Reproducible Research with R, LaTeX, & Sweave

Theresa A Scott, MS

Department of Biostatistics
theresa.scott@vanderbilt.edu
http://biostat.mc.vanderbilt.edu/TheresaScott

This lecture. . .

▷ Learning objectives:

- To understand the concept & importance of reproducible research.
- To understand the role of each software component in the automatic generation of statistical reports.
- To understand how to generate a reproducible statistical report from scratch.

▷ Outline:

- A common (flawed) approach for generating statistical reports.
- A (better) alternative approach.
- How to generate reproducible statistical reports using R, LaTeX, & Sweave.
- Some additional information.

Section I:

A common (flawed) approach for generating statistical reports

Typical steps leading up to the reporting

▷ FIRST,

- Data entry & storage.
- Data cleaning (including checking for, resolving, & correcting data entry errors).
- Data preparation (including transforming/recoding variables, creating new variables, & creating necessary subsets).
- Performing the proposed statistical analyses, including generating desired graphs.
- Recording/saving the desired results/graphs.

▷ FINALLY,

- Writing a results report, which may include documentation text, tables and/or graphs.

'Common' approach: write report around results

▷ First, **POINT & CLICK**

- Use Microsoft (MS) Excel for data entry/cleaning/preparation, & possibly statistical analyses.[1]
- Possibly import the data into SPSS (point & click statistical software package) for data preparation & statistical analyses.
- Possibly use MS Excel to record/format the desired results & generate the desired graphs

▷ Then, **COPY & PASTE/TYPE BY HAND**

- Take advantage of pre-formatted tables & graphs generated by many statistical software packages, like SPSS.
- Copy & paste/type by hand desired results (text, tables, graphs) from data analysis system to a word processor (eg, MS Word).

[1] *BAD IDEA*: Handling of missing data; poor algorithms & unreliable results – see lecture. Okay for data entry.

Problems with 'common' approach

▷ **VIGNETTE 1**: You sit down to finish writing your manuscript. You realize that you need to clarify one result by running an additional analysis. You *first* re-run the primary analysis. Major problem: the primary results don't match what you have in your paper.

▷ **VIGNETTE 2**: When you go to your project folder to run the additional analysis, you find multiple data files, multiple analysis files, & multiple results files. You can't remember which ones are pertinent.

▷ **VIGNETTE 3**: You've just spent the week running your analysis & creating a results report (including tables & graphs) to present to your collaborators. You then receive an email from your PI asking you to regenerate the report based on a subset of the original data set & including an additional set of analyses – she would like it by tomorrow's meeting.

Problems with 'common' approach, *cont'd*

▷ With point & click programs (eg, MS Excel or not using SPSS's log), no way to record/save the steps performed that generated the documented results.

▷ Common to keep analysis code, results, & reports as separate files & to save various versions of each of these as separate files.

- After several modifications of one or more of the files involved, becomes unclear which version of the files *exactly* correspond to the desired analysis & results.

▷ Every time analyses and/or results change, have to *re*generate the results report *by hand* – very time consuming.

▷ Very easy for human error to creep into results report (eg, typing in results by hand, copying/pasting the wrong tables/graphs).

Section II:

A (better) alternative approach

Alternative to 'common' approach

▷ First, use **R** instead of Excel/SPSS for data cleaning/prep & statistical analyses (including graphs).

- R is a *programming language* – removes point & click.
- R is *free* to run, study, change, & improve.
- R runs on Windows, MacOS, Linux & UNIX platforms.
- R has publication quality *graphing capabilities*.
  - Able to generate typical statistical plots (eg, scatterplots, boxplots, & barplots).
  - Follows a 'build your plot in layers' framework – later graphical output (possibly) obscures previous output that it overlaps.
  - Allows you to add additional information (such as points, lines, or a legend) to an existing plot to make it more customized.
  - Also allows you to create a plot 'from scratch' when no existing plot provides a sensible starting point.

Alternative to 'common' approach, *cont'd*

▷ Then, use **LaTeX** instead of MS Word for writing the report.

