Reproducible Statistical Reporting with R, LaTeX, and knitr

Frank E Harrell Jr

Department of Biostatistics
Vanderbilt University School of Medicine

Novartis Biostatistics Conference

East Hanover NJ 2014-10-07

Non-reproducible Research

- Misunderstanding statistics
- "Investigator" moving the target
- Lack of a blinded analytic plan
- Tweaking instrumentation / removing "outliers"
- Pre-statistician "normalization" of data and background subtraction
- Poorly studied high-dimensional feature selection

canstockphoto.com

Non-reproducible Research, *continued*

- Programming errors
- Lack of documentation
- Failing to script multiple-step procedures
  - using spreadsheets and other interactive approaches for data manipulation
- Copying and pasting results into manuscripts
- Insufficient detail in scientific articles
- No audit trail

Goals of Reproducible Analysis/Reporting

- Be able to reproduce your own results
- Allow others to reproduce your results

> *Time turns each one of us into another person, and by making effort to communicate with strangers, we help ourselves to communicate with our future selves. (Schwab and Claerbout)*

- 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

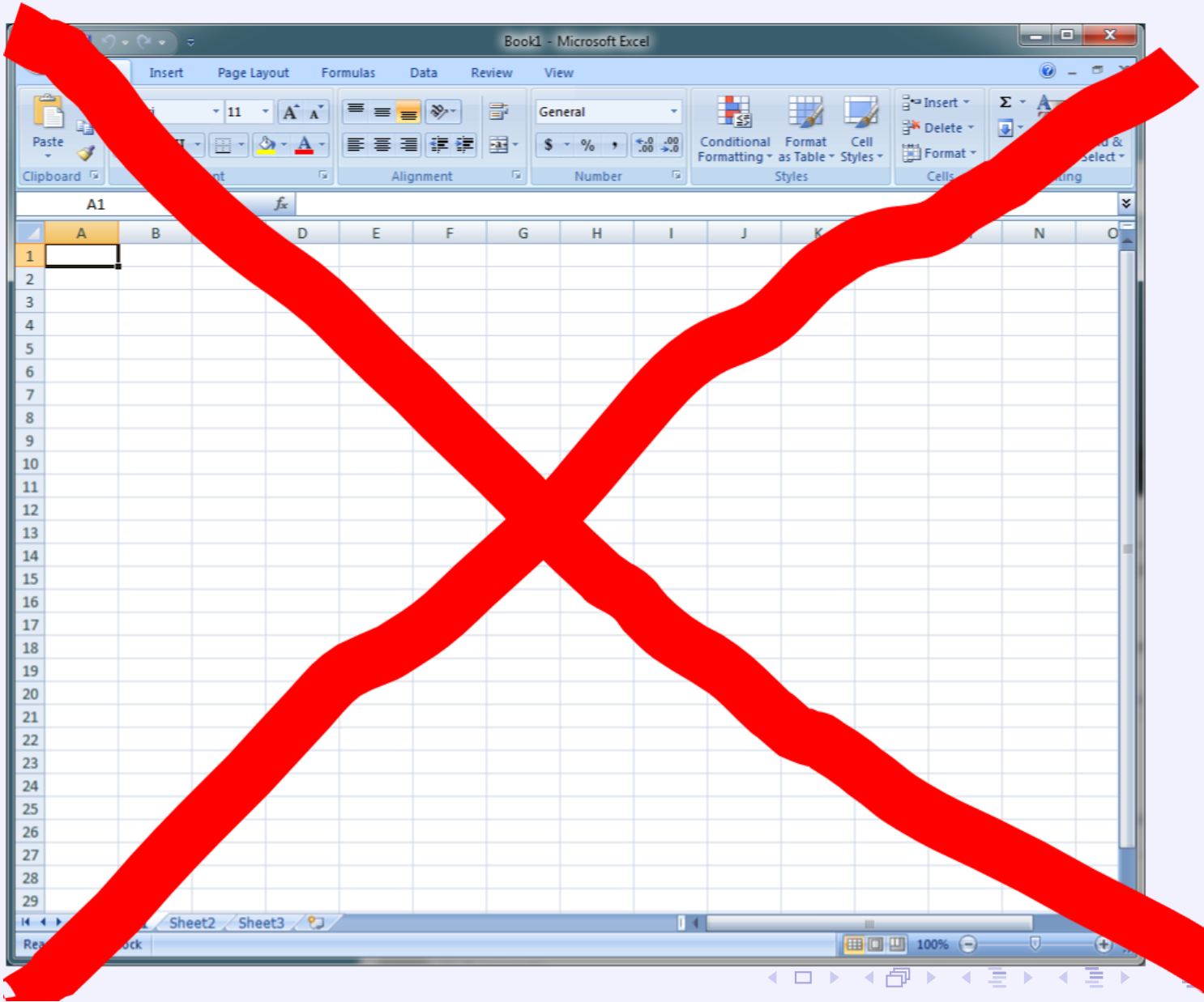
History

- Donald Knuth found his own programming to be sub-optimal
- Reasons for programming attack not documented in code; code hard to read
- Invented **literate programming** in 1984
  - mix code with documentation in same file
  - "pretty printing" customized to each, using TeX
  - not covered here: a new way of programming
- Knuth invented the `noweb` system for combining two types of information in one file
  - *weaving* to separate non-program code
  - *tangling* to separate program code

