Journal of Statistical Software

May 2012, Volume 47, Software Review 2. http://www.jstatsoft.org/

Reviewer: Michael R. MacAskill
University of Otago

DataGraph 3.0

Visual Data Tools, Inc., Chapel Hill, North Carolina, USA. USD 90.
http://www.visualdatatools.com/DataGraph/

Introduction

DataGraph is commercial graphing software for Apple's OS X operating system, focussed on creating publication-quality two-dimensional figures. Higher-dimensional datasets are depicted by mapping additional discrete or continuous variables to attributes such as symbol shape, size, or colour, or by juxtaposing across several adjoining plots (Cleveland 1985).

Concept

The developer has described the application as

> "a combination of a data analysis tool, graphing and structured drawing program. . . structured since the position and information depends on the data and will change automatically with the data" (Adalsteinsson 2008).

In practice, this is seen in such ways as:

- Drawing items (such as a text block, an arrowed label, or a magnification inset) have their location defined in terms of data coordinates rather than as an absolute position on the page. If, for example, the axis limits are changed and the plotted data shifts, such items move accordingly, as they are anchored to a location in the data.
- Graphs have a live connection to the data and update immediately in response to changes.
- Using a simple command line script or a graphical OS X **Automator** action, a **DataGraph** file can be used as a template to create a graph from a new external data source.

The options available make it relatively straightforward to craft Edward Tufte-style charts with a minimum of "chart junk" (Tufte 1983) and the software embodies a somewhat purist

approach to data visualisation. For example, the developer has stated that one feature which will not be added to the software is the ability to create pie charts, due to their low data-to-ink ratio (Adalsteinsson 2010).

Licensing, support, and third-party development

The software is proprietary and closed source, created and maintained by a single developer, Dr. David Adalsteinsson. Documents are, however, saved in an open file format with constituent elements stored in standard OS X package elements such as XML property lists. This ensures that the the underlying data will always be available for extraction even in the absence of the **DataGraph** software.

There is an active web-based discussion and support forum (http://www.visualdatatools.com/phpBB2/), with prompt and detailed posts from the developer. The author has on several occasions used it to request a new feature or report a bug, and within hours been notified of a resulting updated beta version. The beta software is available for all users to download. Releases of the official stable version occur several times per year.

A trial version is available for download from the website (the Mac App Store currently distributes only the paid version). The trial provides all of the functionality of the full version but applies a watermark to output. There is no special educational pricing, perhaps as realistically most users will be from the academic world, but relative to comparable commercial products, the price is highly competitive.

A license to use the application also allows a developer to incorporate its framework in their own software (via a Cocoa-based API, http://www.visualdatatools.com/DataGraph/Framework/). There is a small number of developers on the dedicated sub-forum of the discussion website. When such software is distributed, it is fully functional for any user who also has a licence to use **DataGraph**. Otherwise, a watermark is displayed on any graph output. To allow fully independent distribution, a framework licence can be purchased so that downstream users don't also have to own **DataGraph**. For academic applications, this licence costs US$400. Framework licence options are also available by negotiation for distributing either open source or fully commercial applications.

The software is closely entwined with the underlying OS X frameworks such as Cocoa and thus there are no plans to release it for other desktop operating systems. Due to the commonalities between Apple's desktop OS X and mobile iOS frameworks, however, extension to a mobile platform such as the iPad is feasible and apparently under way.

Data import

Import is generally by way of standard text formats such as comma-separated value (CSV) files. Proprietary binary formats which can be read include MATLAB, **Plot**, and **Cricket Graph**. There is currently no support for importing from native R or **Excel** data files, which must be exported via an interim format such as CSV. Variables can naturally also be pasted from the clipboard, with or without a header row.

The software is said to have been tested with large data sets containing millions of rows. The author has personally plotted data containing tens of thousands of observations and found the output to be effectively instantaneous, even on a reasonably dated machine.

