# Package: SlotLim (via r-universe)

May 21, 2026

**Title** Catch Advice for Data-Limited Vertebrate and Invertebrate Fisheries Managed by Harvest Slot Limits

**Version** 0.0.2

**Description** Catch advice for data-limited vertebrate and invertebrate fisheries managed by harvest slot limits using the 'SlotLim' harvest control rule. The package accompanies the manuscript 'SlotLim: catch advice for data-limited vertebrate and invertebrate fisheries managed by harvest slot limits' (Pritchard et al., in prep). Minimum data requirements: at least two consecutive years of catch data, length–frequency distributions, and biomass or abundance indices (all from fishery-dependent sources); species-specific growth rate parameters (either von Bertalanffy, Gompertz, or Schnute); and either the natural mortality rate ('M') or the maximum observed age ('tmax'), from which M is estimated. The following functions have optional plotting capabilities that require 'ggplot2' installed: 'prop_target', 'TBA', 'SAM', 'catch_advice', 'catch_adjust', and 'slotlim_once'.

**License** GPL-3

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Imports** ggplot2, grid, stats, utils, patchwork

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Repository** https://slotlim.r-universe.dev

**Date/Publication** 2025-11-25 18:01:03 UTC

**RemoteUrl** https://github.com/slotlim/slotlim

**RemoteRef** HEAD

**RemoteSha** 14e9ac79490a3b0bbd273e7989555fc686f9f893

# Contents

---

## `catch_adjust` *catch_adjust*

---

### Description

Calculates the targeted proportions under historical (old) and proposed (new) harvest slot limits using the same survivorship-by-length framework as `prop_target`. The ratio ("catch adjustment") is returned, and (optionally) a historical catch value is scaled by the adjustment. The optional plot overlays old/new in-slot proportions on the normalized survivorship curve with arrows indicating the direction of change (old → new).

### Usage

```
catch_adjust(
  old_minLS = NULL,
  old_maxLS = NULL,
  old_Lc = NULL,
  new_minLS = NULL,
  new_maxLS = NULL,
  new_Lc = NULL,
  catch = NULL,
  M = NULL,
  growth_model = c("vb", "gompertz", "schnute"),
  Linf = NULL,
  K = NULL,
  l0 = 0,
  tmax = NULL,
  Gom_Linf = NULL,
  Gom_K = NULL,
  Gom_l0 = NULL,
  g1 = NULL,
  g2 = NULL,
  l2 = NULL,
  Lmin = NULL,
  plot = FALSE,
```

```
  length_units = NULL
)
```

## Arguments

| | |
|---|---|
| `old_minLS`, `old_maxLS`, `old_Lc` | Numeric. Historical slot limits and length at first capture. |
| `new_minLS`, `new_maxLS`, `new_Lc` | Numeric. New slot limits and length at first capture. If `new_Lc` is `NULL`, `old_Lc` is used. |
| `catch` | Optional numeric. Historical catch to be adjusted. If provided, `adjusted_catch = catch * (prop_new/prop_old)` is also computed. |
| `M` | Numeric or `NULL`. Natural mortality. If `NULL`, defaults to `M = 4.899 * tmax^-0.916`. |
| `growth_model` | One of `"vb"`, `"gompertz"`, `"schnute"`. |
| `Linf`, `K`, `l0` | von Bertalanffy parameters; `l0` is start length (default 0). |
| `tmax` | Numeric. The maximum observed age used to bound the integrals via `l(tmax)` and in the default mortality estimator `M = 4.899 * tmax^-0.916`. |
| `Gom_Linf`, `Gom_K`, `Gom_l0` | Gompertz parameters; requires `0 < Gom_l0 < Gom_Linf`. |
| `g1`, `g2`, `l2` | Schnute parameters; `l2` is length at `tmax`; requires `g1>0`, `l2>0`, and this parameterization assumes `g2 != 0`. |
| `Lmin` | Optional numeric. Lower bound for the curve grid. If `NULL` it uses the model's start length (`l0`, `Gom_l0`, or `0`). |
| `plot` | Logical. If `TRUE`, return a `ggplot2` plot. Default `FALSE` (returns numeric catch adjustment only). |
| `length_units` | Optional character scalar. Units to show in the x-axis label when `plot = TRUE` (e.g., "mm" or "cm"). If `NULL` (default), the label is simply "Length". |

