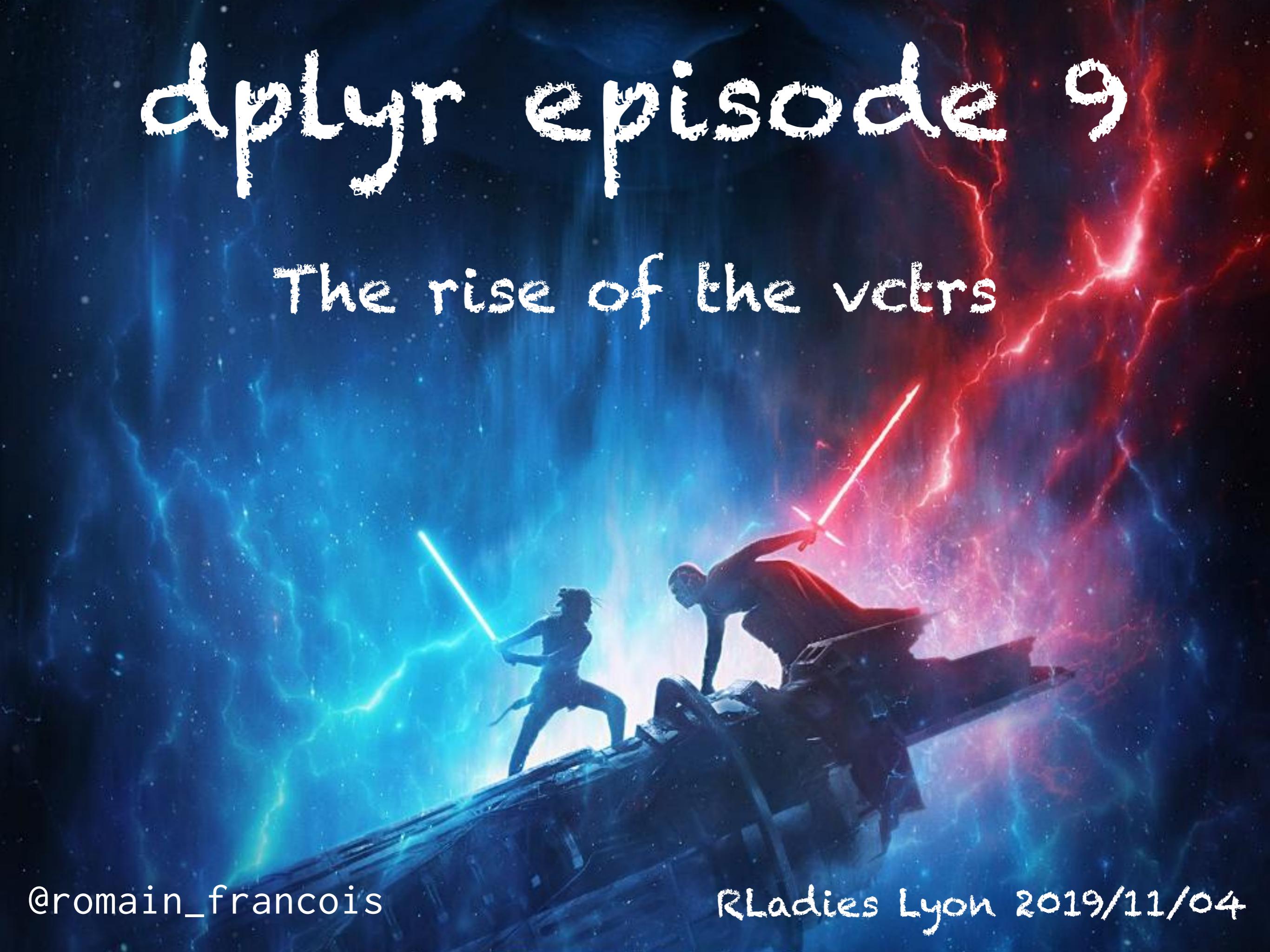


# dplyr episode 9

## The rise of the vctrs



# dplyr episode 9

## summarise() of the vectors



Play (k)

▶ ▶ ⏪ 1:50 / 17:52



)(, Ⓜ )

)(, Ⓜ )

$f( \square \square \square , \ominus )$





Welcome to the Boston useR &  
R-Ladies Boston meetup

Lightning ( ) talks with the tidyverse team

Davis Vaughan  
Gabor Csardi  
Hadley Wickham  
Jenny Bryan

Jim Hester  
Lionel Henry  
Mara Averick  
Max Kuhn

Romain Francois  
Thomas Lin  
Pedersen

dplyr episode 9  
the rise of the vctrs

```
iris %>%  
  group_by(Species) %>%  
  summarise(  
    Sepal.Length = mean(Sepal.Length), ●  
    Sepal.Width = mean(Sepal.Width) ●  
)
```

```
#> # A tibble: 3 x 3  
#>   Species        
#>   <fct>  
#> 1 setosa  
#> 2 versicolor  
#> 3 virginica
```

