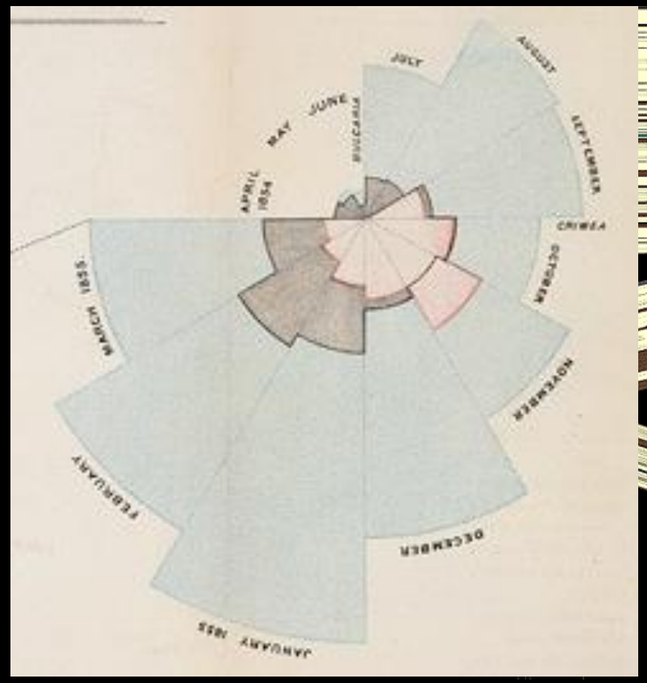
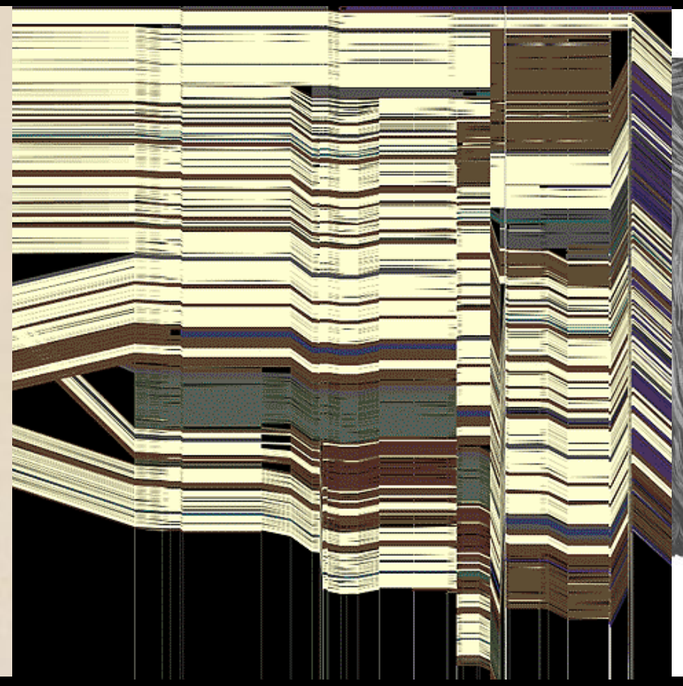
CSE 442 - Data Visualization

Scalable Visualization





Leilani Battle University of Washington

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

Topics

Varieties of “big data”

Visualizing Large Datasets

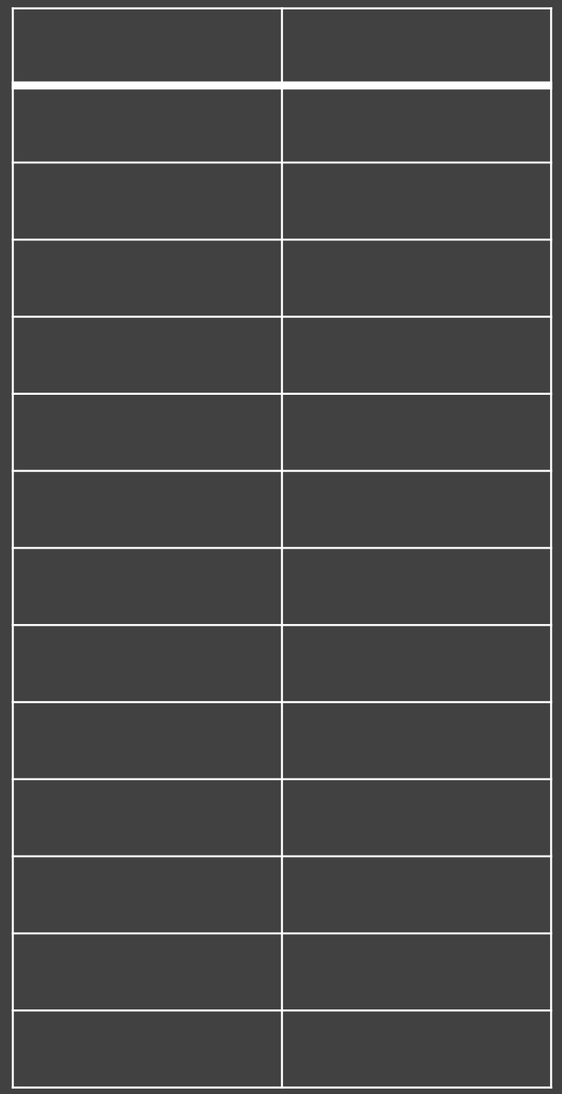
- Design Subtleties
- Examples

Enabling Real-Time Interaction

- imMens
- ForeCache
- Trust, but Verify: Optimistic Vis

Varieties of “big data”…

Many Records

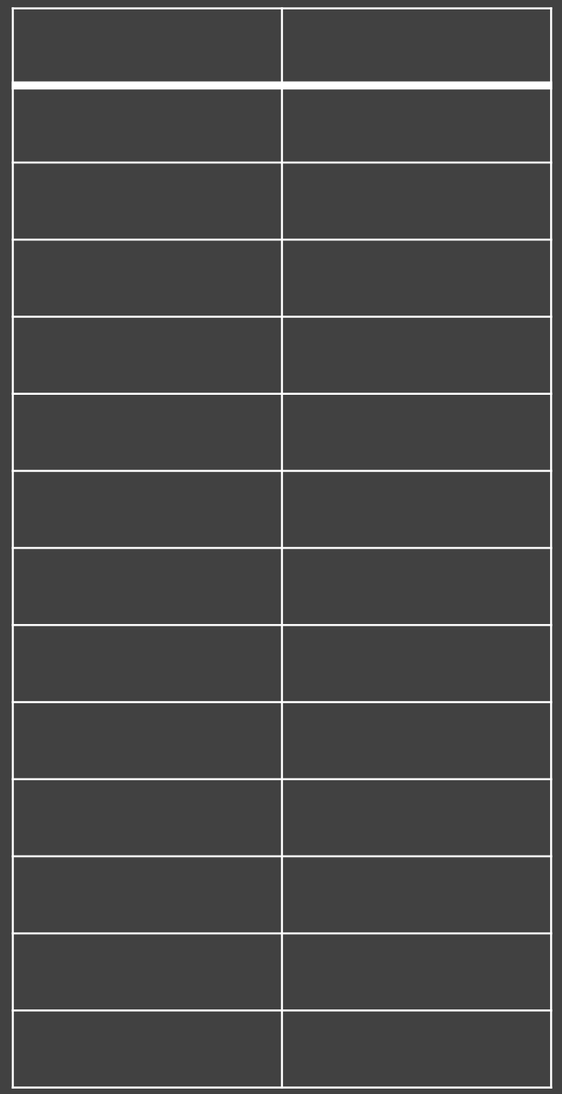


Large DBs have petabytes or more
(but median DB still fits in RAM!)

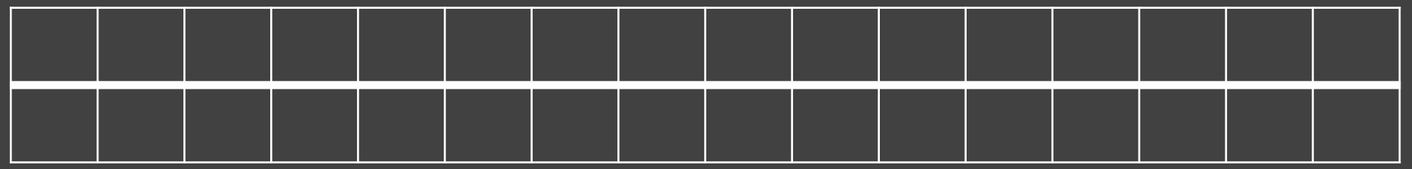
Affects system *and* perceptual scalability

How to manage?
- Parallel data processing
- Reduction: Filter, aggregate, sample

Many Records



Many Columns



Lots of variables (100s-1000s…)

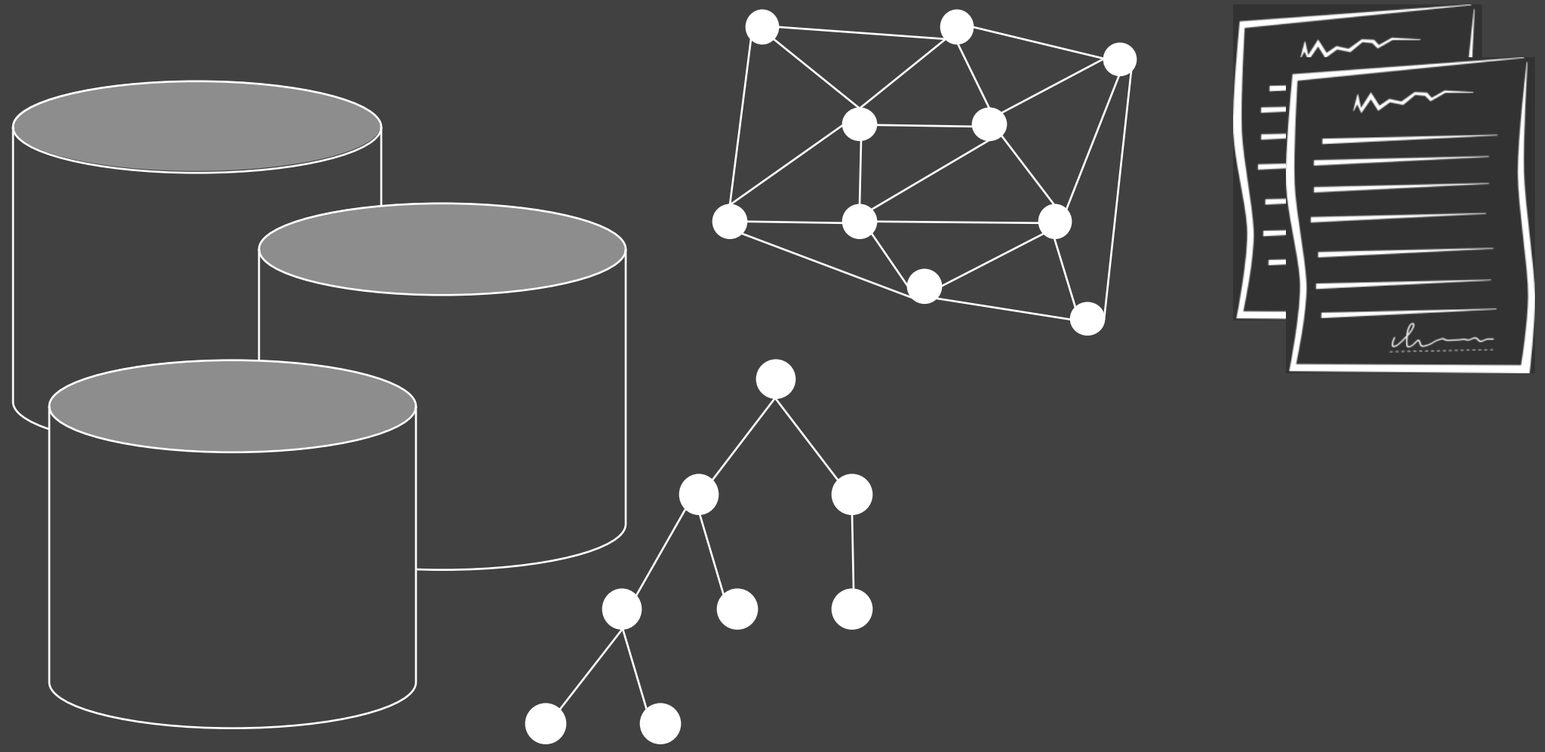
- Select relevant subset
- Dimensionality reduction
- Statistical methods can suggest and order related variables

Requires human judgment

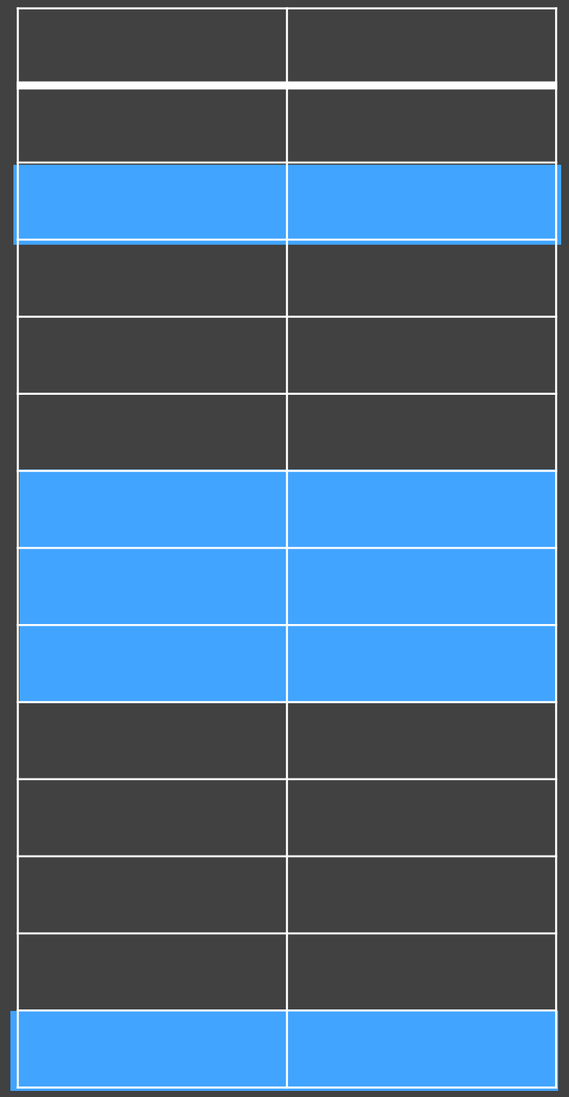
Many Records

Many Columns

Many Sources & Structures

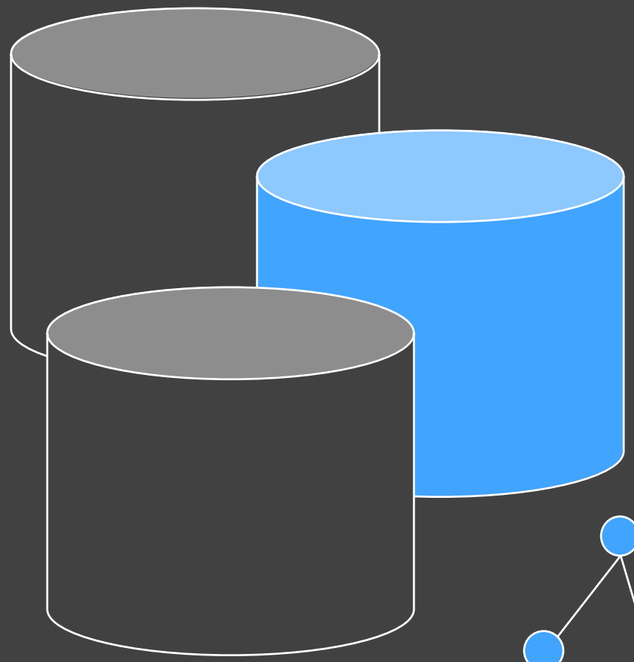


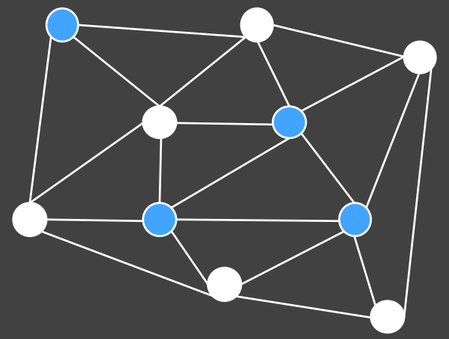
Many Records

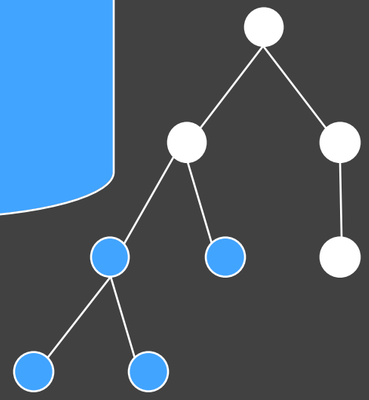


Many Columns

Many Sources & Structures

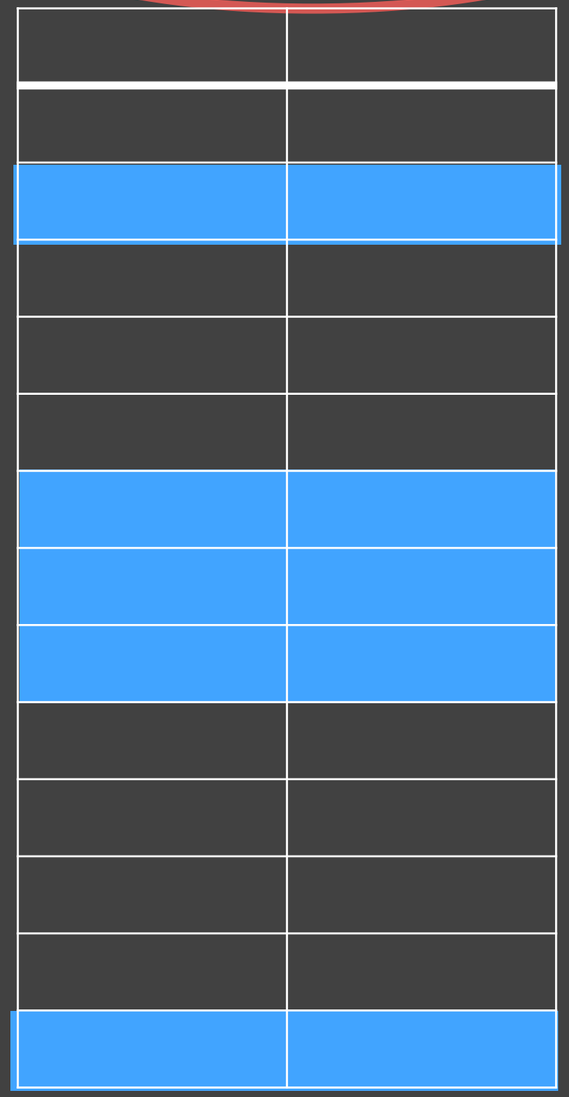






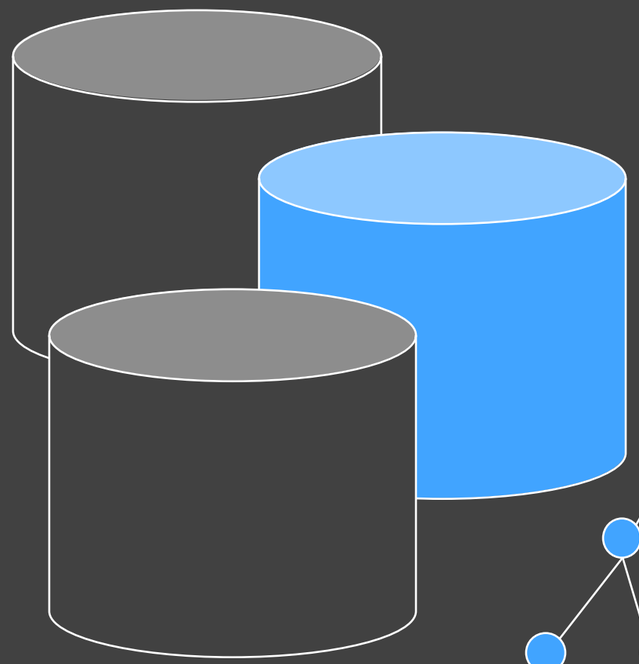
Many Updates

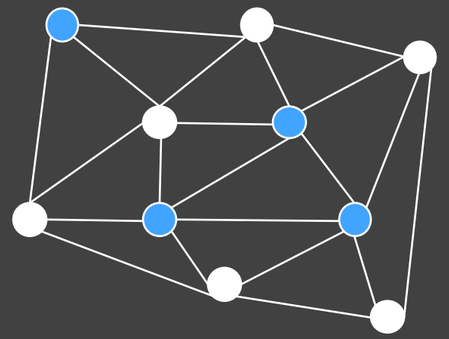
Many Records

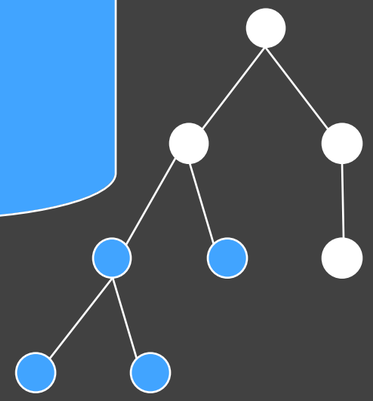


Many Columns

Many Sources & Structures







Many Updates

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.

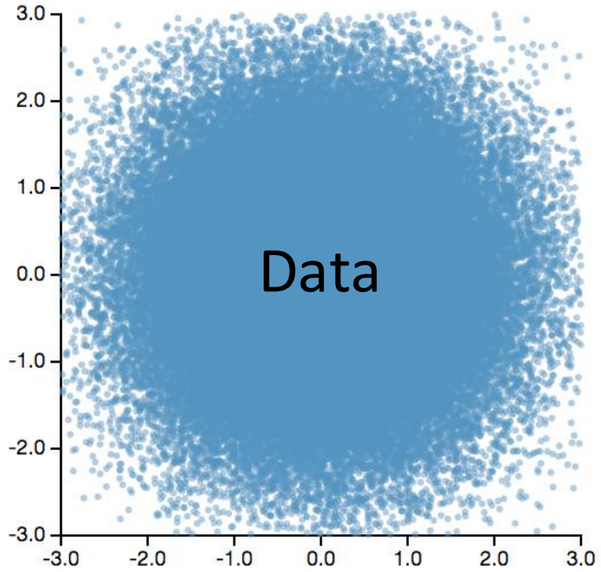
1. Visualizing Large Datasets

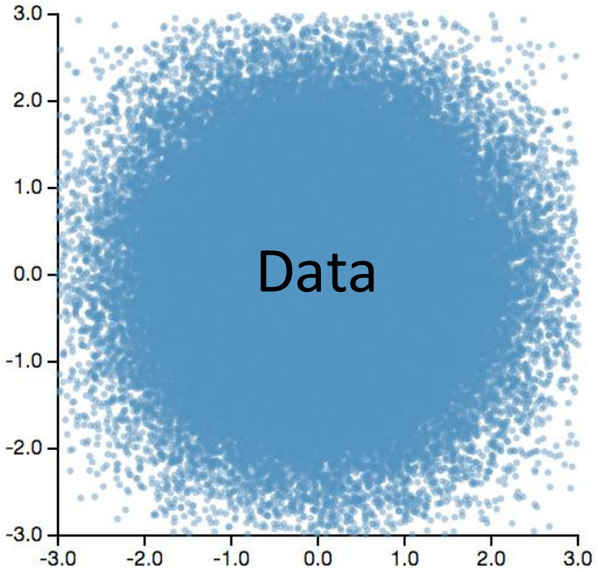
Challenge of Perceptual Scalability

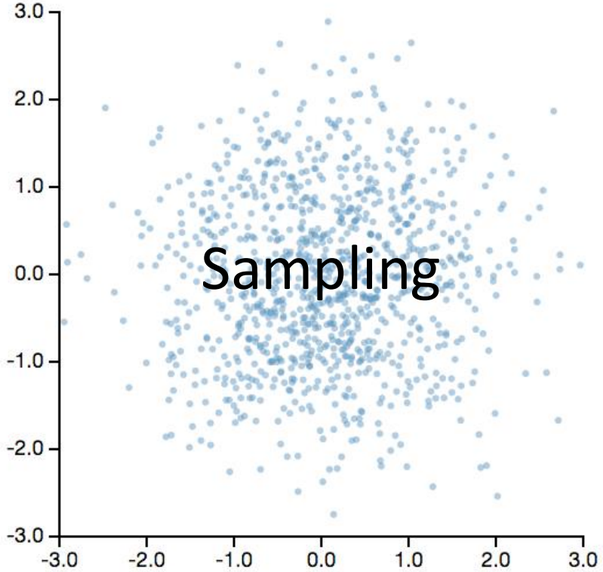
We cannot fit millions+ of data records into a typical image resolution (for example, 500x500 image → only 250,000 pixels).

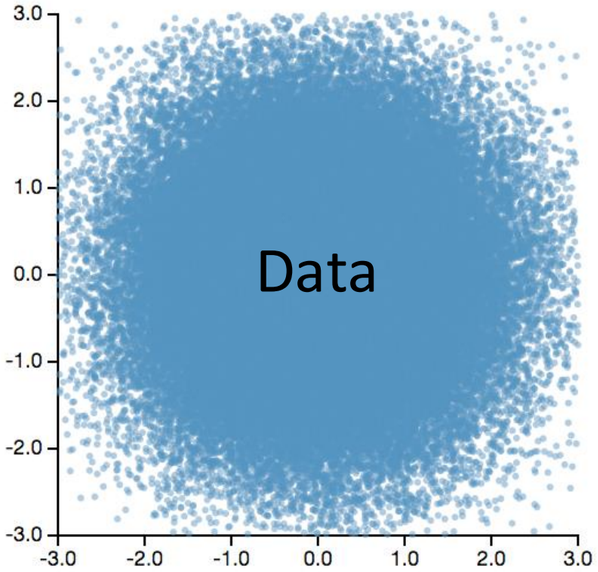
Even if we could, the resulting image would be a wall of ink!

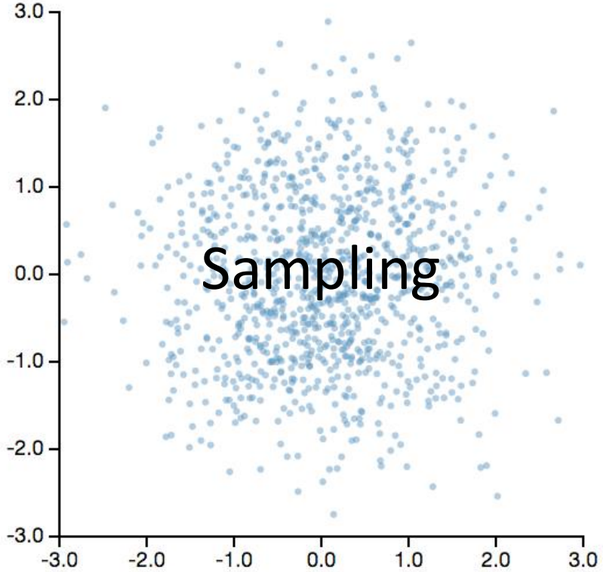
How can we visualize the key properties of a large dataset without rendering every data record?

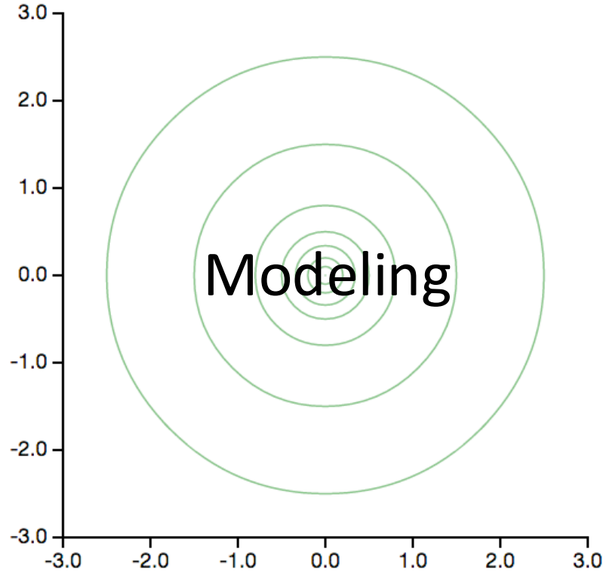

Data
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

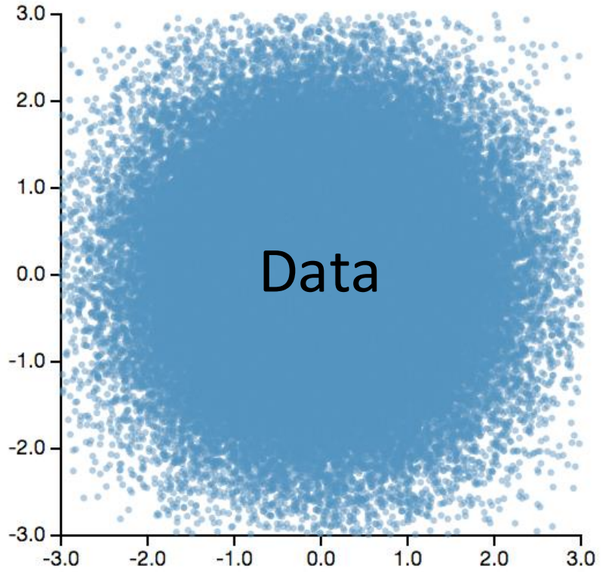

Data
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

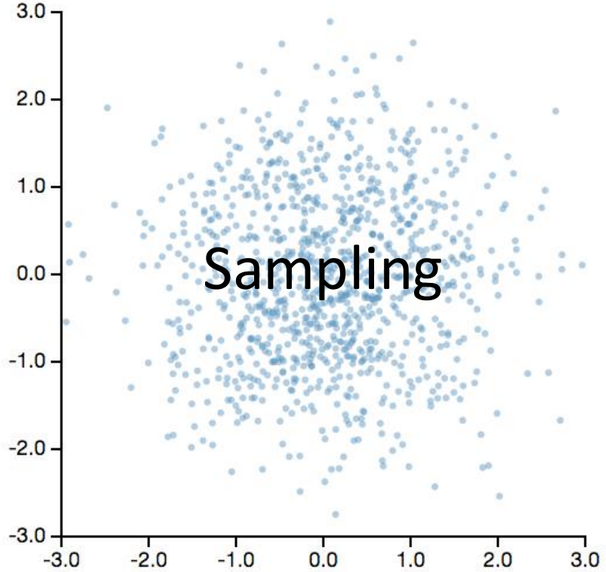

Sampling
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

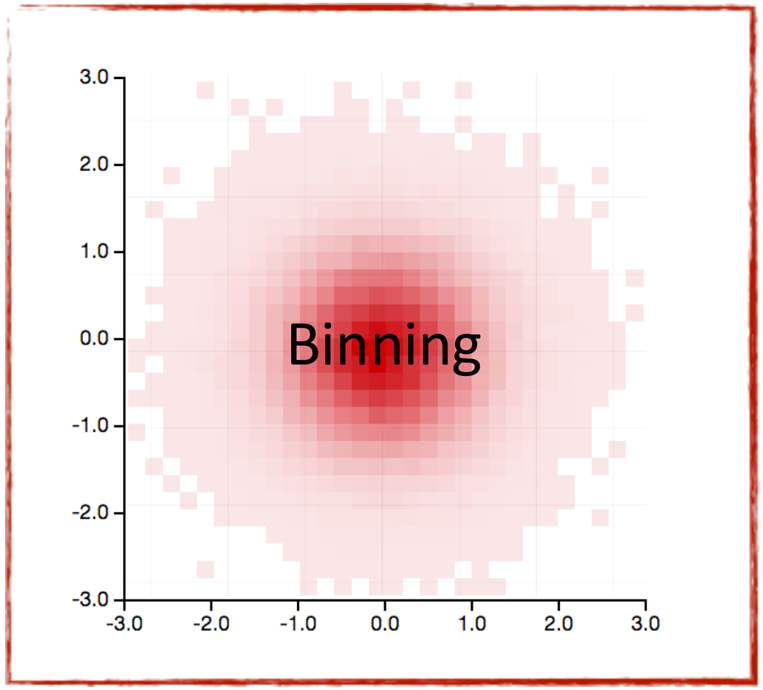

Data
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0

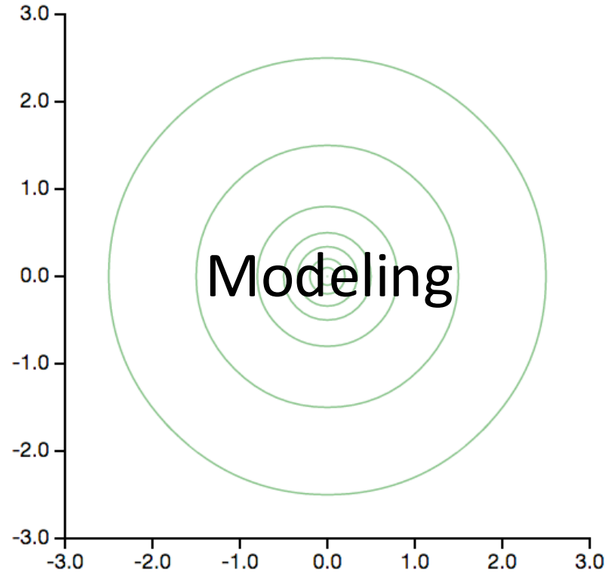

Sampling
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0


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

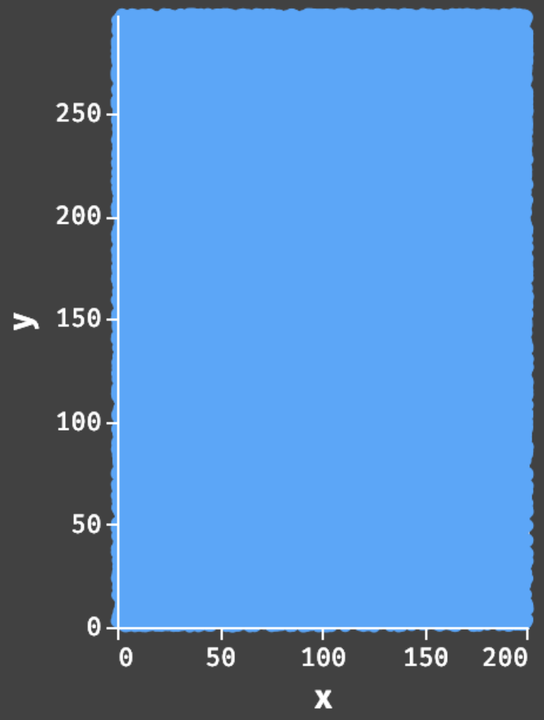

Data
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0


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

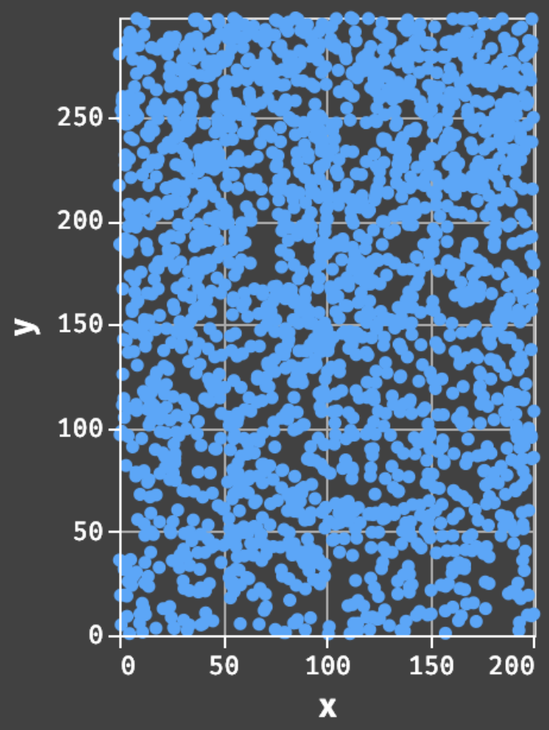

Binning
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-3.0
-2.0
-1.0
0.0
1.0
2.0
3.0


