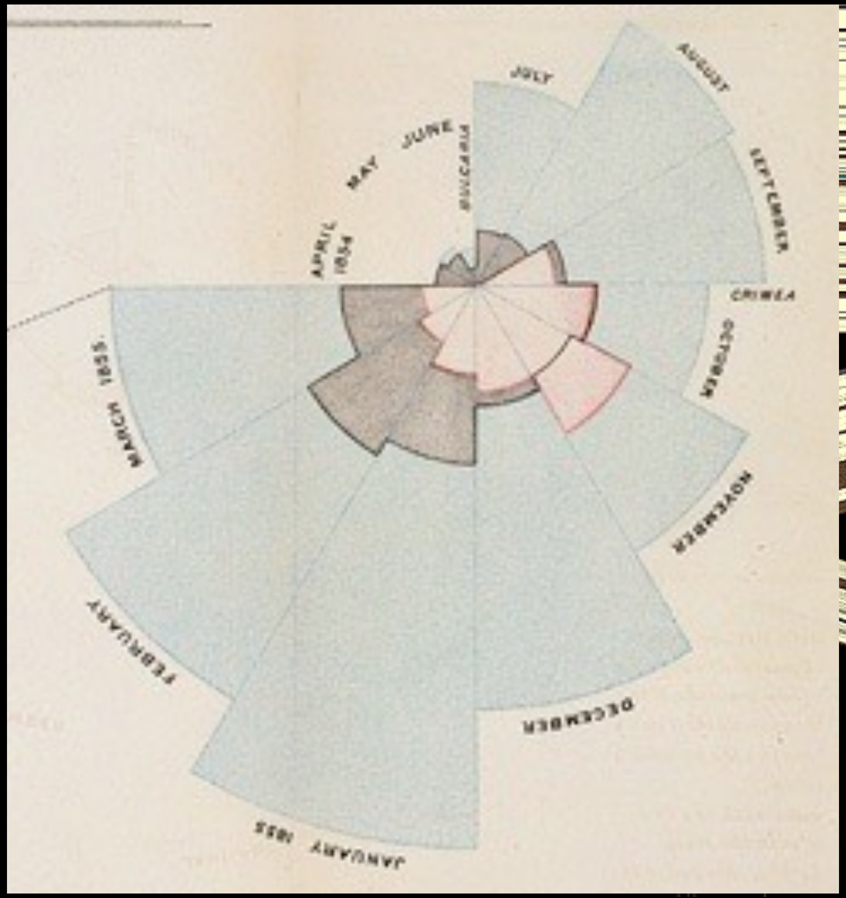
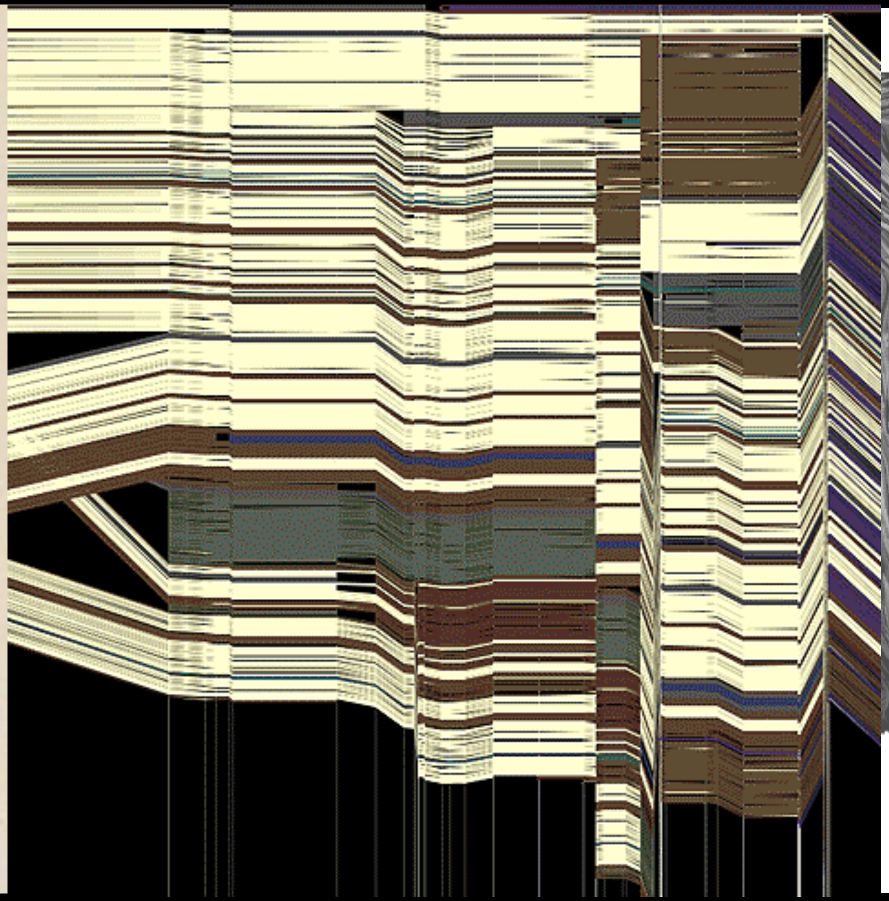
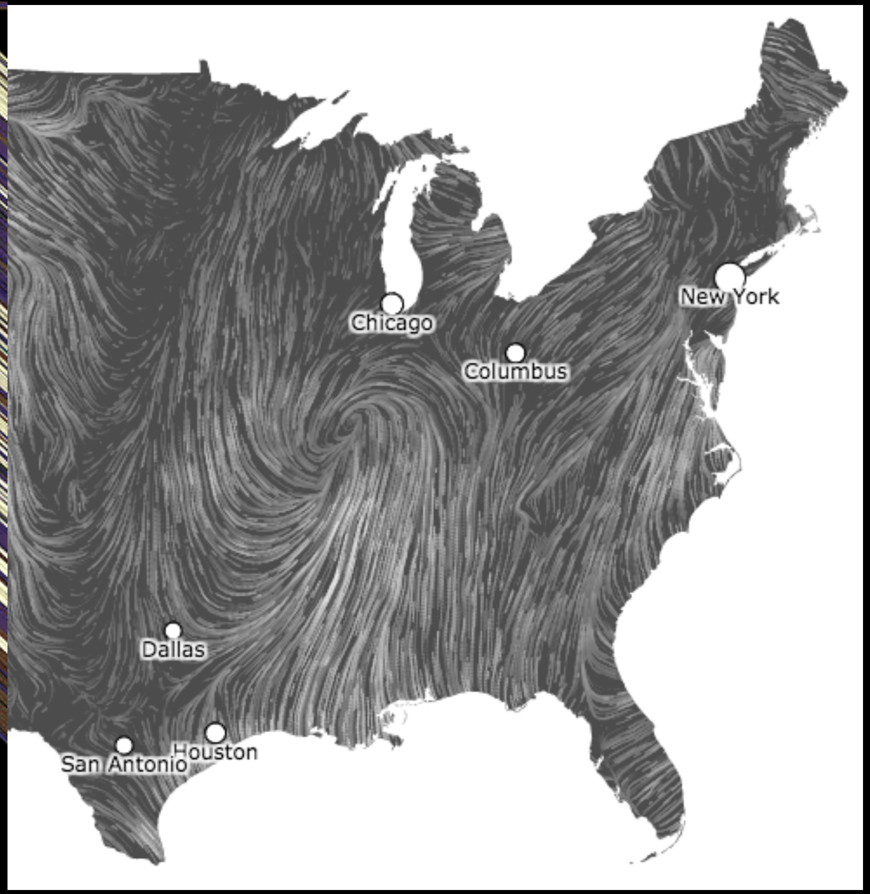
CSE 442 - Data Visualization

Scalable Visualization



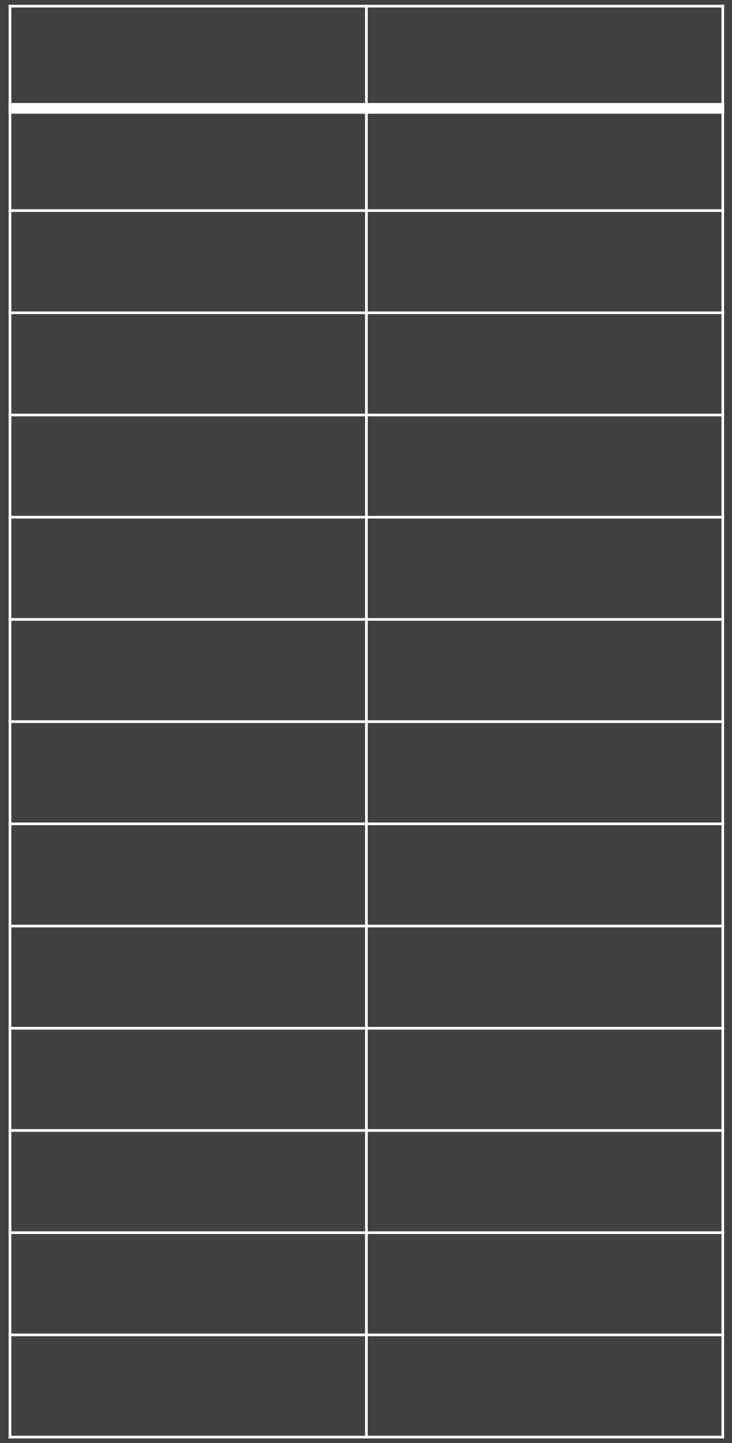




Jeffrey Heer University of Washington

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?

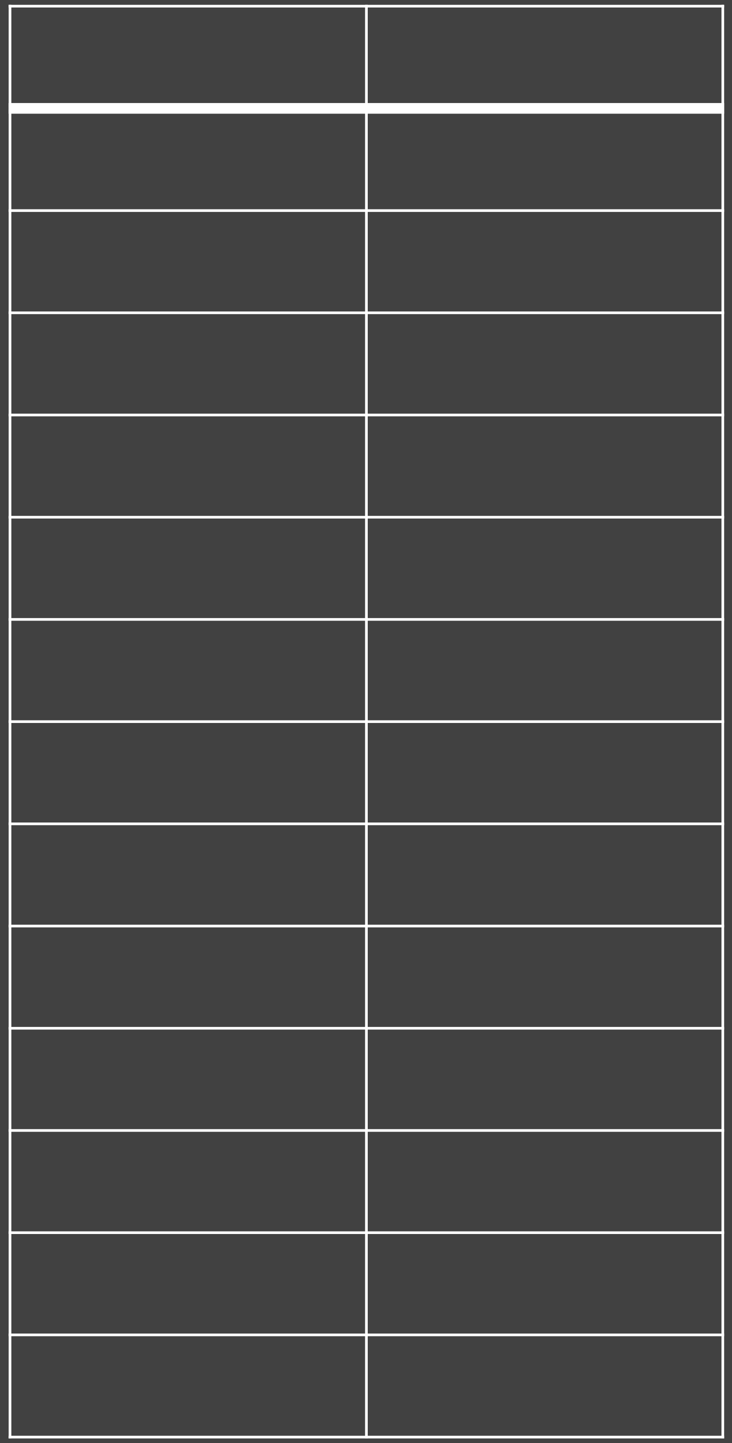
Parallel data processing

Reduction: Filter, aggregate

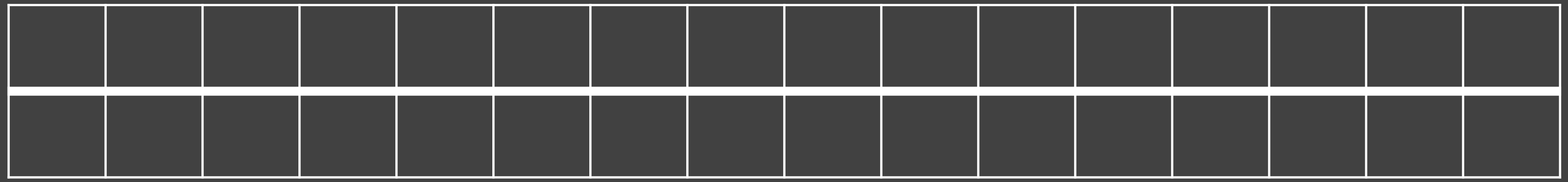
Sample or approximate

Not just about systems. Consider perceptual / cognitive scalability.

Tall Data



Wide data

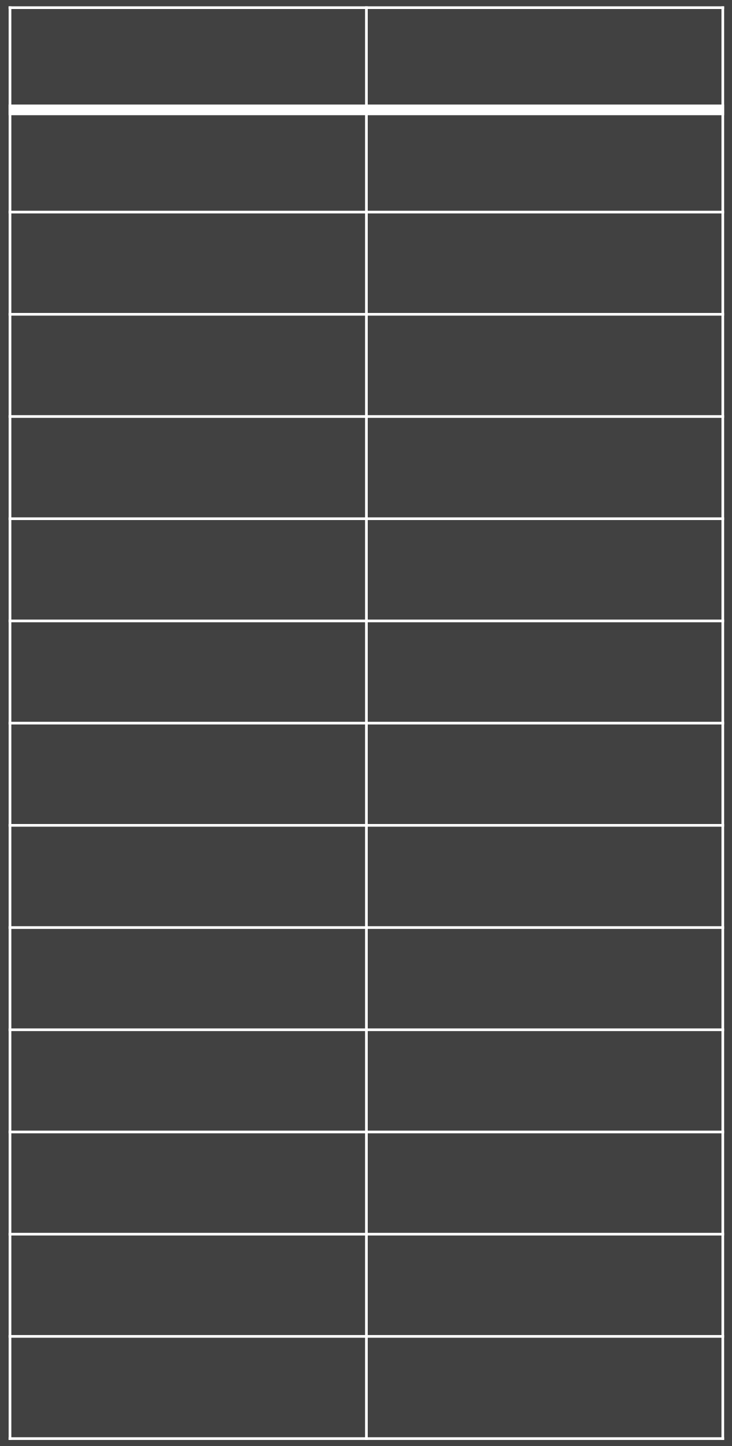


Lots of variables (100s-1000s…)

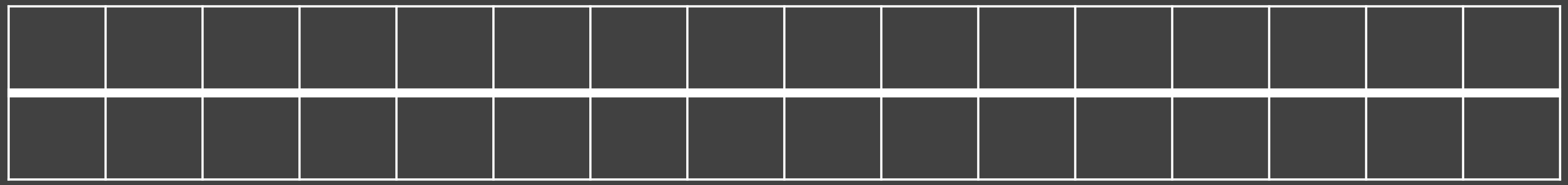
- Select relevant subset
- Dimensionality reduction
- Statistical methods can suggest and order related variables

Requires human judgment

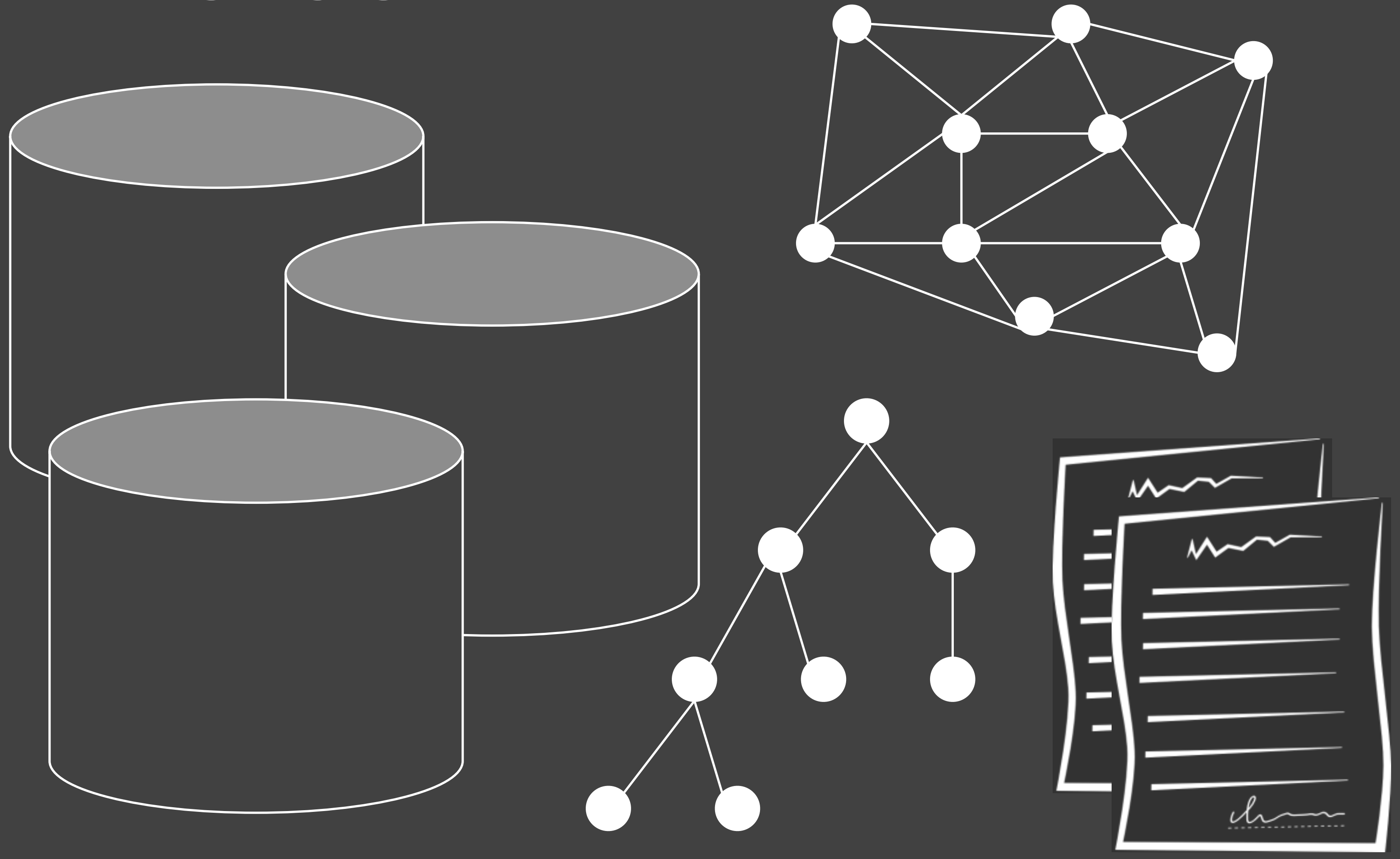
Tall Data



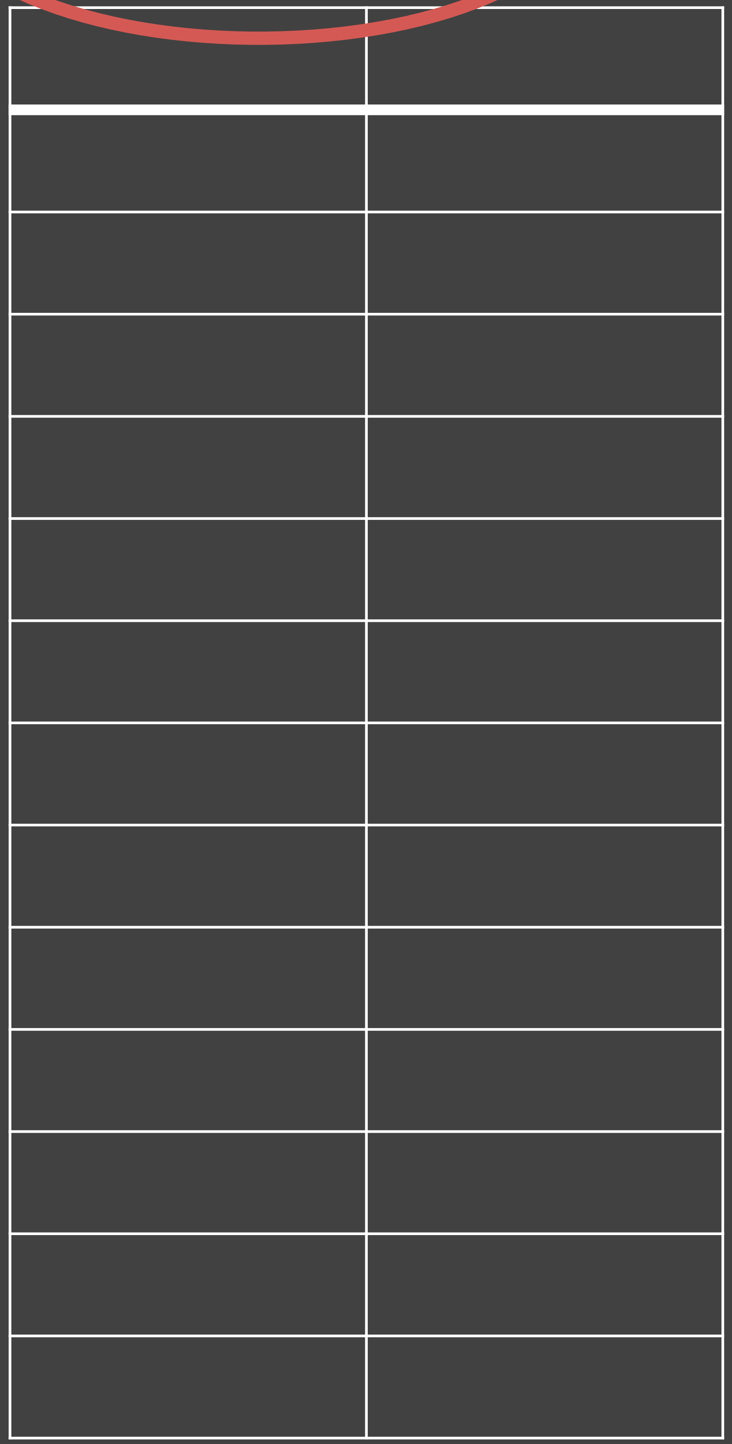
Wide data



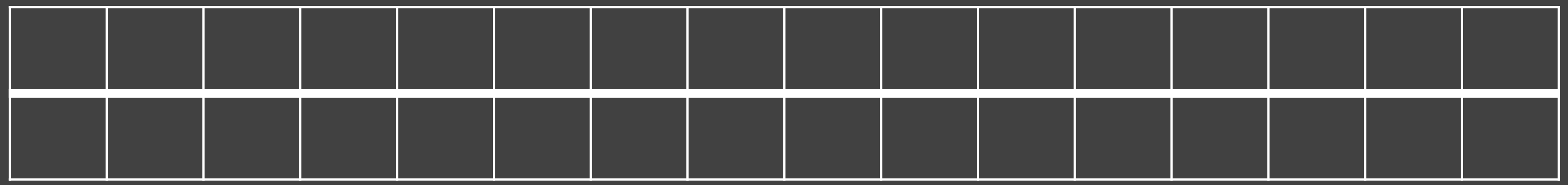
Diverse data



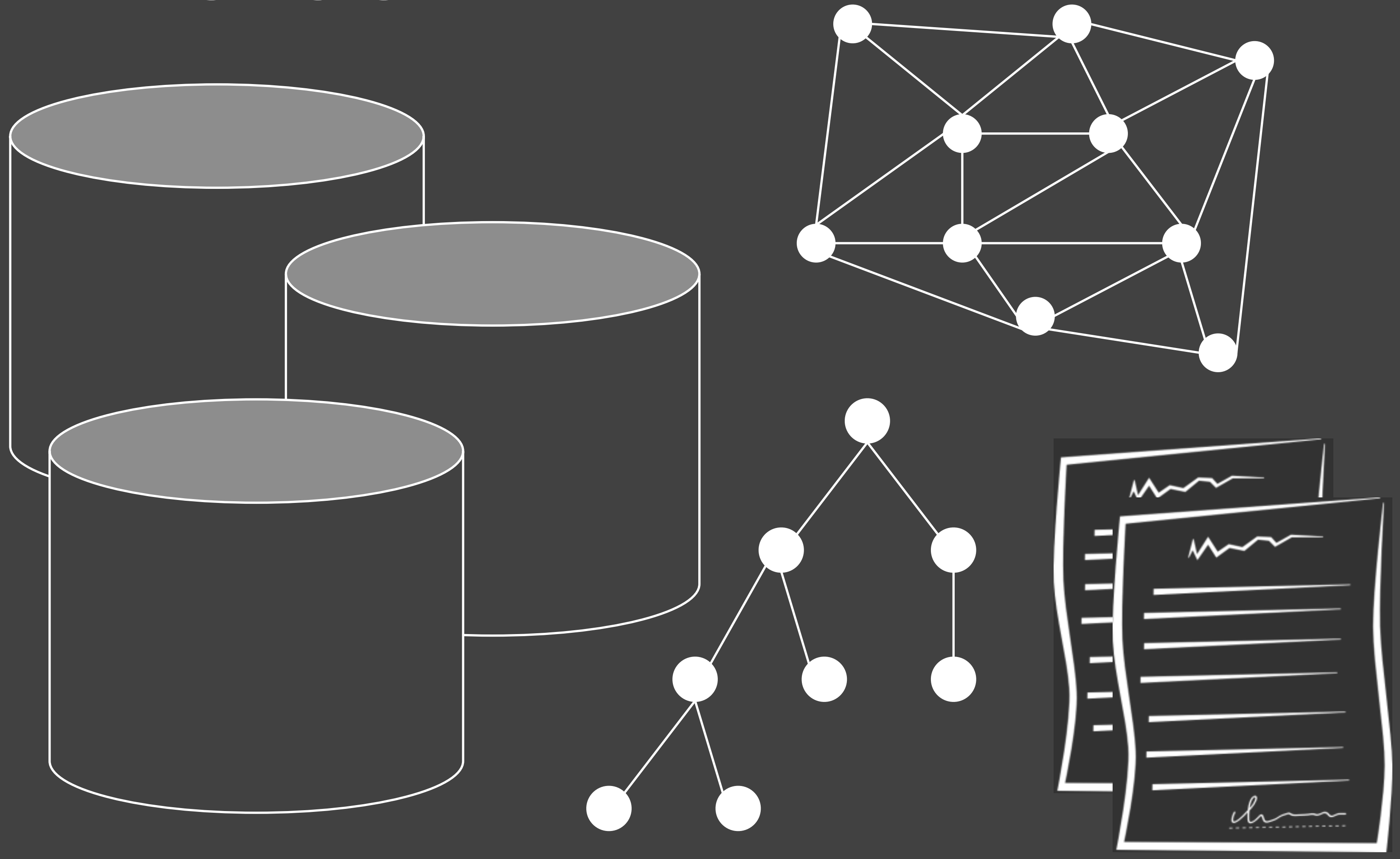
Tall Data



Wide data



Diverse data



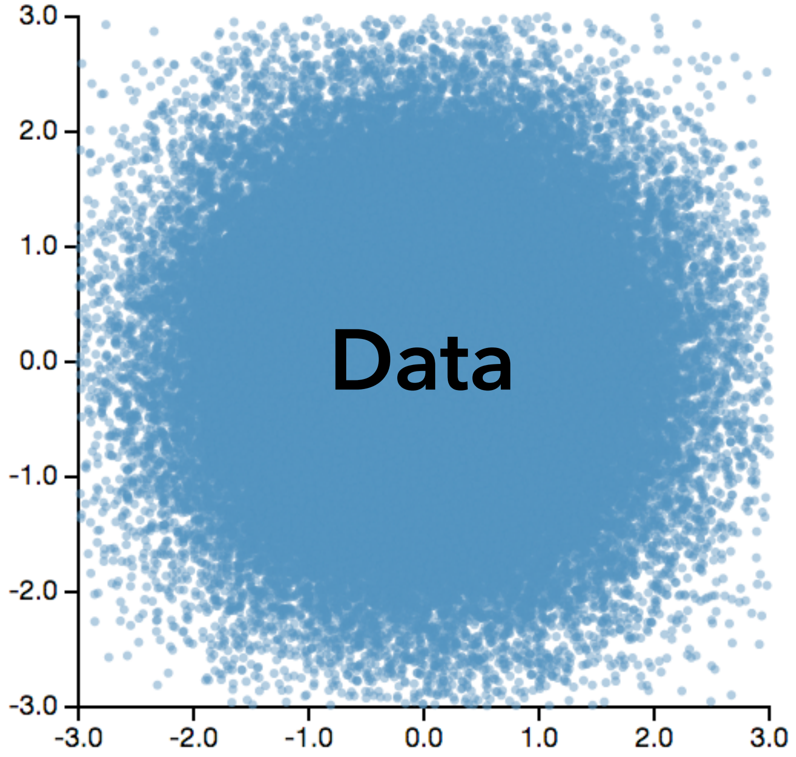
How can we visualize and interact with **billion+ record** databases in real-time?

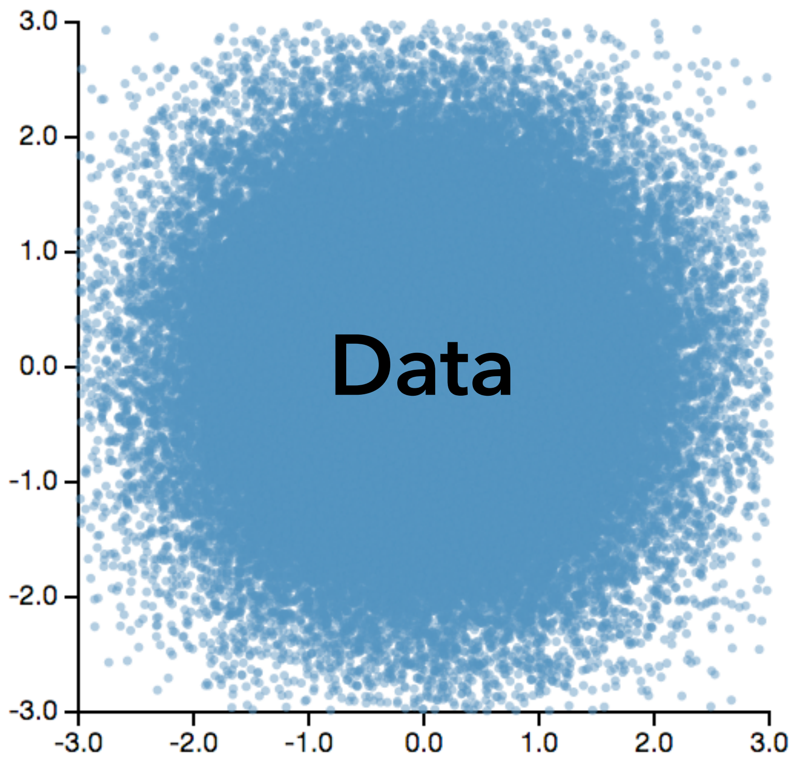
Two Challenges:

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

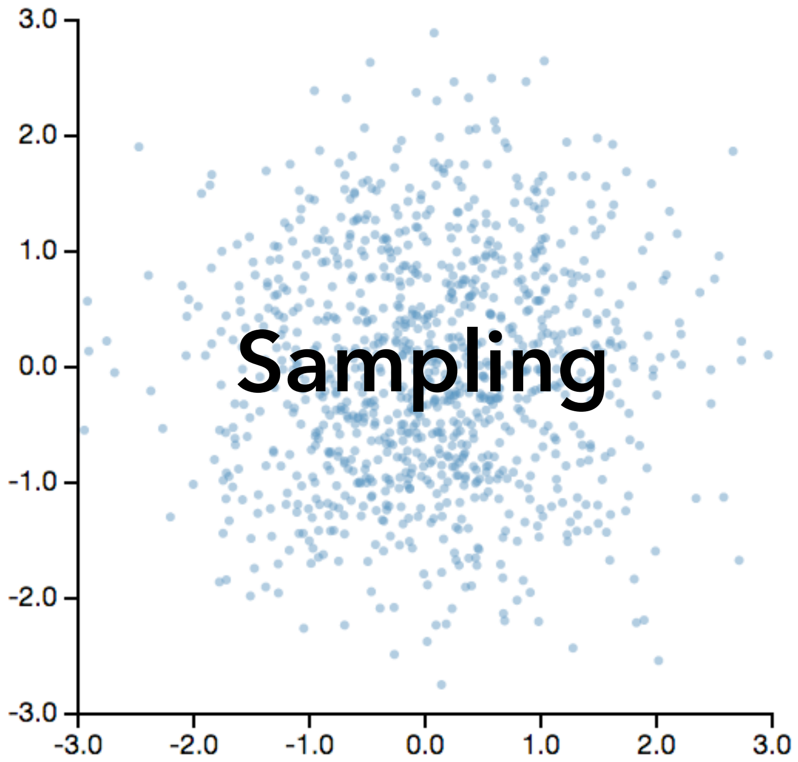
Perceptual and interactive scalability should be limited by the **chosen resolution** of the visualized data, not the number of records.

