

VizDeck: Streamlining Exploratory Visual Analytics of Scientific Data

Daniel Perry
dbperry@uw.edu

Bill Howe
billhowe@cs.washington.edu

Alicia M. F. Key
amfkey@gmail.com

Cecilia Aragon
aragon@uw.edu

University of Washington

Abstract

As research becomes increasingly data-intensive, scientists are relying on visualization very early in the data analysis cycle. We find that existing tools assume a “one-at-a-time” workflow for creating visualizations and impose a steep learning curve that makes it difficult to rapidly create and review visualizations. At the same time, scientists are becoming more cognitively overloaded, spending an increasing proportion of time on data “handling” tasks rather than scientific analysis. In response, we present *VizDeck*, a web-based visual analytics tool for relational data that automatically recommends a set of appropriate visualizations based on the statistical properties of the data and adopts a card game metaphor to present the results to the user. We describe the design of *VizDeck* and discuss the results of a usability evaluation comparing *VizDeck* with three other popular visualization tools. We then discuss design considerations for visualization tools focused on rapid analysis based on observed sensemaking processes.

Keywords: information visualization, visual analytics, user interfaces, scientific collaboration

Introduction

Advances in high-throughput data acquisition technologies have transformed many fields of research from data-poor to data-rich. In turn, the need to analyze potentially large datasets from a variety of different sources has increased the cognitive load of researchers and scientists, taking time away from core research activities. While visual perception has been found to be the highest bandwidth channel from the computer to the human cognitive centers (Ware, 2004), many existing visualization and visual analytics tools require a training period that makes it time-consuming for researchers to bootstrap their usage. Moreover, rapid assessment of new data sources requires *data triage*: quick assessment of the utility, quality, gross patterns, and internal relationships of the data. This task typically requires rapid review of a large number and variety of related visualizations. Existing tools appear to be ill-equipped for data triage tasks, in that they force the user to spend time designing each visualization created, limiting simultaneous review of several visualizations at once. The question of which types of interactive visual interfaces maximize human efficiency and exploratory analysis is still open, despite a great deal of recent work in this area. As scientists and researchers become ever more cognitively overloaded, the discovery of the best techniques to rapidly leverage human pattern recognition and visual explorative abilities is becoming progressively more crucial.

Research has shown that cognitively overloaded users prefer a simple interface optimized for speed and ease-of-use over a feature-rich interface with a steeper learning curve (Aragon & Hearst, 2005). Additionally, the growth of scientific collaborations that span multiple research groups and disciplines often means that analysis tools already integrated into a researcher’s workflow might not be the same across groups. This effect is measurable: in some scientific collaborations, researchers report that the ratio of time they spend “manipulating data” as opposed to “doing science” is approaching 9 to 1 (Hey, Tansley, & Tolle, 2009). As a result, researchers increasingly rely on visual browsing techniques for exploratory analysis to quickly identify patterns for deeper study.

This technology gap for rapid exploratory analysis exists in a variety of domains. Astronomers are beginning to build mash-ups to support ad hoc analysis of object catalogs (ASCOT, 2012). In the life sciences, extemporized integration of laboratory assays, public databases, and computational simulation has become commonplace (Howe et al., 2011). Oceanography is transitioning from an expeditionary science to an observatory-based one, where data is streamed real time from an array of sensors (Baptista, Howe, Freire, Maier, & Silva, 2008), decreasing the time available for analysis. In our own experience, when we suggested sophisticated visual analytics tools to support this interaction style (e.g., Tableau), we found that our collaborators, unable to self-train quickly, reverted to simpler tools such as spreadsheets.

Optimizing data visualization for human attention, especially in the early phases of data exploration and data triage, has emerged as a clear goal. To this end, we developed VizDeck, a method and prototype for web-based visual analytics over relational data that uses a card game metaphor and automatic visualization techniques to assist users in creating interactive visual dashboards with no programming. Informed by the statistical properties of the underlying dataset, VizDeck pre-generates candidate visualizations (and UI widgets) called *vizlets*, displaying them to the user in a grid. These vizlets can then be browsed, discarded, or promoted for inclusion in an interactive dashboard. The user does not need to perform explicit configurations to construct a dashboard; vizlets automatically respond to brushing and linking events triggered by other vizlets.

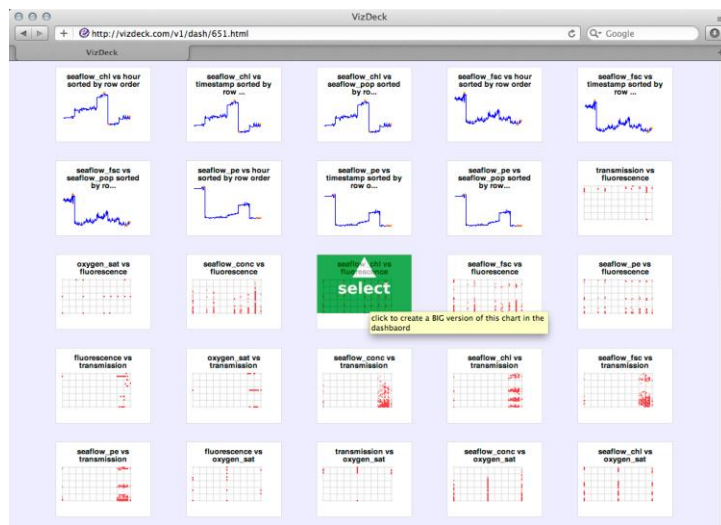


Figure 1. VizDeck automatically generates a “hand” of thumbnail visualizations called vizlets.

Dashboards can then be saved and shared among collaborators by exchanging URLs. Besides affording an undo/redo capability, this technique allows a collaborator to review the steps taken by the original author to create the dashboard. This type of provenance has been shown to improve cross-training and communication between users (Bavoil et al., 2005).

