Readability- use a standard code style

1. Folder structure for projects
2. File structure for scripts
3. Code layout for scripts
   1. Indentation
   2. Encapsulation-curly braces
   3. Vertical and horizontal white space
   4. Line wrapping and line length
4. Naming

A workspace holds projects from the same study or related studies

A workspace can have one or more projects

Each project directory has the following structure:

1. Input datasets
   1. Raw
      1. Data (anonymized)
      2. Metadata
   2. Clean
      1. Data
      2. Metadata
2. Scripts
   1. Stata
   2. R
   3. Batchscript
   4. Python
   5. SAS
3. Intermediate datasets(for analysis)
4. Outputs
   1. Tables
   2. Plots
   3. Logs
   4. Reports

There is a special private workspace that holds identifiable information and only accessible to authorized persons

Identifier workspace

* + Project
    - Raw data
    - Anonymized data
    - Link file
    - Anonymization script (generates link file and anonymized raw data)

Script names: based on what the script does and in Camel Case e.g. CleanDictionary.do

The suffix depends on the type of script e.g. .do for stata

Script layout

1. Header comments
   1. Script name
   2. Script purpose
   3. Input data
   4. Outputs
   5. Authors
   6. First date published
2. Preamble
   1. Settings
      1. Memory space
      2. Seed- for randomization
      3. Version number
      4. References
         * Workspace
         * Input files
         * Output files
         * Working directory
   2. Install/load libraries/dependencies i.e. needed programs
   3. Open log file
3. Body- has working code, and may be split into sections if it is large
   1. Section 1
   2. Section 2
   3. Section 3 ….
4. Footer (for releasing resources)
   1. close log files
   2. close open files
   3. drop references

Reference names: GLOBAL REFERENCES IN CAPS; local references in small letters; use underscore to join words e.g. GLOBAL\_REF i.e. uppercase underscore-separated name; for local reference start with small letter and use camel hump as an alternative to underscore separator

Encapsulation- opening curly brace at end of the line where construct (if, for,etc) begins, and closing brace in own line vertically aligned to construct beginning- option to comment closing brace with the phrase end<construct> for <purpose> e.g. endforloop for cycling through each item in a basket

Indentation for nested construct // Indent statements to show the logical hierarchy.

Keep line length of 80

Remove dead code

Use relative paths