Package: ReliaPlotR (via r-universe)

June 5, 2026

Type Package

Title Interactive Reliability Probability Plots

Version 0.7

Description Build interactive Reliability Probability Plots with 'plotly' by Carson Sievert (2020) <https://plotly.com/r/>, an interactive web-based graphing library.

URL https://paulgovan.github.io/ReliaPlotR/, https://github.com/paulgovan/ReliaPlotR

BugReports https://github.com/paulgovan/ReliaPlotR/issues

License Apache License

Imports plotly, ReliaGrowR, WeibullR, WeibullR.ALT

Suggests ellmer, jsonlite, knitr, mcptools, rmarkdown, spelling, testthat (>= 3.0.0)

Encoding UTF-8

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/testthat/edition 3

Language en-US

Config/pak/sysreqs cmake make libicu-dev libsodium-dev libuv1-dev libssl-dev zlib1g-dev

Repository https://paulgovan.r-universe.dev

Date/Publication 2026-06-05 21:58:19 UTC

RemoteUrl https://github.com/paulgovan/reliaplotr

RemoteRef HEAD

RemoteSha 4034e0e2a06e2098e78e7aa6bfd01f7df407b029

Contents

`plotly_alt` *Interactive ALT Probability Plot.*

Description

Creates an interactive probability plot for an ‘alt‘ object, overlaying one Weibull or lognormal fit line per stress level on a shared probability paper. The ‘alt‘ object must have been processed through [WeibullR.ALT::alt.parallel()] before passing to this function.

Usage

```
plotly_alt(
  alt_obj,
  showConf = TRUE,
  showGrid = TRUE,
  main = "ALT Probability Plot",
  xlab = "Time to Failure",
  ylab = "Probability",
  gridCol = "lightgray",
  signif = 3,
  cols = NULL
)
```

Arguments

`alt_obj`	An object of class ‘'alt'‘ created by the ‘WeibullR.ALT‘ package and fitted with ‘alt.parallel()‘.
`showConf`	Show Fisher-matrix confidence bounds (TRUE) or not (FALSE). Default is TRUE.
`showGrid`	Show grid (TRUE) or hide grid (FALSE). Default is TRUE.

main	Main title. Default is "ALT Probability Plot".
xlab	X-axis label. Default is "Time to Failure".
ylab	Y-axis label. Default is "Probability".
gridCol	Color of the grid. Default is "lightgray".
signif	Significant digits for hover text. Default is 3.
cols	Optional character vector of colors, one per stress level. Recycled to match the number of stress levels. When NULL a 10-color default palette is used.

Details

Building an alt object requires three steps:

1. alt.make() — bundle the per-stress data sets and choose a distribution ("weibull" or "lognormal") and life-stress model ("arrhenius" or "power").
2. alt.parallel() — fit independent Weibull/lognormal models at each stress level (required before calling plotly_alt()).
3. alt.fit() — fit the global life-stress relationship (required for plotly_rel()).

The probability paper transformation (log y-axis for Weibull, normal quantile for lognormal) is the same as in [plotly_wblr()].

Value

A ‘plotly‘ object representing the interactive ALT probability plot.

See Also

[plotly_rel()] for the life-stress relationship plot using the same alt object.

Examples

```
library(WeibullR.ALT)
d1 <- alt.data(c(248, 456, 528, 731, 813, 537), stress = 300)
d2 <- alt.data(c(164, 176, 289), stress = 350)
d3 <- alt.data(c(88, 112, 152), stress = 400)
obj <- alt.fit(
  alt.parallel(
   alt.make(list(d1, d2, d3), dist = "weibull", alt.model = "arrhenius", view_dist_fits = FALSE),
    view_parallel_fits = FALSE
  )
)
plotly_alt(obj)
```

plotly_contour *Interactive Contour Plot*

Description

This function creates an interactive contour plot for one or more ‘wblr‘ objects, each assumed to have confidence contours generated via ‘method.conf = ’lrb’‘. The function overlays all contours in a single plot and displays their respective MLE point estimates.

Usage

```
plotly_contour(
  wblr_obj,
  main = "Contour Plot",
  xlab = "Eta",
  ylab = "Beta",
  showGrid = TRUE,
  cols = NULL,
  gridCol = "lightgray",
  signif = 3
)
```

Arguments

wblr_obj	A single ‘wblr‘ object or a list of ‘wblr‘ objects. Each object must have contours generated using ‘method.conf = ’lrb’‘.
main	Main title for the plot.
xlab	X-axis label (typically Eta or Sigmalog).
ylab	Y-axis label (typically Beta or Mulog).
showGrid	Logical; whether to show grid lines (default TRUE).
cols	Optional vector of colors for each contour/estimate pair. If not provided, colors are chosen from a default palette.
gridCol	Color of the grid lines (default ’lightgray’).
signif	Number of significant digits to display for estimates and contour coordinates. Defaults to 3.

