

What's new in 2021

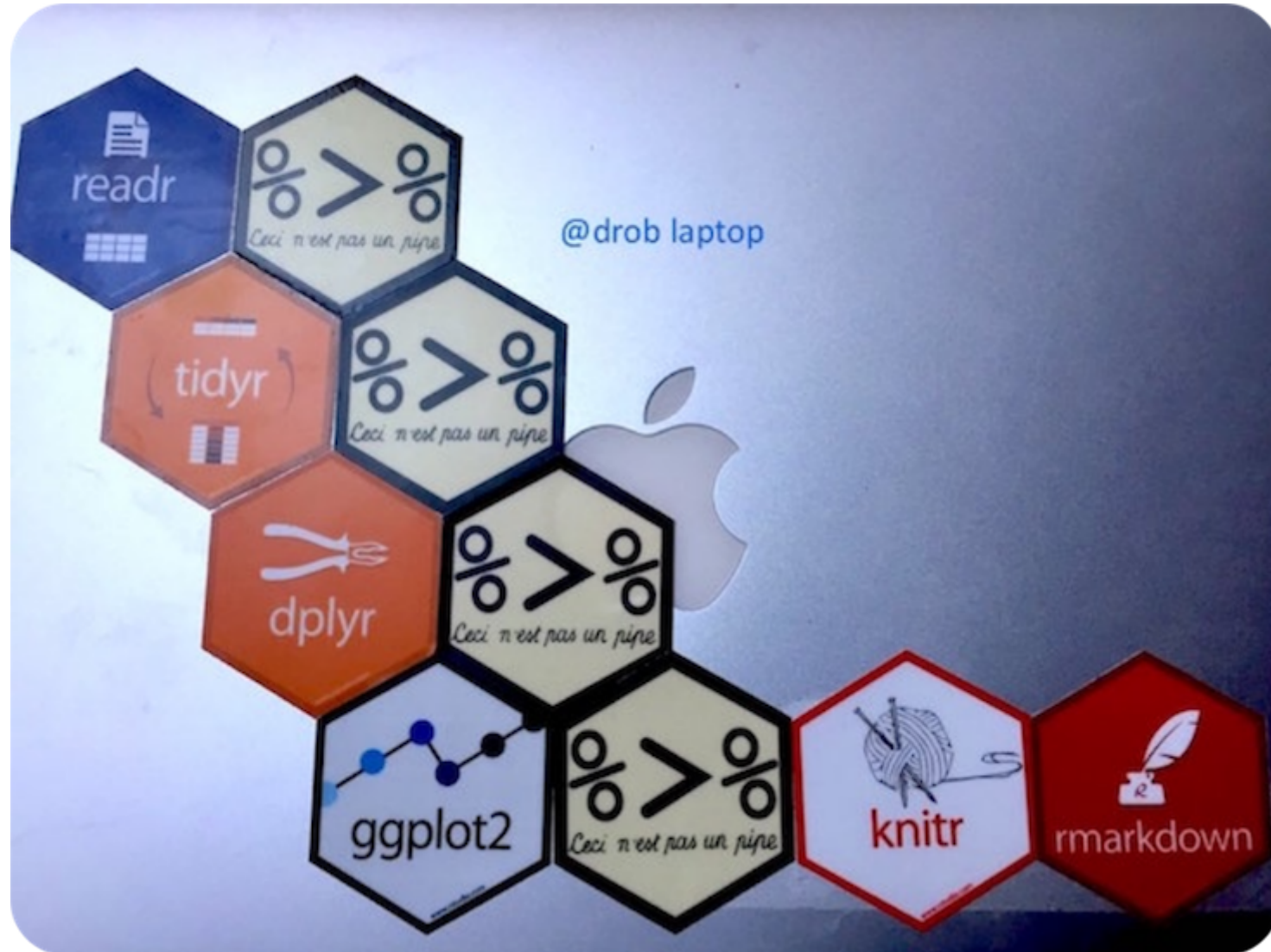
in the **tidyverse** development



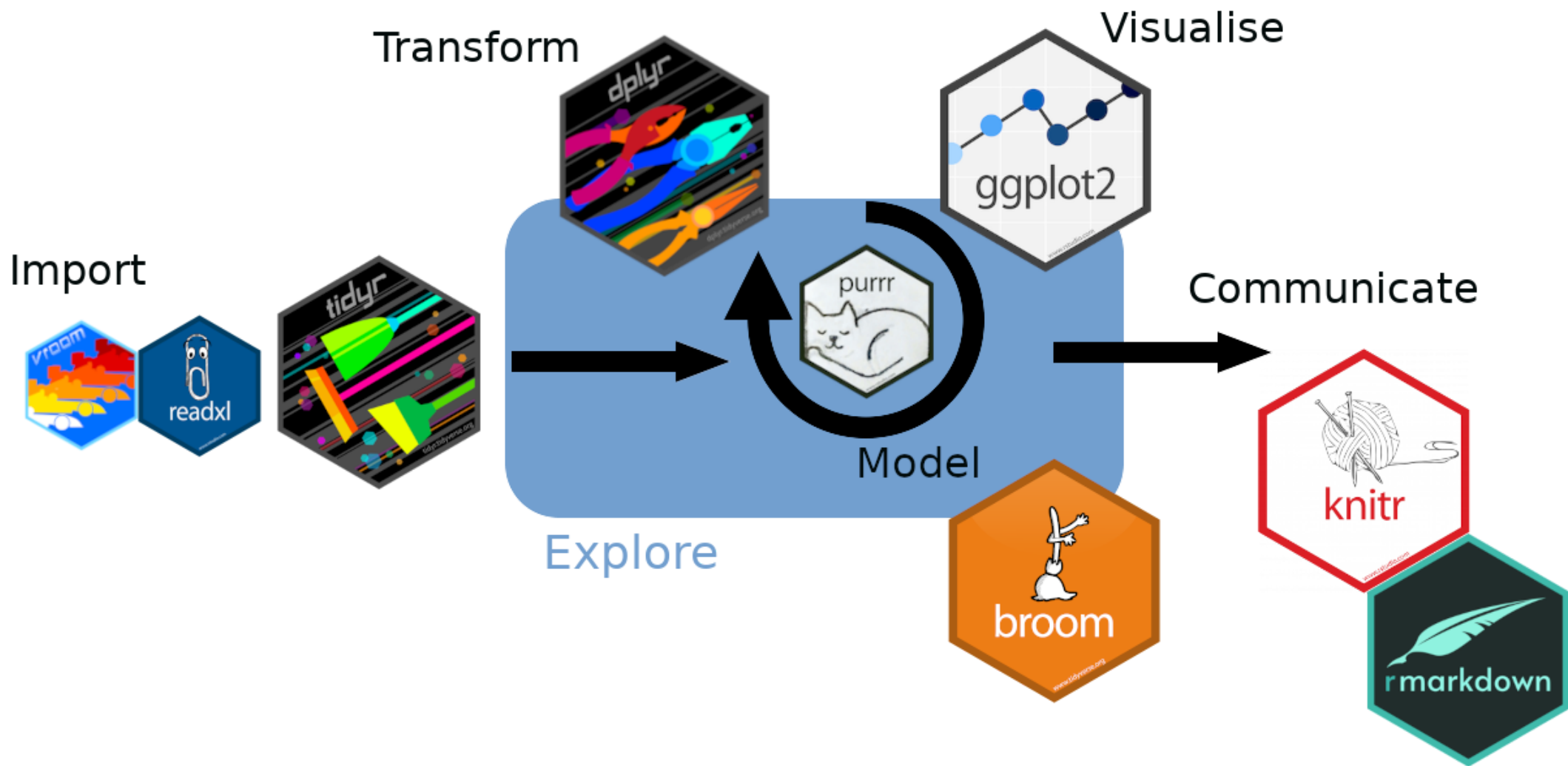
A. Ginolhac | rworkshop | 2021-09-10

Tidyverse: pros & cons

Workflow, by David Robinson



Packages in processes



List of components

Core

- `ggplot2`, for data visualization
- `dplyr`, for data manipulation
- `tidyr`, for data tidying
- `readr`, for data import (`vroom` default in the future?)
- `purrr`, for functional programming
- `tibble`, for tibbles, a modern re-imagining of data frames
- `stringr`, for strings
- `forcats`, for factors

