A Computational Analysis of the Dynamics of R Style Based on 94 Million Lines of Code from All CRAN Packages in the Past 20 Years

Chia-Yi Yen
R-Ladies Taipei / University of Mannheim
yen.chiayi@gmail.com

Mia Huai-Wen Chang
R-Ladies Taipei / Akelius Residential Property AB
mia5419@gmail.com

Chung-hong Chan
Hong Kong R User Group / University of Mannheim
chung-hong.chan@mzes.uni-mannheim.de

THE PROBLEM: There are so many programming style variations (PSV). Which one do you use?

```
myFunction<- function(x,y,z = TRUE)
{
   if(z) {
     x + y
}}
```

```
my_function=function(x, y, z = T){;
if(z) { x+y };
};
```

```
Myfunction<-function(x,y,z=T){
     if(z)
x + y }
}
```

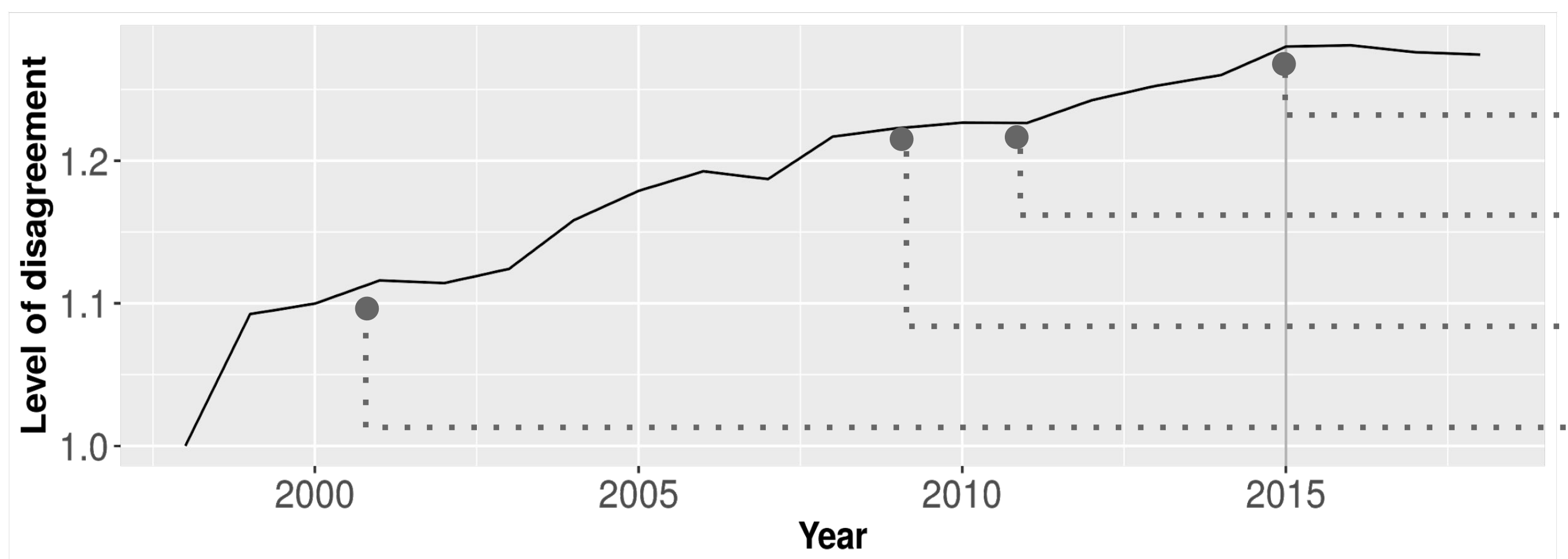
```
MYFUNCTION = function(x, y, z=TRUE){
     if(z) {
        x+y
     }
}
```

4 out of at least possible **7168** variations.

Which of the above cannot be correctly evaluated?

We analysed the distribution of PSV from 1998 to 2018.

A consensus is building...