Details

Confidence contours require a likelihood-ratio-based (`method.conf = 'lrb'`) MLE fit (`method.fit = 'mle'`). The contour traces the locus of (η, β) pairs whose log-likelihood falls within a chi-squared critical value of the MLE peak, giving a joint confidence region for the two Weibull parameters. The MLE point estimate is shown as a marker at the center of each contour.

Value

A ‘plotly‘ object representing the interactive contour plot.

See Also

[plotly_wblr()] for probability plots using the same `wblr` objects.

Examples

```
library(WeibullR)
library(ReliaPlotR)

failures1 <- c(30, 49, 82, 90, 96)
failures2 <- c(20, 40, 60, 80, 100)
obj1 <- wblr.conf(wblr.fit(wblr(failures1), method.fit = "mle"), method.conf = "lrb")
obj2 <- wblr.conf(wblr.fit(wblr(failures2), method.fit = "mle"), method.conf = "lrb")
plotly_contour(list(obj1, obj2), main = "Overlayed Contours")
```

---

plotly_duane *Interactive Duane Plot.*

Description

This function creates an interactive Duane plot for a duane object. The plot displays observed cumulative MTBF against cumulative test time on a log-log scale alongside the fitted Duane line and optional confidence bounds. Positive slope on the log-log scale indicates reliability growth.

Usage

```
plotly_duane(
  duane_obj,
  showConf = TRUE,
  showGrid = TRUE,
  main = "Duane Plot",
  xlab = "Cumulative Time",
  ylab = "Cumulative MTBF",
  pointCol = "black",
  fitCol = "black",
  confCol = "black",
  gridCol = "lightgray",
  signif = 3
)
```

Arguments

duane_obj	An object of class 'duane'. This object is created using the 'duane' function from the ReliaGrowR package.
showConf	Show the confidence bounds (TRUE) or not (FALSE). Default is TRUE.
showGrid	Show grid (TRUE) or hide grid (FALSE). Default is TRUE.
main	Main title. Default is "Duane Plot".

xlab	X-axis label. Default is "Cumulative Time".
ylab	Y-axis label. Default is "Cumulative MTBF".
pointCol	Color of the point values. Default is "black".
fitCol	Color of the model fit. Default is "black".
confCol	Color of the confidence bounds. Default is "black".
gridCol	Color of the grid. Default is "lightgray".
signif	Significant digits of results. Default is 3. Must be a positive integer.

Details

The Duane model fits a power-law relationship between cumulative MTBF and cumulative test time: $\text{MTBF}_c = K \cdot T^{\alpha}$. On a log-log plot this appears as a straight line with slope α (the growth rate) and intercept $\log(K)$. A slope near 0 indicates no growth; a slope near 0.5 is typical for a managed development program.

Value

A 'plotly' object representing the interactive Duane plot.

See Also

[plotly_rga()] for the Crow-AMSAA NHPP growth model on the same data.

Examples

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

---

plotly_exposure *Interactive Exposure Plot.*

Description

The function creates an interactive exposure plot for one or more `exposure` objects. When a list of objects is provided the estimates are overlaid on the same plot, each rendered in a distinct color. The plot shows the instantaneous event rate (events per unit time per system at risk) as a step function, calculated from recurrence data by dividing the event count in each interval by the total system-time at risk during that interval.

Usage

```
plotly_exposure(
  exposure_obj,
  showGrid = TRUE,
  main = "Exposure Plot",
  xlab = "Time",
  ylab = "Event Rate",
  fitCol = "black",
  gridCol = "lightgray",
  signif = 3,
  cols = NULL
)
```

Arguments

exposure_obj	An object of class 'exposure', or a list of such objects. Each object is created using the 'exposure()' function from the 'ReliaGrowR' package.
showGrid	Show grid (TRUE) or hide grid (FALSE). Default is TRUE.
main	Main title. Default is "Exposure Plot".
xlab	X-axis label. Default is "Time".
ylab	Y-axis label. Default is "Event Rate".
fitCol	Color of the event rate step function. Default is "black". Used only for a single exposure object; ignored when 'cols' is provided or multiple objects are supplied.
gridCol	Color of the grid. Default is "lightgray".
signif	Significant digits of results. Default is 3. Must be a positive integer.
cols	Optional character vector of colors, one per exposure object. When provided, each object's step function is drawn in the corresponding color. Recycled if shorter than the number of objects.

