom, there's a 'Custom Label' input field and a 'Advanced' link.

◀ ⏪ This year ⏩ ▶

① Time Range를 This year로 설정

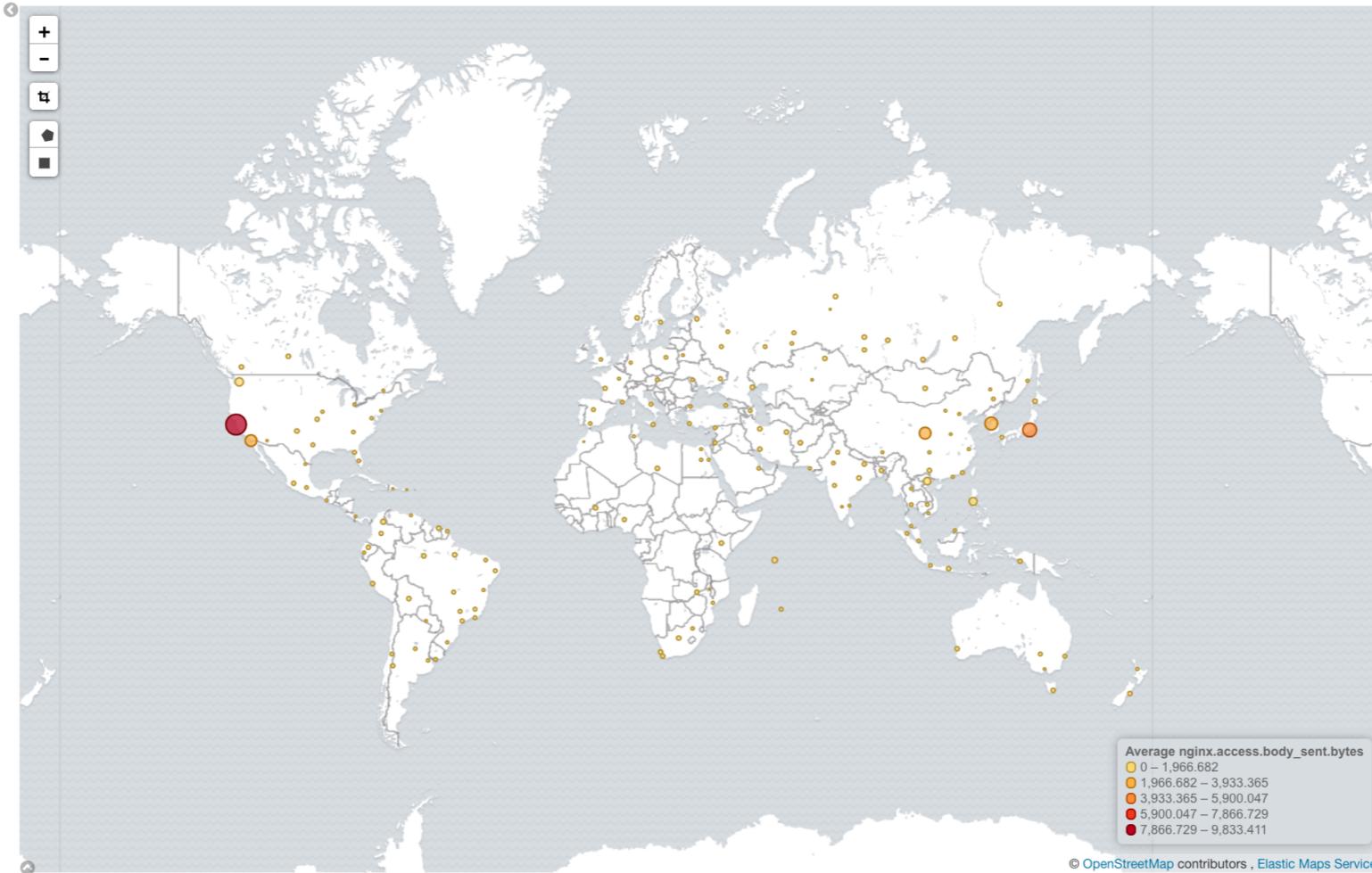
② Metrics aggregation은 고정

③ Geohash aggregation 선택 (필수)

④

Geohash aggregation을  
적용할 Field 선택

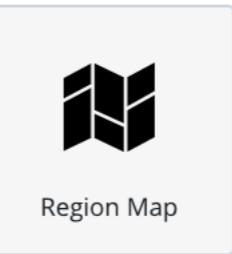
### 예제 3) Coordinate Map



조건

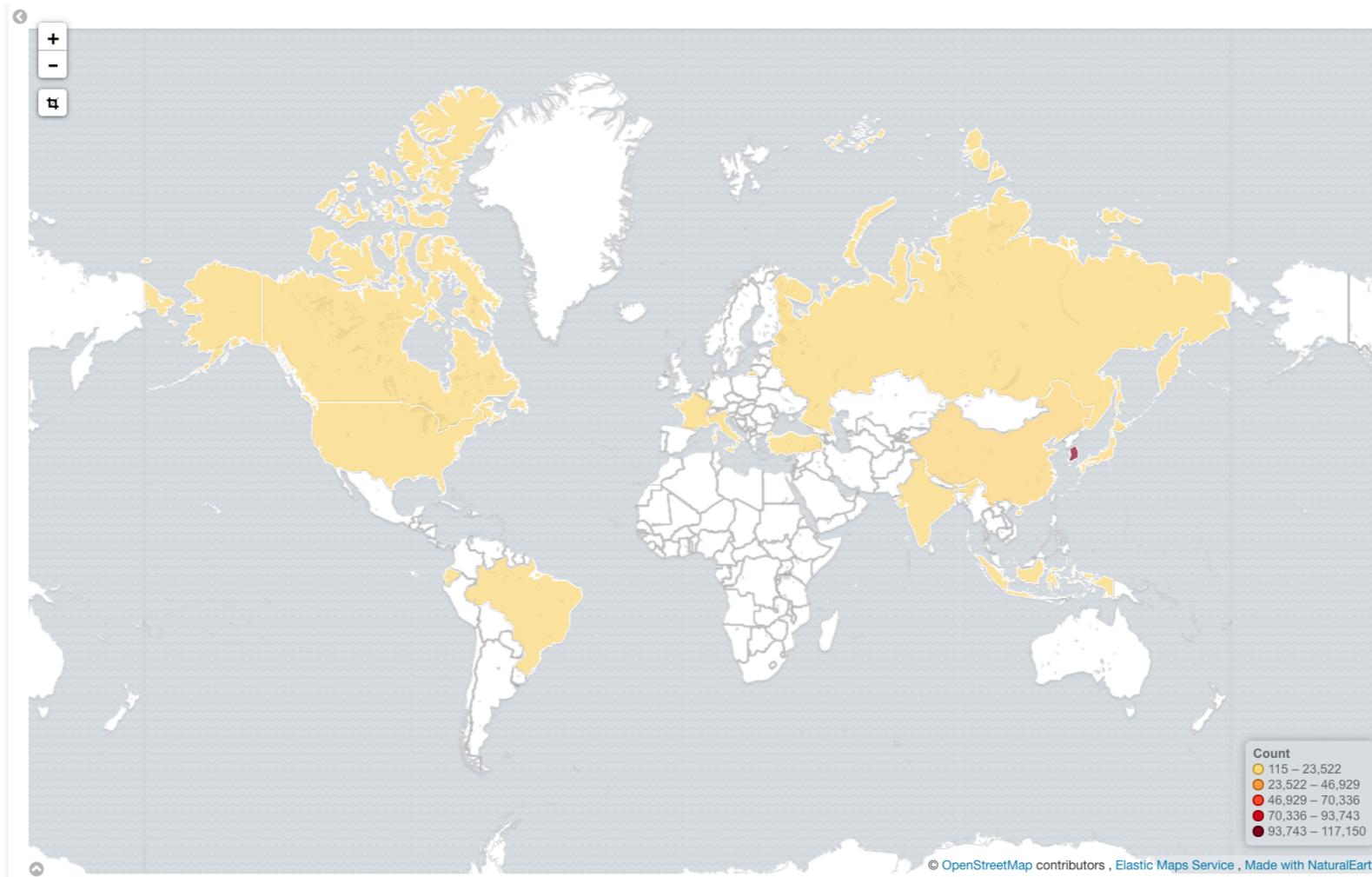
- **nginx-\* index** 중에서
- “@timestamp” 기준 **this year** documents의
- “**nginx.access.geoip.location**” field에 Geohash aggregation을 적용한 후
- **nginx.access.body\_sent.bytes** field의 **평균값** 시각화

# Region Map



- 동, 구, 시, 국가 등의 단위로 지도 상에 데이터 시각화
- 단, Kibana에서 default로 제공하는 Vector Map은 제한적
  - World countries
  - Canada provinces
  - China Provinces
  - France Departments
  - Germany States
  - USA States, zip codes
- 한국 행정구역에 매핑하려면 사전 작업 필요 🏛

## Region Map Object



해석

- nginx-\* index의
- "@timestamp" field 기준 this year documents의
- (nginx.access.geoip.country\_code2 field 별로 documents 개수를 센 후)
- documents 개수가 가장 많았던 nginx.access.geoip.country\_code2 field 15개의 ————— 국가별
- documents 개수 ————— 접속자수

## Region Map Configuration

- ①  
(Region Map 선택 후)  
nginx-\* index 선택

nginx-\*

Data Options ⑧ Metrics 클릭 후 page 157 이동

Buckets

shape field

Aggregation

Terms

Field

nginx.access.geoip.country\_code2

Order By

metric: Count

Order

Descending

Size

15

Group other values in separate bucket i

Show missing values i

◀ ⏪ This year ⏩ ▶

- ⑤ Documents 정렬 metric 선택
- ⑥ Documents 정렬 방식 선택  
(오름/내림)

