

Package: foretell (via r-universe)

October 17, 2026

Type Package

Title Projecting Customer Retention Based on Fader and Hardie
Probability Models

Version 0.3

Author Srihari Jaganathan

Maintainer Srihari Jaganathan <sriharitn@gmail.com>

Description Project Customer Retention based on Beta Geometric, Beta
Discrete Weibull and Latent Class Discrete Weibull Models. This
package is based on Fader and Hardie (2007)
<[doi:10.1002/dir.20074](https://doi.org/10.1002/dir.20074)> and Fader and Hardie et al. (2018)
<[doi:10.1016/j.intmar.2018.01.002](https://doi.org/10.1016/j.intmar.2018.01.002)>.

Depends R (>= 3.0.1)

License GPL-3

Encoding UTF-8

LazyData true

Imports stats,nloptr,pracma,Matrix,MASS,numDeriv

RoxygenNote 7.3.2

Config/pak/sysreqs cmake

Repository <https://sriharitn.r-universe.dev>

Date/Publication 2025-09-04 00:24:39 UTC

RemoteUrl <https://github.com/sriharitn/foretell>

RemoteRef HEAD

RemoteSha ac8f491a574d7bb9939eb20629df151318c8ee85

Contents

BdW	2
bdw1c	3
BetadiscreteWeibull	6
BG	7

customer_retention	8
extltrend	9
geomlc	9
LCW	11
persistence_data	13
sbglc	14
shiftedBetaGeometric	17
sim.bdwlc	18
sim.geomlc	20
sim.sbglc	21
Index	23

BdW	<i>Beta discrete Weibull (BdW) Model for Projecting Customer Retention.</i>
-----	---

Description

BdW is a beta discrete weibull model implemented based on Fader and Hardie probability based projection methodology. The survivor function for BdW is

$$Beta(a, b + t^c)/Beta(a, b)$$

Usage

```
BdW(  
  surv_value,  
  h,  
  lower = c(0.001, 0.001, 0.001),  
  upper = c(10000, 10000, 10000),  
  subjects = 1000  
)
```

Arguments

surv_value	a numeric vector of historical customer retention percentage should start at 100 and non-starting values should be between 0 and less than 100
h	forecasting horizon
lower	lower limit used in R optim routine. Default is c(1e-3, 1e-3).
upper	upper limit used in R optim routine. Default is c(10000, 10000, 10000).
subjects	Total number of customers or subject default 1000

Value

fitted: Fitted values based on historical data

projected: Projected h values based on historical data

max.likelihood: Maximum Likelihood of Beta discrete Weibull

params - a, b and c: Returns a and b paramters from maximum likelihood estimation for beta distribution and c

References

Fader P, Hardie B. How to project customer retention. *Journal of Interactive Marketing*. 2007;21(1):76-90.

Fader P, Hardie B, Liu Y, Davin J, Steenburgh T. "How to Project Customer Retention" Revisited: The Role of Duration Dependence. *Journal of Interactive Marketing*. 2018;43:1-16.

Examples

```
surv_value <- c(100,86.9,74.3,65.3,59.3)
h <- 6
BdW(surv_value,h)
```

bdw1c

*Beta-Discrete-Weibull (BDW) survival: One-and-Done and Cure***Description**

Fits the Beta-Discrete-Weibull (BDW) survival model to a monotonically nonincreasing survival series using unconstrained maximum likelihood estimation. Internal reparameterization ensures $a > 0$, $b > 0$, and $c > 0$ (e.g., via $\log c$), with optional one-and-done mass d and cure fraction π . When $c = 1$, the BDW reduces to the shifted Beta-Geometric (sBG).

Usage

```
bdw1c(
  surv_value,
  h,
  N0 = NULL,
  one_and_done = FALSE,
  cure = FALSE,
  starts_m = 0.6,
  starts_q = 0.1,
  starts_c = 1,
  starts_d = 0.05,
  starts_cure = 0.05,
```

```

compute_se = TRUE,
input = c("auto", "percent", "count"),
percent_tol = 1e-08,
boundary_tol = 0.001
)

```

Arguments

surv_value	Numeric survival series (percent starting at 100, or counts).
h	Integer forecast horizon (≥ 0).
N0	Cohort size if 'input = "percent"'. If 'input = "count"', infer from 'surv_value' and 'N0'.
one_and_done	Logical; include one-and-done parameter d .
cure	Logical; include cure fraction parameter (fraction never churning).
starts_m, starts_q	Starting values for m, q on the original scale in $(0, 1)$.
starts_c	Starting values for c on the original scale.
starts_d, starts_cure	Starting values for d and π on the original scale in $[0, 1]$.
compute_se	Logical; compute robust SEs via Hessian + delta method.
input	One of "auto", "percent", "count". If "auto", infer from 'surv_value' and 'N0'.
percent_tol, boundary_tol	Tolerances for monotonicity checks and boundary SE skipping.