Details

Unlike [plotly_mcf()] which shows cumulative events, the exposure plot shows the instantaneous event rate: the number of events per unit time per system at risk in each interval. A flat exposure plot suggests a constant event rate (homogeneous Poisson process); a declining rate suggests improvement; a rising rate suggests degradation. Use this plot alongside [plotly_mcf()] to diagnose time-dependence in the event process.

Value

A 'plotly' object representing the interactive exposure plot.

See Also

[plotly_mcf()] for the cumulative view; [plotly_nhpp()] for a parametric NHPP model.

Examples

```
library(ReliaGrowR)
ids <- c("A", "A", "A", "B", "B", "C", "C", "C", "C")
times <- c(50, 150, 350, 100, 300, 80, 200, 320, 450)
fit <- exposure(id = ids, time = times)
plotly_exposure(fit)

# Overlay two exposure objects
fit2 <- exposure(id = c("X", "X", "Y"), time = c(60, 220, 180))
plotly_exposure(list(fit, fit2), cols = c("steelblue", "tomato"))
```

---

plotly_mcf *Interactive Mean Cumulative Function Plot.*

Description

The function creates an interactive Mean Cumulative Function (MCF) plot for one or more ‘mcf‘ objects. When a list of objects is provided the models are overlaid on the same plot, each rendered in a distinct color. The MCF is rendered as a step function. Optional confidence bounds are shown as a shaded band around the estimate.

Usage

```
plotly_mcf(
  mcf_obj,
  showConf = TRUE,
  showGrid = TRUE,
  main = "Mean Cumulative Function Plot",
  xlab = "Time",
  ylab = "Mean Cumulative Function",
  fitCol = "black",
  confCol = "black",
  gridCol = "lightgray",
  signif = 3,
  cols = NULL
)
```

Arguments

mcf_obj	An object of class 'mcf', or a list of such objects. Each object is created using the ‘mcf()‘ function from the ‘ReliaGrowR‘ package.
showConf	Show the confidence bounds (TRUE) or not (FALSE). Default is TRUE.
showGrid	Show grid (TRUE) or hide grid (FALSE). Default is TRUE.
main	Main title. Default is "Mean Cumulative Function Plot".
xlab	X-axis label. Default is "Time".
ylab	Y-axis label. Default is "Mean Cumulative Function".

fitCol	Color of the MCF step function. Default is "black". Used only for a single mcf object; ignored when ‘cols‘ is provided or multiple objects are supplied.
confCol	Color of the confidence bounds. Default is "black". Used only for a single mcf object; ignored when ‘cols‘ is provided or multiple objects are supplied.
gridCol	Color of the grid. Default is "lightgray".
signif	Significant digits of results. Default is 3. Must be a positive integer.
cols	Optional character vector of colors, one per mcf object. When provided, each object's step function and confidence bounds are drawn in the corresponding color. Recycled if shorter than the number of objects.

Details

The MCF is a nonparametric estimate of the expected cumulative number of events per system by time t. It is computed from recurrence data (multiple events per system) and rendered as a step function. The slope of the MCF at any point approximates the current event rate (repairs per unit time). Confidence bounds are computed using the Nelson variance estimator.

Value

A ‘plotly‘ object representing the interactive MCF plot.

See Also

[plotly_nhpp()] to overlay a parametric Power Law model on the MCF; [plotly_exposure()] for the cumulative event rate.

Examples

```
library(ReliaGrowR)
ids <- c("A", "A", "A", "B", "B", "C", "C", "C", "C")
times <- c(50, 150, 350, 100, 300, 80, 200, 320, 450)
fit <- mcf(id = ids, time = times)
plotly_mcf(fit)

# Overlay two MCF objects
fit2 <- mcf(id = c("X", "X", "Y"), time = c(60, 220, 180))
plotly_mcf(list(fit, fit2), cols = c("steelblue", "tomato"))
```

---

plotly_nhpp *Interactive NHPP Plot.*

Description

The function creates an interactive Non-Homogeneous Poisson Process (NHPP) plot for one or more ‘nhpp‘ objects. When a list of objects is provided the models are overlaid on the same plot, each rendered in a distinct color. The plot shows the nonparametric Mean Cumulative Function (MCF) alongside the parametric model fit and optional confidence bounds. Vertical lines indicate change points if breakpoints are specified in the nhpp object.

