MIE1512 Data Analytics

Data Exploration and Visualization

Assigned Reading

SeeDB: A Visualization Recommendation Tool

Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya Parameswaran, Neoklis Polyzotis

SEEDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics,

VLDB2016

http://www.vldb.org/pvldb/vol8/p2182-vartak.pdf

UofT

Assigned Reading

Tableau Data Viz Software (www.tableau.com)

Chris Stolte, Pat Hanrahan

Polaris: A System for Query, Analysis and Visualization of Multi-Dimensional Relational Databases,

INFOVIS 2000

https://dl.acm.org/citation.cfm?id=857686

UofT

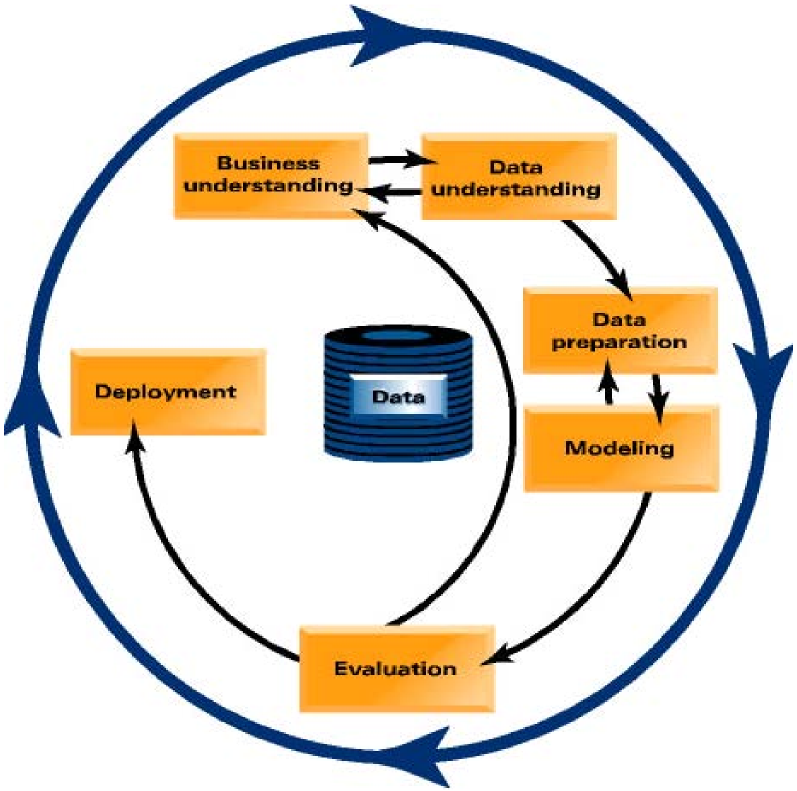
From Pat Hanrahan

My Process

- Pose the question
  - Find or collect the appropriate data
    - Check and verify
      - Clean and normalize
        - Contextualize the data by joining with other data
          - Explore relationships & patterns in the raw data
            - Generalize and summarize
              - Confirm hypotheses and analyze errors
                - Share findings with others
                  - Decide and act

CRISP-DM

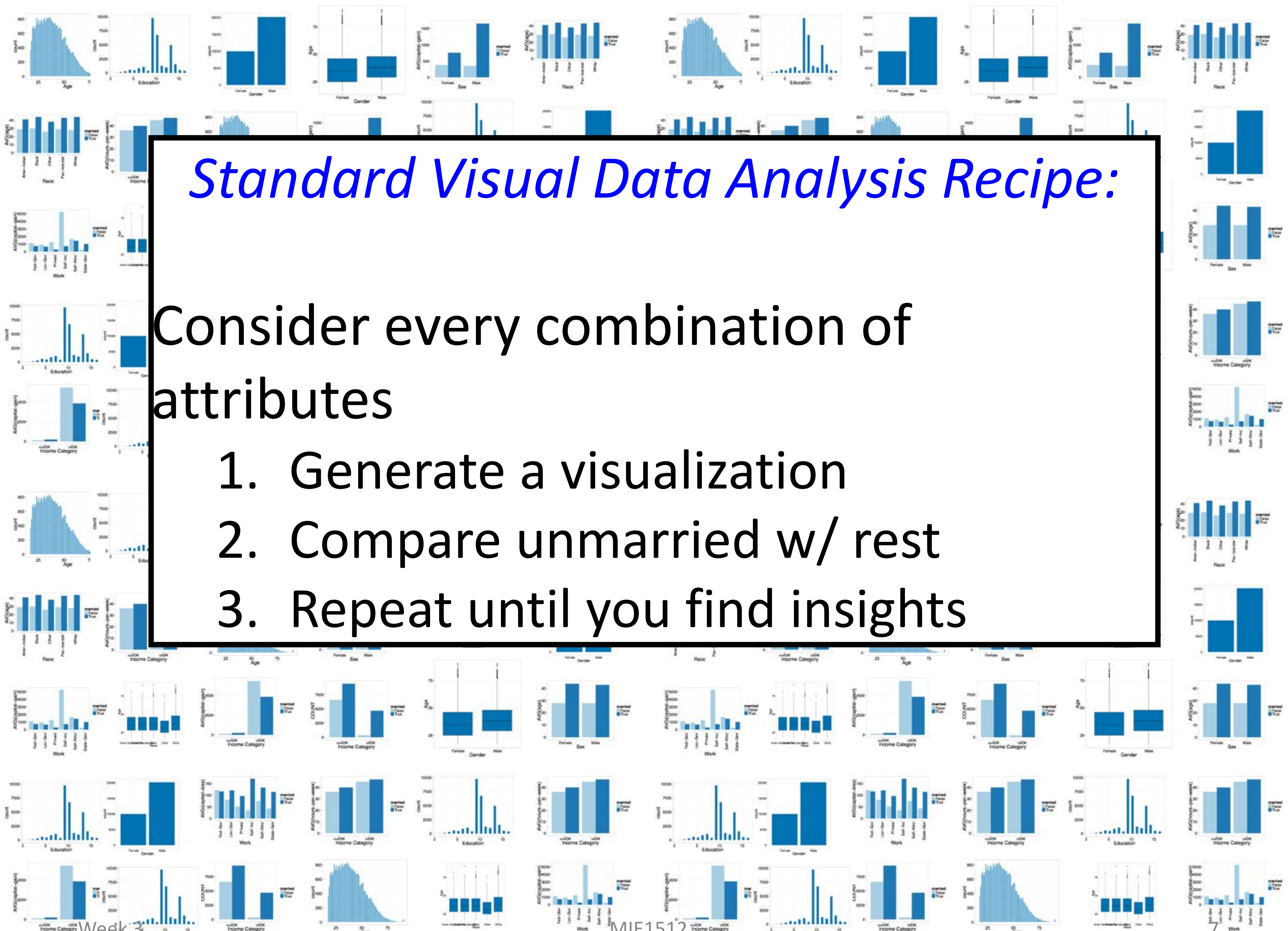
CRoss-Industry Standard Process for Data Mining



