ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #36509

TreeVisual: Design and Evaluation of a Web-Based Visualization Tool for Teaching and Learning Tree Visualization

Brendan O'Handley

Software Engineer at Grafana Labs with a masters in computer science and engineering from the University of Notre Dame. I'm interested in data visualization, education analytics, dashboards and JavaScript.

Yuheng Wu

Chaoli Wang (Associate Professor)

Dr. Chaoli Wang is a professor of computer science and engineering at the University of Notre Dame. He holds a Ph.D. degree in computer and information science from The Ohio State University. Dr. Wang's main research interest is data visualization, in particular on the topics of time-varying multivariate data visualization, flow visualization, as well as information-theoretic algorithms, graph-based techniques, and deep learning solutions for big data analytics.

TreeVisual: Design and Evaluation of a Web-Based Visualization Tool for Teaching and Learning Tree Visualization

Brendan J. O'Handley[1], Yuheng Wu[2], Haobin Duan[3], Chaoli Wang[1]
[1]University of Notre Dame, [2]Peking University, [3]Huazhong University of Science and Technology

Abstract

Trees are a common form of visual representation that encode hierarchical data structures in parent-children relationships. Although many software tools can visualize tree structures, there is a lack of tools designed for teaching and learning tree visualization. This paper presents TreeVisual, an educational software tool that visualizes hierarchical data structures with real-world datasets and allows users to explore and manipulate the various forms of tree visualizations via a web interface. We collect four hierarchical datasets from different domains, including academic publication, government structures, global trading statistics, and tree taxonomy. Leveraging D3.js, TreeVisual supports four popular hierarchical layouts and their variants: pack, radial tree and its collapsible version, sunburst, and treemap and its zoomable version. Furthermore, TreeVisual allows users to comprehend, compare, and critique these layouts in side-by-side display panels, supported by a set of interactive functions. These functions include appearance changes, filtering of intermediate nodes, and selection of leaf nodes. Users can utilize these functions to filter the tree hierarchies, pinpoint nodes of interest, drill down to varying levels of detail, and glean data-related insights. To demonstrate the usefulness of TreeVisual and assess its effectiveness, we conducted a formal user study with students enrolled in a college-level Data Visualization course. The study involves an introduction to the tool, a training session, an in-class quiz integrated into TreeVisual, and a survey. The instructor introduced TreeVisual after related tree visualization concepts and algorithms were covered, followed by an exploration of the tool by the students. Students then performed a pre-quiz with five questions to get familiar with the types of questions that would appear in the actual quiz. A week later, students took the actual quiz of 22 questions in class, followed by a survey that asks for their feedback on tree layouts, datasets, user interface, and general impression of TreeVisual.

1 Introduction

Tree data structures are a widely used data type to represent data, specifically the relationships between points in the data. Many systems can be described using trees. For example, a government can be represented by showing branches such as judicial, legislative, etc., and how they are broken down further by members of each branch. The tree of life can also be represented as such, where the root branches into the kingdom level, each kingdom has child nodes that are phyla, each phylum can branch into classes, and so on. Each representation can show insights in the data, but with such varying systems, it would be beneficial to visualize those systems in a human-understandable way. There are many ways to visualize trees, but many challenges are involved. One key challenge is choosing the right type of layout to visualize the dataset. Selecting the proper layout can show insights more quickly and avoid confusion in the user by not overwhelming them with extraneous information. In data visualization courses, students are often taught a variety of tree visualizations, but it becomes a challenge when comparing each for benefits or drawbacks. Therefore, building a visualization tool that can display a variety of tree layouts and can compare layouts with different datasets would be a great help to students and teachers alike.

This paper presents TreeVisual, a web-based educational tool to assist teachers with instruction and assist students with learning using real-world datasets. TreeVisual displays a variety of hierarchical visualization strategies, including radial, treemap, sunburst, and pack layouts. A user can select two layouts represented side by side and compare them. There are hover events on each node for each layout, and some layouts interact with each other through click events. In addition to the interactive layouts, there is a control panel that allows users to select leaf nodes, filter nodes based on levels, and adjust various attributes of the layout. These features enable users to make sense of the data in meaningful ways, compare layouts, and filter to show complicated datasets in more manageable and understandable ways.

Our primary deliverable is a web-based visualization tool where a teacher can display various tree layouts to a class. Students can also use the tool for further exploration outside of class. TreeVisual has been used in a Data Visualization course in a classroom environment. A formal user study involving students with and without computer science (CS) backgrounds measured its effectiveness as an education tool.

2 Related Work

Visualizing tree data is an essential part of data analysis, and there has been a significant amount of research around it. Hierarchical data structures are most often displayed as trees, where the root node branches into children and those children act as parents to more children. An edge connects each node, and the path starts at a node and follows the edges to a descendant node one or more branches further down the tree structure. The central aspect of tree visualization is visualizing the nodes and edges as they branch from the root. The main challenges of visualizing trees are displaying the whole tree or part of the tree, traversing the paths from node to node by edges, filtering for readability, and generating aesthetically pleasing visualizations. Interested readers may explore more about trees in the following surveys [6, 8, 15]. In TreeVisual, we choose six layouts that show different levels of the trees, and each layout behaves differently in node selection and path traversal. The six layouts are pack, radial tree, collapsible radial tree, sunburst, treemap, and zoomable treemap. The standard treemap displays only leaf nodes, while the zoomable treemap [3] allows users to explore each level of the hierarchy, zooming in or out by clicking the parent of the displayed nodes. The radial trees are closer to more traditional hierarchical views. The standard radial tree automatically shows all the nodes from the root level to the leaf level. The collapsible radial tree [14] exposes a node's children via animated transition upon clicking on that node. Besides treemaps and radial trees, TreeVisual includes two additional layouts: pack and sunburst. The pack layout uses nested circles to represent the hierarchy. The size of each node's circle represents the accumulated attribute values of its leaves. The sunburst layout [5] depicts parent-children relationships by concentric rings.

Computer science education has changed much, but one aspect that has not undergone a significant change is teaching information visualization. As one of the top ten unsolved problems, education and training was listed as a great concern in 2005 [4]. The Gordon Research Conference on Visualization in Science and Education main topic was "Grand Challenges in the Use of Visualization in Science and Education" [1]. To the best of our knowledge, few works focus on teaching data visualization with software tools allowing students to compare a variety of layouts.

The treevis.net project [10] curates over three hundred representations by dimensionality (2D, 3D, hybrid), tree representation (explicit, implicit, hybrid), and alignment(axis, radial, free). Some of these visualizations are static, while others allow a user to traverse the tree. The Interactive Tree of Life [9] is a tool that allows users to upload and describe tree datasets as well as explore the phylogenetic tree of life. This visualization displays a single tree and offers a rich control panel for basic and advanced options. TreeVisual differs from this example because it allows users to compare two tree layouts with a control panel with the essential interaction, selection, filtering, and other visualization techniques. Comparing layouts in a straightforward interface makes TreeVisual very useful to students for grasping the core concepts of tree data structures.

3 Design Principles

In creating a handy tool, we developed an interface that follows well thought-out principles to engage users in the learning process. With TreeVisual, we seek to have an interface that is easy to use and to the point. In addition, we aim to build a tool that is helpful to beginning CS students as well as non-CS students. Therefore, while designing, we kept the following principles in mind.

