Making ltxsparklines: The journey of a CTAN contributor into the world of CRAN

Boris Veytsman

Edward Tufte defines a sparkline as [...] a small intense, simple, word-sized graphic with typographic resolution. Sparklines mean that graphics are no longer cartoonish special occasions with captions and boxes, but rather sparkline graphics can be everywhere a word or number can be: embedded in a sentence, table, headline, map, spreadsheet, graphic. Data graphics should have the resolution of typography [5].

For example, to convey an idea of the evolution over time of TUG membership, we can just insert in the text a simple graph. This graph, better than many words, describes the growth of membership in the pre-Internet era, when the only way to get TeX was to join TUG — and the relative stability for many years now.

Similarly, the famous data set about the flow of Nile at Aswan can be described by a simple word-like graph with the gray line showing the interquartile range. The “resolution of typography” mentioned by Tufte is the natural realm of TeX. Naturally there is a \LaTeX{} package \texttt{sparklines} [2] implementing these small graphs. In this package, the code producing a typical sparkline looks like this:

\begin{sparkline}{10}
\sparkrectangle 0.3 0.8
\sparkdot 0.5 0.62 gray
\sparkdot 1 0.2 black
\spark 0 0 0.1 0.95 0.2 0.8 0.3
0.3 0.4 0.5 0.5 0.62 0.6
0.7 0.7 0.5 0.8 0.4 0.9
0.25 1 0.2 /
\end{sparkline}

While the package is quite versatile, and can make many different kinds of sparklines, it leaves all calculations to the user. It would be much more convenient to plot data using a simple command. While it is possible to write a \TeX{}-only interface for this package from a data processing software also written in pure \TeX{}, for example, datatool [4], in this paper we describe a different approach based on \texttt{R} [3].

Many of us are quite familiar with \texttt{R}. Some, like me, learned to love it after the brilliant lecture by Uwe Ziegenhagen at TUG 2010 [8]. An \texttt{R}-using \TeX{}nician spends most of her time editing and compiling .\texttt{Rnw} files. These files look like \LaTeX{}, but contain “chunks” of \texttt{R} code. Processed by \texttt{R}, these files become .\texttt{tex} files, with \texttt{R} code substituted for the calculation results, figures, etc. The result is a typeset document containing the full report of the research project.

Thus it was a natural decision for me to write an \texttt{R} interface to \LaTeX{} \texttt{sparklines} package. This interface is released as the \texttt{R} package \texttt{ltxsparklines} [6].

The package defines a single \texttt{R} command, named, naturally enough, \texttt{sparkline}. It outputs a \LaTeX{} sparkline environment. The command has a number of possible arguments. Below these arguments are listed with the default values:

\begin{verbatim}
sparkline(x = NULL, y = NULL, xspikes = NULL, yspikes = NULL,
xdots = NULL, ydots = NULL, dotcolor = NULL,
width =
  getOption("ltxsparklines.width"),
rectangle =
  c(NA, NA),
xlim = c(NA, NA),
ylim = c(NA, NA),
clip =
  getOption("ltxsparklines.clip"),
na.rm =
  getOption("ltxsparklines.na.rm"),
bottomline =
  getOption("ltxsparklines.bottomline"),
bottomlinelength =
  NA,
bottomlinex =
  getOption("ltxsparklines.bottomlinex"),
startdotcolor =
  getOption("ltxsparklines.startdotcolor"),
enddotcolor =
  getOption("ltxsparklines.enddotcolor"),
output =
  getOption('ltxsparklines.output'))
\end{verbatim}

The options are fully documented in the package itself. Here we describe the general idea.

Three kinds of sparklines can be created: lines (defined by \texttt{x} and \texttt{y}), bars (defined by \texttt{xspikes} and \texttt{yspikes}), and dots (defined by \texttt{xdots} and \texttt{ydots}). It is possible to combine them, for example, \texttt{\cdot \cdot \cdot \cdot}. One can set both \texttt{x} and \texttt{y} coordinates, or just \texttt{y} coordinates. In the latter case the sequence 1, 2,... will be used for the missing \texttt{x} coordinates. Alternatively the command can be given a time series as an argument, and in this case the time/date values will be used for \texttt{x} coordinates, and the value of the series for \texttt{y} coordinates.

