

Elastic Stack 을 활용한 Data Dashboard 만들기

Week 2 - Kibana를 조금 잘 사용해보자



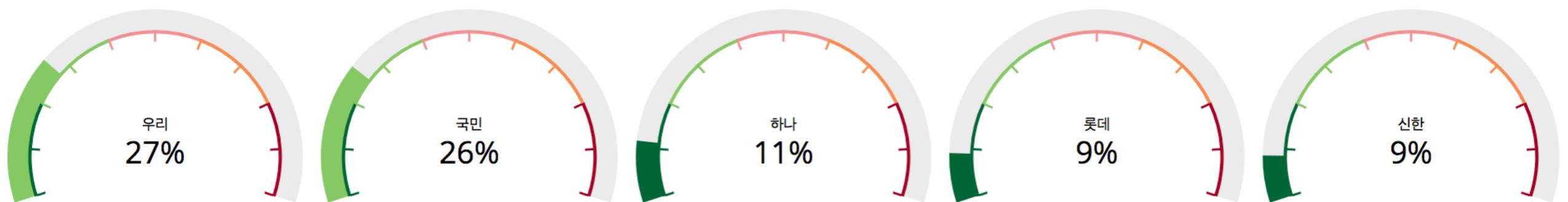
Fast Campus

목차

- 실전 Visualize
 - Gauge 4
 - Region Map 8
 - Heat Map 12
 - Data Table 16
 - Timelion 20
- 실전 Dashboard 23
- 몇 가지 고민들
 - data format 변경 26
 - csv 출력 30
 - JSON Input 36
- Filtering by Field 42
- Lucene Query 48
- Scripted Field 53

문제 1

Visualize - Gauge



exercise_gauge

Index : shopping
Time Range : 2017년 9월 1일 ~ 10월 31일

Visualize - Gauge

문제

상품가격의 합이 가장 높은 5개 카드의 상품가격의 합의 목표를 Percentage Mode로 설정하세요.

사용한 Visualization Type

Gauge

사용한 Aggregation

Metrics : Sum

Buckets : Terms, Sum

목표 구간

0 ~ 5,000,000

5,000,000 ~ 10,000,000

10,000,000 ~ 15,000,000

15,000,000 ~ 20,000,000

20,000,000 ~ 25,000,000

사용 필드

상품가격

결제카드

Visualize - Gauge

shopping

Data Options ▶ X

Metric

Aggregation: Sum
Field:

Custom Label: 매출

Add metrics ◀ Advanced

buckets

Split Group Toggle X

Aggregation:
Field: 결제카드
Order By: Custom Metric
Aggregation:
Field: 상품가격
◀ Advanced

Order: Size: 5

Custom Label: 매출

◀ Advanced

shopping

Data Options ▶ X

Gauge Type: Arc Arc ▼

Percentage Mode:

Vertical Split:

Show Legend:

Show Labels:

Sub Text:

Auto Extend Range: Ranges

| From | To | X |
|------|----------|---|
| 0 | | X |
| | | X |
| | 15000000 | X |
| | | X |
| | | X |

Add Range

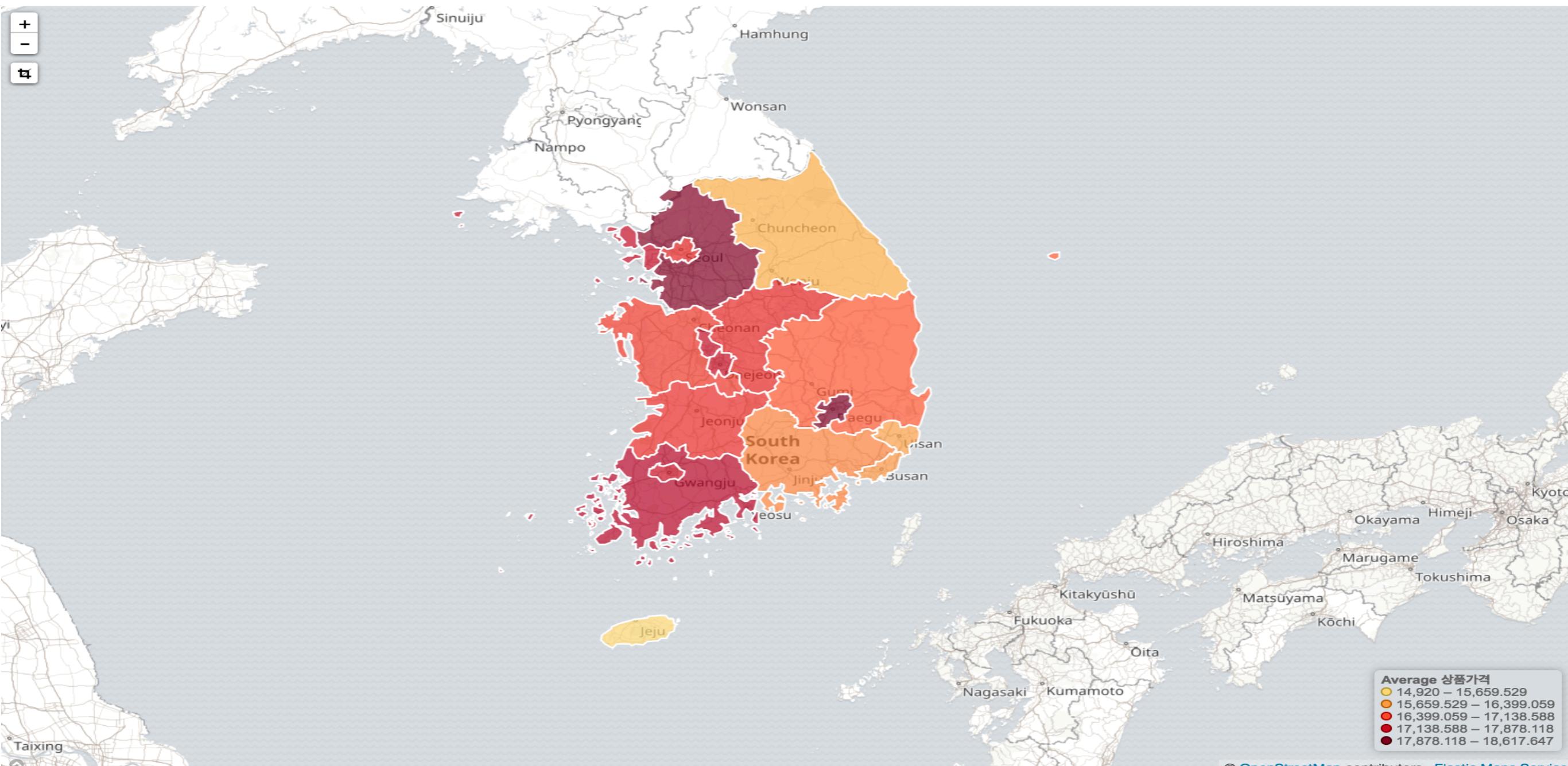
Note: colors can be changed in the legend

Color Options

Style

문제 2

Visualize - Region Map



exercise_region_map

Index : shopping

Time Range : 2017년 9월 1일 ~ 10월 31일

Visualize - Region Map

문제

전국 행정구역 17개별 상품가격의 평균을 나타내시오.