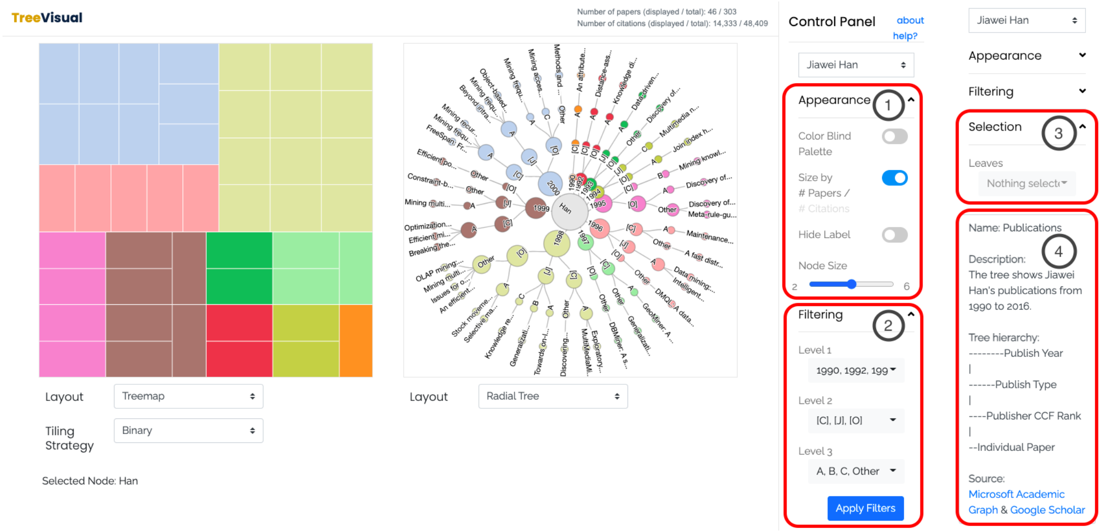
P1: Simplicity. We must design TreeVisual with simplicity. There are hundreds of tree visualizations of varying complexity. Many of these tools are overwhelming to beginning students and present the data so that the core elements of tree data structures are lost in the data. We chose to keep the tool simple, self-explanatory, and with a limited set of ways to interact with the data structures.

P2: Selectivity. With hundreds of tree visualization layouts to choose from, we must select types that display the attributes of trees without overwhelming the learners. It is also vital that we select interactions that are straight to the point and clear in how they represent tree concepts. For example, the radial tree explicitly illustrates a tree, while the treemap shows only the leaves and is, therefore, an implicit representation. We also add features to these visualizations:

making the radial tree collapsible to focus on certain branches and making the treemap zoomable to explore different levels of the data.

P3: Interactivity. Interactivity is crucial to data visualization and should be included in an educational tool. The interaction of the two tree layouts in TreeVisual is vital to allow students to see the patterns in data. There are many ways the layouts interact through node selection and click events. For example, the pack layout allows a user to zoom into a node many levels into the tree. When paired with the treemap, clicking a node further into the tree takes the user through many levels in the zoomable treemap. In addition to the interactivity between layouts, users can filter levels, select leaf nodes, hide and show levels, and change how the size of the nodes is calculated. Each of these interactions updates the layouts to provide instantaneous feedback.

P4: Comparison. Choosing an appropriate layout for a dataset is very important when considering tree visualization. TreeVisual displays two tree layouts generated with the same dataset in two side-by-side display panels. This design allows students to compare the layouts with linked interactions easily. For example, selecting a node in one layout automatically selects the same node in the second. In addition, the two layouts react to data filtering and leaf selection from the control panel in the same way.



Number of papers (displayed / total): 46 / 303
Number of citations (displayed / total): 14,333 / 48,409

Layout: Treemap
Tiling Strategy: Binary
Selected Node: Han

Layout: Radial Tree

Control Panel
Jiawei Han

Appearance
Color Blind Palette
Size by # Papers / # Citations
Hide Label
Node Size 2 6

Filtering
Level 1: 1990, 1992, 199
Level 2: [C], [J], [O]
Level 3: A, B, C, Other

Selection
Leaves: Nothing selecte

Name: Publications

Description:
The tree shows Jiawei Han's publications from 1990 to 2016.

Tree hierarchy:
--------Publish Year
|
------Publish Type
|
----Publisher CCF Rank
|
--Individual Paper

Source:
Microsoft Academic Graph & Google Scholar

(a)

(b)

Figure 1: The user interface of TreeVisual (study component). (a) The entire screenshot with Appearance and Filtering unfolded on the control panel. (b) The remainder of the control panel showing Selection and dataset information.

4 Design and Implementation

TreeVisual was created to help students learn tree structures and their visualizations. The tool allows users to explore real-world datasets from several domains. TreeVisual enables various interactions, including selecting different datasets, filtering datasets by level, and selecting leaves. In addition, we allow users to change the appearance of the layouts by calculating node size by accumulated leaves or the accumulated size attribute, as well as options such as "Color Blind Palette," "Hide Labels," and "Proportional Node Size." TreeVisual has two components, the *study component* and *quiz component*, and the datasets are split equally between these.

The main interface is created using JavaScript, CSS, and HTML. We use the JavaScript libraries D3.js and Intro.js, the former for layout creation and the latter for onboarding instructions. We also use the Bootstrap framework for styling. D3 has built-in support for visualizing hierarchical datasets in different layouts. However, it does not handle the linking between side-by-side visualization layouts. This is our custom TreeVisual tool was created for: handling interactions on one visualization, capturing the event, and mimicking the event in the other visualization.

Figure 1 shows a screenshot of three main parts of TreeVisual: the header on top, the layout panel in the center, and the control panel along the right side. The header displays the title and two different size calculations specific to each dataset (i.e., the total number of leaves and their accumulated size attribute value). In the central panel are the two side-by-side layouts. Below each layout is a select input that allows users to select a different tree layout. The treemap layout has an additional select input for choosing a tiling strategy. Below these select inputs is a text label showing the path to the selected node from the root. We have two links above the control panel to the right that users can choose. First, the "about" link directs users to an about page describing the tool. Below this link is the "help?" link which restarts the onboarding process, giving user instructions. Below these links, users can select a dataset for exploration. Further down, there are three sections, "Appearance (1)," "Filtering (2)," and "Selection (3)." These sections can be expanded or collapsed to allow users access to the interactions within each section. Finally, the is a "Data Description (4)" section at the bottom of the control panel. When hovering over elements, TreeVisual displays tooltips to describe the elements or describe their functions. For example, hovering over nodes may show the node name or the node's whole path from the root (e.g., for treemap). Hovering over the inputs and toggle switches gives the descriptions of their functions.

Name	**Domain**
Christos Faloutsos	Publication
Jiawei Han	Publication
Canada	Government
Philippines	Government
China	Trade
Germany	Trade
Anura	Tree Taxonomy
Carnivora	Tree Taxonomy

Name	**Domain**
Jure Leskovec	Publication
John Stasko	Publication
United Kingdom	Government
United States	Government
United Kingdom	Trade
United States	Trade
Pelecaniformes	Tree Taxonomy
Afrosoricida	Tree Taxonomy

Table 1: The datasets included in TreeVisual. The left and right tables show the datasets included in the study and quiz components, respectively.