History, *continued*

- Leslie Lamport made TeX easier to use with a comprehensive macro package LaTeX in 1986
- Allows the writer to concern herself with structures of ideas, not typesetting
- LaTeX is easily modifiable by users: new macros, variables, *if-then* structures, executing system commands (Perl, etc.), drawing commands, etc.
- S system: Chambers, Becker, Wilks of Bell Labs, 1976
- R created by Ihaka and Gentleman in 1993, grew partly as a response to non-availability of S-Plus on Linux and Mac
- Friedrich Leisch developed `Sweave` in 2002
- Yihui Xie developed `knitr` in 2011

A Bad Alternative to knitr



knitr Approach

- knitr is an R package on CRAN
- Uses noweb and an sweave style in LaTeX
- knitr also works with Markdown and other languages
- knitr is tightly integrated into RStudio

yihui.name/knitr, Xie [2013], http://yihui.github.com/knitr

knitr Approach, *continued*

- *Insertions* are a major component
  - R printout after code chunk producing the output; plain tables
  - single `pdf` or `postscript` graphic after chunk, generates LaTeX `includegraphics` command
  - direct insertion of LaTeX code produced by R functions
  - computed values inserted outside of code chunks
- Major advantages over Microsoft Word: composition time, batch mode, easily maintained scripts, beauty
- `knitr` produces self-documenting reports with nice graphics, to be given to clients
  - showing code demonstrates you are not doing "pushbutton" research

Some knitr Features

- R code set off by lines containing only <<>>=
- LaTeX text starts with a line containing only @
- knitr senses when a chunk produces a graphic (even without *print()* and automatically includes the graphic in LaTeX
- All other lines sent to LaTeX verbatim, R code and output sent to LaTeX by default but this can easily be overridden
- Can specify that a chunk produces markup that is directly typeset; this is how complex LaTeX tables generated by R

Some `knitr` Features, *continued*

- Can include calculated variables directly in sentences, e.g. `And the final answer is \Sexpr{sqrt(9)}.` will produce "And the final answer is 3."
- Easy to customize chunk options and add advanced features such as automatically creating a LaTeX `figure` environment if a caption is given in the chunk header
- Setup for advanced features, including code pretty-printing, shown at `http://biostat.mc.vanderbilt.edu/KnitrHowto`

Some `knitr` Features, *continued*

- Simplified interface to `tikz` graphics
- Simplified implementation of caching
- More automatic pretty–printing; support for LaTeX `listings` package built–in

Running knitr from Command Line

A useful Linux/Unix script if you use `.Rnw` as the suffix. Produces a popup progress window using `xterm`.

```
#!/bin/sh
rm -f messages.txt
xterm -T "‘pwd‘/$1.Rnw" -hold -e R --no-save --no-restore \
 -e "require(knitr); knit('$1.Rnw')"
echo PDF graphics produced:
ls -lgt *.pdf
```

Execute using `knitr my` to run `my.Rnw` and produce `my.tex` etc., then run `pdflatex my` or `latex my`.

Easier: `RStudio`

There are utility functions for extracting just the R output or just the LaTeX text

Enhancing Output

- Graphics size and quality suitable for publication using `hooks`
- Customizing the LaTeX `Sweave.sty` style macro
- Pretty printing of code and output, with shaded boxes
- Direct insertion of LaTeX code created by R functions
  - Allows complex tables with micrographics
- Selectively suppressing parts of R output
- Comments in R code containing symbolic references to LaTeX sections
- Auto-documenting R and package versions used
- Floating figures & captions
- See the `knitrSet` function at `http://biostat.mc.vanderbilt.edu/KnitrHowto` that makes all this easy

