The State of Naming Conventions in R

Rasmus Bååth
rasmus.baath@lucs.lu.se
Lund University Cognitive Science
The only real difficulties in programming are cache invalidation and naming things.

-- Phil Karlton
Outline

- In the R ecosystem many different naming conventions are used.
Outline

- In the R ecosystem many different naming conventions are used.
- This is not a good thing.
Outline

- In the R ecosystem many different naming conventions are used.
- This is not a good thing.
- How to deal with the current naming convention situation.
Different Naming Conventions used in R

• alllowercase
  ○ searchpaths, srcfilecopy
Different Naming Conventions used in R

- alllowercase
  - searchpaths, srcfilecopy
- period.separated
  - as.numeric, read.table
Different Naming Conventions used in R

- alllowercase
  - searchpaths, srcfilecopy

- period.separated
  - as.numeric, read.table

- underscore_separated
  - seq_along, package_version
Different Naming Conventions used in R

- alllowercase
  - searchpaths, srcfilecopy
- period.separated
  - as.numeric, read.table
- underscore_separated
  - seq_along, package_version
- lowerCamelCase
  - colMeans, suppressPackageStartupMessage
Different Naming Conventions used in R

- **alllowercase**
  - searchpaths, srcfilecopy

- **period.separated**
  - as.numeric, read.table

- **underscore_separated**
  - seq_along, package_version

- **lowerCamelCase**
  - colMeans, suppressPackageStartupMessage

- **UpperCamelCase**
  - Vectorize, NextMethod
Different Naming Conventions used in R

- alllowercase
  - searchpaths, srcfilecopy
- period.separated
  - as.numeric, read.table
- underscore_separated
  - seq_along, package_version
- lowerCamelCase
  - colMeans, suppressPackageStartupMessage
- UpperCamelCase
  - Vectorize, NextMethod
- .OTHER_style
  - Cstack_info, Sys.setlocale, Sys.setFileTime
| strip white | period.separated | lowerCamelCase | ? | ? |
Guess the naming convention for parameters of `read.table`!

- `period.separated`: ✔
- `lowerCamelCase`: ✘
- `strip.white`: ✗
Guess the naming convention for parameters of `read.table`!

- `period.separated`: ✔️
- `lowerCamelCase`: ❌
- `strip.white`: ✔️
- `blank lines skip`: ❓
- `strip.white`: ❓
- `blank lines skip`: ❓
Guess the naming convention for parameters of read.table!

<table>
<thead>
<tr>
<th>Parameter</th>
<th>period.separated</th>
<th>lowerCamelCase</th>
</tr>
</thead>
<tbody>
<tr>
<td>strip.white</td>
<td>✔</td>
<td>✘</td>
</tr>
<tr>
<td>blank.lines.skip</td>
<td>✔</td>
<td>✘</td>
</tr>
</tbody>
</table>
Guess the naming convention for parameters of read.table!

<table>
<thead>
<tr>
<th>Parameter</th>
<th>period.separated</th>
<th>lowerCamelCase</th>
</tr>
</thead>
<tbody>
<tr>
<td>strip.white</td>
<td>✔</td>
<td>✘</td>
</tr>
<tr>
<td>blank.lines.skip</td>
<td>✔</td>
<td>✘</td>
</tr>
<tr>
<td>allow escapes</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Guess the naming convention for parameters of `read.table`!

<table>
<thead>
<tr>
<th>Parameter</th>
<th>period.separated</th>
<th>lowerCamelCase</th>
</tr>
</thead>
<tbody>
<tr>
<td>strip.white</td>
<td>✔</td>
<td>✘</td>
</tr>
<tr>
<td>blank.lines.skip</td>
<td>✔</td>
<td>✘</td>
</tr>
<tr>
<td>allowEscapes</td>
<td>✘</td>
<td>✔</td>
</tr>
</tbody>
</table>
Guess the naming convention for parameters of `read.table`!

<table>
<thead>
<tr>
<th>Parameters</th>
<th>period.separated</th>
<th>lowerCamelCase</th>
</tr>
</thead>
<tbody>
<tr>
<td>strip.white</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>blank.lines.skip</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>allowEscapes</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>col names</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Guess the naming convention for parameters of `read.table`!

