**R-coding-requirements**

statistical computing can benefit immensely from following all the best practices from software development: source control, documentation, a project plan, scope document, bug tracking/change control, etc.

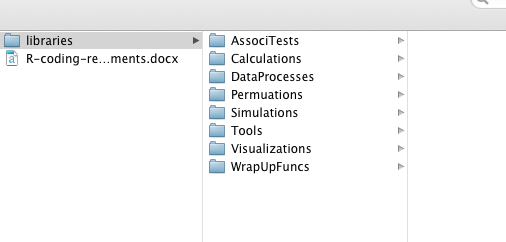
**ALWAYS** do a version control check in right before running results that go out to others, which will assure you know the EXACT state of the code when you produced those results.

1. Follow the Google R Style Guide: <https://google.github.io/styleguide/Rguide.xml>

Advance: Use Sweave to document R code by its math.

2. Organization of R codes according to a hierarchical structure of workflows / working modules / function library files / functions.

2.1 Write your R codes using functions following certain libraries. For example:

* + Data processing library: functions for data cleaning, quality controlling, converting raw data into the data format/structure needed to use, retrieving data from database, etc.
  + Tests library: functions for calculating test statistics.
  + Simulation library: functions for generating simulation data.
  + Permutation library: functions for generating permutation results?
  + Visualization library: functions for visualizing data and results, such as drawing graphs.
  + Calculation library: functions for (numerical) calculations of power, sample size, other meaningful measurements, etc.
  + Tool library: Functions for general purpose of data process.
  + Wrap-up functions: Call other functions to realize specific task(s).
  + Libraries are organized by folders, e.g.:
* Naming library files: libName\_title\_author\_date.R
* List R libraries needed.
* A "main" file that calls the functions to realize data analysis
  + The simpler the better; major coding should be distributed to functions again.
  + **Explicitly define the paths** (for input data folder, output folder, function sources, etc.)

R function writing guideline:

All function much contain the following commenting format, even if keeping some of them empty in the beginning of the writing. (Following the standard R function help file, these components are needed for make R packages.)

* + - **Description**: What the function is about
    - **Arguments**: Input data and controlling options. Short explanation of the requirements for the input (or say "requirements see Details below").
    - **Value**: Output of the function
    - **Author(s)**:
    - **Details**: Detailed explanation for the requirements; explanation of the method; <How is the function realized>
    - **Note**: Other notes <Any comments on this function>
    - **Depends on**: R libraries and functions that this function depends on. (optional)
    - **References**: The function is based on what paper (your paper).
    - **See Also**: similar functions (optional)
    - **Examples**: examples on how the function can be used.
* Principles of writing R functions
  + Easy to understand when reading the comments and the code
  + Easy to use: the input data format is general; put the controlling parameters as options (you can set defaults)
  + Function body: the shorter the better!

Use the R libraries of Dr. Wu’s:

* You can download and recourse all the files and use the functions, but do not change any existing files.
* Write your own functions (and if necessary improve existing functions) in your own files named libName\_title\_author\_date.R

Guidance on professional R programming: <http://www.r-statistics.com/tag/code-management/>