splunk> .conf2017

Detecting Numerical Outliers

Advances

Iman Makaremi | Senior Data Scientist

Matthew Modestino | IT Ops Analytics Practitioner

Tuesday, September 26, 2017 | Washington, DC

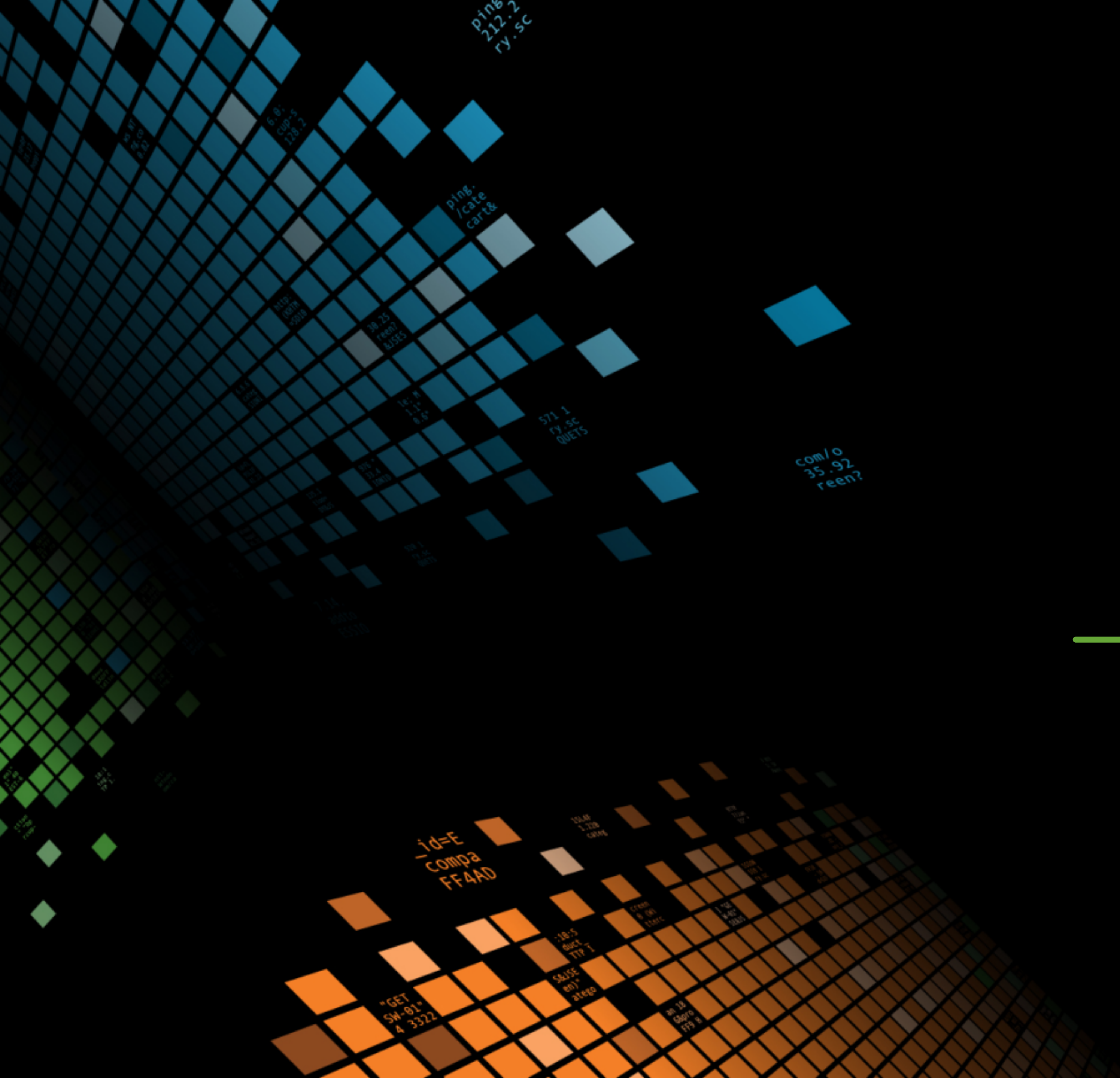
Join the Pony Poll

ponypoll.com/outliers

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 whoami

Introductions

Meet the Splunkers

Hi my name is…

@iman

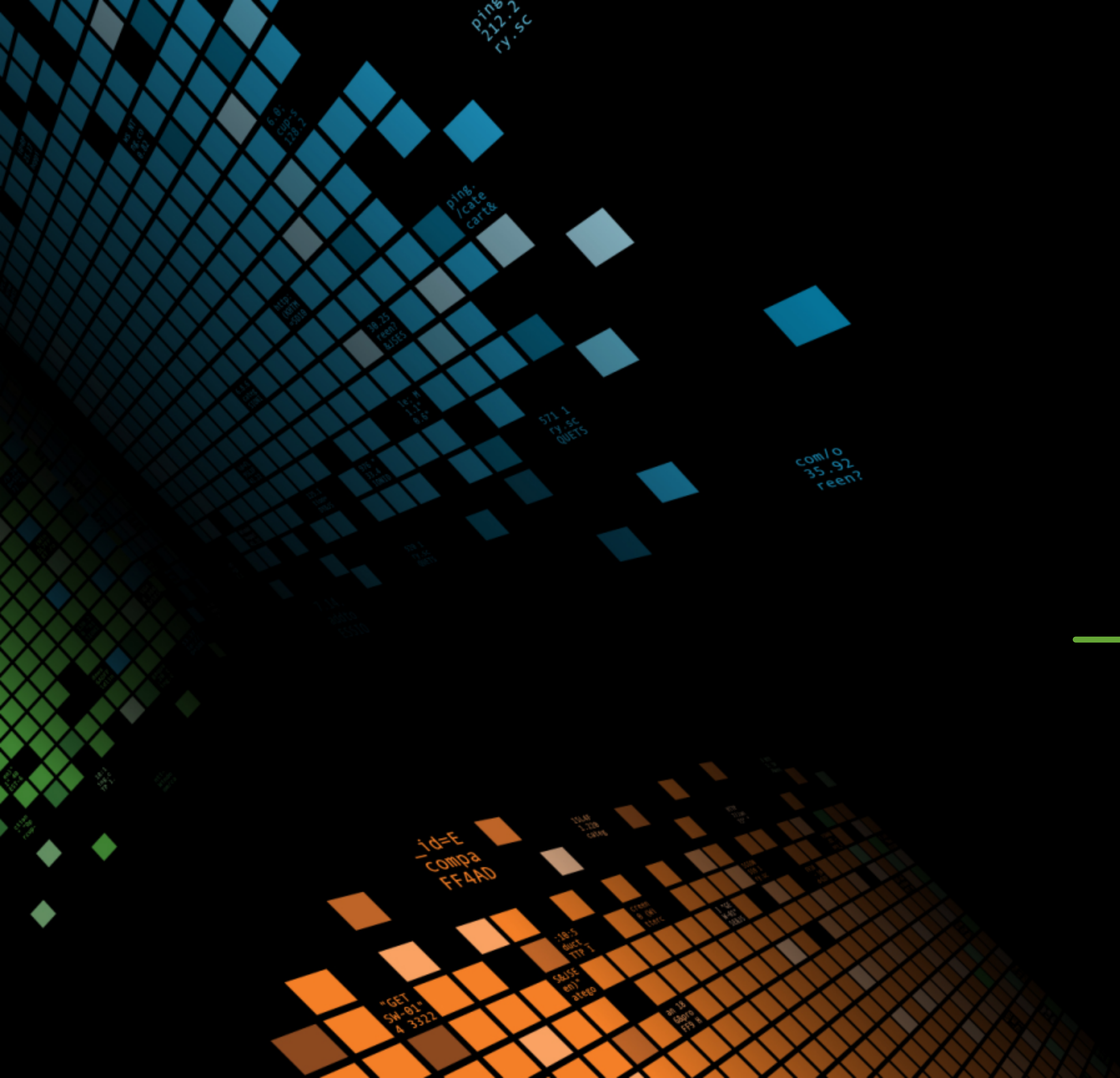
Imakaremi_splunk

imakaremi

@mattymo

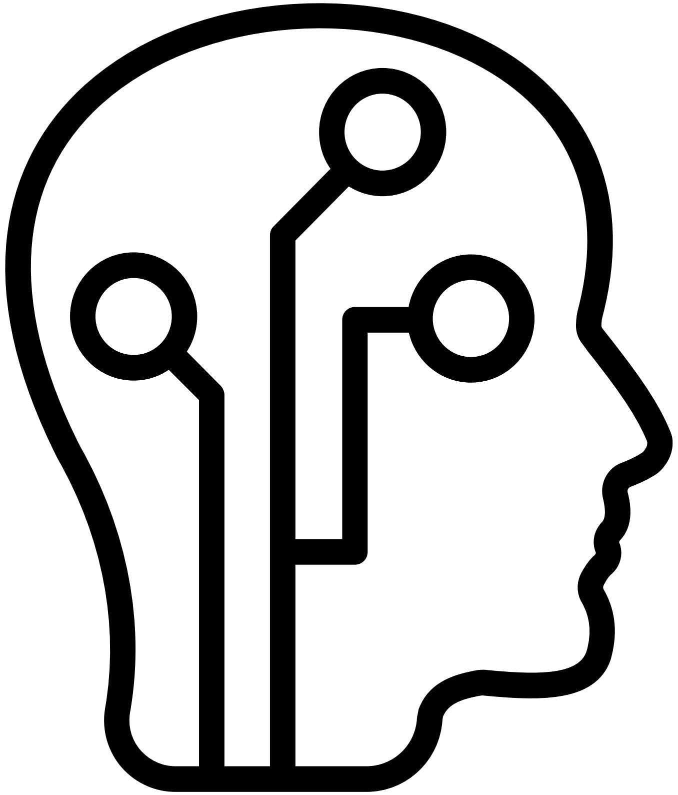
mmodestino_splunk

matthewmodestino



$ ./splunk history

Year in review



Machine Learning Advisory Program

This time last year…

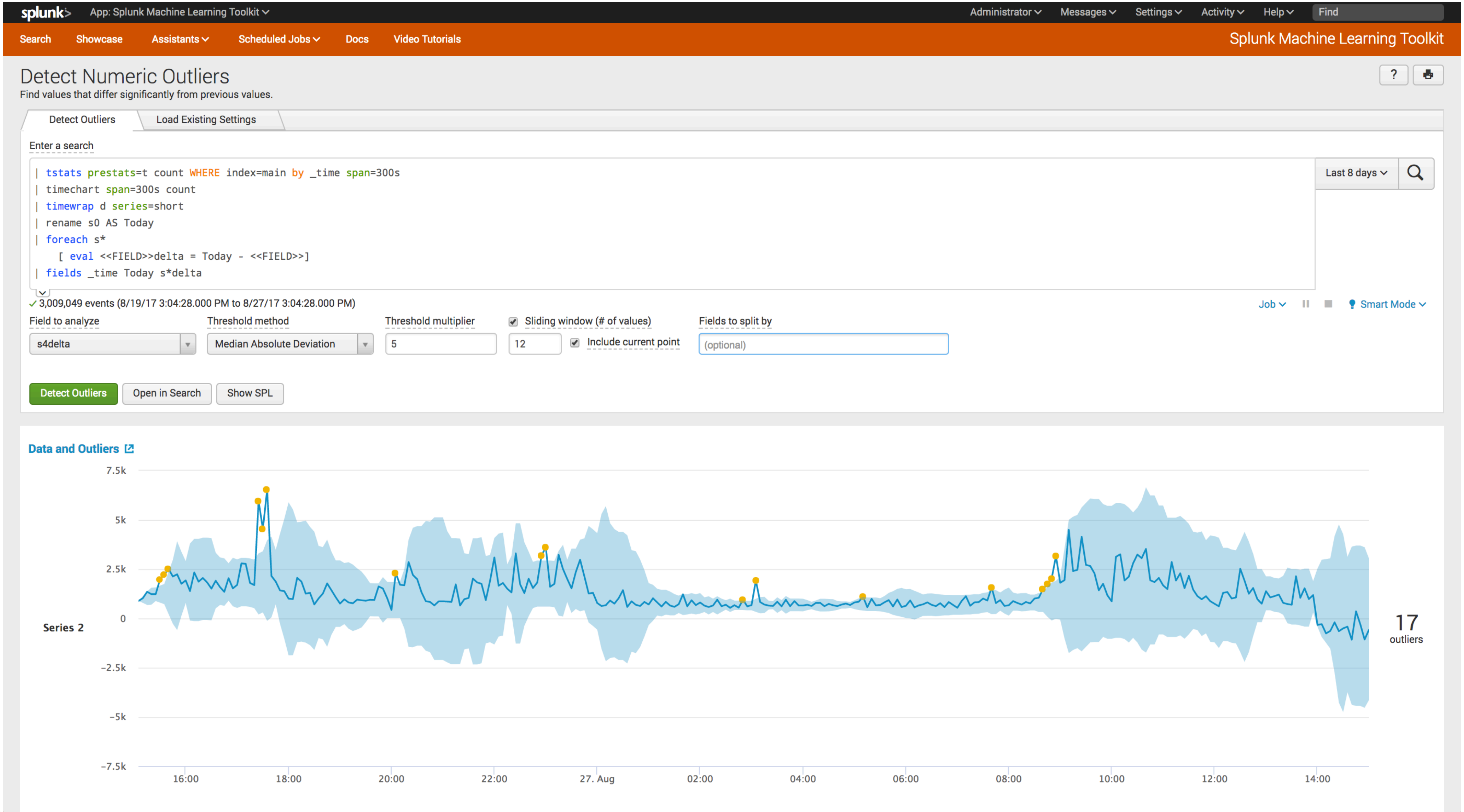
Sharing our experience using the MLTK to build smarter alarms!

Building a Smarter Strategy for Alarms with Splunk Machine Learning Tookit!

Jason Rombough
TELUS, Technology Strategy

Matthew Modestino
Splunk> Advisory Engineer

Modelling Complex System’s Behaviour
Right Algorithm
Custom Visualization
Quick Validation
Generating SPL



Detect Numeric Outliers

Find values that differ significantly from previous values.

