Tree Colors: color schemes for tree structured data

Martijn Tennekes and Edwin de Jonge

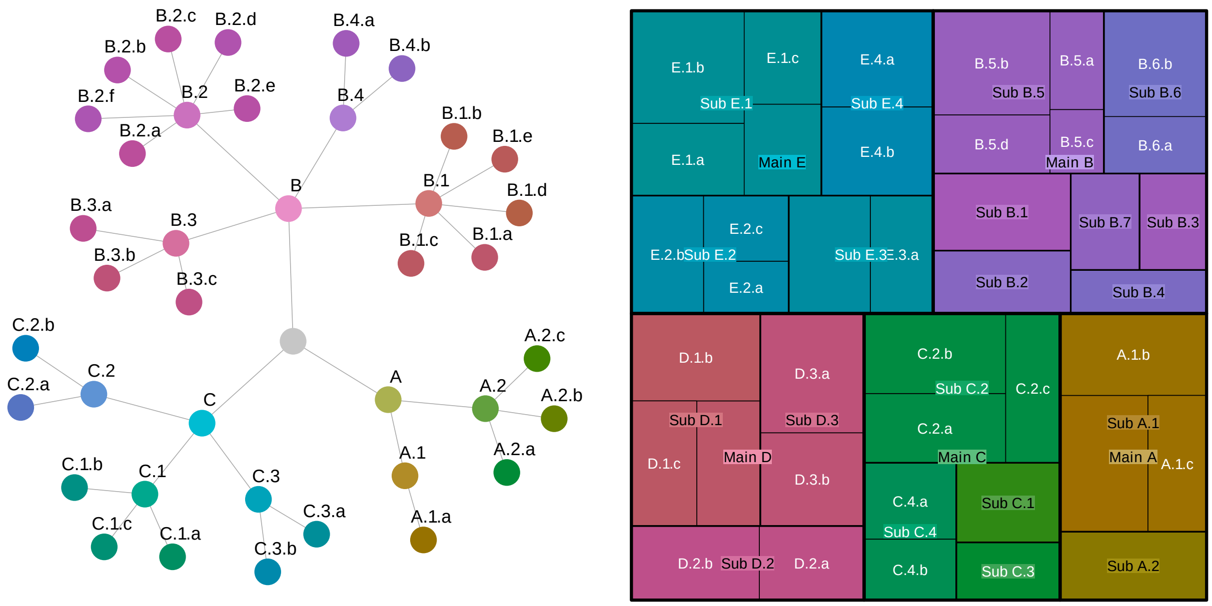


Fig. 1. Tree Colors applied to a directed graph (left) and a treemap (right).

Abstract— We present a method to map tree structures to colors from the Hue-Chroma-Luminance color model, which is known for its well balanced perceptual properties. The Tree Colors method can be tuned with several parameters, which influence on the resulting color palette is discussed in detail. We provide a free and open source implementation with sensible parameter defaults. Categorical data are very common in statistical graphics, and often these categories form a classification tree. We evaluate applying Tree Colors to tree structured data with a survey on a large group of users from a national statistical institute. Our user study suggests that tree color schemes are very useful, not only for improving standard hierarchical visualizations such as treemaps and graphs, but also for unveiling tree structure in non-hierarchical visualizations.

Index Terms—Color palettes, statistical graphics, hierarchical data.

1 Introduction

Data are often hierarchically structured. For example business data are typically broken down by economic activity and demographic data by geographic region. Visual exploration of such data should use the underlying hierarchical structure. Several data visualization methods are useful for this task, for instance treemaps [16, 18]. Color palettes reflecting the hierarchical structure would be very useful in supporting visual analysis.

Assigning colors to categories is far from trivial. On the one hand, qualitative colors should be distinct, but on the other hand they should not introduce perceptual bias by suggesting non-existent order or proximity. The selection of color palettes for categorical data first depends on the type of data. For nominal data, such as gender or nationality, qualitative color palettes are used, while for ordinal data, such as level of urbanization, sequential or diverging palettes are used [2, 21]. However, for hierarchical categories there are no specific guidelines for selecting color palettes, to the best of our knowledge.

Although many tree visualizations are proposed in literature [15], most of them use color to a small extent. A visualization technique that uses color as a major attribute is the InterRing [20], a navigation tool with a radial layout. The leaf nodes are assigned different hue values. The color of a parent node is derived by averaging the colors of its children, where larger branches have more weight. An implicit effect of this method is that colors of higher hierarchical levels are less saturated, except for one-child-per-parent branches. Hierarchical color schemes are also applied to the Hyperbolic Wheel [10], an exploration tool for hierarchical data. These color schemes are abstracted from the Hue-Saturation-Lightness (HSL) space, where brightness decreases proportional from root to leaf nodes, and where child nodes inherit the hue values from their parent nodes and add small hue values to distinct them from their siblings. However, hue values of nodes in the same hierarchical layer may be overlapping.

In our opinion, a good hierarchical color scheme should have at least have the following properties. First it should assign a unique and distinct (as possible) color to each node of the tree, since each node is a different category. The second property is that the assigned colors should reflect the tree structure: a node should have a similar color to its parent and, by transitivity, its children and siblings. Third and last the hierarchical depth of a node should be encoded in its color.

The color palettes that are generated by our proposed method are

• *Martijn Tennekes, Statistics Netherlands. E-mail: m.tennekes@cbs.nl.*
• *Edwin de Jonge, Statistics Netherlands. E-mail: e.dejonge@cbs.nl.*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

