WP.30
ENGLISH ONLY

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(Oslo, Norway, 24-26 September 2012)

Topic (v): Software & tools for data editing and imputation

INNOVATIVE VISUAL TOOLS FOR DATA EDITING

Prepared by Martijn Tennekes (m.tennekes@cbs.nl), Edwin de Jonge (e.dejonge@cbs.nl), and Piet Daas (pjh.daas@cbs.nl), Statistics Netherlands[1].

I. INTRODUCTION

A recent trend in data editing is top-down analysis and correction. Analysts inspect aggregated data, and in case of unexpected outcomes, zoom in on specific values causing these outcomes, and if necessary, correct them. At Statistics Netherlands, this top-down approach is used for several statistics, amongst others the Structural Business Statistics (SBS) and the Road Statistics (Aelen and Smit, 2009; Hacking et al., 2011).

In current official statistics practice, the three most common tools used to inspect data quality are: 1) tables, to inspect small data (sub)sets or aggregated data, 2) bar charts and histograms, to compare values over time or to analyse value distributions, and 3) scatter plots, to analyse the correlation of two variables.

The three tools mentioned above have several drawbacks. First of all, the number of values that can be displayed is limited to around one thousand: a scatter plot of one hundred observations may result in a clear plot, but thousands of points result in visual clutter. Secondly, the number of numerical variables that can be visualised is restricted to two. Finally, another limitation of these tools is that they cannot effectively show values at multiple aggregation levels simultaniously. Since many official statistics are hierarchically structured it is often useful to have an overview of the data errors at multiple levels.

In this paper, we discuss two visualisation methods that aim to solve the limitations mentioned above. The treemap is a visualisation method for hierarchical data. It has been developed in the early nineties of the previous century to study space usage on hard disks (Shneiderman, 1992). However, it is also useful for statistics (Tennekes and de Jonge, 2011c). The other method is the tableplot, which summarises a large multivariate dataset in a single plot (Malik et al., 2010). The tableplot can be used to detect outliers, detect unusual data patters, and to monitor the overall data quality during data editing and imputation (Tennekes et al., 2011). Both methods are implemented in R (Tennekes and de Jonge, 2011a,b; Tennekes, 2012), and are freely available on CRAN.

[1]The views expressed in this paper are those of the author and do not necessarily reflect the policies of Statistics Netherlands.

This paper is outlined as follows. In Section II and III we describe the treemap and tableplot respectively. Both methods will be illustrated with survey data from the Dutch SBS, which is an annual business survey where approximately 50,000 business enterprises respond annually. In Section IV the findings are discussed.

II. TREEMAP

A treemap is a space-filling visualisation of hierarchical data. The plotting area is recursively divided into rectangles, where the area of each rectangle corresponds to the value of an aggregate in the hierarchy. Figure 1 shows an example of a treemap. The main rectangle corresponds to the *total value added*, broken down by *economic sector* according to the Statistical Classification of Economic Activities in the European Community (NACE).

The rectangles in a treemap are characterised by two aesthetics: area size and colour. Each rectangle corresponds to a category in a hierarchical variable and its area corresponds to the aggregated value within this category. Colour is used to represent another variable and can be used in several ways. We discuss two of them: 1) the use of colour to compare data with data from a previous period, and 2) the use of colour to show the correlation with another variable. The corresponding treemap types are called the *comparison treemap* and the *density treemap* respectively.

Treemaps are well suited to support the top-down approach of data editing for two reasons. First, the quality of data is often assessed by studying relative rather than absolute differences between data of comparable units or subsequent time periods. Treemaps facilitate this comparison by both the size and colour of the rectangles shown. Second, statistical data is often hierarchically structured. With treemaps, it is possible to study the relationship at different aggregation levels.

A. Comparison treemaps

The main purpose of comparison treemaps is to detect disruptive or unexpected changes in time. These changes can be caused by real events, but are often indicative for errors in the data. Both cases are of interest: are changes occurring in a single sector? Is it the change big or small compared to other sectors? These questions can be quickly assessed with comparison treemaps.

Figure 1 shows the estimated value added (at factor cost) of all active enterprises in The Netherlands. The sizes of the rectangles correspond to the total value added in the different sectors discerned. A divergent colour scale is used to indicate the growth (or shrinkage) with respect to the previous year. White is used for values that didn't change, blue for increasing and red for decreasing values. The data in Figure 1 is displayed at the highest two NACE levels. For example, the sector *manufacturing* contains several subsectors.

