What Users Don't Expect about Exploratory Data Analysis on Approximate Query Processing Systems

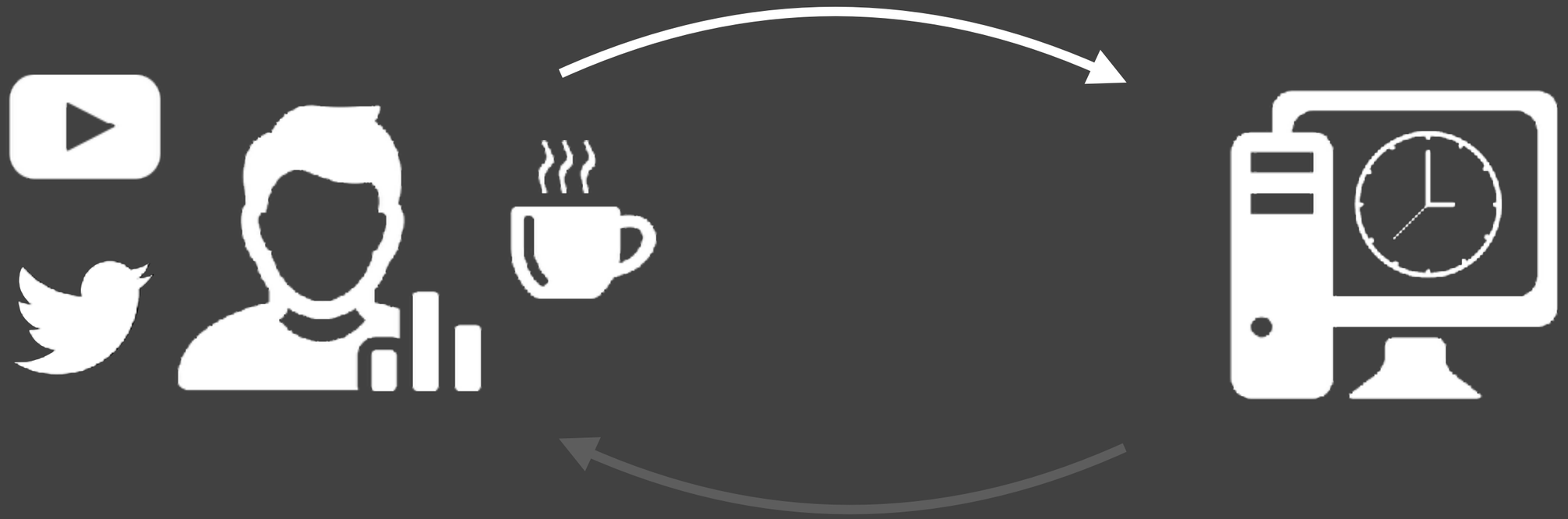
Dominik Moritz @domoritz

Paul G. Allen School of CSE
University of Washington

Danyel Fisher @FisherDanyel

HCI
Microsoft Research

Big Data Visual Analysis

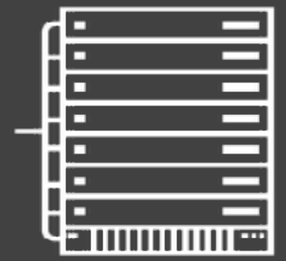


Coffee by jeff from the Noun Project

State of the Art in Big Data Exploration

Distributed Systems

Expensive and high latency.



Indexes (Data Cubes)

Requires pre computation and limited queries.

Sampling

Use a representative subset of the data.

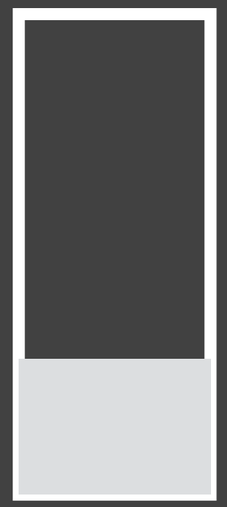
Rubik's Cube by Aleks from the Noun Project

Cluster servers by Branis Panos from the Noun Project

Sampling and Approximate Query Processing (AQP)

Use a representative subset of the data and estimate the true values of aggregate results.

Decide on **acceptable uncertainty** or **timeout**



Sum of 25% = 42

Sum of 100 % = 168 ±10

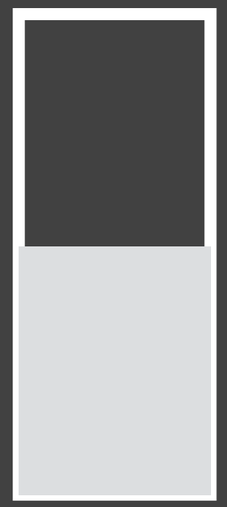
Estimate

Uncertainty

Progressive Visualization with Online Aggregation

Growing sample → continuously improving results

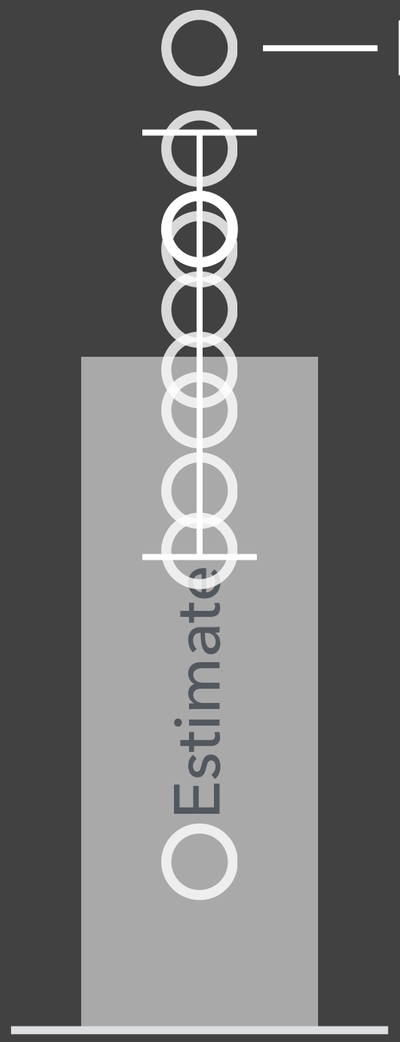
Analysts watch updates until bounds errors are low enough



Sum of 25% = 42 ±10

Sum of 100 % = 168 ±5

Challenges with AQP



Approximate results

→ Convey uncertainty

Probabilistic guarantees

Unbounded errors

Arbitrary aggregation or joins

Optimistic Visualization

A UX approach to challenges with AQP for visual data analysis traditionally treated as database problems.

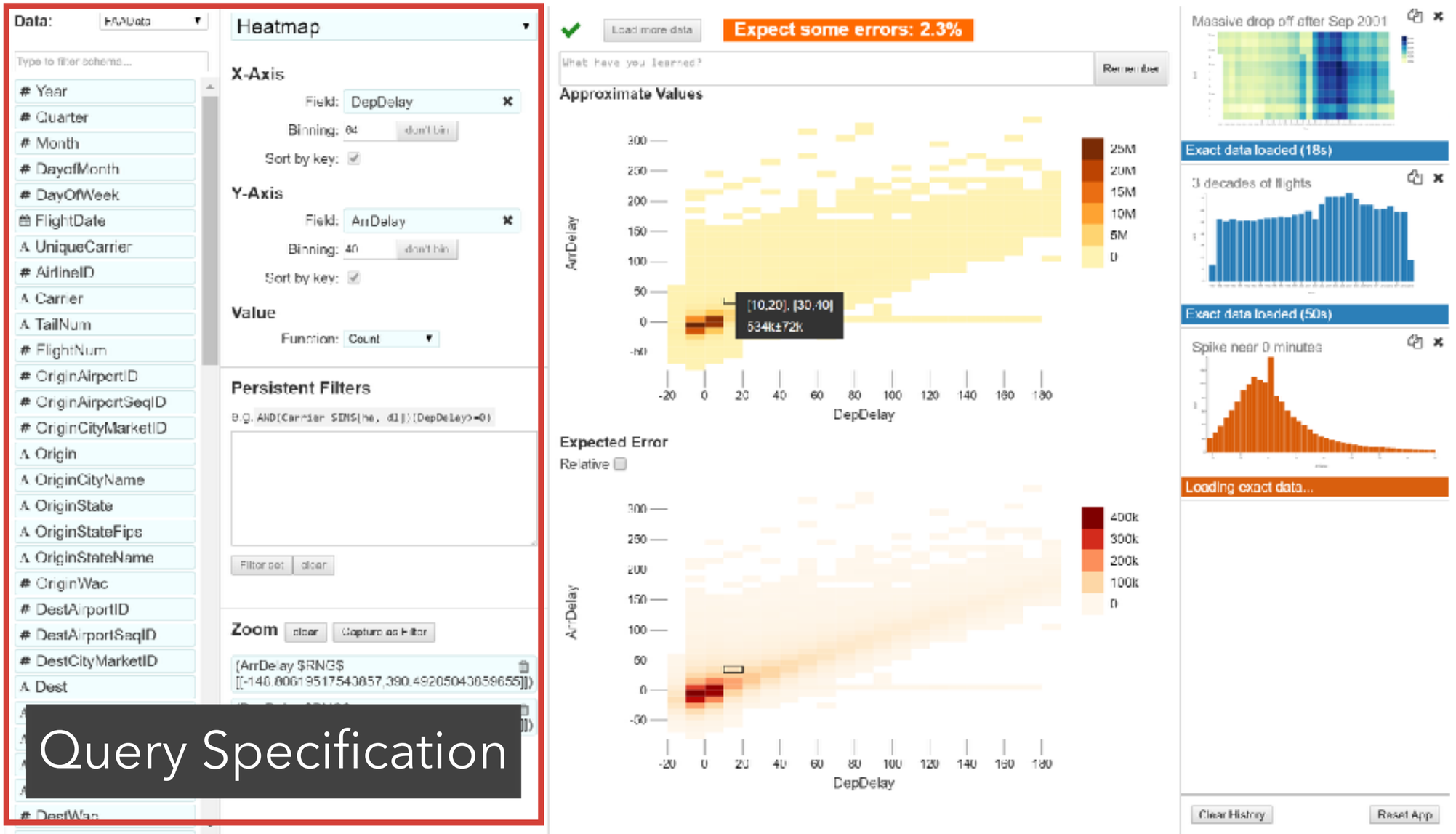
Optimistic Visualization

Assume that approximation is mostly right but offer a way to **detect** and **recover from** mistakes.