To assess how the process of automating the visualization process affected data exploration, we conducted an initial user study of 32 subjects (with 315 trials run) testing their ability to complete data analysis tasks. We also conducted semi-structured interviews with each of the subjects to better understand their sensemaking process. Russell et al. (1993) describe sensemaking as “the process of searching for a representation and encoding data in that representation to answer task-specific questions”, noting that this process requires varying cognitive and external resources. In our design and evaluation of VizDeck as a tool, we were particularly interested in understanding how both efficiency and sensemaking were affected by the automation of the visualization process.

The contributions of this work are 1) a new interface design that accelerates visualization selection and dashboard assembly tasks for researchers when compared to more complex visualization tools, 2) a quantitative and qualitative analysis of users’ rapid exploration of scientific data, and 3) a prototype implementation that is available on the web (<http://escience.washington.edu/vizdeck>).

Related Work

There have been previous examples of systems that attempt to capture the data exploration process. Jankun-Kelly et al. (2002) proposed a model for recording multiple visualizations as part of a research process. Similarly, VisTrails, a system developed by Bavoil et al. (2005), offers a framework for showing multiple visualization pipelines within the same visualization “spreadsheet”. While these tools provide ways to view multiple visualizations, they are designed for experienced users who expect a significant degree of control over the visualization, and are not optimized for the early stage data triage task.

More recent work on intelligent user interfaces attempts to infer the user’s task from behavior, and use this information to recommend visualizations (Gotz & Wen, 2009). Dörk et al. (2008) derive coordinated visualizations from web-based data sources. Additional frameworks for creating mashups or simple interactive applications combining data from multiple sources have been proposed (Elmeleegy, Ivan, Akkiraju, & Goodwin, 2008; Ennals & Garofalakis, 2007), assuming a pre-existing repository of mashup components. Elias and Bezerianos (2011) focus on dashboard creation by both experts and novices and include chart recommendation features. These relatively sophisticated dashboard components must be constructed individually by the user through a multi-step process. Other visualization tools such as Many Eyes (Viegas, Wattenberg, van Ham, Kriss, & McKeon, 2007) and Tableau’s “Show Me” feature (Mackinlay, Hanrahan, & Stolte, 2007), also take an approach of providing an existing menu of predefined visualization types for users to explore. In contrast, VizDeck generates and presents a diverse set of fully realized visualizations for users to browse and choose from.

Previous research points to the importance of evaluating visual analytics tools within the context of the users’ sensemaking process. Scholtz (2006) recognized the importance of going beyond usability as a metric for evaluating visual analytic systems, noting that time and completion should be augmented with other metrics that account for factors including situational awareness, interactions, and creativity. Kang et al. (2009) evaluated Jigsaw, a multi-view system for investigative analysis, and noted the importance of understanding the sensemaking process, especially as it related to visual exploration of data in a study comparing the tool to lower-tech methods, including paper and pencil.

Design Considerations

Objective assessment of visualization quality is considered one of the grand challenges in the visualization community (Johnson, 2004), and is a prerequisite for automatic visualization of data. Seminal approaches to this problem involve heuristics based on human perception (Mackinlay, 1986). In VizDeck, we encode this style of heuristics (when they exist) to influence the recommendations, but augment them by training a model that learns the relationship between statistical features of the data and visualization properties. In this paper, we address the effect of the UI design features on usability and sensemaking rather than focusing on system design or the details of the statistical recommendations. In designing VizDeck to afford rapid exploration of scientific datasets, we took into account 1) Tufte’s (1990) principle of small multiples exploration, 2) automatic brushing and linking, and 3) the importance of human-in-the-loop interactions.

Small Multiples Exploration

We wanted to provide an efficient manner for users to pan the visual field of the dashboard display and detect trends or surprising results in the dataset. These thumbnails, which we refer to as *vizlets* (broadly defined as any UI element that either displays data or accepts user input), were inspired by small multiples (Tufte, 1990). Tufte argued that displaying several simpler visualizations can be more effective than displaying one dense, data-rich visualization. The VizDeck interface uses vizlets to compactly express the structure of many visualizations at once. The user rearranges these vizlets, discards those that are irrelevant, and promotes those that warrant closer inspection (Figure 1). We liken this process to a card player manipulating their hand in preparation for strategic play. To support efficient browsing of many screenfuls of vizlets, we also provide a search-box feature to allow users to inspect groups of related vizlets at once. Search terms may include references to vizlet type such as “scatter” or “line,” or references to attribute names, such as “oxygen” or “fluorescence” (Figure 3). We provide basic auto-completion for the search bar with disjunctive semantics (term1 OR term2 OR term3) to decrease the likelihood of empty result sets.

Automatic Brushing and Linking

Once promoted, vizlets typically expose additional interactivity features. For example, values can be selected in drop down boxes, and points on the scatter plots can be highlighted. These interactions automatically affect the display of *all* other vizlets, both those in the dashboard and those in the grid. Unlike similar features in other tools (e.g., Tableau), no explicit configuration is necessary to establish links between vizlets. As a result, users can assemble coordinated interactive dashboards in just a few seconds with no programming and no explicit configuration; Figure 2 illustrates an example of this feature. The underlying dataset arises from measurements from an environmental flow cytometer called SeaFlow (2011), an instrument deployed on oceangoing vessels that automatically measures population counts of microorganisms. The user has selected a drop down box displaying a list of microorganism types (upper left), a timeline of total particle counts by time (lower left), a histogram of population counts for each type by hour (upper right), and a scatter plot of two optical properties (lower right). Each vizlet must respond to this action, but the exact semantics of these events vary depending on the vizlet type: designers are free to define the specific behaviours of the vizlets as they see fit.

