Chart Constellations: Effective Chart Summarization for Collaborative and Multi-User Analyses

Shenyu Xu[*1], Chris Bryan[*1], Jianping Kelvin Li[*1], Jian Zhao[2], and Kwan-Liu Ma[1]

[1]University of California, Davis, USA
[2]FX Palo Alto Laboratory, Palo Alto, USA
[*] *These authors contributed equally to this work.*

Figure 1: *An analyst is using Constellations to investigate results generated by previous analysts. Constellations organizes these visualizations with projection and clustering. Adjusting the data coverage, encoding choice, and keywords sliders changes how pairwise chart similarities are scored and updates the projected layout and cluster groupings. Several charts are tagged to show how their positions change.*

Abstract
Many data problems in the real world are complex and require multiple analysts working together to uncover embedded insights by creating chart-driven data stories. How, as a subsequent analysis step, do we interpret and learn from these collections of charts? We present Chart Constellations, a system to interactively support a single analyst in the review and analysis of data stories created by other collaborative analysts. Instead of iterating through the individual charts for each data story, the analyst can project, cluster, filter, and connect results from all users in a meta-visualization approach. Constellations supports deriving summary insights about prior investigations and supports the exploration of new, unexplored regions in the dataset. To evaluate our system, we conduct a user study comparing it against data science notebooks. Results suggest that Constellations promotes the discovery of both broad and high-level insights, including theme and trend analysis, subjective evaluation, and hypothesis generation.

1. Introduction

Consider a large, complex dataset. If your task is to analyze this dataset, you might conduct exploratory visual analysis (EVA) by plotting several charts [JKMG07]. This iterative, chart-driven approach is popular and supported by software like Tableau [tab17] and by data science notebooks such as Kaggle Kernels [kag17].

Unfortunately, due to the depth or width of the dataset, such a task may be too much for one person.

Direct solutions to the problem of extracting interesting knowledge from these sorts of domains are not always obvious. An alternative approach is to have multiple analysts work to uncover insights, either as a team (i.e., performing collaborative sensemaking) or independently (each analyst conducts their own investigation).

But the question now becomes, *how then do we understand and learn from this analysis?*

We assume a meta-visualization approach (i.e., [PL13]), using visualizations to analyze and review the other visualizations created by prior analysts. We introduce the *Chart Constellations* system which supports EVA of visualization ensembles and data stories created by multiple users. Our tool is so-named because it projects visualizations as points (much as stars appear to us as points) into a two-dimensional subspace (much like our perspective of stars in outer space is flat) based on similarity. Visualizations that are arranged close together have similar characteristics (conversely, proximate stars are imagined into meaningful patterns).

Constellations is a fully interactive system, including multiple, coordinated interfaces and a backend server with data management components. In addition to providing an ordination-based view of charts which is interactively adjustable based on an introduced distance heuristic, the tool supports visual clustering, selection and filtering mechanisms to extract subsets of charts, inspection of individual charts, a workflow for suggesting new areas to explore in the dataset, and mechanisms for creating new charts.

We base the design of Constellations on a task analysis for reviewing EVA leading to continued investigations (i.e., creating new charts). Particularly in collaborative settings, this promotes a handoff approach for asynchronous data review and further exploration. In addition to the tool itself, we contribute design considerations and rationale for chart summarization and analysis. We conduct a two-part user study that compares our tool with the data notebook interface used by Kaggle, a popular website for publicly sourced data stories. The results provide insights into how enabling a collaborative view of multiple data stories encourages new thinking and different analysis behaviors. This means Constellations promotes both broader and higher-level insights about analyses by prior investigators.

2. Related Work

In this section, we review related work on modeling visualization similarity, supporting collaborative visual analysis and handoff, and current practices in using data science notebooks.

2.1. Modeling Visualization Similarity

When a set of visualizations is created to explore a dataset, the similarities and sequences between the visualizations provide important semantic information about the exploration process, such as the coverage and direction of the exploration. Multiple projects have proposed models for visualization similarity. Image Graphs [Ma99] use parameters in volume rendering to denote the changes between scientific visualizations. Visualizations are treated as nodes in a graph and the differences between parameters are represented as edges connecting the nodes. The P-Set Model [JKMG07] extends the idea of Image Graphs to information visualization, where parameter changes indicate the exploration process of using different visualization techniques. Similarly, Hullman et al. [HDR*13] use a directed graph model for representing the design space of visualization charts. GraphScape [KWHH17] proposes a generative model

for reasoning visualization similarity and sequencing. GraphScape represents visualization states as Vega-Lite [SMWH17] specifications and uses a sophisticated cost function for accounting changes in data transformations and visual encodings among visualization charts. Peltonen et al. [PL13] introduce a nonlinear embedding method for arranging an ensemble of scatter plots into a single projection. Since they only consider one visualization technique, their method determines similarity using dataset intersections only.

For Constellations, we adapt GraphScape's method for determining similarity between visualizations. In addition to data transformation and visual encoding, we extend the definition of similarity to include the application of SQL data queries and chart annotations (keywords) to obtain a more complete assessment and support collaborative visualization and analysis.

2.2. Supporting Collaborative Visual Analysis and Handoff

In the fields of information visualization and visual analytics, a key challenge is supporting collaborative visualization and analysis [CT05, IES*11]. Researchers in these fields have provided system design guidelines [WK06, VWVH*07, HA08, HVW07, MT14], software infrastructures [BE14, MBM*12, LCM15], and user behavior studies [ITC08, IFM*10, Rob08]. Like many of these, Constellations allows an analyst to explore by querying data, creating new visualizations, and attaching descriptive captions and labels.

Effectively summarizing and connecting findings is important for collaborative data analysis and visualization. Keel [Kee06] designed a visual analytics system that infers possible relationships among findings by analyzing their spatial and temporal organization in each team member's graphical interface. Lark [TIC09] offers a mechanism for integrating individual findings through an information visualization pipeline, allowing team members to build on each other's findings. Recently, Badam et al. [BZW*17] introduced an interface design that conveys each analyst's goals and activities for improving coordination within a team of analysts. In Constellations, we use visualization similarity to arrange exploration results and findings. In addition, we adapt the approach used by Sarvghad et al. [ST15] for exploiting analysis history to visualize explored dimensions using a treemap (see Section 4.2.1).

The concept of handoff, or the sharing and asynchronous review of others' work, is an overarching goal for Chart Constellations. The idea here is that one or more prior workers produce a set of output, which is then reviewed, synthesized, and used as input for future (i.e., deeper) tasks. Recent work in handoff has focused on continuing analysis based on partial, prior work. In particular, KT-Graph [ZGI*17] uses both analysis history and user-created externalizations to organize the results of previous investigations into a network graph of text notes and meta-labels. Handoff has also been studied in applications of information help desks [Sha08], product recommendations [SF09], and intelligence analysis [BCB10]. There is an inherent idea of collaborative, team-based investigation, even though the analysis is not simultaneously occurring. Constellations not only visualizes the analysis history of each analyst (i.e., their created charts) but also arranges the charts from multiple analysts into a single view. This allows an analyst to perform post hoc review of the work from all analysts who have created data stories.

2.3. The Rise of Data Science Notebooks

Constellations is also motivated by the rise of data science notebooks. Notebook IDEs are interactive computational environments that combine code execution, text processing, statistics, visualizations, and other rich media. Popular modern notebooks include Mathematica [mat17b], Project Jupyter [jup17], MATLAB Live Editor [mat17a], R Notebooks [rno17], and Kaggle Kernels [kag17]. Notebooks support coding in multiple languages, including MATLAB, Python, Julia, Scala, R, and SQL, and provide extensive tools for data wrangling. In part because they promote literate programming [Knu84], notebooks support a broad range of user tasks including research documentation, exploration and analysis, the running of statistical calculations, and being a means to create, publish, and share data stories.

