

Package ‘ReliaGrowR’

March 31, 2026

Title Reliability Growth Analysis

Version 0.4

Description Modeling and plotting functions for Reliability Growth Analysis (RGA). Models include the Duane (1962) <[doi:10.1109/TA.1964.4319640](https://doi.org/10.1109/TA.1964.4319640)>, Non-Homogeneous Poisson Process (NHPP) by Crow (1975) (No. AMSAATR138), Piecewise Weibull NHPP by Guo et al. (2010) <[doi:10.1109/RAMS.2010.5448029](https://doi.org/10.1109/RAMS.2010.5448029)>, and Piecewise Weibull NHPP with Change Point Detection based on the 'segmented' package by Muggeo (2024) <<https://cran.r-project.org/package=segmented>>.

Imports graphics, plumber, segmented, stats

License CC BY 4.0

Encoding UTF-8

Suggests ellmer, knitr, pkgload, rmarkdown, spelling, testthat (>= 3.0.0), vdiff, WeibullR

Language en-US

URL <https://paulgovan.github.io/ReliaGrowR/>,
<https://github.com/paulgovan/ReliaGrowR>

Config/testthat/edition 3

VignetteBuilder knitr

BugReports <https://github.com/paulgovan/ReliaGrowR/issues>

RoxygenNote 7.3.3

Depends R (>= 3.5)

LazyData true

NeedsCompilation no

Author Paul Govan [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-1821-8492>>)

Maintainer Paul Govan <paul.govan2@gmail.com>

Repository CRAN

Date/Publication 2026-03-31 14:30:02 UTC

Contents

duane	2
grwr_api	4
plot.duane	4
plot.rga	5
plot.rga_predict	6
ppplot.rga	7
predict_rga	8
print.duane	9
print.rdt	10
print.rga	10
print.rga_predict	11
qqplot.rga	12
rdt	12
rga	14
sim_failures	16
testdata	18
weibull_to_rga	19
Index	21

duane	<i>Duane Analysis</i>
-------	-----------------------

Description

This function performs a Duane analysis (1962) [doi:10.1109/TA.1964.4319640](https://doi.org/10.1109/TA.1964.4319640) on failure data by fitting a log-log linear regression of cumulative Mean Time Between Failures (MTBF) versus cumulative time. The function accepts either two numeric vectors (`times`, `failures`) or a data frame containing both.

Usage

```
duane(times, failures = NULL, conf.level = 0.95)
```

Arguments

<code>times</code>	Either: <ul style="list-style-type: none"> • A numeric vector of exact failure times, or • A data frame containing two columns: <code>times</code> and <code>failures</code>. The <code>times</code> column contains exact failure times, and the <code>failures</code> column contains the number of failures at each corresponding time.
<code>failures</code>	A numeric vector of the number of failures at each corresponding time in <code>times</code> . Ignored if <code>times</code> is a data frame. Must be the same length as <code>times</code> if both are vectors. All values must be positive and finite.
<code>conf.level</code>	Confidence level for the confidence bounds (default: 0.95). Must be between 0 and 1 (exclusive).

Details

The scaling relationship between the size of input data (numbers of observations) and speed of algorithm execution is approximately linear ($O(n)$). The function is efficient and can handle large data sets (e.g., thousands of observations) quickly. The function uses base R functions and does not require any additional packages. The function includes comprehensive input validation and error handling to ensure robustness. The function is tested with a standard data set from a published paper and includes unit tests to verify correctness and performance.

Value

A list of class "duane" containing:

times	The input exact failure times.
failures	The input number of failures.
n_obs	The number of observations (failures).
MTBF	The cumulative mean time between failures.
model	The fitted lm (linear model) object containing the regression results.
logLik	The log-likelihood of the fitted model.
AIC	Akaike Information Criterion (AIC).
BIC	Bayesian Information Criterion (BIC).
conf.level	The confidence level.
Cumulative_Time	The cumulative operating times.
Cumulative_MTBF	The cumulative mean time between failures.
Fitted_Values	The fitted values on the MTBF scale.
Confidence_Bounds	Matrix of fitted values and confidence bounds on the MTBF scale.
Residuals_Log	Residuals on the log(MTBF) scale (from the regression).
Residuals_MTBF	Residuals on the MTBF scale (observed - fitted).

