```r
library(dplyr)

r-ladies_global %>%
  filter(from = 'Taipei', travel_to = 'UseR! 2019')
```

R-Ladies Taipei

The Dynamic of R Style

And a glance at UseR! 2019 Toulouse

R-Ladies Taipei

Hello!

I am Yen.

- PhD student @Mannheim Business School, Germany
- Co-founder of R-Ladies Taipei
- Contact me: yen.chiayi@gmail.com

UseR! 2019 @Toulouse

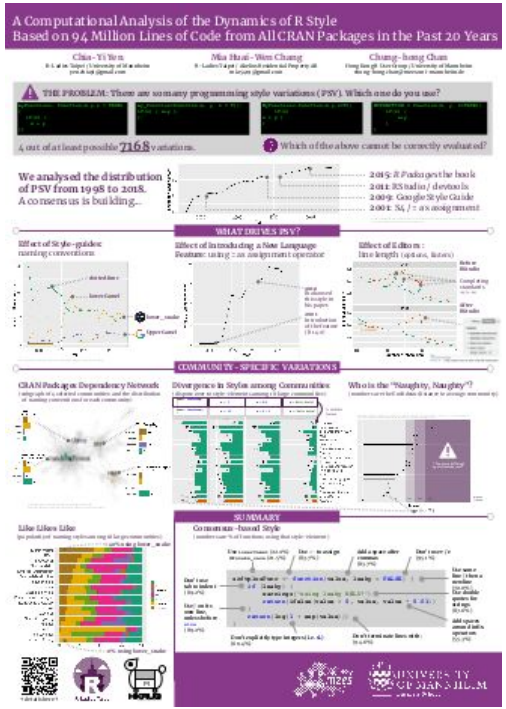
- 2019 @ Toulouse, France
- Talks
  - core team: rstudio team, bioconductor
  - stars: Hadley Wickham
  - diverse field: insurance/pharm / astronomy)
- 2020 @ St. Louis, USA
- R-Ladies Lunch
  - global
  - Taipei 2nd in the world!
- Poster -> dispute across keynote speakers