The overall conclusion by inspecting this treemap, is that value added seems to be increased in most of the sectors. However, there are two red-coloured areas; indicating a decrease. A closer look at the data may reveal more insights. Figure 2 shows a more detailed look at the sector *manufacturing* (the numbers indicate the first three digits of the Dutch business classification system). This figure is less colourful than Figure 1, because the colour scale indicates the actual growth values; apparently, there are very low and very high growth percentages within the numbered sectors. Figure 2 reveals that the loss of the sector *electrical and optical equipment* (observed in Figure 1), is caused by massive losses in sector 323: *manufacturing of audio and video devices*. Business analysts are to judge whether this is an error or a real change.

B. Density treemaps

The main purpose of a density treemap is to score rectangles on another variable. The colour variable is scaled with its corresponding area size variable, effectively creating a density map. This similar to a population density cartographic map, where colour depicts the number of persons per square kilometre; the darker the colour, the more people are living in the corresponding area.

In Figure 3, a treemap is shown where the sizes are determined by the number of persons employed within the *manufacturing* section. The colours indicate how much turnover is generated per person

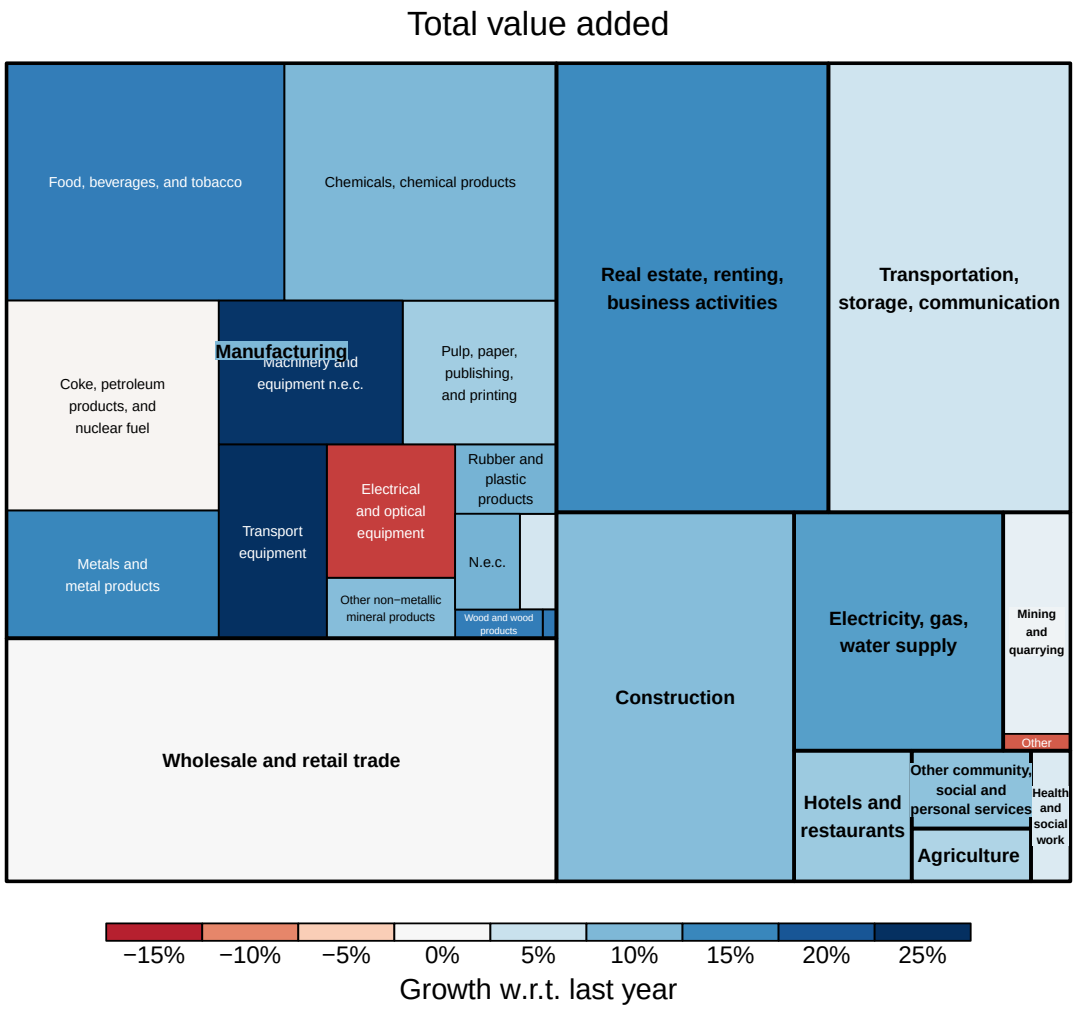

FIGURE 1. Comparison treemap: colours indicate changes in time

employed. A darker colour corresponds to a higher turnover per person. Intuitively, one can understand this treemap by interpreting each pixel as one person employed who is responsible for a certain amount of turnover. Although only a very small part of the people employed are working in the sector *coke, petroleum products, and nuclear fuel* (the red rectangle at the bottom right), this sector clearly generates a relatively large amount of turnover. If this is unexpected for a content matter expert, he can inspect if there are data errors for this sector.

It can also be interesting to create the inverse of Figure 3, where the sizes are determined by turnover and the colours indicate how many persons employed are needed to generate the turnover. Many other quantitative variables can be visualised by density treemaps, for instance the number of persons employed versus the personnel costs.

C. Implementation of treemaps in R

We implemented the treemap method as a package in R, called `treemap` (Tennekes, 2012). Our package facilitates different kinds of treemaps (a.o. the comparison and the density treemap). Colour palettes and font sizes can be adjusted as desired. Currently, the `treemap` package only generates static, non-interactive treemaps. However, with an interactive shell, such as MacroView (Hacking et al., 2011), this can easily be solved.

