# A Mathematical Perspective on Data Science

Dr. Tom LaGatta
Staff Sales Engineer
(previously Staff Data Scientist)

# About Me

- Math PhD from University of Arizona
  - "Geodesics of Random Riemannian Metrics" w/ Janek Wehr
  - Probability + Differential Geometry + Functional Analysis
- Postdoc at Courant Institute @ NYU
  - Finished Geodesics paper, published in Communications in Math. Physics
  - Collaborated with Political Scientists on heterogeneous voting behavior
- Was: Staff Data Scientist at Splunk
  - Helped customers with advanced use cases in Business Analytics, Internet of Things, Machine Learning, Data Science
- Now: Staff Sales Engineer at Splunk
  - Helping big big customers solve big big business problems

# Abstract


- As with all things, the process of analyzing data admits a mathematical description. As a mathematician-turned-data-scientist, I will describe my approach to problem solving, and attempt to loosely formalize "stakeholders", "use cases", "data" and "deliverables" in mathematical language for the enjoyment of this mostly academic audience. In particular, I will describe how query languages are inherently functional, acting as functional transformations of Data into Data, which obey the usual functional composition law. The process of analyzing data results in an iterative sequence of queries, converging to a final query which is satisfactory to the use case. These queries are then organized into deliverables, which can be "dashboards" (web pages with visualizations) or "data products" (with scheduled jobs & analyses running in the background). When this process is done right, it results in the extraction of "value" for stakeholders, which can be measured tangibly in terms of revenue, costs or risk metrics. Sometimes this has a fancy name like "data science", but more often than not, is just the normal operational work of a good data-savvy IT, Security, Tech or Business department in modern enterprises and governmental agencies. There will be no proofs, but I would be very interested to discuss rigorous approaches to social organization & problem solving after the talk.

# Agenda

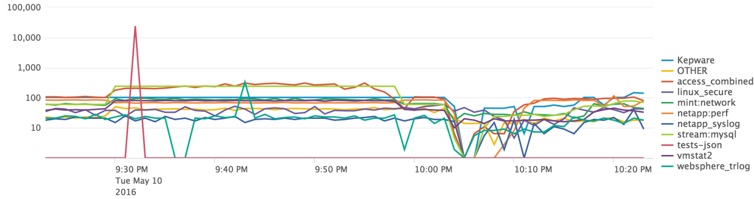
- Basic Definitions:
  - Data, Stakeholders, Use Cases, Deliverables
- Query languages as functional programming
  - Every query is a map f: Data -> Data
  - Example problem solving
- Putting It All Together: Doing Data Science
  - Emphasize actionable insights
  - Tie it together to deliver ”value” to stakeholders

# Basic Definitions

# What is Data?

- "Data" is any informational artifact of real-world phenomena
- A "metric" or "KPI" is any aggregate function of low-level data
- Examples:
  - Semi-structured timestamped events/metrics
  - Structured relational data (rows & columns)
  - Graph data (nodes & edges)
  - "Unstructured" data (images, video, text)

```
199.70.25.81 - - [10/May/2016 22:18:23:078770] "GET /oldlink?item_id=MCF-3&J
SESSIONID=SD4SCL2FF6ADFF2 HTTP 1.1" 404 948 "http://shop.acme.com/cart.do?ac
tion=remove&item_id=MCF-3&product_id=MCF-3" "mozilla/5.0 (Linux; U; Android
2.3.4; en-us; SonyEricssonLT15i Build/4.0.2.A.0.42) AppleWebKit/533.1 (KHTM
L, like Gecko) Version/4.0 Mobile Safari/533.1" 3683
```




- How to model data:
  - Events: marked point processes & time series (Skorokhod space)
  - Relational schema: categories (see David Spivak's work)
  - Other data: depends on the use case, might need new data structures to represent it (incl. vectors, graphs, etc.)

# What is a Stakeholder?

- A ”stakeholder” is a person, group or organization who is invested in the outcome of an initiative.
- Example: A company buys software
  - Stakeholder orgs are IT & the Business
  - Individual stakeholders include Individual Contributors, Managers, Directors & Executives.

