

Elastic Stack 을 활용한 Data Dashboard 만들기

Week 1 - Data를 시각화해보자



kibana

Fast Campus

내용	페이지
강의소개	3
Elastic Stack 소개	7
용어 정리	17
Elastic Stack Workflow	24
Index 등록	27
데이터 탐색	31
Visualize 맛보기	35
Aggregation	
Bucket Aggregation	42
Metric Aggregation	47
Visualize Guide	53
Visuzlie 실전	
Markdown	74
Metric	75
Coordinate Map	86
Region Map	88
Tag Cloud	90
Pie Chart	93

강의가 끝나면	data가 주어지면 dashboard를 구축하고 needs에 맞게 운영할 수 있다.
단,	<ul style="list-style-type: none">모든 기능 100% 마스터는 하지 않을 거고dashboard 구축 및 운영을 위한 전반적인 내용 학습과문제가 생길 시 troubleshoot 하는 방법을 중심으로 배운다.
그러므로	<ul style="list-style-type: none">검색엔진으로서 ElasticsearchElasticsearch Architecture 등은 다루지 않을 것이다.(고급) query 및 query 최적화

FAQ

자주 물어보는 질문 정리 

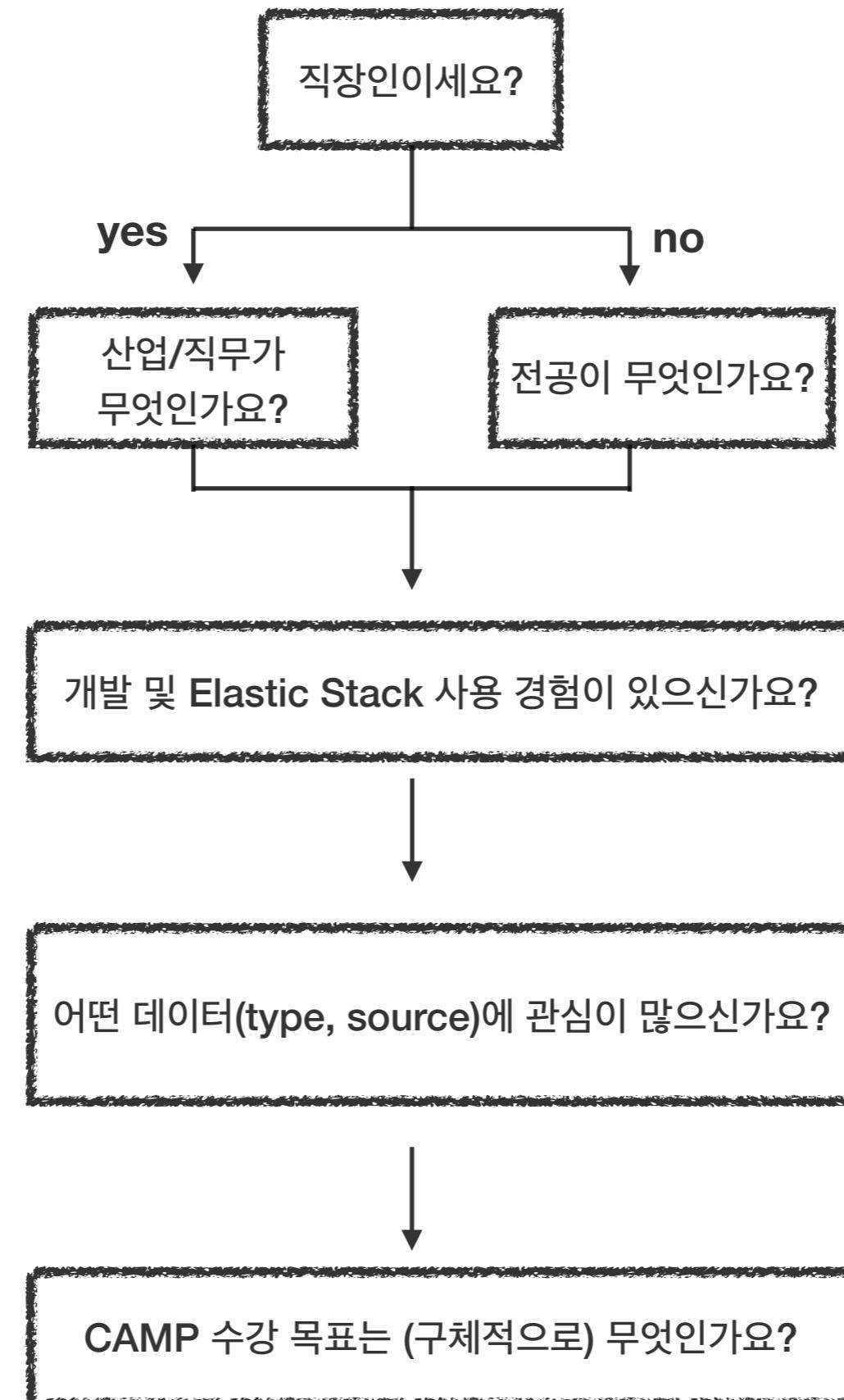
WIKI

Elastic Stack 간단한 사용법 정리 

Questions

- 수업 중 : Slido 
- 수업 외 : [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



용어 정리

ELK Stack?
Elastic Stack?
Elasticsearch?
Elastic?

ELKB Stack?

Elastic Stack이 무엇인지 간략히 살펴보자

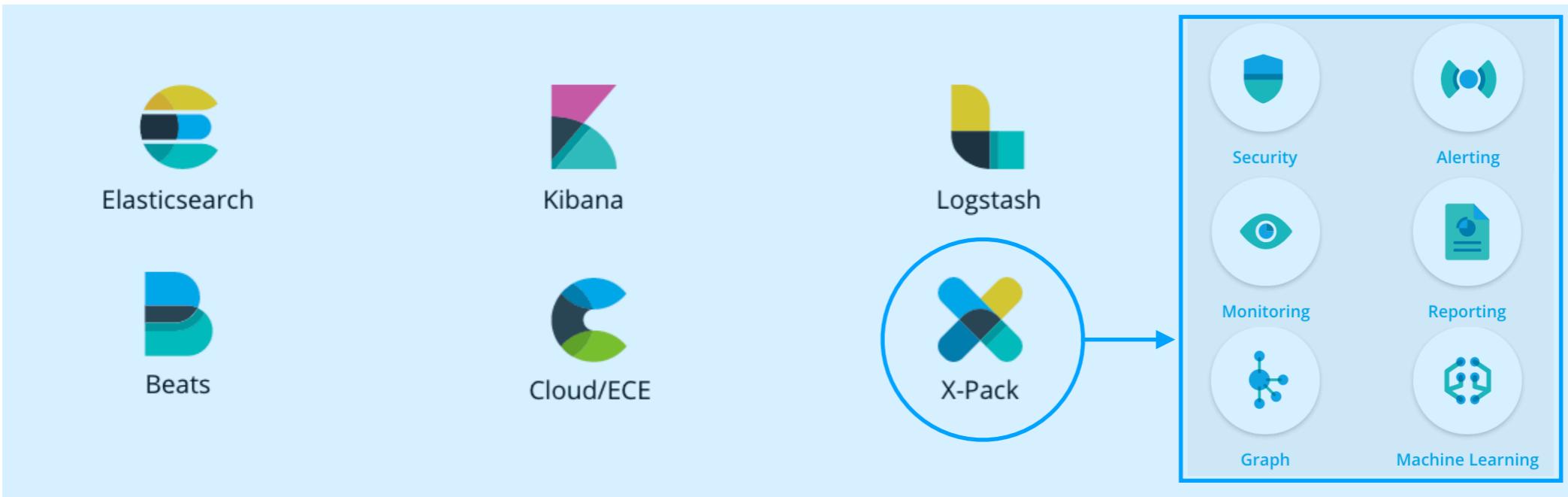
Elastic Stack

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

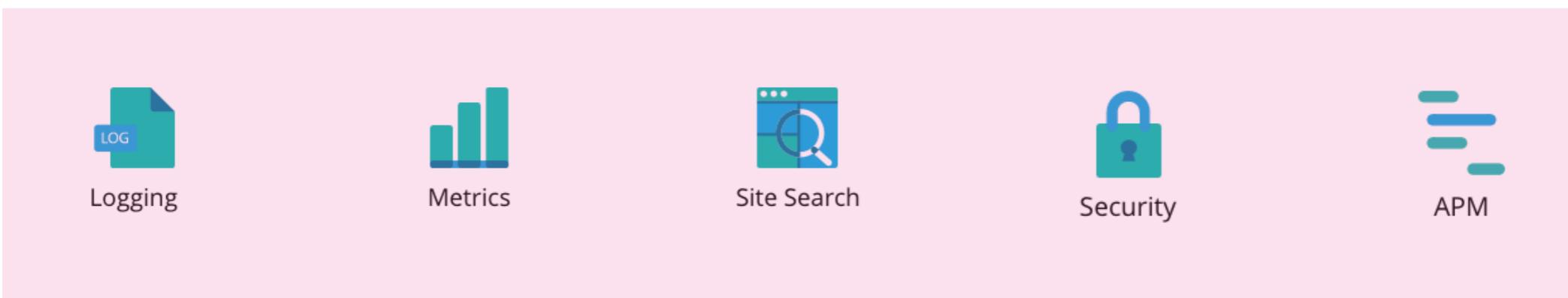
Elastic Stack으로 무얼 할 수 있을까?

=Elastic Stack을 왜 배울까?

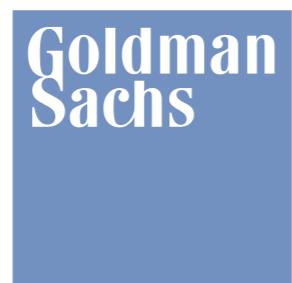
Products



Solutions



Elastic Stack을 실제 Production에서 사용중인 회사는 있을까? 🤔 🤔



NAVER



WIKIPEDIA
The Free Encyclopedia



BBC



tinder



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

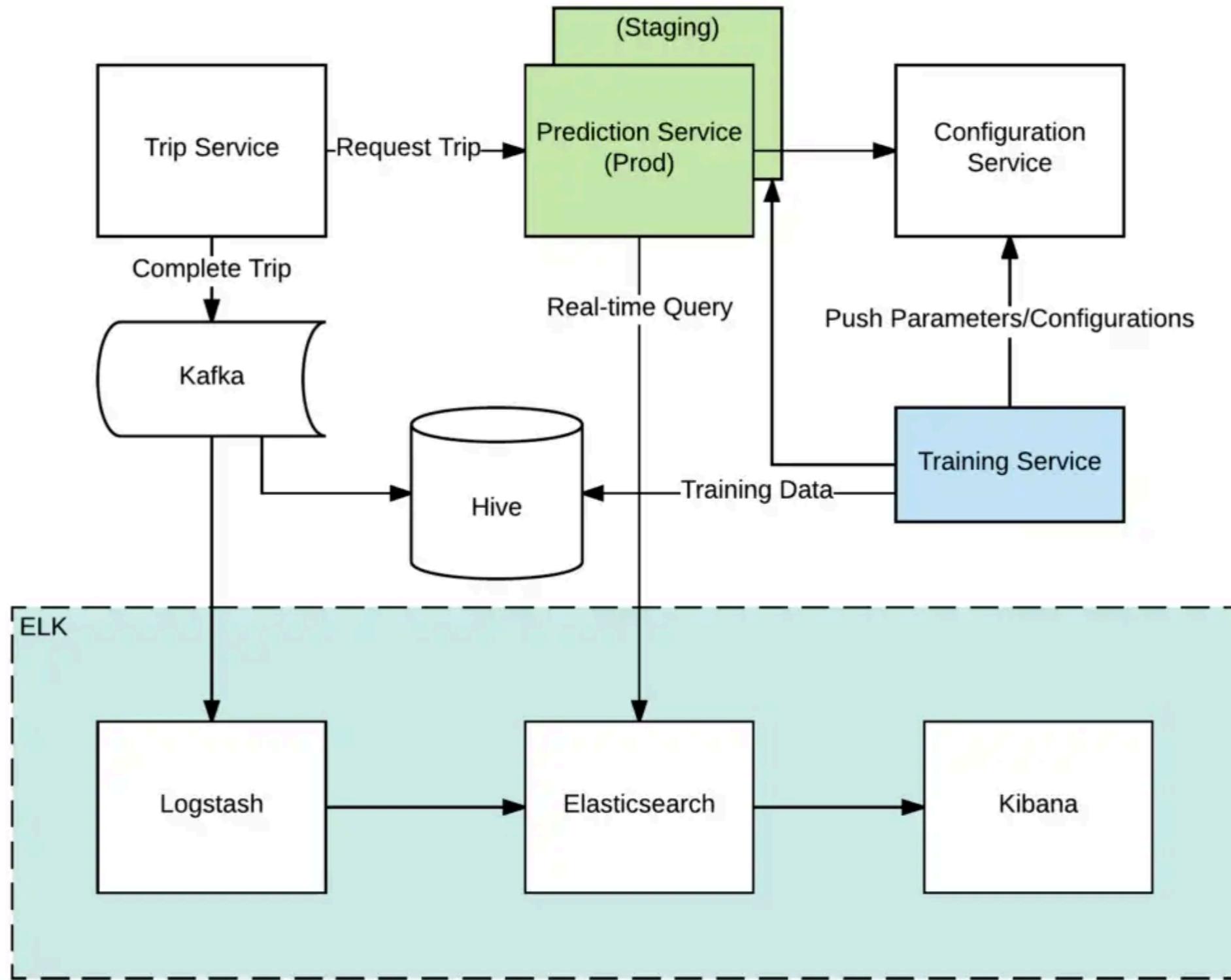
Algorithm (k-nearest neighbors algorithm, KNN)

- 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



용어 정리

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

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

Elasticsearch - Index

RDBMS



```
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



```
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.    "sort": [
28.      {"date": "2018-01-01", "score": 1}
29.    ]
30.  }
31.}
```

⋮

Excel



	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								

Elasticsearch - Type

RDBMS

```
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

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

Excel

	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								

Elasticsearch - Document

RDBMS

```
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

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

Excel

	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								

Elasticsearch - Field

RDBMS

```
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

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

Excel

	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								

Elasticsearch - Mapping

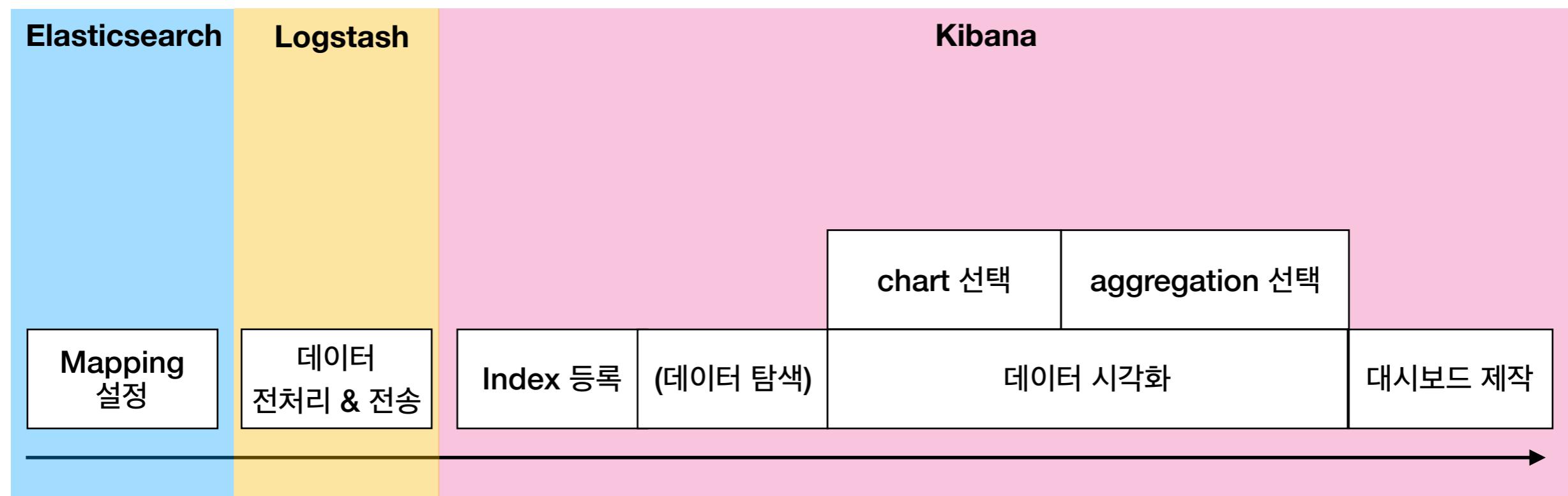
RDBMS

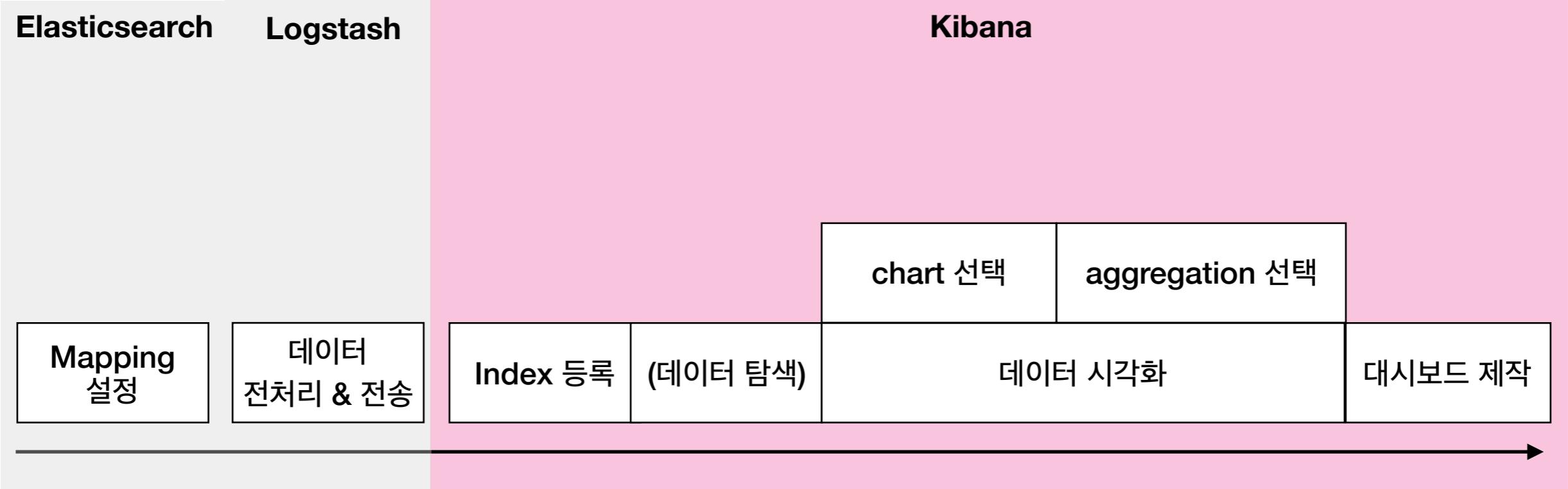
```
mysql> CREATE TABLE Sheet1 (
    -> date DATE,
    -> product VARCHAR(32),
    -> quantity INT(100),
    -> sales INT(100)
    -> );
```

Elasticsearch

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

Elastic Stack Workflow





데이터 등록

Kibana 시작화의 전제조건

- 데이터가 Elasticsearch에 저장되어 있어야 한다
- Elasticsearch에 저장된 데이터를 Kibana에 등록해야 한다

등록 과정 

- Kibana 접속 - Management 선택 - Index Patterns 선택 - Create Index Pattern 선택
- Elasticsearch에 저장한 Index 이름 입력
 - 목적에 맞게 exact match 또는 wildcard match 선택
 - 예를 들어 elasticsearch index 이름이 [“2018_01_01.log”, “2018_01_02.log”] 라고 하자

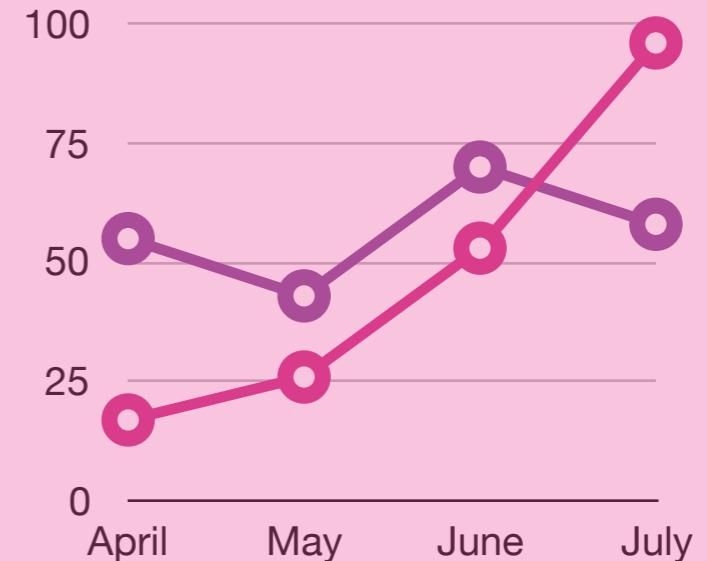
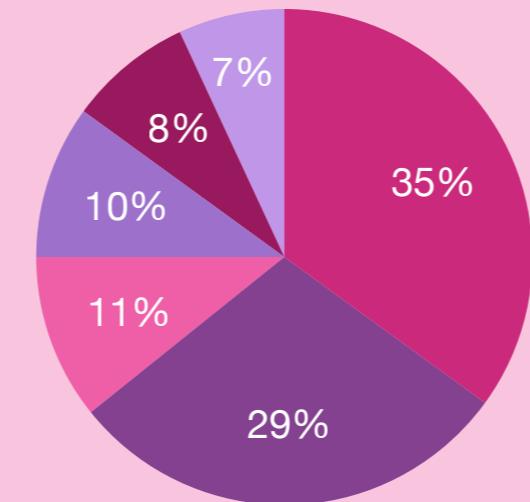
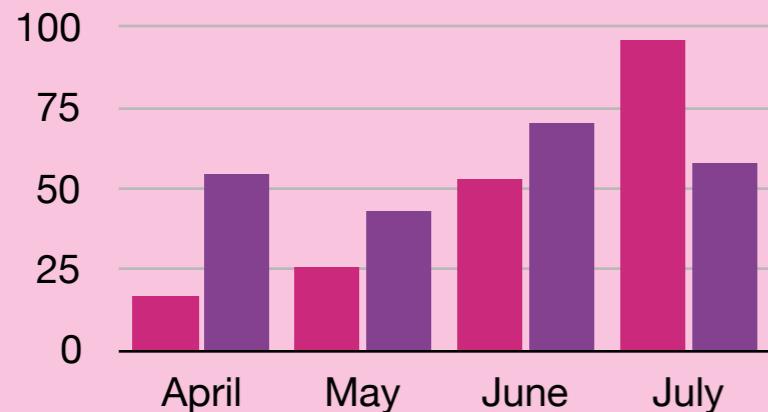
	elasticsearch index	kibana index pattern
exact match	1) 2018_01_01.log 2) 2018_01_02.log	1) 2018_01_01.log 2) 2018_01_02.log
wildcard match	1) 2018_01_01.log 2) 2018_01_02.log	2018_01_*.log

- Timer Filter field name 입력
 - 등록하려는 Elasticsearch Index에서 기준 시간으로 삼을 Date Field를 선택한다.
 - 단, Time Filter를 사용하지 않을 경우 “I don't want to use the Time Filter”를 선택하자.

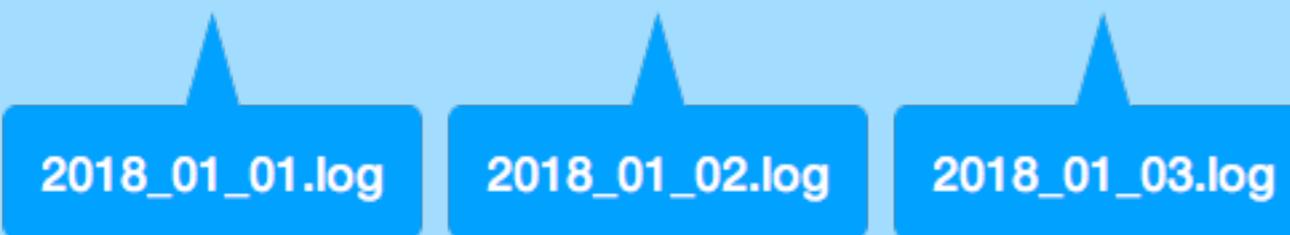
Q. Index 등록시 Wildcard는 왜 필요한지?