4.1 Datasets and Preprocessing

There is a wide range of data with parent-children relationships from various domains in hierarchical structures. To better guide students learning from tree visualization, TreeVisual selects hierarchical datasets from four domains: academic publication, government structures, global trading statistics, and tree taxonomy. These dataset categories represent a broad spectrum of real-world data with a hierarchical structure that students can easily understand. We acquired these datasets through Google Scholar, government official pages, the world bank website, and the Tree-of-Life project. The acquired data are then processed and filtered using scripts to select a representative subset and manually inspected and refined to ensure correctness. Table 1 shows the datasets we include in TreeVisual.

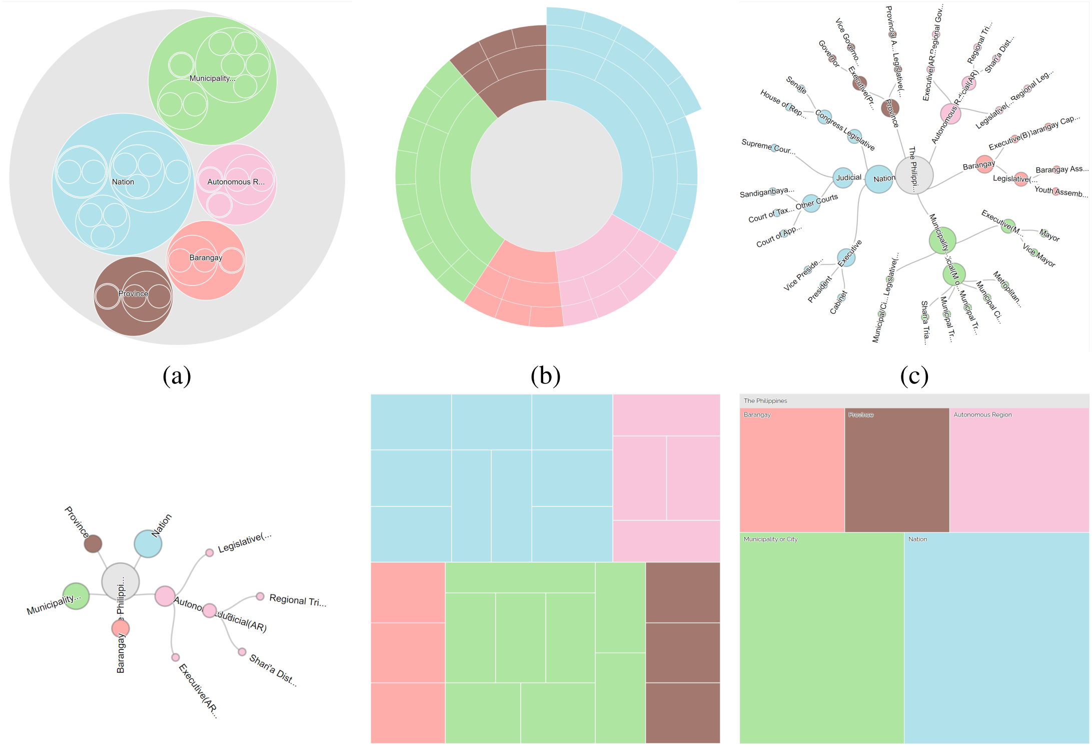
After acquiring raw data from several sources, We need to preprocess them for visualization. Usually, raw data exists in CSV format that contains its path from the root to the leaf in the tree structure. We use a Python script to convert them to a tree object in JSON format. We carefully trim these four categories of datasets to a reasonable size for display and interaction to make clean and neat visualization. If the size of a dataset is larger than a margin, a subset of the original tree structure is selected, which discards deep branches and excess children in one node. After this step, the branching factor and maximum depth of the dataset's tree structure meet the display requirements.

4.2 Tree Layouts

TreeVisual supports four types of tree layouts for users to explore hierarchical structures, parent-children relationships, and node features in the datasets. These include pack, radial tree, sunburst, and treemap layouts. For radial tree and treemap, we provide a collapsible and a zoomable version, respectively.

Pack Layout. The pack layout uses recursively embedded circles to represent the hierarchical relationship of all nodes. Figure 2 (a) and Figure 3 (a) visualize the Philippines (Government) dataset in the pack layout, weighted by the number of leaves and the number of employees, respectively. Every leaf node in the tree corresponds to the most inner circles in the layout, which have the same size when weighted by the number of leaves in one branch. Each interior node in the tree corresponds to a larger circle with one or more child circles inside. The root forms the largest outer circle of grey color. Children of the root are assigned to various colors. Their sizes are determined by the summed weights of all leaves in the current branch. Although the pack layout clearly shows all elements in the tree and the number of nodes in each branch can be easily compared by the circle's size, it does not offer a clear path of edges in the tree because of the nesting of nodes.

Sunburst Layout. The sunburst layout is similar to a pie chart, with the root being the center of the circle (refer to



(a) (b) (c)

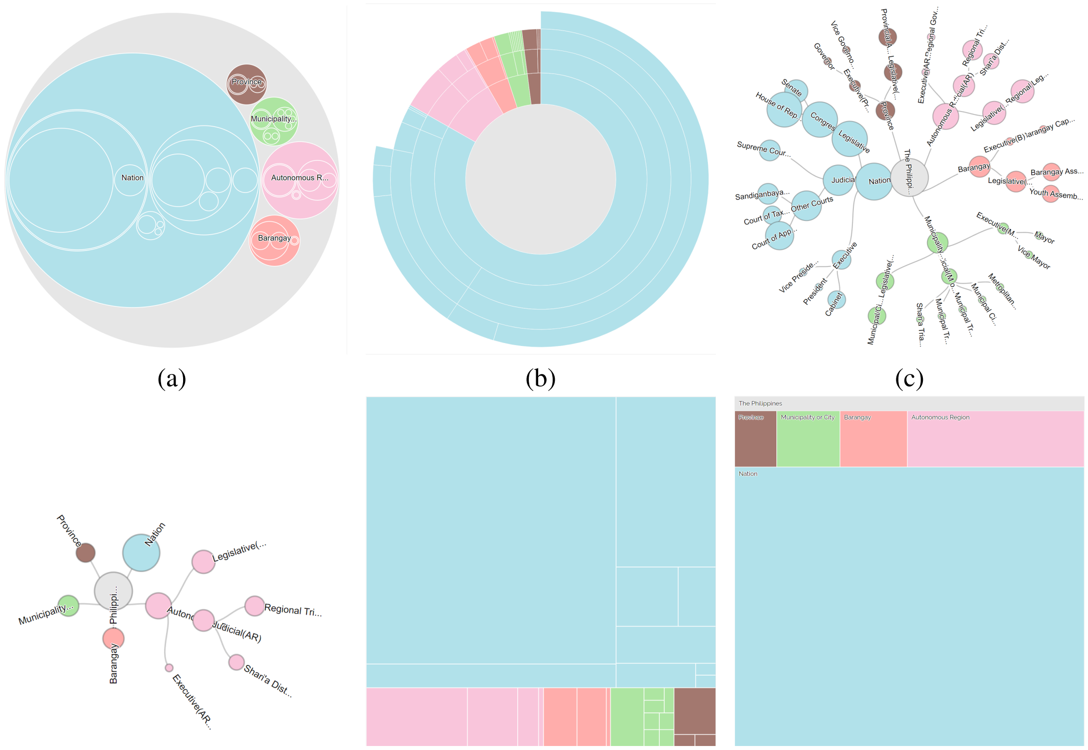
(d) (e) (f)

Figure 2: Six different tree layouts of the Philippines (Government) dataset, weighted by the number of leaves. (a) pack, (b) sunburst, (c) radial tree, (d) collapsible radial tree, (e) treemap, and (f) zoomable treemap.