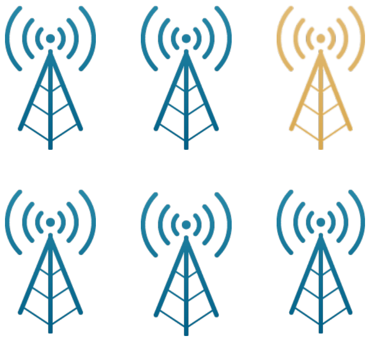
- IT stakeholders have performance metrics (num. outages, mean time to resolution, etc)
- Business stakeholders have different metrics (revenue, cost, risk)
- Customer stakeholders downstream also have value/impact metrics

# What is a Stakeholder? (cont.)

- How to model stakeholders?
  - Follow Game Theory for inspiration (but don't worry about "equilibrium")
  - Create an index set I with all stakeholders. Various actions, outcomes & objectives will have subscripts i based on stakeholders
  - E.g., person i chooses action $a_{i,t}$ at time t
  - I can be hierarchical (Person i contained in org A, so parent(i) = A)
  - "Value" modeled by objective functions ( $U_{i,n}$ = objective #n for person i)
  - Is the action "pivotal" for the outcome? (ie $\mathbb{E}[U|do(a)] > \mathbb{E}[U|do(not\ a)]$ ?)
- Keep track of stakeholders data:
  - Might be high-level (email, Powerpoints) – context is key
  - Might be in databases (transactions, customer records, tickets data)
  - Might be granular events data (web clickstream, logs, mobile, wire data)

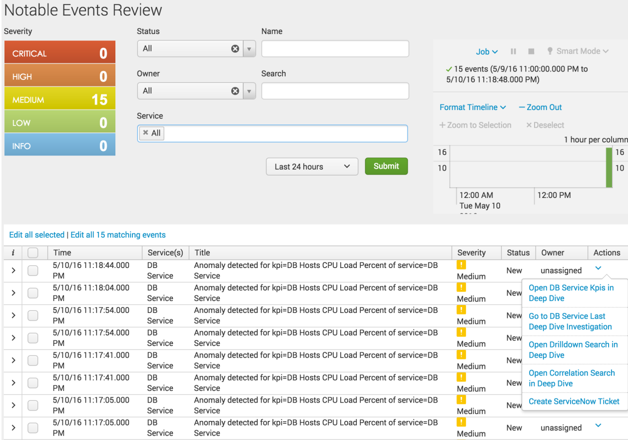
# What is a Use Case?

- A "use case" consists of a business problem, a strategy to alleviate the problem, metrics to evaluate the outcome, data to power a solution, and stakeholders who are involved in its development.
- Use Case: Problem Forecasting.
  - Company has costly network/system outages
  - IT hires a Data Scientist to help build solution.
  - Data includes Infrastructure (CPU, Memory), Operations (Outage Reports), App logs, etc
  - Metric: cost of outages ≈ num outages * time to resolution * cost of labor
  - Stakeholders include IT & Business, and impacted customers



# What is a Deliverable?

- A ”deliverable” is a thing produced to solve a use case
  - Can include ”dashboards”: informational web pages built from data queries
  - Or ”workflows”: notable events deliberated to operations analysts
  - Or full-fledged ”data product”: application which *does* stuff automatically
- Deliverable: Problem Forecasting System
  - Goal: forecast problems before they start, deliver ”proximate root cause” to IT to investigate
  - Data: CPU, Memory, Latency, Service Tickets
  - Build machine learning model to correlate Infrastructure data with Service impact
  - Apply model to incoming events, create ”predicted(Risk_Score)”
  - Surface high-risk events to IT Operations



