Time After Time: Comparing Time Ranges In Splunk

Lisa Guinn

Senior Instructor, Splunk

.conf2016

splunk>

Disclaimer

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 the 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.

Introduction

.conf2016

splunk>

Agenda

- How do we use Splunk to perform time-based analysis?
  - Looking for trends
  - Looking for patterns
  - Building dynamic comparisons and alerts
- This presentation will work through progressively more complex examples

About Me

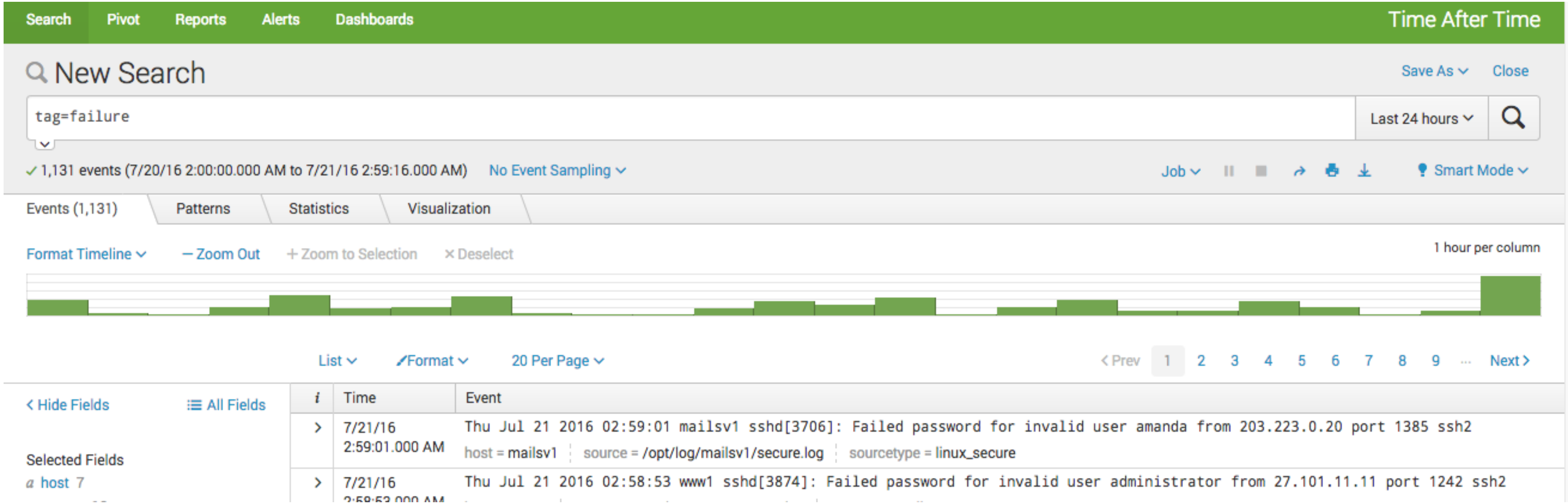
- Splunk Senior Instructor since 2009
- Passionate about solving problems with Splunk
  - #7 on Splunk Answers and proud of it!
- Has a hoodie from every .conf
  - But gave up the goal of owning every Splunk t-shirt
- Find me in the Answers Desk at .conf
  and introduce yourself!

The Searches

Basic Search

- `tag=failure`

Want to try these examples yourself?
Use `index=_internal` instead of `tag=failure`



```
tag=failure
```

✓ 1,131 events (7/20/16 2:00:00.000 AM to 7/21/16 2:59:16.000 AM)

i	Time	Event
>	7/21/16 2:59:01.000 AM	Thu Jul 21 2016 02:59:01 mailsv1 sshd[3706]: Failed password for invalid user amanda from 203.223.0.20 port 1385 ssh2 host = mailsv1 source = /opt/log/mailsv1/secure.log sourcetype = linux_secure
>	7/21/16 2:58:53.000 AM	Thu Jul 21 2016 02:58:53 www1 sshd[3874]: Failed password for invalid user administrator from 27.101.11.11 port 1242 ssh2

