R chart types

As this interface is designed to allow “trellis” style graphics, it is only appropriate to use functions generating graphics that can be drawn in this way. The R “lattice” library provides these functions. The library is loaded with the command “library(lattice)”. The functions available are:

- cloud (3 dimensional view, data shown as ticks)
- xyplot (2 dimensional view, data shown as ticks)
- dotplot (1 or 2 dimension view, data shown as ticks)
- barchart (barchart of quantitative variable across an ordinal variable)
- histogram (histogram of a single quantitative field, requires number of bins [use “nint=”])
- densityplot (density plot of a single quantitative field)
- levelplot (“heatmap” composed of a grid of squares with quantitative x and y axes)

For certain special types of data, it is possible to use these functions also:
contourplot, wireframe, bwplot, stripplot, qq, splom, and paralell.
These are more advanced facilities of R, and will not be available through this system.

Additional parameters also provide functionality for some but not all of these functions -

- colouring of graph bars/ticks by an ordinal field (for example, colour all cars red and all lorries blue)
- displaying two quantitative fields on the same axis (for example, leaf length and leaf width) using different colours
- “conditioning” - showing a set of small graphs, each one representing points matching one value of an ordinal field

Areas where the ability of R to plot graphs does not match the facilities offered by Polaris:

- Plots only allow a maximum of one ordinal (categorical) value; it is impossible to have an “ordinal-ordinal” plot.
- Levelplot can only colour by a quantitative variable; also, it is impossible to plot a levelplot with only an x axis and no y axis
- There is no way to imitate the + operator from the Polaris algebra (putting two different x axes next to each other, with a shared y axis, or vice vesa).

However, the system is much more flexible than Polaris when conditioning is used, and will draw the multiple conditioned graphics in any grid arrangement . By default, it arranges them from left to right, and top to bottom, in order to fill the most space on the page possible.

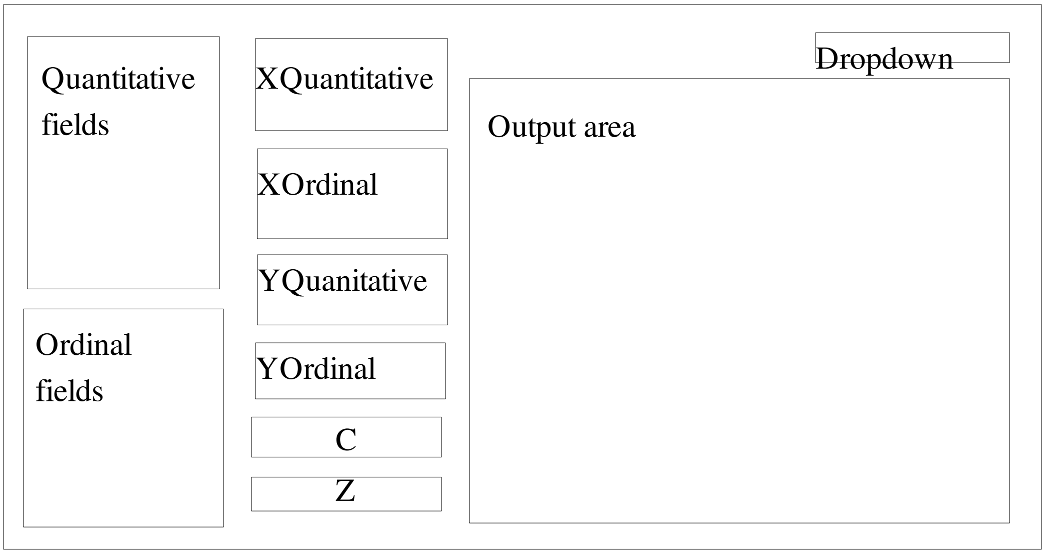
Proposed interface

The interface has six lists, and two slots, and a dropdown.

Lists: OrdinalFields, QuantitativeFields, Xquantitative, Xordinal, Yquantitative, Yordinal

Slots: Z = single quantitative field, C = single ordinal field

When the dataset is chosen, the OrdinalFields and QuantitativeFields lists are populated with fields available to the user. The user drags fields into the other lists or the slots in order to specify the graphic. Only ordinal fields can be dropped in the ordinal lists and slot, and only quantitative in the quantitative lists and slot. If there are multiple ways to draw the graphic, the user can choose between the posibilities using the dropdown.



How to process the input

Work out conditioning for each axis:

if length(Xquantitative) > 0 then Xconditioning = Xordinal
else Xconditioning = rest(Xordinal), Xordinal = first(Xordinal)
if length(Yquantitative) > 0 then Yconditioning = Yordinal
else Yconditioning = rest(Yordinal), Yordinal = first(Yordinal)

Work out the conditioning part of the expression

Conditioning = “”
if length(Xconditioning) > 0 or length(Yconditioning) > 0
Conditioning = “| x1 * x2 * ... * y1 * y2 * y3, layout = c(length(unique(x1))*length(unique(x1))..., length(unique(y1))*length(unique(y2))*...)”
(where x1 = first element of Xconditioning, etc)

Work out the colouring part of the expression

Colouring = “”
if (C not null and length(Xquantitative) < 2 and length(YQuantitative) < 2, Colouring = “,groups = C, auto.key=TRUE”

Now look up from the table the expression to put these into:

		x axis			
		X_{ordinal} null, $X_{\text{quantitative}}$ null	X_{ordinal} not null	length($X_{\text{quantitative}}$) = 1	length(Xquantitative) > 1

y axis	Y_{ordinal} null, $Y_{\text{quantitative}}$ null	(nothing to plot)	(no charts of this type)	densityplot($X_{\text{quantitative}}$ conditioning colouring)
				histogram($X_{\text{quantitative}}$ conditioning colouring)
				dotplot($X_{\text{quantitative}}$ conditioning colouring)
					dotplot(~ $X_{\text{quantitative}}$ conditioning colouring)
					densityplot(~ $X_{\text{quantitative}}$ conditioning colouring)

	Y_{ordinal} not null	barchart(Y_{ordinal} conditioning colouring)
		dotplot(Y_{ordinal} conditioning colouring)
			(no charts of this type)	dotplot(Y_{ordinal} ~ $X_{\text{quantitative}}$ conditioning colouring)	dotplot(Y_{ordinal} ~ $X_{\text{quantitative}}$ conditioning colouring)

	length($Y_{\text{quantitative}}$) = 1	(no charts of this type)	dotplot($Y_{\text{quantitative}}$ ~ X_{ordinal} conditioning colouring)	if Z not null -
				cloud($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ * Z conditioning colouring)
				levelplot($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ * Z conditioning)
				else -
				xyplot($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ conditioning colouring)
					if Z not null -
					cloud($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ * Z conditioning colouring)
					levelplot($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ * Z conditioning)
					else -
					xyplot($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ conditioning colouring)

	length(Yquantitative) > 1	(no charts of this type)	dotplot($Y_{\text{quantitative}}$ ~ X_{ordinal} conditioning colouring)	if Z not null -
				cloud($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ * Z conditioning colouring)
				levelplot($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ * Z conditioning)
				else -
				xyplot($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ conditioning colouring)
					if Z not null -
					cloud($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ * Z conditioning colouring)
					levelplot($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ * Z conditioning)
					else -
					xyplot($Y_{\text{quantitative}}$ ~ $X_{\text{quantitative}}$ conditioning colouring)

Where a table cell contains multiple possiblities, the user can choose between them with the dropdown. The chosen function is sent to R and the result displayed in the output area.