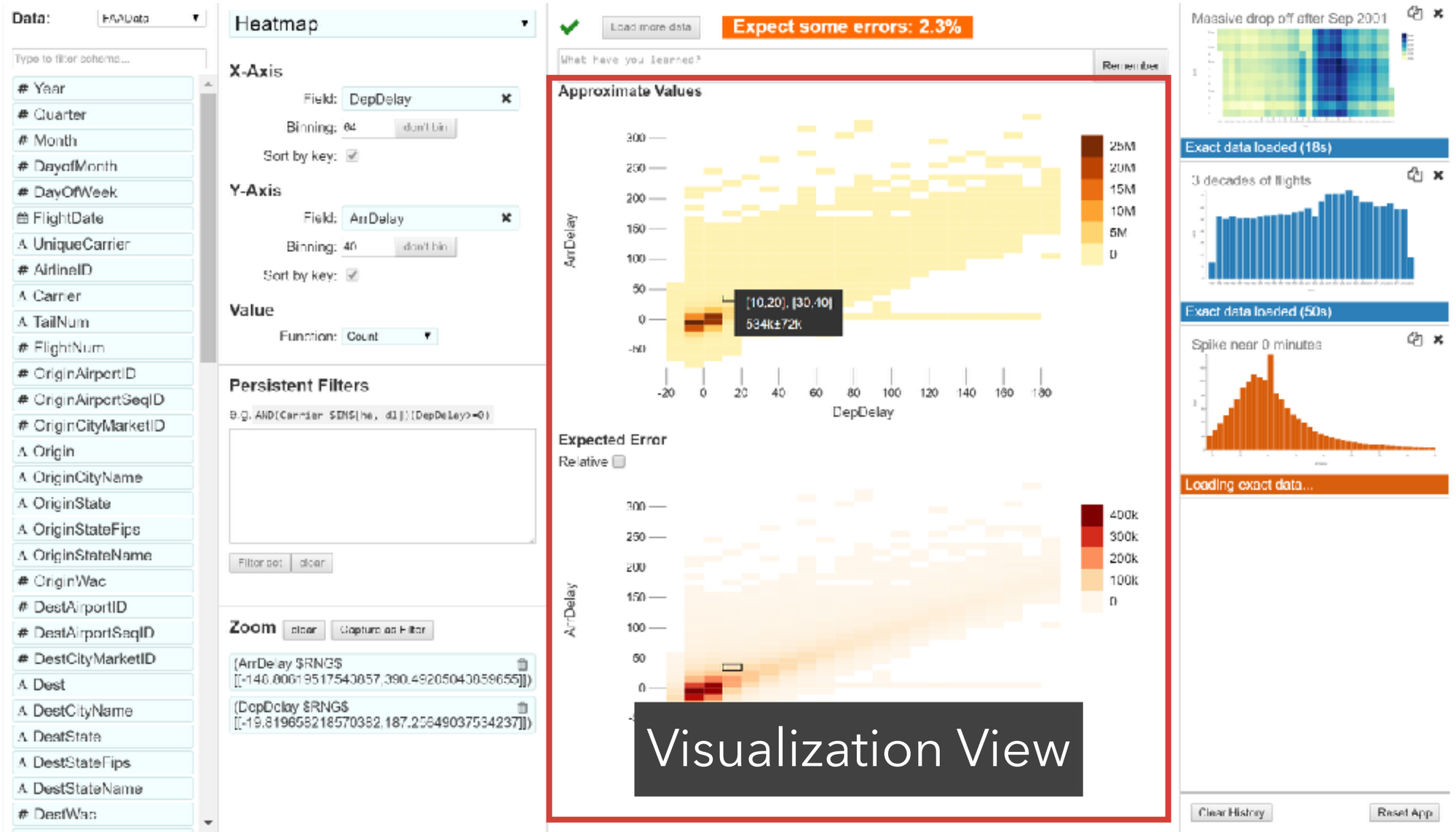
Analysts use initial estimates, run precise query in background, and confirm results later.

Gives users confidence in using AQP.

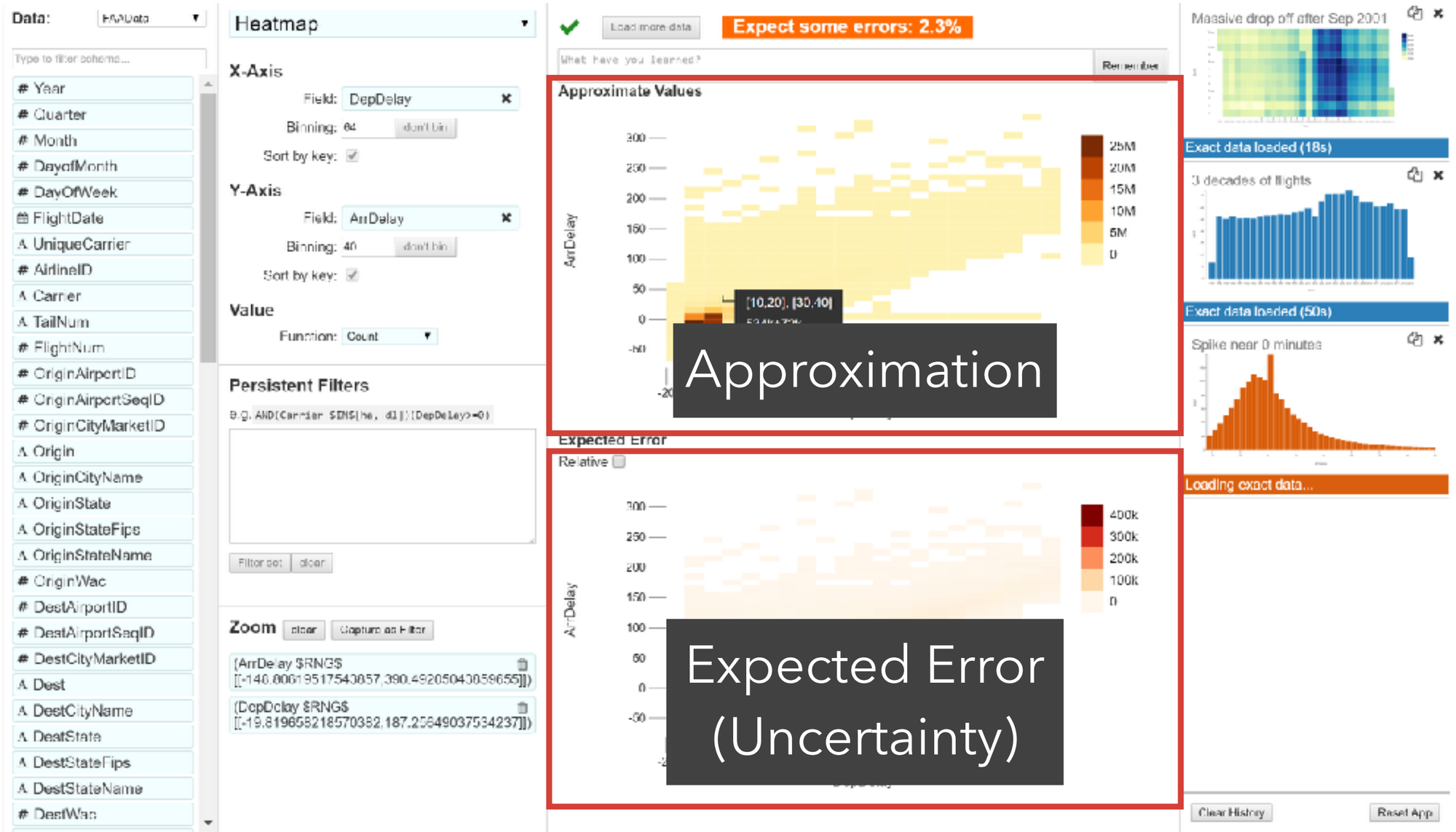
Pangloss implements Optimistic Visualization



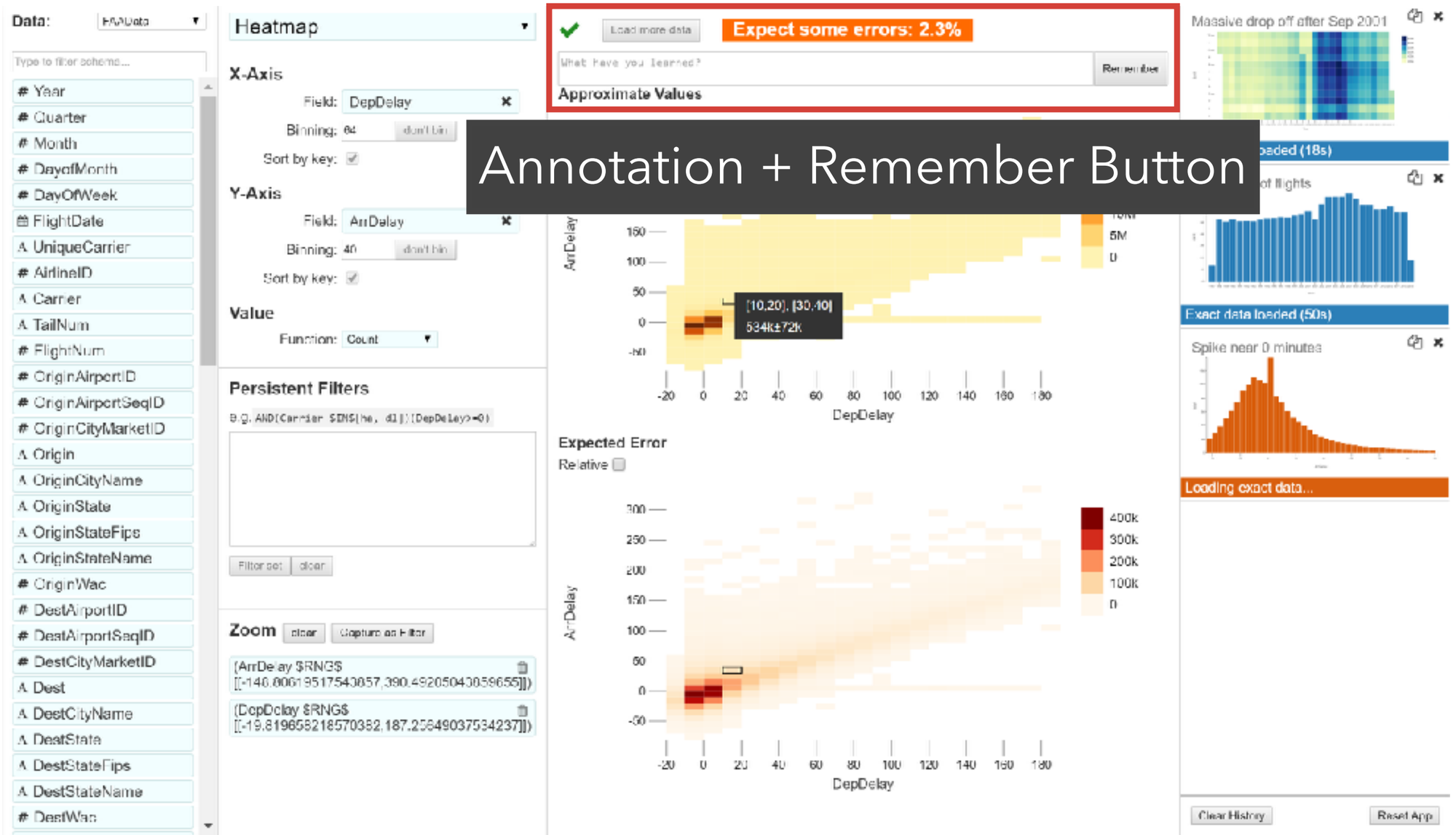
Pangloss implements Optimistic Visualization