① Time Range를 This year로 설정

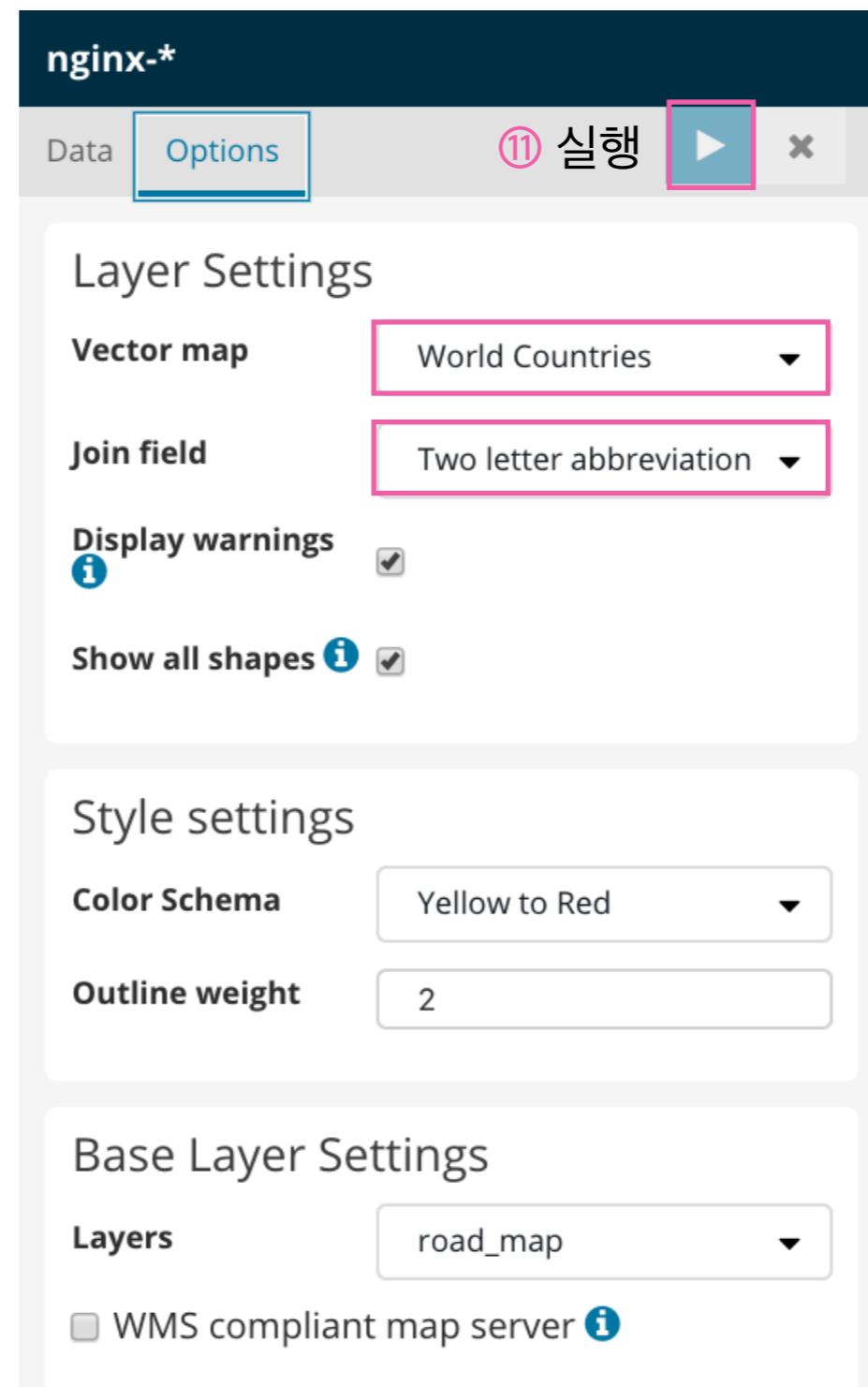
② Metrics aggregation은 고정

③ Terms aggregation 선택 (필수)

- ④
- Terms aggregation 적용할 Field 선택
  - 단, 이 Field는 Vector Map이 인지하는 Field

⑦ 반영할 bucket 개수 입력

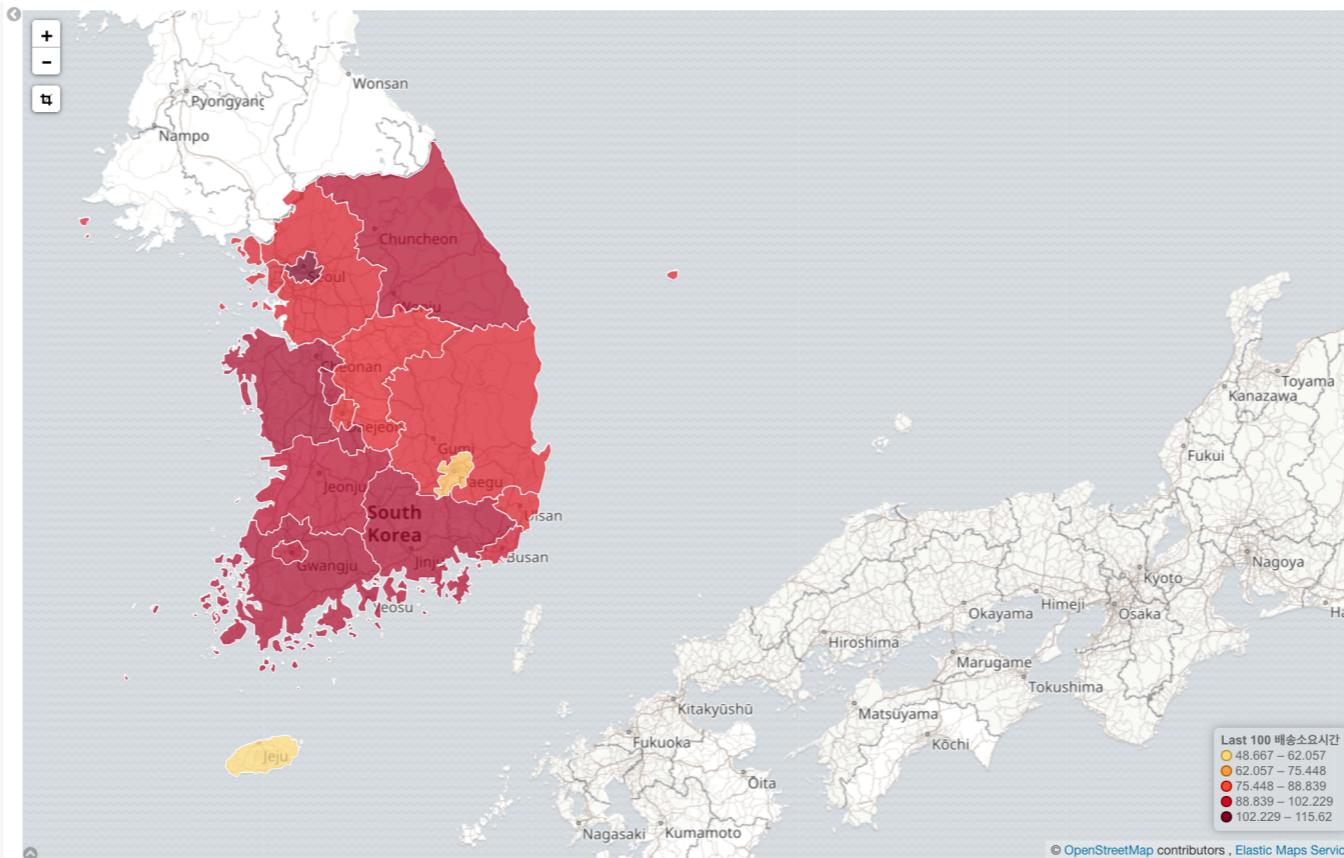
## Region Map Configuration



⑨ World Countries 선택

⑩ Two letter abbreviation 선택

## 예제 4) Region Map



조건

- shopping index의
- “주문시간” 이 this year인 documents의
- “고객주소\_시도” field 별로 (=모든 17개 지역에 대해서)
- “배송소요시간” field 값이 가장 큰 100개 documents를 선별한 후
- “배송소요시간” field의 평균

지역별

평균 배송소요시간

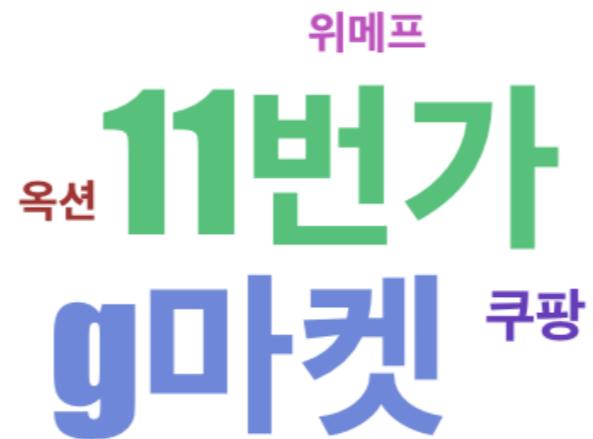
# Tag Cloud



Tag Cloud

- 특정 Field의 value의 중요도 (빈도수 등)을 기준으로 워드 클라우드 형태로 시각화
- 일반적으로 categorical data 등에 적용한다
- Value Count Aggregation 사용시 주의할 점은, document count라는 것이다

## Tag Cloud Object



해석

- shopping index 중에서
- “주문시간” field 기준 this year documents 중에서
- (“구매사이트” field 별로 documents 개수를 센 후에)
- documents 개수가 가장 많았던 “구매사이트” field 5개의
- documents 개수

## Tag Cloud Configuration

①

(Tag Cloud 선택 후)  
shopping index 선택

The screenshot shows the Tag Cloud configuration interface for the 'shopping' index. At the top, there is a navigation bar with tabs for 'Data' and 'Options', a search bar containing 'shopping', and a button labeled '⑧ 실행' (Run). Below the navigation bar, the 'Metrics' section is visible, showing a dropdown menu where 'Tag Size' is selected under 'Count'. The 'Buckets' section includes a 'Tags' dropdown. In the 'Aggregation' section, a dropdown menu is set to 'Terms'. The 'Field' dropdown is set to '구매사이트'. Under 'Order By', the metric 'Count' is selected. In the 'Order' section, 'Descending' is chosen, and the size is set to 5. There are two unchecked checkboxes at the bottom: 'Group other values in separate bucket' and 'Show missing values'.