Unfortunately, while modern data notebooks support collaborative sensemaking in that multiple analysts can work on shared notebooks and manually review each other's work, it is difficult to, as a post hoc or summary step, contextualize findings from several independently-created notebooks, especially when each data story is separately authored. Constellations can bridge this gap by providing simultaneous review of multiple data stories. For more discussion of the limitations of data notebooks as they pertain to Constellations, see Section 6.1 and the ensuing user study.

3. Task Analysis

Prior work has significantly shaped our motivation for this project. We want to enable flexible exploratory visual analysis (EVA) of prior-created visualizations, which can be highly complex and diverse, comprising many varied charts across different portions of data within a domain. Unfortunately, current systems are either too cumbersome, inefficient, unable to handle the complexity of the task, or (especially in the case of data notebooks) do not facilitate simultaneous multi-user analyses.

We aim to design a visual analytics tool for organizing an ensemble of visualizations and facilitating interactive exploration and analysis. Upon conducting this exploration and analysis, we might be inspired to undertake our own chart creations for further data exploration and/or storytelling. Based on these goals, we identify the following set of high-level tasks that such a system should satisfy. To contextualize *why* each task is necessary and *how* it can be accomplished, we reference the multi-level typology introduced by Brehmer and Munzner [BM13].

Task 1: Provide an ordination-based overview of all charts

| Why? | Produce + Discover → Explore → Summarize |
| How? | Derive + Encode + Arrange + Aggregate |
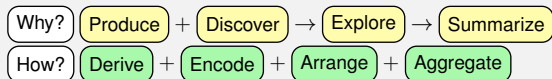
At a minimum, we must be able to review an ensemble of created charts. Data notebooks simply list charts by creation order for a single user. While this shows the temporal progression in a straightforward way, connections between different charts are difficult to identify. On the other hand, arranging similar charts together reveals valuable information about both the results and process of prior data exploration. Ordination is the process of ordering multivariate objects such that similar objects are near each other and dissimilar objects far from each other. Namely, dimensionality reduction techniques can spatially project data (in our case, visualizations) into two dimensions, thereby facilitating EVA.

Doing this enables the analyst to *discover* trends, patterns, and outliers by *exploring* the *summarized* ensemble. To accomplish this, we implement several functions. First, we *produce* a set of *derived*, pairwise distances between each chart (based on their similarities). These are *encoded* and *arranged* in a meaningful way (i.e., placing similar charts near each other). Discrete clusters of charts can be *aggregated* via grouping or binning.

Task 2: Extract a subset of charts from the overview

| Why? | Discover → Search → Compare + Summarize |
| How? | Filter + Select + Arrange + Change + Aggregate |
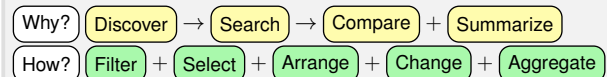
Instead of seeing all charts, we might want to review only a subset at a time. That is, during the *discovery* process, we want to *search* for charts that match specific criteria. These charts are individually *compared* to each other or viewed in *summary*.

This means our system should include functions for querying the charts. For charts shown in the overview (Task 1), we implement *filtering* to hide charts that do not meet desired constraints. Filtering also happens by proactively *selecting* a set of charts (such as with a lasso). The projected positions of charts are updated (i.e., their *arrangement* is *changed*) and new *aggregate* clusters or bins might be subsequently re-computed.

Task 3: Inspect individual charts

| Why? | Discover → Lookup → Identify |
| How? | Select + Encode |
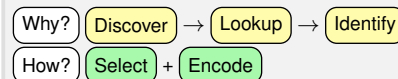
To see the specific charts created by users (i.e., to *discover* the information they contain), the system should enable individual charts to be reviewed at full resolution. That is, the analyst can *look up* or *identify* a single chart. *Selecting* the chart will display it (i.e., *encode* it) for inspection.

Task 4: Suggest areas for further exploration

| Why? | Discover → Explore → Identify |
| How? | Select + Navigate + Filter |
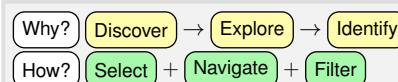
In addition to reviewing prior-created charts, we might want to *discover* unexplored (or underexplored) regions in the dataset and *explore* them. One way to enable this is by *identifying* attributes within the dataset that have thus far been not been highly used. We should therefore let the analyst *navigate* to and *select* dimensions based on their prior usages. In *filtering* the available dimensions to only show these unexplored regions, new avenues for future investigation will become apparent.

Task 5: Crew new charts

| Why? | Produce |
| How? | Select + Arrange + Aggregate + Encode + Record |
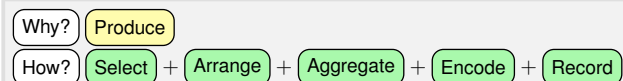
Finally, as a part of exploration, we might conduct our own investigation (i.e., create our own data story). The system should allow us to *produce* new charts. This is done by *selecting* (and optionally *aggregating*) a set of data points from the raw dataset, *arranging* them into a desired organization, *encoding* the data into a visualization, and then *recording* (i.e., saving) the created chart to the system for later review.

4. Constellations: Heuristics and System Design

Tasks 1–5 naturally lend themselves to a design that follows the Visual Information-Seeking Mantra: "*overview first, zoom and filter, details on demand*" [Shn96]. However, before describing Constellations' interface, we first quantify how to measure a set of charts based on the notion of "similarity."

4.1. Transforming Visualizations into Constellations

Given an ensemble of charts—each with its own encoding specifications, referenced data points, and descriptive meta-labels (title, keywords, caption, author, etc.)—how do we determine its similarity with other charts in a corpus?

Prior work has addressed this problem using specific subsets of the possible feature space. GraphScape [KWHH17] defines distances between visualizations based on differences in chart specification properties: visualization techniques, marks, channels, axis settings, etc. Word embedding models, such as word2vec [MCCD13], solely consider text elements but are quite computationally efficient when given text-based feature vectors. Information retrieval methods that only consider data points (i.e., [PL13]) define similarity based on dataset dimensional intersections. Colloquially, charts that visualize the same set of data points are similar. Unfortunately, pure dimensional intersection cannot handle new data outside of the specifically-referenced region nor can it quantify different visualization techniques.

Our approach is to use the outputs from these multiple heuristics. Specifically, the trio of aforementioned metrics—chart encodings, keyword taggings, and dimensional intersections—are combined into a single *aggregate pairwise distance* value for each pair of charts that we want to display in Constellations. This allows us to consider multiple facets of the charts that make up a data story.

Chart Encoding Distance. GraphScape [KWHH17] is a recently-introduced, directed graph model of the visualization design space based on chart encoding similarities. Graph nodes represent chart specifications and edge weights between nodes represent the cost to transform one chart into another. Given two charts, we calculate the encoding distance between them by summing the edge weights along the shortest path traversal between the charts according to GraphScape's graph model [gra17].

Unfortunately, GraphScape's algorithm for real-time calculation is quite expensive due to running a breadth-first search for shortest path finding. However, their code is open-source (see [gra17]) and includes a set of default, static transition costs for encodings that can quickly be added together to compare two charts. This is the approach our system currently uses.
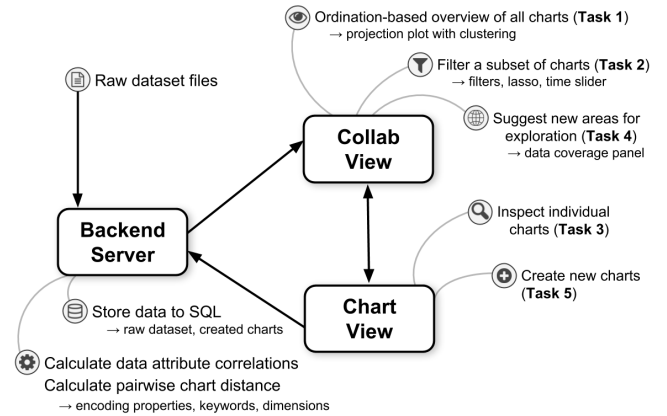

Figure 2: *An overview of the Constellations system, showing the three primary modules and their associated features. Arrows denote navigation or data transfer connections.*

Keyword Tagging Distance. word2vec [wor17a] is a predictive model for calculating vector representations of words (i.e., embeddings) from raw text. In Constellations, we allow for data story charts to have manually-annotated, descriptive keywords attached to them. We use word2vec (downloadable from [wor17b]) to obtain word vectors for the tagged keywords and then compute the pairwise cosine distances between the word vectors for each chart.

