Some New Directions in Statistical Computing

Frank E Harrell Jr Jeffrey Horner

Department of Biostatistics
Vanderbilt University School of Medicine
Nashville TN

5 December 2012

Goals of Reproducible Analysis/Reporting

- Be able to reproduce your own results
- Allow others to reproduce your results
- Reproduce an entire report, manuscript, dissertation, book with a single system command when changes occur in:
  - operating system, stat software, graphics engines, source data, derived variables, analysis, interpretation
- Save time
- Provide the ultimate documentation of work done for a paper

knitr by Yihui Xie, Iowa State University

- Better handling of graphics; no more `print(xyplot())`
- Simplified interface to `tikz` graphics
- Simplified implementation of caching
- More automatic pretty–printing; support for LaTeX `listings` package built–in
- Can specify figure captions in chunk headers along with R graphics parameters
- Easy to include animations in pdf reports
- Chunks can produce multiple plots

- `http://yihui.github.com/knitr`
- `http://cran.r-project.org/web/packages/knitr`
- `http://biostat.mc.vanderbilt.edu/KnitrHowto`

knitr Setup Code to Store Centrally


```
spar <- function(mar=if(!axes)
                 c(2.25+bot-.45*multi,2+left,.5+top+.25*multi,.5+rt) else
                 c(3.25+bot-.45*multi,3.5+left,.5+top+.25*multi,.5+rt),
                 lwd = if(multi)1 else 1.75,
                 mgp = if(!axes) mgp=c(.75, .1, 0) else
                       if(multi) c(1.5, .365, 0) else c(2.4-.4, 0.475, 0),
                 tcl = if(multi)-0.25 else -0.4,
                 bot=0, left=0, top=0, rt=0, ps=if(multi) 14 else 10,
                 mfrow=NULL, axes=TRUE, ...)
  {
    multi <- length(mfrow) > 0
    par(mar=mar, lwd=lwd, mgp=mgp, tcl=tcl, ps=ps, ...)
    if(multi) par(mfrow=mfrow)
  }

render_listings()
unlink('messages.txt') # Start fresh with each run
hook_log = function(x, options) cat(x, file='messages.txt', append=TRUE)
knit_hooks$set(warning = hook_log, message = hook_log)
knit_hooks$set(par=function(before, options, envir)
               if(before && options$fig.show != 'none')
               {
                 p <- c('bty','mfrow','ps','bot','top','left','rt','lwd',
                        'mgp','tcl', 'axes')
                 pars <- opts_current$get(p)
                 pars <- pars[!is.na(names(pars))]
                 if(length(pars)) do.call('spar', pars) else spar()
               })
```

Setup Code, *continued*


```
# Set short aliases for names of commonly used parameters
opts_knit$set(aliases=c(h='fig.height', w='fig.width',
                cap='fig.cap', scap='fig.scap'))
opts_knit$set(eval.after = c('fig.cap','fig.scap'))
## see http://yihui.name/knitr/options#package_options
## Use caption package options to control caption font size
```

Code for Beginning of Report or Chapter


```
<<echo=FALSE>>=
source('...file listed above...')
\SweaveOpts{fig.path='plot-', fig.align='center', w=4.5, h=3.5,
fig.show='hold', fig.pos='htbp', par=TRUE, tidy=FALSE}
@
```

Code for a Chunk


```
<<bigplot,h=7,w=7,cap='A \\textbf{caption} for the figure'>>=
# need to double backslashes to escape them
<<example2,cap=paste('Survival curves for study', study_name)>>=
<<this,results='tex'>>=
# need to put character values in quotes with knitr, unlike Sweave
<<that,ps=6,mfrow=c(2,2)>>=
plot(something)  # Figure (*\ref{fig:xxx-that}*)
[symbolic reference from R to LaTeX]
```

Using a Macro Pre-processor to Factor Out Repetitive Operations

- Consider writing many LaTeX text and R code chunks where few variables change from one section to another
- Key independent or dependent variable that is replaced by another variable, but the remainder of the sentences and R code remain unchanged
- Factor out common text/code
- Need simple syntax for variable name and other substitutions
- Python `pyexpander` pre-processor
  - Arbitrary Python code to manipulate text
  - Must escape `$` using `\$`
  - Define `bash` script in `~/bin` named `pye2r`

```
expander.py --eval $2 $1.pye > $3.Rnw
```

Using pyexpander

- bash script to run pye2r with variable substitutions
- Combines all resulting .Rnw files into one .Rnw file for insertion into master .Rnw file
- Master file has LaTeX and knitr setup along with code chunks not needing to be repeated
- Example: 4 bone mineral density (BMD) measurements, corresponding to four bones, are analyzed in turn
- When one BMD measure is the key predictor, other 3 adjusted for
- Only main analysis variable $v appears in interactions
- Combined processed file allbmd.Rnw is run by knitr inside the master document using \Sexpr{knit_child('allbmd.Rnw')} in a LaTeX chunk