A. 데이터 저장은 **분산**, 검색 및 시각화는 **통합**해서 하기 위해, 즉 관리의 편의성!

Kibana



Elasticsearch

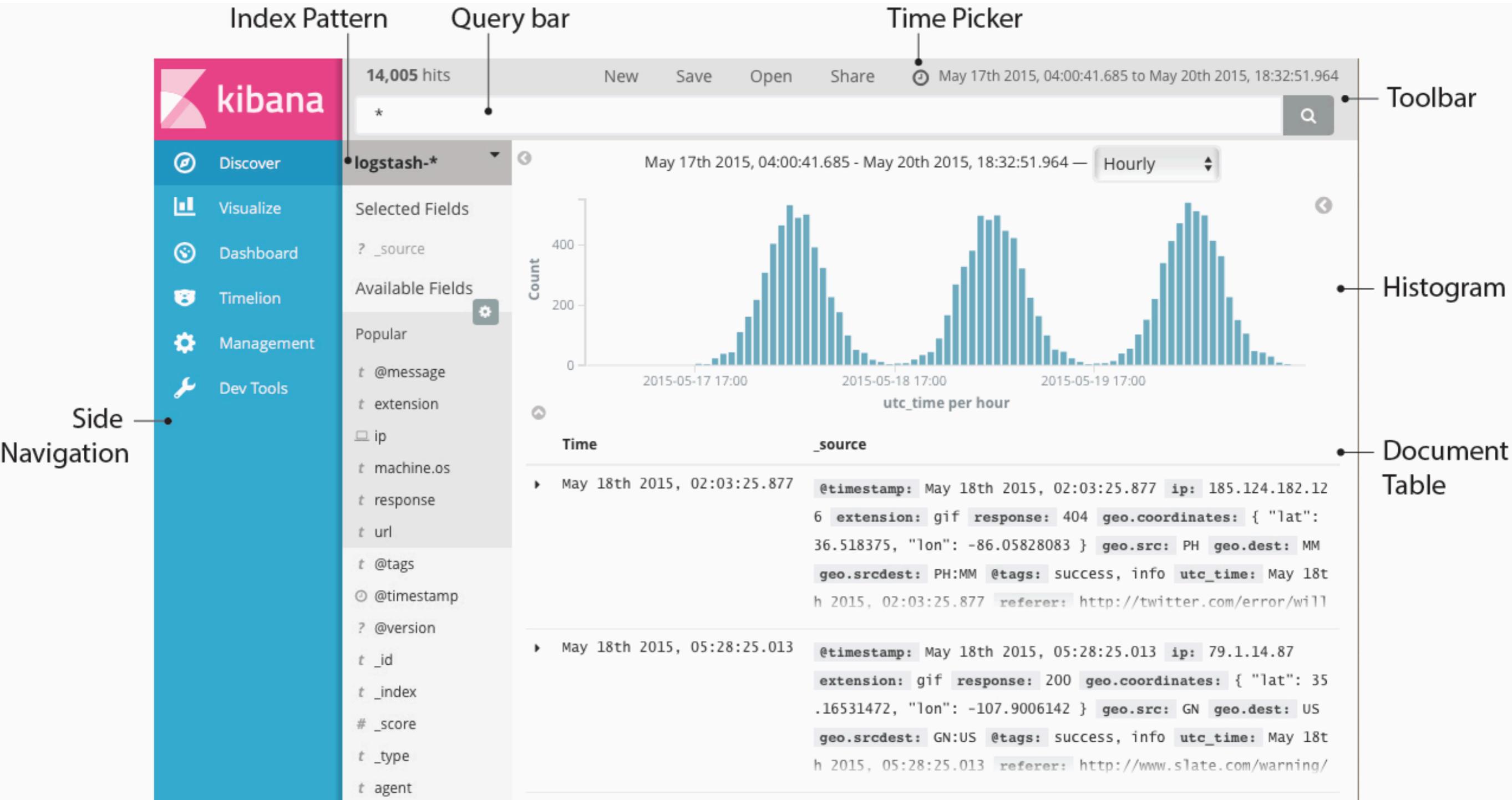


방금 배운 내용을 직접 해보자 ✎

예) id = higee

번호	<i>elasticsearch index</i>	time field	<i>kibana index pattern</i>
1	{id}_2018.03.01	주문시간	ex) higee_2018.03.01
1	{id}_2018.03.02		ex) higee_2018.03.02
2	{id}_2018.03.01, {id}_2018.03.02	주문시간	ex) higee_2018*

데이터 탐색



데이터를 시각화 하기 전에 데이터를 탐색하는 과정

주요 기능

세부 기능

데이터 검색

(복잡한) 조건을 만족하는 데이터 선별적 탐색 가능

데이터 검색 저장

검색한 결과를 저장하여 Visualize에서 사용

데이터 필터링

(간단한) 특정 조건을 만족하는 데이터 선별적 탐색 가능

데이터 조회

- 특정 Document를 Table/JSON 형태 조회
- Histogram 특정 구간 내의 데이터 조회
- Histogram Bin 간격 설정
- Histogram 데이터를 csv 출력
- 특정 Field의 정보만 조회
- 특정 Field 값을 기준으로 정렬

데이터 통계

- (선택한 Time Range 내의) Documents 개수 확인
- 특정 Field Value의 분포 확인 (주로 Categorical Data, 상위 500개)
- 특정 Field에 non-null Value가 아닌 Documents 수 확인

→ “검색 및 필터” 학습 후

실습에 앞서 가장 중요한 2가지 설정을 하자 ↗

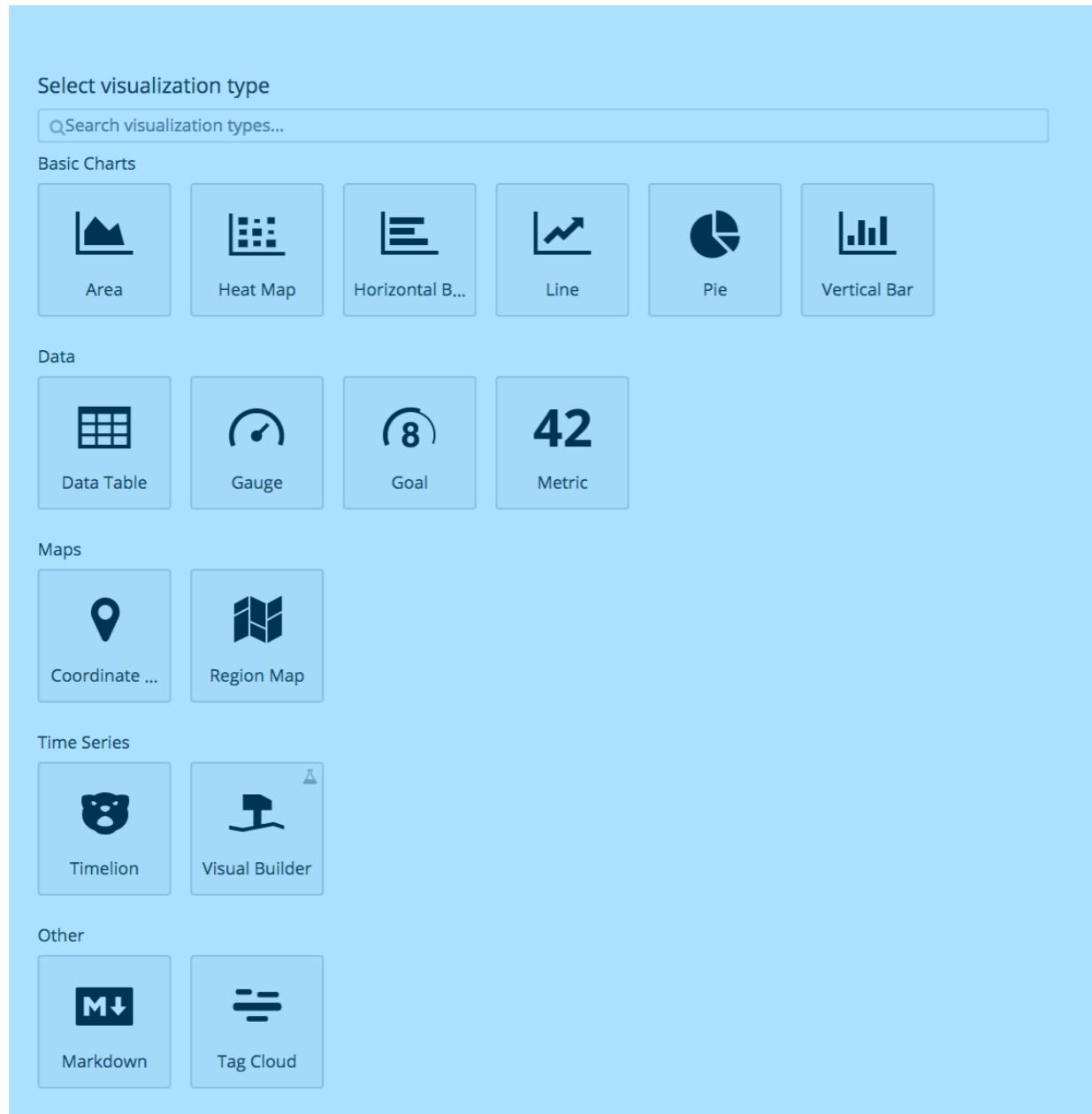
- *Time Picker : Year to Date*
- *Index Pattern : shopping*

번호	항목	체크
1	Document 정보 Table/JSON 형태로 보기	
2	Histogram에서 간격을 Daily로 설정	
3	(2번 설정을 유지한 상태로) 2018년 1월1일 ~ 2018년 1월31일 데이터만 조회	
4	다시 Time Picker를 Year to Date로 설정	
5	Histogram 데이터 csv 출력	
6	고객나이, 고객주소_시도, 상품분류 Field 정보만 보기	
7	고객나이가 적은 순으로 정렬	
8	Documents는 총 몇 개 인가?	
9	상품분류 Field의 Value 분포는 어떤가?	
10	배송메모 Field가 존재하는 Documents는 총 몇 개 인가?	

데이터 시각화 

어떤 시각화를 할 수 있을까?

공식



비공식

network

cohort

dendrogram

:

Kibana Visualize는 어렵나?



그럴 수 있다. 그렇다면 왜?



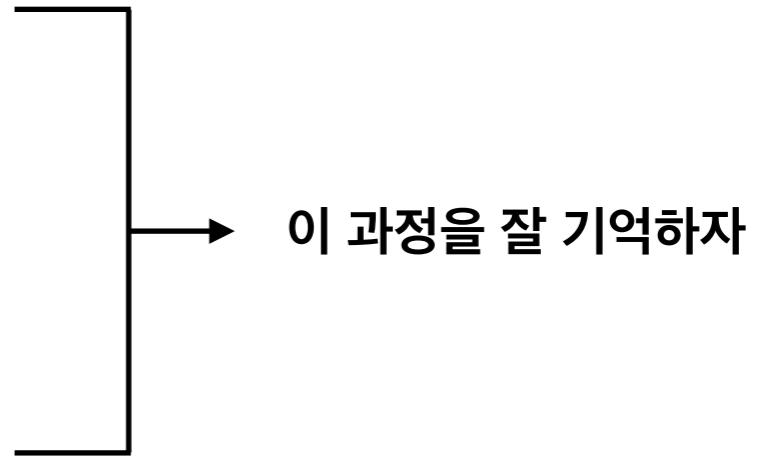
용어가 너무 낯설어서



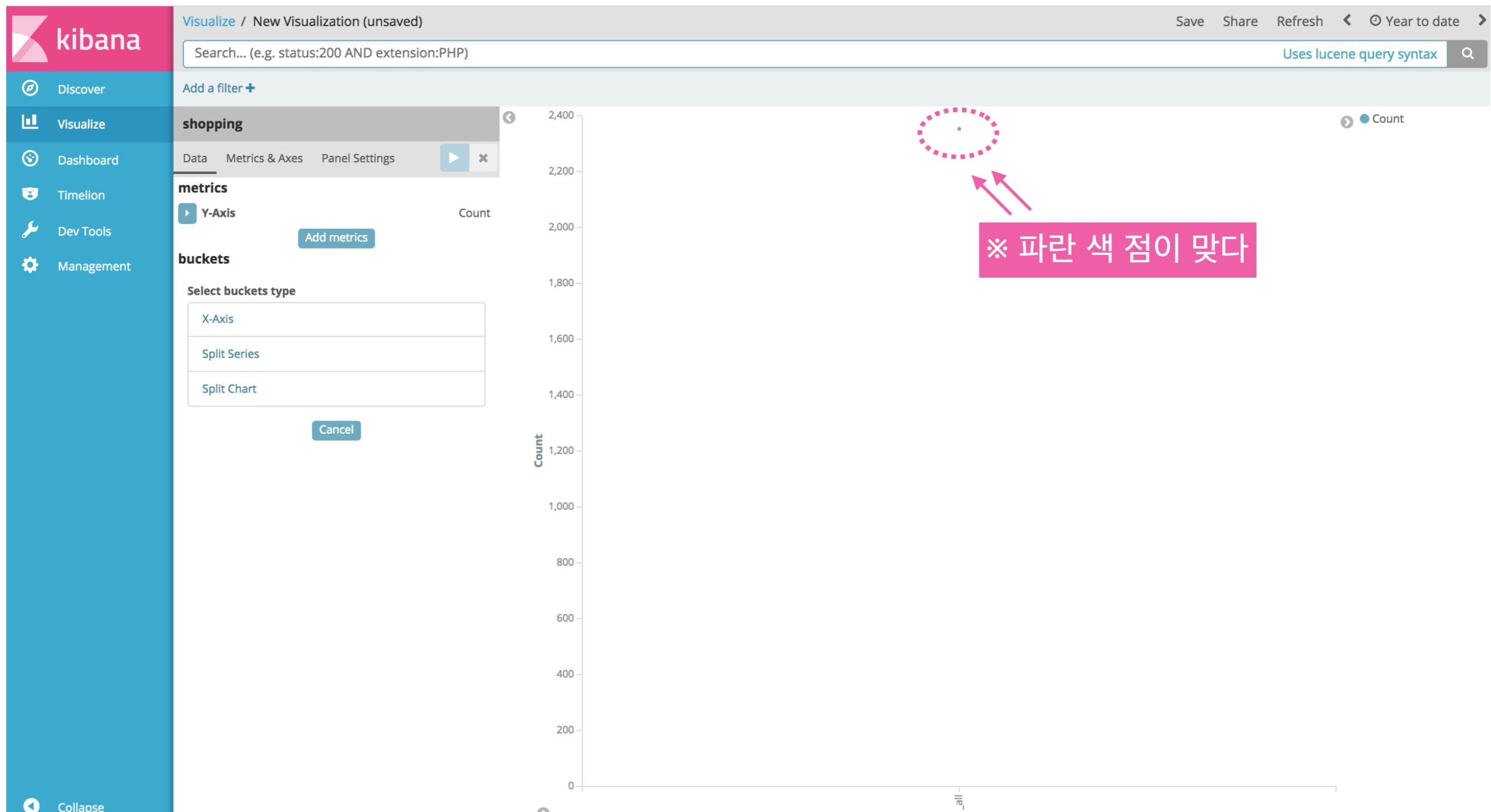
눈 딱 감고 맛보기로 1개만 따라해보자

Visualize 과정

- Kibana 접속
- Visualize 선택
- Create new visualization 선택
- Select visualization type - “Line 선택”
- From a New Search, Select Index - “shopping 선택”



그러면 다음과 같은 화면이 나온다



아직 아무 것도 안한거처럼 보이지만 실제로는 이미 하나의 Visualization을 생성했다.
파란색 점은 다음과 같은 과정으로 표시된 것이다.

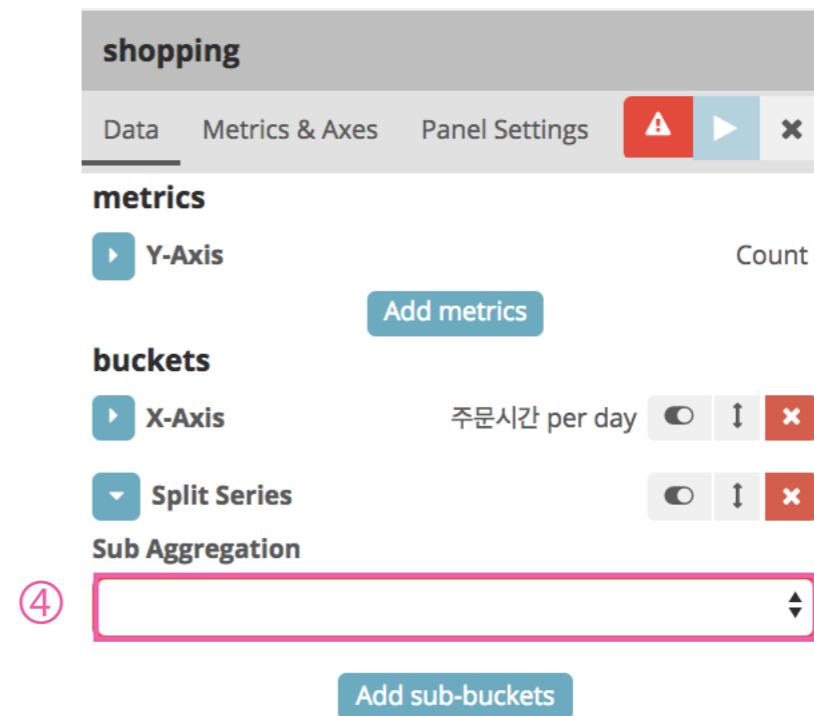
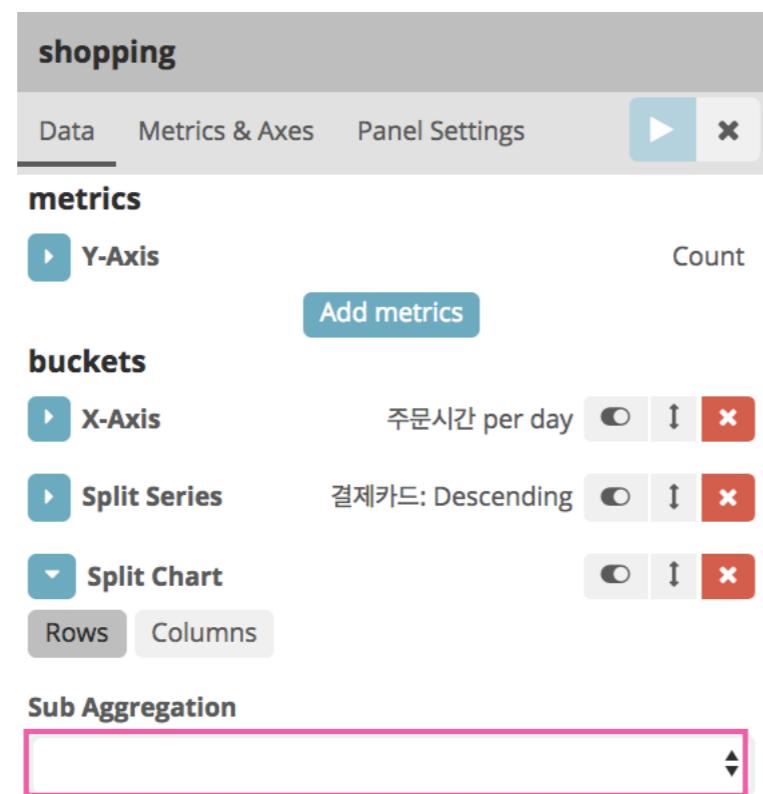
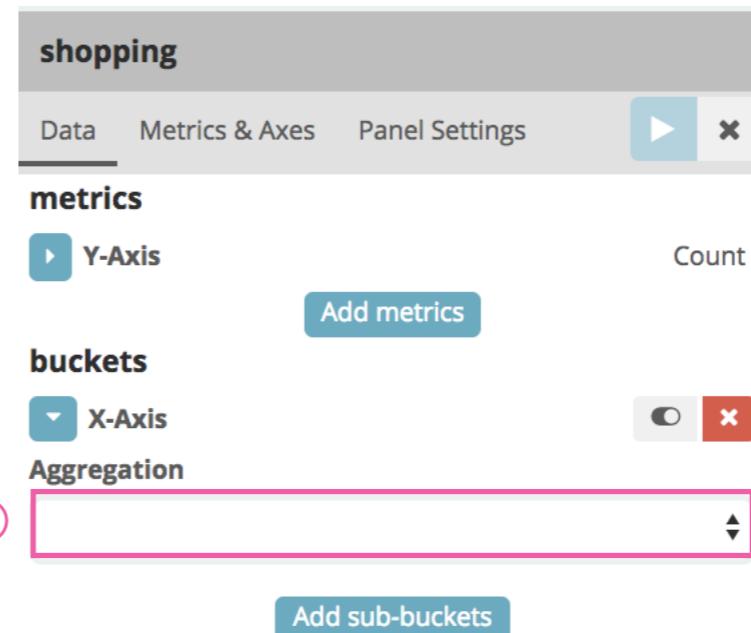
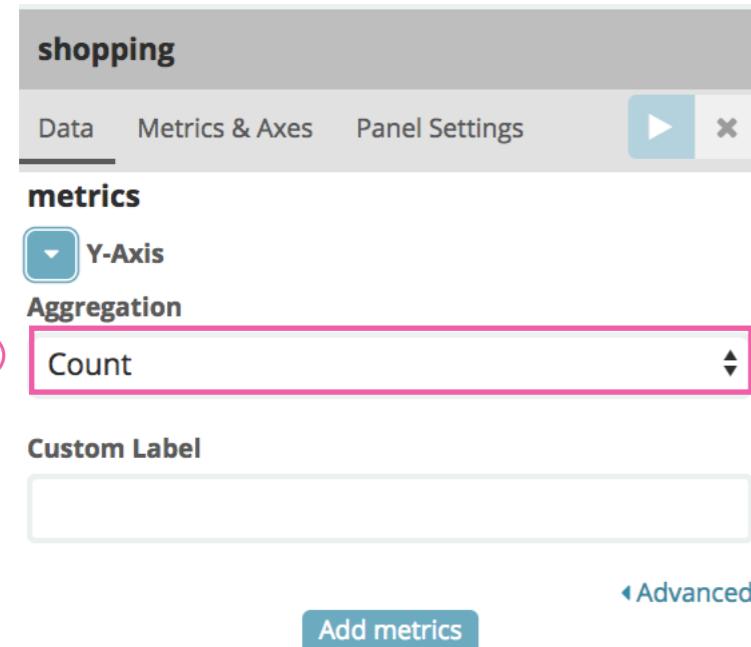
- shopping index에 있는 데이터 중에서
- year to date 기간에 해당하는 데이터만 선별해서
- count 한 후
- y축에 표시해라



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

!!

무얼 누르든 각종 aggregation이 나온다는 걸 볼 수 있다.



Aggregation - Bucket

쉽게 생각해서 Bucket을 Group이라고 생각하자.

그렇다면 Bucket Aggregation은 Grouping 작업 정도로 볼 수 있다.

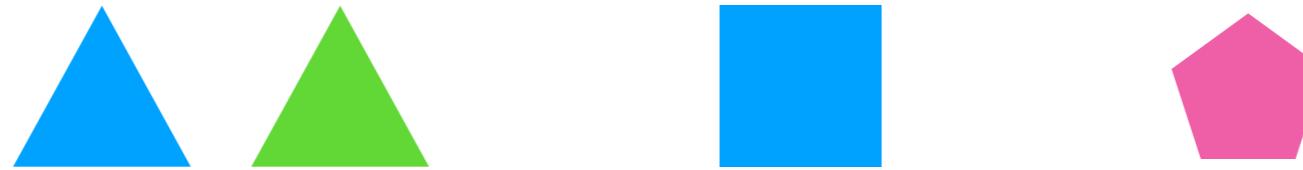
예를 들어 다음과 같은 도형이 있다고 하자.



이 때 위의 다각형을 여러개의 그룹으로 나눠야 한다면 어떻게 할 수 있을까?

다양하게 나눌 수 있겠지만 어떤 방법을 택하든 가장 먼저 하는 작업은 기준을 정하는 일이다.