Dimensional Intersection Distance. We use a simple binary encoding to represent all the data dimensions as a vector. Each chart has its own vector, where the value of each dimension is 1 if the dimension is used for one of the chart encodings and 0 otherwise. With the vectors, we compute the dimensional intersection distance between each pair of charts.

Aggregate Pairwise Distance. After individually calculating distances for the chart encodings, keyword taggings, and dimensional intersections between each pair of charts, the distances are normalized along a [0,1] scale and combined with user-defined scaling weights in the following function:

$$dist(a,b) = w_{gs} * d_{gs}(a,b) + w_{ke} * d_{ke}(a,b) + w_{di} * d_{di}(a,b) \quad (1)$$

This computes the *aggregate pairwise distance* between two charts a and b. Here, d_{gs}, d_{ke}, and d_{di} are the distances for chart encodings, keyword taggings, and dimensional intersections, respectively, and w_{gs}, w_{ke}, and w_{di} are their relative scaling weights. The result is the overall pairwise distance between two charts.

An advantage of this aggregate approach is that new heuristics can easily be added or updated (as future work) to handle more aspects of the data stories. As an example, the system currently does not consider similarities between chart titles or captions (the only text elements it considers are keywords). Additional text embedding models could be used to quantify distances for these elements.

4.2. System Components in Constellations

Figure 2 shows the primary system components for Constellations. The Collab View is where visualizations are projected using ordination (Task 1) and explored via selection and filtering (Task 2); a data coverage panel shows past attribute usage and suggests new avenues for exploration (Task 4). The Chart View facilitates review
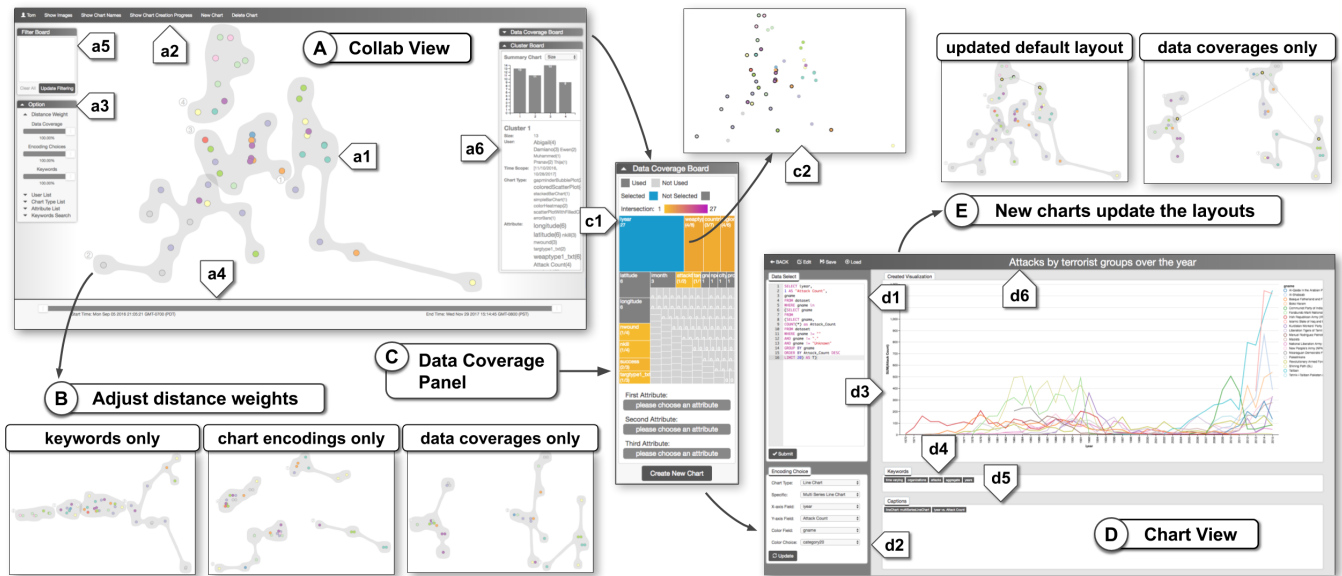
Figure 3: *The Constellations interface. (A) The Collab View projects visualizations as circles. (B) Updating distance weights adjusts the projected layout. (C) The data coverage panel allows for review of prior dimensional exploration, intersection, and correlation. (D) The Chart View facilitates inspection of existing charts and the creation of new ones. (E) Adding new charts updates the projection in the Collab View. See the case study in Section 5 for an explanation of label tags and arrows. Figure 9 in the Appendix shows a larger version of this image.*

of individual charts (Task 3) and the creation of new ones (Task 5). A backend server provides data storage and functions to calculate necessary statistical metrics.

4.2.1. Collab View: Enabling Overview and Analysis

The Collab View (Figure 3(A)) functions as the the main overview. Based on the aggregate pairwise distances, the ensemble of prior-created charts are spatially positioned using dimensionality reduction (a1). Each chart is plotted with a circle mark and uses hue to denote its creation author (using a categorical palette from ColorBrewer [HB03]). A user can toggle between two projection techniques: multidimensional scaling (MDS) [KW78] and t-SNE [MH08]. For simplicity, we use MDS projections in this paper—primarily because it results in a deterministic layout which allows us to make the user study experience more consistent.

Labeling options are available in a top menu (a2), including author polylines: directed arrows between an author's charts that show the creation order. In the left side of the view (a3), the options board contains a set of three distance weight sliders and several filtering options. Adjusting the sliders modifies the scaling weights used in Equation 1, recalculating the aggregate pairwise distances for all charts and updating the projection layout (Figure 3(B), see also Figure 1). Applying a filter removes a subset of charts based on the desired constraint—for example, only displaying charts by a certain user. When filters are applied, a descriptive label is created in the filter board (a5). At the bottom, a timeline slider allows for temporal filtering (a4); a user can also draw a lasso selection directly on the projection to select a subset of charts.

Clustering is used to bin charts into discrete groups. Two metrics are available: *k*-means clustering [M*67] and affinity propagation [FD07]. Clustering can be based on either the aggregate pair-

wise distances between charts or by their projected x/y coordinates. When the projection updates, such as via filtering or adjusting the distance weight sliders, clusters are recalculated. Cluster hulls are drawn using bubble sets [CPC09]. At right, the cluster board shows statistics about the currently-drawn clusters (a6).

Above this is the data coverage panel (Figure 3(C)), which contains a treemap showing the available attributes in the dataset. Each block corresponds to an attribute and is colored according to whether or not it is used by any charts in the projection.

Selecting an attribute in the data coverage panel highlights it blue (c1) and highlights any intersecting dimensions orange. An intersecting dimension is one that co-occurs in at least one chart with the currently selected attribute. The luminance of the orange blocks denotes co-occurrence amount. Below the treemap is a set of dropdown menus. These show the Pearson correlations of attribute pairs, allowing a user to discover the global relations of attributes. By using the treemap to explicitly show the "beaten path" of dimensional analysis that has thus far been conducted along with the global attribute correlations, we provide a user with information about unexplored (or underexplored) regions in the dataset. This promotes the targeted investigation of new areas via the creation of new charts and data stories.

4.2.2. Chart View: Inspect and Create New Charts

The Chart View, shown in Figure 2(D), allows for the full-scale inspection of existing charts and provides functionality to create new ones. To create a new chart, a user enters an SQL query (d1) to retrieve data from the backend server and sets the chart encodings using a set of dropdown menus (d2) to create a visualization (d3). We use Vega-Lite to define encoding markups and render the chart. Custom, user-defined keywords, caption, and title can be manually

added (d4, d5, d6). When a new chart is created, the ensemble of charts available in the Collab View updates (Figure 3(E)).