CRISP-DM

Phases and Tasks

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives	Collect Initial Data	Select Data	Select Modeling Technique	Evaluate Results	Plan Deployment
Assess Situation	Describe Data	Clean Data	Generate Test Design	Review Process	Plan Monitering & Maintenance
Determine Data Mining Goals	Explore Data	Construct Data	Build Model	Determine Next Steps	Produce Final Report
Produce Project Plan	Verify Data Quality	Integrate Data	Assess Model		Review Project
		Format Data			

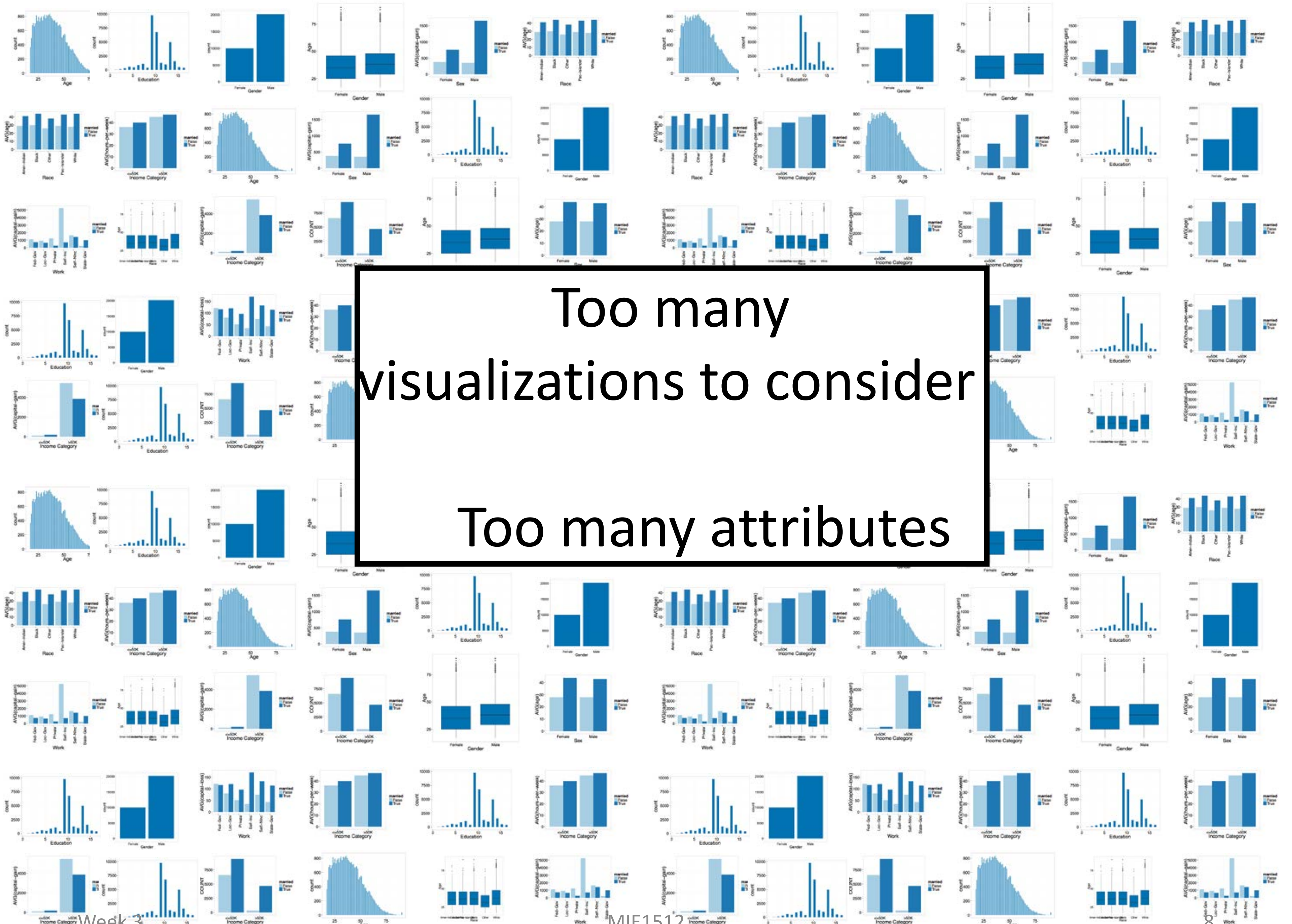


Standard Visual Data Analysis Recipe:

Consider every combination of attributes

1. Generate a visualization
2. Compare unmarried w/ rest
3. Repeat until you find insights

UofT



Too many visualizations to consider

Too many attributes

Outline

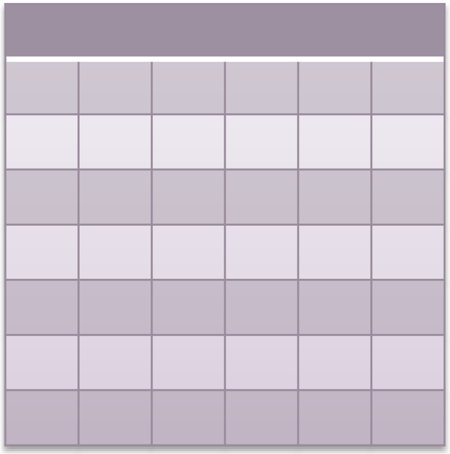
- Scalable Visualization Recommendations
  - Space of visualizations considered by SeeDB
  - Building SeeDB
  - Evaluating SeeDB

Related Work

- Visualization tools:
  - ManyEyes, Tableau/Polaris, Fusion Tables, Spotfire
  - Tableau and Spotfire recommendations (Aesthetics)
- Some Automation: VizDeck, Profiler, Voyager

Space of Visualizations

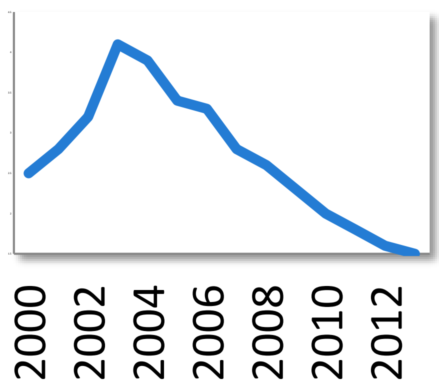
For simplicity, assume a single table
(star schema)

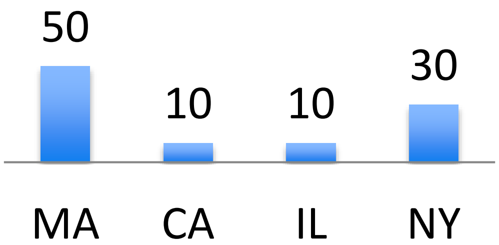


Visualizations = agg. + grp. by queries

Vi = SELECT d, f(m)
FROM table
WHERE ___
GROUP BY d

(d, m, f):
dimension, measure, aggregate





Space of Visualizations

```
Vi = SELECT d, f(m)
FROM table
WHERE ___
GROUP BY d
```

(d, m, f):

dimension, measure, aggregate

{d} : race, work-type, sex etc.