Pangloss implements Optimistic Visualization

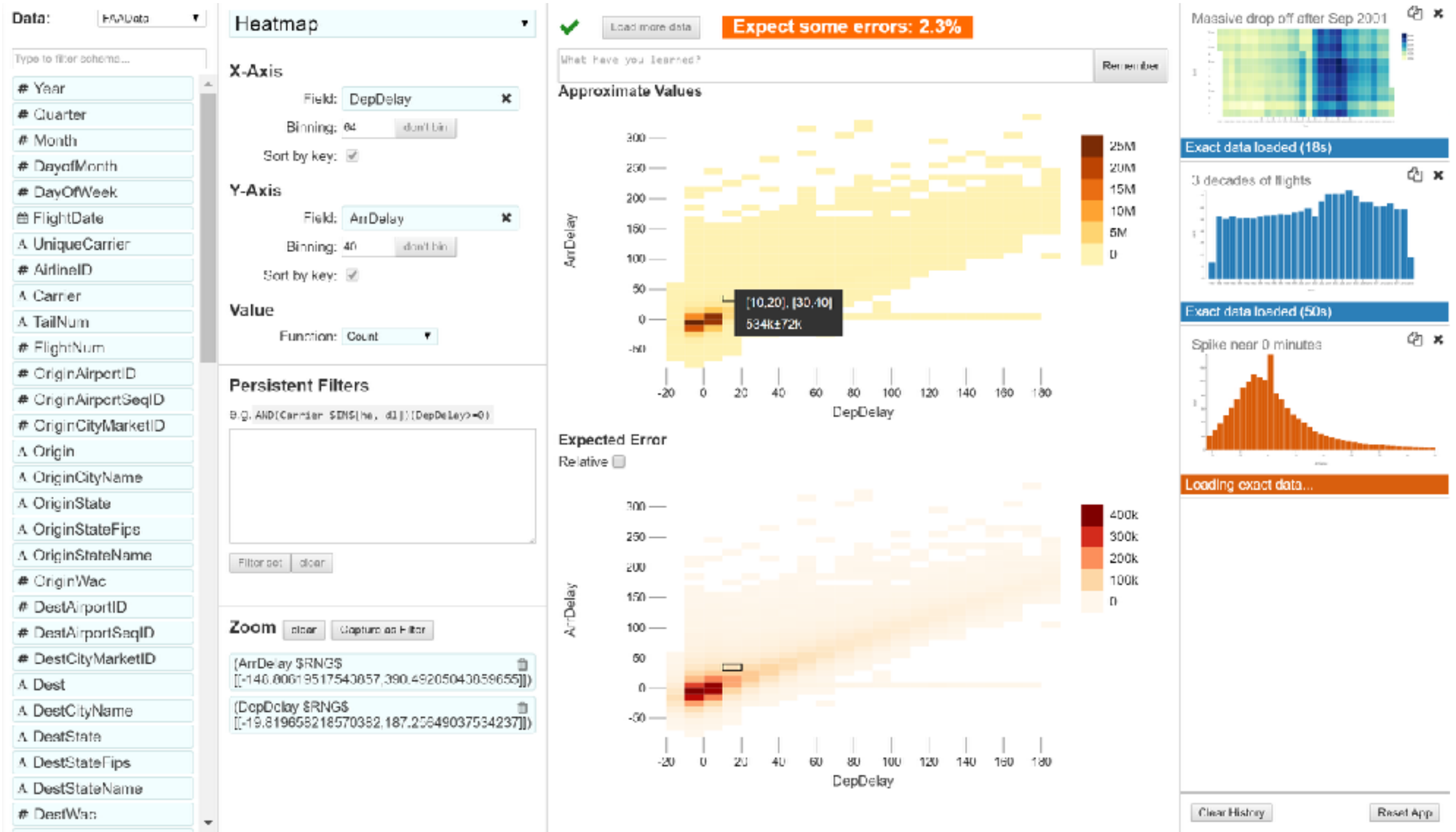


Pangloss implements Optimistic Visualization



Pangloss implements Optimistic Visualization

Pangloss implements Optimistic Visualization



Data: FAADataBarchart

Type to filter schema...

- # Year
- # Quarter
- # Month
- # DayofMonth
- # DayOfWeek
- FlightDate
- A UniqueCarrier
- # AirlineID
- A Carrier
- A TailNum
- # FlightNum
- # OriginAirportID
- # OriginAirportSeqID
- # OriginCityMarketID
- A Origin
- A OriginCityName
- A OriginState
- A OriginStateFips
- A OriginStateName
- # OriginWac
- # DestAirportID
- # DestAirportSeqID
- # DestCityMarketID
- A Dest
- A DestCityName
- A DestState
- A DestStateFips
- A DestStateName
- # DestWac
- A CRSDepTime
- A DepTime
- # DepDelay

X-Axis

Field: Carrier

Binning: 0 bin

Secondary Field:

Sort by key:

Value

Function: Count

Persistent Filters

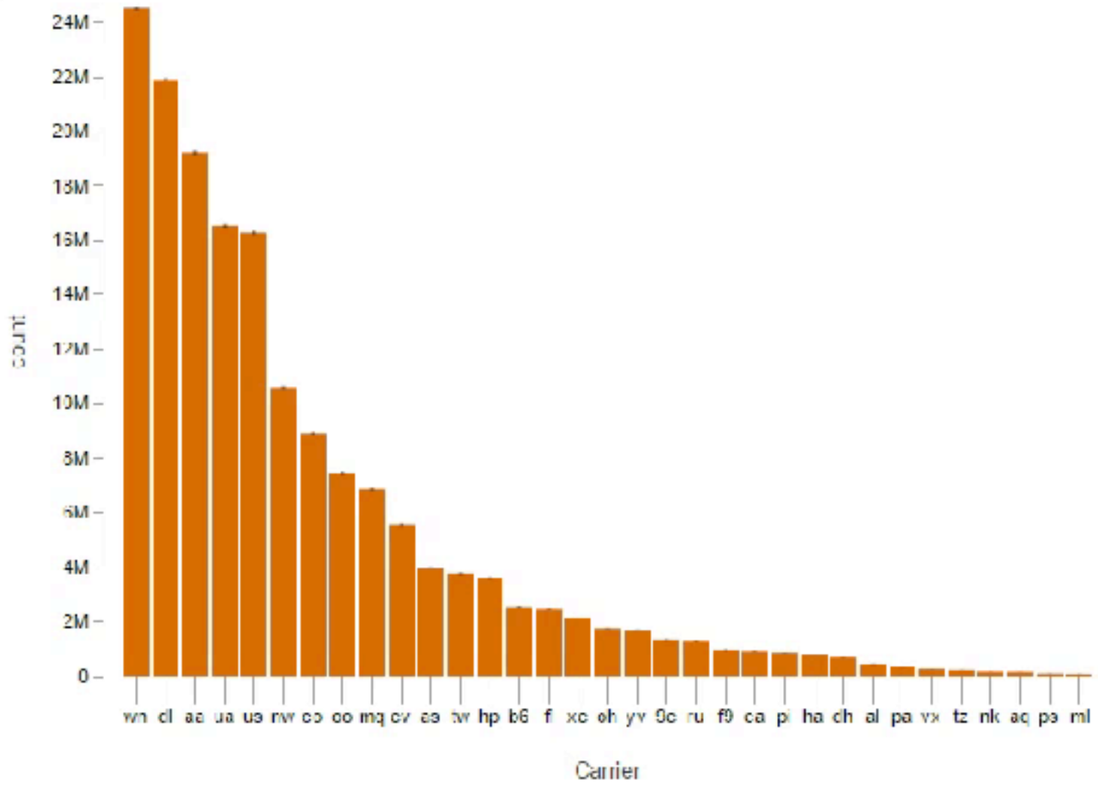
e.g. `AND(Carrier $IN$ [ha, dl])(DepDelay>=0)`

Filter set clear

Zoom clear Capture as Filter

Load more data **Expect almost no errors: 0.2%**

What have you learned? Remember



~100ms query time

170 Million flights (30 years).

Clear History Reset App

Data: FAAData

Type to filter schema...

- # Year
- # Quarter
- # Month
- # DayofMonth
- # DayOfWeek
- FlightDate
- A UniqueCarrier
- # AirlineID
- A Carrier
- A TailNum
- # FlightNum
- # OriginAirportID
- # OriginAirportSeqID
- # OriginCityMarketID
- A Origin
- A OriginCityName
- A OriginState
- A OriginStateFips
- A OriginStateName
- # OriginWac
- # DestAirportID
- # DestAirportSeqID
- # DestCityMarketID
- A Dest
- A DestCityName
- A DestState
- A DestStateFips
- A DestStateName
- # DestWac
- A CRSDepTime
- A DepTime
- # DepDelay

Barchart

X-Axis

Field: Year

Binning: 0 bin

Secondary Field:

Sort by key:

Value

Function: Count

Persistent Filters

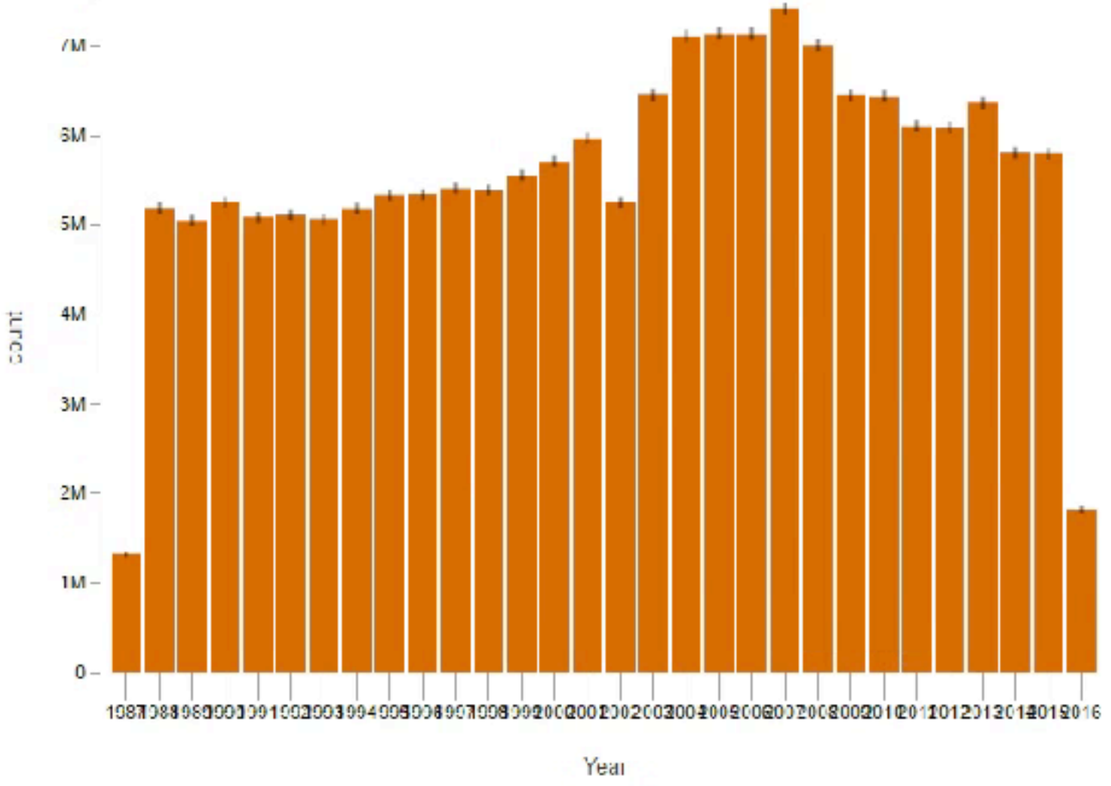
e.g. `AND(Carrier $IN$ [ha, dl])(DepDelay>=0)`

Filter set clear

Zoom clear Capture as Filter

Load more data **Expect almost no errors: 0.3%**

What have you learned? Remember



Text annotations help analysts clarify observations.