## Value

If `plot = FALSE` (default): a numeric scalar `adjust_factor = prop_new/prop_old`. If `plot = TRUE`: a list with

- `prop_old`, `prop_new` — targeted proportions under old/new slots,
- `adjust_factor` — `prop_new/prop_old`,
- `adjusted_catch` — only if `catch` provided,
- `plot` — the `ggplot2` object (or `NULL` if **ggplot2** is unavailable).

## Examples

```
# numeric only
catch_adjust(old_minLS = 130, old_maxLS = 280, old_Lc = 80,
             new_minLS = 100, new_maxLS = 240,
             growth_model = "vb", Linf = 405, K = 0.118, l0 = 0, tmax = 34)
```

```
# with plot (requires ggplot2)
catch_adjust(old_minLS = 130, old_maxLS = 280, old_Lc = 80,
             new_minLS = 100, new_maxLS = 240,
             growth_model = "vb", Linf = 405, K = 0.118, l0 = 0,
             tmax = 34, plot = TRUE, length_units = "mm")
# note that overlapping portions stray from color in legend due to alpha value
catch_adjust(old_minLS = 100, old_maxLS = 150, old_Lc = 80,
             new_minLS = 160, new_maxLS = 300,
             growth_model = "vb", Linf = 405, K = 0.118, l0 = 0,
             tmax = 34, plot = TRUE, length_units = "mm")
```

---

## `catch_advice` *catch_advice*

---

### Description

Calculates the advised catch using the SlotLim framework and (optionally) returns a plot of the percentage change relative to `Cy` across a grid of (`TBA`, `SAM`) values, with the output overlaid.

### Usage

```
catch_advice(
  Cy = NULL,
  TBA = NULL,
  SAM = NULL,
  T1 = NULL,
  T2 = NULL,
  plot = FALSE
)
```

### Arguments

| | |
|---|---|
| `Cy` | Numeric (length 1) > 0. Most recent annual catch, or multi-year average. If landing size restrictions have changed, use `catch_adjust` to adjust the starting catch value accordingly. |
| `TBA` | Numeric (length 1) > 0. Targeted Biomass Adjustment (see `TBA()`). |
| `SAM` | Numeric (length 1) > 0. Size Adherence Multiplier (see `SAM()`). |
| `T1` | Optional numeric (length 1) in (0,1). Maximum allowed proportional *decrease*. If `NULL`, no lower cap. |
| `T2` | Optional numeric (length 1) in (0,1). Maximum allowed proportional *increase*. If `NULL`, no upper cap. |
| `plot` | Logical. If `TRUE`, return a **ggplot2** heatmap (default `FALSE`). |

### Value

- `Ay`: Catch advice (same units as `Cy`).
- `Ay_percent`: Percent change of advice relative to `Cy`.
- `plot`: (only when `plot = TRUE`) a **ggplot2** object visualizing percent change across $TBA \times SAM$.

### See Also

`TBA`, `SAM`

### Examples

```
Cy <- 1000; TBA <- 1.1; SAM <- 0.9
catch_advice(Cy, TBA, SAM)  # compute only


catch_advice(Cy, TBA, SAM, plot = TRUE)
catch_advice(Cy, TBA, SAM, T1 = 0.2, T2 = 0.2, plot = TRUE)
```

---

## `percentile` *percentile*

### Description

Calculates specified percentiles from length-frequency data.