{m} : capital-gain, capital-loss, hours-per-week

{f} : COUNT, SUM, AVG

Building SeeDB: Questions

I. *Interestingness:* How do we determine if a visualization is interesting?

– Utility Metric

II. *Scale:* How to make recommendations efficiently and interactively?

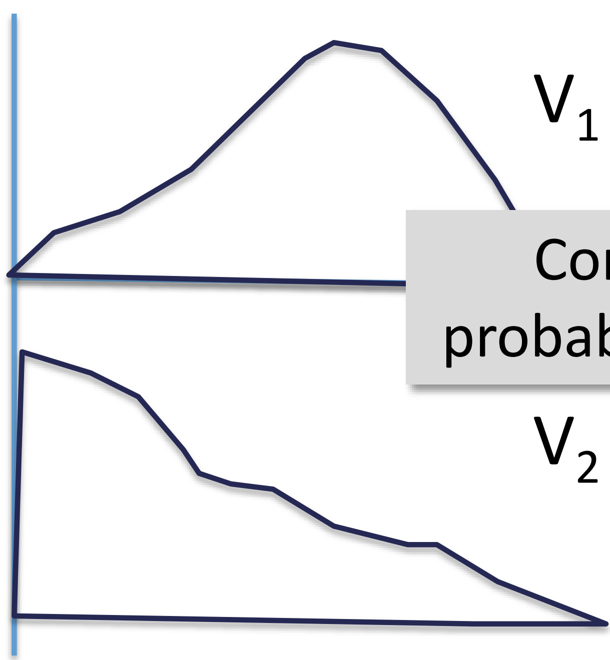
– Optimizations

Deviation-based Utility Metric

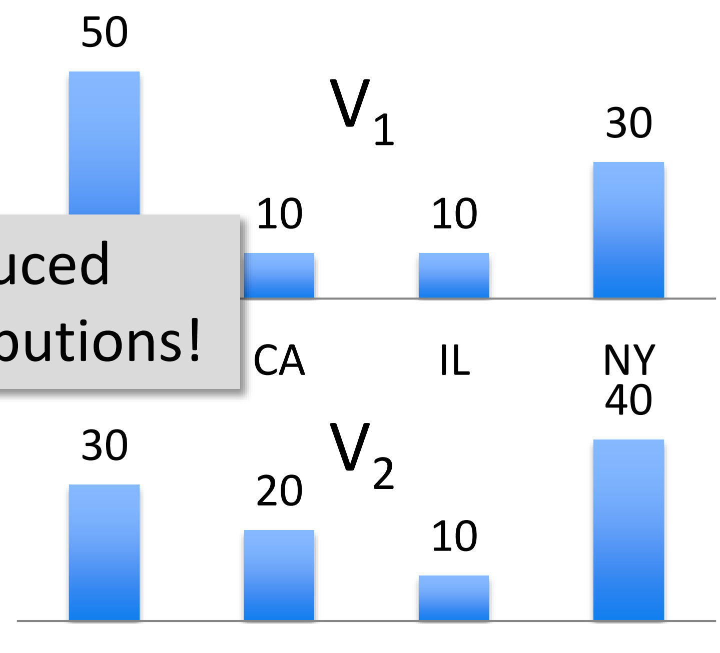
A visualization is interesting if it displays
a large deviation from some reference

Target　　　　　　　　　Reference
Task: compare unmarried adults with all adults

V1 = SELECT d, f(m) FROM table WHERE target GROUP BY d
V2 = SELECT d, f(m) FROM table WHERE reference GROUP BY d



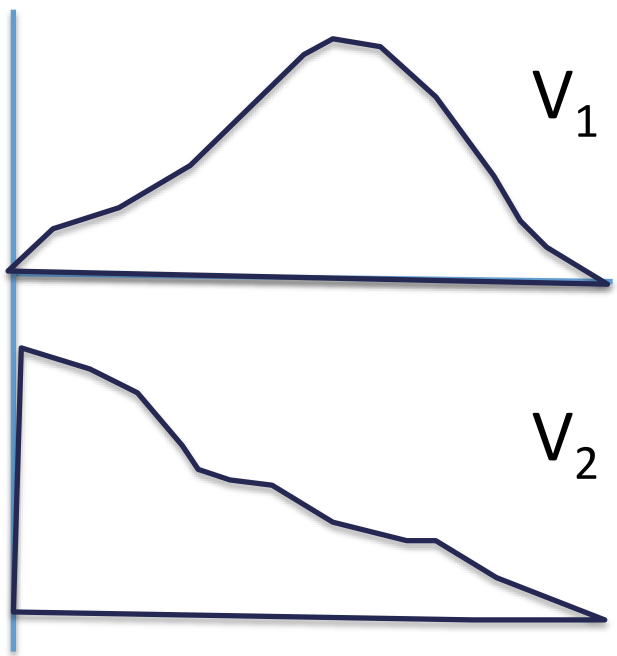
Compare induced probability distributions!



Deviation-based Utility Metric

A visualization is interesting if it displays
a large deviation from some reference

Many metrics for computing distance between distributions



D [P(V1), P(V2)]

Earth mover's distance
L1, L2 distance
K-L divergence

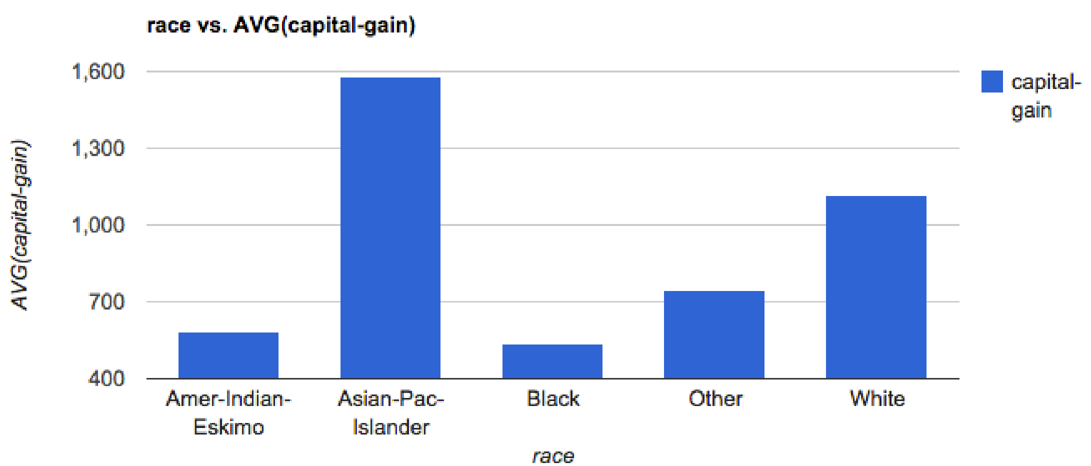
Any distance metric b/n distributions is OK!