4.2.3. Backend Server: Data Management and Services

The backend server is built using Node.js and consists of services to manage and process the data. Raw dataset files are parsed and stored on the server as a MySQL instance. When a new chart is created in the Chart View, its Vega-Lite specifications are saved to the database. This kicks off a script that updates the chart encoding distances, keyword tagging distances, and dimensional intersection distances (Section 4.1) and stores them in a second table. The backend server also stores the Pearson correlations used in the data coverages panel.

5. Case Scenario: Major Tom Among the Stars

The purpose of Constellations is to support sensemaking and the undertaking of new data explorations in collaborative and handoff scenarios. Thus, a user needs to understand the prior work, and then use this knowledge to pursue further investigation. To illustrate how this process can work, this section describes a case scenario using Constellations in a team-based environment.

5.1. Dataset

We collected several open datasets and data stories from Kaggle, including the Global Terrorism Dataset (GTD) [glo17]. This is a large (~130 MB), multidimensional (135 dimensions), geographic dataset stored in a tabular format (CSV file), cataloging more than 170,000 terrorist attacks worldwide from 1970–2016. Popular datasets such as the GTD have hundreds of associated, publicly-available data stories.

We recreated the data stories of 11 users (47 charts total) in our system as closely as possible. Our reasoning for this chart ensemble size is that we want to emulate a team of analysts in this case scenario (we felt that 11 was a reasonable count), and in our review of data stories on Kaggle we noticed that most site users created data stories with between 3 and 6 charts. In addition, due to the perceptual scalability of hues used to color projected chart circles in the Collab View, we are limited in the number of available values (see Section 7.4). Note that Kaggle does not provide a keyword or hashtagging functionality; we manually add these by reviewing each data story's title, chart captions, and descriptions.

5.2. Use Case Scenario

With the above ensemble set of visualizations, we now describe a use case scenario for Constellations. While in reality these data stories are primarily created by independent, non-collaborating users, we assume here that our set of charts have been created by an analytics team under the guidance of Major Tom, the group leader tasked with sensemaking and decision-making.

Task and Setup. Major Tom is the leader of a team of 11 analysts that investigate historical terrorism events around planet Earth, but he has been preoccupied for the last several days and is just now available to review his team's progress. His task is to review the

investigations done by his team into the GTD and, if necessary, perform new investigation. Figure 3(A)-(E) demonstrates his steps.

Part One: Reviewing Prior Work. (A) Major Tom logs into Constellations and loads the GTD. (a1) This provides an overview of his analysts' prior-created charts. Hovering on chart circles pops up preview tooltips that provide quick review of the charts. (a2) This can also be accommodated by clicking the "Show Charts" label in the top bar, which shows a small image preview of each projected chart. (a3, a4) By checking names in the options board and dragging the timeline slider, Major Tom filters to only display charts created by a subset of his team and within a specific time period. (a5) Currently loaded filters are shown in the filter board. (a6) He also uses the cluster board to review information about the binned charts groups.

(a3) Major Tom next analyzes the charts based on various aggregate pairwise distance settings by adjusting the distance weight sliders. (B) This re-runs Equation 1 with new weights for each distance metric and re-projects the charts using MDS. He tests out three settings: keywords only, chart encodings only, and data coverages only. In particular, the "keywords only projection" shows a tight, centrally-located grouping of charts. His team tagged these charts with closely related keywords.

(C) Major Tom opens the data coverage board to investigate which dimensions have thus far been explored. (c1) The treemap shows that iyear is the most popular attribute: its rectangle is the largest with 27 total charts using this dimension. Clicking on iyear highlights it blue; other rectangles are colored shades of orange according to how often they co-occur with iyear. (c2) Charts in the projection that use iyear are highlighted.

Part Two: Creating New Charts. At this point, Major Tom has a sense of his team's prior work. He now wants to create new charts that explore the dataset in new ways. (c1) Using the data coverage panel, he sees that gname (the "group name" behind a terrorist event) is only used in one chart and not with iyear at all. However, gname seems important, so Major Tom decides it is worth investigating further.

(D) Clicking the "Create New Chart" button in the data coverage panel takes Major Tom to the Chart View. (d1) He writes a SQL command in the data selection panel to retrieve the preselected attributes. (d2) He then selects the desired encoding choices and (d3) creates a line chart visualizing gname over time. (d4, d5, d6) After setting the keywords, caption, and title, he saves this chart and creates two subsequent charts that spatially plot and aggregate gname in other ways.

(E) After creating this trio of charts, he returns to the Collab View. His charts are added to the ensemble—now 50 total charts by 12 users. The updated default layout is quite similar to the prior version (in (a1)), but adjusting the distance weight sliders to use "data coverages only" provides a noticeably changed layout since the new charts combine data dimensions in novel ways.

6. Comparative User Study

Tasks 1–5 in Section 3 represent the high-level functionalities that Constellations is intended to support. To better understand the specific insights and user engagement behaviors that are supported,

we conducted a controlled user study that compares Constellations with Kaggle's data notebook interface.

6.1. Baseline Condition: Kaggle Kernels

Kernels is Kaggle's browser-based, data notebook interface for creating data stories [kag17]. Currently, Kaggle hosts over 5,600 open datasets, contains over 250,000 Kernels, and has more than a million registered users [Gol17], making it the *de facto* standard for online data notebooks. Kernels is officially described as a "*collaboration product*" [Mon16], but unfortunately the Kaggle website only allows for the selection and review of one Kernel at a time. This provides a real-world motivation for why a tool such as Constellations has value: it lets us simultaneously review and synthesize a set of Kernels created by multiple users for a shared dataset.

To support our study, we extended and implemented a Kernel interface which, for a set of users, presents their charts in a data notebook format (see the Appendix for a screenshot). While a direct comparison between Constellations and Kernels may seem unfair at first, we note that there is no current solution for ensemble review of data stories on Kaggle. This is despite the fact that the Kaggle website describes Kernels as "*the best place to explore data science results and share your own work*" [kag17].

6.2. Study Design and Procedure

We employed a between-subject design in our study; Figure 4 shows the outline. Each participant was first assigned to a group— either Kernels or Constellations—which was used for the duration of the session. Since the focus is reviewing prior work done by others, we disabled the chart creation functionality, though the Chart View page was still usable for inspection.

Training Stage. Each participant was given a hands-on tutorial for the assigned interface. An administrator explained system features and interactions. Participants could then ask questions and play around with the system until they felt ready to proceed.

During training, participants viewed the Company HR dataset from Kaggle (shown in Figure 1) which contains 26 charts from 5 users [hra17]. For the Task and Freeform Stages, we use the 47 GTD charts created by 11 users described in Section 5.1.

Task Stage. In this stage, we measure how efficiently each interface supports a trio of specific investigative tasks. Each requires the user to look at charts in at least two data stories.

(t1) Idiom counting: List the three most commonly used chart types in descending order.
(t2) Dimensional intersection: Give the name of the data dimension that is most often paired with <DimensionA>.
(t3) User comparison: Between the charts made by <UserA> and <UserB>, which of the two charts are the most similar based on chart encodings?

Compared to the higher-level Tasks 1–5 described in Section 3, t1–t3 represent specific, low-level insights about a data story ensemble that an analyst might commonly wish to realize: (t1) identifying popular visual encodings, (t2) identifying common attribute pairs, and (t3) understanding where the analysis of two users is
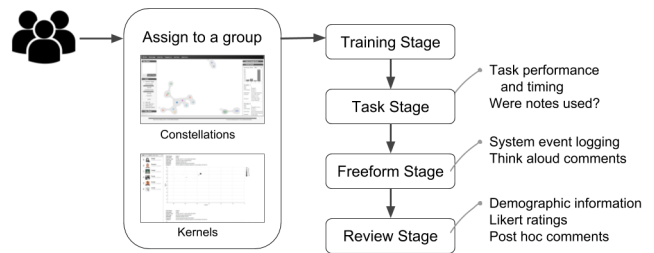

Figure 4: *The user study flow. Participants were assigned either the Kernels or Constellations interface. For each stage, we note the data points collected for analysis.*

most similar. These low-level tasks are performed in service of the higher-level Tasks 1–5, and are specifically designed to be suitable for direct performance testing.