### Usage

```
percentile(
  LF = NULL,
  probs = c(0.025, 0.975),
  na.rm = TRUE,
  sort_probs = TRUE,
  unique_probs = TRUE
)
```

### Arguments

| | |
|---|---|
| `LF` | Numeric vector of length-frequency data (e.g., `data$length`). |
| `probs` | Numeric vector of probabilities in $[0, 1]$ indicating which percentiles to calculate. Default is `c(0.025, 0.975)` as per SlotLim. |
| `na.rm` | Logical; if `TRUE` (default), `NA`s are removed before computing percentiles. If `FALSE`, `NA` values may propagate to the result. |

| | |
|---|---|
| `sort_probs` | Logical; if `TRUE` (default), `probs` are sorted ascending (labels follow the returned order). If `FALSE`, percentiles are returned in the input order. |
| `unique_probs` | Logical; if `TRUE` (default), duplicate `probs` are deduplicated (first occurrence kept for labeling). |

## Details

Uses `stats::quantile(..., type = 7)`, the R default. Labels drop trailing zeros (e.g., `L_5` not `L_5.0`).

## Value

A named list (length = length of `probs`) where each element corresponds to the requested percentile. Names are formatted as `L_x`, where `x` is the percentile value in percent (e.g., `L_2.5`, `L_97.5`).

## Examples

```
length_data <- c(10, 9, 7, 10, 11, 13, NA, 11, 6, 20)
percentile(length_data)  # default 2.5th and 97.5th
percentile(length_data, probs = c(0.05, 0.95)) # 5th and 95th percentiles
```

---

# `prop_target` *prop_target*

## Description

Calculates the proportion of normalized survivorship $S(L)$ falling inside harvest slot limits `[minLS, maxLS]` relative to the exploitable population (>`L_c`), where $S(L) = \exp(-M\, t(L))$ and $t(L)$ is the inverse age-from-length for a chosen growth model.

## Usage

```
prop_target(
  minLS = NULL,
  maxLS = NULL,
  Lc = NULL,
  M = NULL,
  growth_model = c("vb", "gompertz", "schnute"),
  Linf = NULL,
  K = NULL,
  l0 = 0,
  tmax = NULL,
  Gom_Linf = NULL,
  Gom_K = NULL,
  Gom_l0 = NULL,
  g1 = NULL,
  g2 = NULL,
```

```
  l2 = NULL,
  Lmin = NULL,
  plot = FALSE,
  length_units = NULL
)
```

## Arguments

| | |
|---|---|
| `minLS`, `maxLS` | Numeric. Minimum and maximum harvest slot limits (same units as length). |
| `Lc` | Numeric. Lower cutoff; individuals below `Lc` are *not exploitable*. |
| `M` | Numeric or `NULL`. Natural mortality. If `NULL`, defaults to $M = 4.899\, t_{max}^{-0.916}$. |
| `growth_model` | Character. One of `"vb"`, `"gompertz"`, `"schnute"`. |
| `Linf`, `K`, `l0` | VB parameters; `l0` is the start length (default 0). |
| `tmax` | Numeric. Maximum age used to determine $l(t_{max})$ and set the upper integration bound. |
| `Gom_Linf`, `Gom_K`, `Gom_l0` | Gompertz parameters; requires `0` < `Gom_l0` < `Gom_Linf`. |
| `g1`, `g2`, `l2` | Schnute parameters; `l2` = `l(t_{max})`; requires `g1` > `0`, `l2` > `0`, `g2` != `0`. |
| `Lmin` | Optional numeric. Lower bound for the curve grid. If `NULL` it uses the model's start length (`l0`, `Gom_l0`, or `0`). |
| `plot` | Logical. If `TRUE`, return a **ggplot2** visual; default `FALSE`. |
| `length_units` | Optional character scalar. Units to display in the x-axis label when `plot` = `TRUE` (e.g., `"cm"` or `"mm"`). If `NULL` (default), the label is simply `"Length"`. |

## Details

Supported growth models (reparameterized to avoid negative length-at-age-0 and to give exact $t(L_{start}) = 0$):