Computing Expected Trend

Race vs. AVG(capital-gain)

Reference Trend

SELECT race, AVG(capital-gain) FROM census GROUP BY race



$P(V_1)$

Expected

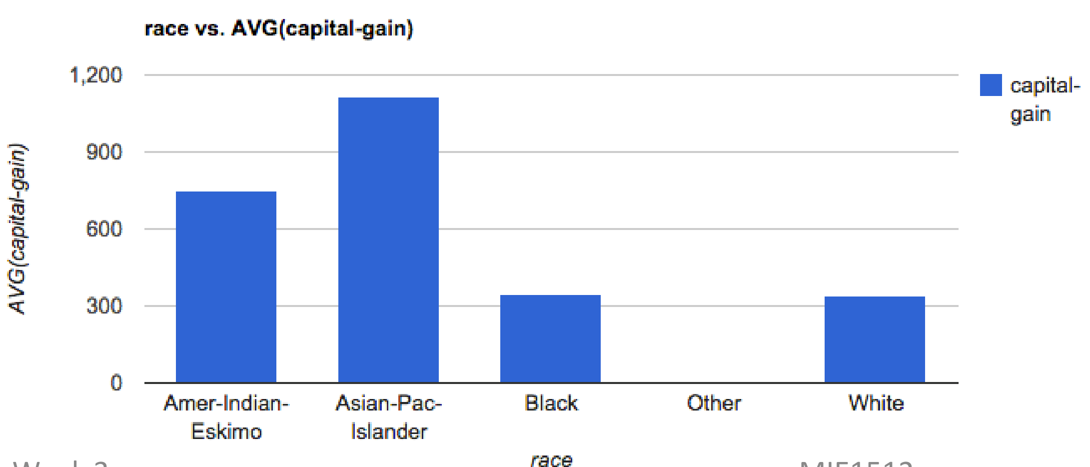
Distribution

Computing Actual Trend

Race vs. AVG(capital-gain)

Target Trend

SELECT race, AVG(capital-gain) FROM census GROUP BY race WHERE marital-status='unmarried'