Figure 2 (b) and Figure 3 (b)). Each branch in the tree takes up a part of the full pie, and the arc length of each slice (and its central angle) is proportional to the total weight of leaves in this branch. In one branch, the circle's inner (outer) part corresponds to the lower (higher) depth of nodes from the root. The circumference corresponds to all the leaves in the tree. When a node is clicked, it will be selected as the new root, and its subtree is then zoomed in, which can be explored more clearly. If the tree has leaves with different depths, the sunburst layout will have some parts of the pie more protruding than other parts of the pie. Thus, the sunburst layout can show all leaves' depth variation and the proportion of total weights of leaves of each tree branch. However, it takes some learning to understand the sunburst behavior and presentation, which is not instantly intuitive compared to other layouts like a radial tree.

Radial Tree Layout. The radial tree layout is similar to the classical one based on nodes and edges (refer to Figure 2 (c) and Figure 3 (c)). Unlike classical tree layouts arranged from top to bottom, the nodes in a radial tree are arranged in a circular shape, which better utilizes the screen space. Each edge connects a child and its parent. The size of each node is encoded by the accumulated attribute values of all its descendant leaves. All elements in the tree are shown clearly with their name and associated value for comparison. Users can easily examine the path from a node to the root. However, the original radial tree layout is static, offering no interactive exploration. Therefore, we also provide a collapsible version. Figure 2 (d) and Figure 3 (d) visualize the Philippines (Government) dataset in the collapsible radial tree layout. Instead of showing all tree elements simultaneously, it only shows the selected subset of the original tree, providing a more focused view on parts of the tree without being overwhelmed by the entire dataset.

Treemap Layout. The treemap layout is composed of multiple rectangles representing all leaves in the tree. Figure 2 (e) and Figure 3 (e) visualize the Philippines (Government) dataset in the treemap layout, with the size of a rectangle proportional to the number of leaves and the number of employees, respectively. Since the overall area of a node is directly related to its data in the leaves, the treemap layout gives good summarization of leaf nodes and their relative size, and comparing different nodes can be done easily. However, the entire tree structure and the parent-children relationship cannot be clearly observed since they are presented in a flattened way. Four tiling strategies are available (*binary*, *dice*, *slice*, *slice & dice*), which determines how space in treemap is split into multiple regions, as shown in



(a) (b) (c)

(d) (e) (f)

Figure 3: Six different tree layouts of the Philippines (Government) dataset, weighted by the number of employees. (a) pack, (b) sunburst, (c) radial tree, (d) collapsible radial tree, (e) treemap, and (f) zoomable treemap.

Figure 4. The interactive, zoomable version of the treemap layout only shows one layer of the parent and children. One can click on a rectangle to move back and forth between parent and children of the current node, thus making the tree explorable branch by branch. However, the entire dataset cannot be seen at once for the zoomable version, and leaves in different nodes cannot be directly compared.

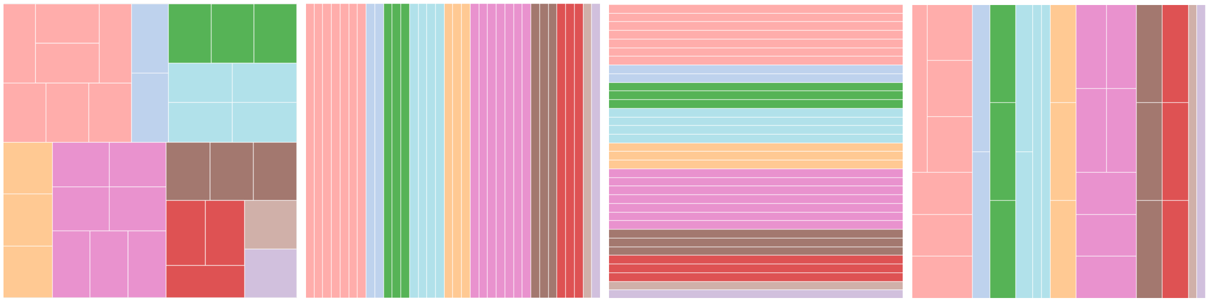
4.3 Interaction Functions

TreeVisual offers users filtering, selection, and a unique set of linking behavior so that they may explore and learn about tree data structures as well as tree visualizations. These functions allow users to view branches and leaves of interest without clutter and explore paths in ways unique to specific layouts.

Filtering. The filtering function allows users to select nodes by level to quickly visualize a large dataset or better understand the relationship between branches. On loading a dataset, TreeVisual collects nodes for each level, excluding the root or leaf levels. A user can choose to include or exclude nodes for each intermediate level. In Figure 5, we show the radial tree layout with the Anura (Frogs & Toads) (Tree Taxonomy) dataset. This dataset has two levels that can be filtered where nodes can be included or excluded. The root and the leaf nodes cannot be filtered. We show a filtered version on the left and an unfiltered one on the right. Some larger datasets are prefiltered on dataset selection, indicated by a tooltip at the top-right corner of the dataset select input.

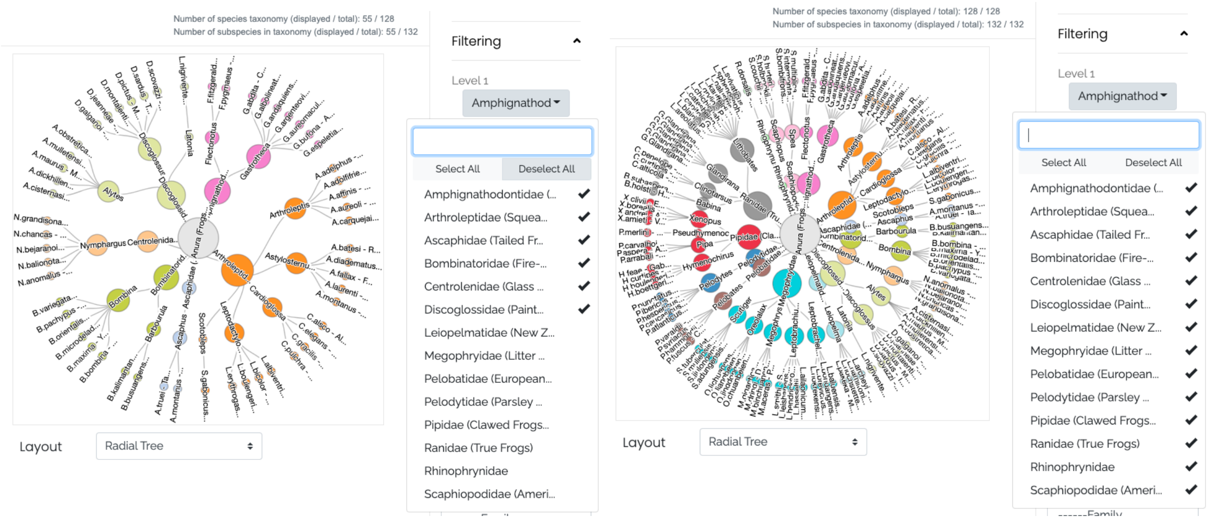
Selection. The selection function allows users to choose a leaf or a set of leaves to highlight. When selecting a leaf, that leaf's outline is set in bold, while the unselected leaves are drawn semi-transparently. It is important to note, when filtering nodes, those leaves that are not displayed are still listed in the select input but are sorted to the bottom and rendered inactive to choose. In Figure 6, we can see an assortment of leaves has been selected.

