YOU

POLICE

DuckDB
d plyr

Kirill Müller, R/Pharma 2025

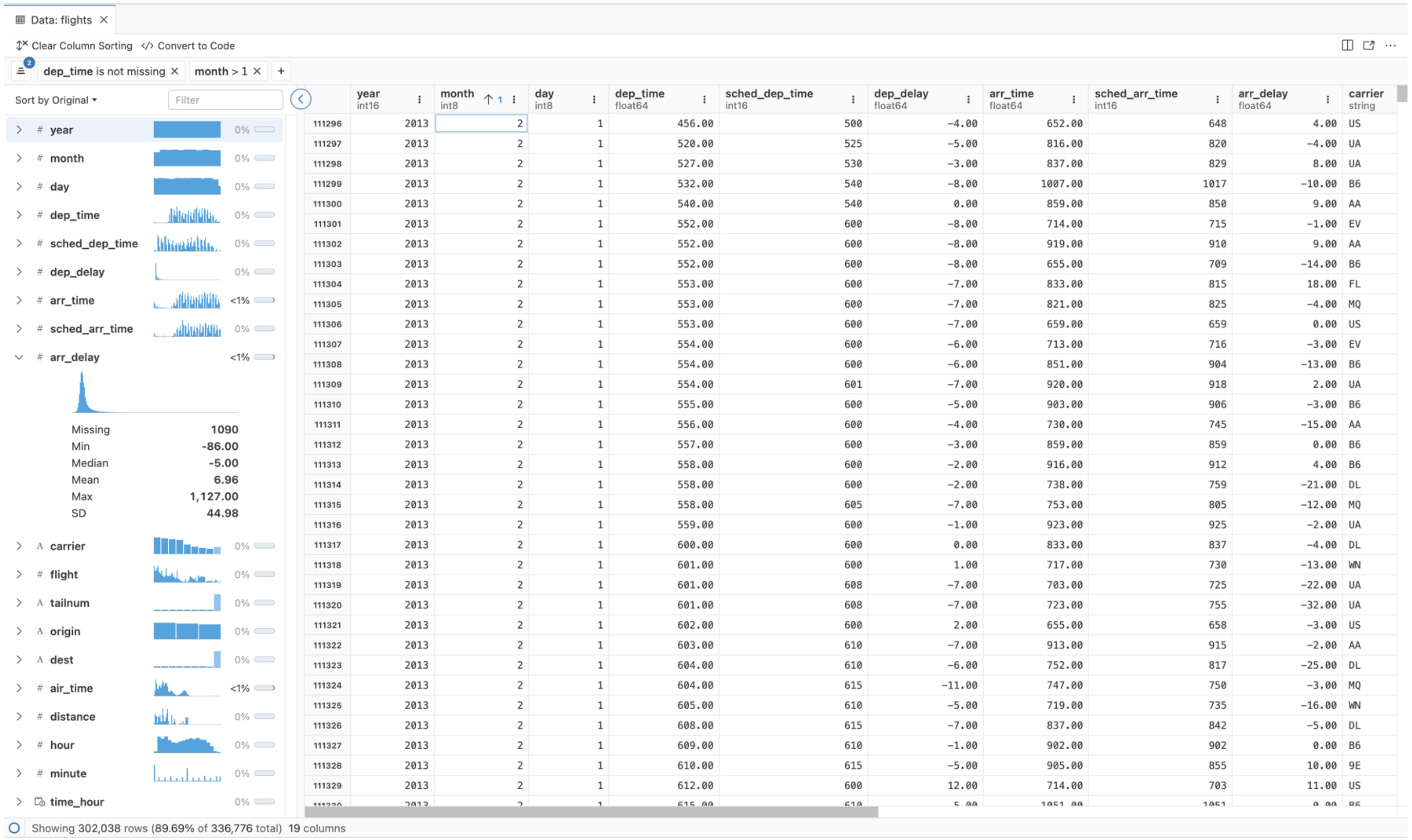
cynkra

posit™

DuckDB

Embedded

Database



		Missing
#	year	0%
#	month	0%
#	day	0%
#	dep_time	0%
#	sched_dep_time	0%
#	dep_delay	0%
#	arr_time	<1%
#	sched_arr_time	0%
#	arr_delay	<1%
A	carrier	0%
#	flight	0%
A	tailnum	0%
A	origin	0%
A	dest	0%
#	air_time	<1%
#	distance	0%
#	hour	0%
#	minute	0%
	time_hour	0%