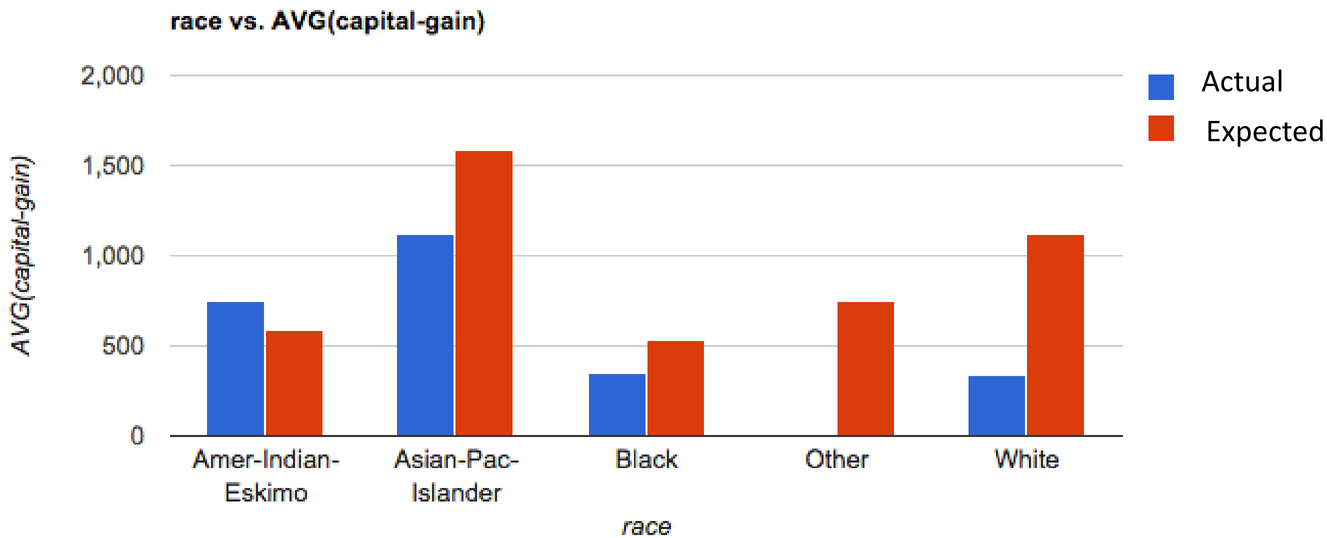


$P(V_2)$

Actual Distribution

UofT

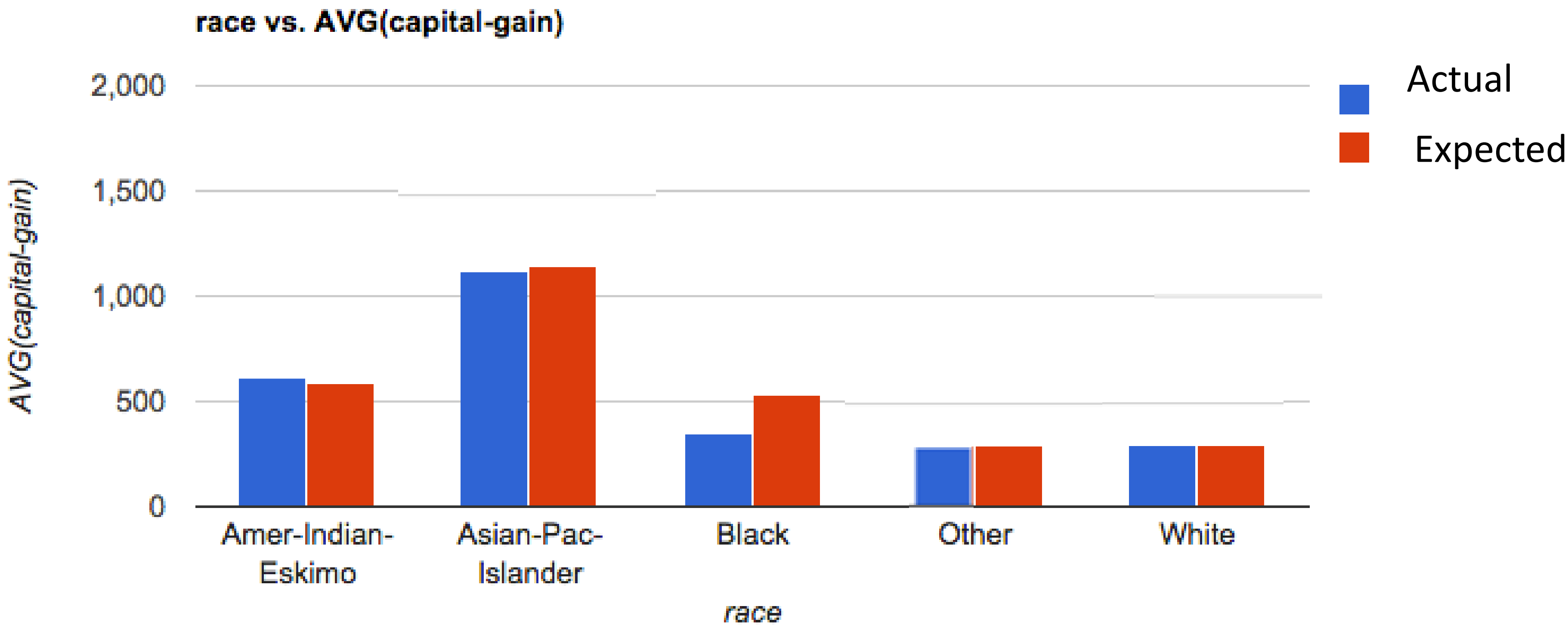