내각의 합



색



마찬가지로 bucket aggregation이 하는 역할도 위와 같이 데이터를 일정한 기준으로 나누는 것이다.
kibana에서 사용 가능한 bucket aggregation 종류가 어떤 것이 있는지 살펴보자

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": "2017-11-06T22:51:39",  
    "ip": "27.119.249.209",  
    "좌표": "37.23486, 126.60655",  
    "가격": 26000,  
    "분류": "청바지",  
    "성별": "여성"  
},  
{  
    "시간": "2018-01-06T13:51:39",  
    "ip": "27.119.249.209",  
    "좌표": "37.2299, 126.59903",  
    "가격": 35000,  
    "분류": "니트",  
    "성별": "남성"  
},  
:  
:
```

Date Range/Histogram Agg



2017년 // 2018년

IPv4 Range Agg



0.0.0.0 ~ 127.255.255.255 // 128.0.0.0 ~ 191.255.255.255

Range, Histogram Agg



0 < 가격 < 20000 // 20000 < 가격 < 40000

Terms Agg



남성 // 여성

Filter Agg



청바지 // 여성 // 2018년 이후

Aggregation - Metric

Bucket은 일종의 Grouping 작업이라고 했다.

다만 우리가 무언가를 시각화 한다고 했을 때 Grouping만으로는 유의미한 결과를 볼 수 없다.

Bucket Aggregation에서 사용했던 예를 다시 보자

색



위와 같이 색을 기준으로 Group을 나누면 난 후, 각종 집계 정보까지 함께 볼 수 있어야 유의미하다고 할 수 있다.

	파란색	녹색	자주색
개수	2	1	1
내각의 합의 평균	270	180	540

집계된 결과를 바탕으로 (대부분의) Visualization이 이루어진다는 점에서 Metric Aggregation은 중요하다.

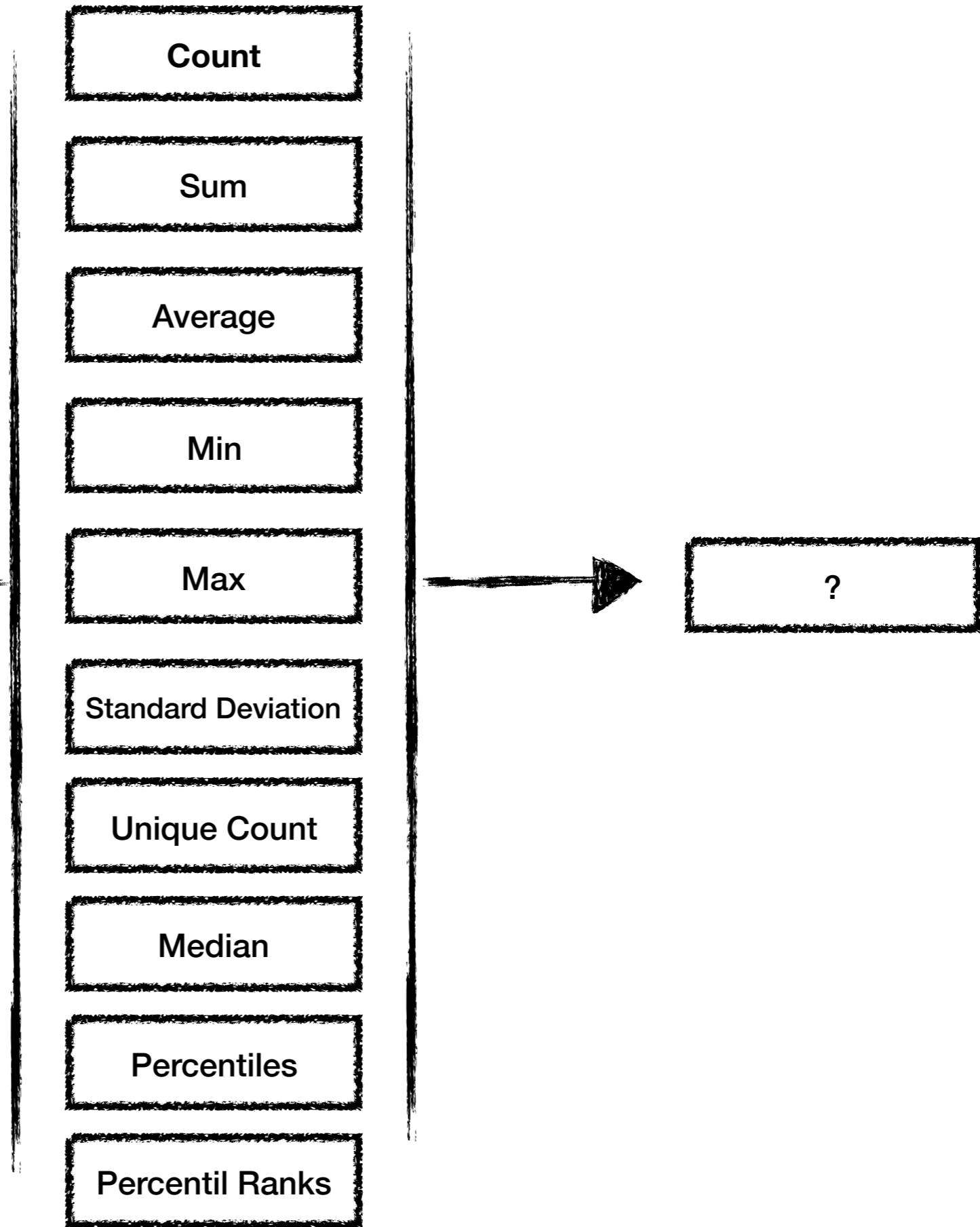
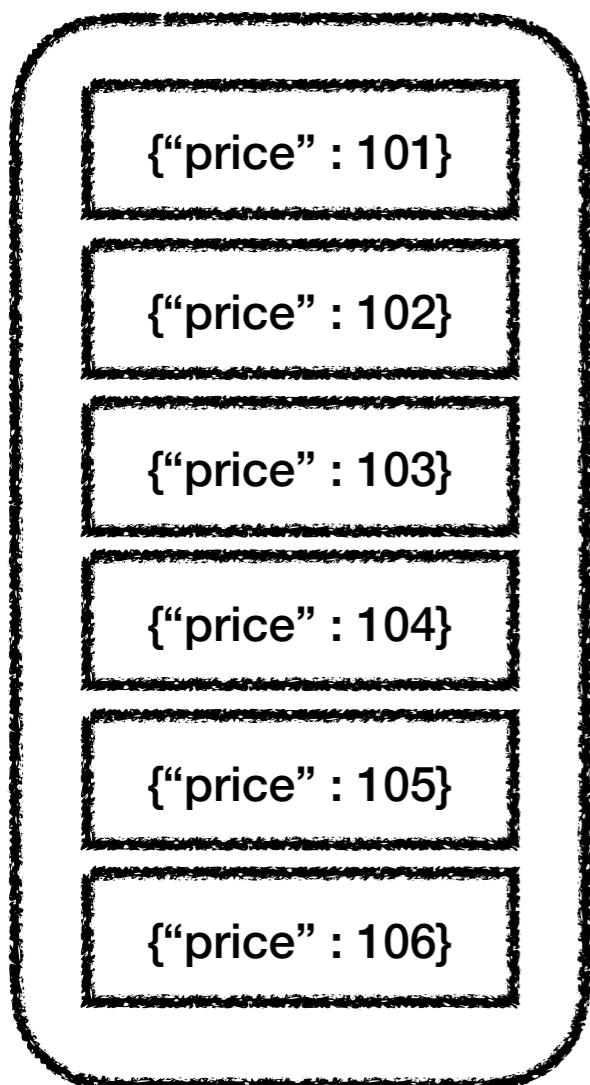
Kibana에서 사용할 수 있는 Metric Aggregation이 어떤 것이 있는지 살펴보자

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

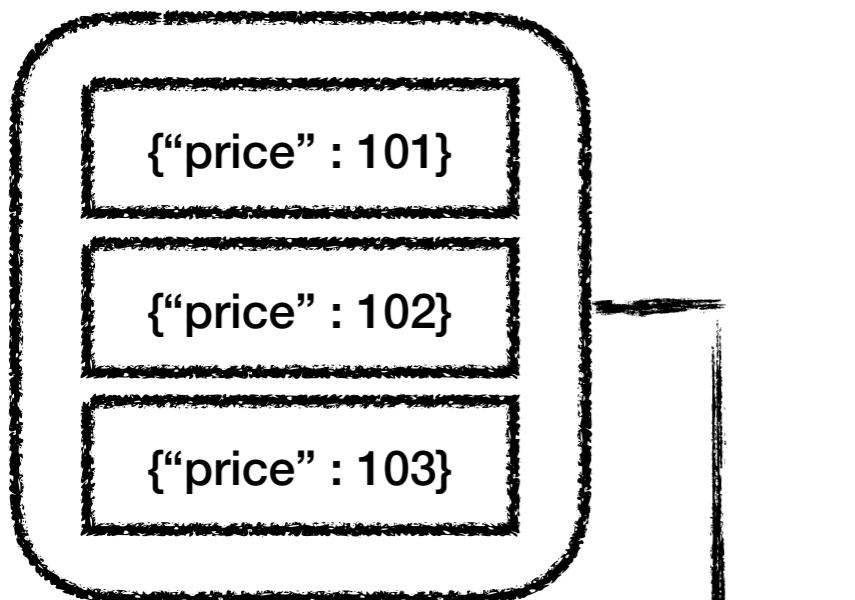


- Number Field : Concat, Sum, Min, Max, Count
- 기타 Field : Concat

Single Bucket



Bucket 1



Count

Sum

Average

Min

Max

Standard Deviation

Unique Count

Median

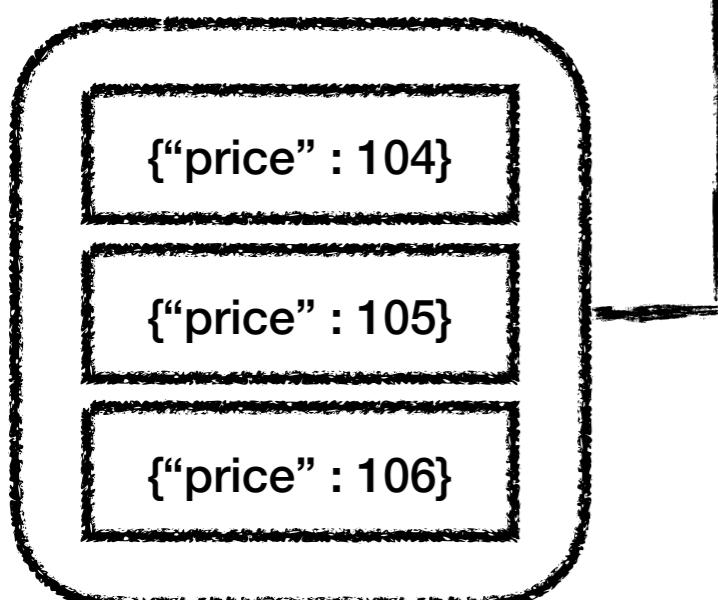
Percentiles

Percentil Ranks

?

?

Bucket 2



{“번호” : 1, “날짜” : “10-01”, “역” : 강남 }

{“번호” : 2, “날짜” : “10-02”, “역” : 신사 }

{“번호” : 3, “날짜” : “10-03”, “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

Aggregation도 해봤으니 이제 Visualization도 바로 할 수 있을까?

Data Table

The screenshot shows the Kibana Visualize interface with a search bar at the top. On the left, there's a sidebar with icons for Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management. A red box highlights the 'Visualize' icon. The main area shows a visualization titled 'shopping' with a 'Count' metric set to 'Count'. The value is 1,356. Below this, there's a section for 'Select buckets type' with two options: 'Split Rows' and 'Split Table'. A green box highlights the 'Split Rows' option, and a green hand icon points to it. A red box highlights the 'buckets' section. At the bottom, there are 'Cancel' and 'Export' buttons.

Visualize / New Visualization (unsaved)

Save Share Refresh ⏪ ⏩ Last 90 days

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

Discover

Visualize

Dashboard

Timelion

Dev Tools

Management

shopping

Count

1,356

Add a filter +

metrics

Metric

buckets

Select buckets type

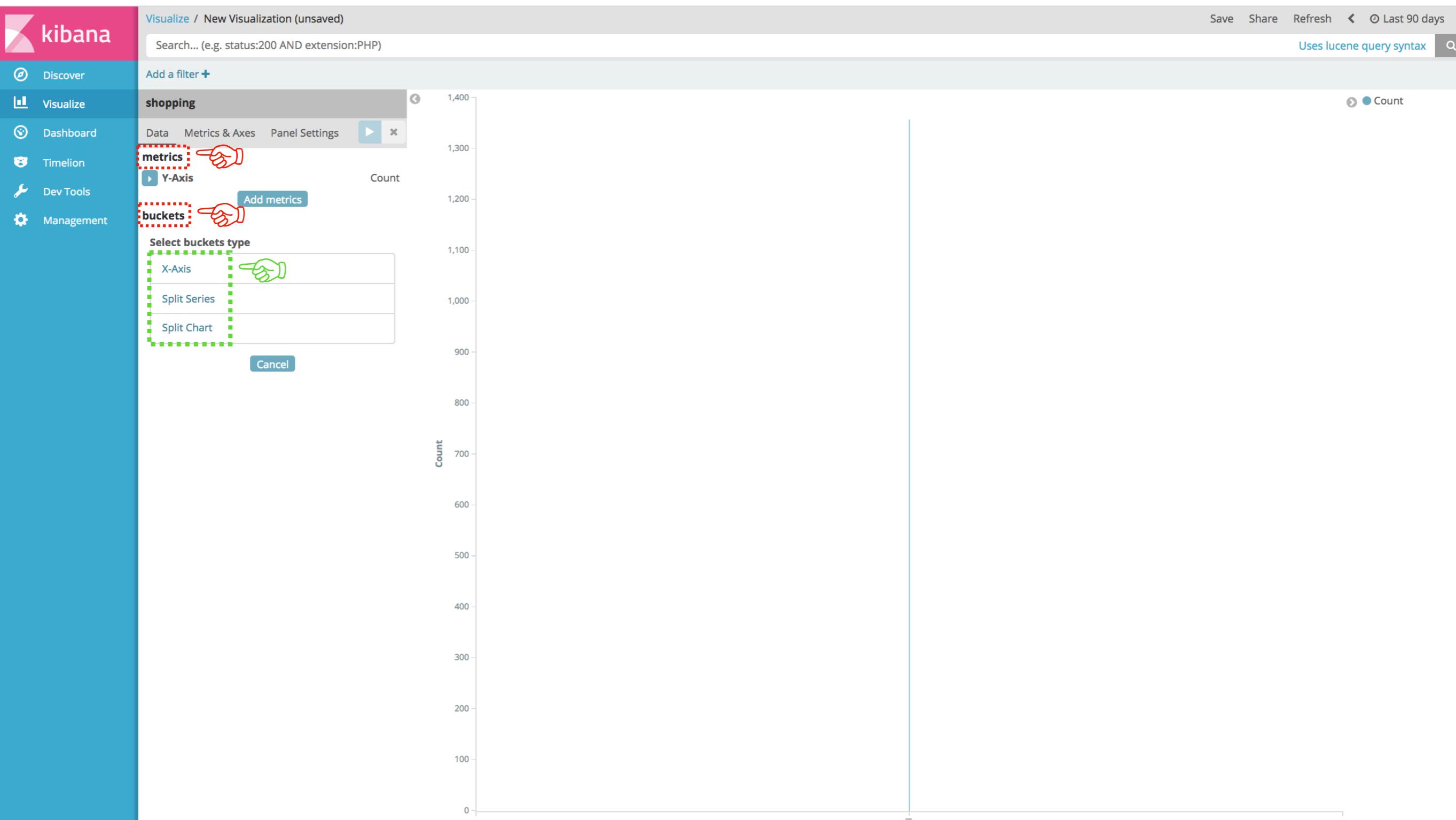
Split Rows

Split Table

Cancel

Export: Raw Formatted

Area



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

