Generating Reports with R, Under Program Control

Jon Meek
meekjt (at) gmail.com
meekj (at) ieee.org

`https://github.com/meekj/TRU-May2016`

21-May-2016 / Trenton R Users

Generating Reports - Overview

- These slides were made with R + LaTeX + Beamer + Sweave
- R + LaTeX + Sweave
  - Publication quality PDF
  - Page breaks are troublesome, and unnecessary for browser viewing!
  - LaTeX is somewhat complex, and can be a pain with syntax errors
- R + Markdown + knitr → nice HTML output
  - Simple syntax
  - Easy to show R code
  - No page constraints
- I used to say (until (exactly) one month ago):
  - For all of above, self-contained flexibility is limited, a template method may be required if there are a variable number of plots, complex CLI arguments, etc.
- Now I say:
  - At least with knitr, creating reports using R with a program determined set of of plots, tables, and other content, is straight forward
  - Adding the 'littler' and 'docopt' packages allows such reports to be generated from the command line / cron job in a UNIXy standard way

Running R - Development and Report Generation

- Interactive data analysis & development
  - ESS - Emacs Speaks Statistics
  - RStudio - Best for non-Emacs users
  - In both, code can be selected then executed
  - RStudio provides an interface to knitr
- Running batch jobs
  - `R CMD Sweave ~/wpl/talks/tru-201605/Rreports.Rnw`
  - `#!/usr/local/bin/Rscript`
  - `#!/usr/local/bin/r` - "littler" better CLI
- Reports
  - R + LaTeX + Sweave → publication quality PDF
  - R + Markdown + knitr → nice HTML output
- Interactive Web Applications
  - Shiny - Full featured interactive applications (add a report gen button?)
  - ggvis, GoogleVis, etc - JavaScript active graphics

Example Sweave PDF

- nas-perf20140421.pdf

Minimal knitr - The Code

~~~
# A Minimal R / knitr Example
## Jon Meek - TRU May 2016 - Trenton R Users
## Initialize and plot
Set "echo=TRUE, message=TRUE, warning=TRUE" to show all work.

```{r initialize, echo=TRUE, message=TRUE, warning=TRUE}
## First R code block, this is a R comment
library(ggplot2)

FigureWidth  <- 16
FigureHeight <- 8

xv <- 0:99
yv <- sin(xv / 5)

ggplot() + geom_line(aes(x = xv, y = yv)) + geom_point(aes(x = xv, y = yv))
```
## The plot in it's own chunk, code and messages supressed

```{r Example1, echo=FALSE, message=FALSE, fig.width = FigureWidth, fig.height = FigureHeight}

ggplot() + geom_line(aes(x = xv, y = yv)) + geom_point(aes(x = xv, y = yv))

```
Generate HTML with a command like:

    Rscript -e "library(knitr); knit2html('minimal-knitr.Rmd')"

this assumes that the user is in the directory containing the code, and wants the HTML output in the same direc

Or, in a more production friendly way:

    Rscript -e "setwd('/n2/r-reports/'); library(knitr); knit2html('/usr/local/bin/minimal-knitr.Rmd')"
~~~

Minimal knitr - The Result

- Look at HTML output, note the single file, graphics are embedded
- Development is simple, just run R blocks in .Rmd file

-
  ```
  Rscript -e "setwd('/n2/r-reports/'); library(knitr); \
  knit2html('/usr/local/bin/minimal-knitr.Rmd')"
  ```
  - setwd sets the directory where the intermediate .md and final .html files are written
  - The argument to knit2html is the path to the ”source code”.
  - RStudio has an interactive GUI way to do the same thing (I believe)
- We used only knitr
- Do more with the rmarkdown package & pandoc
- knitr can produce other types of output

The knitr Breakthrough

- Note that the knit2html argument is a file
- Suggests limited flexibility with a static .Rmd file
- "No one seemed to care" - based on many searches over time
- Solvable via template or brute force programming in Perl, Python, etc
- But it's ugly, requires multiple files, and dev / testing is troublesome
- But, it turns out that there is a way to feed knit2html within a program!!
    - `hdata <- knit2html(text = kdata)`
    - where kdata is a character vector of R code
    - and the output is a character vector of HTML
- Now we can simply generate reports under program control
- For development, `eval(parse(text = kdata))` assuming kdata is pure R
- Note that multiple plots per R code chunk was always possible, but adding HTML requires leaving a code chunk

Command Line Interface

- Would like to run R analysis like any other UNIX program
  - Provide R programs to end users (non - UseRs)
  - Use common “–option argument” syntax
  - Also good for cron jobs
- Solution: “littler” and the docopt package
- Much better than using environment variables to get options to Rscript

A More Complex Example

- Water levels for tidal Delaware River and Bay
- Some parameters determined at runtime: days in month, measurement stations
- Might want to add a special analysis based on the data
- The wl-st-points.r program:

  -
    ```
    wl-st-points.r -outdir /n2/r-reports \
    -datadir  /lab/R/noaa/data -month 201604
    ```
  - Look at the HTML output
  - Look at the code

Other R Related Projects

- libpcapR - Rcpp interface to tcpdump packet capture files (alpha now)
- MACaddrR - Replaces manufacturer portion of hex address with abbreviated mfg name
- iperf network stress testing tools (Perl, C++, analysis in R)
- Rcpp log file parser to load data frames
- Estimated bandwidth usage from log files using Rcpp
- Use Perl Net::Netmask data to identify network that contains an IP address (currently vaporware!)