

# Elastic Stack 을 활용한 Data Dashboard 만들기

Week 1 - Data를 시각화해보자



kibana

Fast Campus

내용	페이지
강의소개	3
Elastic Stack 소개	7
Elasticsearch	
특징	17
용어 정리	23
Elastic Stack Workflow	30
Kibana 소개	33
Index 등록	40
데이터 탐색	52
Visualize 맛보기	76
Aggregation	
Bucket Aggregation	84
Metric Aggregation	90
Visualize 안내	99
Visualize 실전	
Markdown	120
Metric	123
Coordinate Map	144
Region Map	148
Tag Cloud	153
Pie Chart	158

강의가 끝나면

data가 주어지면 dashboard를 구축하고 needs에 맞게 운영할 수 있다.

그러므로

- 모든 기능 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.

## FAQ

자주 물어보는 질문 정리 

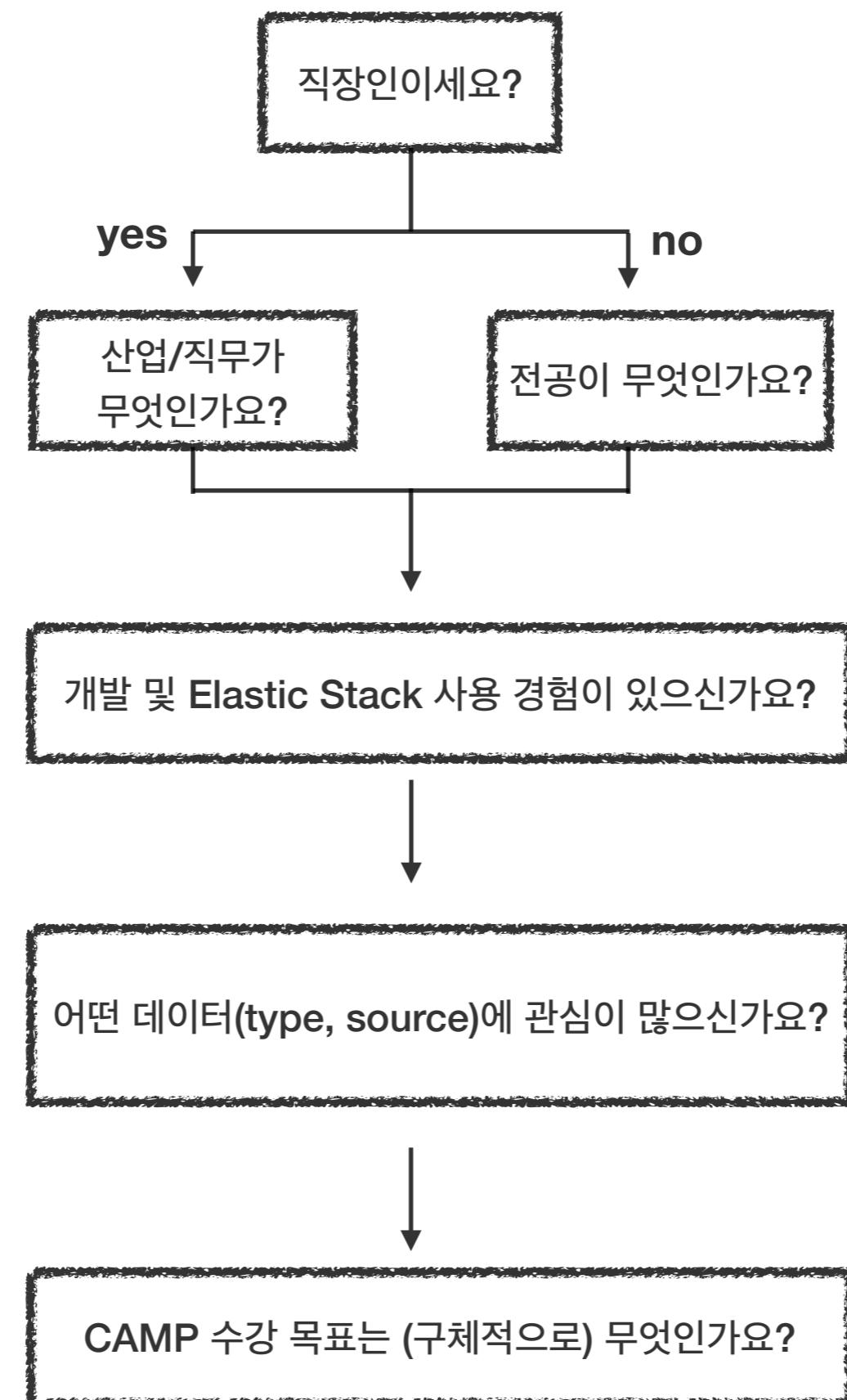
## Wiki

Elastic Stack 간단한 사용법 정리 

## Questions

- Kakao Open Chat (#elastic4)
- [패스트캠퍼스] Elastic Stack을 활용한 Data Dashboard 만들기 CAMP 
- Elastic Stack and Product Documentation 
- Discuss the Elastic Stack 
- Facebook Elasticsearch Korea Group 
- Stack Overflow 

## Online Sources



개요

ELK Stack?  
Elastic Stack?  
Elasticsearch?  
Elastic?

ELKB Stack?

Elastic Stack이 무엇인지 간략히 살펴보자

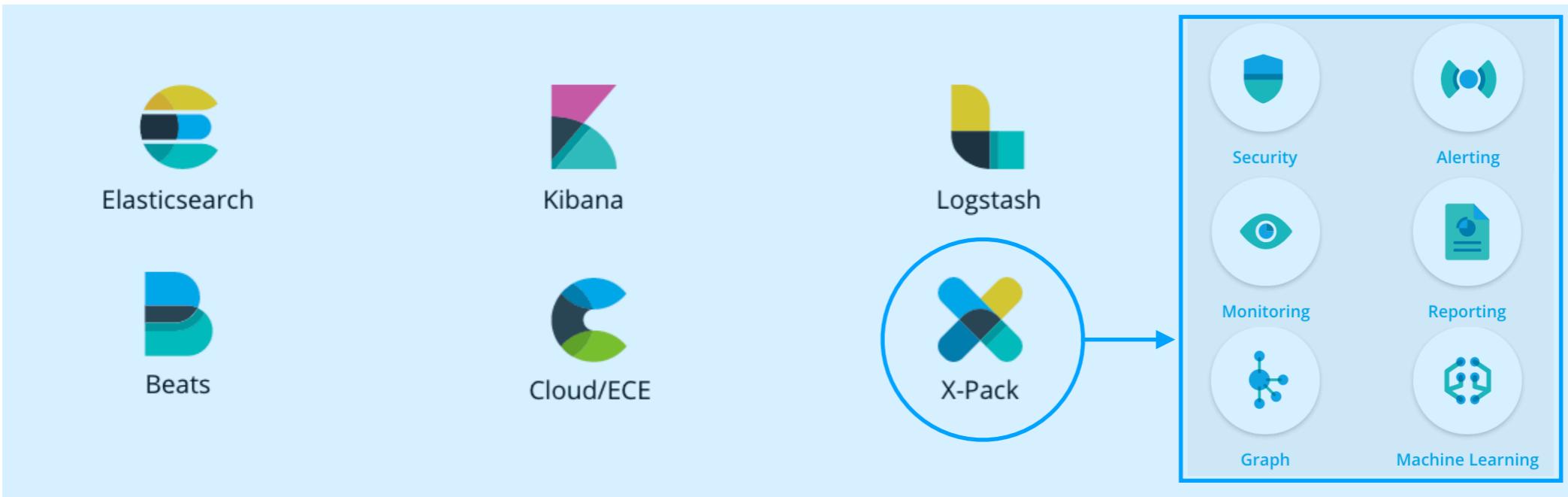
# Elastic Stack

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>	

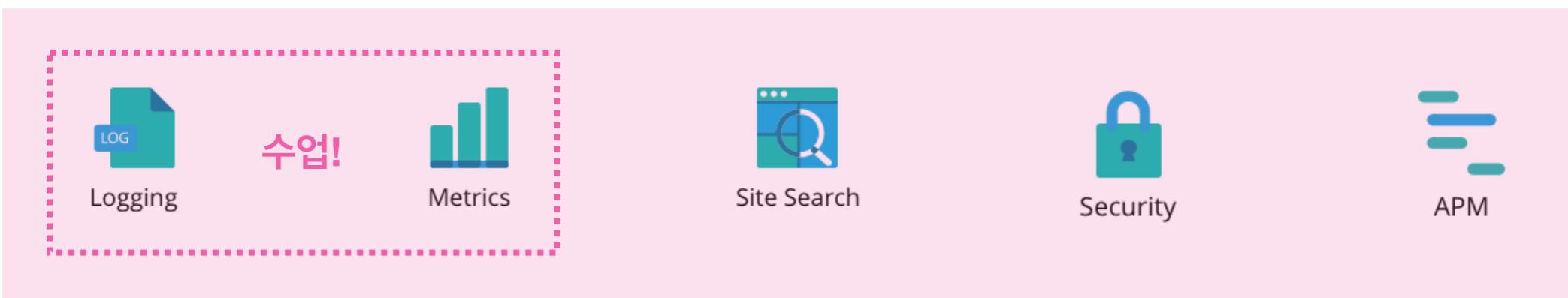
Elastic Stack으로 무얼 할 수 있을까?

= Elastic Stack을 왜 배울까?

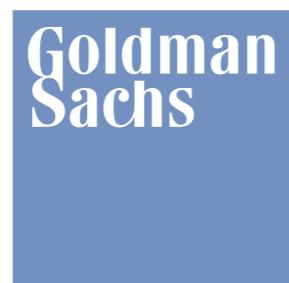
## Products



## Solutions



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



# NAVER



WIKIPEDIA  
The Free Encyclopedia



# BBC



# 해결하려는 문제만큼 Elastic Stack을 어떻게 사용하는지도 회사마다 다양하다



## Event prediction and forecasting

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

## Engineering Standards

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

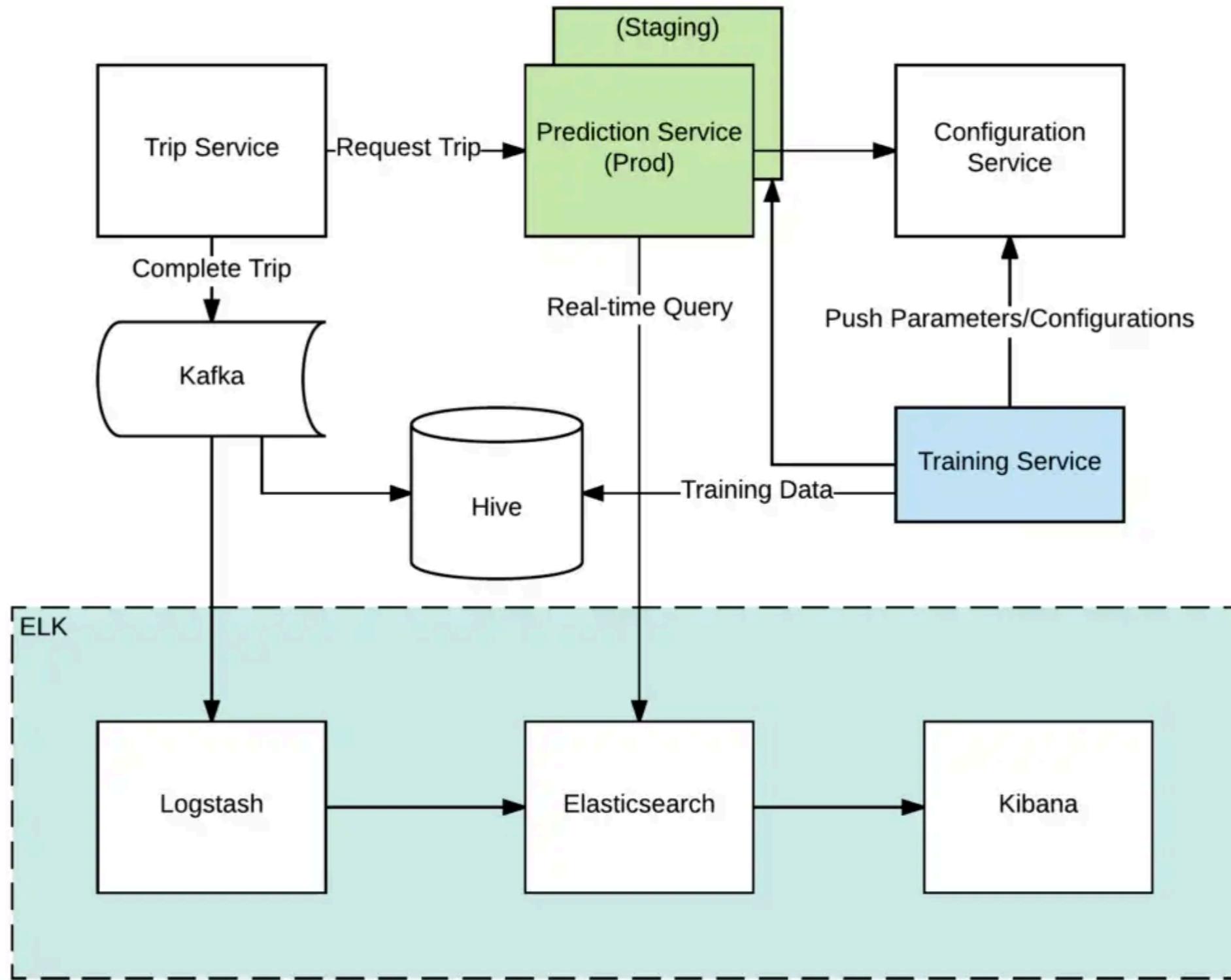
## Algorithm (k-nearest neighbors algorithm, KNN)

- 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.

# System Architecture



# Content Warning

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

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

## (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"
        }
      }
    }
  }
}
```

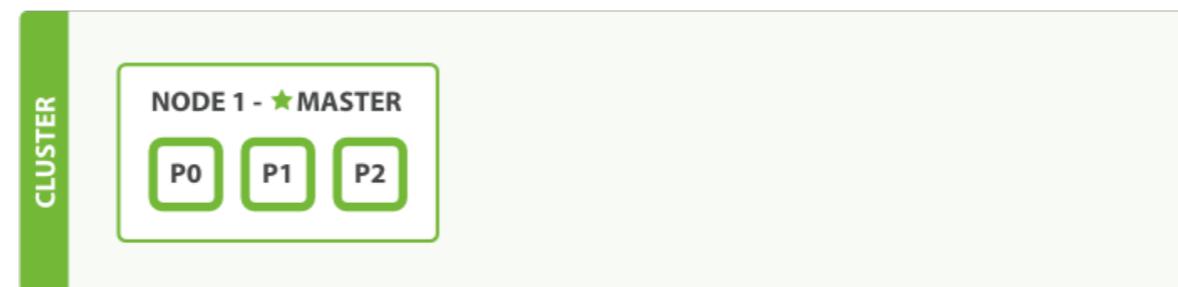


이처럼 강제로 설정하지 않는 이상 기본적으로 모든 field를 Indexing한다

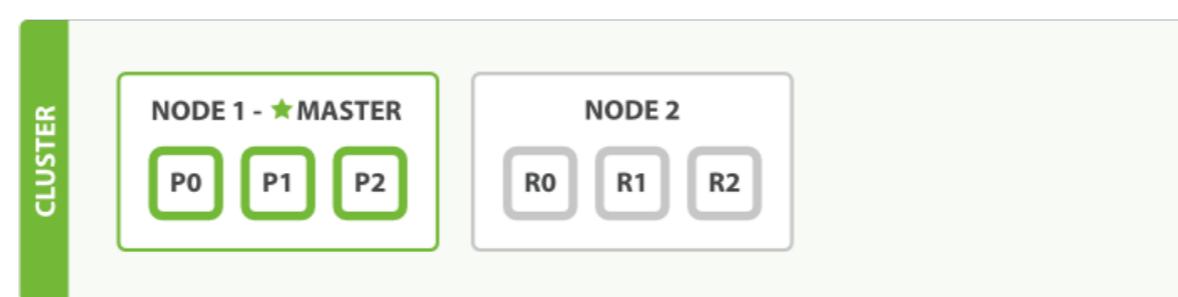
# (Horizontal) Scalability

운영 중인 elasticsearch cluster에 간단한 설정을 통해 elasticsearch node 추가 가능

심화



Node (≒ 서버) 1개 추가



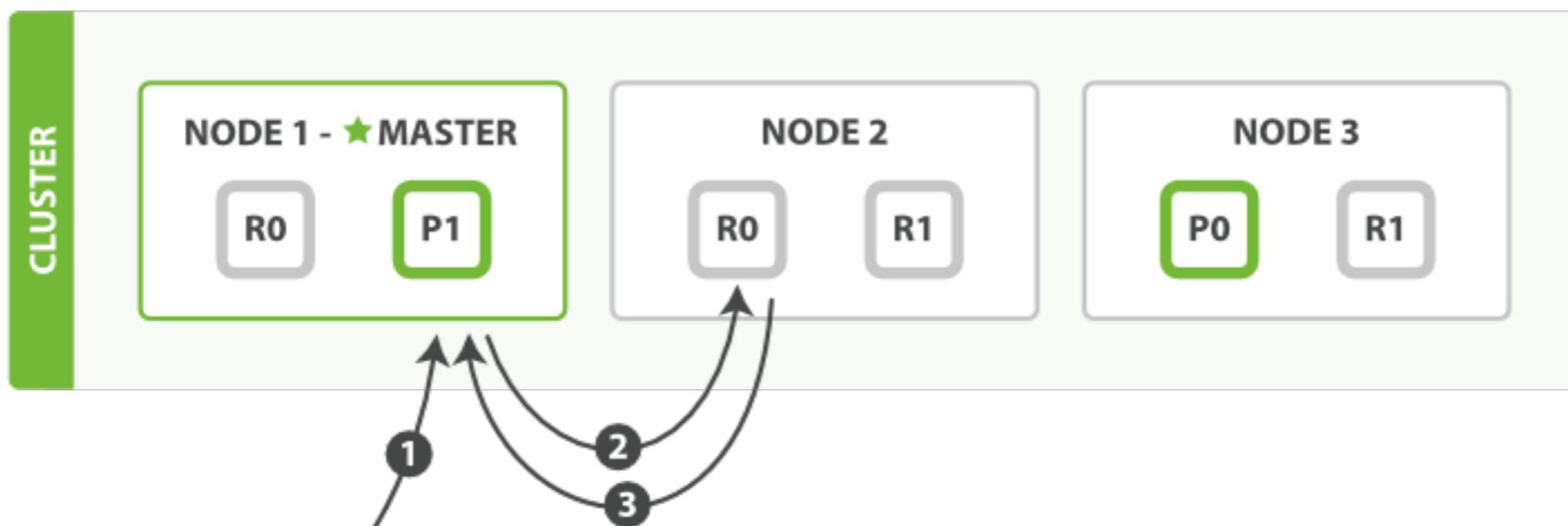
Node (≒ 서버) 1개 추가



# Distributed Operations

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

심화

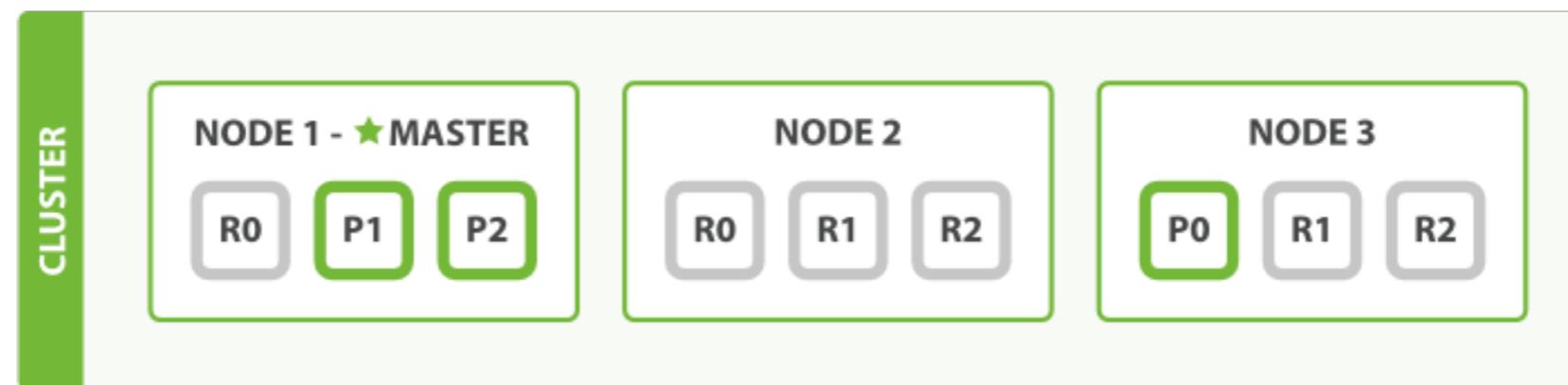


- ① User가 NODE1에 검색 request 전송
- ② NODE1은 NODE2에 request 전달
- ③ NODE2는 request 처리 후 NODE1에 결과 전송
- ④ NODE1는 User에게 결과 전송

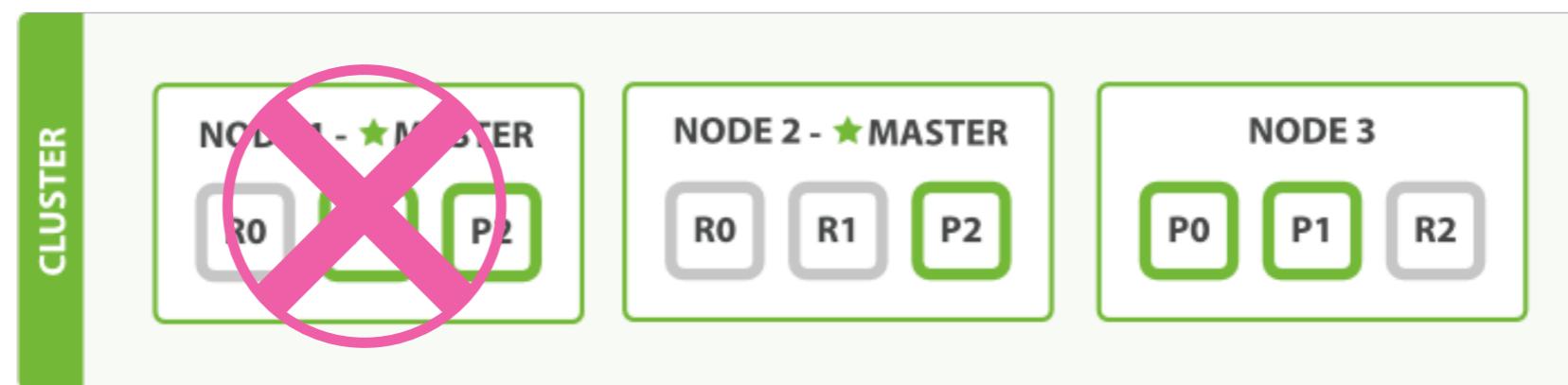
# Fault Tolerance

(Replica Shards를 설정을 통해) 특정 Node가 다운되어도 데이터 유실 없이 운영할 수 있다

심화



Node1이 다운되어도 Node2 & Node3 데이터로 백업 가능



용어 정리

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

- 위의 비교는 어디까지나 이해를 돋기 위한 목적일 뿐 정확히 일치하지는 않는다
- 6.0.0 이후에는 Index 1개에 Type 1개가 되어 사실상 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  }
```

## 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	E	F	G	H
1								
2								
3		date	product	quantity	sales			
4	01/01/2018	Onepice	2	39,000				
5	01/01/2018	Cardigan	1	37,000				
6	01/01/2018	Knit	3	69,000				
7	01/01/2018	Jeans	1	78,000				
8	01/01/2018	T-Shirt	5	89,000				
9	01/01/2018	Pants	1	55,000				
10	01/01/2018	Knit	3	69,000				
11	01/01/2018	Jeans	1	78,000				
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								
34								
35								
36								
37								
38								
39								
40								
41								
42								
43								
44								
45								
46								
47								
48								
49								
50								
51								
52								
53								
54								
55								
56								
57								
58								
59								
60								
61								
62								
63								
64								
65								
66								
67								
68								
69								
70								
71								
72								
73								
74								
75								
76								
77								
78								
79								
80								
81								
82								
83								
84								
85								
86								
87								
88								
89								
90								
91								
92								
93								
94								
95								
96								
97								
98								
99								
100								

## 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

	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

### 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		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

### Schema

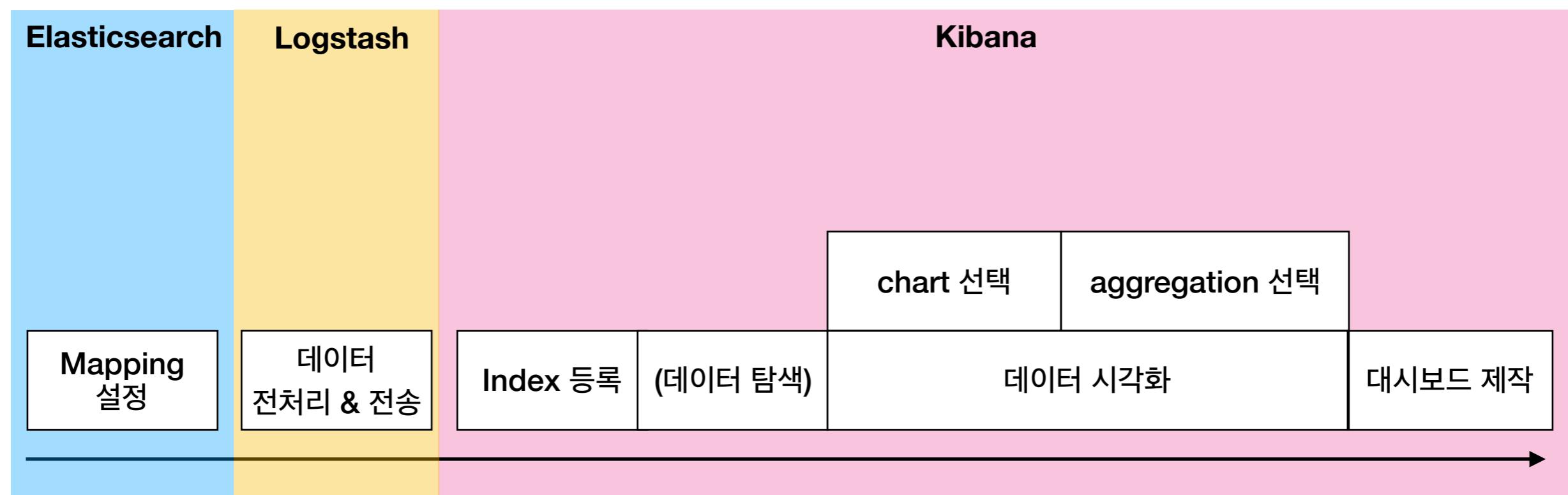
```
mysql> CREATE TABLE Sheet1 (
    -> date DATE,
    -> product VARCHAR(32),
    -> quantity INT(100),
    -> sales INT(100)
    -> );
```

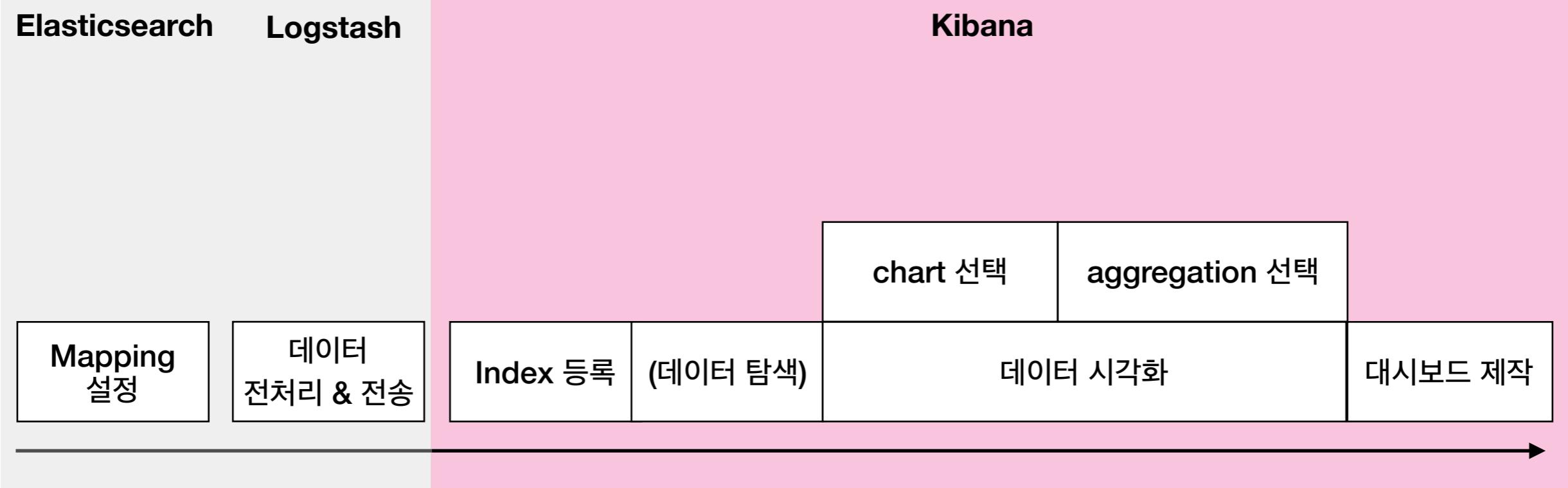
## Elasticsearch

### Mapping

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

## Elastic Stack Workflow

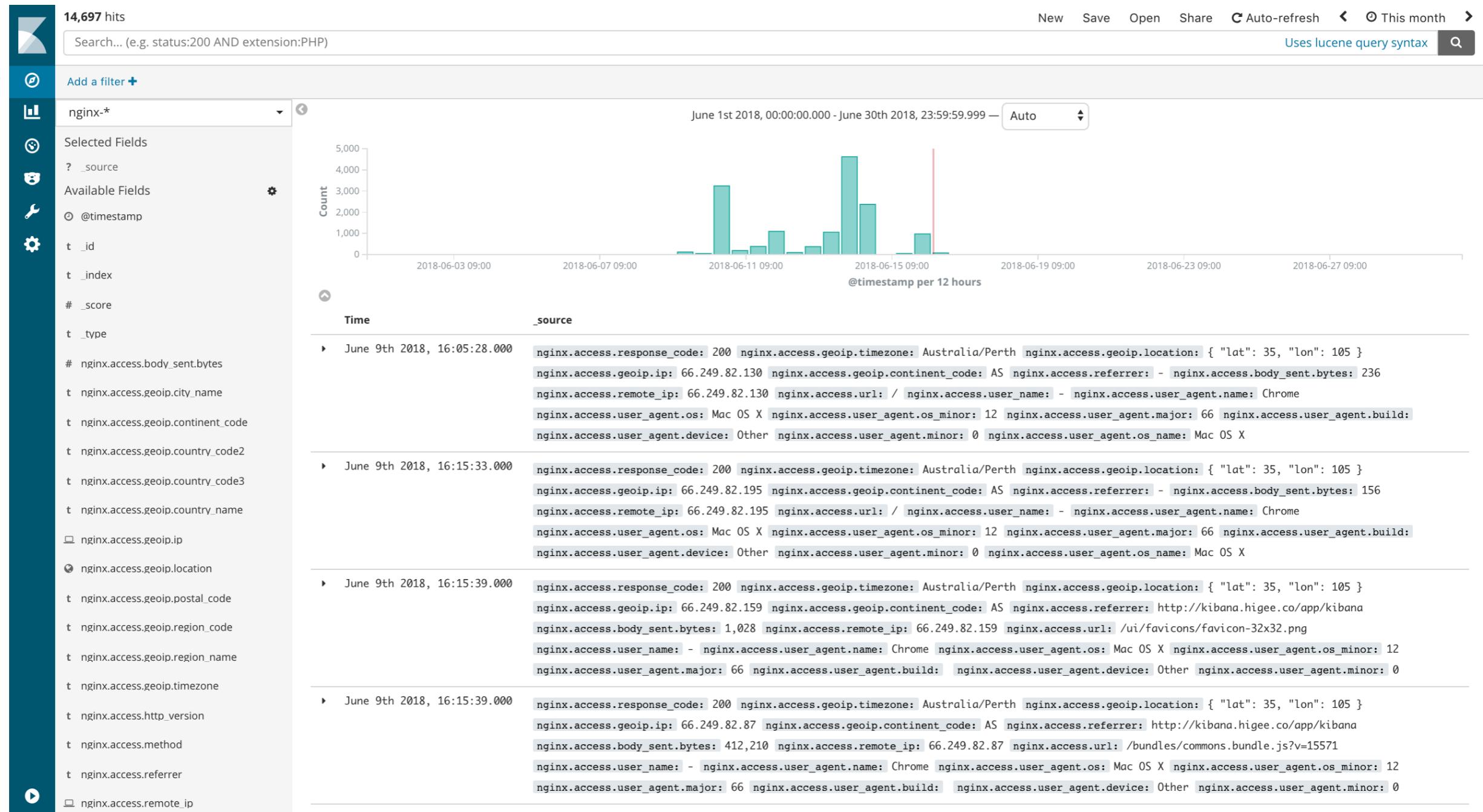




## Kibana 화면 소개

# Discover

## 데이터 검색 및 필터 등을 이용한 간단한 EDA 작업



# Visualize

## Dashboard에 배치할 Visualization 생성



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

Dashboard / nginx

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

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

Add a filter +

[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

1-50 of 14,751 < >

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

[nginx] tag-cloud

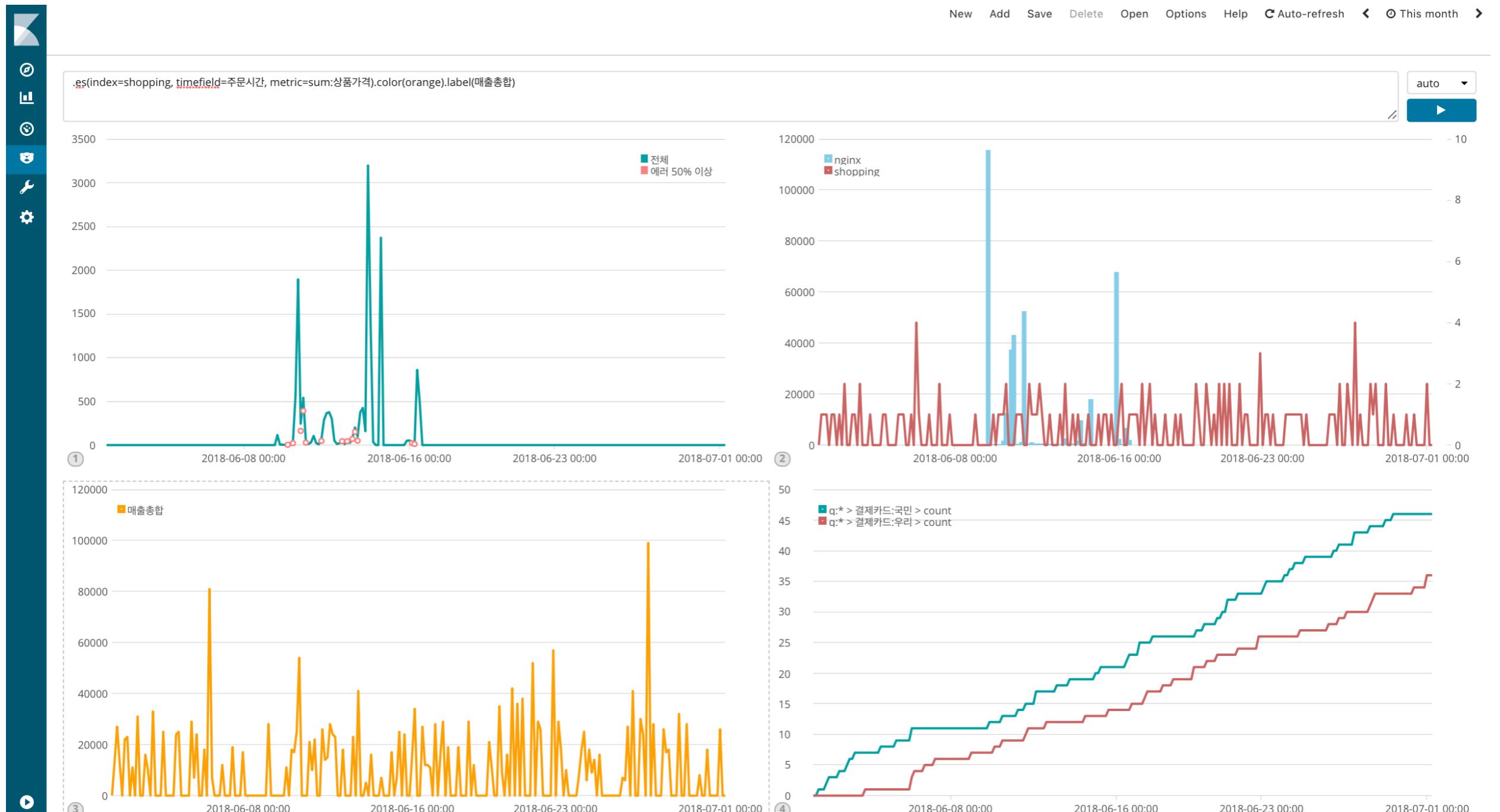
Mac OS X  
Windows 10 Other iOS  
Windows 7

[nginx] metric

14,754 req

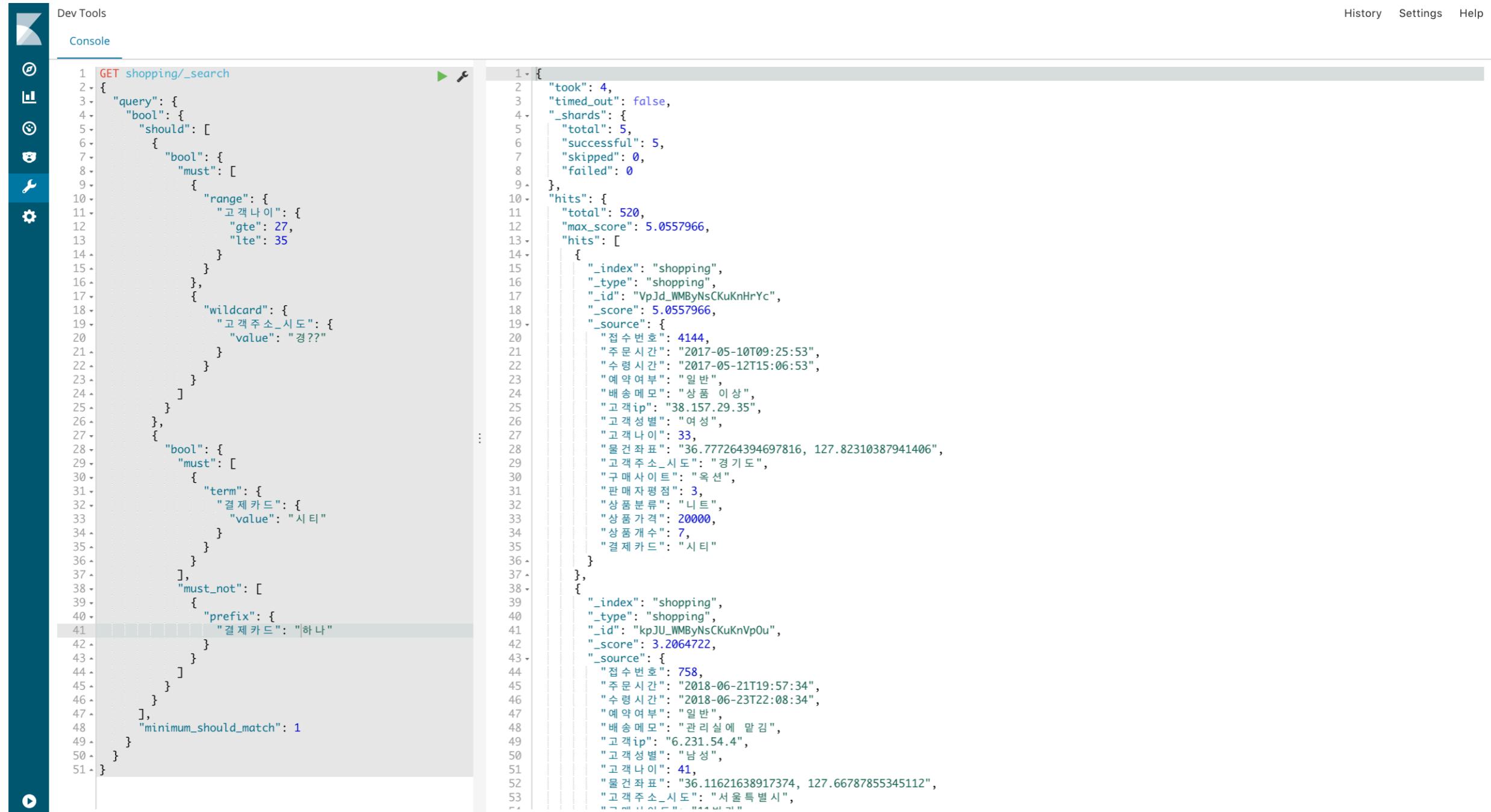
# Timelion

## Visualization의 하나인 (시계열에 특화된) Timelion 생성



# Dev Tools

## Elasticsearch REST API를 위한 UI



The screenshot shows the Elasticsearch Dev Tools interface with the 'Console' tab selected. On the left, there's a sidebar with icons for Dev Tools, History, Settings, Help, and a gear icon. A pink hand icon points to the gear icon.

