基于pgfSweave的漂亮中文动态文档

谢益辉

2010 年 2 月 8 日

摘要

本文介绍了动态统计报告的原理以及R/Sweave的传统实现过程，重点引入pgfSweave包的说明以及它与LyX的配合使用方法，最终我们可以在各种工具的配合下只使用一种前台编写出漂亮的动态统计报告。

1 引言

我们的工作总是在各种软件中换来换去，把A软件的结果导入B，把B的表格和图形贴入C，数据在D目录下，程序在路人甲手中……这所有的工作都可以联合到一个工具中，就是Sweave。Sweave本身的原理非常简单，它就是把做上特定标记的R代码用R跑一遍，把结果写回原代码所在的地方。它之所以强大，基本上是因为R的强大（每个成功的奥特曼背后都有一个默默挨打的小怪兽）。这个工作的可能性直接源于所有的工具都在和纯文本文件打交道：R代码是纯文本（尽管它可以处理很多二进制文件）、LaTeX文档是纯文本（尽管它可以嵌入二进制的图形等对象）。

然而天外有天 — pgfSweave包在动态文档的美观问题上提供了更惊艳的解决方案。它依靠TikZ把Sweave生成的PDF图形转化为某种LaTeX代码，即用LaTeX代码重现原PDF图形。这样一来，图中所有元素的外观便和文档能保持一致了，因为此时整个LaTeX文档就彻底沦为文本文件了。图中的数学公式也能搭车转为地道的LaTeX数学公式，而不再依赖于生成它的工具（如R）。还有一点副作用就是，中文也能搭车跟进来。

除了在美化图形方面的工作不够之外，Sweave的另一个不足之处就是不能缓存运行结果，即：每次运行Sweave文档都要“从头再来”；cacheSweave包提出了解决方案，它能缓存数据对象，但对图形对象依然无能为力，一幅PDF图形，怎么缓存？世上木有不可能的事情，pgfSweave就在这个问题上继续迈进一步，使得图形也一样可以缓存，这样的话只要画图的R代码没有变化，图形就不必重画，而是直接从缓存中读取即可，所以加快了编译的整体速度。

2 了解pgfSweave包

确切地说，pgfSweave包主要仍然是基于Sweave的，只不过略微修改了一个driver函数而已。它将Sweave的`\includegraphics{*.pdf}`换成了相应的`\input{*.tikz}`或其它格式，其实

转换tikz格式的工作是tikzDevice包完成的。

首先，我们需要为pgfSweave包配置一个缓存路径[1]：

```
> setCacheDir("cache")
> pdf.options(family = "GB1")
```

然后在确保LaTeX宏包pgf已经安装（Rnw文档中要写明`\usepackage{tikz}`），并且Sweave.sty在TeX的搜索路径中。如果你不知道这个文件在哪儿能找到，Windows用户可以运行：

```
> file.path(R.home(), "share", "texmf")
```

```
[1] "C:\\PROGRA~1\\R/share/texmf"
```

pgfSweave包的主要函数就是`pgfSweave()`，它与`Sweave()`函数用法相似 — 提供一个`*.Rnw`文件给它即可。如果需要编译为PDF，可以加上`pdf = TRUE`的参数。内部运行细节本文暂且忽略。值得一提的是它对图形中数学公式的完美再现；请对比一下这个公式和图1中的数学公式：$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp(x^2/2)dx$，它们俩有差异么？貌似木有。

3 中文环境配置

为了顺利使用中文和pgfSweave生成动态中文文档，我们有一些需要注意的地方。

3.1 R语言

建议使用UTF-8编码，否则中文字符可能会被Sweave吃掉。设置UTF-8编码的语句为：

```
> options(encoding = "UTF-8")
```

可以把它放在Rprofile.site文件中，以保证每次启动R之后默认的编码就是UTF-8，或者写在Rnw文档的前面也可以。

3.2 LaTeX

据我所知，ctex宏包是最好的支持中文的LaTeX宏包了，不用它用谁呢？

3.3 LyX

一切发明创造都是为了偷懒，说得冠冕堂皇一些就是为了提高效率。懒人写LaTeX文档自然不会从`\documentclass{xxx}`写起，LyX便是偷懒的好帮手，不过从R到Sweave到LaTeX到LyX这一路配置还是有点麻烦的，没有付出哪有回报呢。

