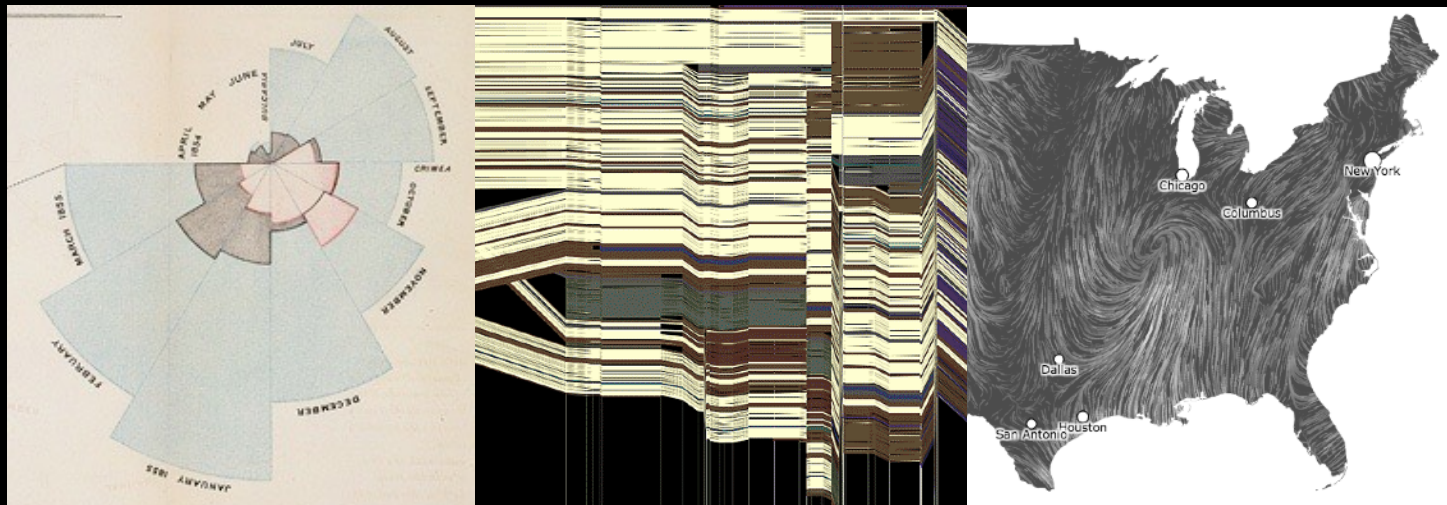


CSE 442 - Data Visualization

Scalability



Jeffrey Heer University of Washington

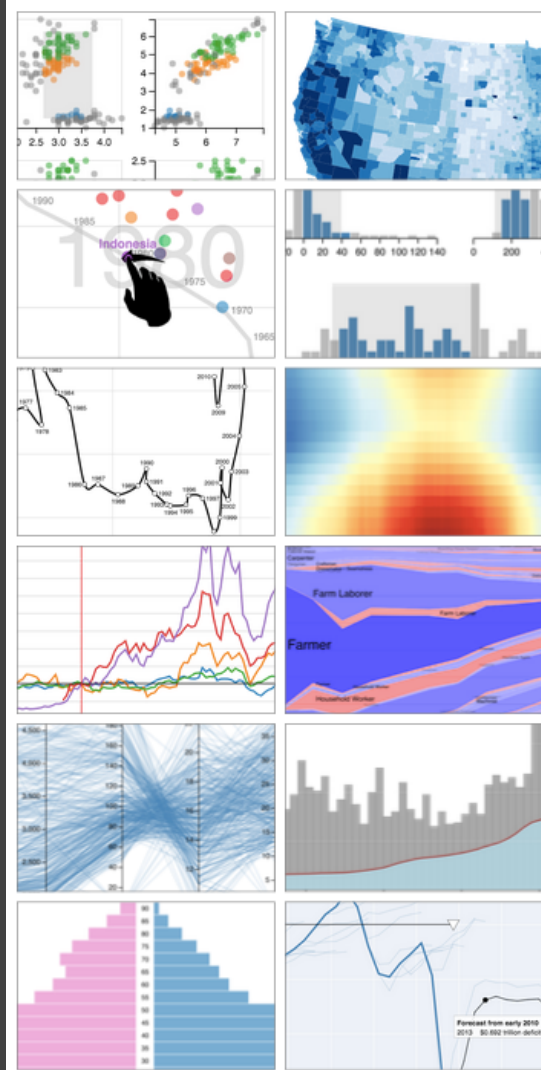
Session Outline

The Varieties of “Big Data”

Scalable Plotting Techniques

Scalable Interaction

Sampling Methods



The Varieties of “Big Data”

Tall Data

Lots of records

Large DBs have petabytes or more
(but median DB still fits in RAM!)

How to manage?

Parallel data processing

Reduction: Filter, aggregate

Sample or approximate

Not just about systems. Consider
perceptual / cognitive scalability.

Tall Data

Wide data

Lots of variables (100s-1000s...)

Select relevant subset

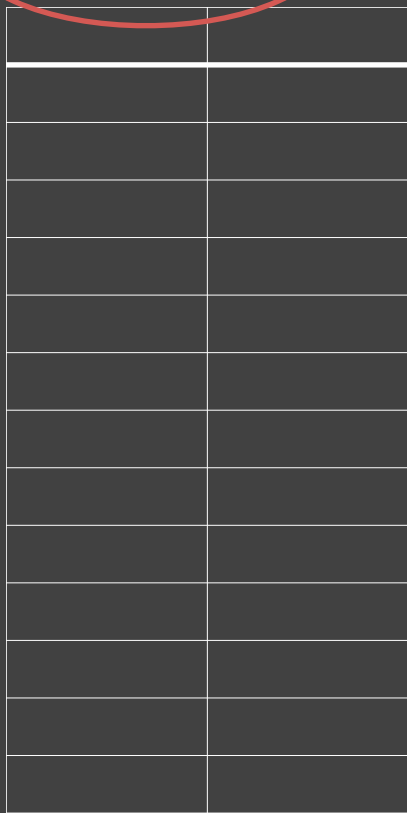
Dimensionality reduction

Statistical methods can suggest
and order related variables

Requires human judgment

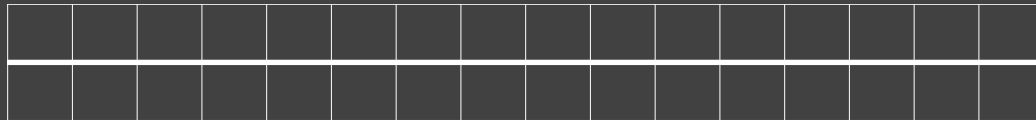
[illegible][illegible]

Tall Data



A diagram representing 'Tall Data' as a table with 2 columns and 20 rows. The word 'Tall' is circled in red in the header.

Wide data



A diagram representing 'Wide Data' as a table with 20 columns and 2 rows. The word 'Wide' is bolded in the header.

Diverse data



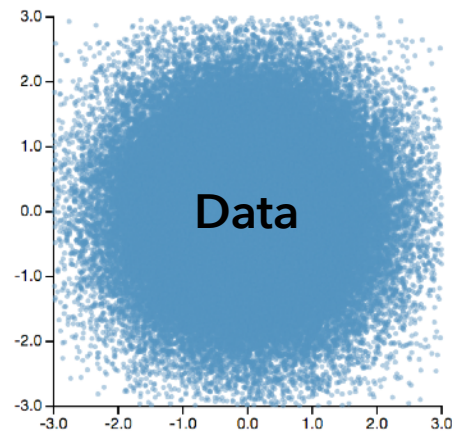
How can we visualize and
interact with **billion+ record**
databases in real-time?

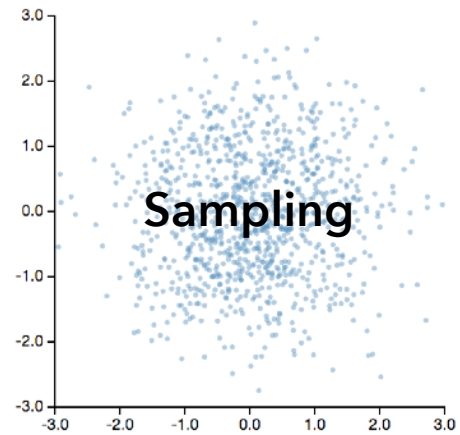
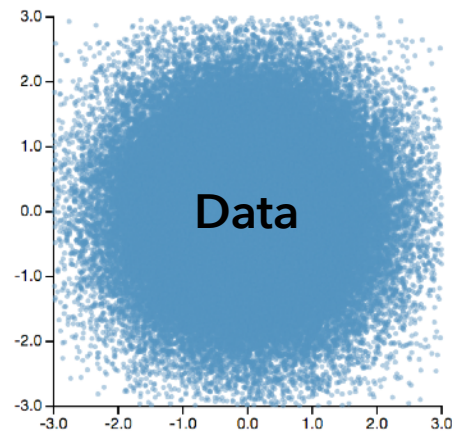
Two Challenges:

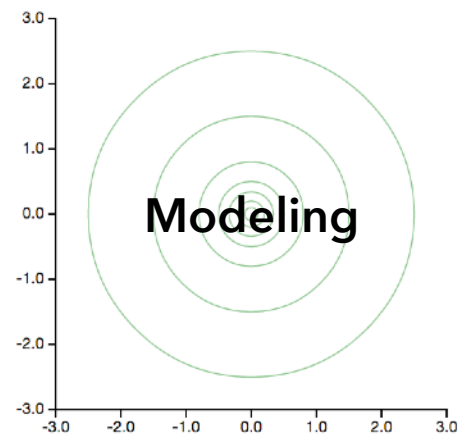
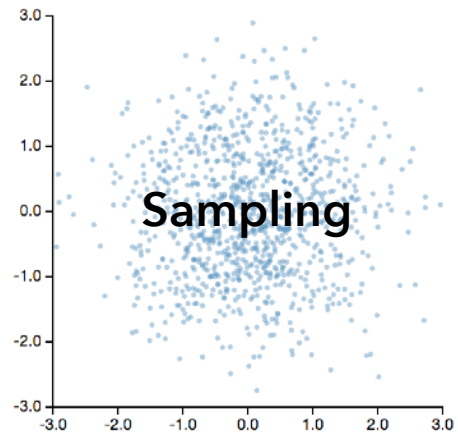
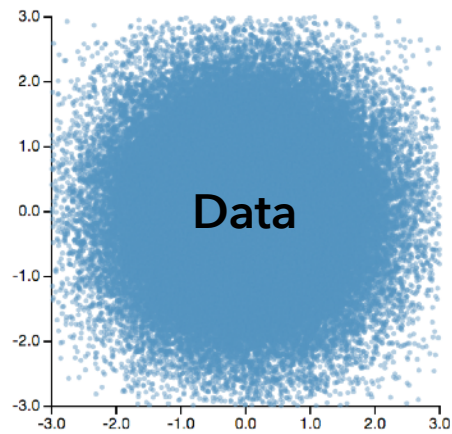
1. Effective **visual encoding**
2. Real-time **interaction**

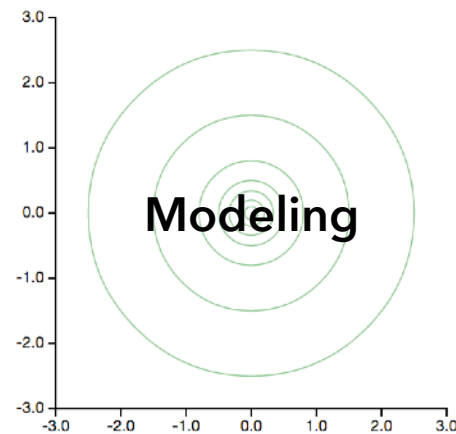
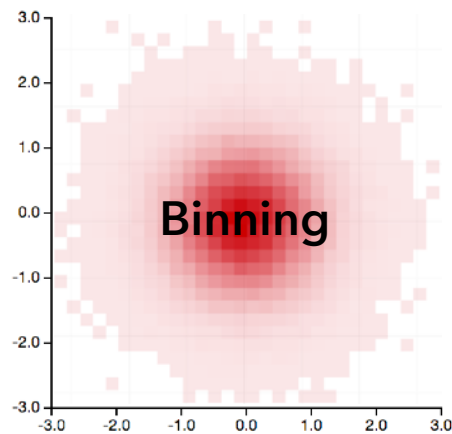
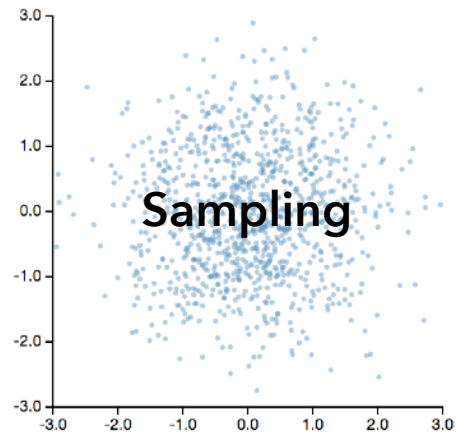
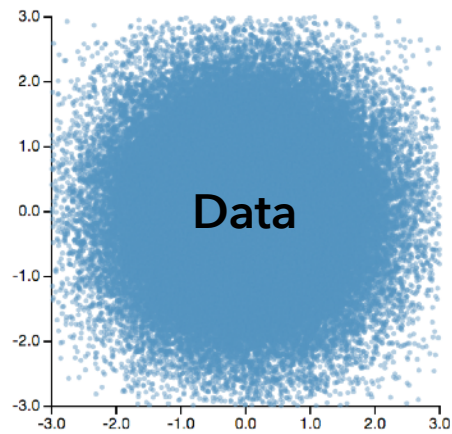
Perceptual and interactive scalability should be limited by the **chosen resolution** of the visualized data, not the number of records.

Scalable Plotting Techniques

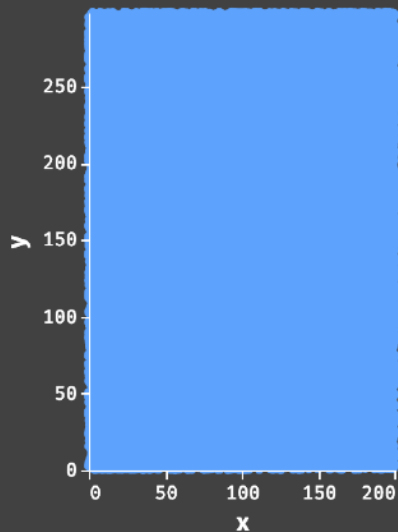




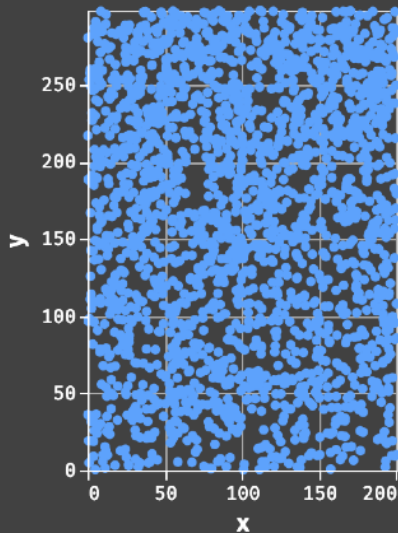




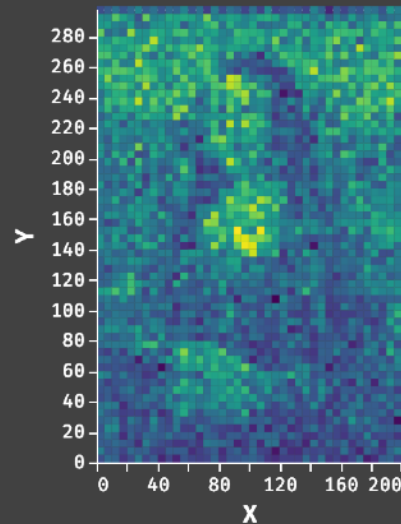
How to **Visualize** a Billion+ Records



Data



Sampling



Binned Aggregation

Decouple the visual complexity from the raw data through aggregation.

Bin > Aggregate (> Smooth) > Plot

1. Bin Divide data domain into discrete “buckets”

Categories: Already discrete (but watch out for high cardinality)

Numbers: Choose bin intervals (uniform, quantile, ...)

Time: Choose time unit: Hour, Day, Month, etc.

Geo: Bin x, y coordinates *after* cartographic projection

Bin > Aggregate (> Smooth) > Plot

1. Bin Divide data domain into discrete “buckets”

Categories: Already discrete (but watch out for high cardinality)

Numbers: Choose bin intervals (uniform, quantile, ...)

Time: Choose time unit: Hour, Day, Month, etc.

Geo: Bin x, y coordinates *after* cartographic projection

2. Aggregate Count, Sum, Average, Min, Max, ...

Bin > Aggregate (> Smooth) > Plot

1. Bin Divide data domain into discrete “buckets”

Categories: Already discrete (but watch out for high cardinality)

Numbers: Choose bin intervals (uniform, quantile, ...)

Time: Choose time unit: Hour, Day, Month, etc.

Geo: Bin x, y coordinates *after* cartographic projection

2. Aggregate Count, Sum, Average, Min, Max, ...

3. Smooth Optional: smooth aggregates [Wickham '13]

Bin > Aggregate (> Smooth) > Plot

1. Bin Divide data domain into discrete “buckets”

Categories: Already discrete (but watch out for high cardinality)

Numbers: Choose bin intervals (uniform, quantile, ...)

Time: Choose time unit: Hour, Day, Month, etc.

Geo: Bin x, y coordinates *after* cartographic projection

2. Aggregate Count, Sum, Average, Min, Max, ...

3. Smooth Optional: smooth aggregates [Wickham '13]

4. Plot Visualize the aggregate values

Binned Plots by Data Type

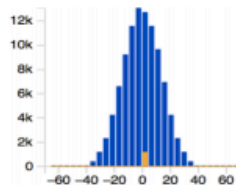
Numeric

Ordinal

Temporal

Geographic

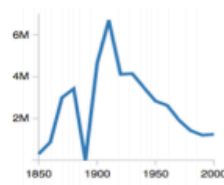
1D



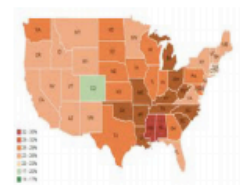
Histogram



Bar Chart

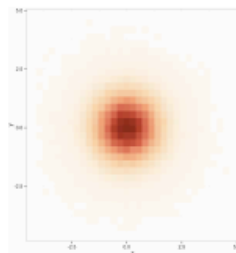


Line Graph /
Area Chart

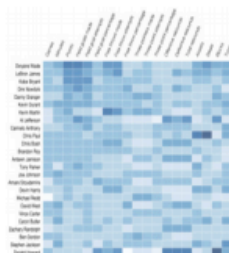


Choropleth Map

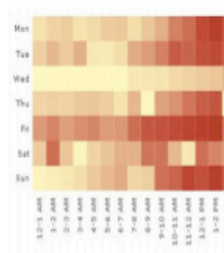
2D



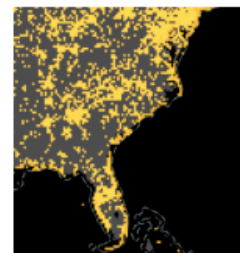
Binned
Scatter Plot



Heatmap



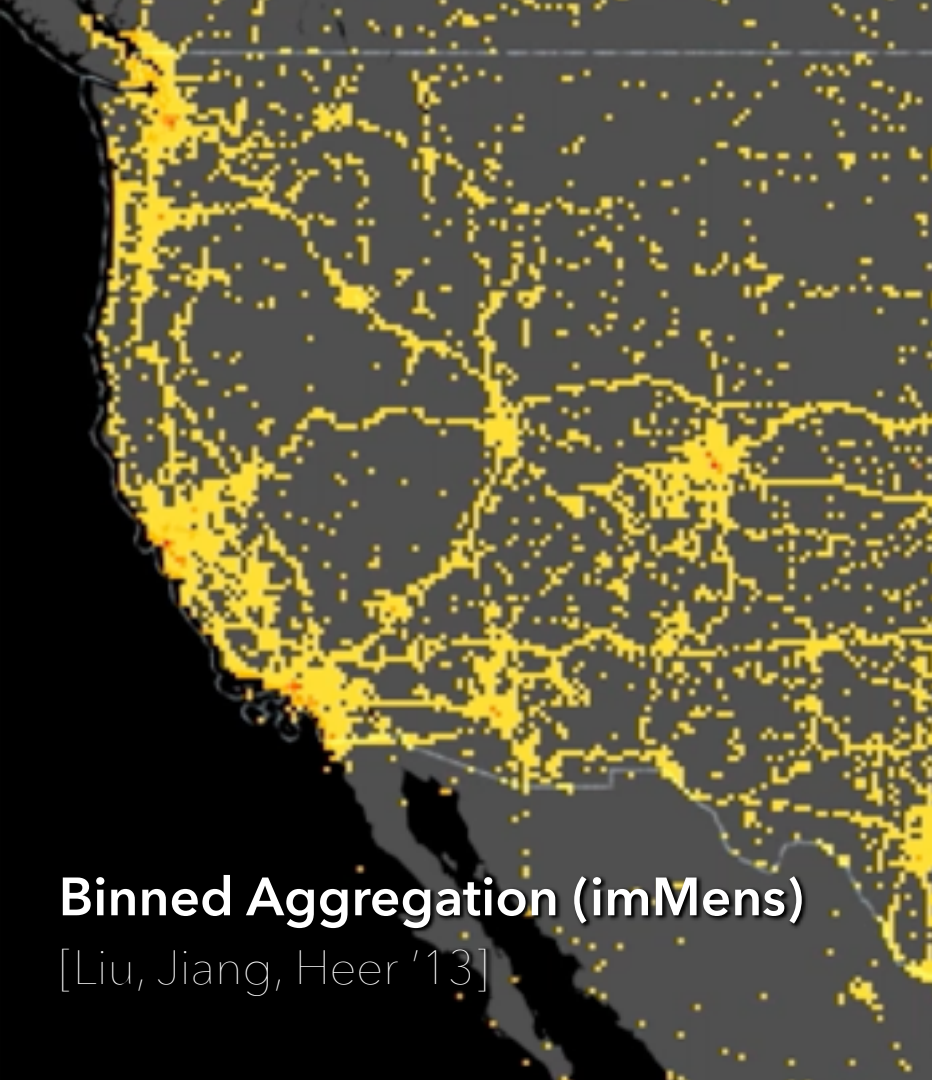
Temporal
Heatmap



Geographic
Heatmap

Examples



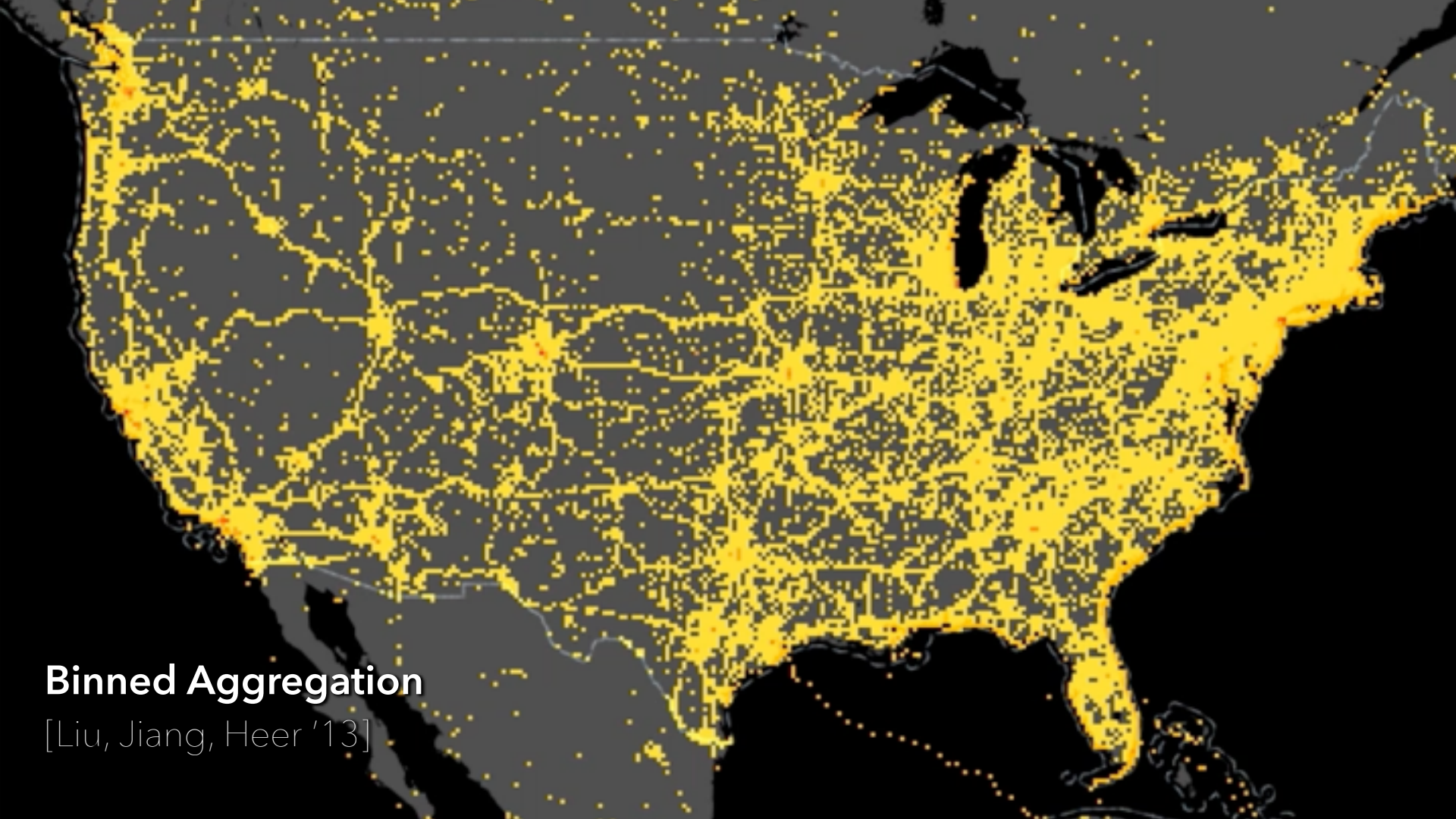


Binned Aggregation (imMens)

[Liu, Jiang, Heer '13]



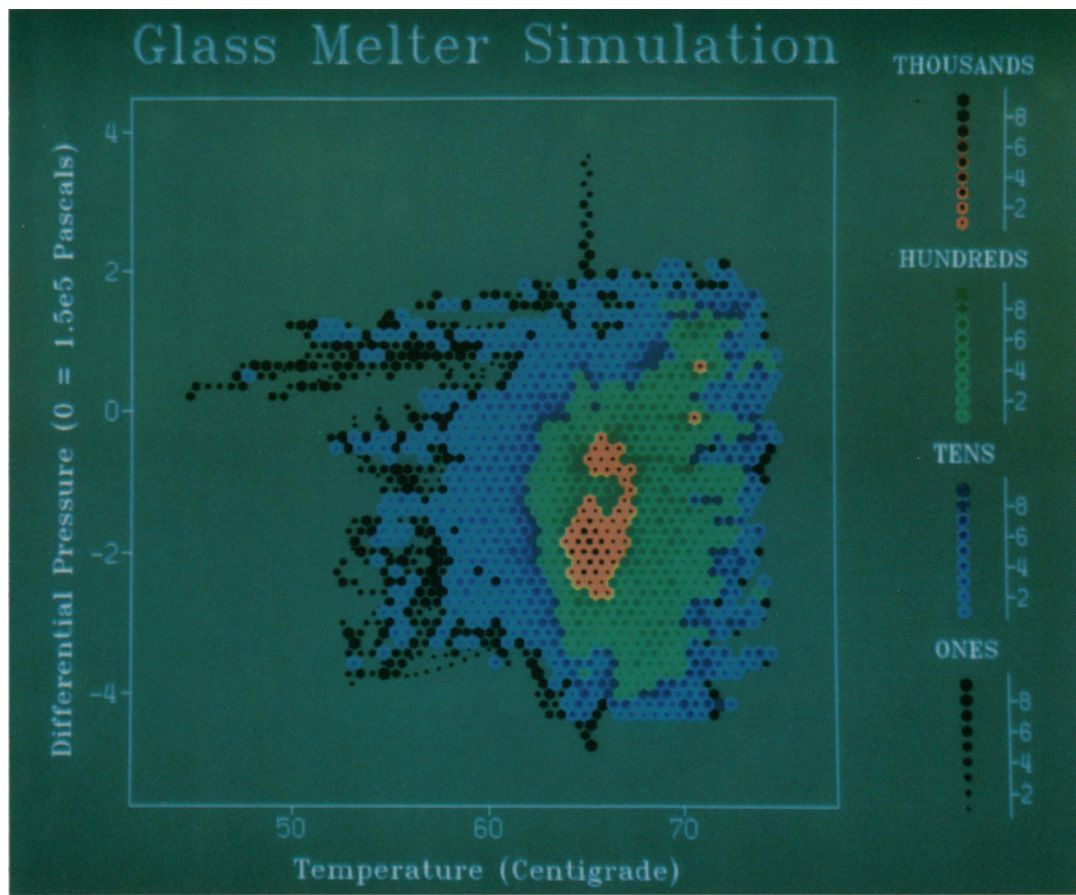
Sampling Google Fusion Tables



Binned Aggregation

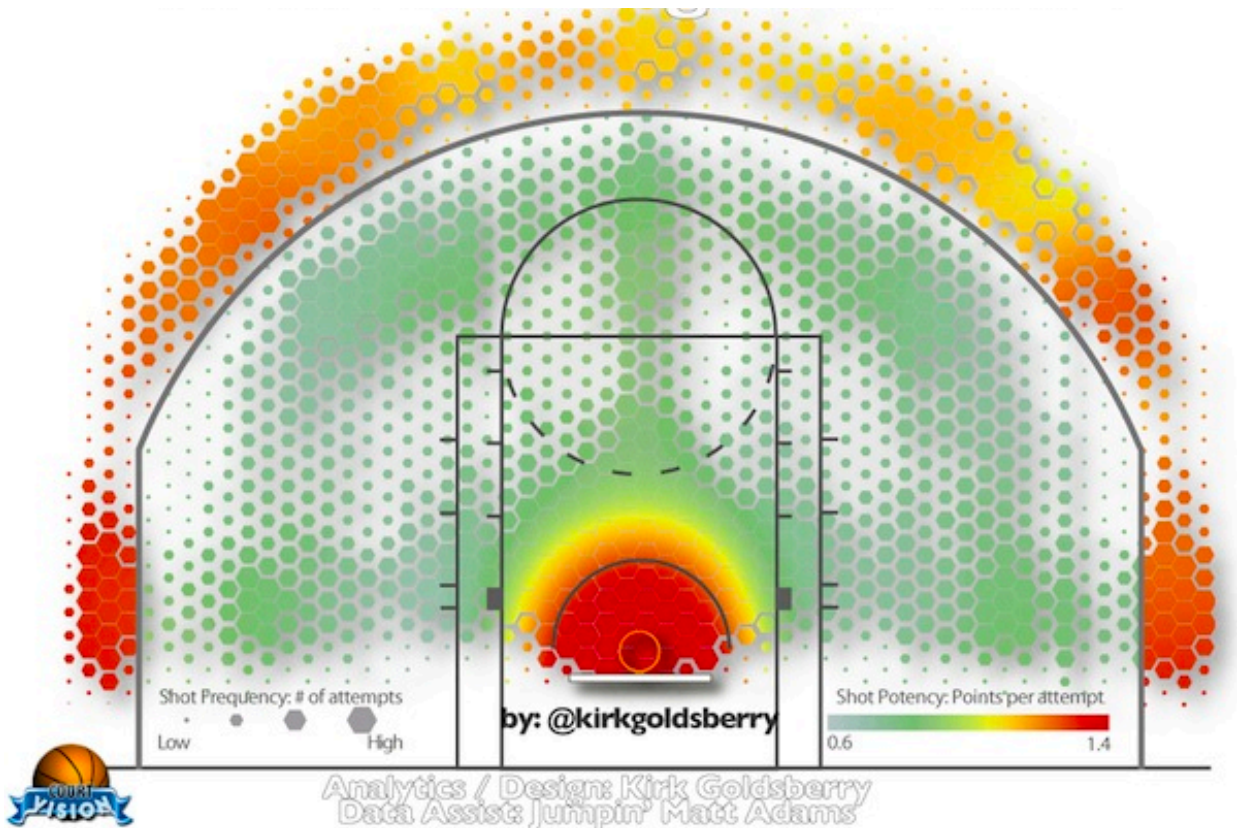
[Liu, Jiang, Heer '13]