The main area displays a search query in the 'Query' section and its corresponding JSON response in the 'Scripted Fields' section.

**Query:**

```
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                },
37                "must_not": [
38                  {
39                    "prefix": {
40                      "결제카드": "하나"
41                    }
42                  }
43                ]
44              }
45            }
46          ],
47        },
48        "minimum_should_match": 1
49      }
50    }
51 }
```

**Scripted Fields:**

```
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",
54          "구매사이트": "옥션"
55        }
56      }
57    ]
58  }
59 }
```

# Management

## Kibana 설정 수정

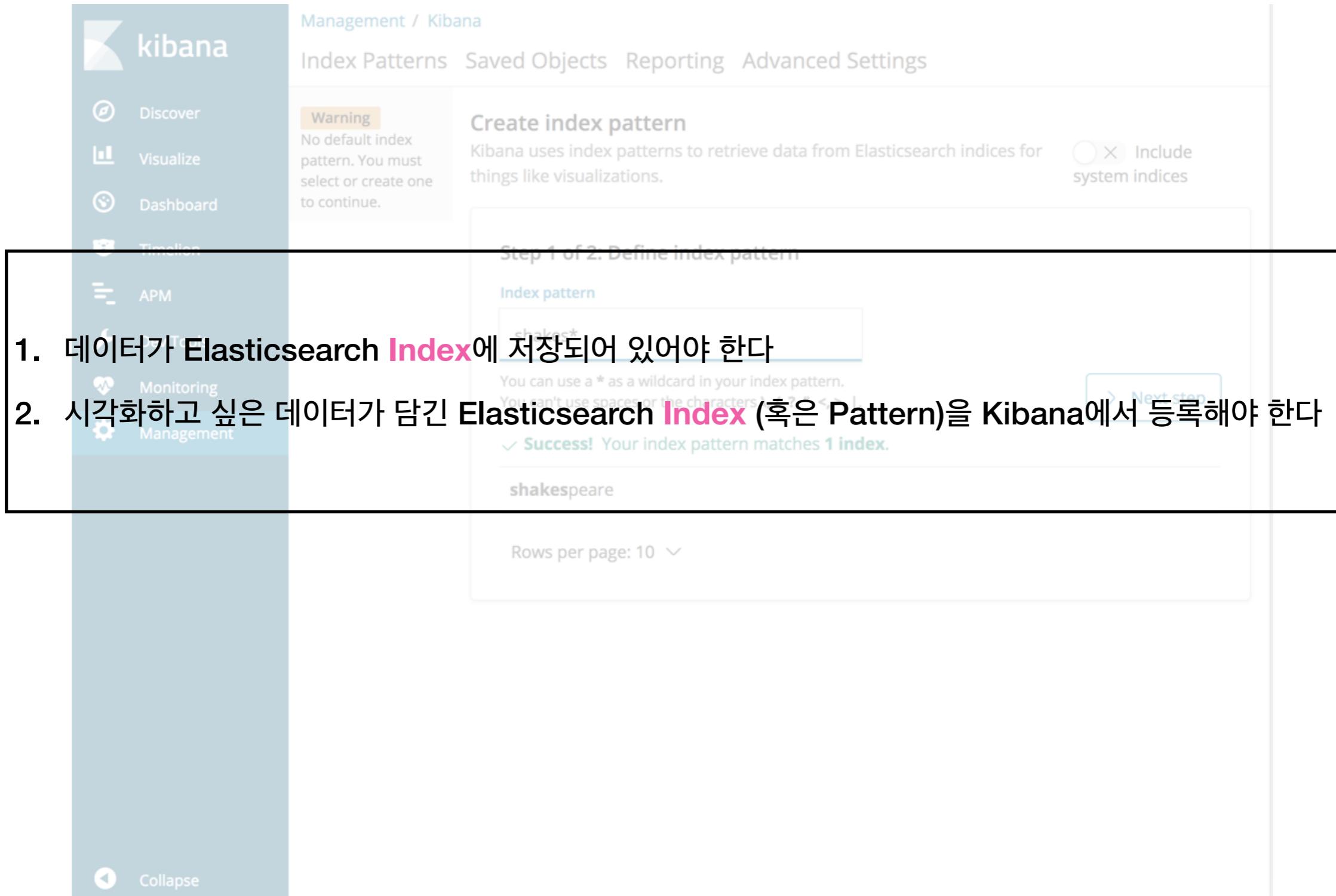
The screenshot shows the Kibana Management interface. On the left is a vertical sidebar with icons for Management, Index Patterns, Visualize, Discover, Settings, and Help. The main area is titled "Management" and shows "Version: 6.2.4". A sub-section titled "Kibana" is selected, indicated by a blue border. Below it are three tabs: "Index Patterns", "Saved Objects", and "Advanced Settings". A pink hand icon points to the "Settings" icon in the sidebar.

**Index 등록**

## Management 페이지로 이동하자

The screenshot shows the Elasticsearch Management interface. On the left is a vertical sidebar with icons for various management functions: Visualize (highlighted in pink), Discover, Settings, Index Patterns, Visualize, Advanced Settings, and a gear icon. The main area is titled "Management" and displays the version "Version: 6.2.4". A section titled "Kibana" is shown, containing a small icon and a link. Below this are three navigation links: "Index Patterns", "Saved Objects", and "Advanced Settings".

# Kibana 시작화의 전제 조건



The screenshot shows the Kibana Management interface. On the left sidebar, under the 'Management' section, there are icons for Discover, Visualize, Dashboard, Timeline, APM, Monitoring, and Management. The 'Management' section is highlighted with a black border. At the top, the navigation bar includes 'Management / Kibana', 'Index Patterns', 'Saved Objects', 'Reporting', and 'Advanced Settings'. Below the navigation, a 'Warning' message states: 'No default index pattern. You must select or create one to continue.' To the right, the main content area is titled 'Create index pattern' with the sub-instruction 'Step 1 of 2. Define index pattern'. It shows an input field containing 'shakespeare\*', with a note: 'You can use a \* as a wildcard in your index pattern. You can't use spaces or the characters ? < > | !'. Below the input field, a success message says: 'Success! Your index pattern matches 1 index.' with a checkmark icon. A 'Next step' button is visible. At the bottom of the main content area, there is a 'Rows per page: 10' dropdown.

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

# Index Patterns 등록 1단계

## 시각화 할 Elasticsearch Index (Patterns) 등록

1. Kibana 접속 -> Management 클릭 -> Index Patterns 클릭 -> Create Index Pattern 클릭
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**
- 2'. 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.

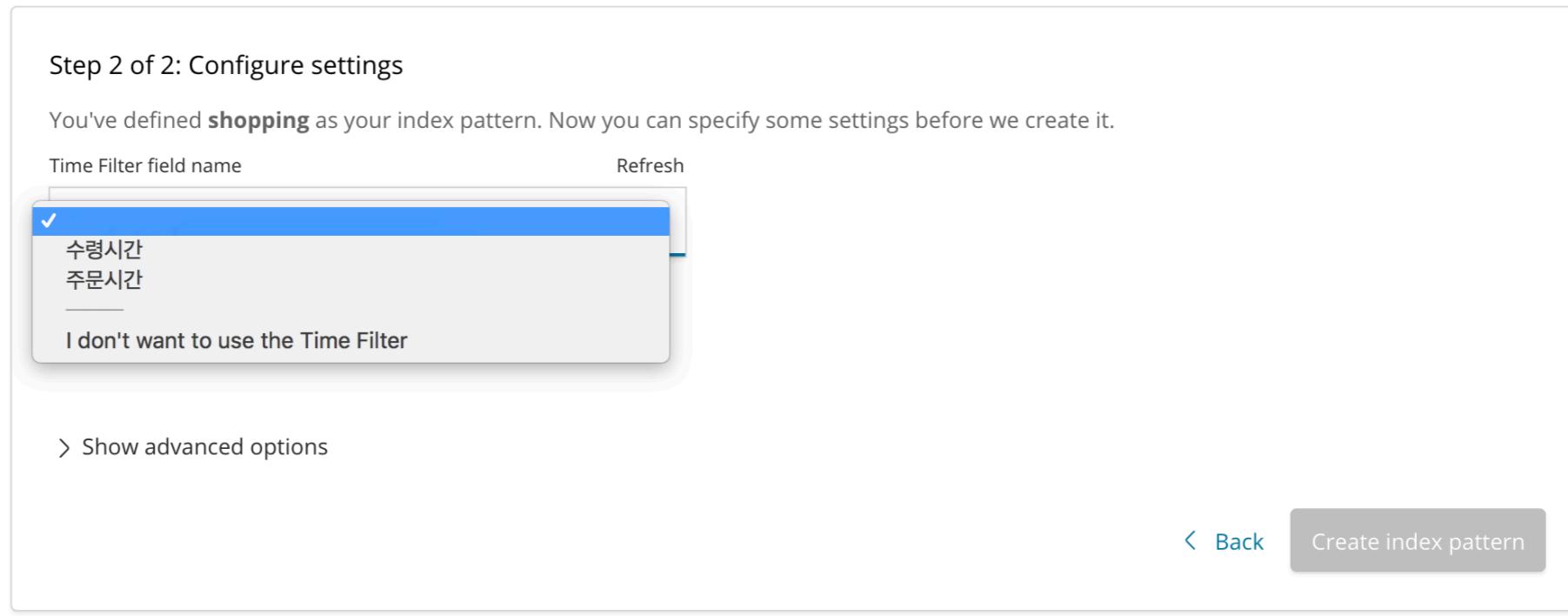
Time Filter field name  Refresh

수령시간  
 주문시간

I don't want to use the Time Filter

>Show advanced options

< Back Create index pattern



## 방금 배운 내용을 직접 해보자 ↗

예) id = higee

	<i>elasticsearch index</i>	<i>time field</i>	<i>kibana index pattern</i>
1	{id}_2018.06.17	주문시간	ex) higee_2018.06.17
2	{id}_2018.06.18	주문시간	ex) higee_2018.06.18
3	{id}_2018.06.19		ex) higee_2018.06.19

↓

(Elasticsearch Index는 실습용으로 사전에 생성 완료)

# Time Field를 선택한 경우

(바로 다음에 배울) Discover에서 보면 아래와 같이 나온다



# Time Field를 선택하지 않은 경우

(바로 다음에 배울) Discover에서 보면 아래와 같이 나온다

30 hits      New    Save    Open    Share

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

Add a filter +

Selected Fields: higee\_2018.06.19

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
- # 판매자평점

\_source

- 접수번호: 28 주문시간: June 19th 2018, 22:12:44.000 수령시간: June 20th 2018, 23:02:44.000 예약여부: 일반 배송메모: 주소 오류 고객ip: 110.163.10.67 고객성별: 여성 고객나이: 35 물건좌표: 36.9445 10196489894, 126.27522376689687 고객주소\_시도: 서울특별시 구매사이트: g마켓 판매자평점: 4 상품분류: 티셔츠 상품가격: 24,000 상품개수: 1 결제카드: 시티 \_id: X5JeBmQByNsCKuKnhs7L \_type: higee\_201 8.06.19 \_index: higee\_2018.06.19 \_score: -
- 접수번호: 1 주문시간: June 19th 2018, 22:11:07.000 수령시간: June 19th 2018, 22:40:07.000 예약여부: 예약 배송메모: 상품 이상 고객ip: 48.44.128.146 고객성별: 남성 고객나이: 21 물건좌표: 36.99106 192555027, 126.07622159065694 고객주소\_시도: 광주광역시 구매사이트: g마켓 판매자평점: 1 상품분류: 셔츠 상품가격: 17,000 상품개수: 1 결제카드: 신한 \_id: RJJJeBmQByNsCKuKneC5o \_type: higee\_2018.06.19 \_index: higee\_2018.06.19 \_score: -
- 접수번호: 13 주문시간: June 19th 2018, 22:07:37.000 수령시간: June 22nd 2018, 01:49:37.000 예약여부: 예약 배송메모: 상품 이상 고객ip: 148.108.45.0 고객성별: 여성 고객나이: 37 물건좌표: 35.31978 581757502, 128.59762252935715 고객주소\_시도: 대전광역시 구매사이트: g마켓 판매자평점: 3 상품분류: 코트 상품가격: 9,000 상품개수: 7 결제카드: 하나 \_id: UJJJeBmQByNsCKuKngMSL \_type: higee\_2018.06.19 \_index: higee\_2018.06.19 \_score: -
- 접수번호: 20 주문시간: June 19th 2018, 21:49:58.000 수령시간: June 22nd 2018, 14:31:58.000 예약여부: 일반 배송메모: 환불 요청 고객ip: 0.208.222.89 고객성별: 여성 고객나이: 61 물건좌표: 36.97688 20109463, 128.17307800840646 고객주소\_시도: 충청남도 구매사이트: 11번가 판매자평점: 2 상품분류: 셔츠 상품가격: 17,000 상품개수: 1 결제카드: 신한 \_id: V5JeBmQByNsCKuKnhM5B \_type: higee\_2018.06.19 \_index: higee\_2018.06.19 \_score: -
- 접수번호: 16 주문시간: June 19th 2018, 21:47:47.000 수령시간: June 22nd 2018, 13:01:47.000 예약여부: 일반 배송메모: 무인택배함에 보관 고객ip: 111.139.30.8 고객성별: 여성 고객나이: 63 물건좌표: 3 6.09413895455355, 128.33508336386473 고객주소\_시도: 서울특별시 구매사이트: 11번가 판매자평점: 3 상품분류: 청바지 상품가격: 12,000 상품개수: 7 결제카드: 우리 \_id: U5JeBmQByNsCKuKngM78 \_type: higee\_2018.06.19 \_index: higee\_2018.06.19 \_score: -
- 접수번호: 14 주문시간: June 19th 2018, 19:40:20.000 수령시간: June 20th 2018, 09:54:20.000 예약여부: 일반 배송메모: 관리실에 맡김 고객ip: 229.206.127.166 고객성별: 여성 고객나이: 25 물건좌표: 3 6.99456267681507, 128.62837338138763 고객주소\_시도: 충청북도 구매사이트: g마켓 판매자평점: 1 상품분류: 셔츠 상품가격: 22,000 상품개수: 7 결제카드: 국민 \_id: UZJeBmQByNsCKuKngM7f \_type: higee\_2018.06.19 \_index: higee\_2018.06.19 \_score: -
- 접수번호: 25 주문시간: June 19th 2018, 18:46:21.000 수령시간: June 23rd 2018, 21:52:21.000 예약여부: 일반 배송메모: 환불 요청 고객ip: 110.228.12.53 고객성별: 여성 고객나이: 32 물건좌표: 36.6025 5181324332, 128.30569355790078 고객주소\_시도: 서울특별시 구매사이트: 티몬 판매자평점: 4 상품분류: 자켓 상품가격: 17,000 상품개수: 1 결제카드: 롯데 \_id: XJJJeBmQByNsCKuKnhc6j \_type: higee\_2018.06.19 \_index: higee\_2018.06.19 \_score: -
- 접수번호: 23 주문시간: June 19th 2018, 16:53:26.000 수령시간: June 20th 2018, 21:03:26.000 예약여부: 일반 배송메모: 무인택배함에 보관 고객ip: 174.221.203.124 고객성별: 남성 고객나이: 34 물건좌표: 35.554869010416375, 128.93929278360747 고객주소\_시도: 제주특별자치도 구매사이트: g마켓 판매자평점: 1 상품분류: 코트 상품가격: 19,000 상품개수: 7 결제카드: 신한 \_id: WpJeBmQByNsCKuKnhc6G \_type: higee\_2018.06.19 \_index: higee\_2018.06.19 \_score: -

# 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 \, /, ?, ", <, >, |.

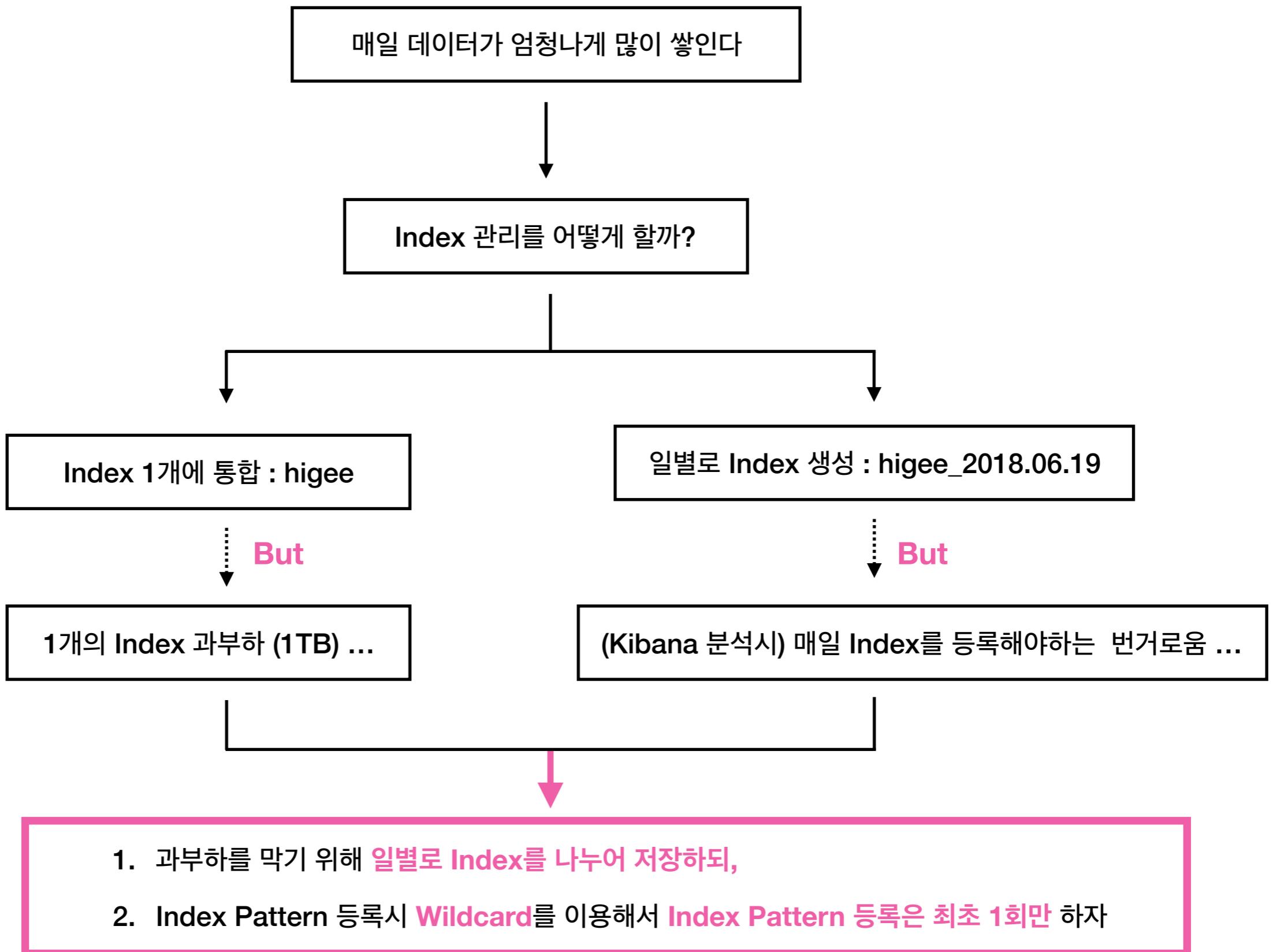
✓ Success! Your index pattern matches 1 index.

shopping

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

> Next step

## Index Patterns - Wildcard 사용



## Index Patterns - Wildcard 사용

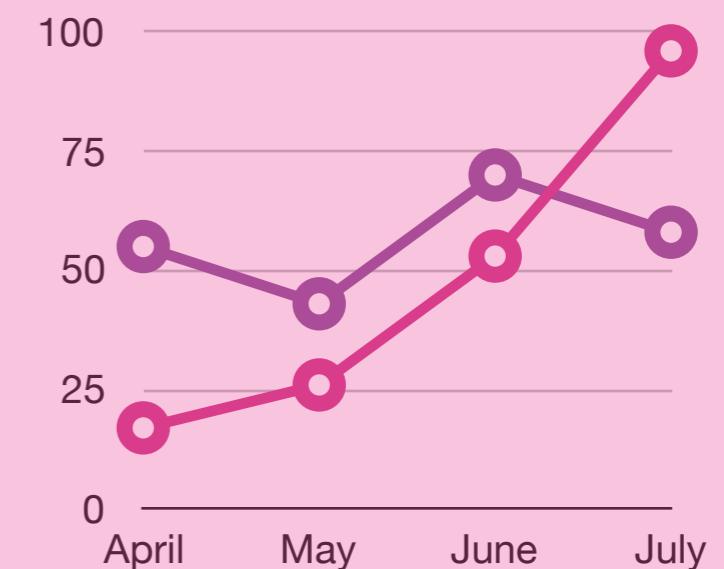
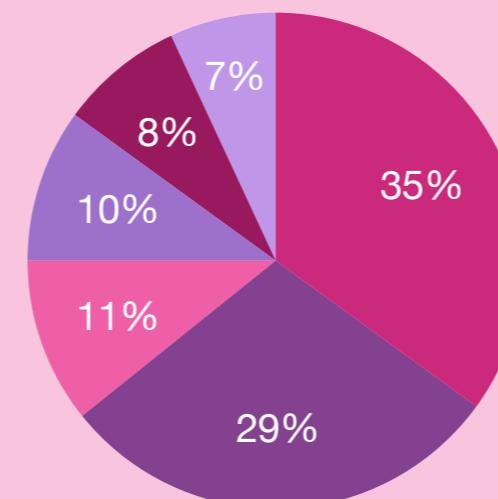
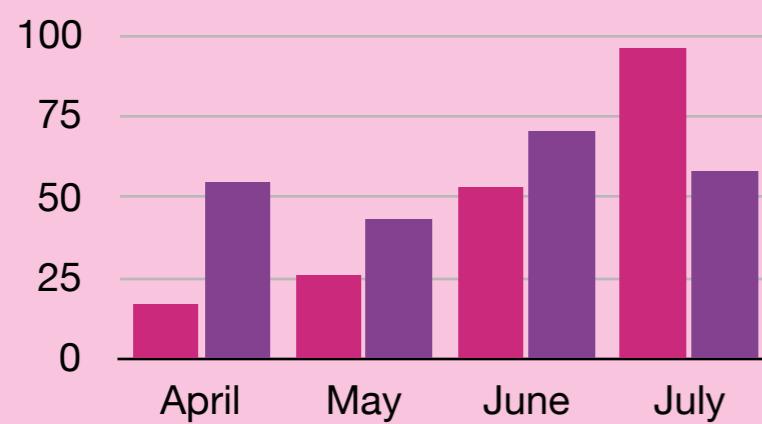
higee\_\*

데이터 저장은 분산, 검색 및 시각화는 통합

higee\_2018.06.17

higee\_2018.06.18

### Kibana



## 방금 배운 내용을 직접 해보자 ↗

예) id = higee

번호	<i>elasticsearch index</i>	<i>time field</i>	<i>kibana index pattern</i>
1	{id}_2018.06.17 {id}_2018.06.18 {id}_2018.06.20	주문시간	ex) higee_*

데이터 탐색

## Discover 페이지로 이동하자

14,697 hits

New Save Open Share C Auto-refresh < ⏪ This month ⏩

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

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\_code

t nginx.access.geoip.country\_code2

t nginx.access.geoip.country\_code3

t nginx.access.geoip.country\_name

nginx.access.geoip.ip

nginx.access.geoip.location

t nginx.access.geoip.postal\_code

t nginx.access.geoip.region\_code

t nginx.access.geoip.region\_name

t nginx.access.geoip.timezone

t nginx.access.http\_version

t nginx.access.method

t nginx.access.referrer

nginx.access.remote\_ip

June 1st 2018, 00:00:00.000 - June 30th 2018, 23:59:59.999 — Auto

Count

2018-06-03 09:00 2018-06-07 09:00 2018-06-11 09:00 2018-06-15 09:00 2018-06-19 09:00 2018-06-23 09:00 2018-06-27 09:00

2018-06-03 09:00 2018-06-07 09:00 2018-06-11 09:00 2018-06-15 09:00 2018-06-19 09:00 2018-06-23 09:00 2018-06-27 09:00

@timestamp per 12 hours

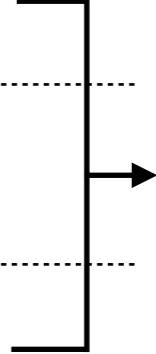
Time	_source
▶ June 9th 2018, 16:05:28.000	nginx.access.response_code: 200 nginx.access.geoip.timezone: Australia/Perth nginx.access.geoip.location: { "lat": 35, "lon": 105 } nginx.access.geoip.ip: 66.249.82.130 nginx.access.geoip.continent_code: AS nginx.access.referrer: - nginx.access.body_sent.bytes: 236 nginx.access.remote_ip: 66.249.82.130 nginx.access.url: / nginx.access.user_name: - nginx.access.user_agent.name: Chrome nginx.access.user_agent.os: Mac OS X nginx.access.user_agent.os_minor: 12 nginx.access.user_agent.major: 66 nginx.access.user_agent.build: nginx.access.user_agent.device: Other nginx.access.user_agent.minor: 0 nginx.access.user_agent.os_name: Mac OS X
▶ June 9th 2018, 16:15:33.000	nginx.access.response_code: 200 nginx.access.geoip.timezone: Australia/Perth nginx.access.geoip.location: { "lat": 35, "lon": 105 } nginx.access.geoip.ip: 66.249.82.195 nginx.access.geoip.continent_code: AS nginx.access.referrer: - nginx.access.body_sent.bytes: 156 nginx.access.remote_ip: 66.249.82.195 nginx.access.url: / nginx.access.user_name: - nginx.access.user_agent.name: Chrome nginx.access.user_agent.os: Mac OS X nginx.access.user_agent.os_minor: 12 nginx.access.user_agent.major: 66 nginx.access.user_agent.build: nginx.access.user_agent.device: Other nginx.access.user_agent.minor: 0 nginx.access.user_agent.os_name: Mac OS X
▶ June 9th 2018, 16:15:39.000	nginx.access.response_code: 200 nginx.access.geoip.timezone: Australia/Perth nginx.access.geoip.location: { "lat": 35, "lon": 105 } nginx.access.geoip.ip: 66.249.82.159 nginx.access.geoip.continent_code: AS nginx.access.referrer: http://kibana.higee.co/app/kibana nginx.access.body_sent.bytes: 1,028 nginx.access.remote_ip: 66.249.82.159 nginx.access.url: /ui/favicon/favicon-32x32.png nginx.access.user_name: - nginx.access.user_agent.name: Chrome nginx.access.user_agent.os: Mac OS X nginx.access.user_agent.os_minor: 12 nginx.access.user_agent.major: 66 nginx.access.user_agent.build: nginx.access.user_agent.device: Other nginx.access.user_agent.minor: 0
▶ June 9th 2018, 16:15:39.000	nginx.access.response_code: 200 nginx.access.geoip.timezone: Australia/Perth nginx.access.geoip.location: { "lat": 35, "lon": 105 } nginx.access.geoip.ip: 66.249.82.87 nginx.access.geoip.continent_code: AS nginx.access.referrer: http://kibana.higee.co/app/kibana nginx.access.body_sent.bytes: 412,210 nginx.access.remote_ip: 66.249.82.87 nginx.access.url: /bundles/commons.bundle.js?v=15571 nginx.access.user_name: - nginx.access.user_agent.name: Chrome nginx.access.user_agent.os: Mac OS X nginx.access.user_agent.os_minor: 12 nginx.access.user_agent.major: 66 nginx.access.user_agent.build: nginx.access.user_agent.device: Other nginx.access.user_agent.minor: 0

## 용어를 살펴보자

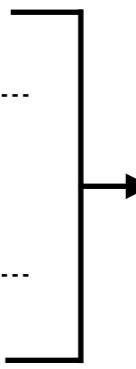
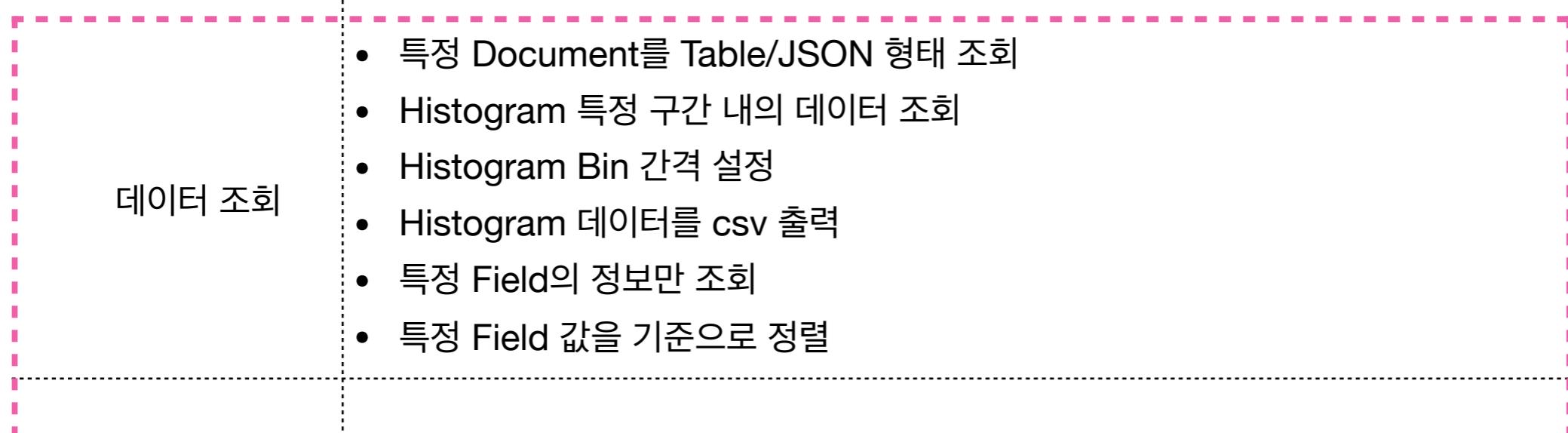
The screenshot illustrates the Kibana interface with several annotated sections:

- Query Bar:** Located at the top left, it shows a search bar with the placeholder "Search... (e.g. status:200 AND extension:PHP)". Above the search bar, it displays "14,697 hits". To the right of the search bar are buttons for "New", "Save", "Open", "Share", "Auto-refresh" (with a checkbox), and a date range selector ("This month"). A pink hand icon points to the search bar.
- Index Pattern:** Located on the far left, this sidebar lists various index patterns such as "Selected Fields" (including `? _source`), "Available Fields" (including `@timestamp`, `t _id`, `t _index`, etc.), and "nginx-\*". A pink hand icon points to the "nginx-\*" dropdown.
- Time Picker:** Located at the top right, it shows a date range from "June 1st 2018, 00:00:00.000" to "June 30th 2018, 23:59:59.999" with an "Auto" button. A pink hand icon points to the date range selector.
- Histogram:** This section contains a histogram chart titled "Count" versus "@timestamp per 12 hours". The x-axis shows dates from June 3rd to June 27th, 2018. The y-axis ranges from 0 to 5,000. The chart shows two major peaks: one around June 11th with a count of approximately 3,000, and another larger peak around June 15th with a count of approximately 4,000. A pink hand icon points to the histogram area.
- Document Table:** This section displays a table of log entries. The columns are "Time" and "\_source". Each row represents a log entry with timestamp and detailed log fields. A pink hand icon points to the first log entry.
- Side Navigation:** Located on the far left, this sidebar lists various index patterns and fields, similar to the main Index Pattern sidebar but with a different layout.

## 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

74 hits

New Save Open Share C 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: t \_id, t \_index, # \_score, t \_type, t 결제카드, □ 고객ip, # 고객나이, t 고객성별, t 고객주소\_시도, t 구매사이트, ○ 물건좌표, t 배송메모, # 배송소요시간, # 상품가격, # 상품개수, t 상품분류, ○ 수령시간, t 연령대, t 예약여부, # 접수번호

Time: June 1st 2018, 00:00:00.000 - June 17th 2018, 15:08:46.839 — Auto

Count: 주문시간 per 12 hours

Time Interval	Count
June 1st 2018, 00:00:00.000 - June 1st 2018, 09:00:00.000	5
June 1st 2018, 09:00:00.000 - June 1st 2018, 18:00:00.000	3
June 1st 2018, 18:00:00.000 - June 2nd 2018, 03:00:00.000	3
June 2nd 2018, 03:00:00.000 - June 2nd 2018, 09:00:00.000	2
June 2nd 2018, 09:00:00.000 - June 2nd 2018, 15:00:00.000	1
June 2nd 2018, 15:00:00.000 - June 3rd 2018, 03:00:00.000	2
June 3rd 2018, 03:00:00.000 - June 3rd 2018, 09:00:00.000	4
June 3rd 2018, 09:00:00.000 - June 3rd 2018, 15:00:00.000	2
June 3rd 2018, 15:00:00.000 - June 4th 2018, 03:00:00.000	1
June 4th 2018, 03:00:00.000 - June 4th 2018, 09:00:00.000	5
June 4th 2018, 09:00:00.000 - June 4th 2018, 15:00:00.000	1
June 4th 2018, 15:00:00.000 - June 5th 2018, 03:00:00.000	2
June 5th 2018, 03:00:00.000 - June 5th 2018, 09:00:00.000	2
June 5th 2018, 09:00:00.000 - June 5th 2018, 15:00:00.000	1
June 5th 2018, 15:00:00.000 - June 6th 2018, 03:00:00.000	2
June 6th 2018, 03:00:00.000 - June 6th 2018, 09:00:00.000	1
June 6th 2018, 09:00:00.000 - June 6th 2018, 15:00:00.000	3
June 6th 2018, 15:00:00.000 - June 7th 2018, 03:00:00.000	4
June 7th 2018, 03:00:00.000 - June 7th 2018, 09:00:00.000	2
June 7th 2018, 09:00:00.000 - June 7th 2018, 15:00:00.000	1
June 7th 2018, 15:00:00.000 - June 8th 2018, 03:00:00.000	2
June 8th 2018, 03:00:00.000 - June 8th 2018, 09:00:00.000	1
June 8th 2018, 09:00:00.000 - June 8th 2018, 15:00:00.000	4
June 8th 2018, 15:00:00.000 - June 9th 2018, 03:00:00.000	1
June 9th 2018, 03:00:00.000 - June 9th 2018, 09:00:00.000	4
June 9th 2018, 09:00:00.000 - June 9th 2018, 15:00:00.000	1
June 9th 2018, 15:00:00.000 - June 10th 2018, 03:00:00.000	1
June 10th 2018, 03:00:00.000 - June 10th 2018, 09:00:00.000	3
June 10th 2018, 09:00:00.000 - June 10th 2018, 15:00:00.000	1
June 10th 2018, 15:00:00.000 - June 11th 2018, 03:00:00.000	3
June 11th 2018, 03:00:00.000 - June 11th 2018, 09:00:00.000	4
June 11th 2018, 09:00:00.000 - June 11th 2018, 15:00:00.000	1
June 11th 2018, 15:00:00.000 - June 12th 2018, 03:00:00.000	2
June 12th 2018, 03:00:00.000 - June 12th 2018, 09:00:00.000	1
June 12th 2018, 09:00:00.000 - June 12th 2018, 15:00:00.000	3
June 12th 2018, 15:00:00.000 - June 13th 2018, 03:00:00.000	1
June 13th 2018, 03:00:00.000 - June 13th 2018, 09:00:00.000	3
June 13th 2018, 09:00:00.000 - June 13th 2018, 15:00:00.000	1
June 13th 2018, 15:00:00.000 - June 14th 2018, 03:00:00.000	2
June 14th 2018, 03:00:00.000 - June 14th 2018, 09:00:00.000	3
June 14th 2018, 09:00:00.000 - June 14th 2018, 15:00:00.000	1
June 14th 2018, 15:00:00.000 - June 15th 2018, 03:00:00.000	3
June 15th 2018, 03:00:00.000 - June 15th 2018, 09:00:00.000	2
June 15th 2018, 09:00:00.000 - June 15th 2018, 15:00:00.000	1
June 15th 2018, 15:00:00.000 - June 16th 2018, 03:00:00.000	4
June 16th 2018, 03:00:00.000 - June 16th 2018, 09:00:00.000	2
June 16th 2018, 09:00:00.000 - June 16th 2018, 15:00:00.000	1
June 16th 2018, 15:00:00.000 - June 17th 2018, 03:00:00.000	4
June 17th 2018, 03:00:00.000 - June 17th 2018, 09:00:00.000	2
June 17th 2018, 09:00:00.000 - June 17th 2018, 15:00:00.000	1

클릭

June 17th 2018, 13:38:01.000: 접수번호: 3,146 주문시간: June 17th 2018, 13:38:01.000 수령시간: June 19th 2018, 23:58:01.000 예약여부: 예약 배송메모: 부재중 고객ip: 176.147.96.208 고객성별: 남성 고객나이: 26 물건좌표: 35.66222235714742, 128.86281484306664 고객주소\_시도: 강원도 구매사이트: 11번가 판매자평점: 3 상품분류: 자켓 상품가격: 10,000 상품개수: 1 결제카드: 우리 \_id: bJJc\_WMBByNsCKuKn9LIZ \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 4 주문시간\_요일: 일 연령대: 20대 주문시간\_요일\_sort: 7 배송소요시간: 58

June 17th 2018, 12:57:41.000: 접수번호: 1,913 주문시간: June 17th 2018, 12:57:41.000 수령시간: June 20th 2018, 19:11:41.000 예약여부: 예약 배송메모: 상품 이상 고객ip: 42.206.62.51 고객성별: 남성 고객나이: 45 물건좌표: 35.05070912494262, 128.01427506670828 고객주소\_시도: 서울특별시 구매사이트: 옥션 판매자평점: 3 상품분류: 스웨터 상품가격: 19,000 상품개수: 7 결제카드: 하나 \_id: k5Jc\_WMBByNsCKuKnxa3K \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 3 주문시간\_요일: 일 연령대: 40대 주문시간\_요일\_sort: 7 배송소요시간: 78

June 17th 2018, 09:21:50.000: 접수번호: 2,821 주문시간: June 17th 2018, 09:21:50.000 수령시간: June 17th 2018, 15:45:50.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 83.120.178.37 고객성별: 남성 고객나이: 18 물건좌표: 36.5614902811839, 126.99503562607154 고객주소\_시도: 서울특별시 구매사이트: g마켓 판매자평점: 2 상품분류: 티셔츠 상품가격: 16,000 상품개수: 1 결제카드: 우리 \_id: o5JU\_WMBByNsCKuKnp5v\_ \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 0 주문시간\_요일: 일 연령대: 10대 주문시간\_요일\_sort: 7 배송소요시간: 6

June 17th 2018, 05:34:18.000: 접수번호: 4,037 주문시간: June 17th 2018, 05:34:18.000 수령시간: June 20th 2018, 14:20:18.000 예약여부: 일반 배송메모: 관리실에 맡김 고객ip: 138.47.51.1 고객성별: 여성 고객나이: 25 물건좌표: 36.9942450004816, 126.61644584928862 고객주소\_시도: 부산광역시 구매사이트: 11번가 판매자평점: 4 상품분류: 니트 상품가격: 23,000 상품개수: 7 결제카드: 국민 \_id: 65Jd\_WMBByNsCKuKnFrW0 \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 20 주문시간\_요일: 토 연령대: 20대 주문시간\_요일\_sort: 6 배송소요시간: 80

