Announcements…

- TCE website still open - please fill it out!

So You Have Too Much Data. What Now?

CS444

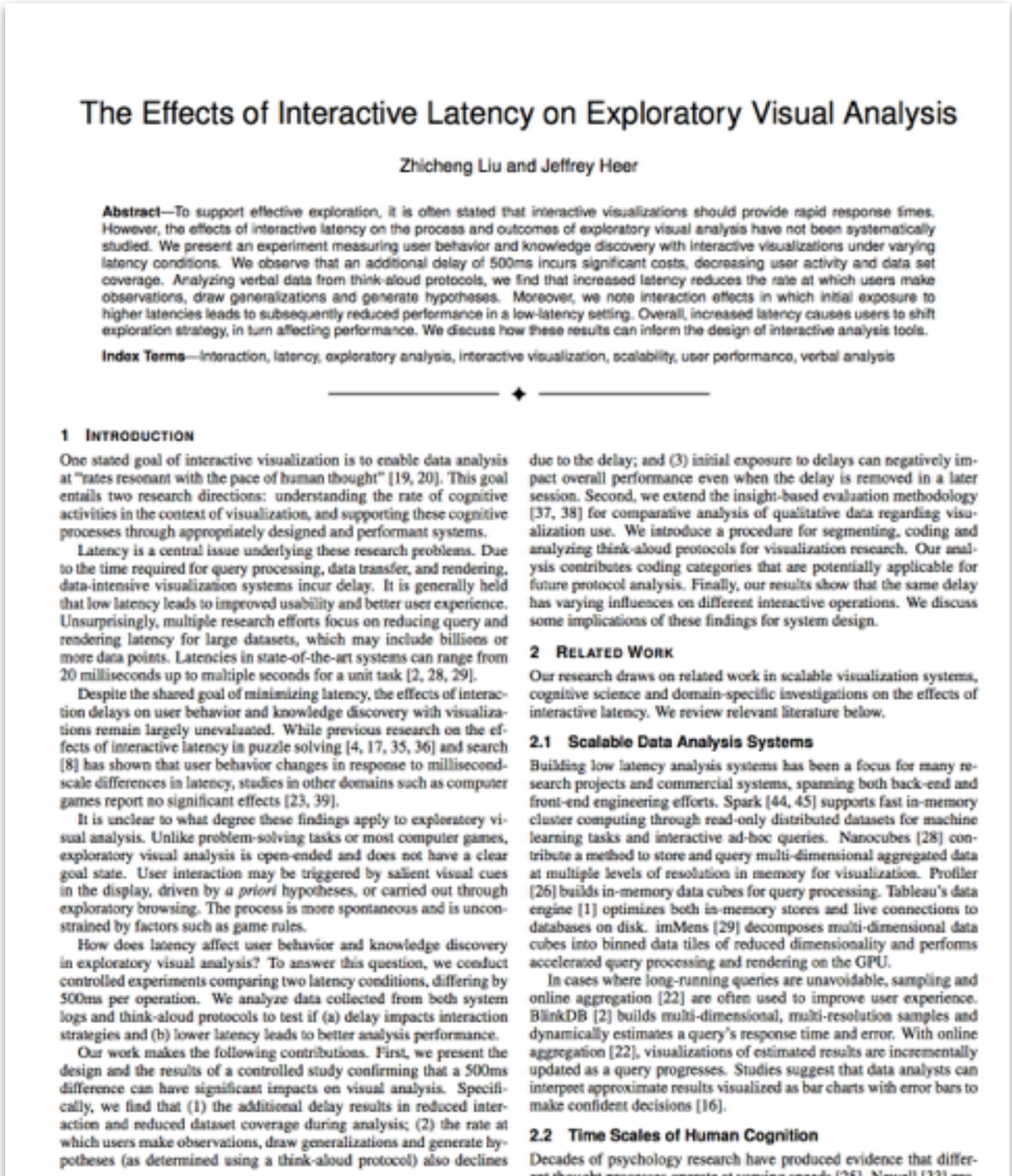
Previously…

- “Overview, zoom-and-filter, details-on-demand”
  - These are requirements for the **experience** of an interactive visualization
- But how do we **implement** them?
- Today’s lecture is a sampling of ongoing research work in the area

Do we care about this?

- A half-second latency between query and response changes user strategies in interactive data analysis

- Order effect: if first interaction is high-latency, user performance is degraded throughout entire session



The Effects of Interactive Latency on Exploratory Visual Analysis

Zhicheng Liu and Jeffrey Heer

Abstract—To support effective exploration, it is often stated that interactive visualizations should provide rapid response times. However, the effects of interactive latency on the process and outcomes of exploratory visual analysis have not been systematically studied. We present an experiment measuring user behavior and knowledge discovery with interactive visualizations under varying latency conditions. We observe that an additional delay of 500ms incurs significant costs, decreasing user activity and data set coverage. Analyzing verbal data from think-aloud protocols, we find that increased latency reduces the rate at which users make observations, draw generalizations and generate hypotheses. Moreover, we note interaction effects in which initial exposure to higher latencies leads to subsequently reduced performance in a low-latency setting. Overall, increased latency causes users to shift exploration strategy, in turn affecting performance. We discuss how these results can inform the design of interactive analysis tools.

Index Terms—Interaction, latency, exploratory analysis, interactive visualization, scalability, user performance, verbal analysis

1 Introduction

One stated goal of interactive visualization is to enable data analysis at "rates resonant with the pace of human thought" [19, 20]. This goal entails two research directions: understanding the rate of cognitive activities in the context of visualization, and supporting these cognitive processes through appropriately designed and performant systems.

Latency is a central issue underlying these research problems. Due to the time required for query processing, data transfer, and rendering, data-intensive visualization systems incur delay. It is generally held that low latency leads to improved usability and better user experience. Unsurprisingly, multiple research efforts focus on reducing query and rendering latency for large datasets, which may include billions or more data points. Latencies in state-of-the-art systems can range from 20 milliseconds up to multiple seconds for a unit task [2, 28, 29].

Despite the shared goal of minimizing latency, the effects of interaction delays on user behavior and knowledge discovery with visualizations remain largely unevaluated. While previous research on the effects of interactive latency in puzzle solving [4, 17, 35, 36] and search [8] has shown that user behavior changes in response to millisecond-scale differences in latency, studies in other domains such as computer games report no significant effects [23, 39].

It is unclear to what degree these findings apply to exploratory visual analysis. Unlike problem-solving tasks or most computer games, exploratory visual analysis is open-ended and does not have a clear goal state. User interaction may be triggered by salient visual cues in the display, driven by *a priori* hypotheses, or carried out through exploratory browsing. The process is more spontaneous and is unconstrained by factors such as game rules.

How does latency affect user behavior and knowledge discovery in exploratory visual analysis? To answer this question, we conduct controlled experiments comparing two latency conditions, differing by 500ms per operation. We analyze data collected from both system logs and think-aloud protocols to test if (a) delay impacts interaction strategies and (b) lower latency leads to better analysis performance.

Our work makes the following contributions. First, we present the design and the results of a controlled study confirming that a 500ms difference can have significant impacts on visual analysis. Specifically, we find that (1) the additional delay results in reduced interaction and reduced dataset coverage during analysis; (2) the rate at which users make observations, draw generalizations and generate hypotheses (as determined using a think-aloud protocol) also declines due to the delay; and (3) initial exposure to delays can negatively impact overall performance even when the delay is removed in a later session. Second, we extend the insight-based evaluation methodology [37, 38] for comparative analysis of qualitative data regarding visualization use. We introduce a procedure for segmenting, coding and analyzing think-aloud protocols for visualization research. Our analysis contributes coding categories that are potentially applicable for future protocol analysis. Finally, our results show that the same delay has varying influences on different interactive operations. We discuss some implications of these findings for system design.

2 Related Work

Our research draws on related work in scalable visualization systems, cognitive science and domain-specific investigations on the effects of interactive latency. We review relevant literature below.

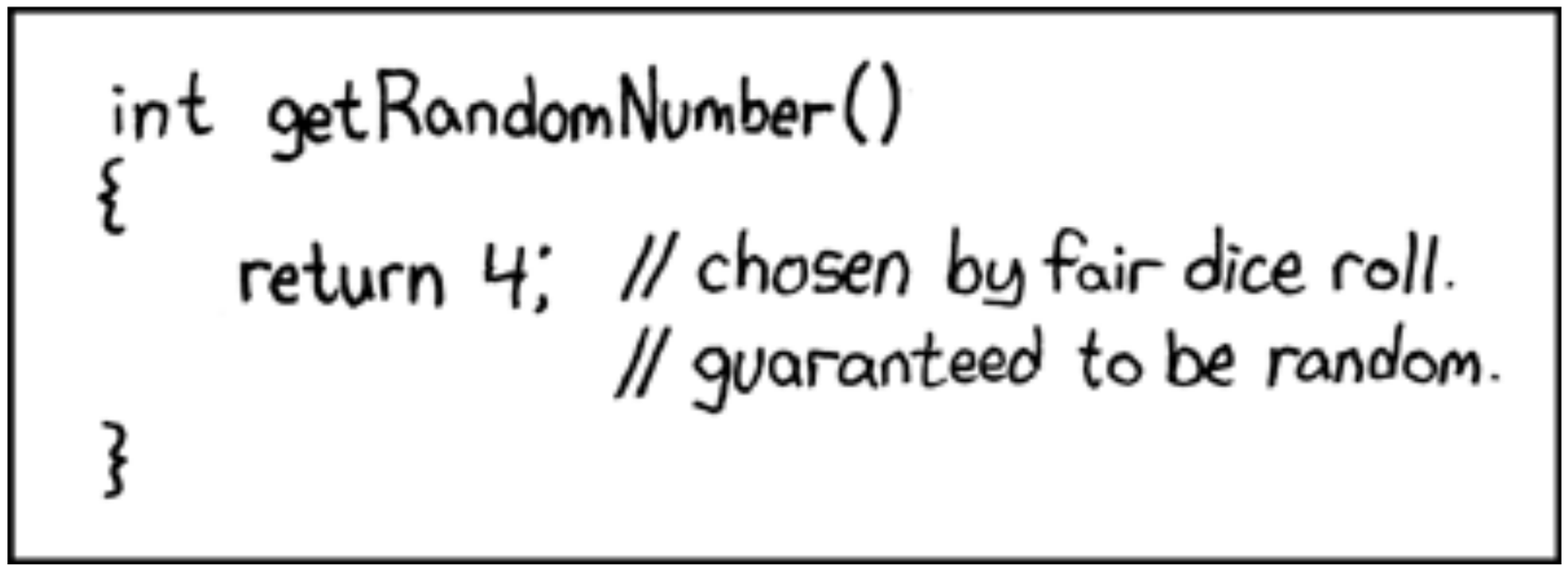
2.1 Scalable Data Analysis Systems

Building low latency analysis systems has been a focus for many research projects and commercial systems, spanning both back-end and front-end engineering efforts. Spark [44, 45] supports fast in-memory cluster computing through read-only distributed datasets for machine learning tasks and interactive ad-hoc queries. Nanocubes [28] contribute a method to store and query multi-dimensional aggregated data at multiple levels of resolution in memory for visualization. Profiler [26] builds in-memory data cubes for query processing. Tableau's data engine [1] optimizes both in-memory stores and live connections to databases on disk. imMens [29] decomposes multi-dimensional data cubes into binned data tiles of reduced dimensionality and performs accelerated query processing and rendering on the GPU.

In cases where long-running queries are unavoidable, sampling and online aggregation [22] are often used to improve user experience. BlinkDB [2] builds multi-dimensional, multi-resolution samples and dynamically estimates a query's response time and error. With online aggregation [22], visualizations of estimated results are incrementally updated as a query progresses. Studies suggest that data analysts can interpret approximate results visualized as bar charts with error bars to make confident decisions [16].

2.2 Time Scales of Human Cognition

Decades of psychology research have produced evidence that differ-



https://xkcd.com/221/

Sampling

If it’s good enough for stats, it should be good enough for vis (right?)

Why sampling?

- In statistics, we do it for two reasons:
  - For many questions, **we don’t need the entire population to get good answers**
  - And it’s too costly anyway
- In vis, we want to reduce running time, latency, or **time to next question**

Incremental Analytics

Session: Visualization + Visual Analysis CHI 2012, May 5–10, 2012, Austin, Texas, USA

Trust Me, I'm Partially Right: Incremental Visualization Lets Analysts Explore Large Datasets Faster

Danyel Fisher*, Igor Popov†, Steven M. Drucker*, mc schraefel†

*Microsoft Research
1 Microsoft Way,
Redmond, WA USA
{danyelf, sdrucker}@microsoft.com

†Electronics and Computer Science
University of Southampton
Southampton, Hampshire, UK SO17 1BJ
{mc+w, ip2go9}@ecs.soton.ac.uk

ABSTRACT

Queries over large scale (petabyte) data bases often mean waiting overnight for a result to come back. Scale costs time. Such time also means that potential avenues of exploration are ignored because the costs are perceived to be too high to run or even propose them. With *sampleAction* we have explored whether interaction techniques to present query results running over only incremental samples can be presented as sufficiently trustworthy for analysts both to make closer to real time decisions about their queries and to be more exploratory in their questions of the data. Our work with three teams of analysts suggests that we can indeed accelerate and open up the query process with such incremental visualizations.

query costs in a variety of ways. Strategies to accelerate large scale data processing are represented in systems like Dremel [9] and C-Store [18] that churn through large collections of data by pre-structuring the data and moving the computation closer to the data.

