Eurographics Conference on Visualization (EuroVis) 2019
M. Gleicher, H. Leitte, and I. Viola
(Guest Editors)

Volume 38 (2019), Number 3

InsideInsights: Integrating Data-Driven Reporting in Collaborative Visual Analytics

A. Mathisen,[1] T. Horak,[2] C. N. Klokmose,[3] K. Grønbæk,[1] and N. Elmqvist[4]

[1]Department of Computer Science, Aarhus University, Denmark
[2]Interactive Media Lab, Technische Universität Dresden, Germany
[3]Department of Digital Design and Information Studies, Aarhus University, Denmark
[4]College of Information Studies, University of Maryland, College Park, MD, USA

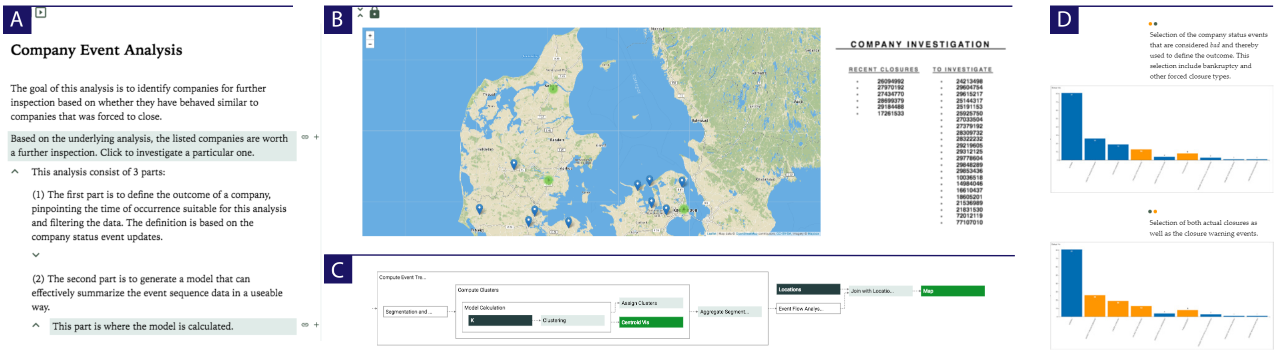

Figure 1: *The InsideInsights system: (a) a narration hierarchy allows gradually expanding and reviewing details. The annotation cells are linked to presentation views, either showing (b) selected visualizations, or (c) a part of the underlying analysis pipeline. Furthermore, (d) annotation cells can encapsulate multiple states for a linked component.*

Abstract

Analyzing complex data is a non-linear process that alternates between identifying discrete facts and developing overall assessments and conclusions. In addition, data analysis rarely occurs in solitude; multiple collaborators can be engaged in the same analysis, or intermediate results can be reported to stakeholders. However, current data-driven communication tools are detached from the analysis process and promote linear stories that forego the hierarchical and branching nature of data analysis, which leads to either too much or too little detail in the final report. We propose a conceptual design for integrated data-driven reporting that allows for iterative structuring of insights into hierarchies linked to analytic provenance and chosen analysis views. The hierarchies become dynamic and interactive reports where collaborators can review and modify the analysis at a desired level of detail. Our web-based INSIDEINSIGHTS *system provides interaction techniques to annotate states of analytic components, structure annotations, and link them to appropriate presentation views. We demonstrate the generality and usefulness of our system with two use cases and a qualitative expert review.*

CCS Concepts

• ***Human-centered computing*** → ***Visual analytics; Visualization toolkits; Collaborative and social computing;***

1. Introduction

Modern society—from business and journalism to medicine and policy—is increasingly data-driven. Data analysis itself is iterative, non-linear, and fragmented with both top-down and bottom-up characteristics [PC05, KPHH12, Tuk77]; sometimes the analyst lets the data itself inform assessments in an exploratory fashion, and sometimes the analyst will formulate priori hypotheses that are tested in a confirmatory fashion. Furthermore, data analysis is seldom confined to a single person; multiple collaborators will often contribute to the same analysis, or intermediate results can be shared with stakeholders. Thus, an important aspect of collaborative data analysis is to enable communication of insights even before the analysis has concluded [MT14, ST15, ZGI*18]. However, transferring knowledge is especially difficult for non-routine and ill-defined tasks [Sha08], hence effective communication requires a correspondingly rich mechanism to match the complexity of exploratory data analysis. In the context of reporting insights, a proven communication method is *data-driven storytelling* [GP01, SH10], but such mechanisms are not well integrated within the iterative and branching nature of sensemaking. For example, Tableau [Sof18] allows users to build stories to convey their insights, but this is done as a separate task once data analysis has concluded. Similarly, computational notebooks [RTH18, KRA*18, UPSK18] (such as Observable and Jupyter) have limitations when used for exploration and presentation simultaneously. Furthermore, slideshows and notebooks alike force analysts to linearize their findings and fixate on the level of detail presented to the audience, which again is not commensurate with its incremental nature.

In this paper, we present INSIDEINSIGHTS, a conceptual design and a web-based system for capturing both low-level insights and high-level abstractions during data analysis. The idea is to support data-driven reporting as an integral part of the analysis process by bridging proven concepts from literate computing, provenance tracking and storytelling. In essence, InsideInsights is a hierarchical insight management system allowing analysts to interchangeably (1) coalesce findings into higher-level abstractions, and (2) subdivide composite assumptions into items that can eventually be validated. The result is a multi-level report supporting hierarchical (pyramidal) thinking [Min09] that a collaborator or stakeholder can expand to the desired level of detail in order to understand the findings and the process. Our system (Figure 1) provides a structured data-flow environment where analysts can load, transform, and visualize data on an interactive canvas. Thus, InsideInsights supports the entire sensemaking loop [PC05], from information foraging and marshalling, to schematizing and presentation.

The proposed concepts are a step on the way to what we call *literate analytics*, where the goal is to document not only the final outcomes, but also provide a narrative of the actual analysis process itself. We envision that by unifying the analysis and communication components of sensemaking, our tools can better support the dynamic data analysis scenarios that organizations increasingly face [KPHH12, UPSK18]. The contributions of our work are the following: (1) a conceptual design that address core challenges for integrating data-driven reporting by bridging literate computing, provenance tracking, and presentation techniques. (2) the InsideInsights system, a proof-of-concept implementation using modern web technologies; (3) application scenarios of the system to two real world datasets; and (4) findings from an expert review indicating the soundness of our approach.

2. Background

In this section, we review existing work on supporting exploratory data analysis and data-driven communication.

2.1. Tracking Exploratory Data Analysis

Exploratory data analysis (EDA) is typically an iterative process that involves multiple cycles of, e.g., cleaning, modeling, visualizing, and interpreting data [Tuk77, PC05, KPHH12]. Consequently, analysts will often try multiple analysis avenues, of which several will be dead ends, until they arrive at usable insights and conclusions [KPHH12]. Furthermore, exploration often proceeds iteratively through several levels of abstraction: sometimes from low-level insights to high-level explanations, and sometimes from general schemas to concrete data [PC05, CBYE18]. In this work, we are interested in the entire data analysis pipeline, from cleaning and modelling to visualization and interpretation.

Maintaining an accurate record of this non-linear process is challenging, and has led to a large body of research focused on provenance tracking [DF08, RESC16] to support recall, replication, communication, and presentation. Interactive visualization is particularly useful for communicating provenance, and several systems thus incorporate provenance tracking. For example, GRASPARC tracks exploration histories as a branching hierarchy [BPW*93], which was later extended by Shrinivasan and van Wijk [SvW08] by allowing users to append annotations. VisTrails [BCC*05] was an early provenance management system for scientific visualization and analysis workflows. Similarly, the Burrito system automatically captures low-level computational activities to allow researchers to compare different analysis stages [GS12]. Gotz et al. [GZ09] instead used such low-level user events to infer high-level semantic meanings for the analytic process. Similar to our approach, this results in a hierarchical structure but is fixed to predefined task classes and probably is not able to adequately represent the analysts mental model. Further, as with all approaches of fully automated tracking of user interactions, it remains challenging to make sense of the resulting, often large, provenance graphs.

