Paper Reviewed (1)

- Chris Stolte, Diane Tang, Pat Hanrahan
  "*Query, Analysis, and Visualization of Hierarchically Structured Data Using Polaris*"

Overview

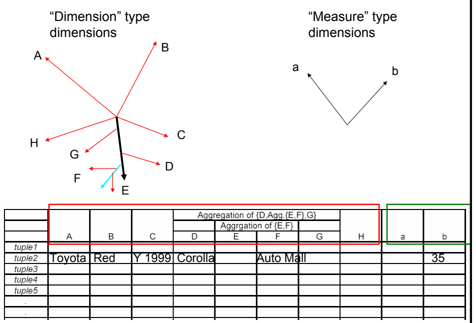
- Hierarchical Structure of Data
- Relational Databases VS. Data Cubes
- Nest Operand VS. Dot Operand
- New Interface in support of data cube
- Critiques

Hierarchical Structure of Data

- How to derive the Hierarchical Structure of Data
  - Known hierarchical structure (country, province,city)
  - Using data mining algorithm (decision trees, clustering technique)
- Benefit of hierarchical structure over relational structure
  - Flexible and efficient in obtaining data summaries of different aspects of data during data exploration process.
  - Support "semantic zooming" visualization
- Realization of organizing data into hierarchical structure
  - Concept of Data Cube

Relational Database VS Data Cubes

- Aspects of data dimensions
  - **Relational Database**: Dimensions are independent
  - **Data Cube**: Dimensions can be hierarchically dependent
- Aspect of data summary
  - **Relational Database:** Use SQL queries to retrieve
  - **Data Cube:** Aggregated values (summation, average, etc.) are readily stored in the cells of data cube

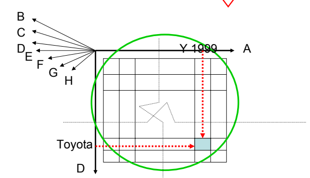


We might want to know the summation of values of dimension b where values corresponds to only dimension A and dimension D (*Ex: # of sales of used cars of different years + model*):

- Relational databases:

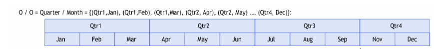
SELECT *A, D, sum (b)*
FROM *table*
GROUP BY *A,D*

- Data Cubes:



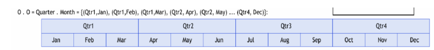
Nest Operand VS. Dot Operand

- Nest operand (no hierarchy implication)



The datasets do not have any data of October. So after nesting, we do not see Oct nested under Qtr4

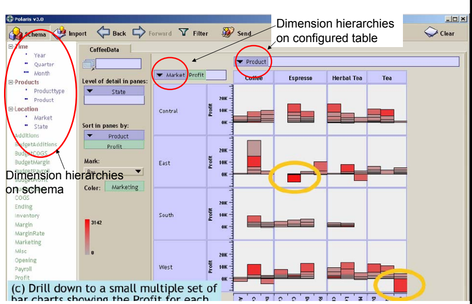
- Dot operand (hierarchy implication)

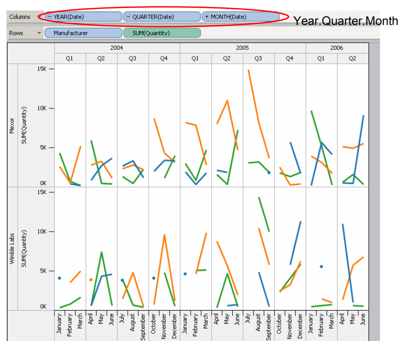


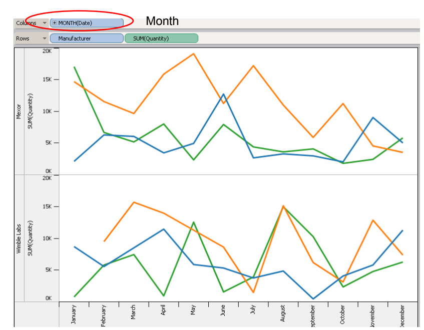
Semantically, Quarter and Month have hierarchy implications. So after doting, Oct is still displayed under Qtr4 even that there is no corresponding data