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

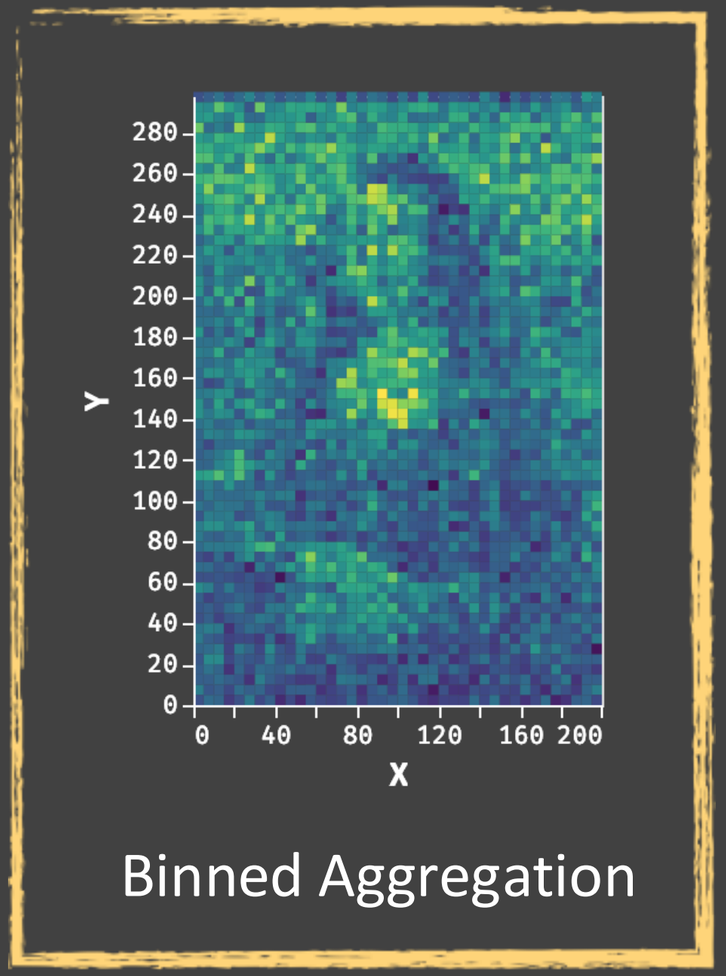
How to **Visualize** a Billion+ Records



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.

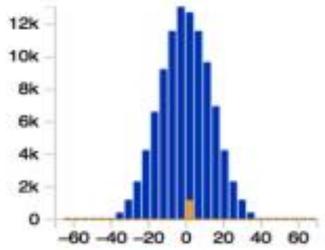
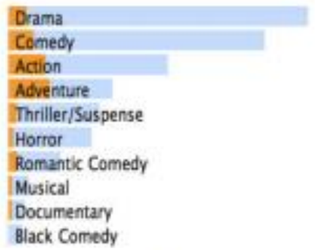
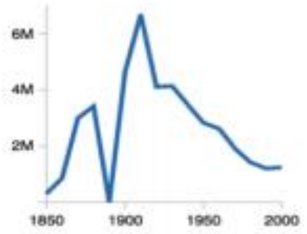
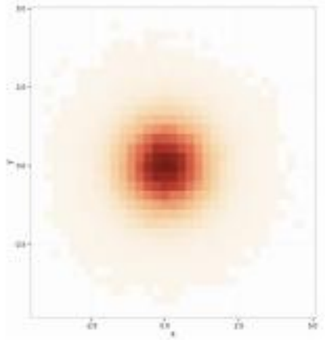
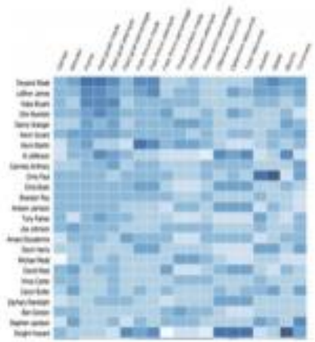
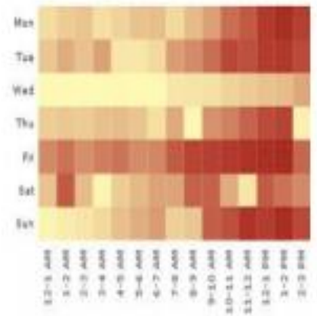
Geo: Bin x, y coordinates *after* cartographic projection

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

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

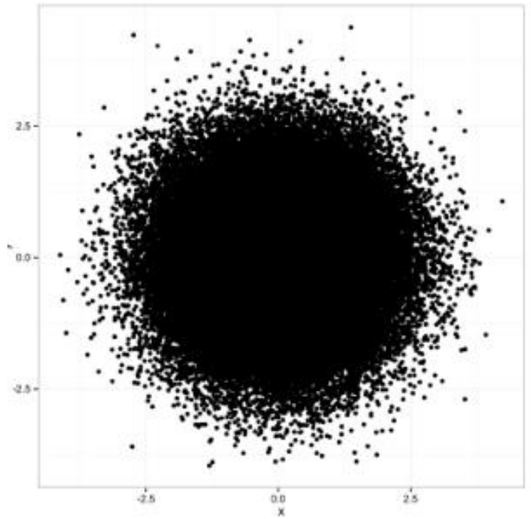
4. Plot Visualize the aggregate values

Binned Plots by Data Type

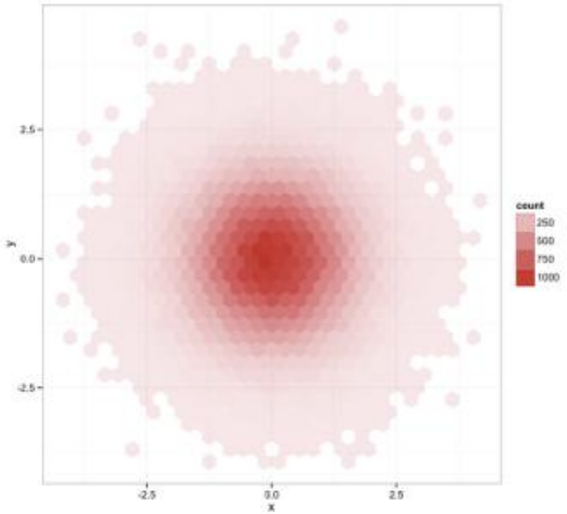
	Numeric	Ordinal	Temporal	Geographic
1D	 Histogram	 Bar Chart	 Line Graph / Area Chart	 Choropleth Map
2D	 Binned Scatter Plot	 Heatmap	 Temporal Heatmap	 Geographic Heatmap

Design Subtleties…

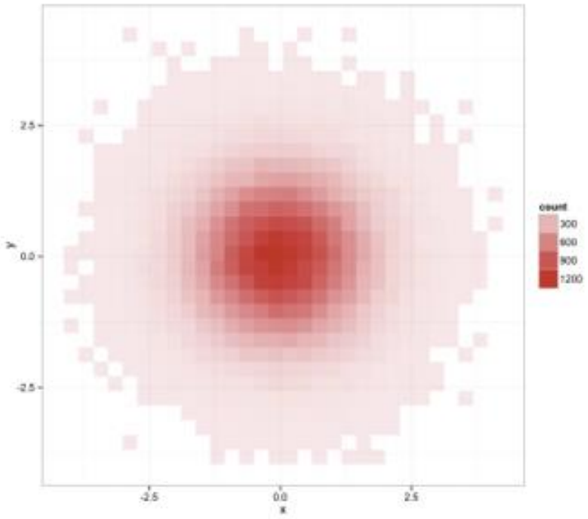
Hexagonal or Rectangular Bins?



100,000 Data Points



Hexagonal Bins

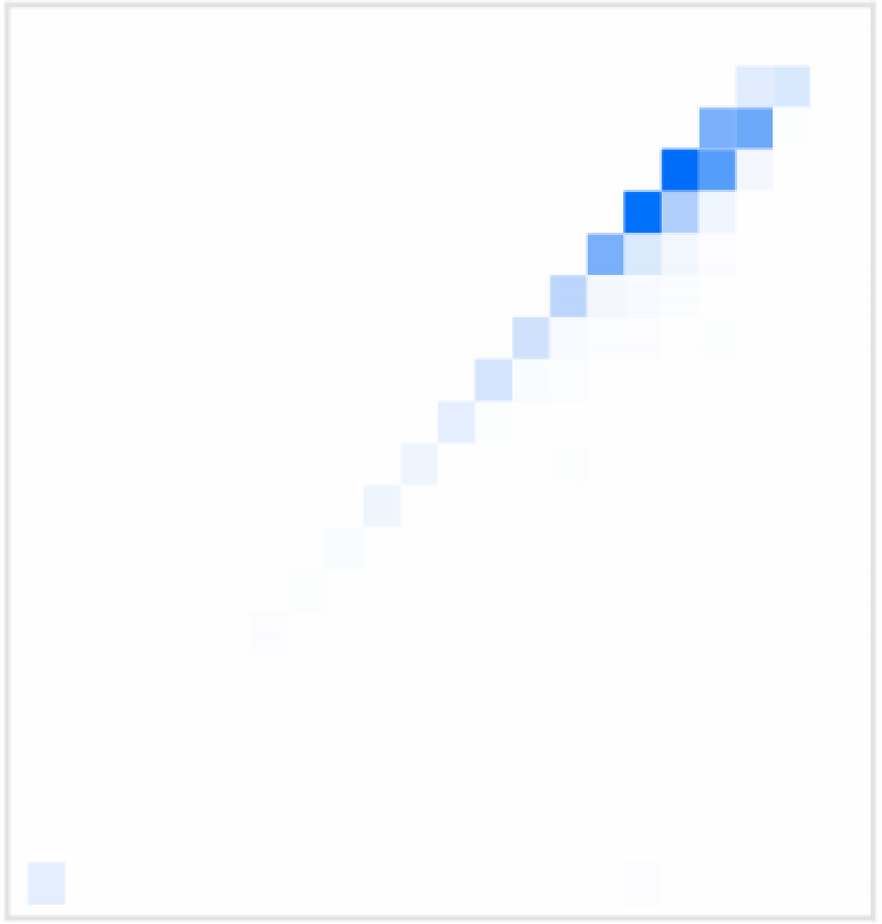


Rectangular Bins

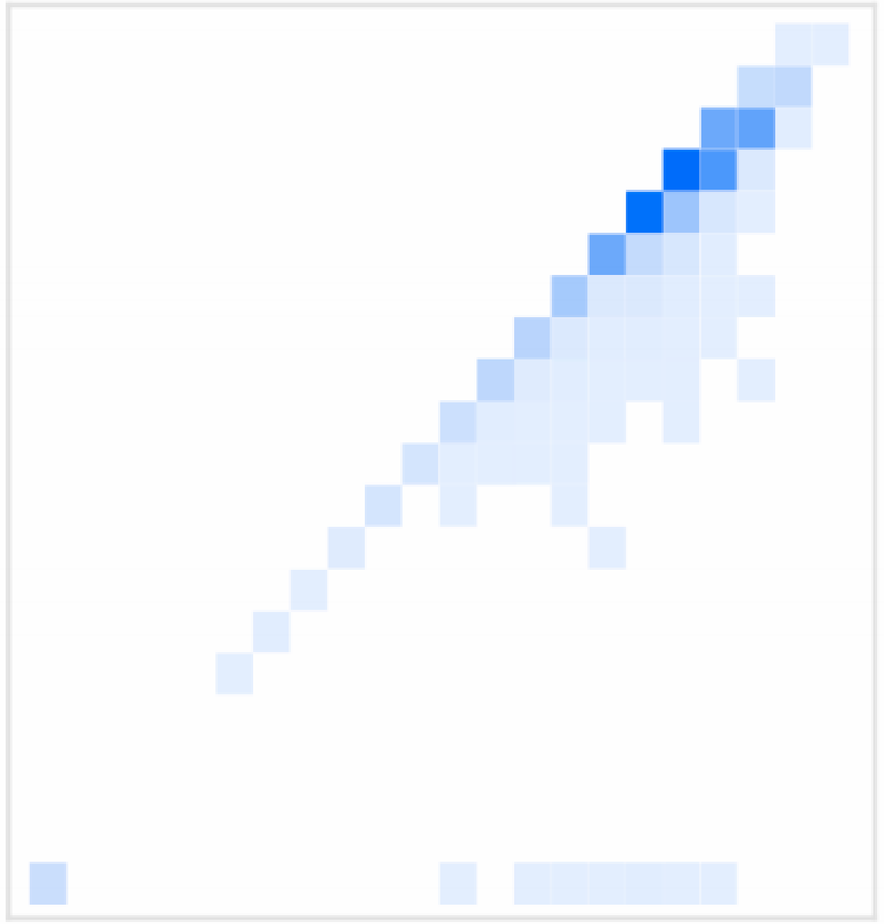
Hex bins better estimate density for 2D plots,

but the *improvement is marginal* [Scott 92].
Rectangles support *reuse* and *visual queries.*

Color Scale: Discontinuity after Zero



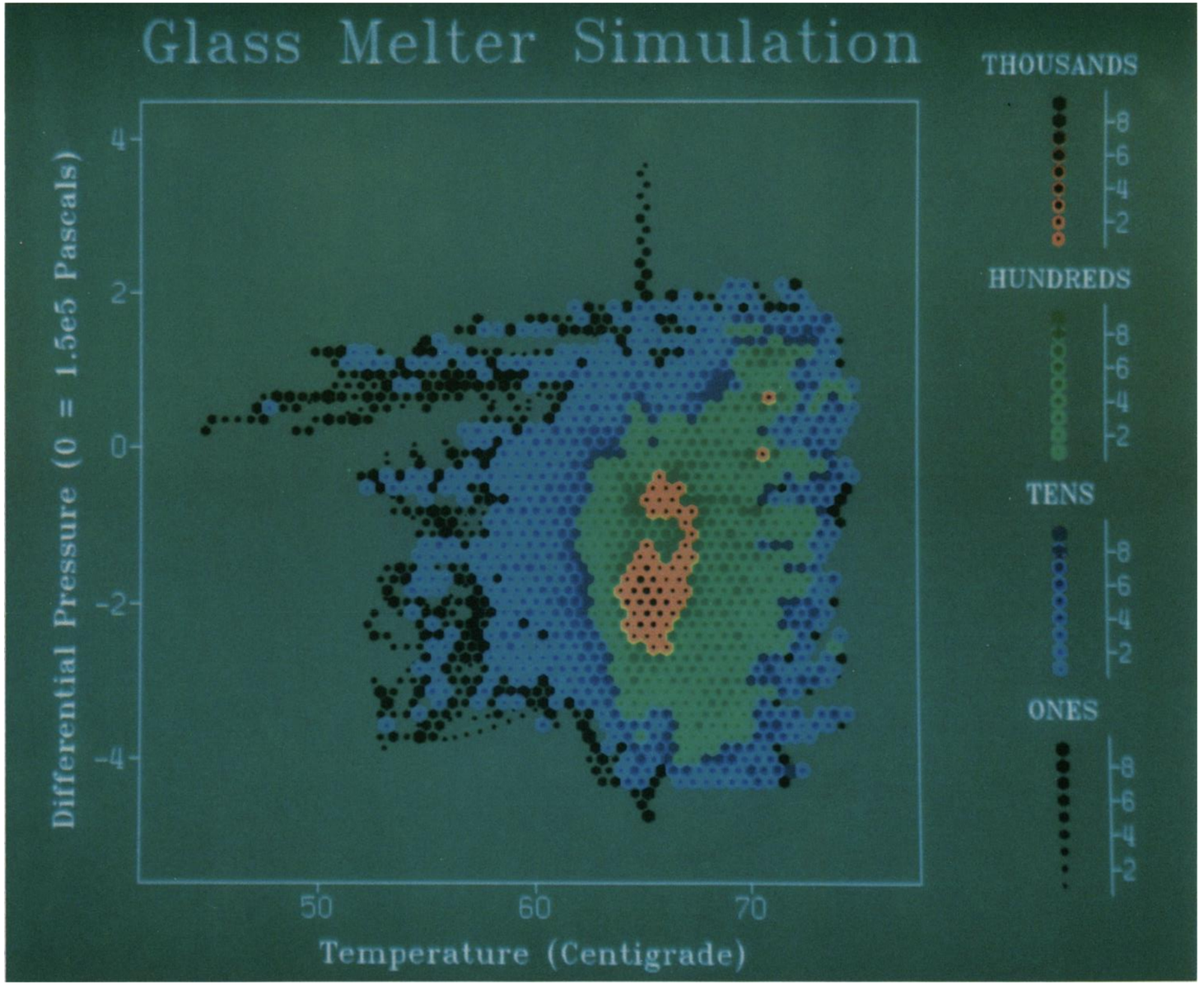
Standard Color Ramp
Counts near zero are white.



Add Discontinuity after Zero
Counts near zero remain visible.

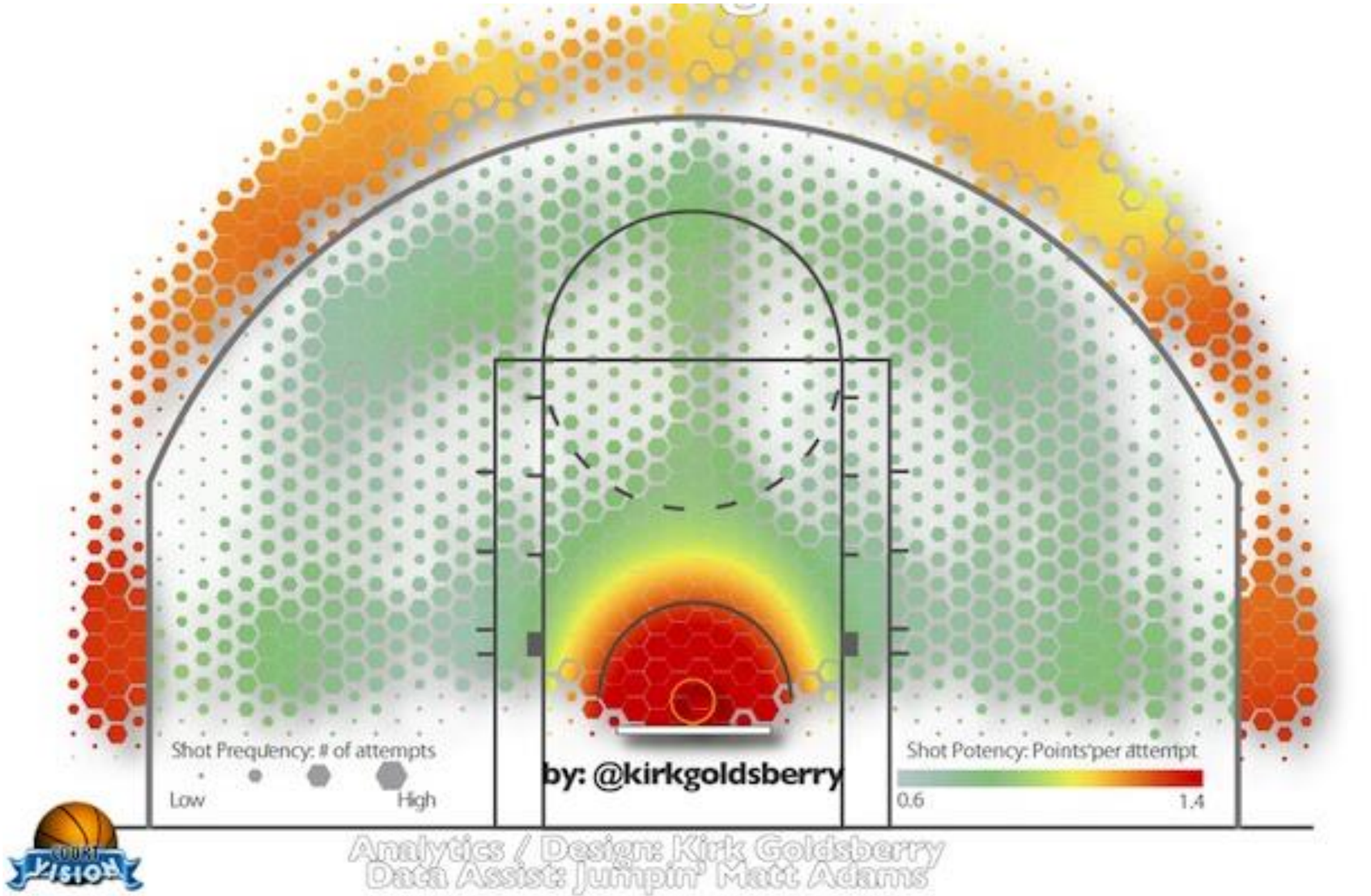
Examples

Example: Binned Scatter Plots



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

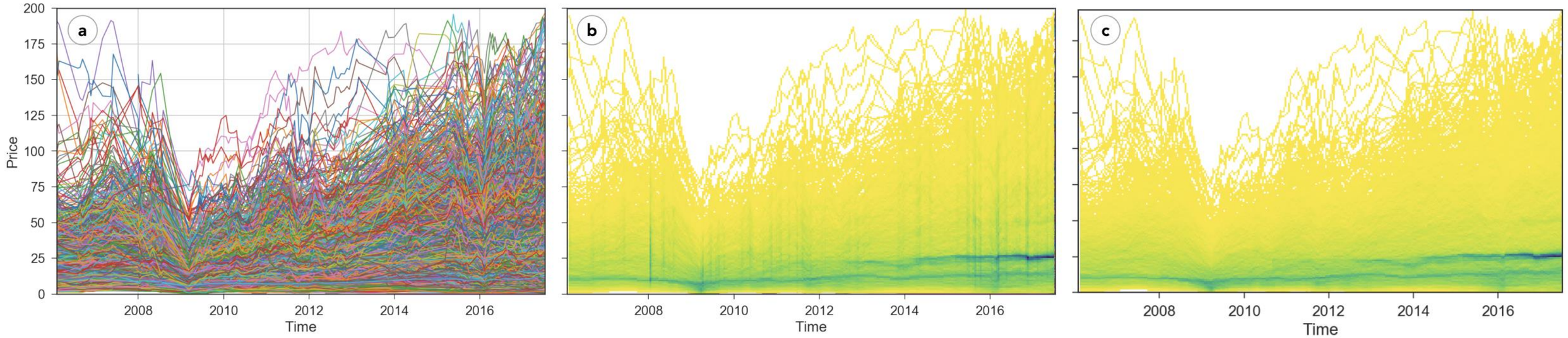
Example: Basketball Shot Chart



NBA Shooting 2011-12
[Goldsberry]

Example: Density Line Chart

[Moritz & Fisher]



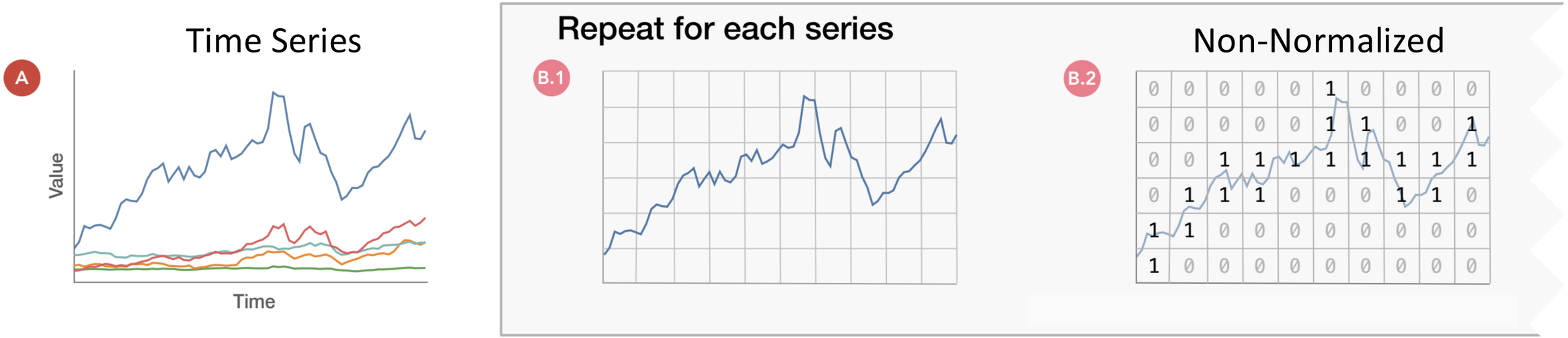
Line Chart

Non-Normalized Heatmap

Normalized “DenseLines”

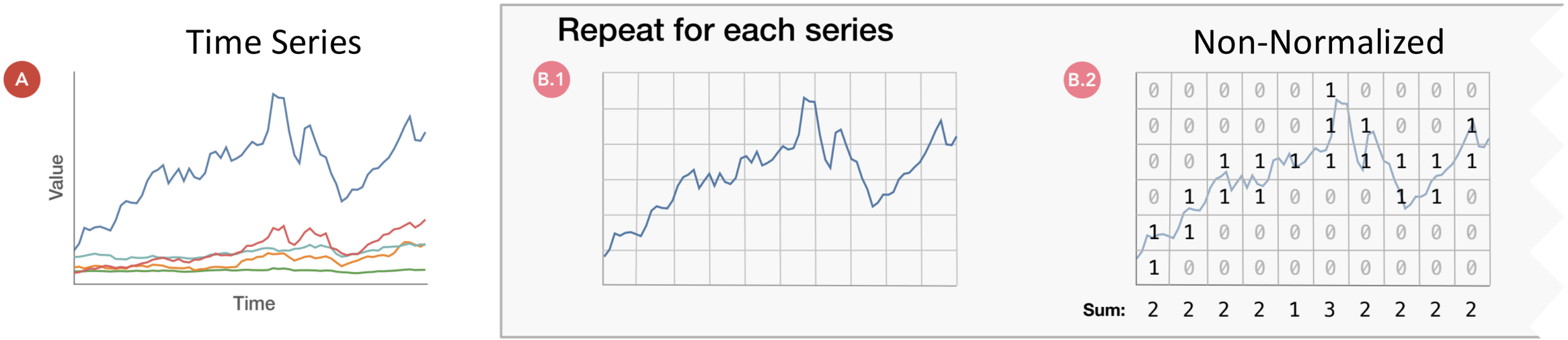
Example: Density Line Chart

[Moritz & Fisher]



Example: Density Line Chart

[Moritz & Fisher]

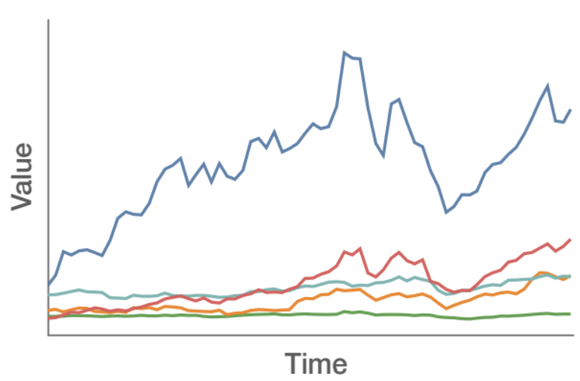


Example: Density Line Chart

[Moritz & Fisher]

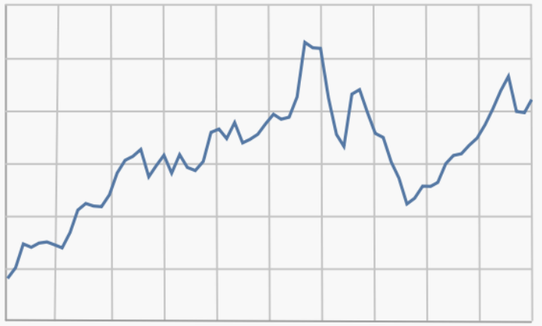
Time Series

A



Repeat for each series

B.1



Non-Normalized

B.2

0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1
0	1	1	1	0	0	0	1	1	0
1	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
Sum:	2	2	2	2	1	3	2	2	2	2

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

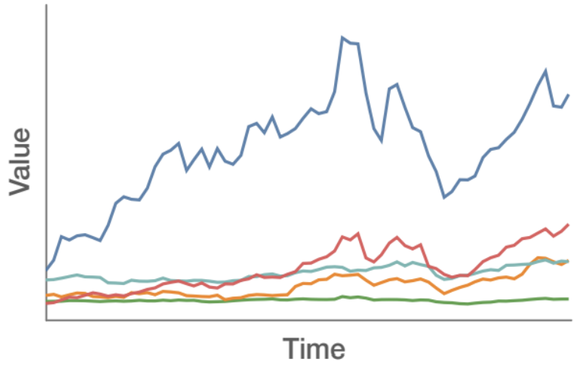
Approx. Arc-Length Normalized

Example: Density Line Chart

[Moritz & Fisher]

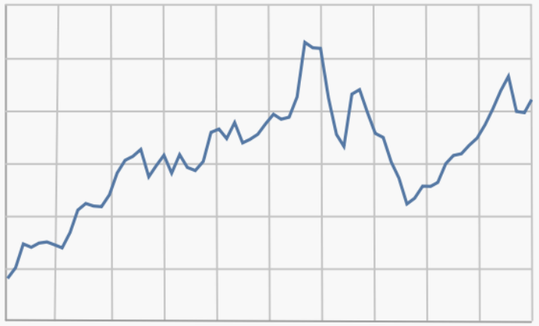
A

Time Series



Repeat for each series

B.1



B.2

Non-Normalized

0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1
0	1	1	1	0	0	0	1	1	0
1	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
Sum:	2	2	2	2	1	3	2	2	2	2

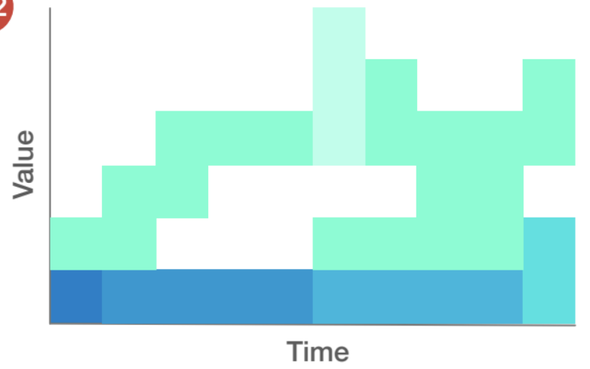
B.3

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

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

C.2



Approx. Arc-Length Normalized

Aggregate

Color

Summary: Perceptual Scalability

Encoding millions+ of data points into one image is a challenge.

Our goal is to reduce the data size to match our target pixel resolution.

Data reduction strategies such as sampling, filtering, modeling, and (binned) aggregation each have their own pros and cons.