- **von Bertalanffy (VB)** with start length $l_0$:

$$t(l) = -\frac{1}{K}\ln\left(\frac{L_\infty - l}{L_\infty - l_0}\right), \quad l(t) = L_\infty\big(1 - (1 - l_0/L_\infty)\, e^{-Kt}\big)\,.$$

- **Gompertz** with start length $l_0$ (requires $0 < l_0 < L_\infty$):

$$t(l) = -\frac{1}{K}\ln\left(\frac{\ln(l/L_\infty)}{\ln(l_0/L_\infty)}\right), \quad l(t) = L_\infty\,(l_0/L_\infty)^{e^{-Kt}}.$$

- **Schnute** with $l(0) = 0$ and $l(t_{max}) = l_2$:

$$t(l) = -\frac{1}{g_1}\ln\left(1 - \frac{l^{g_2}}{l_2^{g_2}}\,(1 - e^{-g_1 t_{max}})\right), \quad l(t) = \left(\frac{l_2^{g_2}}{1 - e^{-g_1 t_{max}}}\,(1 - e^{-g_1 t})\right)^{1/g_2}.$$

Survivorship is normalized at the model start so that $S(L_{start}) = 1$: `l0` for vB, `Gom_l0` for Gompertz (requires `0` < `Gom_l0` < `Gom_Linf`), and 0 for Schnute.

Targeted proportion:

$$\frac{\int_{\max(minLS, L_c)}^{\min(maxLS,\, l(t_{max}))} S(L)\, dL}{\int_{\max(L_c,\, L_{start})}^{l(t_{max})} S(L)\, dL}.$$

We clamp only near the upper limit to avoid `log(0)` and never shift the start, preserving $t(L_{start}) = 0$.

### Value

If `plot = FALSE` (default): numeric scalar (the targeted proportion). If `plot = TRUE`: list with `proportion` and `plot` (a ggplot object).

### Examples

```
# Numeric only
prop_target(minLS=120, maxLS=240, Lc=80,
  growth_model="vb", Linf=405, K=0.118, l0=0, tmax=34, plot=FALSE)


# With plot (requires ggplot2)
out <- prop_target(minLS=120, maxLS=240, Lc=80,
 growth_model="schnute", g1=0.2, g2=0.2, l2=405, tmax=34, plot=TRUE, length_units = "mm")
out$plot
```

---

## `rb` *rb*

### Description

Calculates the proportional rate of change in an abundance or biomass index (`rb`) between consecutive data points using one of three methods:

- `"annual"` Change between the two most recent data points: $(x_1 - x_2)/x_2$. Requires at least 2 values.
- `"1over2"` Change between the most recent value and the mean of the two values prior: $(x_1 - \bar{x}_{2:3})/\bar{x}_{2:3}$. Requires at least 3 values.
- `"2over3"` Change between the mean of the two most recent values and the mean of the three values prior: $(\bar{x}_{1:2} - \bar{x}_{3:5})/\bar{x}_{3:5}$. Requires at least 5 values.

### Usage

```
rb(
  b_index = NULL,
  method = c("annual", "1over2", "2over3"),
  na.rm = FALSE,
  digits = NULL
)
```

### Arguments

| | |
|---|---|
| `b_index` | Numeric vector of abundance or biomass indices in descending time order (most recent first). |
| `method` | Character string; one of `"annual"` (default), `"1over2"`, or `"2over3"`. |
| `na.rm` | Logical; if `TRUE`, `NA`s are removed before computing. If `FALSE` (default) and `NA`s are present in the needed positions, the result may be `NA`. |
| `digits` | Optional integer. If supplied, the result is rounded using `round(x, digits)`. If `NULL` (default), full precision is returned. |

### Details

Validates that sufficient data are available for the chosen method and guards against (near-)zero denominators. If a needed denominator is `NA` (after `na.rm`) or numerically zero, an error is thrown.

### Value

A numeric scalar: the proportional rate of change `rb`. Positive values indicate an increase; negative values indicate a decrease.

### Note

`b_index` must be in descending time order (most recent first). Indices should be non-negative (e.g., CPUE).