June 17th 2018, 03:34:01.000: 접수번호: 253 주문시간: June 17th 2018, 03:34:01.000 수령시간: June 17th 2018, 13:18:01.000 예약여부: 일반 배송메모: 환불 요청 고객ip: 43.242.134.156 고객성별: 여성 고객나이: 20 물건좌표: 36.74300021070816, 127.14772123494897 고객주소\_시도: 충청남도 구매사이트: g마켓 판매자평점: 4 상품분류: 셔츠 상품가격: 5,000 상품개수: 1

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

74 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: ? \_source

Available Fields: t \_id, t \_index, # \_score, t \_type, t 결제카드, # 고객ip, # 고객나이, t 고객성별, t 고객주소\_시도, t 구매사이트, # 물건좌표, t 배송메모, # 배송소요시간, # 상품가격, # 상품개수, t 상품분류, t 수령시간, t 연령대, t 예약여부, # 접수번호, # 주문시간, # 주문시간\_시간대, t 주문시간\_요일, # 주문시간\_요일\_sort

June 1st 2018, 00:00:00.000 - June 17th 2018, 15:08:46.839 — Auto

Count

주문시간 per 12 hours

Time	_source
June 17th 2018, 13:38:01.000	접수번호: 3,146 주문시간: June 17th 2018, 13:38:01.000 수령시간: June 19th 2018, 23:58:01.000 예약여부: 예약 배송메모: 부재증 고객ip: 176.147.96.208 고객성별: 남성 고객나이: 26 물건좌표: 35.6622235714742, 128.86281484306664 고객주소_시도: 강원도 구매사이트: 11번가 판매자평점: 3 상품분류: 자켓 상품가격: 10,000 상품개수: 1 결제카드: 우리 _id: bJJc_WMBByNsCKuKn9LIZ _type: shopping _index: shopping _score: - 주문시간_시간대: 4 주문시간_요일: 일 연령대: 20대 주문시간_요일_sort: 7 배송소요시간: 5

**클릭**

8

Table JSON

t _id	t _index	# _score	t _type	t 결제카드	# 고객ip	# 고객나이	t 고객성별	t 고객주소_시도	t 구매사이트	# 물건좌표	t 배송메모	# 배송소요시간	# 상품가격	# 상품개수	t 상품분류	t 수령시간	t 연령대	t 예약여부	# 접수번호	# 주문시간	# 주문시간_시간대	t 주문시간_요일	# 주문시간_요일_sort
bJJc_WMBByNsCKuKn9LIZ	shopping	-	shopping	우리	176.147.96.208	26	남성	강원도	11번가	35.6622235714742, 128.86281484306664	부재증	58	10,000	1	자켓	June 19th 2018, 23:58:01.000	20대	예약여부	3,146	June 17th 2018, 13:38:01.000	4	일	7

[View surrounding documents](#) [View single document](#)

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

74 hits

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

New Save Open Share Auto-refresh Month to date

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, ④ 물건좌표, t 배송메모, # 배송소요시간, # 상품가격, # 상품개수, t 상품분류, ⑤ 수령시간, t 연령대, t 예약여부, # 접수번호, ⑥ 주문시간, # 주문시간\_시간대, t 주문시간\_요일, # 주문시간\_요일\_sort

June 1st 2018, 00:00:00.000 - June 17th 2018, 15:08:46.839 — Auto

Count

주문시간 per 12 hours

Time \_source

June 17th 2018, 13:38:01.000 접수번호: 3,146 주문시간: June 17th 2018, 13:38:01.000 수령시간: June 19th 2018, 23:58:01.000 예약여부: 예약 배송메모: 부재증 고객ip: 176.147.96.208 고객성별: 남성 고객나이: 26 물건좌표: 35.66222235714742, 128.86281484306664 고객주소\_시도: 강원도 구매사이트: 11번가 판매자평점: 3 상품분류: 자켓 상품가격: 10,000 상품개수: 1 결제카드: 우리 \_id: bJJc\_WMBByNsCKuKn9LIZ \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 4 주문시간\_요일: 일 연령대: 20대 주문시간\_요일\_sort: 7 배송소요시간: 5

**클릭**

8

Table JSON

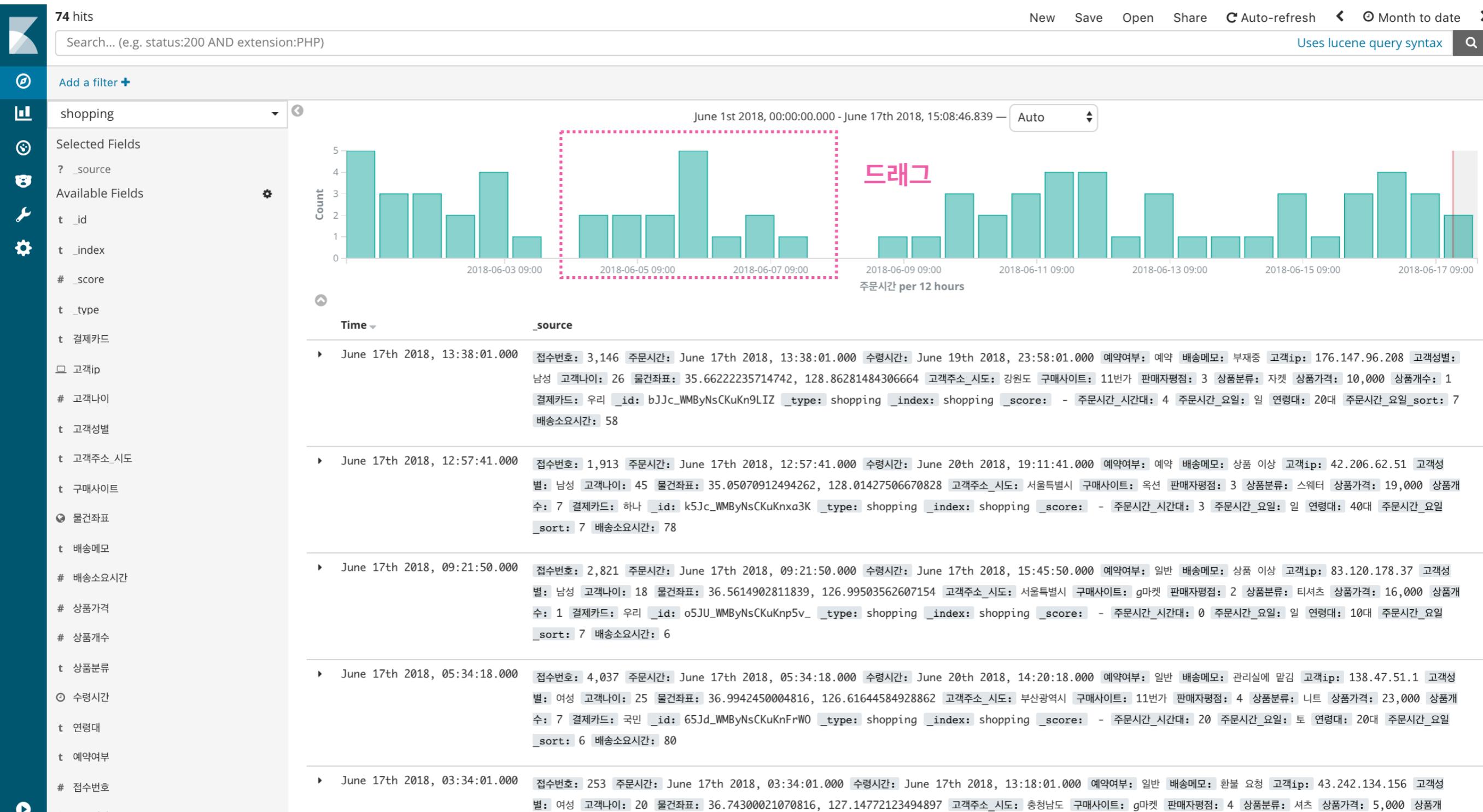
View surrounding documents View single document

```

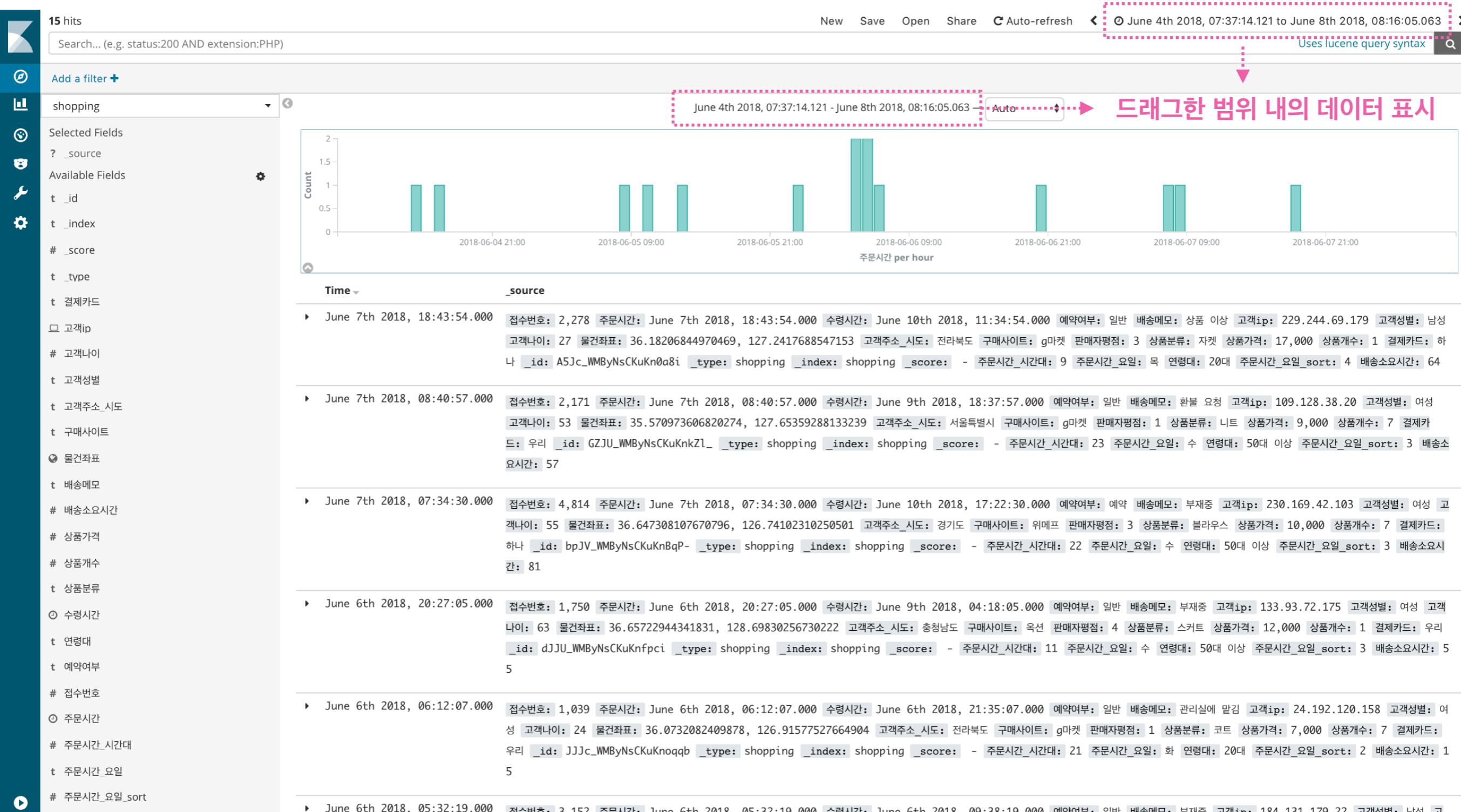
1 {
2   "_index": "shopping",
3   "_type": "shopping",
4   "_id": "bJJc_WMBByNsCKuKn9LIZ",
5   "_version": 1,
6   "_score": null,
7   "_source": {
8     "접수번호": 3146,
9     "주문시간": "2018-06-17T04:38:01",
10    "수령시간": "2018-06-19T14:58:01",
11    "예약여부": "예약",
12    "배송메모": "부재증",
13    "고객ip": "176.147.96.208",
14    "고객성별": "남성",
15    "고객나이": 26,
16    "물건좌표": "35.66222235714742, 128.86281484306664",
17    "고객주소_시도": "강원도",
18    "구매사이트": "11번가",
19    "판매자평점": 3,
20    "상품분류": "자켓",
21    "상품가격": 10000,
22    "상품개수": 1,
23    "결제카드": "우리"
24  },
25  "fields": {
26    "주문시간_시간대": [

```

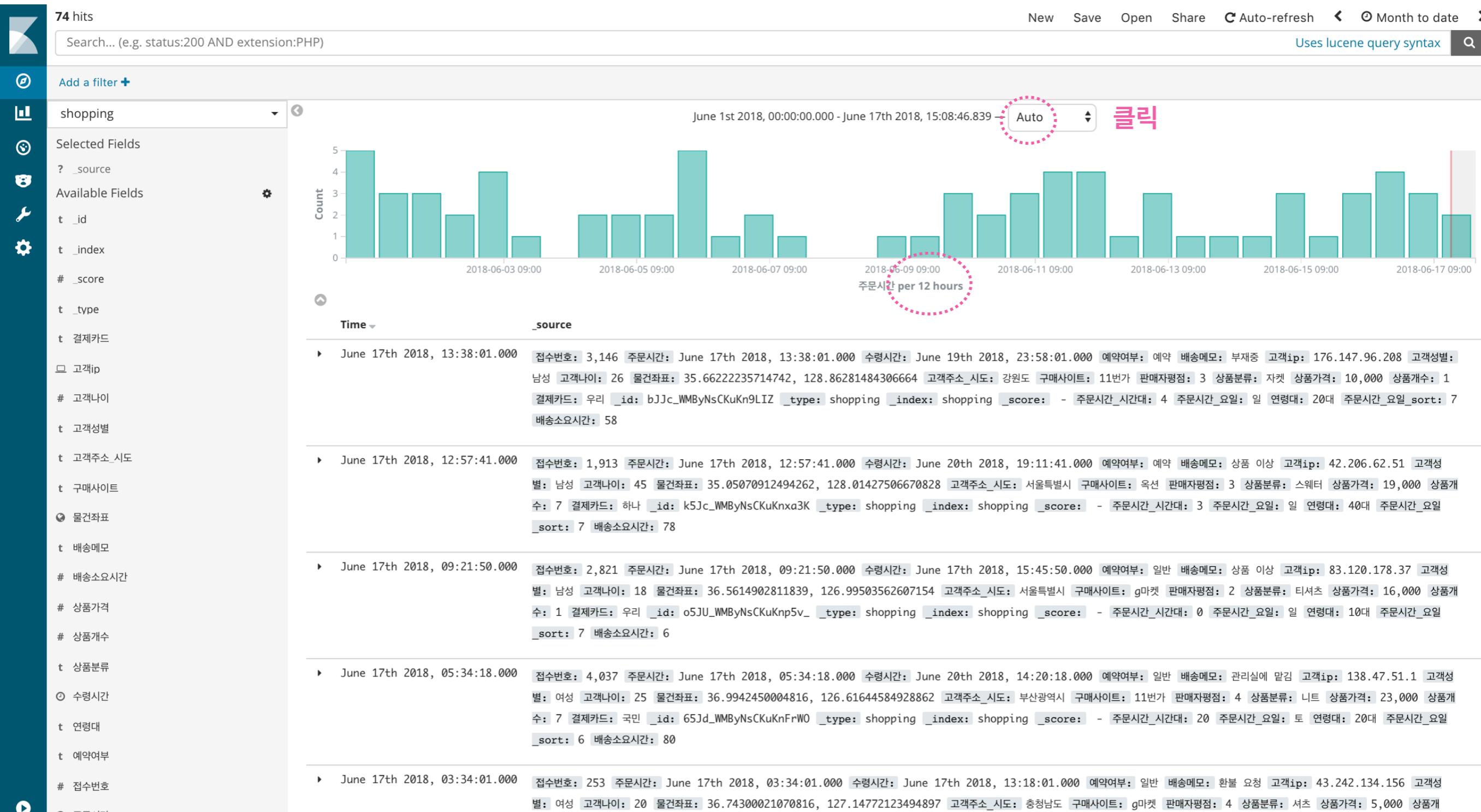
## 데이터 조회 - Histogram 특정 구간 내의 데이터 조회



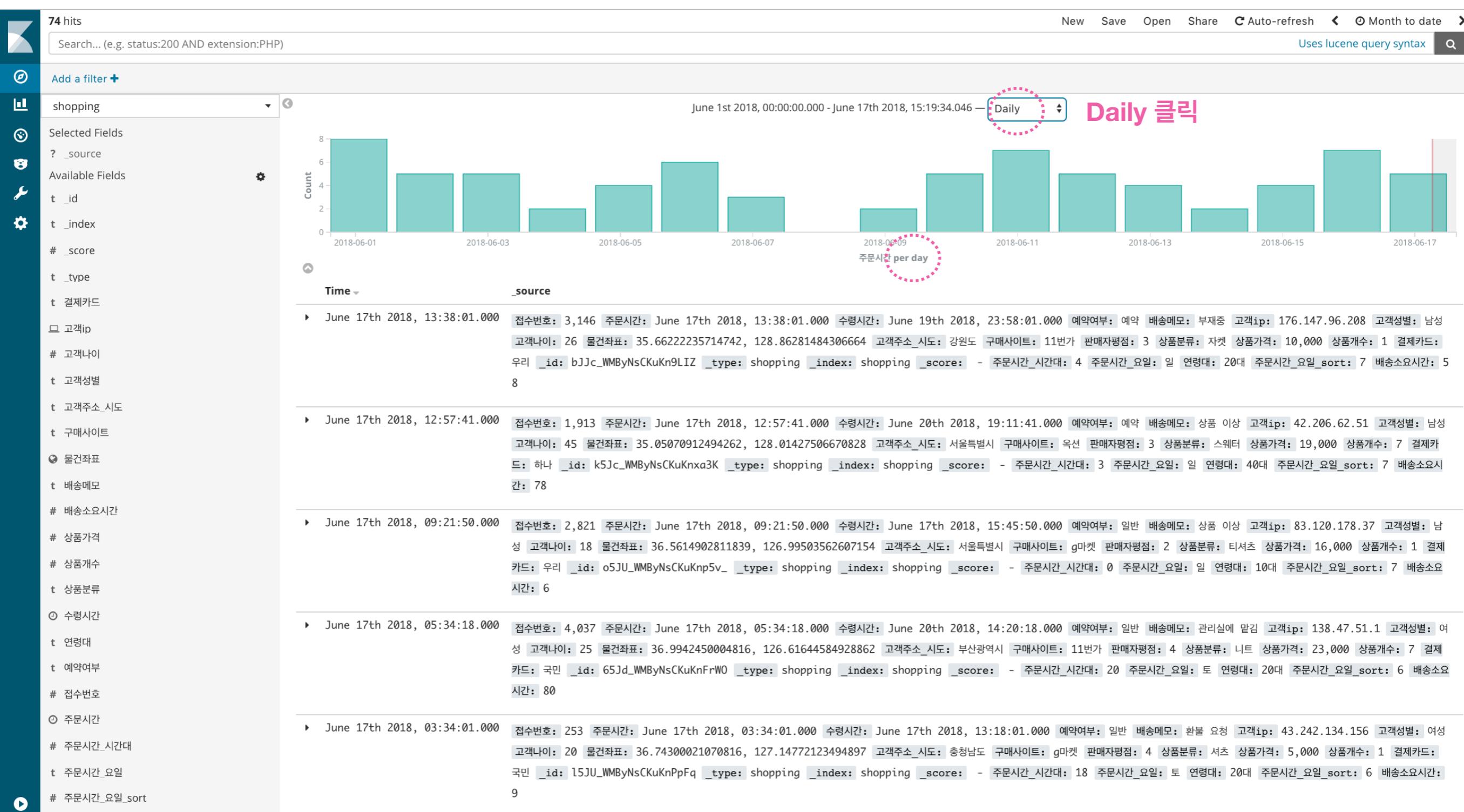
## 데이터 조회 - Histogram 특정 구간 내의 데이터 조회



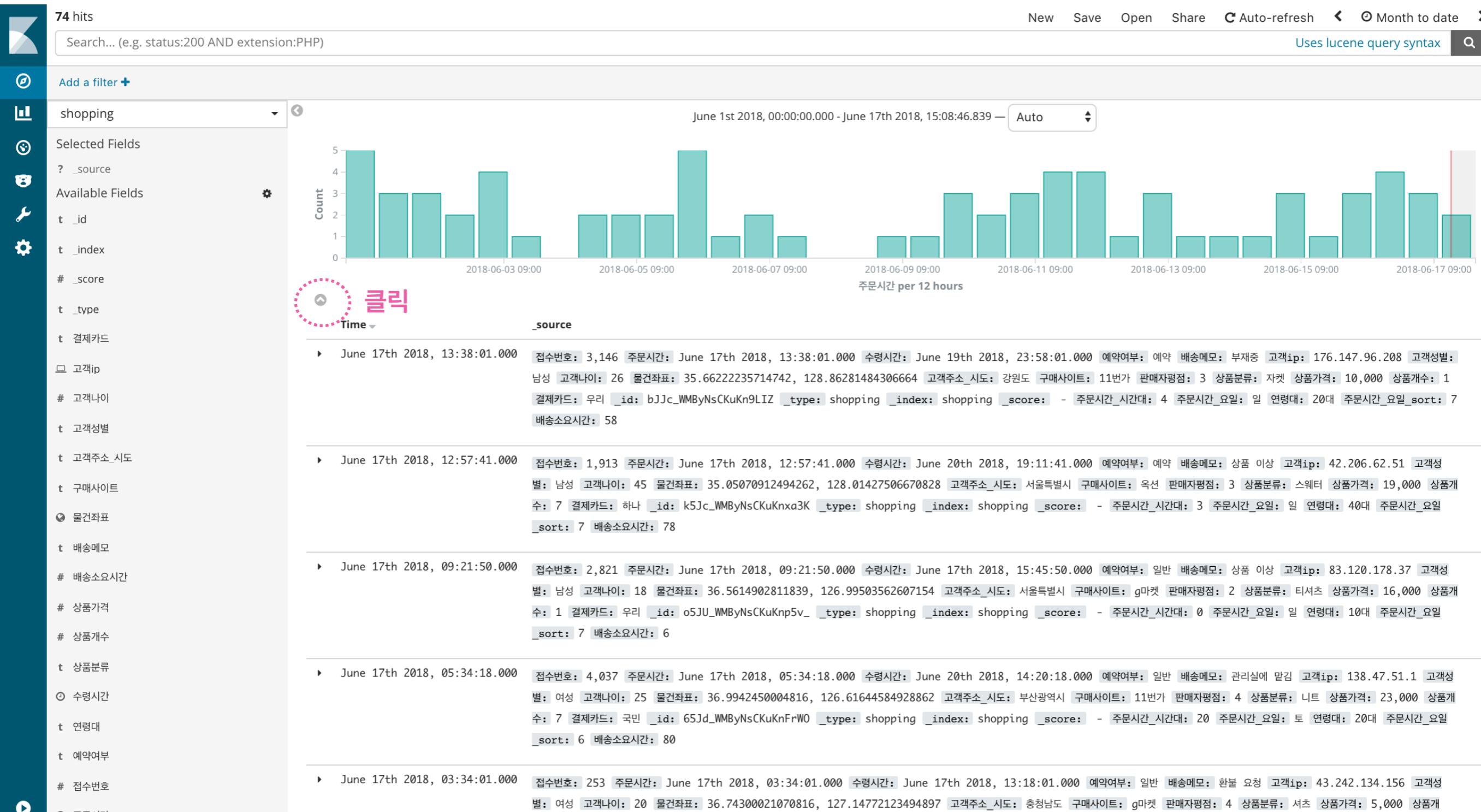
## 데이터 조회 - Histogram 간격 설정



## 데이터 조회 - Histogram 간격 설정



## 데이터 조회 - Histogram 데이터를 csv 출력



## 데이터 조회 - Histogram 데이터를 csv 출력

2,453 hits      New Save Open Share Auto-refresh Year to date  

Add a filter  Selected Fields 

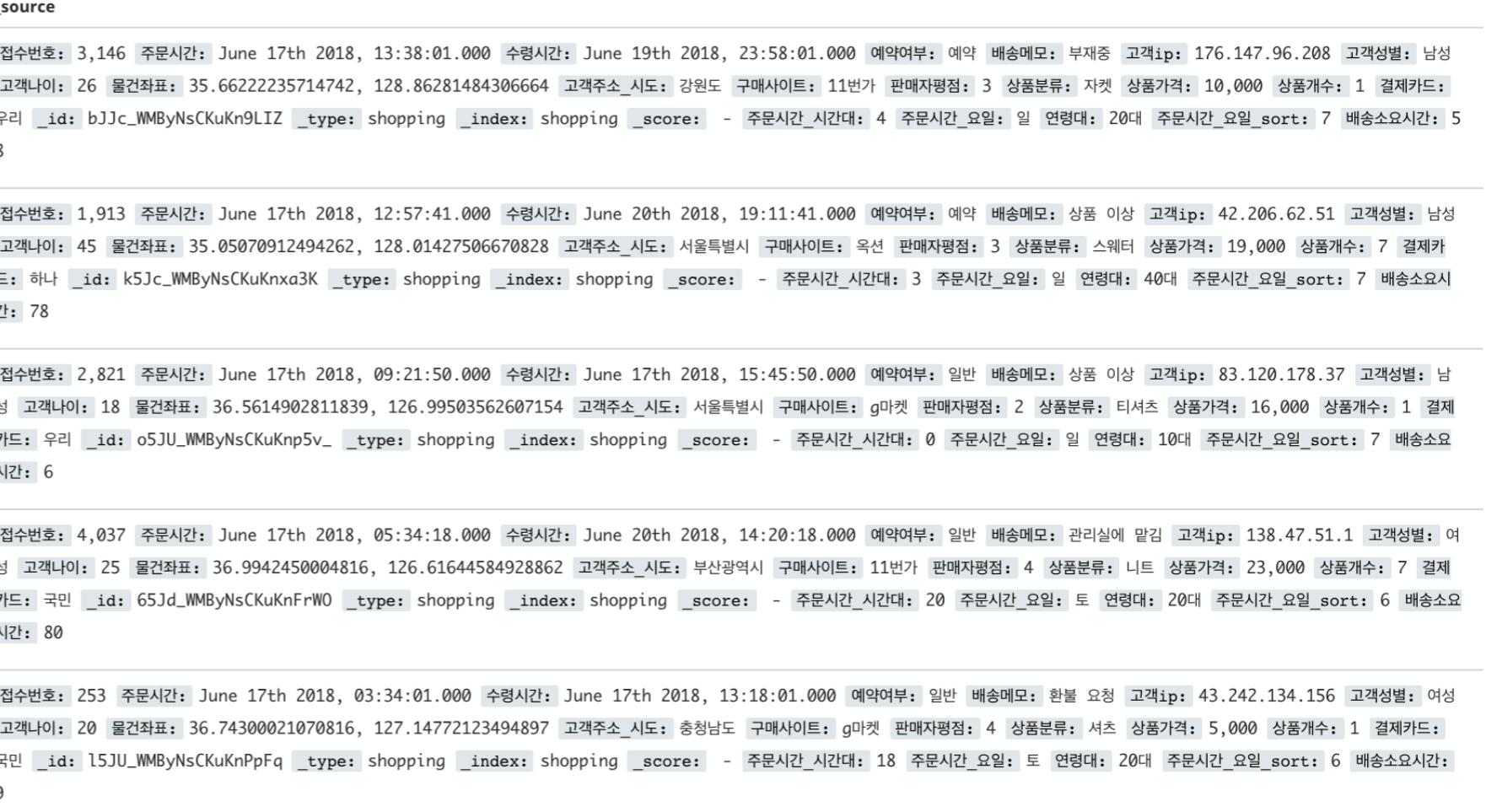
January 1st 2018, 00:00:00.000 - June 17th 2018, 15:23:35.669 — Auto

Table Request Response Statistics  
2018-01-09 1b  
2018-01-10 28

Export: Raw Formatted

**1. 최하단까지 스크롤 다운**

**2. 클릭**

Time 

Page Size 10

Time	_source
June 17th 2018, 13:38:01.000	접수번호: 3,146 주문시간: June 17th 2018, 13:38:01.000 수령시간: June 19th 2018, 23:58:01.000 예약여부: 예약 배송메모: 부재중 고객ip: 176.147.96.208 고객성별: 남성 고객나이: 26 물건좌표: 35.66222235714742, 128.86281484306664 고객주소_시도: 강원도 구매사이트: 11번가 판매자평점: 3 상품분류: 자켓 상품가격: 10,000 상품개수: 1 결제카드: 우리 _id: bJJc_WMBByNsCKuKn9LIZ _type: shopping _index: shopping _score: - 주문시간_시간대: 4 주문시간_요일: 일 연령대: 20대 주문시간_요일_sort: 7 배송소요시간: 5 8
June 17th 2018, 12:57:41.000	접수번호: 1,913 주문시간: June 17th 2018, 12:57:41.000 수령시간: June 20th 2018, 19:11:41.000 예약여부: 예약 배송메모: 상품 이상 고객ip: 42.206.62.51 고객성별: 남성 고객나이: 45 물건좌표: 35.05070912494262, 128.01427506670828 고객주소_시도: 서울특별시 구매사이트: 옥션 판매자평점: 3 상품분류: 스웨터 상품가격: 19,000 상품개수: 7 결제카드: 하나 _id: k5Jc_WMBByNsCKuKnxa3K _type: shopping _index: shopping _score: - 주문시간_시간대: 3 주문시간_요일: 일 연령대: 40대 주문시간_요일_sort: 7 배송소요시간: 78
June 17th 2018, 09:21:50.000	접수번호: 2,821 주문시간: June 17th 2018, 09:21:50.000 수령시간: June 17th 2018, 15:45:50.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 83.120.178.37 고객성별: 남성 고객나이: 18 물건좌표: 36.5614902811839, 126.99503562607154 고객주소_시도: 서울특별시 구매사이트: g마켓 판매자평점: 2 상품분류: 티셔츠 상품가격: 16,000 상품개수: 1 결제카드: 우리 _id: o5JU_WMBByNsCKuKnp5v_ _type: shopping _index: shopping _score: - 주문시간_시간대: 0 주문시간_요일: 일 연령대: 10대 주문시간_요일_sort: 7 배송소요시간: 6
June 17th 2018, 05:34:18.000	접수번호: 4,037 주문시간: June 17th 2018, 05:34:18.000 수령시간: June 20th 2018, 14:20:18.000 예약여부: 일반 배송메모: 관리실에 맡김 고객ip: 138.47.51.1 고객성별: 여성 고객나이: 25 물건좌표: 36.9942450004816, 126.61644584928862 고객주소_시도: 부산광역시 구매사이트: 11번가 판매자평점: 4 상품분류: 니트 상품가격: 23,000 상품개수: 7 결제카드: 국민 _id: 65Jd_WMBByNsCKuKnFrW0 _type: shopping _index: shopping _score: - 주문시간_시간대: 20 주문시간_요일: 토 연령대: 20대 주문시간_요일_sort: 6 배송소요시간: 80
June 17th 2018, 03:34:01.000	접수번호: 253 주문시간: June 17th 2018, 03:34:01.000 수령시간: June 17th 2018, 13:18:01.000 예약여부: 일반 배송메모: 환불 요청 고객ip: 43.242.134.156 고객성별: 여성 고객나이: 20 물건좌표: 36.74300021070816, 127.14772123494897 고객주소_시도: 충청남도 구매사이트: g마켓 판매자평점: 4 상품분류: 셔츠 상품가격: 5,000 상품개수: 1 결제카드: 국민 _id: l5JU_WMBByNsCKuKnPpFq _type: shopping _index: shopping _score: - 주문시간_시간대: 18 주문시간_요일: 토 연령대: 20대 주문시간_요일_sort: 6 배송소요시간: 9

# 데이터 조회 - Histogram 데이터를 csv 출력

2,453 hits

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

New Save Open Share Auto-refresh < ⌂ Year to date >

Add a filter +

Selected Fields: shopping

Available Fields: t \_id, t \_index, # \_score, t \_type, t 결제카드, # 고객ip, # 고객나이, t 고객성별, t 고객주소\_시도, t 구매사이트, # 물건좌표, t 배송메모, # 배송소요시간, # 상품가격, # 상품개수, t 상품분류, ⌂ 수령시간, t 연령대, t 예약여부, # 접수번호, ⌂ 주문시간, # 주문시간\_시간대

January 1st 2018, 00:00:00.000 - June 17th 2018, 15:23:35.669 — Auto

Table Request Response Statistics

Date	Count
2018-01-09	16
2018-01-10	28

Export: Raw Formatted

Page Size 10

Time	_source
June 17th 2018, 13:38:01.000	접수번호: 3,146 주문시간: June 17th 2018, 13:38:01.000 수령시간: June 19th 2018, 23:58:01.000 예약여부: 예약 배송메모: 부재중 고객ip: 176.147.96.208 고객성별: 남성 고객나이: 26 물건좌표: 35.6622235714742, 128.86281484306664 고객주소_시도: 강원도 구매사이트: 11번가 판매자평점: 3 상품분류: 자켓 상품가격: 10,000 상품개수: 1 결제카드: 우리 _id: bJJc_WMBByNsCKuKn9LIZ _type: shopping _index: shopping _score: - 주문시간_시간대: 4 주문시간_요일: 일 연령대: 20대 주문시간_요일_sort: 7 배송소요시간: 5
June 17th 2018, 12:57:41.000	접수번호: 1,913 주문시간: June 17th 2018, 12:57:41.000 수령시간: June 20th 2018, 19:11:41.000 예약여부: 예약 배송메모: 상품 이상 고객ip: 42.206.62.51 고객성별: 남성 고객나이: 45 물건좌표: 35.05070912494262, 128.01427506670828 고객주소_시도: 서울특별시 구매사이트: 옥션 판매자평점: 3 상품분류: 스웨터 상품가격: 19,000 상품개수: 7 결제카드: 하나 _id: k5Jc_WMBByNsCKuKnxa3K _type: shopping _index: shopping _score: - 주문시간_시간대: 3 주문시간_요일: 일 연령대: 40대 주문시간_요일_sort: 7 배송소요시간: 78
June 17th 2018, 09:21:50.000	접수번호: 2,821 주문시간: June 17th 2018, 09:21:50.000 수령시간: June 17th 2018, 15:45:50.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 83.120.178.37 고객성별: 남성 고객나이: 18 물건좌표: 36.5614902811839, 126.99503562607154 고객주소_시도: 서울특별시 구매사이트: g마켓 판매자평점: 2 상품분류: 티셔츠 상품가격: 16,000 상품개수: 1 결제카드: 우리 _id: o5JU_WMBByNsCKuKn5v_ _type: shopping _index: shopping _score: - 주문시간_시간대: 0 주문시간_요일: 일 연령대: 10대 주문시간_요일_sort: 7 배송소요시간: 6
June 17th 2018, 05:34:18.000	접수번호: 4,037 주문시간: June 17th 2018, 05:34:18.000 수령시간: June 20th 2018, 14:20:18.000 예약여부: 일반 배송메모: 관리실에 맡김 고객ip: 138.47.51.1 고객성별: 여성 고객나이: 25 물건좌표: 36.9942450004816, 126.61644584928862 고객주소_시도: 부산광역시 구매사이트: 11번가 판매자평점: 4 상품분류: 니트 상품가격: 23,000 상품개수: 7 결제카드: 국민 _id: 65Jd_WMBByNsCKuKnFrW0 _type: shopping _index: shopping _score: - 주문시간_시간대: 20 주문시간_요일: 토 연령대: 20대 주문시간_요일_sort: 6 배송소요시간: 80
June 17th 2018, 03:34:01.000	접수번호: 253 주문시간: June 17th 2018, 03:34:01.000 수령시간: June 17th 2018, 13:18:01.000 예약여부: 일반 배송메모: 환불 요청 고객ip: 43.242.134.156 고객성별: 여성 고객나이: 20 물건좌표: 36.74300021070816, 127.14772123494897 고객주소_시도: 충청남도 구매사이트: g마켓 판매자평점: 4 상품분류: 셔츠 상품가격: 5,000 상품개수: 1 결제카드:

