

Fast-forwarding to Desired Visualizations with

Aditya Parameswaran
Assistant Professor
University of Illinois

<http://data-people.cs.illinois.edu>



With: *Tarique Siddiqui, John Lee, Albert Kim, Ed Xue, Chao Wang, Sean Zou, Changfeng Liu, Lijin Guo, Xiaofu Yu, and Karrie Karahalios*



NIH Big Data to Knowledge (BD2K)



The Democratization of Data Science: The Emergence of Data Visualization Tools



Data Visualization Tools



- Billions in revenue
- Huge audience
- Interactions not code

Data Visualization *is* Data Science for the 99%!

However, these tools are SERIOUSLY limited in their power...

Deriving insights is laborious and time-consuming!

↑ errors ↑ frustration ↑ wasted time ↓ insights ↓ exploration

Standard Data Visualization Recipe:

1. Load dataset into data viz tool
2. Start with a desired hypothesis/pattern
3. Select viz to be generated
4. See if it matches desired pattern
5. Repeat 3-4 until you find a match

Laborious and Time-consuming!

Key Issue:

- visualizations can be generated by
- varying subsets of data, and
- varying attributes being visualized

Too many visualizations to look at to find desired visual patterns!

Broadly Applicable



Carnegie Mellon University
Scott Institute
for Energy Innovation



- find keywords with similar CTRs to a specific one
- find solvents with desired properties
- find aspects on which two sets of genes differ
- find sensors with anomalous behavior

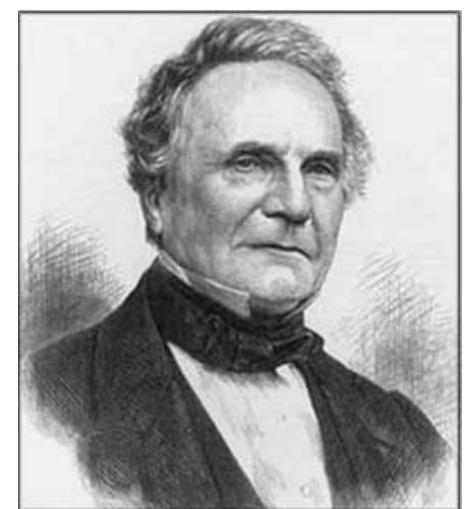
Common theme: **manual labor** for finding desired patterns to test hypotheses, derive insights

Lessons from History: Use Automation!

"Astronomers surely will not have to continue to exercise the patience which is required for computation. It is this that deters them from ... working on hypotheses and from discussion of observations... For it is **unworthy of excellent men to lose hours like slaves in the labor of calculation** data which could be safely relegated (to) machines." visualization
[Gottfried Leibniz, 1700s]



"... **intolerable labor and fatiguing monotony of a continued repetition of similar calculations** visualization representing the lowest occupation of human intellect"
[Charles Babbage, 1800s]



Source: "The Information" by James Gleick, highly recommended!

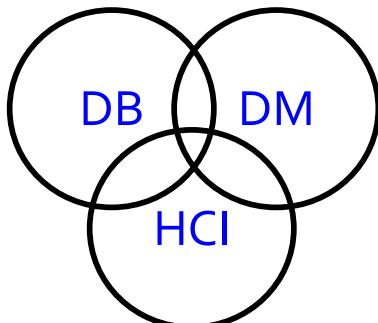
Key Insight : Automation

We can automate that!