사용한 Visualization Type

Region Map

사용한 Aggregation

Metrics : Average

Buckets : Terms

사용하는 필드

상품가격

고객주소_시도

Visualize - Region Map

shopping

Data Options ▶ ×

metrics

▼ Value

Aggregation

▼

Field

▼

Custom Label

▼

◀ Advanced

buckets

▼ shape field

Aggregation

Terms ▼

Field

▼

Order By

▼

Order Size

▼ ▼

Custom Label

▼

◀ Advanced

shopping

Data Options ▶ ×

Layer Settings

Vector map ▼

Join field ▼

Style Settings

Color Schema Yellow to Red ▼

Basic Settings

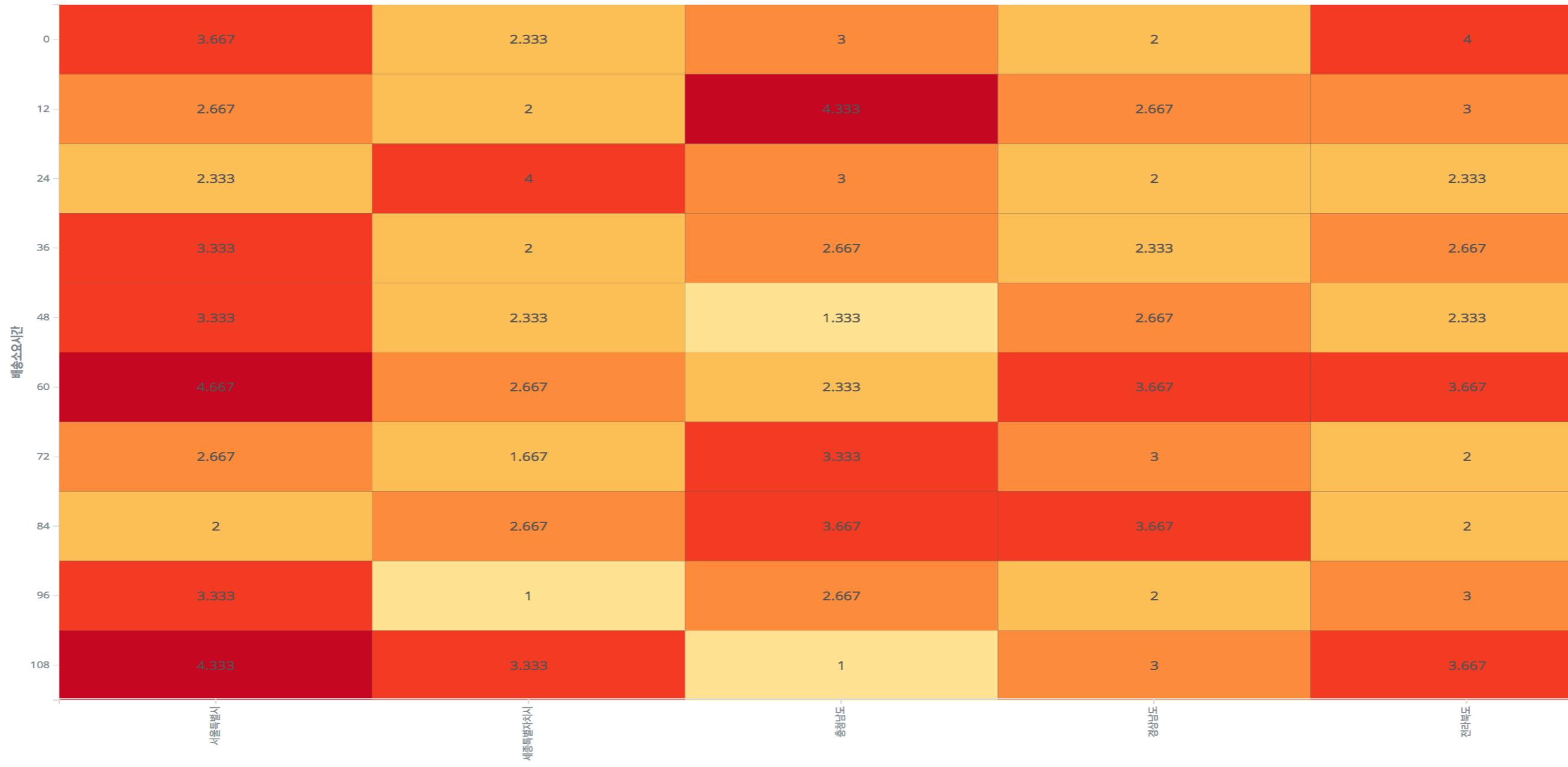
Legend Position

bottom right ▼

Show Tooltip

문제 3

Visualize - Heat Map



exercise_heatmap

Index : shopping

Time Range : 2017년 9월 1일 ~ 10월 31일

Visualize - Heat Map

문제

상품가격의 합이 가장 높은 5개 지역과 배송소요시간 (12시간 단위)에 따라 가장 어린 고객 3명의 판매자평점의 평균을 표시하세요

사용한 Visualization Type

Heat Map

사용한 Aggregation

Metrics : Top Hit, Average

Buckets : X-axis (Terms, Sum), Y-axis (Histogram)

사용하는 필드

판매자평점

고객주소_시도

배송소요시간

상품가격

고객나이

Visualize - Heat Map

shopping

Data Options ▶

metrics

Value

Aggregation

Top Hit

Field

Aggregate With i Size i

Average 5

Sort On

고객나이

Order

Custom Label

▶ Advanced

shopping

Data Options ▶

◀ Advanced

buckets

X-Axis

Aggregation

Field: 고객주소_시도

Order By: Custom Metric

Aggregation

Field: 상품가격

◀ Advanced

Order: Size

Size: 5

Custom Label: 주소

◀ Advanced

Y-Axis

Sub Aggregation

Histogram

Field

Interval: 12

shopping

Data Options ▶

Basic Settings

Show Tooltips

Highlight

Legend Position: right

Heatmap Settings

Color Schema: Yellow to Red

Reverse Color Schema:

Color Scale: linear

Scale to Data Bounds:

Percentage Mode:

Number of colors: 6

▶ Custom Ranges

▶ Show Labels

문제 4

Visualize - Data Table

| 날짜 | 인기 Top 3 | 매출 | 매출 증감 | 매출 누적 |
|--------|---------------|---------|----------|-----------|
| 09월01일 | 가디건, 원피스, 원피스 | 402,000 | - | 402,000 |
| 09월02일 | 티셔츠, 가디건, 셔츠 | 382,000 | -20,000 | 784,000 |
| 09월03일 | 청바지, 팬츠, 니트 | 507,000 | 125,000 | 1,291,000 |
| 09월04일 | 셔츠, 가디건, 원피스 | 423,000 | -84,000 | 1,714,000 |
| 09월05일 | 남방, 니트, 청바지 | 393,000 | -30,000 | 2,107,000 |
| 09월06일 | 점퍼, 코트, 남방 | 383,000 | -10,000 | 2,490,000 |
| 09월07일 | 셔츠, 니트, 셔츠 | 442,000 | 59,000 | 2,932,000 |
| 09월08일 | 셔츠, 남방, 니트 | 376,000 | -66,000 | 3,308,000 |
| 09월09일 | 가디건, 남방, 셔츠 | 226,000 | -150,000 | 3,534,000 |
| 09월10일 | 가디건, 코트, 니트 | 366,000 | 140,000 | 3,900,000 |
| 09월11일 | 스웨터, 셔츠, 팬츠 | 319,000 | -47,000 | 4,219,000 |
| 09월12일 | 자켓, 셔츠, 티셔츠 | 345,000 | 26,000 | 4,564,000 |
| 09월13일 | 자켓, 가디건, 셔츠 | 278,000 | -67,000 | 4,842,000 |
| 09월14일 | 팬츠, 점퍼, 가디건 | 278,000 | 0 | 5,120,000 |
| 09월15일 | 티셔츠, 티셔츠, 코트 | 306,000 | 28,000 | 5,426,000 |
| 09월16일 | 가디건, 니트, 팬츠 | 323,000 | 17,000 | 5,749,000 |
| 09월17일 | 블라우스, 청바지, 자켓 | 245,000 | -78,000 | 5,994,000 |
| 09월18일 | 스웨터, 자켓, 블라우스 | 366,000 | 121,000 | 6,360,000 |
| 09월19일 | 팬츠, 남방, 티셔츠 | 417,000 | 51,000 | 6,777,000 |
| 09월20일 | 티셔츠, 니트, 가디건 | 282,000 | -135,000 | 7,059,000 |

Export: [Raw](#) [Formatted](#)

1 2 3 4 »

exercise_data_table

Index : shopping
Time Range : 2017년 9월 1일 ~ 10월 31일

Visualize - Data Table

문제

일별(주문시간 기준)로 다음 각 정보를 표시하세요.

