Varieties of “big data”...
Tall Data

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

How to manage?
Parallel data processing
Reduction: Filter, aggregate
Sample or approximate

Not just about systems. Consider perceptual / cognitive scalability.
Tall Data

Wide data

Lots of variables (100s-1000s…)
Select relevant subset
Dimensionality reduction
Statistical methods can suggest and order related variables
Requires human judgment
Tall Data

Wide data

Diverse data
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
Data

Sampling
Data

Sampling

Modeling
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

<table>
<thead>
<tr>
<th></th>
<th>Numeric</th>
<th>Ordinal</th>
<th>Temporal</th>
<th>Geographic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D</td>
<td><strong>Histogram</strong></td>
<td><strong>Bar Chart</strong></td>
<td><strong>Line Graph</strong>/</td>
<td><strong>Choropleth Map</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Area Chart</strong></td>
<td></td>
</tr>
<tr>
<td>2D</td>
<td><strong>Binned Scatter Plot</strong></td>
<td><strong>Heatmap</strong></td>
<td><strong>Temporal Heatmap</strong></td>
<td><strong>Geographic Heatmap</strong></td>
</tr>
</tbody>
</table>
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]
The non-normalized heatmap suffers from artifacts, seen as vertical stripes. Binned charts convey high points across the top, a collective dip in stocks during the crash of 2008, and two distinct bands of $25 and $15 stocks.
Example: Density Line Chart

- **Time Series**
- **Repeat for each series**
- **Non-Normalized**
Example: Density Line Chart [Moritz & Fisher]

Time Series

Repeat for each series

Non-Normalized

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Sum: 2  2  2  2  1  3  2  2  2  2
Example: Density Line Chart

Time Series

Repeat for each series

Non-Normalized

Approx. Arc-Length Normalized
Example: Density Line Chart

Time Series

Repeat for each series

Non-Normalized

Approx. Arc-Length Normalized

Aggregate

Color
Example: Density Line Chart  [Moritz & Fisher]

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

Without normalization, the steep lines are over-represented.
2. Enabling Real-Time Interaction
Interactive Scalability Strategies

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

1. **Query Database**  Offload to a scalable backend
   Tableau, for example, issues aggregation queries.
   Analytical databases are designed for fast, parallel execution.
   But round-trip queries to the DB may still be too slow…

2. **Client-Side Indexing / Data Cubes**

3. **Prefetching**

4. **Approximation**
Interactive Scalability Strategies

1. Query Database

2. Client-Side Indexing / Data Cubes  Query data summaries
   Build sorted indices or data cubes to quickly re-calculate aggregations as needed on the client.

3. Prefetching

4. Approximation
Interactive Scalability Strategies

1. Query Database
2. Client-Side Indexing / Data Cubes
3. **Prefetching** Request data *before* it is needed
   Reduce latency by speculatively querying for data before it is needed. Requires prediction models to guess what is needed.
4. Approximation
Interactive Scalability Strategies

1. Query Database
2. Client-Side Indexing / Data Cubes
3. Prefetching
4. Approximation  Give fast, approximate answers

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

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

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

[Liu, Jiang & Heer ‘13]

Strategies: Client-Side Data Cubes
Sampling
Google Fusion Tables

Binned Aggregation
imMens
Binned Aggregation

imMens
5-D Data Cube

Month, Day, Hour, X, Y

~2.3B bins
5-D Data Cube

Month, Day, Hour, X, Y

~2.3B bins
Multivariate Data Tiles

1. Send data, not pixels
2. Embed multi-dim data
Full 5-D Cube
For any pair of 1D or 2D binned plots, the maximum number of dimensions needed to support brushing & linking is **four**.
Full 5-D Cube

Σ

Σ

Σ

Σ

3-D cubes

3-D data tiles

13 3-D Data Tiles
Full 5-D Cube → ~2.3B bins

13 3-D Data Tiles → ~17.6M bins (in 352KB!)
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
~50fps querying of visual summaries of 1B data points.
Limitations and Questions

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

Does super-low-latency interaction really matter?
Is it worth it to go to all of this trouble? (Short answer: yes!)
High latency leads to reduced analytic output [Liu & Heer, InfoVis 2014]
Administrivia
Final Project Schedule

Proposal — Fri Feb 18
Prototype — Mon Feb 28
Demo Video — Wed Mar 9
Video Showcase — Thu Mar 10 (in class)
Deliverables — Mon Mar 14

Logistics
Starting planning your video now!
Read the video guide on the course website.
Tell a story, don’t just catalog features.
How does interactive latency affect exploratory analysis with visualizations?

