Package 'cyclomort'

May 23, 2025

Type Package

Title Survival Modeling with a Periodic Hazard Function

Version 1.0.3

Date 2025-05-13

Description Modeling periodic mortality (or other time-to event) processes from right-censored data. Given observations of a process with a known period (e.g. 365 days, 24 hours), functions determine the number, intensity, timing, and duration of peaks of periods of elevated hazard within a period. The underlying model is a mixed wrapped Cauchy function fitted using maximum likelihoods (details in Gurarie et al. (2020) <doi:10.1111/2041-210X.13305>). The development of these tools was motivated by the strongly seasonal mortality patterns observed in many wild animal populations. Thus, the respective periods of higher mortality can be identified as ``mortality seasons".

License GPL (>= 3)

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports flexsurv, lubridate, magrittr, mvtnorm, plyr, scales, stats, survival

URL https://github.com/EliGurarie/cyclomort

BugReports https://github.com/EliGurarie/cyclomort/issues

Suggests knitr, rmarkdown, ggplot2

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Eliezer Gurarie [aut, cre], Thompson Peter [aut]

Maintainer Eliezer Gurarie <egurarie@esf.edu>

Repository CRAN

Date/Publication 2025-05-23 19:42:05 UTC

Contents

`censor_cycloSurv` *Censor and Trim*

Description

Functions for right-censoring and left-trimming survival data. They are convenient for comparing cyclomort fits before and after some cut-off time, as in the example below.

Usage

```
censor_cycloSurv(x, censor.time)

trim_cycloSurv(x, trim.time)
```

Arguments

`x`	cycloSurv object
`censor.time`	time of (right) censoring, or vector of times of censoring
`trim.time`	time of (left) trimming

Value

Censored Surv object

Trimmed Surv object

Examples

```
## load Western Arctic Herd data and convert to cycloSurv
data(wah_morts)
wah <- with(wah_morts, create_cycloSurv(start = start, end = end,
                                        event = fate == "dead", period = 365))

# censor and trim
cutoff = "2016-01-01"
wah_pre = censor_cycloSurv(wah, censor.time = cutoff)
wah_post = trim_cycloSurv(wah, trim.time = cutoff)

# combine into dataframe
par.init <- par(no.readonly = TRUE)

par(mfrow = c(1,2))
plot(wah_pre[,1], 1:length(wah_pre), xlim = range(wah_pre[,1:2]), type= "n", main = "pre")
segments(wah_pre[,1], 1:length(wah_pre), wah_pre[,2], 1:length(wah_pre), col = wah_pre[,3]+1)
plot(wah_post[,1], 1:length(wah_post), xlim = range(wah_post[,1:2]), type= "n", main = "post")
segments(wah_post[,1], 1:length(wah_post), wah_post[,2], 1:length(wah_post), col = wah_pre[,3]+1)

# fit seasonal model before and after
wah_fit_pre <- fit_cyclomort(wah_pre, n.seasons = 1)
wah_fit_post <- fit_cyclomort(wah_post, n.seasons = 1)

# some evidence of a shift, though confidence intervals are wide
summary(wah_fit_pre)
summary(wah_fit_post)


par(mfrow = c(1,2))
plot(wah_fit_pre, plotCI = TRUE, breaks = 10); title("pre cut-off")
plot(wah_fit_post, plotCI = TRUE, breaks = 10); title("post cut-off")

par(par.init)
```

`create_cycloSurv` *Create a cycloSurv object*

Description

`cycloSurv` is a superclass of `Surv`, the standard data type for survival analysis in R, with an additional period attribute necessary for estimating periodic hazard functions.

Usage

```
create_cycloSurv(start, end, event, t0 = NULL, period, timeunits = "days")
```

Arguments

start	a vector measuring time an individual enters a population (can be POSIX, numeric, or Date)
end	a vector measuring time an individual leaves a population, e.g. via death (or other precipitation event of interest) or censoring. (as a POSIXct, numeric, or Date)
event	the status indicator, normally 0=alive/censored, 1=dead.
t0	reference time for event times. By default, t0 is set to January 1 of the first year of observations if times are POSIXct. There are many reasons why a biological year may more conveniently start on a different day. All else being equal, it can be useful to start a "mortality year" at a period of low mortality to better isolate the seasons of higher mortality.
period	length of one period in the input data
timeunits	units that dates are inputted in if dates are being used

Value

an object of class cycloSurv which is identical to and compatible with a 'Surv object, with, however, an addition "period" attribute.