Recently Kery et al. [KHM17] described how analysts rarely use conventional version control, such as Git, thus motivating their design of a lightweight inline versioning method for programming environments to support EDA. In our work, we also employ provenance tracking of the entire analysis pipeline. However, the tracking is triggered by user annotations to support lightweight versioning, and to avoid cluttering the provenance history.

2.2. Data-Driven Storytelling

Data analysis is arguably meaningless if its findings cannot be effectively communicated to stakeholders, other analysts, or the general population [TC05]. One way to convey data analysis results is through data-driven storytelling [LRIC15], where annotations, rich media (videos, images, sound, tables, etc), and visualizations are combined into a narrative [GP01, SH10, HD11]. The increased focus on creating interactive presentations to communicate findings has led to several related concepts. Examples include active reading with live documents [WKM02], literate hierarchies using supplemental materials [GRSG17], interactive narratives [CH18], and external representations [Kir10, RRH*19]. However, because design choices for visual communication impact comprehension [Kir10, HDR*13], there often is no single presentation that fits all audiences. For example, a C-level executive may only need the top three take-aways from a quarterly sales report, whereas a strategic account manager will want to understand the provenance and details of the same analysis.

Current tool support for data-driven storytelling mostly consists of linearizing analytical findings into a sequence [SH14], which requires the designer to fixate on a specific abstraction level in their narrative. This discards the richness of non-linear analyses and makes it difficult to review the findings at the desired level of detail. Separating analysis from presentation also fragments the sensemaking process further, often making the communication of analytic products an afterthought instead of a mechanism to support collaboration throughout the exploration [RTH18, KRA*18, VW06]. For this reason, Gratzl et al. [GLG*16] created a method for seamlessly switching between visual exploration, authoring, and presentation specifically for interactive visualizations using provenance tracking of user interaction. In our work, we extend previous work by focusing on the entire analysis pipeline, i.e., the provenance of cleaning and modelling parts as well as user interaction in visualization parts. Furthermore, our hierarchical insight structure allow users to build interactive reports targeting multiple user types while still preserving the provenance in full detail.

2.3. Notebooks and Toolkits

In recent years, literate computing in the form of computational notebooks [KRKP*16, RNA*17, Goo18, Obs18, TME08] have become an essential part of data science [RTH18, KRA*18], even of entire organizations [UPSK18]. Notebooks combine executable code, their output, and text in a single document, which has proven to be very useful for quick prototyping and exploration. Thus, notebooks already possess qualities suitable for data-driven communication and replicability. For these reasons, recent efforts have tried to enhance their functionality; for example, synchronous collaboration in notebooks is becoming the de facto standard [Goo18, RNA*17], and reactive execution flows are also being supported [Obs18, BMR*19]. Nevertheless, the linear document nature that is fundamental to all computational notebooks yields a tension between exploration and presentation [RTH18, KRA*18], making it difficult to support non-linear workflows that alternate between generating and presenting insights.

Beyond notebooks, several toolkits exist for quickly creating visualizations for exploratory data analysis [SWH14, SMWH17a, MNV16, YEB18]. However, these systems generally do not provide support for insight management during EDA. A prominent example is the Tableau [Sof18] suite that takes analysts from data preparation, to visual exploration, and finally to presentation. However, narration is typically a separate step in current tools, or done in completely different tools [BE18, NK18]. In our work, we build on the success of collaborative notebooks and pipeline-based methods for data analysis by implementing our system on top of Codestrates [RNA*17] and Vistrates [BMR*19] (cf. Section 4.4).

3. Design Framework: Data-Driven Reporting

The sensemaking loop [PC05] is an iterative process for foraging data, building schemas from specific findings, and creating presentations from the schemas. In this way, the analyst engages in a gradual, bottom-up, and data-driven refinement that slowly increases in abstraction level. The whole process is iterative, meandering, and prone to branching [BPW*93, SvW08]: sometimes, the analyst will backtrack to explore a different avenue of inquiry; sometimes entire paths will be abandoned; and sometimes the current strand of investigation proves to be the right one. Supporting this type of workflow is a challenge in itself, and in collaborative scenarios it is also a challenge to share, hand off, and communicate insights from such a process, as amply pointed out in past work [HA08, Sha08, KPHH12, MT14, RTH18, ZGI*18]. In addition, there is no watertight boundary between analysis and presentation [PC05]; sometimes a presentation may even return back to exploration. Thus, as described already in the initial research agenda for visual analytics [TC05], all this indicates an explicit need for integrating presentation in the analytic process.

For example, consider a team of analysts tasked with finding corrupt companies in financial data. In order to identify potential violators, they may analyze known cases and identify typical conducts, e.g., specific registration changes, financial statements, or shareholder changes. Within the team, the members have varying competences; some are expert programmers, some have knowledge of data science methods, and others possess domain knowledge. Sharing data work and insights in this environment is a challenge as highlighted above, and especially when sharing partial progress. In addition, the team occasionally needs to consult with their stakeholders to discuss next steps as well as share their work with other teams within the organization to support reuse and to align knowledge about risk patterns. This scenario is based on data from the Danish Business Authorities, indicating that blending analysis and reporting is an intrinsic challenge within larger organizations (we expand on this scenario in Section 5.2). In this section, we first explore core challenges for integrating data-driven reporting and then describe design concepts that address them.

3.1. Challenges for Integrating Data-Driven Reporting

Through an analysis of related work and systems, we have identified three core challenges for integrating data-driven reporting:

C1 – Annotations & External Representations. Insights are pieces of information relevant to the analysis [Nor06] that can be characterized in manifold ways—complex, deep, qualitative, or even unexpected—and refer to different artifacts, e.g., single data points, patterns, or the overall analysis structure. Capturing and presenting such insights is a challenge and often require rich mechanisms for creating interactive narratives [WKM02, LRIC15, CH18], annotating visualization states [VWVH*07, HVW09], and constructing external representations [Kir10, RRH*19]. Still, few tools incorporate such methods as part of the analysis environment. For example, in notebooks, it is mainly the layout that specifies the relation between annotations and computations, which means that describing the iterative nature of an analysis is challenging [KRA*18]. This challenge therefore includes maintaining exactly what parts of the analysis annotations refer to and simultaneously supporting interactive representations [HVW09, Kir10].

C2 – Adaptive Details. Expertise and level of interest vary in collaborative workflows, which means that every detail is not always beneficial, but should be readily available for review when required. This challenge has been highlighted in various ways in related work. It has been described how preparing a notebook for presentation often requires the analyst to delete parts—parts that may be of benefit at a later stage [RTH18, KRA*18]. Rule et al. [RDTH18] have subsequently motivated how dynamic restructuring of notebooks can support sharing. Research has also shown how hierarchical structuring of supplemental materials can benefit comprehension [GRSG17] and how analysis history can support collaboration and task hand-off [ST15, ZGI*18]. Thus, it is not only a challenge to dynamically maintain details of an analysis, but also in a way that indicates how insights are related. Current tools provide little support for such multi-level and broad-audience analysis products that can capture the full richness of the data analysis process, including not just findings, but also dead ends and failed hypotheses.

C3 – Fragmented Workflows. An overarching challenge is the non-linear nature of collaborative data analysis [Sha08, HA08, KPHH12, MT14]. An analysis frequently alternates between exploration and reporting, with the same analysis parts often revisited several times. However, in most current visual analytics tools—with a few exceptions [GLG*16]—analysis and presentation are separate stages that are poorly integrated [RTH18, KRA*18]. By forcing the user to create presentations as a separate activity, the rich provenance of the analysis is easily lost and it becomes nontrivial to go back to the analysis stage or reuse analysis parts.

3.2. Linking Annotations to Analysis States

To address C1, we adopt the concept of linking annotations to visualization states [VWVH*07, HVW09], but we expand it to incorporate the full provenance of the analysis pipeline. Conceptually, an annotation can be linked to a snapshot of the analysis parts it describes (Figure 2a). For example, the states can capture selections in visualizations, input values of UI elements, or computational output. The subset of the analysis which an annotation depends on is defined by how various parts are linked. For example, annotating the first cleaning step does not depend on a later visualization, but annotating the visualization depends on the first cleaning step. While the provenance information is automatically maintained, the user defines which states should be captured by triggering the tracking through annotating insights. As this force the user to actively create annotations throughout an analysis, it addresses the issue of making sense of a potentially inflated provenance graph after EDA.

