Examples for `ltxsparklines` package

Boris Veytsman

January 2, 2017

Contents

1 Setting up

The package *ltxsparklines* uses LaTeX package *sparklines* [Löffler and Luecking, 2014] for drawing sparklines in TeX documents: probably prepared with *Sweave* or *knitr* (but you are, of course, may choose other methods to generate TeX documents). Accordingly, many parameters that determine how do sparklines look are defined in the TeX file rather than in *R*.

Here are some global parameters with the values chosen for this document:

```
% The height of the sparklines in ex units
\renewcommand\sparklineheight{1.75}
% The line width
\setlength\sparklinethickness{0.4pt}
% The color of the sparkline
\definecolor{sparklinecolor}{named}{blue}
% The color of the sparkine rectangle when present
\definecolor{sparkrectanglecolor}{gray}{0.8}
% The dot width
\setlength\sparkdotwidth{2pt}
% The color of the spikes
\definecolor{sparkspikecolor}{named}{red}
% The color of the bottom line when present
\definecolor{bottomlinecolor}{gray}{0.2}
```

```
% The thickness of the bottom line
\setlength\sparkbottomlinethickness{.2pt}
% The clipping separation (need sparklines v1.7 or later)
\setlength\sparklineclipsep{1pt}
```

Now we call the library and setting the options.

```
> library(ltxsparklines)
```

The most important option is `ltxsparklines.output`. The package produces a string to be included in the TEX file. Being a TEX command, the string contains many backslashes. Unfortunately the ways *knitr* and *Sweave* treat backslashes are different. Moreover, *Sweave* uses different techniques in inline expressions (those within `\Sexpr`) and chunks created with `results=tex` option. To make the long story short,

1. Use `output='inlineSweave'` for *Sweave* `\Sexpr`.
2. Use `output='knitr'` everywhere else, *including Sweave chunks!*

This document was created in *Sweave*, since the line

```
> options(ltxsparklines.output='inlineSweave') # Comment out
>                                              # if using knitr
```

The other options are mostly self-explanatory; try `help(ltxsparklines)` for more details:

```
> options(
+     ltxsparklines.width = 10,
+     ltxsparklines.clip = FALSE,
+     ltxsparklines.na.rm = TRUE,
+     ltxsparklines.bottomline = FALSE,
+     ltxsparklines.bottomlinex = c(NA, NA),
+     ltxsparklines.startdotcolor = NA,
+     ltxsparklines.enddotcolor = NA,
+     ltxsparklines.dotcolor='blue'
+     )
```

2 Some simple examples

The package can take quite versatile input data. You can give it two vectors, one for x, another for y coordinates of the sparkline: `\Sexpr{sparkline(x=c(1,4,8,10), y=c(5,6,12,3), enddotcolor='red')}` 

If you use `xspikes` and `yspikes` instead, you will have a small bar chart (or *spike chart*): `\Sexpr{sparkline(xspikes=c(1,4,8,10), yspikes=c(5,6,12,3), width=3)}` .

You can even make a sort of a scatter plot with dots: `\Sexpr{sparkline(xdots =c(1,4,8,10), ydots=c(5,6,12,3), dotcolor=c('green', 'red'))}` In most cases, however, you probably want to combine dots and lines: `\Sexpr{sparkline(x =c(1,4,8,10), y=c(5,6,12,3), xdots=c(1,4,8,10), ydots=c(5,6,12,3), dotcolor=c('green', 'red'))}`

Instead of a pair of vectors you can use a matrix; in this case x and y are assumed to be the *columns*: `\Sexpr{mat <- matrix(c(1, 2, 3, 4, 5, 17, 10, 12, 11, 10), ncol=2, byrow=F); sparkline(mat, rectangle=c(10,16), startdotcolor='blue')}`. You can also try a data frame. Here is the famous *cars* data set: `\Sexpr{sparkline(yspikes=cars, width=20)}` This plot shows stopping distance vs. speed; to reverse the axis do just `\Sexpr{sparkline(yspikes=cars[,c(2,1)], width=20)}`

If you have just one vector, it is assumed to be the y coordinates, and the x coordinates are 1, 2, ...: `\Sexpr{sparkline(c(1,8,-5,10), startdotcolor='green', bottomline=TRUE)}`.

If you have a time series, it will be recognized as well: here are the measurements of the annual flow of the river Nile at Aswan (formerly 'Assuan'), 1871-1970: `\Sexpr{sparkline(Nile, width=30)}`.

The parameter `na.rm` selects whether to remove the `NA` values and connect the data over them: compare `\Sexpr{sparkline(c(3,5,4,NA,12,9), na.rm=TRUE)}`, and the similar command with `na.rm` set to `FALSE` `\Sexpr{sparkline(c(3,5,4,NA,12,9), na.rm=FALSE)}`.