① This year

① Time Range를 This year로 설정

② Metrics aggregation은 고정

③ Terms aggregation 선택 (필수)

④ Terms aggregation 적용할 Field 선택

⑤

Documents 정렬 metric 선택

⑥

Documents 정렬 방식 선택  
(오름/내림)

⑦ 반영할 bucket 개수 입력

## 예제 5) Tag Cloud

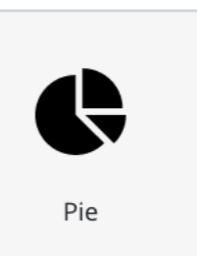


조건

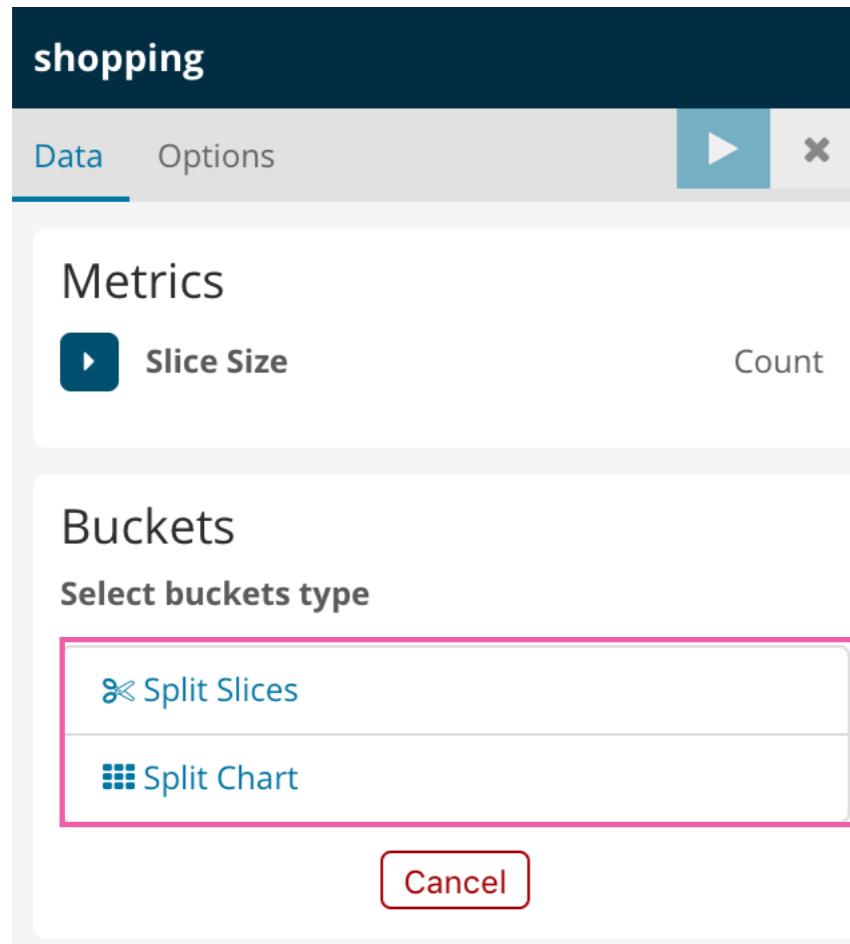
- **nginx-\*** index 중에서
- “@timestamp” field 기준 **this year** documents의
- **nginx.access.body\_sent.bytes** field의 평균이 높았던 **nginx.access.user\_agent.name** field 5개의
- **nginx.access.body\_sent.bytes** field의 중위값

**이제는 Buckets 쪽에 집중해보자**

# Pie Chart

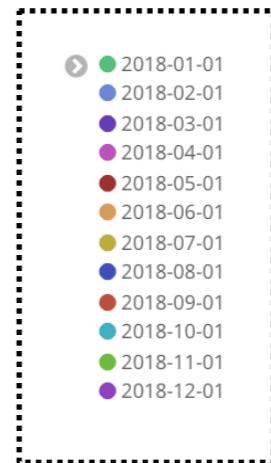


- 특정 Field 값의 분포를 시각화 할 때 유용
- 주로 Categorical Field Data에 적용

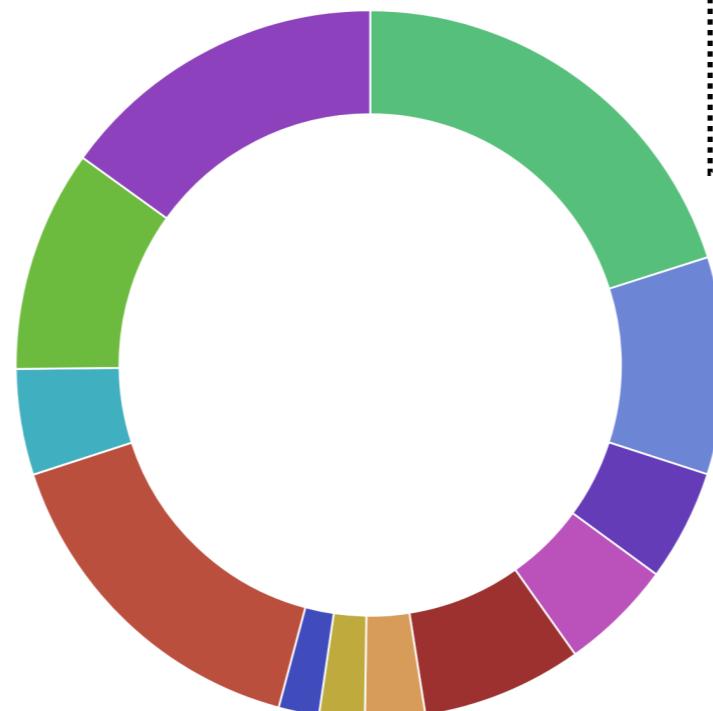


왜 이렇게 나올까?

## Pie Chart Object



\* 2018-01-01 : 2018년 1월



해석

- shopping index 중에서
- “주문시간” field 기준 this year documents를
- “주문시간” field를 기준으로 월별로 나눈 후
- 월별 documents 개수

월별

주문수

## Pie Chart Configuration - Split Slices (Date Histogram)

①

(Pie Chart 선택 후)  
shopping index 선택

The screenshot shows the configuration interface for a pie chart. At the top, there's a header with the index name 'shopping' and tabs for 'Data' and 'Options'. To the right of the header are buttons for navigating time ranges ('This year') and a close button ('x'). Below the header, the configuration is divided into sections:

- Metrics:** A section containing a button for 'Slice Size' and a dropdown for 'Count'.
- Buckets:** A section with a dropdown for 'Split Slices' and a toggle switch that is currently off.
- Aggregation:** A section with a dropdown menu set to 'Date Histogram'.
- Field:** A dropdown menu set to '주문시간'.
- Interval:** A dropdown menu set to 'Monthly'.