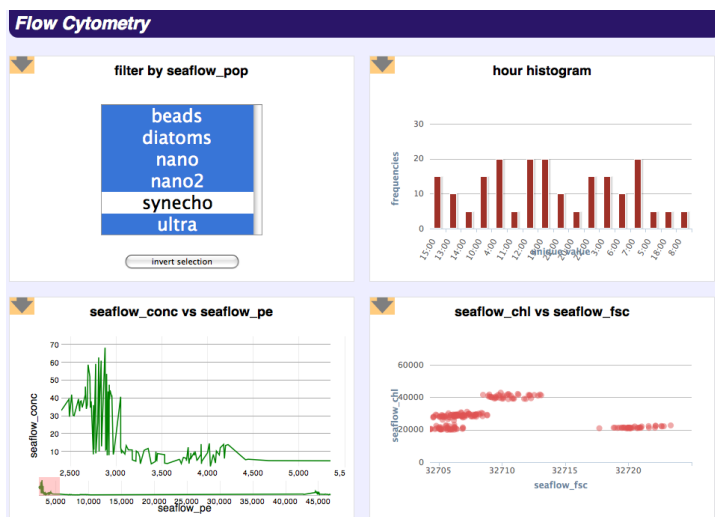


Figure 2. An interactive dashboard in VizDeck where the user has added four vizlets to the dashboard.

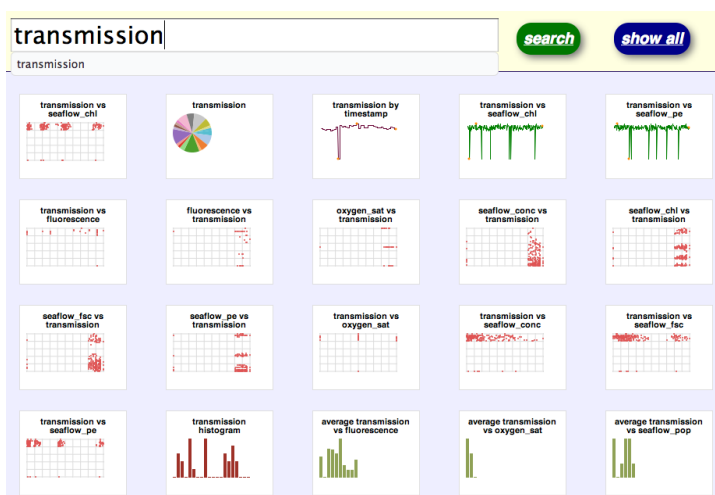


Figure 3. Keyword search equipped with auto-completion can be used to filter the grid of vizlets.

Human in the Loop Interactions

As we have described, the vizlet designers have some control over how their visualizations will be displayed and when they will be recommended. However, we emphasize human-in-the-loop interactions to improve the recommendations over time, independently of the static, design-time considerations. In particular, we log the interactions of the user and use the results to inform a decision tree-based ranking function such that the system learns over time which types of vizlets are more likely to be effective for a dataset with particular features.

Users either promote or discard vizlets based on the thumbnail's perceived utility for their task. When a user promotes a thumbnail vizlet to a full-sized vizlet, we interpret that as a "like" vote from the user, and a discard action as a "dislike" vote from the user. We describe each vizlet with a feature vector and use it to train a linear model of historical voting data to assign a score to the candidate vizlets with respect to the underlying data. We are currently addressing appropriately weighting these human-in-the-loop interactions with more structured ranking heuristics based on display type. These details are largely independent of the UI methodology we evaluate in this paper, and are not discussed in detail.

Implementation

The VizDeck system is comprised of client and server subsystems. The server is written in Ruby and is responsible for connecting to data sources and retrieving results of queries presented by the user on the client side. The JavaScript client runs in the browser and is responsible for rendering the vizlets and controlling user interactions. VizDeck currently consumes data from SQLShare, a database-as-a-service application that favors ad hoc integration, query, sharing, and visualization over predefined schemas (Howe et al., 2011). Each dashboard is automatically assigned a unique, permanent URL when first created. This URL can be shared with other users at any point in the authoring process allowing researchers to share interesting findings.

When the server connects to a data source and retrieves results for a query, it creates compact definitions of vizlets called protovizlets. A protovizlet is a set of key/value pairs that collectively describe the visualization or UI widget so it can be rendered on the client side. At a minimum, the protovizlet must contain a "title" (a string displayed to the user) and a "type" (a string indicating the rendering method to use on the client; e.g., scatter, bar, etc). In the current system, we offer a variety of simple vizlet types: bar charts, histograms, line charts, timeline plots, pie charts, maps, and multi-select boxes (for selecting values to filter the data displayed in other vizlets). The client then converts these protovizlets into interactive vizlets that are displayed to the user. Before the protovizlets are sent to the client, the server examines the scores and sorts the protovizlets by their score. The score is computed by estimating the likelihood of selection of each vizlet based on a learned relationship between statistical features of the data used to generate the vizlet and the visualization type. To train the model, we use log data collected by the system as ground truth. Each time a user promotes a vizlet, we interpret the promotion as a vote that the vizlet was considered important enough to inspect more closely. Similarly, a discard action is a vote that the vizlet was not useful to the user. A positive net sum of these votes across users indicates the classification as "good" or "bad." We then attempt to predict this classification from features of the data. For example, a line chart is less useful for sparse and irregular data, and a bar chart is not useful when there are too many or too few categories. The details of the scoring algorithm are outside the scope for this paper. The server also compares the protovizlets to each other with similarity scores to diversify the order in which the vizlets are to be presented to the user. Finally, the server selects the top k protovizlets to recommend to the user.

