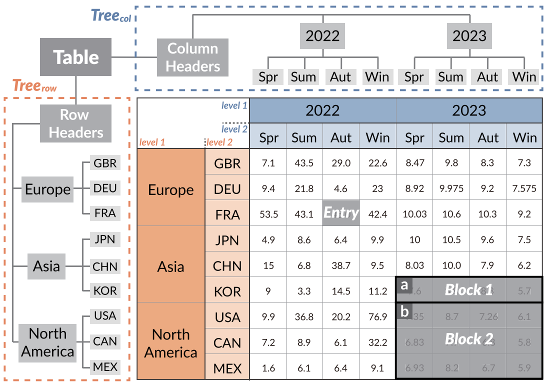
CoInsight: Visual Storytelling for Hierarchical Tables With Connected Insights

Guozheng Li, Runfei Li, Yunshan Feng, Yu Zhang, Yuyu Luo, and Chi Harold Liu

***Abstract*—Extracting data insights and generating visual data stories from tabular data are critical parts of data analysis. However, most existing studies primarily focus on tabular data stored as flat tables, typically without leveraging the relations between cells in the headers of hierarchical tables. When properly used, rich table headers can enable the extraction of many additional data stories. To assist analysts in visual data storytelling, an approach is needed to organize these data insights efficiently. In this work, we propose CoInsight, a system to facilitate visual storytelling for hierarchical tables by connecting insights. CoInsight extracts data insights from hierarchical tables and builds insight relations according to the structure of table headers. It further visualizes related data insights using a nested graph with edge bundling. We evaluate the CoInsight system through a usage scenario and a user experiment. The results demonstrate the utility and usability of CoInsight for converting data insights in hierarchical tables into visual data stories.**

***Index Terms*—Data insight, hierarchical table, table data visualization, tabular data, visual storytelling.**



level 1	level 2	2022 Spr	2022 Sum	2022 Aut	2022 Win	2023 Spr	2023 Sum	2023 Aut	2023 Win
Europe	GBR	7.1	43.5	29.0	22.6	8.47	9.8	8.3	7.3
	DEU	9.4	21.8	4.6	23	8.92	9.975	9.2	7.575
	FRA	53.5	43.1	Entry	42.4	10.03	10.6	10.3	9.2
Asia	JPN	4.9	8.6	6.4	9.9	10	10.5	9.6	7.5
	CHN	15	6.8	38.7	9.5	8.03	10.0	7.9	6.2
	KOR	9	3.3	14.5	11.2	[illegible]	Block 1	[illegible]	5.7
North America	USA	9.9	36.8	20.2	76.9	[illegible]	8.7	7.26	6.1
	CAN	7.2	8.9	6.1	32.2	6.83	Block 2	[illegible]	5.8
	MEX	1.6	6.1	6.4	9.1	6.93	8.2	6.7	5.9

Fig. 1. The labels of a hierarchical table form a tree structure. Row and column headers form subtrees. (a) indicates a data block with a column locator (*2023*, ∗) and a row locator (*Asia*, *KOR*). (b) indicates a data block with a column locator (*2023*, ∗) and a row locator (*North America*, ∗).

I. Introduction

TABULAR data has emerged as a widely used method for organizing real-world data, with wide applicability in various fields [1], [2], [3], [4], [5], [6], [7], [8], [9]. To obtain meaningful information underlying the data, discovering data insights such as interesting data patterns like trends and outliers is a common and significant analytical task [10]. Furthermore, connecting data insights through narrative visualizations to construct visual data stories can improve data comprehension [11], [12].

Existing studies on data insight extraction and data story generation mainly focus on flat tables [13], [14], [15], [16], [17], [18], [19]. By incorporating a hierarchical structure [20], [21], [22] in table headers, hierarchical tables can present data more effectively in a two-dimensional space [1], [2], [23], [24]. Hierarchical tables are widely used in many application scenarios, including government statistical reports and scientific experiment records [25]. As shown in Fig. 1, the headers of hierarchical tables consist of nominal and temporal data fields with interrelationships, *e.g.*, Europe and GBR. A common insight extraction method is based on enumeration, which computationally traverses the underlying multi-dimensional data subsets and finds candidate data insights from various analytical perspectives. The data fields in the hierarchical table headers of the underlying multi-dimensional data often enlarge the enumeration space of data insights. Therefore, algorithms potentially extract many data insights from hierarchical tables. However, existing methods do not fully leverage the relations between cells in hierarchical headers. In addition, data storytelling requires connecting data insights into a cohesive narrative to serve the user's communication objectives [12]. Therefore, visual storytelling for hierarchical tables is a challenging task, and an effective strategy for organizing abundant data insights is beneficial.

To bridge the above gaps, our key idea is to discern the relations inherent in hierarchical table headers, based on which we can connect the insights extracted from hierarchical tables to structure storytelling systematically. To this end, we propose CoInsight, a system to facilitate the visual storytelling of hierarchical tables by constructing an insight graph. Insight graph construction is to extract and connect insights. Because the hierarchical tables can be organized in different states by different

Manuscript received 1 October 2023; revised 1 December 2023; accepted 22 December 2023. Date of publication 15 April 2024; date of current version 19 June 2024. This work was supported in part by the National Key R&D Program of China under Grant 2021YFB3301500, in part by NSFC under Grant 62302038 and Grant U2268205, and in part by China Railway Group Company Ltd. Science and Technology Research and Development Program under Grant N2022J014. Recommended for acceptance by N. Elmqvist, S. Liu, and V. Pascucci. *(Corresponding author: Yuyu Luo.)*

Guozheng Li, Runfei Li, Yunshan Feng, and Chi Harold Liu are with the Beijing Institute of Technology, Beijing 100811, China (e-mail: guozheng.li@bit.edu.cn; chiliu@bit.edu.cn).

Yu Zhang is with the University of Oxford, OX1 2JD Oxford, U.K.

Yuyu Luo is with the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (e-mail: yuyuluo@hkust-gz.edu.cn).

This article has supplementary downloadable material available at http://doi.org/10.1109/TVCG.2024.3388553, provided by the authors.

Digital Object Identifier 10.1109/TVCG.2024.3388553

table transformations, the insight extraction is based on various states of hierarchical tables. For each state of a hierarchical table, we define the search space and traverse the space to extract data insights. The insight relation construction is to build the relations between extracted insights to guide users' explorations and facilitate narratives. In hierarchical tables, each extracted data insight corresponds to a data block identified by a cell in the row and column headers, as shown in Fig. 1(a). Based on whether the extracted insights are in the same state, we identify two kinds of relations: intra-state and inter-state relations. The CoInsight system visualizes the insight graph using the nested graph visual representation with edge bundling.

We evaluate the effectiveness of the CoInsight system through a usage scenario and a user experiment. First, the usage scenario based on a console sales dataset showcases that CoInsight can enhance users' capability of exploration and visual storytelling for hierarchical tables. Additionally, we compare CoInsight and two insight extraction techniques designed for flat tables: PowerBI QuickInsight [10] and AWS QuickSight [26]. Participants were tasked with discovering compelling data stories from the table within a predefined time frame. Subsequently, we engaged visualization experts to evaluate the quality of the generated visual data stories. The experiment results highlight the effectiveness of CoInsight in enabling users to uncover longer and more coherent data narratives compared to other techniques.

In summary, the main contributions of this paper are as follows:

- An insight graph construction framework to extract and connect insights from hierarchical tables, comprised of the insight extraction and relation construction modules.
- The CoInsight prototype system is designed to assist users in visual storytelling for hierarchical tables.
- A user experiment and a usage scenario to demonstrate the utility and usability of CoInsight for connecting insights into data stories.

CoInsigh is available at https://github.com/bitvis2021/CoInsighthttps://github.com/bitvis2021/CoInsight.

II. Related Work

In this section, we review the related studies on automatic data visualization and visual storytelling.

A. Automatic Data Visualization

Many studies focus on automatic data visualization for tabular data. We categorize them into two classes: insight-driven and non-insight-driven.

Insight-driven: Some automatic data visualization (*a.k.a.* visualization recommendation) approaches for visualizations are built upon insights. After mining data insights, the corresponding visual representations are created based on established rules. SeeDB [27] defines an interesting visualization as one that greatly differs from a reference provided by a user. In contrast, Zenvisage [28] aims to identify interesting visualizations similar to user-specified patterns. Foresight [29] identifies six types of data patterns as insights, whereas Tang et al. [30] mainly focus on outstanding values and trends. Both of these two works utilize insight metrics (scores) to rank the generated results. On the contrary, BigIN4 [31] enables users to interactively identify insight types and specify the thresholds based on their specific analysis needs. The approaches above lack a unified definition of insights and rely on a straightforward enumeration based on experiential categorization. Compared to them, QuickInsights [10] introduces a comprehensive formulation of data insights, and MetaInsight [32] extends it by incorporating highlights. MetaInsight also finds ordinaries and exceptions among basic insights that share certain relations to achieve structured knowledge representation. Unlike the studies above, we aim to generate connected insights and systematically structure storytelling by capturing relations in the hierarchical table headers.

