3. Dashboard Cookbook - Advanced Analytics

2023-07-07 10:57:40 AM to 2023-07-07 11:57:40 AM

Welcome to the dashboards demo for dashboards!

This is one of a series of demonstration dashboards see:

1. Basics
2. Time Series
3. Advanced Analytics
4. Advanced Techniques

Checkout these micro learning videos:

- Create a Dashboard , Create a Simple Dashboard , Customize a Dashboard , Share a Dashboard Inside Your Organization

This is a set of dashboards that demonstrate how to write and format search output, and match that to various panel types.

Advanced Analytics Operators: Time compare, Predict, outlier, geo lookup, smooth, transpose

Sumo Logic features many advanced analytics operators and techniques for data vizualization such as stacking over time, outlier, predict and others.

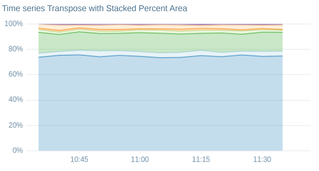
Stacking

Enable the stacking feature to compact bar or area charts over time.

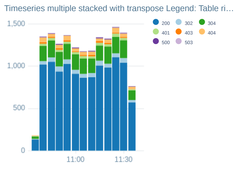
This is most useful with timeslice ... transpose multiple series data.

```
_sourceCategory=Labs/Apache/Access
| timeslice by 5m
| count _timeslice, status_code
 | transpose row _timeslice column status_code
```

Time series Transpose with Stacked Percent Area



Timeseries multiple stacked with transpose Legend: Table ri…

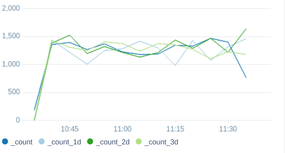


Time Compare

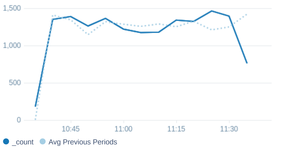
For a series over time we can add time compare to one or more previous periods. "this time last week", or this can be an average "average for this time for last 3 weeks"

```
_sourceCategory = Labs/Apache/Access
| timeslice by 5m
|count by _timeslice
 | compare with timeshift 7d 2
```

Time Compare



Time Compare (Avg)



Geo Lookup Maps

we can use geo lookup to add lat and long so we can use the map panel.

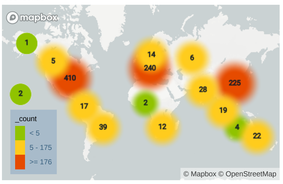
```
_sourceCategory = Labs/Akamai*| parse "\"cliIP\":\"*\"" as c
| lookup latitude, longitude, country_code from geo://locati

| count by latitude, longitude
```

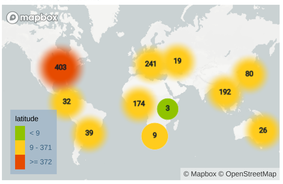
Note to cheat on a live dash where you are only allowed 1000 groups you can avg by cou

```
| avg(latitude) as latitude, avg(longitude) as longitude, cc
```

Geo lookup



Geo lookup avg by country



Logs and Metrics on Same Chart

We can put both metrics and logs series on the same chart. #A is logs, #B is metrics

here we have a logs query counting errors in instance logs, and logs showing avg cpu anc

logs

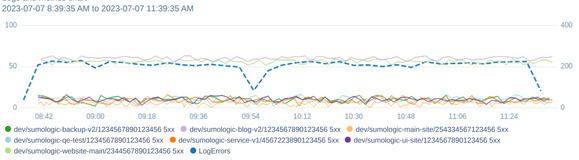
```
_sourcecategory=labs/apache/error error | timeslice
| count as LogErrors by _timeslice
```

metrics:

```
 Namespace=AWS/ApplicationELB (metric=HTTPCode_ELB_5XX_Count)
 | sum by loadbalancer
```

Logs and metrics chart

2023-07-07 8:39:35 AM to 2023-07-07 11:39:35 AM



Outlier

Given a series of time-stamped numerical values, using the outlier operator in a query can identify values in a sequence that seem unexpected, and would identify an alert or violation, for example, for a scheduled search.

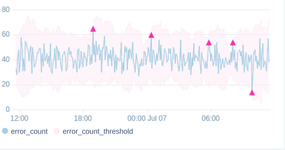
To do this, the Outlier operator tracks the moving average and standard deviation of a numerical field. An outlier is identified based on a specified threshold of standard deviations around the expected value. If a data point is outside the threshold, it is considered to be an outlier.

```
_sourceCategory=Labs/Apache/Access and status_code=404
    | timeslice 1m
    | count as error_count by _timeslice
    | outlier error_count
```

Outliers configurable:
`window=10,threshold=3,consecutive=3,direction=+-`

Count Outlier

2023-07-06 11:39:35 AM to 2023-07-07 11:39:35 AM



Count Outlier Custom Params

2023-07-06 11:39:35 AM to 2023-07-07 11:39:35 AM

Smooth Operator (Trend)

If so you will love how you can add a trendline with the smooth operator.

```
_sourceCategory = Labs/Apache/Access
 | timeslice by 5m
|count by _timeslice  | sort _timeslice asc
 |smooth _count as trend
```

Always sort smooth timeseries or as the operator does not do this by default

Adding Trend: The 'smooth operator'

2023-07-07 8:39:35 AM to 2023-07-07 11:39:35 AM

Predict

Uses a series of time-stamped numerical values to predict future values. The predict operator can be useful in the following cases:- As an early warning system, alerting you when a threshold is about to be reached.