Details

The baseline BDW survival (no one-and-done, no cure) at discrete times $t = 0, 1, \dots$ is

$$S(t) = \frac{B(a, b + t^c)}{B(a, b)}.$$

With one-and-done mass d and cure fraction π , survival for $t \geq 1$ is

$$S(t) = (1 - d) \left[\pi + (1 - \pi) \frac{B(a, b + t^c)}{B(a, b)} \right], \quad S(0) = 1.$$

Parameters (a, b, c) (and optionally d, π) are estimated by maximizing the likelihood implied by the observed survival series using an internal unconstrained parameterization. If `compute_se = TRUE`, standard errors on the original scale are obtained via the delta method.

Beta reparameterization by mean and polarization index: The model uses a numerically stable and interpretable reparameterization of the Beta distribution $Beta(a, b)$ in terms of a **mean** $m \in (0, 1)$ and a **polarization (concentration) index** $q \in (0, 1)$.

Standard Beta density:

$$f(x \mid a, b) = \frac{x^{a-1}(1-x)^{b-1}}{B(a, b)}, \quad x \in (0, 1), \quad a, b > 0.$$

Mapping to mean and polarization:

$$m = \frac{a}{a+b}, \quad q = \frac{1}{1+a+b}.$$

Here, m is the expected probability, and q summarizes concentration: small q means large $a+b$ (high concentration); large q means small $a+b$ (diffuse).

Inverse mapping (used internally for estimation):

$$a = m \left(\frac{1}{q} - 1 \right), \quad b = (1-m) \left(\frac{1}{q} - 1 \right).$$

This transformation is bijective for $m \in (0, 1)$, $q \in (0, 1)$, and guarantees $a > 0, b > 0$.

Starting values: The arguments `starts_m` and `starts_q` provide starting values for m and q , respectively. They are converted to (a, b) via the inverse mapping above:

$$a_{\text{start}} = \text{starts_m} \left(\frac{1}{\text{starts_q}} - 1 \right), \quad b_{\text{start}} = (1 - \text{starts_m}) \left(\frac{1}{\text{starts_q}} - 1 \right).$$

This parameterization typically improves optimization stability and makes starting values more interpretable.

Value

An object of class `bdw_fit` with the elements shown below:

`y` Numeric vector of observed survival values on the input scale.

`input` Character scalar, one of "input", "percent", "count", or "prob"; echoes how `y` was interpreted.

`N0` Scalar. Reference cohort size used for count/percent scaling.

`flags` Named logical vector with model switches, e.g. `c(one_and_done = TRUE/FALSE, cure = TRUE/FALSE)`.

`coef_params` Named numeric vector of parameters on the natural scale. For sBG: $c(a, b)$ and optionally `d` (one-and-done) and `cure`. For BDW: $c(a, b, c)$ and optionally `d`, `cure`.

`coef_repar` Named numeric vector with the mean-polarization reparameterization, typically $c(m, q)$ (and `c` for BDW if modeled on the log/exp scale).

`logits` Named numeric vector of unconstrained optimization variables (e.g., `theta_m`, `theta_q`, `theta_d`, `theta_cure`, `theta_c`).

`fitted` Numeric vector $S(0:T_{\text{obs}} - 1)$ on the probability scale; the in-sample fit.

`projected` Numeric vector $S(0:T_{\text{last}})$ (fit + horizon) on the probability scale.

`logLik` Maximized log-likelihood.

`convergence` Optimizer return code (0 indicates successful convergence).

`optim_out` Raw optimizer list (as returned by `optim()`); useful for debugging.

`vcov_theta` Variance-covariance matrix on the unconstrained scale (logits).

`vcov_params` Variance-covariance matrix for `coef_params` (delta-method mapped from `vcov_theta`).

`vcov_repar` Variance-covariance matrix for `coef_repar` (e.g., m, q).

`se_params` Named vector of standard errors for `coef_params`.

`se_repar` Named vector of standard errors for `coef_repar`.

`se_note` Character note if SEs are approximate/unstable (e.g., near-PD fix).

References

- Fader P, Hardie B. How to project customer retention. *Journal of Interactive Marketing*. 2007;21(1):76-90.
- Fader P, Hardie B, Liu Y, Davin J, Steenburgh T. "How to Project Customer Retention" Revisited: The Role of Duration Dependence. *Journal of Interactive Marketing*. 2018;43:1-16.

Examples

```
## Not run:
N0 <- 500; S <- c(100, 86.9, 74.3, 65.3, 59.3, 55.1, 51.7, 49.1)
fit <- sbg1c(S, h=6, N0=N0, input="percent")
summary(fit)
plot(fit, scale="percent")

## End(Not run)
```

BetadiscreteWeibull *Beta discrete Weibull (bdw) Distribution Family Function*

Description

Provides functions for the probability mass function (PMF), cumulative distribution function (CDF), quantile function, and random variate generation for the BdW distribution.