1. Visualizing Large Datasets

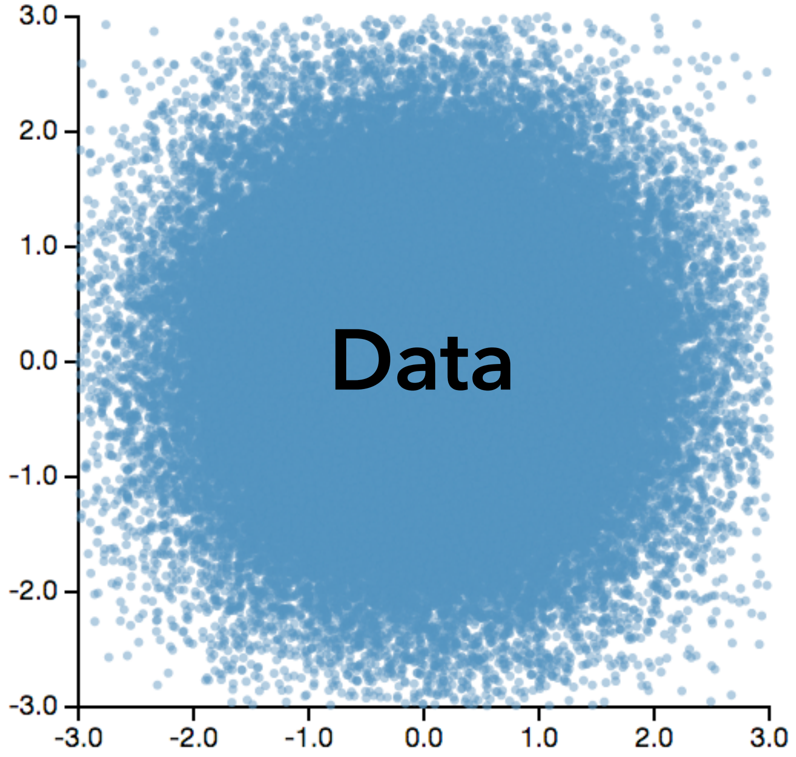

3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
Data
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

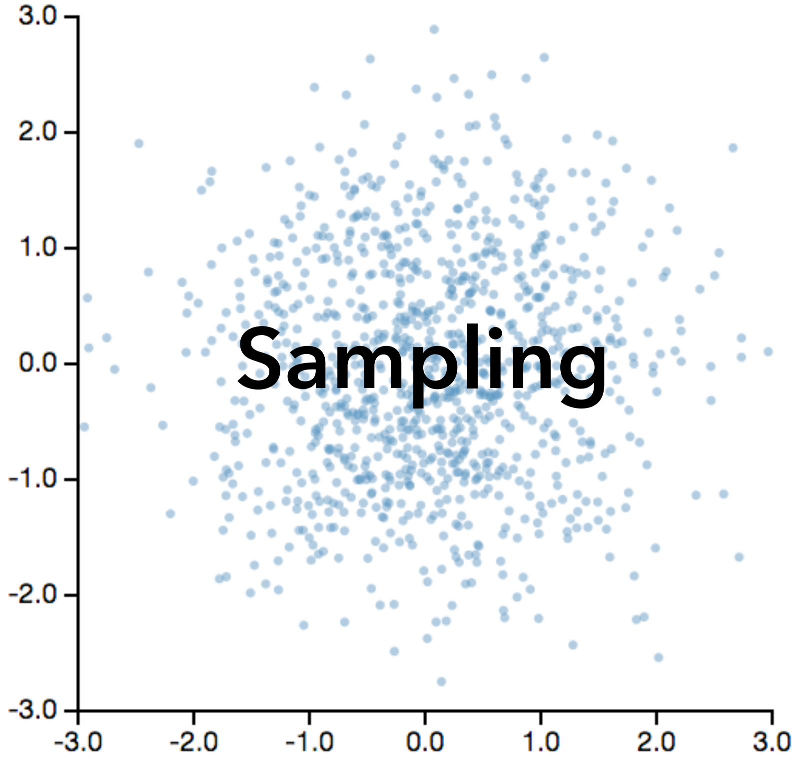


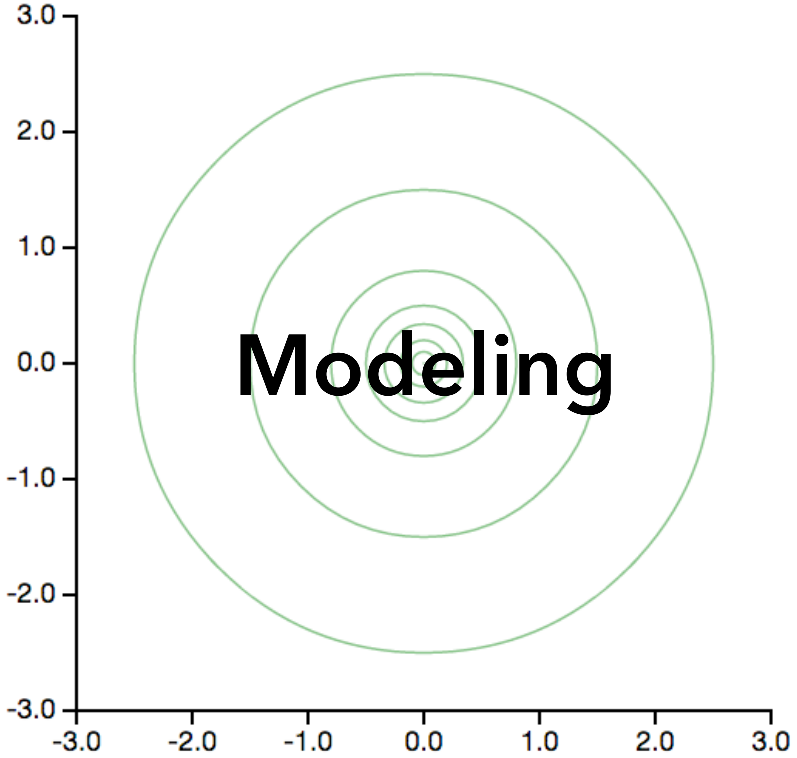
Data
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

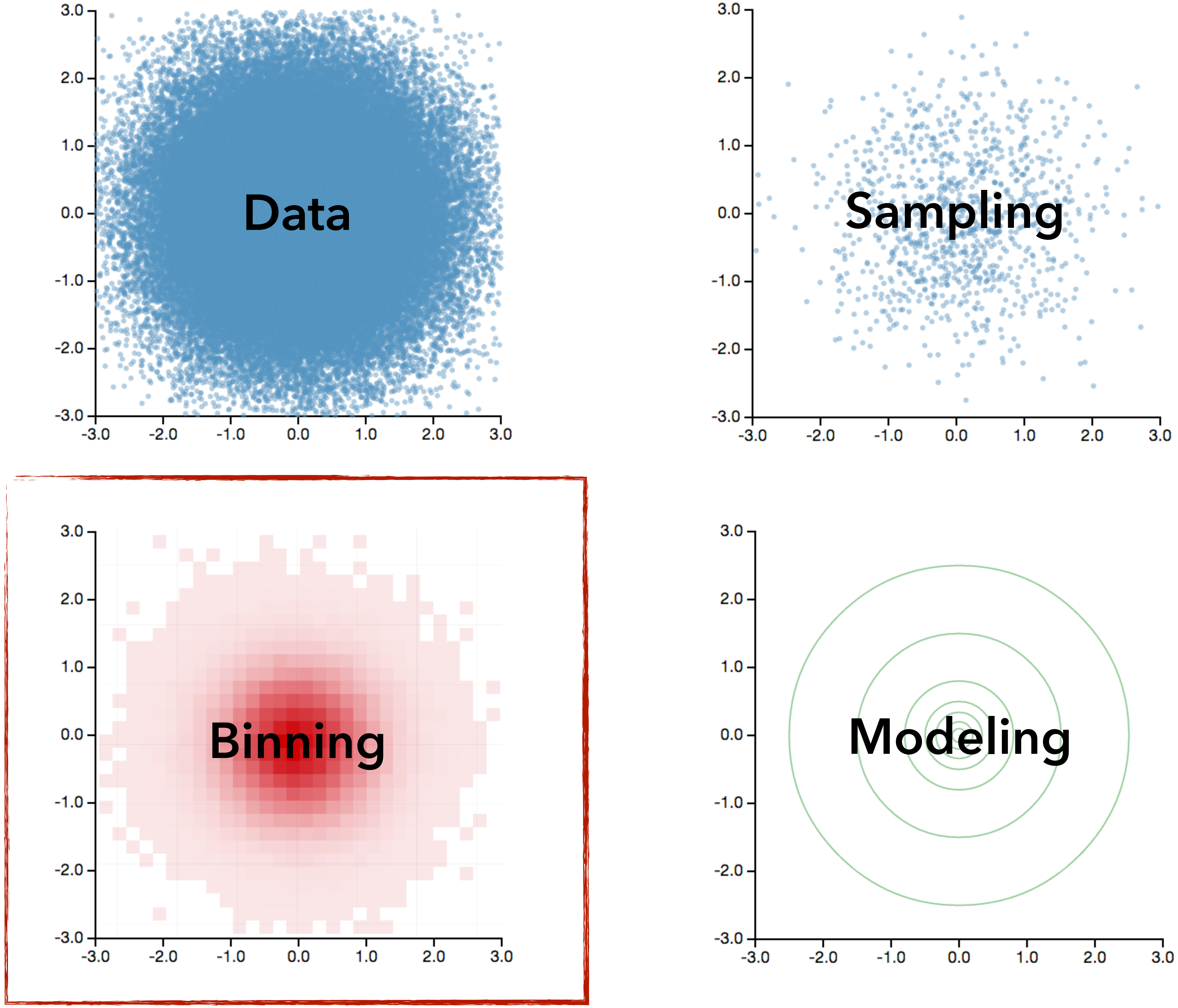


Sampling
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

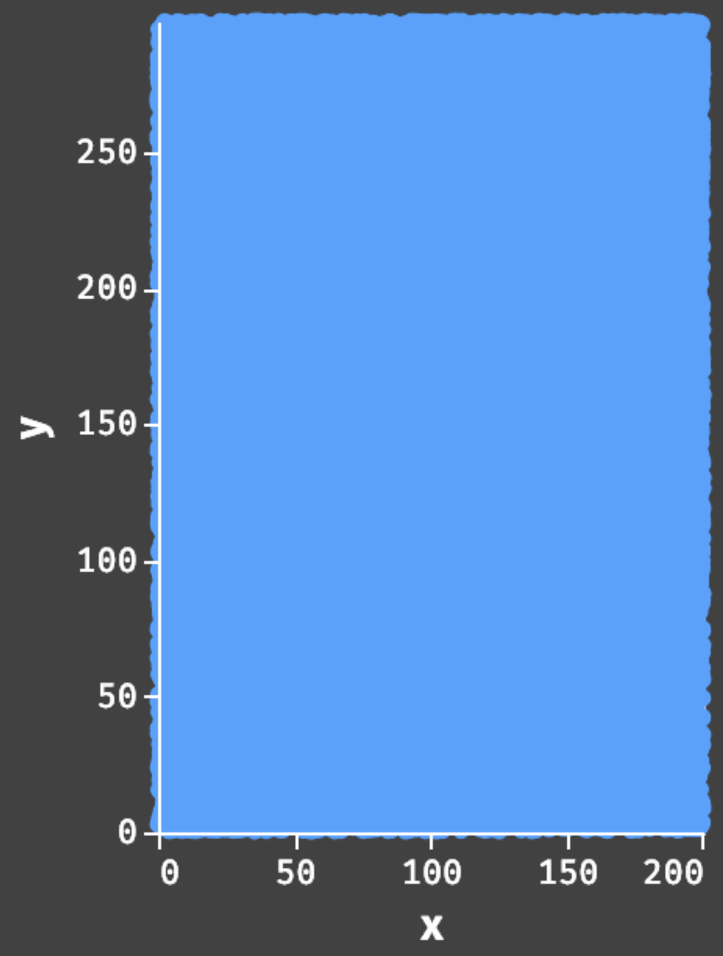

Data
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0


Sampling
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

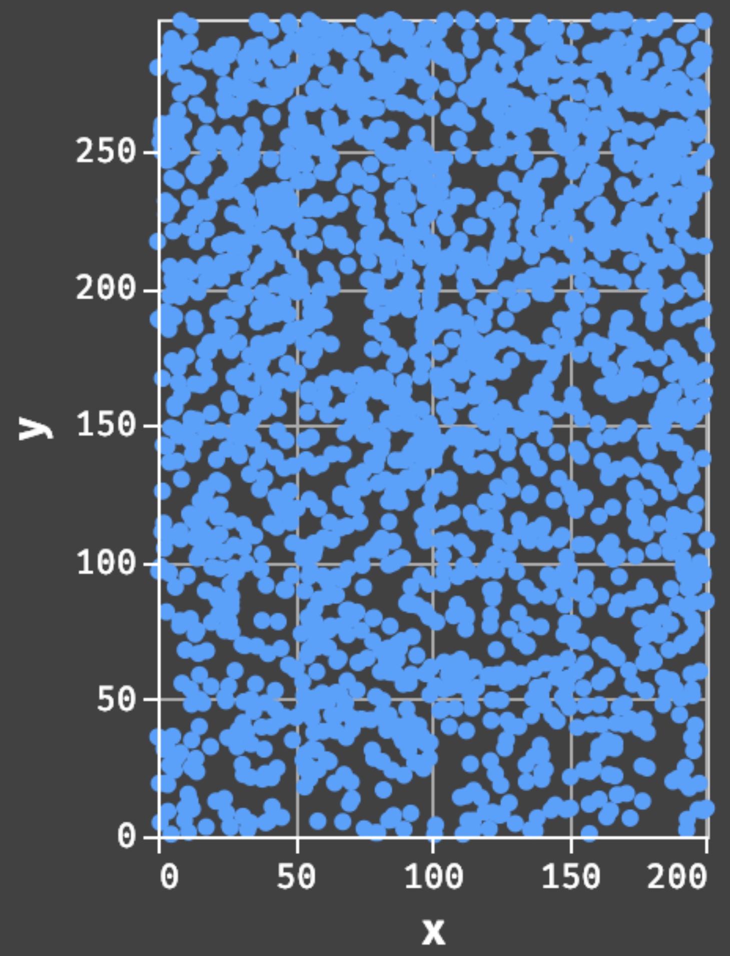

Modeling
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0


Data
Sampling
Binning
Modeling
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0

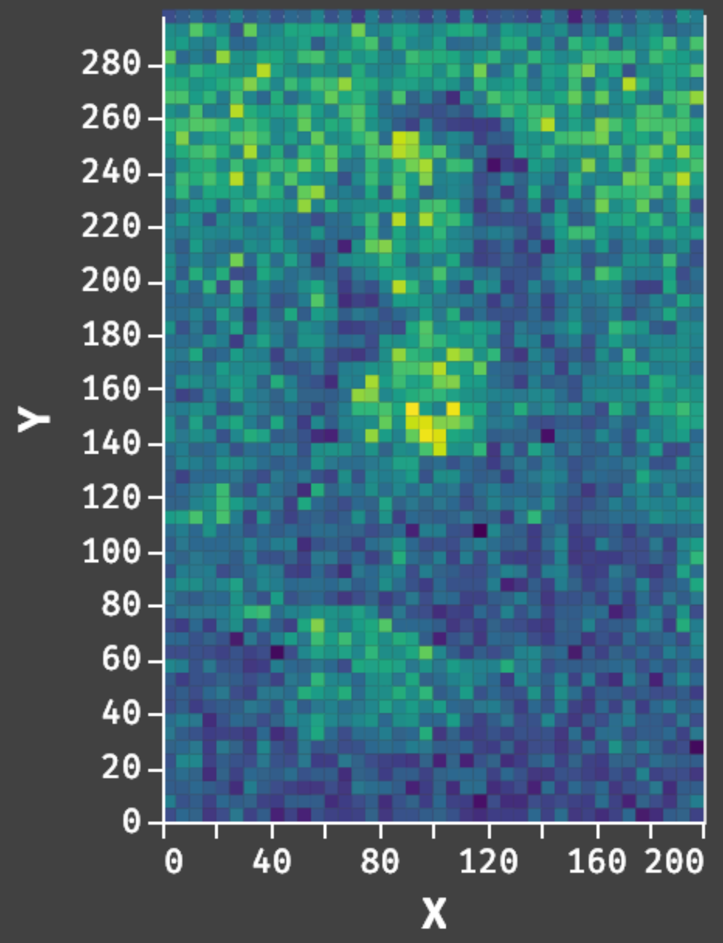
How to **Visualize** a Billion+ Records



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.

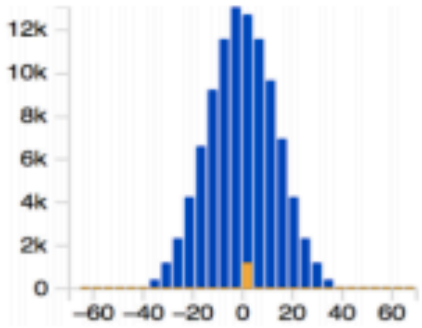
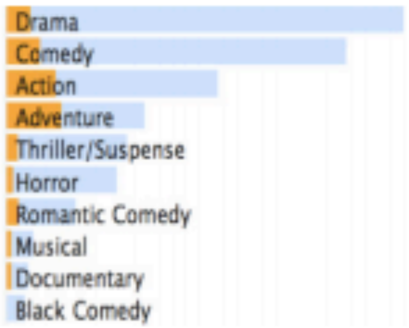
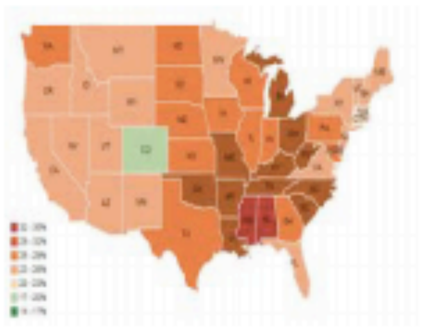
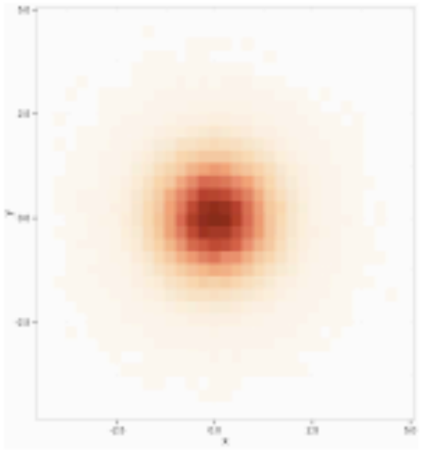
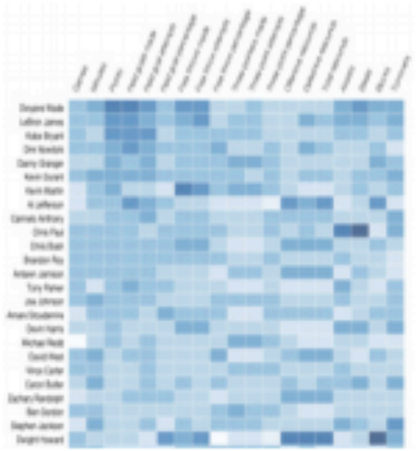
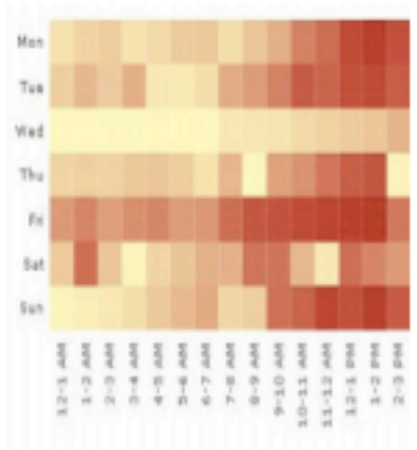
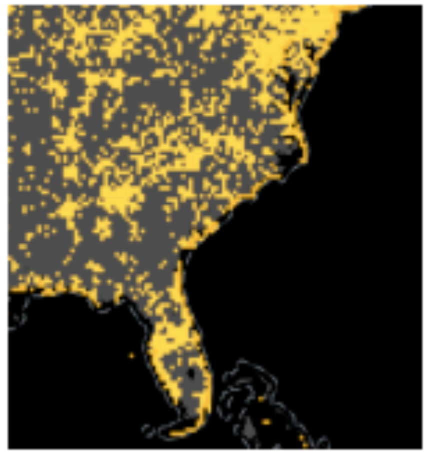
Geo: Bin x, y coordinates *after* cartographic projection

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

(**3. Smooth** Optional: smooth aggregates [Wickham ’13])

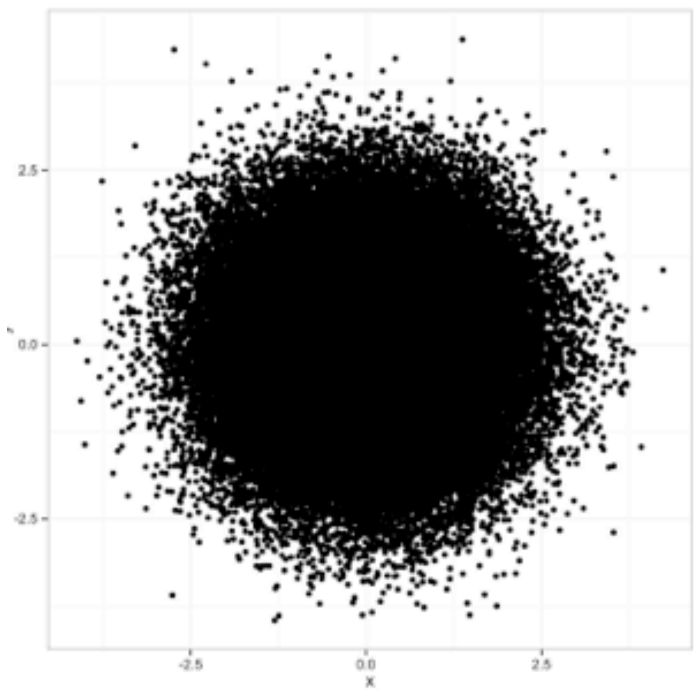
4. Plot Visualize the aggregate values

Binned Plots by Data Type

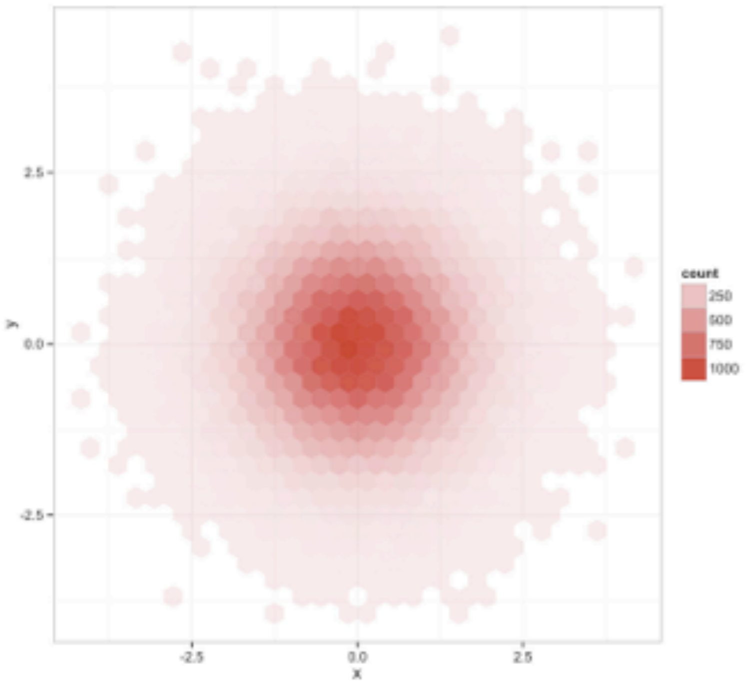
	Numeric	Ordinal	Temporal	Geographic
1D	 Histogram	 Bar Chart	 Line Graph / Area Chart	 Choropleth Map
2D	 Binned Scatter Plot	 Heatmap	 Temporal Heatmap	 Geographic Heatmap

Design Subtleties…

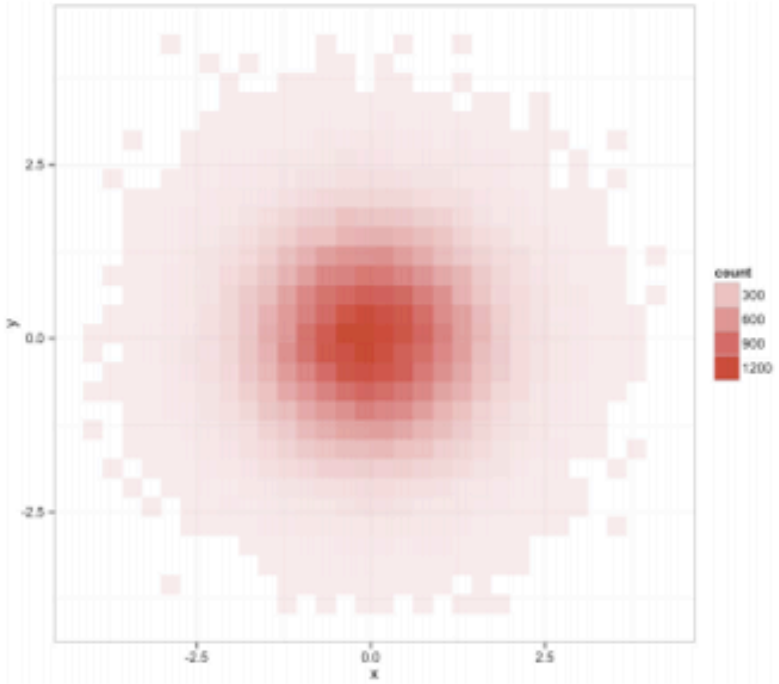
Hexagonal or Rectangular Bins?



100,000 Data Points



Hexagonal Bins

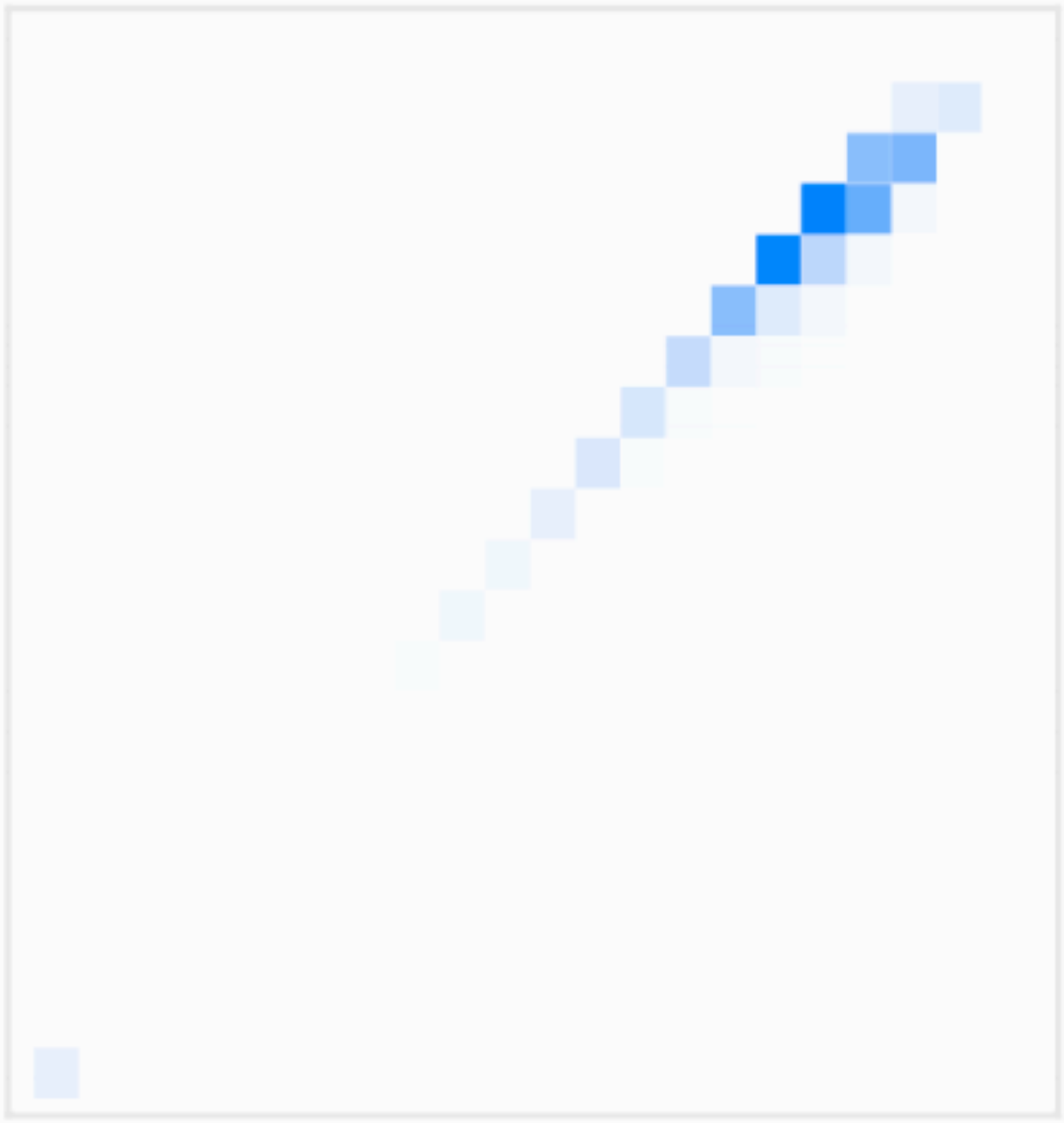


Rectangular Bins

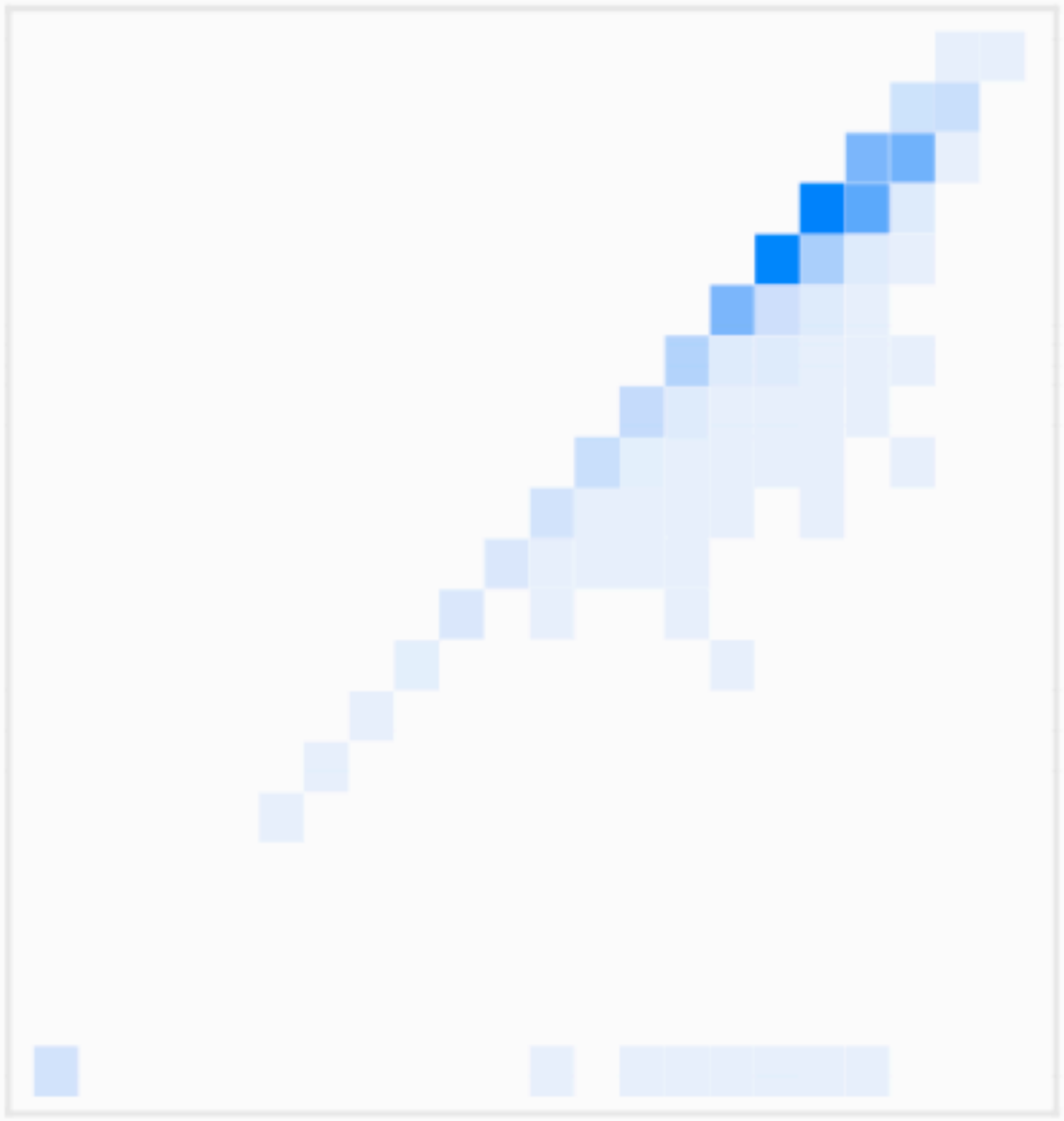
Hex bins better estimate density for 2D plots, but *the improvement is marginal* [Scott 92].

Rectangles support *reuse* and *visual queries*.

Color Scale: Discontinuity after Zero

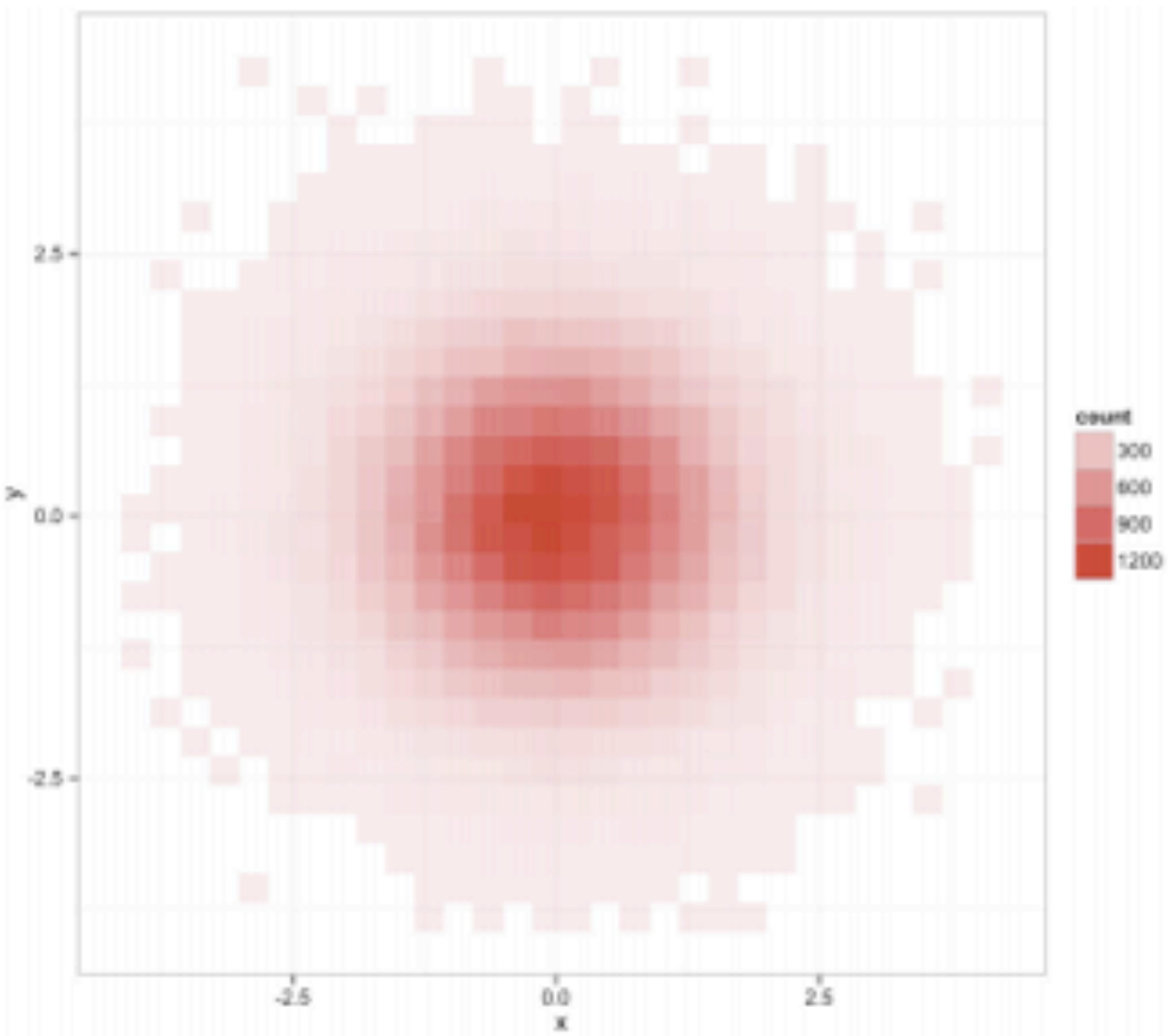


Standard Color Ramp
Counts near zero are white.

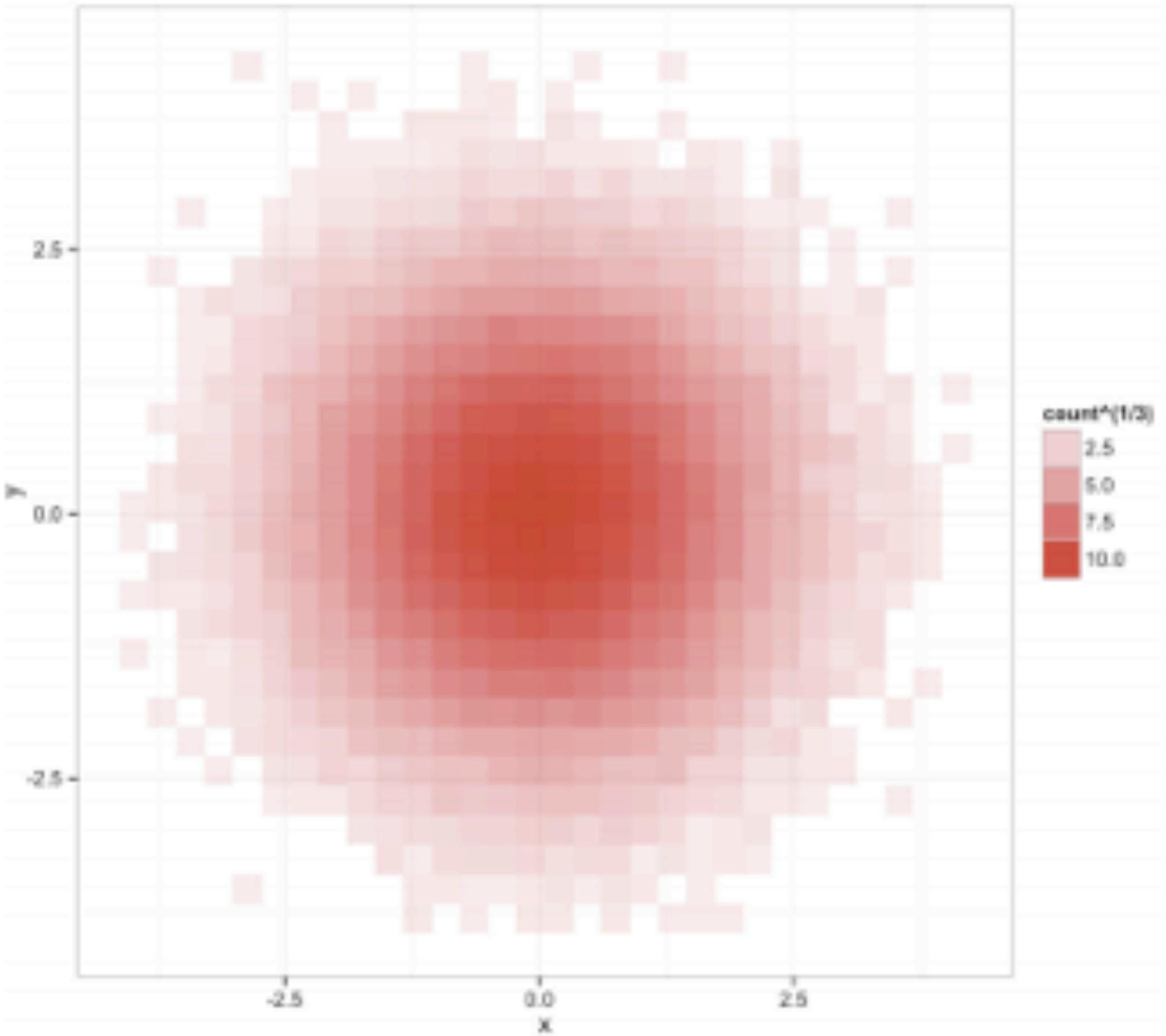


Add Discontinuity after Zero
Counts near zero remain visible.

Color / Opacity Ramps



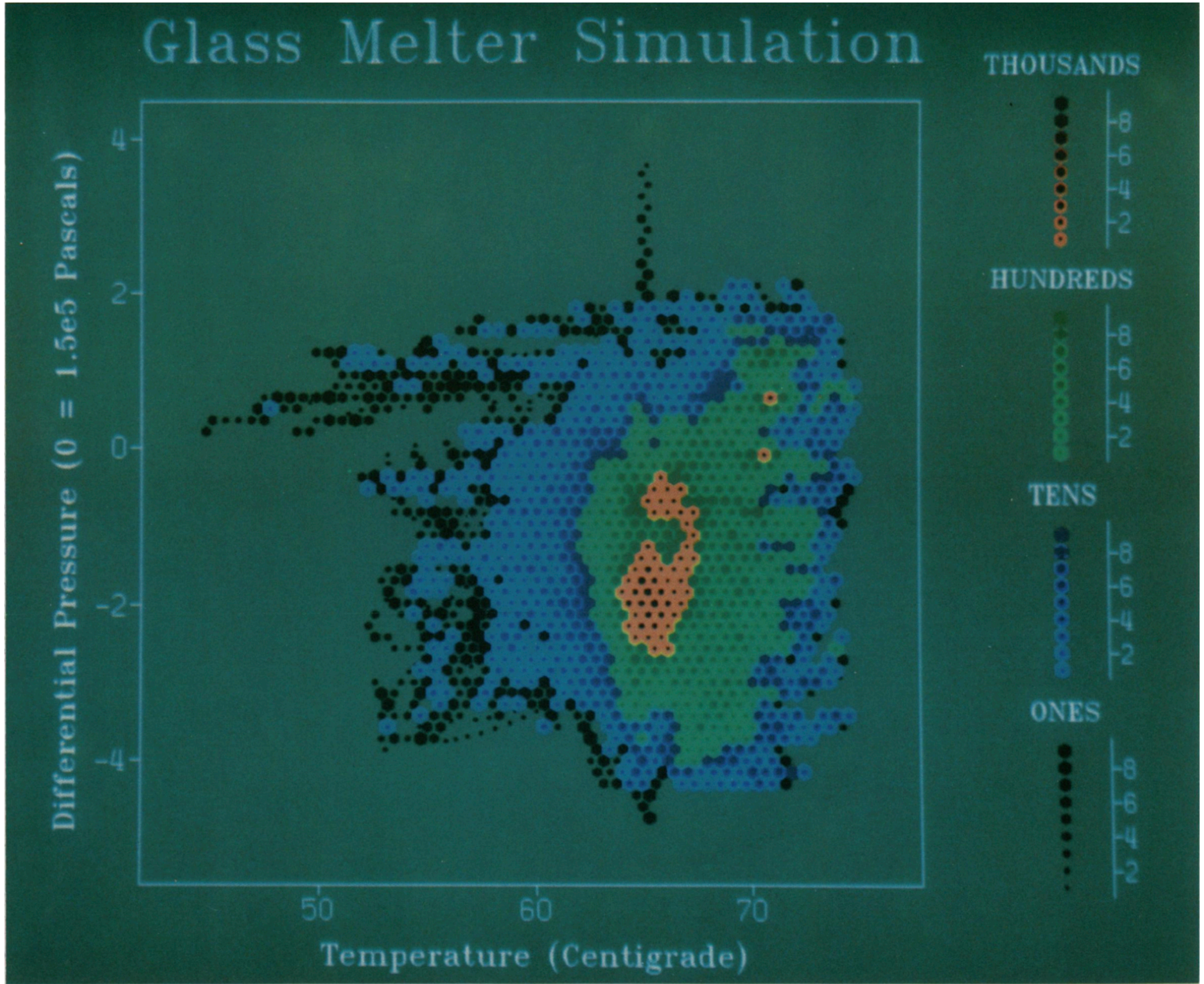
Linear interpolation in RGBA
is not perceptually linear.



Perceptual color spaces
approximate perceptual linearity.