See Also

Other Duane functions: [plot.duane\(\)](#), [print.duane\(\)](#)

Examples

```
times <- c(100, 200, 300, 400, 500)
failures <- c(1, 2, 1, 3, 2)
fit1 <- duane(times, failures, conf.level = 0.90)
print(fit1)

df <- data.frame(times = times, failures = failures)
fit2 <- duane(df, conf.level = 0.95)
print(fit2)
```

`grwr_api`*ReliaGrowR API*

Description

This function provides an interface to the ReliaGrowR API.#' This function provides an interface to the ReliaGrowR API.

Usage

```
grwr_api()
```

Value

Launches the ReliaGrowR API on a local server.

Examples

```
## Not run:  
grwr_api()  
  
## End(Not run)
```

`plot.duane`*Plot Method for Duane Analysis*

Description

Generates a Duane plot (log-log or linear scale) with fitted regression line and optional confidence bounds.

Usage

```
## S3 method for class 'duane'  
plot(  
  x,  
  log = TRUE,  
  conf.int = TRUE,  
  legend = TRUE,  
  legend.pos = "topleft",  
  ...  
)
```

Arguments

x	An object of class "duane".
log	Logical; whether to use logarithmic scales for axes (default: TRUE).
conf.int	Logical; whether to plot confidence bounds (default: TRUE).
legend	Logical; whether to include a legend (default: TRUE).
legend.pos	Position of the legend (default: "topleft").
...	Further arguments passed to plot().

Value

Invisibly returns NULL.

See Also

Other Duane functions: [duane\(\)](#), [print.duane\(\)](#)

Examples

```
times <- c(100, 200, 300, 400, 500)
failures <- c(1, 2, 1, 3, 2)
fit <- duane(times, failures)
plot(fit, main = "Duane Plot", xlab = "Cumulative Time", ylab = "Cumulative MTBF")
```

plot.rga

Plot Method for RGA Objects

Description

This function generates plots for objects of class rga.

Usage

```
## S3 method for class 'rga'
plot(
  x,
  conf_bounds = TRUE,
  legend = TRUE,
  log = FALSE,
  legend_pos = "bottomright",
  ...
)
```

Arguments

x	An object of class rga, which contains the results from the RGA model.
conf_bounds	Logical; include confidence bounds (default: TRUE).
legend	Logical; show the legend (default: TRUE).
log	Logical; use a log-log scale (default: FALSE).
legend_pos	Position of the legend (default: "bottomright").
...	Additional arguments passed to plot().

Value

Invisibly returns NULL.

See Also

Other Reliability Growth Analysis: [plot.rga_predict\(\)](#), [predict_rga\(\)](#), [print.rga\(\)](#), [print.rga_predict\(\)](#), [rga\(\)](#)

Examples

```
times <- c(100, 200, 300, 400, 500)
failures <- c(1, 2, 1, 3, 2)
result <- rga(times, failures)
plot(result,
      main = "Reliability Growth Analysis",
      xlab = "Cumulative Time", ylab = "Cumulative Failures"
    )
```

plot.rga_predict *Plot Method for rga_predict Objects*

Description

Plots observed data, the fitted reliability growth curve, and the forecast with optional confidence bounds for an rga_predict object.

Usage

```
## S3 method for class 'rga_predict'
plot(x, conf_bounds = TRUE, legend = TRUE, legend_pos = "bottomright", ...)
```

Arguments

x	An object of class rga_predict.
conf_bounds	Logical; include confidence bounds (default: TRUE).
legend	Logical; show the legend (default: TRUE).
legend_pos	Position of the legend (default: "bottomright").
...	Additional arguments passed to plot().

Value

Invisibly returns NULL.