Examples

```
startTimes = as.Date(origin = "2010-01-01",
                     c(0, 0, 0, 50, 0, 50, 100, 150, 0, 100)) #in days
endTimes = as.Date(origin = "2010-01-01",
                   c(50, 50, 100, 150, 150, 200, 200, 250, 350, 500)) #in days
censored = c(1, 1, 0, 1, 1, 0, 1, 0, 0, 0)
period = 365
morts = create_cycloSurv(start = startTimes, end = endTimes,
                         event = censored, period = period)
```

---

factorfit_cyclomort *Factorial analysis of seasonal survival models*

Description

This function takes a Y~X style formula to compare null models of pooled data against separately fitted models against a given factor. For now this works only for a single discrete factor.

Usage

```
factorfit_cyclomort(f, data = NULL, n.seasons = 2, ...)
```

Arguments

f	formula object used for identifying different classes
data	a data frame containing a cycloSurv object detailing mortalities for a set of observations and a factor identifying the value of a categorical variable for each observation
n.seasons	number of seasons to fit model to
...	additional arguments to fit_cyclomort call

Value

table comparing outputs from null (factor has no effect on mortality and they are all in the same group) model to multi-factor model using AIC, log-likelihood and likelihood ratio test

Examples

```
# fit factorial model
data(seasonalsex)
seasonalsex.factorfit <- factorfit_cyclomort(event ~ sex, data = seasonalsex, n.seasons = 1)

# summary
summary(seasonalsex.factorfit, coefs = TRUE)
plot(seasonalsex.factorfit)
```

---

findDelta *Converting between Rho to Delta*

Description

Functions for converting the concentration parameter rho to the season duration parameter delta and vice versa. They are: `findDelta(rho)`. These are mainly internal.

Usage

```
findDelta(rho)

findRho(delta)

DeltaToRho(delta, rho)
```

Arguments

rho	concentration parameter on interval [0, 1]
delta	duration parameter

Value

duration parameter delta

concentration parameter rho on interval [0, 1]

Examples

```
findDelta(rho = 0.9); findRho(0.0167)
findDelta(rho = 0.1); findRho(0.218)

# Plot the relationship
oldpar <- par(no.readonly = TRUE)
par(mfrow = c(1,2))
rhos <- seq(0, 1, length = 1e3)
plot(rhos, findDelta(rhos), ylab = "deltas", type = "l")
deltas <- seq(0, .5, length = 1e3)
plot(deltas, findRho(deltas), ylab = "rhos", type = "l")
par(oldpar)
```

---

fit_cyclomort *Estimate periodic hazard function.*

Description

This function takes time-to-event data formatted as a `cycloSurv` object and estimates an underlying hazard function for a given number of seasons.

Usage

```
fit_cyclomort(
  x,
  inits = NULL,
  n.seasons = 2,
  method = "L-BFGS-B",
  period = NULL
)
```

Arguments

`x`	a cycloSurv object recording start and end times as well as status (dead/censored) and the length of one full period
`inits`	set of initial guesses; a named vector or list with values for "peak" and "duration". Leaving some or all of these parameters as NULL will trigger the automatic selection of an initial guess.
`n.seasons`	number of seasons to fit model to
`method`	method for optim call
`period`	expected periodicity of survival data. Can be passed in with cycloSurv input parameter

Value

a cmfit object containing parameter estimates for peaks, durations, and weights for each season

Examples

```
# Simulate data
T.morts1 <- simulate_cycloSurv(1000, period = 365,
                               meanhazard = 0.3 / 365,
                               peaks = c(0.25 * 365, 0.75 * 365),
                               durations = c(0.3 * 365, 0.1 * 365),
                               weights = c(0.7,0.3),
                               plotme = FALSE)

# Estimate simulated data
fits <- fit_cyclomort(T.morts1, n.seasons = 2)
fits

# Plot results
plot(fits, nreps = 1000, monthlabs = TRUE)
# NB: `nreps` is for the bootstrap of the confidence interval
# The default (5000) is slower but smoother

# Actual parameter values from simulated data
attributes(T.morts1)
```

---

guess_initial_parameters

Produce initial parameter estimates based on mortality data

Description

Uses a basic flexsurvreg exponential mortality model to find the average hazard value, and fits a mixed normal distribution model to estimate the peaks, season durations, and weight distributions for the model. These estimates are not meant to be fully accurate but instead are meant to be good initial guesses for the fit_cyclomort function.