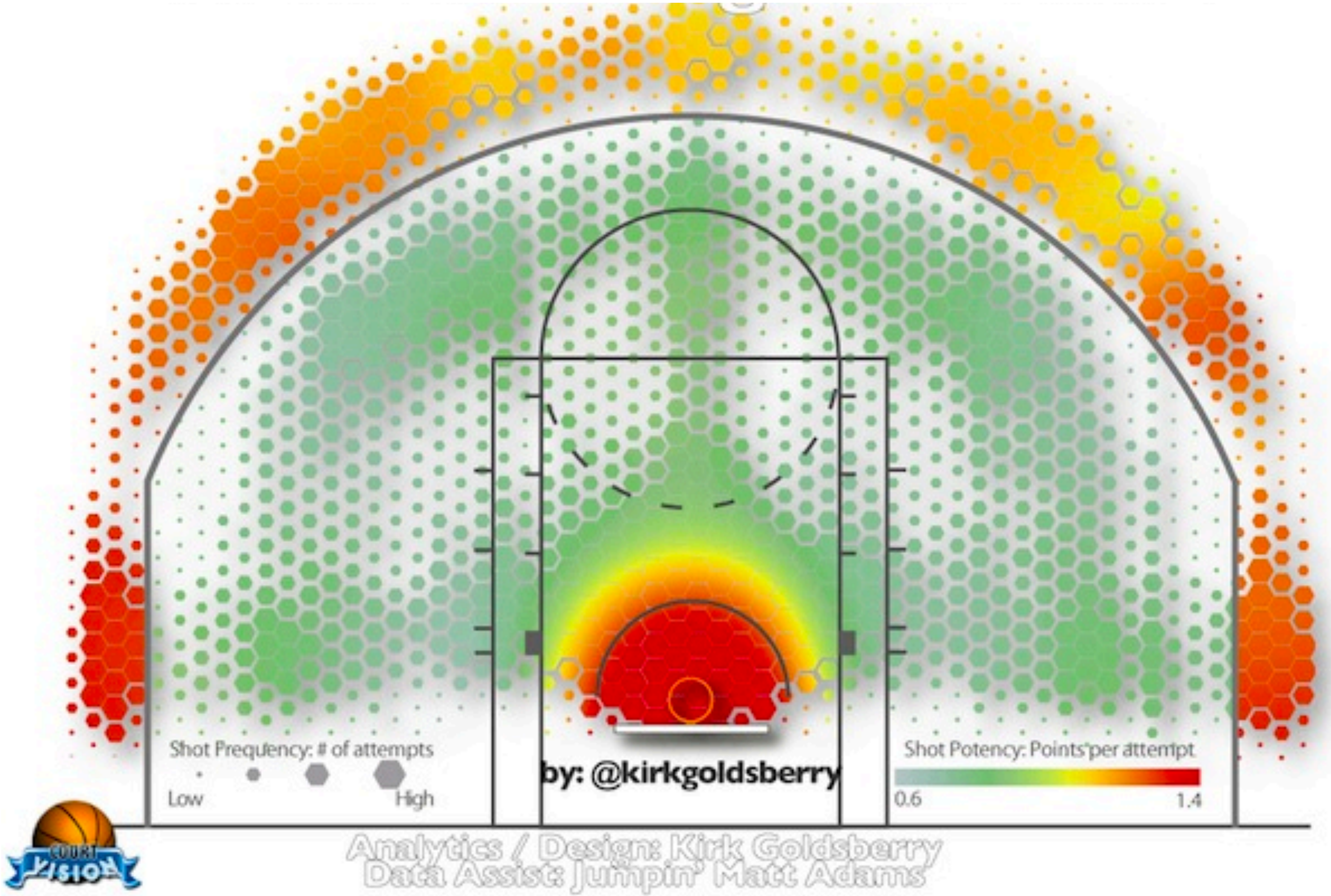
Examples

Example: Binned SPLOM



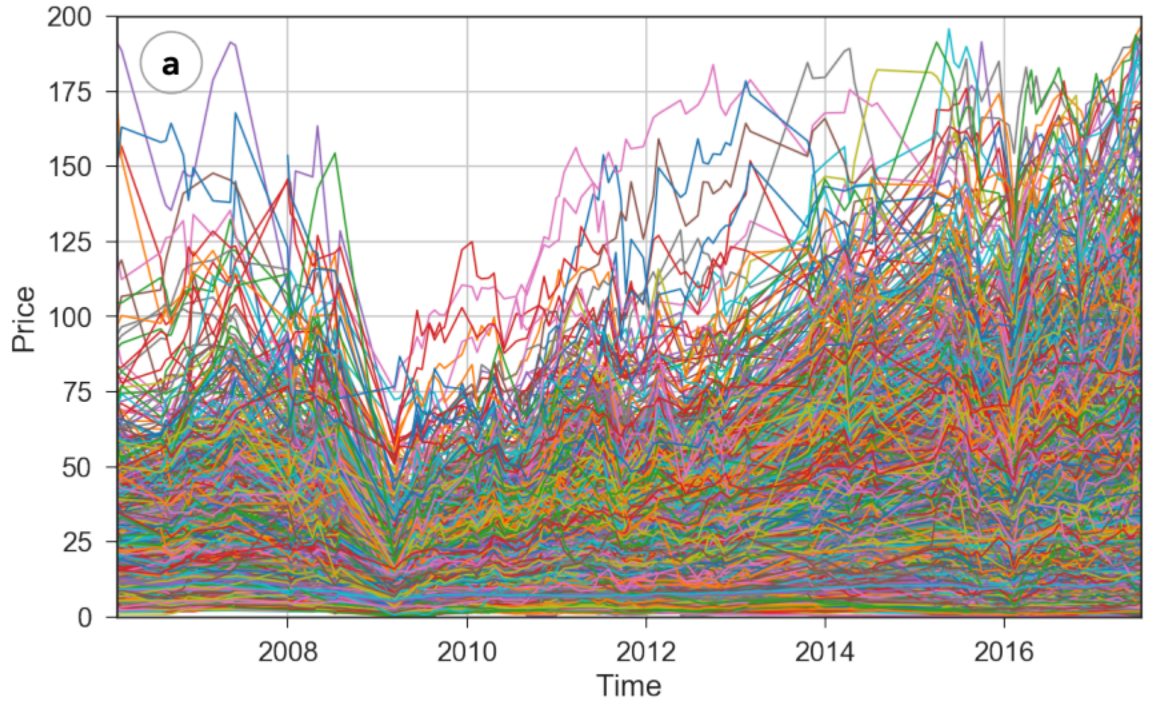
Scatterplot Matrix Techniques for Large N
[Carr et al. '87]

Example: Basketball Shot Chart

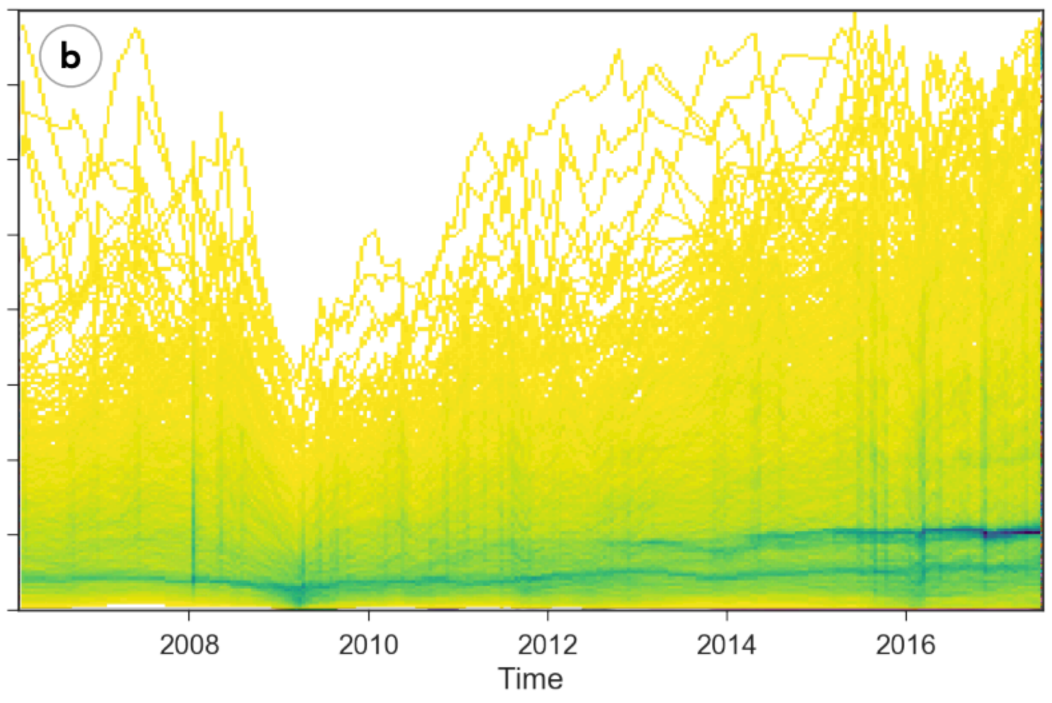


NBA Shooting 2011-12
[Goldsberry]

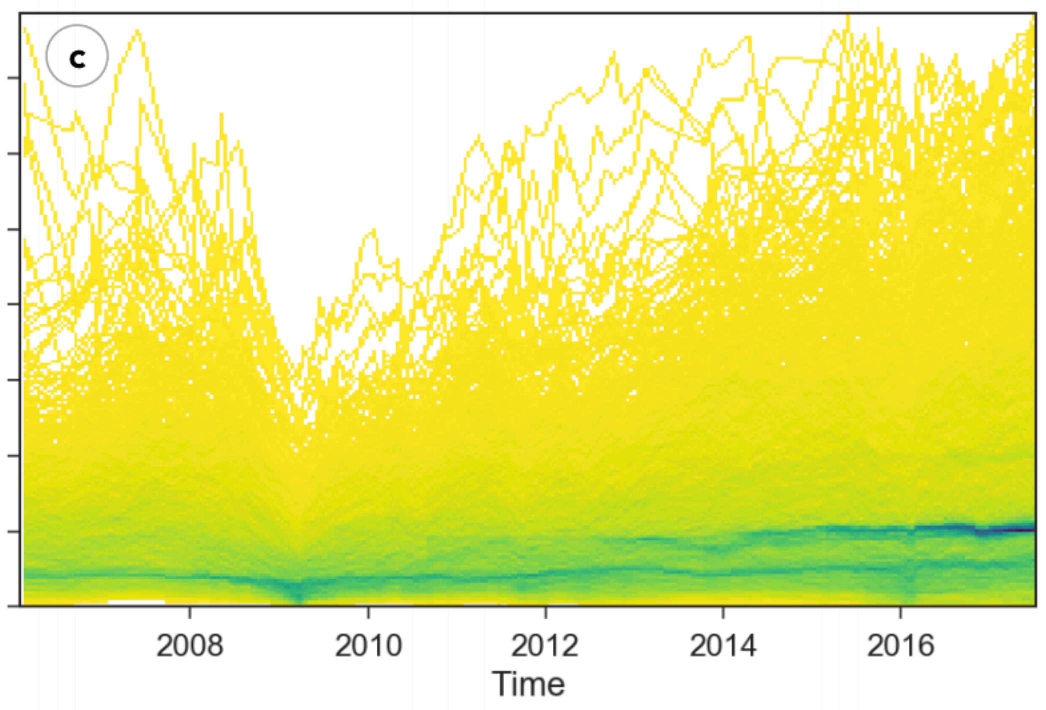
Example: Density Line Chart [Moritz & Fisher]



Line Chart



Non-Normalized Heatmap



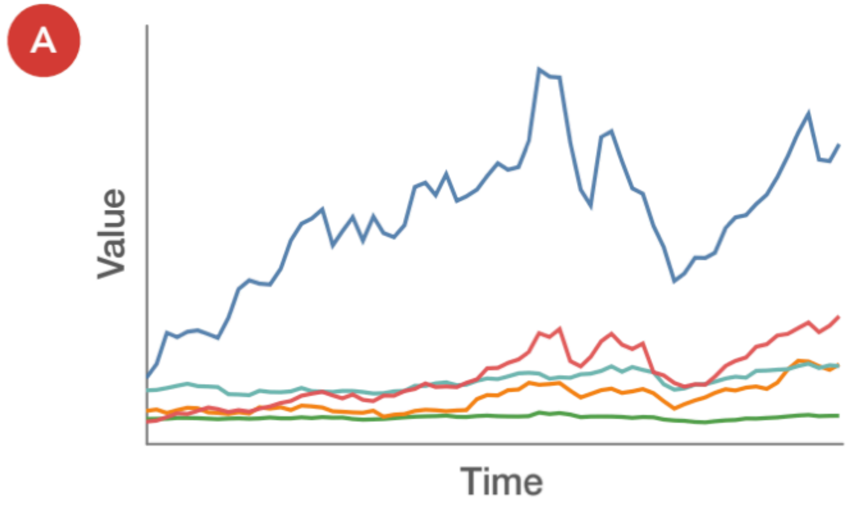
Normalized "DenseLines"

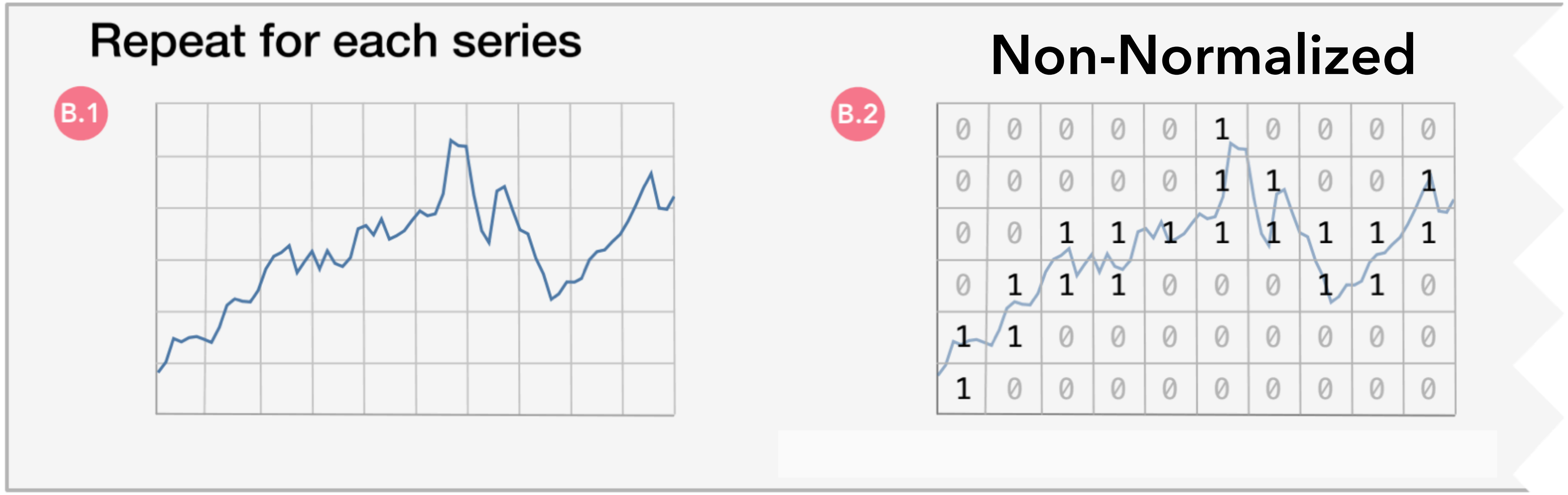
The non-normalized heatmap suffers from artifacts, seen as vertical stripes.

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.

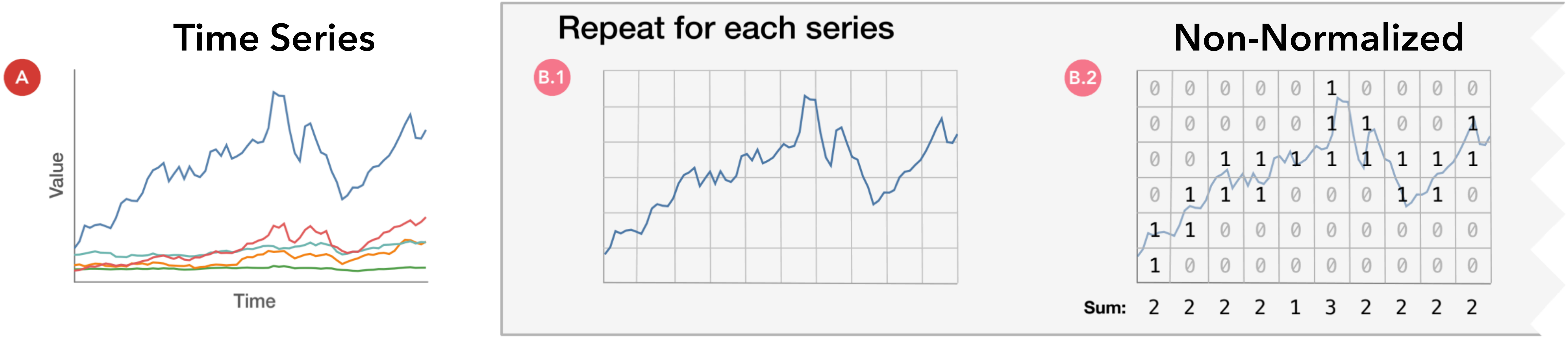
Example: Density Line Chart [Moritz & Fisher]

Time Series



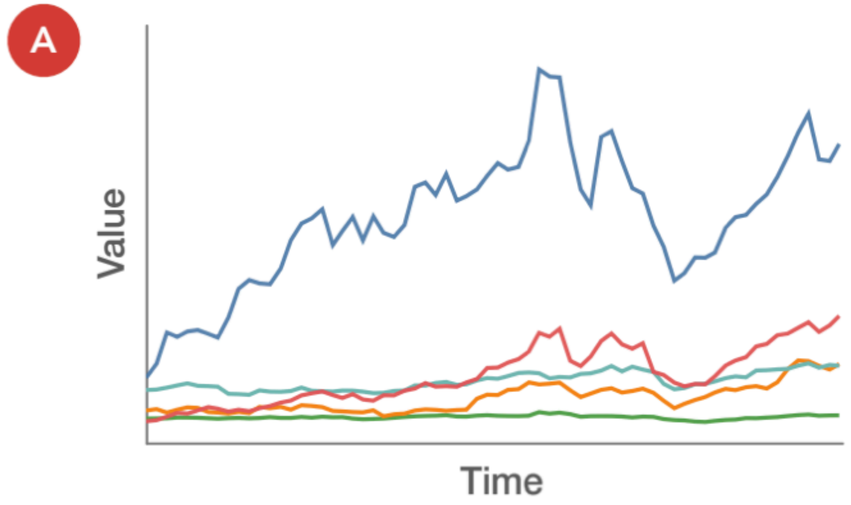


Example: Density Line Chart [Moritz & Fisher]

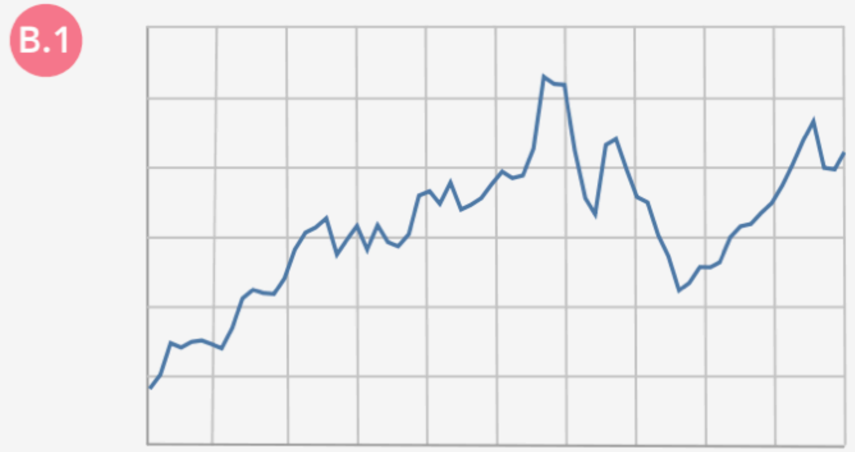


Example: Density Line Chart [Moritz & Fisher]

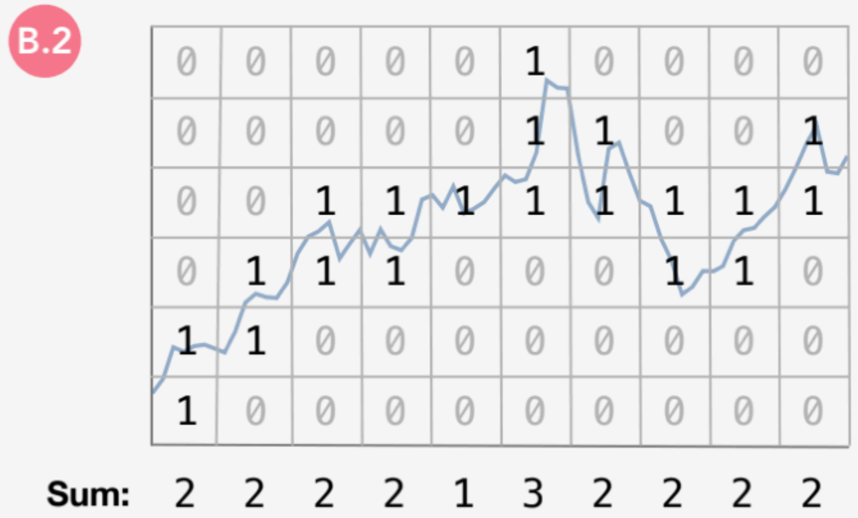
Time Series



Repeat for each series



Non-Normalized



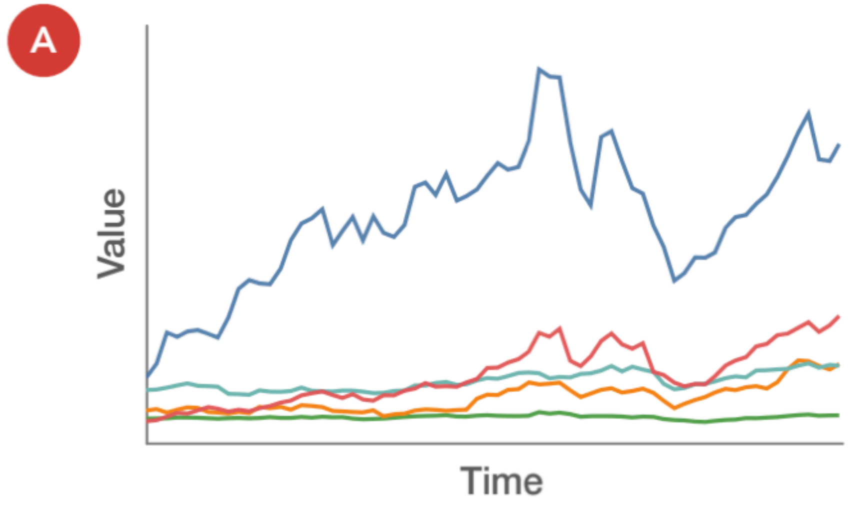
B.3

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

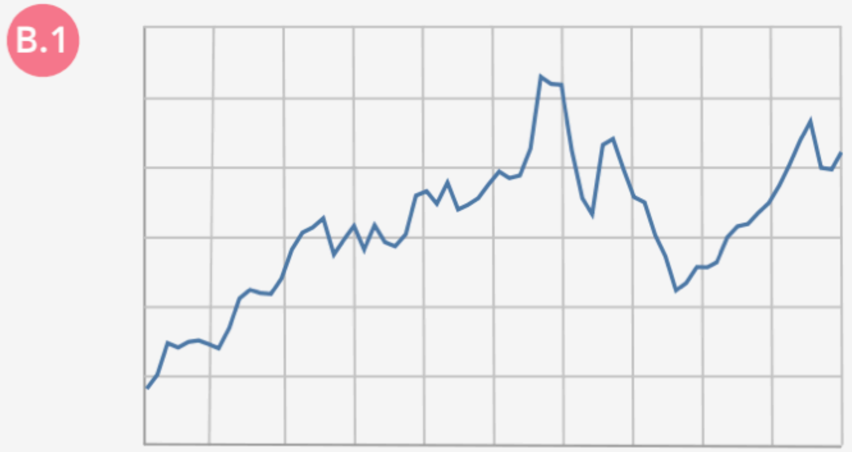
Approx. Arc-Length Normalized

Example: Density Line Chart [Moritz & Fisher]

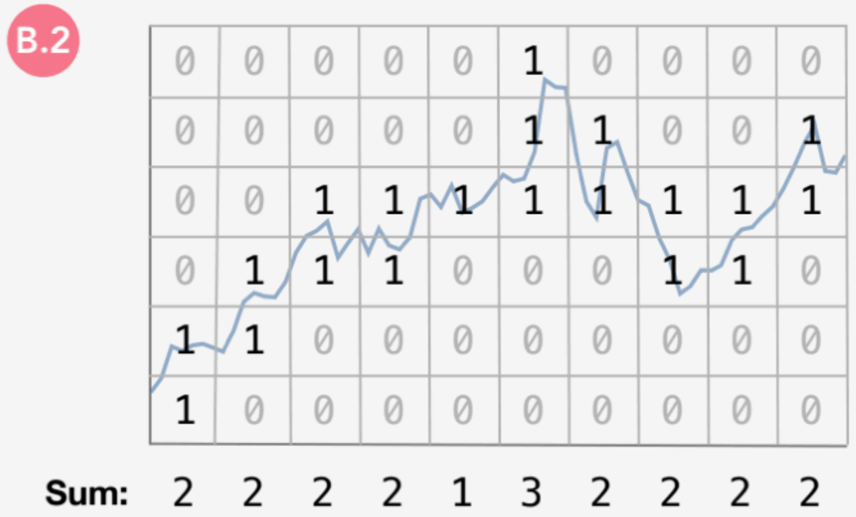
Time Series

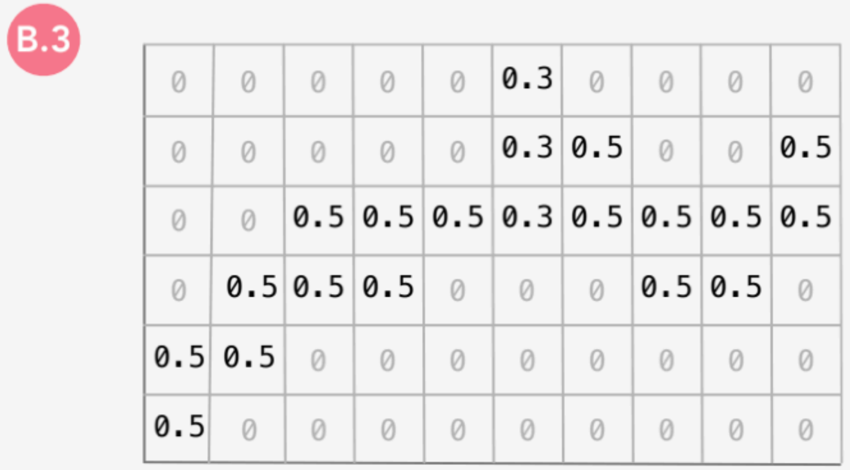


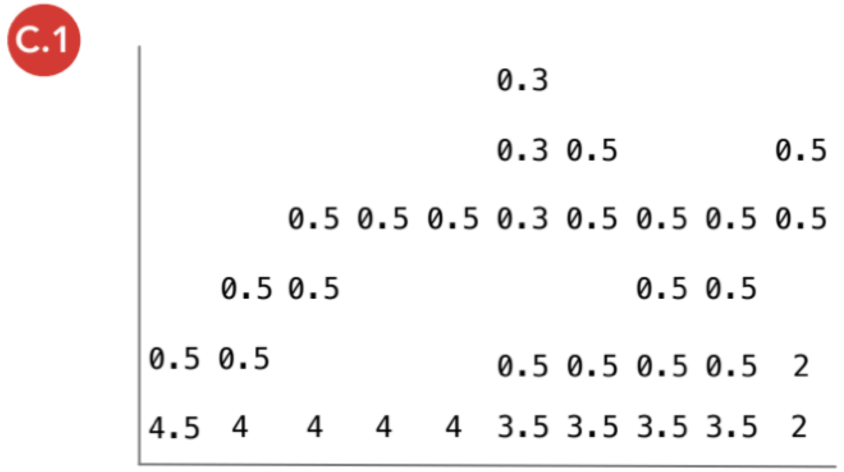
Repeat for each series

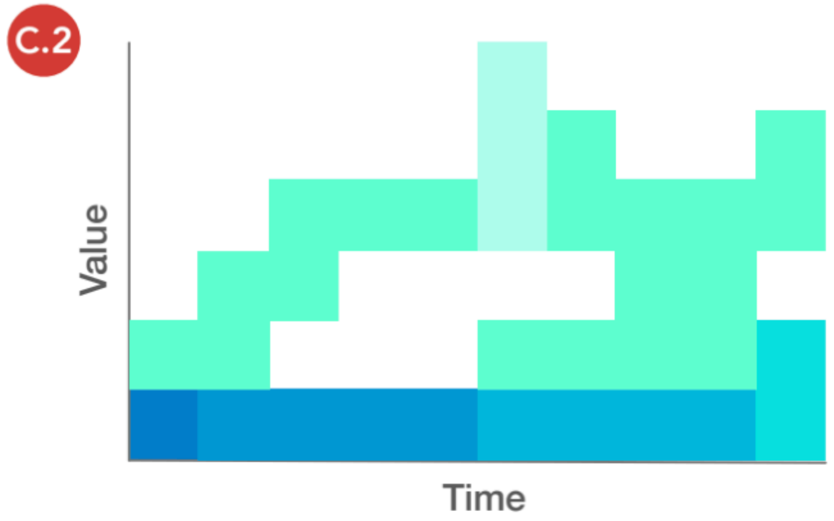


Non-Normalized







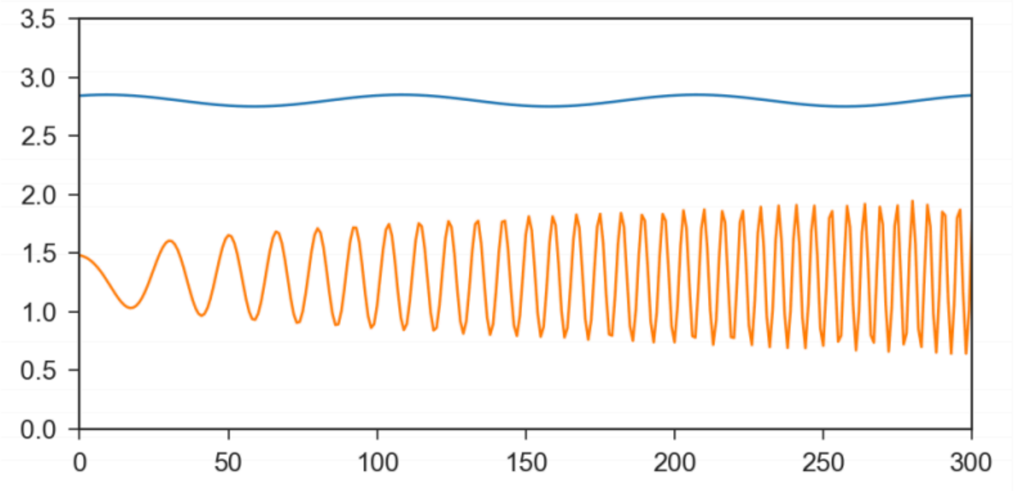


Approx. Arc-Length Normalized

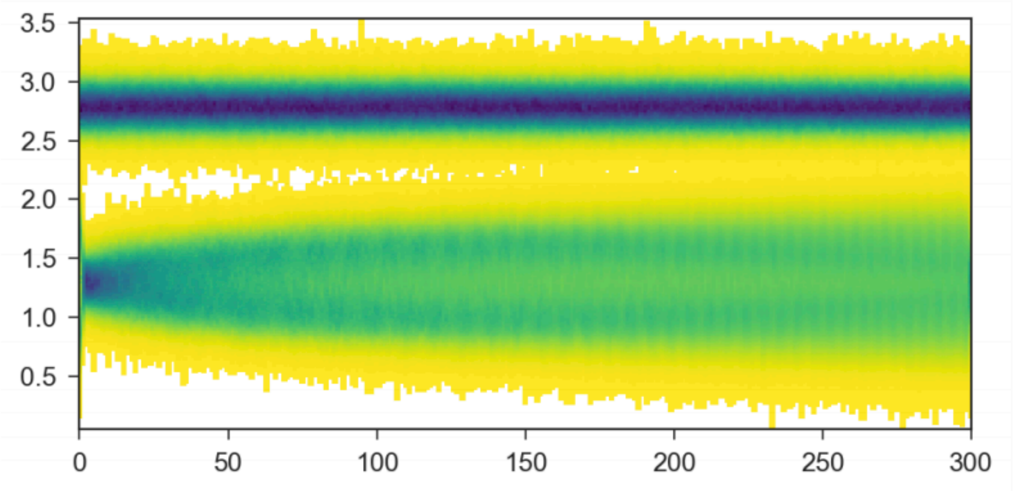
Aggregate

Color

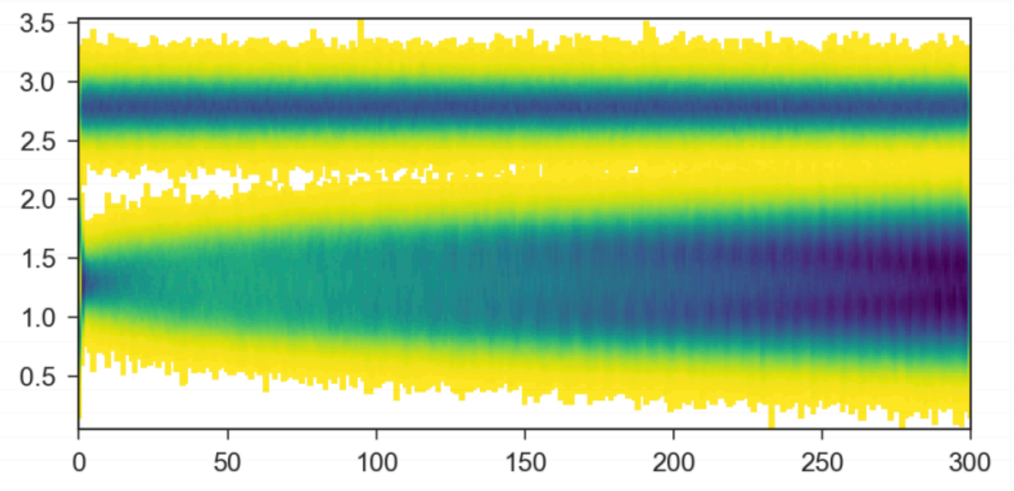
Example: Density Line Chart [Moritz & Fisher]



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.

2. Enabling Real-Time Interaction

Interactive Scalability Strategies

1. Query Database
2. Client-Side Indexing / Data Cubes
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. Client-Side Indexing / Data Cubes

3. Prefetching

4. Approximation

Interactive Scalability Strategies

1. **Query Database**
2. **Client-Side Indexing / Data Cubes** Query data summaries

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

3. **Prefetching**
4. **Approximation**

Interactive Scalability Strategies

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

These strategies are **not** mutually exclusive!

Systems can apply them in tandem.

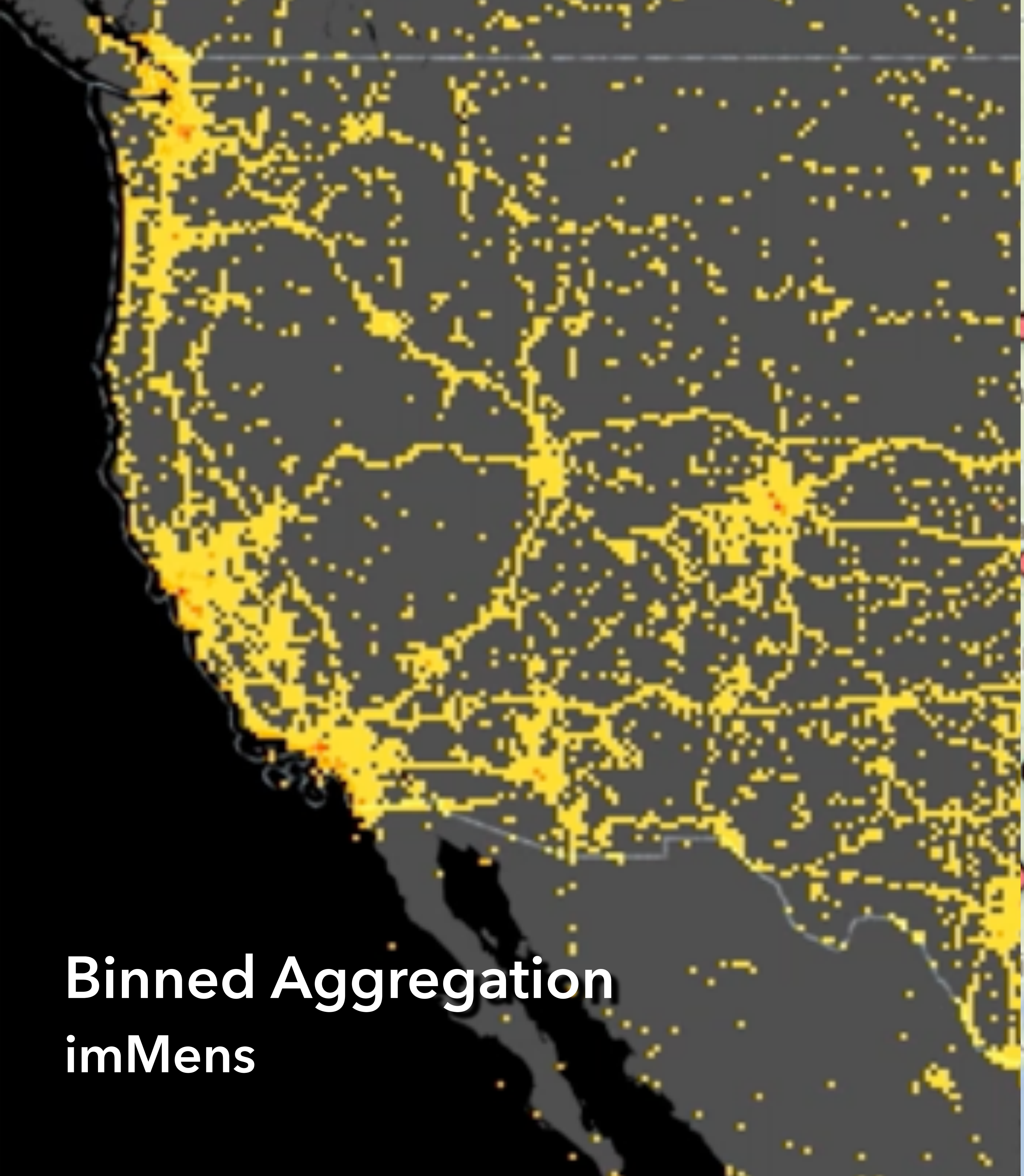
imMens

[Liu, Jiang & Heer '13]