Example: Binned Scatter Plots



Scatterplot
Matrix
Techniques
for Large N
[Carr et al. '87]

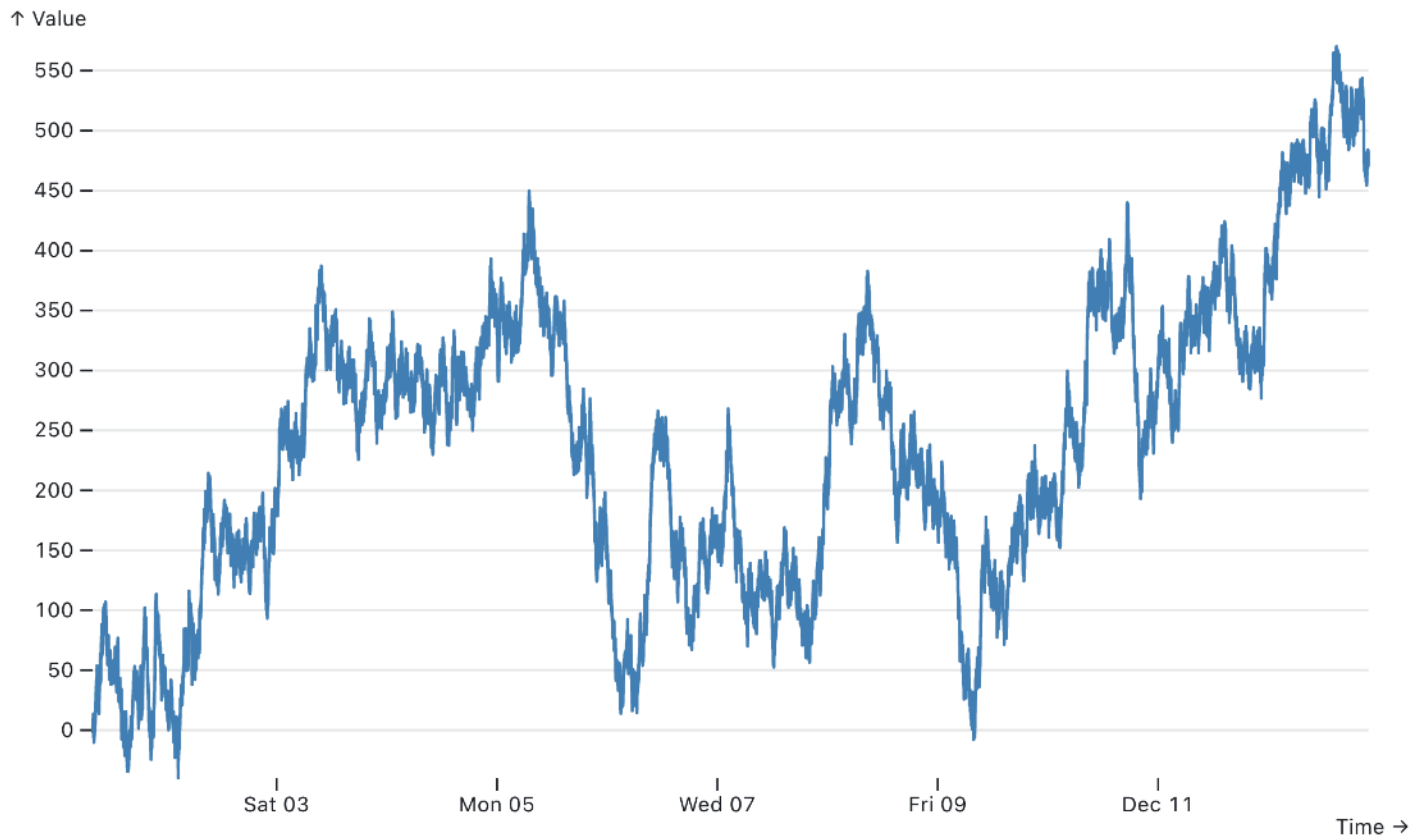
Example: Basketball Shot Chart



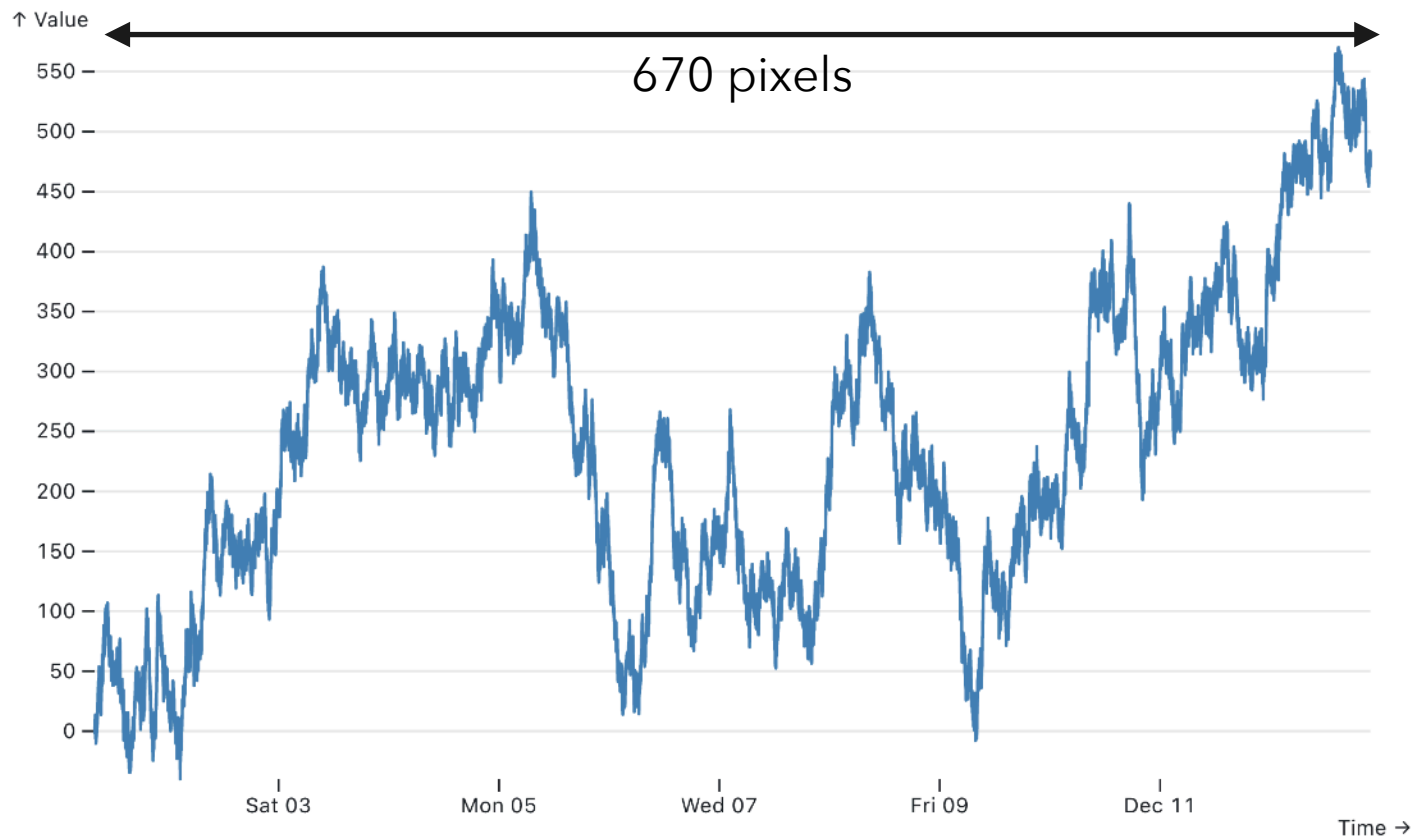
NBA Shooting 2011-12
[Goldsberry]

Time Series

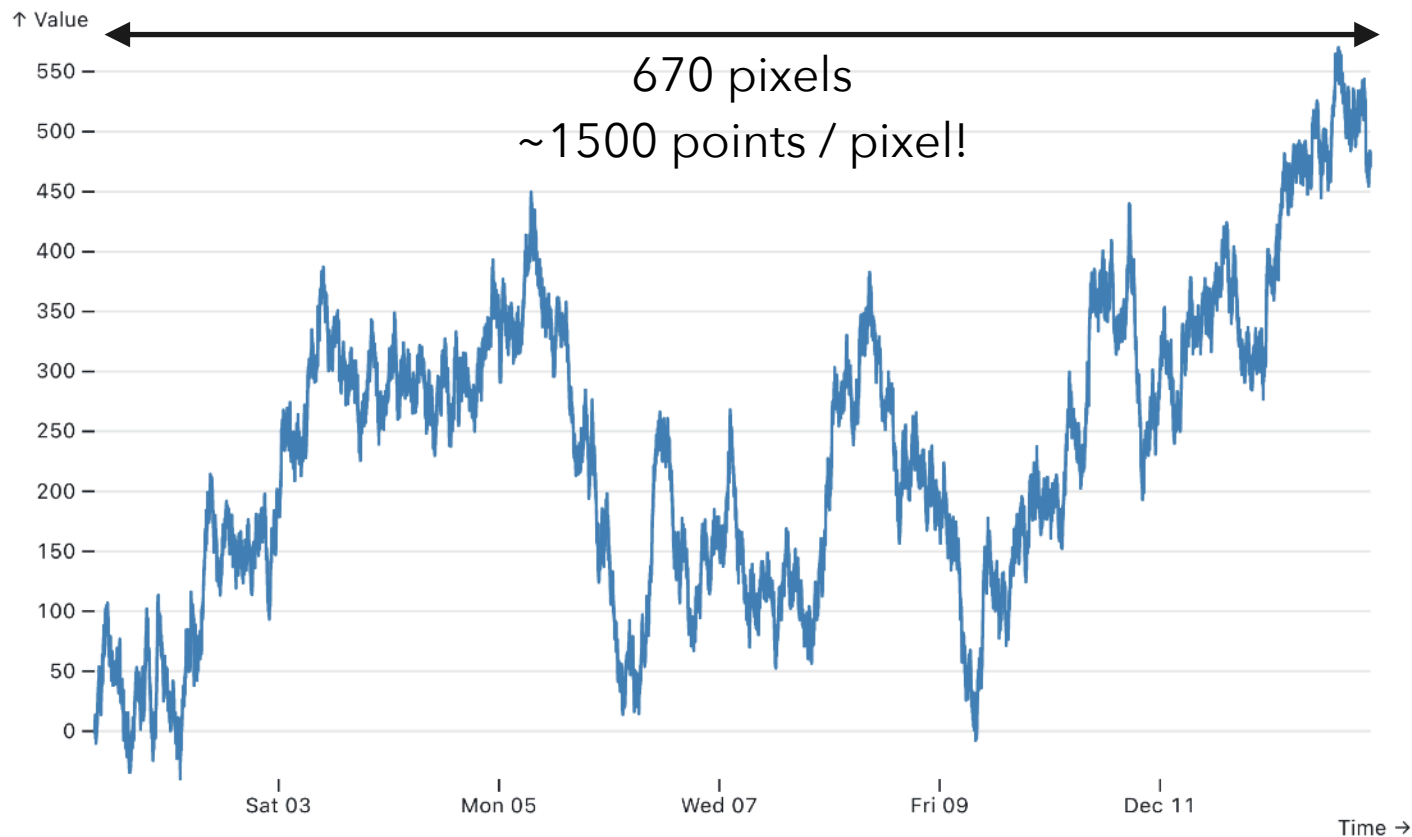
Time Series: 1M samples, 1 sample/second



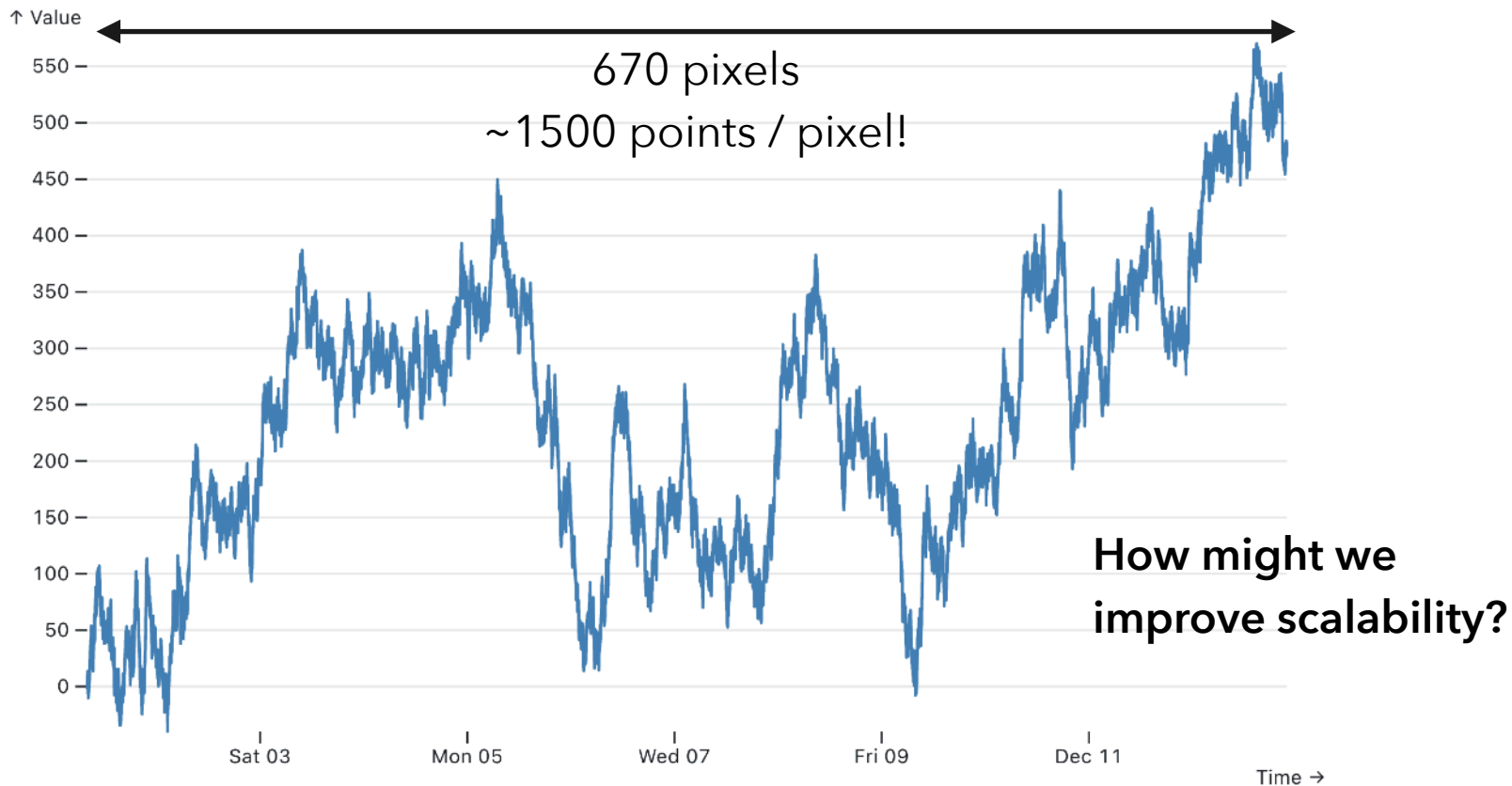
Time Series: 1M samples, 1 sample/second



Time Series: 1M samples, 1 sample/second



Time Series: 1M samples, 1 sample/second



Time-Series Aggregation [Jugel'14]



Insight: the resolution is bound by the number of pixels.

Time-Series Aggregation [Jugel'14]



Insight: the resolution is bound by the number of pixels.

1. Compute average value per pixel (1 point/pixel)
...this may miss extreme (min, max) values



Time-Series Aggregation [Jugel'14]



Insight: the resolution is bound by the number of pixels.

1. Compute average value per pixel (1 point/pixel)

...this may miss extreme (min, max) values



2. Plot min/max values per pixel (2 points/pixel)

...this does better, but still misrepresents



Time-Series Aggregation [Jugel'14]



Insight: the resolution is bound by the number of pixels.

1. Compute average value per pixel (1 point/pixel)

...this may miss extreme (min, max) values



2. Plot min/max values per pixel (2 points/pixel)

...this does better, but still misrepresents



3. [M4](#): min/max values & timestamps (4 points/pixel)

...this provides provable fidelity to the full data!



M4 Data Reduction in the Database

```
SELECT min(t), arg_min(v,t) FROM Q GROUP BY $pixel UNION  
SELECT max(t), arg_max(v,t) FROM Q GROUP BY $pixel UNION  
SELECT arg_min(t,v), min(v) FROM Q GROUP BY $pixel UNION  
SELECT arg_max(t,v), max(v) FROM Q GROUP BY $pixel
```

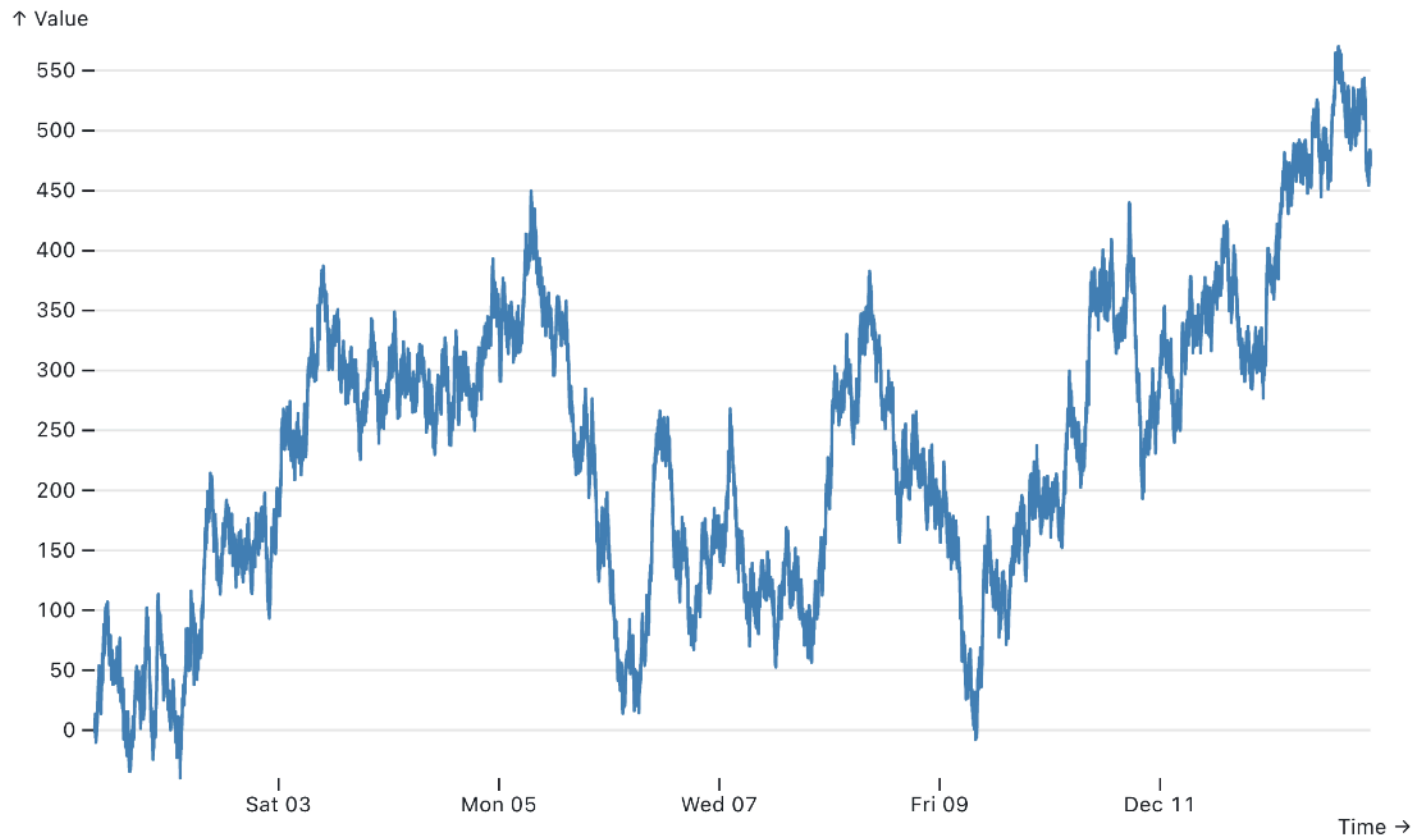
Q: query that returns a time series (t, v)

\$t1, \$t2: global min/max timestamps

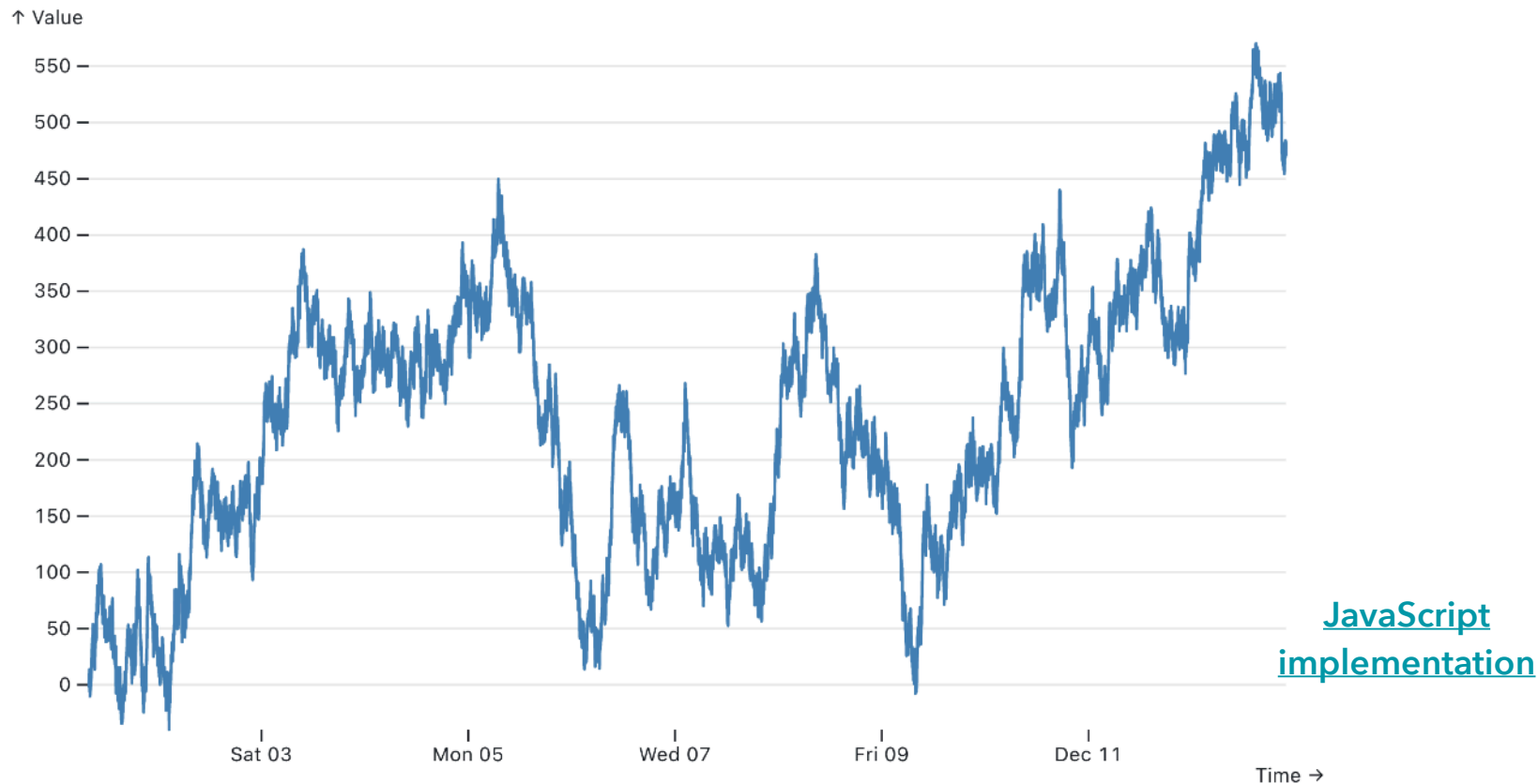
\$w: chart width in pixels

$\text{\$pixel} = \text{floor}(\text{\$w}(\text{t} - \text{\$t1}) / (\text{\$t2} - \text{\$t1}))$

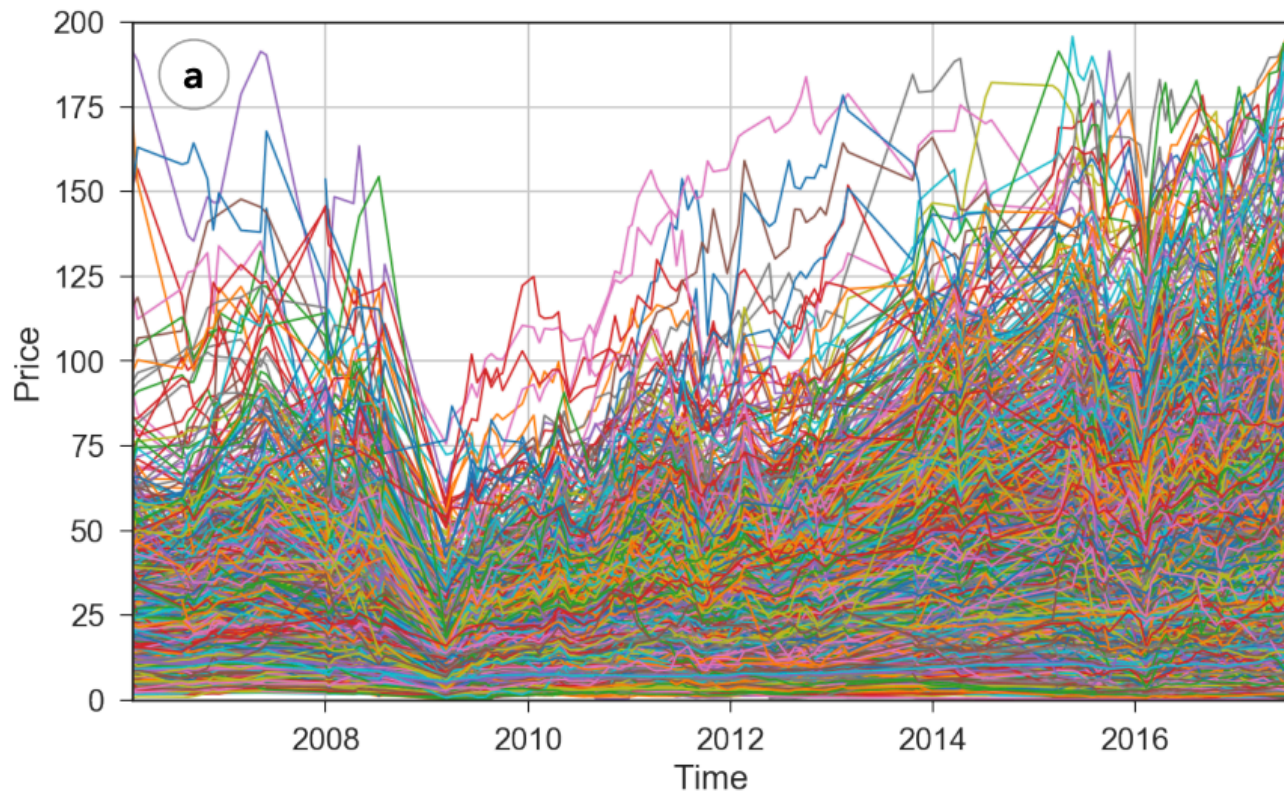
Time Series: 1M samples, 1 sample/second



M4: 1M samples -> 2,653 plotted points

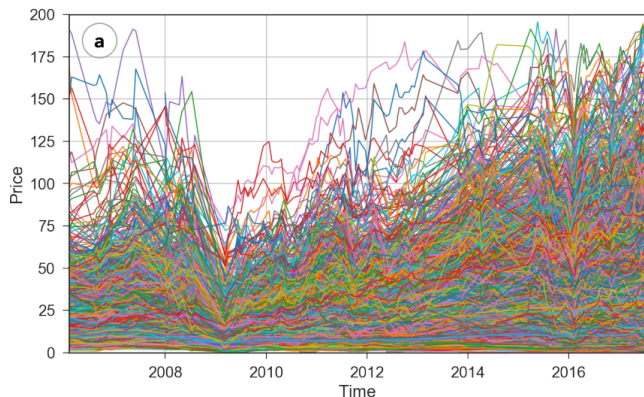


But what about multiple time-series?