Time Fields in Splunk

- Time provides context for understanding events
- All events in Splunk have a timestamp
- Internal time fields

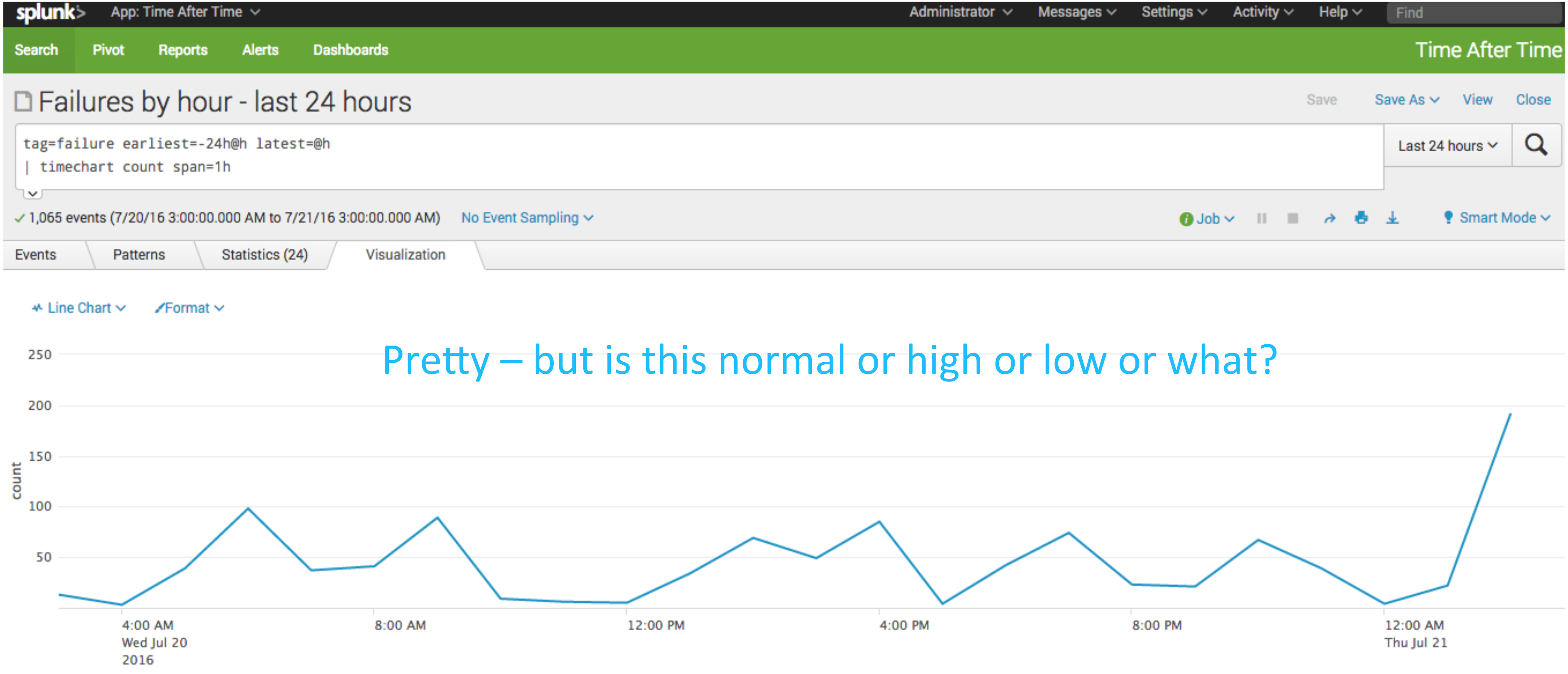
_time	event time stored in UTC
_indextime	UTC time when event was indexed