Non-insight-driven: Many approaches do not take into account the insights derived from the data. Instead, they rely on the dataset itself or statistical information as the basis for making these recommendations. By translating some manual heuristics for visualization into a series of rules [33], some systems can automatically create visualizations to assist users in exploring datasets. Based on criteria of expressiveness and effectiveness, APT [34] focuses on recommending and ranking visual encodings of a single view. On the other hand, Show Me [35] proposes some novel heuristics to support the automatic construction of small multiple views. Voyager [36], [37] extends this line of work by recommending variable selections and data transformations and also supports interactive browsing and editing of the recommended results. Although these rule-based methods are effective in specific scenarios, they have other issues such as time-consuming rule design and limited generalizability. To overcome the above limitations, researchers also utilized ML-based strategies. DeepEye [14], [15] employs a decision tree to select data mapping strategies and ranks the visualizations by leveraging either learning-to-rank [38] or a rule-based partial order approach. Additionally, DeepEye [39], [40] supports the automatic creation of visualizations in response to user keyword searches. Expanding upon these capabilities, Luo et al. [41] develop a transformer-based model named ncNet, designed to facilitate the visualization authoring using natural language, moving beyond mere keyword-based queries. ncNet is trained on a large-scale natural language to visualization benchmark [42]. Draco [43] trains models using experimental data and incorporates domain knowledge in visual design as constraints. Different from the above two studies regarding learning tasks, data quantity and data quality. Vizml [44] trains models using a large corpus of datasets to learn design choices without considering data queries. In contrast, Table2Charts [45] handles both design choices and data queries based on a deep Q-learning network. Our work first derives novel and interconnected insights from the hierarchical tables and then introduces a graph-based module to assist users in visual storytelling.

B. Visual Storytelling

Data stories present a sequence of story pieces that uses visualizations to communicate data insights [46]. Visual storytelling has been gaining growing interest from researchers in the visualization community. Recent studies have conducted extensive investigations of storytelling and narrative visualization techniques [47], [48]. After analyzing a curated collection of recent data-driven stories, Stolper et al. [12] outline four key goals in the design of data stories: enabling communication and explanation, building connections about story elements, improving navigation, and supporting controlled exploration. Zhao and Elmqvist [49] present a taxonomy according to the format of data stories and divide them into six types, including audience, data, and media. Crafting a data-driven story is challenging for users, which motivates the development of various authoring tools. For instance, Ellipsis [50] is a general tool for authoring narrative visualizations, combining various elements such as dynamic annotations and decoupled coordination of visualization components. Some authoring tools are designed to generate a certain type of narrative visualization. For instance, InfoNice [51] is specially designed for creating data-driven infographics for visual storytelling. Brehmer et al. build a design space specifically for timeline-based visualizations [52] and divide them into three dimensions: layout, scale, and representation. Furthermore, they developed Timeline Storyteller [53] to illustrate various aspects of sequential data.

Similar to our work, many studies are designed for the visual storytelling of tabular data. The first category is to connect visualizations based on their visual encodings. For example, GraphScape [54] builds connections between visualizations based on the edit operations to represent their similarity. Furthermore, Chart Constellations [55] incorporates keyword taggings and dimensional intersections into measuring similarity between charts, thereby providing a more comprehensive result. Our method provides a different perspective to connect data visualizations based on the relations of underlying data. Recent studies also focus on automatic story-generation techniques. DataShot [56] organizes visualizations into topics based on data insights as a fact sheet to tell a data story. Based on the defined data facts, Calliope [13] further identifies six types of logical relations and utilizes these to organize the generated story pieces. Unlike the two works above that automatically generate data facts, ChartStory [57] directly takes charts pre-generated by users as inputs and then recommends partition, layout, and caption of story pieces to create a complete narrative.

Compared to our method, existing studies about visual storytelling are not designed for hierarchical tables and do not fully leverage the relations between cells in table headers.

III. Insight Graph Construction

In this section, we first present the data model for the hierarchical table, then introduce the details of CoInsight, including table transformation, insight extraction, and relation construction.

A. Background: Data Model

A table comprises interconnected elements that can be categorized as *entries* and *labels* [58].

Entries indicate quantitative data items in the table content, and labels are located in table headers and utilized to specify entries. We classify labels into row headers and column headers based on their placement in tables.

A tree structure is formed by the labels of a hierarchical table, where both row and column headers form subtrees, as illustrated in Fig. 1. Specifically, we denote the subtree formed by row headers as Tree$_{row}$ and the one formed by column headers as Tree$_{col}$. We define a *level* attribute for each label, indicating its depth in the structure of the tree.

Based on the associations between labels and entries of hierarchical tables [59], we can use two paths from the root node to the leaf node in Tree$_{row}$ and Tree$_{col}$ to locate an entry. We employ a label sequence to denote each path, and each element in the sequence corresponding to a node within the tree. As a result, an entry can be specified by two sequences, denoted as Seq$_{row}$ and Seq$_{col}$. Because the notation of Seq$_{col}$ is similar to Seq$_{row}$, we only define and explain Seq$_{row}$ below.

$$Seq_{row} = (label_1, label_2, \ldots, label_k), \tag{1}$$

where k is the level of the leaf node in Tree$_{row}$, and label$_i$ refers to the label at level i in the aforementioned sequence.

Data Block: A *data block* is composed of multiple continuous entries (see Fig. 1). Based on the entry locators, we define Loc$_{row}$ and Loc$_{col}$ to specify data blocks. Each locators of a data block consists of a set of entry locators. Formally, the notation of Loc$_{row}$ is as follows:

$$Loc_{row} = [Seq_{row_1}, Seq_{row_2}, \ldots, Seq_{row_n}]. \tag{2}$$

In the above notation, n represents the number of entries within the block. Furthermore, we design a rule to simplify the block locators by merging the sequences to all the child nodes in a subtree as a wildcard (denoted as $*$). For example, the Loc$_{row}$ of the highlighted block (b) in Fig. 1 is (*North America*, $*$).

Data Scope: We define *data scope* as a special kind of data block. In particular, both the row and column locators of data scope consist of only one label sequence after simplifications. The notation of the row locator of a data scope (Loc$_{row}$) is as follows. For instance, the Loc$_{row}$ of the data block (b) shown in Fig. 1 is (*North America*, $*$) and the Loc$_{col}$ is (*2023*, $*$). Since the definition conforms to Notation 3, the data block (b) can be regarded as a data scope.

$$Loc_{row} = (label_1, label_2, \ldots, (label_k \; or \; *)), \; k \geq 1. \tag{3}$$

To obtain all data scopes within a hierarchical table, we enumerate the labels in both Tree$_{row}$ and Tree$_{col}$, extracting a series of label sequences. Within a data scope, each entry has multiple attributes. One attribute is the value of table entries, while the other attributes are derived from their corresponding labels. For example, the highlighted entry in Fig. 1 has five attributes, including two nominal attributes from the row headers, two temporal attributes from the column headers, and one quantitative attribute from the entry itself.

Related Entries: In addition to using labels to locate entries, as mentioned above, we also use labels to define relations between entries. More specifically, a group of entries that share identical labels are related. The number of identical labels determines the degree of their relations. The entries within the same block often exhibit relations. For example, the entries in the data block (b) share two common labels *2023* and *North America*, as shown in Fig. 1.

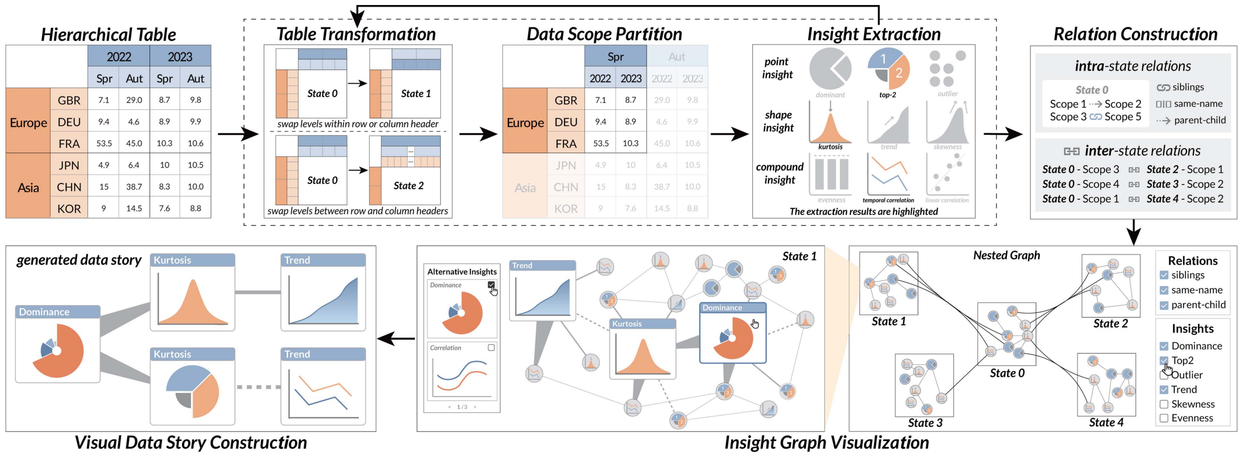
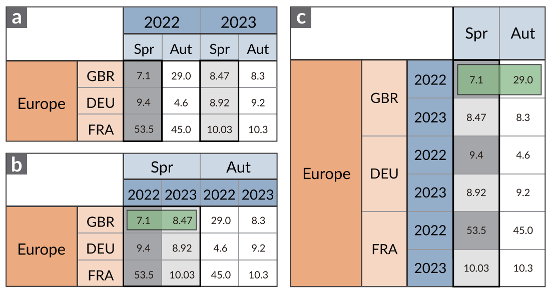


Fig. 2. The pipeline of visual data storytelling for hierarchical tables based on CoInsight. The modules in the row above indicate the insight graph construction process. The modules at the bottom indicate the insight graph visualization and users' explorations. After inputting the hierarchical table, the insight graph construction process conducts data transformations, including swapping levels within row or column headers and swapping levels between row and column headers. The pipeline then traverses all the data scope partition results and extracts data insights for each scope, including point insights, shape insights, and compound insights. The above steps are carried out iteratively. The relation construction module adds relations among all extracted data insights based on the hierarchical header. CoInsight further visualizes the constructed insight graph using a nested graph layout. Users can explore the insights with the guidance of insight graph and generate data stories.


a

		2022		2023	
		Spr	Aut	Spr	Aut
Europe	GBR	7.1	29.0	8.47	8.3
	DEU	9.4	4.6	8.92	9.2
	FRA	53.5	45.0	10.03	10.3

b

		Spr		Aut	
		2022	2023	2022	2023
Europe	GBR	7.1	8.47	29.0	8.3
	DEU	9.4	8.92	4.6	9.2
	FRA	53.5	10.03	45.0	10.3

c

			Spr	Aut
Europe	GBR	2022	7.1	29.0
		2023	8.47	8.3
	DEU	2022	9.4	4.6
		2023	8.92	9.2
	FRA	2022	53.5	45.0
		2023	10.03	10.3

Fig. 3. Three states based on hierarchical table transformations. The state$_a$ is the original hierarchical table highlighting two separated data blocks in dark gray and light gray. The state$_b$ is the transformation result after swapping levels within the column headers. The state$_c$ is the result after moving a level from the column header to the row header. The inter-state relation exists in two data scopes highlighted in green, because they share same labels in row locators.