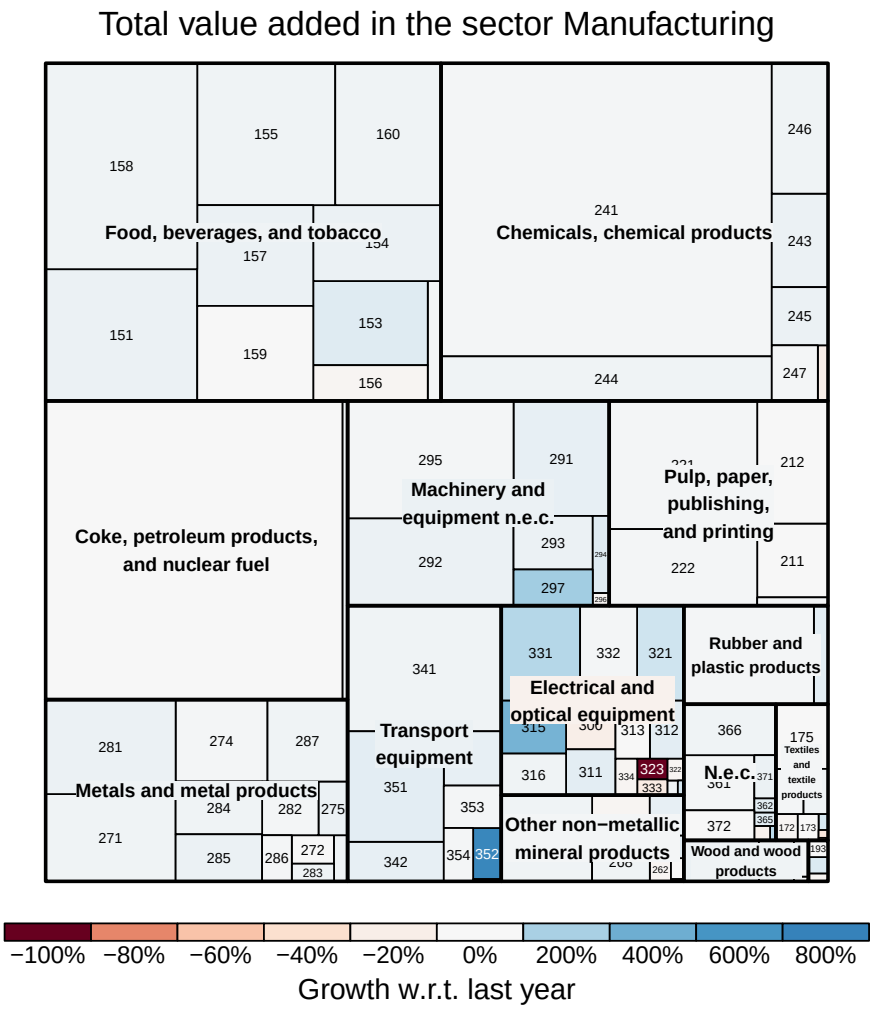

FIGURE 2. Comparison treemap of the manufacturing sector only

III. TABLEPLOT

A tableplot is a tabular shaped visualisation of a large multivariate dataset, where columns represent variables and rows groups (bins) of objects. First, the data is sorted along a variable and then equally divided into a fixed number of row bins, by default 100. For each numeric variable, the mean values per row bin are plotted as a bar chart, where the fractions of missing values determine the brightness of the bars (brighter colours indicate more missing values). For each categorical variable, a stacked bar chart of the category fractions per row bin is plotted, where missing values are depicted in red as an extra category.

Tableplots are very well suited for studying the distributions of variable values, the correlation between variables, and the occurrence and selectivity of missing values. Major advantages of the tableplot in comparison to other statistical visualisation techniques are: 1) it visualises multivariate datasets and 2) it can handle very large data sources. The amount of data that can be displayed is in theory unlimited, since data is summarised. Our R-implementation is bounded to two billion records.

There are three soft quality measures can be used to study data with tableplots: 1) the smoothness of data distributions, 2) the distribution of correlated variables, and 3) occurrence and selectivity of missing values. The application of these measures will be illustrated in the following examples.