Script maker that runs pye2r


```
pye2r repbmd 'v="hip";oth1="lumbar";oth2="femur";oth3="forearm"' hip
pye2r repbmd 'v="lumbar";oth1="hip";oth2="femur";oth3="forearm"' lumbar
pye2r repbmd 'v="femur";oth1="hip";oth2="lumbar";oth3="forearm"' femur
pye2r repbmd 'v="forearm";oth1="hip";oth2="lumbar";oth3="femur"' forearm
cat hip.Rnw lumbar.Rnw femur.Rnw forearm.Rnw > allbmd.Rnw
rm -f hip.Rnw lumbar.Rnw femur.Rnw forearm.Rnw
```

repbmd.pye: Common Code

- Chunk names prefixed by $v to create unique chunk names that start with hip, lumbar, etc.
- $v is base name of the main variable of interest, and 0, 26, 52 are appended to denote the BMD measure at baseline (time 0), 26w, and 52w.

```
% Usage: ./maker
$begin
$extend(v,oth1,oth2,oth3)    $# allow referencing by $v not just $(v)
$py(v0=v+"0")                $# create new variable with 0 appended
$py(v26=v+"26")
$py(v52=v+"52")
$py(oth10=oth1+"0")
$py(oth20=oth2+"0")
$py(oth30=oth3+"0")
$py(vupper=str.title(v))     $# capitalize first letters
$extend(oth10,oth20,oth30,v0,v26,v52,vupper)
$# allow easy referencing of new var
```

Common Code, *continued*


```
\section{Analysis of $vupper Bone Mineral Density}
\subsection{26w BMD and Baseline Predictors}
. . .
<<$v-sat26,results="asis">>=
f <- ols($v26 ~ sex*rcs($v0,5) + rcs($oth10,5) + rcs($oth20,5) +
         rcs($oth30,5) + sex*rcs(wt0,5) + trtp + rcs(age,5) + race + sex
         blppar + bltscgrp, data=d)
print(f, coefs=FALSE, latex=TRUE)
latex(anova(f), file='', table.env=FALSE)
@
```

But . . .

- Developer plans to implement macro pre-processing inside `knitr`
- This would allow LaTeX sections to appear conditional on earlier computations

Possible Future Extensions

- Tables with thumbnail graphics
  - auto–enlarge to full resolution
- Animations inside pdfs (already in knitr)
- "Live" reports viewed with web browser
  - drill–down to raw data
  - alternate graphics, e.g., ECDF and histogram under box plot

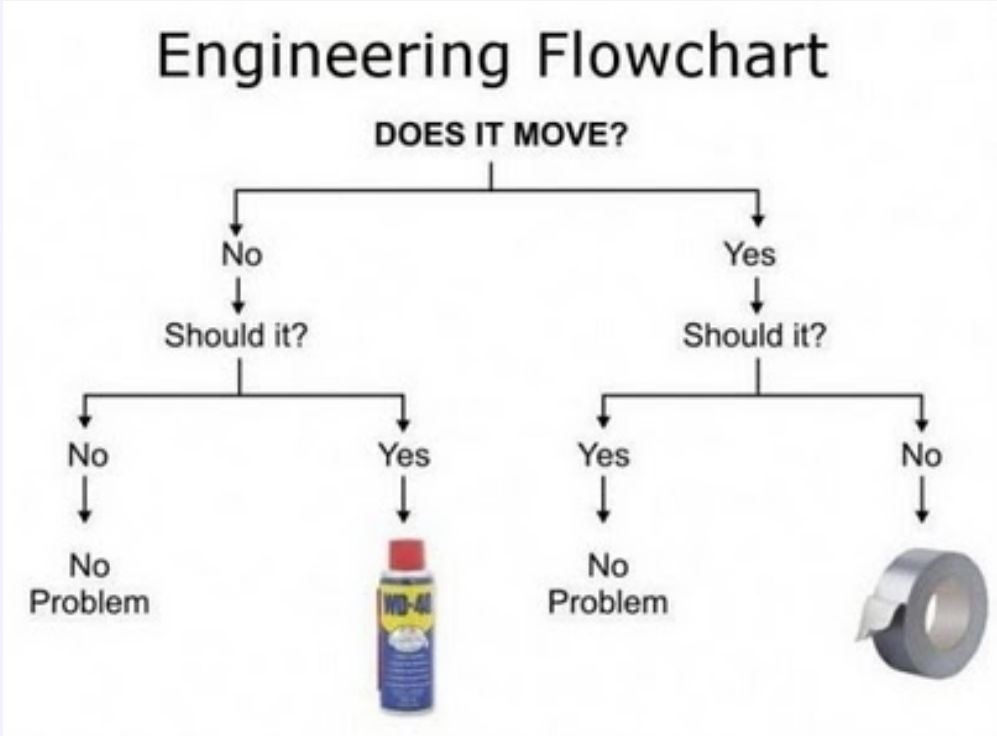
See https://github.com/downloads/yihui/knitr/knitr-graphics.pdf for how to do smooth animations.

New Department Priorities

- Reusable tools for common tasks are widely recognized
  - Example: PS (Dupont & Plummer, 1990)
- One need: extensible power/sample size/precision resource running on every platform
  - Would be enhanced with easy mechanism for community participation using e.g. `Github`
  - Need to be able to quickly implement simple closed–form calculations up to simulators
- Interactive "How to work with a biostatistician" to jump–start collaborations/clinics
- Interactive data preparation (spreadsheet from . . .) and upload

Example: Simulate Power of Two-Sample Survival Test Under Complex Conditions

- R `Hmisc` package `spower` function
- Intervention:control hazard ratio (HR): $h(t)$
  - allows for delayed treatment effect or decay
- Drop-in $f(t)$; drop-out $g(t)$
- User-specified $S(t)$; simplest: 2 t's and $S(t)$'s, Weibull distribution auto-fitted
- Simulation to estimate power of log–rank test and multiplicative margin of error for effective HR estimate



This work used only free software

LaTeX GNU/Linux R

Flowchart from Google+ Technics