This concept also addresses C3 in that it is possible to return to old stages of an analysis and use the annotations to recall any reasoning. However, changing the analysis state during EDA can have the side effect of negating certain annotations as they no longer describe the active analysis states. We therefore introduce the concept of *active* and *inactive* annotations. If the current states matches the linked states of an annotation, it is active; otherwise it is inactive. Analogously, manually activating the annotation restores the snapshot of the analysis system. This enables bi-directional interaction for exploring annotations, contrary to the one-directional interaction in, e.g., Idyll [CH18]. But, it also introduces design challenges for adequately indicating when annotations become active or inactive. Finally, annotations can be grouped based on whether they describe, and thus depend, on the same analysis parts (represented as stacks in Figure 2a). If each annotation in a stack describes a different state, only one annotation can be active.

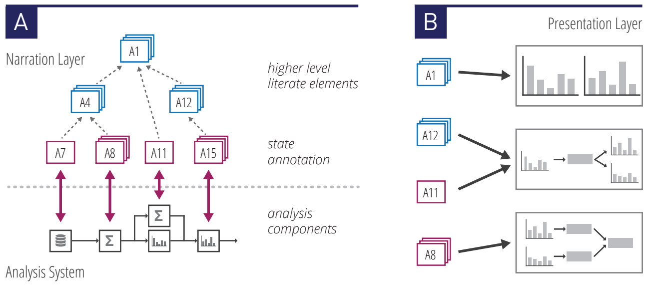

Figure 2: *Bridging narration and analysis: (a) the dynamic annotation hierarchy ties low-level insights to analysis states and links them to higher-order schemas; further, (b) the annotations can be linked to specific presentation views.*

3.3. Dynamic Insight Hierarchies

For tackling challenge C2 and supporting the flexible nature of the sensemaking loop, we propose to maintain all annotations in a hierarchical structure. This is motivated by the observation that people tend to organize information in hierarchies, in order to break down ideas and make them easier to capture [Min09]. The natural alternative to a hierarchical information structure is a graph structure. However, it has been shown that hierarchical structures can better support overview and comprehension, especially for readers with low prior knowledge of the content [ATM09, SVLV16]. In contrast, graph structures are better for certain information seeking tasks for readers with higher prior knowledge, but this structure imposes a more demanding process on the reader [ATM09, SVLV16]. To build up such a hierarchy, analysts can capture an annotation in a cell (as in computational notebooks). Similar to the concept of structured writing [Hor98], these annotation cells allow the analyst to organize the analysis details and abstract multiple specific insights into generalized structures that we call *dynamic insight hierarchies*.

As annotations can be grouped based on the same analysis parts they describe, related annotation cells can automatically be kept as groups within the hierarchy. The active annotation cell can then be shown as the top (visible) cell in a group and the inactive annotations can be thought of as tabs with alternative explanations. By supporting this as the default behavior, users are assisted in adding new annotations to relevant places of the hierarchy. This way, the annotations can capture the alternative decisions that were considered (and thus which ones were omitted), kept in a semantically sound location of the hierarchy. For example, there is typically a specific model or visualization selection that proves to be the ideal choice for a given analysis. However, this behavior does not fit all analysis types, and if multiple annotations about the same visualization state are created there is more than one active annotation. Thus, it should be possible to split up annotation groups to have annotations about the same analysis parts in different places of the hierarchy. Note, all annotation cells in a hierarchy does not have to depend on analysis states. It should also be possible to create normal cells as in the literate computing paradigm [KRKP*16] within the hierarchy that e.g. describe the logic behind an algorithm.

The insight hierarchies can be constructed during the analysis process and at any point be restructured, thus supporting challenge C3. Conceptually speaking, annotation cells can either be *merged* by introducing a parent cell, which summarizes the high-level meaning or interpretation of the underlying cells, or *split* by adding children to an existing cell, and thereby subdividing the parent into multiple supporting pieces of evidence (Figure 2a). Supporting both types of operations interchangeable is critical to support the non-linear nature of EDA. By structuring insights during the analysis, the author is already creating a narrative similar to data-driven storytelling [GP01, SH10, HD11], i.e. a guided way of understanding the current content. In addition, the dynamic insight hierarchy supports precisely the adaptive level-of-detail outlined in challenge C2. A viewer presented with an insight hierarchy will initially only see the top-level element (or potentially only a few levels down). Depending on their expertise, available time, and interest, the viewer can then expand annotations with children as far as desired, potentially down to the atomic annotations at the bottom of the hierarchy. Readers can also choose to stay on a single level-of-detail and thus use the report as a linear story.

3.4. Hierarchical Presentation Views

To complement the annotations and address C1, we propose the concept of *presentation views*. By default, annotation cells remember the analysis view during their creation, allowing to restore it later along with the analysis state. Presentation views are an addition that allow to define a different representation. For example, a user can manually create a canvas containing visualizations alongside with hand-drawn elements. This flexible creation of external representations supports the comprehension of insights in the same way visual elements are used in data-driven storytelling [LRIC15]. Similar to linking annotations to analysis states, annotation cells can be linked to presentation views (Figure 2b). When browsing the annotation cells, the linked view is automatically shown to the user and the analysis state restored. By supporting easy creation of such views throughout the analysis process, the gap between the analysis and the presentation (C3) is effectively minimized.

Furthermore, traditional presentation tools such as Microsoft PowerPoint, or even Tableau's story points, force the analyst to linearize a rich and branching analysis process into a flat sequence. Linking the information cells of the dynamic insight hierarchy to presentation views effectually creates *hierarchical presentations*, further supporting C2. In our concept, it is simply the insight hierarchy that defines the hierarchy of the report. Instead of merely being able to go backwards and forwards in the presentation sequence, our hierarchical approach also provides "up" and "down" operations that will *roll-up* and *drill-down* into the hierarchy, respectively (Figure 3). This way, the reader can lower or raise the abstraction level as needed. To support interactive reading, presentation views remain interactive such that readers can explore the visualizations, and the annotation cells will then adapt to user interaction by changing to *active* or *inactive*. Hierarchical presentations can be explored directly in the workspace interface where the insight hierarchy is constructed, but it is also possible to create dedicated presentation interfaces that support different types of review experiences and controls (Figure 3).

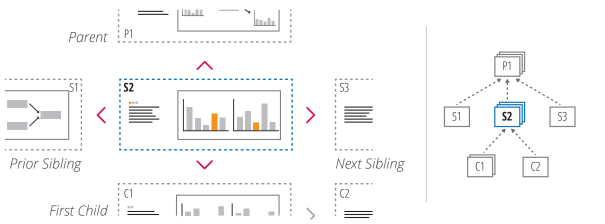

Figure 3: *The hierarchical presentation concept. Navigating the report entails not just visiting annotations on one level, but also ascending and descending the hierarchy.*

4. The InsideInsights System

The InsideInsights system is a proof-of-concept of the design described above, implemented as a collaborative web-based solution (github.com/90am/insideinsights). The system is build on top of the computational notebook Codestrates [RNA*17] and the Vistrates component model [BMR*19]. The novel combination of literate computing, hierarchical structure, provenance tracking, and presentation techniques allows for an iterative workflow interleaving composing analysis parts, capturing insights, and sharing the current progress. The workflow is therefore similar to the open-ended workflow of computational notebooks. In this section, we will explain the system with its three main parts: (A) compose analysis, (B) develop insight hierarchy, and (C) share and review.

4.1. Part A: Compose Analysis

The InsideInsights system supports a visual data-flow approach, where a user performs sensemaking by assembling components to transform and visualize the data on an interactive canvas (Figure 1c). For this purpose, Vistrates [BMR*19] provides a component template and a reactive data-flow based execution model within a notebook environment. A component in Vistrates is essentially a piece of code that consumes one or multiple input sources, process the data (e.g., to create a view), and eventually provide an output. The component observes the data output of its sources and reacts on any updates, causing a recalculation of any views, data, or filters. A wide range of existing components (e.g., aggregation methods, visualizations, input controls) can be instantiated from a global repository, or new ones programmed and shared, either by inheriting from a prototype or written from scratch. Instantiating a component will insert the prototype code in the notebook. This way, components can be edited or reconfigured in the notebook anytime. The configuration of a component contains input mappings as well as the current state of the component.

