Package 'surviveR'

November 7, 2026

Title Tidyverse-Based Reliability and Survival Data Visualization Tool

Version 1.0.0

Description A suite of reliability/survival data visualization tools that leverage the Tidyverse Grammar of Graphics. Time-based aesthetics may be specified alongside standard non-aesthetic arguments to generate highly customizable graphs.

License GPL (>= 3)

Depends R (>= 3.5.0)

Encoding UTF-8

Imports ggplot2, tibble, survival, flexsurv, broom, rlang, dplyr, forcats, stringr, tidyr, scales, vctrs

LazyData true

Config/roxygen2/version 8.1.1

URL https://github.com/KLGORE/surviveR

BugReports https://github.com/KLGORE/surviveR/issues

Suggests knitr, rmarkdown, systemfonts

Config/testthat/edition 3

NeedsCompilation no

Author Kristen Gore [aut, cre, cph],
Samuel Jeffe [aut],
Emma Schneider [aut],
M Caputo [ctb],
Izzy Robbins [ctb],
Alexander Stonas [ctb],
Serenna Walter [ctb],
William Q. Meeker [ctb],
Luis A. Escobar [ctb],
Francis G. Pascual [ctb] (comment: Statistical Methods for Reliability Data, 2nd ed., John Wiley & Sons, 2022)

Maintainer Kristen Gore <kgore@willamette.edu>

Repository CRAN

Date/Publication 2026-11-07 14:30:02 UTC

Contents

device_repair . . . 2
geom_coxph . . . 3
geom_coxph2 . . . 5
geom_event . . . 8
geom_km . . . 10
geom_km2 . . . 13
geom_mcf . . . 16
geom_survreg . . . 19
geom_survreg2 . . . 21
geom_weibull . . . 25
mfg . . . 27
surviveR . . . 27

Index **28**

`device_repair` *Repairability dataset*

Description

Simulated dataset containing event data for 60 devices.

Usage

```
device_repair
```

Format

A data frame with 239 rows and 5 variables:

device_id Unique ID for each device

repair_time Time at which event occurs (unit=days)

status Censoring indicator. 1 = no censoring. 0 = right censoring. The time value corresponding to status=0 is the out-of-service time for the device

device_type Device SKU (A or B)

mfg_location Manufacturing location (OR, WA, or CA)

geom_coxph *Create a Cox Proportional Hazards (CoxPH) Curve Using the Tidyverse Grammar-of-Graphics*

Description

Create CoxPH curve by specifying variables in aesthetics mapping

Usage

```
geom_coxph(
  mapping = NULL,
  ...,
  conf.int = 0.95,
  failure = FALSE,
  data = NULL,
  inherit.aes = TRUE
)
```

Arguments

mapping	Aesthetics mapping with the following inputs: • time: time column • time2: optional secondary time column (e.g. time2 or event/status column specified in Surv() arguments) • treatments: optional column or vector corresponding to the treatment variable. Aliased arguments include treatment, trt, and trts. • strata: optional column or vector corresponding to the stratifying variable.
...	Non-aes graphical arguments pertaining to the primary failure/survival curve (color, linetype, linewidth, etc).
conf.int	Confidence level for displayed confidence intervals. Must be a value between 0 and 1. Values of FALSE, NULL, or NA will result in no confidence intervals being plotted.
failure	TRUE or FALSE (default). Values of TRUE will result in a failure curve being plotted.
data	The reference dataset. May be inherited from the plot data or a previous surviveR layer data.
inherit.aes	TRUE or FALSE. TRUE inherits arguments from earlier ggplot calls. Default=TRUE.

Details

The Cox proportional hazards regression model is a versatile semiparametric method commonly used to model survival data. It applies to cases in which hazards can be modeled as follows:

$$h(t) = h_0(t)\exp(b_1x_1 + b_2x_2 + \ldots)$$

Upon estimation of the model coefficients, the above formula can be reexpressed in terms of the survival probability. This expression is given by

$$S(t) = S_0(t)^{\exp(\beta)}$$
$$\log[S(t)] = \exp(\beta)\log[S_0(t)]$$

Full details of this model are found in the "Regression Models and Life-Tables" paper by D.R. Cox (1972). Note: Though this function accepts strata in its arguments, the CPH predictions are identical to that of the model in which the stratifying variable is a treatment instead. Specifying strata only impacts the line type.

Note on additional non-aesthetics arguments:
Users may customize graphical attributes of the primary CoxPH curve(s) and confidence intervals. Options include the following:

- CoxPH Curve(s): color, linewidth, linetype, alpha
- Confidence Interval(s): conf.color, conf.fill, conf.linewidth, conf.linetype, conf.alpha
  - *conf.color*: Color of confidence interval lines. Value provided must be a string.
  - *conf.fill*: Default = NULL, which indicates that confidence intervals will be color-coded by treatment. Other accepted values include the following:
    - String of color name (ex: "black")
    - TRUE indicates the color-coding fill color by treatment.
    - Values of FALSE, NA, and "transparent" may be used to omit shading.
  - *conf.linewidth*: Line width of confidence interval lines.
  - *conf.linetype*: Line type of confidence interval lines ("dotted", "dashed", "solid", etc).
  - *conf.alpha*: Opacity value between 0 and 1. 0 = transparent. 1 = fully opaque.