Enter a search

```
| tstats prestats=t count WHERE index=main by _time span=300s
| timechart span=300s count
| timewrap d series=short
| rename s0 AS Today
| foreach s*
    [ eval <<FIELD>>delta = Today - <<FIELD>>]
| fields _time Today s*delta
```

Last 8 days

✓ 3,009,049 events (8/19/17 3:04:28.000 PM to 8/27/17 3:04:28.000 PM)

Field to analyze: s4delta

Threshold method: Median Absolute Deviation

Threshold multiplier: 5

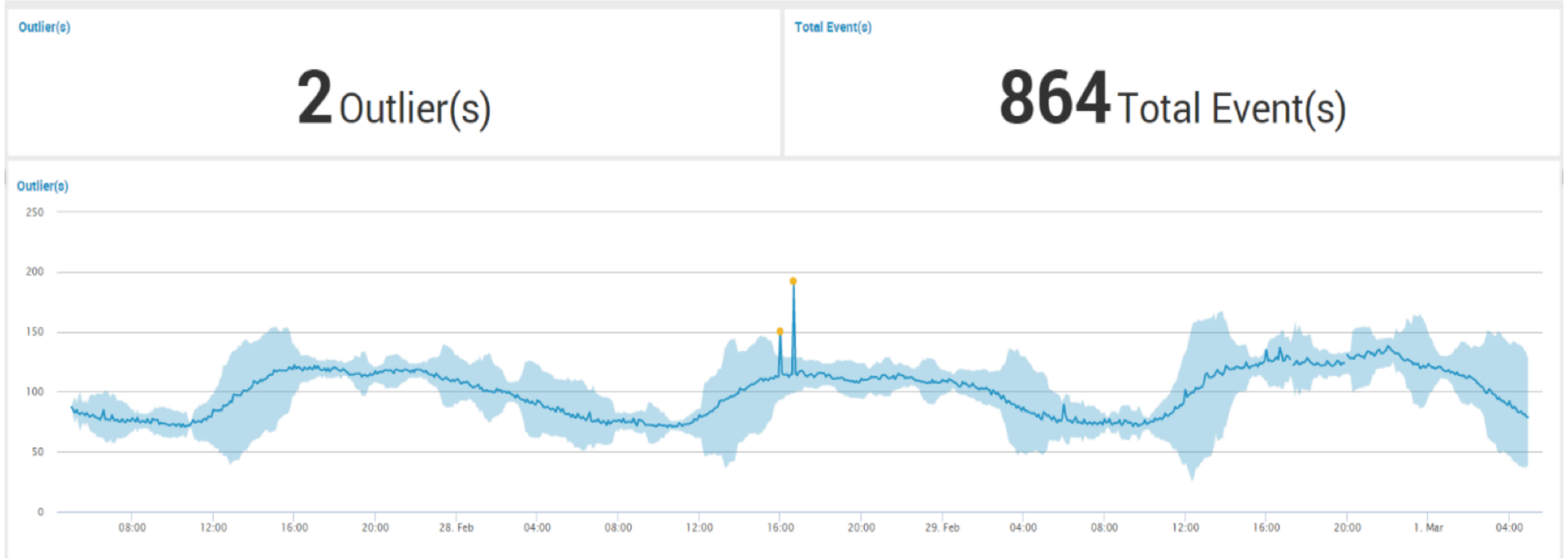
☑ Sliding window (# of values): 12

☑ Include current point

Fields to split by: (optional)

Data and Outliers

17 outliers



Outlier(s)

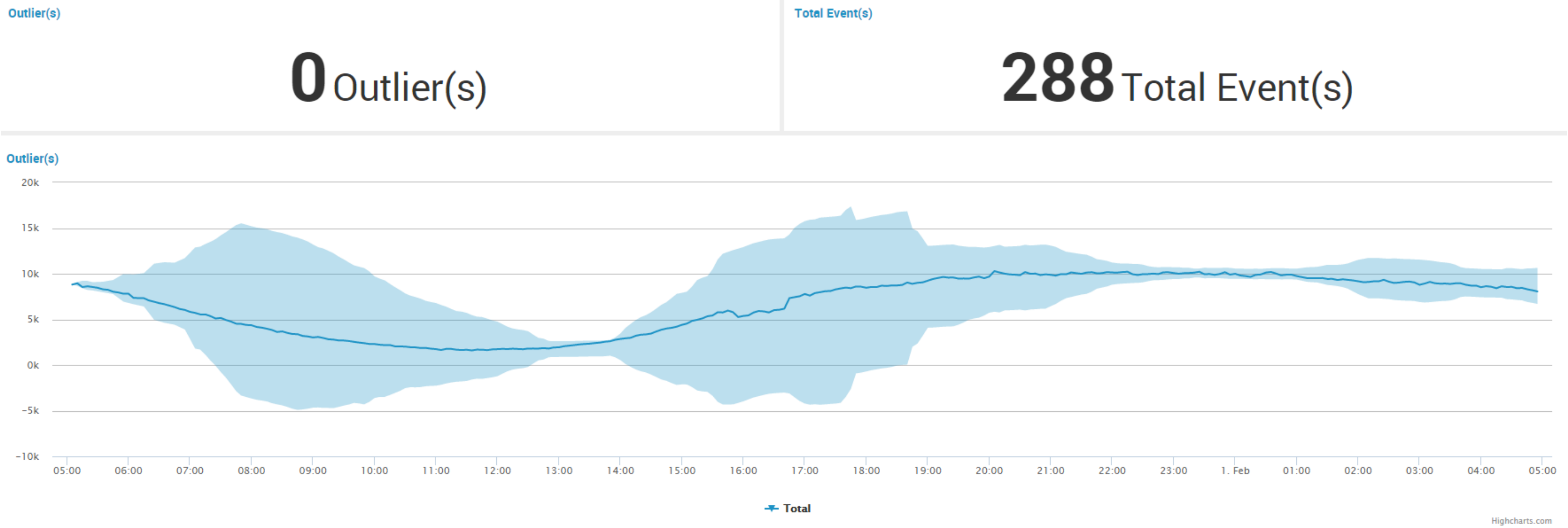
2 Outlier(s)

Total Event(s)

864 Total Event(s)

Outlier(s)

Splunk Machine Learning Toolkit



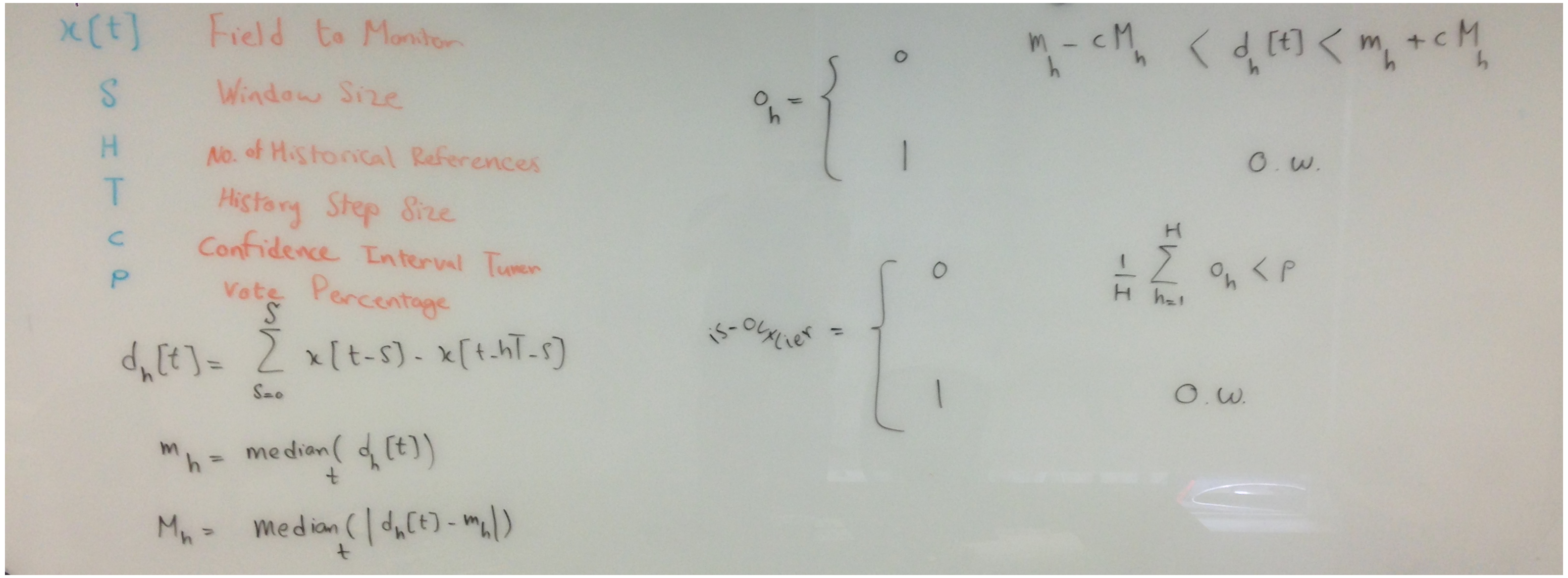
Outlier(s)

0 Outlier(s)

Total Event(s)

288 Total Event(s)

Outlier(s)



$x[t]$ Field to Monitor

S Window Size

H No. of Historical References

T History Step Size

c Confidence Interval Tuner

P Vote Percentage

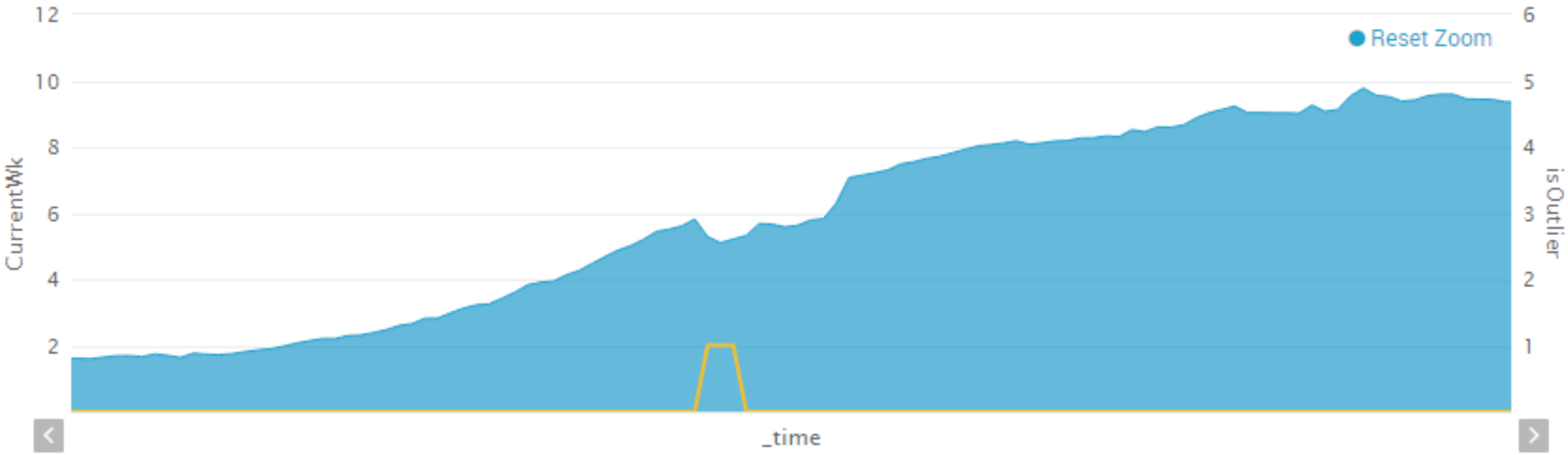
$$d_h[t] = \sum_{s=0}^{S} x[t-s] - x[t-hT-s]$$

$$m_h = \underset{t}{\text{median}}(d_h[t])$$

$$M_h = \underset{t}{\text{median}}(|d_h[t] - m_h|)$$

$$o_h = \begin{cases} 0 & m_h - cM_h < d_h[t] < m_h + cM_h \\ 1 & o.w. \end{cases}$$

$$\text{is-outlier} = \begin{cases} 0 & \frac{1}{H}\sum_{h=1}^{H} o_h < P \\ 1 & o.w. \end{cases}$$



Share the Recipe!

How can I detect outliers in my data?

Tales from production

What did we hear from customers and the community?

Try it out on your KPIs!

Tell us what works, what doesn't & how to make it better & easier!

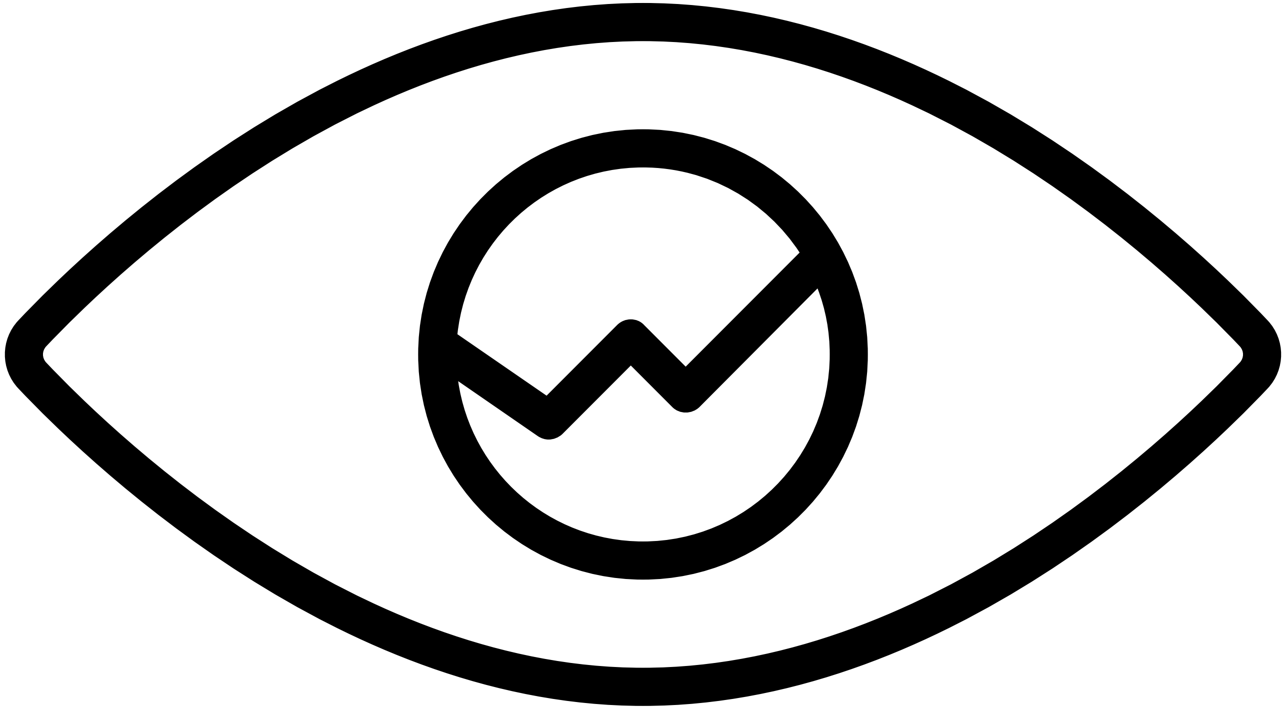
https://github.com/matthewmodestino

$ ./splunk show dependencies

What do we need to get started?