Evaluation Methodology

We conducted a quantitative and qualitative evaluation to assess VizDeck's performance for finding relevant information within a scientific dataset when compared to three commonly-used data visualization tools. We hypothesized that subjects would be able to compare, analyze, and explore information within the dataset faster with VizDeck's automated approach. The study included 32 participants (14 female), ranging in occupation and ages (19 to 60). Subjects were recruited through campus mailing lists. Sixty-six percent of the subjects were from science, engineering, or medical fields, while the others came from social science, business, education, and public policy. As data analytics

becomes an increasingly pervasive need among various research communities, we decided we did not want to limit participant involvement to a particular field. Almost seventy percent of the subjects identified their primary occupation as students. Thirteen of the participants reported having moderate to high levels of experience with data visualization tools, with the most commonly reported methods of visualization being Excel, R, and MATLAB.

We evaluated VizDeck along with three additional interactive visualization software tools: ManyEyes, Google Fusion Tables, and Tableau. We selected these particular tools on the basis of their functional similarity to VizDeck and ability to rapidly visualize data with no programming necessary. Many researchers employ different tools depending on their domain and require training or development skills; we distinguish our focus from that of high performance, 3D scientific visualization tools such as ParaView (Aherns, Geveci, & Law, 2005). We were primarily interested in exploratory data analysis and “data triage” tasks that emphasize “rapid prototyping” of visualizations rather than in-depth analysis. We therefore excluded tools such as IDL and MATLAB from our study because they are programming environments rather than end-user applications. We wanted to better understand how scientists and researchers who do not necessarily have a programming background can quickly and easily make discoveries in their data. We also did not test against Microsoft Excel since we were interested in tools that offered interactive visualization functionalities with varying degrees of automation.

The three tools evaluated within the user study were:

Tableau Desktop: Tableau is a highly polished, feature-rich commercial visualization tool that allows users to drag and drop field names to visualize data and create interactive dashboards. The desktop version of the software runs on Windows and can open data files from a variety of Windows compatible formats. It also provides a “Show Me” feature, a set of UI commands for automatic presentation (Mackinlay et al., 2007).

Google Fusion Tables: Fusion Tables is a Google web application that allows users to visualize and publish data as maps, timelines and charts. Data tables can be uploaded from spreadsheets, CSV, or KML format and the tables are hosted online (Gonzalez, 2010).

Many Eyes: Many Eyes is an online data visualization tool produced by IBM Research and the IBM Cognos Group. It allows users to cut and paste data into a field and then select visualization types and share these with others. The goal of the site is to promote a “social style of data analysis” in which visualizations promote discussions among users (Viegas et al., 2007).

Experimental Design

We conducted a user study with 32 participants each performing tasks on two of the four interfaces (a software error excluded one set of trials). The study included both between-subjects and within-subjects variables, since each subject evaluated two software interfaces. Subjects were asked to utilize one of the data visualization tools to complete five tasks regarding data about the periodic table of elements. The data set included information on element names, atomic number, boiling points, melting points, first-ionization energies, group number and period number. Properties included those of chemical elements from Hydrogen (H, atomic number 1) to Uranium (U, atomic number 92). For each element, available data included: atomic number, boiling and melting points in Kelvin (K), first-ionization energy in electronvolts (eV), period (1-7), and group (1-17). These are basic chemical properties that are taught in undergraduate chemistry coursework and considered important to understand relationships and chemical properties of the elements.

The performance of each subject was evaluated for the use of two of the software interfaces. In total 15-16 subjects performed the five questions on each interface. Each task was video recorded for researcher analysis, and VizDeck interactions were logged by the software system. Subjects were explicitly not given a training tutorial on any of the systems with the underlying assumption that training sessions would take valuable time away from researchers and scientists needing to perform science or analyze data. Subjects were informed that they could use the help documentation located within the software tool if necessary, although it was explained that time was a determining factor in evaluating how well they performed each task. The experimenter did not answer any direct questions regarding the capabilities of features of the interface, only noting that each question was in fact answerable using the given software interface. After subjects had completed the computer tasks, they were asked to describe the ease of use and their experience with each interface. This method was chosen over methods where

users might explain their thought process during the task itself in order to provide a more realistic estimate of time on task. This study was conducted in a university lab setting using desktop computers.

Task Description

The tasks were chosen to represent several types of visual data exploration tasks that could be answered with a brief inspection of a simple visualization: identifying a trend in a single variable, finding the mean, comparing two properties, finding the highest value, and recognizing a visual relationship between properties. These interactions were chosen as important qualifiers based on heuristics for presenting data for visual quantitative analysis (Few, 2009). The task order was randomized to account for learning effects, and two sets (Set A and Set B) of nearly identical tasks were alternated so that the answers were different between the interfaces.

The evaluation tasks for both Set A and Set B were as follows:

Task 1. Identify a Trend within a Group. Is there a noticeable trend for melting points (Set A) or boiling points (Set B) of the elements in Group 17 (Set A) or Group 12 (Set B)?

Task 2. Find the Average for a Given Period: Identify the correct average first-ionization energy for Period 3 (Set A), or correct average first-ionization energy for Period 6 (Set B).

Task 3. Compare Two Properties. Compare the melting points and first-ionization energies for Group 6 elements (Set A), or boiling points and first-ionization energies for Group 17 elements (Set B), and identify whether both properties were increasing, one was increasing while the other was decreasing, or both were decreasing.

Task 4. Find the Highest Value. Find the element with the highest boiling point (Set A) or the highest melting point (Set B) out of a collection of four elements. Finding the highest value in a list of numbers would not typically require visualization; we included this question primarily as a control.

Task 5. Fit a Curve or Line. Find a relationship between boiling point and some additional property that visually matched a given curve (Set A) or line (Set B). This question was unique in that it had users consider a visual representation of the data. We intended to capture situations where researchers might anticipate certain relationships in the data but are uncertain which variables will exhibit it. For example, clinical studies may generate hundreds of measurements about each patient and researchers may be looking for any measurements that visually follow the pattern of expression of a particular symptom.