Clear History Reset App

Data: FAAData

Type to filter schema...

- # Year
- # Quarter
- # Month
- # DayofMonth
- # DayOfWeek
- FlightDate
- A UniqueCarrier
- # AirlineID
- A Carrier
- A TailNum
- # FlightNum
- # OriginAirportID
- # OriginAirportSeqID
- # OriginCityMarketID
- A Origin
- A OriginCityName
- A OriginState
- A OriginStateFips
- A OriginStateName
- # OriginWac
- # DestAirportID
- # DestAirportSeqID
- # DestCityMarketID
- A Dest
- A DestCityName
- A DestState
- A DestStateFips
- A DestStateName
- # DestWac
- A CRSDepTime
- A DepTime
- # DepDelay

Barchart

X-Axis

Field: Year

Binning: 0 bin

Secondary Field:

Sort by key:

Value

Function: Count

Persistent Filters

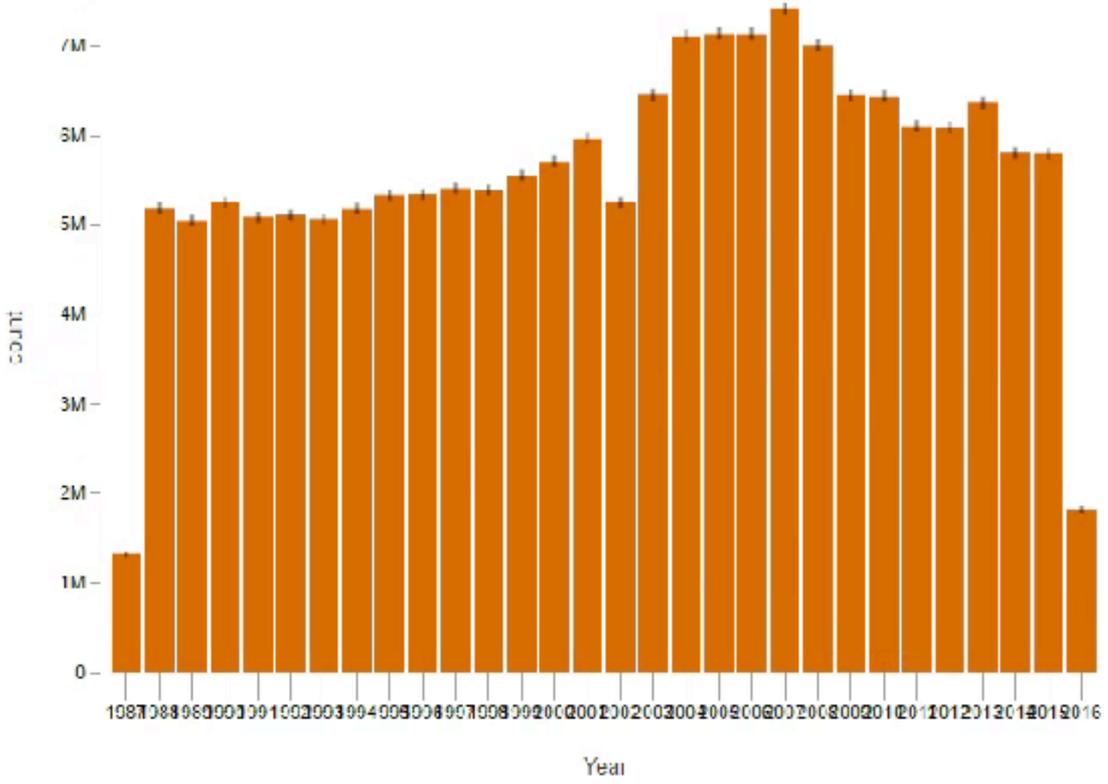
e.g. `AND(Carrier $IN$ ['ha', 'dl'])(DepDelay>=0)`

Filter set clear

Zoom clear Capture as Filter

Load more data **Expect almost no errors: 0.3%**

3 decades of flights Remember



"Remember" button moves query into the background

Clear History Reset App

Data: FAAData

Type to filter schema...

- # Year
- # Quarter
- # Month
- # DayofMonth
- # DayOfWeek
- FlightDate
- A UniqueCarrier
- # AirlineID
- A Carrier
- A TailNum
- # FlightNum
- # OriginAirportID
- # OriginAirportSeqID
- # OriginCityMarketID
- A Origin
- A OriginCityName
- A OriginState
- A OriginStateFips
- A OriginStateName
- # OriginWac
- # DestAirportID
- # DestAirportSeqID
- # DestCityMarketID
- A Dest
- A DestCityName
- A DestState
- A DestStateFips
- A DestStateName
- # DestWac
- A CRSDepTime
- A DepTime
- # DepDelay

Barchart

X-Axis

Field: Year

Binning: 0 bin

Secondary Field:

Sort by key:

Value

Function: Count

Persistent Filters

e.g. `AND(Carrier $IN$['ha', 'dl'])(DepDelay>=0)`

Filter set clear

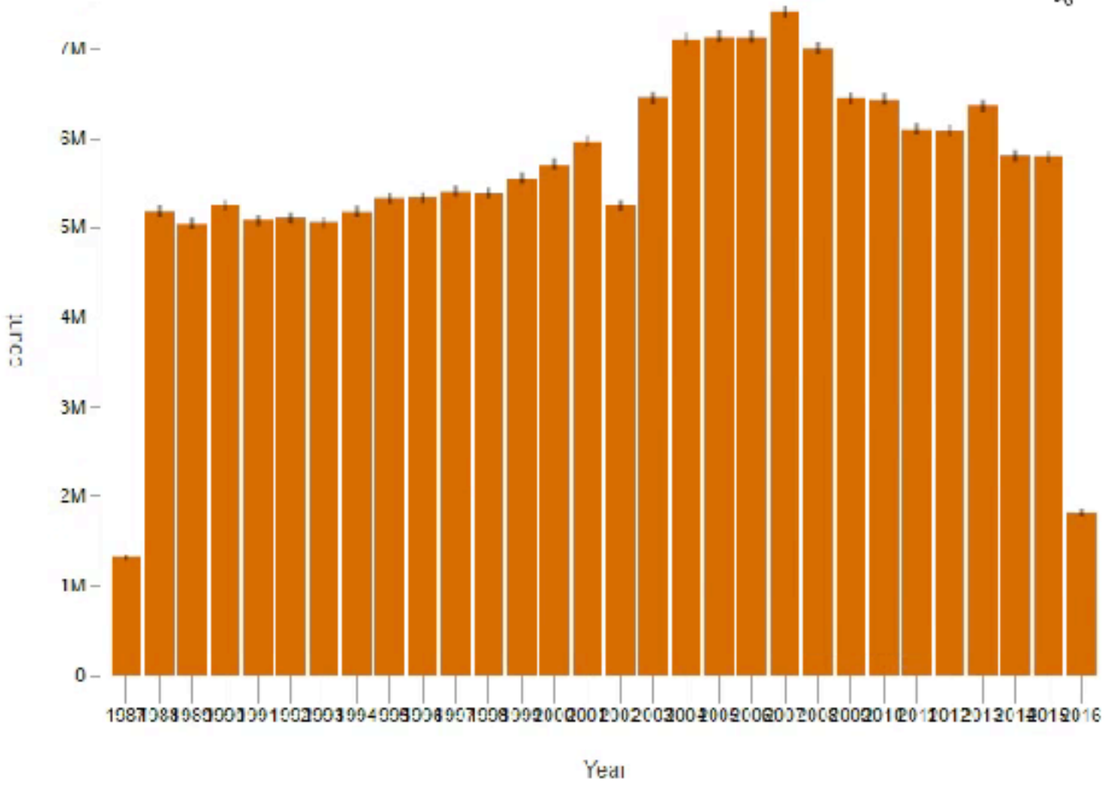
Zoom clear Capture as Filter

Load more data

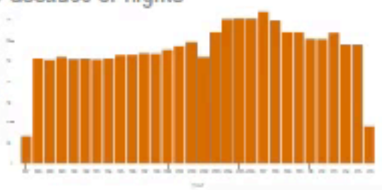
Expect almost no errors: 0.3%

What have you learned?

Remember



3 decades of flights



Loading exact data...

Clear History Reset App

Continue exploration without waiting

Data: FAAData

orig

OriginAirportID
OriginAirportSeqID
OriginCityMarketID
A Origin
A OriginCityName
A OriginState
A OriginStateFips
A OriginStateName
OriginWac

Barchart

X-Axis

Field: OriginCityName

Binning: 0 bin

Secondary Field:

Sort by key:

Value

Function: Count

Persistent Filters

e.g. `AND(Carrier $IN$ [ha, dl])(DepDelay>=0)`

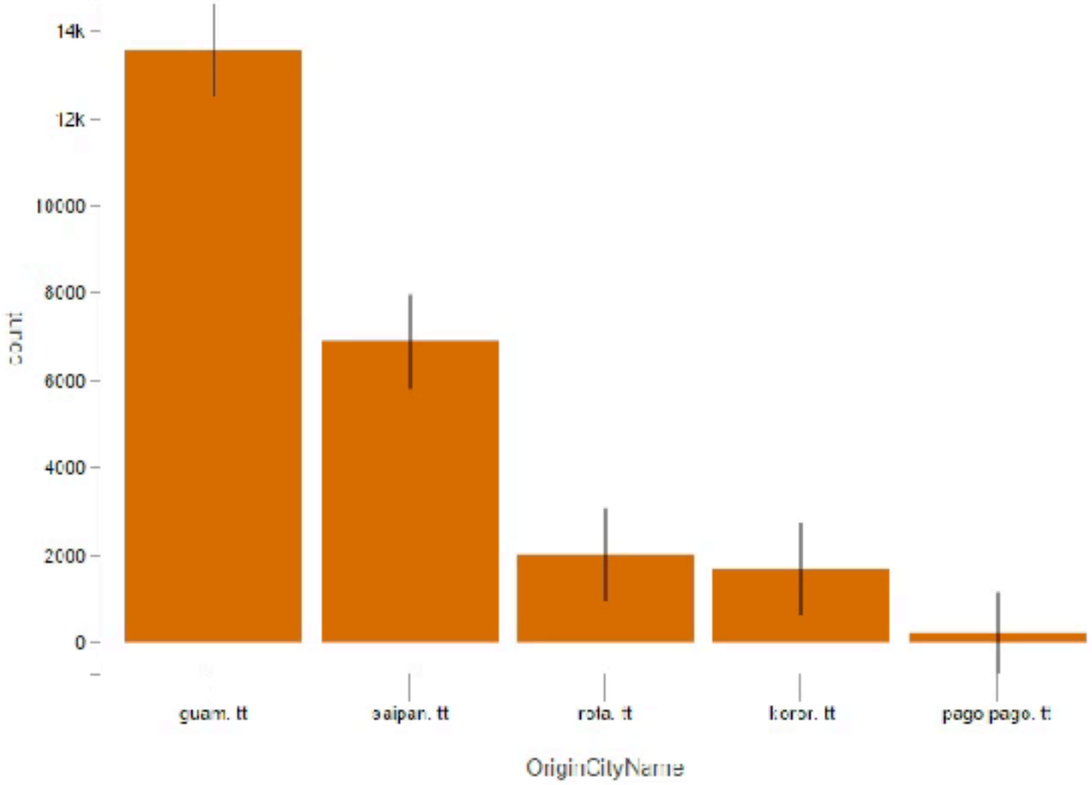
(OriginState=tt)

