Package: fuzzySurvival (via r-universe)

October 15, 2026

Title Fuzzy Survival Analysis with Vague Event Times and Fuzzy Censoring

Version 0.1.0

Description Fuzzy survival analysis for data in which event times and/or failure indicators are vague, for instance when a failure is only known to have occurred during the second quarter of a year, or when it is uncertain whether a subject lost to follow-up had in fact failed. Vague information is represented by fuzzy numbers. The package provides a fuzzy product-limit estimator obtained from nested alpha-cuts of the fuzzy survival function, a fuzzy Cox proportional hazards model with fuzzy coefficient and hazard ratio estimates, and a fuzzy log-rank test. Type-reduction utilities for summarizing fuzzy estimates by single numbers, a simulation engine for validation studies, plotting, print and summary methods, and alpha-cut based confidence bands are provided. The bounds are computed at the alpha-cuts of the fuzzy event times, so the alpha-cut intervals returned for the survival function, the regression coefficients and the test statistic are nested by construction.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, graphics, grDevices, survival (>= 3.2-13)

Suggests testthat (>= 3.0.0), knitr, rmarkdown, covr

VignetteBuilder knitr

Config/testthat/edition 3

LazyData true

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Mukul Bijalwan [aut, cre] (ORCID: <https://orcid.org/0009-0001-3040-6912>), Gunjan Aggrwal [aut], Mukul Jain [aut]

Maintainer Mukul Bijalwan <mukulbijalwan555@gmail.com>

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

Date/Publication 2026-10-15 19:16:57 UTC

RemoteUrl https://github.com/cran/fuzzySurvival

RemoteRef HEAD

RemoteSha 248aa51023d1db17715e90739fb01b1be9f84dfe

Contents

alpha_cuts . . . 2
as_crisp . . . 3
as_fuzzy_number . . . 4
c.fuzzy_number . . . 5
defuzz . . . 5
defuzz.fcox . . . 7
defuzz.fkm . . . 8
exp.fuzzy_number . . . 9
fcox . . . 9
fkm . . . 11
fkm_fuzzy_surv . . . 14
flogrank . . . 14
fuzzy-arith . . . 15
fuzzy_dfs . . . 16
fuzzy_number . . . 17
fuzzy_spread . . . 18
fuzzy_surv . . . 19
is_crisp . . . 20
is_fuzzy_number . . . 21
plot.fkm . . . 21
plot.fuzzy_number . . . 22
sim_fuzzy_surv . . . 23
summary.fcox . . . 25
summary.fkm . . . 26

Index **28**

`alpha_cuts` *Alpha-cuts, support, core and membership of fuzzy numbers*

Description

`alpha_cuts()` evaluates the nested alpha-cuts, `support()` and `core()` return the 0-cut and the 1-cut, and `membership()` recovers the membership function from the alpha-cut function.

Usage

```
alpha_cuts(x, level = c(0, 0.25, 0.5, 0.75, 1))

support(f)

core(f)

membership(x, at, by = 0.001)
```

Arguments

x, f	A fuzzy_number object (f is a fuzzy number, x is generic in the underlying function).
level	Alpha level(s) at which to evaluate the cut, in [0, 1].
at	Points at which to evaluate the membership function.
by	Spacing of the level grid used by membership() and defuzz(); 1/by levels are used in total.

Details

membership() works by inverting the alpha-cut function: for a value s the membership is the highest level at which s still lies inside the alpha-cut, i.e. $\mu(s) = \sup\{\alpha : f_\alpha^- \leq s \leq f_\alpha^+\}$. Because the cut family is nested this is the usual definition, computed on a grid of 1/by levels.

Value

alpha_cuts() and support() / core() return data frames with columns id, level, lower, upper; membership() returns an n by length(at) matrix.

Examples

```
fn <- fuzzy_number_trapezoid(3, 5, 7, 9)
alpha_cuts(fn, c(0, 0.25, 1))
support(fn)
membership(fn, c(2, 4, 6, 8, 10))
```

---

as_crisp *Crisp representatives of fuzzy numbers*

Description

Turns a fuzzy number into a numeric vector, e.g. to hand over to the functions of survival, or to compare the fuzzy estimators with their crisp counterparts.

Usage

```
as_crisp(f, level = 1, method = c("midpoint", "lower", "upper"))
```

Arguments

f	A fuzzy_number.
level	Alpha level of the representative.
method	"midpoint" for the middle of the alpha-cut, "lower" or "upper" for its end points. "midpoint" at level = 1 is the middle of the core, i.e. the most plausible value.

Value

A numeric vector with the same length as f.