Linking. TreeVisual offers a variety of functions that link 'click' behavior between tree layouts. First, for all layouts, if a user hovers on a node in one layout, the same node in the second layout is given a bold outline. In addition, the path of the clicked node is updated at the bottom of the layout panel. Next, there is a set of layouts that have behavior specific to them when a node is clicked. These include pack, collapsible radial tree, sunburst, and zoomable treemap.



(a) (b) (c) (d)

Figure 4: Four treemap tiling strategies visualized with the Christos Faloutsos (Publication) dataset. (a) binary, (b) dice, (c) slice, and (d) slice & dice.



Number of species taxonomy (displayed / total): 55 / 128
Number of subspecies in taxonomy (displayed / total): 55 / 132

Layout: Radial Tree

Number of species taxonomy (displayed / total): 128 / 128
Number of subspecies in taxonomy (displayed / total): 132 / 132

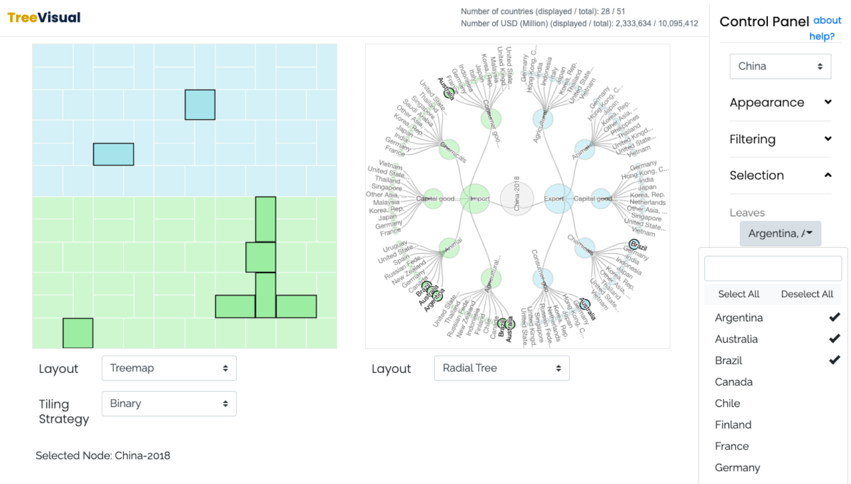
Layout: Radial Tree

(a) (b)

Figure 5: Filtering the Anura (Frog & Toads) (Tree Taxonomy) dataset. (a) Filtered by removing several nodes from the root's children. (b) Unfiltered.

The pack layout allows users to click a node at any tree level to zoom into that level. The collapsible radial tree layout allows a user to click a node to expand or collapse the children of that node. The sunburst has two functions. When displaying the layout on initialization, all nodes are displayed, and a user can click any node to zoom into that level. From that level, only the children of that node, the node itself, and the parent of the node are displayed. Clicking the parent zooms back one level, clicking any one of the children zooms in, and clicking the same node does not fire off the zoom. The zoomable treemap has similar click behavior as the sunburst but does not initially show the entire tree. It does allow a user to zoom into a child or zoom out by clicking a parent node.

TreeVisual implements a unique set of functions in linking the behaviors of these layouts. That is, we consider how far a click event traverses a path and the speed at which elements are displayed or zoomed into. A custom JavaScript library was built in TreeVisual specifically for this linking. As the first linking behavior, the pack and sunburst layouts allow a user to traverse more than one level. When a user clicks a node more than one level into a tree with either of these layouts, this behavior overrides the single-level traversal layouts, namely, collapsible radial tree and zoomable treemap. Clicking a node in the pack or sunburst two or more levels from the root simulates multiple click events on single-level layouts using recursive functions and timing functions based on the speed at which the single-level layouts behave. For example, when the pack layout is paired with the zoomable treemap and a node two levels down the root is selected, the zoomable treemap has two simulated click events to zoom in two levels. The second linking behavior is as follows. When a user clicks a node, the collapsible radial tree collapses or expands that node's children. If the children are visible when the node is clicked, they are collapsed. Otherwise, they are expanded and made visible. This



Number of countries (displayed / total): 28 / 51
Number of USD (Million) (displayed / total): 2,333,634 / 10,095,412

Argentina ✓
Australia ✓
Brazil ✓
Canada
Chile
Finland
France
Germany

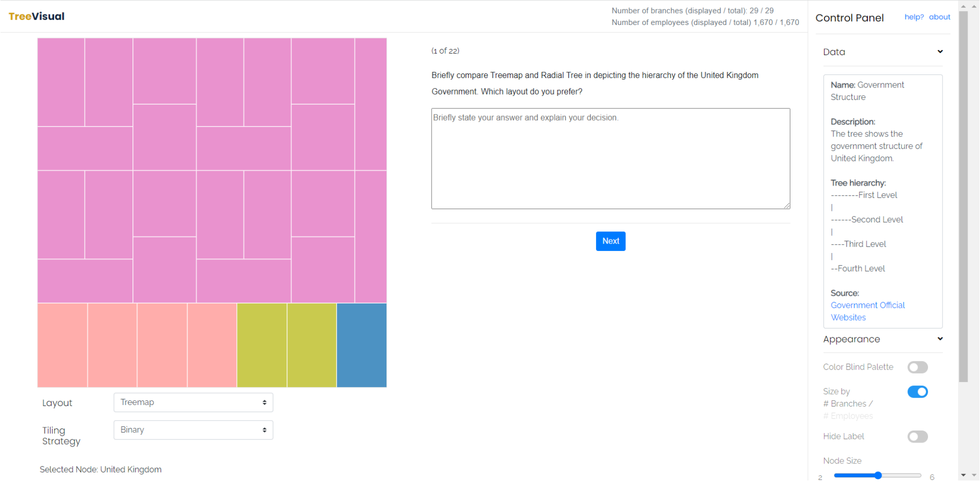
Selected Node: China-2018

Figure 6: Leaf selection of Argentina, Australia, and Brazil for the China (Trade) dataset. Although three countries are selected, eight leaf nodes are highlighted. These countries appear as children of multiple trade goods' nodes. The visualization reveals many goods imported to China from a country or exported from China to a country. Therefore, countries as leaves can repeat under the parent of 'goods' more than once.

happens one level at a time, only expanding one branch at a time, and must be taken into account when paired with a layout that can traverse many levels with a single click. For example, when paired with the pack layout and a user clicks a node in the pack two levels down from the root, the collapsible radial tree expands a branch to that node and exposes the node's children. Next, assuming that the branch's children are still exposed, and a user clicks a node in the pack layout that is within a different branch outside of what is displayed, the children of the exposed branch in the tree layout are collapsed, and the new branch of children are exposed. Linking the behavior between these layouts creates a fluid connection between each type of tree diagram. This, in turn, helps users understand traversing paths and exploring nodes better.