The component model allows for the interactive pipeline abstraction in our system (Figure 1c) showing the data flow between components. The pipeline supports both component configuration and inspection of views (cf. Figure 9). With InsideInsights, we also extend the core functionality of Vistrates by introducing composite components, which allow users to build hierarchical pipeline parts that can be collapsed or open as seen in Figure 4. Besides helping to structure the pipeline in a meaningful way, these composite components also allow users to share and instantiate a set of pre-configured components. We will return to the implementation of composite components in section 4.4. To ease the understanding of the pipeline, we also automatically distinguish between four high-level component types. Dark components are data sources, which include both data as well as parameter inputs. Green components are visualizations, and light components are computations with no view, such as a filter or a k-means clustering algorithm. The unfilled components are composites. A key strength of our system is the ability to seamlessly switch between working in the InsideInsights interface and developing components with the underlying notebook interface. It is the individual components that decide what is piped downstream, but a typical visualization component will output the current selection and store this as its state. The generality of our approach is inherited from the generality of the underlying component model. In essence, there is no restriction on what a component can do as long as it adheres to the general template. Still, the integrated data-driven reporting design can also be implemented with other analysis systems.

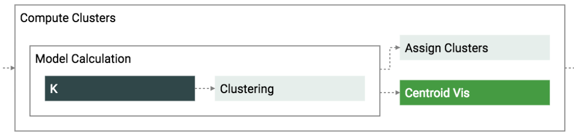

Figure 4: *Part of a pipeline showing a clustering component that takes as input a parameter K. The resulting centroids are visualized in Centroid Vis. The components have been grouped into two (currently open) composites.*

At any point when constructing the pipeline, the analyst can capture important insights in the data by creating an annotation cell to describe specific component. The full provenance of the analysis, i.e., the states of the previous components in the pipeline, is automatically maintained in the background. We describe this in detail in Section 4.4. Annotation cells can contain any HTML content, similar to text cells in, e.g., Observable [Obs18]. As mentioned in Section 3.3, annotations about the same component are by default grouped together in the hierarchy, and the different annotations are then represented as dots (see Figure 5). Orange dots are *active* annotations and grey are *inactive*. Unless the annotations describe the exact same set of states (cf. subsection 4.4), usually only one annotation will be active at the time, and thus be the visible one. Otherwise, the latest active annotation will be visible as default, and the user can browse both active and inactive annotations. When the component states change, the active and inactive annotations will update accordingly, surfacing the current active annotation. While it is not possible to split up these groups in the current prototype, it is trivial to implement. Alternatively, annotation cells without links to any component states can also be created in the hierarchy.

4.2. Part B: Develop Insight Hierarchy

The *dynamic insight hierarchy* is always visible, even during analysis, on the left side of the screen in our prototype (Figure 1a). This hierarchy contains all annotation cells, both those that depend on the analysis state (cells with dots above) and cells with general descriptions. Composing a data-driven report in InsideInsights can be

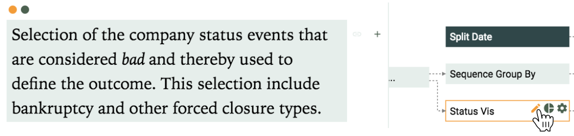

Figure 5: *Creating an annotation cell linked to a component state can be done simply using the pen icon on the specific component. Whenever the state of the component changes active (orange dots) and inactive (grey dots) annotations update accordingly.*

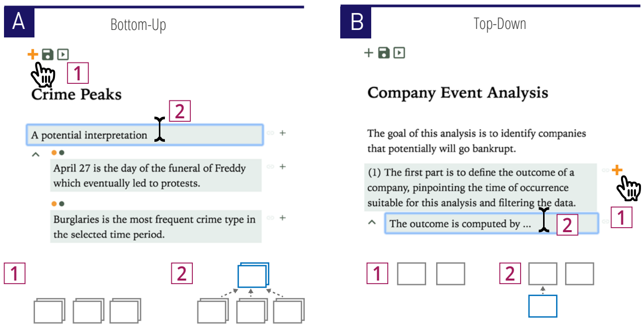

Figure 6: *Illustration of bottom-up (a) vs. top-down (b) analysis workflows as supported by InsideInsights. In (a) multiple annotations are summarized in a new parent, and in (b) a high-level goal is further specified with a new child.*

performed in both top-down and bottom-up fashion (Figure 6). For top-down, the user may create annotation cells with merely textual or rich media content that are not yet linked to any analysis state or presentation view. In the confirmatory fashion of a top-down approach, such cells would describe high-level hypotheses about the data. A series of conceptual *split* operations would populate these hypotheses with annotation cells as children, and this process would be repeated until the analyst can anchor each claim in actual data. For a bottom-up, exploratory approach, the analyst would instead let the data itself guide them, surfacing specific low-level insights in the dataset and then letting these be gradually aggregated into higher-level findings using annotation cells in conceptual *merge* operations. Specifically, the hierarchy can be extended by selecting an existing cell and append a child (Figure 6b), or by selecting multiple cells and append a parent (Figure 6a). Alternatively, the pen icon on a component is a shortcut for creating an annotation cell that is linked to that component (cf. Figure 5). We expect that a practical analysis and authoring session in InsideInsights will alternate between both of these modes. Thus, an existing cell can at any point be selected and moved around in the hierarchy.

Analysts can also link a *presentation view* to an annotation cell. We currently support two view types: (1) the analysis pipeline itself, configured with certain composites open or closed, or (2) a canvas with selected component views, such as a single full-screen visualization or a dashboard of multiple visualizations, rich media and hand-drawn elements (Figure 7). When an annotation cell has a linked presentation view, that view will be shown instead of the full pipeline when reviewing the cell later on (see Part C). This allows the analyst to author external representations that show specific items of interest and hide the complexity of the full pipeline (Figure 7). To attach a specific cell to a presentation view, users will first create an empty slide (or canvas) and populate it with items from the pipeline as well as rich media (text, images, video, etc). They can then use the link icon next to the selected cell to connect the presentation view to the cell.

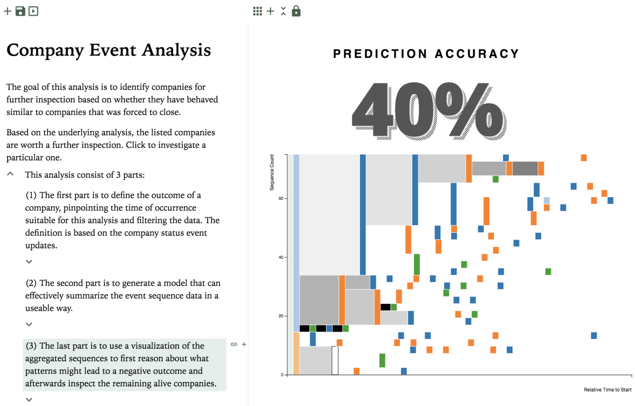

Figure 7: *A presentation slide with two freely arranged views; an event flow visualization and a prediction accuracy view. The presentation slide is linked to the bottom narration cell as indicated by the link icon next to the cell.*

4.3. Part C: Share and Review

Sharing and reviewing a report in InsideInsights is not a separate mode; at any point during the exploration process, a user can traverse the growing insight hierarchy, either by clicking on cells or by using the arrow keys. Sharing an analysis is simply done by sharing a link to the web-based document—multiple collaborators can even modify the same document simultaneously. Having said that, InsideInsights also supports a dedicated presentation mode, where the report can be navigated similar to a traditional slideshow and where edit controls are removed. The reviewer can choose to stay on a single level in the hierarchy, and thus read a linear story, but a reviewer can also choose to traverse the hierarchy to learn additional details or abstractions. As mentioned previously, InsideInsights supports bi-directional interaction (Figure 5). Activating an annotation can be done by clicking the appropriate dot, which will restore the analysis state of that annotation. Vice versa, modifying the analysis state (e.g. interacting with a visualization) will update active and inactive annotations, and surface the active ones. This way, the visible narrative always match the current analysis state. Thus, if a reviewer modifies the analysis such that it violates a given reasoning, the insight hierarchy will reflect this by making the involved annotations inactive.