Usage

```
dbdw(x, shape1, shape2, shape3, log = FALSE)
pbdw(x, shape1, shape2, shape3, lower.tail = TRUE, log.p = FALSE)
qbdw(p, shape1, shape2, shape3, lower.tail = TRUE, log.p = FALSE)
rbdw(n, shape1, shape2, shape3)
```

Arguments

x	Vector of non-negative integers for dbdw and pbdw.
p	Vector of probabilities ($0 \leq p \leq 1$) for qbdw.
n	Number of random variates to generate for rbdw.
shape1	First shape parameter "a" (must be > 0).
shape2	Second shape parameter "b" (must be > 0).
shape3	Second shape parameter "c" (must be > 0).
log	Logical; if TRUE, probabilities are returned on the log scale (for dsbg and psbg).
lower.tail	Logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise $P(X > x)$ (for psbg).
log.p	Logical; if TRUE, probabilities are returned on the log scale (for psbg and qsbg).

Value

- dbdw: A numeric vector of PMF values.
- pbdw: A numeric vector of CDF values.
- qbdw: A numeric vector of quantile values.
- rbdw: A numeric vector of random variates.

References

Fader P, Hardie B. How to project customer retention. *Journal of Interactive Marketing*. 2007;21(1):76-90.

Fader P, Hardie B, Liu Y, Davin J, Steenburgh T. "How to Project Customer Retention" Revisited: The Role of Duration Dependence. *Journal of Interactive Marketing*. 2018;43:1-16.

Examples

```
# PMF example
dbdw(1:5, shape1 = 2, shape2 = 3, shape3 = 0.5)

# CDF example
pbdw(1:5, shape1 = 2, shape2 = 3, shape3 = 0.5)

# Quantile example
qbdw(c(0.1, 0.5, 0.9), shape1 = 2, shape2 = 3, shape3 = 0.5)

# Random variates
rbdw(10, shape1 = 2, shape2 = 3, shape3 = 0.5)
```

 BG

Beta Geometric (BG) Model for Projecting Customer Retention.

Description

BG is a beta geometric model implemented based on Fader and Hardie probability based projection methodology. The survivor function for BG is

$$Beta(a, b + t) / Beta(a, b)$$

Usage

```
BG(surv_value, h, lower = c(0.001, 0.001), subjects = 1000)
```

Arguments

surv_value	a numeric vector of historical customer retention percentage should start at 100 and non-starting values should be between 0 and less than 100
h	forecasting horizon
lower	lower limit used in R optim routine. Default is c(1e-3, 1e-3).
subjects	Total number of customers or subject default 1000

Value

- fitted: Fitted values based on historical data
- projected: Projected h values based on historical data
- max.likelihood: Maximum Likelihood of Beta Geometric
- params - a, b: Returns a and b paramters from maximum likelihood estimation for beta distribution

References

Fader P, Hardie B. How to project customer retention. Journal of Interactive Marketing. 2007;21(1):76-90.

Examples

```
surv_value <- c(100,86.9,74.3,65.3,59.3)
h <- 6
BG(surv_value,h)
```

customer_retention	<i>Observed % Customers Surviving at Least 0-12 Years</i>
--------------------	---

Description

A dataset containing customer retention.

Usage

```
data(customer_retention)
```

Format

A data frame 13 observations and 3 variables.

Details

- year** Time in years
- regular** % of regular customers surviving
- high_end** % of high_end customers surviving

References

Fader P, Hardie B. How to project customer retention. Journal of Interactive Marketing. 2007;21(1):76-90.

exltrend	<i>Excel based trendlines for projecting customer retention.</i>
----------	--

Description

exltrend generates Microsoft(r) Excel(r) based linear, logarithmic, exponential, polynomial of order 2, power trends.

Usage

```
exltrend(surv_value, h)
```

Arguments

surv_value	a numeric vector of historical customer retention percentage should start at 100 and non-starting values should be between 0 and less than 100
h	forecasting horizon

Value

fitted:	A data frame of fitted Values based on historical data for linear (lin.p), exponential (exp.p), logarithmic (log.p), polynomial (poly.p) of order 2 and power (pow.p) trends.
projected:	A data frame of projected h values based on historical data for linear (lin.p), exponential (exp.p), logarithmic (log.p), polynomial (poly.p) of order 2 and power (pow.p) trends.

Examples

```
surv_value <- c(100,86.9,74.3,65.3,59.3)
h <- 6
exltrend(surv_value,h)
```

geom1c	<i>Geometric survival: One and Done and Cure Models</i>
--------	---

Description

Fit a geometric survival model (optionally with one-and-done and cure), using logit parameterization, robust SEs from the Hessian, and S3 methods for printing, summarizing, predicting, plotting.