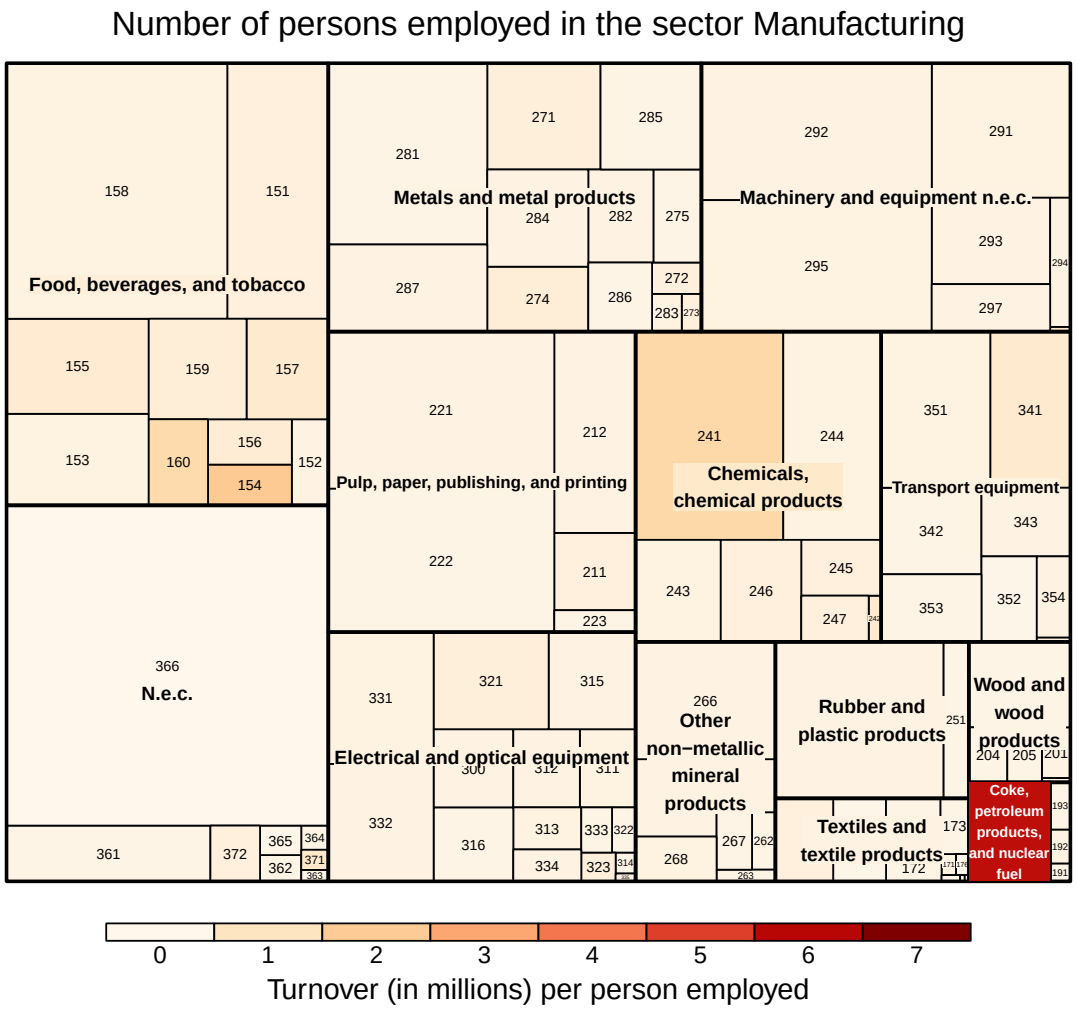

FIGURE 3. Density treemap: colours represent densities

A. Examples

Figure 4 shows a tableplot of the unprocessed SBS-survey data combined with data from the Value Added Tax (VAT) register. It contains more than 50.000 enterprises. The dataset is sorted along the SBS-reported turnover, and then divided into 100 row bins. The first two columns represent the business register variables size class (in terms of number of employees) and economic sector. The third and fourth column are the sorted SBS-turnover, respectively in numeric and categorical format. The fifth and sixth column represent the number of persons employed (SBS), the seventh column is legal form from the business register, and the last two are VAT-turnover. Observe that all numeric variables have logarithmic scales.

Consider column 3, in which the sorted variable turnover is plotted. The bottom row bin contains negative values and, indicated by its brighter colour, also missing values. In column 4, the categorical version of turnover, more information is revealed: 25% of the records have a reported turnover value of 10,000 (i.e. 10 million euro) or more. From the bottom row bin, it follows that approximately 260 turnover values (half a row bin) are missing. Furthermore, there are several negative turnover values included.

