splunk> .conf2017

Power Of SPL

Stephen Luedtke | Sr. Technical Marketing Manager

September 27, 2017 | Washington, DC

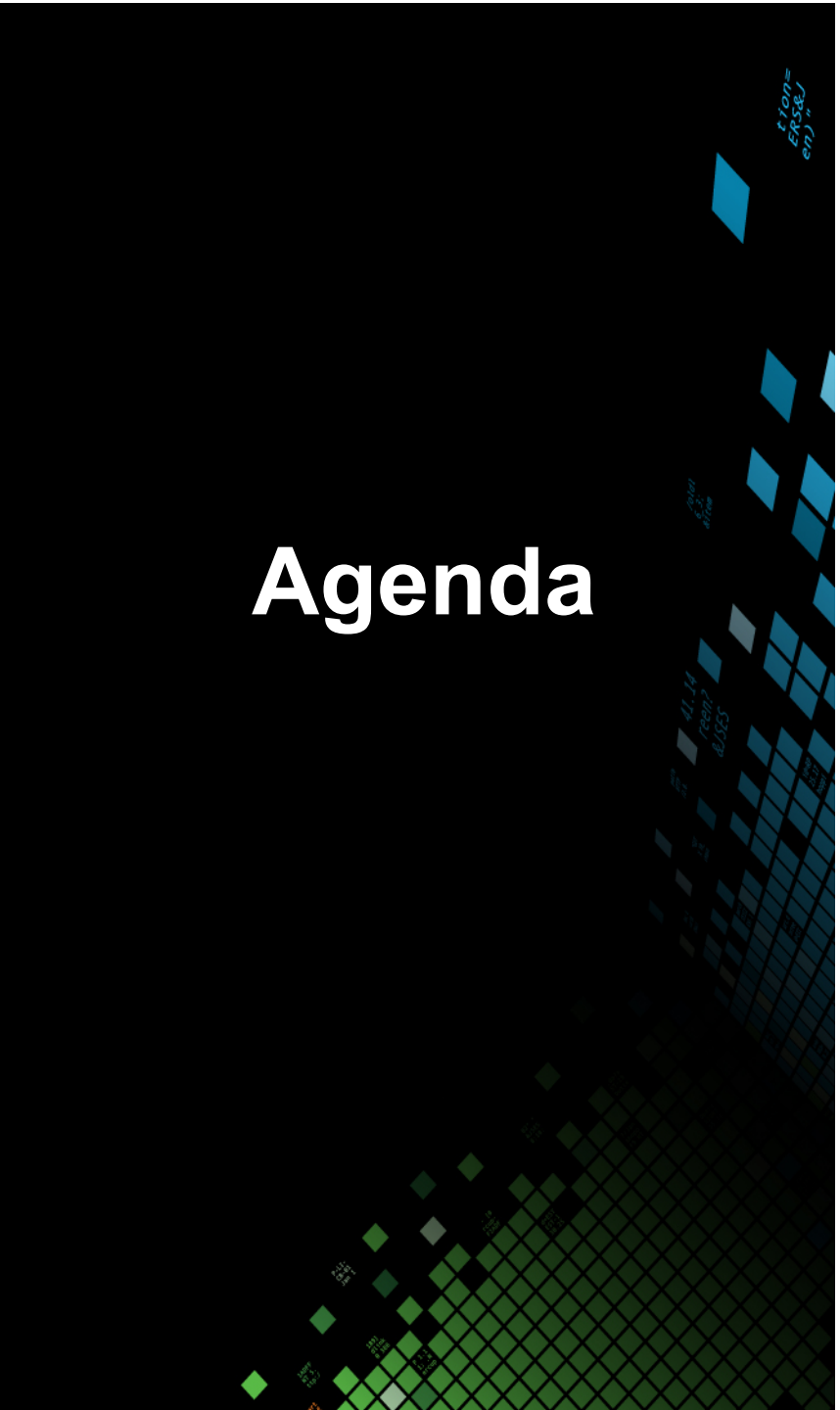
Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

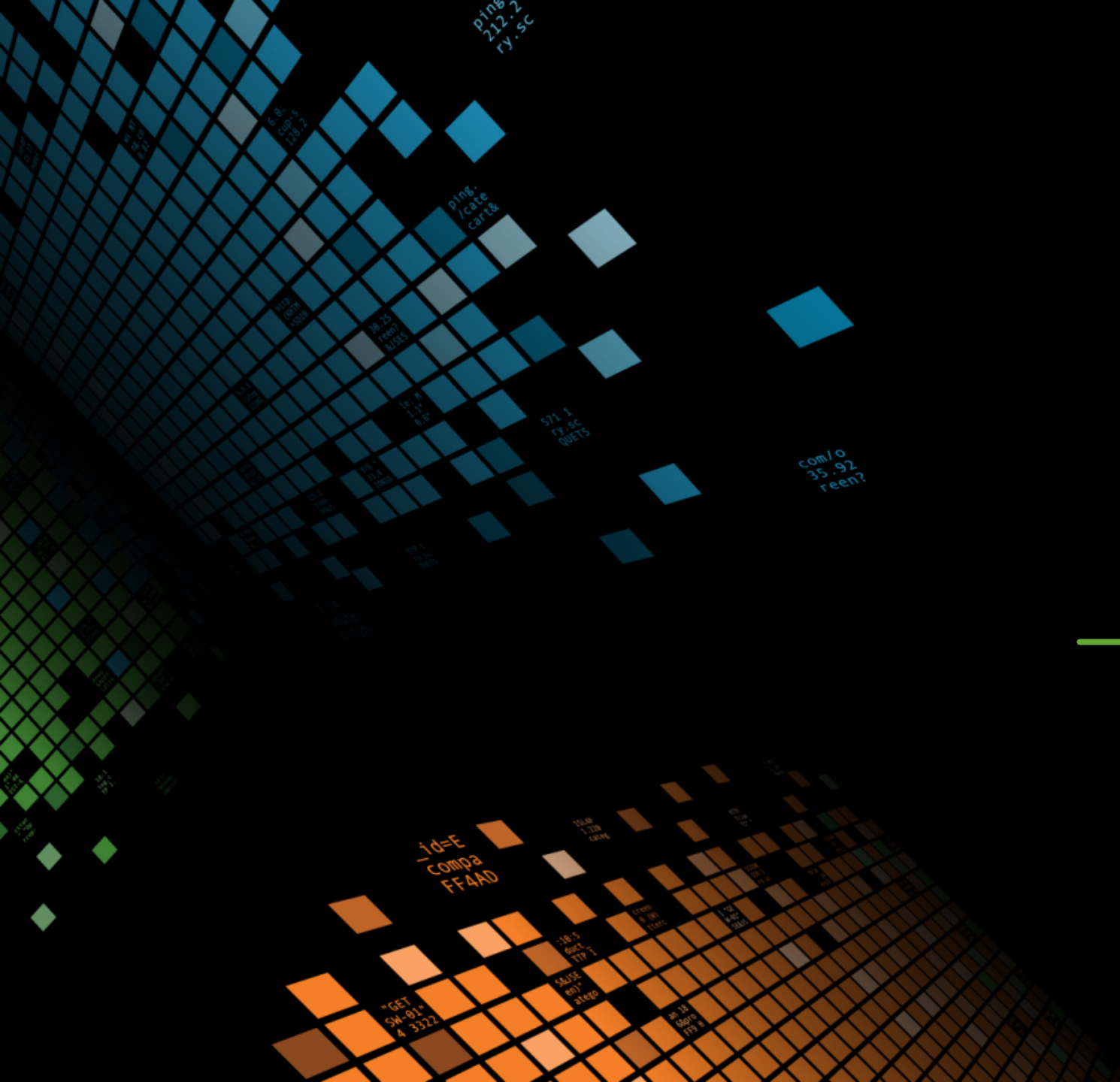
The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

splunk> .conf2017

Agenda



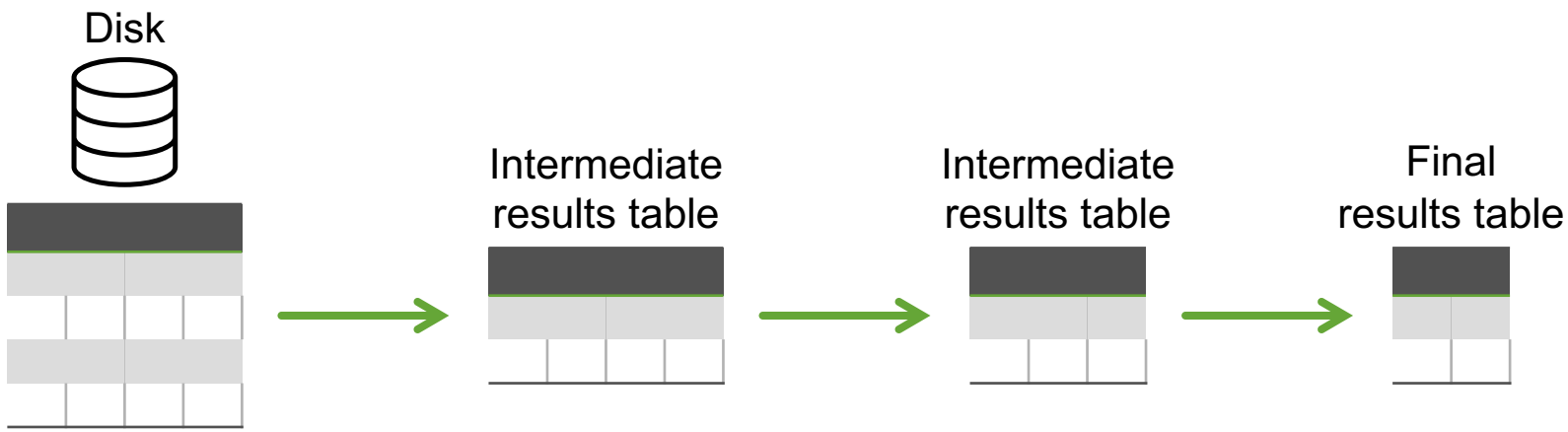
1. Overview & Anatomy of a Search
   - Quick refresher on search language and structure
2. SPL Commands and Examples
   - Searching, charting, converging, mapping, transactions, anomalies, exploring
3. Custom Commands
   - Extend the capabilities of SPL
4. Q&A



SPL Overview

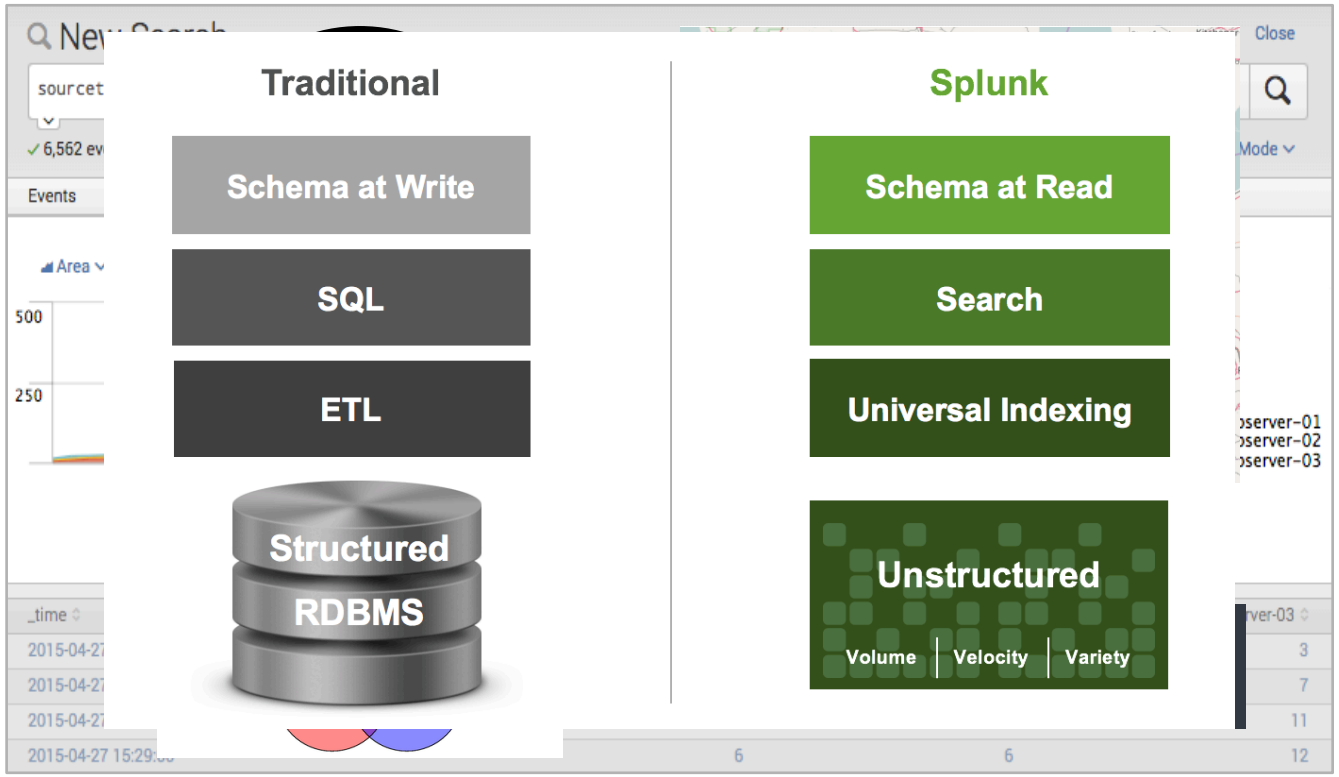
SPL Overview

- Over 140 search commands
- Syntax was originally based upon the **Unix pipeline** and SQL and is optimized **for time-series data**
- The scope of SPL includes data searching, filtering, modification, manipulation, enrichment, insertion and deletion
- Includes machine learning such as anomaly detection



Why Create A New Query Language?

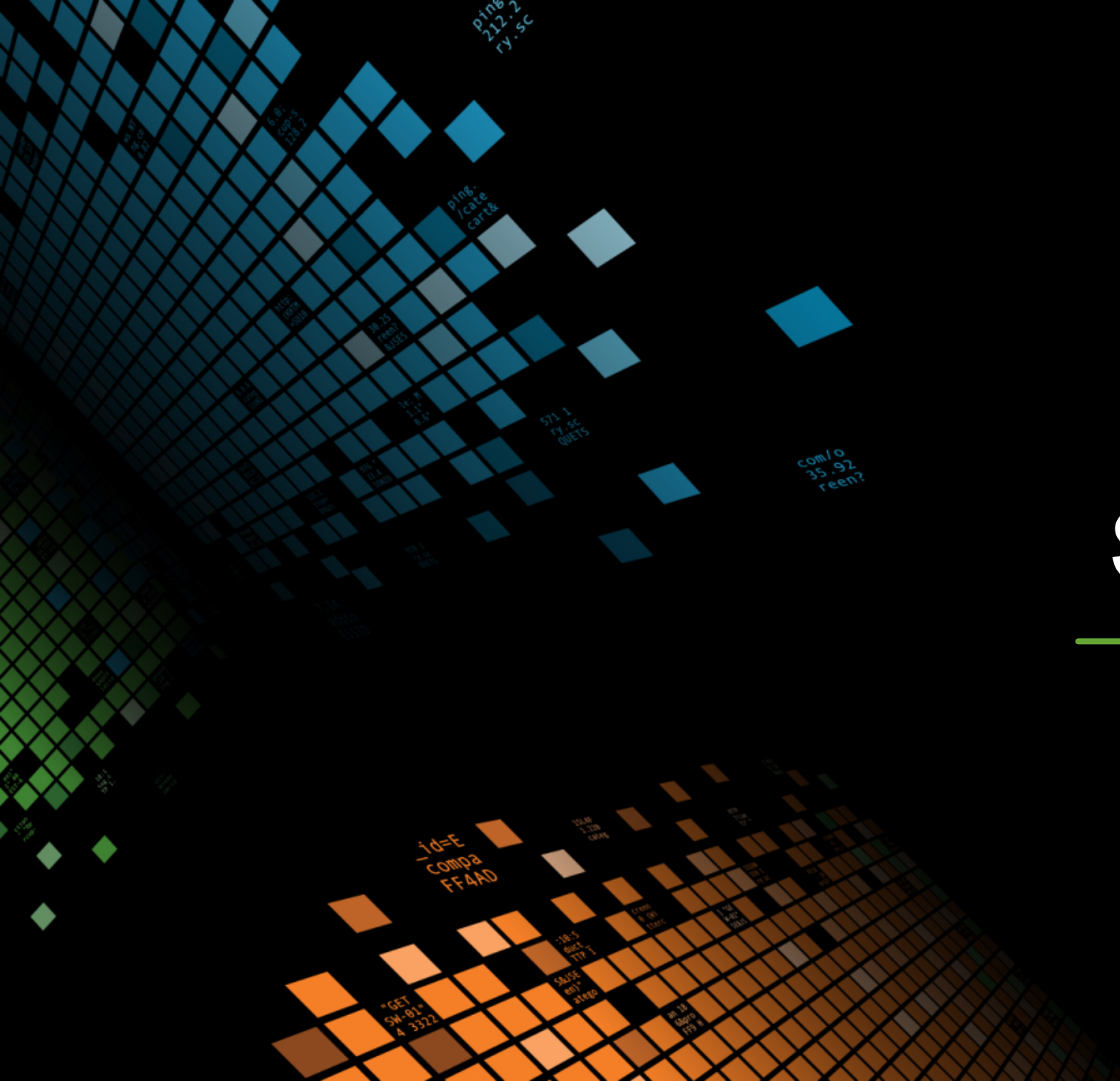
- Flexibility and effectiveness on *small* and ***big*** data
- Late-binding schema
- More/better methods of correlation
- Not just analyze, but visualize



SPL Basic Structure

search and filter | munge | report | cleanup

```
sourcetype=access*
| eval KB=bytes/1024
| stats sum(KB) dc(clientip)
| rename sum(KB) AS "Total KB" dc(clientip) AS "Unique Customers"
```



SPL Examples

SPL Examples And Recipes

- Find the needle in the haystack
- Charting statistics and predicting values
- Enriching and converging data sources
- Map geographic data in real time
- Identifying anomalies
- Transactions
- Data exploration & finding relationships between fields
- Custom commands

SPL Examples And Recipes

- **Find the needle in the haystack**
- Charting statistics and predicting values
- Enriching and converging data sources
- Map geographic data in real time
- Identifying anomalies
- Transactions
- Data exploration & finding relationships between fields
- Custom Commands