4.4 Quiz Component

In addition to the study component, TreeVisual also includes a quiz component that tests students' knowledge about tree visualization and acquiring information from tree interactions. We also use the quiz component to assess TreeVisual's effect on helping learn tree visualization.



Number of branches (displayed / total): 29 / 29
Number of employees (displayed / total) 1,670 / 1,670

(1 of 22)

Briefly compare Treemap and Radial Tree in depicting the hierarchy of the United Kingdom Government. Which layout do you prefer?

Briefly state your answer and explain your decision.

Name: Government Structure

Description:
The tree shows the government structure of United Kingdom.

Tree hierarchy:
--------First Level
|
------Second Level
|
----Third Level
|
--Fourth Level

Source:
Government Official Websites

Selected Node: United Kingdom

Figure 7: The TreeVisual's quiz component. Compared with the study component shown in Figure 1, the right-side display panel is replaced with a Q&A panel.

We can see from Figure 7 that the user interface of the quiz component is similar to that of the study component, except for several major differences. First, the right panel is replaced with a question and answer (Q&A) part. Second, the drop-down menu for changing datasets is removed since each question is associated with a specific dataset. Third, in the control panel, the dataset description section is moved to the top to give users a quicker look at its information.

We aim at providing instructors with high flexibility in preparing quizzes. The quiz component supports three types of questions: multiple-choice questions with a single answer (radiobutton), multiple-choice questions with several answers (checkbox), and open-ended (textbox) questions. Each question can be associated with a dataset, in which case the left panel displays the corresponding tree visualization. In addition, instructors can specify available layouts for this question. The question can also be a general one without an associated dataset, in which case the left panel disappears, and the Q&A area is placed at the center.

Users sign in to the quiz component with their usernames. When they finish one question and click the "Next" button, their responses are sent and stored at the backend, and they will proceed to the following question. They can also leave the question unanswered and confirm the blank answer to move on. The quiz is a one-way process. Users cannot return to previous questions as they proceed. If they re-enter the quiz with the same username, TreeVisual will resume their progress.

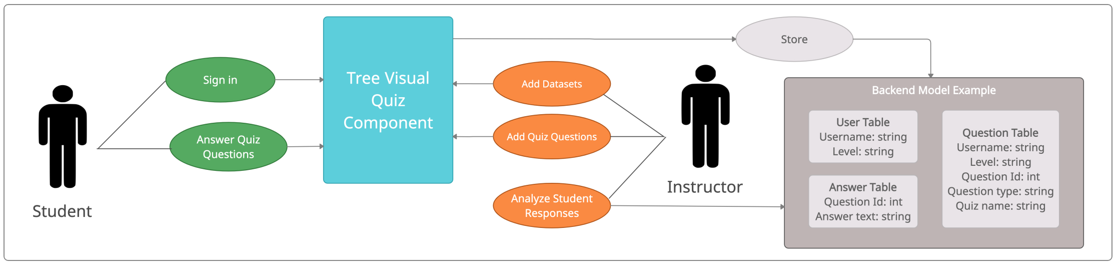


Figure 8: The use case diagram of TreeVisual.

In TreeVisual, the instructor can configure the datasets and quiz questions and access stored data based on individual student entries, as illustrated in Figure 8. First, the instructor can load hierarchical datasets into the repository folder. The dataset must be written in JSON format and has a hierarchical structure similar to the original TreeVisual datasets (i.e., attributes must match). Then, the instructor needs to update the HTML select element in the "index.js" file so that a user can select that dataset. The instructor can also add a custom quiz by adding a JSON file with a list of questions to the repository. Finally, the instructor can configure how student information and answers are stored. TreeVisual allows a user to sign in with a username and academic level. We stored the data in a PostgreSQL database accessed by a Ruby on Rails web application. Our database was queryable for both student information and their quiz responses. We implemented a feature so that students can resume their quizzes in case of interruption. We did not offer instructors our backend and database storage due to privacy concerns over student data.

5 User Study

To assess the effect of TreeVisual on learning tree visualization, we conducted a user study with two parts: a quiz on TreeVisual and an online survey. The quiz examined how well students understand the concepts and features of different tree layouts with the help of TreeVisual. The survey collected participants' detailed feedback on whether and how TreeVisual helped them learn tree visualization. These combined results revealed the effectiveness of TreeVisual.

5.1 General Information

A total of 26 students were recruited from the Data Visualization course at our university. Among them, 21 were CS majors. All participants had learned about tree visualization before the experiment. We introduced TreeVisual in one class with slides and handouts covering the basic use and features of the tool. The tool was hosted on Heroku, and students accessed it with their preferred browser. Before the class ended, all students took the *pre-quiz* with five questions (**Q1**-**Q5**) to familiarize themselves with the incoming quiz. They were then given one week to explore the study mode freely, and none had access to the actual quiz. In the following class (a week later), students completed the actual quiz with 22 questions (**Q6**-**Q27**).

5.2 Quiz

We organized the quiz using TreeVisual's quiz component. The quiz consisted of 27 questions (**Q1**-**Q5** are the pre-quiz and **Q6**-**Q27** are the actual quiz). Each question either asked about a specific dataset or asked generally about tree visualization. As mentioned in Section 4.4, each question was of one of three types: radiobutton, checkbox, or textbox. The associated dataset, type, and specific content of each question are listed in the Appendix. All responses were recorded and graded. For a radiobutton question, a user earned 1 point for correct answer and 0 otherwise; for a checkbox and textbox question, a user earned 1 point for correct answer, 0.5 point for a partially correct answer, and 0 otherwise. We also recorded the time each user took on each question.

The average score and standard deviation for each question are shown in Table 2. All participants completed the actual quiz, but one did not complete the pre-quiz. We can see that participants performed well in both quizzes, with an accuracy of around 96% in most questions, and they performed slightly better in the actual quiz than in the pre-quiz.

Q13 and **Q24** scored the lowest and were more diverse in scores than other questions. **Q13** was associated with the United Kindom's trade data. It asked about the approximate percentage of manufactured goods among United Kindom's importation, which required displaying the dataset with the sunburst layout and visually evaluating the portion by looking at the corresponding angle. One possible reason for the low score was the difficulty of this visual inspection. **Q24** asked which of the six layouts explicitly showed the path from the root to the leaves. The main reason for the low score was the vague nature of this question. While the radial tree and its collapsible version inarguably display the path across layers, the sunburst and pack layout also reflect the hierarchy and implicitly imply the path. The result of these two questions suggests that, while TreeVisual helps students learn the concepts and features of tree visualization, more instructions are needed for learning to extract information from a given tree layout and understanding the nature of different layouts.

Question	Q1	Q2	Q3	Q4	$Q5$	Q6	Q7	Q8	Q9	Q10
Avg Score	0.96	0.92	0.88	0.96	1.00	0.92	1.00	0.96	1.00	0.96
Std Dev	0.20	0.27	0.32	0.20	0.00	0.18	0.00	0.19	0.00	0.19

Question	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20
Avg Score	0.96	0.96	0.81	0.96	0.81	0.96	0.96	0.90	0.88	0.88
Std Dev	0.13	0.19	0.39	0.19	0.39	0.19	0.13	0.24	0.32	0.32

Question(s)	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q1-Q5	Q6-Q27	Q1-Q27
Avg Score	1.00	0.88	0.88	0.71	1.00	0.98	0.98	0.91	0.93	0.93
Std Dev	0.00	0.32	0.32	0.40	0.00	0.10	0.10	0.28	0.23	0.24

Table 2: The average score and standard deviation of each quiz question (**Q1**-**Q5**: pre-quiz, **Q6**-**Q27**: actual quiz).