Evaluation and Results

We found that the particular visualization tool used did have an impact on completion rate and time to completion for each task. The completion rate for the combined tasks shows that Fusion (97%) and VizDeck (94%) were comparable, with better performances than with Many Eyes (78%) and Tableau (68%), as shown in Figure 4 (left). Fusion (105 seconds) took on average the shortest amount of time to complete for all tasks combined, followed by VizDeck (130 seconds), Many Eyes (157 seconds) and Tableau (172 seconds) as shown in Figure 4 (right). We only included times on tasks that were successfully completed because we did not want to unduly weight the instances where subjects took a short amount of time but did not provide a correct answer. The withdrawal of these times made it challenging to establish statistical significance for time to complete. Therefore, we wanted a way to combine both the completion rate and the time on task to provide a more comprehensive and appropriate measure of efficiency.

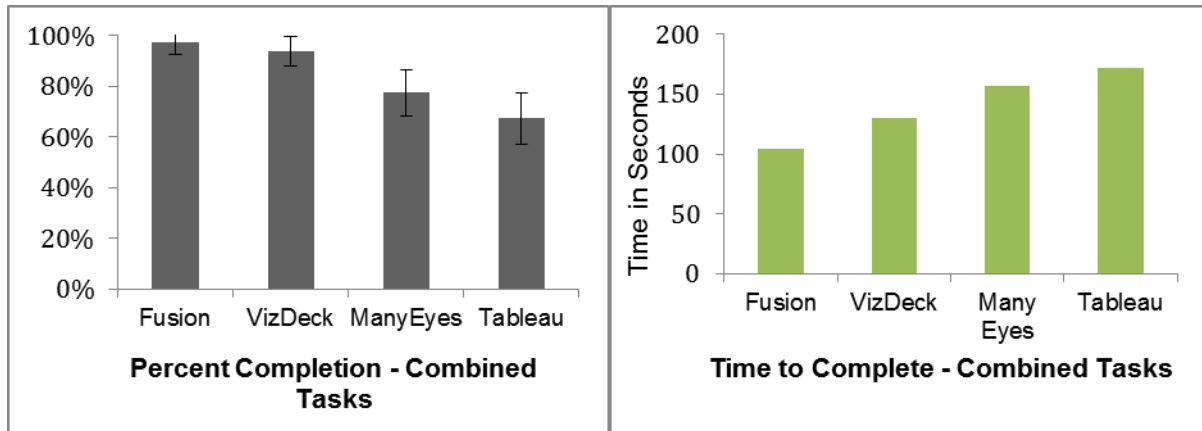


Figure 4. Combined task percent completion with error bars calculated at a 95% Confidence Level using the Adjusted Wald Method for Confidence Intervals (left) and mean time to complete combined tasks (right) for each visualization tool.

The National Institute of Standards and Technology (2001) efficiency metric provides a combined score for completion success and time to complete. The metric is a ratio of the task completion rate and time to complete and the Common Industry Format (CIF) for Usability Test Reports describes this NIST measurement as the “core measure of efficiency” (NIST, 2001). The NIST ranked the software tools similarly to the completion percentage, as shown in Figure 4. One of the challenges of using the NIST efficiency metric is that it does not provide a way to reliably determine the statistical significance of the result since it uses the mean before combining the time and completion, and cannot be calculated for each trial run. Additionally, it requires a reasonable unit of scale for each task, making it harder to compare results across questions.

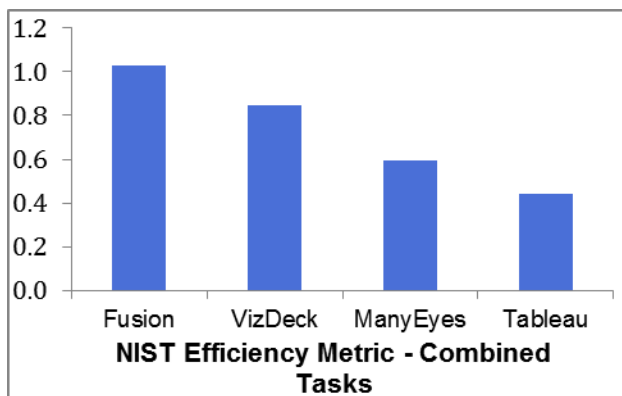


Figure 5. The NIST efficiency metric for combined tasks for each software tool.

Results: Time and Success Score

To determine the statistical significance of the results for each question we used the Time and Success Metric (TSM) as an efficiency metric that allows us to calculate a combined score for each individual trial (Perry, Aragon, Meier, Pepper, & Pritoni, 2001). The TSM takes time and the completion for each trial run as inputs, and is normalized between 0 and 1 by using a logistic transformation, offering a way to compare results across questions. The TSM score is given by the following equation:

$$M_I = \frac{2s}{1 + e^{x_i}}$$

where

$$x_i = \frac{t}{k} \quad (t = \text{time to complete task, and } k = 50, \text{ an empirically determined constant})$$

$s = 0$, if subject failed to complete task

1, if subject completed task

The Time and Success Metric score was calculated for each question, showing variability in ranking score dependent on the question (Figure 6). The data are not normally distributed, so we used the non-parametric Kruskal-Wallis H test to determine statistical results. Table 1 shows the Chi-square value (H) and significance level for each task (Table 1). Pairwise comparisons for the Time and Success scores using Mann-Whitney U tests also indicated significant differences between the tools (Table 2).

Table 1

Chi-square H value and significance level (p-value) for each task.

Task	H	P-value
T1. Trend	12.019	<0.01
T2. Average	18.885	<0.01
T3. Compare	30.681	<0.01
T4. Highest	18.331	<0.01
T5. Fit Line	12.689	<0.01

Table 2

Mann-Whitney U Test comparing the Time and Success Scores for tools. The bold border indicates tests involving VizDeck. Shaded boxes indicate tests for which VizDeck's performance was measurably better.

