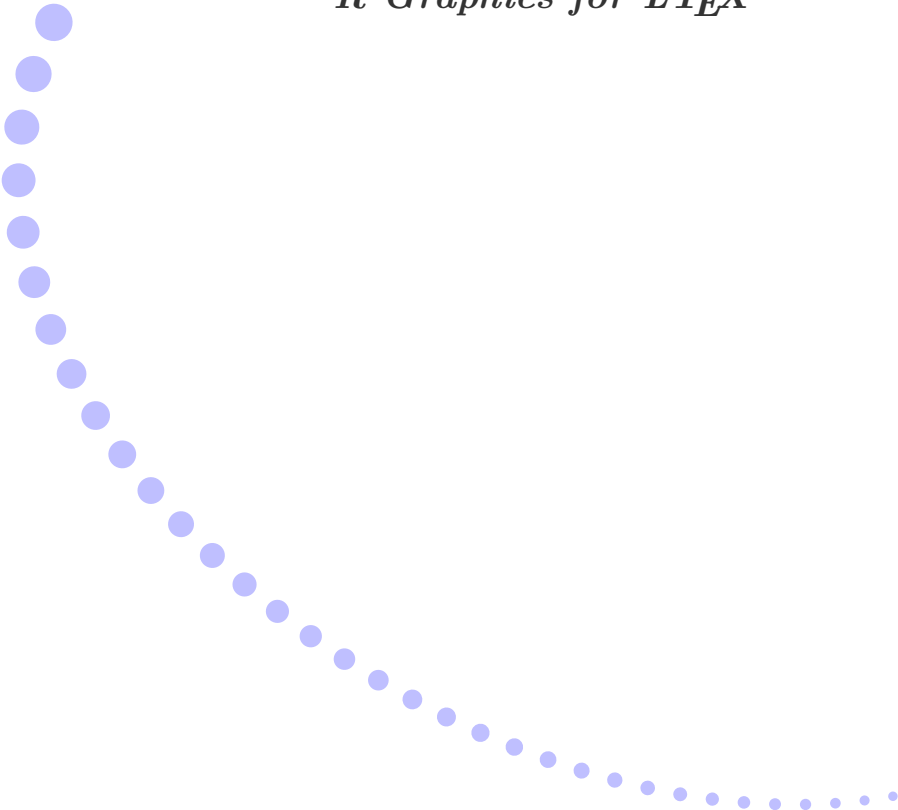


TikZ Device

R Graphics for L^AT_EX



The **tikzDevice** Package

Charlie Sharpsteen and Cameron Bracken

July 18, 2009

<http://r-forge.r-project.org/projects/tikzdevice>

Contents

1	Introduction	4
I	Usage and Examples	4
2	Loading the Package	4
3	The <code>tikz()</code> Function	5
3.1	Description	5
3.2	Usage	5
3.3	Examples	6
3.3.1	Basic Usage	6
4	R Options that Affect <code>tikzDevice</code>	8
5	The <code>getLatexCharMetrics()</code> and <code>getLatexStrWidth()</code> Functions	8
5.1	Description	8
5.2	Usage	8
5.3	Examples	8
II	Installation Guide	9
6	Installing \LaTeX	9
7	Installing <i>TikZ</i>	9
8	Instaling <code>tikzDevice</code>	9
III	Package Internals	10
9	Background	10
10	System Requirements	10
11	Character Metrics and String Width	10
11.1	Dictionaries	10
11.2	What the Heck are Acent, Decent and Width	10

12 On the Importance of Font and Style Consistency in Reports	10
13 The pgfSweave Package and Automatic Report Generation	10

1 Introduction

The **tikzDevice** package allows for R graphics output in a native L^AT_EX format. That is, the output of the `tikz()` function is plain-text files that can be interpreted using TikZ, a package for L^AT_EX. These files can be directly included in L^AT_EX documents by way of the `\input{}` statement. Allowing L^AT_EX to handle both typesetting and figure composition bestows the resulting document with a clean, unified look as there are no discontinuities in the size and selection of fonts used in the output.

This document is divided into three parts. The first part describes the functions that the package makes available to the R user and provides examples of their capabilities. Besides the R environment, use of the TikZ device device requires the user to have a working L^AT_EX compiler along with an installed version of the TikZ package. The second part of this documentation offers suggestions on how to get these tools working properly.

The third part of the documentation is intended for those who are curious as to the details of how this package has been implemented. It attempts to explain how the TikZ package does the things that it does and why it chooses to do them that way. The authors have attempted to write this part of the documentation in a way that is accessible to users as well as developers. This has been done in the hope that this project may serve as a case study in creating an R graphics device. This part of the documentation may also help those considering undertaking the transition from casual package-building to full-on hacking of the R internals.