Binned aggregation is the most common technique and can be applied to many visualization types.

2. Enabling Real-Time Interaction

Challenge of Interactive Scalability

Each **interaction** triggers a **query** over the original data, which will alter the data we need to render in the visualization.

We may have to retrieve the necessary query results from a **back-end database system** before we can render them.

Interactive Scalability Strategies

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

[Battle & Scheidegger 2020]

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

[Battle & Scheidegger 2020]

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

[Battle & Scheidegger 2020]

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

[Battle & Scheidegger 2020]

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.

[Battle & Scheidegger 2020]

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.

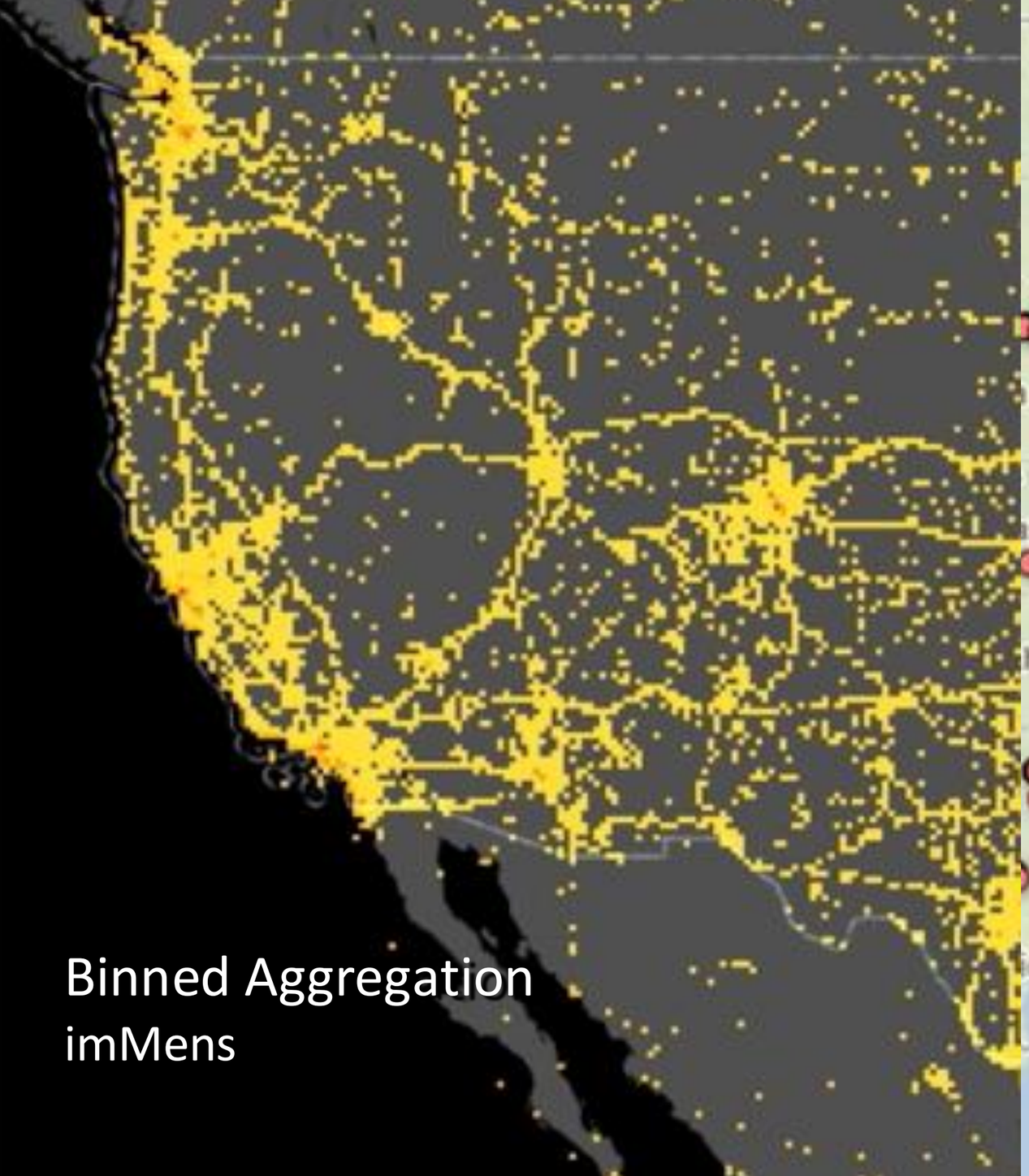
[Battle & Scheidegger 2020]

imMens

[Liu, Jiang & Heer ’13]

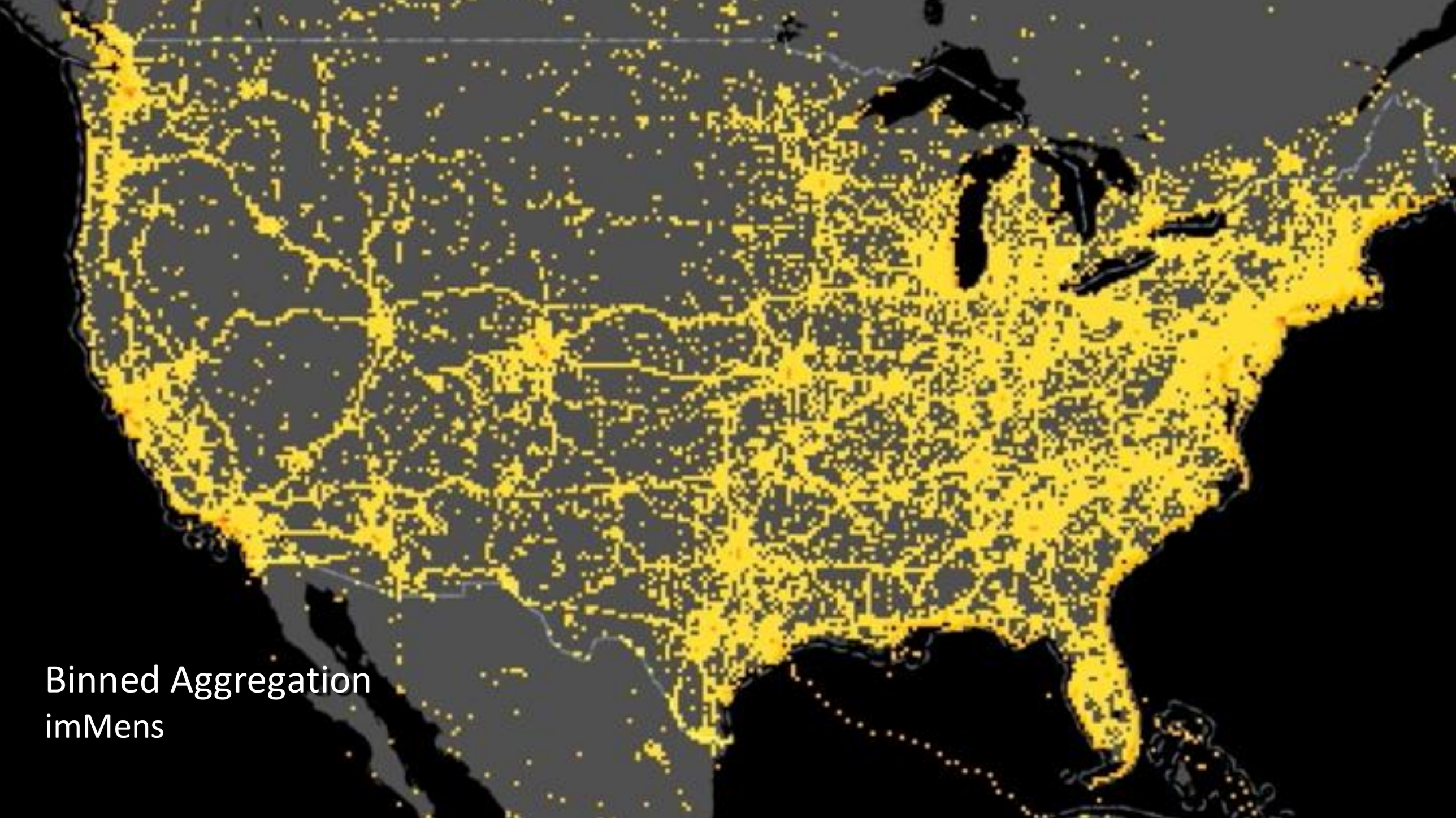
Strategies: Client-Side Data Cubes

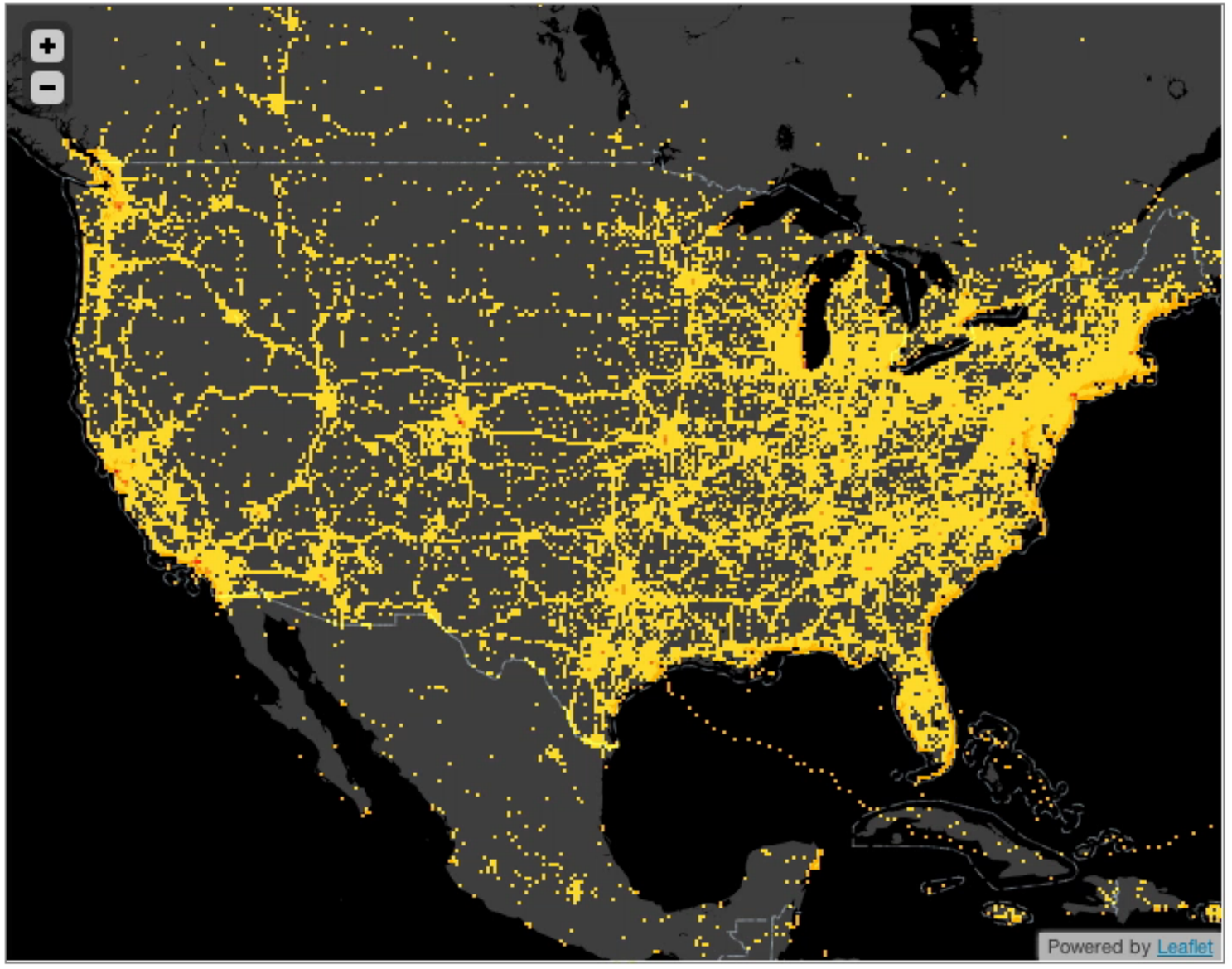

Montana
Minnesota
South Dakota
Wisconsin
Michigan
Toronto
Montreal
Maine
Oregon
Idaho
New York
Illinois
Pennsylvania
Nevada
Utah
Missouri
West Virginia
Kentucky
Virginia
Connecticut
New Jersey
Delaware
Maryland
District of Columbia
Oklahoma
Tennessee
Mississippi
Alabama
Georgia
New Mexico
Phoenix
San Diego
Dallas
Texas
Houston
San Antonio
Gulf of California
Monterrey
Mexico
Havana
Sampling
Google Fusion Tables

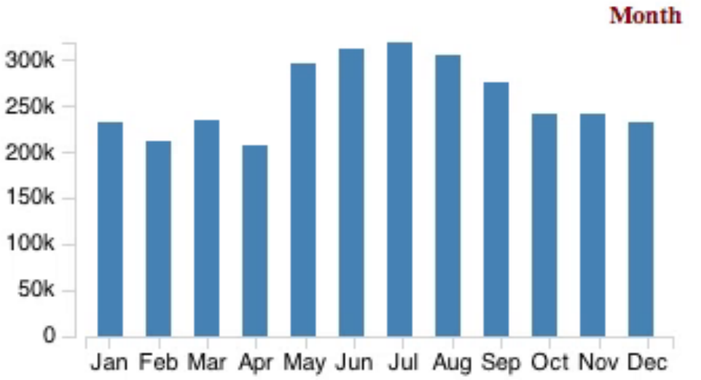


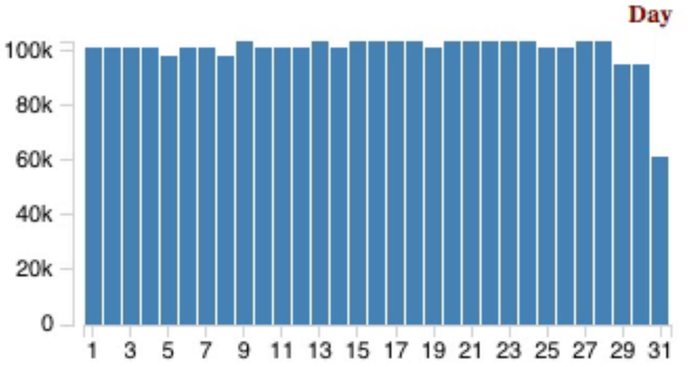
Binned Aggregation
imMens

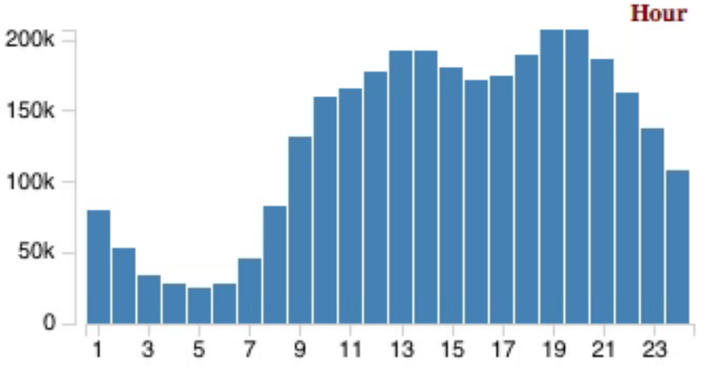
Sampling
Google Fusion Tables


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

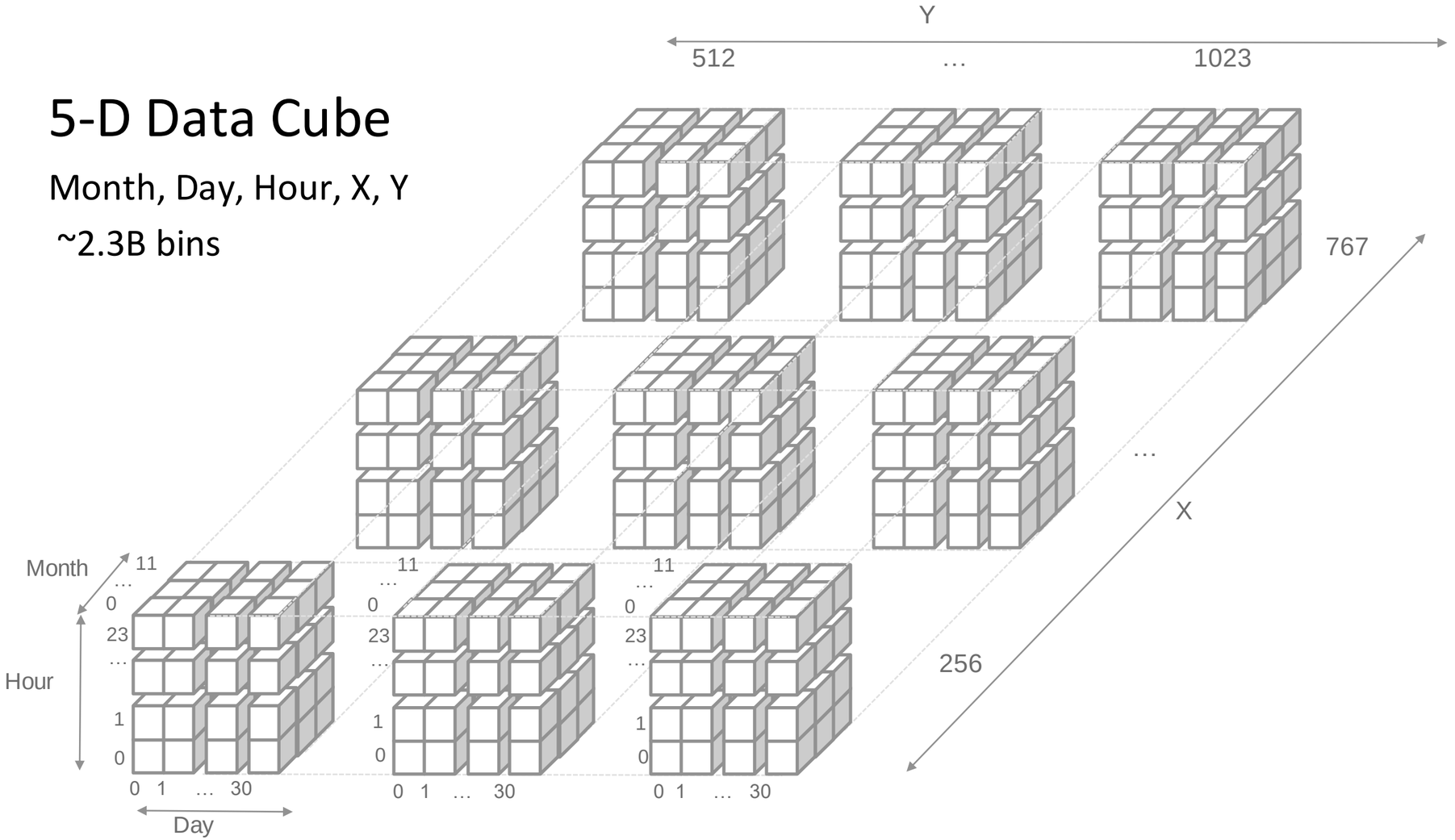

Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31


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

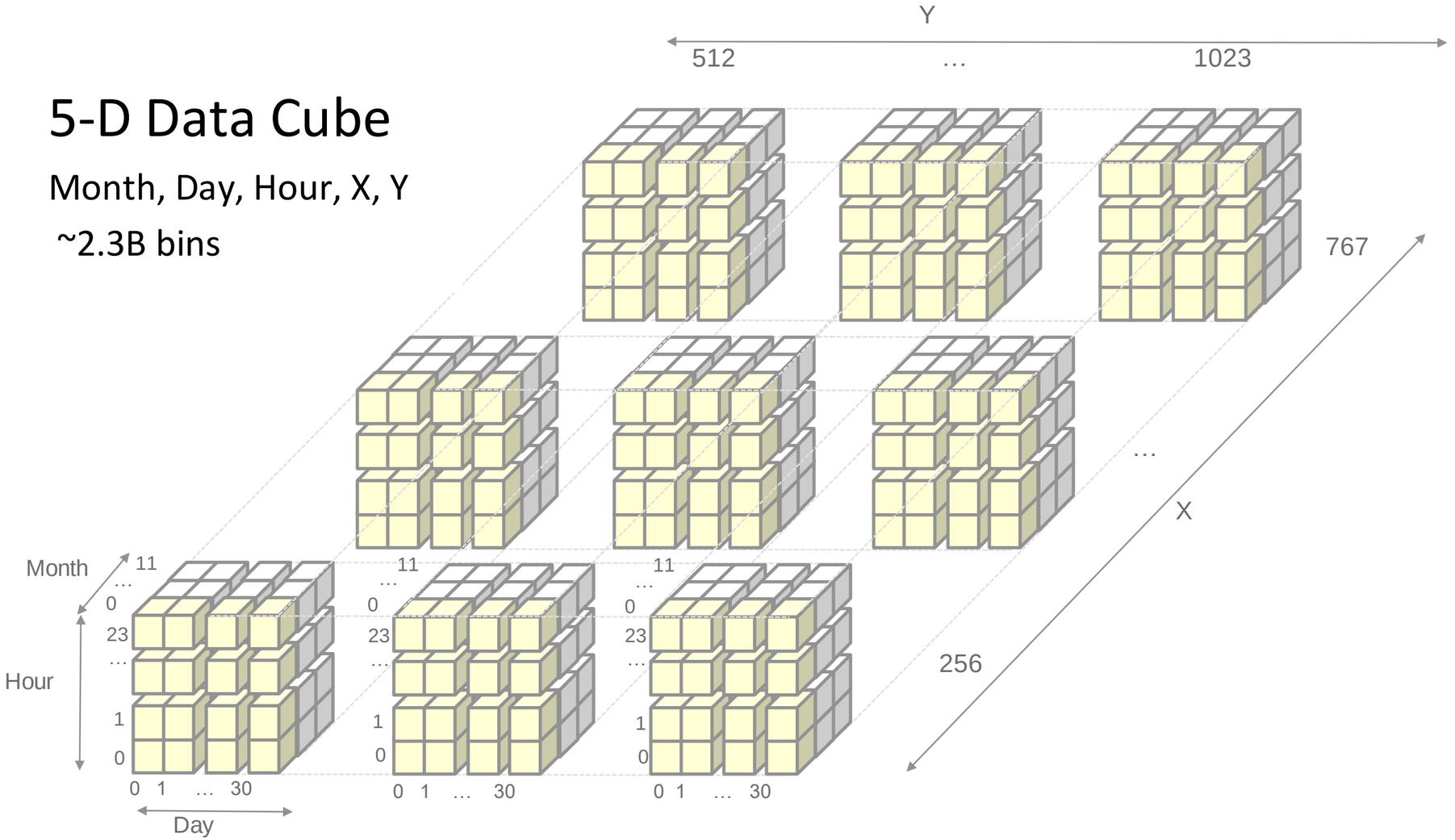
~2.3B bins

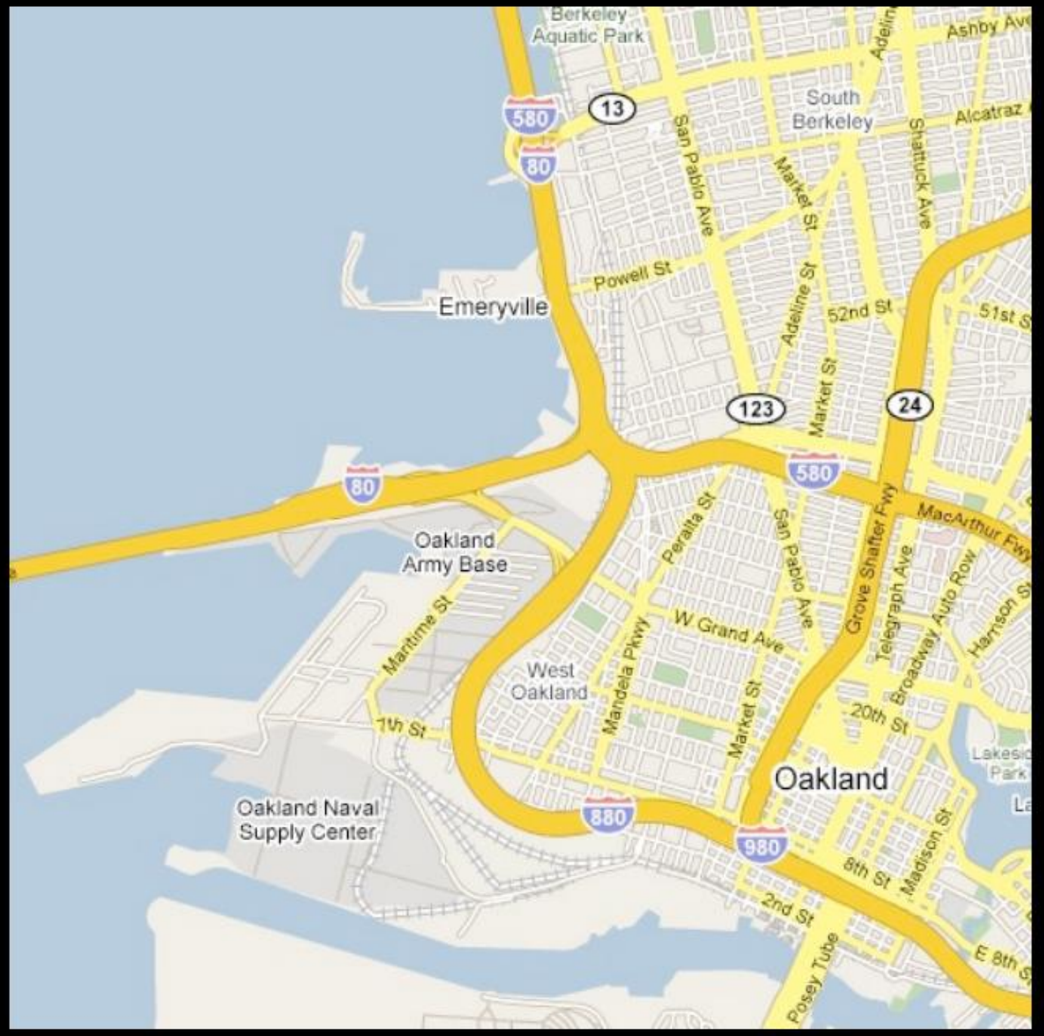


5-D Data Cube

Month, Day, Hour, X, Y

~2.3B bins




Berkeley Aquatic Park
Ashby Ave
Adelin
South Berkeley
Alcatraz
580
13
80
San Pablo Ave
Market St
Shattuck Ave
Powell St
Emeryville
Adeline St
52nd St
51st St
123
24
Market St
580
MacArthur Fwy
80
Oakland Army Base
Peralta St
San Pablo Ave
Grove Shafter Fwy
Telegraph Ave
Broadway Auto Row
Harrison St
Maritime St
W Grand Ave
West Oakland
Mandela Pkwy
Market St
20th St
11th St
Oakland
Lakeside Park
880
980
Oakland Naval Supply Center
8th St
Madison St
2nd St
E 8th St
Posey Tube


Emeryv
80

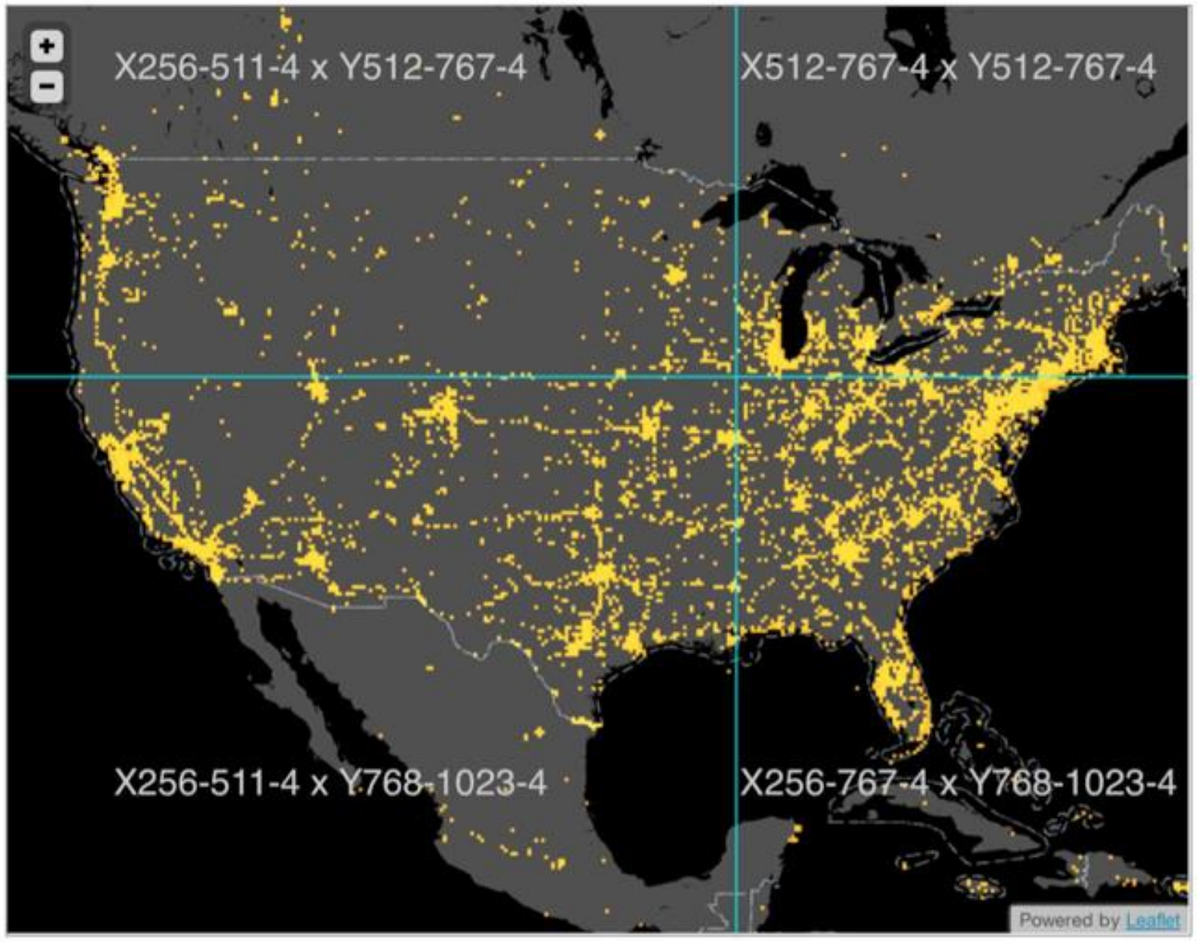

Berkeley
Aquatic Park
Ashby Ave
13
80
80
South
Berkeley
Alcatraz
San Pablo Ave
Market St
Shattuck Ave
Powell St
Adeline St
52nd St
51st S
123
Market St
24
580


Oakland
Army Base
Maritime St
7th St
Oakland Naval
Supply Center

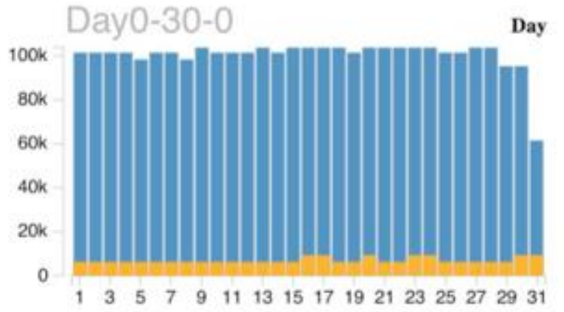

Peralta
San Pablo Ave
Grove Shafter
Telegraph Ave
Broadway Auto Row
Harrison St
W Grand Ave
West
Oakland
Mandela Pkwy
Market St
20th St
Oakland
880
980
8th St
Madison St
2nd St
E 8th St
Posey Tube