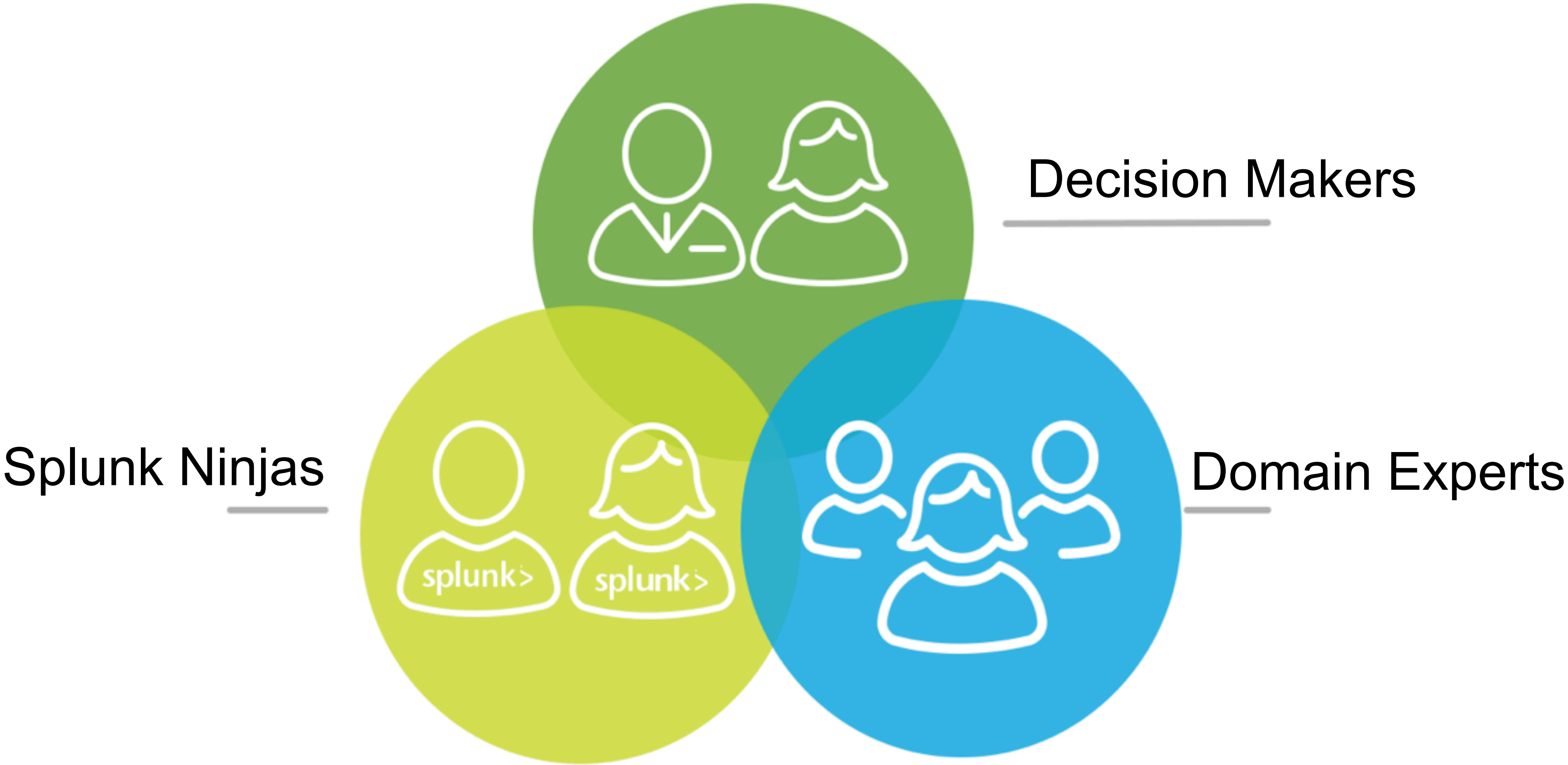
Key Performance Indicators

What do you care about, and what do you do when it breaks?



Splunk Center Of Excellence

Your best and brightest, doing what they do best!

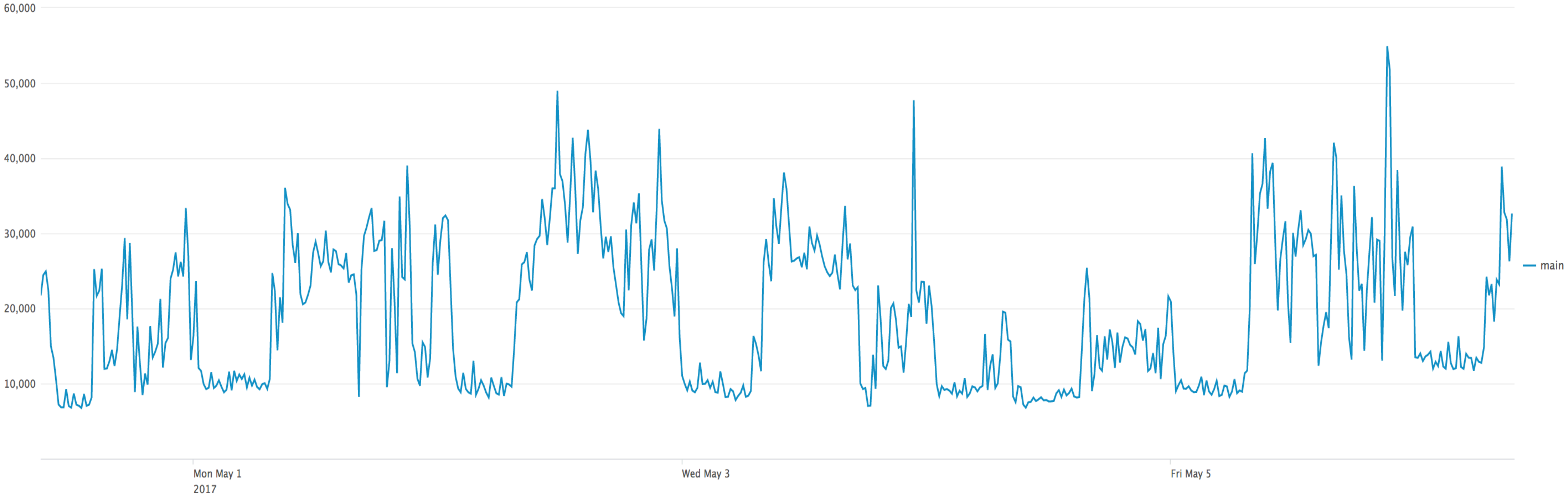


$ ./splunk firstTimeRun

Timewrap and Median Absolute Deviation

Pick A KPI

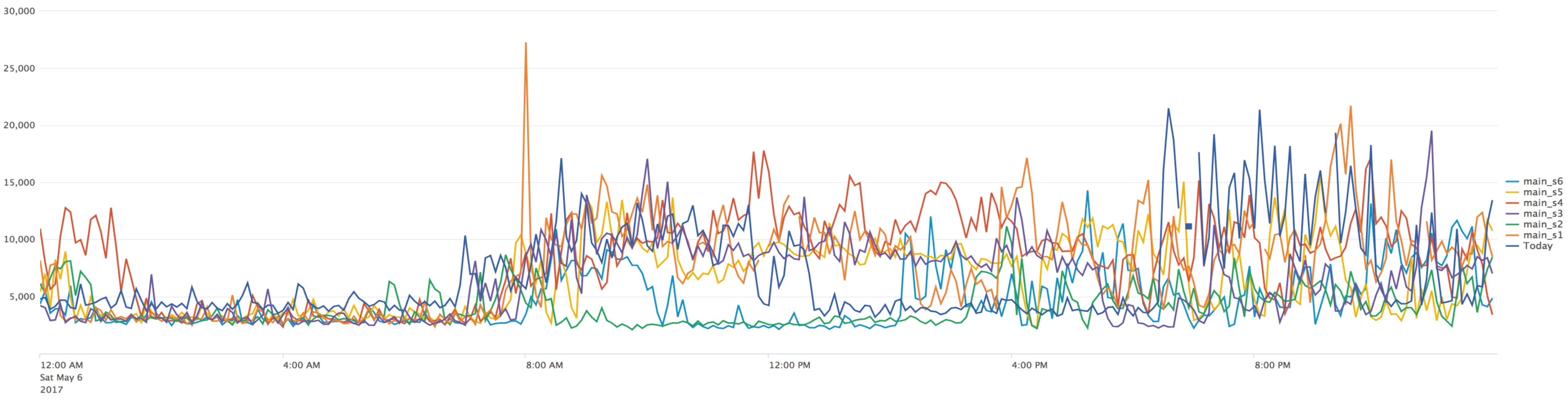
Something that matters to your service or environment

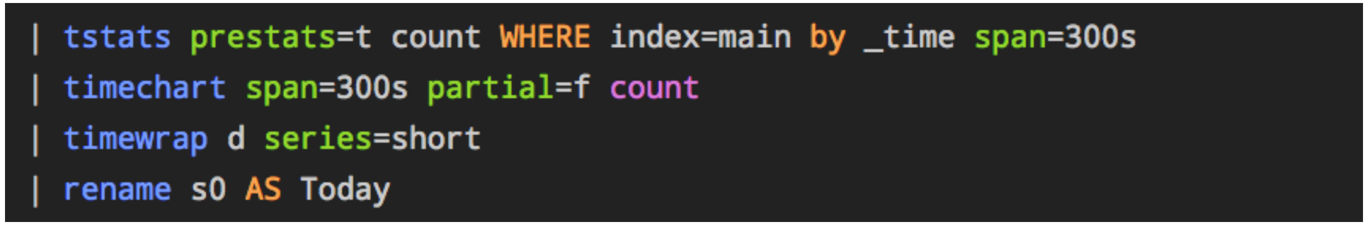


```
| tstats prestats=t count WHERE index=main by _time span=300s
```

Timewrap

Bending stats and time

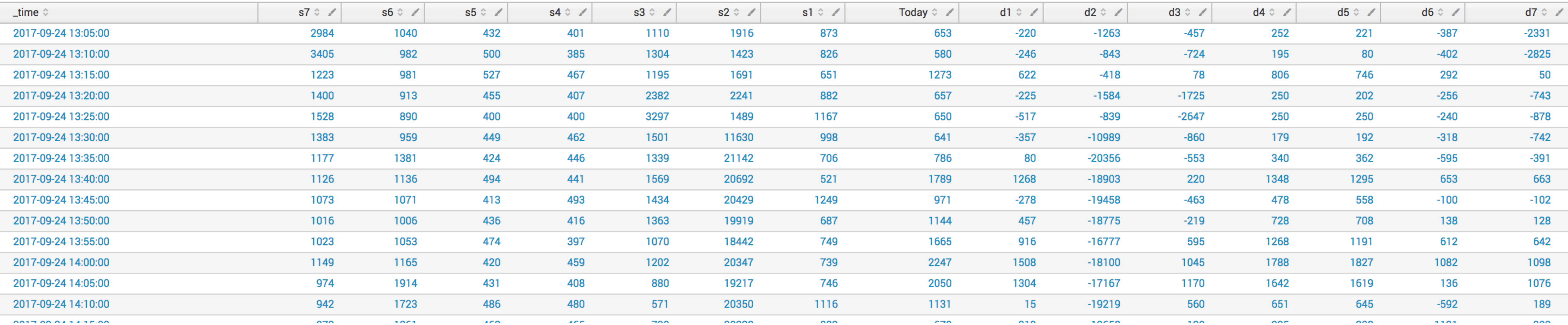




```
| tstats prestats=t count WHERE index=main by _time span=300s
| timechart span=300s partial=f count
| timewrap d series=short
| rename s0 AS Today
```

Deltas

Is today like the other days?