2015: *R Packages* the book

2011: RStudio / devtools

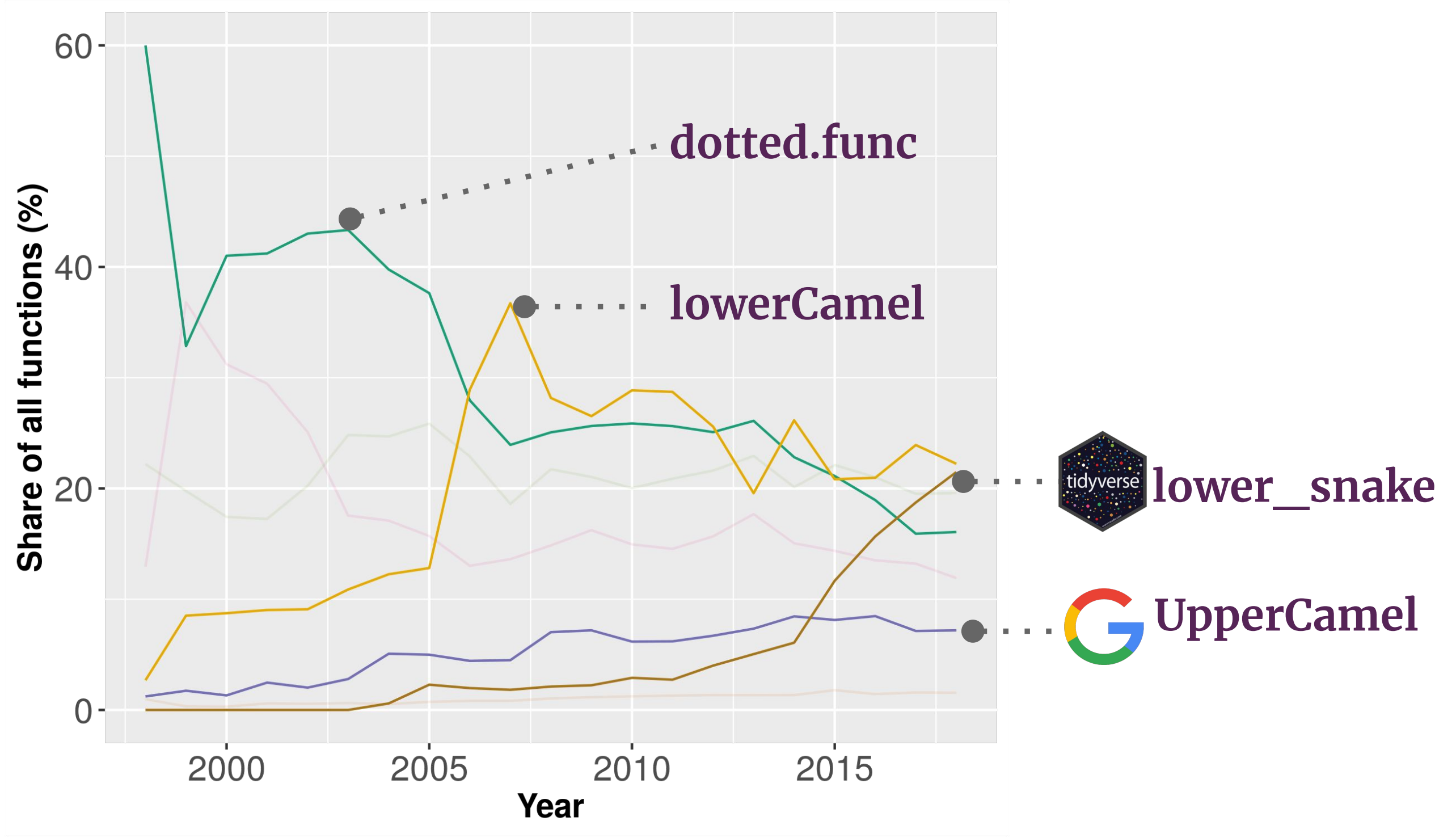
2009: Google Style Guide

2001: S4 / = as assignment

# level of disagreement is quantified by normalized Shannon entropy of the PSV distribution per year. Lower value indicates a dominance of certain styles.

