

Linked Data Views

By Graham Wills
gwills@research.bell-labs.com

Introduction

I think of a “data view” very generally as anything that gives the user a way of looking at data so as to gain insight and understanding. A data view is usually thought of as a bar chart, scatterplot, or other graphical tool, but I use the term to include a display of the results of a regression analysis, a neural net prediction or a set of descriptive statistics. In a simple case, a scroll bar is a view of a document, linked to a textual representation beside it. Selecting an area in the scroll bar using the thumb links to the associated text view to display new textual information. In general, a data view is a representation the user can look at and study to help understand relationships and determine features of interest in the data they are studying. Typically there are parameters or variations in the method of display so that some way of interacting with the view to modify its behavior is necessary.

Also typical is the desire to explain something of interest found in a view. Do data form two clusters under this particular projection of the grand tour? Is there a change in the relationship between salary and years playing baseball when the latter is greater than five years? When we see something interesting, we want to explain it, usually by considering other data views or by including additional variables and see if they can explain the feature, or indeed if they have any effect whatsoever. In a regression analysis, you can just simply add a variable to the set of explanatory variables (taking due care with respect to multicollinearity and other confounding factors). If a histogram of X shows something of interest, you can “add” a variable Y to it by making a scatterplot of X against Y . If you want to explain something in a scatterplot, then it is possible to turn it into a rotating point cloud in 3D, and using projection pursuit or grand tour techniques, you can go to still higher dimensions.

Despite the undoubted utility of this approach, it does present some problems that prevent it from being a complete solution. The main ones are:

- As plots become increasingly complex, they become harder to interpret. Few people have problems with most one-dimensional plots. Scatterplots, tables and grouped boxplots or other displays involving two dimensions are easily learnable. But the necessity of

spinning and navigating a 3D point cloud, or understanding the contributions to a multivariate projection make these views less intuitive.

- It is harder to accommodate differences in the basic types of the data. High-dimensional projection techniques assume the variables are rational, as do techniques that display multivariate glyphs and, to a large extent, parallel axes techniques. Given a table of two categorical variables, adding a rational variable requires changing to quite a different type of view, such as a trellis display.
- Data that are of a type specific to a particular domain can be impossible to add directly. Exploring relationships in multivariate data collected at geographical locations, on nodes of a graph, or on parts of a text document is very hard because of the difficulty of building views that correlate the statistical element and the structural element of the data. Often, two completely different packages are used for the analysis, with results from one package mangled to fit the input form of the other package – a frustrating task.

A good solution to these problems is the linked data views paradigm. The idea is fairly simple; instead of creating one complex view, instead create several simpler views and link them together so that when the user interacts with one view (for example, to indicate a feature of interest), the other views will update and show the results of such an interaction. This allows the user to use views that require less interpretation and views that are directly aimed at particular combinations of data. It also allows the easy integration of domain-specific views; views of networks or maps can easily be linked to more general-purpose views.

I do not mean to argue that the linked data views is a uniformly superior method to that of monolithic complex views mentioned above. That is not the case, as there are examples where a single multivariate technique is necessary to see a given feature, and multiple simpler views simply won't do. However, for many problems, especially those where conditional distributions are of interest, the linked data views technique works extremely effectively.

Starting with Scatterplots

One of the earliest linked views work to achieve wide attention was the scatterplot brushing technique of Becker, Cleveland and Wilks (1987). By arranging scatterplots of n variables in a table so that all the $n(n-1)$ ordered combinations of axes are present, the eye can quickly scan a row or column and see how a given variable depends on each other variable. This useful arrangement technique is enhanced by the use of a brush. In this

context, a brush is a shape that is dragged around the view by the user, and performs some operation on the graphical elements it passes over. In typical scatterplot brushing tools, the data points brushed over are painted in a different color, both in the panel in which the brush is active, and in all other panels of the window. In our terminology, the brush is the mechanism that links the scatterplot data views.