# Queries as Functional Programming

# Query Languages

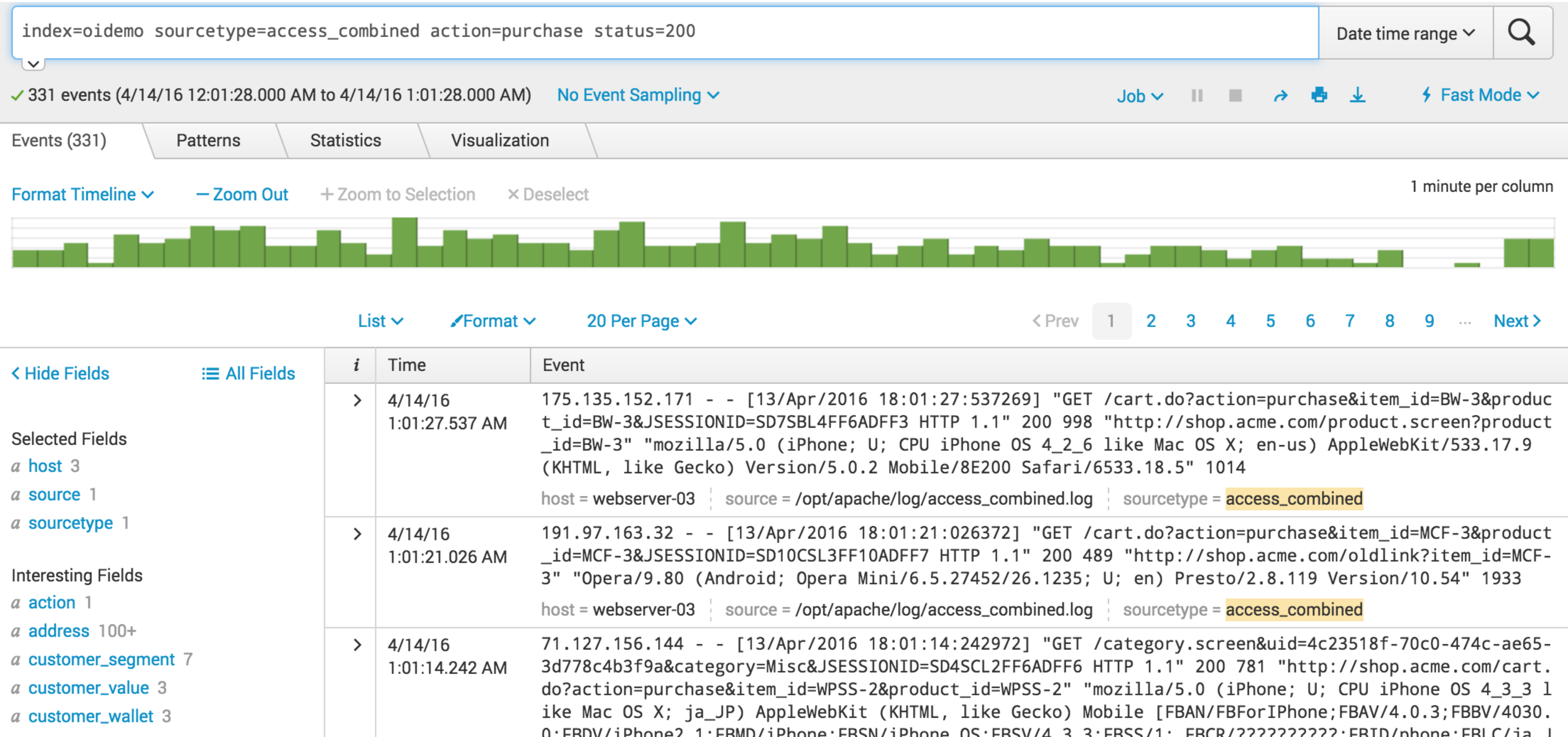
- Query languages provide a formulaic approach to working with data
- A ”query” is a string which tells where to get the data, what to do with the data, and where to put the data (incl visualization or DB)
- Mathematically, every query is a FUNCTION f : Data -> Data
- Queries can be composed (with | symbol), and analysis is iterative
- Example 1: Successful Purchase Actions from Web Logs

```
sourcetype=access_combined action=purchase status=200
```