<table>
<thead>
<tr>
<th>Parameter</th>
<th>period.separated</th>
<th>lowerCamelCase</th>
</tr>
</thead>
<tbody>
<tr>
<td>strip.white</td>
<td>✔</td>
<td>❌</td>
</tr>
<tr>
<td>blank.lines.skip</td>
<td>✔</td>
<td>❌</td>
</tr>
<tr>
<td>allowEscapes</td>
<td>❌</td>
<td>✔</td>
</tr>
<tr>
<td>col.names</td>
<td>✔</td>
<td>❌</td>
</tr>
</tbody>
</table>
Guess the naming convention for parameters of `read.table`!

<table>
<thead>
<tr>
<th>Parameter</th>
<th>period.separated</th>
<th>lowerCamelCase</th>
</tr>
</thead>
<tbody>
<tr>
<td>strip.white</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>blank.lines.skip</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>allowEscapes</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>col.names</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>col classes</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
Guess the naming convention for parameters of `read.table`!

<table>
<thead>
<tr>
<th>Parameter</th>
<th>period.separated</th>
<th>lowerCamelCase</th>
</tr>
</thead>
<tbody>
<tr>
<td>strip.white</td>
<td>✔</td>
<td>✘</td>
</tr>
<tr>
<td>blank.lines.skip</td>
<td>✔</td>
<td>✘</td>
</tr>
<tr>
<td>allowEscapes</td>
<td>✘</td>
<td>✔</td>
</tr>
<tr>
<td>col.names</td>
<td>✔</td>
<td>✘</td>
</tr>
<tr>
<td>colClasses</td>
<td>✘</td>
<td>✔</td>
</tr>
</tbody>
</table>
Guess the naming convention for parameters of `read.table`!

<table>
<thead>
<tr>
<th>Parameter</th>
<th>period.separated</th>
<th>lowerCamelCase</th>
</tr>
</thead>
<tbody>
<tr>
<td>strip.white</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>blank.lines.skip</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>allowEscapes</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>col.names</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>colClasses</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>strings as factors</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Parameter</td>
<td>period.separated</td>
<td>lowerCamelCase</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>strip.white</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>blank.lines.skip</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>allowEscapes</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>col.names</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>colClasses</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>stringsAsFactors</td>
<td>✗</td>
<td>✔</td>
</tr>
</tbody>
</table>
Unofficial naming conventions guidelines.

- Bioconductor’s coding standards.
  - `readTable`, `stringsAsFactors`.
Unofficial naming conventions guidelines.

- Bioconductor’s coding standards.
  - readTable, stringsAsFactors.
- Hadley Wickham’s style guide
  - read_table, strings_as_factors.
Unofficial naming conventions guidelines.

- Bioconductor’s coding standards.
  - readTable, stringsAsFactors.
- Hadley Wickham’s style guide
  - read_table, strings_as_factors.
- Colin Gillespie’s R style guide
  - ReadTable, strings_as_factors
Unofficial naming conventions guidelines.

- Bioconductor’s coding standards.
  - readTable, stringsAsFactors.
- Hadley Wickham’s style guide
  - read_table, strings_as_factors.
- Colin Gillespie’s R style guide
  - ReadTable, strings_as_factors
- Google’s R style guide
  - ReadTable, strings.as.factors
What naming conventions are used in practice?

- Comprehensive R Archive Network to the rescue!
What naming conventions are used in practice?

- Comprehensive R Archive Network to the rescue!
- I downloaded all (4411) packages on CRAN.
What naming conventions are used in practice?

- Comprehensive R Archive Network to the rescue!
- I downloaded all (4411) packages on CRAN.
- Got 339032 parameter names and 76176 function names.
What naming conventions are used in practice?

- Comprehensive R Archive Network to the rescue!
- I downloaded all (4411) packages on CRAN.
- Got 339032 parameter names and 76176 function names.
- Removed the class part of S3 functions, (plot.mcmc -> plot).
What naming conventions are used in practice?

- Comprehensive R Archive Network to the rescue!
- I downloaded all (4411) packages on CRAN.
- Got 339032 parameter names and 76176 function names.
- Removed the class part of S3 functions, (plot.mcmc -> plot).
- Counted how many of the functions and parameters matched the different naming conventions.
Why heterogenous naming conventions are a bad thing.

- It is not aesthetically pleasing.
Why heterogenous naming conventions are a bad thing.