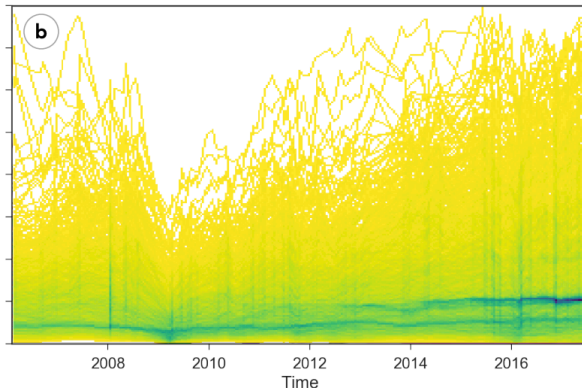


Perceptual scalability
breaks down...

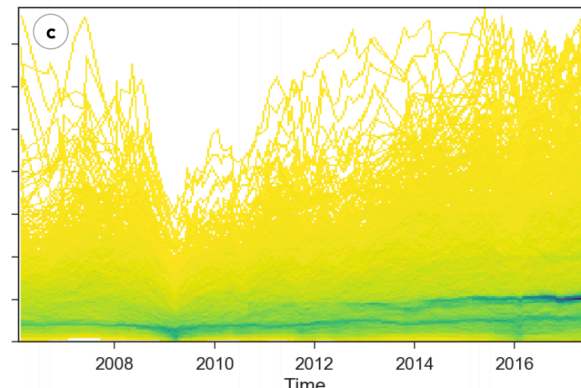
Density Line Chart [Moritz & Fisher]



Line Chart



Non-Normalized Heatmap



Normalized "DenseLines"

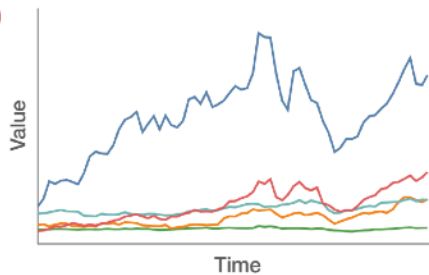
The non-normalized heatmap suffers from artifacts, seen as vertical stripes.

Binned charts convey high points across the top, a collective dip in stocks during the crash of 2008, and two distinct bands of \$25 and \$15 stocks.

Density Line Chart [Moritz & Fisher]

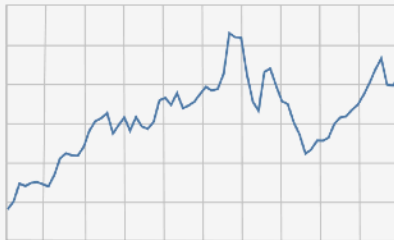
A

Time Series



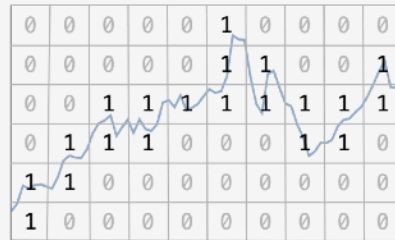
Repeat for each series

B.1



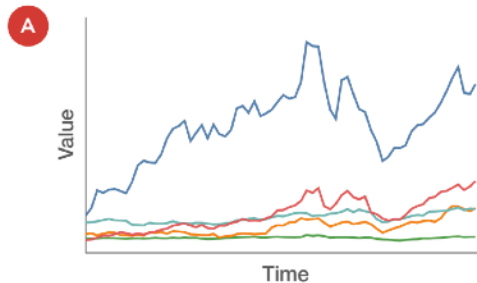
B.2

Non-Normalized



Sum: 2 2 2 2 1 3 2 2 2 2

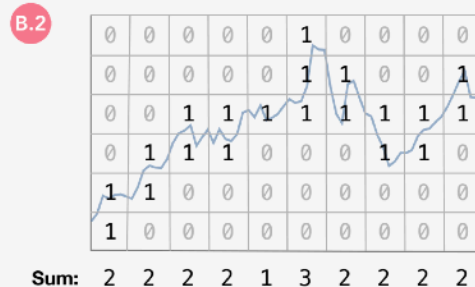
Density Line Chart [Moritz & Fisher]



Repeat for each series



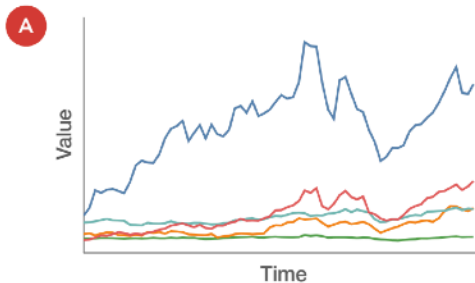
Non-Normalized



Approx. Arc-Length Normalized

Density Line Chart [Moritz & Fisher]

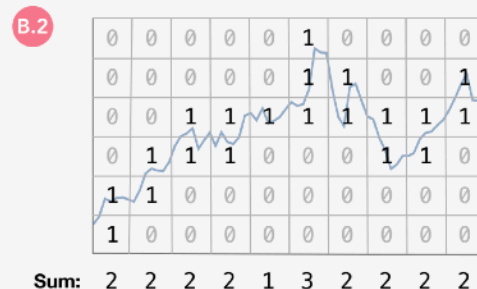
Time Series



Repeat for each series



Non-Normalized



B.3

0	0	0	0	0	0.3	0	0	0	0
0	0	0	0	0	0.3	0.5	0	0	0.5
0	0	0.5	0.5	0.5	0.3	0.5	0.5	0.5	0.5
0	0.5	0.5	0.5	0	0	0	0.5	0.5	0
0.5	0.5	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0

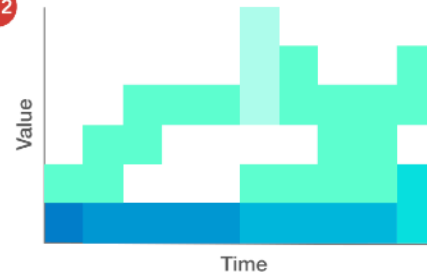
Approx. Arc-Length Normalized

C.1

					0.3				
					0.3	0.5			0.5
			0.5	0.5	0.5	0.3	0.5	0.5	0.5
		0.5	0.5				0.5	0.5	
0.5	0.5					0.5	0.5	0.5	2
4.5	4	4	4	4	3.5	3.5	3.5	3.5	2

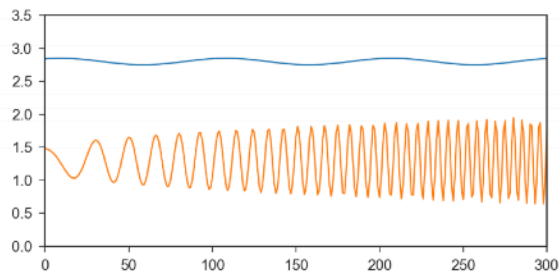
Aggregate

C.2

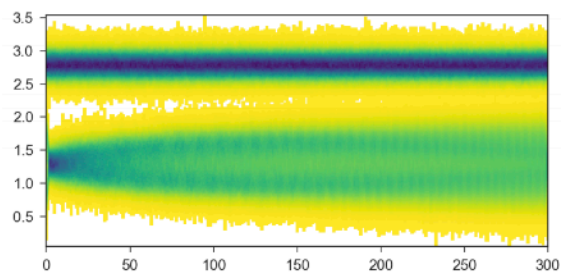


Color

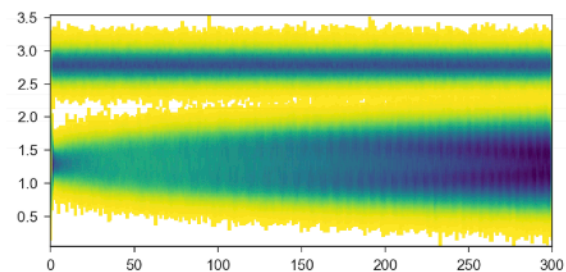
Density Line Chart [Moritz & Fisher]



Example Time Series



10k Series, Normalized



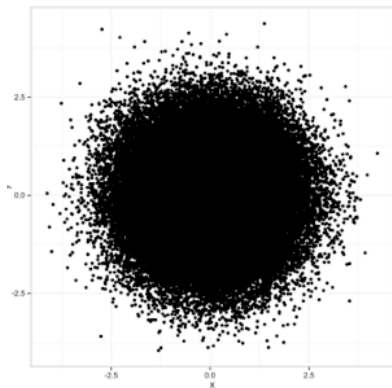
10k Series, Non-Normalized

The density of the second group appears to increase to the right!

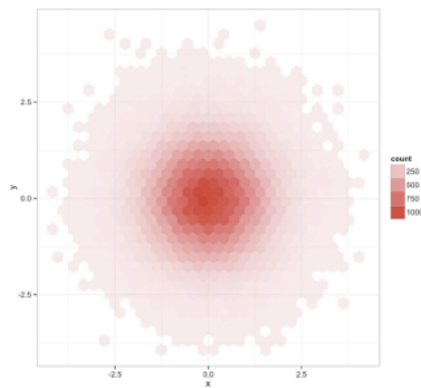
Without normalization, the steep lines are over-represented.

Design Subtleties

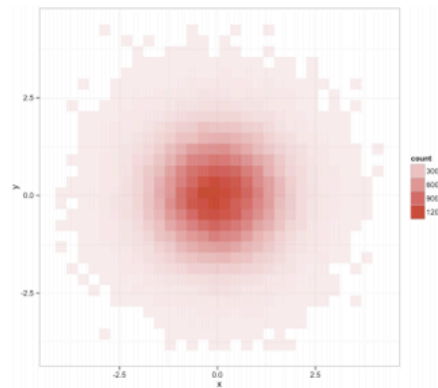
Hexagonal or Rectangular Bins?



100,000 Data Points



Hexagonal Bins

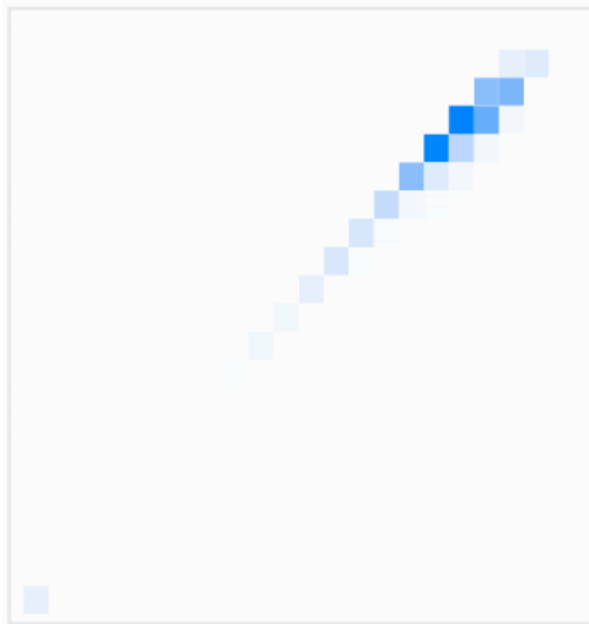


Rectangular Bins

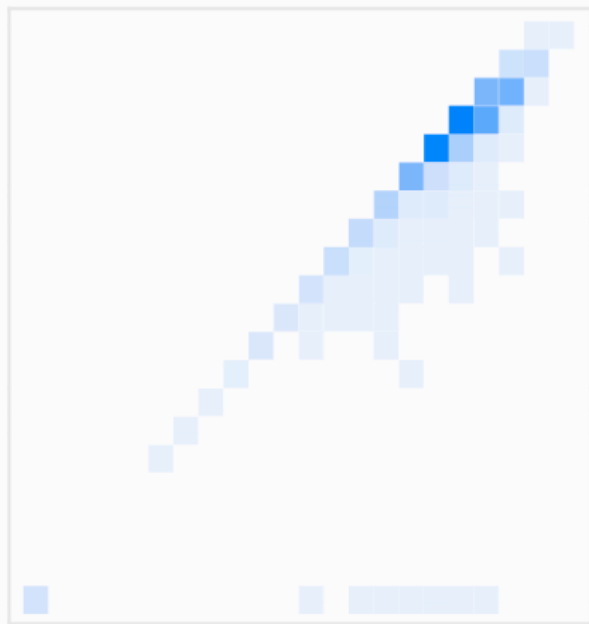
Hex bins better estimate density for 2D plots,
but the *improvement is marginal* [Scott 92].

Rectangles support *reuse* and *visual queries*.

Color Scale: Discontinuity after Zero



Standard Color Ramp
Counts near zero are white.



Add Discontinuity after Zero
Counts near zero remain visible.

Questions?

Administrivia

Final Project Schedule

~~Proposal~~ ————— ~~Fri Nov 7~~

~~Prototype~~ ————— ~~Wed Nov 19~~

Demo Video **Wed Dec 3**

Video Showcase Thu Dec 4 (in class)

Deliverables Mon Dec 8

Logistics

Upload your video to YouTube (unlisted is fine)

Submit the video URL on Gradescope

Be sure to include all team members!

Demo Video Guidelines

Your video should communicate your chosen topic and goals along with your visualization designs.

Typically videos use a mixture of static slides and interactive screen capture with overlaid narration.

The initial frame of your video should include your project name and the team members' names.

You might show your page as-is, or you might take excerpts (cropped views) of your page for a better video narrative. Whatever communicates best.

Demo Video Guidelines, Cont.

Your video should communicate how your designs enable understanding of your chosen topic & data.

Do not laundry list the various features you implemented. Instead focus on what viewers can learn from your submission.

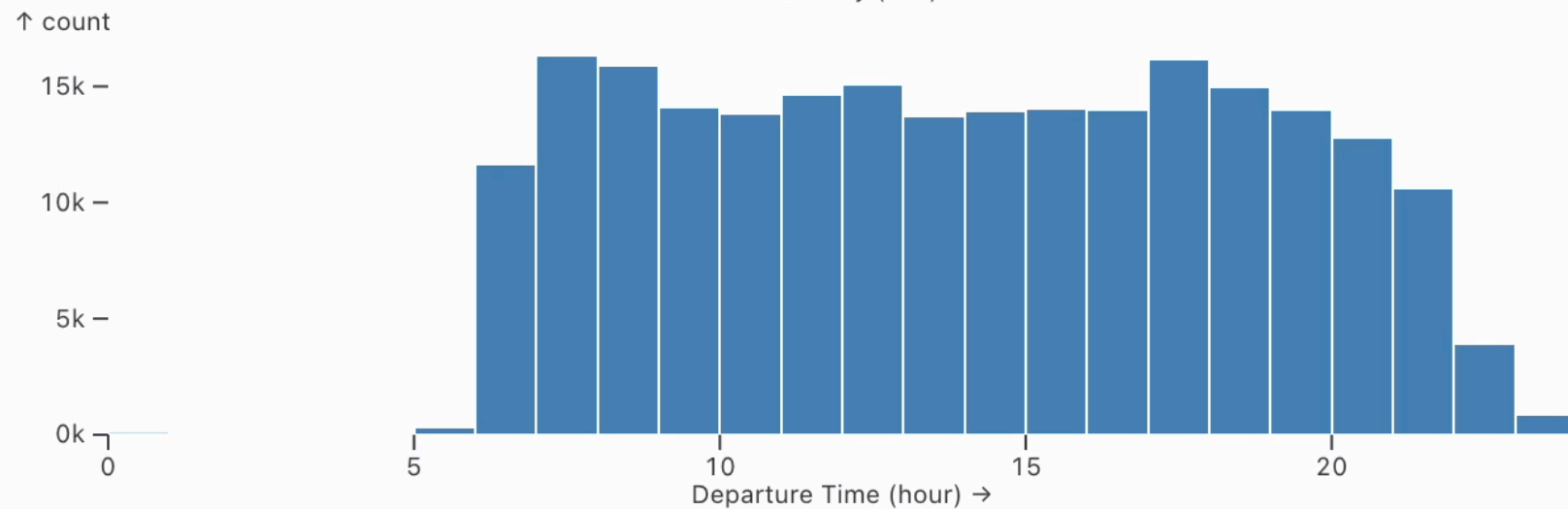
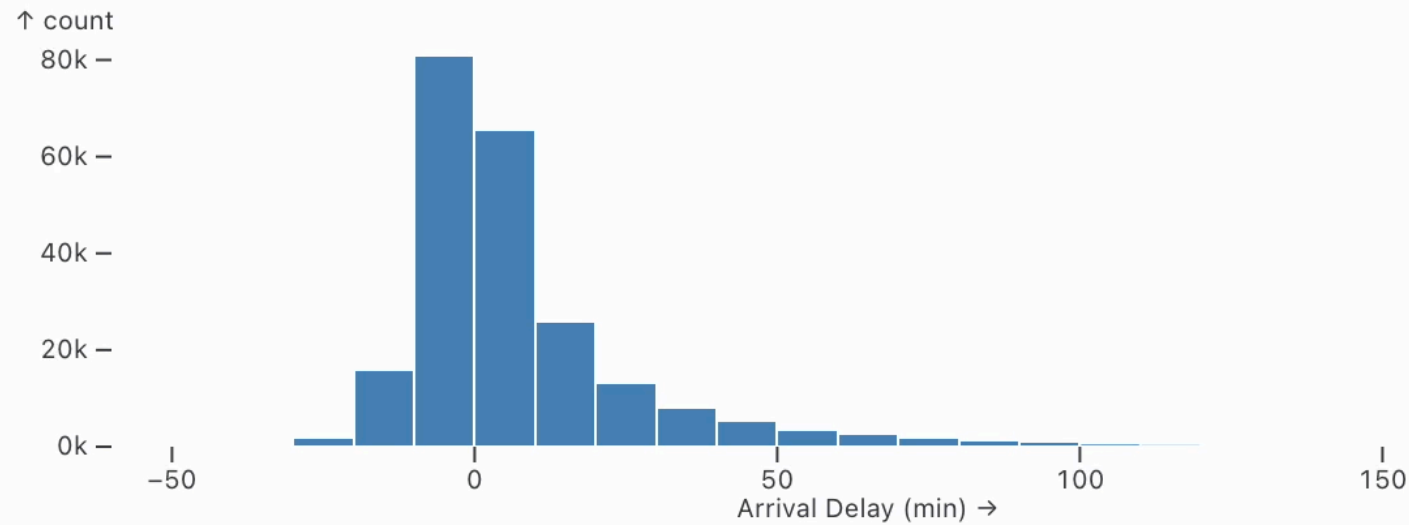
Walk us through an envisioned use case from the perspective of a viewer, demonstrating the kind of insights/explanations one might gain.

Keep it tight! 90 seconds goes quickly :)

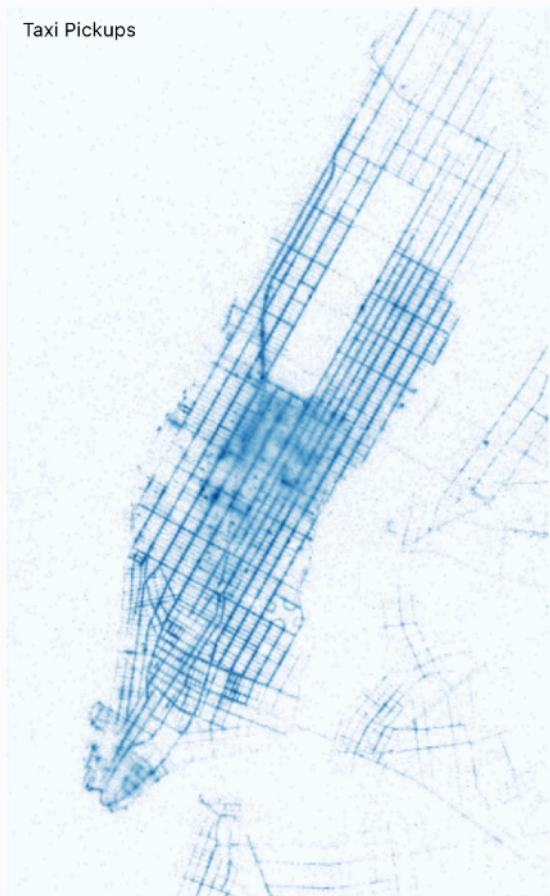
Scalable Interaction

Flight Delays

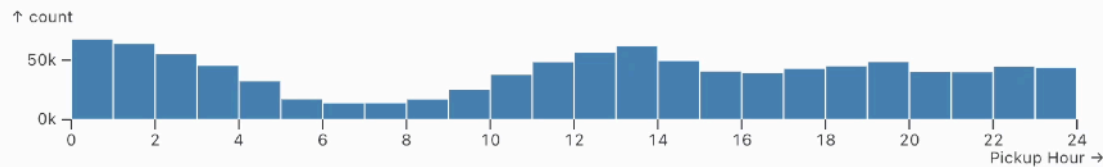
250k Records



Taxi Pickups



Taxi Dropoffs

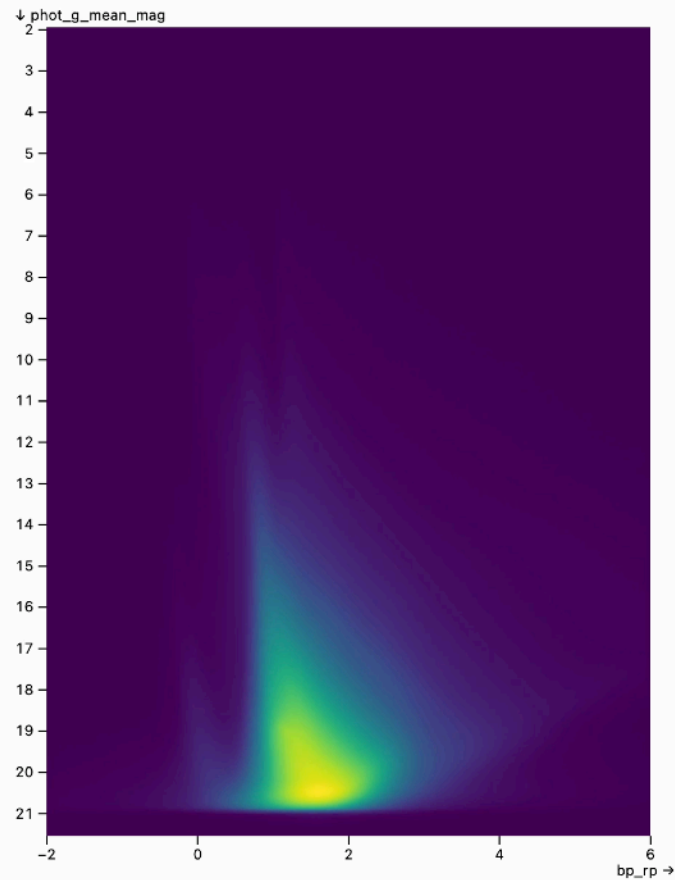
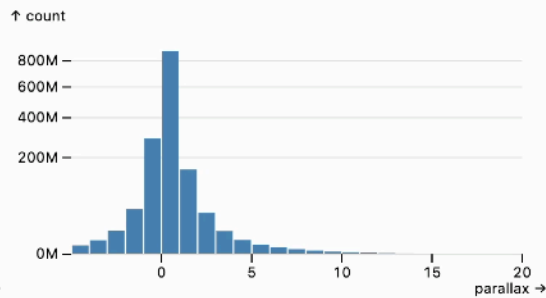
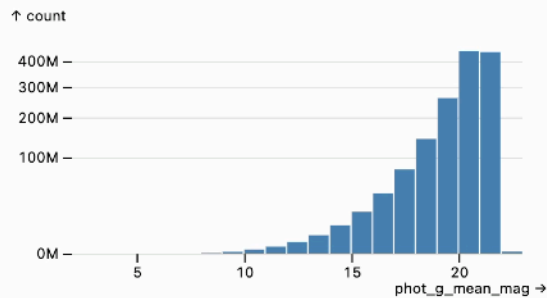
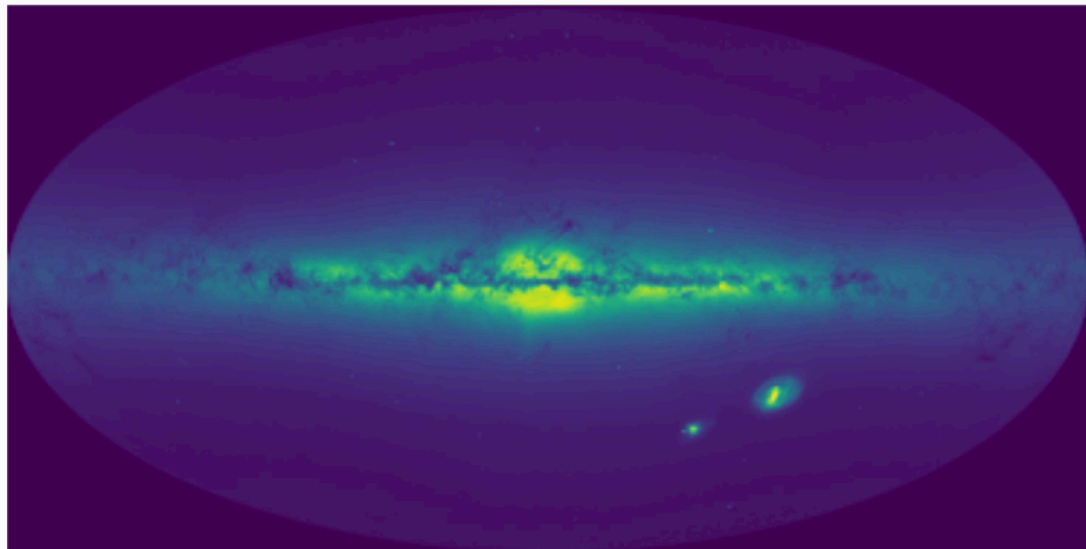


NY Taxi Rides

1M Records

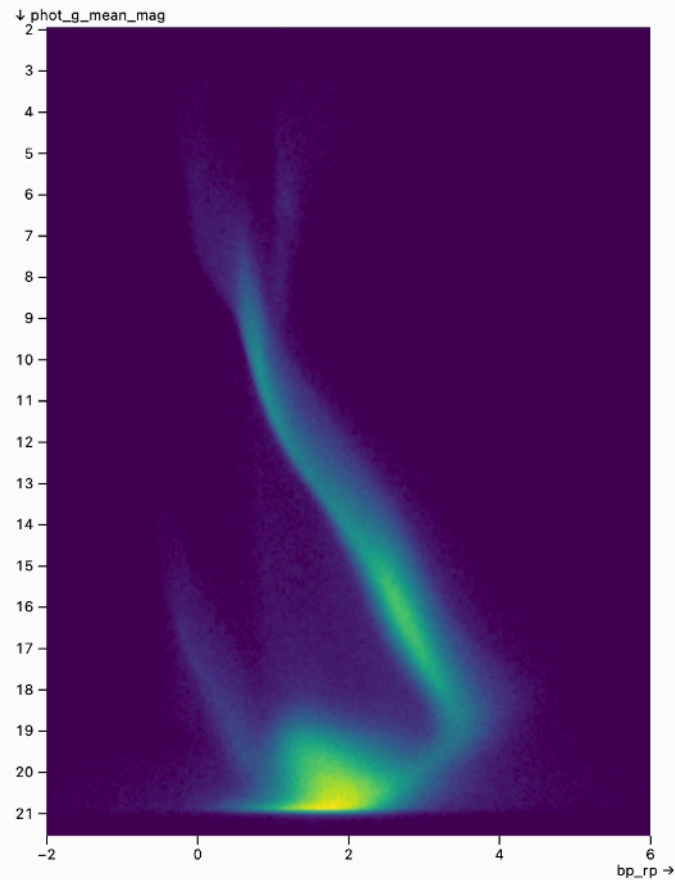
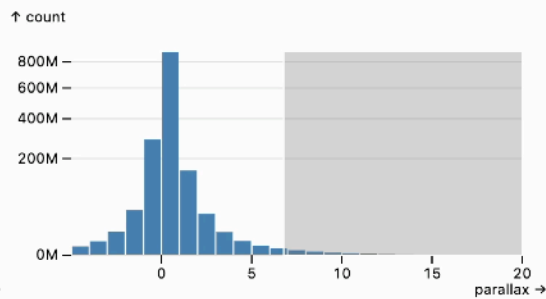
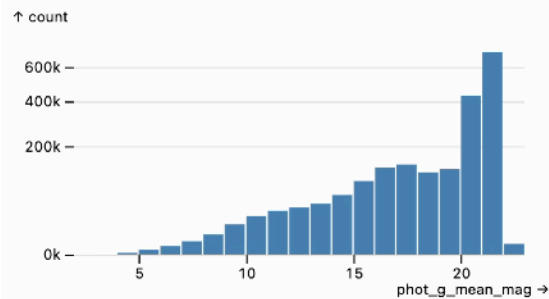
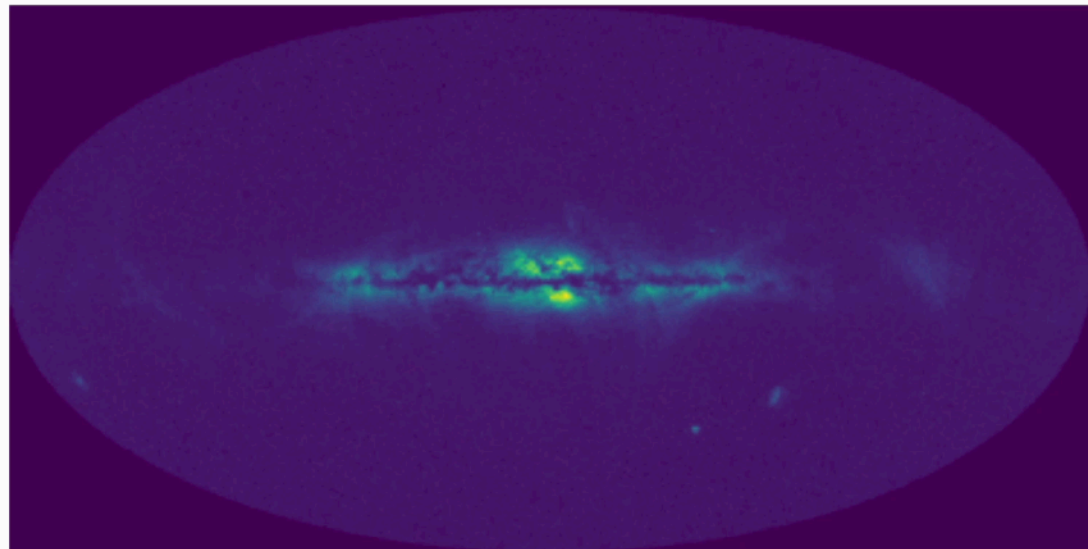
Jan 1-3, 2010

Sample Size Bin Width Color



Gaia Star Catalog · 1.8B Records

Sample Size Bin Width Color



Gaia Star Catalog · 1.8B Records

Interactive Scalability Strategies

1. Query Database
2. Indexing / Preaggregation
3. Prefetching
4. Approximation

Interactive Scalability Strategies

1. Query Database Offload to a scalable backend...

Tableau, for example, issues aggregation queries.

Analytical databases are designed for fast, parallel execution.

But round-trip queries to the DB may still be too slow...