Value

a ggplot object

See Also

The geom_coxph2, which accepts a survival formula in lieu of variable specification using aesthetics mappings.

Note: the geom_coxph2 function allows for an input of a survival formula rather than specifying elements through the aesthetics mapping.

Examples

```
library(ggplot2); library(dplyr)

data(mfg, package = "surviveR")

# CoxPH survival plot with treatment variable
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_coxph(aes(time = ttf, time2 = status,
        treatments = mfg_location), conf.int = 0.90)


# CoxPH failure plot with treatment variable
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_coxph(aes(time = ttf, time2 = status,
        treatments = mfg_location), failure = TRUE)


# CoxPH survival plot with treatment and strata variables
mfg %>%
    ggplot() +
    geom_coxph(aes(time = ttf, time2 = status,
        treatments = device_type, strata = mfg_location)) +
    facet_grid(.~mfg_location)


# Overlaying KM curve on CoxPH fit
mfg %>%
    ggplot() +
    geom_coxph(aes(time = ttf, time2 = status,
        treatments = device_type, strata = mfg_location)) +
    geom_km(color = "black", conf.int = FALSE) +
    facet_grid(.~mfg_location)

# Note: By default, geom_km, geom_coxph, and geom_survreg
# inherit the formula and failure variables.
```

---

geom_coxph2 *Fit a Cox proportional Hazards Regression Model*

Description

Utilizes tidyverse syntax to visualize Cox proportional hazards predictions.

Usage

```
geom_coxph2(
  formula = NULL,
  ...,
  conf.int = 0.95,
  failure = FALSE,
  data = NULL,
  inherit.aes = TRUE
)
```

Arguments

formula	A formula of the form you would input into survival::coxph
...	Non-aes graphical arguments pertaining to the primary failure/survival curve (color, linetype, linewidth, etc).
conf.int	Confidence level for displayed confidence intervals. Must be a value between 0 and 1. Values of FALSE, NULL, or NA will result in no confidence intervals being plotted.
failure	TRUE or FALSE (default). Values of TRUE will result in a failure curve being plotted.
data	The reference dataset. May be inherited from the plot data or a previous surviveR layer data.
inherit.aes	TRUE or FALSE. TRUE inherits arguments from earlier ggplot calls. Default = TRUE.

Details

The Cox proportional hazards regression model is a versatile semiparametric method commonly used to model survival data. It applies to cases in which hazards can be modeled as follows:

$$h(t) = h_0(t)\exp(b_1x_1 + b_2x_2 + \ldots)$$

Upon estimation of the model coefficients, the above formula can be reexpressed in terms of the survival probability. This expression is given by

$$S(t) = S_0(t)^{\exp(\beta)}$$

$$\log[S(t)] = \exp(\beta)\log[S_0(t)]$$

Full details of this model are found in the "Regression Models and Life-Tables" paper by D.R. Cox (1972). Note: Though this function accepts strata in its arguments, the CPH predictions are identical to that of the model in which the stratifying variable is a treatment instead. Specifying strata only impacts the line type.

Note on additional non-aesthetics arguments:
Users may customize graphical attributes of the primary CoxPH curve(s) and confidence intervals. Options include the following:

- CoxPH Curve(s): color, linewidth, linetype, alpha
- Confidence Interval(s): conf.color, conf.fill, conf.linewidth, conf.linetype, conf.alpha
  - *conf.color*: Color of confidence interval lines. Value provided must be a string.
  - *conf.fill*: Default = NULL, which indicates that confidence intervals will be color-coded by treatment. Other accepted values include the following:
    - String of color name (ex: "black")
    - TRUE indicates the color-coding fill color by treatment.
    - Values of FALSE, NA, and "transparent" may be used to omit shading.
  - *conf.linewidth*: Line width of confidence interval lines.
  - *conf.linetype*: Line type of confidence interval lines ("dotted", "dashed", "solid", etc).
  - *conf.alpha*: Opacity value between 0 and 1. 0 = transparent. 1 = fully opaque.

Value

a ggplot object

See Also

geom_coxph

Examples

```
library(ggplot2); library(dplyr)

data(mfg, package = "surviveR")

# CoxPH survival plot with treatment variable
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_coxph2(
        formula = Surv(ttf,status) ~ mfg_location,
        conf.int = 0.90)

# ^ equivalent to
# mfg %>% filter(device_type == "A") %>%
#      ggplot() +
#      geom_coxph(aes(time = ttf, time2 = status,
#           treatments = mfg_location), conf.int = 0.90)


# CoxPH failure plot with treatment variable
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_coxph2(
        formula = Surv(ttf,status) ~ mfg_location,
        failure = TRUE)
```

```
# ^ equivalent to
# mfg %>% filter(device_type == "A") %>%
#      ggplot() +
#      geom_coxph2(
#           formula = Surv(ttf,status) ~ mfg_location,
#           failure = TRUE)


# CoxPH survival plot with treatment and strata variables
mfg %>%
    ggplot() +
    geom_coxph2(formula = Surv(ttf, status) ~
        device_type + strata(mfg_location)) +
    facet_grid(.~mfg_location)

# ^ equivalent to
# mfg %>%
#      ggplot() +
#      geom_coxph(aes(time = ttf, time2 = status,
#          treatments = device_type, strata = mfg_location)) +
#      facet_grid(.~mfg_location)


# Overlaying KM curve on CoxPH fit
mfg %>%
    ggplot() +
    geom_coxph2(formula = Surv(ttf,status) ~
        device_type + strata(mfg_location)) +
    geom_km2(color = "black", conf.int = FALSE) +
    facet_grid(.~mfg_location)

# Note: By default, geom_km2, geom_coxph2, and
# geom_survreg2 inherit the formula and failure variables.
```