Usage

```
guess_initial_parameters(x, n, null_fits)
```

Arguments

x	cycloSurv object representing time of death or censorship
n	expected number of mortality seasons within a period
null_fits	original estimate for mortality rate assuming constant hazard function

Value

a named vector of guesses for parameter values, used to initialize the fitting process

loglike — *Obtain log-likelihood value from a data set given a set of parameter values*

Description

Obtain log-likelihood value from a data set given a set of parameter values

Usage

```
loglike(x, gammas, mus, rhos)
```

Arguments

x	a cycloSurv object
gammas	k-vector of average hazard values for each component
mus	k-vector of peaks
rhos	k-vector of concentration parameters

Value

the maximum likelihood value for this set of data

loglike_optim — *Log-likelihood function*

Description

Internal function used for computing the log-likelihood of a parameterized model within `fit_cyclomort`.

Usage

```
loglike_optim(pars, x)
```

Arguments

pars	named vector including "gamma", "mu", and "rho" parameters for the appropriate number of seasons
x	times of death or censoring as Surv objects

Value

likelihood value given named vector of parameters as well as set of observations

See Also

`fit_cyclomort`

nwt_morts *Mortality data for Northwest territory boreal woodland caribou.*

Description

Mortality data for Northwest territory boreal woodland caribou, anonymized and randomized by year, thereby retaining the multi-seasonal signal without, with grateful acknowledgements to A. Kelly and N. Larter.

Usage

```
data(nwt_morts)
```

Format

Data frame with 370 rows and the following columns:

id ID of animal

start Date of beginning of collaring

end Date of death or censoring

status "Mort" or "Cens" (dead or censored)

Source

Government of Northwest Territories, Canada

Examples

```
data(nwt_morts)
require(ggplot2); require(magrittr); require(plyr)
ggplot(nwt_morts %>% arrange(start) %>% mutate(id = factor(id, levels = id)),
aes(x = start, y = id, col = status)) +
  geom_errorbarh(aes(xmin = start, xmax = end))
```

plot.cmfactorfit *Plot cmfactorfit objects*

Description

Plot cmfactorfit objects

Usage

```
## S3 method for class 'cmfactorfit'
plot(x, fit = "both", colors = NULL, legend = TRUE, ...)
```

Arguments

x	a cmfactorfit object
fit	a character (either "null", "alt", or "both") that dictates what fits will be plotted
colors	vector of colors (one component for each individual fit being plotted) for the hazard estimates
legend	boolean parameter dictating whether or not a legend will be added to the plot
...	additional parameters to pass to the plot.cmfit function. Perhaps most usefully: lowering the default nreps (e.g. to 1000) makes plotting much faster.

Value

a plot comparing the hazard estimates from the null model with the individual estimates from each factor level

Examples

```
# fit factorial model
data(seasonalsex)
seasonalsex.factorfit <- factorfit_cyclomort(event ~ sex, data = seasonalsex, n.seasons = 1)

# summary
summary(seasonalsex.factorfit, coefs = TRUE)
plot(seasonalsex.factorfit)
```

---

plot.cmfit *Plot cmfit objects*

Description

Plot cmfit objects

Usage

```
## S3 method for class 'cmfit'
plot(
  x,
  plotCI = TRUE,
  CI.level = 0.95,
  histogram = TRUE,
  add = FALSE,
  monthlabs = FALSE,
  nreps = 5000,
  hazcolor = "black",
  alpha = 0.3,
  ymax = NULL,
  prediction = NULL,
```

```
  yaxt = par()$yaxt,
  ...
)
```

Arguments

x	a cmfit object
plotCI	whether confidence intervals should also be drawn.
CI.level	confidence level (default 0.95) for CIs (if CI is TRUE)
histogram	boolean dictating whether a histogram of actual mortalities will be included in the plot
add	boolean dictating whether the plot will be added to an existing plot
monthlabs	whether or not to label the x-axis with months - suitable for (common) annual seasonal data. If FALSE, labels are numeric within the period [0,1]
nreps	number of samples from parameter estimates for confidence intervals (see predict.cmfit)
hazcolor	color of lines for hazard function and confidence intervals
alpha	transparency of confidence interval polygon
ymax	maximum value for the y-axis - can be useful for scaling purposes
prediction	an optional predict.cmfit object- otherwise the function will estimate this every time which can be a bit slow.
yaxt	location for y-axis label
...	additional parameters to hist (e.g., number of breaks)

Value

a plot comparing the estimated mortality curve (based on parameter estimates) and the actual results (as a histogram).