2. Indexing / Preaggregation

3. Prefetching

4. Approximation

Interactive Scalability Strategies

1. Query Database ...or alternative data frame implementation

Python: [Polars](#), [Vaex](#), [Modin](#), [cuDF](#)

R: [dbplyr](#)

All: [DuckDB](#)

2. Indexing / Preaggregation

3. Prefetching

4. Approximation

Interactive Scalability Strategies

1. Query Database

2. Indexing / Preaggregation Query data summaries

Build sorted indices or pre-aggregated data to quickly re-calculate aggregations as needed on the client.

3. Prefetching

4. Approximation

Interactive Scalability Strategies

1. Query Database

2. Indexing / Preaggregation

3. **Prefetching** Request data *before* it is needed

Reduce latency by speculatively querying for data before it is needed. Requires prediction models to guess what is needed.

4. **Approximation**

Interactive Scalability Strategies

1. Query Database

2. Indexing / Preaggregation

3. Prefetching

4. **Approximation** Give fast, approximate answers

Reduce latency by computing aggregates on a sample, ideally with approximation bounds characterizing the error.

Interactive Scalability Strategies

1. Query Database
2. Indexing / Preaggregation
3. Prefetching
4. Approximation

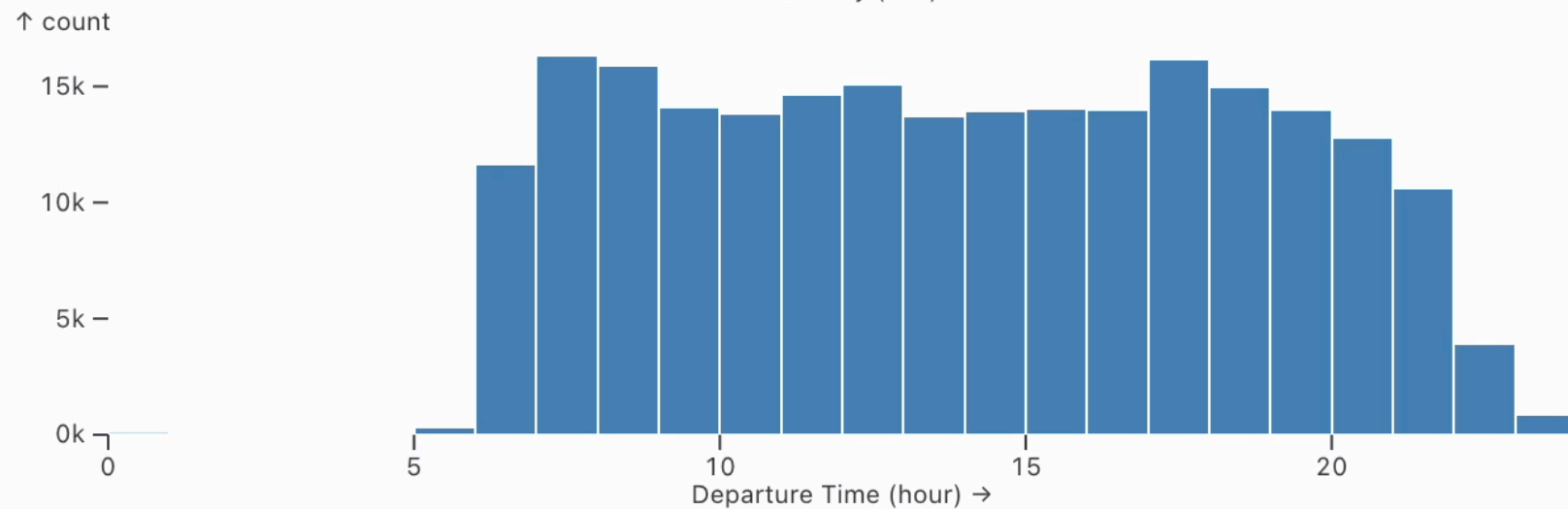
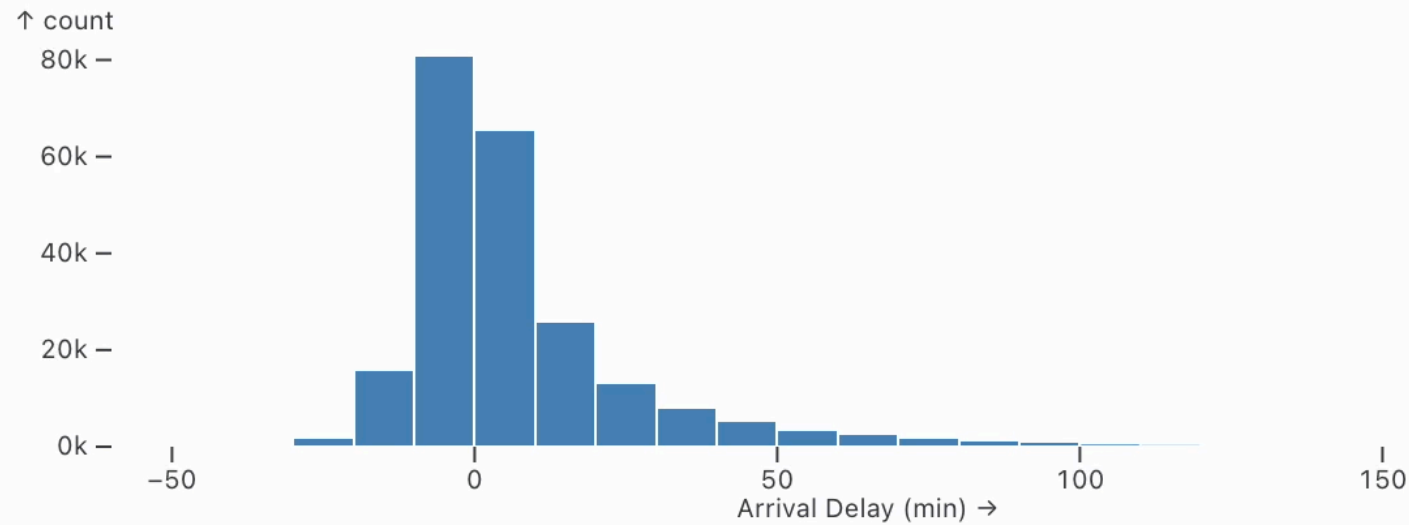
These strategies are **not** mutually exclusive!

Systems can apply them in tandem.

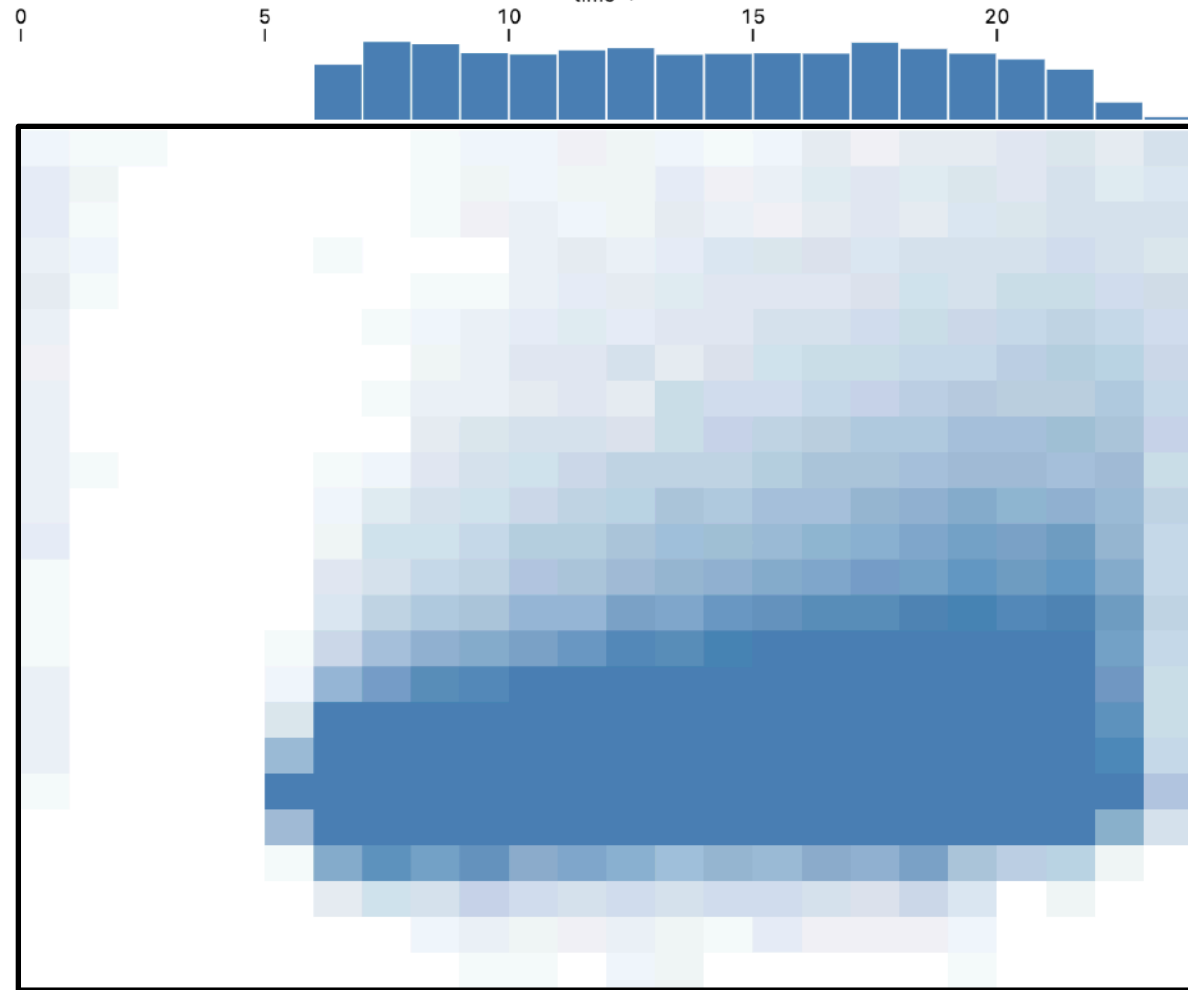
Preaggregation

Flight Delays

250k Records



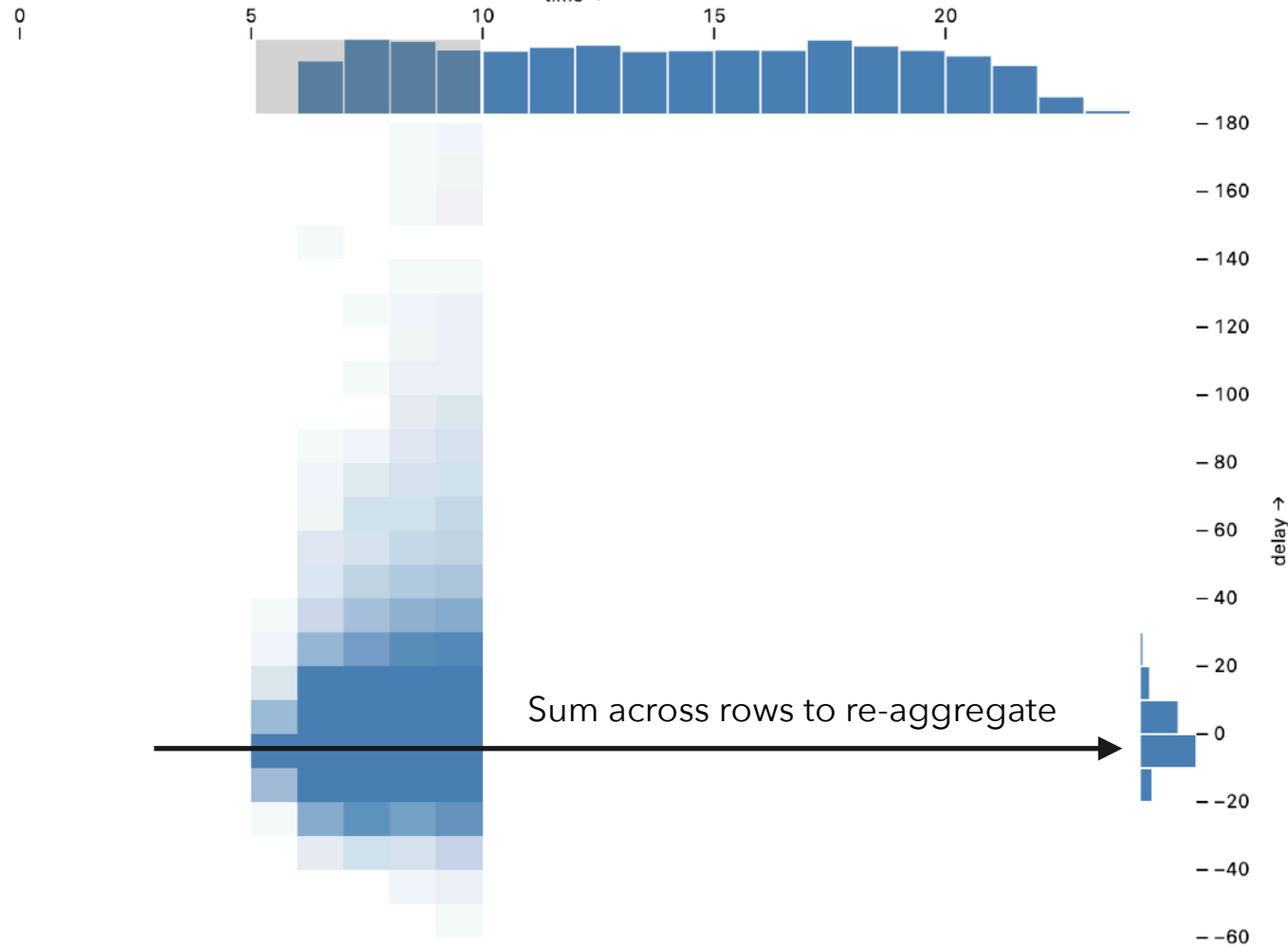
Time Resolution Delay Resolution



Flight Delays
250k Records

Preaggregate

Time Resolution Delay Resolution



Flight Delays
250k Records

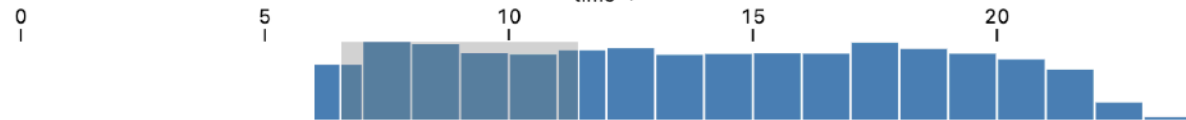
Time Resolution

bins

Delay Resolution

bins

time →



Flight Delays

250k Records

- 180

- 160

- 140

- 120

- 100

- 80

- 60

- 40

- 20

- 0

- -20

- -40

- -60

delay →



Time Resolution

bins ▾

Delay Resolution

bins ▾

time →

0
|

5
|

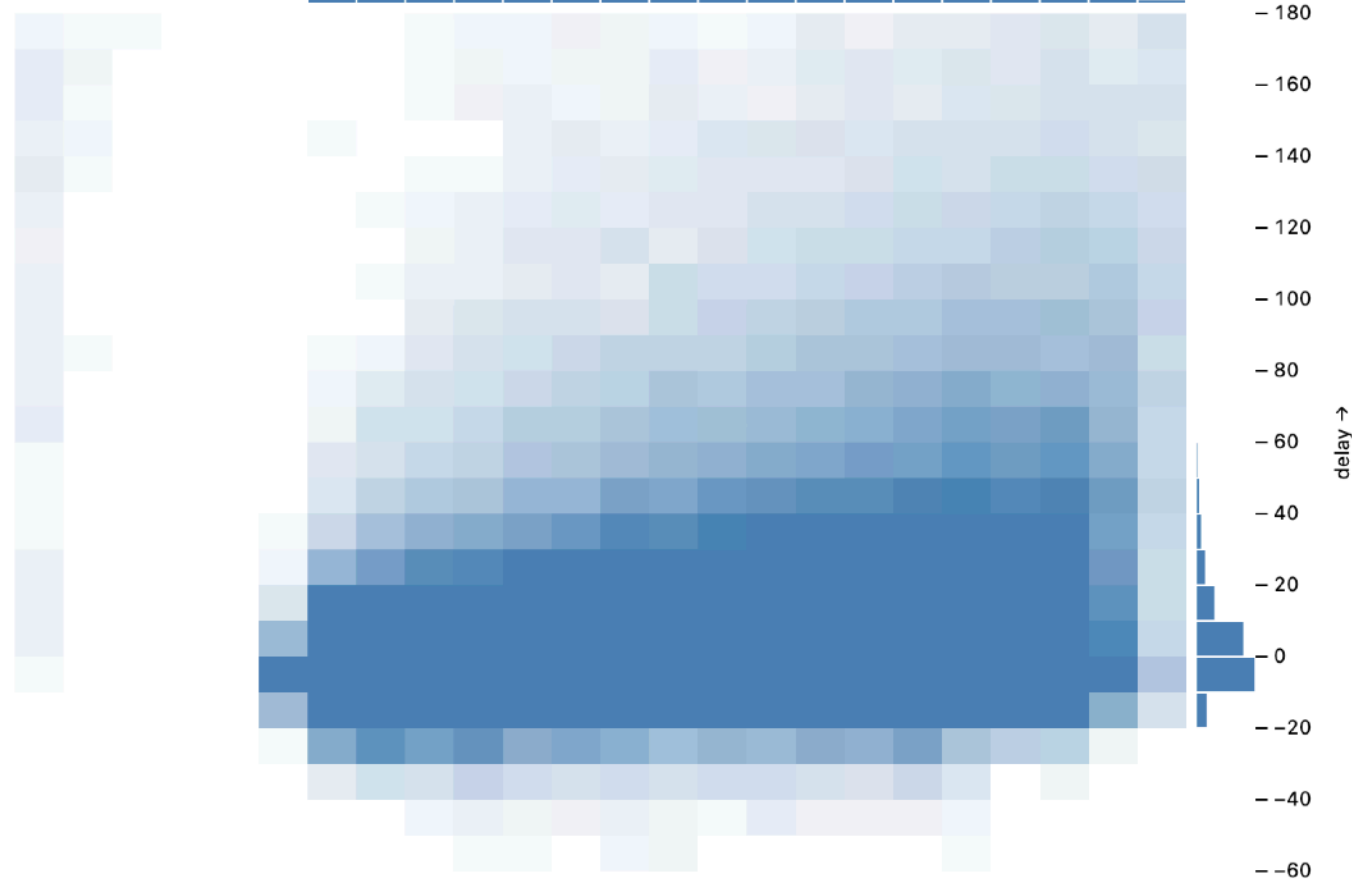
10
|

15
|

20
|

Flight Delays

250k Records



Time Resolution

pixels ▾

Delay Resolution

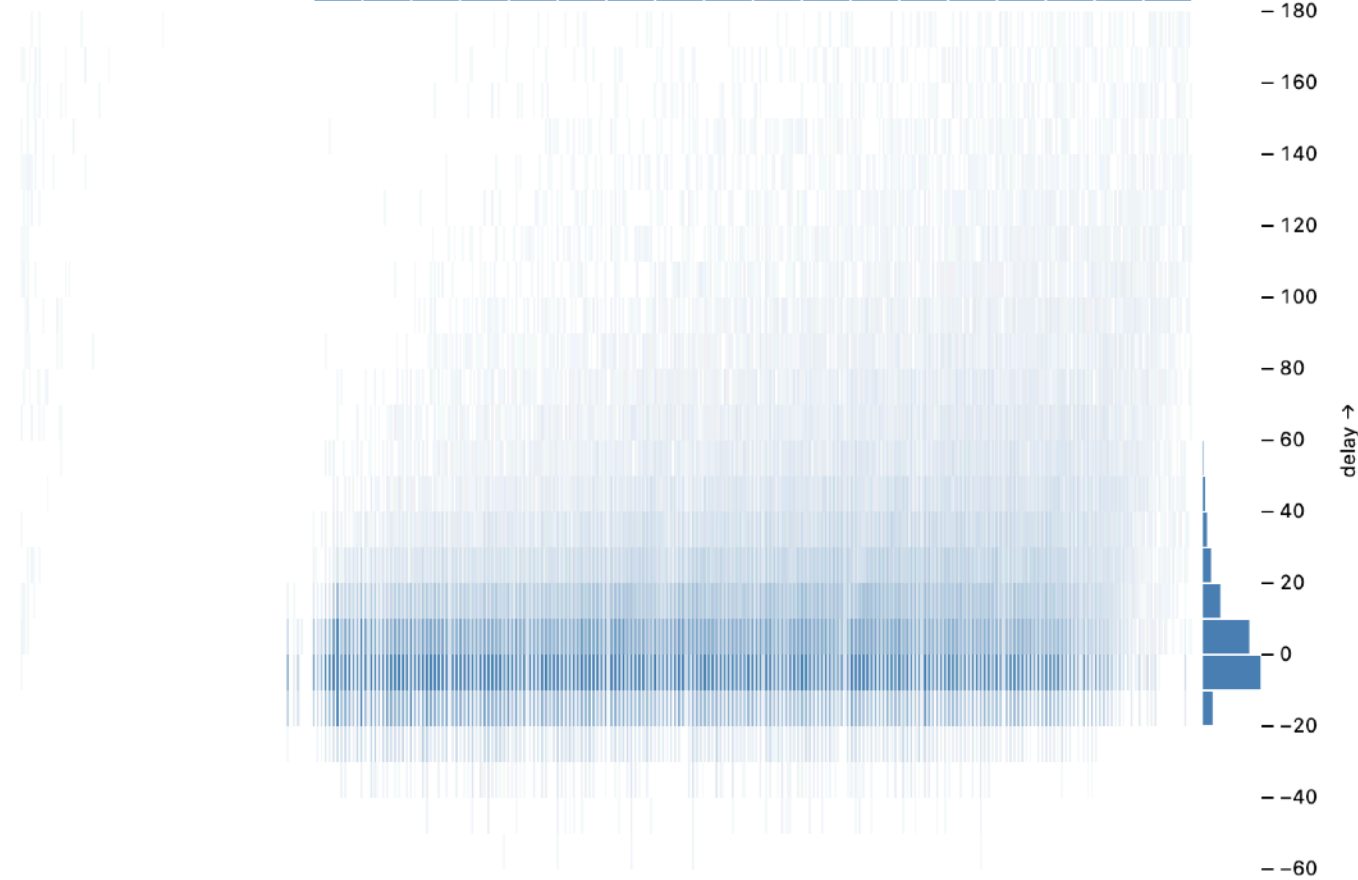
bins ▾

time →

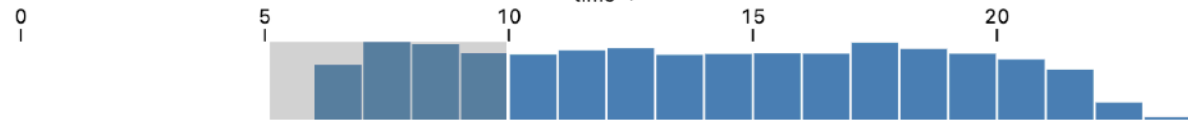


Flight Delays

250k Records



Time Resolution Delay Resolution



Flight Delays
250k Records

- 180

- 160

- 140

- 120

- 100

- 80

- 60

- 40

- 20

- 0

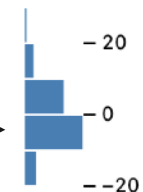
- 20

- 40

- 60

delay →

Sum across rows to re-aggregate



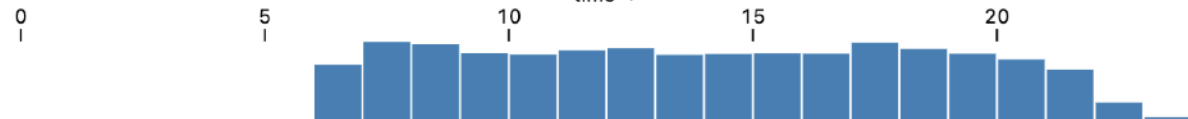
Time Resolution

pixels ▾

Delay Resolution

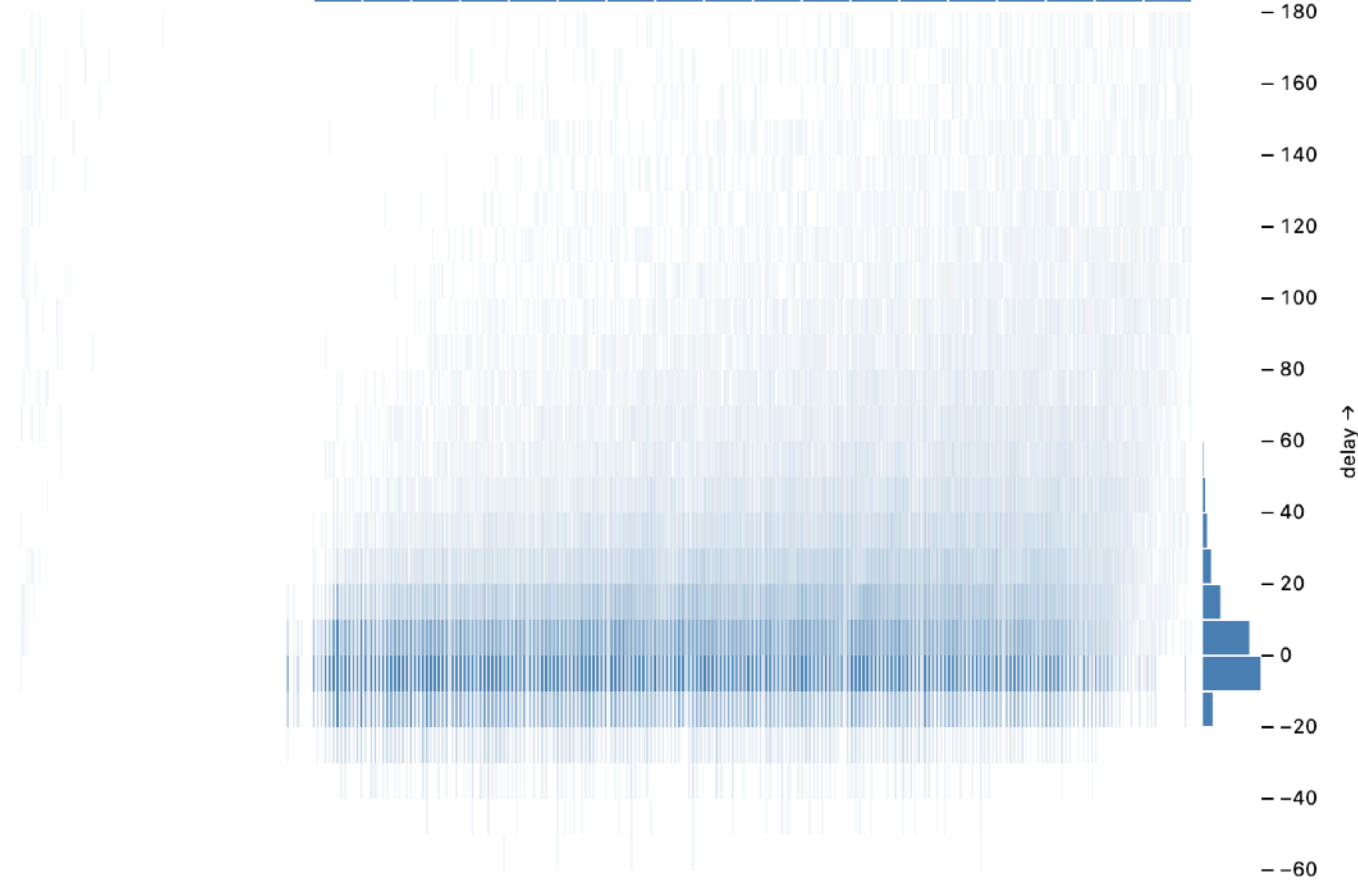
bins ▾

time →



Flight Delays

250k Records



Time Resolution

bins ▾

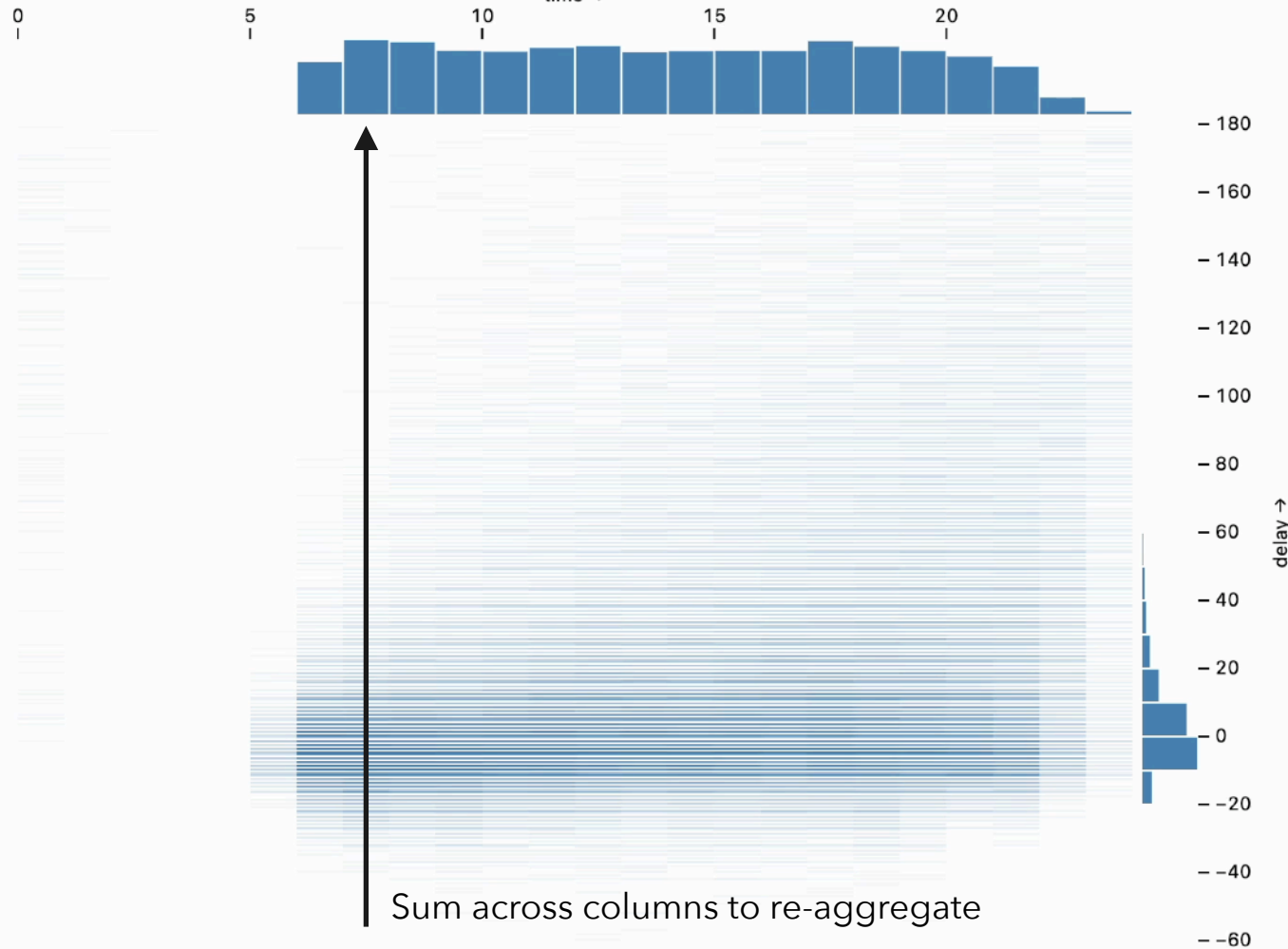
Delay Resolution

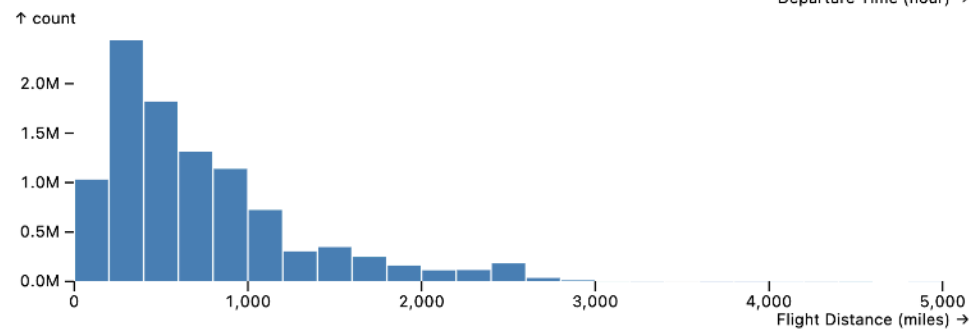
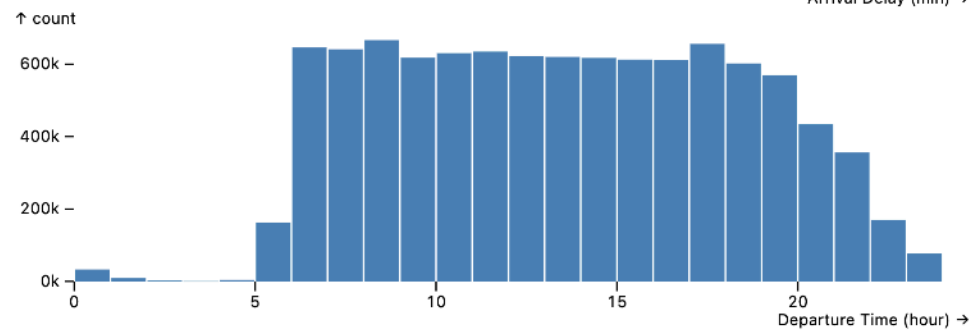
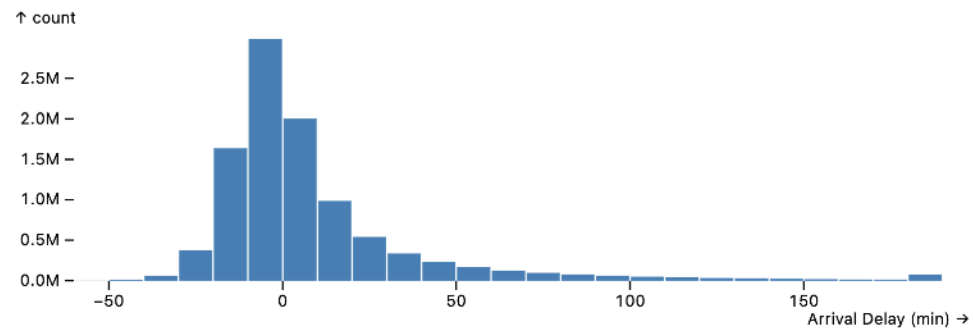
pixels ▾