---

geom_event *Create event plot*

Description

Create an event plot from provided data

Usage

```
geom_event(
  mapping = NULL,
  ...,
  inherit.aes = TRUE,
  order = TRUE,
  xmax = NULL,
```

```
  data = NULL
)
```

Arguments

mapping	Aesthetics mapping with the following inputs: • time: time column • time2: optional secondary time column (e.g. time2 or event/status column specified in Surv() arguments) • treatments: A single treatments or a vector of treatments. Aliased arguments include treatment, trt, and trts. • strata: A single stratum or a vector of strata values
...	Other graphical arguments pertaining to the primary failure/survival curve (color, linetype, linewidth, etc).
inherit.aes	Logical, whether to inherit graphics from the ggplot call. TRUE by default.
order	Logical, whether to order the event plot based on when it is censored. FALSE by default.
xmax	Maximum maximum time for the x-axis. Defaults to 1.1x maximum of (time1,time2) after the data is converted to interval censored format.
data	The data to use, if not provided in the ggplot call

Details

This function creates an event plot from the provided data. On an event plot, the x-axis is time, while the y-axis has each event separated. By default, the graph shows solid lines where it is known that an event does not occur, and dashed lines where an event might have occurred, based on the provided data. If exact information is known about the event it is marked by an X. Otherwise, the time period where an event could have occurred is separated by triangles enclosing the dashed area.

Value

a ggplot object

Examples

```
library(ggplot2); library(dplyr)

data(mfg, package = "surviveR")

# Univariate event plot (no treatments)
mfg %>%
    filter(device_type=="A", mfg_location =="OR") %>%
    ggplot() +
```

```
    geom_event(
        aes(time=ttf, time2=status, id=device_id)
    )


# Univariate event plot (no treatments), unsorted

mfg %>%
    filter(device_type=="A", mfg_location =="OR") %>%
    ggplot() +
    geom_event(
        aes(time=ttf, time2=status, id=device_id),
        order = FALSE
    )

# Event Plot With Treatments

mfg %>%
    filter(mfg_location == "OR") %>%
    ggplot() +
    geom_event(
        aes(time=ttf,time2=status,id=device_id, treatments=device_type)
    )

# Event Plot With Treatments & Other Customizations
mfg %>%
    filter(mfg_location == "OR") %>%
    ggplot() +
    geom_event(
        aes(time=ttf, time2=status, id=device_id, treatments=device_type),
        #size = 2
    )  +
    facet_wrap(~ mfg_location, scales = "free_y")
```

---

geom_km *Create a Kaplan-Meier Curve Using the Tidyverse Grammar-of-Graphics*

Description

Create Kaplan-Meier curve using aesthetic mapping arguments

Usage

```
geom_km(
  mapping = NULL,
  ...,
  conf.int = 0.95,
  failure = FALSE,
```

```
  data = NULL,
  inherit.aes = TRUE
)
```

Arguments

`mapping`	Aesthetics mapping with the following inputs: • time: time column • time2: optional secondary time column (e.g. time2 or event/status column specified in Surv() arguments) • treatments: optional column or vector corresponding to the treatment variable. Aliased arguments include treatment, trt, and trts. • strata: optional column or vector corresponding to the stratifying variable.
`...`	Non-aes graphical arguments pertaining to the primary failure/survival curve (color, linetype, linewidth, etc).
`conf.int`	Confidence level for displayed confidence intervals. Must be a value between 0 and 1. Values of FALSE, NULL, or NA will result in no confidence intervals being plotted.
`failure`	TRUE or FALSE (default). Values of TRUE will result in a failure curve being plotted, FALSE will result in a survival curve.
`data`	The reference dataset. May be inherited from the plot data or a previous surviveR layer data.
`inherit.aes`	TRUE or FALSE. TRUE inherits arguments from earlier ggplot calls. Default = TRUE.

Details

The `geom_km` function generates KM curves that display nonparametric survival estimates S(t) versus time (t). Users may plot cumulative failure curves in lieu of the survival probability by using the `failure=TRUE` argument. For more statistical details on the Kaplan-Meier methodology, please see chapter 3 of *Statistical Methods for Reliability Data (2nd edition)* by Meeker, Escobar, and Pascual.

Note on additional non-aesthetics arguments:
Users may customize graphical attributes of the primary KM curve(s) and confidence intervals. Options include the following:

- KM Curve(s): color, linewidth, linetype, alpha
- Confidence Interval(s): conf.color, conf.fill, conf.linewidth, conf.linetype, conf.alpha
  - *conf.color*: Color of confidence interval lines. Value provided must be a string.
  - *conf.fill*: Default = NULL, which indicates that confidence intervals will be color-coded by treatment. Other accepted values include the following:

    * String of color name (ex: "black")
    * TRUE indicates the color-coding fill color by treatment.
    * Values of FALSE, NA, and "transparent" may be used to omit shading.
  - *conf.linewidth*: Line width of confidence interval lines.
  - *conf.linetype*: Line type of confidence interval lines ("dotted", "dashed", "solid", etc).
  - *conf.alpha*: Opacity value between 0 and 1. 0 = transparent. 1 = fully opaque.