Examples

```
as_crisp(fuzzy_number_trapezoid(3, 5, 7, 9))
as_crisp(fuzzy_number_trapezoid(3, 5, 7, 9), level = 1, method = "lower")
```

---

as_fuzzy_number *Coerce to a* fuzzy_number*, unwrapping data frame columns*

Description

A fuzzy_number placed in a data frame becomes a list column holding one object, so that the columns of every subject stay together. This helper accepts either a fuzzy_number or such a list column and always returns a fuzzy_number, which is what the user-facing functions call internally.

Usage

```
as_fuzzy_number(x)
```

Arguments

x	A fuzzy_number, a list column holding one, or a numeric vector (promoted to a crisp fuzzy number).

Value

A fuzzy_number.

Examples

```
d <- data.frame(t = I(list(fuzzy_number_interval(4, 6))))
d$t
alpha_cuts(d$t, 1)
is_crisp(d$t)
```

c.fuzzy_number — *Combine fuzzy numbers*

Description

Combine fuzzy numbers

Usage

```
## S3 method for class 'fuzzy_number'
c(...)
```

Arguments

`...`	`fuzzy_number` objects (or numeric vectors, which are promoted to crisp fuzzy numbers) of equal length, or of length one.

Value

A `fuzzy_number` of length `max(length(...))`.

Examples

```
c(fuzzy_number_triangle(1, 2, 3), fuzzy_number_crisp(5))
```

defuzz — *Defuzzification of fuzzy numbers*

Description

A fuzzy number carries a whole family of alpha-cuts; to obtain a single crisp value a *defuzzification* (type reduction) rule is needed. All the rules implemented here are the classical type-reduction rules of fuzzy set theory, written on the level-valued-set (alpha-cut) representation used by the package. Writing $[l_j, r_j]$ for the alpha-cut at level α_j, $s_j = (l_j + r_j)/2$ for its middle and $\Delta_j = \alpha_j - \alpha_{j-1}$ for the width of the level strip:

Usage

```
defuzz(
  x,
 method = c("midrange", "centroid", "gmirl", "sgmrl", "ms", "msm", "yager", "min",
    "max"),
  level = 0.5,
  by = 0.001,
  ...
)
```

Arguments

`x`	A `fuzzy_number`, or an object with a `fuzzy_number` method (such as the fuzzy survival curve returned by `fkm_fuzzy_surv()`).
`method`	Defuzzification rule; see Details. A vector of rules returns one row per rule.
`level`	Alpha level, only used by `method = "midrange"`.
`by`	Spacing of the level grid used for the integrals.
`...`	Further arguments, such as `times` when `x` is a fitted estimator.

Details

`midrange` `s(alpha)`, the middle of the alpha-cut at level `alpha` (mean of range).

`centroid` $\sum_j \alpha_j s_j \Delta_j / \sum_j \alpha_j \Delta_j$, the centre of gravity of the membership function.

`gmirl` graded mean integration representation level, $\sum_j \alpha_j^2 s_j \Delta_j / \sum_j \alpha_j^2 \Delta_j$.

`sgmrl` simplified graded mean, $\sum_j \alpha_j s_j / \sum_j \alpha_j$.

`ms` smallest element of the supremum cut, l_1.

`msm` smallest mean of the supremum, $(l_1 + l_0)/2$.

`yager` Yager's ranking; for a continuous membership function it reduces to the middle of the core, $(l_1 + r_1)/2$.

`min`, `max` the smallest and largest element of the support.

For a symmetric fuzzy number the `centroid`, `gmirl`, `sgmrl`, `yager` and `midrange` rules all return its centre of symmetry, and `min`/`max` its symmetric end points, which makes the rules easy to check.

Passing several rules at once returns one row per rule, which is a quick way of comparing the type reductions on the same fuzzy number.

Value

A numeric vector with one value per fuzzy number, or a matrix with one row per requested rule when `method` has length greater than one.

See Also

`fuzzy_spread()`, `fkm_fuzzy_surv()`

Examples

```
fn <- fuzzy_number_trapezoid(3, 5, 7, 9)
defuzz(fn, "centroid")
defuzz(fn, "midrange", level = 0.5)

# a symmetric fuzzy number: every centred rule gives 5
defuzz(fuzzy_number_triangle(2, 5, 8), c("centroid", "gmirl", "sgmrl"))
```

defuzz.fcox *Defuzzification of fuzzy Cox coefficients*

Description

Defuzzification of fuzzy Cox coefficients

Usage

```
defuzz.fcox(
  x,
 method = c("midrange", "centroid", "gmirl", "sgmrl", "ms", "msm", "yager", "min",
    "max"),
  level = 0.5,
  by = 0.001,
  conf.int = 0.95,
  exponentiate = FALSE
)
```

Arguments

x	A fcox fit.
method	Defuzzification rule; see Details. A vector of rules returns one row per rule.
level	Alpha level, only used by method = "midrange".
by	Spacing of the level grid used for the integrals.
conf.int	Confidence level of the (fuzzy) confidence interval.
exponentiate	Return defuzzified hazard ratios instead of coefficients.