See Also

predict.cmfit

Examples

```
# Simulate data
T.morts1 <- simulate_cycloSurv(1000, period = 365,
                               meanhazard = 0.3 / 365,
                               peaks = c(0.25 * 365, 0.75 * 365),
                               durations = c(0.3 * 365, 0.1 * 365),
                               weights = c(0.7,0.3),
                               plotme = FALSE)

# Estimate simulated data
fits <- fit_cyclomort(T.morts1, n.seasons = 2)
fits

# Plot results
```

```
plot(fits, nreps = 1000, monthlabs = TRUE)
# NB: `nreps` is for the bootstrap of the confidence interval
# The default (5000) is slower but smoother

# Actual parameter values from simulated data
attributes(T.morts1)
```

`predict.cmfit` *Prediction method for cyclomort fits*

Description

Obtain predictions and confidence intervals for the hazard function or the time to event from a fitted cyclomort object.

Usage

```
## S3 method for class 'cmfit'
predict(
  object,
  ...,
  t = seq(0, object$period, length = 500),
  type = "hazard",
  CI = FALSE,
  CI.level = 0.95,
  nreps = 1000
)
```

Arguments

`object`	a cmfit object
`...`	(not implemented)
`t`	times for prediction. By default, covers 100 observations over a single period.
`type`	either `hazard` or `timetoevent` - dictates what exactly will be predicted
`CI`	a boolean dictating whether or not to compute confidence intervals
`CI.level`	confidence level (default 0.95) for CIs (if CI is TRUE)
`nreps`	number of samples drawn to generate confidence intervals. The default 10^3 is generally sufficient, and very fast for the hazard function, but possibly prohibitively slow for the time-to-event functionality.

Details

Confidence intervals are produced by sampling from the multivariate normal distribution of the MLE parameter estimates accounting for the covariance in the estimates by using the Hessian of the MLE.

Value

a list of vectors containing predictions for each value in t, as well as (optional) confidence intervals.

Examples

```
# simulate two-peak mortality process
sim.morts <- simulate_cycloSurv(300, period = 1, peaks = c(0.3, 0.8),
                      durations = c(0.15, 0.20), weights = c(3, 2)/5,
                      meanhazard = 1, plotme = FALSE, max.periods = 6)
sim.morts <- simulate_cycloSurv(300, period = 365, peaks = c(0.3, 0.8)*365,
                                durations = c(0.15, 0.20)*365, weights = c(3, 2)/5,
                                meanhazard = 1/365, plotme = FALSE, max.periods = 6)

# estimate parameters
sim.morts.fit <- fit_cyclomort(sim.morts, n.seasons = 2)

# compute predictions for one moment in time (with 95% confidence interval)
predict(sim.morts.fit, CI = TRUE, type = "hazard")

# compute predictions for a range of times
predict(sim.morts.fit, t = 1:365, CI = FALSE, type = "hazard")

# these predictions are used (internally) in the plot.cmfit method:


plot(sim.morts.fit, CI.level = 0.95, months = FALSE, histogram = FALSE, monthlabs = TRUE)
plot(sim.morts.fit, CI.level = 0.8, months = FALSE, histogram = FALSE, add = TRUE)
plot(sim.morts.fit, CI.level = 0.5, months = FALSE, histogram = FALSE, add = TRUE)


# predict time to event given a start at times (this is a very slow calculation!)


timetoeventprediction <- predict(sim.morts.fit, t = seq(1,365,3), type = "timetoevent",
                       CI = TRUE, nreps = 1e2)


# the following object contains a prediction
data(timetoeventprediction)

with(timetoeventprediction, {
  plot(t, fit, type = "l", lwd = 2,  main = "expected time to event",
       ylim = c(100,365), ylab = "days")
  lines(t, CI[1,], lty = 3)
  lines(t, CI[2,], lty = 3)
})
```

---

seasonalsex *Simulated data of seasonal mortality data for two sex groups*

Description

See examples below for the process of simulating and visualizing these data using `simulate_cycloSurv`, and an example of analyzing these data with `factorfit_cyclomort`.