Failures Per Hour

? What is the pattern of failures over the last 24 hours?

-
  ```
  tag=failure earliest=-24h@h latest=@h
  | timechart count span=1h
  ```

Failures Per Hour Results



Failures by hour - last 24 hours

```
tag=failure earliest=-24h@h latest=@h
| timechart count span=1h
```

✓ 1,065 events (7/20/16 3:00:00.000 AM to 7/21/16 3:00:00.000 AM) No Event Sampling

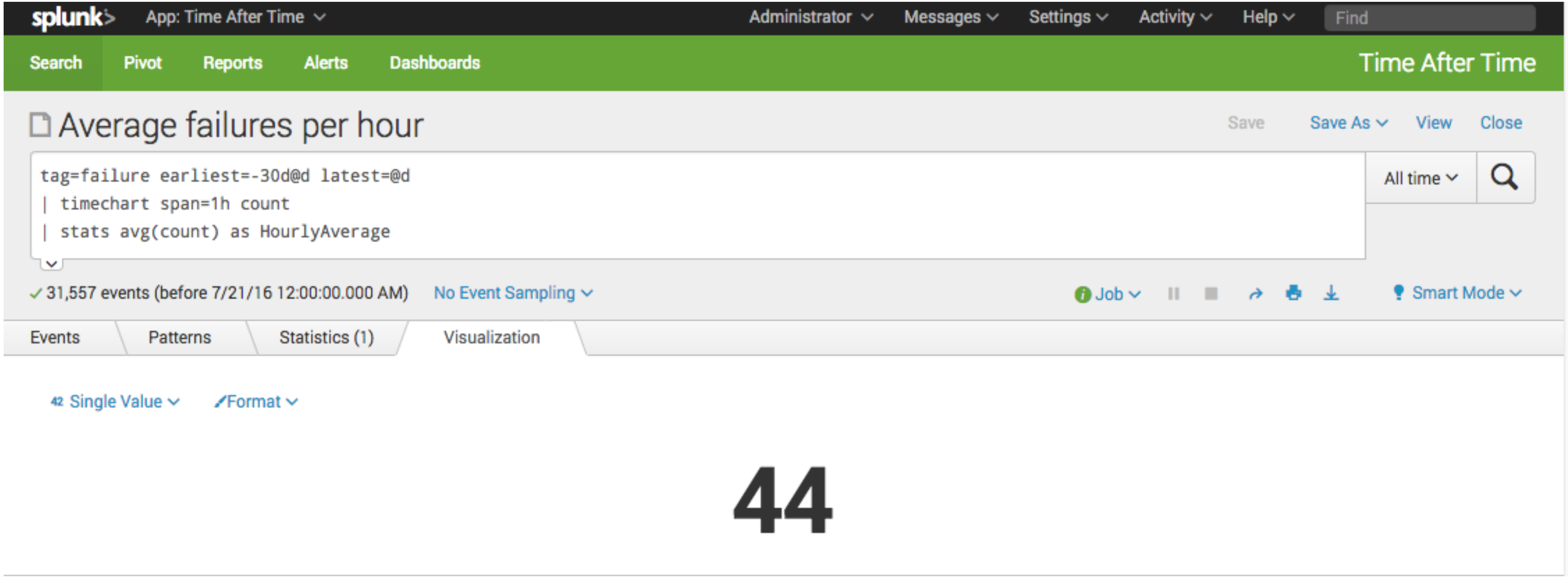
Pretty – but is this normal or high or low or what?