New Saved Search.csv

Show All

다운로드 완료

## 데이터 조회 - 특정 Field의 정보만 조회

74 hits

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

New Save Open Share Auto-refresh Month to date

Uses lucene query syntax

Add a filter +

Selected Fields: shopping

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

June 1st 2018, 00:00:00.000 - June 17th 2018, 15:33:54.214 — Auto

Count

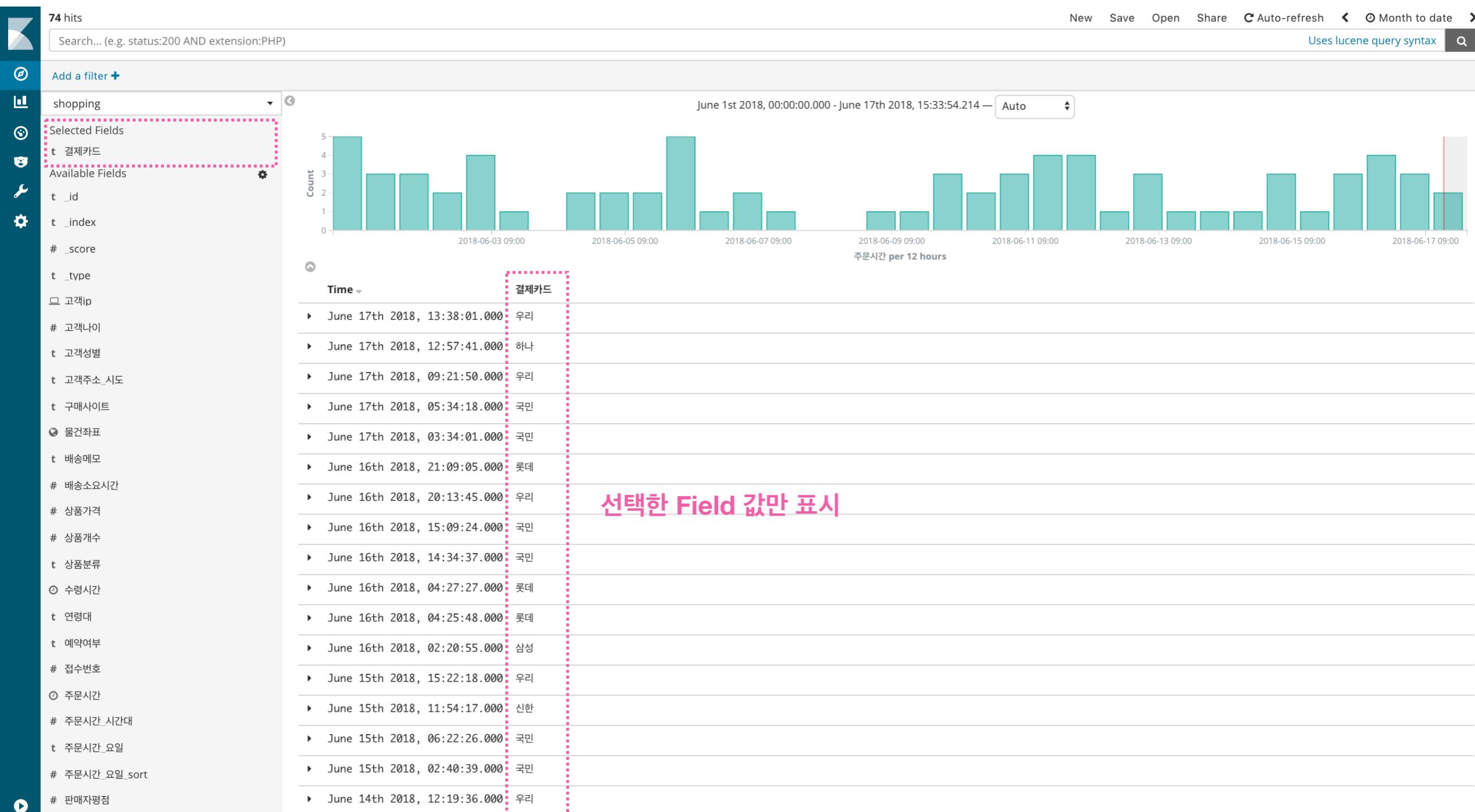
주문시간 per 12 hours

Time source

**클릭**

Time	source
June 17th 2018, 13:38:01.000	접수번호: 3,146 주문시간: June 17th 2018, 13:38:01.000 수령시간: June 19th 2018, 23:58:01.000 예약여부: 예약 배송메모: 부재중 고객ip: 176.147.96.208 고객성별: 남성 고객나이: 26 물건좌표: 35.66222235714742, 128.86281484306664 고객주소_시도: 강원도 구매사이트: 11번가 판매자평점: 3 상품분류: 자켓 상품가격: 10,000 상품개수: 1 결제카드: 우리 _id: bJJc_WMBByNsCKuKn9LIZ _type: shopping _index: shopping _score: - 주문시간_시간대: 4 주문시간_요일: 일 연령대: 20대 주문시간_요일_sort: 7 배송소요시간: 5
June 17th 2018, 12:57:41.000	접수번호: 1,913 주문시간: June 17th 2018, 12:57:41.000 수령시간: June 20th 2018, 19:11:41.000 예약여부: 예약 배송메모: 상품 이상 고객ip: 42.206.62.51 고객성별: 남성 고객나이: 45 물건좌표: 35.05070912494262, 128.01427506670828 고객주소_시도: 서울특별시 구매사이트: 옥션 판매자평점: 3 상품분류: 스웨터 상품가격: 19,000 상품개수: 7 결제카드: 하나 _id: k5Jc_WMBByNsCKuKnxa3K _type: shopping _index: shopping _score: - 주문시간_시간대: 3 주문시간_요일: 일 연령대: 40대 주문시간_요일_sort: 7 배송소요시간: 78
June 17th 2018, 09:21:50.000	접수번호: 2,821 주문시간: June 17th 2018, 09:21:50.000 수령시간: June 17th 2018, 15:45:50.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 83.120.178.37 고객성별: 남성 고객나이: 18 물건좌표: 36.5614902811839, 126.99503562607154 고객주소_시도: 서울특별시 구매사이트: g마켓 판매자평점: 2 상품분류: 티셔츠 상품가격: 16,000 상품개수: 1 결제카드: 우리 _id: o5JU_WMBByNsCKuKnp5v_ _type: shopping _index: shopping _score: - 주문시간_시간대: 0 주문시간_요일: 일 연령대: 10대 주문시간_요일_sort: 7 배송소요시간: 6
June 17th 2018, 05:34:18.000	접수번호: 4,037 주문시간: June 17th 2018, 05:34:18.000 수령시간: June 20th 2018, 14:20:18.000 예약여부: 일반 배송메모: 관리실에 맡김 고객ip: 138.47.51.1 고객성별: 여성 고객나이: 25 물건좌표: 36.9942450004816, 126.61644584928862 고객주소_시도: 부산광역시 구매사이트: 11번가 판매자평점: 4 상품분류: 니트 상품가격: 23,000 상품개수: 7 결제카드: 국민 _id: 65Jd_WMBByNsCKuKnFrW0 _type: shopping _index: shopping _score: - 주문시간_시간대: 20 주문시간_요일: 토 연령대: 20대 주문시간_요일_sort: 6 배송소요시간: 80
June 17th 2018, 03:34:01.000	접수번호: 253 주문시간: June 17th 2018, 03:34:01.000 수령시간: June 17th 2018, 13:18:01.000 예약여부: 일반 배송메모: 환불 요청 고객ip: 43.242.134.156 고객성별: 여성 고객나이: 20 물건좌표: 36.74300021070816, 127.14772123494897 고객주소_시도: 충청남도 구매사이트: g마켓 판매자평점: 4 상품분류: 셔츠 상품가격: 5,000 상품개수: 1 결제카드: 국민 _id: l5JU_WMBByNsCKuKnPpFq _type: shopping _index: shopping _score: - 주문시간_시간대: 18 주문시간_요일: 토 연령대: 20대 주문시간_요일_sort: 6 배송소요시간: 9

## 데이터 조회 - 특정 Field의 정보만 조회



## 데이터 조회 - 특정 Field 값을 기준으로 정렬

74 hits

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

New Save Open Share Auto-refresh Month to date

Uses lucene query syntax

Add a filter +

shopping

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  
# 배송소요시간  
# 상품가격  
# 상품개수  
t 상품분류  
⌚ 수령시간  
t 연령대  
t 예약여부  
# 접수번호  
⌚ 주문시간  
# 주문시간\_시간대  
t 주문시간\_요일  
# 주문시간\_요일\_sort

June 1st 2018, 00:00:00.000 - June 17th 2018, 15:33:54.214 — Auto

Count

Time source

June 17th 2018, 13:38:01.000

접수번호: 3,146 주문시간: June 17th 2018, 13:38:01.000 수령시간: June 19th 2018, 23:58:01.000 예약여부: 예약 배송메모: 부재중 고객ip: 176.147.96.208 고객성별: 남성 고객나이: 26 물건좌표: 35.66222235714742, 128.86281484306664 고객주소\_시도: 강원도 구매사이트: 11번가 판매자평점: 3 상품분류: 자켓 상품가격: 10,000 상품개수: 1 결제카드: 우리 \_id: bJJc\_WMBByNsCKuKn9LIZ \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 4 주문시간\_요일: 일 연령대: 20대 주문시간\_요일\_sort: 7 배송소요시간: 5 8

June 17th 2018, 12:57:41.000

접수번호: 1,913 주문시간: June 17th 2018, 12:57:41.000 수령시간: June 20th 2018, 19:11:41.000 예약여부: 예약 배송메모: 상품 이상 고객ip: 42.206.62.51 고객성별: 남성 고객나이: 45 물건좌표: 35.05070912494262, 128.01427506670828 고객주소\_시도: 서울특별시 구매사이트: 옥션 판매자평점: 3 상품분류: 스웨터 상품가격: 19,000 상품개수: 7 결제카드: 하나 \_id: k5Jc\_WMBByNsCKuKnxa3K \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 3 주문시간\_요일: 일 연령대: 40대 주문시간\_요일\_sort: 7 배송소요시간: 78

June 17th 2018, 09:21:50.000

접수번호: 2,821 주문시간: June 17th 2018, 09:21:50.000 수령시간: June 17th 2018, 15:45:50.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 83.120.178.37 고객성별: 남성 고객나이: 18 물건좌표: 36.5614902811839, 126.99503562607154 고객주소\_시도: 서울특별시 구매사이트: g마켓 판매자평점: 2 상품분류: 티셔츠 상품가격: 16,000 상품개수: 1 결제카드: 우리 \_id: o5JU\_WMBByNsCKuKnp5v\_ \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 0 주문시간\_요일: 일 연령대: 10대 주문시간\_요일\_sort: 7 배송소요시간: 6

June 17th 2018, 05:34:18.000

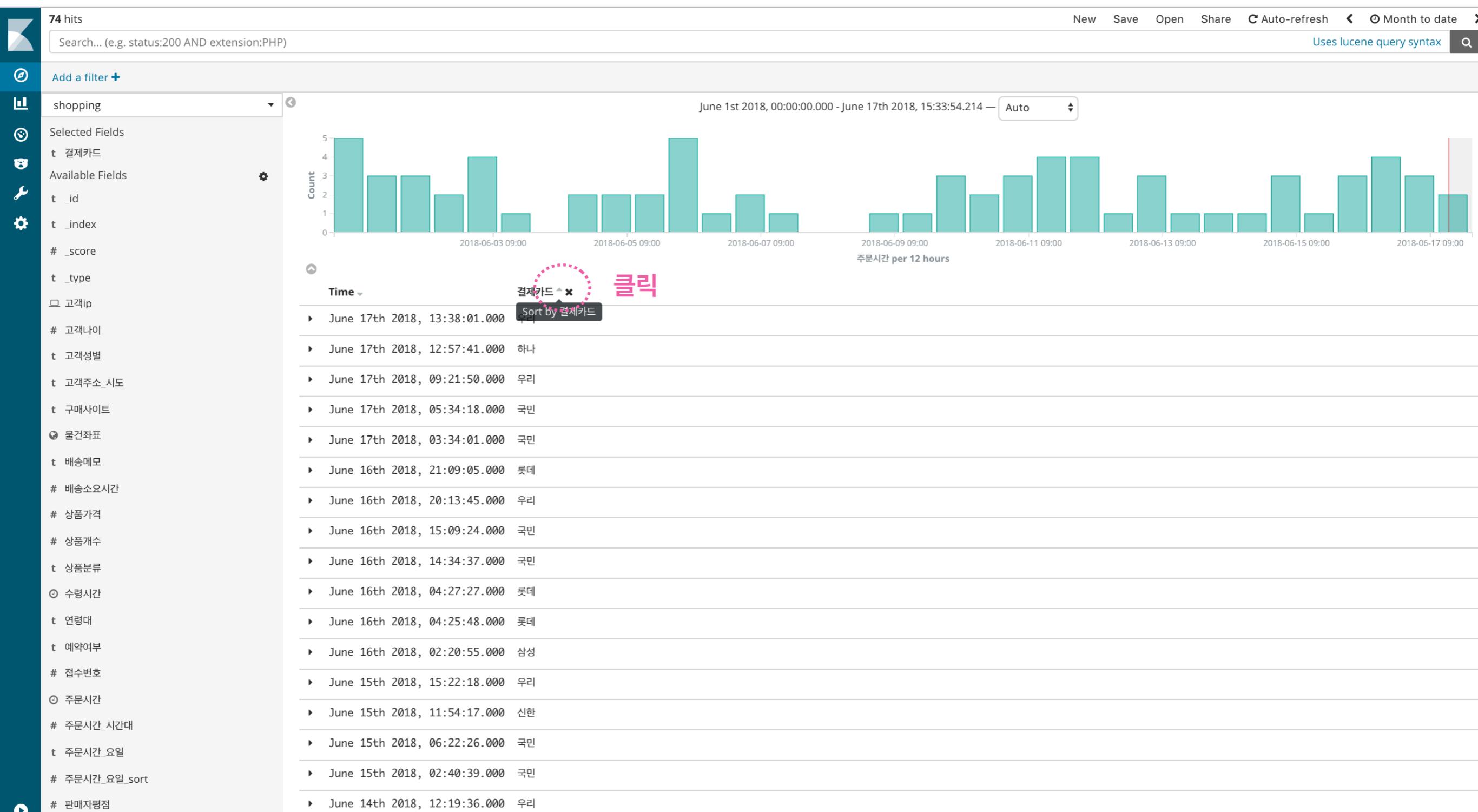
접수번호: 4,037 주문시간: June 17th 2018, 05:34:18.000 수령시간: June 20th 2018, 14:20:18.000 예약여부: 일반 배송메모: 관리실에 맡김 고객ip: 138.47.51.1 고객성별: 여성 고객나이: 25 물건좌표: 36.9942450004816, 126.61644584928862 고객주소\_시도: 부산광역시 구매사이트: 11번가 판매자평점: 4 상품분류: 니트 상품가격: 23,000 상품개수: 7 결제카드: 국민 \_id: 65Jd\_WMBByNsCKuKnFrW0 \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 20 주문시간\_요일: 토 연령대: 20대 주문시간\_요일\_sort: 6 배송소요시간: 80

June 17th 2018, 03:34:01.000

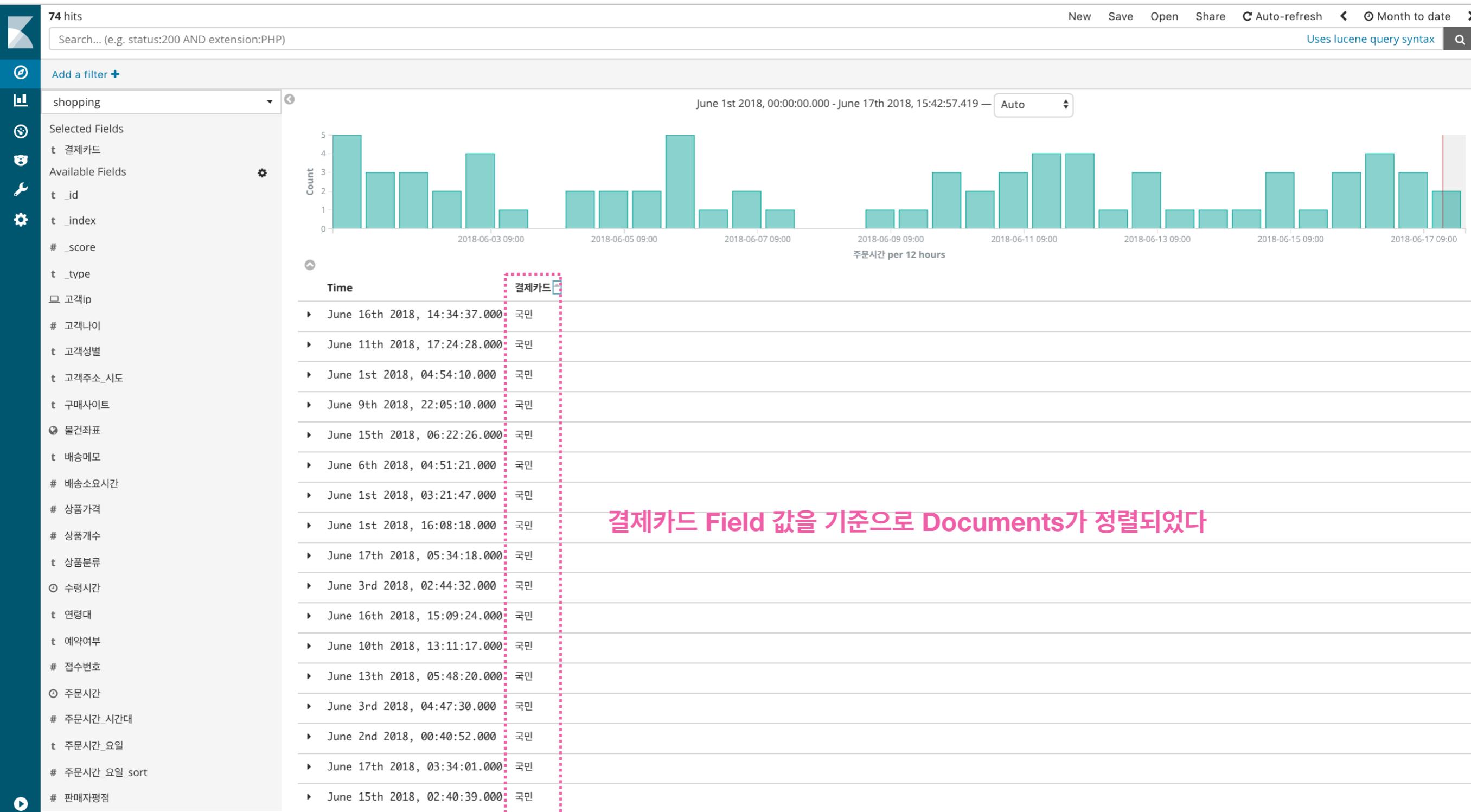
접수번호: 253 주문시간: June 17th 2018, 03:34:01.000 수령시간: June 17th 2018, 13:18:01.000 예약여부: 일반 배송메모: 환불 요청 고객ip: 43.242.134.156 고객성별: 여성 고객나이: 20 물건좌표: 36.74300021070816, 127.14772123494897 고객주소\_시도: 충청남도 구매사이트: g마켓 판매자평점: 4 상품분류: 셔츠 상품가격: 5,000 상품개수: 1 결제카드: 국민 \_id: l5JU\_WMBByNsCKuKnPpFq \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 18 주문시간\_요일: 토 연령대: 20대 주문시간\_요일\_sort: 6 배송소요시간: 9

**클릭**

## 데이터 조회 - 특정 Field 값을 기준으로 정렬



## 데이터 조회 - 특정 Field 값을 기준으로 정렬



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

**Month to Date 사이의 Documents 개수**

74 hits

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

New Save Open Share Auto-refresh Month to date Uses lucene query syntax

Add a filter +

Selected Fields: shopping

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

Time range: June 1st 2018, 00:00:00.000 - June 17th 2018, 15:08:46.839 — Auto

Count per 12 hours:

Time Interval	Count
2018-06-01 00:00:00.000 - 2018-06-01 12:00:00.000	5
2018-06-01 12:00:00.000 - 2018-06-02 00:00:00.000	3
2018-06-02 00:00:00.000 - 2018-06-02 12:00:00.000	3
2018-06-02 12:00:00.000 - 2018-06-03 00:00:00.000	2
2018-06-03 00:00:00.000 - 2018-06-03 12:00:00.000	4
2018-06-03 12:00:00.000 - 2018-06-04 00:00:00.000	1
2018-06-04 00:00:00.000 - 2018-06-04 12:00:00.000	2
2018-06-04 12:00:00.000 - 2018-06-05 00:00:00.000	2
2018-06-05 00:00:00.000 - 2018-06-05 12:00:00.000	2
2018-06-05 12:00:00.000 - 2018-06-06 00:00:00.000	5
2018-06-06 00:00:00.000 - 2018-06-06 12:00:00.000	1
2018-06-06 12:00:00.000 - 2018-06-07 00:00:00.000	1
2018-06-07 00:00:00.000 - 2018-06-07 12:00:00.000	2
2018-06-07 12:00:00.000 - 2018-06-08 00:00:00.000	1
2018-06-08 00:00:00.000 - 2018-06-08 12:00:00.000	3
2018-06-08 12:00:00.000 - 2018-06-09 00:00:00.000	1
2018-06-09 00:00:00.000 - 2018-06-09 12:00:00.000	2
2018-06-09 12:00:00.000 - 2018-06-10 00:00:00.000	3
2018-06-10 00:00:00.000 - 2018-06-10 12:00:00.000	4
2018-06-10 12:00:00.000 - 2018-06-11 00:00:00.000	4
2018-06-11 00:00:00.000 - 2018-06-11 12:00:00.000	3
2018-06-11 12:00:00.000 - 2018-06-12 00:00:00.000	1
2018-06-12 00:00:00.000 - 2018-06-12 12:00:00.000	3
2018-06-12 12:00:00.000 - 2018-06-13 00:00:00.000	1
2018-06-13 00:00:00.000 - 2018-06-13 12:00:00.000	2
2018-06-13 12:00:00.000 - 2018-06-14 00:00:00.000	1
2018-06-14 00:00:00.000 - 2018-06-14 12:00:00.000	3
2018-06-14 12:00:00.000 - 2018-06-15 00:00:00.000	2
2018-06-15 00:00:00.000 - 2018-06-15 12:00:00.000	3
2018-06-15 12:00:00.000 - 2018-06-16 00:00:00.000	2
2018-06-16 00:00:00.000 - 2018-06-16 12:00:00.000	3
2018-06-16 12:00:00.000 - 2018-06-17 00:00:00.000	2
2018-06-17 00:00:00.000 - 2018-06-17 12:00:00.000	5

Time: June 17th 2018, 13:38:01.000

\_source: 접수번호: 3,146 주문시간: June 17th 2018, 13:38:01.000 수령시간: June 19th 2018, 23:58:01.000 예약여부: 예약 배송메모: 부재중 고객ip: 176.147.96.208 고객성별: 남성 고객나이: 26 물건좌표: 35.66222235714742, 128.86281484306664 고객주소\_시도: 강원도 구매사이트: 11번가 판매자평점: 3 상품분류: 자켓 상품가격: 10,000 상품개수: 1 결제카드: 우리 \_id: bJJc\_WMBByNsCKuKn9LIZ \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 4 주문시간\_요일: 일 연령대: 20대 주문시간\_요일\_sort: 7 배송소요시간: 58

Time: June 17th 2018, 12:57:41.000

\_source: 접수번호: 1,913 주문시간: June 17th 2018, 12:57:41.000 수령시간: June 20th 2018, 19:11:41.000 예약여부: 예약 배송메모: 상품 이상 고객ip: 42.206.62.51 고객성별: 남성 고객나이: 45 물건좌표: 35.05070912494262, 128.01427506670828 고객주소\_시도: 서울특별시 구매사이트: 옥션 판매자평점: 3 상품분류: 스웨터 상품가격: 19,000 상품개수: 7 결제카드: 하나 \_id: k5Jc\_WMBByNsCKuKnxa3K \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 3 주문시간\_요일: 일 연령대: 40대 주문시간\_요일\_sort: 7 배송소요시간: 78

Time: June 17th 2018, 09:21:50.000

\_source: 접수번호: 2,821 주문시간: June 17th 2018, 09:21:50.000 수령시간: June 17th 2018, 15:45:50.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 83.120.178.37 고객성별: 남성 고객나이: 18 물건좌표: 36.5614902811839, 126.99503562607154 고객주소\_시도: 서울특별시 구매사이트: g마켓 판매자평점: 2 상품분류: 티셔츠 상품가격: 16,000 상품개수: 1 결제카드: 우리 \_id: o5JU\_WMBByNsCKuKn5v\_ \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 0 주문시간\_요일: 일 연령대: 10대 주문시간\_요일\_sort: 7 배송소요시간: 6

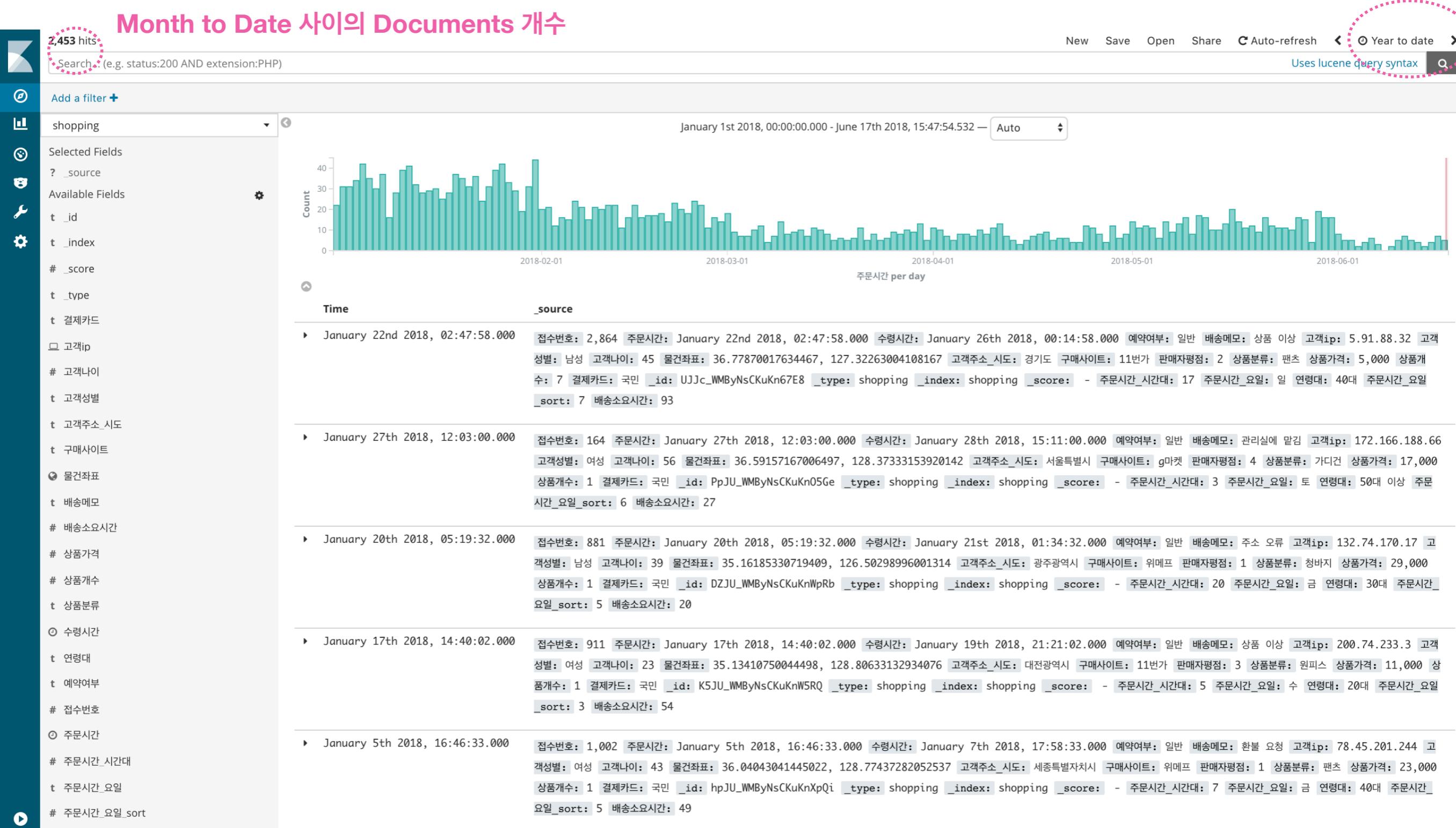
Time: June 17th 2018, 05:34:18.000

\_source: 접수번호: 4,037 주문시간: June 17th 2018, 05:34:18.000 수령시간: June 20th 2018, 14:20:18.000 예약여부: 일반 배송메모: 관리실에 맡김 고객ip: 138.47.51.1 고객성별: 여성 고객나이: 25 물건좌표: 36.9942450004816, 126.61644584928862 고객주소\_시도: 부산광역시 구매사이트: 11번가 판매자평점: 4 상품분류: 니트 상품가격: 23,000 상품개수: 7 결제카드: 국민 \_id: 65Jd\_WMBByNsCKuKnFrW0 \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 20 주문시간\_요일: 토 연령대: 20대 주문시간\_요일\_sort: 6 배송소요시간: 80

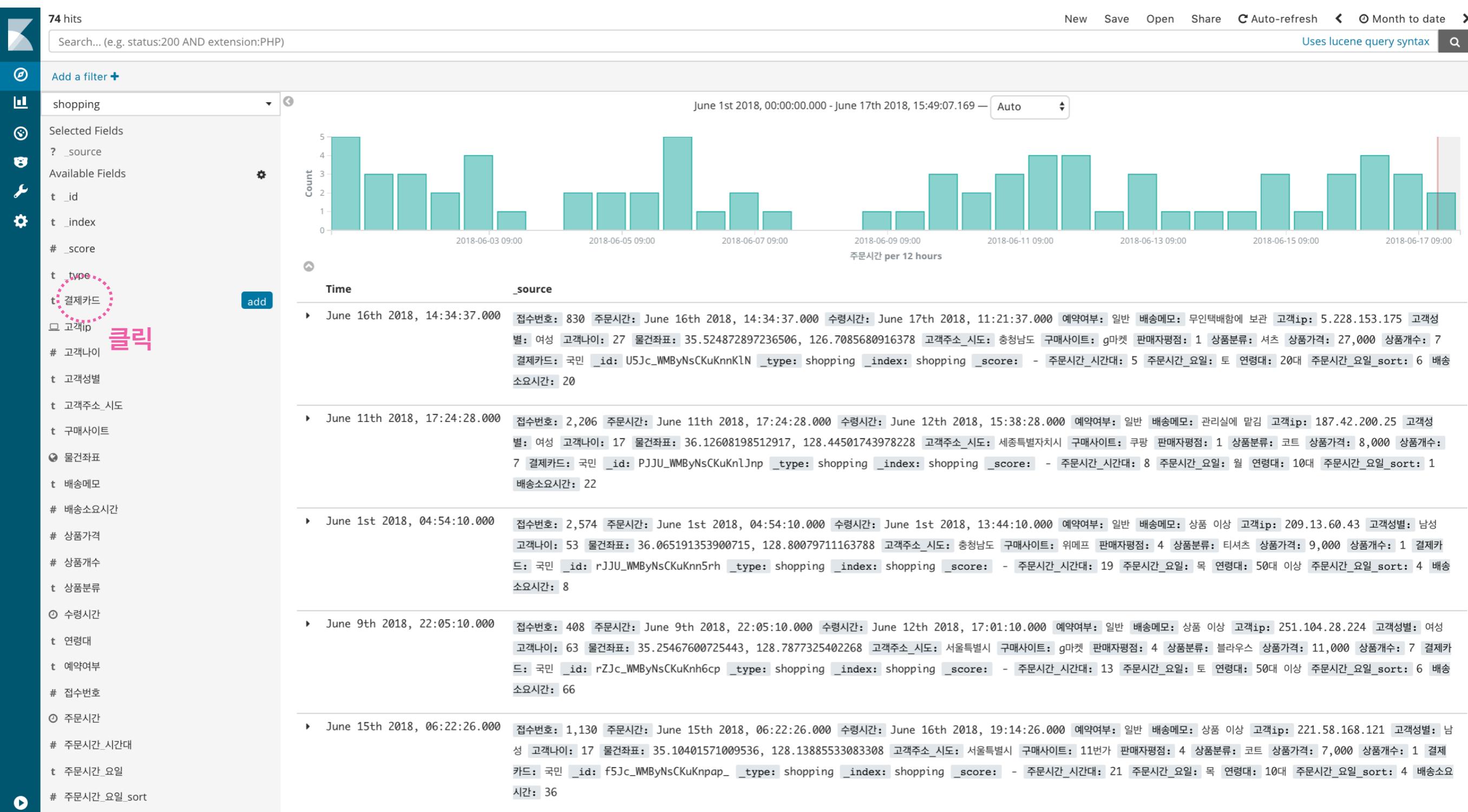
Time: June 17th 2018, 03:34:01.000

\_source: 접수번호: 253 주문시간: June 17th 2018, 03:34:01.000 수령시간: June 17th 2018, 13:18:01.000 예약여부: 일반 배송메모: 환불 요청 고객ip: 43.242.134.156 고객성별: 여성 고객나이: 20 물건좌표: 36.74300021070816, 127.14772123494897 고객주소\_시도: 충청남도 구매사이트: g마켓 판매자평점: 4 상품분류: 셔츠 상품가격: 5,000 상품개수: 1

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



## 데이터 통계 - 특정 Field Value의 분포 확인 (주로 Categorical Data, 상위 500개)



## 데이터 통계 - 특정 Field Value의 분포 확인 (주로 Categorical Data, 상위 500개)

74 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: t \_id, t \_index, # \_score, t \_type, t 결제카드

Count

June 1st 2018, 00:00:00.000 - June 17th 2018, 15:49:07.169 — Auto

Time \_source

Top 5 values in 74 / 74 records

결제카드	Count (%)
국민	39.2%
우리	23.0%
하나	14.9%
롯데	10.8%
신한	8.1%

Visualize

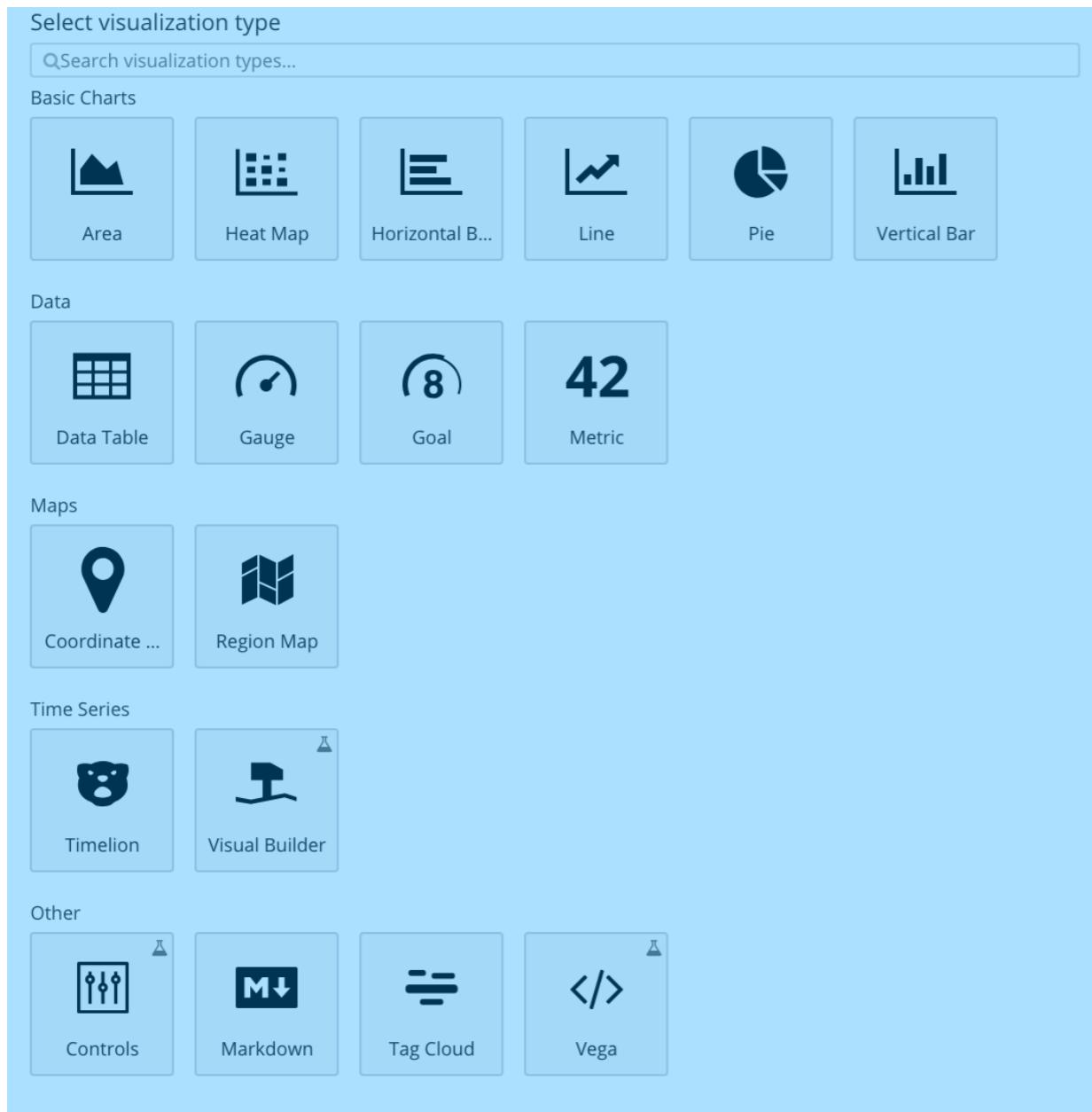
선택한 Field 값의 분포를 보여준다

- June 16th 2018, 14:34:37.000: 접수번호: 830 주문시간: June 16th 2018, 14:34:37.000 수령시간: June 17th 2018, 11:21:37.000 예약여부: 일반 배송메모: 무인택배함에 보관 고객ip: 5.228.153.175 고객성별: 여성 고객나이: 27 물건좌표: 35.524872897236506, 126.7085680916378 고객주소\_시도: 충청남도 구매사이트: g마켓 판매자평점: 1 상품분류: 셔츠 상품가격: 27,000 상품개수: 7 결제카드: 국민 \_id: U5Jc\_WMBByNsCKuKnnKLN \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 5 주문시간\_요일: 토 연령대: 20대 주문시간\_요일\_sort: 6 배송소요시간: 20
- June 11th 2018, 17:24:28.000: 접수번호: 2,206 주문시간: June 11th 2018, 17:24:28.000 수령시간: June 12th 2018, 15:38:28.000 예약여부: 일반 배송메모: 관리실에 맡김 고객ip: 187.42.200.25 고객성별: 여성 고객나이: 17 물건좌표: 36.12608198512917, 128.44501743978228 고객주소\_시도: 세종특별자치시 구매사이트: 쿠팡 판매자평점: 1 상품분류: 코트 상품가격: 8,000 상품개수: 7 결제카드: 국민 \_id: PJJU\_WMBByNsCKuKnJnp \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 8 주문시간\_요일: 월 연령대: 10대 주문시간\_요일\_sort: 1 배송소요시간: 22
- June 1st 2018, 04:54:10.000: 접수번호: 2,574 주문시간: June 1st 2018, 04:54:10.000 수령시간: June 1st 2018, 13:44:10.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 209.13.60.43 고객성별: 남성 고객나이: 53 물건좌표: 36.065191353900715, 128.80079711163788 고객주소\_시도: 충청남도 구매사이트: 위메프 판매자평점: 4 상품분류: 티셔츠 상품가격: 9,000 상품개수: 1 결제카드: 국민 \_id: rJJU\_WMBByNsCKuKn5rh \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 19 주문시간\_요일: 목 연령대: 50대 이상 주문시간\_요일\_sort: 4 배송소요시간: 8
- June 9th 2018, 22:05:10.000: 접수번호: 408 주문시간: June 9th 2018, 22:05:10.000 수령시간: June 12th 2018, 17:01:10.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 251.104.28.224 고객성별: 여성 고객나이: 63 물건좌표: 35.25467600725443, 128.7877325402268 고객주소\_시도: 서울특별시 구매사이트: g마켓 판매자평점: 4 상품분류: 블라우스 상품가격: 11,000 상품개수: 7 결제카드: 국민 \_id: rZJc\_WMBByNsCKuKh6cp \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 13 주문시간\_요일: 토 연령대: 50대 이상 주문시간\_요일\_sort: 6 배송소요시간: 66
- June 15th 2018, 06:22:26.000: 접수번호: 1,130 주문시간: June 15th 2018, 06:22:26.000 수령시간: June 16th 2018, 19:14:26.000 예약여부: 일반 배송메모: 상품 이상 고객ip: 221.58.168.121 고객성별: 남성 고객나이: 17 물건좌표: 35.10401571009536, 128.13885533083308 고객주소\_시도: 서울특별시 구매사이트: 11번가 판매자평점: 4 상품분류: 코트 상품가격: 7,000 상품개수: 1 결제카드: 국민 \_id: f5Jc\_WMBByNsCKuKnpap\_ \_type: shopping \_index: shopping \_score: - 주문시간\_시간대: 21 주문시간\_요일: 목 연령대: 10대 주문시간\_요일\_sort: 4 배송소요시간: 36

데이터 시각화

## 어떤 시각화를 할 수 있을까?

공식



비공식

network

cohort

dendrogram

:

Kibana Visualize는 어렵나?



그럴 수 있다. 그렇다면 왜?



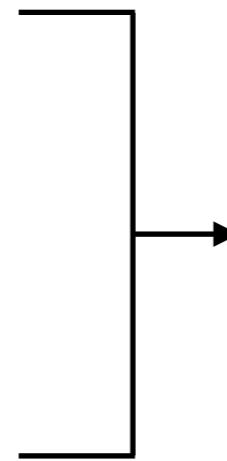
용어가 너무 낯설어서



눈 딱 감고 맛보기로 1개만 따라해보자

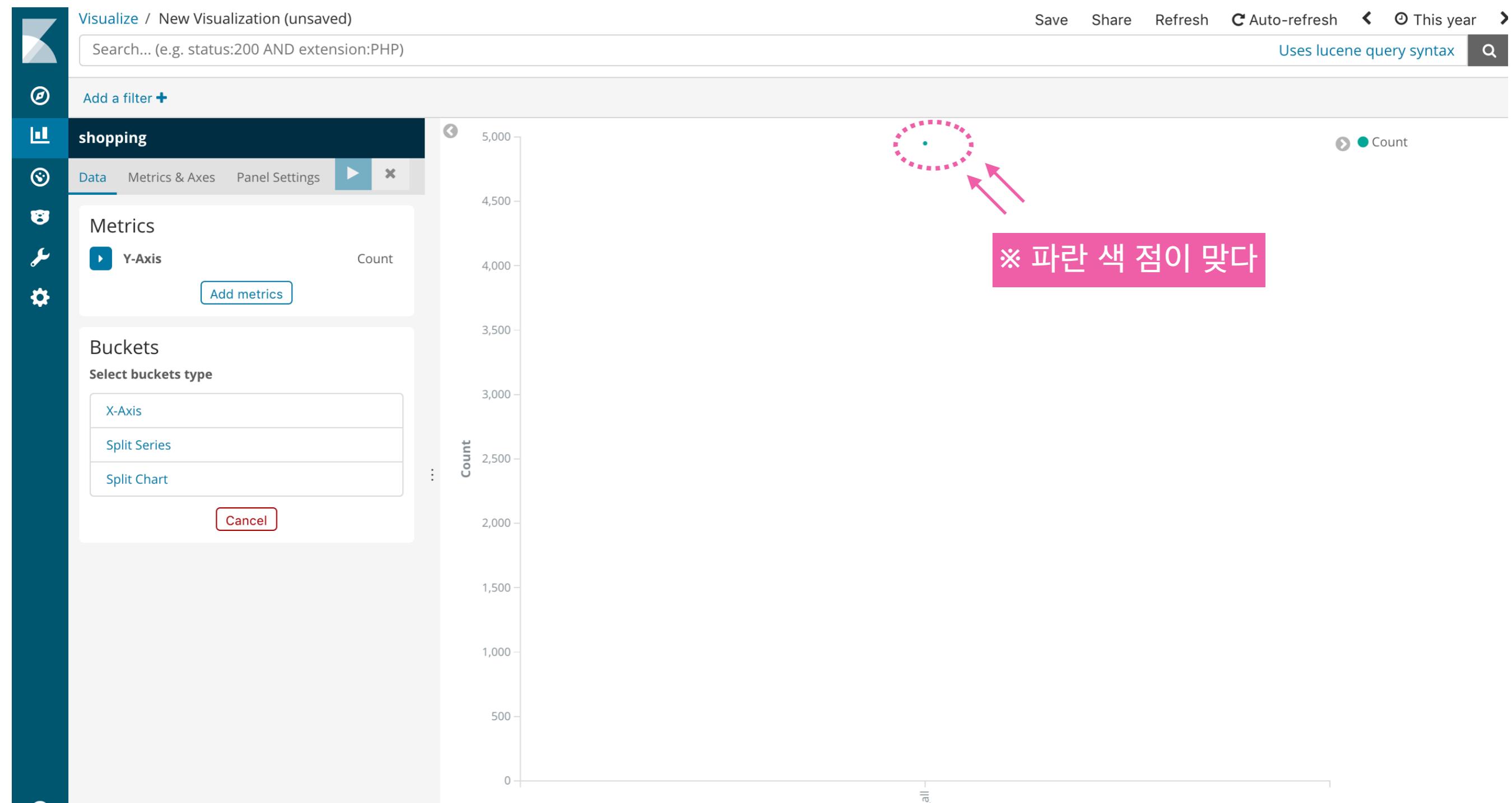
## Visualize 과정

- 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

## 무얼 누르든 aggregation을 선택해야 한다

This screenshot shows the Kibana Data panel for a 'shopping' visualization. The 'Metrics' section is active, displaying a Y-axis dropdown set to 'Count'. Below it is a 'Custom Label' input field and an 'Add metrics' button. A pink box highlights the 'Aggregation' dropdown, which is also set to 'Count'. At the bottom left is an 'Advanced' link, and at the bottom right is another 'Add metrics' button.

