

Presentations and posters

Tools for Reproducible Research

Karl Broman

Biostatistics & Medical Informatics, UW–Madison

kbroman.org

github.com/kbroman

@kbroman

Course web: kbroman.org/Tools4RR

It's arguably less critical that presentation slides or a poster be reproducible. Nevertheless, there can be great personal advantage to the automated generation of figures and such in slides or a poster: if the primary data should change, or if some analysis mistake is discovered, it will be easier to revise the presentation.

My primary goal is to get you to ditch Powerpoint/Keynote in favor of reproducible alternatives. I will primarily focus on the Beamer package for LaTeX, for both slides and posters. But I will also touch upon the use of slidify to make Markdown-based slides for a talk.

Powerpoint/Keynote

- + Standard
- + Easy to share slides
- + WYSIWYG (mostly)
- + Fancy animations
- Font problems
- Lots of copy-paste
- Hard to get equations
- Not reproducible

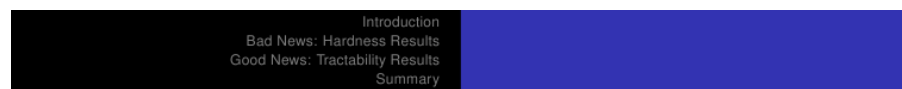
2

Powerpoint and Keynote do have their advantages, the principal one being that everyone is using these tools, which makes it easy to share slides with friends.

But we've all seen terrible font problems in important presentations, mostly due to incompatibilities between Windows and Mac versions of Powerpoint: fonts should be, but aren't, embedded in the presentation.

And insertion of figures requires tedious copy-paste, usually followed by manual resizing and adjustment of figure placement. And if the figures are revised (because the data changed or some mistake was found in the analysis), we'll have to repeat all of that.

L^AT_EX Beamer package



On the Complexity of SNP Block Partitioning Under the Perfect Phylogeny Model

Jens Gramm¹ Tzvika Hartman² Till Nierhoff³
Roded Sharan⁴ Till Tantau⁵

¹Universität Tübingen, Germany

²Bar-Ilan University, Ramat-Gan, Israel

³International Computer Science Institute, Berkeley, USA

⁴Tel-Aviv University, Israel

⁵Universität zu Lübeck, Germany

Workshop on Algorithms in Bioinformatics, 2006



Until recently, I'd been making L^AT_EX slides using the `article` document class, just revising the page size and make the fonts big.

The Beamer package for L^AT_EX is easier, but I was turned off by the standard slides that people were producing with Beamer, such as the one shown: far too much junk on the screen, and on every single slide.

You can get rid of all of that. All of the slides I'm making for this course are produced with Beamer.

There's good facility for adding simple animations (progressively showing or hiding different elements on the slide).

But you **are** writing L^AT_EX, so the coding can be a bit verbose.

Get rid of the junk

```
\usetheme{default}  
\beamertemplatenavigationsymbolsempy
```

4

The first thing to do is to get rid of all of the junk.

It's surprisingly easy: default theme and remove navigation symbols.

Change colors

```
\definecolor{foreground}{RGB}{255,255,255}
\definecolor{background}{RGB}{24,24,24}
\definecolor{title}{RGB}{107,174,214}
\definecolor{subtitle}{RGB}{102,255,204}
\definecolor{hilit}{RGB}{102,255,204}
\definecolor{lolit}{RGB}{155,155,155}

\setbeamercolor{titlelike}{fg=title}
\setbeamercolor{subtitlle}{fg=subtitle}
\setbeamercolor{institute}{fg=lolit}
\setbeamercolor{normal text}{fg=foreground,bg=background}
\setbeamercolor{item}{fg=foreground} % color of bullets
\setbeamercolor{subitem}{fg=lolit}
\setbeamercolor{itemize/enumerate subbody}{fg=lolit}
\setbeamerfont{itemize subitem}{\textendash}
\setbeamerfont{itemize/enumerate subbody}{size=\footnotesize}
\setbeamerfont{itemize/enumerate subitem}{size=\footnotesize}

\newcommand{\hilit}{\color{hilit}}
\newcommand{\lolit}{\color{lolit}}
```

5

I prefer light text on a dark background.