Strategies: Client-Side Data Cubes

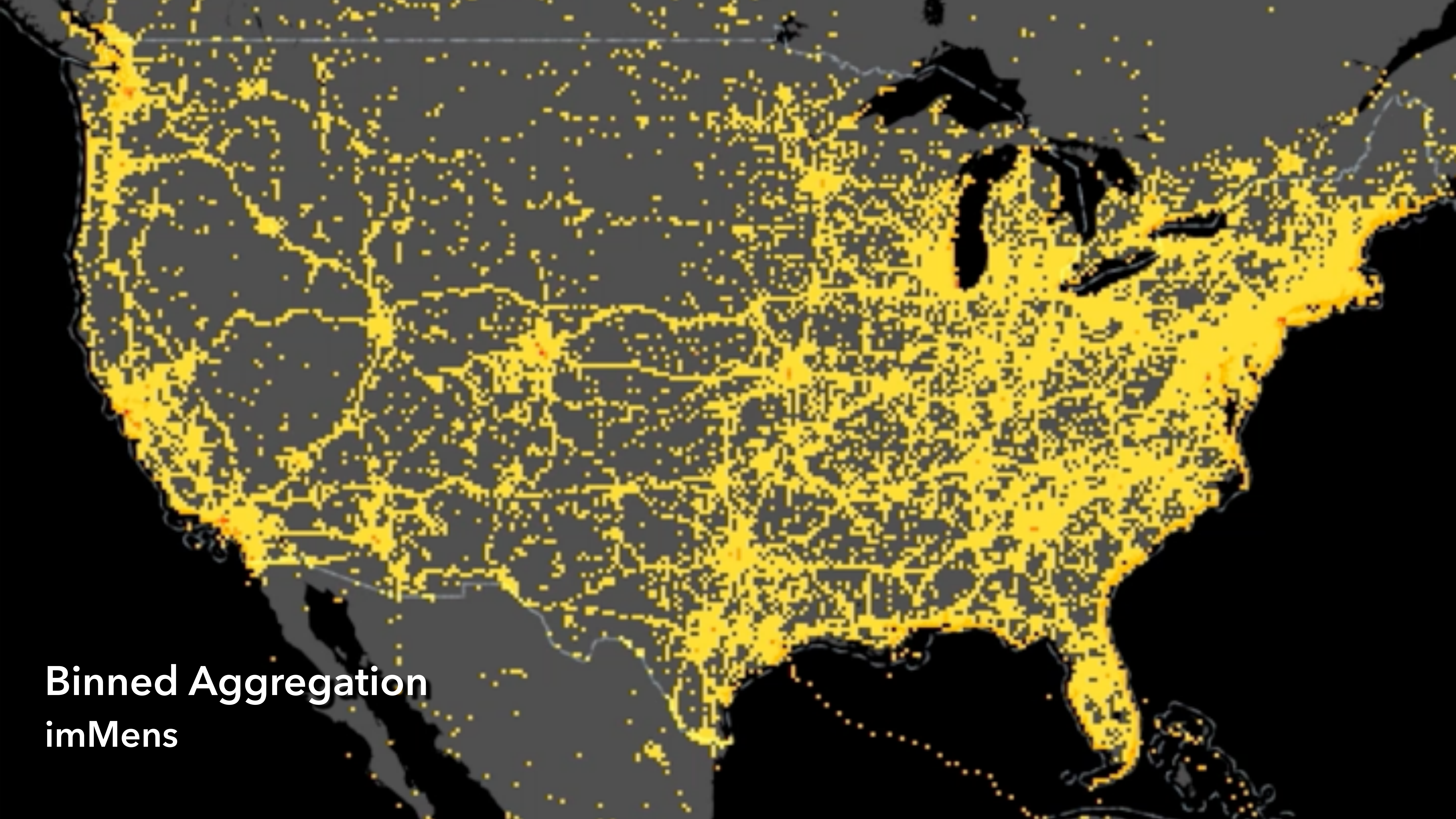
Sampling

Google Fusion Tables



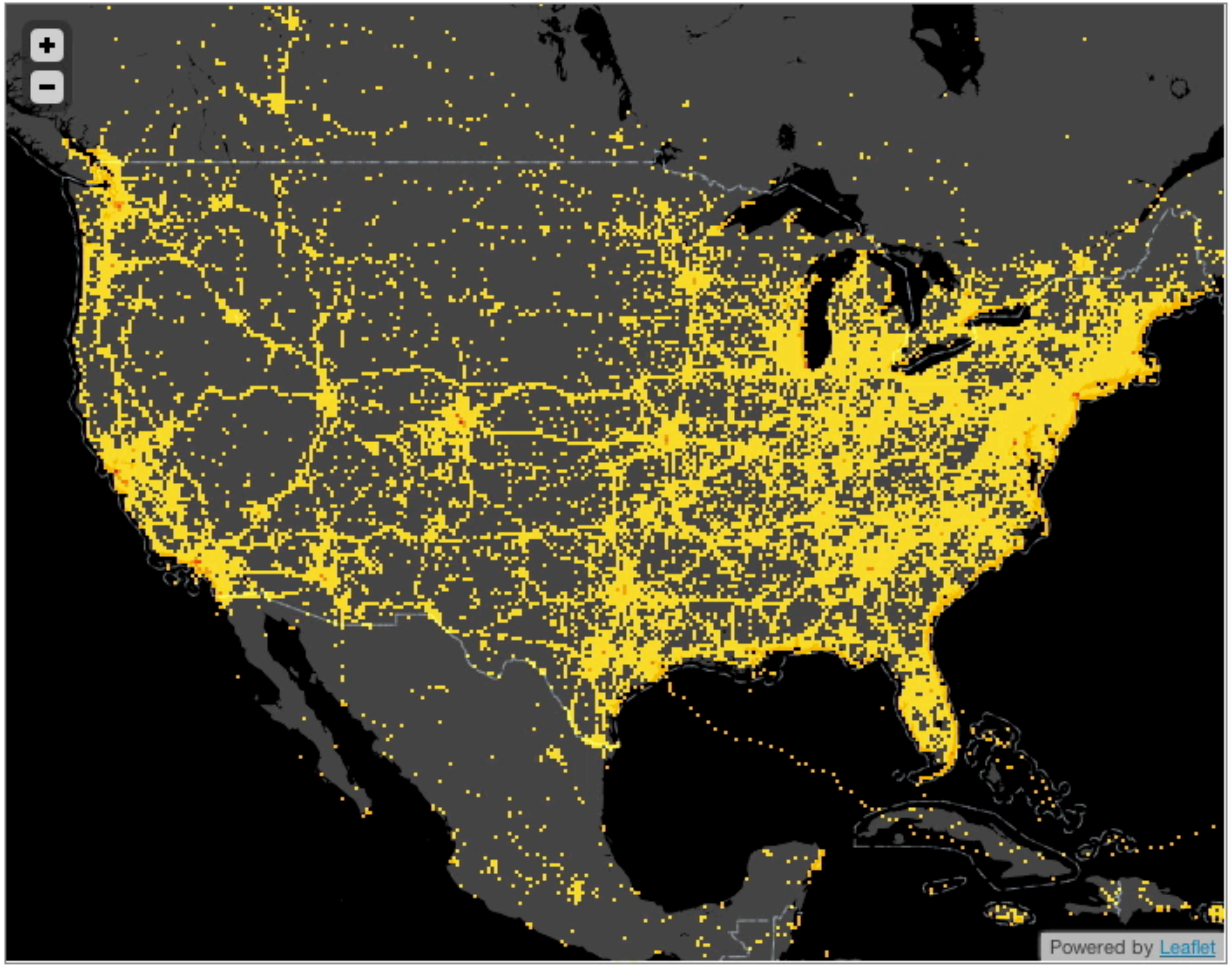
Binned Aggregation
imMens

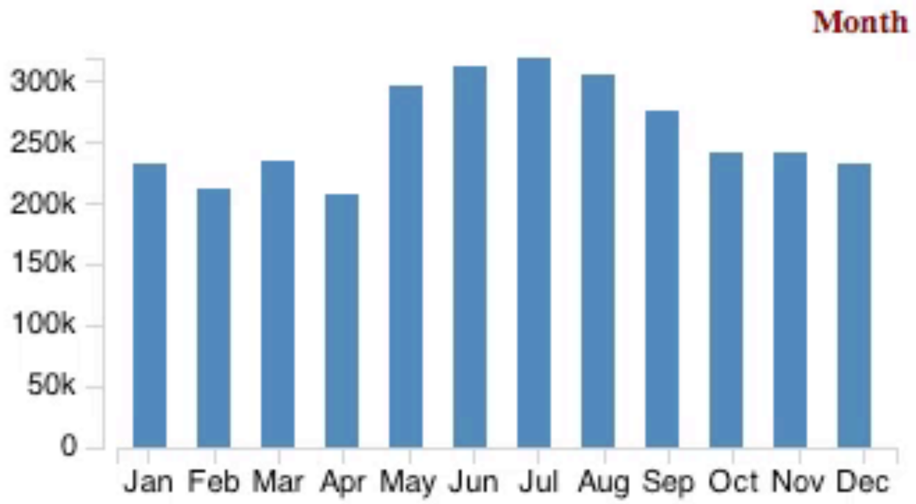
Sampling
Google Fusion Tables

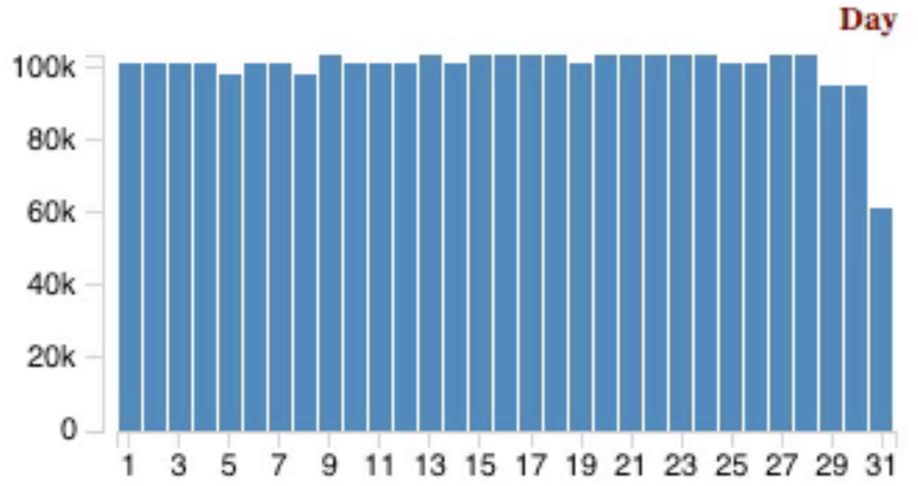


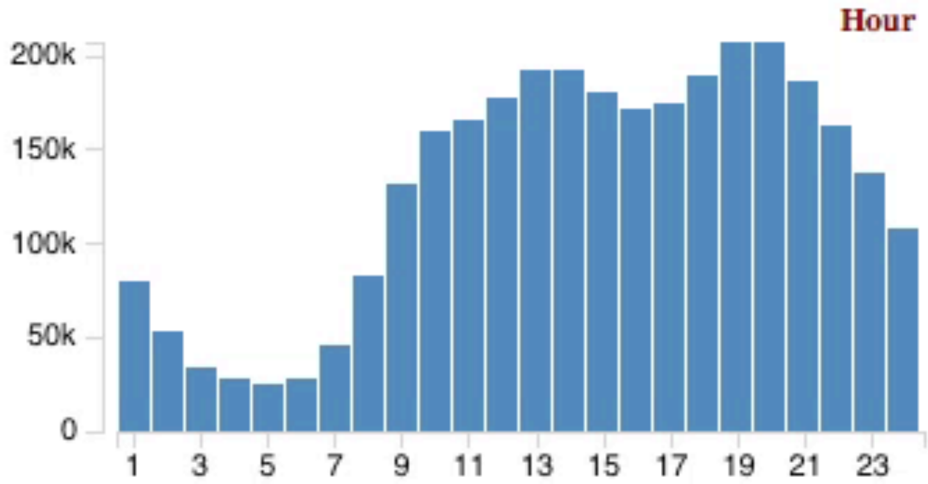
Binned Aggregation

imMens


Powered by Leaflet


Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

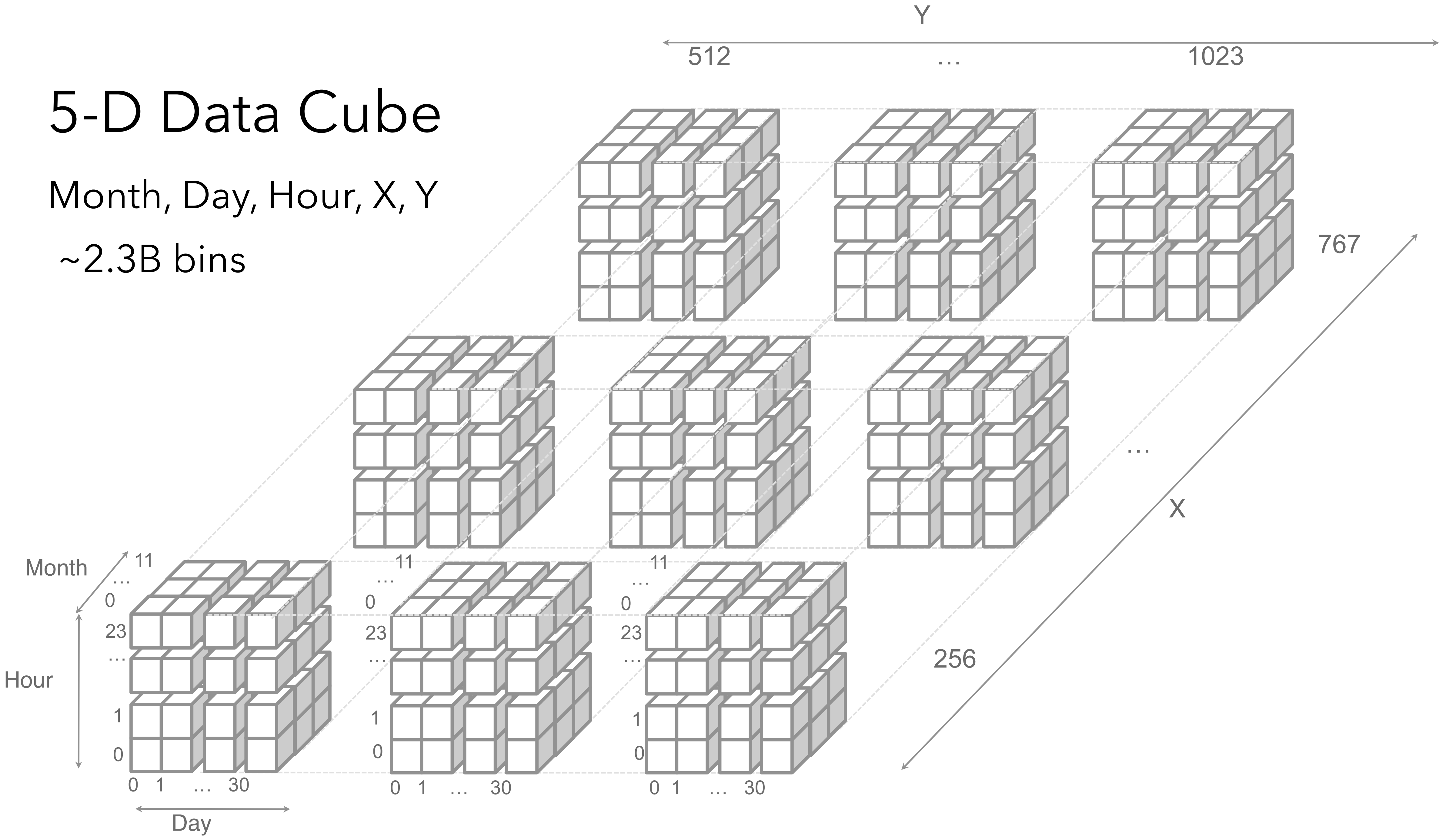

Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31


Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23

5-D Data Cube

Month, Day, Hour, X, Y

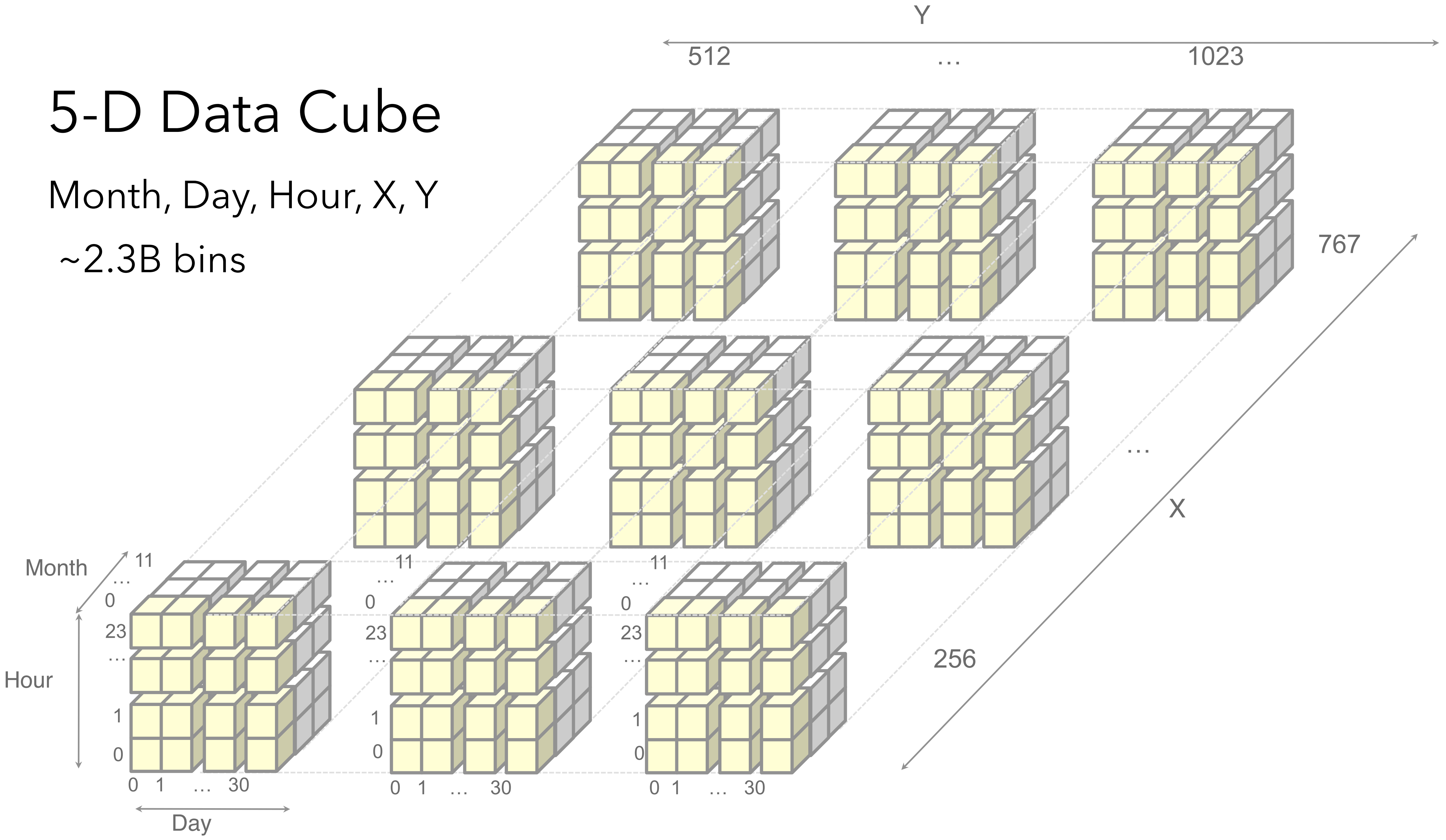
~2.3B bins

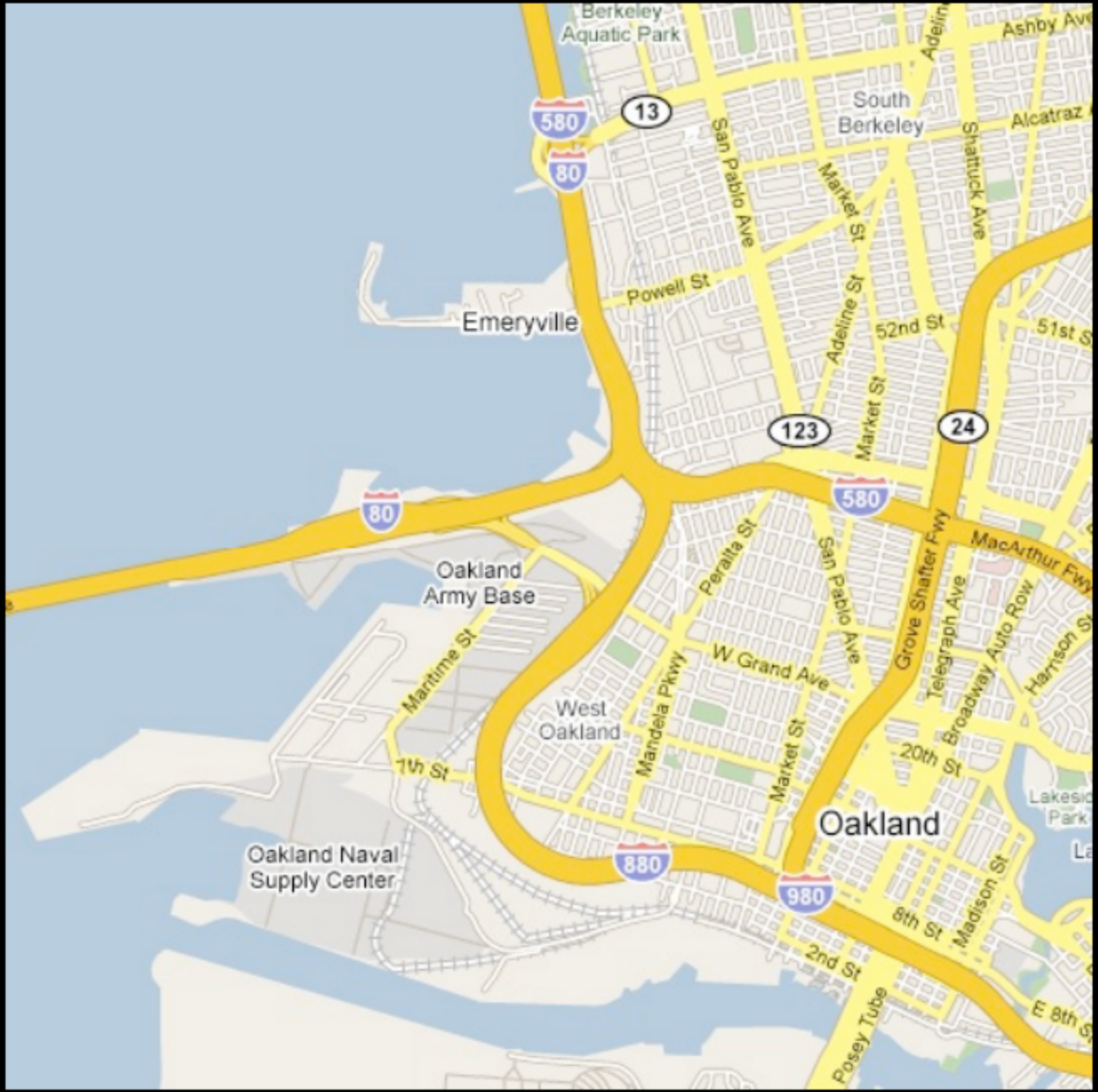


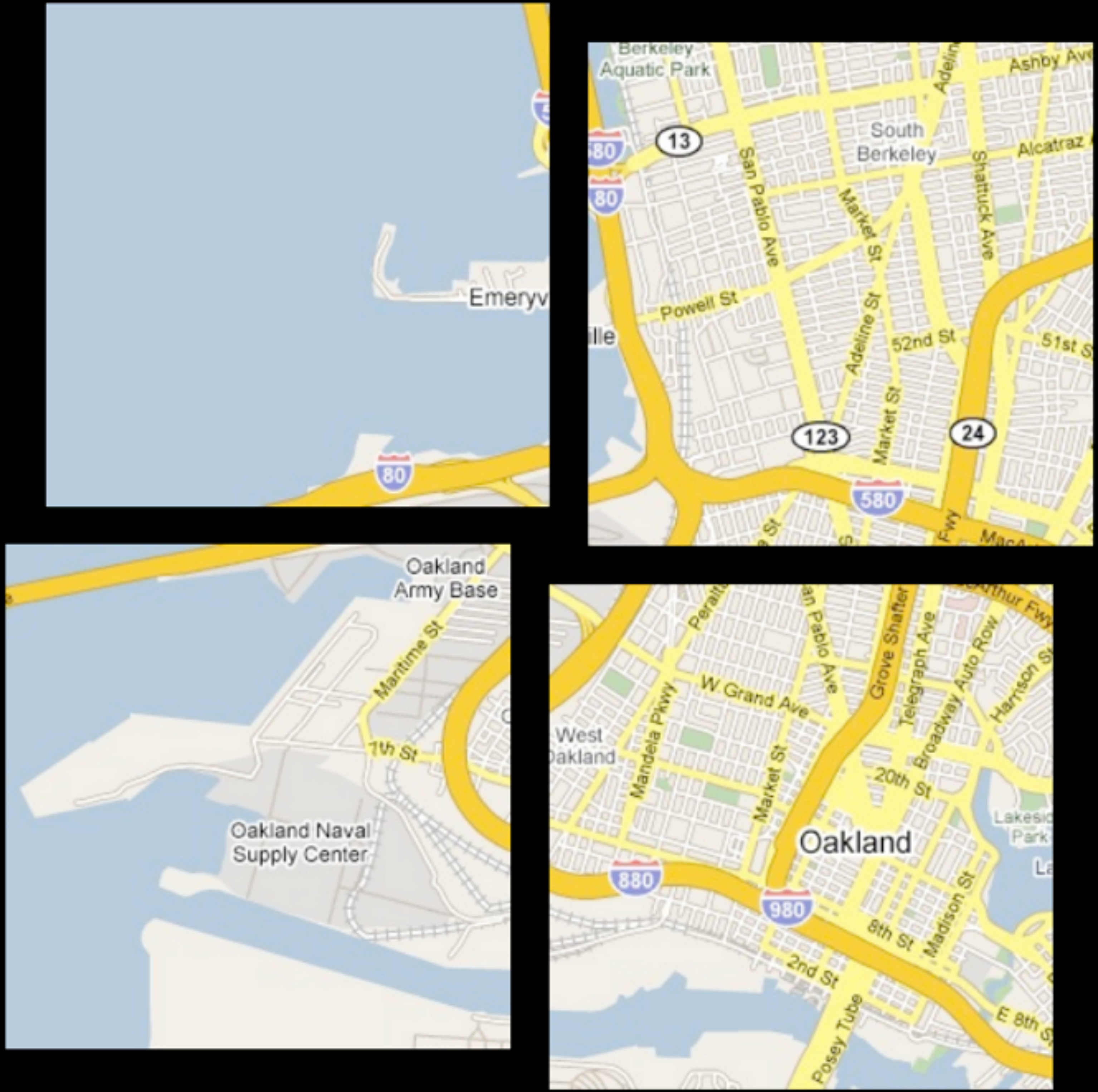
5-D Data Cube

Month, Day, Hour, X, Y

~2.3B bins



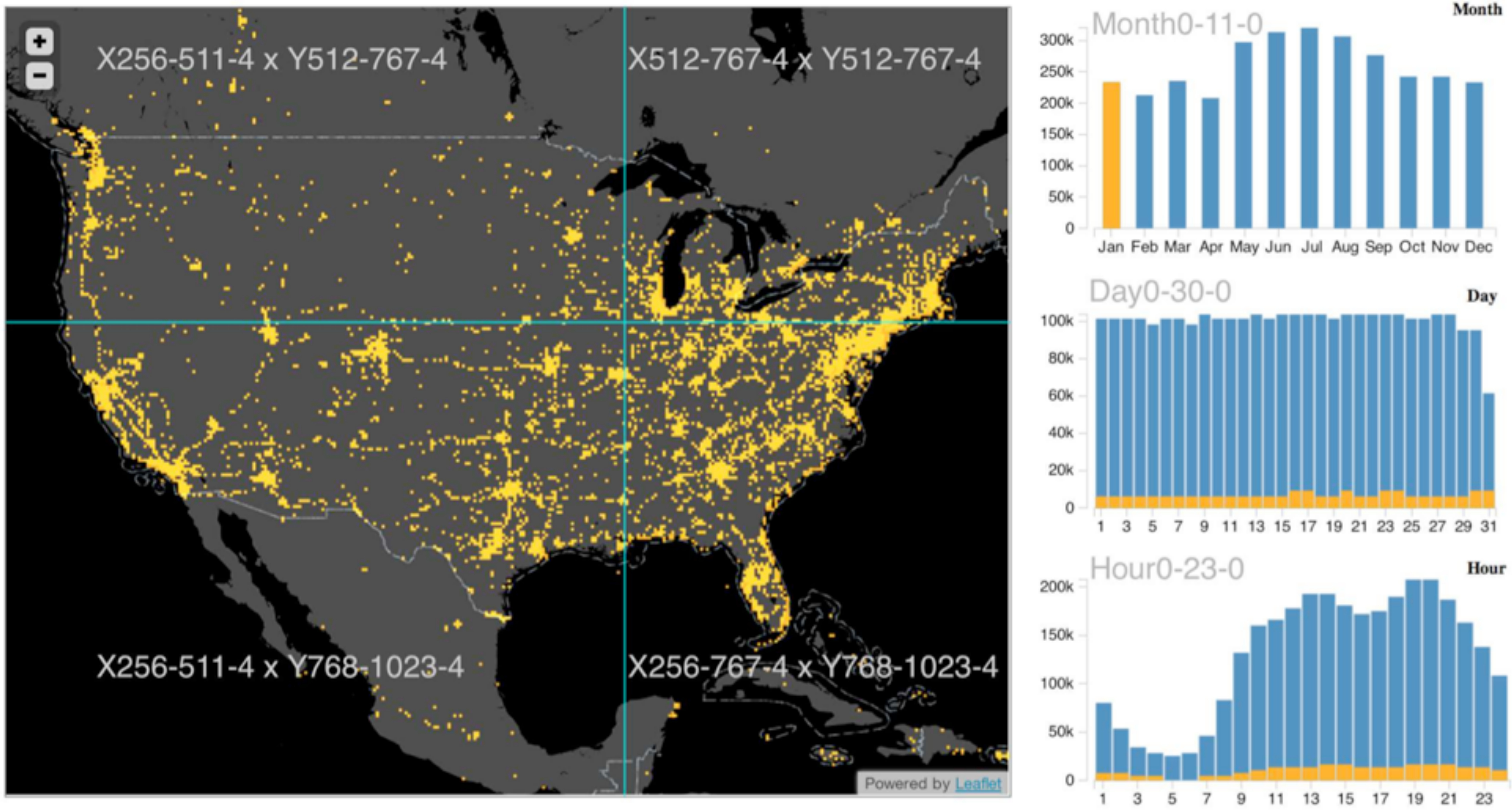

Berkeley
Aquatic Park
580
13
80
South
Berkeley
Ashby Ave
San Pablo Ave
Market St
Shattuck Ave
Powell St
Emeryville
Adeline St
52nd St
Market St
123
24
80
580
MacArthur Fwy
Oakland
Army Base
Peralta St
San Pablo Ave
Grove Shafter Fwy
Telegraph Ave
Broadway Auto Row
W Grand Ave
Maritime St
West
Oakland
Mandela Pkwy
Market St
20th St
7th St
Oakland
Oakland Naval
Supply Center
880
980
8th St
Madison St
2nd St
Posey Tube

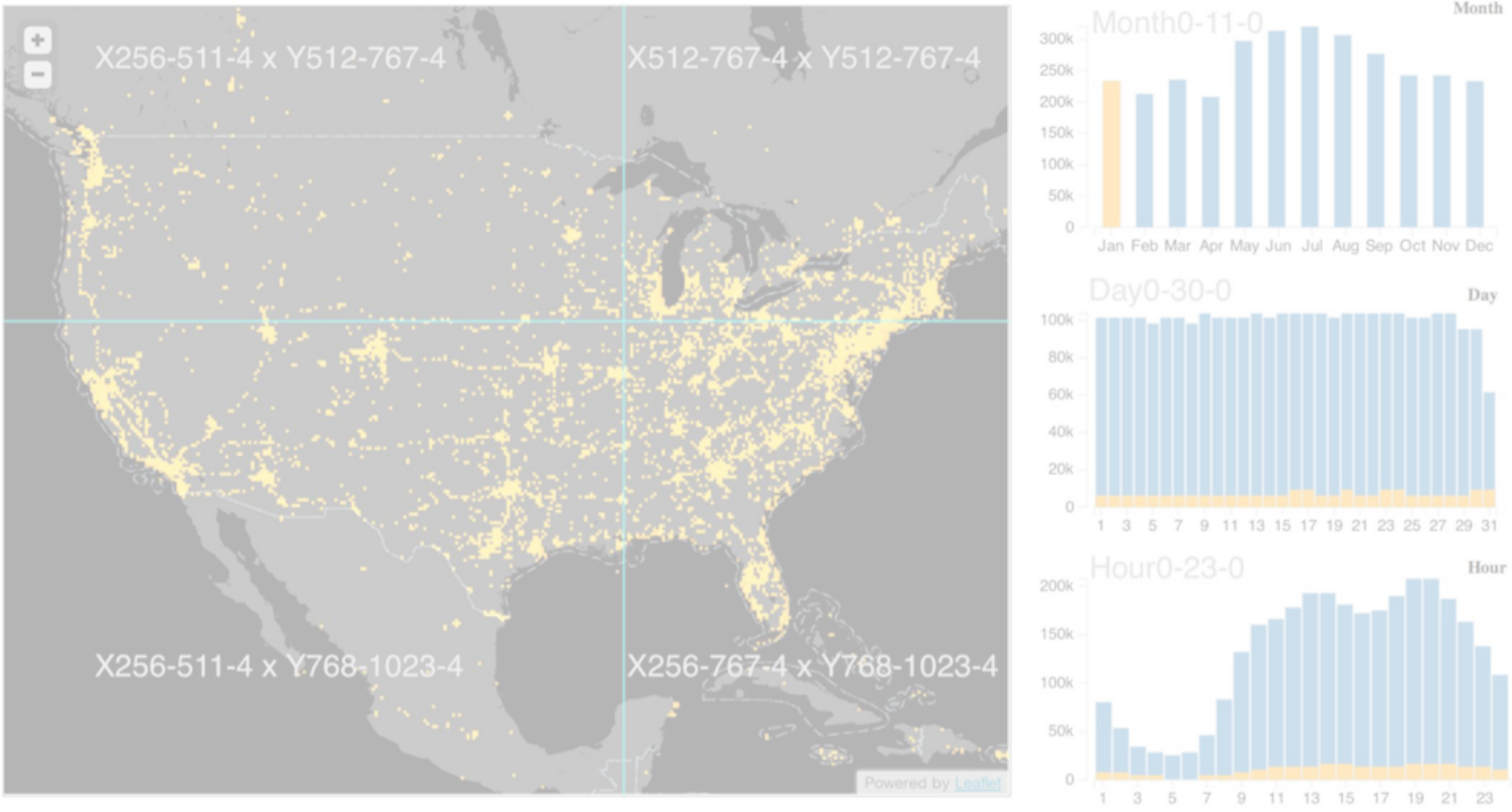


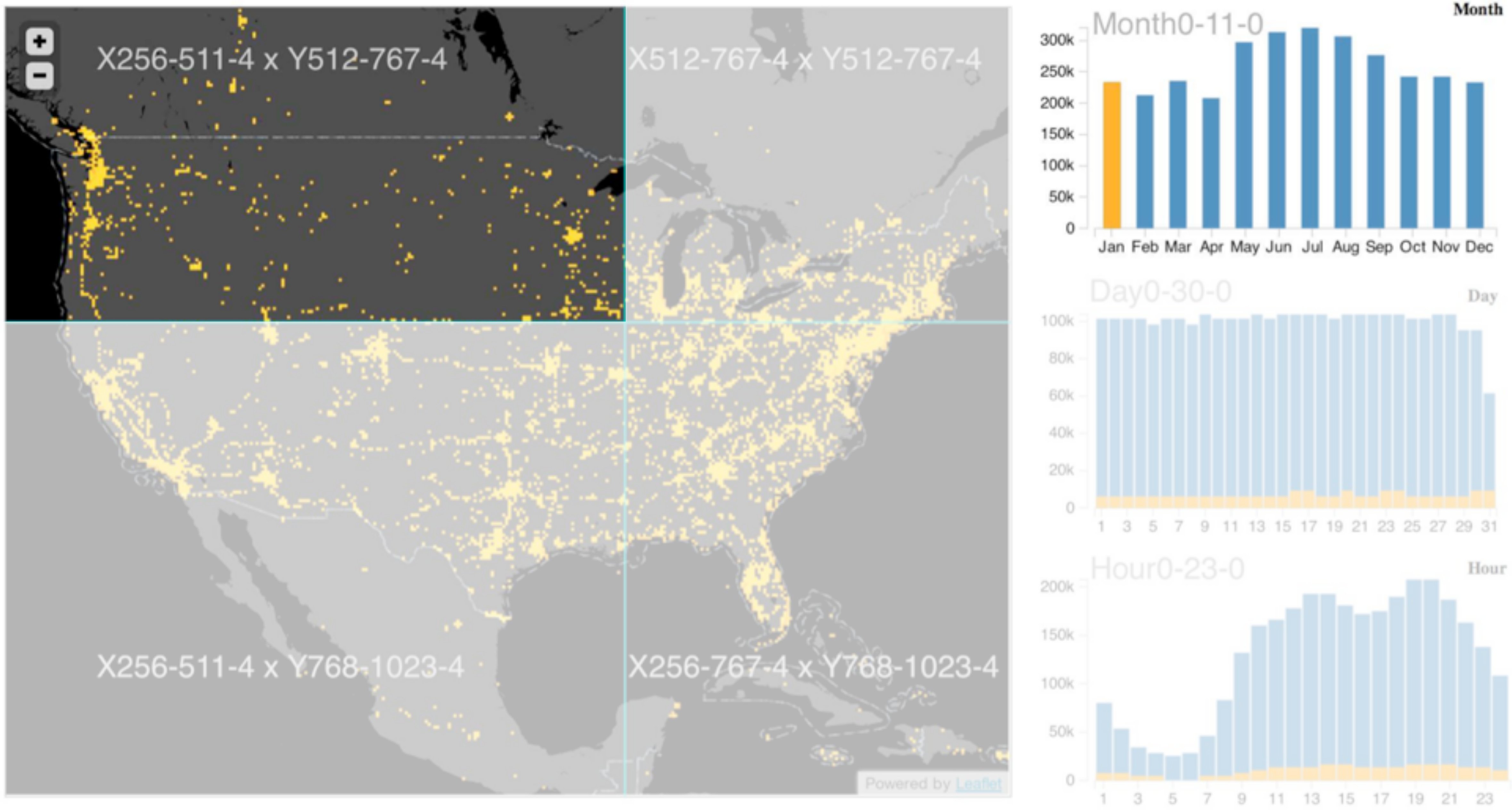
Aquatic Park
South Berkeley
13
San Pablo Ave
Market St
Shattuck Ave
Powell St
Adeline St
52nd St
123
24
580
80
Oakland Army Base
Maritime St
Oakland Naval Supply Center
W Grand Ave
Mandela Pkwy
Grove Shafter
Telegraph Ave
Broadway Auto Row
20th St
Oakland
880
980
8th St
Madison St
2nd St
Posey Tube

Multivariate Data Tiles

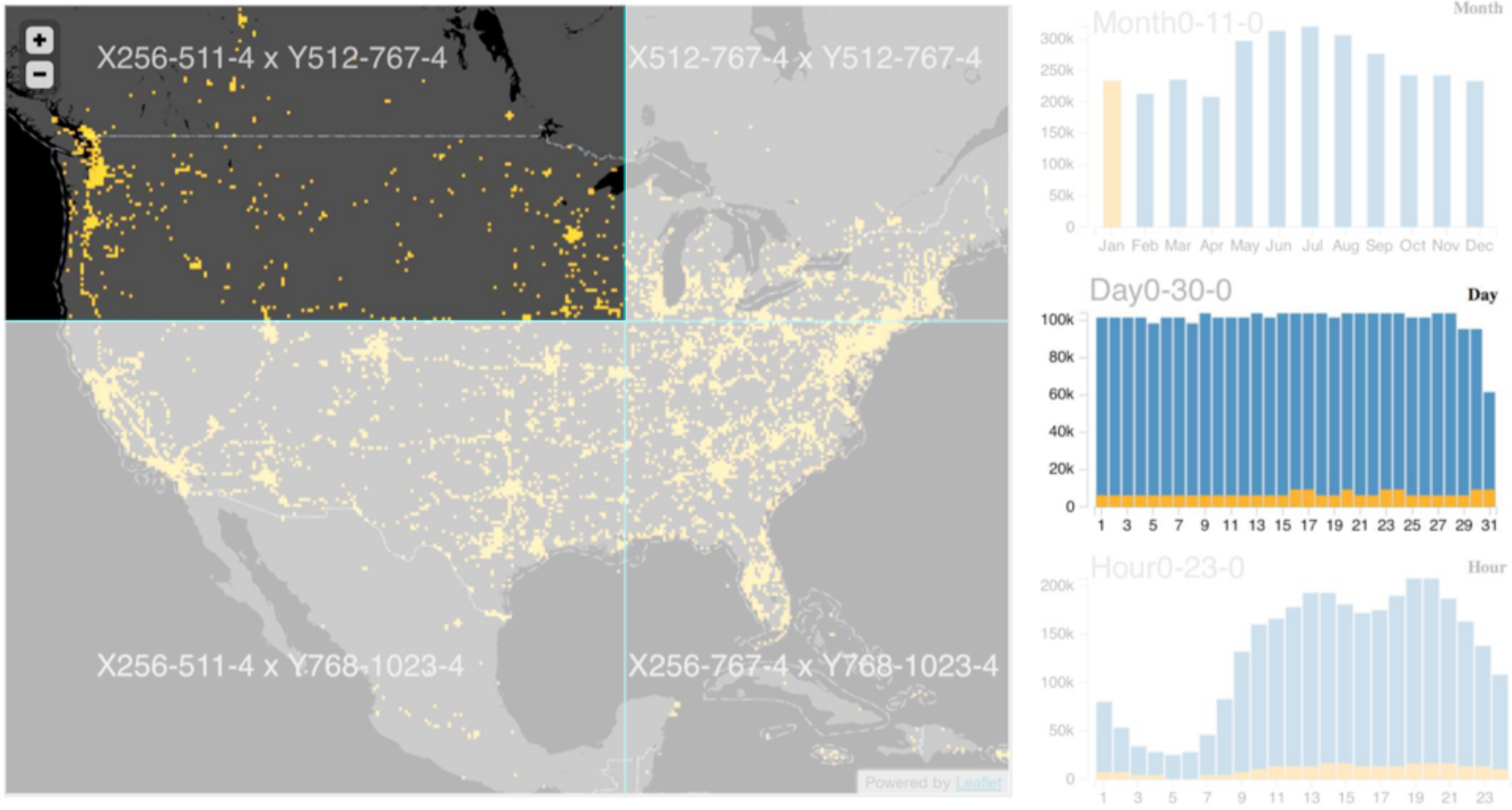
1. Send data, not pixels
2. Embed multi-dim data