Species	Sepal.Length	Sepal.Width
setosa	5.01	3.43
versicolor	5.94	2.77
virginica	6.59	2.97

A Star Wars scene showing Stormtroopers on a platform above a hangar bay. In the foreground, a Stormtrooper in black armor stands on the left, looking down. On the right, another Stormtrooper in white armor stands on a circular platform. In the background, a large hangar bay with various vehicles and equipment is visible.

The Empire  
is back

# "tibble" results : packing

```
describe <- function(x) {  
  tibble(mean = mean(x), sd = sd(x))  
}
```

```
iris %>%  
  group_by(Species) %>%  
  summarise(  
    Sepal.Length = describe(Sepal.Length),  
    Sepal.Width = describe(Sepal.Width),  
)
```

```
#> # A tibble: 3 x 3
```

Species	Sepal.Length\$mean <dbl>	\$sd <dbl>	Sepal.Width\$mean <dbl>	\$sd <dbl>
setosa	5.01	0.352	3.43	0.379
versicolor	5.94	0.516	2.77	0.314
virginica	6.59	0.636	2.97	0.322

# splicing packing

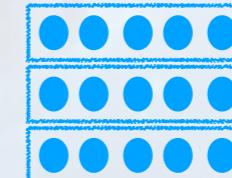
```
quantile(iris$Sepal.Length)
#> 0% 25% 50% 75% 100%
#> 4.3 5.1 5.8 6.4 7.9
tibble(!!!!quantile(iris$Sepal.Length))
#> # A tibble: 1 x 5
#>   `0%` `25%` `50%` `75%` `100%`
#>   <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1     4.3    5.1    5.8    6.4    7.9
```

```
quantibble <- function(x, ...) {
  tibble(!!!!quantile(x, ...))
}
```

```
quantibble(iris$Sepal.Length)
#> # A tibble: 1 x 5
#>   `0%` `25%` `50%` `75%` `100%`
#>   <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1     4.3    5.1    5.8    6.4    7.9
```

```
iris %>%
  group_by(Species) %>%
  summarise(q = quantibble(Sepal.Length))
```

```
#> # A tibble: 3 x 2
#>   Species      q$`0%` $`25%` $`50%` $`75%` $`100%
#>   <fct>        <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
#> 1 setosa       4.3     4.8     5       5.2     5.8
#> 2 versicolor   4.9     5.6     5.9     6.3     7
#> 3 virginica    4.9     6.22    6.5     6.9     7.9
```



packing

```
iris %>%  
  group_by(Species) %>%  
  summarise(q = quantibble(Sepal.Length))  
#> # A tibble: 3 x 2  
#>   Species      q$'0%'  $'25%'  $'50%'  $'75%'  $'100'%  
#>   <fct>        <dbl>     <dbl>     <dbl>     <dbl>     <dbl>  
#> 1 setosa       4.3       4.8       5         5.2       5.8  
#> 2 versicolor  4.9       5.6       5.9       6.3       7  
#> 3 virginica   4.9      6.22      6.5       6.9      7.9
```

Revenge of the Sith  
Attack of the Clones  
Episodes I, II & III  
Star Wars



# auto splice

```
quantibble <- function(x, ...) {  
  tibble(!!!quantile(x, ...))  
}
```

```
iris %>%  
  group_by(Species) %>%  
  summarise(quantibble(Sepal.Length))
```

```
#> # A tibble: 3 x 6
```

```
#>   Species <fct>
```

```
#> 1 setosa
```

```
#> 2 versicolor
```

```
#> 3 virginica
```

Species	`0%`	`25%`	`50%`	`75%`	`100%`
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 setosa	4.3	4.8	5	5.2	5.8
2 versicolor	4.9	5.6	5.9	6.3	7
3 virginica	4.9	6.22	6.5	6.9	7.9

# packing

```
iris %>%
  group_by(Species) %>%
  summarise(model = broom::tidy(lm(Sepal.Length ~ Sepal.Width)))
```

#> # A tibble: 6 x 2

Species	model\$term	\$estimate	\$std.error	\$statistic	\$p.value
setosa	(Intercept)	2.64	0.310	8.51	3.74e-11
setosa	Sepal.Width	0.690	0.0899	7.68	6.71e-10
versicolor	(Intercept)	3.54	0.563	6.29	9.07e- 8
versicolor	Sepal.Width	0.865	0.202	4.28	8.77e- 5
virginica	(Intercept)	3.91	0.757	5.16	4.66e- 6
virginica	Sepal.Width	0.902	0.253	3.56	8.43e- 4