	Task 1	Task 2	Task 3	Task 4	Task 5
Fusion vs. ManyEyes	Fusion U=59.5, P<.05	no sig. diff.	Fusion U=7, P<.01	no sig. diff.	no sig. diff.
Fusion vs. Tableau	Fusion U=56, P<.05	no sig. diff.	Fusion U=16, P<.01	Fusion U=58.5, P<.05	Fusion U=39.5, P<.01
VizDeck vs. Fusion	no sig. diff.	VizDeck U = 51, P <.01	Fusion U = 69.5, P<.05	Fusion U = 16, P<.01	no sig. diff.
VizDeck vs. ManyEyes	VizDeck U=69.5, P<.05	VizDeck U=34, P<.01	VizDeck U=40.5, P<.01	ManyEyes U=63.5, P<.01	no sig. diff.
VizDeck vs. Tableau	VizDeck U=60, P<.05	VizDeck U=38, P<.01	VizDeck U=48, P<.01	no sig. diff.	VizDeck U=53, P<.01
Tableau vs. ManyEyes	no sig. diff.	no sig. diff.	no sig. diff.	ManyEyes U=90.5, P<.05	no sig. diff.

The Time and Success scores (Figure 6) show Fusion and VizDeck achieved better scores across all tasks, with the exception of VizDeck's performance on Task 4 ("Find the Highest Value"). This task was specifically designed knowing the limitations with VizDeck. Results showed Fusion performed better at identifying a trend, comparing two properties, and sorting data from high to low, while VizDeck outscored the three other tools at finding an average. Overall VizDeck's method of automated visualization outperformed Tableau and Many Eyes on the majority of tasks and attained comparable performance to Fusion on Tasks 1-3 and Task 5. Our goal was not only to analyze time and accuracy on tasks but also to better understand challenges and opportunities in users' sensemaking processes in using these systems to facilitate rapid exploratory analysis of data. After gathering the quantitative results, we then turned to qualitative data to extract user sensemaking strategies.

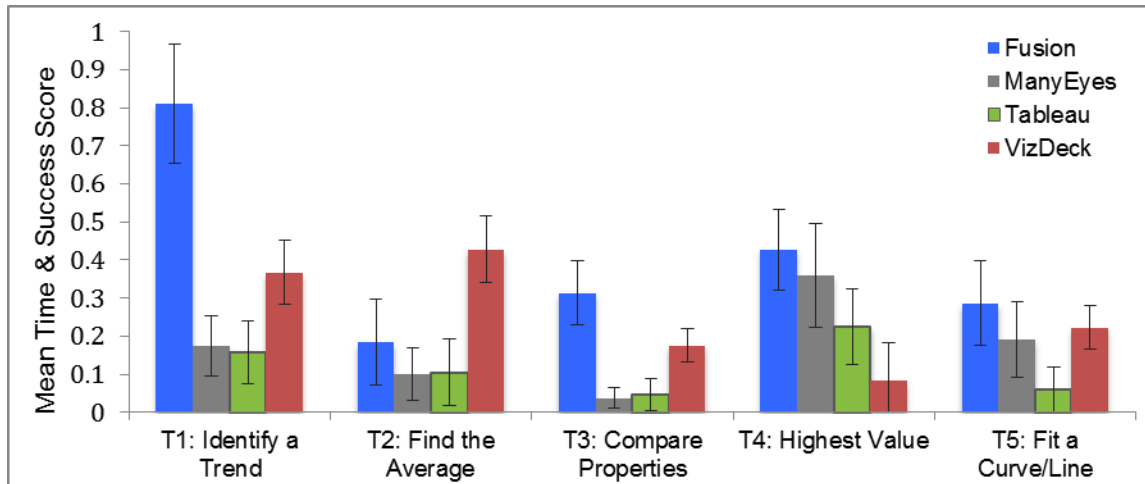


Figure 6. The mean *Time and Success* score with 95% Confidence Levels for each task where higher (the closer to 1) is better performance. Fusion Tables results are attributable in part to the subjects' use of non-visualization features that are incomparable to the other systems.

Discussion

While the five tasks as quantitative measurements give indicators of software performance, users' qualitative evaluations and visualization tactics provide a way to interpret the results within the context of the sensemaking process. Participant interview responses and comments, along with analysis of the video and log data, provide a further context for understanding users' investigative and analytic strategies that are described below.

Catch and Release. One commonly used strategy for analyzing the visual data presented in VizDeck was what we have termed a "catch and release" tactic, in which users promote one or more vizlet thumbnails to the dashboard to inspect them, and then, if they did not contain relevant information, "release" them back into the display area below the dashboard by clicking the discard icon. While these select and discard interactions provided helpful data for the human-in-the-loop interactions, this strategy actually cost users valuable time in the search process.

One seeming reason for such a strategy was users' reliance on the vizlet titles to provide a context for the thumbnail view. By first promoting the vizlet to the dashboard, users could gain a clearer view of the title, and decide whether the visualization contained relevant data. One participant identified the challenge of gaining cues from the titles in the thumbnail view noting, "It was easy to use to drag and drop [vizlets to the dashboard], but it was harder to read the titles, and sometimes they weren't visible."

Research on information scent processes provides the context of information foraging (Pirulli, Card, & Van Der Wege, 2001) as a way to better understand this *catch and release* strategy. VizDeck automated much of the work for the user in creating the visualization; yet conversely, it increased the need to provide clearly visible and explicit information scents, or cues, for each visualization thumbnail. Further exploration is needed to determine information scent cues that could most effectively allow users to scan the thumbnails without the additional time taken evaluating the image in a dashboard view. One possible method of overcoming this might be to provide an enlargement of the vizlet upon mouseover. To further support a strategy like *catch and release*, it is critical that visual interfaces provide at a glance metadata about the visualization.

