A review of interactive graphics for data exploration and analysis

Tom Elliott

iNZight Analytics Ltd

University of Auckland

Abstract

1 Introduction

- The importance of data visualization in data analysis
- How tools have changed over the years
- Web front-end technologies are now the dominant platform
- R has a rich ecosystem of packages for data visualization
- A range of tools for specific tasks, or out-dated technologies

Discussions by Cook, 2007, Theus and Urbanek, 2014, and Ward, 2015 have highlighted the importance of data visualization in the data analysis process. There are a wide range of visualisations and definitions of interactivity, which have evolved over time and vary depending by context. In this short review, we will focus on interactive graphics for data exploration and analysis, with an emphasis on the tools and frameworks used to produce them, specifically for reproduction and extension using R and web technologies.

2 Interactive graphics

There are many definition of interactive graphics depending on the context. In our case, we are interested in the technical aspects of creating interactive graphics for data analysis, specifically the availability of web technologies to create interactive graphics that are generated and powered by R. In this context, an interactive graphic is one that allows users to interact with the data, using the graphic as a user interface (UI) (Young, 2011). Users should be able to dig down into the data (filtering, subsetting, and faceting) as well as extract summaries (e.g., means, medians, etc.) and other statistics.

Another key aspect in our context is the ability to link multiple graphics together. This allows users to create a selection of grahpics that are automatically linked together, and interaction with one is echoed in the others. In ViSta (Young & Bann, 1996), such displays are referred to as a *SpreadPlot* (Young et al., 2003), which are pre-defined sets of linked graphics for exploring a certain idea. For example, cliking a bar in a bar chart might highlight (the parts of) bars in other charts, or highlight points in a scatterplot that fall into the selected category.

Some common techniques used in these graphics include:

- Brushing, which is the act of selecting a subset of data points by dragging the mouse over the graphic.
- buttons that alter the plot in some way
- clicking (on points, objects, etc)
- drag-and-drop of plot features (e.g., axes, order of categories)
- context menus (right-click drop-downs) that provide access to common tasks, such as adding trend lines, filtering values, etc.

There are existing tools that allow users to do the above, such as ViSta (Young, 2011; Young & Bann, 1996) and Mondrian (Theus & Urbanek, 2014). Mondrian provides a simple menu-based interface driven by variable selection, allowing users to choose variables and then decide which chart type to use. Created charts are opened in a separate window, and automatically get linked to any existing or future charts. This allows users to quickly build up a set of linked graphics that can be used to explore the data. In their book, Theus and Urbanek, 2014 provide a comprehensive overview of the Mondrian system, including demonstrations of exploratative data analysis through the use of interactive linked graphics. Perhaps one downside of Mondrian is that it is a desktop application, and as such is limited to the local machine. Also, it uses outdated technology (Java) [check this], and is not easily extensible as say a web-based system.

ViSta (Visual Statistics) is another system that provides interactive graphics for data exploration and analysis. It places emphasis on the full analysis process, with pre-defined sets of linked graphics (SpreadPlots) that allow users to explore the data in a structured way. These are, arguably, more opinionated than Mondrian, but provide a more structured approach to data analysis.

Another new development is plotscaper, an R and Typescript package that allows users to create interactive linked graphics, with an emphasis on portraying summary statistics across linked plots Bartonicek, 2024. This work is still in development, but shows promise for creating a modern web-based system for interactive graphics in R. However, not yet sophisticated or feature-rich.

The key points here are that these tools invite users to explore data, rather than simply look at it, and the level of interactivity is closer to the data than other more static graphics, for example as generated using plotly or similar tools. The end goal of our wider project is to develop a data analytivs environment.

Others to discuss:

- Plotly — web-based interactive graphics, but have limited functionality for linked graphics, and are more focused on the visualisation aspect.
- d3 — a powerful web-based graphics library that allows users to create custom graphics, but requires a lot of technical expertise.

3 Interactive data analysis

Interactive data analysis allows users (here analysts) to ask questions of the data and get answers to those questions, through models and visualizations. Typically, this is done through software such as R or SAS or Python, where users can iterate through models and tweak their results. Visualising the data is an integral part of the data analysis pipeline, and is used not only for exploratory data analysis, but also checking models and communicating results.