Task order was consistent. The participant was given a sheet of paper with the task description at the top. To begin a task, the participant clicked a "Start Task" label in the interface. When finished, the user clicked the label again (stopping a system timer) and verbally stated the answer to the administrator. The task description sheet was available as an optional scratch pad. After finishing t1–t3, the subject progressed to the Freeform State.

Freeform Analysis Stage. In this stage, participants conducted an undirected, freeform analysis and review—there was no explicit "answer the question" task. To motivate this stage, we put the participants into a scenario: They were now the team lead for a group of analysts. As the leader, their mission was twofold: summarize the investigated dataset and understand the insights and analyses performed by the team. Participants were given ten minutes to use the tool for this review. While working, participants used think aloud protocol to describe their cognitive processes and actions [FKG93].

Review Stage. To conclude the study, participants completed a short questionnaire. This collected demographic information and queried the perceived usefulness of interface features using a Likert scale (1 – strongly disagree, 7 – strongly agree). Participants were also allowed to give any desired commentary, suggestions, and/or criticisms about their experience.

6.3. Participant Recruitment and Apparatus

We recruited a total of 20 students at the University of California, Davis (age $\mu = 26.45$ years, $\sigma = 4.46$ years, 3 females). Because study subjects had to pose as "analysts" in the Freeform Stage, we recruited both undergraduate and graduate students from computer science who had experience with visualization design and/or data analysis. Figure 8(P1) lists the familiarities of participants with regards to reading and interpreting visualizations: both interfaces had similarly experienced users. All participants were proficient in English; one was vaguely familiar with the Terrorism dataset (though not at a level that was considered confounding). Two others had created Kernels before (each less than five), though both were in the Constellation group.

The hardware apparatus was a 27-inch monitor (Apple Thunderbolt display with 2560×1440 resolution) connected to a MacBook
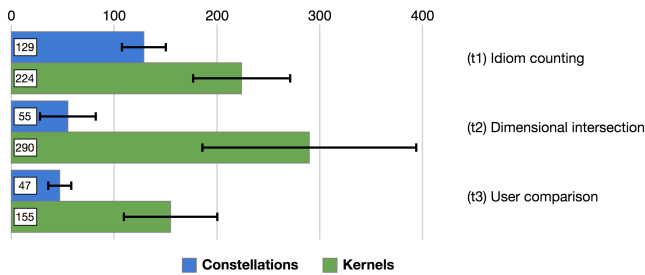
Figure 5: *Completion time for tasks t1–t3 in the Task Stage. Bars show mean completion time in seconds; error bars indicate 95% confidence intervals. For each task, Constellations is significantly faster than Kernels ($p < 0.005$).*

Figure 6: *Plots of participant investigation patterns during the Freeform Stage. The x-axis represents time in minutes (10 minute limit). Numbers on the right indicate total counts of the actions.*

Pro running macOS Sierra with mouse and keyboard. The interfaces were presented using Google Chrome in full-screen mode. Quicktime Player recorded both audio and screen capture.

6.4. Study Results and Discussions

Where applicable, we report Mann-Whitney U tests to indicate if there is a statistically significant difference between Constellations and Kaggle (using a threshold of $p = 0.05$) by providing the U and p values.

Overall, sessions generally lasted between 40–60 minutes. Average session time for Constellations ($\mu = 58.6$, $\sigma = 8.74$) was significantly longer than for Kaggle ($\mu = 42.7$, $\sigma = 8.31$): $U = 9, p < .005$. One reason for the time difference is that training was much longer for Constellations ($\mu = 22.8$) than Kaggle ($\mu = 4.95$). As Constellations is a more complex system with richer interactions, this result is not surprising.

6.4.1. Task Stage Performance: Constellations is Faster

We measure performance for t1–t3 in the Task Stage using two primary data points: response correctness and timing. For both interfaces, subjects had high correctness. All Constellations subjects answered t1–t3 correctly, and for Kernels subjects, t1 and t2 each had one incorrect response and t3 two incorrect responses.

Figure 5 shows how long it took to complete each task. In terms of speed, Constellations enabled significantly faster analysis than Kernels for each task: t1: $U = 10, p < .005$, t2: $U = 1, p < .0005$, t3: $U = 6, p < .005$. This is not a surprising result to us, as these tasks are designed to force browsing across the work of multiple users. Due to Kernels' linear presentation order, this results in a more tedious, iterative scanning behavior.

The third data point from the Task Stage is the answer to the question, *how many subjects jotted notes on their sketchpads while completing each task?* For Kernels, the answers are t1 = 9, t2 = 10, and t3 = 4. For Constellation, the breakdown is t1 = 7, t2 = 1, and t3 = 0. Put another way, Kernels required manual note taking at a much higher rate than Constellation, especially for t2 and t3.

6.4.2. Interaction Logging Shows Users Behavior Pattens

We captured participant actions during the Freeform Stage via server-side logging (Figure 6). Kernels is a simpler system and really only has two primary interactions: clicking a user in the sidebar
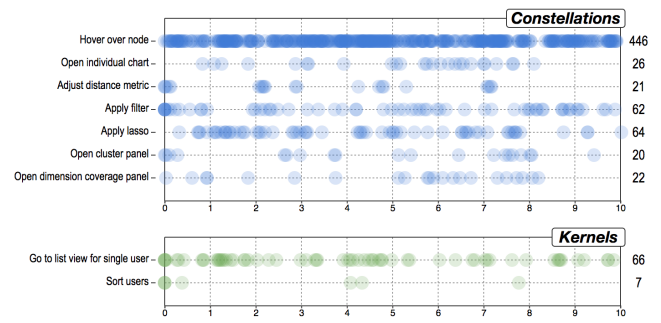
to load his/her data story, and sorting the users in the sidebar (only done 7 total times).

Constellations has a more complex user experience. Figure 6 lists commonly-invoked actions. Hovering on a projected chart node (and showing its tooltip) was the most popular event, occurring 446 times. Opening and inspecting individual charts at full resolution was a relatively rare occurrence (26 times).

Lassos (64) and sidebar/timestep filtering (62), which removed charts from the display, were more common than adjusting the distance metric sliders (21). Toggling the cluster (21) and data coverage board (22) was performed a similar number of times. Interestingly, interactions with the data coverage panel tended to be later in the investigation while cluster board interactions were distributed throughout. Possibly, this is because participants wanted to review the projected charts first and opening the cluster board toggled the cluster bubbles visibility.

6.4.3. Freeform Commentary: Constellations supports High-Level Insights

We are interested in seeing how Constellations and Kernels facilitate and encourage new insights, such as realization of factoids and generation of hypotheses. Compared to Kernels, we believe Constellations will promote both deeper and broader knowledge acquisition. To formalize this, we list out eleven specific insights that we think either one or both of the interfaces will promote (Table 1).

To define this list of insights, we reflected on the specific types of educational outcomes that these systems should support. In particular, we use Bloom's Taxonomy [AS94] as a guide to hierarchically classify the insights into tiers based on complexity and specificity: low- or view-level insights (basic facts and comprehension), sensemaking insights (analysis and synthesis at a global or temporal level), and subjective judgments (generating hypotheses and evaluating quality).

The think aloud comments from the Freeform Stage act as a data corpus for deriving participant insights. A curator reviewed the screencasts of participants and categorized applicable spoken statements into insights. That is, each time a phrase was said that could be construed as a new learning or realization, it was categorized by the type of insight it represented. It's important to note that, while we recognize the potential for bias in this sort of data collection, the majority of statements fell clearly into a single insight category.