See Also

Other Reliability Growth Analysis: [plot.rga\(\)](#), [predict_rga\(\)](#), [print.rga\(\)](#), [print.rga_predict\(\)](#), [rga\(\)](#)

Examples

```
times <- c(100, 200, 300, 400, 500)
failures <- c(1, 2, 1, 3, 2)
fit <- rga(times, failures)
fc <- predict_rga(fit, times = c(1500, 2000))
plot(fc, main = "RGA Forecast", xlab = "Cumulative Time", ylab = "Cumulative Failures")
```

ppplot.rga

P-P Plot for RGA Objects

Description

This function creates a P-P plot for a fitted Reliability Growth Analysis (RGA) model. Currently only supports the Crow-AMSAA model. A P-P plot compares the empirical cumulative distribution function (CDF) to the theoretical CDF specified by the model. If the model fits well, the points should fall approximately along a straight line.

Usage

```
ppplot.rga(x, main = "P-P Plot", ...)
```

Arguments

x	An object of class rga.
main	Title of the plot.
...	Additional arguments passed to <code>plot()</code> .

Value

A P-P plot comparing empirical and theoretical CDFs.

See Also

Other goodness-of-fit: [qqplot.rga\(\)](#)

Examples

```
times <- c(5, 10, 15, 20, 25)
failures <- c(1, 2, 1, 3, 2)
fit <- rga(times, failures)
ppplot.rga(fit)
```

predict_rga

Forecast Cumulative Failures from a Reliability Growth Model

Description

Takes a fitted rga object and a vector of cumulative times, returning predicted cumulative failures with confidence bounds as an rga_predict S3 object.

Usage

```
predict_rga(object, times, conf_level = 0.95)
```

Arguments

object	An object of class rga returned by rga().
times	A numeric vector of cumulative times at which to forecast. All values must be finite and > 0. A warning is issued if any value is at or below the maximum observed cumulative time (hindcasting).
conf_level	The desired confidence level (default 0.95). Must be a single finite numeric in (0, 1).

Value

An object of class rga_predict containing:

times	The forecast cumulative times.
cum_failures	Predicted cumulative failures.
lower_bounds	Lower confidence bounds.
upper_bounds	Upper confidence bounds.
conf_level	The confidence level used.
model_type	Either "Crow-AMSAA" or "Piecewise NHPP".
rga_object	The original rga object (used by the plot method).

See Also

Other Reliability Growth Analysis: [plot.rga\(\)](#), [plot.rga_predict\(\)](#), [print.rga\(\)](#), [print.rga_predict\(\)](#), [rga\(\)](#)

Examples

```
times <- c(100, 200, 300, 400, 500)
failures <- c(1, 2, 1, 3, 2)
fit <- rga(times, failures)
fc <- predict_rga(fit, times = c(1500, 2000))
print(fc)
```

print.duane	<i>Print method for duane objects.</i>
-------------	--

Description

This function prints a summary of the Duane analysis result.

Usage

```
## S3 method for class 'duane'
print(x, ...)
```

Arguments

x	An object of class "duane" returned by the duane_plot function.
...	Additional arguments (not used).

Value

Invisibly returns the input object.

See Also

Other Duane functions: [duane\(\)](#), [plot.duane\(\)](#)

Examples

```
times <- c(100, 200, 300, 400, 500)
failures <- c(1, 2, 1, 3, 2)
fit <- duane(times, failures)
print(fit)
```

print.rdt	<i>Print method for rdt objects</i>
-----------	-------------------------------------

Description

This function provides a formatted print method for objects of class rdt.

Usage

```
## S3 method for class 'rdt'  
print(x, ...)
```

Arguments

x	An object of class rdt.
...	Additional arguments (not used).

Value

Invisibly returns the input object.

Examples

```
plan <- rdt(target = 0.9, mission_time = 1000, conf_level = 0.9, beta = 1, n = 10)  
print(plan)
```

print.rga	<i>Print method for rga objects.</i>
-----------	--------------------------------------

Description

This function prints a summary of the results from an object of class rga.