Usage

```
plotly_nhpp(
  nhpp_obj,
  showConf = TRUE,
  showGrid = TRUE,
  main = "NHPP Plot",
  xlab = "Cumulative Time",
  ylab = "Mean Cumulative Function",
  pointCol = "black",
  fitCol = "black",
  confCol = "black",
  gridCol = "lightgray",
  breakCol = "black",
  signif = 3,
  cols = NULL
)
```

Arguments

nhpp_obj	An object of class 'nhpp', or a list of such objects. Each object is created using the 'nhpp()' function from the 'ReliaGrowR' package.
showConf	Show the confidence bounds (TRUE) or not (FALSE). Default is TRUE.
showGrid	Show grid (TRUE) or hide grid (FALSE). Default is TRUE.
main	Main title. Default is "NHPP Plot".
xlab	X-axis label. Default is "Cumulative Time".
ylab	Y-axis label. Default is "Mean Cumulative Function".
pointCol	Color of the MCF data points. Default is "black". Used only for a single nhpp object; ignored when 'cols' is provided or multiple objects are supplied.
fitCol	Color of the model fit. Default is "black". Used only for a single nhpp object; ignored when 'cols' is provided or multiple objects are supplied.
confCol	Color of the confidence bounds. Default is "black". Used only for a single nhpp object; ignored when 'cols' is provided or multiple objects are supplied.
gridCol	Color of the grid. Default is "lightgray".
breakCol	Color of the breakpoints. Default is "black". Used only for a single nhpp object; ignored when 'cols' is provided or multiple objects are supplied.
signif	Significant digits of results. Default is 3. Must be a positive integer.
cols	Optional character vector of colors, one per nhpp object. When provided, each object's points, fit line, confidence bounds, and breakpoints are all drawn in the corresponding color. Recycled if shorter than the number of objects.

Details

The nonparametric MCF is overlaid with the fitted parametric Power Law NHPP model, $E[N(t)] = \lambda t^{\beta}$. Use this plot to assess whether the parametric model fits the observed event history and to identify change points. Confidence bounds are based on the Fisher matrix approximation of the fitted parameters. For piecewise models, vertical dotted lines mark each breakpoint.

Value

A ‘plotly‘ object representing the interactive NHPP plot.

See Also

[plotly_mcf()] for the nonparametric MCF alone; [plotly_rga()] for a Crow-AMSAA cumulative-failures view.

Examples

```
library(ReliaGrowR)
times <- c(100, 200, 300, 400, 500)
events <- c(1, 2, 1, 3, 2)
fit <- nhpp(time = times, event = events)
plotly_nhpp(fit)

# Piecewise model with a breakpoint
times2 <- c(100, 200, 300, 400, 500, 600, 700, 800, 900, 1000)
events2 <- c(1, 2, 1, 1, 1, 2, 3, 1, 2, 4)
fit2 <- nhpp(time = times2, event = events2, breaks = 500, method = "LS")
plotly_nhpp(fit2, breakCol = "red")

# Overlay two models
plotly_nhpp(list(fit, fit2))
```

---

plotly_rel *Interactive ALT Life-Stress Relationship Plot.*

Description

Creates an interactive life-stress relationship (Arrhenius or Power Law) plot for an ‘alt‘ object. Displays the characteristic-life estimates per stress level, the fitted relationship line, optional percentile bands with shading, and an optional goal-condition marker. The ‘alt‘ object must have been processed through [WeibullR.ALT::alt.fit()] before passing to this function.

Usage

```
plotly_rel(
  alt_obj,
  showPerc = TRUE,
  showGoal = TRUE,
  showGrid = TRUE,
  main = "Life-Stress Relationship",
  xlab = "Stress",
  ylab = "Time to Failure",
  fitCol = "red",
  ptCol = "black",
  percCol = "blue",
```

```
  goalCol = "orange",
  gridCol = "lightgray",
  signif = 3,
  percentiles = c(10, 90)
)
```

Arguments

alt_obj	An object of class ‘’alt’‘ created by the ‘WeibullR.ALT‘ package and fitted with ‘alt.fit()‘.
showPerc	Show percentile lines with shading (TRUE) or not (FALSE). Default is TRUE.
showGoal	Show the goal-condition marker when one is present in ‘alt_obj‘ (TRUE) or not (FALSE). Default is TRUE.
showGrid	Show grid (TRUE) or hide grid (FALSE). Default is TRUE.
main	Main title. Default is "Life-Stress Relationship".
xlab	X-axis label. Default is "Stress".
ylab	Y-axis label. Default is "Time to Failure".
fitCol	Color of the fitted relationship line. Default is "red".
ptCol	Color of the characteristic-life scatter points. Default is "black".
percCol	Color of the percentile lines and fill. Default is "blue".
goalCol	Color of the goal-condition marker. Default is "orange".
gridCol	Color of the grid. Default is "lightgray".
signif	Significant digits for hover text. Default is 3.
percentiles	Numeric vector of percentiles to draw as reference lines with shading. Sorted internally; shading is filled between adjacent lines. Default is ‘c(10, 90)‘.