Average Failures Per Hour

- ? What is the average number of failures per hour, based on the last 30 days?
  - Count the number of failures per hour
  - Average the hourly count

```
tag=failure earliest=-30d@d latest=@d
| timechart span=1h count
| stats avg(count) as HourlyAverage
```

Average Failures Per Hour Results



Average failures per hour

```
tag=failure earliest=-30d@d latest=@d
| timechart span=1h count
| stats avg(count) as HourlyAverage
```

✓ 31,557 events (before 7/21/16 12:00:00.000 AM) No Event Sampling

44

Comparing The Hourly Data To The Average

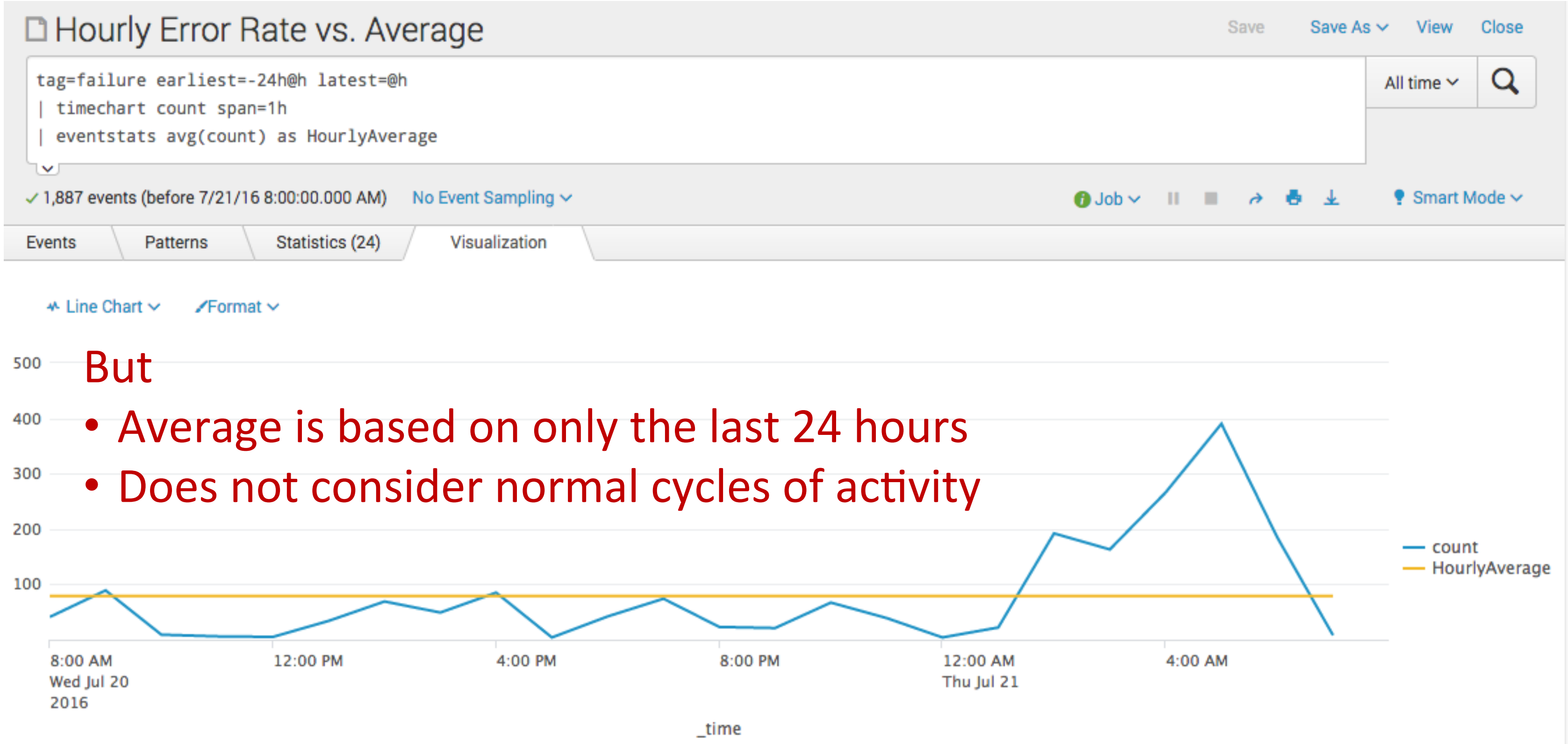
- That was cool, but how do I compare this with the hour-by-hour results?

-
  ```
  tag=failure earliest=-24h@h latest=@h
  | timechart count span=1h
  | eventstats avg(count) as HourlyAverage
  ```

hour by hour count

add a column for the average

Results Of The Comparison



```
tag=failure earliest=-24h@h latest=@h
| timechart count span=1h
| eventstats avg(count) as HourlyAverage
```