- LaTeX is a document preparation system, *not* a word processor.
  - Rather than type words & then format them using drop-down menus, the formatting is part of the text (specified using commands).
  - *Saves you time*.
- LaTeX contains features for
  - (1) automatic formatting of title pages, section headers, headers/footers, & bulleted/ enumerated lists;
  - (2) cross-referencing of sections, tables, & figures;
  - (3) typesetting of complex mathematical formulas;
  - (4) creating tables & inserting graphs; &
  - (5) automatic generation of bibliographies & indexes (eg, table of contents).

Alternative to 'common' approach, *cont'd*

▷ **A PROBLEM REMAINS**: Have removed point & click with R & have saved time spent formatting with LaTeX, but still haven't removed the need to copy & paste results and/or type them by hand.

▷ **BETTER APPROACH:** Embed the analysis into the report.

- That is, embed the R code to clean/prep the data & to perform the desired statistical analysis into the LaTeX document that contains the documentation text of the report.

▷ Possible using a tool called **Sweave**.

- Actually, a *function* in R.
- Called 'S-weave' not 'R-weave' because R is an implementation of the *S* language.
- Created by Friedrich Leisch, PhD.

Better approach: using Sweave

▷ When the 'weaved' document is run through Sweave all of the data analysis output (including text, tables & graphs) is created on the fly & inserted into the LaTeX report document.

- No longer need to copy & paste results and/or type them by hand.

▷ The statistical report is now completely *reproducible*.

- Allows for truly **reproducible research**.

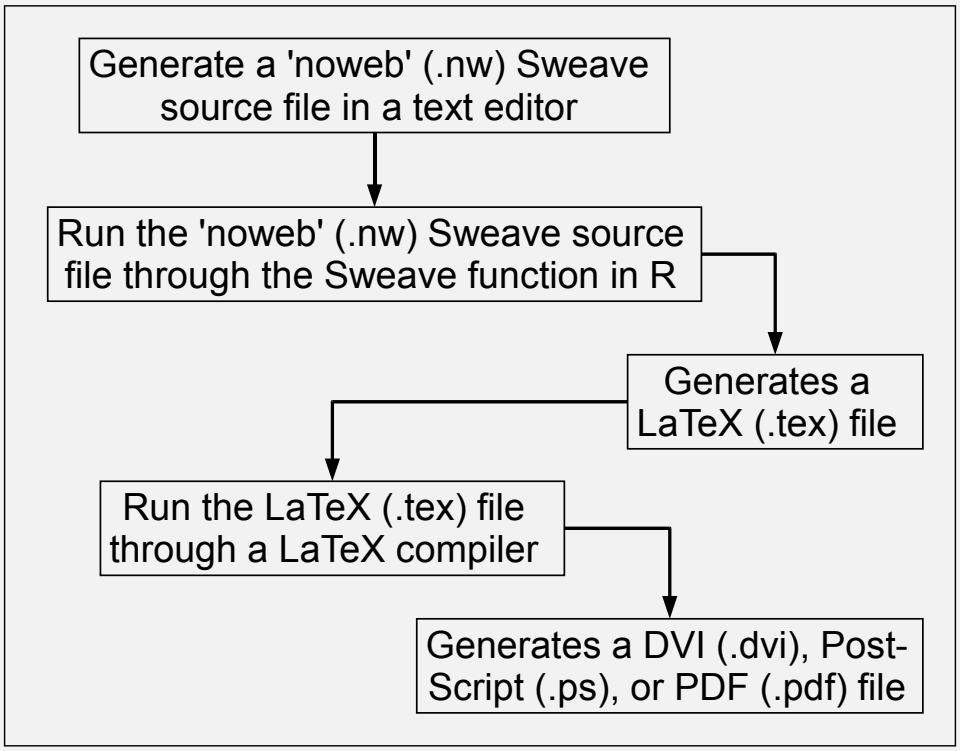
▷ Also, the report is now *dynamic*.

- Can be easily regenerated when the data or analyses change – all of the results/tables/figures are automatically updated.

Section III:

How to generate reproducible statistical reports using R, LaTeX, & Sweave

Diagram of process



'Noweb' (`.nw`) Sweave source file

▷ **'Noweb'**: *literate-programming* tool; allows you to combine program source code & corresponding documentation into single file.