We further split the students by their major and level and split the questions by their type and dataset. Table 3 shows the average score and standard deviation for CS (21) and non-CS (5) students and graduate (7) and undergraduate (19) students. We see no significant difference between the groups, which means that TreeVisual is designed sufficiently user-friendly and requires no prior knowledge in CS or other domains. Table 4 shows the average score and standard deviation for different types of questions and tree- and non-tree questions (i.e., whether or not the question is accompanied by tree visualization). It can be seen that participants performed better on textbox questions than radiobutton and checkbox questions. Since textbox questions concerned more on the nature of different layouts (for example, **Q17**: which layout(s) is better at depicting the trade dataset) while radiobutton and checkbox questions concerned more on dataset information, this result further corroborates our previous finding that TreeVisual is effective in helping to understand the concepts and features of tree visualization but needs more instructions on extracting information. The fact that students performed similarly on tree and non-tree questions supports this finding.

	CS	Non-CS	Grad	Undergrad
Avg Score	0.93	0.93	0.93	0.92
Std Dev	0.24	0.24	0.23	0.26

Table 3: Quiz results for students split by their major and level.

	Radiobutton	Checkbox	Textbox	Tree	Non-Tree
Avg Score	0.93	0.91	0.96	0.93	0.92
Std Dev	0.26	0.28	0.15	0.24	0.24

Table 4: Quiz results for questions split by type and tree/non-tree.

5.3 Survey

We used a survey to collect participants' comments on TreeVisual. The survey contained 27 questions (**S1**-**S27**) which were divided into four categories: *Tree Layouts*, *Datasets*, *User Interface*, and *General*. All but the last question asked participants to rate one aspect of TreeVisual from 1 to 5 (1 is strongly disagree and 5 is strongly agree). The last question was an open-ended question asking for comments. The specific questions are included in the Appendix.

We received 22 responses (four students did not complete the survey). The average rating and standard deviation for each category are shown in Table 5. Generally speaking, students rated TreeVisual positively with an average score of 4.516. However, among the four categories, *Tree Layouts* and *User Interface* were rated lower, with an average score of 4.472 and 4.462, respectively. The low score of *Tree Layouts* category was mainly attributed to two questions: **S4**: "*The treemap layout and its zoomable version help me compare their advantages and disadvantages in visualizing and exploring the tree hierarchy*", which scored 4.09, and **S6**: "*The treemap tiling strategies (binary, dice, slice, and slice & dice) help me better compare and understand their advantages and disadvantages in organizing leaves in the hierarchy*," which scored 3.55. The low score of *User Interface* was attributed to question **S12**: "*The 'About' page provides me with sufficient information to gain an overview of TreeVisual*," which scored 3.68.

These numeric results showed that TreeVisual was positively rated among participants and can improve users' understanding of tree visualization. The main reason for the low scores of treemap-related questions is the nature of treemap: it only displays all leaves and no hierarchy, thus less intuitive to be perceived as a tree. This rationale was echoed in the quiz, and several students responded that treemap was less effective in displaying different levels of a data hierarchy.

	Tree Layout	Dataset	User Interface	General	All Categories
Average Rating	4.47	4.75	4.46	4.71	4.51
Standard Deviation	0.79	0.43	0.80	0.64	0.77

Table 5: The average rating and standard deviation of each survey question category and all categories.

5.4 General Findings

We draw similar conclusions from the quiz and survey. The results show that TreeVisual helps students learn the concepts and features of tree layouts by providing well-selected datasets and interactive functions. The tool is user-friendly, and no prior knowledge in CS or other domains is needed. It also enhances users' ability to extract information from given tree data, yet is less effective than learning concepts. Also, more instructions are required to help demonstrate the use of treemaps.

6 Conclusions and Future Work

We have presented TreeVisual, an educational tool that provides an accessible web interface for students to explore and interact with hierarchical data visualizations of real-world datasets to assist in teaching and learning tree visualization. TreeVisual consists of the study and quiz components. This tool supports six tree layouts and contains datasets from four domains of different sizes and characteristics. The effectiveness of TreeVisual was measured by studying a class taken by CS and non-CS majors. In the future, we want to improve and build upon TreeVisual in the following ways. First, we will refactor the custom JavaScript linking and filtering functions for speed gain and reduced CPU usage. Second, with this refactoring in place, we will incorporate the D3.js updating capability to update the layouts as users interact with them instead of re-rendering from scratch. Third, we will consider student feedback about the layouts, such as making the warning for prefiltered data more noticeable and adding more tooltips to describe the data elements.

TreeVisual is part of our effort to create VisVisual [11], a collection of educational applications for teaching and learning data visualization. We have released TreeVisual online at `https://sites.nd.edu/chaoli-wang/visvisual`, and the code can be downloaded from `https://github.com/bohandley/tree-visual`.

Previously developed educational tools include GraphVisual [7], FlowVisual [13, 12], and VolumeVisual [2]. Each tool in the VisVisual library can be used on its own, within the classroom, or used for self-learning.

Acknowledgements

This research was supported in part by the U.S. National Science Foundation through grant DUE-1833129. We would like to thank Zhichun Guo who contributed to the initial TreeVisual development.

References

[1] The 2015 Gordon Research Conference on Visualization in Science and Education. `https://www.grc.org/visualization-in-science-and-education-conference/2015/`. Accessed: 2021-12-20.

[2] X. Bao, J. Han, and C. Wang. VolumeVisual: Design and evaluation of an educational software tool for teaching and learning volume visualization. In *Proceedings of American Society for Engineering Education Annual Conference*, pages 32.259.1–32.259.14, 2021.

[3] R. Blanch and E. Lecolinet. Browsing zoomable treemaps: Structure-aware multi-scale navigation techniques. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1248–1253, 2007.

[4] C. Chen. Top 10 unsolved information visualization problems. *IEEE Computer Graphics and Applications*, 25(4):12–16, 2005.

[5] J. Claes, I. Vanderfeesten, J. Pinggera, H. A. Reijers, B. Weber, and G. Poels. A visual analysis of the process of process modeling. *Information Systems and e-Business Management*, 13(1):147–190, 2015.

[6] C. Fiedler, W. Scheibel, D. Limberger, M. Trapp, and J. Döllner. Survey on user studies on the effectiveness of treemaps. In *Proceedings of International Symposium on Visual Information Communication and Interaction*, pages 2:1–2:10, 2020.

[7] M. Imre, W. Chang, S. Wang, C. P. Trinter, and C. Wang. GraphVisual: Design and evaluation of a web-based visualization tool for teaching and learning graph visualization. In *Proceedings of American Society for Engineering Education Annual Conference*, pages 28.501.1–28.501.15, 2020.

[8] S. Jurgensmann and H.-J. Schulz. A visual survey of tree visualization. In *IEEE Conference on Information Visualization Posters*, 2010.

[9] I. Letunic and P. Bork. Interactive tree of life (iTOL) v5: An online tool for phylogenetic tree display and annotation. *Nucleic Acids Research*, 49(W1):W293–W296, 2021.

