Package 'enrollcast'

August 26, 2026

Title Project School Enrollment with Grade Progression Ratios

Version 0.1.0

Description Projects school enrollment using the cohort survival / grade progression ratio method described in Webster (1970) <doi:10.1080/00220973.1970.11011238>, implemented as a matrix projection. Works at any level of aggregation and any number of grades. Provides functions to compute progression ratios from historical grade-level enrollment and to project future enrollment forward an arbitrary horizon.

License MIT + file LICENSE

URL https://rorylawless.r-universe.dev/enrollcast, https://github.com/localopen/enrollcast, https://localopen.github.io/enrollcast/

BugReports https://github.com/localopen/enrollcast/issues

Depends R (>= 4.0)

Imports cli (>= 3.4.0), rlang (>= 1.0.0), stats

Suggests knitr, rmarkdown, testthat (>= 3.2.0)

VignetteBuilder knitr

Config/Needs/website pkgdown

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Encoding UTF-8

NeedsCompilation no

Author Rory Lawless [aut, cre, cph] (ORCID: <https://orcid.org/0009-0003-7688-309X>)

Maintainer Rory Lawless <rory@rorylawless.com>

Repository CRAN

Date/Publication 2026-08-26 20:00:02 UTC

Contents

`progression_matrix` *Build the projection matrix*

Description

Assembles the projection matrix used to advance enrollment. Progression ratios are placed on the sub-diagonal (each non-entry grade is fed by the grade below); the entry-grade row is left at zero because entry enrollment is supplied exogenously to `project_enrollment()`. The ratios must form a single low-to-high chain: each `grade_to` must be the grade immediately above its `grade_from` in the resolved order.

Usage

```
progression_matrix(ratios, grade_order = NULL)
```

Arguments

`ratios`	A data frame or data-frame subclass with columns `grade_from`, `grade_to`, and `ratio`, as returned by `progression_ratios()`. `grade_from` and `grade_to` must not be missing. `ratio` must be numeric, non-negative, and finite; an infinite ratio (from a zero-enrollment feeder) is rejected, while `NA`/`NaN` ratios (e.g. from sparse history) are kept in the matrix with a warning.
`grade_order`	Optional character vector giving the low-to-high grade order. If omitted, the order is reconstructed from the transition chain. Every non-entry grade in `grade_order` must appear as a `grade_to` in `ratios`. Must not contain duplicates or missing values.

Value

A square numeric matrix with grade dimnames.

Examples

```
ratios <- data.frame(
  grade_from = c("K", "1"),
  grade_to = c("1", "2"),
  ratio = c(0.92, 0.97)
)
progression_matrix(ratios)
```

progression_ratios *Compute grade progression ratios*

Description

Calculates cohort survival / grade progression ratios from historical grade-level enrollment. For each non-entry grade, the ratio is enrollment in that grade divided by enrollment in the grade below one year earlier, summarised across the available year-to-year transitions.

Usage

```
progression_ratios(
  data,
  year = "year",
  grade = "grade",
  enrollment = "enrollment",
  method = c("mean", "geometric", "median", "last", "weighted"),
  n_years = NULL,
  weights = NULL,
  grade_order = NULL
)
```

Arguments

`data`	A long data frame or data-frame subclass of historical enrollment with one row per grade per year. Enrollment may be `NA`, but non-missing values must be finite and non-negative; `NaN` and infinite values are rejected. Year values must be coercible to finite integers and must not be missing.
`year`, `grade`, `enrollment`	Distinct, non-missing character scalars naming columns in `data`. Defaults are `"year"`, `"grade"`, and `"enrollment"`.
`method`	How to summarise per-year ratios into one ratio per grade: `"mean"` (default), `"geometric"`, `"median"`, `"last"` (most recent transition only), or `"weighted"`.
`n_years`	Optional. Use only the most recent `n_years` available adjacent-year transitions. If `n_years` exceeds the number of available transitions, all are used.
`weights`	For `method = "weighted"`, a finite, non-missing, non-negative numeric vector aligned most-recent to oldest, with one weight per transition year used and a positive sum. Do not supply weights for other methods.
`grade_order`	Optional character vector giving the low-to-high grade order. If omitted, factor levels, numeric ordering, or (with a warning) alphabetical ordering is used.

Details

Only transitions between observed consecutive calendar years are used. If the history has one or more calendar-year gaps but still contains an adjacent year pair, the gaps are reported in a warning and are not bridged. Histories with no adjacent year pair are rejected. Gap detection examines the

complete supplied history before `n_years` selects recent adjacent transitions, so an older gap still warns even when it lies outside the selected transitions.

Value

A data frame with columns `grade_from`, `grade_to`, and `ratio`, one row per non-entry grade.

Examples

```
history <- data.frame(
  year = rep(2021:2023, each = 3),
  grade = factor(rep(c("K", "1", "2"), 3), levels = c("K", "1", "2")),
  enrollment = c(100, 90, 80, 110, 95, 88, 120, 99, 91)
)
progression_ratios(history)

# For method = "weighted", weights align most-recent to oldest: here the
# 2022->2023 transition gets weight 2 and 2021->2022 gets weight 1.
progression_ratios(history, method = "weighted", weights = c(2, 1))

# The same K -> 1 ratio via stats::weighted.mean(). Unlike `weights`
# above, weighted.mean() pairs each weight with the value at the same
# position, and the per-year ratios run oldest to newest -- so the
# weights must be reversed to line up.
k_ratios <- c(95 / 100, 99 / 110) # 2021->2022, then 2022->2023
stats::weighted.mean(k_ratios, rev(c(2, 1)))
```

`project_enrollment` *Project enrollment forward*

Description

Projects grade-level enrollment forward an arbitrary horizon using the grade progression ratio method. Internally builds a projection matrix from `ratios` and advances enrollment one year at a time (one matrix-vector product per projected year), overwriting the entry grade with the supplied exogenous value each year. `ratios` is optional when a `schedule` is supplied.