So while computational and storage approaches make large scale queries possible, they still often restrict either the number and types of queries that might be run, or avenues that might be explored because the queries must be designed with such care to be worth the wait and the cost of queuing for the resource.

One possible technique, proposed by Hellerstein and others [4], is to query databases incrementally, looking at ever-

Incremental Analytics

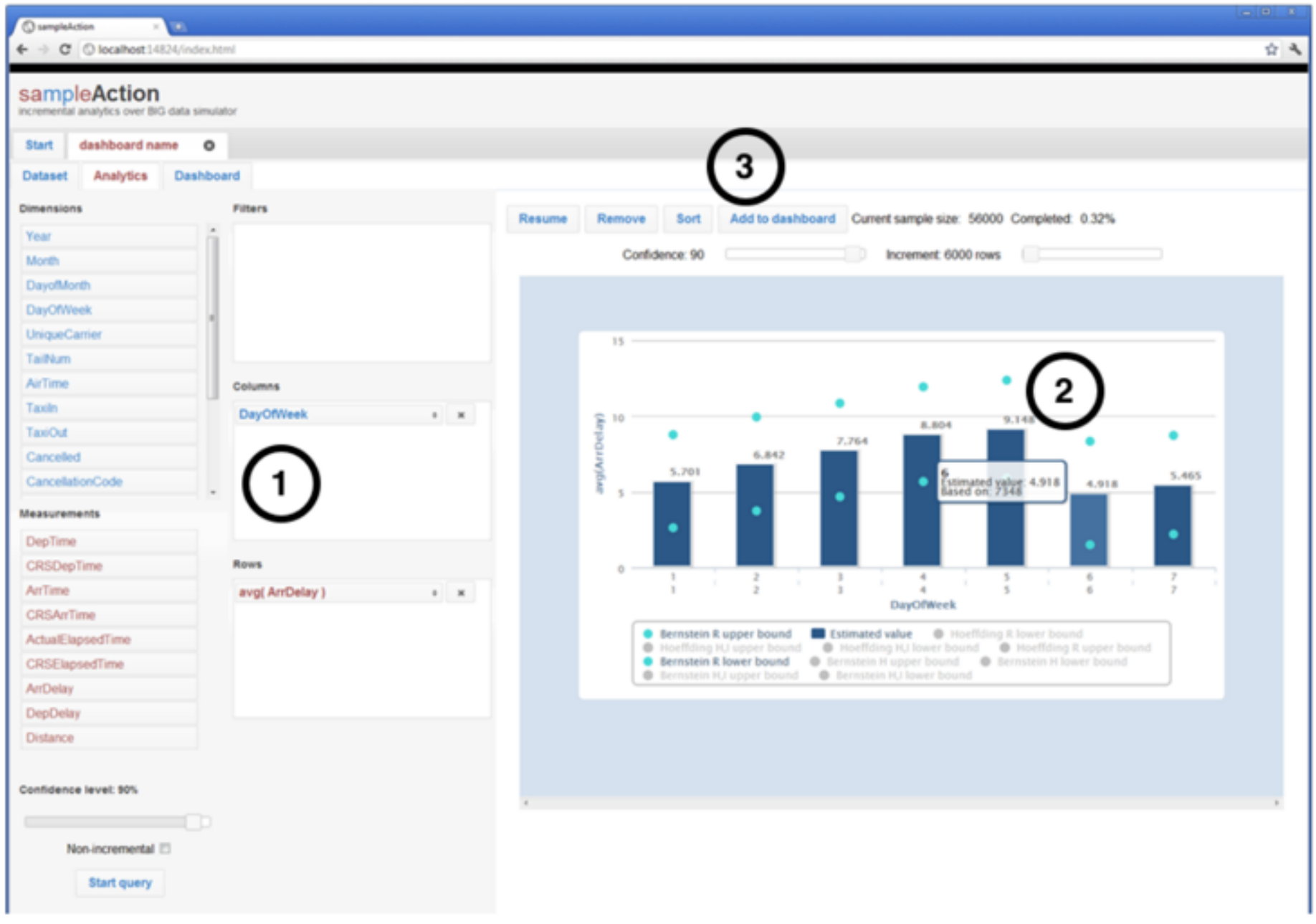
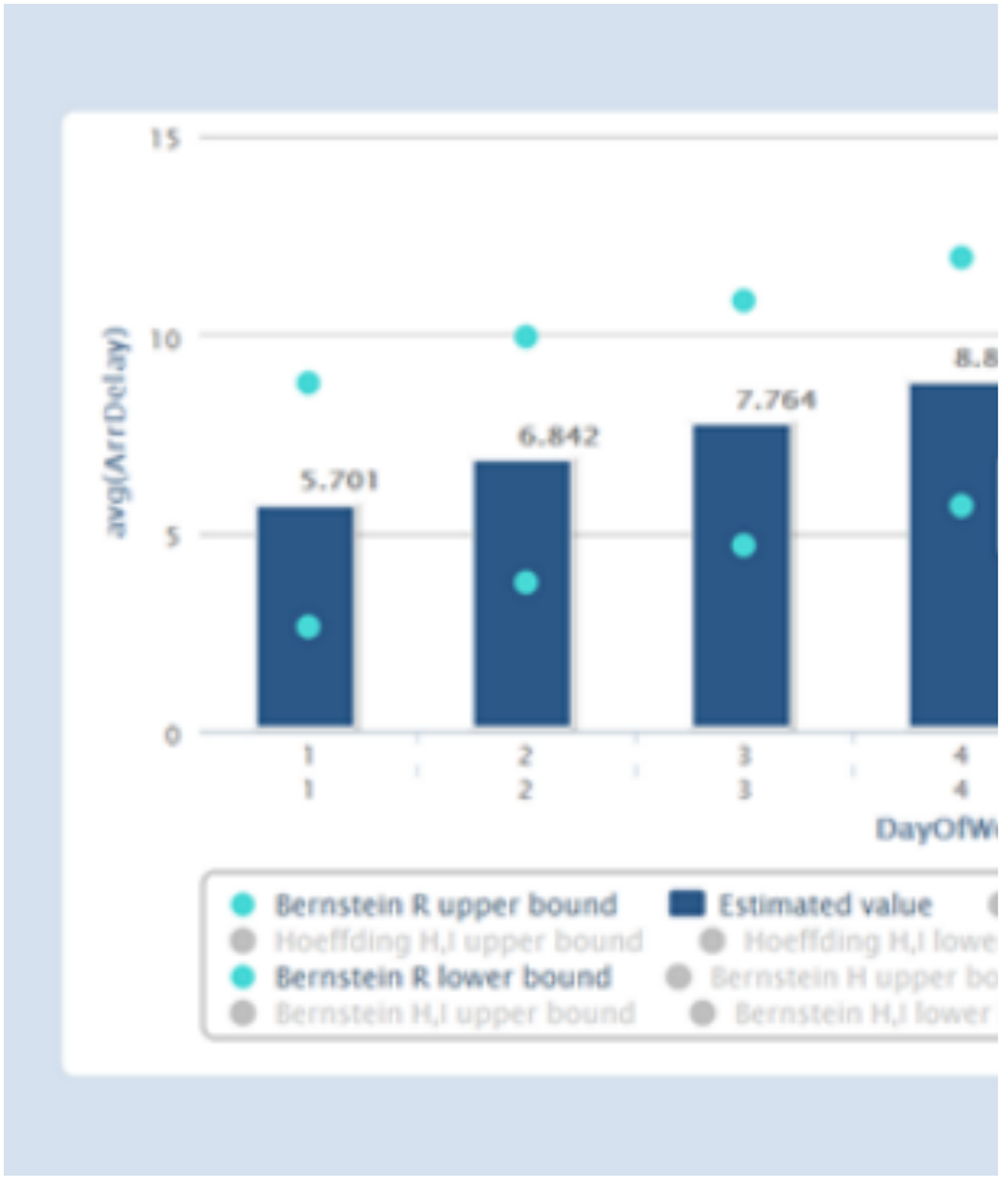


Figure 1. The Analytics panel in sampleAction showing an incremental visualization in progress. The analyst is looking at flight delays by day of week. (1) Selecting columns to be shown in (2) the visualization. Dark blue bars show current estimates; pale blue dots show the expected range of values. This prototype interface includes multiple selectable bounding algorithms. (3) A progress indicator showing that 0.32% of the database has been seen so far.

Incremental Analytics

- Show **uncertainty range**
  - These come from “concentration bounds”
- As you get more data, uncertainty drops.

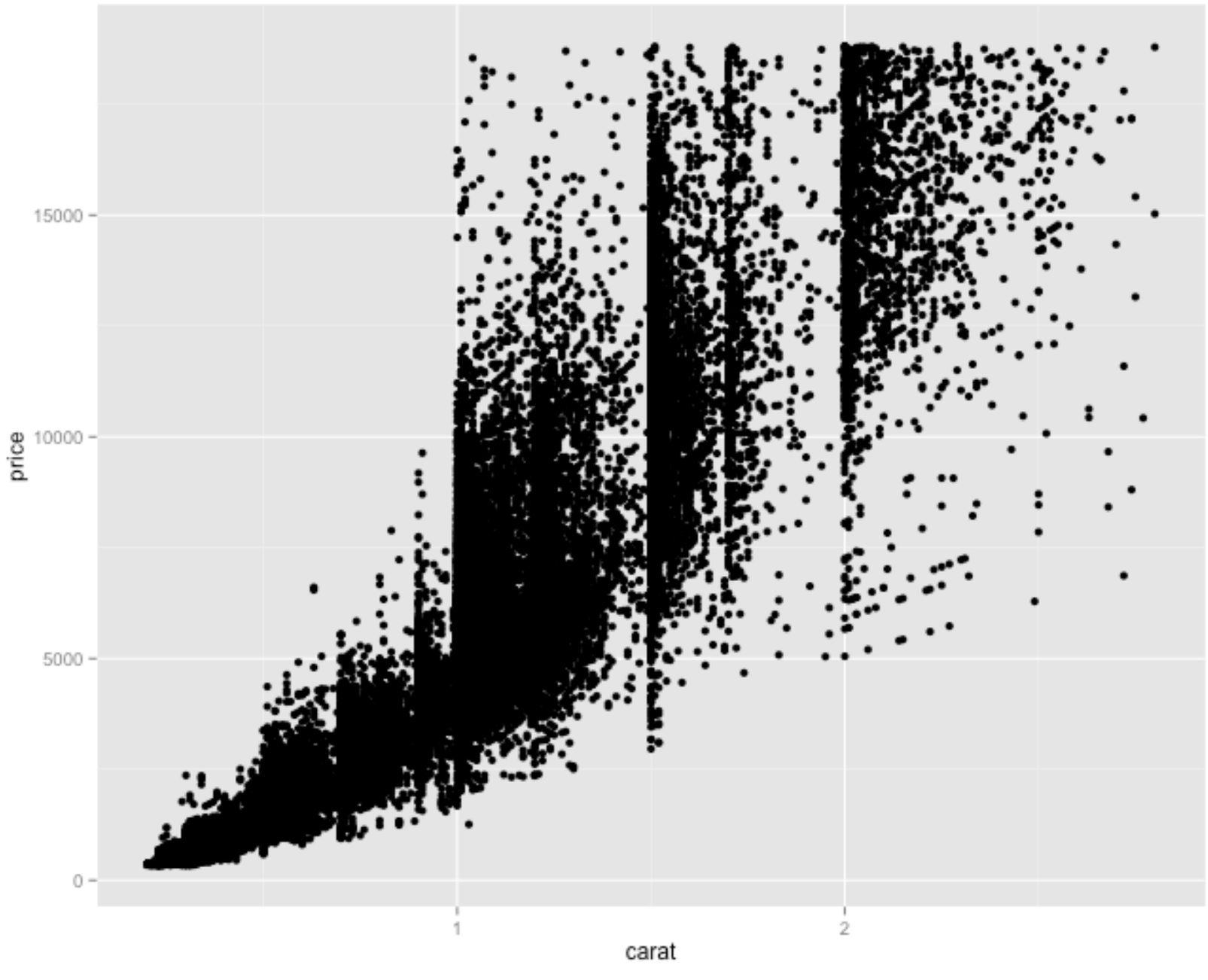


How do we **build** this?