▷ **Sweave source file**: a text file which consists of a sequence of R code & LaTeX documentation segments called *chunks*:

- **LaTeX documentation chunks** start with a line that has only an `@` ('at') sign.
  - Default for the first chunk is documentation – no @ sign needed.
- **R code chunks** start with a line that has only <<>>=.
  - <<>>= syntax can be modified to have additional control.
- **IMPORTANT**: Because the Sweave source file is a pre-cursor to a LaTeX document it must also include the *file structure items* necessary for a LaTeX document.
- Created in any text editor (eg, Notepad) & saved to relevant project folder/directory (eg, where data files are located).

Simple example: `example.nw`

```
\documentclass[12pt]{article}
\usepackage[margin=1.0in]{geometry}
\title{Sweave Example}
\author{Jane Doe, MS}
\begin{document}
\maketitle

\section{Analysis \& Results}
The \texttt{mtcars} (`Motor Trend Car Road Tests') data set is
comprised of 11 aspects of automobile design and performance
(columns) for 32 automobiles (rows).  We wish to know if there
is a significant difference in the quarter mile track times
(\texttt{qsec}) between the different cylinder classes
(\texttt{cyl}; 4, 6, and 8).

<<>>=
data(mtcars)
names(mtcars)
with(mtcars, tapply(X = qsec, INDEX = list(cyl),
   FUN = median, na.rm = TRUE))
with(mtcars, kruskal.test(qsec ~ cyl))$p.value
@

\end{document}
```

LaTeX file structure items

1st LaTeX documentation chunk

R code chunk

Return to a LaTeX documentation chunk

LaTeX file structure item

.nw file must end with a single blank line!

'Sweaving' the `.nw` file

▷ At the R command line prompt ('>'), execute the Sweave function by specifying a single argument – the name of the `.nw` file.

- Example: `Sweave("example.nw")`
  - File name specified in quotes & must include extension (`.nw`).
- IMPORTANT: The R session's 'working directory' must be the folder/directory in which the `.nw` file is located – see R lectures.
- Will receive screen output:
```
Writing to file example.tex
Processing code chunks...
```
- If all goes well, will receive the screen output
```
You can now run LaTeX on 'example.tex'
```
& a new command line prompt.
  - `.tex` LaTeX file is created in same folder/directory as `.nw` file.
- If error occurs, will be told which code chunk error occurred in – referenced by number (1, 2, . . . ; <<>>= counted).

What changes from the `.nw` to the `.tex` file

```
\documentclass[12pt]{article}
\usepackage[margin=1.0in]{geometry}
\title{Sweave Example}
\author{Jane Doe, MS}
\usepackage{.../Sweave}
\begin{document}
\maketitle

\section{Analysis \& Results}
The \texttt{mtcars} (`Motor Trend Car Road Tests') data set is comprised of 11 aspects of
automobile design and performance (columns) for 32 automobiles (rows).  We wish to know if
there is a significant difference in the quarter mile track times (\texttt{qsec}) between
the different cylinder classes (\texttt{cyl}; 4, 6, and 8).

\begin{Schunk}
\begin{Sinput}
> data(mtcars)
> names(mtcars)
\end{Sinput}
\begin{Soutput}
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear" "carb"
\end{Soutput}
\begin{Sinput}
> with(mtcars, tapply(X = qsec, INDEX = list(cyl), FUN = median, na.rm = TRUE))
\end{Sinput}
\begin{Soutput}
     4      6      8
18.900 18.300 17.175
\end{Soutput}
\begin{Sinput}
> with(mtcars, kruskal.test(qsec ~ cyl))$p.value
\end{Sinput}
\begin{Soutput}
[1] 0.006234986
\end{Soutput}
\end{Schunk}

\end{document}
```

Reference to Sweave style file added*; otherwise, LaTeX input file structure items unmodified

LaTeX documentation chunk unmodified

R code chunk executed – both the R commands and their respective output have been transferred, embedded in `Sinput` and `Soutput` environments, respectively