✓ 1,887 events (before 7/21/16 8:00:00.000 AM)

But

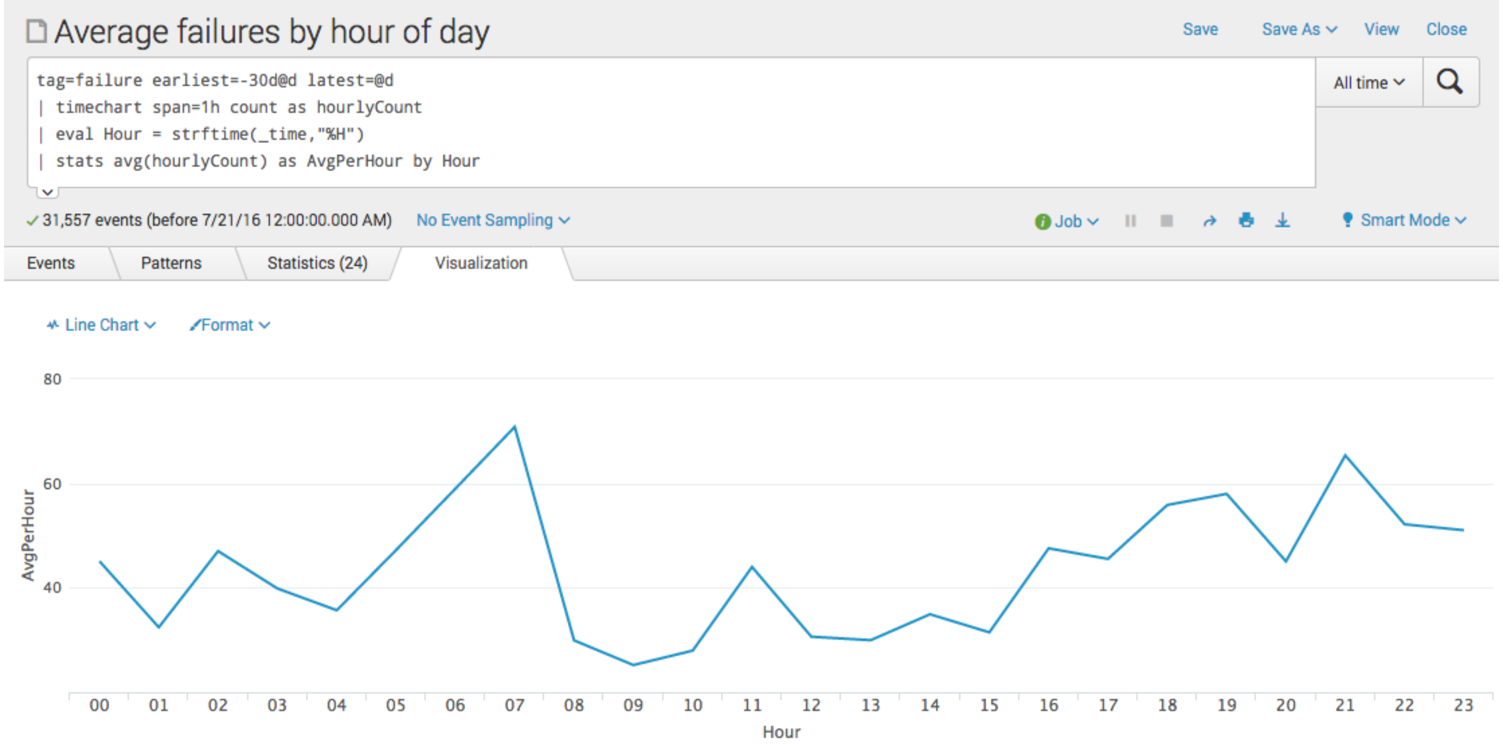
- Average is based on only the last 24 hours
- Does not consider normal cycles of activity