arr_delay

Missing	1090
Min	-86.00
Median	-5.00
Mean	6.96
Max	1,127.00
SD	44.98

	year int16	month ↑ 1 int8	day int8	dep_time float64	sched_dep_time int16	dep_delay float64	arr_time float64	sched_arr_time int16	arr_delay float64	carrier string
111296	2013	2	1	456.00	500	-4.00	652.00	648	4.00	US
111297	2013	2	1	520.00	525	-5.00	816.00	820	-4.00	UA
111298	2013	2	1	527.00	530	-3.00	837.00	829	8.00	UA
111299	2013	2	1	532.00	540	-8.00	1007.00	1017	-10.00	B6
111300	2013	2	1	540.00	540	0.00	859.00	850	9.00	AA
111301	2013	2	1	552.00	600	-8.00	714.00	715	-1.00	EV
111302	2013	2	1	552.00	600	-8.00	919.00	910	9.00	AA
111303	2013	2	1	552.00	600	-8.00	655.00	709	-14.00	B6
111304	2013	2	1	553.00	600	-7.00	833.00	815	18.00	FL
111305	2013	2	1	553.00	600	-7.00	821.00	825	-4.00	MQ
111306	2013	2	1	553.00	600	-7.00	659.00	659	0.00	US
111307	2013	2	1	554.00	600	-6.00	713.00	716	-3.00	EV
111308	2013	2	1	554.00	600	-6.00	851.00	904	-13.00	B6
111309	2013	2	1	554.00	601	-7.00	920.00	918	2.00	UA
111310	2013	2	1	555.00	600	-5.00	903.00	906	-3.00	B6
111311	2013	2	1	556.00	600	-4.00	730.00	745	-15.00	AA
111312	2013	2	1	557.00	600	-3.00	859.00	859	0.00	B6
111313	2013	2	1	558.00	600	-2.00	916.00	912	4.00	B6
111314	2013	2	1	558.00	600	-2.00	738.00	759	-21.00	DL
111315	2013	2	1	558.00	605	-7.00	753.00	805	-12.00	MQ
111316	2013	2	1	559.00	600	-1.00	923.00	925	-2.00	UA
111317	2013	2	1	600.00	600	0.00	833.00	837	-4.00	DL
111318	2013	2	1	601.00	600	1.00	717.00	730	-13.00	WN
111319	2013	2	1	601.00	608	-7.00	703.00	725	-22.00	UA
111320	2013	2	1	601.00	608	-7.00	723.00	755	-32.00	UA
111321	2013	2	1	602.00	600	2.00	655.00	658	-3.00	US
111322	2013	2	1	603.00	610	-7.00	913.00	915	-2.00	AA
111323	2013	2	1	604.00	610	-6.00	752.00	817	-25.00	DL
111324	2013	2	1	604.00	615	-11.00	747.00	750	-3.00	MQ
111325	2013	2	1	605.00	610	-5.00	719.00	735	-16.00	WN
111326	2013	2	1	608.00	615	-7.00	837.00	842	-5.00	DL
111327	2013	2	1	609.00	610	-1.00	902.00	902	0.00	B6
111328	2013	2	1	610.00	615	-5.00	905.00	855	10.00	9E
111329	2013	2	1	612.00	600	12.00	714.00	703	11.00	US