### See Also

`TBA`

### Examples

```
cpue <- c(0.75, 0.70, 1.49, 1.20, 1.10)  # most recent first
rb(b_index = cpue) # annual method by default
rb(b_index = cpue, method = "1over2")
rb(b_index = cpue, method = "2over3")

cpue2 <- c(0.75, NA, 1.49, 1.20, 1.10)
rb(cpue2, method = "1over2", na.rm = TRUE, digits = 2)
```

---

## `SAM` *SAM*

## Description

Calculate the size adherence multiplier (SAM), which evaluates adherence to harvest slot limits by comparing (`lower`, `upper`) percentiles of length–frequency data to `minLS` and `maxLS`. Optionally, produce a graph showing how SAM varies across a grid of (`lower`, `upper`) values.

When `lower` >= `minLS` and `upper` <= `maxLS`, neither of the slot limits are violated and the multiplier is calculated without constraint. When `lower` < `minLS` or `upper` > `maxLS`, at least one slot limit is violated and the multiplier is capped at `constraint` (default = 1).

## Usage

```
SAM(
  lower = NULL,
  upper = NULL,
  minLS = NULL,
  maxLS = NULL,
  constraint = 1,
  digits = 2,
  plot = FALSE,
  res = 1,
  lower_percentile = 2.5,
  upper_percentile = 97.5,
  length_units = NULL
)
```

## Arguments

| | |
|---|---|
| `lower` | Numeric (length 1). Lower percentile of catch length (e.g., 2.5th). |
| `upper` | Numeric (length 1). Upper percentile of catch length (e.g., 97.5th). |
| `minLS` | Numeric (length 1). Minimum landing size (must be > 0). |
| `maxLS` | Numeric (length 1). Maximum landing size (must be > 0). |
| `constraint` | Numeric (length 1) in $[0, 1]$. Cap applied *when either* slot limit is violated (default = 1). |
| `digits` | Integer. Number of decimal places used to round outputs (default = 2). Set `digits` = `NA` to prevent rounding. |
| `plot` | Logical. If `TRUE`, include a **ggplot2** plot of the calculated value on a grid of (`lower`, `upper`) combinations (default `FALSE`). |
| `res` | Numeric > 0. Grid step for plotting when `plot` = `TRUE`. Smaller values increase smoothness but can be slower (default `1`). |
| `lower_percentile`, `upper_percentile` | Numbers used only for axis labels when `plot` = `TRUE` (defaults `2.5` and `97.5`). |
| `length_units` | Optional character scalar. Units to display in the x/y-axis labels when `plot` = `TRUE` (e.g., `"cm"` or `"mm"`). If `NULL` (default), units are omitted. |

## Details

The unconstrained multiplier is $(1 + lower\_adherence) \times (1 + upper\_adherence)$. If any slot limit is violated, the multiplier is `pmin(constraint, multiplier)`.

### Value

A list with:

**lower_adherence** Relative deviation of `lower` from `minLS`: $(lower - minLS)/minLS$.

**upper_adherence** Relative deviation of `upper` from `maxLS`: $(maxLS - upper)/maxLS$.

**SAM** Size adherence multiplier. `SAM > 1` increases the advised catch; `SAM < 1` decreases it.

**plot** (only when `plot = TRUE`) a **ggplot2** object visualizing SAM over a grid. Illogical combinations of percentiles are shaded grey (e.g., $L_{2.5} > L_{97.5}$).

### See Also

`percentile` for computing percentiles from length–frequency data.

### Examples

```
SAM(lower = 13, upper = 24, minLS = 12, maxLS = 24)                      # no violation
SAM(lower = 13, upper = 25, minLS = 12, maxLS = 24, constraint = 0.95)  # violation with constraint


out <- SAM(
  lower = 13, upper = 25,
  minLS = 12, maxLS = 24,
  res = 0.5,
  lower_percentile = 5, upper_percentile = 95,
  constraint = 1,
  plot = TRUE,
  length_units = ”cm”)
out$SAM
```

---

## `slotlim_once` *slotlim_once*

---

### Description

Run a single SlotLim pass: compute `rb`, `P`, `TBA`, `SAM`, and catch advice `Ay_percent`; optionally show a composite plot (`P`, `TBA`, `SAM`, `Ay_percent`).