This screenshot shows the Kibana Data panel for a 'shopping' visualization. The 'Metrics' section is active, showing a Y-axis dropdown set to 'Count' and an 'Add metrics' button. Below it is a 'Buckets' section with an X-axis dropdown set to '주문시간 per week' and an 'Add sub-buckets' button. A pink box highlights the 'Aggregation' dropdown in the Buckets section, which is currently empty ('Select an aggregation').

This screenshot shows the Kibana Data panel for a 'shopping' visualization. The 'Metrics' section is active, showing a Y-axis dropdown set to 'Count' and an 'Add metrics' button. Below it is a 'Buckets' section with an X-axis dropdown set to '주문시간 per week' and a 'Split Series' option. A pink box highlights the 'Sub Aggregation' dropdown in the Buckets section, which is currently empty ('Select an aggregation').

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

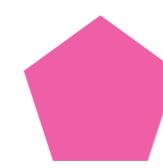
예를 들어 다음과 같은 도형이 있다고 하자



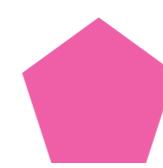
위의 도형을 여러 그룹으로 나눠야 한다면 어떻게 할 수 있을까?

어떤 방법을 택하든 가장 먼저 하는 작업은 기준을 정하는 것이다

내각의 합



색



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

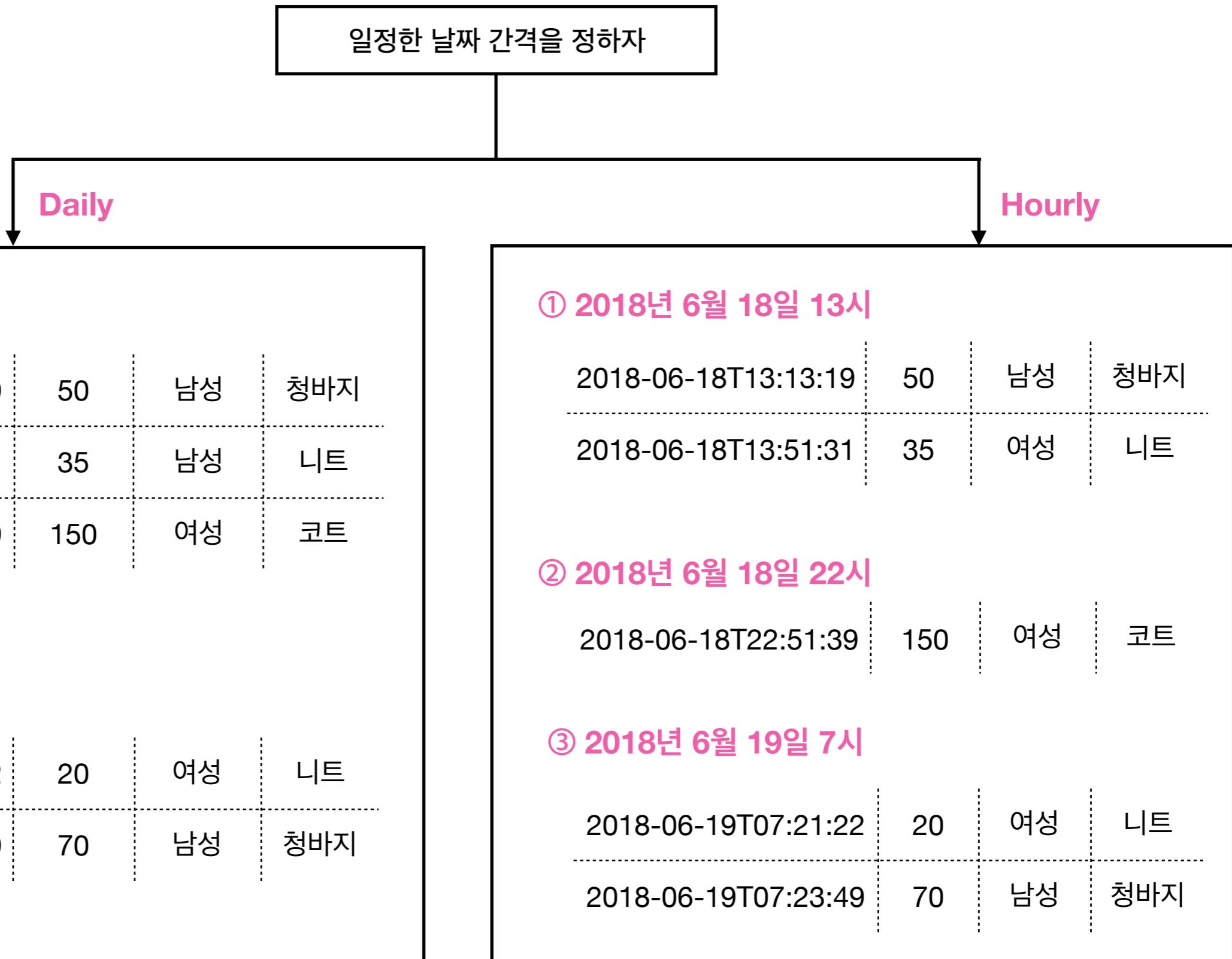
## Kibana가 지원하는 Bucket Aggregation

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	(Background 대비) Foreground에서 특별한 값	서울에서 “특별한” 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
2018-06-18T13:13:19	50	남성	청바지
2018-06-18T13:51:31	35	남성	니트
2018-06-18T22:51:39	150	여성	코트
2018-06-19T07:21:22	20	여성	니트
2018-06-19T07:23:49	70	남성	청바지

## Date Histogram Aggregation으로 Bucket을 생성하자



## Terms Aggregation으로 Bucket을 생성하자

(Categorical) Field를 정하자

성별

분류

① 성별 = 남성

2018-06-18T13:13:19	50	남성	청바지
2018-06-18T13:51:31	35	남성	니트
2018-06-19T07:23:49	70	남성	청바지

② 성별 = 여성

2018-06-19T07:21:22	20	여성	니트
2018-06-18T22:51:39	150	여성	코트

① 분류 = 청바지

2018-06-18T13:13:19	50	남성	청바지
2018-06-19T07:23:49	70	남성	청바지

② 분류 = 코트

2018-06-18T22:51:39	150	여성	코트
---------------------	-----	----	----

③ 분류 = 니트

2018-06-19T07:21:22	20	여성	니트
2018-06-18T13:51:31	35	여성	니트

## 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

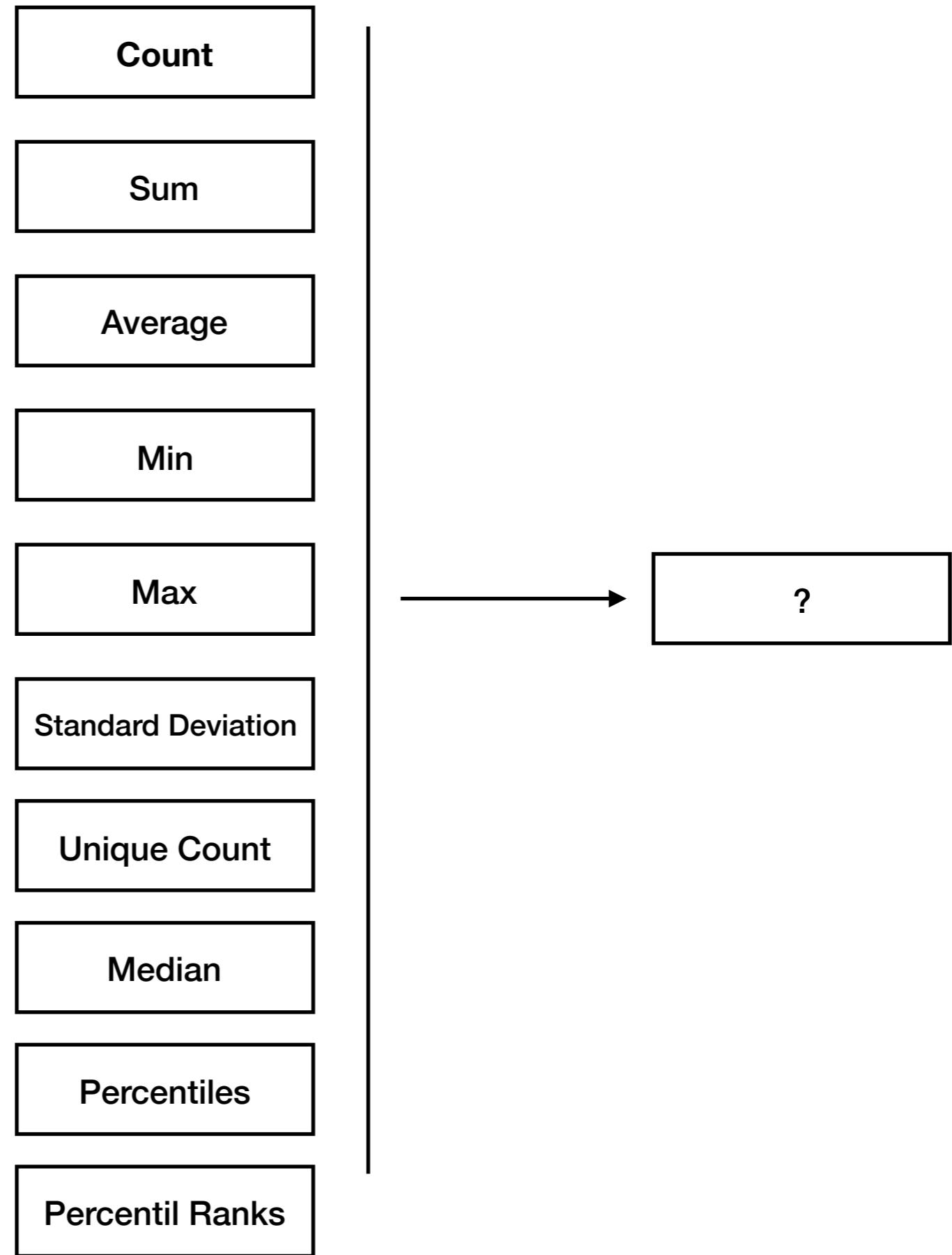
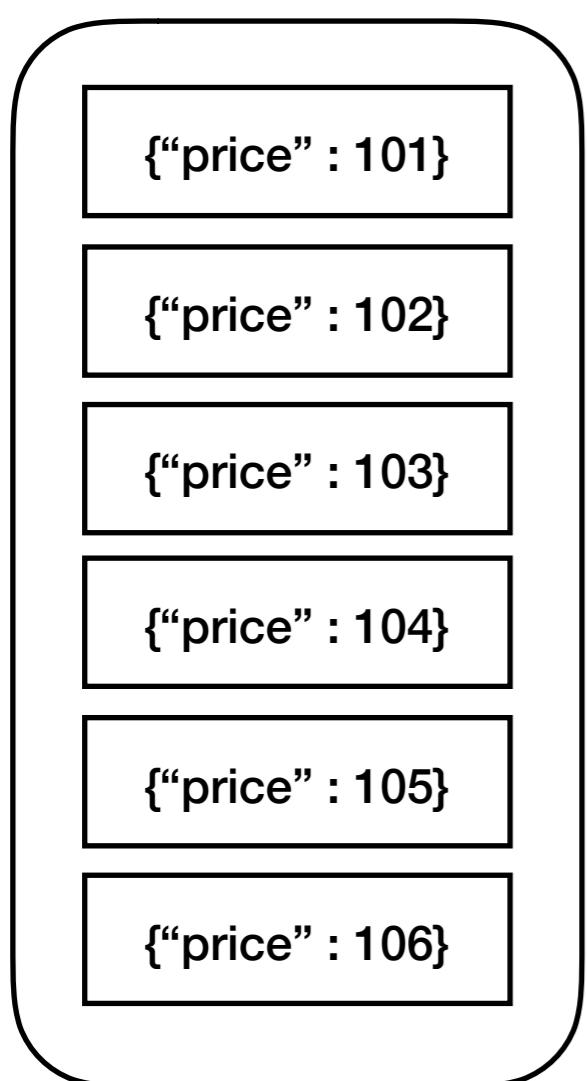
다시 해석해보면 “**파란색 2개, 녹색 1개, 자주색 1개**”이다. 유의미한가?

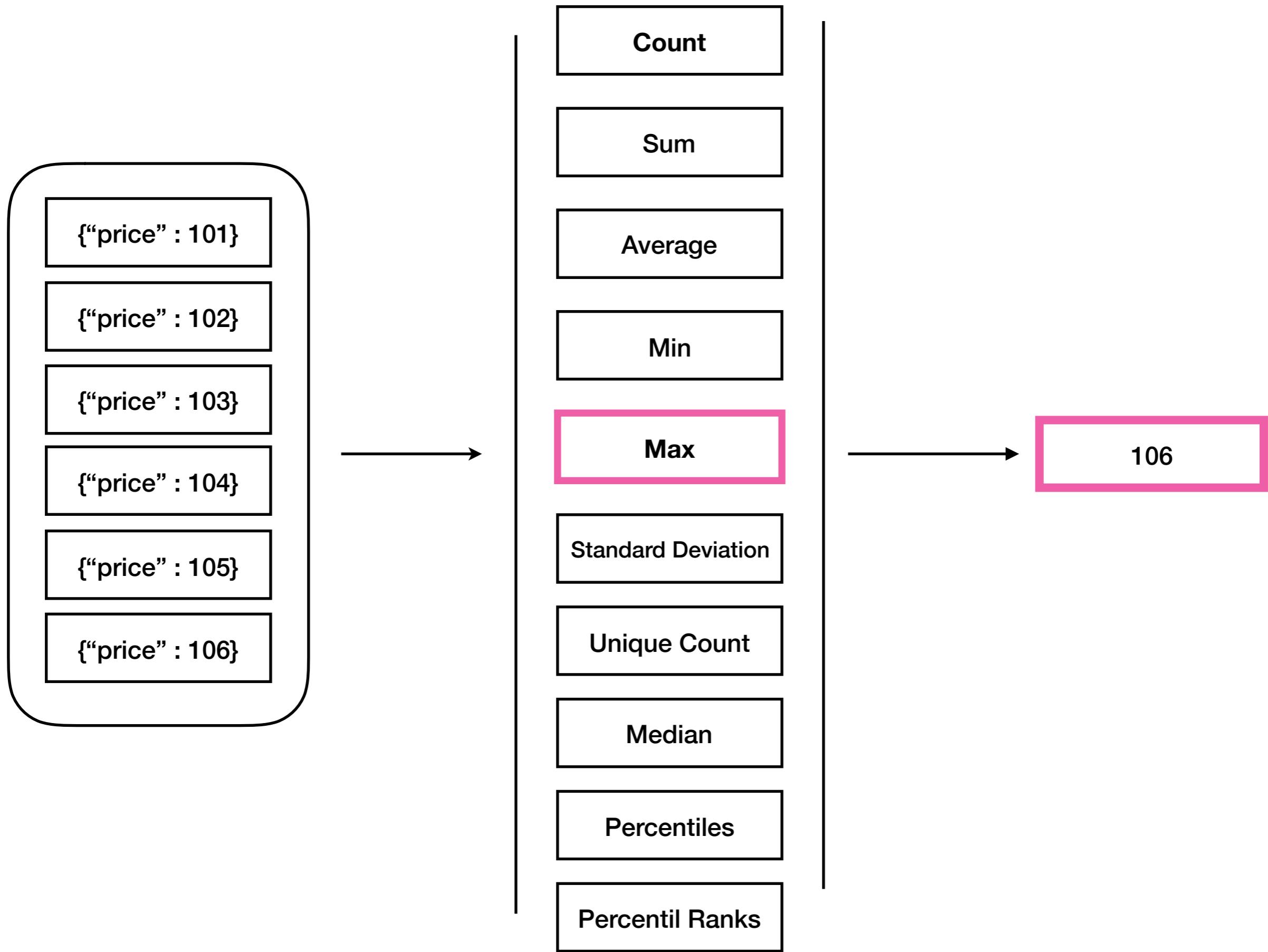
**Metrics Aggregation** = (Bucket 내의 Documents 단위로) 특정 연산을 수행하는 Aggregation

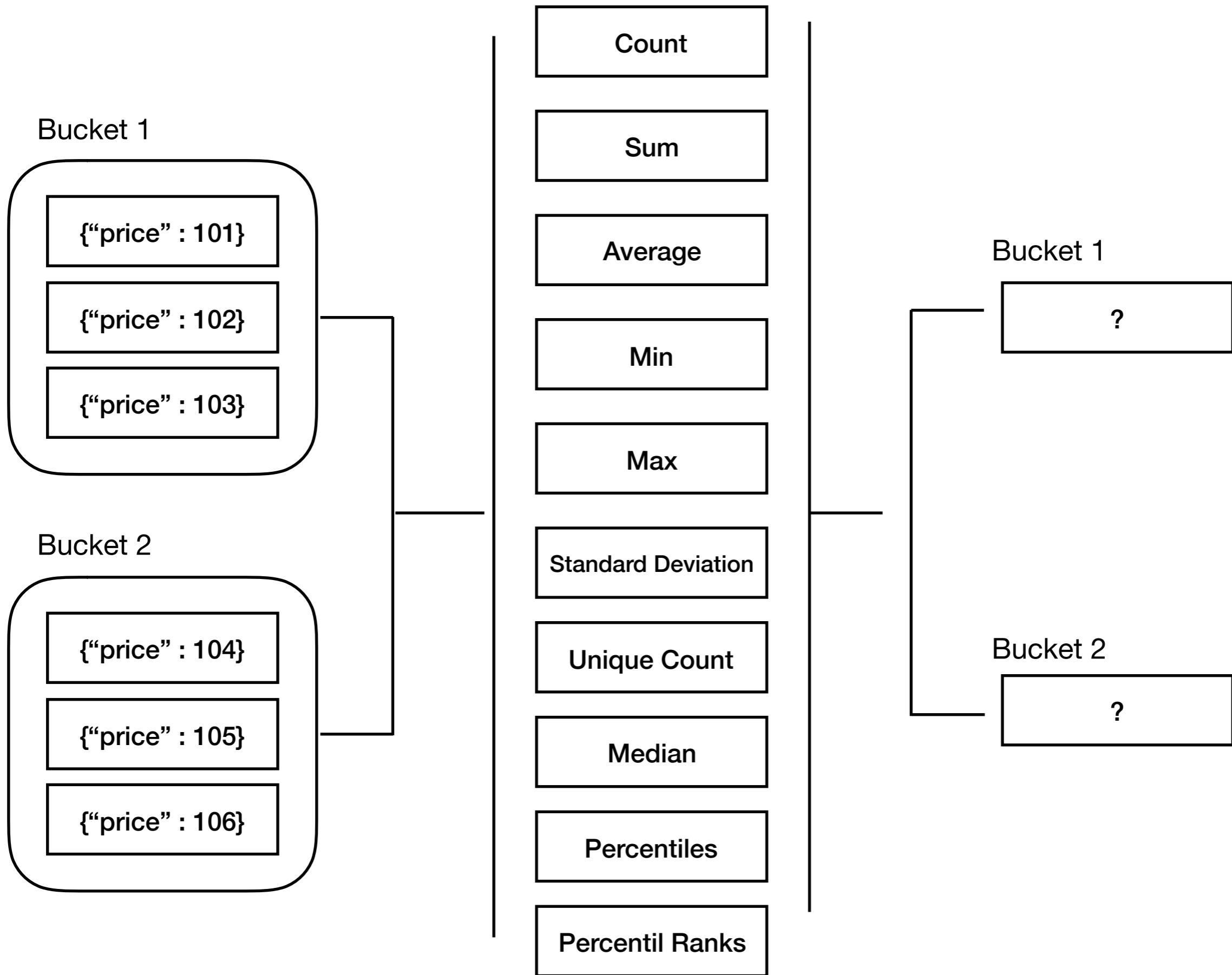
## Kibana가 지원하는 Metrics Aggregation

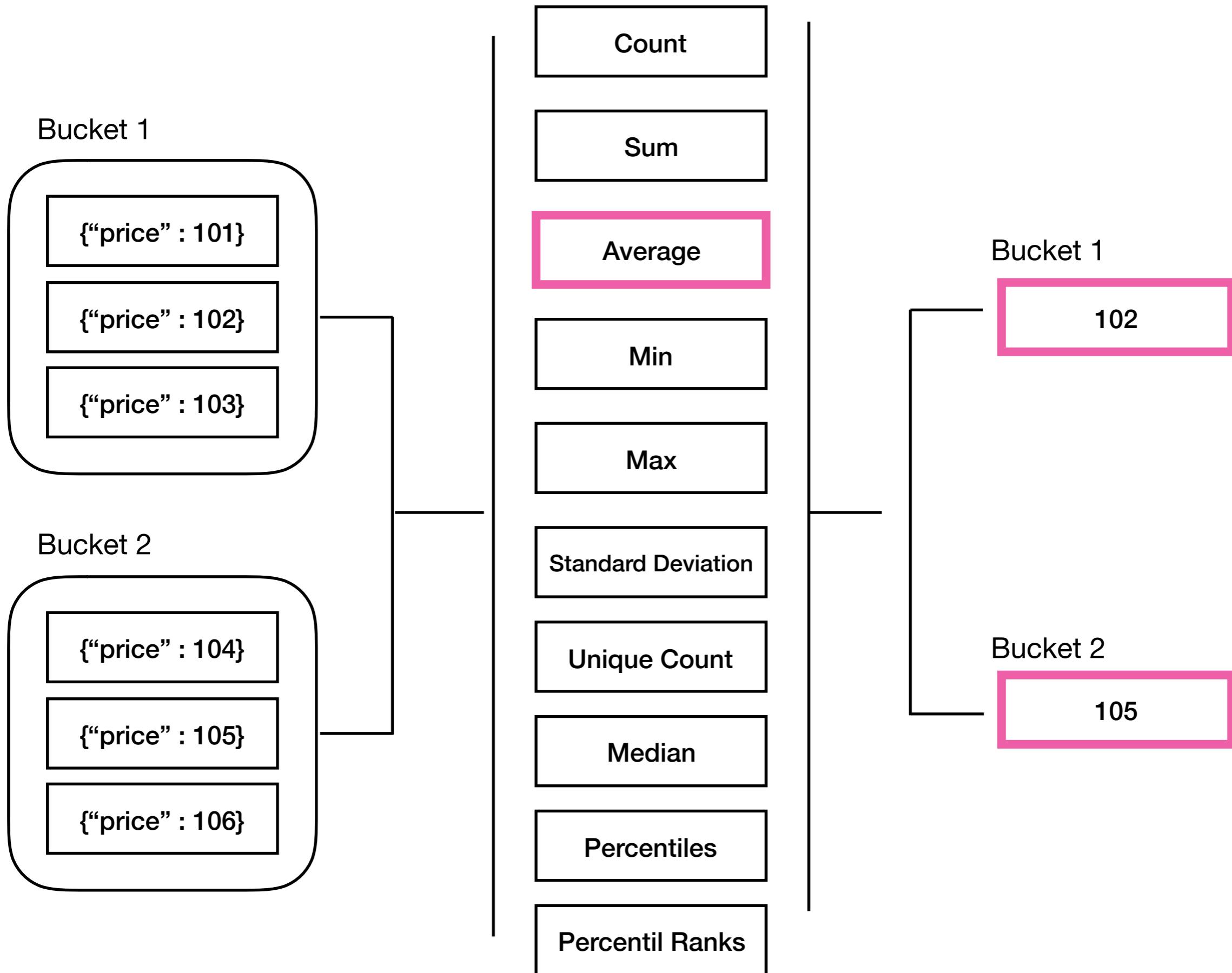
종류	적용 가능 Type	상세
Value Count	All	(Bucket 내) Document의 개수 계산
Avg	Number	(Bucket 내) Document의 특정 Field Values의 평균 계산
Sum	Number	(Bucket 내) Document의 특정 Field Values의 합 계산
Min/Max	Number	(Bucket 내) Document의 특정 Field Values의 최소/최대 계산
Extended Stats	Number	(Bucket 내) Document의 특정 Field Values의 기초 통계값 계산
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 ⏵

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

Add a filter +

**shopping**

Data Options ▶ ×

**Metrics** 👉

Metric Count Add metrics

**Buckets** 👉

Select buckets type

Split Rows 👉

Split Table 👉

Cancel

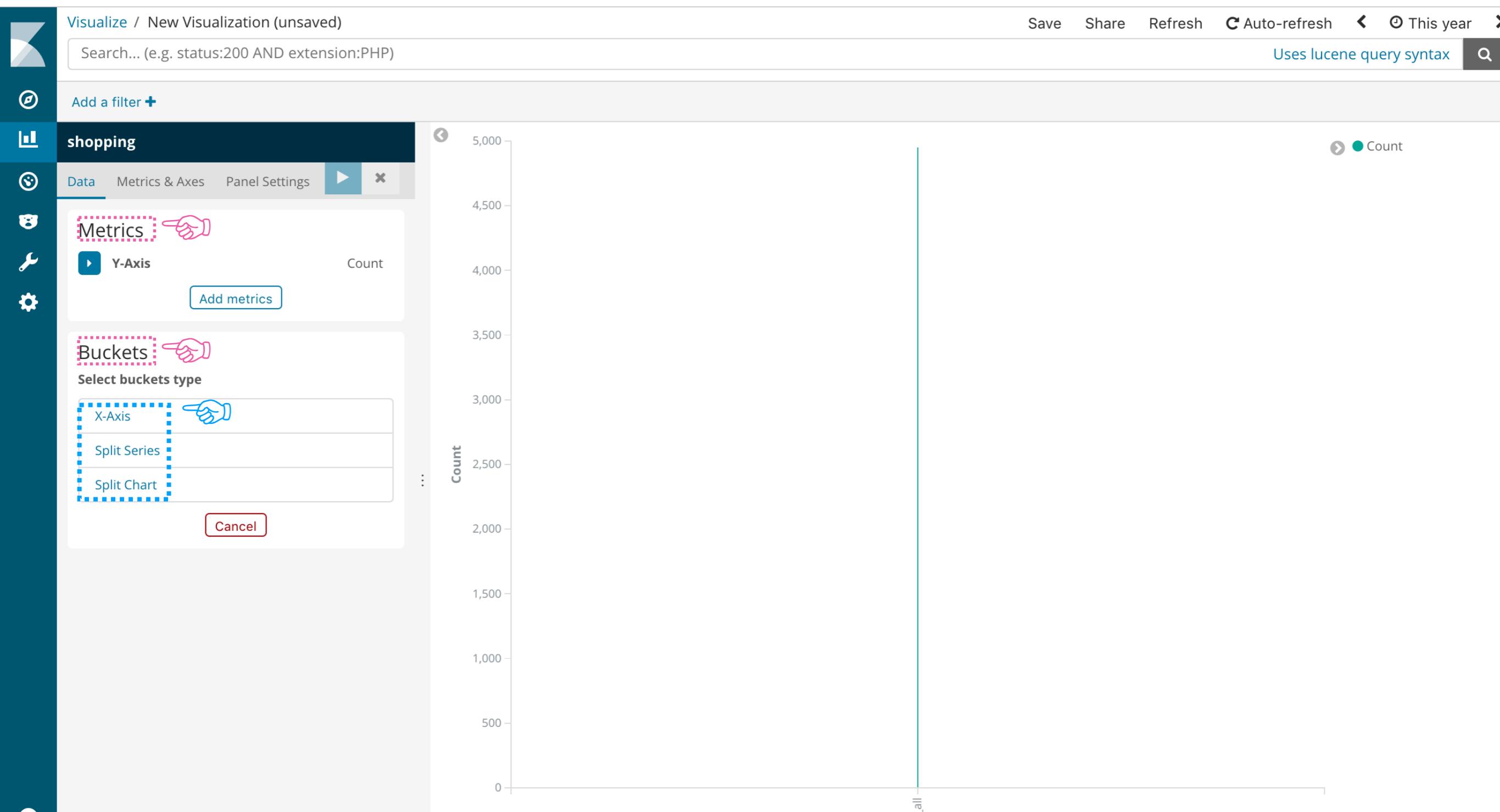
Count ◀

4,951

Export: Raw ⬇️ Formatted ⬇️

The screenshot shows a visualization interface with a sidebar on the left containing various icons and a main workspace on the right. In the main workspace, there is a search bar at the top with placeholder text '(e.g. status:200 AND extension:PHP)'. To the right of the search bar are buttons for 'Save', 'Share', 'Refresh', 'Auto-refresh' (with a lightning bolt icon), and date range selection ('This year'). Below the search bar is a 'Count' section displaying the number '4,951'. At the bottom of the workspace, there are 'Export' options for 'Raw' and 'Formatted' data, each with a download icon.

# 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

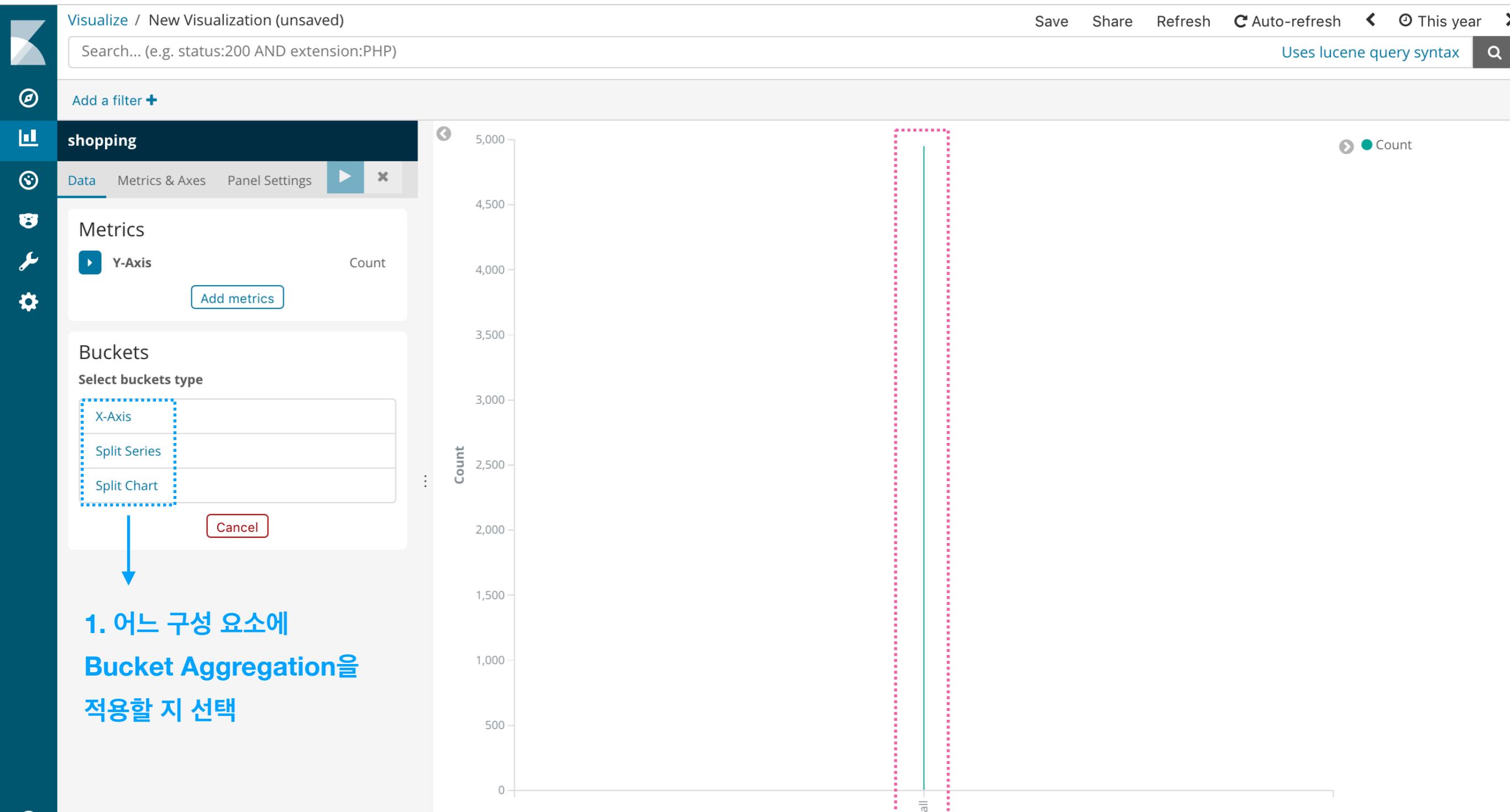
예: 전국 학생들의 지역별 평균 키를 구한다고 하자

**키의 평균을 구하는 작업** : metrics

**학생들을 지역별로 나누는 작업** : buckets

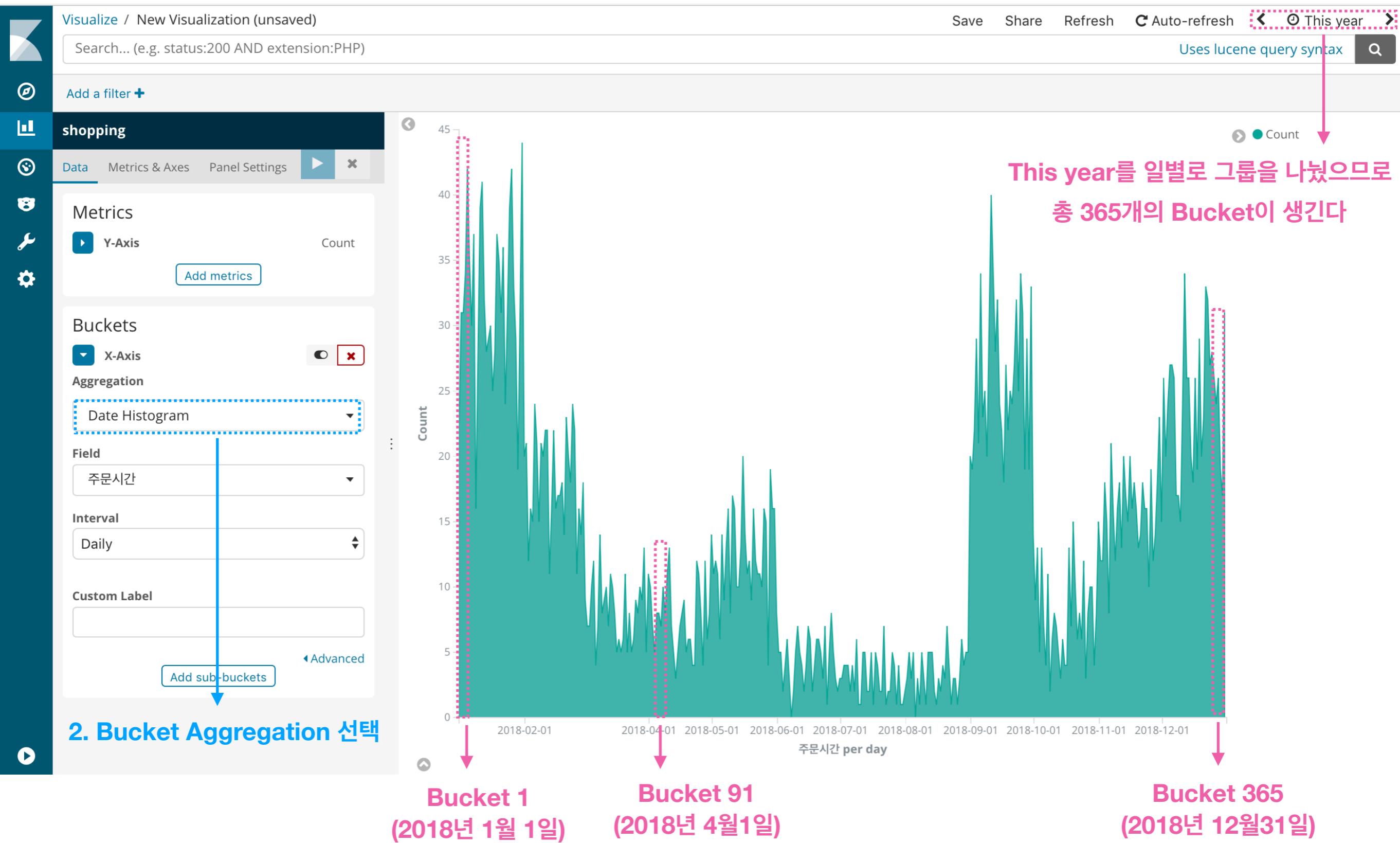
## **2. 개별적 구성요소는 Visualization Type마다 상이할 수 있다**

