TikZ annotations for ggplots with the ggtikz package

November 4, 2021

Contents

1 Prerequisites 2
  1.1 LaTeX side .............................................. 2
  1.2 R side .................................................... 2

2 Basic usage with \texttt{ggtikz()} 2

3 Advanced usage with canvases and annotations 3
  3.1 Single-panel plots ........................................ 3
     3.1.1 Annotation relative to the whole plot .............. 3
     3.1.2 Annotation relative to the panel .................. 4
     3.1.3 Annotation relative to data coordinates .......... 5
     3.1.4 Mixing panel and data references ................. 6
     3.1.5 Turning off clipping ................................ 7
  3.2 Multi-panel plots: wrap .................................. 9
     3.2.1 Annotations in separate panels, relative to data or panel
          coordinates ........................................... 9
     3.2.2 Annotations in separate panels, relative to data coordinates 11
  3.3 Multi-panel plots: grid ................................ 12
  3.4 Re-using annotations ..................................... 13

4 Transformed scales 15

5 Using \texttt{Inf} and -\texttt{Inf} in annotations 17

6 Using styles defined in the surrounding document 17
1 Prerequisites

1.1 LaTeX side

As the name implies, \texttt{ggtikz} requires \texttt{tikz}, which must be loaded in the document’s preamble. Furthermore, the \texttt{calc tikz} library is required.

Thus, the preamble must contain:

\begin{verbatim}
\usepackage{tikz}
\usetikzlibrary{calc}
\end{verbatim}

1.2 R side

The \texttt{tikzDevice} package is required to render plots and \texttt{ggtikz} annotations to the \texttt{tikz} format. We also have to make some base plots, using \texttt{ggplot2}.

Here, we set the graphics device to \texttt{tikz} – \texttt{ggtikz} does not work with any other graphics device!

```
library(knitr)
library(ggplot2)
library(ggtikz)
opts_chunk$set(
  dev = "tikz",
  error = TRUE,
  cache = FALSE,
  external = TRUE,
  fig.path = "example-vignette-figures/",
  fig.width = 3,
  fig.height = 3,
  fig.align = "center"
)
```

2 Basic usage with \texttt{ggtikz()}

For simple one-step annotations, the \texttt{ggtikz} helper function is available.

It accepts a \texttt{ggplot} object as its first argument. Further arguments are passed on to \texttt{ggtikzAnnotation} (see section 3).

```
p <- ggplot(mtcars, aes(x=disp, y=mpg)) + geom_point()
ggtikz(p, "\fill[red] (0.5,0.5) circle (2mm);", xy="plot")
```
3 Advanced usage with canvases and annotations

With `ggtikz()`, only a single annotation can be added to a plot. If multiple annotations are needed, then we first need to create a `ggtikzCanvas()`, to which one or more `ggtikzAnnotation()` can be added.

3.1 Single-panel plots

Let’s create a single-panel plot for annotation.

```r
p <- ggplot(mtcars, aes(disp, mpg)) +
  geom_point() +
  theme(plot.background=element_rect(color = "black", size = 1))
```

We can then set up an annotation canvas and add tikz annotations. Note that first, we print the base plot to the device \(^1\), and then the annotation canvas. The annotation canvas does not take care of drawing the annotated plot (the `ggtikz()` helper does handle this with the `draw = TRUE` parameter).

3.1.1 Annotation relative to the whole plot

\(^1\)no explicit calls to `tikz()` and `dev.off()` are needed, because knitr opens and closes the device automatically.
3.1.2 Annotation relative to the panel

canvas <- ggtikzCanvas(p)
annotation <- ggtikzAnnotation(
  "
  \draw (0,0) -- (1,1);
  \draw (0,1) -- (1,0);
  \fill[red] (0.5,0.5) circle (2mm);
  ",
  xy = "panel",
  panelx = 1, panely = 1
)
p
canvas + annotation

# first draw the plot
# then draw the annotations
3.1.3 Annotation relative to data coordinates

In addition to unitless tikz coordinates, you can also use absolute lengths, such as the 1 cm in the example below.

```r
canvas <- ggtikzCanvas(p)
annotation <- ggtikzAnnotation(
  "\draw[thick,red] (100,20) -| (400,15);
\draw[<->] (153,24) -- ++(30:1cm) node[at end, anchor=south] {Interesting!};",
  xy = "data",
  panelx = 1, panely = 1
)
p + canvas + annotation
```
3.1.4 Mixing panel and data references