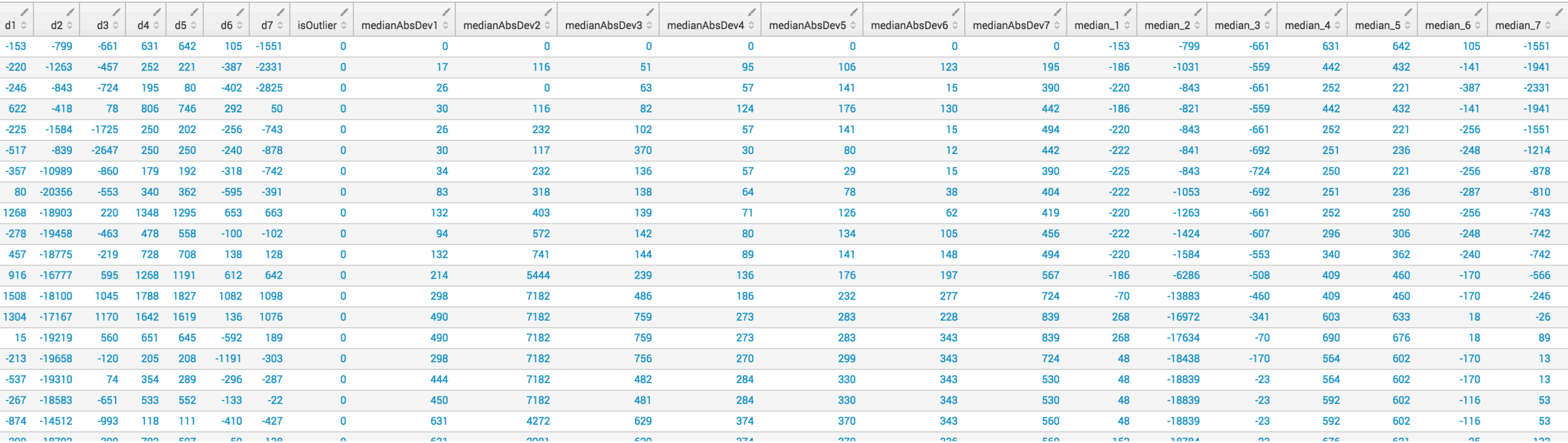


_time	s7	s6	s5	s4	s3	s2	s1	Today	d1	d2	d3	d4	d5	d6	d7
2017-09-24 13:05:00	2984	1040	432	401	1110	1916	873	653	-220	-1263	-457	252	221	-387	-2331
2017-09-24 13:10:00	3405	982	500	385	1304	1423	826	580	-246	-843	-724	195	80	-402	-2825
2017-09-24 13:15:00	1223	981	527	467	1195	1691	651	1273	622	-418	78	806	746	292	50
2017-09-24 13:20:00	1400	913	455	407	2382	2241	882	657	-225	-1584	-1725	250	202	-256	-743
2017-09-24 13:25:00	1528	890	400	400	3297	1489	1167	650	-517	-839	-2647	250	250	-240	-878
2017-09-24 13:30:00	1383	959	449	462	1501	11630	998	641	-357	-10989	-860	179	192	-318	-742
2017-09-24 13:35:00	1177	1381	424	446	1339	21142	706	786	80	-20356	-553	340	362	-595	-391
2017-09-24 13:40:00	1126	1136	494	441	1569	20692	521	1789	1268	-18903	220	1348	1295	653	663
2017-09-24 13:45:00	1073	1071	413	493	1434	20429	1249	971	-278	-19458	-463	478	558	-100	-102
2017-09-24 13:50:00	1016	1006	436	416	1363	19919	687	1144	457	-18775	-219	728	708	138	128
2017-09-24 13:55:00	1023	1053	474	397	1070	18442	749	1665	916	-16777	595	1268	1191	612	642
2017-09-24 14:00:00	1149	1165	420	459	1202	20347	739	2247	1508	-18100	1045	1788	1827	1082	1098
2017-09-24 14:05:00	974	1914	431	408	880	19217	746	2050	1304	-17167	1170	1642	1619	136	1076
2017-09-24 14:10:00	942	1723	486	480	571	20350	1116	1131	15	-19219	560	651	645	-592	189

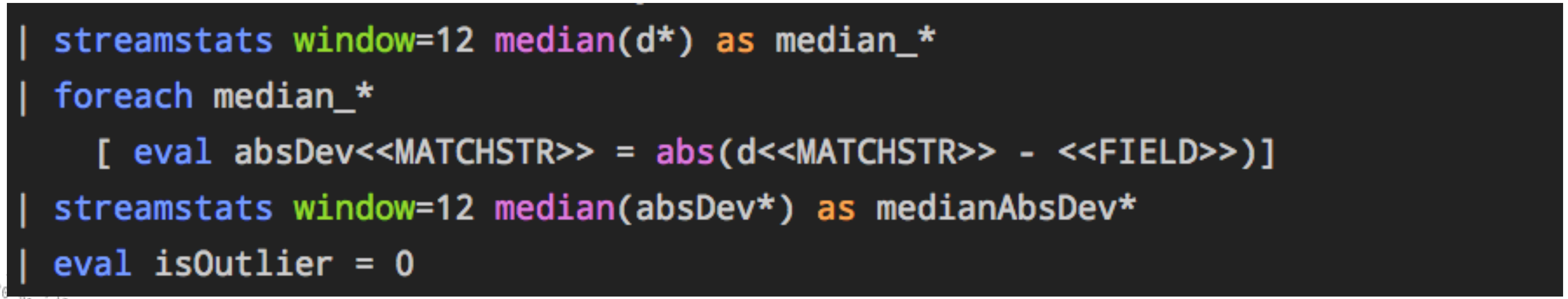
```
| tstats prestats=t count WHERE index=main by _time span=300s
| timechart span=300s partial=f count
| timewrap d series=short
| rename s0 AS Today
| foreach s*
    [ eval d<<MATCHSTR>> = Today - <<FIELD>>]
```

Median Absolute Deviation

Calculate median and median absolute deviation



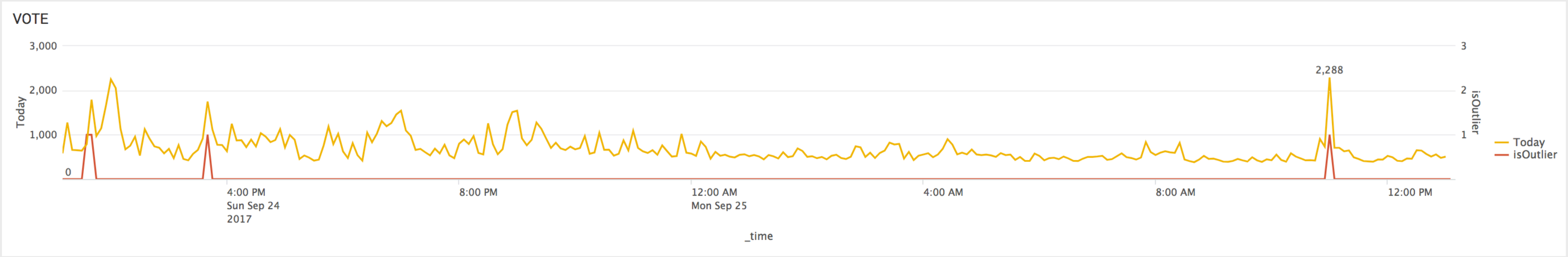
d1	d2	d3	d4	d5	d6	d7	isOutlier	medianAbsDev1	medianAbsDev2	medianAbsDev3	medianAbsDev4	medianAbsDev5	medianAbsDev6	medianAbsDev7	median_1	median_2	median_3	median_4	median_5	median_6	median_7
-153	-799	-661	631	642	105	-1551	0	0	0	0	0	0	0	0	-153	-799	-661	631	642	105	-1551
-220	-1263	-457	252	221	-387	-2331	0	17	116	51	95	106	123	195	-186	-1031	-559	442	432	-141	-1941
-246	-843	-724	195	80	-402	-2825	0	26	0	63	57	141	15	390	-220	-843	-661	252	221	-387	-2331
622	-418	78	806	746	292	50	0	30	116	82	124	176	130	442	-186	-821	-559	442	432	-141	-1941
-225	-1584	-1725	250	202	-256	-743	0	26	232	102	57	141	15	494	-220	-843	-661	252	221	-256	-1551
-517	-839	-2647	250	250	-240	-878	0	30	117	370	30	80	12	442	-222	-841	-692	251	236	-248	-1214
-357	-10989	-860	179	192	-318	-742	0	34	232	136	57	29	15	390	-225	-843	-724	250	221	-256	-878
80	-20356	-553	340	362	-595	-391	0	83	318	138	64	78	38	404	-222	-1053	-692	251	236	-287	-810
1268	-18903	220	1348	1295	653	663	0	132	403	139	71	126	62	419	-220	-1263	-661	252	250	-256	-743
-278	-19458	-463	478	558	-100	-102	0	94	572	142	80	134	105	456	-222	-1424	-607	296	306	-248	-742
457	-18775	-219	728	708	138	128	0	132	741	144	89	141	148	494	-220	-1584	-553	340	362	-240	-742
916	-16777	595	1268	1191	612	642	0	214	5444	239	136	176	197	567	-186	-6286	-508	409	460	-170	-566
1508	-18100	1045	1788	1827	1082	1098	0	298	7182	486	186	232	277	724	-70	-13883	-460	409	460	-170	-246
1304	-17167	1170	1642	1619	136	1076	0	490	7182	759	273	283	228	839	268	-16972	-341	603	633	18	-26
15	-19219	560	651	645	-592	189	0	490	7182	759	273	283	343	839	268	-17634	-70	690	676	18	89
-213	-19658	-120	205	208	-1191	-303	0	298	7182	756	270	299	343	724	48	-18438	-170	564	602	-170	13
-537	-19310	74	354	289	-296	-287	0	444	7182	482	284	330	343	530	48	-18839	-23	564	602	-170	13
-267	-18583	-651	533	552	-133	-22	0	450	7182	481	284	330	343	530	48	-18839	-23	592	602	-116	53
-874	-14512	-993	118	111	-410	-427	0	631	4272	629	374	370	343	560	48	-18839	-23	592	602	-116	53

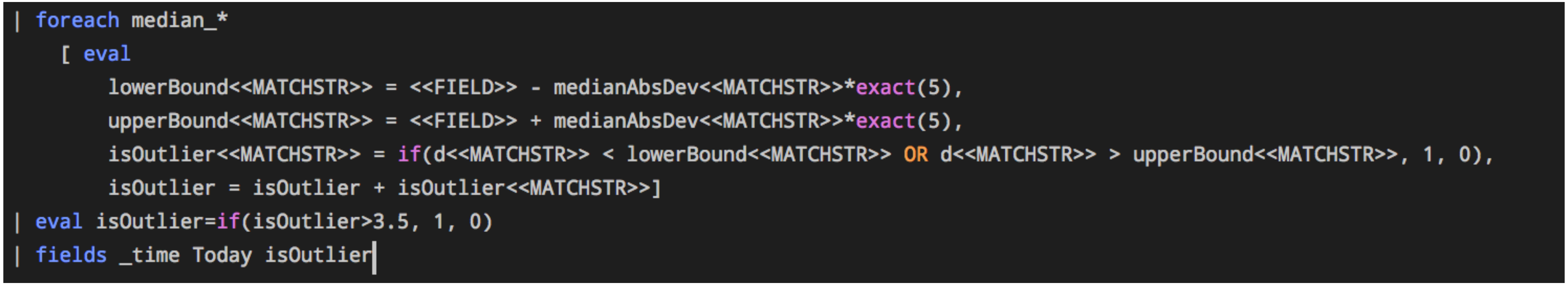


```
| streamstats window=12 median(d*) as median_*
| foreach median_*
    [ eval absDev<<MATCHSTR>> = abs(d<<MATCHSTR>> - <<FIELD>>)]
| streamstats window=12 median(absDev*) as medianAbsDev*
| eval isOutlier = 0
```

Vote

Are these the droids we are looking for?