Usage

```
data(seasonalsex)
```

Format

Simulated data of single-season mortalities for two sex groups:

sex female (F) or male (M)

event `cycloSurv` object of (censored) survival data

Examples

```
# useful packages
require(ggplot2); require(magrittr); require(plyr)

# Example of simulating multi-factor data:
## Not run:**
n <- 100
T.male = simulate_cycloSurv(n, period = 1, meanhazard = 0.3, peaks = .25, durations = .3)
T.female = simulate_cycloSurv(n, period = 1, meanhazard = 0.3, peaks = .75, durations = .3)
T.joint <- with(rbind(T.male, T.female) %>% data.frame,
          create_cycloSurv(start = start, end = stop,
                           event = status, period = 1))
seasonalsex <- data.frame( sex = rep(c("M","F"), each = n), T = T.joint)
## End(**Not run**)

# load and visualize simulated sex-specific survival data
data("seasonalsex")

seasonsex.df <- cbind(seasonalsex, as.matrix(seasonalsex$event) %>% as.data.frame) %>%
  arrange(sex,stop) %>% mutate(id = 1:length(start) %>% factor,
                               status = c("Dead", "Censored")[2-status])

require(ggplot2)
ggplot(seasonsex.df, aes(x = start, y = id, col = status)) +
  geom_errorbarh(aes(xmin = start, xmax = stop)) +
  facet_wrap(.~sex, scales = "free_y", ncol = 1) +
  ggtitle("Simulated sex-specific mortality data")

seasonsex.df$time.trunc <- with(seasonsex.df, stop - floor(stop))
with(seasonsex.df, {
  hist(time.trunc[sex == "M"], col = rgb(0,0,1,.3), breaks = seq(0,1,.1),
       bor = NA, freq = FALSE, xlab = "Time (within period)",
       main = "Male vs. Female (simulated) mortalities")
  hist(time.trunc[sex == "F"], col = rgb(1,0,0,.3), breaks = seq(0,1,.1),
       bor = NA, add = TRUE, freq = FALSE)
  lines(density(time.trunc[sex == "M"], from = 0, to = 1), col = "darkblue", lwd = 2)
```

```
  lines(density(time.trunc[sex == "F"], from = 0, to = 1), col = "darkred", lwd = 2)
  legend("topleft", fill = c("blue", "red"), legend = c("M","F"), title = "Sex")
})

# test differences

sex.fit <- factorfit_cyclomort(event ~ sex, data = seasonalsex, n.seasons = 1)
summary(sex.fit)

plot(sex.fit, ymax = 1.3)
```

`select_seasons` *Select the number of mortality seasons*

Description

Compute a delta AIC table (and, optionally, likelihood ratio tests) for a sequence of models with a different number of seasons

Usage

```
select_seasons(x, max.season = 4, lrt = FALSE, print = TRUE)
```

Arguments

`x`	`create_cycloSurv` object
`max.season`	maximum number of seasons to fit
`lrt`	whether or not to perform and return the complete results of nested likelihood ratio tests
`print`	boolean parameter; if TRUE the function prints the table out as a side effect of creating the object

Value

a list containing (1) a list of all the fitted objects, and (2) an AIC (and, optionally, LRT) summary table. Also prints both tables by default.

Examples

```
T.morts1 <- simulate_cycloSurv(1000, period = 1,
                               meanhazard = 0.3,
                               peaks = c(0.25, 0.75),
                               durations = c(0.2, 0.1),
                               weights = c(0.3, 0.7),
                               plotme = FALSE)
```

```
model_selection = select_seasons(T.morts1, max.season = 4)
summary(model_selection$fits)
```

`simulate_cycloSurv` *Simulate periodic mortality process*

Description

Simulate periodic mortality process

Usage

```
simulate_cycloSurv(
  n,
  period = 1,
  meanhazard = 0.5,
  peaks = c(0.25, 0.75),
  durations = c(0.2, 0.1),
  weights = c(0.5, 0.5),
  censoring = "random",
  censor.times = max.periods * period/2,
  max.periods = 10,
  n.times = 1000,
  plotme = TRUE
)
```

Arguments

`n`	number of simulated mortality/censoring events
`period`	length of one mortality cycle
`meanhazard`	average hazard value
`peaks`	k-vector of peaks
`durations`	k-vector of season length parameters, based on concentration parameter from wrapped Cauchy distribution
`weights`	k-vector of weights ((k-1)-vector is also accepted)
`censoring`	the type of censoring in the simulated data. Either "none" (all data is uncensored), "fixed" (all data is censored at a specified time), or "random" (data is randomly censored throughout).
`censor.times`	numeric or vector listing times for censoring (only applicable if censoring == "fixed").
`max.periods`	maximum number of cycles
`n.times`	number of x-values for plots (a higher value results in more precision for curves)
`plotme`	if TRUE, produces a set of plots for the simulation to display its accuracy