Value

a ggplot object

See Also

The geom_km2, which accepts a survival formula in lieu of variable specification using aesthetics mappings.

Examples

```
library(ggplot2); library(dplyr)

data(mfg, package = "surviveR")

# Univariate survival plot (no treatment or strata variables)
mfg %>% filter(device_type == "A", mfg_location=="OR") %>%
    ggplot() +
    geom_km(aes(time = ttf, time2 = status), conf.int = 0.90)


# Univariate failure plot (no treatment or strata variables)
mfg %>% filter(device_type == "A", mfg_location == "OR") %>%
    ggplot() +
    geom_km(aes(time = ttf, time2 = status), failure = TRUE)


# Survival plot with treatment variable
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_km(aes(time = ttf, time2 = status,
        treatments = mfg_location), conf.int = 0.90)


# Survival plot with treatment and strata variables
mfg %>%
    ggplot() +
    geom_km(aes(time = ttf, time2 = status,
        treatments = device_type, strata = mfg_location)) +
    facet_grid(.~mfg_location)


# Overriding color & removing confidence intervals (ideal for overlaying fits)
```

```
mfg %>%
    ggplot() +
    geom_km(aes(time = ttf, time2 = status,
       treatments = mfg_location),
       conf.int = FALSE, color="black")


# Overlaying KM curve on semiparametric & parametric fits
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_coxph(aes(time = ttf, time2 = status,
        treatments = mfg_location)) +
    geom_km(color = "black", conf.int = FALSE)


mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_survreg(aes(time = ttf, time2 = status,
        treatments = mfg_location)) +
    geom_km(color = "black", conf.int = FALSE) +
    facet_grid(.~mfg_location)


# Note: By default, geom_km2, geom_coxph2, and geom_survreg2
# inherit the formula and failure variables.
```

---

geom_km2 *Create a Kaplan-Meier Curve Using the Tidyverse Grammar-of-Graphics*

Description

Create Kaplan-Meier curve by providing a survival formula

Usage

```
geom_km2(
  formula = NULL,
  ...,
  conf.int = 0.95,
  failure = FALSE,
  data = NULL,
  inherit.aes = TRUE
)
```

Arguments

formula	A formula compatible with survfit(), or a survfit object

`...`	Non-aes graphical arguments pertaining to the primary failure/survival curve (color, linetype, linewidth, etc).
`conf.int`	Confidence level for displayed confidence intervals. Must be a value between 0 and 1. Values of FALSE, NULL, or NA will result in no confidence intervals being plotted.
`failure`	TRUE or FALSE (default). Values of TRUE will result in a failure curve being plotted, FALSE will result in a survival curve.
`data`	The reference dataset. May be inherited from the plot data or a previous surviveR layer data.
`inherit.aes`	TRUE or FALSE. TRUE inherits arguments from earlier ggplot calls. Default = TRUE.

Details

The `geom_km2` function generates KM curves that display nonparametric survival estimates S(t) versus time (t). Unlike its companion function `geom_km`, `geom_km2` allows users to input a survival formula instead of specifying the aesthetics mapping. Cumulative failure curves can be plotting in lieu of the survival probability by using the `failure=TRUE` argument. For more statistical details on the Kaplan-Meier methodology, please see chapter 3 of *Statistical Methods for Reliability Data (2nd edition)* by Meeker, Escobar, and Pascual.

Note on additional non-aesthetics arguments:
Users may customize graphical attributes of the primary KM curve(s) and confidence intervals. Options include the following:

- KM Curve(s): color, linewidth, linetype, alpha
- Confidence Interval(s): conf.color, conf.fill, conf.linewidth, conf.linetype, conf.alpha
  - *conf.color*: Color of confidence interval lines. Value provided must be a string.
  - *conf.fill*: Default = NULL, which indicates that confidence intervals will be color-coded by treatment. Other accepted values include the following:
    - String of color name (ex: "black")
    - TRUE indicates the color-coding fill color by treatment.
    - Values of FALSE, NA, and "transparent" may be used to omit shading.
  - *conf.linewidth*: Line width of confidence interval lines.
  - *conf.linetype*: Line type of confidence interval lines ("dotted", "dashed", "solid", etc).
  - *conf.alpha*: Opacity value between 0 and 1. 0 = transparent. 1 = fully opaque.

Value

a ggplot object

See Also

geom_km.