- Instead of asking server for entire dataset, ask for “1000 values at random”
  - or “next 1000 values”
- Compute based only on those values

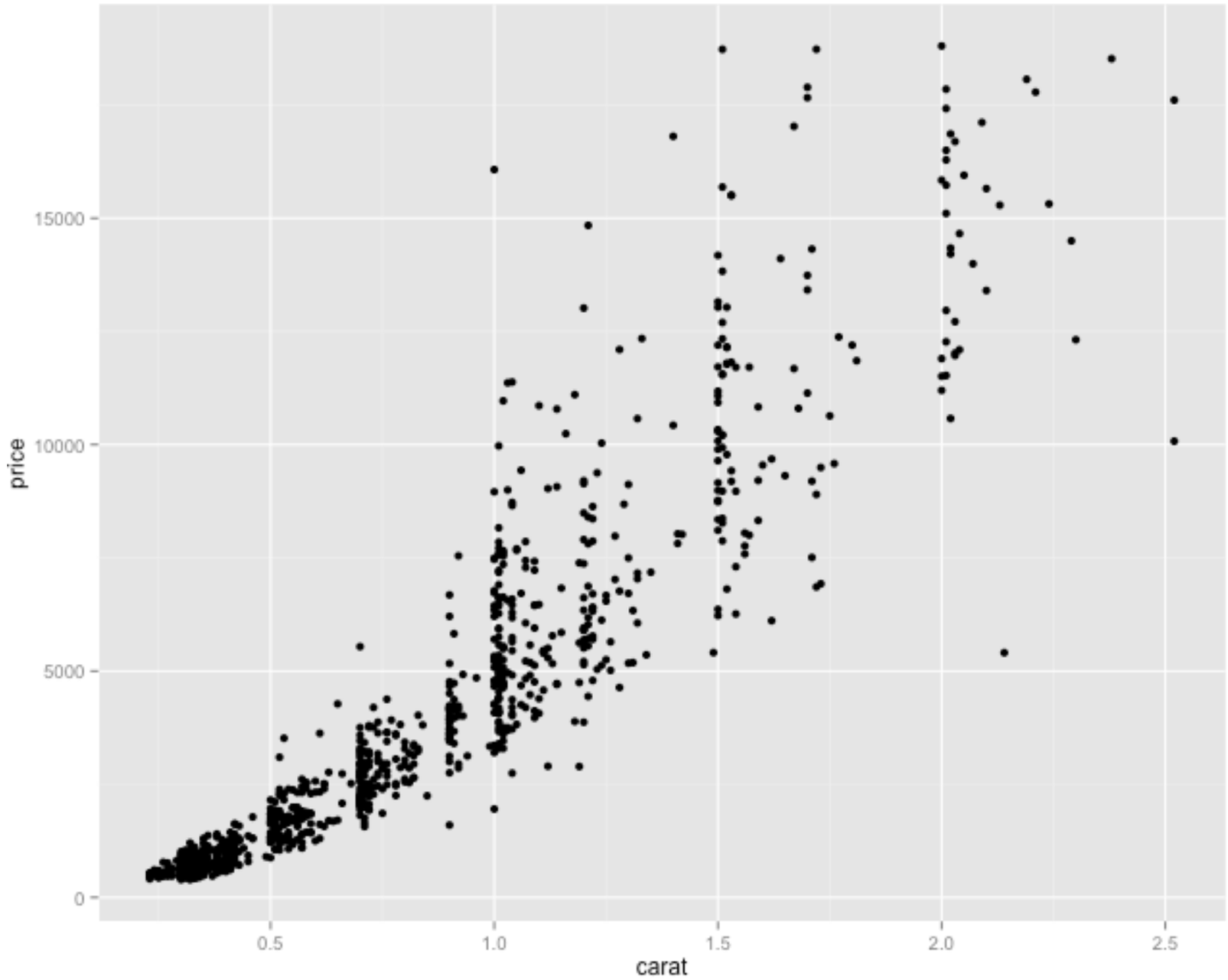
Sampling demo

```
> ggplot(filter(diamonds, carat < 3), aes(x=carat, y=price)) + geom_point()
```



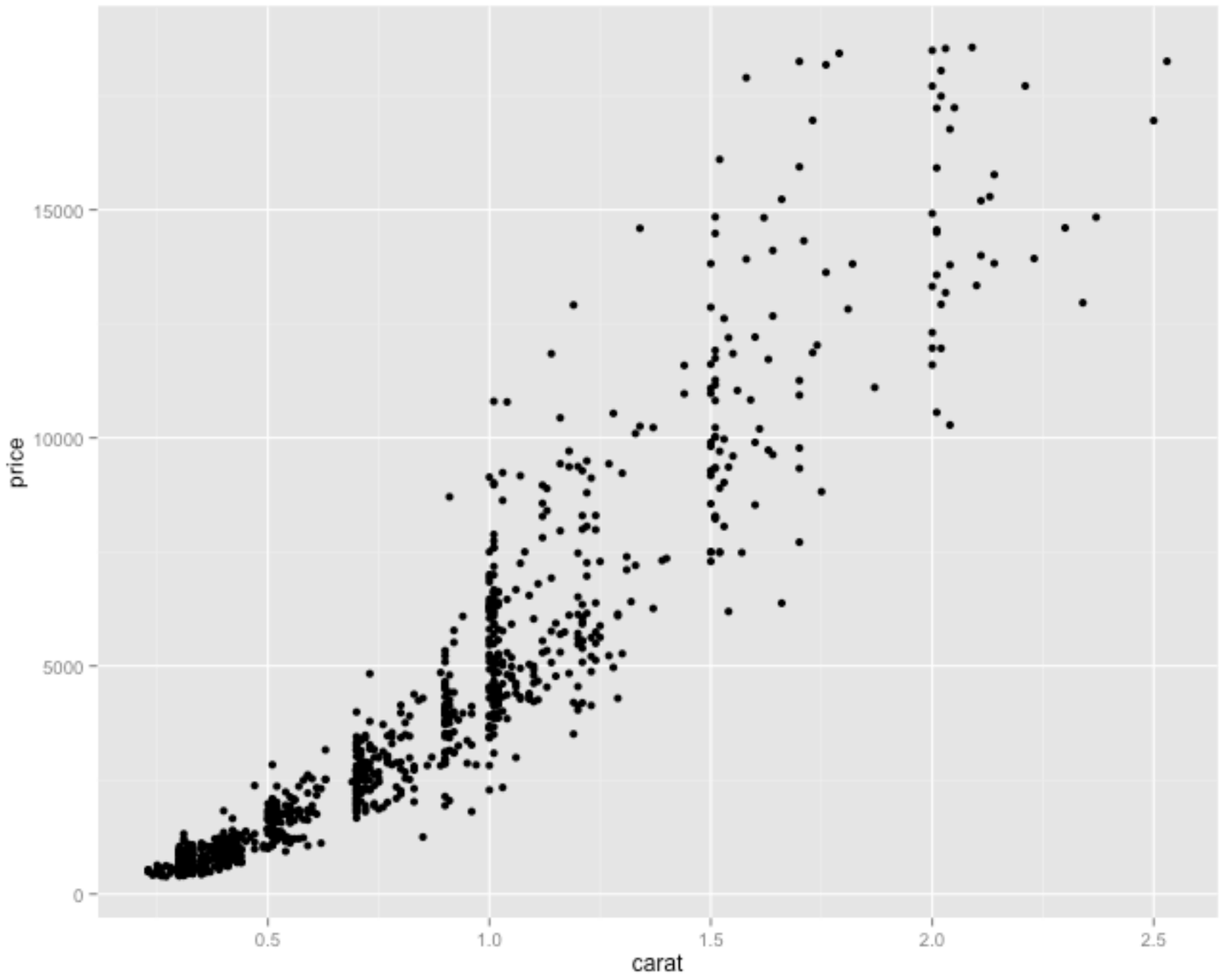
Sampling demo

```
> ggplot(filter(sample_n(diamonds, 1000), carat < 3), aes(x=carat, y=price)) + geom_point()
```



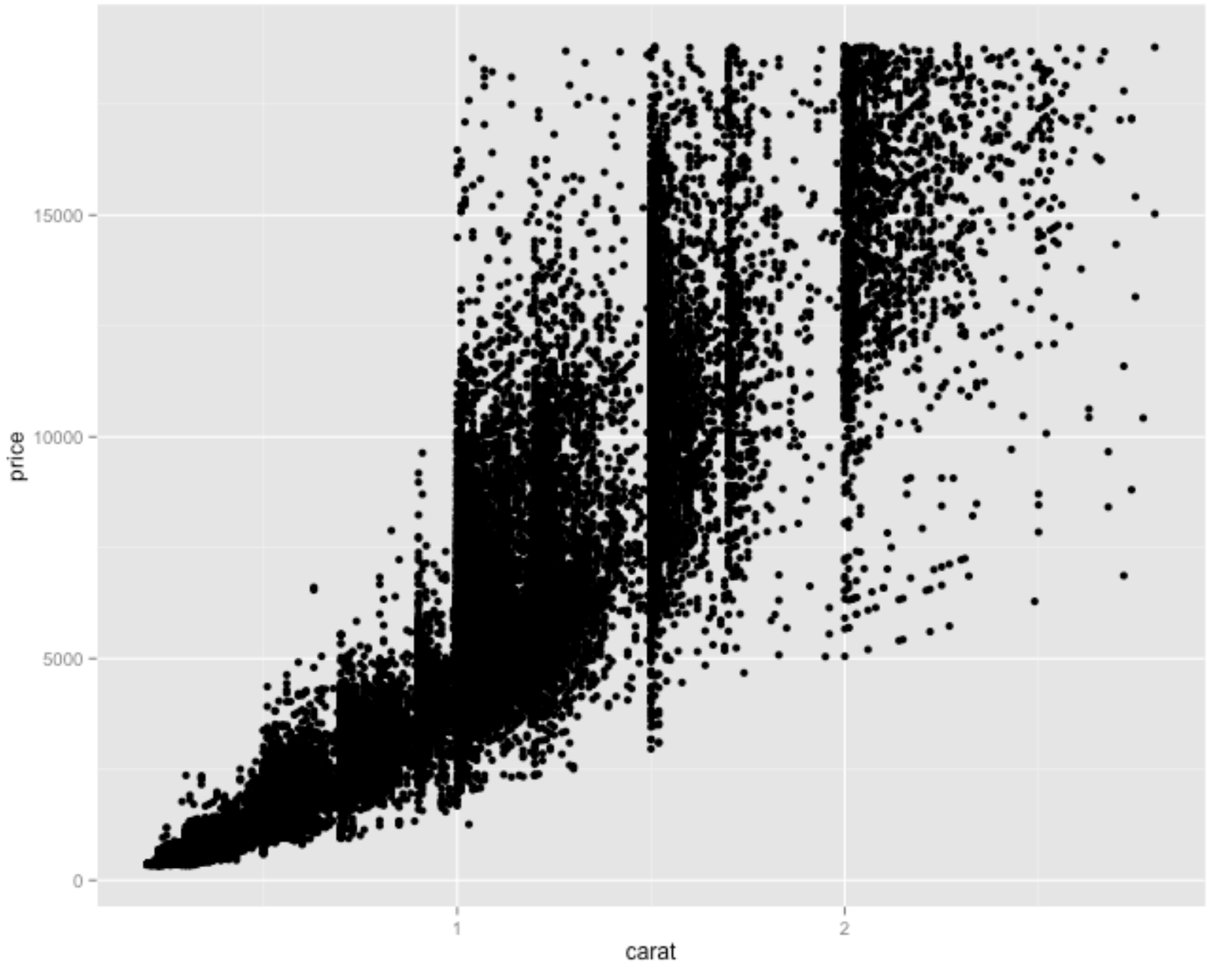
Sampling demo

```
> ggplot(filter(sample_n(diamonds, 1000), carat < 3), aes(x=carat, y=price)) + geom_point()
```



