General-Purpose Diagrams

Paul Murrell

July 9, 2003

grid provides a lot of general support for arranging graphical elements. The main purpose for this is of course to allow for the production of flexible statistical graphics. However, these capabilities are also very useful for producing general diagrams.

A very simple example is the problem of dividing a page into 10 “lines” (I needed to do this to provide lined paper for my son to begin practicing printing).

The following code uses a grid layout to divide the page into 10 equally-sized regions, then draws a horizontal line at the bottom of each region. It’s pretty basic stuff, but something like this is very natural in grid and surprisingly painful in other graphical software.

```r
> push.viewport(viewport(layout = grid.layout(10, 1)))
> for (i in 1:9) {
+   push.viewport(viewport(layout.pos.row = i))
+   grid.lines(x = c(0, 1), y = 0)
+   pop.viewport()
+ }
> pop.viewport()
```

```
_____________________
_____________________
_____________________
_____________________
_____________________
_____________________
_____________________
_____________________
_____________________
```

1
A slightly more complex example involves creating a calendar. Here we make use of both "lines" and "null" units in a grid layout to fix the size of the heading row, and "mm" units to position the date labels within each day cell.

I have used the date/time functions in base to generate a series of dates (from Sunday Nov 3 2002, for 4 weeks).

```r
> dates <- seq(ISOdate(2002, 11, 2), by = "day", length = 4 * 7)
> day <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
+          "Friday", "Saturday")
```

Now we generate a grid, draw the heading in a loop first, then do each of the days in another loop.

```r
> push.viewport(viewport(layout = grid.layout(5, 7, heights = unit(rep(1, + 5), c("lines", rep("null", 4))))))
> for (i in 1:7) {
+     push.viewport(viewport(layout.pos.row = 1, layout.pos.col = i))
+     grid.rect()
+     grid.text(day[i], gp = gpar(fontface = "bold"))
+     pop.viewport()
+ }

> for (i in 1:7) {
+     for (j in 1:4) {
+         index <- (j - 1) * 7 + i
+         push.viewport(viewport(layout.pos.row = j + 1, layout.pos.col = i, +                   ))
+         grid.rect()
+         grid.text(paste(months(dates, abbrev = TRUE)[index],
+                         as.POSIXlt(dates)$mday[index]), x = unit(1, "mm"),
+                     y = unit(1, "npc") - unit(1, "mm"), just = c("left",
+                                "top"), gp = gpar(fontsize = 8))
+         pop.viewport()
+     }
+ }
> pop.viewport()
```
<table>
<thead>
<tr>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nov 3</td>
<td>Nov 4</td>
<td>Nov 5</td>
<td>Nov 6</td>
<td>Nov 7</td>
<td>Nov 8</td>
<td>Nov 9</td>
</tr>
<tr>
<td>Nov 10</td>
<td>Nov 11</td>
<td>Nov 12</td>
<td>Nov 13</td>
<td>Nov 14</td>
<td>Nov 15</td>
<td>Nov 16</td>
</tr>
<tr>
<td>Nov 17</td>
<td>Nov 18</td>
<td>Nov 19</td>
<td>Nov 20</td>
<td>Nov 21</td>
<td>Nov 22</td>
<td>Nov 23</td>
</tr>
<tr>
<td>Nov 24</td>
<td>Nov 25</td>
<td>Nov 26</td>
<td>Nov 27</td>
<td>Nov 28</td>
<td>Nov 29</td>
<td>Nov 30</td>
</tr>
</tbody>
</table>