Thinking Ahead of the Visualization. Another common strategy observed in the tasks was users' attempts to guess the visualization type that would be most relevant to the given task. This was most evident in interactions with Many Eyes, as users had to select different visualization types during their initial interaction with the interface. Users therefore spent large amounts of time switching between visualization types in their attempts to find the right one. One user of Many Eyes went back and forth between the different visualization types (histogram, bar, scatter plot, etc.) eleven times before settling on one. A subject who was a software trainer by occupation, noted the challenge of having to guess which visualization type would be most useful, saying "[With Many Eyes] I have to think ahead of the data about

what I would like.” This challenge highlights one of the key benefits of VizDeck’s automated approach in pre-creating visualizations of a variety of different types based on the properties of the data. VizDeck notably reduced the “visualization choice” problem to one of a “search-and-browse” problem.

Table Filtering. A common strategy employed by subjects using Fusion Tables was the use of the table view to scan and order the data. It should be noted that while both Many Eyes and Tableau afforded a view of the raw data in table format, users that encountered this view rarely used it to complete tasks, instead going back to the visualizations to answer questions. Conversely, there were numerous Fusion Tables users who attempted to complete tasks using one of the visualization views, and then reverted back to the table view.

User comments provide insights into the sensemaking processes that led users to use the table view instead of visualizations. One user commented that it “felt more instinctual, primarily because I have experience with Excel and Access,” a sentiment that was echoed by several users. Another user stated, “it was easy to sort through things.” Two considerations arise from such comments: one is that the table view was indeed more familiar to many users than the visualizations, due to their experience with Excel, and second is that the ability to manipulate and “sort” through the data (which Fusion afforded and the other tools’ table views did not) was key to Fusion Tables’ successful use. The latter observation was corroborated by the several failed attempts of users who tried to use the raw data view in Many Eyes and Tableau. In these tools, because the raw data was static, and had no affordances for sorting or filtering the data, users found it challenging to find the information they needed. Raw data browsing of large datasets (GB and TB) are notably difficult to inspect, which is why we had focused our evaluation on tools that allowed for rapid visualization. However, users’ familiarity with filtering data within Microsoft Excel raises compelling questions as to how similar functionality could be supported within an interface that scales with larger datasets.

Visual Data Insecurity. Another interesting strategy that emerged from our observations was a sense of insecurity that many users displayed regarding their selection of the correct visualization. Subjects using Fusion Tables would often cycle through several iterations of visualizations (changing the properties of the x-axis and y-axis) before selecting a visualization to answer a question. This often meant that users scanned and passed over relevant data in their quest for a better visualization. This process was similarly observed in VizDeck users for the “Fitting the Curve” task. When looking at the outliers of the time-to-complete for VizDeck for this particular task, two (of the three outlier) subjects actually selected the correct visualization, but continued to promote several additional vizlets to the dashboard. These actions suggest that users were insecure about their first visual instinct, even when it had been accurate.

The majority of the Fusion users rapidly applied filters, and were able to cycle through all options until there were no longer any combinations of properties left to view. This strategy seemed to provide an “end of the line” assurance and additional security to the users, in which they would then go back to the most relevant visualization created. Such observations provide insights into a much deeper challenge: how can visual analytic interfaces be built to support a sense of security and assurance to users that they have indeed checked the most relevant visual options? We posit that tools should be designed to provide easy means of supporting this verification process and not just the discovery process.

One of the primary issues that many subjects identified was the trade-off between ease-of-use in automating the visualization creation process and the control in manipulating these visualizations. Many users expressed the opinions that VizDeck was easy to learn, and quick to view. However, some users commented on the lack of power in customizing what they were viewing. One user noted “the learning curve for [VizDeck] is not as steep” as that for Fusion Tables, but that with Fusion Tables “I feel like I have more control over it.” One participant, a researcher with extensive experience using PyLab and MATLAB, noted a preference for greater analytic control in his evaluation of VizDeck, commenting that “the graphical web based interface is convenient but feels shallow for bigger more complex problems.”

Conversely, there were numerous subjects who found the simplicity of the automated “instant-view” thumbnails of VizDeck familiar to other graphical interfaces they had used. A chemistry student noted that VizDeck “feels like it was made for my generation, quick to observe the trends, and you can search for whatever you want.” This trade-off over how much the system or the human does the work in creating the visualization was a key motivator in the design and development of VizDeck. We were keenly aware that this was not going to be a tool designed for fine-grained visualization or complex function control. The user task “Find the Average Period” provides the strongest example of VizDeck’s successful use of automation (with an average completion time that was 67% faster than that of any of tool). In this

task, it was clear that the human had to do less work to view and compare data on averages than they did with the other systems.

A “sort” option has not been integrated into the VizDeck tool, resulting in users’ poor performance on the “Find the Highest Value” task. This limitation in rearranging the data meant that users had to scroll through a chart, and we anticipated that VizDeck was unlikely to support this task. One option is to provide improved sorting functionality once the user has selected one of the automated vizlets. This challenge once again highlights the tension between the work required in creating the visualizations and the users’ sense of control in manipulating the data.

Future Work

There are several limitations in the VizDeck tool we would like to address in future work. We intend to refine the ranking function, which will become stronger in classifying vizlets as we gather data from the human-in-the-loop processes. We would also like to add additional vizlet types to offer the most appropriate visual cues for users to explore the data. Additionally, we plan on modifying VizDeck to better understand how much control should be provided in functionality to sort and organize data along several different categories, such as by high/low. Further work is also needed to connect VizDeck to other data repositories such as Google Fusion Tables and Microsoft Data Market. As data is increasingly shared among researchers in varying and diverse domains, we decided not to limit the background or experience of the subjects participating in the study. However, we do see the benefit of conducting a longer-term evaluation of VizDeck within the context of a particular domain and a much larger dataset to gain greater insights into the process of sensemaking in an applied context.