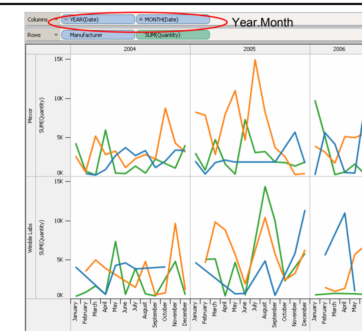
New Interface in support of data cube

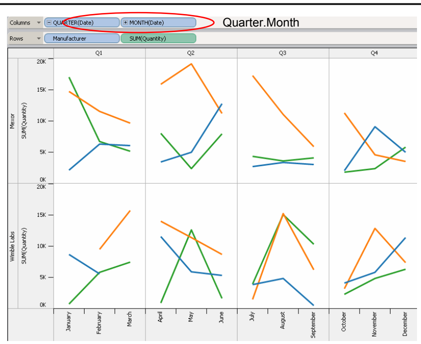
- Display dimensions hierarchies for more quickly configuring the table (determine the number of panes
  - On the schema
  - On the "shelves" of table
- Distinguish between "Node" and "Path
  - *Example: When selecting dimension "Month" from schema, Default is Year.Quarter.Month. But can change to "Month" or "Year.Month" or "Quarter.Month*
- Change level of detail within panes to reflect the change of dimension hierarchy (will change number of marks within panes as well)

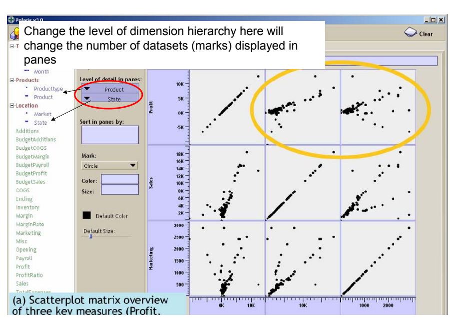












Critiques

- Pros
  - Provides interfaces for non-expert to retrieve data that involve complex data query algebra
  - Construct a robust formalism for presenting data cubes, which help reveal many aspects of data summary (different abstraction level of data and different detailed level of data)
  - Can also be an visualization tool for understanding the data mining model, which configure the hierarchical data structure.
- Cons
  - Did not use intuitive navigation techniques to facilitate changing views of data
  - Systems designed heavily focus on presenting summary of data. Could lead users only concentrate on this part of data analysis

Paper Reviewed (2)

- Chris Stolte, Diane Tang, Pat Hanrahan
  "Multi-scale visualization using data cubes"

Overviews

- Features Supported
  - Data abstraction and visual abstraction
  - Allow independently zooming along one or more dimensions
- Formalism guiding the Multi-scale visualization
  - Zoom graph
  - Polaris specification
- Proved effective design pattern
- Critique

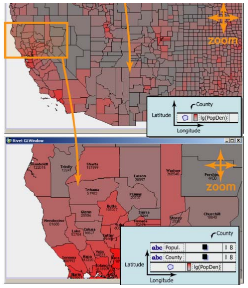
Data Abstraction

Table 3.a: Sales Roll Up by Model by Year by Color

Model	Year	Color	Sales by Model by Year by Color	Sales by Model by Year	Sales by Model
Chevy	1994	black	50		
		white	40		
				90	
	1995	black	85		
		white	115		
				200	
					290

Most detailed data: Sales by Model (M) and by Year (Y) and by Color (C)
Intermediate detailed data: Sales by M and Y or by C and Y or by M and C
Most abstract data: Sales by M or sales by Y or Sales by C

Visual Abstraction



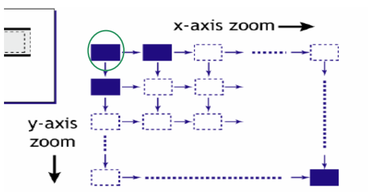
Abstract visual representation: Smaller area without texts to denote the County

Detailed visual representation: Lager area and texts to denote the County

Multiple Zoom Path

- Data sets are organized using multiple hierarchies (e.g.: some dimensions of data sets can be aggregated into different meaningful hierarchical level).
- So it is an advantage to be able to zoom in/out along those dimensions or combination of those dimensions.
- See later Example that zoom in X dimension and Y dimension independently.

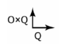
Zoom Graph



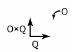
Nodes in the graph are the zoomed visualization, which can be described by Polaris specification.