Average Failures By Hour Of Day

? What is the number of failures for each hour, averaged over the last 30 days? We should end up with 24 averages, one for each hour of the day.

-
  ```
  tag=failure earliest=-30d@d latest=@d
  | timechart span=1h count as hourlyCount
  | eval Hour = strftime(_time,"%H")
  | stats avg(hourlyCount) as AvgPerHour by Hour
  ```

Average Failures By Hour Of Day



```
tag=failure earliest=-30d@d latest=@d
| timechart span=1h count as hourlyCount
| eval Hour = strftime(_time,"%H")
| stats avg(hourlyCount) as AvgPerHour by Hour
```

✓31,557 events (before 7/21/16 12:00:00.000 AM)

Compare The Last 24 Hrs With The Average Of The Last 30 Days

? Now that we have the average from the last search, how do we compare it with what is happening today?

- One solution: add in the last 24 hours as a subsearch

-
  ```
  tag=failure earliest=-30d@d latest=@d
    | timechart span=1h count as hourlyCount
    | eval Hour = strftime(_time,"%H")
    | stats avg(hourlyCount) as AvgPerHour by Hour
    | join Hour
      [ search tag=failure earliest=-24h@h latest=@h
      | timechart span=1h count as hourlyCount
      | eval Hour = strftime(_time,"%H") ]
  ```

Limitations Of Subsearches

- Join limitations
  - Maximum number of results = 50,000
  - Maximum subsearch run time = 60 seconds
  - Other types of subsearches have similar limitations
- It is inefficient to traverse the same data twice
- Solution
  - Traverse the data only once
  - Use `eval` command to categorize the event as current or historical for the calculations

Without A Join, It's Not Really Harder…

- Compare the last 24 hours with the average of the last 30 days

-
  ```
  tag=failure earliest=-30d latest=@d
  | eval StartTime=relative_time(now(),"-1d@d")
  | eval Series=if(_time>=StartTime,"Today","Average")
  | eval Hour = strftime(_time,"%H")
  | stats count by Hour Series
  | chart avg(count) by Hour Series
  ```