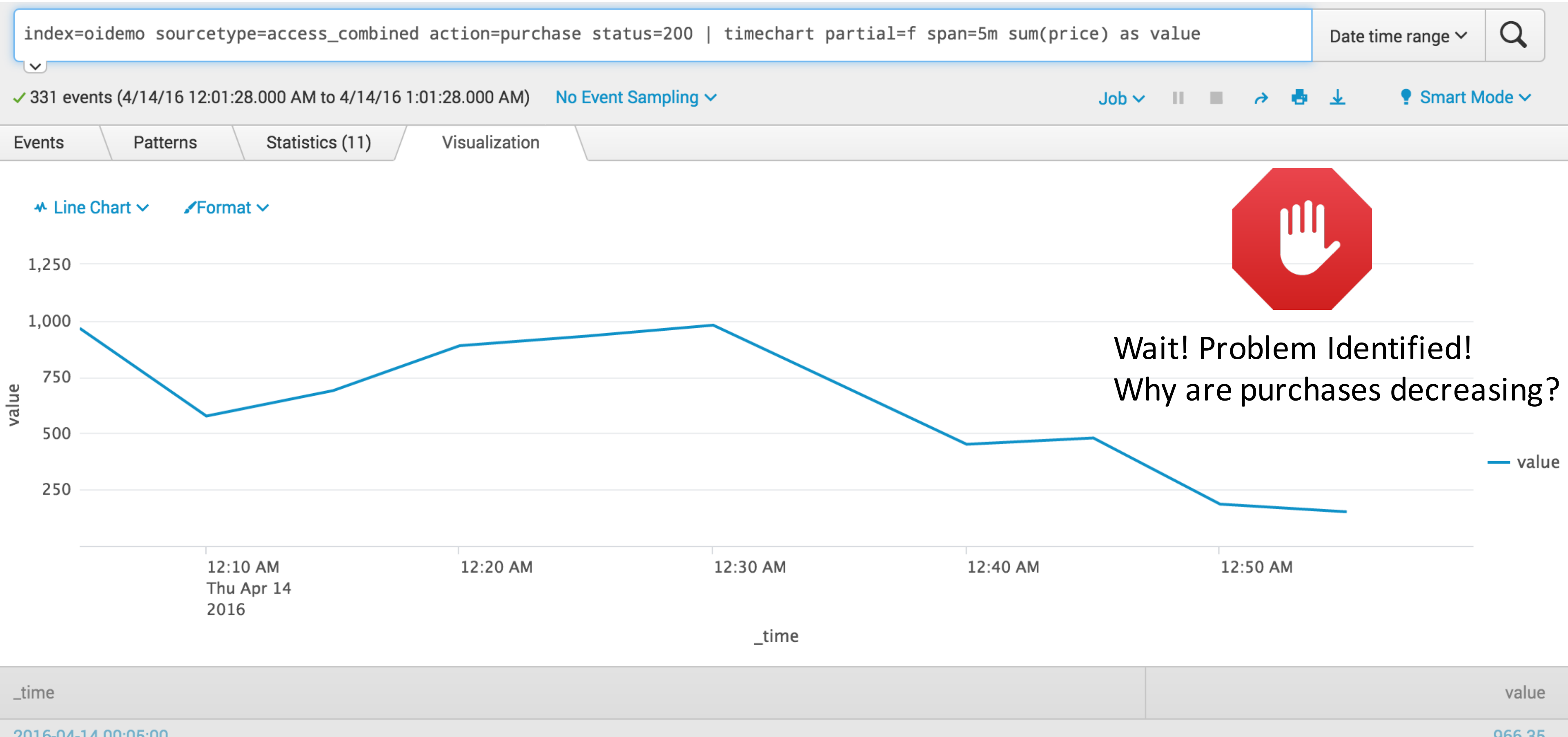
- Example 2: Plot Purchase Value as Metric Timeseries

```
sourcetype=access_combined action=purchase status=200
| timechart partial=f span=5m sum(price) as value
```

# 1: Successful Purchase Actions from Web Logs



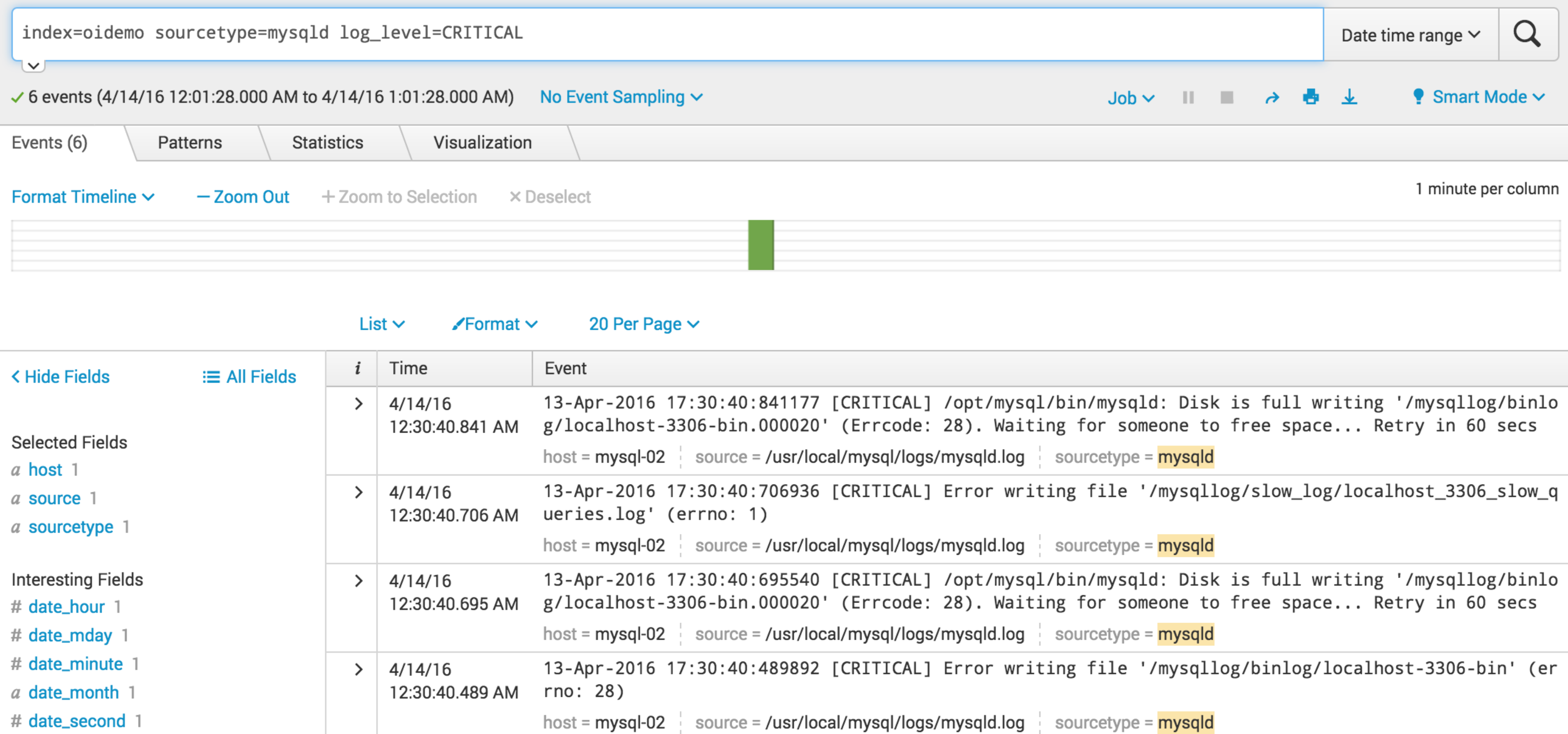
# 2: Plot Purchase Value as Metric Timeseries