Search And Filter

Examples

- **Keyword search:**

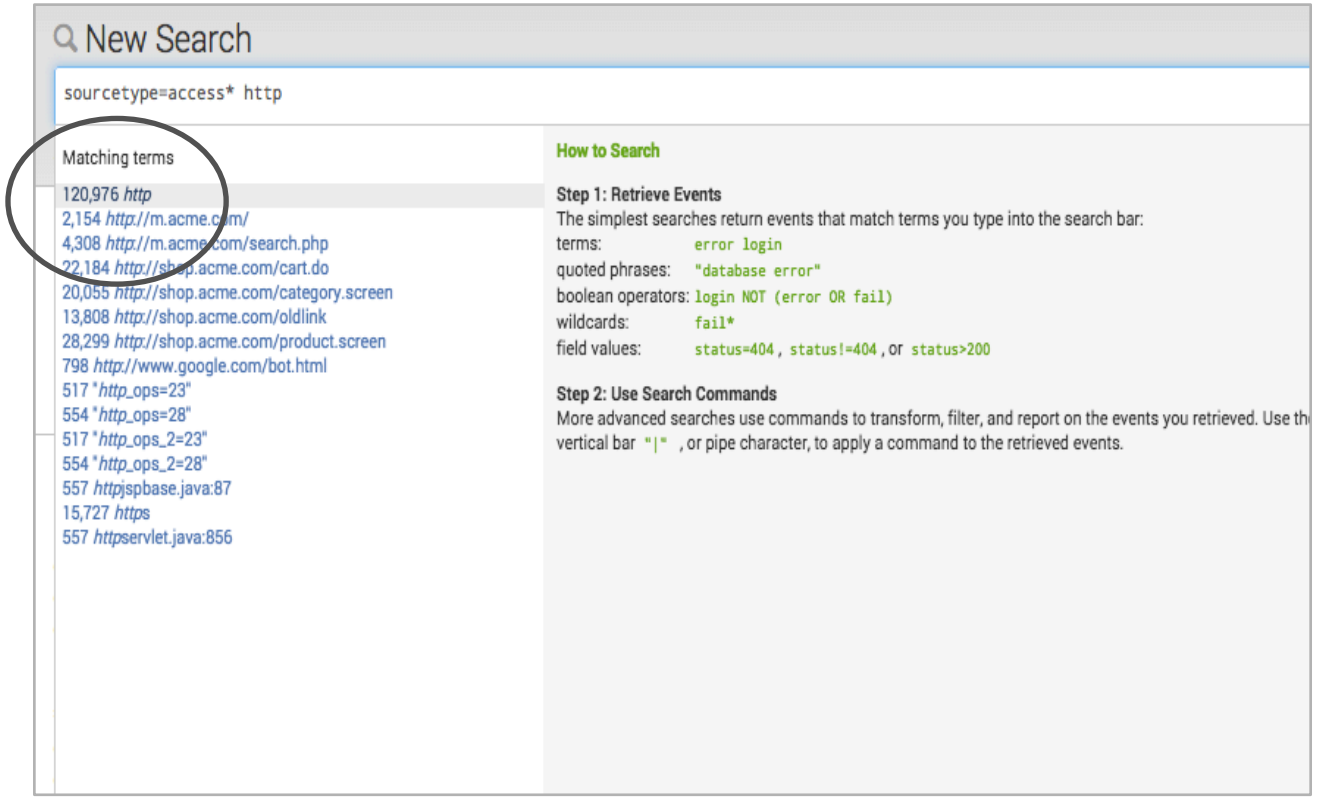
  sourcetype=access* http

- Filter:

  sourcetype=access* http
  host=webserver-02

- Combined:

  sourcetype=access* http
  host=webserver-02 (503 OR 504)



splunk> .conf2017

Search And Filter

Examples

- Keyword search:

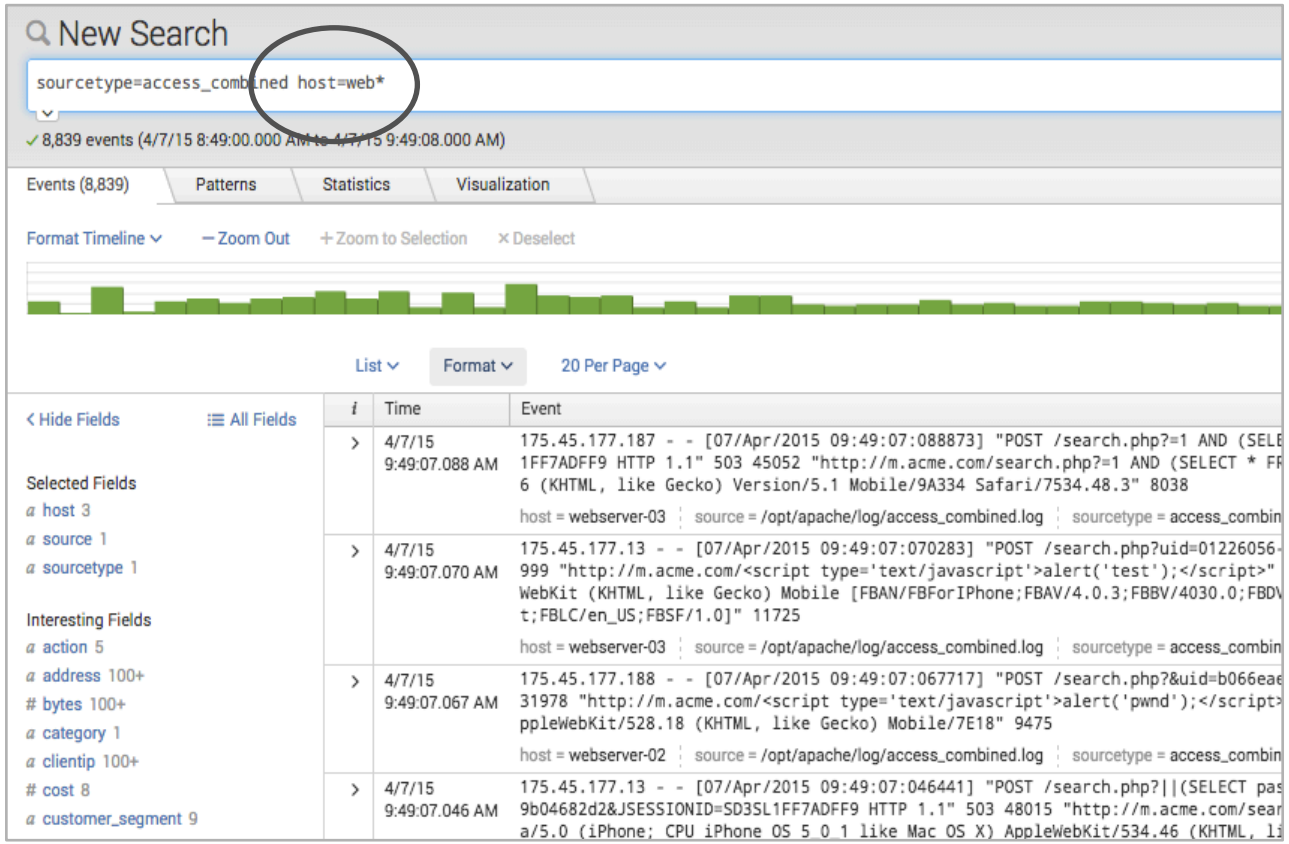
  sourcetype=access* http

- **Filter:**

  sourcetype=access* http
  host=webserver-02

- Combined:

  sourcetype=access* http
  host=webserver-02 (503 OR 504)



splunk> .conf2017

Search And Filter

Examples

- Keyword search:

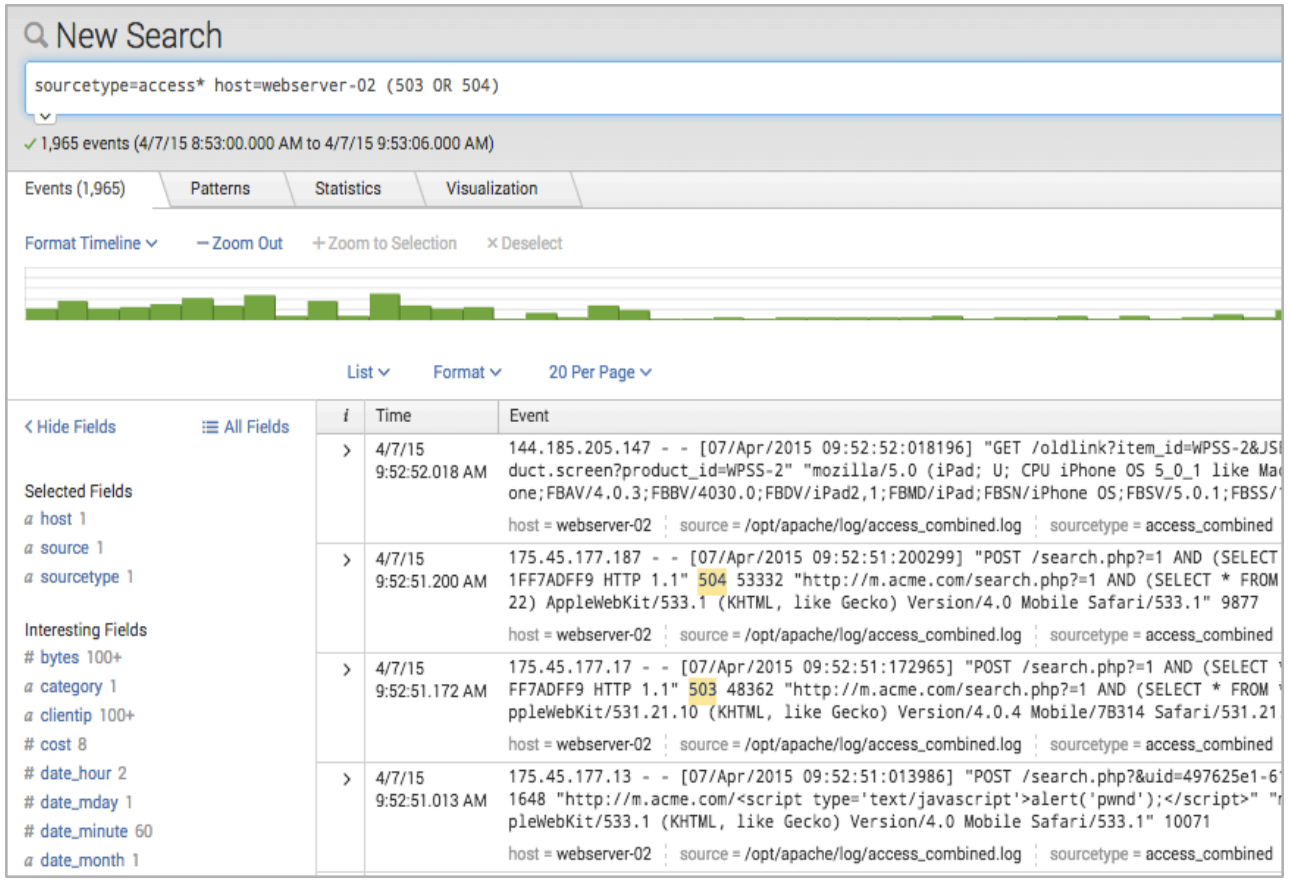
  sourcetype=access* http

- Filter:

  sourcetype=access* http
  host=webserver-02

- **Combined:**

  sourcetype=access* http
  host=webserver-02 (503 OR 504)



splunk> .conf2017

Eval – Modify or Create New Fields and Values

Examples

- **Calculation:**

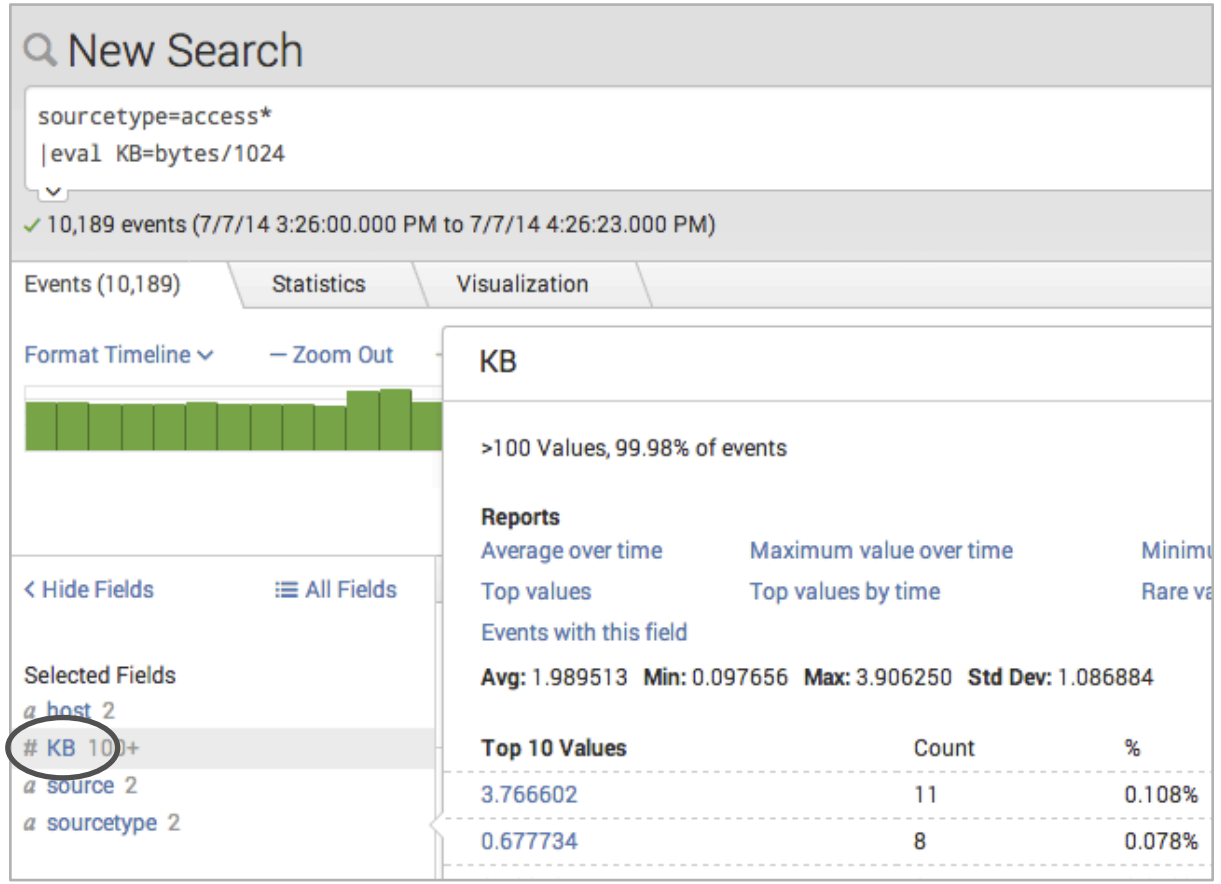
  sourcetype=access*
  |eval KB=bytes/1024

- Evaluation:

  sourcetype=access*
  | eval http_response =
  if(status != 200, ”Error", ”OK”)

- Concatenation:

  sourcetype=access*
  | eval connection = device.” - ".clientip



splunk> .conf2017

Eval – Modify or Create New Fields and Values

Examples

- Calculation:

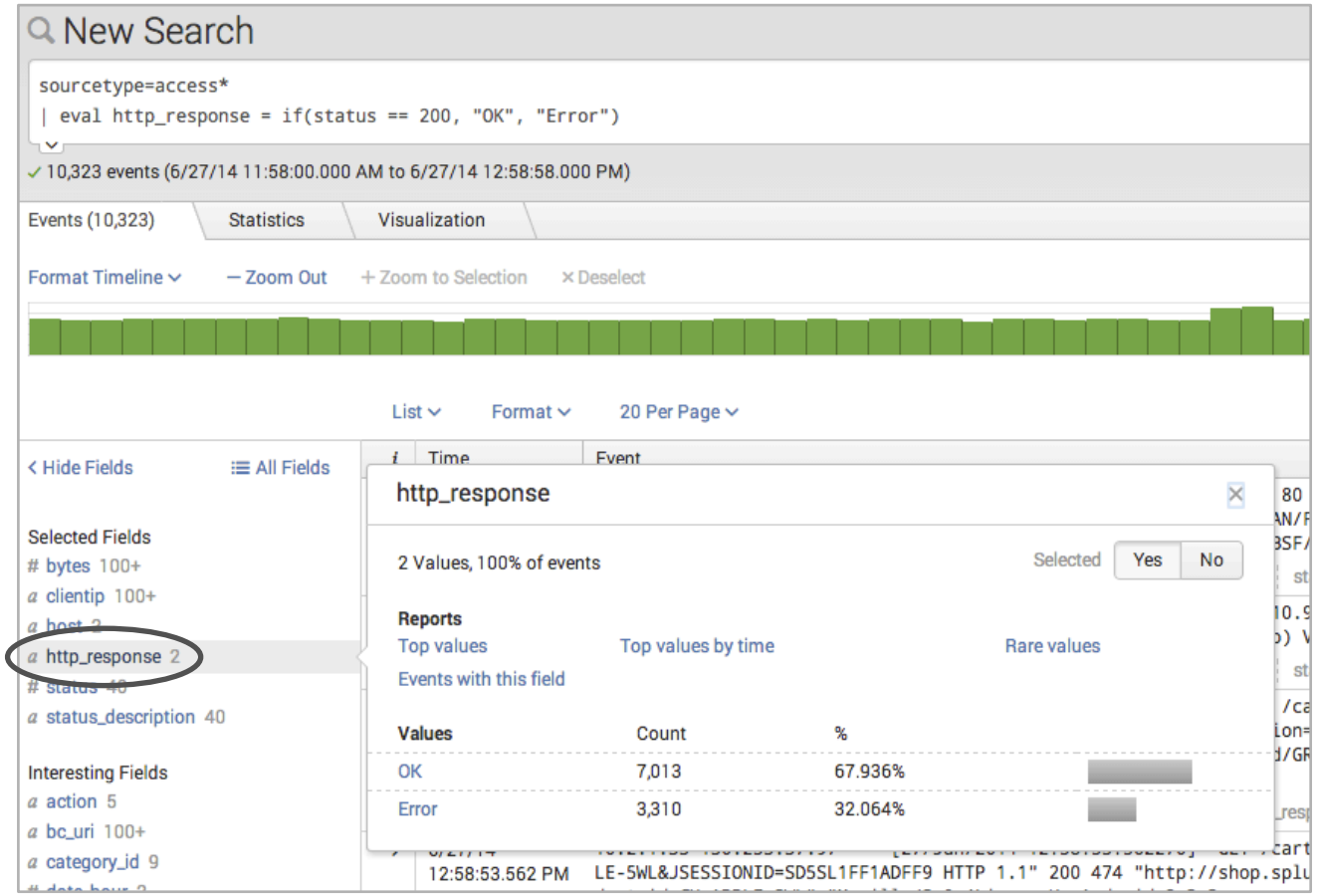
  sourcetype=access*
  |eval KB=bytes/1024

- **Evaluation:**

  sourcetype=access*
  | eval http_response =
  if(status != 200, ”Error", ”OK”)

- Concatenation:

  sourcetype=access*
  | eval connection = device.” - ".clientip



Eval – Modify or Create New Fields and Values

Examples

- Calculation:

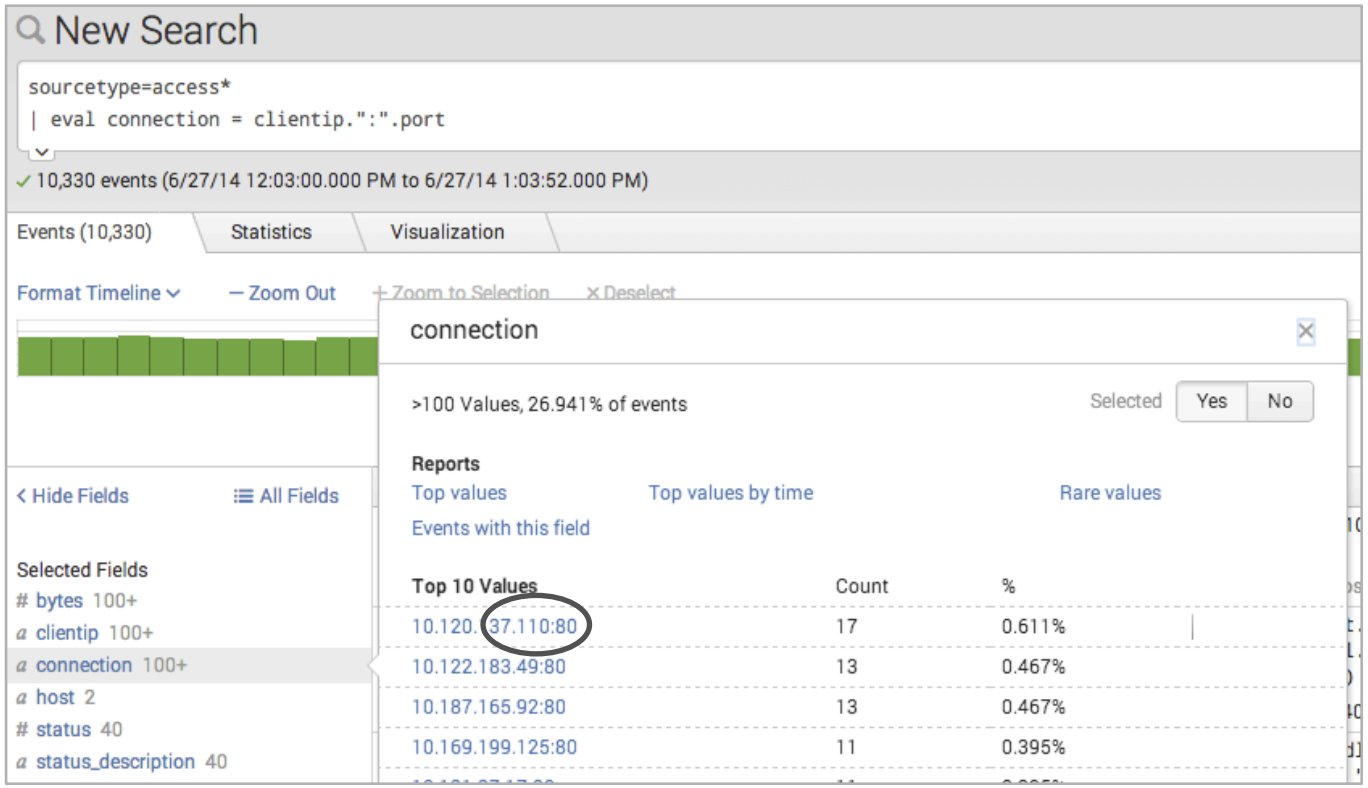
```
sourcetype=access*
|eval KB=bytes/1024
```

- Evaluation:

```
sourcetype=access*
| eval http_response =
if(status != 200, ”Error", ”OK”)
```

- **Concatenation:**

```
sourcetype=access*
| eval connection = device.” - ".clientip
```



splunk> .conf2017

Eval – Just Getting Started!