3 Scaling and clipping

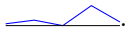
When producing several sparklines, one may want to make them comparable. This means that theis x or y ranges better be the same. The options `xlim` and `ylim` can be used to set them. Note that each of these options has the form `c(min, max)`; if either limit is `NA`, the value is taken from the data. Compare, for example, `\Sexpr{sparkline(c(2,4,8,12,16))}`, `\Sexpr{sparkline(c(2,4,8,12,16), ylim=c(NA,20))}`, and `\Sexpr{sparkline(c(2,4,8,12,16), ylim=c(-10,20))}`.

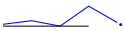
Note that `xlim` and `ylim` just change the scaling of the data, mapping it to the range of the the sparkline area. This means that data outside `xlim` or `ylim` is still plotted, but will overlay the text around like this: `\Sexpr{sparkline(c(2,20,1,16,4), ylim=c(NA,10), xlim=c(2,5))}`. Starting with the version v1.7 the LaTeX *sparklines* package provides the option of clipping. You can use it setting `clip` to `TRUE`, either in the command itself, or globally with `options(ltxsparklines.clip=TRUE`. This produces the following result: `\Sexpr{sparkline(c(2,20,1,16,4), ylim=c(NA,10), xlim=c(2,5), clip=TRUE)}`. Note that clipping is done in LaTeX rather than in *R*, so the closeness of the clipping is define by

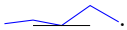
the LaTeX length `\sparklineclipsep`.

4 Additional features

Sometimes you may want to draw a horizontal rectangle showing, for example, interquartile range or 95% interval for the data. The argument `rectangle=c(min, max)` can be used for this: `\Sexpr{sparkline(Nile, rectangle=quantile(Nile, c(0.025, 0.975)), width=30)}`  .

You can show the x axis with the argument `bottomline=TRUE`: `\Sexpr{sparkline(c(2,4,1,12,3), bottomline=TRUE)}`  .

The argument `bottomlinelength` gives the length of the bottom line (in data coordinates) (by default, the line is drawn across the full rectangle): `\Sexpr{sparkline(c(2,4,1,12,3), bottomline=TRUE, bottomlinelength=4)}`  .

Starting with version 1.7 of the LaTeX *sparklines* package there is an alternative mechanism for extensible bottom lines. The argument `bottomlinex`, accordingly, has two arguments, with the start and the end of a bottom line. If both are not `NA`, the bottom line will be drawn: `\Sexpr{sparkline(c(2,4,1,12,3), bottomlinex=c(2,4))}`  .

5 A more complex example

The Nile data in the previous section show a very long graph. It is interesting to look how did the flow change from a decade to decade. To do this we will make a table with decades in one column and sparklines in another. To compare the data, we will use the same vertical limits and plot a rectangle showing the span between 25% and 75% of the total data (the interquartile range). The table is completely constructed in an *R* chunk.

There are differences between the ways such chunks are constructed in *Sweave* and *knitr*: the former uses `<<results=tex>>` chevron, while the latter uses `<<results='asis'>>` chevron. You may use the same chevron if you define `tex <- 'asis'`, but *knitr* will still issue warnings. In both cases you need to set the output to `'knitr'` to get the correct output.

The construction is rather involved, so we do it step by step.

We create the vector `decadeStarts` with first year of the decade, and store the range and interquartile range of the total:

```
> decadeStarts <- seq(min(time(Nile)), max(time(Nile)),
+                     by=10)
> rangeNile <- range(Nile)
> iqrNile <- quantile(Nile, c(0.25, 0.75))
```

The function `printDecade` outputs a line like `1871--1880 & SPARKLINE\\`:

```
> printDecade <- function (start) {
+     end <- start+9
+     paste0(start, "--", end, " & ",
+            sparkline(window(Nile, start, end),
+                      width=20,
+                      xlim=c(start, end),
+                      ylim=rangeNile,
+                      rectangle=iqrNile,
+                      output='knitr'),
+            "\\\\")
+ }
```

Now we can print the table

```
> # Table header
> result <- paste("\\begin{tabular}{ll}",
+                 "\\toprule",
+                 "Decade & Annual flow\\\\",
+                 "\\midrule",
+                 sep="\n")
> # Table body
> result <- paste(result,
+                 paste(sapply(decadeStarts, printDecade),
+                       collapse="\n"),
+                 sep="\n")
> # Table footer
> result <- paste(result,
+                 "\\bottomrule",
+                 "\\end{tabular}",
+                 sep="\n")
> # Printing table
> cat(result)
```

Decade	Annual flow
1871–1880	
1881–1890	
1891–1900	
1901–1910	
1911–1920	
1921–1930	
1931–1940	
1941–1950	
1951–1960	
1961–1970	

References

Andreas Löffler and Dan Luecking. *Sparklines*, October 2014. URL http://www.ctan.org/pkg/sparklines.