Value

a cycloSurv object (a subclass of a Surv object; see Surv)

Examples

```
par.init <- par(no.readonly = TRUE)
par(oma = c(2,0,2,0))
T.morts1 <- simulate_cycloSurv(1000, period = 1,
                               meanhazard = 0.3,
                               peaks = c(0.25, 0.75),
                               durations = c(0.2, 0.1),
                               weights = c(0.3, 0.7),
                               plotme = TRUE)

with(attributes(T.morts1),
     title(paste0("mean hazard: ", meanhazard, "; peaks: ",
                  paste(peaks, collapse = ",")), outer = TRUE))

par(oma = c(2,0,2,0))
T.morts2 <- simulate_cycloSurv(300, period = 365,
                               meanhazard = 0.5/365,
                               peaks = c(100, 250),
                               durations = c(20, 40),
                               weights = c(0.4, 0.6),
                               plotme = TRUE,
                               max.periods = 5)

with(attributes(T.morts2),
     title(paste0("mean hazard: ", round(meanhazard, 3), "; peaks: ",
                  paste(peaks, collapse = ",")), outer = TRUE))



par(mfrow = c(1,1))
require(magrittr)
h <- with(as.matrix(T.morts1) %>% data.frame %>% subset(status == 1),
    hist(stop - floor(stop), breaks = 20, col = "grey", bor = "darkgrey"))

with(attributes(T.morts1), curve(mwc(x, mus = peaks,
                                     rhos = findRho(durations), gammas = weights,
                                     tau = period)* mean(h$counts), add = TRUE))
par(par.init)
```

---

summary.cmfactorfit *Summary method for cyclomort factorial fit*

Description

Summary method for cyclomort factorial fit

Usage

```
## S3 method for class 'cmfactorfit'
summary(object, ..., coefs = FALSE)
```

Arguments

object	a cmfactorfit object - the output of factorfit_cyclomort.
...	(not implemented)
coefs	whether or not to report the individual summaries of each model component along with the statistical test results

Value

a table comparing log-likelihood and AIC between null and multi-factor model, and a p-value from likelihood ratio test, optionally combined with the individual model summaries.

Examples

```
# fit factorial model
data(seasonalsex)
seasonalsex.factorfit <- factorfit_cyclomort(event ~ sex, data = seasonalsex, n.seasons = 1)

# summary
summary(seasonalsex.factorfit, coefs = TRUE)
plot(seasonalsex.factorfit)
```

---

summary.cmfit *Provide a short summary of cmfit (parameter estimates for periodic mortality curves) objects*

Description

Provide a short summary of cmfit (parameter estimates for periodic mortality curves) objects

Usage

```
## S3 method for class 'cmfit'
summary(object, date = FALSE, ...)
```

Arguments

object	a cmfit object
date	logical dictating whether peaks of high mortality are expressed as Dates
...	(not implemented)

Value

a list containing a short summary of the estimates for each parameter along with confidence intervals and AIC

Examples

```
# Simulate data
T.morts1 <- simulate_cycloSurv(1000, period = 365,
                               meanhazard = 0.3 / 365,
                               peaks = c(0.25 * 365, 0.75 * 365),
                               durations = c(0.3 * 365, 0.1 * 365),
                               weights = c(0.7,0.3),
                               plotme = FALSE)

# Estimate simulated data
fits <- fit_cyclomort(T.morts1, n.seasons = 2)
fits

# Plot results
plot(fits, nreps = 1000, monthlabs = TRUE)
# NB: `nreps` is for the bootstrap of the confidence interval
# The default (5000) is slower but smoother

# Actual parameter values from simulated data
attributes(T.morts1)
```

---

summary.cmfitlist *Summary method for cmfitlist objects*

Description

Summary method for cmfitlist objects

Usage

```
## S3 method for class 'cmfitlist'
summary(object, ..., coefs = TRUE)
```

Arguments

object	a cmfitlist object - output of select_seasons
...	(currently not implemented)
coefs	whether or not to return model coefficients along with statistic test table.