- For resource and capacity planning, helpful for determining seasonal impacts, like a Cyber Monday rush on an e-commerce site. Improved risk calculation.

The predict operator supports two predictive models:

- Auto-regressive. Uses an advanced auto-regressive (AR) algorithm to learn patterns in the data. It automatically detects the cyclical patterns in the data and uses the cycles in its prediction.
- Linear regression. Uses existing data over the query time range as a training set to generate a linear model, and then extrapolates future values using this model.

```
_sourceCategory = Labs/Apache/Access
| timeslice by 1m
|count by _timeslice  | sort _timeslice asc
| predict _count by 1m model=ar, ar.window=5, forecast=100
```

Predict the future

2023-07-07 5:39:36 AM to 2023-07-07 11:39:36 AM

Scatter

Scatter charts display two independent numeric fields allowing you to see any correlation between them. You can visually determine how your fields relate to and affect one another.

The aggregate field is displayed as a collection of points. Each point requires two numerical fields for the X and Y axes.

```
_sourceCategory=service "message=User logged in" remote_ip
| parse "[remote_ip=*]" as remote_ip
| lookup latitude, longitude from geo://location on ip = remote_ip
| count as logins by latitude, longitude
```

Scatter

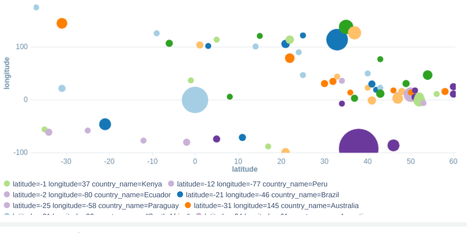
Bubble

Bubble charts display three dimensions of data. A bubble chart is a two dimensional scatter chart where each data point is represented by its size, the third dimension. This allows you to visualize the counts associated with each point. Bubble charts require at least one aggregate dimension and two other numeric dimensions.

The X dimension must be numeric and is displayed against the X axis of the bubble chart. The Y dimension must be numeric and is displayed against the Y axis of the bubble chart. The Z dimension is normally the aggregate field and shows the value of each bubble.

```
_sourceCategory=service "message=User logged in" remote_ip
| parse "[remote_ip=*]" as remote_ip
| lookup latitude, longitude, city, state from geo://location on ip = remote_ip
| count as logins by city, latitude, longitude
```

Bubble: lat x long x logins by country code

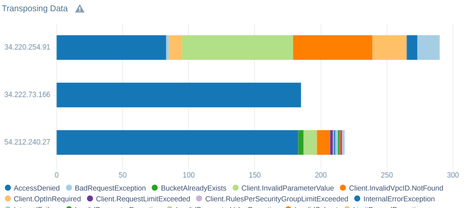


Transpose For Non Time Series Data

we can also use the transpose operator to stack by something other than a time series. Useful for building high density charts where we have a number of both row and column categories.

```
_sourceCategory=labs/aws/cloudtrail* errorcode
| json field=_raw "errorCode" | count by errorcode, _sourcehost
 | transpose row _sourcehost column errorcode
```

Transposing Data



Transaction Sankey Flow Charts

Using the transaction operator you can track your customer's movements through the log events that determine the states of

From the results of your query, you'd visualize your customers as they move or "flow" via a sankey diagram.

For example we have a demo coffee bar web service. Key purchase events might be:

- Payment in progress = new payment
- Calculation result = contains product order and total amount
- Query for db items sold - db query
- Update amount - inventory is recorded
- "Payment processed successfully" or "Payment failed"

```
_sourcecategory = kubernetes/the/coffee/bar/ns/sumologic/the/coffee/bar/* "trace_id"
service="the-cashdesk"
("Payment in progress" or "Payment processed successfully" or "Payment failed" or "
| json field=_raw "log"
| parse field=log "* * - * - * - * trace_id=* - span_id=*" as d,t,user,level,event,
| parse regex field=event "(?<stage>Payment in progress|Payment processed successfu
| tolowercase(stage) as stage | replace (stage," ","") as stage
| transaction on trace_id with states paymentinprogress, calculationresult, queryd
updateamount, paymentprocessedsuccessfully,paymentfailed  in stage results by flow
| count ,max(latency) by fromstate,tostate
```

Sankey Transaction Flow Diagram

2023-07-07 5:39:37 AM to 2023-07-07 11:39:37 AM