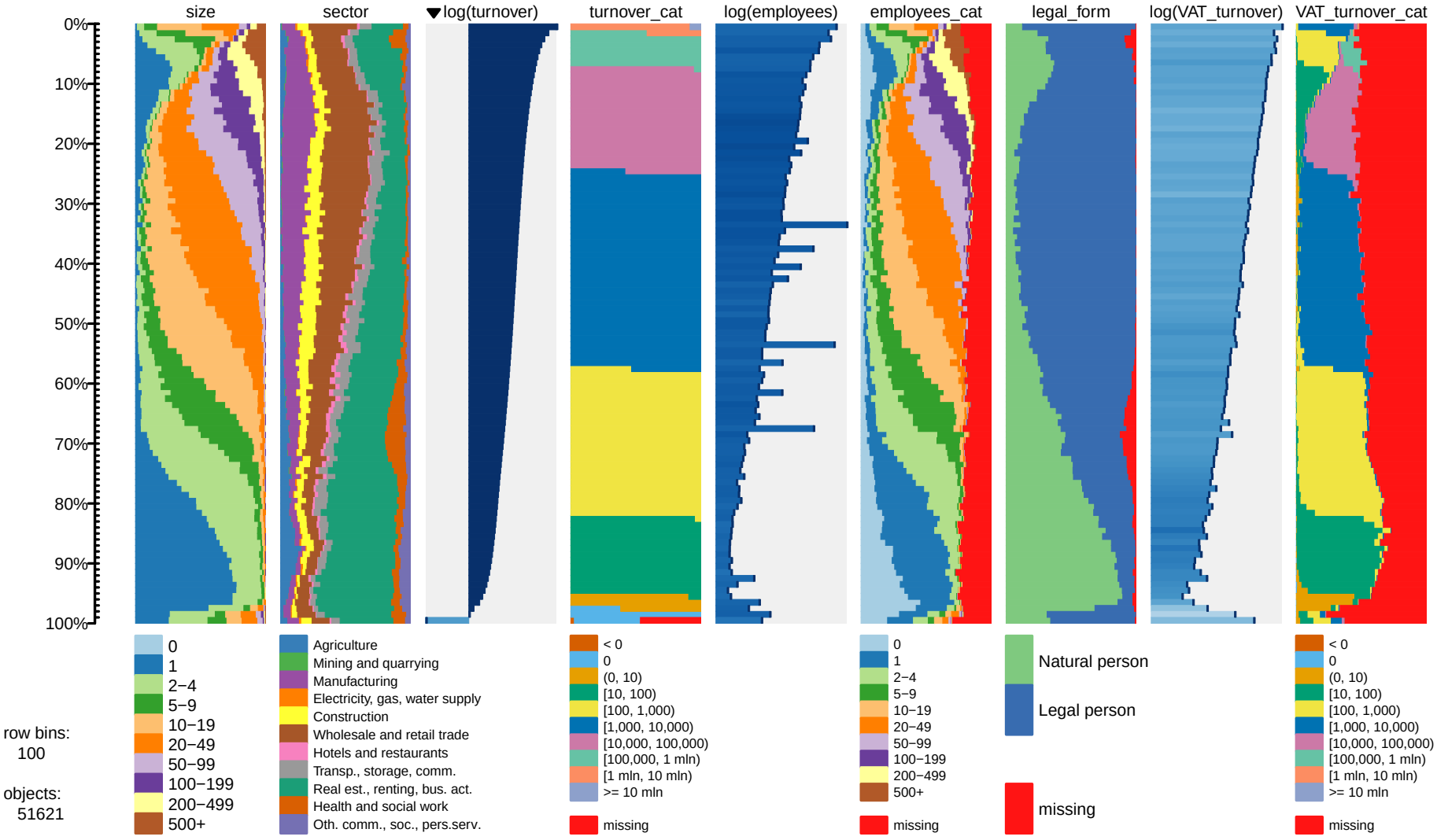

FIGURE 4. Tableplot of the Dutch Structural Business Statistics survey data combined with Value Added Tax and Business Register data.

To illustrate the quality measures mentioned earlier, we describe the figure with each of the measures.
Smoothness of data distributions It is to be expected that *employees* is highly correlated with *turnover*. Although the distributions of the values of these variables are quite similar in Figure 4, a couple of bins contain large employee values (or low turnover values), caused by representative outliers or errors. The distributions of the other variables seem fairly smooth, but are not without problems, as we will see further on.
Distribution of correlated variables The tableplot of Figure 4 reveals a few striking data patterns. The most prominent is a disturbance at the top (between 0% and 15%) observed in *size*, *sector*, *employees_cat*, *legal_form*, and *VAT_turnover_cat*. This pattern is caused by the occurrence of so-called unit measure errors: in this case, respondents overlook that economic values should be reported in thousands of euros instead of euros.
Selectivity of missing values The variable *employees* contains many missing values, almost 25% (obvious in the sixth and seven column). The missing values of *legal_form* (column 7) are caused by the fact that businesses within the sector Health and social work are not obligated to register at the Chamber of Commerce in the Netherlands. *VAT_turnover* also contains many missing values, especially for large businesses. This is caused by differences between legal and practical organisational structures.

To illustrate the usefulness of tableplots for monitoring the data editing and imputation process, we distinguish three stages of data processing: unprocessed, edited, and prepared for final publication. For each stage, a tableplot was generated (Figure 5).