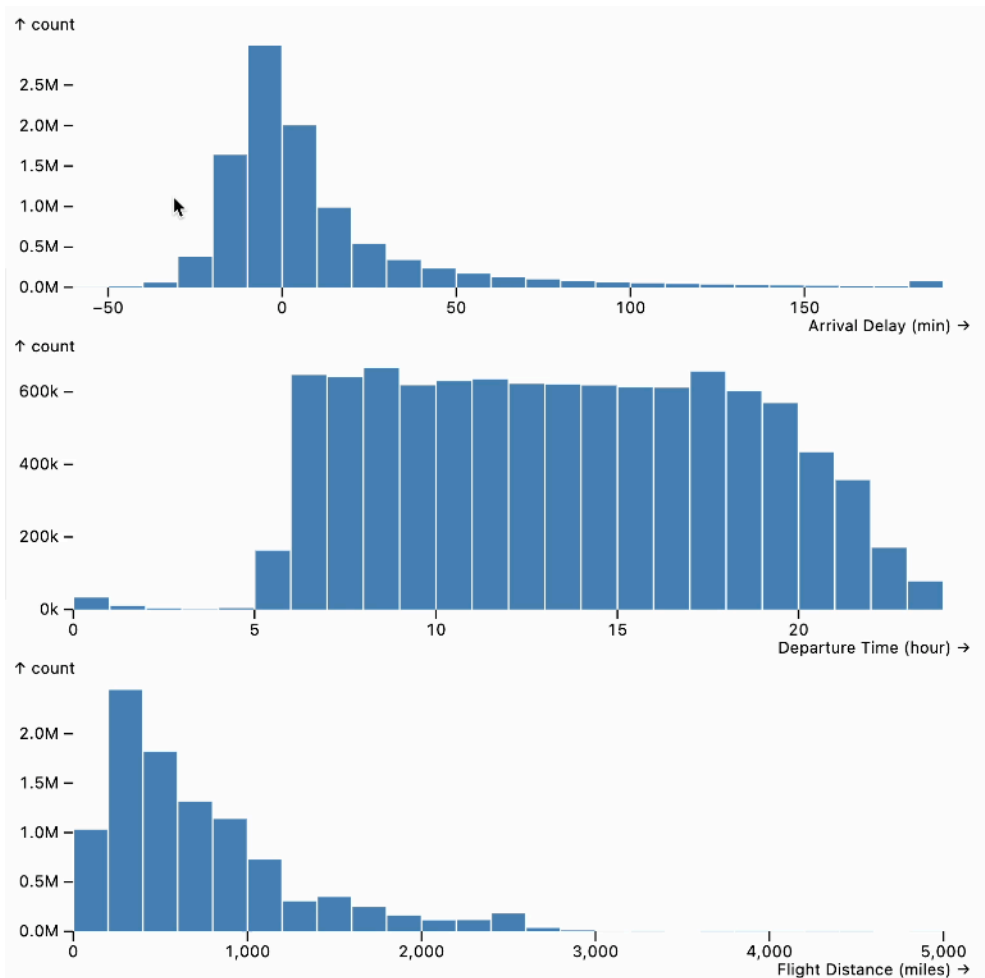
time →

Flight Delays

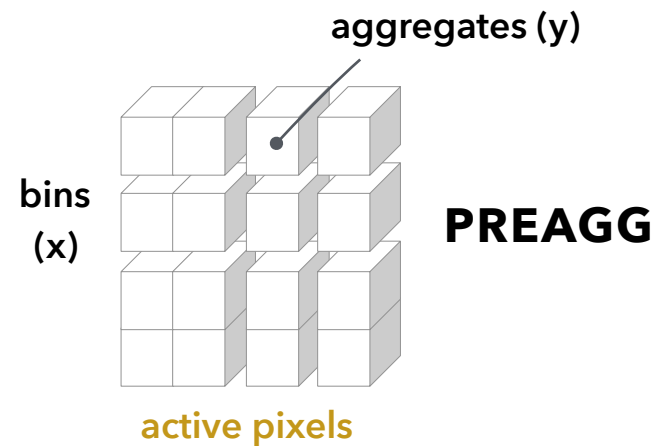
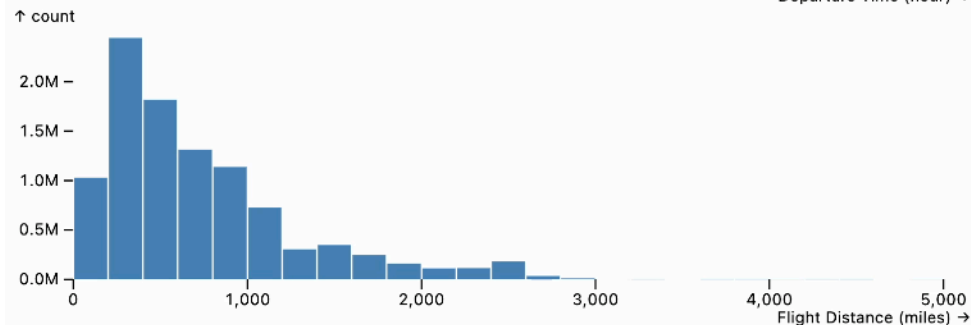
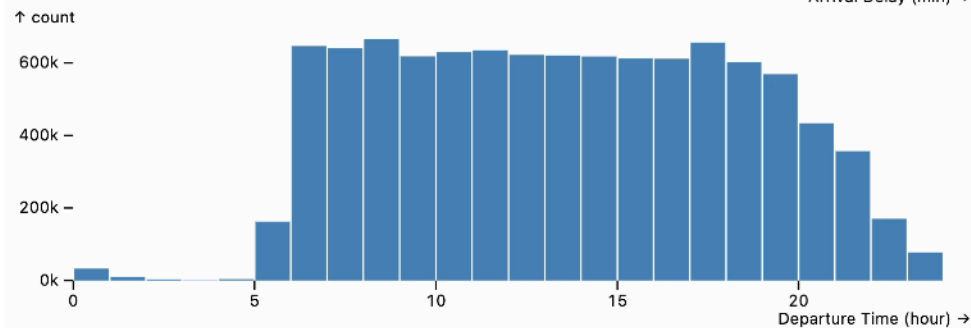
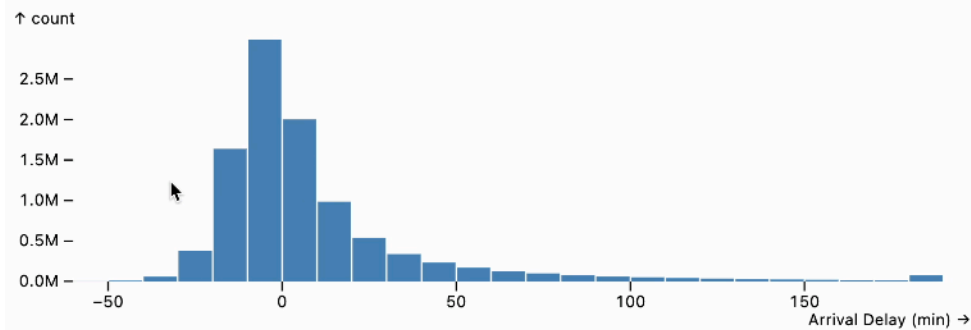
250k Records



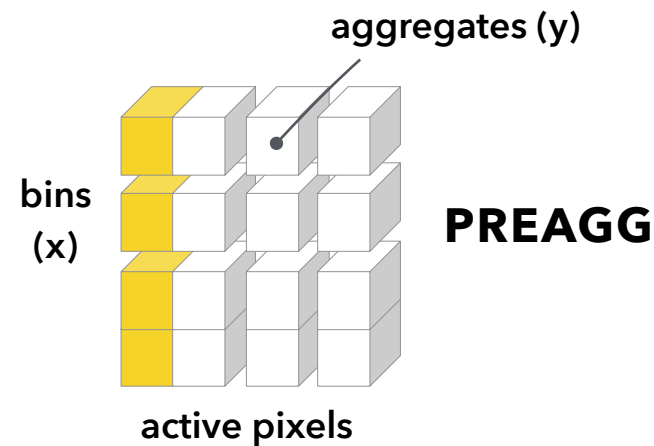
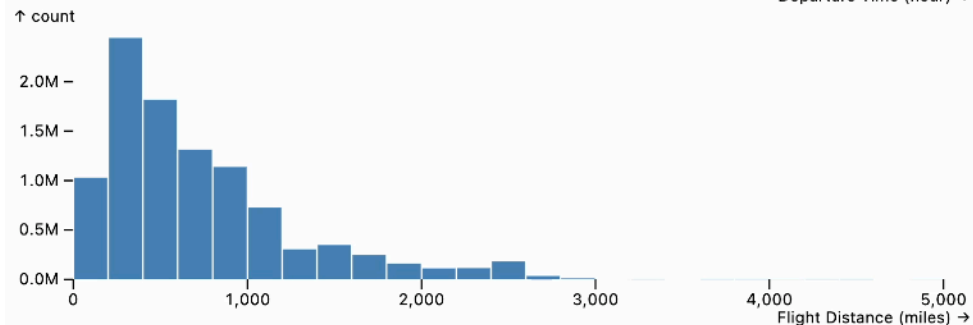
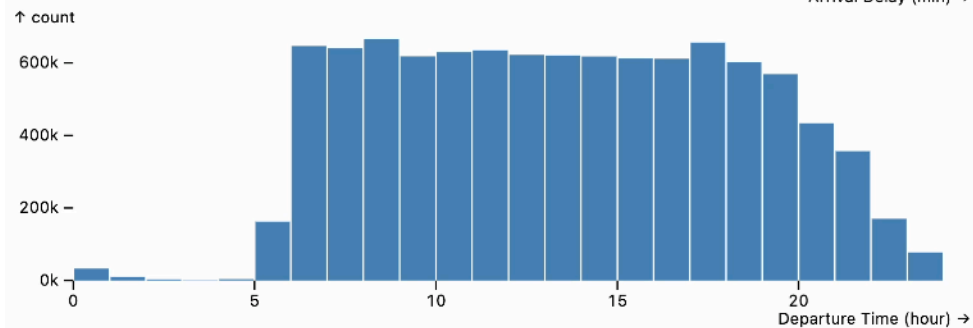
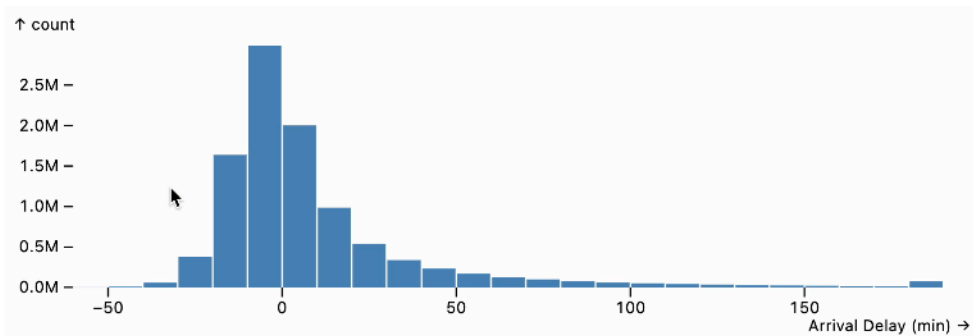




```
SELECT  
  $d0 + $step * FLOOR((delay-$d0)/$step) AS x,  
  COUNT(*) AS y  
FROM flights  
WHERE x BETWEEN $brush0 AND $brush1  
GROUP BY x
```

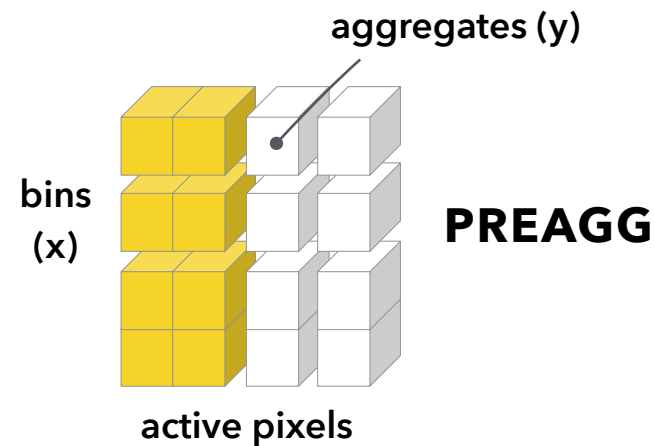
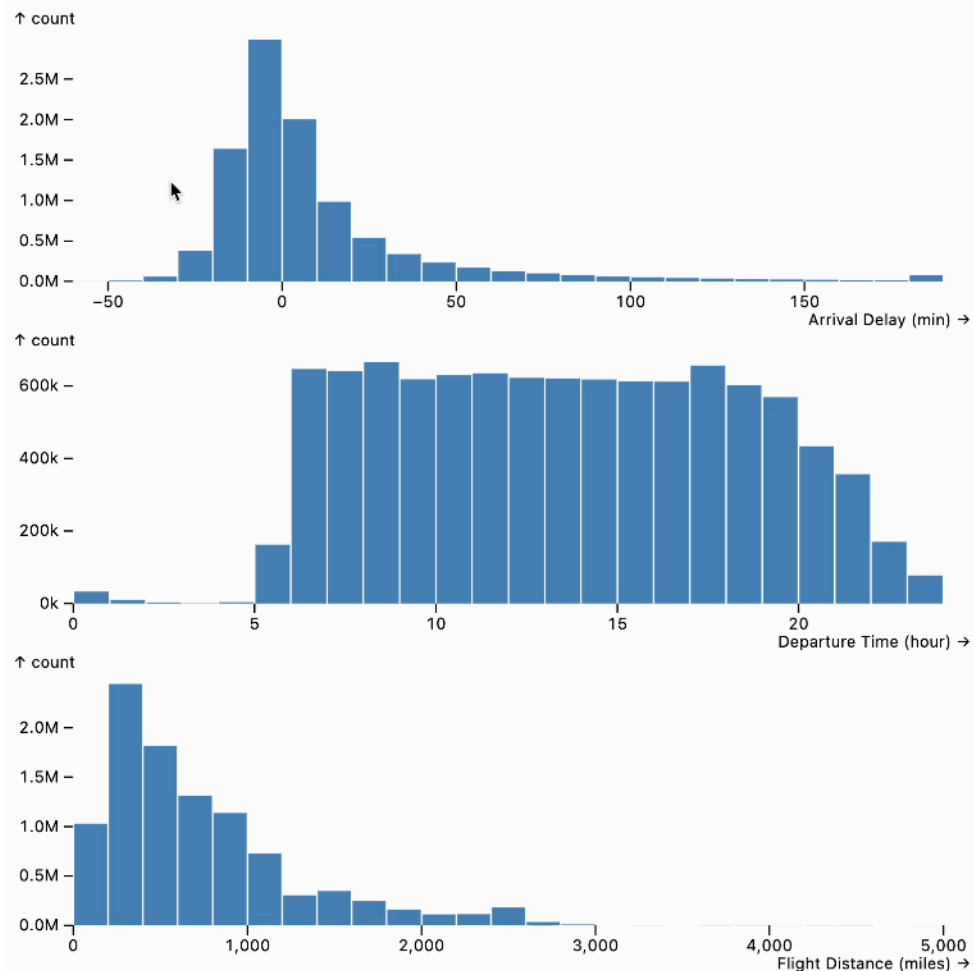



```
CREATE TABLE preagg_a097caa4 AS
SELECT
  $d0 + $step * FLOOR((delay-$d0)/$step) AS x,
  COUNT(*) AS y,
  FLOOR($pixels * (time-$t0)/($t1-$t0)) AS active,
FROM flights
GROUP BY x, active
```

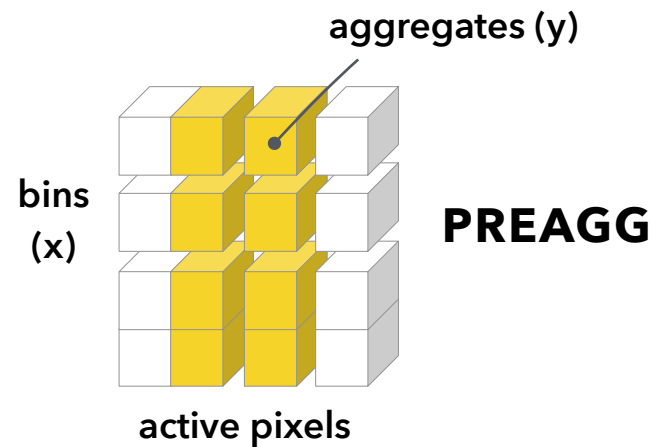
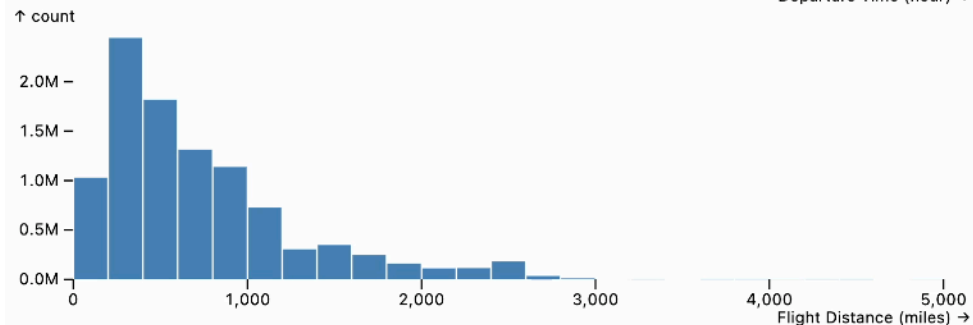
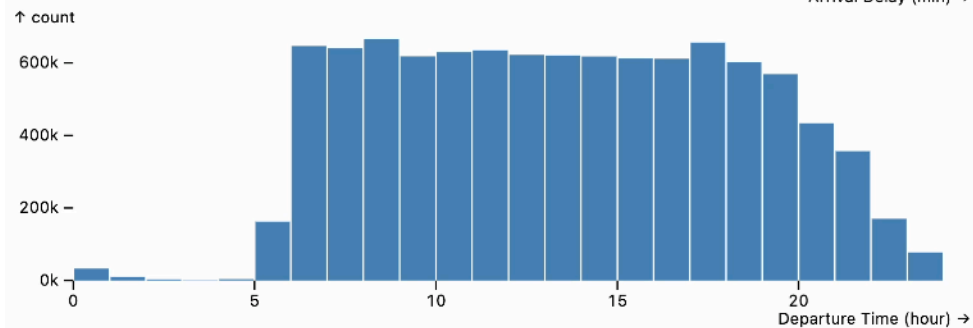
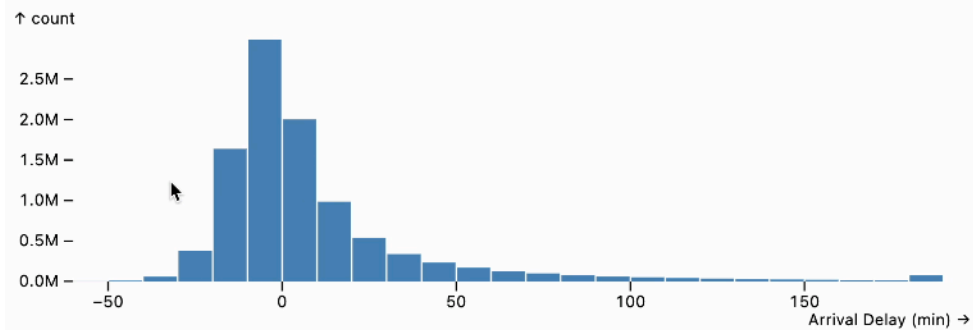
```
CREATE TABLE preagg_a097caa4 AS
SELECT
  $d0 + $step * FLOOR((delay-$d0)/$step) AS x,
  COUNT(*) AS y,
  FLOOR($pixels * (time-$t0)/($t1-$t0)) AS active,
FROM flights
GROUP BY x, active
```

```
SELECT x, SUM(y)
FROM preagg_a097caa4
WHERE active BETWEEN $active0 AND $active1
GROUP BY x
```

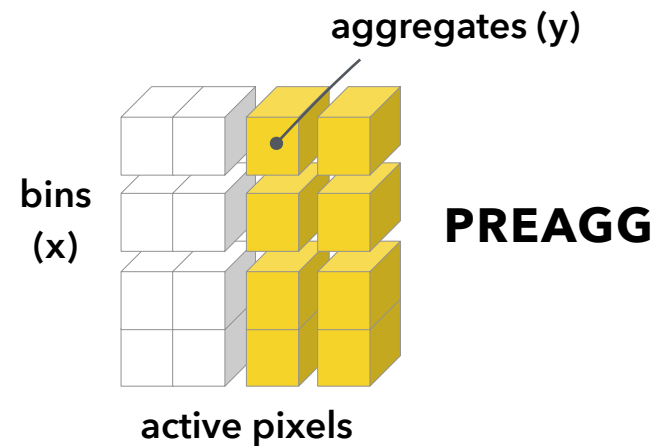
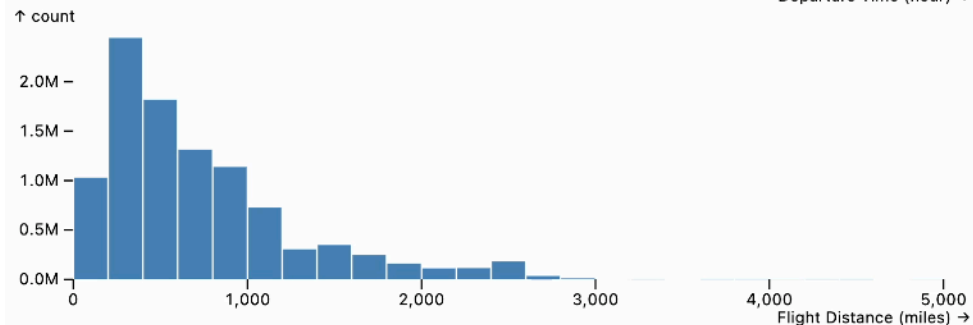
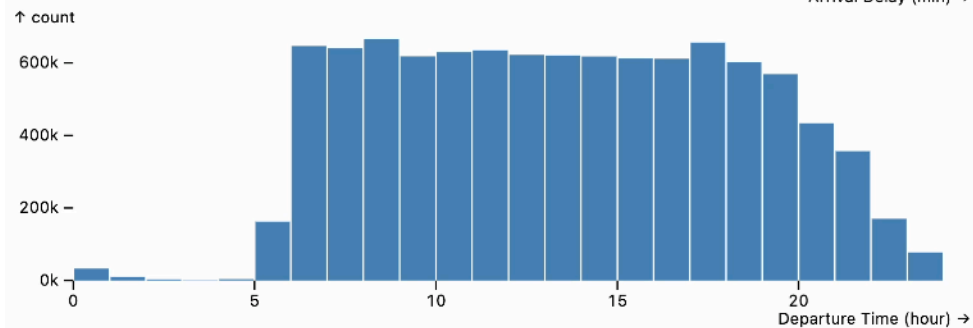
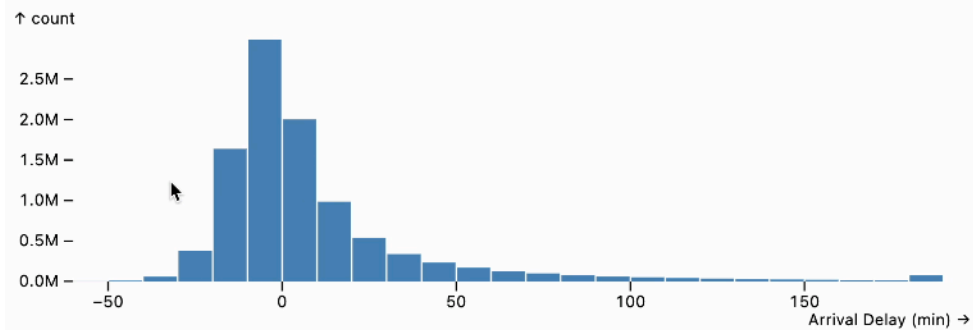
```
CREATE TABLE preagg_a097caa4 AS
SELECT
  $d0 + $step * FLOOR((delay-$d0)/$step) AS x,
  COUNT(*) AS y,
  FLOOR($pixels * (time-$t0)/($t1-$t0)) AS active,
FROM flights
GROUP BY x, active

SELECT x, SUM(y)
FROM preagg_a097caa4
WHERE active BETWEEN $active0 AND $active1
GROUP BY x
```