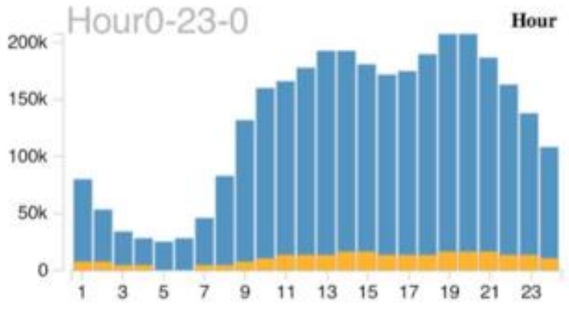
Multivariate Data Tiles

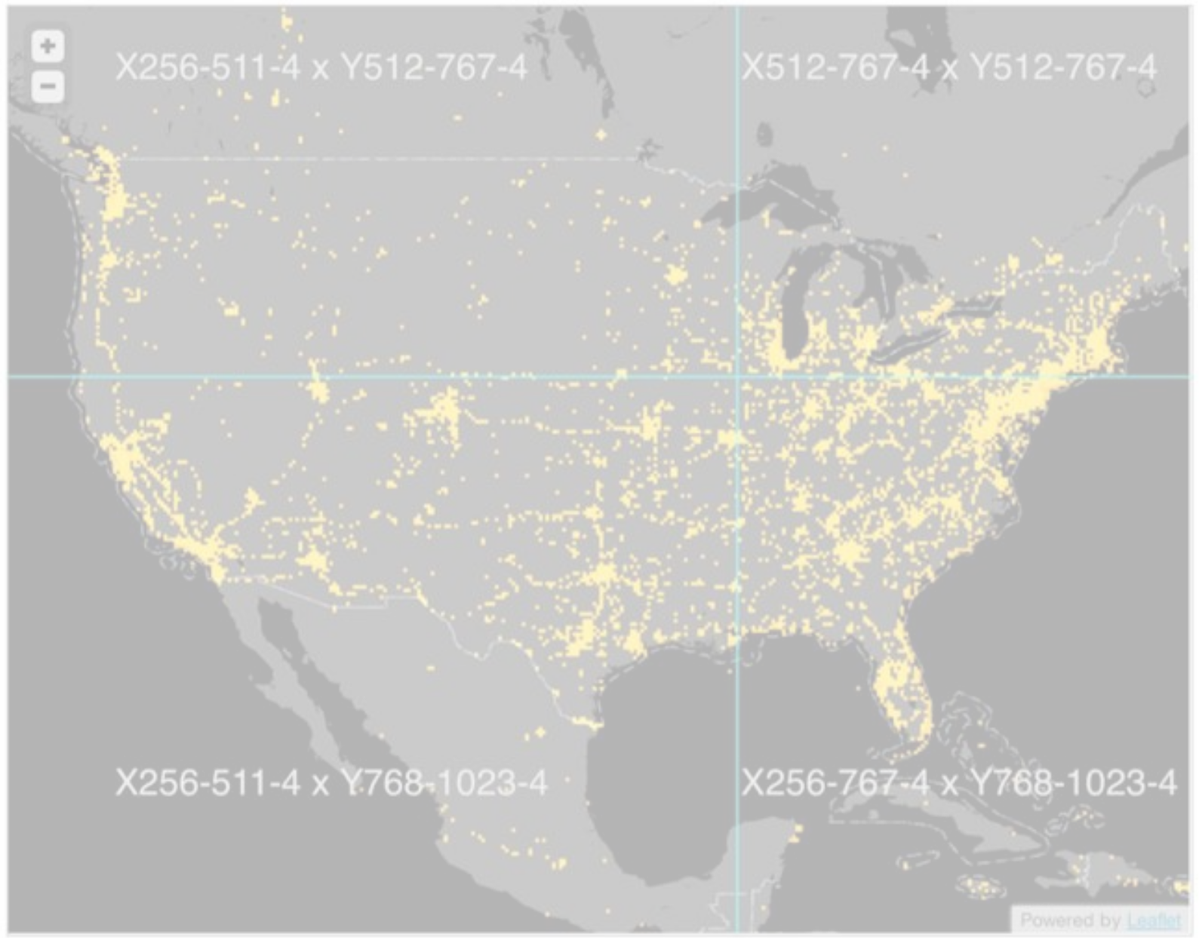
1. Send data, not pixels
2. Embed multi-dim data


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

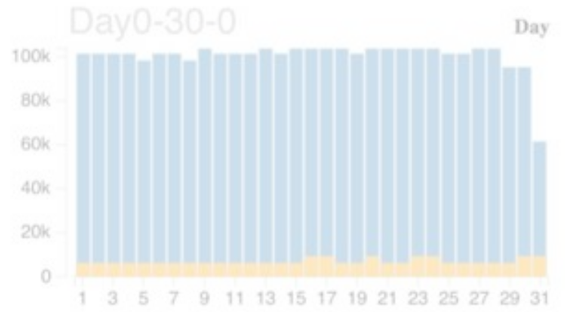

Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

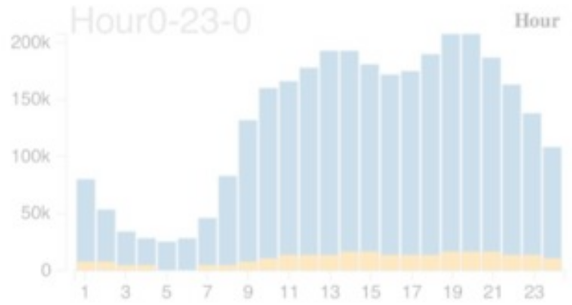

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

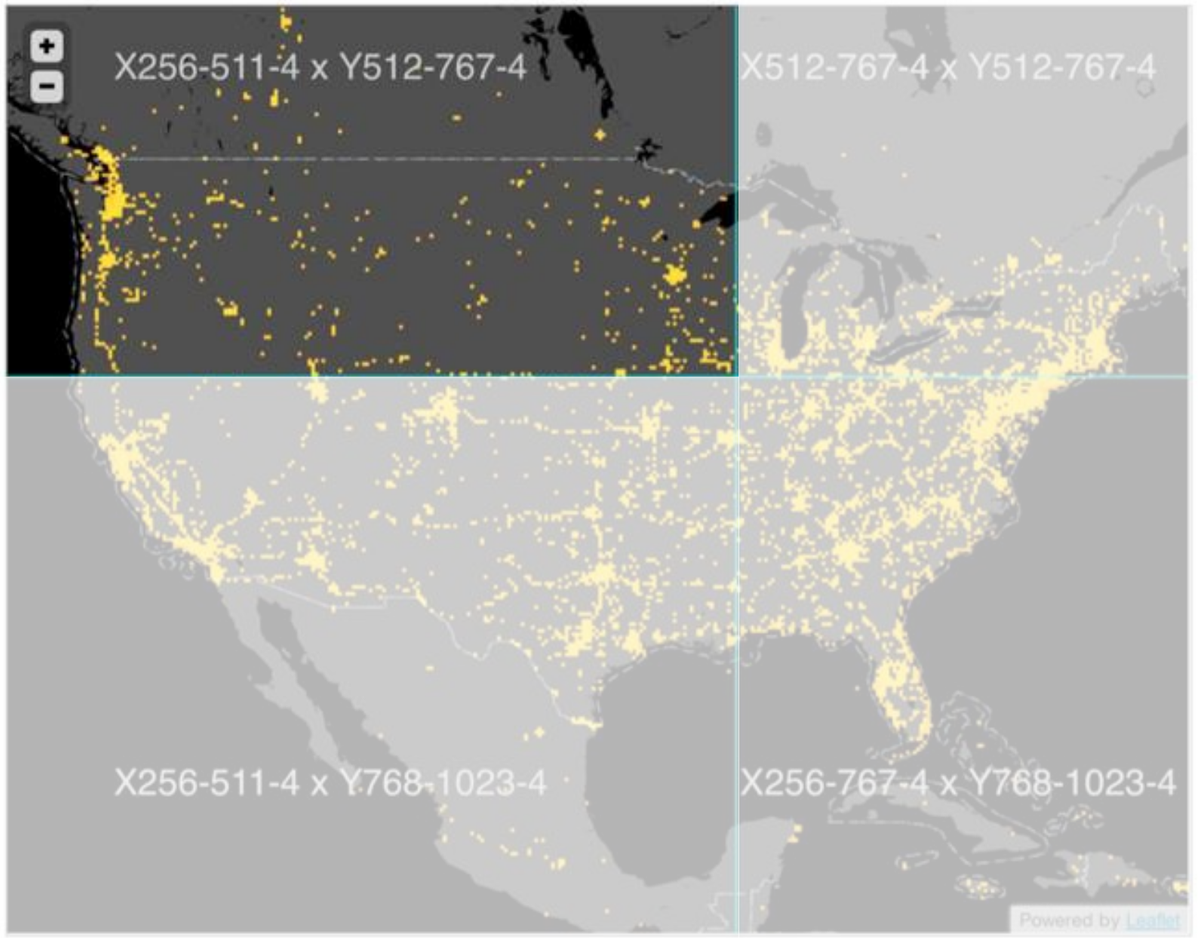

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


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

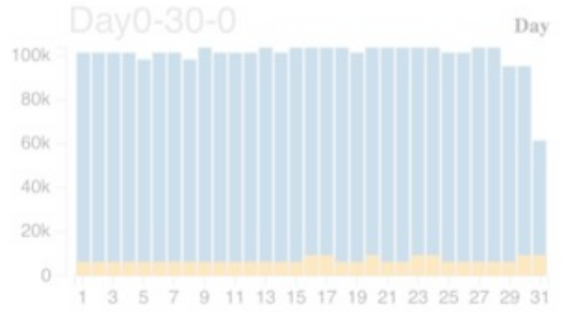

Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

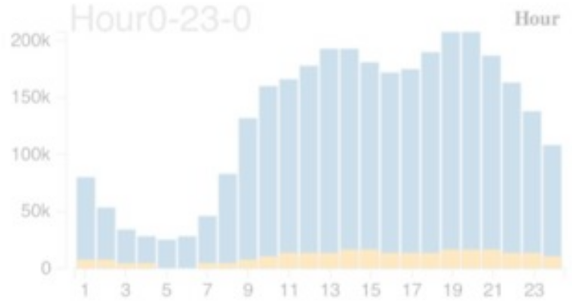

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

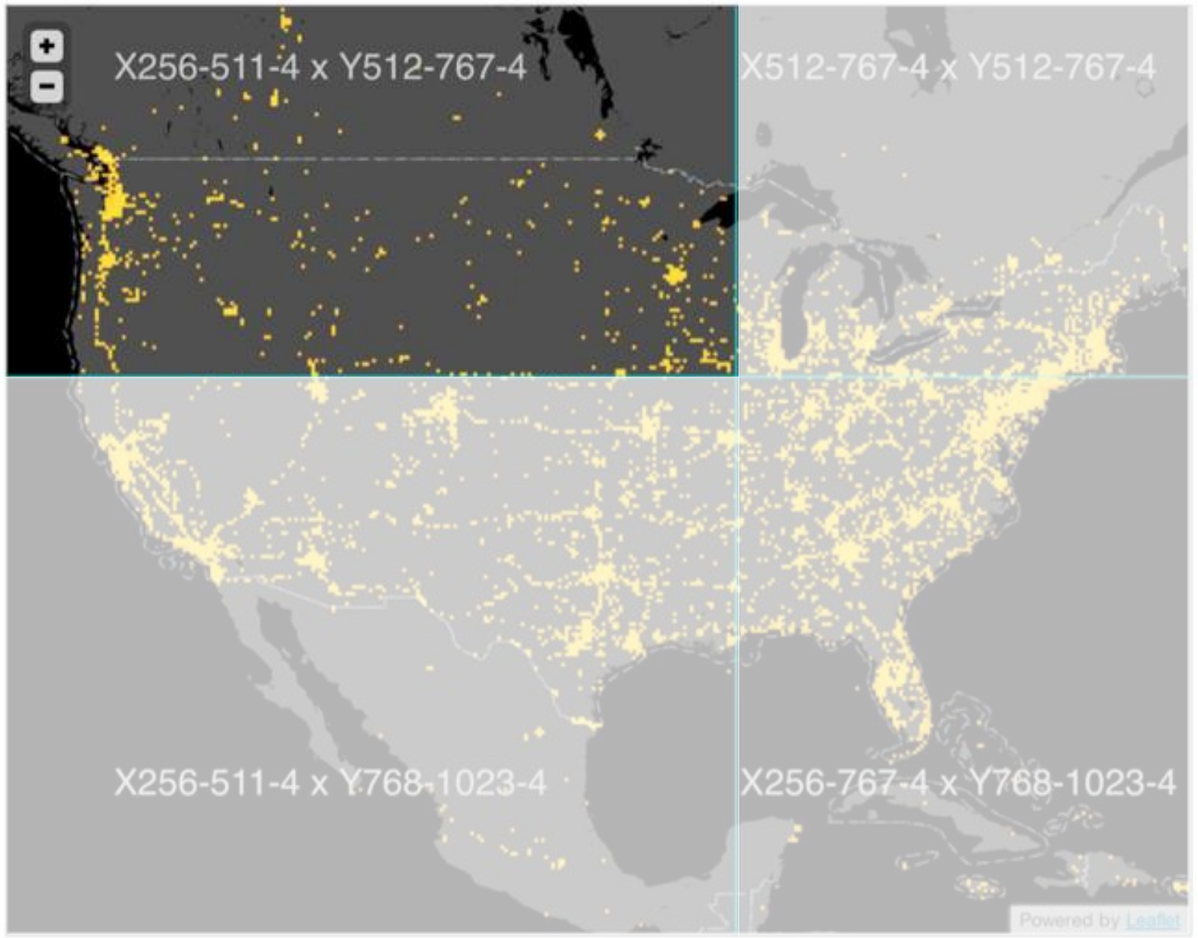

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


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

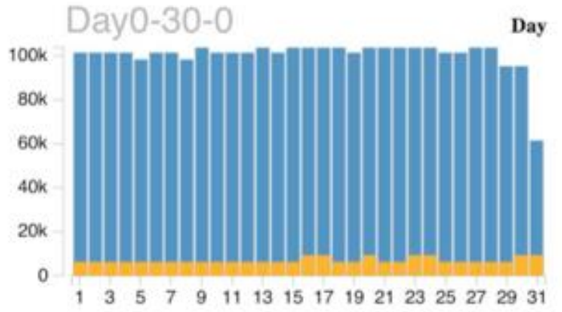

Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

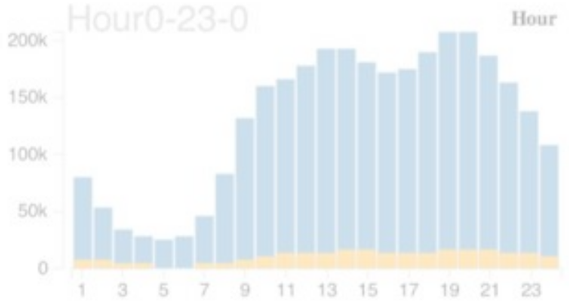

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

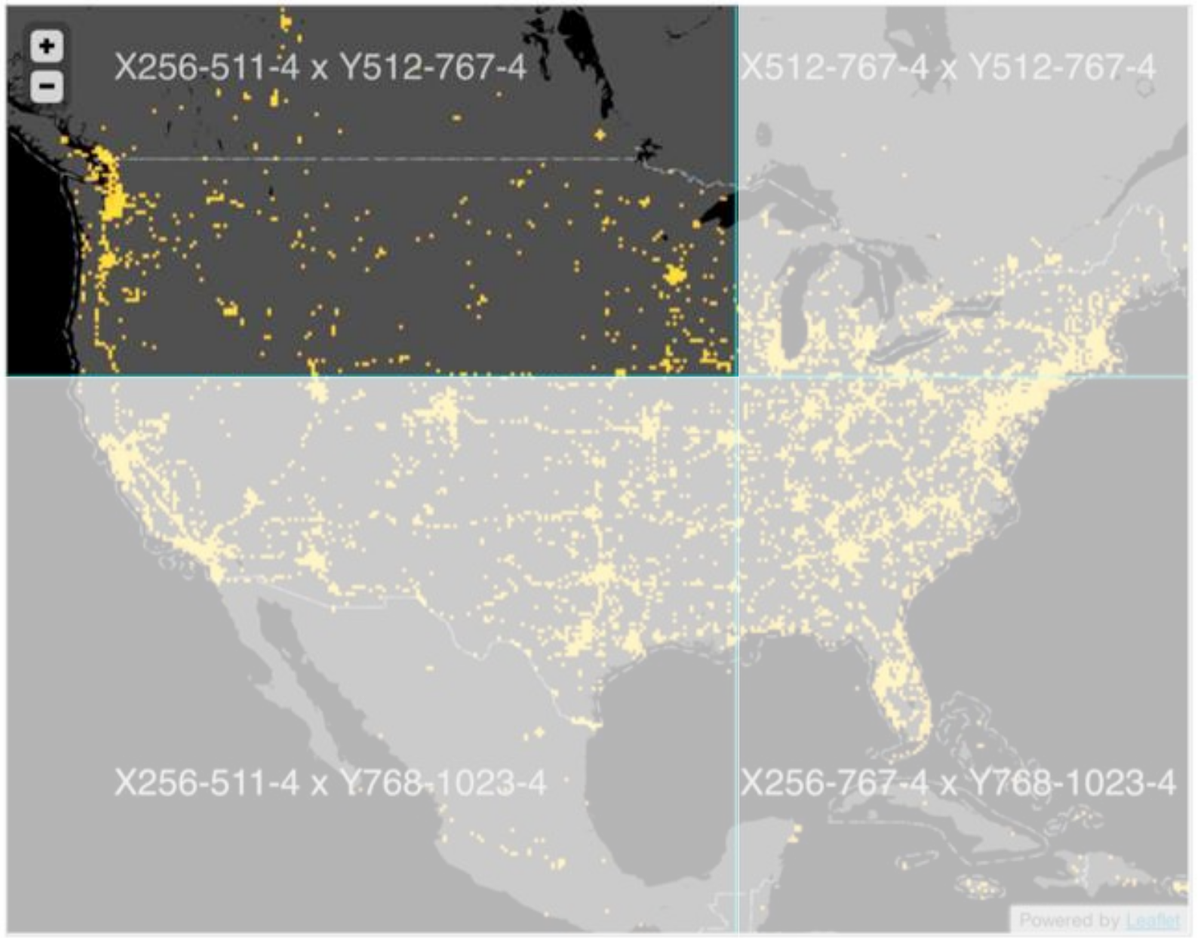

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


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

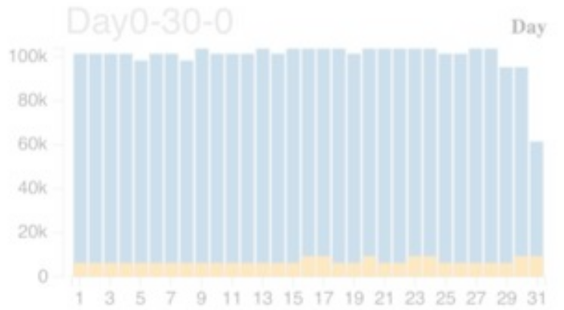

Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

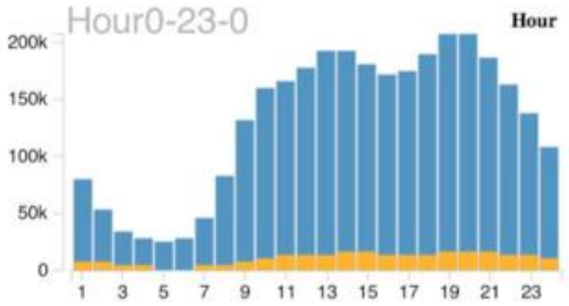

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

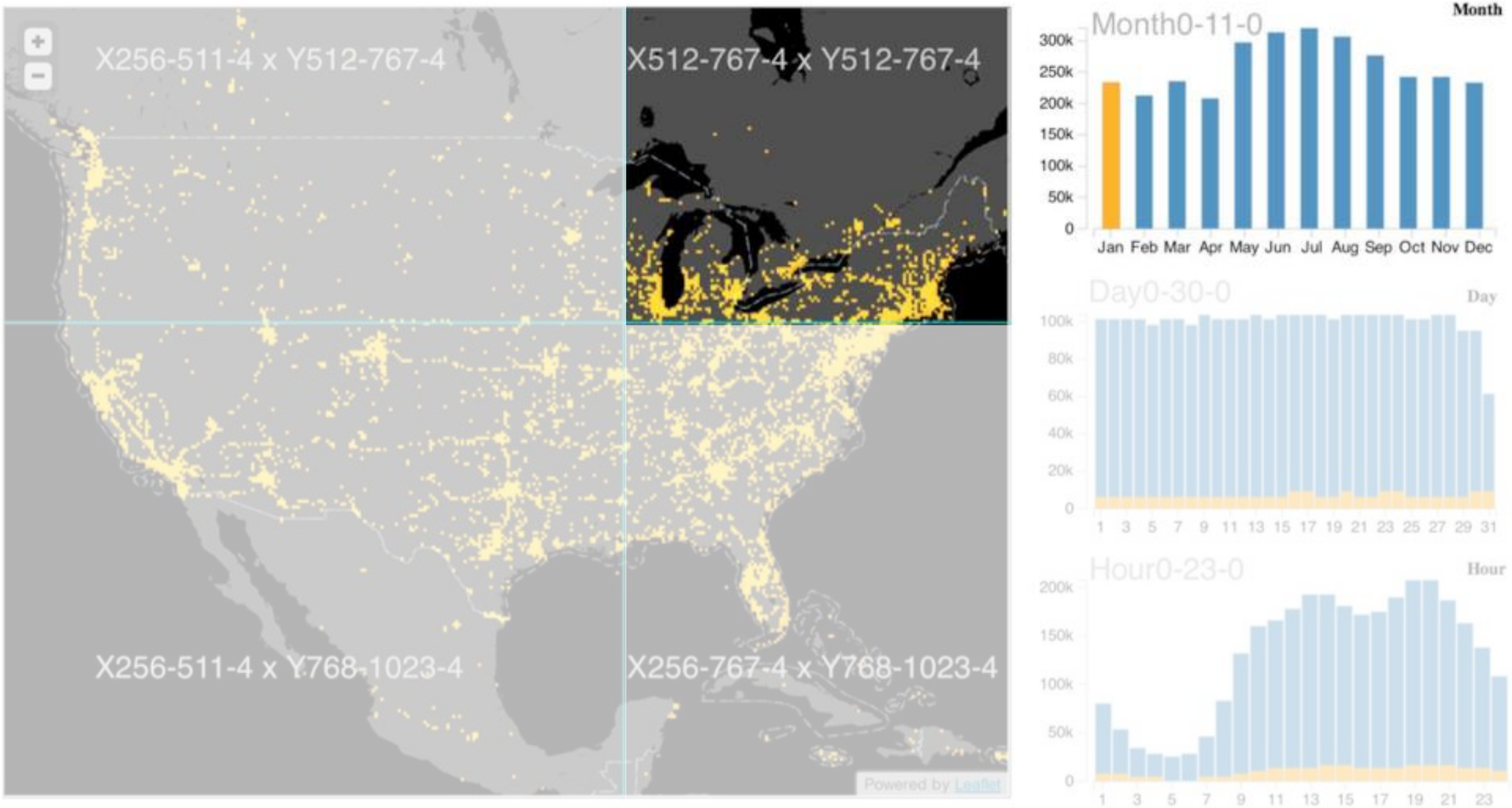

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


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
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

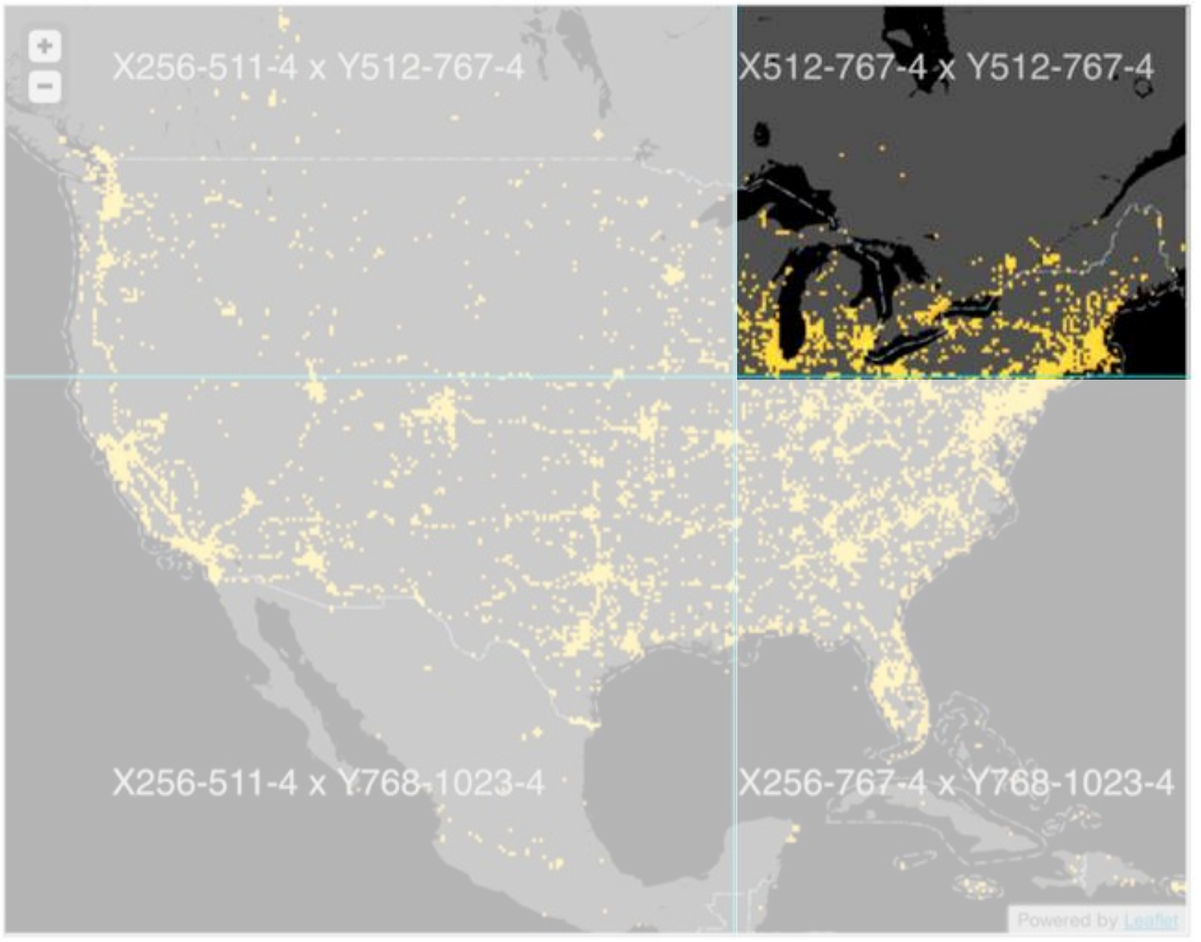

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


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

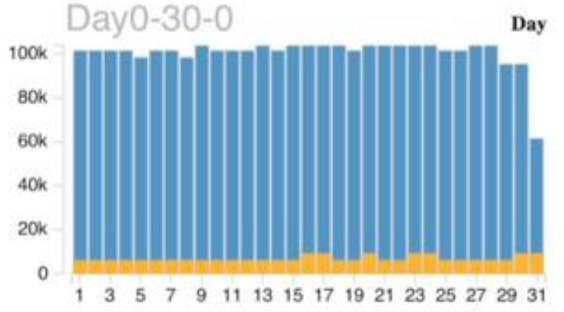
Month
Month0-11-0
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

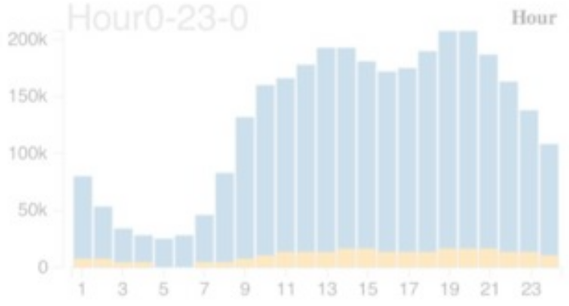
Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

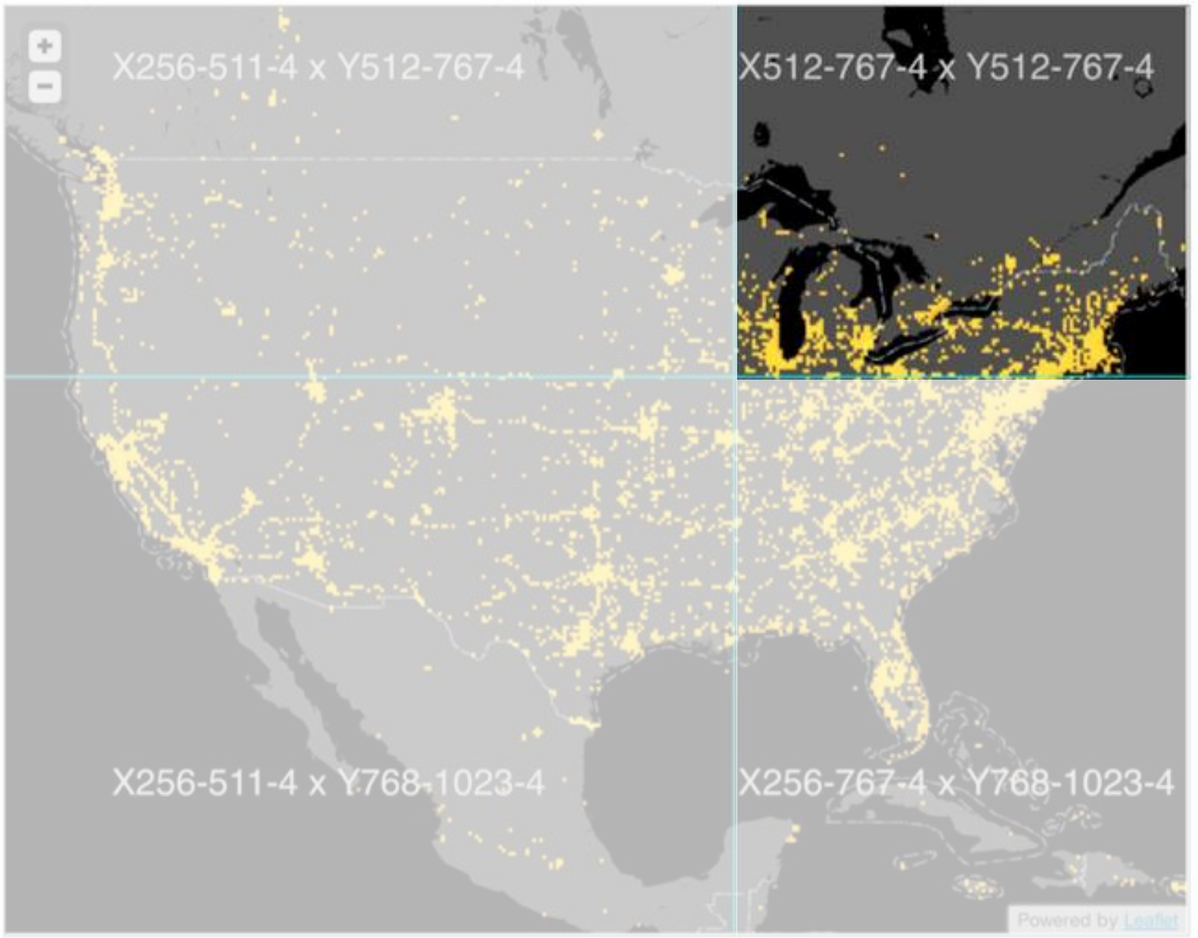
Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


