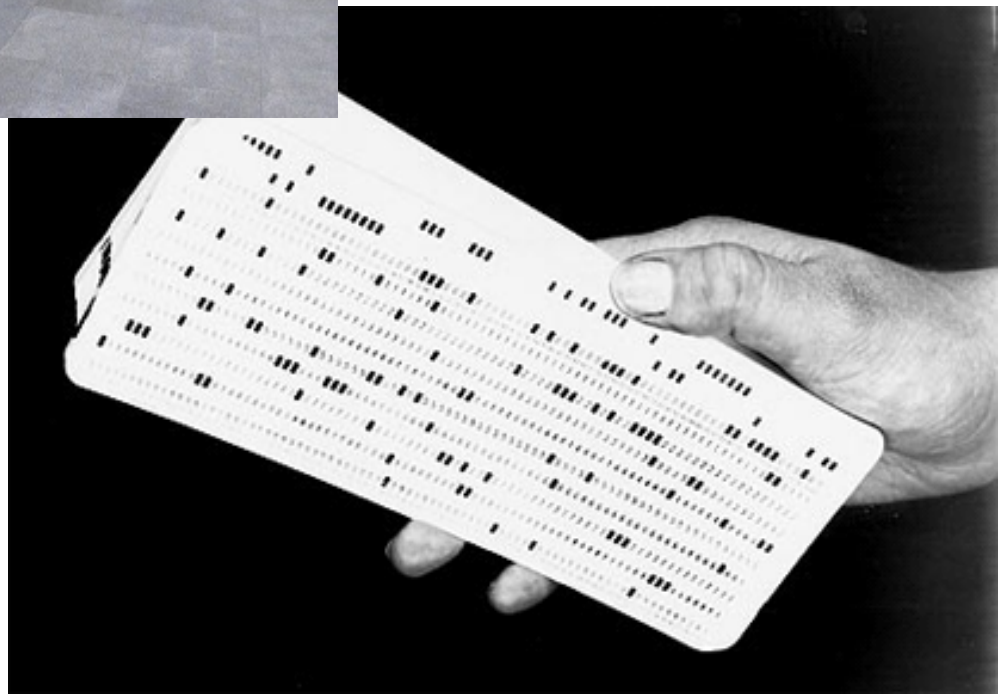
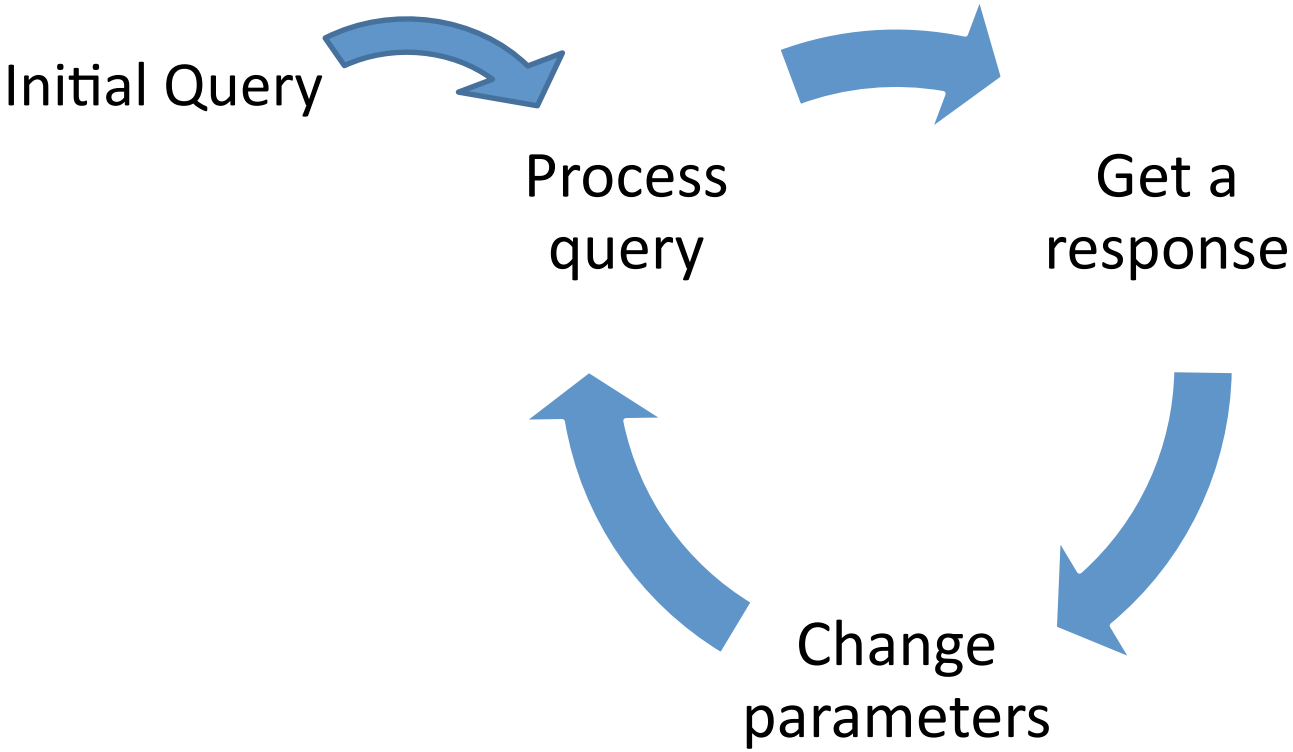
INCREMENTAL, APPROXIMATE DATABASE QUERIES AND UNCERTAINTY FOR EXPLORATORY VISUALIZATION