Extended

- Modelling
 - `modelr`, for modelling within a pipeline
 - `broom`, for models -> tidy data
- Programming
 - `rlang`, low-level API
 - `glue`, alternative to paste
- Working with specific types of vectors:
 - `hms`, for times
 - `lubridate`, for date/times
 - `vctrs`, for vectors
- Importing other types of data:
 - `feather`, for sharing data
 - `fs`, for cross platform file system ops
 - `haven`, for SPSS, SAS and Stata files
 - `httr`, for web apis
 - `jsonlite` for JSON
 - `readxl`, for `.xls` and `.xlsx` files
 - `rvest`, for web scraping
 - `xml2`, for XML files
 - `DBI`, for relational databases

source: <https://tidyverse.tidyverse.org/>. H. Wickham

Tidyverse criticism, a dialect



Dr Gavin Simpson 🇬🇧🇪🇺🇩🇰 @ucfagls · Jan 12, 2017

Replying to @ucfagls and @hadleywickham

not even mainly about the pipes. The consistency, verb usage, terminology, lots of NSE in tidyverse is different to std R 1/2



Hadley Wickham ✓

@hadleywickham

yeah. I think the tidyverse is a dialect. But its accent isn't so thick

7:21 PM · Jan 12, 2017



♡ 11 💬 1 🔗 Copy link to Tweet

[Tweet your reply](#)

Criticism, controversy

- [In StackOverflow's comment](#)

5 I would simply do `library(data.table) ; melt(setDT(have), id = 1:2, measure = patterns("genotype", "freq"))` but that wasn't developed by Hadley, so you can safely ignore. – [David Arenburg](#) 13 hours ago

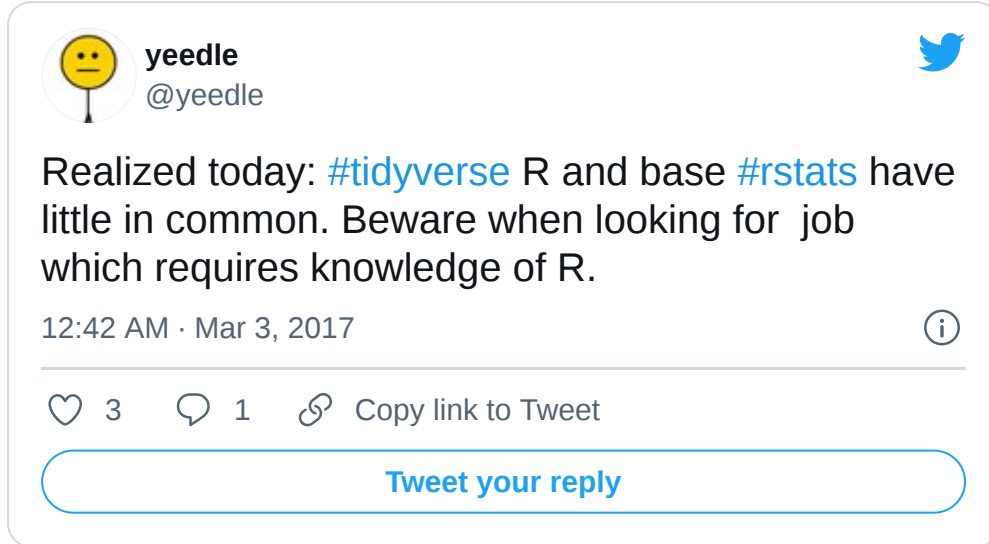
@DavidArenburg this works beautifully. consider making it an official answer. – [Stephen Turner](#) 13 hours ago

- See the popularity of the [data.table versus dplyr](#) question.