# 대표적인 bucket type 몇 개를 살펴보자 X-Axis Before

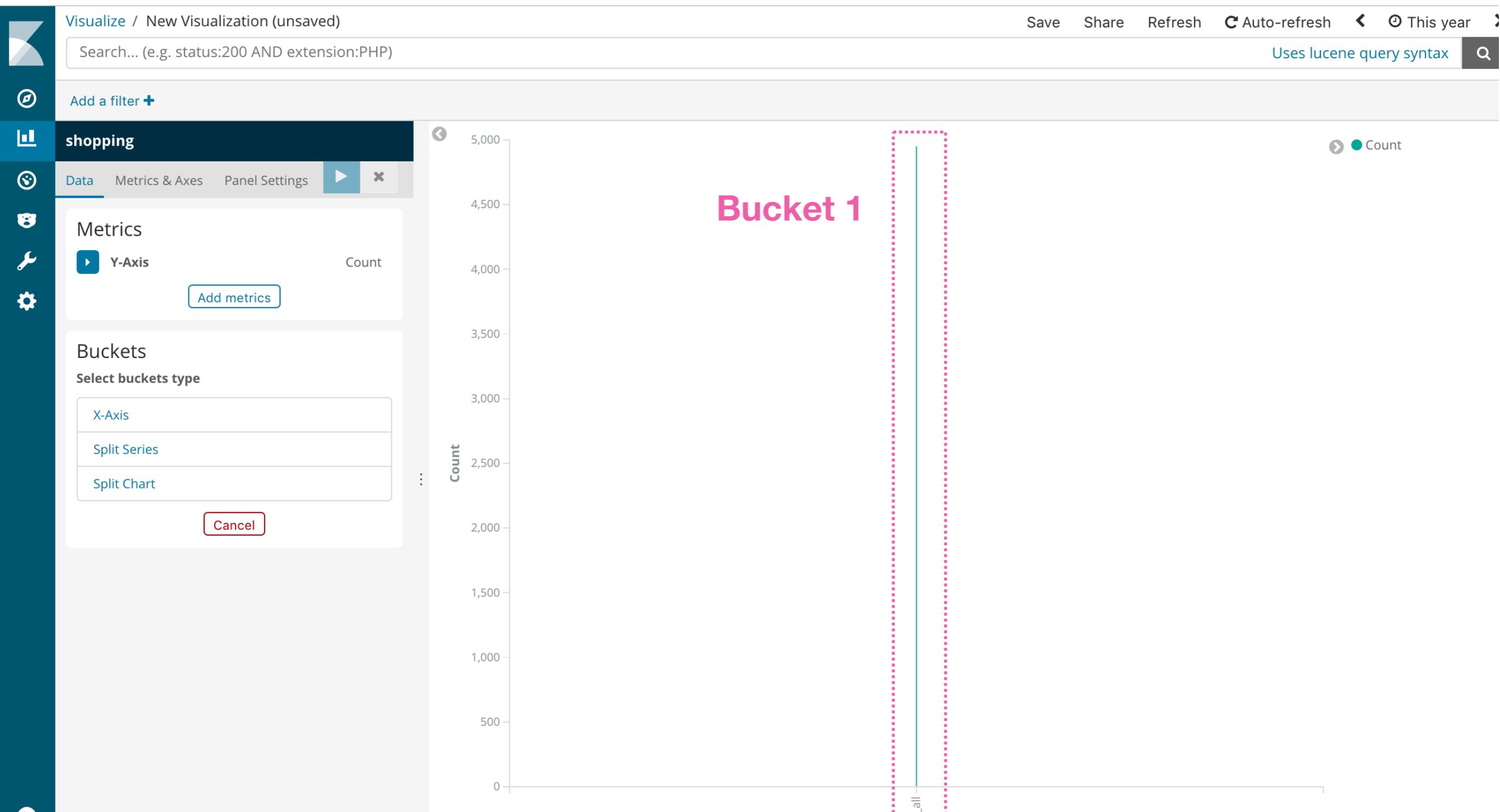


# 대표적인 bucket type 몇 개를 살펴보자 X-Axis

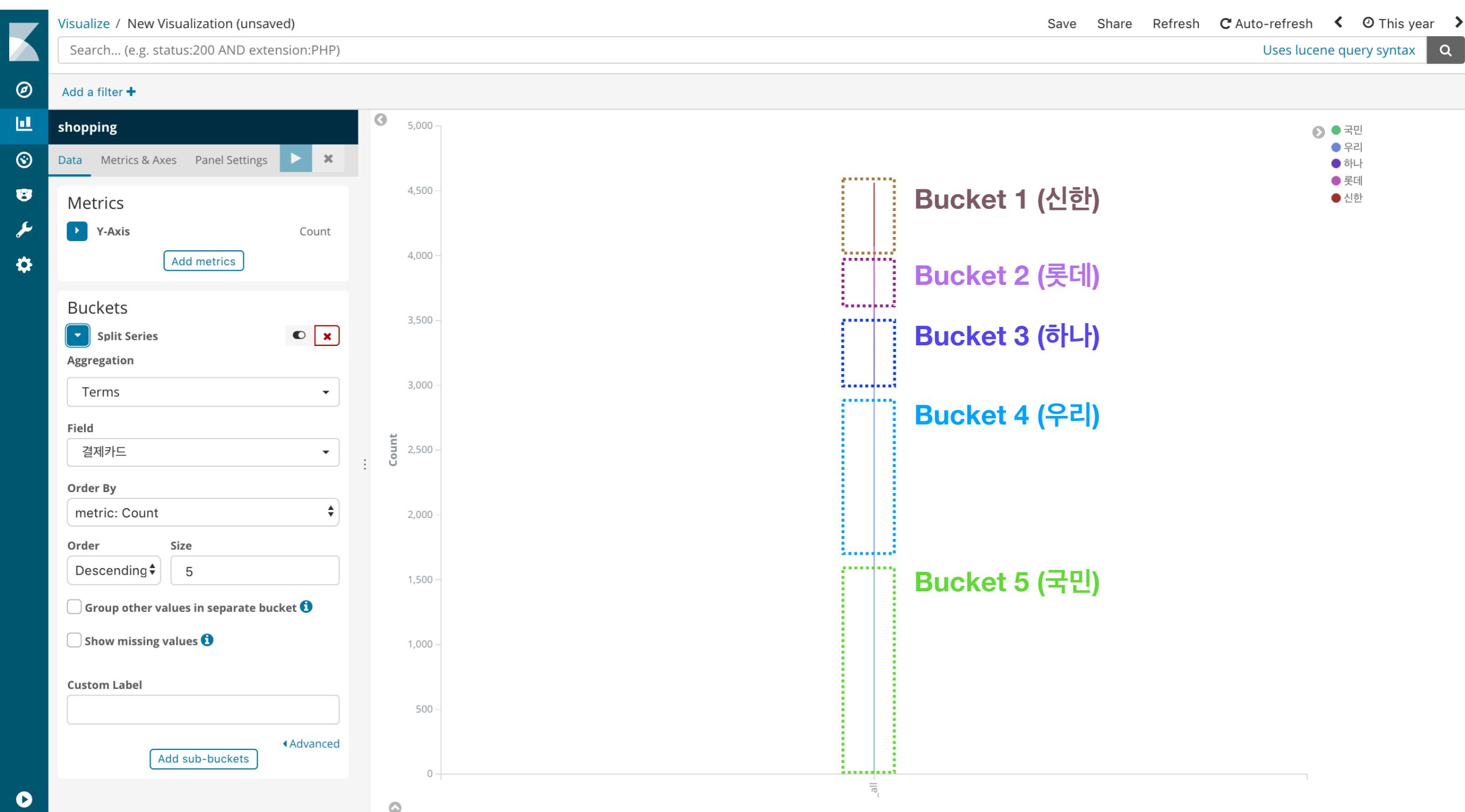
After



# 대표적인 bucket type 몇 개를 살펴보자 Split Series Before

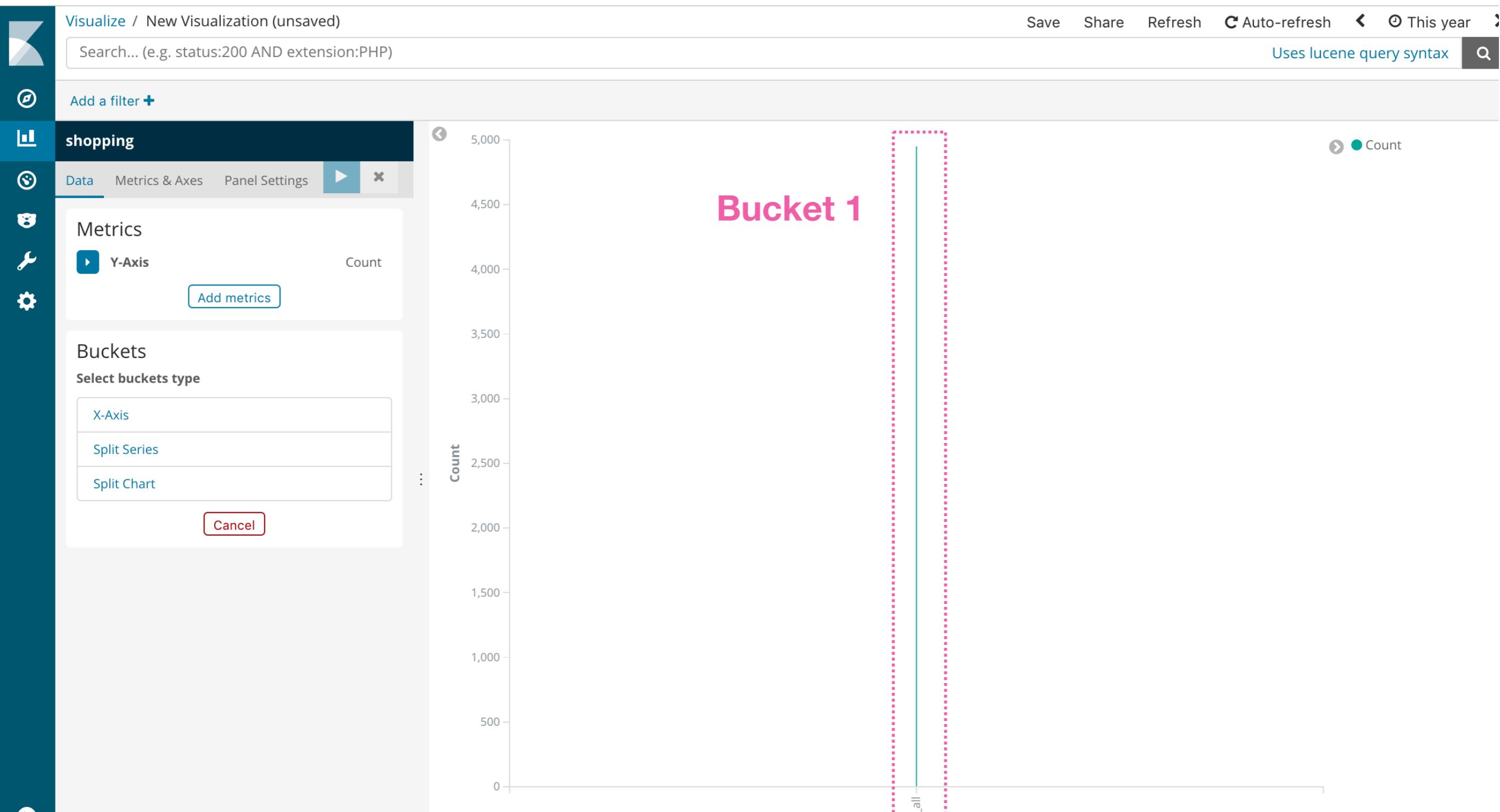


# 대표적인 bucket type 몇 개를 살펴보자 Split Series After



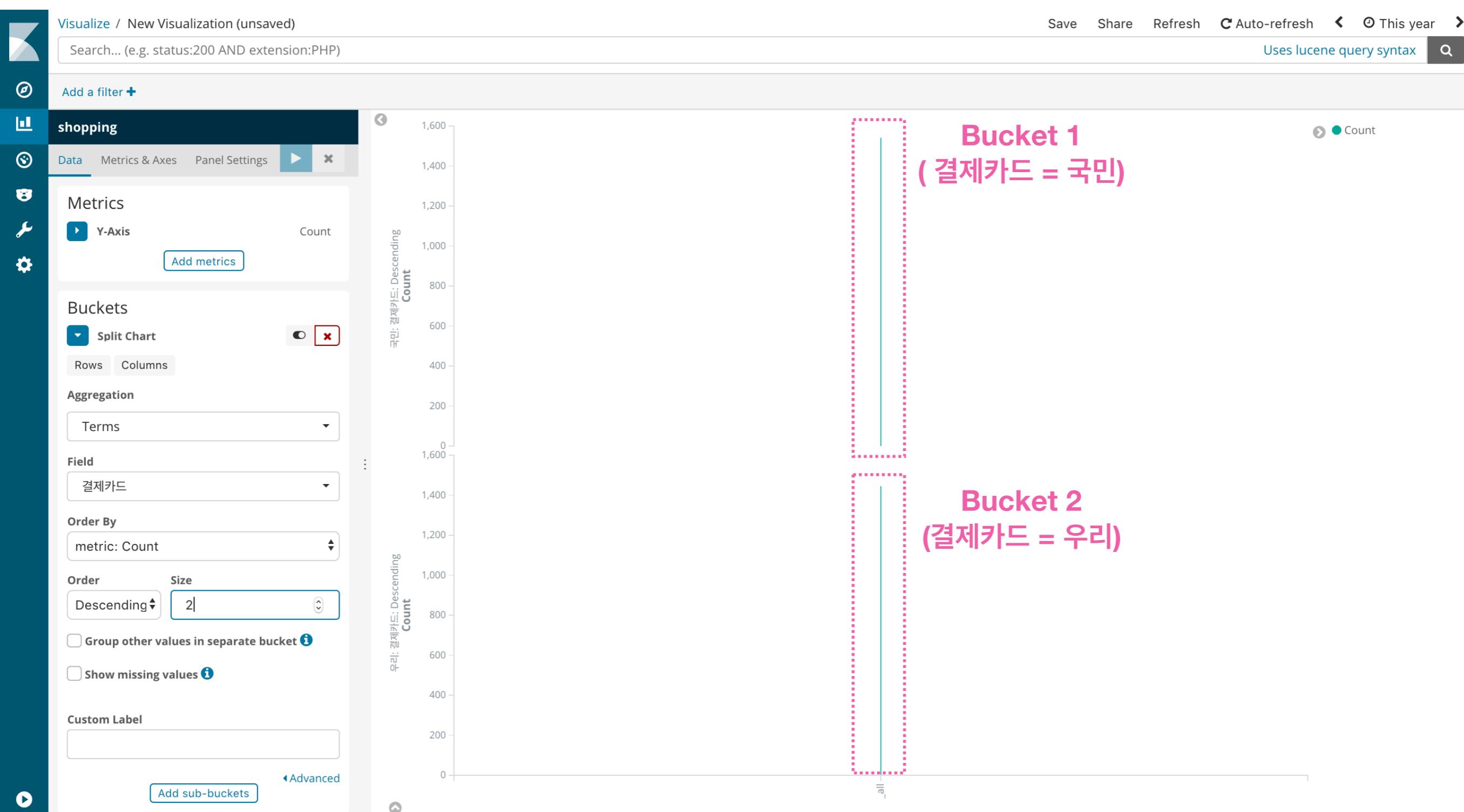
# 대표적인 bucket type 몇 개를 살펴보자 Split Chart

Before



# 대표적인 bucket type 몇 개를 살펴보자 Split Chart

After



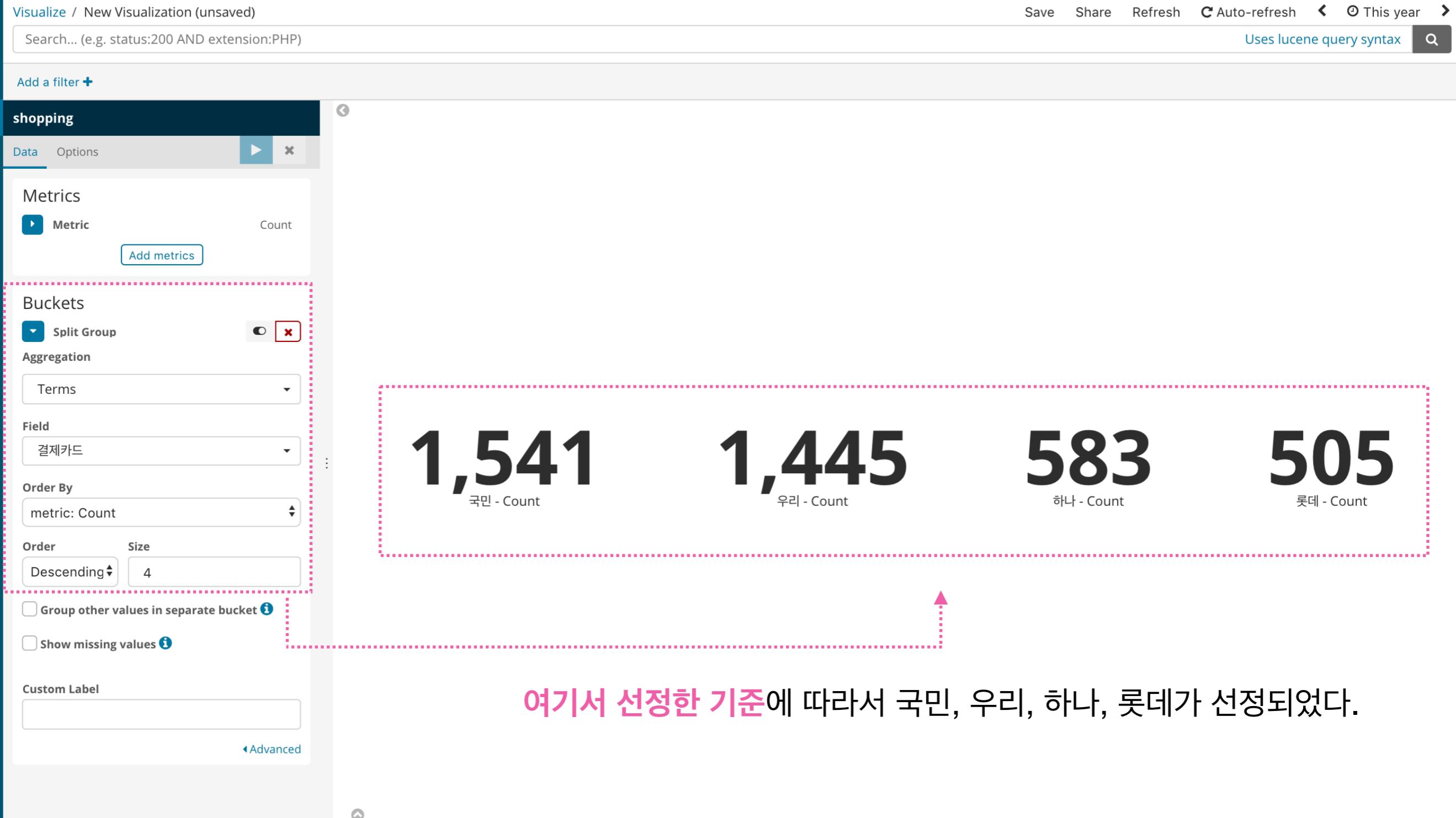
보통은 이 작업의 반복이지만 **Term Aggregation**으로  
Bucket을 나눌 경우 한 단계 더 고려해야한다

- 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 Aggregation을 수행한다**



shopping

Data Options

### Metrics

Metric Count

### Buckets

#### Aggregation

Terms

Field 결제카드

Order By metric: Count

Order Descending 4

Group other values in separate bucket

Show missing values

Custom Label

1. 결제카드로 Bucket을 구분해서...

국민	하나	신한	롯데	시티	우리

2. Bucket 별 Count를 구하고...

국민	하나	신한	롯데	시티	우리
1541	583	487	505	229	1445

3. Count가 큰 순으로 정렬해서...

국민	우리	하나	롯데
1541	1445	583	505

4. 상위 4개를 선정해라

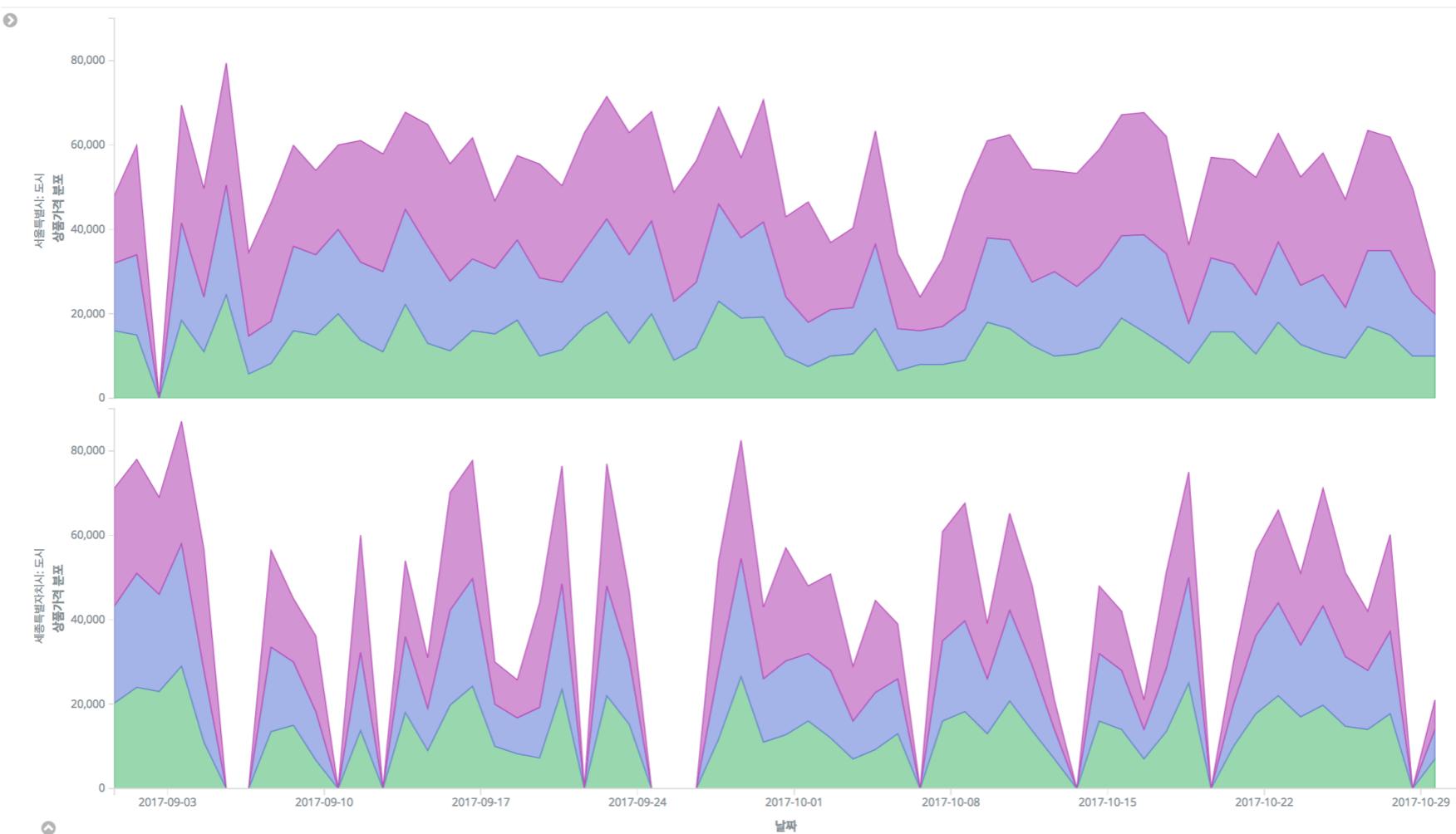
국민	우리	하나	롯데	신한	시티
1541	1445	583	505	487	229

## 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

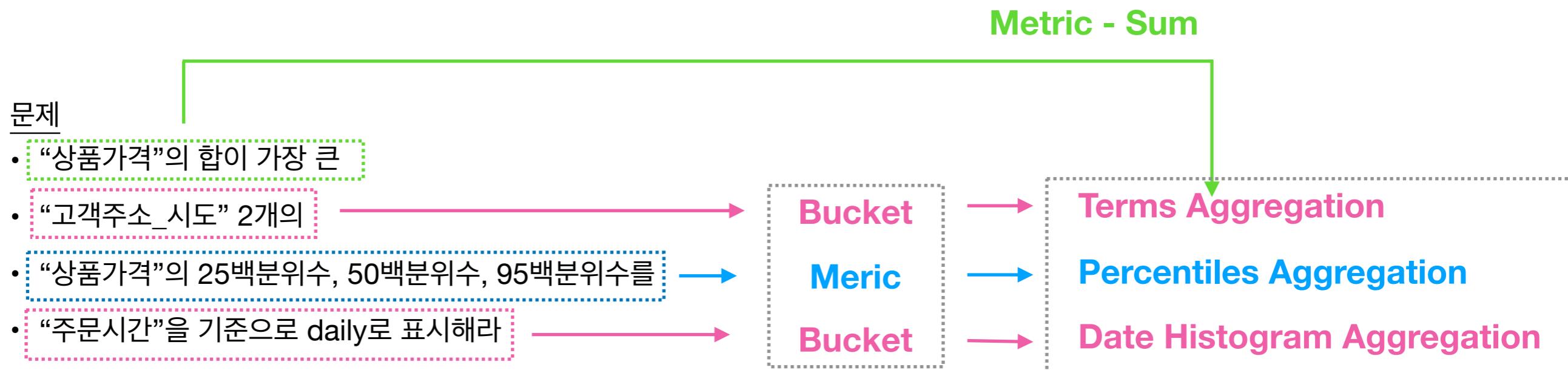
- “상품가격”의 합이 가장 큰 Bucket
- “고객주소\_시도” 2개의 Meric
- “상품가격”의 25백분위수, 50백분위수, 95백분위수를 Bucket
- “주문시간”을 기준으로 daily로 표시해라 Bucket