Usage

```
project_enrollment(
  base,
  ratios = NULL,
  horizon = NULL,
  entry = NULL,
  schedule = NULL,
  start_year = NULL
)
```

Arguments

base	Most recent observed enrollment: either a data frame with columns grade and enrollment (optionally year), or a named numeric vector. Grade values or vector names must be present and unique. Enrollment must be finite, non-missing, and non-negative.
ratios	A data frame or data-frame subclass with columns grade_from, grade_to, and ratio, as returned by progression_ratios(). grade_from and grade_to must not be missing. ratio must be numeric, non-negative, and finite; an infinite ratio (from a zero-enrollment feeder) is rejected, while NA/NaN ratios (e.g. from sparse history) are kept in the matrix with a warning.
horizon	Number of years to project (a positive integer).
entry	Exogenous entry-grade enrollment for each projected year: a numeric vector of length horizon, or a data frame with an enrollment or value column. Values must be finite, non-missing, and non-negative. If NULL, the entry grade is held constant at its base value and a warning is issued.
schedule	Optional prebuilt projection schedule: a list of per-year steps, each list(matrix = <square projection as produced by swing_schedule(). When supplied, ratios and entry must be NULL and horizon defaults to the schedule length. Each matrix must be numeric and square; non-missing coefficients must be finite and non-negative. NA/NaN coefficients are preserved and trigger a warning. A missing coefficient can make its output row missing. If that missing enrollment remains after entry replacement, the next matrix multiplication spreads missingness to all grade results because zero times a missing value is still missing. A non-NULL entry value then restores only the entry grade. Matrix row and column names must be unique and identical in the same order; all steps must use the same names. A step's entry must be NULL or one finite, non-negative number.
start_year	Optional integer label for the base year; output years run from start_year + 1. An explicit value and all resulting years must be within the R integer range. If NULL, the year is derived from base$year when present; that column must contain one unambiguous integer within the same range. With no year column, output years are 1..horizon.

Value

A long data frame with columns year, grade, and enrollment, covering the projected years only.

Examples

```
history <- data.frame(
  year = rep(2021:2023, each = 3),
  grade = factor(rep(c("K", "1", "2"), 3), levels = c("K", "1", "2")),
  enrollment = c(100, 90, 80, 110, 95, 88, 120, 99, 91)
)
ratios <- progression_ratios(history)
base <- subset(history, year == 2023, c("grade", "enrollment"))
project_enrollment(base, ratios,
  horizon = 3, entry = c(125, 130, 128),
  start_year = 2023
```

```
)
```

`swing_schedule` *Build a swing/recovery projection schedule*

Description

Assembles a per-year `project_enrollment()` schedule for a school passing through a temporary relocation ("swing"): enrollment is held flat at the depressed observed level during the swing (identity steps), scaled by year-over-year recovery multipliers for the recovery window (diagonal steps), then projected with the grade progression ratio method (the normal projection matrix) for the remaining years.

Usage

```
swing_schedule(
  ratios,
  horizon,
  swing_years,
  recovery,
  entry = NULL,
  grade_order = NULL
)
```

Arguments

`ratios`	A data frame or data-frame subclass with columns `grade_from`, `grade_to`, and `ratio`, as returned by `progression_ratios()`. `grade_from` and `grade_to` must not be missing. `ratio` must be numeric, non-negative, and finite; an infinite ratio (from a zero-enrollment feeder) is rejected, while `NA`/`NaN` ratios (e.g. from sparse history) are kept in the matrix with a warning.
`horizon`	Number of years to project (a positive integer).
`swing_years`	Number of leading years the school is swinging (a non-negative integer); enrollment is held flat at `base`.
`recovery`	Recovery multipliers applied for one year each, immediately after the swing and compounding on the prior year: a numeric vector (whole-school, one multiplier per recovery year) or a grade-by-year numeric matrix (one row per grade). Values must be finite, non-missing, and non-negative. Named matrix rows are matched and reordered by grade; unnamed rows are interpreted in projection grade order. Use `numeric(0)` for no recovery window.
`entry`	Exogenous entry-grade enrollment for the normal grade progression ratio years only: one finite, non-missing, non-negative numeric value for each of the `horizon - swing_years - length(recovery)` years after recovery. Must be empty when there are no normal years.
`grade_order`	Optional character vector giving the low-to-high grade order. If omitted, the order is reconstructed from the transition chain. Every non-entry grade in `grade_order` must appear as a `grade_to` in `ratios`. Must not contain duplicates or missing values.

Value

A list of `horizon` projection steps suitable for the `schedule` argument of `project_enrollment()`.

Examples

```
ratios <- data.frame(
  grade_from = c("K", "1"), grade_to = c("1", "2"), ratio = c(0.92, 0.97)
)
schedule <- swing_schedule(ratios,
  horizon = 6, swing_years = 2,
  recovery = c(1.10, 1.10, 1.05), entry = 130
)
project_enrollment(c(K = 80, `1` = 66, `2` = 60), schedule = schedule)
```

Index