Value

A data frame with the point estimate and the fuzzy interval.

Examples

```
data(fuzzy_dfs)
fit <- fcox(fuzzy_surv(ft, fs) ~ age + arm, data = fuzzy_dfs)
defuzz(fit, "centroid", exponentiate = TRUE)
```

defuzz.fkm — *Defuzzification of an* `fkm` *fit*

Description

Defuzzification of an `fkm` fit

Usage

```
defuzz.fkm(
  x,
 method = c("midrange", "centroid", "gmirl", "sgmrl", "ms", "msm", "yager", "min",
    "max"),
  level = 0.5,
  by = 0.001,
  times = NULL
)
```

Arguments

`x`	An `fkm` fit.
`method`	Defuzzification rule; see Details. A vector of rules returns one row per rule.
`level`	Alpha level, only used by `method = "midrange"`.
`by`	Spacing of the level grid used for the integrals.
`times`	Times at which the survival probability is evaluated; defaults to the distinct times of the fit.

Value

A numeric vector with one value per requested time.

Examples

```
data(fuzzy_dfs)
fit <- fkm(fuzzy_surv(ft, fs) ~ 1, data = fuzzy_dfs)
defuzz(fit, "centroid", times = c(6, 12, 24))
```

exp.fuzzy_number *Exponential of a fuzzy number*

Description

`exp()` is increasing, so the alpha-cuts are mapped end point by end point.

Usage

```
## S3 method for class 'fuzzy_number'
exp(x)
```

Arguments

`x`	A `fuzzy_number`.

Value

A `fuzzy_number`.

Examples

```
exp(fuzzy_number_trapezoid(-1, 0, 1, 2))
```

fcox *Fuzzy Cox proportional hazards model*

Description

A Cox model for vague event times. At each alpha level the model is fitted with `survival::coxph()` to the lower and to the upper end of the time cuts, and the two fits bound the regression coefficients. The alpha-cut of the coefficient of a covariate is

$$\beta_\alpha = [\min_{a \le \alpha} \hat{\beta}_a^-, \max_{a \le \alpha} \hat{\beta}_a^+],$$

where the cumulative min/max over the levels up to α is what makes the intervals nested, so the result is a fuzzy number rather than a family of unrelated intervals. Hazard ratios are obtained by exponentiating the fuzzy coefficients (an increasing, hence membership-preserving map).

Usage

```
fcox(
  formula,
  data,
  time,
  status,
  covariates = NULL,
  levels = NULL,
  ties = "efron",
  ...
)

## S3 method for class 'formula'
fcox(
  formula,
  data,
  time,
  status,
  covariates = NULL,
  levels = NULL,
  ties = "efron",
  ...
)

## Default S3 method:
fcox(
  formula,
  data,
  time,
  status,
  covariates = NULL,
  levels = NULL,
  ties = "efron",
  ...
)
```

Arguments

`formula`	A model formula such as `fuzzy_surv(time, status) ~ age + arm`.
`data`	A data frame containing the variables of `formula`.
`time, status`	Accepted instead of `formula`.
`covariates`	A data frame or matrix of covariates, used instead of the right hand side of `formula`.
`levels`	Alpha levels at which the model is fitted.
`ties, ...`	Passed to `survival::coxph()` when the model is refitted at each end of the alpha-cuts.

Details

Confidence intervals combine the fuzzy coefficient with the standard error of the crisp (level 1) fit, and the p-values are the smallest and largest values attainable given the fuzzy coefficient and that standard error.

Value

An object of class `fcox`, a list with components

`beta`	a `fuzzy_number` of regression coefficients,
`se`	standard errors of the level 1 fit,
`var`	variance of the level 1 fit,
`fit`	the crisp `survival::coxph()` fit at level 1,
`p.value`	a `fuzzy_number` of attainable p-values.

See Also

`defuzz()`, `fkm()`, `flogrank()`

Examples

```
data(fuzzy_dfs)
fit <- fcox(fuzzy_surv(ft, fs) ~ age + arm, data = fuzzy_dfs)
summary(fit)
defuzz(fit, "centroid", exponentiate = TRUE)
```

`fkm` *Fuzzy Kaplan-Meier estimator*

Description

The fuzzy Kaplan-Meier estimator of the survival function for vague event times and/or vague failure indicators.