### Usage

```
slotlim_once(
  Cy = NULL,
  b_index = NULL,
  method = c(”annual”, ”1over2”, ”2over3”),
  minLS = NULL,
  maxLS = NULL,
  Lc = NULL,
```

```
  growth_model = c("vb", "gompertz", "schnute"),
  Linf = NULL,
  K = NULL,
  l0 = 0,
  tmax = NULL,
  Gom_Linf = NULL,
  Gom_K = NULL,
  Gom_l0 = NULL,
  g1 = NULL,
  g2 = NULL,
  l2 = NULL,
  M = NULL,
  lower = NULL,
  upper = NULL,
  LF = NULL,
  probs = c(0.025, 0.975),
  constraint = 1,
  T1 = NULL,
  T2 = NULL,
  plots = FALSE,
  length_units = NULL
)
```

## Arguments

| | |
|---|---|
| `Cy` | Numeric. Historical catch. |
| `b_index` | Numeric vector of a biomass or abundance index in descending time order (most recent first). |
| `method` | Character. Method for calculating `rb` ("annual", "1over2", or "2over3"). |
| `minLS`, `maxLS`, `Lc` | Numeric. Slot limits and length at first capture. |
| `growth_model` | One of `"vb"`, `"gompertz"`, `"schnute"`. |
| `Linf`, `K`, `l0` | von Bertalanffy (vB) parameters; `l0` is the start length (default 0). |
| `tmax` | Numeric. Maximum observed age; used for integration bounds and (if `M` is `NULL`) to compute default `M`. |
| `Gom_Linf`, `Gom_K`, `Gom_l0` | Gompertz parameters; requires `0` < `Gom_l0` < `Gom_Linf`. |
| `g1`, `g2`, `l2` | Schnute parameters; `l2` is length at `tmax`; requires `g1` > `0`, `l2` > `0`, and this parameterization assumes `g2` != `0`. |
| `M` | Numeric or `NULL`. Natural mortality. If `NULL`, defaults to $M = 4.899 \times tmax^{-0.916}$. |
| `lower`, `upper` | Optional values at specified percentiles. If provided, used directly by `SAM()`. |
| `LF` | Optional numeric vector of length-frequency data. If `lower`/`upper` are `NULL` and `LF` is supplied, the function computes percentiles via `percentile(LF, probs)` and uses them. |
| `probs` | Numeric vector of probabilities in $[0, 1]$ passed to `percentile()` when `LF` is used. Default `c(0.025, 0.975)`. |

| | |
|---|---|
| `constraint` | Numeric (default 1). Passed to `SAM()`. |
| `T1, T2` | Optional numerics passed to `catch_advice()`. |
| `plots` | Logical; if `TRUE`, a $2 \times 2$ composite plot is printed (if **patchwork** is available). |
| `length_units` | Optional character; x-axis units for the `prop_target` and `SAM` plots (e.g., "mm"). |

## Details

Precedence for size inputs: if both `lower` and `upper` are provided, they are used. Otherwise, if `LF` is provided, they are derived via `percentile(LF, probs)`. Else error.

## Value

A list with `Ay`, `Ay_percent`, `TBA`, `SAM`, `rb`, `P`, and (if `plots=TRUE`) a composite plot. Also returns the resolved `M` and the `lower`/`upper` bounds actually used; `tmax` is echoed back.