Danyel Fisher
Microsoft Research



Exploratory Visualization



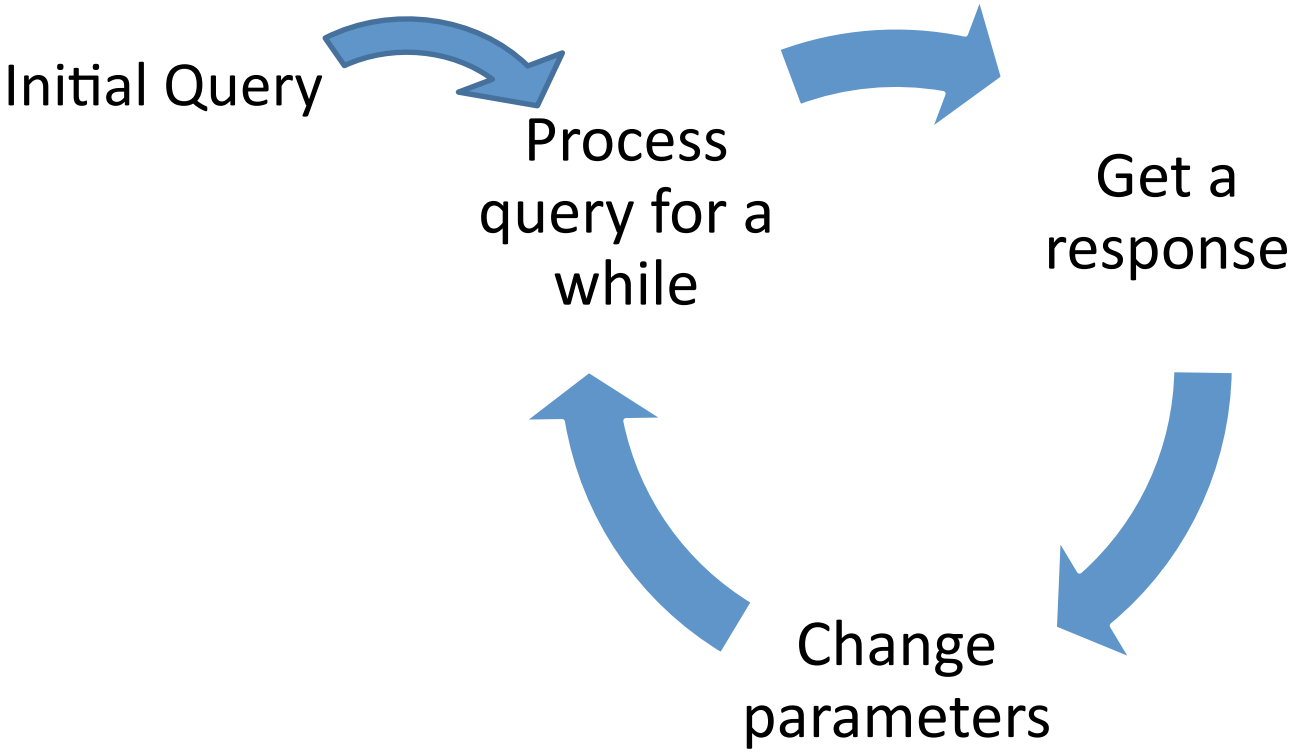
Handling Big Data for Infovis

- Megabytes: More data than there are pixels on screen
  - Need to summarize, zoom
- Gigabytes: Mor re in memory
  - Need to think
- Terabytes: Mor re on a single disk
  - Yo