Showing 302,038 rows (89.69% of 336,776 total) 19 columns

DuckDB vs. SQLite

Embedded

Database

Analytical

DuckDB

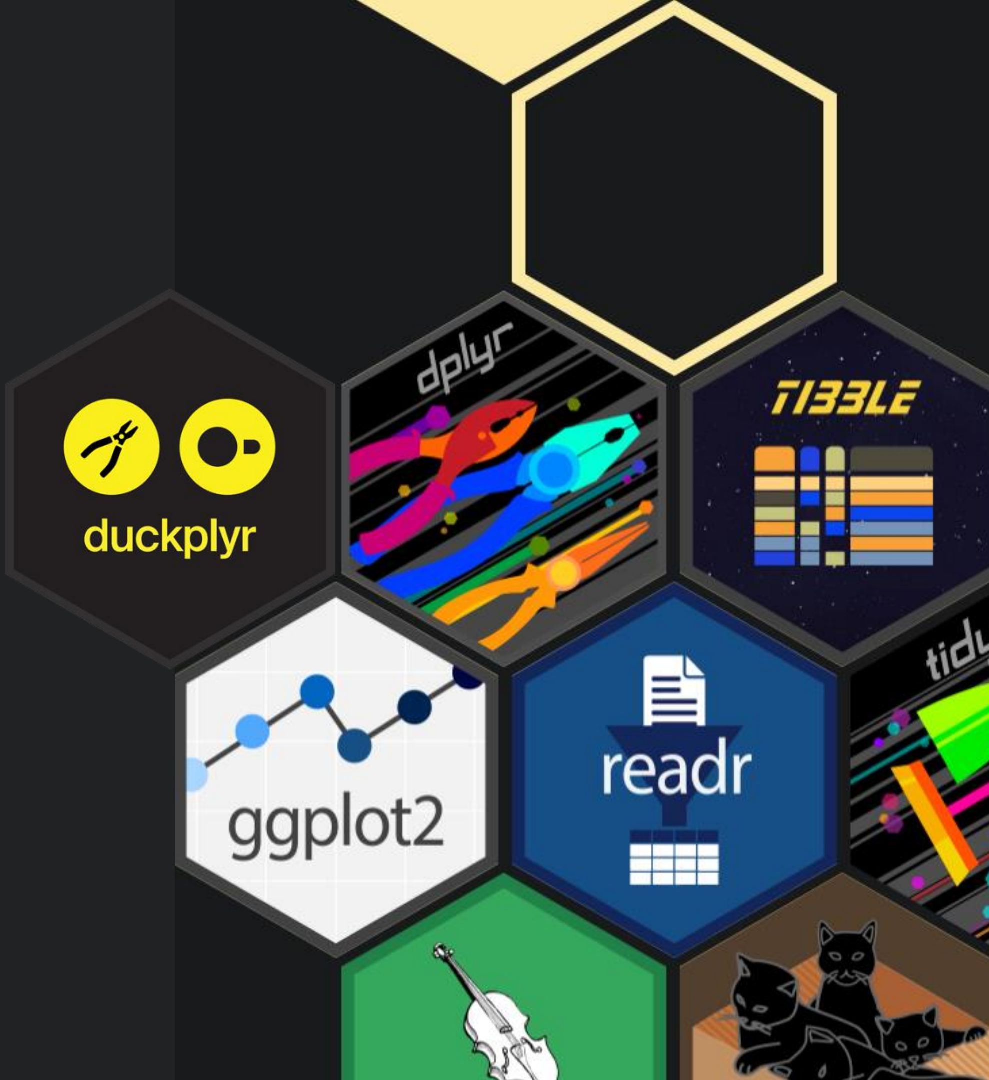
- SQLite for analytics
- Command line interface
- Python/Rust/WASM/… package
- R package