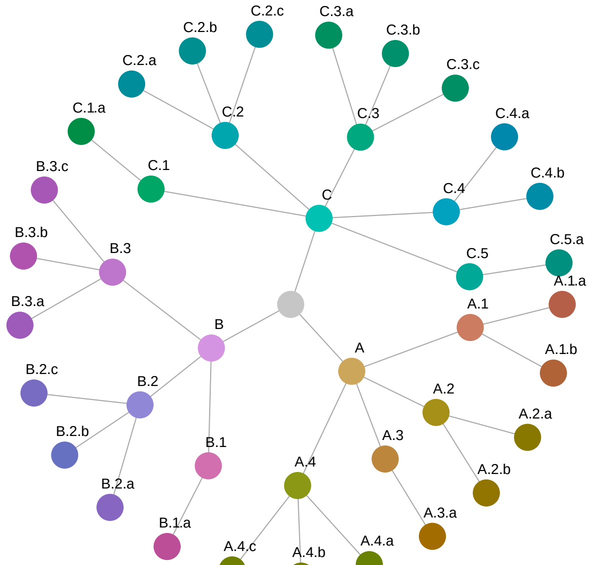
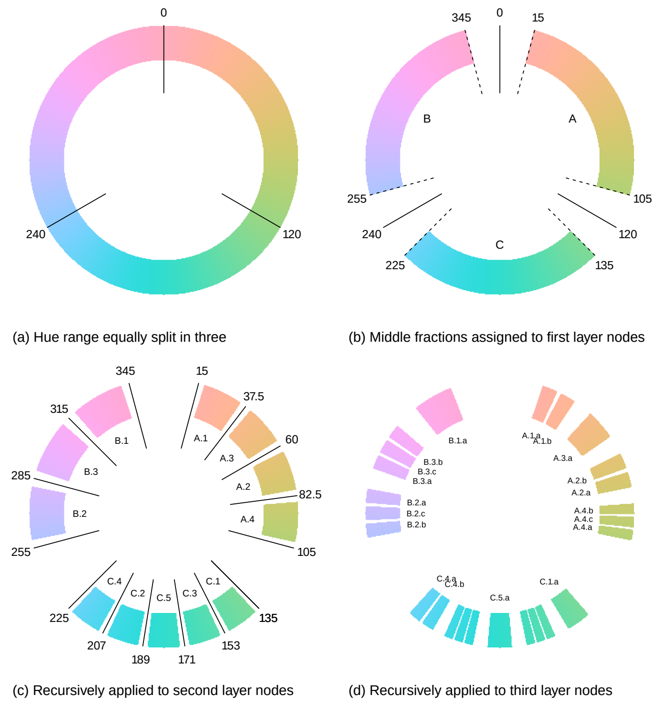


Fig. 2. Radial graph with Tree Colors.



(a) Hue range equally split in three

(b) Middle fractions assigned to first layer nodes

(c) Recursively applied to second layer nodes

(d) Recursively applied to third layer nodes

Fig. 3. Assignment of hue values.

called Tree Colors. To ensure well balanced perceptual properties, we use the Hue-Chroma-Luminance (HCL) space, a transformation of the CIELUV color space, that is designed with the aim to control human color perception [7]. Colors with different hue values are perceptually uniform in colorfulness and brightness, which does not hold for the popular Hue-Saturation-Value (HSV) and HSL color spaces [21].

This paper is outlined as follows. In Section 2 we describe the proposed method. We provide several applications of statistical graphics that use Tree Colors in Section 4. The conducted user survey to evaluate the method is described in Section 5. We conclude with a discussion in Section 6.

2 METHOD

Our method maps a tree structure on colors in HCL space, such that it reflects the hierarchical properties of the tree. We use the hue parameter H, with range [0, 360], for the tree structure, where the hue of each child node resembles the hue of its parent. The chroma and luminance parameters C and L, both with range [0, 100], are used to discriminate the different hierarchical levels.

An example of Tree Colors with a radial tree graph is shown in Figure 2. Although other graph layouts may be more suitable to highlight tree structure, for instance the Fruchterman-Reingold algorithm [6] applied in the graph in Figure 1, the applied radial Reingold-Tilford layout [13] preserves the original order of nodes in each hierarchical layer, which is in this case purely alphabetical. This layout helps to illustrate the assignment of Tree Colors to the node of a tree.

2.1 Hue values

Hue values are selected using the following recursive algorithm that assigns to each node v of a tree structure a hue value H_v. The inputs for the algorithm are a hue range r, a hue fraction f, a boolean permutation flag `perm` and a boolean reverse flag `rev`. The root of a tree starts with hue range $r = [H_{start} = 0, H_{end} = 360]$:

AssignHue(v, r, f, `perm`, `rev`)

1. Select the middle hue value in r as the hue value of node v, which is H_v [1].
2. Let N be the number of child nodes of v. If $N > 0$:
   - i divide r in N equal parts r_i with $i = 1, \dots, N$;
   - ii if `perm` then permute the r_i's;
   - iii if `rev` then reverse the even-numbered r_i's;
   - iv reduce each r_i by keeping it middle fraction f;
   - v for each child node v_i DO AssignHue(v_i, r_i, f, `perm`, `rev`).

[1] The root node itself is colored grey, so its hue is irrelevant.

This algorithm is illustrated in Figure 3. In (a) the full hue range (for a constant $C = 60$ and $L = 70$) is split in three equal parts, since the root node has three children. Of each part, only the middle fractions (with $H_{frac} = 0.75$ by default) are kept in (b). The ranges are assigned to A, B, and C. In (c) and (d) these steps are recursively taken for the deepest two hierarchical layers.

2.1.1 Hue permutations and reversals

In most hierarchical structures, there is no order between siblings. When the nodes in such structure are plotted in a linear or radial layout, the colors of the siblings should not introduce a perceptual order. Therefore, the assigned hue ranges are by default (perm=true) permuted among the siblings.

The used permutation order is based on the five-elements-permutation $[1,3,5,2,4]$. The permuted order is determined by equally spreading the siblings on a circle in the original order, and to pick the siblings at angles of 0 modulo 144 degrees. Notice that the difference of any two adjacent siblings in $[1,3,5,2,4]$ is exactly $2/5 * 360 = 144$ degrees, also between the last and the first sibling. For the cases with more than five siblings this picking angle is rounded down if needed. It may occur that a sibling is picked twice while others have not yet been picked, for instance when 360 is a multiple of the rounded picking angle. In these cases, the next sibling is picked and the process continues with the same picking angle. For the three and four siblings case we use the permutations $[1,3,2]$ and $[1,3,2,4]$ respectively.

The permutations for three to twelve siblings for a hue range between 120 (green) and 240 (blue) are depicted in Figure 4. Note that the order of the five-siblings case, which is [1, 3, 5, 2, 4], is the position of the siblings A, B, C, D, and E respectively. Therefore, the permutation of them is A, D, B, E, and C.

Furthermore, adjacent leaf nodes with a different parent should have dissimilar colors in order to differentiate between branches. Therefore, the permuted color ranges within even numbered branches are by default reversed (rev=true). This is needed because the first category

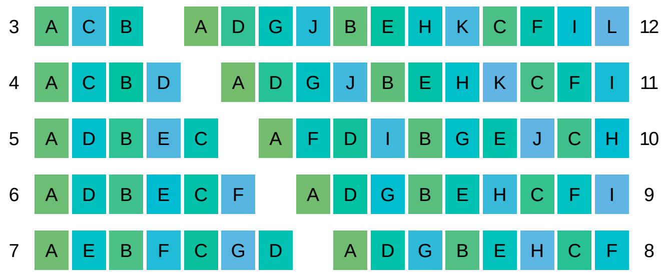


Fig. 4. Permutations of siblings.