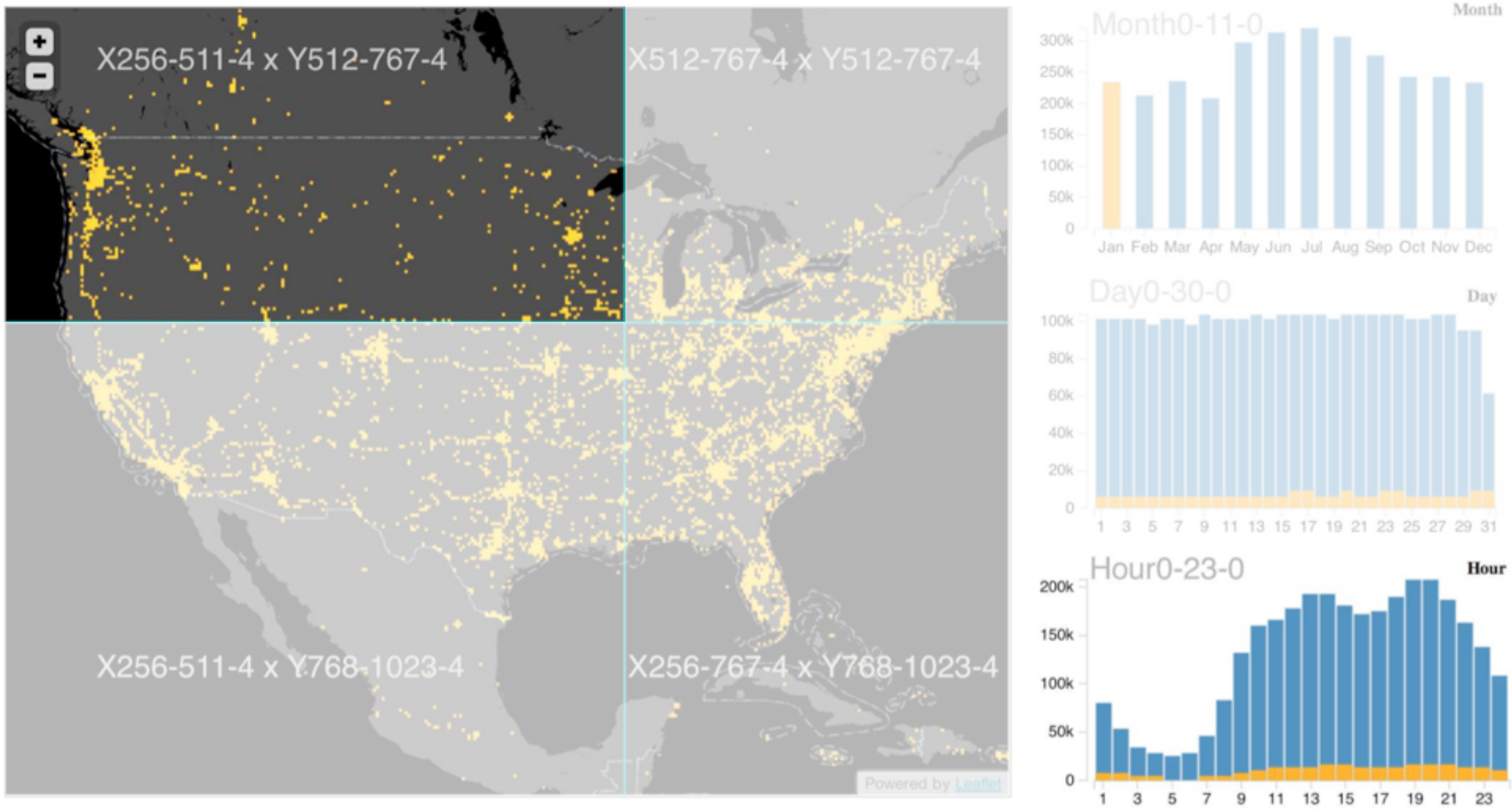

X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23

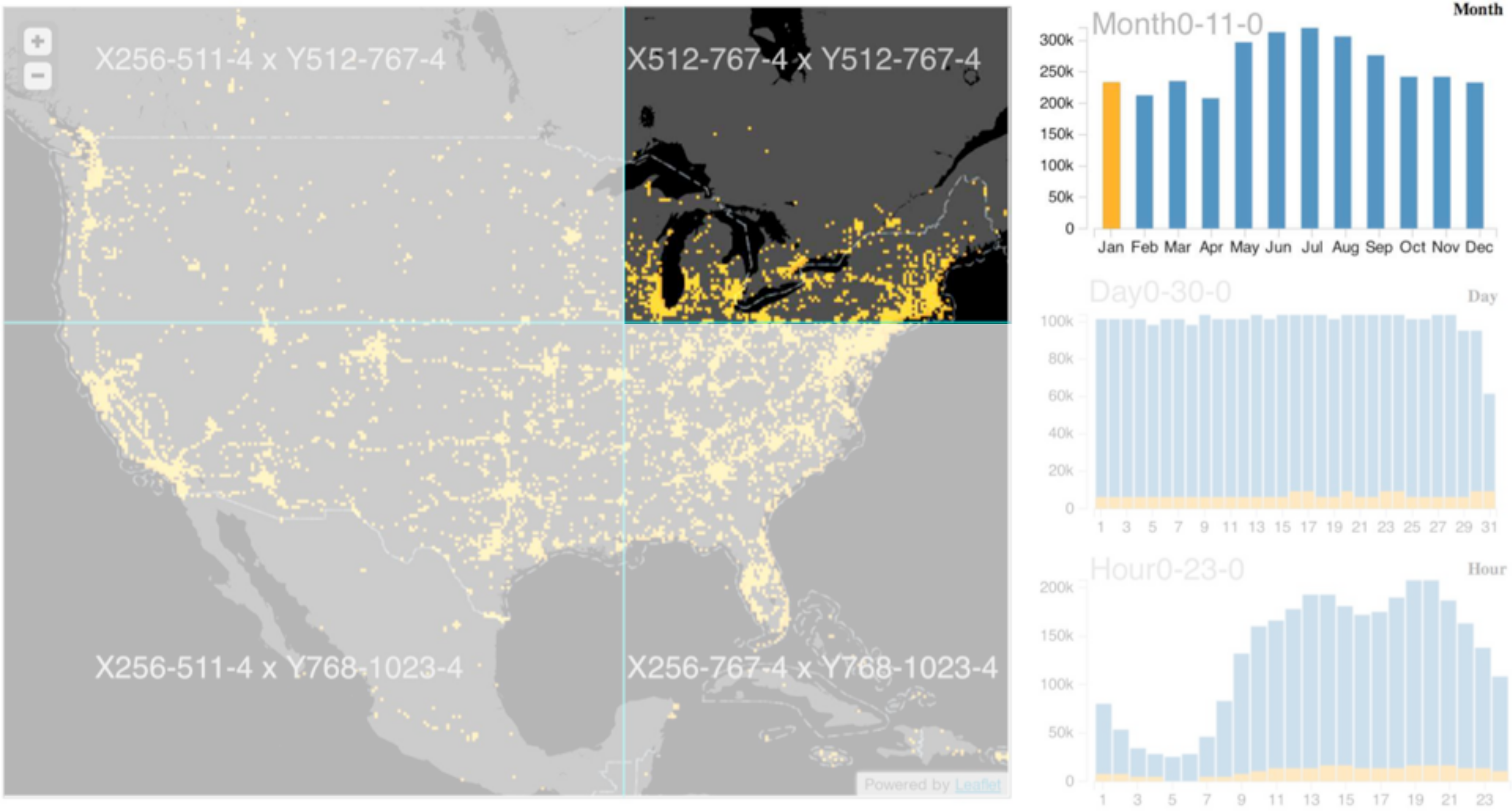

X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
300k 250k 200k 150k 100k 50k 0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
100k 80k 60k 40k 20k 0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
200k 150k 100k 50k 0
1 3 5 7 9 11 13 15 17 19 21 23



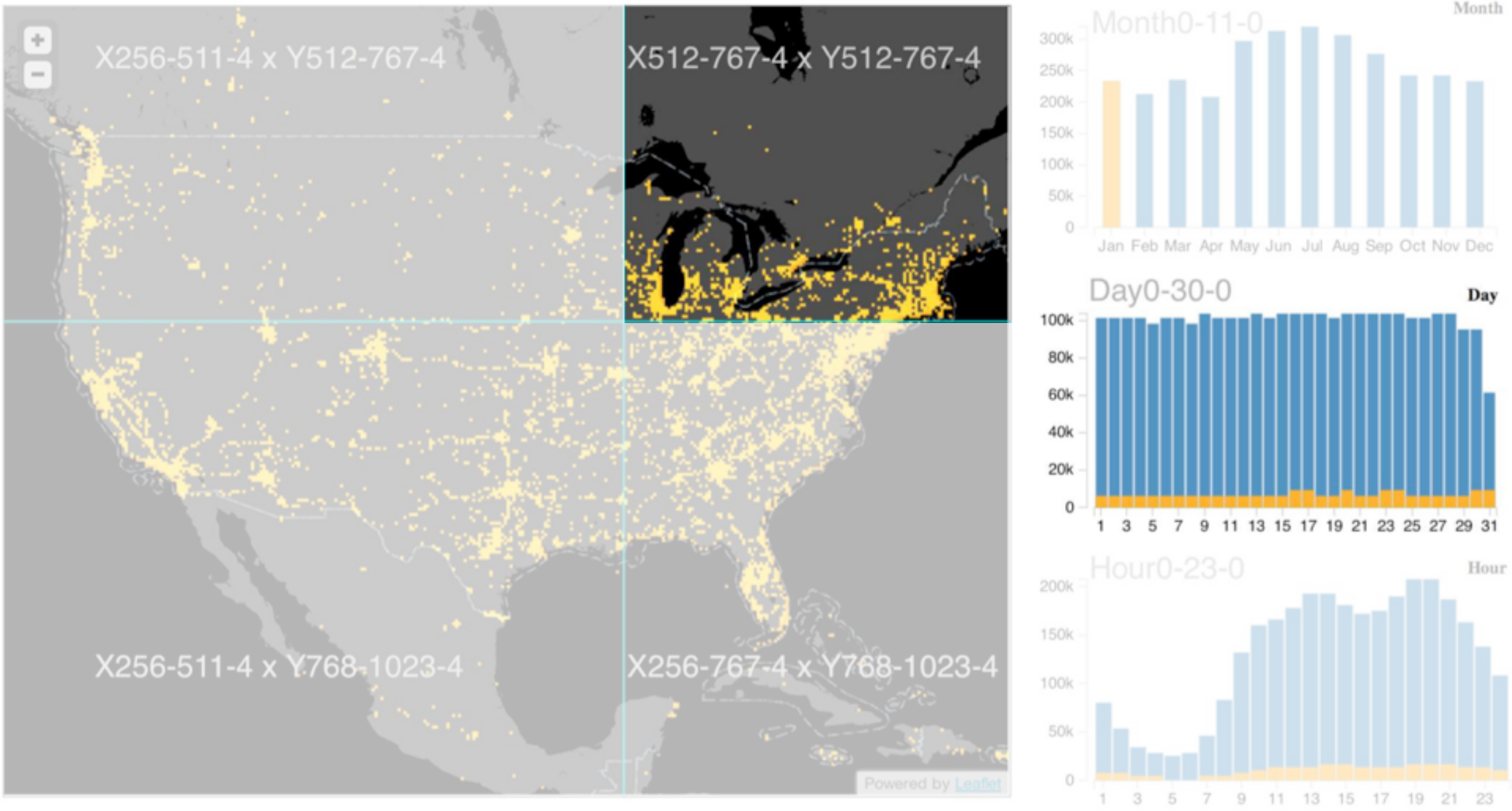
X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23

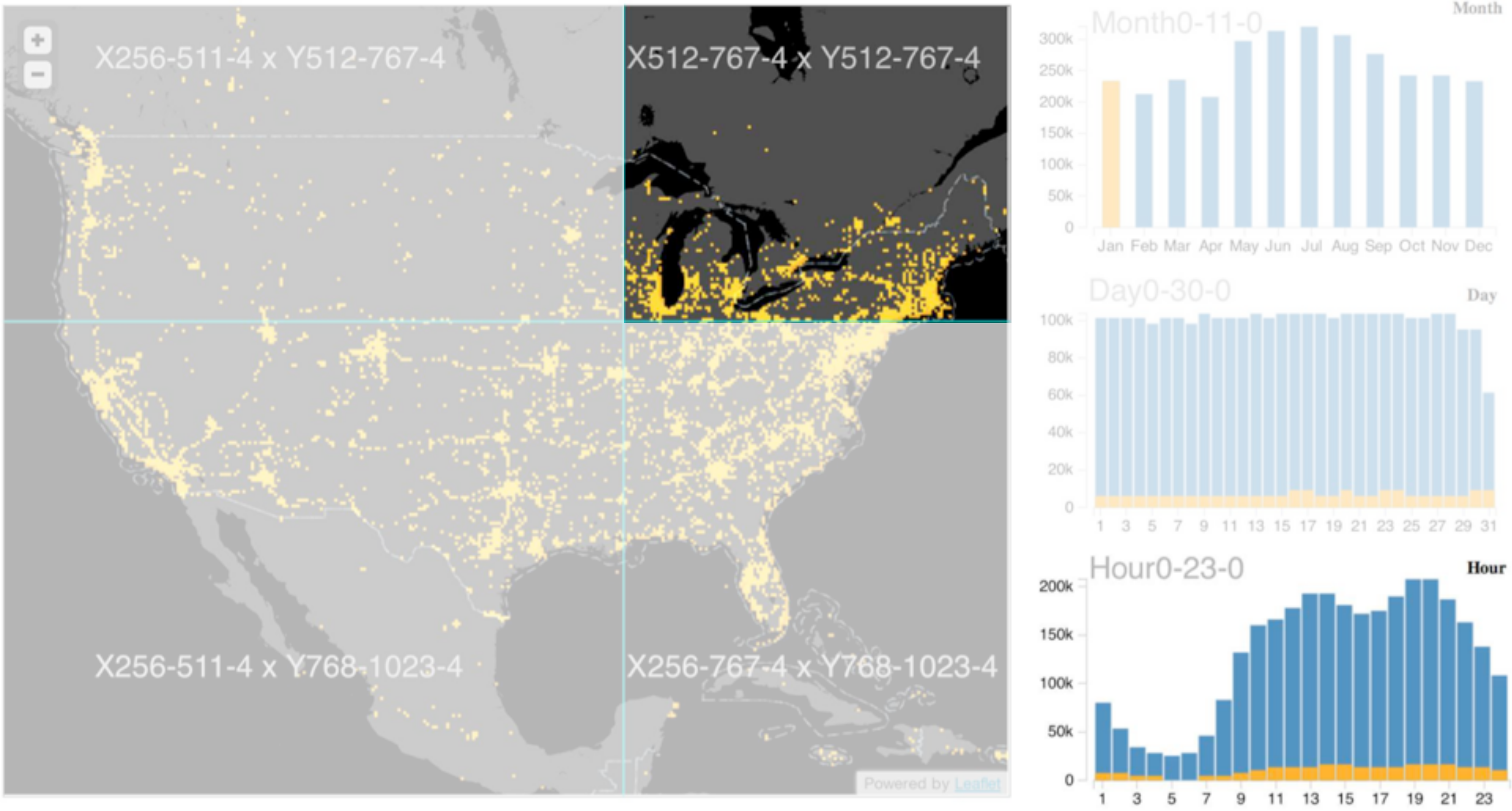

X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
0 50k 100k 150k 200k 250k 300k
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
0 20k 40k 60k 80k 100k
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
0 50k 100k 150k 200k
1 3 5 7 9 11 13 15 17 19 21 23

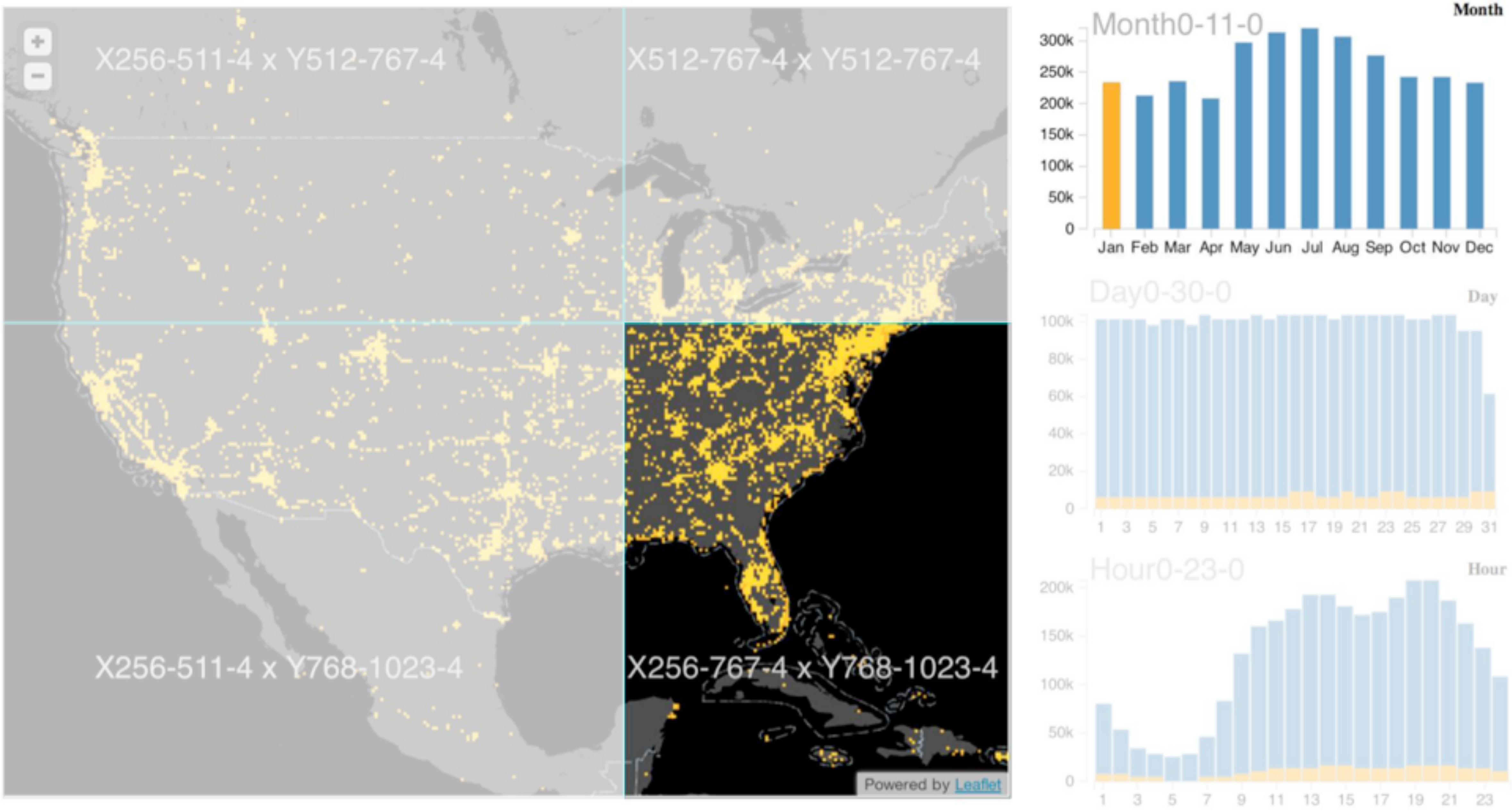


X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23

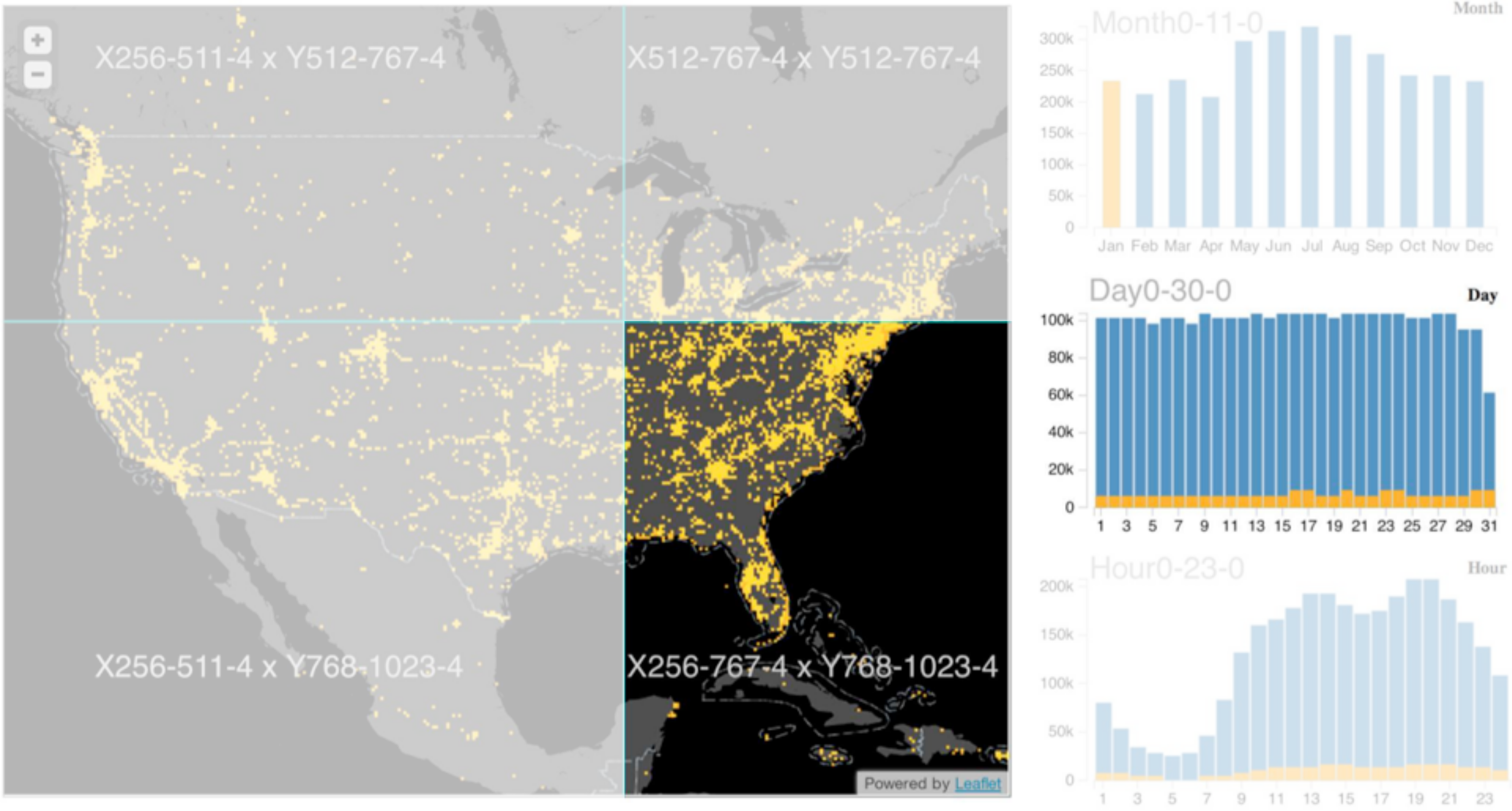

X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
Hour0-23-0
Hour

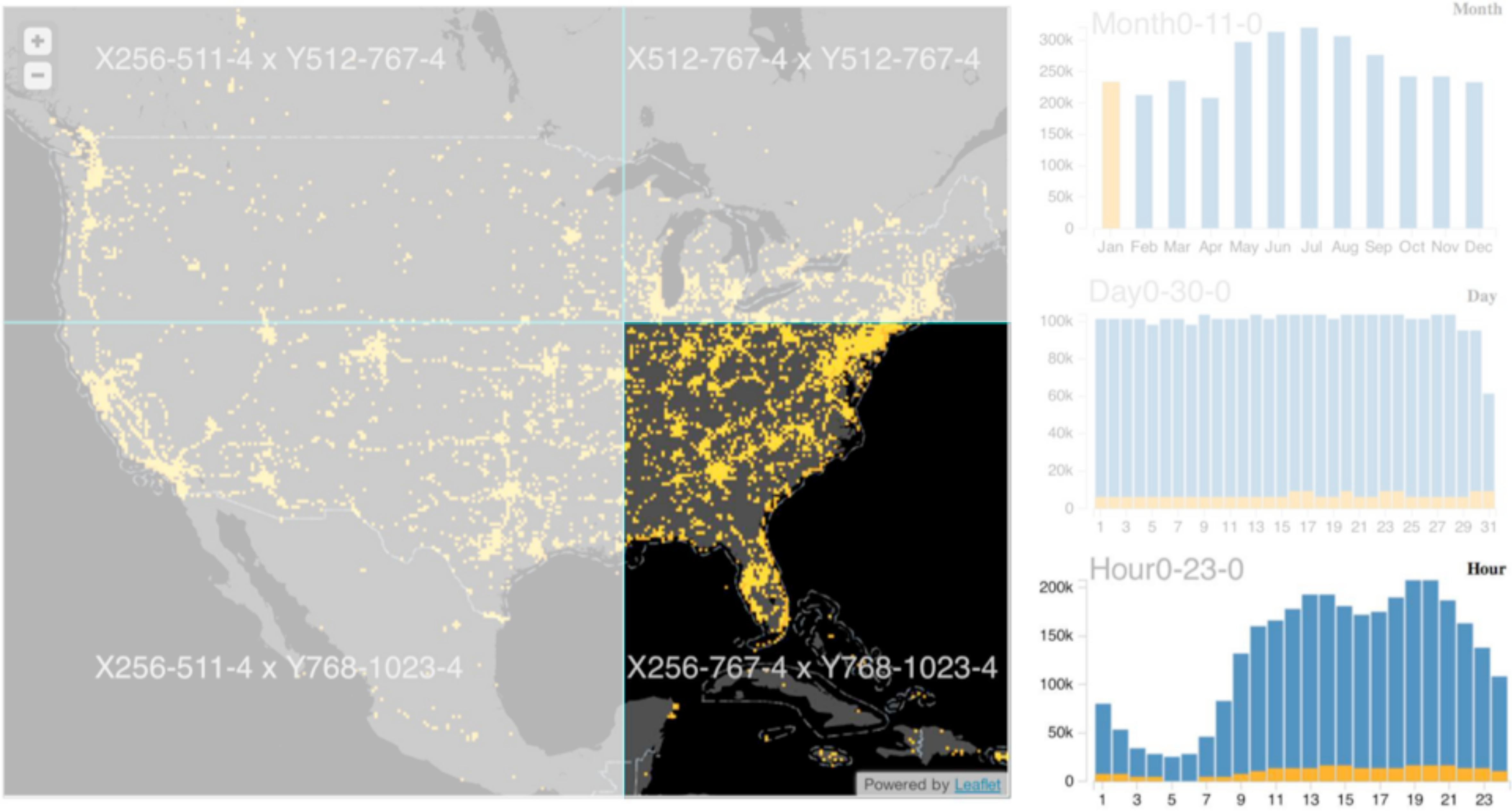


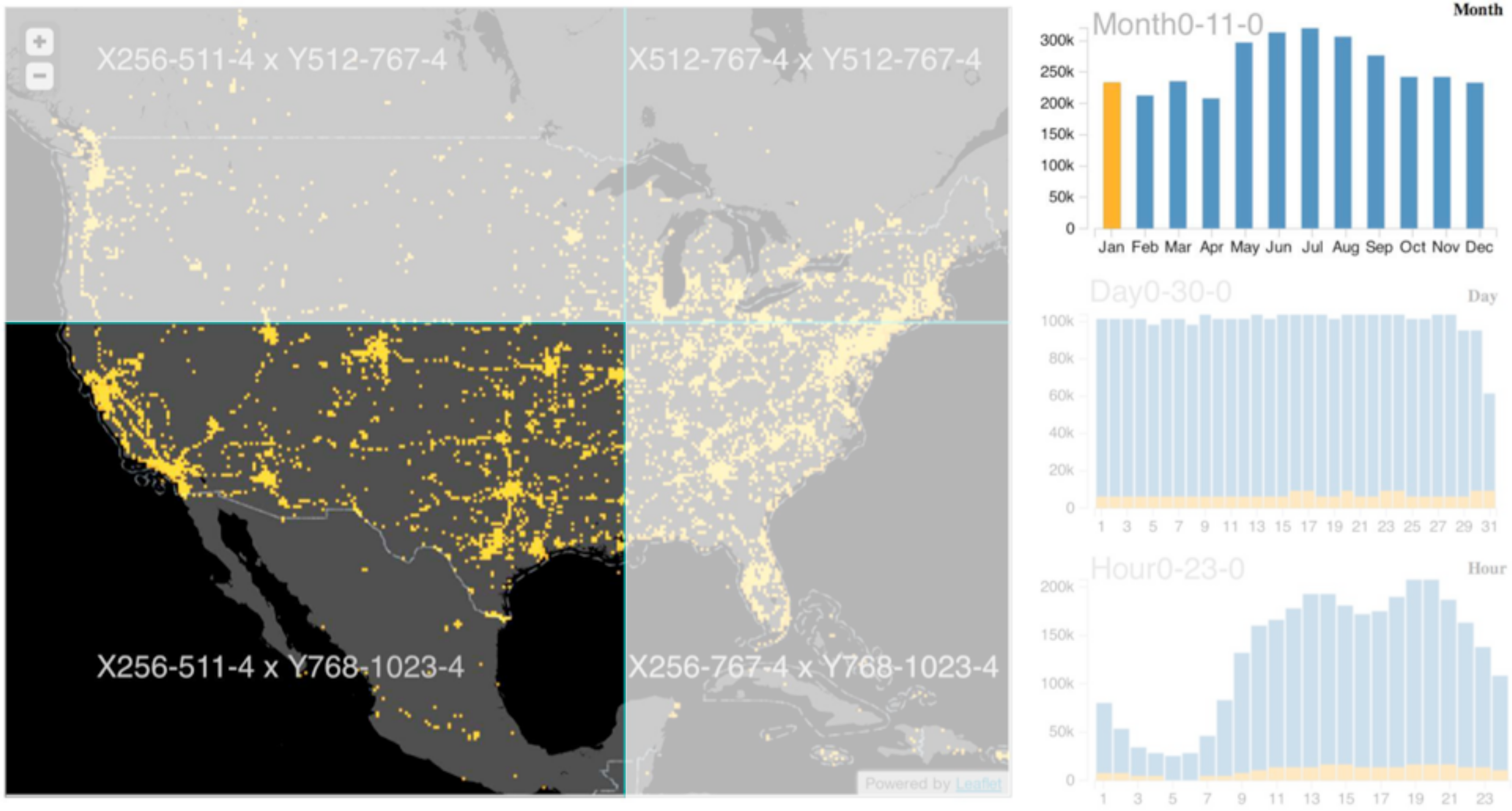
X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
Hour0-23-0
Hour



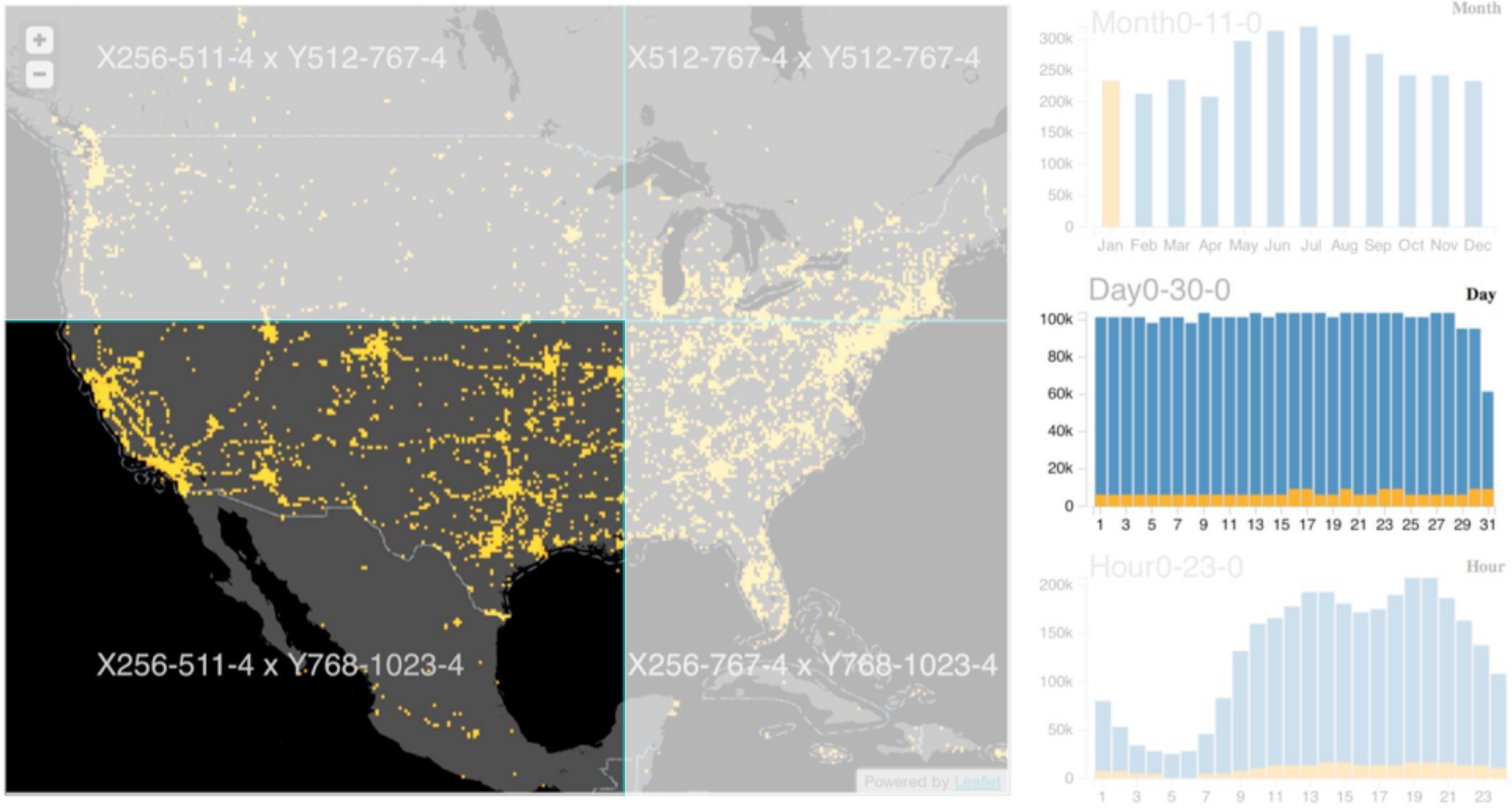
X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23

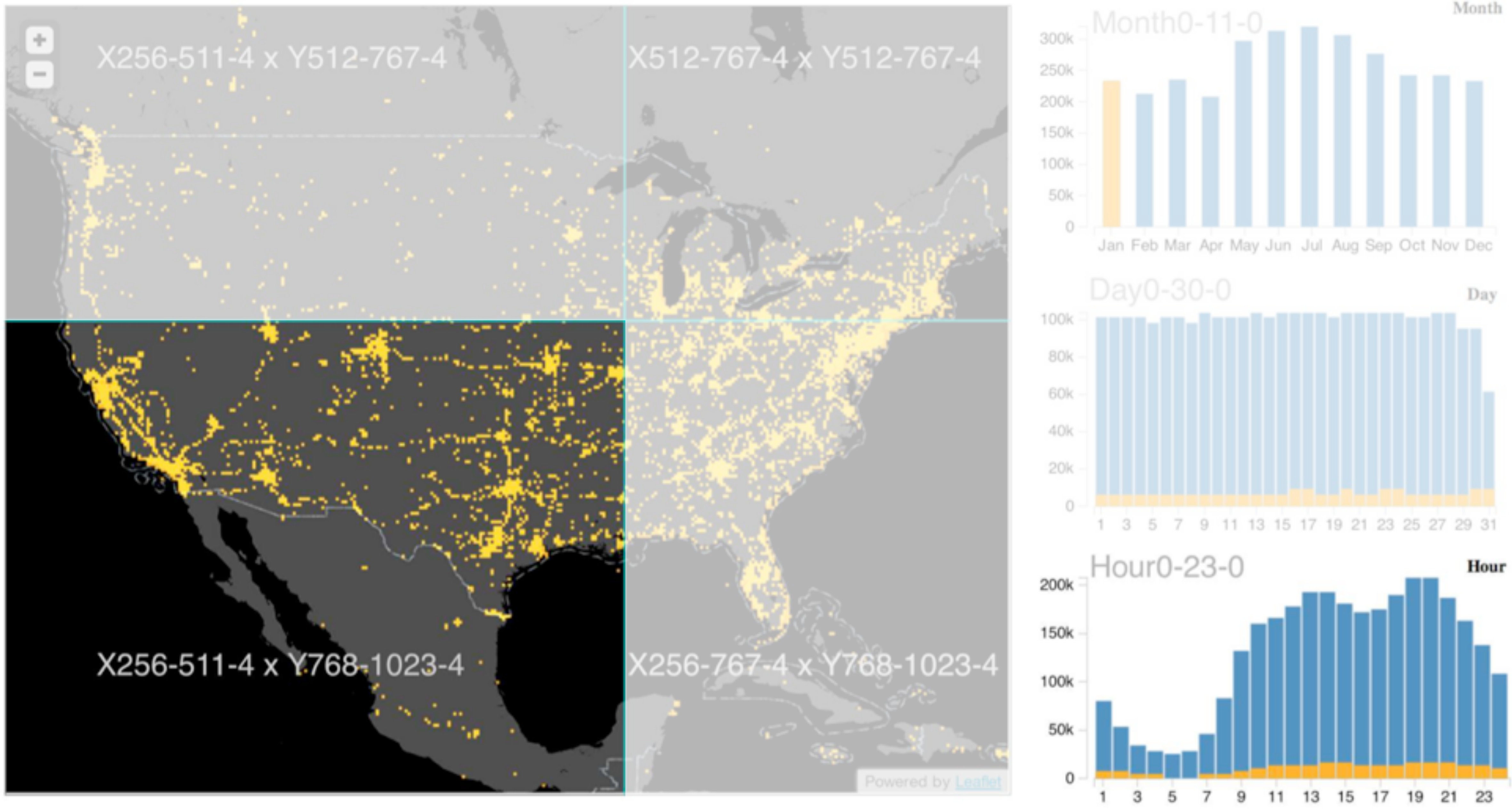

X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23

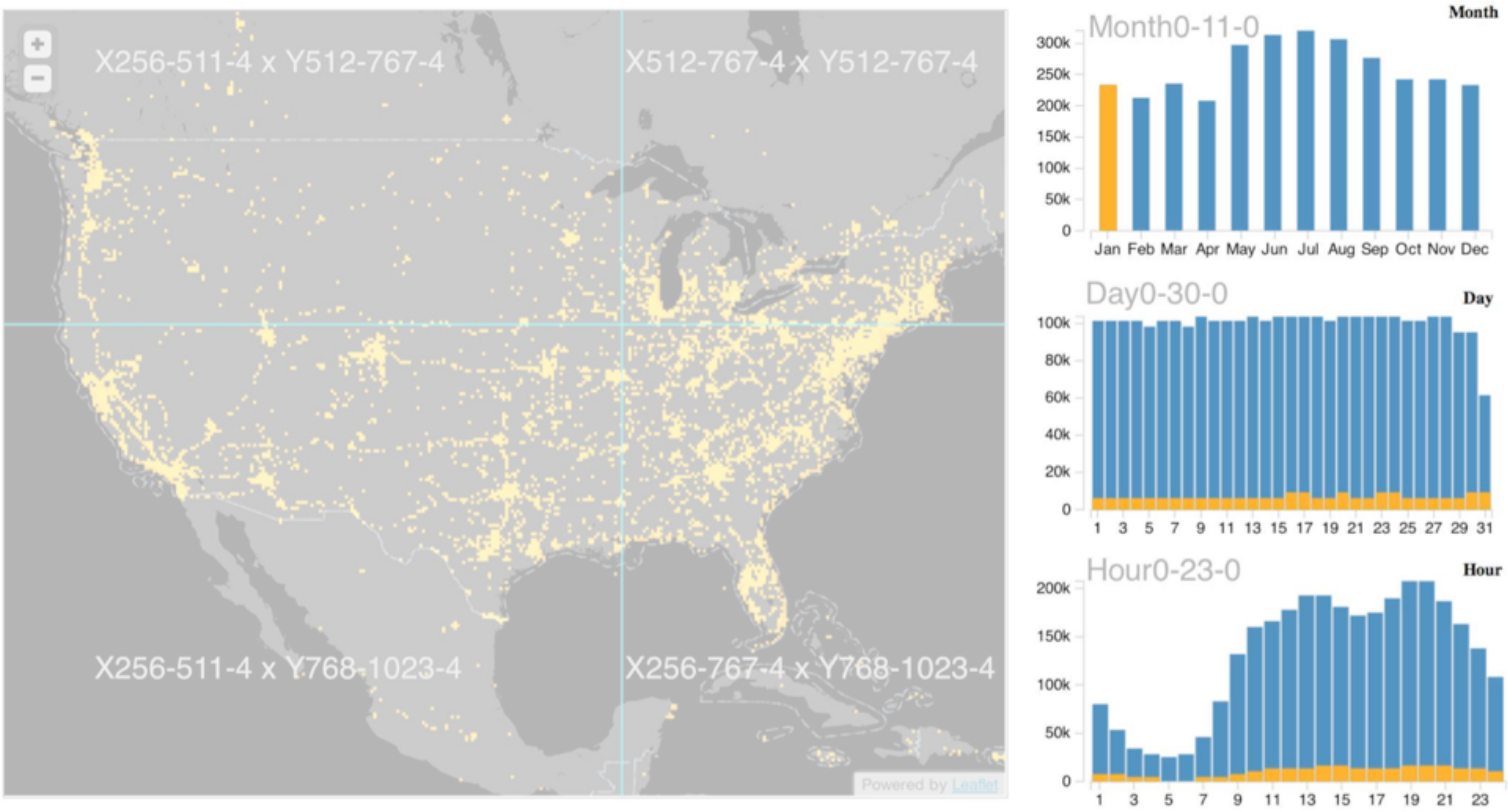


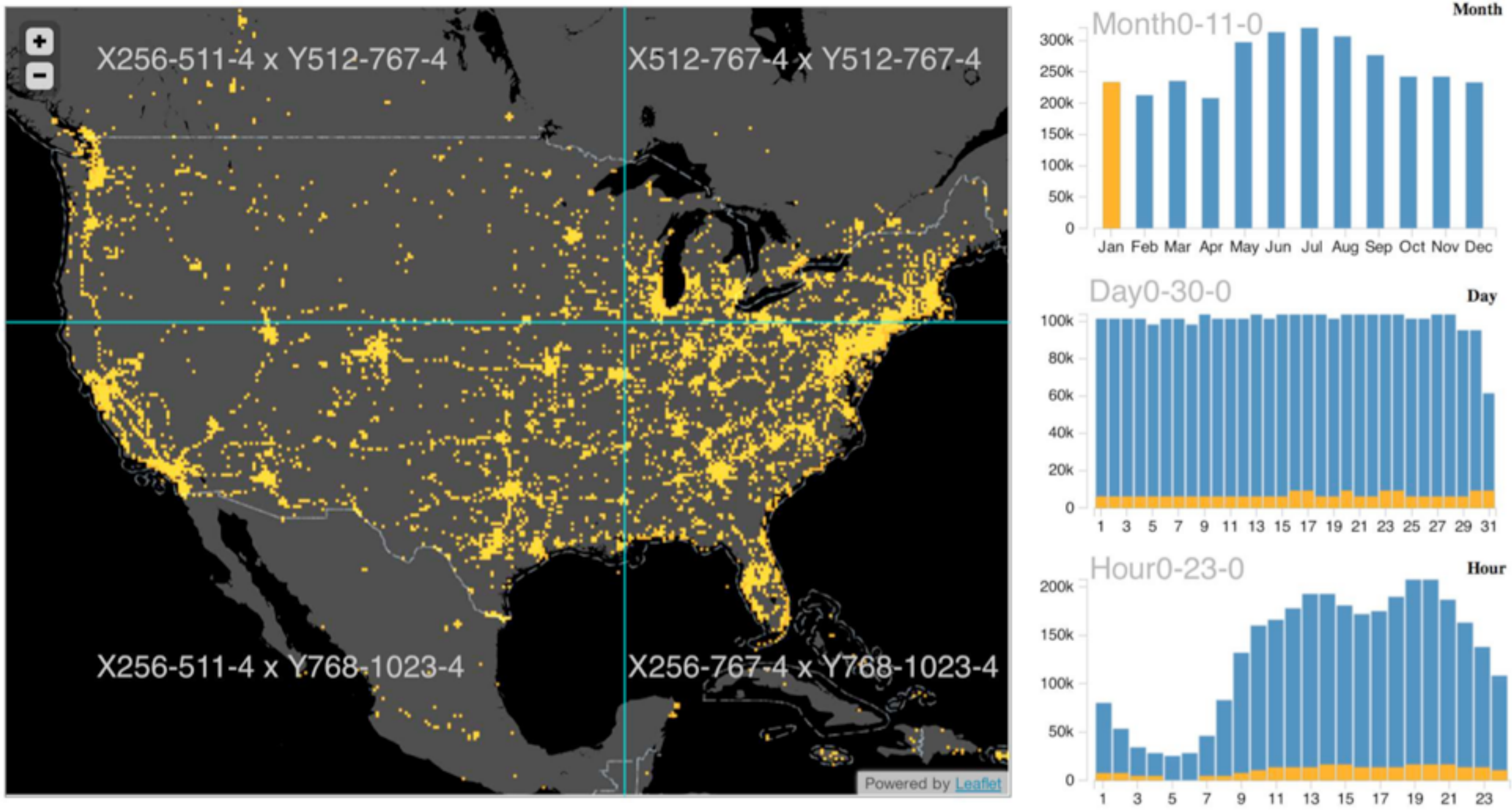
X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
0 50k 100k 150k 200k 250k 300k
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
0 20k 40k 60k 80k 100k
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
0 50k 100k 150k 200k
1 3 5 7 9 11 13 15 17 19 21 23