B. Table Transformation

As mentioned above, a data scope groups related and adjacent entries together. However, some non-adjacent entries may also exhibit correlations in hierarchical tables. For instance, two blocks highlighted in dark gray and light gray in Fig. 3(a) share the same labels *Europe* and *Spr*. The entries in these two blocks are correlated according to the definition in Section III-A, but the positions of these two blocks are not adjacent. Consequently, they cannot form a data scope.

To address the above problem, we attempt to change table headers' hierarchical structure and table content arrangement through table transformations. Through data transformation, new tables can preserve the same entries as the original ones but with different relative positions. We define one type of arrangement for the hierarchical tables as a *state*. Specifically, the original hierarchical table is denoted as state$_0$. Each state has its data model, including Tree$_{row}$ and Tree$_{col}$. We primarily consider the following two types of data transformations.

(1) Swap levels within the row or column headers: This transformation exchanges two levels of row or column headers. As shown in Fig. 3, state$_b$ is the result of swapping *level* 1 and *level* 2 in the column headers of state$_a$. After transformation, two non-adjacent blocks highlighted in state$_a$ are rearranged adjacently in state$_b$, forming a data scope identified by Loc$_{row}$(*Europe*, $*$) and Loc$_{col}$(*Spr*, $*$).

(2) Swap levels between row and column headers: This transformation moves a level from row headers to the last level in column headers or vice versa. For example, as shown in Fig. 3, state$_c$ is the result of moving *level* 1 in column headers to the last level in row headers of state$_a$. As a result, the two highlighted blocks in state$_a$ are rearranged into a continuous data scope in state$_c$. The data scope is identified by Loc$_{row}$ is (*Europe*, $*$) and Loc$_{col}$ is (*Spr*).

C. Insight Extraction

Insights indicate interesting data patterns from a specific perspective [10], assisting analysts in gaining a deeper understanding of the data. Although our work focuses on hierarchical tabular data, the definition of insights is essentially the same as flat tables. We define a data scope as the elementary unit for insight extraction, as explained in Section III-A, because the data scopes obtained by partitioning table entries from different states can preserve the structure of a hierarchical table.

1) Data Processing: The data scope can preserve the structure of hierarchical tables and group related entries together. Consequently, when computing insights based on data scopes, it is essential to take the inherent hierarchical structure into consideration rather than treating all entries within the data scope as equivalent. This section introduces the data processing of data scope before insight calculation to preserve the logical relationships between entries in various aspects.

Aggregation: The locators of a data scope contain wildcards when the labels of the data scope span multiple subtrees or leaf nodes in Tree_{row} or Tree_{col}. Consequently, our method aggregates the entries according to their labels at the level where the wildcard is positioned. Taking the data scope (b) in Fig. 1 as an example, its Loc_{row} is (*North America*, $*$) and Loc_{col} is (*2023*, $*$), with both of them containing wildcards. As a result, we perform aggregation from the perspectives of both row and column headers, respectively. First, we aggregate based on the row headers. Since the wildcard is positioned at $level$ 2, we group the entries based on the labels at $level$ 2 in row headers and calculate the aggregated values, *i.e.*, aggregating the four values for each row within this scope. The aggregation method based on column headers is similar, where we aggregate the three values for each column within this scope. The aggregation strategy may incorporate multiple operations, such as *min*, *max*, *mean*, and *sum*, with the *sum* being the default. Aggregating entries within a data scope based on the hierarchical structure of the labels allows for a comprehensive overview of the data, which expands the insight calculations.

Grouping: Similar to the aggregation operation, the grouping operation also targets data scopes with wildcards in locators. However, the critical difference is that the grouping operation divides a data scope into many finer-grained subsets without performing aggregate functions. We group the entries based on the last label in both Seq_{row} and Seq_{col} of each entry. For example, the data scope in Fig. 1(b) can be grouped based on the label *Canada* in the row headers, meaning the four entries in the second row can be grouped together. It can also be grouped based on the label Sum in the column headers, grouping the three entries in the second column together. After grouping, we can perform inter-group comparative analysis on the entries, facilitating the calculation of compound insights, which will be detailed in Section III-C-2.

2) Insight Calculation: Our insight definitions are built upon existing studies [10], [29], [30]. Following the insight classification in QuickInsights [60], we categorized these insights into three types: point insights, shape insights, and compound insights. We also defined a *score* for each insight to quantify its significance. As mentioned above, insight calculation is performed based on data scopes as the elementary unit. We calculate all kinds of insights for a data scope and record their scores. If the score exceeds the *threshold* (a hyperparameter determined based on existing work or domain expertise), we can consider that the data scope has this insight.

This section provides a detailed explanation of each type of insight, including its semantics and visual representation. Due to space limitations, details of the insight score calculation are given in the supplemental materials. We denote the collection of values within the processed data scope explained in Section III-C-1 as d.

Point insights: describe the characteristic of having several prominent data values within a data scope, including

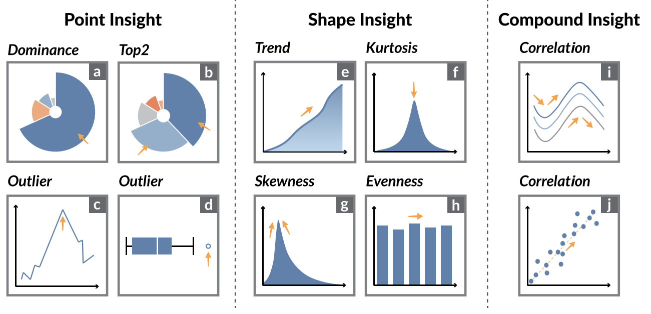


Fig. 4. Three types of extracted data insights: point insight, shape insight, and compound insight. The data insights can be further divided into eight subtypes: dominance, top2, outlier, trend, kurtosis, skewness, evenness, and correlation.

Dominance, *Top2* and *Outlier*. The calculation of *Dominance* and *Top2* insights requires all the entries within the data scope to be non-negative.

- *Dominance:* indicates that the leading value dominates (accounting for more than 50% of the sum of) the scope. We utilize a radial plot to represent this insight, as shown in Fig. 4(a).
- *Top2:* indicates that the leading two values hold a significantly higher proportion than others. Similar to *Dominance*, we also use a radial plot to visualize *Top2* insights, as shown in Fig. 4(b).
- *Outliers:* signify the presence of exceptional values within the data scope, which deviates significantly from the norm. If the corresponding labels in the data scope are temporal, we utilize a line chart to visualize outliers, as shown in Fig. 4(c). On the other hand, we use a box plot if the labels are nominal, as shown in Fig. 4(d).

Shape insights: describe the overall trends or distributions within a data scope, encompassing *Trend*, *Kurtosis*, *Skewness*, and *Evenness*. Specifically, *trend* is calculated only when the labels within the data scope are temporal.

- *Trend:* describes the presence of an upward or downward trend in time-series data. The trend in a data scope is presented using a line chart, as shown in Fig. 4(e).
- *Kurtosis:* focuses on the peakedness of the distribution. A kurtosis insight indicates that the distribution of the data scope is more concentrated around the mean compared to a standard normal distribution. We use a density plot to show the kurtosis insight (Fig. 4(f)).
- *Skewness:* describes the asymmetry in the data distribution. It measures how unevenly data points are distributed around the mean and whether the distribution is skewed to the left or right. Similar to kurtosis, we also use a density plot to represent the skewness insight, as shown in Fig. 4(g).
- *Evenness:* describes whether the distribution of dataset is uniform, *i.e.*, whether data points are evenly spread across the entire data scope. We use bar charts to represent evenness, as shown in Fig. 4(h).

Compound insights: involve comparing different groups of entries within a data scope, primarily focusing on *correlation*.

It helps uncover relationships and dependencies between various data subsets within the scope. Because the calculation of compound insights involves data comparisons, we only compute compound insights for data scopes that have undergone the grouping process mentioned in Section III-C1.

As shown in Fig. 4(i), we utilize the multiple-line chart to visualize the correlation insight if the corresponding data type is temporal. Otherwise, we use a scatter plot to show the correlation insight (Fig. 4(j)).