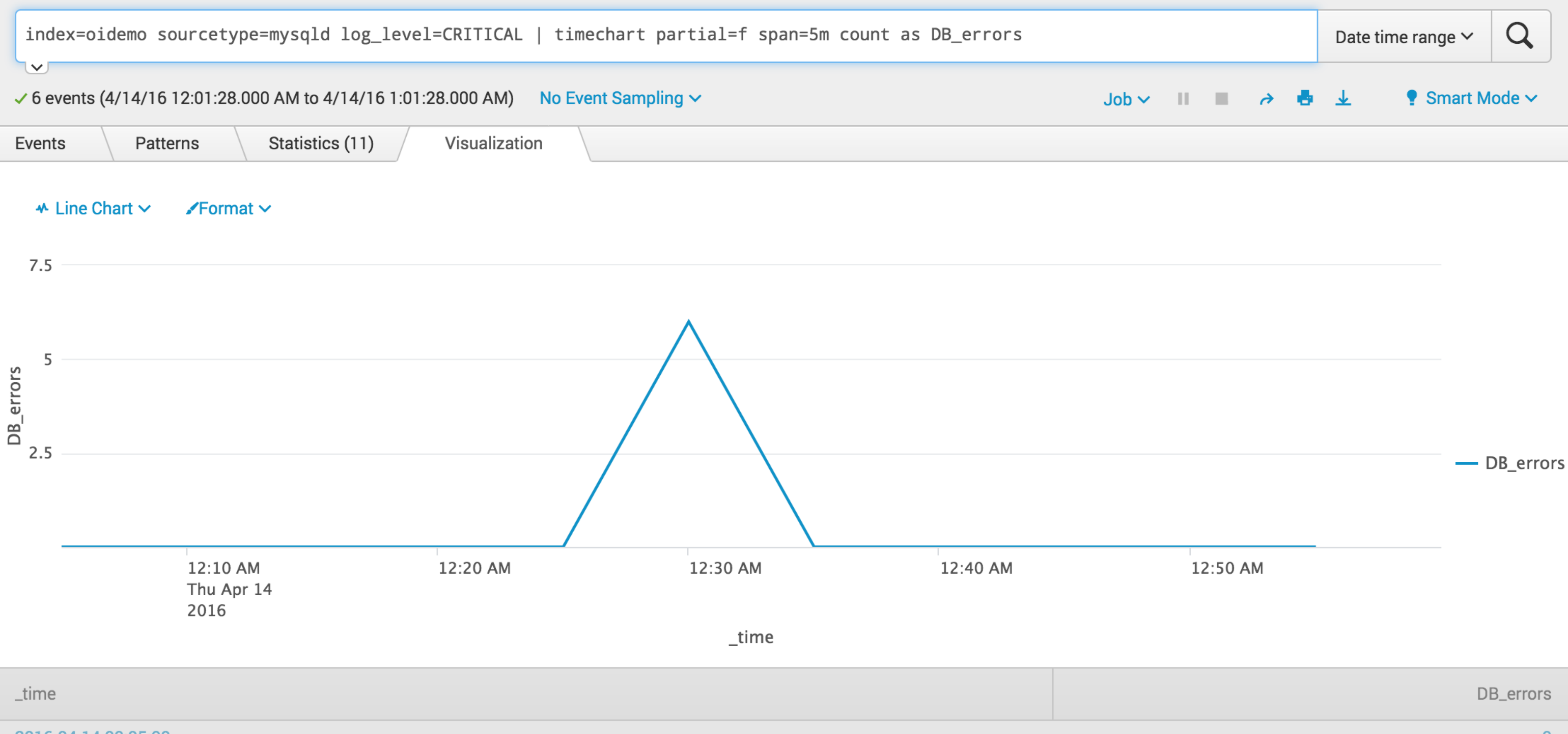

Wait! Problem Identified!

Why are purchases decreasing?