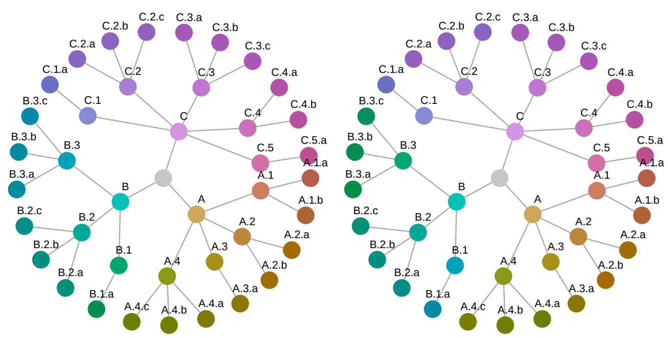


Fig. 5. Radial graphs with permutation disabled. Reversal of even-numbered branches is disabled on the left, and enabled on the right.

is always mapped to the lowest hue value, and the last category to a higher hue value. In Figure 4, category A has the most greenish color in all shown cases, while the last categories are cyan or blue. To reverse the hue ranges in an alternating way, so only the even-numbered, the hue distance between any two adjacent nodes with different parents will increase, and thus easier to tell apart.

Note that the labeling in Figure 3 shows that the assignment of colors is permuted and also reversed for even-numbered branches. The three top-layer hue ranges [0, 120], [120, 240], and [240, 360] are permuted and therefore assigned to A, C, and B respectively. Since branches A and C are odd-numbered, their hue ranges are permuted but not reserved. The permutations of these branches are [A.1, A.3, A.2, A.4], and [C.1, C.3, C.5, C.2, C.4]. Since branch B is odd-numbered, its hue ranges are not only permuted, but also reversed: [B.2, B.3, B.1].

The result of these permutations is that siblings are better discriminated in each subsequent hierarchical layer, which is illustrated in Figure 2. For comparison, the permutation is turned off in the graphs shown in Figure 5. In the left graph, the hue values form a gradual hue circle with only small hue gaps between branches, which are caused by hue fraction f (see Subsection 2.1.2). In the right graph the even-numbered branches are reversed. The leaf nodes of branch B are now more distinct from the other leaf nodes. However, the distinction between branches of the second hierarchical layer is still less than with permutation enabled (Figure 2).

2.1.2 Hue fraction

The fraction f is needed to introduce a 'hue gap' between nodes with a different parent. This choice is a trade-off between discriminating different main branches and discriminating different leaf nodes. If $f = 0$, the hue ranges are diminished to single hue points, which implies that each main branch is assigned a constant hue. On the other end of the extreme, if $f = 1$, the full hue range is available at each hierarchical layer, which makes leaf nodes easier to distinguish, but harder to take apart from leaf nodes of other branches. However for a radial or linear

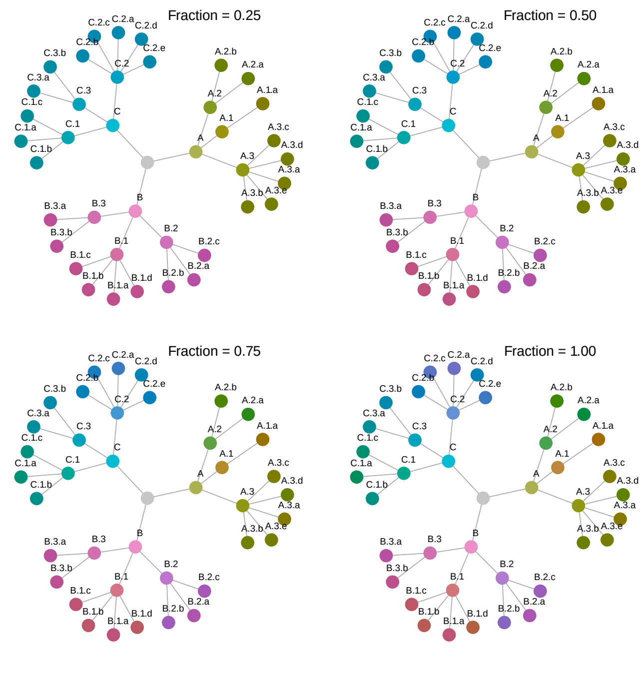


Fig. 6. Graphs with different fraction values.

layout this can be alleviated by using permutation and reversal.

The choice of f depends on several aspects, such as the application, the size and dimensions of the hierarchical data, and on the used visualization method. In Figures 6 a graph with a Fruchterman-Reingold layout [6] is shown with different values of f. For such explicit tree visualizations, high values of f can be chosen to discriminate the leaf nodes without loosing tough of the global tree structure which is clearly visible, also without Tree Colors. Even values of (or close to) 1.00 are appropriate here. To be on the save side, we suggest $f = 0.75$ for explicit tree visualizations as a guideline.

For implicit tree visualizations where the tree structure is not clearly visible without colors, lower values of f are more suitable. This is illustrated with a treemap and different values of f in Figure 7. We applied the ordered treemap layout [1] for these treemaps. For $f = 0.75$ and especially $f = 1.00$, it is difficult to quickly see the global tree structure; the main categories A and C are hard to take apart as well as the categories B and D. Therefore we suggest $f = 0.50$ as a rule of thumb for implicit tree visualizations.

2.2 Chroma and luminance values

To show depth of a node, we use bright colors for nodes high in the tree and dark colors for nodes lower in the tree. The intuition behind this approach is that dark colors are often associated by humans with high particle densities. In most tree representations, the density of nodes will increase with the depth of the tree. Therefore, we let luminance decrease linearly with depth. We set the default luminance value for the first (highest) layer below the root as $L_1 = 70$. For the other layers $i = 2, \ldots, d$, where d is the depth of the tree, the luminance value is defined as

$$L_i = (i-1)\beta^L + L_1, \quad (1)$$

where the default value for the slope parameter β^L is set to -10. In case the root node is visualized, it is colored grey. Its luminance value is specified by $L_0 = L_1 - \beta^L$.