Usage

```
fkm(
  formula,
  data,
  time,
  status,
  group = NULL,
  levels = NULL,
  conf.type = c("log", "log-log", "plain"),
  conf.int = 0.95,
  ...
```

```
)

## S3 method for class 'formula'
fkm(
  formula,
  data,
  time,
  status,
  group = NULL,
  levels = NULL,
  conf.type = c("log", "log-log", "plain"),
  conf.int = 0.95,
  ...
)

## Default S3 method:
fkm(
  formula,
  data,
  time,
  status,
  group = NULL,
  levels = NULL,
  conf.type = c("log", "log-log", "plain"),
  conf.int = 0.95,
  ...
)
```

Arguments

`formula`	A model formula, e.g. `fuzzy_surv(time, status) ~ group`, or a `fuzzy_surv` object.
`data`	A data frame containing the variables of `formula`.
`time, status`	Accepted as an alternative to `formula`: a vector of fuzzy or crisp times, and a vector of failure indicators.
`group`	Optional grouping variable; fits a separate curve per level.
`levels`	Alpha levels at which the bounds are evaluated. Level 0 (the support, i.e. the completely vague case) and level 1 (the core, i.e. the fully plausible case) are always included.
`conf.type`	Transformation used for the confidence bands of the two bounds, as in `survival::survfit()`.
`conf.int`	Confidence level of the bands.
`...`	Further arguments passed to `survival::survfit()` or `survival::coxph()`.

Value

An object of class `fkm`, a list with, among others, the components

`time`	the distinct times at which the curves are evaluated.
`lower`, `upper`	matrices of the lower and upper survival bounds, one row per alpha level.
`levels`	the alpha levels of those rows.
`surv`	the defuzzified curve (see `defuzz()`).
`n.risk`, `n.event`	numbers at risk and of events.

Method

Let $\tilde{t}_i$ be the fuzzy event time of subject i. For an alpha level α the set of data compatible with "the true times have membership at least α" is $D_\alpha = \prod_i [\tilde{t}_i^-(\alpha), \tilde{t}_i^+(\alpha)]$. Replacing every vague time by the lower end of its alpha-cut makes the failures happen as early as possible, so the resulting Kaplan-Meier curve $\hat{S}_\alpha^-$ is the *lower* bound of the survival function, and the upper end gives the *upper* bound $\hat{S}_\alpha^+$. The alpha-cut of the fuzzy survival function is then

$$\hat{S}_\alpha(t) = [\, \hat{S}_\alpha^-(t), \hat{S}_\alpha^+(t) \,].$$

Because the time cuts are nested, the risk sets of $\hat{S}_\alpha^-$ shrink as α grows, hence $\hat{S}_\alpha^-$ is non-decreasing and $\hat{S}_\alpha^+$ non-increasing in α: the reported intervals are nested alpha-cuts of a proper fuzzy number, and they collapse onto the classical Kaplan-Meier curve when the data are crisp.

Vague failure indicators are handled by the fractional estimator: a subject with degree of belief π that the event occurred contributes a fractional number of events $d_j+ = \pi$, which reduces exactly to the classical estimator when all π are 0 or 1.

See Also

`fkm_fuzzy_surv()`, `fcox()`, `flogrank()`

Examples

```
data(fuzzy_dfs)

# overall fuzzy Kaplan-Meier curve
fit <- fkm(fuzzy_surv(ft, fs) ~ 1, data = fuzzy_dfs)
summary(fit, times = c(6, 12, 24))

# a defuzzified curve, and the alpha-cut bounds
plot(fit)
defuzz(fit, "centroid", times = 12)

# by treatment arm
fit2 <- fkm(fuzzy_surv(ft, fs) ~ arm, data = fuzzy_dfs)
plot(fit2, type = "diff")
```

fkm_fuzzy_surv — *Fuzzy survival curve as fuzzy numbers*

Description

Turns the fuzzy survival curve of an `fkm()` fit into a vector of fuzzy numbers, one per requested time, so that `defuzz()`, `alpha_cuts()` and `membership()` can be applied to the survival probability itself.

Usage

```
fkm_fuzzy_surv(x, times = NULL)
```

Arguments

`x`	An `fkm` fit.
`times`	Times at which the survival probability is evaluated. Defaults to the distinct times of the fit.

Value

A `fuzzy_number` whose alpha-cut at level `alpha` is `[S^-_alpha(t), S^+_alpha(t)]`.