Large

Fast

Featureful

duckplyr


duckplyr
dplyr
TIBBLE
ggplot2
readr

DuckDB interfaces

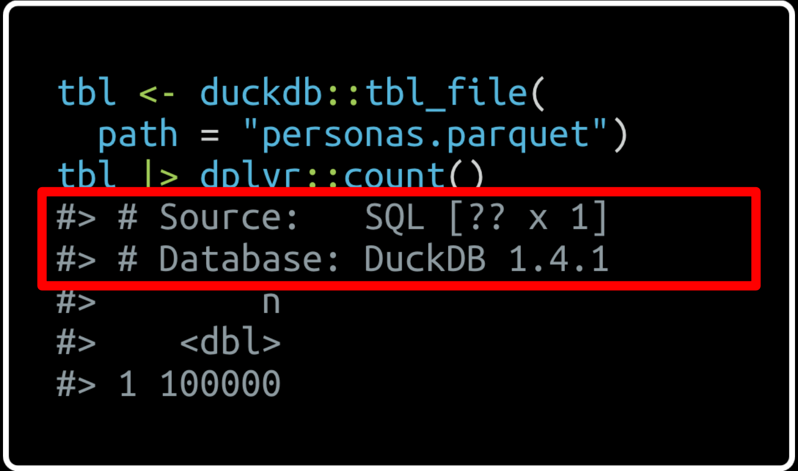
dplyr-like	dbplyr	duckplyr
SQL	DBI SQL	DuckDB

```sql
SELECT COUNT(*)
FROM 'personas.parquet';
-- ┌──────────────┐
-- │ count_star() │
-- │    int64     │
-- ├──────────────┤
-- │    100000    │
-- └──────────────┘
```

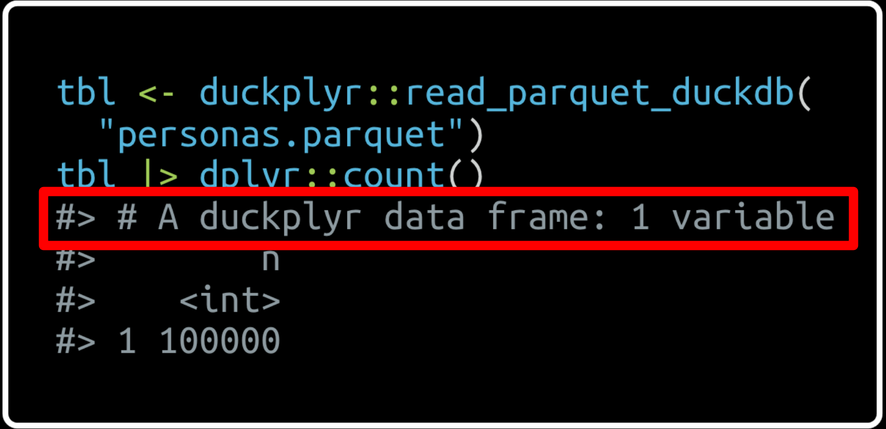
```r
tbl <- duckdb::tbl_file(
  path = "personas.parquet")
tbl |> dplyr::count()
#> # Source:   SQL [?? x 1]
#> # Database: DuckDB 1.4.1
#>        n
#>    <dbl>
#> 1 100000
```

```r
library(DBI)
con <- dbConnect(duckdb::duckdb())
dbGetQuery(
  con,
  "SELECT COUNT(*)
   FROM 'personas.parquet'")
#>   count_star()
#> 1        1e+05
```

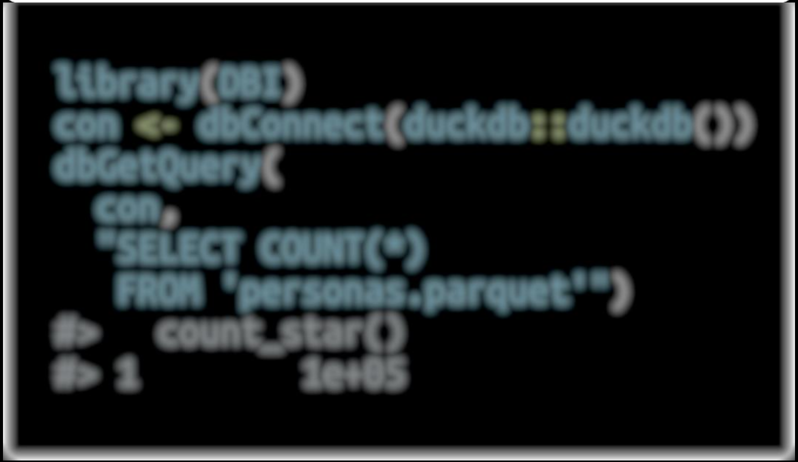
```sql
SELECT COUNT(*)
FROM 'personas.parquet';
-- ┌──────────────┐
-- │ count_star() │
-- │    int64     │
-- ├──────────────┤
-- │    100000    │
-- └──────────────┘
```