The reference frames for x and y coordinates can be separately assigned as data or panel. However, note that the plot reference frame must be given for both x and y directions (with the xy argument), and cannot be mixed!

```r
canvas <- ggtikzCanvas(p)
annotation <- ggtikzAnnotation(
  "\fill[red] (0.5,30) circle (2mm);",
  x = "panel", y = "data",
  panelx = 1, panely = 1
)
p + canvas + annotation
```
3.1.5 Turning off clipping

It is possible to turn off clipping for annotations, in order to draw outside of the plot area.

```r
canvas <- ggtikzCanvas(p)
annotation_clip <- ggtikzAnnotation(
  "\fill[red] (0.1,0) circle (5mm);",
  xy = "panel",
  panelx = 1, panely = 1
)

annotation_unclip <- ggtikzAnnotation(
  "\fill[blue] (0.9,0) circle (5mm);",
  xy = "panel",
  panelx = 1, panely = 1,
  clip = "off"
)

annotation_unclip2 <- ggtikzAnnotation(
  "\draw[thick, dashed] (0,0) -- (0.5,-0.2) -- (1,0);",
  xy = "panel",
  panelx = 1, panely = 1,
  clip = "off"
)
p
p + canvas + annotation_clip + annotation_unclip + annotation_unclip2
```
However, note that the surrounding plot area is not automatically unclipped to accommodate for the annotations. This can be alleviated manually by increasing the plot margins.

\[ p + theme(plot.margin = margin(t=0.5, b = 1, unit = "cm")) \]
\[ canvas + annotation_clip + annotation_unclip + annotation_unclip2 \]
Alternatively, \texttt{ggtikz} comes with a knitr hook to automatically unclip TikZ files:

\begin{verbatim}
set_ggtikz_unclip_hook()
\end{verbatim}

Now clipping can be disabled for chunks with the \textit{chunk option} \texttt{unclip = TRUE} – however, this only works in conjunction with \texttt{external = FALSE}. If the option \texttt{external = TRUE}, then the resulting file is immediately compiled to pdf and not accessible for further post-processing.

\begin{verbatim}
# chunk options: external=FALSE, unclip=TRUE
p <- p + canvas + annotation_unclip2
\end{verbatim}

Unset the hook to restore the default clipping behavior:

\begin{verbatim}
unset_ggtikz_unclip_hook()
\end{verbatim}

### 3.2 Multi-panel plots: wrap

\begin{verbatim}
p_wrap <- p + facet_wrap(~cyl, scales="free", ncol=2)
\end{verbatim}

#### 3.2.1 Annotations in separate panels, relative to data or panel coordinates
canvas <- ggtikzCanvas(p_wrap)

# Relative to data coordinates
annotation1 <- ggtikzAnnotation("\
  node[pin=90:(110,27), circle, fill=red, inner sep=0, outer sep=0, minimum size=2pt] at (110,27) {};", xy = "data", panelx = 1, panely = 1)

# Relative to data coordinates
annotation2 <- ggtikzAnnotation("\
  node[pin=90:(200,19), circle, fill=red, inner sep=0, outer sep=0, minimum size=2pt] at (200,19) {};", xy = "data", panelx = 2, panely = 1)

# Relative to panel coordinates
annotation3 <- ggtikzAnnotation("\
  node[draw, anchor=center] at (0.5, 0.5) {Center of panel};", xy = "panel", panelx = 1, panely=2)

p_wrap
canvas + annotation1 + annotation2 + annotation3
3.2.2 Annotations in separate panels, relative to data coordinates

canvas <- ggtikzCanvas(p_wrap)
annotation1 <- ggtikzAnnotation(""
  \node[pin=90:(110,27), circle, fill=red, 
      inner sep=0, outer sep=0, minimum size=2pt] 
  at (110,27) 
  {}; 
", xy = "data", 
panelx = 1, panely = 1
)
annotation2 <- ggtikzAnnotation(""
  \node[pin=90:(200,19), circle, fill=red, 
      inner sep=0, outer sep=0, minimum size=2pt] 
  at (200,19) 
  {}; 
", xy = "data", 
panelx = 2, panely = 1
)
p_wrap
canvas + annotation1 + annotation2
3.3 Multi-panel plots: grid

Annotations can also be made on individual panels of plots faceted with `facet_grid`.

```r
p_grid <- p + facet_grid(gear~cyl, scales="free", as.table=FALSE)

canvas <- ggtikzCanvas(p_grid)
annot_grid1 <- ggtikzAnnotation("\\node[fill=white, draw, text width=2cm] at (0.5,0.5) 
\{panelx=1, panely=1\};",
xy = "panel",
panelx = 1, panely = 1)
annot_grid2 <- ggtikzAnnotation("\\node[fill=white, draw, text width=2cm] at (0.5,0.5) 
\{panelx=2, panely=3\};",
xy = "panel",
panelx = 2, panely = 3)
annot_grid3 <- ggtikzAnnotation("\draw[<-, blue] (90,15) -- ++(30:5mm) 
node [at end, anchor=south west] {(90,15)};
",
xy = "data",
)```

3.4 Re-using annotations

Annotations can be re-used between plots and \texttt{gg tikz} canvases. However, be aware that panel position specifications rely on the visual position of the panels, and \textit{not on the value of the facet variables}.