Usage

```
geom1c(
  surv_value,
  h,
  N0 = NULL,
  one_and_done = FALSE,
  cure = FALSE,
  starts_p = 0.15,
  starts_d = 0.05,
  starts_cure = 0.05,
  compute_se = TRUE,
  input = c("auto", "percent", "count"),
  percent_tol = 1e-08,
  boundary_tol = 0.001
)
```

Arguments

<code>surv_value</code>	Numeric survival series (percent starting at 100, or counts).
<code>h</code>	Integer forecast horizon (≥ 0).
<code>N0</code>	Cohort size if <code>input="percent"</code> .
<code>one_and_done</code>	Logical; include one-and-done parameter d .
<code>cure</code>	Logical; include cure fraction parameter.
<code>starts_p</code>	Starting values for p on the original scale in $(0, 1)$.
<code>starts_d, starts_cure</code>	Starting values for d and π on the original scale in $[0, 1]$.
<code>compute_se</code>	Logical; compute robust SEs via Hessian + delta method.
<code>input</code>	One of <code>"auto"</code> , <code>"percent"</code> , <code>"count"</code> .
<code>percent_tol, boundary_tol</code>	Tolerances for monotonicity and boundary SE skipping.

Details

Let $p \in (0, 1)$ be the per-period churn probability in a geometric model. The baseline geometric survival (no one-and-done, no cure) at discrete times $t = 0, 1, \dots$ is

$$S(t) = (1 - p)^t, \quad S(0) = 1.$$

With one-and-done mass d and cure fraction π , survival for $t \geq 1$ is

$$S(t) = (1 - d) [\pi + (1 - \pi)(1 - p)^{t-1}], \quad S(0) = 1.$$

Parameters are estimated by maximizing the likelihood implied by the observed survival series using an internal unconstrained parameterization for numerical stability (e.g., a logit transform for p, d, π). If `compute_se = TRUE`, standard errors on the original scale are obtained via the delta method from the Hessian on the working scale.

Value

An object of class `geom_fit` with the elements shown below:

`y` Numeric vector of observed survival values on the input scale.

`input` Character scalar, one of "input", "percent", "count", or "prob"; echoes how `y` was interpreted.

`N0` Scalar. Reference cohort size used for count/percent scaling.

`flags` Named logical vector with model switches, e.g. `c(one_and_done = TRUE/FALSE, cure = TRUE/FALSE)`.

`coef_params` Named numeric vector of parameters on the natural scale. For Geom: `p` and optionally `d` (one-and-done) and `cure`.

`fitted` Numeric vector $S(0:T_{\text{obs}} - 1)$ on the probability scale; the in-sample fit.

`projected` Numeric vector $S(0:T_{\text{last}})$ (fit + horizon) on the probability scale.

`logLik` Maximized log-likelihood.

`convergence` Optimizer return code (0 indicates successful convergence).

`optim_out` Raw optimizer list (as returned by `optim()`); useful for debugging.

`vcov_theta` Variance-covariance matrix on the unconstrained scale (logits).

`vcov_params` Variance-covariance matrix for `coef_params` (delta-method mapped from `vcov_theta`).

`vcov_repar` Variance-covariance matrix for `coef_repar` (e.g., `p`).

`se_params` Named vector of standard errors for `coef_params`.

`se_repar` Named vector of standard errors for `coef_repar`.

`se_note` Character note if SEs are approximate/unstable (e.g., near-PD fix).

Examples

```
## Not run:
N0 <- 500; S <- c(100,94,90,87,85,84,83,82,81,80,79)
fit <- geom1c(S, h=6, N0=N0, input="percent")
summary(fit)
plot(fit, scale="percent")

## End(Not run)
```

Description

LCW is a latent class weibull model implementation based on Fader and Hardie probability based projection methodology. The survivor function for LCW is

$$wS(t|t1, c1) + (1 - w)S(t|t2, c2), 0 < w < 1$$

Usage

```
LCW(
  surv_value,
  h,
  lower = c(0.001, 0.001, 0.001, 0.001, 0.001),
  upper = c(0.99999, 10000, 0.99999, 10000, 0.99999),
  subjects = 1000
)
```

Arguments

<code>surv_value</code>	a numeric vector of historical customer retention percentage should start at 100 and non-starting values should be between 0 and less than 100
<code>h</code>	forecasting horizon
<code>lower</code>	lower limit used in R opt im rotuine. Default is <code>c(0.001, 0.001, 0.001, 0.001, 0.001)</code> .
<code>upper</code>	upper limit used in R opt im rotuine. Default is <code>c(0.99999, 10000, 0.99999, 10000, 0.99999)</code> .
<code>subjects</code>	Total number of customers or subject default 1000

Value

<code>fitted:</code>	Fitted Values based on historical data
<code>projected:</code>	Projected h values based on historical data
<code>max.likelihood:</code>	Maximum Likelihood of LCW
<code>params - t1, t2, c1, c2, w:</code>	Returns t1,c1,t2,c2,w paramters from maximum likelihood estimation

References

Fader P, Hardie B. How to project customer retention. *Journal of Interactive Marketing*. 2007;21(1):76-90.