X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
0 50k 100k 150k 200k 250k 300k
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
0 20k 40k 60k 80k 100k
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
0 50k 100k 150k 200k
1 3 5 7 9 11 13 15 17 19 21 23

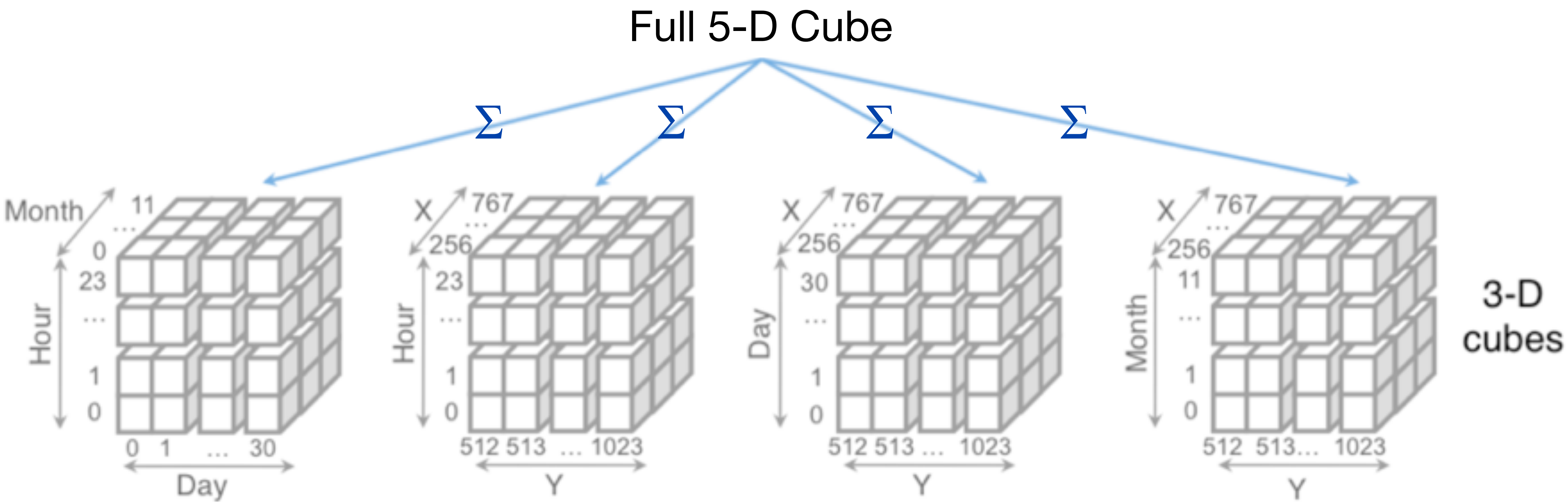


X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
Hour0-23-0
Hour

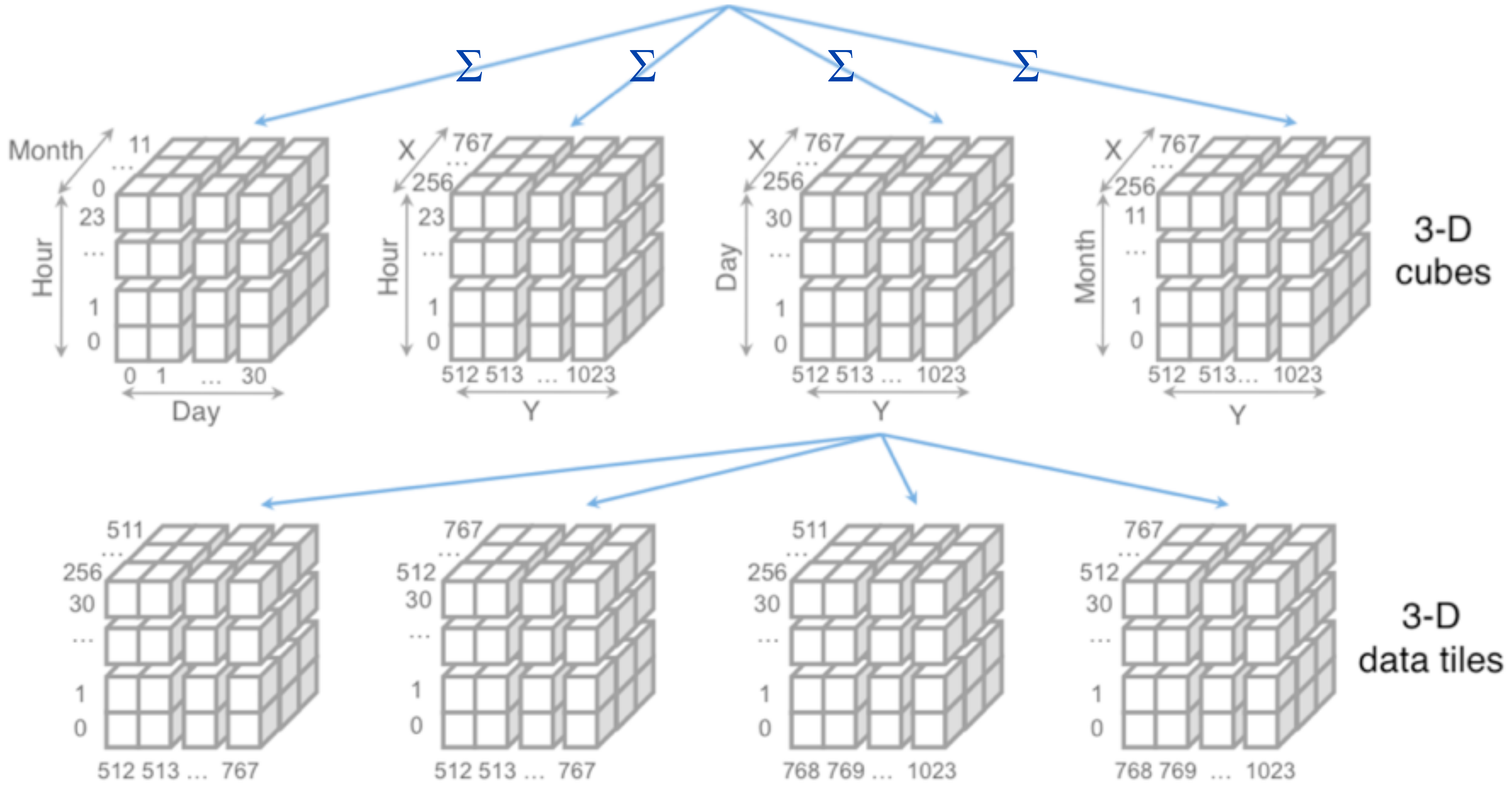

X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month
Month0-11-0
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day
Day0-30-0
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour
Hour0-23-0
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23

Full 5-D Cube

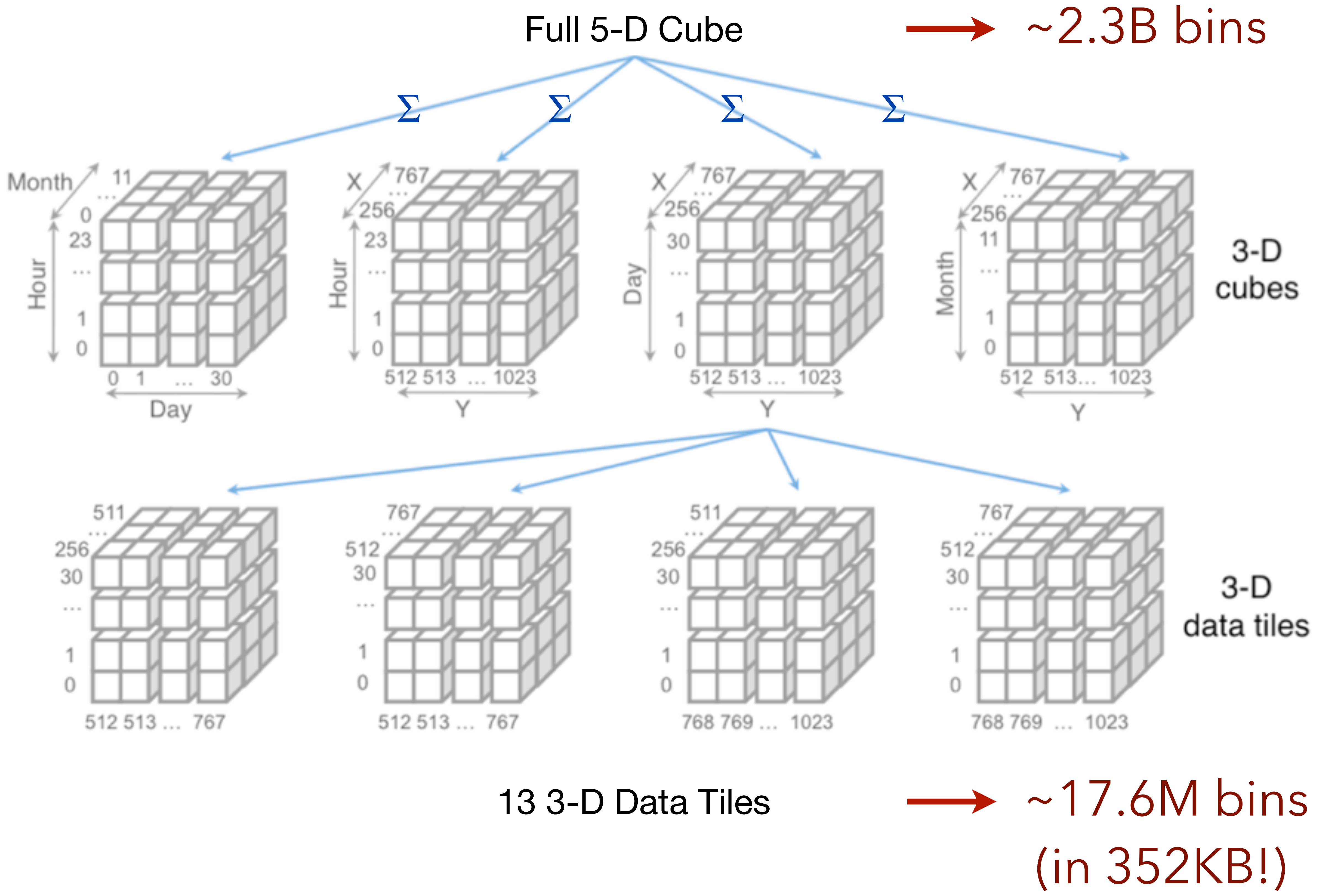


For any pair of 1D or 2D binned plots, the maximum number of dimensions needed to support brushing & linking is **four**.

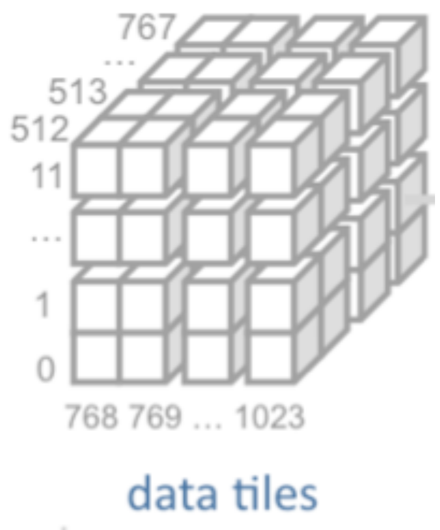
Full 5-D Cube



13 3-D Data Tiles

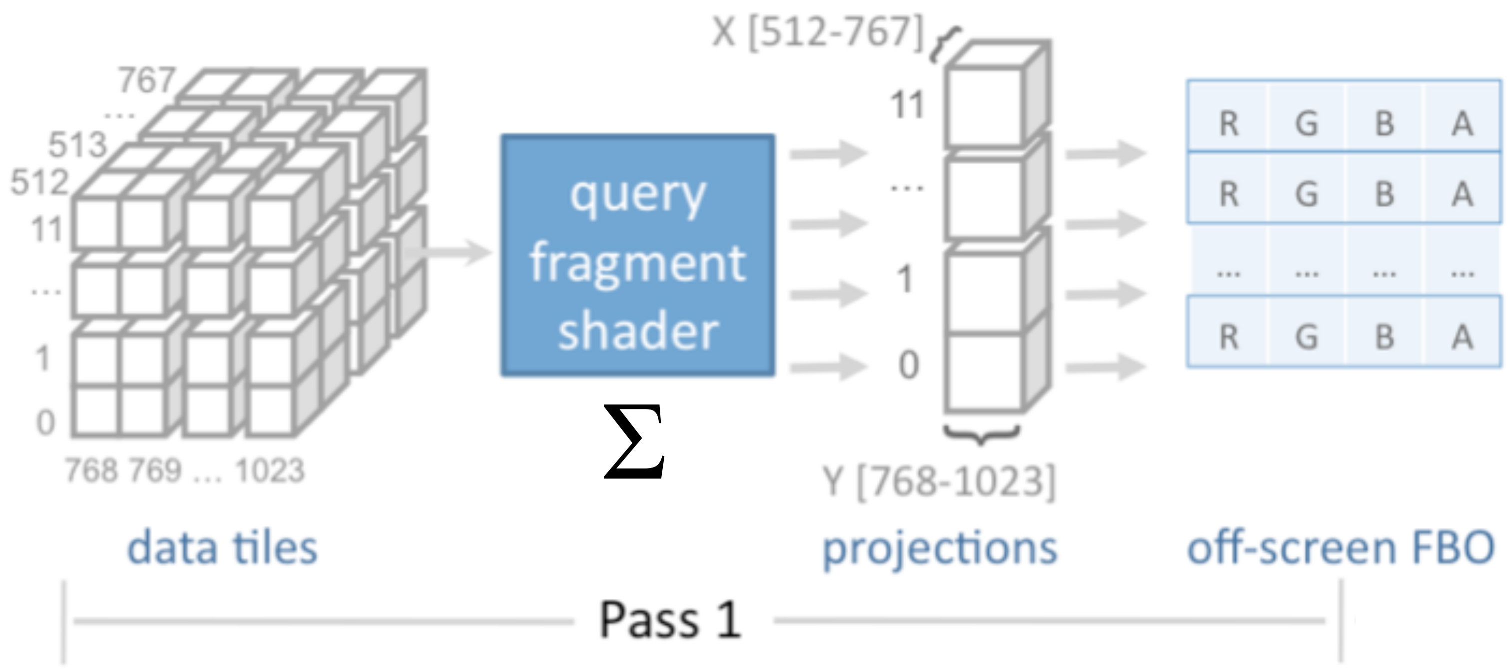

Full 5-D Cube
~2.3B bins
Σ
Σ
Σ
Σ
Month
Hour
Day
X
Y
3-D
cubes
3-D
data tiles
13 3-D Data Tiles
~17.6M bins
(in 352KB!)

Query & Render on GPU (WebGL)



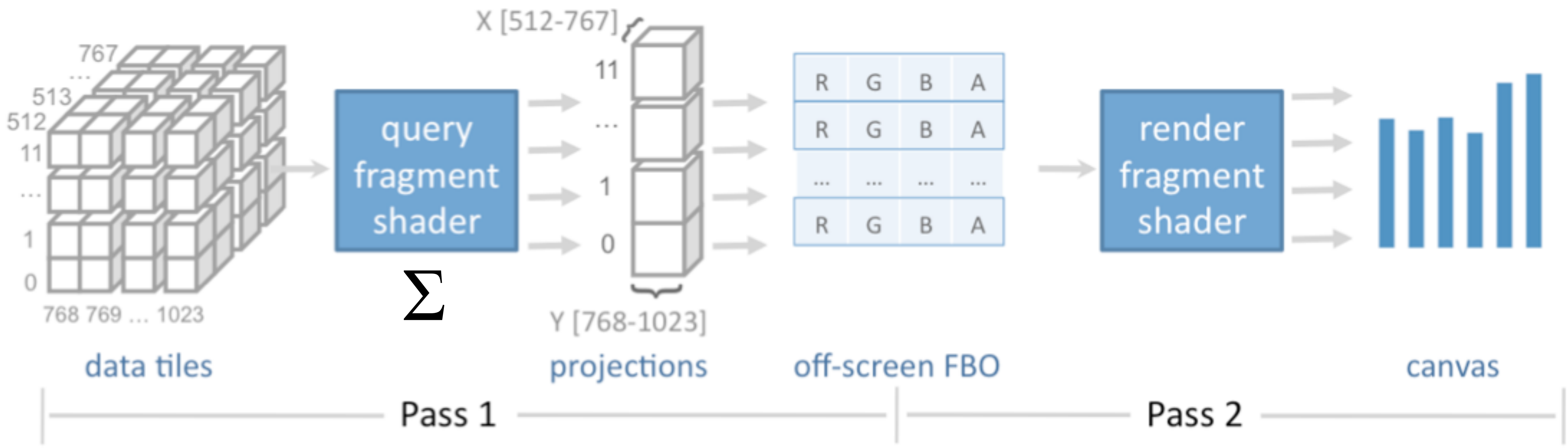
Pre-compute tiles & send from server.
Bind data tiles as image textures.

Query & Render on GPU (WebGL)



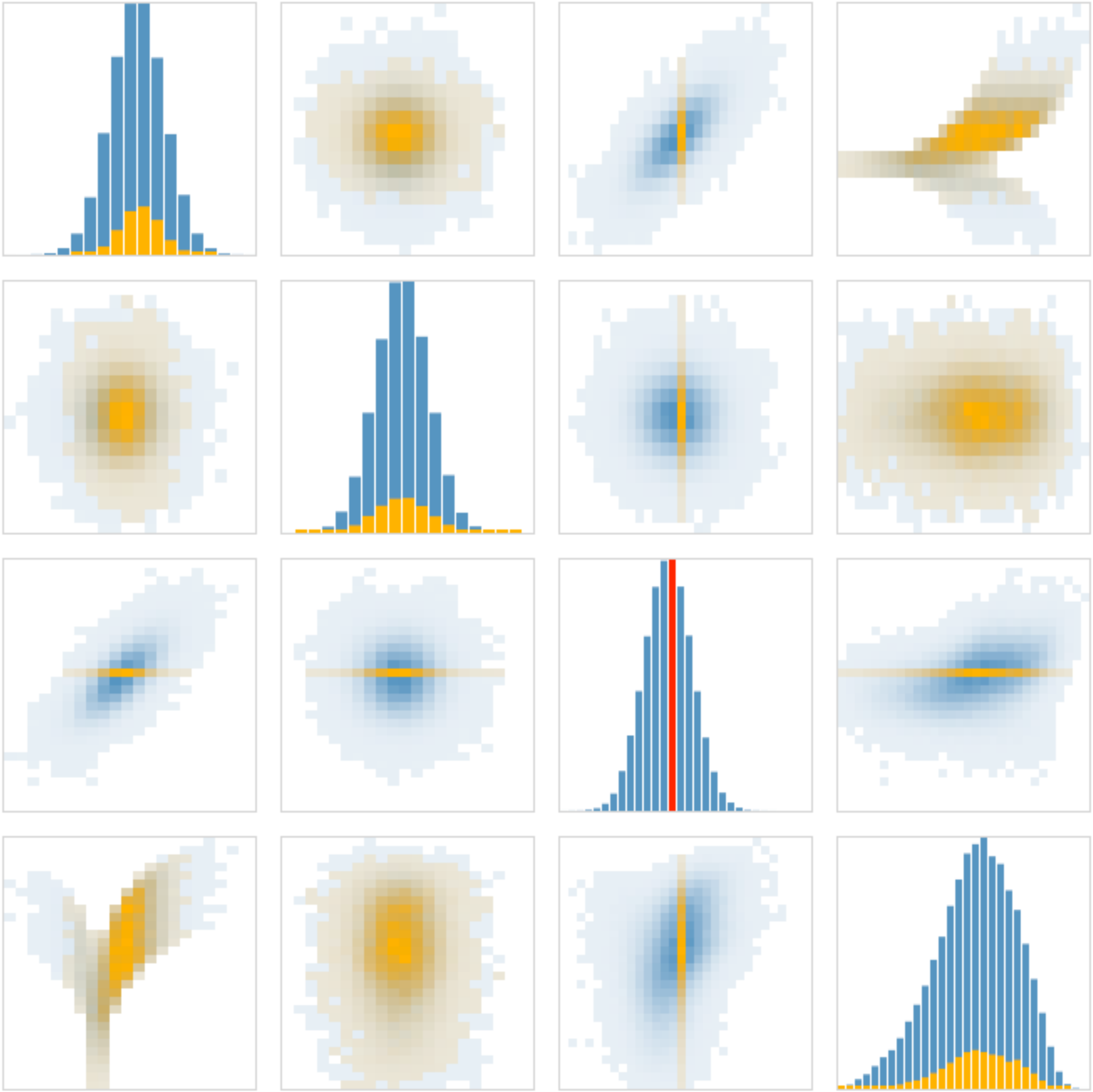
Compute aggregation for each output bin.
Executes in parallel on GPU.

Query & Render on GPU (WebGL)



Accumulate results in offscreen buffer.
Render resulting plots in second pass.

Performance Benchmarks

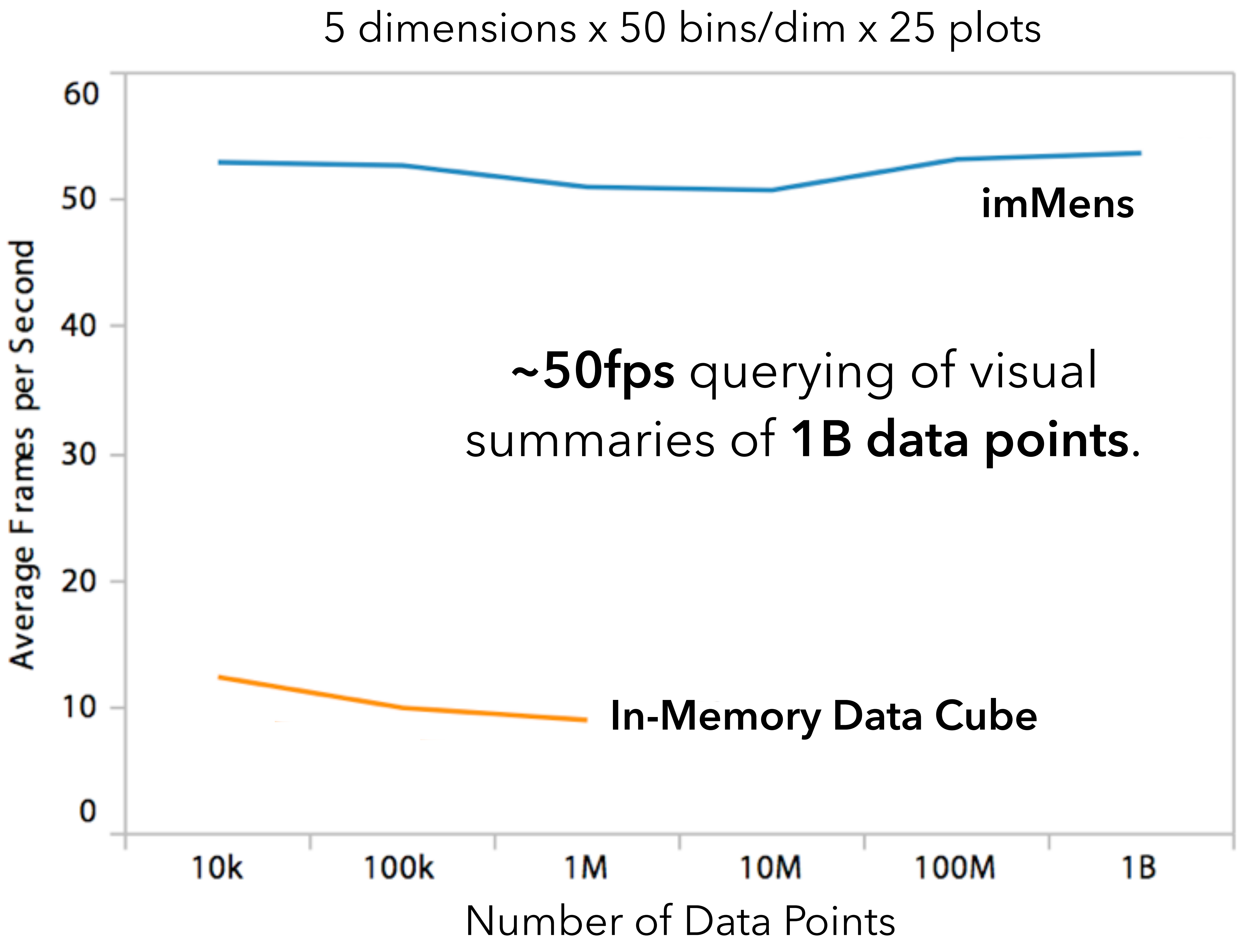


Simulate interaction: brushing & linking across binned plots.

- 4x4 and 5x5 plots
- 10 to 50 bins

Measure time from selection to render.

Test setup:
2.3 GHz MacBook Pro
NVIDIA GeForce GT 650M
Google Chrome v.23.0


5 dimensions x 50 bins/dim x 25 plots
60
50
40
30
20
10
0
Average Frames per Second
10k
100k
1M
10M
100M
1B
Number of Data Points
imMens
In-Memory Data Cube
~50fps querying of visual summaries of 1B data points.

Limitations and Questions

But where do the multivariate data tiles come from?

They must be provided by a backend server. This can be time-consuming, particularly if supporting deep levels of zooming. imMens assumes that tiles have either been pre-computed or that a backing database can suitably generate them on demand.

Does super-low-latency interaction really matter?

Is it worth it to go to all of this trouble?

How does **interactive latency** affect exploratory analysis with visualizations?

[Liu & Heer '14]

Prior Work

Higher latency entails higher action costs, subjects satisfice by selecting strategies that *reduce short-term effort* with no guarantee that the final outcome is optimized. [Gray & Boehm-Davis]

Prior Work

Higher latency entails higher action costs, subjects satisfice by selecting strategies that *reduce short-term effort* with no guarantee that the final outcome is optimized. [Gray & Boehm-Davis]

300ms latency reduces the number of Google searches; effect persists for days. [Brutlag et al]

Prior Work

Higher latency entails higher action costs, subjects satisfice by selecting strategies that *reduce short-term effort* with no guarantee that the final outcome is optimized. [Gray & Boehm-Davis]

300ms latency reduces the number of Google searches; effect persists for days. [Brutlag et al]

When the cost of acquiring information is increased, subjects change strategy and rely more on working memory. [Ballard et al]

Prior Work

Higher latency entails higher action costs, subjects satisfice by selecting strategies that *reduce short-term effort* with no guarantee that the final outcome is optimized. [Gray & Boehm-Davis]

When confronted with increased latencies, users resort to more mental planning, at times making fewer errors and performing better on tasks with *verifiable outcomes*. [O'Hara & Payne]

Prior Work

Higher latency entails higher action costs, subjects satisfice by selecting strategies that *reduce short-term effort* with no guarantee that the final outcome is optimized. [Gray & Boehm-Davis]

When confronted with increased latencies, users resort to more mental planning, at times making fewer errors and performing better on tasks with *verifiable outcomes*. [O'Hara & Payne]

But what about *open, exploratory analysis tasks*?

Experiment Design

2 (Latency) x 2 (Scenario) Design

Latency: +0ms / +500ms

Scenario: Mobile Check-ins / FAA Flight Delays

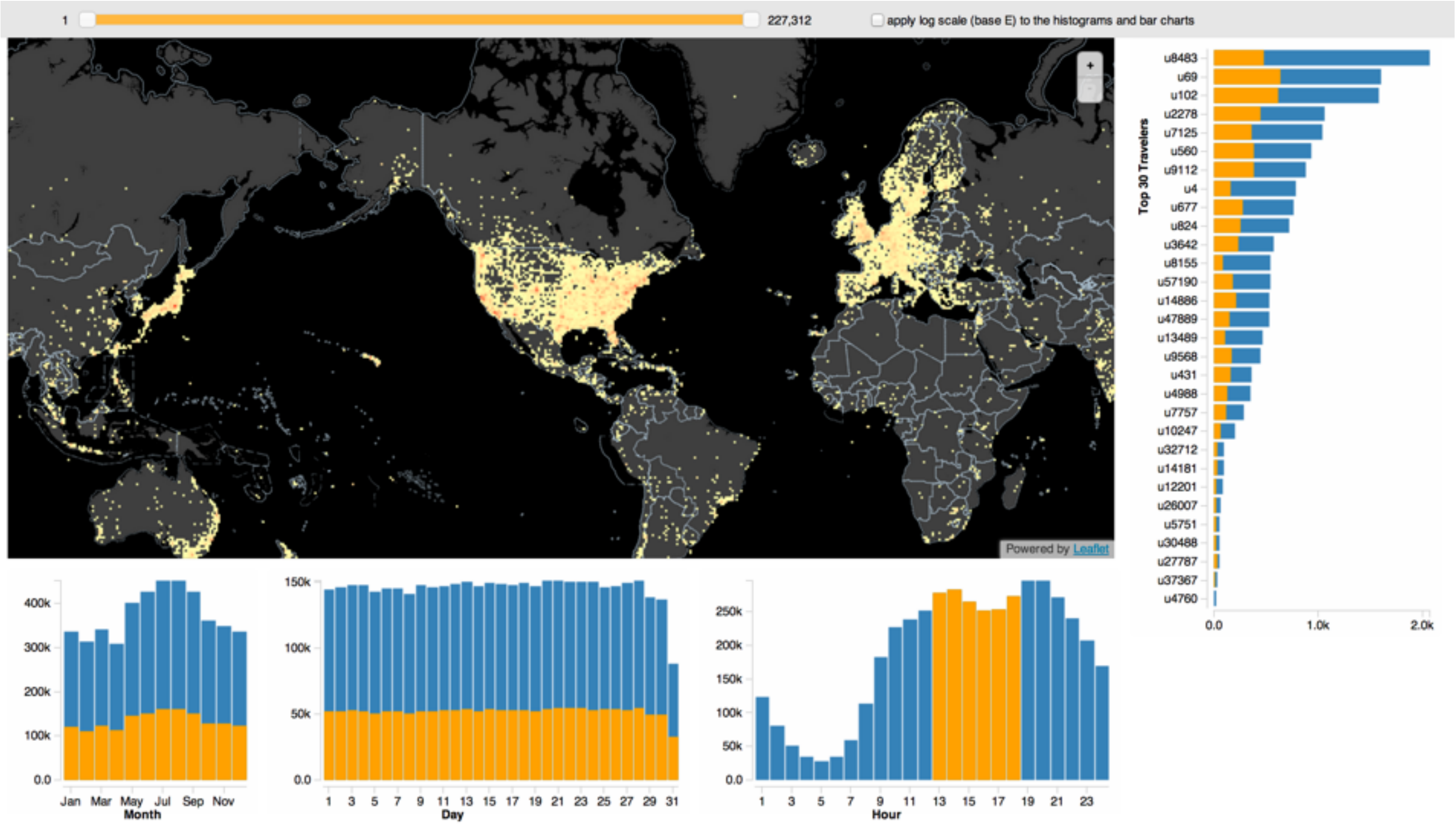
Exploratory Analysis Tasks (2 per session)

imMens with brush, pan, zoom, adjust scales

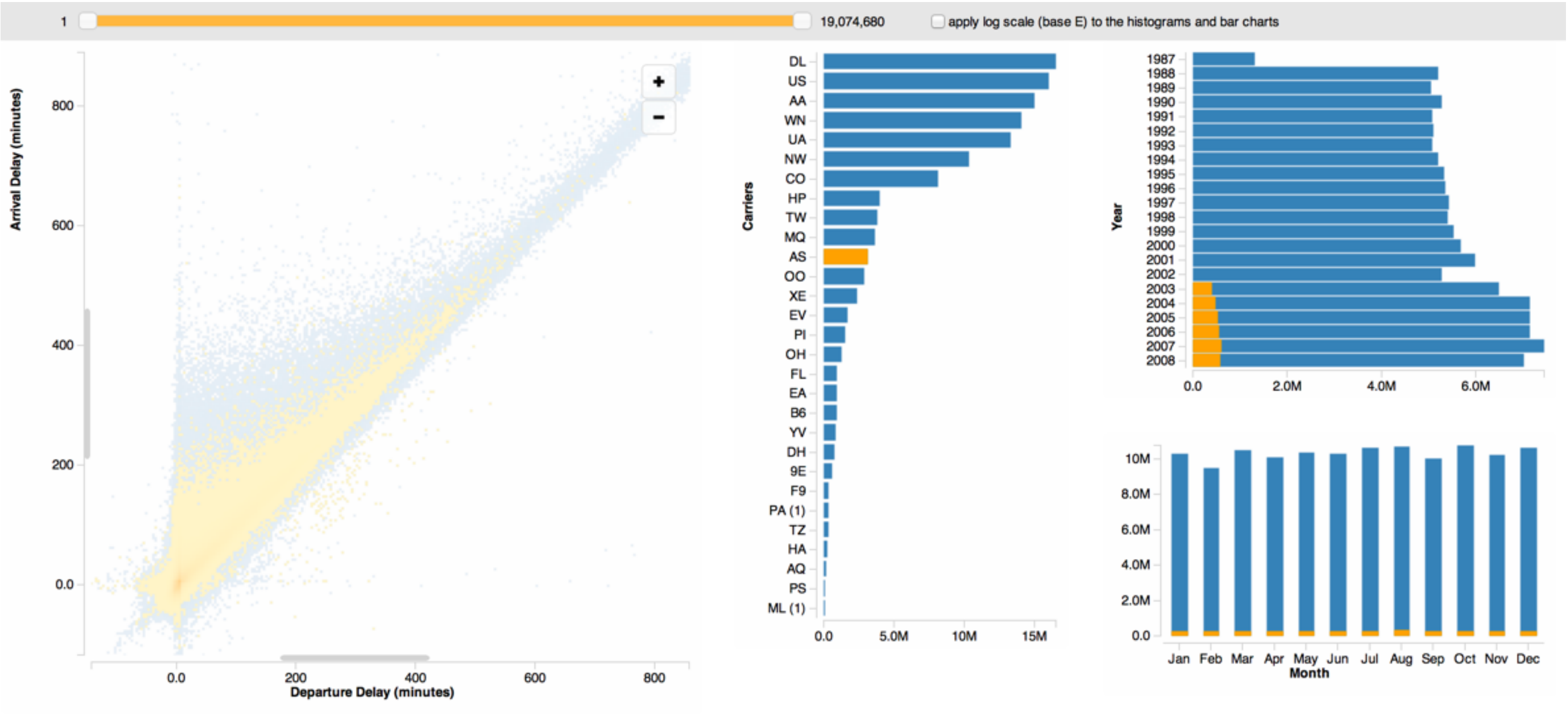
Users asked to explore data and share findings

Log events, record audio and screen capture

16 subjects, all familiar with data analysis + vis



4.5m Mobile Check-Ins



140m FAA Flight Delay Records

Data Collection & Analysis

Event Log Analysis

Analyze triggered & processed user input events

Assess data set coverage (# unique tiles)

Verbal Protocol Analysis

Think-aloud protocol: verbalize thought process

Transcribe sessions; Code actions and insights

Analyze number and type of coded events

Latency Study Results

Higher latency leads to…

Latency Study Results

Higher latency leads to…

Reduced user activity and data set coverage

Latency Study Results

Higher latency leads to…

Reduced user activity and data set coverage

Significantly fewer brushing actions

Latency Study Results

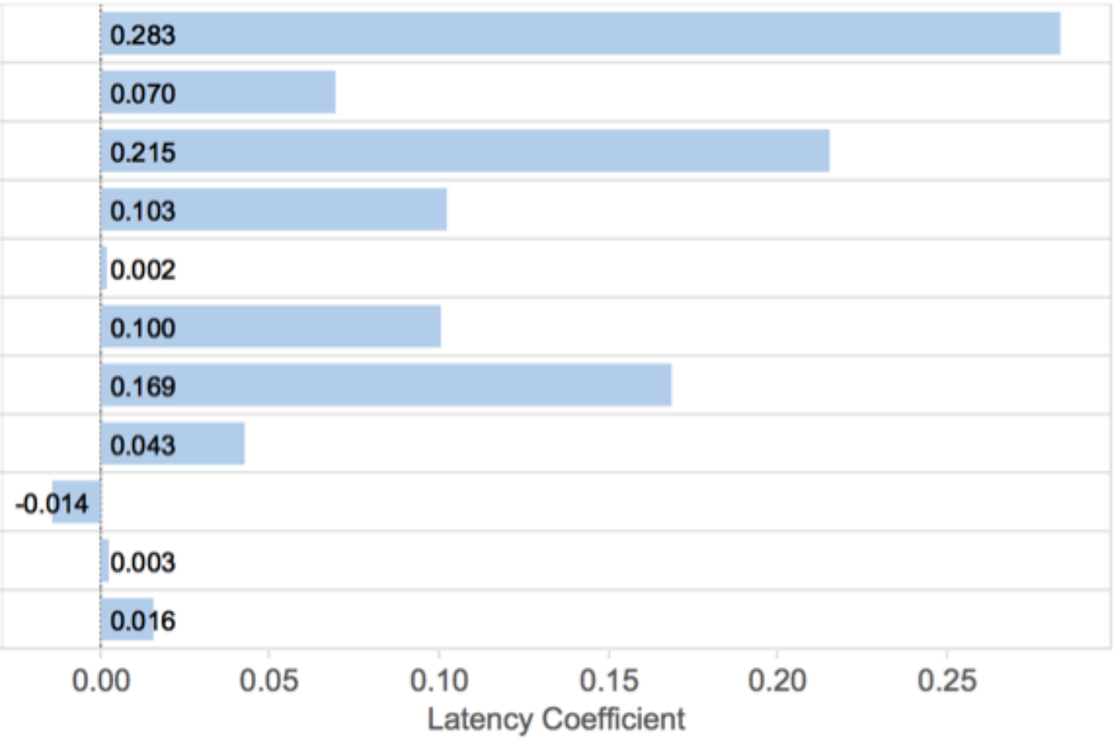
Higher latency leads to…

Reduced user activity and data set coverage

Significantly fewer brushing actions

Less observation, generalization & hypothesis

Verbal Category	likelihood-ratio test: Chisq(1, N=32)	p value	significance
Observation	5.4812	0.01922	*
Observation (Single View)	1.5706	0.2101	
Observation (Multiple Views)	3.3119	0.06878	.
Generalization	8.9763	0.002735	**
Generalization (Single View)	0.2641	0.6073	
Generalization (Multiple Views)	8.5054	0.003541	**
Hypothesis	8.3999	0.003752	**
Question	0.7416	0.3891	
Interface	0.4651	0.4953	
Recall	0.0202	0.8869	
Simulation	0.6983	0.4033	



Latency Study Results

Higher latency leads to…

Reduced user activity and data set coverage

Significantly fewer brushing actions

Less observation, generalization & hypothesis

Interaction effect: Exposure to delay reduces subsequent performance in low-latency interface.

Latency Study Results

Higher latency leads to…

Reduced user activity and data set coverage

Significantly fewer brushing actions

Less observation, generalization & hypothesis

Interaction effect: Exposure to delay reduces subsequent performance in low-latency interface.

Different interactions exhibit varied sensitivity to latency. Brushing is highly sensitive!

Latency Study Results

Higher latency leads to…

Reduced user activity and data set coverage

Significantly fewer brushing actions

Less observation, generalization & hypothesis

Interaction effect: Exposure to delay reduces subsequent performance in low-latency interface.

Different interactions exhibit varied sensitivity to latency. Brushing is highly sensitive!

