Course title

Lecture title

Your name

University | Course code

Contents

Before you begin

This template is aimed at people who want to knit their R Markdown documents to *both* HTML and PDF with as few surprises as possible. As the name suggests, I predominantly use it for my lecture notes. But I find that it works well for writing papers too.

See the package README for a longer description, as well as potential gotchas and limitations (e.g. font support for different LaTeX engines).

Template features

Here are some examples of features not available in vanilla R Markdown and how to use them.

Multi-column environments

Multi-column environments are supported via's Pandoc's fenced_divs syntax and some preamble sugar (bundled together with the template). For example, a two-column section would look like this.

Here is some example **dplyr** code.

```
library(dplyr)

mtcars %>%
  group_by(am) %>%
  summarise(mean(mpg))
```

```
## # A tibble: 2 x 2
##      am `mean(mpg)`
##   <dbl>       <dbl>
## 1     0        17.1
## 2     1        24.4
```

And the **data.table** equivalent.

```
library(data.table)

mtcars_dt = as.data.table(mtcars)
mtcars_dt[, mean(mpg), by = am]
```

```
##    am       V1
## 1:  1 24.39231
## 2:  0 17.14737
```

The same idea can be extended to additional columns and the individual column widths are also adjustable.

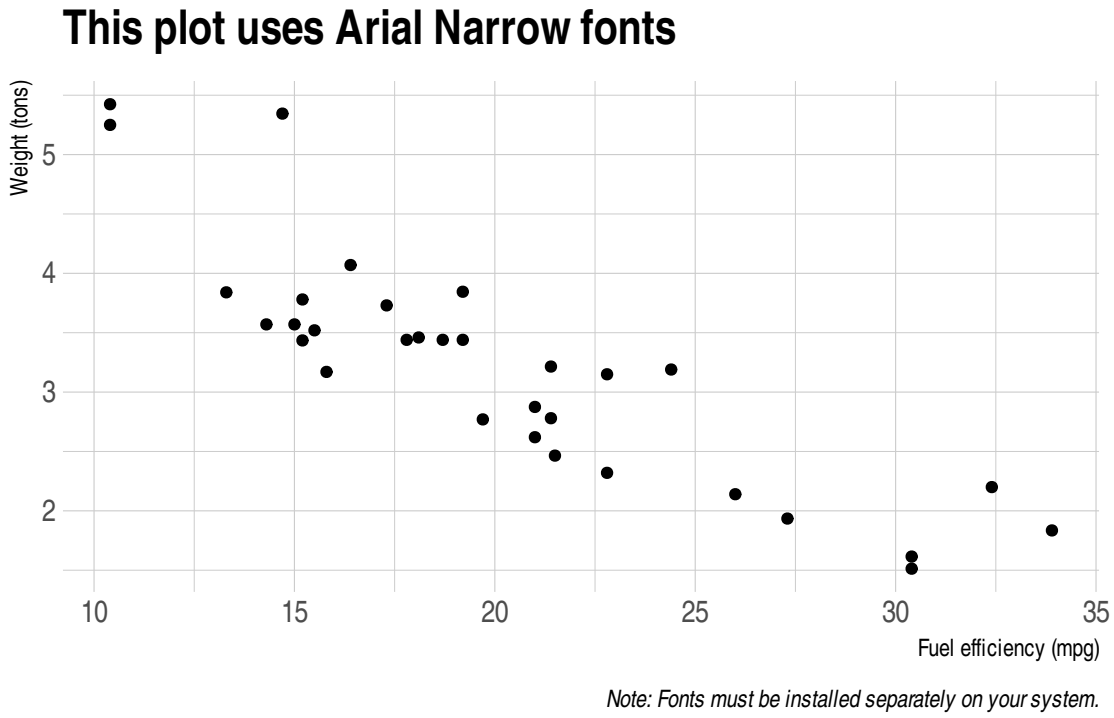
PDF support for non-standard fonts

This is an easy one; simply a matter of adding `dev: cairo_pdf` to the YAML. But it's nice not having to remember that every time, no?

Note: As the figure caption suggests, to run this next chunk you'll need to add Arial Narrow to your font book if it's not installed on your system already.

```
library(ggplot2)
library(hrbrthemes)

ggplot(mtcars, aes(mpg, wt)) +
  geom_point() +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="This plot uses Arial Narrow fonts",
       caption="Note: Fonts must be installed separately on your system.") +
  theme_ipsum()
```



Ignore interactive content when exporting to PDF

In general, this template tries to do a good job of automatically handling (i.e. ignoring) interactive content when exporting to PDF. A notable exception is with embedded interactive content like external GIFs. In this case, rather than typing the usual, say, `![](mind-blown.gif)` directly in the Rmd file, you should include the figure with `knitr::include_graphics` in an R chunk. This will allow you to control whether it renders, conditional on output format. For example, the following chunk will render an actual GIF when the knit target is HTML format, versus a message when that target is PDF format.

```
## Sorry, this GIF is only available in the the HTML version of the notes.
```