One of the reasons this technique is so effective is that in each linked view, there is a one-to-one correspondence between cases of the data matrix and graphical representations of these cases, so that in each scatterplot we have complete freedom as to what color or glyph to use to represent this data item. Intuitively, it is easy for us to think of the 'red, square' item, and locate it in each view. The conceptual model (which can easily be the internal data structure, too!) is of adding a few extra columns to the data matrix to represent color, glyph, and visibility and using the brushing technique to modify the values in these columns. The table below shows such a model:

V1	V2	V3	Visible	Color	Glyph
Cork	22.3	5	0	green	circle
Dublin	12.3	5	1	green	circle
Kerry	18.8	6	1	red	circle

Table 1. Sample data (V1, V2, V3), with added variables representing graphical information for display purposes

To manage a brush over a data view, the program must calculate what cases are under the brush and manipulate the values of one or more of the additional graphical variables for each such case. Each linked view must then update to reflect the changes.

Even when restricted to data views that display graphical elements for each case, this is a powerful tool. An example of a successful tool in this area is XGobi (Swayne, Cook and Buja, 1998). XGobi is a X-Windows based tool that presents the user with several glyph-based views (dotplots, scatterplots, rotating plots, grand tour and projection pursuit tours), and uses brushing to link the views along the above lines.

Generalizing the Implementation

The above approach runs into problems with more than small amounts of data. If you have tens of thousands of points, often you want to look at views that aggregate the data, such as bar charts, histograms and frequency tables. Linking them would be useful. Also useful would be a more general method of linking; perhaps we want to link cases with ones in another data matrix, using a user-defined linking function. And, referring back to the opening

paragraph, it would be very helpful to be able to link graphical plots of data to models of the data.

The statistical analysis package DataDesk (Velleman, 1997) was originally built as a teaching tool, but is now a full-featured statistical package that has linked views designed in at the core. Brushing works with aggregated views such as bar charts and histograms as well as within unaggregated views, and the outputs of analyses such as regression and correlation analysis can be visualized and are linked to the other views. If a user spots some unusual cases in a view of residuals from an analysis, they can brush those points, see if there is anything that might explain it in other variables, modify the model and instantly see the residual view update to reflect the new model. Figure 1 shows an example of such linking.

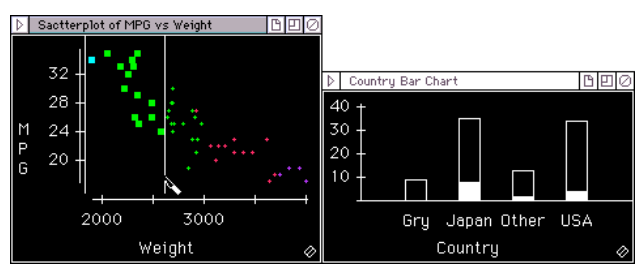


Figure 1. Linked views in DataDesk. Points selected in a scatterplot of Miles per Gallon vs. Weight are highlighted in the bar chart of Country. Selecting the points in the low end of the weight scale shows which country makes the lightest cars.

There are some design decisions that have to be made when linking aggregated views. For example, suppose we wish to link a bar chart and a scatterplot as in figure 1. Looking at table 1, how can we represent the *Visible* attribute consistently in both? Not too hard, items with zero visibility do not appear in the scatterplot and are ignored when creating the bar chart. The *Glyph* attribute is also simple – bar charts cannot use glyphs and that attribute must be ignored. *Color*, however, may be dealt with in several ways. Ignoring the coloring by mapping all the colors to a single neutral color is the method used by DataDesk. Another method, used by EDV (Wills, 1997), is to color portions of the bar in proportion to the number of cases in the bar with each color. This is achieved by dividing the bar up into segments, one for each defined color, with the size of these segments proportional to the number of cases in that bar with the given color. The overall affect is to produce a dynamically changing stacked bar chart.

An alternative method is used by LispStat (Tierney, 1990), in which each data item is assigned its own place in the bar and that section of the bar is colored appropriately. This solution is very close to the 'one-to-one' relationship method in the previous section, as each bar is really a set of stacked rectangles, one for each case. Both drawing and brushing over the bars is handled as if the bars were a

collection of separate little boxes. Figure 2 shows the difference between the three approaches for some sample data.