Easily summarized

- [data.table](#) is faster, for less than 10 m rows, negligible.
- [tidyfast](#) for `data.table` speed and `tidyverse` syntax
- [tidytable](#) for `data.table` speed and `tidyverse` syntax
- [poorman](#) for zero dependencies but slow ;)

Criticism, finding a job



Personal complains

- Still young, change quickly but [lifecycle](#)
- Backward compatibility is not always maintained.
- [tibbles](#) are nice, recent embedding of [matrices](#) doesn't solve bioconductor integration
- [rownames](#) still an issue, one must be careful not to loose them

No need for opposition base / tidyverse

Learning the *tidyverse* does not prevent to learn *R base*, it helps to get things done early in the process

Community complaints

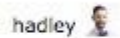


👤 @DirkEddelbuettel no one is calling you a bad person. You're acting unprofessionally by refusing to use official names (of people and packages) but that doesn't make you a bad person

👤 @DirkEddelbuettel your points were sufficiently vague to be unfalsifiable. There seems little point in me disagreeing with them because there are no firm definitions or concrete examples



That makes it religion.

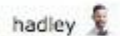


👤 @DirkEddelbuettel the OED defines religion as "The belief in and worship of a superhuman controlling power, especially a personal God or gods." The tidyverse doesn't require belief in or worship of any controlling powers, superhuman or otherwise.



..... Which happens to have an unacceptable tail of depends, is known to break code, and to underperform in timing comparisons.

- 1) Does tidyverse have more depends than other packages, true or false?
- 2) It is known to have broken previously working code, true or false? [And I narrowed this now....]



👤 @DirkEddelbuettel you claim was that it has an unacceptable tail of depends. What is unacceptable?

yst 20:28



3) Is (eg SO) full of timing comparisons where data.table or base run circles around tidyverse, true or false?



👤 @DirkEddelbuettel Yes, the tidyverse has broken previously working code. If that was a criteria to not use a package, there wouldn't be many remaining (including Rcpp)

source: [SO, R chat room, 29 Nov 2017](#)

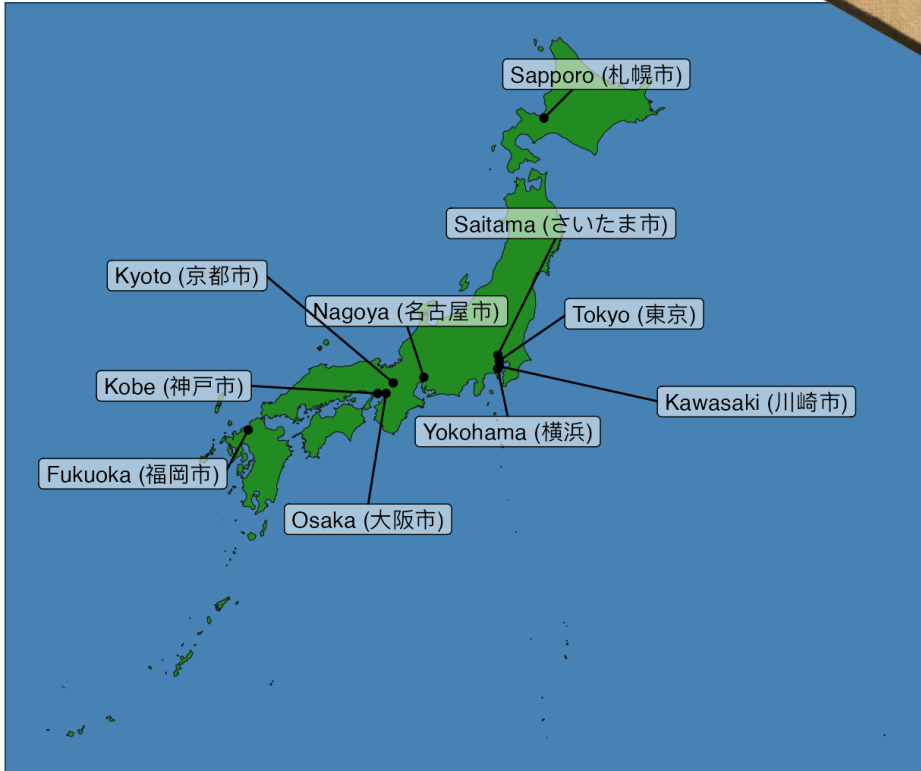
2021 developments

ragg, fonts working across platforms (Thomas Pedersen)



Ragg device (macOS)	Ragg device (Windows)	Ragg device (Linux)
This is English, この文は日本語です 🚀	This is English, この文は日本語です 🚀	This is English, この文は日本語です 🚀
Cairo device (macOS)	Cairo device (Windows)	Cairo device (Linux)
This is English, □□□□□□□□ □	This is English, □□□□□□□□ □	This is English, この文は日本語です 🚀
Quartz device (macOS)	Windows device (Windows)	
This is English, □□□□□□□□ □□		