Value

a data frame describing the AIC, log-likelihood, number of parameters and parameter estimates for each model

Examples

```
T.morts1 <- simulate_cycloSurv(1000, period = 1,
                               meanhazard = 0.3,
                               peaks = c(0.25, 0.75),
                               durations = c(0.2, 0.1),
                               weights = c(0.3, 0.7),
                               plotme = FALSE)


model_selection = select_seasons(T.morts1, max.season = 4)
summary(model_selection$fits)
```

---

timetoeventprediction *Example fitted time to event prediction*

Description

Example fitted time to event prediction

Usage

```
data(timetoeventprediction)
```

Format

An object of class `list` of length 6.

Examples

```
## Code that generates this object - following example in vignette
set.seed(10)
T.morts.sim <- simulate_cycloSurv(300, period = 365, meanhazard = 1/365,
peaks = c(100, 250), durations = c(25, 40), weights = c(0.4, 0.6), plotme = FALSE)
T.morts.fit <- fit_cyclomort(T.morts.sim, n.seasons = 2)
timetoeventprediction <- predict(T.morts.fit, t = 1:365,
                                 type = "timetoevent", CI = TRUE, nreps = 100)

data(timetoeventprediction)
with(timetoeventprediction, {
plot(t, fit, type = "l", ylab = "Time to event", ylim = range(CI), lwd = 2)
lines(t, CI[1,])
lines(t, CI[2,])})
```

wah_morts *Mortality data for Western Arctic Herd Caribou*

Description

Anonymized mortality data on Western Arctic Herd caribou collected by the U.S. National Park Service, Alaska, with grateful acknowledgments to K. Joly.

Usage

```
data(wah_morts)
```

Format

Data frame with 171 rows and the following columns:

id ID of animal

start Date of beginning of collaring

end Date of death or censoring

fate One of "dead", or "censored"

Source

U.S. National Park Service, Alaska

Examples

```
data(wah_morts)
require(ggplot2); require(magrittr); require(plyr)
ggplot(wah_morts %>% arrange(start),
aes(x = start, y = id, col = fate)) +
  geom_errorbarh(aes(xmin = start, xmax = end))
```

wc *Wrapped Cauchy and Integrated Wrapped Cauchy functions*

Description

Fundamental periodic hazard function, mixed hazard function, and their (analytical) integrals.

Usage

```
wc(t, mu, rho, tau)

iwc(t, mu, rho, tau)

mwc(t, mus, rhos, gammas, tau)

imwc(t, mus, rhos, gammas, tau)
```

Arguments

t	time (numeric, can be vectorized)
mu	mean peak
rho	concentration parameter (0 <= rho <= 1)
tau	period
mus	k-vector of mean peaks (assuming k seasons)
rhos	k-vector of concentration parameters
gammas	k-vector of average hazard values for each component

Details

These functions are mainly internal. wc and iwc are both parameterized in terms of peak mean μ, concentration parameter ρ, and period τ and are "unweighted", i.e.

$$\int_0^\tau f(t)dt = \tau$$

The mixture model versions, mwc and imwc, are correspondingly parameterized in terms of vectors mus, rhos, and also gammas which correspond to the mean hazard contribution of each peak, such that

$$\int_0^\tau f(t)dt = k\gamma\tau$$

Value

numeric value (or vector of values of same length as t) of the respective function

Examples

```
# wrapped Cauchy functions
curve(wc(x, mu = 100, rho = .7, tau = 365), xlim = c(0,365), n = 1e4,
      ylab = "hazard", xlab = "time")
curve(wc(x, mu = 100, rho = .5, tau = 365), add = TRUE, col = 2)
curve(wc(x, mu = 100, rho = .3, tau = 365), add = TRUE, col = 3)

# mixed wrapped Cauchy functions
curve(mwc(x, mus = c(0.125, 0.5), rhos = c(0.7, 0.5),
          gammas = c(2, 1), tau = 1), xlim = c(0,1), ylab = "hazard", xlab = "time")
```

```
curve(mwc(x, mus = c(0.25, 0.75), rhos = c(0.3, 0.8),
          gammas = c(0.6, 0.4), tau = 1), add = TRUE, col = 2)
curve(mwc(x, mus = c(0.25, 0.5, 0.75), rhos = c(0.6, 0.5, 0.4),
          gammas = c(0.5, 0.2, 0.3), tau = 1), add = TRUE, col = 3)
```

Index