The State of Naming Conventions in R

by Rasmus Bååth

Abstract Most programming language communities have naming conventions that are generally agreed upon, that is, a set of rules that governs how functions and variables are named. This is not the case with R, and a review of unofficial style guides and naming convention usage on CRAN shows that a number of different naming conventions are currently in use. Some naming conventions are, however, more popular than others and as a newcomer to the R community or as a developer of a new package this could be useful to consider when choosing what naming convention to adopt.

Introduction

Most programming languages have official naming conventions, official in the sense that they are issued by the organization behind the language and accepted by its users. This is not the case with R. There exists the *R internals* document[1] which covers the coding standards of the R core team but it does not suggest any naming conventions. Incoherent naming of language entities is problematic in many ways. It makes it more difficult to guess the name of functions (for example, is it `as.date` or `as.Date`?). It also makes it more difficult to remember the names of parameters and functions. Two different functions can have the same name, where the only difference is the naming convention used. This is the case with `nrow` and `NROW` where both functions count the rows of a a data frame, but their behaviors differ slightly.

There exist many different naming conventions and below is a list of some of the most common. All are in use in the R community and the example names given are all from functions that are part of the **base** package. As whitespace cannot be part of a name, the main difference between the conventions is in how names consisting of multiple words are written.

alllowercase All letters are lower case and no separator is used in names consisting of multiple words as in `searchpaths` or `srcfilecopy`. This naming convention is common in MATLAB. Note that a single lowercase name, such as `mean`, conforms to all conventions but UpperCamelCase.

period.separated All letters are lower case and multiple words are separated by a period. This naming convention is unique to R and used in many core functions such as `as.numeric` or `read.table`.

underscore_separated All letters are lower case and multiple words are separated by an underscore as in `seq_along` or `package_version`. This naming convention is used for function and variable names in many languages including C++, Perl and Ruby.

lowerCamelCase Single word names consist of lower case letters and in names consisting of more than one word all, except the first word, are capitalized as in `colMeans` or `suppressPackageStartupMessage`. This naming convention is used, for example, for method names in Java and JavaScript.

UpperCamelCase All words are capitalized both when the name consists of a single word, as in `Vectorize`, or multiple words, as in `NextMethod`. This naming convention is used for class names in many languages including Java, Python and JavaScript.

If you are a newcomer to R or if you are developing a new package, how should you decide which naming convention to adopt? While there exist no official naming conventions there do exist a number of R style guides that include naming convention guidelines. Below is a non-exhaustive list of such guides.

- **Bioconductor's coding standards**
  http://wiki.fhcrc.org/bioc/Coding_Standards
- **Hadley Wickham's style guide**
  http://stat405.had.co.nz/r-style.html
- **Google's R style guide**
  http://google-styleguide.googlecode.com/svn/trunk/google-r-style.html
- **Colin Gillespie's R style guide**
  http://csgillespie.wordpress.com/2010/11/23/r-style-guide/

Following a style guide will lead to good internal consistency in your code but you are still faced with the choice of naming conventions as there seems to be no consensus between style guides. The coding standards of the Bioconducor project recommend that both function and variable names are written in lowerCamelCase while Hadley Wickham's style guide recommends using underscore_separated names. Google's R style guide proposes UpperCamelCase for function names and period.separated variable names. Colin Gillespie's R style guide agrees with Google's on the the naming of functions but recommends underscore_separated variable names.

[1] http://cran.r-project.org/doc/manuals/R-ints.html

Naming conventions on CRAN

One thing to consider when choosing to adopt a naming convention is what conventions are already popular in the R community. For example, it is safe to say that it would be unconventional to release a package where function names are in all caps as in old FORTRAN. A good source of information regarding the current naming convention practices of the R community is the Comprehensive R Archive Network (CRAN). The function and parameter names used in CRAN packages should reflect the names R users are using, as CRAN is arguably the most common source for add-on packages.

In order to look into this I downloaded the documentation and the NAMESPACE files for all packages on CRAN [2]. The NAMESPACE files were used to extract function names and out of the 4108 packages on CRAN, function names from 2668 packages were retrieved. The reason why it was not possible to get function names from all packages is that while all CRAN packages now include a NAMESPACE file, not all NAMESPACE files explicitly export function names. S3 functions were converted not to include the class name, for example, `plot.myclass` just became `plot`. This was done in order to avoid inflating the number of period.separated function names. The documentation files were used to pick out the parameter names for all documented functions in order to get at what naming conventions are used when naming variables. In total 62,497 function names and 316,852 parameter names were retrieved.

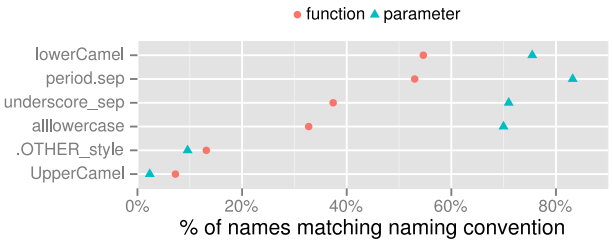

Figure 1: The percentage of function and parameter names from CRAN that matches the five naming conventions.

Figure 1 shows the percentage of function and parameter names that matches the five naming conventions, with lowerCamelCase and period.separated being the most common conventions. The impression, however, is that naming convention usage is quite heterogeneous as all of the five naming conventions seem to be used to some degree. Included in the figure is also the percentage of names that do not match any specified naming convention. These are labeled .OTHER_style. (Examples of such names would be `as.Date` and `Sys.setlocale`). Note that a name can match many naming conventions, especially all names that are alllowercase also match period.separated, underscore_separated and lowerCamelCase conventions. This explains why the parameter names match the top four naming conventions to a higher degree than the function names, as parameter names tend to be single letter words to a larger degree than function names (the single most common parameter name being `x`).

How common is it actually to mix naming conventions in the same package, given that there are many different naming conventions in use in the R community? Counting the minimum number of naming conventions required to cover all function names in each package on CRAN shows that while the largest group (43%) of packages stick to using one naming convention, 28% mix two naming conventions and 28% mix three or more.

Comparing the naming conventions advocated by the style guides with the situation on CRAN shows that some of the proposed naming conventions fit less well with the CRAN data. Both Google and Colin Gillespie propose using UpperCamelCase for function names, which seems to be far from the norm as only 7% of the function names on CRAN conform to this convention. Using underscore_separated names, as the style guide of Hadley Wickham proposes, is also relatively rare as compared to using lowerCamelCase or period.separated names. None of the style guides propose the naming convention that fits the CRAN data best, that is, to name functions using lowerCamelCase and variables using period.separated names. Although a case can be made for using the same naming convention for both variables and functions as, strictly speaking, functions are assigned to variables in R.

Both the CRAN data and the style guides show that there is no consensus regarding naming conventions in R and this it likely to continue as naming conventions, to a large degree, are a matter of taste and habit. If one believes that more homogeneous naming conventions are desirable it is a bit distressing that an entity as influential as Google issues naming convention guidelines that are not compatible with the current usage in the R community. What could help might be to raise awareness in the R community about naming conventions; writers of books and tutorials on R could make a difference here by treating naming conventions when introducing the R language. What is most important, however, is to keep a consistent naming convention style within your code base, whether you are working on a personal project or developing a package.

Rasmus Bååth
Lund University Cognitive Science
Lund University
Sweden
rasmus.baath@lucs.lu.se

[2] The files were retrieved from CRAN on 2012-11-13.