Category		Insight	Example Usage
View-level insights	(i1)	Review information for a single chart	*"Here they're looking at the number of attacks in Iraq specifically."*
	(i2)	Explore a subset of charts via selection	*"[Applies filter] This guy only used bar charts."*
	(i3)	Directly compare two charts	*"UserA and UserB came up with similar views, these are both line charts"*
Sensemaking insights	(i4)	Analyzing high-level patterns and trends	*"It seems like most users focused on specific analysis [areas], and don't diverge."*
	(i5)	Defining groupings, clusters, outliers	*"People always make the same type of chart, like these [points at cluster bubble]"*
	(i6)	Data coverage insights	*"Basically, most people want to see attack count with other attributes."*
	(i7)	Temporal-based or storytelling insights	*"First UserA looked at the world, and now he narrowed it down to India."*
Subjective judgments	(i8)	Asks hypothesis about individual charts	*"My follow-up question for these types of charts is..."*
	(i9)	Asks hypothesis at a global level	*"I wonder why no user is looking at wounded, spatial, and weapon combinations?"*
	(i10)	Evaluation about individual charts	*"I'm not really sure if I believe their analysis based on just this one chart."*
	(i11)	Evaluation about users	*"I don't like UserA's graphs as much, they don't seem as informative."*

Table 1: *This table lists the types of insights that study participants discovered during freeform analysis. Insights are loosely categorized by their objectives. For each, we include an example statement said by a participant during his/her session.*
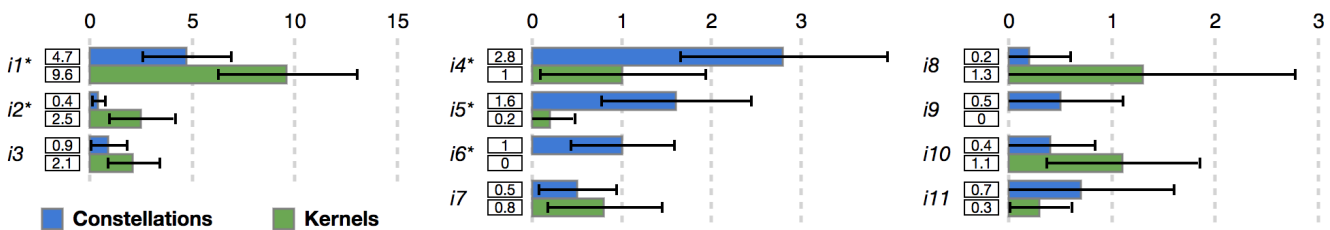

Figure 7: *The average number of insights discovered by a participant during the Freeform Stage. Error bars indicate 95% confidence intervals. Label asterisks indicate a statistical difference of $p < 0.05$ between Constellations and Kernels for that insight.*

For ambiguous statements (those that could potentially be placed in multiple categories), the curator tried to be as fair and consistent as possible across all subjects, so that all insights for both interfaces were binned according to the same evaluation standards.

Figure 7 shows the average number of insights (for each insight type) generated by participants for the two interfaces. For each insight type, we label if there is a significant difference between Constellations and Kernels. Interestingly, there are more insights realized on average using Kernels (18.9 vs. 13.7) though this was not found to be a statistically significant difference.

More than 75% of Kernels' insights (14.2) were view-level (i1, i2, i3). Since Kernels shows a list of charts, this localized knowledge acquisition makes sense. This behavior also applies to sensemaking insights ((i7), where Kernels subjects would note on the chart progression within a single data story but not between users or at a global level. For subjective judgments, we likewise tend to see hypotheses and evaluations only about individual charts (i8, i10).

In contrast, Constellations had much better insight ratios for higher-level, sensemaking insights (i4, i5, i6) and judgments (i9, i11) that required multi-user or collaborative awareness. For example, Kernels had no data coverage insights (i6) nor global hypotheses (i10). This illustrates a key takeaway between these interfaces: Kernels emphasizes linear, focused, and iterative processing, while Constellations promotes more open-ended, global, and complex understanding.

6.4.4. Survey Ratings and Criticisms

Figure 8 lists ratings for Constellation and Kernels from the Review Stage questionnaire. Both systems were rated as easy to learn and use (G1, G2) and were seen as effective for reviewing the ensemble of data stories (S1, S5). Constellations rates higher at a statistically significant level for several system functionalities: facilitating comparisons (S2), showing high-level trends and patterns (S3), and organizing charts into meaningful groupings (S4). Kernels rates higher at showing the temporal ordering of data stories (S6). Regarding the perceived usefulness to personal investigation strategies, Kernels rates higher at reviewing individual charts (C1), but Constellations promotes participants to think about filtering (in Kernels this was done by switching between a "Show All" list option) (C2). Other strategies (C3–C6) have statistically similar scores. Kernels provides no methods for viewing data coverages or chart clusterings, but these are seen as important to user investigation in Constellations (C7, C8). Of the other interface features that are specific to Constellations (F1–F7), all are rated positively. The lowest, adjusting distance weights (F1), was rarely done in the Freeform Stage (only 21 times, see Figure 6).

At the end of the questionnaire, we solicited general feedback about the system. For Constellations, there were three notable system requests, each put forth by at least two subjects:

(1) **Add a chart comparison panel** allowing two charts to be displayed together. Our system did not have this sort of functionality; to compare charts, a user had to hover back and forth between charts to show their tooltips.

(2) Provide more intuitive and **meaningful explanation of cluster groupings.** While the cluster board lists statistics and data types for each cluster, subjects wanted a less text-heavy summary. One
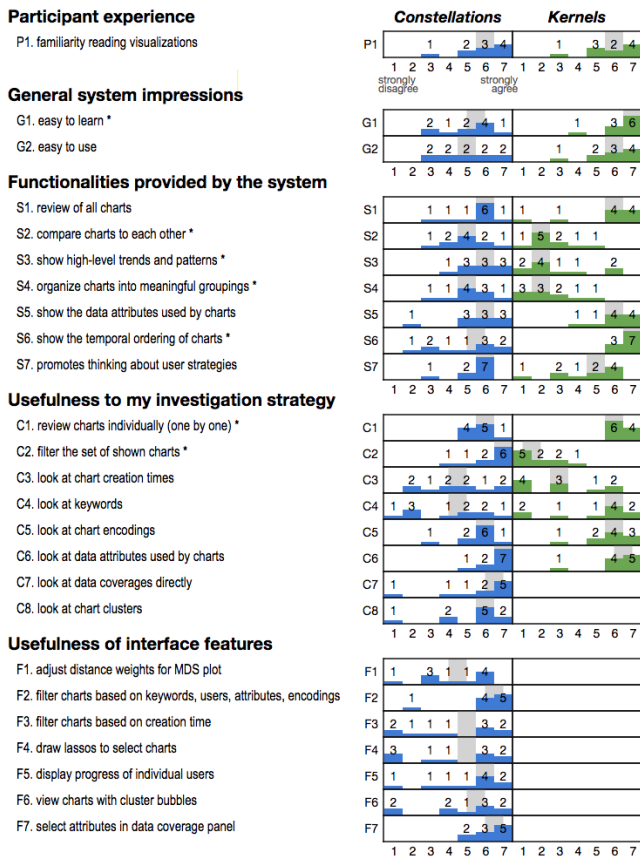
Participant experience

P1. familiarity reading visualizations

General system impressions

G1. easy to learn *
G2. easy to use

Functionalities provided by the system

S1. review of all charts
S2. compare charts to each other *
S3. show high-level trends and patterns *
S4. organize charts into meaningful groupings *
S5. show the data attributes used by charts
S6. show the temporal ordering of charts *
S7. promotes thinking about user strategies

Usefulness to my investigation strategy

C1. review charts individually (one by one) *
C2. filter the set of shown charts *
C3. look at chart creation times
C4. look at keywords
C5. look at chart encodings
C6. look at data attributes used by charts
C7. look at data coverages directly
C8. look at chart clusters

Usefulness of interface features

F1. adjust distance weights for MDS plot
F2. filter charts based on keywords, users, attributes, encodings
F3. filter charts based on creation time
F4. draw lassos to select charts
F5. display progress of individual users
F6. view charts with cluster bubbles
F7. select attributes in data coverage panel

Figure 8: *Participants' ratings about various system aspects during the Review Stage. Median ratings are indicated in gray. Asterisks indicate a statistical difference of $p < 0.05$ between Constellations and Kernels for that system aspect.*

participant suggested encoding homogeneity or stability metrics onto the clusters to help show their semantics.