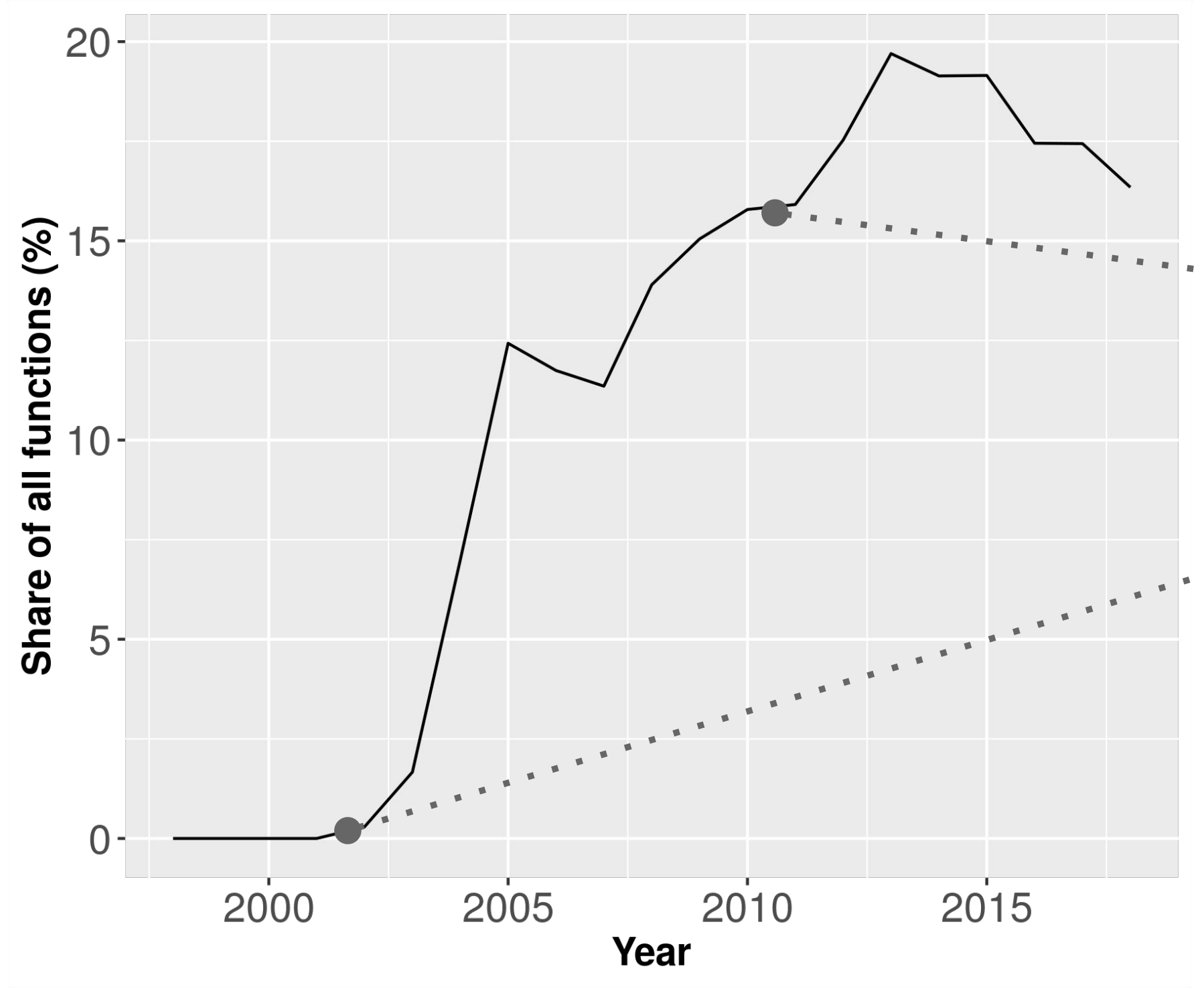
WHAT DRIVES PSV?