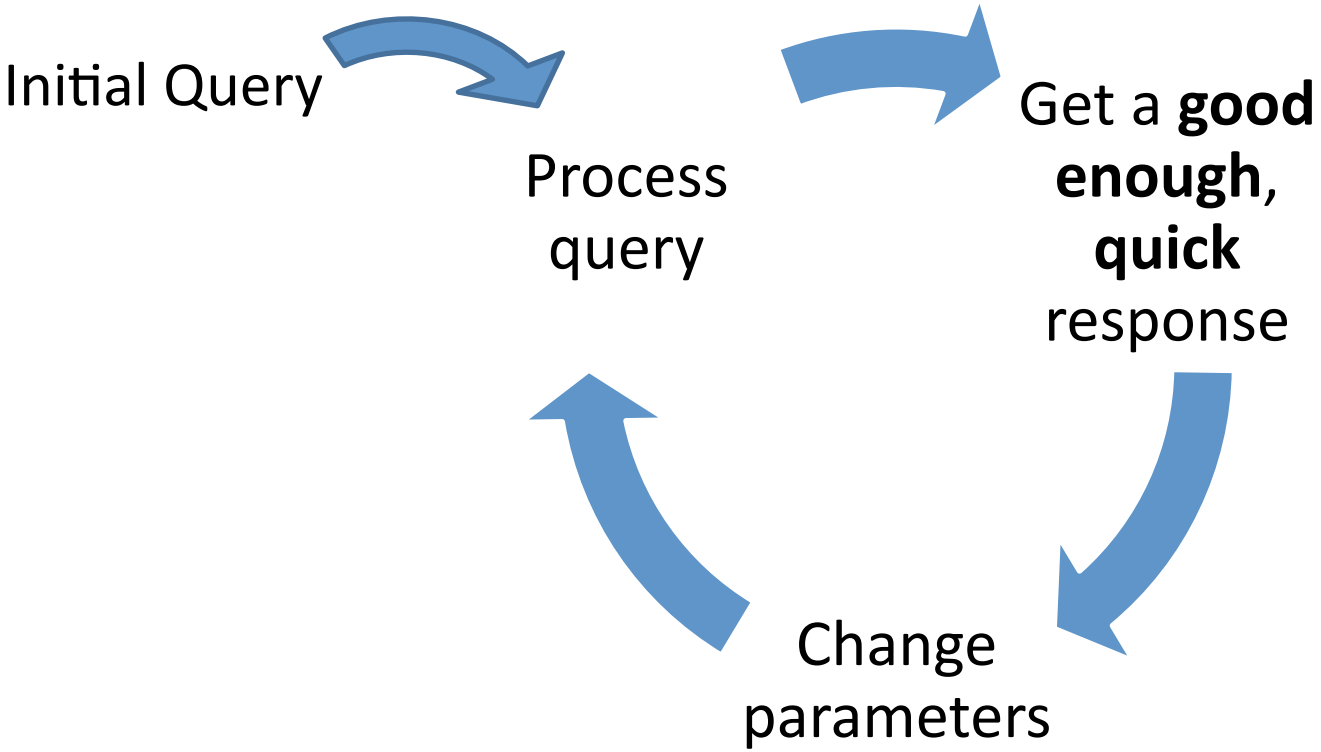
Extreme Visualization:
Squeezing a Billion Records into a Million Pixels

Ben Shneiderman
Human-Computer Interaction Lab & Department of Computer Science
University of Maryland
College Park, MD 20742
ben@cs.umd.edu

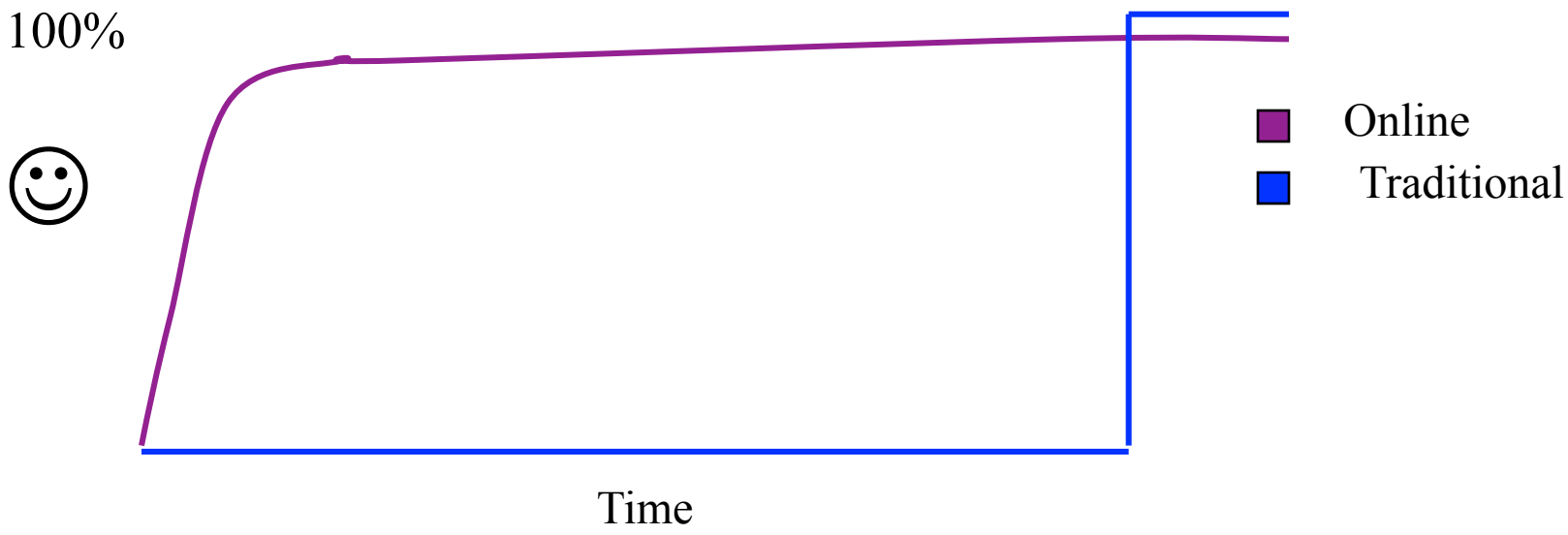
Big Data Visualization



Exploratory Big Data Visualization



Happiness over time



What is “good enough”?

- “I can act on this query”
- “I realize that this query is incorrect”
  - create a new query
- “I want a detailed response”
  - Wait for the full query to complete

What is “quick”?