In Figure 8 a table of colors are depicted for various chroma and luminance values and a constant hue of $H = 300$. Brighter colors (with higher values of L) have the tendency to become too saturated in our opinion, for instance, the colors with $L = 70$ and $C \geq 70$. However, for dark colors, high values of C may help to discriminate them distinguish them from other dark colors with different hue values. The question

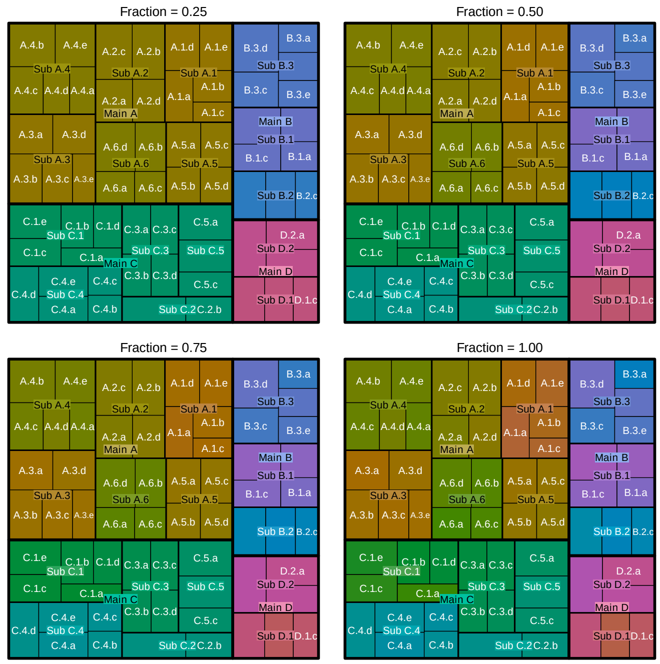


Fig. 7. Treemaps with different fraction values.

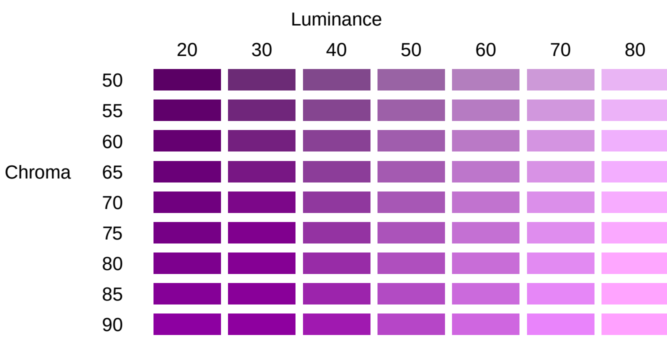


Fig. 8. Colors for different L and C values with a constant $H = 300$.

therefore is, what values of C are needed for what values of L in order to distinguish the colors easily without using too much saturation.

For different pairs of C and L, color palettes with a fixed hue range from 120 and 360 are depicted in Figure 9. Among the bright color palettes (with $L = 70$) the saturation level of $C = 60$ is sufficient to discriminate the colors easily. For the dark color paletetes (with $L = 30$), saturation values of $C = 80$ or higher are not superfluous, especially because the assigned hue range is often very narrow for nodes low in the tree.

Therefore, we propose to increase C with depth. Let $C_1 = 60$ be the chroma value for the first layer. For layer $i = 2, \ldots, d$ the chroma value is defined as

$$C_i = (i-1)\beta^C + C_1, \qquad (2)$$

where the slope parameter is set to $\beta^C = 5$ by default. The chroma value for the root node is irrelevant, since it is colored grey.

So per hierarchical layer i, we have specified a fixed pair of L_i and C_i based on the parameters L_1, β^L, C_1, and β^C. For the default values of these parameters, we depicted the pairs with a fixed hue values between 120 and 360 in Figure 10.

Although we propose default parameter values for luminance and chroma, the choice of these values depends on the number of hierchical layers in the data, and on which layer the attention is focused. The default parameter values provide colors that are well distinguishable

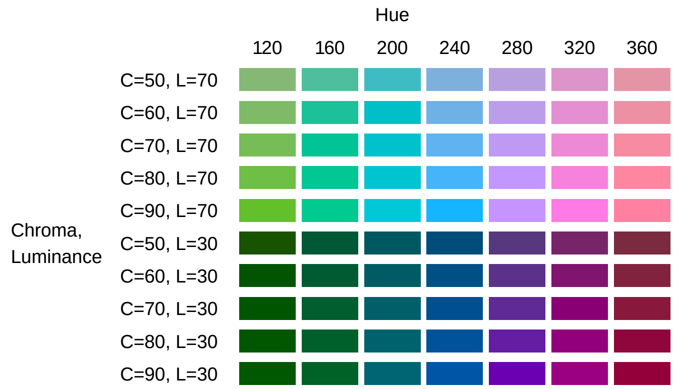


Fig. 9. Colors palettes for different pairs of L and C.

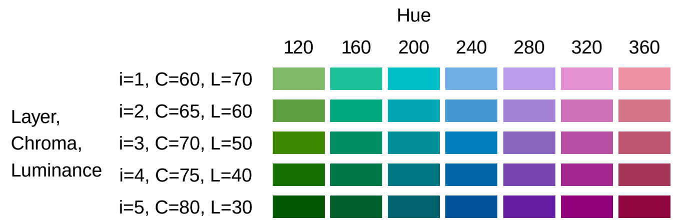


Fig. 10. Colors palettes for matched pairs of default L and C values for the top five hierarchical layers.

within the first three layers. Therefore, they are particularly useful for datasets up to three hierarchical layers. However, when only leaf nodes of the third layer are visualized, such as in treemaps of data that have a complete tree structure of depth three, a higher L_1 value may be prefered. Also, when the dataset contains five of more layers a higher value of L_1 may also be prefered. In these cases, we suggest L_1 to be 80 or even 90. In order to prevent colors that become too saturated, we suggest C_1 to be 55 or 50 in this situation. However, when more discrimination among the colors is needed, higher values of C_1 may be chosen.