◀ ⏪ This year ⏩ ▶

① Time Range를 This year로 설정

⑥ 실행

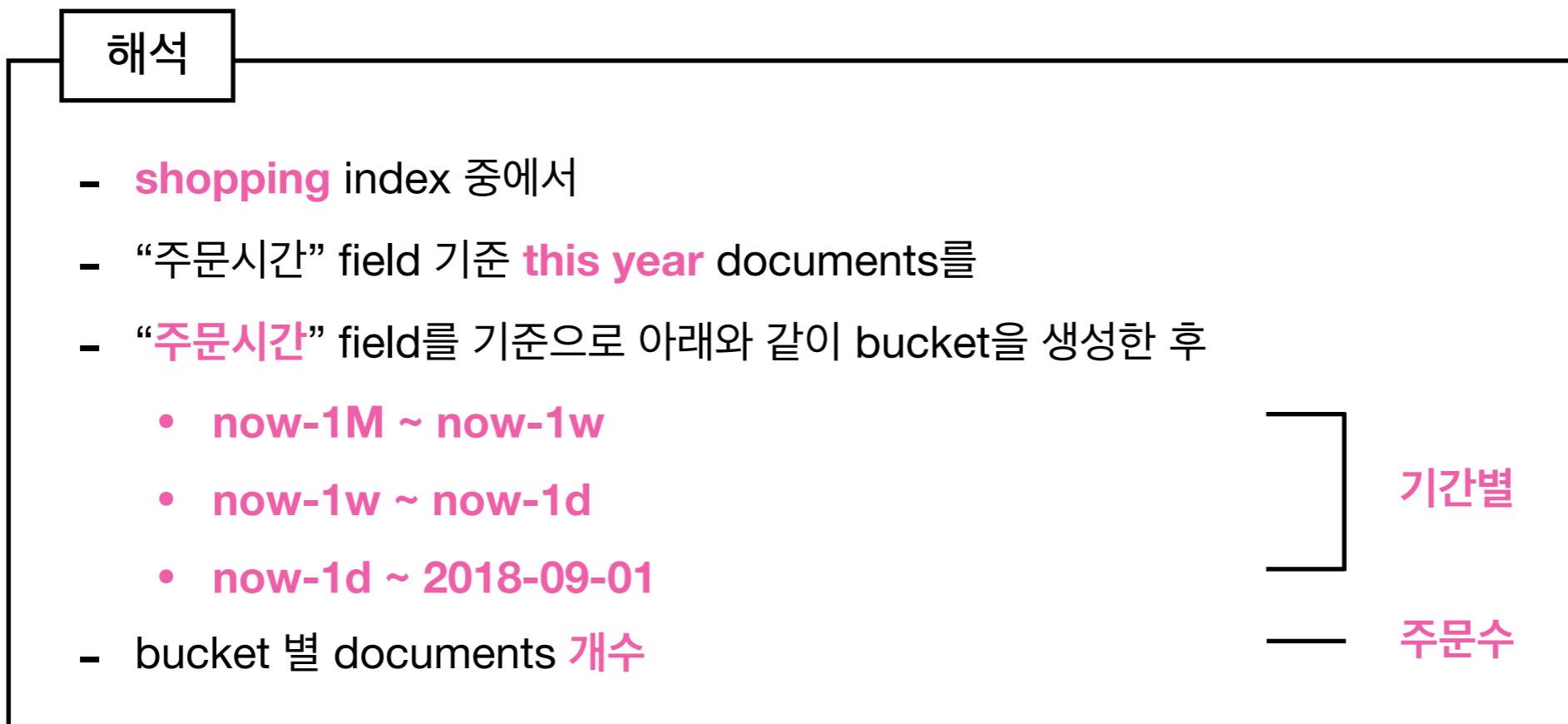
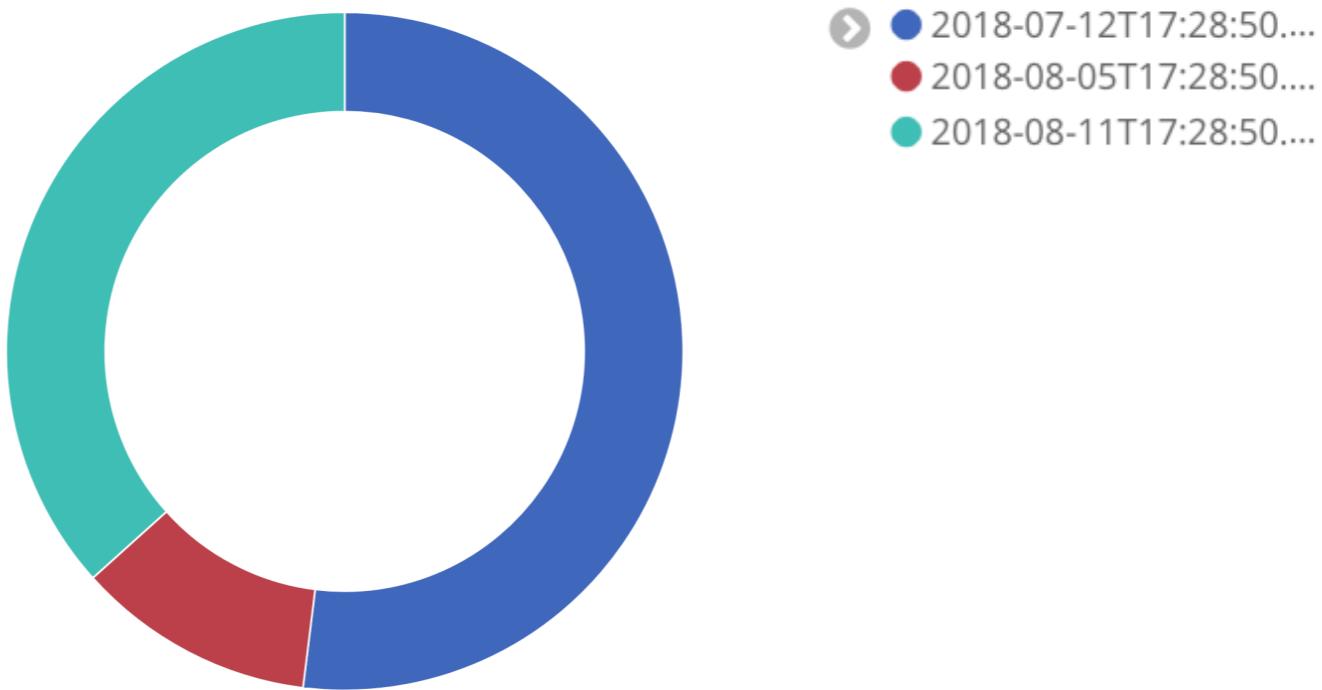
② Metrics aggregation은 고정

③ Date Histogram aggregation 선택

④ Date Histogram Aggregation  
적용할 Field 선택

⑤ Date Histogram 간격 설정

## Pie Chart Object



## Pie Chart Configuration - Split Slices (Date Range)

①

(Pie Chart 선택 후)  
shopping index 선택

The screenshot shows the Kibana configuration interface for a pie chart. At the top, the index name 'shopping' is selected. Below it, the 'Data' tab is active. A button labeled '⑥ 실행' (Run) is highlighted with a blue border. In the 'Metrics' section, 'Slice Size' is selected. Under 'Buckets', 'Split Slices' is chosen. In the 'Aggregation' section, 'Date Range' is selected. The 'Field' dropdown is set to '주문시간'. Below the field dropdown, there are three rows of date range inputs. Each row consists of two input fields: 'From' and 'To'. The first row has 'From: now-1M' and 'To: now-1w'. The second row has 'From: now-1w' and 'To: now-1d'. The third row has 'From: now-1d' and 'To: 2018-09-01'. Each row has a red 'X' icon to its right.