See Also

`defuzz()`, `fkm()`

Examples

```
data(fuzzy_dfs)
fit <- fkm(fuzzy_surv(ft, fs) ~ 1, data = fuzzy_dfs)
s <- fkm_fuzzy_surv(fit, times = c(6, 12, 24))
alpha_cuts(s, c(0, 0.5, 1))
membership(s, 0.6)
```

flogrank — *Fuzzy log-rank test for a difference between two groups*

Description

The log-rank statistic is computed on the lower and on the upper end of the time cuts at each alpha level, giving a fuzzy chi-square statistic and hence a fuzzy p-value. When the data are crisp the bounds coincide with the classical log-rank test.

Usage

```
flogrank(formula, data, time, status, group = NULL, levels = NULL, ...)
```

Arguments

`formula`	A model formula `fuzzy_surv(time, status) ~ group`.
`data`	A data frame containing the variables of `formula`.
`time, status, group`	Accepted instead of `formula`.
`levels`	Alpha levels at which the statistic is evaluated.
`...`	Further arguments, ignored.

Value

An object of class `flogrank`, a list with the fuzzy statistic `statistic`, the fuzzy p-value `p.value`, the crisp level 1 fit `fit` and the observed and expected number of events per group.

See Also

`fkm()`, `fcox()`

Examples

```
data(fuzzy_dfs)
fit <- flogrank(fuzzy_surv(ft, fs) ~ arm, data = fuzzy_dfs)
fit
defuzz(fit$p.value, "centroid")
```

`fuzzy-arith` *Arithmetic on fuzzy numbers*

Description

Point-wise extension of the arithmetic operations through the alpha-cut representation (Zadeh's extension principle), evaluated on a grid of `1/by` levels. Division is only defined for cuts that do not contain zero.

Usage

```
## S3 method for class 'fuzzy_number'
Ops(e1, e2)
```

Arguments

`e1, e2`	`fuzzy_number` objects or numeric vectors (promoted to crisp fuzzy numbers). Their lengths must be equal, or one of them must be one.

Value

A `fuzzy_number` of the common length.

Examples

```
x <- fuzzy_number_trapezoid(1, 2, 3, 4)
x + 10
x * x
```

`fuzzy_dfs` *Simulated disease-free survival with vague event times*

Description

A synthetic example dataset of 240 subjects with a treatment arm, age and sex, generated by `sim_fuzzy_surv()` with quarterly examinations. Use it to see the whole workflow: `fkm()`, `fcox()` and `flogrank()`.

Usage

```
fuzzy_dfs
```

Format

A data frame with 240 rows and the following columns:

`id` subject identifier.

`age` age in years.

`arm` treatment arm, `"Treated"` or `"Control"`.

`sex` `"Male"` or `"Female"`.

`ft` fuzzy event time: an interval fuzzy number when the failure was only known to occur between two examinations, a crisp fuzzy number otherwise.

`fs` failure indicator, strictly between 0 and 1 when the status of a censored record is uncertain.

`true_time`, `true_status`, `observed` the unobserved quantities used to generate the fuzzy ones, useful for checking coverage.

See Also

`sim_fuzzy_surv()`, `fkm()`, `fcox()`

Examples

```
data(fuzzy_dfs)
str(fuzzy_dfs)
fit <- fkm(fuzzy_surv(ft, fs) ~ arm, data = fuzzy_dfs)
plot(fit, type = "diff")
```

`fuzzy_number` *Vague (fuzzy) quantities on the real line*

Description

A *fuzzy number* is a fuzzy set on the real line described by its nested alpha-cuts

$$F_\alpha = [\, f_\alpha^-, f_\alpha^+ \,], \quad 0 < \alpha \leq 1,$$

where f_α^- is non-decreasing and f_α^+ is non-increasing in α, F_0 is the support and F_1 is the core. Every object of class `fuzzy_number` stores such an alpha-cut function; the membership function is recovered numerically from it, so that both representations are always consistent.

Usage

```
fuzzy_number_trapezoid(l, cl, cr, r, label = NULL)

fuzzy_number_triangle(a, m, b, label = NULL)

fuzzy_number_interval(lo, hi, label = NULL)

fuzzy_number_crisp(x, label = NULL)

fuzzy_number_from_acut(acut, support = NULL, label = NULL)
```

Arguments

`l`, `cl`, `cr`, `r`	Left and right end points and core end points of the trapezoid; must satisfy `l <= cl <= cr <= r`. All must be finite.
`label`	Optional character label (e.g. a variable name).
`a`, `m`, `b`	Left end point, mode and right end point of a triangle.
`lo`, `hi`	End points of a flat (crisp-valued on the interval) fuzzy number.
`x`	Values of crisp fuzzy numbers.
`acut`	Function of a vector of alpha levels returning a list with matrices `lower` and `upper` of dimension `n` by `length(alpha)`.
`support`	Optional `n` by 2 matrix of support end points; by default the alpha-cut at level 0 is used.