Usage

```
## S3 method for class 'rga'  
print(x, ...)
```

Arguments

x	An object of class rga, which contains the results from the RGA model.
...	Additional arguments (not used).

Value

Invisibly returns the input object.

See Also

Other Reliability Growth Analysis: [plot.rga\(\)](#), [plot.rga_predict\(\)](#), [predict_rga\(\)](#), [print.rga_predict\(\)](#), [rga\(\)](#)

Examples

```
times <- c(100, 200, 300, 400, 500)
failures <- c(1, 2, 1, 3, 2)
result <- rga(times, failures)
print(result)
```

`print.rga_predict` *Print Method for rga_predict Objects*

Description

Prints a formatted table of forecast cumulative failures with confidence bounds for an `rga_predict` object.

Usage

```
## S3 method for class 'rga_predict'
print(x, ...)
```

Arguments

`x` An object of class `rga_predict`.
`...` Additional arguments (not used).

Value

Invisibly returns the input object.

See Also

Other Reliability Growth Analysis: [plot.rga\(\)](#), [plot.rga_predict\(\)](#), [predict_rga\(\)](#), [print.rga\(\)](#), [rga\(\)](#)

Examples

```
times <- c(100, 200, 300, 400, 500)
failures <- c(1, 2, 1, 3, 2)
fit <- rga(times, failures)
fc <- predict_rga(fit, times = c(1500, 2000))
print(fc)
```

`qqplot.rga`*Q-Q Plot for RGA Objects*

Description

This function creates a Q-Q plot for a fitted Reliability Growth Analysis (RGA) model. Currently only supports the Crow-AMSAA model. A Q-Q plot compares the quantiles of the empirical data to the quantiles of the theoretical distribution specified by the model. If the model fits well, the points should fall approximately along a straight line.

Usage

```
qqplot.rga(x, main = "Q-Q Plot", ...)
```

Arguments

<code>x</code>	An object of class <code>rga</code> .
<code>main</code>	Title of the plot.
<code>...</code>	Additional arguments passed to <code>stats::qqplot()</code> .

Value

A Q-Q plot comparing empirical and theoretical quantiles.

See Also

Other goodness-of-fit: [ppplot.rga\(\)](#)

Examples

```
times <- c(5, 10, 15, 20, 25)
failures <- c(1, 2, 1, 3, 2)
fit <- rga(times, failures)
qqplot.rga(fit)
```

`rdt`*Reliability Demonstration Test (RDT) Plan Calculator*

Description

This function calculates the required test time or sample size for a Reliability Demonstration Test (RDT) based on specified reliability, mission time, confidence level, and Weibull shape parameter.

Usage

```
rdt(
  target,
  mission_time,
  conf_level,
  beta = 1,
  f = 0,
  n = NULL,
  test_time = NULL
)
```

Arguments

target	Required reliability at mission time ($0 < \text{target} < 1$).
mission_time	Mission duration (time units). Must be greater than 0.
conf_level	Desired confidence level (e.g., 0.9 for 90% confidence). The confidence level must be between 0 and 1 (exclusive).
beta	Weibull shape parameter (beta=1 corresponds to exponential distribution). Must be greater than 0. Default is 1.
f	Number of allowable failures during the test (non-negative integer). Default is 0 (zero-failure test plan). Increasing f reduces the required test time or sample size at the cost of accepting more observed failures.
n	Sample size (optional, supply if solving for test_time). Must be a positive integer.
test_time	Test time per unit (optional, supply if solving for n). Must be greater than 0.

Value

The function returns an object of class rdt that contains:

Distribution	Type of distribution used (Exponential or Weibull).
Beta	Weibull shape parameter.
Allowed_Failures	Number of allowable failures during the test.
Target_Reliability	Specified target reliability.
Mission_Time	Specified mission time.
Required_Test_Time	Calculated required test time (if n is provided).
Input_Sample_Size	Provided sample size (if test_time is calculated).
Required_Sample_Size	Calculated required sample size (if test_time is provided).
Input_Test_Time	Provided test time (if n is calculated).

