Tabulator: Automatic Table Design for the Web

Pontus Orraryd*
Oskar Ankarberg†

1 Literature Review

This is an early version of "Related Work" that will be in our paper:

1.1 Automatic Graphic Design

Automating the design of graphs is nothing new. Mackinlay introduced APT back in 1986 which set the groundwork for a systematic way to automate visualizations of relational data[Mackinlay 1986]. Polaris [Pat Hanrahan 2002] (later commercialized as Tableau) uses the VizQL specification language which was based on APT. While our system will not use much of their exact ideas since we use web technologies for the actual drawing of our tables, we will still look at how these systems work.

1.2 Table Design

Tables may be one of the oldest ways to present data. A lot of research have been done on what makes a good table. Tufte said that a good table should be free from visual clutter ("chartjunk") [Tufte 1990]. Tschichold said that "(rules) should be used only when they are absolutely necessary". [Tschichold 1967]

Stephen Fews book describes multiple ways to improve table design.[Fry 2004] He identifies how tables can be divided into two categories: quantitative-to-categorical or quantitative-to-quantitative. Similarly to Tufte, Fry puts a lot of emphasis on white space and how it is vital in making it easy to quickly scan the data. He goes into detail on different design choices and will serve as our main guide when we implement our system.

1.3 Web framework

https://datatables.net is a framework for designing tables on the web. The framework does not aim to provide the most correct scientific visualization related to the provided data, it is rather a tool for developers to generate tables that "look good", and not always display the data in an effective way according to scientific principles [Tufte 1990] [Fry 2004]. Our work is different since we strive to do the opposite. Our system will strive create tables that follow design principles laid out by professionals in the data visualization field.

Another web app, which is based upon Bertins matrix analysis method[Bertin 1981], is Bertifier [Perin et al.], which is an interactive tool for exploring tabular data. The tool allows you to reorder the data, add graphs and much more. It is more intended for exploratory data analysis than for creating effective graphics. While we will certainly use certain ideas from this app, our solution targets creating well produced tables for publishing.

*orraryd@stanford.edu
†oankarbe@stanford.edu

2 Project Milestones

Our system is implemented in Javascript and is using D3.js. We currently have a very basic version which can load a csv file and draw a decent looking table.

In spirit of agile design, we have identified tasks we hope to implement. We hope to get most of these done by the final deadline:

- **Consider more design principles**: We still feel we need to read more about how to design effective tables so we will review some readings once our system is in a good state.
- **Repeated Nominal Values**: Add function for sorting and removing repeated nominal values, as suggested by Few.
- **Find a solution for displaying numbers correct**: We need a concise way of displaying and arranging numbers w/o decimals in each column.
- **Drag columns**:The user should be able to move around a column and swap place with other columns.
- **Adjustable margin per column and row**: Our system should (either automatically or by user interaction) be able to add margins to individual table cells or entire columns.
- **Automatic text formatting**: We want to add some automatic text formatting for our system. For example, we'd like the system to recognize names in camel case or snake case and automatically add an underscore between the words. (for example *number_of_items* would become *Number of items*)
- **File uploading and handling**: We currently have a csv file hardcoded into the system. We need to implement a way for the user to upload any type of file. Preferably we want to extend the system to support multiple file formats.
- **Save to file**: Related to the previous point, we need to add functionality to save the table to an image and/or pdf.
- **Handle ordinal data**: We still have not figured out a good way to deal with ordinal data. We want to determine when a column contains ordinal data and potentially somehow identify the order of the different elements for sorting.
- **Arrange the first column without it's header**: We want to be able to identify when the first column is not really data, but categories (In other words, the table is bidirectional). If that is the case we will format the column appropiately.
- **User should be able to define datatypes**: The user should be able to define each columns data type, Ordinal, Nominal, Ratio and Interval in cases where system fail to do so automatically.

References

BERTIN, J. 1981. *Graphics and Graphic Information Processing*. Walter De Gruyter Inc.

BOSTOCK, M. D3. `https://d3js.org/`.

FRY, S. 2004. *Show Me the Numbers*. Analytics Press.

JOCK D. MACKINLAY, P. H., AND STOLTE, C. 2007. Show me: Automatic presentation for visual analysis. *IEEE Trans Visualization Comp Graphics 13*, 6, 1137–1144.

MACKINLAY, J. 1986. *Automating the Design of Graphical Presentations of Relational Information*. PhD thesis, Stanford University.

PAT HANRAHAN, CHRIS STOLTE, D. T. 2002. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. *IEEE Trans on Visualization Comp Graphics 8*, 1.

PERIN, C., DRAGICEVIC, P., AND FEKETE, J.-D. Bertifier. `http://www.bertifier.com/`.

TSCHICHOLD, J. 1967. *Asymmetric Typography*. Reinhold Publishing.

TUFTE, E. R. 1990. *Envisioning Information*. Graphics Press LLC.

TUFTE, E. R. 2001. *The Visual Display of Quantitative Information*. Graphics Press LLC.