Smoothness of data distributions The tableplots of each subsequent production stage show increasing smoothness of the data. The unprocessed dataset contains a lot of noise suggesting many errors. The edited dataset is already a lot smoother, and the final dataset even more. Exception is the variable *book_profit* which still contains negative values in the final dataset, but this may be correct.
Distribution of correlated variables Besides the disturbance at the top that was already observed in Figure 4, there is also a clear disturbance at the bottom three row bins. Enterprises with an unprocessed turnover value of 0 or less clearly have other characteristics than enterprises with a small positive turnover value. Comparison of the unprocessed data and the edited data reveals that the distributions of the variables, especially those of the categorical variables, in the edited data are more in line with turnover. Also, the suspicious set of enterprises with a turnover of 0 or less is reduced to the lowest bin; about one percent. The final SBS-dataset obtained contains nicely distributed variables. However, a suspicious set of enterprises can still be observed in the bottom row bin. These are very likely subsidised companies. The disturbance at the top appears to have been solved during data editing.
Selectivity of missing values We already observed that there are missing *turnover* values in the bottom row bin, and that a lot of *employee* values are missing in the unprocessed dataset. Also the variable *book_profits* contains many missing values in the unprocessed dataset. The edited dataset hardly contains missing values, except for the variable *book_profit*. The final dataset does not have any values missing (all bars of the numeric variables are dark blue). This clearly shows the result of the applied data imputation strategy.