Desiderata for automation:

- Expressive – specify what you want
- Interactive – interact with results, cater to non-programmers
- Scalable – get interesting results quickly

Drawing from



Enter Zenvisage:
(zen + envisage: to effortlessly visualize)



Overview

ZenVisage

Dataset

Real Estate

Category

city
metro
county
state

X-axis

month
year
quarter

Y-axis

soldpricepersqft
listingpricepersqft
pctdecreasing
foreclosuresratio
pctincreasing
listingprice
soldprice
pricetorentratio
pctforeclosed
saletolistratio
pctpricereductions
numberforrent
turnover

ZQL Table

Similarity

- Euclidean Distance
- Segmentation
- DTW
- MVIP

Aggregation Method

- Sum
- Average

Number of Results

K-means Cluster Size

Input equation add

Options

Consider x-range

Show scatterplot

Results

city: The Village (0.84)

city: Amherst (0.81)

city: Dubuque (0.81)

city: Edmond (0.80)

city: McCandless Township (0.79)

city: Eau Claire (0.79)

Representative patterns

Dunedin (1382)

Raleigh (346)

North Miami (49)

Outliers

Temperance

Elizabeth

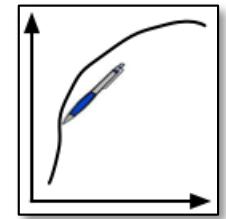
Rock Island

9

Zenvisage: Two Modes

- **First Mode:** Interactions, drawing, drag-and-drop

- Simple needs
- Starting point / context



- **Second Mode:** the Zenvisage Query Language (ZQL)

- Sophisticated needs
- Multiple steps

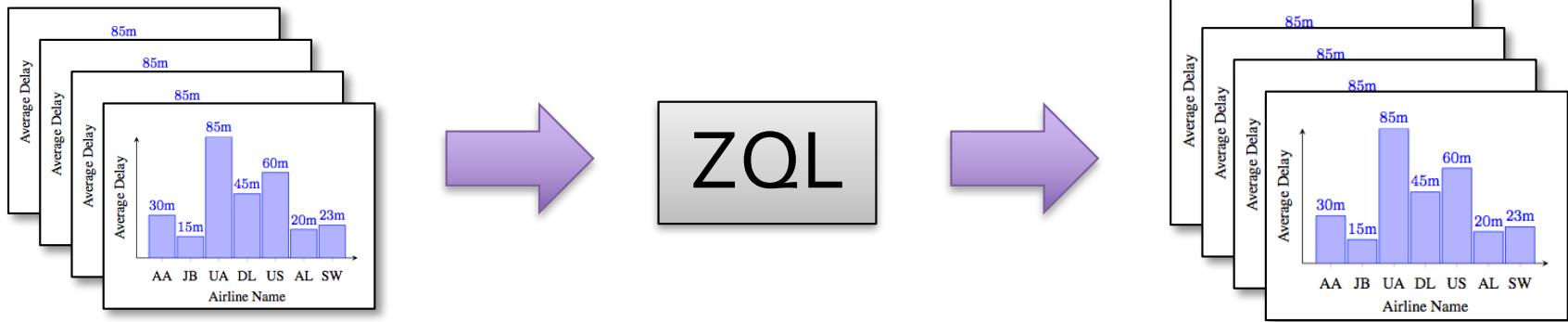
X	Y	Z	Constraints	Process
<input type="text"/> 				

Can switch back and forth, as user needs evolve

Both modes developed after many discussions with potential users

ZQL: High Level Overview

ZQL is a viz exploration language



- Inspired from QBE & VizQL / Grammar of Graphics
- Captures four key operations on viz collections

Compose

Filter

Compare

Sort

- Incorporates **data mining primitives**
- Powerful; formally demonstrated “completeness”

ZQL: A Bird's Eye View

Name X Y Z Constraints Process

Name	X	Y	Z	Constraints	Process
<input type="text" value="*f1"/>	<input type="text" value="'quarter'"/>	<input type="text" value="'soldprice'"/>	<input type="text" value="'metro'. 'Peoria'"/>	<input type="text"/>	<input type="text"/>
<p style="text-align: center;">+</p> <p style="text-align: center;">Submit</p>					

*Output spec
and identifiers*

*Composition of visualizations, often using
values from previous steps*

*Sorting, comparing, and
filtering visualizations*

*f1

'quarter'