Effect of Style-guides: naming conventions



# de-emphasize 3 naming conventions (ALLUPPER, alllower, other) for simplicity

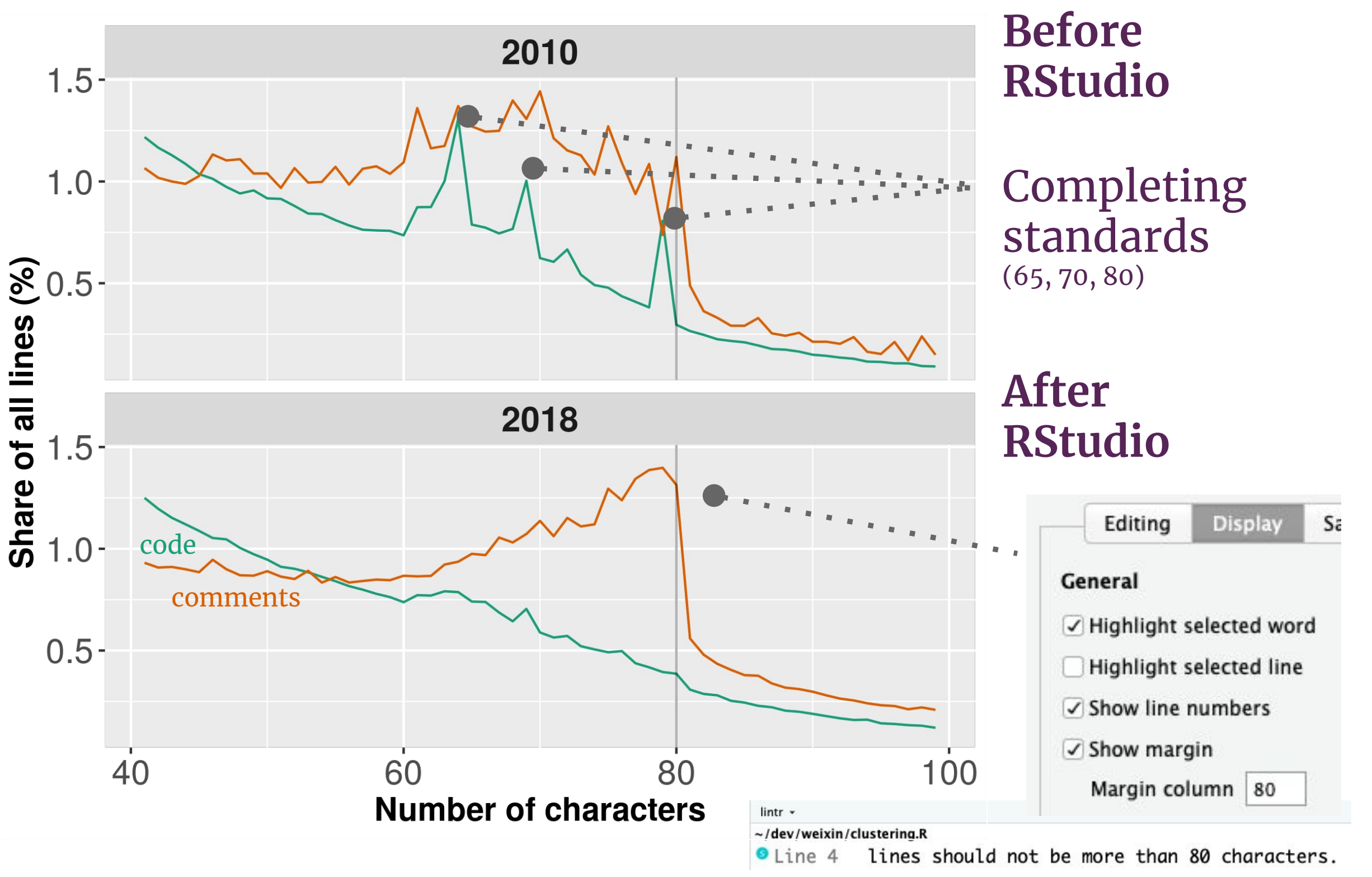
Effect of Introducing a New Language Feature: using = as assignment operator



2010
Ihaka used this style in his paper.

2001
Introduction of the feature (R 1.4.0)

Effect of Editors : line length (options, linters)