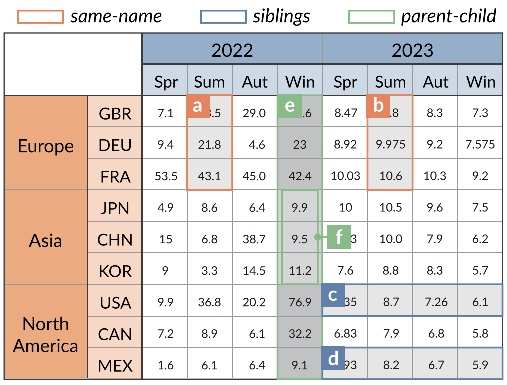
D. Insight Relation Construction

Different data scopes that share the same labels within their locators are related. Therefore, the inherent relations exist between the data scopes within the hierarchical tables at different states. We define the data scope relations based on the relations between locators, thereby reflecting the relations between insights.

We divide insight relations into two categories. The intra-state relations connect insights within the same state, while the inter-state relations involve insights between different states. We transform the relations between insights into the relations between data scopes with the insights and further determine the relations between data scopes based on their locators.

1) Intra-State Relations: Relations between locators: As described in Section III-A, locators of a data scope consist of Loc_{row} and Loc_{col}. We first define the *length* of a locator, which represents the total number of labels excluding wildcards. For example, the length of the locator (*Asia*, $*$) is 1. For the same locators, we define their relations as *identical*. We define their relations as *same-name* for two locators with the same length and their last labels after excluding wildcards are identical. For example, the relation between locator (*2022*, *Spr*) and (*2023*, *Spr*) is *same-name*. For two locators with the same length and only the last label differs, we define their relationship as *siblings*. For example, the relation between locator (*2022*, *Spr*) and (*2023*, *Sum*) are *siblings*. For two locators with inclusion relation for their scope, we define their relationship as *parent-child*. For example, the relation between (*Asia*, $*$) and (*Asia*, *JPN*) is *parent-child*.

Intra-state relations between insights: Based on the above definition of relations between locators, we can obtain relations between Loc_{row} and Loc_{col} for each pair of data scopes. To ensure a close intra-state relation between two data scopes, we specify that the related data scopes should have one identical locator. Consequently, the relation between the other locators exactly indicates the relation between these two data scopes. For example, as shown in Fig. 5, the Loc_{row} of data scope (a) (denoted as DS_a) and (b) (denoted as DS_b) are identical. The Loc_{col} of DS_a is (*2022*, *Sum*) and the Loc_{col} of DS_b is (*2023*, *Sum*). These two locators are in a same-name relation. Therefore, the relation between DS_a and DS_b is defined as same-name. Similarly, DS_c and DS_d share an identical locator along column, and their row locators (*North America, USA*) and (*North America, MEX*) have a sibling relation. Therefore, the relation between DS_c and DS_d is siblings. DS_e and DS_f have an identical column locator (*2022, Win*), while the row locator



same-name
siblings
parent-child
a
b
c
d
e
f

		2022				2023			
		Spr	Sum	Aut	Win	Spr	Sum	Aut	Win
Europe	GBR	7.1	.5	29.0	.6	8.47	.8	8.3	7.3
	DEU	9.4	21.8	4.6	23	8.92	9.975	9.2	7.575
	FRA	53.5	43.1	45.0	42.4	10.03	10.6	10.3	9.2
Asia	JPN	4.9	8.6	6.4	9.9	10	10.5	9.6	7.5
	CHN	15	6.8	38.7	9.5	3	10.0	7.9	6.2
	KOR	9	3.3	14.5	11.2	7.6	8.8	8.3	5.7
North America	USA	9.9	36.8	20.2	76.9	35	8.7	7.26	6.1
	CAN	7.2	8.9	6.1	32.2	6.83	7.9	6.8	5.8
	MEX	1.6	6.1	6.4	9.1	93	8.2	6.7	5.9

Fig. 5. Three relations between the data scopes: same-name relation, siblings relation, and parent-child relation. The scopes denoted (a) and (b) form a same-name relation, as they corresponds to the sales in summers of different years. The scopes denoted (c) and (d) form a siblings relation, as they corresponds to the sales in different regions of North America. The scopes denoted (e) and (f) form a parent-child relation, as (f) corresponds to the sales in a sub-region of (e).

of DS_e is $(*)$ and DS_f is (*Asia*, $*$). Therefore, their relation is defined as parent-child.

2) Inter-State Relations: Table transformations alter the hierarchical structure of row and column headers and generate different states, as explained in Section III-B. As mentioned before, two data scopes with an intra-state relation need to share one identical locator. However, data scopes between different states rarely share identical locators because of the diverse hierarchical structures of table headers in different states. Therefore, we specify that two data scopes in different states are related as long as their locators have at least one identical label. For example, we refer to the data scope highlighted in green in Fig. 3(b) as DS_b, and in Fig. 3(c) as DS_c. DS_b and DS_c have inter-state relations because their row locators share the same labels *Europe* and *GBR*.

IV. The CoInsight System

We have designed and implemented the CoInsight prototype system to facilitate visual data storytelling for hierarchical tables.

A. Design Considerations

We distill four design considerations for the system to assist users in understanding the extracted insights and relations, as well as facilitating the interactive construction of visual data stories. The design considerations are primarily derived from existing literature and common challenges faced by analysts when constructing data stories.

DC1: Present the extracted insights and relations clearly: As described in Section III-C2, the constructed insight graph consists of three insight types and four relation types. To construct a narrative data story, establishing the connections between story elements is essential [12], [57], [61]. Therefore, the system needs to provide an overview [62], [63], [64] and visualize both the insights and relations intuitively to reduce the cognitive burden on users. In addition to displaying the insight types, the

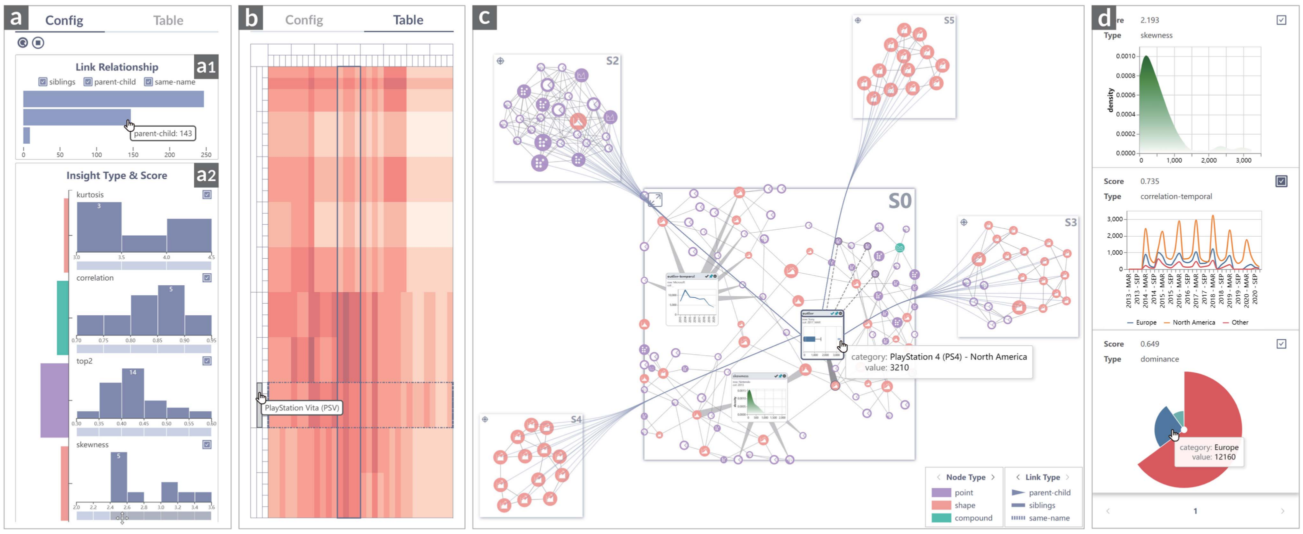


Fig. 6. The user interface of the CoInsight prototype system. (a) Insight distribution panel. (a1) shows the distribution of insight relations, while (a2) presents the distribution of different kinds of insights and the score distribution for each insight. (b) Hierarchical table panel encodes the density of each table cell as the amount of its related insights and highlights the data scope selected by users. (c) Insight graph panel presents insights and their relations in the form of a nested graph with edge bundling. (d) Insight selection panel displays all the insights contained within a data scope.

system should enable users to check the insight details to better understand the data patterns.

DC2: Associate extracted data insights with the original hierarchical tables: It is essential to facilitate and encourage authors to acquire the underlying data [46], which can promote the credibility and transparency of created data stories. As a result, the system should associate the extracted data insights with tables to provide users with context from the original hierarchical tables. As described in Section III-A, our insight extraction method partitions hierarchical tables into different data scopes based on the header structure and treats them as the elementary unit. Therefore, the system needs to enable users to check the corresponding context within the original hierarchical tables.

DC3: Enable users to filter the data insights of interest flexibly: Avoiding information overload is critical to ensure a clear presentation of information to users, which can be realized by incorporating users' interactions [11], [65]. The process of insight graph construction extracts numerous data insights and insight relations. Therefore, the system should support flexible filtering to avoid overwhelming users with abundant extracted insights and insight relations

DC4: Create a data story that visually presents the results of a user's exploration: The outcome of the system should be a comprehensive insight story that visually presents the details of insights and their relations. To support exploration and authoring activities, the system should allow users to save insights, enabling them to construct data stories flexibly during explorations [66], [67].

B. User Interface and Interaction

The user interface of CoInsight consists of four panels: insight distribution panel, hierarchical table panel, insight graph panel, and insight selection panel, as shown in Fig. 6. These panels support the following process of hierarchical table exploration and visual storytelling.

First, the user uploads a hierarchical table to CoInsight to initialize the data exploration. Next, CoInsight creates various states through data transformations, extracts insights from each state, and establishes relations between these insights to build an insight graph. Then, CoInsight visualizes the extracted insights from different aspects. The user can filter the results of insight extraction and check insights in detail.