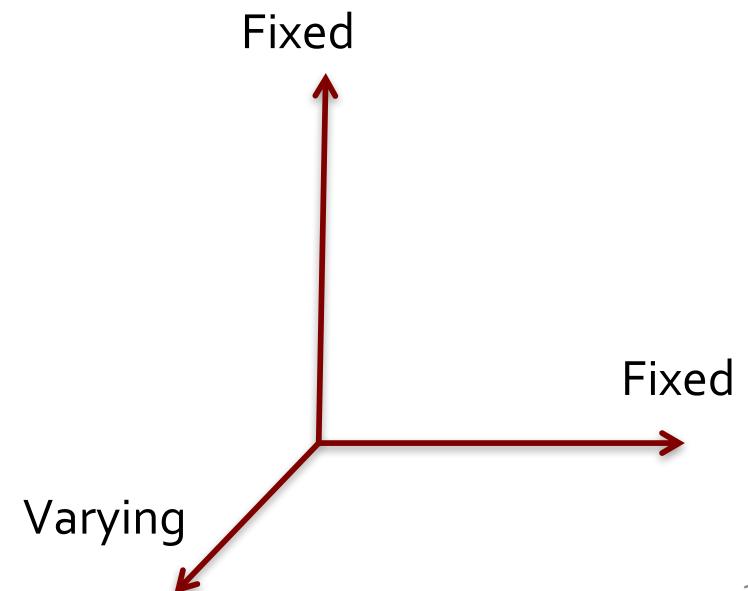
'soldprice'

'metro'. 'Peoria'

Example 1: Comparisons

Find the states where the *soldprice* trend is most similar to (or most different from) the *soldpricepersqft* trend.

→ *Comparing a pair of y-axes for different “z”*



Example 1: Comparisons

ZQL Table

Q1	Q2	Q3	Q4	Q5	Q6	Clear
----	----	----	----	----	----	-------

Name	X	Y	Z	Constraints	Process
- f1	x1<-'year'	y1<-'soldprice'	z1<-'state'.*		
- f2	x1	y2<-'soldpriceper:	z1		v1<-argmin_{z1}[k=3]DEuclidean(f1,f2)
- *f3	x1	y3<-'soldprice', 'sc	v1		

+ Submit

Results

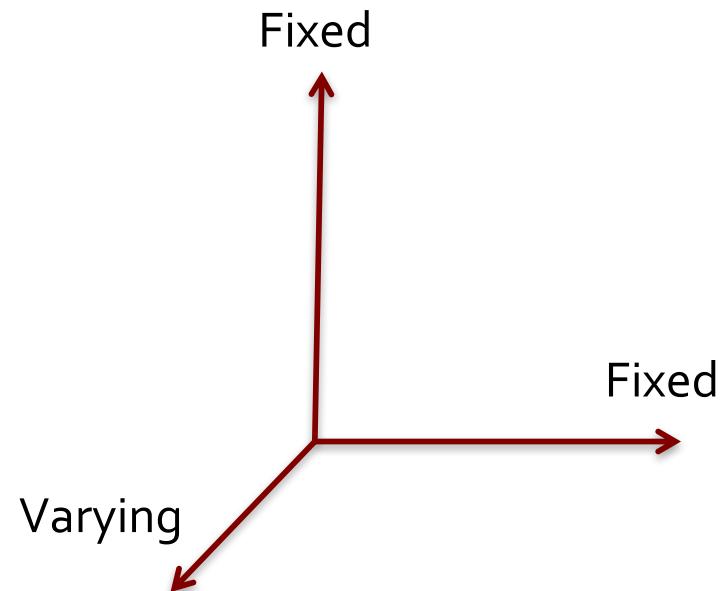
Representative patterns ⓘ

Outliers ⓘ

Example 2: Drill-downs

Find *cities in NY* where the trend for *soldprice* is most different from (or most similar to) the *overall NY trend*.

