dplyr 1.1.2

dplyr

Overview

dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges:

- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.

These all combine naturally with `group_by()` which allows you to perform any operation "by group". You can learn more about them in `vignette("dplyr")`. As well as these single-table verbs, dplyr also provides a variety of two-table verbs, which you can learn about in `vignette("two-table")`.

If you are new to dplyr, the best place to start is the data transformation chapter in R for data science.

Backends

In addition to data frames/tibbles, dplyr makes working with other computational backends accessible and efficient. Below is a list of alternative backends:

- arrow for larger-than-memory datasets, including on remote cloud storage like AWS S3, using the Apache Arrow C++ engine, Acero.

dplyr 1.1.2

- dbplyr for data stored in a relational database. Translates your dplyr code to SQL.
- duckdb for large datasets that are still small enough to fit on your computer.
- sparklyr for very large datasets stored in Apache Spark.

Installation

```
# The easiest way to get dplyr is to install the whole tidyverse:
install.packages("tidyverse")

# Alternatively, install just dplyr:
install.packages("dplyr")
```

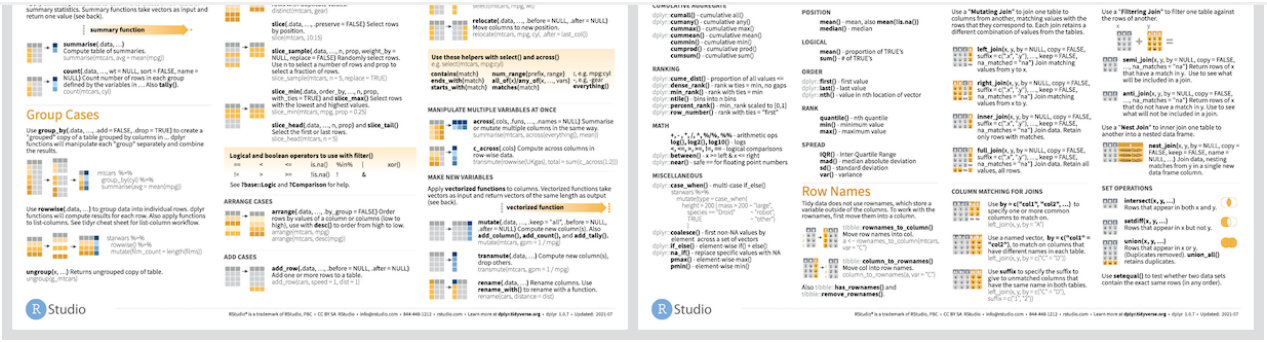
Development version

To get a bug fix or to use a feature from the development version, you can install the development version of dplyr from GitHub.

```
# install.packages("pak")
pak::pak("tidyverse/dplyr")
```

Cheat Sheet

dplyr 1.1.2



Usage

```
library(dplyr)

starwars %>%
  filter(species == "Droid")
#> # A tibble: 6 × 14
#>   name   height  mass hair_color skin_color  eye_color birth_year sex   gen
#>   <chr>   <int> <dbl> <chr>      <chr>       <chr>           <dbl> <chr> <ch
#> 1 C-3PO     167    75 <NA>       gold        yellow            112 none  mas
#> 2 R2-D2      96    32 <NA>       white, blue red                33 none  mas
#> 3 R5-D4      97    32 <NA>       white, red  red                NA none  mas
#> 4 IG-88     200   140 none       metal       red                15 none  mas
#> 5 R4-P17     96    NA none       silver, red red, blue          NA none  fem
#> # ℹ 1 more row
#> # ℹ 5 more variables: homeworld <chr>, species <chr>, films <list>,
#> #   vehicles <list>, starships <list>

starwars %>%
  select(name, ends_with("color"))
#> # A tibble: 87 × 4
#>   name           hair_color skin_color  eye_color
#>   <chr>          <chr>      <chr>       <chr>
#> 1 Luke Skywalker blond      fair        blue
#> 2 C-3PO          <NA>       gold        yellow
#> 3 R2-D2          <NA>       white, blue red
#> 4 Darth Vader    none       white       yellow
#> 5 Leia Organa    brown      light       brown
#> # ℹ 82 more rows
```

dplyr 1.1.2

```
  select(name:mass, bmi)
#> # A tibble: 87 × 4
#>   name           height  mass   bmi
#>   <chr>           <int> <dbl> <dbl>
#> 1 Luke Skywalker    172    77  26.0
#> 2 C-3PO             167    75  26.9
#> 3 R2-D2              96    32  34.7
#> 4 Darth Vader       202   136  33.3
#> 5 Leia Organa       150    49  21.8
#> # ℹ 82 more rows

starwars %>%
  arrange(desc(mass))
#> # A tibble: 87 × 14
#>   name      height  mass hair_color skin_color eye_color birth_year sex   g
#>   <chr>      <int> <dbl> <chr>      <chr>      <chr>           <dbl> <chr> <
#> 1 Jabba De…    175  1358 <NA>       green-tan… orange          600   herm… m
#> 2 Grievous     216   159 none       brown, wh… green, y…        NA   male  m
#> 3 IG-88        200   140 none       metal      red              15   none  m
#> 4 Darth Va…    202   136 none       white      yellow           41.9 male  m
#> 5 Tarfful      234   136 brown      brown      blue             NA   male  m
#> # ℹ 82 more rows
#> # ℹ 5 more variables: homeworld <chr>, species <chr>, films <list>,
#> #   vehicles <list>, starships <list>

starwars %>%
  group_by(species) %>%
  summarise(
    n = n(),
    mass = mean(mass, na.rm = TRUE)
  ) %>%
  filter(
    n > 1,
    mass > 50
  )
#> # A tibble: 8 × 3
#>   species      n  mass
#>   <chr>    <int> <dbl>
#> 1 Droid        6  69.8
```

dplyr 1.1.2

```
#> 5 Mirlatan     2  53.1
#> # ℹ 3 more rows
```

Getting help

If you encounter a clear bug, please file an issue with a minimal reproducible example on GitHub. For questions and other discussion, please use community.rstudio.com or the manipulatr mailing list.

Please note that this project is released with a Contributor Code of Conduct. By participating in this project you agree to abide by its terms.

LINKS

View on CRAN

Browse source code

Report a bug

Learn more

LICENSE

Full license

MIT + file LICENSE

COMMUNITY

Contributing guide

Code of conduct

Getting help

dplyr 1.1.2

DEVELOPERS

Hadley Wickham
Author, maintainer

Romain François
Author

Lionel Henry
Author

Kirill Müller
Author

Davis Vaughan
Author

posit
Copyright holder, funder

Developed by Hadley Wickham, Romain François, Lionel Henry, Kirill Müller, Davis Vaughan, posit.

Site built with pkgdown 2.0.7.