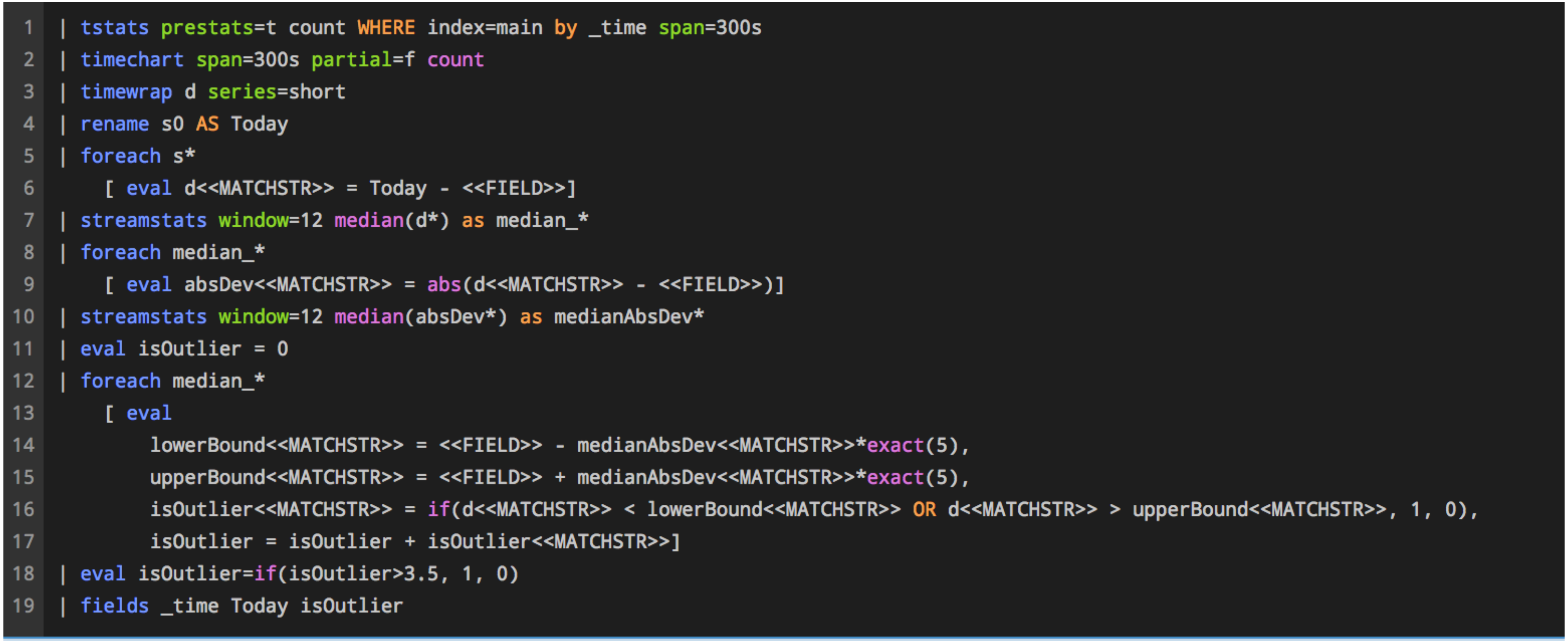




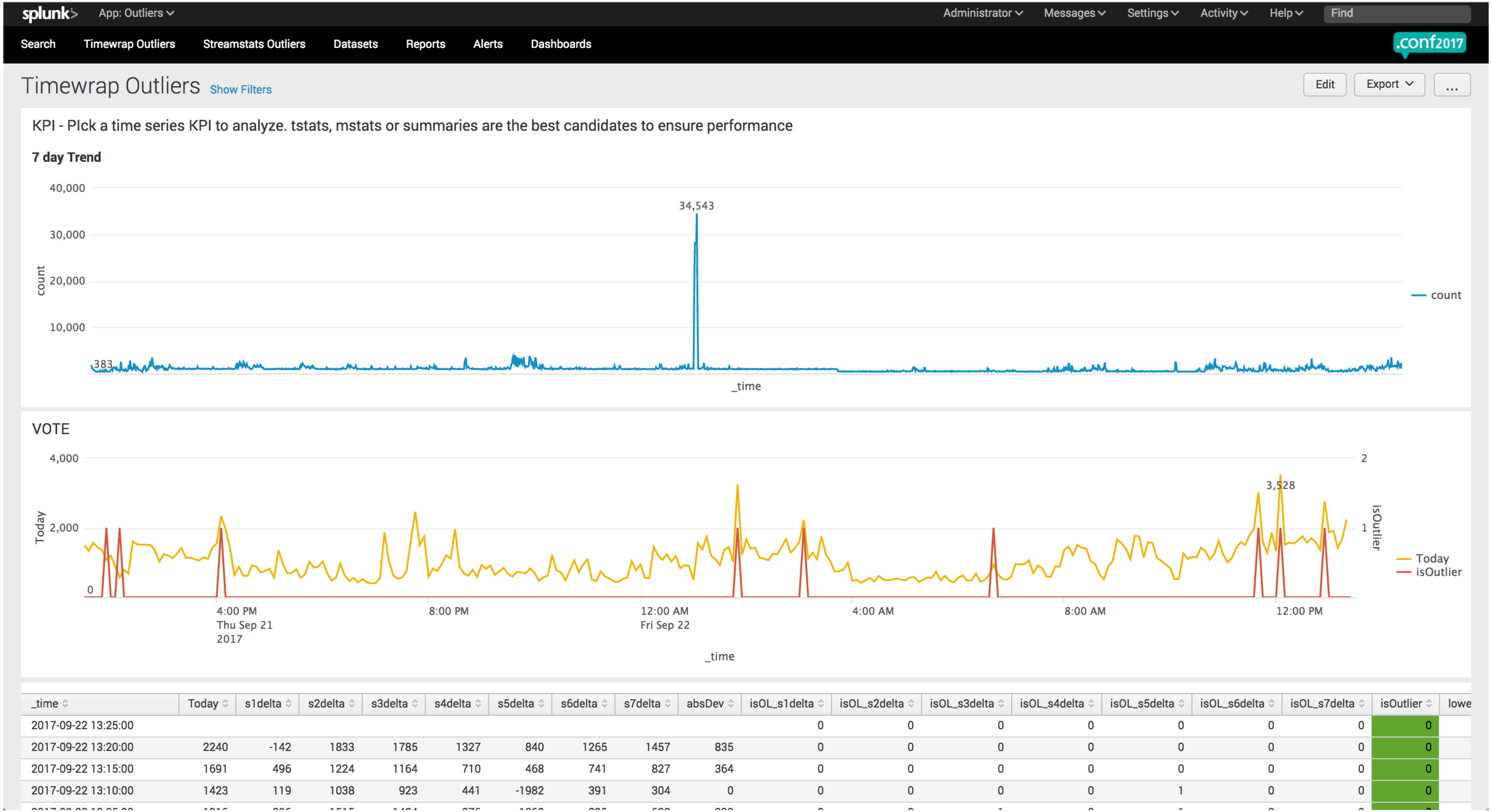
```
| foreach median_*
    [ eval
        lowerBound<<MATCHSTR>> = <<FIELD>> - medianAbsDev<<MATCHSTR>>*exact(5),
        upperBound<<MATCHSTR>> = <<FIELD>> + medianAbsDev<<MATCHSTR>>*exact(5),
        isOutlier<<MATCHSTR>> = if(d<<MATCHSTR>> < lowerBound<<MATCHSTR>> OR d<<MATCHSTR>> > upperBound<<MATCHSTR>>, 1, 0),
        isOutlier = isOutlier + isOutlier<<MATCHSTR>>]
| eval isOutlier=if(isOutlier>3.5, 1, 0)
| fields _time Today isOutlier
```

Put it all together

Timechart, Timewrap, Foreach, Streamstats, Eval



```
| tstats prestats=t count WHERE index=main by _time span=300s
| timechart span=300s partial=f count
| timewrap d series=short
| rename s0 AS Today
| foreach s*
    [ eval d<<MATCHSTR>> = Today - <<FIELD>>]
| streamstats window=12 median(d*) as median_*
| foreach median_*
    [ eval absDev<<MATCHSTR>> = abs(d<<MATCHSTR>> - <<FIELD>>)]
| streamstats window=12 median(absDev*) as medianAbsDev*
| eval isOutlier = 0
| foreach median_*
    [ eval
        lowerBound<<MATCHSTR>> = <<FIELD>> - medianAbsDev<<MATCHSTR>>*exact(5),
        upperBound<<MATCHSTR>> = <<FIELD>> + medianAbsDev<<MATCHSTR>>*exact(5),
        isOutlier<<MATCHSTR>> = if(d<<MATCHSTR>> < lowerBound<<MATCHSTR>> OR d<<MATCHSTR>> > upperBound<<MATCHSTR>>, 1, 0),
        isOutlier = isOutlier + isOutlier<<MATCHSTR>>]
| eval isOutlier=if(isOutlier>3.5, 1, 0)
| fields _time Today isOutlier
```



Timewrap Outliers

Show Filters

KPI - PIck a time series KPI to analyze. tstats, mstats or summaries are the best candidates to ensure performance

_time	Today	s1delta	s2delta	s3delta	s4delta	s5delta	s6delta	s7delta	absDev	isOL_s1delta	isOL_s2delta	isOL_s3delta	isOL_s4delta	isOL_s5delta	isOL_s6delta	isOL_s7delta	isOutlier	lowe
2017-09-22 13:25:00										0	0	0	0	0	0	0	0	
2017-09-22 13:20:00	2240	-142	1833	1785	1327	840	1265	1457	835	0	0	0	0	0	0	0	0	
2017-09-22 13:15:00	1691	496	1224	1164	710	468	741	827	364	0	0	0	0	0	0	0	0	
2017-09-22 13:10:00	1423	119	1038	923	441	-1982	391	304	0	0	0	0	0	1	0	0	0	

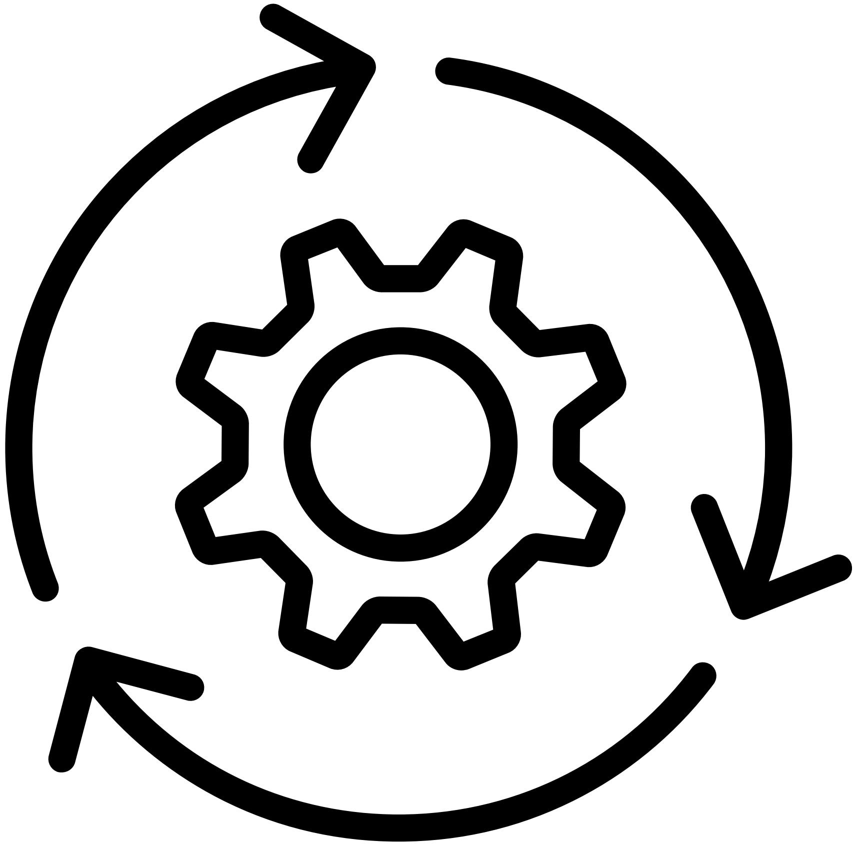
splunk> .conf2017

$ ./splunk outliers redux

Streamstats all the way down!

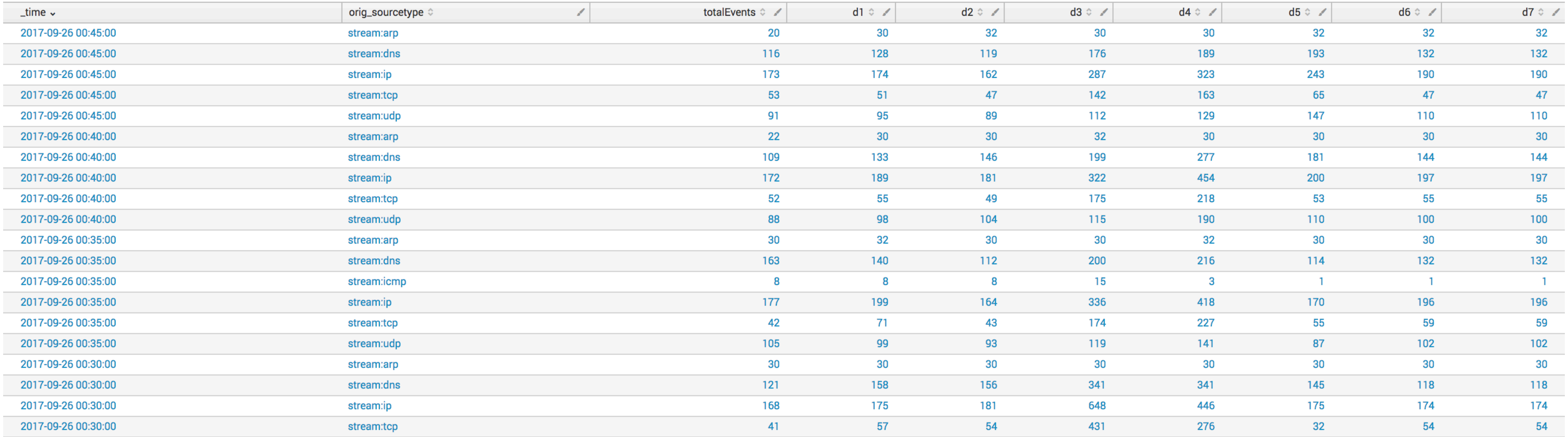
Scaling out the approach

Neat trick! Do it again! But this time, can we run it on multiple fields?

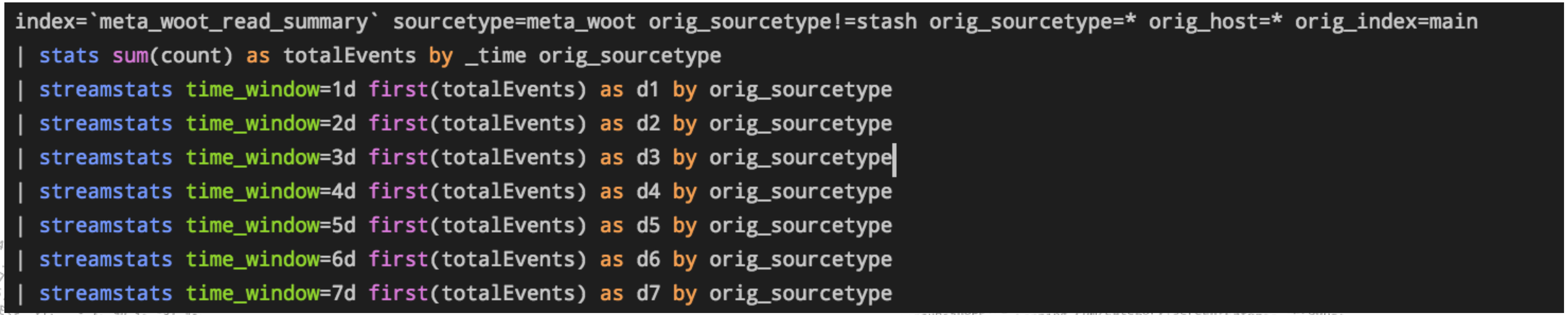


Streamstats

Streamstats-y way to do Timewrap, with the ability to split by!