- 상품가격이 가장 비쌌던 상품분류 3개
- 상품가격의 합
- 상품가격의 합의 일별 증감
- 상품가격의 일별 누적합

사용한 Visualization Type

Data Table

사용한 Aggregation

Metrics : Top Hit, Concatenate, Sum, Derivative, Cumulative Sum

Buckets : Date Histogram

사용하는 필드

상품가격

상품분류

주문시간

Visualize - Data Table

shopping

Data Options ▶ ×

▼ Metric (On) (Up) (Down)

Aggregation

Top Hit

Field

Concatenate

Aggregate With (i) Size (i)

Concatenate

Sort On

Order

Custom Label

인기 Top 3

◀ Advanced

▼ Metric (On) (Up) (Down)

Aggregation

Sum

Field

매출

Metric

Derivative

Aggregation

Date Histogram

Field

매출 증감

Advanced

▼ Metric (On) (Up) (Down)

Aggregation

Cumulative Sum

Metric

날짜

Custom Label

매출 누적

buckets

▼ Split Rows (On) (Off)

Aggregation

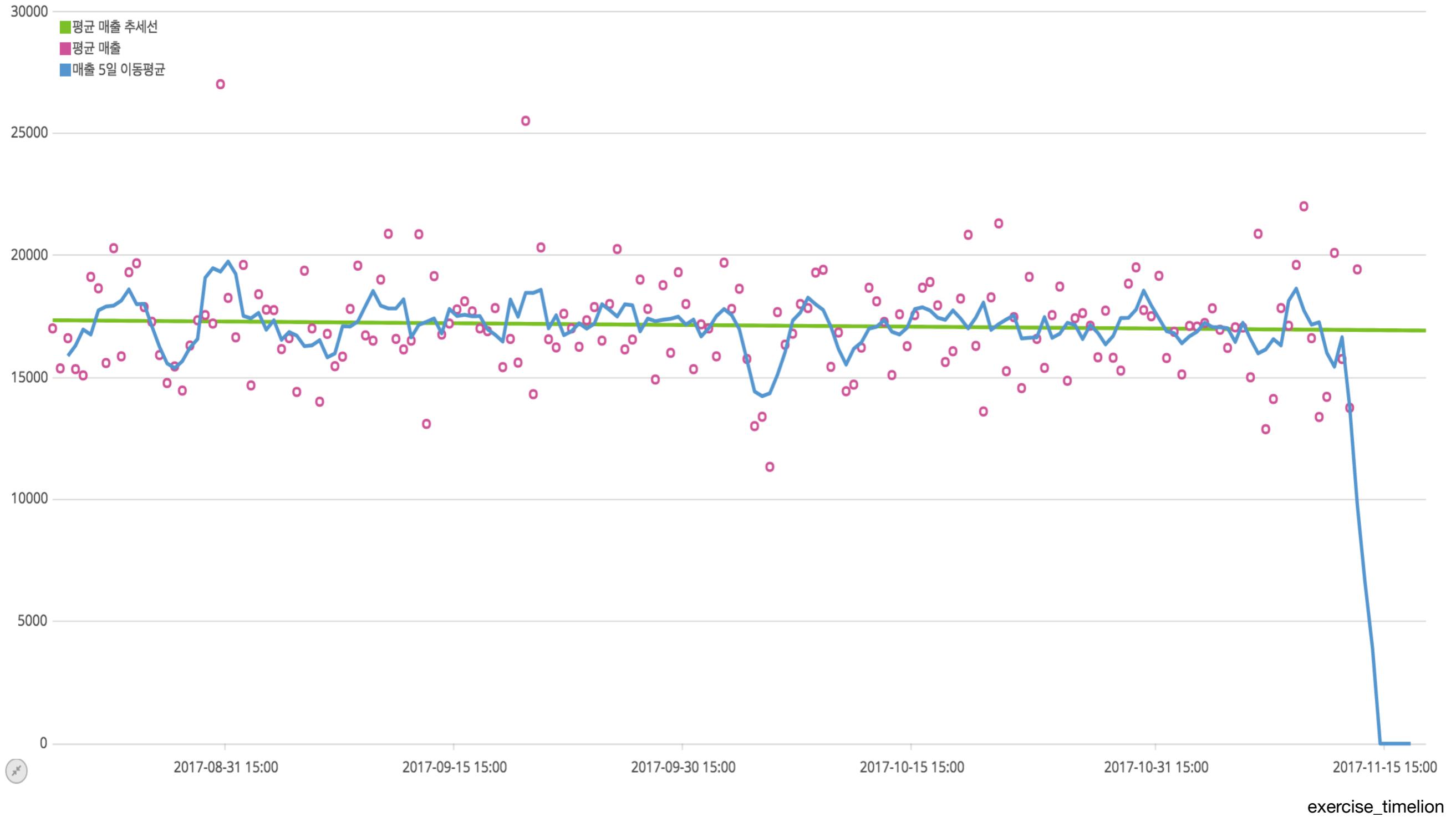
Field

Interval

Custom Label

문제 5

Week1 복습 - Visualize (Timelion)



Index : shopping
Time Range : 2017년 9월 1일 ~ 10월 31일

Week1 복습 - Visualize (Timelion)

문제

그래프 1 : 상품가격의 평균의 추세선을 그리세요

그래프 2 : 상품가격의 평균을 나타내는 그래프를 그리세요

그래프 3 : 상품가격 평균의 5일 이동평균선을 그리세요

힌트

.es(☆☆☆☆☆☆☆=▽▽▽:상품가격).▨▨▨▨▨▨(mode=linear).△△△△△('평균 매출 추세선').□□□□□(#7EC327),

.es(☆☆☆☆☆☆☆=▽▽▽:상품가격).◑◑◑◑◑◑◑(symbol=circle).△△△△△('평균 매출').□□□□□(#cf5297),

.es(☆☆☆☆☆☆☆=▽▽▽:상품가격).◇◇◇◇◇◇◇◇◇◇(window=5).△△△△△('매출 5일 이동평균').□□□□□(#5297cf)

다음을 약간 변경하면 됩니다.

.es(metric=avg:상품가격)

.es().trend(mode=log)

.es().label('예시 라벨')

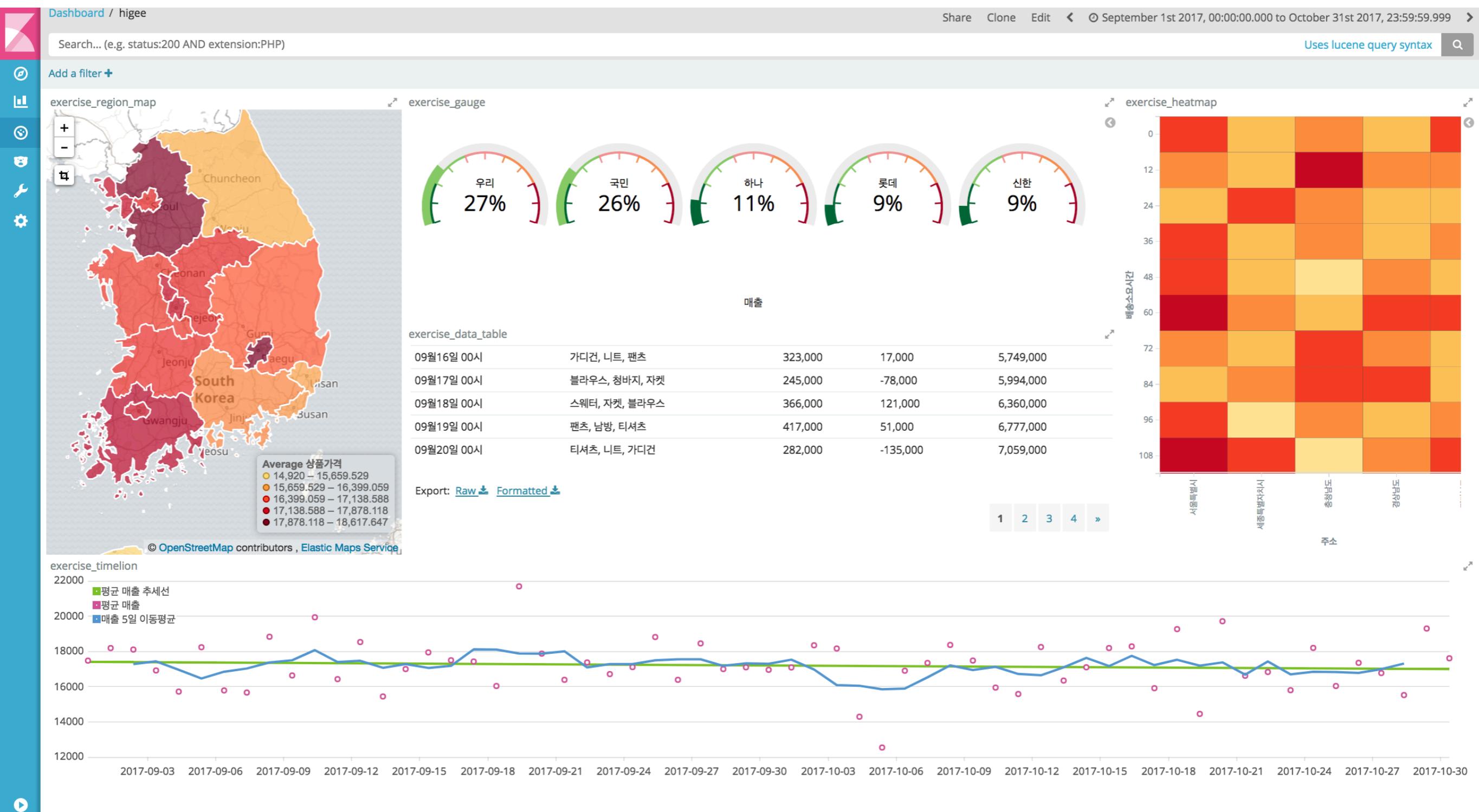
.es().color(black)