There is quite a degree of flexibility in parsing and interpreting date formats. The software can also interpret some strictly non-numeric characters quantitatively. When importing or typing a value such as 180° or $3/2\pi$, the cell retains those literal contents but they are evaluated as required behind the scenes to a decimal quantity. Calculated variables can be created from expressions containing standard mathematical and statistical functions.

Graphics export

Charts can be exported to a range of standard bitmap – PNG (portable network graphics), TIFF (tagged image file format), JPEG (Joint Photographic Expert Group) – and vector – PDF (portable document format), EPS (encapsulated postscript), SVG (scalable vector graphics) – file formats. Charts are copied to the clipboard as PDF data, compatible with most OS X Cocoa-based applications. More unusual is animated output to QuickTime movie format. The user defines an animation variable, such as the date on a time series chart. The data is then 'played' to the screen as the variable steps through its range over a specified duration. Transparency is supported in PNG, vector or movie formats.

Interface and design

The interface is based upon a unified document window, with three panes for the spreadsheet-like data table, the chart itself, and the series of commands used to create that chart (see Figure 1). Additional panes can be displayed as required, such as one showing the list of variable and constant definitions. A floating "scratch pad" can be used to test formulae and expressions, and acts as a clipboard to hold various information.

Graph types

Commercial graphing software titles are often marketed on the basis of how many "types of graphs" they offer. The user is given many options to choose from, although some may differ only in the options applied to a common template rather than by being fundamentally different graph forms. An alternative approach is to provide a smaller core set of graph types, but allow the user flexibility in layering, combining, and customizing them. This is the philosophy of the R package **ggplot2**, for example (Wickham 2009). The downside of this power and flexibility is the learning curve required for less technical users, faced with the entire R programming environment. **DataGraph** strikes a middle ground. It has a polished graphical interface, making it easier to use than R-based solutions, but it lacks the powerful data manipulation features and the huge variety of graphical forms available to R. It does, however, have an extensive capability for the the superposition and juxtaposition of graph types in a way that is unusual for commercial GUI-based graphing software.

The flexibility of the graph commands available in **DataGraph** is illustrated in Figures 2 and 3. Figure 2 shows a simple scatterplot created with the `Points` command. This has been augmented by superimposing two `Fit` commands and two marginal `Histogram` commands. That is, the fit lines and the histograms are not special options within the `Points` command. They are stand-alone entities, which have their own attributes for spatial location and the data they represent. For example, the model fits could be based on data other than that plotted in the main figure (such as a variable with its outliers removed). Whether the fit line