Details

The package uses fuzzy numbers for two kinds of vague information in survival data: vague (interval-like) event times such as "the failure occurred in the second quarter of the year", and vague failure indicators such as "lost to follow-up, possibly with an unobserved failure".

Value

An object of class `fuzzy_number`: a list with components `acut` (alpha-cut function), `support` (`n` by 2), `label` and `n`.

Standard shapes

`fuzzy_number_trapezoid()` trapezoidal, `(l, cl, cr, r)`.

`fuzzy_number_triangle()` triangular, a special case.

`fuzzy_number_interval()` flat core: the value is plausible to degree one on the whole interval (a "vague period").

`fuzzy_number_crisp()` a point mass; a degenerate fuzzy number.

A crisp vector is accepted wherever a fuzzy number is expected and is promoted with `fuzzy_number_crisp()`, so a fuzzy estimator reproduces the classical one exactly on crisp data.

See Also

`alpha_cuts()`, `membership()`, `defuzz()`, `fkm()`, `fcox()`

Examples

```
# "the failure occurred in the second quarter" (months 4 to 6)
q2 <- fuzzy_number_interval(4, 6)
alpha_cuts(q2, c(0, 0.5, 1))

# a rounded vague time: certainly between 3 and 9, surely at 5-7
fuzzy_number_trapezoid(3, 5, 7, 9)

# membership functions
fuzzy_number_triangle(0, 1, 2)
```

`fuzzy_spread` *Width (diffusion) of fuzzy numbers*

Description

Width (diffusion) of fuzzy numbers

Usage

```
fuzzy_spread(f, level = 0)
```

Arguments

`f`	A `fuzzy_number`.
`level`	Alpha level(s).

Value

Numeric vector or matrix of widths `f^+_alpha - f^-_alpha`.

Examples

```
fuzzy_spread(fuzzy_number_triangle(1, 2, 3), c(0, 0.5, 1))
```

---

fuzzy_surv — *Survival data with vague times or vague failure indicators*

Description

`fuzzy_surv()` is the `fuzzySurvival` analogue of `survival::Surv()`: it bundles a vector of (possibly fuzzy) times with a vector of failure indicators, so that the estimators can be used inside model formulas.

Usage

```
fuzzy_surv(time, status, ...)
```

Arguments

`time`	A `fuzzy_number` vector of event times, a plain numeric vector of crisp times, or a list of length `n` of fuzzy numbers.
`status`	Numeric vector of failure indicators. Values in `[0, 1]` give the degree of belief that the event occurred; values outside `[0, 1]` are an error.
`...`	Additional components stored in the result, e.g. a grouping or stratification variable.

Details

The failure indicator may itself be vague. A numeric `status` in `[0, 1]` is read as a *degree of belief* that the subject failed, and subjects whose indicator is strictly between 0 and 1 are handled by the fractional Kaplan-Meier estimator (they enter the risk set with a fractional weight). This models records such as "lost to follow-up, last seen alive, but an unobserved event cannot be excluded".

Value

An object of class `fuzzy_surv`, a list with components `time` (a `fuzzy_number`), `status` (numeric) and `n`.

See Also

`fkm()`, `fcox()`

Examples

```
# crisp times
fuzzy_surv(c(2, 5, 7, 9), c(1, 0, 1, 1))

# vague event times: "failed during the second quarter" for subject 2
tm <- fuzzy_number_crisp(c(2, 6, 9))
tm[2] <- fuzzy_number_interval(4, 6)
fuzzy_surv(tm, c(1, 1, 1))

# vague censoring: subject 2 failed with degree of belief 0.5
fuzzy_surv(c(2, 5, 7, 9), c(1, 0.5, 1, 1))
```

---

is_crisp *Is a fuzzy number crisp?*

Description

Is a fuzzy number crisp?

Usage

```
is_crisp(f, tol = 1e-08)
```

Arguments

`f`	A `fuzzy_number`.
`tol`	Tolerance for comparing the end points of the alpha-cuts.

Value

A logical vector of length `length(f)`.

Examples

```
is_crisp(c(fuzzy_number_crisp(3), fuzzy_number_interval(1, 2)))
```

is_fuzzy_number — *Test whether an object is a fuzzy number*

Description

Test whether an object is a fuzzy number

Usage

```
is_fuzzy_number(x)

is.fuzzy_number(x)
```

Arguments

x	Any object.

Value

TRUE if x inherits from fuzzy_number.