Splunk Search Quick Reference Guide

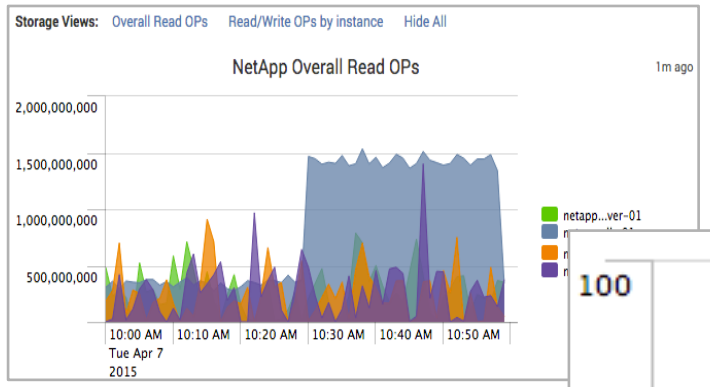
Common Eval Functions	The eval command calculates an expression and puts the resulting value into a field (e.g. "...| eval force = mass * acceleration"). The following table lists some of the functions used with the eval command. You can also use basic arithmetic operators (+ - * / %), string concatenation (e.g., "...| eval name = last . "," . first"), and Boolean operations (AND OR NOT XOR < > <= >= != = == LIKE).

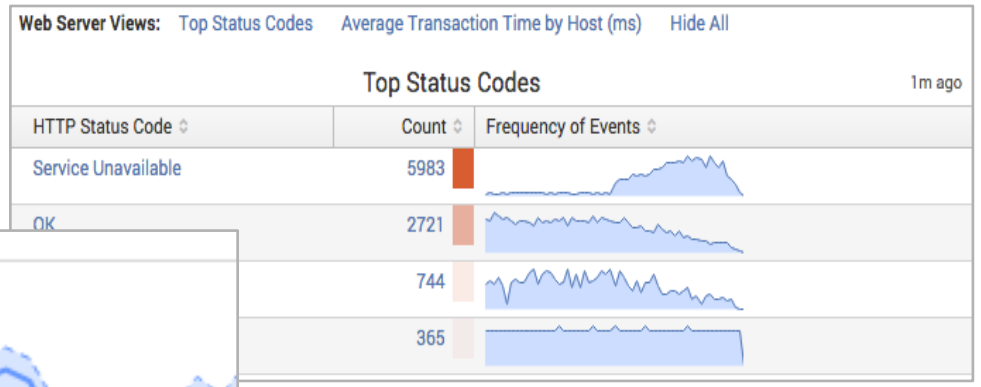
Function	Description	Examples
`abs(X)`	Returns the absolute value of X.	abs(number)
`case(X,"Y",…)`	Takes pairs of arguments X and Y, where X arguments are Boolean expressions. When evaluated to TRUE, the arguments return the corresponding Y argument.	case(error == 404, "Not found", error == 500,"Internal Server Error", error == 200, "OK")
`ceil(X)`	Ceiling of a number X.	ceil(1.9)
`cidrmatch("X",Y)`	Identifies IP addresses that belong to a particular subnet.	cidrmatch("123.132.32.0/25",ip)
`coalesce(X,…)`	Returns the first value that is not null.	coalesce(null(), "Returned val", null())
`cos(X)`	Calculates the cosine of X.	n=cos(0)
`exact(X)`	Evaluates an expression X using double precision floating point arithmetic.	exact(3.14*num)
`exp(X)`	Returns eX.	exp(3)
`if(X,Y,Z)`	If X evaluates to TRUE, the result is the second argument Y. If X evaluates to FALSE, the result evaluates to the third argument Z.	if(error==200, "OK", "Error")
`isbool(X)`	Returns TRUE if X is Boolean.	isbool(field)
`isint(X)`	Returns TRUE if X is an integer.	isint(field)
`isnull(X)`	Returns TRUE if X is NULL.	isnull(field)
`isstr()`	Returns TRUE if X is a string.	isstr(field)
`len(X)`	This function returns the character length of a string X.	len(field)
`like(X,"Y")`	Returns TRUE if and only if X is like the SQLite pattern in Y.	like(field, "addr%")

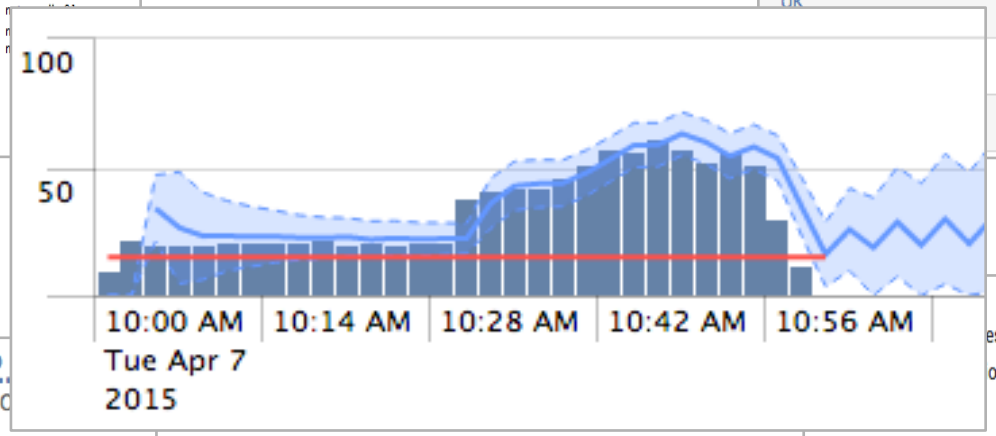
SPL Examples And Recipes

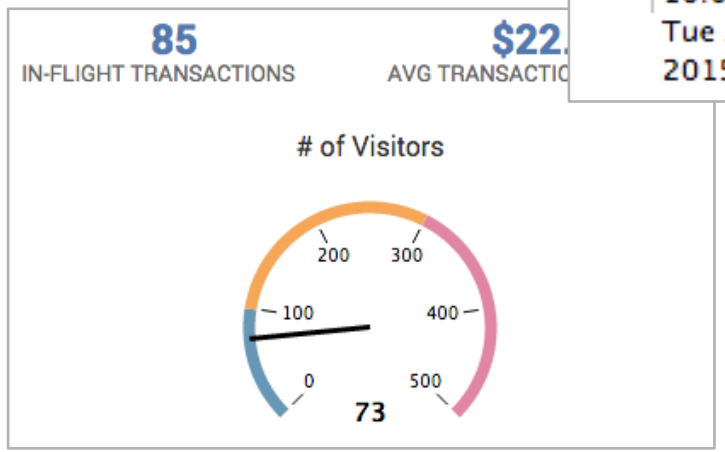
- Find the needle in the haystack
- **Charting statistics and predicting values**
- Enriching and converging data sources
- Map geographic data in real time
- Identifying anomalies
- Transactions
- Data exploration & finding relationships between fields
- Custom commands

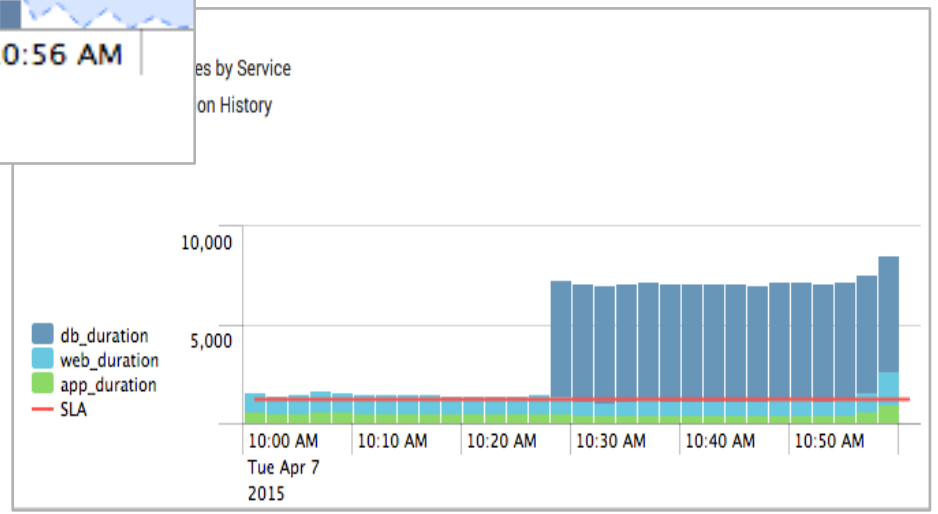
Stats, Timechart, Eventstats, Streamstats











Stats – Calculate Statistics Based on Field Values

Examples

- **Calculate stats and rename**

  Index=power_of_spl

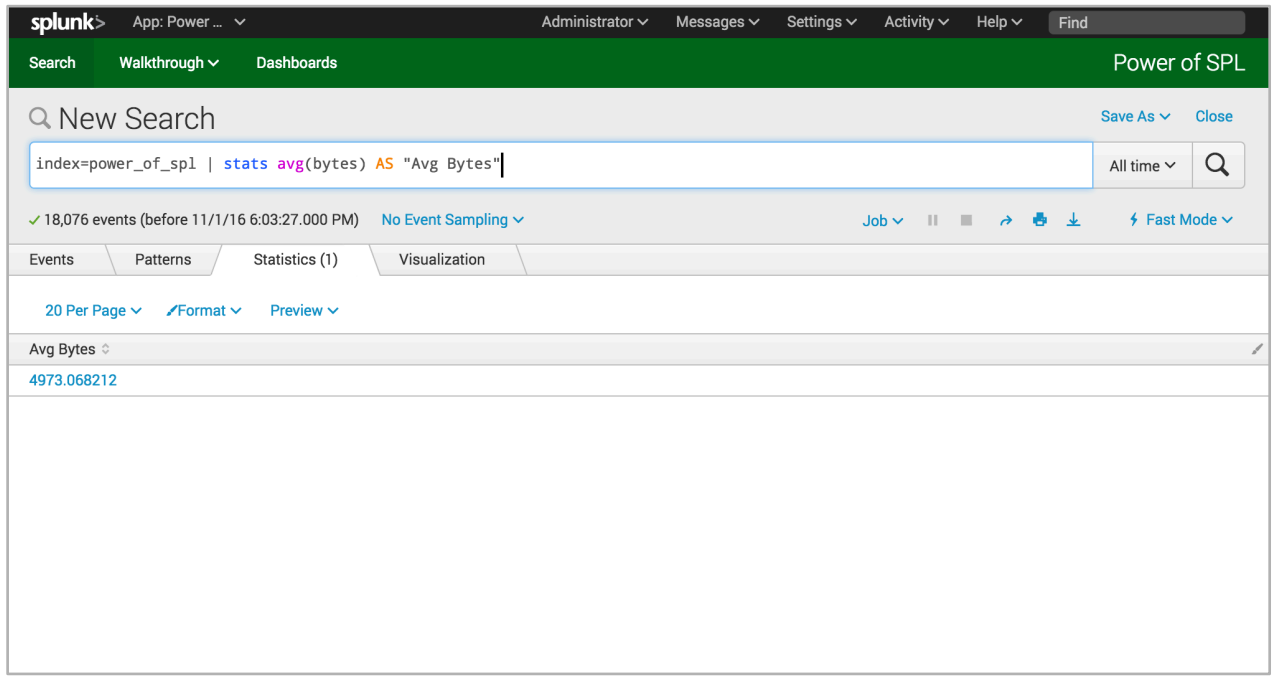
  | stats avg(bytes) AS “Avg Bytes”

- Multiple statistics

  index=power_of_spl | stats avg(bytes) AS bytes sparkline(avg(bytes)) AS Bytes_Trend min(bytes) max(bytes)

- By another field

  index=power_of_spl
  | stats avg(bytes) AS avg_bytes sparkline(avg(bytes)) AS Bytes_Trend min(bytes) max(bytes) **by** clientip | sort - avg_bytes



Stats – Calculate Statistics Based on Field Values

Examples

- Calculate stats and rename

  Index=power_of_spl

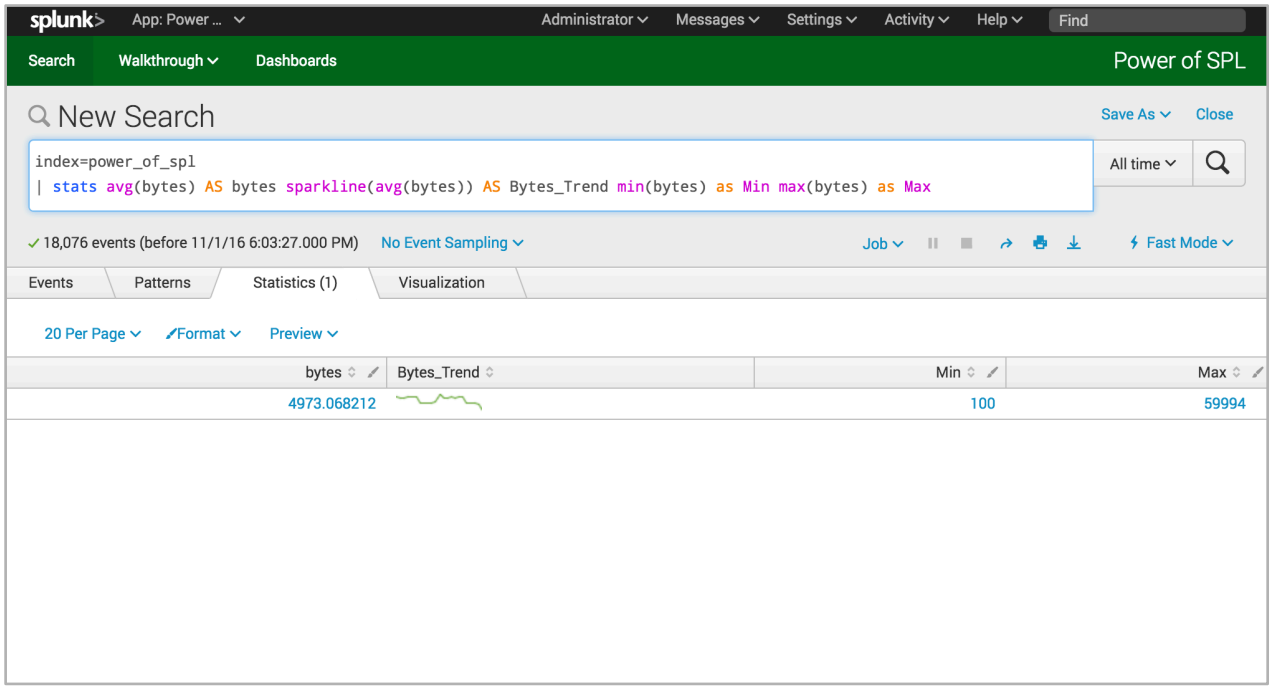
  | stats avg(bytes) AS “Avg Bytes”

- **Multiple statistics**

  index=power_of_spl | stats avg(bytes) AS bytes sparkline(avg(bytes)) AS Bytes_Trend min(bytes) max(bytes)

- By another field

  index=power_of_spl
  | stats avg(bytes) AS avg_bytes sparkline(avg(bytes)) AS Bytes_Trend min(bytes) max(bytes) **by** clientip | sort - avg_bytes



Stats – Calculate Statistics Based on Field Values

Examples

- Calculate stats and rename

  Index=power_of_spl

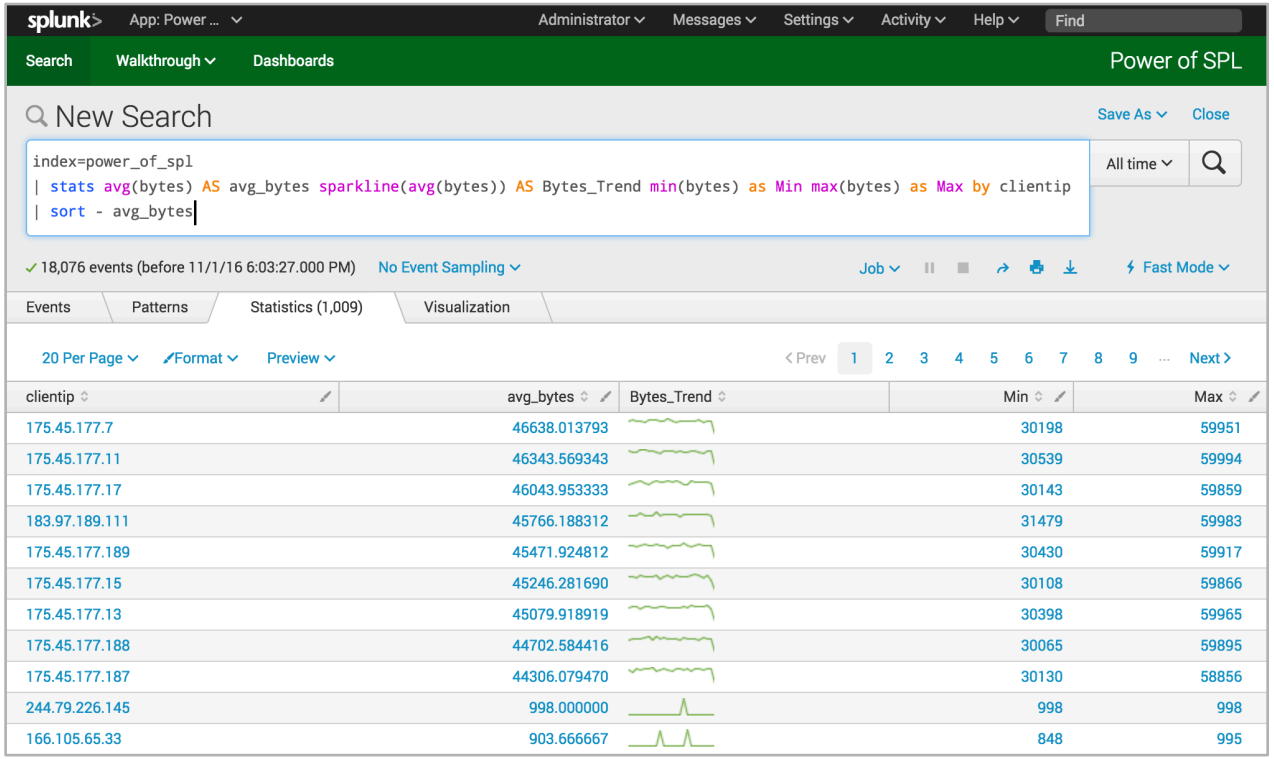
  | stats avg(bytes) AS “Avg Bytes”

- Multiple statistics

  index=power_of_spl | stats avg(bytes) AS bytes sparkline(avg(bytes)) AS Bytes_Trend min(bytes) max(bytes)