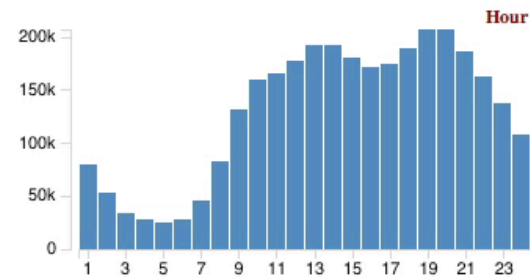
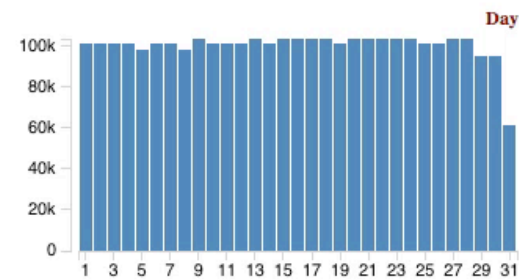
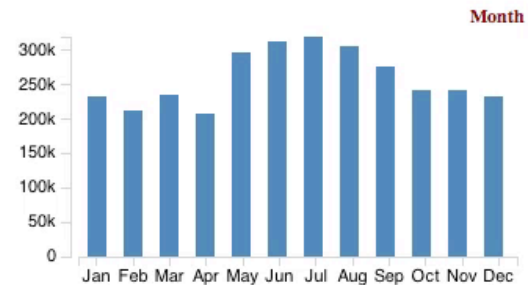
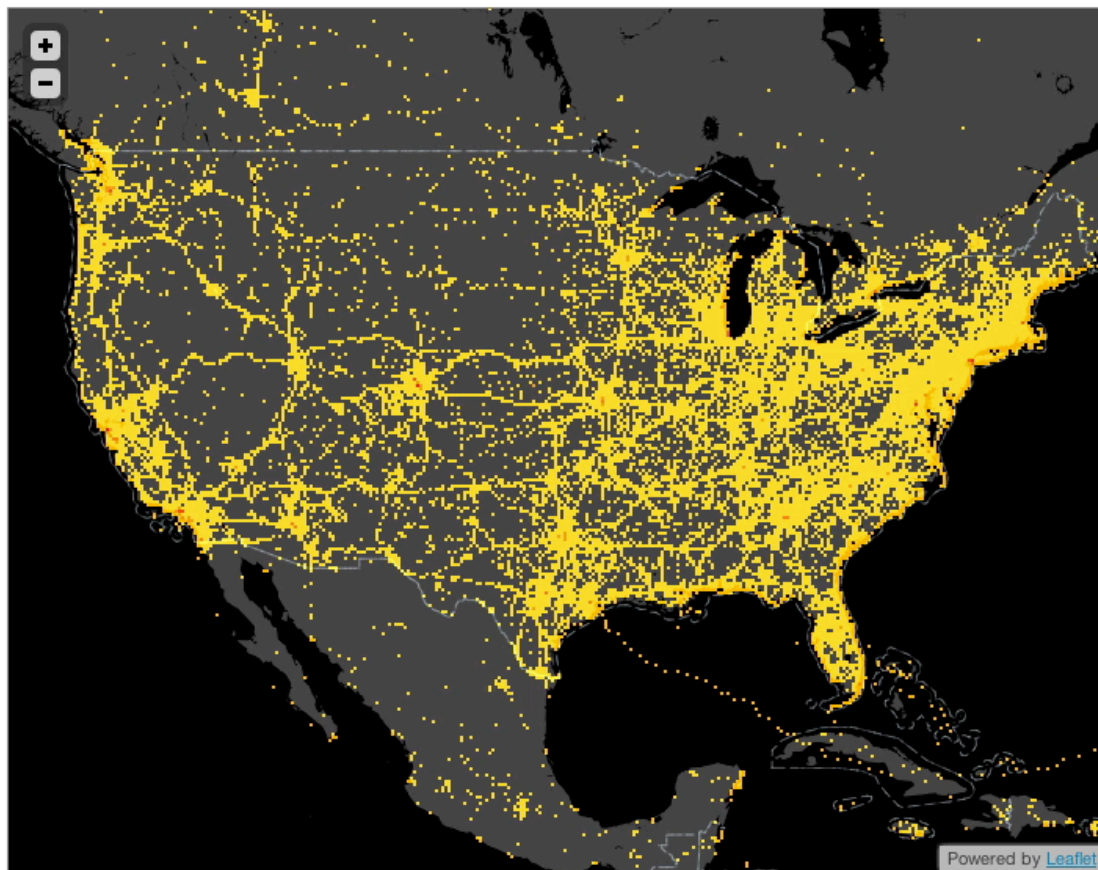
```
CREATE TABLE preagg_a097caa4 AS
SELECT
  $d0 + $step * FLOOR((delay-$d0)/$step) AS x,
  COUNT(*) AS y,
  FLOOR($pixels * (time-$t0)/($t1-$t0)) AS active,
FROM flights
GROUP BY x, active
```

```
SELECT x, SUM(y)
FROM preagg_a097caa4
WHERE active BETWEEN $active0 AND $active1
GROUP BY x
```

```
CREATE TABLE preagg_a097caa4 AS
SELECT
  $d0 + $step * FLOOR((delay-$d0)/$step) AS x,
  COUNT(*) AS y,
  FLOOR($pixels * (time-$t0)/($t1-$t0)) AS active,
FROM flights
GROUP BY x, active
```

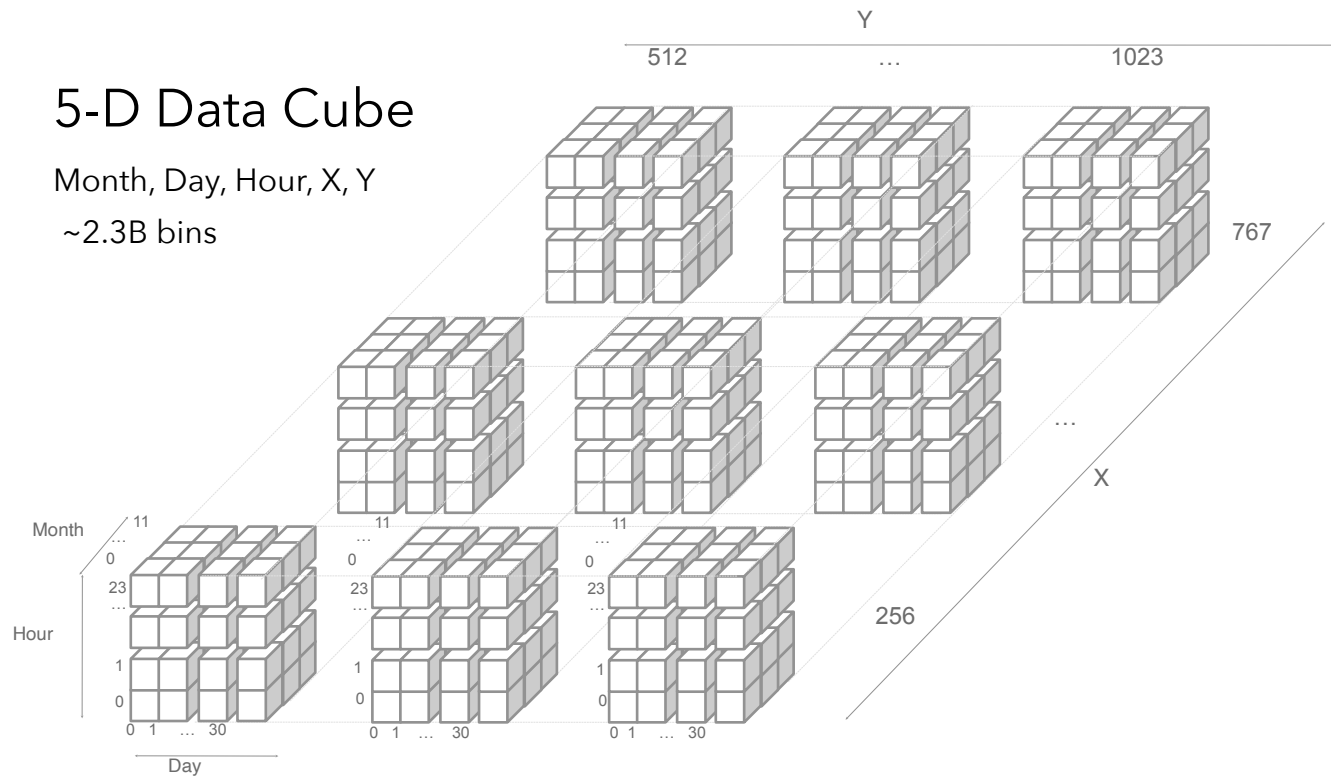
```
SELECT x, SUM(y)
FROM preagg_a097caa4
WHERE active BETWEEN $active0 AND $active1
GROUP BY x
```

5-D Data Cube

Month, Day, Hour, X, Y

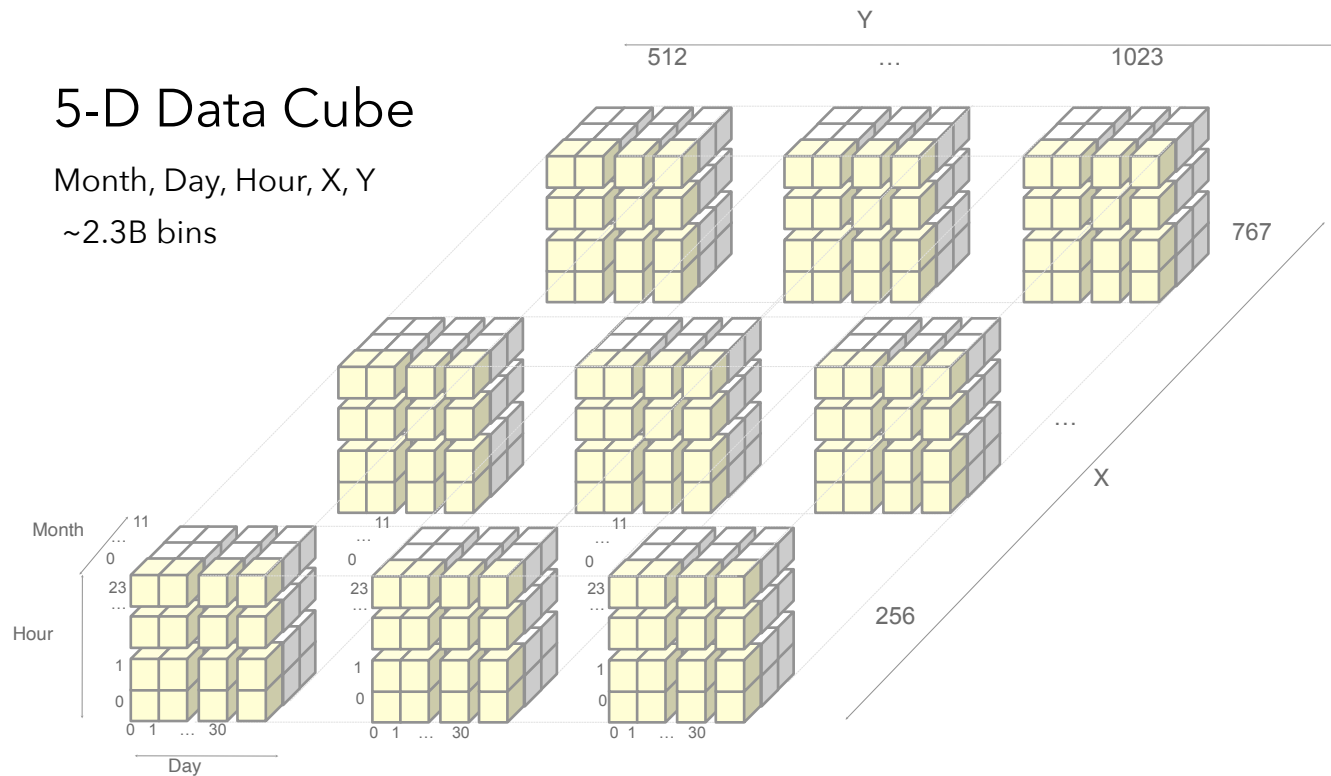
~2.3B bins

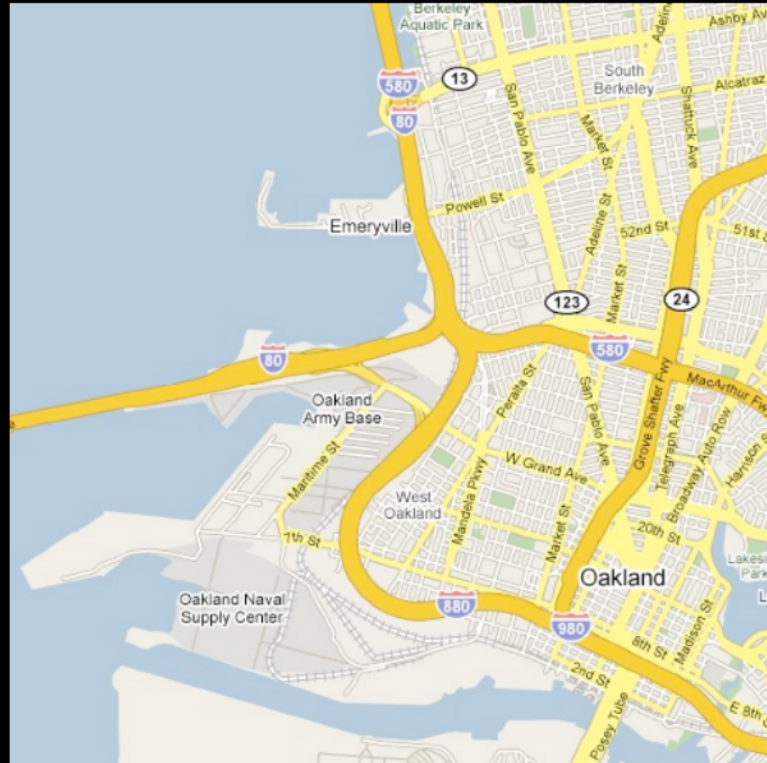


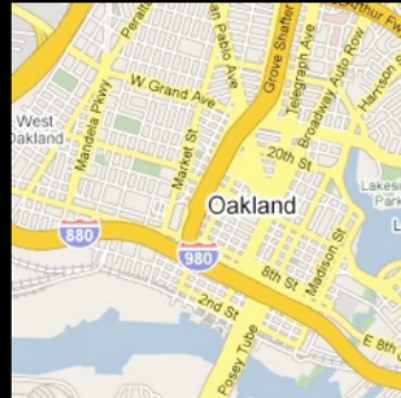
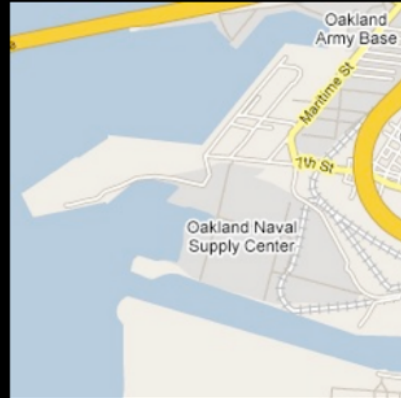
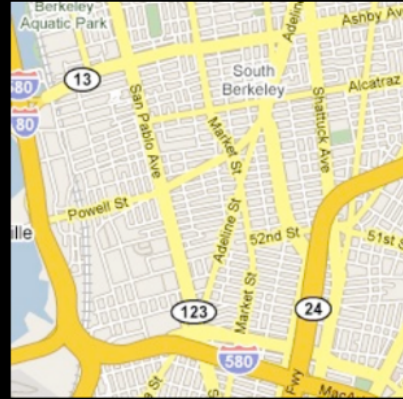
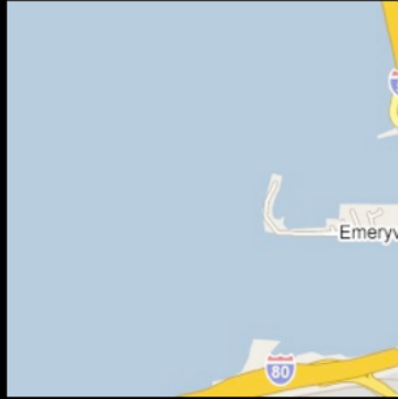
5-D Data Cube

Month, Day, Hour, X, Y

~2.3B bins

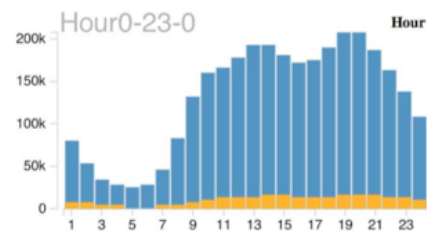
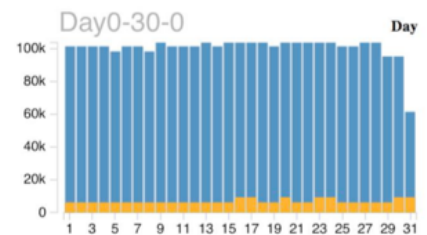
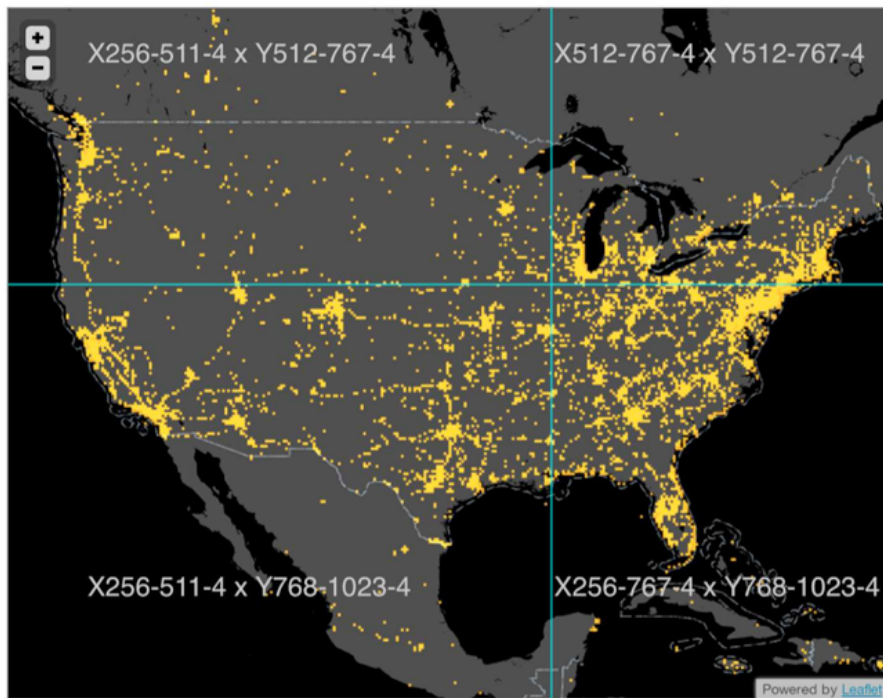


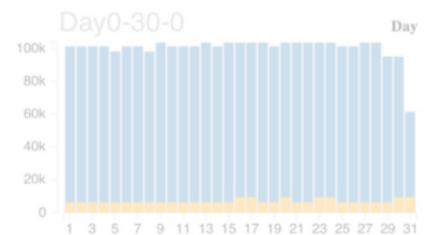
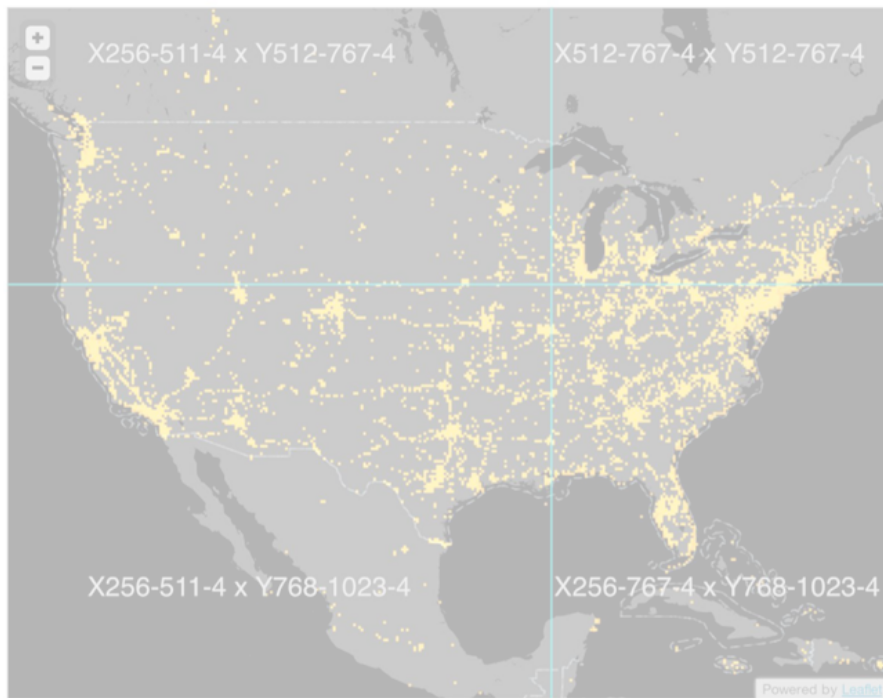


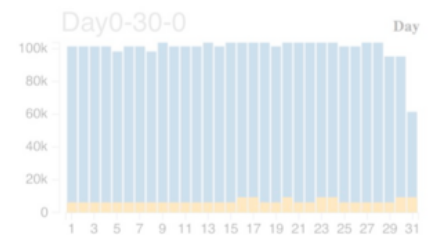
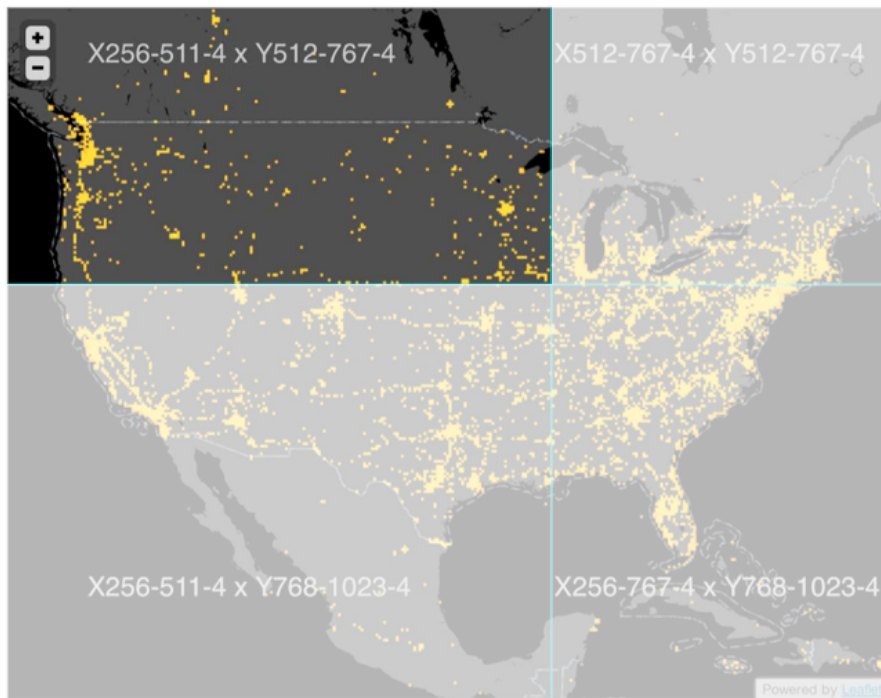


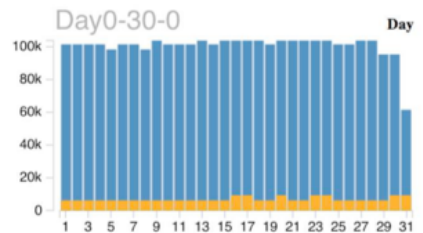
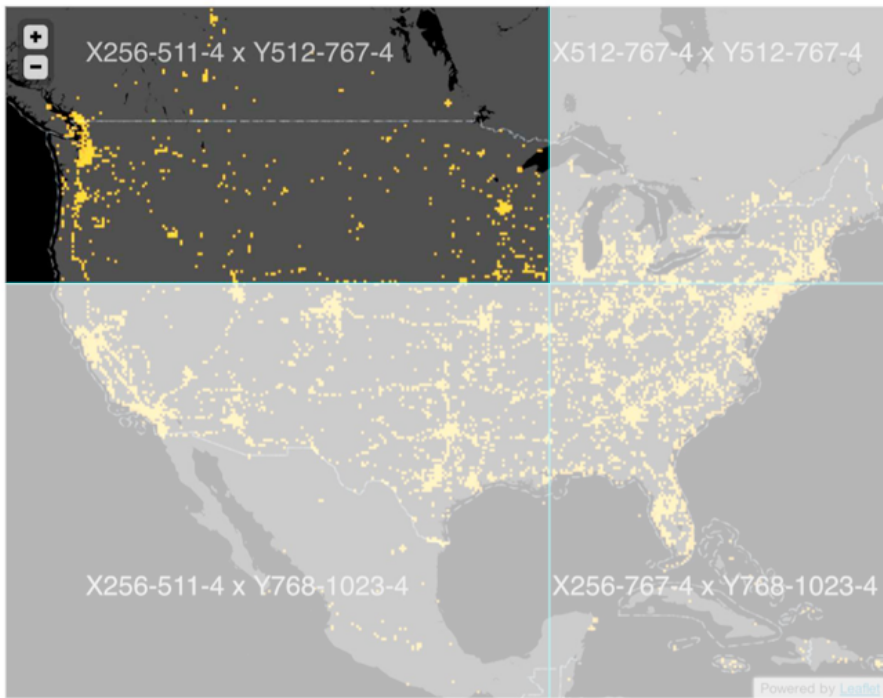
Multivariate Data Tiles

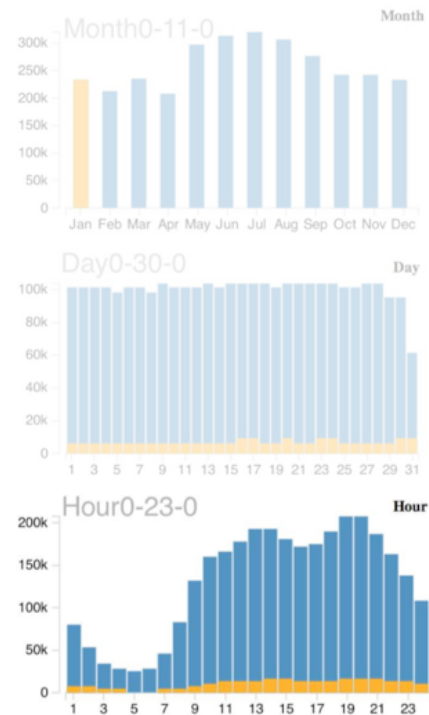
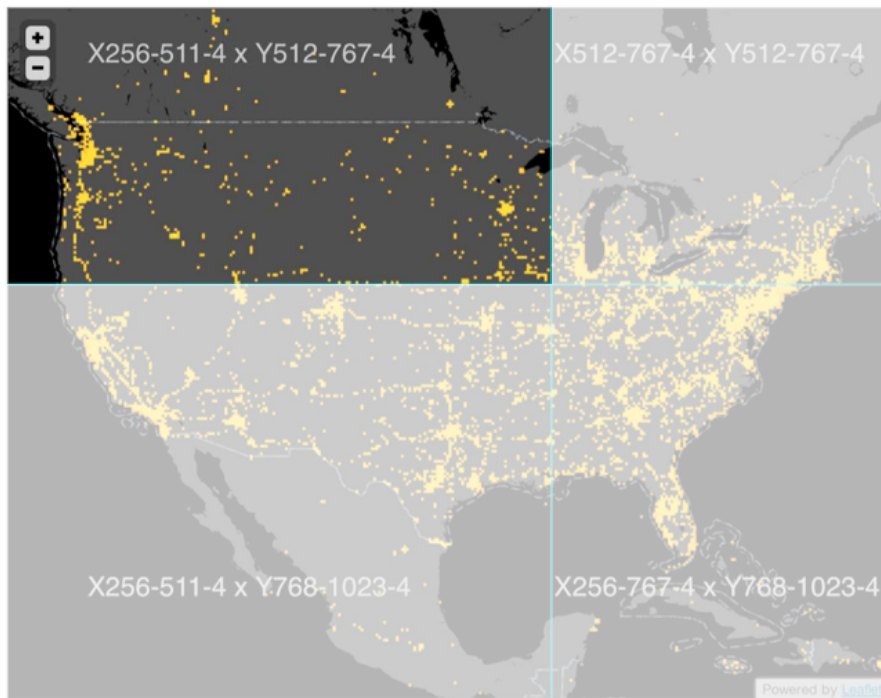
1. Send data, not pixels
2. Embed multi-dim data

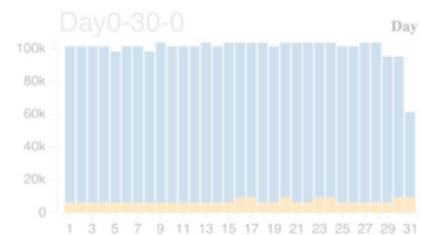
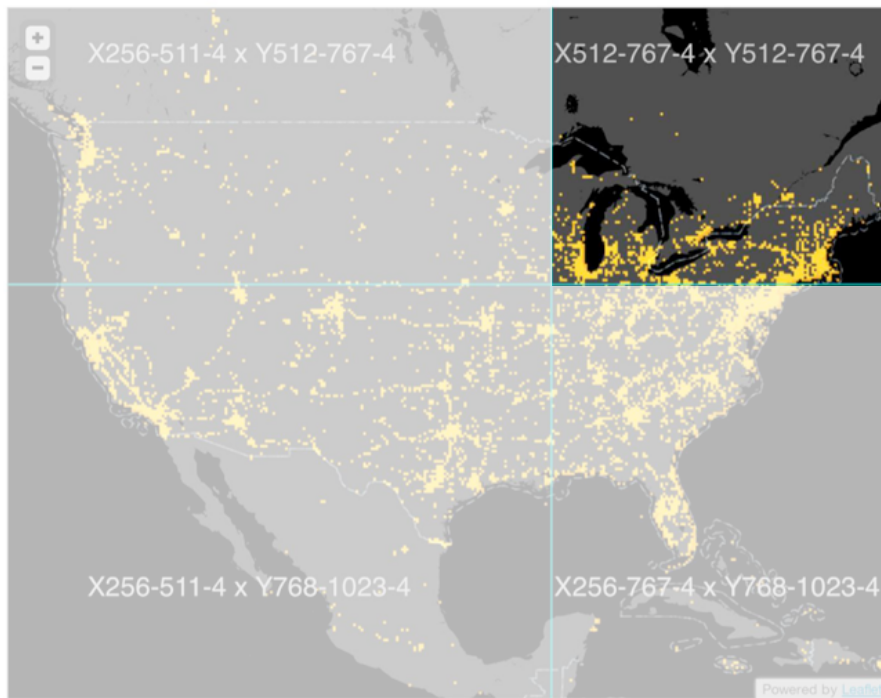


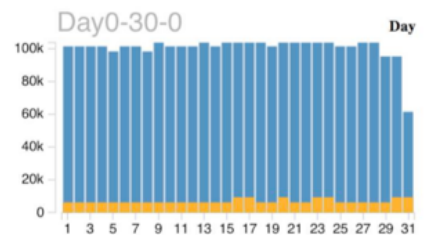
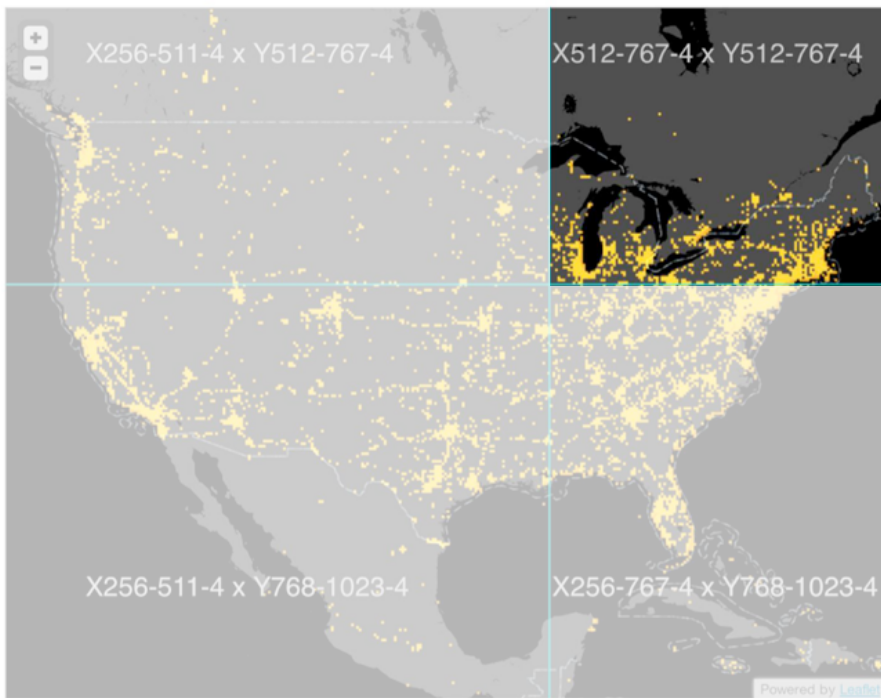


