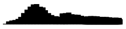
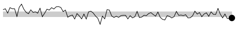
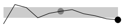
Making ltxsparklines: The journey of a CTAN contributor into the world of CRAN

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 *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 0.52  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 *R* [3]

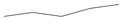
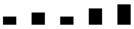
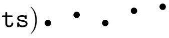
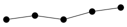
Many of us are quite familiar with *R*. Some, like me, learned to love it after the brilliant lecture by Uwe Ziegenhagen at TUG 2010 [8]. An *R*-using TEXnician spends most of her time editing and compiling `.rnw` files. These files look like LATEX, but contain "chunks" of *R* code. Processed by *R*,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

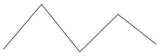
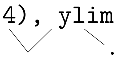
Thus it was a natural decision for me to write an *R* interface to LATEX *sparklines* package. This interface is released as the *R* package *ltxsparklines* [6].

The package defines a single *R* command, named, naturally enough, `sparkline`. It outputs a LATEX `sparkline` environment. The command has a number of possible arguments. Below these arguments are listed with the default values:

```
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'))
```

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

Other arguments can change the appearance of the graphics. Compare, for example, the result of the call `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 beginning 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 Comprehensive 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 package was accepted after several rounds of e-mail exchanges 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 TEX installation, and my examples needed a fairly recent version of LATEX *sparklines* package (some of the recent changes in the LATEX package were written by me to facilitate *R* interface). At the end I just included the full code in my LATEX source rather than call `\usepackage{sparklines}`. I would venture 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 TEXers to create dynamic reports with *R*.

References

[1] Friedrich Leisch and R-core. *Sweave User Manual*, November 2016. `https://stat.ethz.ch/R-manual/R-devel/library/utils/doc/Sweave.pdf`.

[2] Andreas Loeffler and Dan Luecking. *Sparklines*, December 2016. `https://ctan.org/pkg/sparklines`.

[3] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011.

[4] Nicola L.C. Talbot. *User Manual for datatool bundle*, July 2016. `https://ctan.org/pkg/datatool`.

[5] E.R. Tufte. *Beautiful Evidence*. Graphics Press, 2006.

[6] Boris Veytsman. *Package 'ltxsparklines'*, January 2017.
CRAN: `https://CRAN.R-project.org/package=ltxsparklines`
Github: `https://github.com/borisveytsman/ltxsparklines`.

[7] Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. ISBN 978-1498716963.

[8] Uwe Ziegenhagen. Dynamic reporting with R/Sweave and LATEX. *TUGboat*, 31(2):189–192, 2010. `https://tug.org/TUGboat/tb31-2/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`