_time	orig_sourcetype	totalEvents	d1	d2	d3	d4	d5	d6	d7
2017-09-26 00:45:00	stream:arp	20	30	32	30	30	32	32	32
2017-09-26 00:45:00	stream:dns	116	128	119	176	189	193	132	132
2017-09-26 00:45:00	stream:ip	173	174	162	287	323	243	190	190
2017-09-26 00:45:00	stream:tcp	53	51	47	142	163	65	47	47
2017-09-26 00:45:00	stream:udp	91	95	89	112	129	147	110	110
2017-09-26 00:40:00	stream:arp	22	30	30	32	30	30	30	30
2017-09-26 00:40:00	stream:dns	109	133	146	199	277	181	144	144
2017-09-26 00:40:00	stream:ip	172	189	181	322	454	200	197	197
2017-09-26 00:40:00	stream:tcp	52	55	49	175	218	53	55	55
2017-09-26 00:40:00	stream:udp	88	98	104	115	190	110	100	100
2017-09-26 00:35:00	stream:arp	30	32	30	30	32	30	30	30
2017-09-26 00:35:00	stream:dns	163	140	112	200	216	114	132	132
2017-09-26 00:35:00	stream:icmp	8	8	8	15	3	1	1	1
2017-09-26 00:35:00	stream:ip	177	199	164	336	418	170	196	196
2017-09-26 00:35:00	stream:tcp	42	71	43	174	227	55	59	59
2017-09-26 00:35:00	stream:udp	105	99	93	119	141	87	102	102
2017-09-26 00:30:00	stream:arp	30	30	30	30	30	30	30	30
2017-09-26 00:30:00	stream:dns	121	158	156	341	341	145	118	118
2017-09-26 00:30:00	stream:ip	168	175	181	648	446	175	174	174
2017-09-26 00:30:00	stream:tcp	41	57	54	431	276	32	54	54

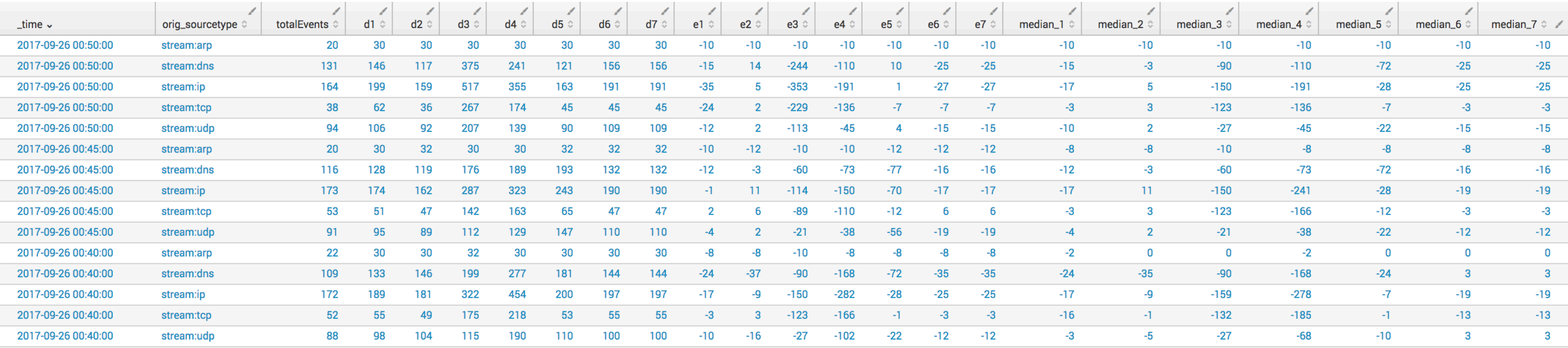


```
index=`meta_woot_read_summary` sourcetype=meta_woot orig_sourcetype!=stash orig_sourcetype=* orig_host=* orig_index=main
| stats sum(count) as totalEvents by _time orig_sourcetype
| streamstats time_window=1d first(totalEvents) as d1 by orig_sourcetype
| streamstats time_window=2d first(totalEvents) as d2 by orig_sourcetype
| streamstats time_window=3d first(totalEvents) as d3 by orig_sourcetype
| streamstats time_window=4d first(totalEvents) as d4 by orig_sourcetype
| streamstats time_window=5d first(totalEvents) as d5 by orig_sourcetype
| streamstats time_window=6d first(totalEvents) as d6 by orig_sourcetype
| streamstats time_window=7d first(totalEvents) as d7 by orig_sourcetype
```

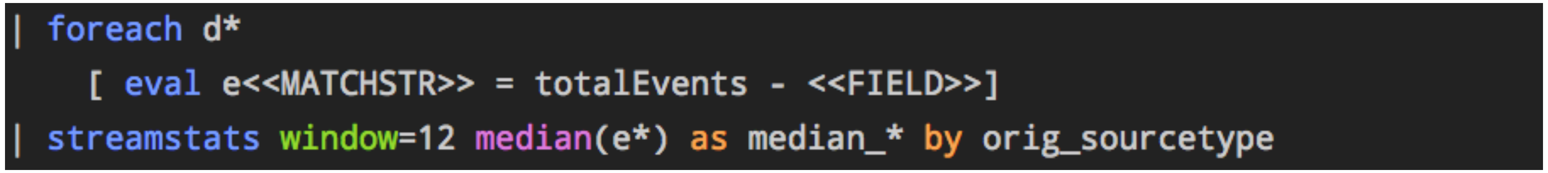
splunk> .conf2017

Deltas

e = deltas



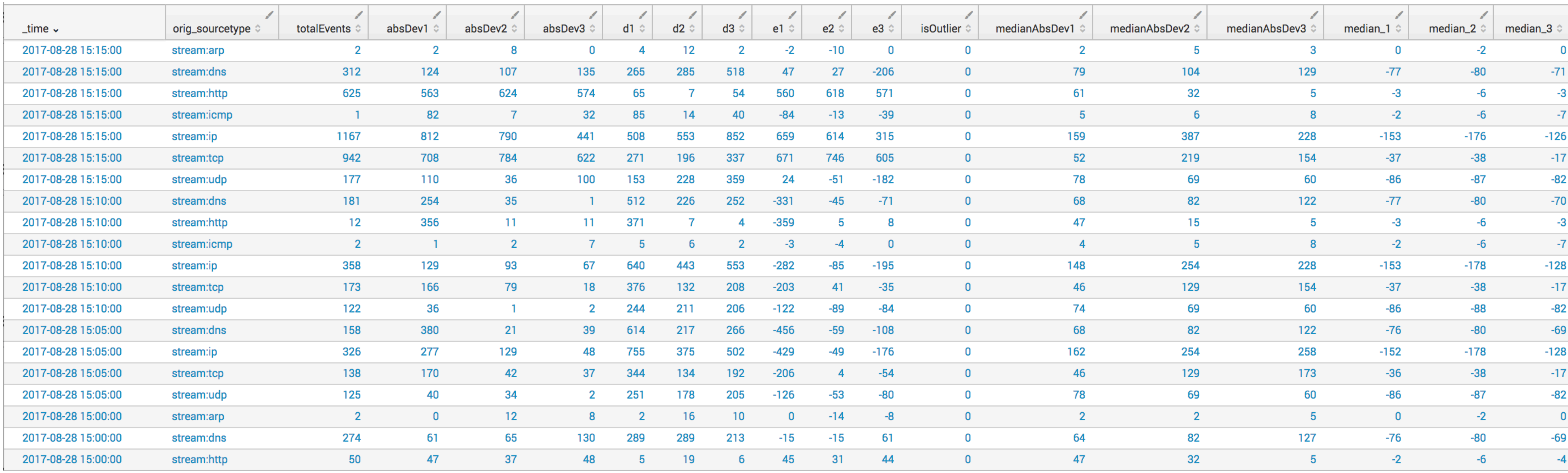
_time	orig_sourcetype	totalEvents	d1	d2	d3	d4	d5	d6	d7	e1	e2	e3	e4	e5	e6	e7	median_1	median_2	median_3	median_4	median_5	median_6	median_7
2017-09-26 00:50:00	stream:arp	20	30	30	30	30	30	30	30	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10
2017-09-26 00:50:00	stream:dns	131	146	117	375	241	121	156	156	-15	14	-244	-110	10	-25	-25	-15	-3	-90	-110	-72	-25	-25
2017-09-26 00:50:00	stream:ip	164	199	159	517	355	163	191	191	-35	5	-353	-191	1	-27	-27	-17	5	-150	-191	-28	-25	-25
2017-09-26 00:50:00	stream:tcp	38	62	36	267	174	45	45	45	-24	2	-229	-136	-7	-7	-7	-3	3	-123	-136	-7	-3	-3
2017-09-26 00:50:00	stream:udp	94	106	92	207	139	90	109	109	-12	2	-113	-45	4	-15	-15	-10	2	-27	-45	-22	-15	-15
2017-09-26 00:45:00	stream:arp	20	30	32	30	30	32	32	32	-10	-12	-10	-10	-12	-12	-12	-8	-8	-10	-8	-8	-8	-8
2017-09-26 00:45:00	stream:dns	116	128	119	176	189	193	132	132	-12	-3	-60	-73	-77	-16	-16	-12	-3	-60	-73	-72	-16	-16
2017-09-26 00:45:00	stream:ip	173	174	162	287	323	243	190	190	-1	11	-114	-150	-70	-17	-17	-17	11	-150	-241	-28	-19	-19
2017-09-26 00:45:00	stream:tcp	53	51	47	142	163	65	47	47	2	6	-89	-110	-12	6	6	-3	3	-123	-166	-12	-3	-3
2017-09-26 00:45:00	stream:udp	91	95	89	112	129	147	110	110	-4	2	-21	-38	-56	-19	-19	-4	2	-21	-38	-22	-12	-12
2017-09-26 00:40:00	stream:arp	22	30	30	32	30	30	30	30	-8	-8	-10	-8	-8	-8	-8	-2	0	0	-2	0	0	0
2017-09-26 00:40:00	stream:dns	109	133	146	199	277	181	144	144	-24	-37	-90	-168	-72	-35	-35	-24	-35	-90	-168	-24	3	3
2017-09-26 00:40:00	stream:ip	172	189	181	322	454	200	197	197	-17	-9	-150	-282	-28	-25	-25	-17	-9	-159	-278	-7	-19	-19
2017-09-26 00:40:00	stream:tcp	52	55	49	175	218	53	55	55	-3	3	-123	-166	-1	-3	-3	-16	-1	-132	-185	-1	-13	-13
2017-09-26 00:40:00	stream:udp	88	98	104	115	190	110	100	100	-10	-16	-27	-102	-22	-12	-12	-3	-5	-27	-68	-10	3	3



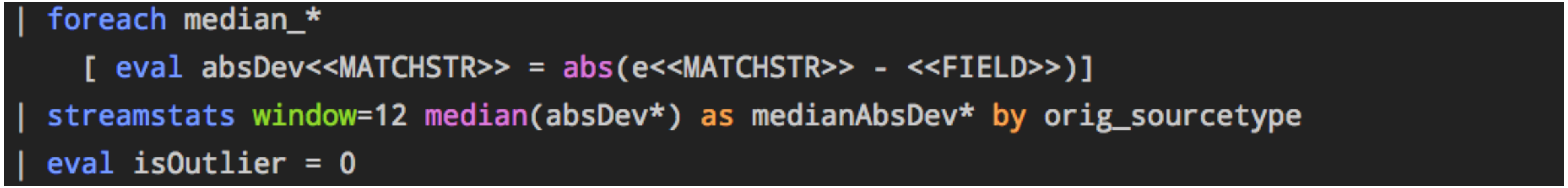
```
| foreach d*
    [ eval e<<MATCHSTR>> = totalEvents - <<FIELD>>]
| streamstats window=12 median(e*) as median_* by orig_sourcetype
```

Math

Median Absolute Deviation



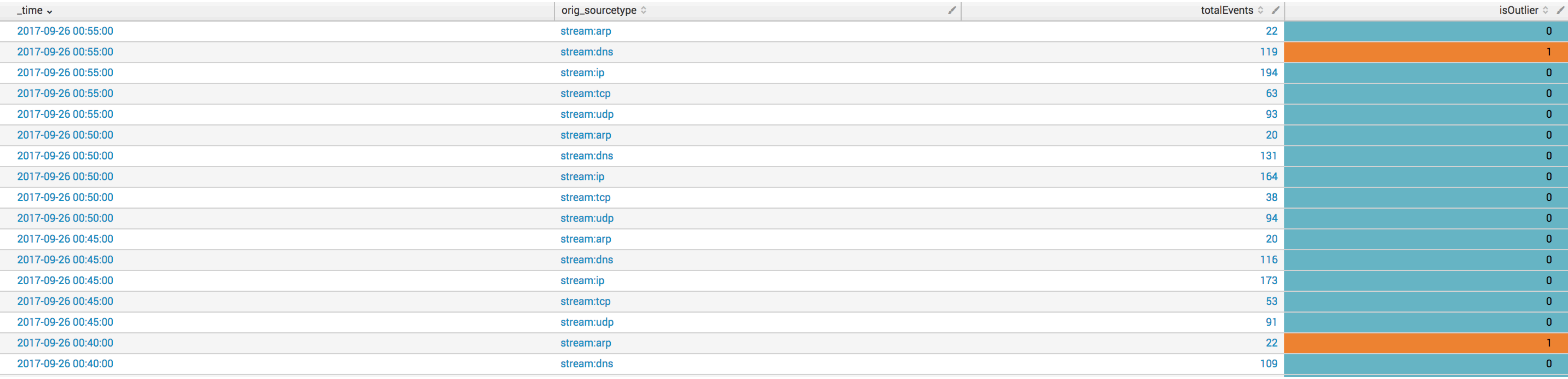
_time	orig_sourcetype	totalEvents	absDev1	absDev2	absDev3	d1	d2	d3	e1	e2	e3	isOutlier	medianAbsDev1	medianAbsDev2	medianAbsDev3	median_1	median_2	median_3
2017-08-28 15:15:00	stream:arp	2	2	8	0	4	12	2	-2	-10	0	0	2	5	3	0	-2	0
2017-08-28 15:15:00	stream:dns	312	124	107	135	265	285	518	47	27	-206	0	79	104	129	-77	-80	-71
2017-08-28 15:15:00	stream:http	625	563	624	574	65	7	54	560	618	571	0	61	32	5	-3	-6	-3
2017-08-28 15:15:00	stream:icmp	1	82	7	32	85	14	40	-84	-13	-39	0	5	6	8	-2	-6	-7
2017-08-28 15:15:00	stream:ip	1167	812	790	441	508	553	852	659	614	315	0	159	387	228	-153	-176	-126
2017-08-28 15:15:00	stream:tcp	942	708	784	622	271	196	337	671	746	605	0	52	219	154	-37	-38	-17
2017-08-28 15:15:00	stream:udp	177	110	36	100	153	228	359	24	-51	-182	0	78	69	60	-86	-87	-82
2017-08-28 15:10:00	stream:dns	181	254	35	1	512	226	252	-331	-45	-71	0	68	82	122	-77	-80	-70
2017-08-28 15:10:00	stream:http	12	356	11	11	371	7	4	-359	5	8	0	47	15	5	-3	-6	-3
2017-08-28 15:10:00	stream:icmp	2	1	2	7	5	6	2	-3	-4	0	0	4	5	8	-2	-6	-7
2017-08-28 15:10:00	stream:ip	358	129	93	67	640	443	553	-282	-85	-195	0	148	254	228	-153	-178	-128
2017-08-28 15:10:00	stream:tcp	173	166	79	18	376	132	208	-203	41	-35	0	46	129	154	-37	-38	-17
2017-08-28 15:10:00	stream:udp	122	36	1	2	244	211	206	-122	-89	-84	0	74	69	60	-86	-88	-82
2017-08-28 15:05:00	stream:dns	158	380	21	39	614	217	266	-456	-59	-108	0	68	82	122	-76	-80	-69
2017-08-28 15:05:00	stream:ip	326	277	129	48	755	375	502	-429	-49	-176	0	162	254	258	-152	-178	-128
2017-08-28 15:05:00	stream:tcp	138	170	42	37	344	134	192	-206	4	-54	0	46	129	173	-36	-38	-17
2017-08-28 15:05:00	stream:udp	125	40	34	2	251	178	205	-126	-53	-80	0	78	69	60	-86	-87	-82
2017-08-28 15:00:00	stream:arp	2	0	12	8	2	16	10	0	-14	-8	0	2	2	5	0	-2	0
2017-08-28 15:00:00	stream:dns	274	61	65	130	289	289	213	-15	-15	61	0	64	82	127	-76	-80	-69
2017-08-28 15:00:00	stream:http	50	47	37	48	5	19	6	45	31	44	0	47	32	5	-2	-6	-4