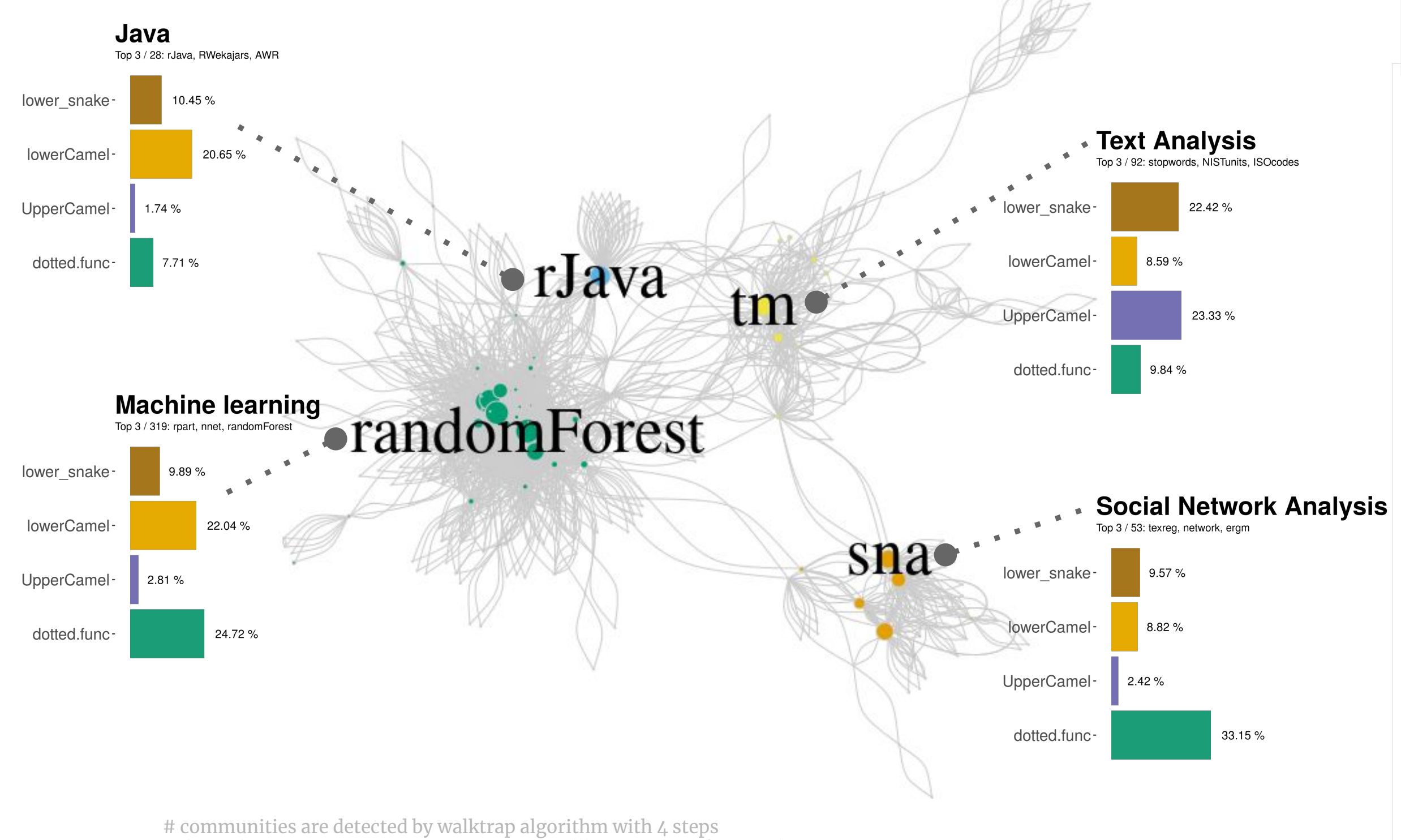


~/dev/weixin/clustering.R
Line 4   lines should not be more than 80 characters.