# auto splice

```
iris %>%
  group_by(Species) %>%
  summarise(broom::tidy(lm(Sepal.Length ~ Sepal.Width)))
```

#> # A tibble: 6 x 6

Species	term	estimate	std.error	statistic	p.value
setosa	(Intercept)	2.64	0.310	8.51	3.74e-11
setosa	Sepal.Width	0.690	0.0899	7.68	6.71e-10
versicolor	(Intercept)	3.54	0.563	6.29	9.07e- 8
versicolor	Sepal.Width	0.865	0.202	4.28	8.77e- 5
virginica	(Intercept)	3.91	0.757	5.16	4.66e- 6
virginica	Sepal.Width	0.902	0.253	3.56	8.43e- 4

across  
Quake



```
summarise(  
  across(<selection>, <function>  
)
```

across()  
1 function

```
iris %>%
  group_by(Species) %>%
  summarise(across(starts_with("Sepal"), mean))
#> # A tibble: 3 x 3
#>   Species      Sepal.Length  Sepal.Width
#>   <fct>          <dbl>        <dbl>
#> 1 setosa         5.01         3.43
#> 2 versicolor    5.94         2.77
#> 3 virginica     6.59         2.97
```

across()  
1 Lambda

```
iris %>%
  group_by(Species) %>%
  summarise(across(starts_with("Sepal"), ~mean(.)))
#> # A tibble: 3 x 3
#>   Species      Sepal.Length  Sepal.Width
#>   <fct>          <dbl>        <dbl>
#> 1 setosa         5.01         3.43
#> 2 versicolor    5.94         2.77
#> 3 virginica     6.59         2.97
```

across()  
1 function

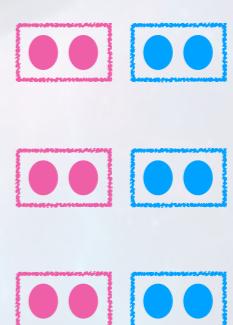
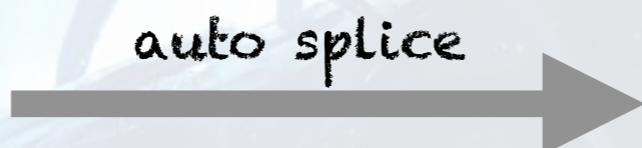
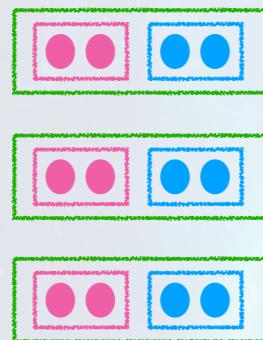
```
iris %>%
  group_by(Species) %>%
  summarise(
    across(starts_with("Sepal"), mean),
    across(starts_with("Petal"), median)
  )
#> # A tibble: 3 × 5
#>   Species      Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
#>   <fct>          <dbl>        <dbl>         <dbl>        <dbl>
#> 1 setosa           5.01         3.43          1.5          0.2
#> 2 versicolor       5.94         2.77          4.35         1.3
#> 3 virginica        6.59         2.97          5.55         2
```

```
summarise(  
  across( <selection>, <list of fns>  
)
```

# across()

function List  
"packed" by function

```
iris %>%
  group_by(Species) %>%
  summarise(
    across(starts_with("Sepal"), list(mean = mean, sd = sd)))
#> # A tibble: 3 x 3
#>   Species      mean$Sepal.Length $Sepal.Width    sd$Sepal.Length $Sepal.Width
#>   <fct>          <dbl>           <dbl>          <dbl>           <dbl>
#> 1 setosa         5.01            3.43          0.352          0.379
#> 2 versicolor     5.94            2.77          0.516          0.314
#> 3 virginica      6.59            2.97          0.636          0.322
```



# across()

# + tidyverse::unpack()

```
iris %>%  
  group_by(Species) %>%  
  summarise(  
    across(starts_with("Sepal"), list(mean = mean, sd = sd))  
  ) %>%  
  tidyverse::unpack(c(mean, sd), names_sep = "_")  
#> # A tibble: 3 x 5  
#>   Species  mean_Sepal.Length  mean_Sepal.Width  sd_Sepal.Length  sd_Sepal.Width  
#>   <fct>          <dbl>            <dbl>            <dbl>            <dbl>  
#> 1 setosa           5.01             3.43            0.352            0.379  
#> 2 versico...        5.94             2.77            0.516            0.314  
#> 3 virginici...     6.59             2.97            0.636            0.322
```