시각화하려는 문제는 명확한데,

어디에 들어가서 어떻게 조작해야되는지 모르겠다

1. 큰 틀은 비슷하다

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

buckets : 위의 metrics를 적용할 그룹을 정의하는 부분

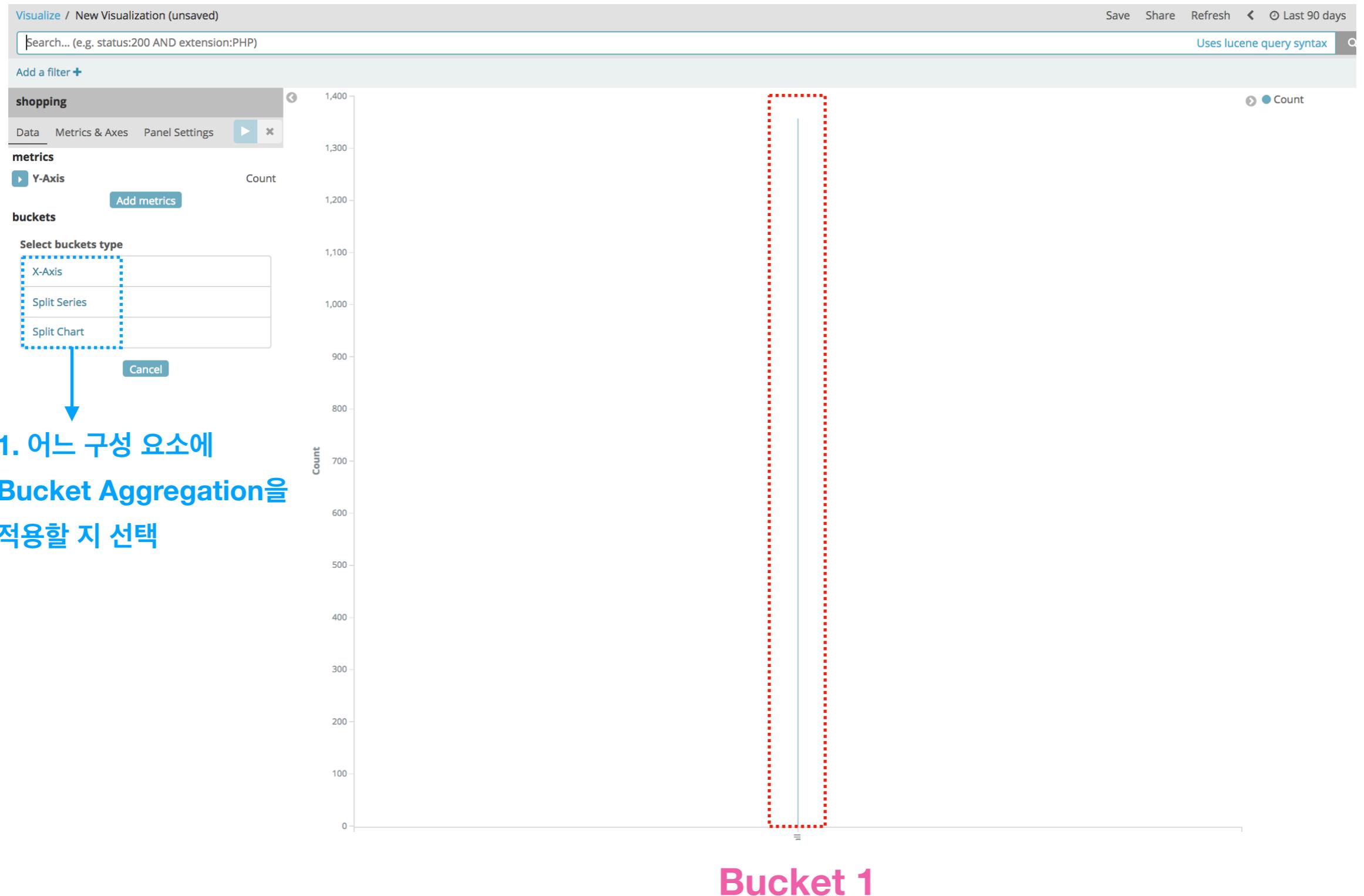
예: 전국 학생들의 지역별 평균 키를 구한다고 하자

키의 평균을 구하는 작업 : metrics

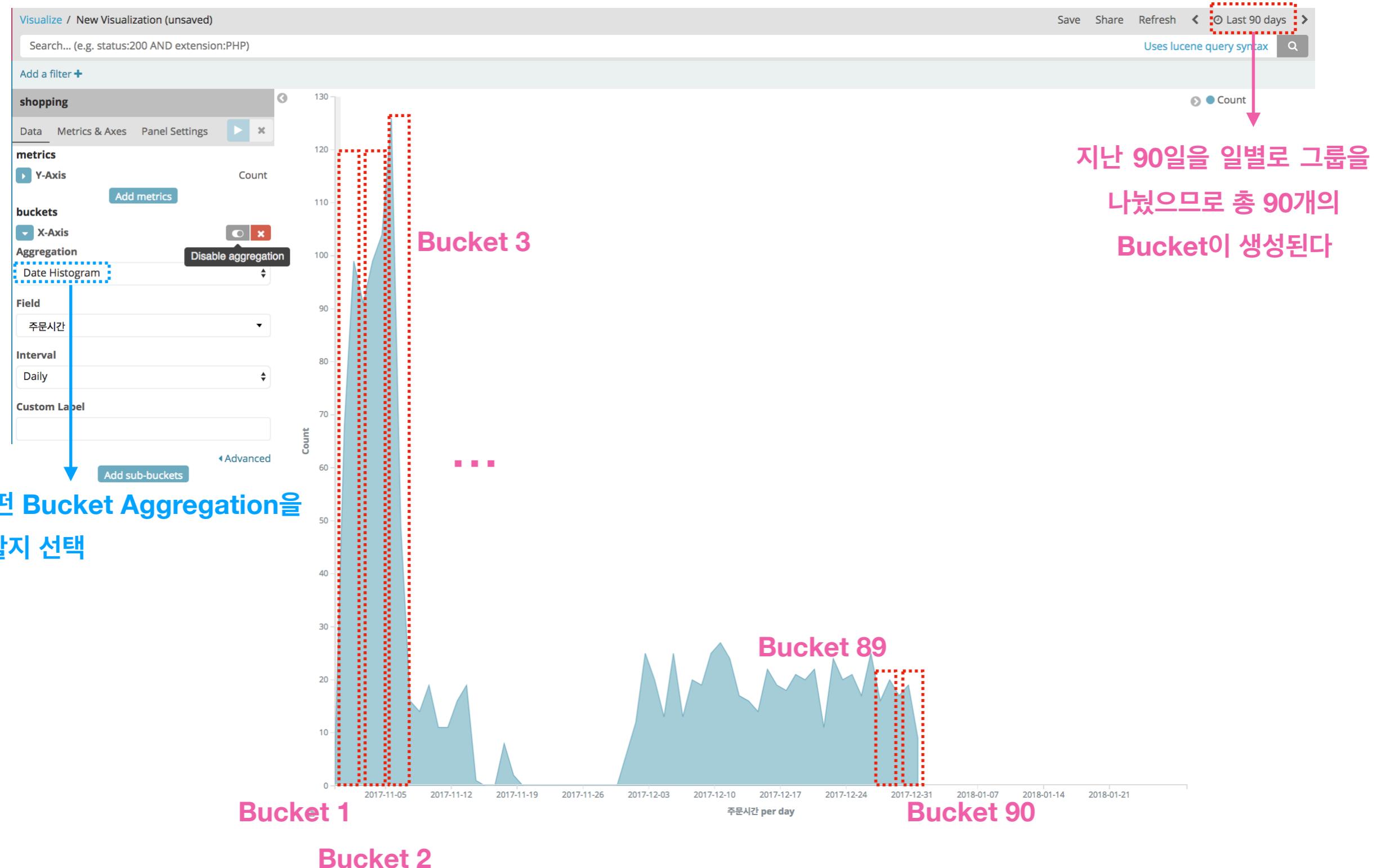
학생들을 지역별로 나누는 작업 : buckets

2. 개별적 구성요소는 Visualization Type마다 상이할 수 있다

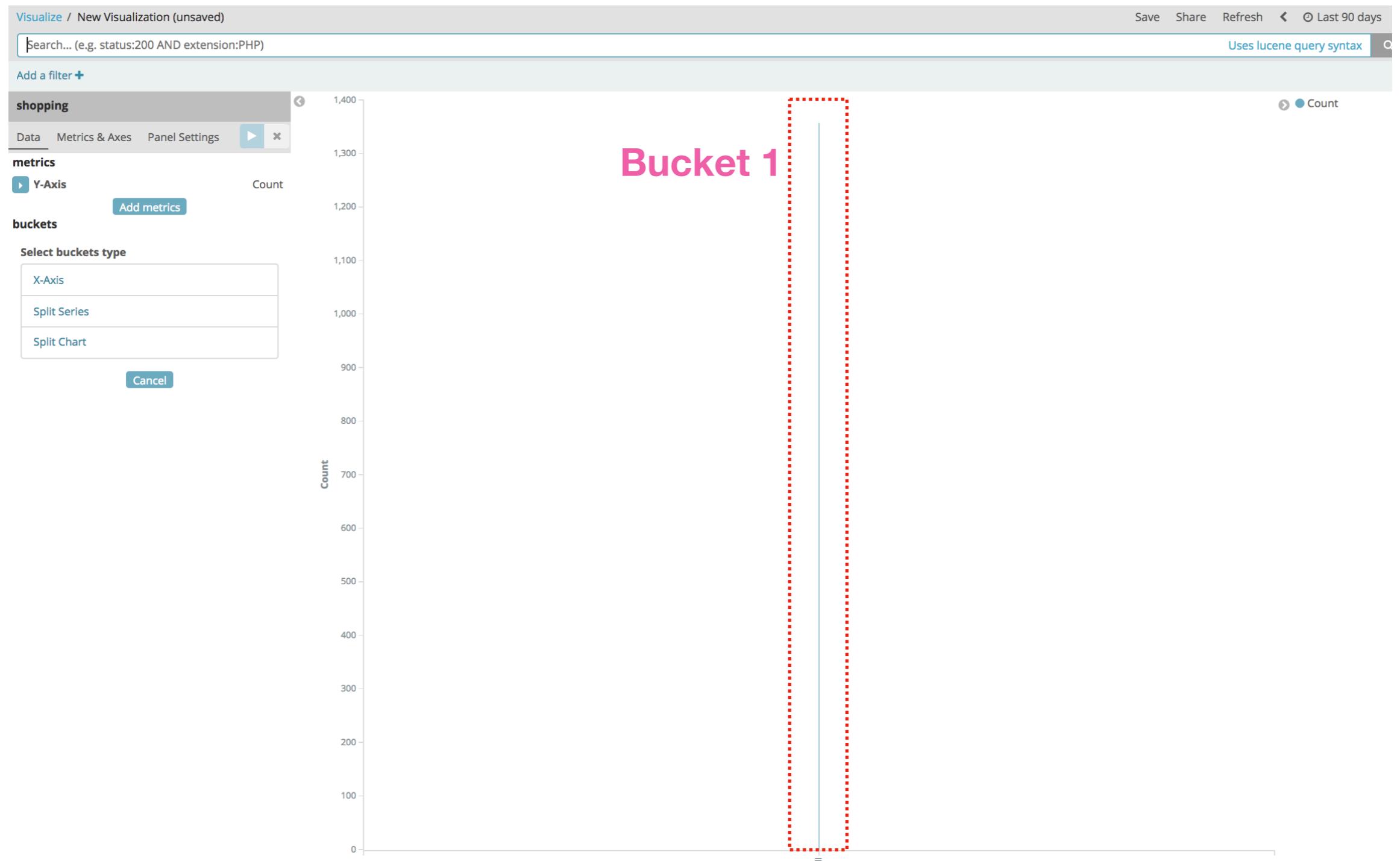
대표적인 buckets type 몇 개를 살펴보자 X-Axis | Before



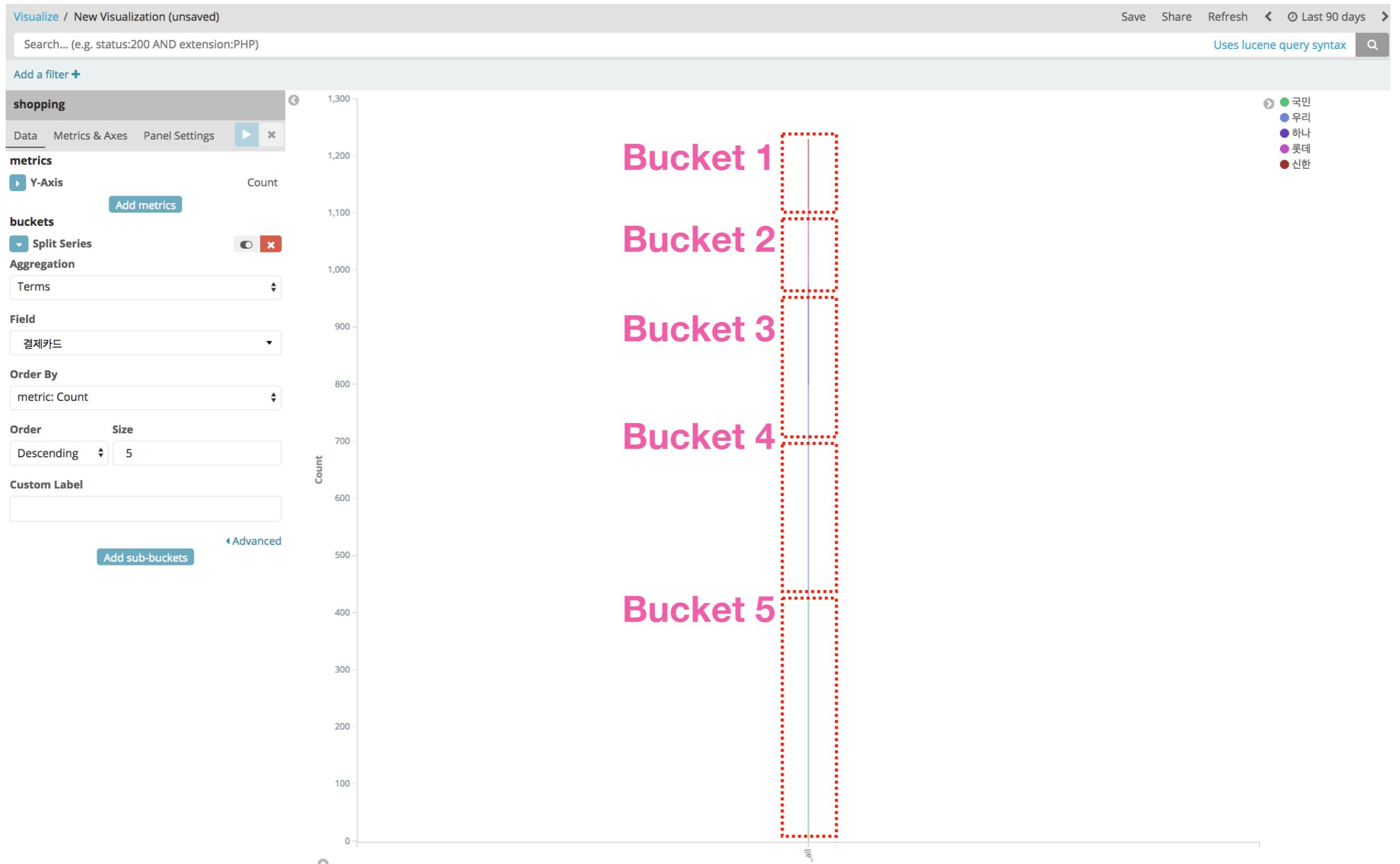
대표적인 buckets type 몇 개를 살펴보자 X-Axis After



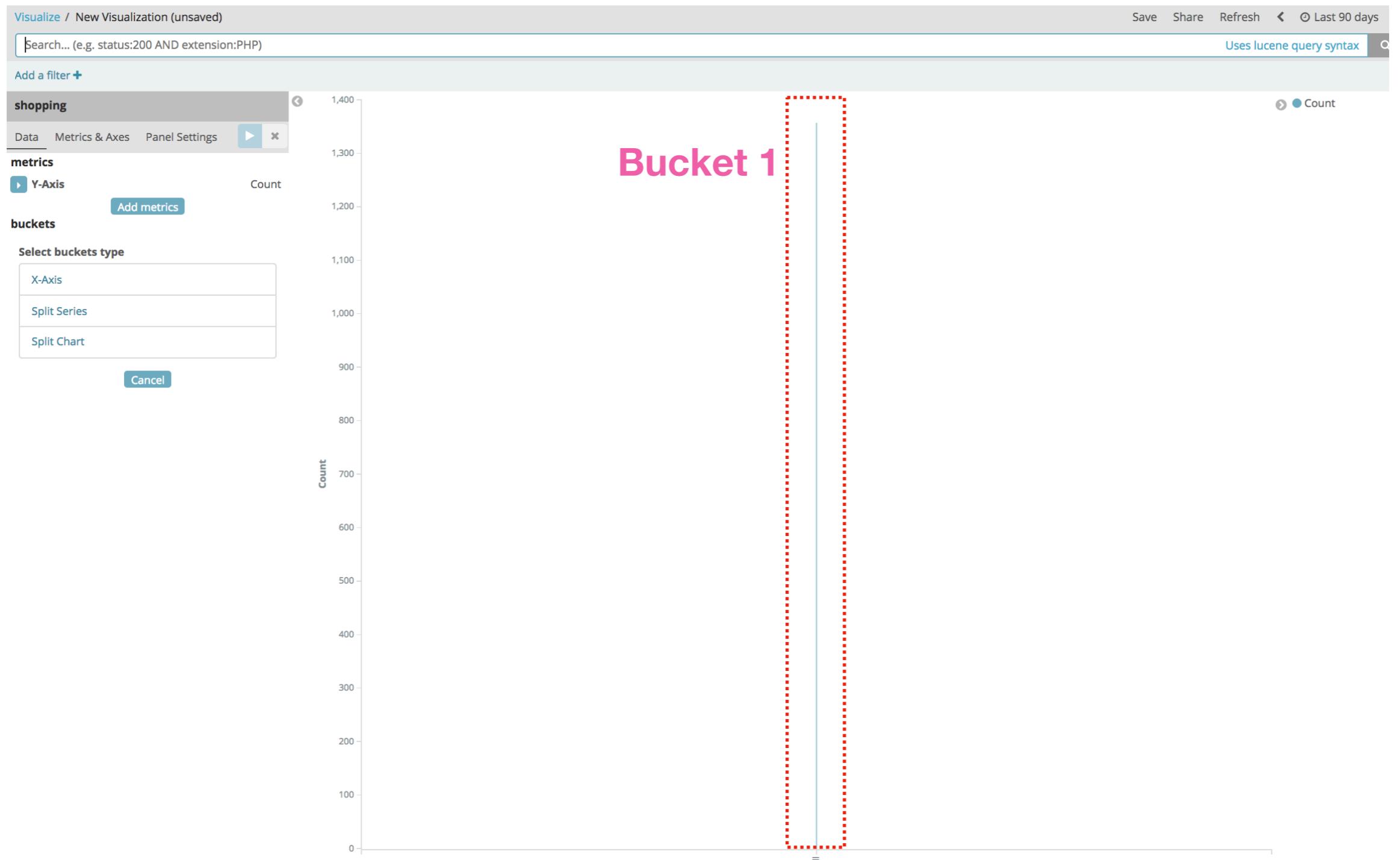
대표적인 buckets type 몇 개를 살펴보자 Split Series | Before



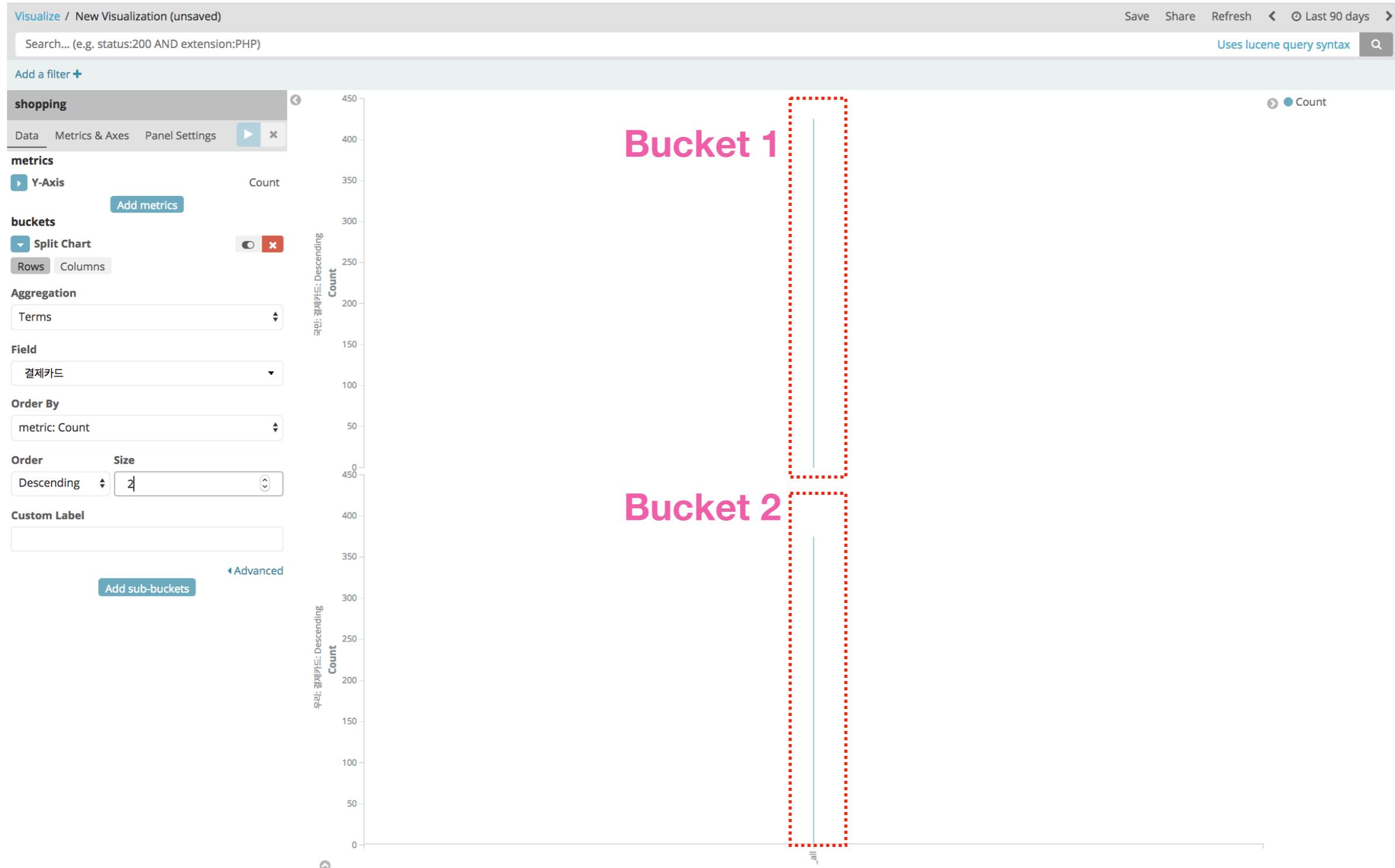
대표적인 buckets type 몇 개를 살펴보자 Split Series After



대표적인 buckets type 몇 개를 살펴보자 Split Chart | Before



대표적인 buckets type 몇 개를 살펴보자 Split Chart After



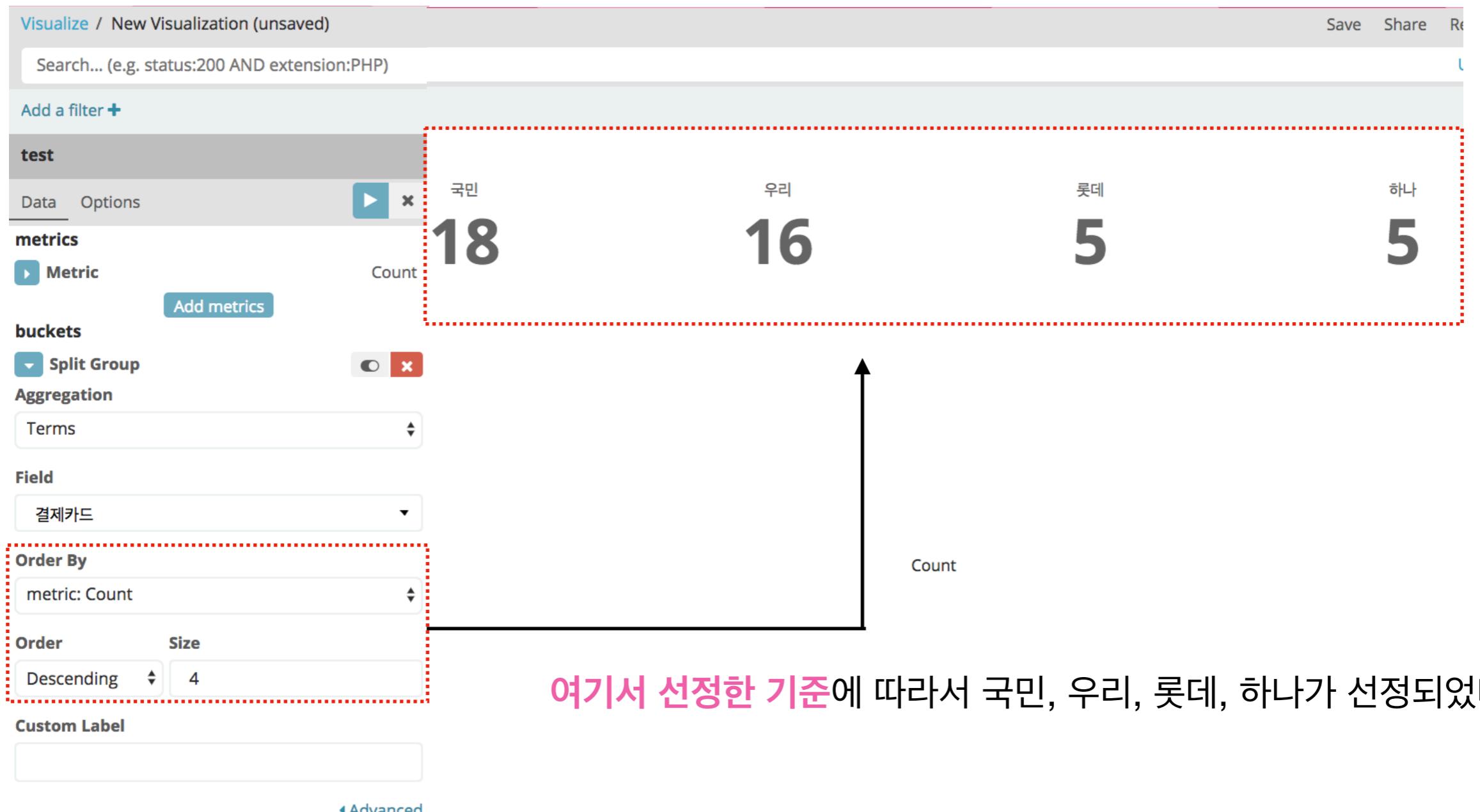
보통은 이 작업의 반복이지만 **Term Aggregation**으로
Bucket을 나눌 경우 한 단계 더 고려해야한다

- 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



test

Data Options ▶ ×

metrics

▶ Metric Count Add metrics

buckets

▼ Split Group Toggle ×

Aggregation

Terms

Field

결제카드 → 1. 결제카드로 Bucket을 구분해서... 국민 | 하나 | 신한 | 롯데 | 시티 | 우리 ...

Order By

metric: Count → 2. Bucket 별 Count를 구하고... 국민 | 하나 | 신한 | 롯데 | 시티 | 우리 ...
18 | 5 | 3 | 5 | 2 | 16 ...

Order Size

Descending ▼ 4 → 4. 상위 4개를 선정해라 국민 | 우리 | 롯데 | 하나
18 | 16 | 5 | 5

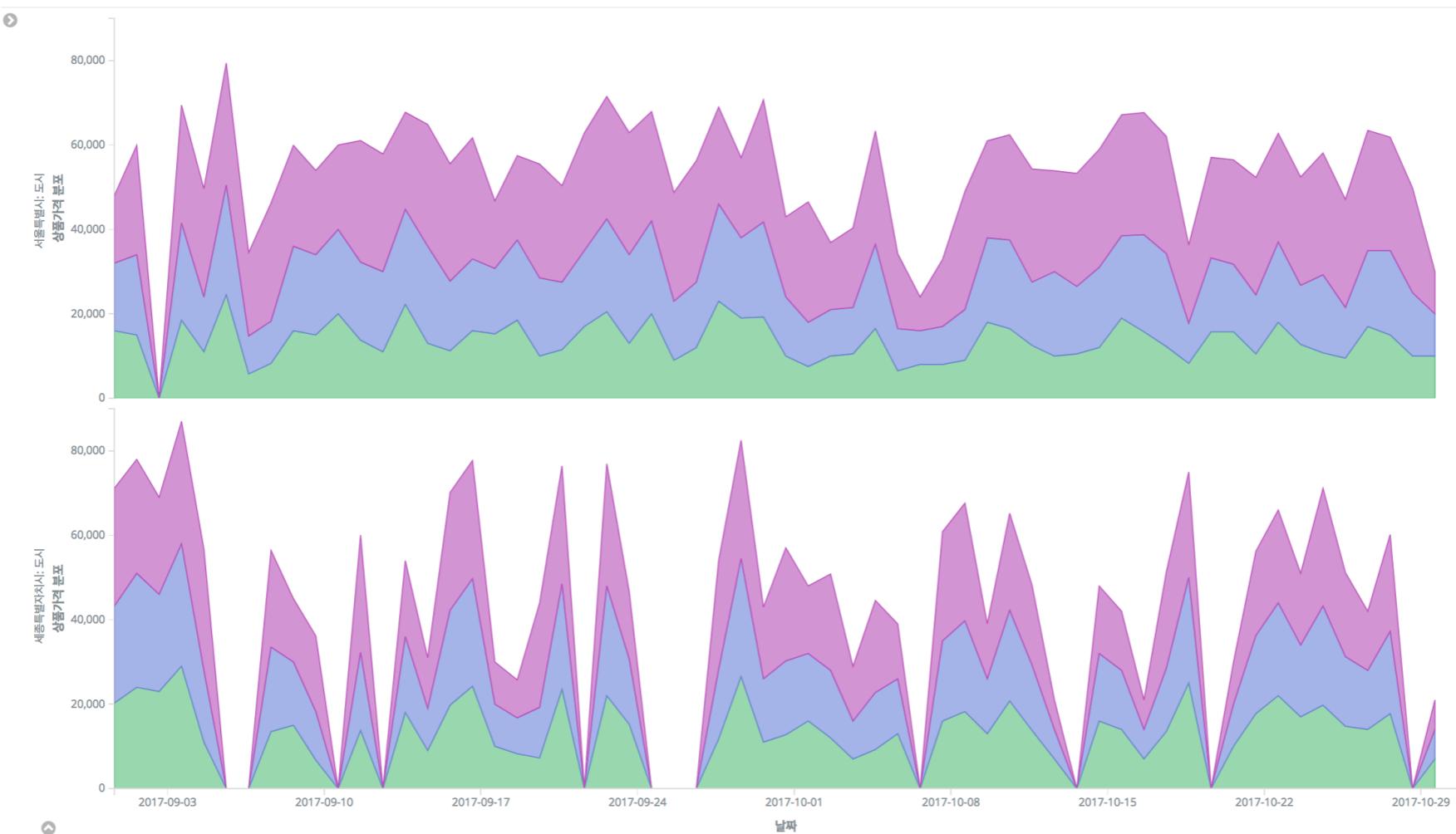
Custom Label → 3. Count가 큰 순으로 정렬해서... 국민 | 우리 | 롯데 | 하나 | 신한 | 시티 ...
18 | 16 | 5 | 5 | 3 | 2 ...

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 25분위, 50분위, 95분위를
- “주문시간”을 기준으로 daily로 표시



문제에서 metrics 영역과 buckets 영역으로 구분한다

문제

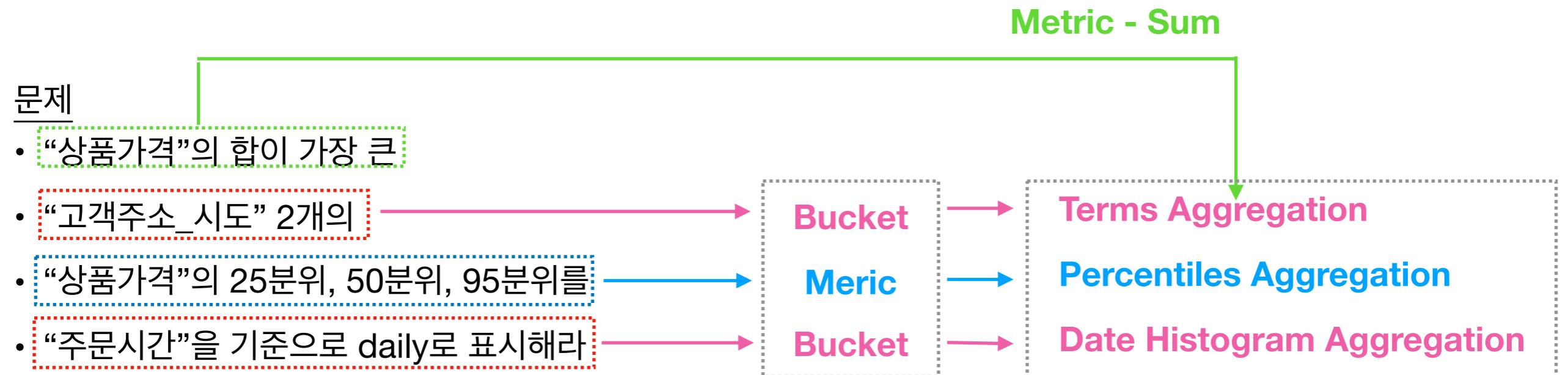
- “상품가격”의 합이 가장 큰 Bucket
- “고객주소_시도” 2개의 Bucket
- “상품가격”의 25분위, 50분위, 95분위를 Meric
- “주문시간”을 기준으로 daily로 표시해라 Bucket

metrics와 buckets 내에서 사용할 aggregation을 선택한다

문제

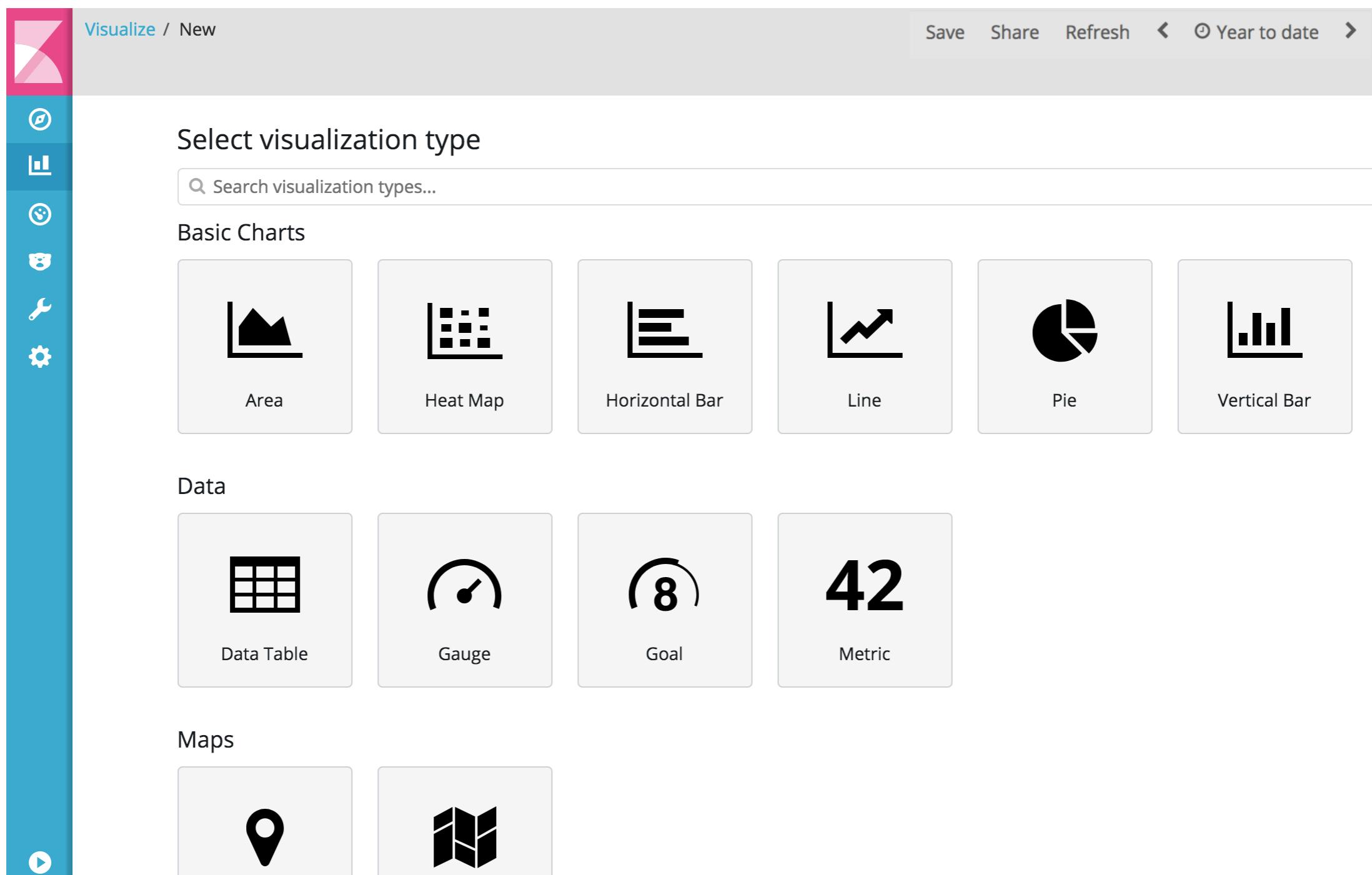
- “상품가격”의 합이 가장 큰
 - “고객주소_시도” 2개의
 - “상품가격”의 25분위, 50분위, 95분위를
 - “주문시간”을 기준으로 daily로 표시해라
-
- ```
graph LR; Q1["상품가격의 합이 가장 큰"] --> A1["Terms Aggregation"]; Q2["고객주소_시도 2개의"] --> A2["Percentiles Aggregation"]; Q3["상품가격의 25분위, 50분위, 95분위를"] --> A3["Date Histogram Aggregation"]; Q4["주문시간을 기준으로 daily로 표시해라"] --> A4["Bucket"]; Q5["Merica"] --> A5["Bucket"]
```
- The diagram illustrates the mapping of search queries to Elasticsearch aggregations. On the left, four queries are listed with their corresponding aggregation types on the right. The first query, "상품가격"의 합이 가장 큰, is mapped to "Terms Aggregation". The second query, "고객주소\_시도" 2개의, is mapped to "Percentiles Aggregation". The third query, "상품가격"의 25분위, 50분위, 95분위를, is mapped to "Date Histogram Aggregation". The fourth query, "주문시간"을 기준으로 daily로 표시해라, is mapped to "Bucket". Additionally, there is a separate entry "Merica" which also maps to a "Bucket". Arrows connect each query to its respective aggregation type.

