An example poster using KnitR and LaTeX

Daniel Guest
University of Minnesota, Department of Psychology, Auditory Perception and Cognition Lab

Introduction

- This is an example poster created in LaTeX
- A few LaTeX packages and other pieces of code and software are used to make this all work:
  - The beamer and beamerposter packages are integral to this poster
  - A slightly modified version of the beamerposter theme available from `http://www.nathanieljohnston.com/2009/08/latex-poster-template/` was used to style the visual aspects of the poster
  - KnitR was used to allow the output of R code (text, numbers, and figures) to be included directly in the poster

Motivations

Why use LaTeX to make a poster? What advantages does it offer over PowerPoint or other alternatives?

- LaTeX emphasizes a clear separation of content from form, allowing you to focus on the "what" rather than the "how"
- KnitR allows for syntax-highlighted R code, or the output of R code, to be included directly in the poster
- LaTeX has great tools for high-quality typesetting of math, like $\int 2x^2 dx$
- Bibliography systems like biblatex make managing citations and including them in your poster easy
- If there's something unique you need to be able to do (like write International Phonetic Alphabet, or draw diagrams), you can likely extend LaTeX through packages to do so

How it works

- The files
  - beamerthemconfposter.sty — contains definitions of the title and blocks, as well as colors and theme options
  - beamerposter.sty — provides the beamerposter package
  - poster.Rnw — the source file, which is turned into a .tex file by KnitR
- The source file
  - The poster is composed of a title and columns, with each column being subdivided into blocks
  - Blocks contain the content of the poster, and come in two flavors – normal (like the first block, "Introduction") and alert (like the this block, "Motivations")
  - R code is delimited by special characters «» and @

Some examples

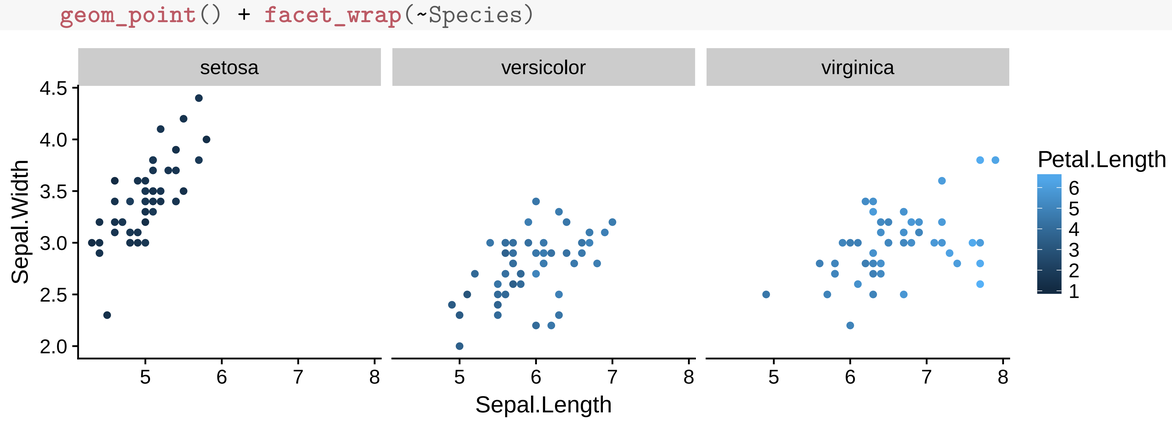
KnitR supports high-quality syntax-highlighting of R (and many other languages), as well as directly including the output of R in the output document

```
my_data = iris
my_summary = my_data %>% group_by(Species) %>% summarize(sepallen=mean(Sepal.Length),
                                                         sepalwid=mean(Sepal.Width),
                                                         petallen=mean(Petal.Length),
                                                         petalwid=mean(Petal.Width))
my_summary
```

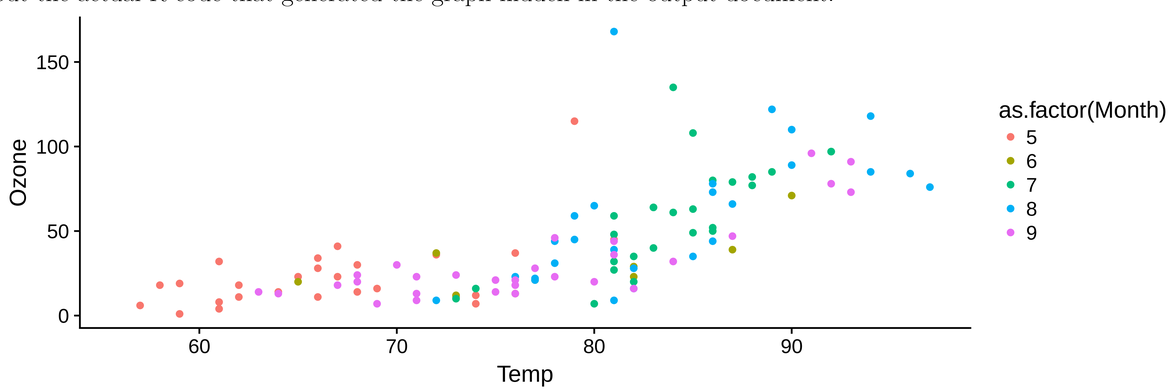
```
## # A tibble: 3 x 5
##      Species sepallen sepalwid petallen petalwid
##       <fctr>    <dbl>    <dbl>    <dbl>    <dbl>
## 1     setosa      5.0      3.4      1.5     0.25
## 2 versicolor      5.9      2.8      4.3     1.33
## 3  virginica      6.6      3.0      5.6     2.03
```

This functionality includes graphs!

```
ggplot(my_data, aes(x=Sepal.Length, y=Sepal.Width, color=Petal.Length)) +
    geom_point() + facet_wrap(~Species)
```



And, there's a variety of options to control whether or not your R code is visible or hidden and the size and characteristics of your output text and/or graphs. For example, here's a graph using the airquality data set in R, but the actual R code that generated the graph hidden in the output document.



Conclusions