Examples

```
library(ggplot2); library(dplyr)

data(mfg, package = "surviveR")

# Univariate survival plot (no treatment or strata variables)
mfg %>% filter(device_type == "A", mfg_location == "OR") %>%
    ggplot() +
    geom_km2(formula = Surv(ttf,status) ~ 1, conf.int = 0.90)

# ^ equivalent to
# mfg %>% filter(device_type == "A", mfg_location == "OR") %>%
#      ggplot() +
#      geom_km(aes(time = ttf, time2 = status), conf.int = 0.90)


# Univariate failure plot (no treatment or strata variables)
mfg %>% filter(device_type == "A", mfg_location == "OR") %>%
    ggplot() +
    geom_km2(
        formula = Surv(ttf,status) ~ 1, failure = TRUE)

# ^ equivalent to
# mfg %>% filter(device_type == "A", mfg_location == "OR") %>%
#      ggplot() +
#      geom_km(aes(time = ttf, time2 = status), failure = TRUE)


# Survival plot with treatment variable
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_km2(
        formula = Surv(ttf,status) ~ mfg_location, conf.int = 0.90)

# ^ equivalent to
# mfg %>% filter(device_type == "A") %>%
#      ggplot() +
#      geom_km(aes(time = ttf, time2 = status,
#           treatments = mfg_location), conf.int = 0.90)


# Survival plot with treatment and strata variables
mfg %>%
    ggplot() +
    geom_km2(
        formula = Surv(ttf, status) ~ device_type +
                  strata(mfg_location)) +
    facet_grid(.~mfg_location)

# ^ equivalent to
# mfg %>%
#      ggplot() +
```

```
#      geom_km(aes(time = ttf, time2 = status,
#           treatments = device_type, strata = mfg_location)) +
#      facet_grid(.~mfg_location)


# Overlaying KM curve on semiparametric & parametric fits
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_coxph2(formula = Surv(ttf, status) ~ mfg_location) +
    geom_km2(color = "black", conf.int = FALSE) +
    facet_grid(.~mfg_location)


mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_survreg2(formula = Surv(ttf, status) ~ mfg_location) +
    geom_km2(color = "black", conf.int = FALSE) +
    facet_grid(.~mfg_location)

# Note: By default, geom_km2, geom_coxph2, and
# geom_survreg2 inherit the formula and failure variables.
```

`geom_mcf` *Create Mean Cumulative Function Using the Tidyverse Grammar-of-Graphics*

Description

Create MCF curve(s) using aesthetic mapping arguments

Usage

```
geom_mcf(mapping = NULL, ..., conf.int = 0.95, data = NULL, inherit.aes = TRUE)
```

Arguments

`mapping`	Aesthetics mapping with the following inputs: • time: Event times (must be numeric). • ids: Unit ID column. Alias arguments include id, Id, Ids, ID, IDS. • treatments: A single treatment or a vector of treatments. Alias arguments include treatment, trt, or trts. • strata: Grouping variable, typically used for faceting.
`...`	Non-aes graphical arguments pertaining to the MCF line (color, linetype, linewidth, etc).
`conf.int`	Confidence level for displayed confidence intervals. Must be a value between 0 and 1. Values of FALSE, NULL, or NA will result in no confidence intervals being plotted.

`data`	The data to use, if not provided in the ggplot call.
`inherit.aes`	TRUE or FALSE. TRUE inherits arguments from earlier ggplot calls. Default = TRUE.

Details

The mean cumulative function (MCF) is a nonparametric survival statistic used to model the average number of events for a repairable system. Suppose a system consists of N total system units (hereby termed "units"). Furthermore, let i index the units, and let k index all event times t. Vector $(t_1, t_2, ..., t_n)$ represents the collective event times of all units in the system.

Each raw observation minimally consists of the following:

- The unit ID: i
- The event time: t
- The risk set status for unit i at time t: $\delta_i(t)$ where values of 1 indicate that unit i is still active (i.e. in the risk set) and values of 0 indicate that unit i is no longer active (i.e. NOT in the risk set)
- The event status of unit i at time t: $d_i(t_k)$ where values of 1 indicate that an event occurred for unit i at time t_k and values of 0 indicate that an event did NOT occur for unit i at time t_k.

Letting $d_\cdot(t_k)$ denote the total number of events that happen at time t_k and $\delta_\cdot(t_k)$ represent the risk set at time t_k, the MCF estimate at time t_j may be expressed as

$$\sum_{k=1}^{n} \frac{d_\cdot(t_k)}{\delta_\cdot(t_k)} = \sum_{k=1}^{n} \bar{d}(t_k)$$

where $\bar{d}(t_k)$ is the mean cumulative number of events per unit at time t_k.

The confidence intervals geom_mcf displays are Wald estimates, which leverage asymptotic normality assumptions in their calculation. The nonparametric standard deviation for the MCF at time t_k is estimated using the following expression:

$$\sqrt{\sum_{i=1}^{n} \left\{ \sum_{k=1}^{j} \frac{\delta_i(t_k)}{\delta_\cdot(t_k)} \times [d_i(t_k) - \bar{d}(t_k)] \right\}^2}$$

Note: The statistical validity (read: bias) of the above MCF statistics hinges on a fairly strong assumption: the recurrence rate of the events is constant for all units. This assumption can be violated in common scenarios, so one must check diagnostics to verify the appropriateness of the metric.

Many of the details and notation of this page were heavily adapted from *Statistical Methods for Reliability Data (2nd edition)* by Meeker, Escobar, and Pascual. For more statistical details on the MCF methodology, please reference pages 45-47 and 522-528 of this fantastic book.

Note on additional non-aesthetics arguments:
Users may customize graphical attributes of the primary MCF curve(s) and confidence intervals. Options include the following:

- MCF Curve(s): color, linewidth, linetype, alpha

- Confidence Interval(s): conf.color, conf.fill, conf.linewidth, conf.linetype, conf.alpha

  - *conf.color*: Color of confidence interval lines. Value provided must be a string.
  - *conf.fill*: Default = NULL, which indicates that confidence intervals will be color-coded by treatment. Other accepted values include the following:
    - String of color name (ex: "black")
    - TRUE indicates the color-coding fill color by treatment.
    - Values of FALSE, NA, and "transparent" may be used to omit shading.
  - *conf.linewidth*: Line width of confidence interval lines.
  - *conf.linetype*: Line type of confidence interval lines ("dotted", "dashed", "solid", etc).
  - *conf.alpha*: Opacity value between 0 and 1. 0 = transparent. 1 = fully opaque.