① Time Range를 This year로 설정

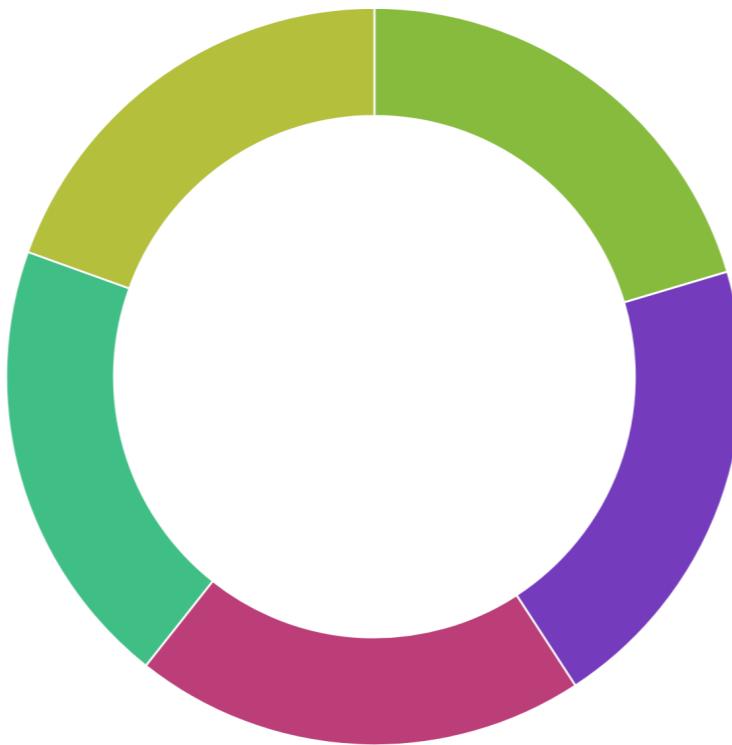
② Metrics aggregation은 고정

③ Date Range aggregation 선택

④ Date Range aggregation 적용 Field 선택

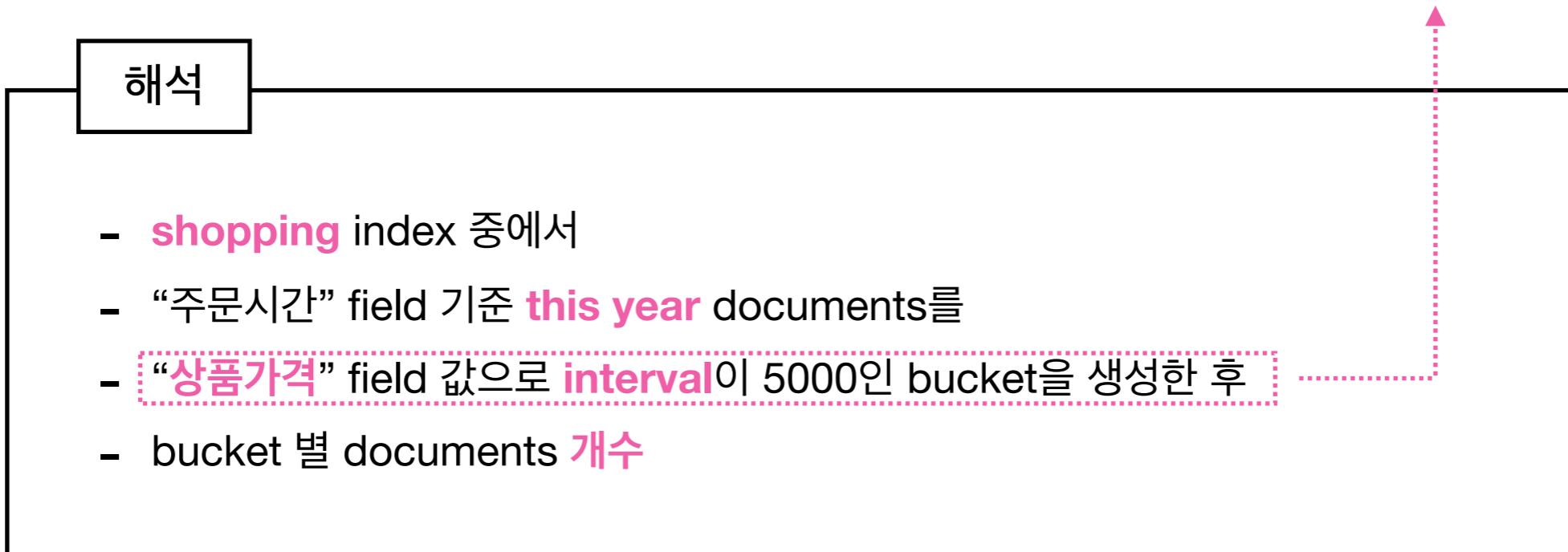
⑤ Bucket 별 Date Range 직접 입력

## Pie Chart Object



- 5,000
- 10,000
- 15,000
- 20,000
- 25,000

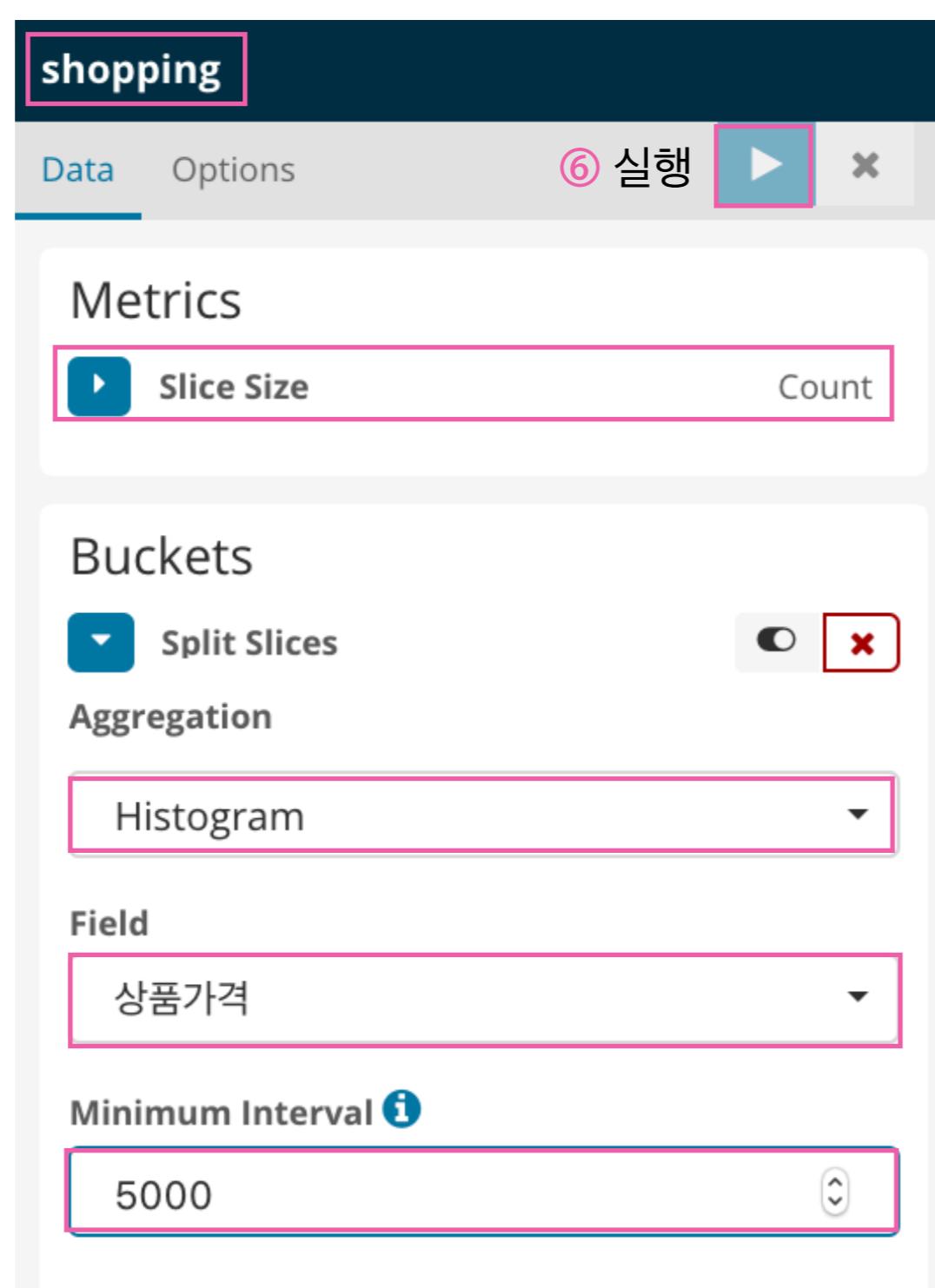
bucket	의미 ( $x = \text{상품가격}$ )
5,000	$0 \leq x < 5,000$
10,000	$5,000 \leq x < 10,000$
15,000	$15,000 \leq x < 20,000$
20,000	$20,000 \leq x < 25,000$
25,000	$25,000 \leq x < 30,000$



## Pie Chart Configuration - Split Slices (Histogram)

①

(Pie Chart 선택 후)  
shopping index 선택



① Time Range를 This year로 설정

⑥ 실행

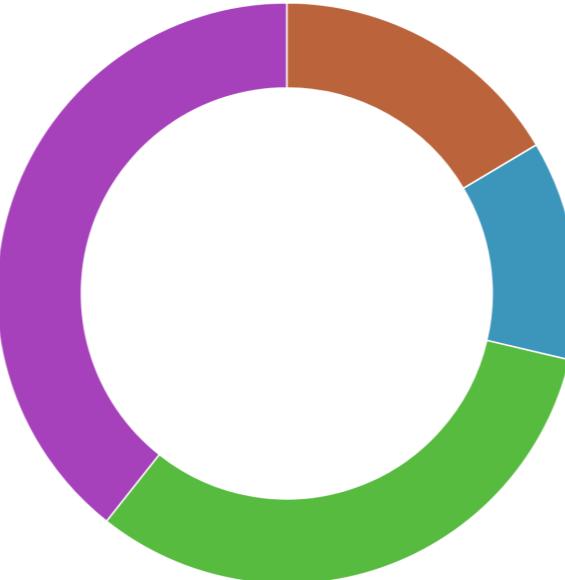
② Metrics aggregation은 고정

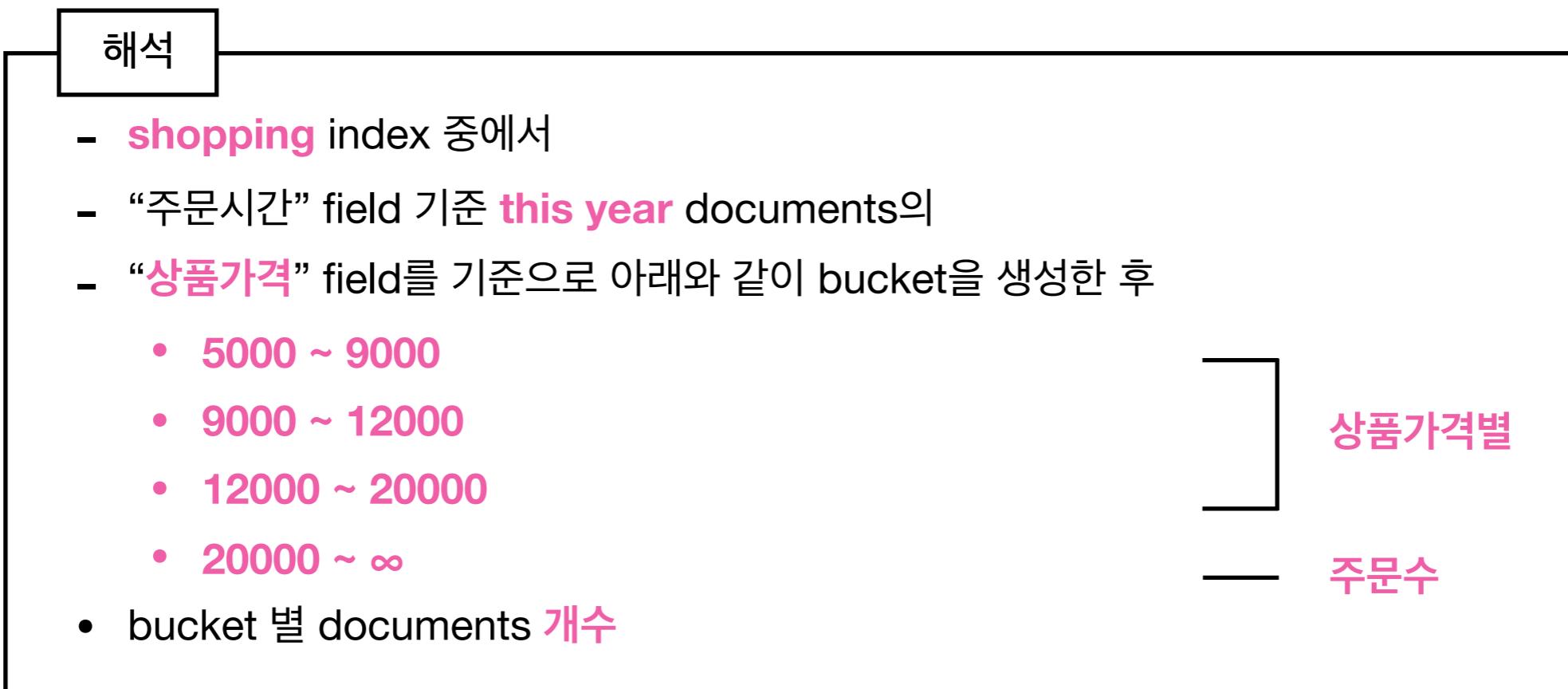
③ Histogram aggregation 선택

④ Histogram aggregation 적용할 Field 선택

⑤ Histogram 간격 설정

## Pie Chart Object

- 
- ▶ 5,000 to 9,000
  - ▶ 9,000 to 12,000
  - ▶ 12,000 to 20,000
  - ▶ 20,000 to  $+\infty$