Examples

```
#' # Example 1: Calculate required test time
plan1 <- rdt(target = 0.9, mission_time = 1000, conf_level = 0.9, beta = 1, n = 10)
print(plan1)
# Example 2: Calculate required sample size
plan2 <- rdt(target = 0.9, mission_time = 1000, conf_level = 0.9, beta = 1, test_time = 2000)
print(plan2)
```

 rga

Reliability Growth Analysis.

Description

This function performs reliability growth analysis using the Crow-AMSAA model by Crow (1975) (AMSAATR138) or piecewise NHPP model by Guo et al. (2010) [doi:10.1109/RAMS.2010.5448029](https://doi.org/10.1109/RAMS.2010.5448029). It fits a log-log linear regression of cumulative failures versus cumulative time. The function accepts either two numeric vectors (`times`, `failures`) or a data frame containing both. The Piecewise NHPP model can automatically detect change points or use user-specified breakpoints.

Usage

```
rga(
  times,
  failures,
  times_type = c("failure_times", "cumulative_failure_times"),
  model_type = "Crow-AMSAA",
  breaks = NULL,
  conf_level = 0.95,
  method = c("LS", "MLE")
)
```

Arguments

<code>times</code>	Either a numeric vector of failure-time inputs or a data frame containing both time inputs and failure counts. If <code>times_type = "failure_times"</code> (default), <code>times</code> is treated exactly as in previous versions of the function and is cumulatively summed inside <code>rga()</code> . If <code>times_type = "cumulative_failure_times"</code> , <code>times</code> is treated as already cumulative and is used directly without applying <code>cumsum()</code> . If a data frame is provided, it must contain two columns: <code>times</code> and <code>failures</code> .
<code>failures</code>	A numeric vector of the number of failures at each corresponding time in <code>times</code> . Must be the same length as <code>times</code> if both are vectors. All values must be positive and finite. Ignored if <code>times</code> is a data frame.
<code>times_type</code>	Character scalar indicating how to interpret <code>times</code> . <code>"failure_times"</code> (default) preserves the current behavior and cumulatively sums <code>times</code> inside <code>rga()</code> . <code>"cumulative_failure_times"</code> treats <code>times</code> as already cumulative and skips that internal <code>cumsum()</code> .

model_type	The model type. Either Crow-AMSAA (default) or Piecewise NHPP with change point detection.
breaks	An optional vector of breakpoints for the Piecewise NHPP model.
conf_level	The desired confidence level, which defaults to 95%. The confidence level is the probability that the confidence interval contains the true mean response.
method	Estimation method: "LS" (default) for least-squares log-log regression, or "MLE" for maximum likelihood estimation of the Crow-AMSAA model. "MLE" is not supported for model_type = "Piecewise NHPP".

Details

The scaling relationship between the size of input data (numbers of observations) and speed of algorithm execution is approximately linear ($O(n)$). The function is efficient and can handle large data sets (e.g., thousands of observations) quickly. The function uses the segmented package for piecewise regression, which employs an iterative algorithm to estimate breakpoints. The number of iterations required for convergence may vary depending on the data and initial values. In practice, the function typically converges within a few iterations for most data sets. However, in some cases, especially with complex data or poor initial values, it may take more iterations.