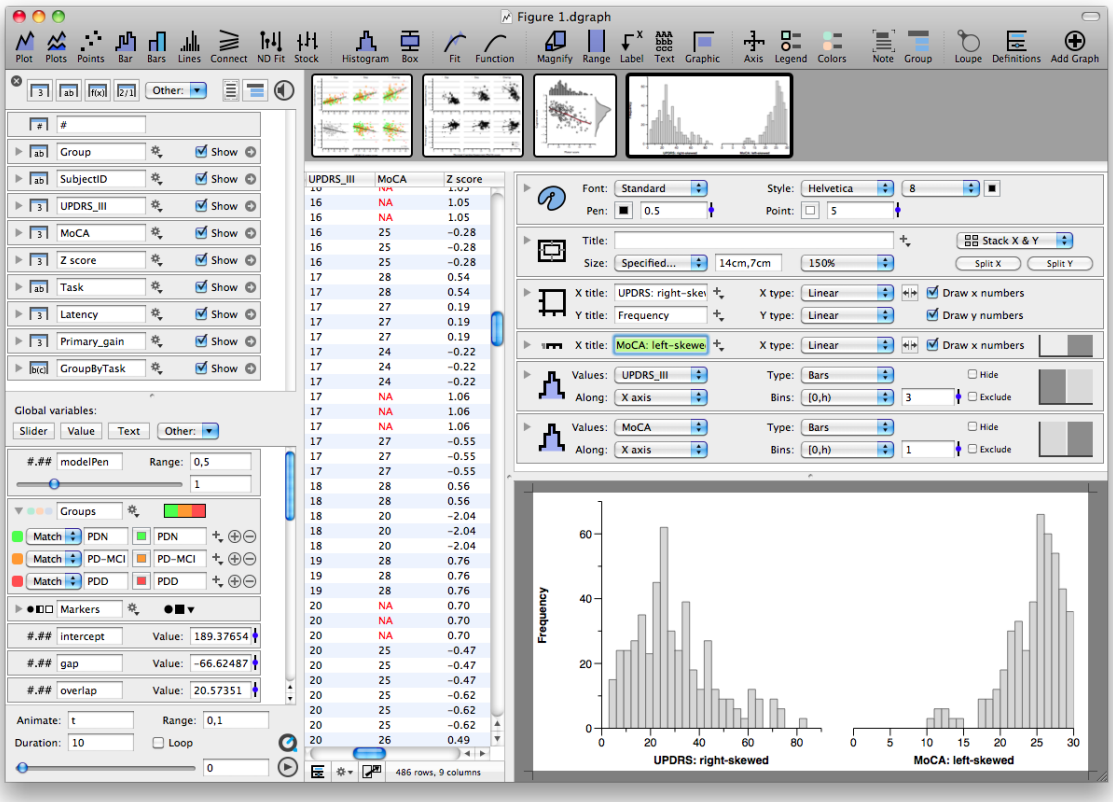


Figure 1: A typical document interface layout (the size, positioning, and visibility of the constituent panes can be altered as required). In this example, we see a chart in the lower right corner, based upon data in the spreadsheet to its left. Above the graph is a stack of boxes representing the settings and commands used to create the chart, each with a disclosure triangle to reveal more detailed options. The top three boxes are present by default, and contain appearance settings for the chart (such as fonts, colours, and sizing). To create this particular figure required adding the three lower commands: one creates an additional x axis, while the remaining pair define each of the two histograms. At the extreme left is a list of the attributes of the variables in the spreadsheet, and a variety of other user-defined constants. The row of thumbnails at the top allows switching between the several charts within this file, all based on data from the same spreadsheet.

is drawn above or behind the data points is controlled simply by dragging the `Fit` entry higher or lower than the `Points` entry in the list of commands. Figure 3 meanwhile, is simpler than Figure 2, consisting only of a `Points` command. With new data and with different options applied, the same command results in a visualisation with a distinctly different form and appearance.

DataGraph includes other canonical graph forms, such as bar and line graphs. The flexible `Connect` command draws curved or straight lines between pairs of (x, y) coordinates, allowing the creation of figures ranging from vector fields to Gantt project management charts. Other commands allow lines and regions to be drawn to delineate areas of a figure. The flexibility of these tools can be used to overcome features which are not currently provided by the software.