- Milliseconds: Feels real-time
- Seconds: Laggy but possible
- Minutes: Forget context
- Hours: Forget question

Part I: Incremental Databases

Part II: Visualizations

Part III: A Prototype

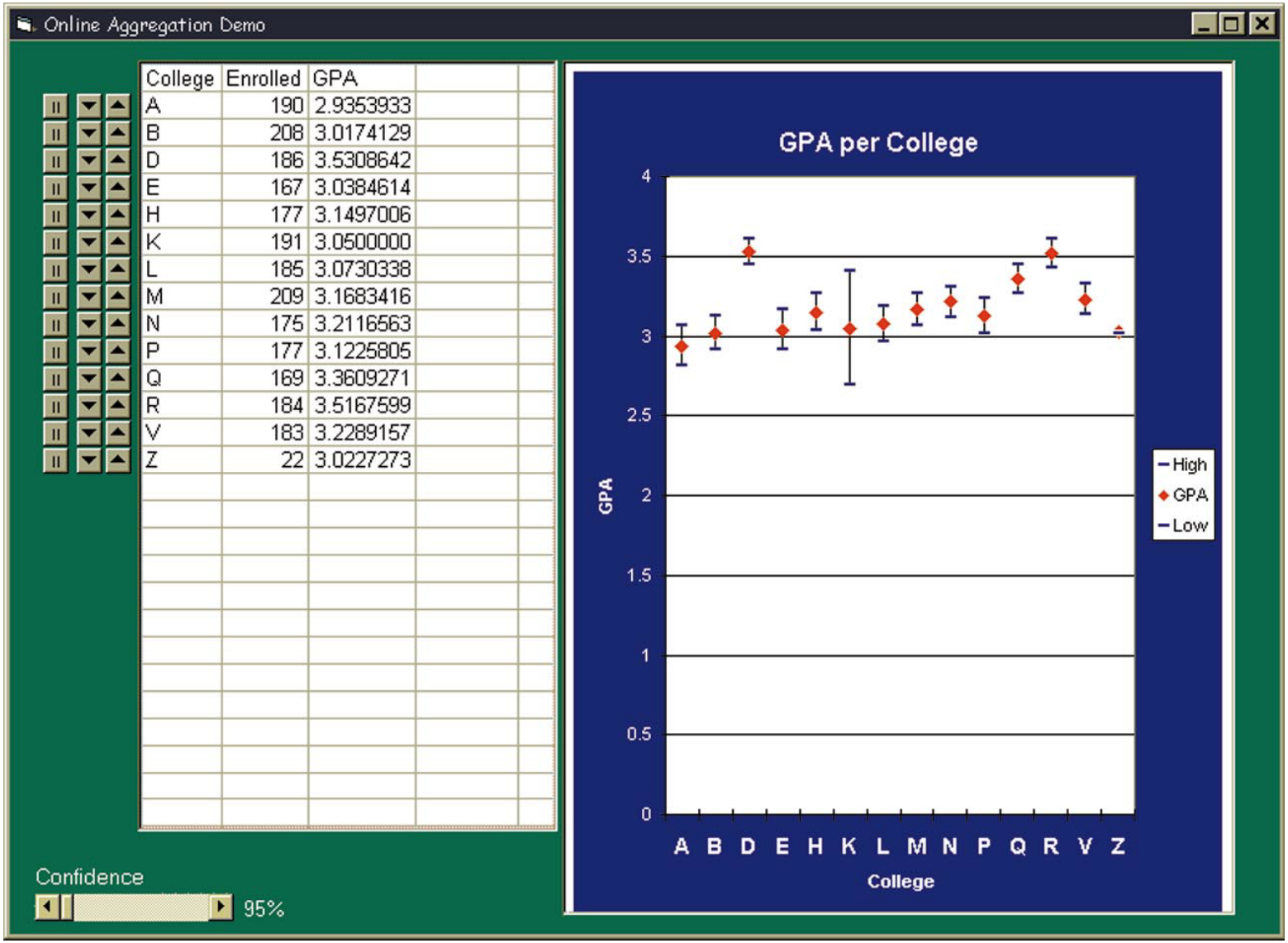
Part I

INCREMENTAL DATABASES

Techniques for Speeding Big Data

- **Pre-aggregate (e.g. OLAP)**
  - Fast but inflexible
- **Parallel Computation**
  - Hadoop Pig, Sawzall, DryadLinq: use existing data structures and add visualizations
  - Dremel: Use novel data structures
- **Sampling & approximate queries**

Control (1999)



Sampling and Approximate Queries

- Joe Hellerstein (et al)'s CONTROL project
- General concept:
  - Grab a little bit (more) of the database quickly
  - Estimate value & size of confidence interval
  - Repeat
- What can we do? **Aggregate**.
  - Some aggregates are very good. **AVERAGE. COUNT. SUM.**
  - Some aggregates are really bad. **MAX (or Top-K). MIN.**
  - Some aggregates have loose approximations. **PERCENTILE. COUNT DISTINCT.**

Is that powerful enough?

- Some things are just histograms:
  - Bar chart (sum)
  - Tag Clouds
  - Treemap (multilevel sum)
- Don't do a scatterplot, do a 2D histogram
- Even some machine learning:
  - K-Means: Locate average of group, find centroids, repeat

Computing Confidence Interval

- Estimator
  - Total elements
  - Mean seen so far
  - Number of elements that cross the filter so far
- Intuition: if you know about how the data you've seen so far behaves, you can guess the rest. Based on:
  - Standard deviation seen so far
  - Also nice: data min/max
  - Some theorems: std dev overall

Why Not Just Do a Straight Sample?