Polaris Specification and its conventions

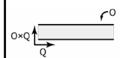
Table algebra :dot (.), cross (x), nest (/), and concatenate (+)



:Used to describe the table structure



:Used to describe any dimensions needed but not already encoded in the table structure

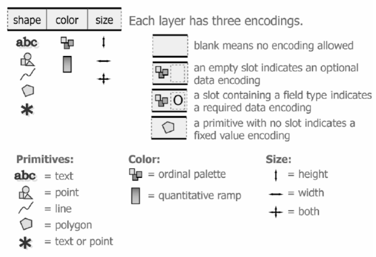


:Used to describe a layer in the visualization

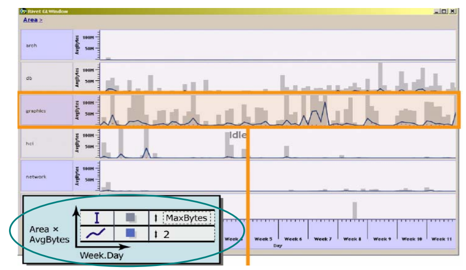
shape	color	size

:Each layer can have three types of visual encodings

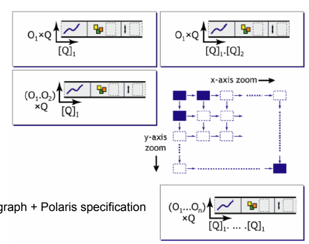
More on Polaris Encoding



Example: conventions of Polaris specification VS. visualization

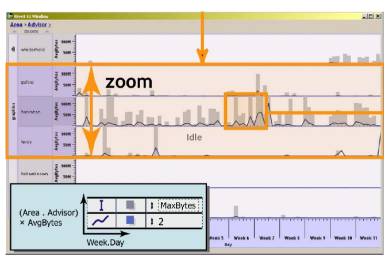


[Zoom graph]+[Polaris specification] VS. multi-scale visualization



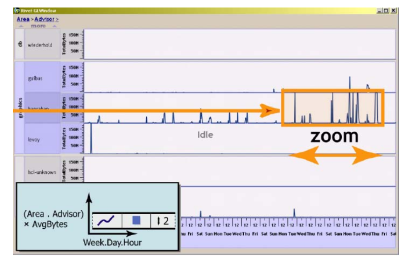
Zoom graph + Polaris specification

Y-axis (Dimension *User*) Zoom (previous example)



Dimension *User* has the hierarchical structure: *Area->Advisor->Project->Username*

X-axis (Dimension *Time*) Zoom (previous example)



Dimension *Time* has the hierarchical structure: *Week->Day->Hour->Minute*

Effective Design Pattern

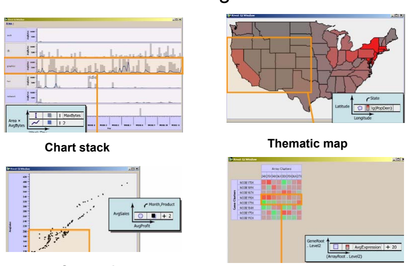


Chart stack **Thematic map** **Scatter plot** **Matrices**

Critiques

- Pros
  - Support normal zooming and semantic zooming (make use of the "structured" nature of data) on databases visualization
  - Try to formalize the relationship between zooming and data semantics. Not just treat zooming as a HCI technique
- Cons
  - The generality of proposed formalism for zooming has not been proved (currently applicable to 4 design patterns)
  - Did not address Focus+Context or retaining original visualization for referencing after zooming

Paper Reviewed (3)

- Mihael Ankerst, David H. Jones, Anne Kao, Changzhou Wang

"DataJewel: Tightly Integrating Visualization with Temporal Data Mining"

Overviews

- Temporal Databases
- Information Tasks of Temporal Data Mining
- Non-expert integrated Solutions-DataJewel
- Aircraft Maintenance Data Scenario
- Critiques

Temporal Databases

Column: Time Stamp + Event Attributes
Row : Time + Events

Time Stamp Event Attributes

Date	Airlines	Model	Problem ID	System Affected
1-Nov	Air Canada	737	A	engne fuel
	Air Canada	737	B	engne fuel
	Air Canada	747	C	commication
	United Airlines	747	B	communications
	United Airlines	737	B	engine fuel
	United Airlines	737	A	engie fuel
2-Nov	...	...	...	...
...	...	...	...	...