Value

The function returns an object of class `rga` that contains:

times	The input time vector, stored exactly as supplied.
cum_times	The cumulative time vector used for fitting.
times_type	How times was interpreted: "failure_times" or "cumulative_failure_times".
failures	The input number of failures.
n_obs	The number of observations (failures).
cum_failures	Cumulative failures.
model	The fitted model object (lm (linear model) or segmented).
residuals	Model residuals on the log-log scale. These represent deviations of the observed log cumulative failures from the fitted values and are useful for diagnostic checking.
logLik	The log-likelihood of the fitted model. The log-likelihood is a measure of model fit, with higher values indicating a better fit.
AIC	Akaike Information Criterion (AIC). AIC is a measure used for model selection, with lower values indicating a better fit.
BIC	Bayesian Information Criterion (BIC). BIC is another criterion for model selection.
breakpoints	Breakpoints (log scale) if applicable.
fitted_values	Fitted cumulative failures on the original scale.
lower_bounds	Lower confidence bounds (original scale).
upper_bounds	Upper confidence bounds (original scale).
betas	Estimated beta(s). Betas are the slopes of the log-log plot.

betas_se Standard error(s) of the estimated beta(s).
 growth_rate Estimated growth rate(s). Growth rates are calculated as 1 - beta.
 lambdas Estimated lambda(s). Lambdas are the intercepts of the log-log plot.

See Also

Other Reliability Growth Analysis: [plot.rga\(\)](#), [plot.rga_predict\(\)](#), [predict_rga\(\)](#), [print.rga\(\)](#), [print.rga_predict\(\)](#)

Examples

```
times <- c(100, 200, 300, 400, 500)
failures <- c(1, 2, 1, 3, 2)
result1 <- rga(times, failures)
print(result1)

df <- data.frame(times = times, failures = failures)
result2 <- rga(df)
print(result2)

cum_times <- cumsum(times)
result2b <- rga(cum_times, failures, times_type = "cumulative_failure_times")
print(result2b)

result3 <- rga(times, failures, model_type = "Piecewise NHPP")
print(result3)

result4 <- rga(times, failures, model_type = "Piecewise NHPP", breaks = c(450))
print(result4)
```

sim_failures

Simulate Failures from a Conditional Weibull Model

Description

Simulates which units in a non-failed population fail next by using a Weibull life model conditional on each unit's current runtime. When a positive window is supplied, the function calibrates a Weibull scale parameter (unless provided directly) so that the expected number of failures within the window matches n . Units are then sampled with probability proportional to their conditional Weibull failure probability over the window, and failure times are drawn from the truncated conditional Weibull distribution. The full fleet is returned: selected units are labelled "Failure" and the remaining units are labelled "Suspension".

Usage

```
sim_failures(n, runtimes, replace = FALSE, window = NULL, beta = 1, eta = NULL)
```

Arguments

n	Positive integer. Number of failures to simulate.
runtimes	Numeric vector of positive values. The current operating runtime of each unit in the non-failed population.
replace	Logical scalar. If TRUE, sampling is done with replacement (a unit may be selected more than once). Default is FALSE.
window	NULL or a single positive numeric. The width of the observation window. When NULL (default), event times equal current runtimes. When provided, failure times are sampled from the conditional Weibull distribution over (runtime, runtime + window], and suspension times are runtime + window.
beta	Positive numeric scalar. Weibull shape parameter used to model the age-dependent hazard. Defaults to 1, corresponding to an exponential-process assumption.
eta	NULL or a single positive numeric. Weibull scale parameter. When NULL and window is supplied, the scale is calibrated so that the expected number of failures across the fleet during the window matches n.

Details

When window = NULL, the function returns the current fleet state at the supplied runtimes. In this case, failing units are selected using relative Weibull hazard weights implied by beta.

Value

A data frame with length(runtimes) rows sorted ascending by runtime, containing:

index	Integer index of the unit in runtimes.
runtime	Simulated event time.
type	Character; "Failure" for selected units, "Suspension" for the rest.

The returned object also carries attributes weibull_beta and weibull_eta describing the Weibull parameters used for the simulation.

See Also

Other data preparation: [weibull_to_rga\(\)](#)

Examples

```
set.seed(42)
runtimes <- c(100, 500, 200, 800, 300)
result <- sim_failures(2, runtimes, beta = 1.5)
print(result)

# With an observation window
set.seed(42)
result_w <- sim_failures(2, runtimes, window = 50, beta = 1.5)
print(result_w)
```

testdata	<i>Reliability Test Data</i>
----------	------------------------------

Description

A dataset containing example reliability test data from the military report "Reliability Growth Prediction" (1986) by The Analytical Sciences Corporation. This dataset includes cumulative ETI, failure counts, cumulative MTBF, report numbers, flags, and causes for two different LRUs (G1 and G2).