term aggregation으로 bucket을 나눌 경우 sorting을 위한 aggregation을 정의한다



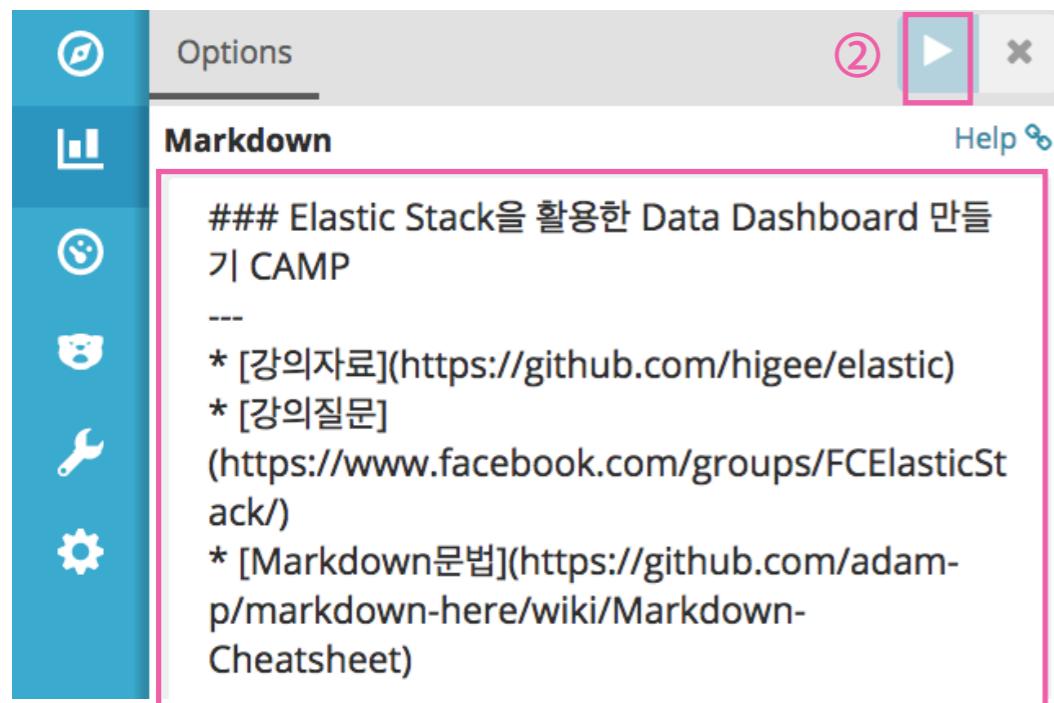
## 이제는 실제로 직접 해보면서 익혀보자

- 단, 특정한 목적을 가진 Visualization을 만들기보다는 우선 Visualization에 익숙해지는 걸 목표로 하자
- Time Picker는 Year to Date로 설정하자 (조회하는 시점에 따라 데이터 다소 다를 수 있음)



# Markdown

- 안내사항 등 메세지를 남기고 싶은 경우 markdown syntax를 이용해서 작성



①

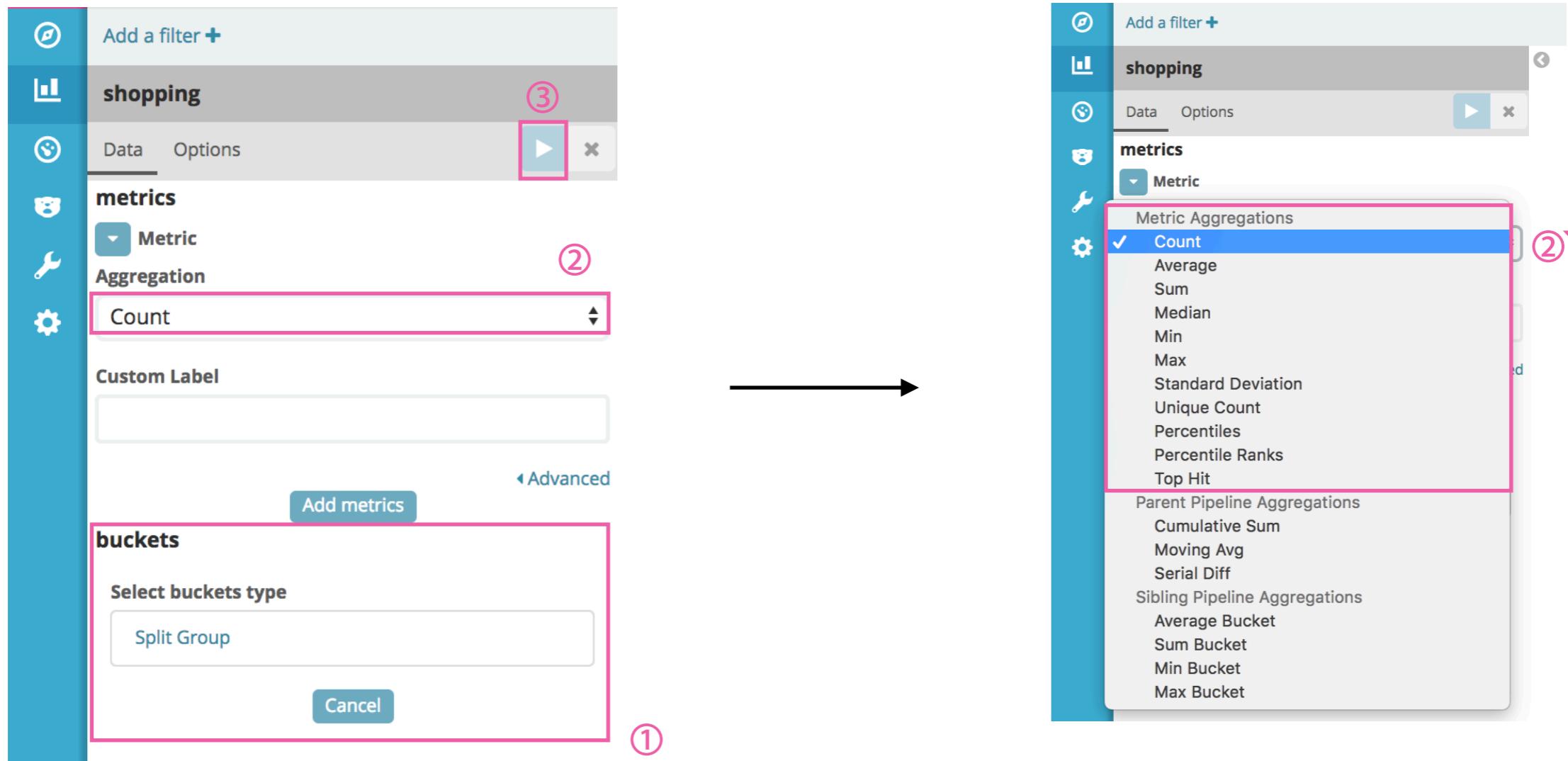
Elastic Stack을 활용한 Data Dashboard 만들기 CAMP

- 강의자료
- 강의질문
- Markdown문법

1. ①에 적당한 내용을 입력하자 🎉
2. ②를 누르고 결과를 확인하자

# Metric

- KPI 같은 지표를 숫자 형태로 보고 싶을 때 사용
- 우선 buckets는 고정하고 다양한 metric aggregation을 학습하자



1. buckets은 고정하기 위해 ①은 원래 상태로 두자
2. ②를 눌러서 ②' 중 하나씩 골라보자
3. ③을 눌러서 결과를 확인하자
4. ②'에 있는 aggregation을 다 해볼 때 까지 2와 3을 반복하자

# Metric - count

The screenshot shows the Elasticsearch Metrics interface for the 'shopping' index. The 'metrics' section has 'Metric' expanded, and 'Count' is selected under 'Aggregation'. A pink box highlights the 'Count' dropdown. The 'Custom Label' field contains the Korean word '개수' (Count). An arrow points from the interface to the result '2,404' on the right, which is also labeled '개수' below it. A pink box highlights the 'buckets' section, which is currently set to 'Split Group'. A callout box on the right provides context:

- shopping index 중에서
- year to date 기간 동안의
- documents 개수

1. buckets은 고정하기 위해 ①은 원래 상태로 두자
2. ② 눌러서 Count aggregation 선택
3. ③ 눌러서 시작화

# Metric - average

The screenshot shows a search interface with the following configuration:

- shopping** (highlighted with a pink box)
- Data Options** (highlighted with a pink box)
- ④** (highlighted with a pink box)
- metrics**
- Metric** (highlighted with a pink box)
- Aggregation**
- ② Average** (highlighted with a pink box)
- Field**
- ③ 상품가격** (highlighted with a pink box)
- Custom Label**
- 평균가격**
- Advanced**
- Add metrics**
- buckets** (highlighted with a pink box)
  - Select buckets type
  - Split Group
  - Cancel



16,914.3

평균가격

- shopping index 중에서
- year to date 기간 동안의
- 모든 “상품가격” field의 value를
- average한 결과

1. buckets은 고정하기 위해 ①은 원래 상태로 두자
2. ②를 눌러서 Average aggregation 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 시각화

# Metric - sum

shopping

Data Options

metrics

Metric

Aggregation

② Sum

Field

③ 상품가격

Custom Label

상품가격합계

① buckets

Select buckets type

Split Group

Cancel

Add metrics

Advanced



40,668,000

상품가격합계

- shopping index 중에서
- year to date 기간 동안의
- 모든 “상품가격” field의 value를
- sum한 결과

1. buckets은 고정하기 위해 ①은 원래 상태로 두자
2. ②를 눌러서 sum aggregation 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 시각화

# Metric - Median

The screenshot shows a search interface with the following configuration:

- shopping** (highlighted with a pink box)
- Data Options** (highlighted with a pink box)
- metrics** (highlighted with a pink box)
- Metric** dropdown: **Median** (highlighted with a pink box)
- Aggregation** dropdown: **Field** (highlighted with a pink box)