B. Implementation of tableplots in R

We implemented the tableplot in R-packages `tabplot` and `tabplotGTK` (Tennekes and de Jonge, 2011a,b). The former package enables users to create tableplots by command line interface. The latter package is a graphical user interface shell.
Our implementation facilitates the following settings. We recommend to experiment with them.

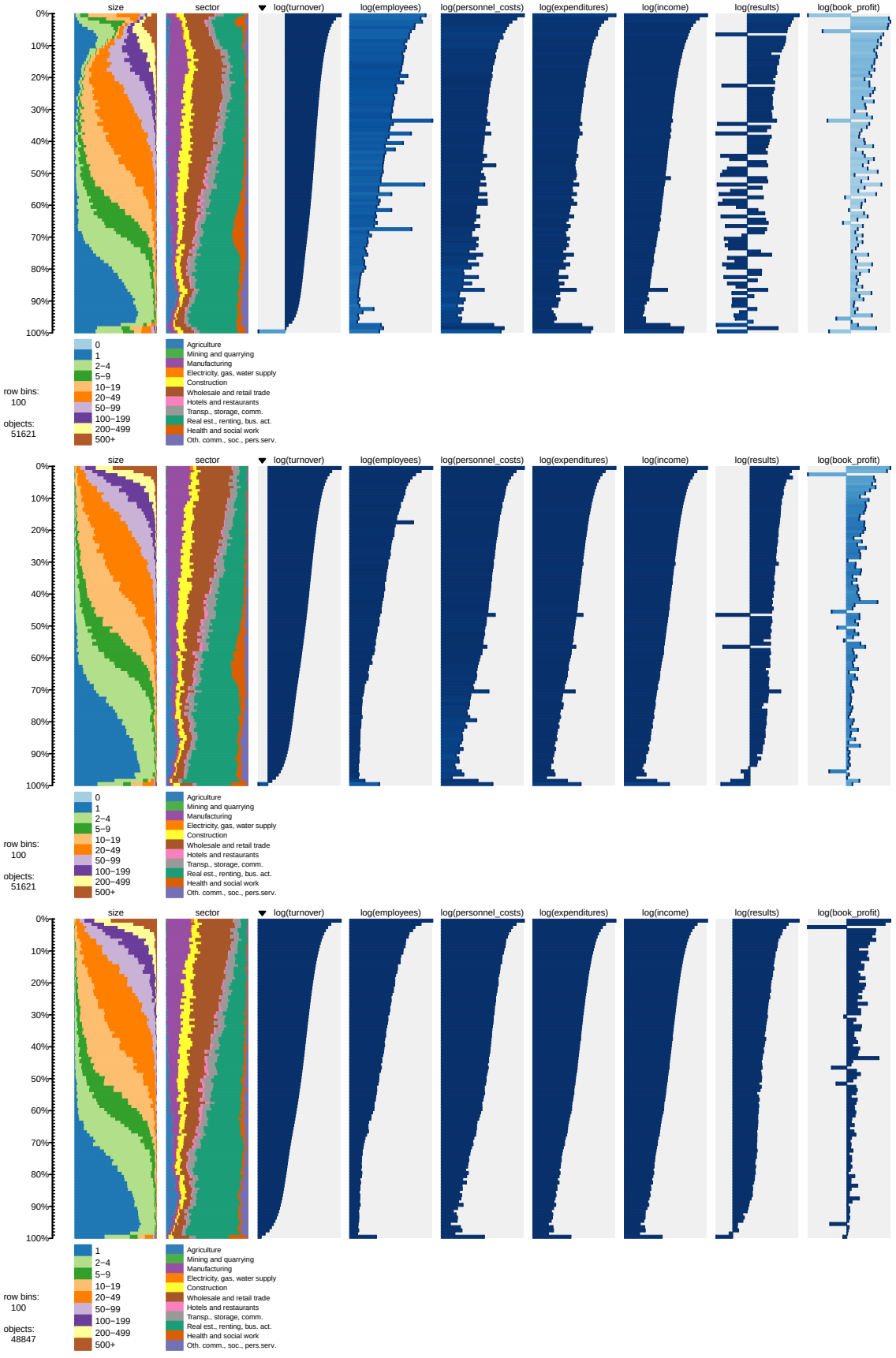

FIGURE 5. Tableplots of three SBS-data stages: unprocessed, edited, and final.

