

Effortless Data Exploration with **zenvisage** : An Expressive and Interactive Visual Analytics System



Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, Aditya Parameswaran

zenvisage.github.io



Motivation

Everyone doing exploratory data analysis uses some combination of the following workflow:

1. Load dataset into an interactive viz tool like Excel or Tableau
2. Select visualization to be generated
3. See if the visualization satisfies desired “insights” or “visual property”
4. If yes, stop; if not, back to step 2

With LARGE datasets and LARGE # of attributes, this is a tedious and time consuming process, all for a single visual property.

Motivation

This is a **real** problem!

- **Advertising Data Analysis:** (our collaborators at Turn Inc.)
 - a. Finding keywords with similar click-through rates
- **Genomic Data Analysis:** (our collaborators at the NIH center at Illinois)
 - a. Finding pairs of genes that can visually explain the difference between clinical outcomes
- **Environmental Data Analysis:** (our collaborators at the Great Lakes initiative)
 - a. Finding sensors (on buoys) that are behaving anomalously
- **Engineering Data Analysis:** (our collaborators at CMU)
 - a. Finding solvents with desired behaviours (“hockey stick” shape for a certain property)
- ...

Common theme: There are multiple settings where finding the “right” visualization that reveals the desired insight can take **hours or days!**

Enter zenvisage



Zenvisage = **zen** + **envisage** (to “effortlessly” visualize)

A visual data exploration system for *“fast-forwarding to desired insights”*

We’ve been building the system for the last 2 years; being developed in collaboration with the 4 collaborator groups

A significant generalization of the previous system SeeDB

Zenvisage: our design goals



Expressive: Specify desired insights using a declarative “data exploration” language for experts

Interactive: For non-experts, support simple interaction primitives to support effortless data exploration

Scalable: Must be able to traverse through a large space of visualizations and recommend interesting ones instantly

Expressiveness via ZQL

We've developed a data exploration language called [ZQL \(Zenvisage Query Language\)](#) enabling users to specify the desired visual insights

Using a small number of ZQL lines (**often < 2 lines**), users can specify desired trends, patterns, insights from visualizations

ZQL draws from QBE (Query By Example)/SQL + ggplot/polaris algebra

We've formally developed a [visual exploration algebra](#), and shown that ZQL is **visual exploration complete** with respect to that algebra
⇒ ZQL has *nice, formal semantics!*

Expressiveness via ZQL

All within two-three lines:

- Find x and y attributes on which chairs and desks differ the most
- Find products whose sales over years and the profit over years trends are most dissimilar
- From among products that are similar to staplers on sales over time, find typical trends on profits over time
- Find products whose sales over time has an increasing trend while profit over time has a decreasing trend

Interactivity : The “Drag-and-Drop” Perspective

The screenshot displays a web-based data visualization tool interface. The browser address bar shows `localhost:8999`. The main interface is titled "TOOL A" and features a "Custom Query Builder" section with buttons for "Draw", "Modify", "Clear", and "Submit". A "Predicate =" input field is also present.

The interface is divided into several sections:

- Left Panel (TOOL A):** Contains filters for "Datasets" (Student, Income, Real Estate), "Category" (Metro, State, City, County), "X-axis" (Quarter, Month, Year), and "Y-axis" (SoldPrice, SaleToListRatio, PctPriceReductions, NumberForRent, ListingPricePerSqft, PctDecreasing, PriceToRentRatio, PctForeclosed, ListingPrice, Turnover, Foreclosures-Ratio, SoldPricePerSqft, PctIncreasing). The "Aggregation Method" is set to "Average".
- Main Chart:** A line chart showing the average sold price (avg(SoldPrice) K) over time (2004-2014). The price starts around 140K, peaks at approximately 350K in 2010, and then declines to about 140K by 2014.
- Existing Trends Panel:** Displays three smaller line charts for different locations:
 - 1 : Houston (Count=191)
 - 2 : Dakota (Count=173)
 - 3 : Hudson (Count=63)
- Bottom Section:** Contains two more line charts for locations:
 - 1 : Jessamine
 - 2 : Dawson
 - 3 : Richmond
 - 4 : Henry

Information icons (i) are visible in the bottom right corner of the main chart area and the bottom right corner of the bottom section.

Interactivity: The ZQL Perspective

localhost:8999

Apps ★ Bookmarks

Zenvisage

Hide Query Builder Draw Modify Clear

Predicate = Submit

ID	X	Y	Z	Constraints	Viz	Process	Delete?
0	Year	SoldPrice	v1=City	State NY	Pick Viz	v2=increasing f1=v1 K=10	X
1	Year	PctForeclosed	v1	Type or c Type or	Pick Viz	absent	X
2	Year	PctIncreasing	v2	Type or c Type or c Type or	Pick Viz	v=distance 1 2 OPT=MAX f1=v2 K=10	X
3	Pick X	PctForeclosed PctIncreasing	v3	Type or c Type or c Type or	Barchart	Enter function Enter parameters	X

Stop

Add Rows

The figure displays four bar charts arranged in a 2x2 grid. Each chart shows the average percentage of foreclosed houses (avg(pctForeclosed)) on the y-axis against the year (1.0 to 12.0) on the x-axis. The top-left chart is for 'Freeport' with 'PctForeclosed' on the y-axis (0.0 to 22). The top-right chart is for 'Freeport' with 'PctIncreasing' on the y-axis (0 to 100). The bottom-left chart is for 'Bethlehem' with 'PctForeclosed' on the y-axis (0.0 to 2.8). The bottom-right chart is for 'Bethlehem' with 'PctIncreasing' on the y-axis (0 to 100).

What else?

- Expressiveness & Interactivity -- briefly covered
- Scalability:
 - Automated query translation and execution for ZQL
 - Proposed General Query Optimization techniques, similar to MQO

Evaluation

- Performance of optimization techniques
- Usability and effectiveness via a user study (12 people w/ varying experience with data exploration and programming); findings:
 - **Fast and accurate:** up to 100% faster than baseline, 30% more accurate results
 - **Easy:** Even non-programmers could learn a subset of ZQL and use it within a short period
 - **Better:** Unanimously, everyone preferred zenvisage over the baseline and wanted to incorporate it into their current data analysis workflow

What else?

Project webpage: <http://zenvisage.github.io/>

Technical Report: <http://data-people.cs.illinois.edu/zenvisage.pdf>

20+ pages (!!!) paper, circa April 2016: LOTS MORE HERE!!!

under review at VLDB, also up on ArXiv

To be open-sourced soon (in a few weeks)!!

Email me: tsiddiq2@illinois.edu