## Pie Chart Configuration - Split Slices (Range)

①

(Pie Chart 선택 후)  
shopping index 선택

Metrics

Slice Size Count

Buckets

Split Slices

Aggregation

Range

Field

상품가격

From	To
0	5000
5000	9000
9000	12000
12000	20000
20000	

◀ This year ➡

① Time Range를 This year로 설정

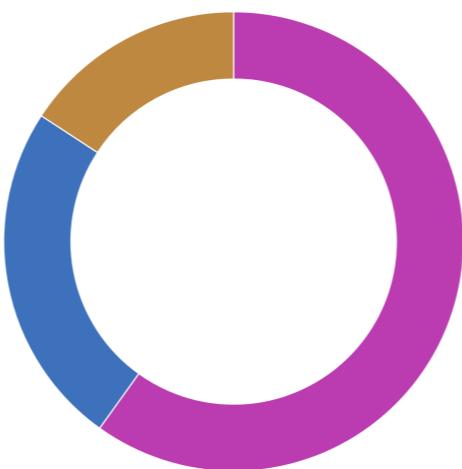
② Metrics aggregation은 고정

③ Range aggregation 선택

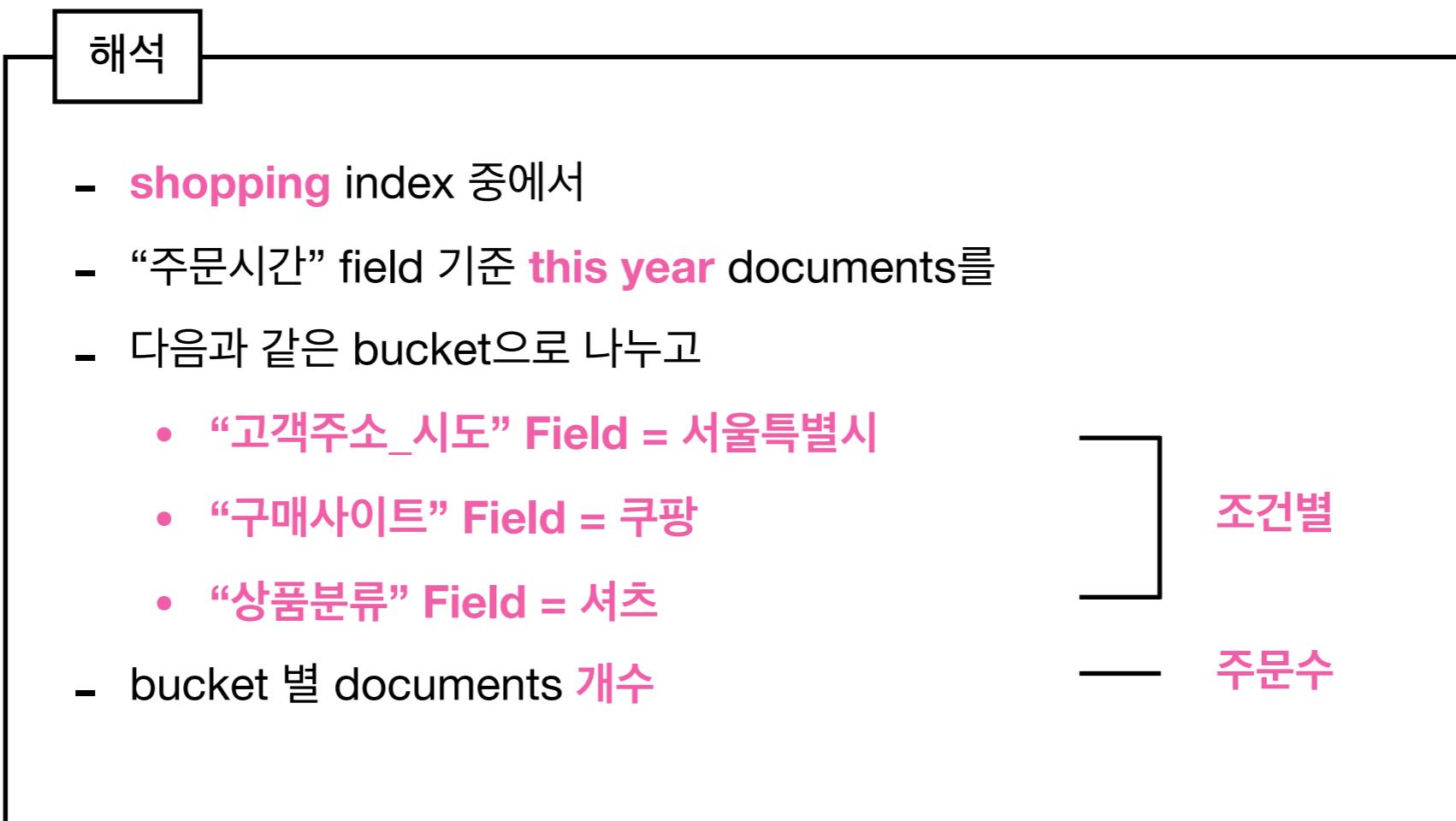
④ Range aggregation 적용할 Field 선택

⑤ bucket 별 간격 설정

## Pie Chart Object



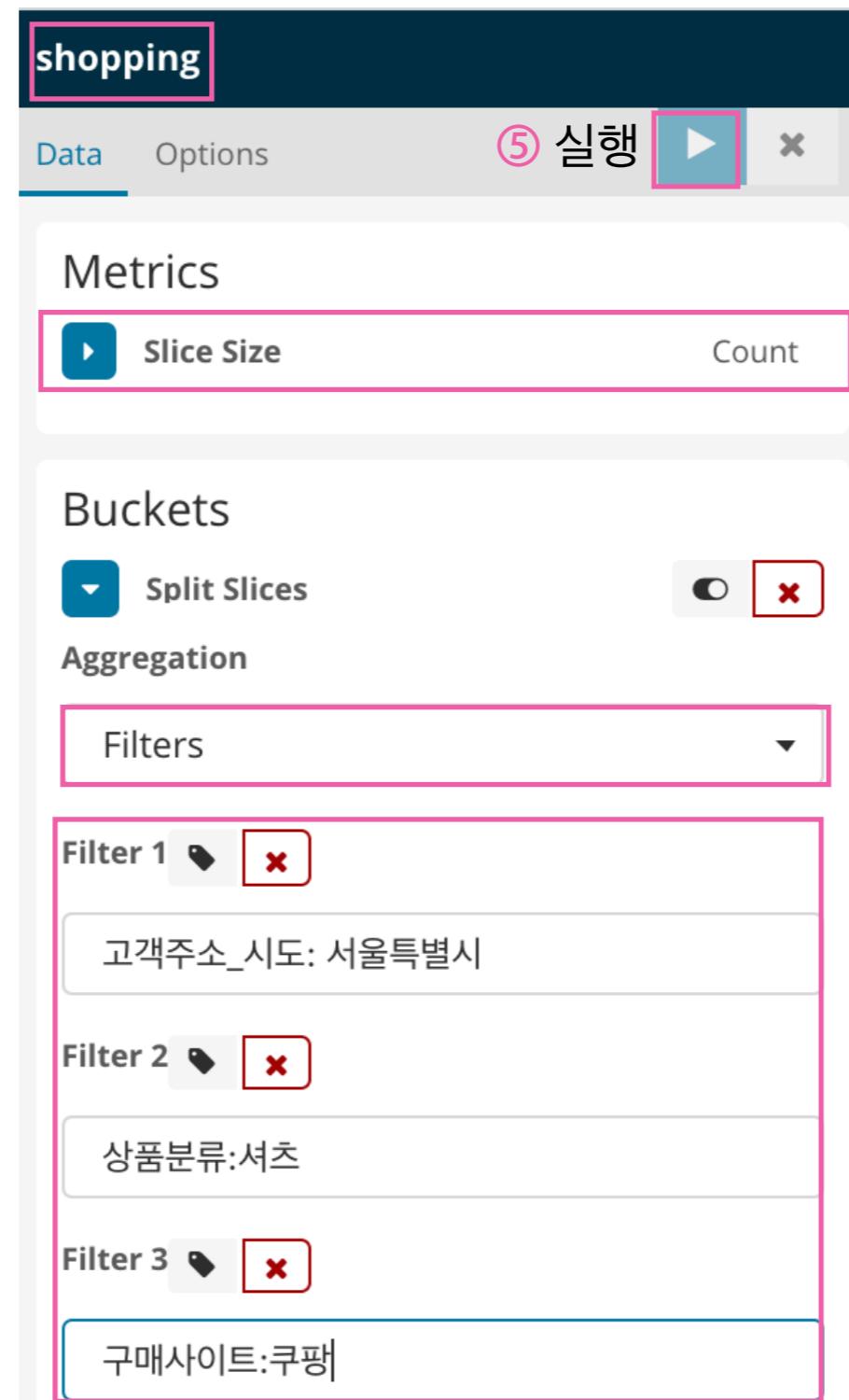
- ▶ ● 고객주소\_시도: 서울특별시
- 구매사이트: 쿠팡
- 상품분류: 셀프