The tricky part is that Beamer has special names for everything.

It would be best if I created a new theme, but I don't want to take the time to figure that out.

Also, slide numbers and fonts

```
% slide number
\setbeamertemplate{footline}{%
  \raisebox{5pt}{\makebox[\paperwidth]{\hfill\makebox[20pt]{\lollipops
    \scriptsize\insertframenumbers}}}\hspace*{5pt}}

% font
\usepackage{fontspec}
% http://www.gust.org.pl/projects/e-foundry/tex-gyre/
%   ...   heros/qhv2.004otf.zip
\setsansfont
  [ ExternalLocation = ../fonts/ ,
    UprightFont = *-regular ,
    BoldFont = *-bold ,
    ItalicFont = *-italic ,
    BoldItalicFont = *-bolditalic ]{texgyreheros}
% Palatino for notes
\setbeamerfont{note page}{family*=pplx,size=\footnotesize}
```

6

I also want the slide number in the bottom-right, and I want a different font: something a bit more blocky, which I think is easier to read on the screen.

Title slide

```
\title{Put title here}
\subtitle{And maybe a subtitle}
\author{Author name}
\institute{Biostatistics \& Medical Informatics,
  UW{\textendash}Madison}
\date{\tt \scriptsize biostat.wisc.edu/{\textasciitilde}kbroman}

\begin{document}

{
\setbeamertemplate{footline}{} % no slide number here
\frame{
  \titlepage

\note{
  Summary of the talk, as a note.
}
} }
```

7

The title slide is created with `\titlepage`, having first defined `\title`, `\author`, etc.

The extra curly braces are to get the “no slide number” to apply just to the title slide. You can put notes on slides and then make a version that has the slide above the notes. See what I do with the slides for this course, or ask me for help.

Typical slide

```
\begin{frame}{Title of slide}

\vspace{24pt} \begin{itemize} \itemsep8pt
\item Bullet 1
\item Bullet 2
\item Bullet 3
\end{itemize}

\note{
  Put a note here
}
\end{frame}
```

8

A typical slide is set between `\begin{frame}{title}` and `\end{frame}`.

You get bullet points with the `itemize` environment. I'll mess around a bit with `\vspace` and `\itemsep`. And I'll create shortcuts with `\newcommand` for these.

Slide with a figure

```
\begin{frame}{Title of slide}

\centerline{\includegraphics[height=0.75\textheight]{%
             Figs/a_figure.png}}

\note{
  Put a note here
}
\end{frame}
```

9

I'd typically generate figures externally and include them with `\includegraphics`.

Figures with KnitR

```
<<knitr_options, echo=FALSE>>=
opts_chunk$set(echo=FALSE, fig.height=7, fig.width=10)
change_colors <-
function(bg=rgb(24,24,24, maxColorValue=255), fg="white")
  par(bg=bg, fg=fg, col=fg, col.axis=fg, col.lab=fg,
      col.main=fg, col.sub=fg)
@

<<pdf_figure>>=
change_colors()
par(las=1)
n <- 100
x <- rnorm(n)
y <- 2*x + rnorm(n)
plot(x, y, pch=16, col="slateblue")
@
```

10

You could use a knitr code chunk, in the same way we discussed for manuscripts, in the last lecture.

Figures with KnitR

```
% << >>= all on one line!  
<<png_figure, dev="png", fig.align="center",  
  dev.args=list(pointsize=30),  
  fig.height=15, fig.width=15, out.height="0.75\\textheight",  
  out.width="0.75\\textheight">>=  
change_colors(bg=rgb(32,32,32,maxColorValue=255))  
par(las=1)  
n <- 251  
x <- y <- seq(-pi, pi, len=n)  
z <- matrix(ncol=n, nrow=n)  
for(i in seq(along=x))  
  for(j in seq(along=y))  
    z[i,j] <- sin(x[i]) + cos(y[j])  
image(x,y,z)  
@
```

11

To create a PNG figure (which can give much smaller file sizes for things like an image or a dense scatterplot), use the chunk option `dev="png"`.

For some reason, RGB colors don't match well between PNG files and the PDF, so I have to muck about to get the background of the PNG to match the background on the slides.

