

Package ‘summarisebig’

September 22, 2026

Type Package

Title Grouped Summaries for Large 'arrow' Datasets

Version 0.1.0

Description Provides a 'dplyr'-like interface for grouped summaries on large 'arrow' datasets. Complete summaries are attempted in 'arrow' first. When a result can be reconstructed from 'arrow'-computable sufficient statistics, an explicit MapReduce-style reduction and 'R' finalization strategy is available. For arbitrary 'R' functions that require raw group observations, complete groups are materialized in bounded chunks, with optional parallel execution and shared-memory processing. The MapReduce strategy follows the programming model described by Dean and Ghemawat (2008) [<doi:10.1145/1327452.1327492>](https://doi.org/10.1145/1327452.1327492).

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.3.0)

Imports arrow, dplyr, future, future.mirai, futurize, furrr (>= 0.4.0), purrr, rlang

Suggests knitr, mori, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

VignetteBuilder knitr

Config/Needs/website pkgdown

URL <https://github.com/larry77/summarisebig>,
<https://larry77.github.io/summarisebig/>

BugReports <https://github.com/larry77/summarisebig/issues>

NeedsCompilation no

Config/roxygen2/version 8.1.0

Author Lorenzo Isella [aut, cre]

Maintainer Lorenzo Isella <lorenzo.isella@gmail.com>

Repository CRAN

Date/Publication 2026-09-21 22:30:02 UTC

Contents

summarise_big	2
Index	5

summarise_big	<i>Summarise a large Arrow dataset by group</i>
---------------	---

Description

summarise_big() is designed for grouped computations on Arrow datasets when the result may be expressible in Arrow, reconstructible from compact Arrow-computable state, or require arbitrary R code on complete groups.

Usage

```
summarise_big(  
  .data,  
  ...,  
  .by,  
  .order_by = NULL,  
  .workers = 2L,  
  .strategy = c("parallel_chunks", "shared_chunk", "map_reduce"),  
  .chunk_rows = 1e+06,  
  .task_rows = 1e+05,  
  .oversize = c("error", "warning"),  
  .map_reduce = NULL,  
  .finalize = NULL,  
  .seed = TRUE,  
  .try_arrow = TRUE,  
  .tmp = tempfile("summarise-big-"),  
  .keep_tmp = FALSE  
)
```

Arguments

- .data An Arrow Dataset or Arrow object supporting dplyr operations.
- ... Named grouped summary expressions for the ordinary strategies.
- .by A single grouping column.
- .order_by Optional character vector of columns defining deterministic order within groups for order-sensitive R summaries.
- .workers Number of local Mirai workers. Parallel package tests should use at most two workers. Defaults to 2.
- .strategy One of "parallel_chunks", "shared_chunk", or "map_reduce".
- .chunk_rows Target maximum number of rows per materialized disk chunk. Complete groups are never split.

<code>.task_rows</code>	Target rows per worker task within <code>shared_chunk</code> . Complete groups are never split.
<code>.oversize</code>	What to do when one group exceeds <code>.chunk_rows</code> .
<code>.map_reduce</code>	Named list of Arrow-computable grouped reduction expressions, normally supplied as one-sided formulas.
<code>.finalize</code>	R function applied to the compact table produced by <code>.map_reduce</code> . It must return a data frame.
<code>.seed</code>	Passed to <code>futurize::futurize()</code> for parallel random-number handling.
<code>.try_arrow</code>	If TRUE, ordinary strategies attempt the complete summary in Arrow before materializing any raw groups.
<code>.tmp</code>	Temporary directory used for group-safe Parquet repartitioning.
<code>.keep_tmp</code>	If TRUE, retain the temporary dataset after completion.

Details

The ordinary strategies first attempt the complete summary lazily in Arrow. If Arrow cannot execute it, `parallel_chunks` materializes different group-safe chunks independently, while `shared_chunk` materializes one chunk at a time and can share it between workers with `mori`.

With `strategy = "map_reduce"`, Arrow computes the expressions supplied in `.map_reduce`, only that reduced table is collected, and `.finalize` runs in ordinary R.

Value

A data frame containing grouped summary results.

Examples

```
tab <- arrow::Table$create(data.frame(
  grp = c("a", "a", "b", "b"),
  x = c(1, 2, 10, 20)
))

# Arrow fast path.
summarise_big(tab, result = mean(x), .by = grp)

# Arrow reduction followed by an ordinary R finalizer.
summarise_big(
  tab,
  .by = grp,
  .strategy = "map_reduce",
  .map_reduce = list(n = ~ dplyr::n(), sx = ~ sum(x)),
  .finalize = function(d) {
    dplyr::mutate(d, result = sx / n)
  }
)

# Arbitrary R function on complete groups in a Parquet Dataset.
# This example is self-contained, but starts parallel workers and therefore
```

```
# may take more than a few seconds on some systems.
custom_slope <- function(x, y) {
  unname(stats::coef(stats::lm(y ~ x))[2])
}

path <- tempfile("summarisebig-example-")
arrow::write_dataset(
  data.frame(
    grp = rep(c("a", "b"), each = 4),
    x = rep(1:4, 2),
    y = c(2, 4, 6, 8, 1, 2, 4, 8)
  ),
  path
)
ds <- arrow::open_dataset(path)

summarise_big(
  ds,
  result = custom_slope(x, y),
  .by = grp,
  .strategy = "parallel_chunks",
  .workers = 2,
  .chunk_rows = 4
)

rm(ds)
gc()
unlink(path, recursive = TRUE, force = TRUE)
```

Index

summarise_big, [2](#)