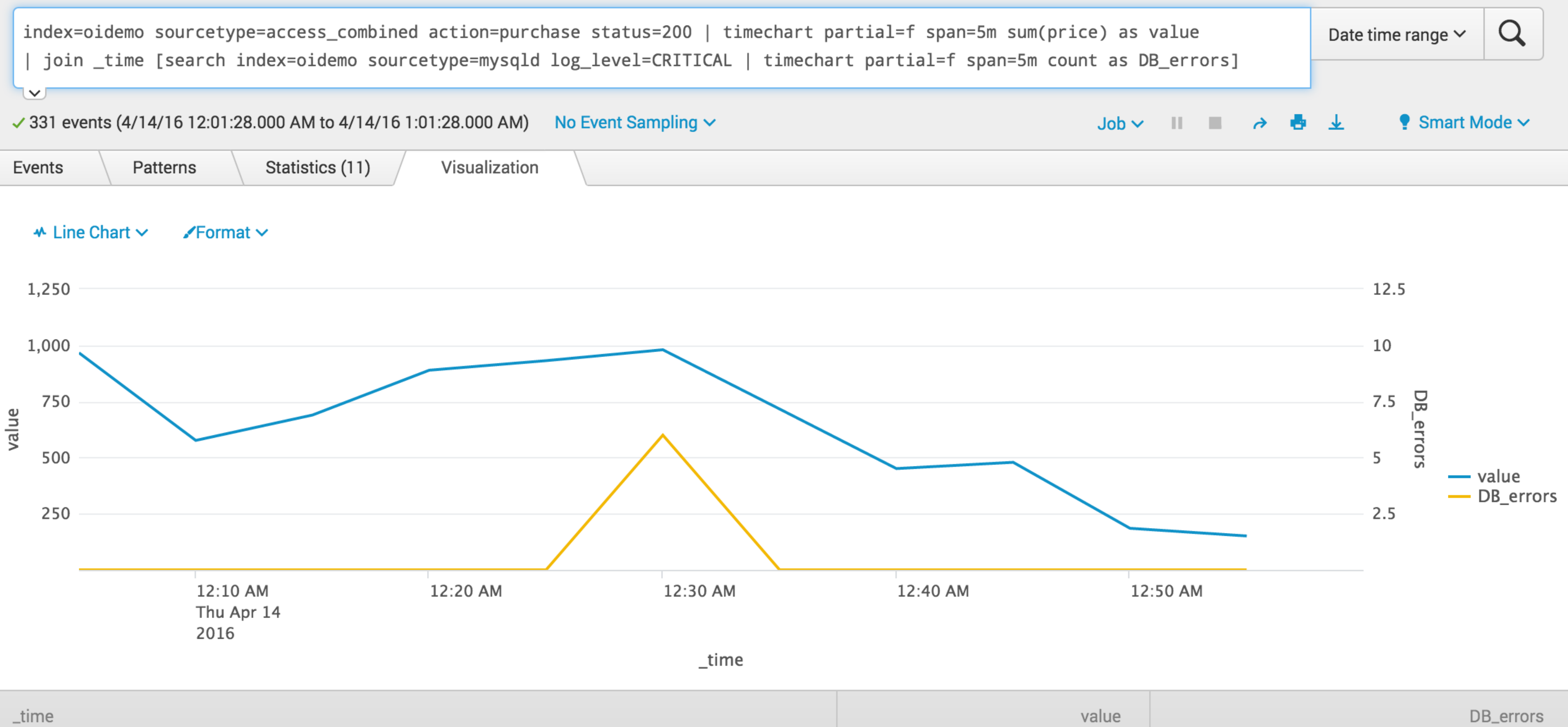
# 3: Investigate Database Errors



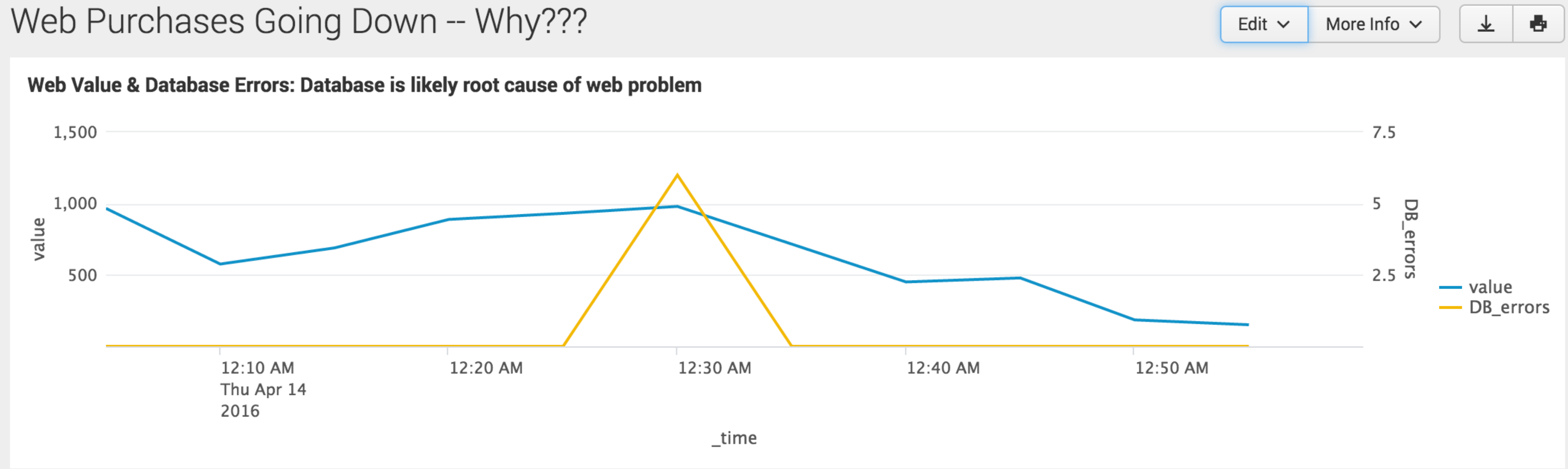