Information Tasks of Temporal Data Mining

- Which event has anomaly during the a certain period of time?
- Is there any other event that has the similarly abnormal pattern like the already observed event?
  - Within same event attribute
  - Cross event attributes
- *Example:*

  During 1990 to 2000:
  - Which airplane system has significantly low or high relative frequency of being affected by problems reported?
  - Which else airplane system has the similar troublesome situation? (within event attribute)
  - Which model, airline, etc has the similar troublesome situation? (cross event attribute)

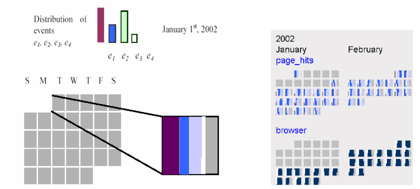
Non-expert Integrated Solutions-DataJewel

- [Visualization guided] + [Domain expert centric] data mining
- Innovative Temporal Data Visualization: CalendarView
- Visualization Interaction
  - Select Date Range, Ascending/Descending order, Interactive color assignment, Zooming, Detail on Demand
- Data Mining algorithm
  - *LongestStreak*: Single Event Anomaly Identification
  - *MatchingEvents:* Events Anomaly identification within Event Attribute
  - *MatchingEvents2:* Events Anomaly identification across Event Attribute
- Aggregated Database
  - Data amount is reduced by computing statistics summary

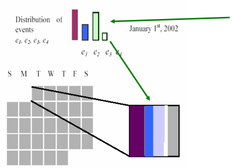
Visualization guided + Domain expert centric

- Overview of data are first given by visualization
- Domain expert iteratively takes following actions based on his knowledge and the visualized overview of data
  - Filter data by selecting date range, or
  - Interact with the visualization to explore patterns, or
  - Initiate data mining when spotting suspicious patterns
- Also can select different visualization techniques in accordance with the data size

CalendarView(1)



CalendarView(2)

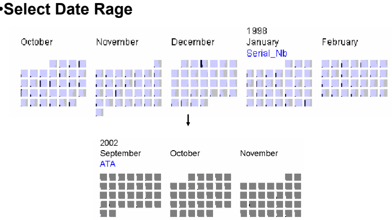


data of each day is encoded in the calendar day as a histogram where height indicates occurring frequency while color means different events

Event dates is represented by visual metaphor of a calendar

Visualization Interaction(1)

•**Select Date Rage**

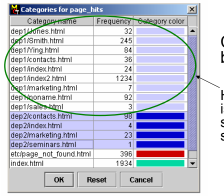


•**Ascending/Descending order**

rarest event in the front/ most frequent in the front

Visualization Interaction(2)

•**Interactive color assignment**



Conceptual generalization by giving same colors:

Htmls hitted in the directory *dep1* is abstracted/generalized into the same event by assigning them the same color

Data Mining algorithm

- LongestStreak
  - Calculate "relative frequency" of event E of each day
  - Calculate the mean and deviation of the relative frequencies of event E
  - Days in which the relative frequency of event E is significantly lower or over the mean value are labeled "*significant day*"
  - Return the *longest streak* of *consecutive* significant days by darkening them
- MatchingEvents
  - Calculate "significant days" for all other events in the same event attribute
  - For every event, assign bit 1 to significant days, bit 0 otherwise. Therefore, every event has its own "bit sequence"
  - Compare the bit sequences between event E with all other events; the most matched event is the correlated event to event E
  - Return both event E and the correlated event by changing their color
- MatchingEvents
  - Similar to MatchingEvents, but cross attributes comparisons

Aggregated Databases

- Original relational tables are compressed by computing the summary statistics: count(), sum(), average(), etc.
  - Example:

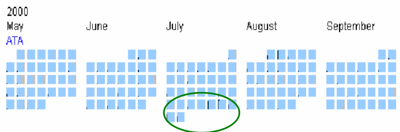
    Wireless signal disconnect 50 times a day. Without aggregation, 50 records!

    By calculating average disconnect time or count times of disconnection, 50 records becomes 1 record.
- # of events/day VS. # of distinct events/day
  - In aircraft maintenance domain:

    Average # of events per day: 402

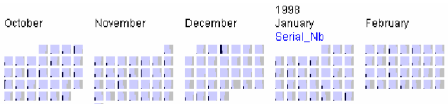
    Average # of distinct events per day (by aggregation): 32
- Greatly reduce memory capacity requirement!