.es().points(symbol=diamond)

.es().movingaverage(window=10)

문제 6

Dashboard



Dashboard

문제

Dashboard를 새롭게 생성하고 위에서 만든 visualization을 하나하나 추가하세요.

단, visualization 사이즈 및 배치는 보기 편하게 재량껏 해주세요.

예)

- higee_week2_gauge
- higee_week2_region_map
- higee_week2_heat_map
- higee_week2_data_table
- higee_week2_timelion

완료 후에 dashboard를 index id로 저장하세요

예) higee

Managing Field

Managing Field

2017년 11월 20일 17시 44분

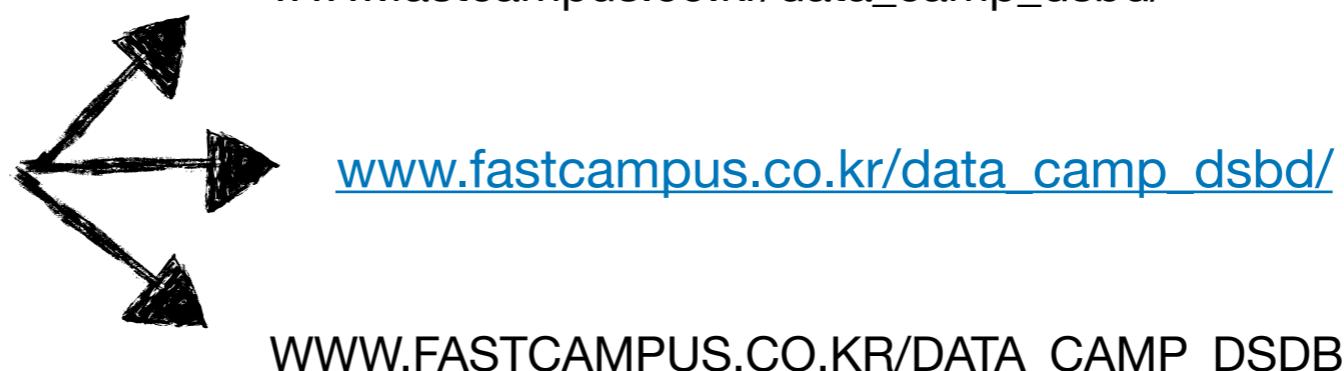
Date를 다르게 표현할 수 없나?



11/20/2017 5:44pm

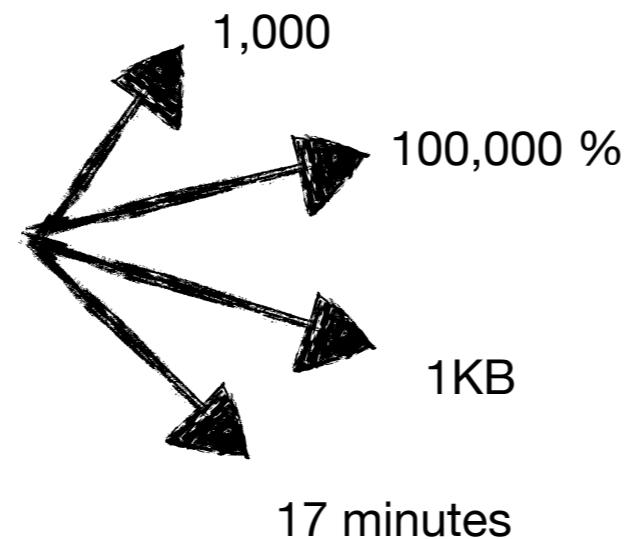
www.fastcampus.co.kr/data_camp_dsbd/

String를 다르게 보여줄 수 없나?



WWW.FASTCAMPUS.CO.KR/DATA_CAMP_DSDB

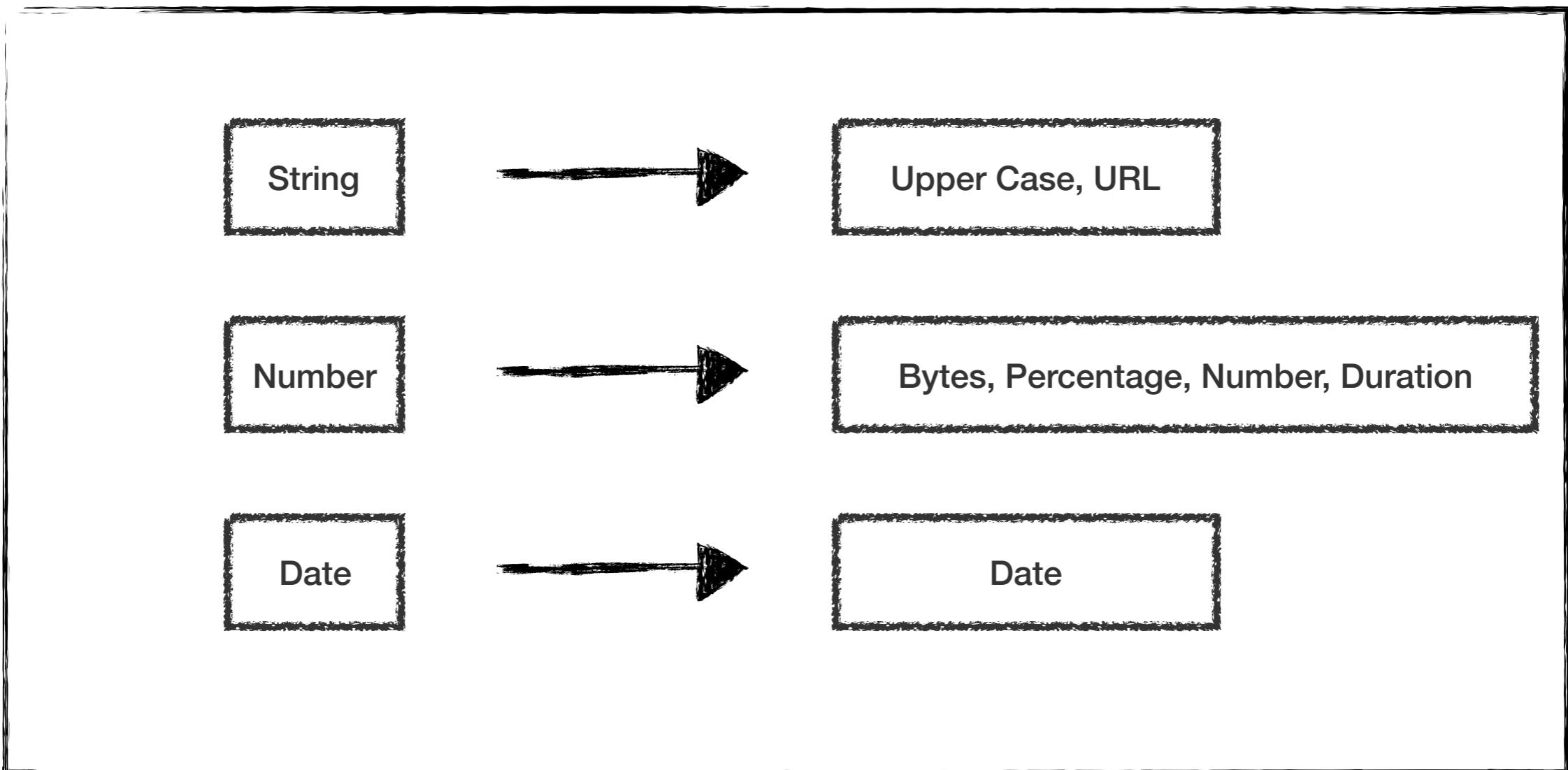
Number를 다르게 보여줄 수 없나?