# 4: Plot Database Errors



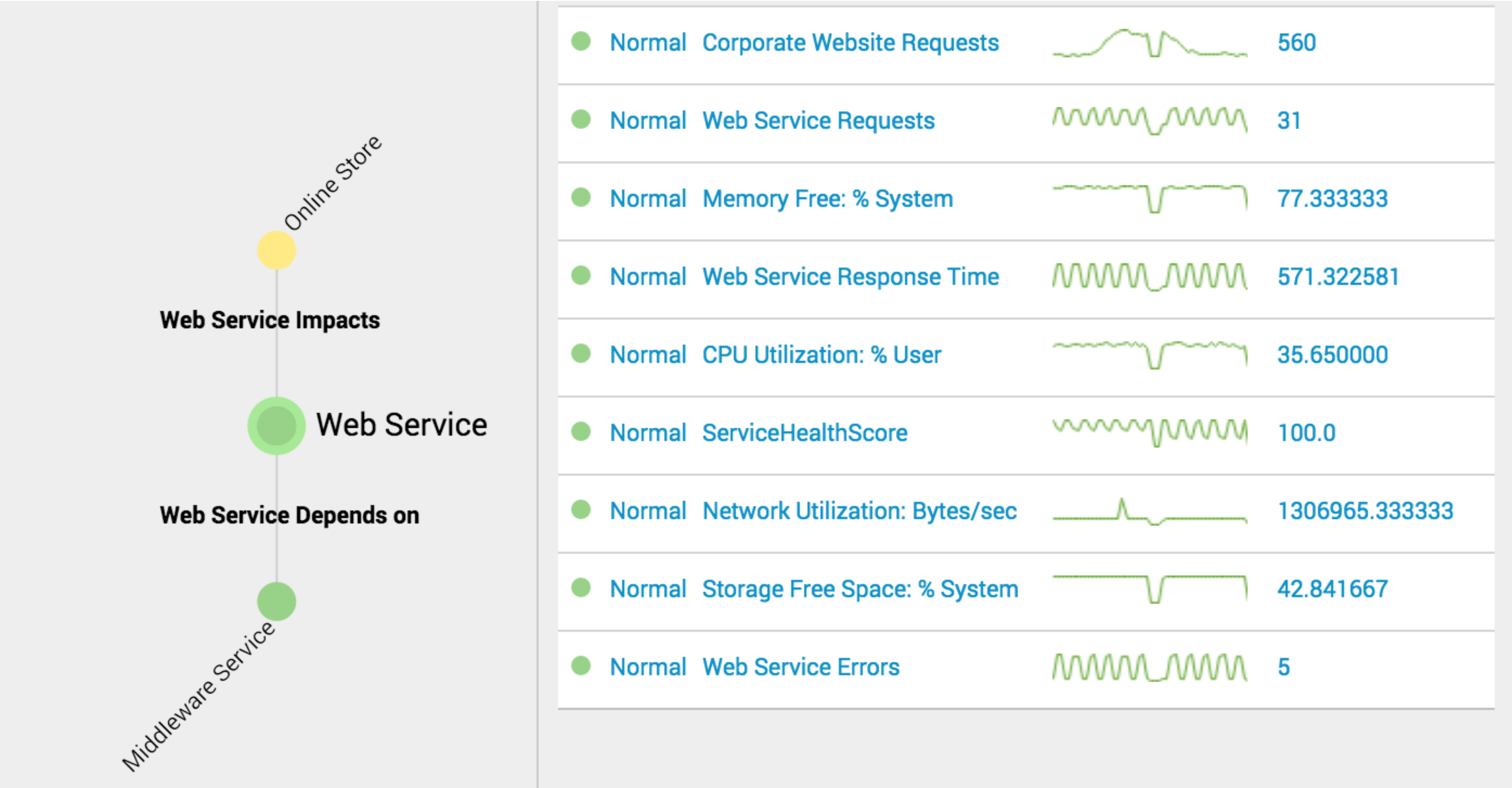
# 5: Correlate DB problems with purchase value



