

# **lattice** and **grid**

Paul Murrell

April 5, 2023

The **lattice** package is built on top of **grid** and provides a quite sophisticated example of writing high-level plotting functions using **grid**. Because **lattice** consists of **grid** calls, it is possible to both add **grid** output to **lattice** output, and **lattice** output to **grid** output.

```
> library(grid)
```

## Adding **grid** to **lattice**

Panel functions in **lattice** can include **grid** calls. The following example adds a horizontal line at 0 to a standard `xyplot` (see Figure 1):

```
> xyplot(y ~ x | g, panel = function(x, y) {  
+   panel.xyplot(x, y);  
+   grid.lines(unit(c(0, 1), "npc"), unit(0, "native"),  
+             gp = gpar(col = "grey"))  
+ })
```

The following example writes a left-justified label in each strip (see Figure 2):

```
> xyplot(y ~ x | g, strip = function(which.given, which.panel, ...) {  
+   grid.rect()  
+   grid.text(paste("Variable ", which.given, ": Level ",  
+                 which.panel[which.given], sep = ""),  
+            unit(1, "mm"), .5, just = "left")  
+ })
```

## Adding **lattice** to **grid**

It is also possible to use a **lattice** plot as an element of a **grid** image. The following example splits up the page so that there is an `xyplot` beside a panel of text (see Figure 3). First of all, the **lattice** plot is created, but not drawn. **grid** is used to create some regions and the **lattice** plot is drawn into one of those regions.

```
> someText <- paste("A panel of text", "produced using", "raw grid code",  
+                  "that could be used", "to describe",
```

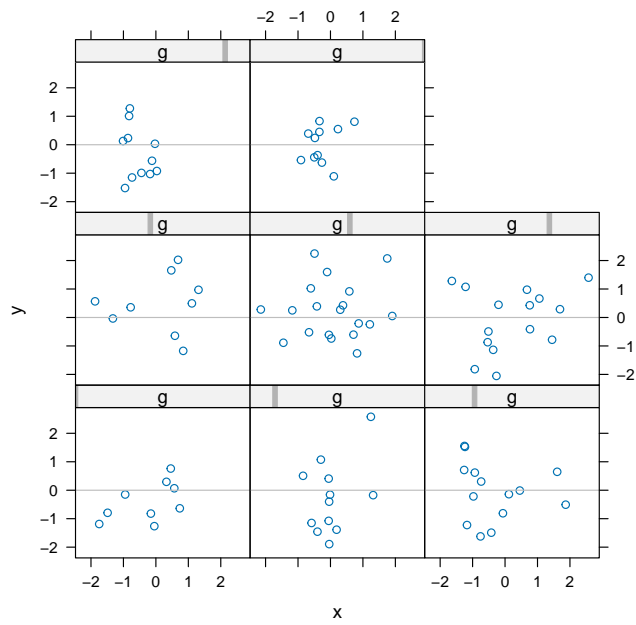


Figure 1: A **lattice** panel function using **grid**.

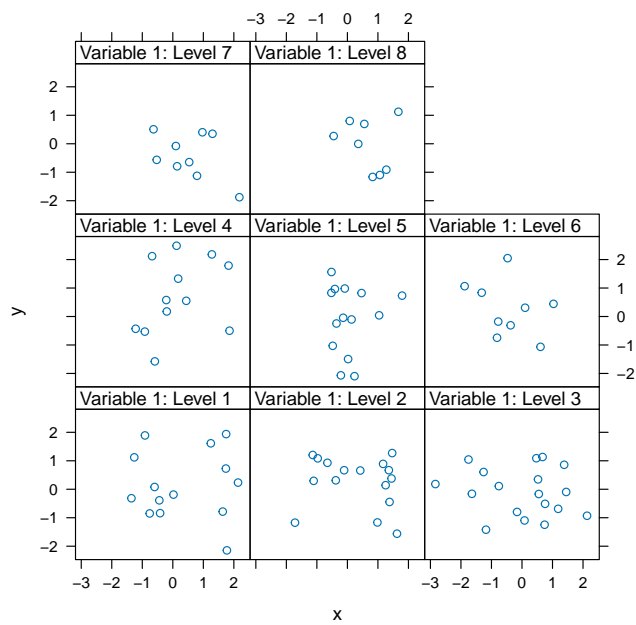


Figure 2: A **lattice** strip function using **grid**.

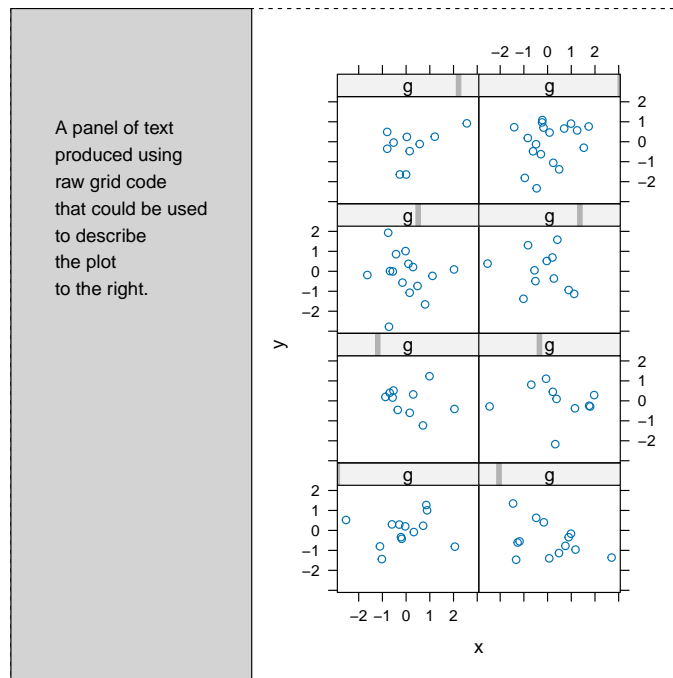


Figure 3: A **lattice** plot used as a component of a larger **grid** image.

```

+           "the plot", "to the right.", sep = "\n")
> latticePlot <- xyplot(y ~ x | g, layout = c(2, 4))
> grid.rect(gp = gpar(lty = "dashed"))
> pushViewport(viewport(layout = grid.layout(1, 2,
+           widths = unit.c(unit(1, "strwidth", someText) +
+           unit(2, "cm"),
+           unit(1, "null")))))
> pushViewport(viewport(layout.pos.col = 1))
> grid.rect(gp = gpar(fill = "light grey"))
> grid.text(someText,
+           x = unit(1, "cm"), y = unit(1, "npc") - unit(1, "inches"),
+           just = c("left", "top"))
> popViewport()
> pushViewport(viewport(layout.pos.col = 2))
> print(latticePlot, newpage = FALSE)
> popViewport(2)

```