17 minutes

Managing Field

실습



Index : wee2_{id}
Date Range : 2017.11.20~11.20

Managing Field

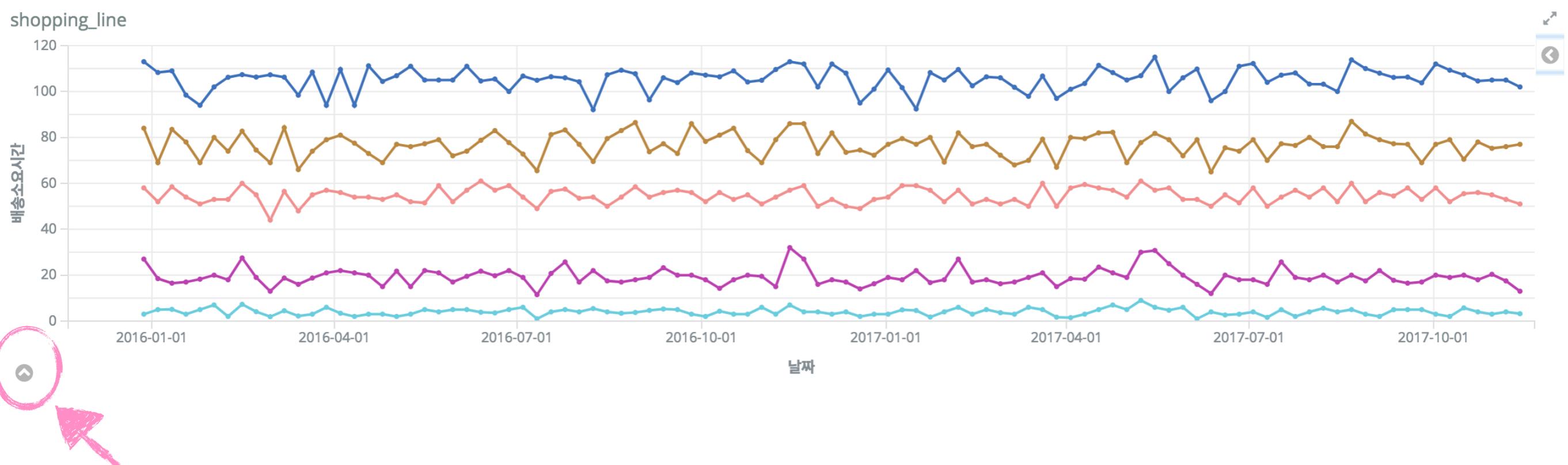
주의!!

Data Format이 변하는 것이지 Data 자체가 변하는 것이 아니다.
그러므로 Elasticsearch에 저장된 데이터 자체는 변하지 않는다!

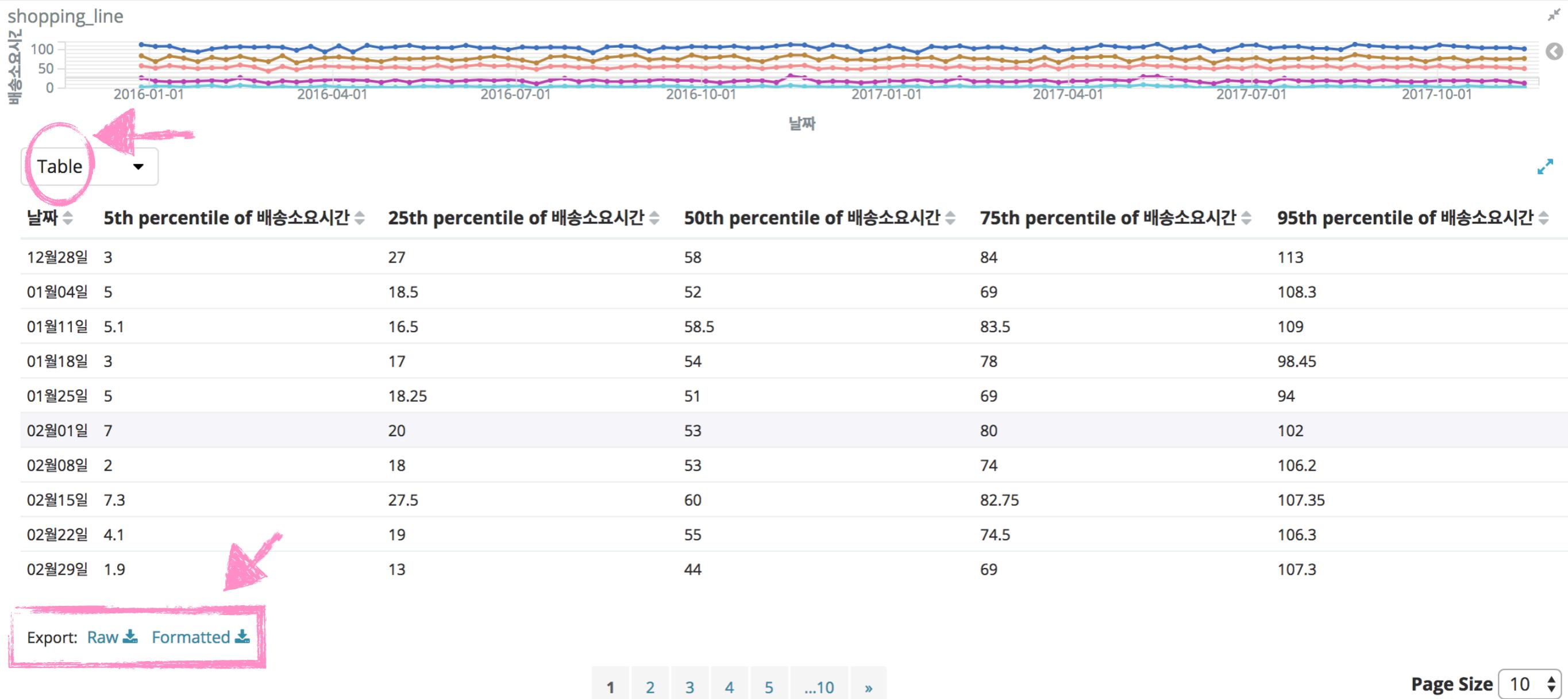
Visualization Spy

Visualization 결과를 csv로 다운 받을 수 없나?

Visualization Spy



Visualization Spy



Visualization Spy

| | A | B | C |
|----|--------------------------|---------------------------|--------------------------|
| | 5th percentile of 배송소요시간 | 25th percentile of 배송소요시간 | 5th percentile of 배송소요시간 |
| 1 | 날짜 | | |
| 2 | 1.45126E+12 | 3 | 27 |
| 3 | 1.45187E+12 | 5 | 18.5 |
| 4 | 1.45247E+12 | 5.1 | 16.5 |
| 5 | 1.45308E+12 | 3 | 17 |
| 6 | 1.45368E+12 | 5 | 18.25 |
| 7 | 1.45428E+12 | 7 | 20 |
| 8 | 1.45489E+12 | 2 | 18 |
| 9 | 1.45549E+12 | 7.3 | 27.5 |
| 10 | 1.4561E+12 | 4.1 | 19 |
| 11 | 1.4567E+12 | 1.9 | 13 |
| 12 | 1.45731E+12 | 4.5 | 18.75 |
| 13 | 1.45791E+12 | 2.2 | 16 |
| 14 | 1.45852E+12 | 3 | 18.75 |
| 15 | 1.45912E+12 | 6 | 21 |
| 16 | 1.45973E+12 | 3.4 | 22 |
| 17 | 1.46033E+12 | 2 | 21 |
| 18 | 1.46094E+12 | 3 | 20 |
| 19 | 1.46154E+12 | 3 | 15 |
| 20 | 1.46215E+12 | 2 | 21.75 |
| 21 | 1.46275E+12 | 3 | 15 |
| 22 | 1.46336E+12 | 5 | 22 |
| 23 | 1.46396E+12 | 4 | 21 |
| 24 | 1.46457E+12 | 5 | 17 |
| 25 | 1.46517E+12 | 5 | 19.5 |
| 26 | 1.46578E+12 | 3.85 | 21.75 |
| 27 | 1.46638E+12 | 3.55 | 19.75 |
| 28 | 1.46699E+12 | 5 | 22 |