2.3 Parameter overview

An overview of all parameters that are used in the described method is provided in Table 1. Due to the ranges of $[0, 100]$, the luminance and chroma parameters are restricted to the following contraints:

$$0 \le (d-1)\beta^L + L_1 \le 100 \qquad (3)$$

and

$$0 \le (d-1)\beta^C + C_1 \le 100. \qquad (4)$$

Parameter		Range	Default value
Hue start	H_{start}	0 to 360	0
Hue end	H_{end}	0 to 360	360
Hue fraction	H_f	0 to 1	0.75 (explicit) 0.50 (implicit)
Hue permutations	H_{perm}	Boolean	TRUE
Hue reverse	H_{rev}	Boolean	TRUE
Luminance first level value	L_1	0 to 100	70
Luminance slope value	β^L		-10
Chroma first level value	C_1	0 to 100	60
Chroma slope value	β^C		5

Table 1. Parameters

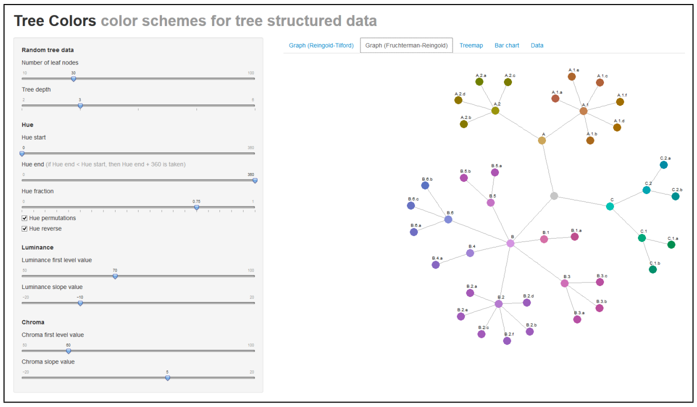


Fig. 11. Screenshot.

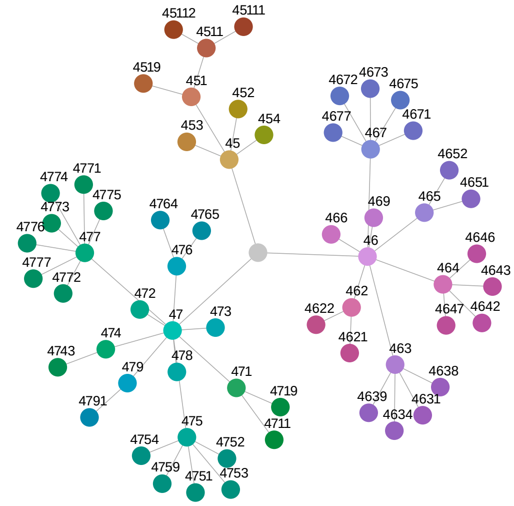


Fig. 12. Graph of all wholesale and retail trade NACE codes produced by the Fruchterman-Reingold layout algorithm

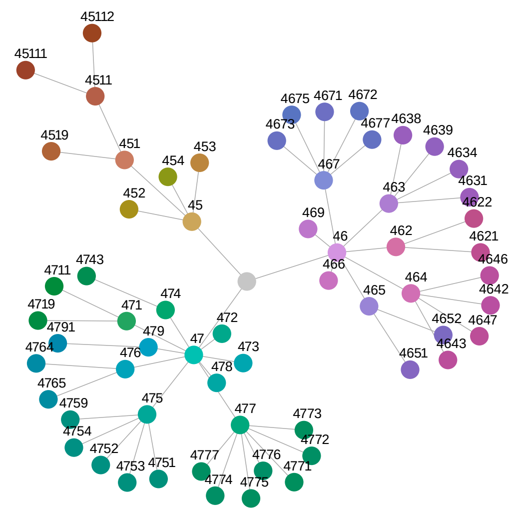


Fig. 13. Graph of all wholesale and retail trade NACE codes produced by the Kamada-Kawai layout algorithm

3 Software

The Tree Colors method is implemented in the treemap package [17] of the statistical software environment R [12] and can be downloaded from The Comprehensive R Archive Network (CRAN). All treemaps in this paper are created directly with this package, without post-processing. The implemented layout algorithms are the ordered treemap layout algorithm (pivot by size) [1] and the squarified treemap layout algorithm [3]. By default, Tree Colors are used to emphasize the hierarchical structure of the data. However, the color attribute can also be used for a second variable or to compare two hierarchical datasets [18].

The graphs in this paper are also created with the treemap package. The graph layouts are processed with the dependency package igraph [4].

To visualize Tree Colors and to tune its parameters the treemap package contains an interactive tool. After installation R and the package treemap, only two lines of code are required to start this tool:

```
library(treemap)
treecolors()
```

A screenshot of this interactive tool is shown in Figure 11. On the left side panel, the user can create random tree structured data, and experiment with the parameters. On the main panel, four visualizations can be shown: two graphs (Reingold-Tilford and Fruchterman-Reingold), a treemap and a bar chart. It also provides a table of the data that includes the color values in hexadecimal format and the HCL values of all tree nodes.

4 Applications

4.1 Econommic activity

National statistics on business enterprises are often published per economic sector. Economic sectors are typically structured hierarchically. For instance, the class of bakeries may have a parent class food manufacturers, which in turn is a child of the class of manufacturers. All member states of the European Union use same classification system of economic activity, namely the *Nomenclature statistique des activités économiques dans la Communauté européenne* (NACE) system [5]. This system has 21 main economic sectors and consists of four hierarchical layers.

In Figure 12, the NACE codes of the sector G, wholesale and retail trade, are depicted by a graph with the Fruchterman-Reingold layout algortihm [6]. In our opinion, Tree Colors help to clarify the tree structure.

Figure 13 shows the same graph with a different layout using the Kamada-Kawai algorithm [9]. In comparison to the Figure 12, the leaf nodes are all forced more outwards by the Kamada-Kawai algorithm. This may clarify the tree structure better, but at the cost of possible artifacts. In this case there are the two crossed edges, namely 465-4651 and 464-4643. The Tree Colors of the corresponding nodes help to discriminate the two local branches that overlap each other.

One of the largest European statistics that uses the NACE system are the Structural Business Statistics (SBS) that cover industry, construction, trade and services. The main target variables are turnover, number of persons employed, total purchases, and financial result. These statistics are produced for the entire European Union. However, the applied production method which include data collection, editing, analyses and estimation may vary from country to country, depending on the budget, legislation, and the availability of administrative data sources such as tax data and the chamber of commerce. As for the Netherlands, data from small enterprises, i.e. less than 10 employees, are taken directly from tax administrations, data from medium enterprises (up to 50 employees) are taken from a sample survey, and data from large enterprises (50 employees or more) are taken from a full sample survey [14].

