Applied Biostatistics

https://moodle.epfl.ch/course/view.php?id=15590

- Course organization
- Reproducible Research
- Hypothesis testing - review of basic notions

Organisation

- Instructor : *Darlene Goldstein* (me))
- Course meeting time : Monday 8.15 – 10.00, CM 1 120
- Lab/Exercice session : Meeting lab time Tuesday 16.00-18.00 (zoom)
- Course note :
  - 1 short report ~ 3-5 pages (1/6) ; can be done in groups of 1-4 persons
  - 1 article review~ 1-2 pages (1-1/2/6) ; can be done in groups of 1-4 persons
  - 1 longer ***individual*** report~ 5-7 pages (4/6); data analysis report
- Software : R Statistical Software. http://cran.r-project.org/

Reproducible research principle

- Claerbout : ‘An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.’
- Wavelet community, Stanford University
  - Buckheit and Donoho : ‘When we publish articles containing figures which were generated by computer, we also publish the *complete software environment* which generates the figures.’
- Anecdotes
  - ‘Final’ versions of figs for publication
  - Lost or stolen work
  - Communication
  - Applying old/existing methods on new data
  - Reconstructing work of others

Steps leading to a report

- Data entry and storage
- Data cleaning – check, resolve, correct data entry errors
- Prepare data for analysis – transform/recode variables, create new variables, *etc*.
- Carry out statistical analyses
- Save desired results/graphs
- Write the results report, which may include *documentation text, tables and/or graphs*

Report preparation

- A common approach is to write the report around the results
- Results commonly obtained via ‘point and click’ approach (*e.g.* MS Excel, SPSS,)
- Then copy/paste or – worse – type by hand the results into the word processor used to create the report

NOT A GOOD METHOD – DON’T DO THIS ! ! ! ! :

- no documentation on how the results were obtained, how missing data are handled, *etc*.
- unreliable results

Problems with this approach : examples

- You need to run an additional analysis ; when you re-run the primary analysis, the *results don't match* what you have in your manuscript
- You go to the project folder to run additional analyses and find *multiple* data files, multiple analysis files, multiple results files and can't remember which ones are relevant
- You have spent a week running your analysis and creating a results report (including tables and 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 and including an additional set of analyses – AND she would like it by tomorrow's meeting ! !

Problems with this approach : specifics

- With point and click programs, *no way to record/save* the steps that generated the documented results
- Common to keep analysis code, results, reports as separate files and save various versions of each of these separately ; after several modifications, *unclear which version* corresponds to the desired analysis/results
- Every time analyses and/or results change, have to regenerate the results report by hand – *wastes time* ! !
- Easy to introduce *human error* into report – typing in results by hand, copying/pasting the wrong tables/graphs, *etc.*

Research practice

- *Discipline* in software building
- From the start, *expect* it to be made available to others as part of the publication of their work
- *Avoid copy/paste/editing* in a way that is not
- reproducible (Also think in terms of program re-use)

Literate Programming

- Combining the use of a text formatting language (such as TeX) and a conventional programming language (like C or R) so as to maintain documentation and source code together, the art of writing computer programs for the human reader
- may use *inverse comment convention*
- A kind of literate programming where the program code is marked to distinguish it from the text, rather than the other way around as in normal programs:
- Literate programming paradigm :
  - parse the source document and separate code from narrative
  - execute source code and return results
  - mix rsults from the source code with the original narrative

WEB (not www)

- WEB (Donald Knuth), noweb (Norman Ramsey)
- a WEB system consists of two processors, called *WEAVE* and *TANGLE*
  - WEAVE “weaves” the document for a human reader, producing TeX output
  - TANGLE “tangles” the document for a computer, producing a plain programming language file to be compiled, linked and executed
- WEB (and variants) are not the only environments for Literate Programming
- We will focus on using knitr with R

Good/ bad practices (1)

- Manage all source files under the *same directory* and use *relative path names* whenever possible – absolute paths can break code/reproducibility
- *Do not* change the working directory after computing started ; if necessary, set at *beginning* of R session, and if absolutely unavoidable then *restore* the directory later
- Compile documents in a *'clean' R session* : existing objects in a current session may contaminate the code
- (OK to do interactive data analysis while checking results for code chunks, but at end, compile report in batch mode with a new R session so that all results are freshly generated from code)

Good/bad practices (2)

- Avoid commands that need *human interaction*, since human input can be unpredictable (and therefore not reproducible) ; instead, explicitly code for the required input
- Avoid environment variables for data analysis ; if you need to set up options, do it *inside* the source document
- Attach `sessionInfo()` and instructions on how to compile the document

Barriers to reproducible research

- Huge data
- Data confidentiality issues
- Software version and configuration – changing versions/ availability
- Competition

Tools in R

- CRAN Task Views : https://cran.r-project.org/web/views/
- Reproducible research in R : https://cran.r-project.org/web/views/ReproducibleResearch.html
- Compendium concept
  - dynamic document
  - data
  - auxiliary software

Editor

- Could use *ANY* text editor with the knitr package, since the documents are *plain text files*
- Special text editors are *more useful* :
  - input R code chunks more easily
  - more convenient to call R and knitr to compile source documents to pdf/html within an editor, as well as sending R code chunks to R from within the editor directly

  Several editors available, *e.g.* :
-
  - RStudio – has the most comrehensive support for knitr (and Sweave)
  - LyX – front end for LaTeX with a GUI to help with document writing
  - Emacs/ESS (Emacs Speaks Statistics) – supports statistical software packages, including R