→ *Comparing across different granularities of “z”*



Example 2: Drill-downs

ZQL Table

Q1	Q2	Q3	Q4	Q5	Q6	Clear
----	----	----	----	----	----	-------

Name	X	Y	Z	Constraints	Process	
-	f1	x1<-'year'	y1<-'soldprice'	z1<-'state'.*	state='NY'	
-	f2	x1	y1	z2<-'city'.*	state='NY'	v2<-argmin_{z2}[k=3]DEuclidean(f1,f2)
-	*f3	x1	y1	v2		
+ Submit						

Results

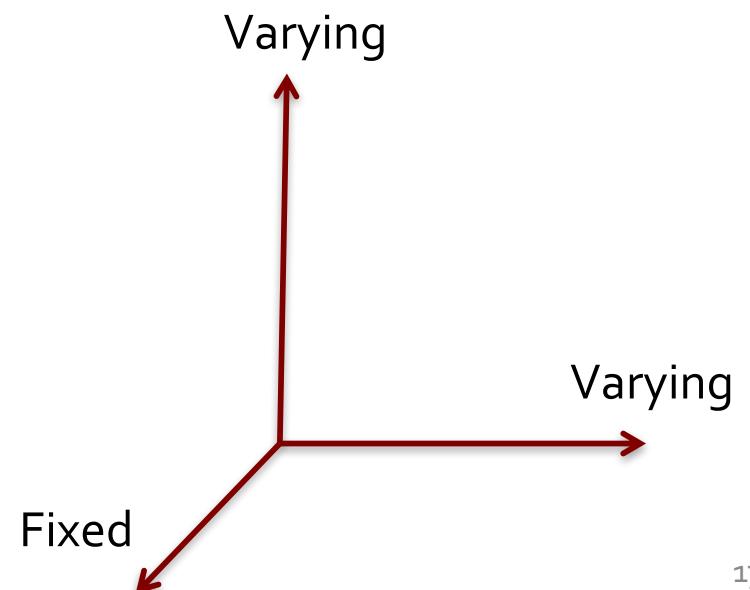
Representative patterns ⓘ

Outliers ⓘ

Example 3: Explanations/Diffs

Find visualizations on which the *states of CA* and *NY* are most different (or most similar).

→ *Comparing across different "x", "y" for two "z"*



Example 3: Explanations/Diffs

ZQL Table

Q1	Q2	Q3	Q4	Q5	Q6	Clear
----	----	----	----	----	----	-------

Name	X	Y	Z	Constraints	Process
- f1	x1<-*	y1<-*	'state'.'CA'		
- f2	x1	y1	'state'.'NY'		x2,y2<-argmin_{x1,y1}[k=1]DEuclidean(f1,f2)
- *f3	x2	y2	'state'.'CA'		