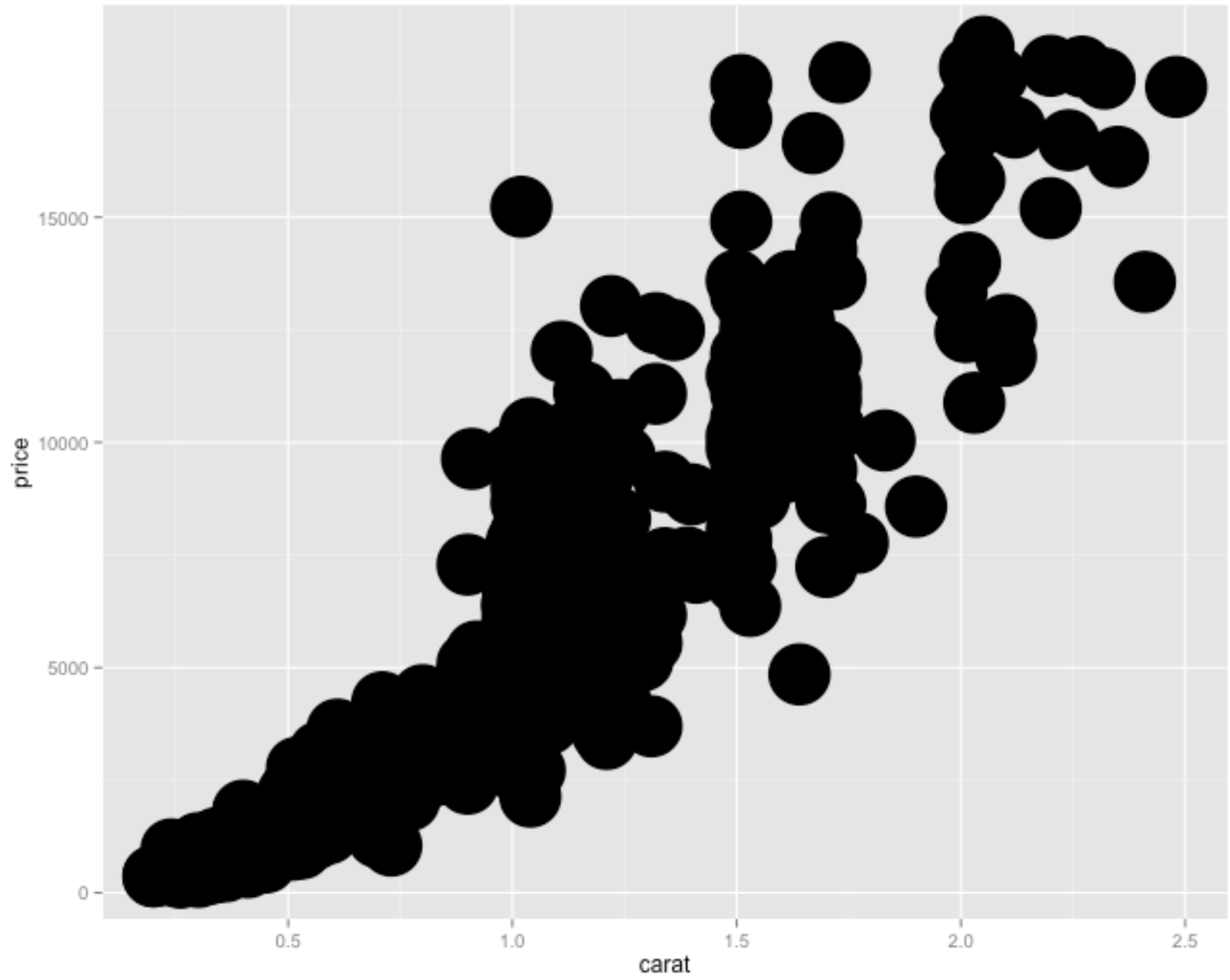
Sampling demo

```
> ggplot(filter(diamonds, carat < 3), aes(x=carat, y=price)) + geom_point()
```



Sampling demo

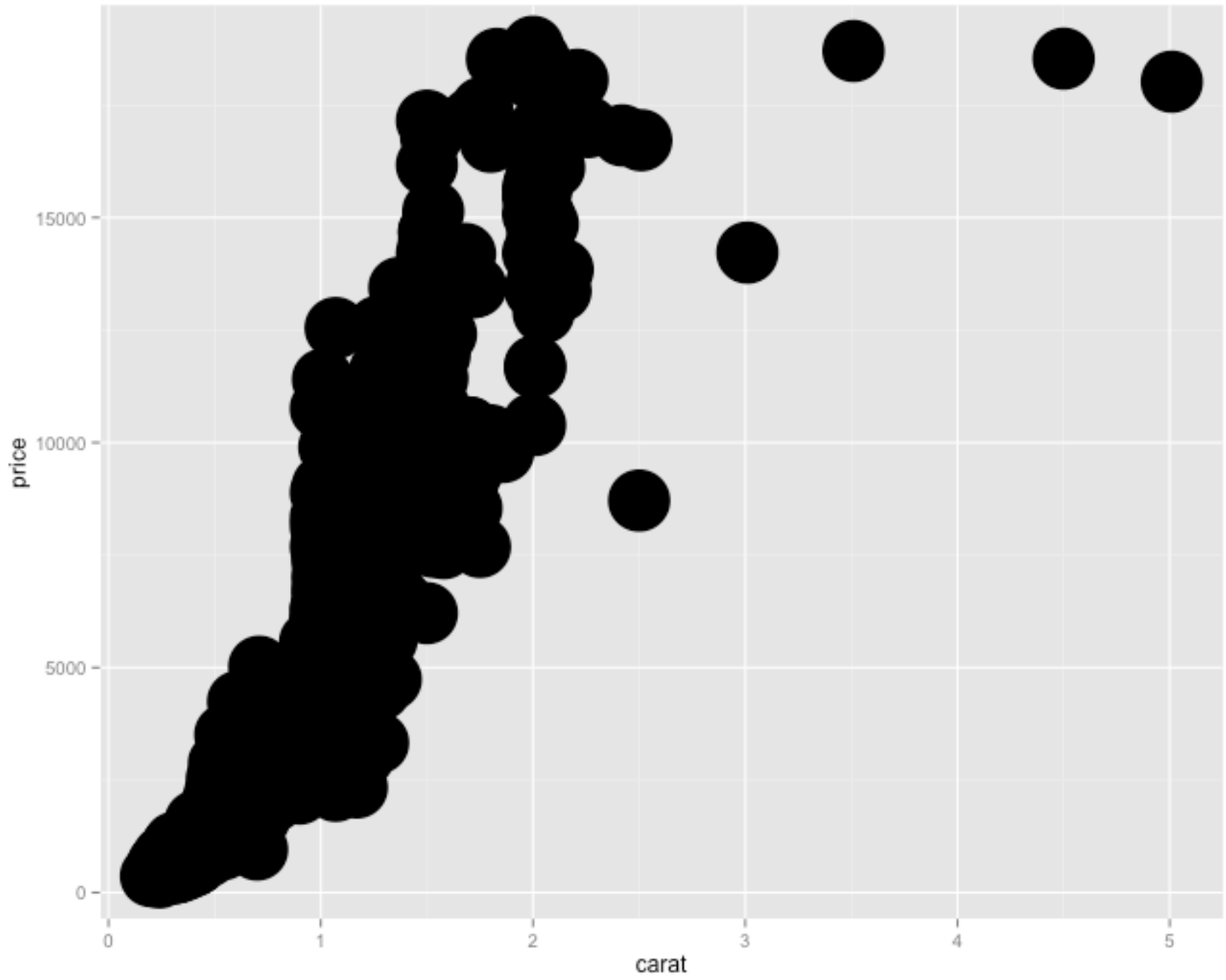
```
> ggplot(filter(sample_n(diamonds, 1000), carat < 3), aes(x=carat, y=price)) +
  geom_point(size=2*sqrt(58700 / 1000))
```



But what about outliers?

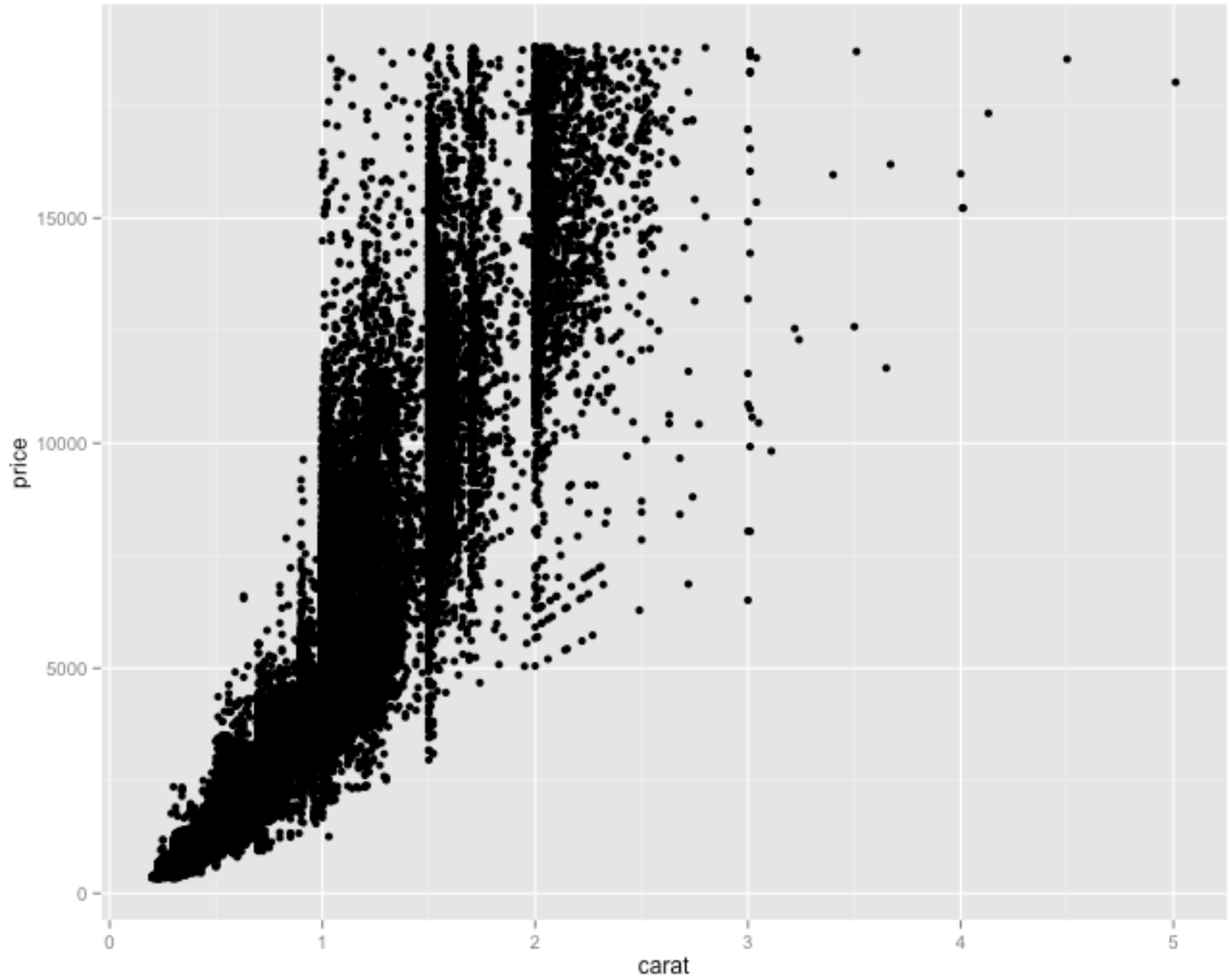
(After about 20 tries…)

```
> ggplot(sample_n(diamonds, 1000), aes(x=carat, y=price)) + geom_point(size=2*sqrt(58700/1000))
```



Without filtering outliers..

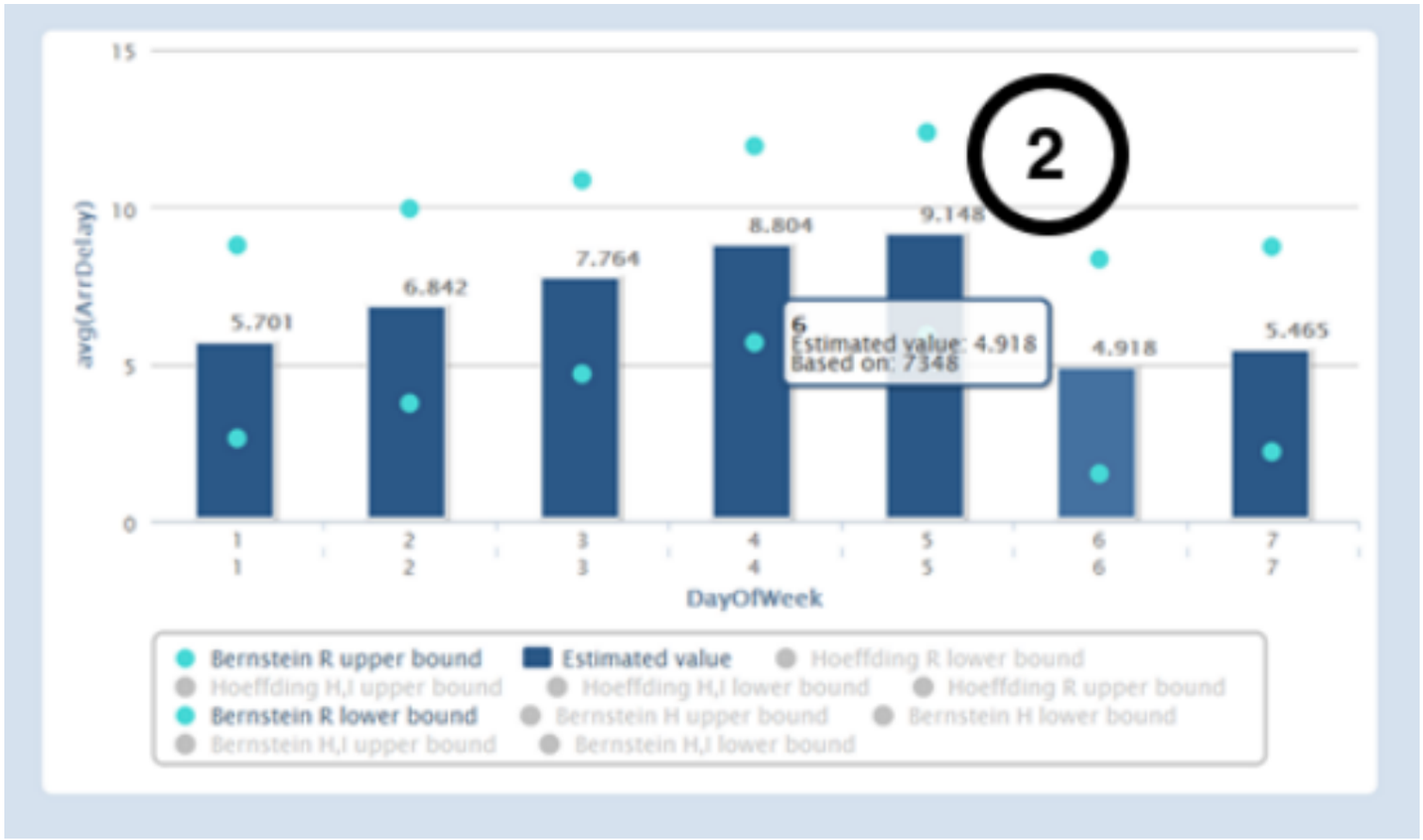
```
> ggplot(diamonds, aes(x=carat, y=price)) + geom_point()
```