Examples

```
is.fuzzy_number(fuzzy_number_triangle(1, 2, 3))
is.fuzzy_number(1:3)
```

plot.fkm — *Plot fuzzy Kaplan-Meier curves*

Description

Draws the alpha-cut bounds of the fuzzy survival function. With type = "diff" the fuzzy difference of two groups' curves is drawn instead, the fuzzy counterpart of a survival difference plot.

Usage

```
## S3 method for class 'fkm'
plot(
  x,
  type = c("surv", "diff", "table"),
  levels = c(0, 0.25, 0.5, 0.75, 1),
  cols = c("#2166AC", "#B2182B"),
  fill = 80,
  xlab = "Time",
  ylab = "Survival probability",
  main = "Fuzzy Kaplan-Meier estimate",
```

```
  lty = 1,
  legend = TRUE,
  times = NULL,
  ...
)
```

Arguments

x	An fkm fit (a grouped fit for type = "diff").
type	"surv" for the survival curve, "diff" for the difference between the first two groups, "table" for the numbers at risk.
levels	Alpha levels whose bounds are drawn.
cols	Colours for the lower and upper bound.
fill	Alpha transparency of the shaded region between the bounds.
xlab, ylab, main	Axis labels and title.
lty	Line type of the defuzzified curve.
legend	Whether to draw a legend.
times	Times at which the difference is evaluated.
...	Further graphical parameters.

Value

NULL invisibly; called for the side effect.

Examples

```
data(fuzzy_dfs)
fit <- fkm(fuzzy_surv(ft, fs) ~ 1, data = fuzzy_dfs)
plot(fit)

fit2 <- fkm(fuzzy_surv(ft, fs) ~ arm, data = fuzzy_dfs)
plot(fit2, type = "diff")
```

---

plot.fuzzy_number *Plot the membership function of fuzzy numbers*

Description

Plot the membership function of fuzzy numbers

Usage

```
## S3 method for class 'fuzzy_number'
plot(
  x,
  y,
  n = 20,
  main = "Membership function",
  xlab = "Value",
  ylab = "Membership",
  ...
)
```

Arguments

x	A fuzzy_number.
y	Unused, for compatibility with the generic.
n	Maximum number of fuzzy numbers to draw.
main, xlab, ylab	Plot labels.
...	Further graphical parameters.

Value

NULL invisibly; called for the side effect.

Examples

```
plot(c(fuzzy_number_trapezoid(0, 1, 2, 4), fuzzy_number_interval(5, 6)))
```

---

sim_fuzzy_surv *Simulate survival data with vague event times and vague indicators*

Description

Generates true event times from a Weibull model, applies censoring, and then coarsens the information the way a real record does: the failure is only known to have occurred between two examinations, and a fraction of the censored records carries an uncertain failure status.

Usage

```
sim_fuzzy_surv(
  n = 200,
  beta = c(age = 0.3, arm = -0.6),
  mu = log(24),
  sigma = 0.6,
  shape = 1.4,
  cov_mean = 0,
```

```
  cov_sd = 1,
  censor_max = NULL,
  censor_rate = 1/40,
  exam_every = 3,
  prop_vague = 0.6,
  prop_uncertain = 0.2,
  seed = NULL
)
```

Arguments

`n`	Number of subjects.
`beta`	Named vector of log hazard ratios, one per covariate column.
`mu`, `sigma`, `shape`	Weibull parameters: $T = \exp(\mu + \sigma W)$ with W Weibull of the given shape.
`cov_mean`, `cov_sd`	Mean and standard deviation of the covariates; a two-level covariate (a treatment arm) is created when any name of `beta` contains `"arm"`.
`censor_max`, `censor_rate`	Censoring: a uniform cut-off on `censor_max` or an exponential rate `censor_rate` on the observed time scale.
`exam_every`	Spacing of the examinations, in time units. A failure between two examinations is reported as a vague time spanning them.
`prop_vague`	Proportion of the failures that are reported as vague (interval) times rather than exact ones.
`prop_uncertain`	Proportion of the censored records whose failure status is uncertain; their degree of belief is drawn uniformly on `[0.4, 0.9]`.
`seed`	Optional seed for reproducibility.

Details

The true times and statuses are returned alongside the fuzzy ones, so that the coverage of the fuzzy estimators can be checked directly.

Value

A data frame with the covariates, the fuzzy time column `ft`, the fuzzy status column `fs`, and the `true_time`/`true_status` used to generate them.

