Assignment 3

Quirin Stier
Learn to Code

December 5, 2021

Exercise Create your first package. Create your first package with the reader and writer from the previous assignment. The package should be called "DataIO" standing for "Data Input Output".

1. Create a package with function call
   - package.skeleton()
   - Use parameters to name your package
2. Put your scripts in directory R
3. Create a man file for each script in directory man
4. Fill out the Description file properly
   - Authors with Authors@R: c(person("First", "Last", email= "mail@provider.net", role=c("aut","cre","cph"), person("First", "Last", role=c("aut"))
   - License: GPL-3
5. Fill out the Namespace file properly
   - Each function, you want to use later (external functions) are exported
   - Each function, which is only a helper function in a own script is no exported (internal functions)
   - Define the libraries you are using in your scripts (import)

Exercise Creating scripts for packages. Apply following layout:

1. First line is function name
2. After function name, a commented section follows:
   (a) DESCRIPTION: A 1-2 sentence description of what your method does
   (b) INPUT: A detailed definition of your input with minimum description

(c) OUTPUT: A detailed definition of your output with minimum description

(d) End comment just states the author, year and potential comments about edits

3. Start the main program
4. Begin with assertions (ensure correct input)
5. Followed by main program (which is usually modularized, but here it is too simple, you will learn that later!)
6. End with return statement

Exercise Manual for scripts. Now that you created your first script, you should write manuals to describe what your script does. The manual is written code-like and consists of mandatory and optional variables to be filled out. The style is like \variable{your text goes here}. The manual contains following variables:

1. name: name of your script
2. alias: alias of your script
3. title: title of your script
4. description: 1-3 sentence description
5. usage: examplary function call stating all parameters in their default setting. This needs to be exact, otherwise R will create errors when checking or building packages
6. arguments: your function parameters go here like \item{parameter description}. You can write multiple items below each other to list multiple parameters.
7. value: your function return values. Same style as arguments, see item above
8. details: a more detailed description of what your package does.
9. references: state your references here. Use the apa style for referencing. This is optional.
10. author: name
11. seealso: Reference other R packages
12. examples: give an example which shows the proper use of this particular function. Examples should be of minimum length, short run time and small size of required RAM.
13. keyword: single words describing what your package does
14. concepts: short sentences describing what your package does

\code{\link{aPackageYouAreUsing}} links the code manual for another package or function

Exercise Writing tests. Tests are created in order to test both the script for intended input but also for ill posed input. This should ensure both correct working of your algorithm and the assertions to detect input for which the algorithm was not built. All in all, your algorithm should exactly do what it was intended for. Remember to test the output both for correct datastructure and correct values, when given correct input. For easy start, create a R script and state your test calls there to test your package and check different inputs and assert the output. The strategy is similar to the examples from the manuals for a R package. Use minimum length, run time and size. Check correct and ill posed input. Check the correct data structure of output and of course the correct values which can be known based on the input you are giving and your knowledge about your own function and package.