COMMUNITY-SPECIFIC VARIATIONS

CRAN Packages Dependency Network

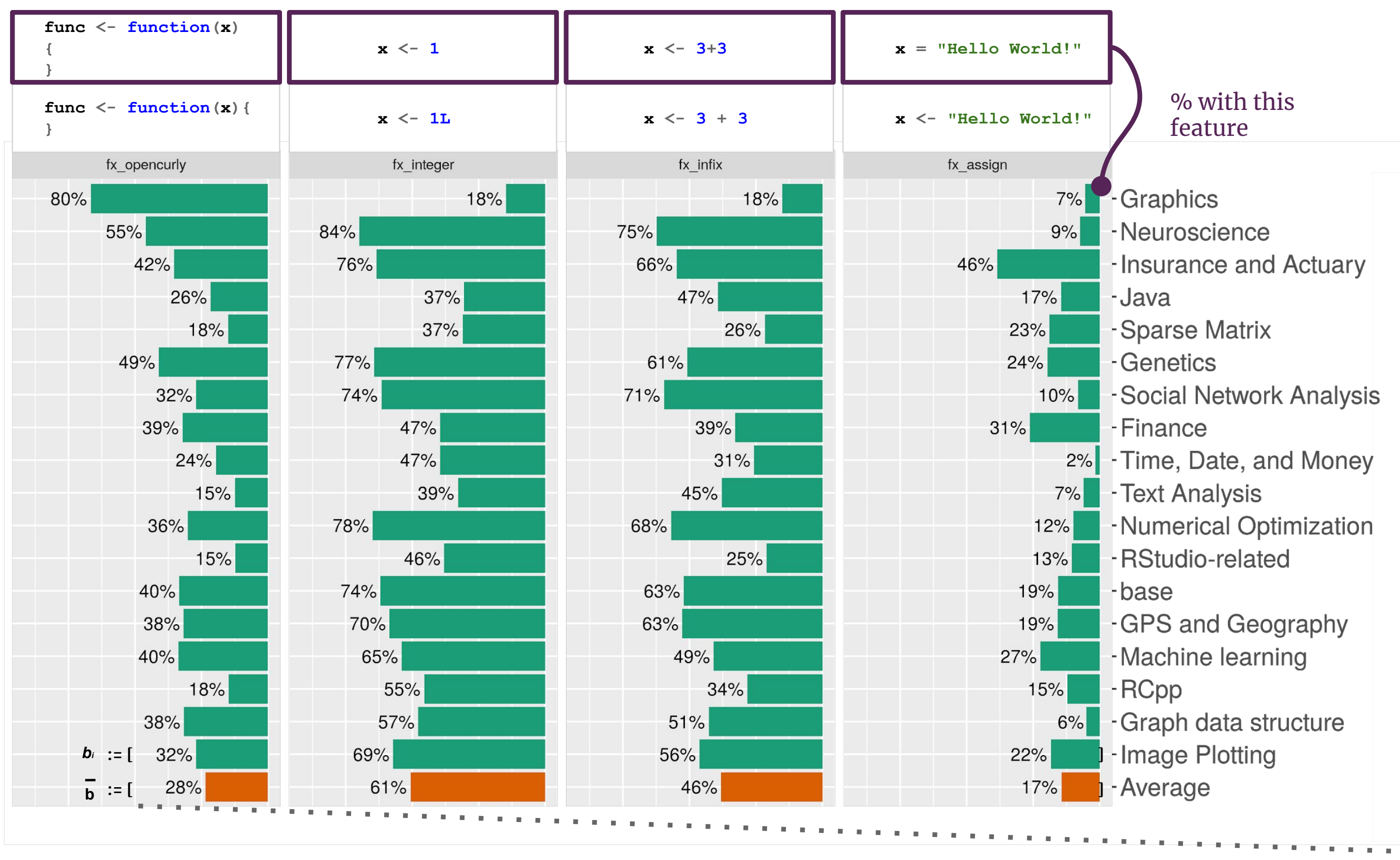
(subgraph of 4 selected communities and the distribution of naming conventions for each community)



# communities are detected by walktrap algorithm with 4 steps
# discard 3 naming conventions (ALLUPPER, alllower, other) for simplicity