Filter set clear

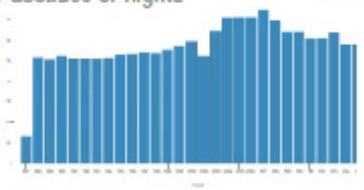
Zoom clear Capture as Filter

Load more data **Expect some errors: 7.5%**

What have you learned? Remember



3 decades of flights

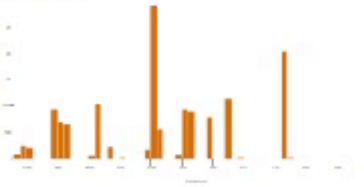


Exact data loaded (61.156s)

ha has short, mid and long range flights

Exact data loaded (61.153s)

mid range flights have distinct distances

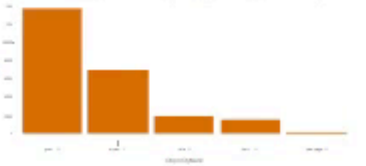


Loading exact data...

mostly ca to ha

Loading exact data...

tt is in the pacific (saipan, guam



Clear History Reset App

Orange → Approximate Blue → Precise

Data: FAAData

orig

OriginAirportID
OriginAirportSeqID
OriginCityMarketID
A Origin
A OriginCityName
A OriginState
A OriginStateFips
A OriginStateName
OriginWac

Heatmap

X-Axis

Field: OriginState

Binning: 0

Sort by key:

Y-Axis

Field: DestState

Binning: 0

Sort by key:

Value

Function: Count

Persistent Filters

e.g. AND(Carrier IN[ha, dl])(DepDelay>=0)

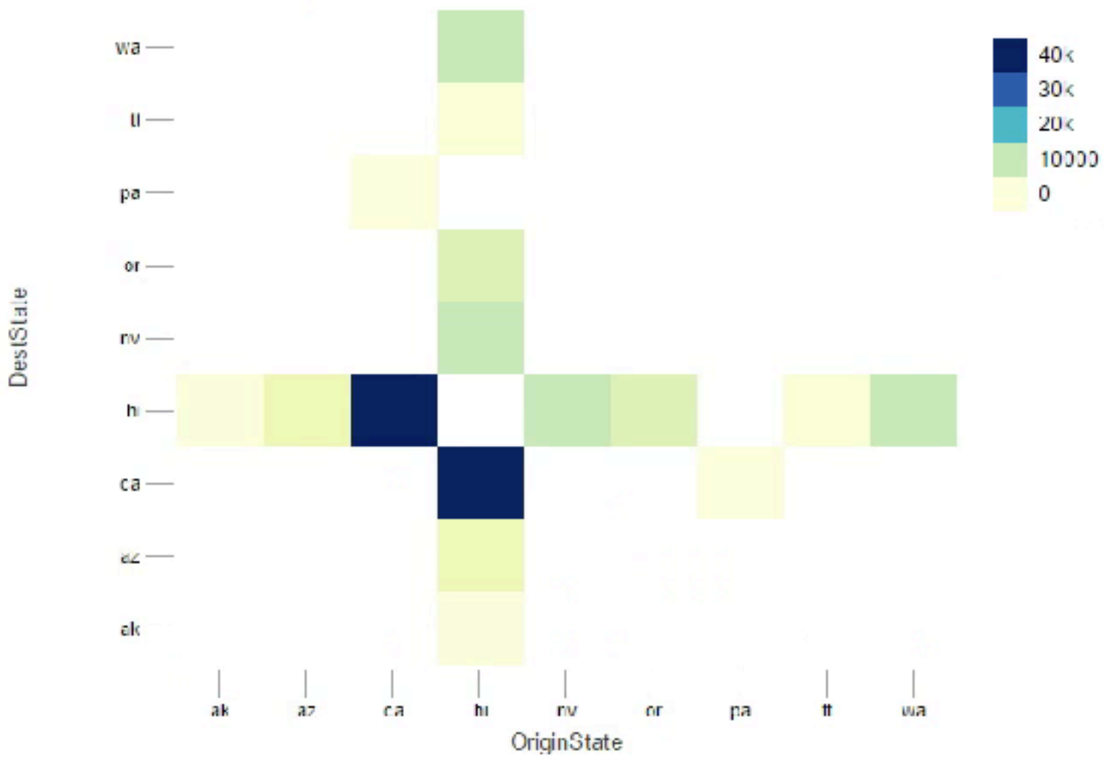
AND(Carrier-ha)(Distance RNG [[2168.9792406152524,3201.5703990534505]])

Zoom

mostly ca to ha

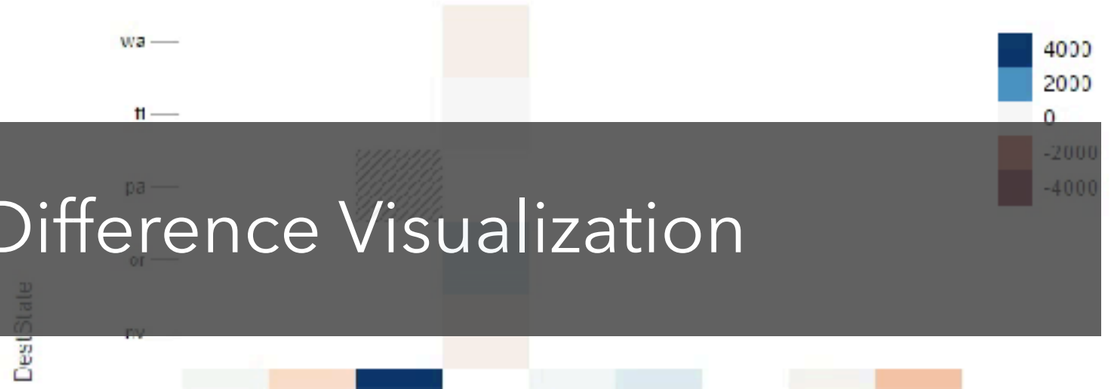
The visualization is read only because you're looking at the history. Return to the working vis or make a copy of the current chart.

Exact Data



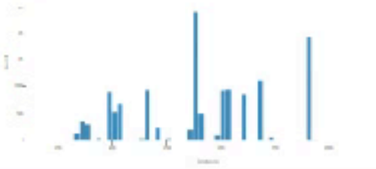
Difference to Approximate Data

Relative



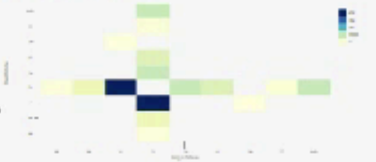
Exact data loaded (61.153s)

mid range flights have distinct distances



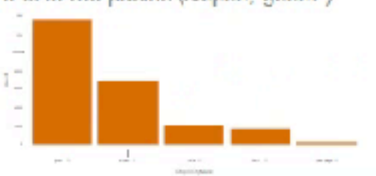
Exact data loaded (61.149s)

mostly ca to ha



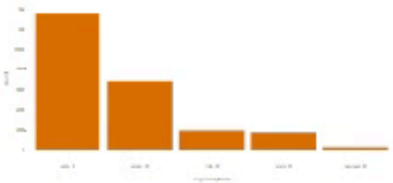
Exact data loaded (60.013s)

tt is in the pacific (saipan, guam)



Loading exact data ...

You are looking at the history and cannot make any changes



Return to editing

Clear History

Reset App

Difference Visualization

Evaluation

Lab Study

5 users

Flight delay data
(170 Million records)

1 hour each

Case Study

3 teams

Product insights,
Social media,
Bing

~1+ hour exploration

Findings from the study

AQP works: “seeing something right away at first glimpse is really great”

Optimism works: “I was thinking what to do next– and I saw that it had loaded, so I went back and checked it . . . [the passive update is] very nice for not interrupting your workflow.”

Need for guarantees: “[with a competitor] I was willing to wait 70-80 seconds. It wasn’t ideally interactive, but it meant I was looking at **all** the data.”

Findings from the study (cont)

“When I’m using your system, there is a path that I need to follow.”

“Now that I’ve been sitting here for an hour, after I go back, it makes a lot of sense [to have these annotations], but as I was doing it, I was thinking, ‘I want to move on, I want to move on.”

Adopt Optimistic Visualization

Uncertainty Visualization is not strictly required

Precise query can benefit from highly optimized Databases

Optimistic Visualization can help with adoption of AQP

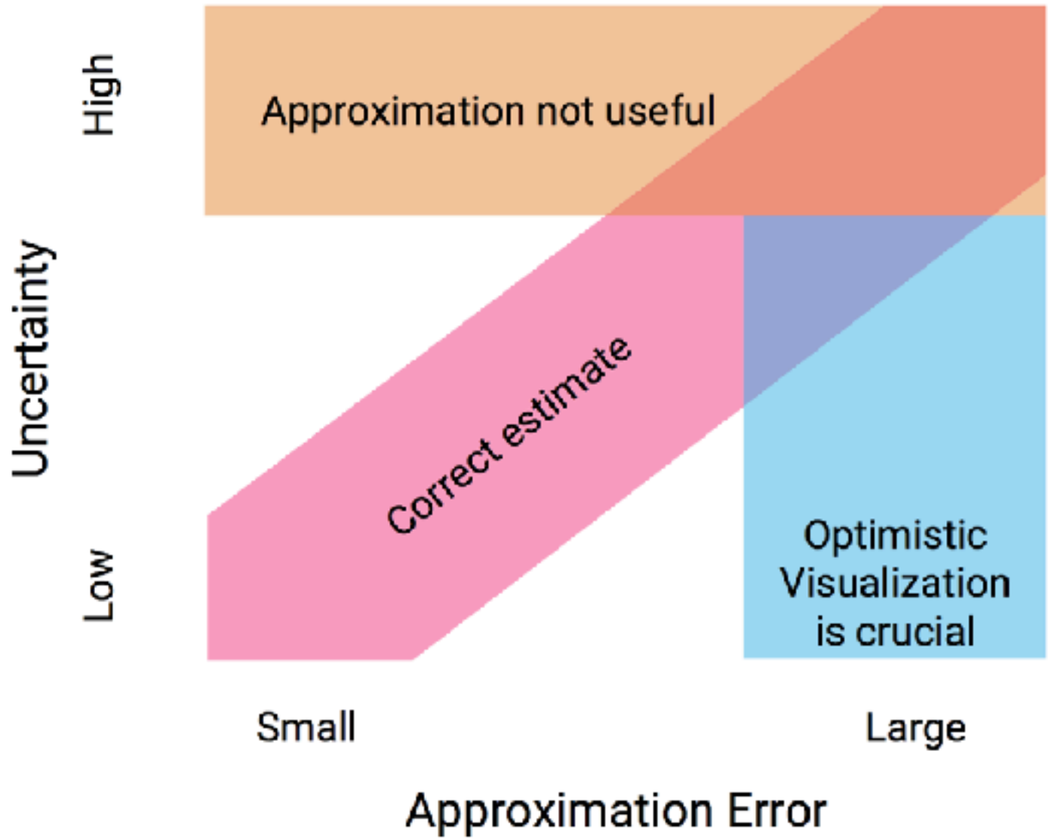
Understanding Approximation Error

Approximation Error

The true error of the approximation. Only known after we run the full query!

