



Credibly Curious

Nick Tierney's (mostly) rstats blog

[Blog](#)[About](#)[Publications](#)[Talks](#)[Resources](#)[Software](#)

2018-06-20

Naming Things

Categories: [rstats](#) [Teaching](#)

5 minute read

There are only two hard things in Computer Science: cache invalidation and naming thing – Phil Karlton ([seen in r-packages](#))

I'm not really sure what cache invalidation is, but I certainly agree that naming things is pretty damn hard. I thought it would be interesting to share some thoughts and discussion that I've been having about naming packages. There was some [recent discussion on tidy names](#) on Twitter, and I felt like this was a good time to discuss this idea. Well, that, and this blog post has been sitting in draft form for 1 year on my computer.

Naming Packages

This initial discussion began at the [rOpenSci Unconf in 2017](#), where [Noam Ross](#) and [Jim Hester](#) talked about a few different ways to think about naming packages in R, that fell into several categories. I thought it would be fun to discuss these!

The oldschool (mixedCaseRs)

The oldSchool namer generally mixes case in an R package, often capitalising the “R”, or going all in on ALL CAPS.

Examples:

- [twitterR](#)
- [formatR](#)
- [UpSetR](#)
- [ochRe](#)
- [MASS](#)
- [Matrix](#)

Although personally I wouldn't use this style as it can make it difficult to type, they have a certain charm, and are easy to google - provided you spell it right. And these are still seriously useful - Matrix provides extensive support for modelling, UpSetR produces great plots as an alternative to venn diagrams.

The Modern (lowercasers)

The modern namer has everything in lowercase and usually carefully places an `r`. For example:

- [dplyr](#)
- [tidyr](#)
- [purrr](#)
- [furrr](#)
- [knitr](#)
- [naniar](#)

These names are pretty popular amongst the tidyverse set of R packages

The Post Modern (names)

This involves using just the original name, as is. This name must also be a thing, or a person.

Examples:

- broom
- checkers
- forecast
- greta
- fable
- english

Notably, `forecast` and `english` were postmodern before modern package names existed.

The “gg”

The growth of `ggplot2` packages to add extra themes or geoms, starting with `gg` , these packages let you know what they’re on about:

- ggthemes
- gglabeller
- ggforce
- gganimate
- ggalluvial
- ggtree
- ggfocus

- [gghighlight](#)

The variety in all of these packages speaks to the utility, extensibility, and general awesomeness of `ggplot2` .

Tidy-preface

With the announcement of The Tidyverse being made official by Hadley at UseR! in 2016 (see the [video](#) around 44 minutes) - there has been a trend of packages to preface their name with `tidy` :

- [tidytext](#)
- [tidygenomics](#)
- [tidyzillow](#)
- [tidylex](#)
- [tidycensus](#)
- [tidyhydat](#)
- [tidykml](#)

This is super cool, and a neat way to find useful packages. It's also worth mentioning that the tidyverse is only officially about 2 years old - and there has been a lot of really nice, positive [growth in R packages](#), [community](#), and [a team](#) of great people working on making great software.

And more?

There are for sure other ways to categorize the names - I don't think that you can categorize the whole naming process of R packages into just three categories - there is a rich process that happens when you name something, and I won't want to distill it down to a few labels!

Update on 2018-06-24: The [twitter thread](#) for this blogpost had some nice discussion and additions from [Noam Ross](#) and [Jenny Bryan](#). I ended up using this script to search for some evidence:

```
library(tidyverse)

pkg_names <- available.packages() %>%
  tibble::as_tibble() %>%
  dplyr::select(Package)

# package with a "2" in the name
pkg_names$Package[str_detect(pkg_names$Package, "2")]

# package with a "." in the name
pkg_names$Package[str_detect(pkg_names$Package, "\\.")]
```

The Acronyms!

Name your package as an acronym! These were kind of hard to search for so I could only think of three - there are more, I'm sure!

- [brms](#) - Bayesian Regression Models using Stan
- [mgcv](#) - Mixed GAM Computation Vehicle
- [eechidna](#) - Exploring Election and Census Highly Informative Data Nationally for Australia

The modifiers!

For example, modify the package with “lite” or “Lite” to indicate that this is a stripped down version of something else:

- [jsonlite](#)
- [svglite](#)
- [mongolite](#)

- [viridisLite](#)

The subcategories!

Got a package that spans across multiple categories? See for example the [future](#) family of packages and more:

- [future.batchtools](#)
- [future.apply](#)
- [aws.alex](#)
- [aws.s3](#)
- [rpart.plot](#)

Adding a 2

Some (approximate) backstory - ggplot was going great, but Hadley Wickham wanted to change the API from a nested function call to using `+` - the API change was so dramatic that he had to release a separate package, `ggplot2`. You can also use `2` to indicate “to” - from X to Y, from R to JAGS, R to virtual reality, R to D3:

- [ggplot2](#)
- [rehape2](#)
- [xml2](#)
- [tcltk2](#)
- [fpp2](#)
- [r2vr](#)
- [r2d3](#)
- [R2jags](#)

- [bib2df](#)

How to make a name?

Coming up with package name often takes a lot of time and thought. Personally, I've changed the names of the packages `visdat` and `narnia` three times. `visdat` was initially `footprintr`, then `vizdat`, and then finally `visdat`. `narnia` was initially `ggmissing`, then `naniar`, then `narnia`, then back to `naniar`. Why the changes from `narnia` and back and forth? Well, mainly I was worried about some awkward cease and desist letters from the estate of CS Lewis once the package got on CRAN, and I'm not really sure how to change the name back on CRAN - I'm pretty sure that once it is on CRAN, that's it.

But, my story aside, here are some general tips.

I find that I generally need to sit on a name for a while and use it, describe it to a few people. I find that once I state the name of the package to them, they often have to ask how to spell it, and if that process feels difficult multiple times, and if I can't work out how to explain why I named it that, then I generally try and rename it. Often this comes down to a few things:

- Is the package name hard to say?
- Is the package name hard to spell?
- Can you remember the name of the package?
- Does the package name evoke the task it does, or perhaps fit into some sort of larger pun or story.

Further resources for naming

I would look at the `available` package for checking if the name exists. Maybe one day this will come up with something creative or unusual.

Hadley has a [good guide](#) to naming packages, but I think that this process requires a few iterations.


Also another post by [Yihui Xie](#) has a good post on naming things - make sure it is easy to type!

0 Comments - powered by [utteranc.es](#)

Write

Preview

Sign in to comment

 Styling with Markdown is supported

Sign in with GitHub

© [Nicholas Tierney](#) 2013-20 | [Github](#) | [Twitter](#) | [RSS](#) | [Edit this page](#)