We are interested in extending the interactive graphics component to more comprehensively cover the data analysis process. This is how Mondrian works, allowing users to build models (trend lines, etc.) and explore the data visually quite extensively without the need to write code or fit formal models. The key features that allow this are:

- Linked graphics, which allows users to see how variables interact with each other more easily.
- UI commands, such as context menus (right-click drop-downs) that provide access to common tasks, such as adding trend lines, filtering values, etc.

4 Next steps

```
- What are the key features of interactive data analysis/visualization?
```

`- Whare are some use cases?`

`- Here's our proposal using the rserve ecosystem`

As part of a wider project to develop tools for building modern front-end apps powered by R, using `Rserve`, we are looking at how to integrate interactive graphics into this ecosystem. A key feature will be the dissociation of the front-end graphics from the back-end microdata, which could pave the way for privacy-preserving data analysis and visualisation tools.

One issue faced by research organisations is the technical expertise required to control access to data, and often expensive virtual environments are created to allow researchers to access the data. Alternatively, a selection of outputs are created and published, which can be time consuming and restrict what data is available.

We want to develop an ecosystem based on disparate front-end and back-end systems, where the front-end is a web-based application that can be easily deployed and accessed, while the back-end is a secure R server that provides fine-tuned control over what results get shared with the front-end. Confidentialisation, rounding, and suppression techniques can be used to ensure that no individual data points are shared with the front-end, while still allowing users to explore the data through interactive graphics such as bar charts and histograms.

Some key features of our proposed system are:

- Linked graphics, which allow users to explore the data more easily, but specifically do the linking on the server. This requires that charts are updated from the server with new data or entirely new chart types as the user interacts with the data.

- Summaries and statistical information, which can be calculated on the server and sent to the front-end for display, such as trend lines, means, confidence intervals, etc.

- Fine-grained access control, so users have access to only the data they are allowed to see, and no more, with all outputs generated at request time (not cached).

- Sharing of results, so users can save their work and share it with others (who must have the appropriate access rights, otherwise the shared URL will not work).

Another advantage of this system is that it allows users to interact with large, real-time datasets that are analysed and processed on powerful servers, rather than on the user's local machine. This is also useful for long-running processes, for example fitting complex models or running simulations. The user can start the job and then wait until the results are available. In such cases caching on the server can be used to ensure that results are available even if the user closes the connection and returns later.

The key to building this system will be to create some simple widget libraries that allow users to easily get started with interactive graphics, and then extend these to more complex use cases and custom chart types.

5 Conclusion

Acknowledgements

Supported by the Ngā Puanga Pūtaiao Fellowships from Government funding, administered by the Royal Society Te Apārangi.

Acronyms

UI user interface

References

Bartonicek, A. (2024). Plotscaper: Explore your data with interactive figures. https://doi.org/10.32614/cran.package.plotscaper

Cook, D. (2007). *Interactive and dynamic graphics for data analysis: With r and ggobi* (D. F. Swayne, Ed.) [Includes bibliographical references (p. 177-184) and index]. Springer Science+Business Media, LLC.

Theus, M., & Urbanek, S. (2014). *Interactive Graphics for Data Analysis: Principles and Examples*. Chapman; Hall/CRC. https://doi.org/10.1201/b17187

Ward, M. (2015). *Interactive data visualization: Foundations, techniques, and applications* (G. G. Grinstein & D. Keim, Eds.; 2. ed.) [Includes bibliographical references and index]. CRC Press, Taylor & Francis Group, an AK Peters book.

Young, F. W. (2011). *Visual statistics: Seeing data with dynamic interactive graphics* (P. Valero-Mora & M. Friendly, Eds.; Online-Ausg.) [Includes bibliographical references (p. 339-349) and indexes. - Electronic reproduction; Palo Alto, Calif; ebrary; 2011; Available via World Wide Web; Access may be limited to ebrary affiliated libraries]. Wiley-Interscience.

Young, F. W., & Bann, C. M. (1996). *ViSta: The visual statistics system* (tech. rep.). Technical Report 94–1 (c), UNC LL Thurstone Psychometric Laboratory Research Memorandum.

Young, F. W., Valero-Mora, P., Faldowski, R. A., & Bann, C. M. (2003). Gossip: The architecture of spreadplots. *Journal of Computational and Graphical Statistics*, *12*(1), 80–100. https://doi.org/10.1198/1061860031356