- **By another field**

  index=power_of_spl | stats avg(bytes) AS avg_bytes sparkline(avg(bytes)) AS Bytes_Trend min(bytes) max(bytes) by clientip | sort - avg_bytes



splunk> .conf2017

Timechart – Visualize Statistics Over Time

Examples

- **Visualize stats over time**

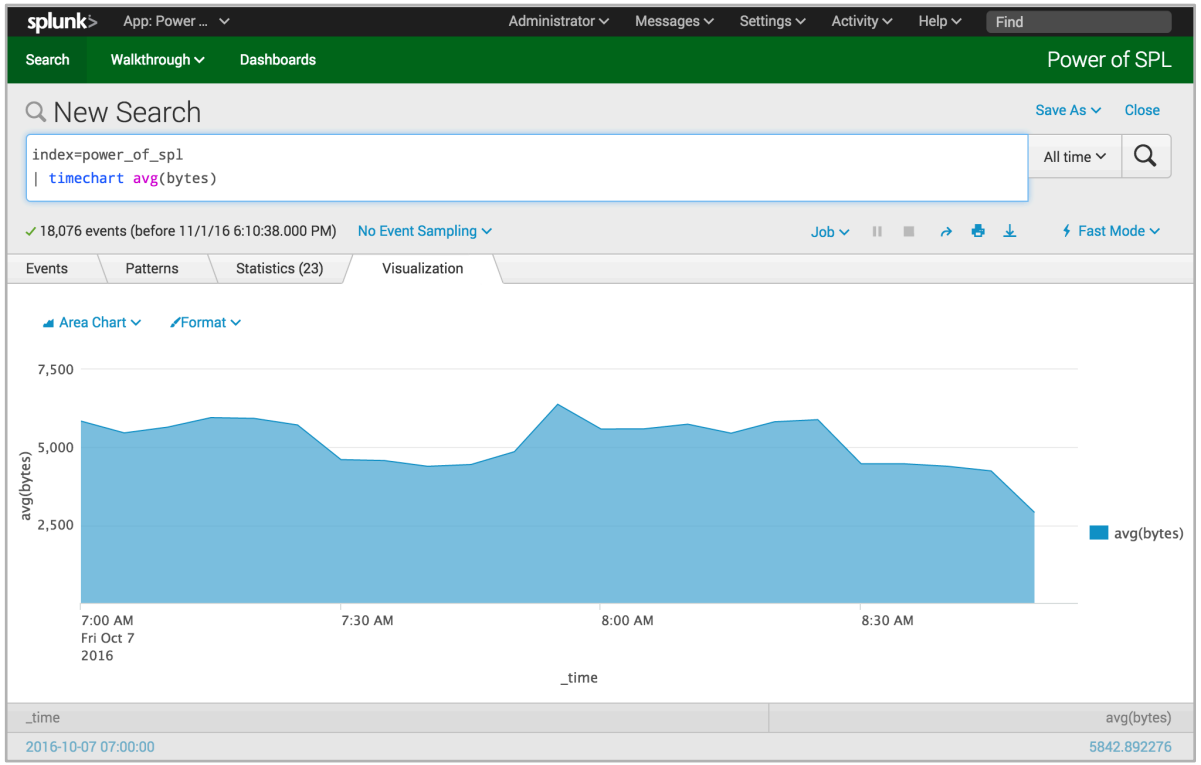
  index=power_of_spl
  | timechart avg(bytes)

- Add a trendline

  index=power_of_spl
  | timechart avg(bytes) as bytes
  | trendline sma5(bytes)

- Add a prediction overlay

  index=power_of_spl
  | timechart avg(bytes) as bytes
  | predict future_timespan=5 bytes



splunk> .conf2017

Timechart – Visualize Statistics Over Time

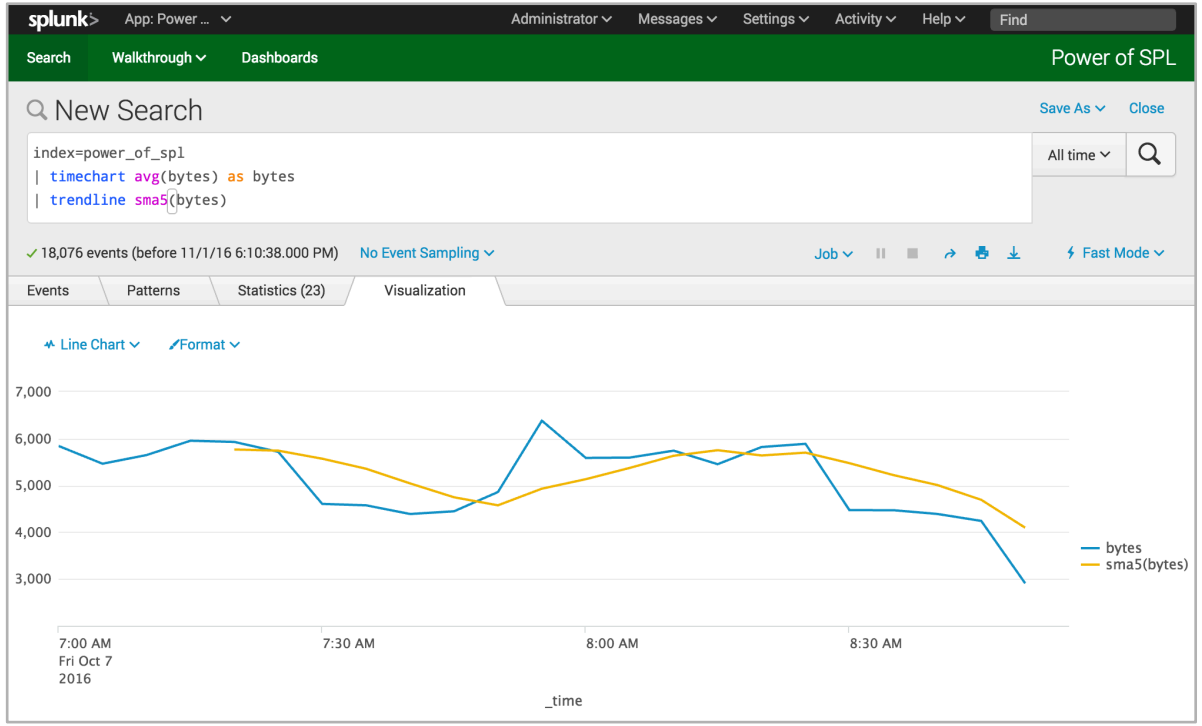
Examples

- Visualize stats over time

```
index=power_of_spl
| timechart avg(bytes)
```

- **Add a trendline**

```
index=power_of_spl
| timechart avg(bytes) as bytes
| trendline sma5(bytes)
```



- Add a prediction overlay

```
index=power_of_spl
| timechart avg(bytes) as bytes
| predict future_timespan=5 bytes
```

splunk> .conf2017

Timechart – Visualize Statistics Over Time

Examples

- Visualize stats over time

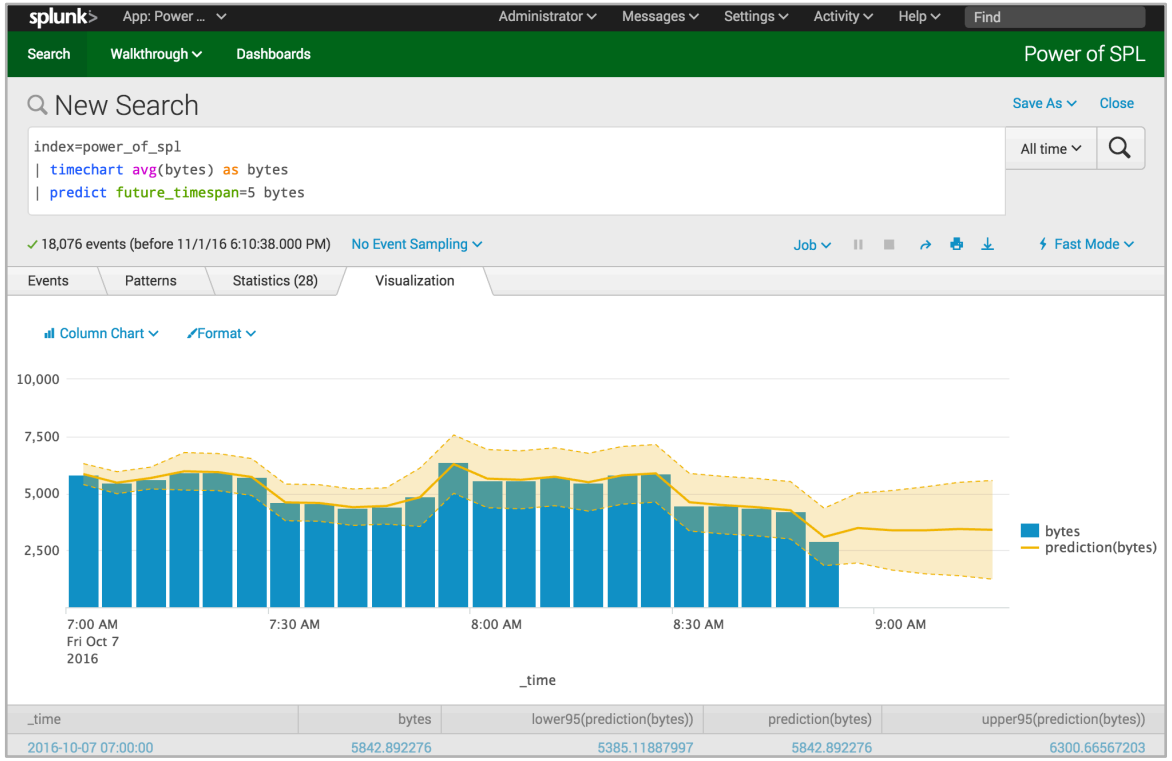
```
index=power_of_spl
| timechart avg(bytes)
```

- Add a trendline

```
index=power_of_spl
| timechart avg(bytes) as bytes
| trendline sma5(bytes)
```

- **Add a prediction overlay**

```
index=power_of_spl
| timechart avg(bytes) as bytes
| predict future_timespan=5 bytes
```



splunk> .conf2017

Streamstats – Cumulative/Running Totals Statistics

Examples

- **Cumulative/Running Totals**

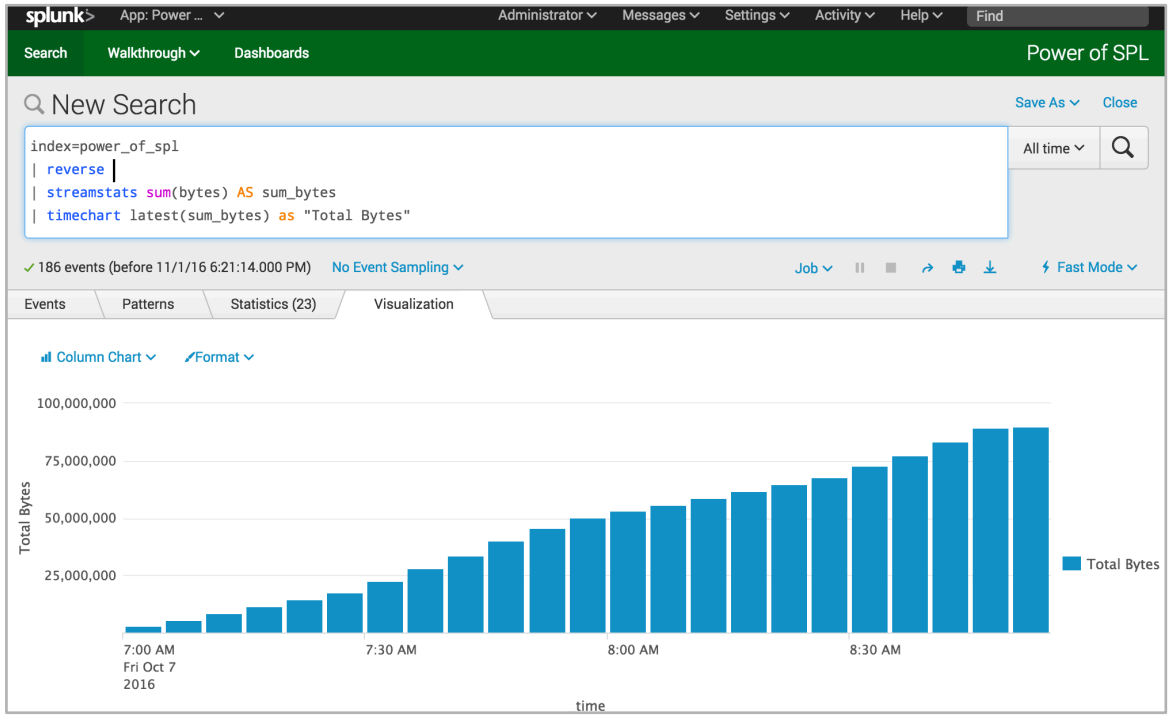
  index=power_of_spl
  | reverse
  | streamstats sum(bytes) AS sum_bytes
  | timechart latest(sum_bytes) as "Total Bytes"

- Summary Statistics

  index=power_of_spl

  | eventstats avg(bytes) AS overall_avg_bytes

  | stats avg(bytes) as clientip_avg_bytes by clientip overall_avg_bytes



splunk> .conf2017

Streamstats – Cumulative/Running Totals Statistics

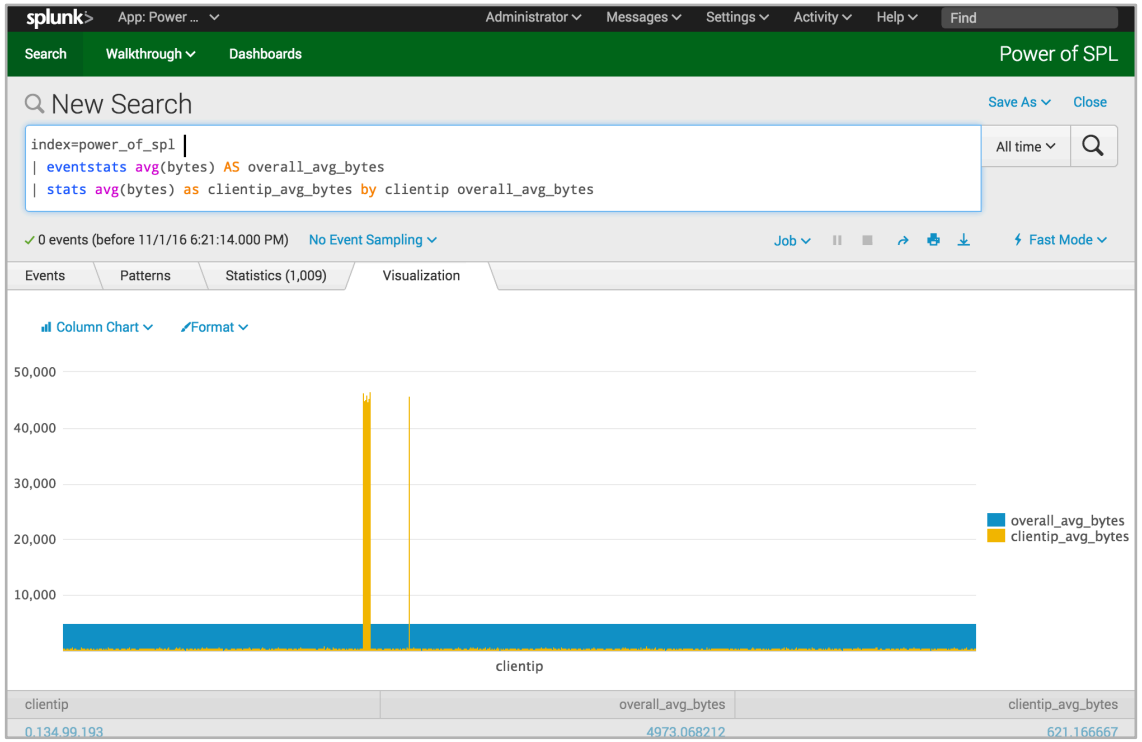
Examples

- Cumulative/Running Totals

```
index=power_of_spl
| reverse
| streamstats sum(bytes) AS sum_bytes
| timechart latest(sum_bytes) as "Total Bytes"
```

- **Summary Statistics**

```
index=power_of_spl
| eventstats avg(bytes) AS overall_avg_bytes
| stats avg(bytes) as clientip_avg_bytes by clientip overall_avg_bytes
```



splunk> .conf2017

Stats/Timechart – But Wait, There's More!

Splunk Search Quick Reference Guide

Common Stats Functions	Common statistical functions used with the chart, stats, and timechart commands. Field names can be wildcarded, so avg(*delay) might calculate the average of the delay and xdelay fields.
avg(X)	Returns the average of the values of field X.
count(X)	Returns the number of occurrences of the field X. To indicate a specific field value to match, format X as eval(field="value").
dc(X)	Returns the count of distinct values of the field X.
earliest(X)	Returns the chronologically earliest seen value of X.
latest(X)	Returns the chronologically latest seen value of X.
max(X)	Returns the maximum value of the field X. If the values of X are non-numeric, the max is found from alphabetical ordering.
median(X)	Returns the middle-most value of the field X.
min(X)	Returns the minimum value of the field X. If the values of X are non-numeric, the min is found from alphabetical ordering.
mode(X)	Returns the most frequent value of the field X.
perc<X>(Y)	Returns the X-th percentile value of the field Y. For example, perc5(total) returns the 5th percentile value of a field "total".
range(X)	Returns the difference between the max and min values of the field X.
stdev(X)	Returns the sample standard deviation of the field X.
stdevp(X)	Returns the population standard deviation of the field X.
sum(X)	Returns the sum of the values of the field X.
sumsq(X)	Returns the sum of the squares of the values of the field X.
values(X)	Returns the list of all distinct values of the field X as a multi-value entry. The order of the values is alphabetical.
var(X)	Returns the sample variance of the field X.

splunk> .conf2017

SPL Examples And Recipes

- Find the needle in the haystack
- Charting statistics and predicting values
- **Enriching and converging data sources**
- Map geographic data in real time
- Identifying anomalies
- Transactions
- Data exploration & finding relationships between fields
- Custom commands

Converging Data Sources

Index Untapped Data: Any Source, Type, Volume

splunk>

Ask Any Question

- Application Delivery
- IT Operations
- Security, Compliance and Fraud
- Business Analytics
- Industrial Data and the Internet of Things

splunk> .conf2017

Lookup – Converging Data Sources

Examples

- **Enrich data with lookups**

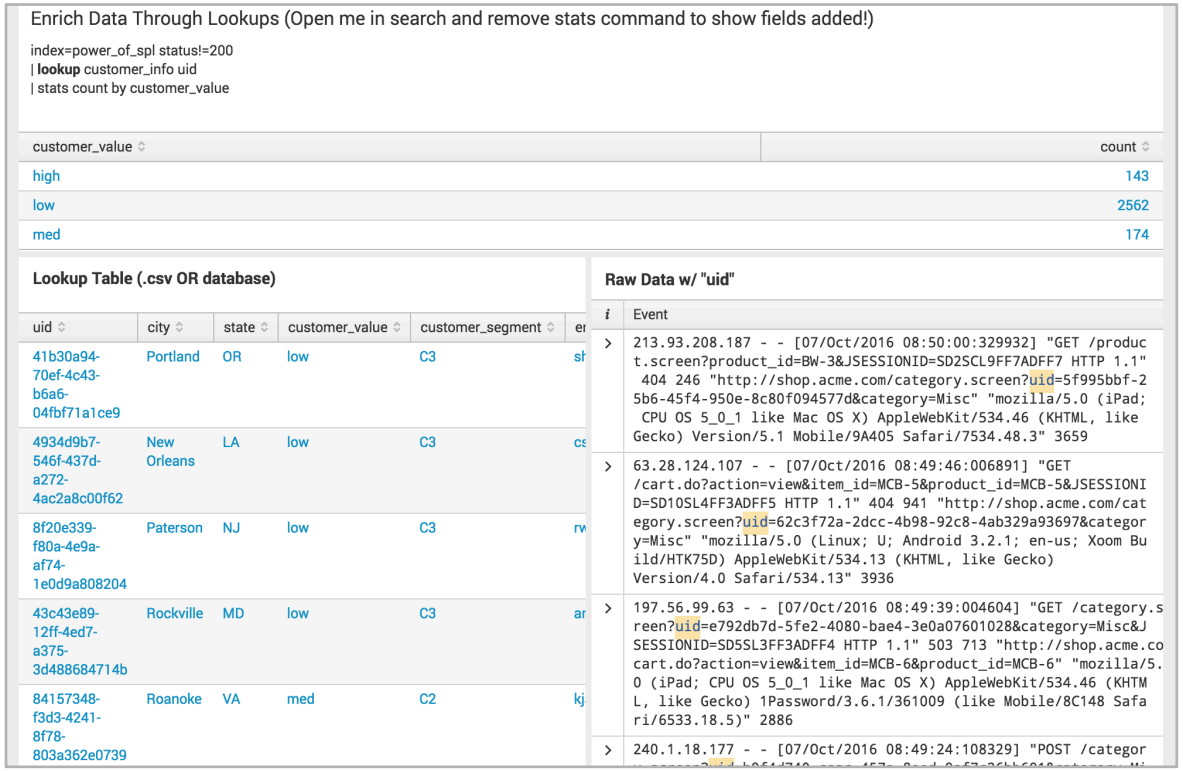
  index=power_of_spl status!=200
  | lookup customer_info uid
  | stats count by customer_value