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

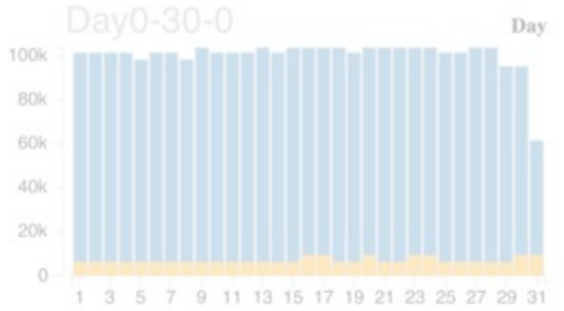

Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

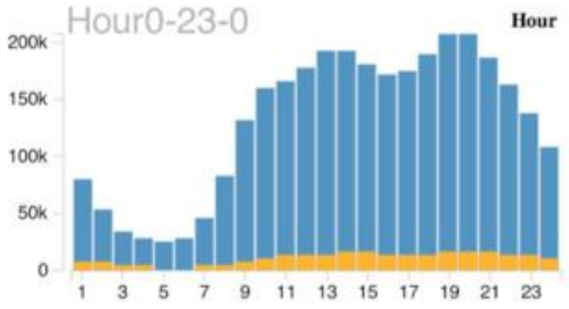

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

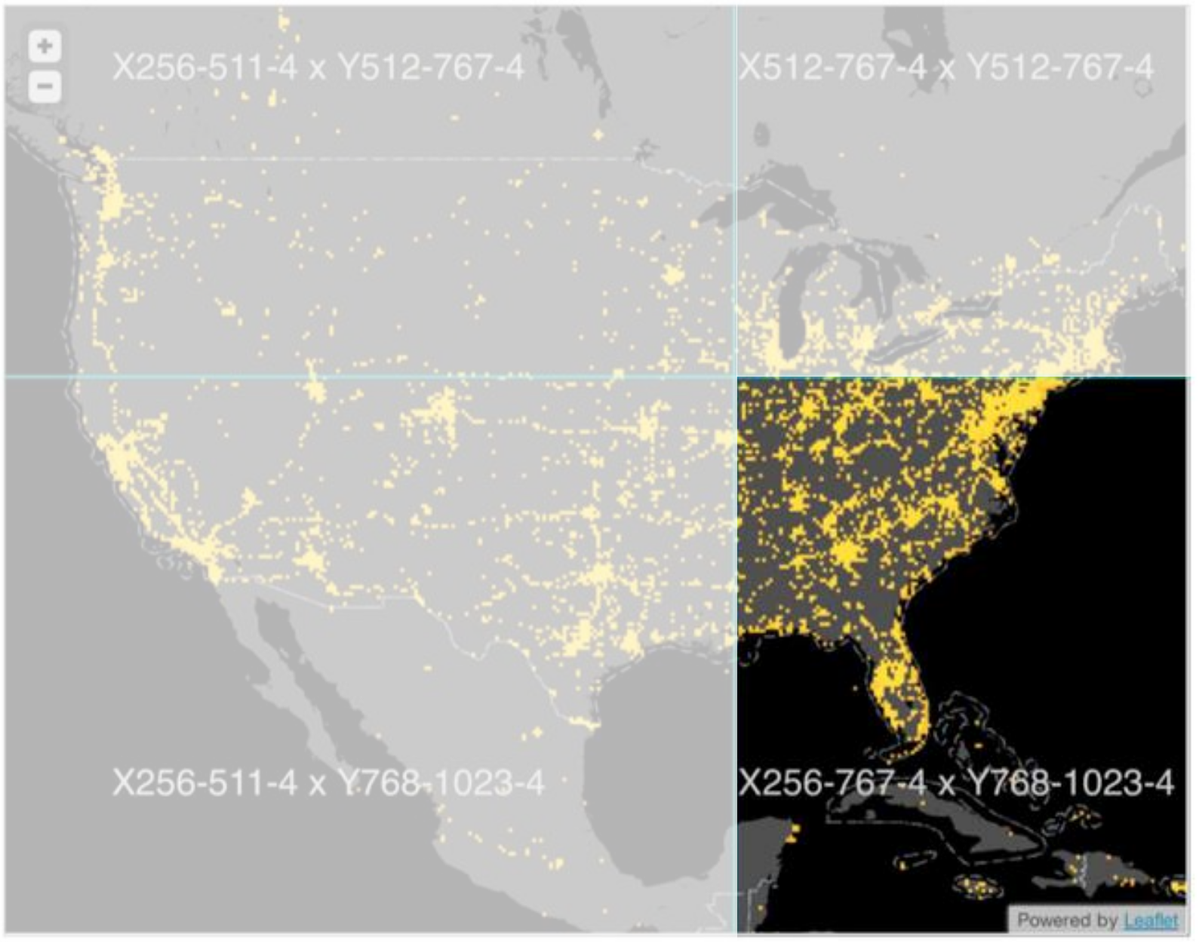

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


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

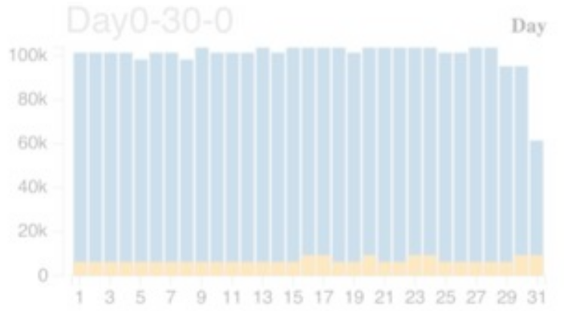

Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

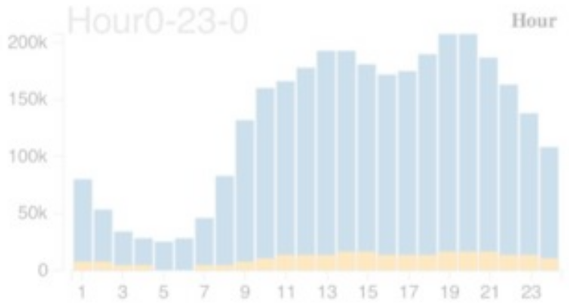

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

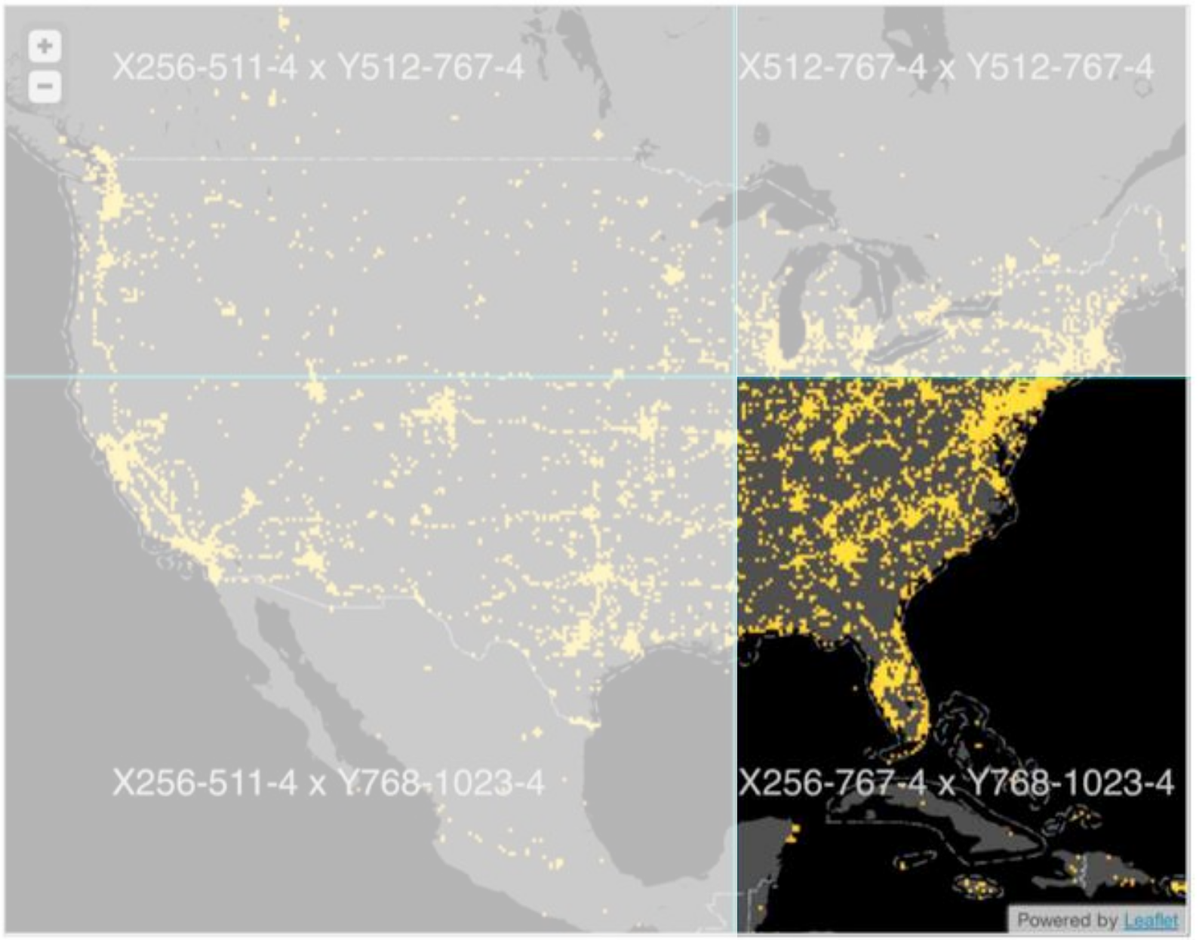

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


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

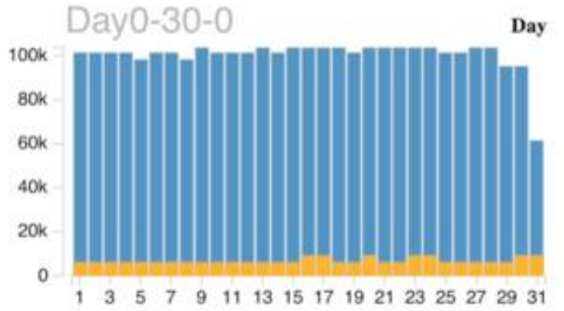

Month
Month0-11-0
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

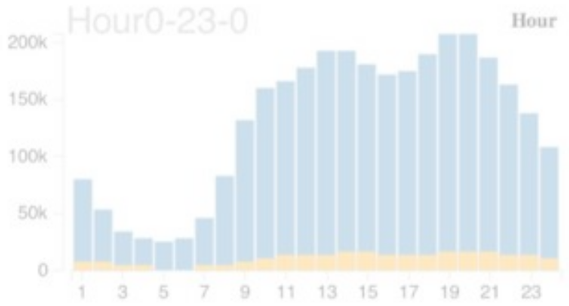

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

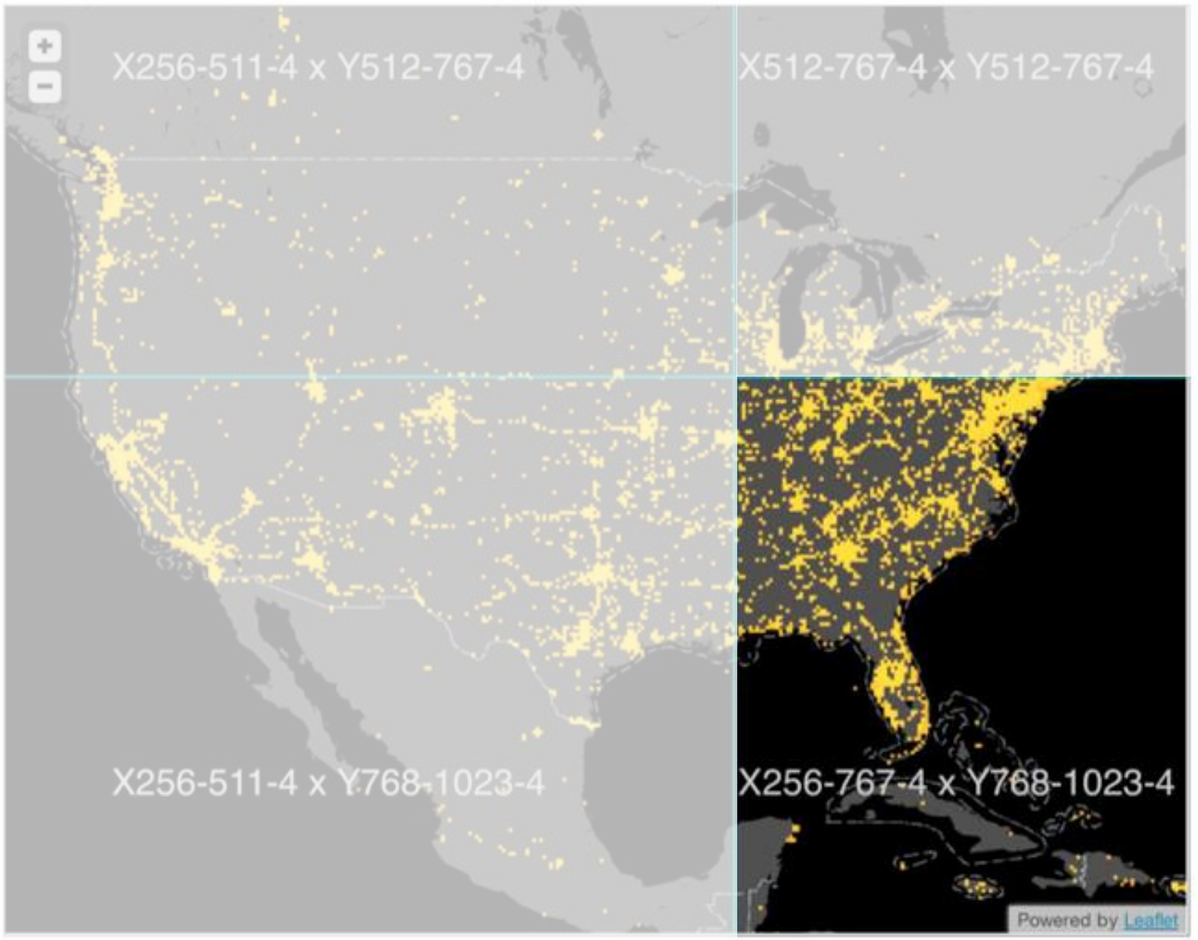

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


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

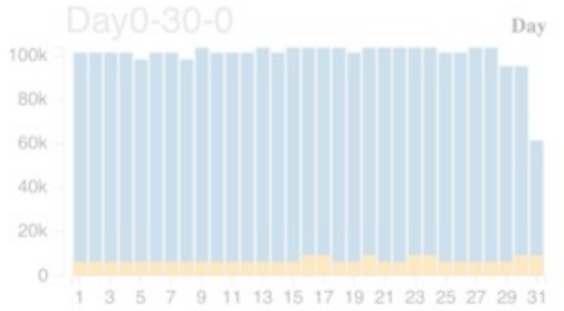

Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

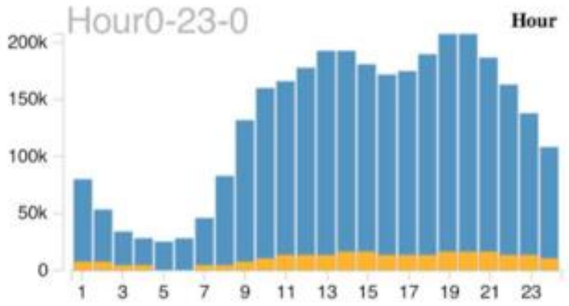

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

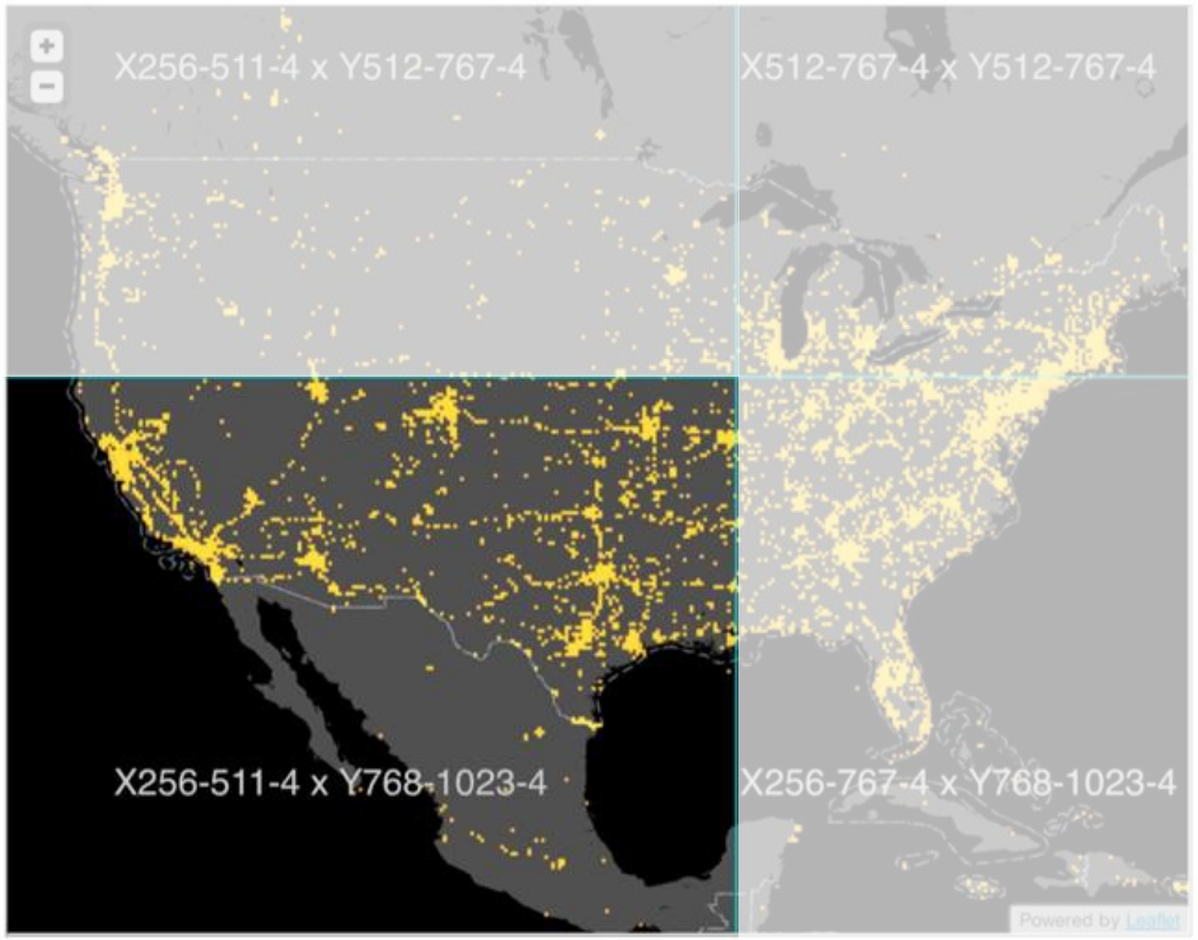

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


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

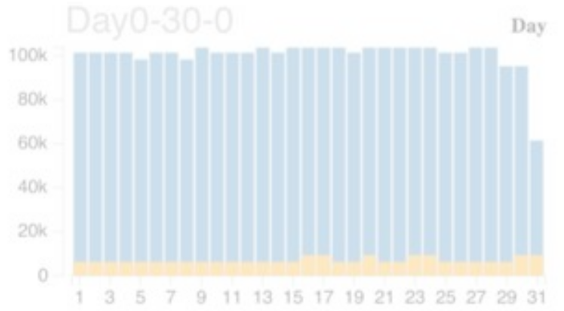

Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

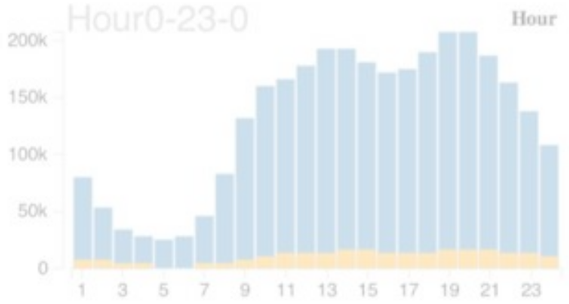

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

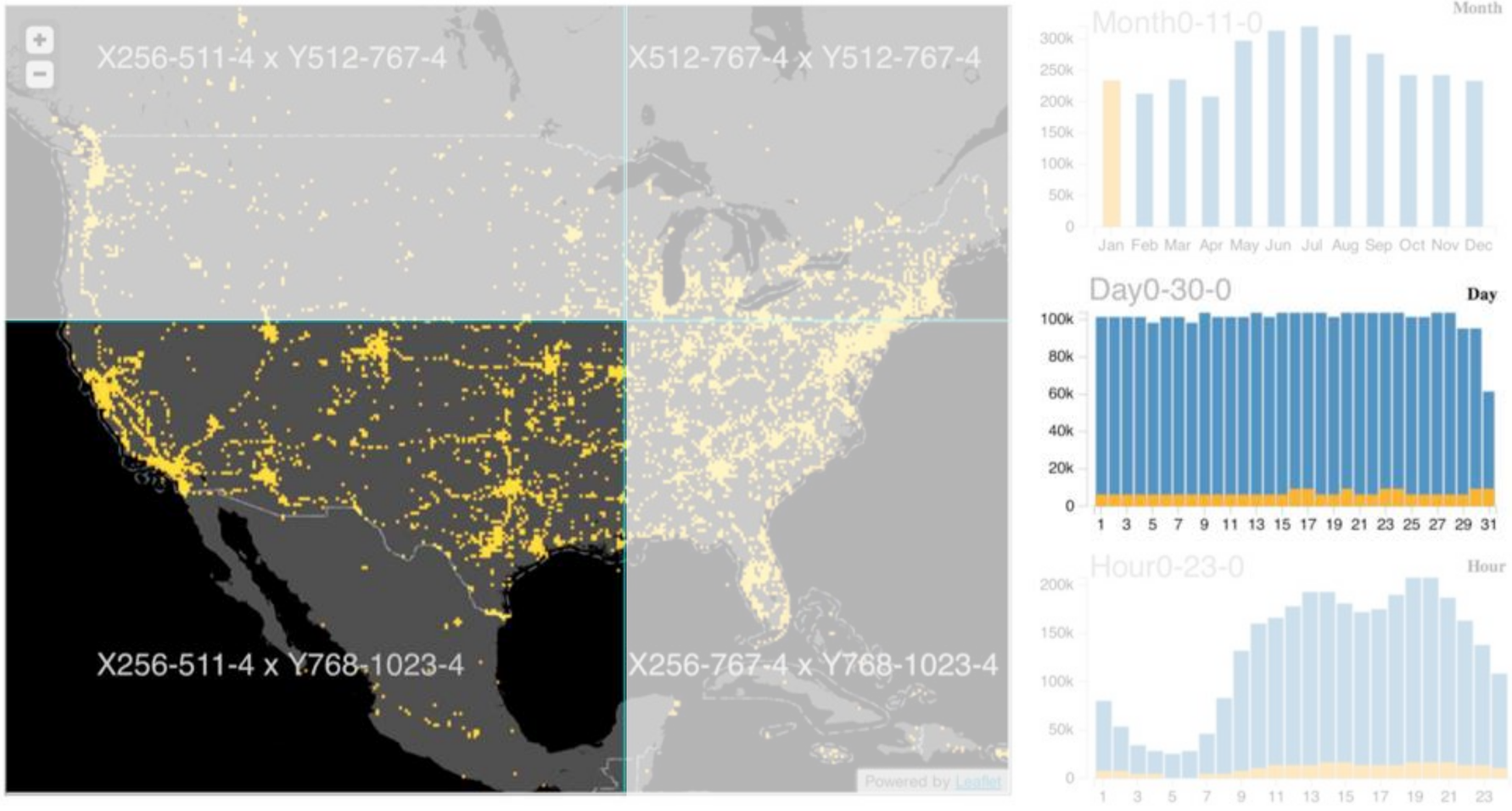

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23

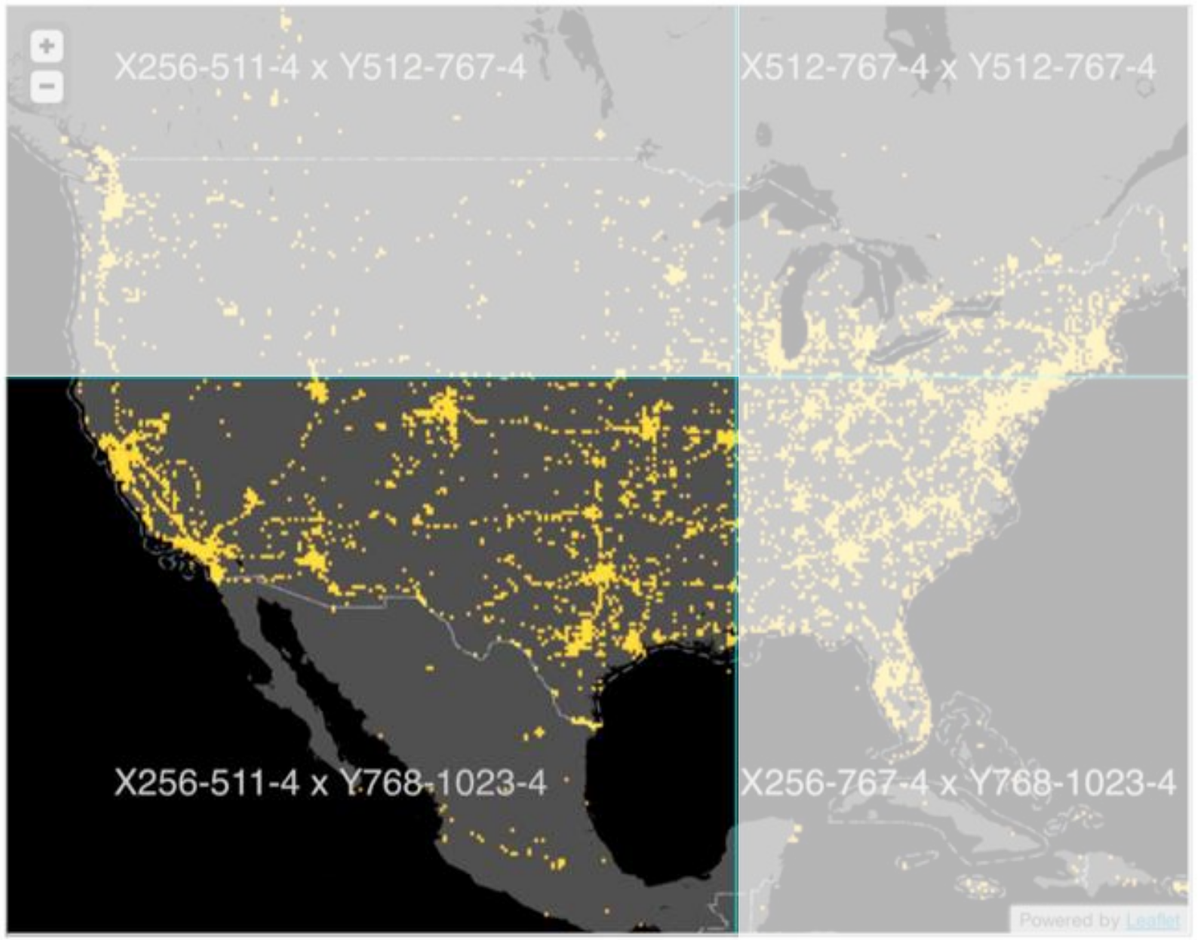

X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet


Month
Month0-11-0
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

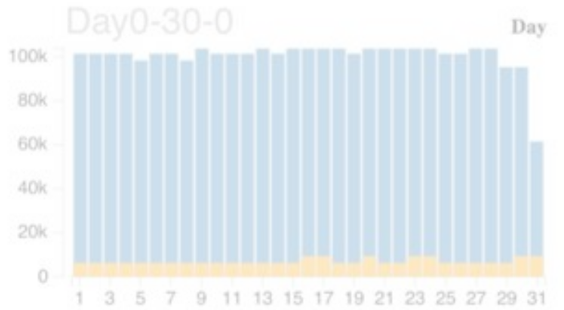

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

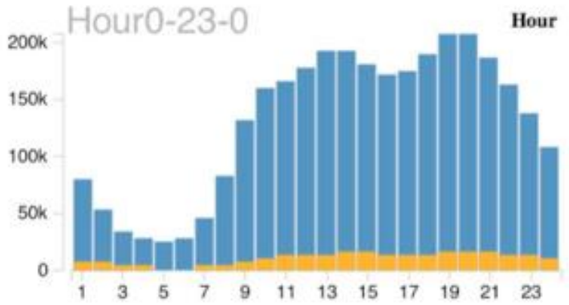

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23

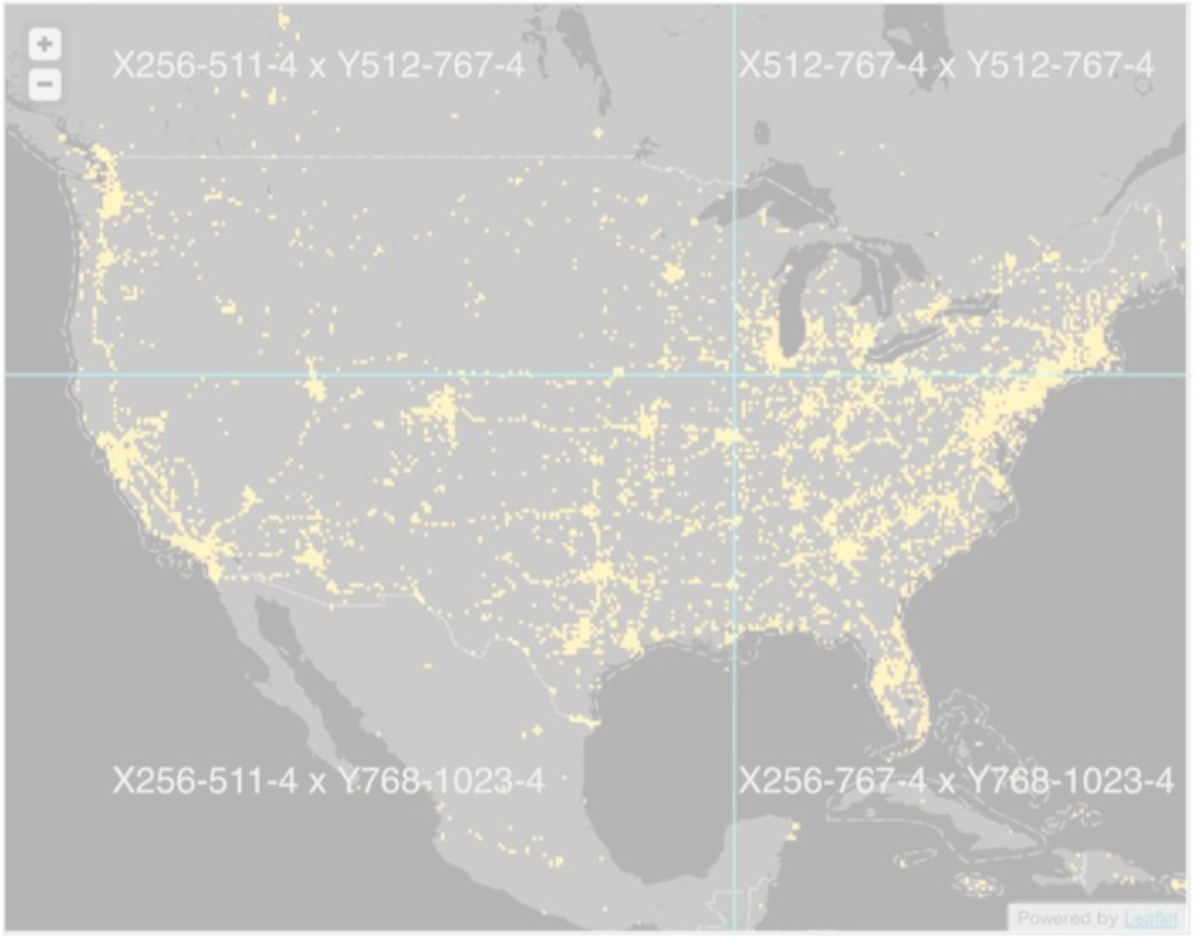

X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet
Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
Hour0-23-0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

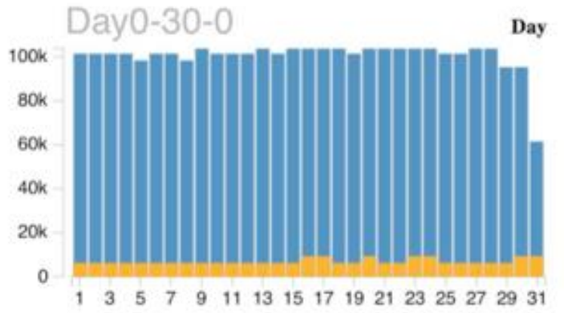

Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

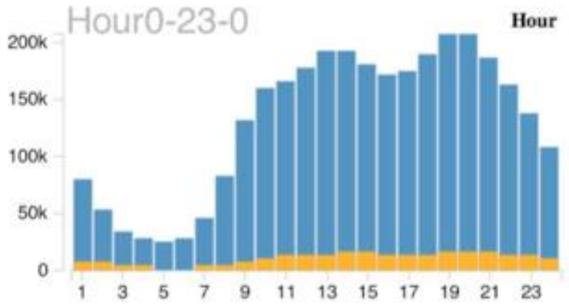

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

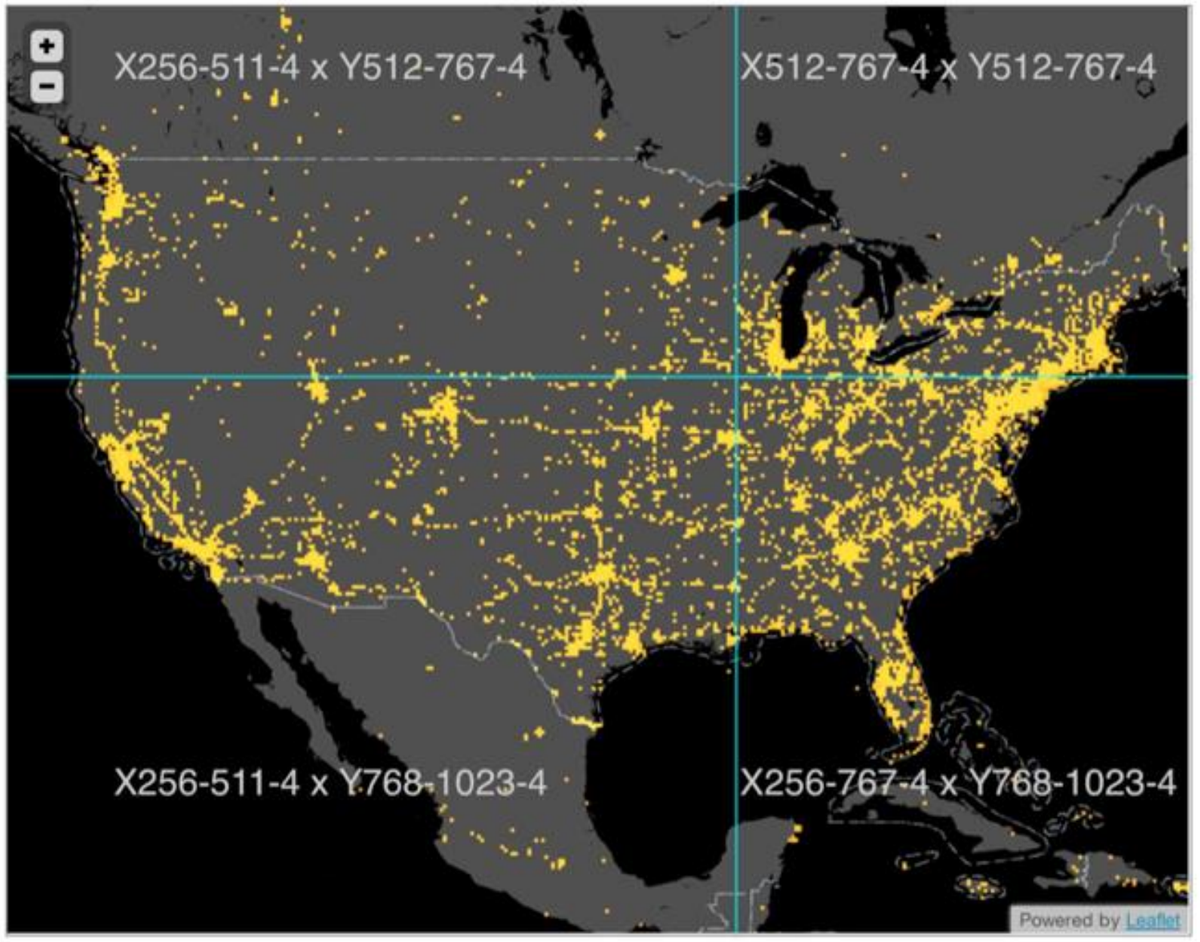

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23


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

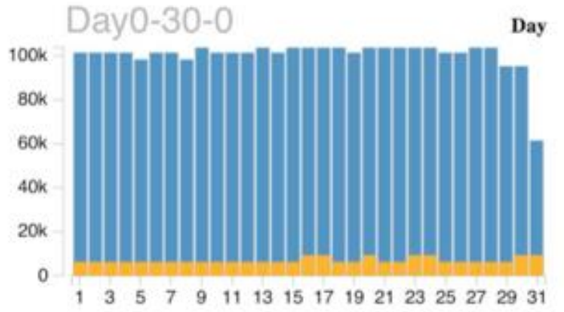

Month0-11-0
Month
300k
250k
200k
150k
100k
50k
0
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