[Liu & Heer ’14]
Prior Work

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

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

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

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

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

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

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

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

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

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

But what about open, exploratory analysis tasks?
Experiment Design

2 (Latency) x 2 (Scenario) Design

Latency: +0ms / +500ms
Scenario: Mobile Check-ins / FAA Flight Delays

Exploratory Analysis Tasks (2 per session)
imMens with brush, pan, zoom, adjust scales
Users asked to explore data and share findings
Log events, record audio and screen capture

16 subjects, all familiar with data analysis + vis
140m FAA Flight Delay Records
Data Collection & Analysis

Event Log Analysis
Analyze triggered & processed user input events
Assess data set coverage (# unique tiles)

Verbal Protocol Analysis
Think-aloud protocol: verbalize thought process
Transcribe sessions; Code actions and insights
Analyze number and type of coded events
Latency Study Results

Higher latency leads to...
Latency Study Results

Higher latency leads to...
Reduced user activity and data set coverage
Latency Study Results

Higher latency leads to...
Reduced user activity and data set coverage
Significantly fewer brushing actions
Latency Study Results

Higher latency leads to...
Reduced user activity and data set coverage
Significantly fewer brushing actions
Less observation, generalization & hypothesis

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

Latency Coefficient
Latency Study Results

Higher latency leads to...
Reduced user activity and data set coverage
Significantly fewer brushing actions
Less observation, generalization & hypothesis

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

Higher latency leads to...
Reduced user activity and data set coverage
Significantly fewer brushing actions
Less observation, generalization & hypothesis

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

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

Higher latency leads to...
Reduced user activity and data set coverage
Significantly fewer brushing actions
Less observation, generalization & hypothesis

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

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

**In short: milliseconds matter!** And imMens was not a waste of time… 😊
ForeCache

[Battle, Chang, & Stonebraker ’16]

Strategies: Query Database, Prefetching
ForeCache is also a Data Tile-Based System

Manage a Cache of Tiles from DB

Example Tile-Based Views

(a) Satellite Imagery  
(b) Multidimensional  
(c) Timeseries (Heart rate Monitoring)
Key Idea: Model & Predict User Behavior

1. Classify Analysis Phase

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

Train a machine learning classifier (SVM) to predict phase. The input data is the activity trace of user interactions.
Key Idea: Model & Predict User Behavior

1. Classify Analysis Phase

2. Apply Prediction Models

Actions-Based: Use recent interactions to predict next ones.
   You pan left twice; what is the probability you will do it again?

Signature-Based: Match to data characteristics of interest.
   What data tiles are visually similar to current focus tiles?

These models are weighted based on the analysis phase.
Actions-Based for navigation. Signature-Based for sensemaking. Both applied equally for foraging.
Application: MODIS Satellite Data

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

ForeCache improves latency:
- 430% better than current non-prefetching systems
- 88% better than existing prediction methods
Falcon
[Moritz, Howe, & Heer ’19]

Strategies: Query Database, Client-Side Data Cubes, Prefetching
Falcon
uwdata.github.io/falcon
0.008% of the data
How does Falcon support fine-grained real-time interaction?
Brushing is more common and people are sensitive to latencies. Prioritize brushing latency over view switching latency.

5x speedup

View switch delay → distance
Key Idea:

User-centered prefetching and indexing to support all brushing interactions with one view. Re-compute if the user switches the view.
brushes in the precomputed view

serves requests from a data cube

bruses in the precomputed view

serves requests from a data cube


interacts with a new view

query for new data cubes
 Constant data & time. Client only.

💡 Aggregation decouples interactions from queries over the raw data.

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


```
brushes in the precomputed view
```

```
serves requests from a data cube
```

```
interacts with a new view
```

```
query for new data cubes
```
1.7 B stars.
1.2 TB of data.
Visualizations running in my browser.
Data stored in OmniSci database.
"With Falcon it feels like I'm really interacting with my data."

Data Platform Engineer at Stitch Fix
What if data is too large to query in a reasonable time?
Trust, but Verify: Optimistic Vis

[Moritz, Fisher, Ding & Wang ‘17]

Strategies: Query Database, Approximation
Latencies reduce engagement and lead to fewer observations.

Approximation: Trade Accuracy for Speed

Approximate query processing (AQP)
Uncertainty estimation in statistics
Uncertainty visualization
Probabilistic programming
Approximate hardware
Pick your poison:
1. Trust the approximation, or
2. Wait for everything to complete.
Optimistic Visualization

Trust but Verify

This glass is half full
What if we think of the issues with approximation as user experience problems?
Optimistic Visualization

1. Analysts uses initial estimates.
2. Precise queries run in the background.

Analysts can use approximations and also trust them.
Pangloss Implements Optimistic Visualization

Heatmap

X-Axis
- Field: DepDelay
- Binning: 64
- Sort by key: 

Y-Axis
- Field: ArrDelay
- Binning: 40
- Sort by key: 

Value
- Function: Count

Persistent Filters
- e.g., "Carrier $d$[a, d] DepDelay > 8"

Expected Error
- Relative

Approximate Values
- DepDelay
- ArrDelay

Expected Errors: 2.3%
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.