(3) Two subjects wanted to **track the thought processes of the individual users** by explicitly arranging their charts temporally. The polyline functionality in our system was considered too rudimentary for this task. In contrast, the Kernels interface is well suited to this type of presentation.

7. Discussion

In this section, we discuss the key takeaways from our study, propose a new Task 6 that our system supports, and discuss the limitations and future directions of the current Constellations system.

7.1. Observing Participant Strategies

Based on a review of session screencasts, think aloud commentaries, and system logging events, we identify several strategies that study participants employed to analyze charts.

Kernel Strategies. The ten Kernels participants employed an iterative technique for analyzing charts. They each would select a data story and then linearly review its charts in order. Occasionally, they might go back to see an earlier chart (such as if two charts showed

similar but not equivalent plots), but this was the only "breaking" from the iterative exploration.

Constellations Strategies. In contrast, a wide set of strategies were employed by participants using Constellations. Subjects commonly switched strategies several times during investigation:

- **Cluster surveying.** A common initial action was to scan over the charts within each cluster, using the tooltip to quickly highlight individual charts.
- **Selective filtering.** Participants would regularly filter charts based on 1–2 specific constraints—for example, filtering by specific users or by chart types (i.e., only bar charts). Lassos were used to select groups of spatially co-located charts.
- **Temporal and polyline review.** Some subjects adjusted the time slider to filter charts by creation date. Other subjects used author polylines to show chart creation order. The polyline functionality was used to find which analyst had explored the "farthest" or had crossed multiple clusters with created charts.
- **Dimensional investigation.** Using the data coverage panel, a few subjects investigated data intersections. This usually happened later in their analysis.

Interestingly, the distance weight sliders were not modified with regularity (see Figure 6); our impression is that most participants trusted the default similarity scores. Notably, one Constellations user employed our system in an unexpected way—investigating the "impact" that chart authors had on the projection (see Section 7.3).

7.2. Study Takeaways and and Lessons Learned

Overall, the findings in the user study support the notion that Constellations promotes both *broad* and *high-level* insight in EVA. In Constellations, users tended to view charts through a time-agnostic lens; for Kernels, this was the primary method of investigation. This brings an interesting takeaway: *Constellations and Kernels can be combined.* There is no constraint that prevents a data notebook interface from being integrated with Constellations. This can provide a "best of both worlds" solution that leverages the complementary strengths of each interface.
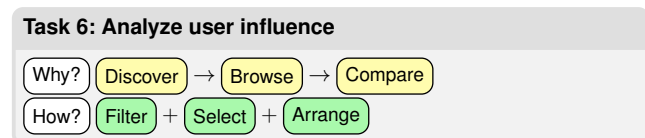
While our evaluation focuses on the review of data stories for a single dataset (GTD), we have tested ensembles of data stories from several Kaggle datasets. It is also worthwhile to test *non*-Kaggle datasets, especially ones created in truly collaborative, team-based environments. While our heuristics give good results thus far, it's possible that data stories created in different (i.e., real-world hand-off) scenarios will show different characteristics.

We also note an important study limitation: our population is familiar with visualization and the concept of ordination. For non-experts, such concepts may be more difficult to intuitively grasp. Future evaluations can assess the efficacy of Constellations in lay user contexts.

7.3. Serendipitously Supporting an Unexpected Task

One study participant used the projection plot in an unexpected manner, subverting our original intentions. Instead of investigating the GTD's data stories directly, he wanted to explore the influence

of users on the projection layout. He would remove one user at a time and observe how the projection updated. His intuition was that less important users would only cause a small update to the layout. However, if a user created "important" charts (or created many charts), removing the user's charts would cause a large update to the projection. For our part, we did not consider this task when designing our system, even though it is (inadvertently) supported by our implementation. Based on this finding, we formalize a Task 6:

> **Task 6: Analyze user influence**
>
> Why? [Discover] → [Browse] → [Compare]
> How? [Filter] + [Select] + [Arrange]

The *how* for this process is similar to Task 2 ("extract a subset of charts for analysis"), as the analyst simply *filters* the display by *selecting* constraints: "user" in this case. The resultant plot removes the selected user's charts and updates the projection's *arrangement*. The difference between Task 6 and Task 2 comes in the *why*: the analyst is *discovering* how the projection updates its layout by *browsing* and *comparing* the new layout to its prior state.

7.4. Current System Limitations and Future Directions

Projecting charts by similarity is useful when we have little knowledge about the underlying dataset or its ensemble of data stories. We believe Constellations adeptly supports this type of freeform, exploratory analysis. When we consider scenarios like hypothesis verification and focused analysis, then projecting charts with a notion of axial relevance can lead to more coherent insights as opposed to dimension-free techniques like MDS and t-SNE. An example of this is the InterAxis system [KCPE16], which projects points values as a linear combination of data attributes bound to an axis. To integrate the InterAxis approach in Constellations, if a user decided some visualizations were important, these charts could be placed along an axis in the projection plot. More relevant charts would move closer to the anchor points and less relevant charts farther away.

From a software engineering perspective, there are several improvements that can be made. One current system limitation is the use of hues to denote chart authors in the Collab View (Figure 3(a1)). Humans have difficulty accurately comparing colors as the number of categories scales up [Hea96], limiting the number of data stories that can currently be loaded into the system.

The heuristics discussed in Section 4.1 can be updated and augmented with more refined or advanced distance methods. A stronger focus on text-based clustering (using titles and captions) could help better reveal charts with similar semantic insights. Likewise, some users expressed interest in a more powerful temporal view of the charts to better follow the reasoning processes of prior analysts. Then they could verify, for instance, whether prior users often start investigations with similar kinds of charts.

8. Conclusion

In this paper, we present Chart Constellations to support metavisual analysis by effectively summarizing the results of prior ana-

lysts who have created data stories about a shared dataset. Constellations provides various statistical heuristics to organize and project results into a single view; visualizations containing related insights are placed closer to each other. Additionally, Constellations provides methods to identify unexplored (or underexplored) subspaces of the data, promoting targeted exploration into new regions.

Results of a user study show that our system is more effective than a conventional data notebook interface (Kaggle Kernels) in gaining high-level insights from a set of visualization results as well as understanding the exploration process. The user study additionally indicates that effectively summarizing a set of different visualizations is beneficial to collaborative visual analysis.

9. Acknowledgement

References

[AS94] ANDERSON L. W., SOSNIAK L. A.: *Bloom's taxonomy*. University of Chicago Press, 1994. 8

[BCB10] BIER E. A., CARD S. K., BODNAR J. W.: Principles and tools for collaborative entity-based intelligence analysis. *IEEE Transactions on Visualization and Computer Graphics 16*, 2 (2010), 178–191. doi:http://dx.doi.org/10.1109/TVCG.2009.104. 2

[BE14] BADAM S. K., ELMQVIST N.: Polychrome: A cross-device framework for collaborative web visualization. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (2014), ACM, pp. 109–118. 2

[BM13] BREHMER M., MUNZNER T.: A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2376–2385. 3

[BZW*17] BADAM S. K., ZENG Z., WALL E., ENDERT A., ELMQVIST N.: Supporting team-first visual analytics through group activity representations. In *Proceedings of Graphics Interface Conference* (2017). 2

[CPC09] COLLINS C., PENN G., CARPENDALE S.: Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1009–1016. 5

[CT05] COOK K. A., THOMAS J. J.: *Illuminating the path: The research and development agenda for visual analytics*. National Visualization and Analytics Ctr, 2005. 2

[FD07] FREY B. J., DUECK D.: Clustering by passing messages between data points. *Science 315*, 5814 (2007), 972–976. 5