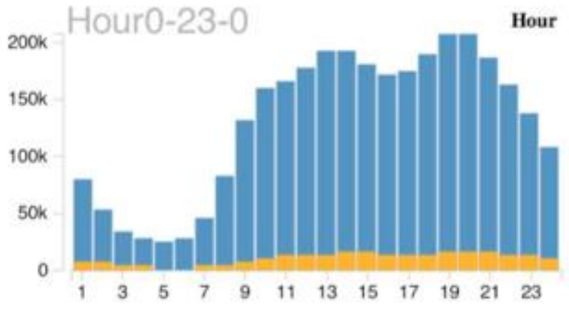

Day0-30-0
Day
100k
80k
60k
40k
20k
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31


Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23

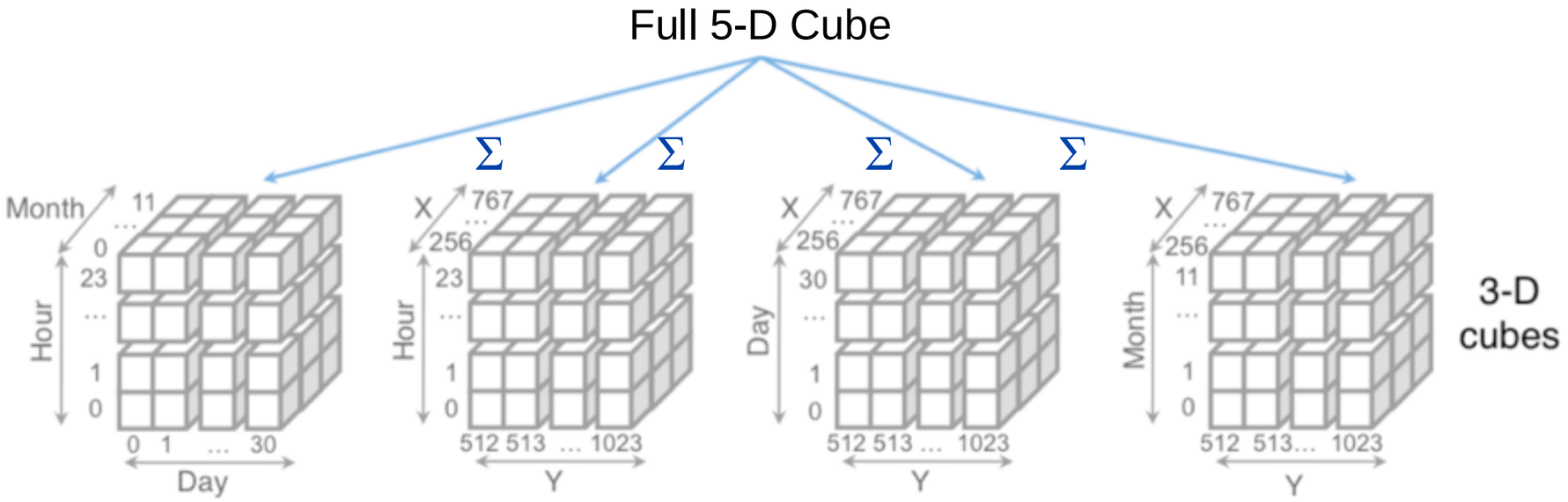

X256-511-4 x Y512-767-4
X512-767-4 x Y512-767-4
X256-511-4 x Y768-1023-4
X256-767-4 x Y768-1023-4
Powered by Leaflet


Month0-11-0
Month
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

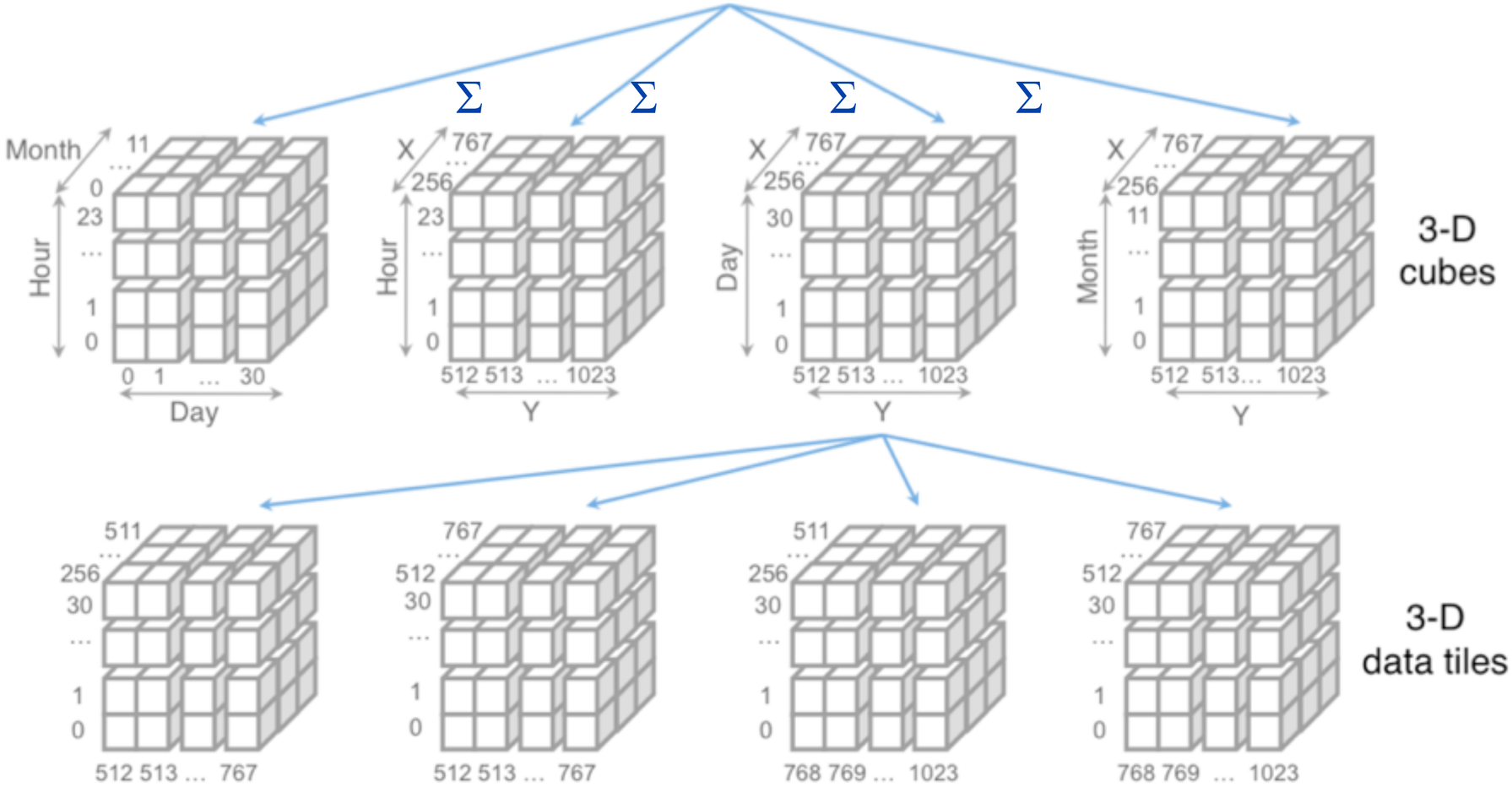

Hour0-23-0
Hour
200k
150k
100k
50k
0
1 3 5 7 9 11 13 15 17 19 21 23

Full 5-D Cube

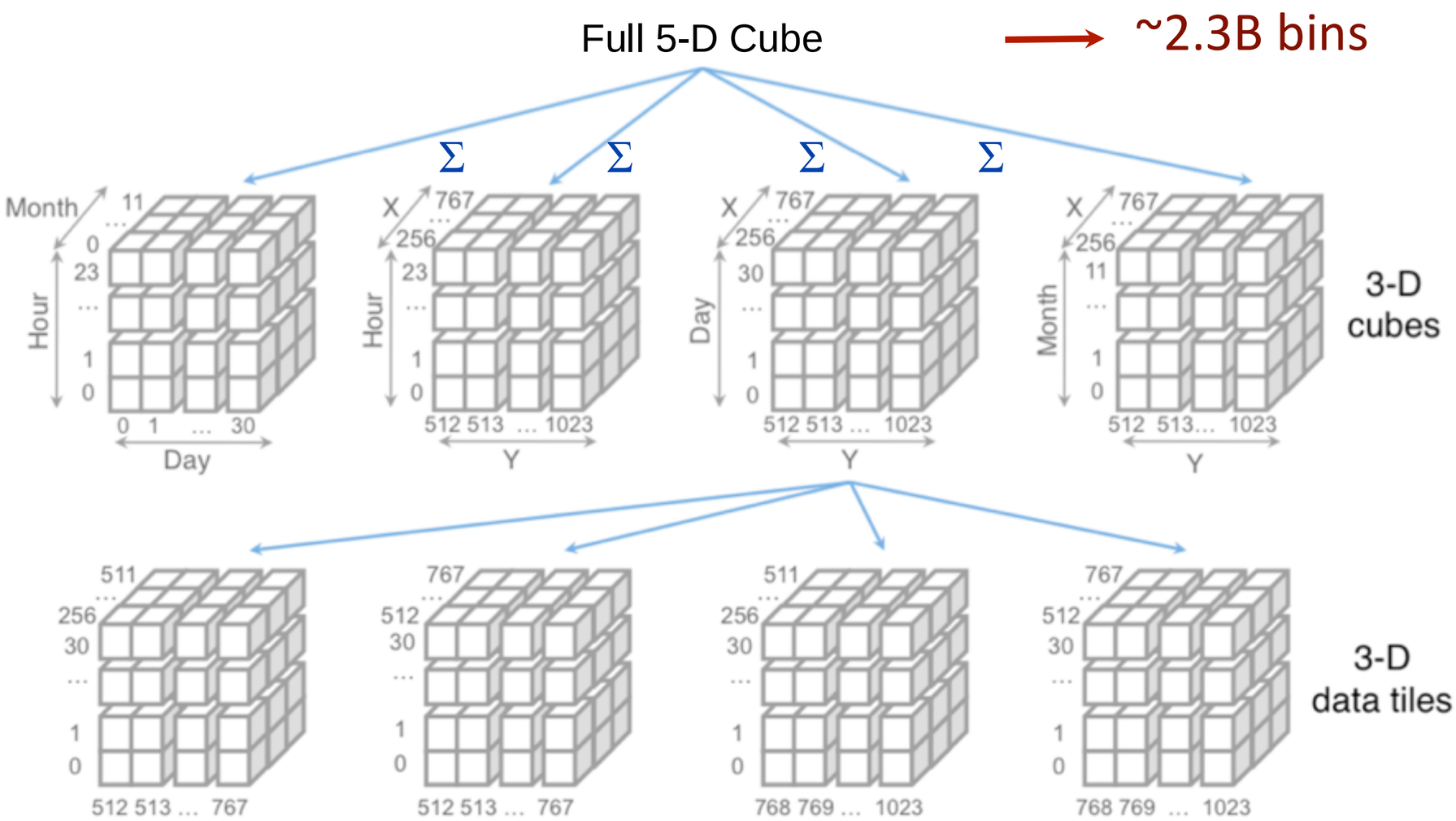


For any pair of 1D or 2D binned plots, the maximum number of dimensions needed to support brushing & linking is **four**.

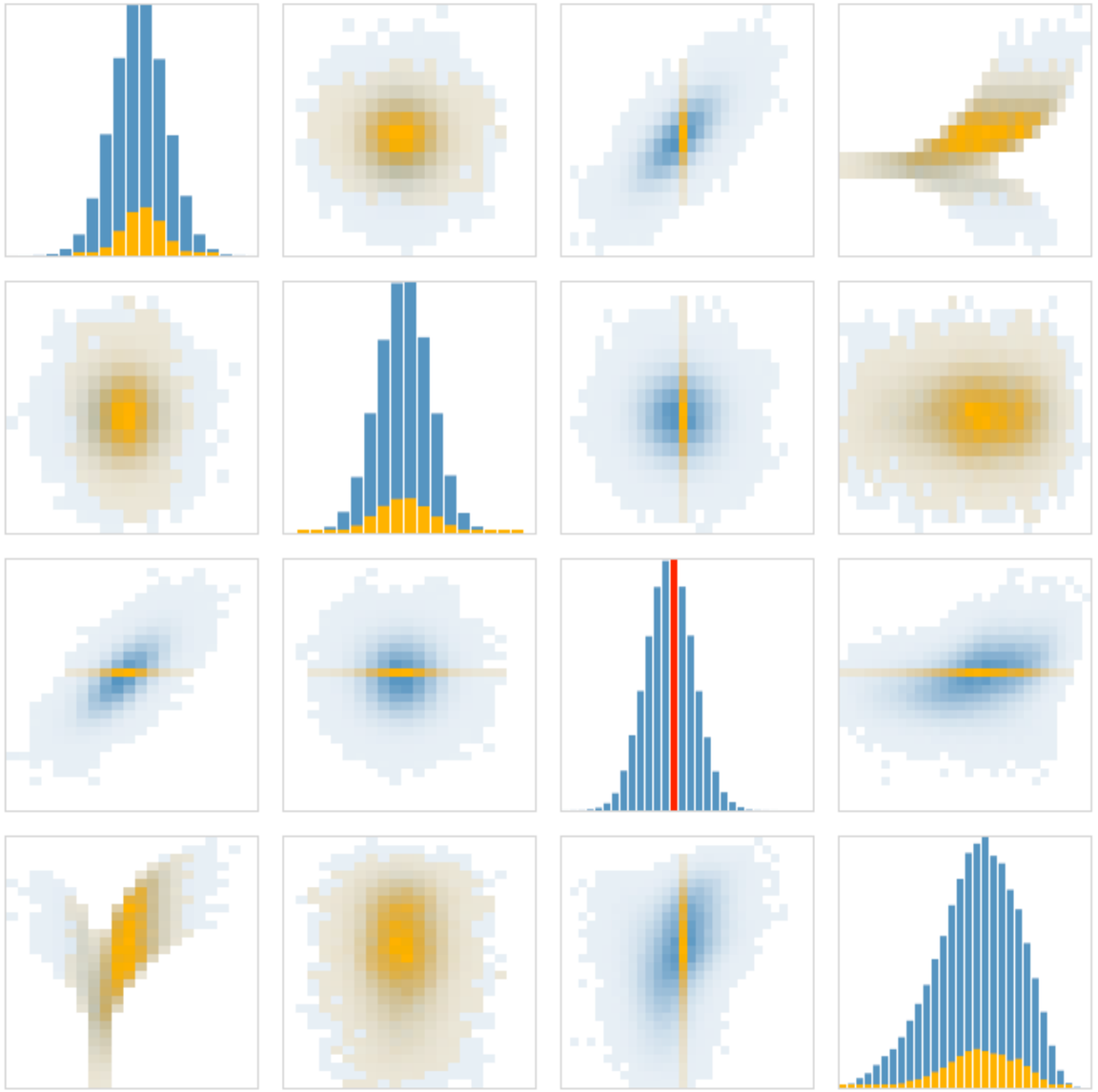
Full 5-D Cube



13 3-D Data Tiles


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

Performance Benchmarks



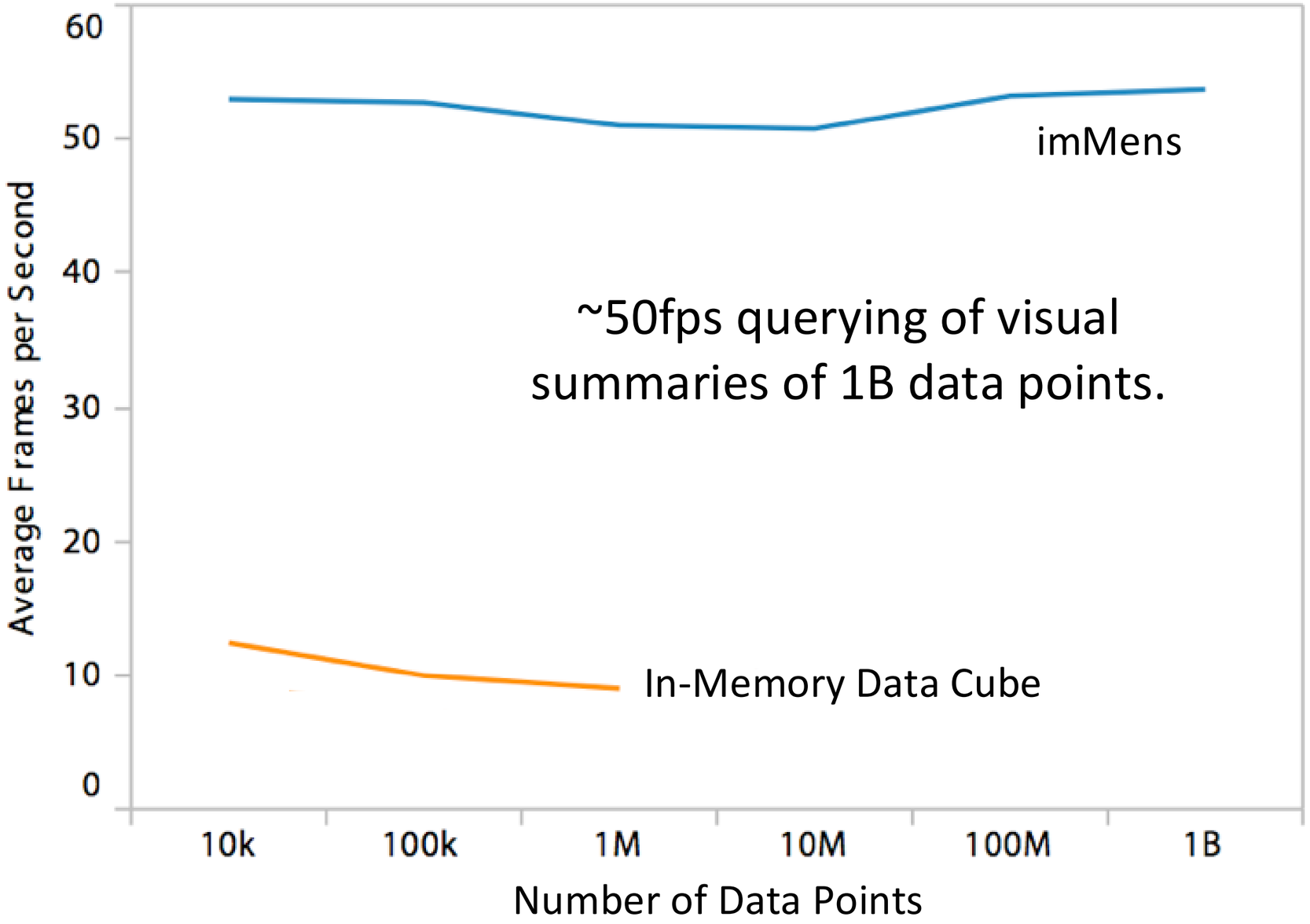
Simulate interaction:
brushing & linking
across binned plots.

- 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



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? (Short answer: yes!)

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

How does interactive latency affect exploratory analysis with visualizations?

[Liu & Heer ’14]

Prior Work – Negatives to Latency

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 – Positives to Latency

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 – Positives to Latency

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*?

Addressed by Liu & Heer.

Experiment Design

2 (Latency) x 2 (Scenario) Design

Latency: +0ms / +500ms

Scenario: Mobile Check-ins / FAA Flight Delays

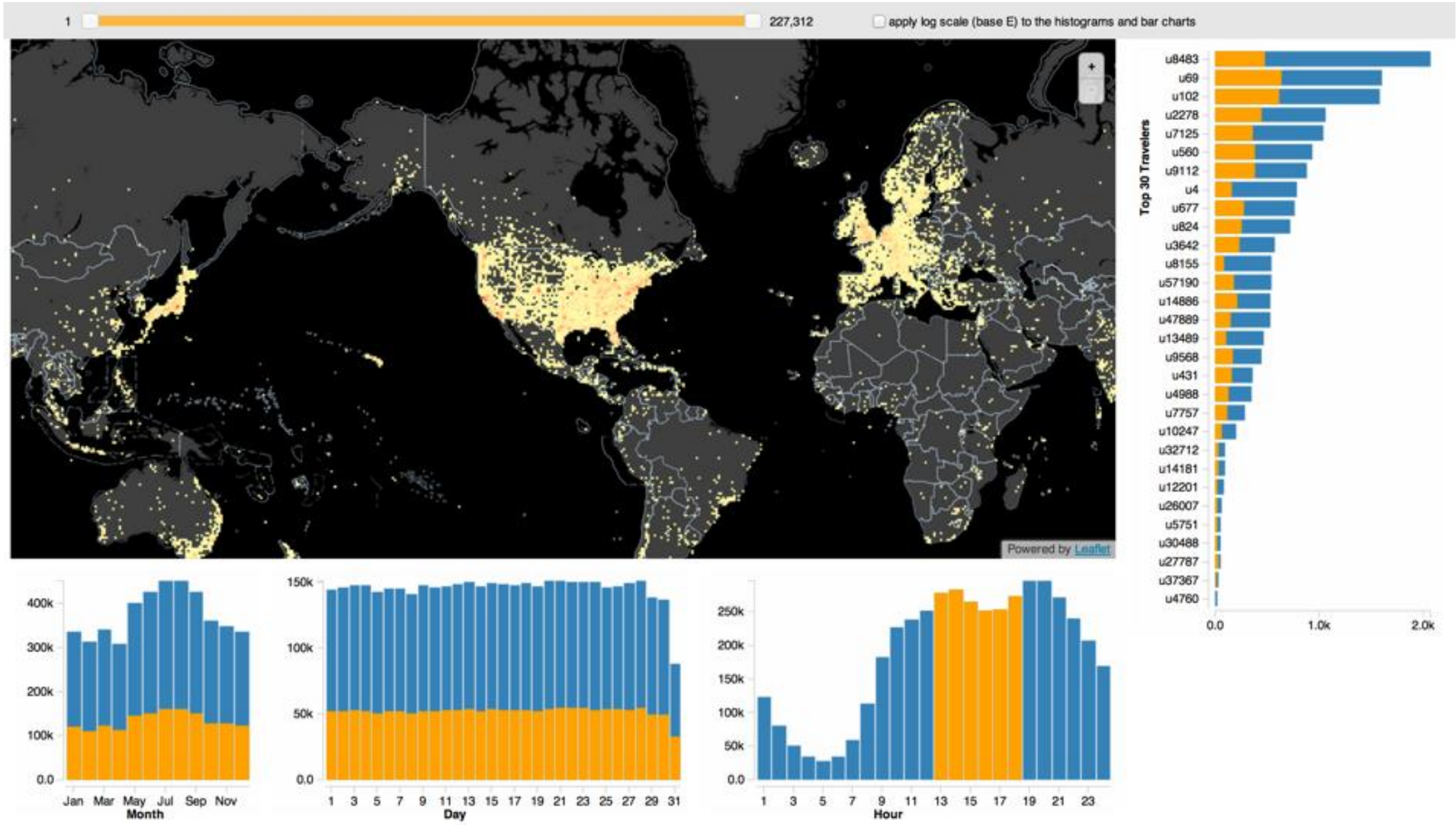
Exploratory Analysis Tasks (2 per session)

imMens with brush, pan, zoom, adjust scales

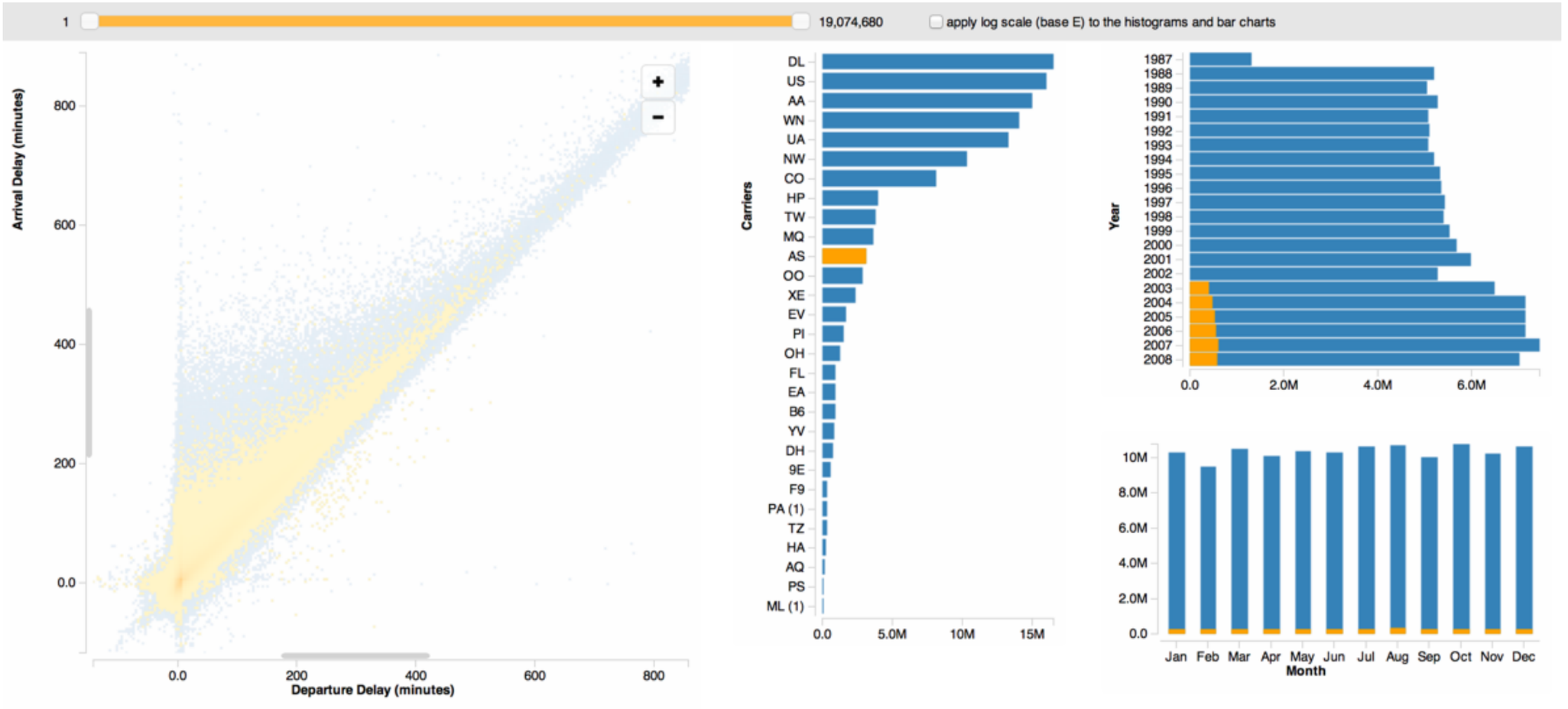
Users asked to explore data and share findings

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

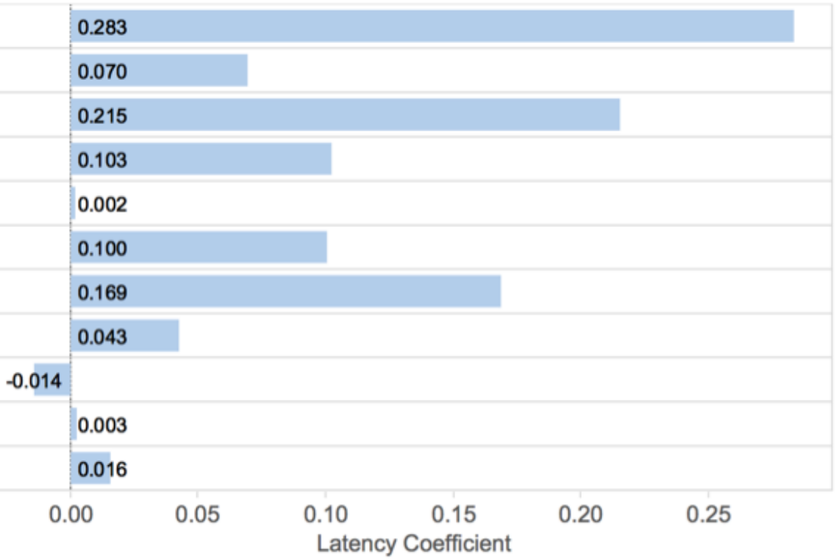
Transcribe sessions; Code actions and insights

Analyze number and type of coded events

Latency Study Results

Higher latency leads to…
Reduced user activity and data set coverage
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

Less observation, generalization & hypothesis

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

Less observation, generalization & hypothesis

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

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

Break Time!

Administrivia

Final Project Schedule

~~*Proposal Wed Feb 19*~~

Prototype ***Tues Mar 4***

Demo Video Tue Mar 11

Video Showcase Thu Mar 13 (in class)

Deliverables Tue Mar 18

No late days!!!

Milestone Prototype

Publish work to Gitlab pages for us to examine and share feedback. You **are not** expected to have complete, polished content at this point.

You **are** expected to provide prototype work that communicates your design goals. For example: initial visualizations, sketches, storyboards, and text annotations / idea descriptions.