Divergence in Styles among Communities

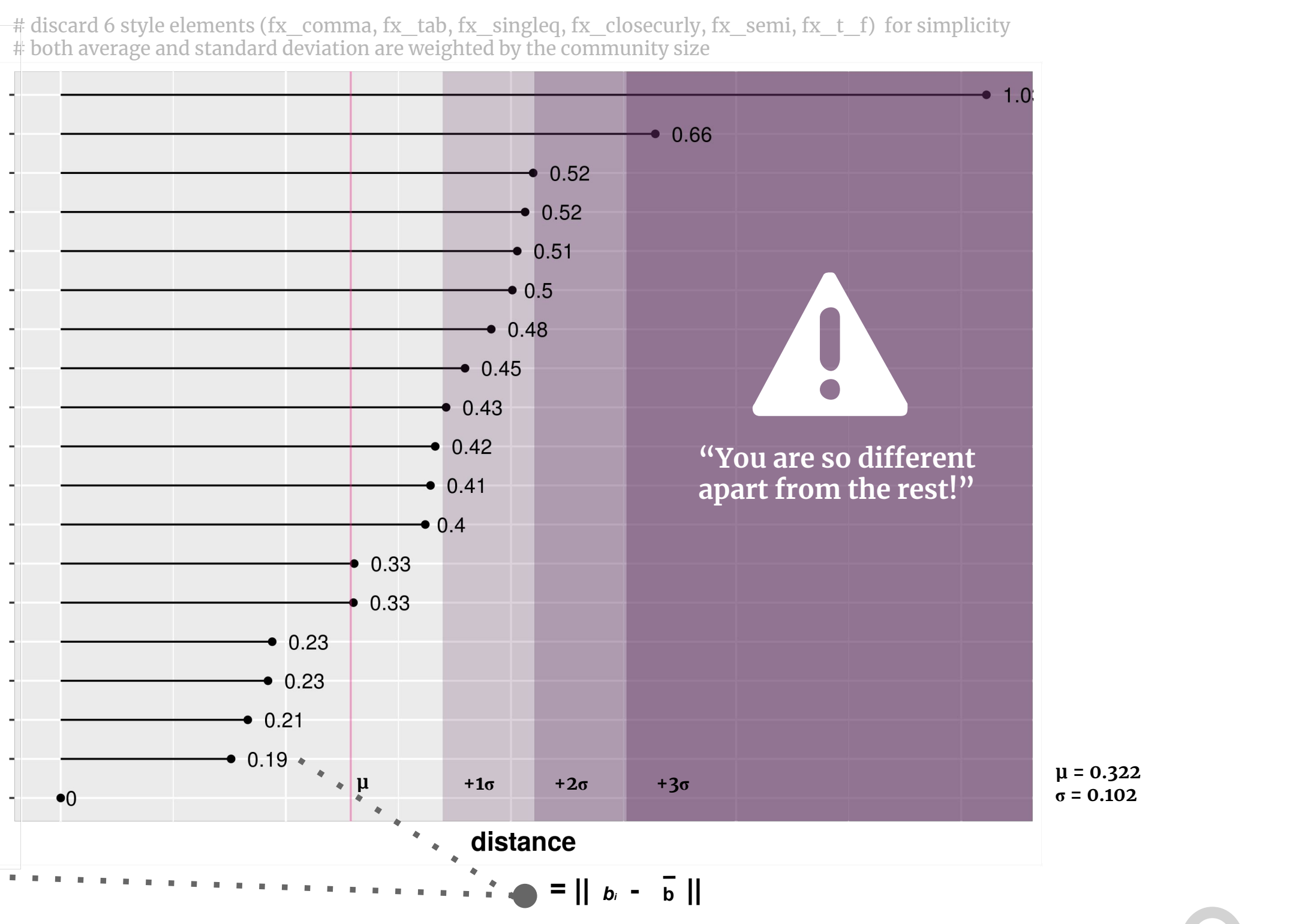
(dispute over 10 style-elements among 18 large communities)



Who is the “Naughty, Naughty”?

(numbers are the Euclidean distance to average community)