Uncertainty

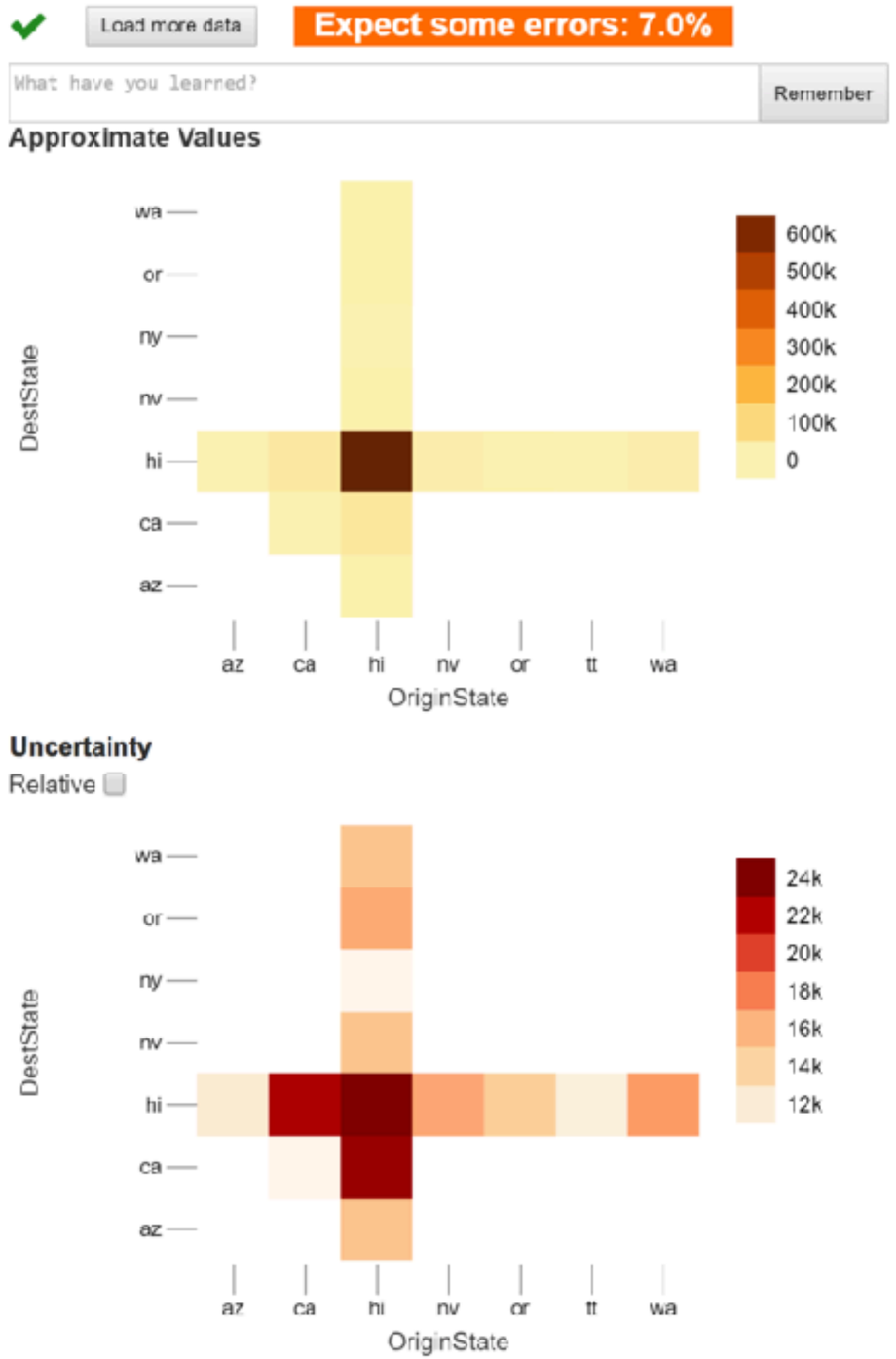
Expected approximation error.



2D Uncertainty

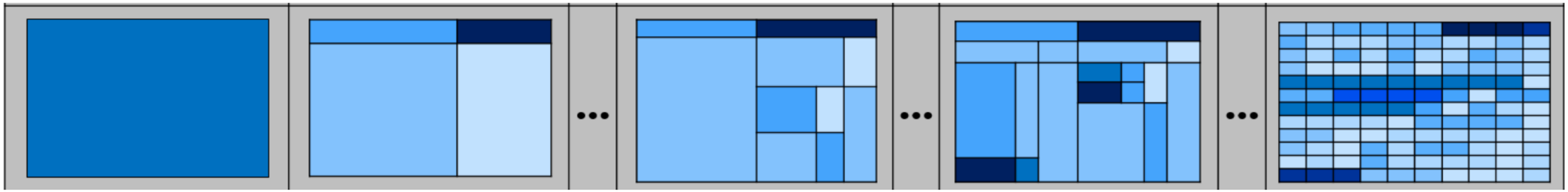
No best practices

Currently: juxtaposed heatmaps



2D Uncertainty

Percentage different? vs **Value** different?

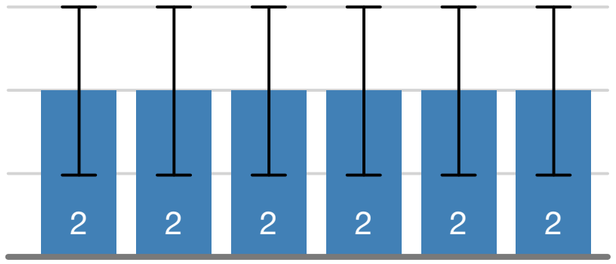


[IncVisage]