raw

| | A | B | C |
|----|--------------------------|---------------------------|--------------------------|
| | 5th percentile of 배송소요시간 | 25th percentile of 배송소요시간 | 5th percentile of 배송소요시간 |
| 1 | 날짜 | | |
| 2 | 12월28일 | 3 | 27 |
| 3 | 01월04일 | 5 | 18.5 |
| 4 | 01월11일 | 5.1 | 16.5 |
| 5 | 01월18일 | 3 | 17 |
| 6 | 01월25일 | 5 | 18.25 |
| 7 | 02월01일 | 7 | 20 |
| 8 | 02월08일 | 2 | 18 |
| 9 | 02월15일 | 7.3 | 27.5 |
| 10 | 02월22일 | 4.1 | 19 |
| 11 | 02월29일 | 1.9 | 13 |
| 12 | 03월07일 | 4.5 | 18.75 |
| 13 | 03월14일 | 2.2 | 16 |
| 14 | 03월21일 | 3 | 18.75 |
| 15 | 03월28일 | 6 | 21 |
| 16 | 04월04일 | 3.4 | 22 |
| 17 | 04월11일 | 2 | 21 |
| 18 | 04월18일 | 3 | 20 |
| 19 | 04월25일 | 3 | 15 |
| 20 | 05월02일 | 2 | 21.75 |
| 21 | 05월09일 | 3 | 15 |
| 22 | 05월16일 | 5 | 22 |
| 23 | 05월23일 | 4 | 21 |
| 24 | 05월30일 | 5 | 17 |
| 25 | 06월06일 | 5 | 19.5 |
| 26 | 06월13일 | 3.85 | 21.75 |
| 27 | 06월20일 | 3.55 | 19.75 |
| 28 | 06월27일 | 5 | 22 |

format

여러 visualization data를 csv로 직접 추출해보자

JSON Input

JSON Input

Term Aggregation 시 5개 이하 Bucket은 제외할 수 없나?

이동평균을 구할 때 window size를 매번 다르게 할 수 없나?

missing data가 있을 때 특정한 값으로 대체할 수 없나?

Serial Diff Aggregation 시 여러개 전 Bucket하고 비교할 수 없나?

Date Histogram에서 날짜가 없는 Documents는 특정 날에 포함되게 할 수 없나?

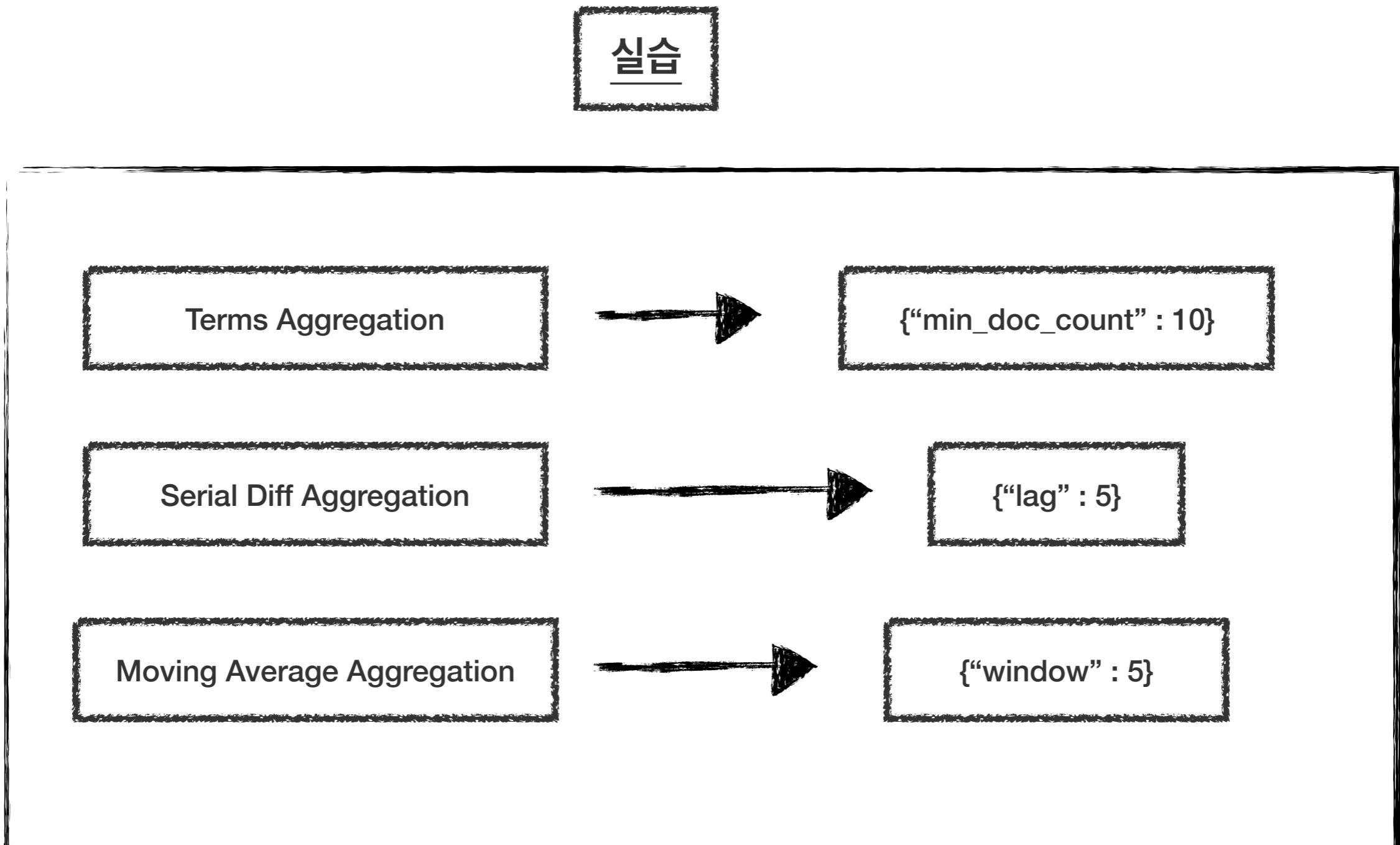
JSON Input

The screenshot shows the Datawrapper interface for configuring a JSON input visualization. On the left, there's a sidebar with various icons and a main panel for defining the data source. The main panel includes sections for 'shopping' (selected), 'metrics' (sum of 상품가격), 'Aggregation' (Sum), 'Field' (상품가격), 'Custom Label' (missing-data handled), 'JSON Input' (containing the value {"missing":100}), and 'buckets' (Split Rows). On the right, the data preview table shows a series of rows from 11월18일 03시 to 11월18일 10시. The 'default' column contains values like 10,000, while the 'missing-data handled' column contains values like 10,000 and 100. The row for 11월18일 07시 has a red box around its value '0' in the 'missing-data handled' column.

| 시간 | default | missing-data handled |
|------------|---------|----------------------|
| 11월18일 03시 | 10,000 | 10,000 |
| 11월18일 04시 | 10,000 | 10,000 |
| 11월18일 05시 | 10,000 | 10,000 |
| 11월18일 06시 | 10,000 | 10,000 |
| 11월18일 07시 | 0 | 100 |
| 11월18일 08시 | 0 | 100 |
| 11월18일 09시 | 10,000 | 10,000 |
| 11월18일 10시 | 10,000 | 10,000 |

Visualization : test_missing_json
Date Range : 2017.11.18 ~ 2017.11.18

JSON Input



Index : {id}
Date Range : Free

어떤 Aggregation에
어떤 parameter가 쓰이는지
다 외워야 되는지?