In Figure 14, the net turnover of the Dutch wholesale and retail trade enterprises in 2011 is visualized by a treemap. The Tree Colors of the non-leaf nodes are used for the text label backgrounds. The number

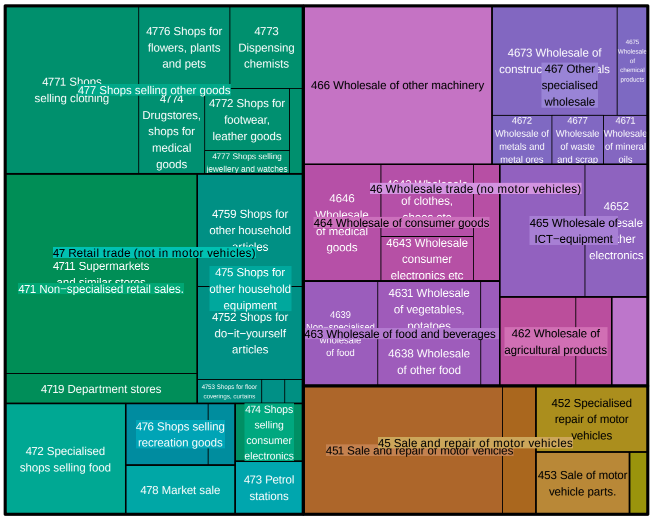


Fig. 14. Turnover among Dutch wholesale and retail trade enterprises in 2011

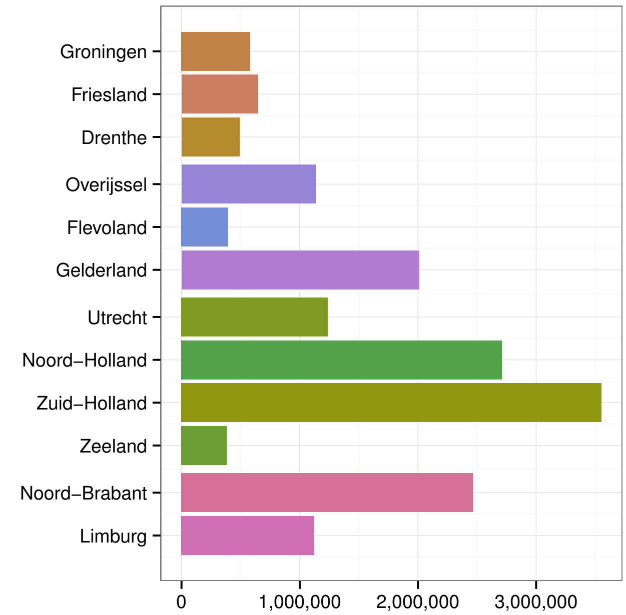


Fig. 15. Dutch population in 2012 per province

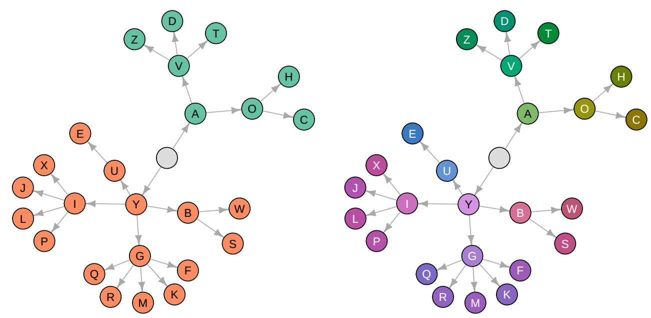


Fig. 16. Graphs applied to Dataset 1 with Main Branch Colors (left) and Tree Colors (right)

of digits in the category names represent the hierarchical layer. Each of the three main branches of this sector (45, 46 and 47) clearly has a distinct hue range. Furthermore, some leaf nodes, for instance the pink colored 466, are a brighter than others, because they represent the third rather than the fourth NACE layer.

4.2 Regional classifications

Many publications in official statistics are broken down by region, especially regarding demographics. In many situations, thematic maps are useful as a data visualization tool for spacial statistics, in particular choropleths and cartograms. However, non-spatial visualization methods are often sufficient for the task at hand. In those cases, the geographic location of the regions is less important for the analysis than the comparison of some specific target variable between regions. Tree Colors can improve the discrimination between the regions in a subtle may.

In Figure 15 a bar chart created with the R package ggplot2 [19] is depicted of the Dutch population broken down by its twelve provinces. For comparing the populations to each other, a bar chart is a good working horse, since length is perceived quite accurately [11]. We applied Tree Colors to add information about the geographic location of the provinces. Typically, the provinces are grouped by the country parts north, east, west, and south. We use those country parts as the first hierarchical layer, and the provinces themselves as the second hierarchical layer. The obtained Tree Colors discriminate the provinces while grouping the provinces by country code. To enhance the groupings, the vertical space between any two adjacent bars that represent provinces in different country parts is slightly increased.

5 User Study

A user survey is conducted to evaluate our proposed method. We let participants compare Tree Colors (TC) to Main Branch Colors (MBC). MBC is a qualitative color scale that assigns distinct qualitative colors to the children of the root and gives their offspring the same color as their parent.

5.1 User survey setup

The survey was taken by employees of Statistics Netherlands. Although no specific demographic characteristics are asked, the respondents typically have at least a bachelor's degree in quantitative sciences. Furthermore all employees are aged from 18 to 65 years old, and gender is approximately equally divided.

Many people have a color vision deficiency, which clearly effects the perception of Tree Colors. Although Tree Colors were not developed with color blindness in mind, it is important to know whether a respondent has a color vision deficiency. First, we directly asked the respondents whether they are (partially) color blind. Respondents who did no know the answer were tested for color blindness using the Ishihara test [8].

Three visualization methods were used in the survey, namely the (directed) graph, the treemap and the bar chart. For each of the three methods, respondents received questions for two charts of two different datasets, one with Tree Colors and one with Main Branch Colors, which are qualitative color palettes that are assigned to the first hierarchical layer. The colors form these palettes are taken from ColorBrewer palettes [2], which are popular in cartography and statistical visualizations.

In order to exclude the effects of all other aesthetics but color as much as possible, we distributed two different versions of the survey. Each respondent was randomly assigned to one of the version. The difference between the two versions was that the datasets for the MBC and TC questions were swapped. Therefore, each chart was tested with both color schemes but in different groups. A graph, a treemap and a bar chart that were included in the surveys are depicted in Figure 16, 17, and 18 respectively. An overview of the charts used in the two versions of the questionnaire is provided in Table 2.