Part I

Usage and Examples

2 Loading the Package

The functions in the **tikzDevice** package are made accessible in the R environment by using either the `library()` or `require()` functions like so:

```
require(tikzDevice)
```

Upon loading, the package will search for a usable L^AT_EX compiler. Access to L^AT_EX is essential for the device to produce correct output as the compiler is queried for font metrics several times during device output. For more information on why communication between the device and L^AT_EX is necessary, see [Part III](#). If the search for a compiler is successful the package startup message should look similar to the following:

```
filehash: Simple key-value database (2.0-1 2008-12-19)
tikzDevice: A Device for R Graphics Output in PGF/TikZ Format (v0.3.5)
Checking for a LaTeX compiler...

pdfTeX 3.1415926-1.40.9-2.2 (Web2C 7.5.7)
kpathsea version 3.5.7
...

A working LaTeX compiler was found in:
    The R environment variable R_LATEXCMD

Global option tikzLatex set to:
    /usr/texbin/latex
```

If a working \LaTeX compiler cannot be found, the **tikzDevice** package will fail to load and a warning message will be displayed:

```
An appropriate LaTeX compiler could not be found.
Access to LaTeX is currently required in order for the
TikZ device to produce output.

The following places were tested for a valid LaTeX compiler:

    A pre-existing value of the global option tikzLatex
    The R environment variable R_LATEXCMD
    The R environment variable R_PDFLATEXCMD
    The global option latexcmd
    The PATH using the command latex
    The PATH using the command pdflatex
...

Error : .onLoad failed in 'loadNamespace' for 'tikzDevice'
Error: package/namespace load failed for 'tikzDevice'
```

In this case, **tikzDevice** has done it's very best to locate a working compiler and came up empty. For information on how to deal with this situation, consult [Part II](#) of this documentation.

3 The tikz() Function

3.1 Description

The **tikz()** function provides most of the functionality of the **tikzDevice** package. This function is responsible for creating new R graphics devices that translate the output of graphics functions to the *TikZ* format. The device supports many levels of output that range from stand-alone \LaTeX documents that may be compiled directly to code chunks that must be incorporated into existing \LaTeX documents using the `\include{}` function.

3.2 Usage

The **tikz()** function opens a new graphics device and may be called with the following arguments:

```
tikz(file = "Rplots.tex", width = 7, height = 7,
      bg="white", fg="black", standAlone = FALSE, bareBones = FALSE )
```

file A character string indicating the desired path to the output file. It is recommended, but not required, that the filename end in `.tex`.

width The width of the output figure, in **inches**.

height The height of the output figure, in **inches**.

bg The starting background color for the plot.

fg The starting foreground color for the plot.

standAlone A logical value indicating whether the resulting file should be suitable for direct processing by \LaTeX .

bareBones A logical value indicating whether the resulting TikZcode produced without being placed within a \LaTeX `tikzpicture` environment.

The first five options should be familiar to anyone who has used the default graphics devices shipped with R. The options `file`, `width`, `height`, `bg` and `fg` represent the standard graphics parameters currently implemented by **tikzDevice**. The last two options, `standAlone` and `bareBones`, are specific to the `tikz()` graphics device and affect the structure the output file. Using these options `tikz()` supports three modes of output:

- Graphics production as complete \LaTeX files suitable for compilation.
- Graphics production as complete figures suitable for inclusion in \LaTeX files.
- Graphics production as raw figure code suitable for inclusion in an enclosing `tikzpicture` environment in a \LaTeX file.

The next section provides examples of how to use each type of output.

3.3 Examples

3.3.1 Basic Usage

The most common use of the `tikz()` function is to produce a plot that will be included in another \LaTeX document, such as a report. Running the following example in `langR` will produce a very simple graphic using the `plot()` function.

```
require(tikzDevice)
tikz('simpleEx.tex',width=3.5,height=3.5)
plot(1,main='Hello World!')
dev.off()
```

A simple \LaTeX document is then required to display the output. This document must include the `TikZ` as one of the packages that it loads. The `TikZ` package provides several optional libraries that provide additional functionality, however none of these libraries are currently required to use the output of `tikz()`. Inside the \LaTeX document, the contents of the file `simpleEx.tex` are imported using the `\include{}` command.

```

\documentclass{article}

% All LaTeX documents including tikz()
% output must include this package!
\usepackage{tikz}

\begin{document}
  \begin{figure}[!h]
    \centering

    % The output from tikz() is
    % imported here.
    \input{simpleEx.tex}

    \caption{Simple Example}
  \end{figure}
\end{document}