See Also

`fkm()`, `fcox()`

Examples

```
set.seed(1)
d <- sim_fuzzy_surv(n = 60, beta = c(age = 0.4, arm = -0.5))
head(d[, c("arm", "fs")])
fit <- fkm(fuzzy_surv(ft, fs) ~ arm, data = d)
plot(fit, type = "diff")
```

summary.fcox *Summaries and plots of a fuzzy Cox fit*

Description

Summaries and plots of a fuzzy Cox fit

Usage

```
## S3 method for class 'fcox'
summary(
  object,
  conf.int = 0.95,
  exponentiate = FALSE,
  digits = 4,
  method = "centroid",
  ...
)

## S3 method for class 'fcox'
print(x, conf.int = 0.95, ...)

## S3 method for class 'fcox'
coef(object, exponentiate = FALSE, ...)

## S3 method for class 'fcox'
plot(
  x,
  type = c("hr", "membership"),
  conf.int = 0.95,
  xlab = "Hazard ratio",
  main = "Fuzzy Cox model",
  legend = TRUE,
  ...
)
```

Arguments

object	A fcox fit.
conf.int	Confidence level of the reported intervals.
exponentiate	Report hazard ratios instead of coefficients.
digits	Number of significant digits.
method	Defuzzification rule, see defuzz().
...	Further arguments, ignored by summary().
x	A fcox fit.

type — Either "hr" for a forest plot of the fuzzy hazard ratios, or "membership" for their membership functions.

xlab, main, legend — Plot labels, and whether to draw a legend.

Value

summary() returns a list with the coefficient table and the number of subjects and events; the plot method returns NULL invisibly.

Examples

```
data(fuzzy_dfs)
fit <- fcox(fuzzy_surv(ft, fs) ~ age + arm, data = fuzzy_dfs)
summary(fit)
plot(fit)
```

---

summary.fkm *Summaries of a fuzzy Kaplan-Meier fit*

Description

Summaries of a fuzzy Kaplan-Meier fit

Usage

```
## S3 method for class 'fkm'
summary(
  object,
  times = NULL,
  digits = 4,
  method = "centroid",
  conf.int = 0.95,
  alpha = 0,
  ...
)

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

Arguments

object — An fkm fit.

times — Times at which to report the curves.

digits — Number of significant digits.

method — Defuzzification rule, see defuzz().

`conf.int`	Confidence level of the reported bands.
`alpha`	Alpha level of the reported band, `0` being the widest.
`...`	Further arguments, ignored by `summary()`.
`x`	The data frame returned by `summary()`, used by its print method.

Value

A data frame with one row per time and columns for the alpha-cut bounds `lower`/`upper`, the defuzzified value `surv`, the numbers at risk, of events and censored, and the band on the defuzzified curve.

Index

∗ **datasets**
fuzzy_dfs, 16

alpha_cuts, 2
alpha_cuts(), *14*, *18*
as_crisp, 3
as_fuzzy_number, 4

c.fuzzy_number, 5
coef.fcox *(*summary.fcox*)*, 25
core *(*alpha_cuts*)*, 2

defuzz, 5
defuzz(), *3*, *11*, *13*, *14*, *18*, *25*, *26*
defuzz.fcox, 7
defuzz.fkm, 8

exp.fuzzy_number, 9

fcox, 9
fcox(), *13*, *15*, *16*, *18*, *19*, *24*
fkm, 11
fkm(), *11*, *14–16*, *18*, *19*, *24*
fkm_fuzzy_surv, 14
fkm_fuzzy_surv(), *6*, *13*
flogrank, 14
flogrank(), *11*, *13*, *16*
fuzzy-arith, 15
fuzzy_dfs, 16
fuzzy_number, 17
fuzzy_number_crisp *(*fuzzy_number*)*, 17
fuzzy_number_crisp(), *18*
fuzzy_number_from_acut *(*fuzzy_number*)*, 17
fuzzy_number_interval *(*fuzzy_number*)*, 17
fuzzy_number_trapezoid *(*fuzzy_number*)*, 17
fuzzy_number_triangle *(*fuzzy_number*)*, 17
fuzzy_spread, 18
fuzzy_spread(), *6*
fuzzy_surv, 19

is.fuzzy_number *(*is_fuzzy_number*)*, 21
is_crisp, 20
is_fuzzy_number, 21

membership *(*alpha_cuts*)*, 2
membership(), *3*, *14*, *18*

Ops.fuzzy_number *(*fuzzy-arith*)*, 15

plot.fcox *(*summary.fcox*)*, 25
plot.fkm, 21
plot.fuzzy_number, 22
print.fcox *(*summary.fcox*)*, 25
print.fkm_summary *(*summary.fkm*)*, 26

sim_fuzzy_surv, 23
sim_fuzzy_surv(), *16*
summary.fcox, 25
summary.fkm, 26
support *(*alpha_cuts*)*, 2
survival::coxph(), *9–12*
survival::Surv(), *19*
survival::survfit(), *12*