

# Package ‘enrollcast’

October 25, 2026

**Title** Project School Enrollment with Grade Progression Ratios

**Version** 0.2.0

**Description** Projects school enrollment using the cohort survival / grade progression ratio method described in Webster (1970) [doi:10.1080/00220973.1970.11011238](https://doi.org/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://localopen.github.io/enrollcast/>,  
<https://github.com/localopen/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** 8.1.1

**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** 2026-10-24 23:30:02 UTC

Contents

|                              |          |
|------------------------------|----------|
| progression_matrix . . . . . | 2        |
| progression_ratios . . . . . | 3        |
| project_enrollment . . . . . | 4        |
| swing_schedule . . . . .     | 6        |
| <b>Index</b>                 | <b>8</b> |

---

|                    |                                    |
|--------------------|------------------------------------|
| progression_matrix | <i>Build the projection matrix</i> |
|--------------------|------------------------------------|

---

Description

Builds the matrix that advances enrollment by one year. Progression ratios go on the sub-diagonal, so each grade above the entry grade is fed by the grade below it. The entry-grade row is zero because you supply entry enrollment to `project_enrollment()` yourself. The ratios must form a single chain from the lowest grade to the highest: each `grade_to` must be the grade directly above its `grade_from`.

Usage

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

Arguments

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ratios      | A data frame or data-frame subclass with columns <code>grade_from</code> , <code>grade_to</code> , and <code>ratio</code> , as returned by <code>progression_ratios()</code> . <code>grade_from</code> and <code>grade_to</code> must not be missing. <code>ratio</code> must be numeric, non-negative, and finite. An infinite ratio (from a feeder grade with zero enrollment) is an error. NA/NaN ratios, for example from sparse history, stay in the matrix with a warning. |
| grade_order | Optional character vector giving the low-to-high grade order. If omitted, the order is rebuilt from the chain of transitions. Every grade in <code>grade_order</code> except the entry grade must appear as a <code>grade_to</code> in <code>ratios</code> . It 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 | <i>Compute grade progression ratios</i> |
|--------------------|-----------------------------------------|

---

## Description

Calculates cohort survival / grade progression ratios from historical grade-level enrollment. For each grade above the entry grade, the ratio is its enrollment 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

|                                      |                                                                                                                                                                                                                                                                                                  |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>data</code>                    | 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. |
| <code>year, grade, enrollment</code> | Distinct, non-missing character scalars naming columns in data. Defaults are "year", "grade", and "enrollment".                                                                                                                                                                                  |
| <code>method</code>                  | How to summarise per-year ratios into one ratio per grade: "mean" (default), "geometric", "median", "last" (most recent transition only), or "weighted".                                                                                                                                         |
| <code>n_years</code>                 | Optional. Use only the <code>n_years</code> most recent transitions between consecutive years. If there are fewer, all are used.                                                                                                                                                                 |
| <code>weights</code>                 | For <code>method = "weighted"</code> , 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.                                                                   |
| <code>grade_order</code>             | 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 consecutive calendar years are used. If the history has gaps but at least one pair of consecutive years, the gaps are reported in a warning and never bridged. A history with no consecutive pair is an error. The gap check covers the whole history before `n_years` picks the most recent transitions, so an old gap still warns even if it falls 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. 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 | <i>Project enrollment forward</i> |
|--------------------|-----------------------------------|

---

**Description**

Projects grade-level enrollment forward any number of years with the grade progression ratio method. It builds a projection matrix from ratios and advances enrollment one year at a time (one matrix-vector product per year), replacing the entry grade each year with the value you supply. `ratios` is optional when you supply a schedule.

**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 <a href="#">progression_ratios()</a> . grade_from and grade_to must not be missing. ratio must be numeric, non-negative, and finite. An infinite ratio (from a feeder grade with zero enrollment) is an error. NA/NaN ratios, for example from sparse history, stay 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 stays at its base value, with a warning.                                                                                                                          |
| schedule   | Optional prebuilt projection schedule: a list of per-year steps, such as one produced by <a href="#">swing_schedule()</a> . See the Schedules section for what a step contains. When supplied, ratios and entry must be NULL, and horizon defaults to the number of steps.                                                                                                                         |
| start_year | Optional integer label for the base year; output years start at start_year + 1. It and every resulting year must fit in R's integer range. If NULL, the year comes from base\$year when present; that column must hold a single integer value in the same range. Without a year column, output years are 1 to horizon.                                                                             |

**Value**

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

**Schedules**

Each step is `list(matrix = <square projection matrix>, entry = <NULL or a single number>)`, with an optional total. Each matrix must be numeric and square, and its non-missing coefficients must be finite and non-negative. Row and column names must be unique and identical, in the same order, and every step must use the same names. A step's entry must be NULL or one finite, non-negative number.