It's also a bit of work to get the resolution and text size just right.

I split the initial line defining the code chunk across multiple lines here, so it could all be seen, but in practice the whole `<< >>=` bit needs to be on one line.

Slides with notes

```
\documentclass[12pt,t]{beamer}
\setbeameroption{hide notes}
\setbeamertemplate{note page}[plain]
```

```
\documentclass[12pt,t,handout]{beamer}
\setbeameroption{show notes}
\setbeamertemplate{note page}[plain]
\def\notescolors{1}
```

```
\ifx\notescolors\undefined % slides
  \definecolor{foreground}{RGB}{255,255,255}
  \definecolor{background}{RGB}{24,24,24}
\else % notes
  \definecolor{background}{RGB}{255,255,255}
  \definecolor{foreground}{RGB}{24,24,24}
\fi
```

12

To create a version of your slides with notes, include `\note{ }` on every slide.

I then include the code in the top box in the slide version, the middle box in the note version, and the stuff at the bottom in both. The bit at the bottom selects colors to be light text on a dark background in the slides and dark text on a light background in the notes version.

I wrote a ruby script to create a notes version from the slide version (replace the code in the top box with the code in the middle box).

I then use `pdfnup` (part of PDFjam) to make 2-up pages (slides at the top, notes at the bottom). The only problem with `pdfnup` is that it strips off all of the hyperlinks.

Simple animations

```
\begin{frame}{Bullets entering one at a time}

\bbi
\item Bullet 1
\onslide<2->{\item Bullet 2}
\onslide<3->{\item Bullet 3}
\onslide<4->{\item Bullet 4}
\ei

\note{
  Do this sparingly.
}
\end{frame}
```

13

It's easy to add a bit of animation, such as with bullets appearing one by one. Use `\onslide` or `\only`.

Here, the bullets will appear one at a time.

Beamer just expands the PDF, with this slide becoming multiple pages.

Simple animations

```
\begin{frame}{Bullets entering one at a time}

\bbi
\item {\lollit \only<1>{\color{foreground}} Bullet 1}
\item {\lollit \only<2>{\color{foreground}} Bullet 2}
\item {\lollit \only<3>{\color{foreground}} Bullet 3}
\item {\lollit \only<4>{\color{foreground}} Bullet 4}
\ei

\note{
  Do this sparingly.
}
\end{frame}
```

In this version, the bullets will go from dim to bright, one at a time.

Slidify and R Markdown

```
## Slide title

- Bullet 1
- Bullet 2
- Bullet 3
- Bullet 4

---

## A figure

```{r a_figure, echo=FALSE, fig.align="center"}
par(las=1)
n <- 100
x <- rnorm(n)
y <- 2*x + rnorm(n)
plot(x, y, pch=16, col="slateblue")
```
```

15

Slidify makes it super easy to create html-based slides with R Markdown. Three dashes separate slides, and two pound symbols (section heading) indicate the slide title.

The chief advantage is that you can make nice slides with very little markup. And there are a ton of options, like having embedded quizzes.

The disadvantage is that it's a bit harder to get fine control of the layout. And I've found it a bit risky to use html-based slides for a presentation. PDF is more trustworthy.

In principle, you can use pandoc to convert the slides to PDF, but I've not been happy with the result. You could also print them from the browser, but I only got a good result with Safari. (Firefox included some links on the first page, and Chrome produced total garbage.)

Personally, I'm going to stick with Beamer for important presentations, but slidify seems good for informal presentations (e.g., to collaborators) or for a course.

Using slidify

```
library(devtools)
install_github("slidify", "ramnathv")
install_github("slidifyLibraries", "ramnathv")

library(slidify)
setwd("~/Docs/Talks/")
author("slidify_example")

# edit ~/Docs/Talks/slidy_example/index.Rmd

slidify("index.Rmd")
browseURL("index.html")
```

16

To use slidify, download the slidify and slidifyLibraries packages from GitHub, and use `author()` to create the file to edit, and then `slidify()` to compile the result.

Note that `author("slidify_example")` changes the working directory.