Submit

Results

Representative patterns ⓘ

Outliers ⓘ

ZQL Query Execution

Let's use a relational database as a backend

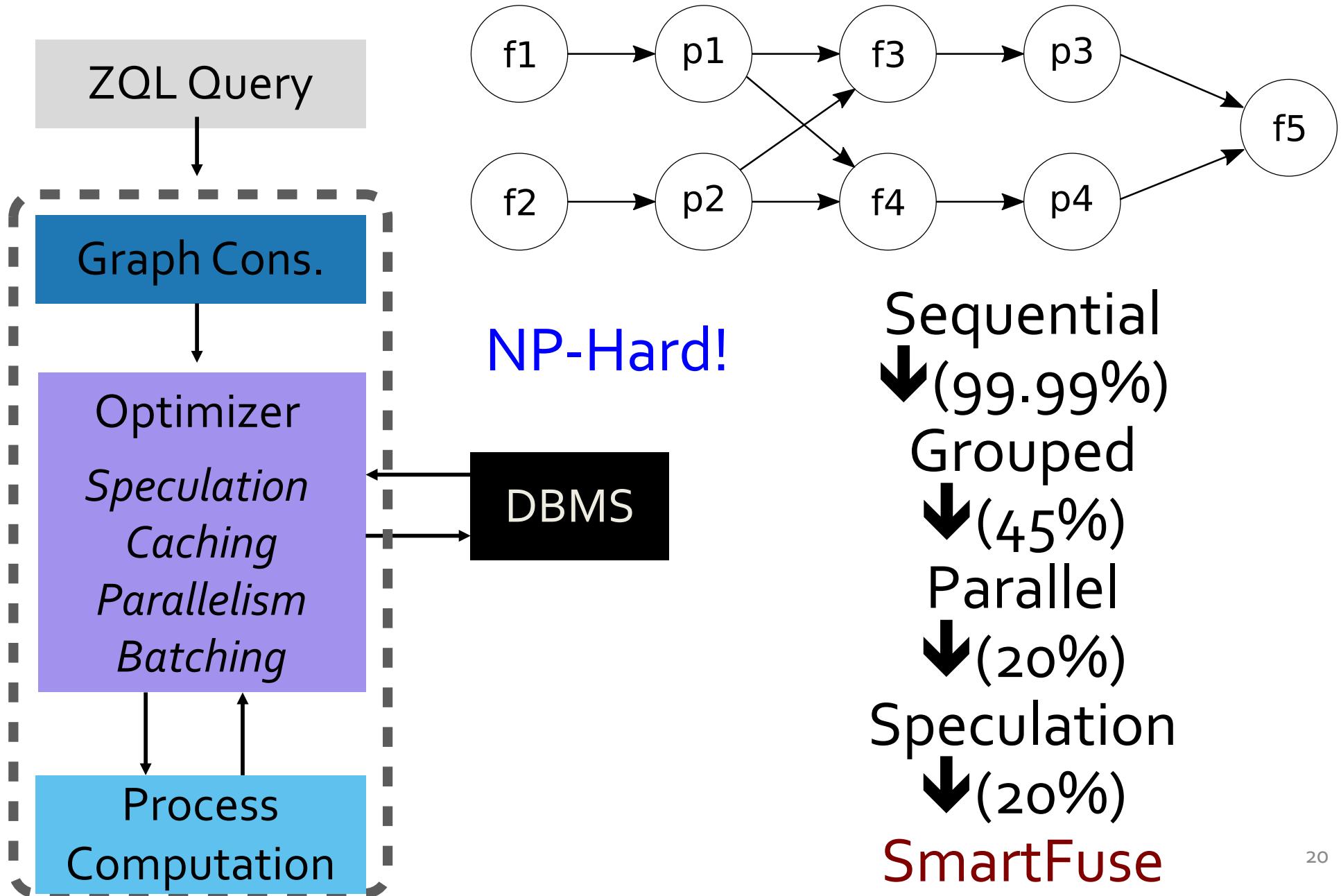
Naïve translation approach:

For each line of ZQL:

Issue one SQL query for each combination of X, Y, Z;
Apply further processing on result

Often 1000s of SQL queries issued per ZQL query!
→ *wasteful, extremely high latency*

SmartFuse: Intelligent Query Optimizer



User Study Takeaways (20 Participants)

Faster $\mu = 115\text{s}$, $\sigma = 51.6$ vs. $\mu = 172.5\text{s}$, $\sigma = 50.5$

More accurate $\mu = 96.3\%$, $\sigma = 5.82$ vs. $\mu = 69.9\%$, $\sigma = 13.3$

"In Tableau, there is no pattern searching. If I see some pattern in Tableau, such as a decreasing pattern, and I want to see if any other variable is decreasing in that month, I have to go one by one to find this trend. But here I can find this through the query table."

"you can just [edit] and draw to find out similar patterns. You'll need to do a lot more through Matlab to do the same thing."

"The obvious good thing is that you can do complicated queries, and you don't have to write SQL queries... I can imagine a non-cs student [doing] this."