R-Ladies Taipei

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



R-Ladies Taipei

HKRUG
Hong Kong R User Group

R-Ladies Taipei

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
    }
}
```

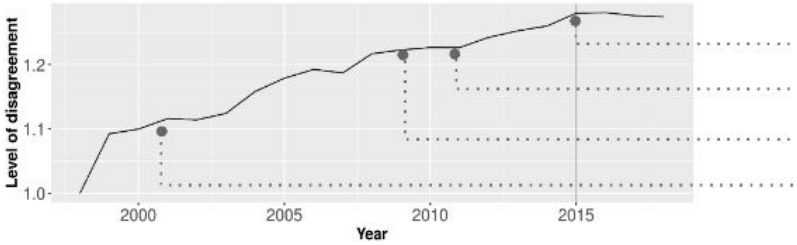
4 out of at least possible **7168** variations.

Which of the above cannot be correctly evaluated?

R-Ladies Taipei

> *Given no dominant style guide in the past 20 years,*
> *what happens to the **programming styles** in R community then?*

We analysed the distribution of PSV from 1998 to 2018.
A consensus is building...

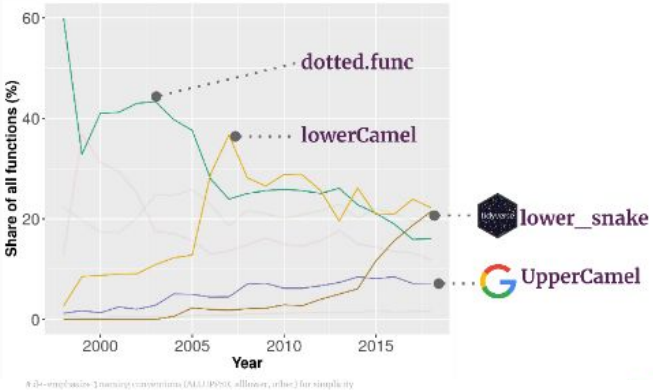


2015: *R Packages* the book
2011: RStudio / devtools
2009: Google Style Guide
2001: S4 / = as assignment

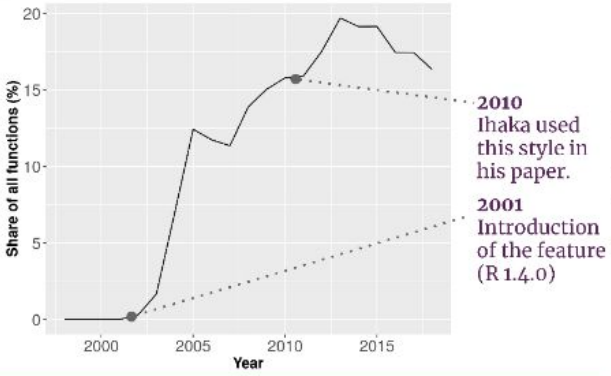
What Drives Programming Style Variation?

R-Ladies Taipei

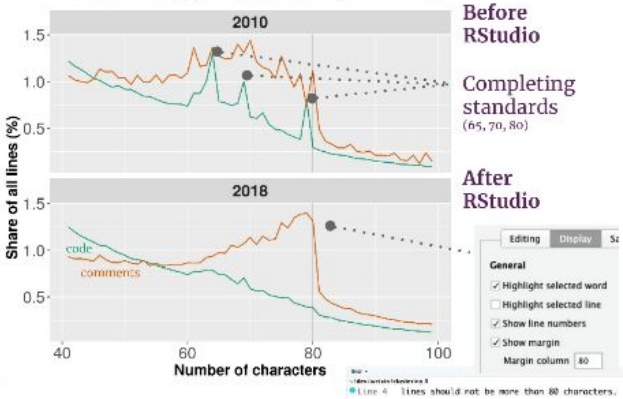
Effect of Style-guides:
naming conventions



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



Effect of Editors :
line length (options, linters)



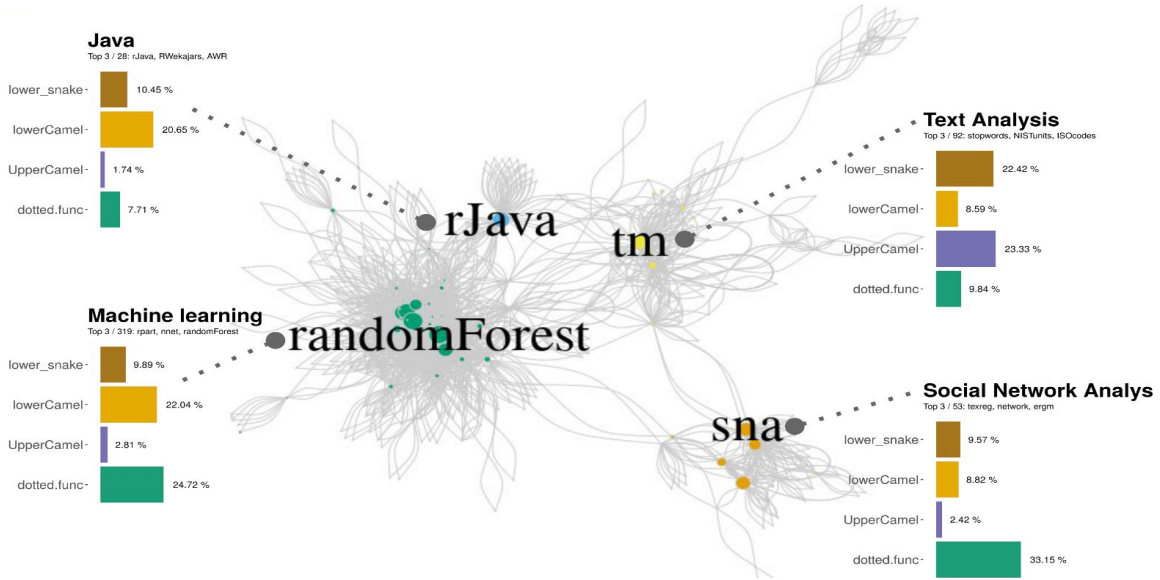
CRAN Packages Dependency Network

R-Ladies Taipei

ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics

A system for 'declaratively' creating graphics, based on "The Grammar of Graphics". You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Version:	3.2.0
Depends:	R (≥ 3.2)
Imports:	digest, grDevices, grid, gtable (≥ 0.1.1), lazyeval, MASS, mgcv, reshape2, rlang (≥ 0.3.0), scales (≥ 0.5.0), stats, tibble, viridisLite, withr (≥ 2.0.0)
Suggests:	covr, dplyr, ggplot2movies, hexbin, Hmisc, knitr, lattice, mapproj, maps, maptools, multcomp, munsell, nlme, profvis, quantreg, rgeos, rmarkdown, rpart, sf (≥ 0.7-3), svglite



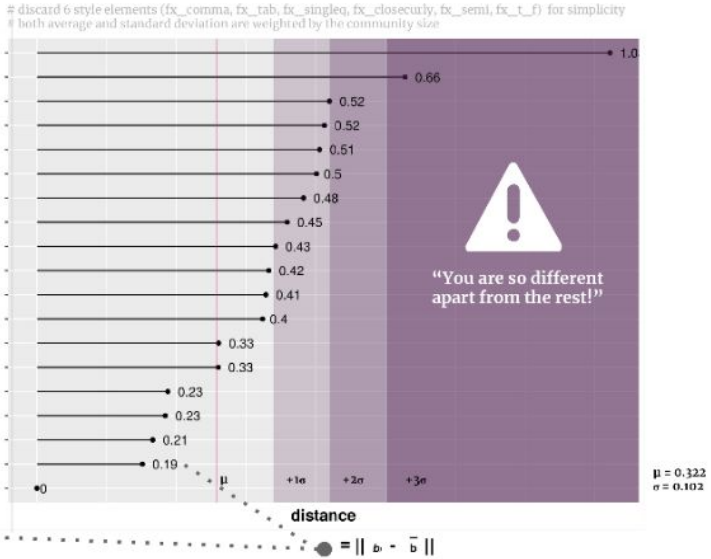
R-Ladies Taipei

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)



R-Ladies Taipei

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%)

R-Ladies Taipei

Thank you for your attention!

↑ details here ↑