- Don’t know how good you are without confidence intervals
- May need *larger* sample (over memory) to get tight intervals.
- How big is big enough?

Why Isn't Everyone Doing This?

- A good sample is random … but a random sample requires accessing (potentially) all rows
- Need to maintain some data for bounds
  - E.g. column min, max
- Databases don't support incremental callbacks
- Joins can be tricky (but NoSQL?)

Part II

THE VISUALIZATION CHALLENGES

Uncertainty Visualization

- “Confidence” is something like “uncertainty”
- Lots of sources of uncertainty have been studied
  - Credibility of sources
  - Model uncertainty
  - Simulation uncertainty
  - Incompleteness

- *Statistical and Quantitative Uncertainty*

Olston & Mackinlay (2002)

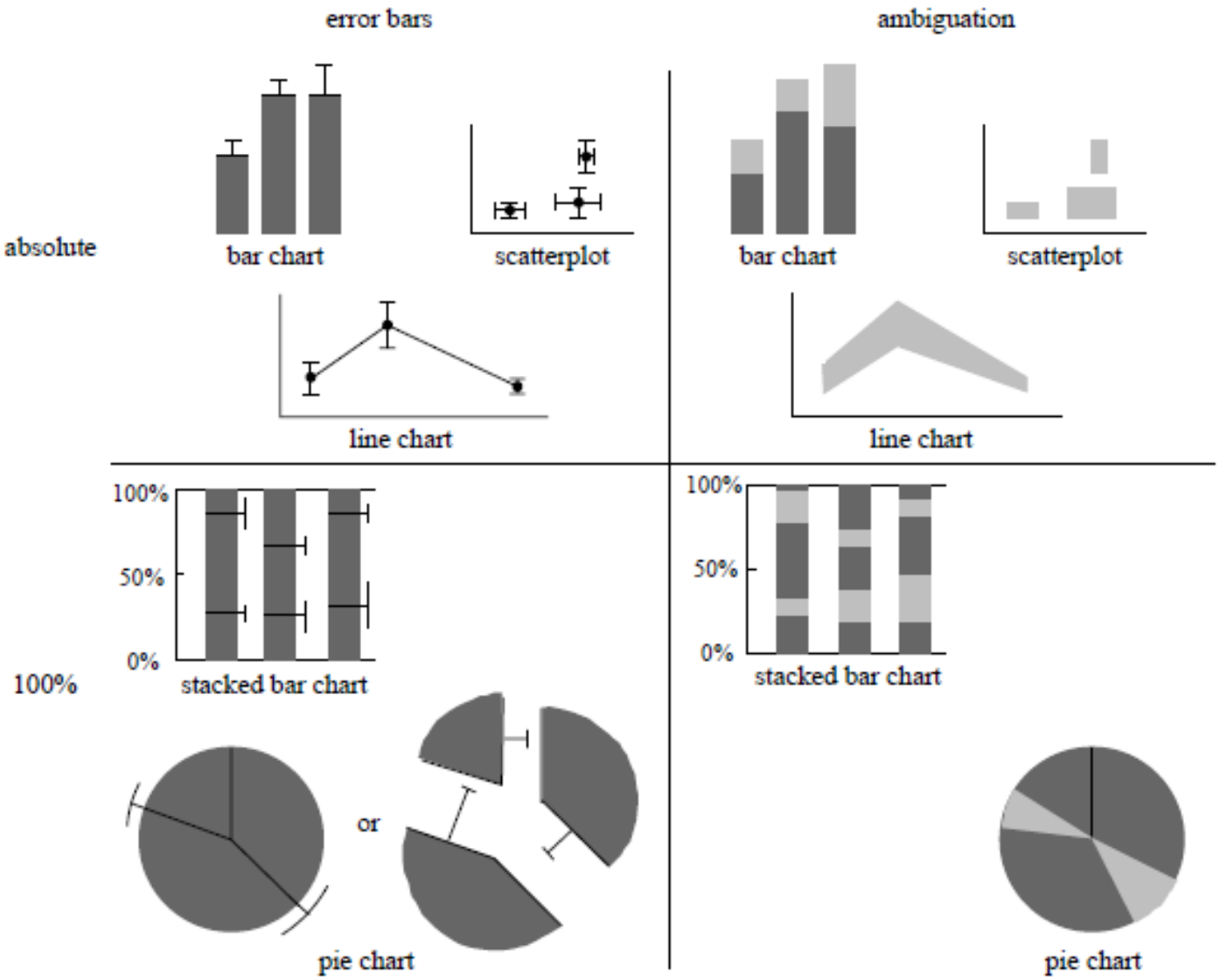
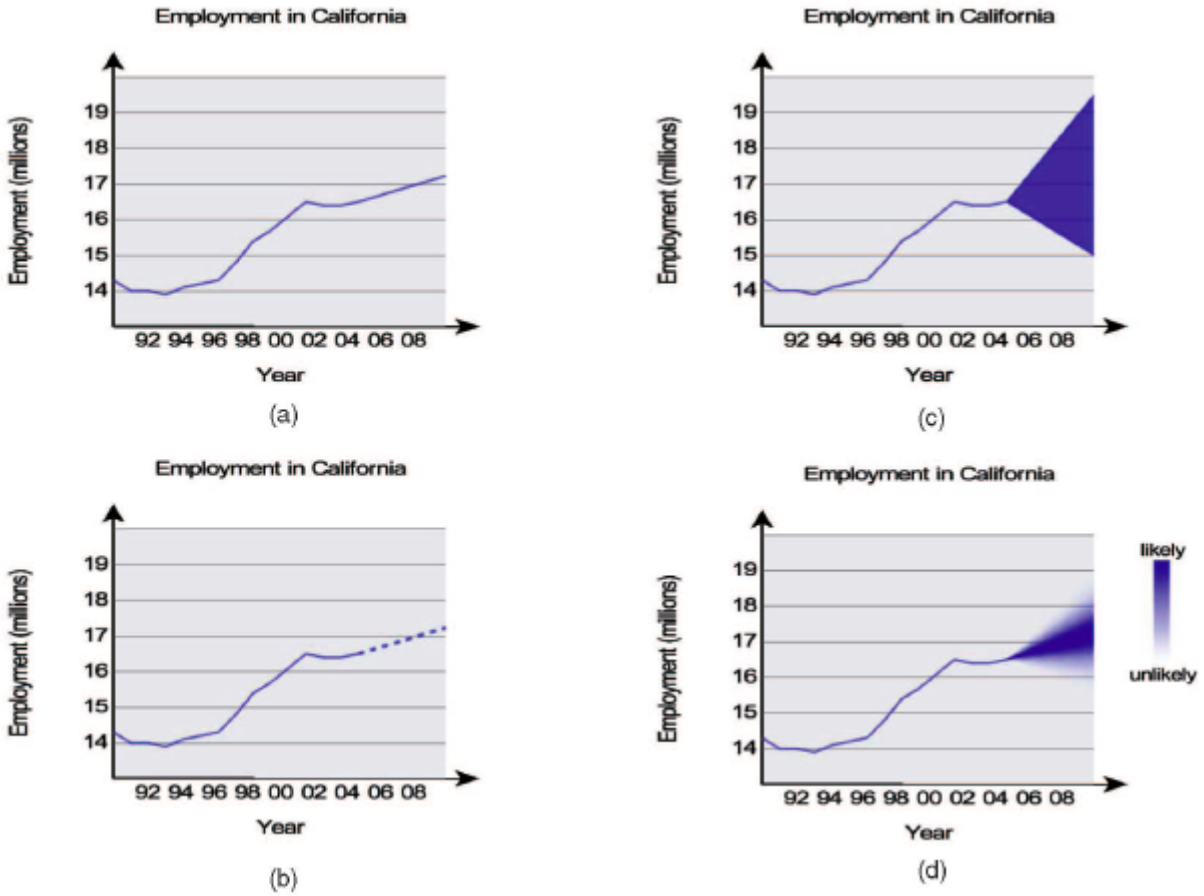