Fader P, Hardie B, Liu Y, Davin J, Steenburgh T. "How to Project Customer Retention" Revisited: The Role of Duration Dependence. *Journal of Interactive Marketing*. 2018;43:1-16.

Examples

```
surv_value <- c(100,86.9,74.3,65.3,59.3,55.1,51.7,49.1,46.8,44.5,42.7,40.9,39.4)
h <- 6
LCW(surv_value,h)
```

persistence_data *Drug persistence (retention) rates by different therapeutic class.*

Description

A dataset containing drug persistence of patients in different therapeutic classes.

Usage

```
data(persistence_data)
```

Format

A data frame 334 observations and 3 variables:

therapy Type of therapy. Unique values include: "Hypertension" "Ocular Hypertension" "Statin" "Insulin" "Epilepsy" "RA" "Osteoporosis" "Alzheimer" "ADHD" "Atrial Fibrillation". See references below. Data was extracted using <https://automeris.io/WebPlotDigitizer/> and discretized using akima package.

time_period Time Period

value % Patients retained

A data frame with 334 rows and 3 variables

References

Hypertension: Solomon M, Goldman D, Joyce G, Escarce J. Cost Sharing and the Initiation of Drug Therapy for the Chronically Ill. *Archives of Internal Medicine*. 2009;169(8):740-748.

Ocular Hypertension: Campbell J, Schwartz G, LaBounty B, Kowalski J, Patel. Patient adherence and persistence with topical ocular hypotensive therapy in real-world practice: a comparison of bimatoprost 0.01% and travoprost Z 0.004% ophthalmic solutions. *Clinical Ophthalmology*. 2014;8:927-935.

Statin: Kiss Z, Nagy L, Reiber I, Paragh G, Molnar M, Rokszin G et al. Persistence with statin therapy in Hungary. *Archives of Medical Science*. 2013;9(3):409-417.

Insulin: Roussel R, Charbonnel B, Behar M, Gourmelen J, Emery C, Detournay B. Persistence with Insulin Therapy in Patients with Type 2 Diabetes in France: An Insurance Claims Study. *Diabetes Therapy*. 2016;7(3):537-549.

Epilepsy: Lai E, Hsieh C, Su C, Yang Y, Huang C, Lin S et al. Comparative persistence of antiepileptic drugs in patients with epilepsy: A STROBE-compliant retrospective cohort study. *Medicine*. 2016;95(35):e4481.

RA: Neovius M, Arkema E, Olsson H, Eriksson J, Kristensen L, Simard J et al. Drug survival on TNF inhibitors in patients with rheumatoid arthritis comparison of adalimumab, etanercept and infliximab. *Annals of the Rheumatic Diseases*. 2013;74(2):354-360.

Osteoporosis: Kishimoto H, Maehara M. Compliance and persistence with daily, weekly, and monthly bisphosphonates for osteoporosis in Japan: analysis of data from the CISA. *Archives of Osteoporosis*. 2015;10(27):1-6.

Alzheimer: Suh D, Thomas S, Valiyeva E, Arcona S, Vo L. Drug persistency of two cholinesterase inhibitors: rivastigmine versus donepezil in elderly patients with Alzheimer’s disease. *Drugs & Aging*. 2005;22(8):695-707.

ADHD: Beau-Lejdstrom R, Douglas I, Evans S, Smeeth L. Latest trends in ADHD drug prescribing patterns in children in the UK: prevalence, incidence and persistence. *BMJ Open*. 2016;6(6):1-8.

Atrial Fibrillation: Gomes T, Mamdani M, Holbrook A, Paterson J, Juurlink D. Persistence With Therapy Among Patients Treated With Warfarin for Atrial Fibrillation. *Archives of Internal Medicine*. 2012;172(21):1687-1689.

sbglc	<i>Shifted Beta Geometric (sbg) survival: One and Done and Cure Models</i>
-------	--

Description

Fits the shifted Beta-Geometric (sBG) survival model to a monotonically nonincreasing survival series using unconstrained optimization (internal reparameterization ensures $a>0, b>0$); optional one-and-done mass d and cure fraction).

Usage

```
sbglc(  
  surv_value,  
  h,  
  N0 = NULL,  
  one_and_done = FALSE,  
  cure = FALSE,  
  starts_m = 0.6,  
  starts_q = 0.1,  
  starts_d = 0.05,  
  starts_cure = 0.05,  
  compute_se = TRUE,  
  input = c("auto", "percent", "count"),  
  percent_tol = 1e-08,  
  boundary_tol = 0.001  
)
```

Arguments

- | | |
|--------------------|---|
| surv_value | Numeric survival series (percent starting at 100, or counts). |
| h | Integer forecast horizon (≥ 0). |
| N0 | Cohort size if ‘input = “percent”‘. |
| one_and_done | Logical; include one-and-done parameter d . |
| cure | Logical; include cure fraction parameter (fraction never churning). |
| starts_m, starts_q | Starting values for m, q on the original scale in $(0, 1)$. |

starts_d, starts_cure	Starting values for d and π on the original scale in $[0, 1]$.
compute_se	Logical; compute robust SEs via Hessian + delta method.
input	One of "auto", "percent", "count". If "auto", infer from 'surv_value' and 'N0'.
percent_tol, boundary_tol	Tolerances for monotonicity checks and boundary SE skipping.