Method	Color scheme	Version 1	Version 2
Graph	Main Branch Colors	Dataset 1	Dataset 2
Graph	Tree Colors	Dataset 2	Dataset 1
Treemap	Tree Colors	Dataset 3	Dataset 4
Treemap	Main Branch Colors	Dataset 4	Dataset 3
Bar chart	Main Branch Colors	Dataset 5	Dataset 6
Bar chart	Tree Colors	Dataset 6	Dataset 5

Table 2. Overview of charts in the survey

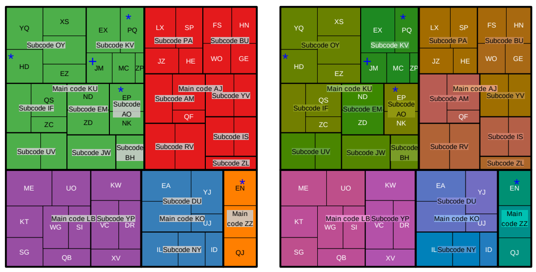


Fig. 17. Treemaps applied to Dataset 4 with Main Branch Colors (left) and Tree Colors (right)

The three visualization methods represent three levels of hierarchical explicitness. The graph clearly is an explicit tree visualization by the layout of the nodes and by the edges, which are depicted as directed arcs pointed from the root node. The treemap is an implicit visualization method, in which the hierarchical structure is less clear in comparison to the graph. Except from color, the hierarchical structure is represented by the stacking of the rectangles, the thickness of the rectangle lines, and by the label fonts. The bar chart is in essence a non-hierarchical visualization method. However, the bar charts that were included in the survey contained beside color an extra subtle hierarchical element in the spacing between the bars. The bars represent the leaf nodes of the tree structure and the amount of spacing between two bars is determined by the sibling relationship.

All datasets used in the survey are hierarchically structured with three hierarchical layers. The labels of the data points are randomized letters, such that respondents can not extract information about the hierarchy from the printed labels.

The survey contained reading and evaluation questions. Per chart, respondents got one or two reading questions and one evaluation question:

Relations Which code(s) are most similar to X? The answer options consisted of one code from a different main branch than X, one or two codes from the same main branch but a different sub branch, and one code from the same sub branch as X. From the data structure point of view, we considered the last code as the correct answer.

Offspring How many sub-codes does main branch X have? This was an open question.

Help What do you think of the following statement? The charts colors helped me to answer the question(s) above. This question is a five-level Likert item with the answer options Strongly disagree (SD), Disagree (D), Neutral (N), Agree (A) and Strongly agree (SA).

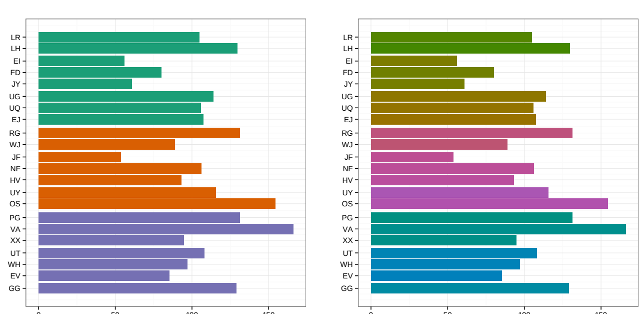


Fig. 18. Bar charts applied to Dataset 5 with Main Branch Colors (left) and Tree Colors (right)

Per visualization method, respondents where asked those reading questions for two plots of different datasets, one with Main Branch Colors and one with Tree Colors.

Next, respondents were asked to evaluate these two plots with three questions:

Prettiness Which chart is the prettiest?

Interpretation Which colors contributed most in interpreting the chart?

Overview Which colors provided the best overview?

Finally, respondent were given the possibility to write down comments or suggestions.

5.2 User survey results

We recruited 95 respondents with normal color vision and 10 respondents with a color vision deficiency.

The results of the reading questions are summarized in the left part of Figure 19. Per dataset the percentages of correct answers, that is, from a data structure point of view, are depicted as orange points for the Main Branch Colors and as green points for the Tree Colors. For each percentage, the corresponding 95% confidence interval is depicted as a line. The right part of Figure 19 shows per visualization method the distribution of answers to the question whether colors help in answering the reading questions.

Of all three visualization methods, the reading questions regarding the graphs received the highest percentages of correct answers for both color schemes. The reason is the questions are fairly easy to answer by the explicit tree layouts of graphs. The Tree Colors score slightly better on the relationship questions. The questions about the offspring resulted in 100% scores for all four color scheme - dataset combinations. As for the graphs, respondents experienced more help from Tree Colors than from Main Branch Colors.

The percentages of correct answers for the treemaps were lower than for the graphs. It turned out the many respondents also took into account other aesthetics of the rectangles, in particular aspect ratio and area size, for answering the questions. The Tree Colors scored better the Main Branch Colors on the relationship question for Dataset 3. However, the scores on this question were identical for Dataset 4. This difference may be caused by the different color scheme instances that were applied. The Main Branch Colors scored better than the Tree Colors on the questions about the number of sub-codes. A possible explanation of this result could be that respondents clustered the sub-code colors by hue. For instance, for Figure 17(left) respondents were asked how many sub-codes Main code AJ has. The correct answer is 7, while one out of four respondents thought the correct answer is 5. Probably, respondents considered subcodes PA and BU to be part of a different main code than the other 5 sub-codes. The amount of help that respondents got from the applied colors was similar for Tree Colors and Main Branch Colors.

As for the bar charts, the majority of the respondents did not observe the underlying data hierarchy when the Main Branch Colors were applied. Apparently, the spacing between the bars did not stand out clearly in comparison to other aesthetics such as color and length. For example, the question that belongs to Figure **??**(left) was which codes are most similar to UT, where the possible answers are NF, XX,