4.4. Implementation Details

The InsideInsights system is implemented with standard modern web technologies (JavaScript, HTML, CSS, etc.) as a top layer on the existing technology stack consisting of Webstrates [KEB*15], Codestrates [RNA*17, BRK18], and Vistrates [BMR*19]. Codestrates is a JavaScript-based computational notebook that provides collaborative editing of code through the DOM synchronization mechanism of Webstrates. On top of this, Vistrates provides a general component model, a data-flow based execution model, and view abstractions such as the pipeline and visualization dashboards. In theory, any computational or visualization component can be programmed using Vistrates. The InsideInsights system extends the existing technologies with composite components, annotation-driven provenance tracking, and interactive hierarchical reports.

Composites components are implemented with the classic composite software pattern [Gam95] applied to the existing component model. Thus, a composite component is a component itself that contains a list of child components and specifications of their configurations. This way, a composite component can be instantiated as any other component; which also instantiates child components with the specified configurations. Already instantiated components can be combined to form a new composite by the user. Subsequently, the user can share the new type of composite through the component repository.

As components store their configuration and state encoded as JSON [BMR*19], the provenance tracking in InsideInsights is implemented using these specifications. The analysis state of component X is therefore defined by the state specification of X itself along with the states of the previous components in the pipeline that X depend on. This way, the pipeline defines a state dependency graph which is automatically maintained by the system. But, as mentioned previously, the provenance tracking in InsideInsights is triggered by user annotations. Thus, not all state changes will result in an expansion of the state dependency graph; whenever a user links an annotation to a component, only the relevant states for that insight are appended to the graph. Restoring an analysis state then simply consists of traversing the state dependency graph and update the relevant components. The insight management and provenance tracking—forming the InsideInsights system—is implemented as a meta-package for Vistrates. Along with the state dependency graph, this package maintains annotation and state links along with mappings between text cells in the underlying notebook and the annotation cells of the InsideInsights system. This way, there is a one-to-one mapping between text cells in the two interfaces.

5. Usage Scenarios

There are many possible applications for a data-driven reporting system such as InsideInsights. To showcase the breadth of our system, we provide two conceptually different application examples:

- **Crime data analysis:** Baltimore crimes data analysis, where findings inform the next steps in the analysis, yielding a shareable data-driven report. This conceptual scenario illustrates how a basic workflow can unfold.
- **Company event analysis:** The development of a method for finding companies that may face imminent bankruptcy. The method is based on prior development of new analysis methods in collaboration with the Danish Business Authorities (DBA) [MG17]. This scenario is in part informed by the existing collaborative practices at DBA, which involve people with varying expertise, and in part by how InsideInsights can foster new collaborative work practices.

5.1. Baltimore Crime Peaks

Let us follow a fictional crime analyst (John) who has been tasked with analyzing crime peaks in the city of Baltimore, MD. John loads a dataset of 110,074 crimes (data.baltimorecity.gov) from the period of 2012 to 2015 into the InsideInsights system with a CSV component. He generates a line chart of crimes over time by first instantiating an existing composite component from the repository that consists of three internal components; time-based aggregation, a line chart visualization and a data filter. He then configures the composite component in the pipeline by selecting the CSV component as the data source and by specifying property mappings, i.e., the names of the time and crime count variables. Upon completion, he identifies two large peaks warranting further investigation. He therefore creates two state annotations on the line chart using the pen icon, one for each crime peak. This way he can focus his analysis on one peak without having to remember the other.

The first crime peak occurs on April 27, 2015. John creates a bar chart of the crime type distribution for the selected date, which shows large peaks for aggravated assaults and burglaries. He quickly annotates both selections. To confirm that the peak is not only caused by one of the individual crime types, he creates a line chart of crimes over time separated by crime type. Then he merges the state annotations for the crime type visualizations into a high-level annotation that describes the intuition and purpose behind this additional comparison. He also links the high-level annotation to a presentation view with the two crime type visualizations that support this comparison, by organizing the views on a canvas and pressing the link icon next to the annotation cell.

John now creates a map visualization where the selected crimes are shown. Interestingly, almost all aggravated assaults occur at the same location, while burglaries are scattered throughout town. John finds this pattern curious and annotates both states of the map. Since the states of the map are linked to previous components, it is easy for John to retrace how he arrived at these findings and understand exactly how the visible crimes have been selected. As his analysis is set to April 27, John googles this date and finds multiple articles about the funeral of Freddie Gray, a young African-American man killed by Baltimore police. John then notes that the location of the aggravated assaults coincides with the protests that followed the funeral. Therefore, he merges the underlying annotations into a high-level interpretation, noting that the increase in burglaries was likely caused by looters taking advantage of the confusion of the riots. The resulting analysis is shown in Figure 8.

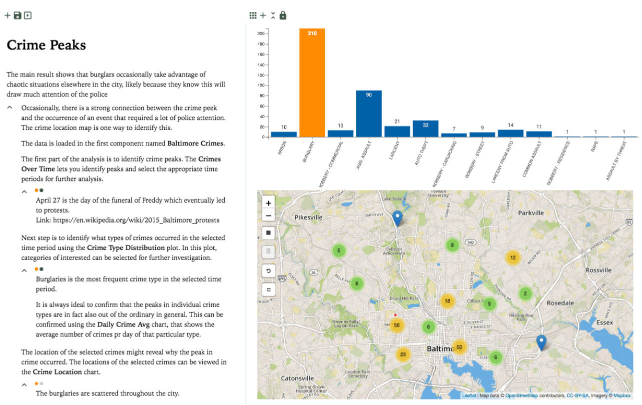

Figure 8: *Crime peaks analysis document after John's initial analysis session.*

5.2. Danish Company Event Analysis

Elina is a fictional analyst at the DBA. The DBA maintains information about Danish companies, performing analyses to aid political decisions and to catch non-compliant behavior. Elina has been tasked with finding early indicators of involuntary closure. She starts by loading a dataset of 10,005 events into InsideInsights. Her goal is to develop an analysis method that subsequently can be used on other subsets of the database that consist of more than 50 million events (datacvr.virk.dk/data). Events include business type updates, board member changes, accountant replacements, etc.

Elina quickly realizes that given the status registration event in the data, it is not straightforward to define what qualifies as an involuntary closure. Using a bar chart of the different status updates and their frequency, she creates an initial definition. She adds annotations to the visualization selections motivating why these have been chosen. Further, she merges these into annotations describing how the definition is currently calculated. Elina then links the high-level annotations to presentation views with the appropriate visualizations. To check the validity of this definition, Elina shares her current work with a domain expert on the registration data. The expert browses the insight hierarchy aided by the presentations views to understand the current definition. To improve the definition, he modifies the visualization selections and adjusts the annotations to also include companies that have closed due to other involuntary closure types than just bankruptcy. Further, he selects the date of the outcome to be the first time the companies were warned by the DBA. Since both the calculation and motivation have been captured in the InsideInsights system, future investigations into troublesome companies can now easily be performed with the same definition.

Elina continues her analysis by envisioning the computational parts she needs: (1) outcome definition and event filtering, (2) event aggregation, and (3) event hierarchy and prediction potential visualizations. She creates high-level annotations describing her envisioned analysis, and then repeatedly splits this description into multiple children that eventually can be realized in the pipeline. After composing the pipeline for the second and third part of the analysis, Elina realizes that the initially envisioned hierarchical aggregation of the raw event sequences is insufficient. The event sequences vary too much to be efficiently aggregated. Elina therefore develops another version; however, she keeps the first attempt as a subtree to motivate why the new approach was needed. As the new approach requires a component not already in the repository, she uses her current report to coordinate with an internal programmer. The programmer then implements components that cluster the sequences and computes high-level events. Elina now uses these components in her pipeline prior to the hierarchical aggregation. She stores her entire exploration using state annotations that she links to each part of the planned analysis. The resulting event flow visualization of the new model can be seen in Figure 9.