- Search Inception!

  index=power_of_spl
  [search index=power_of_spl | stats sum(bytes) as total_bytes by clientip
  | sort - total_bytes | head 1 | return clientip]
  | stats count by clientip status uri | sort - count

- Append multiple searches

  index=power_of_spl
  | timechart span=15s avg(bytes) as avg_bytes
  | appendcols [search index=power_of_spl
  | stats stdev(bytes) as stdev_bytes] | eval 2stdv_upper = avg_bytes + stdev_bytes*2 | filldown 2stdv_upper | eval 2stdv_lower = avg_bytes - stdev_bytes*2 | filldown 2stdv_lower
  | eval 2stdv_lower = if('2stdv_lower' <0,0,'2stdv_lower') | fields - stdev_bytes



splunk> .conf2017

Converging Data Sources

Examples

- Enrich data with lookups

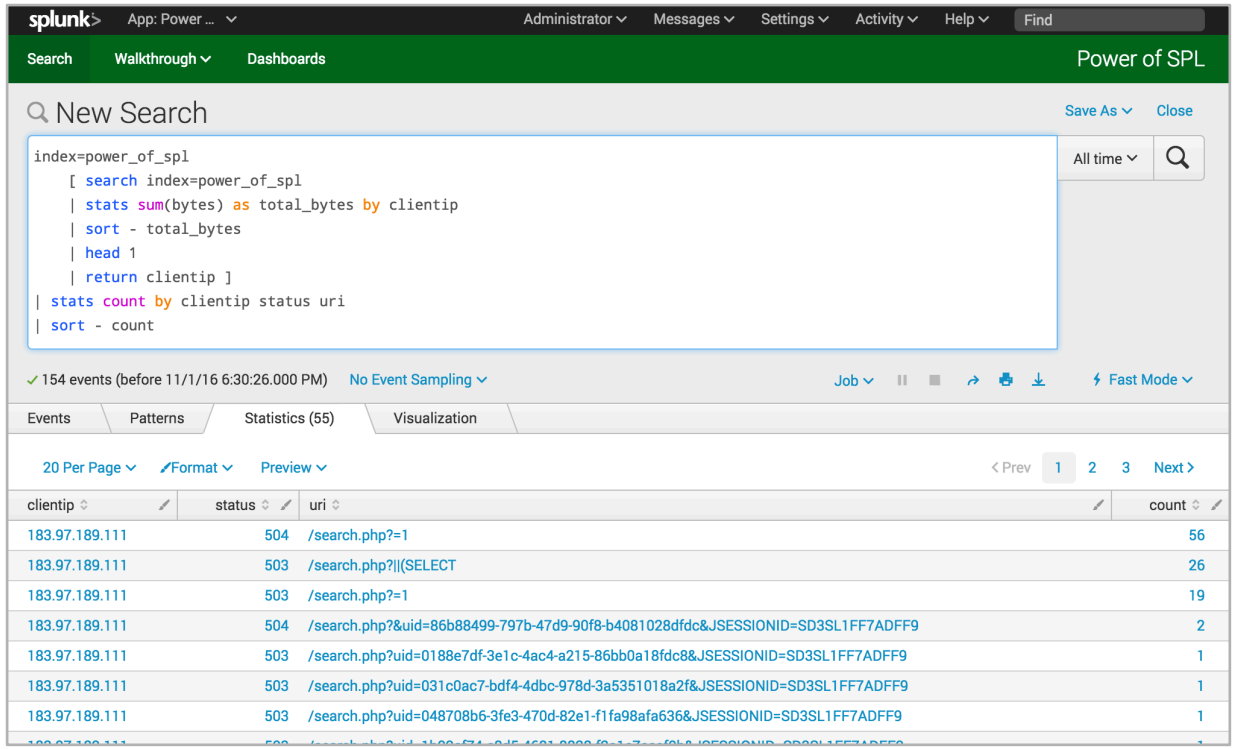
```
index=power_of_spl status!=200
| lookup customer_info uid
| stats count by customer_value
```

- **Search Inception!**

```
index=power_of_spl
[ search index=power_of_spl | stats sum(bytes)
as total_bytes by clientip
| sort - total_bytes | head 1  | return clientip ]
| stats count by clientip status uri | sort - count
```

- Append multiple searches

```
index=power_of_spl
| timechart span=15s avg(bytes) as avg_bytes
| appendcols [ search index=power_of_spl
| stats stdev(bytes) as stdev_bytes] | eval 2stdv_upper = avg_bytes +
stdev_bytes*2 | filldown 2stdv_upper | eval 2stdv_lower = avg_bytes -
stdev_bytes*2 | filldown 2stdv_lower
| eval 2stdv_lower = if('2stdv_lower' <0,0,'2stdv_lower') | fields -
stdev_bytes
```



splunk> .conf2017

appendcols - Converging Data Sources

Examples

- Enrich data with lookups

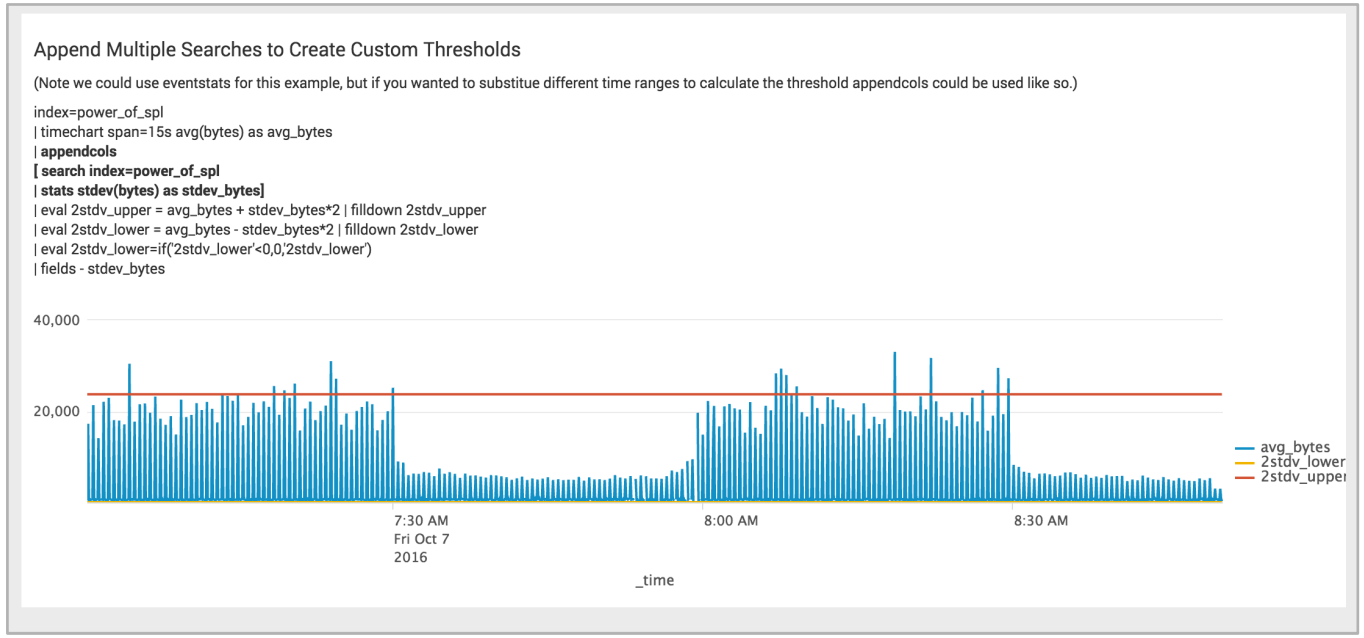
```
index=power_of_spl status!=200
| lookup customer_info uid
| stats count by customer_value
```

- Search Inception!

```
index=power_of_spl
[ search index=power_of_spl | stats sum(bytes)
as total_bytes by clientip
| sort - total_bytes | head 1  | return clientip ]
| stats count by clientip status uri | sort - count
```

- **Append multiple searches**

index=power_of_spl
| timechart span=15s avg(bytes) as avg_bytes
| appendcols [search index=power_of_spl
| stats stdev(bytes) as stdev_bytes] | eval 2stdv_upper = avg_bytes + stdev_bytes*2 | filldown 2stdv_upper | eval 2stdv_lower = avg_bytes - stdev_bytes*2 | filldown 2stdv_lower
| eval 2stdv_lower = if('2stdv_lower' <0,0,'2stdv_lower') | fields - stdev_bytes



splunk> .conf2017

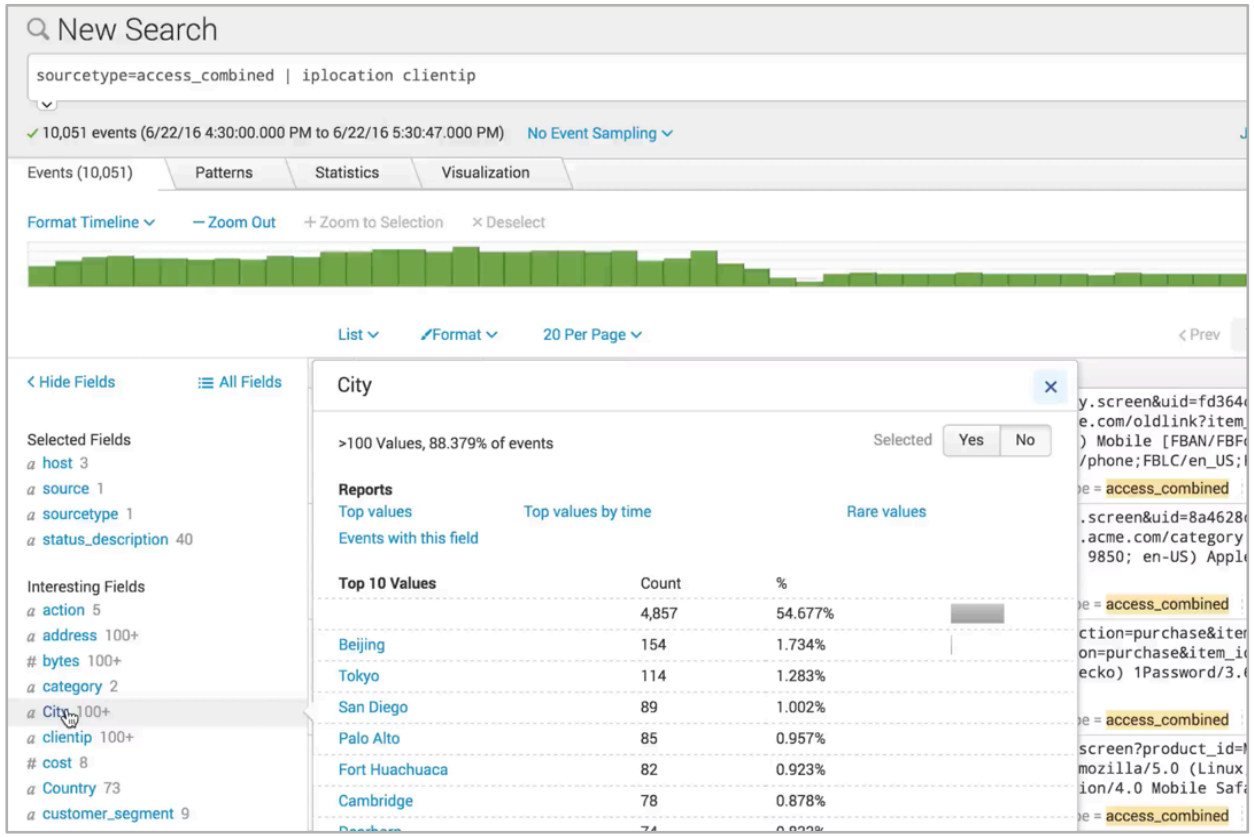
SPL Examples And Recipes

- Find the needle in the haystack
- Charting statistics and predicting values
- Enriching and converging data sources
- **Map geographic data in real time**
- Identifying anomalies
- Transactions
- Data exploration & finding relationships between fields
- Custom commands

iplocation – Geographic Data

Examples

- **Assign Lat/Lon to IP addresses**
  - **… | iplocation clientip**
- Visualize statistics geographically
  - … | geostats sum(price) by product
- Use custom choropleths
  - … | geom <featureCollection> <featureId>
- Track object movements
  - … | table _time latitude longitude vehicleId

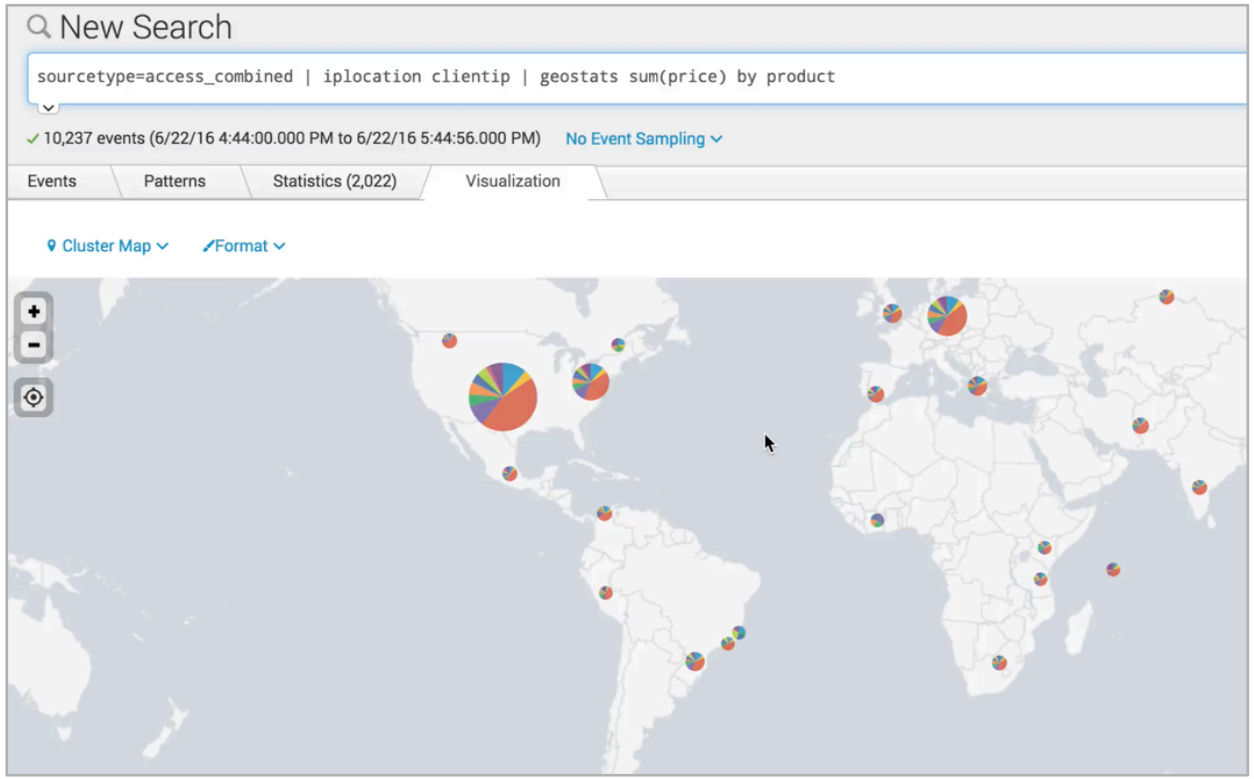


splunk> .conf2017

geostats – Geographic Data

Examples

- Assign Lat/Lon to IP addresses
  - … | iplocation clientip
- **Visualize statistics geographically**
  - **… | geostats sum(price) by product**
- Use custom choropleths
  - … | geom <featureCollection> <featureId>
- Track object movements
  - … | table _time latitude longitude vehicleId