JSON Input

1. 사용하려는 Aggregation 검색(ex: Moving Average)
2. Parameter List 확인
3. Kibana JSON Input에 비슷하게 넣어보기
4. Visualization Spy - Request에서 확인하고 디버깅하기

```
POST /_search
{
  "size": 0,
  "aggs": {
    "my_date_histo": {
      "date_histogram": {
        "field": "date",
        "interval": "1M"
      },
      "aggs": {
        "the_sum": {
          "sum": { "field": "price" }
        },
        "the_movavg": {
          "moving_avg": {
            "buckets_path": "the_sum",
            "window": 30,
            "model": "simple"
          }
        }
      }
    }
  }
}
```

COPY

The screenshot shows the Kibana Metrics visualization configuration interface for a dashboard named 'shopping'. The configuration includes:

- Metrics**: Metric is set to 'Moving Avg'.
- Aggregation**: Aggregation is set to 'Count'.
- Custom Label**: An empty input field.
- JSON Input**: A text input containing the Elasticsearch search query from the left panel: `{"windo":10}`.
- buckets**: A section with 'Split Rows' checked and '주문시간 per 3 hours' selected. It also includes 'Add metrics' and 'Add sub-buckets' buttons.
- Advanced**: A large preview window showing the resulting Elasticsearch request. The request includes a date histogram with a 'window' of 3 hours and a 'size' of 0, followed by an aggregation stage with a 'moving_avg' of size 10.

Filtering by Field

**Dashboard는 만들었는데
원하는 조건의 데이터만 보고 싶으면?**

Filter by Field

1. Discover/Visualize/Dashboard 상관없이 왼쪽 상단의
2. 기본적으로 다음과 같은 화면이 나온다.

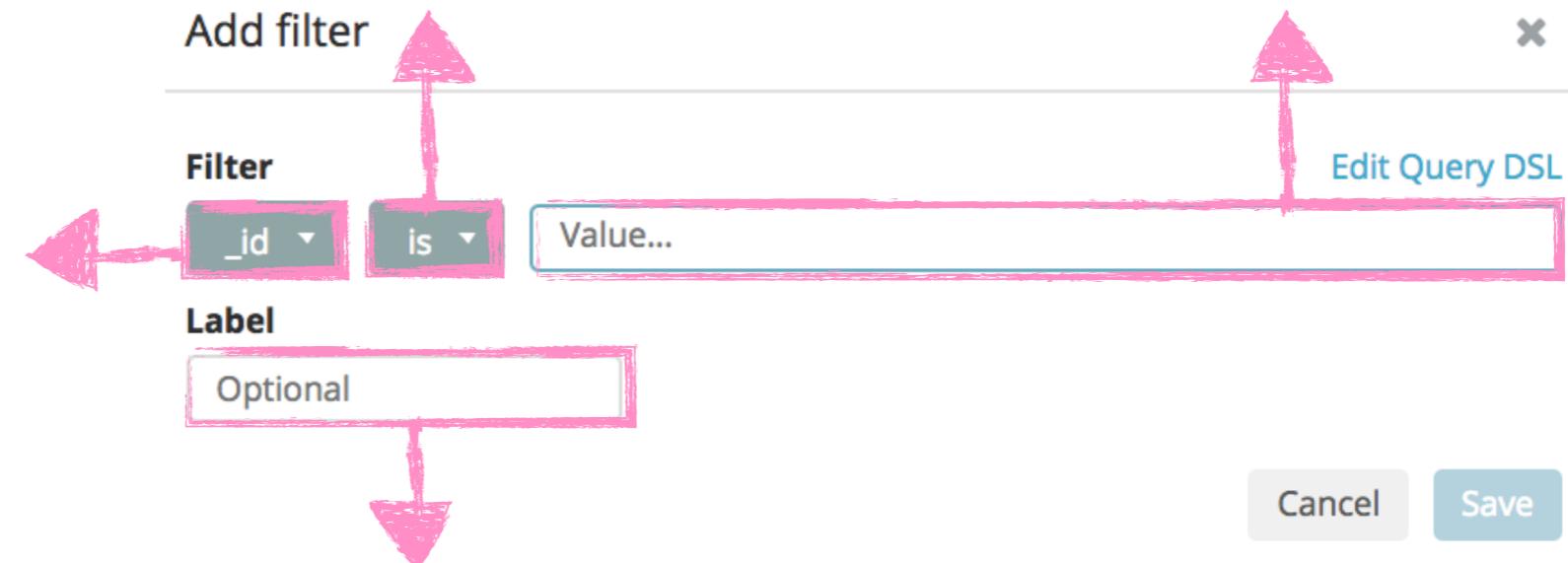
Add a filter +

선택

필터링 할 조건
예) 일치한다, 일치하지 않는다

비교할 값
예) 성별 is 여성, 지역 is not 서울

필터링 조건을 걸 Field 선택
예) 성별, 지역, 나이



가독성을 높이기 위해 필터에 이름 주기
예) 20대 여성, 서울 직장인

3. Field, Operator, Value, (Label)을 채우고 Save를 누른다.

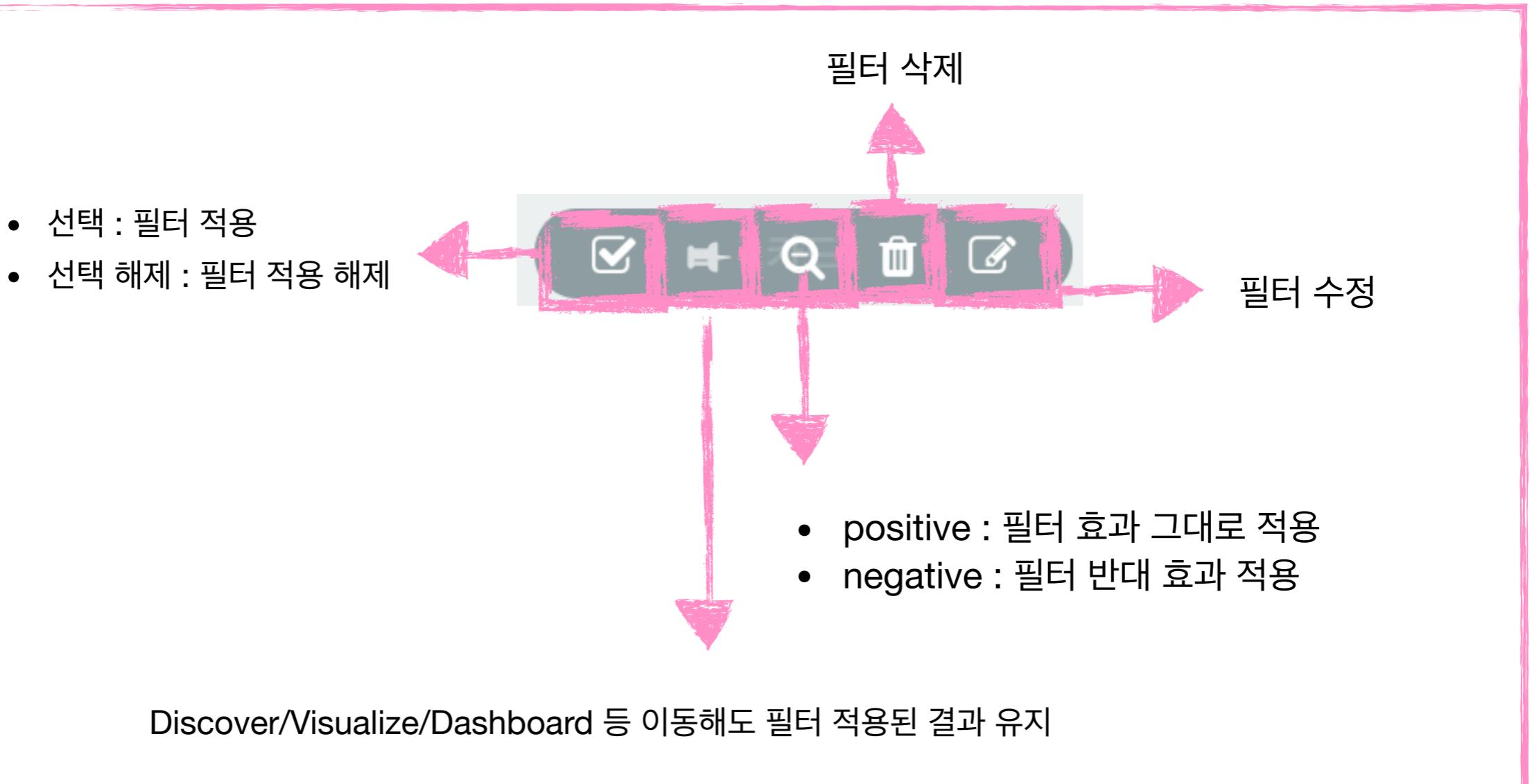
Filter by Field

Operator 설명

- is Field가 Value인 Documents 검색
- is not Field가 Value가 아닌 Documents 검색
- is one of Field가 Value 중에 하나라도 해당되는 Documents 검색
- is not one of Field가 Value 중에 하나라도 해당 안되는 Documents 검색
- exists Field가 존재하는 Documents 검색
- does not exist Field가 존재하지 않는 Documents 검색
- is between Field가 Value~Value 사이인 Documents 검색
- is not between Field가 Value~Value 사이에 없는 Documents 검색

Filter by Field

Filter를 적용하면 다음과 같은 화면이 표시된다.