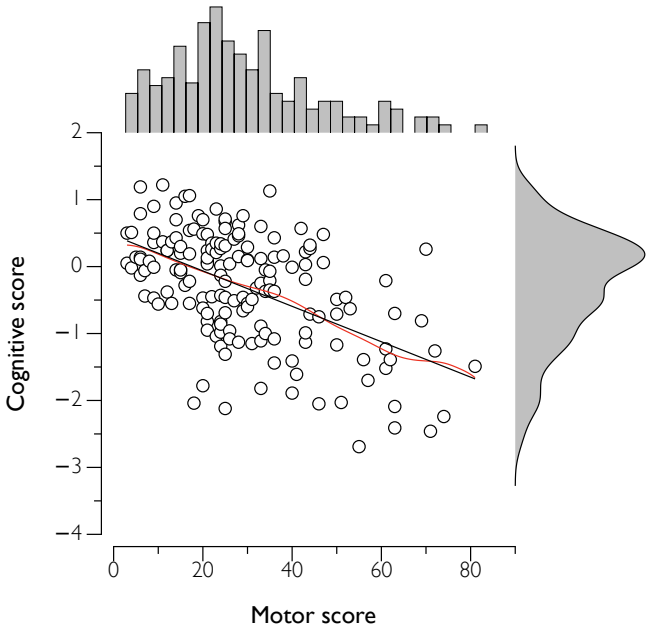


Figure 2: Scatter plot of the relationship between two continuous variables, illustrated by linear regression (black line) and by a LOESS fit (red line). The distributions of the two variables are represented by a marginal histogram (along the x axis) or kernel density estimation (y axis). Arbitrary figures can be created by layering and combining such plot commands.

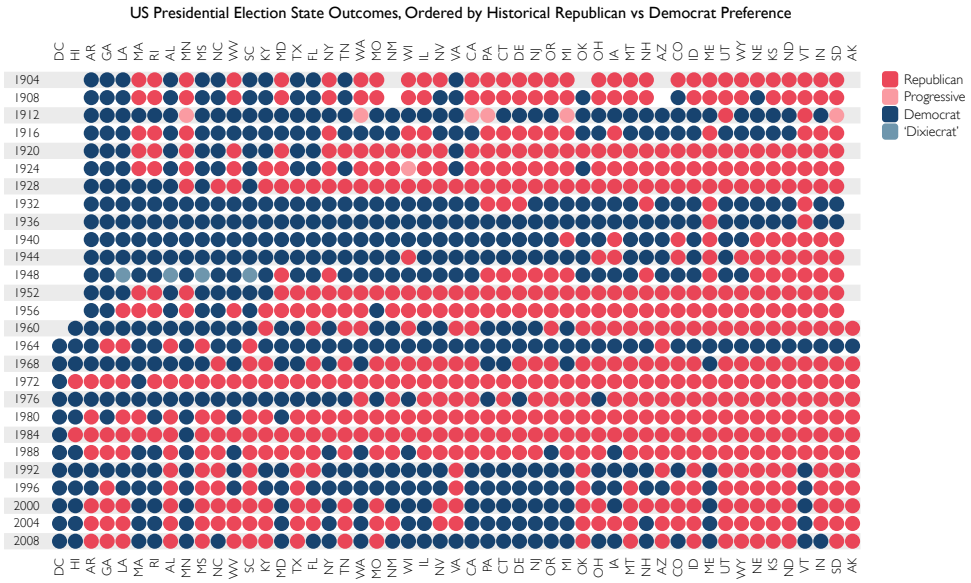
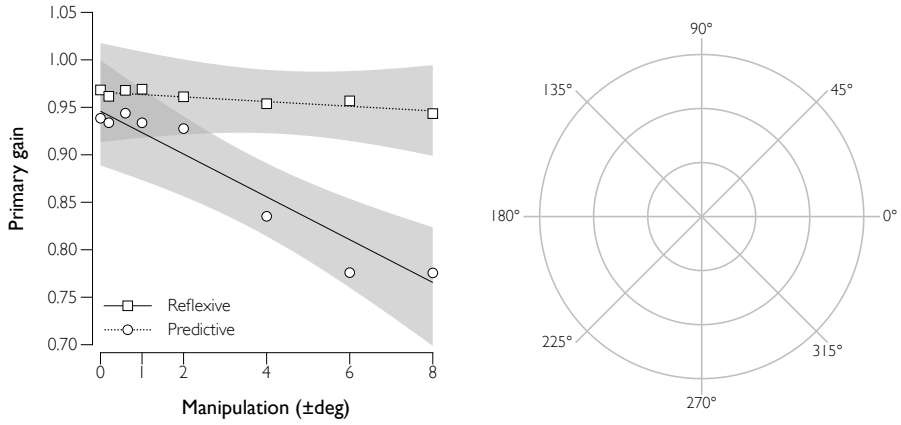


Figure 3: A graph also using the `Points` command which provided the basis for the scatter plot in Figure 2. With a different dataset containing a third variable mapped to colour, and using different display options, the visual form is substantially changed.