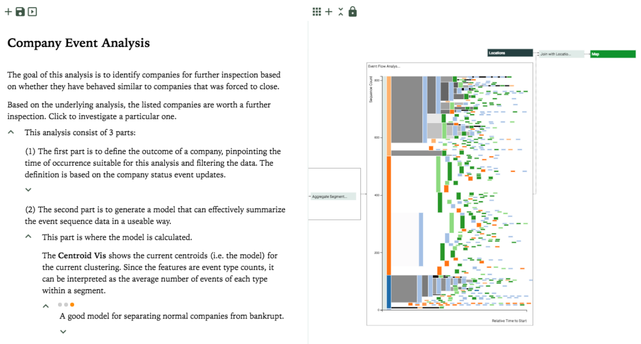

Figure 9: *Company event analysis document with an active model that is good for separating the bankrupt (dark) from the normal (light) event sequence flows [MG17].*

One year later, the DBA has started to collect additional registration events, and Jenna is accordingly tasked with revising the analysis. She goes through Elina's annotations, motivations, and descriptions to gain an understanding of the current method. She quickly identifies where to modify the analysis and begins her work. Because the narrative is linked to the program, she can easily update old motivations that no longer hold true for the updated analysis.

6. Expert Review

We conducted an expert review of the InsideInsights system with two experienced visualization researchers: an associate professor of computer science with 15 years of experience (P1), and a visual analytics specialist employed in industry for the past 3 years following his Ph.D. (P2). Our goal was to gather early feedback and insights on the potential use of the integrated data-driven reporting functionalities. We chose an expert review because qualitative methods are generally better suited to capturing the form of high-level sensemaking tasks typically performed using exploratory data analysis [Car08, Pla04], and expert reviews in particular have been shown to be effective for visualization evaluation [TM05].

6.1. Method

Each session was structured in two phases. In **Phase 1**, participants were given the full report of the company event analysis scenario, and in **Phase 2**, they were given only the analysis pipeline of the crime peaks scenario without annotations. For the full report (Phase 1), the experts were tasked with explaining the analysis and the interplay between the involved components, modify parameters to generate a better model, and subsequently summarize what the most important events were. For the analysis pipeline in Phase 2, the experts were tasked with exploring the data and building a narration hierarchy of their insights from scratch. Prior to beginning, participants were given an introduction to the purpose of our research and a tour of the main features in our prototype. During the phases, the participants were instructed to follow a think-aloud protocol. Sessions lasted around an hour and were screen-captured and voice-recorded. Also, observation notes were collected.

6.2. Expert Feedback & Findings

The tasks were designed to be open-ended since we wanted the participants to explore all parts of the system and get their feedback on its usefulness. Both participants were able to solve the tasks; however, the open-ended nature of the evaluation resulted in them employing different approaches in Phase 1. P1 mainly followed the narrative and observed the attached views and how the pipeline unfolded, while P2 instead investigated the pipeline and consulted the narrative only when something was unclear.

In general, both participants were engaged by the dynamic workflow of the prototype, and they both felt that there was *"great potential"* in the concept. P2 explicitly mentioned that they would like to spend more time using the prototype. The participant specifically emphasized that gradually diving into further detail was a nice way to understand an analysis, either by going down the narration hierarchy or by browsing the pipeline hierarchy. Both participants pronounced the close link between narrative and analysis useful.

P1 noted that in our prototype, it is hard to navigate both hierarchies (narration and pipeline) at once since they do not necessarily align. While this provides freedom to develop many complex types of data reports, it is *"freedom under responsibility,"* as P2 said. This complexity also manifested itself when the participants found a lack of additional explanations in certain parts of the analysis in Phase 1. In addition, P1 found it confusing when the narration changed perspective between method annotations (how) and analysis result annotations (what). The participant suggested to explicitly provide *how*, *what*, and *why* annotations—similar to the structured tagging in CommentSpace [WHHA11]—and allow the user to explicitly choose a certain perspective to read, or to have different narratives for different audiences. Having explicit annotation types could also support authors in developing the narration, as it would highlight what has already been described and what is missing.

Overall, the participants also noted that the current InsideInsights prototype has usability limitations. Some interactions still have latency due to the size of the datasets and the complexity of the pipeline. However, these issues can be resolved by optimizing the underlying analysis system, as well as by offloading computation onto the server. Other usability issues arise when the analytical components in InsideInsights did not offer functionality desired by the participants. P1 also suggested providing a more restricted (or guided) interaction to assist novices. Once more experience is gained, novices can then move on to the full system.

7. Discussion & Future Work

We believe that our proposed system is a first step to better support existing analysis and reporting workflows. In the following, we will discuss current limitations of our implementation as well as multiple interesting avenues of future research.

7.1. Limitations

While the contribution of our work resides within data-driven reporting, our work requires the scaffolding of an analytical system to provide the necessary functionality. We chose to build on the existing Vistrates [BMR*19] and Codestrates [RNA*17] frameworks for this purpose, but our efforts understandably still fall short in certain aspects compared to mature data analysis systems such as Tableau. Specifically, user-friendliness of the analysis system was touched upon by the expert reviewers (Section 6). This is a limitation that could be remedied by integrating our data-driven reporting method in other analytical environments in the future. In addition, InsideInsights provides visual feedback about how an annotation connects to the pipeline, but the visual feedback in the reverse direction can be improved. While annotations update when the user interacts with pipeline components, this can happen out of sight of the user when the narration hierarchy becomes sufficiently large. To overcome this issue, improved visual indicators for occurring changes could make such connections apparent.

Another challenge is how to help non-experts navigate the dynamic insight hierarchy. We currently represent the insight hierarchy as indented text cells, but this design may have certain limitations with respect to scalability. Although collapsing branches of the hierarchy allow users to keep cells of interest within view, indenting cells exceedingly have visual limitations. Thus, exploring alternative cell layouts or interaction methods becomes important. On a conceptual level, the grouping of related annotations supports the user in keeping the hierarchy of a manageable size, but keeping entirely different analysis paths as subbranches may make the hierarchy too deep. Supporting the user to maintain different variations of a story may be a way to address this limitation, e.g., by having several hierarchies or utilize different cell types (cf. Section 6.2). In other words, our novel integration of data-driven reporting is also yielding interesting design challenges for future work.

Finally, our experts also noted that the freedom afforded by InsideInsights can be a burden to the analyst, and that some structured guidance would be helpful. Supporting insight generation [CBYE18, WMA*16] and automatically inferring visual analytic activities from user interactions [GZ09] are already active research focuses. Combining these methods with support for generating comprehensive annotations could be interesting.

7.2. Implications for Visual Analytics

Recent developments on interactive notebooks [She14] have set a trend towards reproducible and shareable data analytics. InsideInsights continues this trend by allowing the creation of data-driven reports that can be accessed by many different users, thereby making data analytics accessible to a wider audience. The interaction methodology we propose can essentially change how analysts and stakeholders collaborate by supporting the creation of common ground, which is a vital part of collaborative data analysis [HA08].

While our current prototype is built on top of a collaborative notebook [RNA*17] and a visual analytics component model [BMR*19], the design itself is independent of any data analysis system that exposes its internal state. This can be achieved in at least two general ways: either by directly supporting provenance tracking through an API to navigate and reactivate states, or by exposing declarative specifications for the internal state, such as as in Vega [SWH14] and Vega-lite [SMWH17b]. The latter is exactly the type of system our current prototype is built upon. By exposing their internal state in this way, future data analytics applications can more easily become part of larger data analysis ecologies.

7.3. Towards Literate Analytics

The key contribution of InsideInsights is to bridge data-driven narration, provenance, and collaboration to assist users in organizing and understanding findings at variable levels of detail. The long-term goal behind our work is to empower users with the ability to dynamically structure their analysis to promote comprehension of increasingly complex data and algorithms. Similar to how the motivation behind *literate programming* was to write programs not only for the sake of the computer but also to promote human understanding [Knu84], there is an increasing focus to promote human understanding of data analytics [WKM02, CH18, KPN16]. Accordingly, we think the method presented in this paper is a step on the way to a new paradigm we tentatively call *literate analytics.*

The descendants of literate programming [MP14, WKD19] are a testament to the unique way narration and annotation can support human comprehension. *Literate computing* [MP14] extended the literate programming concept by combining narrative with executable code. However, current literate computing solutions (e.g., notebooks [KRKP*16, Obs18, Goo18, RNA*17]) do not support the creation of hierarchical structures within the document—an important concept of literate programming. Our work combines hierarchies with interactive analytics within the same document.