## Pie Chart Configuration - Split Slices (Filter)

①

(Pie Chart 선택 후)  
shopping index 선택



< ⏪ This year ⏩ >

① Time Range를 This year로 설정

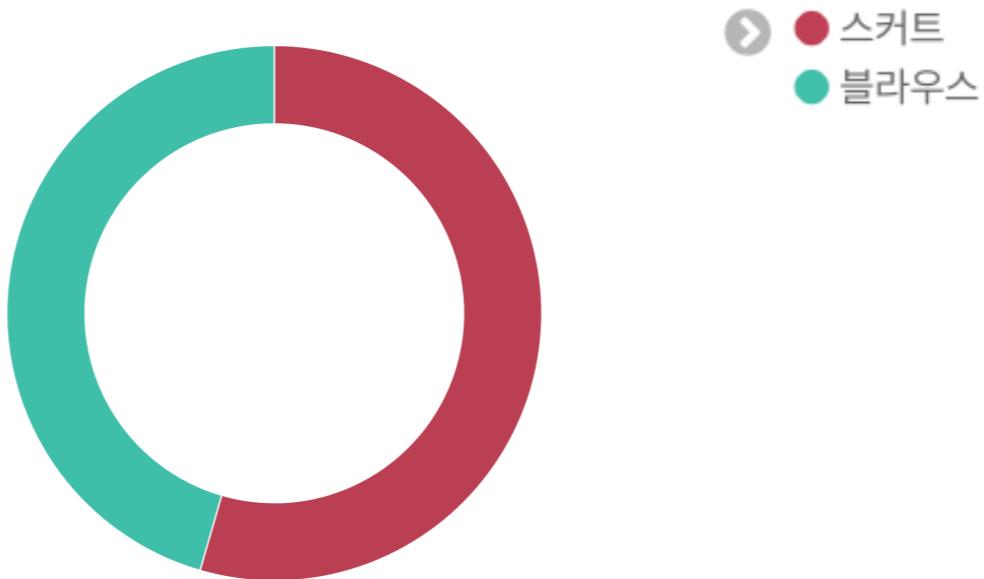
② Metrics aggregation은 고정

③ Filters aggregation 선택

④ Filter를 작성하여 Bucket 생성

(문법은 나중에 배웁니다)

## Pie Chart Object



해석

- shopping index 중에서
- “주문시간” field 기준 this year의
- 모든 documents 대비 다음 조건을 만족하는 특정 documents에서
  - “고객주소\_시도” Field = 서울특별시
  - $20 \leq$  “고객나이” Field  $\leq 35$
  - “고객성별” Field = 여성
- “interesting or unusual”한 “상품분류” field 2개로 bucket을 생성 후의
- bucket 별 documents 개수

## Pie Chart Configuration - Split Slices (Significant Terms)

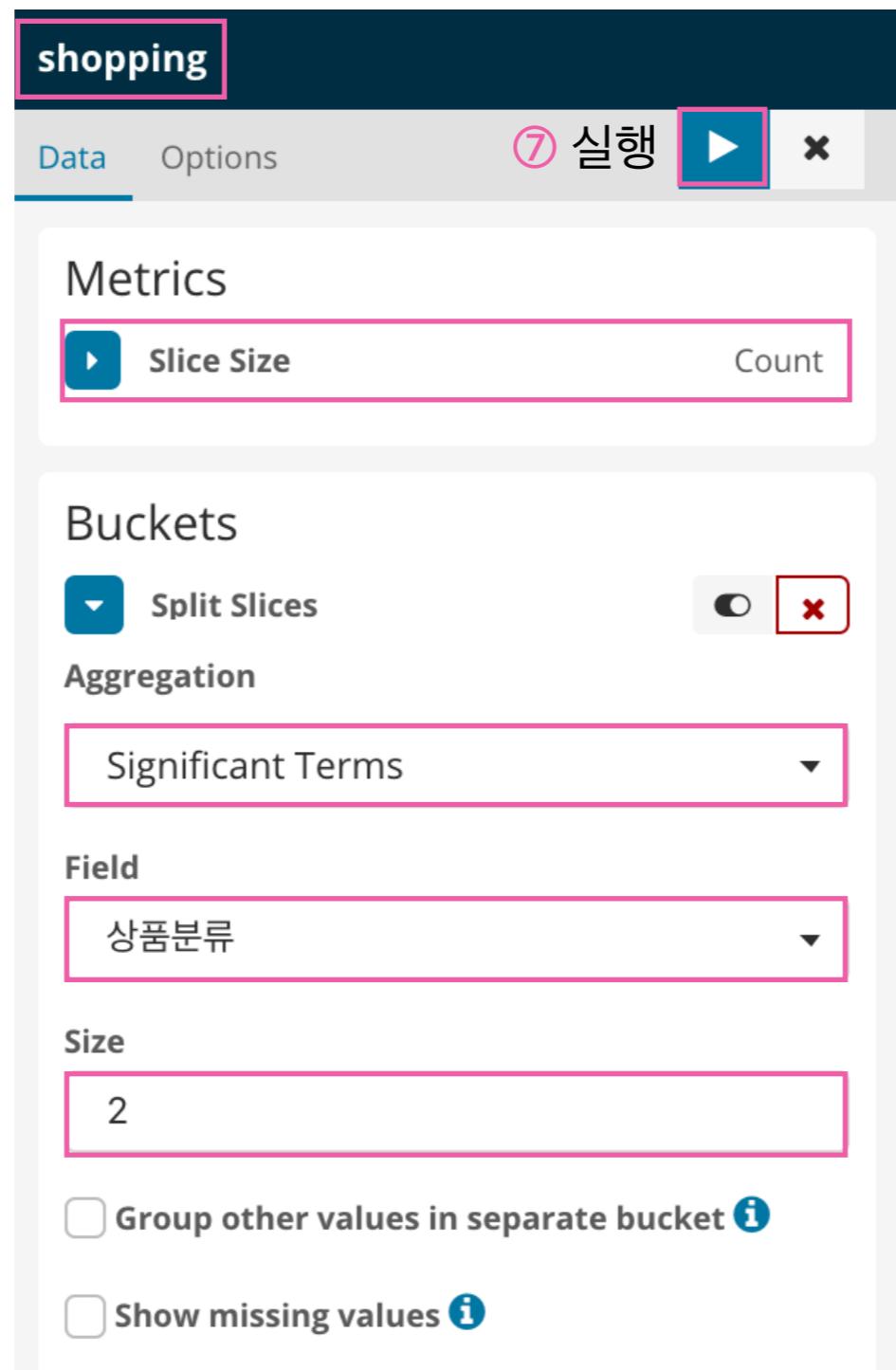
⑧ Search bar에 입력

고객주소\_시도:서울특별시 AND 고객나이:[20 TO 35] AND 고객성별:여성

⑨ 클릭

⑩

(Pie Chart 선택 후) shopping index 선택



① Time Range를 This year로 설정

② Metrics aggregation은 고정

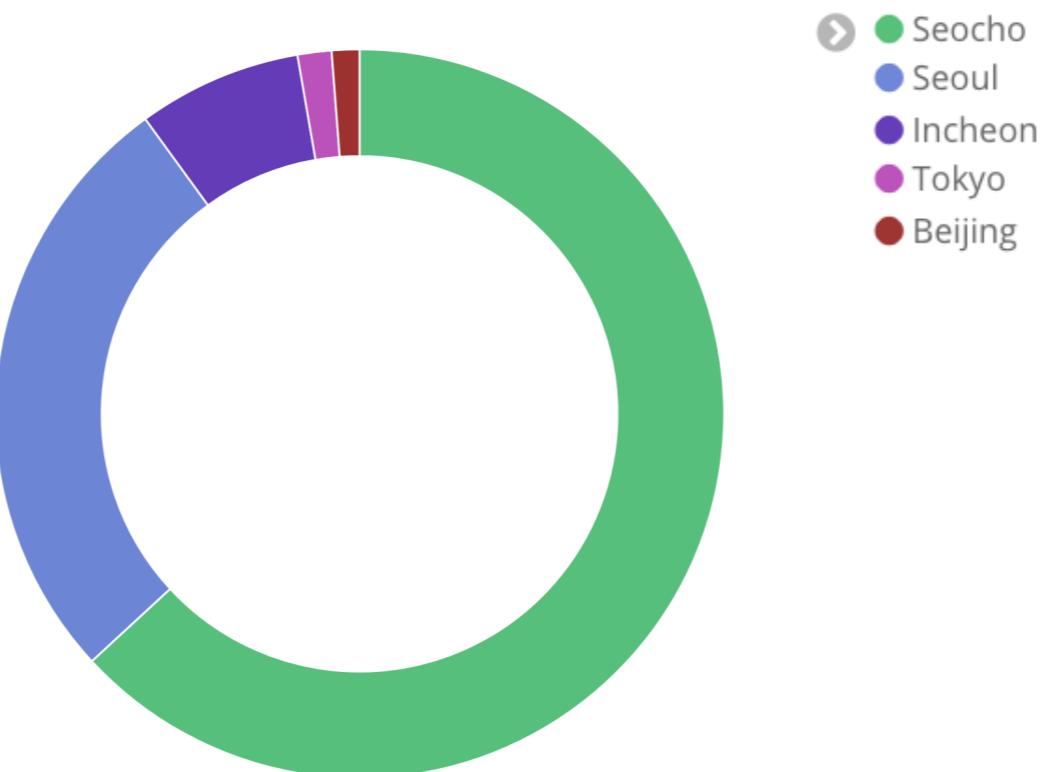
③ Significant Terms aggregation 선택

④

Significant Terms aggregation  
적용할 Field 선택

⑤ 생성할 Bucket 개수 설정

## 예제 6) Tag Cloud



조건

- nginx-\* index 중에서
- "@timestamp" field 기준 “2018년 6월 1일 ~ 2018년 8월 12일” documents의
- documents 개수가 가장 많았던 “nginx.access.geoip.city\_name” field 5개의 ————— 도시별
- documents 개수 ————— 주문수

이번에는 Split Charts 후에 Split Slices를 적용하자



그전에 이게 왜 필요할까?