The number of row bins: Analogue to histograms, the number of used bins is a trade-off between a good polished but potentially meaningless plot of the data and a very detailed but noisy plot. Our experiences are that a number of 100 bins is a good starting value, independent of the size of the dataset.

The sorted variable: It is important to have a suitable sorting variable. It is highly recommended to use a numerical variable, since sorting on categorical variables results in a block-type tableplot that provides less information.

Zoom window: The range of the vertical axis indicates what part of the dataset is shown. It is possible to change this range, while keeping the same number of row bins. By zooming in, more detailed information is revealed.

Data filter: To study particular groups within the data, a filter can be applied. For instance, the SBS-data could be filtered by *sector* such that for each sector a distinct tableplot is generated.

Categorisation: To more clearly study the data distribution of a numerical variable, it is often useful to additionally include a categorical variant. In this way, more information about the data distributions within the bins is revealed. See for instance the variable *employees* in Figure 4 (columns 5 and 6).

It is possible to create tableplots for datasets with a theoretical upper limit of two billion (2^{31}) records. So far, we have succesfully created a tableplots of a dataset consisting of 54 million records on a modest personal computer. The operation time was about six minutes.
A detailed description of the `tabplot` and `tabplotGTK` packages can be found in the help files and in the vignette (Tennekes and de Jonge, 2011a,b).

IV. DISCUSSION

We have described two visualisation methods that can support the data editing and imputation process. Both methods are able to reveal potential data errors.
The strength of the treemap is that navigation throughout different aggregation layers is easy and intuitive. This makes this method well suited for top-down editing. A disadvantage of treemaps is that missing values cannot be visualised. Moreover, treemaps of data where missing values are omitted do not make a lot of sense. This is a common problem in top-down data editing: total values can only be estimated when the data is complete, or when missing values are (temporarily) imputed. Another disadvantage of treemaps is that the position of the rectangles cannot be determined beforehand. However, the applied layout algorithm respects the determined positions as much as possible while minimising the aspect ratios of the rectangles.
Tableplots are well suited to explore large datasets, especially due to the handling of missing values and the ability to show a dozen of variables of a dataset. Notice that the tableplot is a multiple bivariate rather than a multivariate method: it shows the relationship between the sorted and each other variable.
A major difference with the treemap is that the tableplot is essentially a bottom-up method: records are aggregated according to the order of the sorted variable and not according to a provided aggregation scheme. The tableplot can be used complementary during top-down data editing: it shows data errors that may be difficult to find with top-down editing. A tableplot can be created for a filtered data set, for instance a specific economic sector, and from that point the data can be inspected by zooming in.
At the time of writing this paper, both methods have not yet been taken into production. Plans are to embed the methods in the top-down editing tool of Statistics Netherlands, MacroView, which supports R visualisations. However, the interface between MacroView and R should be improved, especially regarding interaction. The treemap is easier to integrate with MacroView than the tableplot, since the treemap is a truly top-down method that follows the different aggregation levels of the data. Moreover, the tableplot should preferable be used at full screen size, since it has a rather high information density. Currently, we are working on a highly interactive web-based interface for the `tabplot` package.

References

Aelen, F. and Smit, R. (2009). Towards an efficient data editing strategy for economic statistics at statistics netherlands. European Establishment Statistics Workshop.

Hacking, W., Ossen, S., Meijers, R., and Kruiskamp, P. (2011). Macroview: a generic software package for developing macro-editing tools. In *Work Session on Statistical Data Editing*.

Malik, W., Unwin, A., and Gribov, A. (2010). An interactive graphical system for visualizing data quality - tableplot graphics. In Loracek-Junge, H. and Weihs, C., editors, *Classification as a Tool for Research, Proceedings of the 11th IFCS Conference*, pages 331–339. Berlin: Springer.

Shneiderman, B. (1992). Tree visualization with treemaps. a 2d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99.

Tennekes, M. (2012). *treemap: Treemap visualization*. R package version 1.0-3.

Tennekes, M. and de Jonge, E. (2011a). *tabplot: Tableplot, a visualization of large datasets*. R package version 0.11.

Tennekes, M. and de Jonge, E. (2011b). *tabplotGTK: A graphical user interface for the tabplot package*. R package version 0.5.

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

Tennekes, M., de Jonge, E., and Daas, P. (2011). Visual profiling of large statistical datasets. In *Proceedings of the New Techniques and Technologies for Statistics conference, NTTS 2011*.