(a) Confidence bands around two model fits.

(b) Frame for a polar coordinates plot.

Figure 4: **DataGraph** does not currently provide built-in support for confidence bands around model fits or polar plots. The available commands, however, are sufficiently flexible to allow such features to be constructed manually. In (a), `Region` commands have been used to fill in the zones between invisible lines representing the confidence limits. In (b), graph axes expressed in polar (r, θ) coordinates have been transformed via expressions to Cartesian (x, y) form. The actual underlying Cartesian axes have been hidden, and the polar graph frame drawn using a single `Connect` command for the axial lines and a `Plot` command for each of the concentric circles.

For example, **DataGraph**'s simple curve fitting module does not directly produce confidence or prediction bands around regression lines. Nor does it currently have a polar coordinates plot option. The various drawing and plotting commands are, however, sufficiently flexible that both features can be constructed 'manually' despite not being built-in (see Figure 4). Such features might ideally be incorporated in the core functionality of the software. That they can, however, be implemented by the user, illustrates that one is not limited to the default templates and tools provided by the developer, which is often a constraint with commercial graphing software.

Depicting one-dimensional distributions is well catered for by the `Histogram` and `Boxplot` commands. The histogram offers standard or cumulative frequency, or a smooth kernel density estimate (see Figure 2). An alternative, better suited to simultaneously depicting the distribution of multiple variables, is the `Boxplot` command. This is capable of representing distributions either via a box and whisker plot, as individual jittered data points, or as single or multiple histograms.

Other features

Expression variables can be created by processing other numeric or categorical variables, utilizing a variety of statistical and mathematical functions. One-off expressions can be entered into graphical commands. For example, a text label can include a reference to the mean of a variable, which will update automatically when the underlying data is altered. Arbitrary

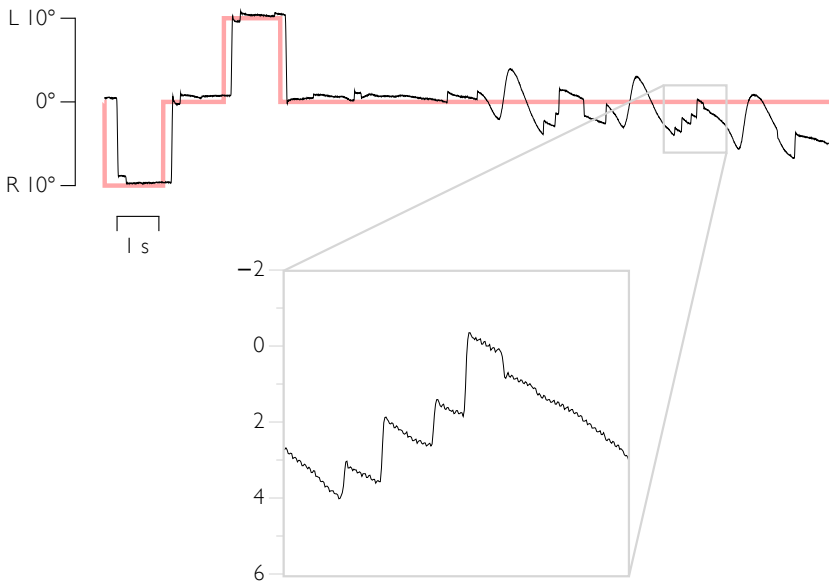


Figure 5: The `Magnify` tool allows a subset of the data to be displayed in more detail. Elements can be defined to appear in the magnified inset, in the background, or in both. For example, here the red line appears only in the main body of the figure. The different axis scales can optionally be shown for each view.

analytical functions can be plotted without needing to have underlying empirical data in the spreadsheet. The user can define parameters which can be varied in real time, including by use of graphical sliders.