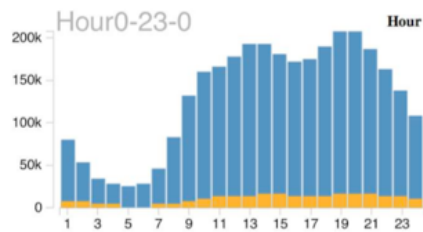
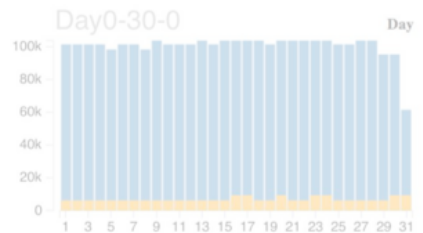
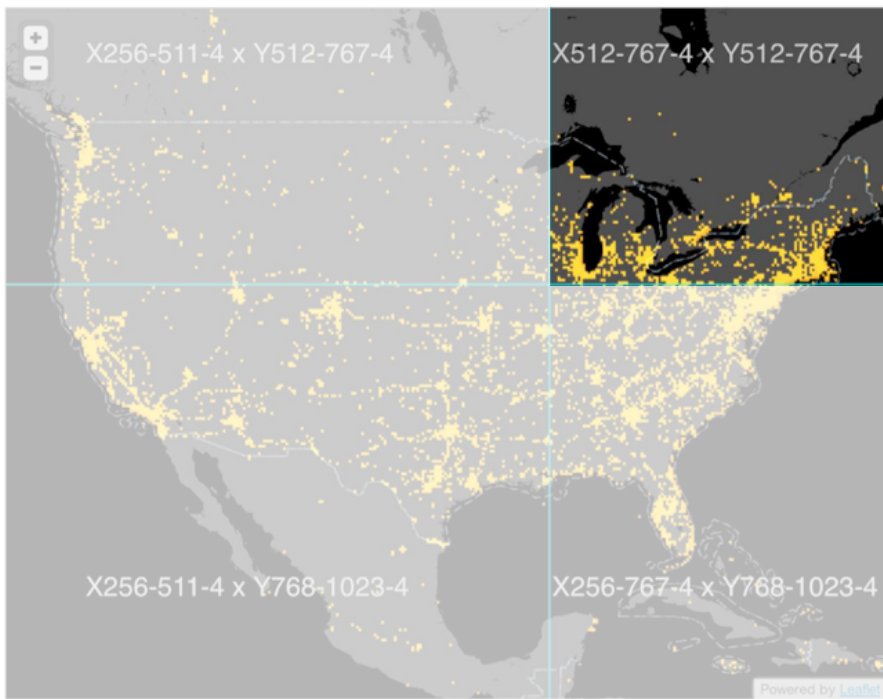


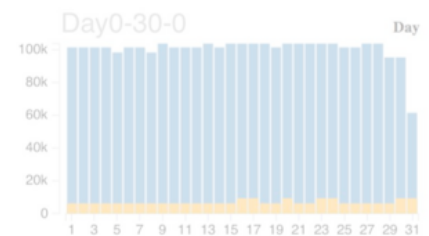
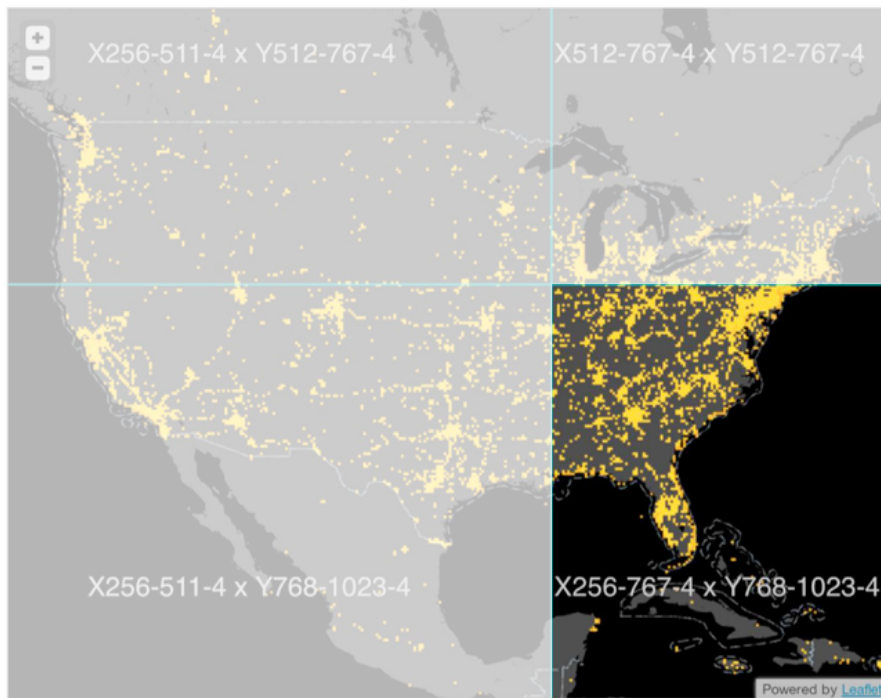


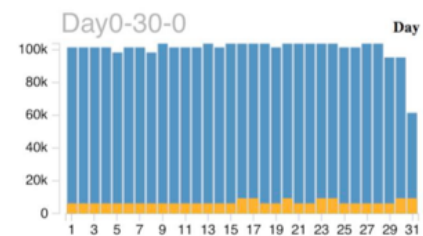
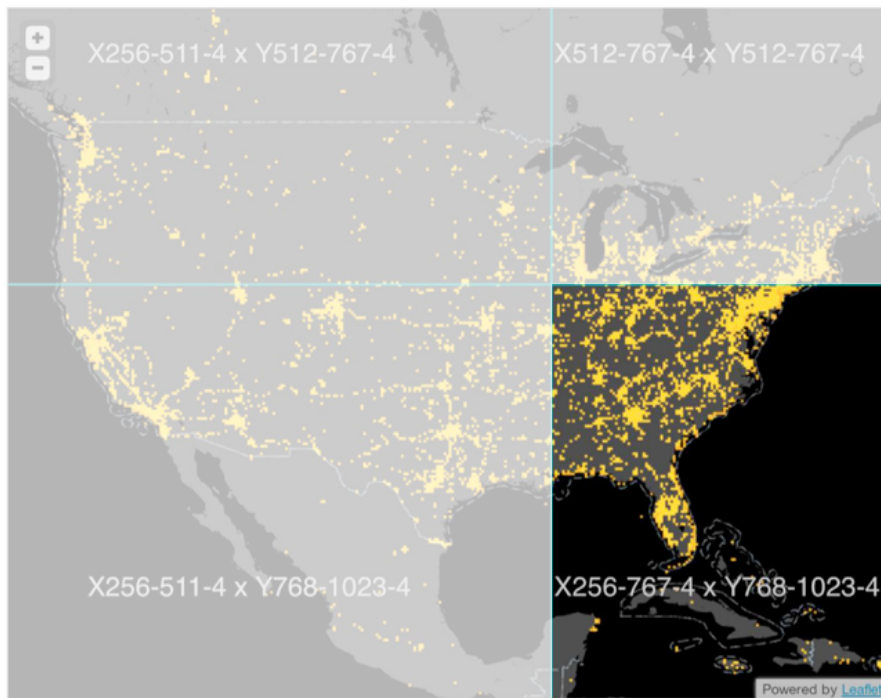


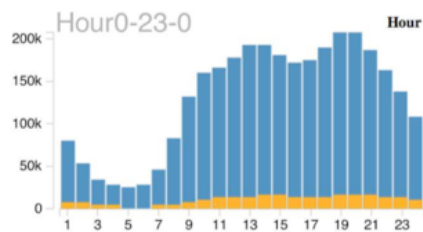
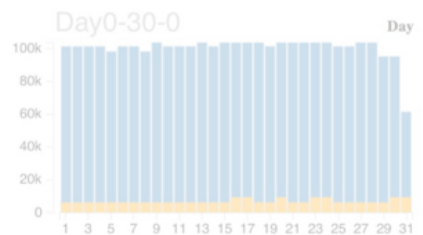
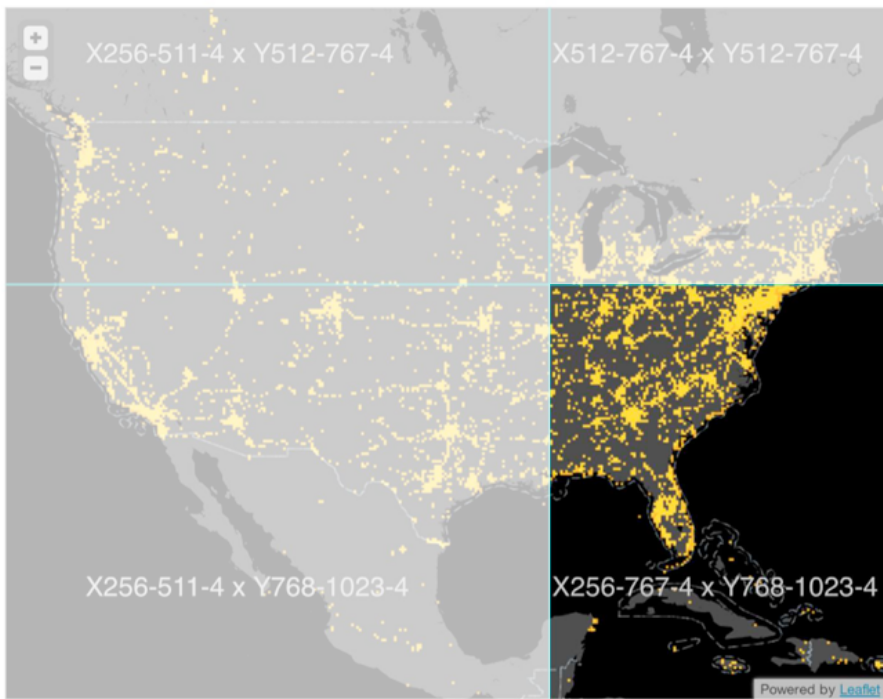


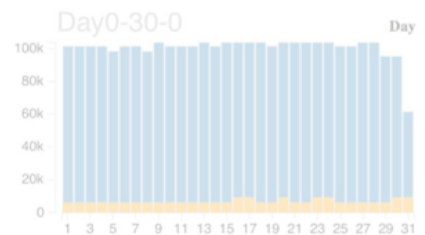
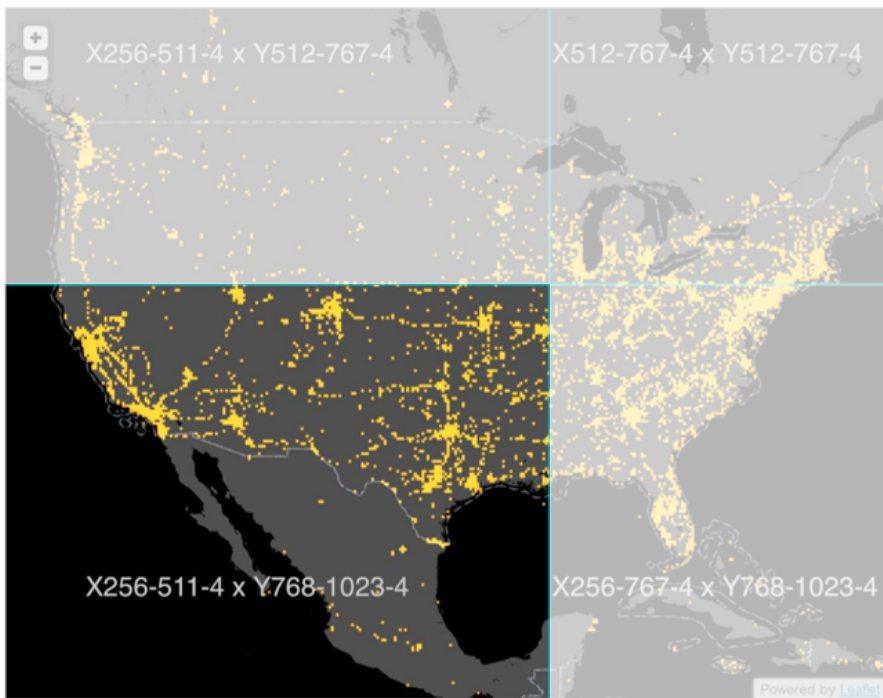


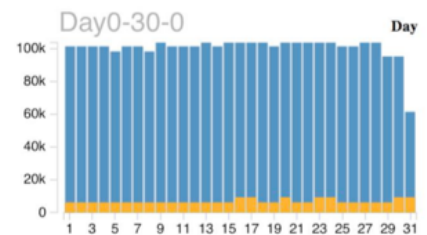
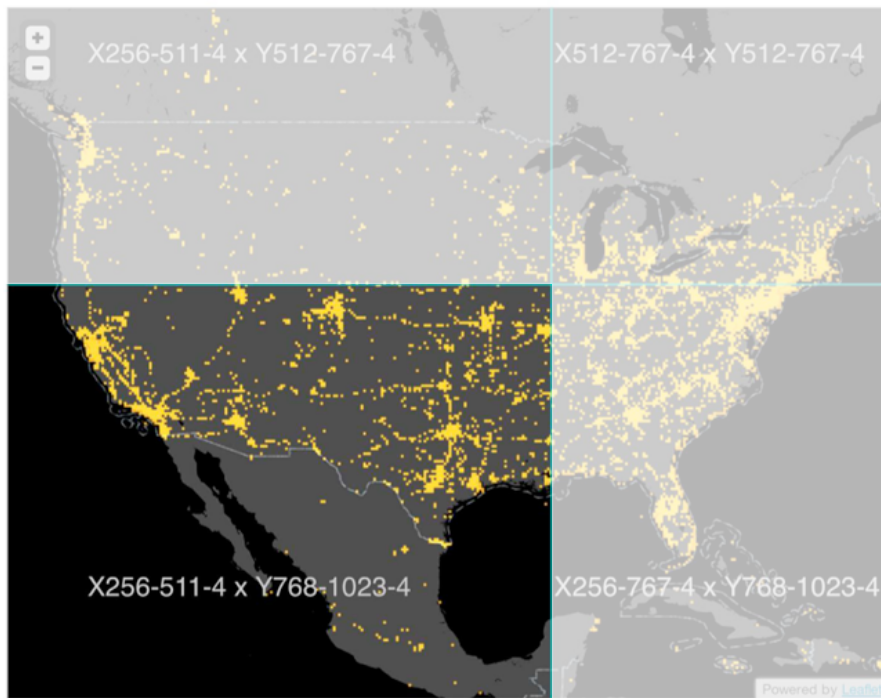


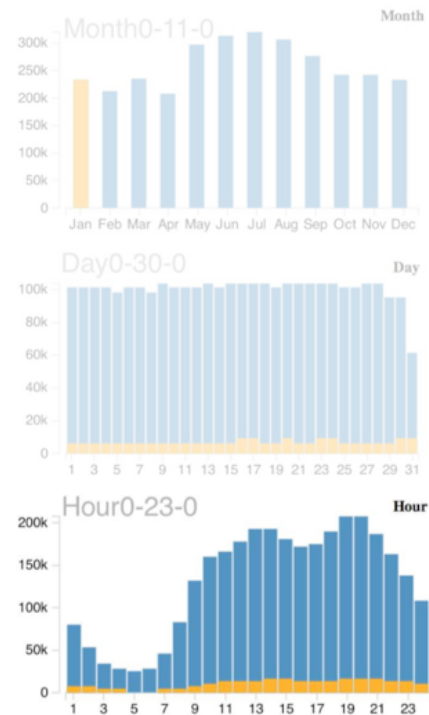
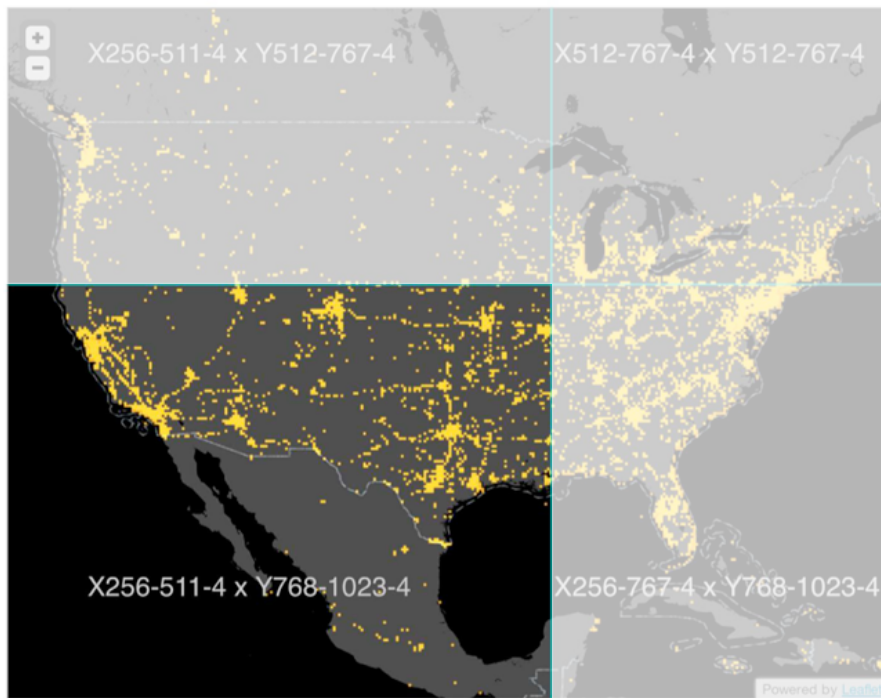


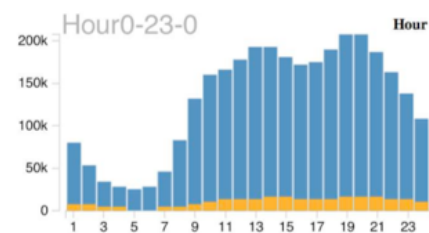
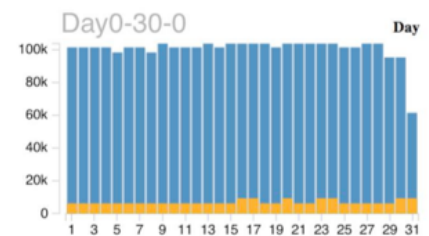
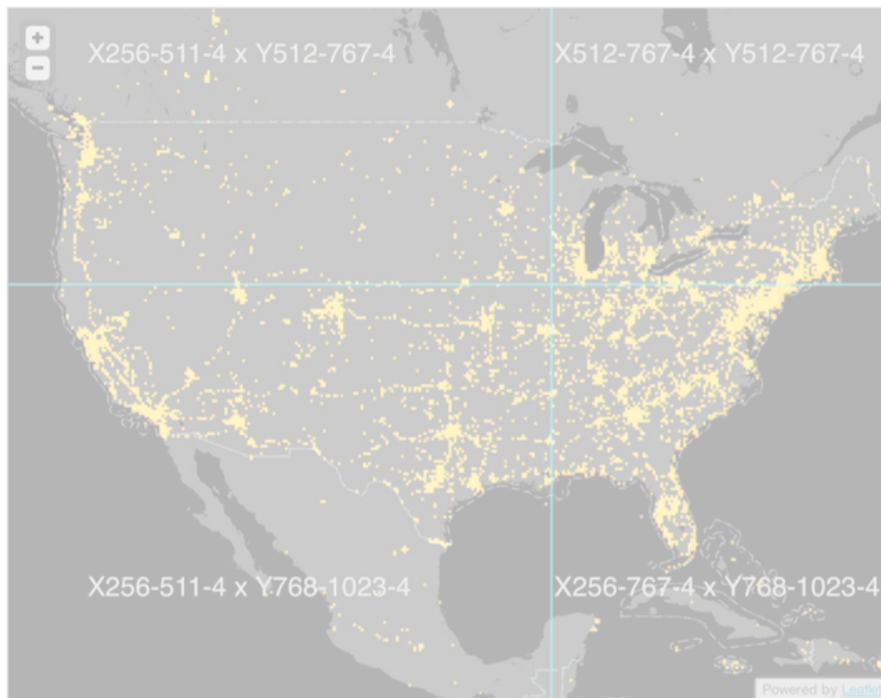


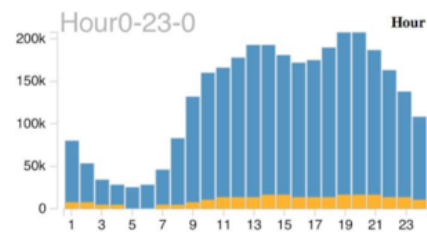
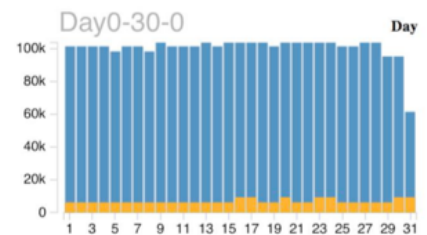
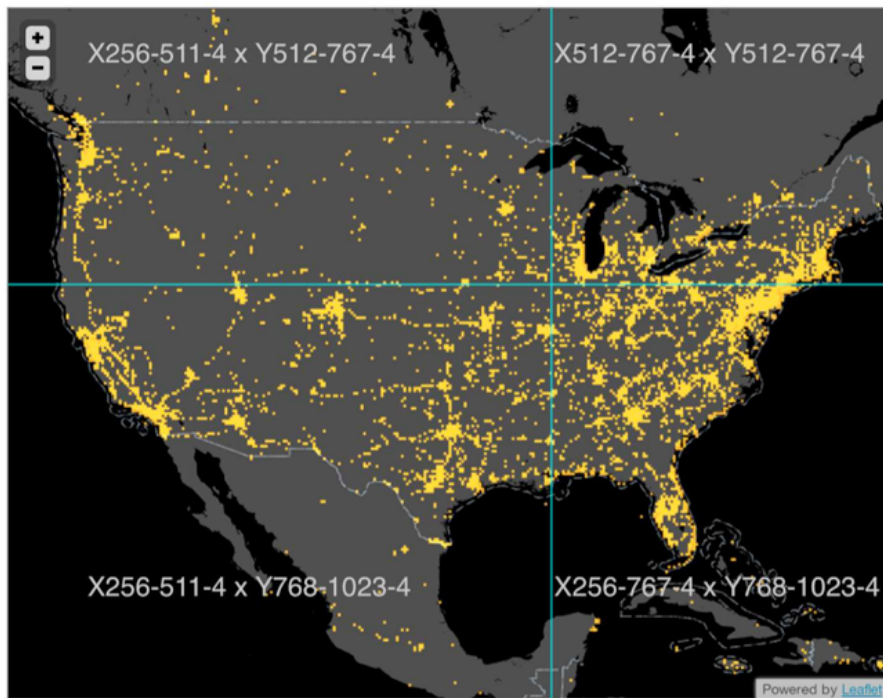




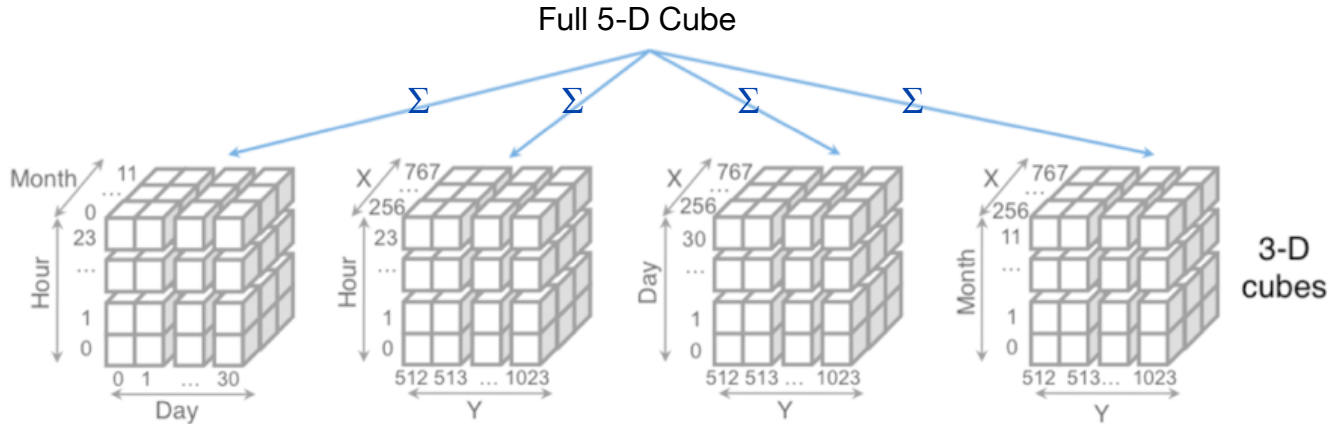






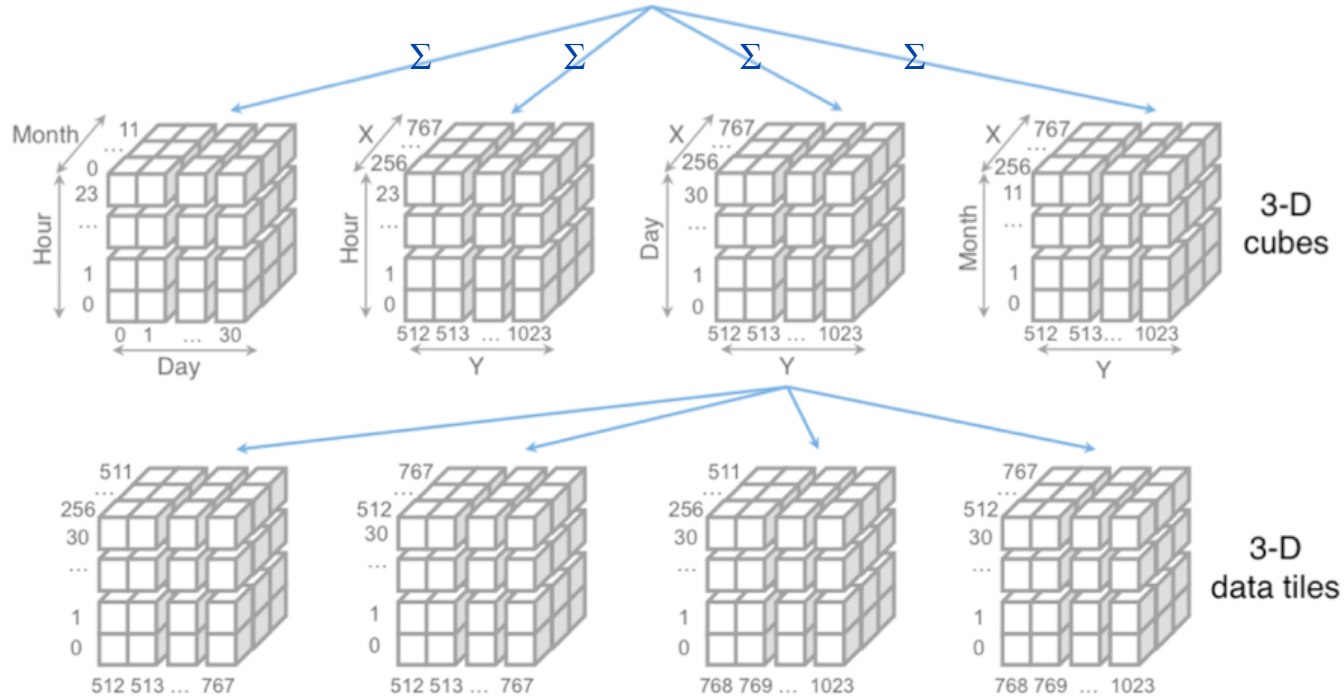


Full 5-D Cube

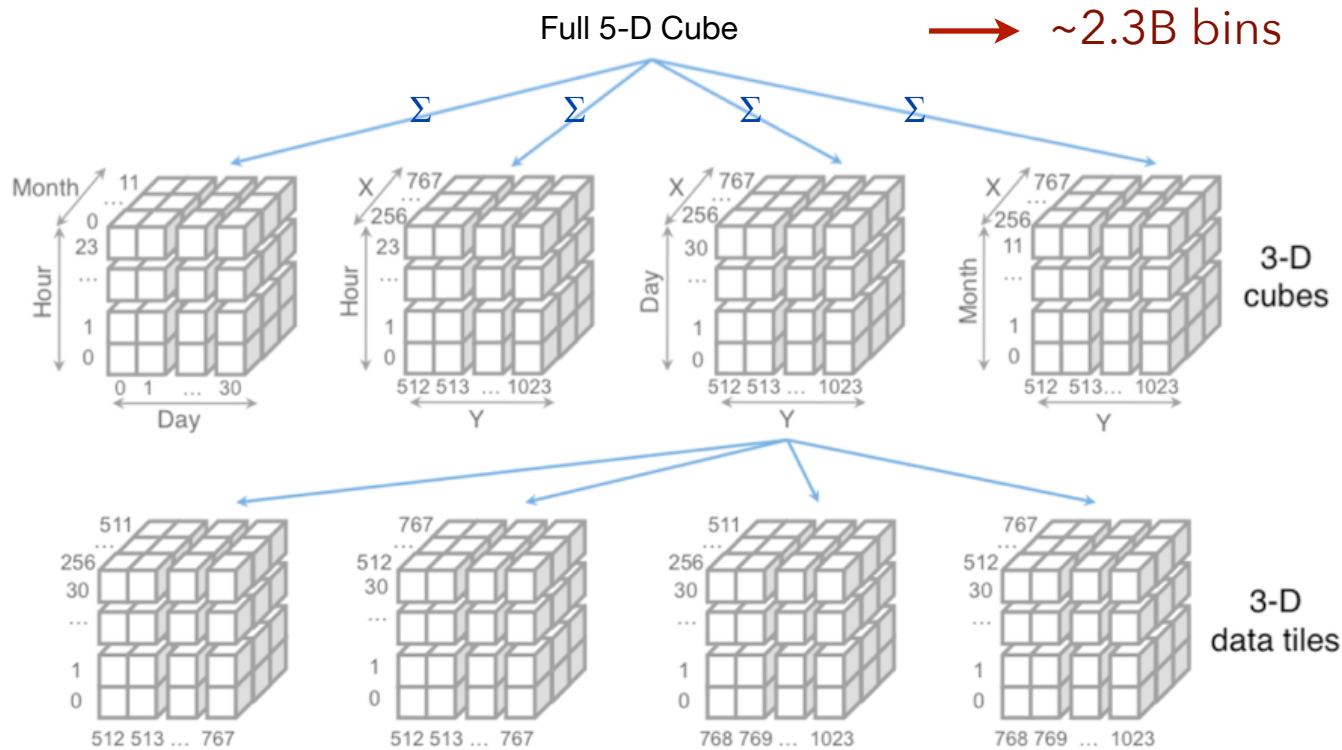


For any pair of 1D or 2D binned plots, the maximum number of dimensions needed to support brushing & linking is **four**.