LaTeX input file structure item unmodified

*: provides environments for typesetting R (S) input/output; exact path (...) will be different on your computer

Compiling the `.tex` file

▷ On a Linux/Unix machine:

- Open a Terminal shell & using the `cd` command, move to the relevant working directory (where the `.nw`/`.tex` files are saved).
- To create a PDF (`.pdf`) file[2], execute the `pdflatex` command at the command prompt – eg, `pdflatex example`
  - *Do not* need to specify `.tex` extension.
  - `.pdf` file created in same directory as the `.nw`/`.tex` files.
- Often necessary to compile the `.tex` file *twice*[3] – use `&&`
  - Example: `latex example && latex example`
- If all goes well, will be returned to a new command prompt.
- Other options: R's `system()` function or a shell script (see Sweave manual FAQ A.3).

[2] Use the `latex` command to create a DVI (`.dvi`) file.

[3] For elements like a table of contents & cross-referencing (ie, section, table, & figure labeling)

Compiling the `.tex` file, *cont'd*

▷ On a Windows/Mac machine:

- Use MikTeX (free @ `http://miktex.org`).
  - May have problem referencing Sweave style (.sty) file because of the *space* in the 'Program Files' folder name – see Sweave manual (FAQ A.12) for solution.
- Can also use a text editor like WinEdt, which by default is already configured for MikTeX – point & click capabilities.
  - Free @ `http://www.winedt.com/`; MikTeX must be installed.
- If no WinEdt:
  - Open a Terminal shell by clicking on 'Run' from the 'Start' menu & typing 'C:/command' (or 'cmd').
  - Using the `cd` command, move to the relevant working directory.
  - Use commands similar to `latex` and `pdflatex`.[4]

[4] Usually not necessary to compile the `.tex` file twice – MikTeX compiles as many times as necessary.

Final output – PDF (`.pdf`) results report

NOTE: PDF file has been cropped

Sweave Example

Jane Doe, MS

May 6, 2008

1 Analysis & Results

The `mtcars` ('Motor Trend Car Road Tests') data set is comprised of 11 aspects of automobile design and performance (columns) for 32 automobiles (rows). We wish to know if there is a significant difference in the quarter mile track times (`qsec`) between the different cylinder classes (`cyl`; 4, 6, and 8).

```
> data(mtcars)
> names(mtcars)

 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
[11] "carb"

> with(mtcars, tapply(X = qsec, INDEX = list(cyl), FUN = median,
+     na.rm = TRUE))

     4      6      8
18.900 18.300 17.175

> with(mtcars, kruskal.test(qsec ~ cyl))$p.value

[1] 0.006234986
```


Modifying R code chunk output – <<>>= options

▷ Named 'flags' (separated by commas) can be specified within the <<>>= R code chunk header to pass options to Sweave, which control the final output.

- `echo` flag: value indicating whether to include (`true`) or not include (`false`) the R *code* (commands) in the output file.
- `results` flag: value indicating whether to include (`verbatim`) or not include (`hide`) the results of the R code (ie, what is normally printed to the screen) in the output file.
- When just <<>>= is specified, Sweave implements the *default* values of the `echo` & `results` flags – as we saw, both the R code & its results are included in the output file.
  - <<>>= is equivalent to <<`echo = true, results = verbatim`>>=.
- Often use <<`echo = false, results = hide`>>= for R code chunks that contain data input, cleaning, & preparation steps.

<<>>= options, *cont'd*

▷ Can generate *tables* using a `results = tex` flag.

- R code chunk contains the code that generates the LaTeX syntax to create a table.
  - LaTeX syntax is inserted in the `.tex` file; the table is created when the `.tex` file is compiled.
- LaTeX syntax generating functions available from the `Hmisc` & `xtable` add-on packages[5] – `latex()` & `xtable()`/ `print.xtable()` functions, respectively.
  - Contain arguments to specify formatting of the table, table caption (for 'List of Tables'), & cross-referencing.
- `\usepackage{}` statements (additonal LaTeX file structure items) often needed – see additional example posted on website.