Value

a ggplot object

Examples

```
library(ggplot2); library(dplyr)

data(mfg, package = "surviveR")

# Univariate MCF plot (no treatment or strata variables)
device_repair %>% filter(device_type == "A") %>%
   ggplot() +
   geom_mcf(aes(time = repair_time, ids = device_id))


# MCF plot with treatment variable
device_repair %>%
   ggplot() +
   geom_mcf(aes(time = repair_time, ids = device_id,
      treatment = mfg_location))


# MCF plot with treatment and strata variables
device_repair %>%
   ggplot() +
   geom_mcf(aes(time = repair_time, ids = device_id,
      treatment = mfg_location, strata = device_type)) +
   facet_grid(.~device_type)

# MCF plot with additional confidence interval arguments
device_repair %>% filter(device_type == "A") %>%
   ggplot() +
   geom_mcf(aes(time = repair_time, ids = device_id),
          conf.int = 0.95, conf.fill = "gray30",
          conf.linetype = "dotted", conf.linewidth = 0.5,
          conf.color = "darkred", conf.alpha = 0.15)
```

geom_survreg *Create a Parametric Survival Curve Using the Tidyverse Grammar-of-Graphics*

Description

Create a parametric survival curve using aesthetic mapping arguments

Usage

```
geom_survreg(
  mapping = NULL,
  dist = "weibull",
  ...,
  conf.int = 0.95,
  failure = FALSE,
  length.out = 1000,
  xmax = NULL,
  data = NULL,
  inherit.aes = TRUE
)
```

Arguments

mapping	Aesthetics mapping with the following inputs: • time: time column • time2: optional secondary time column (e.g. time2 or event/status column specified in Surv() arguments) • treatments: optional column or vector corresponding to the treatment variable. Aliased arguments include treatment, trt, and trts. • strata: optional column or vector corresponding to the stratifying variable.
dist	Distribution to be used in parametric survival fit. Default value: "weibull". Accepts all distributions accepted by flexsurv::flexsurvreg, including "genf","genf.orig","gengamma", "gengamma.orig", "exp", "weibull", "weibullPH", "lnorm", "gamma", "gompertz", "llogis", "exponential", and "lognormal". For a list of distributions supported by this function, please reference the documentation for flexsurv::flexsurvreg.
...	Non-aes graphical arguments pertaining to the primary failure/survival curve (color, linetype, linewidth, etc).

conf.int	Confidence level for displayed confidence intervals. Must be a value between 0 and 1. Values of FALSE, NULL, or NA will result in no confidence intervals being plotted.
failure	TRUE or FALSE (default). Values of TRUE will result in a failure curve being plotted.
length.out	Length of the time vector used in the prediction dataset being plotted.
xmax	Maximum time value on x-axis. Defaults to 1.15 times max(time) if exact or right-censored data and 1.15 times max(time2) for interval-censored data.
data	The reference dataset. May be inherited from the plot data or a previous surviveR layer data.
inherit.aes	TRUE or FALSE. TRUE inherits arguments from earlier ggplot calls. Default = TRUE.

Details

The geom_survreg function generates parametric survival curves for survival S(t) versus time (t). It leverages the flexsurvreg function in the flexsurv::flexsurv-package package. For more information on the distributions supported by geom_survreg and geom_survreg2, please see the documentation for flexsurv::flexsurvreg.

Note that geom_survreg (and, by extension, geom_survreg2) may run sluggishly with large datasets.

Note on additional non-aesthetics arguments:
Users may customize graphical attributes of the primary parametric curve(s) and confidence intervals. Options include the following:

- Parametric Curve(s): color, linewidth, linetype, alpha
- Confidence Interval(s): conf.color, conf.fill, conf.linewidth, conf.linetype, conf.alpha
  - *conf.color*: Color of confidence interval lines. Value provided must be a string.
  - *conf.fill*: Default = NULL, which indicates that confidence intervals will be color-coded by treatment. Other accepted values include the following:
    - String of color name (ex: "black")
    - TRUE indicates the color-coding fill color by treatment.
    - Values of FALSE, NA, and "transparent" may be used to omit shading.
  - *conf.linewidth*: Line width of confidence interval lines.
  - *conf.linetype*: Line type of confidence interval lines ("dotted", "dashed", "solid", etc).
  - *conf.alpha*: Opacity value between 0 and 1. 0 = transparent. 1 = fully opaque.

Value

a ggplot object

See Also

geom_survreg2, which accepts a survival formula in lieu of variable specification using aesthetics mappings.