Regression fitting includes single and multiple linear regression; polynomial, exponential, power, and LOESS fits; and arbitrary fit functions. Residuals can be viewed, with outliers interactively removed. For more complicated models, a fit equation derived in other software can be drawn.

A 'loupe tool' allows the user to zoom in on the graph smoothly and interactively to inspect the data. Right-clicking on data points reveals their coordinates, which can be copied, or displays the corresponding spreadsheet cells. In final output, a magnification callout allows a section of the data to be highlighted and displayed in detail (see Figure 5).

Target audience

This software is targeted towards the needs of researchers but is not oriented towards a particular field. The only domain-specific form of graph is the 'Stock' command for depicting share prices. Otherwise it caters for the commonest visualisations used to depict distributions and two-dimensional relationships, and should be suited for applications in all quantitative fields.

Advantages and limitations

Being proprietary software, **DataGraph** has the polished interface and attentive customer support that is necessary to continue attracting and retaining customers. This comes with the attendant risks of a closed-source codebase, being developed by a one-person company. Researchers can place considerable effort in creating visualisations, and need to be confident that they will be accessible well into the future. In mitigation, **DataGraph** is already a mature product, based on modern frameworks. If development ceased tomorrow, it would remain fit-for-purpose for some years to come. And unlike most commercial graphing software, users with the requisite development skills can incorporate DataGraph's framework to extend their own projects.

The program is focussed on producing publication-quality graphics. There is no capacity to produce the three-dimensional extrusions added as decorations to the essentially 2D graphs produced in other, more business-oriented programs. There is also no capacity for visualizing data or functions which are truly 3D, unless users create an effective mapping of the third or higher dimension to attributes such as symbol size or colour. The developer has stated that 3D capacity will not be added, as the program's architecture and interface is inherently structured around 2D analysis.

The most common forms of 2D graphs are catered for, with the significant omission of a built-in command to produce non-Cartesian polar plots (although it has been stated by the developer that it may be added in a future release). There is also a paucity of options for advanced depictions of purely categorical data, such as mosaic or parallel coordinate plots.

Increasingly, sophisticated online visualisations are being created that allow the audience to interact with the data. **DataGraph** exports only non-interactive content. This is, however, perhaps a reflection of the lack of an agreed standard for interactive graphics rather than a criticism of this software in particular. To produce online interactive charts currently requires either partnering with expert graphic designers or programmers; attempting oneself to master one of several rapidly-evolving general purpose technologies, such as D3 (Bostock, Ogievetsky, and Heer 2011), or trying to shoe-horn data into the particular form required by effective but highly domain-specific tools, such as Hans Rosling's **GapMinder** (`http://www.gapminder.org/`). There remains an unfilled niche for simple desktop graphing software which could give any user the direct capacity to create an interactive portal to their data. When working within the current constraints of traditional academic publishing formats, however, **DataGraph** will be a useful tool for many Mac-using researchers.

References

Adalsteinsson D (2008). "**DataGraph** Support Forum: EPS Unreadable by Illustrator CS3, Post 4." URL `http://www.visualdatatools.com/phpBB2/viewtopic.php?t=233`.

Adalsteinsson D (2010). "**DataGraph** Support Forum: Pie and Radar Graphs, Post 2." URL `http://www.visualdatatools.com/phpBB2/viewtopic.php?t=654`.

Bostock M, Ogievetsky V, Heer J (2011). "D^3: Data-Driven Documents." *IEEE Transactions on Visualization and Computer Graphics*, **17**(12), 2301–2309.

Cleveland WS (1985). *The Elements of Graphing Data*. Wadsworth, Monterey.

Tufte ER (1983). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire.

Wickham H (2009). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag, New York.

Reviewer:

Michael R. MacAskill
Department of Medicine
University of Otago
and
New Zealand Brain Research Institute
66 Stewart St
Christchurch 8011, New Zealand
E-mail: michael.macaskill@nzbri.org
URL: http://nzbri.org/macaskill

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 47, Software Review 2 *Published:* 2012-05-31
May 2012