```r
p_grid2 <- p + facet_grid(gear~cyl, scales="free", as.table=TRUE)
canvas2 <- ggtikzCanvas(p_grid2)
p_grid2
canvas2 + annot_grid1 + annot_grid2 + annot_grid3
```
It is also not possible to add annotations to a plot for which the requested panels are not available.
4 Transformed scales

TikZ coordinates can automatically be transformed to accommodate transformed scales, such as log-transformed ones. This is activated by setting `transform = TRUE` in the call to `ggtikzAnnotation`. With `transform = FALSE`, annotations made with data coordinates are out of place in log plots, but the transformation calculation can be done manually.

```r
p_log <- ggplot(mtcars, aes(mpg, disp)) +
geom_point() +
  scale_x_continuous(trans="log10")
```

```r
canvas_log <- ggtikzCanvas(p_log)
# Untransformed coordinates: wrong position
annot_log <- ggtikzAnnotation(
  "\fill[red] (1,100) circle (2mm);
\node[anchor=west, text=red] at (1, 100)"
```
The circle is not at (1,100)!

Transformed coordinates: correct position
The literal coordinate in the node text was wrapped in an \mbox LaTeX command to prevent automatic transformation -- it can't distinguish between coordinates which are supposed to be text, and actual coordinates.

```
annot_log2 <- ggtikzAnnotation(
  "\fill[blue] (20,200) circle (2mm);
  \node[anchor=south, text=blue] at (20, 200)
  {This circle is at (\mbox{20,200})!!};
  ", xy = "data", transform = TRUE, panelx = 1, panely = 1
)
```

Untransformed coordinates, calculated manually by hand:
# Transforming coordinates: incorrect position

```
annot_log3 <- ggtikzAnnotation(
  "\fill[magenta] (1.477,400) circle (2mm);
  \node[anchor=east, text=magenta] at (1.477, 400)
  {This circle is at (30,400)!!};
  ", xy = "data", transform = FALSE, panelx = 1, panely = 1
)
```

```
p_log <- canvas_log + annot_log + annot_log2 + annot_log3
```
5 Using \textit{Inf} and -\textit{Inf} in annotations

In \texttt{ggplot2}, \texttt{-Inf} and \texttt{Inf} can be used to refer to the edge of a panel. This is also possible in \texttt{ggtikzAnnotations}, by setting \texttt{replace_inf = TRUE}. \texttt{-Inf} and \texttt{Inf} will then be automatically replaced to refer to the left/bottom or right/top edge of a panel, respectively.

\begin{verbatim}
p_log_border <- p_log + theme(panel.border = element_rect(fill=NA, size = 2))
canvas_log_border <- ggtikzCanvas(p_log_border)
annot_inf <- ggtikzAnnotation(  "\\draw[red, thick] (-Inf,200) -| (20,-Inf); \\draw[green, thick] (-Inf,200) |- (20,-Inf);  ", xy = "data", replace_inf = TRUE, panelx = 1, panely = 1 )
p_log_border
canvas_log_border + annot_inf
\end{verbatim}

6 Using styles defined in the surrounding document

Annotations can access styles which are defined in the containing document before the relevant \texttt{.tikz} file is included, allowing you to re-use global styles. Note that by default, \texttt{knitr} sets the option \texttt{external} to \texttt{TRUE}. Therefore, \texttt{tikz} graphics are pre-compiled to \texttt{pdf}. In that case, the \texttt{tikzDevice} needs to know about these styles, or an error will occur during externalization.
\tikzset{loud/.style={
\hspace{1cm}
draw=yellow,
\hspace{1cm}
fill=red,
\hspace{1cm}
text=blue
}}

\node[loud] at (0.5,0.5) {Look at me!};

\begin{knitrout}
\flushleft
\small
\verbatim
\begin{verbatim}
canvas <- ggtikzCanvas(p)
styled_annot <- ggtikzAnnotation("\node[loud] at (0.5,0.5) {Look at me!};", xy = "plot")
p
\end{verbatim}
\end{knitrout