YAML header

```
---
title      : Slidify example
subtitle   : Tools for reproducible research
author     : Karl Broman
job        : Biostatistics & Medical Informatics, UW-Madison
framework  : io2012          # {io2012, html5slides, shower, ...}
highlighter : highlight.js   # {highlight.js, prettify, highlight}
hitheme    : tomorrow       #
widgets    : [mathjax]      # {mathjax, quiz, bootstrap}
mode       : standalone     # {selfcontained, standalone, draft}
---
```

17

There's a bit at the top of the file to define the slide title and layout.

framework defines the slide style. **highlighter** is the method to give syntax highlighting. With **mode** "standalone," some otherwise-external files are embedded in the html file.

YAML is a "human-readable data serialization format." (Serialization means it can be easily transmitted over a network.) It's a well-defined way of describing potentially complex data objects.

Change the title slide colors

```
<style>
.title-slide {
  background-color: #EEE;
}

.title-slide hgroup > h1,
.title-slide hgroup > h2 {
  color: #005;
}
</style>
```

18

The default colors for the title slide with framework `io2012` are really terrible. Include a bit of CSS code in your `.Rmd` file to fix that.

There are a bunch of named colors in html, or you can use codes like “`#005;`” or “`#000055;`” for RGB (R=00, G=00, B=55).

Beamer-based posters

```
\documentclass[final,plain]{beamer}
\usepackage[size=custom,width=152.4,height=91.44,scale=1.2]{%
  beamerposter}

\newlength{\sepwid}
\newlength{\onecolwid}
\newlength{\halfcolwid}
\newlength{\twocolwid}
\newlength{\threecolwid}

\setlength{\sepwid}{0.0192\paperwidth}
\setlength{\onecolwid}{0.176\paperwidth}
\setlength{\halfcolwid}{0.0784\paperwidth}
\setlength{\twocolwid}{0.3712\paperwidth}
\setlength{\threecolwid}{0.5664\paperwidth}
\setlength{\topmargin}{-0.5in}
\usetheme{confposter}
```

19

Beamer can be used to make posters, too. It's a lot of work, but with my latest poster, I'm finally convinced of the value of these large-format posters.

At UW-Madison, the Digital Media Center will print posters for you (\$5 per square foot for paper; \$7 per square foot for cloth). The cloth ones eliminate the need to carry a tube, but they need to be packed gently and probably still require a bit of ironing on the other end.

width and **height** are in centimeters. Use **scale** to increase the font sizes overall. Then carefully calculate the **sepwid**, **onecolwid**, etc., so that everything fits just right.

Basic code for a poster

```
\title{Data visualizations should be more interactive}
\author{Karl W Broman}
\institute{University of Wisconsin--Madison}

\begin{frame}[t]
\begin{columns}[t]
  \begin{column}{\sepwid}\end{column} % empty spacer column
  \begin{column}{\onecolwid}
    \begin{exampleblock}{\Large Introduction}{
      \begin{itemize} \itemsep18pt
        \item Bullet 1
        \item Bullet 2
      \end{itemize}
    }
    \colonevsep % between blocks
    \begin{block}{Barriers}{
    }
  \end{column}
\end{columns}
\end{frame}
```

20

The whole thing is within a single `frame` environment, and then with `columns` and `column`.

The blocks within a column are within `exampleblock` or `block` environments.

Between-block spacing

```
\newcommand{\colonevsep}{\vspace{16mm}}  
\newcommand{\coltwovsep}{\vspace{35.5mm}}  
\newcommand{\colthreevsep}{\vspace{14mm}}  
\newcommand{\colfourvsep}{\vspace{16mm}}  
\newcommand{\colfivevsep}{\vspace{23mm}}
```

I define different between-block spacings for each column, to get them all to have the same length.

Summary

- ▶ Use LaTeX/Beamer or Slidify to create reproducible slides.
- ▶ Use LaTeX/Beamer to create reproducible posters.
- ▶ Include KnitR code chunks to create figures directly.
- ▶ Or keep the code for figures separate.

22

To make reproducible slides/posters, you need to dump PowerPoint.

With each of these approaches, you can use KnitR code chunks. But I still tend to produce the figures separately and include them with `\includegraphics`.