Figure 1: Error bars and ambiguation applied to some common chart types.

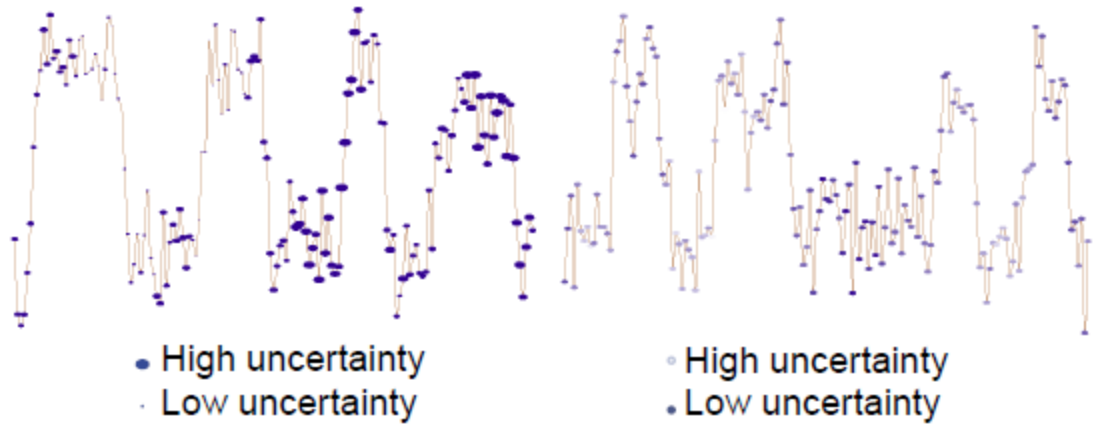
Streit & Pham (2008)