Filter by Field

실습

1. 전체 Documents 개수는?
2. 고객성별이 여성인 Documents 개수는 ?
3. 결제카드가 우리 또는 국민인 Documents 개수는?
4. 결제카드가 우리 또는 국민이면서 고객나이가 30대인 Documents 개수는?
5. 결제카드가 우리 또는 국민이면서 고객나이가 30대이면서 고객주소_시도가 서울특별시가 아닌 Documents 개수는?
6. 구매사이트가 쿠팡 또는 옥션이면서 고객성별이 여성이며 상품개수가 1~3인 Documents 개수는?
7. 상품개수가 0~20이거나 3~5인 Documents 개수는?
8. 고객성별이 남성이면서 20대이거나 고객성별이 여성이면서 30대인 Documents 개수는?
9. 구매사이트가 22번가(오타 아니에요)와 매우 비슷한 Documents 개수는?

Index : shopping
Time Range : 2017년 9월 1일 ~ 10월 31일
Page : Discover

Lucene Query

**Dashboard는 만들었는데
원하는 조건의 데이터만 보고 싶으면?**

+

더 스마트한 검색!!

Lucene Query

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

Uses lucene query syntax



Discover/Visualize/Dashboard에 가면 상단에 위와 같은 검색 창이 있다.

구글에 검색하는 것처럼 검색하면 원하는 결과가 나올 때도 있다.

구글검색에도 규칙이 있듯이, **Lucene Query**를 익히면 **Kibana**에서도 꽤 괜찮은 결과를 얻을 수 있다.

Filter by Field

| 종류 | 기능 | 예시 |
|------------------------|---|-------------------|
| Keyword 검색 | Field에 상관없이 Value 일치하는 Documents 검색 | 여성 |
| Field Match 검색 | 특정 Field의 Value가 일치하는 Documents 검색 | 고객성별:여성 |
| Exact Field Match 검색 | 특정 Value가 정확히 모두 일치하는 Documents 검색 | 배송메모:"상품 이상" |
| Must be 검색 | 특정 Field가 존재하는 Documents 검색 | _exists_:구매사이트 |
| Must not be present 검색 | 특정 Field가 존재하지 않는 Documents 검색 | _missing_:구매사이트 |
| AND 검색 | 특정 조건들을 모두 만족하는 Documents 검색 | 여성 AND 셔츠 |
| OR 검색 | 특정 조건들 중 적어도 1개를 만족하는 Documents 검색 | 남성 OR 셔츠 |
| NOT 검색 | 특정 조건을 만족하지 않는 Documents 검색 | NOT 옥션 |
| Term 검색 | 조건 중 적어도 하나라도 만족하는 Documents 검색 | 상품분류: (니트 코트) |
| Fuzzy 검색 | 검색어와 유사한 Documents 검색 | 경상북도~ |
| Proximity 검색 | 검색어의 순서를 변경해서 찾을 수 있는 Documents 검색 | "이상 상품"~2 |
| Numeric Value 검색 | Numeric Field Value로 Documents 검색 | 상품가격:>5000 |
| Range 검색 | Numeric Field 범위로 Documents 검색 | 고객나이 : [10 TO 30] |
| Wildcard ? 검색 | Wildcard ? (임의의 한 글자)를 활용해서 Documents 검색 | 서?특별시 |
| Wildcard * 검색 | Wildcard * (* 앞에 글자로 시작하는 모든 결과)를 활용해서 Documents 검색 | 쿠* |

[source](#)

Filter by Field

실습

1. 전체 Documents 개수는?
2. 고객성별이 여성인 Documents 개수는 ?
3. 결제카드가 우리 또는 국민인 Documents 개수는?
4. 결제카드가 우리 또는 국민이면서 고객나이가 30대인 Documents 개수는?
5. 결제카드가 우리 또는 국민이면서 고객나이가 30대인이면서 고객주소_시도가 서울특별시가 아닌 Documents 개수는?
6. 구매사이트가 쿠팡 또는 옥션으면서 고객성별이 여성이며 상품개수가 1~3인 Documents 개수는?
7. 상품개수가 0~20이거나 4~5인 Documents 개수는?
8. 고객성별이 남성이면서 20대이거나 고객성별이 여성이면서 30대인 Documents 개수는?
9. 구매사이트가 22번가(오타 아니에요)와 매우 비슷한 Documents 개수는?
10. 고객주소_도시가 “경상”으로 시작하는 Documents 개수는?
11. 상품분류가 “셔”로 시작하는 2글자인 Documents 개수는?

Index : shopping
Time Range : 2017년 9월 1일 ~ 10월 31일
Page : Discover

Scripted Field

Scripted Field

Number Field 간 연산은 안되나?

Date Field 간 연산은 안되나?

Date Field에서 연/월/일/시/분/초 등 접근 안되나?

String Field에 String을 추가할 수 없나?

기존 Field에 조건을 적용해서 새로운 Field를 만들 수 없나?

Scripted Field

1. Management 이동
2. Index Patterns 선택
3. Index 선택 ({id}로 된 index 선택)
4. Scripted Fields 선택
5. Add Scripted Field 선택

The screenshot shows the Kibana Management interface under the Indices section, specifically for the 'shopping' index pattern. A pink arrow points from the 'Script' section of the configuration form towards the right side of the slide.

Management / Kibana / Indices / shopping

Index Patterns Saved Objects Advanced Settings

+ Create Index Pattern

★ shopping

Create Scripted Field

Name: New Scripted Field

Language: painless

Type: number

Format (Default: Number): - default - ⚠ Warning

Popularity: 0

Script:

```
script
```

52.78.61.155 | 2023-09-12 10:10:00 | caution

Scripted Field 코드 작성 부분

Scripted Field

Numeric Field

- 기본 문법 : `doc['Field명'].value`
- 활용
 - 단순
 - `doc['상품가격'].value + 10`
 - `doc['상품가격'].value - 10`
 - `doc['상품가격'].value * 10`
 - `doc['상품가격'].value / 10`
 - `doc['상품가격'].value % 10`
 - 복잡
 - `doc['상품가격'].value * doc['상품개수'].value`
 - `doc['상품가격'].value / doc['고객나이'].value`
 - `doc['판매자평점'].value + doc['상품가격'].value`

실습

1. $(상품가격 + 상품개수) * 3$
2. $(고객나이 - 판매자평점)$
3. $\text{판매자평점} * \text{판매자평점} / \text{고객나이}$

Scripted Field

Date Field

- 기본 문법 : doc['Field명'].date
- 활용 (클릭)
 - doc['t'].date.year
 - doc['t'].date.month
 - doc['t'].date.dayOfMonth
 - doc['t'].date.dayOfWeek
 - doc['t'].date.hourOfDay
 - doc['t'].date.minuteOfDay
 - doc['t'].date.secondOfDay



실습

1. 주문시간의 year를 Value로 갖는 Field
2. 주문시간의 요일(1~7)를 Value로 갖는 Field
3. 주문시간의 시간대(0~23)를 Value로 갖는 Field

Scripted Field

Logical Field

예시

```
if (doc['고객나이'].value < 20) {  
    return "10대"  
}  
if (doc['고객나이'].value < 30) {  
    return "20대"  
}  
if (doc['고객나이'].value < 40) {  
    return "30대"  
}  
if (doc['고객나이'].value < 50) {  
    return "40대"  
}  
return "50대 이상"
```



실습

- 상품개수에 따라 카테고리화 하기
- 1~2 : 저소비
 - 3~5 : 평균
 - 6~7 : 과소비