Details

The baseline sBG survival (no one-and-done, no cure) at discrete times $t = 0, 1, \dots$ is

$$S(t) = \frac{B(a, b + t)}{B(a, b)}.$$

With one-and-done mass d and cure fraction π , survival for $t \geq 1$ is

$$S(t) = (1 - d) \left[\pi + (1 - \pi) \frac{B(a, b + t)}{B(a, b)} \right], \quad S(0) = 1.$$

Parameters (a, b) (and optionally d, π) are estimated by maximizing the likelihood implied by the observed survival series using an internal unconstrained parameterization. If `compute_se = TRUE`, standard errors on the original scale are obtained via the delta method.

Beta reparameterization by mean and polarization index: The model uses a numerically stable and interpretable reparameterization of the Beta distribution $Beta(a, b)$ in terms of a **mean** $m \in (0, 1)$ and a **polarization (concentration) index** $q \in (0, 1)$.

Standard Beta density:

$$f(x | a, b) = \frac{x^{a-1}(1-x)^{b-1}}{B(a, b)}, \quad x \in (0, 1), \quad a, b > 0.$$

Mapping to mean and polarization:

$$m = \frac{a}{a + b}, \quad q = \frac{1}{1 + a + b}.$$

Here, m is the expected probability, and q summarizes concentration: small q means large $a + b$ (high concentration); large q means small $a + b$ (diffuse).

Inverse mapping (used internally for estimation):

$$a = m \left(\frac{1}{q} - 1 \right), \quad b = (1 - m) \left(\frac{1}{q} - 1 \right).$$

This transformation is bijective for $m \in (0, 1)$, $q \in (0, 1)$, and guarantees $a > 0, b > 0$.

Starting values: The arguments `starts_m` and `starts_q` provide starting values for m and q , respectively. They are converted to (a, b) via the inverse mapping above:

$$a_{\text{start}} = \text{starts_m} \left(\frac{1}{\text{starts_q}} - 1 \right), \quad b_{\text{start}} = (1 - \text{starts_m}) \left(\frac{1}{\text{starts_q}} - 1 \right).$$

This parameterization typically improves optimization stability and makes starting values more interpretable.

Value

An object of class `sbg_fit` with the elements shown below:

`y` Numeric vector of observed survival values on the input scale.

`input` Character scalar, one of "input", "percent", "count", or "prob"; echoes how `y` was interpreted.

`N0` Scalar. Reference cohort size used for count/percent scaling.

`flags` Named logical vector with model switches, e.g. `c(one_and_done = TRUE/FALSE, cure = TRUE/FALSE)`.

`coef_params` Named numeric vector of parameters on the natural scale. For sBG: `c(a,b)` and optionally `d` (one-and-done) and `cure`. For BDW: `c(a,b,c)` and optionally `d`, `cure`.

`coef_repar` Named numeric vector with the mean-polarization reparameterization, typically `c(m,q)` (and `c` for BDW if modeled on the log/exp scale).

`logits` Named numeric vector of unconstrained optimization variables (e.g., `theta_m`, `theta_q`, `theta_d`, `theta_cure`, `theta_c`).

`fitted` Numeric vector $S(0:T_{\text{obs}} - 1)$ on the probability scale; the in-sample fit.

`projected` Numeric vector $S(0:T_{\text{last}})$ (fit + horizon) on the probability scale.

`logLik` Maximized log-likelihood.

`convergence` Optimizer return code (0 indicates successful convergence).

`optim_out` Raw optimizer list (as returned by `optim()`); useful for debugging.

`vcov_theta` Variance-covariance matrix on the unconstrained scale (logits).

`vcov_params` Variance-covariance matrix for `coef_params` (delta-method mapped from `vcov_theta`).

`vcov_repar` Variance-covariance matrix for `coef_repar` (e.g., `m,q`).

`se_params` Named vector of standard errors for `coef_params`.

`se_repar` Named vector of standard errors for `coef_repar`.

`se_note` Character note if SEs are approximate/unstable (e.g., near-PD fix).

References

Fader P, Hardie B. How to project customer retention. *Journal of Interactive Marketing*. 2007;21(1):76-90.