# across()

Manual packing

Single function returning a data frame

```
iris %>%  
  group_by(Species) %>%  
  summarise(  
    across(  
      starts_with("Sepal"),  
      ~ tibble(mean = mean(.x), sd = sd(.x))  
    )  
  )  
  
#> # A tibble: 3 x 3  
#>   Species Sepal.Length$mean $sd Sepal.Width$mean $sd  
#>   <fct>     <dbl> <dbl>       <dbl> <dbl>  
#> 1 setosa      5.01  0.352      3.43  0.379  
#> 2 versicolor  5.94  0.516      2.77  0.314  
#> 3 virginica   6.59  0.636      2.97  0.322
```

# across()

single function

```
iris %>%  
  group_by(Species) %>%  
  summarise(  
    across(starts_with("Sepal"),  
           ~quantibble(.x, probs = c(.25, .5, .75)))  
)
```

```
#> # A tibble: 3 x 3
```

Species	Sepal.Length\$`25%`	\$`50%`	\$`75%`	Sepal.Width\$`25%`	\$`50%`	\$`75%`
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 setosa	4.8	5	5.2	3.2	3.4	3.68
2 versicolor	5.6	5.9	6.3	2.52	2.8	3
3 virginica	6.22	6.5	6.9	2.8	3	3.18

# Data Frames as Vectors of Rows

Recently, I have been working a lot on {vctrs}. This package is an attempt to abstract the atomic types in R, such as integer, character and double, alongside the recursive types of list and data.frame, to extract a set of common principles. From this analysis, a growing toolkit of functions for working with vector types has been built around two themes of *size* and *prototype*. vctrs is a *fun* package to work on, and even more fun to build on top of.

[http://bit.ly/vctrs\\_rows](http://bit.ly/vctrs_rows)

The goal of this post is two fold. First, I want to show off a few functions and packages that have been built using this new toolkit which contribute to why I think this package is so fun. Second, I'd like to introduce a shift in the way you might normally think about data frames, *from a vector of columns to a vector of rows*.

If you've never heard of vctrs before, there's a reason for that. For the most part, it's a developer focused package, and honestly if you never knew this package existed, but it's the foundation for several level packages that were built on top of it, and we've done our job. A few packages that rely heavily on vctrs are now available.



vctrs: Tools for making size and type  
consistent functions - Hadley Wickham

HADLEY WICKHAM

vctrs is a new package that provides tools (cognitive and computational) to ensure that functions behave consistently with respect to inputs of varying length and type. The end goal of vctrs is to be invisible to the end user



Questions?

# Experimental helpers

```
pack_by <- rlang::list2
pack_in <- function(...) {
  exprs <- map(rlang::list2(...), ~expr((!!!x)(.)))
  expr <- expr(tibble(!!!exprs))

  rlang::new_function(alist(.=), expr)
}

f <- pack_in(mean = mean, sd = sd)
f
#> function (.)
#> tibble(mean = <mean>(.), sd = <sd>(.))
#> <environment: 0x7fb58f7d5c78>
f(iris$Sepal.Length)
#> # A tibble: 1 x 2
#>   mean    sd
#>   <dbl> <dbl>
#> 1 5.84  0.828
```

## pack\_by()

```
iris %>%
  group_by(Species) %>%
  summarise(
    across(starts_with("Sepal"), pack_by(mean = mean, sd = sd))
  )
#> # A tibble: 3 x 3
#>   Species      mean$Sepal.Length $Sepal.Width sd$Sepal.Length $Sepal.Width
#>   <fct>          <dbl>            <dbl>           <dbl>            <dbl>
#> 1 setosa        5.01             3.43            0.352            0.379
#> 2 versicolor    5.94             2.77            0.516            0.314
#> 3 virginica     6.59             2.97            0.636            0.322
```

## pack\_in()

```
iris %>%
  group_by(Species) %>%
  summarise(
    across(starts_with("Sepal"), pack_in(mean = mean, sd = sd))
  )
#> # A tibble: 3 x 3
#>   Species      Sepal.Length$mean     $sd  Sepal.Width$mean     $sd
#>   <fct>          <dbl> <dbl>           <dbl> <dbl>
#> 1 setosa        5.01  0.352            3.43  0.379
#> 2 versicolor    5.94  0.516            2.77  0.314
#> 3 virginica     6.59  0.636            2.97  0.322
```