geom – Geographic Data

Examples

- Assign Lat/Lon to IP addresses

  … | iplocation clientip

- Visualize statistics geographically

  … | geostats sum(price) by product

- **Use custom choropleths**

  … | geom <featureCollection> <featureId>

- Track object movements

  … | table _time latitude longitude vehicleId

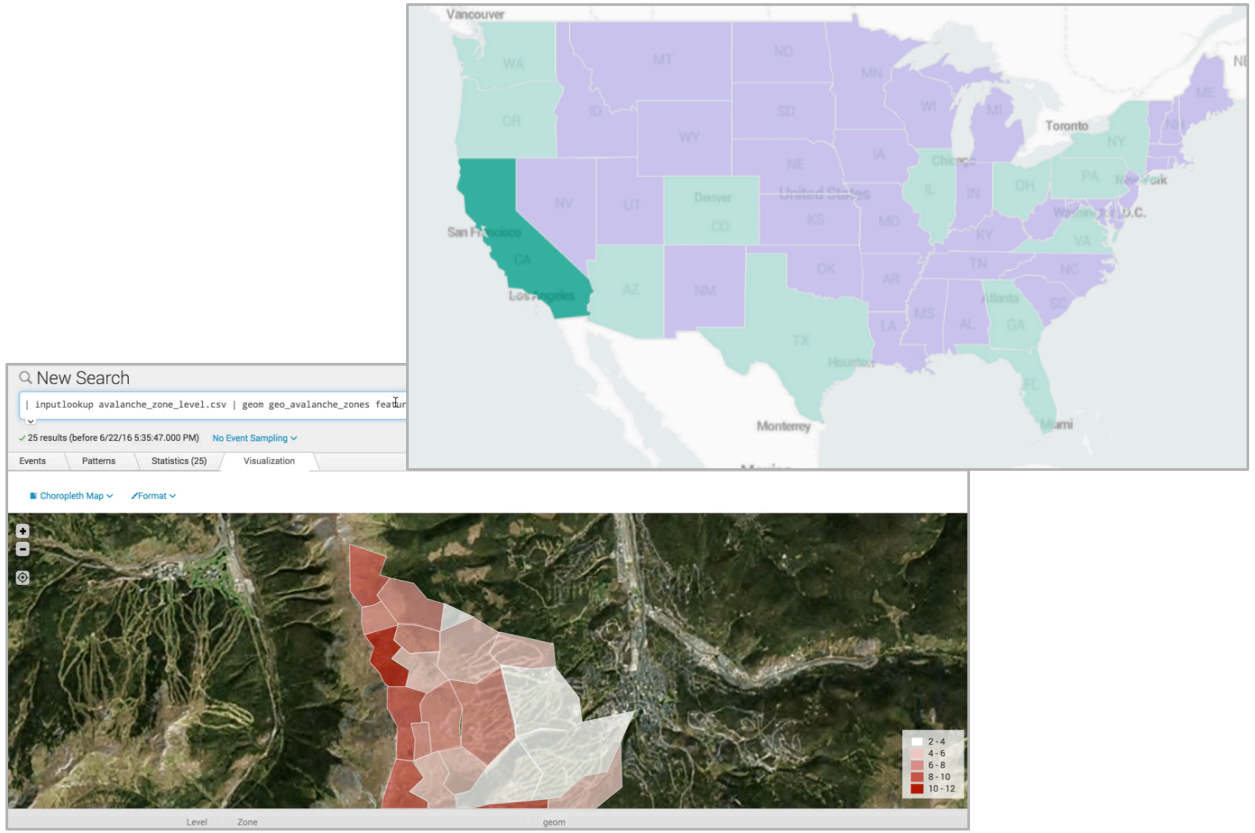
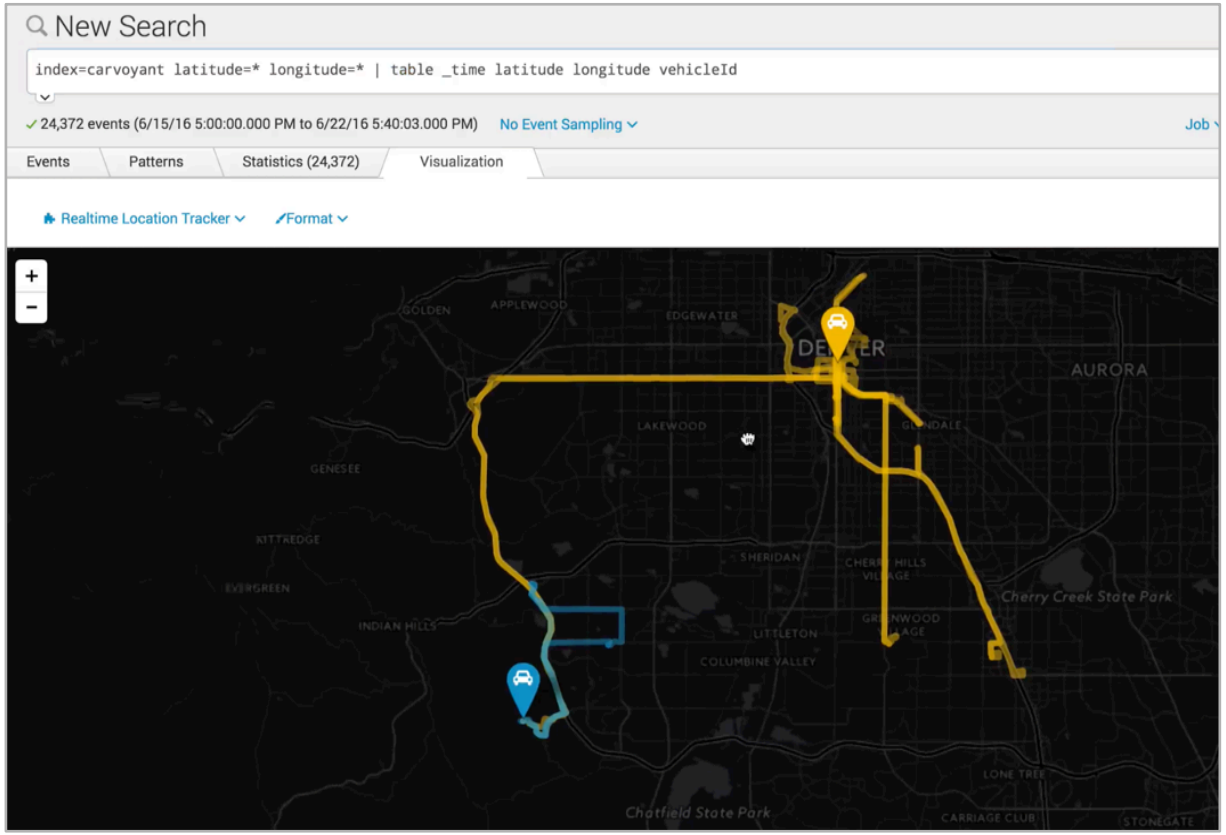


table – Geographic Data

Examples

- Assign Lat/Lon to IP addresses
  - … | iplocation clientip
- Visualize statistics geographically
  - … | geostats sum(price) by product
- Use custom choropleths
  - … | geom <featureCollection> <featureId>
- **Track object movements**
  - **… | table _time latitude longitude vehicleId**



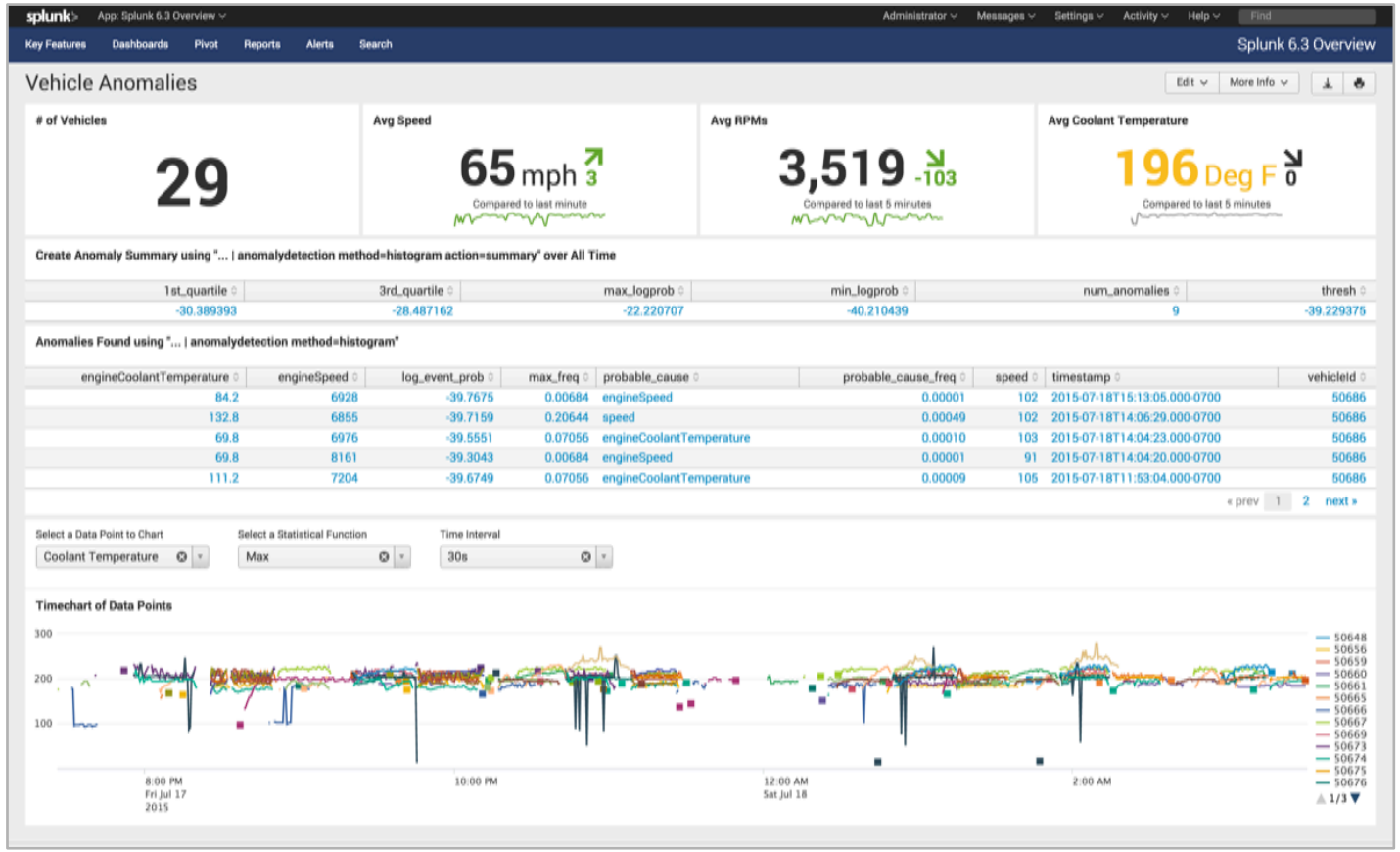
SPL Examples And Recipes

- Find the needle in the haystack
- Charting statistics and predicting values
- Enriching and converging data sources
- Map geographic data in real time
- **Identifying anomalies**
- Transactions
- Data exploration & finding relationships between fields
- Custom commands

Anomaly Detection – Find anomalies in your data

Examples

- Find anomalies
  | inputlookup car_data.csv | anomalydetection
- Summarize anomalies
  | inputlookup car_data.csv | anomalydetection action=summary
- Use IQR and remove outliers
  | inputlookup car_data.csv | anomalydetection method=iqr action=remove



splunk> .conf2017

SPL Examples And Recipes

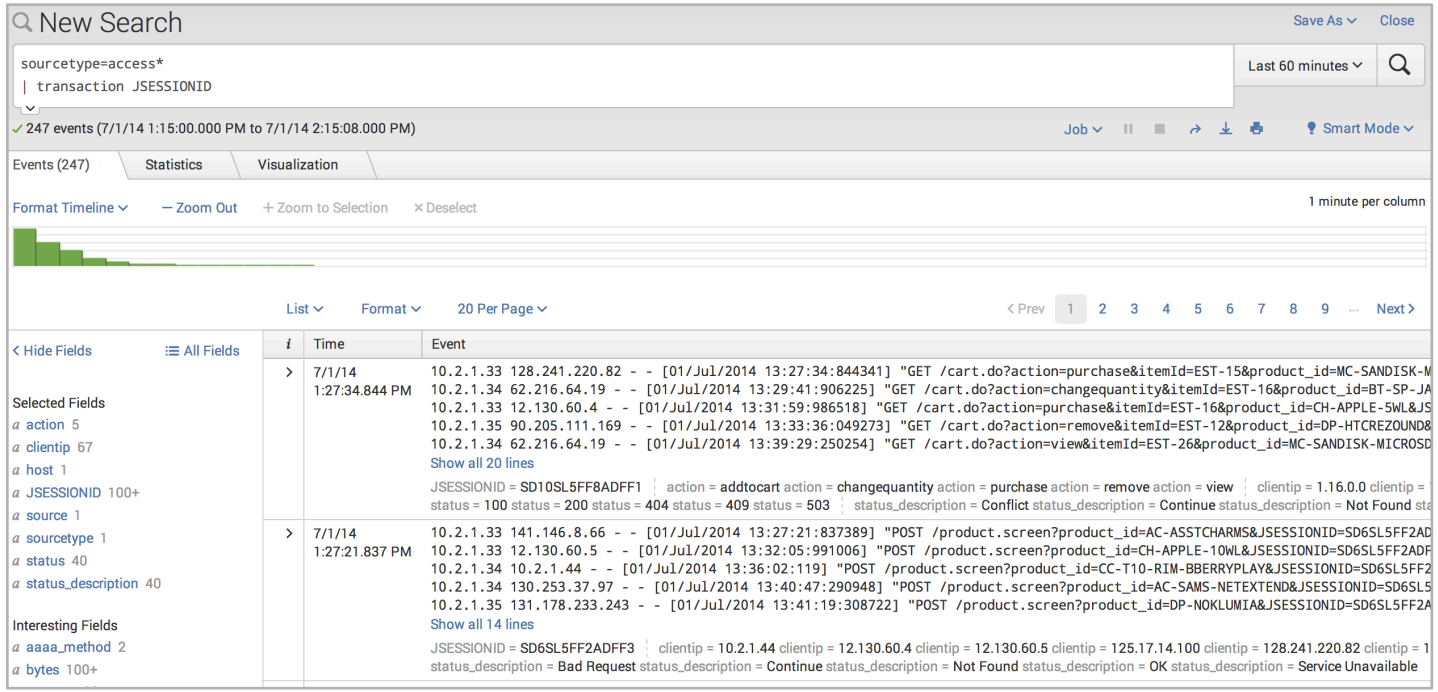
- Find the needle in the haystack
- Charting statistics and predicting values
- Enriching and converging data sources
- Map geographic data in real time
- Identifying anomalies
- **Transactions**
- Data exploration & finding relationships between fields
- Custom commands

splunk> .conf2017

Transaction – Group Related Events Spanning Time

Examples

- **Group by session ID**
  sourcetype=access*
  | transaction JSESSIONID
- Calculate session durations
  sourcetype=access*
  | transaction JSESSIONID
  | stats min(duration) max(duration) avg(duration)
- Stats is better
  sourcetype=access*
  | stats min(_time) AS earliest max(_time) AS latest by JSESSIONID
  | eval duration=latest-earliest
  | stats min(duration) max(duration) avg(duration)

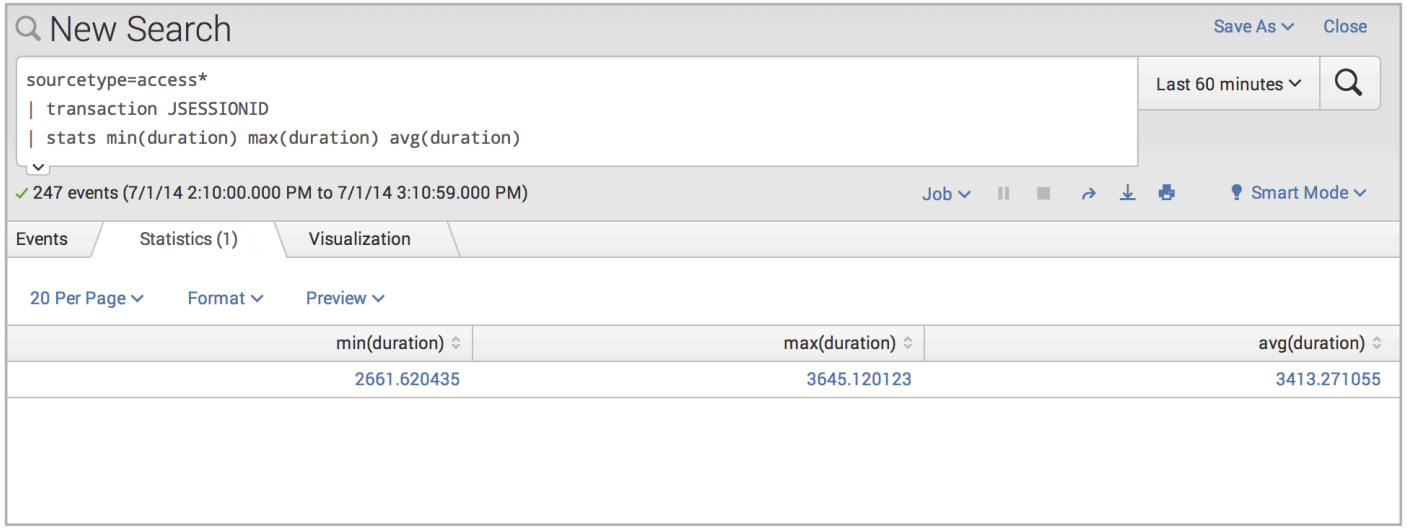


splunk> .conf2017

Transaction – Group Related Events Spanning Time

Examples

- Group by session ID
  sourcetype=access*
  | transaction JSESSIONID
- **Calculate session durations**
  sourcetype=access*
  | transaction JSESSIONID
  | stats min(duration) max(duration) avg(duration)
- Stats is better
  sourcetype=access*
  | stats min(_time) AS earliest max(_time) AS latest by JSESSIONID
  | eval duration=latest-earliest
  | stats min(duration) max(duration) avg(duration)

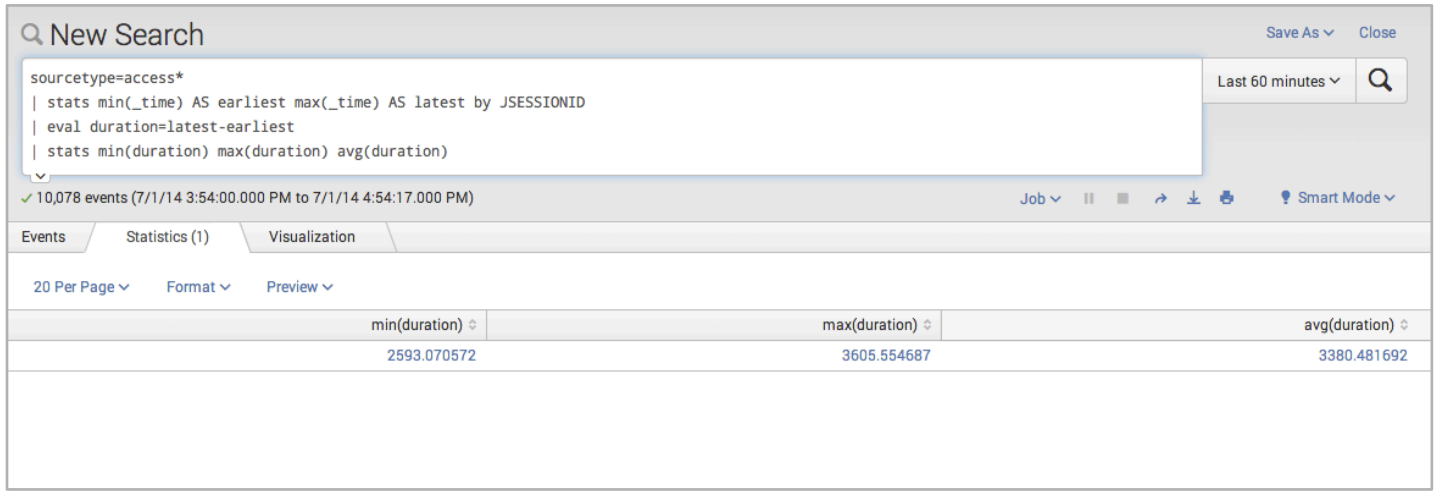


splunk> .conf2017

Transaction – Group Related Events Spanning Time

Examples

- Group by session ID
  sourcetype=access*
  | transaction JSESSIONID
- Calculate session durations
  sourcetype=access*
  | transaction JSESSIONID
  | stats min(duration) max(duration) avg(duration)
- **Stats is better**
  sourcetype=access*
  | stats min(_time) AS earliest max(_time) AS latest by JSESSIONID
  | eval duration=latest-earliest
  | stats min(duration) max(duration) avg(duration)



splunk> .conf2017

SPL Examples And Recipes

- Find the needle in the haystack
- Charting statistics and predicting values
- Enriching and converging data sources
- Map geographic data in real time
- Identifying anomalies
- Transactions
- **Data exploration & finding relationships between fields**
- Custom commands