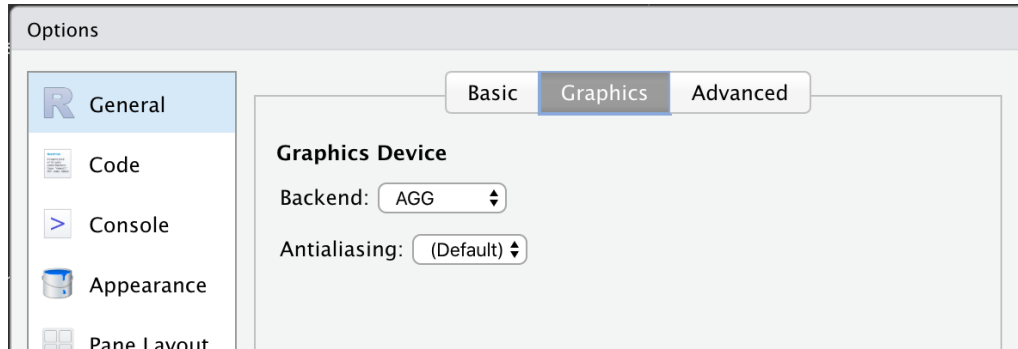
Location of largest cities in Japan (日本) 🇯🇵



Source: [ragg](#), [blog post](#)

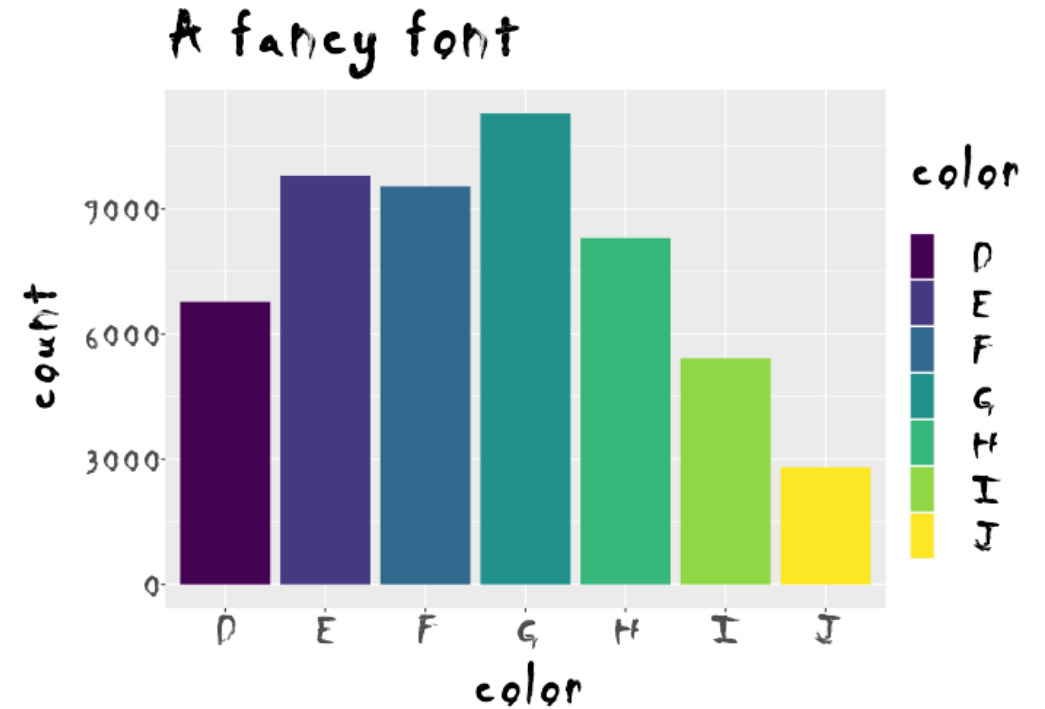
How to use ragg

Use in RStudio



With `knitr`

```
knitr::opts_chunk$set(dev = "ragg_png")
```



Sliding windows (Davis Vaughan)



slider

```
library(slider)
slide_dbl(1:5, ~ mean(.x), .before = 1, .after = 1)
```

```
[1] 1.5 2.0 3.0 4.0 4.5
```