Code for Beginning of Report or Chapter

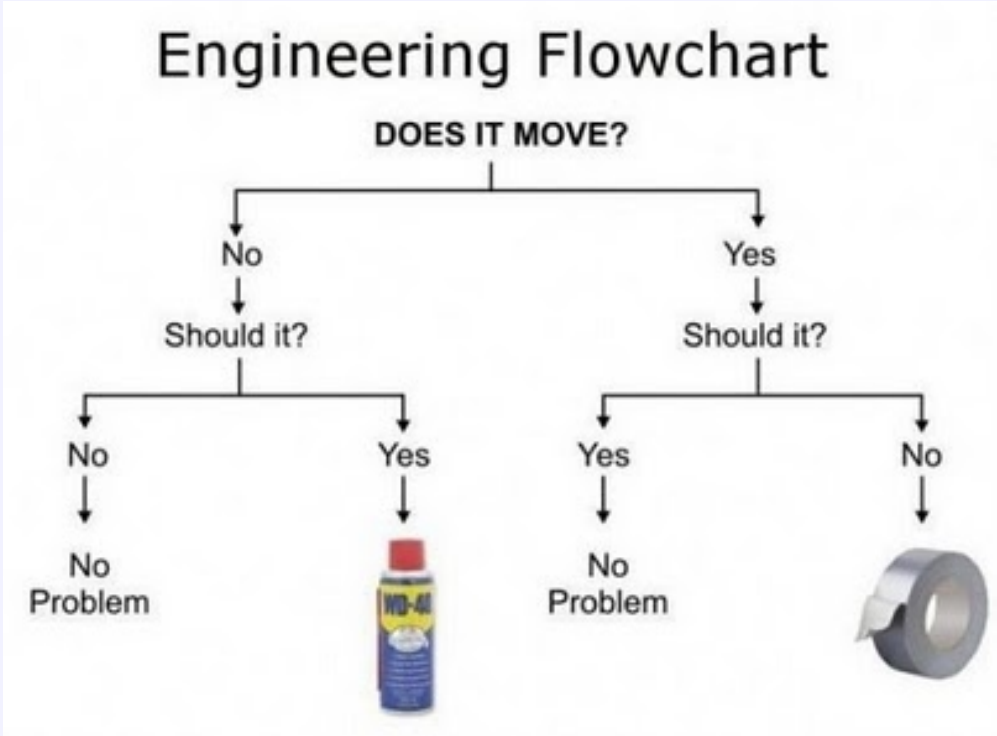

```
<<echo=FALSE>>=
# Default figure size 4.5" by 3.5"
knitrSet(w=4.5, h=3.5)
@
```

Several more parameters can be given to `knitrSet` to set default figure centering, floating figure position, caching, code echo, etc.

Code for a Chunk


```
<<bigplot,h=7,w=7,cap='A \\textbf{caption} for the figure'>>=
# need to double backslashes to escape them
# produces bigplot.pdf in pdf subdirectory
<<example2,cap=paste('Survival curves for study', study_name)>>=
<<this,results='tex'>>=
# for a chunk that produces LaTeX markup
# 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]
```

cap is the figure caption, and scap can be given for a short caption to appear in the table of figures. ps is pointsize, mfrow is for making a matrix of figures using base graphics. Many other parameters are passed to the spar function defined on the KnitrHowto wiki.



This work used only free software

LaTeX GNU/Linux R

Flowchart from Google+ Technics

References

Y. Xie. *Dynamic Documents with R and knitr*. Chapman and Hall, 2013. ISBN 978-1482203530.

Reproducible Research

Frank E Harrell Jr
Department of Biostatistics
Vanderbilt University School of Medicine
Nashville TN

Much of research that uses data analysis is not reproducible. This can be for a variety of reasons, the most major one being poor design and poor science. Other causes include tweaking of instrumentation, the use of poorly studied high-dimensional feature selection algorithms, programming errors, lack of adequate documentation of what was done, too much copy and paste of results into manuscripts, and the use of spreadsheets and other interactive data manipulation and analysis tools that do not provide a usable audit trail of how results were obtained. Even when a research journal allows the authors the "luxury" of having space to describe their methods, such text can never be specific enough for readers to exactly reproduce what was done. All too often, the authors themselves are not able to reproduce their own results. Being able to reproduce an entire report or manuscript by issuing a single operating system command when any element of the data change, the statistical computing system is updated, graphics engines are improved, or the approach to analysis is improved, is also a major time saver.

It has been said that the analysis code provides the ultimate documentation of the "what, when, and how" for data analyses. Eminent computer scientist Donald

Knuth invented literate programming in 1984 to provide programmers with the ability to mix code with documentation in the same file, with "pretty printing" customized to each. Lamport's LaTeX, an offshoot of Knuth's TeX typesetting system, became a prime tool for printing beautiful program documentation and manuals. When Friedrich Leisch developed Sweave in 2002, Knuth's literate programming model exploded onto the statistical computing scene with a highly functional and easy to use coding standard using R and LaTeX and for which the Emacs text editor has special dual editing modes using ESS. This approach has now been extended to other computing systems and to word processors. Using R with LaTeX to construct reproducible statistical reports remains the most flexible approach and yields the most beautiful reports, while using only free software. One of the advantages of this platform is that there are many high-level R functions for producing LaTeX markup code directly, and the output of these functions are easily directly to the LaTeX output stream created by Sweave.

See ctspedia.org, reproducibleresearch.net, groups.google.com/group/reproducible-research, and biostat.mc.vanderbilt.edu/SweaveLatex for more information.