## Examples

```
# Minimal, fast example (no plotting), passing lower/upper directly:
slotlim_once(
  Cy = 1000,
  b_index = c(0.5, 0.6, 0.7, 0.6, 0.5), method = "2over3",
  minLS = 120, maxLS = 240, Lc = 80,
  growth_model = "vb", Linf = 405, K = 0.118, l0 = 0,
  tmax = 34,
  lower = 100, upper = 220
)


# Derive lower/upper from length-frequency percentiles:
set.seed(1)
LF <- rnorm(200, mean = 180, sd = 40)  # toy example LF

# Compute M from tmax:
slotlim_once(
  Cy = 1000,
  b_index = c(0.5, 0.6, 0.7, 0.6, 0.5),
  minLS = 120, maxLS = 240, Lc = 80,
  growth_model = "vb", Linf = 405, K = 0.118, l0 = 0,
  tmax = 34,
  LF = LF, probs = c(0.05, 0.95),
  method = "1over2"  # rb method chosen
)

# Use explicit M (still provide tmax for bounds):
slotlim_once(
  Cy = 1000,
  b_index = c(0.5, 0.6, 0.7, 0.6, 0.5),
  minLS = 120, maxLS = 240, Lc = 80,
  growth_model = "vb", Linf = 405, K = 0.118, l0 = 0,
  tmax = 34,
  M = 0.19,
```

```
  LF = LF, probs = c(0.025, 0.975),
  method = "1over2"  # rb method chosen
)

# Plotting example (needs ggplot2 and patchwork):
slotlim_once(
  Cy = 1000,
  b_index = c(0.5, 0.6, 0.7, 0.6, 0.5),
  minLS = 120, maxLS = 240, Lc = 80,
  growth_model = "vb", Linf = 405, K = 0.118, l0 = 0,
  tmax = 34,
  LF = LF, probs = c(0.025, 0.975),
  method = "1over2",  # rb method chosen
  plots = TRUE, length_units = "mm"
)
```

---

## `TBA` *TBA*

### Description

Calculate the targeted biomass adjustment (TBA), which dampens the influence of proportional rate of change $rb$ on catch advice when the proportion of abundance targeted by harvest slot limits is small. Optionally, produce a graph showing how TBA varies across a grid of (`P_targeted`, `rb`) values.

The TBA is calculated as $1 + (P_{\text{targeted}} \times rb)$.

### Usage

```
TBA(P_targeted = NULL, rb = NULL, digits = 2, plot = FALSE)
```

### Arguments

| | |
|---|---|
| `P_targeted` | Numeric (length 1) in [0, 1]. Proportion of abundance targeted by harvest slot limits (e.g., from `prop_target()`). |
| `rb` | Numeric (length 1). Proportional rate of change in a biomass index (e.g., from `rb()`). Typical values lie in [-1, 1], but larger magnitudes are allowed. |
| `digits` | Integer. Number of decimal places used to round outputs (default = 2). Set `digits` = `NA` to prevent rounding. |
| `plot` | Logical. If `TRUE`, include a **ggplot2** plot of the calculated value on a grid of (`P_targeted`, `rb`) combinations (default `FALSE`). |

### Details

The adjustment dampens large changes in biomass indices when the targeted proportion is small by multiplying `rb` by `P_targeted`. The plot shades the surface of $1 + P \times rb$. A dashed horizontal line marks `rb` = `0`.

## Value

A list with:

**P_targeted** Input targeted proportion (numeric scalar).

**rb** Input proportional rate of change (numeric scalar).

**damped_change** $P_{\mathrm{targeted}} \times rb$ (numeric scalar).

**TBA** Targeted biomass adjustment multiplier $1 + P_{\mathrm{targeted}} \times rb$ (numeric scalar). `TBA > 1` increases advised catch; `TBA < 1` decreases it.

**plot** (only when `plot=TRUE`) a **ggplot2** object visualizing TBA over a grid.

## See Also

`prop_target` for targeted proportion; `rb` for proportional rate of change.

## Examples

```
TBA(P_targeted = 0.5, rb = -0.5)               # compute only

# compute + plot (requires ggplot2)
out <- TBA(P_targeted = 0.5, rb = -0.5, digits = 2, plot = TRUE)
out$plot
```

# Index