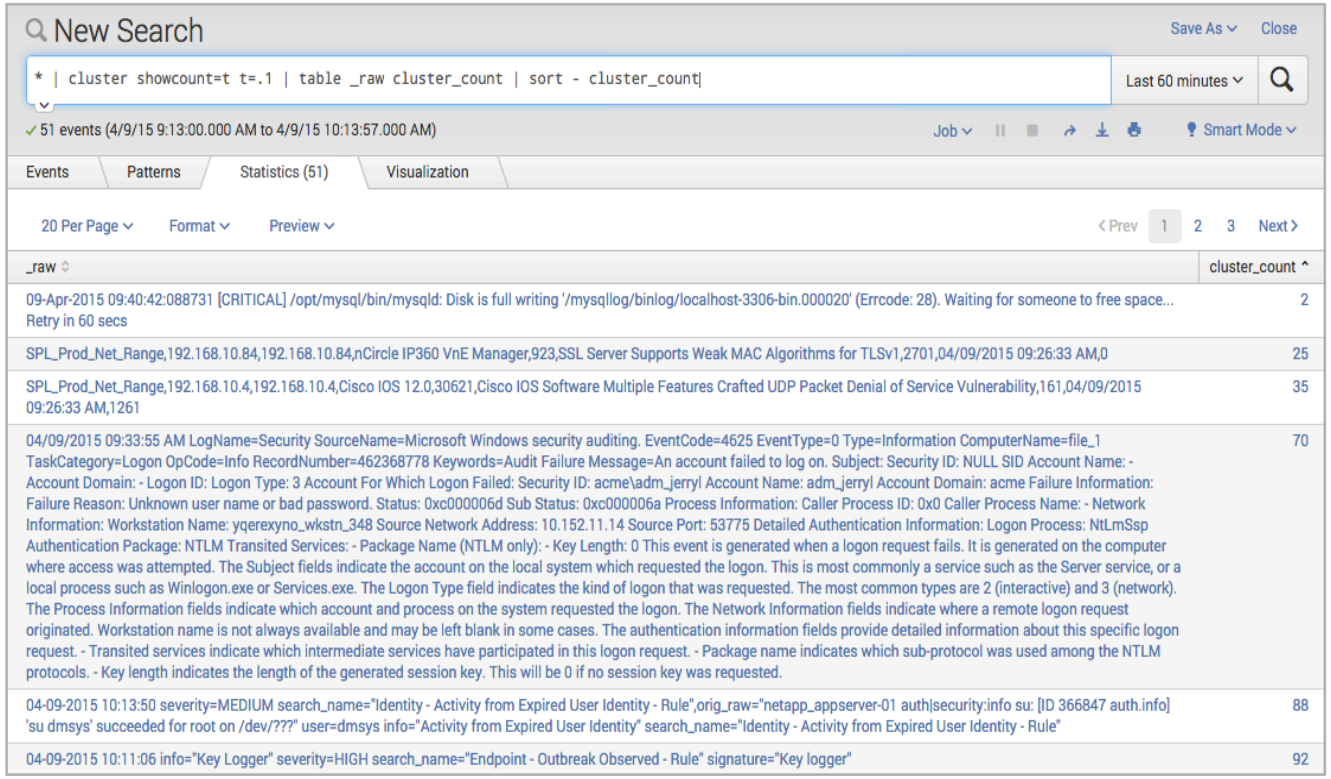
Data Exploration

- | analyzefields
- | anomalies
- | arules
- | associate
- | cluster
- | contingency
- | correlate
- | fieldsummary

Cluster – Exploring Your Data

Examples

- **Find most/least common events**
  *** | cluster showcount=t t=.1**
  | table _raw cluster_count
- Display Summary of Fields
  sourcetype=access_combined
  | fields – date* source* time*
  | fieldsummary maxvals=5
- Show patterns of co-occurring fields
  sourcetype=access_combined
  | fields – date* source* time* | correlate
- View field relationships
  sourcetype=access_combined
  | contingency uri status
- Find predictors of fields
  sourcetype=access_combined
  | analyzefields classfield=status



splunk> .conf2017

fieldsummary – Exploring Your Data

Examples

- Find most/least common events

  * | cluster showcount=t t=.1
  | table _raw cluster_count

- **Display Summary of Fields**

  sourcetype=access_combined
  | fields – date* source* time*
  | fieldsummary maxvals=5

- Show patterns of co-occurring fields

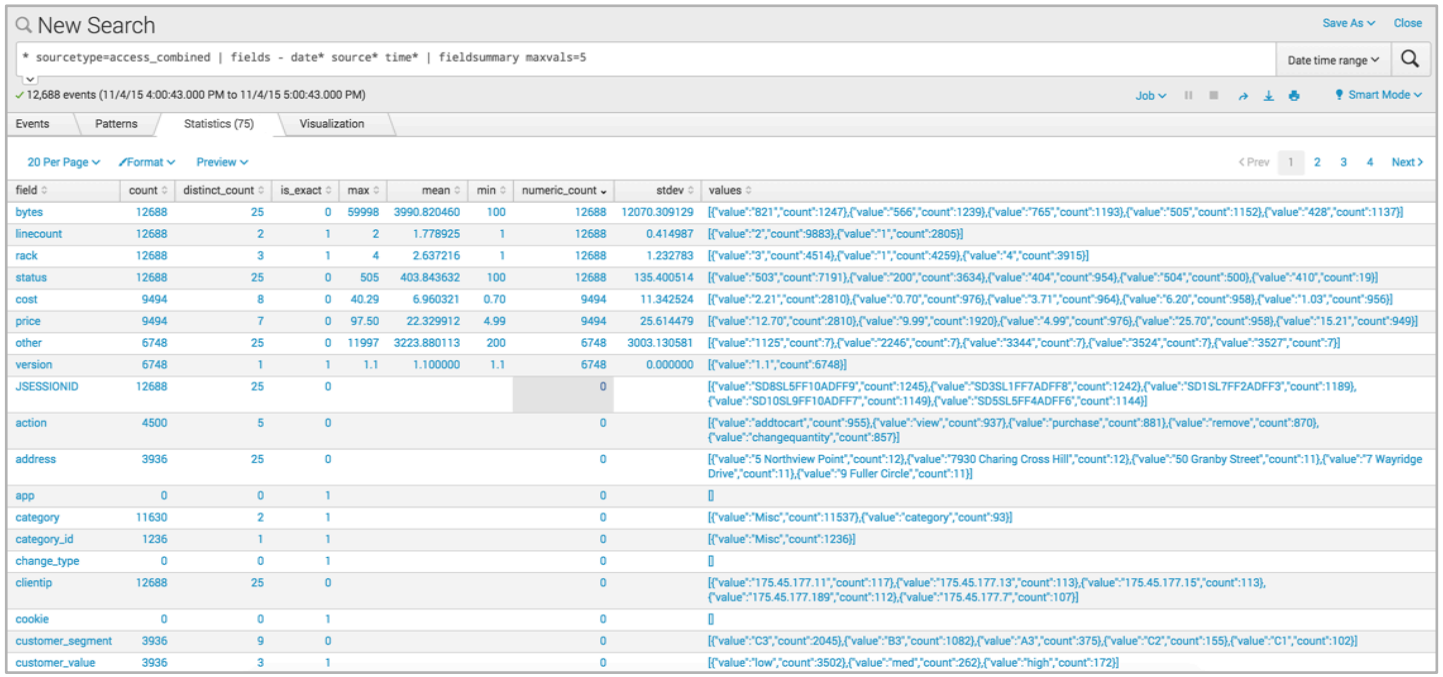
  sourcetype=access_combined
  | fields – date* source* time* | correlate

- View field relationships

  sourcetype=access_combined
  | contingency uri status

- Find predictors of fields

  sourcetype=access_combined
  | analyzefields classfield=status



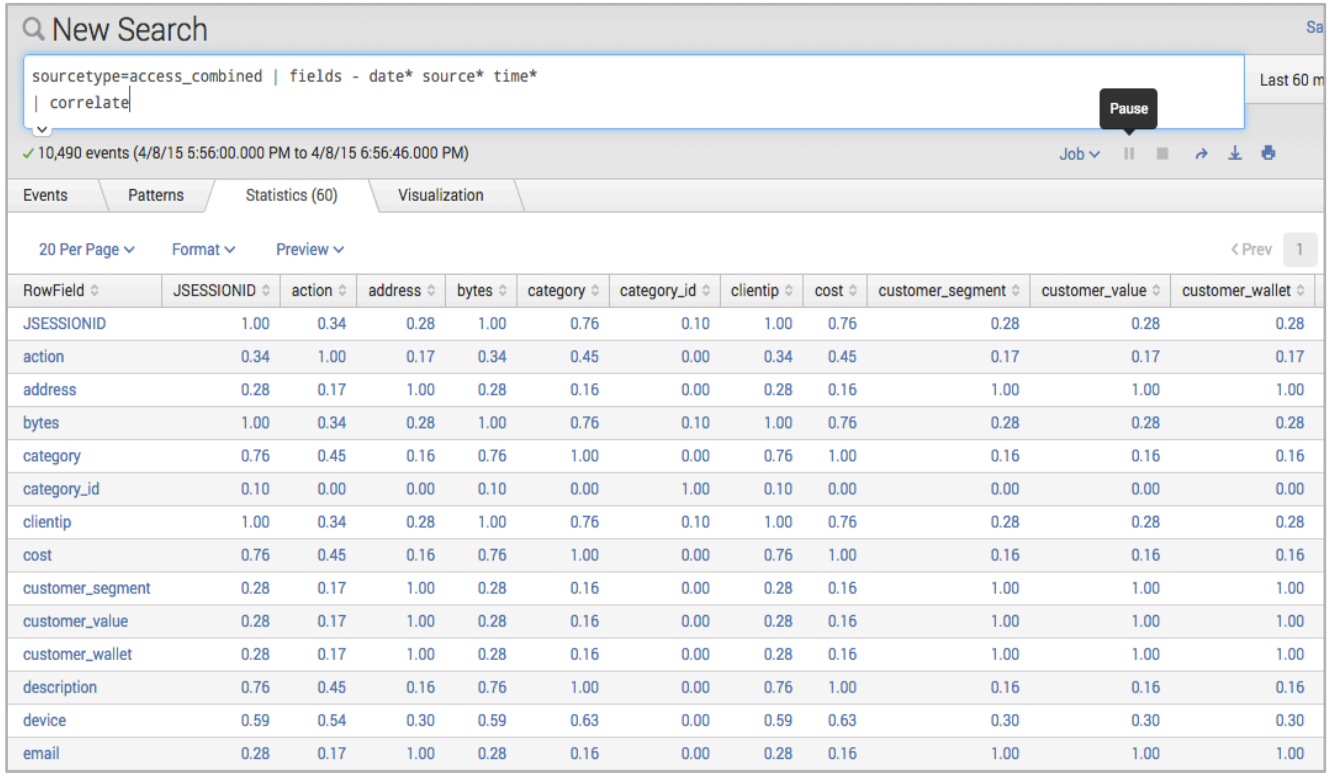
field	count	distinct_count	is_exact	max	mean	min	numeric_count	stdev	values
bytes	12688	25	0	59998	3990.820460	100	12688	12070.309129	[{"value":"821","count":1247},{"value":"566","count":1239},{"value":"765","count":1193},{"value":"505","count":1152},{"value":"428","count":1137}]
linecount	12688	2	1	2	1.778925	1	12688	0.414987	[{"value":"2","count":9883},{"value":"1","count":2805}]
rack	12688	3	1	4	2.637216	1	12688	1.232783	[{"value":"3","count":4514},{"value":"1","count":4259},{"value":"4","count":3915}]
status	12688	25	0	505	403.843632	100	12688	135.400514	[{"value":"503","count":7191},{"value":"200","count":3634},{"value":"404","count":954},{"value":"504","count":500},{"value":"410","count":19}]
cost	9494	8	0	40.29	6.960321	0.70	9494	11.342524	[{"value":"2.21","count":2810},{"value":"0.70","count":976},{"value":"3.71","count":964},{"value":"6.20","count":958},{"value":"1.03","count":956}]
price	9494	7	0	97.50	22.329912	4.99	9494	25.614479	[{"value":"12.70","count":2810},{"value":"9.99","count":1920},{"value":"4.99","count":976},{"value":"25.70","count":958},{"value":"15.21","count":949}]
other	6748	25	0	11997	3223.880113	200	6748	3003.130581	[{"value":"1125","count":7},{"value":"2246","count":7},{"value":"3344","count":7},{"value":"3524","count":7},{"value":"3527","count":7}]
version	6748	1	1	1.1	1.100000	1.1	6748	0.000000	[{"value":"1.1","count":6748}]
JSESSIONID	12688	25	0				0		[{"value":"SD8SL5FF10ADFF9","count":1245},{"value":"SD3SL1FF7ADFF8","count":1242},{"value":"SD1SL7FF2ADFF3","count":1189},{"value":"SD10SL9FF10ADFF7","count":1149},{"value":"SD5SL5FF4ADFF6","count":1144}]
action	4500	5	0				0		[{"value":"addtocart","count":955},{"value":"view","count":937},{"value":"purchase","count":881},{"value":"remove","count":870},{"value":"changequantity","count":857}]
address	3936	25	0				0		[{"value":"5 Northview Point","count":12},{"value":"7930 Charing Cross Hill","count":12},{"value":"50 Granby Street","count":11},{"value":"7 Wayridge Drive","count":11},{"value":"9 Fuller Circle","count":11}]
app	0	0	1				0		[]
category	11630	2	1				0		[{"value":"Misc","count":11537},{"value":"category","count":93}]
category_id	1236	1	1				0		[{"value":"Misc","count":1236}]
change_type	0	0	1				0		[]
clientip	12688	25	0				0		[{"value":"175.45.177.11","count":117},{"value":"175.45.177.13","count":113},{"value":"175.45.177.15","count":113},{"value":"175.45.177.189","count":112},{"value":"175.45.177.7","count":107}]
cookie	0	0	1				0		[]
customer_segment	3936	9	0				0		[{"value":"C3","count":2045},{"value":"B3","count":1082},{"value":"A3","count":375},{"value":"C2","count":155},{"value":"C1","count":102}]
customer_value	3936	3	1				0		[{"value":"low","count":3502},{"value":"med","count":262},{"value":"high","count":172}]

splunk> .conf2017

Correlate – Exploring Your Data

Examples

- Find most/least common events
  * | cluster showcount=t t=.1
  | table _raw cluster_count
- Display Summary of Fields
  sourcetype=access_combined
  | fields – date* source* time*
  | fieldsummary maxvals=5
- **Show patterns of co-occurring fields**
  sourcetype=access_combined
  | fields – date* source* time* | correlate
- View field relationships
  sourcetype=access_combined
  | contingency uri status
- Find predictors of fields
  sourcetype=access_combined
  | analyzefields classfield=status



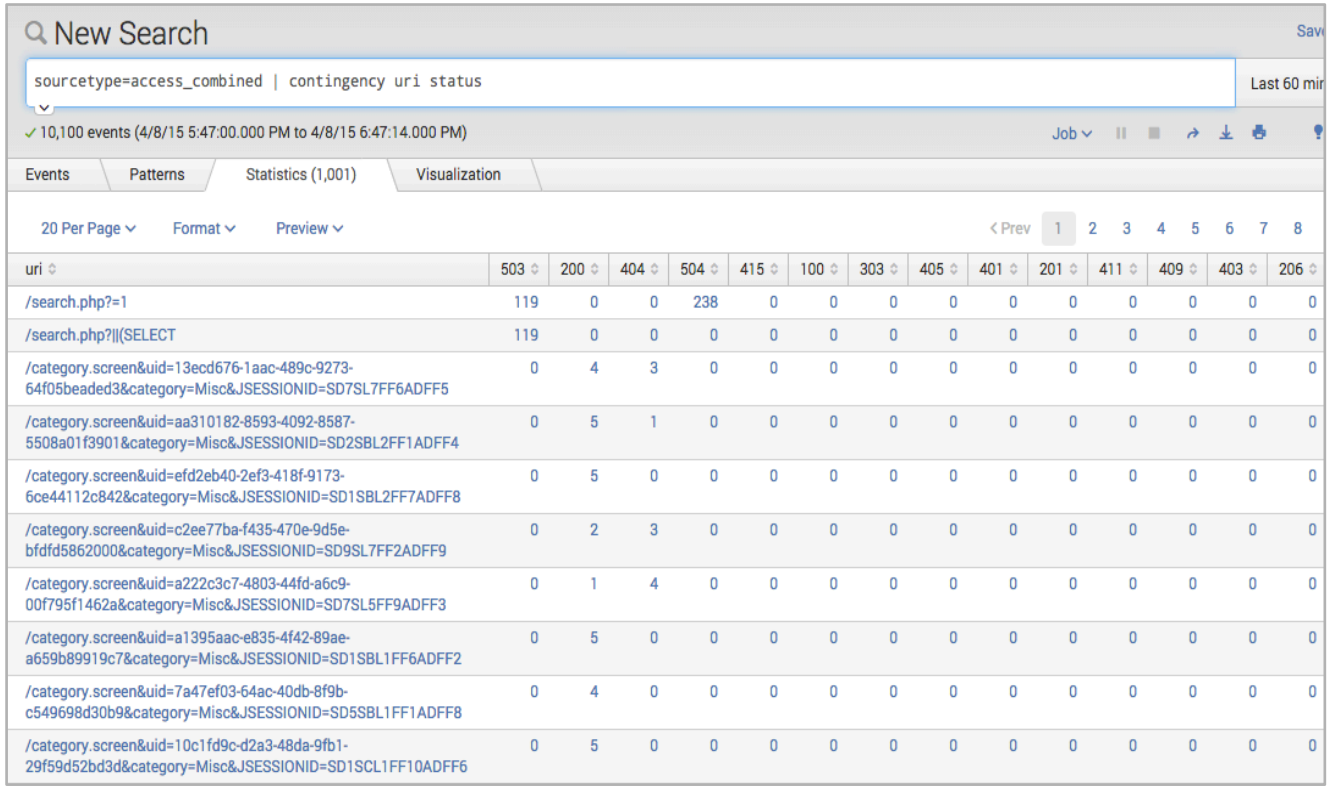
RowField	JSESSIONID	action	address	bytes	category	category_id	clientip	cost	customer_segment	customer_value	customer_wallet
JSESSIONID	1.00	0.34	0.28	1.00	0.76	0.10	1.00	0.76	0.28	0.28	0.28
action	0.34	1.00	0.17	0.34	0.45	0.00	0.34	0.45	0.17	0.17	0.17
address	0.28	0.17	1.00	0.28	0.16	0.00	0.28	0.16	1.00	1.00	1.00
bytes	1.00	0.34	0.28	1.00	0.76	0.10	1.00	0.76	0.28	0.28	0.28
category	0.76	0.45	0.16	0.76	1.00	0.00	0.76	1.00	0.16	0.16	0.16
category_id	0.10	0.00	0.00	0.10	0.00	1.00	0.10	0.00	0.00	0.00	0.00
clientip	1.00	0.34	0.28	1.00	0.76	0.10	1.00	0.76	0.28	0.28	0.28
cost	0.76	0.45	0.16	0.76	1.00	0.00	0.76	1.00	0.16	0.16	0.16
customer_segment	0.28	0.17	1.00	0.28	0.16	0.00	0.28	0.16	1.00	1.00	1.00
customer_value	0.28	0.17	1.00	0.28	0.16	0.00	0.28	0.16	1.00	1.00	1.00
customer_wallet	0.28	0.17	1.00	0.28	0.16	0.00	0.28	0.16	1.00	1.00	1.00
description	0.76	0.45	0.16	0.76	1.00	0.00	0.76	1.00	0.16	0.16	0.16
device	0.59	0.54	0.30	0.59	0.63	0.00	0.59	0.63	0.30	0.30	0.30
email	0.28	0.17	1.00	0.28	0.16	0.00	0.28	0.16	1.00	1.00	1.00

splunk> .conf2017

Contingency – Exploring Your Data

Examples

- Find most/least common events
  * | cluster showcount=t t=.1
  | table _raw cluster_count
- Display Summary of Fields
  sourcetype=access_combined
  | fields – date* source* time*
  | fieldsummary maxvals=5
- Show patterns of co-occurring fields
  sourcetype=access_combined
  | fields – date* source* time* | correlate
- **View field relationships**
  sourcetype=access_combined
  | contingency uri status
- Find predictors of fields
  sourcetype=access_combined
  | analyzefields classfield=status



uri	503	200	404	504	415	100	303	405	401	201	411	409	403	206
/search.php?=1	119	0	0	238	0	0	0	0	0	0	0	0	0	0
/search.php?|(SELECT	119	0	0	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=13ecd676-1aac-489c-9273-64f05beaded3&category=Misc&JSESSIONID=SD7SL7FF6ADFF5	0	4	3	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=aa310182-8593-4092-8587-5508a01f3901&category=Misc&JSESSIONID=SD2SBL2FF1ADFF4	0	5	1	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=efd2eb40-2ef3-418f-9173-6ce44112c842&category=Misc&JSESSIONID=SD1SBL2FF7ADFF8	0	5	0	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=c2ee77ba-f435-470e-9d5e-bfdfd5862000&category=Misc&JSESSIONID=SD9SL7FF2ADFF9	0	2	3	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=a222c3c7-4803-44fd-a6c9-00f795f1462a&category=Misc&JSESSIONID=SD7SL5FF9ADFF3	0	1	4	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=a1395aac-e835-4f42-89ae-a659b89919c7&category=Misc&JSESSIONID=SD1SBL1FF6ADFF2	0	5	0	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=7a47ef03-64ac-40db-8f9b-c549698d30b9&category=Misc&JSESSIONID=SD5SBL1FF1ADFF8	0	4	0	0	0	0	0	0	0	0	0	0	0	0
/category.screen&uid=10c1fd9c-d2a3-48da-9fb1-29f59d52bd3d&category=Misc&JSESSIONID=SD1SCL1FF10ADFF6	0	5	0	0	0	0	0	0	0	0	0	0	0	0

splunk> .conf2017

analyzefields – Exploring Your Data

Examples

- Find most/least common events

  * | cluster showcount=t t=.1
  | table _raw cluster_count

- Display Summary of Fields

  sourcetype=access_combined
  | fields – date* source* time*
  | fieldsummary maxvals=5