Outliers are not the only problem

- **Simple random sampling only works when subpopulation is “easy to access”**
  - This is not only about vis! (political polls…)

Outliers are not the only problem



- **So… why does it work for sampleAction?**

Outliers are not the only problem

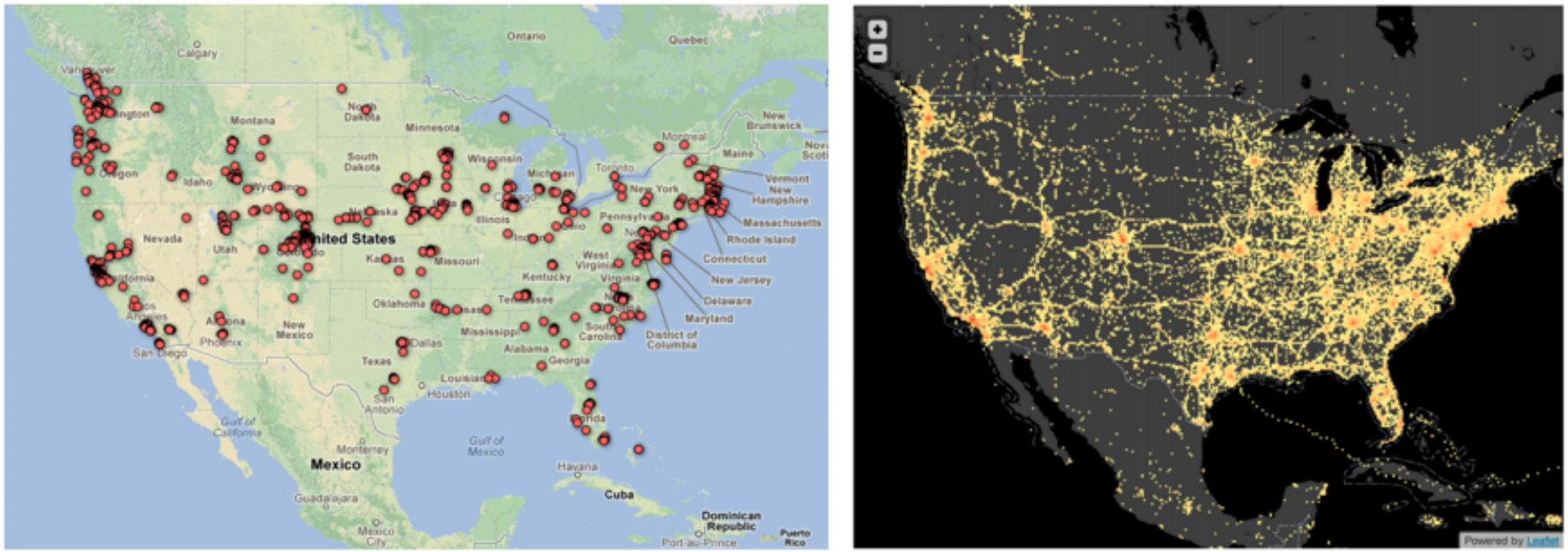
Difficulties with Error Bar Convergence

We did not anticipate the tremendous variance in confidence interval sizes. While Bob never saw a confidence interval much larger than his largest data point, Allan often could not see his data without hiding the confidence intervals. Past literature on visualizing uncertainty [11] has emphasized visualizations that fit the entire uncertainty range on screen; these were not sufficient for some of these preliminary bounds. It would be worthwhile investigating visualizations that can show the size of the interval even past screen borders.

- So… why does it work for sampleAction?
  - **… it kind of doesn't**

Outliers are not the only problem

Zhicheng Liu, Biye Jiang & Jeffrey Heer / imMens: Real-time Visual Querying of Big Data



(a) (b)

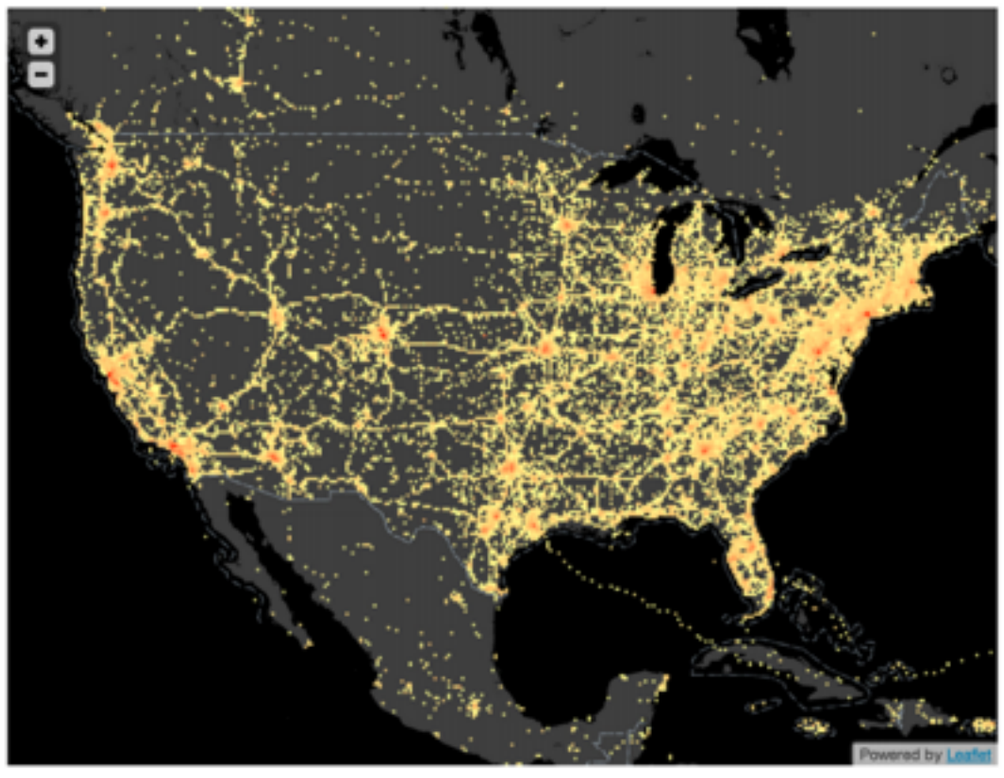
Figure 1: *A symbol map (a) and heatmap (b) visualizing a dataset of Brightkite user checkins. The symbol map visualizes a sample of the data, and the heatmap shows the density of checkins by aggregation. Compared to the heatmap, sampling misses important structures such as inter-state highway travel and Hurricane Ike, while dense regions still suffer from over-plotting.*

What’s going on here?

- **Simple random sampling only works when subpopulation is “easy to access”**

Zhicheng Liu, Biye Jiang & Jeffrey Heer / imMens: Real-time Visual Querying of Big Data

(a)



(b)

How do we solve it?

- Very much an active research problem

amplab
UC Berkeley

BlinkDB

MIT CSAIL

Queries with Bounded Errors and Bounded Response Times on Very Large Data

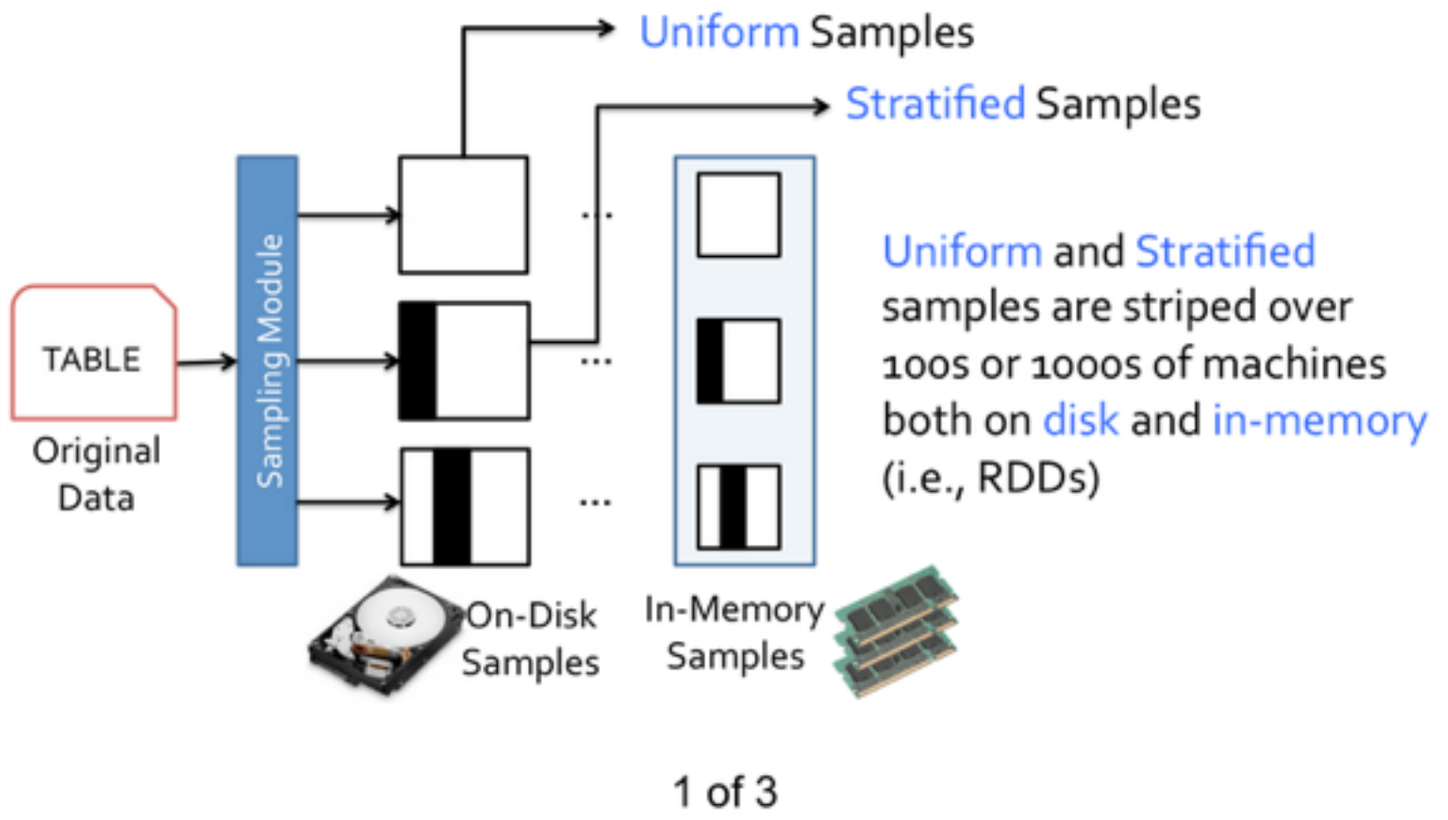
BlinkDB Developer Alpha 0.2.0 Released! ×

BlinkDB is a massively parallel, approximate query engine for running interactive SQL queries on large volumes of data. It allows users to trade-off query accuracy for response time, enabling interactive queries over massive data by running queries on data samples and presenting results annotated with meaningful error bars. To achieve this, BlinkDB uses two key ideas: (1) An adaptive optimization framework that builds and maintains a set of multi-dimensional samples from original data over time, and (2) A dynamic sample selection strategy that selects an appropriately sized sample based on a query's accuracy and/or response time requirements. We have evaluated BlinkDB on the well-known TPC-H benchmarks, a real-world analytic workload derived from Conviva Inc. and are in the process of deploying it at Facebook Inc.