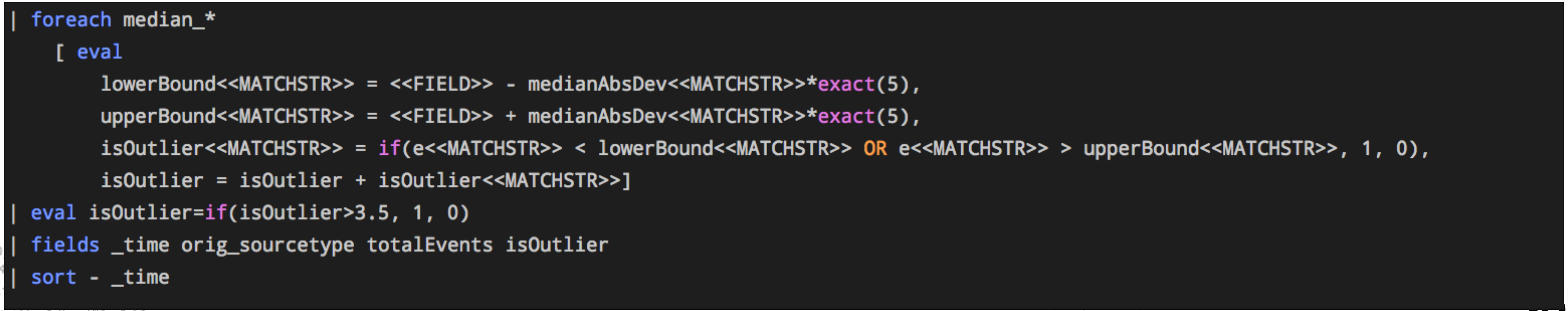
```
| foreach median_*
    [ eval absDev<<MATCHSTR>> = abs(e<<MATCHSTR>> - <<FIELD>>)]
| streamstats window=12 median(absDev*) as medianAbsDev* by orig_sourcetype
| eval isOutlier = 0
```

Vote

Which one of these things, is not like the others?



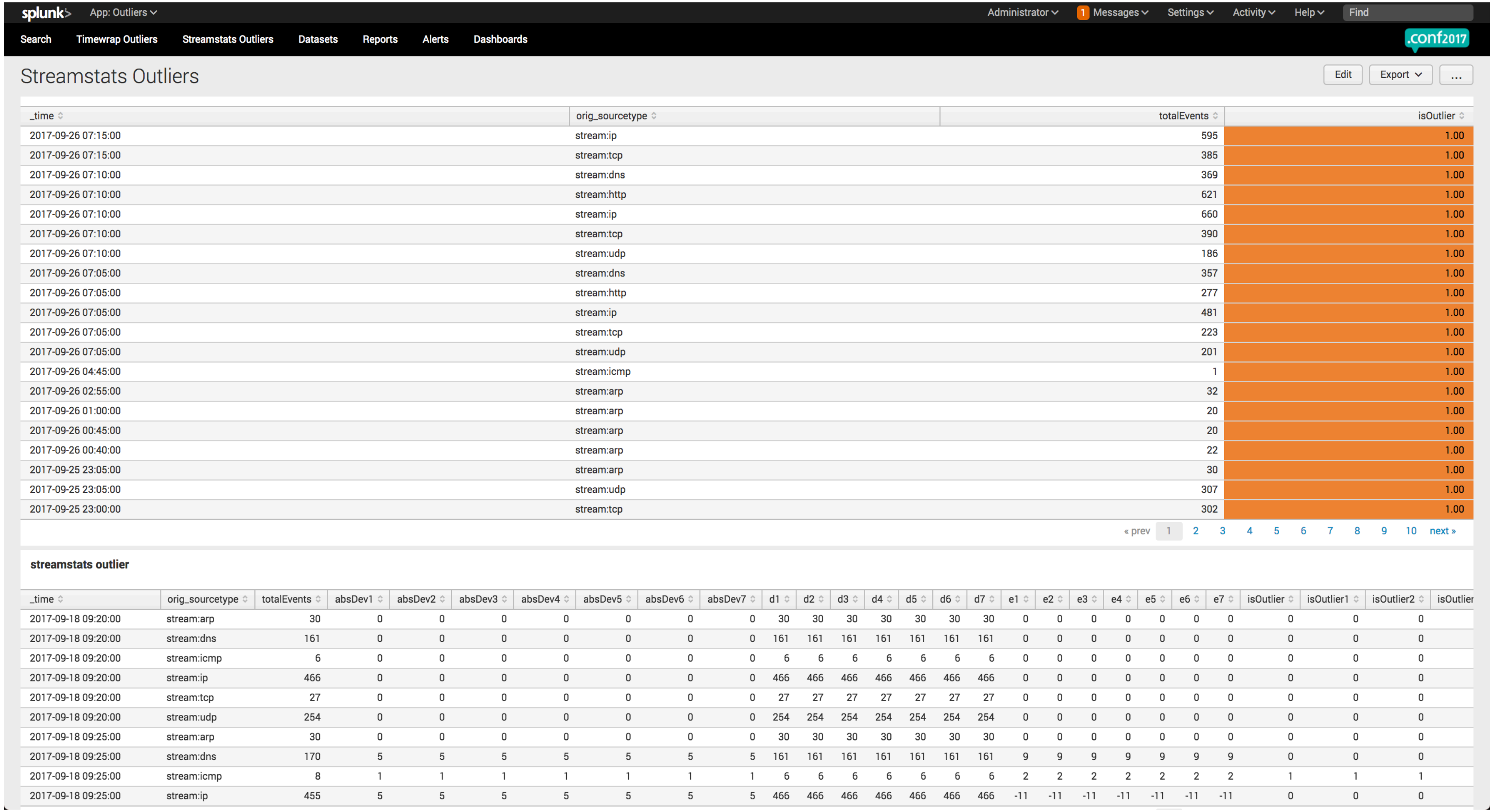
_time	orig_sourcetype	totalEvents	isOutlier
2017-09-26 00:55:00	stream:arp	22	0
2017-09-26 00:55:00	stream:dns	119	1
2017-09-26 00:55:00	stream:ip	194	0
2017-09-26 00:55:00	stream:tcp	63	0
2017-09-26 00:55:00	stream:udp	93	0
2017-09-26 00:50:00	stream:arp	20	0
2017-09-26 00:50:00	stream:dns	131	0
2017-09-26 00:50:00	stream:ip	164	0
2017-09-26 00:50:00	stream:tcp	38	0
2017-09-26 00:50:00	stream:udp	94	0
2017-09-26 00:45:00	stream:arp	20	0
2017-09-26 00:45:00	stream:dns	116	0
2017-09-26 00:45:00	stream:ip	173	0
2017-09-26 00:45:00	stream:tcp	53	0
2017-09-26 00:45:00	stream:udp	91	0
2017-09-26 00:40:00	stream:arp	22	1
2017-09-26 00:40:00	stream:dns	109	0



```
| foreach median_*
    [ eval
        lowerBound<<MATCHSTR>> = <<FIELD>> - medianAbsDev<<MATCHSTR>>*exact(5),
        upperBound<<MATCHSTR>> = <<FIELD>> + medianAbsDev<<MATCHSTR>>*exact(5),
        isOutlier<<MATCHSTR>> = if(e<<MATCHSTR>> < lowerBound<<MATCHSTR>> OR e<<MATCHSTR>> > upperBound<<MATCHSTR>>, 1, 0),
        isOutlier = isOutlier + isOutlier<<MATCHSTR>>]
| eval isOutlier=if(isOutlier>3.5, 1, 0)
| fields _time orig_sourcetype totalEvents isOutlier
| sort - _time
```

Analyze/Act

Review the results and fine tune!



Streamstats Outliers

_time	orig_sourcetype	totalEvents	isOutlier
2017-09-26 07:15:00	stream:ip	595	1.00
2017-09-26 07:15:00	stream:tcp	385	1.00
2017-09-26 07:10:00	stream:dns	369	1.00
2017-09-26 07:10:00	stream:http	621	1.00
2017-09-26 07:10:00	stream:ip	660	1.00
2017-09-26 07:10:00	stream:tcp	390	1.00
2017-09-26 07:10:00	stream:udp	186	1.00
2017-09-26 07:05:00	stream:dns	357	1.00
2017-09-26 07:05:00	stream:http	277	1.00
2017-09-26 07:05:00	stream:ip	481	1.00
2017-09-26 07:05:00	stream:tcp	223	1.00
2017-09-26 07:05:00	stream:udp	201	1.00
2017-09-26 04:45:00	stream:icmp	1	1.00
2017-09-26 02:55:00	stream:arp	32	1.00
2017-09-26 01:00:00	stream:arp	20	1.00
2017-09-26 00:45:00	stream:arp	20	1.00
2017-09-26 00:40:00	stream:arp	22	1.00
2017-09-25 23:05:00	stream:arp	30	1.00
2017-09-25 23:05:00	stream:udp	307	1.00
2017-09-25 23:00:00	stream:tcp	302	1.00

« prev 1 2 3 4 5 6 7 8 9 10 next »

streamstats outlier

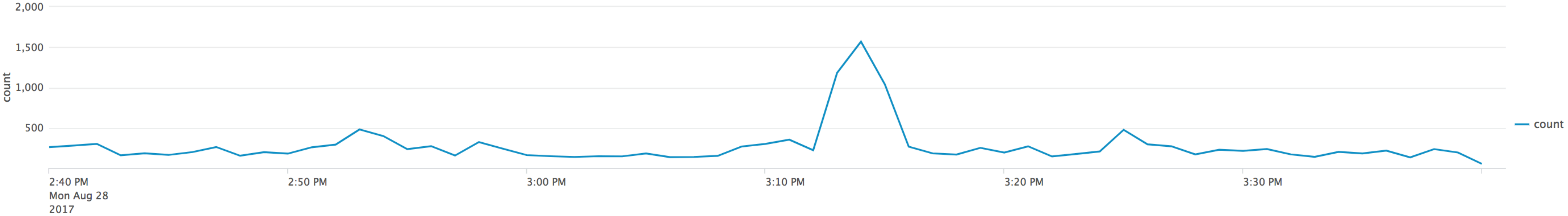
_time	orig_sourcetype	totalEvents	absDev1	absDev2	absDev3	absDev4	absDev5	absDev6	absDev7	d1	d2	d3	d4	d5	d6	d7	e1	e2	e3	e4	e5	e6	e7	isOutlier	isOutlier1	isOutlier2	isOutlier
2017-09-18 09:20:00	stream:arp	30	0	0	0	0	0	0	0	30	30	30	30	30	30	30	0	0	0	0	0	0	0	0	0	0	
2017-09-18 09:20:00	stream:dns	161	0	0	0	0	0	0	0	161	161	161	161	161	161	161	0	0	0	0	0	0	0	0	0	0	
2017-09-18 09:20:00	stream:icmp	6	0	0	0	0	0	0	0	6	6	6	6	6	6	6	0	0	0	0	0	0	0	0	0	0	
2017-09-18 09:20:00	stream:ip	466	0	0	0	0	0	0	0	466	466	466	466	466	466	466	0	0	0	0	0	0	0	0	0	0	
2017-09-18 09:20:00	stream:tcp	27	0	0	0	0	0	0	0	27	27	27	27	27	27	27	0	0	0	0	0	0	0	0	0	0	
2017-09-18 09:20:00	stream:udp	254	0	0	0	0	0	0	0	254	254	254	254	254	254	254	0	0	0	0	0	0	0	0	0	0	
2017-09-18 09:25:00	stream:arp	30	0	0	0	0	0	0	0	30	30	30	30	30	30	30	0	0	0	0	0	0	0	0	0	0	
2017-09-18 09:25:00	stream:dns	170	5	5	5	5	5	5	5	161	161	161	161	161	161	161	9	9	9	9	9	9	9	0	0	0	
2017-09-18 09:25:00	stream:icmp	8	1	1	1	1	1	1	1	6	6	6	6	6	6	6	2	2	2	2	2	2	2	1	1	1	
2017-09-18 09:25:00	stream:ip	455	5	5	5	5	5	5	5	466	466	466	466	466	466	466	-11	-11	-11	-11	-11	-11	-11	0	0	0	