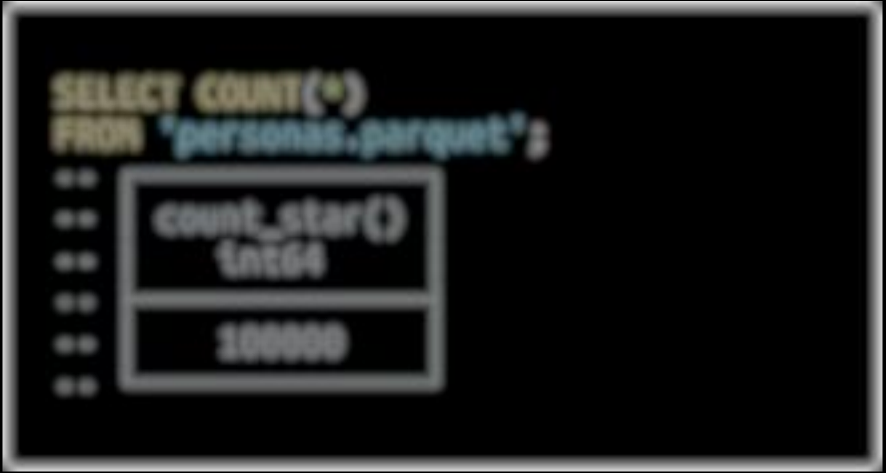
```
tbl <- duckdb::tbl_file(
  path = "personas.parquet")
tbl |> dplyr::count()
#> # Source:   SQL [?? x 1]
#> # Database: DuckDB 1.4.1
#>          n
#>      <dbl>
#> 1 100000
```



```
tbl <- duckplyr::read_parquet_duckdb(
  "personas.parquet")
tbl |> dplyr::count()
#> # A duckplyr data frame: 1 variable
#>          n
#>      <int>
#> 1 100000
```

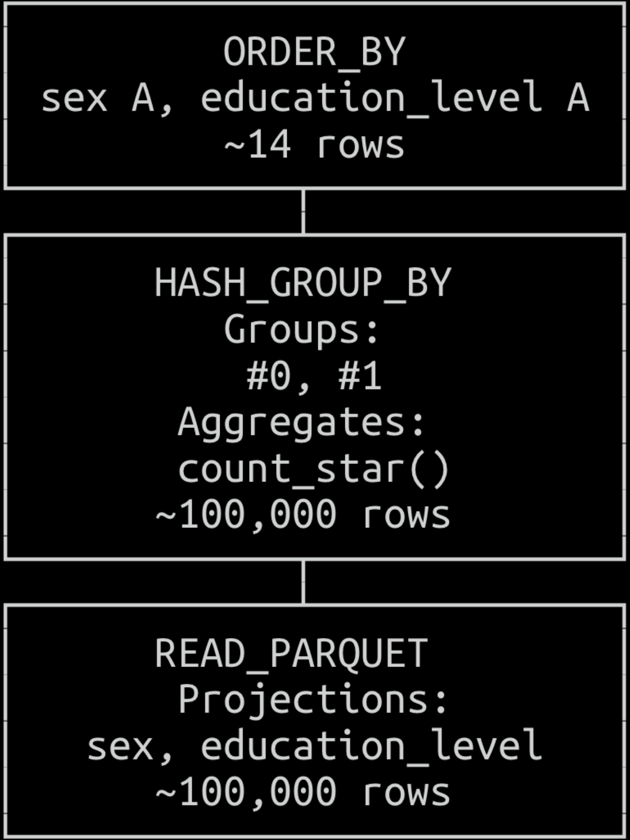


```
library(DBI)
con <- dbConnect(duckdb::duckdb())
dbGetQuery(
  con,
  "SELECT COUNT(*)
   FROM 'personas.parquet'")
#>   count_star()
#> 1        1e+05
```



```
SELECT COUNT(*)
FROM 'personas.parquet';
-- ┌──────────────┐
-- │ count_star() │
-- │    int64     │
-- ├──────────────┤
-- │    100000    │
-- └──────────────┘
```

```r
library(duckplyr)

personas <- read_parquet_duckdb(
  "s3://my-bucket/.../*.parquet"
)

personas_count <-
  personas |>
  count(sex, edu)

personas_count |>
  explain()
```



```
              ORDER_BY
  sex A, education_level A
            ~14 rows

           HASH_GROUP_BY
              Groups:
              #0, #1
            Aggregates:
            count_star()
          ~100,000 rows

           READ_PARQUET
            Projections:
        sex, education_level
          ~100,000 rows
```