Full 5-D Cube



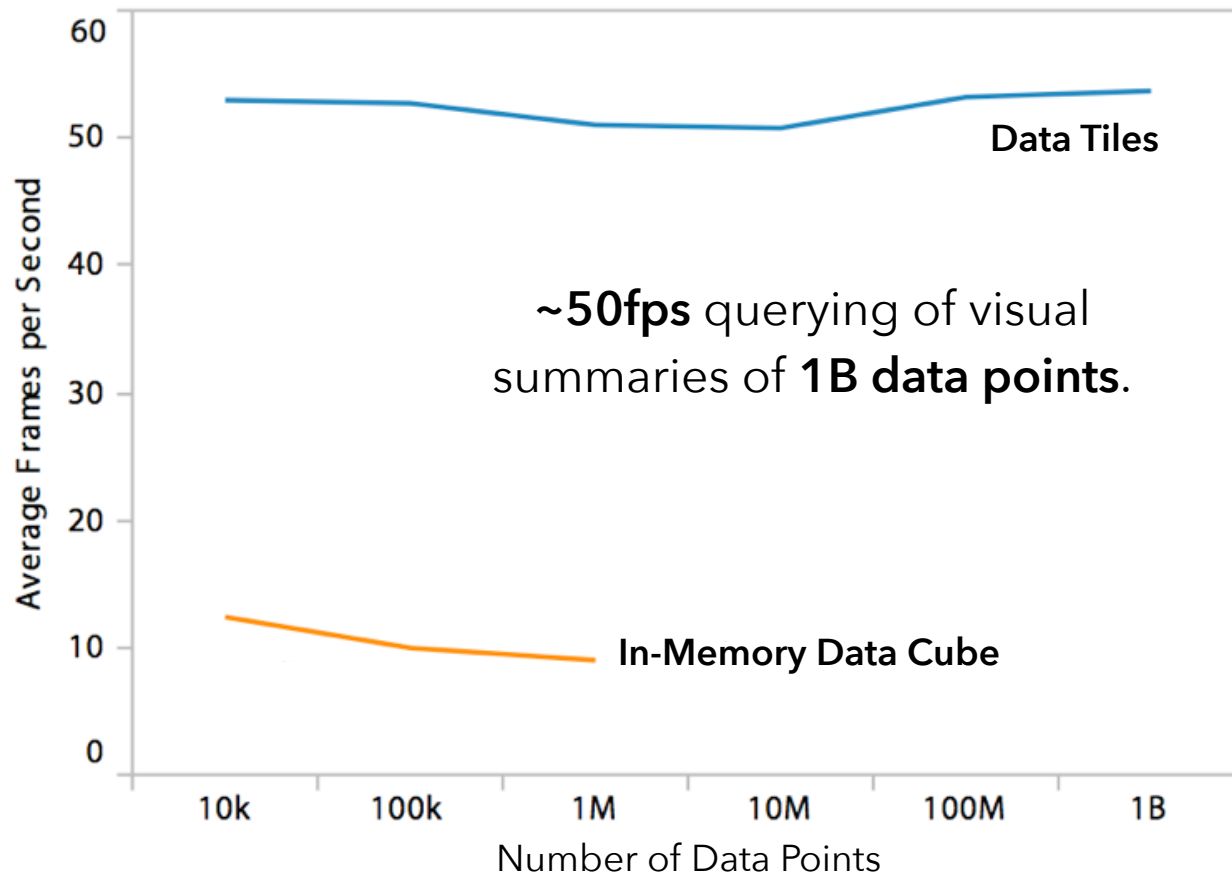
13 3-D Data Tiles



13 3-D Data Tiles

\rightarrow ~17.6M bins
(in 352KB!)

5 dimensions x 50 bins/dim x 25 plots



Limitations and Questions

But where do the preaggregated data tiles come from?

They must be computed, either ahead of time or on-the-fly. Up to the 100M point range, an analytic database can do this on the fly. In the 1B point range, pre-computation avoids delays.

We can also *prefetch*: we can start computing new data tiles as soon as the pointer enters a chart, before a selection is made.

Does super-low-latency interaction really matter?

Is it worth it to go to all of this trouble? (Short answer: yes!)

High latency leads to reduced analytic output [Liu & Heer, InfoVis 2014]

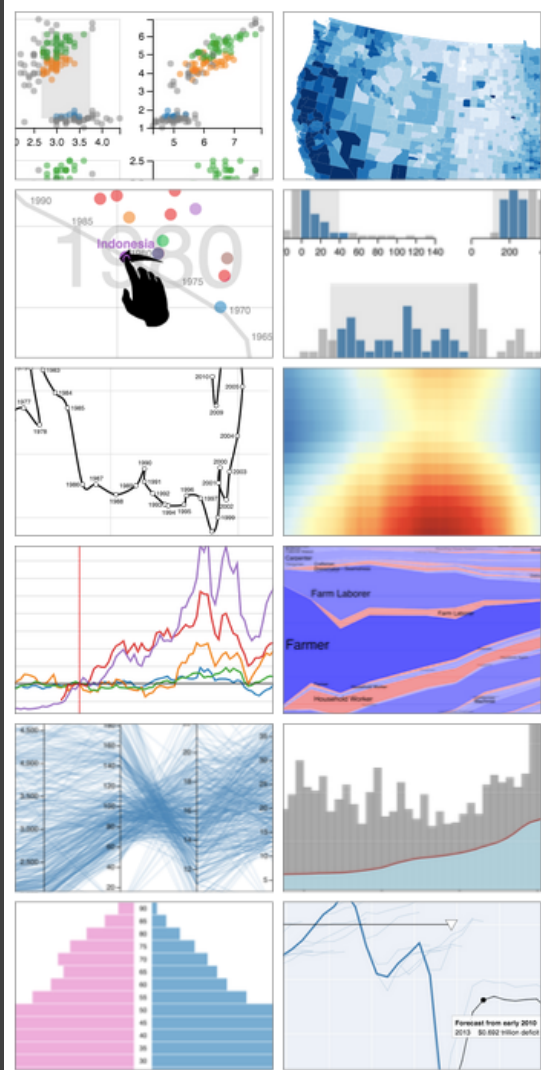
Sampling Methods

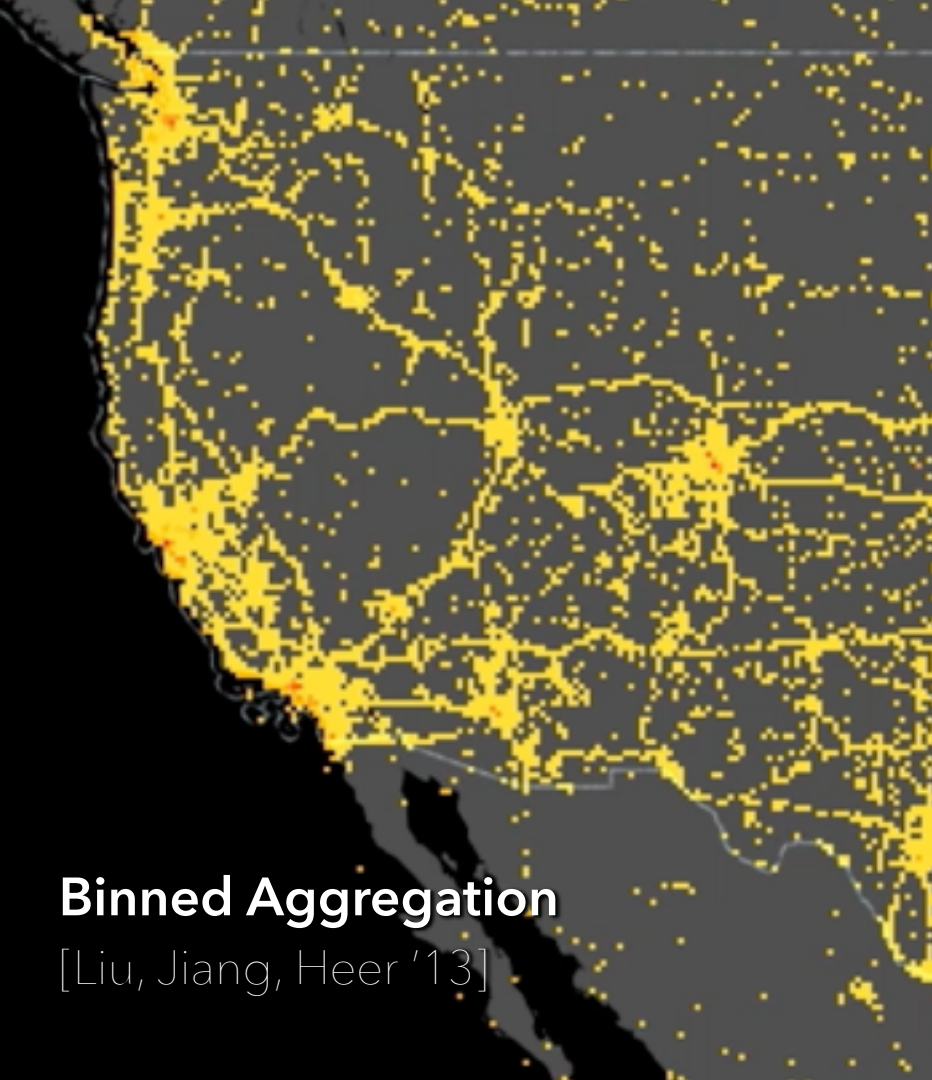
Common Sampling Methods

First-N: Useful for transformation, but not inference.

Random: Good default, but may miss features of interest. Possible in one pass via reservoir sampling, or faster if stored in randomized order.

Stratified: Sample within groups, ensure coverage and balance across those categories.





Binned Aggregation

[Liu, Jiang, Heer '13]



Sampling

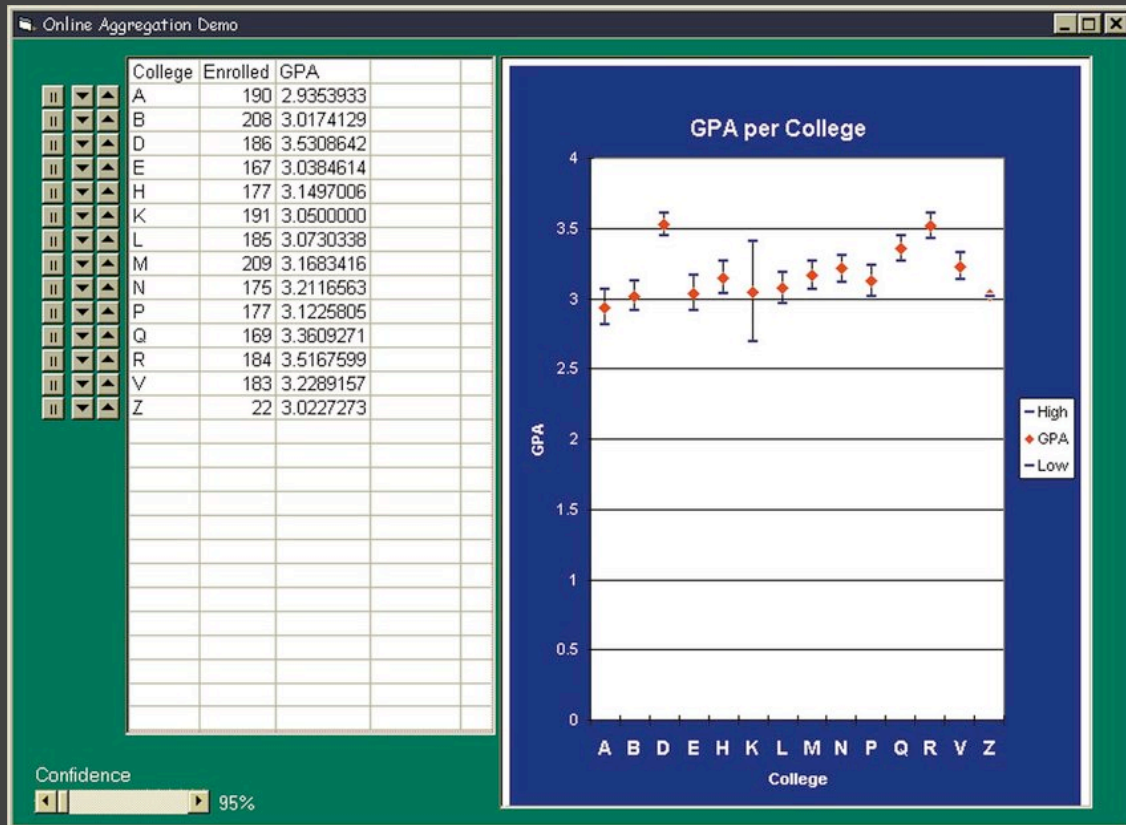
Google Fusion Tables

Online Aggregation [Hellerstein, Haas, Wang '97]

Provide dynamic, *progressive* results as queries run: see results over growing samples.

Visualize current results with confidence intervals to convey uncertainty of estimate.

Challenge: difficult to ensure truly random sampling.

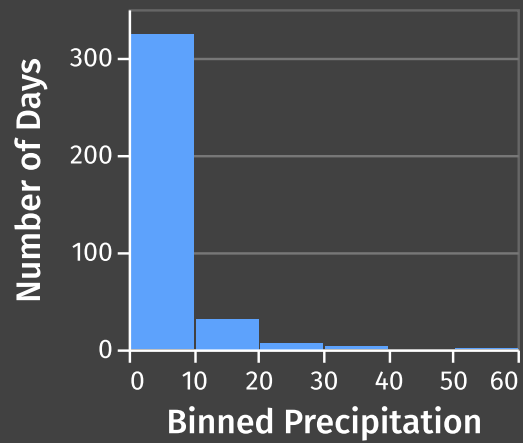


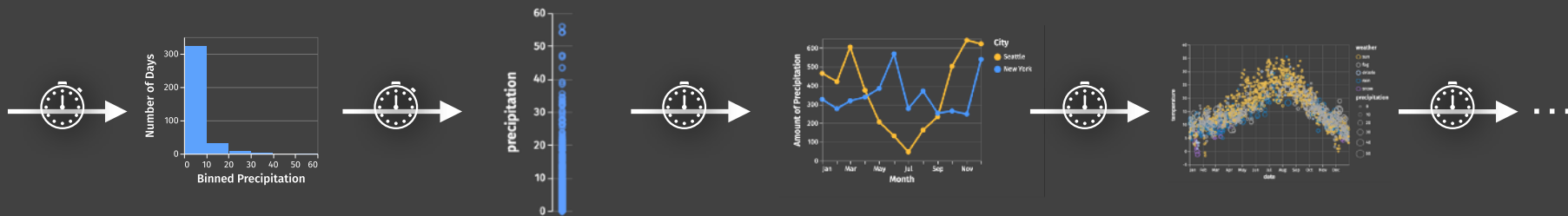
**What if data is too large to
query in a reasonable time?**

Trust, but Verify: Optimistic Vis

[Moritz, Fisher, Ding & Wang '17]

Strategies: Query Database, Approximation

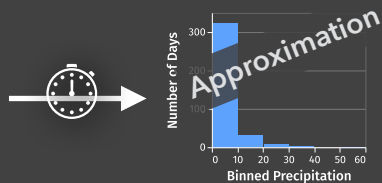




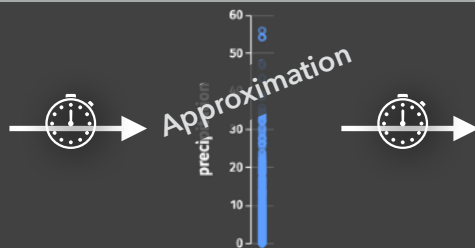
Latencies reduce engagement
and lead to fewer observations.

The Effect of Interactive Latency. Liu, Heer. *IEEE InfoVis 2014*.

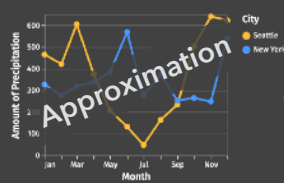
Small chance
of error



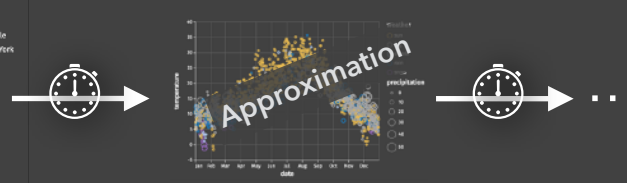
Small chance
of error
Very likely to have at least one error



Small chance
of error



Small chance
of error



Approximation: Trade Accuracy for Speed

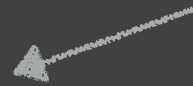
- Approximate query processing (AQP)
- Uncertainty estimation in statistics
- Uncertainty visualization
- Probabilistic programming
- Approximate hardware

Pick your poison:

1. Trust the approximation, or
2. Wait for everything to complete.



This glass
is half full



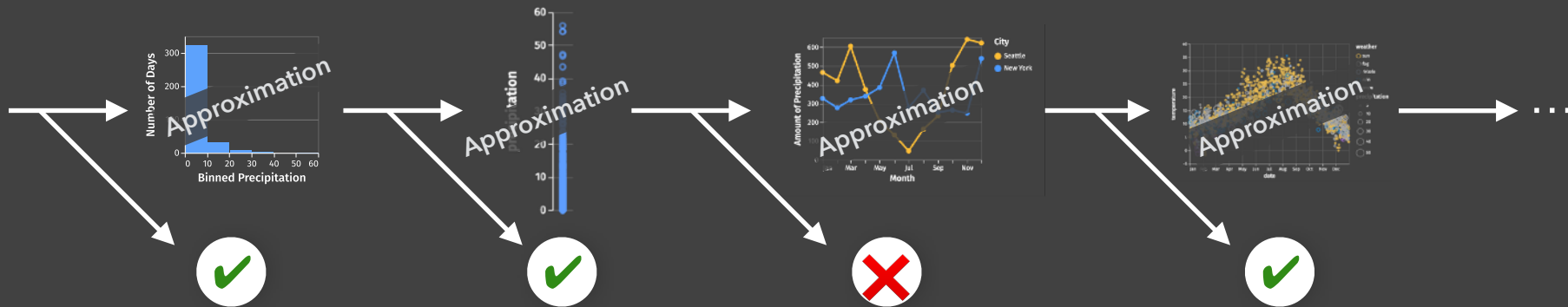
Optimistic Visualization

Trust but Verify

What if we think of the
issues with approximation as
user experience problems?

Optimistic Visualization

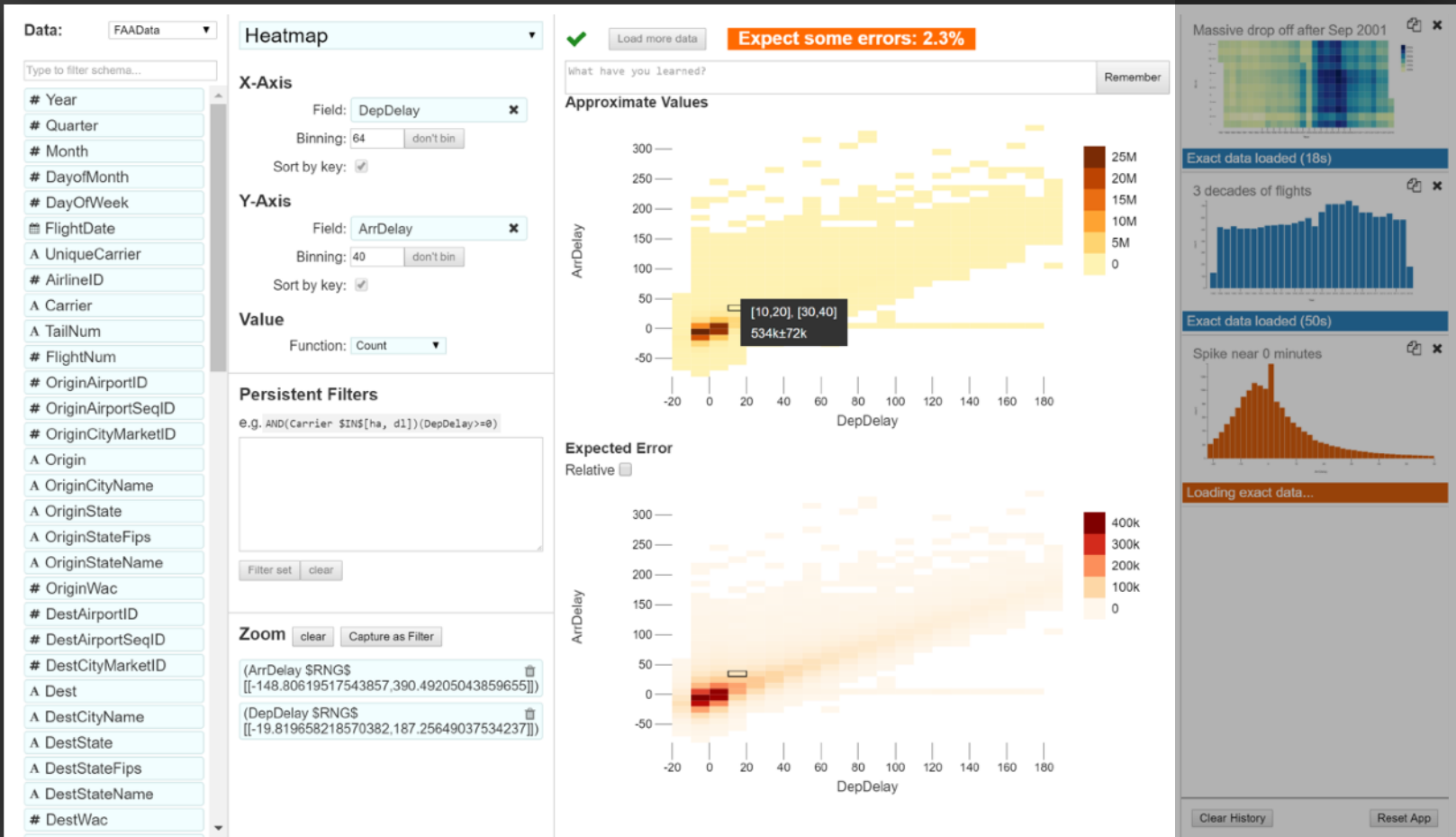
Trust but Verify. Moritz et al. *CHI 2017*.



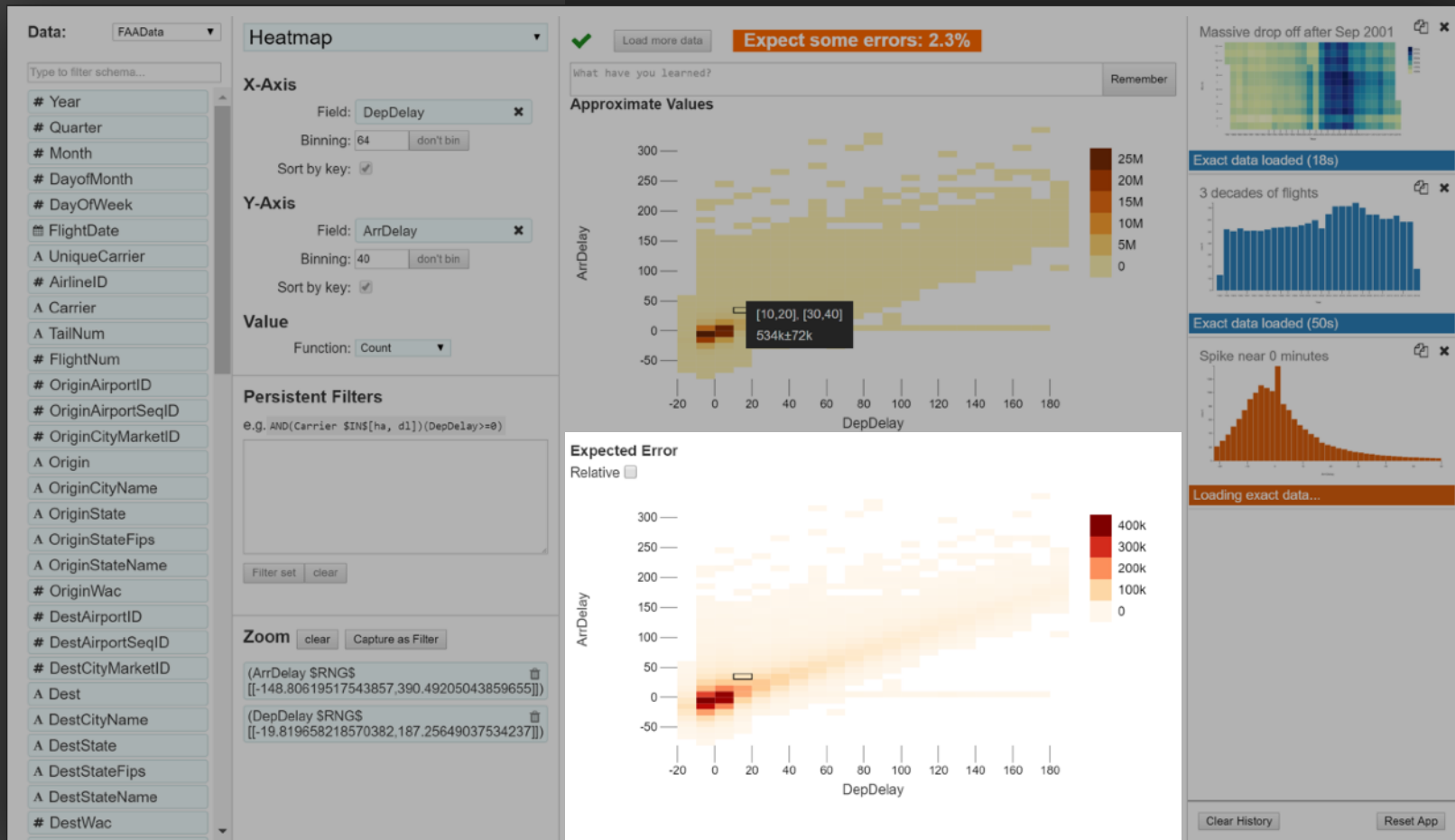
1. Analysts uses initial estimates.
2. Precise queries run in the background.
3. System confirms results. Analyst detects errors.

Analysts can use approximations and also trust them.

Optimistic Visualization



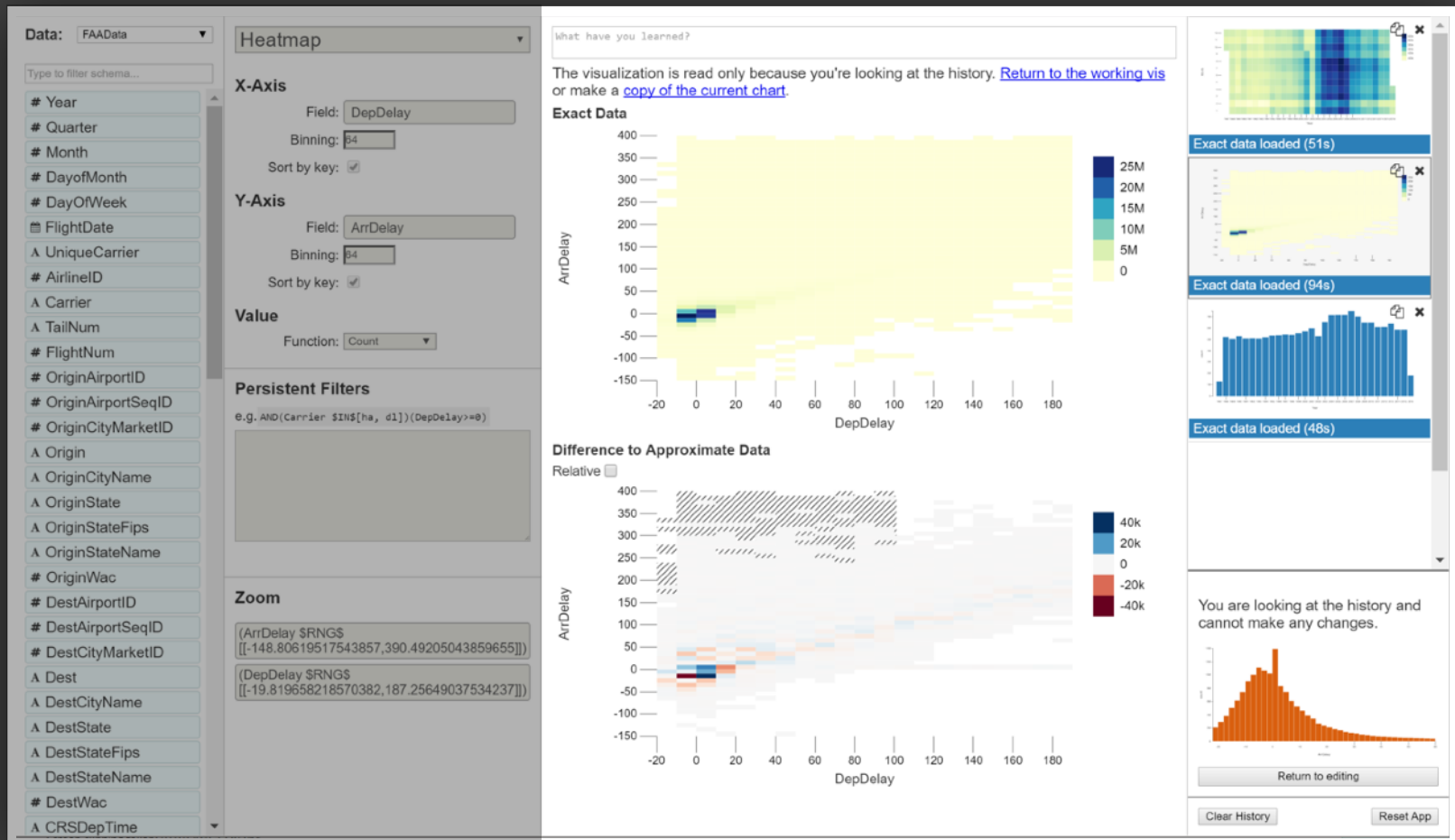
Visualize Uncertainty



Show a History of Previous Charts



Help Analysts Confirm Results



Evaluation

Case studies with teams at Microsoft who brought in their own data.

Approximation works

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

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.”

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.”*

In Conclusion...

Two Challenges:

1. Effective **visual encoding**
2. Real-time **interaction**

Perceptual and interactive scalability should be limited by the **chosen resolution** of the visualized data, not the number of records.

Bin > Aggregate (> Smooth) > Plot

1. **Bin** Divide data domain into discrete “buckets”
2. **Aggregate** Count, Sum, Average, Min, Max, ...
3. **Smooth** *Optional*: smooth aggregates [Wickham '13]
4. **Plot** Visualize the aggregate values

Interactive Scalability Strategies

1. Query Database
2. Client-Side Indexing / Data Cubes
3. Prefetching
4. Approximation

These strategies are **not** mutually exclusive!
Systems can apply them in tandem.