1) Insight Graph Panel: The insight graph panel shows the extracted insights and insight relations as a nested graph with edge bundling. As shown in Fig. 6(c), each square box in the graph, referred to as a *state card*, corresponds to a state. The state card's unique *state-id* is marked at its upper right corner (*e.g.*, S0). A node-link diagram is employed to illustrate the partitioned data scopes and their relations derived from the corresponding state in each state card. Each node represents a data scope, and edges indicate relations within the node-link diagram. The state card positioned at the center of the insight graph panel is larger than the others and represents the focused state.

CoInsight provides three modes for each node: *collapsed*, *expanded*, and *selected*. In the collapsed mode, only the insight type of the node is displayed, while in the expanded mode, the details of the insight are presented through an exploration-supported visualization *(DC1)*. Due to space constraints, CoInsight does not provide detailed annotations of each visual mark and enables users to check the underlying data attributes through hovering interaction. A collapsed node can be expanded by clicking on it. In particular, we define the collapsed mode as an insight node and the expanded mode as an insight card. Users can incorporate insights of interest into a data story, and we refer to the nodes related to these insights within the data story as selected mode.

Inspired by the hierarchical edge bundles techniques [68], we visualize the insight graph using a nested graph with edge

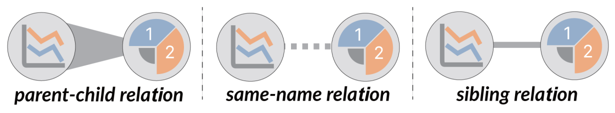


Fig. 7. The visual encodings of three types of relations in the insight graph of the CoInsight system. A line with varying thickness indicates the parent-child relation. A dashed line indicates the same-name relation. A straight line indicates the sibling relation.

bundling. To present the extracted insights and relations clearly, we have designed intuitive visual representations for nodes and links in insight graph.

Nodes: The color of nodes encodes their insight types (*i.e.*, point, shape, and compound). The icons on nodes indicate their subtypes (outlier, dominance, *etc.*), as explained in Section III-C-2.

Edges: We encode different types of relations into the styles of the edges connected to expanded nodes, as shown in Fig. 7. For intra-state relations, an edge with varying line thickness represents a parent-child relation, with the thicker end of the line connecting to the parent node and the thinner end connecting to the child node. Among other uniformly thick edges, straight lines represent sibling relations, and dashed lines represent same-name relations. For inter-state relations, We apply the edge bundling algorithm [69] to avoid occlusions with nodes in the focused state. Moreover, we only display connections between selected nodes to avoid cluttering the graph with many edges. Each cluster of bundled edges connects the selected node in the focused state to nodes in other states related to the selected one.

2) Insight Distribution Panel: The insight distribution panel is designed to provide users with a statistical overview and assist users in flexibly locating data insights of interest. As shown in Fig. 6(a), this panel displays the distribution of extracted insights and relations in the focused state. Additionally, it presents the score distributions of different insight types, obtained through the insight calculation process explained in Section III-C-2. Users can filter the insights and relations within the panel, enabling targeted exploration in the direction of their interest.

Insight Distribution: The insight distribution (Fig. 6(a2)) shows the distribution of the amount of insights (the vertical bar chart on the left) and the computation scores of each insight type (the horizontal histograms on the right). Users can opt to display only the types of insights they are interested in on the insight graph (Fig. 6(c)). Additionally, they can brush the histograms to show only the insights within specific score ranges. These filtering functions allow users to conduct targeted exploration based on their requirements *(DC3)*.

Insight Relation Distribution: The insight relation distribution (Fig. 6(a1)) shows the distribution of the number of insight relations by type. The bar chart displays the distribution of different types of relations, with each bar representing the amount of one type. Users can flexibly filter different types of connections. When edges from the insight graph are filtered, the resulting independent nodes will be hidden. Edge filtering also reduces the complexity of the insight graph, facilitating targeted exploration by users *(DC3)*.

3) Insight Selection Panel: As mentioned in Section III-C-2, we calculate all types of insights for a data scope and record their scores. If the score exceeds a threshold, we consider that the data scope possesses this insight. Therefore, a data scope may contain multiple insights simultaneously, but the insight graph can only display one selected insight. Users can click on a node to check all insights within the data scope in the insight selection panel, as shown in Fig. 6(d). Furthermore, they can update the selected insight of the corresponding node in the insight graph. This allows users to explore the data insights from various perspectives comprehensively.

4) Hierarchical Table Panel: The hierarchical table panel displays the table structure of the focused state, as shown in Fig. 6(b). Due to limited screen space, we enable users to check the labels of each table header via a hovering interaction. We encode the density of each table cell as the amount of its related insights to provide users with an overview of the insight distribution among table content. Based on this panel, users are also allowed to filter the data insights shown in the insight graph by selecting a specific range in the table content. Then, the insight graph panel will only keep the insights related to the selected data items after filtering *(DC3)*. This can help users locate the data insights from the data perspective (*DC2*).

When users hover over a node in the insight graph panel, the cells within the data scope are highlighted in the hierarchical table panel. The data scope corresponding to an expanded node in the insight graph is outlined with a dashed border, while the one corresponding to a selected node is outlined with a solid border. This helps users establish an association between a node in the insight graph and its position in the hierarchical table *(DC2)*.

C. Insight Story

The CoInsight system is designed to generate a data story for the user-uploaded table, allowing users to showcase their understanding of the table. Consequently, the system can organize users' insight exploration results from various states into a complete data story *(DC4)*.

Change Focused State: The number of states for hierarchical tables is exponentially related to the number of levels. We define a focused state in the insight graph to improve the exploration efficiency. The visualization results of each panel in CoInsight are based on the focused state, and the insight graph panel only shows the states related to the focused one. After completing exploration in the focused state, users can switch their focused state to another one. Then, the system will update the states shown in the insight graph panel according to the relations of the new focused state. The visualizations displayed in all other panels are also changed accordingly. It is worth noting that all insights selected by users in previously explored states will be preserved to construct a comprehensive data story *(DC4)*.

Generate Insight Story: During the exploration process, Coinsight records the insights from various states *selected* by users in the insight graph panel (see Section IV-B-1), as well as the relations between them. Then, the system organizes these insights into a data story with a tree structure. For example, Fig. 8(g) shows an example of a data story generated by users

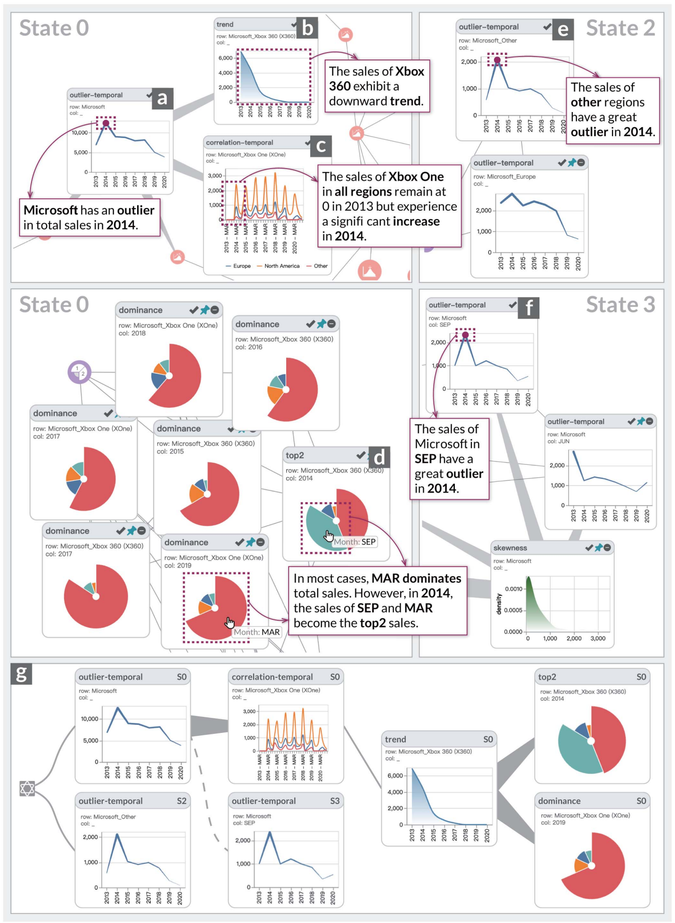

State 0
The sales of Xbox 360 exhibit a downward trend.
Microsoft has an outlier in total sales in 2014.
The sales of Xbox One in all regions remain at 0 in 2013 but experience a signifi cant increase in 2014.
State 2
The sales of other regions have a great outlier in 2014.
State 0
State 3
The sales of Microsoft in SEP have a great outlier in 2014.
In most cases, MAR dominates total sales. However, in 2014, the sales of SEP and MAR become the top2 sales.
a
b
c
d
e
f
g

Fig. 8. A usage scenario demonstrating the exploration of data stories using CoInsight. The data exploration process is illustrated from (a) to (e), with annotations providing detailed explanations of insights. (g) presents the final data story generated using CoInsight, where the star icon signifies the starting point of the data story and the style of edges between insights indicates their relations.

using CoInsight. The nodes indicate the data insights, and the edges between nodes encode their relations, where the style of intra-state relations is consistent with Fig. 7, and inter-state relations are encoded into thinner dashed lines.

V. Usage Scenario

We demonstrate the utility of the CoInsight system through a usage scenario with a data analyst. This scenario focuses on analyzing a game hardware sales dataset, documenting the sales of various game consoles from multiple manufacturers in different regions for each quarter between 2013 and 2020. The data is presented in a hierarchical table. The row headers consist of three levels, representing manufacturers, game consoles, and regions. The column headers have two levels, representing quarters and years.