create categories

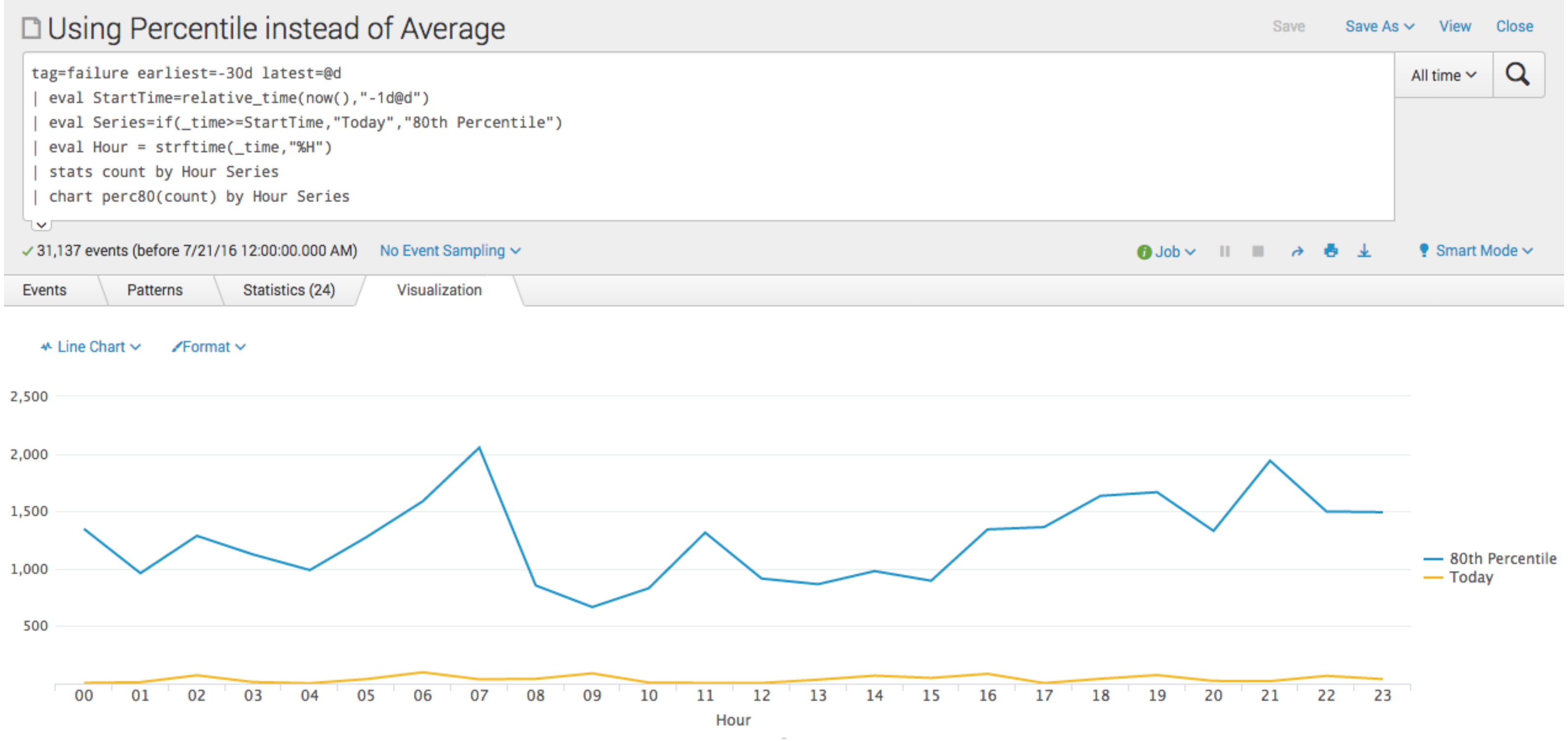
use stats and chart (instead of timechart) to aggregate properly

Alerts And Averages

- Do you want to alert on the **average** or on the **unusual**?
- Use **perc** instead of **avg**

-
  ```
  tag=failure earliest=-30d latest=@d
  | eval StartTime=relative_time(now(),"-1d@d")
  | eval Series=
        if(_time>=StartTime,"Today","80th Percentile")
  | eval Hour = strftime(_time,"%H")
  | stats count by Hour Series
  | chart perc80(count) by Hour Series
  ```

80th Percentile Results



Using Percentile instead of Average

```
tag=failure earliest=-30d latest=@d
| eval StartTime=relative_time(now(),"-1d@d")
| eval Series=if(_time>=StartTime,"Today","80th Percentile")
| eval Hour = strftime(_time,"%H")
| stats count by Hour Series
| chart perc80(count) by Hour Series
```

✓31,137 events (before 7/21/16 12:00:00.000 AM)