- Show patterns of co-occurring fields

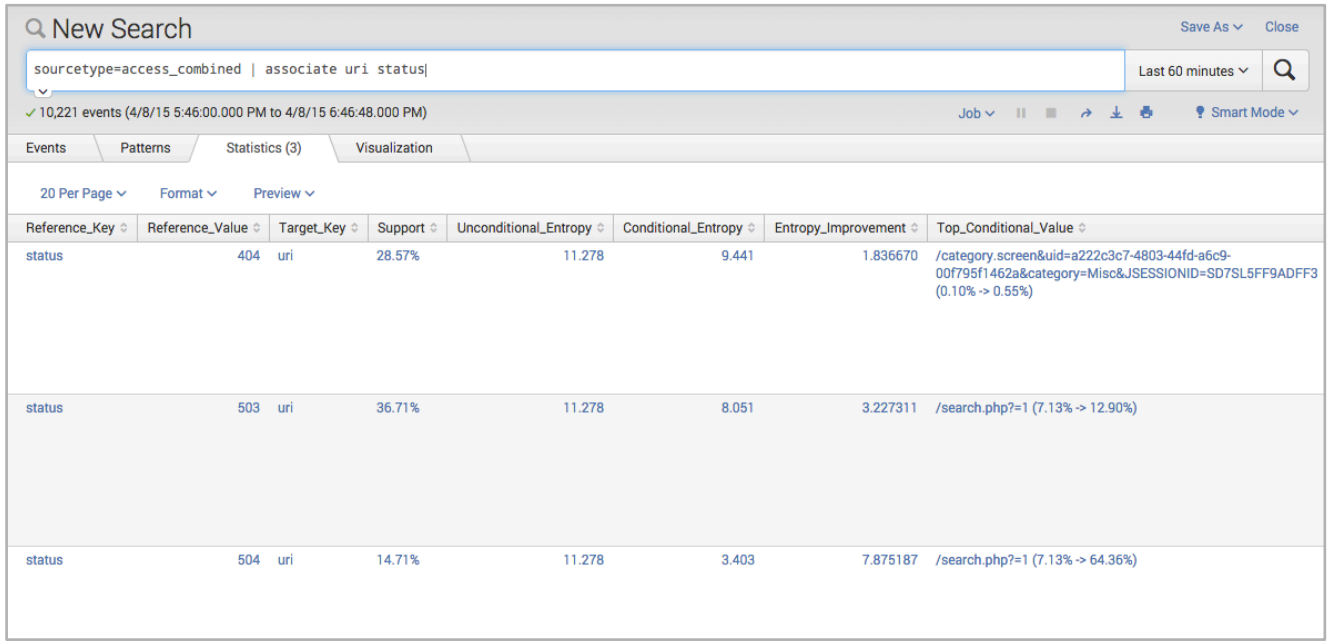
  sourcetype=access_combined
  | fields – date* source* time* | correlate

- View field relationships

  sourcetype=access_combined
  | contingency uri status

- **Find predictors of fields**

  sourcetype=access_combined
  | analyzefields classfield=status

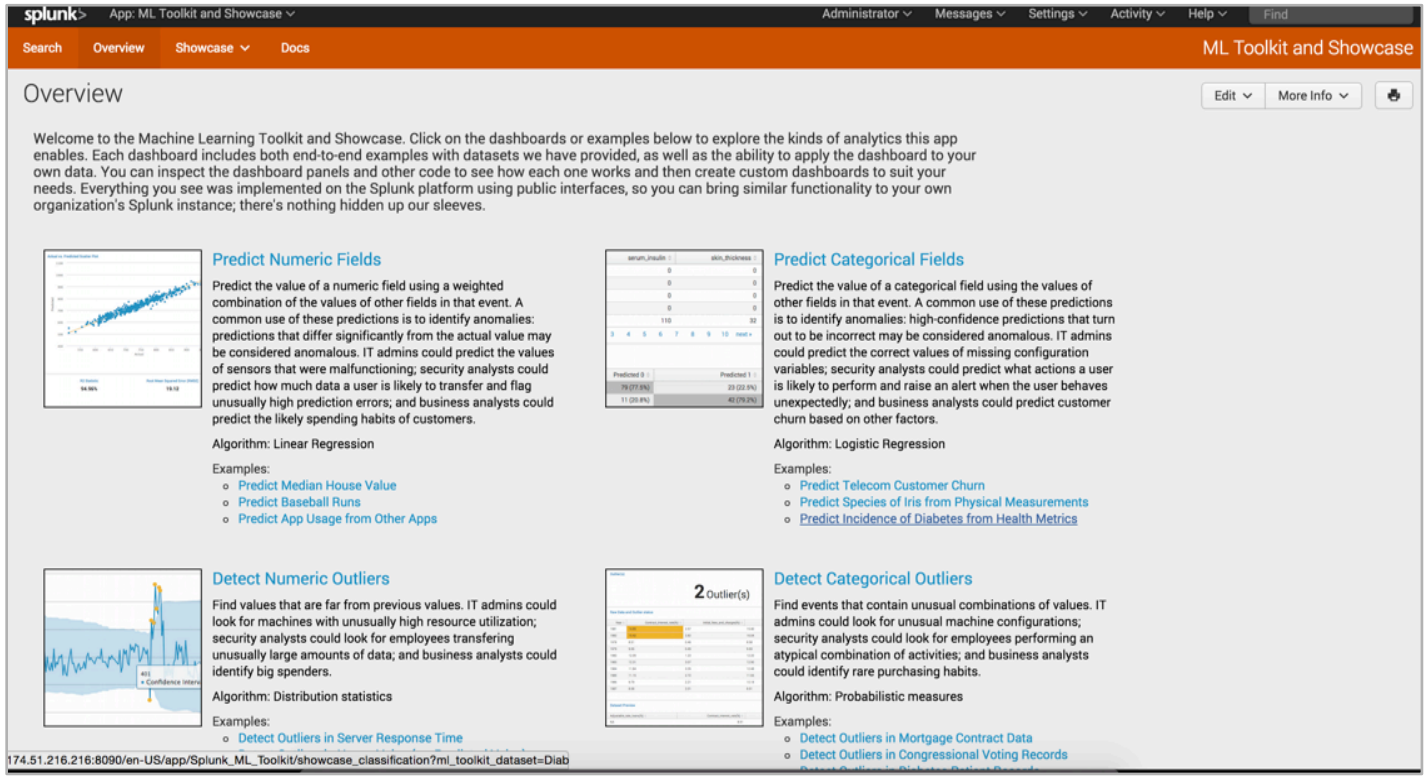


splunk> .conf2017

Machine Learning Toolkit And Showcase

Examples

- Predict Numeric Fields
- Predict Categorical Fields
- Detect Numerical Outliers
- Detect Categorical Outliers
- Forecast Time Series
- Cluster Events



splunk> .conf2017

SPL Examples And Recipes

- Find the needle in the haystack
- Charting statistics and predicting values
- Enriching and converging data sources
- Map geographic data in real time
- Identifying anomalies
- Transactions
- Data exploration & finding relationships between fields
- **Custom commands**

Custom Commands

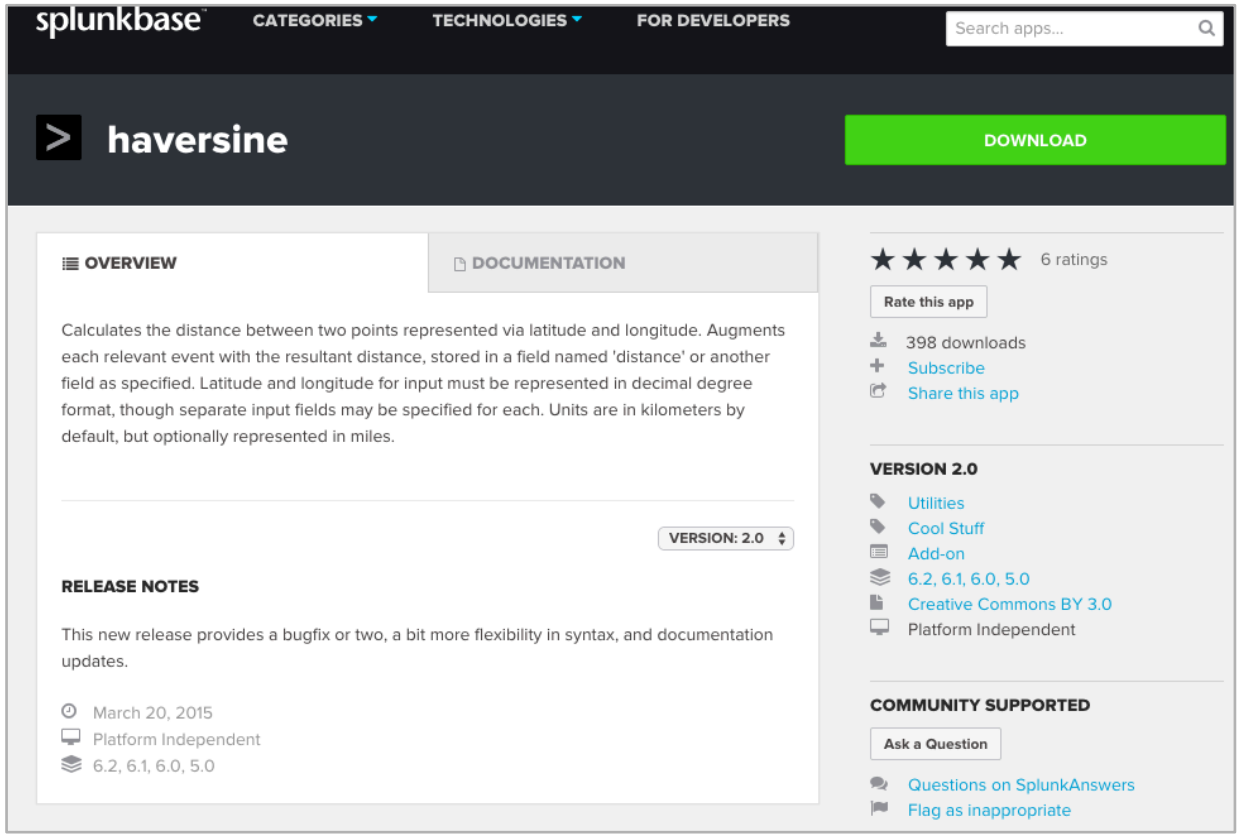
- What is a Custom Command?
  - “| **haversine** origin="47.62,-122.34" outputField=dist lat lon”
- Why do we use Custom Commands?
  - Run other/external algorithms on your Splunk data
  - Save time munging data (see Timewrap!)
  - Because you can!
- Create your own or download as Apps
  - Haversine (Distance between two GPS coords)
  - Timewrap (Enhanced Time overlay)
  - Levenshtein (Fuzzy string compare)
  - Base64 (Encode/Decode)

Custom Commands – Haversine

Examples

- **Download and install App**
  [Haversine]
- *Read documentation* then use in SPL!

```
sourcetype=access*
| iplocation clientip
| search City=A*
| haversine origin="47.62,-122.34" units=mi outputField=dist lat lon
| table clientip, City, dist, lat, lon
```



splunk> .conf2017

Custom Commands – Haversine

Examples

- Download and install App
  Haversine
- ***Read documentation*** **then use in SPL!**

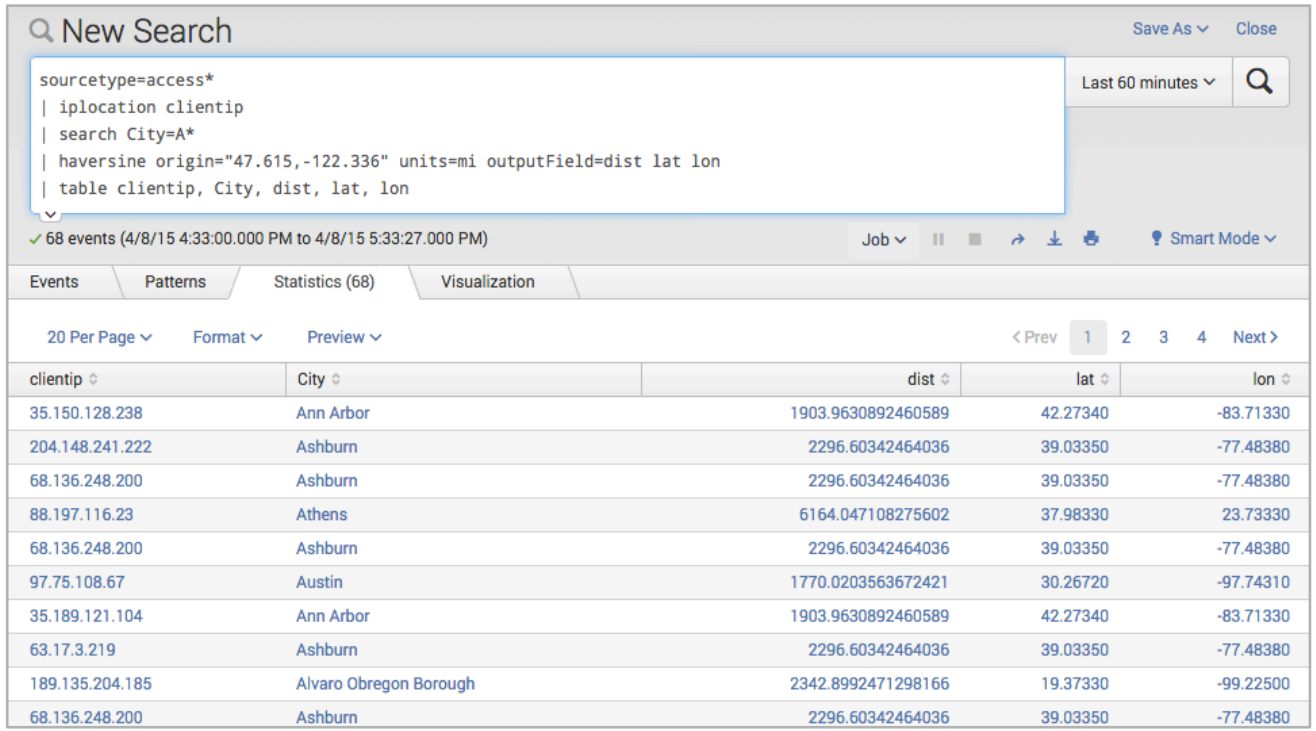
sourcetype=access*

| iplocation clientip

| search City=A*

| haversine origin="47.62,-122.34" units=mi outputField=dist lat lon

| table clientip, City, dist, lat, lon



clientip	City	dist	lat	lon
35.150.128.238	Ann Arbor	1903.9630892460589	42.27340	-83.71330
204.148.241.222	Ashburn	2296.60342464036	39.03350	-77.48380
68.136.248.200	Ashburn	2296.60342464036	39.03350	-77.48380
88.197.116.23	Athens	6164.047108275602	37.98330	23.73330
68.136.248.200	Ashburn	2296.60342464036	39.03350	-77.48380
97.75.108.67	Austin	1770.0203563672421	30.26720	-97.74310
35.189.121.104	Ann Arbor	1903.9630892460589	42.27340	-83.71330
63.17.3.219	Ashburn	2296.60342464036	39.03350	-77.48380
189.135.204.185	Alvaro Obregon Borough	2342.8992471298166	19.37330	-99.22500
68.136.248.200	Ashburn	2296.60342464036	39.03350	-77.48380

splunk> .conf2017

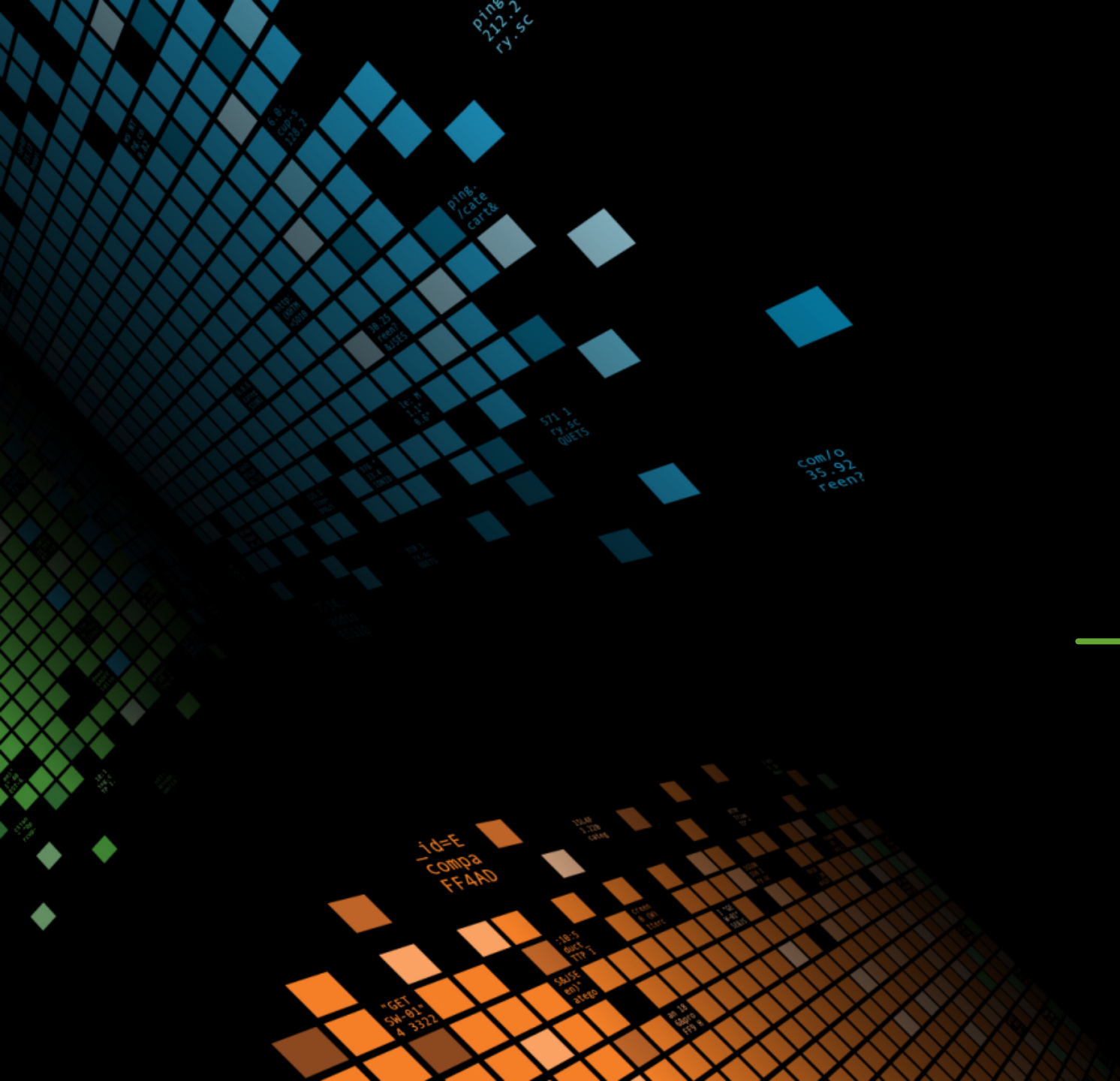
For More Information

- Additional information can be found in:
  - Power of SPL App!
  - Search Manual
  - Blogs
  - Answers
  - Exploring Splunk

splunk> .conf2017

Join The Pony Poll

ponypoll.com/***



Q&A

Thank You

Don't forget to rate this session in the .conf2017 mobile app