- rolling mean

```
mutate(swiss, roll_agri = slide_mean(Agriculture,
                                     before = 7), .keep = "used")
```

	Agriculture	roll_agri
Courtelary	17.0	17.00000
Delemont	45.1	31.05000
Franches-Mnt	39.7	33.93333
Moutier	36.5	34.57500
Neuveville	43.5	36.36000
Porrentruy	35.3	36.18333
Broye	70.2	41.04286
Glane	67.8	44.38750
Gruyere	53.3	48.92500
Sarine	45.2	48.93750
Veveyse	64.5	52.03750
Aigle	62.0	55.22500
Aubonne	67.5	58.22500
Avenches	60.7	61.40000
Cossonay	69.3	61.28750
Echallens	72.6	61.88750
Grandson	34.0	59.47500

clock

- Explicitly handle invalid dates
- Explicitly handle daylight saving time issues
- Expose naive types for representing date-times without a time zone
- Provide calendar types for representing calendar “dates” in alternative ways
- Provide variable precision date-time types

```
library(clock)
date_seq(date_build(2019, 1), by = duration_months(2), total_si
```

```
[1] "2019-01-01" "2019-03-01" "2019-05-01" "2019-07-01" "2019-
[6] "2019-11-01" "2020-01-01" "2020-03-01" "2020-05-01" "2020-
```

New functions in dplyr, across for filtering



Following the success of `across()`, `if_any()` `if_all()` were developed for filtering

```
library(palmerpenguins)

is_big <- function(x) {
  x > mean(x, na.rm = TRUE)
}

# keep rows if all the selected columns are "big"
filter(penguins,
  if_all(contains("bill"), is_big))
```

```
# A tibble: 61 × 8
  species island bill_length_mm bill_depth_mm flipper_length_mm
  <fct>   <fct>         <dbl>         <dbl>             <dbl>
1 Adelie Torgersen         46             21.5             181
2 Adelie Dream            44.1            19.7             172
3 Adelie Torgersen         45.8            18.9             181
4 Adelie Biscoe           45.6            20.3             190
5 Adelie Torgersen         44.1             18             171
6 Gentoo Biscoe           44.4            17.3             196
7 Gentoo Biscoe           50.8            17.3             192
8 Chinstrap Dream         46.5            17.9             181
9 Chinstrap Dream          50             19.5             188
10 Chinstrap Dream         51.3            19.2             195
# ... with 51 more rows, and 2 more variables: sex <fct>, year <dbl>
```

```
filter(penguins,
  if_any(contains("bill"), is_big))
```

```
# A tibble: 296 × 8
  species island bill_length_mm bill_depth_mm flipper_length_mm
  <fct>   <fct>         <dbl>         <dbl>             <dbl>
1 Adelie Torgersen         39.1            18.7             181
2 Adelie Torgersen         39.5            17.4             172
3 Adelie Torgersen         40.3             18             181
4 Adelie Torgersen         36.7            19.3             190
5 Adelie Torgersen         39.3            20.6             196
6 Adelie Torgersen         38.9            17.8             192
7 Adelie Torgersen         39.2            19.6             181
8 Adelie Torgersen         34.1            18.1             171
9 Adelie Torgersen         42             20.2             195
10 Adelie Torgersen         37.8            17.3             188
# ... with 286 more rows, and 2 more variables: sex <fct>, year <dbl>
```

Source: [blog post](#)

Add-ons to mutate



New experimental `.keep` argument:

- "all", the default, retains all variables.
- "used" keeps any variables used to make new variables
- "unused" keeps only existing variables not used to make new variables.
- "none", only keeps grouping keys (like `transmute()`)

```
group_by(penguins, species) %>%  
mutate(body_mass_kg = body_mass_g / 1000,  
       .keep = "used")
```

```
# A tibble: 344 × 3  
# Groups:   species [3]  
  species body_mass_g body_mass_kg  
  <fct>      <int>      <dbl>  
1 Adelie      3750        3.75  
2 Adelie      3800        3.8  
3 Adelie      3250        3.25  
4 Adelie       NA         NA  
5 Adelie      3450        3.45  
6 Adelie      3650        3.65  
7 Adelie      3625        3.62  
8 Adelie      4675        4.68  
9 Adelie      3475        3.48  
10 Adelie     4250        4.25  
# ... with 334 more rows
```

```
group_by(penguins, species) %>%  
mutate(body_mass_kg = body_mass_g / 1000,  
       .keep = "none")
```

```
# A tibble: 344 × 2  
# Groups:   species [3]  
  species body_mass_kg  
  <fct>      <dbl>  
1 Adelie      3.75  
2 Adelie      3.8  
3 Adelie      3.25  
4 Adelie      NA  
5 Adelie      3.45  
6 Adelie      3.65  
7 Adelie      3.62  
8 Adelie      4.68  
9 Adelie      3.48  
10 Adelie      4.25  
# ... with 334 more rows
```

Add-ons to pull



`pull()` can now output a **named** vector

```
as_tibble(swiss, rownames = "location") %>%  
  pull(Agriculture, name = location)
```

Courtelary	Delemont	Franches-Mnt	Moutier	Neuvevill
17.0	45.1	39.7	36.5	43.
Broye	Glane	Gruyere	Sarine	Veveys
70.2	67.8	53.3	45.2	64.
Aubonne	Avenches	Cossonay	Echallens	Grandso
67.5	60.7	69.3	72.6	34.
La Vallee	Lavaux	Morges	Moudon	Nyon
15.2	73.0	59.8	55.1	50.
Oron	Payerne	Paysd'enhaut	Rolle	Veve
71.2	58.1	63.5	60.8	26.
Conthey	Entremont	Herens	Martigwy	Monthe
85.9	84.9	89.7	78.2	64.
Sierre	Sion	Boudry	La Chauxdfnd	Le Locl
84.6	63.1	38.4	7.7	16.
Val de Ruz	ValdeTravers	V. De Geneve	Rive Droite	Rive Gauch
37.6	18.7	1.2	46.6	27.

Add-ons to joins



- New argument `keep` (TRUE/FALSE) for "both x and y be preserved in the output?"

```
band_members
```

```
# A tibble: 3 × 2
  name band
<chr> <chr>
1 Mick Stones
2 John Beatles
3 Paul Beatles
```

```
band_instruments2
```

```
# A tibble: 3 × 2
  artist plays
<chr> <chr>
1 John guitar
2 Paul bass
3 Keith guitar
```

```
left_join(band_members, band_instruments2,
          by = c("name" = "artist"),
          keep = TRUE)
```

```
# A tibble: 3 × 4
  name band artist plays
<chr> <chr> <chr> <chr>
1 Mick Stones <NA> <NA>
2 John Beatles John guitar
3 Paul Beatles Paul bass
```

```
left_join(band_members, band_instruments2,
          by = c("name" = "artist"),
          keep = FALSE)
```

```
# A tibble: 3 × 3
  name band plays
<chr> <chr> <chr>
1 Mick Stones <NA>
2 John Beatles guitar
3 Paul Beatles bass
```

Should we keep artist column?

coalesce, not new but tends to be forgotten



Keep the first non-missing values

```
coalesce(
  c(1, NA, 3, NA),
  c(NA, 2, 5, 6)
)
```

```
[1] 1 2 3 6
```

Works also in rectangle data

```
tribble(
  ~ a,      ~ b,      ~ c,
  "soil",   NA,       NA,
  NA,       "tree",   "Buch",
  NA,       NA,       "Birch") %>%
  mutate(col = coalesce(a, b, c))
```

```
# A tibble: 3 × 4
  a      b      c      col
<chr> <chr> <chr> <chr>
1 soil <NA> <NA> soil
2 <NA> tree Buch tree
3 <NA> <NA> Birch Birch
```

Before we stop

Wrap up:

- RStudio team is growing every month
- maintain incredible amount of code
- still keep it FOSS (MIT license)

Acknowledgments ☐ ☐

- [Alexandre Courtiol lecture](#)
- [Romain François](#)
- [Lionel Henry](#)
- [Hadley Wickham](#)
- [Jennifer Bryan](#)
- [Jim Hester](#)

Thank you for your attention!