Computing Utility



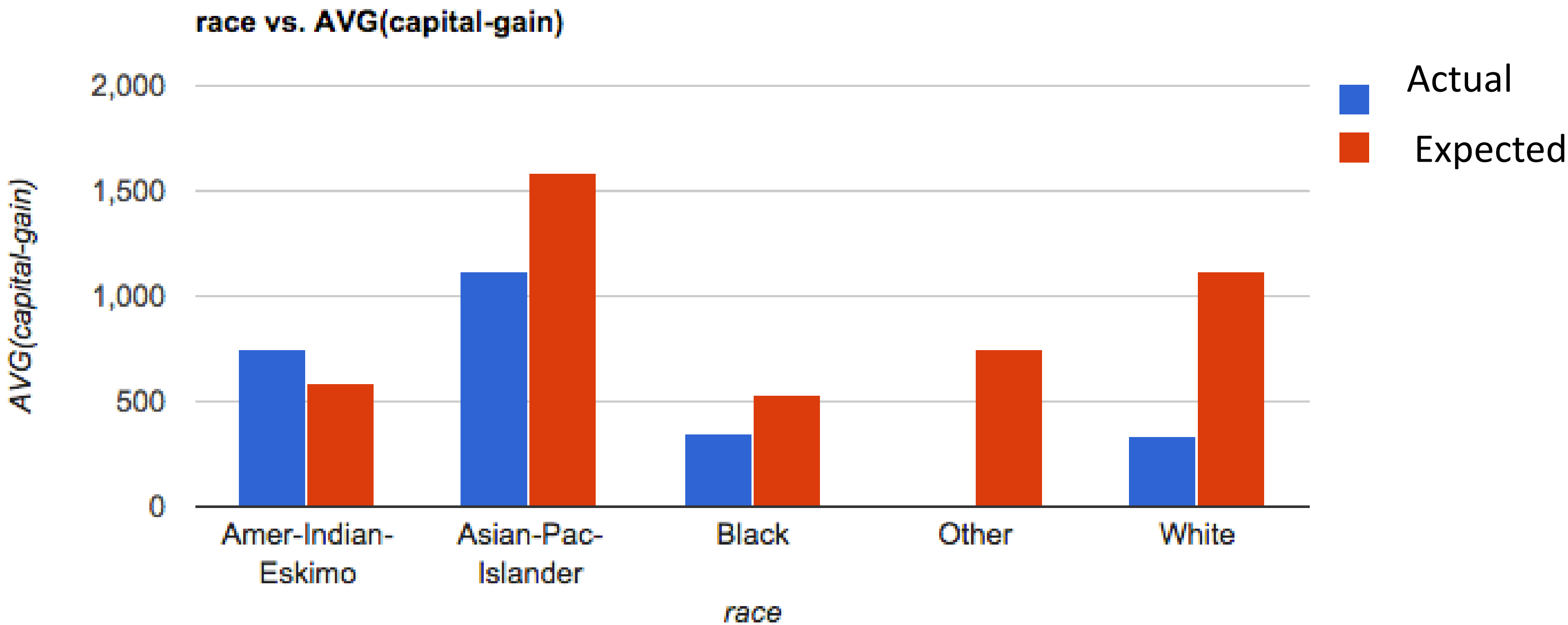
$$\mathbf{U = D[P(\mathit{V_1})\ ,\ P(\mathit{V_2})]}$$