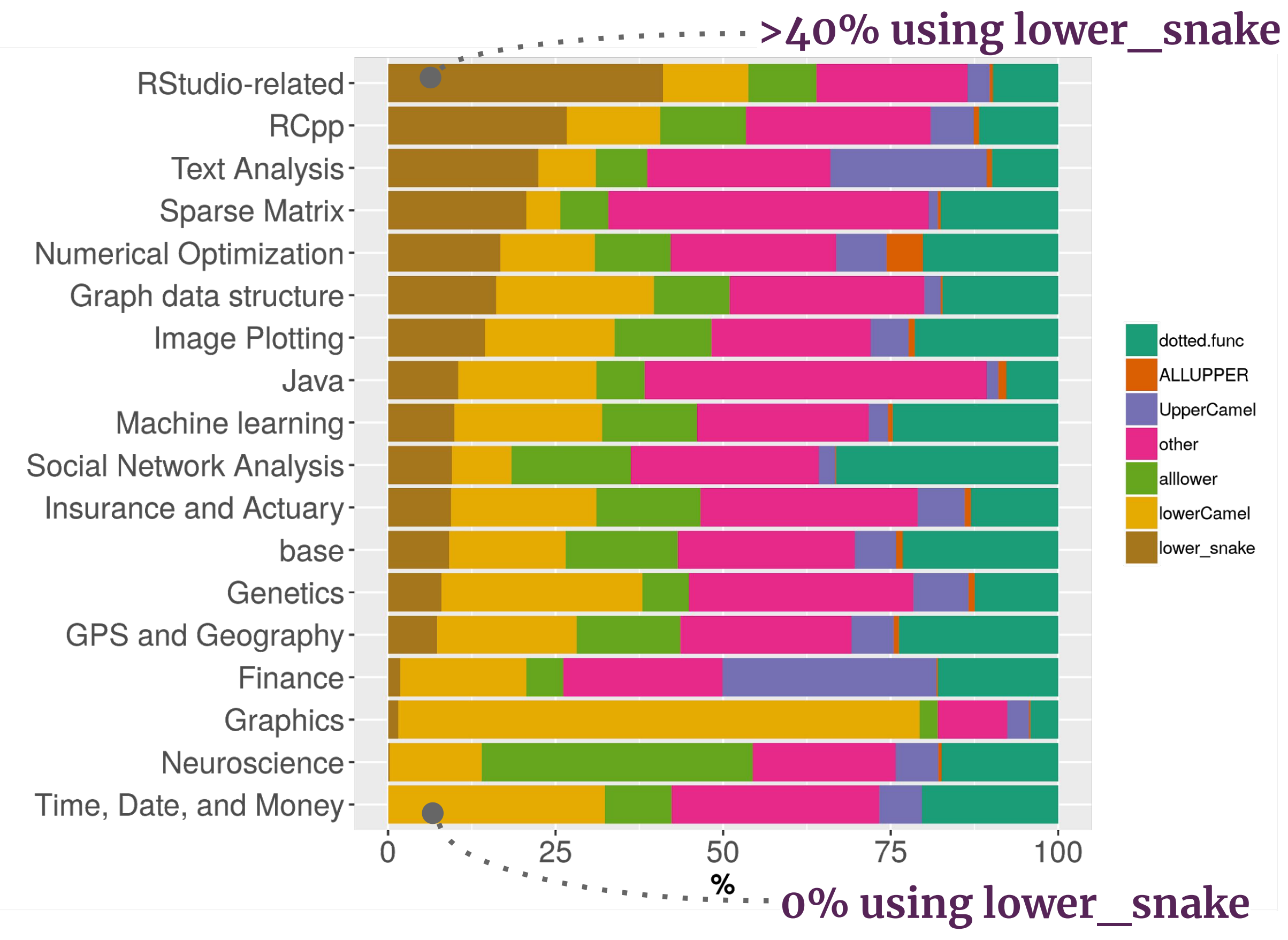
# discard 6 style elements (fx_comma, fx_tab, fx_singleq, fx_closecurly, fx_semi, fx_t_f) for simplicity
# both average and standard deviation are weighted by the community size



“You are so different apart from the rest!”

Like Likes Like

(popularity of naming styles among 18 large communities)



↑ details here ↑

R-Ladies Taipei

HKRUG
Hong Kong R User Group

SUMMARY

Consensus-based Style

(numbers are % of functions using that style-element)

Use `lowerCamel` (22.2%) or `snake_case` (21.5%)

Use `<-` to assign (83.7%)

Add a space after commas (83.7%)

Don’t use `T` / `F` (95.1%)

Don’t use tab to indent (89.2%)

Use `}` on its own line, unless before `else` (89.2%)

```
softplusFunc <- function(value, leaky = FALSE) {
   if (leaky) {
      warnings("using leaky RELU!")
      return(ifelse(value > 0, value, value * 0.01))
   }
   return(log(1 + exp(value)))
}
```

Use same line `{` then a newline (70.0%)

Use double quotes for strings (87.6%)

Add spaces around infix operators (55.3%)

Don’t explicitly type integers (i.e. `1L`) (60.4%)

Don’t terminate lines with `;` (94.6%)

mzes

UNIVERSITY OF MANNHEIM
Business School