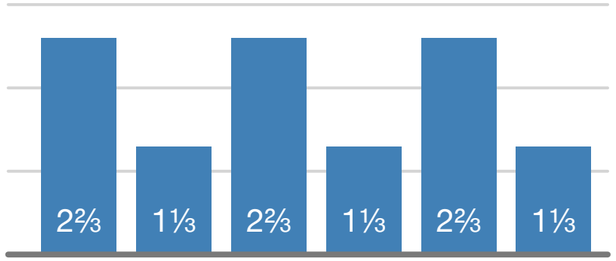
Distribution Uncertainty

Approximation

Distribution Uncertainty: 4

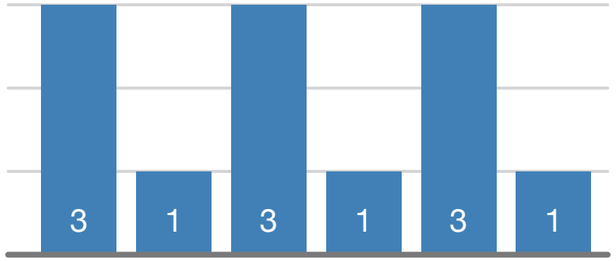


Within Distribution Uncertainty

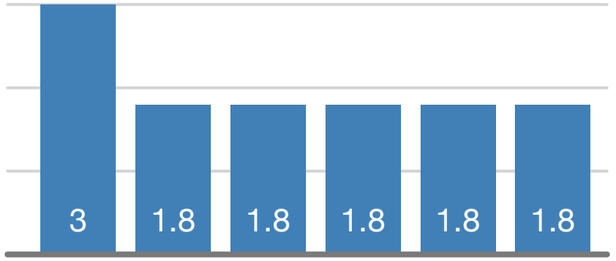


Error: 4
Sum: 12

Outside Distribution Uncertainty

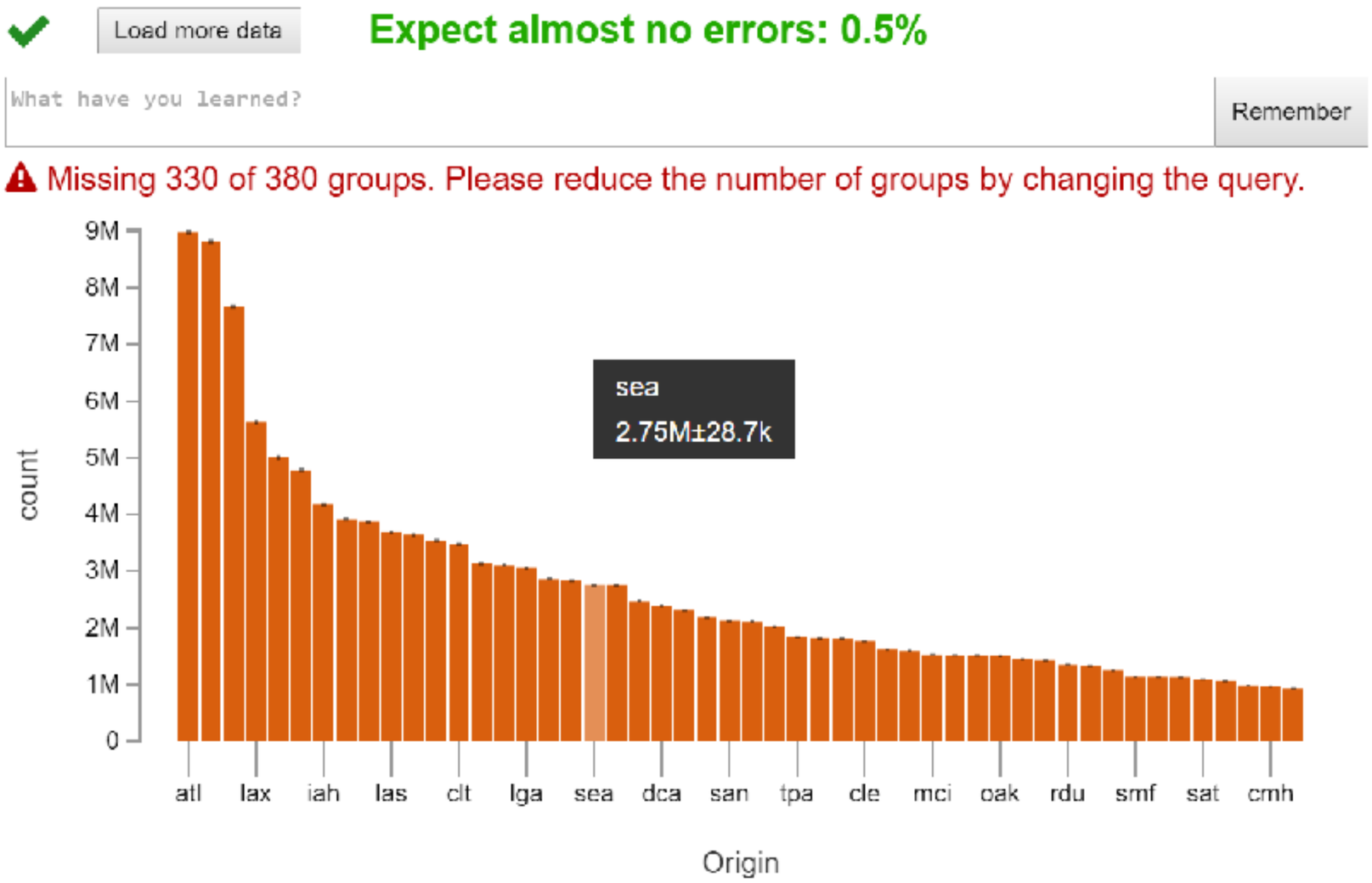


Error: 6
Sum: 12

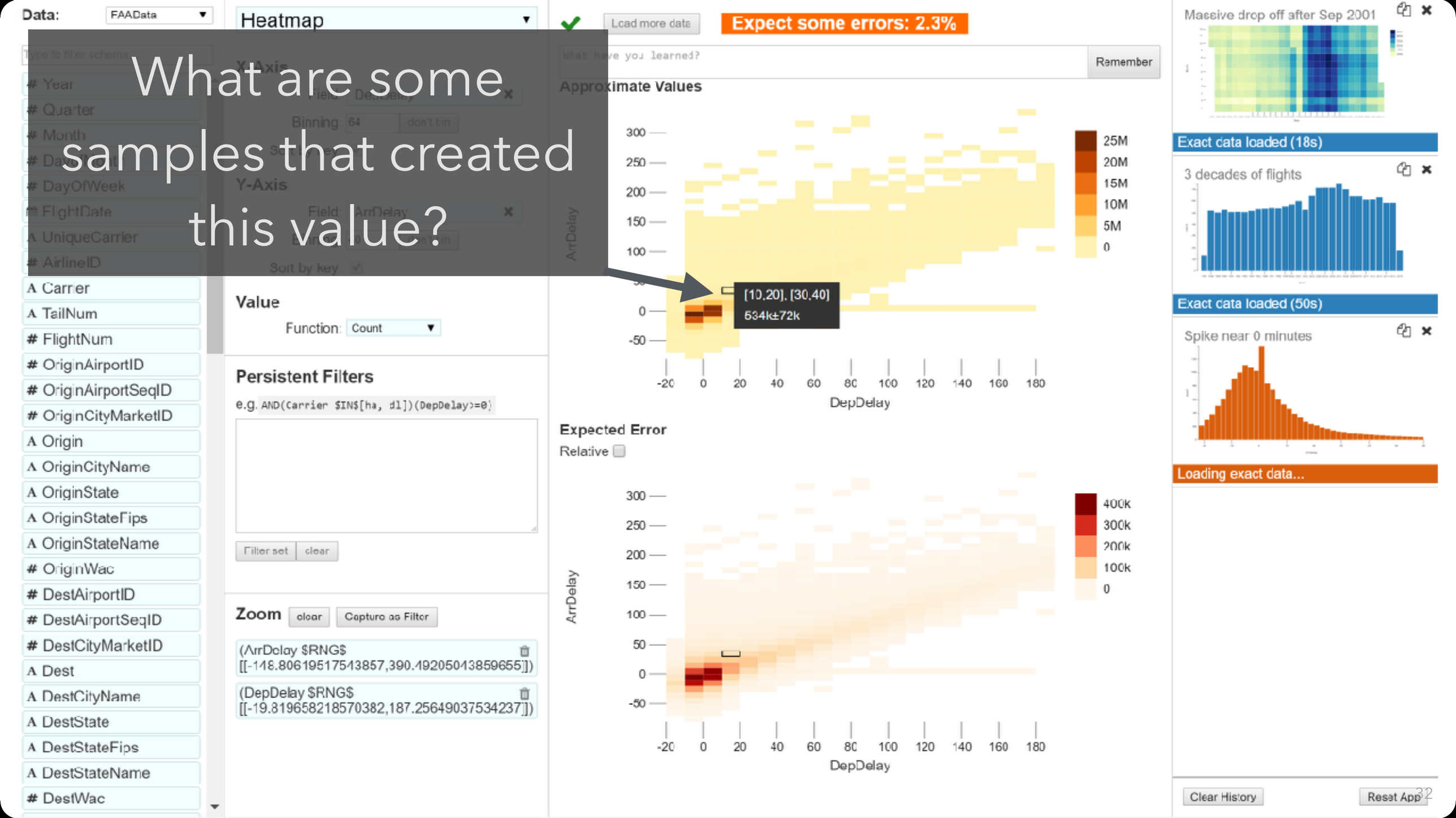


Error: 4
Sum: 12

Distribution Uncertainty



What are some samples that created this value?



Optimizing the Language for Data Exploration

Tweaking SQL for high-level operations & sessions

```
SELECT HISTOGRAM(DISTANCE) WITH ALGORITHM="nice"
SELECT HISTOGRAM(DISTANCE) WITH BUCKETS=(0,10,20,30)
```

Knowing what queries are related in an exploration session enables new optimizations, e.g. ForeCache.

Data: FAADataset

origi

- # OriginAirportID
- # OriginAirportSeqID
- # OriginCityMarketID
- A Origin
- A OriginCityName
- A OriginState
- A OriginStateFips
- A OriginStateName
- # OriginWac

Heatmap

X-Axis

Field: OriginState

Binning: 0

Sort by key:

Y-Axis

Field: DestState

Binning: 0

Sort by key:

Value

Function: Count

Persistent Filters

e.g. `AND(Carrier $IN$[ha, dl])(DepDelay>=0)`

```
AND(Carrier=ha)(Distance $RNG$
[[2168.9792406152524,3201.570399053
4585]])
```

Zoom

mostly ca to ha

The visualization is read only because you're looking at the history. Return to the working vis or make a copy of the current chart.

Exact Data

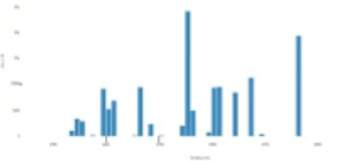
More complex filters
= more samples
= slower performance

Difference to Approximate Data

Relative

Exact data loaded (61.153s)

mid range flights have distinct distances



Exact data loaded (61.149s)

mostly ca to ha

Exact data loaded (60.013s)

tt is in the pacific (saipan, guam)

Loading exact data...

You are looking at the history and cannot make any changes.