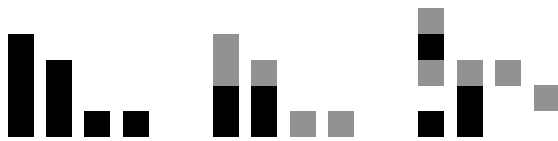


Figure 2. Three methods of linking cases in an aggregated view. Each view shows 4 selected black cases, 4 selected grey cases and 6 unselected cases. In the left bar chart, color is ignored; only the selection state is used. In the middle view, colors are stacked. In the right view, a portion of a bar is allocated to each case, and that portion is colored with the appropriate color.

One of the more powerful novel features of LispStat is due to its implementation in Lisp – as an interpreted language, the user is free to write any function that can link views together, and indeed can write any data view they wish. If you want interactively to color items in a scatterplot based on their distance from the center of the brush, it is an easy job ... as long as you know Lisp.

MANET (Unwin et al., 1996) is a relatively new environment for exploratory data analysis. MANET is an acronym standing for “Missings Are Now Equally Treated”, and this describes an important novel feature of the system; the ability to display information about missing values in views in which they would otherwise appear, and to integrate this information naturally within the view. Thus in a scatterplot of X against Y , cases with missing X values are plotted as a dotplot on the Y axis at locations corresponding to their Y values. This enables the analyst to check for relationships between missing values of X and values of Y . One point to be made here is that the analyst need not do anything special to see if such a pattern exists; it is presented to them as a routine part of the exploration; they are “equally treated”.

Another novel feature of MANET is a technique for dealing with a common problem when analyzing large data sets; screen resolution. A lot of attention has been focused on overplotting for scatterplots and similar views, but MANET is unique in that it addresses *underplotting*. A very common situation is in drawing bars of a histogram or barchart when either an entire bar or the selected section of it has a logical height of a fraction of a pixel. Especially vexing is the case when a bar has a height of one pixel, but only some of its data cases are selected. Both alternatives – drawing the selection and so filling the bar, and drawing nothing at all – give the false and possibly dangerously misleading impression that the bar contains only unselected or only selected items. MANET helps the analyst avoid drawing such a conclusion by drawing a red line under the bar to indicate the presence of such a condition. Figure 3 shows an example of this technique.

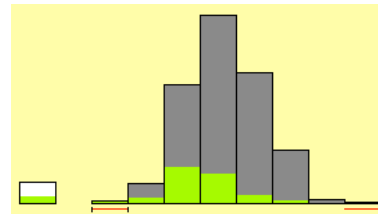


Figure 3. A histogram in MANET. On the left is a bar representing missing values, and under the extremes of the rest of the histogram lines are displayed to indicate that screen resolution may be causing a misleading display.

Specializing the Implementation

Moving in a somewhat opposite direction is research aimed at building views and systems for specific domains. For spatial data, Unwin and Wills (Haslett et al., 1990) built a system that combined a number of statistical views with geographical views. REGARD allowed the user to manipulate maps of geographical information, containing layers of data representing sets of geographical information (towns, rivers, countries, etc.). These different layers contain entities with statistical data, so that user can create data views on one or more of these variables, and use the linking system to tie the views together. The interesting part of the tool, from the point of view of this article, is in the linking between geographical layers. A common scenario is the following: A user has been exploring pollution levels in streams and has selected a group of heavily polluted stream segments. They now want to see the result of that selection in a layer containing regions with population data. In a typical Geographic Information System, the analyst would take the identified stream segments and expand around them to create a set of regions. Then the population regions that intersect this stream buffer region can be identified. In REGARD, this process was generalized by using geographical distance to link layers. This encompasses the case above as well as other common examples like inclusion of points in regions, intersection of regions and points in one layer being selected if close to selected points in another layer.

REGARD also pioneered linking in networks which was further developed in NicheWorks (Wills, 1999). Here the data consist of information on nodes and links of a graph, and the goal is to use the linking mechanism to facilitate exploration of relationships between the two sets of data, as in figure 4. Instead of geographical distance, the concept of distance along the graph becomes the essential factor and the number of useful linking operations is quite large. Depending on the application, selecting a graph node might lead to selecting edges

originating from and/or terminating at the node, nodes down or upstream from the node, all nodes and edges in a connected or doubly-connected component containing the node, nodes and edges n steps away, and so on.