D = EMD, L2 etc.

Low Utility Visualization



High Utility Visualization



What else?

SeeDB opts for deviation from a reference as a utility metric. What else could we use?

UofT

Utility Metric: 5 axes

- Data characteristics
- Task or Insight
- Semantics and Domain Knowledge
- Visual Ease of Understanding
- User Preference

Maximize Surprise (via deviation)

Current Work

Not the current focus

UofT

Why not?

- When would a deviation-based metric not make sense?

UofT

Problem Statement

Across all (d, m, f), where

V1 = SELECT d, f(m) FROM table WHERE target GROUP BY d

V2 = SELECT d, f(m) FROM table WHERE reference GROUP BY d

Goal: return *k* best utility visualizations (d, m, f), (those with largest D[V1, V2])

Vi = (d: dimension, m: measure, f: aggregate)

10s of dimensions, 10s of measures, handful of aggregates

2* d * m * f

➔ **100s of queries for a single user task!**

UofT

Problem Statement

Across all (d, m, f), where

V1 = SELECT d, f(m) FROM table WHERE target GROUP BY d

V2 = SELECT d, f(m) FROM table WHERE reference GROUP BY d

Goal: return *k* best utility visualizations (d, m, f),
(those with largest D[V1, V2])

Vi = (d: dimension, m: measure, f: aggregate)

10s of dimensions, 10s of measures, handful of aggregates

2* d * m * f

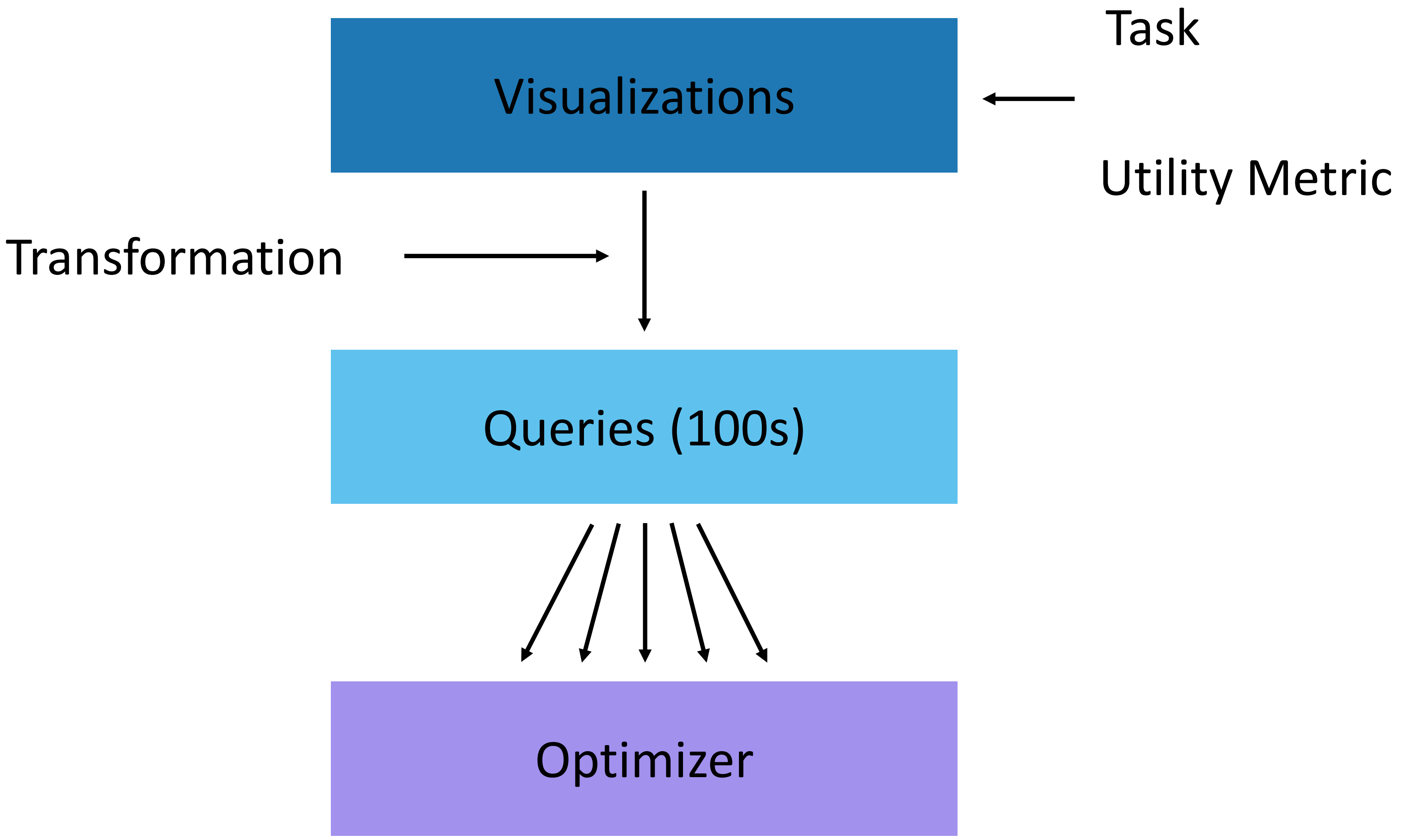
➔**100s of queries for a single user task!**