Examples

```
## Not run:
N0 <- 500; S <- c(100, 86.9, 74.3, 65.3, 59.3, 55.1, 51.7, 49.1)
fit <- sbg1c(S, h=6, N0=N0, input="percent")
summary(fit)
plot(fit, scale="percent")

## End(Not run)
```

shiftedBetaGeometric *Shifted Beta-geometric (sbg) Distribution Family Function*

Description

Provides functions for the probability mass function (PMF), cumulative distribution function (CDF), quantile function, and random variate generation for the SBG distribution.

Usage

```
dsbg(x, shape1, shape2, log = FALSE)
psbg(x, shape1, shape2, lower.tail = TRUE, log.p = FALSE)
qsbg(p, shape1, shape2, lower.tail = TRUE, log.p = FALSE)
rsbg(n, shape1, shape2)
```

Arguments

x	Vector of non-negative integers for dsbg and psbg.
p	Vector of probabilities ($0 \leq p \leq 1$) for qsbg.
n	Number of random variates to generate for rsbg.
shape1	First shape parameter "a" (must be > 0).
shape2	Second shape parameter "b" (must be > 0).
log	Logical; if TRUE, probabilities are returned on the log scale (for dsbg and psbg).
lower.tail	Logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise $P(X > x)$ (for psbg).
log.p	Logical; if TRUE, probabilities are returned on the log scale (for psbg and qsbg).

Details

The Shifted Beta Geometric distribution with two shape parameters shape1 (a) and shape2 (b) has the following CDF:

$$1 - \text{Beta}(a, b + x) / \text{Beta}(a, b)$$

For $x = 1, 2, 3, \dots, n$ and $a > 0$ and $b > 0$.

The Shifted Beta Geometric (sBG) distribution, is a probability mixture model of Beta and Geometric distributions. sBG was introduced by Fader and Hardie, models customer retention by assuming that individuals have heterogeneous dropout probabilities. These probabilities are drawn from a Beta distribution, and each customer's retention follows a geometric process. This combination captures variability in churn behavior across a population, making it well-suited for analyzing survival data, customer lifetime and retention data.

Value

- dsbg: A numeric vector of PMF values.
- psbg: A numeric vector of CDF values.
- qsbg: A numeric vector of quantile values.
- rsbg: A numeric vector of random variates.

References

Fader P, Hardie B. How to project customer retention. *Journal of Interactive Marketing*. 2007;21(1):76-90.

Fader P, Hardie B, Liu Y, Davin J, Steenburgh T. "How to Project Customer Retention" Revisited: The Role of Duration Dependence. *Journal of Interactive Marketing*. 2018;43:1-16.

Examples

```
# PMF example
dsbg(1:5, shape1 = 2, shape2 = 3)

# CDF example
psbg(1:5, shape1 = 2, shape2 = 3)

# Quantile example
qsbg(c(0.1, 0.5, 0.9), shape1 = 2, shape2 = 3)

# Random variates
rsbg(10, shape1 = 2, shape2 = 3)
```

sim.bdw1c

Parametric bootstrap uncertainty bands for a beta discrete weibull survival fit

Description

Computes uncertainty bands for a fitted `sbg_fit` object, returning up to three band types:

- **fundamental** - process (sampling) variability at fixed parameters
- **estimation** - parameter uncertainty mapped to the mean survival curve
- **both** - predictive bands (parameter draw + one process draw)

Usage

```
sim.bdw1c(
  object,
  B1 = 1000,
  B2 = 0,
  level = 0.95,
  scale = c("percent", "count", "prob"),
  seed = NULL,
  verbose = FALSE
)
```

Arguments

object	A fitted <code>sbg_fit</code> from your geometric model constructor (e.g., <code>geom_extended()</code>), containing fields <code>y</code> , <code>N0</code> , <code>h</code> , <code>coef_params</code> , <code>logits</code> , <code>vcov_theta</code> , and <code>flags</code> .
B1	Integer, number of parameter draws for estimation and both bands.
B2	Integer, Average process noise in estimation bands (set <code>B2 = 0</code> for pure estimation bands).
level	Confidence level for bands (default 0.95).
scale	Output scale: "prob", "percent", or "count".
seed	Optional integer seed for reproducibility.
verbose	Logical; print minimal progress messages.

Details

Set `B2 = 0` to skip process simulation inside the estimation bands (fast, pure estimation-only) to average process noise in estimation bands.

Value

A list with data frames named `fundamental`, `estimation`, and `both` (whichever were computed). Each data frame has columns: `time`, `S_hat`, `lower`, `upper`, `part`.

References

King G. , Tomz M. and Wittenberg J. Making the most of statistical analyses: Improving interpretation and presentation. American journal of political science, 2000;347-61.