After uploading the data into the CoInsight prototype system, the analyst can obtain an overview of the amount of extracted insights and their relations in the insight distribution panel (Fig. 6(a)). Then, she can simplify the insight graph by performing filtering operations on both insights and relations. In this scenario, she decides to keep only the parent-child relations and filter out all point insights in the insight graph panel.

The analyst is particularly interested in the sales of Microsoft's gaming consoles. With the assistance of the hierarchical table panel (Fig. 6(b)), she identifies nodes in the insight graph panel (Fig. 6(c)) that correspond to the insights of Microsoft's sales. The analyst observes that Microsoft had an outlier in total sales in 2014, as shown in the insight card Fig. 8(a). The analyst wants to uncover the reasons behind the sales outlier. More specifically, she wants to discover which console, region and month contributed significantly to this outlier.

Based on the parent-child relations displayed in the insight graph, the analyst clicks on the two child nodes of the node described above. One node represents the sales for Xbox 360, while the other represents the sales for Xbox One. The sales of these two consoles make up Microsoft's total sales. By analyzing the insights of these two child nodes, the analyst notices that the sales of Xbox 360 exhibit a downward trend, and there is no abrupt rise in 2014 as shown in Fig. 8(b). Therefore, she believes that Xbox 360 is not the primary reason for the outlier in Microsoft's total sales in 2014. Regarding Xbox One, the analyst discovered that its sales remained at zero in 2013 but experienced a significant increase in 2014, as shown in Fig. 8(c). The analyst speculates that the release of Xbox One in 2014 attracted a large number of users to purchase it. Additionally, despite the declining annual sales of Xbox 360, many users still bought it in 2014. As a result, the analyst suggests that the phenomenon described above is the reason behind the outlier in 2014 from the perspective of different consoles.

Next, the analyst wants to conduct a more detailed analysis from the perspectives of regions and months. She decided to keep point insights that were filtered in the previous step and check the insights for sales of Xbox One and Xbox 360 in 2014. To compare the sales of these two consoles in 2014 with other years, she kept the sibling relations that were also filtered previously and used them as a basis to check the related insights. Most of these nodes have two insights, one calculated from a regional perspective and the other from a monthly perspective, which the analyst can explore independently. Since both of these two analytical perspectives are based on point insights, we only present the visualization results of the monthly perspective in Fig. 8.

Starting with the perspective of months, the analyst finds that in most cases, the sales in March are significantly higher than in other months. However, the sales proportion of Xbox 360 in September 2014 was almost equal to that of March, as shown in Fig. 8(d). Therefore, the analyst believes that the sales of Xbox 360 in September could also be a contributing factor to the outlier.

Then, the analyst switches the above insights to results calculated from a regional perspective to conduct further analysis. Similarly, the analyst finds that most of the sales distribution of Xbox One and Xbox 360 in different regions remains consistent each year, with North America being greater than Europe, followed by other regions. However, the distribution of Xbox 360 differs in 2014, with a higher proportion in other regions. The analyst speculates that this could also be one of the reasons for the outlier in Microsoft's total sales in 2014.

The previous analysis of regions and months revealed anomalies in data proportions. However, the analyst wants to further validate the presence of anomalies by analyzing the absolute data values. The analyst discovers that in the current state, the sales data for Xbox One and Xbox 360 in the same region and same month are not adjacent. Therefore, she decided to switch the focused state from $State_0$ to $State_2$, where the sales in the same region are adjacent. In $State_2$, the analyst finds the node corresponding to Microsoft's sales for all consoles in other regions and discovers that this node has an insight indicating that the sales data for 2014 in other regions is an outlier, as shown in Fig. 8(e). Similarly, the analyst then switches the focused state to $State_3$, where the sales in the same month are adjacent. As shown in Fig. 8(f), she also finds a node's insight in $State_3$, indicating that the sales data for September 2014 is an outlier.

In summary, the analyst has constructed a data story, as shown in Fig. 8(g). The story mainly indicates that Microsoft's total sales data in 2014 is an outlier, which is closely related to the release of Xbox One in 2014. Additionally, the increase in other regions and the spike in September both contributed to the sales outlier in 2014.

VI. User Experiment

We conducted a comparative experiment to evaluate the effectiveness of CoInsight. In the experiment, we chose baselines from commonly used insight extraction techniques, including PowerBI QuickInsight [10], AWS QuickSight [26], and DataPrep.EDA [70]. Through preliminary testing, we found that Dataprep.EDA provides relatively few insights, which cannot form a meaningful data story. Therefore, we ultimately selected PowerBI QuickInsight and AWS QuickSight as baselines.

A. Experiment Setup

Participants and Apparatus: We recruited 12 participants to participate in the experiment. All of them were undergraduate or postgraduate students with different majors, ranging from computer science to mathematics and statistics. Most of them were familiar with data analysis and data visualization. The sessions were carried out in a quiet laboratory, utilizing a Dell Precision T5500 desktop PC with an Intel Xeon Quad-Core processor, 8 GB RAM, and an NVIDIA Quadro 2000 graphics card, connected to a 23-inch HD monitor.

Dataset: The user experiment is based on the dataset described in Section V to avoid confounding variables resulting from varying numbers of insights in different datasets. This dataset documents the quarterly sales of different game consoles from multiple manufacturers in various regions between 2013 and 2020. The data is stored as a hierarchical table, while the two baseline techniques are designed for flat tables. Therefore, we transformed the hierarchical table into a flat one using the baseline techniques. For CoInsight, we used the original hierarchical table as input. In addition, the systems are tested in a counterbalanced order in the experiment to mitigate the learning effect in the experiment results, with details provided in Section VI-B.

B. Experiment Procedure

Our experiment utilizes a within-subjects design, with each user completing the tasks with each of the three systems. In particular, users underwent each of the three systems with balanced Latin square ordering, resulting in six different orders. Given that 12 participants were involved, it means that two participants will be tested on each order. Each user session lasted around one and a half hours.

Training: At the beginning of the experiment, we introduced the background of the experiment dataset. Then, we explained the concept of a data story and provided some examples for clarification. After that, we provided tutorials for the three systems, explaining their functional modules and usage examples. Participants were then free to explore these systems, asking any questions if they needed further explanation. Once all participants had gained a basic understanding of how to use these techniques, we proceeded to introduce the task for our experiment.

Task: Participants were tasked with using the three different systems to discover as many compelling data stories as possible from the experiment dataset within a predefined time frame (twenty minutes in our experiment). Their exploration process was recorded through screen recording and captured screenshots of interesting data stories they discovered, accompanied by textual descriptions. It should be noted that we informed participants in advance that our focus was on the quantity of data stories they created and the quality of each data story (*i.e.*, interestingness and logicality).

Interview: We encouraged participants to think aloud and ask questions during the user experiment. After concluding the experiment, we utilized a five-point Likert scale to gather the participants' evaluations of CoInsight from various perspectives. Then, we conducted an interview with the participants for about 30 minutes to obtain their feedback on CoInsight. Throughout the interview, the participants were prompted to share their thoughts on any aspect of CoInsight.

C. Experiment Result

We evaluate the data stories created by participants from quantitative and qualitative perspectives. The quantitative metrics include the number of stories, the number of insights per story, and the number of insights across all stories. The qualitative metrics involve the quality of stories, encompassing interestingness and logicality. In addition, we have compiled feedback from the participants to demonstrate the utility.

Quantitative Evaluation: We obtain the statistical values based on the data stories generated by participants. The experiment results are shown in Fig. 9(a). These results show evidence suggesting that the CoInsight system offers benefits over the baseline systems from these three aspects. In addition, we also found that the standard deviation is significant in the results. After exploring the original data, we found that some participants prefer to construct long data stories with several data insights, and some prefer to build multiple data stories with relatively few insights, which results from the participants' various understandings of visual data stories.

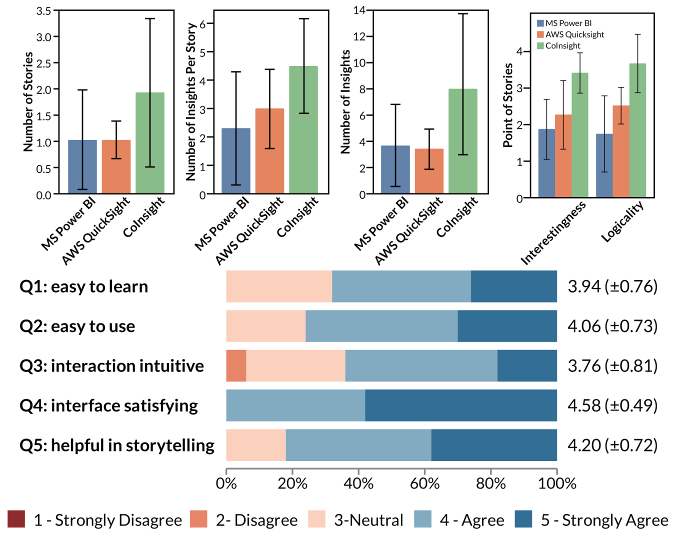


Fig. 9. The result of the user experiment. (a) shows the quantity of data stories created using CoInsight, MS Power BI and AWS QuickSight. The height of bars encodes the mean of the data, while the error bars represent the standard deviation. (b) shows the quality of the generated data stories. The height of the bar encodes the average values of the quality ratings of data stories, with a scoring range of 0 to 5. The error bars represent the standard deviation. (c) illustrates the evaluations from various aspects of CoInsight using a five-point Likert scale, with indicating the average and standard deviations.