Other arguments can change the appearance of the graphics. Compare, for example, the result of the call \texttt{sparkline(c(2, 20, 1, 16, 4)},
ylim=c(0, 8), xlim=c(2, 5)):
and the similar call clipping the output:
sparkline(c(2, 20, 1, 16, 4), ylim=c(0, 8),
xlim=c(2, 5), clip=TRUE):

There are several color-related options, so one
can create bright and distinct sparklines.
With this package the sparklines in the begin-
ning of this paper were typeset as simple as

# TUG membership
sparkline(xspikes=tug$Date,
yspikes=tug$Members,
ylim=c(0,NA))
# Nile flow
sparkline(Nile,
rectangle=quantile(Nile,
c(0.25, 0.75)),
enddotcolor='black',
width=20)

Here tug is a data frame, obtained by reading
the comma separated file tug.csv using R function
read.csv, and Nile is the time series included in
the standard R distribution.

There are two ways to generate .tex files from
R: the more traditional Sweave package [1,8] and the
newer and spiffier knitr [7]. (A review of the recent
book on knitr was published in TUGboat 35:1: tug.
org/books/reviews/tb109reviews-xie.html.) As
might be expected, the ltxsparklines package works
with both.

This is my first contribution to CRAN, The Com-
prehensive R Archive Network, and it was interesting
for me to observe the differences between CRAN and
CTAN. (By the way, CTAN was the first archive
network of this kind, and other projects like CRAN
and CPAN (for Perl developers) used our site as an
inspiration.)

It looks as if CRAN has a more strict editorial
policy than either CTAN or CPAN. Each package
must follow a certain structure including detailed
documentation, and contributions not obeying this
are automatically rejected by the site upload scripts.
After the contribution is approved by the robots,
there is the next line: human maintainers. My pack-
age was accepted after several rounds of e-mail ex-
changes with CRAN admins, and anecdotally this
is rather the rule than an exception. One of the
problems was the compilation of examples. While
the official policy of CRAN allows for documentation
in PDF format, the admins really wanted it to be
compilable on their machines from sources. This was
problematic since CRAN machines had a rather old
\TeXX sparklines package (some of
the recent changes in the \TeXX package were written
by me to facilitate R interface). At the end I just
included the full code in my \TeXX source rather than call \usepackage{sparklines}. I would ven-
ture to say that the interaction with CTAN (or, for
this matter, CPAN) admins is easier for the authors
than dealing with CRAN. On the other hand, I can
understand the rationale behind CRAN strictness.

In summary, it was fun for me to write my first R
package. I hope it might be useful for fellow T\TeXers
to create dynamic reports with R.

References
User Manual, November 2016. https:
//stat.ethz.ch/R-manual/R-devel/library/
utils/doc/Sweave.pdf.
[2] Andreas Loeffler and Dan Luecking. Sparklines,
sparklines.
[3] R Development Core Team. R: A Language
and Environment for Statistical Computing. R
Foundation for Statistical Computing, Vienna,
Austria, 2011.
datatool.
2006.
CRAN: https://CRAN.R-project.org/
package=ltxsparklines
Github: https://github.com/borisveytsman/
ltxsparklines.
and knitr. Chapman and Hall/CRC, Boca
978-1498716963.
[8] Uwe Ziegenhagen. Dynamic reporting with
R/Sweave and \TeXX. TUGboat, 31(2):189–192,
tb98ziegenhagen.pdf.

Boris Veytsman

Systems Biology School &
Computational Materials
Science Center, MS 6A2
George Mason University
Fairfax, VA, 22030, USA
borisv (at) lk (dot) net
http://borisv.lk.net