BlinkDB has been demonstrated live at VLDB 2012 on a 100 node Amazon EC2 cluster answering a range of queries on 17 TBs of data in less than 2 seconds (over 200x faster than Hive), within an error of 2-10%.

How do we solve it?

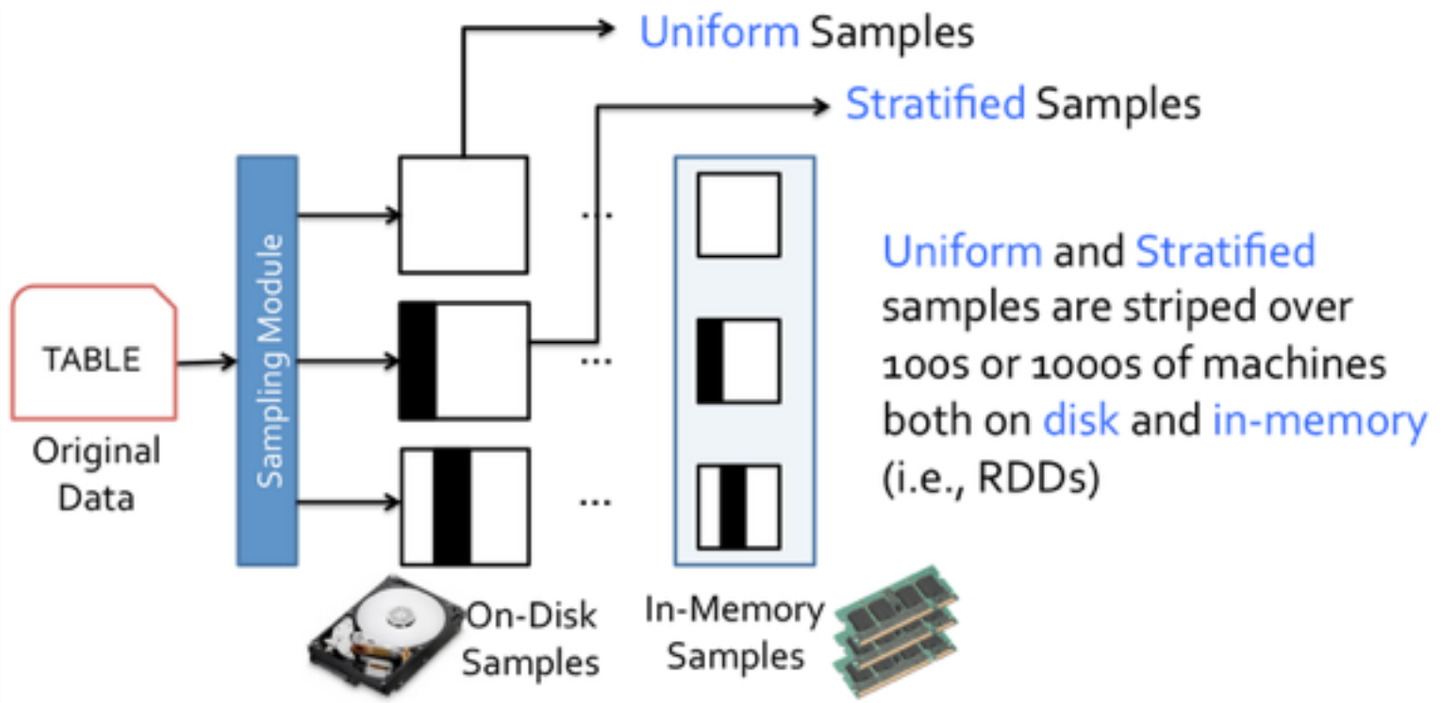
BlinkDB features an Offline Sampling Module that creates Uniform and Stratified samples from underlying data within a given storage budget. The sets of column(s) to stratify on are decided by solving a MILP optimization problem that takes into account the frequently occurring column(s) in the GROUP BY and WHERE clauses of the queries. This module is also responsible for periodically updating, deleting and refreshing the samples to minimize the statistical bias.



How do we solve it?

- Big idea: **stratified samples**

BlinkDB features an Offline Sampling Module that creates Uniform and Stratified samples from underlying data within a given storage budget. The sets of column(s) to stratify on are decided by solving a MILP optimization problem that takes into account the frequently occurring column(s) in the GROUP BY and WHERE clauses of the queries. This module is also responsible for periodically updating, deleting and refreshing the samples to minimize the statistical bias.



How do we solve it?

- Big idea: only **preserve visually important properties**
- http://arxiv.org/pdf/1412.3040.pdf

Rapid Sampling for Visualizations with Ordering Guarantees

Albert Kim
MIT
alkim@csail.mit.edu

Eric Blais
MIT and University of Waterloo
eblais@uwaterloo.ca

Aditya Parameswaran
MIT and Illinois (UIUC)
adityagp@illinois.edu

Piotr Indyk
MIT
indyk@mit.edu

Sam Madden
MIT
madden@csail.mit.edu

Ronitt Rubinfeld
MIT and Tel Aviv University
ronitt@csail.mit.edu

ABSTRACT

Visualizations are frequently used as a means to understand trends and gather insights from datasets, but often take a long time to generate. In this paper, we focus on the problem of *rapidly generating approximate visualizations while preserving crucial visual properties of interest to analysts*. Our primary focus will be on sampling algorithms that preserve the visual property of *ordering*; our techniques will also apply to some other visual properties. For instance, our algorithms can be used to generate an approximate visualization of a bar chart very rapidly, where the comparisons between

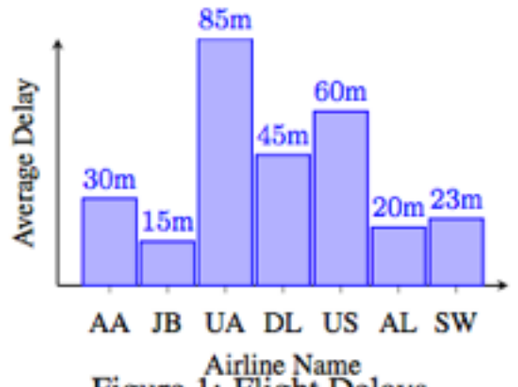


Figure 1: Flight Delays

that preserve visual properties, i.e., those that ensure that the visu-

How do we solve it?

- Big idea: only **preserve visually important properties**
  - **Sample the subset that is most likely to change the output where it matters**

Rapid Sampling for Visualizations with Ordering Guarantees

Albert Kim
MIT
alkim@csail.mit.edu

Eric Blais
MIT and University of Waterloo
eblais@uwaterloo.ca

Aditya Parameswaran
MIT and Illinois (UIUC)
adityagp@illinois.edu

Piotr Indyk
MIT
indyk@mit.edu

Sam Madden
MIT
madden@csail.mit.edu

Ronitt Rubinfeld
MIT and Tel Aviv University
ronitt@csail.mit.edu

ABSTRACT

Visualizations are frequently used as a means to understand trends and gather insights from datasets, but often take a long time to generate. In this paper, we focus on the problem of *rapidly generating approximate visualizations while preserving crucial visual proper-*

LEMMA 2 (HOEFFDING–SERFLING INEQUALITY [46]). *Let* $\mathcal{Y} = y_1, \ldots, y_N$ *be a set of* N *values in* $[0,1]$ *with average value* $\frac{1}{N}\sum_{i=1}^{N} y_i = \mu$. *Let* $Y_1, \ldots, Y_N$ *be a sequence of random variables drawn from* $\mathcal{Y}$ *without replacement. For every* $1 \le k < N$ *and* $\varepsilon > 0$,

$$\Pr\left[\max_{k \le m \le N-1} \left|\frac{\sum_{i=1}^{m} Y_i}{m} - \mu\right| \ge \varepsilon\right] \le 2\exp\left(-\frac{2k\varepsilon^2}{1 - \frac{k-1}{N}}\right).$$

We use the above inequality to get tight bounds for the value of $\sum_{i=1}^{m} Y_i/m$ for all $1 \le m \le N$, with probability δ. We discuss next how to apply the theorem to complete Step 2 of our proof.

THEOREM 3.2. *Let* $\mathcal{Y} = y_1, \ldots, y_N$ *be a set of* N *values in* $[0,1]$ *with average value* $\frac{1}{N}\sum_{i=1}^{N} y_i = \mu$. *Let* $Y_1, \ldots, Y_N$ *be a sequence of random variables drawn from* $\mathcal{Y}$ *without replacement. Fix any* $\delta > 0$ *and* $\kappa > 1$. *For* $1 \le m \le N-1$, *define*

$$\varepsilon_m = \sqrt{\frac{(1 - \frac{m/\kappa - 1}{N})(2\log\log_\kappa(m) + \log(\pi^2/3\delta))}{2m/\kappa}}.$$

Then: $\Pr\left[\exists m, 1 \le m \le N : \left|\frac{\sum_{i=1}^{m} Y_i}{m} - \mu\right| > \varepsilon_m\right] \le \delta.$

PROOF. We have:

$$\begin{aligned}
&\Pr\left[\exists m, 1 \le m \le N : \left|\frac{\sum_{i=1}^{m} Y_i}{m} - \mu\right| > \varepsilon_m\right] \\
&\le \sum_{r \ge 1} \Pr\left[\exists m, \kappa^{r-1} \le m \le \kappa^r : \left|\frac{\sum_{i=1}^{m} Y_i}{m} - \mu\right| > \varepsilon_m\right] \\
&\le \sum_{r \ge 1} \Pr\left[\exists m, \kappa^{r-1} \le m \le \kappa^r : \left|\frac{\sum_{i=1}^{m} Y_i}{m} - \mu\right| > \varepsilon_{\kappa^r}\right] \\
&\le \sum_{r \ge 1} \Pr\left[\max_{\kappa^{r-1} \le m \le N-1} \left|\frac{\sum_{i=1}^{m} Y_i}{m} - \mu\right| > \varepsilon_{\kappa^r}\right].
\end{aligned}$$

LEMMA 3. *Fix* $i \in 1 \ldots k$. *Define* m_i^* *to be the minimal value of* $m \ge 1$ *for which* $\varepsilon_m < \eta_i/4$. *In the running of the algorithm, if for every* $j \in A_{m_i^*}$, *we have* $|\nu_{j,m_i^*} - \mu_j| \le \varepsilon_{m_i^*}$, *then* $m_i \le m_i^*$.

Intuitively, the lemma allows us to establish that $m_i < m_i^*$, the latter of which (as we show subsequently) is dependent on η_i.

PROOF. If $i \notin A_{m_i^*}$, then the conclusion of the lemma trivially holds, because $m_i < m_i^*$. Consider now the case where $i \in A_{m_i^*}$. We now prove that $m_i = m_i^*$. Note that $m_i = m_i^*$ if and only if the interval $[\nu_{i,m_i^*} - \varepsilon_{m_i^*}, \nu_{i,m_i^*} + \varepsilon_{m_i^*}]$ is disjoint from the union of intervals $\bigcup_{j \in A_{m_i^*} \setminus \{i\}} [\nu_{j,m_i^*} - \varepsilon_{m_i^*}, \nu_{j,m_i^*} + \varepsilon_{m_i^*}]$.

We focus first on all j where $\mu_j < \mu_i$. By the definition of η_i, every $j \in A_{m_i^*}$ for which $\mu_j < \mu_i$ satisfies the stronger inequality $\mu_j \le \mu_i - \eta_i$. By the conditions of the lemma (i.e., that confidence intervals always contain the true average), we have that $\mu_j \ge \nu_{j,m_i^*} - \varepsilon_{m_i^*}$ and that $\mu_i \le \nu_{i,m_i^*} + \varepsilon_{m_i^*}$. So we have:

$$\nu_{j,m_i^*} + \varepsilon_{m_i^*} \le \mu_j + 2\varepsilon_{m_i^*} < \mu_j + \frac{\eta_i}{2} \le \mu_i - \frac{\eta_i}{2} < \mu_i - 2\varepsilon_{m_i^*} \le \nu_{i,m_i^*} - \varepsilon_{m_i^*}$$

- The first and last inequalities follow the fact that the confidence interval for ν_j always contains μ_j, i.e., $\mu_j \ge \nu_{j,m_i^*} - \varepsilon_{m_i^*}$;
- the second and fourth follow from the fact that $\varepsilon_{m_i^*} < \eta_i/4$;
- and the third follows from the fact that $\mu_j \le \mu_i - \eta_i$.

Thus, the intervals $[\nu_{i,m_i^*} - \varepsilon_{m_i^*}, \nu_{i,m_i^*} + \varepsilon_{m_i^*}]$ and $[\nu_{j,m_i^*} - \varepsilon_{m_i^*}, \nu_{j,m_i^*} + \varepsilon_{m_i^*}]$ are disjoint. Similarly, for all $j \in A_{m_i^*}$ that satisfies $\mu_j > \mu_i$, we observe that the interval $[\nu_{i,m_i^*} - \varepsilon_{m_i^*}, \nu_{i,m_i^*} + \varepsilon_{m_i^*}]$ is also disjoint from $[\nu_{j,m_i^*} - \varepsilon_{m_i^*}, \nu_{j,m_i^*} + \varepsilon_{m_i^*}]$. □

We are now ready to complete the analysis of the algorithm.