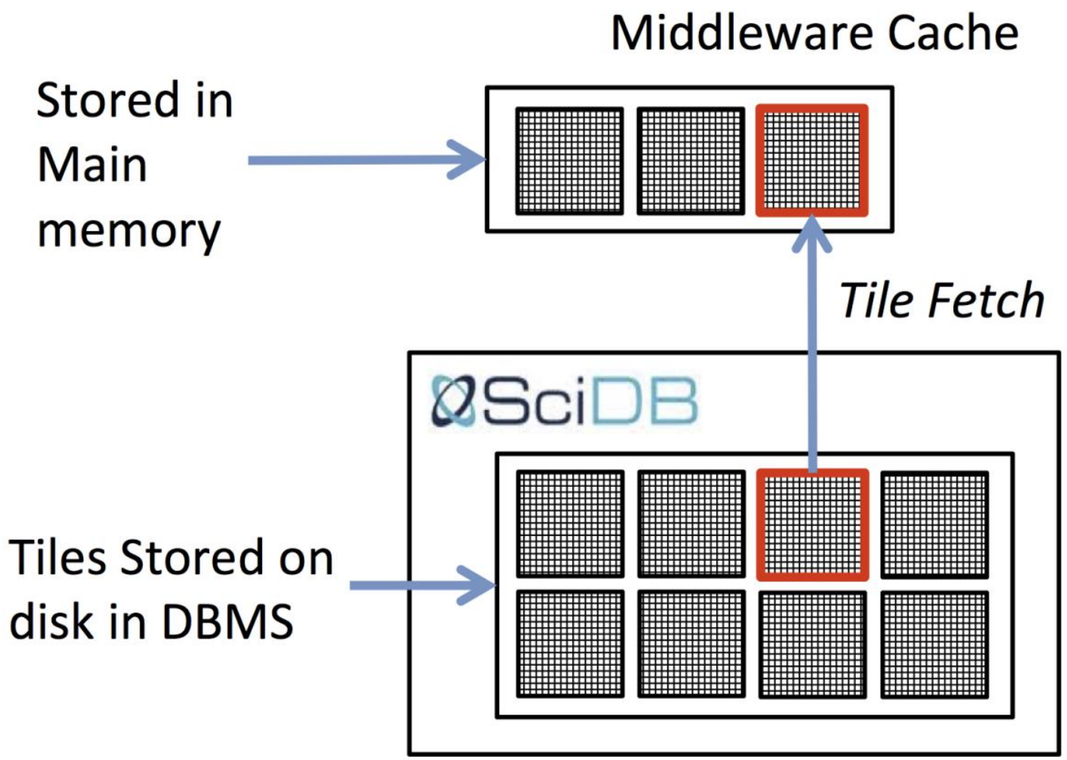
*We should get a sense of what you intend to ultimately submit! Also feel free to ask **us** questions.*

ForeCache

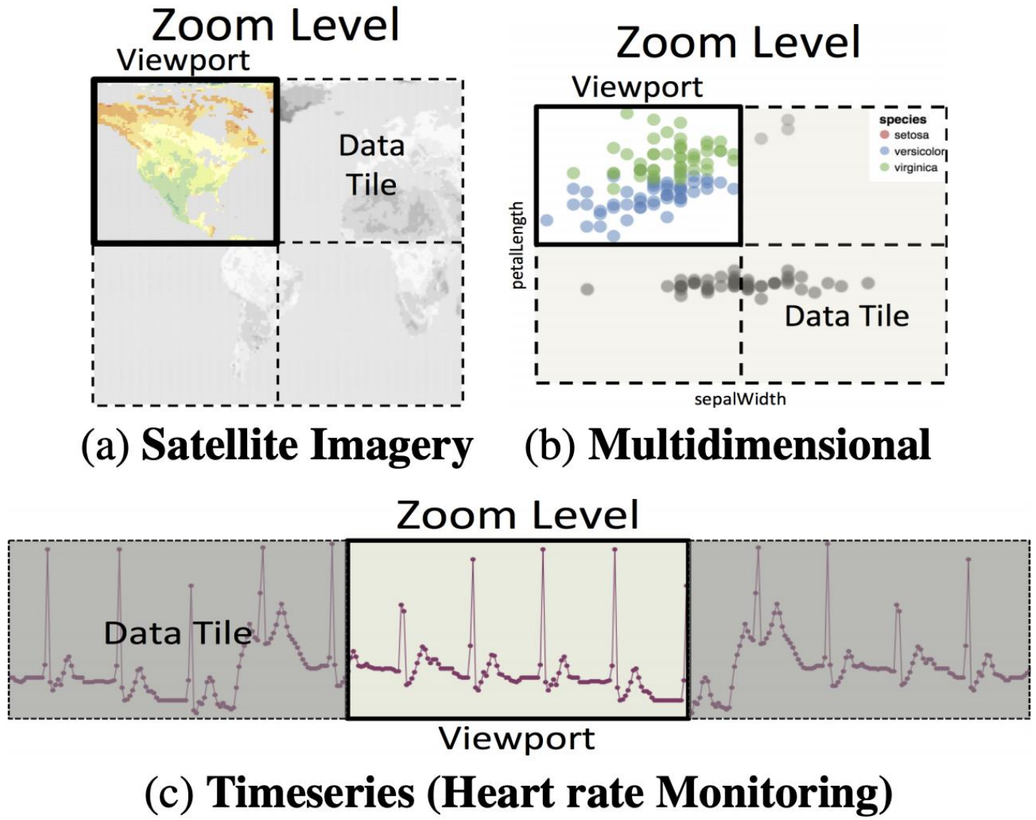
[Battle, Chang, & Stonebraker ’16]

Strategies: Query Database, Data Cubes, Prefetching

ForeCache is also a Data Tile-Based System



Manage a Cache of Tiles from DBMS



Example Tile-Based Views

Key Idea: Model & Predict User Behavior

1. Classify the User's Analysis Phase

Foraging: Searching for patterns of interest

Sensemaking: Closely examine a region-of-interest (ROI)

Navigation: Transition between levels of detail

2. Predict Which Data Tiles Will be Requested

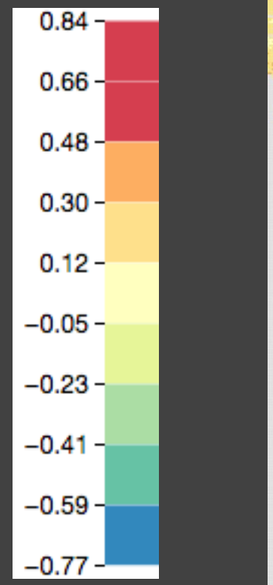
Train a machine learning classifier (SVM) to predict phase.

The input data is the activity trace of user interactions.

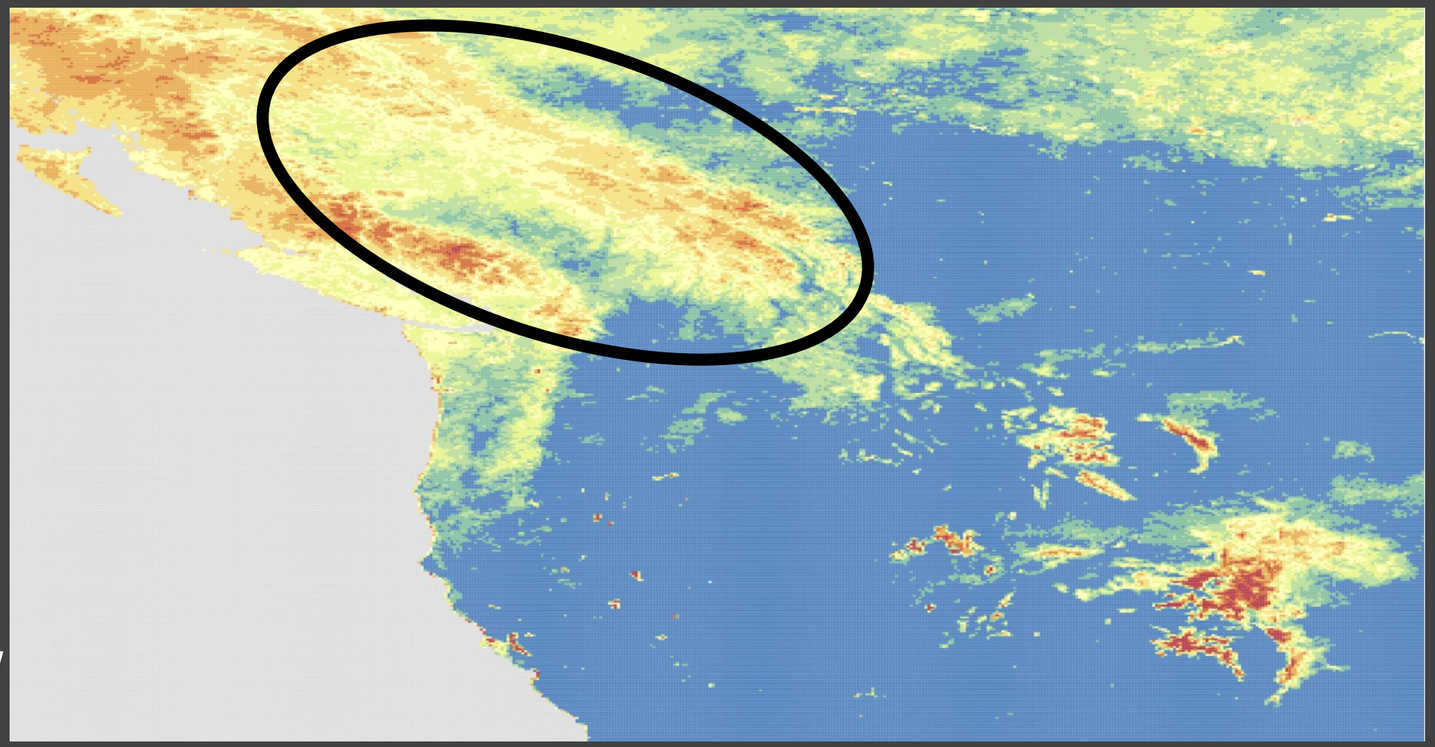
Applying the Three Phases to Exploration Scenarios

Foraging

Snow



No Snow

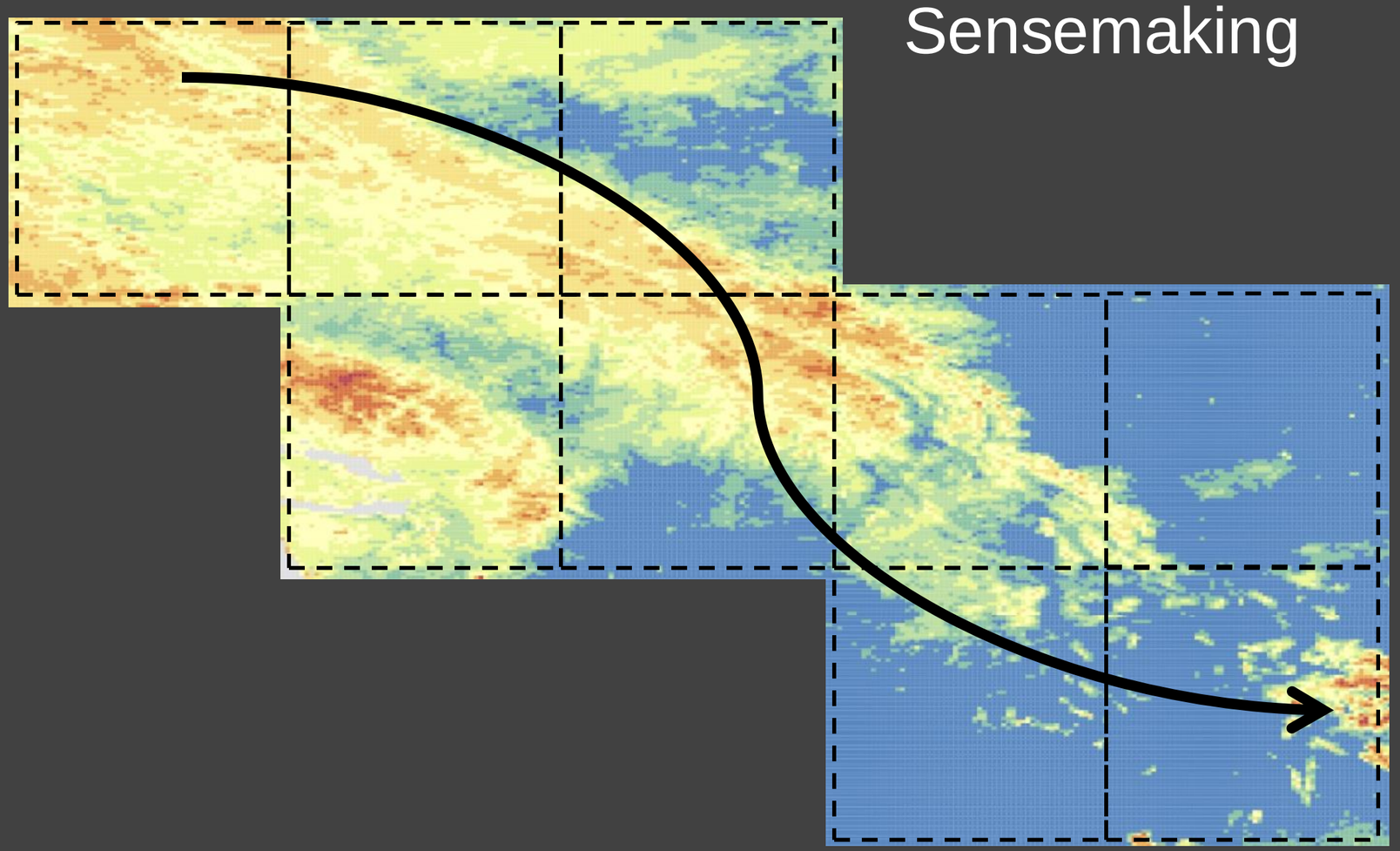


Applying the Three Phases to Exploration Scenarios

Navigation

User zooms in

Applying the Three Phases to Exploration Scenarios



Applying the Three Phases to Exploration Scenarios

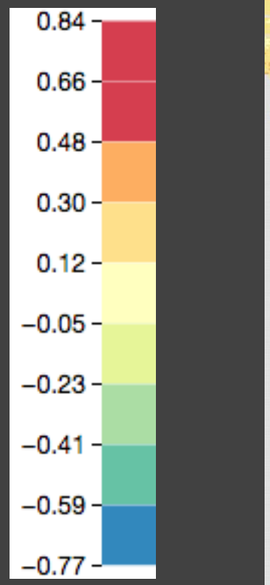
Navigation

User zooms out

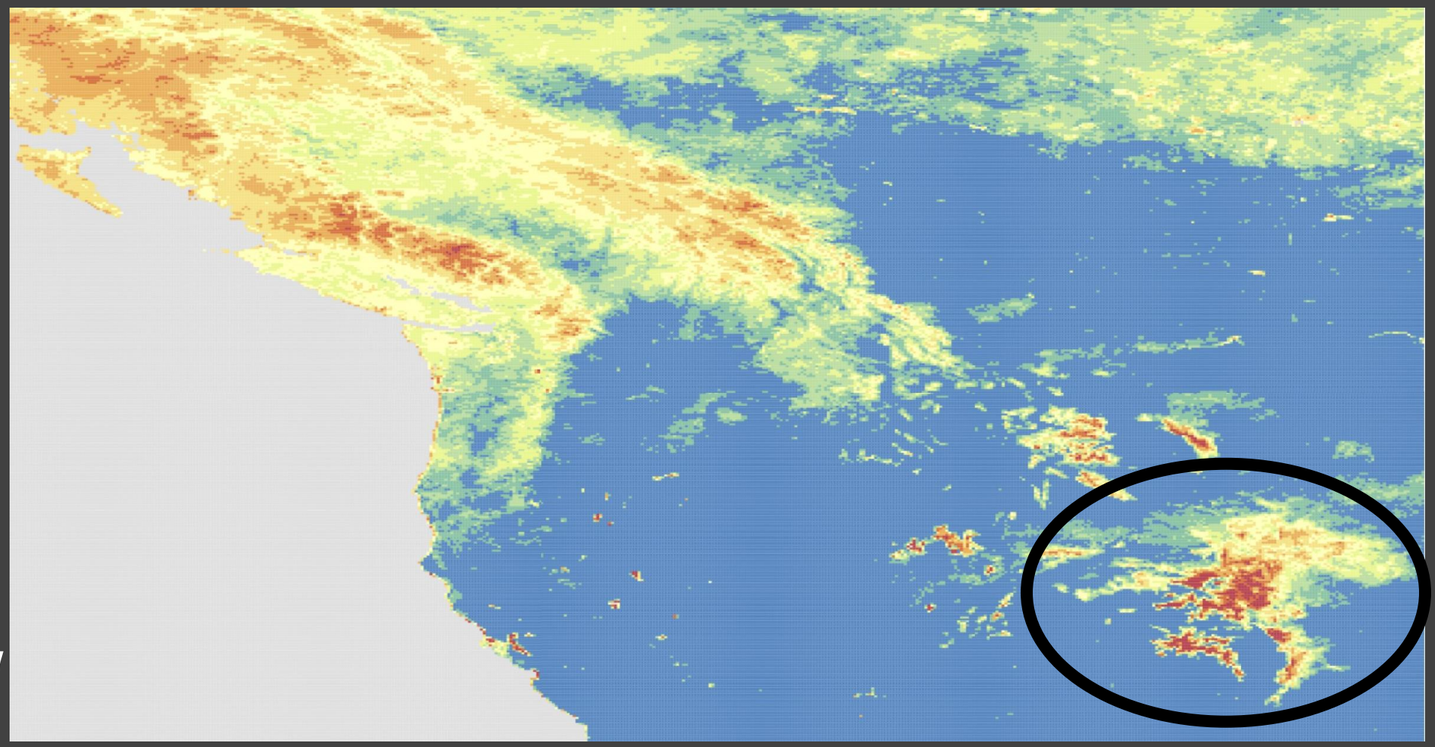
Applying the Three Phases to Exploration Scenarios