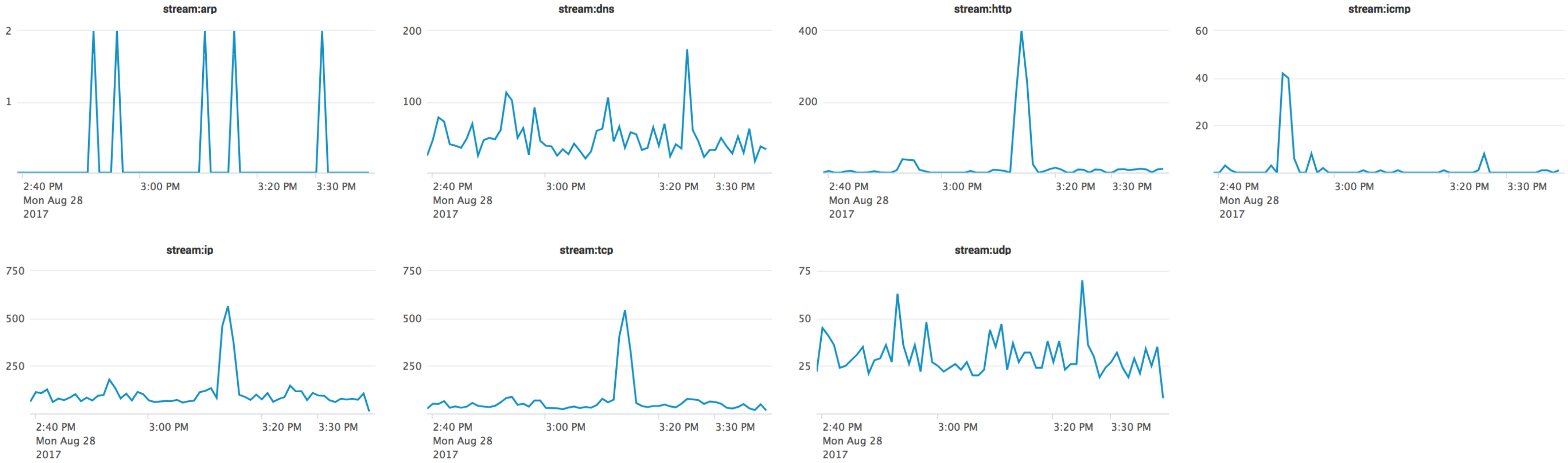
$ su splunkadmin

Watching the watcher

Monitoring Critical Data Feeds

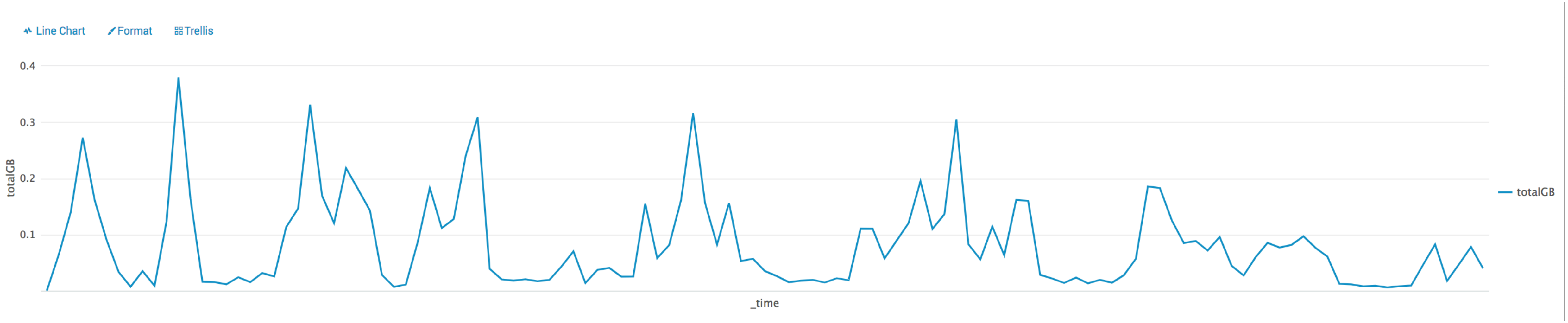
dogfood

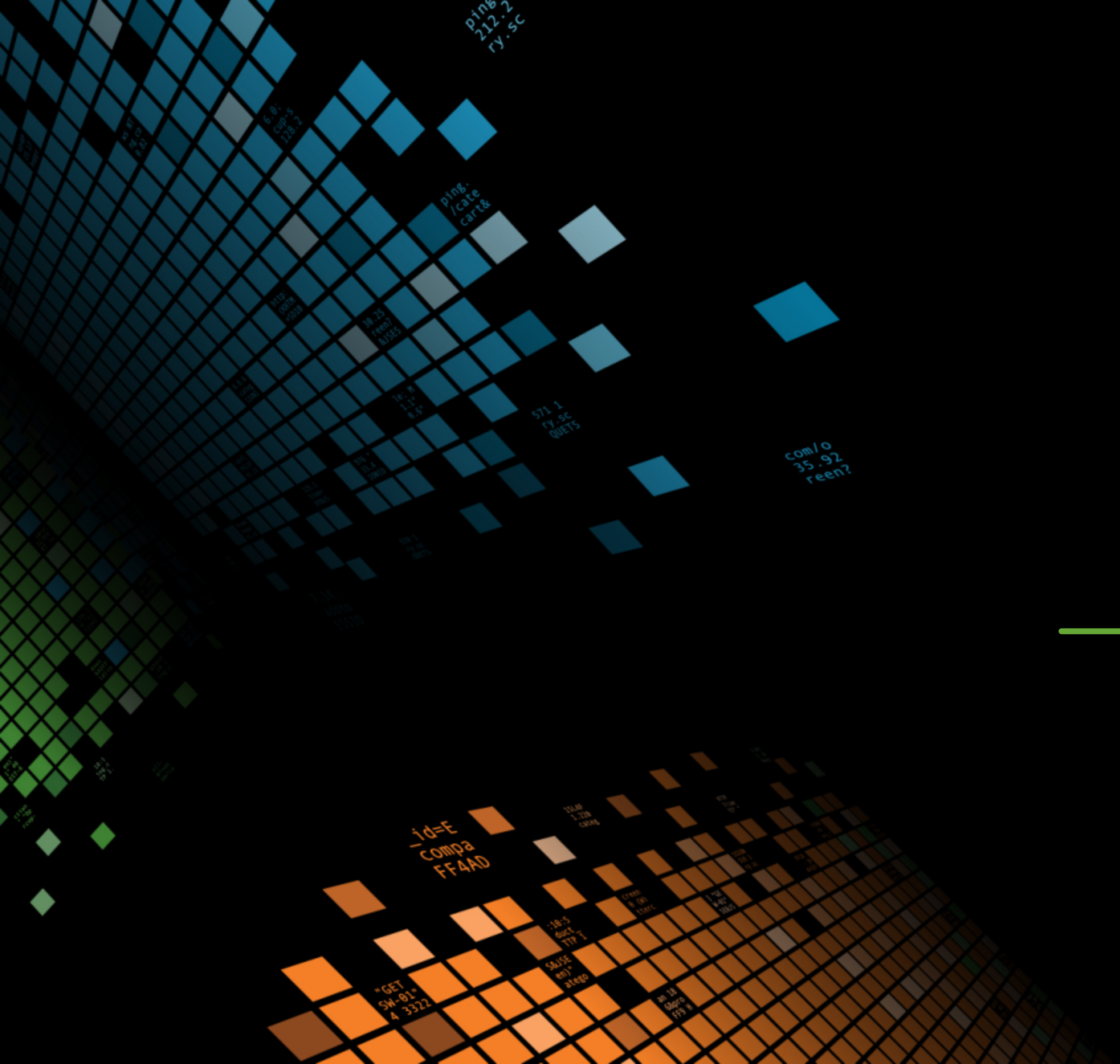




License Analytics

Monitor your indexing rate!



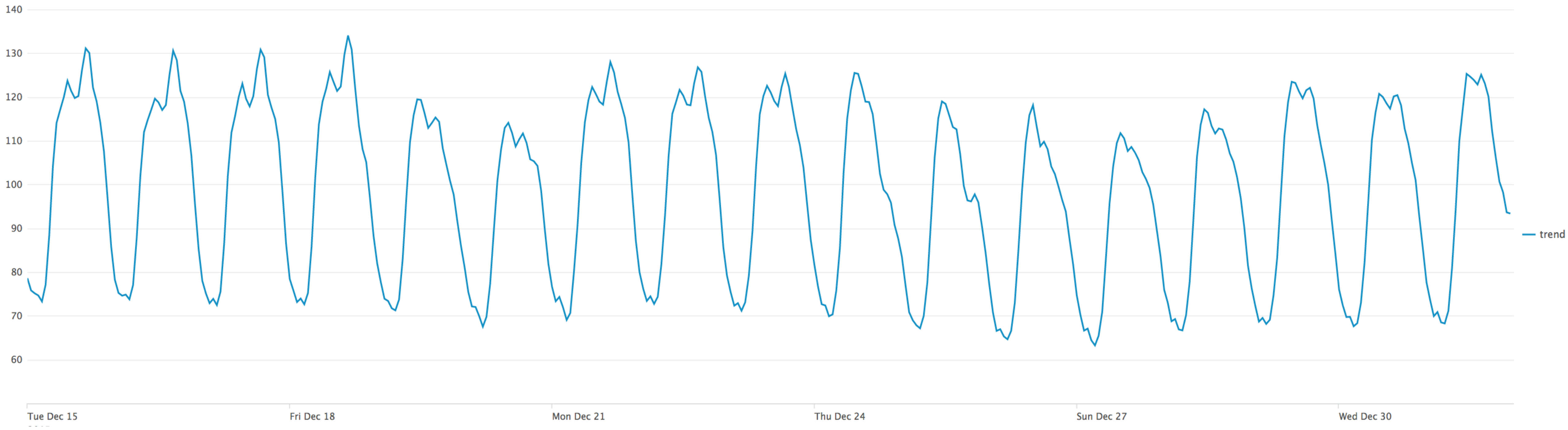


$ su netops

Outlier on the wire

Monitoring User Traffic Patterns

When humans use a system at scale, it generally looks something like this



$ ./splunk summary

In Summary…

In Summary

What did we learn?

✓ Select meaningful metrics or KPI to analyze

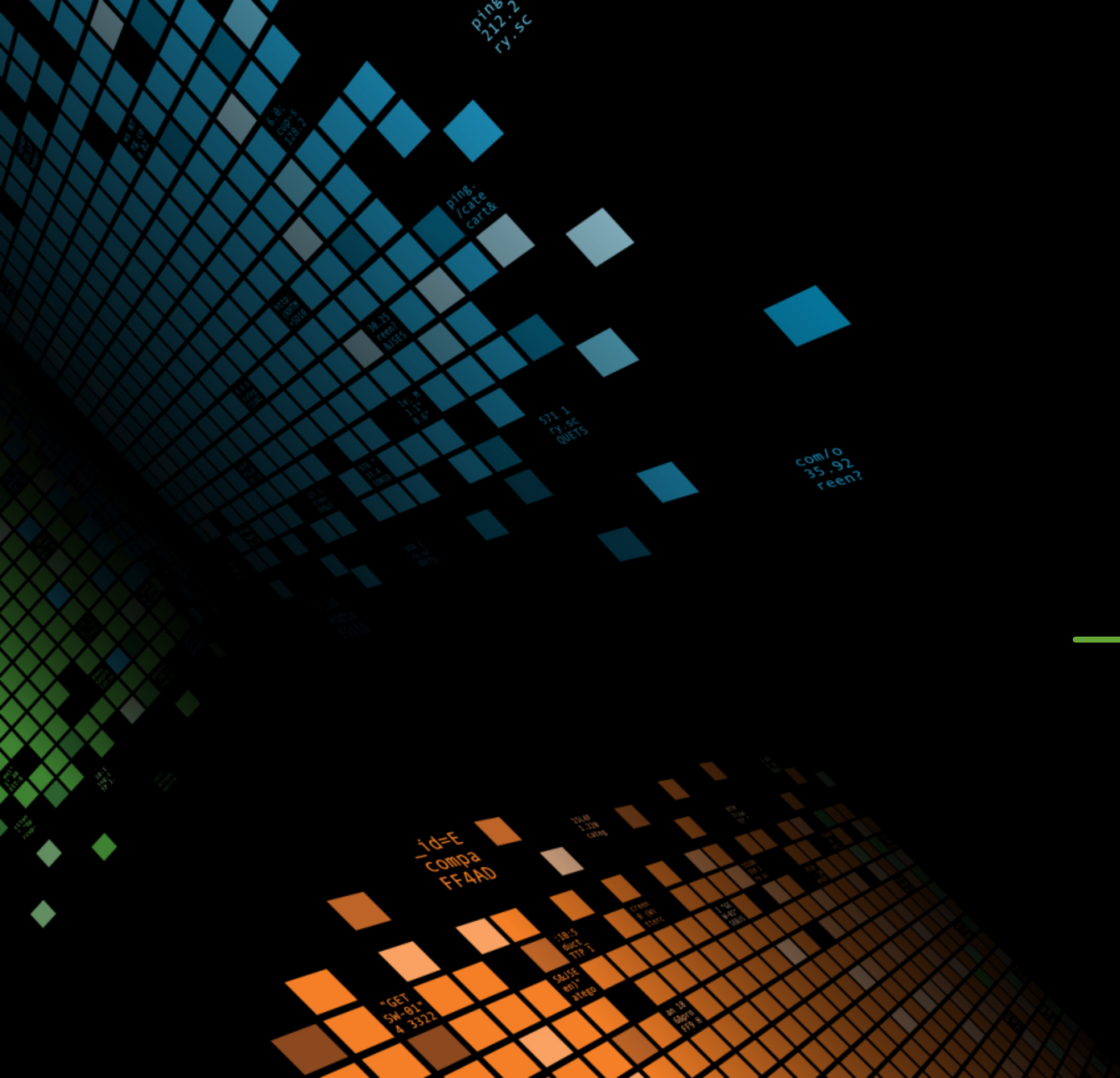
✓ Engage Decision Makers, Domain Experts & Splunk Ninjas

✓ Analyze your data over time & use MLTK to get to know your data!

✓ Use timewrap or streamstats method to baseline & identify outliers

✓ Validate, alarm, iterate!

Making machine data accessible, usable and valuable to everyone.


ping
212.2
ry.sc
ping.
/cate
cart&
571 1
ry.sc
QUETS
com/o
35.92
reen?
_id=E
compa
FF4AD
"GET
SW-01"
4 3322

Q&A

Thank You

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

splunk> .conf2017