A step's total rescales that year's grades, entry grade included, after the entry value is applied. It is either one finite, non-negative school total or "base", the sum of the base grades. The rescaled grades are recorded and carried into the next step. Steps without a total aren't rescaled, so the projection continues from the last rescaled grades with the supplied entry values.

Rescaling needs a finite, positive projected total. A missing or infinite projected total is an error, and so is a positive target on a zero projection. If both the projection and the target are zero, the grades are left as they are. A zero target empties that year, so a later target has only the entry classes that arrive afterwards to rescale.

NA/NaN coefficients are kept, with a warning. A missing coefficient can make its grade missing in the output. If that grade is still missing after the entry value is applied, the next matrix multiplication makes every grade missing, because zero times a missing value is still missing. A non-NULL entry value then restores only the 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)
)
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

Builds a per-year `project_enrollment()` schedule for a school that moves to temporary space ("swing") and then recovers. Every year, classes move up by the grade progression ratios and a new entry class arrives. The grades are then rescaled to the base-year school total in swing years, and to the annual totals you supply in recovery years.

## Usage

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

## Arguments

- |             |                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ratios      | A ratios table as accepted by <code>progression_matrix()</code> , a square projection matrix with grade dimnames, or a list of horizon such tables or matrices, one per projected year including swing and recovery years. All matrices must have identical grade dimnames in the same order, matching <code>grade_order</code> if supplied. To open a grade later, include it in every year's matrix and give it a zero row until it opens. |
| horizon     | Number of years to project (a positive integer).                                                                                                                                                                                                                                                                                                                                                                                             |
| swing_years | Number of swing years at the start (a non-negative integer). In these years the school total stays at the total of the base passed to <code>project_enrollment()</code> , while the grade mix progresses normally.                                                                                                                                                                                                                           |

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| recovery    | Annual school target totals for the years right after the swing, one value per year. Values must be finite, non-missing, and non-negative. Use <code>numeric(0)</code> for no recovery years. These are totals, not year-over-year multipliers.                                                                                                                                                                                                                                                                                     |
| entry       | Required exogenous entry-grade enrollment for every projected year, including swing and recovery: 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. In swing and recovery years the entry grade is rescaled with the other grades, so reported entry enrollment is the supplied value times that year's rescaling factor. A larger entry class still gives a larger reported one. In later years the supplied values are used as given. |
| grade_order | Optional character vector giving the low-to-high grade order for every year. If omitted, each ratios table's order is rebuilt from its chain of transitions, and supplied matrices keep their dimname order. Every year must end up with the same grade order.                                                                                                                                                                                                                                                                      |

### Details

The rescaled grades are the students who move up the next year. After the last swing or recovery year, nothing is rescaled: the projection continues from the recovered grades using the progression ratios and the supplied entry values. If the final target is far from the level the ratios and entry values can sustain, later totals drift toward that level. Recovery growth can also inflate a school's own ratios; in that case, pass ratios as a list and use ordinary ratios for the years after recovery.

To update calls written for `enrollcast` 0.1.0, see `vignette("enrollcast")`.

### Value

A list of horizon projection steps suitable for the `schedule` argument of `project_enrollment()`. Each step contains `matrix` and `entry`; swing steps also have `total = "base"` and recovery steps have a numeric `total`.

### 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(225, 245, 260), entry = rep(90, 6)
)
project_enrollment(c(K = 80, `1` = 66, `2` = 60), schedule = schedule)
```

# Index

progression\_matrix, [2](#)  
progression\_matrix(), [6](#)  
progression\_ratios, [3](#)  
progression\_ratios(), [2](#), [5](#)  
project\_enrollment, [4](#)  
project\_enrollment(), [2](#), [6](#), [7](#)  
  
swing\_schedule, [6](#)  
swing\_schedule(), [5](#)