- It is not aesthetically pleasing.
- It makes R harder to learn.
A Memory Experiment

Mixed capitalization condition (n = 71)

flood Critic victory basis deficit testing
General alcohol Track profile Train equity

All lower case condition (n=77)

profile critic train general flood track
alcohol victory equity testing basis deficit
Participants remembered on average $1.2 \pm 0.6$ more words in the *All lower case* condition. (95% bootstrap CI)
Why heterogenous naming conventions are a bad thing.

- It is not aesthetically pleasing.
- It makes R harder to learn.
- It makes R harder to use.
Heterogenous naming conventions makes R harder to use.

- It is practically impossible to follow one naming convention even if you try.
Heterogenous naming conventions makes R harder to use.

- It is practically impossible to follow one naming convention even if you try.
- Easier to make errors.
Heterogenous naming conventions makes R harder to use.

- It is practically impossible to follow one naming convention even if you try.
- Easier to make errors.
- It is harder to guess names of functions and parameters.
as.date("2013-07-11")
as.date("2013-07-11")

asDate("2013-07-11")
as.date("2013-07-11")
asDate("2013-07-11")
as_date("2013-07-11")
as_date("2013-07-11")
as.date("2013-07-11")
asDate("2013-07-11")
as_date("2013-07-11")
as.Date("2013-07-11")
Heterogenous naming conventions makes R harder to use.

- It is practically impossible to follow one naming convention even if you try.
- Easier to make errors.
- It is harder to guess names of functions and parameters.
- It invites functions with names that just differ by convention.

anova vs Anova  
ncol vs NCOL  
summary vs Summary
So, what to do?

- Use autocompletion.
Autocompletion is great!

- Relieves the cognitive burden of having to remember identifier names exactly.
Autocompletion is great!

- Relieves the cognitive burden of having to remember identifier names exactly.
- Works great in many editors, for example Rstudio.
Autocompletion is great!

- Relieves the cognitive burden of having to remember identifier names exactly.
- Works great in many editors, for example Rstudio.
- Does **not** work if you don’t know the case of the first letter.
> vectorize

(No matches)
> vectorize

Vectorize{base} Vectorize(FUN, vectorize.args = arg.names, SIMPLIFY = TRUE, USE.NAMES = TRUE)

Vectorize creates a function wrapper that vectorizes the action of its argument FUN.
Autocompletion is great!

- Relieves the cognitive burden of having to remember identifier names exactly.
- Works great in many editors, for example Rstudio.
- Does **not** work if you don’t know the case of the first letter.
- Does **not** work if you use the wrong naming convention.
> seqAlong

(No matches)
seqAlong

seq_along {base}  seq_along(along.with)  NULL

Generate regular sequences. seq is a standard generic with a default method. seq.int is a primitive which can be much faster but has a few restrictions. seq_along and seq_len are very fast primitives for two common cases.

Press F1 for additional help
So, what to do?

- Use autocompletion.
- At least follow some naming convention.
So, what to do?

- Use autocompletion.
- At least follow some naming convention.
- But don’t follow Google’s!
Google's R Style Guide

google-styleguide.googlecode.com/svn/trunk/google-r-style.html

Summary: R Style Rules. File Names: end in .R; Identifiers: variable. name,
FunctionName , kConstantName; Line Length: maximum 80 characters; Indentation: ...
So, what to do?

- Use autocompletion.
- At least follow some naming convention.
- But don’t follow Google’s!
- Be consistent.
So, what to do?

- Use autocompletion.
- At least follow some naming convention.
- But don’t follow Google’s!
- Be consistent.

Like hadley_wickham.
Conclusions

- Naming conventions are important.
Conclusions

- Naming conventions are important.
- You’re not totally off if you use lowerCamelCase for function names.
Conclusions

- Naming conventions are important.
- You’re not totally off if you use lowerCamelCase for function names.
- You’re **totally off** if you use UpperCamelCase.
Conclusions

- Naming conventions are important.
- You’re not totally off if you use lowerCamelCase for function names.
- You’re **totally off** if you use UpperCamelCase.
- CRAN is great for statistics about R usage.
From the UseR 2013 Website:

"Following eight successful useR! meetings, the conference is focused on:
1. R as the ‘lingua franca’ of data analysis and statistical computing, ..."
Thanks for listening!
% of function names matching naming convention

Year package was released

- lowercase
- lowerCamel
- period.sep
- UpperCamel
- underscore_sep
- .OTHER_style