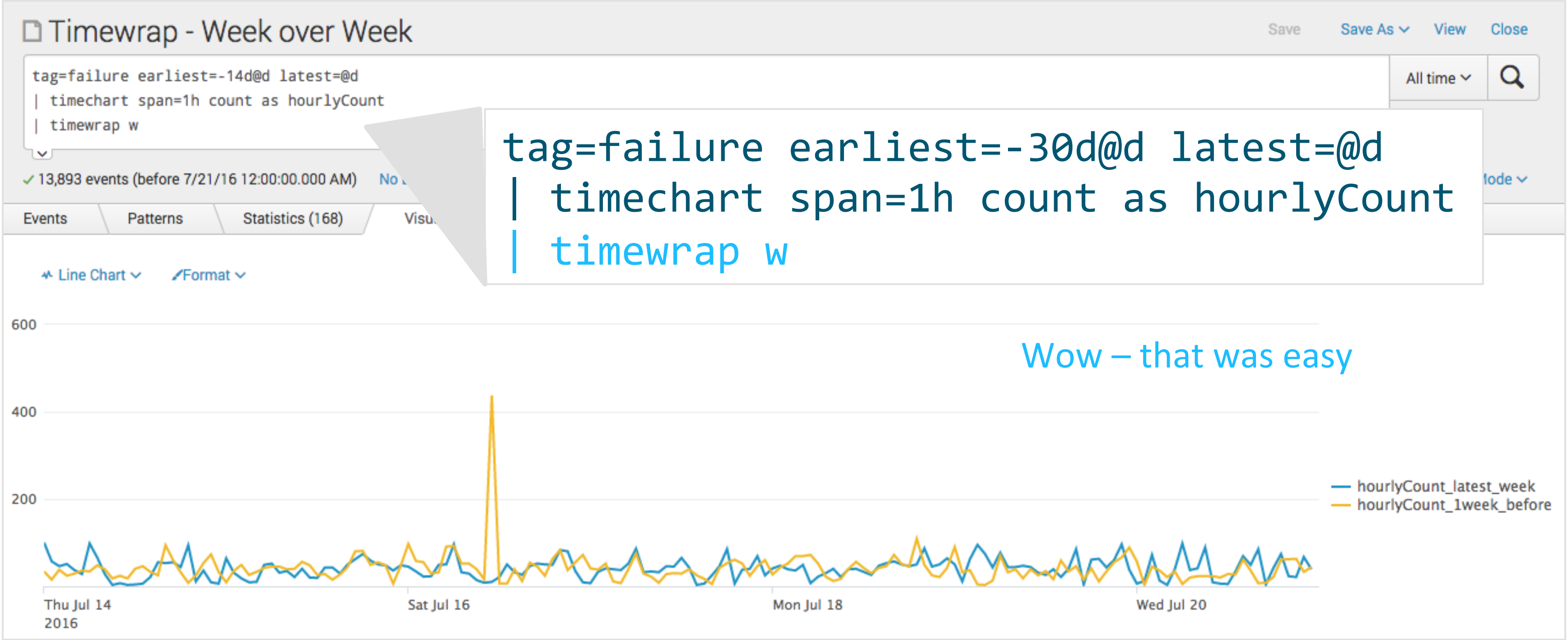
Timewrap

Making your life easier since 2013

- Free app to make time-based reporting easier
  - Works on Splunk 6.4
  - Community supported
- Provides a command **`timewrap`** that works with **`timechart`**
- Details on Splunkbase

  https://splunkbase.splunk.com/app/1645/

Comparing Two Weeks With Timewrap



```
tag=failure earliest=-14d@d latest=@d
| timechart span=1h count as hourlyCount
| timewrap w
```

✓ 13,893 events (before 7/21/16 12:00:00.000 AM)

```
tag=failure earliest=-30d@d latest=@d
| timechart span=1h count as hourlyCount
| timewrap w
```

Wow – that was easy

What Next?

- Ask questions here for a few minutes!
- Visit the Answers Desk and we can work through specific searches
  - I love being stumped by tough questions
  - If not here at .conf, on http://answers.splunk.com

- Other Sessions
  - Splunk Data Collection Best Practices (Wednesday 1:00 pm)
    - Because time-based comparisons depend on good timestamps!
  - Quis Custodiet Ipsos Custodes? (Who watches the watchmen?) OR How do you know when Splunk stops searching? (Tuesday 3:15 pm)

THANK YOU

.conf2016

splunk>