在LyX中需要做的事情是：写个使用ctex宏包的layout文件；建立从Sweave到LaTeX的转换器。

[1] 为了PDF图形设备能使用中文，此处也配置了`pdf.options()`选项。

```
> set.seed(123)
> x <- rnorm(10)
> y <- x + rnorm(5, sd = 0.25)
> model <- lm(y ~ x)
> par(mar = c(4, 4, 1, 2))
> plot(x, y, pch = seq(x), ylab = "$Y=\\beta_0 + \\beta_1 x + \\epsilon$")
> abline(model, col = "red")
> legend("bottomright", legend = sprintf("$\\hat{Y} = %.2f + %.2fx$",
+     coef(model)[1], coef(model)[2]), bty = "n")
> legend("topleft", legend = "$\\epsilon \\sim N(0, \\sigma^2)$",
+     bty = "n")
> mtext("纯正\\LaTeX{}数学公式！ ", side = 4)
> text(-0.5, 1, paste("$\\Phi(x)=\\int_{-\\infty}^x\\frac{1}{\\sqrt{2\\pi}}",
+     "\\exp(x^2/2)dx$"))
```

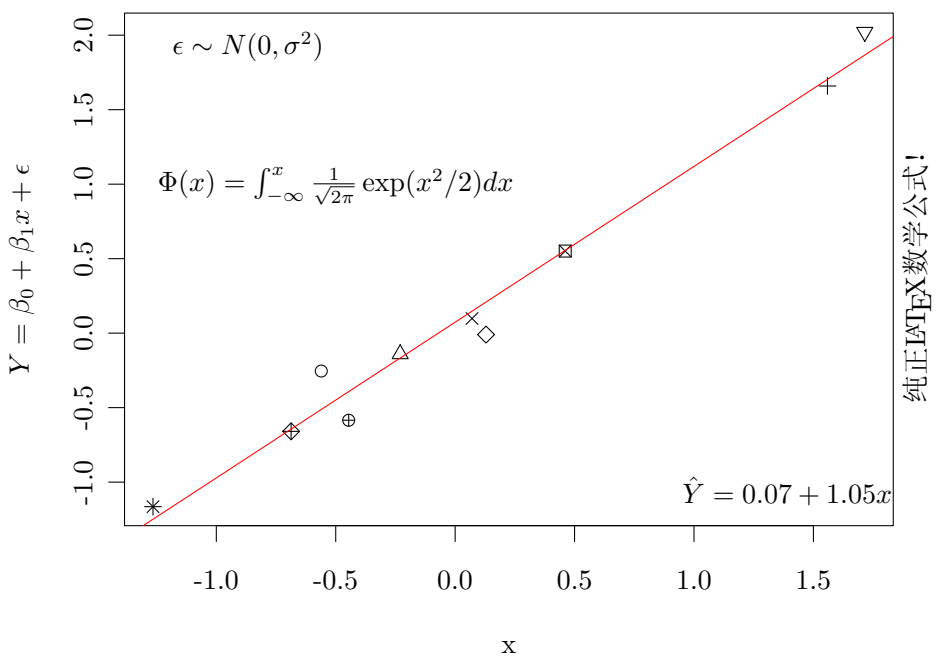

图 1: 一幅pgfSweave生成的图形，演示回归方程。动态生成x和y，建立回归模型，画出散点图并在y轴上标注出理论模型$Y = \beta_0 + \beta_1 x + \epsilon$；散点图的右下角写入一个回归方程的估计，系数从模型中动态获取（保留两位小数）；左上角写出误差项的假设。为了显摆真的很酷，画蛇添足写了个标准正态分布函数。图右侧加入一行中文，有图有真相，中文真的是被支持的。为了显摆LaTeX真的很酷，画蛇添足写了个标准正态分布函数。图右侧加入一行中文，有图有真相，中文真的是被支持的。

4 小结

pgfSweave，牛啊。