Aircraft Maintenance Data Scenario (1)



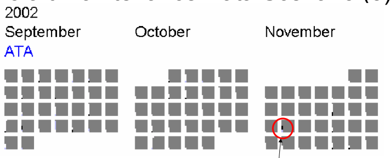
By LongestStreak and then visualization, the high occurrences of engine fuel problem are spotted during the end of July 2000

Aircraft Maintenance Data Scenario (2)



By adding a event attribute of "Plane ID", executing MatchEvents2, and visualization, one airplane correlate to the engine fuel problem is singled out. And we can see the engine fuel problem pattern of that airplane through visualization

Aircraft Maintenance Data Scenario (3)



By conducting MatchEvents and visualization, we can find that it seems that engine fuel problem would co-occur with communication problems

Visualized results of "MatchEvents"

Critiques

- Pros
  - Interaction between data mining and data visualization for efficiently exploring huge databases
  - Non data mining experts can mine more meaningful information
- Cons
  - Application specific

    # of events attribute<10; # of events per event attribute <200; smallest time unit is day
  - Limited tasks

    Limited to find anomalies and correlations
  - Limited Data Type

    Data limited to nominal data type

Paper Reviewed (4)

- Alexander Aiken Jolly Chen Michael Stonebraker Allison Woodruff

  "Tioga-2: A Direct Manipulation Database Visualization Environment"

Overviews

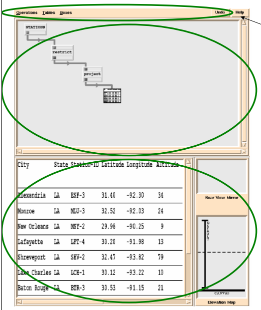
- Intro. Of Tioga-2
- User Interface of Tioga-2
- Model of Presenting Data of Tioga-2
- Details of Presenting Data of Tioga-2
- Miscellaneous of Presenting Data of Tioga-2
- Critiques

Tioga-2

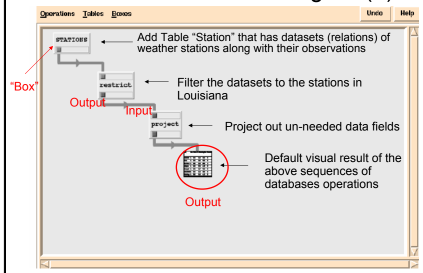
- An visual SDK environment for databases applications
- Visual programming:
  - "Box" represents primitives of program operations and database operations
  - "Arrow" represents the sequencing of the primitives.
- Visual feedback:
  - Visual demonstration of results of each programming steps in real time
  - *Example:*

    Visually shows the data queried for the SQL instructions.
- Focus on the latter part—visual feedback....

User Interface of Tioga-2 (1)



User Interface of Tioga-2 (2)



* Case of US weather stations & weather observation

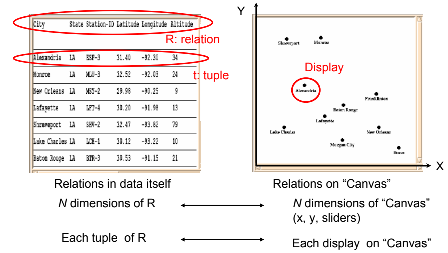
Model of presenting data of Tioga-2 (1)

- "Box" (or primitive procedure) will generate "output", which is the "input" of the successor "Box".
- "Inputs" or "Outputs" of database primitive procedures actually are datasets (relations or tuples). They are referred as "displayable" in the Tiago-2.
- "Displayable" includes:
  - Extended Relations (R)
  - Composite (C)
  - Group (G)

Model of presenting data of Tioga-2 (2)

- Extended Relations:

Relations in data itself + relations on "Canvas"



Model of presenting data of Tioga-2 (3)

- Composite:
  - Data semantic: Union of different relations
  - Visual semantic: Superimposition of "Canvases" (or visualization) of different relations
- Group:
  - Data semantics: Union of different composites
  - Visual semantics: Juxtaposition of visualizations of different composites.
- Elevation:
  - Data semantics: number of tuples shown on the "Canvas"
  - Visual semantics: degree of zooming (the height you watch the image)