➔**Can be even larger. How?**

Even larger space of queries

- Binning
- 3 dimensional or 4 dimensional visualizations
- Scatterplot or map visualizations
- …

For simplicity, let's stick to the current set …


Task
Visualizations
Utility Metric
Transformation
Queries (100s)
Optimizer

Naïve Approach

For each (d, m, f) in sequence

  evaluate queries for V1 (target), V2 (reference)

  compute D[V1, V2]

Return the k (d, m, f) with largest D values

Too long!!

UofT

Issues w/ Naïve Approach

- Repeated processing of same data in sequence across queries

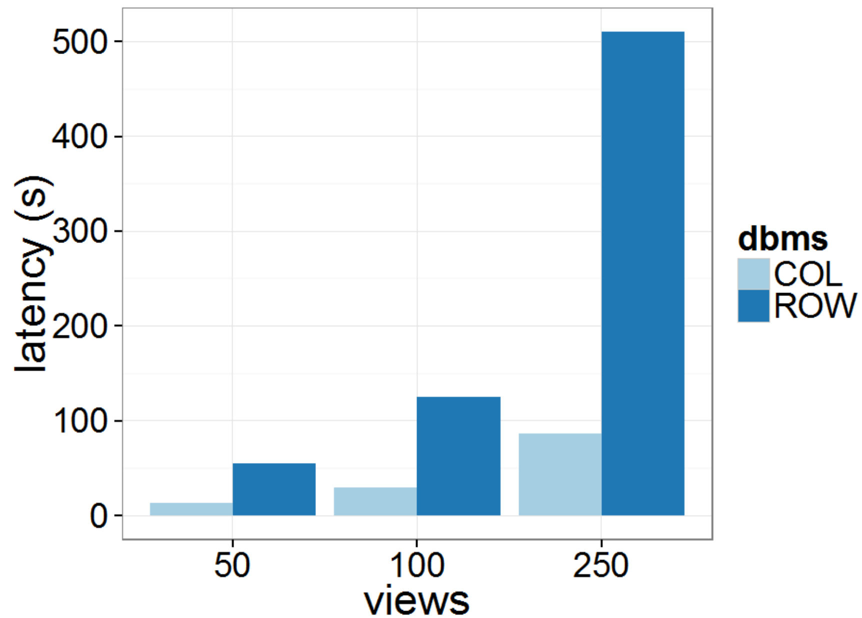
Sharing

- Computation wasted on low-utility visualizations

Pruning

Systems-level optimizations

- Each visualization = 2 SQL queries



- Latency > 100s
- Minimize number of queries and scans

Systems-level optimizations

- Combine aggregate queries on target and ref
- Combine multiple aggregates
  (d1, m1, f1), (d1, m2, f1) → (d1, [m1, m2], f1)
- Combine multiple group-bys*
  (d1, m1, f1), (d2, m1, f1) → ([d1, d2], m1, f1)
  Could be problematic…
- Parallel Query Execution

Combining Multiple Group-bys

- Too few group-bys leads to many table scans
- Too many group-bys hurt performance
  - # groups = Π (# distinct values per attributes)
- Optimal group-by combination ≈ bin-packing
  - Bin volume = log S (max number of groups)
  - Volume of items (attributes) = log ($|a_i|$)
  - Minimize # bins s.t.

$$\Sigma_i \log(|a_i|) <= \log S$$