Foraging

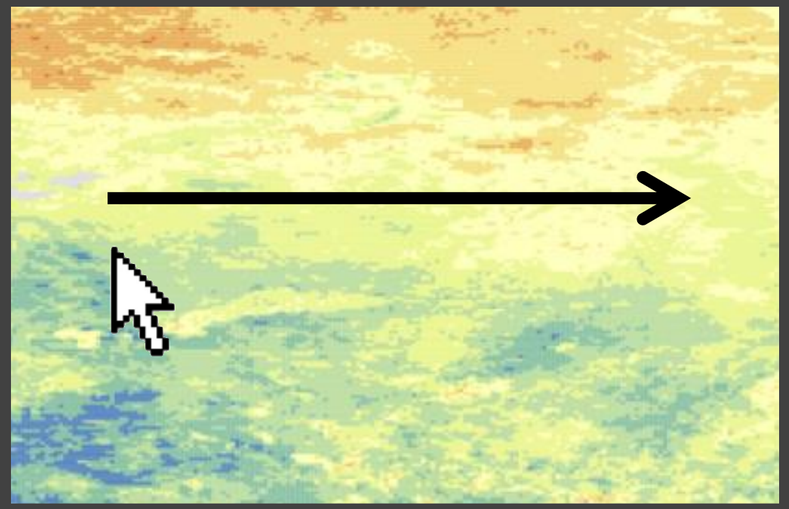
Snow

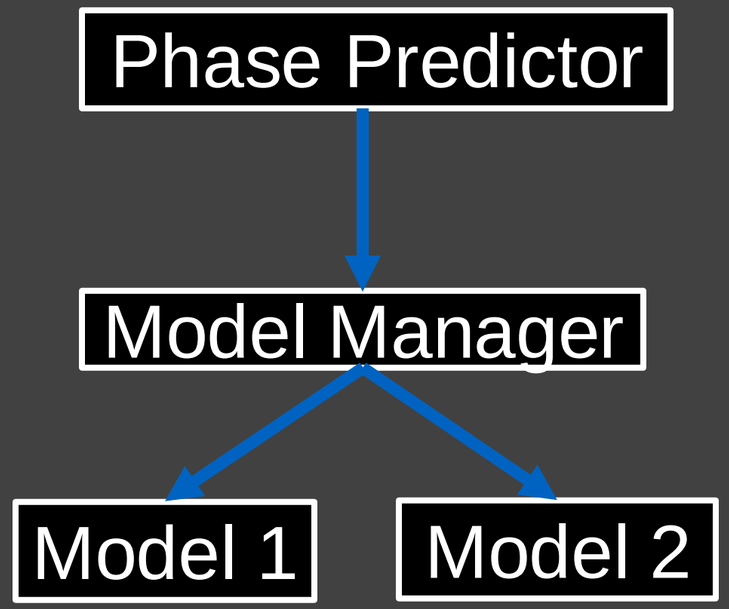


No Snow

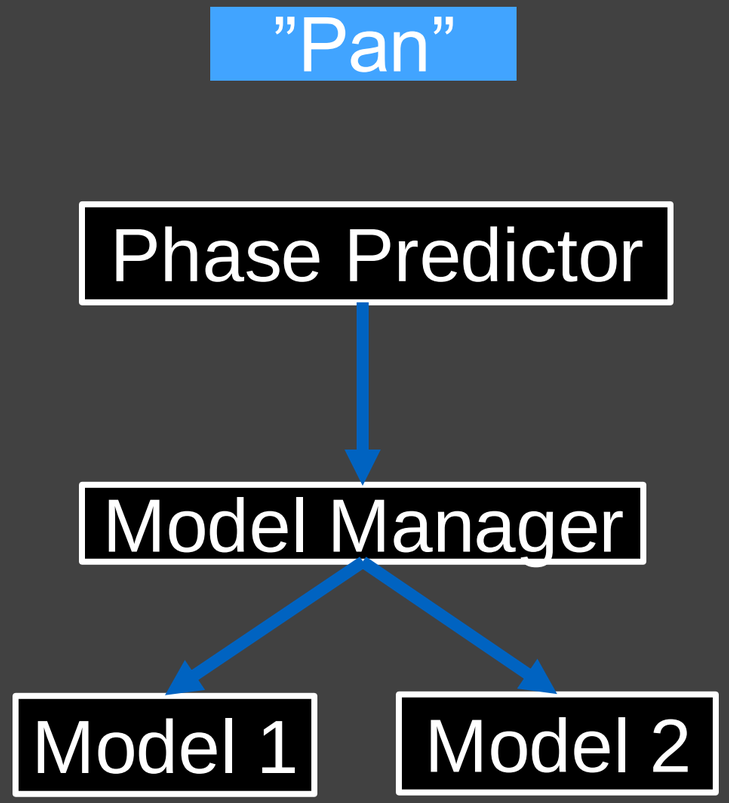


Using Phases to Predict Tiles

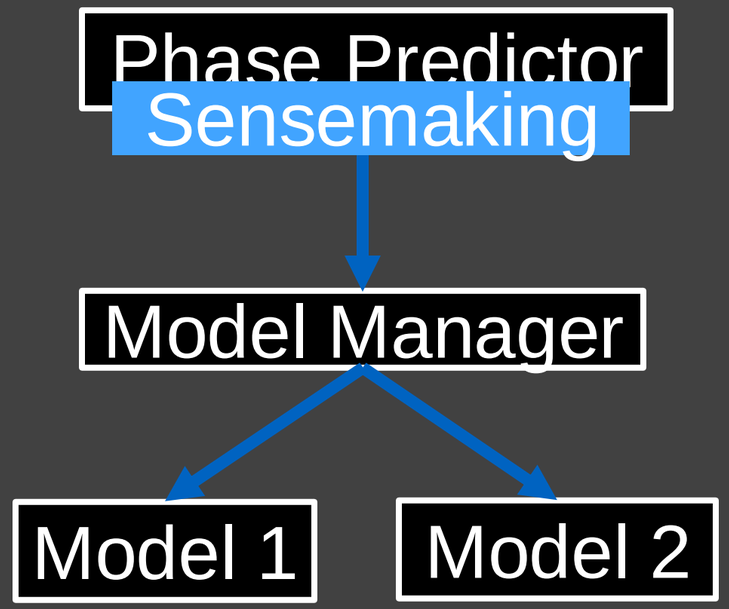




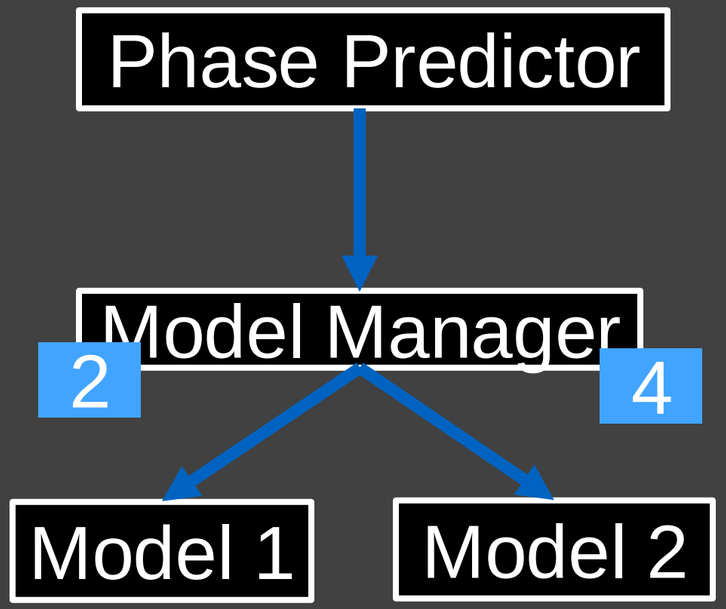
Using Phases to Predict Tiles



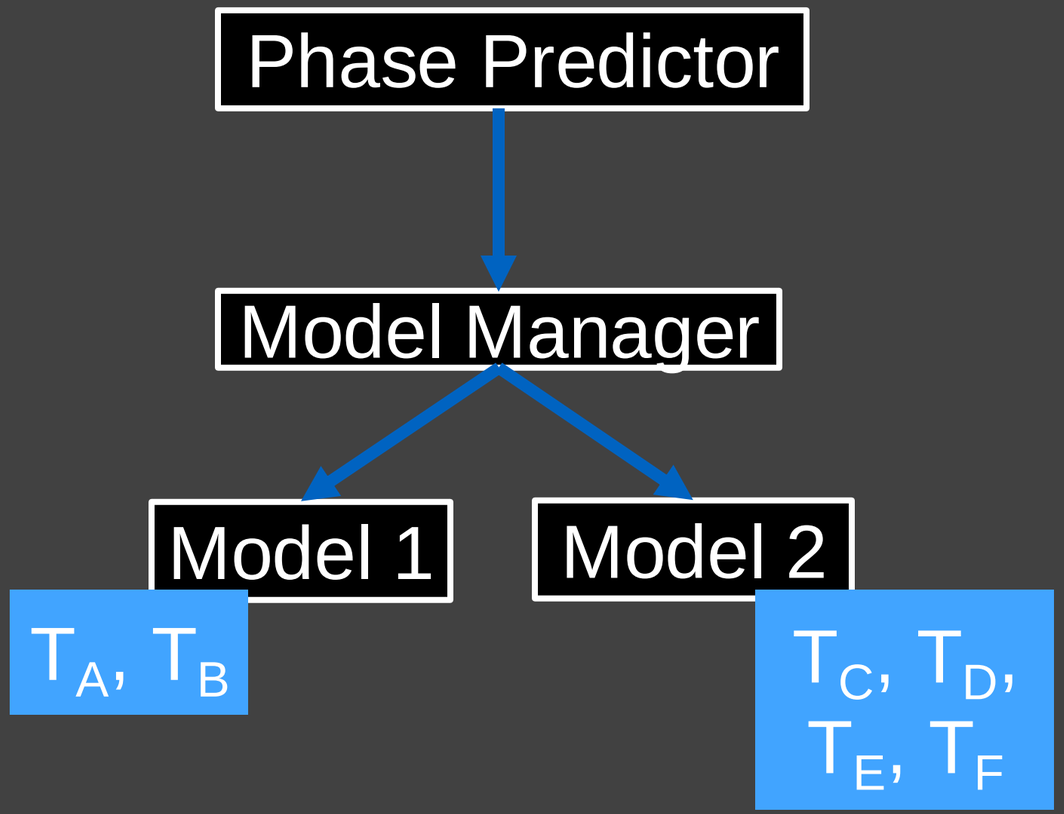
Using Phases to Predict Tiles



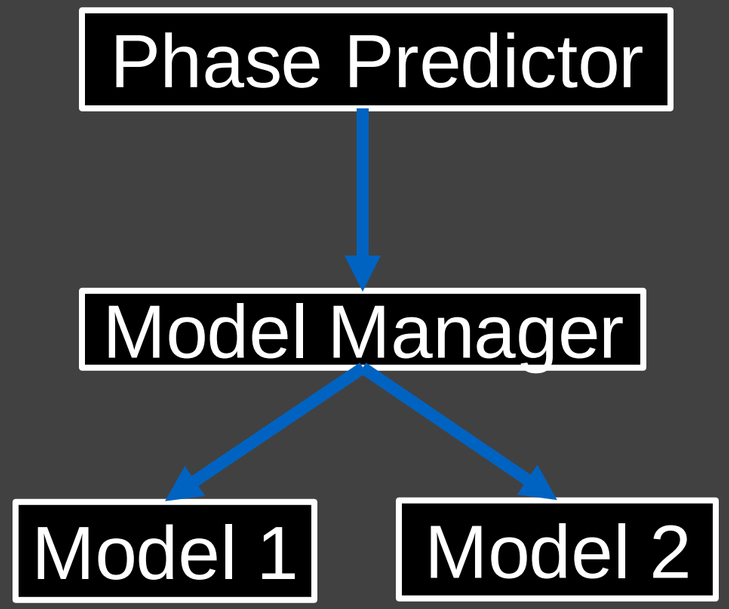
Using Phases to Predict Tiles



Using Phases to Predict Tiles

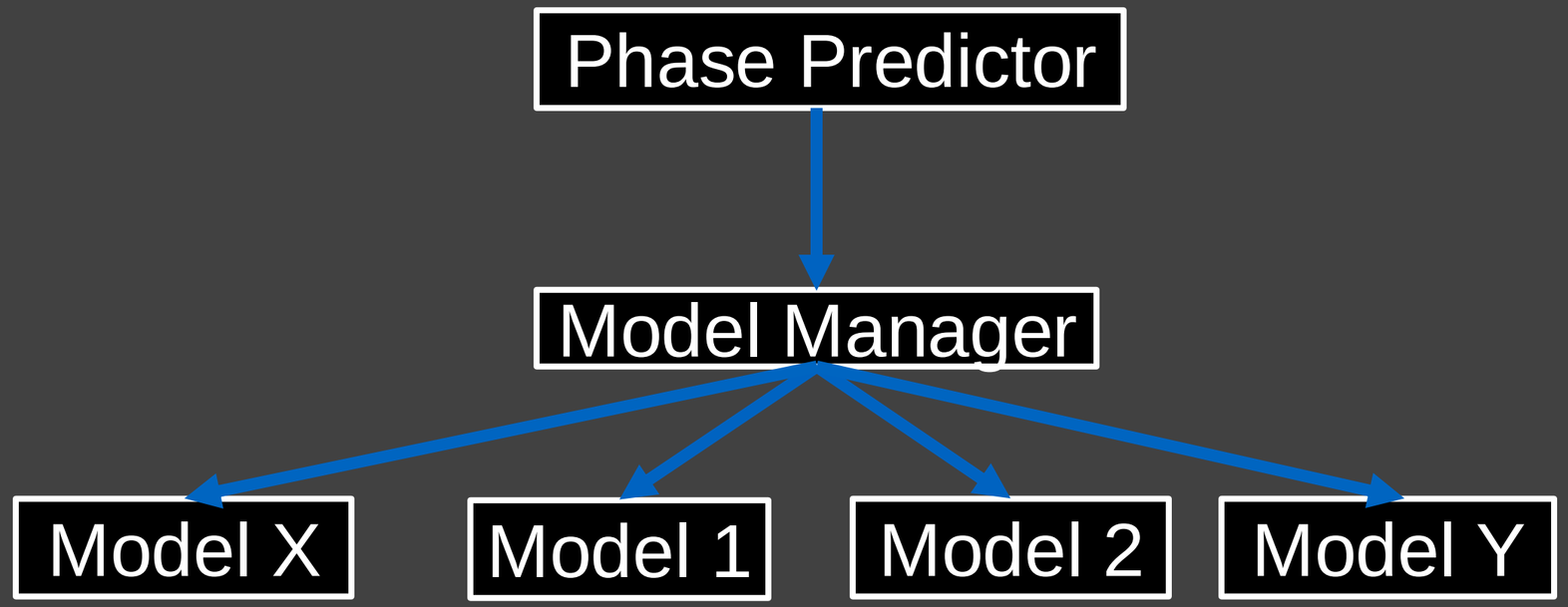


Using Phases to Predict Tiles



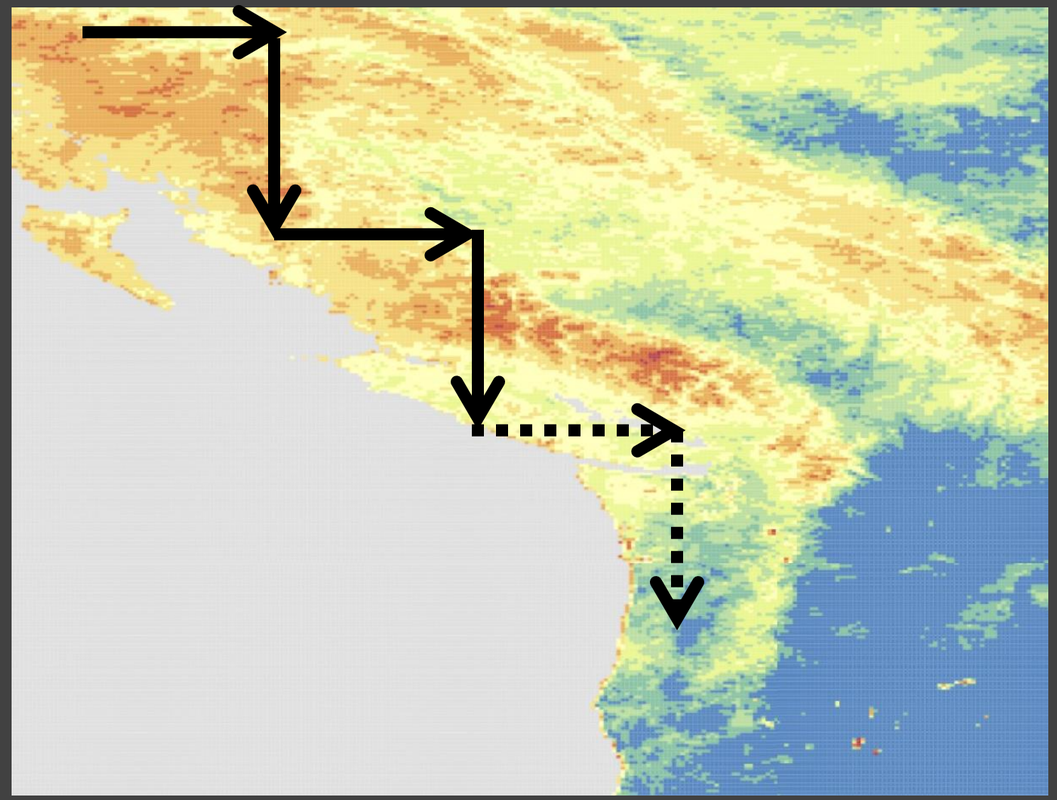
T_A, T_B,
T_C, T_D,
T_E, T_F

Using Phases to Predict Tiles



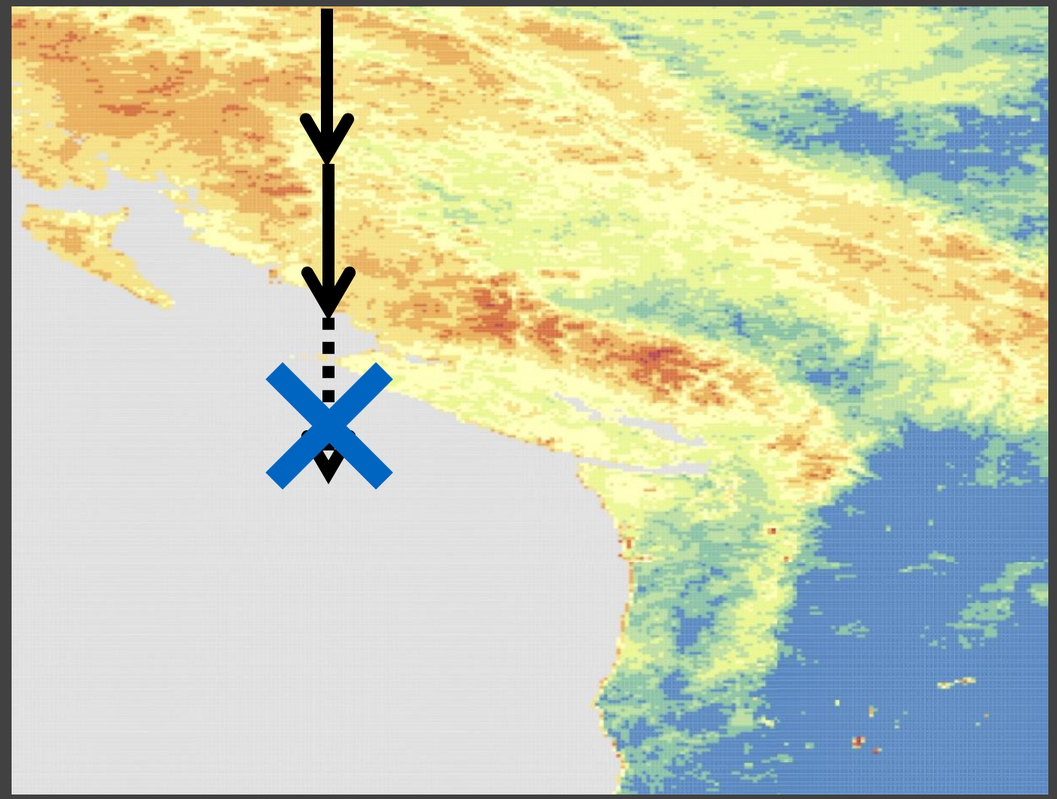
Action-Based Tile Recommendations

Idea: user consistently moves in predictable directions



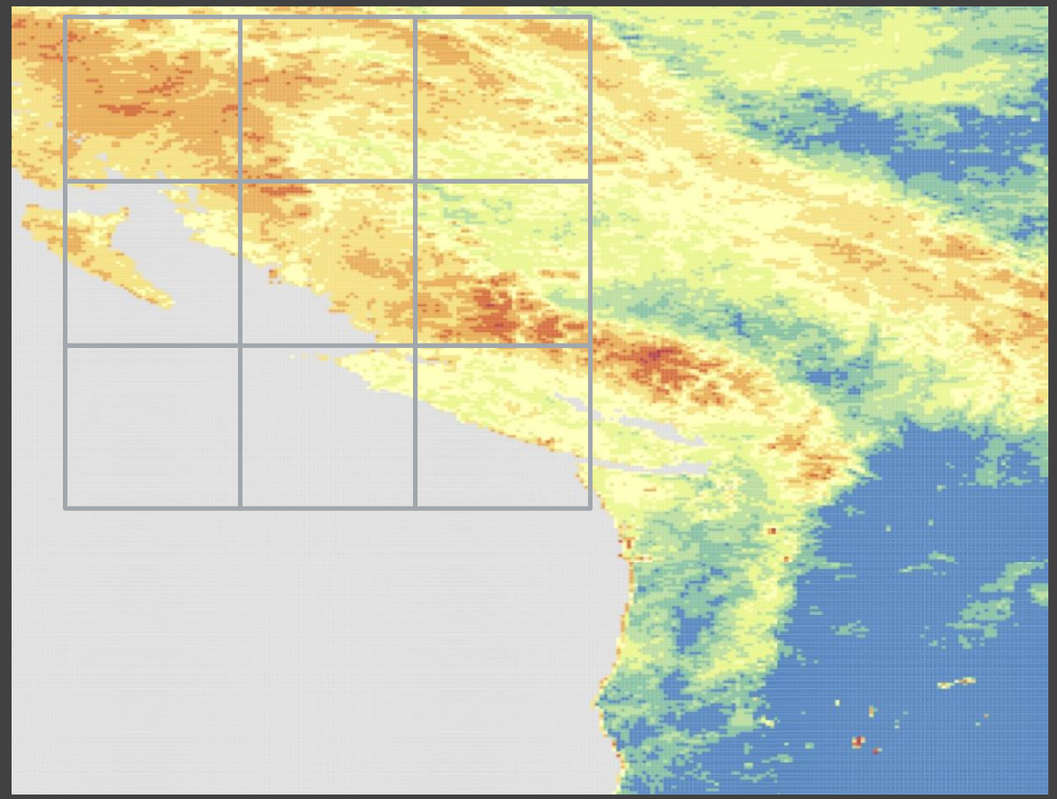
Signature-Based Tile Recommendations

Idea: user wants to see more of the same thing



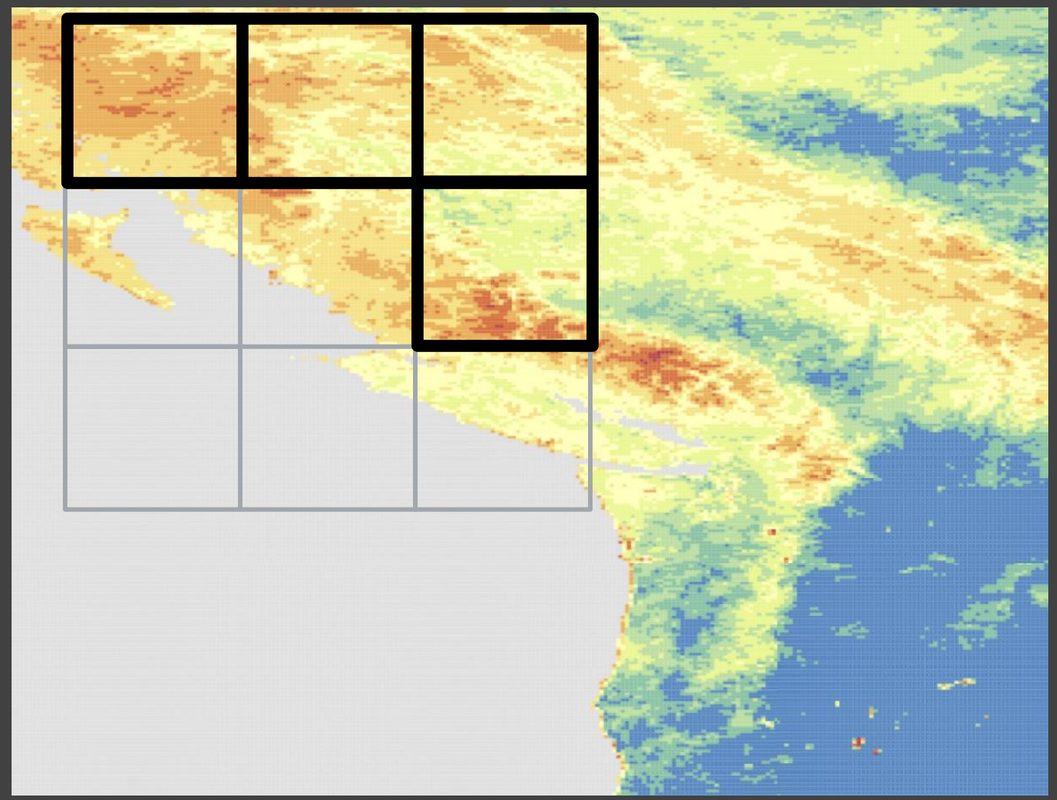
Signature-Based Tile Recommendations

Idea: user wants to see more of the same thing



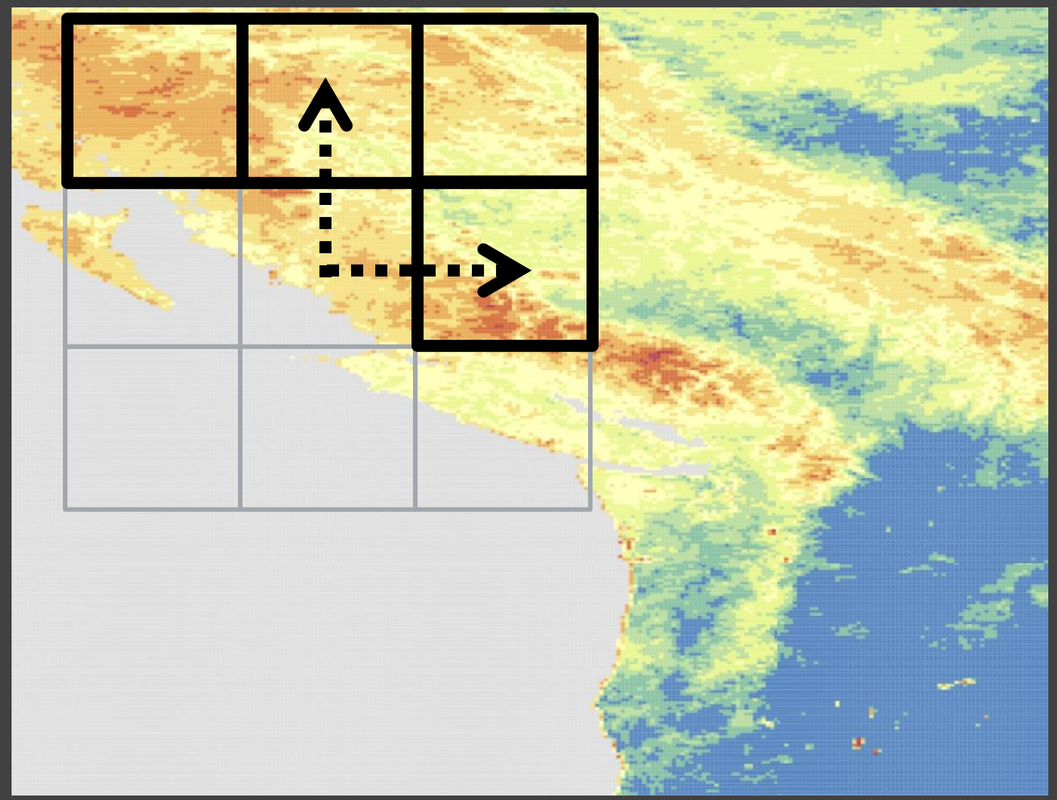
Signature-Based Tile Recommendations

Idea: user wants to see more of the same thing



Signature-Based Tile Recommendations

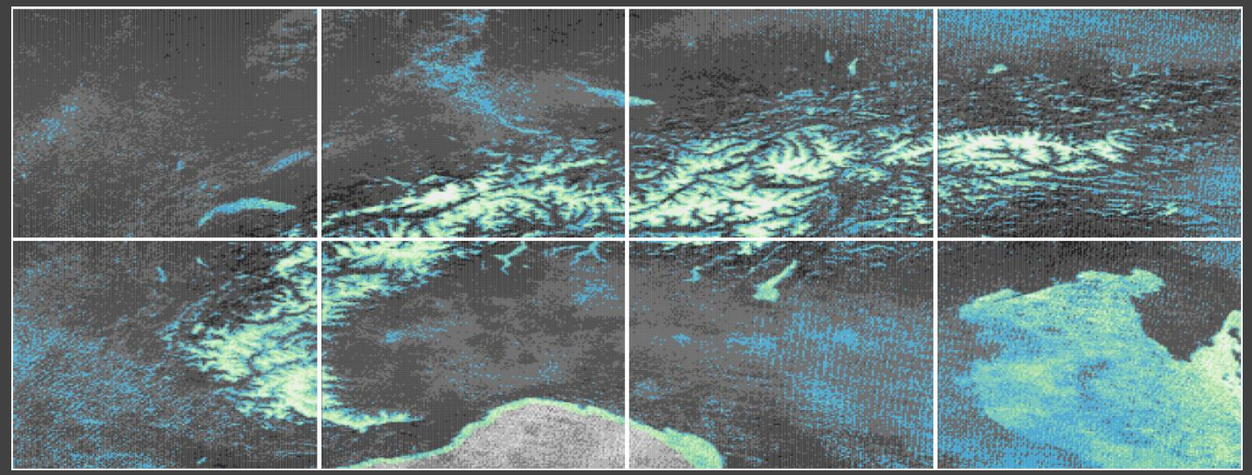
Idea: user wants to see more of the same thing

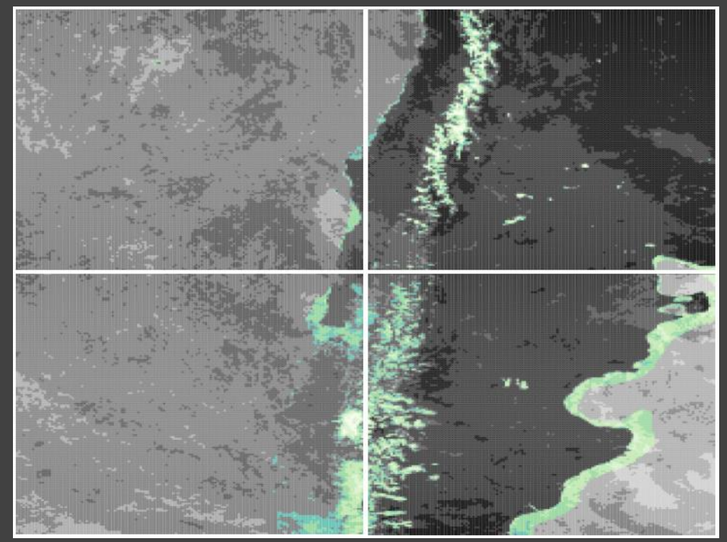


Evaluating ForeCache: A User Study

Participants: 18 earth science researchers

Explored NASA MODIS snow cover queries

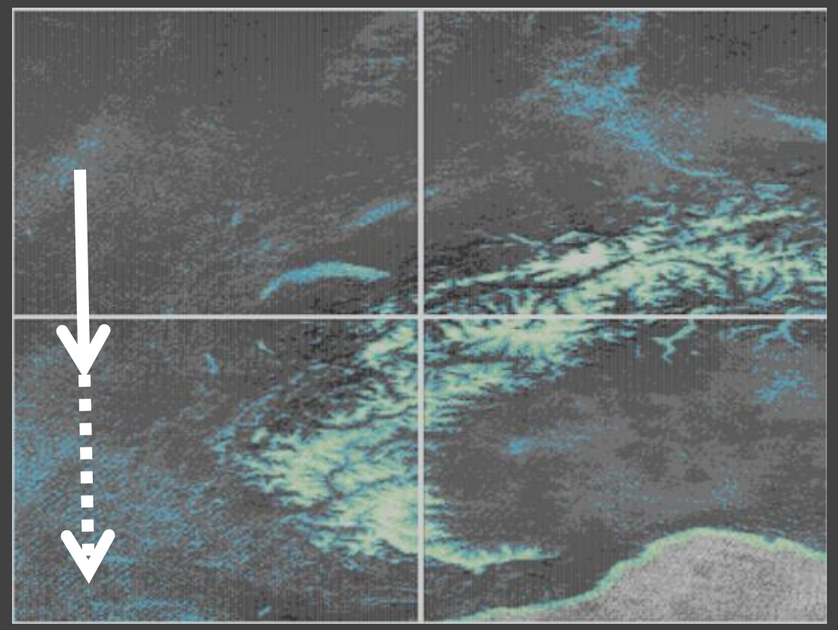




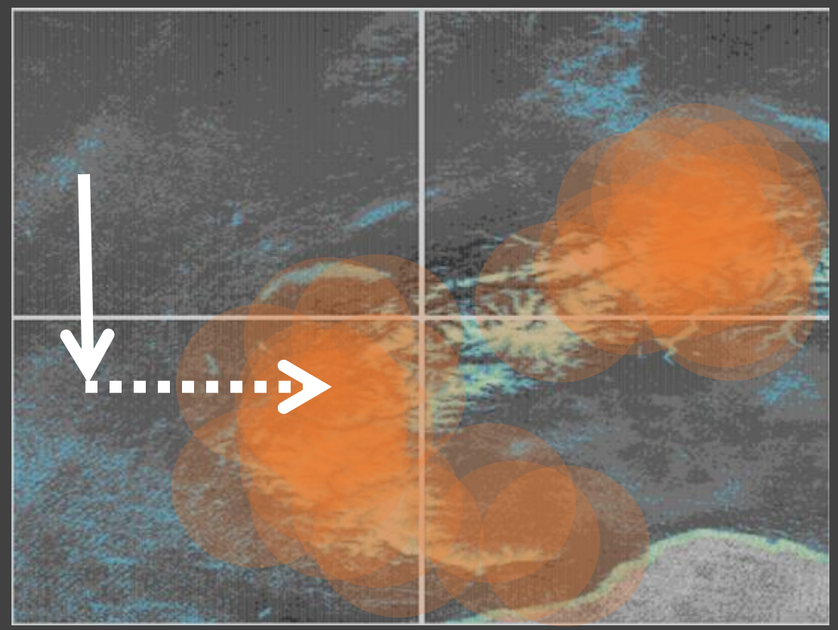
Retrospective Performance Experiments

Compared response times and prediction accuracy to a non-prefetching baseline and two existing pre-fetching methods:

Momentum

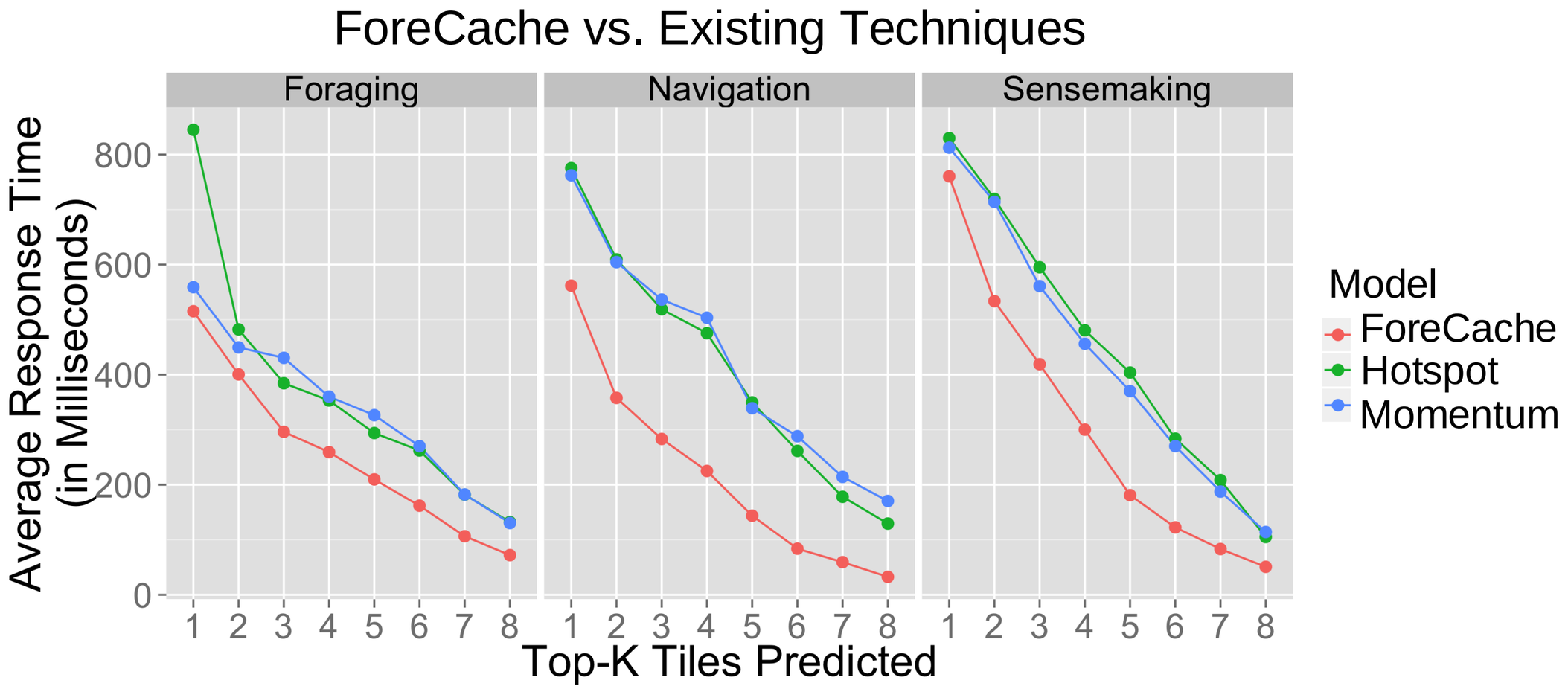


Hotspot



[Doshi et al. 2003]

Results: ForeCache was 20% More Accurate and 88% Faster than Existing Pre-fetching Methods

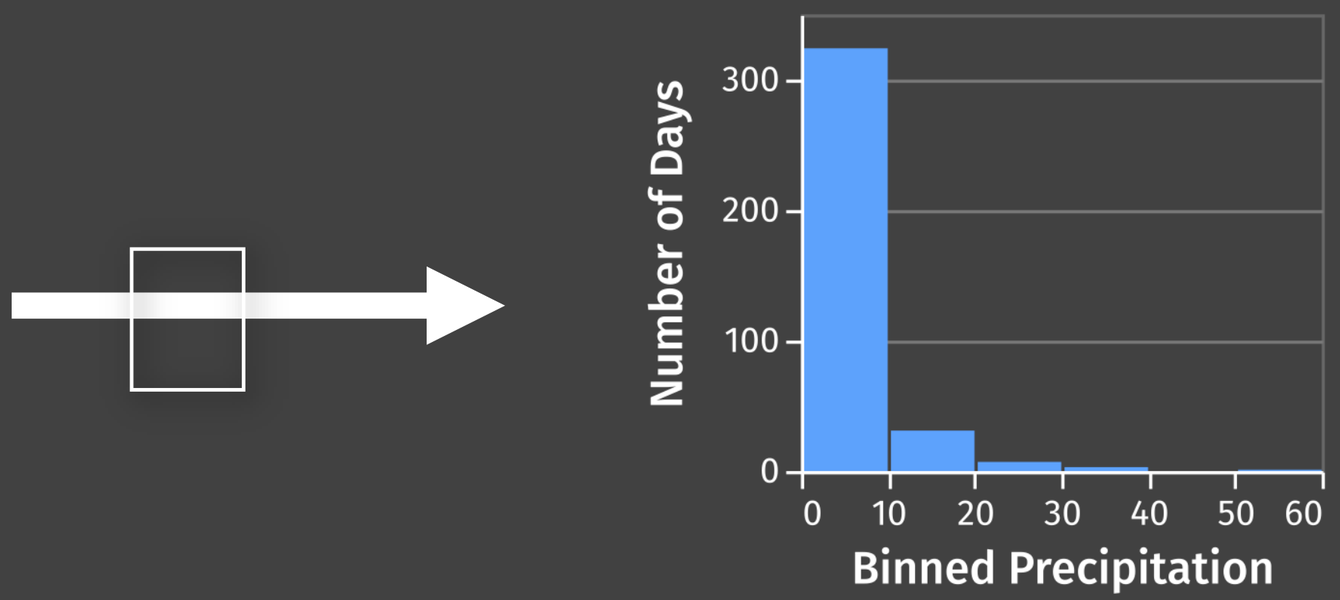


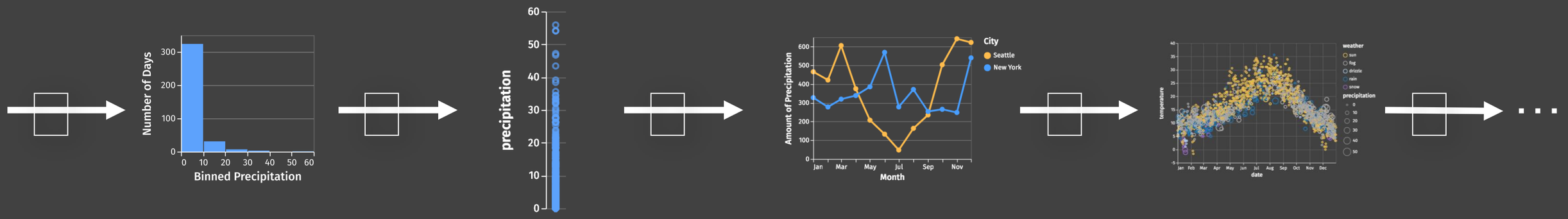
What if the data is too large to query in a reasonable time?

Trust, but Verify: Optimistic Vis

[Moritz, Fisher, Ding & Wang ’17]

Strategies: Query Database, Approximation


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.*

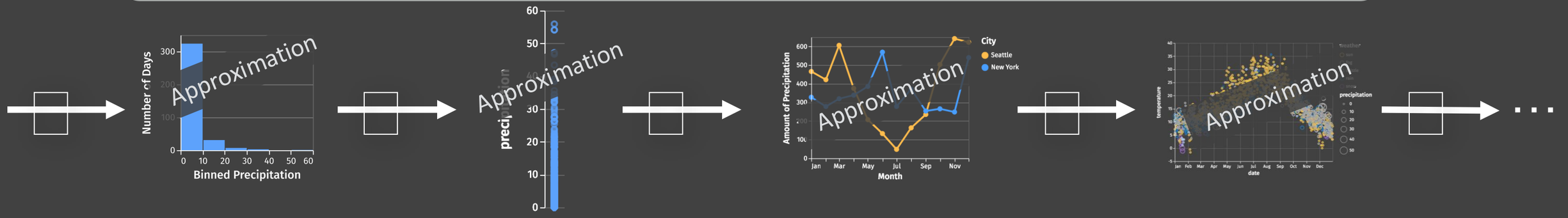
Small chance of error

Small chance of error

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?

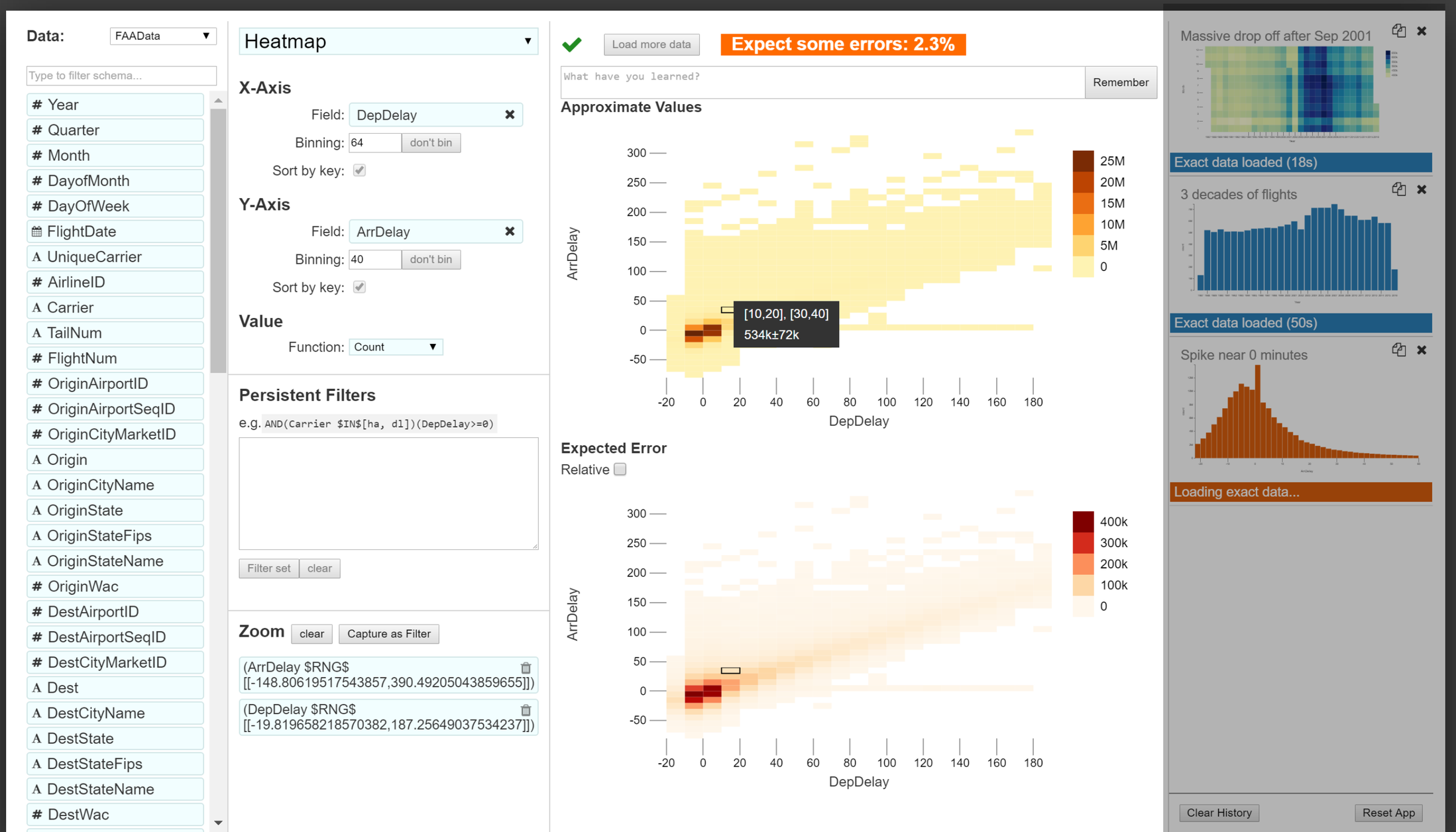
Optimistic Visualization

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.

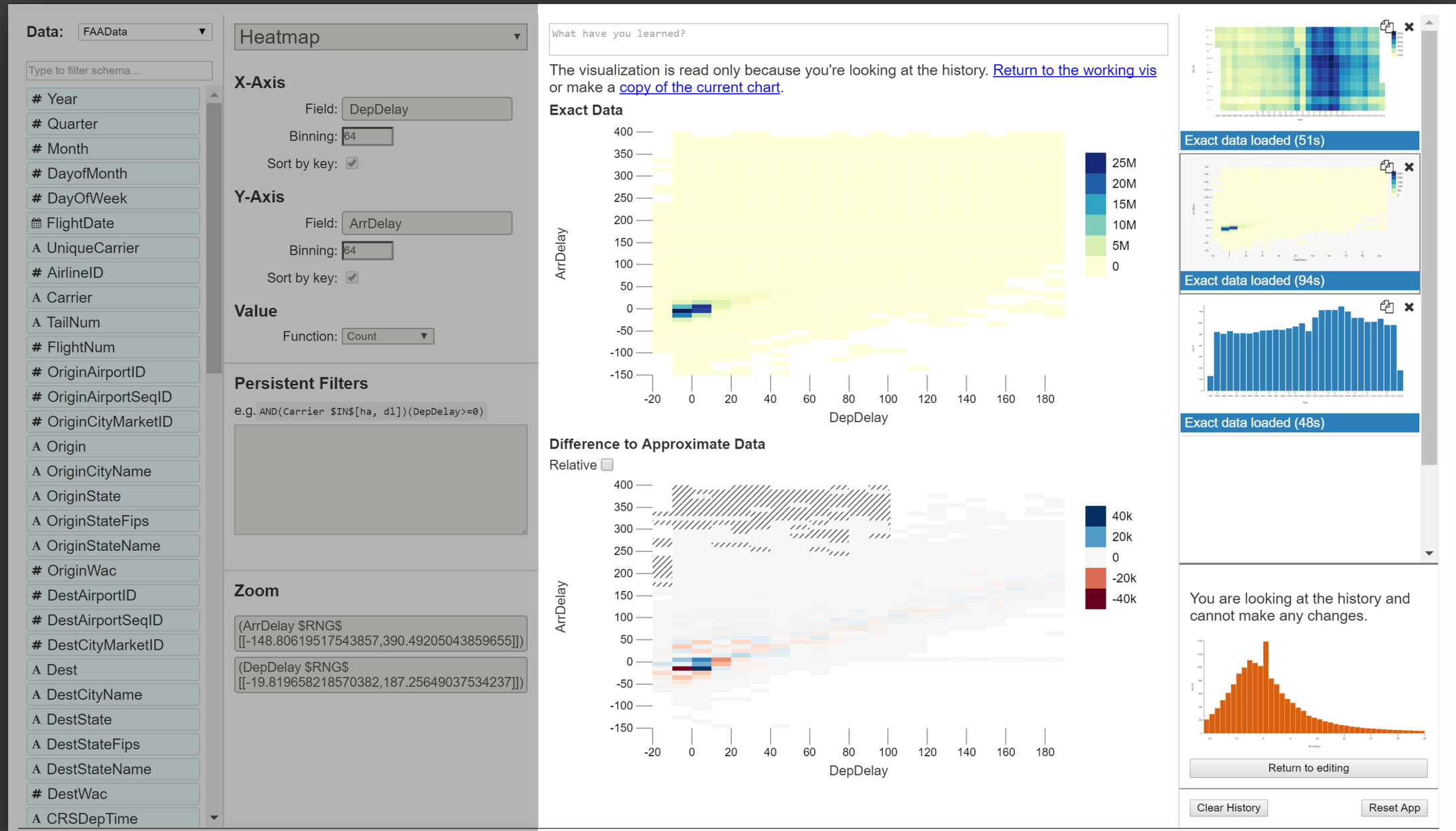
Pangloss Implements Optimistic Visualization



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.