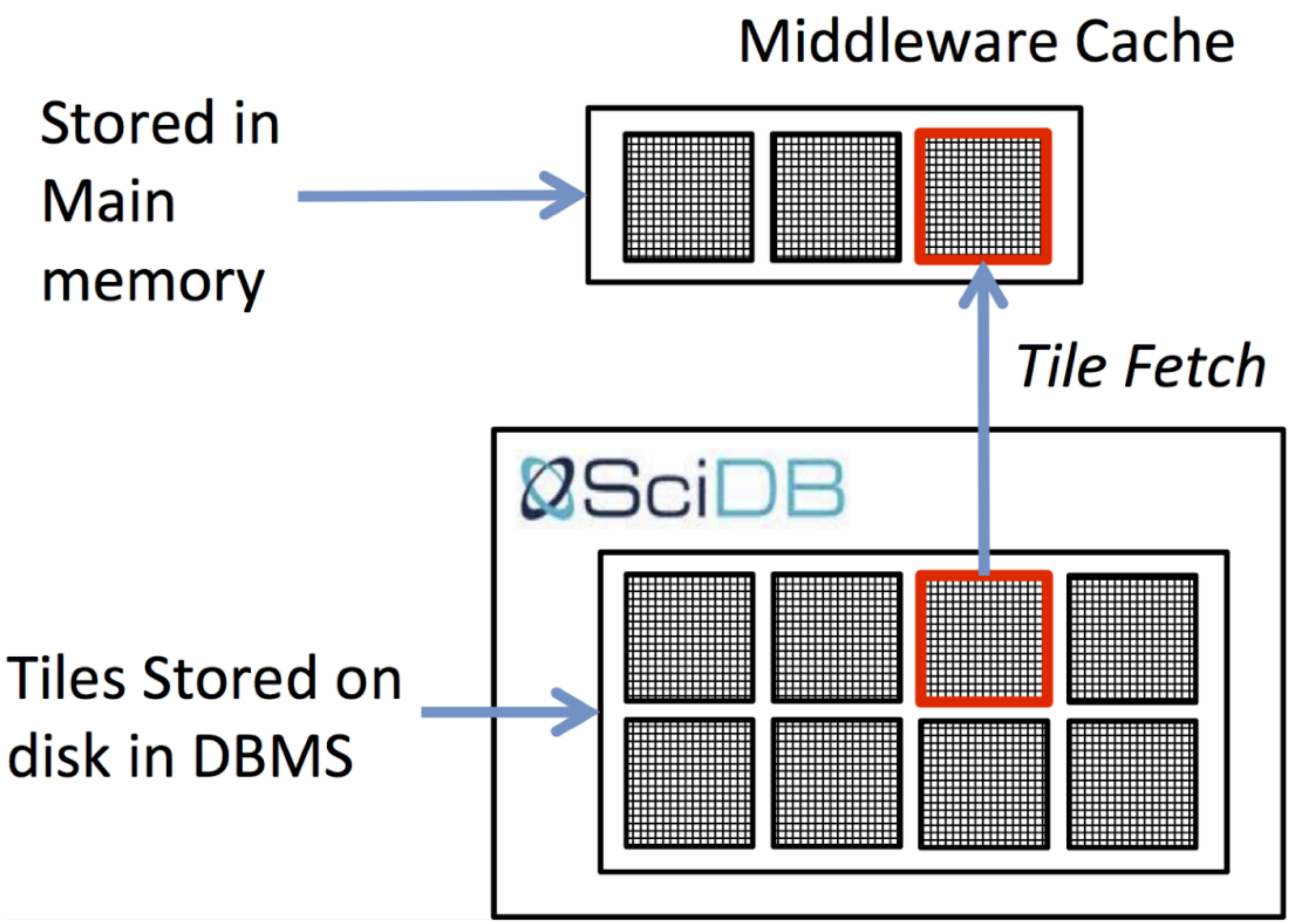
In short: milliseconds matter! And imMens was not a waste of time… 😅

ForeCache

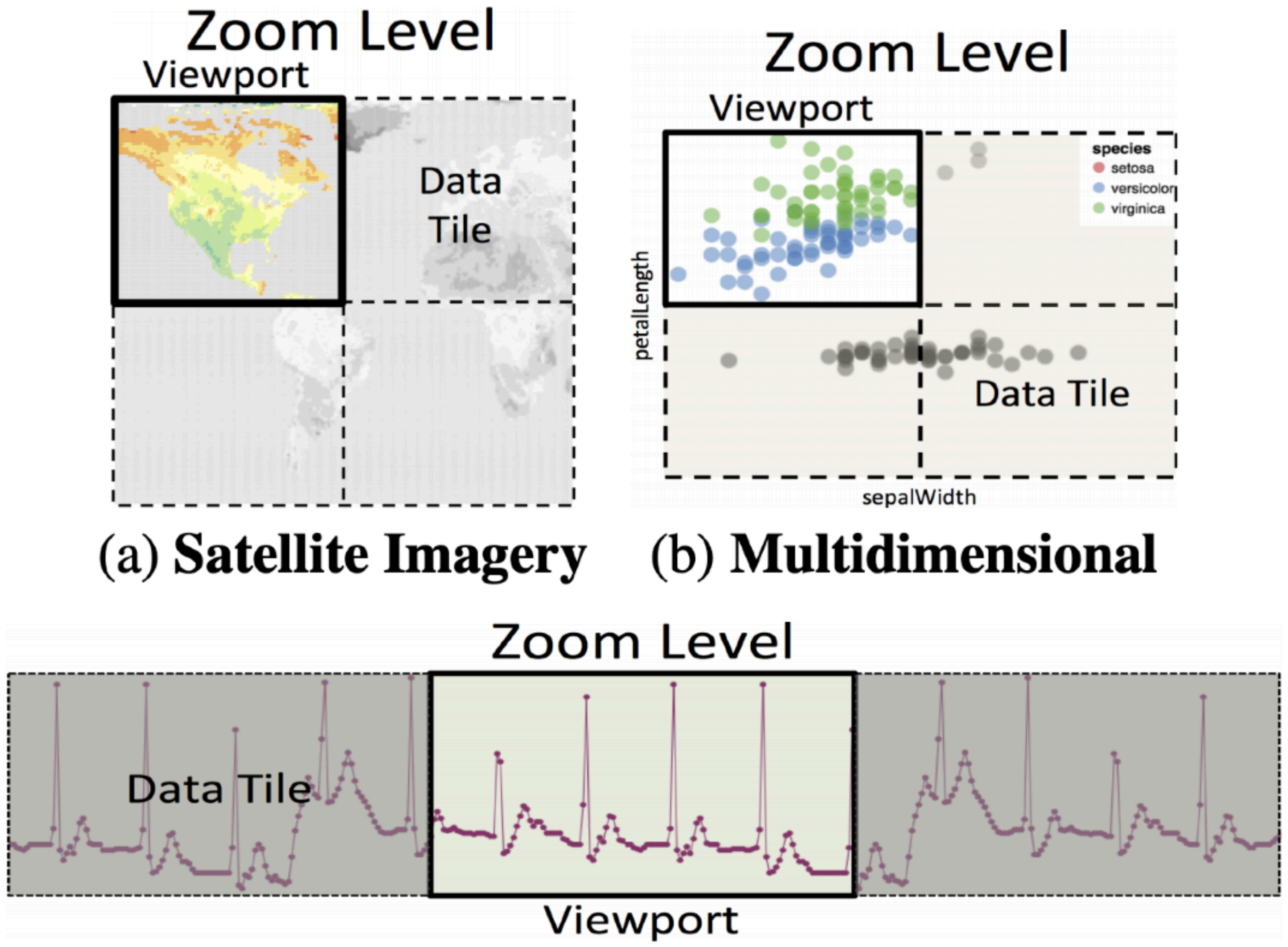
[Battle, Chang, & Stonebraker '16]

Strategies: Query Database, Pre-Fetching

ForeCache is also a Data Tile-Based System



Manage a Cache of Tiles from DB



(a) **Satellite Imagery** (b) **Multidimensional**

(c) **Timeseries (Heart rate Monitoring)**

Example Tile-Based Views

Key Idea: Model & Predict User Behavior

1. Classify Analysis Phase

Foraging: Searching for patterns of interest
Sensemaking: Closely examine a region-of-interest
Navigation: Transition between levels of detail

Train a machine learning classifier (SVM) to predict phase.
The input data is the activity trace of user interactions.

Key Idea: Model & Predict User Behavior

1. Classify Analysis Phase

2. Apply Prediction Models

Actions-Based: Use recent interactions to predict next ones.

You pan left twice; what is the probability you will do it again?

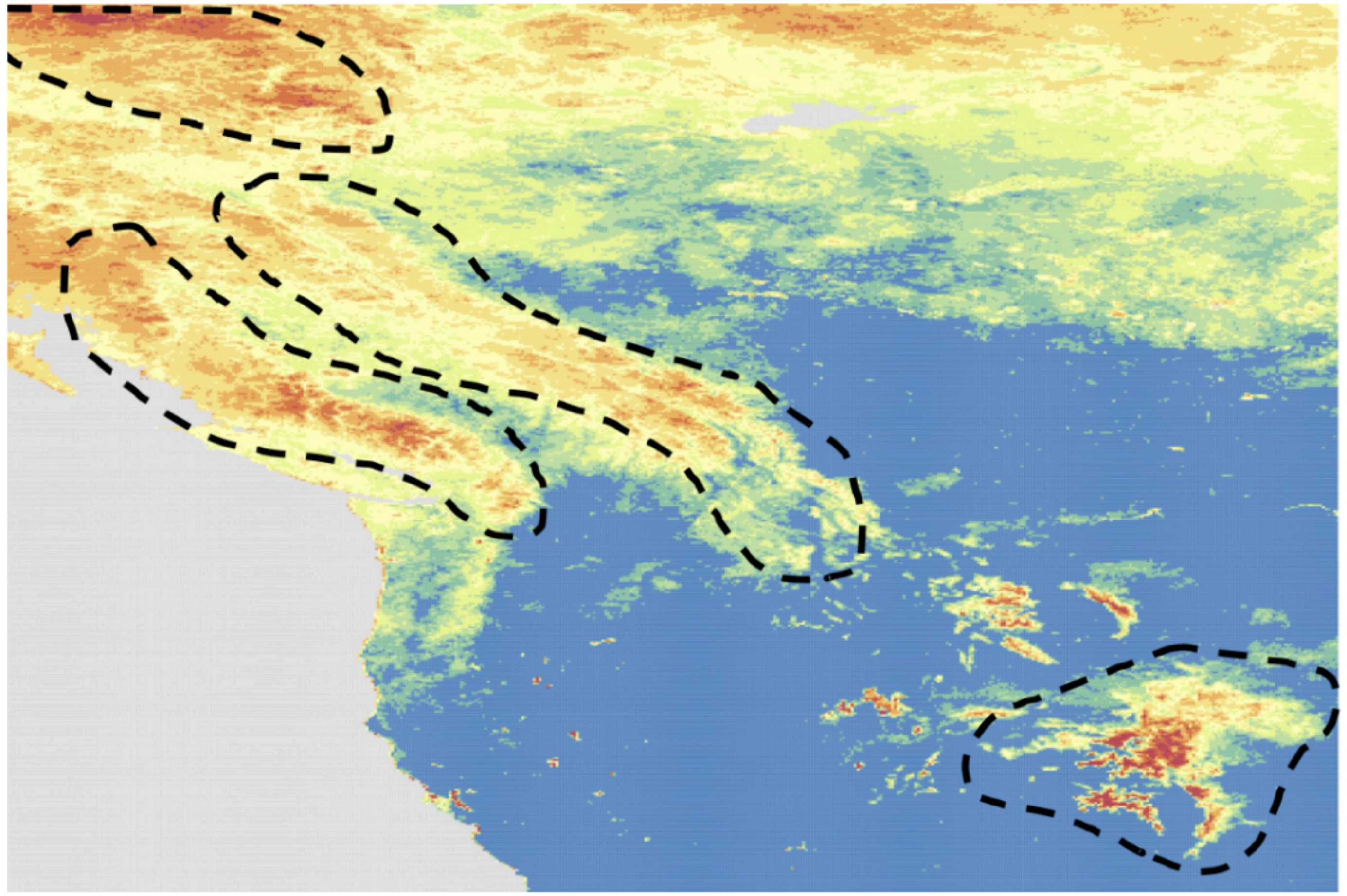
Signature-Based: Match to data characteristics of interest.

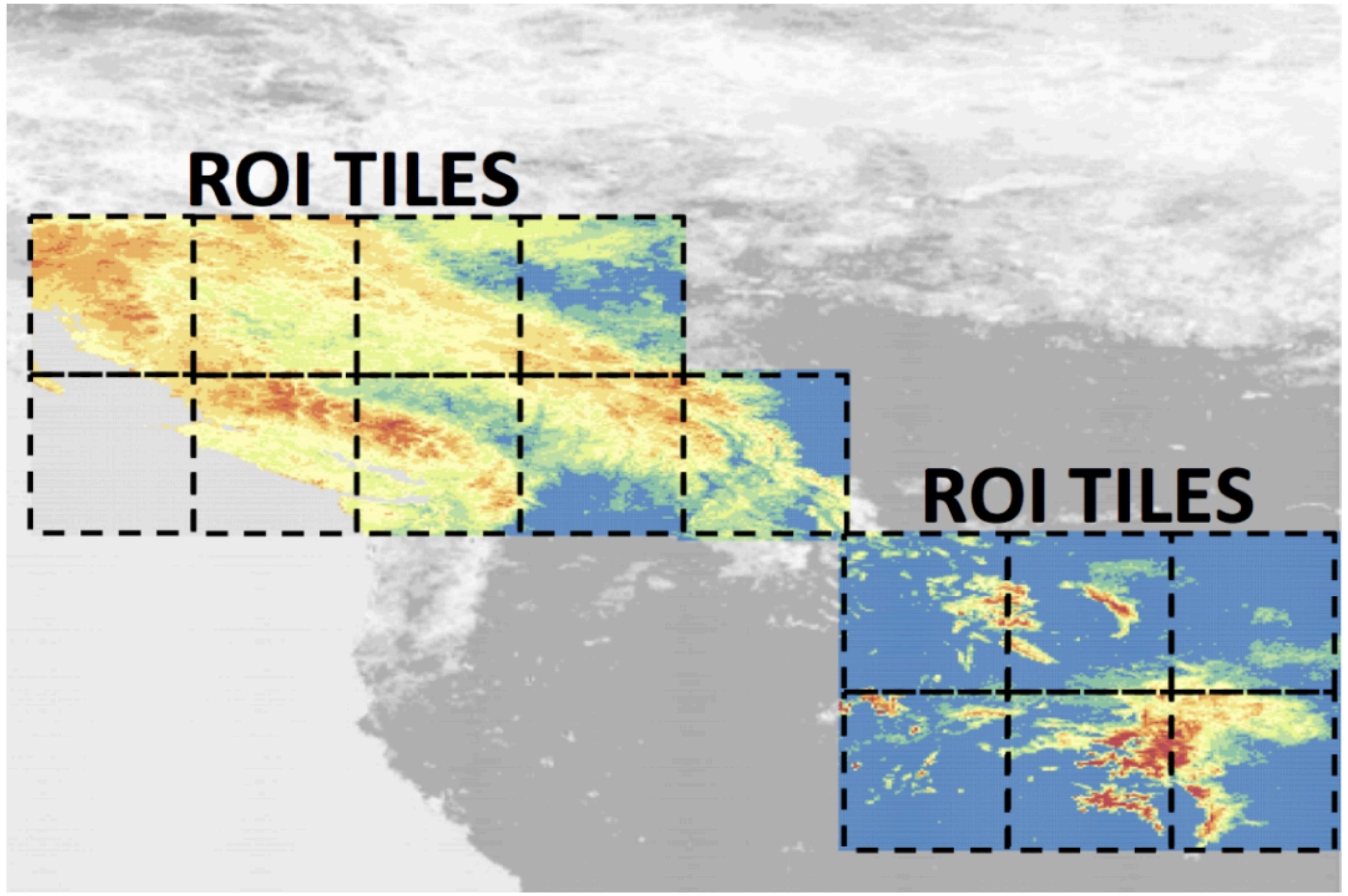
What data tiles are visually similar to current focus tiles?

These models are weighted based on the analysis phase.

Actions-Based for *navigation*. Signature-Based for *sensemaking*. Both applied equally for *foraging*.

Application: MODIS Satellite Data





Analyzing snow cover in a scientific database. ROI = Region of Interest

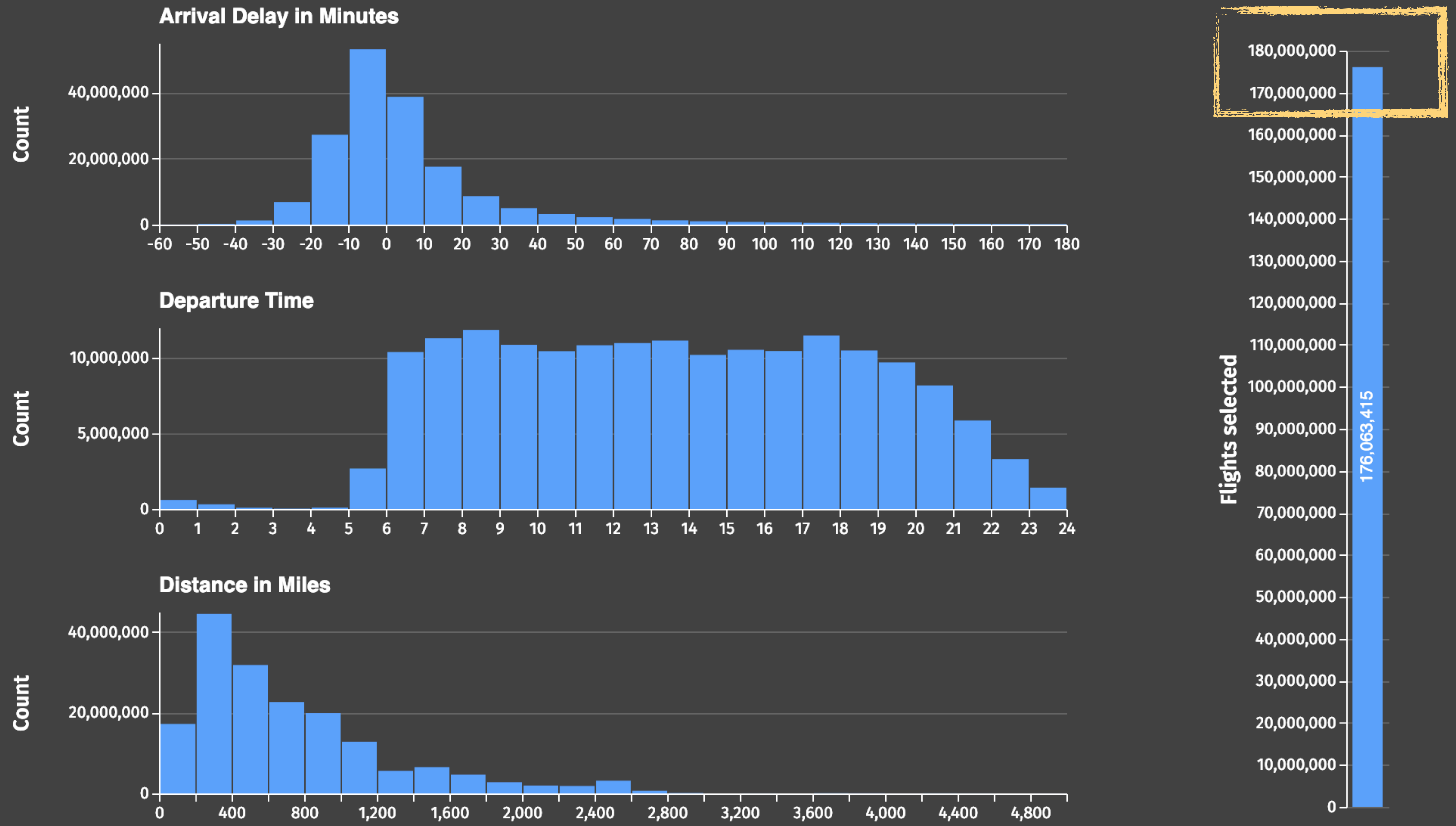
ForeCache improves latency:

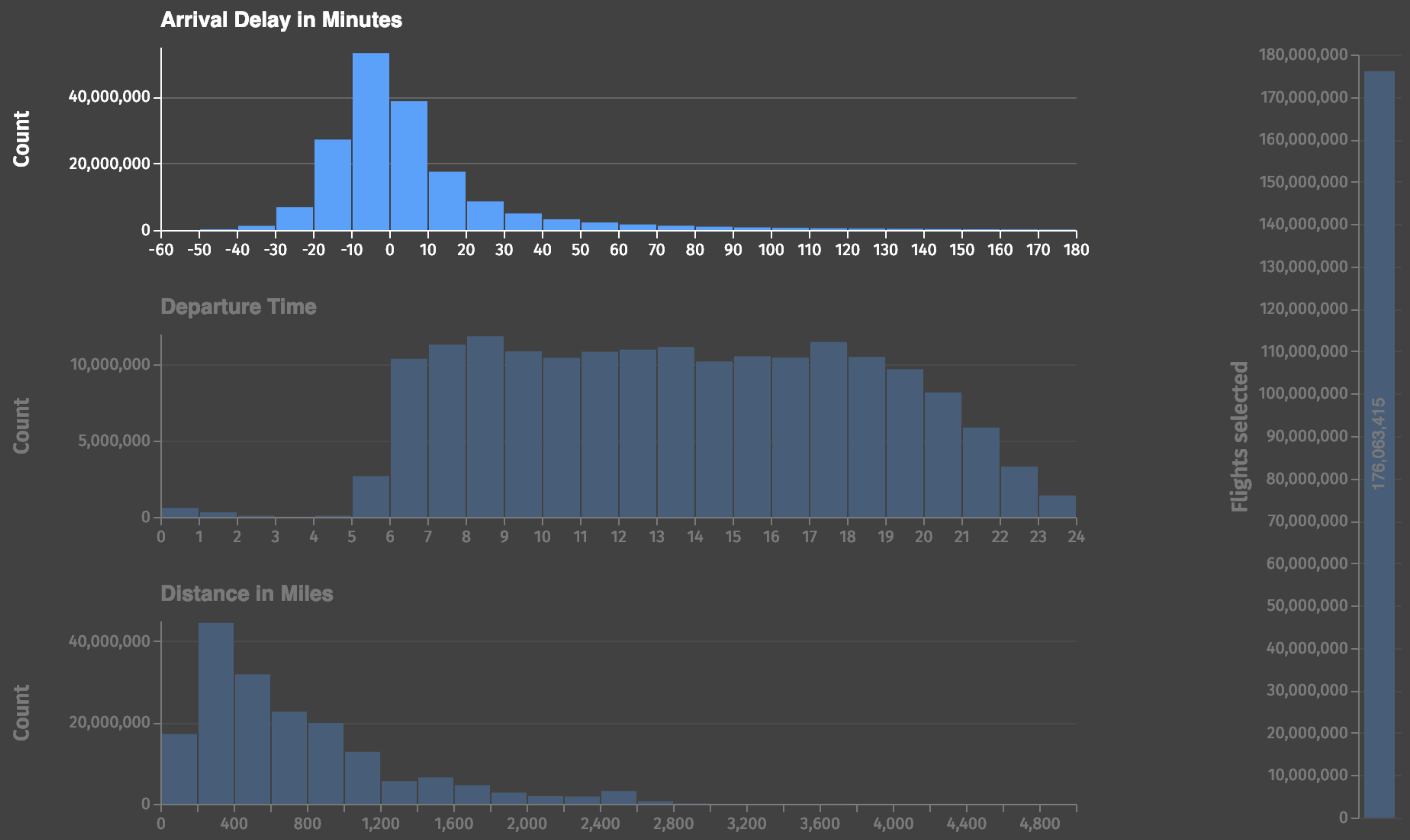
- 430% better than current non-prefetching systems
- 88% better than existing prediction methods

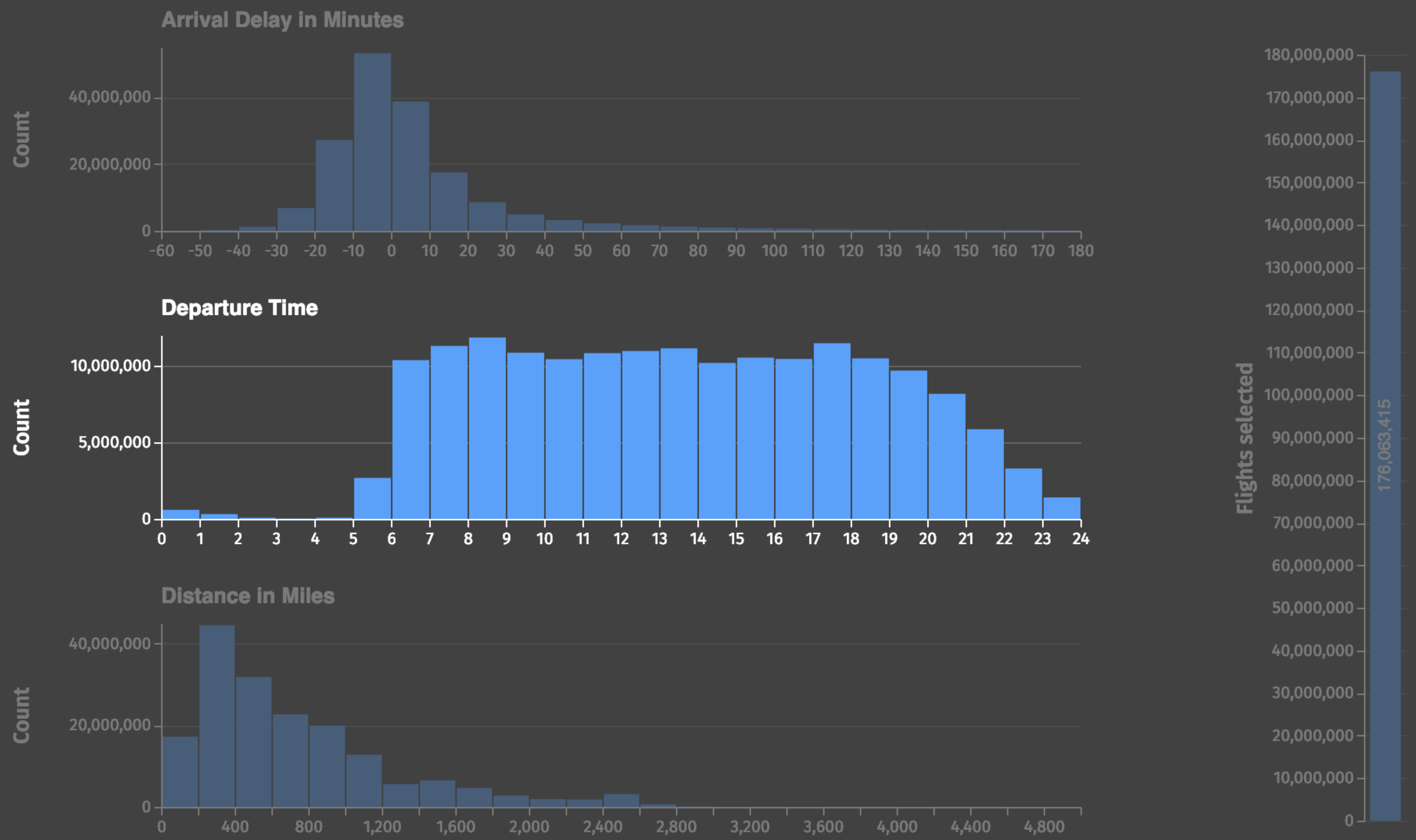
Falcon

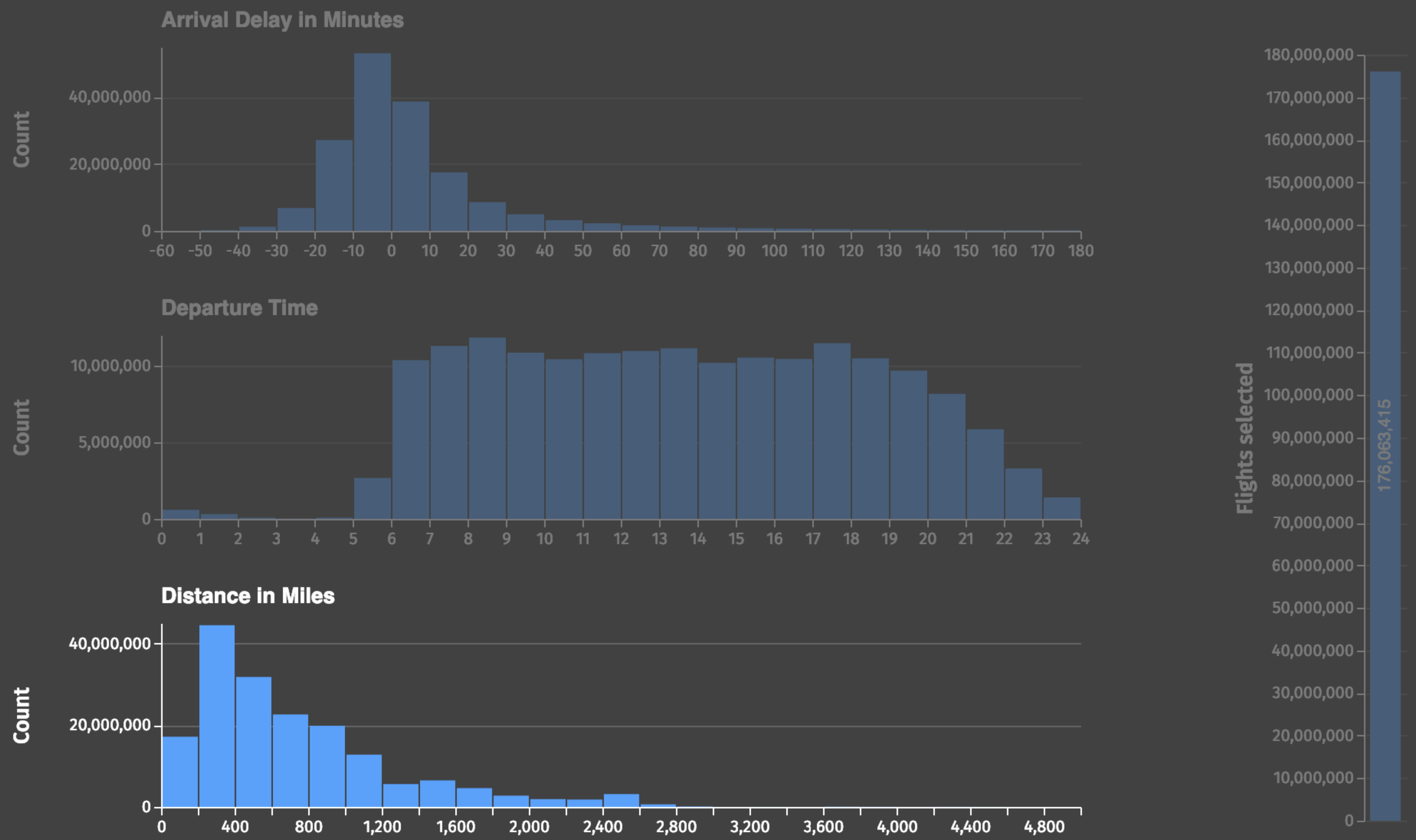
[Moritz, Howe, & Heer '19]

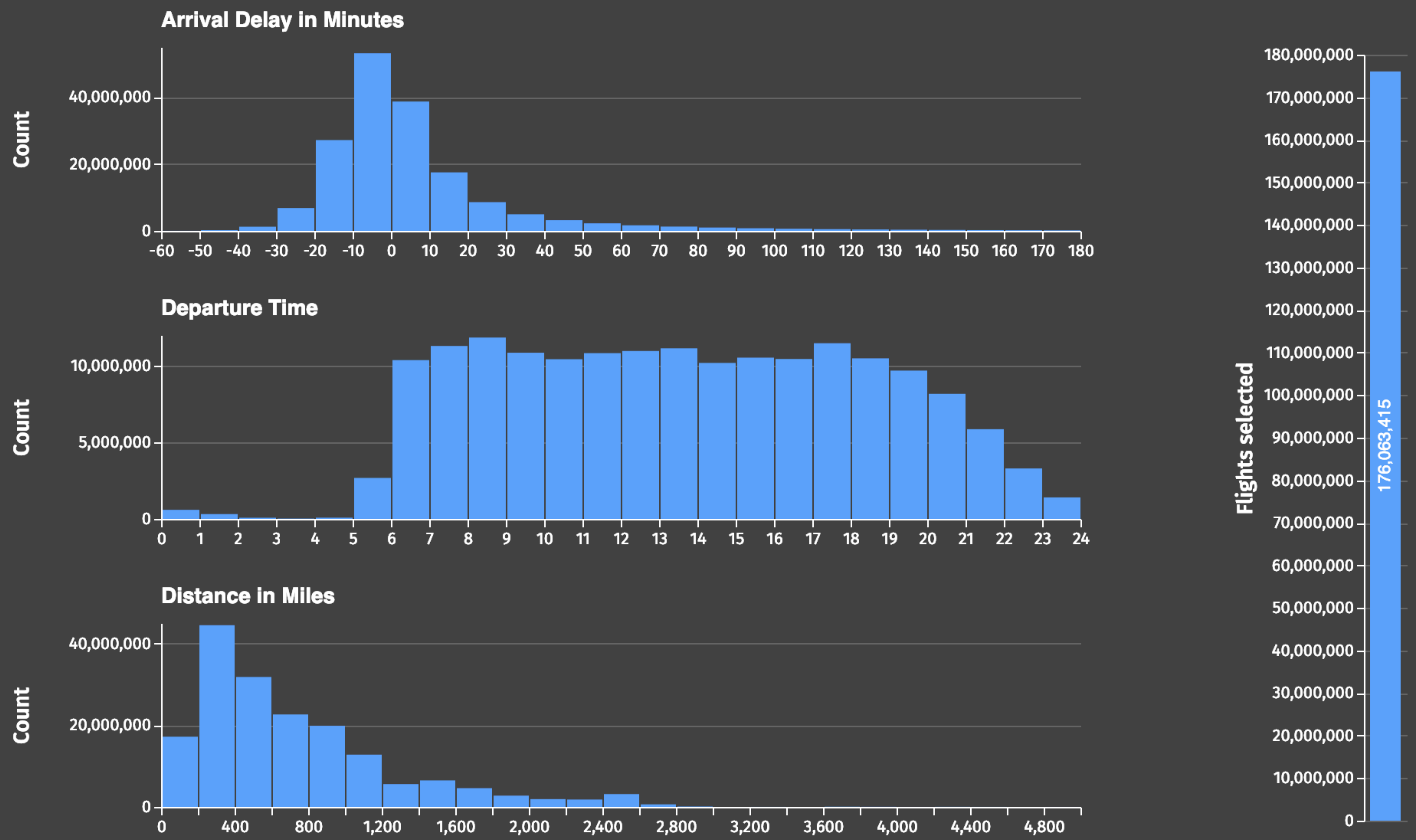
Strategies: Query Database, Client-Side Data Cubes, Pre-Fetching


Arrival Delay in Minutes
Count
40,000,000
20,000,000
0
-60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
Departure Time
Count
10,000,000
5,000,000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Distance in Miles
Count
40,000,000
20,000,000
0
0 400 800 1,200 1,600 2,000 2,400 2,800 3,200 3,600 4,000 4,400 4,800
Flights selected
180,000,000
170,000,000
160,000,000
150,000,000
140,000,000
130,000,000
120,000,000
110,000,000
100,000,000
90,000,000
80,000,000
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
0
176,063,415


Arrival Delay in Minutes
Count
40,000,000
20,000,000
0
-60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
Departure Time
Count
10,000,000
5,000,000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Distance in Miles
Count
40,000,000
20,000,000
0
0 400 800 1,200 1,600 2,000 2,400 2,800 3,200 3,600 4,000 4,400 4,800
Flights selected
176,063,415
180,000,000
170,000,000
160,000,000
150,000,000
140,000,000
130,000,000
120,000,000
110,000,000
100,000,000
90,000,000
80,000,000
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
0

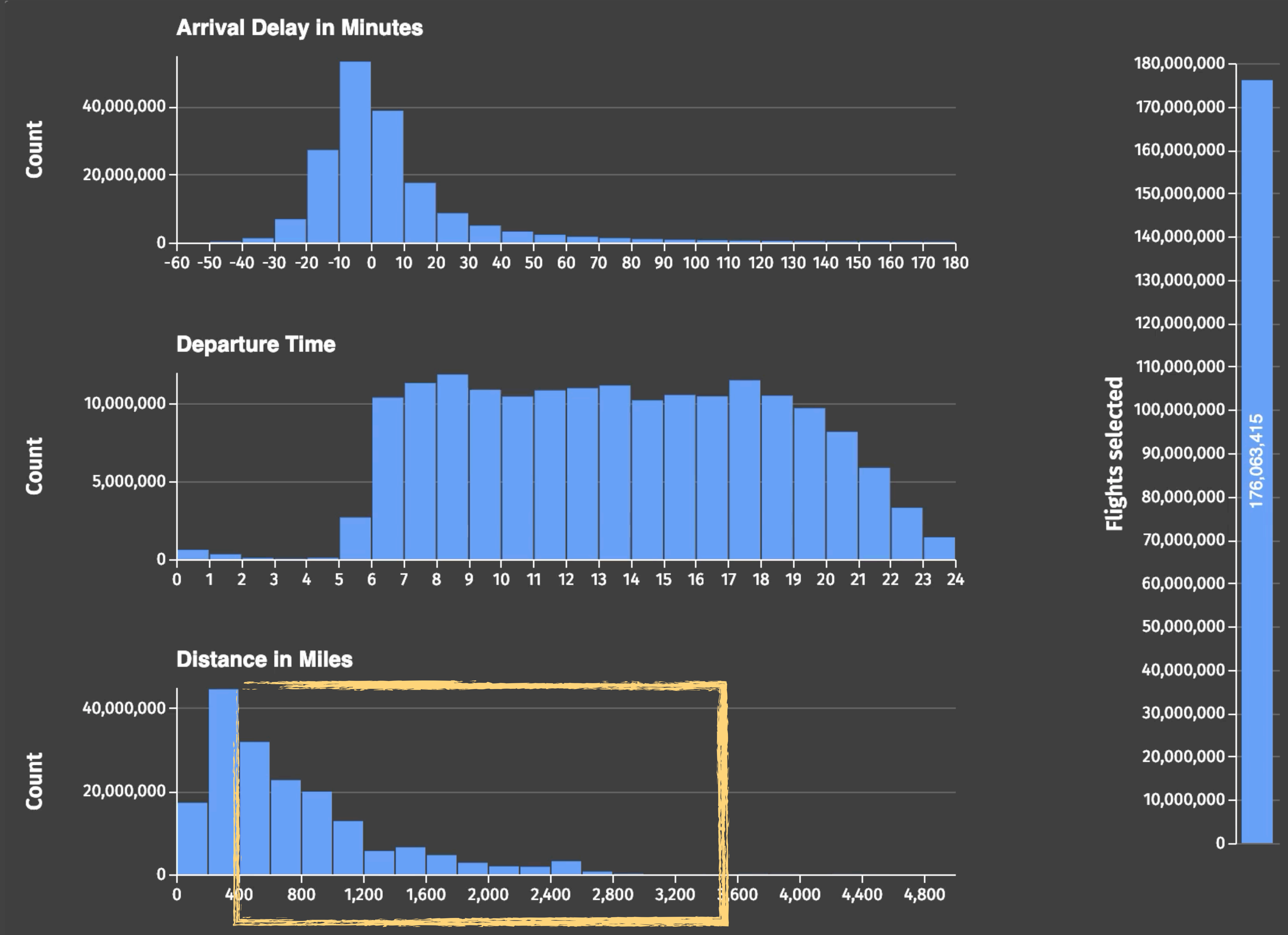

Arrival Delay in Minutes
Count
40,000,000
20,000,000
0
-60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
Departure Time
Count
10,000,000
5,000,000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Distance in Miles
Count
40,000,000
20,000,000
0
0 400 800 1,200 1,600 2,000 2,400 2,800 3,200 3,600 4,000 4,400 4,800
Flights selected
176,063,415
180,000,000
170,000,000
160,000,000
150,000,000
140,000,000
130,000,000
120,000,000
110,000,000
100,000,000
90,000,000
80,000,000
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
0


Arrival Delay in Minutes
Count
Departure Time
Count
Distance in Miles
Count
Flights selected
176,063,415

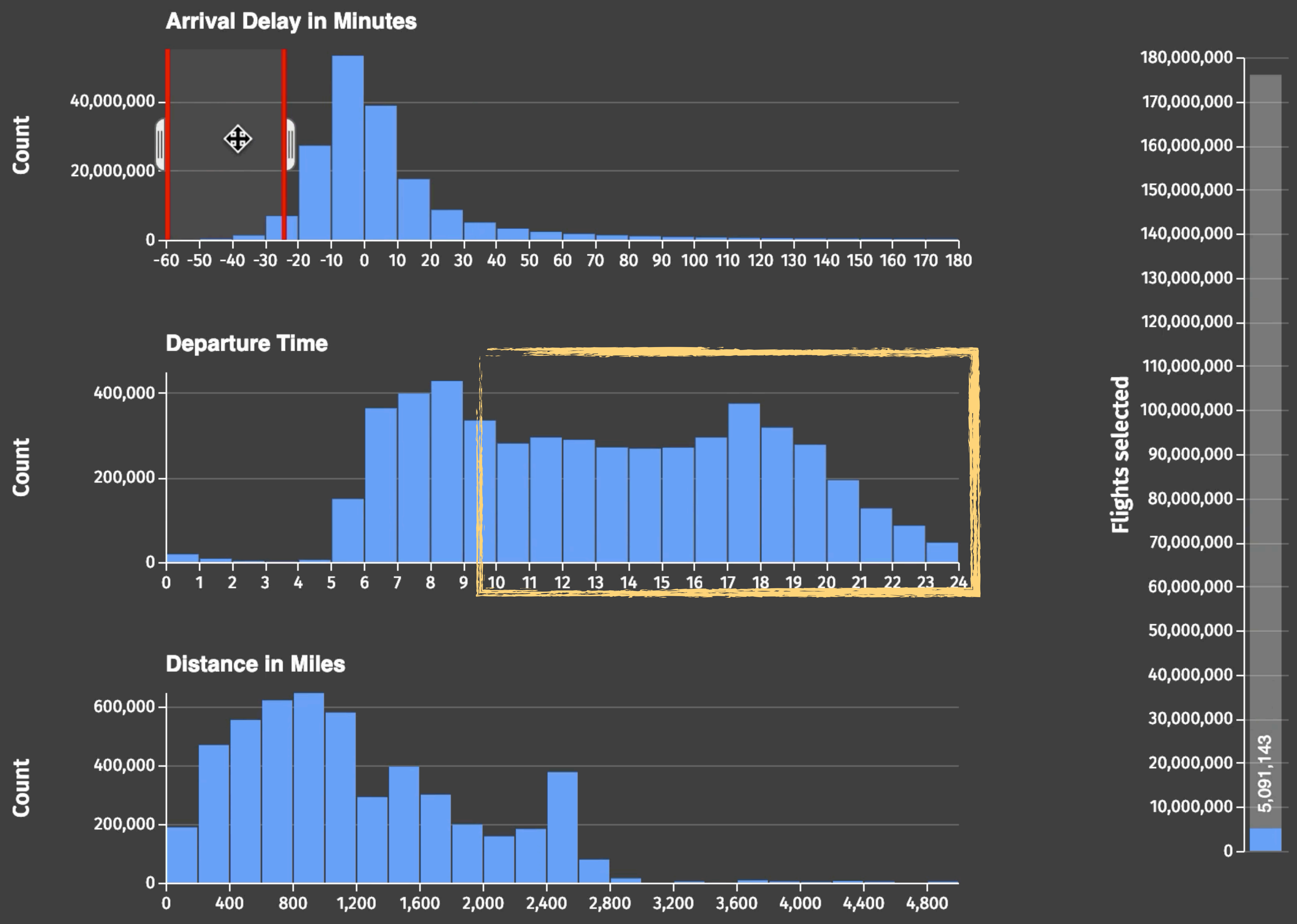

Arrival Delay in Minutes
Count
40,000,000
20,000,000
0
-60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
Departure Time
Count
10,000,000
5,000,000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Distance in Miles
Count
40,000,000
20,000,000
0
0 400 800 1,200 1,600 2,000 2,400 2,800 3,200 3,600 4,000 4,400 4,800
Flights selected
176,063,415
180,000,000
170,000,000
160,000,000
150,000,000
140,000,000
130,000,000
120,000,000
110,000,000
100,000,000
90,000,000
80,000,000
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
0