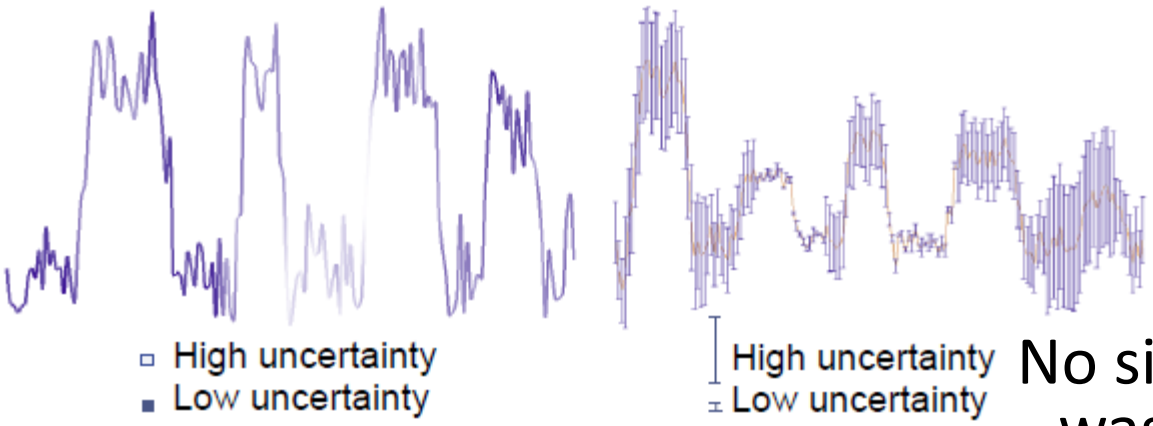


Graph	Growth Rate	Known / Assumed
(a)	0.147	it is certainly 0.147
(b)	$0.147, estimated = true$	it is not necessarily 0.147
(c)	$[-0.3, 0.6]$	it is between -0.3 and 0.6
(d)	$\mu = 0.147, \sigma = 0.1$	it is probably 0.147

Fig. 1. Visualizations of employment numbers in California. Years 2005-

User Study of Uncertainty
Sanyal Zhang et al (2009)