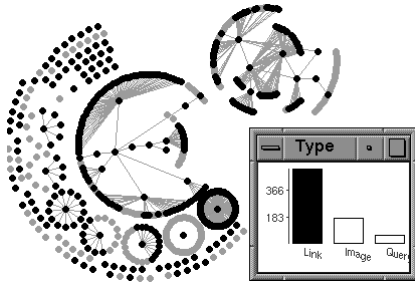


Figure 4. A graph with two components representing sets of URLs (web locations) and their interconnections, linked to a bar chart describing the type of URL. The largest bar, representing "normal" pages (not images or queries/scripts) has been selected.

In a rather different area, strategic computer games are featuring an increasing amount of data view linking. One of the earliest and best implementations is SimCity (Maxis Corporation, 1989), where multiple views of a simulated city continuously change in response to user interactions (and the occasional monster rampage or UFO invasion). SimCity and its descendents feature a complex model, with many variables that are both outputs and, at the next time step, inputs to the model. As a linked views system with millions of users and several versions, it is an excellent example of using simple linked views to understand a complex model and data set. I like the knowledge that my many attempts to build a perfect metropolis have not just been entertaining.

Onwards

There are several existing linked data views environments, each of which has its own contribution to research into visual exploration of data. Whenever I see a new view or technique that I like the look of, I think how it might be added into the environment I use. It's rare that a view cannot be modified to work with others, and the benefits are large; a new view need only focus on what it does best – its own neat or novel feature, relying on the existing views in the system to do their job. The whole is then greater than the sum of the parts.

In the reference and resources sections below, I have not tried to cite the seminal papers or earliest occurrences; I have instead endeavored to look for references that provide a good overview and introduction of the authors' variation on this powerful and important technique. The linked views paradigm is a vital and expanding part of statistical graphics research, and I have no doubt that I've missed exciting new developments and novel techniques

in this brief survey. Drop me a line and let me know about them!

Web Resources

<http://stat.umn.edu/~luke/xls/xlsinfo/xlsinfo.html>

Lisp-Stat program and documentation.

<http://www.bell-labs.com/~gwillis>

A base page from which to access an introduction to the EDV environment and the NicheWorks graph tool

<http://www.datadesk.com/>

Home for both DataDesk and ActivStats (a teaching tool featuring DataDesk for analysis)

<http://www.mathsoft.com/splus/>

Much of the original work on scatterplot brushing was done in the S environment. This is a link to the current commercial version, S-plus.

<http://www.research.att.com/~andreas/xgobi/>

XGobi program and XGVis program (a version of XGobi for graphs and multi-dimensional scaling).

<http://www.simcity.com/home.shtml>

Maxis' site for SimCity

<http://www1.math.uni-augsburg.de/Manet/>

A guide to the MANET system

References

Becker, R.A., Cleveland, W.S. and Wilks, A.R. (1987) "Dynamic Graphics for Data Analysis", *Statistical Science* 2; pp. 355-395

Haslett, J., Wills, G. and Unwin, A. (1990) "SPIDER – An Interactive Statistical Tool for the Analysis of Spatial Data" *Int. Journal of Geographical Information Systems* 4 #3; pp. 285-296

Maxis Corporation (1989) "SimCity [computer program]", Walnut Creek, California

Swayne, D. F., Cook, D. and Buja, A. (1998) "XGobi: Interactive Dynamic Data Visualization in the X-Window System" *Journal of Computational and Graphical Statistics* 7(1) 1998

Tierney (1990) *Lisp Stat: An Object Oriented Environment for Statistical Computing and Dynamic Graphics*, Wiley

(Unwin, A., Hawkins, G., Hofmann, H., and Siegl, B. (1996) "Interactive Graphics for Data Sets with Missing Values - MANET" *Journal of Computational and Graphical Statistics* 5(2) pp. 113-122

Velleman, P.F. (1997) *Learning Data Analysis with DataDesk, Student Version 5.0*. Addison Wesley

Wills (1997) "How to Say 'This is Interesting'" *Proceedings of Section of Stat. Graphics 1997*; pp.25-31 American Statistical Association

Wills, G. (1999) "NicheWorks – Interactive Visualization of Very Large Graphs" *Journal of Computational and Graphical Statistics*, to appear June 1999