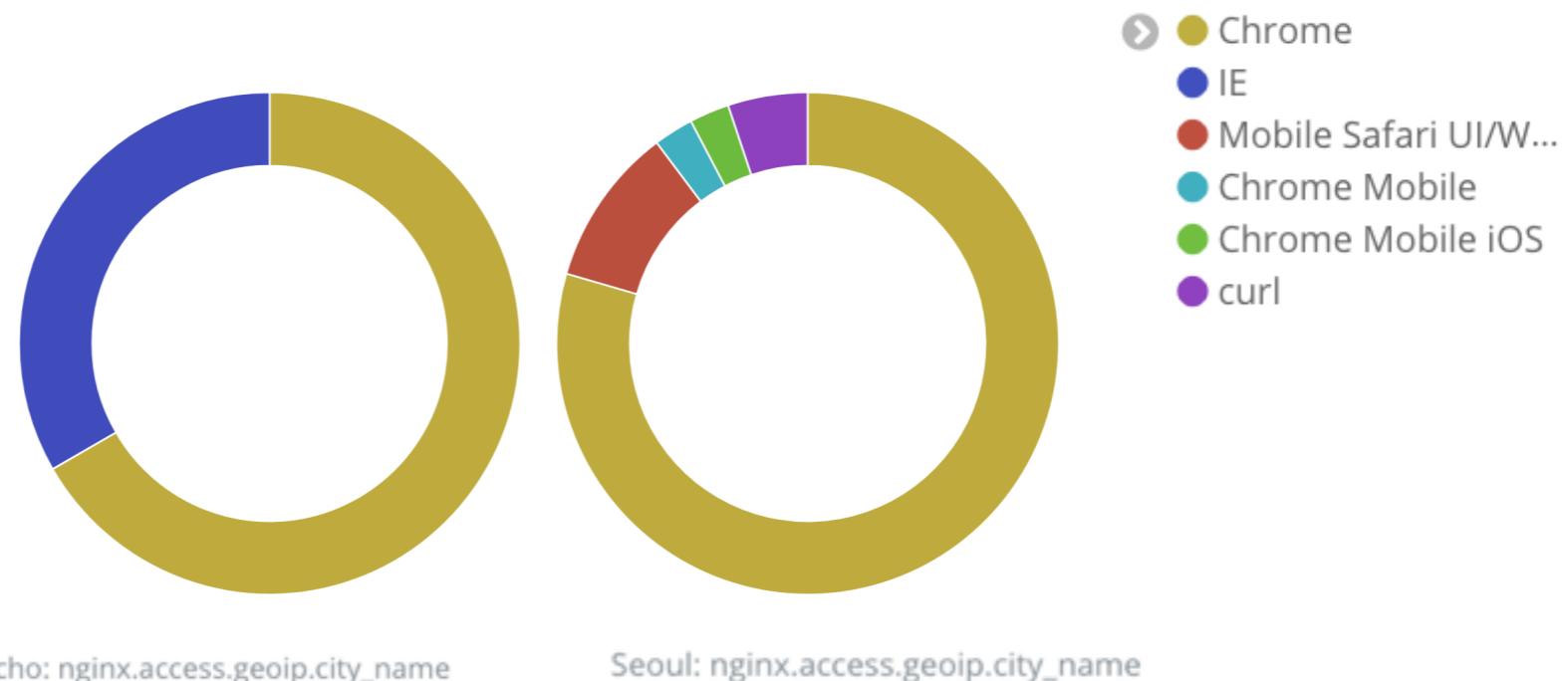


지금까지는 **하나의** Field를 기준으로 시각화 했다



하지만 현실 세계의 문제는 그리 간단하지 않다면?

## Pie Chart Object

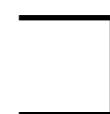


해석

- nginx index 중에서
- “주문시간” field 기준 “**2018년 6월 1일 ~ 2018년 8월 12일**” documents를
- “**nginx.access.geoip.city\_name**” field 값으로 bucket을 생성한 후
- documents 개수가 **가장 많은 2개**의 bucket에 대해서 각각
- “**nginx.access.user\_agent.name**” field 값으로 sub-bucket을 생성한 후
- documents 개수가 **가장 많은 5개**의 sub-bucket 마다의
- “**nginx.access.remote\_ip**” field의 **unique한 value** 개수



도시별



user\_agent별



실질 접속자 수

## Pie Chart Configuration - Split Slices (Significant Terms)

nginx-\*

Data Options X

Metrics

Slice Size

Aggregation

Unique Count

Field

nginx.access.remote\_ip

Buckets

Split Chart X

Rows Columns

Aggregation

Terms

Field

nginx.access.geoip.city\_name

Order By

Custom Metric

Aggregation

Count

Advanced

Order Size

Descending 2

Split Slices X

Sub Aggregation

Terms

Field

nginx.access.user\_agent.name

Order By

Custom Metric

Aggregation

Count

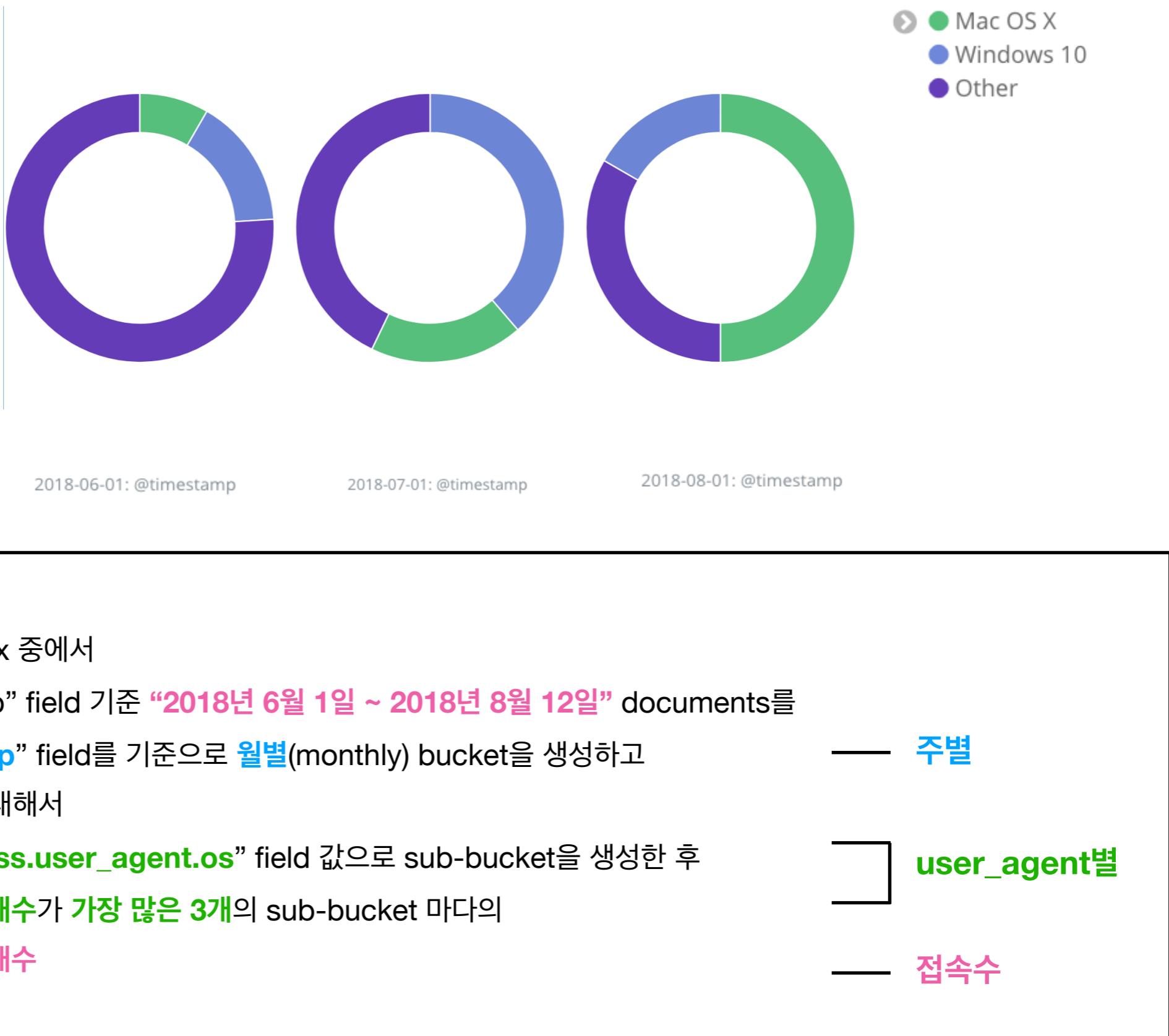
Advanced

Order Size

Descending 5

This screenshot shows the configuration for a pie chart titled "nginx-\*". The "Data" tab is selected. In the "Metrics" section, "Slice Size" is set to "Unique Count" for the field "nginx.access.remote\_ip". The "Buckets" section contains a main "Split Chart" bucket with "Rows" selected. This bucket has an "Aggregation" of "Terms" for the field "nginx.access.geoip.city\_name", ordered by "Custom Metric". It also includes a "Count" sub-aggregation. Below this, there are two "Split Slices" buckets, each with "Size" set to 2 and 5 respectively. Each slice bucket has its own "Sub Aggregation" with "Terms" for the field "nginx.access.user\_agent.name", ordered by "Custom Metric", and a "Count" sub-aggregation.

## 예제 7) Tag Cloud



## 마치기 전에

- Elastic Stack이 무엇을 의미하며 어떤 용도로 쓰이는지 이해한다
- Elasticsearch의 기본 용어를 이해한다
- Kibana의 작업 흐름을 이해한다
  - Kibana에서 Elasticsearch Index를 등록하는 방법을 안다
  - Kibana에서 Discover Page를 이용하는 방법을 안다
  - Kibana에서 Visualize를 하는 큰 흐름을 이해한다
  - 어느 상황에서 어느 Aggregation (Metric & Bucket)을 사용할지 이해한다

**질문 및 Feedback은**

**gshock94@gmail.com로 주세요**