- Field** dropdown: **상품가격** (highlighted with a pink box)
- Custom Label**: 상품가격 중위값
- buckets** (highlighted with a pink box):
  - Select buckets type: Split Group
  - Add metrics** button
  - Cancel** button
- Advanced** link

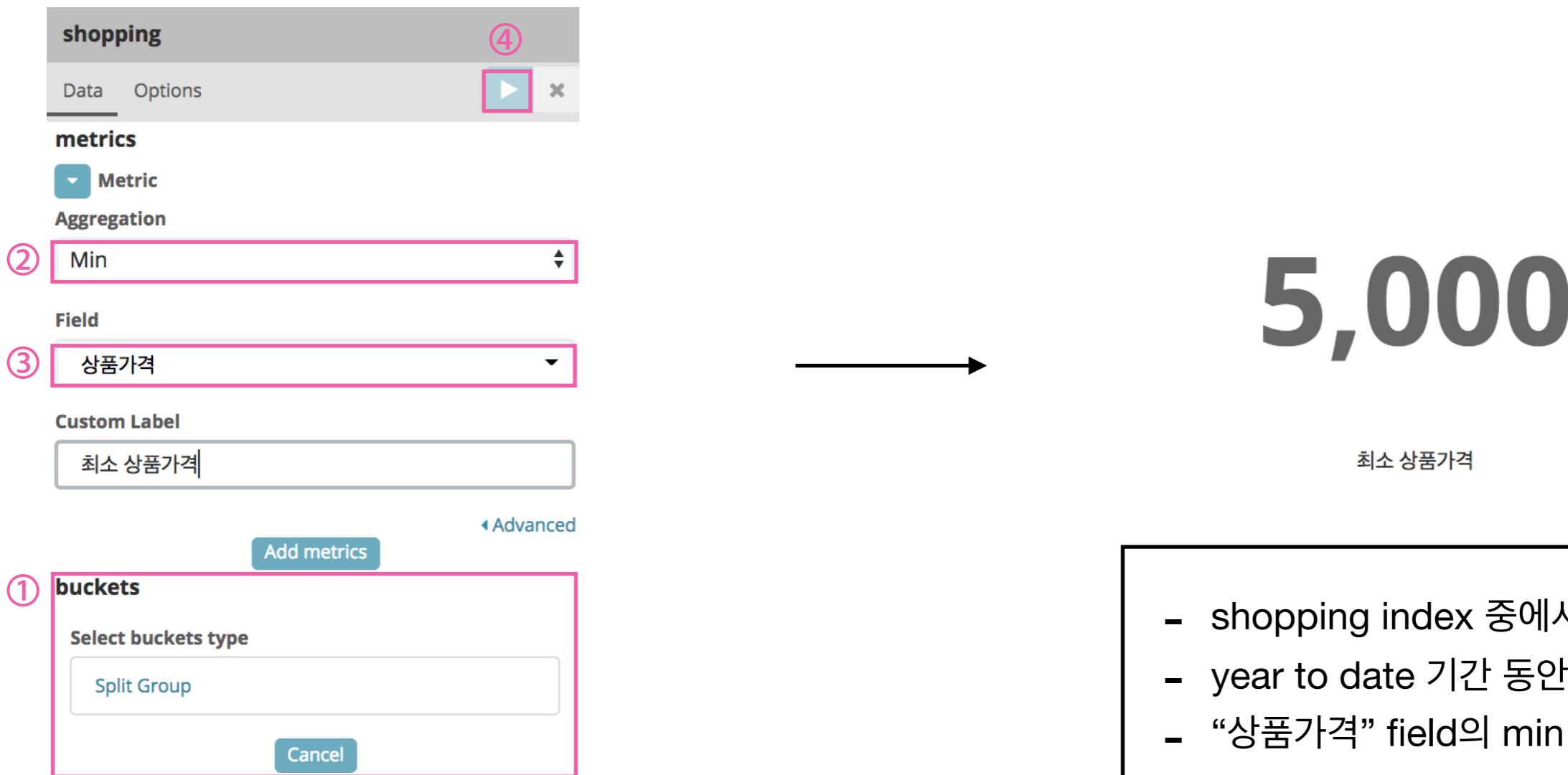
17,000

50th percentile of 상품가격

- shopping index 중에서
- year to date 기간 동안의
- “상품가격” field의 중위값

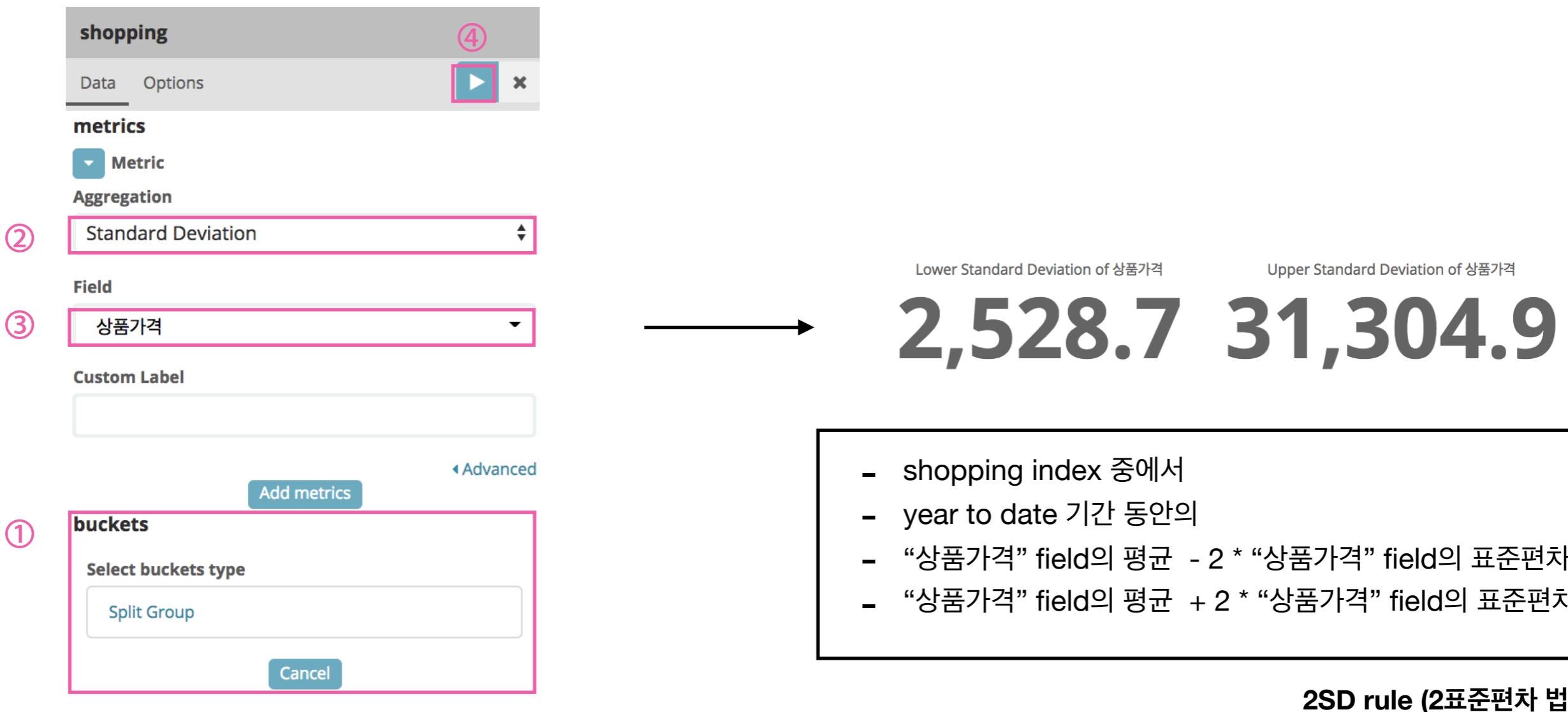
1. buckets은 고정하기 위해 ①은 원래 상태로 두자
2. ②를 눌러서 Median aggregation 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 시각화

# Metric - min/max



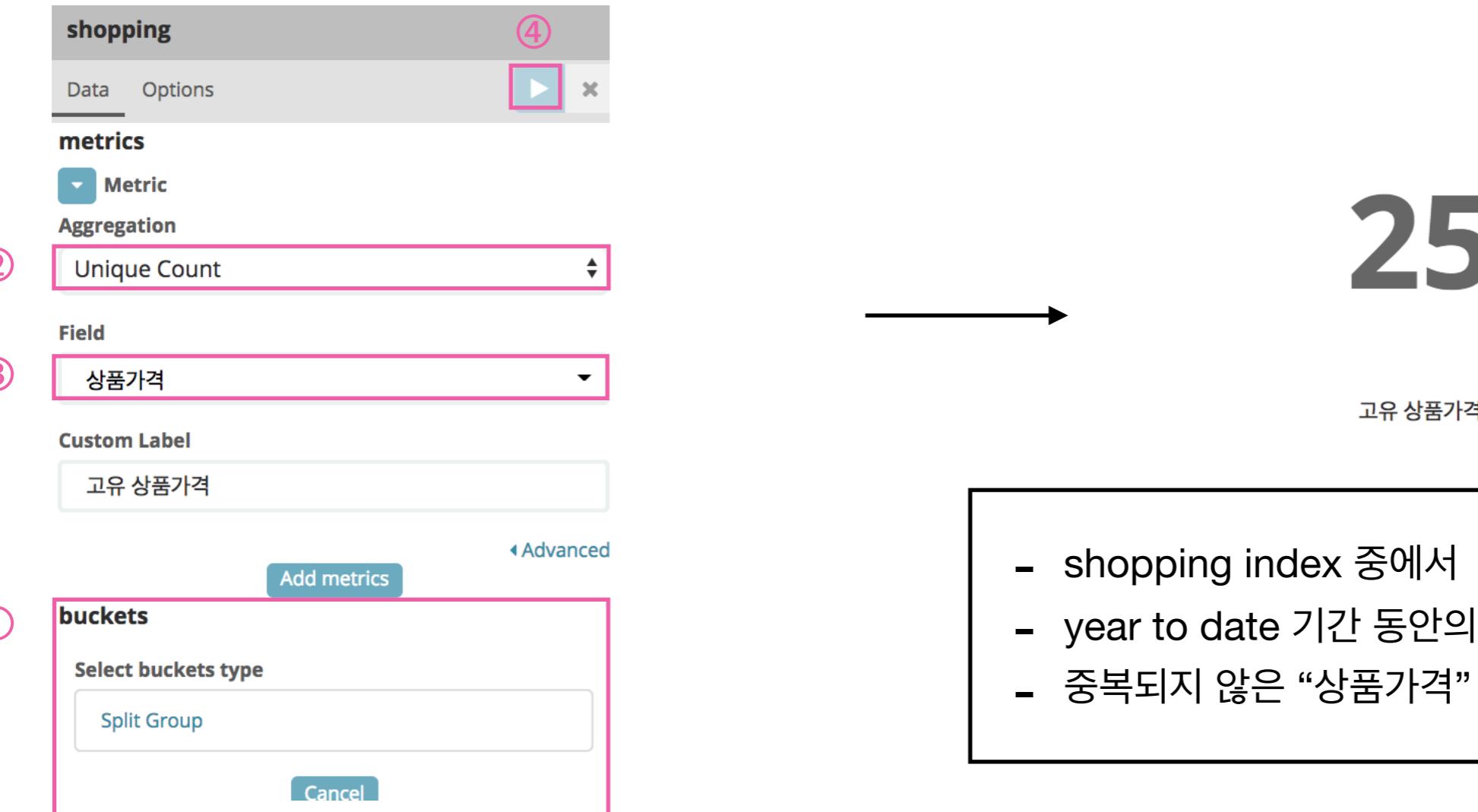
1. buckets은 고정하기 위해 ①은 원래 상태로 두자
2. ②를 눌러서 Min/Max aggregation 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 시각화

# Metric - Standard Deviation



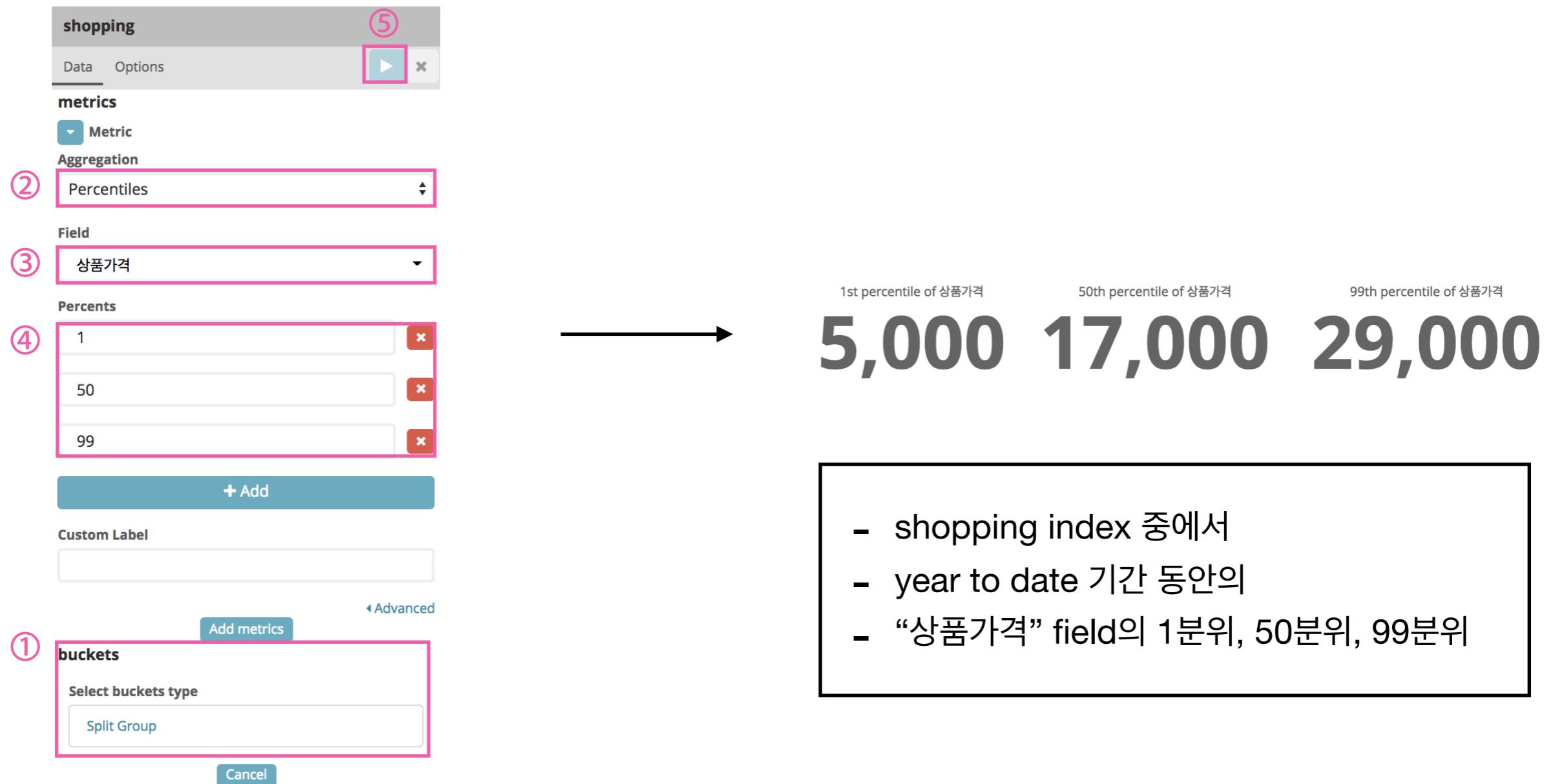
1. buckets은 고정하기 위해 ①은 원래 상태로 두자
2. ②를 눌러서 Standard Deviation aggregation 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 시각화

# Metric - Unique Count



1. buckets은 고정하기 위해 ①은 원래 상태로 두자
2. ②를 눌러서 Unique Count aggregation 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 시작화

# Metric - Percentiles



1. buckets은 고정하기 위해 ①은 원래 상태로 두자
2. ②를 눌러서 Percentiles aggregation 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 조회하려는 백분위수 입력
5. ⑤로 시작화

# Metric - Percentile Rank

The screenshot shows the Elasticsearch interface for defining a metric. The top bar has 'shopping' selected and a count of 5 documents. Below it, the 'metrics' section is open, showing a dropdown for 'Metric' set to 'Percentile Ranks'. The 'Field' dropdown is set to '상품가격'. The 'values' input field contains two values: '10000' and '15000'. A large blue button at the bottom left says '+ Add'. Below this, the 'buckets' section is expanded, showing a dropdown for 'Select buckets type' with 'Split Group' selected. A small 'Advanced' link is visible above the buckets section. At the bottom right is a 'Cancel' button.



Percentile rank 10,000 of "상품가격"      Percentile rank 15,000 of "상품가격"  
**24.033% 44.075%**

- shopping index 중에서
- year to date 기간 동안의
- “상품가격” field의 value가 10000, 15000인
- 데이터의 백분위수

1. buckets은 고정하기 위해 ①은 원래 상태로 두자
2. ②를 눌러서 Percentile Rank aggregation 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 백분위수를 조회하려는 value 입력
5. ⑤로 시각화

# Metric - Top Hit

The screenshot shows the Elasticsearch interface for defining a metric. The top bar has 'shopping' selected and a play button. The 'metrics' section is open, showing:

- ② Top Hit (highlighted in pink)
- Field: 판매자평점
- Aggregate With ④: Average
- Size ⑤: 10
- Sort On: 배송소요시간
- Order: Descending
- Custom Label: (empty)
- Add metrics: Advanced
- ① Buckets: Select buckets type (Split Group)

At the bottom are 'Cancel' and 'OK' buttons.

1.6

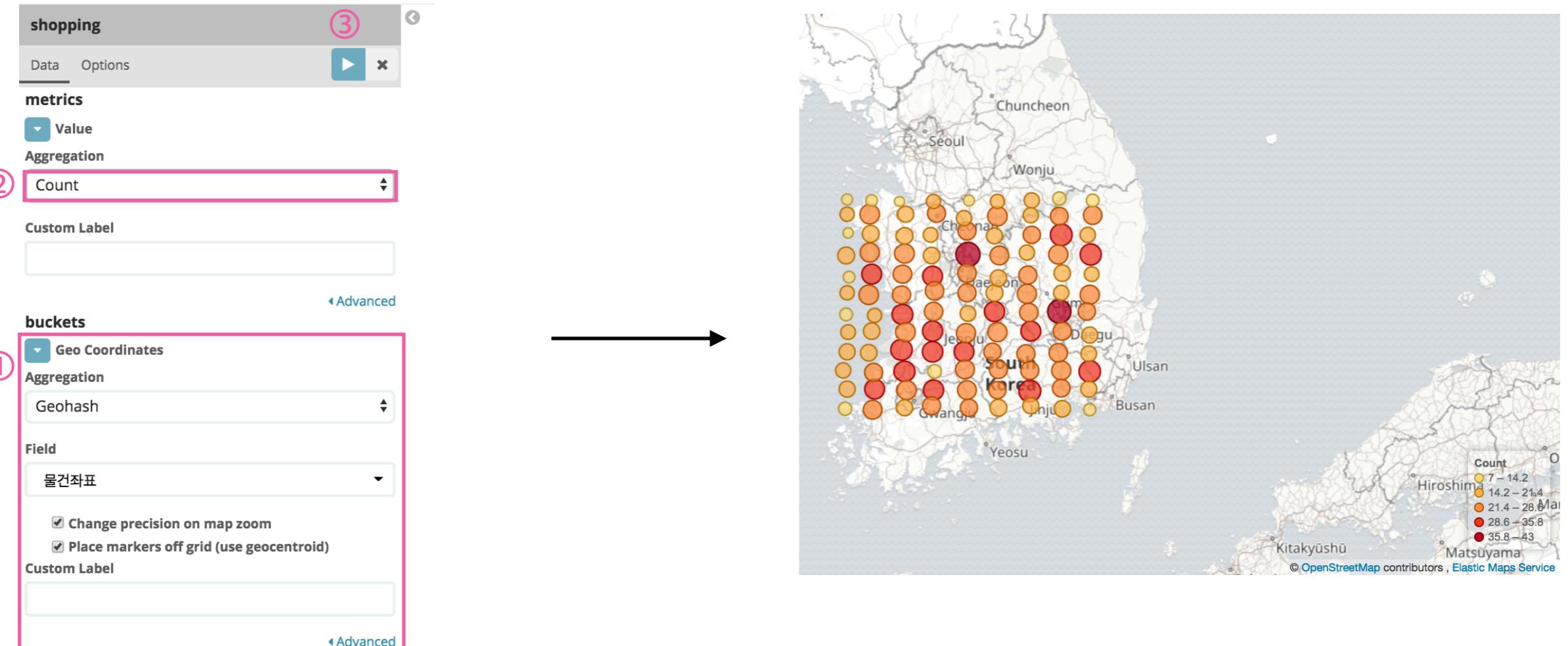
Last 10 판매자평점

- shopping index 중에서
- year to date 기간 동안의
- “배송소요시간” Field의 value가
- 큰
- 10개의 데이터의
- “판매자평점” Field 값의 평균