LEMMA 2 (HOEFFDING–SERFLING INEQUALITY [46]). *Let* $\mathcal{Y} = y_1, \ldots, y_N$ *be a set of* N *values in* $[0,1]$ *with average value* $\frac{1}{N}\sum_{i=1}^{N} y_i = \mu$. *Let* $Y_1, \ldots, Y_N$ *be a sequence of random variables drawn from* $\mathcal{Y}$ *without replacement. For every* $1 \le k < N$ *and* $\varepsilon > 0$,

$$\Pr\left[\max_{k \le m \le N-1} \left|\frac{\sum_{i=1}^{m} Y_i}{m} - \mu\right| \ge \varepsilon\right] \le 2\exp\left(-\frac{2k\varepsilon^2}{1 - \frac{k-1}{N}}\right).$$

We use the above inequality to get tight bounds for the value of $\sum_{i=1}^{m} Y_i/m$
how to appl

THEORE
$[0,1]$ *with*
sequence of
Fix any $\delta >$

$\varepsilon_m =$

Then:

PROOF.

$$\Pr\left[\exists m, 1 \le m \le N : \left|\frac{\sum_{i=1}^{m} Y_i}{m} - \mu\right| > \varepsilon_m\right]$$

$$\le \sum_{r \ge 1} \Pr\left[\exists m, \kappa^{r-1} \le m \le \kappa^r : \left|\frac{\sum_{i=1}^{m} Y_i}{m} - \mu\right| > \varepsilon_m\right]$$

$$\le \sum_{r \ge 1} \Pr\left[\exists m, \kappa^{r-1} \le m \le \kappa^r : \left|\frac{\sum_{i=1}^{m} Y_i}{m} - \mu\right| > \varepsilon_{\kappa^r}\right]$$

$$\le \sum_{r \ge 1} \Pr\left[\max_{\kappa^{r-1} \le m \le N-1} \left|\frac{\sum_{i=1}^{m} Y_i}{m} - \mu\right| > \varepsilon_{\kappa^r}\right].$$

LEMMA 3. *Fix* $i \in 1 \ldots k$. *Define* m_i^* *to be the minimal value of* $m \ge 1$ *for which* $\varepsilon_m < \eta_i/4$. *In the running of the algorithm, if for every* $j \in A_{m_i^*}$, *we have* $|\nu_{j,m_i^*} - \mu_j| \le \varepsilon_{m_i^*}$, *then* $m_i \le m_i^*$.

Intuitively, the lemma allows us to establish that $m_i < m_i^*$, the latter of which (as we show subsequently) is dependent on η_i.

PROOF. If $i \notin A_{m_i^*}$, then the conclusion of the lemma trivially holds, because $m_i < m_i^*$. Consider now the case where $i \in A_{m_i^*}$.

nd only if the
the union of

definition of
stronger in-
mma (i.e., that
we have that
have:

$\le \nu_{i,m_i^*} - \varepsilon_{m_i^*}$

he confidence

$- \varepsilon_{m_i^*}$;
$< \eta_i/4$;

- and the third follows from the fact that $\mu_j \le \mu_i - \eta_i$.

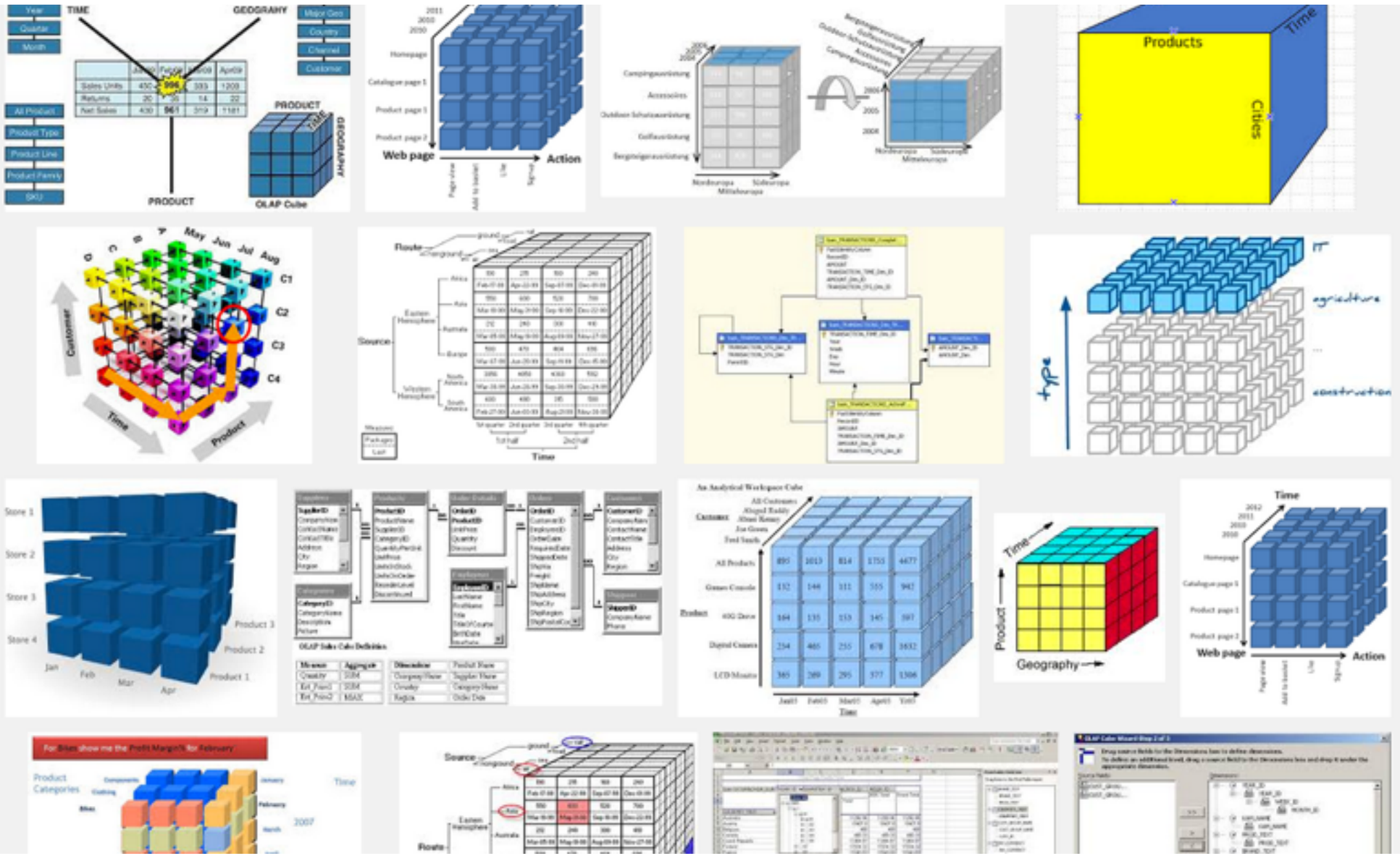
Thus, the intervals $[\nu_{i,m_i^*} - \varepsilon_{m_i^*}, \nu_{i,m_i^*} + \varepsilon_{m_i^*}]$ and $[\nu_{j,m_i^*} - \varepsilon_{m_i^*}, \nu_{j,m_i^*} + \varepsilon_{m_i^*}]$ are disjoint. Similarly, for all $j \in A_{m_i^*}$ that satisfies $\mu_j > \mu_i$, we observe that the interval $[\nu_{i,m_i^*} - \varepsilon_{m_i^*}, \nu_{i,m_i^*} + \varepsilon_{m_i^*}]$ is also disjoint from $[\nu_{j,m_i^*} - \varepsilon_{m_i^*}, \nu_{j,m_i^*} + \varepsilon_{m_i^*}]$. □

We are now ready to complete the analysis of the algorithm.

Do you know the one about the physics student who asked his professor how much math he needed to know?

How do we solve it?

- Big idea: **stratified samples**
- Big idea: only **preserve visually important properties**
- **Sample the subset that is most likely to change the output where it matters**



Data Cubes

Let’s talk aggregation

Relation A

Country	*Device*	*Language*
US	Android	en
US	iPhone	ru
South Africa	iPhone	en
India	Android	en
Australia	iPhone	en

Aggregation B

Country	*Device*	*Language*	*Count*
All	*All*	*All*	5

Group By on *Device, Language* C

Country	*Device*	*Language*	*Count*
All	Android	en	2
All	iPhone	en	2
All	iPhone	ru	1

Cube on *Device, Language* D

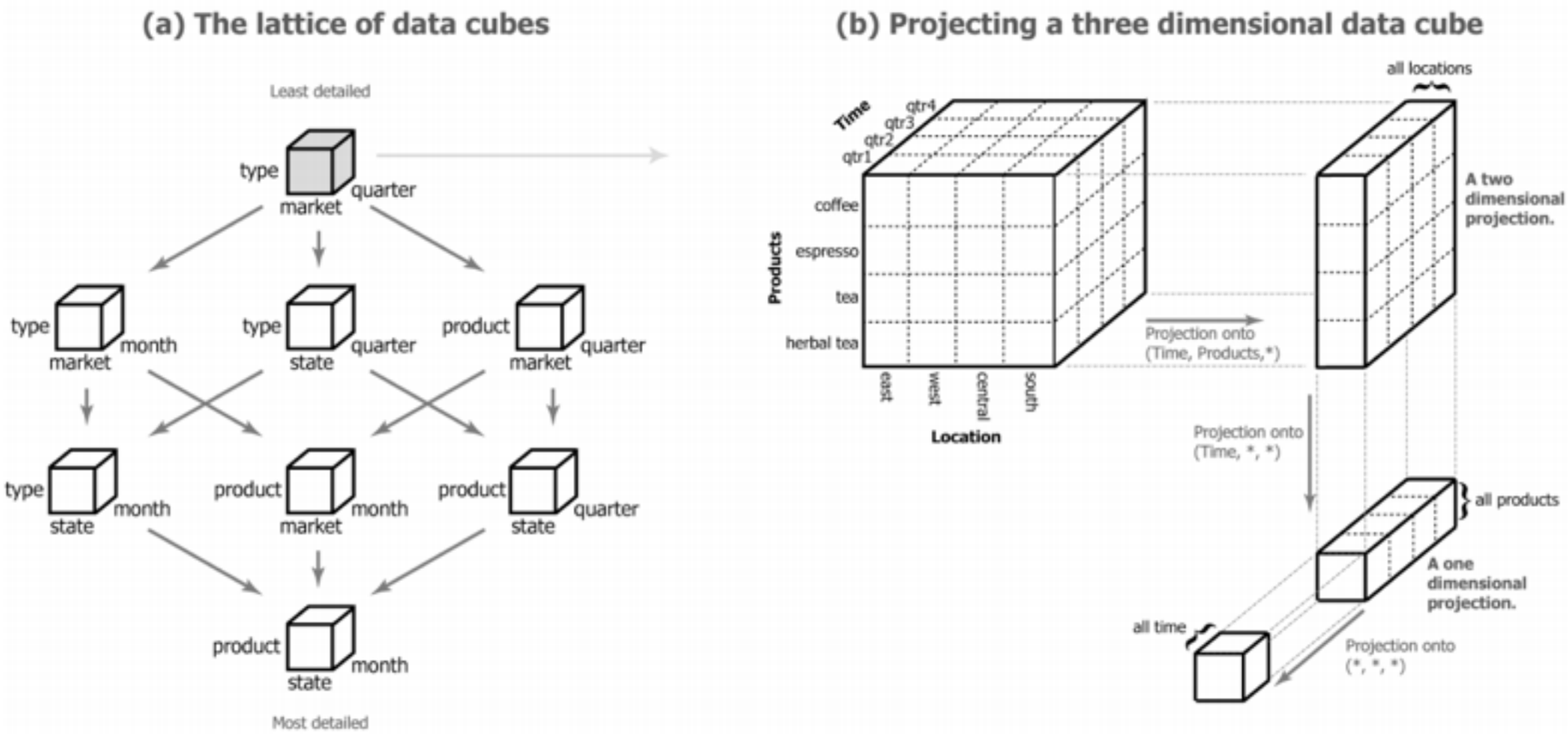
Country	*Device*	*Language*	*Count*
All	*All*	*All*	5
All	Android	*All*	2
All	iPhone	*All*	3
All	*All*	en	4
All	*All*	ru	1
All	iPhone	ru	1
All	Android	en	2
All	iPhone	en	2

Equivalent to Group By on all possible subsets of {*Device, Language*}

Data Cubes

Let’s talk aggregation

Data Cubes: aggregate by collapsing attributes



Multiscale Visualization using Data Cubes, Stolte et al., Infovis 2002

Data Cubes

- There are other axes of aggregation besides columns that we also care about in visualization