The goal in our notion of literate analytics is to document the entire data analysis rationale, including insights about the method and the data as well as interpretations. While our method support the capturing of such documentation, our expert review revealed that it is also important to support completeness of the documentation. In general, we characterize literate analytics as integrated narration and analysis. This integration is manifested in the concept of live documentation, where annotations always match the state of the analysis they aim to describe, and thus enable interactive reading of data analytics. To support this, it becomes important that users understand how the documentation changes when interacting with the analysis, which is another aspect where our current approach can be extended. In addition, literate analytics aims to support documentation throughout the process such that the current progress at any time can be shared and replicated. Although this paper is only an initial step, we hope that our work will help spark future discussions about what literate analytics should incorporate.

8. Conclusion

We have proposed InsideInsights, an exploration of the design for integrated data-driven reporting, where insights are organized into an information hierarchy linked to analytic provenance and presentation views. Our system allows analysts to tag views, items, and entire states of a visualization and computational components with annotations as part of their analysis. We have also presented two scenarios and results from an expert review to demonstrate the validity of the idea. The novel combination of literate computing, provenance tracking, and storytelling elements have the potential to bridge the gap between data analysis and reporting, thus pointing towards a new paradigm we tentatively call “literate analytics.”

Acknowledgments

This work was supported by the DABAI project (IFD-5153-00004B), the Stibo travel grant, and the DFG grant DA 1319/3-3.

References

[ATM09] AMADIEU F., TRICOT A., MARINÉ C.: Interaction between prior knowledge and concept-map structure on hypertext comprehension, coherence of reading orders and disorientation. *Interacting with computers 22*, 2 (2009), 88–97. doi:10.1016/j.intcom.2009.07.001. 4

[BCC*05] BAVOIL L., CALLAHAN S. P., CROSSNO P. J., FREIRE J., SCHEIDEGGER C. E., SILVA C. T., VO H. T.: VisTrails: Enabling interactive multiple-view visualizations. In *Proceedings of the IEEE Conference on Visualization* (Piscataway, NJ, USA, 2005), IEEE, pp. 135–142. doi:10.1109/VISUAL.2005.1532788. 2

[BE18] BATCH A., ELMQVIST N.: The interactive visualization gap in initial exploratory data analysis. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 278–287. doi:10.1109/TVCG.2017.2743990. 3

[BMR*19] BADAM S. K., MATHISEN A., RÄDLE R., KLOKMOSE C. N., ELMQVIST N.: Vistrates: A component model for ubiquitous analytics. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 586–596. doi:10.1109/TVCG.2018.2865144. 3, 5, 7, 9, 10

[BPW*93] BRODLIE K., POON A., WRIGHT H., BRANKIN L., BANECKI G., GAY A.: GRASPARC: a problem solving environment integrating computation and visualization. In *Proceedings of the IEEE Conference on Visualization* (Piscataway, NJ, USA, 1993), IEEE, pp. 102–109. doi:10.1109/VISUAL.1993.398857. 2, 3

[BRK18] BOROWSKI M., RÄDLE R., KLOKMOSE C. N.: Codestrate packages: An alternative to one-size-fits-all software. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), ACM, p. LBW103. doi:10.1145/3170427.3188563. 7

[Car08] CARPENDALE S.: Evaluating information visualizations. In *Information Visualization – Human-Centered Issues and Perspectives*, vol. 4950 of *Lecture Notes in Computer Science*. Springer, London, UK, 2008, pp. 19–45. doi:10.1007/978-3-540-70956-5_2. 9

[CBYE18] CUI Z., BADAM S. K., YALÇIN A., ELMQVIST N.: DataSite: Proactive visual data exploration with computation of insight-based recommendations. *CoRR abs/1802* (2018). URL: http://arxiv.org/abs/1802.08621. 2, 10

[CH18] CONLEN M., HEER J.: Idyll: A markup language for authoring and publishing interactive articles on the web. In *Proceedings of the ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2018), ACM, pp. 977–989. doi:10.1145/3242587.3242600. 2, 3, 4, 10

[DF08] DAVIDSON S. B., FREIRE J.: Provenance and scientific workflows: challenges and opportunities. In *Proceedings of the ACM Conference on Management of Data* (New York, NY, USA, 2008), ACM, pp. 1345–1350. doi:10.1145/1376616.1376772. 2

[Gam95] GAMMA E.: *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995. 7

[GLG*16] GRATZL S., LEX A., GEHLENBORG N., COSGROVE N., STREIT M.: From visual exploration to storytelling and back again. *Computer Graphics Forum 35*, 3 (2016), 491–500. doi:10.1111/cgf.12925. 3, 4

[Goo18] GOOGLE: Colaboratory, 2018. URL: https://colab.research.google.com/. 3, 10

[GP01] GERSHON N., PAGE W.: What storytelling can do for information visualization. *Communications of the ACM 44*, 8 (2001), 31–37. doi:10.1145/381641.381653. 2, 5

[GRSG17] GREENBAUM D., ROZOWSKY J., STODDEN V., GERSTEIN M.: Structuring supplemental materials in support of reproducibility. *Genome Biology 18*, 1 (2017), 64. doi:10.1186/s13059-017-1205-3. 2, 4

[GS12] GUO P. J., SELTZER M. I.: Burrito: Wrapping your lab notebook in computational infrastructure. In *Proceedings of the USENIX Workshop on the Theory and Practice of Provenance* (Boston, MA, USA, 2012), USENIX, pp. 7–10. 2

[GZ09] GOTZ D., ZHOU M. X.: Characterizing users' visual analytic activity for insight provenance. *Information Visualization 8*, 1 (2009), 42–55. doi:10.1109/VAST.2008.4677365. 2, 10

[HA08] HEER J., AGRAWALA M.: Design considerations for collaborative visual analytics. *Information Visualization 7*, 1 (2008), 49–62. doi:10.1057/palgrave.ivs.9500167. 3, 4, 10

[HD11] HULLMAN J., DIAKOPOULOS N.: Visualization rhetoric: Framing effects in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2231–2240. doi:10.1109/TVCG.2011.255. 2, 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. doi:10.1109/TVCG.2013.119. 2

[Hor98] HORN R. E.: Structured writing as a paradigm. In *Instructional Development: State of the Art*. Educational Technology Publications, Englewood Cliffs, NJ, USA, 1998. 4

[HVW09] HEER J., VIÉGAS F. B., WATTENBERG M.: Voyagers and voyeurs: Supporting asynchronous collaborative visualization. *Communications of the ACM 52*, 1 (2009), 87–97. doi:10.1145/1435417.1435439. 3, 4

[KEB*15] KLOKMOSE C. N., EAGAN J. R., BAADER S., MACKAY W., BEAUDOUIN-LAFON M.: Webstrates: shareable dynamic media. In *Proceedings of the ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2015), ACM, pp. 280–290. doi:10.1109/cmv.2007.20. 7

[KHM17] KERY M. B., HORVATH A., MYERS B. A.: Variolite: Supporting exploratory programming by data scientists. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), ACM, pp. 1265–1276. doi:10.1145/3025453.3025626. 2

[Kir10] KIRSH D.: Thinking with external representations. *Ai & Society 25*, 4 (2010), 441–454. doi:10.1007/s00146-010-0272-8. 2, 3

[Knu84] KNUTH D. E.: Literate programming. *The Computer Journal 27*, 2 (1984), 97–111. doi:10.1093/comjnl/27.2.97. 10

[KPHH12] KANDEL S., PAEPCKE A., HELLERSTEIN J. M., HEER J.: Enterprise data analysis and visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2917–2926. doi:10.1109/TVCG.2012.219. 1, 2, 3, 4

[KPN16] KRAUSE J., PERER A., NG K.: Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2016), ACM, pp. 5686–5697. doi:10.1145/2858036.2858529. 10

[KRA*18] KERY M. B., RADENSKY M., ARYA M., JOHN B. E., MYERS B. A.: The story in the notebook: Exploratory data science using a literate programming tool. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), ACM, pp. 174:1–174:11. doi:10.1145/3173574.3173748. 2, 3, 4

[KRKP*16] KLUYVER T., RAGAN-KELLEY B., PÉREZ F., GRANGER B. E., BUSSONNIER M., FREDERIC J., KELLEY K., HAMRICK J. B., GROUT J., CORLAY S., IVANOV P., AVILA D., ABDALLA S., WILLING C.: Jupyter notebooks – a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (Amsterdam, Netherlands, 2016), IOS Press, pp. 87–90. doi:10.3233/978-1-61499-649-1-87. 3, 4, 10