Effortless Visual Exploration of Large Datasets with



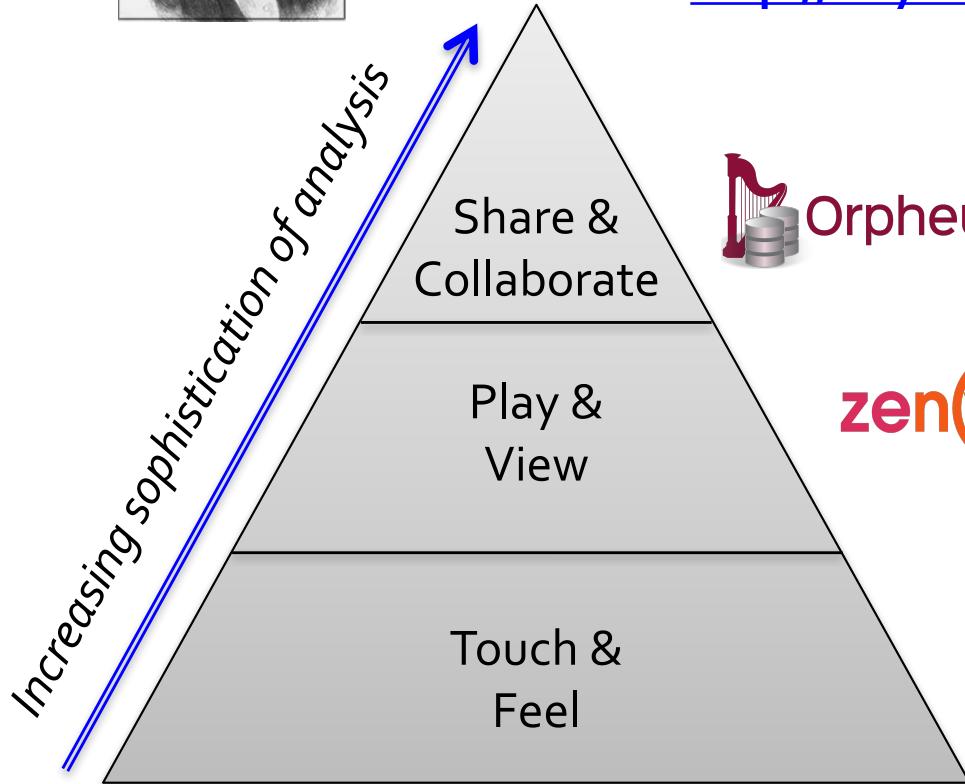
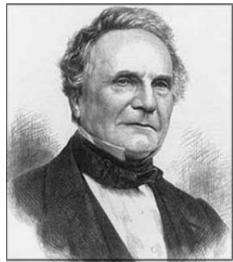
Ingredients

- *Drag-and-drop & sketch interactions*
- *Sophisticated visual expl. language, ZQL*
- *ZQL optimization engine: SmartFuse*
- *Perceptually-aware pattern matching algorithms*

Many other challenges that we have overcome...

Detailed demo – talk to us (Tarique, Ed, me) afterwards!

Broad Agenda: Human-in-the-loop Data Analysis Tools for the 99%



<http://tiny.cc/three-tools>



orpheus-db.github.io

zenvisage.github.io

dataspread.github.io

Please consider using or contributing!

<http://data-people.cs.illinois.edu>; adityagp@twitter

Touch and Feel:



DataSpread is a **spreadsheet-database hybrid**:

Goal: Marrying the flexibility and ease of use of spreadsheets with the scalability and power of databases

Enables the “99%” with large datasets but limited prog. skills to open, touch, and examine their datasets

<http://dataspread.github.io>

[VLDB'15, VLDB'15, ICDE'16]

Collaborate and Share: OrpheusDB

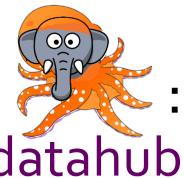
OrpheusDB is a tool for **managing dataset versions** with a database

Goal: building a versioned database system to reduce the burden of recording datasets in various stages of analysis

Enables individuals to collaborate on data analysis, and share, keep track of, and retrieve dataset versions.

<http://orpheus-db.github.io>

[VLDB'16, VLDB'15, VLDB'15, TAPP'15, CIDR'15]

(also part of  : a collab. analysis system w/ MIT & UMD)
datahub