[5] Must be *installed & loaded* – see R lectures

<<>>= options, *cont'd*

▷ Can insert generated *graphs* using a `fig = true` flag.

- R code chunk contains the code that generates the graph.
  - IMPORTANT: R code must generate *only one* figure.
- An EPS & PDF file of the graph are created & saved (by default) to same folder/directory as `.nw` file.
  - Can be saved in a sub-folder/directory – see Sweave manual.
- An `\includegraphics{}` statement is inserted in the `.tex` file, which inserts the saved file when the `.tex` file is compiled.
- By default, no caption is given to inserted graph – causes graph not to be listed in 'List of Figures'.
  - Solution: Wrap R code chunk with `fig = true` flag with `\begin{figure}` & `\end{figure}` environment & a corresponding `\caption{}` statement.
- More in Sweave manual FAQ A.4 - A.11 & Section 4.1.2.

Embedding R code in a LaTeX sentence

▷ Often wish to incorporate a value calculated using R into a LaTeX documentation sentence.

- Can do this using `\Sexpr{`*expr*`}`, where *expr* is R code.
- Example: 'The mean quarter mile track time of the N =

  `\Sexpr{nrow(mtcars)}` cars included in the mtcars data set was

  `\Sexpr{round(mean(mtcars$qsec, na.rm = TRUE), 1)}` seconds.'

  evaluates to 'The mean quarter mile track time of the N = 32 cars included in the mtcars data set was 17.8 seconds.'

▷ The `\Sexpr{}` cannot break over many lines & must not contain curly brackets ({ }).

- More complicated/lengthy expressions can be easily executed & assigned as an object in a *hidden* code chunk & then the assigned object referenced inside the `\Sexpr{}`.

Section IV:

Some additional information

What to do. . .

▷ *When you get an error in the Sweave step*: check R code chunks.

- Recall, will be told in which code chunk the error occurred.
- Check to make sure every R code chunk begins with a <<>>= (with possible flags) & ends with an `@` sign.[6]

▷ *When you get an error in the LaTeX compile step*: check LaTeX documentation chunks & `.tex` file.

- Error could be caused by output inserted in the `.tex` file via a `\Sexpr{}` expression or a `results = tex` flag.
- Comment out LaTeX documentation chunks and/or *whole* R code chunks (from <<>>= to `@`) in `.nw` file using % signs.

▷ *Whenever* `.nw` *file or data file changes*: re-run Sweave step on the (modified & saved) `.nw` file & re-compile resulting `.tex` file.

[6] Even though `@` sign is technically a *header* for a LaTeX documentation chunk, think of it as a *footer* for an R code chunk.

Useful tips/recommendations

▷ Work out details of R code within an R session & then copy & paste correct code to an R code chunk within the `.nw` file.

▷ On a Windows machine, show all file extensions – uncheck the 'Folder Option' to 'Hide file extensions for known file types'.

▷ On a Linux/Unix machine, use `Kate` or `ESS` (Emacs Speaks Statistics) as your text editor; on a Windows machine, use WinEdt.

▷ When you start a new R session,

- (1) Use the `Stangle()` function to extract all of the R code chunks from the `.nw` file & write them to a `.R` code file.
- (2) Use the `source()` function to read in the `.R` code file & execute the R code chunks.
- Allows you to quickly execute all the R code chunks without having to copy/paste from `.nw` file to the R command prompt.

Resources & references

- Today's material (including *extended* Sweave example):
  - http://biostat.mc.vanderbilt.edu/TheresaScott
- R:
  - R lectures (http://biostat.mc.vanderbilt.edu/TheresaScott)
  - Weekly R Clinic: Thursdays 2:00 - 3:00 PM in MCN D-2221; also cover LaTeX & Sweave; http://biostat.mc.vanderbilt.edu/RClinic
- LaTeX:
  - 'The Not So Short Intro to LaTeX' document (http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf)
  - Documentation, tutorials, etc written by CTAN (http://www.ctan.org) & TUG (http://www.tug.org)
- Sweave:
  - The Sweave manual (includes *great* FAQs) & examples (http://www.ci.tuwien.ac.at/~leisch/Sweave/)