[LRIC15] LEE B., RICHE N. H., ISENBERG P., CARPENDALE S.: More than telling a story: Transforming data into visually shared stories. *IEEE* *computer graphics and applications 35*, 5 (2015), 84–90. doi:10.1109/MCG.2015.99. 2, 3, 5

[MG17] MATHISEN A., GRØNBÆK K.: Clear visual separation of temporal event sequences. In *Visualization in Data Science Symposium at IEEE VIS* (2017). 7, 9

[Min09] MINTO B.: *The Pyramid Principle: Logic in Writing and Thinking*. Pearson Education, Upper Saddle River, NJ, USA, 2009. 2, 4

[MNV16] MÉNDEZ G. G., NACENTA M. A., VANDENHESTE S.: iVoLVER: Interactive visual language for visualization extraction and reconstruction. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2016), ACM, pp. 4073–4085. doi:10.1145/2858036.2858435. 3

[MP14] MILLMAN K. J., PÉREZ F.: Developing open-source scientific practice. In *Implementing Reproducible Research*. CRC Press, Boca Raton, FL, USA, 2014. 10

[MT14] MAHYAR N., TORY M.: Supporting communication and coordination in collaborative sensemaking. *IEEE Transactions on Visualization & Computer Graphics*, 12 (2014), 1633–1642. doi:0.1109/tvcg.2014.2346573. 2, 3, 4

[NK18] NOUWENS M., KLOKMOSE C. N.: The application and its consequences for non-standard knowledge work. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), ACM, pp. 399:1–399:12. doi:10/ct5q. 3

[Nor06] NORTH C.: Toward measuring visualization insight. *IEEE Computer Graphics and Applications 26*, 3 (2006), 6–9. doi:10.1109/MCG.2006.70. 3

[Obs18] OBSERVABLEHQ: Observable, 2018. URL: https://beta.observablehq.com. 3, 6, 10

[PC05] PIROLLI P., CARD S.: The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of the International Conference on Intelligence Analysis* (McLean, VA, USA, 2005), vol. 5, The MITRE Corporation, pp. 2–4. 1, 2, 3

[Pla04] PLAISANT C.: The challenge of information visualization evaluation. In *Proceedings of the ACM Conference on Advanced Visual Interfaces* (New York, NY, USA, 2004), ACM, pp. 109–116. doi:10.1145/989863. 9

[RDTH18] RULE A., DROSOS I., TABARD A., HOLLAN J. D.: Aiding collaborative reuse of computational notebooks with annotated cell folding. *Proceedings of the ACM on Human-Computer Interaction 2*, CSCW (2018), 150. doi:10.1145/3274419. 4

[RESC16] RAGAN E. D., ENDERT A., SANYAL J., CHEN J.: Characterizing provenance in visualization and data analysis: an organizational framework of provenance types and purposes. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 31–40. doi:10.1109/TVCG.2015.2467551. 2

[RNA*17] RÄDLE R., NOUWENS M., ANTONSEN K., EAGAN J. R., KLOKMOSE C. N.: Codestrates: Literate computing with webstrates. In *Proceedings of the ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2017), ACM, pp. 715–725. doi:10.1145/3126594.3126642. 3, 5, 7, 9, 10

[RRH*19] ROMAT H., RICHE N., HINCKLEY K., LEE B., APPERT C., PIETRIGA E., COLLINS C.: Activeink:(th) inking with data. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2019), ACM. 2, 3

[RTH18] RULE A., TABARD A., HOLLAN J.: Exploration and explanation in computational notebooks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), ACM, pp. 32:1–32:12. doi:10.1145/3173574.3173606. 2, 3, 4

[SH10] SEGEL E., HEER J.: Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1139–1148. doi:10.1109/TVCG.2010.179. 2, 5

[SH14] SATYANARAYAN A., HEER J.: Authoring narrative visualizations with Ellipsis. *Computer Graphics Forum 33*, 3 (2014), 361–370. doi:10.1111/cgf.12392. 3

[Sha08] SHARMA N.: Sensemaking handoff: When and how? *Proceedings of the American Society for Information Science and Technology 45*, 1 (2008), 1–12. doi:10.1002/meet.2008.1450450234. 2, 3, 4

[She14] SHEN H.: Interactive notebooks: Sharing the code. *Nature News 515*, 7525 (2014), 151. doi:10.1038/515151a. 10

[SMWH17a] 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. doi:10.1109/TVCG.2016.2599030. 3

[SMWH17b] 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. doi:10.1109/TVCG.2016.2599030. 10

[Sof18] SOFTWARE T.: Tableau, 2018. https://www.tableau.com/, accessed September 2018. 2, 3

[ST15] SARVGHAD A., TORY M.: Exploiting analysis history to support collaborative data analysis. In *Proceedings of the Graphics Interface Conference* (2015), Canadian Information Processing Society, pp. 123–130. doi:10.20380/GI2015.16. 2, 4

[SVLV16] SARRAFZADEH B., VTYURINA A., LANK E., VECHTOMOVA O.: Knowledge graphs versus hierarchies: An analysis of user behaviours and perspectives in information seeking. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval* (2016), ACM, pp. 91–100. doi:10.1145/2854946.2854958. 4

[SvW08] SHRINIVASAN Y. B., VAN WIJK J. J.: Supporting the analytical reasoning process in information visualization. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), ACM, pp. 1237–1246. doi:10.1145/1357054.1357247. 2, 3

[SWH14] SATYANARAYAN A., WONGSUPHASAWAT K., HEER J.: Declarative interaction design for data visualization. In *Proceedings of the ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2014), ACM, pp. 669–678. doi:10.1145/2642918.2647360. 3, 10

[TC05] THOMAS J. J., COOK K. A.: *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press, Los Alamitos, CA, USA, 2005. 2, 3

[TM05] TORY M., MÖLLER T.: Evaluating visualizations: Do expert reviews work? *IEEE Computer Graphics and Applications 25*, 5 (2005), 8–11. doi:10.1109/MCG.2005.102. 9

[TME08] TABARD A., MACKAY W. E., EASTMOND E.: From individual to collaborative: the evolution of Prism, a hybrid laboratory notebook. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (New York, NY, USA, 2008), ACM, pp. 569–578. doi:10.1145/1460563.1460653. 3

[Tuk77] TUKEY J. W.: *Exploratory Data Analysis*, vol. 2. Addison-Wesley, Reading, MA, USA, 1977. 1, 2

[UPSK18] UFFORD M., PACER M., SEAL M., KELLEY K.: Beyond interactive: Notebook innovation at Netflix, 2018. URL: https://medium.com/netflix-techblog/notebook-innovation-591ee3221233. 2, 3

[VW06] VIÉGAS F. B., WATTENBERG M.: Communication-minded visualization: A call to action. *IBM Systems Journal 45*, 4 (Apr. 2006), 801–812. doi:10.1147/sj.454.0801. 3

[VWVH*07] VIÉGAS 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), 1121–1128. doi:10.1109/TVCG.2007.70577. 3, 4

[WHHA11] WILLETT W., HEER J., HELLERSTEIN J. M., AGRAWALA M.: CommentSpace: structured support for collaborative visual analysis. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), ACM, pp. 3131–3140. doi:10.1145/1978942.1979407. 9

[WKD19] WOOD J., KACHKAEV A., DYKES J.: Design exposition with literate visualization. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019). To appear. doi:10.1109/TVCG.2018.2864836. 10

[WKM02] WEBER A., KIENLE H. M., MÜLLER H. A.: Live documents with contextual, data-driven information components. In *Proceedings of the ACM Conference on Computer Documentation* (New York, NY, USA, 2002), ACM, pp. 236–247. doi:10.1145/584955.584990. 2, 3, 10

[WMA*16] WONGSUPHASAWAT K., MORITZ D., ANAND A., MACKINLAY J., HOWE B., HEER J.: Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 1–1. doi:10.1109/TVCG.2015.2467191. 10

[YEB18] YALÇIN M. A., ELMQVIST N., BEDERSON B. B.: Keshif: Rapid and expressive tabular data exploration for novices. *IEEE Transactions on Visualization and Computer Graphics 24*, 8 (2018), 2339–2352. doi:10.1109/TVCG.2017.2723393. 3

[ZGI*18] 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 (2018), 340–350. doi:10.1109/tvcg.2017.2745279. 2, 3, 4