Examples

```
library(ggplot2); library(dplyr)

data(mfg, package = "surviveR")

# Parametric survival plot with treatment variable (default: Weibull distribution)
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_survreg(aes(time = ttf, time2 = status,
        treatments = mfg_location), conf.int = 0.90)


# Lognormal survival fit with treatment variable
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_survreg(aes(time = ttf, time2 = status,
        treatments = mfg_location), dist = "lognormal")


# Parametric failure plot with treatment variable
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_survreg(aes(time = ttf, time2 = status,
        treatments = mfg_location), failure = TRUE,
        conf.int = 0.90)


# Weibull fit with treatment and strata variables
mfg %>%
    ggplot() +
    geom_survreg(aes(time = ttf, time2 = status,
        treatments = device_type, strata = mfg_location)) +
    facet_grid(.~mfg_location)


# Overlaying KM curve on top of parametric fit
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_survreg(aes(time = ttf, time2 = status,
        treatments = mfg_location)) +
    geom_km(conf.int = FALSE)

# Note: By default, geom_km, geom_coxph, and
# geom_survreg inherit the mapping (time, time2, treatments, strata) and failure variables.
```

`geom_survreg2` *Create a Parametric Survival Curve Using the Tidyverse Grammar-of-Graphics*

Description

Create a parametric survival curve by providing a survival formula

Usage

```
geom_survreg2(
  formula = NULL,
  dist = "weibull",
  ...,
  conf.int = 0.95,
  failure = FALSE,
  length.out = 1000,
  xmax = NULL,
  data = NULL,
  inherit.aes = TRUE
)
```

Arguments

formula	Survival formula
dist	Distribution to be used in parametric survival fit. Default value: "weibull". Accepts all distributions accepted by flexsurv::flexsurvreg, including "genf","genf.orig" ,"gengamma", "gengamma.orig", "exp", "weibull", "weibullPH", "lnorm", "gamma", "gompertz", "llogis", "exponential", and "lognormal". For a list of distributions supported by this function, please reference the documentation for flexsurv::flexsurvreg.
...	Non-aes graphical arguments pertaining to the primary failure/survival curve (color, linetype, linewidth, etc).
conf.int	Confidence level for displayed confidence intervals. Must be a value between 0 and 1. Values of FALSE, NULL, or NA will result in no confidence intervals being plotted.
failure	TRUE or FALSE (default). Values of TRUE will result in a failure curve being plotted.
length.out	length of the time vector created in the prediction dataset that is plotted
xmax	Maximum time value on x-axis. Defaults to 1.15 times max(time) if exact or right-censored data and 1.15 times max(time2) for interval-censored data.
data	The reference dataset. May be inherited from the plot data or a previous surviveR layer data.
inherit.aes	TRUE or FALSE. TRUE inherits arguments from earlier ggplot calls. Default = TRUE.

Details

This function allows users to visualize parametric survival fits using the ggplot Grammar of Graphics. It leverages the flexsurvreg function in the flexsurv::flexsurv-package package.

For more information on the distributions supported by geom_survreg and geom_survreg2, please see the documentation for flexsurv::flexsurvreg.

Note that flexsurvreg (and, by extension, geom_survreg2) may run sluggishly with large datasets.

Note on additional non-aesthetics arguments:
Users may customize graphical attributes of the primary parametric curve(s) and confidence intervals. Options include the following:

- Parametric Curve(s): color, linewidth, linetype, alpha
- Confidence Interval(s): conf.color, conf.fill, conf.linewidth, conf.linetype, conf.alpha
  - *conf.color*: Color of confidence interval lines. Value provided must be a string.
  - *conf.fill*: Default = NULL, which indicates that confidence intervals will be color-coded by treatment. Other accepted values include the following:
    - String of color name (ex: "black")
    - TRUE indicates the color-coding fill color by treatment.
    - Values of FALSE, NA, and "transparent" may be used to omit shading.
  - *conf.linewidth*: Line width of confidence interval lines.
  - *conf.linetype*: Line type of confidence interval lines ("dotted", "dashed", "solid", etc).
  - *conf.alpha*: Opacity value between 0 and 1. 0 = transparent. 1 = fully opaque.

Value

a ggplot object

See Also

geom_survreg

Examples

```
library(ggplot2); library(dplyr)

data(mfg, package = "surviveR")

# Parametric survival plot with treatment variable
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_survreg2(
        formula = Surv(ttf,status) ~ mfg_location,
        conf.int = 0.90)

# ^ equivalent to
# mfg %>% filter(device_type == "A") %>%
#      ggplot() +
#      geom_survreg(aes(time = ttf, time2 = status,
#           treatments = mfg_location), conf.int = 0.90)
```

```
# Lognormal survival fit with treatment variable
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_survreg2(formula = Surv(ttf,status) ~ mfg_location,
        dist = "lognormal")

# ^ equivalent to
# mfg %>% filter(device_type == "A") %>%
#      ggplot() +
#      geom_survreg(aes(time = ttf, time2 = status,
#           treatments = mfg_location), dist = "lognormal")


# Parametric failure plot with treatment variable and 90% confidence interval
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_survreg2(
        formula = Surv(ttf,status) ~ mfg_location,
        failure = TRUE, conf.int = 0.90)

# ^ equivalent to
# mfg %>% filter(device_type == "A") %>%
#      ggplot() +
#      geom_survreg(aes(time = ttf, time2 = status,
#           treatments = mfg_location),
#           failure = TRUE, conf.int = 0.90)


# Weibull fit with treatment and strata variables
mfg %>%
     ggplot() +
     geom_survreg2(
         formula = Surv(ttf,status) ~ device_type + strata(mfg_location)
     ) +
     facet_grid(.~mfg_location)

# ^ equivalent to
# mfg %>%
#      ggplot() +
#      geom_survreg(aes(time = ttf, time2 = status,
#           treatments = device_type, strata = mfg_location)) +
#      facet_grid(.~mfg_location)


# Overlaying KM curve on top of parametric fit
mfg %>% filter(device_type == "A") %>%
     ggplot() +
     geom_survreg2(formula = Surv(ttf,status) ~ mfg_location) +
     geom_km2(conf.int = FALSE)

# Note: By default, geom_km2, geom_coxph2, and
# geom_survreg2 inherit the formula and failure variables.
```