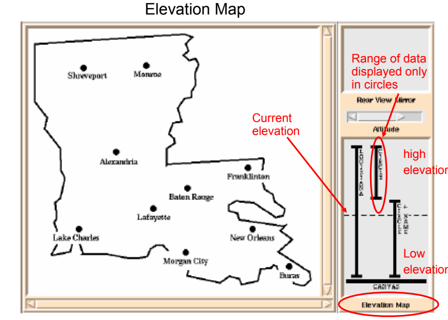
Detail of presenting data of Tioga-2 (1)

- Location and display attributes of data
  - Location attributes determines how to position tuples on 2D canvas (x axis, y axis, sliders)
  - Display attributes determines how tuples look like on 2D canvas (point, line, rectangle, circle, polygon, text, viewer (viewer on canvas))
- Default location and display of tuples (default visualization)
  - Spreadsheet like table
- Operations for altering visualization
  - Add attribute of data itself along with location or of display
  - Set attribute of location or display)
  - Remove attribute of data itself along with location or of display)
  - Swap attribute of data itself along with location or of display)
  - Scale, Translate attribute of location
  - Combine attribute of display)

Detail of presenting data of Tioga-2 (2-1)

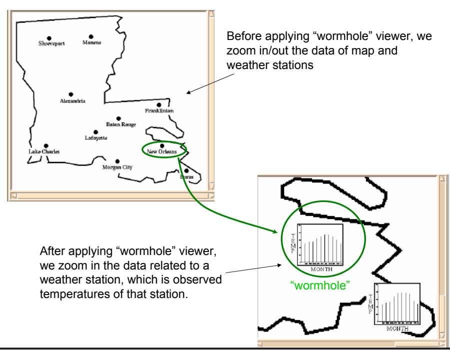
- Drill down
  - Refined view of the same data
  - Changed view of different but related data
  - Rear View Mirror
- Refined view of the same data
  - Set Range: Set range of data that a view can zoom in/out
  - Overlay: Overlay different displays of the same data. *Example: Display texts and circles when zoom in; Display circles only when zoom out*
  - Shuffle: Change drawing order of relations within a composite.
  - Elevation map: a bar-chart display indicating the range of data displayed, overlaid displays, and drawing orders

Elevation Map



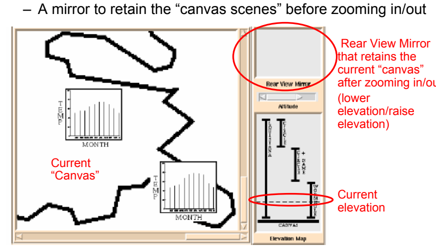
Detail of presenting data of Tioga-2 (2-2)

- Changed view of different but related data
  - Wormholes
    - A viewer mentioned previously
    - A viewer onto another canvas, which visualize datasets relating to the data visualized on the current canvas
    - Defined by parameters of size of the viewer, the destination canvas, the elevation (# of datasets) from which the canvas is viewed, etc.



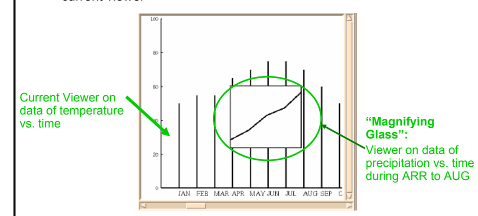
Detail of presenting data of Tioga-2 (2-3)

- Rear View Mirrors
  - A mirror to retain the "canvas scenes" before zooming in/out

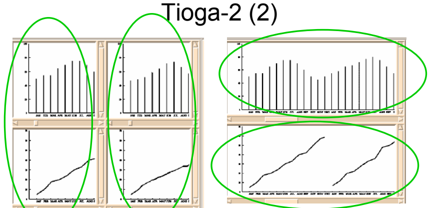


Miscellaneous of presenting data of Tioga-2 (1)

- Slaving Views: Move or delete "slaved" viewers together
- Magnifying Glasses: Overlap viewer of other data on current viewer



Miscellaneous of presenting data of Tioga-2 (2)



- Replicated Viewer
- Stitched View
  Stitch two viewers

Critiques

- Pros
  - Pioneered concept of multi-scale visualization of databases
  - Visualization for aiding programming in real time
- Cons
  - Users are still tasked with being required to be familiar with SQL queries and basic programming primitives– not suitable for general public
  - Users are tasked with configuring visualization- non visualization expert might not feel the advantage of flexibility