1. buckets은 고정하기 위해 ①은 원래 상태로 두자
2. ②를 눌러서 Top Hit aggregation 선택
3. ③을 눌러서, ⑤~⑦에서 선별된 데이터에 ④를 적용할 Field 선택
4. ④를 눌러서, ③에 어떤 aggregation 적용할지 선택
5. ⑤를 눌러서, ⑥~⑦정렬한 뒤 몇 번째 데이터까지 선별할 지 선택
6. ⑥을 눌러서 어떤 Field value를 기준으로 데이터를 정렬할지 선택
7. ⑦을 눌러서, ⑥을 오름차순/내림차순으로 정렬할지 선택
8. ⑦로 시각화

# Coordinate Map

- 개별 geo\_point 데이터를 시각화 할 때 사용
- buckets aggregation은 geo-hash만 지원하며, geo-point로 mapping 되어 있어야 한다



1. ①을 눌러서 buckets은 Geohash로 설정하자
2. ②를 눌러서 어떤 방식으로 지도 상에 값을 나타낼지 선택
3. ③을 눌러서 시각화

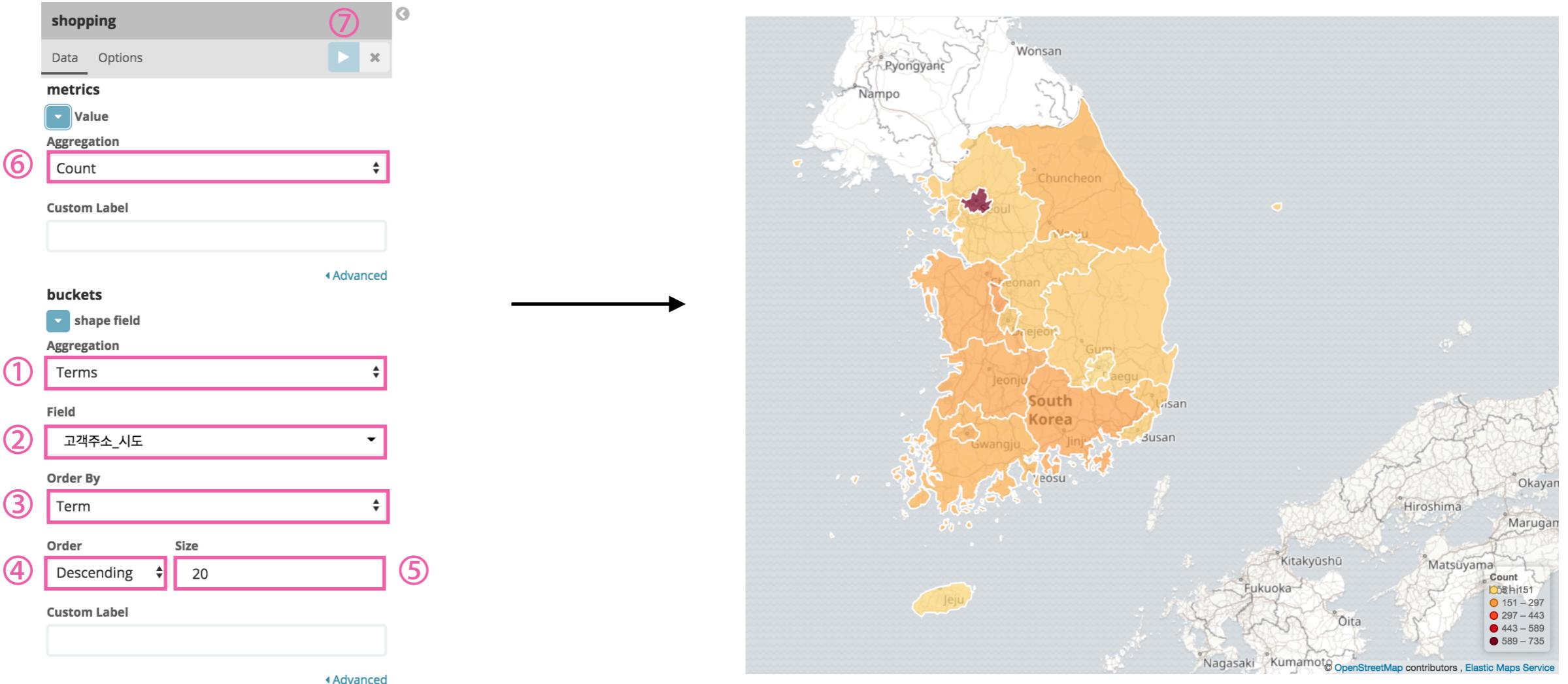
# Coordinate Map - 예제

metric aggregation을 다음과 같이 변경해보자 ↗

1. “배송소요시간”의 평균
2. “상품가격”의 합
3. “상품가격”의 최소/최대
4. 중복되지 않은 “상품분류”의 value 개수
5. “배송소요시간”이 가장 컸던 데이터 3개의 “배송소요시간” 평균

# Region Map

- 행정 구역 등의 단위로 데이터를 지도 상에 시각화
  - 단, 한국을 행정 구역 등으로 구분해서 시각화하려면 사전 작업이 필요하다 



- ①을 눌러서 buckets은 term aggregation으로 설정하자
  - ②를 눌러서, ①을 적용할 Field 선택 (단, Region Map에서는 지도와 호환되는 Field 설정)
  - ③~⑤을 눌러서 bucket을 어떤 기준으로 선택할지 선택
  - ⑥을 눌러서 ①~⑤에서 선택한 bucket에 어떤 값을 채울지 선택
  - ⑦을 눌러서 시각화

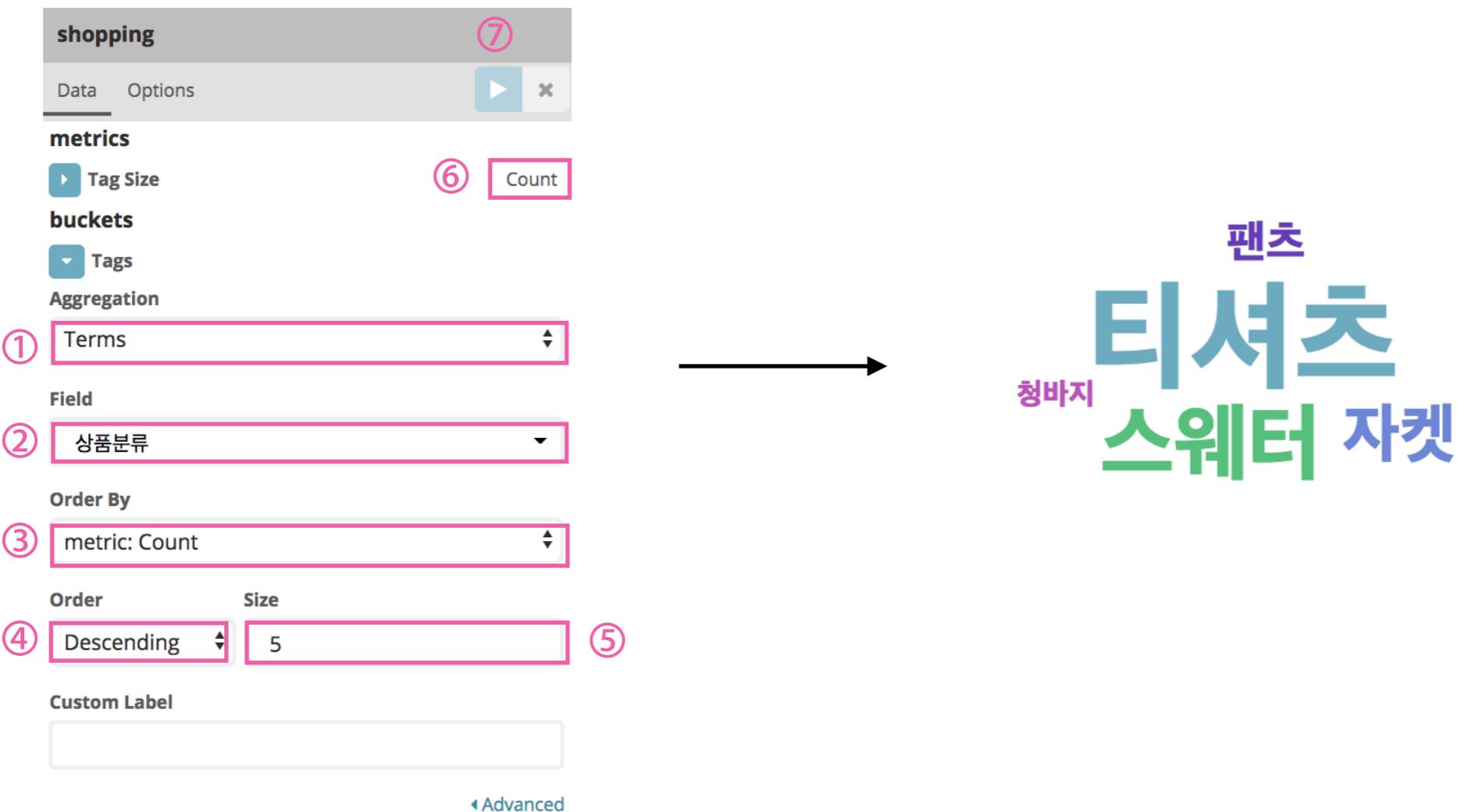
# Region Map - 예제

**metric aggregation**을 다음과 같이 변경해보자 ↗

1. “배송소요시간”의 평균
2. “상품가격”의 합
3. “상품가격”의 최소/최대
4. 중복되지 않은 “상품분류”의 value 개수
5. “배송소요시간”이 가장 컸던 데이터 3개의 “배송소요시간” 평균

# 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



1. ①을 눌러서 buckets은 term aggregation으로 설정하자
2. ②를 눌러서, ①을 적용할 Field 선택
3. ③~⑤를 눌러서 bucket을 어떤 기준으로 선택할지 선택
4. ⑥을 눌러서 ①~⑤에서 선택한 bucket에 어떤 값을 채울지 선택
5. ⑦을 눌러서 시각화

# Tag Cloud - 예제

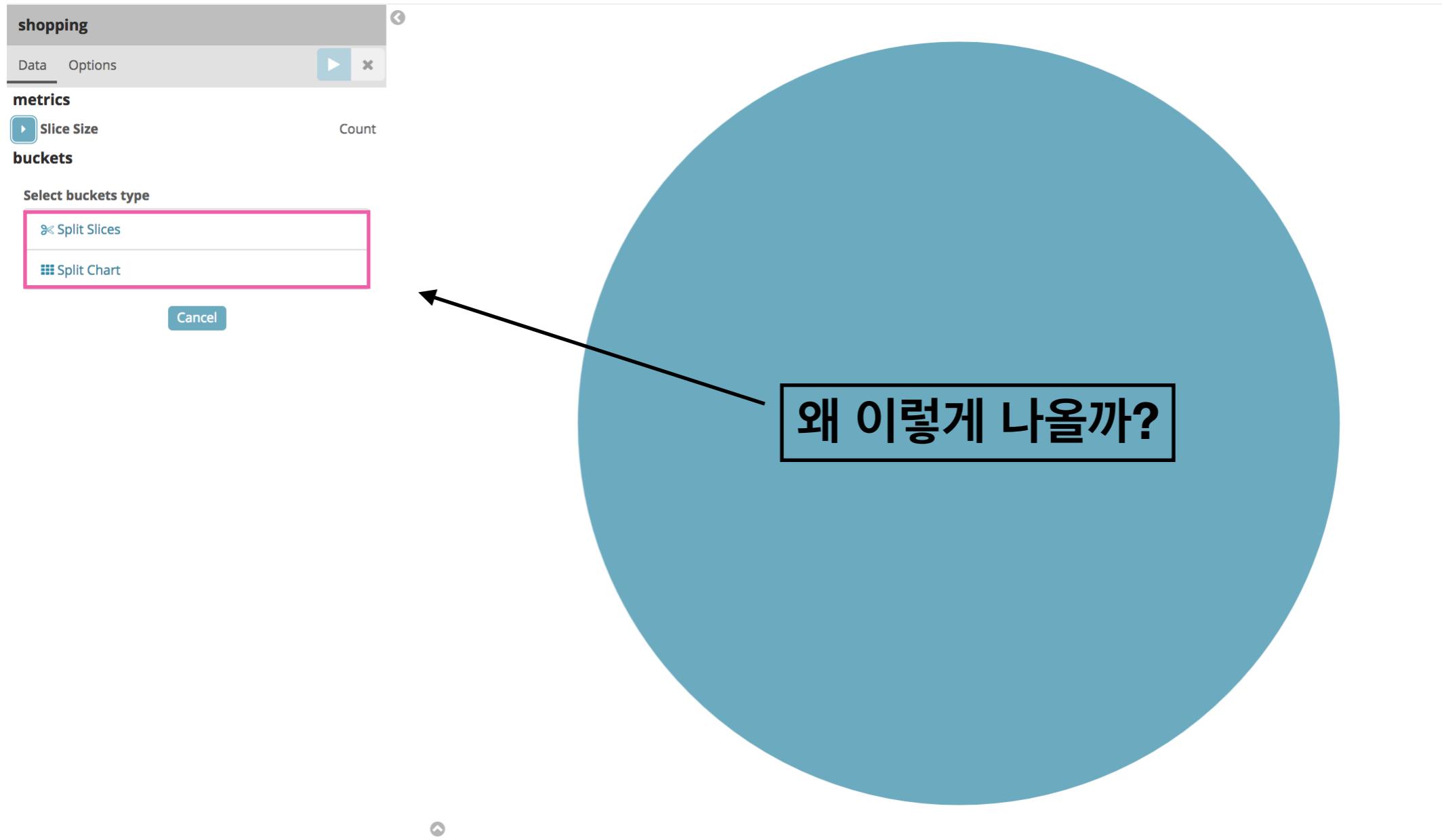
metric aggregation을 다음과 같이 변경해보자 ↗

1. “배송소요시간”의 평균
2. “상품가격”의 합
3. “상품가격”의 중위값
4. “상품가격”의 최소/최대
5. 중복되지 않은 “상품분류”의 value 개수
6. “배송소요시간”이 가장 컸던 데이터 3개의 “배송소요시간” 평균

**이제는 Buckets 쪽에 집중해보자**

# Pie Chart

- 특정 Field의 value를 어떤 기준으로 묶어서 그 분포를 시각화할 때 사용
- metric aggregation는 value count로 고정하고 buckets를 나누는 과정을 학습하자



# Pie Chart - Split Slices (Date Histogram)

shopping

Data Options

metrics

Slice Size

buckets

Split Slices

Aggregation

② Date Histogram

Field

③ 주문시간

Interval

④ Weekly

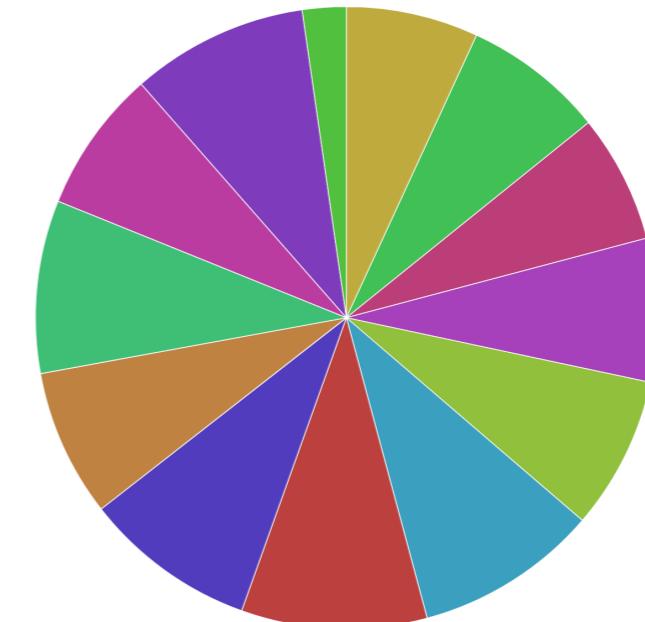
Custom Label

① Count

⑤

Add sub-buckets

Advanced

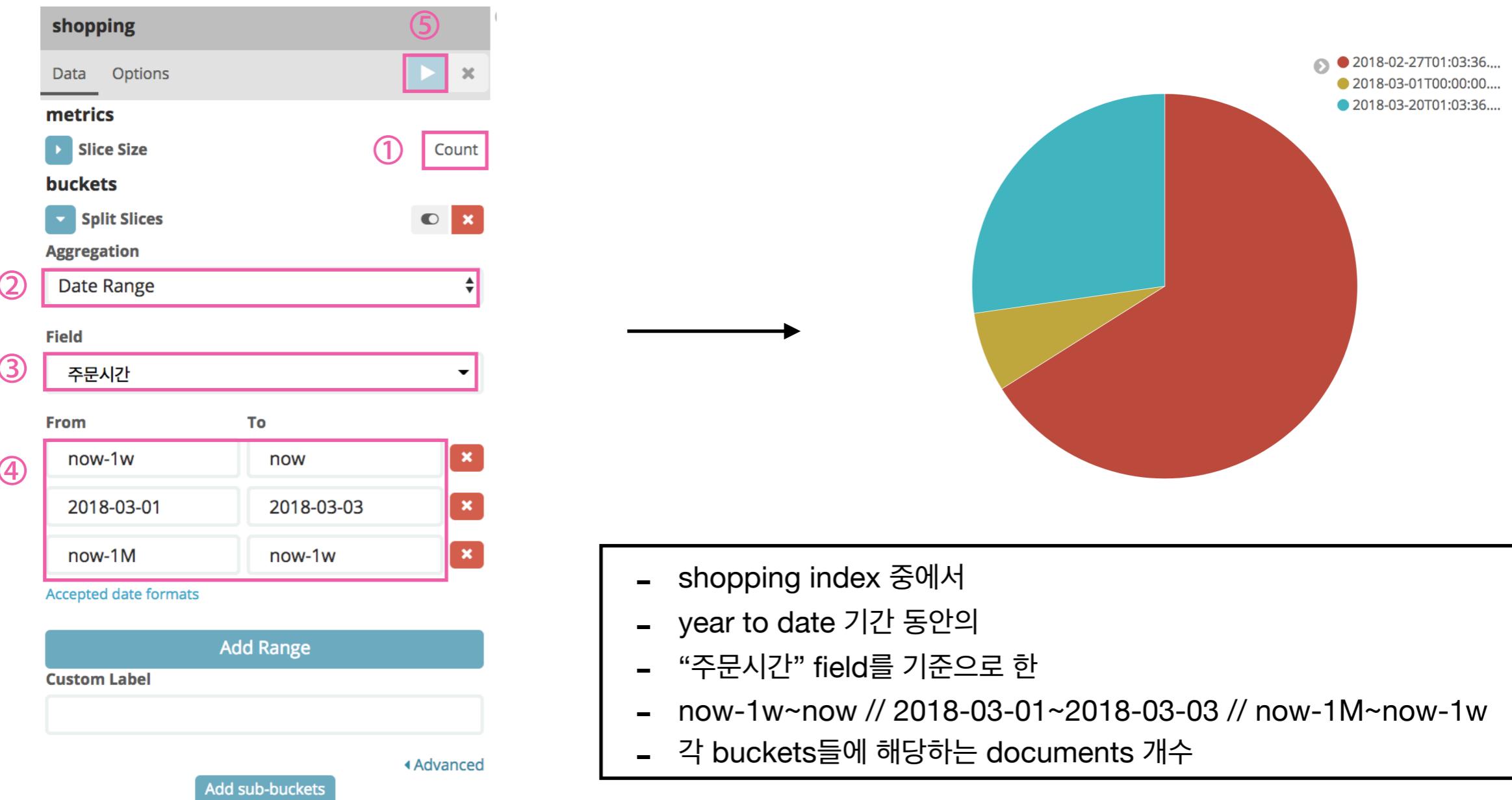


- 01월01일
- 01월08일
- 01월15일
- 01월22일
- 01월29일
- 02월05일
- 02월12일
- 02월19일
- 02월26일
- 03월05일
- 03월12일
- 03월19일
- 03월26일

- shopping index 중에서
- year to date 기간 동안의
- “주문시간” field를 기준으로 한
- 주별 (weekly) documents 개수

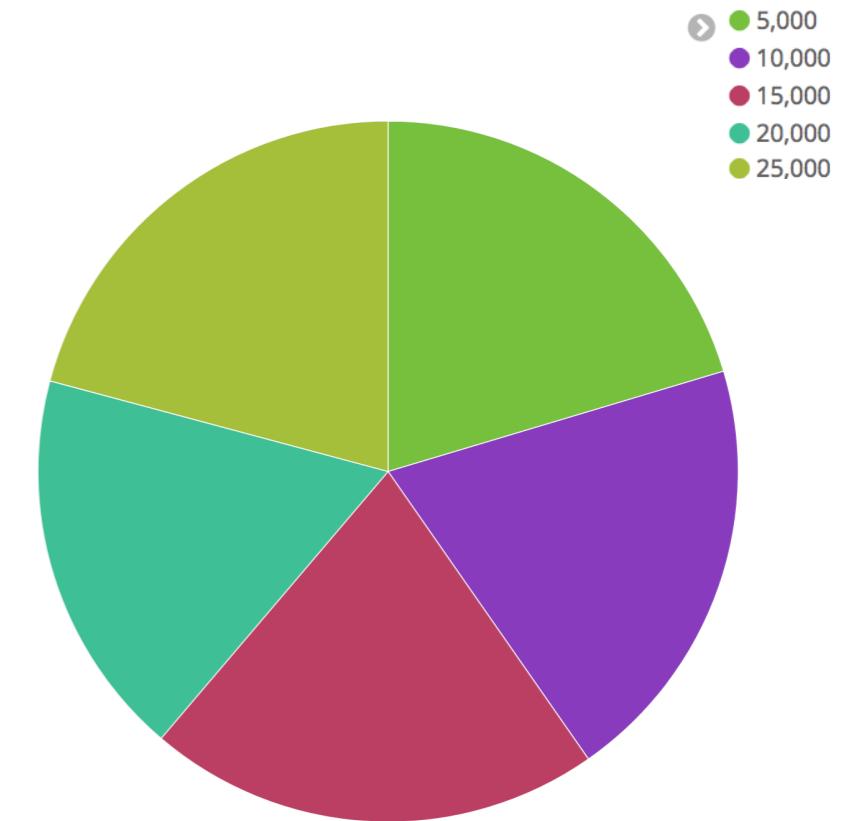
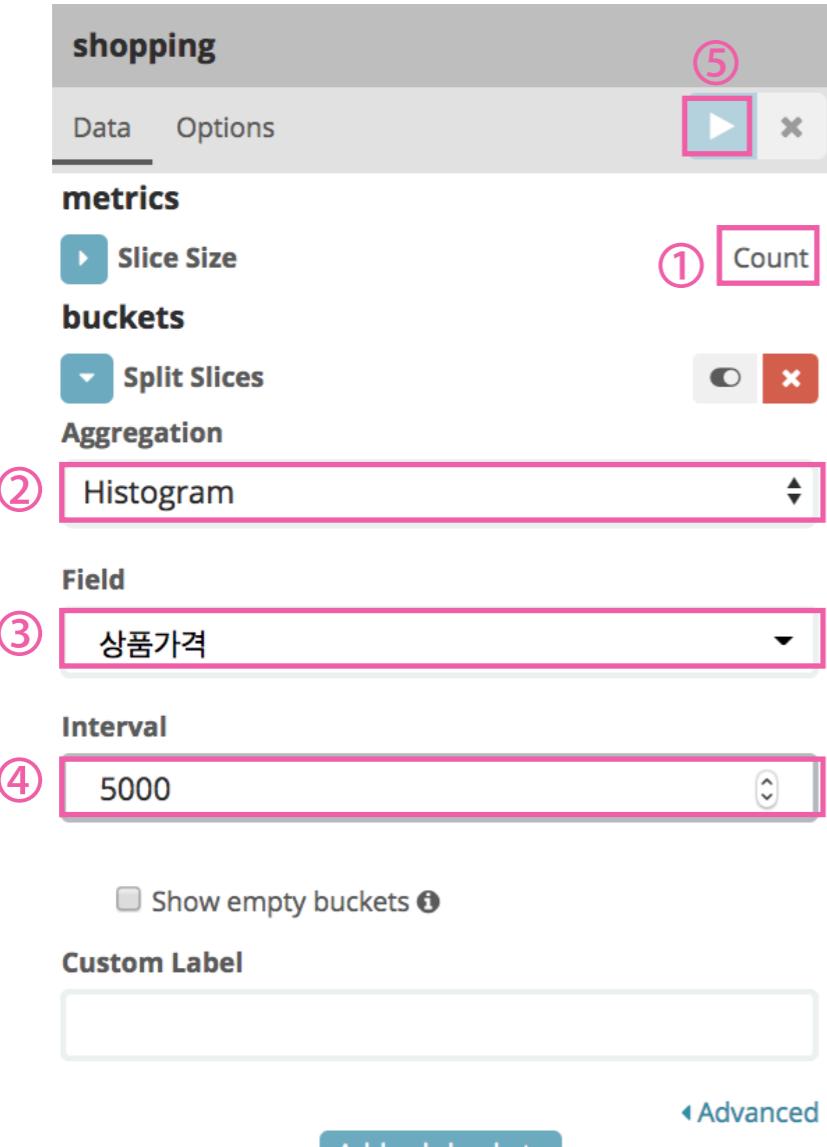
1. metric은 value count로 고정하기 위해 ①은 원래 상태로 두자
2. 날짜를 기준으로 bucket을 생성하기 위해 ②를 눌러서 date histogram 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 ③을 나눌 간격 설정 (histogram이므로 모두 일정)
5. ⑤를 눌러서 시각화

# Pie Chart - Split Slices (Date Range)



1. metric은 value count로 고정하기 위해 ①은 원래 상태로 두자
2. 날짜를 기준으로 bucket을 생성하기 위해 ②를 눌러서 date range 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 ③을 나눌 개별 bucket 별 간격 설정
5. ⑤를 눌러서 시각화

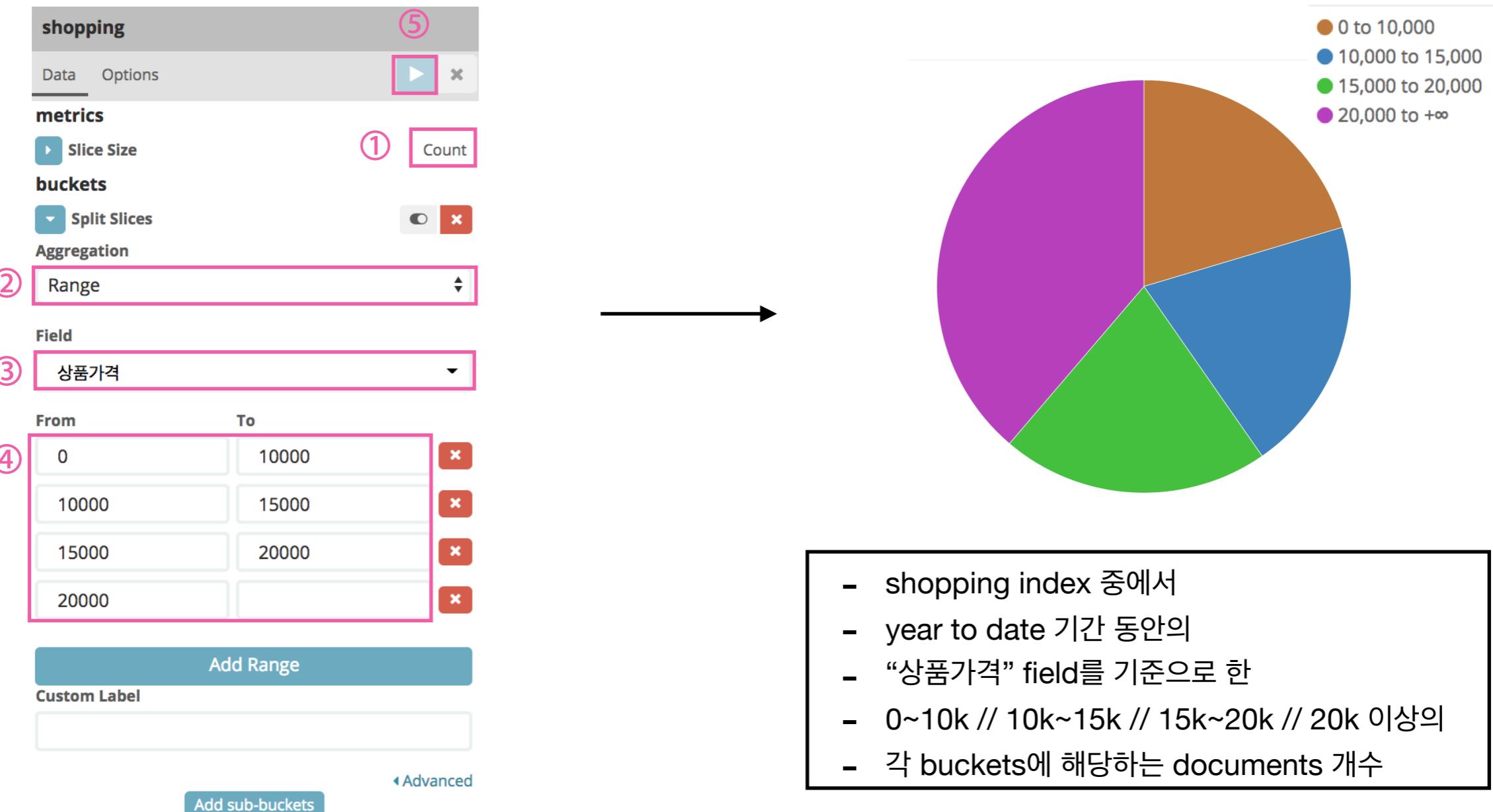