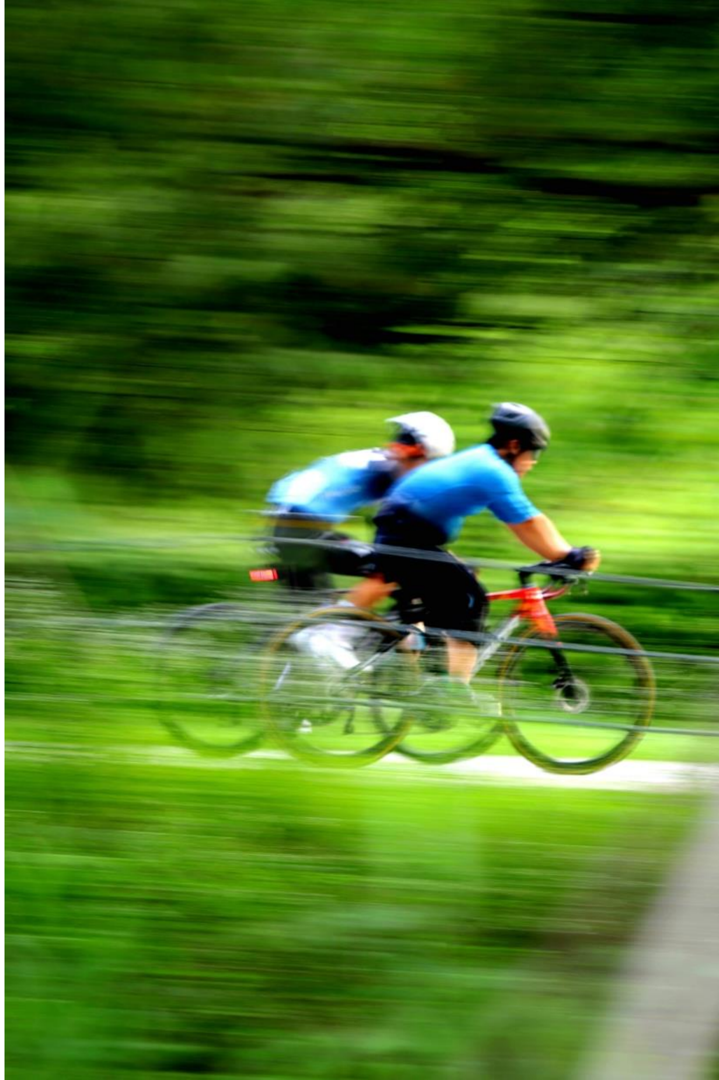
```r
library(duckplyr)

personas <- tibble(
  sex = rep(1:2, each = 7),
  edu = rep(1:7, times = 2))

personas_count <-
  personas |>
  count(sex, edu)

personas_count |>
  explain()
```

```
┌───────────────────────────┐
│          ORDER_BY         │
│      sex A, edu A         │
│         14 rows           │
└─────────────┬─────────────┘
┌─────────────┴─────────────┐
│       HASH_GROUP_BY       │
│          Groups:          │
│           #0, #1          │
│        Aggregates:        │
│        count_star()       │
│          14 rows          │
└─────────────┬─────────────┘
┌─────────────┴─────────────┐
│      R_DATAFRAME_SCAN     │
│        Projections:       │
│          sex, edu         │
│          14 rows          │
└───────────────────────────┘
```

```r
library(tidyverse)



G <- 200
N <- 5000

data <-
  crossing(g1 = 1:G, g2 = 1:G, id = 1:N) |>
  mutate(x = rnorm(n()))

data |>

  summarize(.by = c(g1, g2), m = mean(x)) |>

  ...
#>     user  system elapsed
#>    2.397   1.720   5.172
```