Qualitative Evaluation: We found ten domain experts in visual storytelling to evaluate the quality of data stories constructed by the participants. Each expert has over six years of research experience in visual data storytelling. They have seen many high-quality data stories and also have experience in generating data stories. Note that we do not inform the experts which data stories are generated using our system. We asked them to give a score using a five-point Likert scale from the interestingness and logicality perspectives. The results in Fig. 9(b) demonstrate that the quality of data stories generated using the CoInsight system is better.

Participants Feedback: CoInsight received positive feedback from the participants. As shown in Fig. 9(c), CoInsight received high ratings from participants about interface satisfaction. Many participants agreed on the helpfulness of our system in storytelling. During the interview, several participants commented on the rich and flexible interactions: "*I am impressed that [CoInsight] can help me perform various operations to obtain the data insights of interest.*"(P4) and usability: "*The visual encoding of links in the [CoInsight] system is intuitive, and these links between insights can guide me to construct the data story effectively.*"(P6). Another participant (P5) mentioned that the constructed data stories in tree structures can realize a more flexible and efficient organization of data insights than linear data stories.

VII. Discussion and Future Directions

Supporting more insight types: The CoInsight system currently supports the extraction of three types of insights, which can be further divided into eight subtypes, as shown in Fig. 4. We derive these insights from existing studies related to insight computation of multi-dimensional data [10], [30]. We believe that research studies will continuously develop more insight types and corresponding computation algorithms. CoInsight is flexible to support more insight types because it allows users to define more insight extraction algorithms.

Scalability: CoInsight can support hierarchical tables with arbitrary levels theoretically. However, with the increase in header levels and the expansion of the data scale, the enumeration space of insight extraction correspondingly grows larger. To address this, we employ a parallel computing approach to improve CoInsight's efficiency. In addition, with the increase in data scale and the subsequent rise in the number of insights, the node-link diagram currently in use may become less efficient for visualizing the constructed insight graph. We plan to use more efficient visual representations, such as matrix visualization [71] or NodeTrix [72], to improve the scalability further. After analyzing the hierarchical table dataset provided by HiTab [25], we find that among 10,755 hierarchical tables, the average level of row headers is 1.17 (with a maximum of 4), whereas the average level of column headers is 2.99 (with a maximum of 6). Therefore, we believe that CoInsight can handle common hierarchical tables encountered in real-world scenarios.

Automatic storytelling: Recent studies [13], [56] have proposed various algorithms to construct data stories automatically. In the user experiment, we did not compare CoInsight with these techniques, because these techniques generate data stories with only a few manual adjustments, and users can quickly obtain a large amount of data stories. However, creating compelling data stories requires creativity and logical consideration from data analysis [47], [73]. For example, users cannot determine their exploration direction using these automatic data story generation techniques. One possible future direction is to develop a mixed-initiative approach for insight generation based on CoInsight. For example, CoInsight can incorporate recommendation algorithms so that users may start with recommended data stories. The recommendation algorithms can also build connections among several discrete insights selected by users to construct data stories. In addition, CoInsight is designed to assist data analysts in constructing data stories for hierarchical tables. To further assist ordinary users in understanding the extracted insights and generated data stories, we will improve our insight extraction results through accompanied visualization with text narrations and annotations to clearly express the message.

Logical relations between insights: Existing studies on visual storytelling have identified six logical relations between data insights: similarity, temporal, contrast, cause-effect, elaboration, and generalization. In contrast with the relations based on the structure of hierarchical headers, these relations are defined by the semantics of data insights. In particular, part of these two kinds of relations can also be matched. More specifically, the temporal and contrast relations can be reflected using the same-name and sibling relations, while the elaboration and generalization relations are manifested in the defined parent-child relation. In our work, the data insights with similarity relations are within one insight node in the insight graph. The cause-effect relation is derived based on the inner content using a specific algorithm [74]. CoInsight requires analysts to mine the cause-effect relation manually, as demonstrated in the usage scenario (Section V). In the future, we plan to add more logical relations between insights into the insight graph.

Semantic relations: Hierarchical table headers always contain rich semantic information. Inspired by a recent study [75], we find that automatically measuring semantic similarity between table headers and leveraging them to extract semantic relations between insights presents a valuable research direction in the field of data story generation. The Large language models (LLMs) [76] also bring more possibilities in this direction. In the future, we will consider introducing extensions into CoInsight to calculate semantic relations and design new visual encodings in the insight graph to represent them. Additionally, considering the imperfect reliability of automatically extracted semantic information, we will support users to interactively modify the extracted semantic relations (including additions and deletions), to assist them in creating more comprehensive data stories.

VIII. Conclusion

Previous work provides techniques to extract insights and construct visual data stories for tabular data, but they cannot fully leverage the characteristics of hierarchical tables. In this paper, we have presented the CoInsight system to assist users in constructing visual data stories for hierarchical tables. CoInsight extracts insights from hierarchical tables and utilizes relations between header cells in hierarchical tables to construct insight relations. The extracted insights and insight relations form an insight graph. The CoInsight system enables users to construct a data story by exploring paths in the insight graph. A usage scenario and a user experiment reveal the utility and usability of CoInsight in constructing visual data stories from hierarchical tables.

References

[1] W. Dou, S. Han, L. Xu, D. Zhang, and J. Wei, “Expandable group identification in spreadsheets,” in *Proc. ACM/IEEE Int. Conf. Automated Softw. Eng.*, 2018, pp. 498–508.

[2] Z. Chen and M. Cafarella, “Automatic web spreadsheet data extraction,” in *Proc. Int. Workshop Semantic Search over Web*, 2013, pp. 1:1–1:8.

[3] T. Munzner, *Visualization Analysis and Design*. Boca Raton, FL, USA: CRC, 2014.

[4] M. O. Ward, G. Grinstein, and D. Keim, *Interactive Data Visualization: Foundations, Techniques, and Applications*. Boca Raton, FL, USA: CRC Press, 2010.

[5] C. Chai, J. Liu, N. Tang, G. Li, and Y. Luo, “Selective data acquisition in the wild for model charging,” *Proc. VLDB Endowment*, vol. 15, no. 7, pp. 1466–1478, 2022.

[6] C. Chai, J. Wang, Y. Luo, Z. Niu, and G. Li, “Data management for machine learning: A survey,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4646–4667, May 2023.

[7] X. Qin et al., “Interactively discovering and ranking desired tuples by data exploration,” *VLDB J.*, vol. 31, no. 4, pp. 753–777, 2022.

[8] X. Qin et al., “Synthesizing privacy preserving entity resolution datasets,” in *Proc. Int. Conf. Data Eng.*, 2022, pp. 2359–2371.

[9] Y. Luo, C. Chai, X. Qin, N. Tang, and G. Li, “VisClean: Interactive cleaning for progressive visualization,” *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 2821–2824, 2020.

[10] R. Ding, S. Han, Y. Xu, H. Zhang, and D. Zhang, “QuickInsights: Quick and automatic discovery of insights from multi-dimensional data,” in *Proc. Int. ACM Conf. Manage. Data*, 2019, pp. 317–332.

[11] M. Sun, L. Cai, W. Cui, Y. Wu, Y. Shi, and N. Cao, “Erato: Cooperative data story editing via fact interpolation,” *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 1, pp. 983–993, Jan. 2023.

[12] C. D. Stolper, B. Lee, N. Henry Riche, and J. Stasko, “Emerging and recurring data-driven storytelling techniques: Analysis of a curated collection of recent stories,” Microsoft Research, Tech. Rep. MSR-TR-2016–14, 2016. [Online]. Available: https://www.microsoft.com/en-us/research/publication/emerging-and-recurring-data-driven-storytelling-techniques-analysis-of-a-curated-collection-of-recent-stories/

[13] D. Shi, X. Xu, F. Sun, Y. Shi, and N. Cao, “Calliope: Automatic visual data story generation from a spreadsheet,” *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 453–463, Feb. 2021.

[14] Y. Luo, X. Qin, N. Tang, and G. Li, “DeepEye: Towards automatic data visualization,” in *Proc. Int. Conf. Data Eng.*, 2018, pp. 101–112.

[15] Y. Luo, X. Qin, C. Chai, N. Tang, G. Li, and W. Li, “Steerable self-driving data visualization,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 475–490, Jan. 2022.

[16] Y. Luo, C. Chai, X. Qin, N. Tang, and G. Li, “Interactive cleaning for progressive visualization through composite questions,” in *Proc. Int. Conf. Data Eng.*, 2020, pp. 733–744.

[17] Y. Luo et al., “Deeptrack: Monitoring and exploring spatio-temporal data - A case of tracking COVID-19,” *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 2841–2844, 2020.

[18] J. Tang, Y. Luo, M. Ouzzani, G. Li, and H. Chen, “Sevi: Speech-to-visualization through neural machine translation,” in *Proc. Int. ACM Conf. Manage. Data*, 2022, pp. 2353–2356.

[19] X. Qin, Y. Luo, N. Tang, and G. Li, “Making data visualization more efficient and effective: A survey,” *VLDB J.*, vol. 29, no. 1, pp. 93–117, 2020.

[20] G. Li et al., “Barcodetree: Scalable comparison of multiple hierarchies,” *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 1022–1032, Jan. 2020.

[21] G. Li, M. Tian, Q. Xu, M. J. McGuffin, and X. Yuan, “GoTree: A grammar of tree visualizations,” in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2020, pp. 1–13.

[22] G. Li, M. Tian, Q. Xu, M. J. McGuffin, and X. Yuan, “Tree illustrator: Interactive construction of tree visualizations,” in *Proc. Extended Abstracts ACM Conf. Hum. Factors Comput. Syst.*, 2020, pp. 1–4.