Details

Two life-stress models are supported:

- **Arrhenius**: $\eta = A\exp(E_a/(kT))$, where stress is temperature in Kelvin and the x-axis uses a reciprocal scale.
- **Power Law**: $\eta = A/S^n$, where the x-axis is log-transformed.

The `percentiles` argument controls which quantile lines are drawn with shading between adjacent lines. The goal-condition marker (`showGoal`) is drawn when the `alt` object contains a `$goal` specification from `alt.make()`.

Value

A ‘plotly‘ object representing the interactive life-stress plot.

See Also

[plotly_alt()] for the ALT probability plot using the same `alt` object.

Examples

```
library(WeibullR.ALT)
d1 <- alt.data(c(248, 456, 528, 731, 813, 537), stress = 300)
d2 <- alt.data(c(164, 176, 289), stress = 350)
d3 <- alt.data(c(88, 112, 152), stress = 400)
obj <- alt.fit(
  alt.parallel(
   alt.make(list(d1, d2, d3), dist = "weibull", alt.model = "arrhenius", view_dist_fits = FALSE),
    view_parallel_fits = FALSE
  )
)
plotly_rel(obj)
```

---

plotly_rga *Interactive Reliability Growth Plot.*

Description

The function creates an interactive reliability growth plot for one or more ‘rga‘ objects. When a list of objects is provided the models are overlaid on the same plot, each rendered in a distinct color. The plot includes cumulative failures over time, the model fit, and optional confidence bounds. Vertical lines indicate change points if breakpoints are specified in the rga object.

Usage

```
plotly_rga(
  rga_obj,
  showConf = TRUE,
  showGrid = TRUE,
  main = "Reliability Growth Plot",
  xlab = "Cumulative Time",
  ylab = "Cumulative Failures",
  pointCol = "black",
  fitCol = "black",
  confCol = "black",
  gridCol = "lightgray",
  breakCol = "black",
  signif = 3,
  cols = NULL
)
```

Arguments

rga_obj	An object of class ’rga’, or a list of such objects. Each object is created using the ‘rga()‘ function from the ‘ReliaGrowR‘ package.
showConf	Show the confidence bounds (TRUE) or not (FALSE). Default is TRUE.
showGrid	Show grid (TRUE) or hide grid (FALSE). Default is TRUE.

main	Main title. Default is "Reliability Growth Plot".
xlab	X-axis label. Default is "Cumulative Time".
ylab	Y-axis label. Default is "Cumulative Failures".
pointCol	Color of the point values. Default is "black". Used only for a single rga object; ignored when 'cols' is provided or multiple objects are supplied.
fitCol	Color of the model fit. Default is "black". Used only for a single rga object; ignored when 'cols' is provided or multiple objects are supplied.
confCol	Color of the confidence bounds. Default is "black". Used only for a single rga object; ignored when 'cols' is provided or multiple objects are supplied.
gridCol	Color of the grid. Default is "lightgray".
breakCol	Color of the breakpoints. Default is "black". Used only for a single rga object; ignored when 'cols' is provided or multiple objects are supplied.
signif	Significant digits of results. Default is 3. Must be a positive integer.
cols	Optional character vector of colors, one per rga object. When provided, each object's points, fit line, confidence bounds, and breakpoints are all drawn in the corresponding color. Recycled if shorter than the number of objects.

Details

The Crow-AMSAA (NHPP Power Law) model fits a Non-Homogeneous Poisson Process where the expected cumulative failures follow $E[N(t)] = \lambda t^{\beta}$. A $\beta < 1$ indicates reliability growth (decreasing failure rate), $\beta > 1$ indicates reliability degradation. For piecewise models, a separate set of parameters is fit within each interval defined by `breaks`, and vertical lines mark the change-point times.

Value

A 'plotly' object representing the interactive reliability growth plot.

See Also

[plotly_duane()] for the Duane MTBF growth model on the same data.