No single technique was clear winner

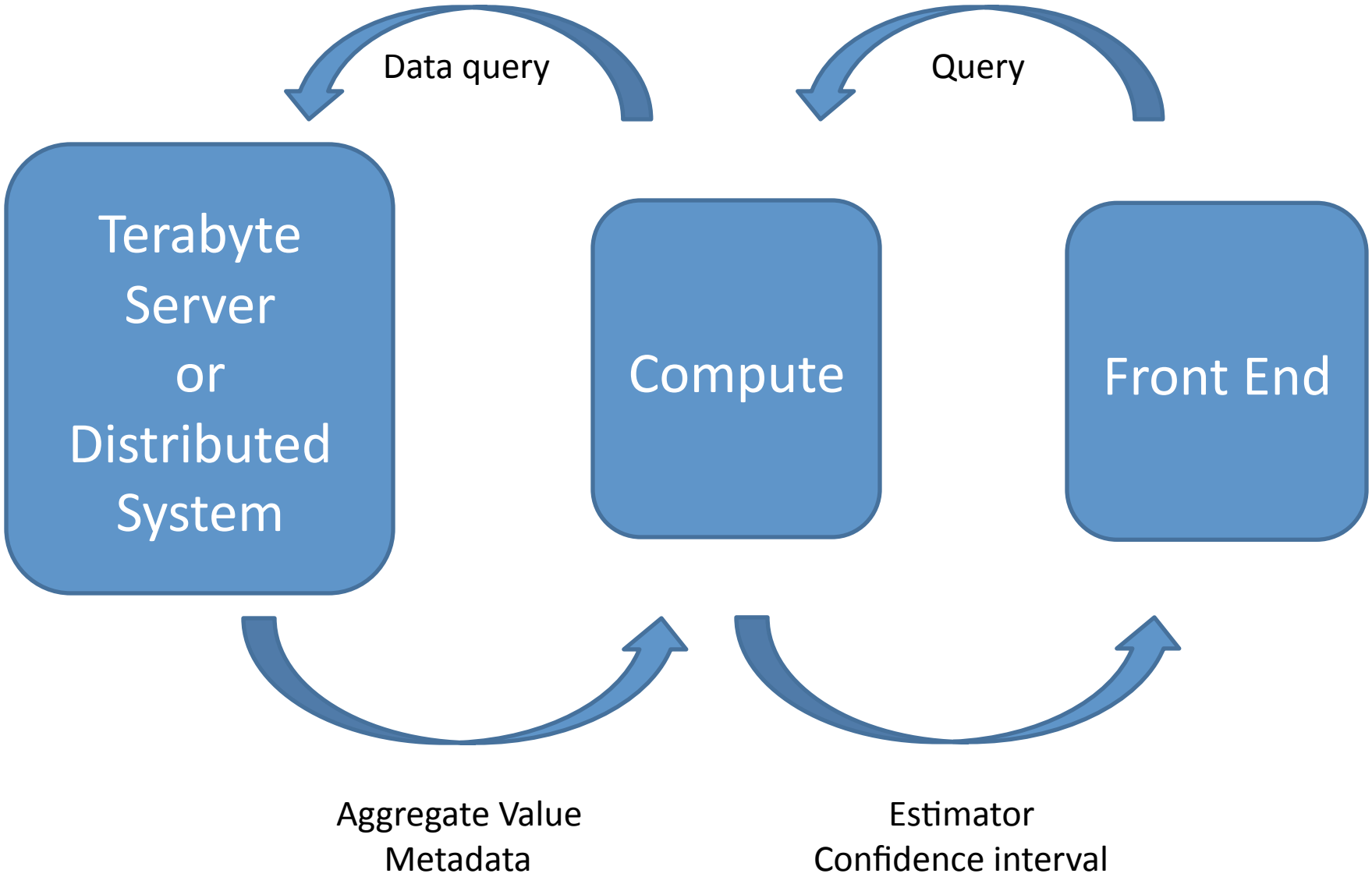
Part III

PROTOTYPING IT

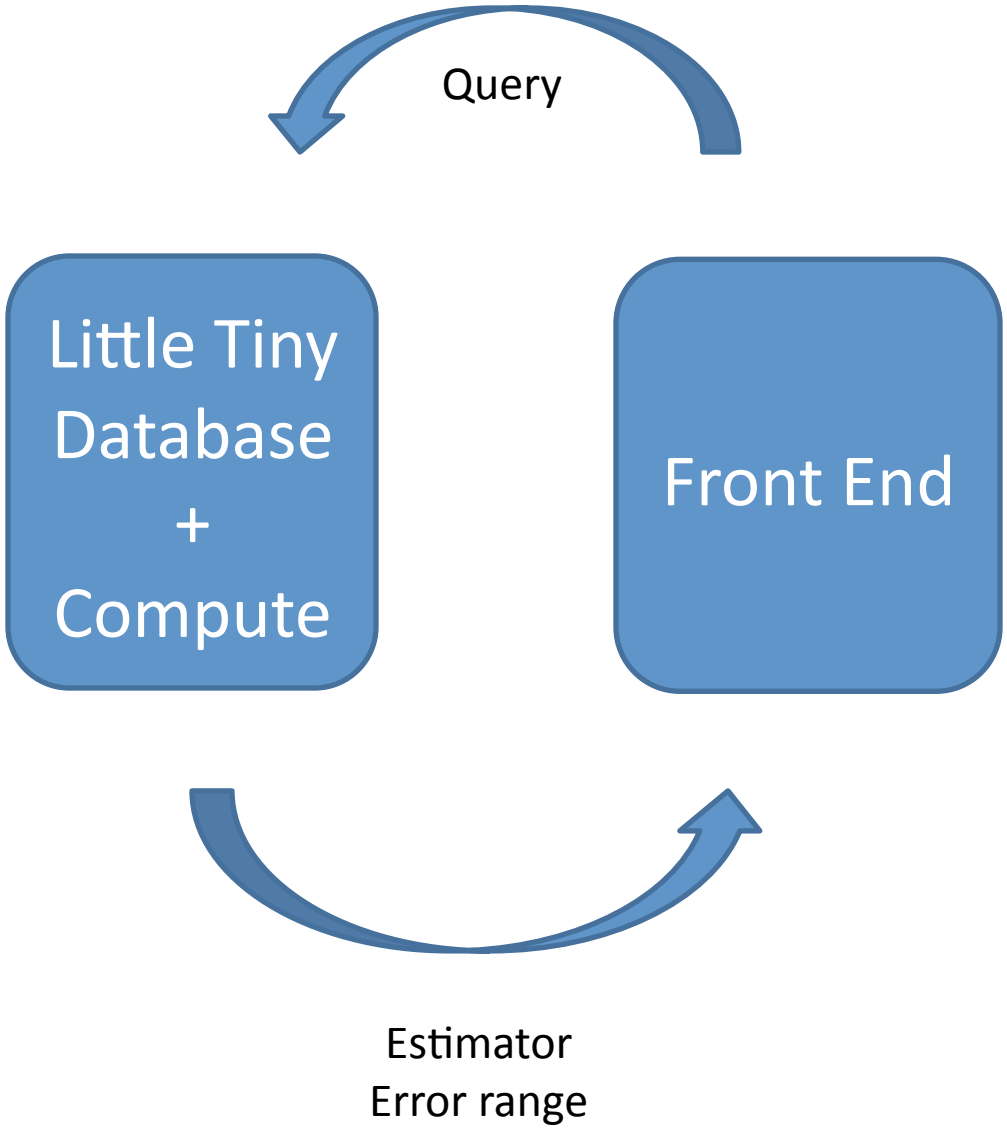
Why Prototype a Front End?

- Back-End
  - Technical implementation
  - Adapt to NoSQL
  - Ways to guide sampling
- Front-End
  - What visual codings work beside error bars? How do we extend to multiple dimensions?
  - How good is “good enough”?
    How fast is “fast enough”?
  - What sorts of problems work well?
  - User experience of these systems
  - Reducing communication overhead

A Full System View



The Desktop Edition



Call to Action

There’s work to be done here

and a very compelling source of approximate data.

Let’s build these!

Thank you!
(this work is not sponsored by DOE)

DANYELF@MICROSOFT.COM

RESEARCH.MICROSOFT.COM/~DANYELF

BONUS SLIDES

Joins

- Lots of database research dedicated to joins
  - "Hash ripple" join
  - 10% sample, twice, is a 1% sample assuming independence
- Sentinel joins:
  - Some joins are impossible to do incrementally

  E.g. (select count(direct reports) where manager=president)
- Or live without them
  - NoSQL has very limited (and expensive) joining
  - Denormalized tables for distributed computation