[23] Z. Chen and M. Cafarella, “Integrating spreadsheet data via accurate and low-effort extraction,” in *Proc. ACM Conf. Knowl. Discov. Data Mining*, 2014, pp. 1126–1135.

[24] Z. Wang et al., “TUTA: Tree-based transformers for generally structured table pre-training,” in *Proc. ACM Conf. Knowl. Discov. Data Mining*, 2021, pp. 1780–1790.

[25] Z. Cheng et al., “HiTab: A hierarchical table dataset for question answering and natural language generation,” in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 1094–1110.

[26] A. W. Services, “Amazon quicksight,” 2012, [Online]. Available: https://aws.amazon.com/quicksight/

[27] M. Vartak, S. Rahman, S. Madden, A. G. Parameswaran, and N. Polyzotis, “SEEDB: Efficient data-driven visualization recommendations to support visual analytics,” in *Proc. Int. Conf. Very Large Data Bases*, 2015, pp. 2182–2193.

[28] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. G. Parameswaran, “Effortless data exploration with zenvisage: An expressive and interactive visual analytics system,” *Proc. VLDB Endowment*, vol. 10, no. 4, pp. 457–468, 2016.

[29] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati, “Foresight: Recommending visual insights,” *Proc. Int. Conf. Very Large Data Bases*, vol. 10, no. 12, pp. 1937–1940, 2017.

[30] B. Tang, S. Han, M. L. Yiu, R. Ding, and D. Zhang, “Extracting top-k insights from multi-dimensional data,” in *Proc. Int. ACM Conf. Manage. Data*, 2017, pp. 1509–1524.

[31] Q. Lin et al., “BigIN4: Instant, interactive insight identification for multi-dimensional Big Data,” in *Proc. ACM Conf. Knowl. Discov. Data Mining*, 2018, pp. 547–555.

[32] P. Ma, R. Ding, S. Han, and D. Zhang, “Metainsight: Automatic discovery of structured knowledge for exploratory data analysis,” in *Proc. Int. ACM Conf. Manage. Data*, 2021, pp. 1262–1274.

[33] Z. Zeng et al., “An evaluation-focused framework for visualization recommendation algorithms,” *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 346–356, Jan. 2022.

[34] J. D. Mackinlay, “Automating the design of graphical presentations of relational information,” *ACM Trans. Graph.*, vol. 5, no. 2, pp. 110–141, 1986.

[35] J. D. Mackinlay, P. Hanrahan, and C. Stolte, "Show me: Automatic presentation for visual analysis," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1137–1144, Nov./Dec. 2007.

[36] K. Wongsuphasawat, D. Moritz, A. Anand, J. D. Mackinlay, B. Howe, and J. Heer, "Voyager: Exploratory analysis via faceted browsing of visualization recommendations," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 649–658, Jan. 2016.

[37] K. Wongsuphasawat et al., "Voyager 2: Augmenting visual analysis with partial view specifications," in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2017, pp. 2648–2659.

[38] T.-Y. Liu, "Learning to rank for information retrieval," *Found. Trends Inf. Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[39] Y. Luo, X. Qin, N. Tang, G. Li, and X. Wang, "DeepEye: Creating good data visualizations by keyword search," in *Proc. Int. ACM Conf. Manage. Data*, 2018, pp. 1733–1736.

[40] X. Qin, Y. Luo, N. Tang, and G. Li, "DeepEye: Visualizing your data by keyword search," in *Proc. Int. Conf. Extending Database Technol.*, 2018, pp. 441–444.

[41] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin, "Natural language to visualization by neural machine translation," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 217–226, Jan. 2022.

[42] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin, "Synthesizing natural language to visualization (NL2VIS) benchmarks from NL2SQL benchmarks," in *Proc. Int. ACM Conf. Manage. Data*, 2021, pp. 1235–1247.

[43] D. Moritz et al., "Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 438–448, Jan. 2019.

[44] K. Z. Hu, M. A. Bakker, S. Li, T. Kraska, and C. A. Hidalgo, "VizML: A machine learning approach to visualization recommendation," in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–12.

[45] M. Zhou et al., "Table2charts: Recommending charts by learning shared table representations," in *Proc. ACM Conf. Knowl. Discov. Data Mining*, 2021, pp. 2389–2399.

[46] B. Lee, N. H. Riche, P. Isenberg, and S. Carpendale, "More than telling a story: Transforming data into visually shared stories," *IEEE Comput. Graph. Appl.*, vol. 35, no. 5, pp. 84–90, Sep./Oct. 2015.

[47] E. Segel and J. Heer, "Narrative visualization: Telling stories with data," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1139–1148, Nov./Dec. 2010.

[48] C. Tong et al., "Storytelling and visualization: An extended survey," *Information*, vol. 9, no. 3, 2018, Art. no. 65.

[49] Z. Zhao and N. Elmqvist, "The stories we tell about data: Surveying data-driven storytelling using visualization," *IEEE Comput. Graph. Appl.*, vol. 43, no. 4, pp. 97–110, Jul.Aug. 2023.

[50] A. Satyanarayan and J. Heer, "Authoring narrative visualizations with ellipsis," *Comput. Graph. Forum*, vol. 33, no. 3, pp. 361–370, 2014.

[51] Y. Wang et al., "InfoNice: Easy creation of information graphics," in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2018, pp. 1–12.

[52] M. Brehmer, B. Lee, B. Bach, N. H. Riche, and T. Munzner, "Timelines revisited: A design space and considerations for expressive storytelling," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 9, pp. 2151–2164, Sep. 2017.

[53] M. Brehmer et al., "Timeline storyteller: The design & deployment of an interactive authoring tool for expressive timeline narratives," in *Proc. Comput. Journalism Symp.*, 2019.

[54] Y. Kim, K. Wongsuphasawat, J. Hullman, and J. Heer, "GraphScape: A model for automated reasoning about visualization similarity and sequencing," in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2017, pp. 2628–2638.

[55] S. Xu, C. Bryan, J. K. Li, J. Zhao, and K. Ma, "Chart constellations: Effective chart summarization for collaborative and multi-user analyses," *Comput. Graph. Forum*, vol. 37, no. 3, pp. 75–86, 2018.

[56] Y. Wang et al., "Automatic generation of fact sheets from tabular data," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 895–905, Jan. 2020.

[57] J. Zhao et al., "Chartstory: Automated partitioning, layout, and captioning of charts into comic-style narratives," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 2, pp. 1384–1399, Feb. 2023.

[58] X. Wang and D. Wood, "Xtable: A tabular editor and formatter," Tech. Rep., 1996.

[59] G. Li, R. Li, Z. Wang, C. H. Liu, M. Lu, and G. Wang, "HiTailor: Interactive transformation and visualization for hierarchical tabular data," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 1, pp. 139–148, Jan. 2023.

[60] M. Research, "Quickinsight," 2016, [Online]. Available: https://www.microsoft.com/research/project/quickinsights/

[61] B. Bach, Z. Wang, M. Farinella, D. Murray-Rust, and N. H. Riche, "Design patterns for data comics," in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2018, pp. 38.

[62] G. Li and X. Yuan, "GoTreeScape: Navigate and explore the tree visualization design space," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 12, pp. 5451–5467, Dec. 2023.

[63] M. Tian, G. Li, and X. Yuan, "LitVis: A visual analytics approach for managing and exploring literature," *J. Visual.*, vol. 26, no. 6, pp. 1445–1458, 2023.

[64] Y. Han, Z. Wang, S. Chen, G. Li, X. Zhang, and X. Yuan, "Interactive assigning of conference sessions with visualization and topic modeling," in *Proc. IEEE Pacific Visual. Symp.*, 2020, pp. 236–240.

[65] M. Lu et al., "Interaction: Interaction enhancement for web-based visualizations," in *Proc. IEEE Pacific Visual. Symp.*, 2017, pp. 61–70.

[66] N. W. Kim et al., "DataToon: Drawing dynamic network comics with pen touch interaction," in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–12.

[67] E. Mörth, S. Bruckner, and N. N. Smit, "ScrollyVis: Interactive visual authoring of guided dynamic narratives for scientific scrollytelling," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 12, pp. 5165–5177, Dec. 2023.

[68] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 5, pp. 741–748, Sep./Oct. 2006.

[69] D. Holten and J. J. Van Wijk, "Force-directed edge bundling for graph visualization," *Comput. Graph. Forum*, vol. 28, no. 3, pp. 983–990, 2009.

[70] J. Peng et al., "DataPrep.EDA: Task-centric exploratory data analysis for statistical modeling in python," in *Proc. Int. ACM Conf. Manage. Data*, 2021, pp. 2271–2280.

[71] B. Alper, B. Bach, N. Henry Riche, T. Isenberg, and J.-D. Fekete, "Weighted graph comparison techniques for brain connectivity analysis," in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2013, pp. 483–492.

[72] N. Henry, J.-D. Fekete, and M. J. McGuffin, "NodeTrix: A hybrid visualization of social networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1302–1309, Nov./Dec. 2007.

[73] H. Li, Y. Wang, and H. Qu, "Where are we so far? understanding data storytelling tools from the perspective of human-AI collaboration," 2023, *arXiv:2309.15723*.

[74] U. Schaechtle, K. Stathis, and S. Bromuri, "Multi-dimensional causal discovery," in *Proc. Int. joint Conf. Artif. Intell.*, 2013, pp. 1649–1655.

[75] Z. Wu et al., "Explainable data transformation recommendation for automatic visualization," *Front. Inf. Technol. Electron. Eng.*, vol. 24, no. 7, pp. 1007–1027, 2023.

[76] W. Yang, M. Liu, Z. Wang, and S. Liu, "Foundation models meet visualizations: Challenges and opportunities," *Comput. Vis. Media*, 2023.