# 6: Save as Deliverable, Send to Stakeholders



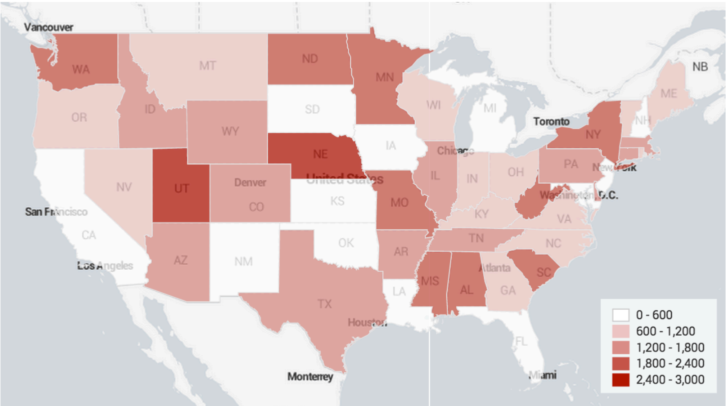
# 7: Move toward proactive monitoring stance



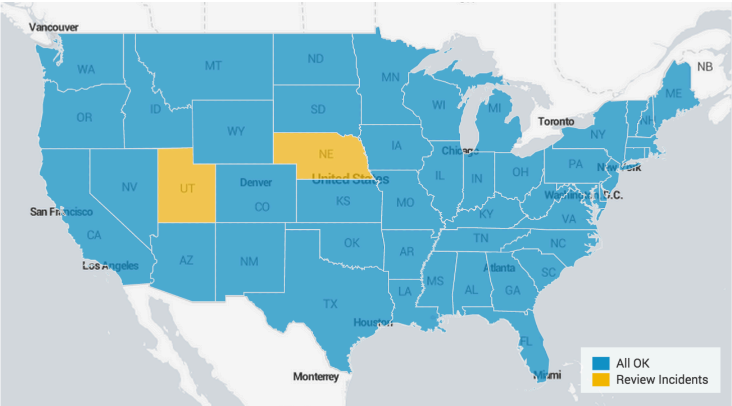
# Putting It All Together: Doing Data Science

# Emphasize Actionable Insights

- Avoid eye-candy visualizations
  - “Laser beam” threat dashboards look cool but are useless
- “How does this help me solve my problem?”
- Guide the viewer to drilldown & act quickly




Confusing viz: not actionable




Good viz: actionable

# Doing Data Science

- Data Scientists resist easy characterization. A bit of:
  - Statistician
  - Software Engineer
  - Business Analyst
  - Spock on the bridge

- Many scales of action:
  - Get your hands dirty when needed
  - But step back and see the big picture
  - Emphasize ”actionable insights” throughout the whole org
  - Also have political credibility to say, ”This is a bad decision, don’t do it”
- Data Scientists guide organizations through building data products to solve big problems and deliver value