# Pie Chart - Split Slices (Histogram)



- shopping index 중에서
- year to date 기간 동안의
- “상품가격” field를
- 5000 간격으로 쪼개어 만든 각 bucket
- 에 해당하는 documents 개수

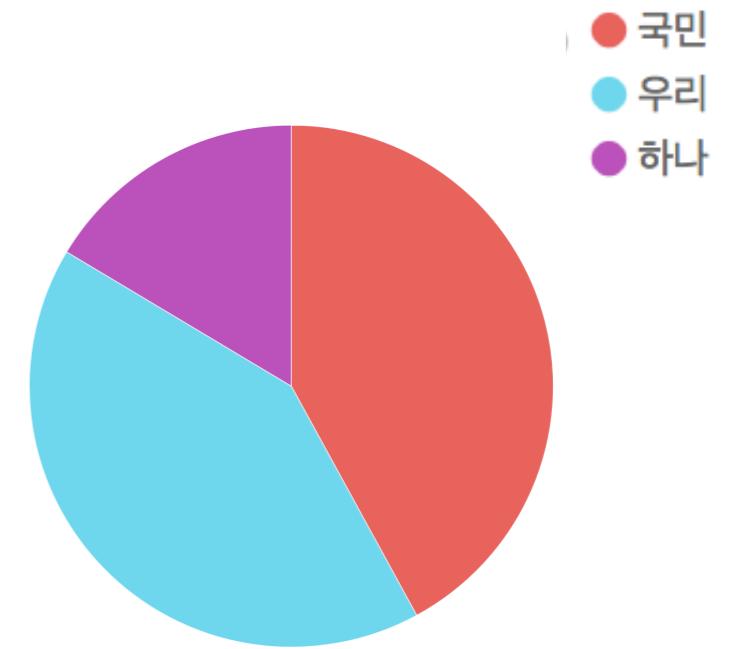
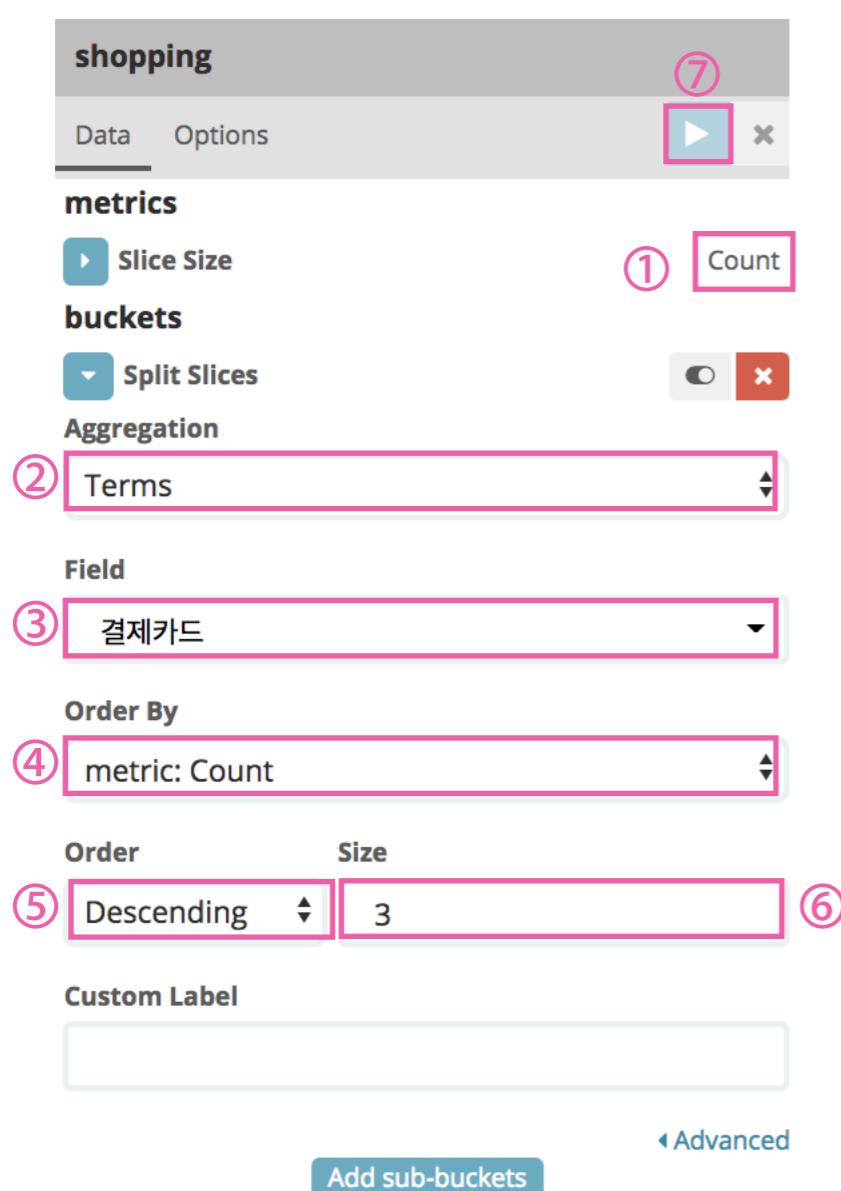
1. metric은 value count로 고정하기 위해 ①은 원래 상태로 두자
2. Number Field의 값을 기준으로 bucket을 생성하기 위해 ②를 눌러서 Histogram 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 ③을 나눌 간격 설정 (histogram이므로 모두 일정)
5. ⑤를 눌러서 시각화

# Pie Chart - Split Slices (Range)



1. metric은 value count로 고정하기 위해 ①은 원래 상태로 두자
2. Number Field의 값을 기준으로 bucket을 생성하기 위해 ②를 눌러서 Range 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 ③을 나눌 개별 bucket 별 간격 설정
5. ⑤를 눌러서 시각화

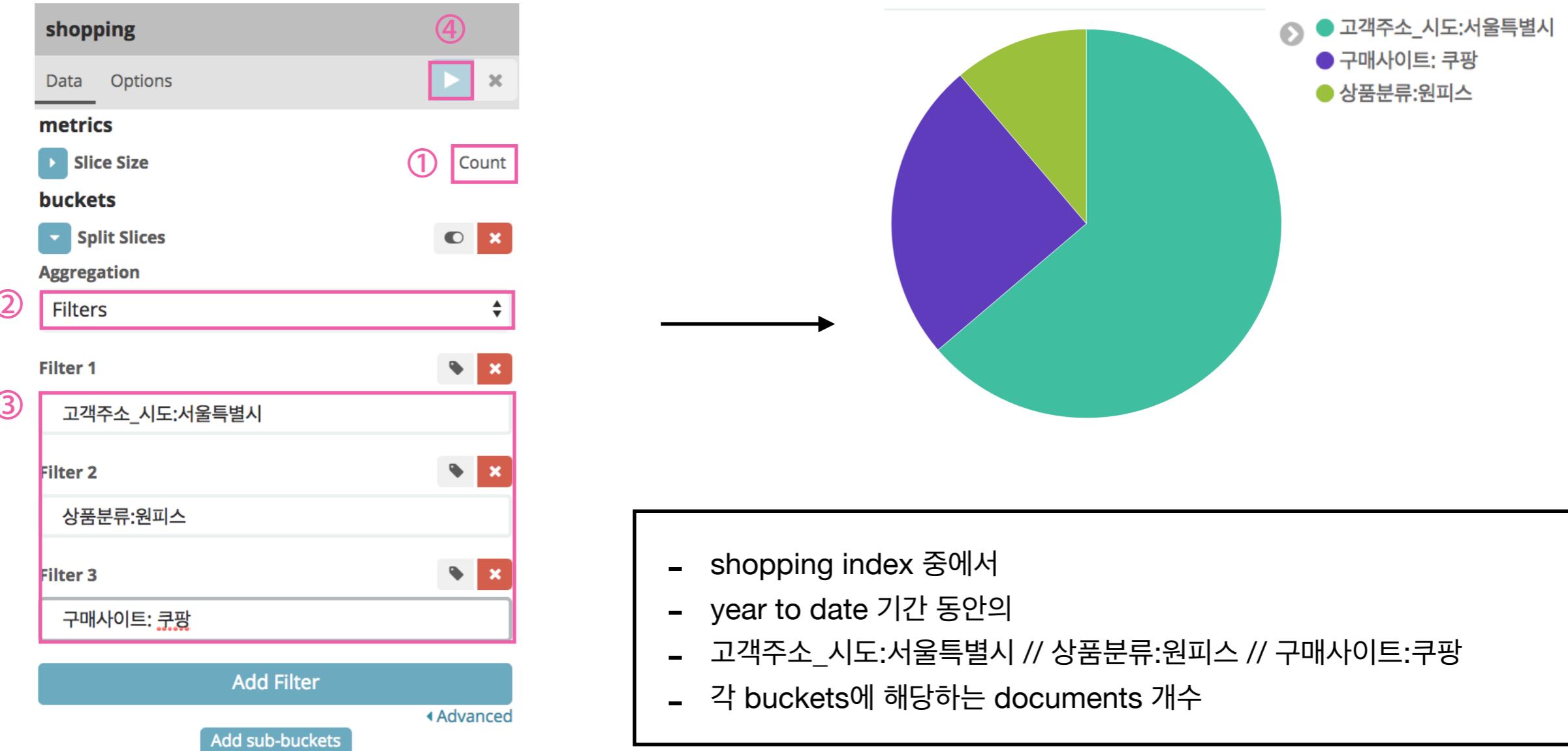
# Pie Chart - Split Slices (Terms)



- shopping index 중에서
- year to date 기간 동안의
- “결제카드” field를 기준으로 한
- “결제카드” 별 documents가 많은
- 상위
- 3개 “결제카드” 별
- documents 개수

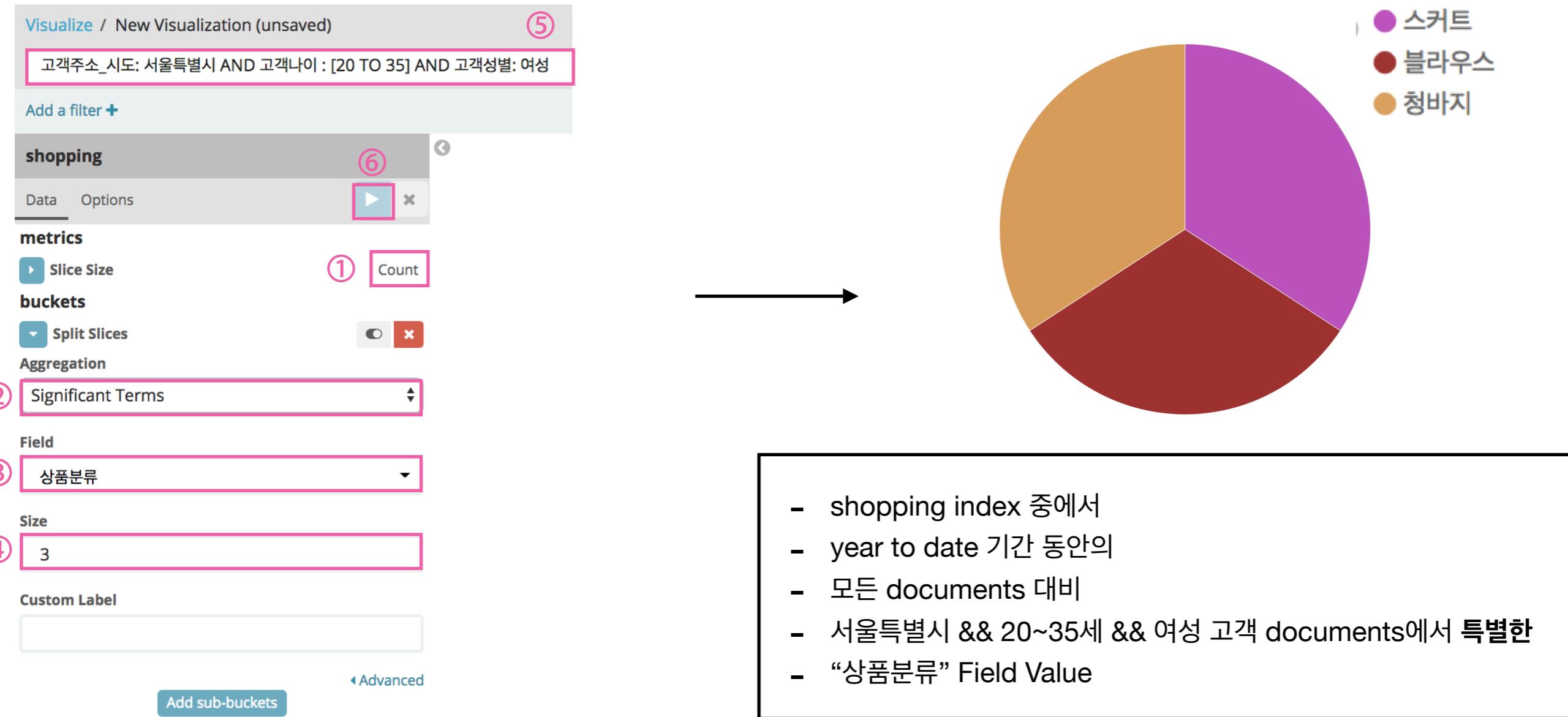
1. metric은 value count로 고정하기 위해 ①은 원래 상태로 두자
2. 특정 Field의 Value를 기준으로 bucket을 생성하기 위해 ②를 눌러서 Terms 선택
3. ③을 눌러서 ②를 적용할 대상 Field 선택
4. ④를 눌러서 ②~③을으로 생성한 buckets을 어떤 값을 기준으로 비교할지 선택
5. ⑤를 눌러서 ④를 정렬할 기준 선택 (오름차순/내림차순)
6. ⑤로 정렬한 bucket 별로 정렬했다면, 몇 개의 bucket을 선택할지 ⑥에 입력

# Pie Chart - Split Slices (Filter)



1. metric은 value count로 고정하기 위해 ①은 원래 상태로 두자
2. 직접 bucket을 지정할 것이기에 ②를 눌러서 Filters 선택
3. ③에 지정할 bucket 조건 입력 : 하나의 document가 여러 bucket에 포함될 수 있음
4. ④를 눌러서 시작화

# Pie Chart - Split Slices (Filter)



1. metric은 value count로 고정하기 위해 ①은 원래 상태로 두자
2. significant terms을 이용해서 bucket을 지정할 것이기에 ②를 눌러서 significant terms 선택
3. ②를 적용할 Field를 ③에서 선택
4. ②~③ 조건을 만족하는 buckets을 몇 개를 생성할지 ④에 입력
5. significant terms는 background 대비 foreground에서 특별한 값을 찾기에 foreground 조건을 ⑤에 입력 왕
6. ⑥을 눌러 시작화

이번에는 Split Charts 후에 Split Slices를 적용하자



그전에 이게 왜 필요할까?

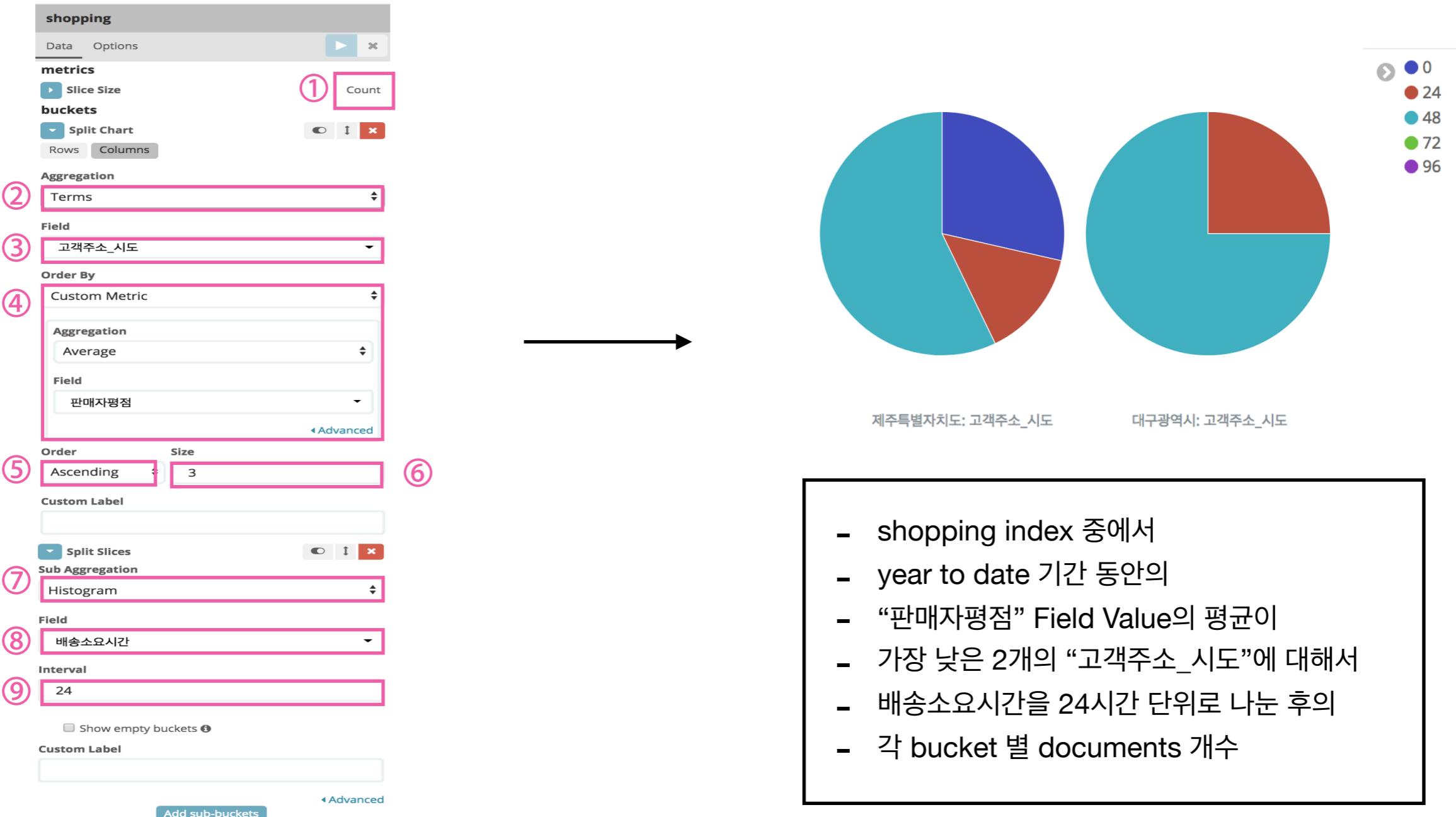


지금까지는 **하나의** Field를 기준으로 시각화 했다



하지만 현실 세계의 문제는 그리 간단하지 않다면?

# Pie Chart - Split Chart (Terms) + Slices (Histogram)



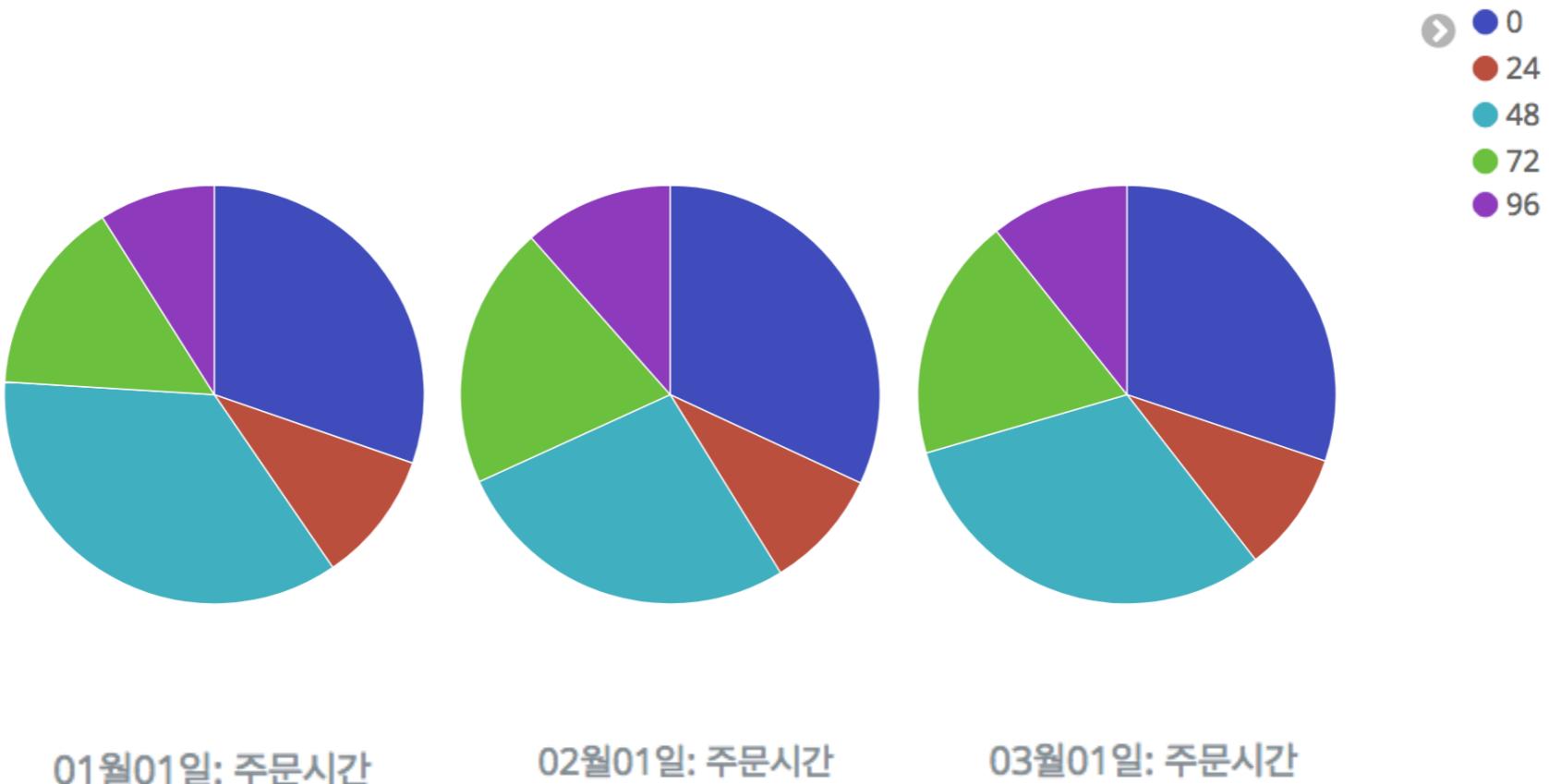
1. metric은 value count로 고정하기 위해 ①은 원래 상태로 두자
2. terms을 이용해서 chart를 나눌 것이기에 ②에서 Split Chart - Terms 선택
3. ②를 적용할 Field를 ③에서 선택
4. ②~③ 조건을 만족하는 buckets을 선정하기 위한 기준을 ④에서 선택
5. ④에서 생성한 buckets을 오름차순/내림차순으로 정렬할지 ⑤에서 선택
6. ⑤에서 정렬이 끝났으면, 몇 개를 선택할지 ⑥에 입력
7. ⑦~⑨에서 Split Slices 조건 선택

# Pie Chart - 예제 1

metrics : 아래 bucket 별 count

Split Chart : “주문시간” Field를 월별로

Split Slices : “배송소요시간” Field를 24시간 단위로

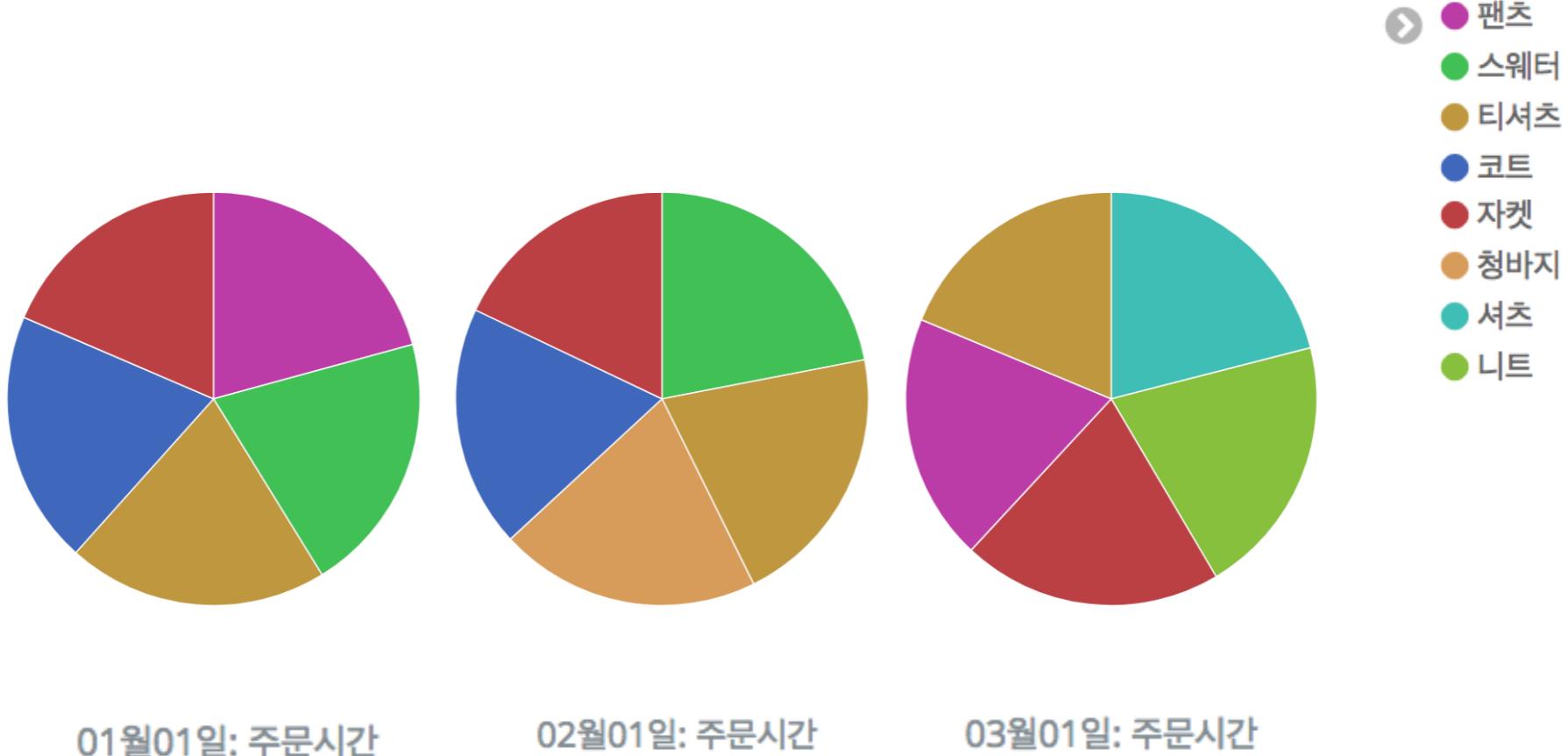


# Pie Chart - 예제 2

metrics : 아래 bucket 별 count

Split Chart : “주문시간” Field를 월별로

Split Slices : “상품분류” Field를 기준으로 가장 빈번하게 나왔던 5개 선정

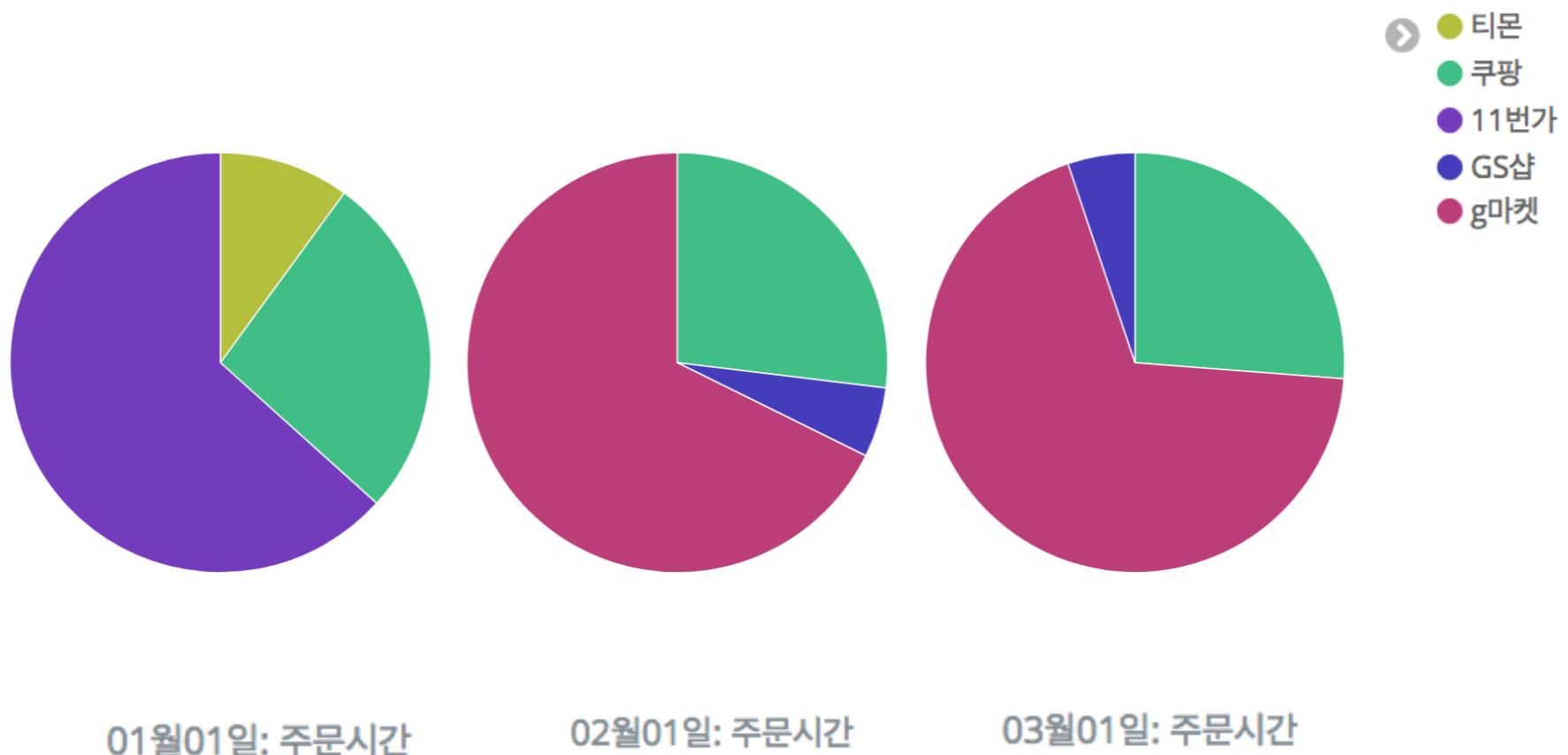


# Pie Chart - 예제 3

metrics : 아래 bucket 별 count

Split Chart : “주문시간” Field를 월별로

Split Slices : “구매사이트” Field를 “상품가격” Field의 평균이 높았던 3개로



# 마치기 전에

- Elastic Stack이 무엇을 의미하며 어떤 용도로 쓰이는지 이해한다
- Elasticsearch의 기본 용어를 이해한다
- Elastic Stack을 사용할 때 큰 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**