Examples

```
## Not run:
N0 <- 400
S <- c(100, 86.9, 74.3, 65.3, 59.3, 55.1, 51.7, 49.1)
fit <- geom1c(S, h = 6, N0 = N0, input = "percent")
b <- sim.geom1c(fit, B1 = 200, B2 = 50, level = 0.90, scale = "percent", seed = 123)
plot(fit, scale = "percent", bands = b)

## End(Not run)
```

sim.geom1c

Parametric bootstrap uncertainty bands for a geometric survival fit

Description

Computes uncertainty bands for a fitted `geom_fit` object, returning up to three band types:

- **fundamental** - process (sampling) variability at fixed parameters
- **estimation** - parameter uncertainty mapped to the mean survival curve
- **both** - predictive bands (parameter draw + one process draw)

Usage

```
sim.geom1c(
  object,
  B1 = 1000,
  B2 = 0,
  level = 0.95,
  scale = c("percent", "count", "prob"),
  seed = NULL,
  verbose = FALSE
)
```

Arguments

object	A fitted <code>sbg_fit</code> from your shifted beta geometric model constructor (e.g., <code>sbg1c()</code>), containing fields <code>y</code> , <code>N0</code> , <code>h</code> , <code>coef_params</code> , <code>logits</code> , <code>vcov_theta</code> , and <code>flags</code> .
B1	Integer, number of parameter draws and fundamental uncertainty.
B2	Average process noise in estimation bands (set <code>B2 = 0</code> for pure estimation bands).
level	Confidence level for bands (default 0.95).
scale	Output scale: "prob", "percent", or "count".
seed	Optional integer seed for reproducibility.
verbose	Logical; print minimal progress messages.

Details

Set `B2 = 0` to skip process simulation inside the estimation bands (fast, pure estimation-only). Fundamental bands require `B2 > 0`.

Value

A list with data frames named `fundamental`, `estimation`, and `both` (whichever were computed). Each data frame has columns: `time`, `S_hat`, `lower`, `upper`, `part`.

References

King G. , Tomz M. and Wittenberg J. Making the most of statistical analyses: Improving interpretation and presentation. American journal of political science, 2000;347-61.

Examples

```
## Not run:
N0 <- 400
S <- c(100, 86.9, 74.3, 65.3, 59.3, 55.1, 51.7, 49.1)
fit <- sbg1c(S, h = 6, N0 = N0, input = "percent")
b <- sim.sbg1c(fit, B1 = 200, B2 = 50, level = 0.90, scale = "percent", seed = 123)
plot(fit, scale = "percent", bands = b)

## End(Not run)
```

sim.sbg1c	<i>Parametric bootstrap uncertainty bands for a shifted geometric survival fit</i>
-----------	--

Description

Computes uncertainty bands for a fitted `sbg_fit` object, returning up to three band types:

- **fundamental** - process (sampling) variability at fixed parameters
- **estimation** - parameter uncertainty mapped to the mean survival curve
- **both** - predictive bands (parameter draw + one process draw)

Usage

```
sim.sbg1c(
  object,
  B1 = 1000,
  B2 = 0,
  level = 0.95,
  scale = c("percent", "count", "prob"),
  seed = NULL,
  verbose = FALSE
)
```

Arguments

object	A fitted <code>sbg_fit</code> from your geometric model constructor (e.g., <code>geom_extended()</code>), containing fields <code>y</code> , <code>N0</code> , <code>h</code> , <code>coef_params</code> , <code>logits</code> , <code>vcov_theta</code> , and <code>flags</code> .
B1	Integer, number of parameter draws for estimation and both bands.
B2	Integer, Average process noise in estimation bands (set <code>B2 = 0</code> for pure estimation bands).

level	Confidence level for bands (default 0.95).
scale	Output scale: "prob", "percent", or "count".
seed	Optional integer seed for reproducibility.
verbose	Logical; print minimal progress messages.

Details

Set $B2 = 0$ to skip process simulation inside the estimation bands (fast, pure estimation-only) to average process noise in estimation bands.

Value

A list with data frames named `fundamental`, `estimation`, and `both` (whichever were computed). Each data frame has columns: `time`, `S_hat`, `lower`, `upper`, `part`.

References

King G. , Tomz M. and Wittenberg J. Making the most of statistical analyses: Improving interpretation and presentation. American journal of political science, 2000;347-61.

Examples

```
## Not run:
N0 <- 400
S <- c(100, 86.9, 74.3, 65.3, 59.3, 55.1, 51.7, 49.1)
fit <- geom1c(S, h = 6, N0 = N0, input = "percent")
b <- sim.geom1c(fit, B1 = 200, B2 = 50, level = 0.90, scale = "percent", seed = 123)
plot(fit, scale = "percent", bands = b)

## End(Not run)
```

Index

* datasets

- customer_retention, [8](#)
- persistence_data, [13](#)

BdW, [2](#)

bdw1c, [3](#)

BetadiscreteWeibull, [6](#)

BG, [7](#)

customer_retention, [8](#)

exltrend, [9](#)

geom1c, [9](#)

LCW, [11](#)

persistence_data, [13](#)

sbg1c, [14](#)

shiftedBetaGeometric, [17](#)

sim.bdw1c, [18](#)

sim.geom1c, [20](#)

sim.sbg1c, [21](#)