Examples

```
library(ReliaGrowR)
times <- c(100, 200, 300, 400, 500)
failures <- c(1, 2, 1, 3, 2)
rga <- rga(times, failures)
plotly_rga(rga)

times <- c(100, 200, 300, 400, 500, 600, 700, 800, 900, 1000)
failures <- c(1, 2, 1, 1, 1, 2, 3, 1, 2, 4)
breakpoints <- 400
rga2 <- rga(times, failures, model_type = "Piecewise NHPP", breaks = breakpoints)
plotly_rga(rga2, fitCol = "blue", confCol = "blue", breakCol = "red")

# Overlay two models
```

```
rga3 <- rga(c(50, 150, 250, 350, 450), c(2, 1, 3, 1, 2))
plotly_rga(list(rga, rga3))
```

---

plotly_wblr *Interactive Probability Plot.*

Description

This function creates an interactive probability plot for one or more wblr objects. When a list of objects is provided the models are overlaid on the same plot, each rendered in a distinct color. All objects must use the same distribution family. It can include confidence bounds, suspension data (single object only), and a results table.

Usage

```
plotly_wblr(
  wblr_obj,
  susp = NULL,
  showConf = TRUE,
  showSusp = TRUE,
  showGrid = TRUE,
  main = "Probability Plot",
  xlab = "Time to Failure",
  ylab = "Probability",
  probCol = "black",
  fitCol = "black",
  confCol = "black",
  intCol = "black",
  gridCol = "lightgray",
  signif = 3,
  cols = NULL
)
```

Arguments

wblr_obj	A single object of class 'wblr', or a list of such objects. This is a required argument. Each object must use the same distribution (e.g. all Weibull or all lognormal). Suspension subplots are only shown when a single object is provided.
susp	An optional numeric vector of suspension data. Default is NULL. Ignored when multiple wblr objects are provided.
showConf	Show the confidence bounds (TRUE) or not (FALSE). Default is TRUE if confidence bounds are available in the wblr object.
showSusp	Show the suspensions plot (TRUE) or not (FALSE). Default is TRUE if susp is provided.
showGrid	Show grid (TRUE) or hide grid (FALSE). Default is TRUE.

main	Main title. Default is 'Probability Plot'.
xlab	X-axis label. Default is 'Time to Failure'.
ylab	Y-axis label. Default is 'Probability'.
probCol	Color of the probability values. Default is 'black'. Used only for a single wblr object; ignored when 'cols' is provided or multiple objects are supplied.
fitCol	Color of the model fit. Default is 'black'. Used only for a single wblr object; ignored when 'cols' is provided or multiple objects are supplied.
confCol	Color of the confidence bounds. Default is 'black'. Used only for a single wblr object; ignored when 'cols' is provided or multiple objects are supplied.
intCol	Color of the intervals for interval censored models. Default is 'black'.
gridCol	Color of the grid. Default is 'lightgray'.
signif	Significant digits of results. Default is 3. Must be a positive integer.
cols	Optional character vector of colors, one per wblr object. When provided, each object's data points, fit line, and confidence bounds are all drawn in the corresponding color. Recycled if shorter than the number of objects.

Details

For Weibull distributions the y-axis uses a double-log transformation (`log(-log(1 - F))`) so that a straight line represents a perfect Weibull fit. For lognormal distributions the y-axis uses the normal quantile transformation (`qnorm(F)`), i.e., standard normal probability paper. Three-parameter Weibull (`dist = 'weibull3p'`) shifts the x-axis by the estimated location (gamma) parameter before applying the same transformation.

Goodness-of-fit is reported as R^2 for rank-regression fits or log-likelihood for MLE fits.

Value

A 'plotly' object representing the interactive probability plot.

See Also

[plotly_contour()] for MLE parameter confidence contours.

Examples

```
library(WeibullR)
library(ReliaPlotR)
failures <- c(30, 49, 82, 90, 96)
obj <- wblr.conf(wblr.fit(wblr(failures)))
plotly_wblr(obj)

suspensions <- c(100, 45, 10)
obj <- wblr.conf(wblr.fit(wblr(failures, suspensions)))
plotly_wblr(obj, suspensions,
  fitCol = "blue",
  confCol = "blue"
)
```

```
inspection_data <- data.frame(
  left = c(0, 6.12, 19.92, 29.64, 35.4, 39.72, 45.32, 52.32),
  right = c(6.12, 19.92, 29.64, 35.4, 39.72, 45.32, 52.32, 63.48),
  qty = c(5, 16, 12, 18, 18, 2, 6, 17)
)
suspensions <- data.frame(time = 63.48, event = 0, qty = 73)
obj <- wblr(suspensions, interval = inspection_data)
obj <- wblr.fit(obj, method.fit = "mle")
obj <- wblr.conf(obj, method.conf = "fm", lty = 2)
suspensions <- as.vector(suspensions$time)
plotly_wblr(obj,
  susp = suspensions, fitCol = "red", confCol = "red", intCol = "blue",
  main = "Parts Cracking Inspection Interval Analysis",
  ylab = "Cumulative % Cracked", xlab = "Inspection Time"
)
failures <- c(25, 30, 42, 49, 55, 67, 73, 82, 90, 96, 101, 110, 120, 132, 145)
fit <- wblr.conf(wblr.fit(wblr(failures), dist = "weibull3p"))
plotly_wblr(fit, fitCol = "darkgreen", confCol = "darkgreen")

# Overlay two Weibull models
obj2 <- wblr.conf(wblr.fit(wblr(c(20, 40, 60, 80, 100)), method.fit = "mle"),
                  method.conf = "lrb")
obj3 <- wblr.conf(wblr.fit(wblr(c(10, 30, 50, 70, 90)), method.fit = "mle"),
                  method.conf = "lrb")
plotly_wblr(list(obj2, obj3))
```