```

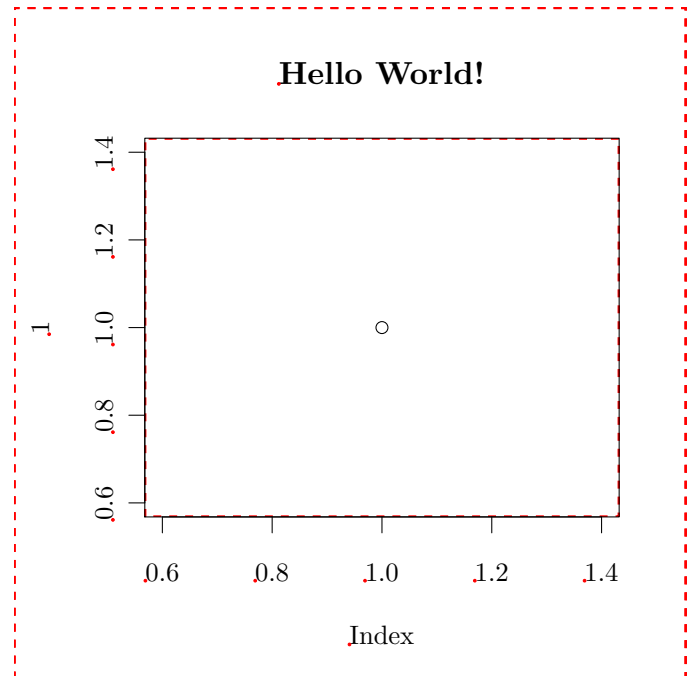


Figure 1: Example of basic `tikz()` usage.

One of the most exciting aspects of the `tikz()` function is that it allows the inclusion of arbitrary \LaTeX code in plotting commands. An important issue to note is that many \LaTeX commands are prefixed by the backslash, `\`, character. This character has a special meaning as an escape character in many computing applications, including R. Therefore, it is necessary to place two backslashes, `\\`, in the input to R commands in order to cause one to appear in the output. The next example demonstrates how to use \LaTeX commands in plot annotation.

```

require(tikzDevice)
tikz('latexEx.tex',width=3.5,height=3.5)

x <- rnorm(5)
y <- x + rnorm(5,sd=0.25)

model <- lm(y~x)
coef <- coefficients( model )
coef <- signif(coef,4)
rsq <- summary( model )$r.squared
rsq <- signif(rsq,4)

plot(x,y,main='Hello \\LaTeX!')
abline(model,col='red')

mtext( paste("Linear model: $y=",
             coef[1],"x + ",coef[2],
             "\\quad R^{2}=",rsq,"$"),
       line=0.5)

dev.off()

```

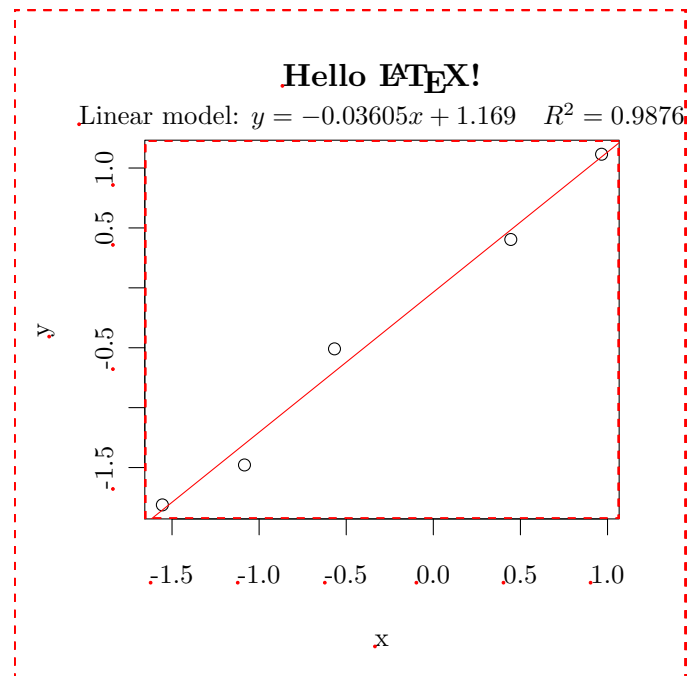


Figure 2: A more complicated example of `tikz()` usage.

4 R Options that Affect `tikzDevice`

5 The `getLatexCharMetrics()` and `getLatexStrWidth()` Functions

5.1 Description

5.2 Usage

5.3 Examples

Part II

Installation Guide

6 Installing \LaTeX

7 Installing *TikZ*

8 Instalng `tikzDevice`

Part III

Package Internals

9 Background

About TikZ, the PicTeX device and the story of two fortran programmers with a dream.

10 System Requirements

11 Character Metrics and String Width

11.1 Dictionaries

11.2 What the Heck are Acent, Decent and Width

12 On the Importance of Font and Style Consistency in Reports

13 The pgfSweave Package and Automatic Report Generation