- For example, ranges

Data Cubes

- There are other axes of aggregation besides columns that we also care about in visualization
- For example, ranges:
  - How many cars sold between 1995 and 1999?
    - 1997 and 2001? 2001 and 2002?
- How do we make it go fast?

immens: Liu, Jiang, Heer, Eurovis 2013

- Preaggregate some dimensions into “data tiles”
- Compute final aggregations on GPUs
- **Incredibly fast and simple**
- Decide on spatial resolution ahead of time
  - Somewhat limited querying power

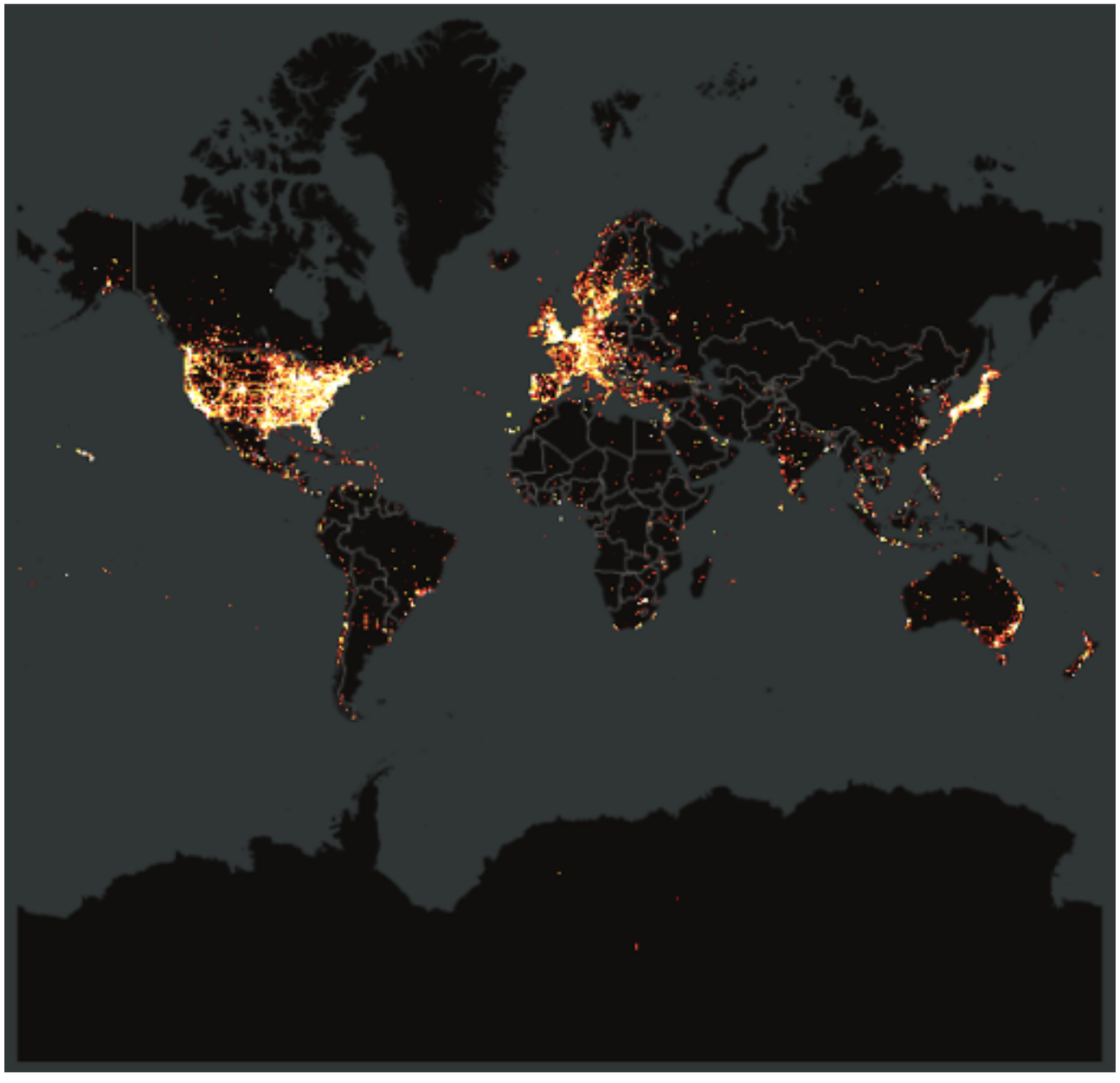
Demo time

- http://vis.stanford.edu/projects/immens/demo/brightkite/

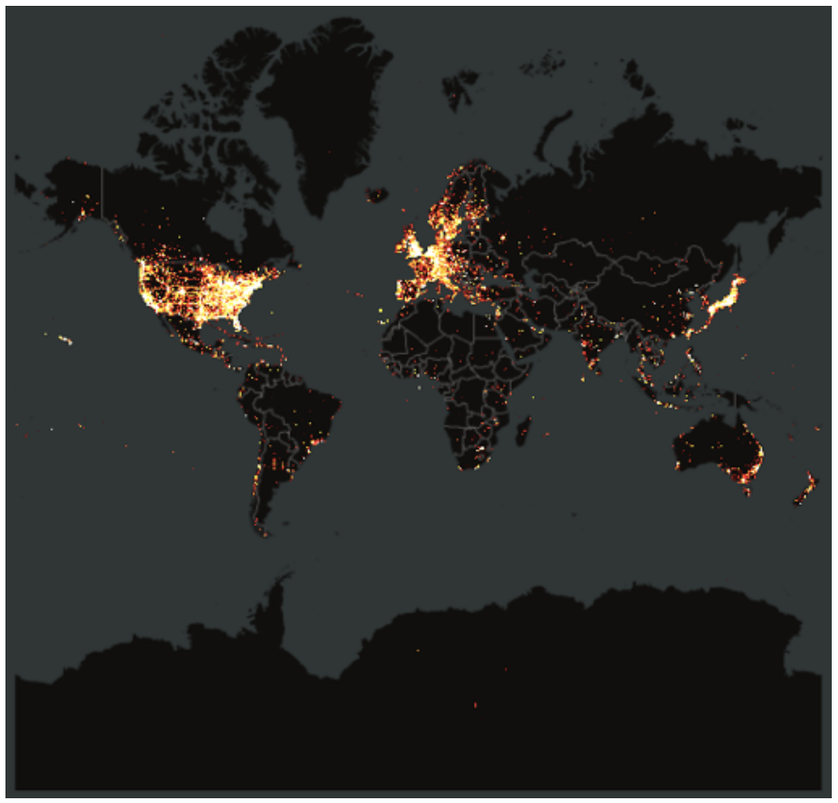
nanocubes: Lins, Klosowski, Scheidegger 2013

- Many aggregations **overlap**
- Build data structure where aggregations over multiple scales are compactly stored and easily combined
- Sufficiently fast (network latency dominates)
- Implementation is more involved, memory usage not ideal

Query: produce a count heatmap of the world for all points in my database



Query: produce a count heatmap of the world for all points in my database

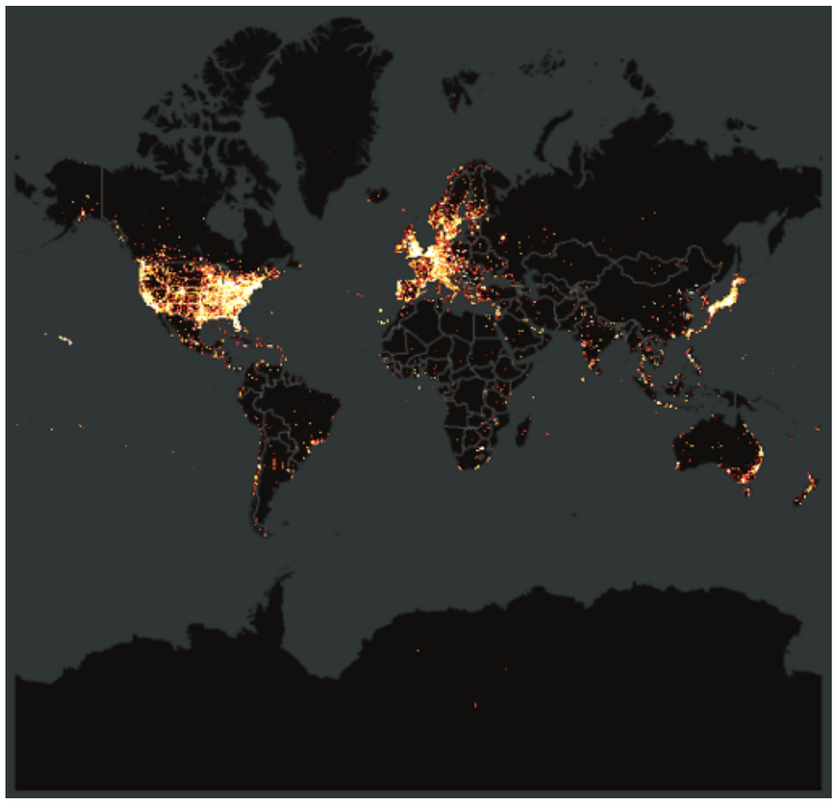


latitude	longitude	device	time
23.4	-40.3	iPhone	Aug 3, 2011 10:00
31.2	-41.3	Android	Aug 3, 2011 10:05
27.8	-39.3	iPhone	Aug 3, 2011 10:07
...	...	...	...
26.1	-38.1	Android	Jun 8, 2012 21:03
27.2	-44.3	Android	Jun 8, 2012 21:04
24.2	-39.7	iPhone	Jun 8, 2012 21:10

n

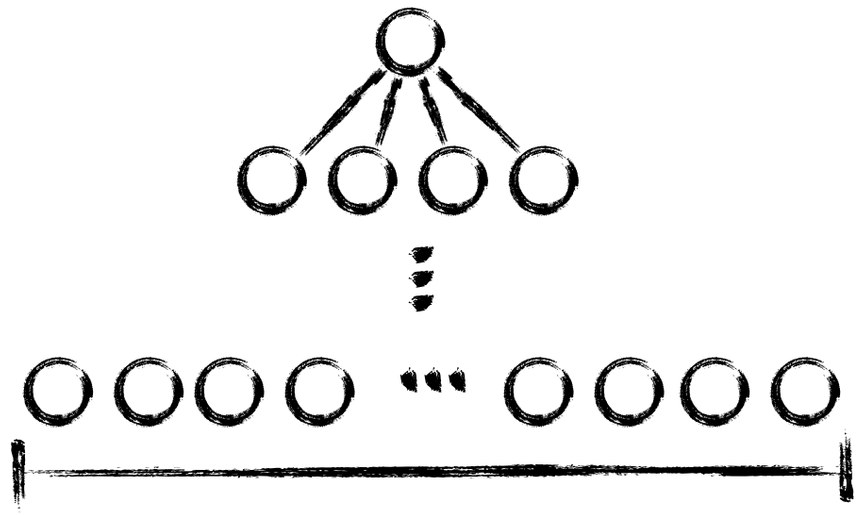
if no aggregation was pre-computed then this query is proportional to “n”

Query: produce a count heatmap of the world for all points in my database



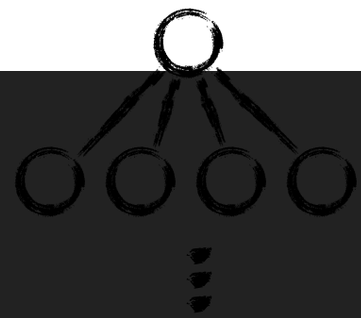
latitude	longitude	device	time
23.4	-40.3	iPhone	Aug 3, 2011 10:00
31.2	-41.3	Android	Aug 3, 2011 10:05
27.8	-39.3	iPhone	Aug 3, 2011 10:07
...	...	...	...
26.1	-38.1	Android	Jun 8, 2012 21:03
27.2	-44.3	Android	Jun 8, 2012 21:04
24.2	-39.7	iPhone	Jun 8, 2012 21:10

n



if we pre-aggregate counts (e.g. quadtree) the query time becomes proportional to the number of reported pixels

Query: produce a count heatmap of the world for all points in my database



What about brushing?

latitude	longitude	device	time
23.4	-40.3	iPhone	Aug 3, 2011 10:00
31.2	-41.3	Android	Aug 3, 2011 10:05
27.8	-39.3	iPhone	Aug 3, 2011 10:07
...	...	...	...
26.1	-38.1	Android	Jun 8, 2012 21:03
27.2	-44.3	Android	Jun 8, 2012 21:04
24.2	-39.7	iPhone	Jun 8, 2012 21:10

n

if we pre-aggregate counts (e.g. quadtree) the query time becomes proportional to the number of reported pixels

nanocubes: Lins, Klosowski, Scheidegger 2013

- Simple 1D example

nanocubes: Lins, Klosowski, Scheidegger 2013

- Simple 2D example

Demo time

- http://nanocubes.net
- http://hdc.cs.arizona.edu/mamba_home/~cscheid/flights_test/