Existing pipelines

- All in
- Best effort
- Targeted



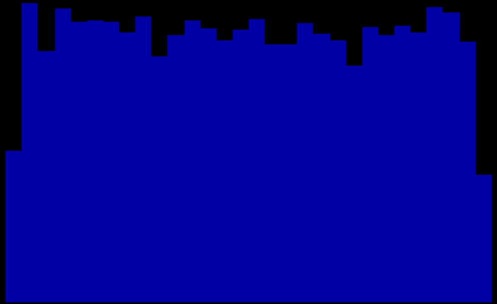
```
library(tidyverse)
library(duckplyr)
#> ✔ Overwriting dplyr methods with duckplyr methods.
#> ℹ Turn off with `duckplyr::methods_restore()`.

G <- 200
N <- 5000

data <-
  crossing(g1 = 1:G, g2 = 1:G, id = 1:N) |>
  mutate(x = rnorm(n()))

data |>

  summarize(.by = c(g1, g2), m = mean(x)) |>

  ...
#>    user  system elapsed
#>   4.073   0.013   0.544
```

```
library(tidyverse)
library(duckplyr)
#> ✔ Overwriting dplyr methods with duckplyr methods.
#> i Turn off with `duckplyr::methods_restore()`.
methods_restore()
#> i Restoring dplyr methods.


data <-
  crossing(g1 = 1:G, g2 = 1:G, id = 1:N) |>
  mutate(x = rnorm(n()))

data |>
  as_duckdb_tibble(prudence = "stingy") |>
  summarize(.by = c(g1, g2), m = mean(x)) |>
  collect() |>
  ...
#>    user  system elapsed
#>   4.142   0.017   0.659
```

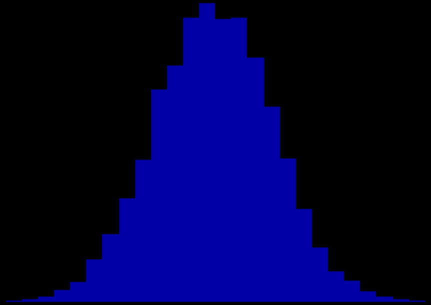
duckplyr

- Output: Proper data frames
- Input: files, data frames, …
- Fallback, translation
- Full DuckDB power

```r
library(tidyverse)

idx <- duckplyr::duckdb_tibble(id = 1:1e4)



idx |>
  mutate(x = dd::random()) |>
  ggplot(aes(x)) +
  geom_histogram()
```



```r
library(tidyverse)

idx <- duckplyr::duckdb_tibble(id = 1:1e4)

duckplyr::db_exec("LOAD stochastic")

idx |>
  mutate(x = dd::dist_normal_sample(0, 1)) |>
  ggplot(aes(x)) +
  geom_histogram()
```



```r
library(duckplyr)

db_exec("INSTALL read_stat")
db_exec("LOAD read_stat")

survey <- read_file_duckdb(
  "survey.sas7bdat",
  "read_stat"
)

survey |>
  count(SEX)
## # A duckplyr data frame
## ...
```

```
install.packages("duckplyr")
library(duckplyr)
methods_restore()
db_exec("INSTALL read_stat")
```

tinyurl.com/duckplyr-25-rp

cynkra