Usage

```
testdata
```

Format

@format ## testdata A data frame with 25 rows and 6 variables:

LRU The Line Replaceable Unit identifier (G1 or G2).

Cum_ETI Cumulative Equivalent Test Hours (ETI).

Failure_Count Cumulative number of failures observed.

Cum_MTBF Cumulative Mean Time Between Failures (MTBF).

Report_No Report number associated with the failure.

Flag A flag indicating special conditions or notes.

Cause Cause of the failure (e.g., D for Design, M for Manufacturing, R for Random, NR for No Report).

```
@usage data(testdata)
```

Examples

```
data(testdata)
head(testdata)
summary(testdata)
str(testdata)
```

weibull_to_rga	<i>Weibull to RGA</i>
----------------	-----------------------

Description

Converts Weibull data (failure, suspension, and interval-censored times) into a format suitable for reliability growth analysis (RGA). The function handles exact failure times, right-censored suspensions, and interval-censored data. It approximates interval-censored failures by placing them at the midpoint of the interval. The output is a data frame with cumulative time and failure counts. This format can be used with RGA models such as Crow-AMSAA.

Usage

```
weibull_to_rga(  
  failures,  
  suspensions = NULL,  
  interval_starts = NULL,  
  interval_ends = NULL  
)
```

Arguments

failures	A numeric vector of exact failure times. Each failure time indicates when an item failed during the observation period.
suspensions	A numeric vector of suspension (right-censored) times. A suspension indicates that the item was removed from observation at that time without failure. This parameter is optional and can be NULL if there are no suspensions.
interval_starts	A numeric vector of interval start times (lower bound of censoring). This parameter is optional and can be NULL if there are no interval-censored data. If provided, it must be the same length as <code>interval_ends</code> .
interval_ends	A numeric vector of interval end times (upper bound of censoring). This parameter is optional and can be NULL if there are no interval-censored data. If provided, it must be the same length as <code>interval_starts</code> .

Value

The data frame contains two columns:

CumulativeTime Cumulative time at each failure event.

Failures Number of failures at each cumulative time point.

The function approximates interval-censored failures by placing them at the midpoint of the interval.

See Also

Other data preparation: [sim_failures\(\)](#)

Examples

```
failures <- c(100, 200, 200, 400)
suspensions <- c(250, 350, 450)
interval_starts <- c(150, 300)
interval_ends <- c(180, 320)
result <- weibull_to_rga(failures, suspensions, interval_starts, interval_ends)
print(result)
```

Index

- * **Duane functions**
 - duane, 2
 - plot.duane, 4
 - print.duane, 9
 - * **Reliability Growth Analysis**
 - plot.rga, 5
 - plot.rga_predict, 6
 - predict_rga, 8
 - print.rga, 10
 - print.rga_predict, 11
 - rga, 14
 - * **data preparation**
 - sim_failures, 16
 - weibull_to_rga, 19
 - * **datasets**
 - testdata, 18
 - * **goodness-of-fit**
 - ppplot.rga, 7
 - qqplot.rga, 12
- duane, 2, 5, 9
- grwr_api, 4
- plot.duane, 3, 4, 9
- plot.rga, 5, 7, 8, 11, 16
- plot.rga_predict, 6, 6, 8, 11, 16
- ppplot.rga, 7, 12
- predict_rga, 6, 7, 8, 11, 16
- print.duane, 3, 5, 9
- print.rdt, 10
- print.rga, 6–8, 10, 11, 16
- print.rga_predict, 6–8, 11, 11, 16
- qqplot.rga, 7, 12
- rdt, 12
- rga, 6–8, 11, 14
- sim_failures, 16, 19
- testdata, 18
- weibull_to_rga, 17, 19