## metrics와 buckets 내에서 사용할 aggregation을 선택한다

### 문제

- “상품가격”의 합이 가장 큰
  - “고객주소\_시도” 2개의
  - “상품가격”의 25백분위수, 50백분위수, 95백분위수를
  - “주문시간”을 기준으로 daily로 표시해라
- 
- The diagram illustrates the mapping of user requirements to Elasticsearch aggregation types. On the left, four user requirements are listed, each enclosed in a colored box: a green box for the first requirement, a pink box for the second, a blue box for the third, and another pink box for the fourth. Arrows point from these requirements to a central column of three boxes labeled 'Bucket', 'Metric', and 'Bucket'. From the central column, arrows point to the right to three corresponding aggregation types: 'Terms Aggregation' (in pink), 'Percentiles Aggregation' (in blue), and 'Date Histogram Aggregation' (in pink). The 'Metric' box is positioned between the first and second arrows in the central column.
- |        |   |                            |
|--------|---|----------------------------|
| Bucket | → | Terms Aggregation          |
| Metric | → | Percentiles Aggregation    |
| 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. On the left is a vertical toolbar with icons for file operations, cell types, and help. The main area has a title bar "Visualize / [shopping] markdown". A pink box highlights the "Markdown" cell 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)
```

A pink border surrounds the entire cell content. At the top right of the cell are buttons for "② 실행" (Run) and other controls. At the bottom right of the cell is the text "① 적당한 내용을 입력하자" with a small crown icon.

# Metric

42

Metric

- KPI 같은 지표를 숫자 형태로 시각화하고 싶을 때 사용

## Metric Object

4,951  
Count

해석

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

## Metric Configuration - Count

The screenshot shows the 'Metric Configuration - Count' dialog. At the top right, there is a time range selector with arrows and the text 'This year'. Below it, a button labeled '④ 실행' (Run) is highlighted with a pink border. On the left, a sidebar has 'shopping' selected. The main area is divided into sections: 'Metrics' (with 'Metric' dropdown), 'Aggregation' (with 'Count' dropdown highlighted with a pink border), 'Custom Label' (empty input field), and 'Buckets' (with 'Select buckets type' dropdown containing 'Split Group'). A 'Cancel' button is at the bottom right of the configuration area.

① Time Range를 This year로 설정

② Count aggregation 선택

③ bucket aggregation은 고정

④ 실행

## Metric Object

**16,990.507**  
Average 상품가격

해석

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

## Metric Configuration - Average

The screenshot shows the 'Metric Configuration - Average' interface. At the top right, there is a time range selector with arrows and the text 'This year'. Below it, a large button labeled 'shopping' has a pink border. To the right of the button, there are tabs for 'Data' and 'Options', a play button, and a close button. The main area is titled 'Metrics' and contains sections for 'Metric' (with a dropdown menu), 'Aggregation' (set to 'Average'), 'Field' (set to '상품가격'), 'Custom Label' (empty), and an 'Advanced' section with a 'Cancel' button. A pink box highlights the 'Buckets' section at the bottom, which contains a 'Select buckets type' dropdown set to 'Split Group'. To the right of the interface, five numbered steps are listed:

- ① Time Range를 This year로 설정
- ② Average aggregation 선택
- ③ Average aggregation을 적용할 Field 선택
- ④ bucket aggregation은 고정
- ⑤ 실행

## Metric Object

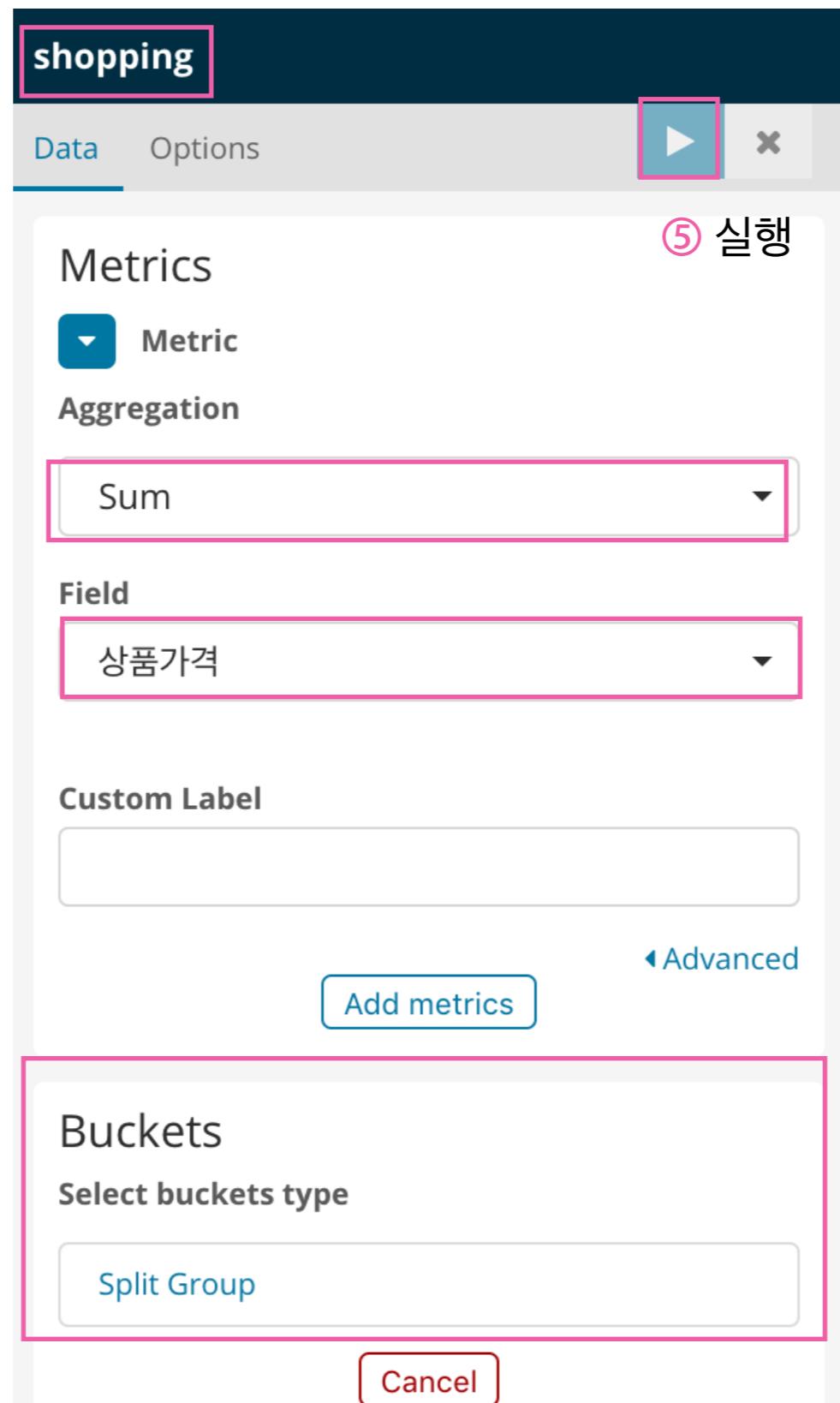
**84,120,000**  
Sum of 상품가격

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

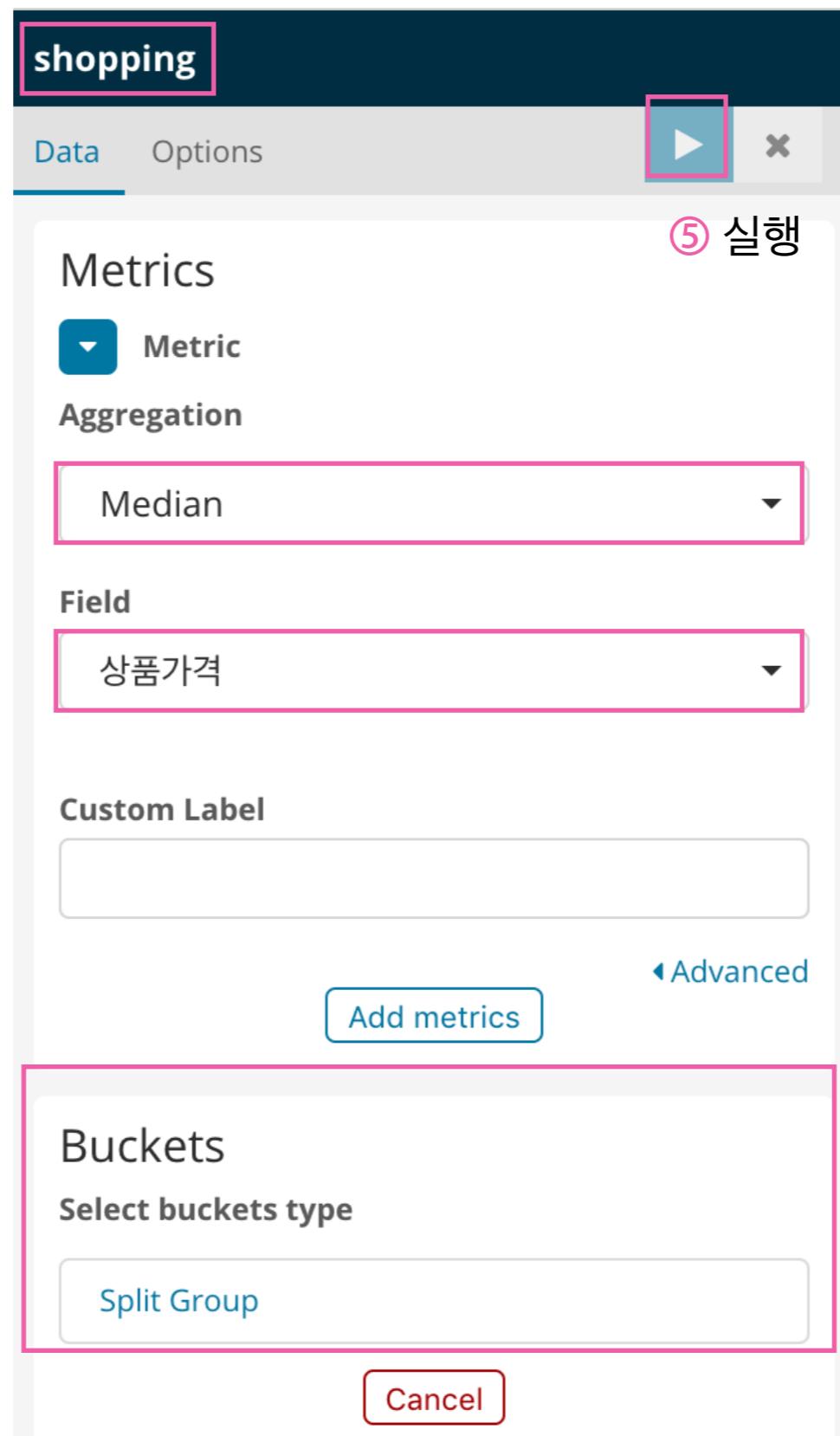
**17,000**  
50th percentile of 상품가격

### 해석

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

## Metric Configuration - Median

① (Metric 선택 후) shopping index 선택



◀ ⏪ This year ⏩ ▶

① Time Range를 This year로 설정

② Median aggregation 선택

③

Median aggregation 적용할 Field 선택

④ bucket aggregation은 고정

## Metric Object

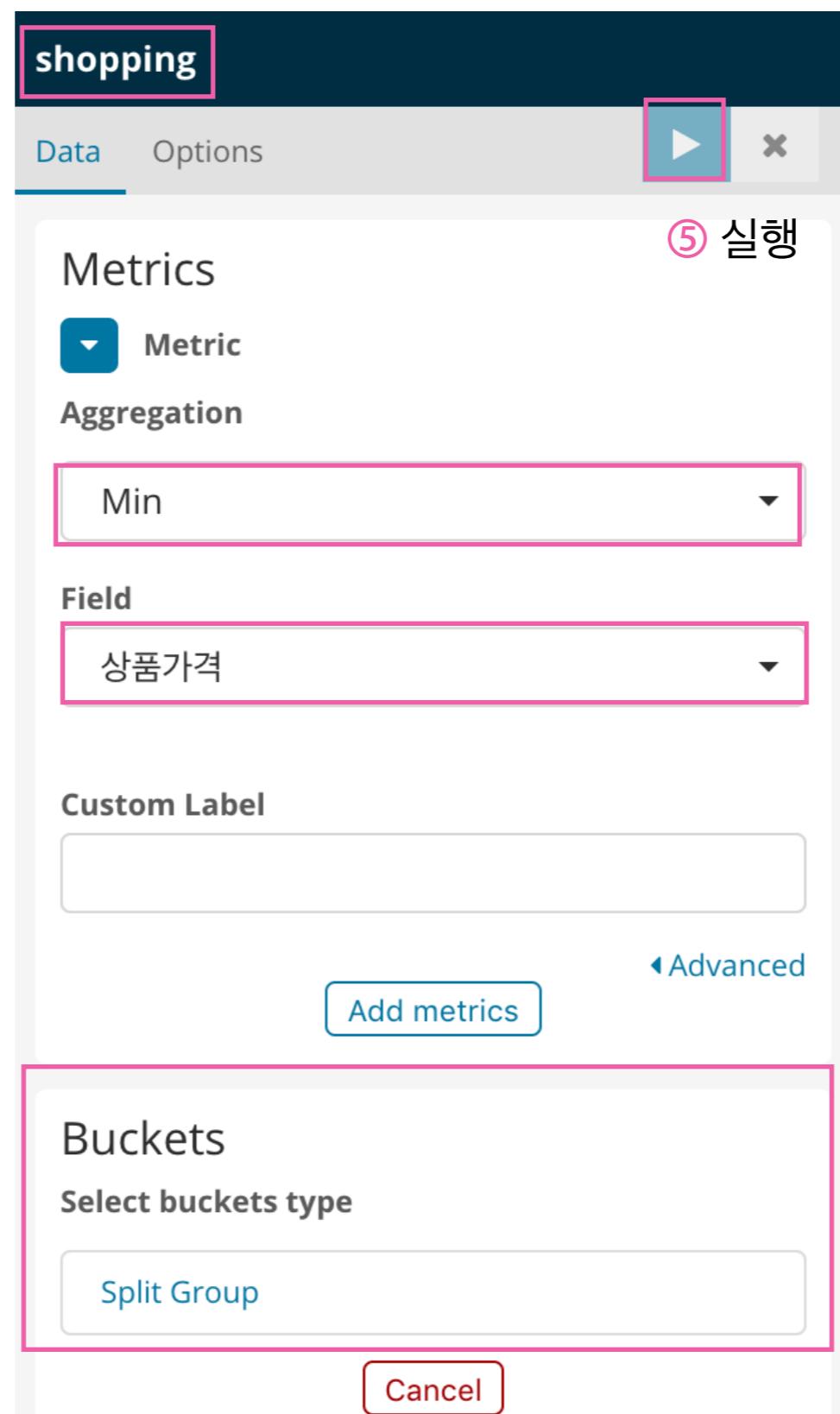
5,000  
Min 상품가격

### 해석

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

## Metric Configuration - Min/Max

① (Metric 선택 후) shopping index 선택



◀ ⏪ This year ⏩ ▶

① Time Range를 This year로 설정

⑤ 실행

② Min/Max aggregation 선택

③  
④ Min/Max aggregation을  
적용할 Field 선택

④ bucket aggregation은 고정

## Metric Object

# 25

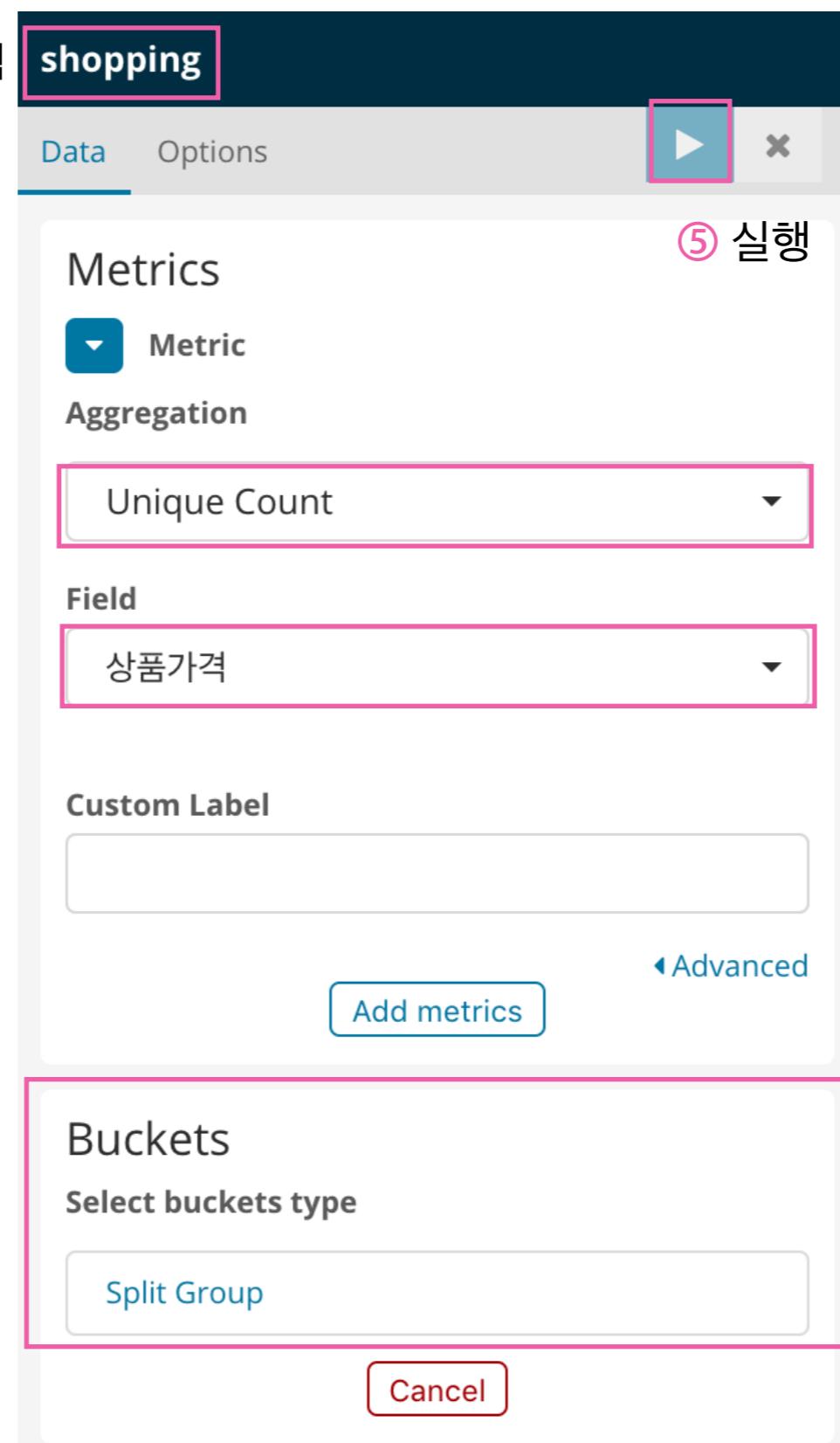
Unique count of 상품가격

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



② Time Range를 This year로 설정

③ Unique Count aggregation 선택

④ Unique Count aggregation을 적용할 Field 선택

⑤ bucket aggregation은 고정

## Metric Object

**5,000**

1st percentile of 상품가격

**17,000**

50th percentile of 상품가격

**29,000**

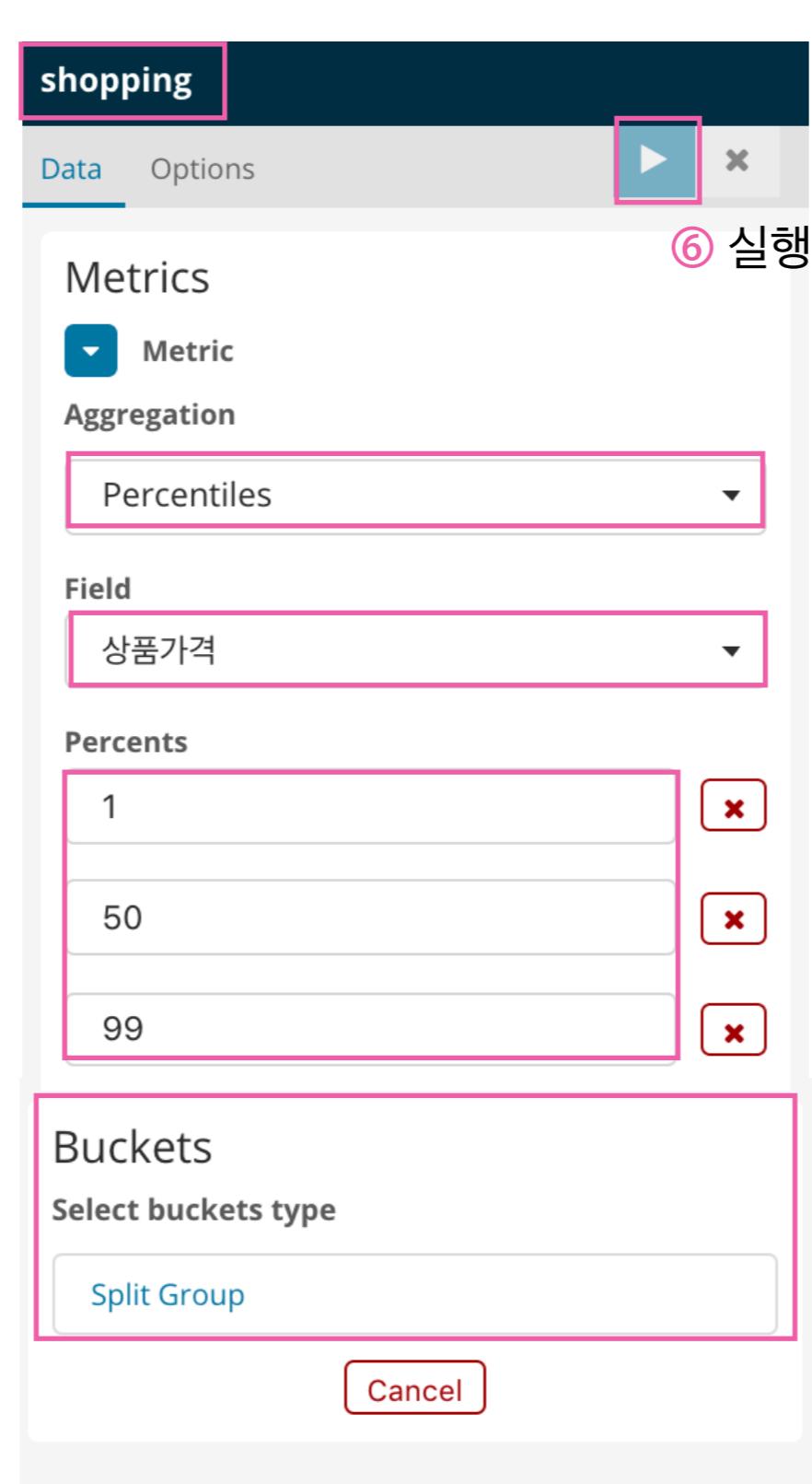
99th percentile of 상품가격

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

**22.844%**

Percentile rank 10,000 of "상품가격"

**42.658%**

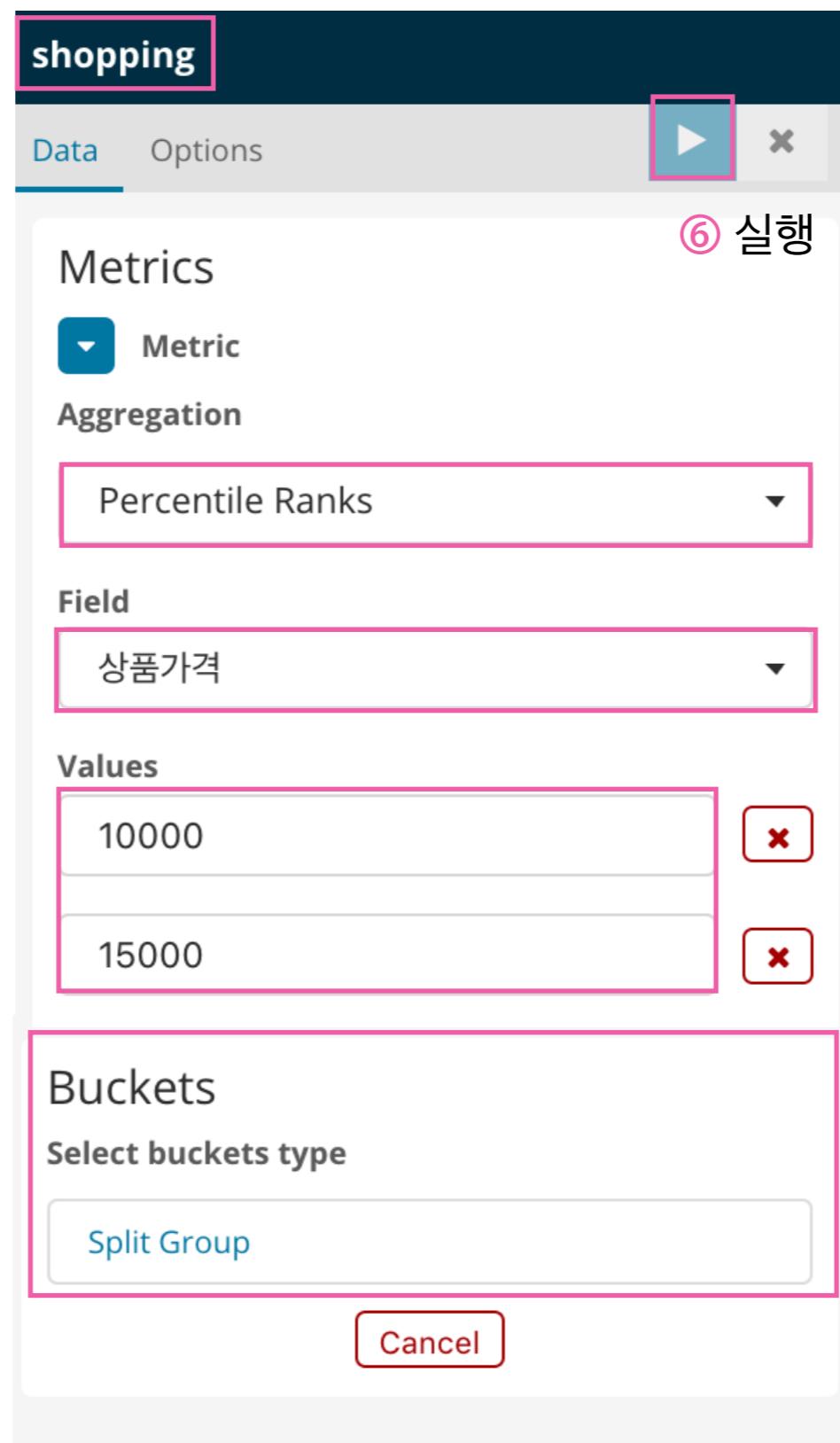
Percentile rank 15,000 of "상품가격"

### 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

# 3.4

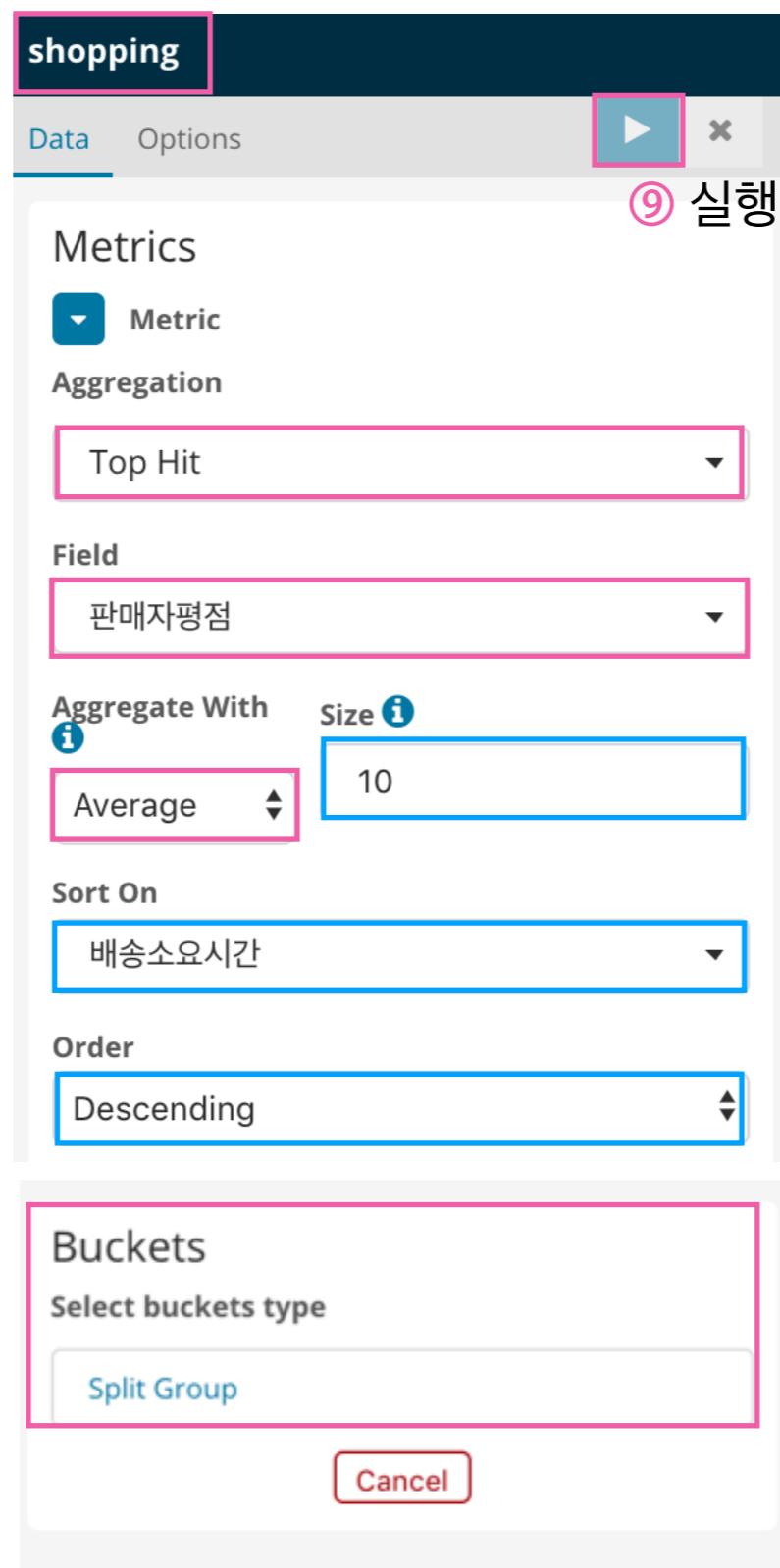
Last 10 판매자평점

### 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



② Top Hit aggregation 선택

③ Time Range를 This year로 설정

④

⑤ Documents 몇 개를 선택할 건지 입력

⑥

⑦ 어느 Field에  
Top Hit aggregation을 적용할지 선택

⑧ bucket aggregation은 고정

⑨ 실행

⑩ Documents를 정렬할 기준 Field 선택

⑪ Documents 정렬 방식 선택 (오름/내림)

## 예제 1) Metric

**284.203**

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

**5,113.372**

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

조건

- **nginx-\* index** 중에서
- “@timestamp” field 기준 “**2018년 6월 1일 ~ 2018년 6월 16일**” 사이 documents들의
- “**nginx.access.body\_sent.bytes**” field의 **50백분위수, 95백분위수**

## 예제 2) Metric

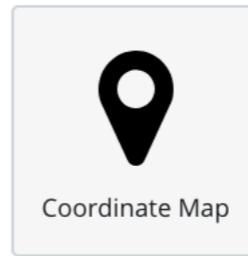
# Mac OS X, iOS

Last 2 nginx.access.user\_agent.os\_name

조건

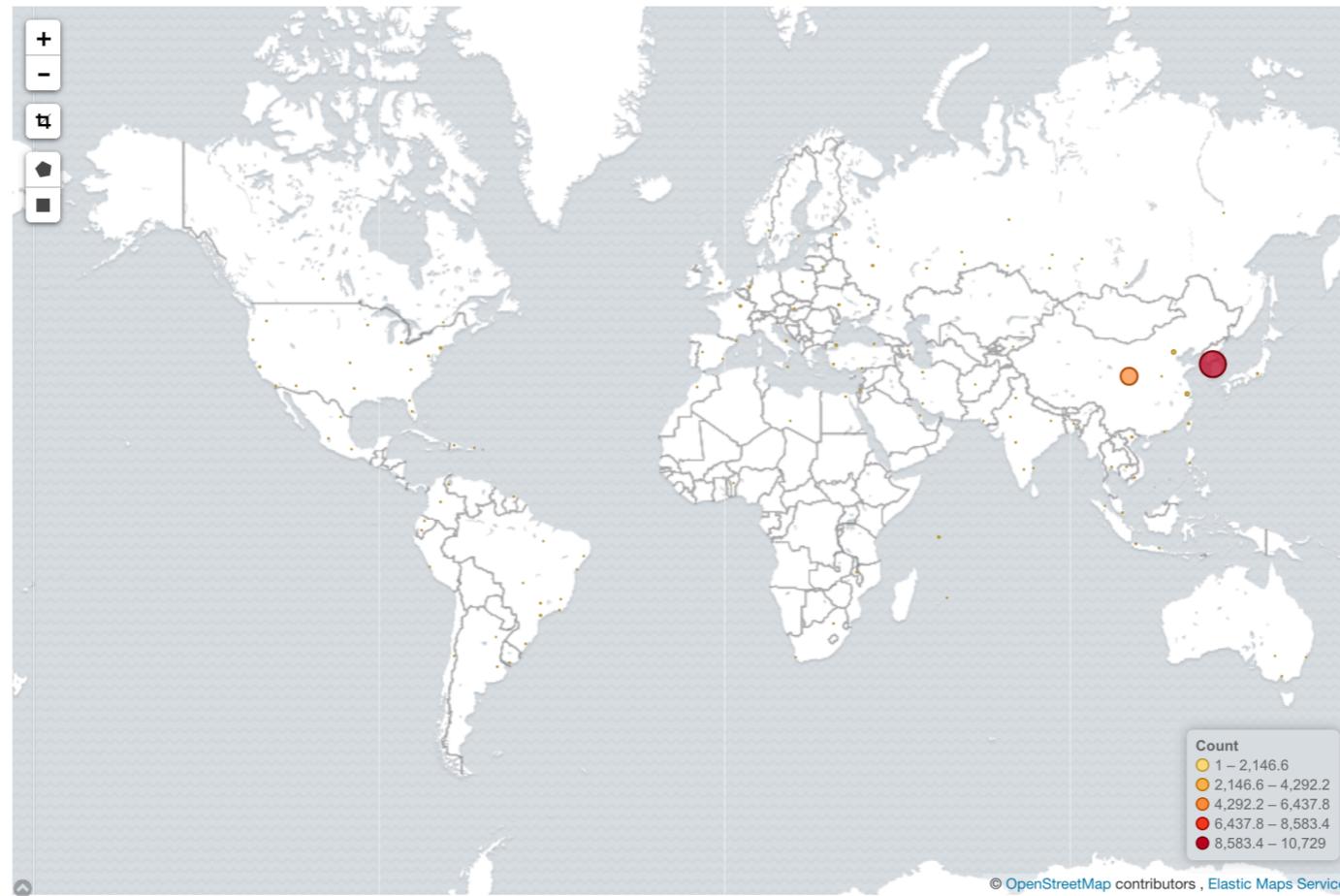
- nginx-\* index 중에서
- "@timestamp" field 기준 “2018년 6월 1일 ~ 2018년 6월 16일” documents들의
- “nginx.access.body\_sent.bytes” field 값이 가장 큰 documents 2개의
- “nginx.access.user\_agent.os\_name” field 표시