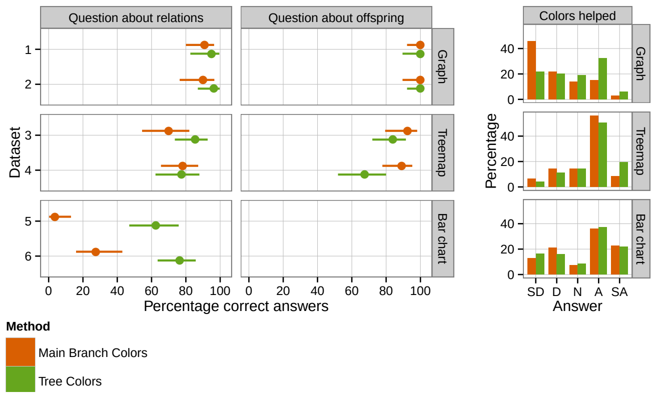


Fig. 19. Results of reading questions

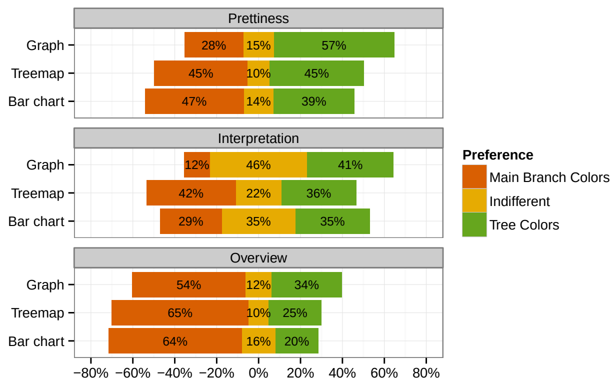


Fig. 20. Results of evaluation questions

EV, and GG. Almost 51% of the respondents chose the codes that belong to the same main branch, XX, EV and GG, and therefore only used color to answer the question. Another 27% only used length to answer the question, since they answered NF. Only 4% of the respondents answered EV, which is given the data hierarchy the correct answer. When Tree Colors were applied, see Figure 18(right), none of the respondents checked XX, EV and GG, 23% chose the equal-length bar NF, and 63% answered EV correctly. By the broadly defined question, it would be a very weak argument to claim the success of Tree Colors on this large difference. However, in our opinion it does indicate that Tree Colors are a valuable attribute to visualize a hierarchical data structure in non-hierarchical plots such as bar charts, line charts, and area charts.

The evaluation questions are summarized in Figure 20. In general, respondents liked both color schemes equally, where Tree Colors were favored for graphs.

The charts with Main Branch Colors provided the best overview for the majority of the respondents. This result is not surprising at all, since a good overview is obtained by discriminating the main branches. Recall that the hue fraction, f, is a trade-off between discrimination of the main branched and discrimination of the leaf nodes. Notice that Tree Colors with $f = 0$ are identical to Main Branch Colors, where only the applied color palette is different.

The interpretation of the graphs was experienced easier with Tree Colors except for the treemaps. This could be explained by level of difficulty of the visualization methods. Both graph and bar chart are easy to read visualization methods that the vast majority of the respondents are familiar with. Therefore, people are able to see the benefits of the additional information that the Tree Colors provide over Main Branch Colors. However, treemaps are still largely unknown by the respondents. Additional information provided by the applied color scheme could therefore be a little confusing.

6 Discussion

We proposed a method to create color palettes for tree structured data. In our opinion, this method improves both hierarchical and non-hierarchical vizualisations methods when the tree structure of the data plays a role in the analysis. Tree Colors satisfy the three properties that we described in Section 1, namely that 1) all colors of a hierarchical color scheme should be unique, 2) the colors should reflect the tree structure in terms of parent-child relationships, and 3) hierarchical depth should be encoded in color.

The conducted user study indicated that Tree Colors are often prefered over standard qualitative color palettes, that we called Main Branch Colors. Respondents read the graph, an explicit tree visualization method, slighly better with Tree Colors. Furthermore, the experienced help from Tree Colors was greater than from Main Branch Colors when applied to graphs.

However, some respondents had trouble to read treemaps with Tree Colors properly. One of the reasons is that people are not familiar with treemaps, making treemaps with Tree Colors even more complex. However, this is not the main explanation since Tree Colors should emphasize the tree structure that is already present in the treemap. Another, more valid argument is that permutation and reversals of hue ranges is developed in particular for linear and radial layouts. In these layouts, the hue values of adjacent nodes are discriminated, and even more when they have different parents. However, when the nodes are plotted in a two-dimensional space without explicit tree structure, such as treemaps, adjacent nodes may get indistinguishable colors, or even worse, siblings may unjustly be clustered. The latter happens when the plotted position of the siblings coincides with the assigned hue values. For further research, it is worthwile to improve the permutation and reversal method when Tree Colors are applied to treemaps and other two-dimensional visualization method with an implicit tree structure.

Acknowledgments

The authors wish to thank their collegues at Statistics Netherlands who participated in the user survey. Also special thanks to Marco Puts, Mattijn Morren, Jessica Solcer, Jelke Bethlehem, and Ger Snijkers for their useful suggestions.

References

[1] B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Trans. Graph.*, 21(4):833–854, 2002.

[2] C. A. Brewer, G. W. Hatchard, and M. A. Harrower. Colorbrewer in print: A catalog of color schemes for maps. *Cartography and Geographic Information Science*, 30(1):5–32, 2003.

[3] M. Bruls, K. Huizing, and J. van Wijk. Squarified treemaps. In *In Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*, pages 33–42. Press, 1999.

[4] G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[5] Eurostat. Nace rev. 2 - statistical classification of economic activities. Online publication, 2008.

[6] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Softw., Pract. Exper.*, 21(11):1129–1164, 1991.

[7] R. Ihaka. Colour for presentation graphics. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna Austria*, 2003.

[8] S. Ishihara. *Tests for colour-blindness*. Handaya, Tokyo, Hongo Harukicho, 1917.

[9] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1):7–15, Apr. 1989.

[10] H.-C. Lam and A. I. . . p. . Ivo D. Dinov, vol. 2012. Hyperbolic wheel: A novel hyperbolic space graph viewer for hierarchical information content. *ISRN Computer Graphics*, 2012:Article ID 609234, 2012.

[11] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, Apr. 1986.

[12] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. ISBN 3-900051-07-0.

[13] E. M. Reingold and J. S. Tilford. Tidier drawings of trees. *IEEE Trans. Software Eng.*, 7(2):223–228, 1981.

[14] Statistics Netherlands. Statline: Trade and industry. Online publication, 2014.

[15] H.-J. Schulz. Treevis.net: A tree visualization reference. *IEEE Comput. Graph. Appl.*, 31(6):11–15, 2011.

[16] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11:92–99, 1992.

[17] M. Tennekes. *treemap: Treemap visualization*, 2013. R package version 2.0.1.

[18] M. Tennekes and E. de Jonge. Top-down data analysis with treemaps. In *Proceedings of the International Conference on Information Visualization Theory and Applications, IVAPP 2011*, 2011.

[19] H. Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009.

[20] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, INFOVIS '02, pages 77–84, 2002.

[21] A. Zeileis, K. Hornik, and P. Murrell. Escaping rgbland: Selecting colors for statistical graphics. *Comput. Stat. Data Anal.*, 53(9):3259–3270, 2009.