[FKG93] FONTEYN M. E., KUIPERS B., GROBE S. J.: A description of think aloud method and protocol analysis. *Qualitative Health Research 3*, 4 (1993), 430–441. 7

[glo17] Global terrorism dataset | kaggle, 2017. URL: https://www.kaggle.com/START-UMD/gtd. 6

[Gol17] GOLDBLOOM A.: We've passed 1 million members | the official blog of kaggle.com, 2017. URL: http://blog.kaggle.com/2017/06/06/weve-passed-1-million-members/. 7

[gra17] Github | uwdata/graphscape, 2017. URL: https://github.com/uwdata/graphscape. 4

[HA08] HEER J., AGRAWALA M.: Design considerations for collaborative visual analytics. *Information visualization 7*, 1 (2008), 49–62. 2

[HB03] HARROWER M., BREWER C. A.: Colorbrewer. org: an online tool for selecting colour schemes for maps. *The Cartographic Journal 40*, 1 (2003), 27–37. 5

[HDR*13] HULLMAN J., DRUCKER S., RICHE N. H., LEE B., FISHER D., ADAR E.: A deeper understanding of sequence in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2406–2415. 2

[Hea96] HEALEY C. G.: Choosing effective colours for data visualization. In *Proceedings of the 7th Conference on Visualization'96* (1996), IEEE Computer Society Press, pp. 263–ff. 11

[hra17] Human resource analytics | kaggle, 2017. URL: https://www.kaggle.com/ludobenistant/hr-analytics. 7

[HVW07] HEER J., VIÉGAS F. B., WATTENBERG M.: Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), ACM, pp. 1029–1038. 2

[IES*11] ISENBERG P., ELMQVIST N., SCHOLTZ J., CERNEA D., MA K.-L., HAGEN H.: Collaborative visualization: definition, challenges, and research agenda. *Information Visualization 10*, 4 (2011), 310–326. 2

[IFM*10] ISENBERG P., FISHER D., MORRIS M. R., INKPEN K., CZERWINSKI M.: An exploratory study of co-located collaborative visual analytics around a tabletop display. In *IEEE Symposium on Visual Analytics Science and Technology* (2010), IEEE, pp. 179–186. 2

[ITC08] ISENBERG P., TANG A., CARPENDALE S.: An exploratory study of visual information analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), ACM, pp. 1217–1226. 2

[JKMG07] JANKUN-KELLY T., MA K.-L., GERTZ M.: A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007). 1, 2

[jup17] Project jupyter, 2017. URL: http://jupyter.org/. 3

[kag17] Kernels | kaggle, 2017. URL: https://www.kaggle.com/kernels. 1, 3, 7

[KCPE16] KIM H., CHOO J., PARK H., ENDERT A.: Interaxis: Steering scatterplot axes via observation-level interaction. *IEEE transactions on visualization and computer graphics 22*, 1 (2016), 131–140. 11

[Kee06] KEEL P. E.: Collaborative visual analytics: Inferring from the spatial organization and collaborative use of information. In *IEEE Symposium on Visual Analytics Science And Technology* (2006), IEEE, pp. 137–144. 2

[Knu84] KNUTH D. E.: Literate programming. *The Computer Journal 27*, 2 (1984), 97–111. 3

[KW78] KRUSKAL J. B., WISH M.: *Multidimensional scaling*, vol. 11. Sage, 1978. 5

[KWHH17] KIM Y., WONGSUPHASAWAT K., HULLMAN J., HEER J.: Graphscape: A model for automated reasoning about visualization similarity and sequencing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017), ACM, pp. 2628–2638. 2, 4

[LCM15] LI J., CHOU J.-K., MA K.-L.: High performance heterogeneous computing for collaborative visual analysis. In *SIGGRAPH Asia Visualization in High Performance Computing* (2015), ACM, p. 12. 2

[M*67] MACQUEEN J., ET AL.: Some methods for classification and analysis of multivariate observations. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability* (1967), vol. 1, pp. 281–297. 5

[Ma99] MA K.-L.: Image graphs-a novel approach to visual data exploration. In *Proceedings of the conference on Visualization* (1999), IEEE Computer Society Press, pp. 81–88. 2

[mat17a] Matlab live editor, 2017. URL: https://www.mathworks.com/products/matlab/live-editor.html. 3

[mat17b] Wolfram mathematica: Modern technical computing, 2017. URL: https://www.wolfram.com/mathematica/. 3

[MBM*12] MCGRATH W., BOWMAN B., MCCALLUM D., HINCAPIÉ-RAMOS J. D., ELMQVIST N., IRANI P.: Branch-explore-merge: facilitating real-time revision control in collaborative visual exploration. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (2012), ACM, pp. 235–244. 2

[MCCD13] MIKOLOV T., CHEN K., CORRADO G., DEAN J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013). 4

[MH08] MAATEN L. V. D., HINTON G.: Visualizing data using t-SNE. *Journal of Machine Learning Research 9*, Nov (2008), 2579–2605. 5

[Mon16] MONTOYA A.: Kaggle kernels: A new name for "scripts", 2016. URL: http://blog.kaggle.com/2016/07/08/kaggle-kernel-a-new-name-for-scripts/. 7

[MT14] MAHYAR N., TORY M.: Supporting communication and coordination in collaborative sensemaking. *IEEE transactions on visualization and computer graphics 20*, 12 (2014), 1633–1642. 2

[PL13] PELTONEN J., LIN Z.: Information retrieval perspective to meta-visualization. In *Asian Conference on Machine Learning* (2013), pp. 165–180. 2, 4

[rno17] R notebooks, 2017. URL: http://rmarkdown.rstudio.com/r_notebooks.html. 3

[Rob08] ROBINSON A. C.: Collaborative synthesis of visual analytic results. In *IEEE Symposium on Visual Analytics Science and Technology* (2008), IEEE, pp. 67–74. 2

[SF09] SHARMA N., FURNAS G.: Artifact usefulness and usage in sensemaking handoffs. *Proceedings of the Association for Information Science and Technology 46*, 1 (Nov. 2009), 1–19. doi:http://dx.doi.org/10.1002/meet.2009.1450460219. 2

[Sha08] SHARMA N.: Sensemaking handoff: When and how? *Proceedings of the American Society for Information Science and Technology 45*, 1 (June 2008), 1–12. doi:http://dx.doi.org/10.1002/meet.2008.1450450234. 2

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of IEEE Symposium on Visual Languages* (1996), IEEE, pp. 336–343. 4

[SMWH17] SATYANARAYAN A., MORITZ D., WONGSUPHASAWAT K., HEER J.: Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics 23*, 1 (2017), 341–350. 2

[ST15] SARVGHAD A., TORY M.: Exploiting analysis history to support collaborative data analysis. In *Proceedings of Graphics Interface Conference* (2015), Canadian Information Processing Society, pp. 123–130. 2

[tab17] Tableau software, 2017. URL: https://www.tableau.com/. 1

[TIC09] TOBIASZ M., ISENBERG P., CARPENDALE S.: Lark: Coordinating co-located collaboration with information visualization. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1065–1072. 2

[VWVH*07] VIEGAS F. B., WATTENBERG M., VAN HAM F., KRISS J., MCKEON M.: Manyeyes: a site for visualization at internet scale. *IEEE transactions on visualization and computer graphics 13*, 6 (2007). 2

[WK06] WATTENBERG M., KRISS J.: Designing for social data analysis. *IEEE Transactions on Visualization and Computer Graphics 12*, 4 (2006), 549–557. 2

[wor17a] Google code archive word2vec, 2017. URL: https://code.google.com/archive/p/word2vec/. 4

[wor17b] word2vec | npm, 2017. URL: https://www.npmjs.com/package/word2vec. 4

[ZGI*17] ZHAO J., GLUECK M., ISENBERG P., CHEVALIER F., KHAN A.: Supporting handoff in asynchronous collaborative sensemaking using knowledge-transfer graphs. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2017), 340–350. 2