[10] H.-J. Schulz. Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications*, 31(6):11–15, 2011.

[11] C. Wang. Visvisual: A toolkit for teaching and learning data visualization. *IEEE Computer Graphics and Applications*. Accepted.

[12] M. Wang, J. Tao, J. Ma, Y. Shen, and C. Wang. FlowVisual: A visualization app for teaching and understanding 3D flow field concepts. In *Proceedings of IS&T Conference on Visualization and Data Analysis*, pages 476–1–476–10, 2016.

[13] M. Wang, J. Tao, C. Wang, C.-K. Shene, and S. H. Kim. FlowVisual: Design and evaluation of a visualization tool for teaching 2D flow field concepts. In *Proceedings of American Society for Engineering Education Annual Conference*, pages 23.609.1–23.609.20, 2013.

[14] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of graphs with radial layout. In *Proceedings of IEEE Symposium on Information Visualization*, pages 43–50, 2001.

[15] X. Zhang and X. Yuan. Treemap visualization. *Journal of Computer-Aided Design & Computer Graphics*, 24(9):1113–1124, 2012.

A Appendix

A.1 Quiz Questions

Open-ended text questions are labeled with a †, single-choice questions with a ⋆, and multiple-choice questions with a ∼.

Pre-Quiz

Q1 ⋆ Which of the following families has the least amount of species?

Q2 ∼ In which of the following years did John Stasko publish eleven papers?

Q3 ⋆ Of the following countries, which does the United Kingdom export the most animal products to?

Q4 ⋆ Which of the following institutions in the legislative branch of the United States has the most number of employees?

Q5 † Which of the following three layouts (Sunburst, Treemap, Treemap (Zoomable)) are best at depicting the hierarchy of the United States Government? Explain your answer.

Actual Quiz

Q1 † Briefly compare Treemap and Radial Tree in depicting the hierarchy of the United Kingdom Government. Which layout do you prefer?

Q2 ⋆ Among all Leskovec's publications in 2010, which paper is cited most?

Q3 ∼ Among Leskovec's publications in 2016, which paper(s) are of type [J]?

Q4 ⋆ How many papers did Stasko publish in 2008?

Q5 ⋆ Between 2010 and 2017, on which year was Leskovec most prolific in terms of the number of papers?

Q6 † What are the pros and cons of the Pack layout when depicting the Leskovec dataset?

Q7 ⋆ Which country is the United Kindom's second-largest exporter of capital goods?

Q8 ⋆ Among the United Kindom's import volume in 2018, at roughly what percentage did manufacture accounts for?

Q9 ⋆ How many countries (regions) are both the United Kingdom's importers and exporters in vegetable?

Q10 ⋆ Which of the following goods did the United Kingdom import from Sweden in 2018?

Q11 ∼ The United States exports food and fuels to which countries?

Q12 † Considering the features of the hierarchy of trade datasets, which layout(s) is (are) better at depicting the trade datasets? Explain your answers.

Q13 † Is Treemap a good layout for depicting the United States trade dataset? Briefly explain.

Q14 ⋆ How many subspecies does the Ardeidae family have?

Q15 ∼ Which of the following species belong to the Ardeidae family?

Q16 ∼ Which of the following families have more than 30 species?

Q17 ⋆ Which of the following genuses has the most subspecies?

Q18 ⋆ Which genus does C.wintoni - De Winton's golden mole belong to?

Q19 ∼ Which of the following layout(s) shows the paths from the root to leaves?

Q20 ⋆ In the Pack layout, if node B is node A's grandchild, which of the following is true about the circles for A and B?

Q21 † What is the benefit of Radial Tree (Collapsible) compared to Radial Tree?

Q22 † Briefly compare the different ways Radial Tree and Pack use to indicate hierarchy.

Datasets Used in Quiz Questions
Pre-Quiz

Q1 Pelecaniformes (Tree Taxonomy)

Q2 John Stasko (Publication)

Q3 United Kingdom (Trade)

Q4 United States (Government)

Q5 United States (Government)

Actual Quiz

Q1 United Kingdom (Government)

Q2 Jure Leskovec (Publication)

Q3 Jure Leskovec (Publication)

Q4 John Stasko (Publication)

Q5 Jure Leskovec (Publication)

Q6 Jure Leskovec (Publication)

Q7 United Kingdom (Trade)

Q8 United Kingdom (Trade)

Q9 United Kingdom (Trade)

Q10 United Kingdom (Trade)

Q11 United States (Trade)

Q12 United States (Trade)

Q13 United States (Trade)

Q14 Pelecaniformes (Tree Taxonomy)

Q15 Pelecaniformes (Tree Taxonomy)

Q16 Pelecaniformes (Tree Taxonomy)

Q17 Afrosoricida (Tree Taxonomy)

Q18 Afrosoricida (Tree Taxonomy)

Q19 None

Q20 None

Q21 None

Q22 None

A.2 Survey Questions

Tree Layouts

S1 The number of layouts provided by TreeVisual is sufficient for me to learn about tree structures.

S2 TreeVisual helps me better understand the differences among tree layouts.

S3 The radial tree layout and its collapsible version help me compare their advantages and disadvantages in visualizing and exploring the tree hierarchy.

S4 The treemap layout and its zoomable version help me compare their advantages and disadvantages in visualizing and exploring the tree hierarchy.

S5 TreeVisual helps me distinguish tree structures better through traversing the pack, zoomable treemap, collapsible radial tree, and sunburst layouts.

S6 The treemap tiling strategies (binary, dice, slice, and slice & dice) help me better compare and understand their advantages and disadvantages in organizing leaves in the hierarchy.

S7 The linking behavior when clicking on and hovering over the nodes helped me understand the data better.

S8 TreeVisual helped me better analyze which layout works best for exploring a given dataset for a given purpose.

Datasets

S9 The datasets provided are representatives of tree structures and helped me better learn and differentiate different layout examples.

S10 The datasets provided under each category (i.e., Publication, Government, Trade, Tree Taxonomy) are sufficient for me to learn tree visualization.

User Interface

S11 The on-boarding feature helps me better get oriented with TreeVisual.

S12 The “About” page provides me with sufficient information to gain an overview of TreeVisual.

S13 The overall interface design of TreeVisual is intuitive and easy to follow.

S14 The interface is responsive with appropriate animations.

S15 The use of colors in the visualization, including the color-blind palette, clearly distinguishes different items.

S16 The “Size by” switch helps me better understand the difference between accumulated leaf nodes and accumulated size attributes of the leaf nodes.

S17 The ability to hide labels allows me to better view and explore the data.

S18 The default node size is appropriate.

S19 The “Filtering” section helps me better understand how to explore the data by filtering nodes at different levels.

S20 The “Leaves” selection helps me better identify and search for specific leaf nodes.

S21 The visual effect for user interface elements helps me distinguish disabled items or controls.

S22 The interactions are sufficient for serving the essential tree navigation need.

S23 The interactions along with the given tooltips are easy to understand and follow.

General

S24 The questions provided in the study guide helped me learn TreeVisual.

S25 TreeVisual provided an enjoyable way to learn about tree visualization.

S26 Using TreeVisual helped me feel confident in my knowledge about tree visualization.

S27 What changes to the items above, or other parts of TreeVisual, would improve your learning of tree visualization?