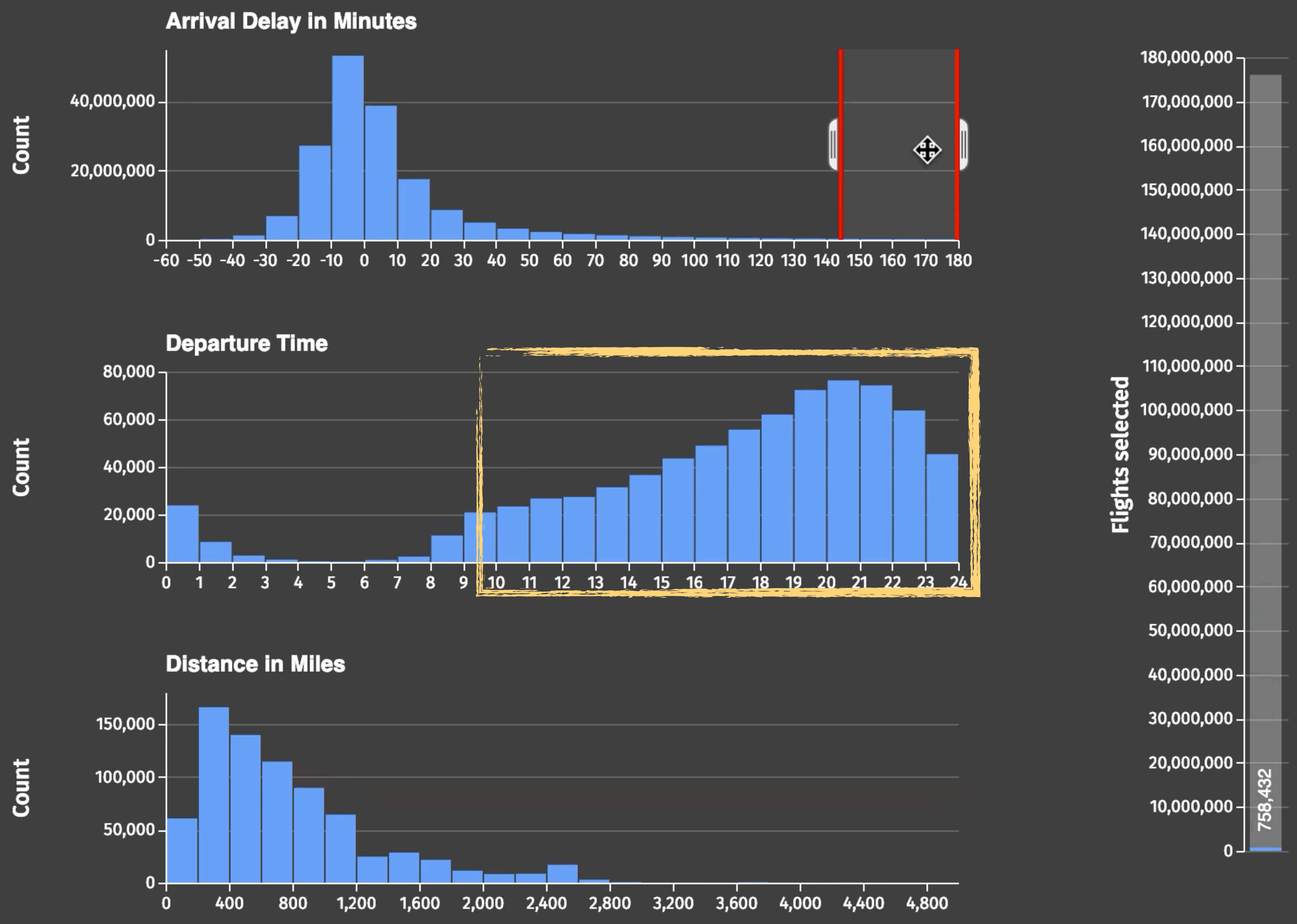
Falcon

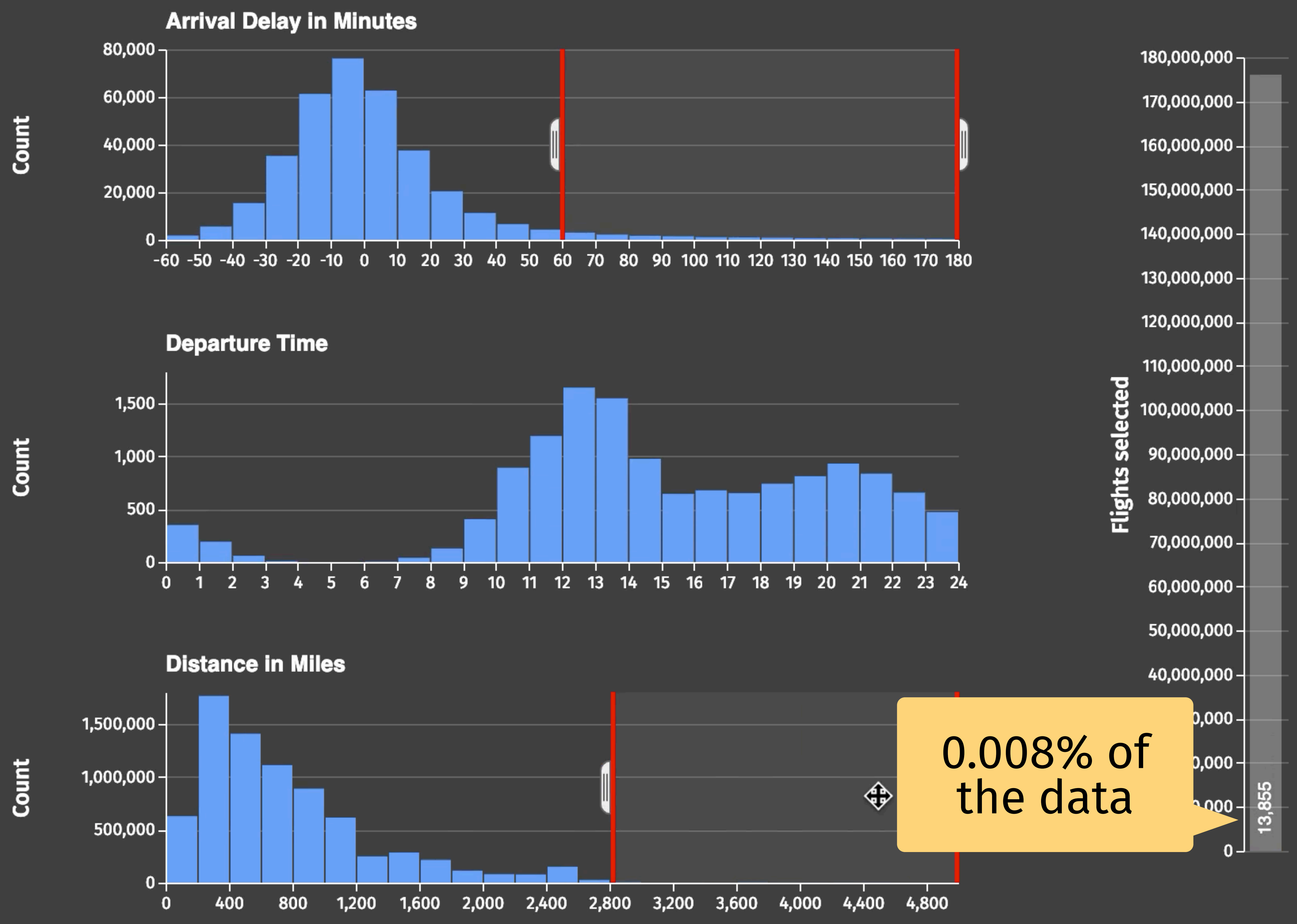
uwdata.github.io/falcon



Arrival Delay in Minutes
Count
40,000,000
20,000,000
0
-60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
Departure Time
Count
10,000,000
5,000,000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Distance in Miles
Count
40,000,000
20,000,000
0
0 400 800 1,200 1,600 2,000 2,400 2,800 3,200 4,000 4,400 4,800
Flights selected
176,063,415
180,000,000
170,000,000
160,000,000
150,000,000
140,000,000
130,000,000
120,000,000
110,000,000
100,000,000
90,000,000
80,000,000
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
0

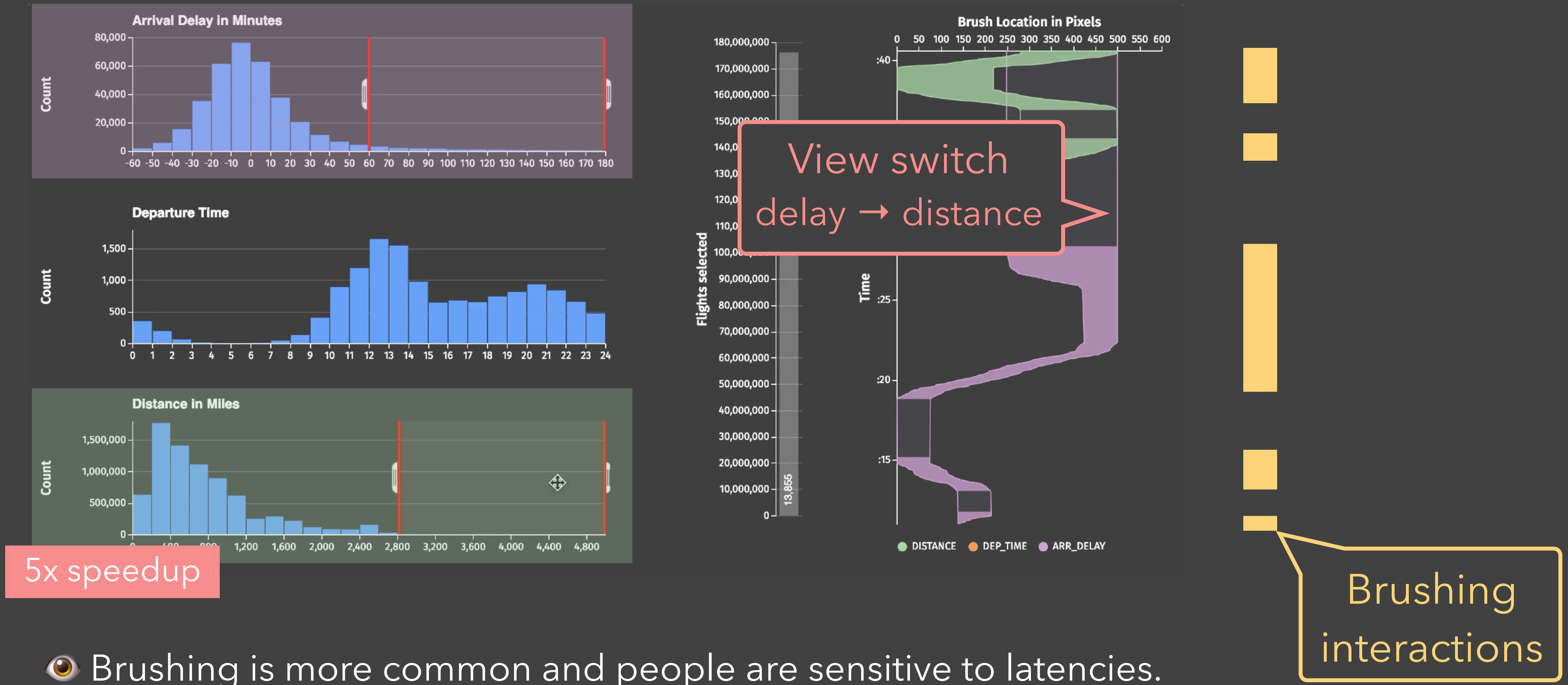

Arrival Delay in Minutes
Count
40,000,000
20,000,000
0
-60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
Departure Time
Count
400,000
200,000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Distance in Miles
Count
600,000
400,000
200,000
0
0 400 800 1,200 1,600 2,000 2,400 2,800 3,200 3,600 4,000 4,400 4,800
Flights selected
180,000,000
170,000,000
160,000,000
150,000,000
140,000,000
130,000,000
120,000,000
110,000,000
100,000,000
90,000,000
80,000,000
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
0
5,091,143


Arrival Delay in Minutes
Count
Departure Time
Count
Distance in Miles
Count
Flights selected
758,432


Arrival Delay in Minutes
Departure Time
Distance in Miles
Count
Flights selected
13,855
0.008% of the data

How does Falcon support fine-grained real-time interaction?

Falcon Interaction Log



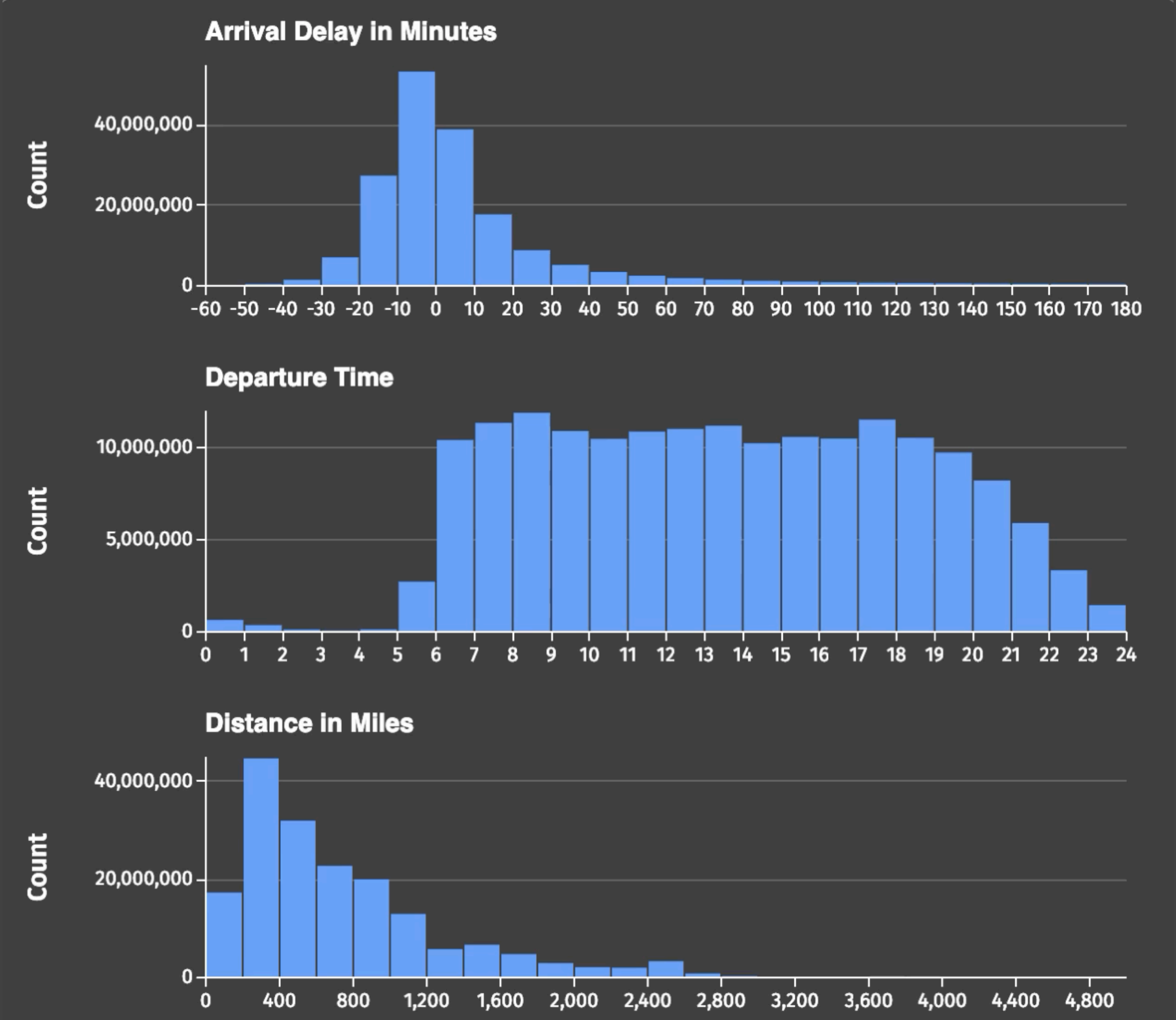
👁️ Brushing is more common and people are sensitive to latencies.

💡 Prioritize brushing latency over view switching latency.

Key Idea:

User-centered prefetching and indexing to support all brushing interactions with one view.

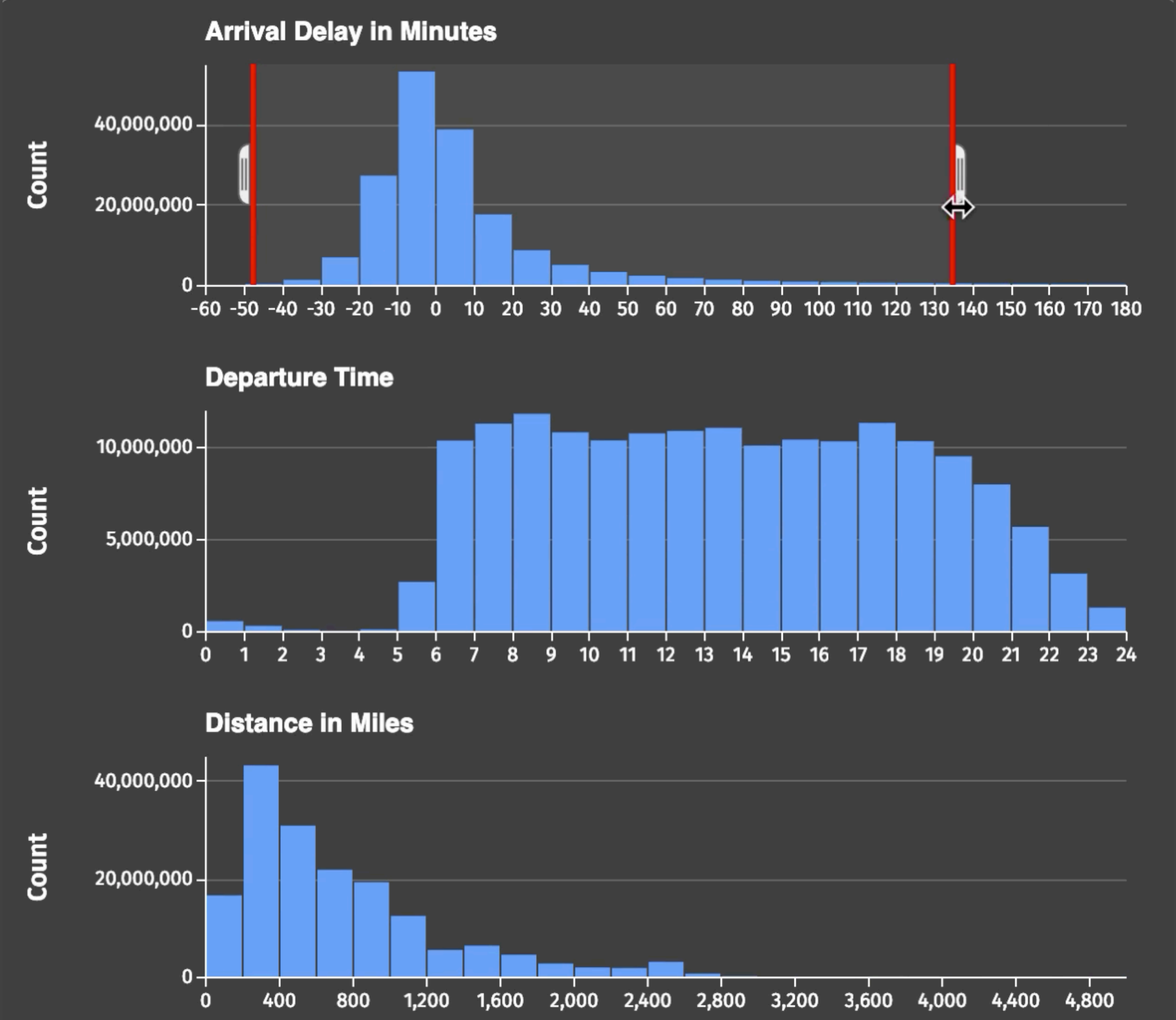
Re-compute if the user switches the view.



brushes in the precomputed view

serves requests from a data cube

Data Cube. Gray et al. *1997.*



brushes in the precomputed view

serves requests from a data cube

Data Cube. Gray et al. *1997.*

interacts with a new view

query for new data cubes

Constant data & time.
Client only.

brushes in the precomputed view

serves requests from a data cube

Data Cube. Gray et al. *1997*.

💡 Aggregation decouples interactions from queries over the raw data.

Requires one pass
over the data.

interacts with a new view

query for new data cubes

💡 View switches are **rare** and users are **not as latency sensitive** with them.

Visualization Systems that Leverage Data Cubes

Problem: The full data cube has size $\prod_i b_i$ where b_i is the number of bins in dimension i.

Visualization Systems that Leverage Data Cubes

Problem: The full data cube has size $\prod_i b_i$ where b_i is the number of bins in dimension i.

Nanocubes. Lins et al. *InfoVis 2013*.

Specialized hierarchical data structure for sparse cubes.
Cubes are still too large for the browser. Hours of build time.

imMens. Liu et al. *Eurovis 2013*.

Dense cube. Decomposed into overlapping cubes.
One cube per pairwise interactions. One brush. Brushing at bin resolution. Hours of build time.

Visualization Systems that Leverage Data Cubes

Problem: The full data cube has size $\prod_i b_i$ where b_i is the number of bins in dimension i.

Nanocubes. Lins et al. *InfoVis 2013*.

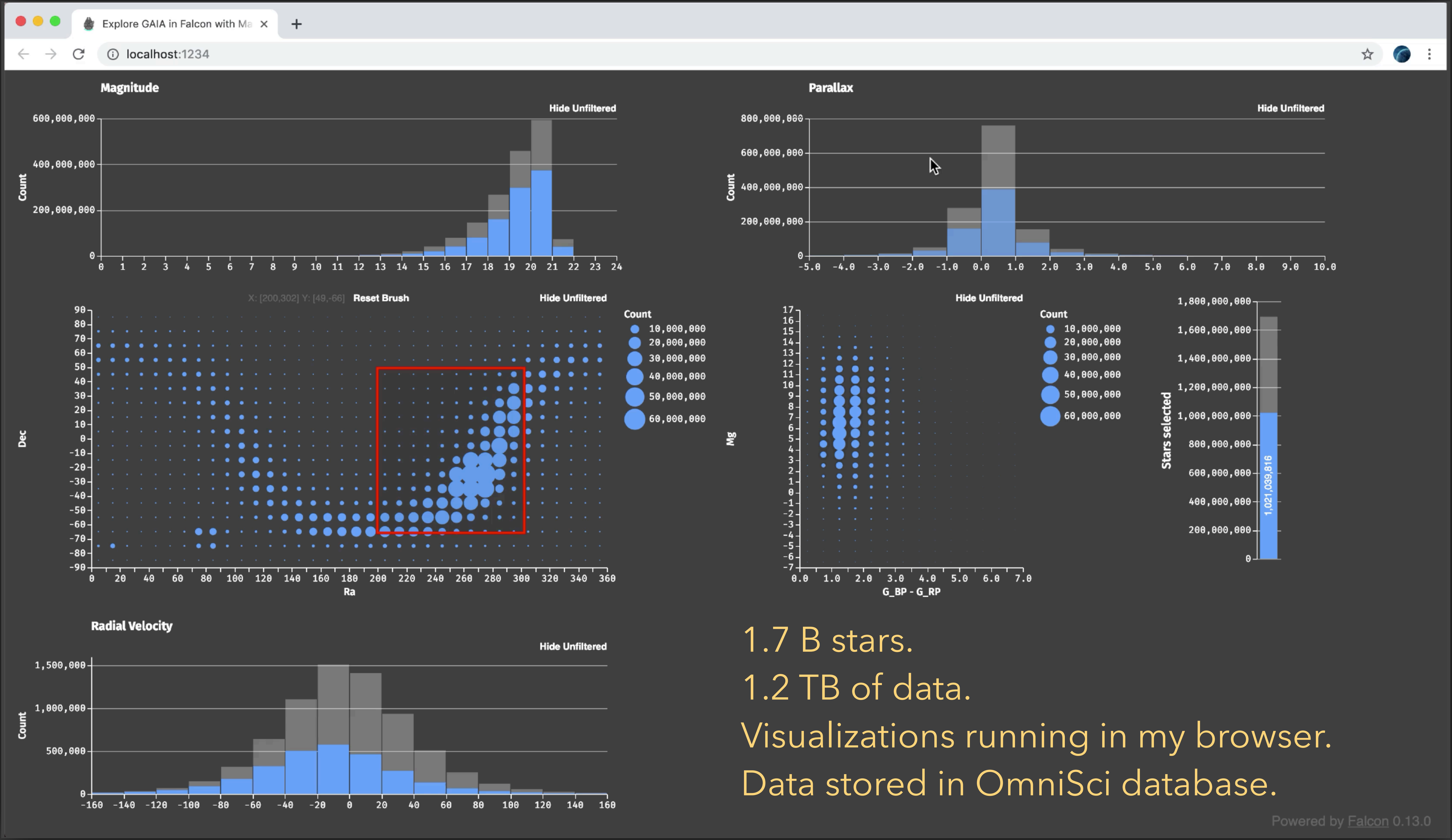
Specialized hierarchical data structure for sparse cubes.
Cubes are still too large for the browser. Hours of build time.

imMens. Liu et al. *Eurovis 2013*.

Dense cube. Decomposed into overlapping cubes.
One cube per pairwise interactions. One brush. Brushing at bin resolution. Hours of build time.

Falcon. Moritz et al. *CHI 2019*.

Small cubes for single active view.
Small cubes are built on the fly. View switches require new cube.


Explore GAIA in Falcon with Ma
localhost:1234
Magnitude
Hide Unfiltered
Count
Parallax
Hide Unfiltered
Reset Brush
Hide Unfiltered
Count
10,000,000
20,000,000
30,000,000
40,000,000
50,000,000
60,000,000
Dec
Ra
Hide Unfiltered
Count
10,000,000
20,000,000
30,000,000
40,000,000
50,000,000
60,000,000
Mg
G_BP - G_RP
Stars selected
1,021,039,816
Radial Velocity
Hide Unfiltered
Count
1.7 B stars.
1.2 TB of data.
Visualizations running in my browser.
Data stored in OmniSci database.
Powered by Falcon 0.13.0

"With Falcon it feels like I'm really interacting with my data."

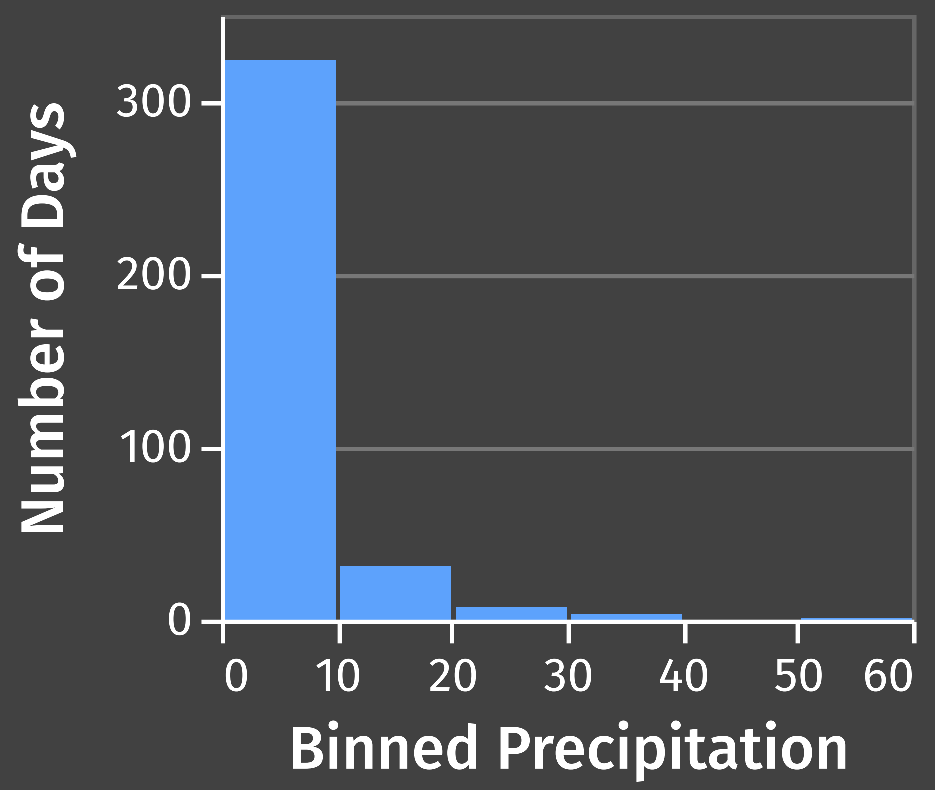
Data Platform Engineer at Stitch Fix

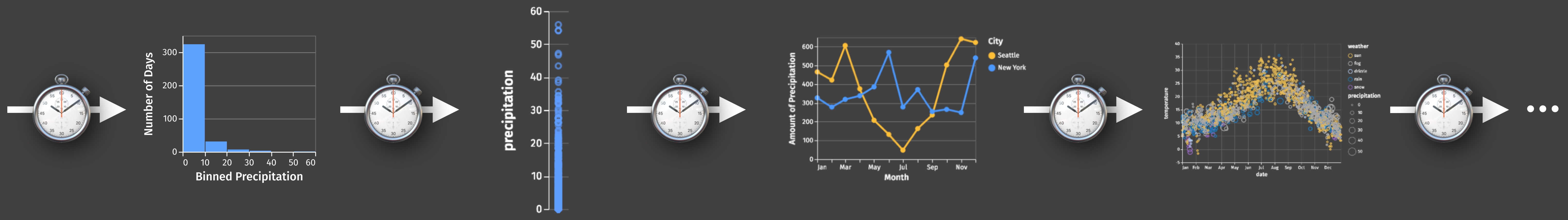
Trust, but Verify: Optimistic Vis

[Moritz, Fisher, Ding & Wang '17]

Strategies: Query Database, Approximation

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


Number of Days
300
200
100
0
0
10
20
30
40
50
60
Binned Precipitation



Latencies reduce engagement
and lead to fewer observations.

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

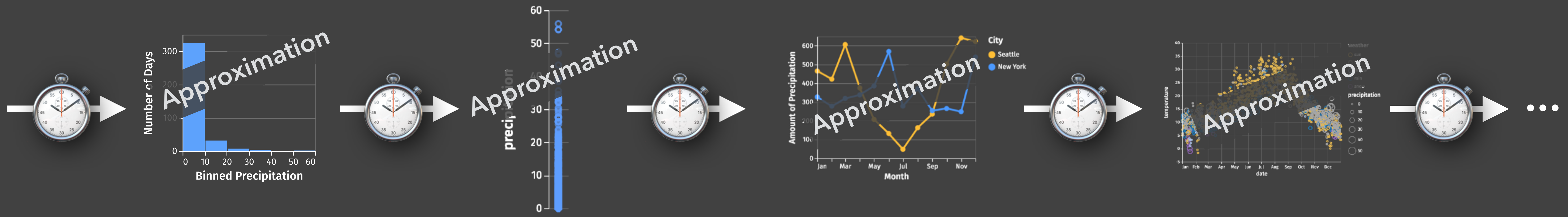
Small chance of error

Small chance of error

Small chance of error

Small chance of error

Very likely to have at least one 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.

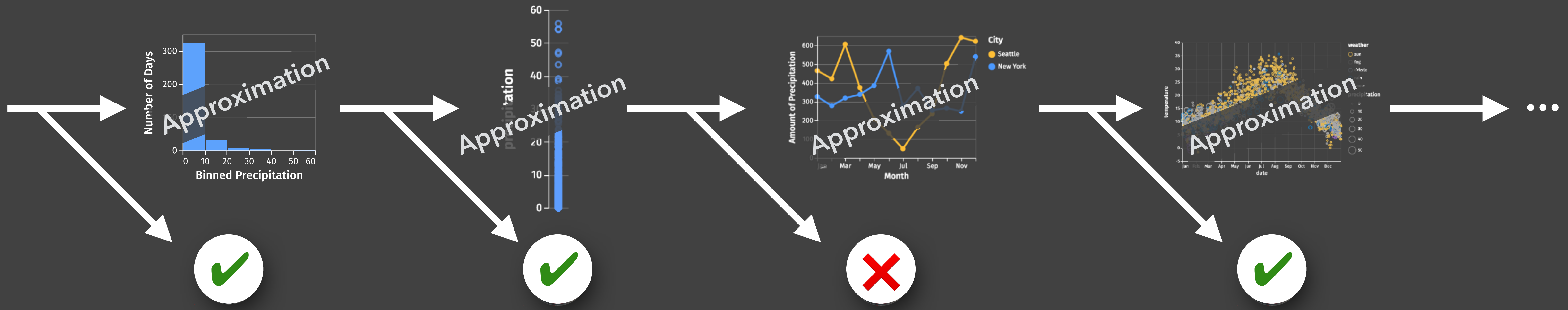
Optimistic Visualization

Trust but Verify

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

Optimistic Visualization

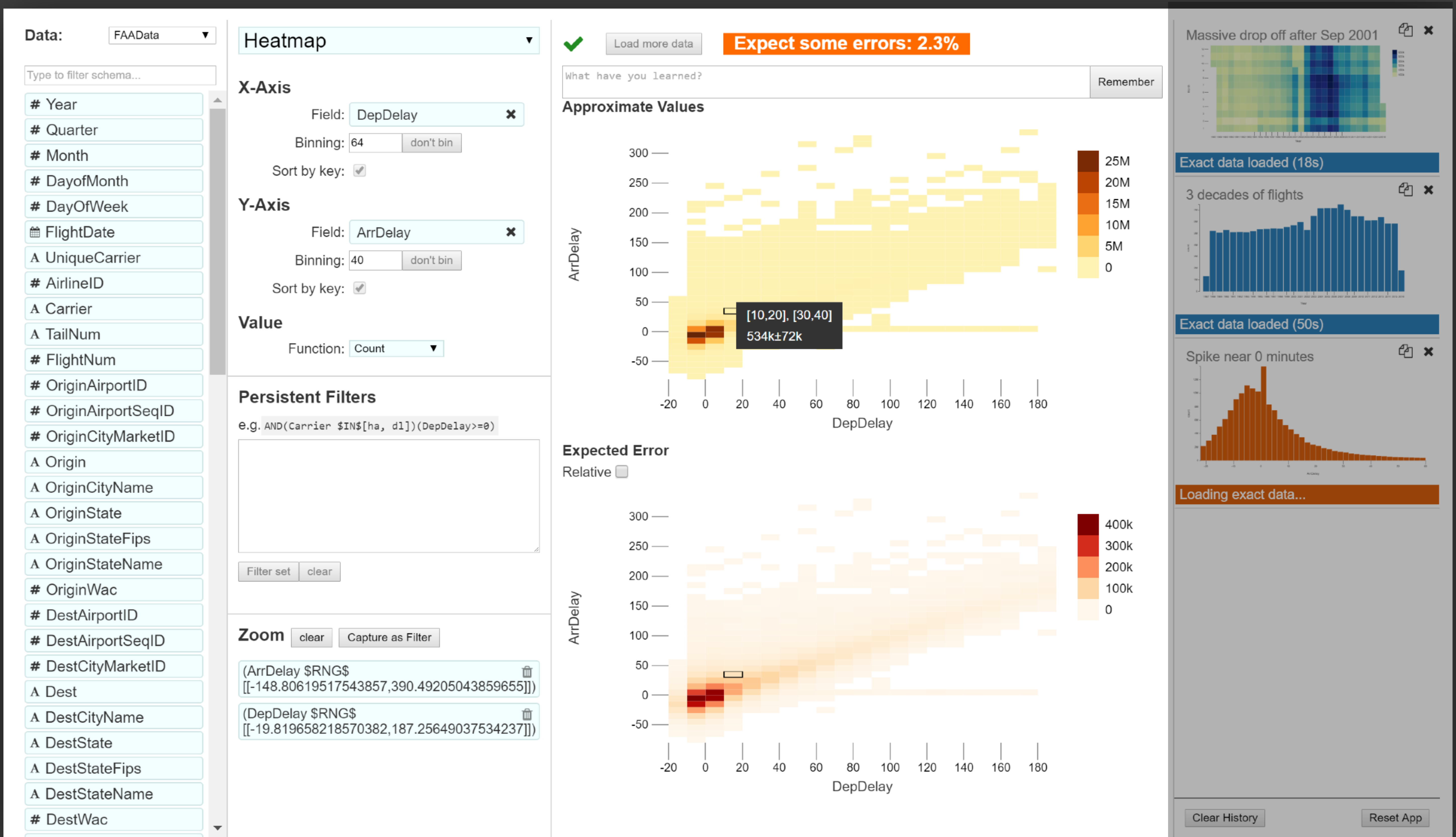
Trust but Verify. Moritz et al. *CHI 2017*.



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.

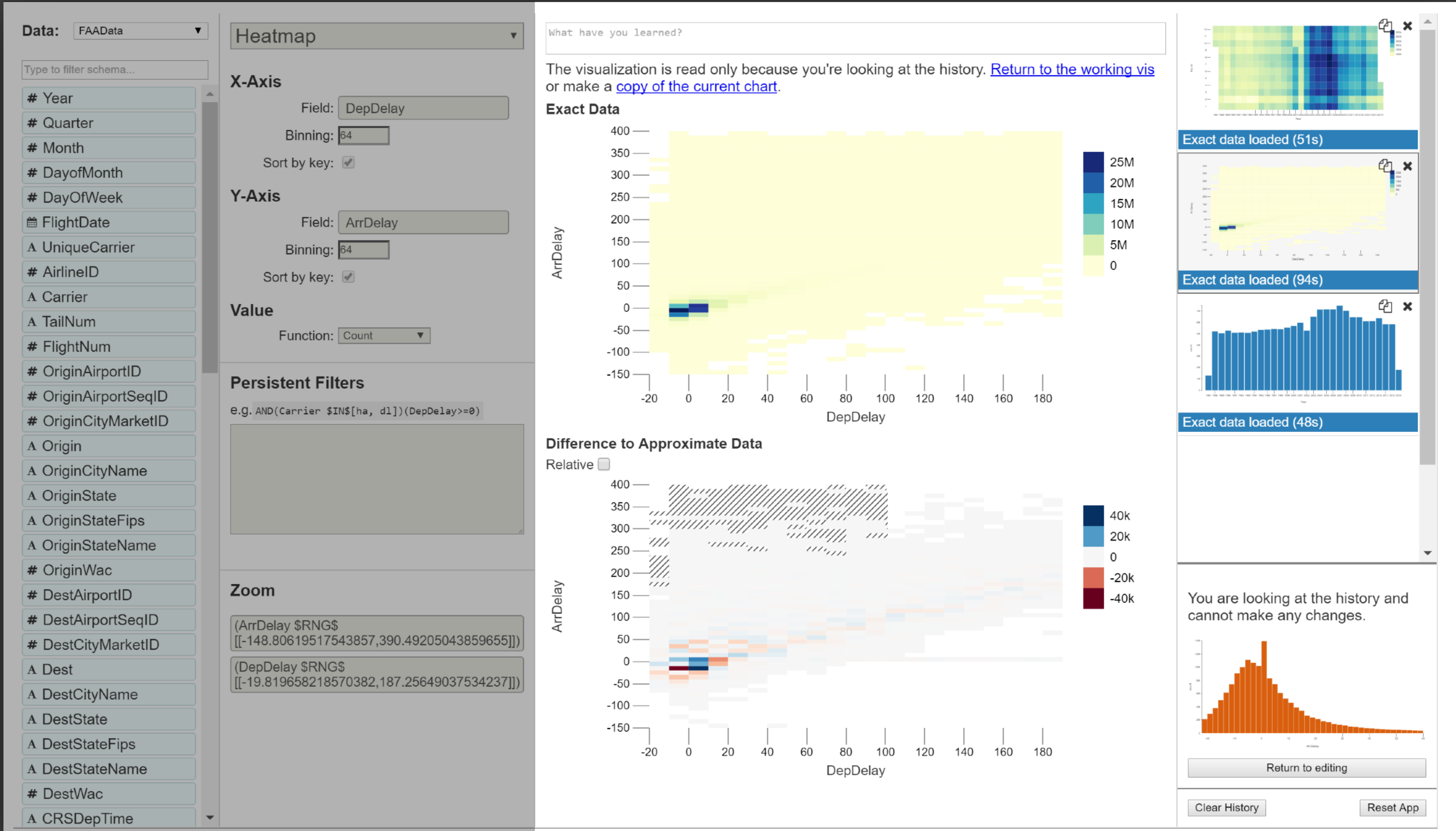
Pangloss Implements Optimistic Visualization



Pangloss Visualizes Uncertainty

Pangloss shows a History of Previous Charts

In Pangloss, Analysts can 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.

REMINDER

FP Prototypes Due Tonight (12/1)

Submit URL on Canvas!