`reliapltr_mcp_server` *Launch a ReliaPlotR MCP Server.*

Description

Starts a Model Context Protocol (MCP) server that exposes five reliability analysis tools to MCP clients such as Claude Code and Claude Desktop.

Usage

```
reliapltr_mcp_server(...)
```

Arguments

`...`	Additional arguments passed to [mcptools::mcp_server()], such as ‘type = "stdio"‘ (default) or ‘type = "http"‘.

Details

‘fit_weibull‘ Fits a Weibull or lognormal distribution to time-to-failure data (optionally with right-censored suspensions) and returns parameter estimates and goodness-of-fit metrics.

‘fit_alt‘ Fits an Accelerated Life Test (ALT) model across multiple stress levels and returns per-stress-level parameters and the global life-stress relationship coefficients.

‘plot_weibull‘ Fits a Weibull or lognormal distribution and returns a Weibull probability plot as a plotly JSON string.

‘plot_alt‘ Fits an ALT model and returns probability and life-stress relationship plots as plotly JSON strings.

‘plot_rga‘ Fits a Crow-AMSAA reliability growth model and returns a cumulative failure plot as a plotly JSON string.

Both ‘mcptools‘ (>= 0.2.0) and ‘ellmer‘ must be installed. They are listed in ‘Suggests‘ and are not automatically installed with ReliaPlotR. ‘jsonlite‘ is also required for the ‘fit_alt‘ tool.

Registering with Claude Code: “‘bash claude mcp add -s user reliapltr – \ Rscript -e "ReliaPlotR::reliapltr_mcp_server()" “‘

Alternatively, use the bundled launcher script: “‘bash claude mcp add -s user reliapltr – \ Rscript /path/to/ReliaPlotR/mcp/server.R “‘

Value

Called for its side effect of launching a blocking MCP server process.

`tidy_alt` *Extract Tidy Parameter Estimates from an alt Object.*

Description

Returns tidy data frames of per-stress-level parameter estimates and (when available) global life-stress relationship coefficients from a fitted ‘alt‘ object. Useful for reporting ALT results in reproducible research workflows.

Usage

```
tidy_alt(alt_obj)
```

Arguments

`alt_obj`	An object of class ‘’alt’‘ created by the ‘WeibullR.ALT‘ package. Must have been processed through [WeibullR.ALT::alt.parallel()]. For ‘$relationship‘ to be non-NULL, [WeibullR.ALT::alt.fit()] must also have been called.

Details

Two life-stress models are supported. For the Arrhenius model, $\eta = \exp(C_1 + C_2 \cdot S)$ where S is stress (typically reciprocal temperature in 1/K). For the Power Law model, $\eta = \exp(C_1)/S^{|C_2|}$. Both `coef1` and `coef2` are on the log scale as returned by `alt.fit()`.

Value

A named list with two elements:

'parallel' A 'data.frame' with one row per stress level, containing columns 'stress', 'P1', 'P2', 'wt', and 'n_failures'. For Weibull models, 'P1' is the scale parameter η and 'P2' is the shape parameter β. For lognormal models, 'P1' is μ_{log} and 'P2' is σ_{log}.

'relationship' A one-row 'data.frame' with columns 'model', 'coef1', and 'coef2' (the global life-stress relationship coefficients), or 'NULL' with a message if 'alt.fit()' has not been called.

References

Nelson, W. B. (1990). *Accelerated Testing: Statistical Models, Test Plans, and Data Analysis*. Wiley.

See Also

[plotly_alt()] for the ALT probability plot, [plotly_rel()] for the life-stress relationship plot.