Return to editing

Clear History

Reset App

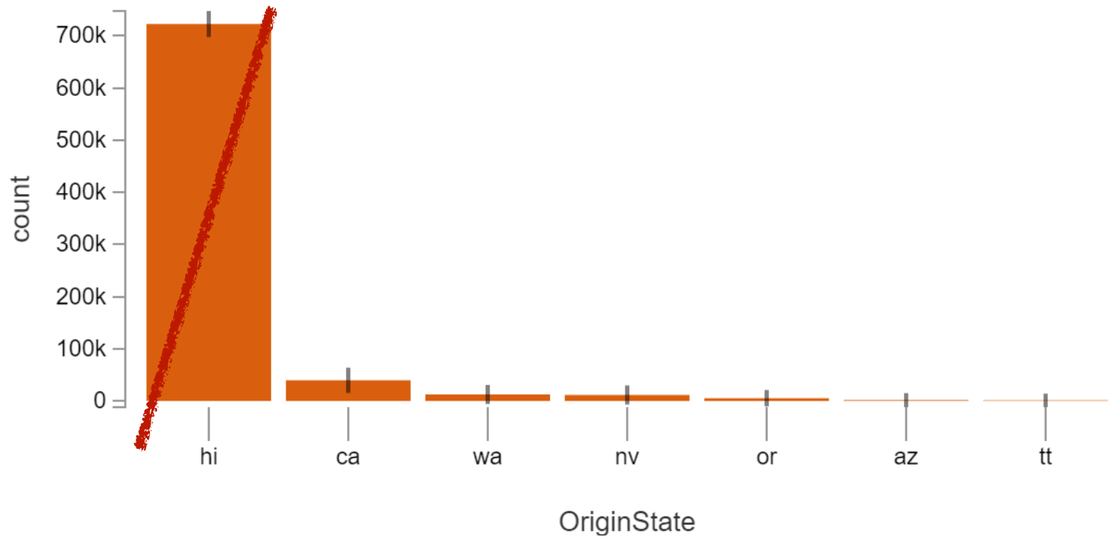
Filtering can show new groups

New Predicate

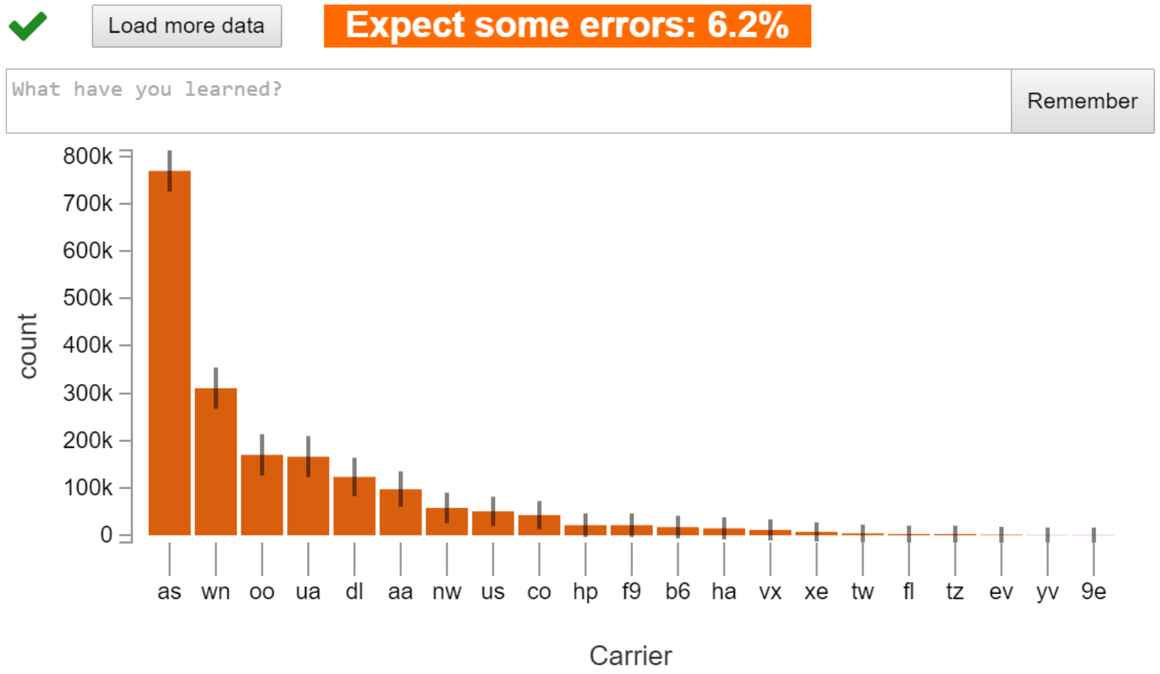
New Query

Different Sample

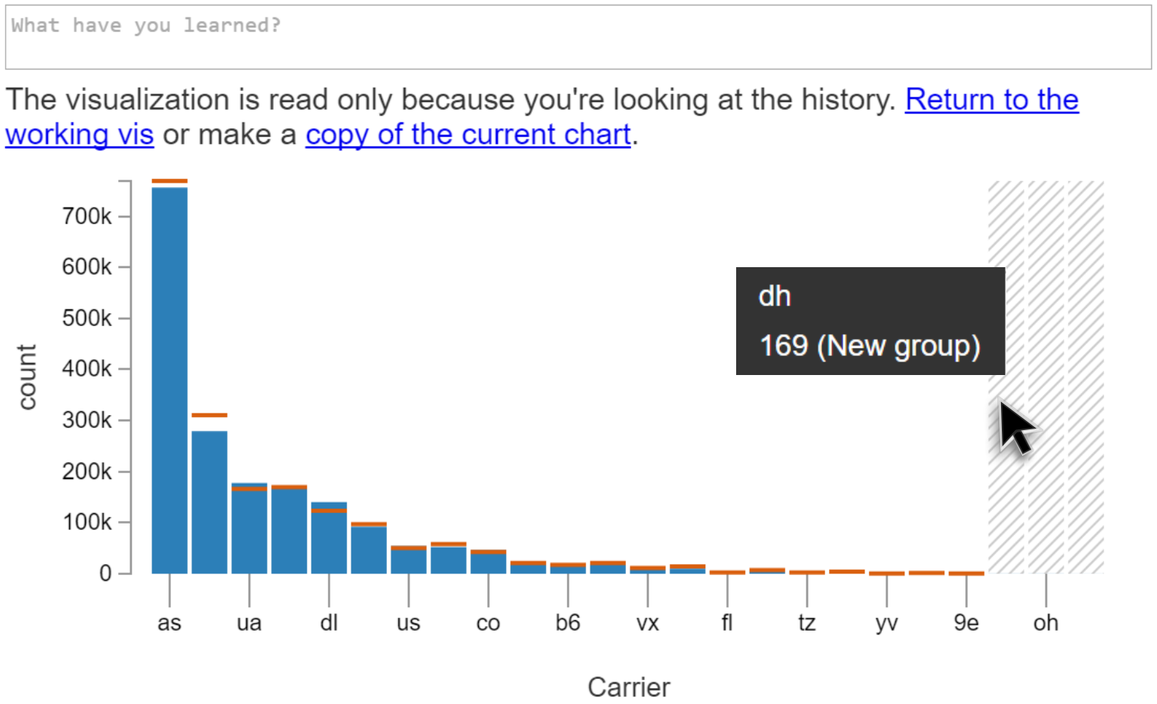
Different Groups



Precise results can show new groups

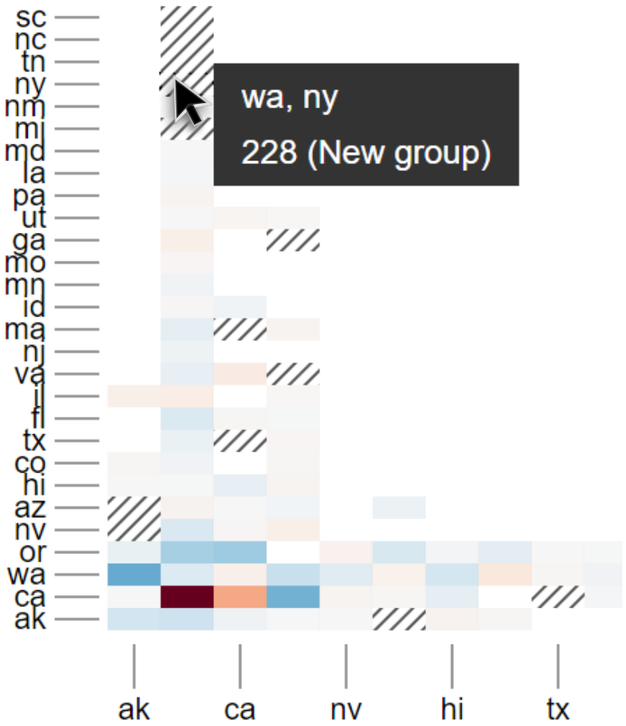


Approximate

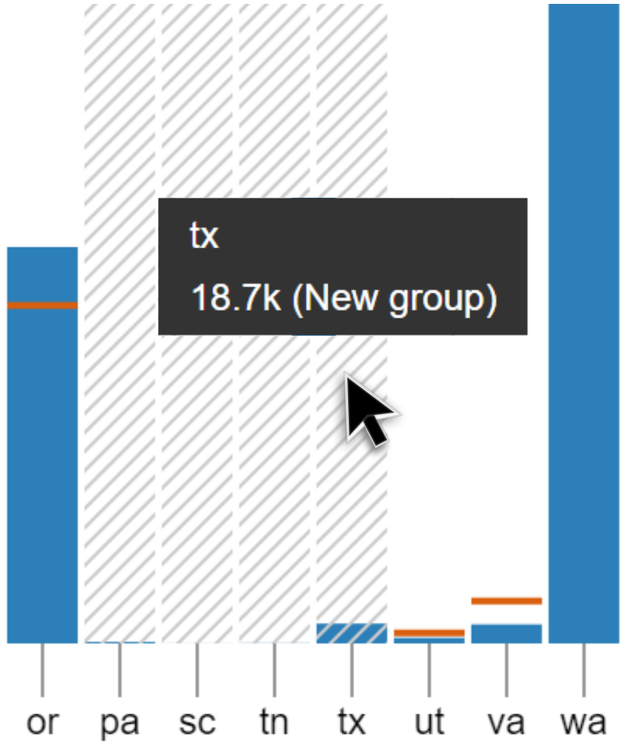


Precise

Vocabulary of visual cues



Heatmap



Barchart

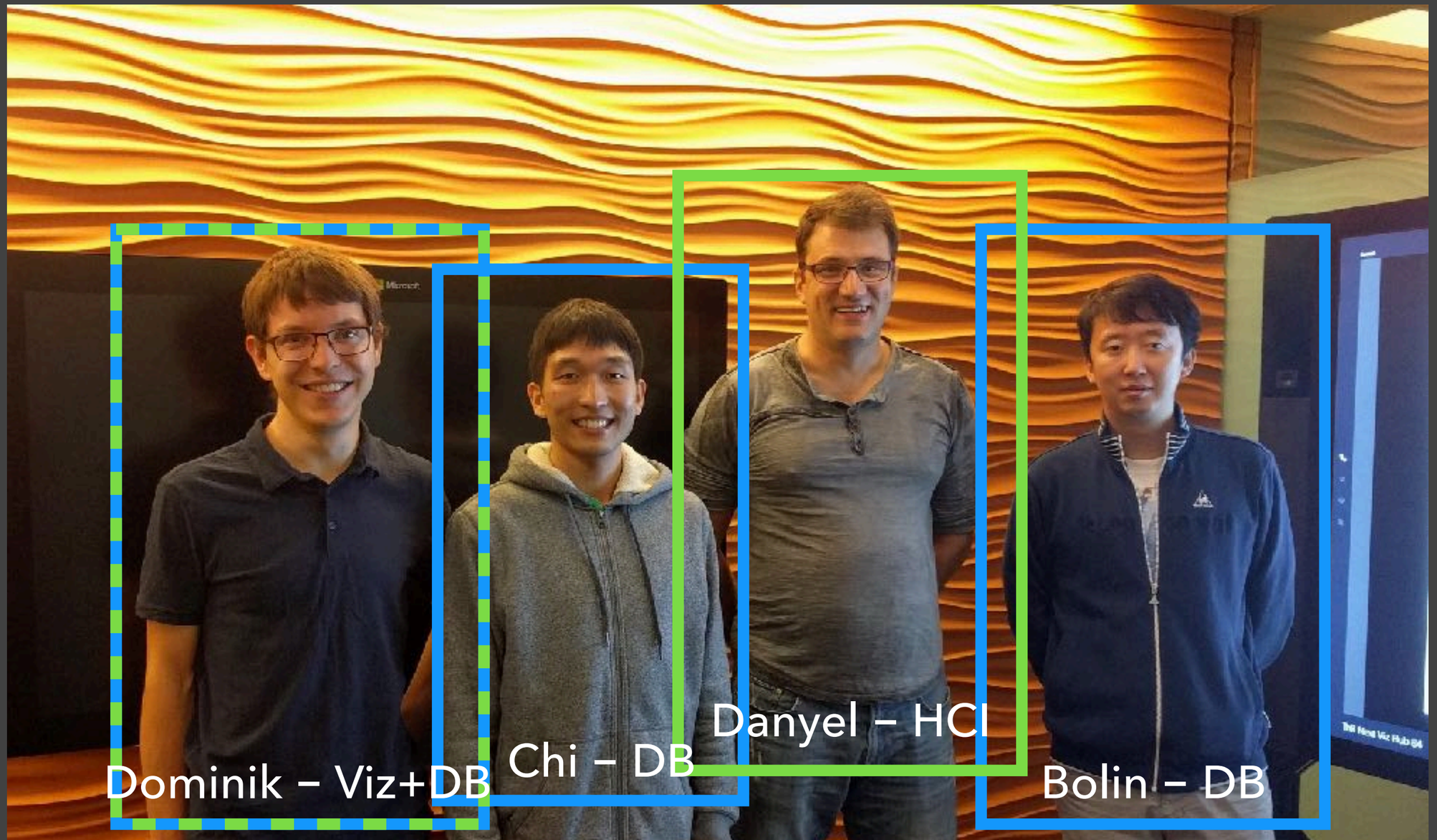
Conclusions

Optimistic Visualization addresses fundamental problems with AQP as **UX problem**

UI tools make invalid assumptions, AQP tools are not designed for visual analytics

Need to continue exploring the UX issues with AQP

AQP needs Multi-Disciplinary Solutions



Challenges with AQP as UX Problem

CHI 2017

Trust, but Verify: Optimistic Visualizations of Approximate Queries for Exploring Big Data

Dominik Moritz
University of Washington
domoritz@cs.uw.edu

Danyel Fisher
Microsoft Research
danyelf@microsoft.com

Bolin Ding, Chi Wang
DMX, Microsoft Research
bolind@microsoft.com,
chiw@microsoft.com

ABSTRACT
Analysts need interactive speed for exploratory analysis, but big data systems are often slow. With sampling, data systems can produce approximate answers fast enough for exploratory visualization, at the cost of accuracy and trust. We propose *optimistic visualization*, which approaches these issues from a user experience perspective. This method lets analysts explore approximate results interactively, and provides a way to detect and recover from errors later. *Pangloss* implements these ideas.

In this paper, rather than addressing the problems with AQP from an algorithmic or systems perspective, we formulate them as user experience problems. What user experience would enable analysts to gain the benefits of approximate queries, while still being able to trust the results?

We propose an approach which we call *optimistic visualization*. Optimistic visualization produces approximate results quickly, and computes precise results in the background. The analyst can make observations on the approximation, and later check

What Users Don’t Expect about Exploratory Data Analysis on Approximate Query Processing Systems

Optimistic Visualization addresses fundamental problems with AQP as **UX problem**

UI tools make invalid assumptions, AQP tools are not designed for visual analytics

Need to continue exploring the UX issues with AQP

Dominik Moritz @domoritz
Danyel Fisher @FisherDanyel
Bolin Ding @AtlasDing
Chi Wang