Conclusion

We introduced VizDeck, a novel web-based visualization tool for relational data that uses a card game metaphor and automatic visualization techniques to aid researchers in exploring datasets. We described the system implementation that allows the visualization thumbnails (or “vizlets”) to be produced and ranked within the dashboard. We then discussed an initial evaluation of VizDeck and three additional data visualization tools to gain greater quantitative and qualitative insights into users’ ability to navigate, compare, and quickly make decisions about data trends. Our quantitative evaluation found an automated approach particularly effective for aggregating results, comparing trends, and visually exploring a scientific dataset. A qualitative analysis yielded evidence of four types of user sensemaking patterns in rapid analysis of scientific data and demonstrated the viability of increasing the level of automation in visual analytics systems.

References

- Aragon, C. & Hearst, M. (2005). Improving aviation safety with information visualization: A flight simulation study. In *Proc. of CHI 2005*, ACM Press.
- Aherns, J., Geveci, B., & Law, C. (2005). ParaView: An End-User Tool for Large Data Visualization. In Hansen, C. & Johnson, C., editors, *The Visualization Handbook*, 717-731, Academic Press.
- Astronomical Collaborative Toolkit (ASCOT). Retrieved February 26, 2012 from <http://ssg.astro.washington.edu/research.shtml>
- Baptista, A.M., Howe, B., Freire, J., Maier, D., & Silva, C.T. (2008). Scientific Exploration in the Era of Ocean Observatories. *Computing in Science and Engineering* 10(3), 53-58.
- Bavoil, L., Callahan, S. P., Crossno, P. J., Freire, J., Scheidegger, C.E., Silva, C.T., & Vo, H. T. (2005). Vistrails: Enabling interactive multiple-view visualizations. In *IEEE Visualization*, 135–142.
- Dörk, M., Carpendale, S., Collins, C. & Williamson, C. (2008). Visgets: Coordinated visualizations for web-based information exploration and discovery. In *IEEE TVCG*, 14, 1205–1212.
- Elias, M. & Bezerianos, A. (2011). Exploration views: Understanding dashboard creation and customization for visualization novices. In *Proc. INTERACT'11*, 274-291.
- Elmeleegy, H., Ivan, A., Akkiraju, R., & Goodwin, R. (2008). Mashup advisor: A recommendation tool for mashup development. In *Proc. of the 2008 IEEE International Conference on Web Services*, 337–344.

-
- Ennals, R.J. & Garofalakis, M.N. Mashmaker: mashups for the masses. (2007). In *Proc. of the 2007 ACM SIGMOD international conference on Management of data*, 1116–1118.
- Few, S. (2009). *Now you see it: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press.
- Gonzalez, H., Halevy, A., Jensen, C. S., Langen, A., Madhavan, J., Shapley, R., & Shen, W. (2010). Google fusion tables: data management, integration and collaboration in the cloud. In *Proceedings of the 1st ACM symposium on Cloud computing*, 175-180. ACM.
- Gotz, D. & Wen, Z. (2009). Behavior-driven visualization recommendation. In *Proc. of the 14th international conference on intelligent user interfaces*, IUI, 315–324.
- Hey, T., Tansley, S., & Tolle, K. editors. (2009). *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research.
- Howe, B., Cole, G., Souroush, E., Koutris, P., Key, A., Khoussainova, N. & Battle, L. (2011). Database-as-a-service for long tail science, In *Proc. of the 23rd Scientific and Statistical Database Management Conference*, 480-489.
- The integrated ocean observing system. Retrieved September 15, 2011 from <http://www.ioos.gov/>
- Jankun-Kelly, T., Ma, K., & Gertz, M. (2002). A model for the visualization exploration process. In *IEEE Visualization*.
- Johnson, C. (2004). Top scientific visualization research problems. *IEEE Computer Graphics and Applications*, 24(4) 13–17.
- Kang, Y., Görg, C., & Stasko, J. (2009). Evaluating Visual Analytics Systems for Investigative Analysis: Deriving Design Principles from a Case Study. *IEEE VAST*, 139–146.
- Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5, 110–141.
- Mackinlay, J. D., Hanrahan, P., & Stolte, C. (2007). Show me: Automatic presentation for visual analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6), 1137-1144.
- National Institute for Standards and Technology (NIST). (2001). *Common Industry Format for Usability Test Reports, Version 2.0*. Gaithersburg: NIST.
- Perry, D. Aragon, A. Meier, A. Peffer, T. & Pritoni, M. (2011). Making Energy Savings Easier: Usability Metrics for Thermostats. *Journal of Usability Studies*, 6, 4, 226-224.
- Pirolli, P., Card, S.K. & Van Der Wege, M.M. (2001). Visual information foraging in a focus+context visualization, *Proc. SIGCHI*, 506-513.
- Russell, D.M., Stefik, M.J., Pirolli, P., & Card, S.K. (1993). The cost structure of sensemaking, in *ACM/IFIPS InterCHI '93 Conference on Human Factors in Software*, 269-276.
- Scholtz, J. (2006). Beyond usability: Evaluation of aspects of visual analytic environments. *IEEE VAST*, 145–150.
- SeaFlow. Retrieved on September 15, 2011 from <http://armbrustlab.ocean.washington.edu/resources/seaflow/>
- Sloan Digital Sky Survey. Retrieved on September 23, 2011 from <http://cas.sdss.org>
- Tufte ER. (1990). Small multiples. In: Tufte ER, editor. *Envisioning Information*, (67-80). Cheshire, Conn: Graphic Press.
- Viegas, F.B., Wattenberg, M., Van Ham, Kriss, F.J., & McKeon, M. (2007). Manyeyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 1121–1128.
- Ware, C. (2004). *Information visualization: perception for design*. Morgan Kaufmann Publisher.