Examples

```
library(WeibullR.ALT)
d1 <- alt.data(c(248, 456, 528, 731, 813, 537), stress = 300)
d2 <- alt.data(c(164, 176, 289), stress = 350)
d3 <- alt.data(c(88, 112, 152), stress = 400)
obj <- alt.fit(
  alt.parallel(
    alt.make(list(d1, d2, d3), dist = "weibull", alt.model = "arrhenius",
             view_dist_fits = FALSE),
    view_parallel_fits = FALSE
  )
)
result <- tidy_alt(obj)
result$parallel
result$relationship
```

---

tidy_rga *Extract Tidy Fitted Values from an rga Object.*

Description

Returns a tidy data frame of fitted cumulative failure counts, confidence bounds, and (when available) Crow-AMSAA model coefficients from an 'rga' object. Suitable for exporting results for use in reproducible research workflows.

Usage

```
tidy_rga(rga_obj)
```

Arguments

rga_obj	An object of class ''rga'', or a list of such objects. Each object is created using the [ReliaGrowR::rga()] function.

Details

The Crow-AMSAA (NHPP Power Law) model gives the expected cumulative failures as $E[N(t)] = \lambda t^{\beta}$. A $\beta < 1$ indicates reliability growth (decreasing failure rate); $\beta > 1$ indicates degradation. The parameters are recovered from the fitted log-log linear model via `coef()`: $\lambda = \exp(\text{intercept})$ and β is the slope coefficient.

Value

A named list with two elements:

'fitted' For a single object, a 'data.frame' with columns 'time', 'cum_failures', 'fitted', 'lower', 'upper'. For a list, a list of such data frames (one per object).

'params' For a single object, a one-row 'data.frame' with columns 'lambda' and 'beta' (Crow-AMSAA Power Law parameters), or 'NULL' if the model coefficients cannot be extracted. For a list, a list of such data frames.

References

Crow, L. H. (1974). Reliability Analysis for Complex Repairable Systems. In *Reliability and Biometry*, SIAM, pp. 379-410.

See Also

[plotly_rga()] for the corresponding interactive reliability growth plot.

Examples

```
library(ReliaGrowR)
times <- c(100, 200, 300, 400, 500)
failures <- c(1, 2, 1, 3, 2)
obj <- rga(times, failures)
result <- tidy_rga(obj)
result$fitted
result$params
```

---

tidy_wblr *Extract Tidy Parameter Estimates from a wblr Object.*

Description

Returns a tidy data frame of distribution parameter estimates, goodness-of-fit metrics, and (optionally) confidence bound data from one or more fitted 'wblr' objects. This makes it straightforward to use ReliaPlotR output in reproducible research workflows — export to CSV, include in tables, or compare multiple fits.

Usage

```
tidy_wblr(wblr_obj)
```

Arguments

wblr_obj	A single object of class ‘’wblr’‘, or a list of such objects. Each object must have been fitted with [WeibullR::wblr.fit()].

Details

Parameters are returned on their natural scales. For Weibull and three-parameter Weibull (`weibull3p`) models, `param1` is the shape parameter β and `param2` is the scale parameter η; `param3` is the location parameter γ for `weibull3p` (NA otherwise). For lognormal models, `param1` is μ_{log} and `param2` is σ_{log}.

Goodness-of-fit is reported as R^2 (column value `"R2"`) for rank-regression fits, or log-likelihood (`"loglikelihood"`) for MLE fits.

Value

A named list with two elements:

‘estimates‘ A ‘data.frame‘ with one row per ‘wblr‘ object and columns ‘dist‘, ‘method_fit‘, ‘param1_name‘, ‘param1‘, ‘param2_name‘, ‘param2‘, ‘param3_name‘, ‘param3‘, ‘gof_metric‘, ‘gof_value‘, ‘method_conf‘, ‘n_failures‘.

‘bounds‘ For a single object, a ‘data.frame‘ of confidence bound data (columns ‘Datum‘, ‘unrel‘, ‘Lower‘, ‘Upper‘), or ‘NULL‘ if no confidence bounds were computed. For a list of objects, a list of such data frames (one per object).

References

Meeker, W. Q., and Escobar, L. A. (1998). *Statistical Methods for Reliability Data*. Wiley.

See Also

[plotly_wblr()] for the corresponding interactive probability plot.

Examples

```
library(WeibullR)
failures <- c(30, 49, 82, 90, 96)
obj <- wblr.conf(wblr.fit(wblr(failures), method.fit = "mle"), method.conf = "lrb")
result <- tidy_wblr(obj)
result$estimates
result$bounds

# List of objects
obj2 <- wblr.conf(wblr.fit(wblr(c(20, 40, 60, 80, 100)), method.fit = "mle"),
                  method.conf = "lrb")
result2 <- tidy_wblr(list(obj, obj2))
result2$estimates
```

Index