# 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 coordinate map named 'nginx-\*'. The top bar has tabs for 'Data' (selected) and 'Options'. A play button icon is highlighted with a pink box, and the number '5' with the text '실행' (Execute) is next to it. Below the tabs, there's a 'Metrics' section with a 'Value' button highlighted with a pink box. Under 'Buckets', 'Geo Coordinates' is selected. In the 'Aggregation' section, 'Geohash' is chosen from a dropdown menu. The 'Field' dropdown contains 'nginx.access.geoip.location'. At the bottom, there are three checked checkboxes: '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. A 'Custom Label' input field is empty. At the bottom right, there's a link to 'Advanced'.

< ⏪ This year ⏩ >

① Time Range를 This year로 설정

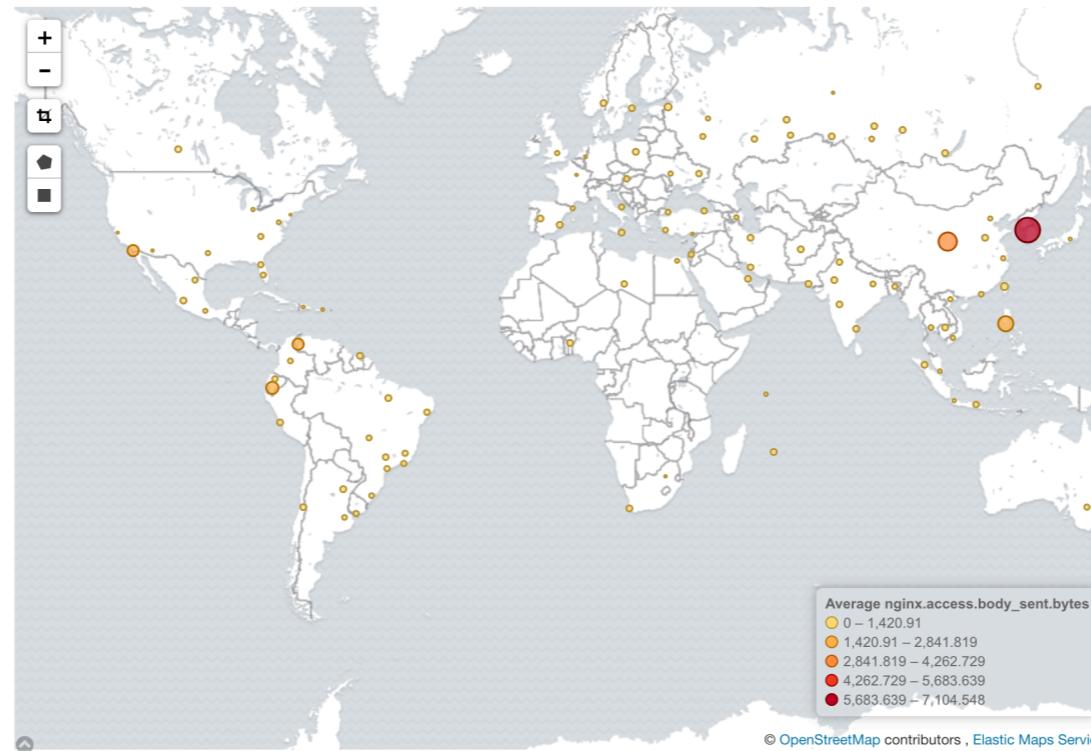
② Metrics aggregation은 고정

③ Geohash aggregation 선택 (필수)

④

Geohash aggregation을  
적용할 Field 선택

### 예제 3) Coordinate Map



조건

- **nginx-\* index** 중에서
- “@timestamp” 기준 “**2018년 6월 1일 ~ 2018년 6월 16일**” documents의
- “**nginx.access.geoip.location**” field에 Geohash aggregation을 적용한 후
- **nginx.access.body\_sent.bytes** field의 평균값 시각화

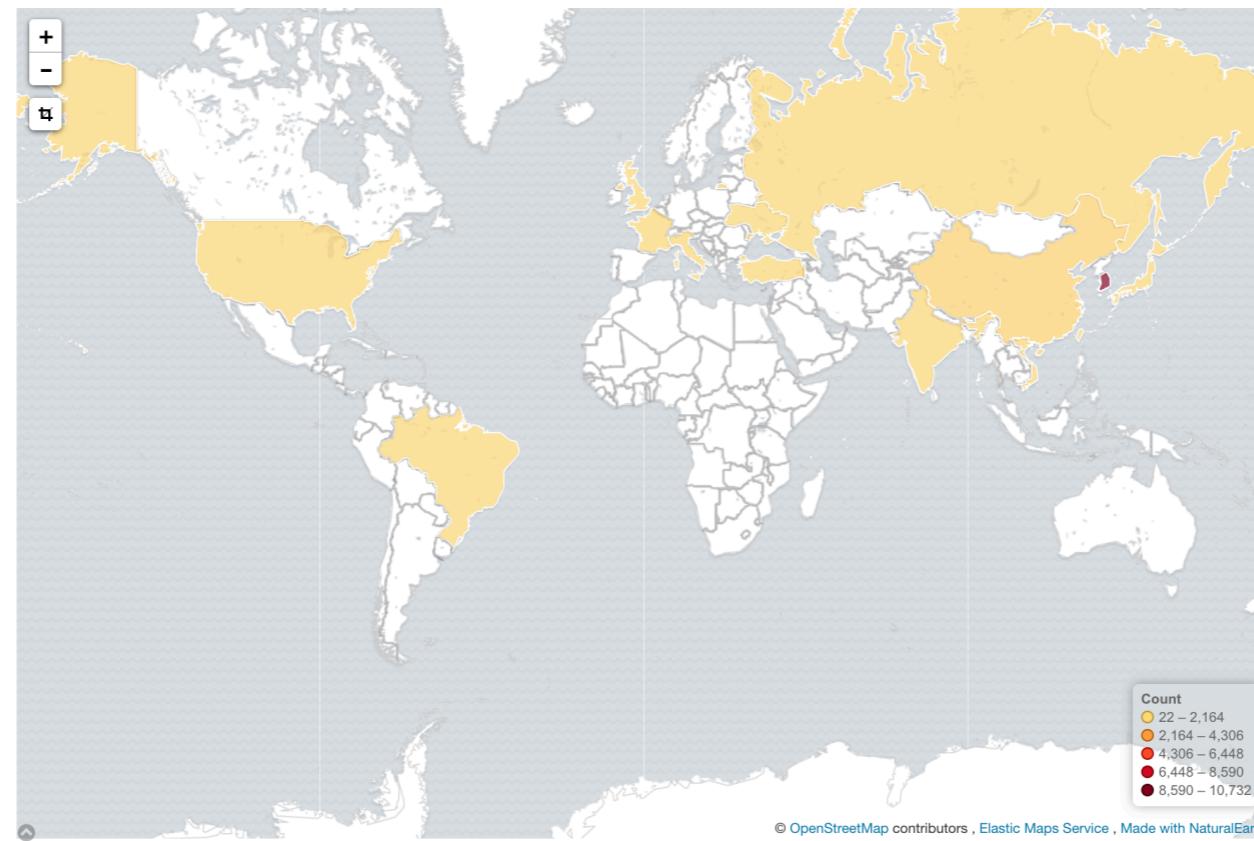
# Region Map



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 “**2018-06-01 ~ 2018-06-16**” 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 클릭 후 page 151 이동

Metrics Value Count

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 ➡

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

⑤

Documents 정렬 metric 선택

⑥

Documents 정렬 방식 선택  
(오름/내림)

## Region Map Configuration

nginx-\*

Data Options ⑪ 실행 ×

Layer Settings

Vector map: World Countries ▾

Join field: Two letter abbreviation ▾

Display warnings

Show all shapes

⑨ World Countries 선택  
⑩ Two letter abbreviation 선택

Style settings

Color Schema: Yellow to Red ▾

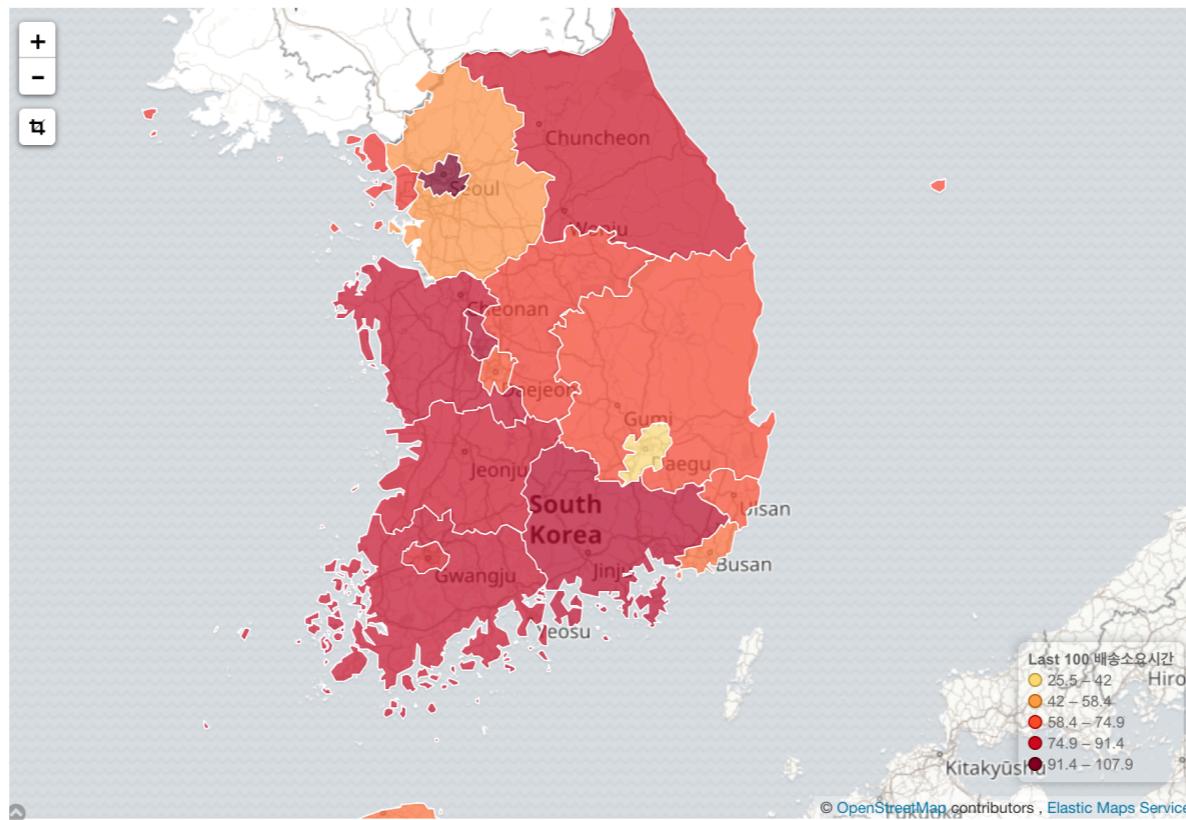
Outline weight: 2

Base Layer Settings

Layers: road\_map ▾

WMS compliant map server i

## 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



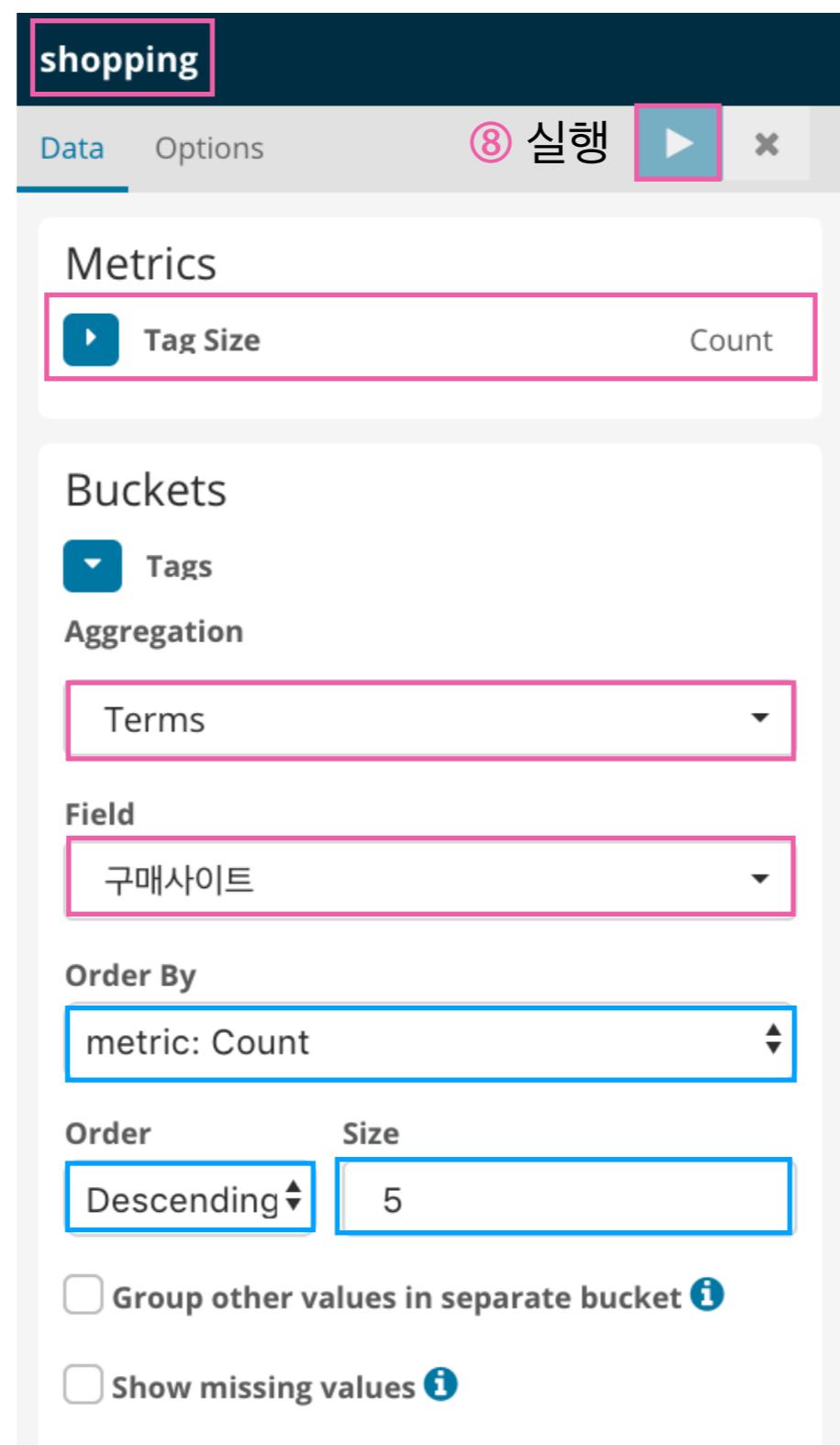
해석

- 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



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

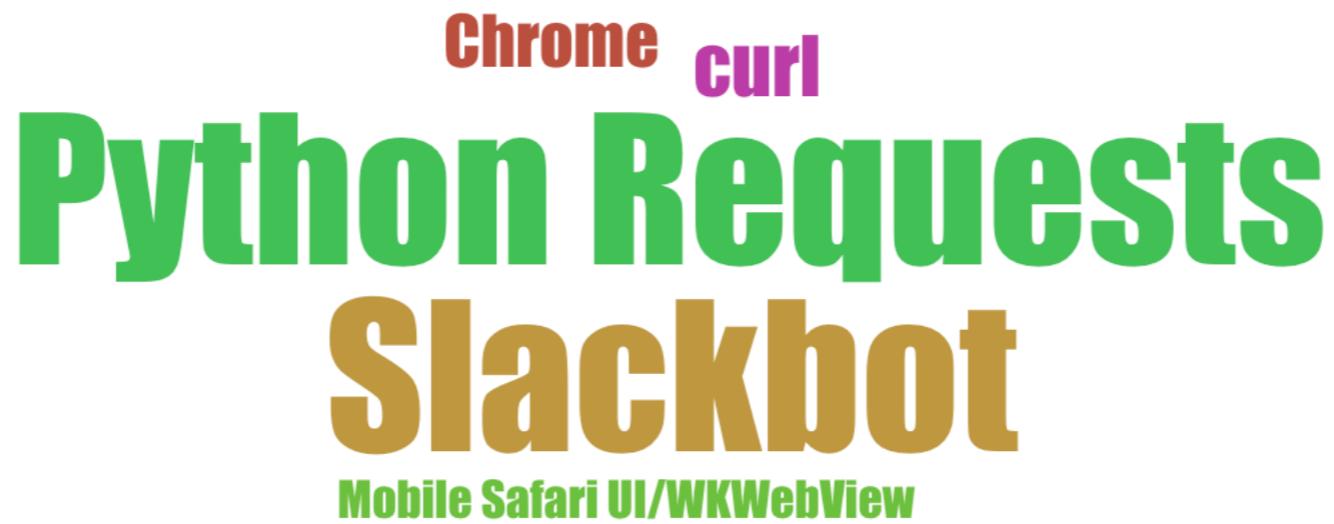
Documents 정렬 metric 선택

⑥

Documents 정렬 방식 선택  
(오름/내림)

⑦ 반영할 bucket 개수 입력

## 예제 5) Tag Cloud



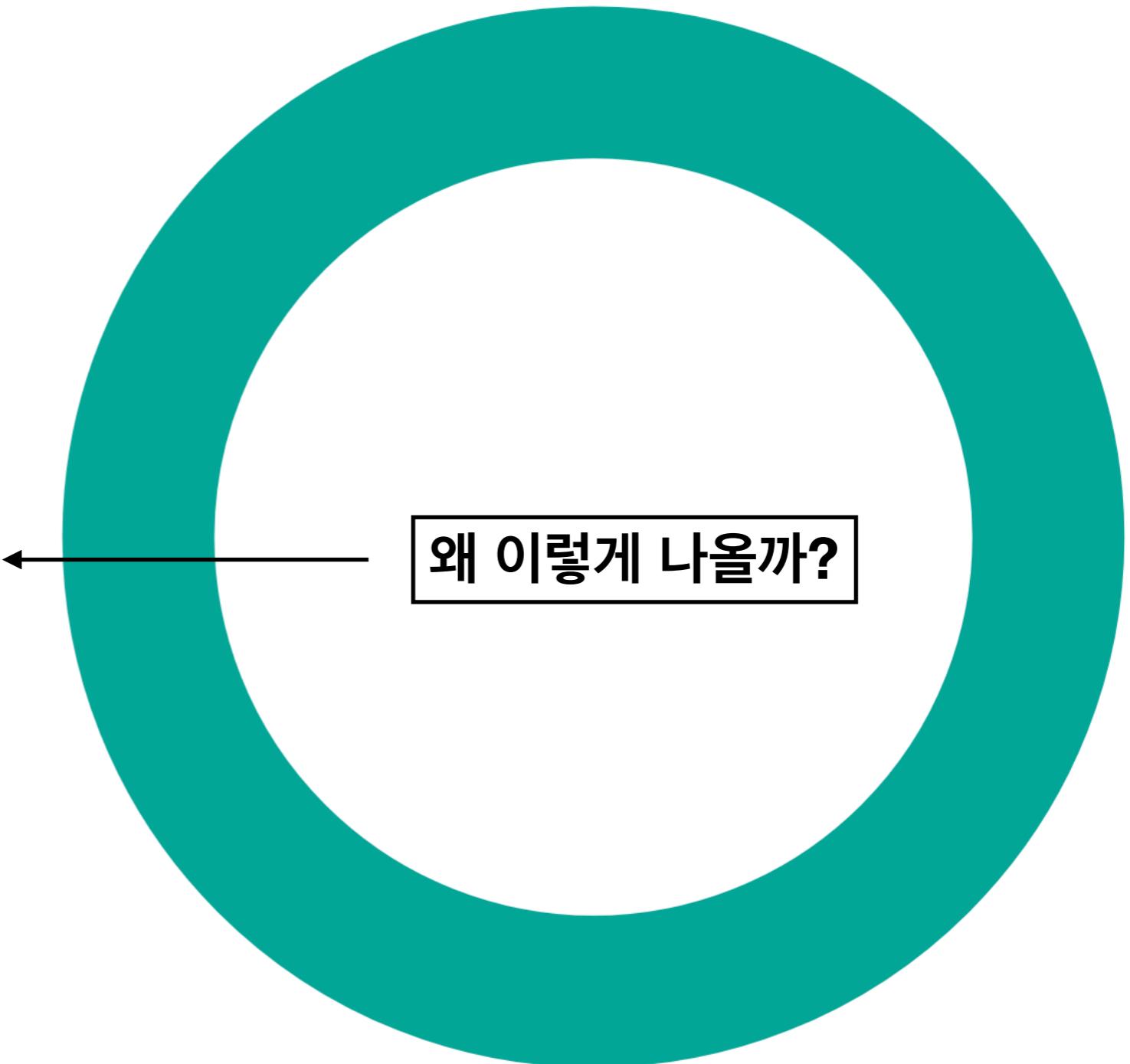
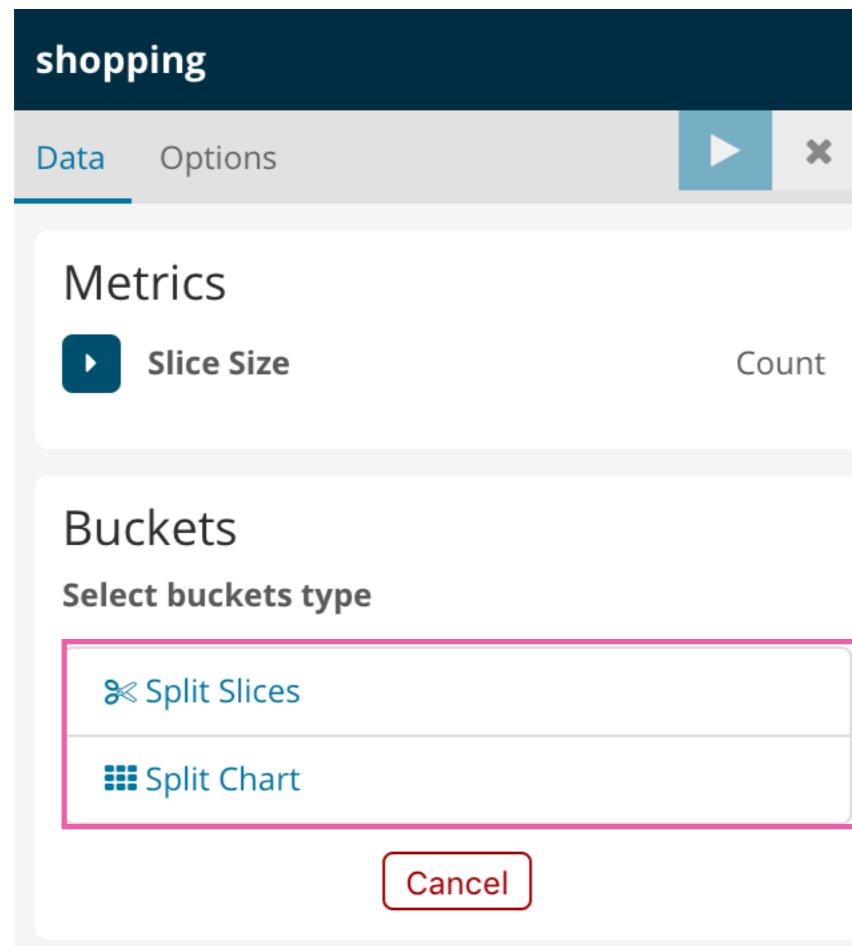
- nginx-\* index 중에서
- "@timestamp" field 기준 “2018년 6월 1일 ~ 2018년 6월 16일” documents의
- nginx.access.body\_sent.bytes field의 평균이 높았던 nginx.access.user\_agent.name field 5개의
- nginx.access.body\_sent.bytes field의 중위값

**이제는 Buckets 쪽에 집중해보자**

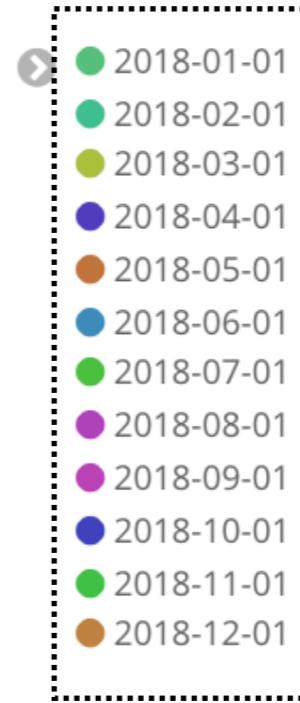
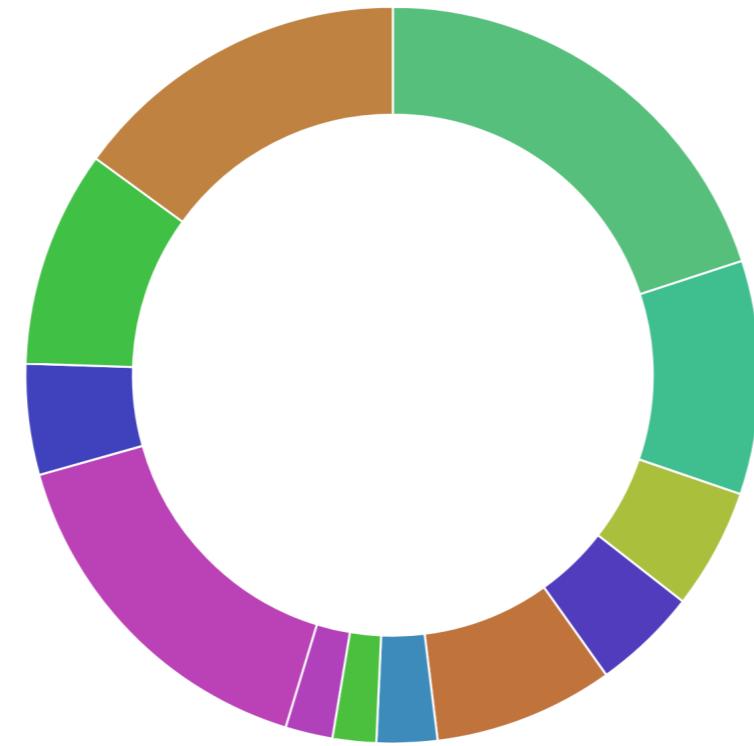
# Pie Chart



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



## 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

shopping

Data Options ▶ ×

Metrics ▶ Slice Size Count

Buckets ▼ Split Slices ✖

Aggregation ▼ Date Histogram

Field ▼ 주문시간

Interval ▼ Monthly

◀ This year ▶

◀ This year ▶

① Time Range를 This year로 설정

⑥ 실행

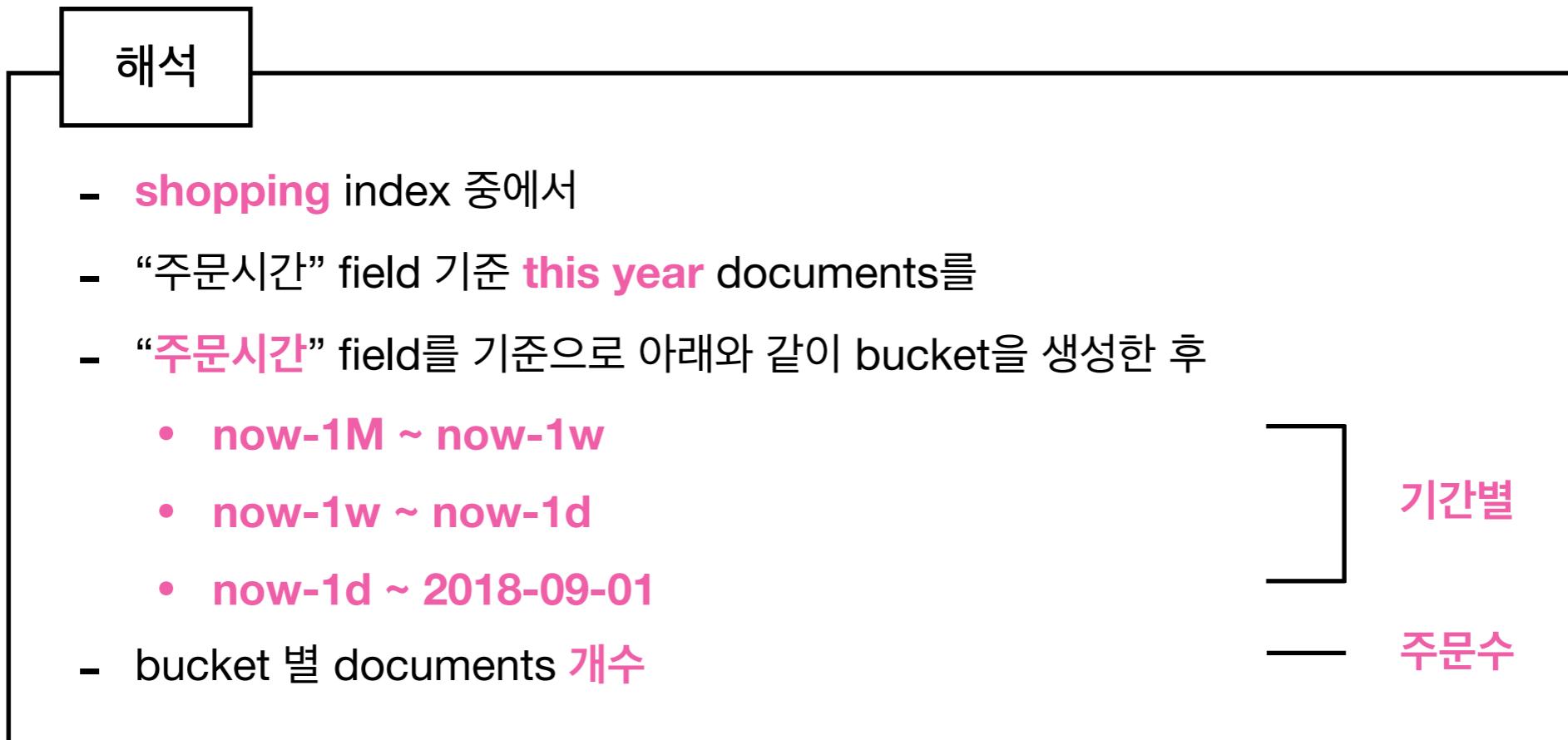
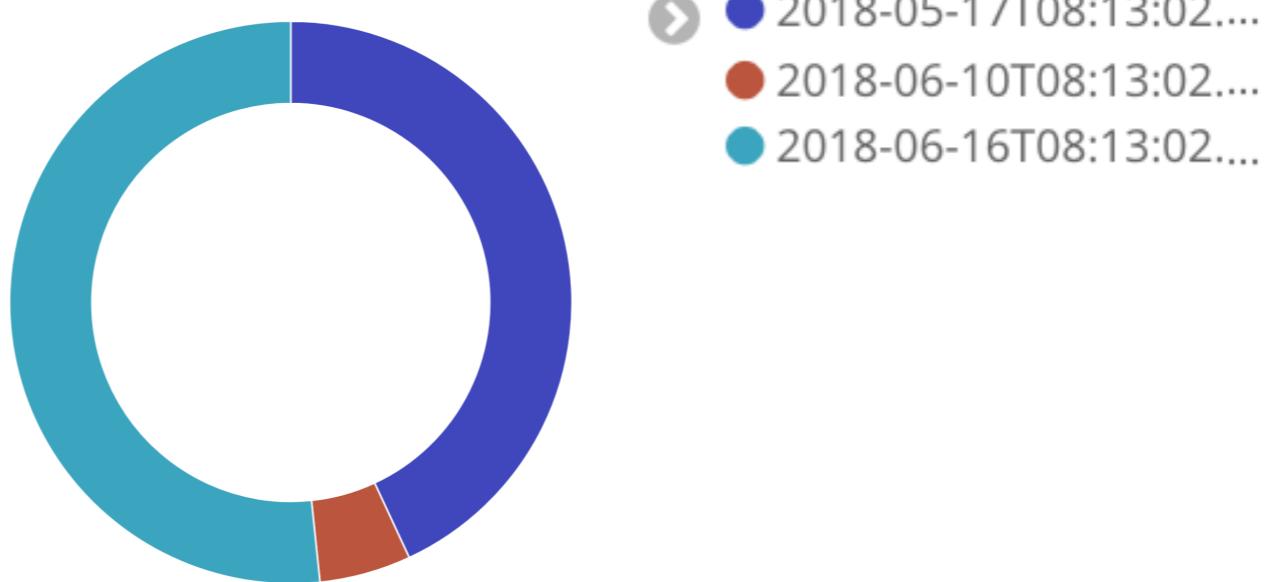
② Metrics aggregation은 고정

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

shopping

Data Options ⑥ 실행

Metrics

Slice Size Count

Buckets

Split Slices

Date Range

Field

주문시간

From	To
now-1M	now-1w
now-1w	now-1d
now-1d	2018-09-01

Accepted date formats

◀ ⏴ This year ➡

① Time Range를 This year로 설정

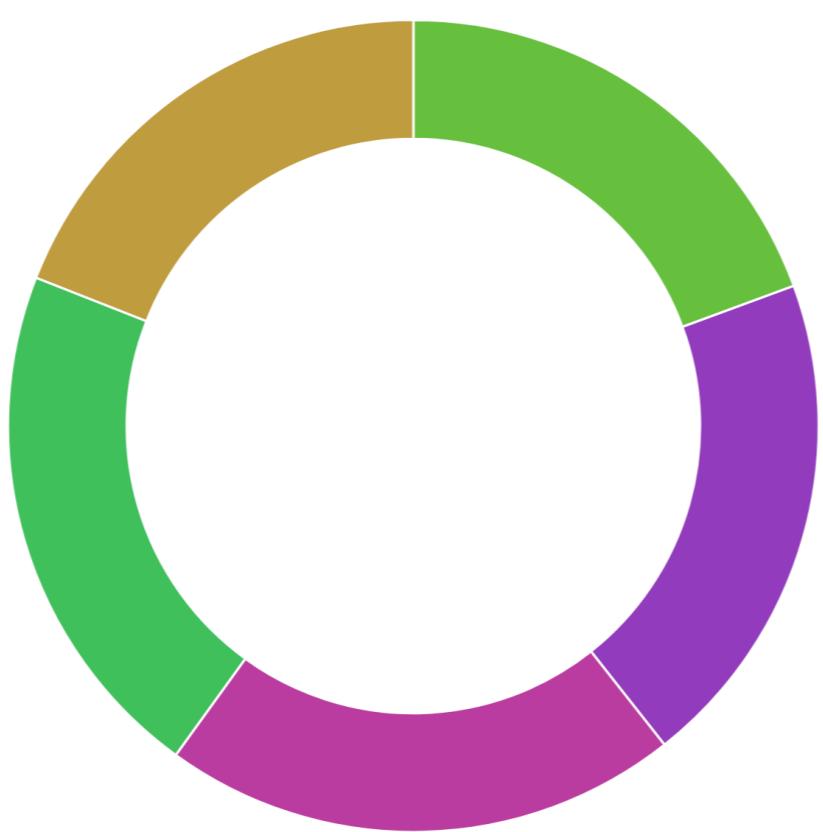
② Metrics aggregation은 고정

③ Date Range aggregation 선택

④ Date Range aggregation 적용 Field 선택

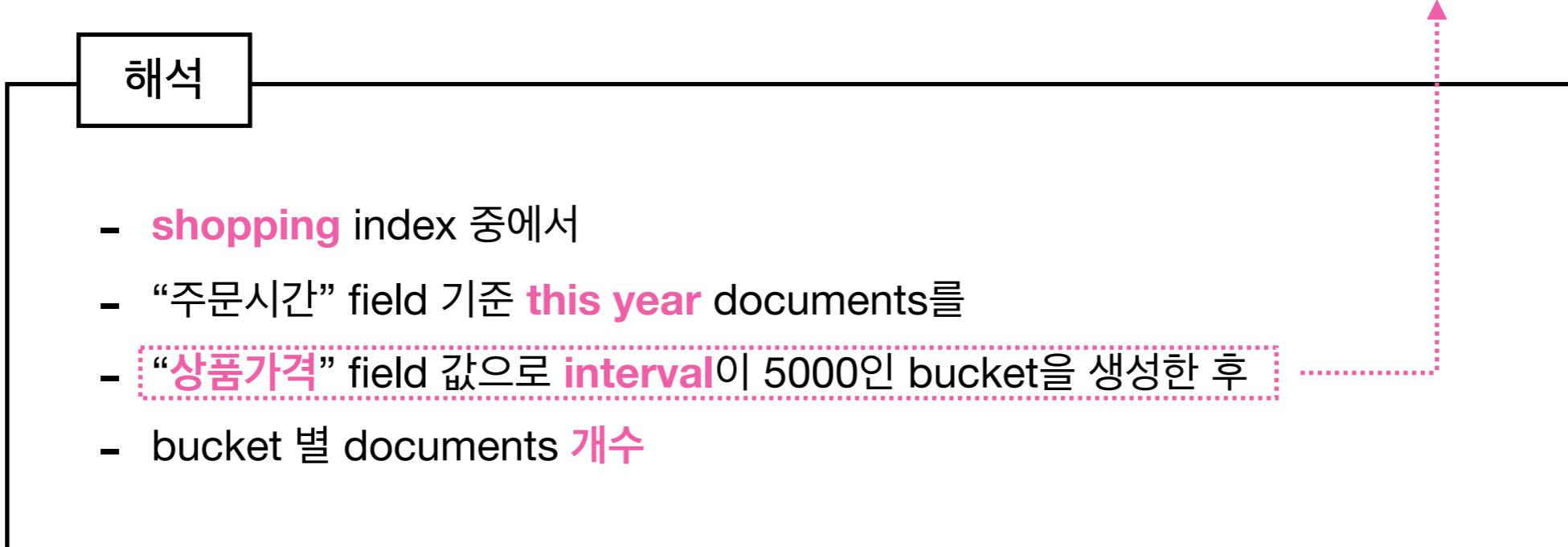
⑤ Bucket 별 Date Range 직접 입력

## Pie Chart Object



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

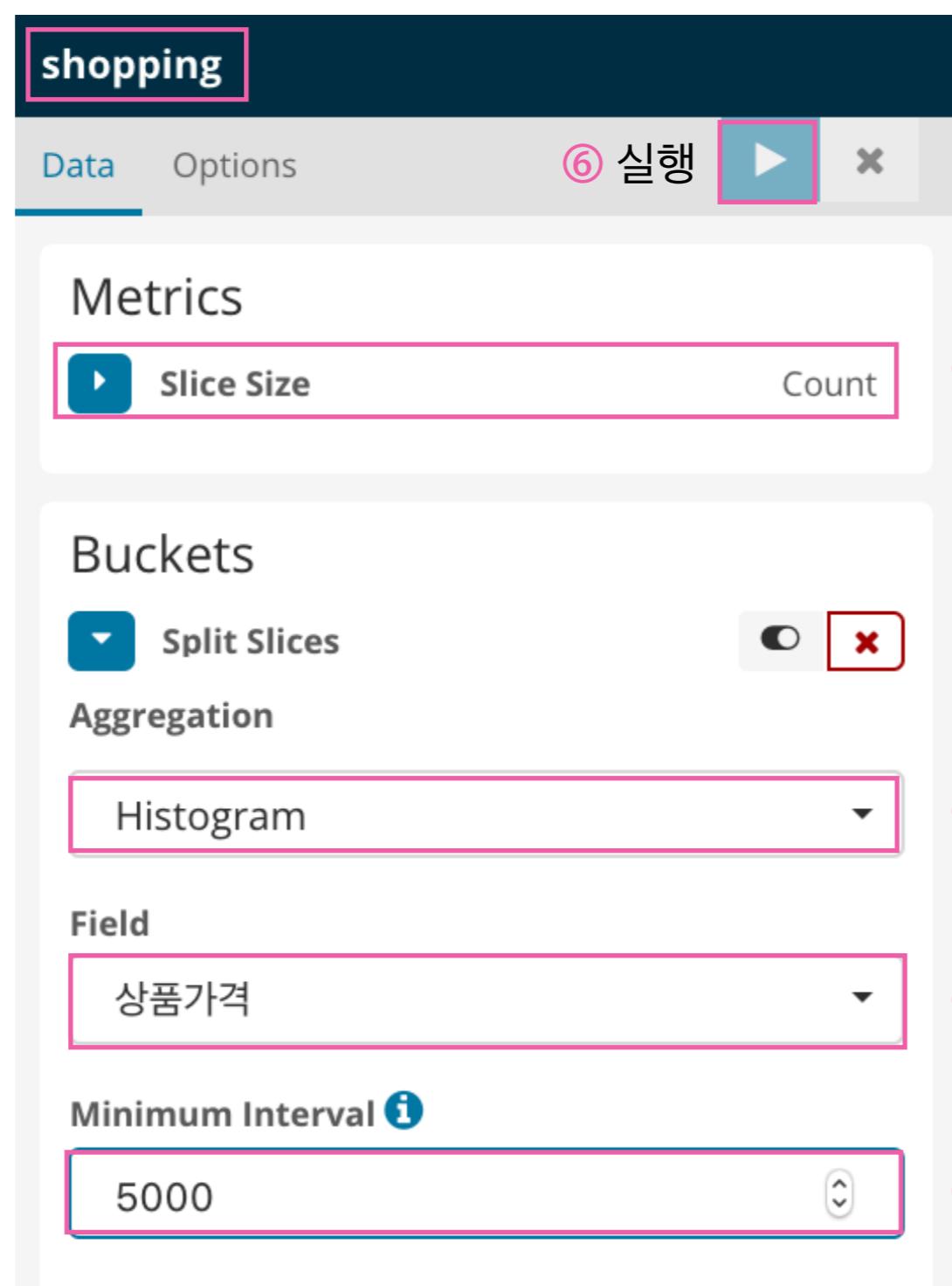
bucket	의미 (x = 상품가격)
5,000	$0 \leq x < 5,000$
10,000	$5,000 \leq x < 10,000$
15,000	$15,000 \leq x < 20,000$
20,000	$20,000 \leq x < 25,000$
25,000	$25,000 \leq x < 30,000$



## Pie Chart Configuration - Split Slices (Histogram)

①

(Pie Chart 선택 후)  
shopping index 선택



① Time Range를 This year로 설정

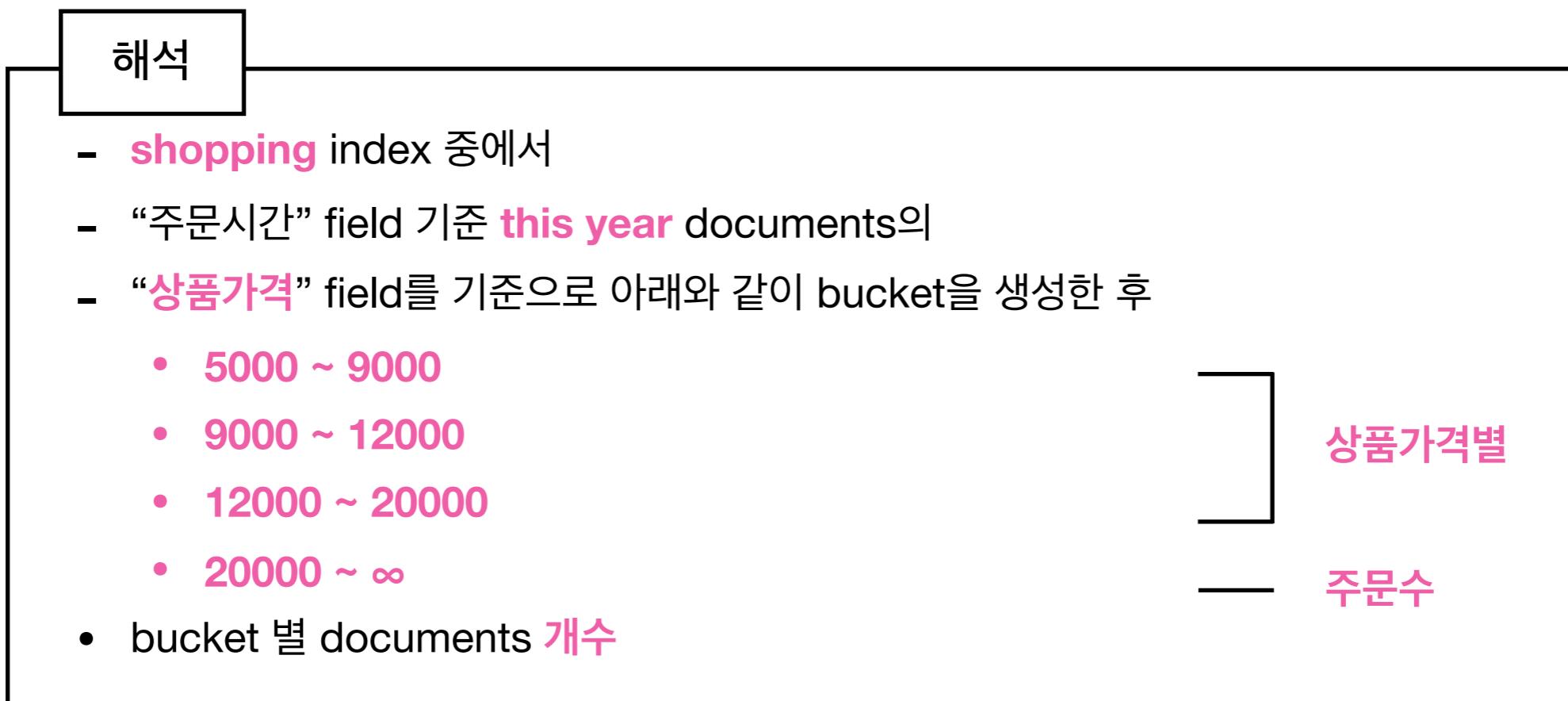
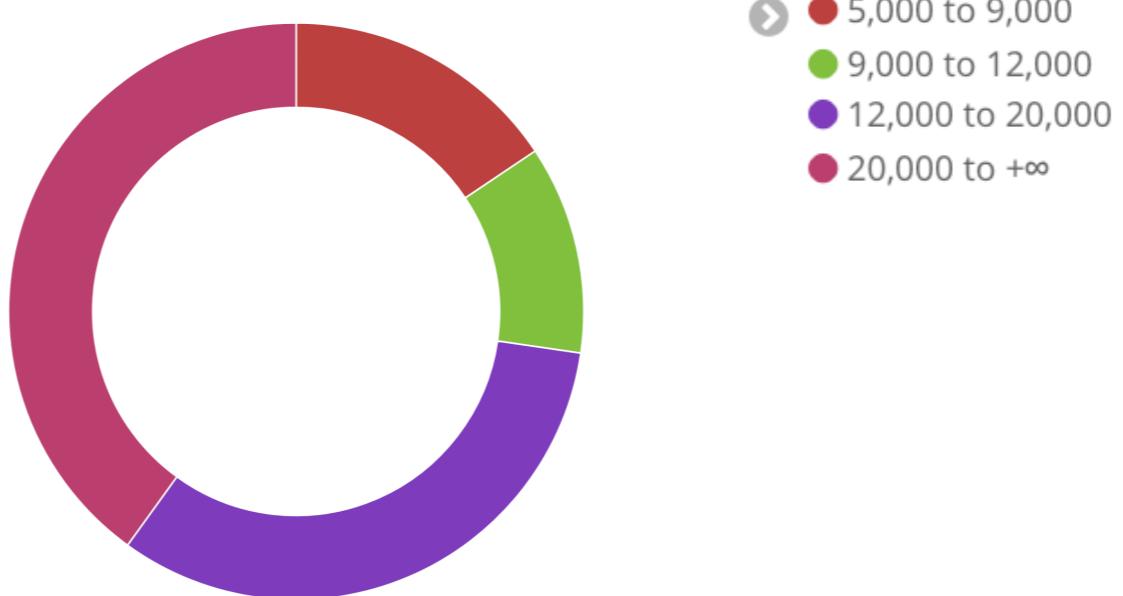
② Metrics aggregation은 고정

③ Histogram aggregation 선택

④ Histogram aggregation 적용할 Field 선택

⑤ Histogram 간격 설정

## Pie Chart Object



## Pie Chart Configuration - Split Slices (Range)

①

(Pie Chart 선택 후)  
shopping index 선택

shopping

Data Options ⑥ 실행

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

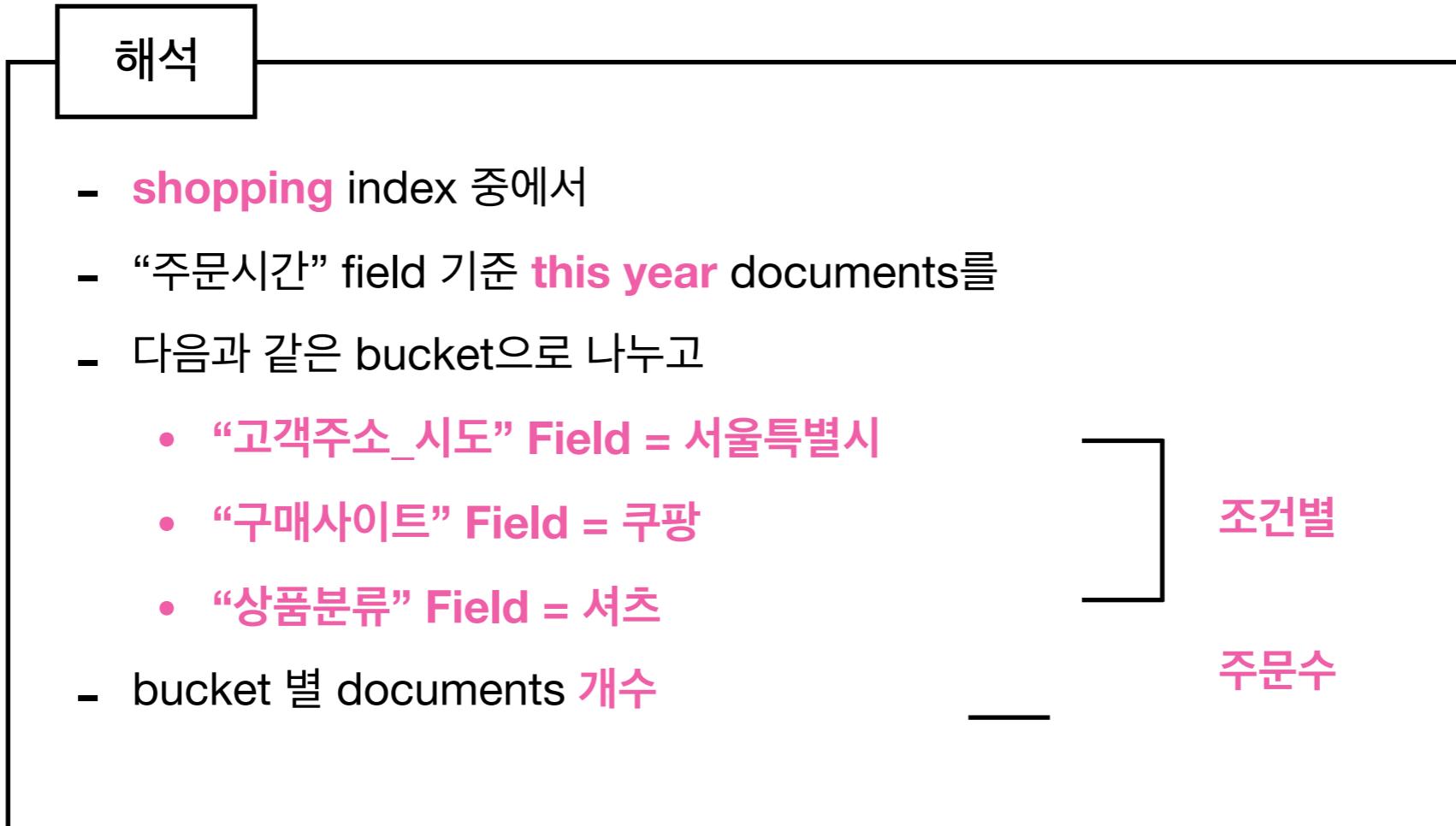
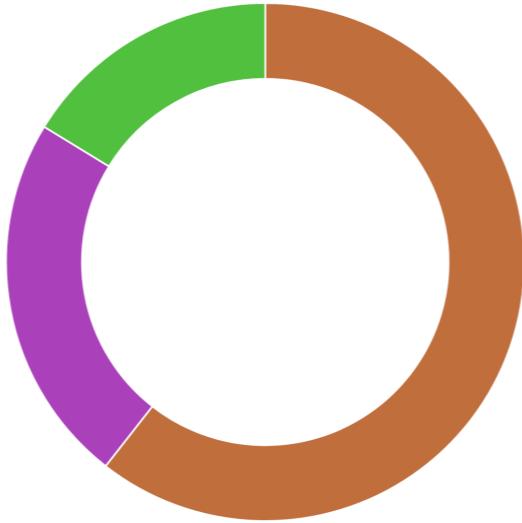
③ Range aggregation 선택

④ Range aggregation 적용할 Field 선택

⑤ bucket 별 간격 설정

## Pie Chart Object

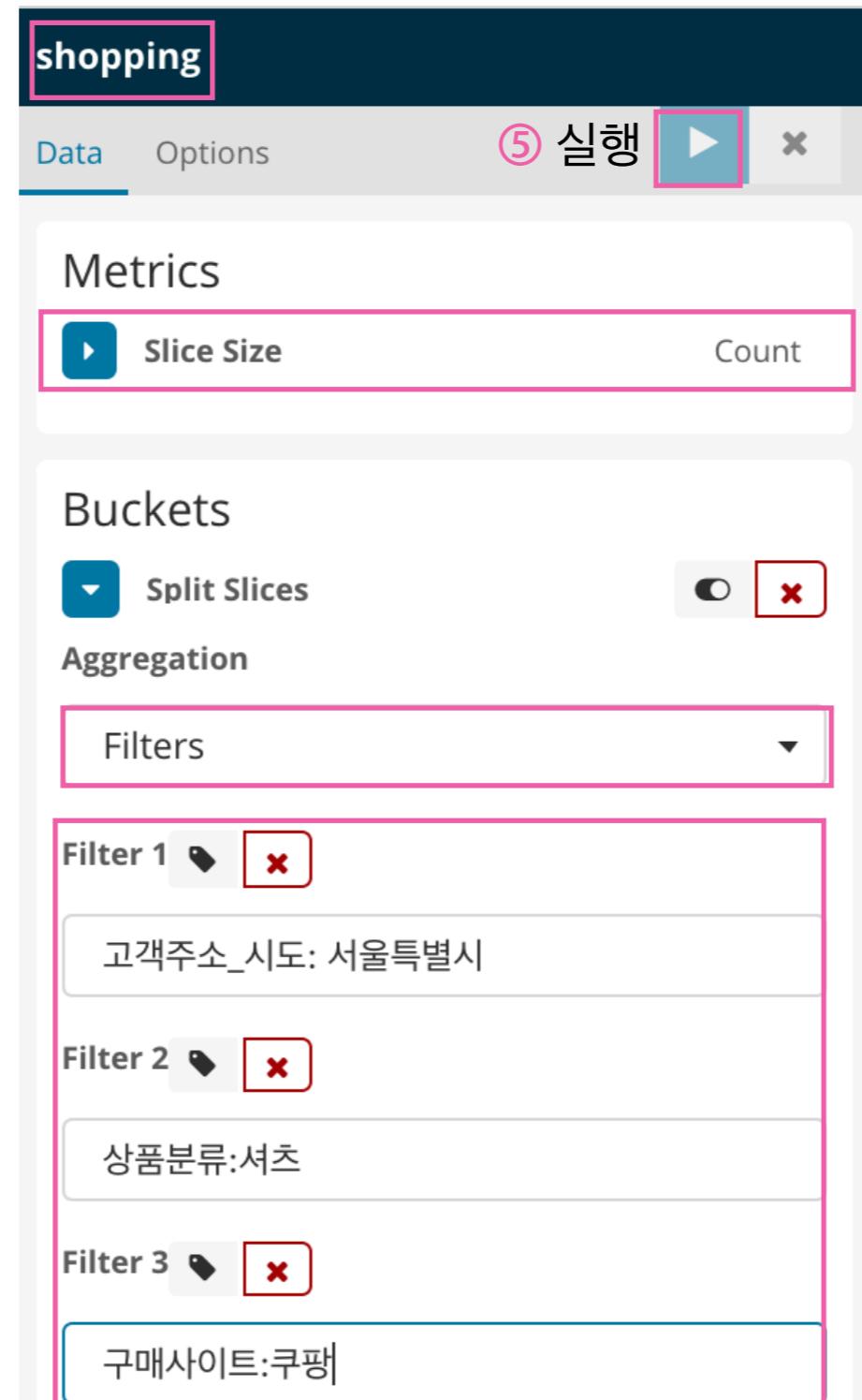
- ▶ ● 고객주소\_시도: 서울특별시
- 구매사이트: 쿠팡
- 상품분류: 셀프



## 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

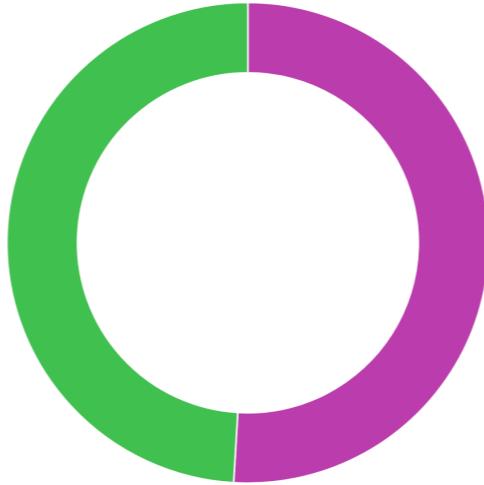
③ Filters aggregation 선택

④ Filter를 작성하여 Bucket 생성

**(문법은 나중에 배웁니다)**

## Pie Chart Object

- ▶ ● 스커트
- 블라우스



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

①

(Pie Chart 선택 후)

shopping index 선택

shopping

Data Options ⑦ 실행 ×

Metrics

Slice Size Count

Buckets

Split Slices

Aggregation

Significant Terms

Field

상품분류

Size

2

Group other values in separate bucket i

Show missing values i

① Time Range를 This year로 설정

② Metrics aggregation은 고정

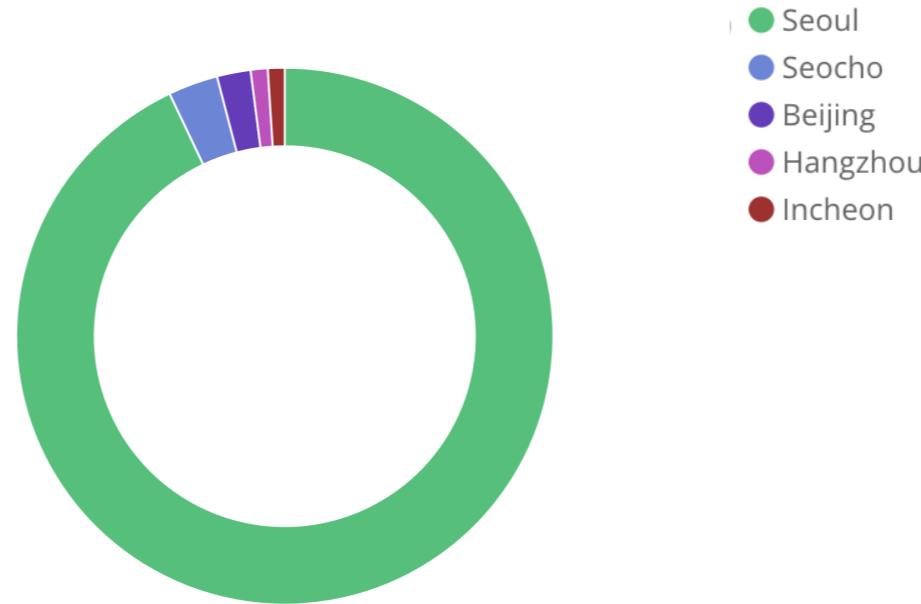
③ Significant Terms aggregation 선택

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 6월 16일” 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

- shopping index 중에서
- “주문시간” field 기준 this year documents를
- “nginx.access.geoip.city\_name” field 값으로 bucket을 생성한 후
- documents 개수가 가장 많은 2개의 bucket에 대해서 각각
- “nginx.access.user\_agent.name” field 값으로 sub-bucket을 생성한 후
- documents 개수가 가장 많은 2개의 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

## 예제 7) Tag Cloud

- Mac OS X
- iOS
- Other
- Windows 10



2018-06-04: @timestamp

2018-06-11: @timestamp

조건

- nginx-\* index 중에서
- "@timestamp" field 기준 “**2018년 6월 1일 ~ 2018년 6월 16일**” documents를
- “**@timestamp**” field를 기준으로 주별(weekly) bucket을 생성하고
- 각 bucket에 대해서
- “**nginx.access.user\_agent.os**” field 값으로 sub-bucket을 생성한 후
- documents 개수가 **가장 많은 3개**의 sub-bucket 마다의
- documents 개수

주별

user\_agent별

접속수

## 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**