`geom_weibull` *Create a Weibull Plot*

Description

Create a Weibull Plot Using the Tidyverse Grammar-of-Graphics

The `geom_weibull` function utilizes survival::survfit to generate a Weibull plot for the provided survival data. As discussed in chapter 6 of *Statistical Methods for Reliability Data (2nd edition)* by Meeker, Escobar, and Pascual (2022), data that is well-modeled by the Weibull distribution should yield a linear log(-log(S(t)) vs. log(t) relationship, where t represents time and $S(t)$ represents the nonparametric survival probability estimate.

The geom_weibull function color-codes data points by the treatment variable. It assigns data point characters according to the strata variable.

Note on additional non-aesthetics arguments:
Users may customize graphical attributes of the data points and corresponding line fit(s). Options include the following:

- Weibull Points: color, size, alpha
- Line fit: line.color, linewidth, linetype, line.alpha
  - *line.color*: Color of fit line(s). Must be a string.
  - *linewidth*: Line width of fit line(s).
  - *linetype*: Line type of the fit line(s) ("dotted", "dashed", "solid", etc).
  - *line.alpha*: Opacity of fit line(s). 0 = transparent. 1 = fully opaque.

Usage

```
geom_weibull(
  mapping = NULL,
  ...,
  line.fit = TRUE,
  data = NULL,
  inherit.aes = TRUE
)
```

Arguments

`mapping` Aesthetics mapping with the following inputs:

- time: time column
- time2: optional secondary time column (e.g. time2 or event/status column specified in Surv() arguments)

- treatments: optional column or vector corresponding to the treatment variable. Aliased arguments include treatment, trt, and trts
- strata: optional column or vector corresponding to the stratifying variable

...	Other graphical arguments pertaining to the primary failure/survival curve (color, linetype, linewidth, etc)
line.fit	TRUE or FALSE. Indicates whether a line should be plotted for each subgroup. Default = TRUE.
data	The reference dataset. May be inherited from the plot data or a previous surviveR layer data.
inherit.aes	TRUE or FALSE. TRUE inherits arguments from earlier ggplot calls. Default = TRUE.

Value

a ggplot object

Examples

```
library(ggplot2); library(dplyr)

data(mfg, package = "surviveR")

# Univariate Weibull plot (no treatment or strata variables)
mfg %>% filter(device_type == "A", mfg_location == "OR") %>%
    ggplot() +
    geom_weibull(aes(time = ttf, time2 = status))


# Weibull plot with treatment variable
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_weibull(aes(time = ttf, time2 = status,
        treatments = mfg_location))


# Weibull plot with no line fit
mfg %>% filter(device_type == "A") %>%
    ggplot() +
    geom_weibull(aes(time = ttf, time2 = status,
        treatments = mfg_location), line.fit = FALSE)


# Weibull plot with treatment and strata variables
mfg %>% ggplot() +
    geom_weibull(aes(time = ttf, time2 = status,
        treatments = device_type, strata = mfg_location)) +
    facet_grid(.~mfg_location, scale = "free_x")
```

mfg *Pilot reliability study for device life*

Description

Simulated dataset containing observations from a fictional 90-day reliability pilot study One hundred customers elect to participate in this study. Time observations for devices that have not failed by the end of the 90-day pilot are censored.

Usage

```
mfg
```

Format

A data frame with 100 rows and 7 variables:

device_id Unique ID for each device

mfg_location Manufacturing location (OR, WA, or CA)

device_type Device SKU (A or B)

pilot_start Start date of pilot study

status Censoring indicator. 1 = no censoring. 0 = right censoring

end_date Date of last observation (either censoring date or failure date)

ttf Time till failure

surviveR *surviveR: Tidyverse-based Reliability and Survival Data Visualization Tool*

Description

A suite of reliability/survival data visualization tools that leverage the Tidyverse Grammar of Graphics. Time-based aesthetics may be specified alongside standard non-aesthetic arguments to generate highly customizable graphs.

Author(s)

Kristen Gore, Ph.D., Sam Jeffe, Emma Schneider

See Also

Useful links:

- https://github.com/KLGORE/surviveR
- Report bugs at https://github.com/KLGORE/surviveR/issues

Index

∗ **datasets**
device_repair, 2
mfg, 27

device_repair, 2

flexsurv::flexsurv-package, *20*, *22*
flexsurv::flexsurvreg, *19, 20*, *22, 23*

geom_coxph, 3, *7*
geom_coxph2, *4*, 5
geom_event, 8
geom_km, 10, *14*
geom_km2, *12*, 13
geom_mcf, 16
geom_survreg, 19, *23*
geom_survreg2, *20*, 21
geom_weibull, *25*, 25

mfg, 27

survival::coxph, *6*
survival::survfit, *25*
surviveR, 27
surviveR-package *(*surviveR*)*, 27